# Computing

## A Level

# Computing

## Exemplar Candidate Work

H447

Unit F454 Computing Project A

October 2015

www.ocr.org.uk

OCR
Oxford Cambridge and RSA

# Contents

# Introduction

This sample material serves as a general guide. It provides the following benefits to a teacher:

- Gives teachers an appreciation of the variety of work that can be produced for this unit
- Shows how the mark scheme has been applied by a senior assessor
- Provides examples of both good and weak application of different parts of the mark scheme
- Provides real examples of work submitted for F454.

It is important to make the point that the teacher support materials play a secondary role to the Specification itself. The Specification is the document on which assessment is based and specifies what content and skills need to be covered in delivering the course. At all times, therefore, this teacher support should be read in conjunction with the Specification. If clarification on a particular point is sought then that clarification should be found in the Specification itself.

This project is an example which has not been scaled or adjusted during the moderation process; as such the marks awarded have been viewed as being 'within tolerance' when assessed by a moderator.

# URS

OCR RECOGNISING ACHIEVEMENT

GCE

## Computing

### Advanced GCE H447 Unit F454

### Coursework Cover Sheet

Please read the instructions printed overleaf before completing this form. One of these cover sheets, suitably completed, should be attached to the assessed work of **each** candidate in the moderation sample.

| | | | |
|---|---|---|---|
| **Examination session** | JUNE | **Year** | 2014 |
| **Centre name** | [redacted] | | |
| **Centre number** | [redacted] | | |
| **Candidate name** | [redacted] | **Candidate number** | [redacted] |

**(a) Definition, Investigation and Analysis**

| | Max Mark | Mark |
|---|---|---|
| (i) Definition - nature of the problem to be investigated | 3 | 2 |
| (ii) Investigation and Analysis | 11 | 9 |
| Total | 14 | 11 |

**(b) Design**

| | Max Mark | Mark |
|---|---|---|
| (i) Nature of the solution | 6 | 5 |
| (ii) Algorithms | 5 | 3 |
| (iii) Test strategy | 5 | 3 |
| Total | 16 | 11 |

**(c) Software Development, and Testing**

| | Max Mark | Mark |
|---|---|---|
| (i) Software Development | 16 | 15 |
| (ii) Testing | 14 | 12 |
| Total | 30 | 27 |

**(d) Documentation**

| | Max Mark | Mark |
|---|---|---|
| | 10 | 9 |
| Total | 10 | 9 |

**(e) Evaluation**

| | Max Mark | Mark |
|---|---|---|
| (i) Discussion of the degree of success in meeting the original objectives | 4 | 4 |
| (ii) Evaluate the users' response to system | 3 | 3 |
| (iii) Desirable extensions | 3 | 3 |
| Total | 10 | 10 |
| Total for project | 80 | 68 |

**CCS297** Devised February 2008 CCS/F454

Oxford Cambridge and RSA Examinations

# Identify

My Client is a biology teacher at a Sixth Form College, a college that specialises in the teaching of higher education. The teacher is one of 13 teaching staff in the biology department which teaches higher level biology to students at the college.

The teacher is in charge of updating the School's Biology VLE (called Moodle) page. I decided to consult the teacher are he's the person who could probably help me more in terms of testing and giving me feedback on things to help improve upon that the other members of the department may not pick up on.

At the moment, the biology teachers have to teach the fieldwork techniques, needed as part of overall grade at A-level Biology, outside. If the weather is bad, it's near impossible to try and do it and so they'd need to put it off until the next lesson. Consequently, they'll get behind and may not be given as much time as other groups to get to grips with the techniques used and how to measure the variables they need to. A mathematical element to this part of the specification is rather heavy as well and can be very difficult to students to grasp. I saw this as an opportunity to help the department by creating a program that could be used inside, by simulating the techniques and ways of performing the field work and to help with the teaching of the statistics that shows what the data received implies.

I approached the teacher and asked whether he would be interested in a program that could be used on days where the weather is poor and can be used by students who are still unsure about what to do without needing to get all the equipment out and having to do the whole process again, saving time for the teacher to continue with their lesson as planned and for the student to have some extra time to become comfortable with it.

# Investigation

## Research Plan:

The means of collecting data I have chosen are; to observe the way it's taught now, write a questionnaire to be completed by the teaching staff and, lastly, to conduct an interview with my end user, The teacher.

## Observation:

I chose to observe the way the biology field work is taught now, first, because it can give me a better understanding of what features I feel may be suitable to include in a new system and it may help me see what aspects of the work seem most confusing to the students and ways I could go about designing a system that would aid in understanding these concepts. Also, an observation is useful to me as I can see how the teaching may be limited in some way that a new system could fix, making it easier to teach and perhaps creating a broader, more diverse teaching system.

I'll conduct this observation on xx/xx/xx because this is the date that the students go to do their fieldwork. I can see their weaknesses and strengths during this time and how the teacher interacts with them and what methods seem to work. This will be greatly beneficial in creating a system that's useful and helpful.

I chose to do the observation first because it seems like the best method of understanding the current system without having spoken to anyone about it and will give me an insight into what are the negatives and positives are from an unbiased point of view. This means I have some initial knowledge that I can use in my interview as a foundation for the system and from that, I can develop are more sophisticated system that can do more of what my end user would like.

## Questionnaire:

Secondly, I decided to go with a questionnaire. Rather unconventional but I'm not making this program for just one person, I'm making it for the whole biology department and so I would like all of their input as to what they would like to see in the program. I think a questionnaire would be suitable because I can gather a lot of information quickly and it'll be fairly concise and not subject to a lot of opinion. This means I can process the information easier and see what the most desired features of new system are and what I should prioritise over other features. Whilst a questionnaire is good in terms of not being subject to too much opinion, this can also be a negative aspect of a questionnaire as I'm limiting the choices the users can use and so I lose some the more personal approach I could have with an interview. I'd need to choice my questions very carefully to try and get the information that I want but also so that the information I get is not too vague and isn't ambiguous.

To ensure that I choose the right questions, I'll draw upon what I learnt during my observation to see what I think needs to be looked at and from this, I can make questions to be answered by the teaching staff to see what they feel.

After I've collected the data, I'll process it in Excel and create graphs to show the potential correlations in the data and it provides a visual representation of the data which can be useful in spotting trends and showing what's popular amongst the staff and what is not.

## Interview

The third and final process of collecting data I will do is an interview. This allows me a more personal approach the attaining data and can be more spontaneous than a questionnaire. What I mean by this is that something rather unexpected could come up in an interview because of the freer environment its set in and because there's no restriction on what to say and how much you can say, I can get a better understanding of what direction I should take the system in. Also, I can ask more in depth questions that require more than a simple "yes" or "no" answer to them which means
that I'll get more information and can see what path my end user would like the system to go in.

Unfortunately an interview can take time, especially if you're interviewing many people, so I think I'll interview my end user, the teacher, because I've already got the whole department to fill in a questionnaire so I shouldn't need to interview all of them. Also, the teacher is the one most likely to be of help so it's more beneficial if I interview him over anyone else I could potentially interview.

As well as allowing me to ask any spontaneous questions that may arise from something I hadn't taken into consideration, it'll allow me to see any fault or difficulties they have that I may have missed or overlooked.

The types of question's I'll ask will be more design orientated. I hope to get most of my information from the observation and the questionnaire so the interview is more on the specifics of the system that I'd like to confirm, as well as any potential problems I still have that I'd like clarified.

By asking design questions about the system, I get the user to really think about the system and what they think would be a beneficial feature and what may be best to leave out. This means I can be sure that what I'm designing is what they definitely want as it is they who have effectively designed it with what they want; I'm just bringing their wants to life.

I think I'll also set up an intermediate interview when things begin to fall into place. This is to show what I have completed already and, although it won't appeal to them aesthetically, will perform the fundamental actions that they're after and show the basic operations of the system. This is where any design improvements will be made and anything deemed unnecessary will be left out to create the system that they absolutely want. I'll set up this interview via email nearer to when the time approaches where I would like their feedback.

## Creating the Questionnaire:

I want to create a questionnaire for the whole (or most) department to fill in that will help give me a rough idea of what
I should ask in the interview or what aspects I think would be good to focus on. I'll send an email, to the teacher, asking for permission to hand out the questionnaires. Once I have the results, I'll process the results in excel.

Hi,
For the new system I proposed a few weeks ago, I'd like to hand a questionnaire round to the department to gather their thoughts on what they think would be some useful features to have and what their preferences are. I was wondering whether this would be ok. The questionnaire should take no more than 5 minutes to complete and will be fairly basic in terms of the responses.

I could attach the questionnaire in an email and perhaps you could reply with the responses but otherwise, I think it may be easier if I printed out hardcopies of the questionnaire to hand out.

Thank you,

The above piece of text is what I intend to send to the teacher. I tried to remain somewhat formal to be respectful.

# Questionnaire

For my computing coursework, I asked if I could create a system that the Biology department could use to help with the teaching of A2 biology Fieldwork. This computerised system would be used when the weather is perhaps too poor to go out or if you would like an electronic method of calculating and teaching $Chi^2$ and perhaps a method of creating your own examples and saving them for later use.

I would like to have your thoughts and opinions on this system and decided to create this questionnaire to be filled in. It should take no more than 5 minutes to complete and the feedback will be invaluable.

Thank you,

1. For a new computerised system, what aspects would you like to be emphasised to greater aid in teaching? (check all that apply)
   - € Using the Quadrats and using random numbers
   - € How to use the multimetre and techniques to standardise the results
   - € Transects and the different methods of collecting the data from it (interrupted etc.)
   - € Soil testing
   - € Suitable table and graphs
   - € Calculating $Chi^2$ and what it shows
2. Would you like simulations/ animations and other interactive elements to make the system more entertaining and helpful?
   - € Yes
   - € No
   - € Maybe
3. I intend to include a feature to create your own examples and save them for later use or perhaps a feature to generate a random set of results. Would you like to see this implemented in:
   - € Soil testing (e.g. generating random numbers and allowing you to input an answer)
   - € Calculating $Chi^2$ with random numbers for more variety
   - € To fill in a table to be displayed
   - € All of the above. A feature that goes through the whole process step by step

4. Would you like to include a quiz/ test at the end or after each section?
    - € Yes
    - € No
    - € Maybe
5. Would you like to have a database of sorts that would allow you to save, edit and delete student's records about how they are progressing?
    - € Yes
    - € No
    - € Maybe
6. If yes to the above, would you like to have a student log in so that they can view their progress and any feedback you leave for them? You would have your own separate log in also where you can edit everything and have access to all parts of the program.
    - € Yes
    - € No
    - € Maybe
7. Would you like to have lots of visual aids as well as text and quizzes? These would include backgrounds and proper images rather than just block colour shapes. Or do you think this may be too distracting?
    - € Yes
    - € No
    - € Use a limited use of graphics to maintain more text
8. Should the font be large, bold etc. or would you like it normal? There could be a range of both to highlight key areas or to highlight words they should know.
    - € A mixture of both
    - € All just one size, one font and one colour
    - € A mixture of both but with different colours to draw attention to differences (may want to differentiate between headings and terminology etc)

Here's a copy of my questionnaire that I printed out.

## Arranging the Interview:

To arrange the interview, I sent an email to the teacher, asking when would be a good time for him to have this interview.

Hi,
I would like to arrange an interview with you to discuss the new computer system and also the methods you use to teach the A2 Biology Fieldwork. I have a basic idea of features that could be implemented and that I think may be beneficial but I would like your expertise on the matter.

Thank you,

**Current method of teaching?**
Does every teacher follow the same teaching method or are some vastly different? What do you think students find the most difficult if you had to choose?

Refer to questionnaire questions:
- Emphasis?
- Simulations?
- Random results creator
- Quiz/test at end?
- Images?
- Text?
- Database

Hardware constraints:
- Everyone use same laptops?

**Any questions for me to answer?**

Here's a copy of my interview plan. As you can probably see, it's not very detailed or in depth about what I'm going to say. I want the interview to be more of an interaction with the end user rather than a script that we both follow. I also made it very loose and this lack of rigidity will create room for spontaneity and will allow for more open interaction. I feel this would be more beneficial as I can really get to see what it is that my end user would want out of this new system.

## Interview Transcript:

This is a transcript of the interview I had with the teacher.

Me: The first question I want to ask is about the method of teaching the field work. Is it the same for every teacher or do they follow a set kind of guidelines that they have to go through or is it completely different depending on who teaches it?

The teacher: we try to have a fairly common framework so what you've experienced in the field will pretty much be the same as every other student. There'll be 3 projects. One of them will involve a line transect, for use with Spearman's Rank. So that's targeted to a particular stats method. Another one will use the top versus bottom counting exercise and you can use that for Chi2. And then the other one involves the difference in an abiotic factor. And you probably did shade versus light.

Me: Yeah

The teacher: And that will be for Standard Error. So we do have to make sure that we all cover the same territory. But the type of exercise we do in the field, which matches the statistical methods. Because at the end of it all, all those statistical methods are what you need to be able to do. Be confident that yo u can use in the EMPA exam next year.

Me: Ok. Yeah that makes sense.

The teacher: So it's targeted at actually something that you'll be examined on.

Me: Yeah, ok.

The teacher: So it does have to be a fairly rigid framework. There is some leeway for some people to do different areas
of the site. For example, my transect, I did it in a completely different place to everyone else because I know the site quite well from a different syllabus that I taught a few years back. I know where I can get suitable results. So…

Me: Yeah, Yeah. So if I made this system and a teacher used it, they wouldn't feel like it was completely different?

The teacher: Oh no, no. We all pretty much use the same sites to collect our data.

Me: Ok that's good. That was my main concern.

The teacher: I've got a good overhead picture of the site if you need to use it on Google Maps.

Me: Yeah I might do that.

The teacher: It's a satellite so it's a good resolution.

Me: Yeah, Yeah that's quite a good idea actually. So do you know where all the teachers tend to go or is it just random for each one?

The teacher: For each project?

Me: Yeah, I know its top and bottom of hill and the transects…

The teacher: Oh. Well the top and bottom of hill is very flexible in terms of where you actually sample. The light and shade one is a bit more of a problem, cos there's a, only a limited number of places where you can do that on either side of those wooded areas. Though, everybody would probably do that in exactly the same place. But it, if you were going to refer to those in your project, everyone would know where you meant, I'm sure; especially if you have a little map or something.

Me: Yeah I was thinking of having a map and sort of circle the areas that you'd go to…

The teacher: Yeah I can help you to pinpoint where they would be.

Me: Yeah, that would be good.

Me: The secondly, when you teach like what do you find the most difficult aspect of the syllabus is like the hardest to grasp for students?

The teacher: Do you mean in this particular section of the syllabus?

Me: Yeah just this particular section.

The teacher: well I think it is the statistics side of it, because many students haven't continued with a subject that is maths based. And their maths skills have sort of fallen by the waste side a little bit. Some students are very up with their maths skills and are quite happy when you teach them statistics but there's a good, high percentage of the, you know, probably more than half of the biology group where it's quite difficult to get them to understand the concepts in the stats and what you're actually doing and the terminology. What do we mean by Standard Deviation and Standard Error? Those sorts of things are difficult to get over. So I think that's the most difficult area in this part of the syllabus to convey to the students who don't have a maths background or don't have it continued with a maths based subject.

Me: Yeah so if I was to include that in the system, you'd like that to be kind of very concise.

The teacher: It would have to be concise and would have to use the same language as we all try to use and if you use the literature that you've been given as a student in biology, you'll get a very clear idea of what that, that terminology is and you'll be able to use it in your work.

Me: And obviously that is in the exam, like in the EMPA's.

The teacher: Yeah we try to use the same terminology as would be encountered by the students in the exam yeah.

Me: Ok yeah that's good.

Me: I wanted to refer back to the questionnaires and a few questions like what would you want to be emphasised most out of anything in this part of the syllabus?

The teacher: Well, our original idea of this was the idea of having a simulation of randomly distributed coverage in a model quadrat. So that's the sort of the thing we'd like to model so that it would be possible for students to get lots of practise at actually working with that. And developing that skill which they then would feel confident about at using when they go out into the field. So that's our key area that we want to use this piece of software for. Any other recording of the user's performance, and so on, would be something that would be useful for me to feedback to them so that they can then have another try. And hopefully being their performance closer to the actual figures that the software is generating. I think that's the key emphasis that we want out of this piece of software. But if you have any other branches that students could then use, I see from your questionnaire, building in ideas about using statistics and so on and that's fine too. It doesn't have to be linear. The student can choose areas, or the teacher who's asking the student to use the software, has, it isn't necessarily going to be a linear path.

Me: Yeah I was thinking of having a menu based selection at the beginning so that they can choose between things
like a quadrat simulator or statistics.

The teacher: So I could say to my students, I'd like you to model to random sampling. I'd like you to do 20 quadrats, and I'd like you to take the results that you get from here at this site and then go into the stats part of the software and use the figures that you've generated to do the stats. That would be useful. That would be useful. Is that what you envisaged?

Me: yeah I was thinking of that.

The teacher: Yeah that's something that would be nice. But certainly not a linear so that once you've done this you now have to go onto this.

Me: Yeah I wasn't thinking of doing that because I know it's not, if someone's good at getting the percentage cover then it will be tedious for them to have to go through it over and over again.

The teacher: So you could have the option to use their own data that they've generated in the quadrat side of it. But also perhaps you could have some stock files of data also in there.

Me: Yeah I was thinking of like a default kind of. A few default files so that if they don't want to do the counting and the percentage cover, they can just go straight to the statistics and there will be data there for them.

The teacher: What would be useful I guess for us, as the teachers, is to be able to add more different default files into the system.

Me: Yeah I was going to get onto that next.

The teacher: Oh sorry. Go ahead.

Me: I thought there could be a kind of database with a log in thing and then you'd be able to log in and set your own examples…

The teacher: Oh yeah that would be lovely. That would be great. Because in fact, we could take data from the previous year or we could make up some data. That would be great, yeah.

Me: I was also thinking of doing the log in thing so that there'd be a student log in so that they'd be able then see how they've progressed through. Like with the percentage cover, we could have it so that you have to get it as close to the value that the computer has to predict it to be and they can see how they're progressing.

The teacher: Yeah that sounds good too. Yeah, yeah.

Me: Also, another thing I want to ask is about hardware constraints. So does, do all the teachers use the same, like a PC or do some use a MAC or something?

The teacher: No we all have Windows based laptops…

Me: Ok that's good.

The teacher: and we would want perhaps to use those to demonstrate your package so that students would become familiar with it on the whiteboard. That would be one aspect that would be important so that you could show the students what you want them to do. And the idea that they could then go and do that independently on any machine in college.

Me: Yeah I was thinking…

The teacher: Is what we would like.

Me: Yeah I was thinking of perhaps getting it on the college, like on the biology…

The teacher: On our little menu

Me: yeah on the little menu thing or on moddle or something so they can access it at home if they want extra practise outside of lessons.

The teacher: Oh yeah sure. I'm not sure how the students stand on Remote Access. Do our students all have remote access to their actual account here?

Me: You can do. You can go on the sixth form website and it has remote access and then moodle and stuff so they can access it from home.

The teacher: Because I know staff have but I wasn't aware of the status of students was on.

Me: Yeah there's a software called VMWare.

The teacher: Yeah that's right. That's the one we use. So as long as you are happy to install VMWare, you can have, as a student, remote access to…

Me: yeah like if you feel you need extra practise outside of lessons then yeah, you can do that.

The teacher: Yeah ok.

The teacher: I'm not sure a lot of students are aware of that. Certainly I find it a little bit, um, I wouldn't use the word unreliable but a little bit odd at home. When I installed it on one of my machines at home, the VMWare, it actually prevented my anti-virus from functioning.

Me: Oh?

The teacher: I don't know what the actual cause was but maybe they shared a driver or something, I don't know. My anti-virus, like most these days, requires access to the providers website, or the providers severs. And that connection was severed when I had VMWare installed on my machine.

Me: Yeah I do find it can be a bit unreliable at times. And I've heard that if you do too many transferring of files and saving stuff in VMWare, it can lose some stuff as well. But I've only heard that from a few people.

The teacher: when I, when I removed VMWare from that machine, everything went back to normal so you know, it was that which was the cause.

Me: Hmm yeah.

The teacher: it works perfectly well on another machine I've got running. Which is windows XP. The machine I had the problem with was a windows 7 machine.

Me: ohhh.

The teacher: which was, umm, quite odd.

The teacher: anyway, we're getting quite, quite off topic there

Me: yeah we are. Ok then, lastly, do you have any extra questions… that you want to ask me about?

The teacher: umm… not at this point. I think you've, you've put forward a lot of ideas there that we've tried to respond to and hopefully you've now got a clearer picture of what your development work needs to be. Can you give us any sort of time frame for completion or anything like that?

The teacher: what, what do you have to work to?

Me: I think it's around sort of December/ Christmas time and it should be completed, or at least nearing the stages of completion. Then I think it might have to get moderated. But I'm not sure if the actual software itself gets looked at or if a copy gets made and then they…

The teacher: and in-between now and that period in December, how often do you, or how many times do you think you and I would need to meet?

Me: I'll probably, umm, especially nearer December, I'll probably meet you quite often and so you can get feedback…

The teacher: say once a week or once a fortnight?

Me: probably about once a week

The teacher: yeah that's fine.

Me: but it all depends on like, when I make a new section, I don't want to get too into it.

The teacher: No because you'll be wasting time

Me: especially if it's something not necessary

The teacher: so you need to get my opinion about what you've developed at certain points so you don't commit into completing it.

Me: Yeah, it will probably be more like a prototype thing, there won't be, they'll be more simple shapes just to see how it interacts and how it works.

The teacher: ok. Well I am really keen for you to produce something that we can use, that would be really nice. Because it would be tailored to us, and our particular needs, rather than trying to buy something off the shelf that's never ever going to fit exactly what you are going to want.

Me: yeah

The teacher: so, yeah! I'm keen for it to work for us and to be a tool for us to use in the, in, as of next summer, which would be nice. Yeah.

The teacher: anything else?

Me: No, I think that was all actually.

The teacher: alright then

Me: Thank you very much

The teacher: No Worries.

# Analysis of results:

## Observation:

My observation was the fieldwork, performed by the students. I thought this would be a more accurate representation of what things need to be looked at. This is because, during a lesson, the new information is fresh in the mind so it's remembered more easily. During the actual fieldwork, they have to remember what it was that they were taught and as a result, I can truly see what needs to be worked upon.

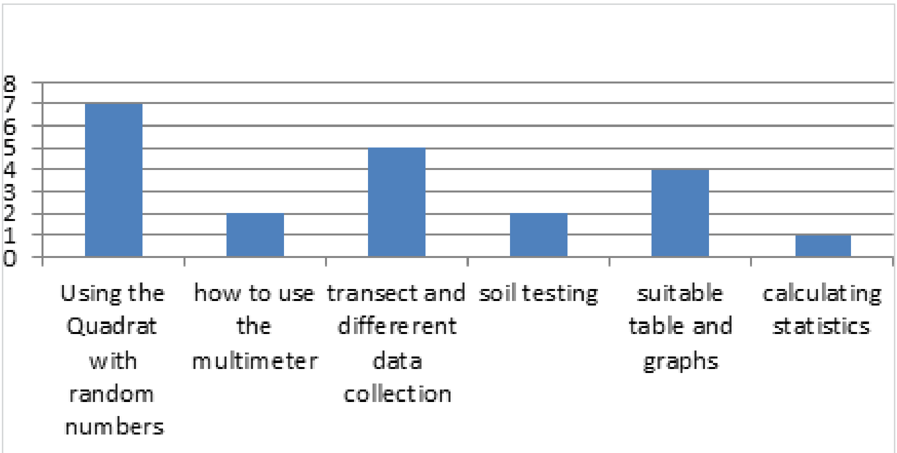
The key thing I noticed was the use of the multimetre. Many people used this tool in many different ways and, whilst this is ok if they standardise it, they did it differently for most readings. This creates unreliable data and is easy marks lost. I think I may use a program like Adobe Flash to create a simulation of the multimetre and good fieldwork techniques and methods of standardising its use.

Another thing I noticed was that, when measuring the actual number of the plant they were looking for (i.e. buttercups), students tended to just look for the flowers. This is not an accurate representation of the amount that there are as each plant can have many different flowers on it. It seems that it was some of the very obvious things that went amiss and, as I said earlier, are easy marks being needlessly lost. I think I'd like to use images and graphics to help with this and add some examples to be completed/ worked through.

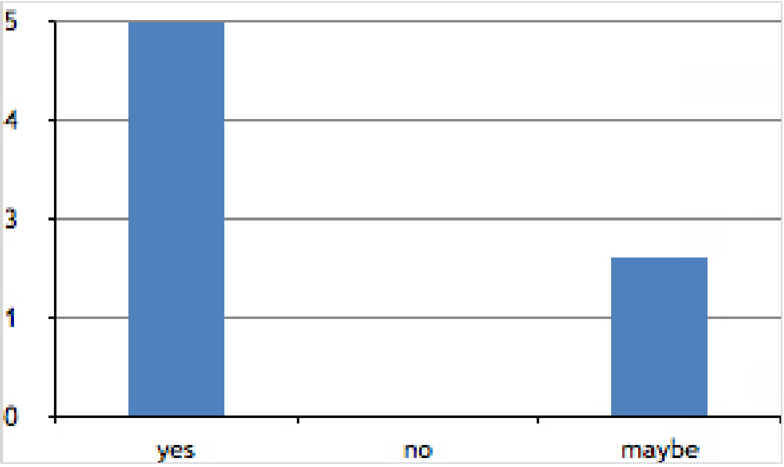
## Questionnaire:

I've processed all the results from my questionnaire in Excel to produce visual representations of the data. The graphs will be presented in the order of the questions so it is easy to see which graph belongs to which question.

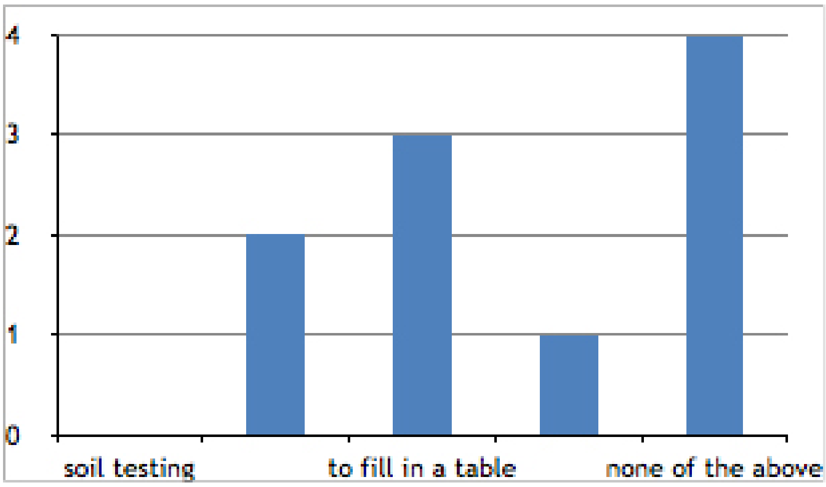
1. What aspects would you like to be emphasised the most?



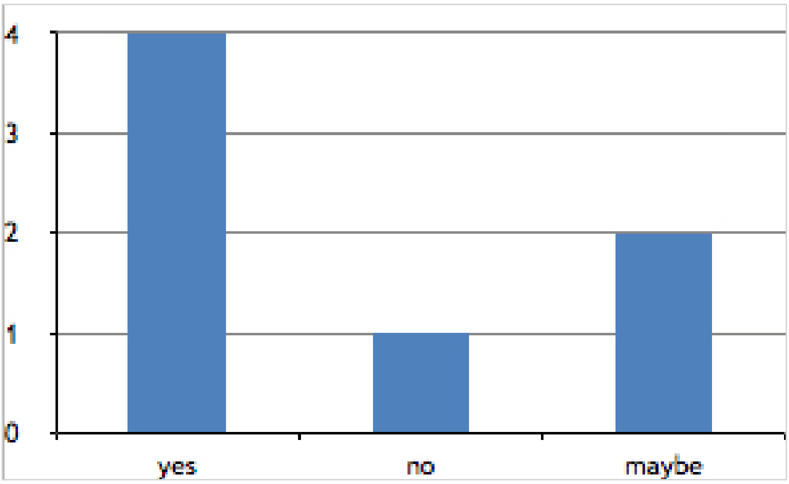
2. Would you like simulations?



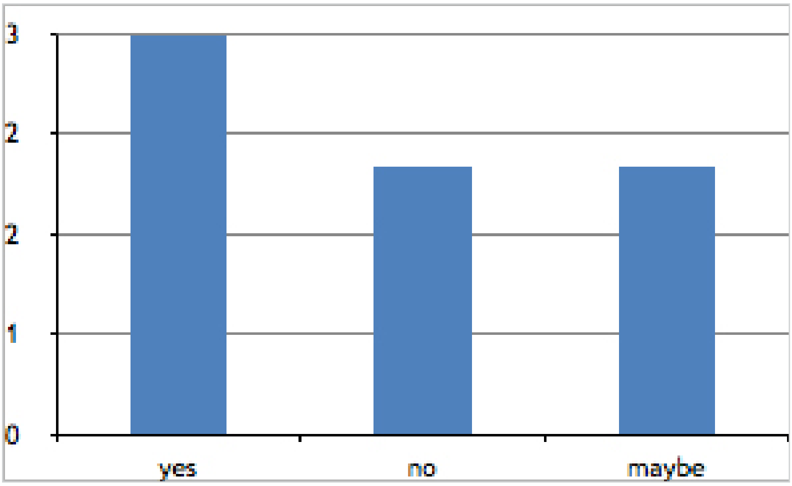

3. In which features would you like to see a random number generator?



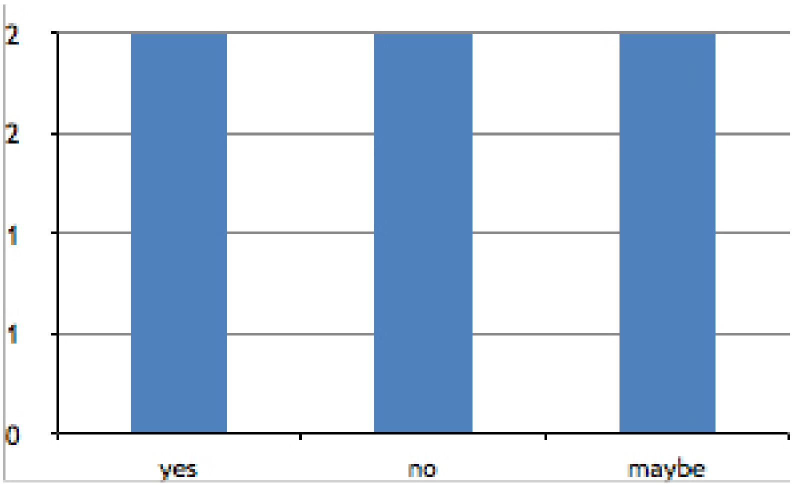

4. Would you like a quiz/test at the end of each section?



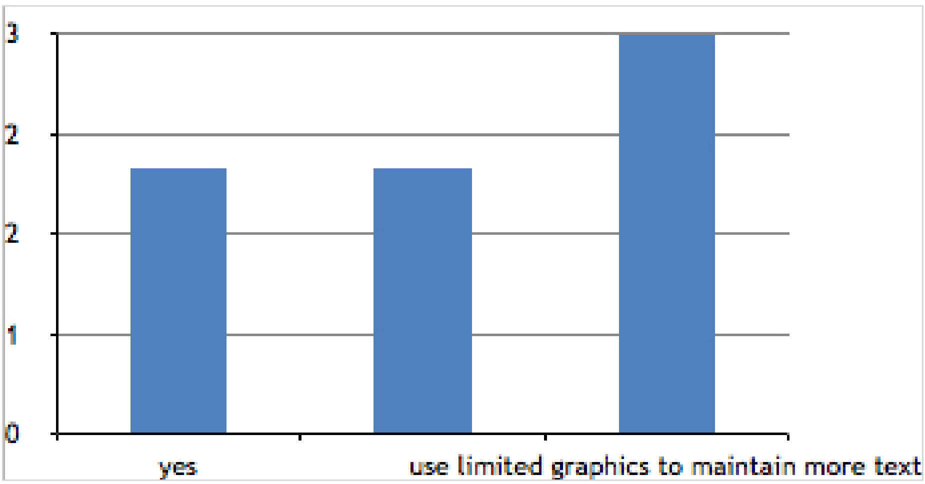
5. Would you like a database of sorts?



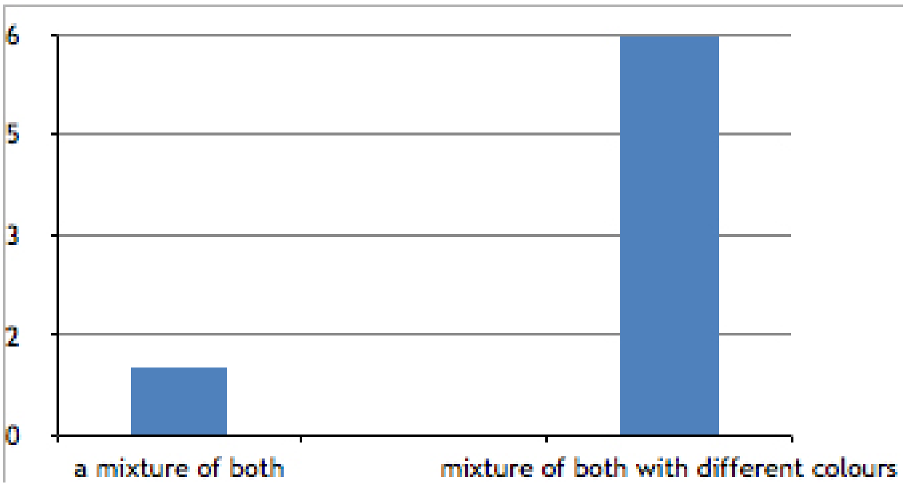

6. If yes to the above, would you like to have a student and teacher log in to view their progress?




7. Would you like to have lots of visual aids as well as text?



8. Should the font be stylised to highlight key areas/ terminology or kept the same throughout?




From my analysis of my questionnaire, I've got a good understanding of some features that are liked and what's wanted but I found that the questionnaire also yielded some mixed results. Especially to the question on having teacher/student log ins. Each selection got 2 votes.

I think the questionnaire has been particularly useful as I have information from most of the end users and have an idea of where to take the project from here. However, I still have to do the interview and I think this will clarify any misconceptions I have and will also help me get a better feel for what's generally wanted. A questionnaire is obviously fairly limited in its scope.

## Interview:

The interview was very important to me because I'm creating a new system so I don't have any other documentation to refer to. As a result, most of my information and understanding of my end users wants would come from the interview.

I decided to section my interview transcript to make it easy to talk about without having to go into too much detail about what part I'm referring to. I sectioned them off using a coloured line down the side.

The blue section was the start of my interview and this is where I wanted to get something very clear before I began understanding the system better and their wants, do they all teach the same way? What use would there be for a system that only a few teachers could use because of the way they teach the syllabus? Thankfully, The teacher assured me that all the teachers follow a fairly strict framework, based around the exams and terminology that students are required to know. This means that I can create a system, perhaps using some paperwork and hand outs that I've been given when I was taught the section, with certain confidence that it's usable to the department.

The red section was something that I hadn't planned to talk about and it was The teacher whom pointed it out. He suggested having a map of the location that the students do the fieldwork in and highlighting, pointing out the areas which they would tend to go. This could then be used to identify which study they would be doing and could almost be used as a menu. You could select one area and it would take you to the part of the syllabus that refers to the sampling you would do in said area. Also, The teacher agreed that he could help me point out the areas of interest which means it will be more accurate.

For the green section, I asked my second planned question which was, 'what's the hardest topic for students to grasp?' I found out that the statistical part of the course was particularly challenging because roughly more than half of the class have decided not to continue with the maths studies. The statistical section is fairly challenging unless you take maths as an A-Level already and, although I'll be sure to include all the correct terminology and model answers to the questions that may be set, I think I'll include a kind of 'Laymen's Terms' of statistics to help some of the students who struggle with it to get their head around it before they go onto the proper mathematical part of it. However, I would need to consult The teacher once more to confirm the implementation of such a feature because he did make it clear that he wanted it to be concise and following the specification. But from my experience in class, this is what tends to confuse people the most because they don't know, nor understand, the terminology and to then be thrust into the deep end of the pool, so to speak, can be daunting.

In the grey and purple section, I wanted to speak a little bit with regards to the questionnaire. I had handed the questionnaires to the teacher to give to the rest of department already and had just been given them back so I hadn't had time to look at them yet. However, I wanted a more in depth interaction to do with some of the questions from
the questionnaire that I think should be expanded further upon.

In the grey section, I asked what aspect of the system he wanted to be emphasised most of all that would aid the department more. The questionnaire also produced this answer but he said that he wanted something that would focus on the placing of quadrats and estimating the percentage cover/ number of a species within that quadrat. He felt like this is the topic that is covered the least because it's hard to get lots of practise doing it, if any beforehand at all. From this, he went on to speak about linearity and how he thought it would be bad for the program to be linear. I agreed with him and said I was going to implement a menu of sorts to help navigate the program with ease and also to allow teachers to decide which topic they wish to begin with, without having to go through each process laboriously. As I said earlier, I think having a detailed map as the starting menu may be a fun idea because it means you can specifically choose what sampling study you wish to do and from that menu, a submenu could appear that would allow you to focus on what aspect of that particular study you're having difficulty with.

In the purple section, we were discussing the implementation of a database of sorts that could be used to keep track of a student's progress and make notes on it. I also suggested the use of a student login so that they could see how they've been progressing and see what things they need to work on to help achieve the best grade they can get.

The yellow section was quite interesting and informative actually. I was asking about hardware constraints such as the laptops they use. I was reassured that the whole department uses window based laptops which is good because it means that there won't be any complications insofar that I'd have to create the system for two completely different operating systems. This wasn't, however, the intriguing part that I found out. A method of accessing your college profile from home is through software called VMWare. The teacher told me of how this prevented his antivirus software from functioning but only on his window's 7 PC and not on his Windows XP PC. This is interesting because I may need to take these factors into consideration when creating the system, especially if I want the software to be usable at home.

The black section was the last section and this is where I asked of any questions the teacher might have for me. His main concern was when the system could be finished and installed. Unfortunately, I wasn't definitively sure of this but I believed it to be around December/ Christmas time. This would mean it could be used during the summer when it'll be taught. He also enquired about follow up meetings that I might have. I said that I would probably be meeting him fairly regularly nearer the deadline and also that I would come to him when I had begun a new section and get him to give me feedback. He was willing to help which is very good for me because it means that I can count on him to give feedback and is willing to use the software and, consequently, make it the best that it could be.

# User Requirements

## Outline of problem:

My end user would like the software to focus, mainly, on the collection of data and how to standardise the techniques to improve reliability and accuracy. He'd also like the statistical elements of the system to be included as it's typically the hardest topic for students to grasp. To do this, I will include simulations of the gathering of data and how to standardise the techniques to get the most reliable results. In this simulation, I'll have a little animation to show the user how to use the program and how to get the data in the most effective means possible. Then the user can place the quadrat using the mouse and estimate a value of the number of 'X.' The user can also expect step by step instructions and diagrams to help with the statistical element to the course. My end user would also like to allow the user to work through examples, set by the teacher or using their own results. Lastly, a clear navigation system is essential. He doesn't want the system to be linear and wants the users of the system to have freedom with what they want to do, and having a linear/ unclear navigational system is confusing and can make the experience of using the system as a learning tool distasteful.

## Key things to include:

**Point 1:** To add a new sample

**Explanation:** This will allow staff and students to add data into the system. Based on the sampling study they have chosen to do.

**Evidence:** This is indirectly implied from the point of non-linearity (Grey section from the interview). Without this feature, you would always need to complete the simulation to get results. The teacher has made it clear that this isn't what he, nor the other biology teachers, wants.

**Point 2:** To calculate the statistical results from the data

**Explanation:** Once data has been received by the user (by means of the simulation or self-inputted with their own results), the program must then work out the stats part. Each step will be noted so that the user can follow what the program has done (in case an arithmetic mistake was made by the user somewhere).

**Evidence:** (Green section from interview). The statistical element to the specification is difficult for students to grasp so it needs to be completed effectively and to the standards, expected of them in their exam, required.

**Point 3:** To save/ edit and delete a sample for later use

**Explanation:** This is for the teachers. They need to be able to save a sample so that, if they want to start the lesson off with the program, they can have a sample ready for the class. This is to save time for the teachers. Including this feature does also mean that I'll need to allow the teachers to retrieve these saved samples and edit it them if they need to. And also, to delete them if they no longer are required.

**Evidence:** the very beginning of the purple section of the interview.

**Point 4:** "Tutorials" on the statistical part to help teach students the maths of the specification.

**Explanation:** A lot of the students have difficulty with the stats part because not all of them have kept their maths in check. These pages need to be very concise and follow the wording of the specification so that they can reproduce the language required of them in the exam. I may also include a "Laymen's Terms" type page to help those who are really struggling to get their head around the topic, before moving on to the proper definitions and terminology.

**Evidence:** The green section of the interview.

**Point 5:** A simulation to place Quadrats and take estimations from them

**Explanation:** This is will be a fairly large part of the program and needs to be able to record all the data that the user inputs, to transfer to the stats section if that is what they wish. I had originally thought of using a program called Adobe Flash for the making of the simulation as I'm fairly adept with its use. However, not everyone may have the required software to run the simulation, thus making the point of the program pointless to say the least. Therefore, I think it's best to do it inside of Visual Basic as well, just to be on the safe side.

**Evidence:** The Grey section of the questionnaire.

**Point 6:** Non-Linearity within the program

**Explanation:** The teacher and I both agree that the system shouldn't be linear. As in, you have to go through it a particular way every time. This would be extremely boring for someone who may just want to go straight to the statistical part of the system, without having to go through the simulation to get results. I had the idea of using the overhead map of the site (Highwoods) as the main menu screen. The sites which you'd go to on the field work would be highlighted and you can select one and only focus on the sampling methods/ stats associated with that sampling study. There will obviously be a menu to return to this menu at any given time.

**Evidence:** The grey section of the interview

Unfortunately, due to time restraints, I do not think it possible to achieve all the things I had initially envisaged. For example, the database element where students and teachers would have their own log in and your data could be tracked. This is a large task and perhaps larger than the rest of the program. Whilst I would like to have included this, it just isn't feasible in the time allotted and I would much rather focus my efforts on the parts the biology team need.

# Hardware and Software Requirements

## Software:

The minimum operating system required for this system will be Windows XP. This is because the college system runs Windows XP and most current machines can operate at least Windows XP. However, newer machines can run higher operating systems such as Windows Vista or Windows 7/8. However, these more advanced systems will easily be able to run a program that is targeted towards Windows XP.

## Hardware:

These are the minimum hardware requirements that a computer needs, to be able to run Windows XP:

- Pentium 233-megahertz (MHz) processor or faster.
- At least 64 megabytes (MB) of RAM
- At least 1.5 gigabytes (GB) of available space on the hard disk
- CD-ROM or DVD-ROM drive
- Keyboard and a Mouse or some other compatible pointing device
- Video adapter and monitor with Super VGA (800 x 600)
- Sound card
- Speakers or headphones

As I said previously, most new computers run much more advanced operating systems that require more advanced hardware. Because of this, these requirements are not going to be difficult for most people to meet.

## Backing Up:

The college system is backed up at least every day by the IT technician. This means that it won't be necessary for the biology team to have to go and buy a memory stick or a portable external HDD to do this. This is because I'm getting the program put onto the college system, rather than having it as something you must download separately. And students at home can log in via the virtual network program provided by the college. This means that space isn't being used up on their computer, allowing it to run more effectively.

# Solution to Problem:

## Spread Sheet Solution:

With spread sheet type software, I'd display the results in a tubular form, for the students to view the data and the results of the statistical methods involved. This would follow a rather logical format and would be simple to comprehend. However, it would be difficult for to save a sample and to then retrieve it. Also, it will mean I can't show the stats independently to the data being used and will only be able to show an answer. This won't be helpful at all.

## Tailor-Made Solution:

Tailor-Made Software will be software that's made specifically for the end user. I would be able to create software that can do everything you need and not a little bit more. The benefits to this is that you won't get a lot of unnecessary features to the program that you won't ever use, but still use up space and will have been paid for. I'd "tailor" to your needs. This means I'd create something that simulates the fieldwork and the stats, with help and advice for both, using the terminology you deem fit. The one negative is that it will take longer to create.

## Conclusion:

In my opinion, Tailor Made software is far better. Not only because of the freedom you can achieve, but also because of the specificity of the system you're after. The program you require is very specific and there isn't software currently out there that can provide what you need. Also, spread sheet software is rather limited for what you want to achieve and much more work will need to go in to make it a beneficial system to the department.

# Email of Confirmation:

I just wanted to ask if you are free to look over some documents and give me your feedback on them. I don't want to continue with this new program if you disagree with some of the things I've proposed.

I've included all the processes that I think the program needs to do for the system to do what it is you require. It's only a general overview of all the features so forgive me if there is something you think is missing from the list. I'll add to the documentation and vice versa, if there's something you think isn't necessary or is perhaps wrong, please tell me so I can make amendments.

Secondly, I've included a document, deciding on the solution to your program. I looked at two ways in which it could be completed, Spread sheet format and Tailor-Made. I've suggested the Tailor-Made format because it will achieve the desired effect, whereas a Spread sheet format would make it rather difficult to help with the statistical portion of the program.

If you could just confirm these documents are correct, I can begin the actual designing of the software. I'll be meeting you more regularly (if you can) about screen designs and the look of the program.

Thanks,

[redacted] – sorry for the delay – I have just been too busy to look at this recently – the document you have sent me looks fine as a summary of what we discussed at the interview.

If you would like to meet again to discuss further I am mainly free during block A this year.

Thanks

# Design

## User Requirements:

Because of time constraints, I have had to remove some of the features that I had originally wanted to implement. These features include: the file handling features (saving a file, changing and deleting) and the separate log in for students and teachers. I decided upon dropping these features because they weren't crucial (I thought) in the new program. I would prefer to hand over something that can actually be used and then, at a later date, implement the full program with the all the features I wanted from the beginning.

1. Simulation 1
2. Simulation 2
3. Simulation 3
4. Taking the readings from the simulation into the stats test
5. Table of results
6. $Chi^2$ stats test
7. Standard Error / Confidence Limits stats test
8. Spearman's Rank Correlation Coefficient
9. Non-linearity throughout the program.

1. Simulation 1 involves counting the exact number of plants in a small sectioned off area. For example: in a 10mx10m grid, the grid will be sectioned into smaller, 1mx1m squares. This would result in 100 different squares within this area. By method of random number generators, the user will take the coordinates (given to them from these random numbers) and measure the number of plants within this square. There will be a button to generate the new numbers, as well as a button to record the result the user has entered into the text box. There shall also be a back button and a help button on the page.
2. Simulation 2 follows the exact same principles as simulation 1 but you measure the percentage cover of the plant instead. It even has the same layout as the one for simulation 1
3. Simulation 3 involves measuring the change of percentage cover of plants from a set point in a line. You also measure another variable where you attempt to find out if there is a significant association between the variables. In this case, the percentage cover of plants and the light intensity. I will have 3 different lanes, each being made up of 20 boxes. As the method of sampling for this simulation is an interrupted belt transect, the user will take recordings at every other box. This will result in 10 readings for percentage cover and 10 for the light intensity. There shall be a back button, a help button and two text boxes to allow the user to input their readings and see if they're correct.
4. On each of the simulations, there will be a button that will take the user to the stats tests that is assigned to that simulation. For simulation 1, the user will be taken to the $Chi^2$ test. For simulation 2, the user will be taken to Standard Error/ Confidence Limits and for simulation 3; the user will be taken to Spearman's Rank Correlation Coefficient test. What I will do I create a global array that will store the recordings as the user submits them. This array will exist in a module outside of the forms and this then means that it can be called upon from anywhere within the program.
5. On each stats test, I will also include the results from the simulation. This is so that the user can see the data they collected themselves and can use it to help them with the stats test. I will use the global array to change the captions of specific labels to the readings. If no readings have been made, I will allow the user the option to either generate random results to populate the table or allow them to input their own data. A pop up message will appear in this case and will provide the user with the options; "Generate Random Results," "Input own results," and "Complete Simulation first."

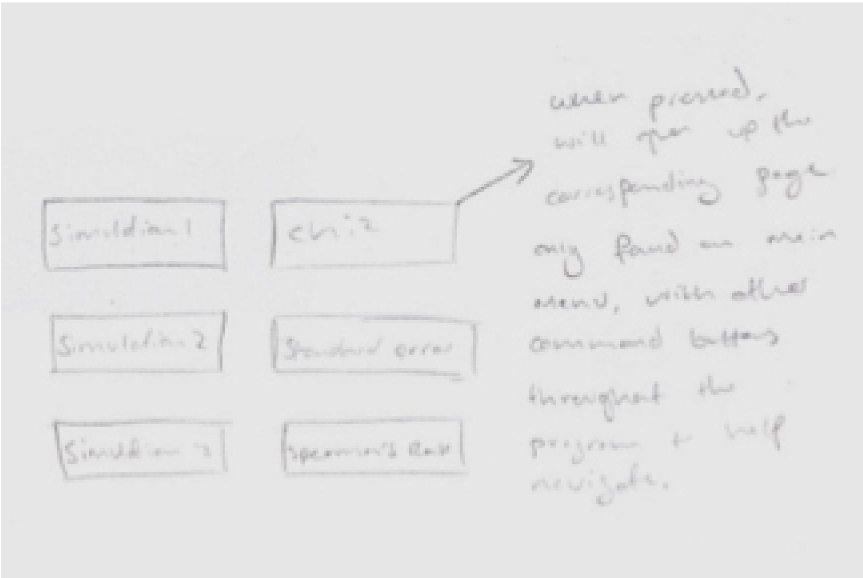
6. The Chi$^2$ test will have the table of results at the top of the page. This will be populated with the readings that the user has either entered themselves or from the simulation or randomly generated. The total number of plants will also be worked out and given to the user. There will be another table of sorts that will guide the user through the process of calculating Chi$^2$. There will be a check button at the end and if everything is correct that the user has inputted, the text boxes will turn green (red if wrong) and a hidden text box will appear. This text box will contain the evaluation from the stats test and what the figures show for the user. There shall also be a back button, a help button and a check button on the screen.
7. Standard Error/ Confidence Limits follow the same layout as the Chi$^2$ page but with slightly different headers and text. Like before, there will be a results table at the top of the page and the mean percentage cover calculated for you. The stats table will be similar in the sense that there will be a table that will have all the steps of the stats test divided up into sections to help make it easier to understand.
8. Spearman's Rank is slightly different in that it requires the user to rank two data sets in ascending order. For example, if 5 was the lowest value the user had recorded, they would rank it 1 and if the highest was 29, it would be ranked 10. The user does this twice for both data sets. They then use the differences in these ranks to calculate the association between the two variables they were measuring (in this case, percentage cover of plants and light intensity). Like it was previously, the stages involved will be broken down so that the user can follow the calculation with greater ease.
9. The Non-Linearity of the program shall be achieved with the use of bottoms, returning the user back to the main menu at any point in the system. I'll also include a "Continue" and "Previous" button to the page, so that the user can go back and forth, at his own pace. The main menu will consist of a map (the site in which the field work is conducted) with the areas of sampling highlighted. These can be selected and further option will then be presented to the user including: going to the simulation, entering your own data into a table, going directly to the stats section and then a tutorial page for that section.

# Initial Designs

## Methods of navigation:

### Command Buttons:

This type of navigation consists of multiple buttons that guide the user through the program. Each button, when pressed, will open a new window for the user to navigate through.



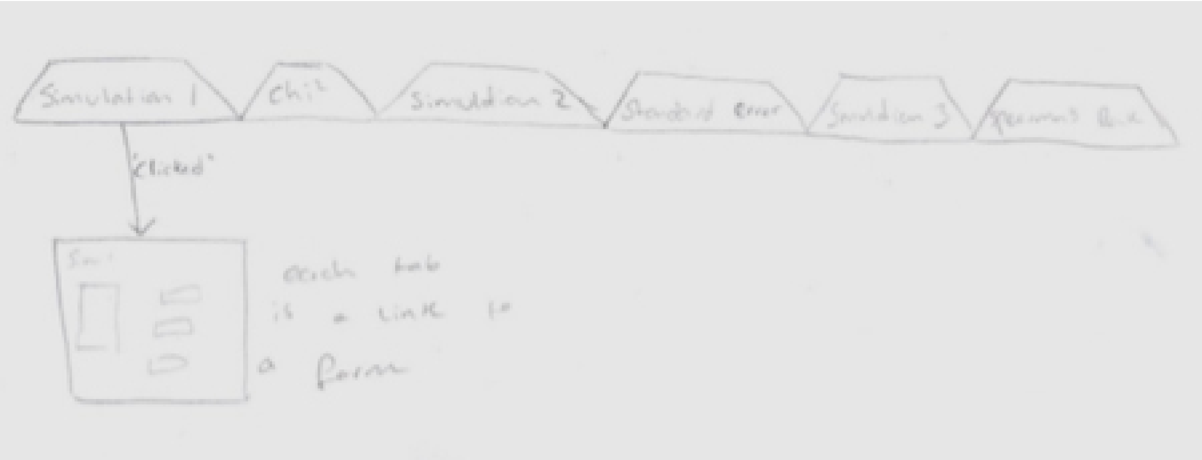

***Advantages:***

- They're very simple to use and the button text can help the user to pick the one they want
- It's rather an intuitive system insofar as it's very rigid and straightforward but can achieve many results.
- Perhaps above all else, this system requires very little training to be used.

***Disadvantages:***

- This system can leave the screen rather cluttered with so many windows open at once.
- Could slow the computer down if too many windows are open at once, perhaps making the system unpleasant to use.

**Tab Navigation:**
This type of interface involves tabs being set out at the top in a line, much like you'd see at the top of a website. When a tab is pressed, the corresponding window will open.




***Advantages:***

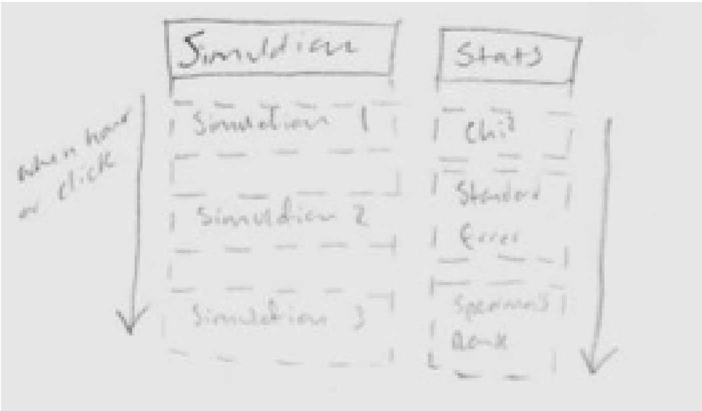
- Everything is in one place so therefore improves the ease of use
- Very simple and easy to use
- Little to no training required to use them

***Disadvantages:***

- May reduce the speed and efficiency of the computer if it has to deal with many windows open at the same time.
- Although everything is all together, you'd have to go through each one to find what you're after; harder to group them in an appropriate order.

### Drop Down Menu Navigation:

This is the typical form of navigation used by most programs. It involves an almost tab type navigation but when pressed, a menu will drop down below the tab with buttons that can be pressed.




***Advantages:***

- Many other programs follow a similar style of navigation so the chances are, no training would be required because they should be relatively familiar to the user.
- Easy to use

***Disadvantages:***

- Finding what you're after can be difficult as the user won't know where each option will be until they've used it a few times.

## Recommendation:

I would recommend the command button interface because they can be grouped together in relation to the category they represent. For example, I can group the Sampling Study 1 with all the constituent parts (simulation, statistics and the file manipulation) in one place and the same with the other Sampling Studies. This would massively reduce confusion and make it very easy to follow and understand.

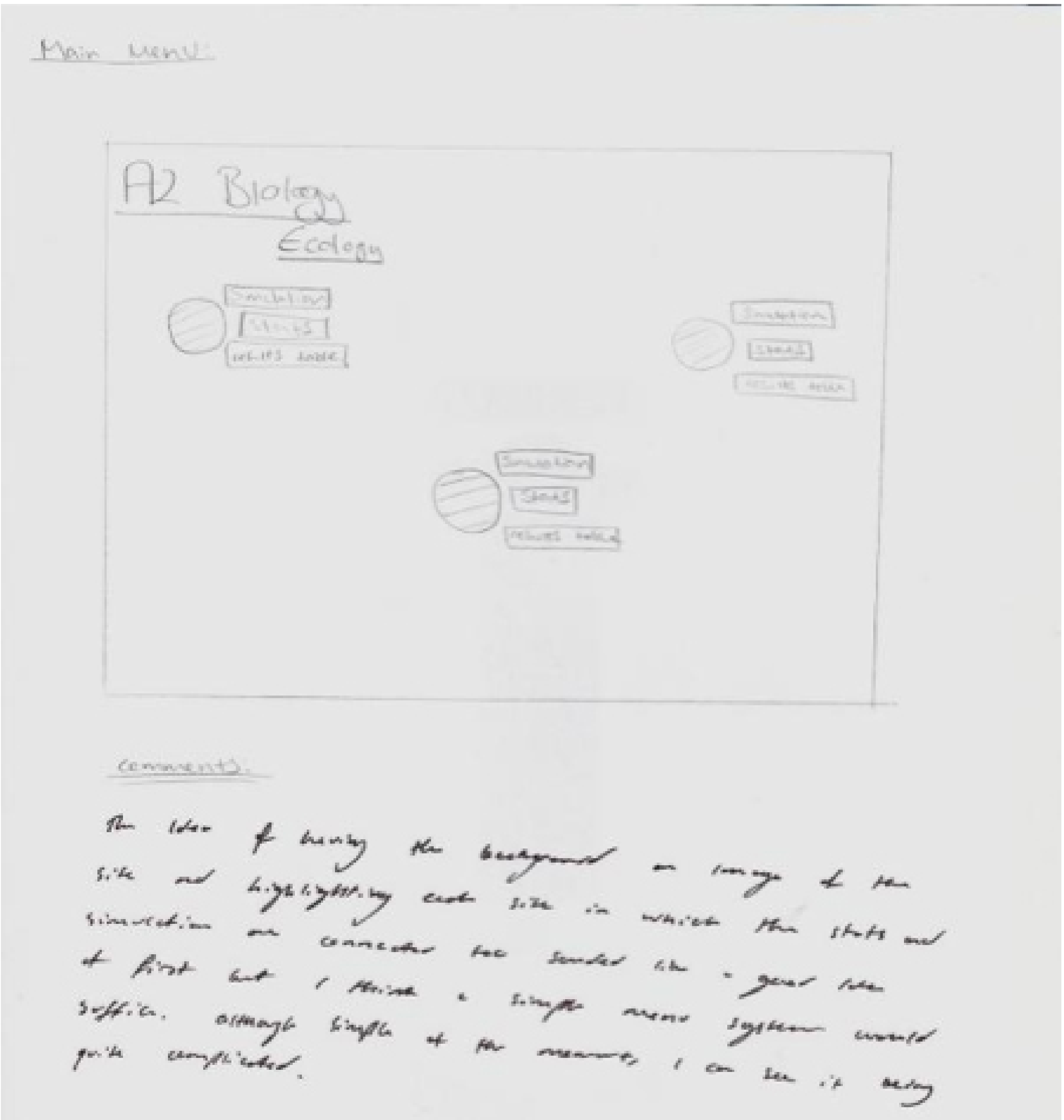
## Email of Confirmation:

[redacted] – I have looked at your navigation suggestions and feel confident in going with your suggestion of command buttons.

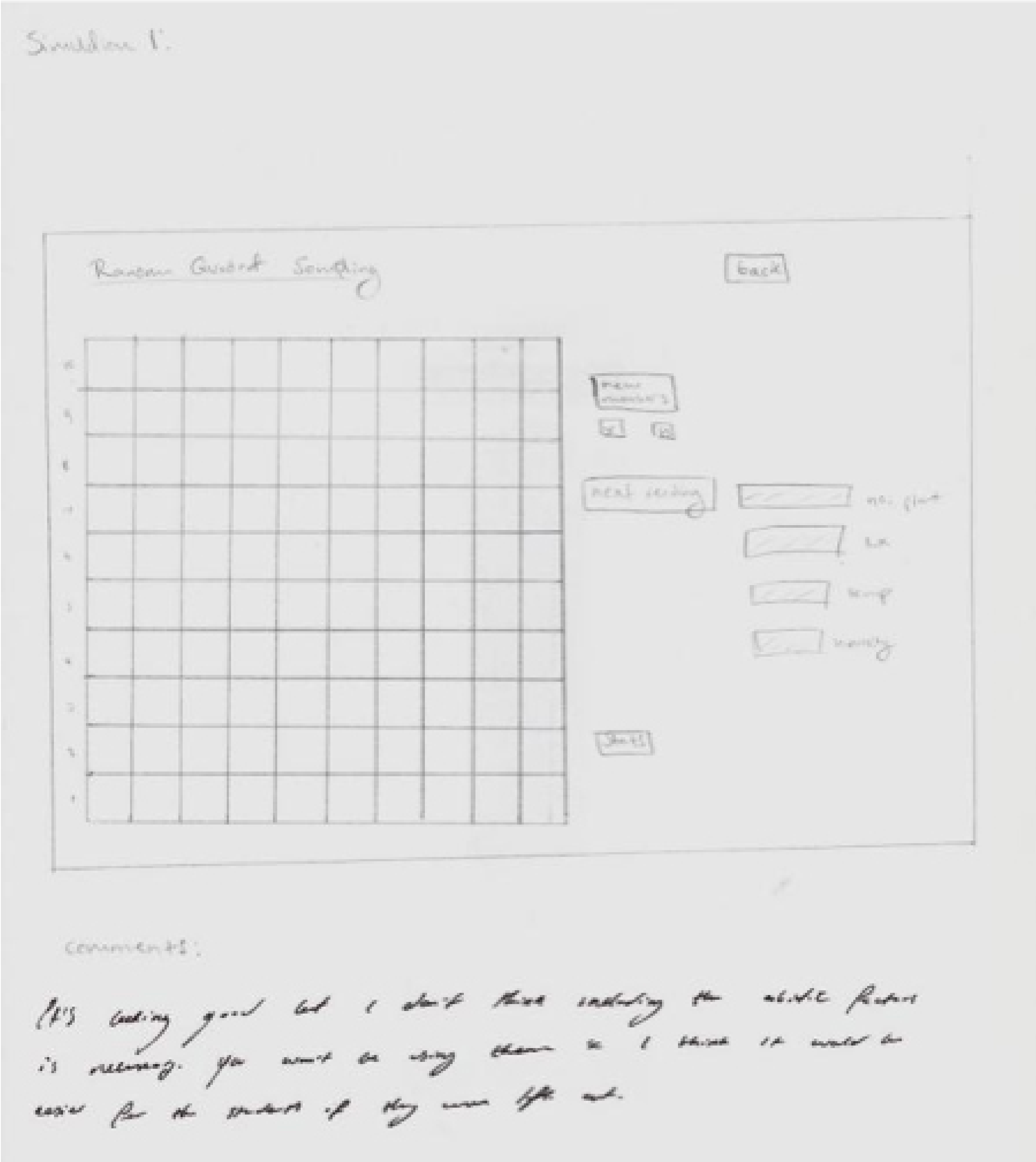
# Initial Screen Designs

Here I've designed some initial screen designs, that I have envisaged the system as. However, my end user may have had a different image and thus, he needs to see them and comment on anything he'd like changed/ added or removed. I'll leave enough space on each page for him to comment on so that everything is together and not separated.

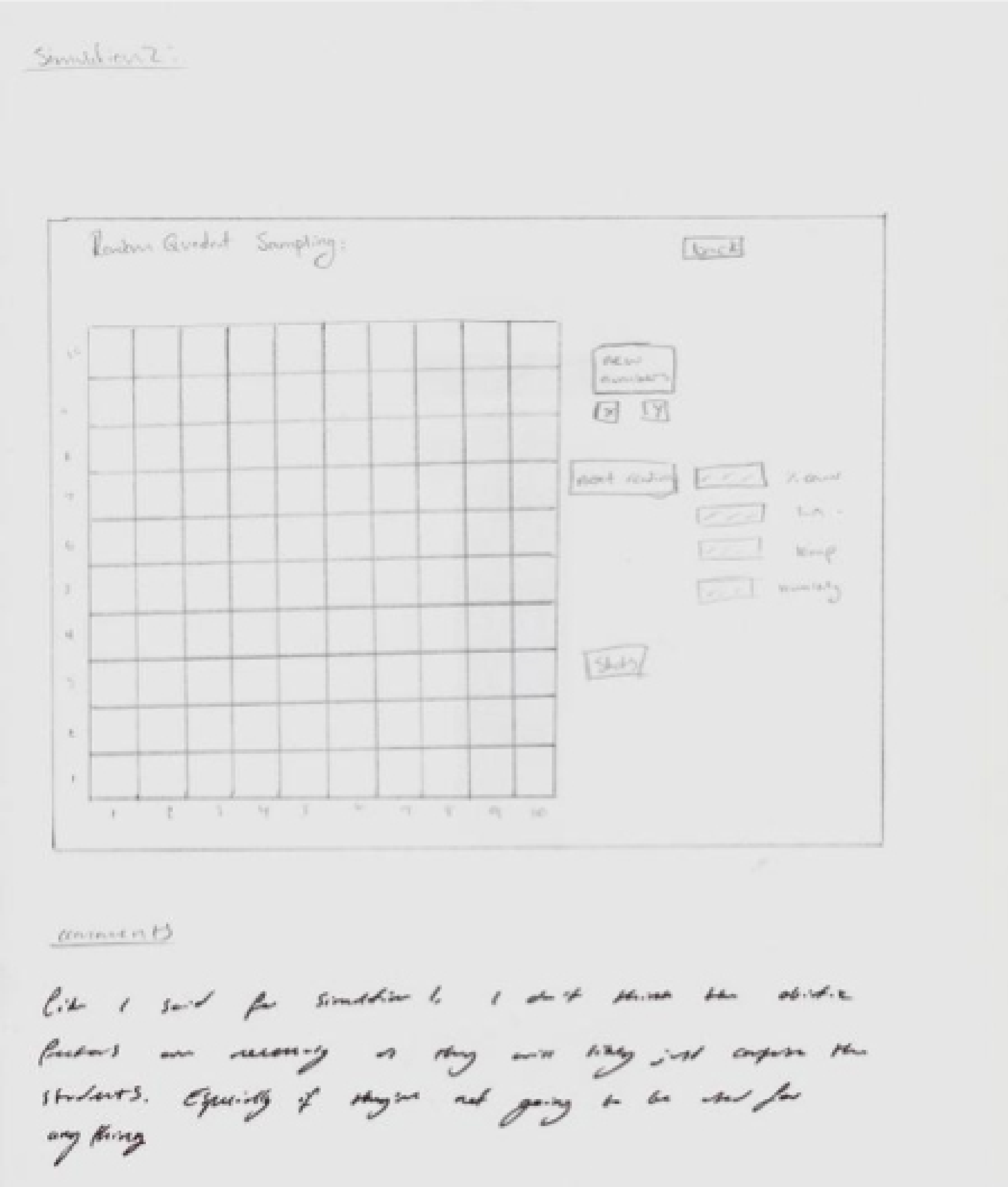
## Home/ Main Menu Screen:



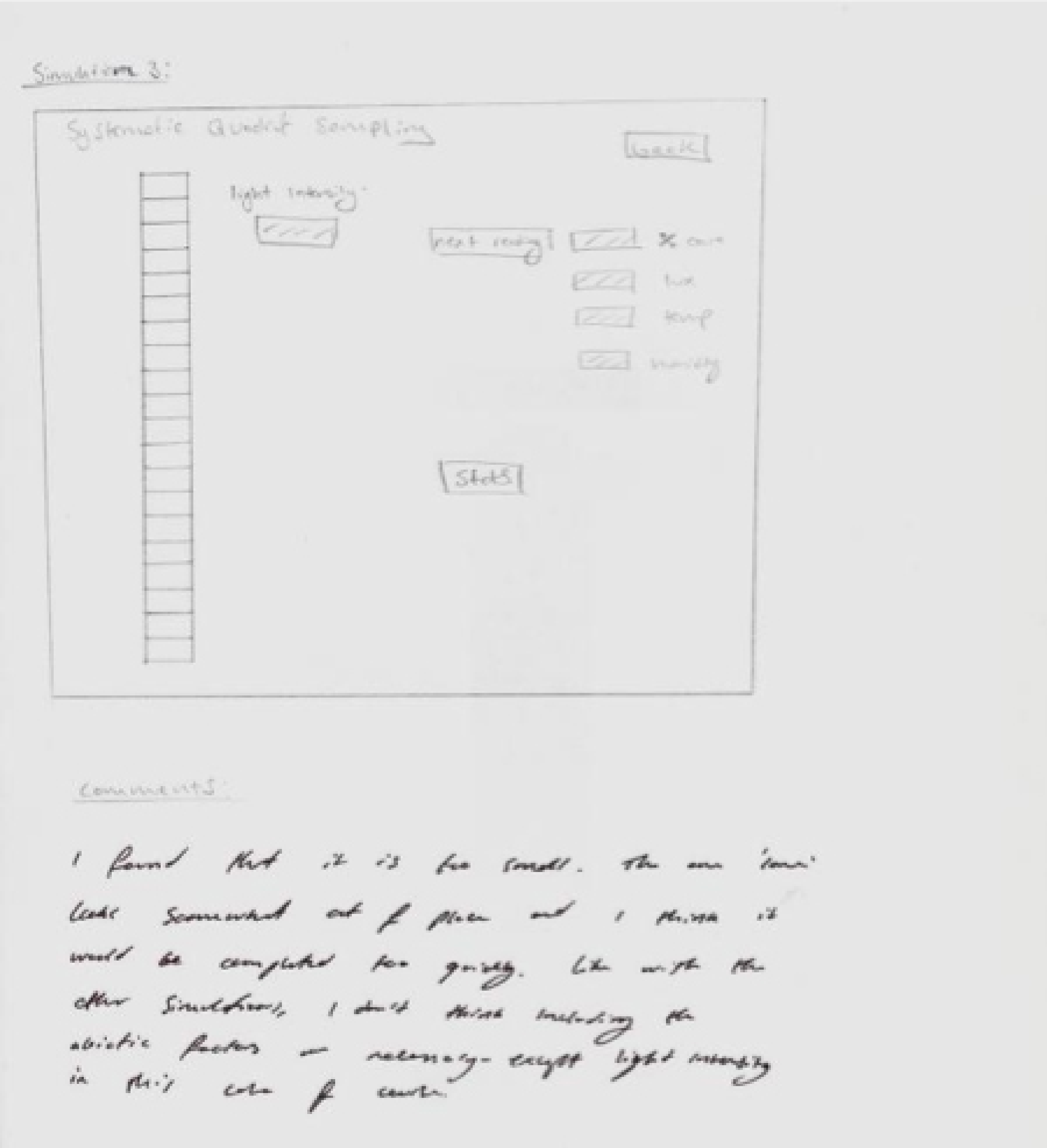
## Simulation 1:



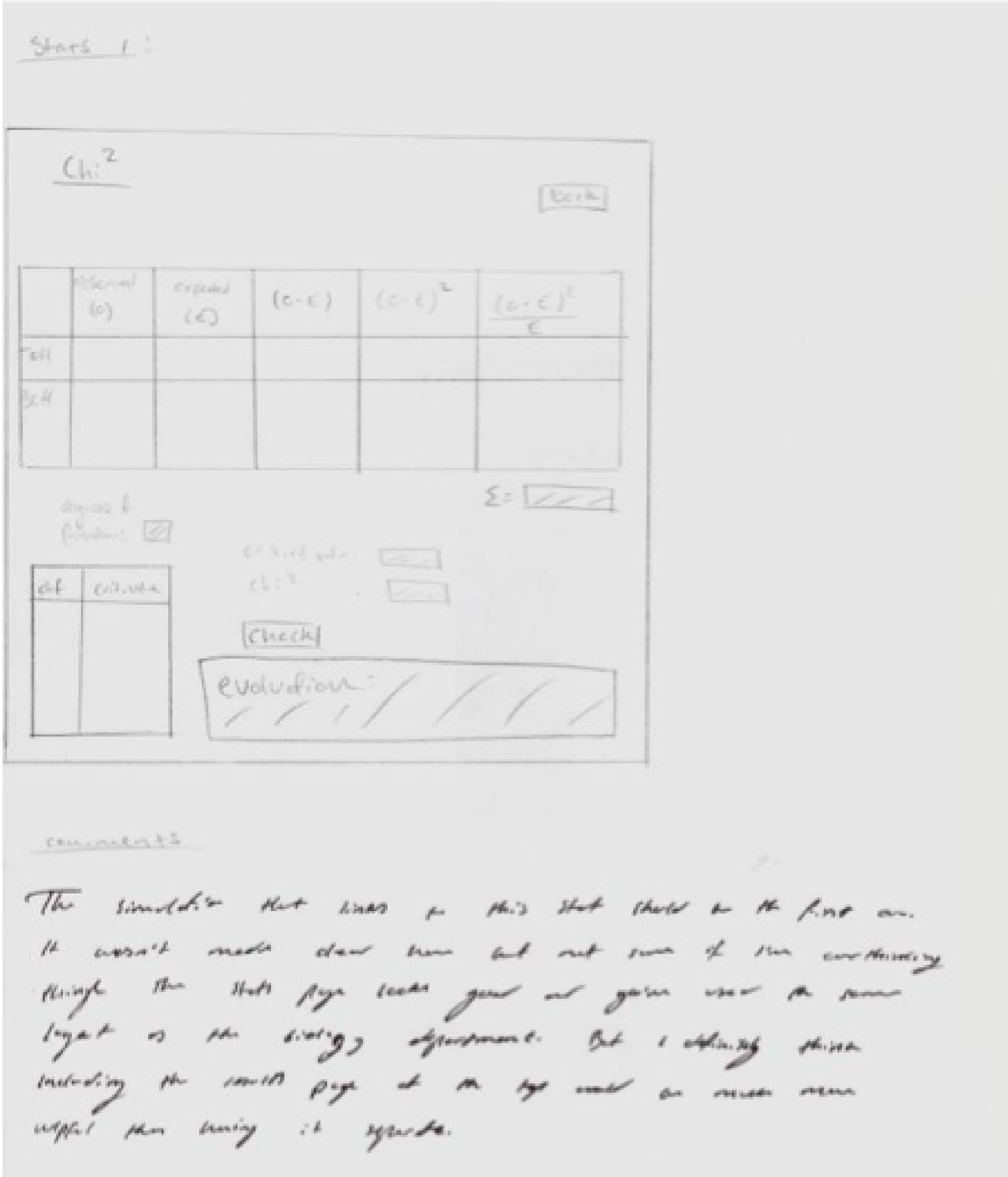
## Simulation 2:



## Simulation 3:

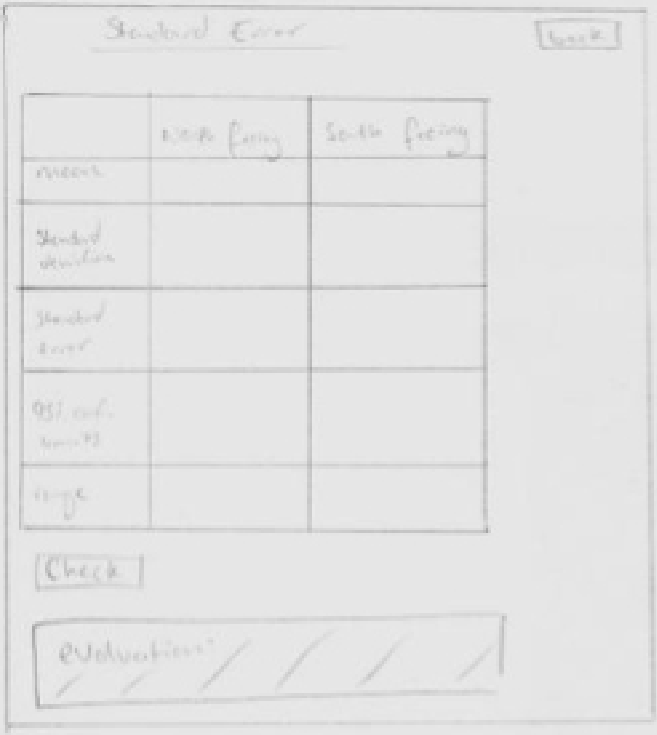


## Chi²:



## Standard Error:

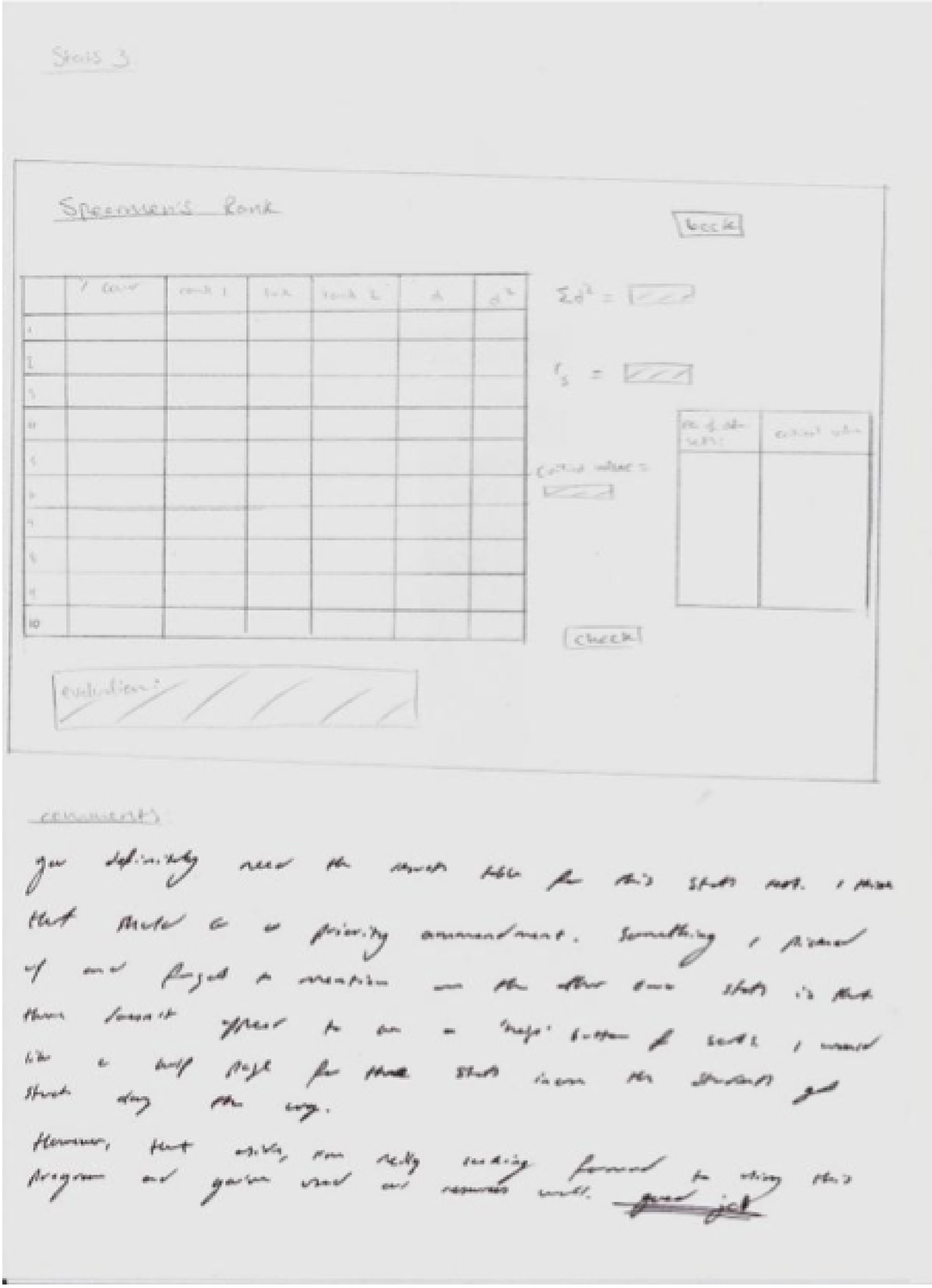
Stats 2




comments:

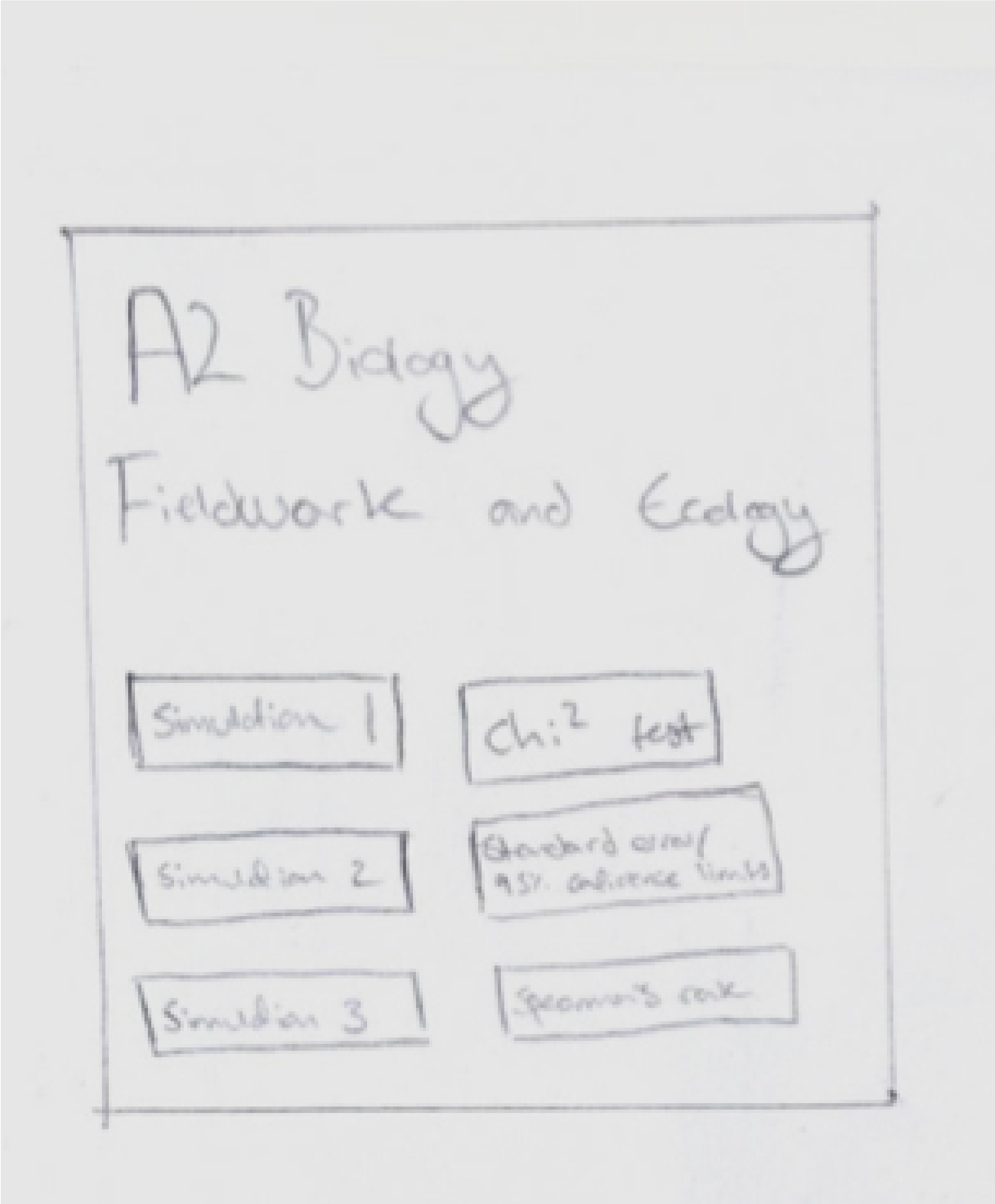
As mentioned earlier, I think the results table would be necessary for the stats. Having them on the same page would make things much easier. Otherwise, looks good and matches how we wanted it to look.

## Spearman's Rank:

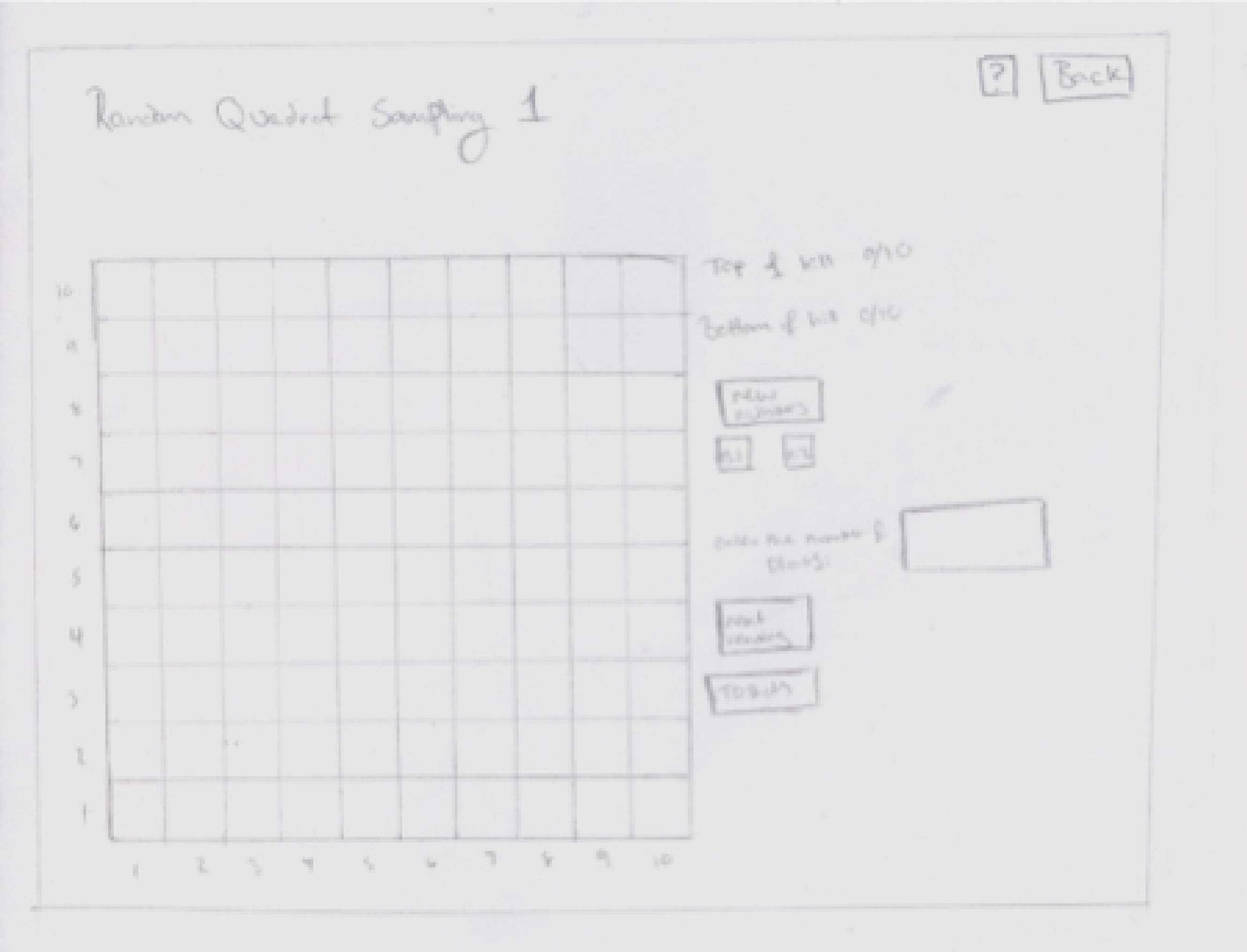


# Final Screen Designs:

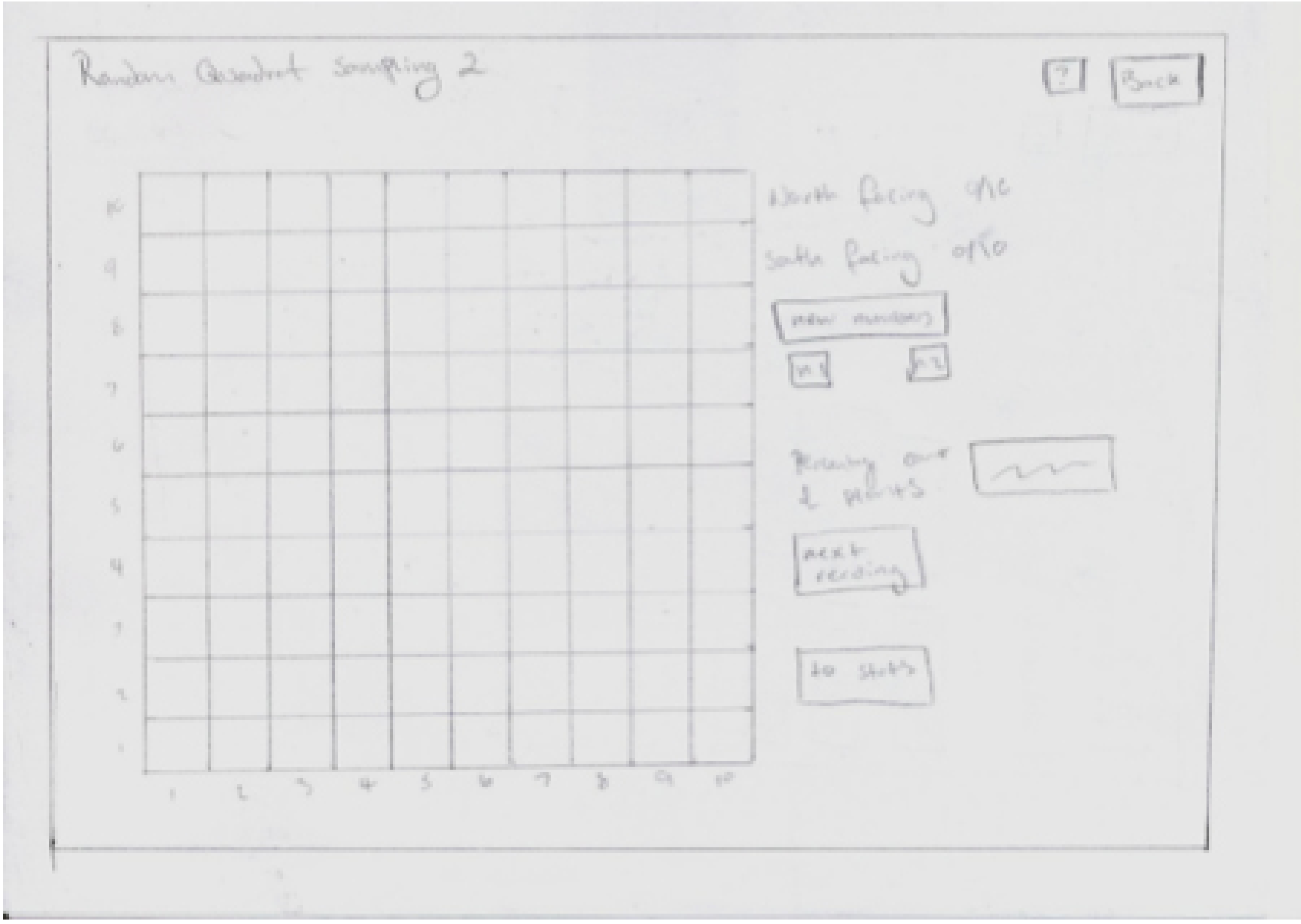
## Main Menu:



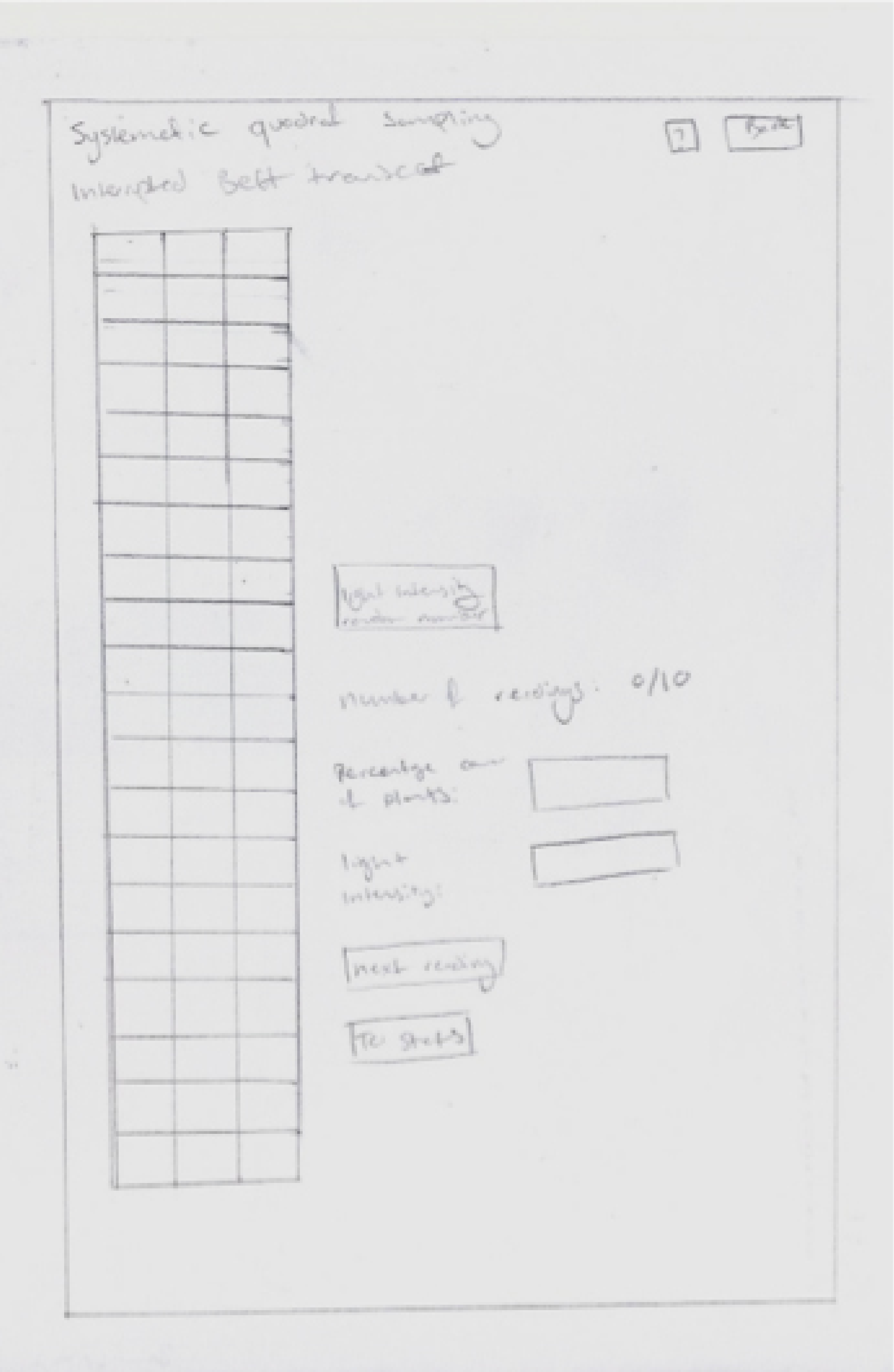
## Simulation 1:



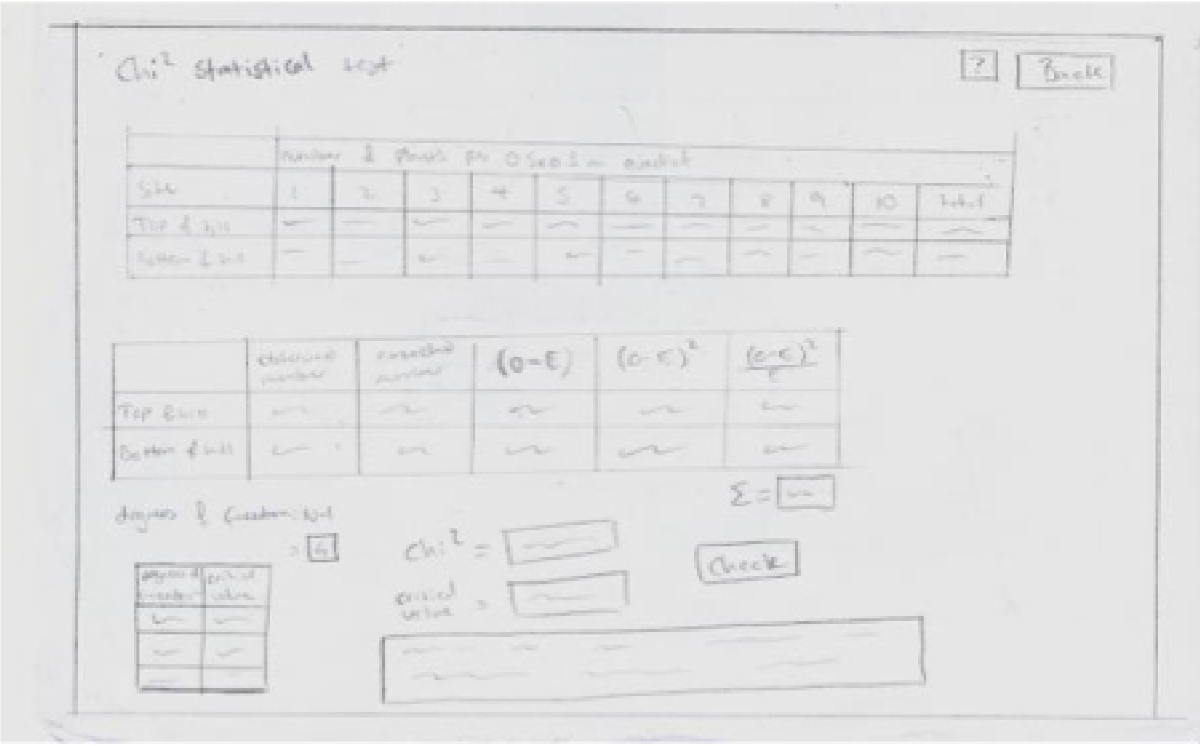
## Simulation 2:



## Simulation 3:

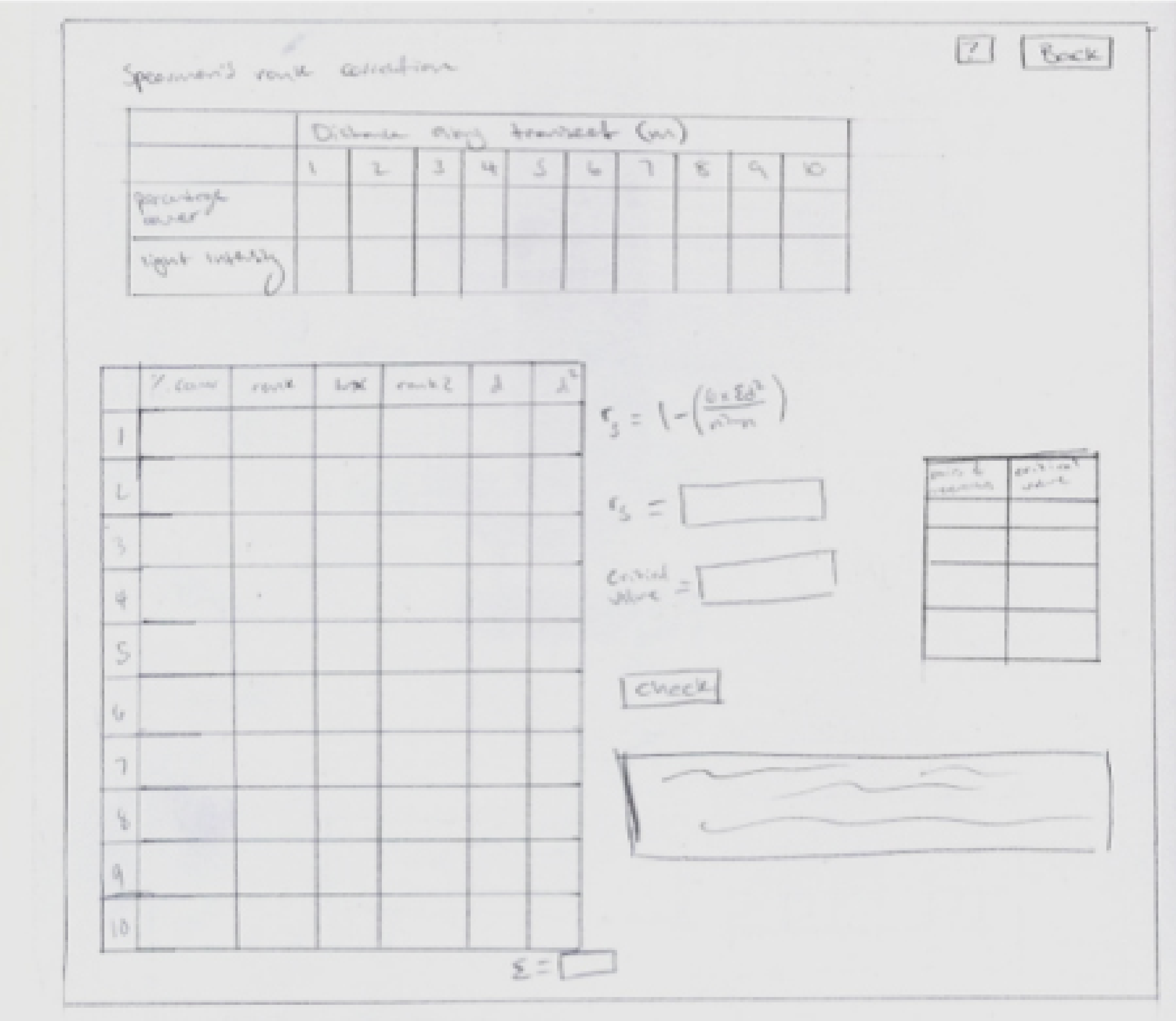


## Chi²:




## Standard Error:

## Spearman's Rank:



# File Organisation:

## Data Structure:

### Random Occurrences (Simulation)

Will be an Array (0-99) of the data type integer.

**Simulation 1:** Random no: in range 0-40

**Simulation 2:** Range of numbers in steps of 5 from 0-95. I.e. no :(0-19)*5
This will be to create a picture box of so many plants in it and the number corresponds to the number that's displayed on the text box. It's a 10x10 grid and each square will be given a number and this number will retrieve a picture box.

**Simulation 3:** random no: in range 0-95 for 20 boxes (with 3 lanes)
Simulation 3 requires data to be taken in a line. I will include 3 lines of boxes and the user will take readings from only one of these lines. The user is record every other box so I will need 20 images boxes per lane (60 boxes total)

**Random Number:** integer in range 0-99 (simulation 1/2). integer in range 0-60 (simulation 3). This variable will be used to store the random numbers that are applied to each of the random image boxes. This is so that the number can be taken and used whenever the image boxes need to be manipulated.

**Box Check:** Boolean in range 0-99 (simulation 1/2). Boolean in range 0-60 (simulation 3). This variable will be applied to every image box. This variable will check if the box that has been selected has been selected previously. If it has, the value for that particular image box will be set to true and can no longer be manipulated by the user or the program.

### Students Estimations:

Array (1-20) of integer data type

- The first 10 readings will be for the first variable (top of hill, north facing, percentage of plants etc)
- The last 10 readings (11-20) will be for the second variable.

**Simulation 1:** integer divisible by 5 in range of 0-95

**Simulation 2:** integer in range of 0-40

**Simulation 3:** integer in range 0-95 (first 10) and integer in range 9000-15000 (last 10)

### Stats Tests:

For Spearman's Rank:

**Percentage of Plants:** integer in range 0-10. This variable will take the first 10 values from the global value that stores all results from the simulation. This is so that the results can later be sorted in ascending order.

**Light intensity:** integer in range 0-10. This variable will take the last 10 values from the global value that stores all results from the simulation. This is so that the results can later be sorted into ascending order and manipulated.

**Stats Table:** in range 0-10 for each column. The stats table comprises of 4 different columns that require readings to be read. The first two of these columns involve the user inputting the ranks for the data they received from the simulation. This is then cross referenced from the sorted data in the variables for percentage of plants and light intensity. The difference of these two ranks needs to also be worked and so too does the squared of those differences.

# Testing and Implementation

My program shall be tested in four different categories to make sure that, once it is finished, that no errors are in it and if any improvements need to be made. These four methods of testing include: Algorithms and Dry runs of them, Development Testing, Alpha Testing and Beta Testing

## Testing Strategy:

### Algorithms:

For each of the user requirements, that I have mentioned previously, I will generate an algorithm of how the program will run and what calculations it must do. I will input some test data into these algorithms and calculate the expected result of the algorithm. At the end, the result from the algorithm should match that of the expected result I calculated. Of course, if I find any errors than I will correct them and then rerun the algorithm like I did before.

### Development Testing:

This form of testing will be continuous throughout the making of the actual system. Whenever I add in a new feature or correct something to an existing one, I shall test it in order to show that it works as it should do. Having designed the algorithms will make the task easier to complete as I am not running in blind so to speak but an algorithm is evidently different to a computer program. I will use a lot of test text boxes that will return the values from calculations and also from any data that I had to move around. This is to ensure that all data is being transferred correctly and not being lost or replaced. Also, having the program show me the results of all its calculations will enable me to spot where any problem may lie. When I do come across an error, I will make sure to document it and then explain the changes I have made to fix it. After this, I will rerun the program. I will continue this until the error is resoled and the section works as intended. Another part to the testing will be the use of valid and invalid data in the inputs. This system requires only numerical inputs so allowing the user to input a string character into the program would only mean the program doesn't ultimately work and would produce a lot of unwanted errors. This is why it is very important to make sure that no parts of the program allow for incorrect inputs into the system.

### Alpha Testing:

The alpha testing will be completed once the program is nearing/ at its finish. This is where I will test every aspect of the program as a finished design. This is different to Development Testing as it is to be done after the program is finished and tests every constituent part of the program and makes sure no sections of the program interfere with other parts of the program. A full set of test data will be created and will be similar to the data the end user would use. Expected results for each section will be calculated and the testing of the program should return the same value. On screen help and validation checks will be put in place if I feel they are necessary and not already in place. Any final changes will be made in this testing phase and once I am happy with the end result, I will give it to my end user for the final phase of testing, the beta testing.

### Beta Testing:

During the beta testing phase, the program will be exported and then installed on the user's system. They will then run the program themselves to check if everything runs as it should and whether or not it is to the desired level. I will also provide them with a testing plan that I would like them to follow of all the main features. This will be useful because it means I can make sure that the user is checking the things I want them to check and what I feel is important to program. They can then use the program however they wish, inputting data of their own. I will include a section for the user that will enable them to add what they thought about the program and perhaps any about the results they got. From this, I will think of any amendments I can make and will finalise the program once more.

# Algorithms:

## Simulation 1:

1. Generate random numbers for the image boxes
   1.1. Apply random number to variable "Rnd_Number()"
   1.2. Apply random number to each image box
   1.3. Load correct image according to random number
   1.4. Set variable "box_check" to FALSE for all image boxes

2. Generate Random Number for Co-ordinates
   2.1. At start of simulation or when "New Numbers" button is pressed, generate two numbers and assign them to two different variables called "rnd_labelX" and "rnd_labelY"

3. Selecting the correct image box
   3.1. Is variable box_check(i) FALSE?
   3.2. Are the co-ordinates of the image box the same as the ones given by the random numbers?
   3.3. Make large image box visible and give it the value of RND_Number(i) that was given to the smaller image box that was selected.
   3.4. Load correct image onto large image box based on the value of RND_Number(i)

4. Submitting the reading
   4.1. Is the large image box visible?
   4.2. Is the value in the text box the same as the RND_Number(i) value of the selected text box
   4.3. Assign the inputted number into the global array called "sim1.readings(i).
   4.4. Increase counter by 1
   4.5. If counter reaches 10, refresh the simulation by assigning each image box and the variable RND_Number" a new randomly generated number. Then load a new image accordingly.
   4.6. If second counter reaches 10, take no more readings

5. Going to Stats Page
   5.1. If all readings have been taken from simulation, open Chi$^2$ stats page.

## Chi$^2$ Stats test:

1. Load values from "sim1.readins"
   1.1. Loop until all values of "sim1.readings(i) have been displayed in the results table

2. Complete the Calculations
   2.1. Add up all the values for both Top of Hill and Bottom of Hill and display the total in the results table also. Assign the value of the Top of Hill to a variable "TOHtotal" and the Bottom of Hill to "BOHtotal"
   2.2. Observed value is the same as the total. Assign the value of TOHobserved to that of TOHtotal and vice versa for Bottom of Hill.
   2.3. Expected Number is equal to the mean of the observed numbers. "expected" = ("TOHobserved" + "BOHtotal")/2
   2.4. The difference (O-E) is worked out by subtracting expected number from observed number.
   TOHdifference = TOHobserved – expected
   BOHdifference = BOHobserved – expected

2.5. The difference squared (O-E)2 is the squared result of the difference
TOHDifferenceSquared = $(\text{TOHdifference})^2$
BOHDifferenceSquared = $(\text{BOHdifference})^2$
2.6. Difference$^2$/E is the difference$^2$ divided by the expected number
TOHDSoverE = TOHDifferenceSquared / expected
BOHDSoverE = BOHDifferenceSquared / expected
2.7. C hi$^2$ equals the sum of the DSoverE.
CHI = TOHDSoverE + BOHDSoverE

3. Verifying User's inputted answers
3.1. Compare user's input for each section to the variable that holds the true value
3.2. If the comparisons are all correct, an evaluation text box will appear, telling the user what the stats has shown them.

## Simulation 2:

1. Generate random numbers for the image boxes
1.1. Apply random number to variable "Rnd_Number()"
1.2. Apply random number to each image box
1.3. Load correct image according to random number
1.4. Set variable "box_check" to FALSE for all image boxes

2. Generate Random Number for Co-ordinates
2.1. At start of simulation or when "New Numbers" button is pressed, generate two numbers and assign them to two different variables called "rnd_labelX" and "rnd_labelY"

3. Selecting the correct image box
3.1. Is variable box_check(i) FALSE?
3.2. Are the co-ordinates of the image box the same as the ones given by the random numbers?
3.3. Make large image box visible and give it the value of RND_Number(i) that was given to the smaller image box that was selected.
3.4. Load correct image onto large image box based on the value of RND_Number(i)

4. Submitting the reading
4.1. Is the large image box visible?
4.2. Is the value in the text box the same as the RND_Number(i) value of the selected text box
4.3. Assign the inputted number into the global array called "sim1.readings(i).
4.4. Increase counter by 1
4.5. If counter reaches 10, refresh the simulation by assigning each image box and the variable RND_Number" a new randomly generated number. Then load a new image accordingly.
4.6. If second counter reaches 10, take no more readings

5. Going to Stats Page
5.1. If all readings have been taken from simulation, open the Standard Error/ 95% Confidence Limits stats page.

## Standard Error/ 95% Confidence Limits:

1. Loading values from "sim1.readings(i)"
   1.1. Loop through the values contained within the variable sim1.readings and display each value in the results table.

2. Complete the calculations
   2.1. Calculate the mean of both sets of data and display it in the results table. Assign the value to the variables NFmean and SFmean.
   2.2. Calculate the standard deviation from the means
   NFStandardDeviation = Standard deviation of North Facing data set
   SFStandardDeviation = Standard Deviation of South Facing data set
   2.3. Calculate Standard Error from the standard deviation calculated from both data sets.
   NFStandardError = NFstandardDeviation /
   SFStandardError = SFStandardDeviation /
   2.4. Calculate the 95% Confidence Limits. Confidence Limits = 2 x Standard Error
   2.5. Calculate the range.
   Range1 = NFmean – Confidence Limits
   Range2 = NFmean + Confidence Limits
   Range3 = SFmean – Confidence Limits
   Range4 = SFmean + Confidence Limits

3. Compare users inputted results with the calculated results

4. If the inputted data is correct, show the evaluation text that tells what the stats shows

## Simulation 3:

6. Generate random numbers for the image boxes
   6.1. Apply random number to variable "Rnd_Number()"
   6.2. Apply random number to each image box
   6.3. Load correct image according to random number
   6.4. Set variable "box_check" to FALSE for all image boxes

7. Generate Random Number for lane
   7.1. At the start of simulation, a random number will be generated between 1 and 3. This will signify which lane the user will take their results from.

8. Random Light Intensity
   8.1. Randomly generate Light Intensity values after every 10 seconds and after a value has been entered into the global array sim1.readings.

9. Selecting the correct image box
   9.1. Is variable box_check(i) FALSE?
   9.2. Is the box in the correct lane?
   9.3. If a box has been selected previously and say we give it the index 1 (box(1)), is the next selected box 2 indices larger (box(3))?
   9.4. Make large image box visible and give it the value of RND_Number(i) that was given to the smaller image box that was selected.
   9.5. Load correct image onto large image box based on the value of RND_Number(i)

10. Submitting the reading
    10.1. Is the large image box visible?
    10.2. Is the value in the first text box the same as the RND_Number(i) value of the selected text box
    10.3. Is the value in the second text box the same as the value of the Light Intensity at the given time?
    10.4. Assign the inputted number into the global array called "sim1.readings(i).
    10.5. Assign the inputted number into the globak array called sim1.readings(i+10)
    10.6. Increase counter by 1
    10.7. When counter reaches 10, stop taking more readings.

11. Going to Stats Page
    11.1. If all readings have been taken from simulation, open the Spearman's Rank Correlation Stats page.

## Spearman's Rank Correlation:

1. Load all values from sim1.readings into results table and stats table where applicable
    1.1. Loop through all the values of sim1.readings and display them in the results table and the stats table

2. Sort the two data sets into ascending order
    2.1. Assign first 10 values within sim1.readings to an array called percPlant and the last 10 values to an array called lux
    2.2. Sort the values using bubble sort for both data sets

3. Verifying Stats
    3.1. Ranks
        3.1.1. Loop for all the inputted ranks from the user. Use the entered value as the pointer for percPlant array and lux array respectively. If the value returned is the same as the value at that particular distance, answer is correct
    3.2. Difference
        3.2.1. Difference is calculated by subtracting rannk2 from rank1
        D(1) = rank1(1) – rank2(1)
    3.3. Difference$^2$
        3.3.1. The squared result of the difference calculated previously
        $D2(1) = d(1)^2$
    3.4. Calculating Spearman's Rank
        3.4.1. Calculating Sum of difference2
        3.4.1.1. Add all of the d2 values up
        3.4.2. Use the sum and input into the Spearman's Rank formula
        1 – (6 x Σd2 / n3-n)
    3.5. Is Critical Value correct?
4. If all is correct, evaluation text box should appear, telling the user what the stats has shown

# Algorithms – Dry Runs:

## Simulation 1:

Simulation 1: box(1) box(10)

1. Generate random numbers for the image boxes will use 8 and 3
   - 1.1. Apply random number to variable "Rnd_Number()" RND_Number(1) = 8 RND_Number(10) = 3
   - 1.2. Apply random number to each image box box(1) = 8 box(10) = 3
   - 1.3. Load correct image according to random number box(1).image = 8.png box(10).image = 3.png
   - 1.4. Set variable "box_check" to FALSE for all image boxes box_check(1) = false box_check(10) = false
2. Generate Random Number for Co-ordinates will use 1 and 10
   - 2.1. At start of simulation or when "New Numbers" button is pressed, generate two numbers and assign them to two different variables called "rnd_labelX" and "rnd_labelY" rnd_labelX = 1 rnd_labelY = 10
3. Selecting the correct image box box(10) is correct – will also test box(1)
   - 3.1. Is variable box_check(i) FALSE? box(1) = false (yes) box(10) = false (yes)
   - 3.2. Are the co-ordinates of the image box the same as the ones given by the random numbers? box(1) = no box(10) = yes
   - 3.3. Make large image box visible and give it the value of RND_Number(i) that was given to the smaller image box that was selected. largeImage = RND_Number(10) = 3 .visible = true
   - 3.4. Load correct image onto large image box based on the value of RND_Number(i)
4. Submitting the reading largeImage.image = 3.png
   - 4.1. Is the large image box visible? Yes
   - 4.2. Is the value in the text box the same as the RND_Number(i) value of the selected text box txt_readings.text = 3 RND_Number(10) = 3 → they're the same
   - 4.3. Assign the inputted number into the global array called "sim1.readings(i). Sim1.readings(1) = 3
   - 4.4. Increase counter by 1 counter = counter + 1 → counter = 1
   - 4.5. If counter reaches 10, refresh the simulation by assigning each image box and the variable RND_Number" a new randomly generated number. Then load a new image accordingly.
   - 4.6. If second counter reaches 10, take no more readings
5. Going to Stats Page
   - 5.1. If all readings have been taken from simulation, open Chi² stats page.

## Simulation 2:

Simulation 2: box(2) box(8)

1. Generate random numbers for the image boxes 9 7
   - 1.1. Apply random number to variable "Rnd_Number()" RND_Number(2)=9 RND_Number(8)=7
   - 1.2. Apply random number to each image box box(2)=9 box(8)=7
   - 1.3. Load correct image according to random number box(2).image = 9.png box(8).image = 7.png
   - 1.4. Set variable "box_check" to FALSE for all image boxes box_check(2)=false box_check(8)=false
2. Generate Random Number for Co-ordinates 1 and 8
   - 2.1. At start of simulation or when "New Numbers" button is pressed, generate two numbers and assign them to two different variables called "rnd_labelX" and "rnd_labelY" rnd_labelX = 1 rnd_labelY = 8
3. Selecting the correct image box will select box(8) → correct box
   - 3.1. Is variable box_check(i) FALSE? yes
   - 3.2. Are the co-ordinates of the image box the same as the ones given by the random numbers? box(2) = no box(8) = yes
   - 3.3. Make large image box visible and give it the value of RND_Number(i) that was given to the smaller image box that was selected. largeimage.visible = true largeimage = 7
   - 3.4. Load correct image onto large image box based on the value of RND_Number(i)
4. Submitting the reading largeimage.image = 7.png
   - 4.1. Is the large image box visible? yes
   - 4.2. Is the value in the text box the same as the RND_Number(i) value of the selected text box txt_reading.text = 7 RND_Number(8) = 7 → they're the same
   - 4.3. Assign the inputted number into the global array called "sim1.readings(i). sim1.reading(1) = 7
   - 4.4. Increase counter by 1 counter = counter + 1 → counter = 1
   - 4.5. If counter reaches 10, refresh the simulation by assigning each image box and the variable RND_Number" a new randomly generated number. Then load a new image accordingly.
   - 4.6. If second counter reaches 10, take no more readings
5. Going to Stats Page
   - 5.1. If all readings have been taken from simulation, open the Standard Error/ 95% Confidence Limits stats page.

## Simulation 3:

Simulation 3: box(1) box(3)

6. Generate random numbers for the image boxes 2 3
   - 6.1. Apply random number to variable "Rnd_Number()" RND_Number(1) = 2 RND_Number(3)=3
   - 6.2. Apply random number to each image box box(1)= 2 box(3)= 3
   - 6.3. Load correct image according to random number box(1).image = 2.png box(3).image = 3.png
   - 6.4. Set variable "box_check" to FALSE for all image boxes box_check(1)= false box_check(3)= false
7. Generate Random Number for lane 2
   - 7.1. At the start of simulation, a random number will be generated between 1 and 3. This will signify which lane the user will take their results from.
8. Random Light Intensity [illegible]
   - 8.1. Randomly generate Light Intensity values after every 10 seconds and after a value has been entered into the global array sim1.readings.
9. Selecting the correct image box box(1) will be correct box – in lane 2 box(3) next
   - 9.1. Is variable box_check(i) FALSE? yes
   - 9.2. Is the box in the correct lane? yes
   - 9.3. If a box has been selected previously and say we give it the index 1 (box(1)), is the next selected box 2 indices larger (box(3))? first selection → ignore box(3) = 2nd selection and +2
   - 9.4. Make large image box visible and give it the value of RND_Number(i) that was given to the smaller image box that was selected. RND_Number(1).visible = true RND_Number(1)= 2
   - 9.5. Load correct image onto large image box based on the value of RND_Number(i)
     largeimage.image = 2.png
10. Submitting the reading
    - 10.1. Is the large image box visible? yes
    - 10.2. Is the value in the first text box the same as the RND_Number(i) value of the selected text box ~~largeimage~~ txtreading.text = 2 RND_Number(1) = 2 → Same value
    - 10.3. Is the value in the second text box the same as the value of the Light Intensity at the given time? light_reading.text = [illegible] light intensity= [illegible] → Same value
    - 10.4. Assign the inputted number into the global array called "sim1.readings(i). sim1.readings(1) = 2
    - 10.5. Assign the inputted number into the globak array called sim1.readings(i+10) sim1.readings(11)= [illegible]
    - 10.6. Increase counter by 1 counter = counter + 1 → counter = 1
    - 10.7. When counter reaches 10, stop taking more readings.
11. Going to Stats Page
    - 11.1. If all readings have been taken from simulation, open the Spearman's Rank Correlation Stats page.

## Chi²:

Chi² Stats test:

1. Load values from "sim1.readins" sim1.readings = 1-20 1-10 = TOH 11-20 = BOH
   1.1. Loop until all values of "sim1.readings(i) have been displayed in the results table
2. Complete the Calculations results(i).text = sim1.reading(i) → results(20).text = sim1.readings(20)
   2.1. Add up all the values for both Top of Hill and Bottom of Hill and display the total in the results table also. Assign the value of the Top of Hill to a variable "TOHtotal" and the Bottom of Hill to "BOHtotal" TOHtotal = 55 BOH total = 155
   2.2. Observed value is the same as the total. Assign the value of TOHobserved to that of TOHtotal and vice versa for Bottom of Hill. TOHobserved = 55 BOHobserved = 155
   2.3. Expected Number is equal to the mean of the observed numbers. "expected" = ("TOHobserved" + "BOHtotal")/2 expected = 105
   2.4. The difference (O-E) is worked out by subtracting expected number from observed number.
   TOHdifference = TOHobserved – expected TOHdifference = -50
   BOHdifference = BOHobserved – expected BOH difference = 50
   2.5. The difference squared $(O-E)^2$ is the squared result of the difference
   TOHDifferenceSquared = $(TOHdifference)^2$ TOHDifferenceSquared = 2500
   BOHDifferenceSquared = $(BOHdifference)^2$ BOH Difference Squared = 2500
   2.6. $Difference^2/E$ is the $difference^2$ divided by the expected number
   TOHDSoverE = TOHDifferenceSquared / expected TOHDSoverE = 23.81
   BOHDSoverE = BOHDifferenceSquared / expected BOHDSoverE = 23.81
   2.7. $Chi^2$ equals the sum of the DSoverE.
   CHI = TOHDSoverE + BOHDSoverE CHI = 47.62
3. Verifying User's inputted answers
   3.1. Compare user's input for each section to the variable that holds the true value
   3.2. If the comparisons are all correct, an evaluation text box will appear, telling the user what the stats has shown them.

## Standard Error:

### Standard Error/ 95% Confidence Limits:

1-10 = NF
11-20 = SF

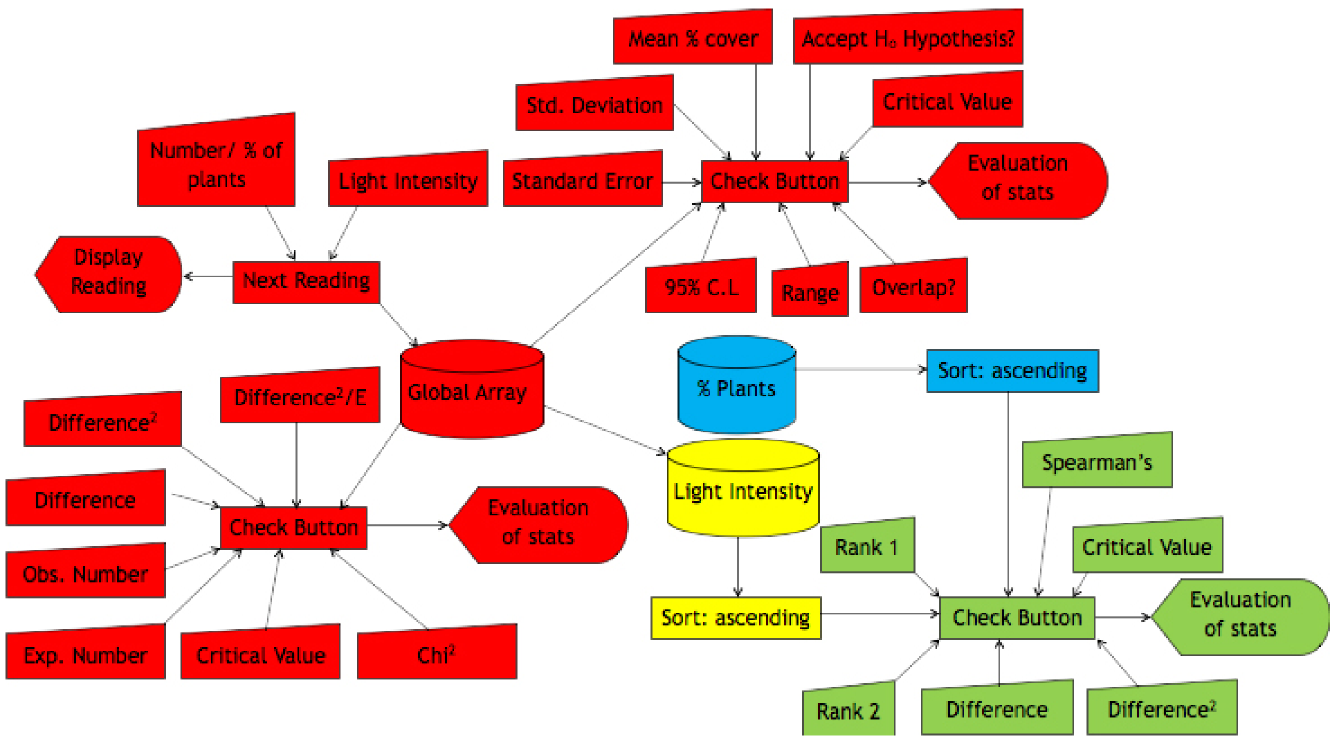
1. Loading values from "sim1.readings(i)" sim1.readings = 1-20
    1.1. Loop through the values contained within the variable sim1.readings and display each value in the results table. results(1).caption = sim1.readings (1) → results(x).caption = sim1.readings(x)
2. Complete the calculations
    2.1. Calculate the mean of both sets of data and display it in the results table. Assign the value to the variables NFmean and SFmean. NFmean = 5.5 SFmean = 15.5
    2.2. Calculate the standard deviation from the means
    NFStandardDeviation = Standard deviation of North Facing data set NFStandardDeviation = 2.87
    SFStandardDeviation = Standard Deviation of South Facing data set SFStandardDeviation = 2.87
    2.3. Calculate Standard Error from the standard deviation calculated from both data sets.
    NFStandardError = NFstandardDeviation /√10 NFStandardError = 0.91
    SFStandardError = SFStandardDeviation /√10 SFStandardError = 0.91
    2.4. Calculate the 95% Confidence Limits. Confidence Limits = 2 x Standard Error CL = 1.82
    2.5. Calculate the range.
    Range1 = NFmean – Confidence Limits range 1 = 3.68
    Range2 = NFmean + Confidence Limits range 2 = 7.32
    Range3 = SFmean – Confidence Limits range 3 = 13.68
    Range4 = SFmean + Confidence Limits range 4 = 17.32
3. Compare users inputted results with the calculated results
4. If the inputted data is correct, show the evaluation text that tells what the stats shows

## Spearman's Rank:

### Spearman's Rank Correlation:

1. Load all values from sim1.readings into results table and stats table where applicable
   - 1.1. Loop through all the values of sim1.readings and display them in the results table and the stats table
2. Sort the two data sets into ascending order
   - 2.1.1. Assign first 10 values within sim1.readings to an array called percPlant and the last 10 values to an array called lux
   - 2.1.2. Sort the values using bubble sort for both data sets
3. Verifying Stats
   - 3.1. Ranks
     - 3.1.1. Loop for all the inputted ranks from the user. Use the entered value as the pointer for percPlant array and lux array respectively. If the value returned is the same as the value at that particular distance, answer is correct
   - 3.2. Difference
     - 3.2.1. Difference is calculated by subtracting rannk2 from rank1
       D(1) = rank1(1) – rank2(1)   D(1) = 2
   - 3.3. Difference$^2$
     - 3.3.1. The squared result of the difference calculated previously
       D2(1) = d(1)$^2$   D2 = 4
   - 3.4. Calculating Spearman's Rank
     - 3.4.1. Calculting Sum of difference$^2$
       - 3.4.1.1. Add all of the d2 values up
     - 3.4.2. Use the sum and input into the Spearman's Rank formula
       $1 - (6 \times \Sigma d^2 / n^3 - n)$
   - 3.5. Is Critical Value correct?
4. If all is correct, evaluation text box should appear, telling the user what the stats has shown

# Systems Flowchart:

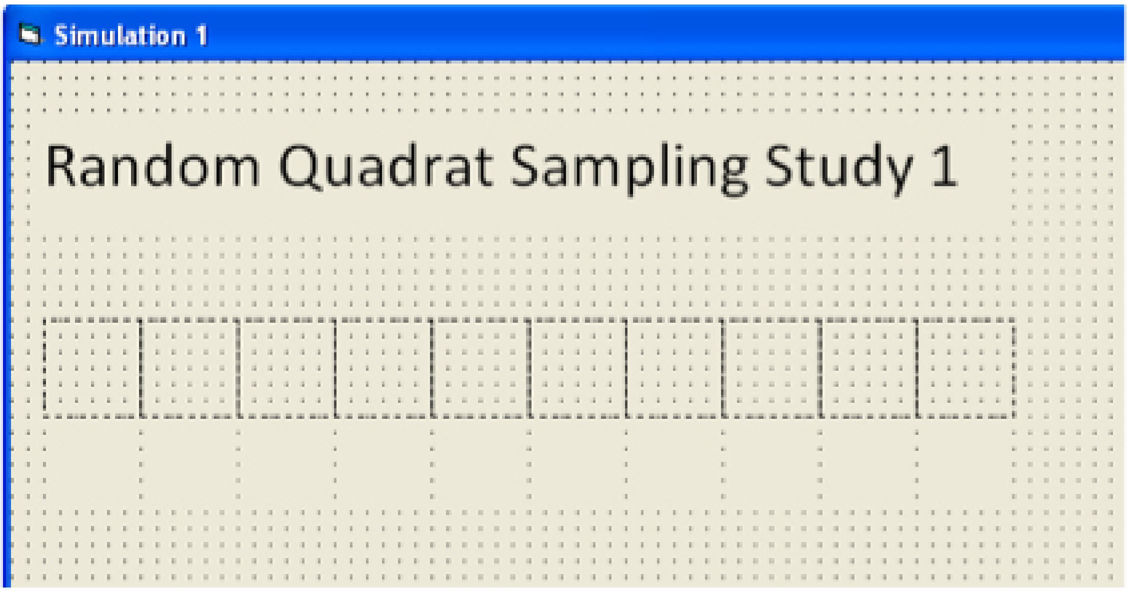


# Development Testing:

## The Sampling Simulation Program:

### Creating the Grid:

The simulation is a big part of the program and something that my end user wants emphasised a lot. So, instead of focusing on the file handling to start off with, I'll try and complete the simulation so that it works.

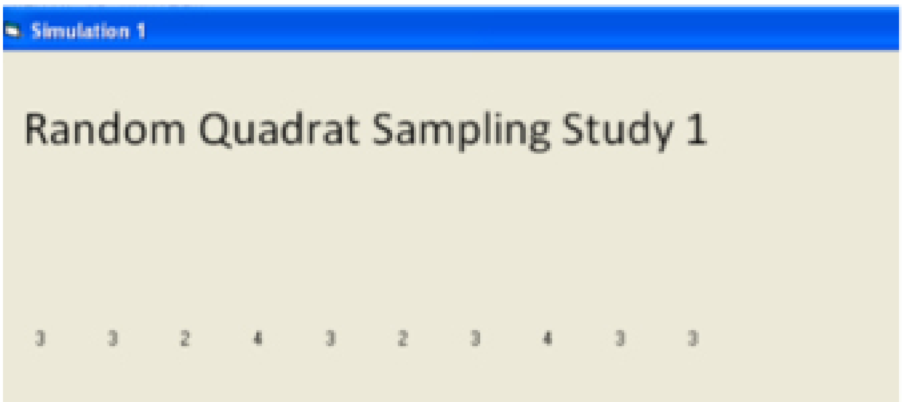



I want to test the simulation first, before I go and do the whole 100 boxes and the collection of the data. Because of this, I'm beginning with just 10 boxes and have labels underneath them.

My reasoning for this is that I'm using image boxes to generate each box and a random number generate to calculate which number that the image box will load (I created some test images in Adobe Fireworks with red circles ranging from 1 to 5). The labels will tell me what the random number is that has been generated and I'll see if it's loaded the correct image accordingly.

```
Dim Sim_Begin As Boolean

Private Sub Form_Load()

'Determines whether the variable "Sim_Begin" is true, based on whether the form is visible or not
If frm_simulation1.Visible = True Then
    Sim_Begin = True
Else
    Sim_Begin = False
End If

'The Random Number generator should activate when the form becomes visible.
If Sim_Begin = False Then
    For i = 0 To 9
        Randomize
        Label12(i).Caption = Int((5 - 1 + 1) * Rnd + 1) 'shows the bounds of the random number generator (between 5 and 1)
        Sim_Begin = True
    Next i
Else
End If


End Sub
```

- This is the code on my form so far. I've declared a global variable called "Sim_Begin." This variable is used to determine whether the simulation should randomly generate numbers or not.
- If I had the code in a Sub_Form, it would randomly generate the numbers once and once only. If the user wishes to return and redo the simulation, they would have to close the program down and reopen it so that it had different results.
- The IF statement asks whether Sim_Begin is FALSE and if it is, it goes into a FOR NEXT loop from 0-9. This is the number of image boxes I have. The code for generating the random number is:
  - Randomize
    Label2(i).caption = Int((5- 1 + 1) * Rnd + 1)
    Sim_Begin = True
- This code begins by using a "Randomize" function without an argument to generate the random number. Label2 is just a means of making sure that the random number generator works and, eventually, that it correspond with the images being loaded. Thus I left it unnamed. After this, I change the value of Sim_Begin to true. Once the form's visibility turns to FALSE again, so too will Sim_Begin.



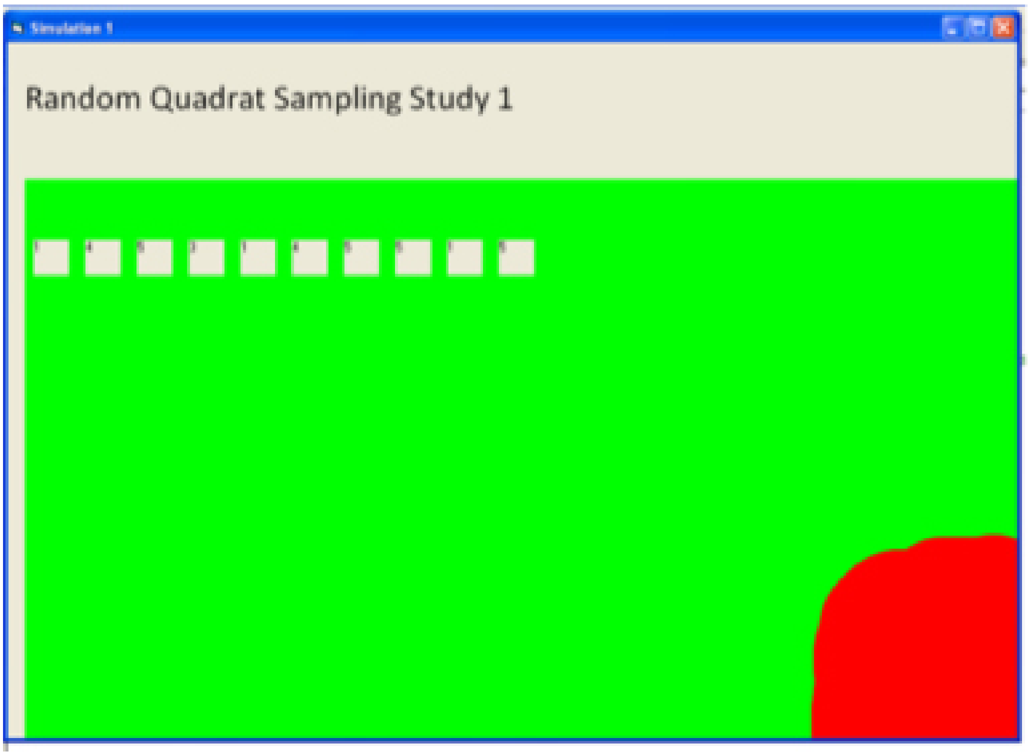

As you can see, the labels have shown me the results of the random generator test and every time I open the page, the values change.

The next course of action is to load the pictures into the image boxes.

```
'The Random Number generator should activate when the form becomes visible.
If Sim_Begin = False Then
    For i = 0 To 9
        Randomize
        Label2(i).Caption = Int((5 - 1 + 1) * Rnd + 1) 'shows the bounds of the random number generator (between 5 and 1)
        box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\1.gif")
```

The code to load the images I used is the one at the bottom of the print screen. I want to just attempt at loading the pictures so I've told the computer to load each image box with a file called "1.gif" What should happen is that each image box should have a green box with a red circle in it.

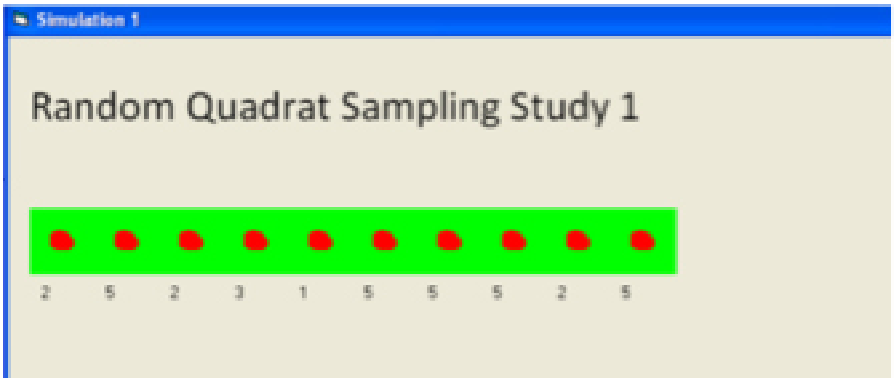
## Error: Image Scale Problem




The problem I've come across is; VB uses a different scale then that used to create the images. Thus, the sizes are largely out of scale. This is a simple fix, needing me to just standardise the image boxes so that they stretch the image down to the size I want.

```
Private Sub Form_Load()

'Determines whe                                on whether the form is visible or not
If frm_simulati
    Sim_Begin = True
Else
    Sim_Begin = False
End If

'The Random Number generator should activate when the form becomes visible.
If Sim_Begin = False Then
    For i = 0 To 9
        Randomize
        Label12(i).Caption = Int((5 - 1 + 1) * Rnd + 1) 'shows the bounds of the random number generator (between 5 and 1)
        box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\1.gif")

'Standardises the image boxes to make sure that they're all uniform
        box(i).Stretch = True
        box(i).Height = 735
        box(i).Width = 735
        Sim_Begin = True
    Next i
Else
```

As you can see, I set the stretch feature of the image box to "TRUE" and I thought it might be useful to set the heights and width that I want my image boxes to have just to be certain that no more errors occur.

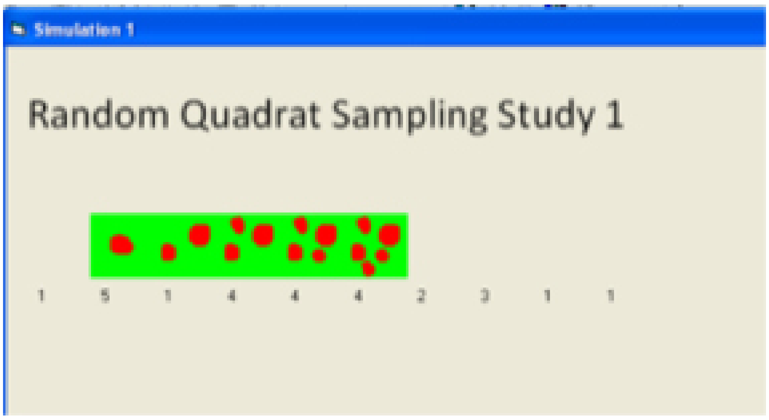



Here's you can see that the program worked like I wanted it too originally. Each image box has stretched the images down the fit into the size I wanted them too. Now the next step is to choice which image box will load what picture based on the value it's gotten from the number generator.

```
'The Random Number generator should activate when the form becomes visible.
If Sim_Begin = False Then
    For i = 0 To 9
        Randomize
        Label12(i).Caption = Int((5 - 1 + 1) * Rnd + 1) 'shows the bounds of the random number generator (between 5 and 1)

'Determines which picture to load based on the numerical value of "i"
            If i = 1 Then
                box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\1.gif")
            ElseIf i = 2 Then
                box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\2.gif")
            ElseIf i = 3 Then
                box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\3.gif")
            ElseIf i = 4 Then
                box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\4.gif")
            ElseIf i = 5 Then
                box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\5.gif")
            End If
```

Here, I have a group of IF statements within each other to choice which image to load based on the value of "i."

## Error: Loading Images



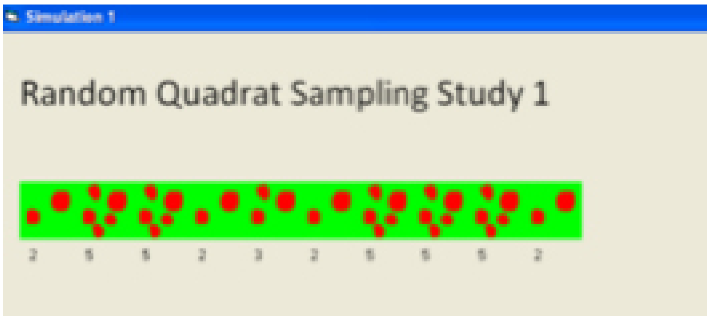

As the above screen shot shows, it did not do what I had hoped. This is because I made a fundamental error and actually confused myself with what was going on in my own code. Furthermore, Arrays start at 0 and so too does my FOR NEXT loop so it was already not going to work by starting the IF statement with i = 1. The problem is that I'm not looking at the random numbers generated. I was looking at the FOR NEXT loop. This meant that only boxes 1-5 would have an image in them and obviously, the images would increase in intervals of 1, much like the FOR NEXT loop does. Technically, the program worked just as it should have done but it was my own fault for not looking into exactly what I was doing.

```
'The Random Number generator should activate when the form becomes visible.

Dim RND_Number As Integer

If Sim_Begin = False Then
    For i = 0 To 9
        Randomize
        RND_Number = Int((5 - 1 + 1) * Rnd + 1) 'shows the bounds of the random number generator (between 5 and 1)
        Label2(i).Caption = RND_Number

'Determines which picture to load based on the numerical value of "i"
         If RND_Number = 1 Then
             box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\1.gif")
         ElseIf RND_Number = 2 Then
             box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\2.gif")
         ElseIf RND_Number = 3 Then
             box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\3.gif")
         ElseIf RND_Number = 4 Then
             box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\4.gif")
         ElseIf RND_Number = 5 Then
             box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\5.gif")
         End If

'Standardises the image boxes to make sure that they're all uniform
        box(i).Stretch = True
        box(i).Height = 735
        box(i).Width = 735
        Sim_Begin = True
    Next i
Else
End If


End Sub
```

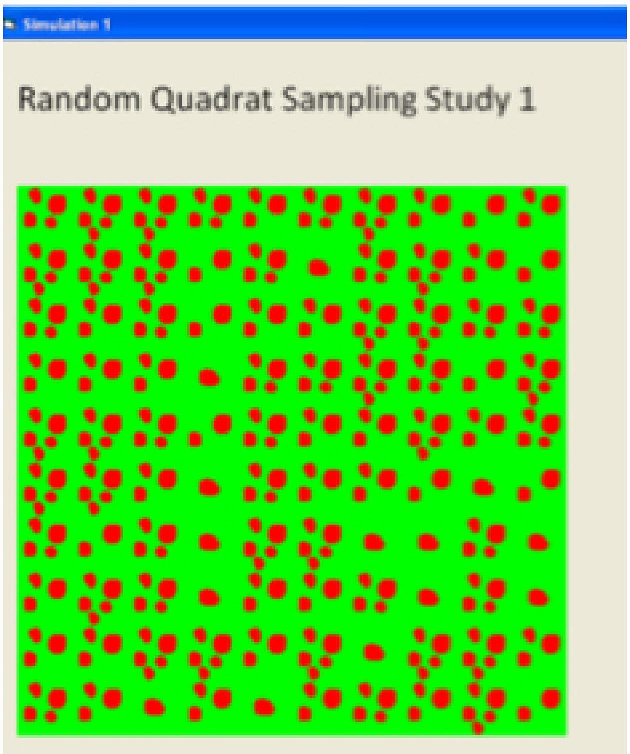
I've changed my code a fair bit to fix the problems I had previously. I've declared a variable called "RND_Number" which will store the random number. This then is what the label2(i).caption makes a comparison with to show the value of the random number.

In the IF statements, I changed all the "IF i =" to "IF RND_Number ="

This should now work as I had hoped originally.



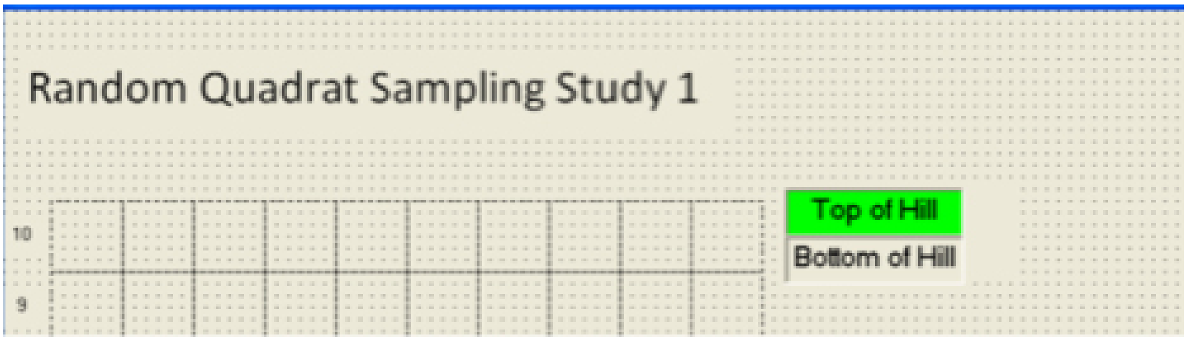
At last, the program works and the correct images have loaded into the correct boxes and to all the boxes.

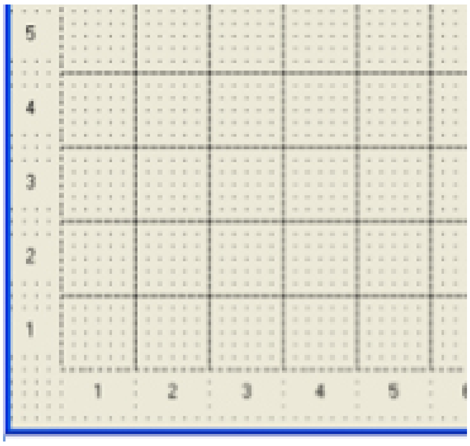



I have now added the 100 boxes and tested the program once again. Thankfully, the program still worked and each box was loaded with an image. Now that this test is out of the way, I can refine it further by adding all the different images I need and by perhaps separating each box so it doesn't appear as horribly confusing. This isn't a necessary refinement for the time being but it'll help make things easier to test and read if it's all smart and near a sort of "finished" state.

**Creating the Simulation Counter:**

Now that I've got the images loading correctly, I need to try and get the actual simulation and the data recording to work. Firstly, I need to create several labels and declare a few new variables.




The labels I created were; a scale along the side of simulation. This is necessary for the user because they need to know where to place their quadrat. The other labels I created were on the die of the simulation, captioned "Top of Hill" and "Bottom of Hill." This is because you take 10 different readings for two different locations. The labels right next to them will be a counter that will record the number of readings the user has already taken. It will be very simple and will simply be a number out of 10.

```
Dim Sim_Begin As Boolean

Dim TOHrecorder As Integer
Dim BOHrecorder As Integer
```

```
'Determines whether the variable "Sim_Begin" is true
If frm_simulation1.Visible = True Then
    Sim_Begin = True
Else
    Sim_Begin = False

    'Initialising TOH and BOH labels
    TOHrecorder = 0
    BOHrecorder = 0

    lbl_TOH.BackColor = &HFF00&
    lbl_BOH.BackColor = &H8000000F
```

In the above screen shot, I've shown that I've declared two new integers called TOHrecorder and BOHrecorder. I need these variables outside of the Form Load event so I declared them as global variables instead of local. Also, in the Form Load event where the visibility of the form is questioned, I've included the initialising of the two labels. Insofar in that I've changed their back colour to be what I want it to start off with and setting the two new variables to be 0.

Now I need to begin with the actual counting up of the variables and changing the colour of the Bottom of Hill to green when the Top of Hill has reached 10 readings. To do this, I'll just add a simple button to simulate this effect.


```
Dim Sim_Begin As Boolean

Dim TOHrecorder As Integer
Dim BOHrecorder As Integer

Private Sub cmd_NextReading()

If TOHrecorder < 10 Then
     TOHrecorder = TOHrecorder + 1
     TOHcounter.Caption = TOHrecorder & "/10"
          If TOHrecorder = 10 Then
               lbl_BOH.BackColor = &HFF00&
               lbl_TOH.BackColor = &H8000000F
          End If
ElseIf BOHrecorder < 10 Then
     BOHrecorder = BOHrecorder + 1
     BOHcounter.Caption = BOHrecorder & "/10"
          If BOHrecorder = 10 Then
               lbl_BOH.BackColor = &H8000000F
          End If
Else
End If


End Sub
```

In this code for the button, I'm asking whether the value of TOHrecorder is less than 10. If it is, I add 1 to its value and then ask again if it now equals 10. If it does, the backcolour changes to a dull grey and Bottom of Hill will go green instead. The same process is repeated until both counters reach 10 where the counting then stops and both boxes turn grey.



The above screen shot is a before and after shot of the 9th and 10th reading of the labels. When it turned, to 10/10, the colour changed back to grey and the Bottom of Hill label turned green, indicating that readings will now be taken into the Bottom of Hill.

### Creating the Random Numbers:

To avoid bias, you need to place the quadrats down randomly. To do this, you'd use a random number table or write down two phone numbers, one above the other and place the quadrat at the coordinates given by the first digits of both numbers. As a result, I need to include another random number generator. The problem with this is that there is a chance that the same combination of numbers may come up. This is, as far as I am aware, an unavoidable circumstance. Therefore, I'll add a button that generates two numbers. Once a reading has been taken, that box will grey out, indicating that it's already been used. This means that, in the event of two numbers coming up again, the user knows not to go there again.

Here's the button I created, with two labels beneath it to show the two numbers generated.


```
Private Sub cmd_rndnumber_Click()

'Random Number Generator #2
Randomize
lbl_X.Caption = Int((10 - 1 + 1) * Rnd + 1)
Randomize
lbl_Y.Caption = Int((10 - 1 + 1) * Rnd + 1)
End Sub
```

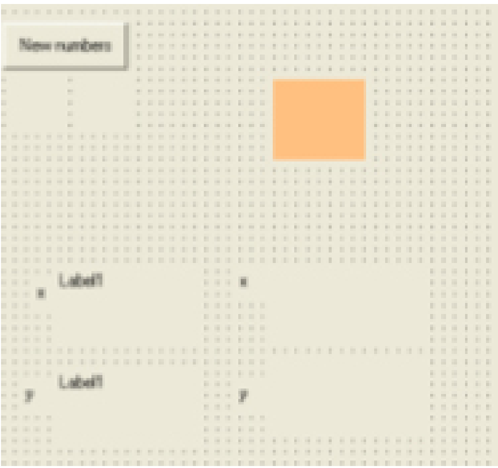
This is the code on the button for generating the random numbers. I have two randomize functions to try and help reduce the likelihood of generating two numbers that have already been used. I then set the caption of these two labels to the value generated.

The problem that I have now is to find out when the correct box has been chosen, based on the numbers generated, and to grey it out. That would most likely require me to create a whole set of identical images with a different background colour though so, for the time being, I'll add a border to the image when it's been selected.

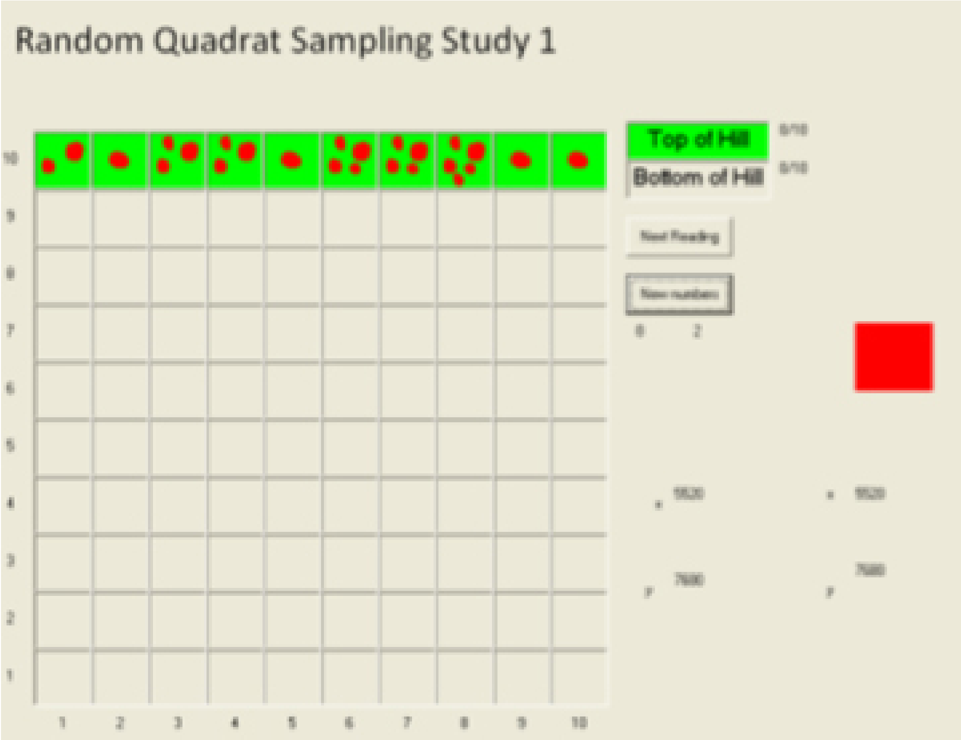
```
Dim BOHrecorder As Integer

Dim rnd_labelX As Integer
Dim rnd_labelY As Integer
Dim random_equationx
Dim random_equationy

Private Sub box_Click(Index As Integer)
Label1.Caption = 8400 - (720 * (rnd_labelY - 1))
Label2.Caption = 480 + (720 * (rnd_labelX - 1))

random_equationx = 480 + (720 * (rnd_labelX - 1))
random_equationy = 8400 - (720 * (rnd_labelY - 1))

Label5.Caption = box(Index).Left
Label6.Caption = box(Index).Top

If box(Index).Top = random_equationy & box(Index).Left = random_equationx Then
    CheckClick.BackColor = &HFF00&
Else
    CheckClick.BackColor = &HFF&
End If

End Sub
```

For this, I've declared two new variables called random_equationX and random_equationY. This is because I need to work out whether the right box has been selected based on the random number generator. Label1 and Label2 are simply for me to see the result of the location of the box I'm clicking on, this is to make sure that the calculations are correct. Label5 and Label6 are labels that show the location of the box I'm selecting. This is to make sure that the calculation is definitely working. Now, when I click on the box, It goes through an IF statement to ask where the location of the box matches the values received through the equation. If it does, it will change a colour of a box, called CheckClick, to green and if not, it'll turn red.

This is the form layout so far.



**Error: Confirming correct box has been selected**



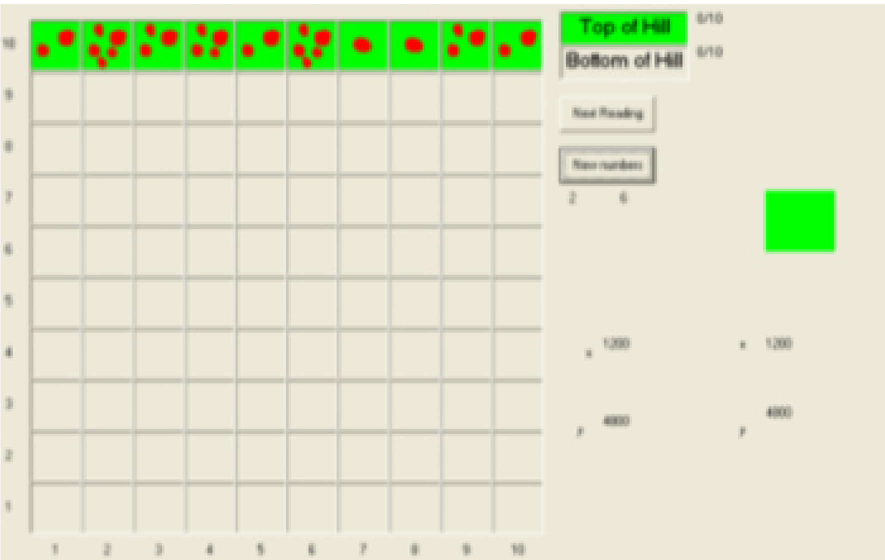

When I ran the program, you can see that the random numbers were 8 and 2. And the two sets of labels show the same numbers. So, everything should work but the box is still turning red and not green. I tried clicking on another box to see if I had accidently coded the colours the other way around. But there was no colour change. I think it is to do with the IF statement I have. I may need to use the test variables I created in the comparison, instead of the equation itself.

The next step I need to take is to create another image box that is much bigger than the others, which will display the image of the image box that has been selected. Once it has been selected and the next reading has been taken, the box will disappear and the image the user had just selected will "grey" out. Indicating that it's a square already been counted.

```
Dim Sim_Begin As Boolean

Dim TOHrecorder As Integer
Dim BOHrecorder As Integer

Dim rnd_labelX As Integer
Dim rnd_labelY As Integer
Dim random_equationx As Integer
Dim random_equationy As Integer
Dim boxX As Integer
Dim BoxY As Integer

Private Sub box_Click(Index As Integer)
'test label
Label1.Caption = 8400 - (720 * (rnd_labelY - 1))
Label2.Caption = 480 + (720 * (rnd_labelX - 1))

'assigning equation to a variable
random_equationx = 480 + (720 * (rnd_labelX - 1))
random_equationy = 8400 - (720 * (rnd_labelY - 1))

'another test variable
Label5.Caption = box(Index).Left
Label6.Caption = box(Index).Top

'Assigning box position to a variable
boxX = box(Index).Left
BoxY = box(Index).Top

'IF for checking if correct box has been chosen
If boxX = random_equationx Then
    If BoxY = random_equationy Then
        CheckClick.BackColor = &HFF00&
    Else
        CheckClick.BackColor = &HFF&
    End If
Else
    CheckClick.BackColor = &HFF&
End If

End Sub
```

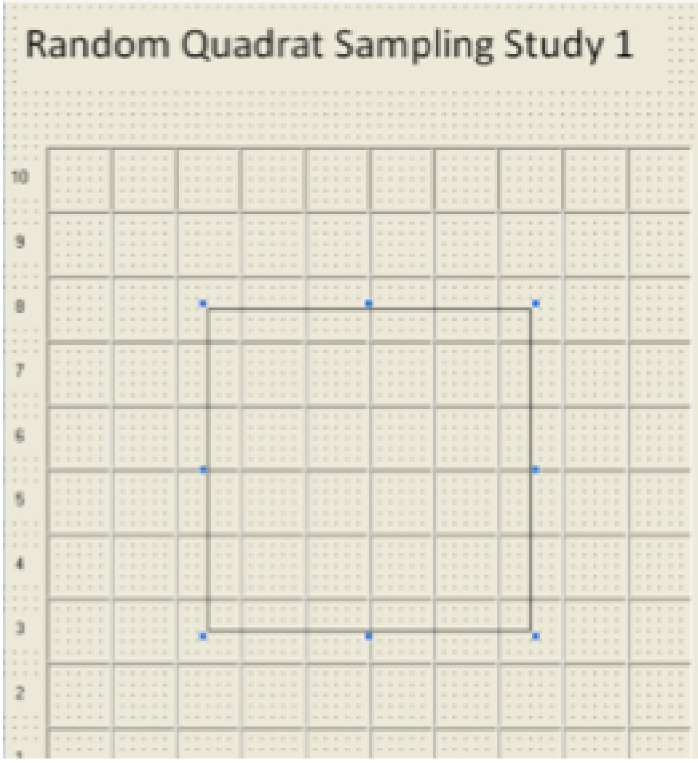
I declared two new variables called boxX and boxY. These variables store the X and Y positions of the selected image box. Everything is the same with the code except for the implementation of these two variables and most noticeably the IF statement. I originally had an "&" symbol connecting the X and the Y positioning but, for testing purposes and ease of understanding, I broke it up so that it can be followed more easily. Firstly, it compares whether boxX and random_equationx have the same value, if not, the colour of the test box will go red. If it is equal, it then compares boxY with random_equationy. If they're equal, then the box will turn green and if not, the box will turn red.

As you can see, it now works as it should do.



**Creating a larger image box:**
The size of the image boxes at the moment won't suffice if I end up having numbers of up to 40. The image is far too small for the user to be able to see all of them so I think it's necessary that I create a larger image box, in the centre of the gird, that'll display the image of the selected box.

The image box is ¼ of the size of the whole grid and will be big enough to display the image for the user to read off of. I have named the image box, "SelectedImage




```
Dim rnd_labelX As Integer
Dim rnd_labelY As Integer
Dim random_equationx As Integer
Dim random_equationy As Integer
Dim boxX As Integer
Dim BoxY As Integer
```

```
Private Sub box_Click(Index As Integer)

'assigning equation to a variable
random_equationx = 480 + (720 * (rnd_labelX - 1))
random_equationy = 8400 - (720 * (rnd_labelY - 1))

'Assigning box position to a variable
boxX = box(Index).Left
BoxY = box(Index).Top

'IF for checking if correct box has been chosen
If boxX = random_equationx Then
    If BoxY = random_equationy Then
        SelectedImage.Picture = box(Index).Picture
        SelectedImage.Visible = True
    Else
        SelectedImage.Visible = False
    End If
Else
    SelectedImage.Visible = False
End If

End Sub
```

I've changed the code I had on the click event, for the image boxes, to include the visibility of the big image box and to change the image inside it to be the correct image of the selected box.

I also initialised the image box in the Form Load event to make sure that it wouldn't do anything unexpected.

```
'Putting a selected image onto a bigger image box - standardising it
SelectedImage.Stretch = True
SelectedImage.Height = 3615
SelectedImage.Width = 3615
SelectedImage.Top = 3720
SelectedImage.Left = 2280
SelectedImage.Visible = False
```

### Allowing only 1 selection per box:

I need to make it so that each box can only be selected once. Once it has been selected, it'll be greyed out and won't be able to be selected again. To do this, I'll make it so that the big image box stays visible once the correct image has been selected and to get rid of it, the user will need to press "Next Reading." This will change the visibility of the image box back to false and will grey out the box that was selected.

```
Dim Sim_Begin As Boolean

Dim TOMrecorder As Integer
Dim BOMrecorder As Integer

Dim rnd_labelX As Integer
Dim rnd_labelY As Integer
Dim random_equationx As Integer
Dim random_equationy As Integer
Dim boxX As Integer
Dim BoxY As Integer
Dim BoxCheck(100) As Boolean
```

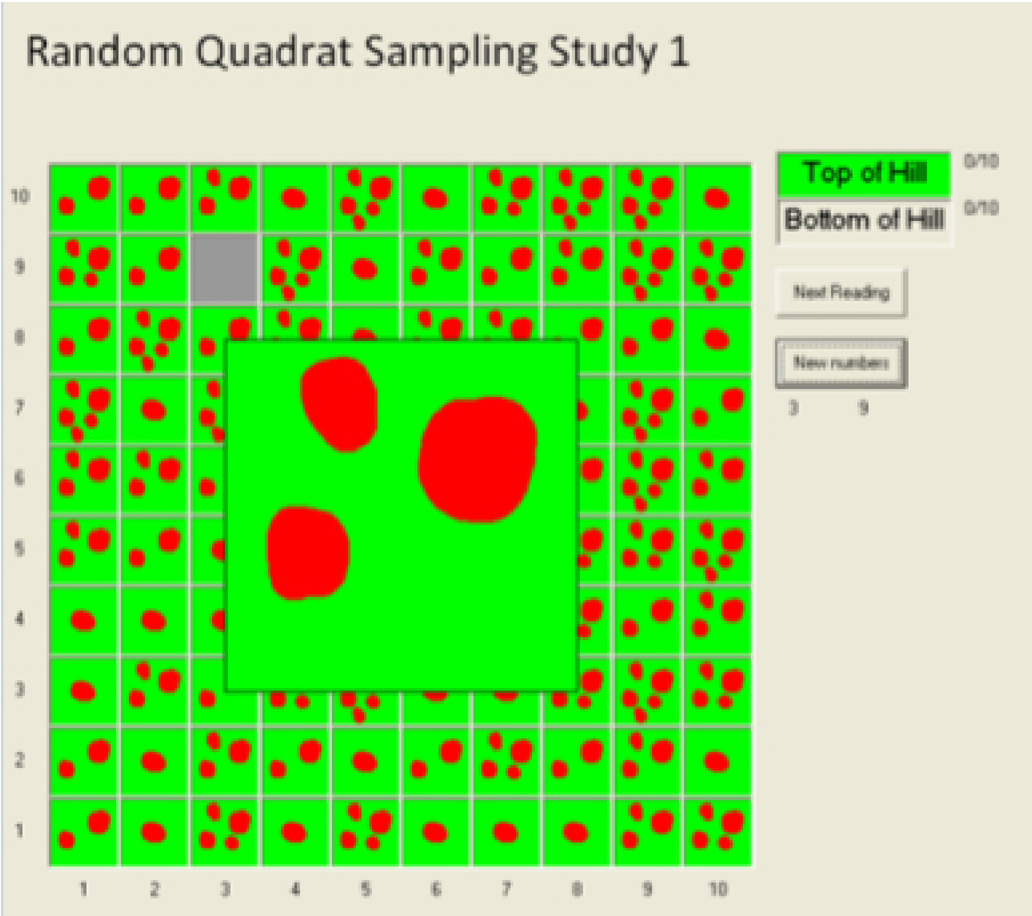
```
Private Sub box_Click(Index As Integer)

'assigning equation to a variable
random_equationx = 480 + (720 * (rnd_labelX - 1))
random_equationy = 8400 - (720 * (rnd_labelY - 1))

'Assigning box position to a variable
boxX = box(Index).Left
BoxY = box(Index).Top

'IF for checking if correct box has been chosen
If BoxCheck(Index) = False Then
    If boxX = random_equationx Then
        If BoxY = random_equationy Then
            SelectedImage.Picture = box(Index).Picture
            SelectedImage.Visible = True
            box(Index).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\grey.gif")
            BoxCheck(Index) = True
        End If
    End If
End If
End Sub
```

To achieve this, I created an array called BoxCheck(100). This array will use the "Index" variable, defined in the click event of the image boxes. Therefore, if box(22) is selected, it will also take the value of 22. This is important because I don't want a box that has already been selected to be selected again. In order to achieve this, I created another IF statement around the one I already had, asking whether Box(Check(Index) = false. If it's false, it means that it hasn't been clicked before.

I also created a simple square that was completely grey and, rather imaginatively named it "grey." When the box is selected, the image loaded to it will change to this grey image.

I have a simple FOR NEXT loop in the Form Load event that goes from 0 to 99 and ensures that, to begin with, all the BoxCheck values are set to false.

Also, when a box is pressed and the big image box comes up, I don't want the user to be able to generate a new pair of numbers and click another image. To stop this, I'll make it so that you have to click on "Next Reading" to set the visibility of the image box to False and to generate a new set of random numbers, the visibility of the big image box must be false.




From the above screen shot, the numbers generated were 3 and 9. When I clicked on the box with those coordinates, it greyed out and the big image displayed the image. I was rapid pressing "New Numbers" and they didn't change.

**Refreshing Simulation for "Bottom Of Hill:"**

The top and bottom of hill are two different sites and therefore, they'll have different numbers of plants. Because of this, I need to generate a new set of plants, and refresh those that have been greyed out, when the 10 readings for the top of hill have been reached.

```
Private Sub cmd_NextReading_Click()

'Counting up
If SelectedImage.Visible = True Then
    If TOHrecorder < 10 Then
        TOHrecorder = TOHrecorder + 1
        TOHcounter.Caption = TOHrecorder & "/10"
        SelectedImage.Visible = False
            If TOHrecorder = 10 Then
                lbl_BOH.BackColor = &HFF00&
                lbl_TOH.BackColor = &H8000000F
                SelectedImage.Visible = False

                'Copy of FOR NEXT to refresh image boxes with new images
                For i = 0 To 99
                    Randomize
                    RND_Number = Int((5 - 1 + 1) * Rnd + 1) 'shows the bounds of the random number generator (between 5 and 1)

                    'Determines which picture to load based on the numerical value of "i"
                    If RND_Number = 1 Then
                        box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\1.gif")
                    ElseIf RND_Number = 2 Then
                        box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\2.gif")
                    ElseIf RND_Number = 3 Then
                        box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\3.gif")
                    ElseIf RND_Number = 4 Then
                        box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\4.gif")
                    ElseIf RND_Number = 5 Then
                        box(i).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\5.gif")
                    End If
                Next i

                'Resetting the BoxCheck to all false
                For y = 0 To 99
                    BoxCheck(y) = False
                Next y
```

In the next reading button event, I added the same cod I had in the sub form load event that loads an image to every box and makes all of the BoxCheck values to false. When the number of readings has reached 10, this code will execute, effectively refreshing everything as It was when the form loaded.

**Taking Readings:**

Now that the foundations of my program are set, I can begin on taking the data. I need to have a text box, in which the number of plants is to be recorded. Once the "Next reading" button has been selected, and data has been entered into the text box, it'll allow you to progress with the simulation but also save the result in an array, of size 0-19 (0-9 for Top, 10-19 for Bottom). The number inputted into the box needs to be the same as the number of plants in the image. To achieve this, I need to create another array which stores the random number, given to each box to load the images.

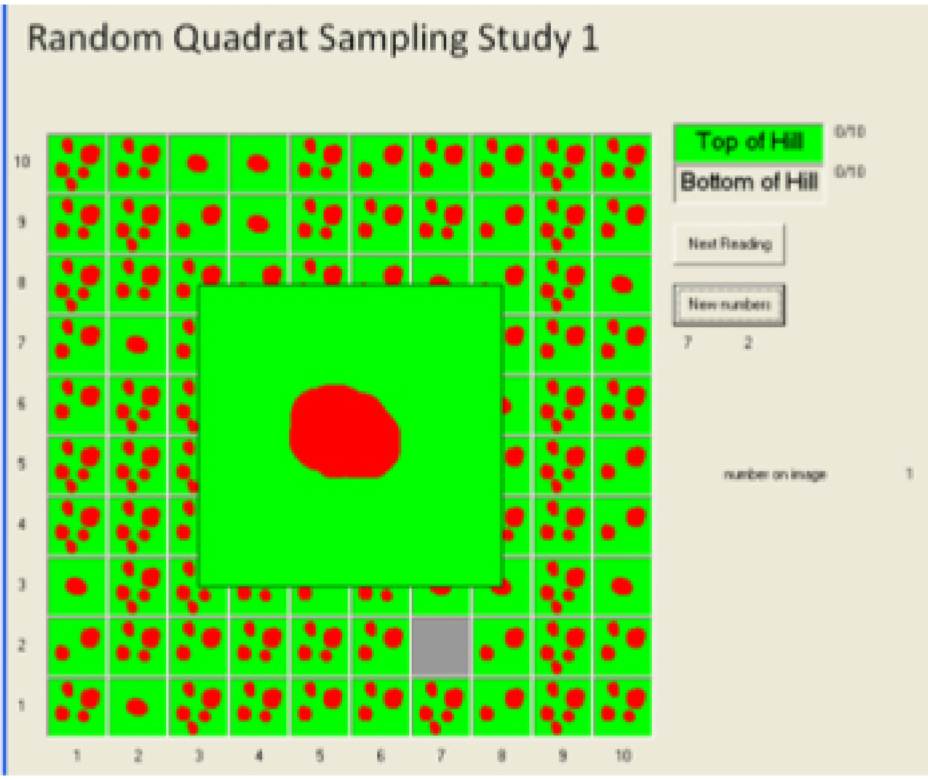
This way, I can compare the number inputted to this array.

```
Dim Sim_Begin As Boolean

Dim TOHrecorder As Integer
Dim BOHrecorder As Integer

Dim rnd_labelX As Integer
Dim rnd_labelY As Integer
Dim random_equationx As Integer
Dim random_equationy As Integer
Dim boxX As Integer
Dim BoxY As Integer
Dim BoxCheck(100) As Boolean

Dim RND_Number(100) As Integer
```

"RND_Number" was a variable I had originally locally declared in my Form Load event that temporarily stored the value of the random number I had generated. However, I realised I don't want this to be temporary so I globally changed it into an array.

```
'IF for checking if correct box has been chosen
If BoxCheck(Index) = False Then
    If boxX = random_equationx Then
        If BoxY = random_equationy Then
            SelectedImage.Picture = box(Index).Picture
            SelectedImage.Visible = True
            box(Index).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\grey.gif")
            BoxCheck(Index) = True
        End If
    End If
End If

'testing whether random number can be recorded
Label12.Caption = RND_Number(Index)

End Sub
```

The above screen shot is of the box click event. I simply added another caption that will display the value of the random number that was assigned to the corresponding box.




This screen shot shows the labels I have created to test the program and, as you can see, the correct number was displayed in the label.

I've now created a text box for the user to write their estimations in. However, if the user forgets to enter data or enters incorrect data, I need to inform them of their mistake. To do this, I added another label that's invisible. This will display the value of the image box's random number. This is necessary because I can't make direct comparison between the actual variable that stores the random number for each box and the value in the text box. So an intermediate is required to make the comparison. I'll add another IF statement within the "Next reading" button click event. This will ask if the text box and the hidden label have the same value, if it does it will continue with the code. If they're not equal or no data has been entered, a message will pop up, notifying them of the error.

```
                End If
            ElseIf BOMrecorder < 10 Then
                BOMrecorder = BOMrecorder + 1
                BOMcounter.Caption = BOMrecorder & "/10"
                txt_readings.Text = ""
                SelectedImage.Visible = False
                    If BOMrecorder = 10 Then
                        lbl_BOM.BackColor = &H8000000F
                    End If
            End If
        ElseIf txt_readings.Text = " " Then
            MsgBox "Please enter a reading before continuing!"
        ElseIf txt_readings.Text <> lbl_hidden.Caption Then
            MsgBox "Incorrect value inputted! Please try again!"
        End If
    End If
End If
```

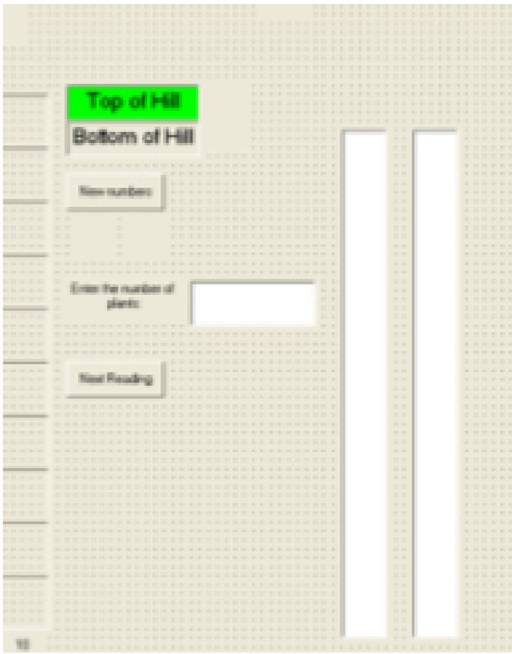
**Recording the Readings:**

I've created a module and therein created a type called Sim1_Readings. This contains variable called "readings" of dimensions 0 to 19. This will store the 20 readings the user will take. I then created a global variable called Sim1 for use throughout the entire program.

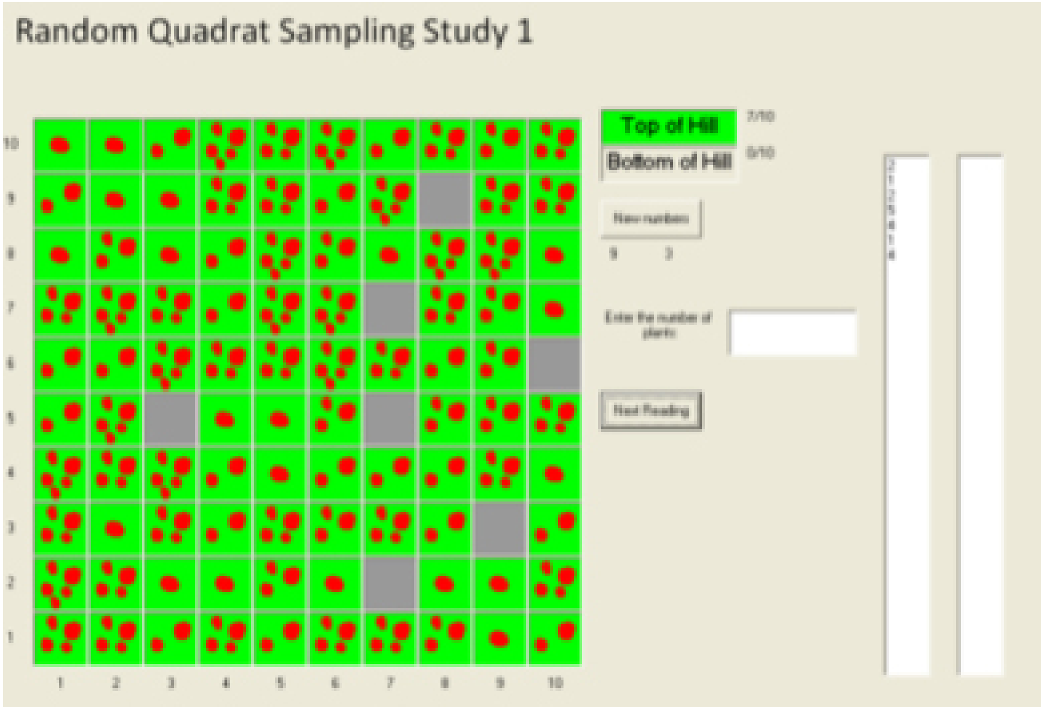
```
'Data Structure (Simulation etc)
Type Sim1_Readings
readings(0 To 19) As Integer
End Type


Global sim1 As Sim1_Readings
```

This means I can use the values recorded from the simulation throughout the whole form and won't be form specific.

```
        TOHrecorder = TOHrecorder + 1
        pointer = pointer + 1
        TOHcounter.Caption = TOHrecorder & "/10"
        sim1.readings(pointer) = txt_readings.Text
        Label13.Text = Label13 & sim1.readings(pointer) & vbNewLine
        txt_readings.Text = ""

    ElseIf BOHrecorder < 10 Then
        BOHrecorder = BOHrecorder + 1
        pointer = pointer + 1
        BOHcounter.Caption = BOHrecorder & "/10"
        sim1.readings(pointer) = txt_readings.Text
        Label14.Text = Label14 & sim1.readings(pointer) & vbNewLine
        txt_readings.Text = ""
```



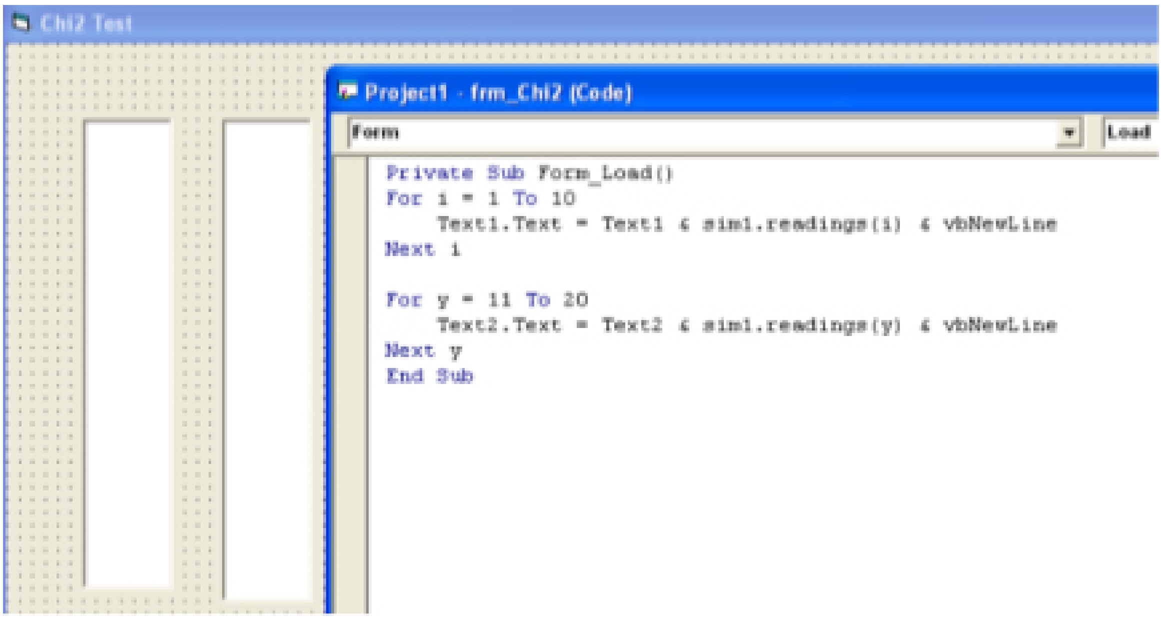

The above screen shots of code shows a pointer (to tell which location the recorded value will be in in the array). The text boxes I have to the side will display the value of the array as the values are recorded. This is merely for testing whether the correct values are being recorded and whether or not the array actually works.



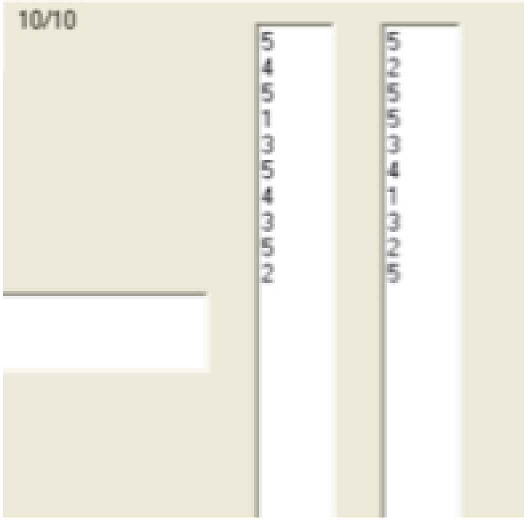

Here's a screen shot of my program running and you can see that the values of the array are being displayed in the box. This shows me that the array works and the simulation is, on a fundamental, now finished. I need to improve the aesthetics of the simulation insofar as I need to actually add "Plants" into the simulation and not these red substitutions. However, this is an end stage refinement.

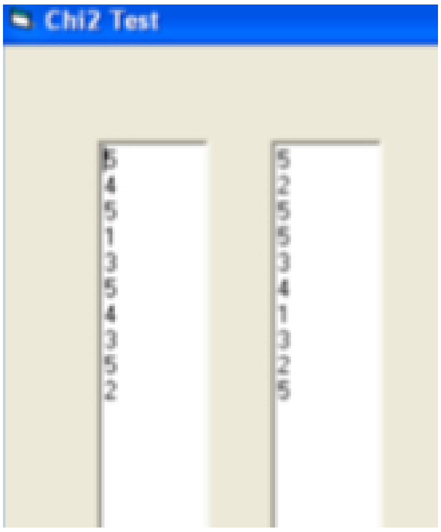
### Transferring Data across forms:

Now that the simulation works, I need to test that the data collected from the simulation can be transferred to another form correctly.



I've added two text boxes to a new form and in the sub form load event; I added two "FOR NEXT" loops that will add the values from the simulation into these text boxes.

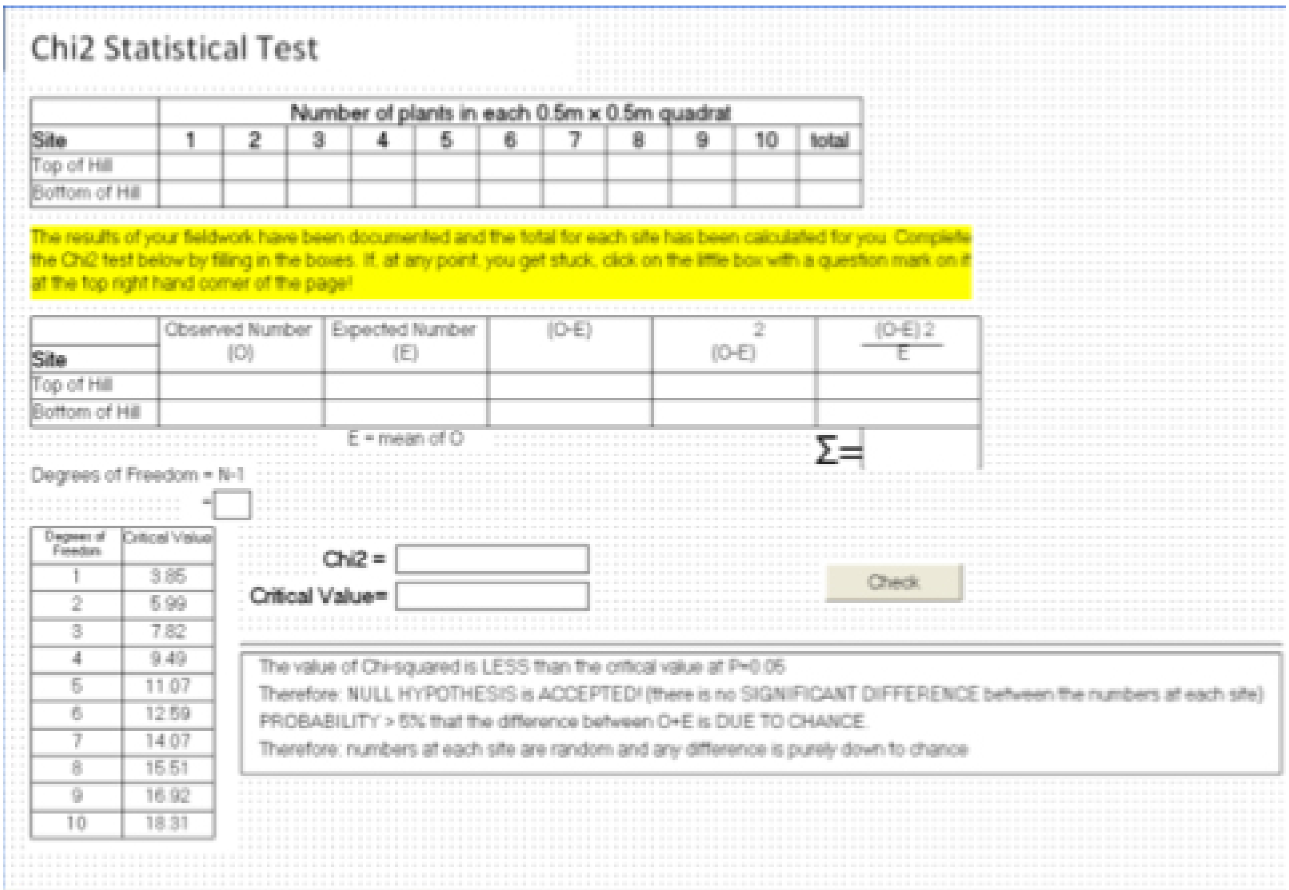



As you can see from the above two screen shots. Both pairs of text boxes have the same values within them. This definitively shows that the transfer of data is working as it should do.

The statistical tests in this part of the biology specification are mainly straightforward mathematical equations but I'll need to break it down into smaller steps so that the user can follow what has happened.

### Chi² Test:



Because there isn't really any code in this form besides some maths equations, I figured it would be easiest to just finish the whole form and then talk about what's happened and where.

Firstly, the table of results at the top of the page takes the values of the array and inputs them into the boxes. It then works out the total of all the results and puts that into a separate text box at the end.

```
Private Sub Form_Load()
TOHtotal = 0
BOHtotal = 0

For i = 1 To 10
    results(i - 1).Caption = sim1.readings(i)
    TOHtotal = TOHtotal + sim1.readings(i)
Next i
lbl_TOHtotal.Caption = TOHtotal

For y = 11 To 20
    results(y - 1).Caption = sim1.readings(y)
    BOHtotal = BOHtotal + sim1.readings(y)
Next y
lbl_BOHtotal.Caption = BOHtotal
```

This is the code I used to transfer the data across from the simulation and working out the total and outputting it.

The next thing I did was to create the table in which the statistical tests follow. Although it would have been much simpler to simply give the user the answer, my end user wants a step by step guide of sorts so I broke each step down.

Absolutely everything on this form, except the table at the top, is controlled by the button with the caption, "Check." All other boxes are mostly text boxes where the user inputs data. Once the check button has been pressed, the computer will go through each box, making sure that the inputted answer is correct. If it is correct, then the background colour of the box will turn green and if it is wrong, it will turn red. Also, it was important that the text boxes that were correct couldn't be changed. To fix this, I also made sure that the text box would "lock" if it was correct. Also, I added a counter called "correct" that is the data type integer which counts up when something is correct and only when the counter reaches max, will the text boxes at the bottom appear. This is because the text boxes (confined into frames) contain the explanation you need to write after you complete the test. These should appear at the end and not halfway through when you press the check button for the first time.

```
Private Sub Command1_Click()

correct = 0

'Top of Hill verification
If correct <> 7 Then
    If Val(lbl_TOHo.Text) = TOHobserved Then
        lbl_TOHo.BackColor = &HFF00&
        lbl_TOHo.Locked = True
        If Val(lbl_TOHe.Text) = expected Then
            lbl_TOHe.BackColor = &HFF00&
            lbl_TOHe.Locked = True
            If Val(lbl_TOHoe.Text) = TOHdifference Then
                lbl_TOHoe.BackColor = &HFF00&
                lbl_TOHoe.Locked = True
                If Val(lbl_TOHoe2.Text) = TOHDiffereceSquared Then
                    lbl_TOHoe2.BackColor = &HFF00&
                    lbl_TOHoe2.Locked = True
                    If Val(lbl_TOHoe2e.Text) = TOHDSoverE Then
                        lbl_TOHoe2e.BackColor = &HFF00&
                        lbl_TOHoe2e.Locked = True
                        correct = correct + 1
                    Else
                        lbl_TOHoe2e.BackColor = &HFF&
                    End If
                Else
                    lbl_TOHoe2.BackColor = &HFF&
                End If
            Else
                lbl_TOHoe.BackColor = &HFF&
            End If
        Else
            lbl_TOHe.BackColor = &HFF&
        End If
    Else
        lbl_TOHo.BackColor = &HFF&
    End If

    'Bottom of Hill verification
```

The screen shot above is from the button click event of the check button. The code you can see is that of the results of the comparisons for the top of the hill. The exact same code if applied to the bottom of the hill but the names of the text boxes are different. I put it into one big IF statement because it means that it will show when the user has made a mistake. Even if they get one box correct but the following box wrong, the box that follows that one will remain colourless. This clearly shows the user where they have made the mistake. I originally had it as separate IF statements but I prefer this idea because it is much clearer.

```
If Val(txt_dof.Text) = 1 Then
    txt_dof.BackColor = &HFF00&
    txt_dof.Locked = True
    correct = correct + 1
Else
    txt_dof.BackColor = &HFF&
    correct = 0
End If

If Val(txt_chi.Text) = CHI Then
    txt_chi.BackColor = &HFF00&
    txt_chi.Locked = True
    correct = correct + 1
Else
    txt_chi.BackColor = &HFF&
    correct = 0
End If
```

This is the code for the various text boxes outside of the grid. These are separate so I decided to keep them in their own IF statements.

## Chi$^2$ Statistical Test

| | Number of plants in each 0.5m x 0.5m quadrat | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Site | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | total |
| Top of Hill | 3 | 6 | 2 | 5 | 3 | 5 | 4 | 2 | 5 | 4 | 39 |
| Bottom of Hill | 1 | 4 | 6 | 2 | 3 | 5 | 8 | 1 | 2 | 8 | 40 |

The results of your fieldwork have been documented and the total for each site has been calculated for you. Complete the Chi2 test below by filling in the boxes. If, at any point, you get stuck, click on the little box with a question mark on it at the top right hand corner of the page!

| Site | Observed Number (O) | Expected Number (E) | (O-E) | (O-E)$^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|---|---|
| Top of Hill | 39 | | | | |
| Bottom of Hill | 40 | | | | |

E = mean of O

Σ=

Degrees of Freedom = N-1

=

For the Chi$^2$ test, the observed number has already been inputted and I have checked it using the Check button. As both text boxes have shown up green, this means that there is no problem with copying the total number of plants recorded at each site.

The value of E is simply the mean of the observed values. This would mean the expected value would be 39.5.
39+40 = 79
79 / 2 = 39.5

| Site | Observed Number (O) | Expected Number (E) | (O-E) |
|---|---|---|---|
| Top of Hill | 39 | 39.5 | |
| Bottom of Hill | 40 | hello world | |

E = mean of O

For the Top of the Hill readings, I entered the expected value I calculated (39.5). This resulted in the box turning green, meaning it was correct. For the Bottom of Hill results, I typed in "hello world." The box remained red, meaning the program worked out that it was wrong.

For the difference between the observed and expected, the Top of Hill should be -0.5 and Bottom of Hill should be 0.5. this is simply subtracting the expected value from the observed value.

| Site | Observed Number (O) | Expected Number (E) | (O-E) | $(O-E)^2$ |
|---|---|---|---|---|
| Top of Hill | 39 | 39.5 | -0.5 | |
| Bottom of Hill | 40 | 39.5 | 0.five | |

E = mean of O

Once again, I used the correct value for Top of Hill and a value that contained string characters in it for the Bottom of Hill. As you can see above, the difference I worked out for Top of Hill was correct as the box turned green. I inputted "0.five" into Bottom of Hill and it turned red, indicating it was wrong.

For the squared difference, the value for both of these results should be 0.25. Squaring a number means that it will always be positive so: 0.5 x 0.25 = 0.25.

| Expected Number (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|
| 39.5 | -0.5 | 0.twofive | |
| 39.5 | 0.5 | 0.25 | |

E = mean of O

$\Sigma=$

As you can see above, I entered the correct value into Bottom of Hill this time round and it turned green, meaning the value was correct. For the Top of Hill, I entered "0.twofive" and it turned red, meaning it was incorrect. I changed the two around because I need to test that the Top of Hill doesn't accept string values also.

For the squared difference over the expected value, the result will be 0.0063291139. This was calculated: 0.25 / 39.5.

Because I have set each value to round up to 2 decimal places, the value that should be seen as correct should be 0.01. For the Bottom of Hill, I will enter the value "0.0one," expecting the result to go red.

| Expected Number (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|
| 39.5 | -0.5 | 0.25 | 0.01 |
| 39.5 | 0.5 | 0.25 | 0.0one |

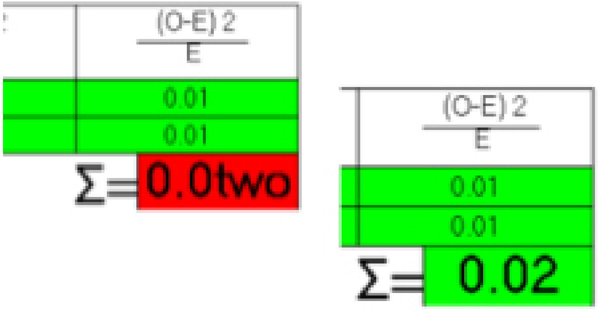
E = mean of O

$\Sigma=$

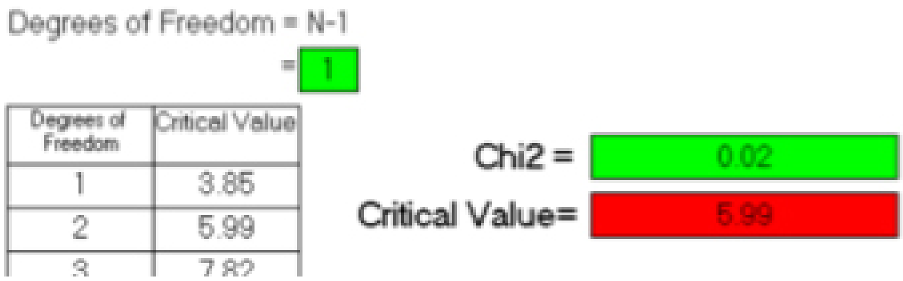
As expected, the Top of Hill turned green and the Bottom of Hill remained red.

For the sum, I need to add the two together. As they are the same value, I can simply multiply 0.01 by 2; this will equal 0.02. However, I will initially enter "0.0two" to make sure that it doesn't turn green.

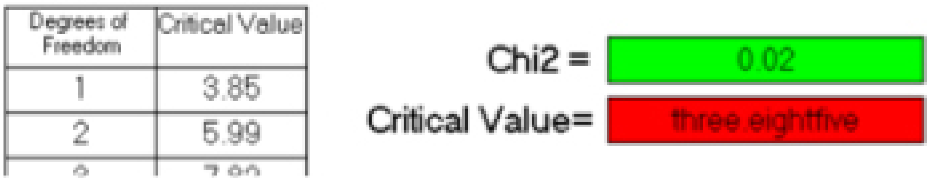
As you can see, the value did not turn the box to green.




Once I entered the value I had worked out, the box turned green, meaning that it was correct.

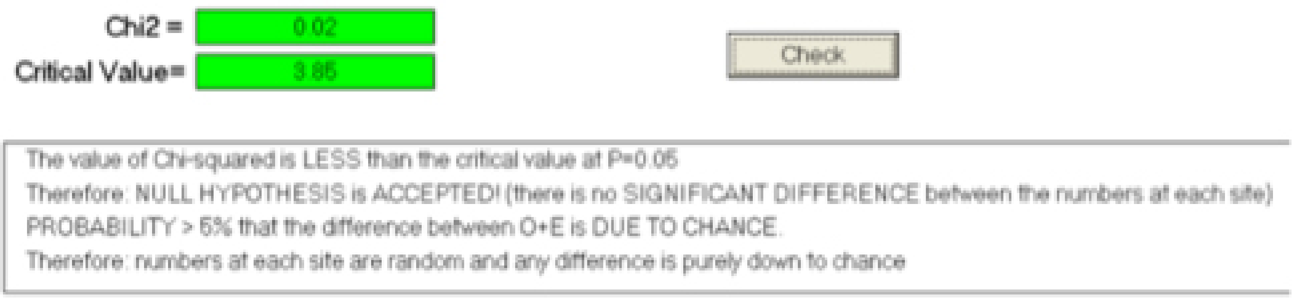



I entered the correct degrees of freedom and the Chi² value in these two text boxes. The critical value should be 3.85 as the Degrees of Freedom equals 1. I entered 5.99 and the box remained red.



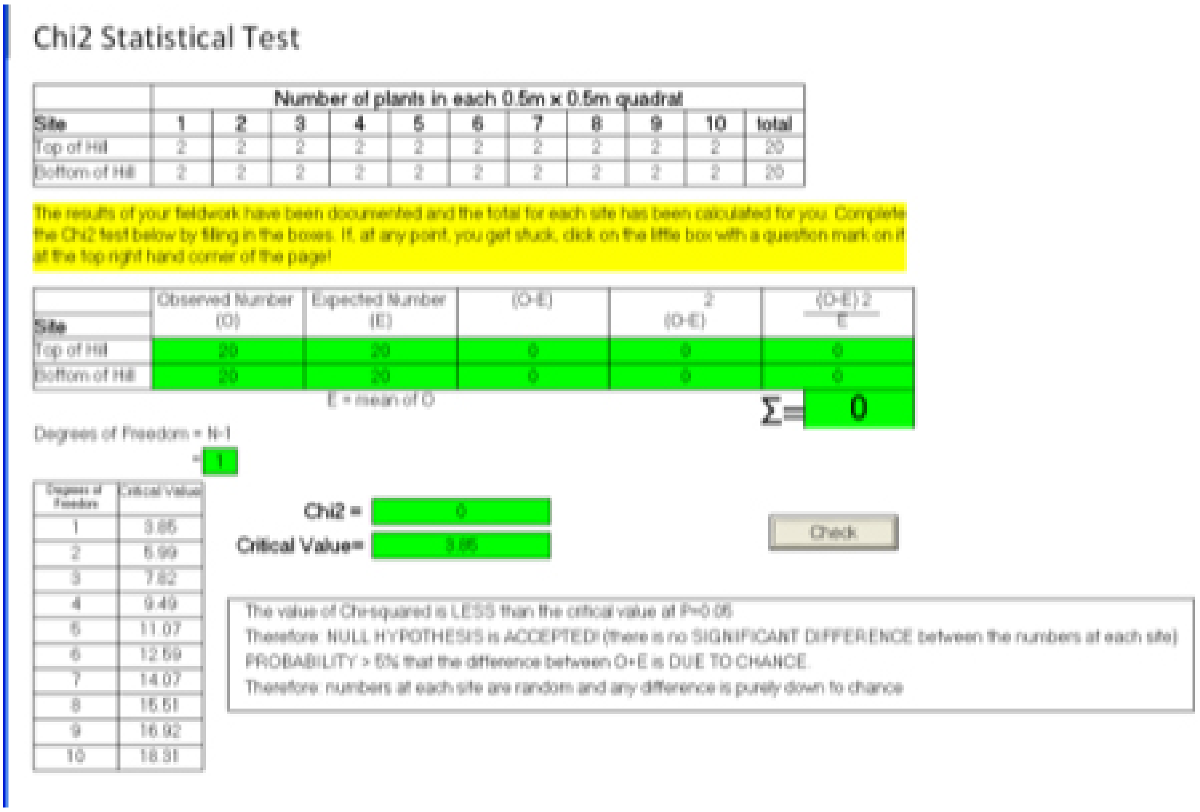

I have now entered "three.eightfive" into the box and the box still remained red, indicating it was wrong.

If I change it to 3.85, the box should go green and one of two hidden text boxes should appear. One for when the Chi² test shows that there is no significant difference in the means and one for one there is a significant difference. In this case, the value of Chi² is less than the critical value. This means that there is no significant difference in the means and that we ought to accept the null hypothesis.



As you can see, the correct text box showed up and the critical value box turned green when I entered the value of 3.85.

Here's the form being run with false test data. As you can see, any wrong answer is in red and the following box will not change colour.




When everything is correct, one of two frames (with text boxes confined to them) will become visible. This is the important evaluation and all the key words the user needs to remember have been capitalised to stand out.

The completion of this form now means that I have finished one simulation and its corresponding statistical test. Creating the initial simulation took a very long time because I was trying to figure out what I needed to do but, now that it is complete, I can now create the following two simulations much more quickly and with fewer errors than before. The same can hopefully be said for the statistics that goes alongside them as well.

## Standard Error/ 95% Confidence Limits:

### Creating a Function for Standard Deviation:

The simulation for Study 2 is exactly the same as Simulation 1 with the exception of some wording of captions and the images that are loaded into the boxes. Due to this, all I needed to do was copy across the same code as Simulation 1 and make these minor changes. However, the stats section of this simulation was rife with errors and problems.

Firstly, I needed to deal with the elements in the sub form load event.

```
Private Sub Form_Load()
For i = 1 To 10
    results(i - 1).Caption = sim1.readings(i)
    NFmean = NFmean + sim1.readings(i)
Next i
lbl_nfmean.Caption = NFmean / 10

For y = 11 To 20
    results(y - 1).Caption = sim1.readings(y)
    SFmean = SFmean + sim1.readings(y)
Next y
lbl_sfmean.Caption = SFmean / 10

NFStandardDeviation = Round(NFStdDev(Val(lbl_nfmean)), 2)
NFStandardError = Round(NFStandardDeviation / Round(Sqr(10), 2), 2)

SFStandardDeviation = Round(SFStdDev(Val(lbl_sfmean)), 2)
SFStandardError = Round(SFStandardDeviation / Round(Sqr(10), 2), 2)

Tclick = False
Nclick = False
Aclick = False
Rclick = False

range1 = (Val(lbl_nfmean) - (2 * NFStandardError))
range2 = (Val(lbl_nfmean) + (2 * NFStandardError))
range3 = (Val(lbl_sfmean) - (2 * SFStandardError))
range4 = (Val(lbl_sfmean) + (2 * SFStandardError))

End Sub
```

The first 11 lines of code is transferring the values recorded from the simulation into a table of results that I have created (similar to the one in the Chi$^2$ stats test. This test works with Means rather than total values so I need to calculate the mean of the results and output that as well.

This stats test is much more complicated as you need to calculate Standard Deviation and Standard Error before you can make an analysis of your results. This means that I needed a way to calculate the Standard Deviation and Standard Error. I didn't want to have to rewrite both sections twice so I created a function to do this for me. Also, this means the code is away from the big bulk of verification that is done at the end so it will make it much easier to read and to identify where a problem lies.

```
Function NFStdDev(mean As Double) As Double
    Dim value(15) As Double
    Dim Temp As Double
    Dim Temp2 As Double

    NFStdDev = 0
    For i = 1 To 15
        value(i) = 0
    Next i

    Temp = 0

    For i = 1 To 10
        value(i) = sim1.readings(i)
        Temp = (value(i) - mean) ^ 2
        Temp2 = Temp2 + Temp
    Next i
    Temp2 = Temp2 / 10
    NFStdDev = Round(Sqr(Temp2), 2)

End Function
```

```
Function SFStdDev(mean As Double) As Double
    Dim value(20) As Double
    Dim Temp As Double
    Dim Temp2 As Double

    SFStdDev = 0
    For y = 11 To 20
        value(i) = 0
    Next y

    Temp = 0

    For y = 11 To 20
        value(y) = sim1.readings(y)
        Temp = (value(y) - mean) ^ 2
        Temp2 = Temp2 + Temp
    Next y
    Temp2 = Temp2 / 10
    SFStdDev = Round(Sqr(Temp2), 2)
```

The above code is the function for Standard Deviation for both the North and South results. The code here is rather straightforward mathematical formula that I needed to break down so that the program could use it with the results from the simulation. Due to this, I encountered no errors. I used variables called Temp and Temp2 to store the results at each step of the FOR NEXT loop. This means that I can break each step down and if an error occurs, I can figure out which section is the problematic part of the code with more ease. To test this data, I made test data to see if the standard deviation was being worked out correctly.

| Site | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|
| North Facing | 55 | 40 | 15 | 30 | 0 | 25 | 60 | 90 | 0 | 35 | 35 |
| South Facing | 50 | 90 | 25 | 45 | 65 | 5 | 95 | 55 | 55 | 90 | 57.5 |

The results of your fieldwork have been documented and the mean for each site has been calculated for y
Complete the Standard Error/ Confidence Limits test below by filling in the boxes. If, at any point, you get s
on the little box with a question mark on it at the top right hand corner of the page!

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 35 | 57.5 |
| Standard Deviation (s) | | |

Check

I've copied the mean of both sites into the stats box and they both came out as green. This means they were correct and the program works as it should so far.

For the Standard Deviation, the value for North Facing should be: 26.55183609 but, as it gets rounded to two decimal places, the correct value should actually be 26.55.

The formula for standard deviation is:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Basically, you have to find the sum of all the value subtracted by the mean and then square it. You then have to divide by the total number of values you have. After that, you have to square root the value.

1. (55-35)2 = 400
2. (40-35)2 = 25
3. (15-35)2 = 400
4. (30-35)2 = 25
5. (0-35)2 = 1225
6. (25-35)2 = 100
7. (60-35)2 = 625
8. (90-35)2 = 3025
9. (0-35)2 = 1225
10. (35-35)2 = 0

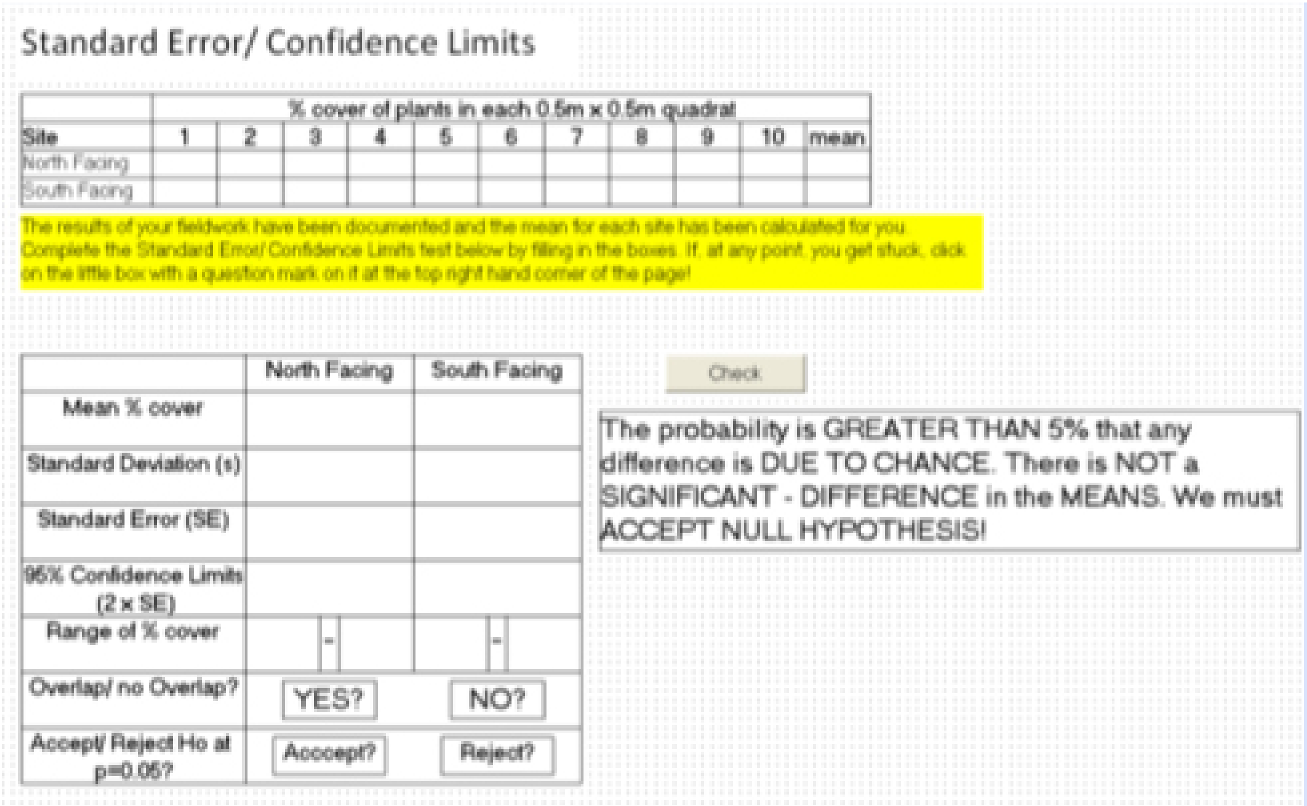
∑ = 7050
7325 / 10 = 705
√705 = 26.55183609

The same calculation will be done for South Facing but using the Bottom of Hill values instead. Doing the same equation will result in the Standard deviation being equal to 27.59075932. This will then be rounded to two decimal places, therefore meaning the correct answer will be 27.59.

This is a very tedious calculation to have to complete over and over again. Hence every biology student must have a calculator and know how to calculate Standard Deviation on it. I worked it out to show how the answer I got was from calculating the Standard Deviation and what it was that the computer was doing.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 35 | 57.5 |
| Standard Deviation (s) | 26.55 | i am not a number |

I entered the value I had worked out for North Facing and it turned green, meaning it was the correct answer and that there was therefore no fault with the program thus far. For South Facing, I entered "i am not a number" and the box remained red. This means the value was recognised as false and therefore the box remained red.



## Standard Error/ Confidence Limits

| | % cover of plants in each 0.5m x 0.5m quadrat | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Site | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | mean |
| North Facing | | | | | | | | | | | |
| South Facing | | | | | | | | | | | |

The results of your fieldwork have been documented and the mean for each site has been calculated for you. Complete the Standard Error/ Confidence Limits test below by filling in the boxes. If, at any point, you get stuck, click on the little box with a question mark on it at the top right hand corner of the page!

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | | |
| Standard Deviation (s) | | |
| Standard Error (SE) | | |
| 95% Confidence Limits (2 x SE) | | |
| Range of % cover | - | - |
| Overlap/ no Overlap? | YES? | NO? |
| Accept/ Reject Ho at p=0.05? | Accept? | Reject? |

Check

The probability is GREATER THAN 5% that any difference is DUE TO CHANCE. There is NOT a SIGNIFICANT - DIFFERENCE in the MEANS. We must ACCEPT NULL HYPOTHESIS!

Here is the form layout I have used for this Stats test. The "Check" button, like last time, checks each box and makes sure that it has the correct data in it. This is where there was a lot of fun and games and where the all my errors occurred.

## Error: Computing Error

```
Private Sub Command1_Click()
If Val(txt_nfm.Text) = Val(lbl_nfmean) Then
    txt_nfm.BackColor = &HFF00&
    txt_nfm.Locked = True
    If Val(txt_nfs.Text) = NFStandardDeviation Then
        txt_nfs.BackColor = &HFF00&
        txt_nfs.Locked = True
        If Val(txt_nfse.Text) = NFStandardError Then
            txt_nfse.BackColor = &HFF00&
            txt_nfse.Locked = True
            If Val(txt_nfcl.Text) = (2 * NFStandardError) Then
                txt_nfcl.BackColor = &HFF00&
                txt_nfcl.Locked = True
                If Val(txt_nfr1.Text) = Val(range1) Then
                    If Val(txt_nfr2.Text) = range2 Then
                        txt_nfr1.BackColor = &HFF00&
                        txt_nfr2.BackColor = &HFF00&
                        txt_nfr1.Locked = True
                        txt_nfr2.Locked = True
                        correct = correct + 1
                    Else
                        txt_nfr1.BackColor = &HFF&
                        txt_nfr2.BackColor = &HFF&
                    End If
                Else
                    txt_nfr1.BackColor = &HFF&
                    txt_nfr2.BackColor = &HFF&
                End If

            Else
                txt_nfcl.BackColor = &HFF&
            End If
        Else
            txt_nfse.BackColor = &HFF&
        End If
    Else
        txt_nfs.BackColor = &HFF&
    End If
Else
    txt_nfm.BackColor = &HFF&
End If
```

This is the first part of the check and checks everything up to the ranges for the North Facing part of the results. The above code is the final version of the code and actually works as it should do.

The big problem I had with this code was the line "IF Val(txt_nfr2.Text) = Val(range1) Then" and ending with "correct = correct + 1". Although this code now works, I had a lot of trouble with it previously.

```
If Val(txt_nfr1.Text) = lbl_nfmean - 2 * NFStandardError Then
    If Val(txt_nfr2.Text) = lbl_nfmean + 2 * NFStandardError Then
        txt_nfr1.BackColor = &HFF00&
        txt_nfr2.BackColor = &HFF00&
        txt_nfr1.Locked = True
        txt_nfr2.Locked = True
        correct = correct + 1
```

The above code was my first attempt. You can see instantly that it wouldn't work because I hadn't used brackets to clearly distinguish between which variables are being affected.

| | North Facing | | | South Facing | | |
|---|---|---|---|---|---|---|
| Mean % cover | 5.5 | | | 15.5 | | |
| Standard Deviation (s) | 2.87 | | | 2.87 | | |
| Standard Error (SE) | 0.91 | | | 0.91 | | |
| 95% Confidence Limits (2 x SE) | 1.82 | | | 1.82 | | |
| Range of % cover | 3.68 | - | 7.32 | 13.68 | - | 17.32 |
| Overlap/ no Overlap? | NO! | | | | | |
| Accept/ Reject Ho at p=0.05? | Reject! | | | | | |

As you can see, the ranged for North Facing are deemed incorrect, although they most definitely are.

5.5-1.82 = 3.68
5.5+1.82 = 7.32

However, the priority of mathematical functions (BIDMAS) meant the answer was not correct.

The answer is seemed clear so I added brackets to make sure that the mathematical operators performed in the correct order and to the correct variables.

```
txt_nfc1.Locked = True
If Val(txt_nfr1.Text) = lbl_nfmean - (2 * NFStandardError) Then
    If txt_nfr1.Text = "3.68" ext) = lbl_nfmean + (2 * NFStandardError) Then
        txt_nfr1.BackColor = &HFF00&
```

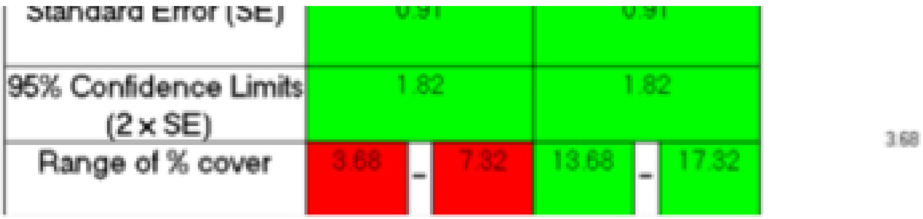
```
txt_nfc1.Locked = True
If Val(txt_nfr1.Text) = lbl_nfmean - (2 * NFStandardError) Then
    If Val(txt_nfr2.Tex lbl_nfmean = "5.5" ean + (2 * NFStandardError) The
```

```
txt_nfc1.Locked = True
If Val(txt_nfr1.Text) = lbl_nfmean - (2 * NFStandardError) Then
    If Val(txt_nfr2.Text) = (2 * NFStandardError) = 1.82 FStandardError) Th
        txt_nfr1.BackColor = &HFF00&
```

The above 3 screen shot is my program running in debug mode. This mode allows me to see each process being executed by the computer. It also can tell me the value assigned to a variable if I hover over it with my mouse. I wanted to check if the error was coming from calculating the maths. As you can see above, the inputted data is 3.68. The mean is 5.5 and the number being added to either side of the mean is 1.82. Due to this, there should be no reason why the box is turning red and not green.

```
test.Caption = lbl_nfmean - (2 * NFStandardError)

If lbl_nfmean - (2 * NFStandardError) = 3.68 Then
    test.BackColor = &HFF00&
End If
```

This is a testing code I wrote that will turn a simple box green if the result is equal to 3.68. I'm using predetermined test data for during this so I know the value it will produce already.




The box I created white in background so it cannot be seen but the number within it can. As you can see, the number produced from the equation was indeed 3.68 but the colour did not turn green. At this point, I was very confused. I tried one more thing which was to assign each of the equation as its own variable and then call the variable into the IF statement when the Check button has been pressed.

```
range1 = (Val(lbl_nfmean) - (2 * NFStandardError))
range2 = (Val(lbl_nfmean) + (2 * NFStandardError))
range3 = (Val(lbl_sfmean) - (2 * SFStandardError))
range4 = (Val(lbl_sfmean) + (2 * SFStandardError))
```

These are the variables that I've created and the equations assigned to them.

```
If Val(txt_nfr1.Text) = Val(range1) Then
    If Val(txt_nfr2.Text) = Val(range2) Then
        txt_nfr1.BackColor = &HFF00&
        txt_nfr2.BackColor = &HFF00&
        txt_nfr1.Locked = True
        txt_nfr2.Locked = True
        correct = correct + 1
```

This means my code now looks like this, with the newly assigned variables being used in the comparison instead of the equation itself.

| | North Facing | | | South Facing | | |
|---|---|---|---|---|---|---|
| Mean % cover | 5.5 | | | 15.5 | | |
| Standard Deviation (s) | 2.87 | | | 2.87 | | |
| Standard Error (SE) | 0.91 | | | 0.91 | | |
| 95% Confidence Limits (2 x SE) | 1.82 | | | 1.82 | | |
| Range of % cover | 3.68 | - | 7.32 | 13.68 | - | 17.32 |

This now shows that my program works. It appeared the problem was that the computer didn't like the fact that there was an equation as part of the comparison in the IF statement.

### Determining whether there's an overlap:

The next problem I encountered was in trying to get the range working. This particular statistical test works out how far the standard error is from the mean. You then need to see if there is an overlap in the ranges between the particular results. The problem I had was; trying to get the program to recognise when there was an overlap in the values.

```
If Val(txt_nfr1.Text) < Val(txt_sfr2.Text) And Val(txt_nfr2.Text) > Val(txt_sfr1.Text) Then
    overlap = True
Else
    overlap = False
End If
If Val(txt_nfr2.Text) < Val(txt_sfr1.Text) And Val(txt_nfr2.Text) > Val(txt_sfr1.Text) Then
    overlap = True
Else
    overlap = False
End If

If correct <> 4 Then
    If overlap = True And Label131.BackColor = &HFFFF& Then
        Label131.BackColor = &HFF00&
        Label131.Caption = "YES!"
        Label131.Top = 8760
        Label131.Left = 4800
        Label132.Visible = False
        correct = correct + 1
    ElseIf overlap = False And Label132.BackColor = &HFFFF& Then
        Label132.BackColor = &HFF00&
        Label132.Caption = "NO!"
        Label132.Top = 8760
        Label132.Left = 4800
        Label131.Visible = False
        correct = correct + 1
    Else
        Label132.BackColor = &HFF&
        Label131.BackColor = &HFF&
    End If
```

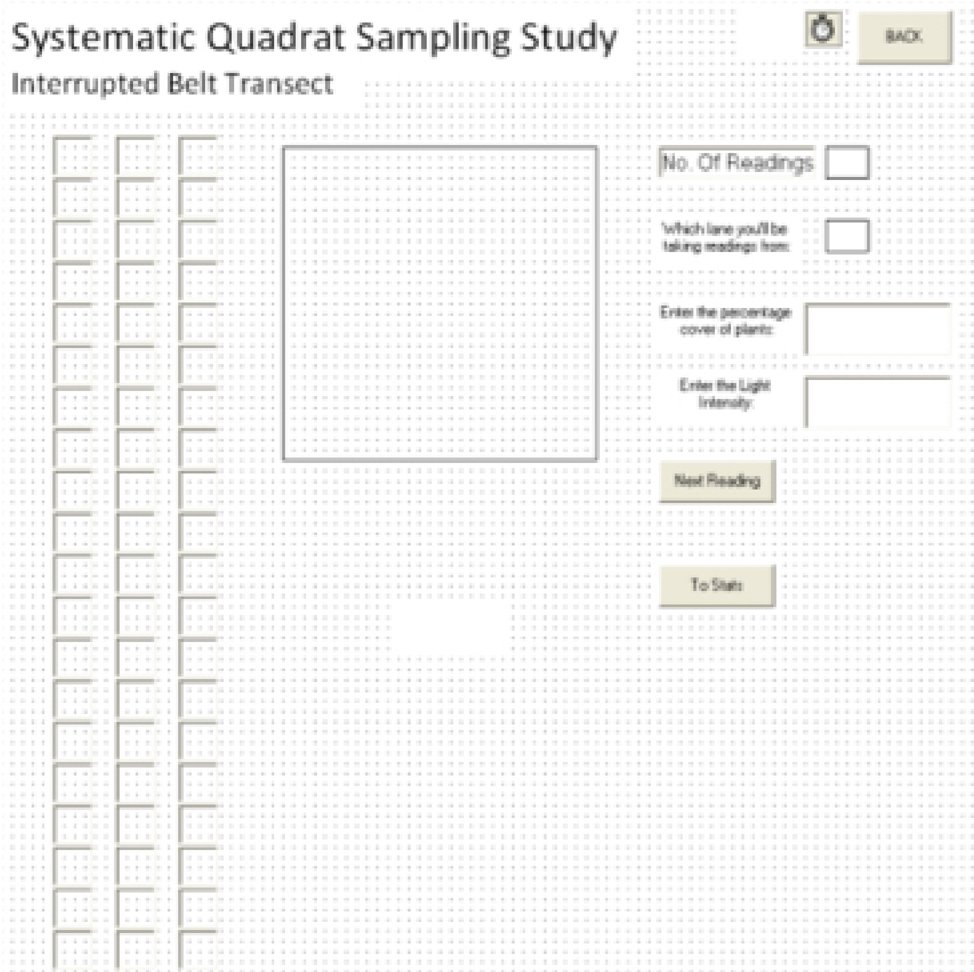
```
Private Sub Label131_Click()
If Yclick = False Then
    Label131.BackColor = &HFFFF&
    Label132.BackColor = &HFFFFFF
    Yclick = True
    Nclick = False
End If
End Sub
```

```
Private Sub Label132_Click()
If Nclick = False Then
    Label132.BackColor = &HFFFF&
    Label131.BackColor = &HFFFFFF
    Nclick = True
    Yclick = False
End If
End Sub
```

The above code is the final version, with the click events of the YES and NO labels. Originally, I had the IF statement separated into two different IF statements however, it always passed by as if it didn't fulfil the requirements. I messed around a lot, trying to get it to work, using different variations in language and structure. After I put the code onto one line, separated by an "and," the code worked. After the comparisons, all that was left was to put the two pieces of code together and ask whether the user has selected the right label based on their results.

## Simulation 3:

### Creating the Form:




This is a screen shot of my form so far for simulation 3. As you can see, it's very different to the other two simulations. Firstly, there are three lanes of boxes that will all load an image in them. The idea behind this simulation is that you're finding out how the percentage cover of the plant changes the further away from a site you. You also test another factor that you measure to see if that is the reason for any possible change. In this study, you measure Light Intensity for every reading you take for the plant. This is results in the two text boxes on the left hand side. Because of the systematic approach to this, I don't need to generate random numbers each time.

### Working out which readings to take:

The difficulty with this simulation from the others is that you need to take a reading at every other square. For example, you would begin at the second box from the bottom and then you'd take your next reading from the one after the next one. To make sure the correct boxes have been selected, I'll need to implement an equation that works out the sequence required for selecting the correct box.

You take 10 readings: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
The 10 boxes you selected are: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19.

This means the formula I need to use will be 2N+1; where "N" will be the value of the counter

The second lane begins with box index 21. This means the formula I need to use for this lane will be 2N+21. Consequently, the third lane, which begins with box index of 41 which means the equation for that line will be 2N+41.

```
Private Sub box_Click(Index As Integer)

If recorder < 10 Then
    If boxcheck(Index) = False Then
        Select Case lane
        Case 1
            selection = (2 * recorder) + 1
            If Index = selection Then
                SelectedImage.Picture = box(Index).Picture
                SelectedImage.Visible = True
                box(Index).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\grey.gif")
                boxcheck(Index) = True
            End If
        Case 2
            selection = (2 * recorder) + 21
            If Index = selection Then
                SelectedImage.Picture = box(Index).Picture
                SelectedImage.Visible = True
                box(Index).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\grey.gif")
                boxcheck(Index) = True
            End If
        Case 3
        selection = (2 * recorder) + 41
            If Index = selection Then
                SelectedImage.Picture = box(Index).Picture
                SelectedImage.Visible = True
                box(Index).Picture = LoadPicture(App.Path & "\images" & "\simulation1" & "\grey.gif")
                boxcheck(Index) = True
            End If
        End Select
    End If
End If
```

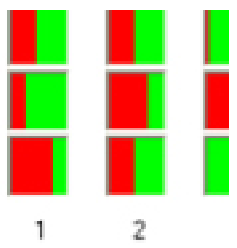
This is the code on the click event on the button. I first check if the recorder is less than 10. This variable is the one that counts how many readings the user has taken. Like the other simulations, boxcheck is the array I have declared that finds out if a box has been selected or not. If the box the user has selected has not been selected before, a SELECT CASE will initiate, with the variable "lane" being used to work it.

```
'random number to determine the lane the user takes the readings from
Randomize
lbl_lane.Caption = Int((3) * Rnd + 1)
lane = lbl_lane.Caption
```

This is the code for the lane variable. It's situated in the FORM LOAD event. I only want one lane to be used so there is no need to keep generating a new number. From this, the values that the variable can take can only be 1, 2 and 3. Due to this, my SELCECT CASE is between 1 and 3 as well.

In the Case for each value, it begins with a variable called "selection. This is the variable that holds the value that will be compared with the index of the selected box. If the two values are the same, it means the user has selected the correct box. This then results in the big image box to display the image for the user to take the readings from.


Lane 1 has been chosen so the big image box should display the second image box from the bottom.




As you can see, the large image box displayed the correct image and it also greyed out the image that the user had selected. Now that I know that this works, I can get the Light Intensity to work and then begin on the saving of the results.

**Generating the Light Intensity:**

Instead of having two different readings for the number of plants at two different sites, this simulation requires the user to record the Light Intensity of each reading. As the light intensity is random and a lot can affect it, I'll need to make the light intensity value change to appear random. To do this, I'll use a timer that loops every 10 seconds, generating another number at each pass. I chose 10 seconds because I want enough time for the user to input their data.

```
'generating intial light intensity value at start of simulation
Randomize
LERandomNumber = Int(((14 - 9 + 1) * Rnd + 9) * 1000)
lbl_LightIntensity.Caption = LERandomNumber


End Sub
```

```
Private Sub tmr_LightIntensity_Timer()
'Event that generates random light intensity value every 10 seconds.
If txt_LightIntensity.Text = "" Then

    Randomize
    LERandomNumber = Int(((14 - 9 + 1) * Rnd + 9) * 1000)
    lbl_LightIntensity.Caption = LERandomNumber
End If

End Sub
```

This is the code I have used to generate the random light intensity value and because I have the code within a timer, it changes every 10 seconds. I also included the same code in the FORM LOAD event because, although I see it unlikely for the user to take a reading and write the light intensity down in 10 seconds, I don't think it looks very good if there's a blank space and then suddenly the number appears. The code for generating the random value every 10 seconds is confined within an IF statement that executes the code within it if the text box that takes the readings is empty. This is to allow the user to type the value of the light intensity without it changing half way through doing so.

9240

10841

The two screen shots above are of the form running and these two values were generated. This means that it works and when I type something into the text box, the value wasn't changing and stayed the same until I clicked off of the box.

**Recording Both Readings:**

Now that I have the simulation and the light intensity working, I can begin on taking the results and adding them into the readings array.

```
If recorder < 10 Then
    If bothReadings = False Then
        If txt_LightIntensity.Text <> "" Then
            If txt_readings.Text <> "" Then
                If txt_readings.Text = lbl_hidden.Caption Then
                    If txt_LightIntensity.Text = lbl_LightIntensity.Caption Then
                        lightIntensity = True
                        Lpointer = Lpointer + 1
                        bothReadings = True
                        Ppointer = Ppointer + 1
                    Else
                        MsgBox "Incorrect Light Intensity Inputted!"
                    End If
                Else
                    MsgBox "Incorrect Percentage Inputted!"
                End If
                If bothReadings = True Then
                    sim1.readings(Ppointer) = txt_readings.Text
                    sim1.readings(Lpointer) = txt_LightIntensity.Text
                    txt_readings.Text = ""
                    txt_LightIntensity = ""
                    recorder = recorder + 1
                    counter.Caption = recorder & "/10"
                    SelectedImage.Visible = False
                    bothReadings = False
                    Text1.Text = Text1 & sim1.readings(Ppointer) & vbNewLine
                    Text2.Text = Text2 & sim1.readings(Lpointer) & vbNewLine
                    Randomize
                    LERandomNumber = Int(((14 - 9 + 1) * Rnd + 9) * 1000)
                    lbl_LightIntensity.Caption = LERandomNumber
                Else
                    bothReadings = False
                End If
            Else
                MsgBox "Incorrect value inputted! Please try again!"
                bothReadings = False
            End If
        Else
            MsgBox "Incorrect value inputted! Please try again!"
            bothReadings = False
        End If
    Else
    End If
End If
```

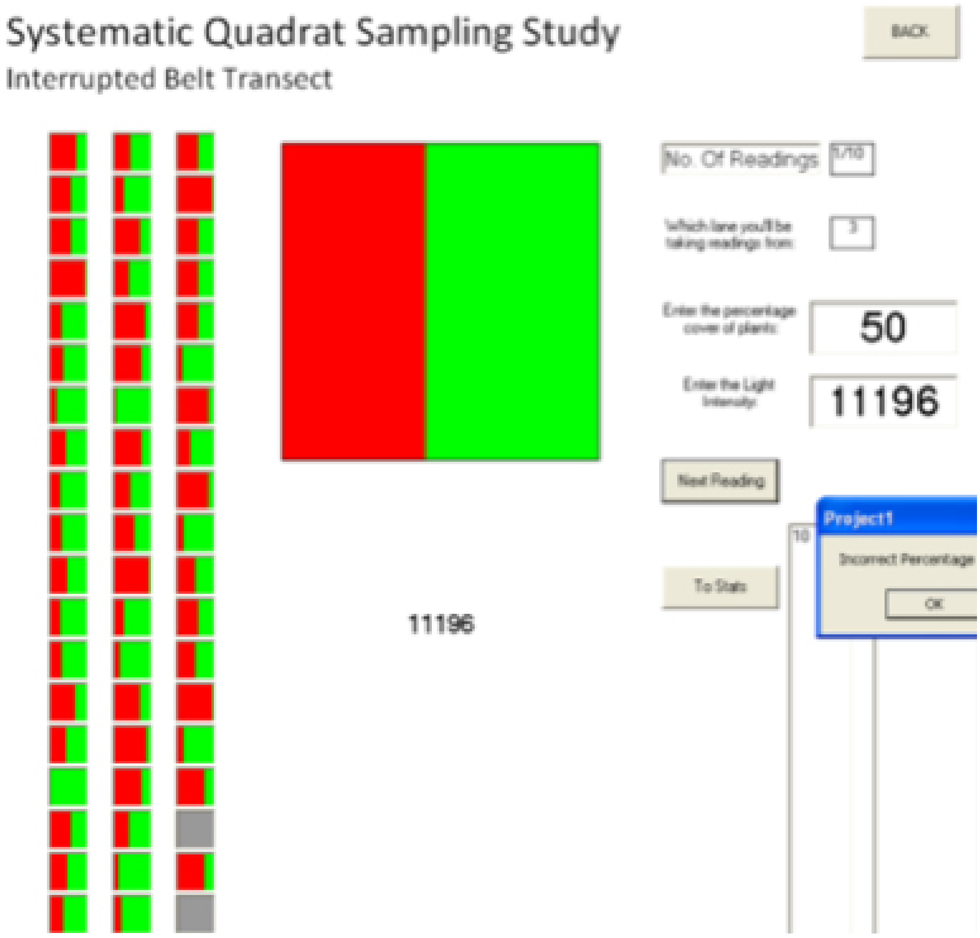
This is the code I have on my button to save the results. Firstly I need to make sure the variable; "recorder" is less than 10. There are only mean to be 10 recordings so it can't go beyond that number. The next 3 lines of code are validation to ensure the IF statement only executes under the right conditions. The first line asks if the variables, "bothReadings" is equal to FALSE. This variable is necessary because I have two individual variables that are being recorded. I need to make sure that they're recorded at the same time. The next two lines of code ask whether the two text boxes, which the user uses to write down the results, are not empty. If they are empty, a message box will pop up, indicating the user has made a mistake and to try again. If they are not empty, a double IF statement will execute. Within this IF statement, two comparisons are made. The first asking whether "txt_readings" has the same data in it as "lbl_hidden." This label is used to record the random number that has been assigned to each box. This therefor gives a solid comparison and if this comparison is correct, the next one will execute. This one asks whether the data in "txt_LightIntensity" is equal to text within "lbl_LightIntensity." If that comparison is correct also, the recording of the results will begin.

Lpointer is the location in the array that the results will be saved in. the L stands for Light so is used for saving the Light Intensity results. Likewise, Ppointer is used for storing the Plant data. Ppointer has been initialised to 0 in the FORM LOAD event and Lpointer has been initialled to 10. This means that, when the first reading is taken, the position of the plant data shall be in postion 1 and the the position of the light intensity data will be in 11. This means that the plant data will end at position 10 and the light intensity at position 20. I also change the variable, "bothReadings" to TRUE.

The next IF statement executes if "bothReadings" is equal to TRUE. If it is, the value of the two sets of data will be recorded at their respective position in terms of Lpointer and Ppointer. It will then clear the text from the text box. This will then activate the timer again for the light intensity value to change. The variable "recorder" will increase by 1. It then changes the text within the label, that shows how many readings you've taken) to be x+1/10. The large image box, "SelectedImage" will change visibility to FALSE. The variable, "bothReadings" will then change to FALSE. This has now effectively reset everything so that a new reading can be taken. Lastly, I've included the randomising of the light intensity value in the IF statement. This is because it may be too easy to select the new image box before the timer has had enough time to complete a cycle.

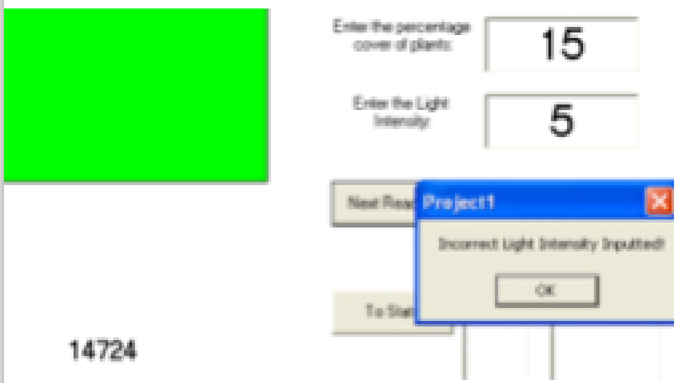
The last two lines of code in the large IF statement are for testing purposes only. I wanted to make sure that the values were being recorder properly. As a result, I added two text boxes that will display the percentage recorded and the light intensity.

**Testing:**

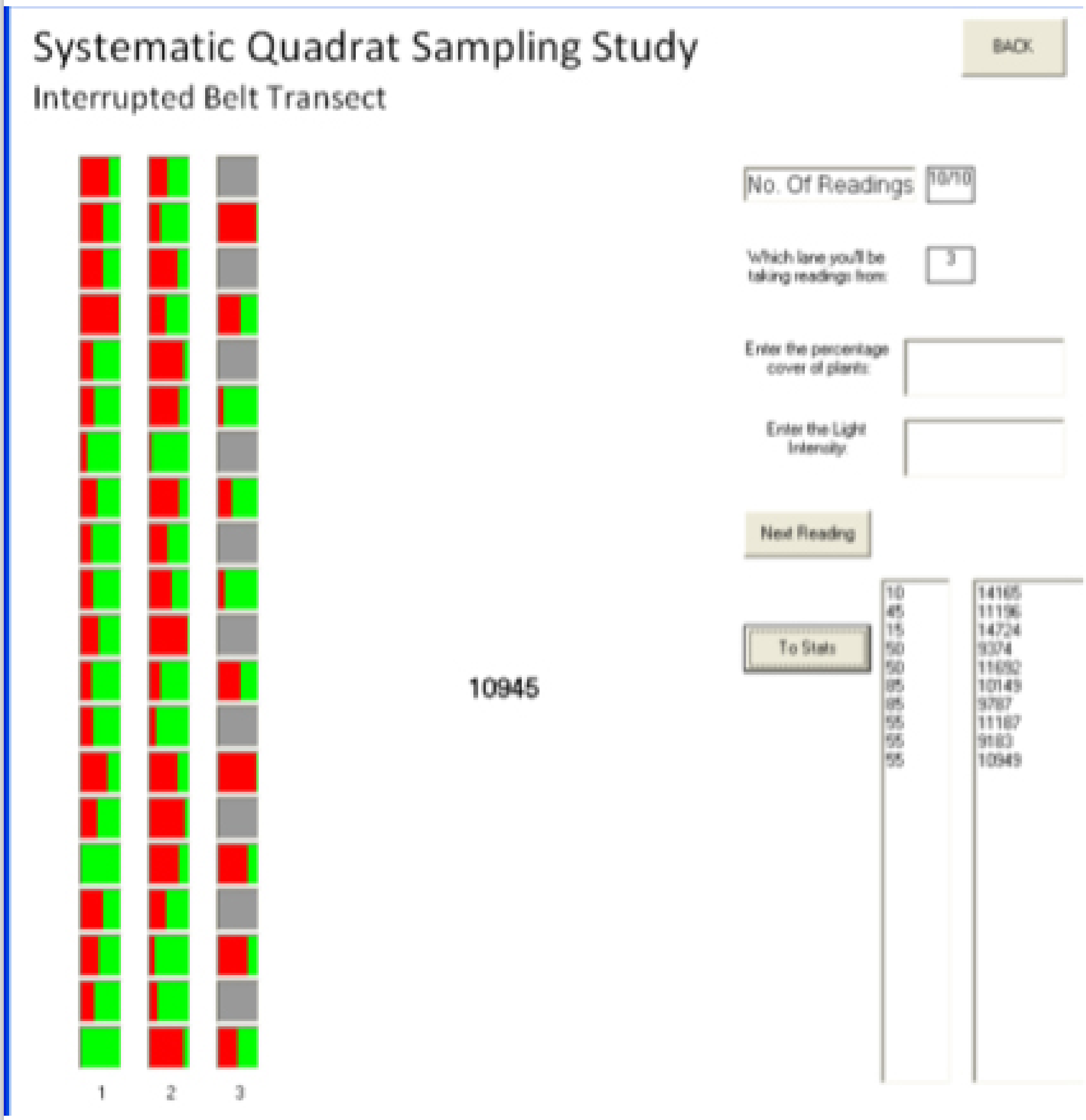


As you can see here, I've taken one reading and am in the process of taking the second, I inputted the percentage of 50 and the program recognised this was incorrect and told me that it was a fault in the percentage reading.


This screen shot shows that the results have now been recorded when the correct percentage was inputted.




Here I was testing to make sure that the program would tell me the correct problem. I entered the wrong light intensity to value and clicked the "Next Reading" button. The program recognised the light intensity was not what it should have been and correctly told me that the light intensity was wrong.




The screen shot above shows the completed simulation. All 10 readings had been taken and consequently, 10 readings for both light intensity and the percentage of plants has been recorded in the two test text boxes. I cannot click any other box and the correct boxes have been greyed out. Also, the randomly generated number to decide which lane you're meant to take your results from works and I could not select a lane other than the third one.

Now all that's left to complete is the statistics for this simulation.

## Spearman's Rank Correlation:

### Sorting the Data and Ranking it:

This statistical test finds out if there is a significant association between the 2 sets of measurements from the same sample. It assesses whether there is a negative or positive correlation and whether this is significant.

I'll need to rank the data in order from lowest to highest, where the lowest value gets the rank of 1. This is then used to find the difference between the two ranks and then you use that to calculate Rs.

```
For i = 1 To 10
    Randomize
    percPlant(i) = Int((20 - 1 + 1) * Rnd + 1)
Next i

'ordering the data into ascending order
For i = 1 To 10
    For j = 2 To 10
        If percPlant(i) > perc(j) Then
            percPlant(i) = percPlant(i) + percPlant(j)
            percPlant(j) = percPlant(i) - percPlant(j)
            percPlant(i) = percPlant(i) - percPlant(j)
        End If
    Next j
Next i

For i = 1 To 10
    Text1.Text = Text1 & percPlant(i) & vbNewLine
Next i
End Sub
```

To problem I am faced with is that I need to allow the user to rank the data, using a text box, and then work out if it's the right rank for the particular number. Because of this, I need a means to compare the ranking given to a set of already ranked data. Therefore I created two new variables called, "percPlant" and "lux." I separated the data in the global array so I was in two smaller arrays. At the moment, I am working with the percPlant variable and trying to order it and display the results in a text box. The method I have used is a bubble sort method.

If, for example, the first value of perPlant was 5 and the second was 2, I would need to swap their positions in the array. The formula I used above is:

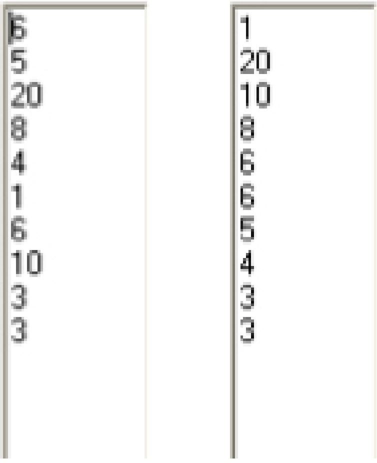
A(i) = A(i) + A(j)
A(j) = A(i) – A(j)
A(i) = A(i) – A(j)

If we follow this through, substituting the values in place of the letter:

A(i) = 5+2 = 7
A(j) = 7-2 = 5
A(i) = 7-5 = 2

As you can see, the positions changed in the variable.

**Error:**



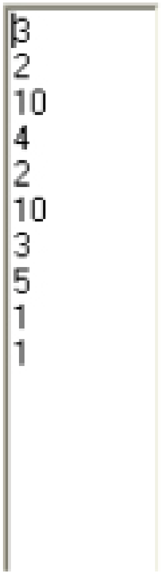

The numbers on the left are the unsorted values and the numbers on the right were meant to be the sorted values. However, it did not work.

I wanted to focus only on the sorting method so I created a public sub routine specifically for the sorting algorithm.

```
Public Sub Sorting(ByRef Values As Variant)
    Dim varSwap As Variant
    Dim swapped As Boolean
    Do
        swapped = False
        For i = 1 To 10
            If Values(i) > Values(i + 1) Then
                varSwap = Values(i)
                Values(i) = Values(i + 1)
                Values(i + 1) = varSwap
                swapped = True
            End If
        Next
    Loop Until Not swapped
End Sub
```

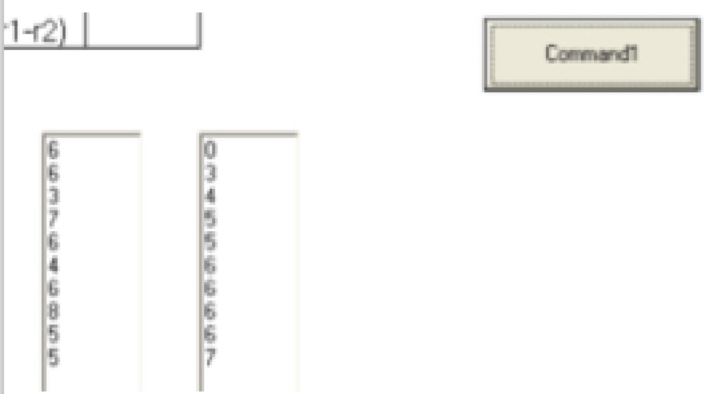
This is the algorithm I used for sorting the values into ascending order. I have a call function on a button event, with the handle "percPlant." I used "Values" in place for percPlant so that I knew which variable I was using.

This is still the bubble sort but in a different format. I have another variable called "varSwap" which temporarily holds a value. If we use the same values as before, 5 and 2, varSwap will equal 5. Values(1) will then be equal to 2 and then values(2) will then be equal to 5. It will keep looping until the Boolean variable, "swapped" does not equal true anymore.




These are the random numbers that I generated to test with.

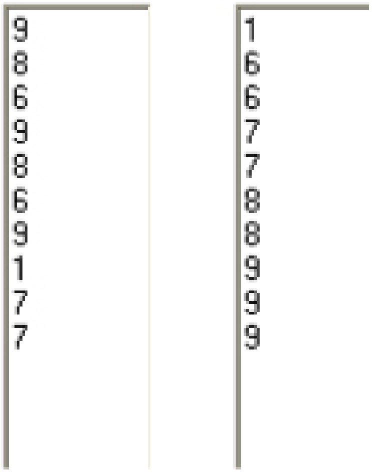
When I click on the button I created, another text box should be filled with the same numbers but sorted in ascending order. The order the numbers should be is: 3,4,5,5,6,6,6,6,7,8




For some reason, the program has missed off the 8 and replaced the number with a 0 at the beginning. The only reason for this, that I can fathom, is that the program tried to access a value in a position that does not have a value in the array percPlant. This would result in the value retrieved being 0.

```
For i = 1 To 10
    If Values(i) > Values(i + 1) Then
        varSwap = Values(i)
```

As I had thought, I have the FOR NEXT loop going up to 10 but the comparison of the second value going up to 100 (i+1) = (10+1). This would then result in the 0 being moved all the way down to the first rank and knocking the 8 out of the ranks. I can simply fix this with changing the FOR NEXT loop's upper limit to 9 instead of being 10.




As you can see, the list of numbers is now sorted in ascending order.

**Comparing Ranks:**

The difficult bit of this code, I believe, is comparing the sorted values to 10 different text boxes that each have their ranks in them and working out if it is correct or not.

| Distance | % | rank1 | lux | rank2 | d (r1-r2) | d^2 |
|---|---|---|---|---|---|---|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |
| 7 | | | | | | |
| 8 | | | | | | |
| 9 | | | | | | |
| 10 | | | | | | |

There are two stacks of labels and two stacks of text boxes. Rank1 and rank2 being the stacks of text boxes. Each stack is a separate control array. My intention is to use the value in the text box as the pointer for the arrays to retrieve the value. I'll then compare this value with the value of the label next to that text box.

| Distance | % | rank1 | lux | rank2 | d (r1-r2) | d^2 |
|---|---|---|---|---|---|---|
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |
| 7 | | | | | | |
| 8 | | | | | | |
| 9 | | | | | | |
| 10 | | | | | | |

Using the example above, the user has decided that the value of 5 is ranked 1. I would then use the value in that text box (1) as the pointer to retrieve the value from the array percPlant (which has already been sorted). I will then compare this retrieved value with the value in the label. If the two match, the value rank is therefore correct and the colour of the text box will turn green.

Lastly, if there are recurring values, the mean of the ranking is given to them instead. For example, say we had two values of 35. If they were meant to be ranked as 3 and as rank 4, they would instead be given the rank of 3.5. I will need to find out if the value inputted into the text box is an integer value or not. If it isn't, I will change the value of the pointer to equal the integer value of the pointer. This is because, if the value the user wrote was 3.5, the ranks would have been 3 and 4. The integer value of 3.5 is 3. This is one of the rankings we want and as the rank for 4 will be the same as the rank for 3, we need only search for the third value.

```
'comparing the sorted data with the users data
For i = 1 To 10
'if value is an integer or not
pointer = txt_r1(i).Text
    If Int(txt_r1(i).Text) = txt_r1(i).Text Then
        pointer = percPlant(pointer)
    Else
        pointer = Int(pointer)
        pointer = percPlant(pointer)
    End If

    If pointer = lbl_p(i) Then
        txt_r1(i).BackColor = vbGreen
        txt_r1(i).Locked = True
    Else
        txt_r1(i).BackColor = vbRed
    End If
Next i

End Sub
```

Because there are only 10 text boxes, I need a FOR NEXT loop with limits 1 and 10. I then assign the pointer the value of the text box in question. If the integer value of the text box is equal to text box, the pointer will change its value to that of the percentage of plants in the percPlant array at the position of the value of pointer. However, if it is not the same, it means that the value the user inputted is not an integer. The program then changes the value into the integer equivalent of that value. Once that is complete, the pointer is then assigned the value of the percPlant value at that position.

If the new value of pointer is equal to the value of the label it refers to, the colour of the text box will turn green and the lock event will change to TRUE.

| Distance | % | rank1 | lux | ra |
|---|---|---|---|---|
| 1 | 65 | 7.5 | 30 | |
| 2 | 60 | 6 | 0 | |
| 3 | 35 | 1 | 55 | |
| 4 | 75 | 9 | 95 | |
| 5 | 55 | 4.5 | 20 | |
| 6 | 40 | 2 | 50 | |
| 7 | 65 | 7.5 | 15 | |
| 8 | 85 | 10 | 45 | |
| 9 | 55 | 4.5 | 20 | |
| 10 | 50 | 3 | 25 | |

As you can see, the rankings were all seen as correct, like they should have been. The sorted data would result in the sequence; 35, 40, 50, 55, 55, 60, 65, 65, 75, 85. This would make the ranks, 1, 2, 3, 4.5, 4.5, 6, 7.5, 7.5, 9, 10.

As you can see to the left, this is what I entered and the ranks were accepted.

Now I need to do the same for the lux values. The process shall be exactly the same as it was for percPlant.

**Comparing "d" and "d-squared" values:**

The next step in the stats is to calculate d and d2. d is the difference between rank1 and rank 2. You then square the result to remover the negative sign.

```
'checking the d values
For i = 1 To 10
    If txt_r1(i).BackColor = vbGreen And txt_r2(i).BackColor = vbGreen Then
        pointer = txt_r1(i)
        difference = pointer
        pointer = txt_r2(i)
        difference = difference - pointer
        If txt_d(i).Text = difference Then
            txt_d(i).BackColor = vbGreen
            txt_d(i).Locked = True
        Else
            txt_d(i).BackColor = vbRed
        End If
    End If
Next i

'checking the d^2 values
For i = 1 To 10
    If txt_d(i).BackColor = vbGreen Then
        difference = txt_d(i)
        difference = difference ^ 2
        If txt_d2(i) = difference Then
            txt_d2(i).BackColor = vbGreen
            txt_d2(i).Locked = True
        Else
            txt_d2(i).BackColor = vbRed
        End If
    End If
Next i
End Sub
```

To check if the "d" values were correct, I used an IF statement to check if the text box for rank 1 and rank 2 was green. If it was, I set the pointer to the value of rank1. I then declared a variable called "difference". This then stores the value that was just given to the pointer. After that, the pointer takes on the value of rank2. I then worked out the difference between the two ranks by subtracting the value of difference by the value of the pointer and assigning this value to difference.

The next IF statement is comparing the value of difference with the value that the user inputted. If they are equal, the text box will turn green and will lock so it cannot be edited further. If not however, the text box will turn red instead. This is all inside a FOR NEXT loop so once the cycle is completed, it will repeat for the second set of text boxes.

For d2, I used the same sort of layout as I did for d. I check if the text box for the d value is green. If it is, I will assign the variable difference to the value within the text box. I will then square this value. Once squared, I check if the value of difference is equal to the value the user has inputted and it if is, the text box will turn green and lock. If not, the text box will just turn red.

| Distance | % | rank1 | lux | rank2 | d (r1-r2) | d^2 |
|---|---|---|---|---|---|---|
| 1 | 75 | 9.5 | 5 | 2 | 7.5 | 56.25 |
| 2 | 55 | 6.5 | 25 | 4.5 | 2 | 4 |
| 3 | 50 | 5 | 35 | 7 | -2 | 4 |
| 4 | 40 | 4 | 60 | 9 | -5 | 25 |
| 5 | 75 | 9.5 | 5 | 2 | 7.5 | 56.25 |
| 6 | 65 | 8 | 5 | 2 | 6 | 36 |
| 7 | 10 | 2.5 | 25 | 4.5 | -2 | 4 |
| 8 | 55 | 6.5 | 40 | 8 | -1.5 | 2.25 |
| 9 | 0 | 1 | 30 | 6 | -5 | 25 |
| 10 | 10 | 2.5 | 90 | 10 | -7.5 | 56.25 |

As you can see, everything turned green, as it should have done. This means that the program works and I can begin on calculating spearman's rank. This requires the user to calculate the sum of d2 and then implementing this value into the Rs equation.

This is because:

– 2 = 7.5
6.5 – 4.5 = 2
5 – 7 = -2
4 – 9 = -5
9.5 – 2 = 7.5
8 – 2 = 6
2.5 – 4.5 = -2
6.5 – 8 = -1.5
1 – 6 = -5
2.5 – 10 = -7.5

### Calculating Spearman's Rank:

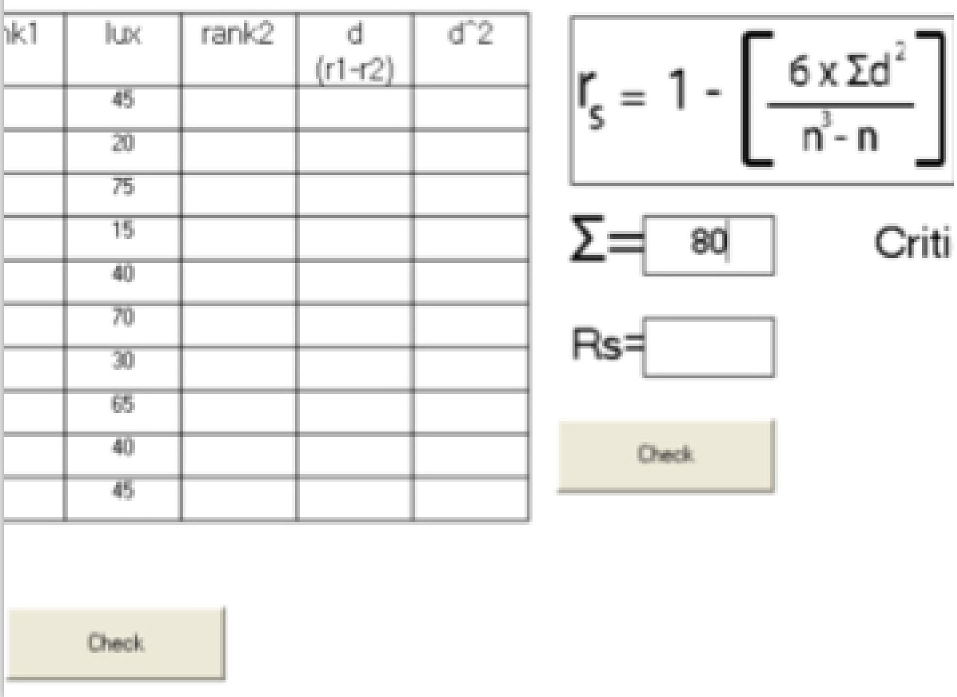
$$r_s = 1 - \left[\frac{6 \times \Sigma d^2}{n^3 - n}\right]$$

This is the equation for calculating the Spearman's Rank Correlation. This value is then compared to a critical value at p=0.05 (5%). To do this, I will create a function where the variable taken shall be the sum of the d2 values. This isn't entirely necessary but it will look neater and make the code more manageable.

```
Function Spearmans(sum As Double)
Dim numerator As Double
Dim denominator As Integer
Dim fraction As Double
numerator = 6 * sum
denominator = (10 ^ 3) - 10
fraction = numerator / denominator
Spearmans = 1 - fraction
End Function
```

This is the function I have written to calculate spearman's rank. I declared 3 variables; one called "numerator," the second called "denominator" and the third one called "fraction." These variables relate to the equation within the bracket.

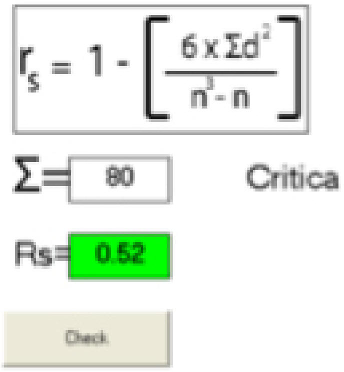
Knowing that my ranking works; it would be unnecessary to keep doing it every time I want to test something. Therefore, I created a button that will take the value in the d2 sum text box and apply it to the spearmans function. The button I used is the one at the very button to the left.



If you work it out using the value I inputted into the box:

6*80 = 480
103-10 = 990
480 / 990 = 0.4848…
1-0.4848 = 0.5151…

However, because I've rounded the value that the function spearmans will return, the value should be 0.52.




As calculated, the value was 0.52 and the box turned green as a result.

### Evaluating Rs and Critical Value:

When you compare the value of Rs and the Critical Value, you ignore any sign that may come before it (-). The sign is only necessary to show you if there is a negative association or a positive association.

```
'evaluation
If txt_rs.BackColor = vbGreen And txt_cv.BackColor = vbGreen Then
    criticalValue = 0.65
    If txt_rs.Text < 0 Then
        criticalValue = 0.65 * 2
        criticalValue = 0.65 - criticalValue
        If txt_rs.Text <= criticalValue Then
            frm_less.Visible = True
        Else
            frm_greater.Visible = True
        End If
    ElseIf txt_rs.Text >= criticalValue Then
        frm_less.Visible = True
    Else
        frm_greater.Visible = True
    End If
End If
```

If the value of RS is less than 0, it means it's negative. To effectively compare the value of Rs and Critical Value, I need to make the value of Critical Value (0.65) negative also. If I double the value and then subtract the original value by this new value, it will become negative. e.g.

0.65 * 2 = 1.3
0.63 – 1.3 = -0.65

If the value of Rs is greater than or equal to the criticl value, then you reject the null hypothesis and claim that there is a significant difference between the two data sets. However, as Rs and the critical value are both negative, I need to ask if the value of Rs is less than or equal to the critical value to achieve the same result. This is because less means it's more negative and therefore, technically, larger if the sign is to be ignored.

For example: if Rs = -0.7 and critical value = -0.65.
-0.7 is less than -0.65
If we ignore the sign, 0.7 is greater than 0.65 but -0.7 is not greater than -0.65.

**Finalising:**

The last thing to do is to make sure that the button cannot be pressed more than once and affect things that shouldn't be affected, not allow blank data when checking and to add the evaluation text box at the end.

```
        If txt_r1(i).Text <> "" Then
            If pointer = lbl_p(i) Then
                txt_r1(i).BackColor = vbGreen
                txt_r1(i).Locked = True
            Else
                txt_r1(i).BackColor = vbRed
            End If
        Else
            MsgBox "Please enter all the ranks for the data."
        End If
    Next i

    'correct
    For i = 1 To 10
        If txt_r1(i).BackColor = vbGreen Then
            correct = correct + 1
        End If
    Next i
```

The above is section of the code that has been edited to make sure no errors can occur from the user. Firstly, I have IF statement that will only execute as long as the text box is not empty. If it's empty, the program will output an error message, asking the user to input data. I have included this in every place where the user needs to input something so it will specify what part is missing.

After that, I have included a new variable called correct. This will count up every time something is correct. The sum total of this variable is 43. At the very end of the button click event, I compare the value of correct to 43. If it's the same, a Boolean variables called "complete" will become TRUE. Otherwise, it will stay false and the program can execute itself once more. One small error I came across when testing was that if you press the check button more than once, the sum of the d2 value will change to red, regardless of whether it was correct or not. This is because I simply counted the all the numbers in the text box without initialising at the end if it was wrong. So, if the value of correct does not equal 43, the Boolean variable will stay false and I will set the variable "sumd" to 0.

Spearman's Rank Correlation

BACK

| Distance along Transect(m) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| % cover of plant | 45 | 55 | 45 | 65 | 90 | 5 | 30 | 75 | 0 | 75 |
| Light Intensity (lux) | 13801 | 13433 | 9471 | 14200 | 13222 | 10167 | 11327 | 10236 | 12408 | 12755 |

The results from the Belt Transect have been recorded and displayed in the results table above. Please complete the Spearman's Rank Correlation test on the data you collected, completing the form out below for each step.

| Distance | % | rank1 | lux | rank2 | d (r1-r2) | d^2 |
|---|---|---|---|---|---|---|
| 1 | 45 | 4.5 | 13801 | 9 | -4.5 | 20.25 |
| 2 | 55 | 6 | 13433 | 8 | -2 | 4 |
| 3 | 45 | 4.5 | 9471 | 1 | 3.5 | 12.25 |
| 4 | 65 | 7 | 14200 | 10 | -3 | 9 |
| 5 | 90 | 10 | 13222 | 7 | 3 | 9 |
| 6 | 5 | 2 | 10167 | 2 | 0 | 0 |
| 7 | 30 | 3 | 11327 | 4 | -1 | 1 |
| 8 | 75 | 8.5 | 10236 | 3 | 5.5 | 30.25 |
| 9 | 0 | 1 | 12408 | 5 | -4 | 16 |
| 10 | 75 | 8.5 | 12755 | 6 | 2.5 | 6.25 |

$$r_s = 1 - \left[\frac{6 \times \Sigma d^2}{n^3 - n}\right]$$

Σ= 108 Critical Value= 0.65

Rs= 0.35

Check

| Pairs of Measurements | Critical Value |
|---|---|
| 5 | 1.00 |
| 6 | 0.89 |
| 7 | 0.79 |
| 8 | 0.74 |
| 9 | 0.68 |
| 10 | 0.65 |
| 12 | 0.59 |
| 14 | 0.54 |
| 16 | 0.51 |
| 18 | 0.48 |

The value of Rs is LESS THAN the critical value.
Therefore: NULL HYPOTHESIS is ACCEPTED! (there is NO SIGNIFICANT CORRELATION between the two variables)
PROBABILITY > 5% that any correlation/ association is DUE TO CHANCE
Therefore: any correlation/ association is NOT SIGNIFICANT an DUE TO CHANCE!

As you can see above, the stats test all works as it should and the correct evaluation come up. 0.35 is less than 0.65 so we accept the null hypothesis. My program is now finished and only a few minor adjustments need to be made.

## Non-Linearity:

Another issue is that you can enter the stats page without having completed the actual simulation first. This is an issue because my end user expressed clearly that he didn't want a program that was linear and forced the user to complete the program in a certain way to achieve the best results. To fix this, I created a global variable of the Boolean data type called "stats." When the program first loads, the stats value is set to FALSE. When any of the stats sections is completed, the variable will be set to "TRUE." So, when you actually enter the stats section, the first thing it does it read the value of the variable. If it is true, it continues as it did before. If the value is FALSE, a message will pop up, telling the user that the simulation has not been completed and they can either enter their own data or generate their own data. When either button is pressed, the value of stats will change to TRUE. This means the buttons can't be pressed again and the stats can continue.

## Chi$^2$ Statistical Test

| | Number of plants in each 0.5m x 0.5m quadrat | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Site** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **total** |
| Top of Hill | | | | | | | | | | | |
| Bottom of Hill | | | | | | | | | | | |

Enter own Readings

Generate Random Readings

The results of your fieldwork have been documented an… d for you. Complete the Chi2 test below by filling in the boxes. If, at any point … question mark on it at the top right hand corner of the page!

You have no readings available to use in the stats. Please either enter your own data and then clicking on "Submit Own Readings" button or clicking "Generate Random Readings" to generate a set of random results.

OK

| **Site** | Observed Number (O) | Expected Number (E) | (O-E) 2 / E |
|---|---|---|---|
| Top of Hill | | | |
| Bottom of Hill | | | |

E = mean of O

Σ=

Degrees of Freedom = N-1

=

| | Number of plants in each 0.5m x 0.5m quadrat | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Site** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **total** |
| Top of Hill | 6 | 5 | 3 | 7 | 5 | 3 | 6 | 8 | 5 | 5 | 53 |
| Bottom of Hill | 2 | 0 | 5 | 0 | 2 | 4 | 1 | 4 | 2 | 2 | 22 |

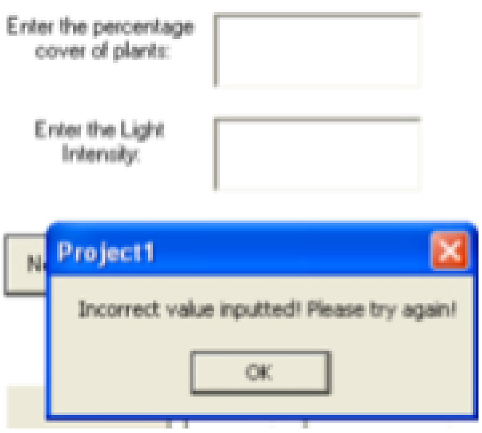
Enter own Readings

Generate Random Readings

# Validation:

Because of the specificity of my program, I already have a lot of validation mechanisms set in place. For example: the user may enter the percentage cover of plants as "hello" but the program will recognise this as incorrect and the user will get it wrong.


The program will not progress unless the correct value has been inputted by the user. The same result will happen if no data is entered. This is necessary in order to prevent empty values being recorded, messing up all the stats tests.




This is applied to every instance where data is inputted into a text box by the user. There cannot be errors for the user to proceed further in the program. For example: during the stats sections of the program, if anything is wrong, it will change the background colour of that textbox to red. If it is correct, than the colour will go bright green. My reasoning for this is because, not only because people relate green for good and red for bad, but the two colours are complimentary colours. This means that, if anything is wrong, the user can spot it very quickly and not have to search for where the problem may lie. Another feature I included is that each box, if left unchanged, will have a separate error message. This means the user knows exactly what they missed out. This is in case the user didn't see a particular text box and missed it out.

For some of the text boxes, however, their value is assigned to a variable. If this variable hasn't been declared as a string data type, there will be a data mismatch error when I run the program. To prevent this problem, I have added an IF statement in front of every block that ends up in the value of the text box being assigned to a variable.

```
For i = 1 To 10
    If IsNumeric(txt_r1(i).Text) = True Then
    'if value is an integer or not
        pointer = txt_r1(i).Text
        If Int(txt_r1(i).Text) = txt_r1(i).Text Then
            pointer = percPlant(pointer)
        Else
            pointer = Int(pointer)
            pointer = percPlant(pointer)
        End If
        If txt_r1(i).Text <> "" Then
            If pointer = lbl_p(i) Then
                txt_r1(i).BackColor = vbGreen
                txt_r1(i).Locked = True
            Else
                txt_r1(i).BackColor = vbRed
            End If
        Else
            MsgBox "Please enter all the ranks for the data", , "Error"
        End If
    Else
        txt_r1(i).BackColor = vbRed
        MsgBox "Only numeric data may be entered into the fields!", , "Error"
        Exit Sub
    End If
Next i
```

As you can see, if the data within the text box is numeric data, the rest of the IF statement will execute. However, if not, then a message box will pop up, telling the user that non numerical data has been entered. The text box in question will also turn red, signalling to the user which box has the wrong data in it. After this, I included an END SUB command. I do not want the program to continue, in spite of the error. This would only mean more error messages would pop up and the user would need to click "OK" multiple times before they can continue. This ultimately means that everything after the incorrect text box will not have been checked; leaving its background colour as white, further helping the user to recognize the incorrect text box.

| Distance | % | rank1 | lux | rank2 | d (r1-r2) | d^2 |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 11 | | | |
| 2 | 2 | h | 12 | | | |
| 3 | 3 | | 13 | | | |
| 4 | 4 | | 14 | | | |
| 5 | 5 | | 15 | | | |
| 6 | 6 | | 16 | | | |
| 7 | 7 | | 17 | | | |

## On Screen Help:

For on screen help, I have accurately captioned labels next to each text box where the user must input something. This is so that the user will know what they must input at each text box they encounter.

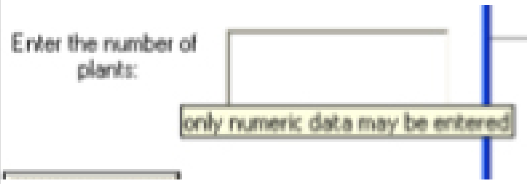
| | North Facing | South Facing |
|---|---|---|
| Mean % cover | | |
| Standard Deviation (s) | | |
| Standard Error (SE) | | |
| 95% Confidence Limits (2 x SE) | | |
| Range of % cover | - | - |
| Overlap/ no Overlap? | YES? | NO? |
| Accept/ Reject Ho at p=0.05? | Accept? | Reject? |

As you can see from above, each text box is categorised under either North Facing or South Facing. From this, each column has a text box beside it that tells the user what sort of data is required for them to input.

**Tool Tips:**

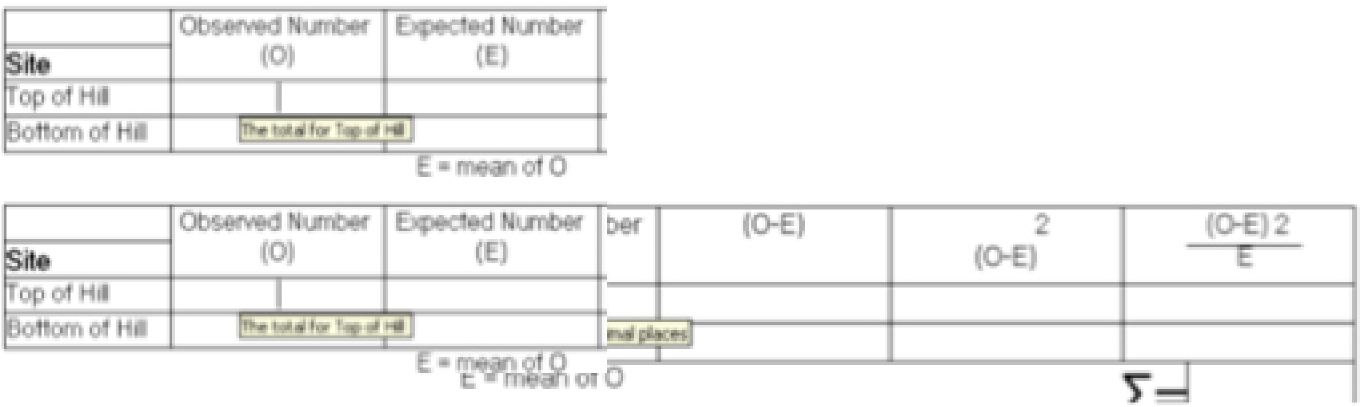
To help my end user, I have included tool tips on text boxes that require an input of data. This is for times when the user may be confused over something that isn't very clear.

For simulations 1, 2 and 3, there is only a text box. The tool tip for these boxes are all the same, "only enter numeric data may be entered"



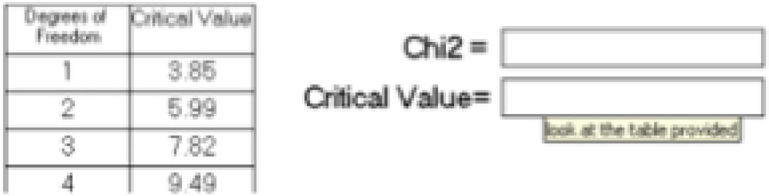

The stats tests are little different as there are a lot of text boxes that the user needs to input data into.

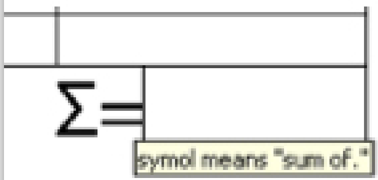
For Chi$^2$ stats test:



Expected Number, (O-E) and (O-E)2/E

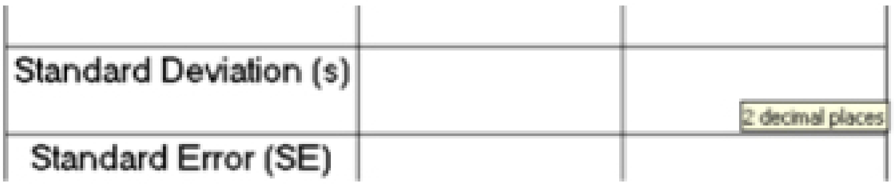
The above text boxes all have the same tool tip as I have provided the formula for them in the caption. The program will only accept values to 2 decimal places so I need to make sure that the user understands that.



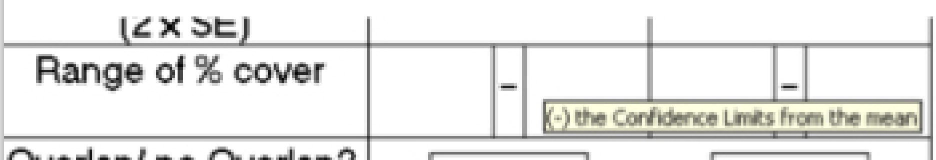



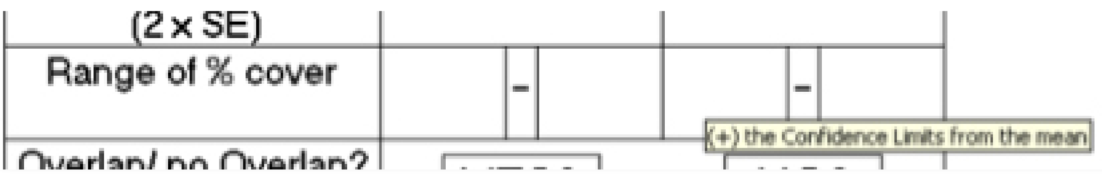


For Standard Error:



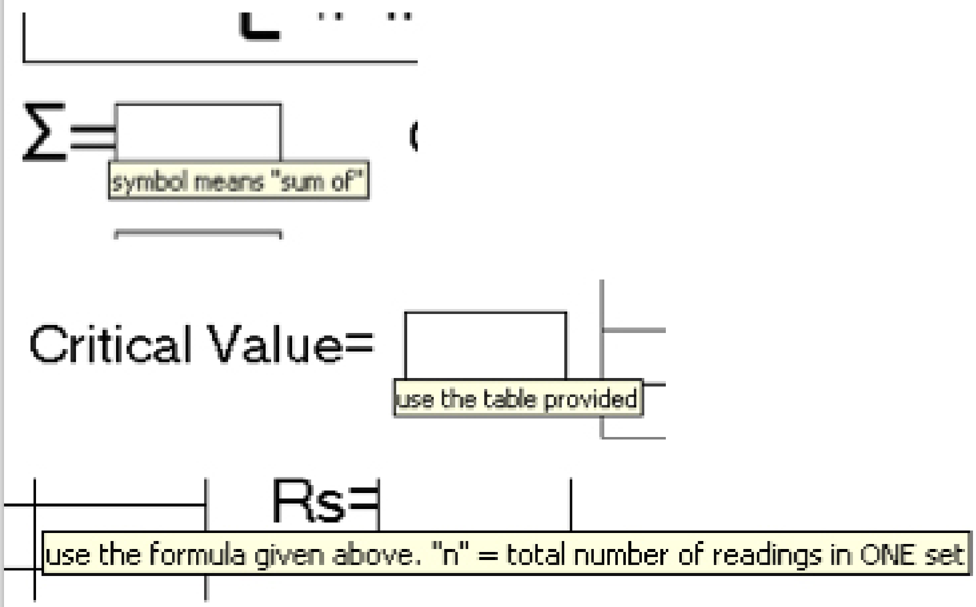


For Spearman's Rank:

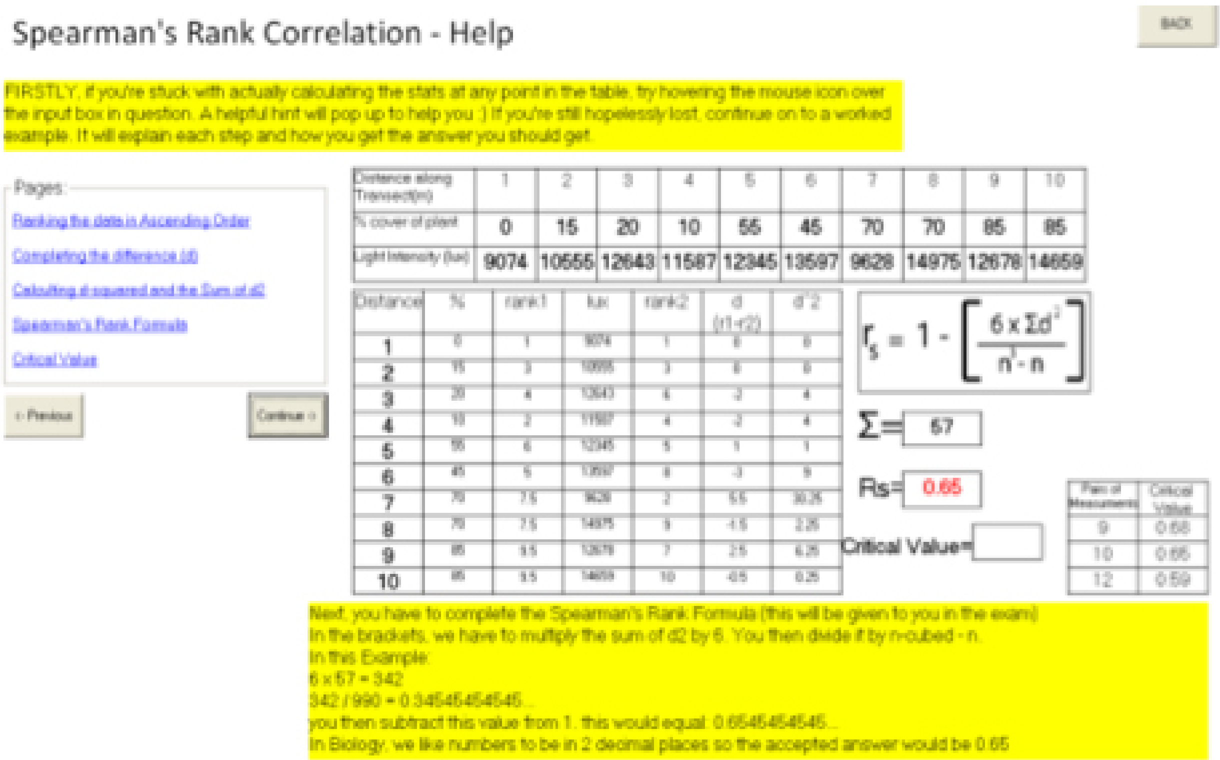
| Distance | % | rank1 | lux | rank2 |
|---|---|---|---|---|
| 1 | 20 | | 11152 | |
| 2 | rank in ascending order where 1 = the smallest number | | | |
| 3 | 25 | | 10425 | |
| 4 | 10 | | 14885 | |
| 5 | 0 | | 10488 | |
| 6 | 85 | | 11842 | |
| 7 | 40 | | 14271 | |
| 8 | 65 | | 14738 | |
| 9 | 25 | | 9344 | |
| 10 | 5 | | 9882 | |

I have used the same tool tip that you see above for all text boxes under the rank1 and rank2 columns. This is because they all require the same input.



**Help Button:**

I have a help button on each page that the user can click on if they are completely stuck. For the Stats test, this will be a walkthrough of the stats test with believable data used.


# Full Alpha Testing:

## Alpha Testing: Plan

### Simulation 1/2:

1. Select a box that is not the one the random numbers have chosen to make sure it doesn't come up as an option
2. Select the correct box – make sure that it appears as it should do
3. Enter string characters into the text box – should not accept answer and error message should display
4. Enter the correct number into the box. The text boxes at the button should display the correct answer in the user box and in the correct box
5. Enter an incorrect number into the box. The user box should display the incorrect box and the correct box should display the correct number.
6. With each pass, the counter should increase by 1 and when it reaches 10, the simulation should refresh itself.
7. Complete the simulation for both sites and the percentage accurate should be calculated correctly.
8. When you enter the stats section through the button, the same data should be taken across also and the correct stats test should appear.

**Chi$^2$ Stats Test:**

1. make sure all data counted up correctly in table of results
2. Click on "Check" button when nothing has been entered to ensure that both O boxes turn red and nothing is accepted
3. Enter string characters into all text boxes (Hx8t). This will check to make sure they aren't accepted.
4. Complete the first set of calculations, clicking on the "check" button each time. This will ensure that only one box turns green at a time and that the following box, which contains the string characters, is still red.
5. Do the same for the second set of readings.
6. For the Chi$^2$ value, enter the value to 3.d.p. should go red as I have set it to 2.d.p.
7. Make sure that correct evaluation box appears based on the results.

**Standard Error:**

1. Make sure all data counted up correctly and mean calculated correctly
2. Click the "check" button with nothing entered to make sure that the boxes turn red and nothing is accepted
3. Enter string characters into ever field (Hk66nE) and click "Check" button once again. The text boxes should not change colour and nothing should have been accepted.
4. For North Facing, complete each step and click "check" button each time. Each box should go green and the one after it should be red as it still contains the string values. Ensures that each step works correctly and that the following step doesn't accept the string characters.
5. Do the same for South Facing to make sure the same results are achieved.
6. Click on the incorrect overlap button (based on results) and consequently, the incorrect hypothesis acceptance button. Both should go red.
7. Click on the correct buttons and the correct evaluation should appear.

**Simulation 3:**

1. Click on the very first box on all three lanes – nothing should happen
2. Click on the second box from the bottom and both incorrect lanes – nothing should happen
3. Click on the second box of the correct lane – the large image box should appear with correct image in it
4. Wait 30 seconds after the light intensity changes – should change 3 times to a different value.
5. Enter string characters (hhYXq) into the light intensity text box
6. Wait another 30 seconds – the changing light intensity value should still be on the value it was when you started
7. Enter the correct percentage and light intensity. The two values should be accepted and will appear in the two text boxes on the form. Displayed in both sets of text boxes.
8. Enter incorrect data into the two text boxes. The data should be accepted but the correct values will be displayed by the right side of boxes and my entered value will be displayed in the left.
9. Counter should be at 2/10.
10. Complete the simulation for all 10 readings. "Next reading" button should disappear and in its place, a label displaying how accurate you were should appear. The percentage should match up to how many I got right and wrong.
11. Results from the simulation should carry on to the stats section.

**Spearman's Rank Correlation:**

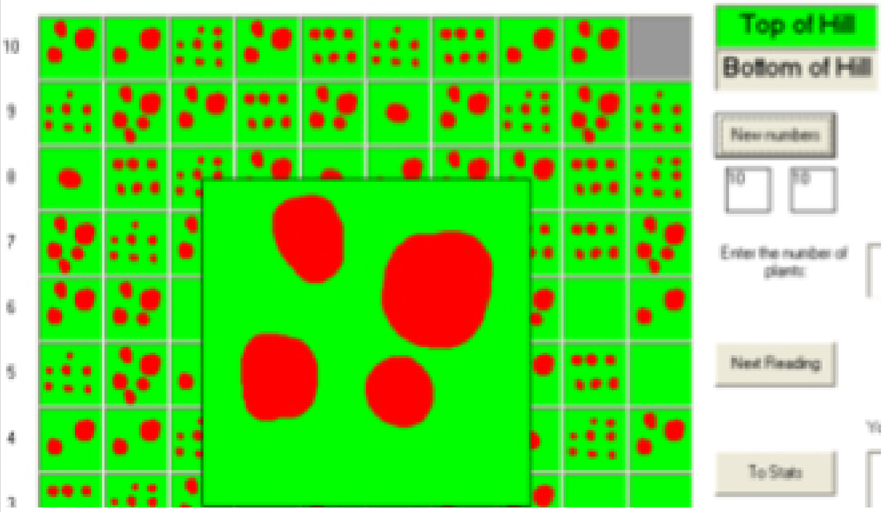
1. Make sure all data is correct in the table of results
2. Click the "check" button without having entered any data to make sure that nothing is accepted and that the boxes turn red.
3. Enter string characters into every field (LL3Vv8) and click "check" once again. Every box should still be red and consequently, no data should have been accepted.
4. Complete the ranks for percentage cover and they should go green
5. Complete the ranks for light intensity and they should go green
6. Enter the correct values for "d" but leave out any negative signs that may appear. The correct boxes should go green and the values that were meant to contain negative values should remain red.
7. Enter the correct "d" for all of the values and then complete the "d2" values accordingly.
8. For the spearman's rank, enter the correct answer but to 3.d.p. the box should go red.
9. Enter the correct spearman's rank to 2.d.p and then enter the correct critical value result. All the boxes should go green and the correct evaluation box will appear.

# Testing:

## Simulation 1:

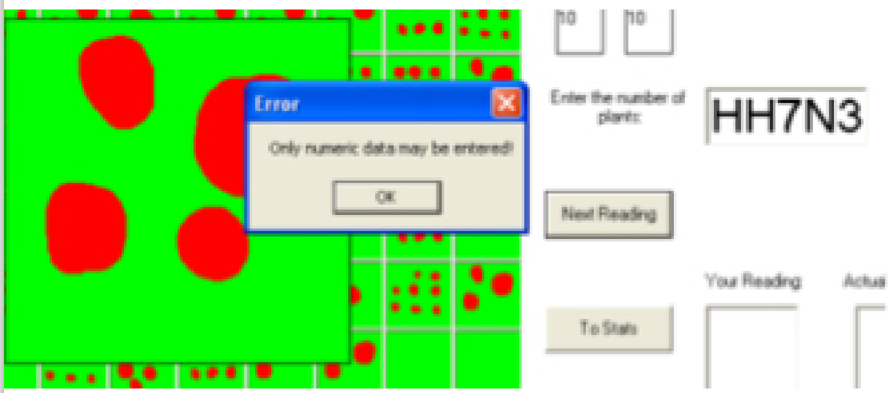
1. Unfortunately, the mouse does not appear in screen shots so I cannot show the results of this test but take my word for it that the box I had incorrectly selected, and any other box, did not show up.
2. To make sure the correct box shows up, I was given the numbers 10, 6 from the random number generator. The number in the simulation at these co-ordinates is 6 plants.


When I click on this box, a much larger box should appear in the centre of the simulation with the same number of plants in it (4).



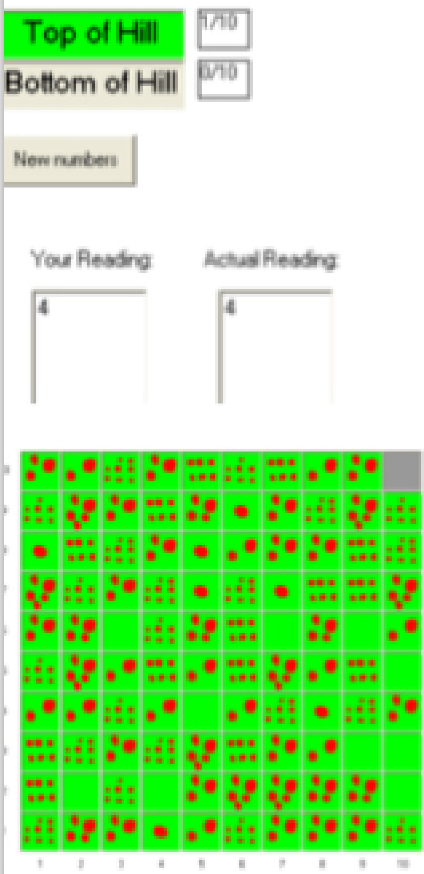

As expected, th correct box appeared once I clicked on the smaller box.

3. The box has 4 plants in it so the accepted number should be 4. I will enter (HH7N3) and the value should be rejected



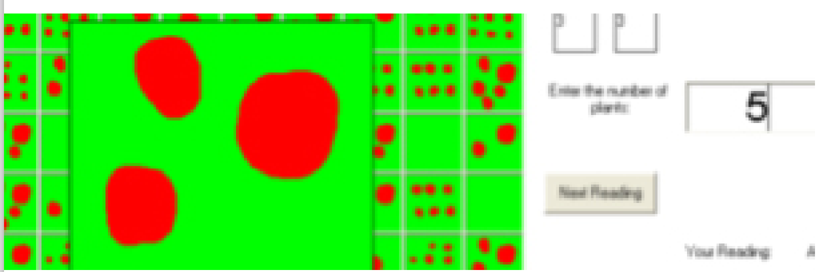

As predicted, the error message came up and the value was not accepted.

4. When I enter thecorrect reading (4), the large image box should disappear, the counter should increase by 1 and there should be a "4" in both text boxes at the bottom of the form.

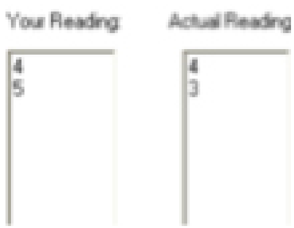



As I had predicted, the counter increased by 1, the value of "4" was displayed in both text boxes and the large image box disappeared.

5. Now If I add an incorrect value, the value should be accepted but the difference is that the second text box, at the bottom of the form, should show the correct value and the first text boxes should display the value that I inputted.




As you can see, the image displays 3 plants and I intend to enter 5 into as the result. The left text box should display a 5 and the right text box should display 3.



As I had expected, the correct reading was displayed in the right text box and my reading was displayed in the left.

6. I have entered only 2 readings so far so I expect the counter to be at 2/10. When he counter gets to 10, the simulation should refresh itself.


As I had expected, the counter was at 2.


As you can see, the counter is at 10 now and the simulation refreshed itself.

7. Once I complete the simulation, the "next reading" button should disappear and be replaced with a percentage of how accurate you were in the simulation, my only mistake was at the beginning so the percentage that is displayed to me should be 95%.

19/20 = 0.95
0.95 * 100 = 95

8. When I enter the stats test, I expect that all the results I recorded should be transferred over correctly. These are the results I collected: 4,5,4,2,0,0,4,6,6,6,7,5,7,8,5,6,0,7,7,2.

| | Number of plants in each 0.5m x 0.5m quadrat | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Site | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | total |
| Top of Hill | 4 | 5 | 4 | 2 | 0 | 0 | 4 | 6 | 6 | 6 | 37 |
| Bottom of Hill | 7 | 5 | 7 | 8 | 5 | 6 | 0 | 7 | 7 | 2 | 54 |

This is the Chi$^2$ test (the correct test) and as you can see, the correct readings were transferred across.

## Chi$^2$:

1. Firstly, I need to make sure that the total is being counted up correctly. If it isn't, the whole stats would be wrong and unreliably.

| | Number of plants in each 0.5m x 0.5m quadrat | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Site | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Top of Hill | 3 | 2 | 0 | 4 | 2 | 1 | 3 | 5 | 2 | 2 |
| Bottom of Hill | 0 | 6 | 2 | 6 | 8 | 2 | 7 | 1 | 8 | 0 |

Using the results above:

3+2+0+4+2+1+3+5+2+2 = 24
0+6+2+6+8+2+7+1+8+0 = 40

Now when I click on "submit own readings," the total should be calculated automatically.

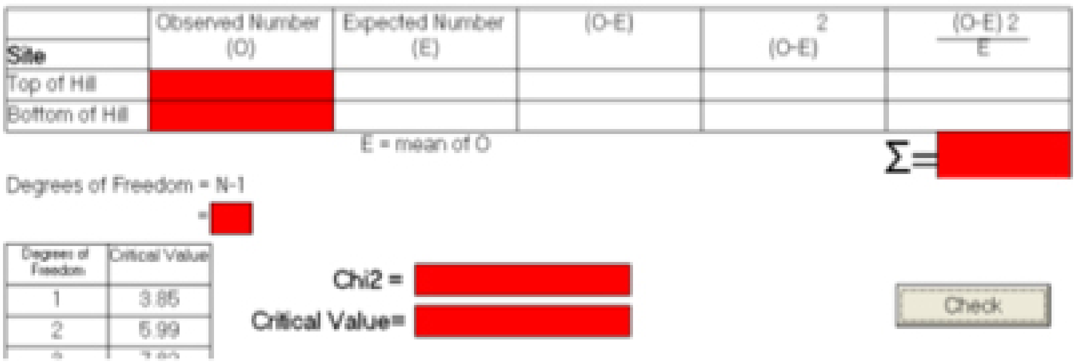
| | Number of plants in each 0.5m x 0.5m quadrat | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Site | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | total |
| Top of Hill | 3 | 2 | 0 | 4 | 2 | 1 | 3 | 5 | 2 | 2 | 24 |
| Bottom of Hill | 0 | 6 | 2 | 6 | 8 | 2 | 7 | 1 | 8 | 0 | 40 |

Enter own Readings

Generate Random Readings

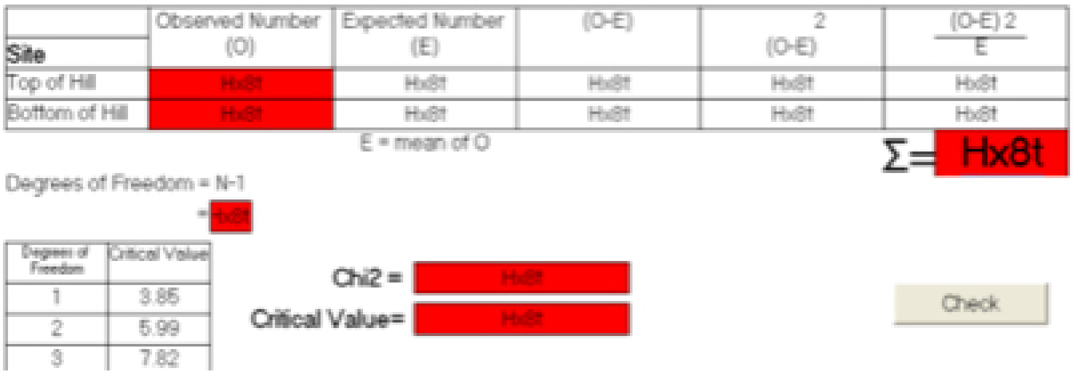
As I had expected, the total had been counted correctly and it did appear automatically.

2. If I click the "check" button now, I expect the stats table to be red in the first two columns and for some of the other text boxes to turn red as well. This would mean that nothing was accepted.



As expected, the boxes turned red. This meant that nothing was accepted that shouldn't have been.

3. When I enter the string characters (Hx8t) into teach field and press the "check" button, I expect nothing to change and that the same text boxes will remain red.




As expected, nothing had changed and the same text boxes remained red.

4. The first box is for the observed number. This is simply the total number calculated at each site. For top of the hill, this should be 24.

| Site | Observed Number (O) | Expected Number (E) | (O-E) |
|---|---|---|---|
| Top of Hill | 24 | Hx8t | Hx8t |

As predicted, the box turned green when the correct answer was inputted. Consequently, the following box turned red as it still contains the string characters I entered earlier.

The Expected Number is the mean of both of the observed numbers. The values for the observed numbers are 24 and 40. Therefore, the mean of those two numbers would be:

24+40 = 64
64 / 2 = 32

This means the value I expect to be accepted should be 32.

| Site | Observed Number (O) | Expected Number (E) | (O-E) | 2 (O-E) |
|---|---|---|---|---|
| Top of Hill | 24 | 32 | Hx8t | Hx8t |

As expected the value of 32 turned the text box green.

The difference is, as the label above the text box shows, the observed number minus the expected number. In this case, the difference should be:

24-32 = -8

| Expected Number (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|
| 32 | -8 | Hx8t | Hx8t |

As expected, the value accepted was -8.

The next step is the square of the difference. This would mean that the answer accepted should be 64

(-8)2=64

| Expected Number (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|
| 32 | -8 | 64 | Hx8t |

As expected, the accepted value was 64.

The last step is to take the value I just got (64) and divide it by the expected number.

64/32 = 2

| Site | Observed Number (O) | Expected Number (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|---|---|
| Top of Hill | 24 | 32 | -8 | 64 | 2 |

As expected, the accepted 2. This shows that the calculations made for Top of Hill work and the correct values were accepted whereas the incorrect ones were not.

5. For Bottom of Hill, the observed number should be 40.

| | | | |
|---|---|---|---|
| Bottom of Hill | 40 | Hx8t | Hx8t |

As expected, the observed number accepted was indeed 40.

The expected number will be exactly the same as the Top of Hill value as they both use the same observed numbers. Therefore, I expect 32 to be the accepted answer.

| Site | Observed Number (O) | Expected Number (E) | (O-E) | $(O-E)^2$ |
|---|---|---|---|---|
| Top of Hill | 24 | 32 | -8 | 64 |
| Bottom of Hill | 40 | 32 | Hx8t | Hx8t |

As expected, 32 was accepted as the correct value.

For the difference, the value I expect to be accepted will be 8.

40-32 = 8

| Site | Observed Number (O) | Expected Number (E) | (O-E) | $(O-E)^2$ |
|---|---|---|---|---|
| Top of Hill | 24 | 32 | -8 | 64 |
| Bottom of Hill | 40 | 32 | 8 | Hx8t |

As expected, 8 was accepted as the correct answer.

For the difference squares, the value I expect to be accepted is 64.

$(8)^2=64$

| Expected Number (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|
| 32 | -8 | 64 | 2 |
| 32 | 8 | 64 | Hx8t |

As expected, the value the program accepted was 64.

Lastly, the same value of 2 should be accepted for the next step. This is because the same value is being inputted; 64.

64/32 = 2

| Observed Number (O) | Expected Number (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|---|
| 24 | 32 | -8 | 64 | 2 |
| 40 | 32 | 8 | 64 | 2 |

As expected, the same value was accepted.

This shows that the table for the stats accepts the values that I have worked out, meaning the program is doing the correct calculations also.

6. Because of the simplicity of the results, I've taken to doing another sample to get results that I can test this on.

| Observed Number (O) | Expected Number (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|---|
| 28 | 38 | -10 | 100 | |
| 48 | 38 | 10 | 100 | |

Using these results, I can see that I will get a decimal number from the last step.

100/38 = 2.6315789754

However, I have set the text box to only accept values to 2.d.p. therefore; the value the text box should accept is 2.63. To test that this is definitely the case, I shall enter 2.631. I expect the value to be rejected.

| Site | Observed Number (O) | Expected Number (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|---|---|
| Top of Hill | 28 | 38 | -10 | 100 | 2.631 |
| Bottom of Hill | 48 | 38 | 10 | 100 | 2.631 |

As expected, the value was rejected. Now, if I enter the value 2.63, the text box should accept the value and turn green.

| Observed Number (O) | Expected Number (E) | (O-E) | $(O-E)^2$ | $\frac{(O-E)^2}{E}$ |
|---|---|---|---|---|
| 28 | 38 | -10 | 100 | 2.63 |
| 48 | 38 | 10 | 100 | 2.63 |

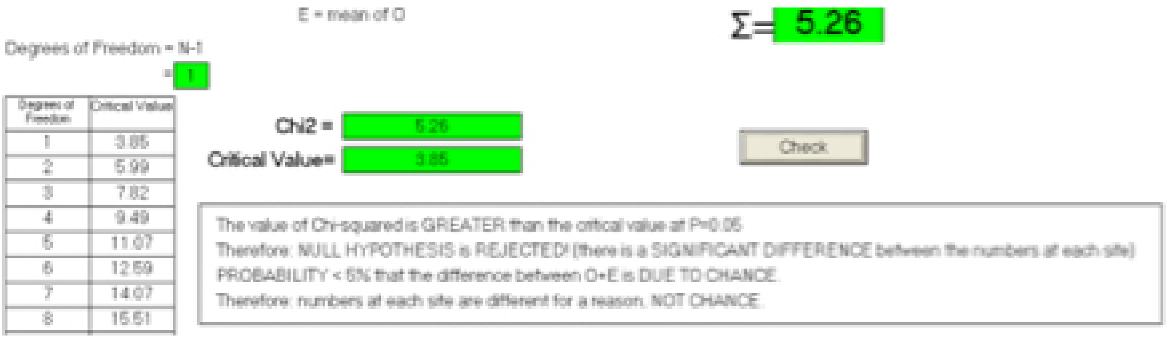
As I expected, the value was 2.63 was accepted.

7. The value of $Chi^2$ is the sum of the last text box of both the Top and Bottom of Hill results.

In this case:

2.63+2.63 = 5.26

The critical value should be 3.85 as the Degrees of Freedom are equal to 1. Therefore, the correct evaluation that should appear when everything is right is the one for when the value of $Chi^2$ is greater than the value of the critical value. This therefore means we reject the null hypothesis and accept that the differences between the two sites are significant and not due to chance.

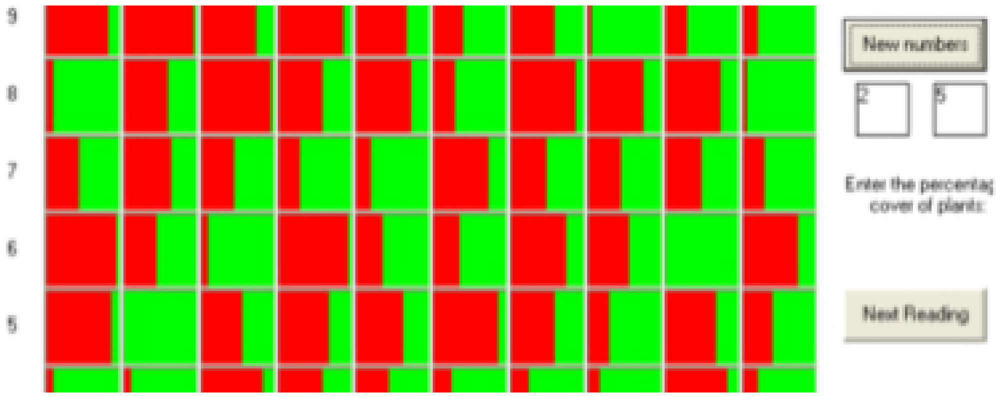



As expected, the correct evaluation box appeared and the value of $Chi^2$ was seen as correct.

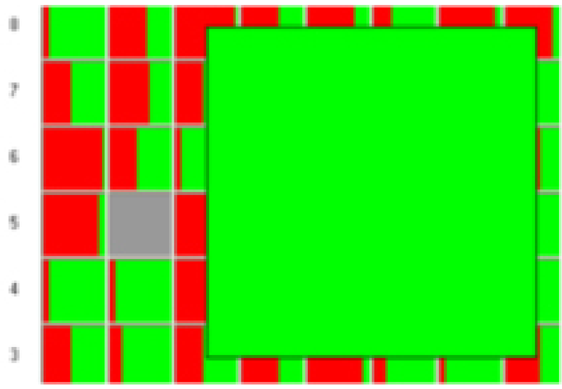
This testing shows me that no errors lie in Simulation 1 and the $Chi^2$ statistical test.

## Simulation 2:

1. Like before, I cannot screen shot my mouse icon so I cannot show that I cannot select an incorrect box but it definitely does not happen.
2. When I select the correct box, I expect the same image to appear on the large image box.

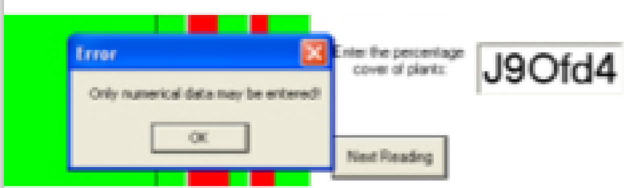



From the above screen shot, you can see that the co-ordinates (2, 5) have been generated. I expect the same image to be shown by the large image box when I select it.



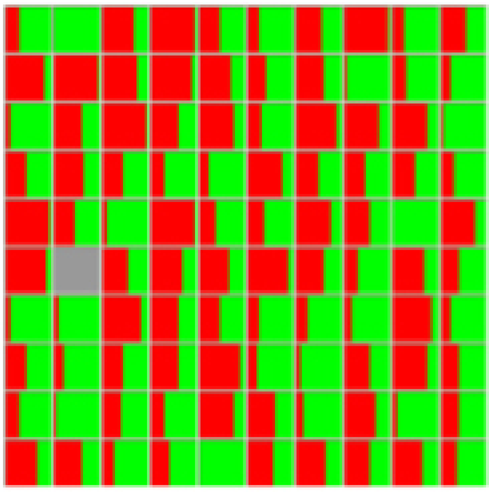
As expected, the correct image was displayed by the larger image box.

3. When I enter string characters (J9Ofd4) into the text box, the value should be rejected when I click on the "next reading" button.

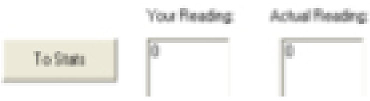



As you can see above, the string characters were rejected by the program, making an error message pop up, telling the user that only numerical data may be entered.

4. If I enter the correct, numerical value into the box, the value should be accepted. At the bottom of the form, there are two text boxes. One that displays the reading that the user inputted and the other one that the displays the actual reading. Both boxes should show the same number in them. In this case, the correct value should be 0.

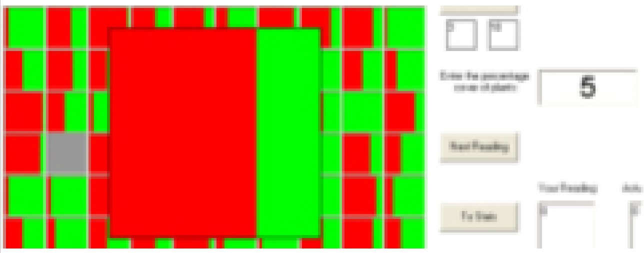


As you can see, the large image box disappeared. This means the value that I entered was accepted by the program and I can proceed with the next reading.




Here are the two text boxes and as you can see, they both display the value I entered before (0). This means that it was the correct percentage.

5. If I enter an incorrect numerical value, the program will still accept it but the right text box will display the value the image actually was.




As you can see, this is most definitely not 5%. However, I want to make sure the right text box displays the correct value (70%) and the left text box displays the value that I enter (5%).


As expected, the correct value was displayed in the left text box and the right text box displayed the value I had entered.

6. I've only taken two readings so I expect the counter to say 2/10.


As I had expected, the counter was at 2/10.

When the counter reaches 10, the simulation should refresh itself and the "South Facing" label should turn green and the "North Facing" label should turn grey.


As you can see, the program refreshed itself and the "South Facing" label turned green.

7. When I complete the simulation, the "Next Reading" button should disappear and in its place, the percentage of how well you did in taking the reading should appear.

The readings I took were: 90,95,50,0,45,80,55,50,0,40,85,95,0,45,60,35,65,90,90,10

The actual readings were: 90,95,50,0,45,80,60,55,0,40,85,95,0,45,65,35,65,85,90,10

I got 4 readings wrong. This means I got 16 correct

As a percentage, this would be 80%

16/20 = 0.8

0.8 x 100 = 80

**You Scored: 80%**

As expected, the percentage was 80%, meaning the program was counting it up correctly as the simulation progressed.

8. The readings I took were: 90,95,50,0,45,80,55,50,0,40,85,95,0,45,60,35,65,90,90,10

When I go to the stats page via the "stats" button, I expect to have those same results carried across.

| | % cover of plants in each 0.5m x 0.5m quadrat | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Site | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | mean |
| North Facing | 90 | 95 | 50 | 0 | 45 | 80 | 55 | 50 | 0 | 40 | 50.5 |
| South Facing | 85 | 95 | 0 | 45 | 60 | 35 | 65 | 90 | 90 | 10 | 57.5 |

As you can see from above, the results have all been taken into the stats correctly.

## Standard Error / 95% Confidence Limits:

1. With the data from the stats, I need to make sure the correct mean has been calculated.

| | % cover of plants in each 0.5m x 0.5m quadrat | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Site | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | mean |
| North Facing | 90 | 95 | 50 | 0 | 45 | 80 | 55 | 50 | 0 | 40 | 50.5 |
| South Facing | 85 | 95 | 0 | 45 | 60 | 35 | 65 | 90 | 90 | 10 | 57.5 |

Using this data, the mean for North Facing would be:

90+95+50+0+45+80+55+50+0+40 = 505

505/10 = 50.5

For South Site:

85+95+0+45+60+35+65+90+90+10 = 575

575/10 = 57.5

This shows that the means were accurately counted by the program.

2. If I press the "check" button without having entered anything, I expect that the text boxes will go red and nothing will be accepted.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | | |
| Standard Deviation (s) | | |
| Standard Error (SE) | | |
| 95% Confidence Limits (2 x SE) | | |
| Range of % cover | - | - |
| Overlap/ no Overlap? | YES? | NO? |
| Accept/ Reject Ho at p=0.05? | Accept? | Reject? |

As expected, the text boxes went red and nothing has been accepted.

3. If I enter string characters (Hk66nE) into the text boxes and click on "check" once again, I expect that nothing shall have changed and the same result will be met.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | Hk66nE | Hk66nE |
| Standard Deviation (s) | Hk66nE | Hk66nE |
| Standard Error (SE) | Hk66nE | Hk66nE |
| 95% Confidence Limits (2 x SE) | Hk66nE | Hk66nE |
| Range of % cover | Hk66nE - Hk66nE | Hk66nE - Hk66nE |
| Overlap/ no Overlap? | YES? | NO? |
| Accept/ Reject Ho at p=0.05? | Accept? | Reject? |

As expected, the same result was met and all text boxes remained red.

4. If I enter the correct mean into the North Facing table (50.5), I predict that the box will turn green (meaning the value was accepted) and the following text box will then turn red.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | Hk66nE |
| Standard Deviation (s) | Hk66nE | Hk66nE |
| Standard Error (SE) | Hk66nE | Hk66nE |
| 95% Confidence Limits (2 x SE) | Hk66nE | Hk66nE |

As I had expected, the text box turned green and the one following it turned red.

90,95,50,0,45,80,55,50,0,40,85,95,0,45,60,35,65,90,90,10

The next step is to calculate the Standard Deviation of the results.

$(90-50.5)^2 = 1560.25$

$(95-50.5)^2 = 1980.25$

$(50-50.5)^2 = 0.25$

$(0-50.5)^2 = 2550.25$

$(45-50.5)^2 = 30.25$

$(80-50.5)^2 = 870.25$

$(55-50.5)^2 = 20.25$

$(50-50.5)^2 = 0.25$

$(0-50.5)^2 = 2550.25$

$(40-50.5)^2 = 110.25$

$\Sigma = 9672.5$

$9672.5 / 10 = 967.25$

$\sqrt{967.25} = 31.1006430802$

The Standard Deviation of the North Facing set of data is 31.1006430802. As the program only accepts numbers to a max of 2.d.p, the accepted answer would be 31.10.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | Hk66nE |
| Standard Deviation (s) | 31.10 | 2 decimal places |
| Standard Error (SE) | Hk66nE | Hk66nE |
| 95% Confidence Limits (2 x SE) | Hk66nE | Hk66nE |

As expected, the value I had calculated (31.10) was accepted.

The next step is to calculate the Standard Error. This is the Standard Deviation divided by the square root of N; where N is the number of readings you have in that set of data.

$31.10 / \sqrt{10} = 9.8346835231$

As the program only accepts values to a maximum of 2.d.p, the accepted value would be 9.83.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | Hk66nE |
| Standard Deviation (s) | 31.10 | Hk66nE |
| Standard Error (SE) | 9.83 | Hk66nE |
| 95% Confidence Limits (2 x SE) | Hk66nE | Hk66nE |

As expected, the value I calculated was accepted by the program and the text box turned green.

The 95% Confidence Limits is simply the Standard Error multiplied by 2.
9.83 x 2 = 19.66

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | Hk66nE |
| Standard Deviation (s) | 31.10 | Hk66nE |
| Standard Error (SE) | 9.83 | Hk66nE |
| 95% Confidence Limits (2 x SE) | 19.66 | Hk66nE |
| Range of % cover | Hk66nE _ Hk66nE | Hk66nE _ Hk66nE |

As expected, the value I had calculated was accepted by the program.
Lastly, I need to work out the range of the results. This is +/- the Confidence Limits from the mean.
50.5 – 19.66 = 30.84
50.5 + 19.66 = 70.16

I expect that the two boxes will go green when the left box has 30.84 written in it and the right box has 70.16 in it.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | Hk66nE |
| Standard Deviation (s) | 31.10 | Hk66nE |
| Standard Error (SE) | 9.83 | Hk66nE |
| 95% Confidence Limits (2 x SE) | 19.66 | Hk66nE |
| Range of % cover | 30.84 _ 70.16 | Hk66nE _ Hk66nE |

As predicted, the text boxes turned green, meaning the range was accepted by the program

5. I expect the same results from the South Facing results also. For the mean, I expect 57.5 to be the accepted value.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | 57.5 |
| Standard Deviation (s) | 31.10 | Hk66nE |
| Standard Error (SE) | 9.83 | Hk66nE |
| 95% Confidence Limits (2 x SE) | 19.66 | Hk66nE |
| Range of % cover | 30.84 - 70.16 | Hk66nE - Hk66nE |

As expected, the accepted value for the mean was 57.5

The next step is to calculate the Standard Deviation of the results.

$(85-57.5)^2 = 756.25$

$(95-57.5)^2 = 1406.25$

$(0-57.5)^2 = 3306.25$

$(45-57.5)^2 = 156.25$

$(60-57.5)^2 = 6.25$

$(35-57.5)^2 = 506.25$

$(65-57.5)^2 = 56.25$

$(90-57.5)^2 = 1056.25$

$(90-57.5)^2 = 1056.25$

$(10-57.5)^2 = 2256.25$

$\Sigma = 10562.50$

$10562.5 / 10 = 1056.25$

$\sqrt{1056.25} = 32.5$

From this, I expect the program to accept the value of 32.5 and therefore, the text box should turn green.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | 57.5 |
| Standard Deviation (s) | 31.10 | 32.5 |
| Standard Error (SE) | 9.83 | Hk66nE |
| 95% Confidence Limits (2 x SE) | 19.66 | Hk66nE |
| Range of % cover | 30.84 - 70.16 | Hk66nE - Hk66nE |

As expected, the accepted value was 32.5

The next step is to calculate the Standard Error. This is down by diving the Standard Deviation by the square root of N, where N is the number of readings in each set of data.

32.5 / √10 = 10.2774023955

However, the program should only accept values to a maximum of 2.d.p. this means that the value the program should accept will be 10.28

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | 57.5 |
| Standard Deviation (s) | 31.10 | 32.5 |
| Standard Error (SE) | 9.83 | 10.28 |
| 95% Confidence Limits (2 x SE) | 19.66 | Hk66nE |
| Range of % cover | 30.84 - 70.16 | Hk66nE - Hk66nE |

As expected, the value of 10.28 was accepted by the program and turned the text box green in colour.

The 95% Confidence Limits is simply the Standard Error multiplied by 2
10.28 x 2 = 20.56

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | 57.5 |
| Standard Deviation (s) | 31.10 | 32.5 |
| Standard Error (SE) | 9.83 | 10.28 |
| 95% Confidence Limits (2 x SE) | 19.66 | 20.56 |
| Range of % cover | 30.84 - 70.16 | Hk66nE - Hk66nE |

As expected, the value was accepted and the text box turned green in colour.

The next step is to calculate the range of the results. This is done by +/- the Confidence Limits from the mean.
57.5 – 20.56 = 36.94
57.5 + 20.56 = 78.06

The left text box should go green when I enter 36.94 and the right text box should go green when I enter 78.06 into it.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | 57.5 |
| Standard Deviation (s) | 31.10 | 32.5 |
| Standard Error (SE) | 9.83 | 10.28 |
| 95% Confidence Limits (2 x SE) | 19.66 | 20.56 |
| Range of % cover | 30.84 - 70.16 | 36.94 - 78.06 |

As expected, the values were accepted and the colour of the text boxes turned to green.

6. The next step is to determine whether the overlap and null hypothesis buttons work. In this example, there is an overlap;

30.84 – 36.94 – 70.16

This means that we should accept the null hypothesis. To make sure that it is definitely the case, I will select "No Overlap" and "Reject" null hypothesis. Initially, both should go yellow and when I click on the "check" button, they should both go red again.

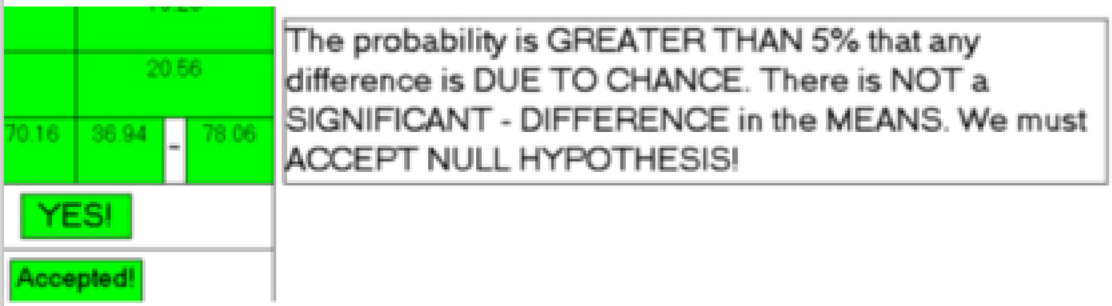
| | | |
|---|---|---|
| Overlap/ no Overlap? | YES? | NO? |
| Accept/ Reject Ho at p=0.05? | Accept? | Reject? |

As predicted, both buttons turned yellow. Now, when I click on the "check" button, they should both go red again.

| | North Facing | South Facing |
|---|---|---|
| Mean % cover | 50.5 | 57.5 |
| Standard Deviation (s) | 31.10 | 32.5 |
| Standard Error (SE) | 9.83 | 10.28 |
| 95% Confidence Limits (2 x SE) | 19.66 | 20.56 |
| Range of % cover | 30.84 - 70.16 | 36.94 - 78.06 |
| Overlap/ no Overlap? | YES? | NO? |
| Accept/ Reject Ho at p=0.05? | Accept? | Reject? |

As expected, they both went red again.

7. I now need to make sure that the correct evaluation message appears. When I select the "Yes" to there being an overlap and then "Accept" null hypothesis, the evaluation message should appear. From this example, it should follow the buttons pretty much. Because there is an overlap, we accept the null hypothesis and any difference we do see is purely down to chance.

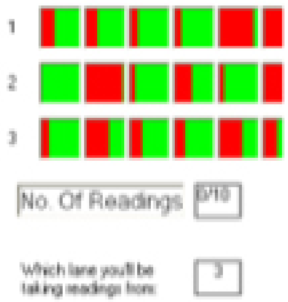



As expected, the correct evaluation message appeared and both the buttons turned green, meaning they were accepted.

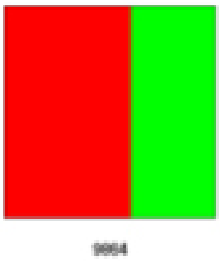
## Simulation 3:

Before I had even started the testing for this final simulation, I discovered that the form height was too long so the bottom was cut off slightly. Thankfully I saw this problem before I had given it to my end user for the Beta Testing.
I simply had to move some of the things around a little and it now fits on the screen.


1. I cannot click on a box other than the correct one. Nothing will happen even if I try
2. Likewise, I cannot select a box other than the one that I am supposed to select.
3. If I select the second image box in the correct lane, I expect the large image box to appear and with the same image as the one that I had selected.




As you can see from the above screen shot, I have to take readings from the third lane. When I click on the image box I have highlighted, I expect the large image box to include the same image.




As expected, the same image was displayed on the large image box.

4. I have set the light intensity to change every 10 seconds and instantly when the "next reading" button is pressed. To ensure this is the case, I will wait for 30 seconds and I expect the light intensity value to have changed 3 times.

Initially:

11702

After 10 seconds:

12222

After 20 seconds:

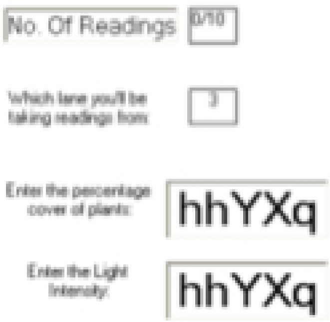
13194

After 30 seconds:

11491

As expected, the value changed every 10 seconds to a different value.

5. When I enter string characters into the text boxes, nothing should happen. The result should not get accepted into the program.




When I clicked on the "next reading" button, nothing happened. This is what I had expected. However, I shall add an error message like I had with the other simulations.

6. I set it so that the light intensity value cannot change if either of the text boxes is not empty. This is to ensure that the value doesn't change when in the middle of entering the previous light intensity. I shall wait 30 seconds again and expect the light intensity to remain the same.

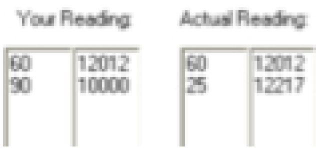

The light intensity value remained unchanged for the 30 seconds.

7. If I enter the correct values in, I expect the values to be accepted and in the two sets of text boxes I have (that show the users inputted value and the actual value), they should both display the same results.


As expected, the same results were displayed.

8. If I now enter incorrect data, I still expect the values to be accepted but the two sets of text boxes at the bottom should display different things. The left set should show the value that I entered and the right set should show the actual value.


As you can see, the data I am about to enter is clearly incorrect. In the left set of text boxes, I expect 90 and 10000 to be displayed and on the right, 25 and 12217.

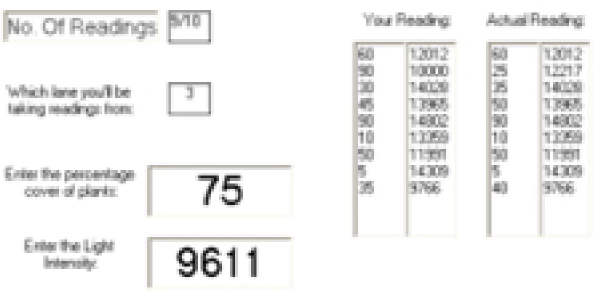



As expected, the values were accepted and the correct values were displayed in the correct set of text boxes.

9. As I have only taken two readings, I expect the counter to be at 2/10.


As expected, the counter is at 2/10.

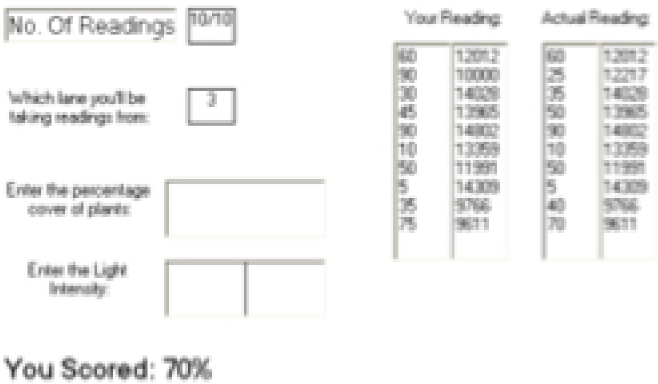
10. When I complete the simulation, I expect the "next reading" button to disappear and to be replaced by a label that tells you how accurate you were in the recording of the values.



Using the above set of results, the values I recorded for the percentage were:
60, 90, 30, 45, 90, 10, 50, 5, 35, 75
The values I recorded for the light intensity were:
12012, 10000, 14028, 13965, 14802, 13359, 11991, 14309, 9766, 9611

From the percentage, I got 4 wrong. If the last value, which I have yet to input, is also wrong, that would make 5. The only light intensity value I got incorrect was the second one which I did on purpose for the testing. This would mean I got 9 out of 10 there. This means the total correct answers I could have achieved will be either 15 or 14.

1. (15/20) x 100 = 75%
2. (14/20) x 100 = 70%

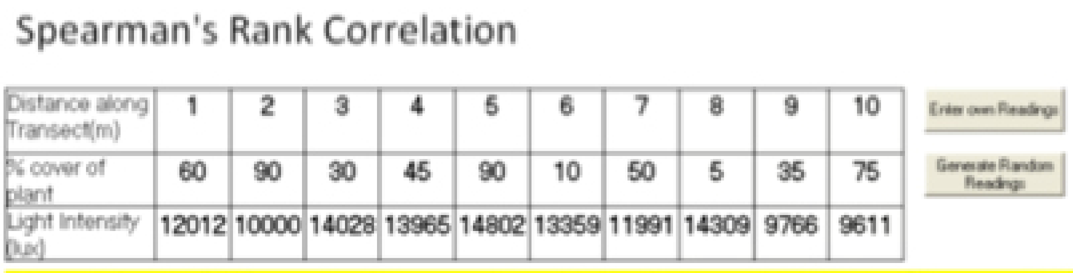



As the last value was incorrect, the total correct answers I scored were 14 out of 20, thus making the percentage 70%. This is what I had expected so I know this now works.

11. The results I recorded from the simulation were:

60, 90, 30, 45, 90, 10, 50, 5, 35, 75
12012, 10000, 14028, 13965, 14802, 13359, 11991, 14309, 9766, 9611

When I click on the "stats" button, I expect to be taken to the Spearman's Rank stats page and for the same values to transfer also.



Spearman's Rank Correlation

| Distance along Transect(m) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| % cover of plant | 60 | 90 | 30 | 45 | 90 | 10 | 50 | 5 | 35 | 75 |
| Light Intensity (lux) | 12012 | 10000 | 14028 | 13965 | 14802 | 13359 | 11991 | 14309 | 9766 | 9611 |



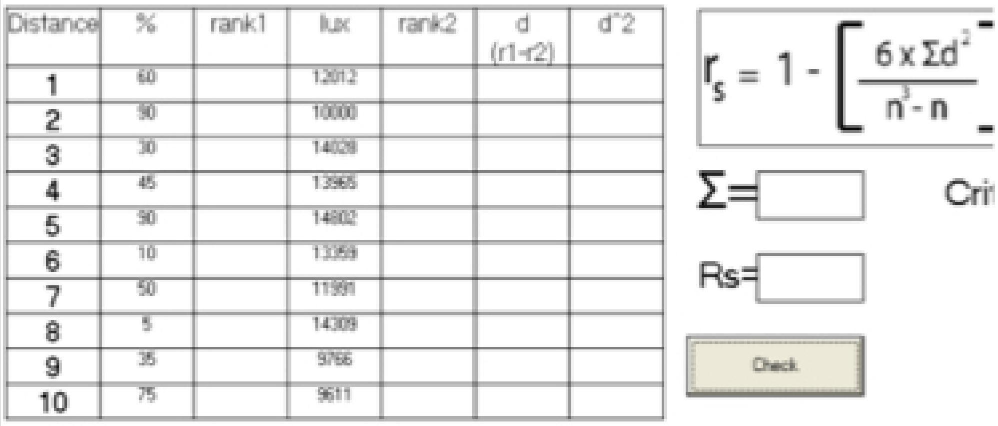
As expected, the correct values were transferred over.

## Spearman's Rank Correlation:

1. I need to ensure that the results in the stats table are the same as the results in the results table. If they are not, then the whole stats will be wrong and the results obtained as a result will be unreliable.

| Distance along Transect(m) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| % cover of plant | 60 | 90 | 30 | 45 | 90 | 10 | 50 | 5 | 35 | 75 |
| Light Intensity (lux) | 12012 | 10000 | 14028 | 13965 | 14802 | 13359 | 11991 | 14309 | 9766 | 9611 |

The results from the Belt Transect have been recorded and displayed in the results table above.
Spearman's Rank Correlation test on the data you collected, completing the form out below for e

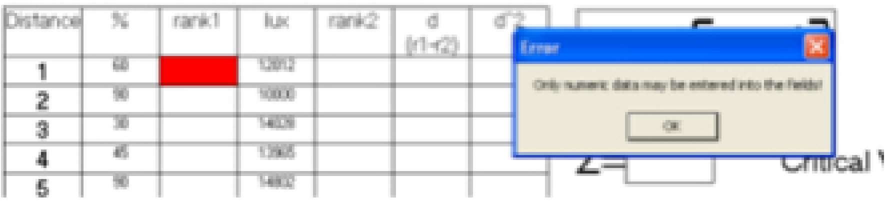


| Distance | % | rank1 | lux | rank2 | d (r1-r2) | d^2 |
|---|---|---|---|---|---|---|
| 1 | 60 | | 12012 | | | |
| 2 | 90 | | 10000 | | | |
| 3 | 30 | | 14028 | | | |
| 4 | 45 | | 13965 | | | |
| 5 | 90 | | 14802 | | | |
| 6 | 10 | | 13359 | | | |
| 7 | 50 | | 11991 | | | |
| 8 | 5 | | 14309 | | | |
| 9 | 35 | | 9766 | | | |
| 10 | 75 | | 9611 | | | |



As expected, the stats table contains the same values as those in the results table. This means the results from the stats test will be reliable.

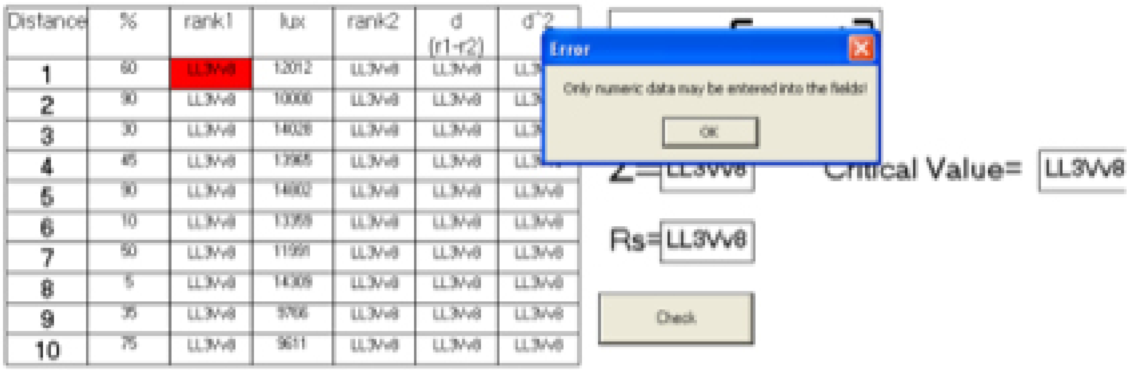
2. I don't want any blanks to be accepted into the test. If I press the "check" button without having entered anything, I expect the program to turn the text boxes red and for nothing else to happen.





The way I programmed it was that the program would exit the checking procedure when it encountered a problem. If I enter nothing into the table, it gets caught up in the very beginning and won't check anything else. This means the program did as expected.

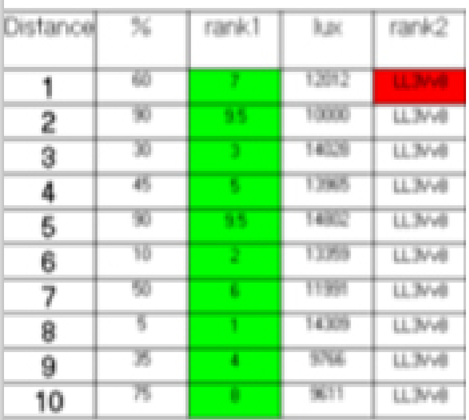
3. I shall enter string characters (LL3Vv8) into every text box. I expect the same result will as before will happen.




As expected, the same error message appeared like last time and nothing has been accepted.

4. Now I need to ensure that the ranks are being calculated correctly. For the percentage:
   60, 90, 30, 45, 90, 10, 50, 5, 35, 75
   7, 9.5, 3, 5, 9.5, 2, 6, 1, 4, 8

These are the expected ranks, in ascending order. There are two "95's" which took the ranks of 9 and 10. Because of this, the mean of the two ranks was worked out and the value was assigned to both of them; in this case, 9.5.




As expected, the ranks were all accepted and consequently, all the text boxes turned green.

5. Now I must do the same for the light intensity values.
   12012, 10000, 14028, 13965, 14802, 13359, 11991, 14309, 9766, 9611
   5, 3, 8, 7, 10, 6, 4, 9, 2, 1.

These are the ranks I expect the light intensity to accept.

| % | rank1 | lux | rank2 | d (r1-r2) |
|---|---|---|---|---|
| 60 | 7 | 12012 | 5 | LL3Vv8 |
| 90 | 9.5 | 10000 | 3 | LL3Vv8 |
| 30 | 3 | 14028 | 8 | LL3Vv8 |
| 45 | 5 | 13965 | 7 | LL3Vv8 |
| 90 | 9.5 | 14802 | 10 | LL3Vv8 |
| 10 | 2 | 13359 | 6 | LL3Vv8 |
| 50 | 6 | 11991 | 4 | LL3Vv8 |
| 5 | 1 | 14309 | 9 | LL3Vv8 |
| 35 | 4 | 9766 | 2 | LL3Vv8 |
| 75 | 8 | 9611 | 1 | LL3Vv8 |

As expected, the ranks were all accepted and the text boxes all turned green.

6. Next I need to calculate the value of "d." however; I need to make sure that it doesn't accept any values that should be negative.

| % | rank1 | lux | rank2 | d (r1-r2) |
|---|---|---|---|---|
| 60 | 7 | 12012 | 5 | 2 |
| 90 | 9.5 | 10000 | 3 | 6.5 |
| 30 | 3 | 14028 | 8 | 5 |
| 45 | 5 | 13965 | 7 | 2 |
| 90 | 9.5 | 14802 | 10 | 0.5 |
| 10 | 2 | 13359 | 6 | 4 |
| 50 | 6 | 11991 | 4 | 2 |
| 5 | 1 | 14309 | 9 | 8 |
| 35 | 4 | 9766 | 2 | 2 |
| 75 | 8 | 9611 | 1 | 7 |

The highlighted text boxes are the ones whose value should be a negative number.

| rank1 | lux | rank2 | d (r1-r2) | d^2 |
|---|---|---|---|---|
| 7 | 12012 | 5 | 2 | LL3Vv8 |
| 9.5 | 10000 | 3 | 6.5 | LL3Vv8 |
| 3 | 14028 | 8 | 5 | LL3Vv8 |
| 5 | 13965 | 7 | 2 | LL3Vv8 |
| 9.5 | 14802 | 10 | 0.5 | LL3Vv8 |
| 2 | 13359 | 6 | 4 | LL3Vv8 |
| 6 | 11991 | 4 | 2 | LL3Vv8 |
| 1 | 14309 | 9 | 8 | LL3Vv8 |
| 4 | 9766 | 2 | 2 | LL3Vv8 |
| 8 | 9611 | 1 | 7 | LL3Vv8 |

As expected, the text boxes which I took the negative sign away from turned red.

7. The next step is to rectify the wrong answers in the "d" column and then complete the "$d^2$" column. In this example:

$(2)^2 = 4$
$(6.5)^2 = 42.25$
$(-5)^2 = 25$
$(-2)^2 = 4$
$(-0.5)^2 = 0.25$
$(-4)^2 = 16$
$(2)^2 = 4$
$(-8)^2 = 64$
$(2)^2 = 4$
$(7)^2 = 79$

| rank2 | d (r1-r2) | d^2 |
|---|---|---|
| 5 | 2 | 4 |
| 3 | 6.5 | 42.25 |
| 8 | -5 | 25 |
| 7 | -2 | 4 |
| 10 | -0.5 | 0.25 |
| 6 | -4 | 16 |
| 4 | 2 | 4 |
| 9 | -8 | 64 |
| 2 | 2 | 4 |
| 1 | 7 | 49 |

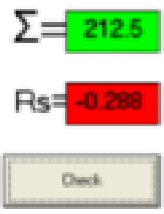
As expected, the values I calculated were correct and the text boxes turned green.

8. Now I need to calculate the Spearman's Rank. To do this, I will need to work out the sum of all the $d^2$ values
   Σ=212.5

$$r_s = 1 - \left[\frac{6 \times \Sigma d^2}{n^3 - n}\right]$$

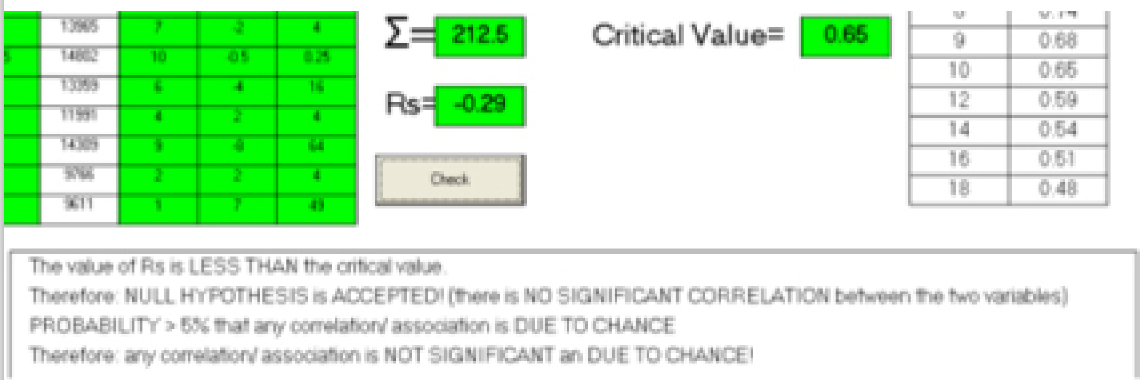
This is the formula for Spearman's Rank.
6 x 212.5 = 1275
103-10 = 990
1275/990 = 1.287878787…
1-1.28787878787 = -0.287878787…

As my system should only accept values to a maximum of 2.d.p, I will enter it to 3.d.p to make sure that this is the case. I will input -0.288 and I expect the text box to go red.




As predicted, the Spearman's Rank wasn't accepted and the box remained red.

9. If I now enter -0.29, the text box should go green. If I also enter in the correct critical value, then an evaluation box should appear.
   The evaluation box should say to accept the null hypothesis. This is because the value of Rs is less than the critical value (ignoring any signs)
   0.29 < 0.65




As predicted, the value of Spearman's Rank was accepted when I changed it to 2.d.p and the correct evaluation box appeared as a result.

# Beta Testing:

## Simulation 1/2:

This is the beta testing for the program I have come to you about it. Basically, you will test the program as a whole and record anything you think is good or needs improvement. I will provide you with some instruction to follow so that you can see how the program works. If something seems unclear and you would not have realised how to do it without the instructions to understand, make a note of it (in the space I will also provide). I will take into account everything you say and will make the amendments you suggest.

This is to finalise the program for you to use as a fully functional program.

### Simulation 1/2:

Firstly, the coordinates for the quadrat sampling are given to you in on the right. You can generate new coordinates by clicking on the button just above these coordinates. This is because you may get repeated coordinates so I thought it good to allow you the option of changing the coordinates at your will.

The 10x10 grid on the left is the sampling area. When you select the correct box, a larger image box should pop up on the screen, displaying the same image as the one you just selected. Enter the value into the text box on the left and press the "Next Reading" button. You should see the reading you entered in the large text box at the bottom of the simulation. If you countered correctly, it should be displayed twice. If you got it wrong, the correct value will be displayed on the right and your value on the left.

For the purposes of the testing, could you write down the results you get. This will be to show that the program works (or doesn't if that may be the case).

Continue doing the simulation until you get to 10 readings. The program may freeze momentarily but the simulation should be refreshed with all new images. The same process as you did for the first 10 readings will apply to these readings as well.

Once you have completed all 20 readings, the simulation should stop. You shouldn't be able to take any more readings and a percentage should be given to you, in place of the "Next Reading" button. This should display the correct percentage based on how many readings you got correct.

If you're on Simulation 1, the stats button should take you to Chi$^2$ and if you're on Simulation 2, the button should take you to the Standard Error stats page.

**Comments:**

Simulation 1: 9, 5, 0, 8, 1, 1, 4, 7.5, 6/1, 5, 9, 9, 0, 3, 7, 2, 1, 1

Simulation 2: 15, 80, 35, 40, 15, 60, 60, 95, 55, 50/25, 60, 60, 40, 35, 60, 95, 30, 25, 10

I like how the simulation is easy to use and you can see the results you have entered better. My main concern is that it may not be entirely obvious about pressing the button for the random numbers each time. It seemed straight forward to me but I don't know whether this was as a result of having seen this first.

## Simulation 3:

### Simulation 3:

You'll see that you have three different "lanes" for the transect lines. You will be given a random number below which will tell you which lane you will be taking the readings from. You'll also see a box with a number that constantly changes. This is the Light Intensity.

Because the simulation is of an interrupted belt transect, you will need to take readings from every other box. This therefore means that you will need to start on the second box on the lane given to you and take readings from every other box from that one. Like the other simulations, there will be a set of text boxes that display the value you enter and the actual value. There are two in each set however because you're dealing with a set variable of Light Intensity now as well.

Select the first box and record the reading and the light intensity. The counter should increase by 1 and the two readings should be displayed in the text boxes at the bottom, as well as the correct values. The simulation only goes to 10 as you're only taking 10 readings. Once the 10 readings have been completed, the evaluation text box will appear, displaying the correct text according to the stats results.

Once the stat's has been completed, press the help button on the top corner and follow the guide. I would like you to see if the vocabulary and information is clear and concise and if it could be used as a good guide for students to use.

Comments:

% : 10, 30, 25, 25, 65, 45, 85, 0, 15, 20

Light Intensity = 14138, 11642, 9058, 10158, 13921, 9857, 11928, 12943, 12133, 13185

This simulation is definitely the most confusing. I will obviously be using the program alongside reading the review so my students will understand but it is still somewhat confusing.

I think leaving it as it is for now will be the best course of action and I'll notify you of how the students react to the program and any bits they found misleading.

## Chi²:

### Chi²:

Firstly, could you make sure that the results you got from Simulation 1 match those that have been displayed in the results table? I'd then like you to add up the total number of plants counted at each site and write the results in the space provided. I'd like you to complete the stats and write down the workings you entered into the program. When everything is correct, an evaluation text box will appear at the bottom of the screen. This should be correct depending on the result from the stats.

Once the stats has been collected, access the help page (button on the top right) and go through the guide. I want to see if you think it is clear and concise and contains all the information that is required.

Comments:

The results in the table match those I recorded in the simulation. – Also, the results table is a good format and easy to understand.

total number of plants: 46
: 38

expected = $\frac{46 + 38}{2}$ = 42

O − E = 4
= −4

$(O-E)^2$ = 16
= 16

$\frac{(O-E)^2}{E}$ = 0.38

$Chi^2$ = 0.76

Dof = ~~[illegible]~~ 1

critical value = 3.84

I think the stats is very good and well made. I think the colour changing is a good indication to show if it's right or wrong.
The step by step guide is exactly what I wanted also. Well done!

## Standard Error:

### Standard Error:

Like you did with the Chi² stats test, make sure that the results table has recorded the correct results. Then calculate the mean for both sets of data. Then complete the rest of the stats, writing down the answers you got for each step. When all of it has been completed and it's all correct, the evaluation text box should pop up on the side and display the correct text according to the results from the stats.

Once the stat's has been completed, press the help button on the top corner and follow the guide. I would like you to see if the vocabulary and information is clear and concise and if it could be used as a good guide for students to use.

Comments:

The results in the table were also correct.

mean : ~~46.5~~ 52.5
= 45 ~~40.65~~ 43

Standard deviation = 25.77
= 25.95

Standard error = 8.15
= 8.21

95% confidence limits: 16.3
16.42

Range = 36.2 – 68.8
= 26.58 – 59.42

There is an overlap and so accept the H₀

You've done a really good work portraying these stats tests in an effective way.

The help page is also clear and concise and I can see it being a big help for the department.

## Spearman's Rank:

### Spearman's Rank Correlation:

Like you did previously, make sure that the readings in the results table matches up with the readings that you recorded. Then check to see that they have been displayed in the stats table properly also.

Like last time, complete the stats and record what you wrote down and any calculations you may have done. This stats test is a little more long winded as you have to write down the results of each calculation for each reading.

Once the stats test is complete and everything is correct, an evaluation text box should pop up, displaying the correct text according to the stats results.

Lastly, press the help button on the top corner and follow the guide. I would like you to see if the vocabulary and information is clear and concise and if it could be used as a good guide for students to use.

Comments:

all the results were in the results table correctly

| Rank 1 | Rank 2 | d | $d^2$ |
|---|---|---|---|
| 2 | 10 | -8 | 64 |
| 7 | 4 | 3 | 9 |
| 5.5 | 1 | 4.5 | 20.25 |
| 5.5 | 3 | 2.5 | 6.25 |
| 9 | 9 | 0 | 0 |
| 8 | 2 | 6 | 36 |
| 10 | 5 | 5 | 25 |
| 1 | 7 | -6 | 36 |
| 3 | 6 | -3 | 9 |
| 4 | 8 | -4 | 16 |

Sum = 221.5

$r_s = 0.78$

As you said, it's rather long winded but it works nonetheless and looks very easy to use.

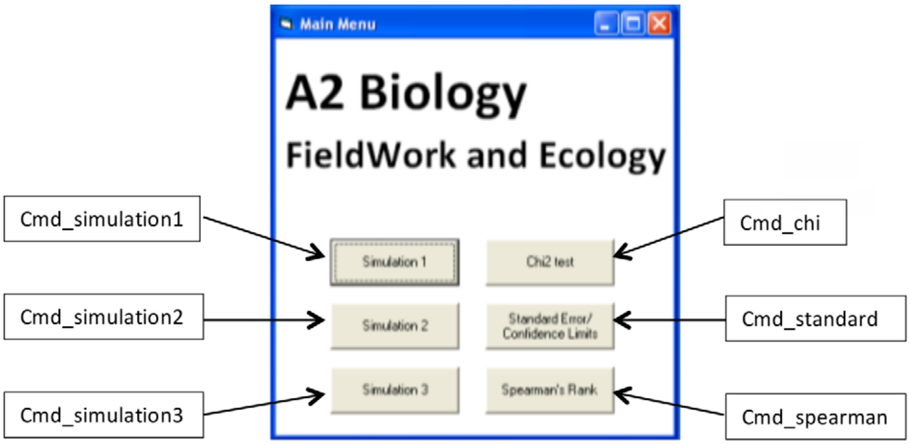
Once again, the ~~stats~~ help page is very clear and concise and will be a great help to the department.

# Technical Documentation:

## Main Module:

```
'Data Structure (Simulation etc)
Type Sim1_Readings
readings(1 tO 20) As Integer
End Type

'Global Variables
Global sim1 As Sim1_Readings
Global stats As Boolean
```

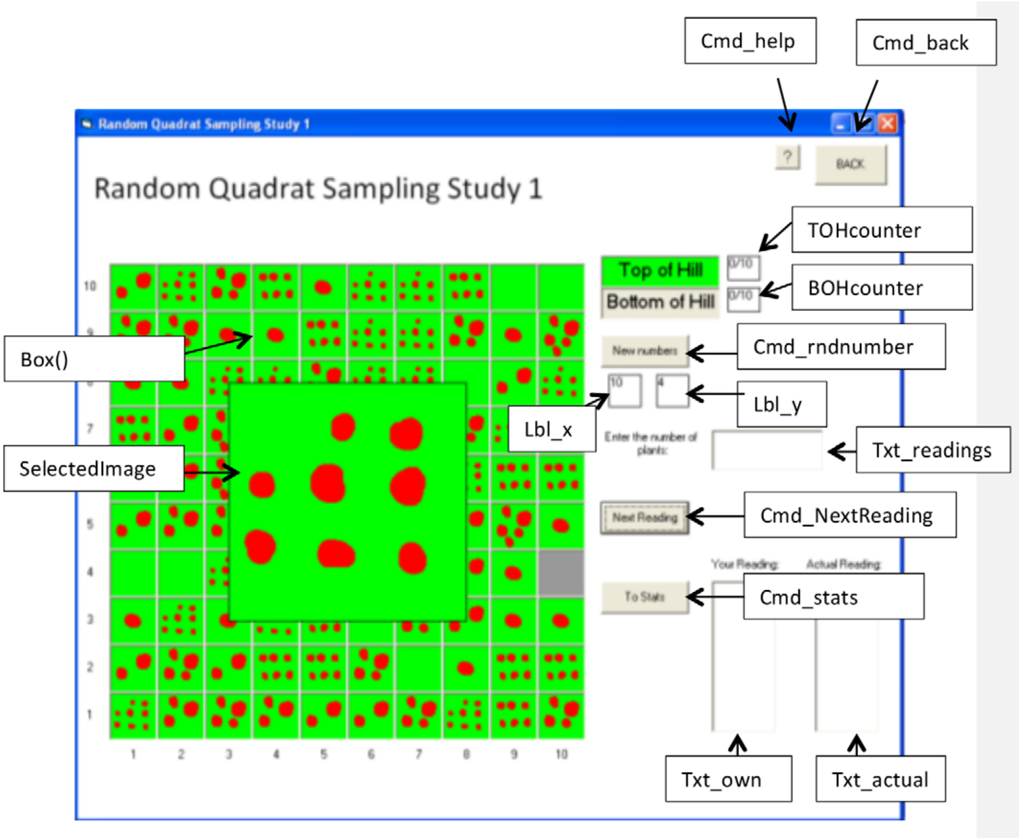
## Main Menu:




```
Private Sub cmd_chi2_Click()
Unload me
frm_Chi2.Show
End Sub

Private Sub cmd_simulation1_Click()
Unload me
frm_simulation1.Show
End Sub

Private Sub cmd_simulation2_Click()
Unload me
frm_simulation2.Show
End Sub

Private Sub cmd_simulation3_Click()
Unload me
frm_simulation3.Show
End Sub

Private Sub cmd_spearmans_Click()
Unload me
frm_Spearmans.Show
```

```
End Sub

Private Sub cmd_standard_Click()
Unload me
frm_StandardError.Show
End Sub

Private Sub Form_Load()
stats = False
For I = 1 tO 20
    sim1.readings(I) = 0
Next I
End Sub
```

## frm_simulation1:




```
Dim Sim_Begin As Boolean

Dim TOHrecorder As Integer
Dim BOHrecorder As Integer

Dim rnd_labelX As Integer
Dim rnd_labelY As Integer
Dim random_equationx As Integer
Dim random_equationy As Integer
Dim boxX As Integer
Dim boxY As Integer
Dim boxcheck(100) As Boolean
```

```
Dim RND_Number(100) As Integer

Dim pointer As Integer

Dim percent As Integer
Dim simFinished As Boolean



Private Sub box_Click(Index As Integer)

'assigning equation to a variable
random_equationx = 480 + (720 * (rnd_labelX - 1))
random_equationy = 8400 - (720 * (rnd_labelY - 1))

'Assigning box position to a variable
boxX = box(Index).Left
boxY = box(Index).top

'IF for checking if correct box has been chosen
If BOHrecorder < 10 Then
    If boxcheck(Index) = False Then
        If boxX = random_equationx Then
            If boxY = random_equationy Then
                SelectedImage.Picture = box(Index).Picture
                SelectedImage.Visible = True
                box(Index).Picture = LoadPicture(App.Path & "\images" & "\
simulation1" & "\grey.gif")
                boxcheck(Index) = True
            End If
        End If
    End If
End If
'saving random number
lbl_hidden.Caption = RND_Number(Index)

End Sub

Private Sub cmd_NextReading_Click()

'Counting up
If IsNumeric(txt_readings.Text) = True Then
    If txt_readings.Text <> "" Then
        If BOHrecorder < 10 Then
            If SelectedImage.Visible = True Then
                If TOHrecorder < 10 Then
                    TOHrecorder = TOHrecorder + 1
                    pointer = pointer + 1
                    TOHcounter.Caption = TOHrecorder & "/10"
                    sim1.readings(pointer) = txt_readings.Text
```

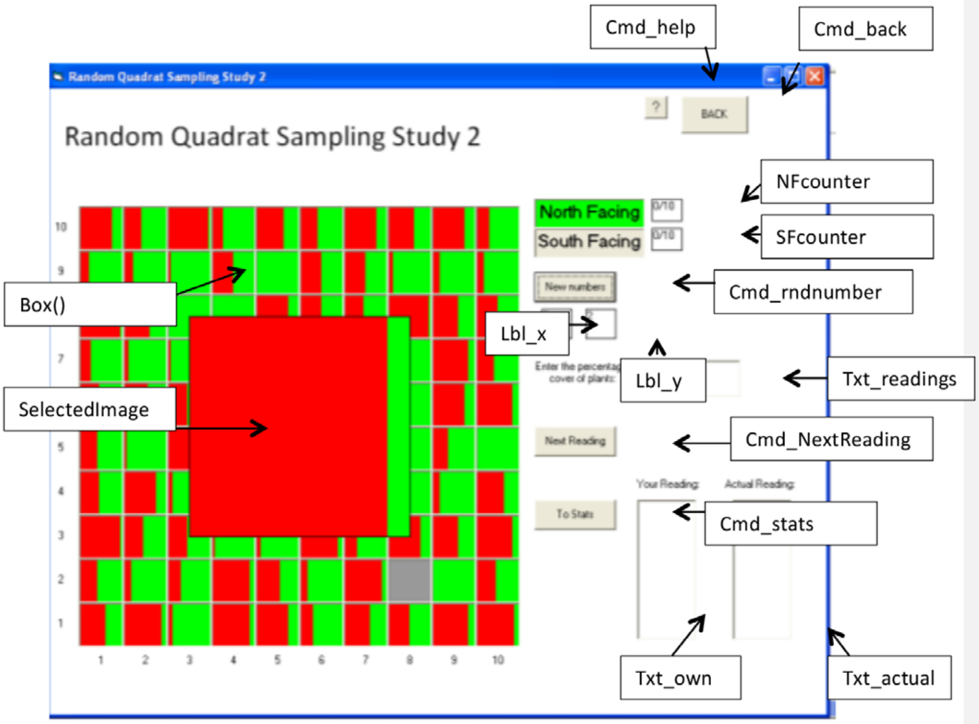
```
                    SelectedImage.Visible = False
                    If txt_readings.Text = lbl_hidden.Caption Then
                        percent = percent + 1
                    End If
                    txt_own.Text = txt_own & txt_readings.Text & vbNewLine
                    txt_actual.Text = txt_actual & lbl_hidden.Caption &
vbNewLine
                    txt_readings.Text = ""
                        If TOHrecorder = 10 Then
                            lbl_BOH.BackColor = &HFF00&
                            lbl_TOH.BackColor = &h8000000f
                            SelectedImage.Visible = False

                            'Copy of FOR NEXT to refresh image boxes with new
images
                            For I = 0 tO 99
                                Randomize
                                RND_Number(I) = Int(Rnd(1) * 9)

                                sTemp = "\images" & "\simulation1\" & RND_
Number(I) & ".gif"
                                box(I).Picture = LoadPicture(App.Path & sTemp)
                            Next I

                            'Resetting the BoxCheck to all false
                            For y = 0 tO 99
                                boxcheck(y) = False
                            Next y

                        End If
                ElseIf BOHrecorder < 10 Then
                    BOHrecorder = BOHrecorder + 1
                    pointer = pointer + 1
                    BOHcounter.Caption = BOHrecorder & "/10"
                    sim1.readings(pointer) = txt_readings.Text
                    SelectedImage.Visible = False
                    If txt_readings.Text = lbl_hidden.Caption Then
                        percent = percent + 1
                    End If
                    txt_own.Text = txt_own & txt_readings.Text & vbNewLine
                    txt_actual.Text = txt_actual & lbl_hidden.Caption &
vbNewLine
                    txt_readings.Text = ""
                    If BOHrecorder = 10 Then
                        lbl_BOH.BackColor = &h8000000f
                        percent = (percent / 20) * 100
                        lbl_percent.Caption = "You Scored: " & percent & " %"
                        lbl_percent.Visible = True
                        cmd_NextReading.Visible = False
                        simFinished = True
```

```
                    End If
                End If
            End If
        End If
    End If
Else
    MsgBox "Only numeric data may be entered!", , "Error"
End If

End Sub

Private Sub cmd_rndnumber_Click()

'Random Number Generator #2
If SelectedImage.Visible = False Then
    Randomize
    rnd_labelX = Int((10 - 1 + 1) * Rnd + 1)
    lbl_X.Caption = rnd_labelX
    Randomize
    rnd_labelY = Int((10 - 1 + 1) * Rnd + 1)
    lbl_Y.Caption = rnd_labelY
End If
End Sub

Private Sub cmd_stats_Click()
If simFinished = True Then
    stats = True
    Unload me
    frm_Chi2.Show
Else
    MsgBox "You have not completed the simulation yet! Please do so as it is
necessary for the stats!", , "Error"
End If
End Sub

Private Sub Command2_Click()
Unload me
MainMenu.Show
End Sub

Private Sub Form_Load()

'The Program to load random images into a 10x10 grid

'Determines whether the variable "Sim_Begin" is true, based on whether the form
is visible or not
If frm_simulation1.Visible = True Then
    Sim_Begin = True
Else
    Sim_Begin = False
```

```
    'Initialising TOH and BOH labels
    TOHrecorder = 0
    BOHrecorder = 0

    lbl_TOH.BackColor = &HFF00&
    lbl_BOH.BackColor = &h8000000f

    TOHcounter.Caption = TOHrecorder & "/10"
    BOHcounter.Caption = BOHrecorder & "/10"

End If

'The Random Number generator should activate when the form becomes visible.

If Sim_Begin = False Then
    For I = 0 tO 99
       Randomize
               RND_Number(I) = Int(Rnd(1) * 9)

        sTemp = "\images" & "\simulation1\" & RND_Number(I) & ".gif"
        box(I).Picture = LoadPicture(App.Path & sTemp)

'Standardises the image boxes to make sure that they're all uniform
       box(I).Stretch = True
       box(I).Height = 735
       box(I).Width = 735
       Sim_Begin = True
    Next I
Else
End If

'Generating Random Numbers for box co-ordinates
Randomize
rnd_labelX = Int((10 - 1 + 1) * Rnd + 1)
lbl_X.Caption = rnd_labelX
Randomize
rnd_labelY = Int((10 - 1 + 1) * Rnd + 1)
lbl_Y.Caption = rnd_labelY

'Putting a selected image onto a bigger image box - standardising it
SelectedImage.Stretch = True
SelectedImage.Height = 3615
SelectedImage.Width = 3615
SelectedImage.top = 3720
SelectedImage.Left = 2280
SelectedImage.Visible = False

'Greying out box if selected (part1)
For y = 0 tO 99
    boxcheck(y) = False
Next y
```

```
'initialising some variables
pointer = 0
percent = 0
simFinished = False

End Sub
```

## Frm_simulation2:




```
Dim Sim_Begin As Boolean

Dim NFrecorder As Integer
Dim SFrecorder As Integer

Dim rnd_labelX As Integer
Dim rnd_labelY As Integer
Dim random_equationx As Integer
Dim random_equationy As Integer
Dim boxX As Integer
Dim boxY As Integer
Dim boxcheck(100) As Boolean

Dim RND_Number(100) As Integer

Dim pointer As Integer
Dim percent As Integer
Dim simFinished As Boolean
```

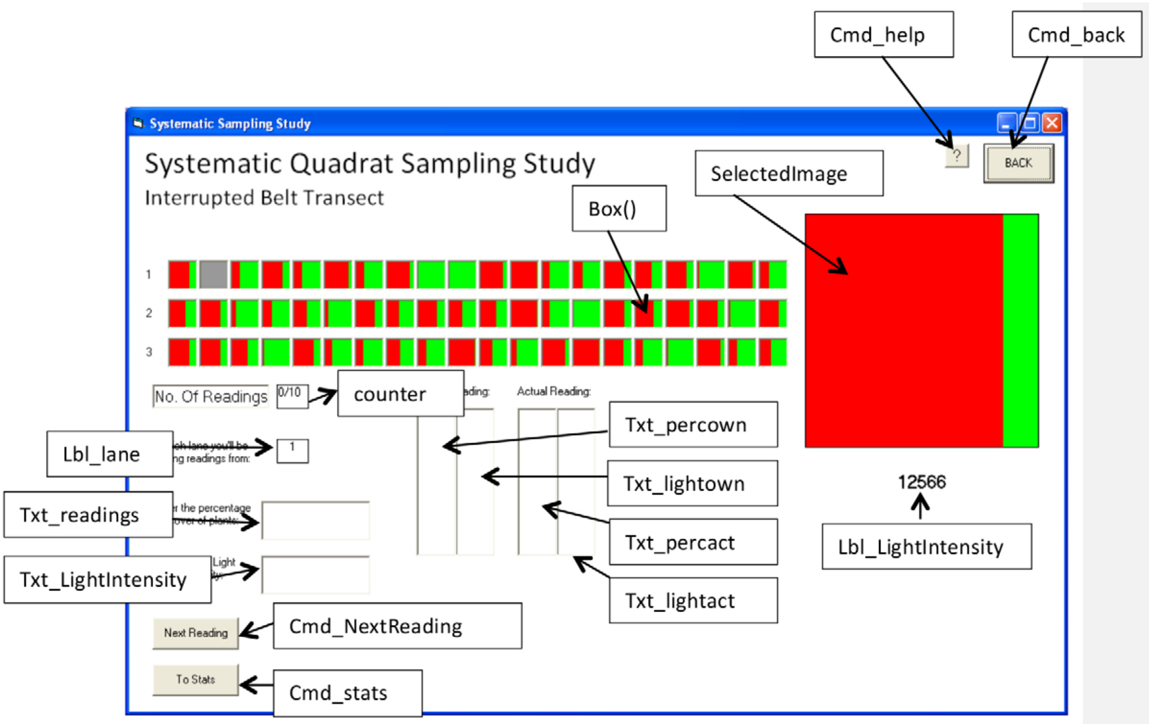
```
Private Sub box_Click(Index As Integer)

'assigning equation to a variable
random_equationx = 480 + (720 * (rnd_labelX - 1))
random_equationy = 8400 - (720 * (rnd_labelY - 1))

'Assigning box position to a variable
boxX = box(Index).Left
boxY = box(Index).top

'IF for checking if correct box has been chosen
If SFrecorder < 10 Then
    If boxcheck(Index) = False Then
        If boxX = random_equationx Then
            If boxY = random_equationy Then
                SelectedImage.Picture = box(Index).Picture
                SelectedImage.Visible = True
                box(Index).Picture = LoadPicture(App.Path & "\images" & "\
simulation1" & "\grey.gif")
                boxcheck(Index) = True
            End If
        End If
    End If
End If
'recording the random number
lbl_hidden.Caption = RND_Number(Index)

End Sub


Private Sub cmd_NextReading_Click()

'Counting up
If txt_readings.Text <> "" Then
    If SFrecorder < 10 Then
        If SelectedImage.Visible = True Then
            If NFrecorder < 10 Then
                NFrecorder = NFrecorder + 1
                pointer = pointer + 1
                NFcounter.Caption = NFrecorder & "/10"
                sim1.readings(pointer) = txt_readings.Text
                If txt_readings.Text = lbl_hidden.Caption Then
                    percent = percent + 1
                End If
                txt_own.Text = txt_own & txt_readings.Text & vbNewLine
                txt_actual.Text = txt_actual & lbl_hidden.Caption & vbNewLine
                txt_readings.Text = ""
```

```
                SelectedImage.Visible = False
                    If NFrecorder = 10 Then
                        lbl_BOH.BackColor = &HFF00&
                        lbl_TOH.BackColor = &h8000000f
                        SelectedImage.Visible = False

                        'Copy of FOR NEXT to refresh image boxes with new
images
                        For I = 0 tO 99
                            Randomize
                            RND_Number(I) = Int(Rnd(1) * 20) * 5

                            sTemp = "\images" & "\simulation2\" & RND_Number(I)
& ".gif"
                            box(I).Picture = LoadPicture(App.Path & sTemp)
                        Next I
                        'Resetting the BoxCheck to all false
                        For y = 0 tO 99
                            boxcheck(y) = False
                        Next y

                    End If
            ElseIf SFrecorder < 10 Then
                SFrecorder = SFrecorder + 1
                pointer = pointer + 1
                SFcounter.Caption = SFrecorder & "/10"
                sim1.readings(pointer) = txt_readings.Text
                txt_readings.Text = ""
                If txt_readings.Text = lbl_hidden.Caption Then
                    percent = percent + 1
                End If
                txt_own.Text = txt_own & txt_readings.Text & vbNewLine
                txt_actual.Text = txt_actual & lbl_hidden.Caption & vbNewLine
                SelectedImage.Visible = False
                    If SFrecorder = 10 Then
                        lbl_BOH.BackColor = &h8000000f
                        percent = (percent / 20) * 100
                        lbl_percent.Caption = "You Scored: " & percent & "%"
                        lbl_percent.Visible = True
                        cmd_NextReading.Visible = False
                        simFinished = True
                    End If
            End If
        End If
    End If
End If


End Sub
```

```
Private Sub cmd_rndnumber_Click()

'Random Number Generator #2
If SelectedImage.Visible = False Then
    Randomize
    rnd_labelX = Int((10 - 1 + 1) * Rnd + 1)
    lbl_X.Caption = rnd_labelX
    Randomize
    rnd_labelY = Int((10 - 1 + 1) * Rnd + 1)
    lbl_Y.Caption = rnd_labelY
End If
End Sub

Private Sub cmd_stats_Click()
If simFinished = True Then
    stats = True
    Unload me
    frm_StandardError.Show
Else
    MsgBox "You have not completed the simulation yet! Please do so as it is
necessary for the stats!", , "Error"
End If
End Sub

Private Sub cmd_back_Click()
Unload me
MainMenu.Show
End Sub


Private Sub Form_Load()
Dim sTemp As String
'The Program to load random images into a 10x10 grid

'Determines whether the variable "Sim_Begin" is true, based on whether the form
is visible or not
If frm_simulation1.Visible = True Then
    Sim_Begin = True
Else
    Sim_Begin = False
End If

'Initialising TOH and BOH labels
NFrecorder = 0
SFrecorder = 0

lbl_TOH.BackColor = &HFF00&
lbl_BOH.BackColor = &h8000000f

NFcounter.Caption = NFrecorder & "/10"
SFcounter.Caption = SFrecorder & "/10"
```

```
'The Random Number generator should activate when the form becomes visible.

If Sim_Begin = False Then
    For I = 0 tO 99
       Randomize
        RND_Number(I) = Int(Rnd(1) * 20) * 5

        sTemp = "\images" & "\simulation2\" & RND_Number(I) & ".gif"
        box(I).Picture = LoadPicture(App.Path & sTemp)

'Standardises the image boxes to make sure that they're all uniform
       box(I).Stretch = True
       box(I).Height = 735
       box(I).Width = 735
       Sim_Begin = True
    Next I
Else
End If

Randomize
lbl_X.Caption = Int((10 - 1 + 1) * Rnd + 1)
Randomize
lbl_Y.Caption = Int((10 - 1 + 1) * Rnd + 1)


'Putting a selected image onto a bigger image box - standardising it
SelectedImage.Stretch = True
SelectedImage.Height = 3615
SelectedImage.Width = 3615
SelectedImage.top = 3720
SelectedImage.Left = 2280
SelectedImage.Visible = False

'Greying out box if selected (part1)
For y = 0 tO 99
    boxcheck(y) = False
Next y


pointer = 0
simFinished = False

End Sub
```

## Frm_simulation3:




```
Dim Sim_Begin As Boolean

Dim random_equationy As Integer
Dim selection As Integer

Dim boxX As Integer
Dim boxY As Integer

Dim recorder As Integer

Dim boxcheck(60) As Integer
Dim RND_Number(60) As Integer

Dim lane As Integer
Dim LERandomNumber As Integer

Dim bothReadings As Boolean
Dim Ppointer As Integer
Dim Lpointer As Integer
Dim percent As Integer
Dim simFinished As Boolean




Private Sub box_Click(Index As Integer)
'selecting box and checking if it's correct
If recorder < 10 Then
    If boxcheck(Index) = False Then
```

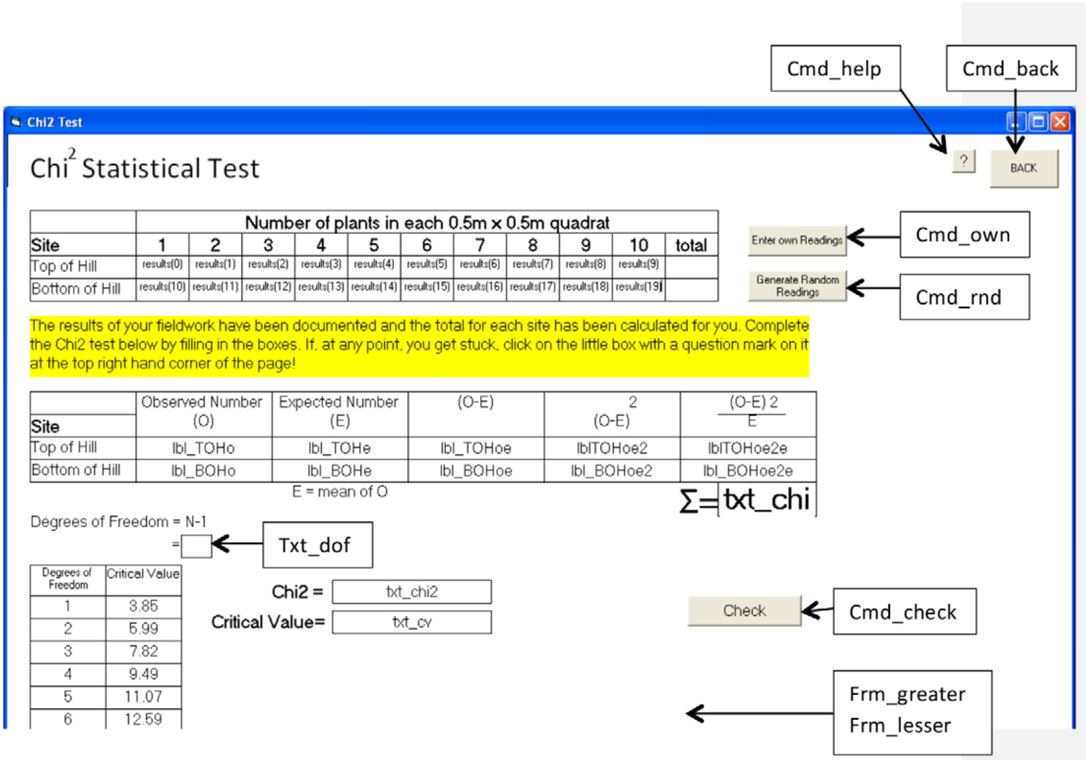
```
        Select Case lane
        Case 1
            selection = (2 * recorder) + 1
            If Index = selection Then
                SelectedImage.Picture = box(Index).Picture
                SelectedImage.Visible = True
                box(Index).Picture = LoadPicture(App.Path & "\images" & "\
simulation1" & "\grey.gif")
                boxcheck(Index) = True
            End If
        Case 2
            selection = (2 * recorder) + 21
            If Index = selection Then
                SelectedImage.Picture = box(Index).Picture
                SelectedImage.Visible = True
                box(Index).Picture = LoadPicture(App.Path & "\images" & "\
simulation1" & "\grey.gif")
                boxcheck(Index) = True
            End If
        Case 3
        selection = (2 * recorder) + 41
            If Index = selection Then
                SelectedImage.Picture = box(Index).Picture
                SelectedImage.Visible = True
                box(Index).Picture = LoadPicture(App.Path & "\images" & "\
simulation1" & "\grey.gif")
                boxcheck(Index) = True
            End If
        End Select
    End If
End If


lbl_hidden.Caption = RND_Number(Index)

End Sub

Private Sub cmd_NextReading_Click()
'recording the reading
If IsNumeric(txt_LightIntensity.Text) = True And IsNumeric(txt_readings.Text) =
True Then
    If recorder < 10 Then
        If bothReadings = False Then
            If txt_LightIntensity.Text <> "" Then
                If txt_readings.Text <> "" Then
                    Lpointer = Lpointer + 1
                    bothReadings = True
                    Ppointer = Ppointer + 1
                Else
                    MsgBox "Both Values must be entered!", , "Error"
                End If
```

```
                If bothReadings = True Then
                    sim1.readings(Ppointer) = txt_readings.Text
                    sim1.readings(Lpointer) = txt_LightIntensity.Text
                    If txt_readings.Text = lbl_hidden.Caption Then
                        percent = percent + 1
                    End If
                    If txt_LightIntensity.Text = lbl_LightIntensity.Caption
Then
                        percent = percent + 1
                    End If
                    txt_percown.Text = txt_percown & txt_readings.Text &
vbNewLine
                    txt_lightown.Text = txt_lightown & txt_LightIntensity.Text
& vbNewLine
                    txt_percact.Text = txt_percact & lbl_hidden.Caption &
vbNewLine
                    txt_lightact.Text = txt_lightact & lbl_LightIntensity.
Caption & vbNewLine
                    txt_readings.Text = ""
                    txt_LightIntensity = ""
                    recorder = recorder + 1
                    counter.Caption = recorder & "/10"
                    SelectedImage.Visible = False
                    bothReadings = False
                    Randomize
                    LERandomNumber = Int(((14 - 9 + 1) * Rnd + 9) * 1000)
                    lbl_LightIntensity.Caption = LERandomNumber
                Else
                    bothReadings = False
                End If
            End If
        End If
    End If
End If
If recorder = 10 Then
    percent = (percent / 20) * 100
    lbl_percent.Caption = "You Scored: " & percent & "%"
    lbl_percent.Visible = True
    cmd_NextReading.Visible = False
    simFinished = True
End If

End Sub

Private Sub cmd_stats_Click()
'Loads stats if sim is finished
If simFinished = True Then
    stats = True
    Unload me
    frm_Spearmans.Show
```

```
Else
    MsgBox "You have not completed the simulation yet! Please do so as it is
necessary for the stats!", , "Error"
End If
End Sub

Private Sub cmd_back_Click()
'returns to Main Menu
Unload me
MainMenu.Show
End Sub

Private Sub Form_Load()

'Determines whether the variable "Sim_Begin" is true, based on whether the form
is visible or not
If frm_simulation1.Visible = True Then
    Sim_Begin = True
Else
    Sim_Begin = False
End If

'initialising the recorder
recorder = 0

counter.Caption = recorder & "/10"

'loading in the images
If Sim_Begin = False Then
    For I = 0 tO 59
       Randomize
        RND_Number(I) = Int(Rnd(1) * 20) * 5

        sTemp = "\images" & "\simulation3\" & RND_Number(I) & ".gif"
        box(I).Picture = LoadPicture(App.Path & sTemp)

'Standardises the image boxes to make sure that they're all uniform
       box(I).Stretch = True
       box(I).Height = 465
       box(I).Width = 465
       Sim_Begin = True
    Next I
End If

'Putting a selected image onto a bigger image box - standardising it
SelectedImage.Stretch = True
SelectedImage.Height = 3615
SelectedImage.Width = 3615
SelectedImage.top = 1200
SelectedImage.Left = 10440
SelectedImage.Visible = False
```

```
'Greying out box if selected (part1)
For y = 0 tO 59
    boxcheck(y) = False
Next y
'random number to determine the lane the user takes the readings from
Randomize
lbl_lane.Caption = Int((3) * Rnd + 1)
lane = lbl_lane.Caption

'generating intial light intensity value at start of simulation
Randomize
LERandomNumber = Int(((14 - 9 + 1) * Rnd + 9) * 1000)
lbl_LightIntensity.Caption = LERandomNumber


readings = False
lightIntensity = False
Ppointer = 0
Lpointer = 10
percent = 0
simFinished = False


End Sub

Private Sub tmr_LightIntensity_Timer()
'Event that generates random light intensity value every 10 seconds.
If txt_LightIntensity.Text = "" Then
    If txt_readings.Text = "" Then
        Randomize
        LERandomNumber = Int(((14 - 9 + 1) * Rnd + 9) * 1000)
        lbl_LightIntensity.Caption = LERandomNumber
    End If
End If

End Sub
```

## Frm_Chi²:




```
Dim TOHtotal As Integer
Dim BOHtotal As Integer

Dim TOHobserved As Integer
Dim BOHobserved As Integer
Dim expected As Double
Dim TOHdifference As Double
Dim BOHdifference As Double
Dim TOHDiffereceSquared As Double
Dim BOHDiffereceSquared As Double
Dim TOHDSoverE As Double
Dim BOHDSoverE As Double
Dim CHI As Double

Dim verify1 As Boolean
Dim coorect As Integer
Dim dof As Integer
Dim check As Boolean

Private Sub cmd_help_Click()
frm_Chi2_Help.Show
End Sub

Private Sub cmd_own_Click()
'Submitting your own results into the table
If stats = False Then
    For I = 1 tO 10
        If IsNumeric(results(I - 1)) = True Then
```

```
            results(I - 1).Locked = True
            sim1.readings(I) = results(I - 1).Text
            TOHtotal = TOHtotal + sim1.readings(I)
        Else
            MsgBox "Please enter all the values into the table", , "Error"
            Exit Sub
        End If
    Next I

    For I = 11 tO 20
        If IsNumeric(results(I - 1)) = True Then
            results(I - 1).Locked = True
            sim1.readings(I) = results(I - 1).Text
            BOHtotal = BOHtotal + sim1.readings(I)
        Else
            MsgBox "Please enter all the values into the table", , "Error"
            Exit Sub
        End If
    Next I
    'works out all the stats for the user's values to be compared too
    stats = True
    TOHobserved = TOHtotal
    BOHobserved = BOHtotal
    expected = Round((TOHobserved + BOHobserved) / 2, 2)
    TOHdifference = TOHobserved - expected
    BOHdifference = BOHobserved - expected
    TOHDiffereceSquared = Round(TOHdifference * TOHdifference, 2)
    BOHDiffereceSquared = Round(BOHdifference * BOHdifference, 2)
    TOHDSoverE = Round(TOHDiffereceSquared / expected, 2)
    BOHDSoverE = Round(BOHDiffereceSquared / expected, 2)
    CHI = Round(TOHDSoverE + BOHDSoverE, 2)
    verify1 = False
End If

End Sub

Private Sub cmd_check_Click()

correct = 0

'Top of Hill verification
If correct <> 7 Then
    If Val(lbl_TOHo.Text) = TOHobserved Then
        lbl_TOHo.BackColor = &HFF00&
        lbl_TOHo.Locked = True
        If Val(lbl_TOHe.Text) = expected Then
            lbl_TOHe.BackColor = &HFF00&
            lbl_TOHe.Locked = True
            If Val(lbl_TOHoe.Text) = TOHdifference Then
                lbl_TOHoe.BackColor = &HFF00&
                lbl_TOHoe.Locked = True
```

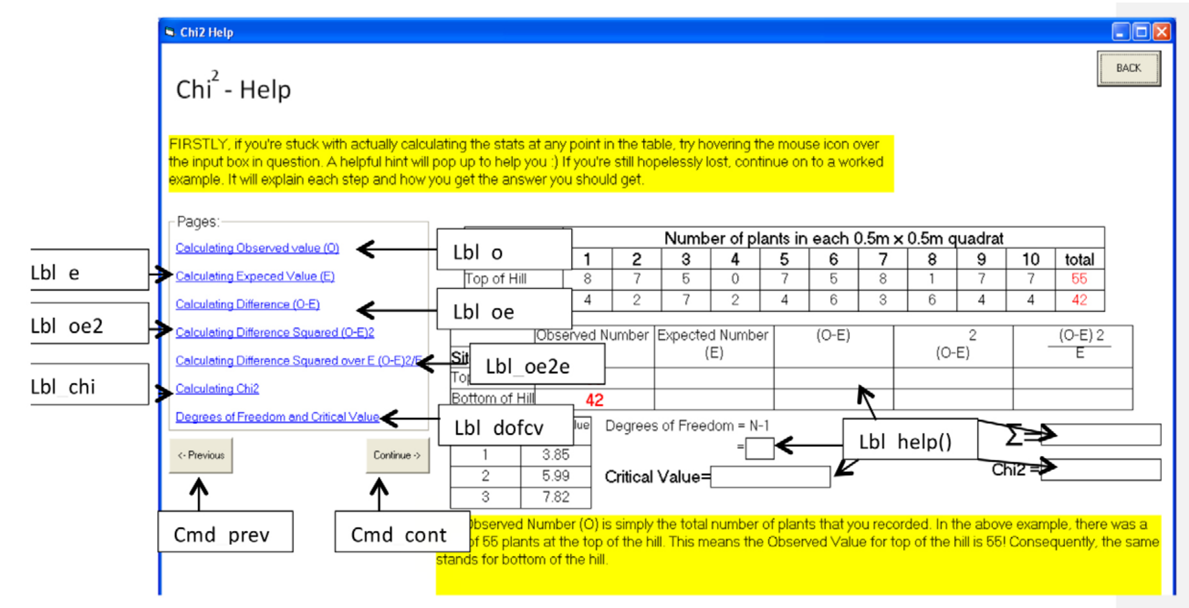
```
                If Val(lbl_TOHoe2.Text) = TOHDiffereceSquared Then
                    lbl_TOHoe2.BackColor = &HFF00&
                    lbl_TOHoe2.Locked = True
                    If Val(lbl_TOHoe2E.Text) = TOHDSoverE Then
                        lbl_TOHoe2E.BackColor = &HFF00&
                        lbl_TOHoe2E.Locked = True
                        correct = correct + 1
                    Else
                        lbl_TOHoe2E.BackColor = &HFF&
                        correct = 0
                    End If
                Else
                    lbl_TOHoe2.BackColor = &HFF&
                    correct = 0
                End If
            Else
                lbl_TOHoe.BackColor = &HFF&
                correct = 0
            End If
        Else
            lbl_TOHe.BackColor = &HFF&
            correct = 0
        End If
    Else
        lbl_TOHo.BackColor = &HFF&
        correct = 0
    End If

    'Bottom of Hill verification

    If Val(lbl_BOHo.Text) = BOHobserved Then
        lbl_BOHo.BackColor = &HFF00&
        lbl_BOHo.Locked = True
        If Val(lbl_BOHe.Text) = expected Then
            lbl_BOHe.BackColor = &HFF00&
            lbl_BOHe.Locked = True
            If Val(lbl_BOHoe.Text) = BOHdifference Then
                lbl_BOHoe.BackColor = &HFF00&
                lbl_BOHoe.Locked = True
                If Val(lbl_BOHoe2.Text) = BOHDiffereceSquared Then
                    lbl_BOHoe2.BackColor = &HFF00&
                    lbl_BOHoe2.Locked = True
                    If Val(lbl_BOHoe2E.Text) = BOHDSoverE Then
                        lbl_BOHoe2E.BackColor = &HFF00&
                        lbl_BOHoe2E.Locked = True
                        correct = correct + 1
                    Else
                        lbl_BOHoe2E.BackColor = &HFF&
                        correct = 0
                    End If
                Else
```

```
                    lbl_BOHoe2.BackColor = &HFF&
                    correct = 0
                End If
            Else
                lbl_BOHoe.BackColor = &HFF&
                correct = 0
            End If
        Else
            lbl_BOHe.BackColor = &HFF&
            correct = 0
        End If
    Else
        lbl_BOHo.BackColor = &HFF&
    End If
    'chi2
    If Val(txt_chi.Text) = CHI Then
        txt_chi.BackColor = &HFF00&
        txt_chi.Locked = True
        correct = correct + 1
    Else
        txt_chi.BackColor = &HFF&
        correct = 0
    End If
    'degrees of freedom
    If Val(txt_dof.Text) = 1 Then
        txt_dof.BackColor = &HFF00&
        txt_dof.Locked = True
        correct = correct + 1
    Else
        txt_dof.BackColor = &HFF&
        correct = 0
    End If
    'Chi2 again
    If Val(txt_chi2.Text) = CHI Then
        txt_chi2.BackColor = &HFF00&
        txt_chi2.Locked = True
        correct = correct + 1
    Else
        txt_chi2.BackColor = &HFF&
        correct = 0
    End If
    'critical value
    If Val(txt_cv.Text) = 3.85 Then
        txt_cv.BackColor = &HFF00&
        txt_cv.Locked = True
        correct = correct + 1
    Else
        txt_cv.BackColor = &HFF&
        correct = 0
    End If
    'evaluation text box
```

```
    If correct = 7 Then
        If CHI < 3.85 Then
            frm_less.Visible = True
        Else
            frm_greater.Visible = True
        End If
    End If

Else
End If

End Sub

Private Sub Command1_Click()
frm_error.Visible = False
End Sub

Private Sub cmd_back_Click()
Unload me
MainMenu.Show
End Sub

Private Sub cmd_rnd_Click()
'generating random results to be displayed in the results table
If stats = False Then
    For I = 1 tO 10
        Randomize
        results(I - 1).Text = Int(Rnd(1) * 9)
        sim1.readings(I) = results(I - 1)
        results(I - 1).Locked = True
        TOHtotal = TOHtotal + sim1.readings(I)
    Next I
    lbl_TOHtotal.Caption = TOHtotal
    'calculating stats for comparison with users data
    For I = 11 tO 20
        Randomize
        results(I - 1).Text = Int(Rnd(1) * 9)
        sim1.readings(I) = results(I - 1)
        results(I - 1).Locked = True
        BOHtotal = BOHtotal + sim1.readings(I)
    Next I
    lbl_BOHtotal.Caption = BOHtotal
    'Calculting stats for comparison with users data
    stats = True
    TOHobserved = TOHtotal
    BOHobserved = BOHtotal
    expected = Round((TOHobserved + BOHobserved) / 2, 2)
    TOHdifference = TOHobserved - expected
    BOHdifference = BOHobserved - expected
    TOHDiffereceSquared = Round(TOHdifference * TOHdifference, 2)
    BOHDiffereceSquared = Round(BOHdifference * BOHdifference, 2)
```

```
    TOHDSoverE = Round(TOHDiffereceSquared / expected, 2)
    BOHDSoverE = Round(BOHDiffereceSquared / expected, 2)
    CHI = Round(TOHDSoverE + BOHDSoverE, 2)
    verify1 = False
End If
End Sub

Private Sub Form_Load()
TOHtotal = 0
BOHtotal = 0

If stats = True Then
    'filling in results table
   For I = 1 tO 10
        results(I - 1).Text = sim1.readings(I)
        TOHtotal = TOHtotal + sim1.readings(I)
        results(I - 1).Locked = True
    Next I
    lbl_TOHtotal.Caption = TOHtotal

    For y = 11 tO 20
        results(y - 1).Text = sim1.readings(y)
        BOHtotal = BOHtotal + sim1.readings(y)
        results(I - 1).Locked = True
    Next y
    lbl_BOHtotal.Caption = BOHtotal
    'calculating stats for comparison with users data
    TOHobserved = TOHtotal
    BOHobserved = BOHtotal
    expected = Round((TOHobserved + BOHobserved) / 2, 2)
    TOHdifference = TOHobserved - expected
    BOHdifference = BOHobserved - expected
    TOHDiffereceSquared = Round(TOHdifference * TOHdifference, 2)
    BOHDiffereceSquared = Round(BOHdifference * BOHdifference, 2)
    TOHDSoverE = Round(TOHDiffereceSquared / expected, 2)
    BOHDSoverE = Round(BOHDiffereceSquared / expected, 2)
    CHI = Round(TOHDSoverE + BOHDSoverE, 2)
    verify1 = False
    cmd_own.Visible = False
    cmd_rnd.Visible = False
End If

If stats = False Then
    frm_error.Visible = True
End If

End Sub
```

## Frm_Chi2_Help:




```
Dim page As Integer

Private Sub cmd_cont_Click()
If page <> 6 Then
    page = page + 1
    Select Case page
        Case 0
            lbl_BOHtotal.ForeColor = vbRed
            lbl_TOHtotal.ForeColor = vbRed
            Text1.Text = "The Observed Number (O) is simply the total number of plants that you recorded. In the above example, there was a total of 55 plants at the top of the hill. This means the Observed Value for top of the hill is 55! Consequently, the same stands for bottom of the hill."
            For I = 0 tO 13
                lbl_help(I).Caption = ""
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
            lbl_help(0).ForeColor = vbRed
            lbl_help(1).ForeColor = vbRed
        Case 1
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = "The Expected Number (E) is the mean of the two Observed Numbers. You calculate the mean by adding all the numbers together and then dividing this result by how many number you had in total. In this example, the mean would equal 48.5. " & vbNewLine & "55+42 =97" & vbNewLine & "97/2 = 48.5" & vbNewLine & "IMPORTANT! Biology likes values to 2 decimal places! This example doesn't show it but beware! You hath been warned!"
            For I = 0 tO 13
                lbl_help(I).Caption = ""
```

```
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = "48.5"
            lbl_help(6).Caption = "48.5"
            lbl_help(2).ForeColor = vbRed
            lbl_help(6).ForeColor = vbRed
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
        Case 2
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = "(O-E) is calculated by working out the difference
between the value of E and the value of O. This is done by subtracting the
value of E from the value of O." & vbNewLine & "For Top of Hill: 55-48.5 = 6.5"
& vbNewLine & "For Bottom of Hill: 42-48.5 = -6.5"
            For I = 0 tO 13
                lbl_help(I).Caption = ""
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = "48.5"
            lbl_help(6).Caption = "48.5"
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
            lbl_help(3).Caption = "6.5"
            lbl_help(7).Caption = "-6.5"
            lbl_help(3).ForeColor = vbRed
            lbl_help(7).ForeColor = vbRed
        Case 3
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = "For this bit, all you need to do is square the
difference. This is important as squaring a number will also get rid of any
negative signs (if there are any)." & vbNewLine & "For Top of Hill: (6.5)^2 =
42.25" & vbNewLine & "For Bottom of Hill: (-6.5)^2 = 42.25"
            For I = 0 tO 13
                lbl_help(I).Caption = ""
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = "48.5"
            lbl_help(6).Caption = "48.5"
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
            lbl_help(3).Caption = "6.5"
            lbl_help(7).Caption = "-6.5"
            lbl_help(4).Caption = "42.25"
            lbl_help(8).Caption = "42.25"
            lbl_help(4).ForeColor = vbRed
            lbl_help(8).ForeColor = vbRed
        Case 4
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
```

```
            Text1.Text = "This last little bit is taking the squared number and
dividing it by E. " & vbNewLine & "For Top of Hill: 42.25 / 48.5 = 0.8711340206"
& vbNewLine & "For Bottom of Hill: 42.25 / 48.5 = 0.8711340206" & vbNewLine &
"IMPORTANT!!! Biology likes values to 2 decimal places! Therefore, the value we
would generally accept would be 0.87"
            For I = 0 tO 13
                lbl_help(I).Caption = ""
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = "48.5"
            lbl_help(6).Caption = "48.5"
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
            lbl_help(3).Caption = "6.5"
            lbl_help(7).Caption = "-6.5"
            lbl_help(4).Caption = "42.25"
            lbl_help(8).Caption = "42.25"
            lbl_help(5).Caption = "0.87"
            lbl_help(9).Caption = "0.87"
            lbl_help(5).ForeColor = vbRed
            lbl_help(9).ForeColor = vbRed
        Case 5
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = "The value of Chi2 is the sum of the difference
squared over expected results." & vbNewLine & "The weird symbol is the greek
letter sigma and stands for 'Sum of'" & vbNewLine & "In this example, the sum
will be:" & vbNewLine & "0.87+0.87 = 1.74"
            For I = 0 tO 13
                lbl_help(I).Caption = ""
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = "48.5"
            lbl_help(6).Caption = "48.5"
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
            lbl_help(3).Caption = "6.5"
            lbl_help(7).Caption = "-6.5"
            lbl_help(4).Caption = "42.25"
            lbl_help(8).Caption = "42.25"
            lbl_help(5).Caption = "0.87"
            lbl_help(9).Caption = "0.87"
            lbl_help(10).Caption = "1.74"
            lbl_help(11).Caption = "1.74"
            lbl_help(10).ForeColor = vbRed
            lbl_help(11).ForeColor = vbRed

        Case 6
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = "Degrees of Freedom can be calculated with N-1, where
```

```
N is equal to the number of sets of data you have. We have two data sets (Top
of Hill and Bottom of Hill) so N=2." & vbewline & "This means the Degrees of
Freedom is equal to 1 (2-1=1)" & vbNewLine & "We obtain the Critical Value by
using the data table (this will be given to you)." & vbNewLine & "At Degrees of
Freedom: 1, the Critical Value = 3.85" & vbNewLine & "And that's all there is
to it :D"
            For I = 0 tO 13
                lbl_help(I).Caption = ""
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = "48.5"
            lbl_help(6).Caption = "48.5"
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
            lbl_help(3).Caption = "6.5"
            lbl_help(7).Caption = "-6.5"
            lbl_help(4).Caption = "42.25"
            lbl_help(8).Caption = "42.25"
            lbl_help(5).Caption = "0.87"
            lbl_help(9).Caption = "0.87"
            lbl_help(10).Caption = "1.74"
            lbl_help(11).Caption = "1.74"
            lbl_help(12).Caption = "3.85"
            lbl_help(13).Caption = "1"
            lbl_help(12).ForeColor = vbRed
            lbl_help(13).ForeColor = vbRed

    End Select
End If

End Sub

Private Sub Command2_Click()
Unload me
frm_Chi2.Show
End Sub


Private Sub cmd_prev_Click()
If page <> 0 Then
    page = page - 1

    Select Case page
                Case 0
            lbl_BOHtotal.ForeColor = vbRed
            lbl_TOHtotal.ForeColor = vbRed
            Text1.Text = "The Observed Number (O) is simply the total number of
plants that you recorded. In the above example, there was a total of 55 plants
at the top of the hill. This means the Observed Value for top of the hill is
55! Consequently, the same stands for bottom of the hill."
            For I = 0 tO 13
```

```
                lbl_help(I).Caption = ""
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
            lbl_help(0).ForeColor = vbRed
            lbl_help(1).ForeColor = vbRed
        Case 1
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = "The Expected Number (E) is the mean of the two
Observed Numbers. You calculate the mean by adding all the numbers together and
then dividing this result by how many number you had in total. In this example,
the mean would equal 48.5. " & vbNewLine & "55+42 =97" & vbNewLine & "97/2 =
48.5" & vbNewLine & "IMPORTANT! Biology likes values to 2 decimal places! This
example doesn't show it but beware! You hath been warned!"
            For I = 0 tO 13
                lbl_help(I).Caption = ""
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = "48.5"
            lbl_help(6).Caption = "48.5"
            lbl_help(2).ForeColor = vbRed
            lbl_help(6).ForeColor = vbRed
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
        Case 2
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = "(O-E) is calculated by working out the difference
between the value of E and the value of O. This is done by subtracting the
value of E from the value of O." & vbNewLine & "For Top of Hill: 55-48.5 = 6.5"
& vbNewLine & "For Bottom of Hill: 42-48.5 = -6.5"
            For I = 0 tO 13
                lbl_help(I).Caption = ""
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = "48.5"
            lbl_help(6).Caption = "48.5"
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
            lbl_help(3).Caption = "6.5"
            lbl_help(7).Caption = "-6.5"
            lbl_help(3).ForeColor = vbRed
            lbl_help(7).ForeColor = vbRed
        Case 3
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = "For this bit, all you need to do is square the
difference. This is important as squaring a number will also get rid of any
negative signs (if there are any)." & vbNewLine & "For Top of Hill: (6.5)^2 =
```

```
42.25” & vbNewLine & “For Bottom of Hill: (-6.5)^2 = 42.25”
            For I = 0 tO 13
                lbl_help(I).Caption = “”
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = “48.5”
            lbl_help(6).Caption = “48.5”
            lbl_help(0).Caption = “55”
            lbl_help(1).Caption = “42”
            lbl_help(3).Caption = “6.5”
            lbl_help(7).Caption = “-6.5”
            lbl_help(4).Caption = “42.25”
            lbl_help(8).Caption = “42.25”
            lbl_help(4).ForeColor = vbRed
            lbl_help(8).ForeColor = vbRed
        Case 4
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = “This last little bit is taking the squared number and
dividing it by E. “ & vbNewLine & “For Top of Hill: 42.25 / 48.5 = 0.8711340206”
& vbNewLine & “For Bottom of Hill: 42.25 / 48.5 = 0.8711340206” & vbNewLine &
“IMPORTANT!!! Biology likes values to 2 decimal places! Therefore, the value we
would generally accept would be 0.87”
            For I = 0 tO 13
                lbl_help(I).Caption = “”
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = “48.5”
            lbl_help(6).Caption = “48.5”
            lbl_help(0).Caption = “55”
            lbl_help(1).Caption = “42”
            lbl_help(3).Caption = “6.5”
            lbl_help(7).Caption = “-6.5”
            lbl_help(4).Caption = “42.25”
            lbl_help(8).Caption = “42.25”
            lbl_help(5).Caption = “0.87”
            lbl_help(9).Caption = “0.87”
            lbl_help(5).ForeColor = vbRed
            lbl_help(9).ForeColor = vbRed
        Case 5
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = “The value of Chi2 is the sum of the difference
squared over expected results.” & vbNewLine & “The weird symbol is the greek
letter sigma and stands for ‘Sum of’” & vbNewLine & “In this example, the sum
will be:” & vbNewLine & “0.87+0.87 = 1.74”
            For I = 0 tO 13
                lbl_help(I).Caption = “”
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = “48.5”
```

```
            lbl_help(6).Caption = "48.5"
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
            lbl_help(3).Caption = "6.5"
            lbl_help(7).Caption = "-6.5"
            lbl_help(4).Caption = "42.25"
            lbl_help(8).Caption = "42.25"
            lbl_help(5).Caption = "0.87"
            lbl_help(9).Caption = "0.87"
            lbl_help(10).Caption = "1.74"
            lbl_help(11).Caption = "1.74"
            lbl_help(10).ForeColor = vbRed
            lbl_help(11).ForeColor = vbRed
        Case 6
            lbl_BOHtotal.ForeColor = vbBlack
            lbl_TOHtotal.ForeColor = vbBlack
            Text1.Text = "Degrees of Freedom can be calculated with N-1, where
N is equal to the number of sets of data you have. We have two data sets (Top
of Hill and Bottom of Hill) so N=2." & vbewline & "This means the Degrees of
Freedom is equal to 1 (2-1=1)" & vbNewLine & "We obtain the Critical Value by
using the data table (this will be given to you)." & vbNewLine & "At Degrees of
Freedom: 1, the Critical Value = 3.85" & vbNewLine & "And that's all there is
to it :D"
            For I = 0 tO 13
                lbl_help(I).Caption = ""
                lbl_help(I).ForeColor = vbBlack
            Next I
            lbl_help(2).Caption = "48.5"
            lbl_help(6).Caption = "48.5"
            lbl_help(0).Caption = "55"
            lbl_help(1).Caption = "42"
            lbl_help(3).Caption = "6.5"
            lbl_help(7).Caption = "-6.5"
            lbl_help(4).Caption = "42.25"
            lbl_help(8).Caption = "42.25"
            lbl_help(5).Caption = "0.87"
            lbl_help(9).Caption = "0.87"
            lbl_help(10).Caption = "1.74"
            lbl_help(11).Caption = "1.74"
            lbl_help(12).Caption = "3.85"
            lbl_help(13).Caption = "1"
            lbl_help(12).ForeColor = vbRed
            lbl_help(13).ForeColor = vbRed
    End Select
End If
End Sub

Private Sub Form_Load()
page = 0
lbl_BOHtotal.ForeColor = vbRed
lbl_TOHtotal.ForeColor = vbRed
```

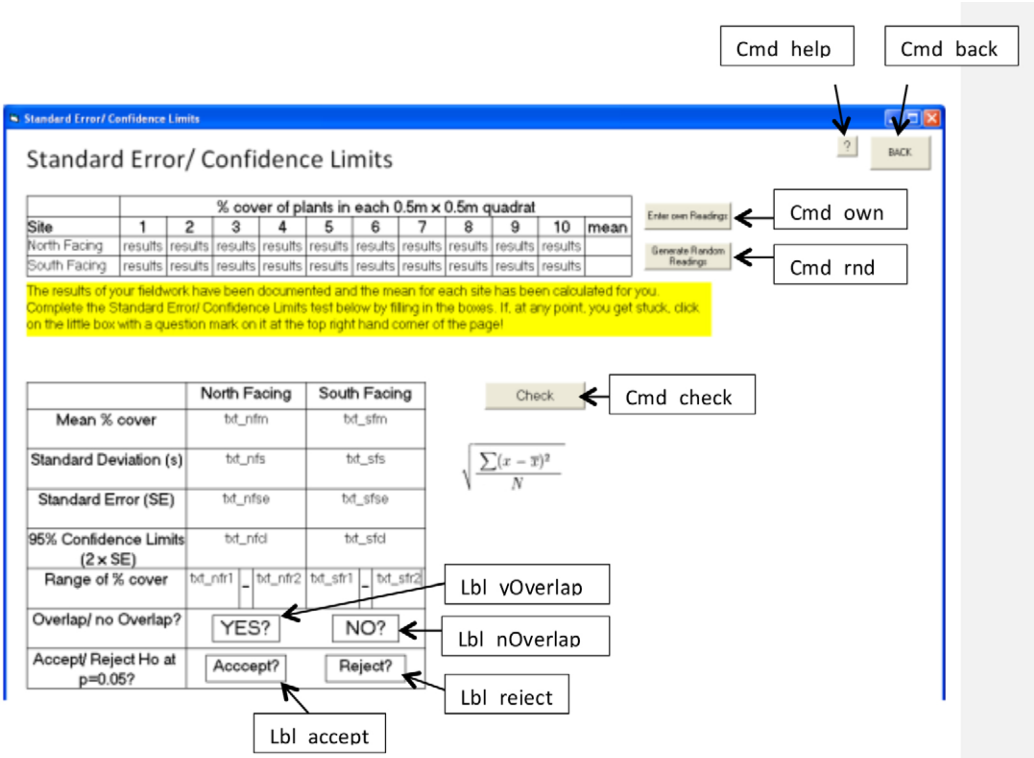
```
Text1.Text = "The Observed Number (O) is simply the total number of plants that
you recorded. In the above example, there was a total of 55 plants at the top
of the hill. This means the Observed Value for top of the hill is 55!
Consequently, the same stands for bottom of the hill."
For I = 0 tO 13
    lbl_help(I).Caption = ""
    lbl_help(I).ForeColor = vbBlack
Next I
lbl_help(0).Caption = "55"
lbl_help(1).Caption = "42"
lbl_help(0).ForeColor = vbRed
lbl_help(1).ForeColor = vbRed
End Sub

Private Sub lbl_e_Click()
page = 1
lbl_BOHtotal.ForeColor = vbBlack
lbl_TOHtotal.ForeColor = vbBlack
Text1.Text = "The Expected Number (E) is the mean of the two Observed Numbers.
You calculate the mean by adding all the numbers together and then dividing
this result by how many number you had in total. In this example, the mean
would equal 48.5. " & vbNewLine & "55+42 =97" & vbNewLine & "97/2 = 48.5" &
vbNewLine & "IMPORTANT! Biology likes values to 2 decimal places! This example
doesn't show it but beware! You hath been warned!"
For I = 0 tO 13
    lbl_help(I).Caption = ""
    lbl_help(I).ForeColor = vbBlack
Next I
lbl_help(2).Caption = "48.5"
lbl_help(6).Caption = "48.5"
lbl_help(2).ForeColor = vbRed
lbl_help(6).ForeColor = vbRed
lbl_help(0).Caption = "55"
lbl_help(1).Caption = "42"
End Sub

Private Sub lbl_oe_Click()
page = 2
lbl_BOHtotal.ForeColor = vbBlack
lbl_TOHtotal.ForeColor = vbBlack
Text1.Text = "(O-E) is calculated by working out the difference between the
value of E and the value of O. This is done by subtracting the value of E from
the value of O." & vbNewLine & "For Top of Hill: 55-48.5 = 6.5" & vbNewLine &
"For Bottom of Hill: 42-48.5 = -6.5"
For I = 0 tO 13
    lbl_help(I).Caption = ""
    lbl_help(I).ForeColor = vbBlack
Next I
lbl_help(2).Caption = "48.5"
lbl_help(6).Caption = "48.5"
lbl_help(0).Caption = "55"
```

```
lbl_help(1).Caption = "42"
lbl_help(3).Caption = "6.5"
lbl_help(7).Caption = "-6.5"
lbl_help(3).ForeColor = vbRed
lbl_help(7).ForeColor = vbRed
End Sub

Private Sub lbl_oe2_Click()
page = 3
lbl_BOHtotal.ForeColor = vbBlack
lbl_TOHtotal.ForeColor = vbBlack
Text1.Text = "For this bit, all you need to do is square the difference. This
is important as squaring a number will also get rid of any negative signs (if
there are any)." & vbNewLine & "For Top of Hill: (6.5)^2 = 42.25" & vbNewLine &
"For Bottom of Hill: (-6.5)^2 = 42.25"
For I = 0 tO 13
    lbl_help(I).Caption = ""
    lbl_help(I).ForeColor = vbBlack
Next I
lbl_help(2).Caption = "48.5"
lbl_help(6).Caption = "48.5"
lbl_help(0).Caption = "55"
lbl_help(1).Caption = "42"
lbl_help(3).Caption = "6.5"
lbl_help(7).Caption = "-6.5"
lbl_help(4).Caption = "42.25"
lbl_help(8).Caption = "42.25"
lbl_help(4).ForeColor = vbRed
lbl_help(8).ForeColor = vbRed
End Sub

Private Sub lbl_oe2e_Click()
page = 4
lbl_BOHtotal.ForeColor = vbBlack
lbl_TOHtotal.ForeColor = vbBlack
Text1.Text = "This last little bit is taking the squared number and dividing it
by E. " & vbNewLine & "For Top of Hill: 42.25 / 48.5 = 0.8711340206" &
vbNewLine & "For Bottom of Hill: 42.25 / 48.5 = 0.8711340206" & vbNewLine &
"IMPORTANT!!! Biology likes values to 2 decimal places! Therefore, the value we
would generally accept would be 0.87"
For I = 0 tO 13
    lbl_help(I).Caption = ""
    lbl_help(I).ForeColor = vbBlack
Next I
lbl_help(2).Caption = "48.5"
lbl_help(6).Caption = "48.5"
lbl_help(0).Caption = "55"
lbl_help(1).Caption = "42"
lbl_help(3).Caption = "6.5"
lbl_help(7).Caption = "-6.5"
lbl_help(4).Caption = "42.25"
```

```
lbl_help(8).Caption = "42.25"
lbl_help(5).Caption = "0.87"
lbl_help(9).Caption = "0.87"
lbl_help(5).ForeColor = vbRed
lbl_help(9).ForeColor = vbRed

End Sub

Private Sub lbl_chi_Click()
page = 5
lbl_BOHtotal.ForeColor = vbBlack
lbl_TOHtotal.ForeColor = vbBlack
Text1.Text = "The value of Chi2 is the sum of the difference squared over
expected results." & vbNewLine & "The weird symbol is the greek letter sigma
and stands for 'Sum of'" & vbNewLine & "In this example, the sum will be:" &
vbNewLine & "0.87+0.87 = 1.74"
For I = 0 tO 13
    lbl_help(I).Caption = ""
    lbl_help(I).ForeColor = vbBlack
Next I
lbl_help(2).Caption = "48.5"
lbl_help(6).Caption = "48.5"
lbl_help(0).Caption = "55"
lbl_help(1).Caption = "42"
lbl_help(3).Caption = "6.5"
lbl_help(7).Caption = "-6.5"
lbl_help(4).Caption = "42.25"
lbl_help(8).Caption = "42.25"
lbl_help(5).Caption = "0.87"
lbl_help(9).Caption = "0.87"
lbl_help(10).Caption = "1.74"
lbl_help(11).Caption = "1.74"
lbl_help(10).ForeColor = vbRed
lbl_help(11).ForeColor = vbRed


End Sub

Private Sub lbl_dofcv_Click()
page = 6
lbl_BOHtotal.ForeColor = vbBlack
lbl_TOHtotal.ForeColor = vbBlack
Text1.Text = "Degrees of Freedom can be calculated with N-1, where N is equal
to the number of sets of data you have. We have two data sets (Top of Hill and
Bottom of Hill) so N=2." & vbewline & "This means the Degrees of Freedom is
equal to 1 (2-1=1)" & vbNewLine & "We obtain the Critical Value by using the
data table (this will be given to you)." & vbNewLine & "At Degrees of Freedom:
1, the Critical Value = 3.85" & vbNewLine & "And that's all there is to it :D"
For I = 0 tO 13
    lbl_help(I).Caption = ""
    lbl_help(I).ForeColor = vbBlack
```

```
Next I
lbl_help(2).Caption = "48.5"
lbl_help(6).Caption = "48.5"
lbl_help(0).Caption = "55"
lbl_help(1).Caption = "42"
lbl_help(3).Caption = "6.5"
lbl_help(7).Caption = "-6.5"
lbl_help(4).Caption = "42.25"
lbl_help(8).Caption = "42.25"
lbl_help(5).Caption = "0.87"
lbl_help(9).Caption = "0.87"
lbl_help(10).Caption = "1.74"
lbl_help(11).Caption = "1.74"
lbl_help(12).Caption = "3.85"
lbl_help(13).Caption = "1"
lbl_help(12).ForeColor = vbRed
lbl_help(13).ForeColor = vbRed


End Sub

Private Sub lbl_o_Click()
page = 0
lbl_BOHtotal.ForeColor = vbRed
lbl_TOHtotal.ForeColor = vbRed
Text1.Text = "The Observed Number (O) is simply the total number of plants that
you recorded. In the above example, there was a total of 55 plants at the top
of the hill. This means the Observed Value for top of the hill is 55!
Consequently, the same stands for bottom of the hill."
For I = 0 tO 13
    lbl_help(I).Caption = ""
    lbl_help(I).ForeColor = vbBlack
Next I
lbl_help(0).Caption = "55"
lbl_help(1).Caption = "42"
lbl_help(0).ForeColor = vbRed
lbl_help(1).ForeColor = vbRed
End Sub
```

## Frm_StandardError:




```
Dim NFmean As Double
Dim SFmean As Double
Dim NFStandardDeviation As Double
Dim NFStandardError As Double
Dim SFStandardDeviation As Double
Dim SFStandardError As Double
Dim ConfidenceLimits As Double
Dim overlap As Boolean
Dim hypothesis As Boolean

Dim Yclick As Boolean
Dim Nclick As Boolean
Dim Aclick As Boolean
Dim Rclick As Boolean

Dim range1 As Double
Dim range2 As Double
Dim range3 As Double
Dim range4 As Double



Function NFStdDev(mean As Double) As Double
'A function to calculate the Standard Deviation from the North Facing values
    Dim Value(15) As Double
    Dim Temp As Double
    Dim Temp2 As Double

    NFStdDev = 0
```

```
    For I = 1 tO 15
        Value(I) = 0
    Next I

    Temp = 0

    For I = 1 tO 10
        Value(I) = sim1.readings(I)
        Temp = (Value(I) - mean) ^ 2
        Temp2 = Temp2 + Temp
    Next I
    Temp2 = Temp2 / 10
    NFStdDev = Round(Sqr(Temp2), 2)

End Function

Function SFStdDev(mean As Double) As Double
'A function to calculate the Standard Deviation of the South Facing values
    Dim Value(20) As Double
    Dim Temp As Double
    Dim Temp2 As Double

    SFStdDev = 0
    For y = 11 tO 20
        Value(I) = 0
    Next y

    Temp = 0

    For y = 11 tO 20
        Value(y) = sim1.readings(y)
        Temp = (Value(y) - mean) ^ 2
        Temp2 = Temp2 + Temp
    Next y
    Temp2 = Temp2 / 10
    SFStdDev = Round(Sqr(Temp2), 2)

End Function

Private Sub cmd_help_Click()
frm_StandardError_help.Show
End Sub

Private Sub cmd_own_Click()
If stats = False Then
'Registering users own data into the results table
    For I = 1 tO 10
        sim1.readings(I) = results(I - 1).Text
        NFmean = NFmean + sim1.readings(I)
        results(I - 1).Locked = True
    Next I
```

```
    lbl_nfmean.Caption = NFmean / 10

    For y = 11 tO 20
        sim1.readings(y) = results(y - 1).Text
        SFmean = SFmean + sim1.readings(y)
        results(I - 1).Locked = True
    Next y
    lbl_sfmean.Caption = SFmean / 10
'calculating stats for comparison with users own data
    NFStandardDeviation = Round(NFStdDev(Val(lbl_nfmean)), 2)
    NFStandardError = Round(NFStandardDeviation / Sqr(10), 2)

    SFStandardDeviation = Round(SFStdDev(Val(lbl_sfmean)), 2)
    SFStandardError = Round(SFStandardDeviation / Sqr(10), 2)

    range1 = (Val(lbl_nfmean) - (2 * NFStandardError))
    range2 = (Val(lbl_nfmean) + (2 * NFStandardError))
    range3 = (Val(lbl_sfmean) - (2 * SFStandardError))
    range4 = (Val(lbl_sfmean) + (2 * SFStandardError))
    stats = True
End If
End Sub

Private Sub cmd_rnd_Click()
If stats = False Then
'generating a random set of results and displaying them iun the results table
    For I = 1 tO 10
        Randomize
        results(I - 1).Text = Int(Rnd(1) * 20) * 5
        sim1.readings(I) = results(I - 1).Text
        results(I - 1).Locked = True
        NFmean = NFmean + sim1.readings(I)
    Next I
    lbl_nfmean.Caption = NFmean / 10

    For I = 11 tO 20
        Randomize
        results(I - 1).Text = Int(Rnd(1) * 20) * 5
        sim1.readings(I) = results(I - 1).Text
        results(I - 1).Locked = True
        SFmean = SFmean + sim1.readings(I)
    Next I
    lbl_sfmean.Caption = SFmean / 10
'calculating stats for comparison with users own values
    NFStandardDeviation = Round(NFStdDev(Val(lbl_nfmean)), 2)
    NFStandardError = Round(NFStandardDeviation / Sqr(10), 2)

    SFStandardDeviation = Round(SFStdDev(Val(lbl_sfmean)), 2)
    SFStandardError = Round(SFStandardDeviation / Sqr(10), 2)
```

```
    range1 = (Val(lbl_nfmean) - (2 * NFStandardError))
    range2 = (Val(lbl_nfmean) + (2 * NFStandardError))
    range3 = (Val(lbl_sfmean) - (2 * SFStandardError))
    range4 = (Val(lbl_sfmean) + (2 * SFStandardError))
    stats = True
End If

End Sub

Private Sub cmd_check_Click()
If stats = True Then
'checking all text boxes with the computer calculated stats to see if they are correct
'North Facing
    If Val(txt_nfm.Text) = Val(lbl_nfmean) Then
        txt_nfm.BackColor = &HFF00&
        txt_nfm.Locked = True
        If Val(txt_nfs.Text) = NFStandardDeviation Then
            txt_nfs.BackColor = &HFF00&
            txt_nfs.Locked = True
            If Val(txt_nfse.Text) = NFStandardError Then
                txt_nfse.BackColor = &HFF00&
                txt_nfse.Locked = True
                If Val(txt_nfcl.Text) = (2 * NFStandardError) Then
                    txt_nfcl.BackColor = &HFF00&
                    txt_nfcl.Locked = True
                    If Val(txt_nfr1.Text) = Val(range1) Then
                        If Val(txt_nfr2.Text) = Val(range2) Then
                            txt_nfr1.BackColor = &HFF00&
                            txt_nfr2.BackColor = &HFF00&
                            txt_nfr1.Locked = True
                            txt_nfr2.Locked = True
                            correct = correct + 1
                        Else
                            txt_nfr1.BackColor = &HFF&
                            txt_nfr2.BackColor = &HFF&
                        End If
                    Else
                        txt_nfr1.BackColor = &HFF&
                        txt_nfr2.BackColor = &HFF&
                    End If

                Else
                    txt_nfcl.BackColor = &HFF&
                End If
            Else
                txt_nfse.BackColor = &HFF&
            End If
        Else
            txt_nfs.BackColor = &HFF&
        End If
```

```
    Else
        txt_nfm.BackColor = &HFF&
    End If
'South Facing
    If Val(txt_sfm.Text) = Val(lbl_sfmean) Then
        txt_sfm.BackColor = &HFF00&
        txt_sfm.Locked = True
        If Val(txt_sfs.Text) = SFStandardDeviation Then
            txt_sfs.BackColor = &HFF00&
            txt_sfs.Locked = True
            If Val(txt_sfse.Text) = SFStandardError Then
                txt_sfse.BackColor = &HFF00&
                txt_sfse.Locked = True
                If Val(txt_sfcl.Text) = (2 * SFStandardError) Then
                    txt_sfcl.BackColor = &HFF00&
                    txt_sfcl.Locked = True
                    If Val(txt_sfr1.Text) = Val(range3) Then
                        If Val(txt_sfr2.Text) = range4 Then
                            txt_sfr1.BackColor = &HFF00&
                            txt_sfr2.BackColor = &HFF00&
                            txt_sfr1.Locked = True
                            txt_sfr2.Locked = True
                            correct = correct + 1
                        Else
                            txt_sfr1.BackColor = &HFF&
                            txt_sfr2.BackColor = &HFF&
                        End If
                    Else
                        txt_sfr1.BackColor = &HFF&
                        txt_sfr2.BackColor = &HFF&
                    End If

                Else
                    txt_sfcl.BackColor = &HFF&
                End If
            Else
                txt_sfse.BackColor = &HFF&
            End If
        Else
            txt_sfs.BackColor = &HFF&
        End If
    Else
        txt_sfm.BackColor = &HFF&
    End If
'working out if there is an overlap in the ranges
    If Val(txt_nfr1.Text) < Val(txt_sfr2.Text) And Val(txt_nfr2.Text) >
Val(txt_sfr1.Text) Then
        overlap = True
    ElseIf Val(txt_nfr2.Text) < Val(txt_sfr1.Text) And Val(txt_nfr2.Text) >
Val(txt_sfr1.Text) Then
        overlap = True
```

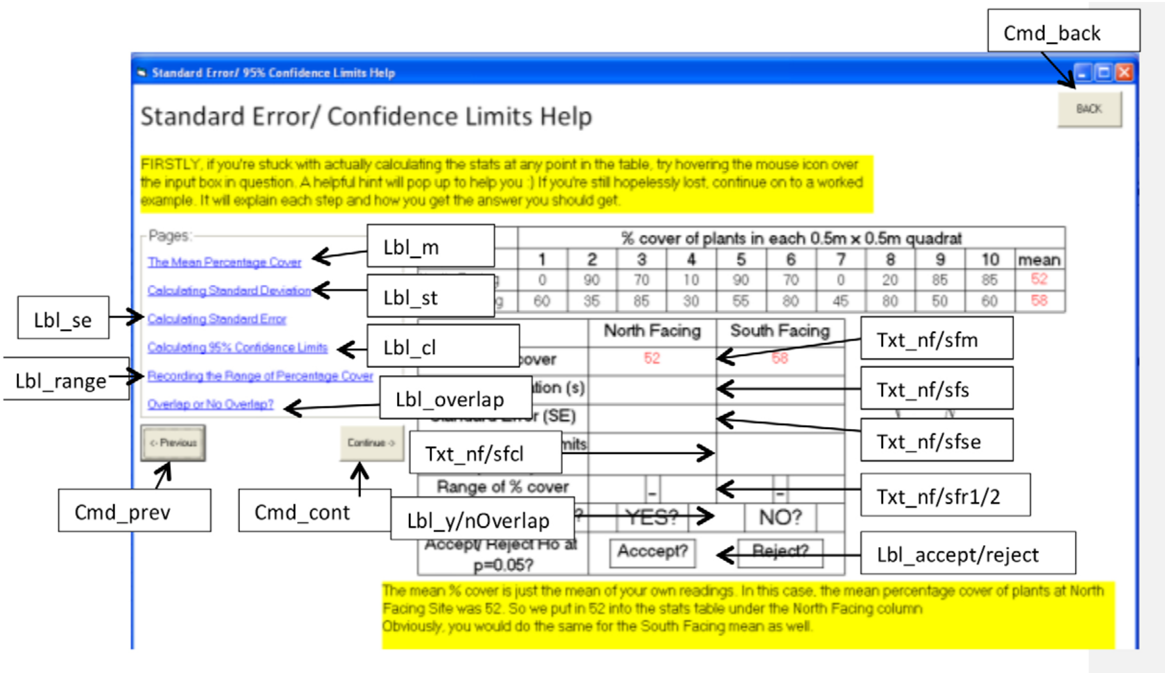
```
    Else
        overlap = False
    End If

    If correct <> 4 Then
'comparing the selected label with the overlap.
        If overlap = True And lbl_yOverlap.BackColor = &HFFFF& Then
            lbl_yOverlap.BackColor = &HFF00&
            lbl_yOverlap.Caption = "YES!"
            lbl_yOverlap.top = 8760
            lbl_yOverlap.Left = 4800
            lbl_nOverlap.Visible = False
            correct = correct + 1
        ElseIf overlap = False And lbl_nOverlap.BackColor = &HFFFF& Then
            lbl_nOverlap.BackColor = &HFF00&
            lbl_nOverlap.Caption = "NO!"
            lbl_nOverlap.top = 8760
            lbl_nOverlap.Left = 4800
            lbl_yOverlap.Visible = False
            correct = correct + 1
        Else
            lbl_nOverlap.BackColor = &HFF&
            lbl_yOverlap.BackColor = &HFF&
        End If
'depending on the overlap and selected label - compares uses response to
accepting or rejecting Null Hypothesis
        If lbl_yOverlap.BackColor = &HFF00& And lbl_accept.BackColor = &HFFFF&
Then
      lbl_accept.BackColor = &HFF00&
            lbl_accept.Caption = "Accepted!"
            lbl_accept.top = 9480
            lbl_accept.Left = 4680
            lbl_reject.Visible = False
            correct = correct + 1
        ElseIf lbl_nOverlap.BackColor = &HFF00& And lbl_reject.BackColor =
&HFFFF& Then
            lbl_reject.BackColor = &HFF00&
            lbl_reject.Caption = "Reject!"
            lbl_reject.top = 9480
            lbl_reject.Left = 4680
            lbl_accept.Visible = False
            correct = correct + 1
        Else
            lbl_accept.BackColor = &HFF&
            lbl_reject.BackColor = &HFF&
        End If
    End If
    If correct = 4 Then
    'working out which evaluation text box to display at the end of the stats
        If lbl_accept.Caption = "Accepted!" Then
```

```
            accept.Visible = True
        ElseIf lbl_reject.Caption = "Reject!" Then
            reject.Visible = True
        End If
    End If
End If








End Sub

Private Sub cmd_back_Click()
Unload me
MainMenu.Show
End Sub



Private Sub Command3_Click()
frm_error.Visible = False

End Sub

Private Sub Form_Load()
'initialising variables
NFStandardDeviation = 0
NFStandardError = 0
SFStandardDeviation = 0
SFStandardError = 0

Yclick = False
Nclick = False
Aclick = False
Rclick = False

If stats = True Then
    'loading north facing values
    For I = 1 tO 10
        results(I - 1).Text = sim1.readings(I)
        NFmean = NFmean + sim1.readings(I)
    Next I
    lbl_nfmean.Caption = NFmean / 10
    'loading South Facing values
    For y = 11 tO 20
        results(y - 1).Text = sim1.readings(y)
        SFmean = SFmean + sim1.readings(y)
```

```
    Next y
    lbl_sfmean.Caption = SFmean / 10
    'calculting stats for comparison with users inputted values
    NFStandardDeviation = Round(NFStdDev(Val(lbl_nfmean)), 2)
    NFStandardError = Round(NFStandardDeviation / Sqr(10), 2)

    SFStandardDeviation = Round(SFStdDev(Val(lbl_sfmean)), 2)
    SFStandardError = Round(SFStandardDeviation / Sqr(10), 2)

    range1 = (Val(lbl_nfmean) - (2 * NFStandardError))
    range2 = (Val(lbl_nfmean) + (2 * NFStandardError))
    range3 = (Val(lbl_sfmean) - (2 * SFStandardError))
    range4 = (Val(lbl_sfmean) + (2 * SFStandardError))
Else
    frm_error.Visible = True
End If
End Sub



Private Sub lbl_yOverlap_Click()
If Yclick = False Then
    lbl_yOverlap.BackColor = &HFFFF&
    lbl_nOverlap.BackColor = &HFFFFFF
    Yclick = True
    Nclick = False
End If
End Sub
Private Sub lbl_nOverlap_Click()
If Nclick = False Then
    lbl_nOverlap.BackColor = &HFFFF&
    lbl_yOverlap.BackColor = &HFFFFFF
    Nclick = True
    Yclick = False
End If
End Sub
Private Sub lbl_accept_Click()
If Aclick = False Then
    lbl_accept.BackColor = &HFFFF&
    lbl_reject.BackColor = &HFFFFFF
    Aclick = True
    Rclick = False
End If
End Sub
Private Sub lbl_reject_Click()
If Rclick = False Then
    lbl_reject.BackColor = &HFFFF&
    lbl_accept.BackColor = &HFFFFFF
    Rclick = True
    Aclick = False
```

```
End If
End Sub
```

## Frm_StandardError_Help:




```
Dim page As Integer

Private Sub cmd_cont_Click()
If page <> 5 Then
    page = page + 1
    Select Case page

    Case 0
    'starting help page
        lbl_nfmean.ForeColor = vbRed
        lbl_sfmean.ForeColor = vbRed
        txt_nfm.ForeColor = vbRed
        txt_sfm.ForeColor = vbRed
        txt_nfs.ForeColor = vbBlack
        txt_sfs.ForeColor = vbBlack
        txt_sfse.ForeColor = vbBlack
        txt_nfse.ForeColor = vbBlack
        txt_nfcl.ForeColor = vbBlack
        txt_sfcl.ForeColor = vbBlack
        txt_nfr1.ForeColor = vbBlack
        txt_nfr2.ForeColor = vbBlack
        txt_sfr1.ForeColor = vbBlack
        txt_sfr2.ForeColor = vbBlack
        lbl_yOverlap.ForeColor = vbBlack
        lbl_accept.ForeColor = vbBlack
```

```
        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = ""
        txt_sfs.Text = ""
        txt_sfse.Text = ""
        txt_nfse.Text = ""
        txt_nfcl.Text = ""
        txt_sfcl.Text = ""
        txt_nfr1.Text = ""
        txt_nfr2.Text = ""
        txt_sfr1.Text = ""
        txt_sfr2.Text = ""

        txt_help.Text = "The mean % cover is just the mean of your own
readings. In this case, the mean percentage cover of plants at North " &
vbNewLine & "Facing Site was 52. So we put in 52 into the stats table under the
North Facing column" & vbNewLine & "Obviously, you would do the same for the
South Facing mean as well."

    Case 1
    'second help page
        lbl_nfmean.ForeColor = vbBlack
        lbl_sfmean.ForeColor = vbBlack
        txt_nfm.ForeColor = vbBlack
        txt_sfm.ForeColor = vbBlack
        txt_nfs.ForeColor = vbRed
        txt_sfs.ForeColor = vbRed
        txt_sfse.ForeColor = vbBlack
        txt_nfse.ForeColor = vbBlack
        txt_nfcl.ForeColor = vbBlack
        txt_sfcl.ForeColor = vbBlack
        txt_nfr1.ForeColor = vbBlack
        txt_nfr2.ForeColor = vbBlack
        txt_sfr1.ForeColor = vbBlack
        txt_sfr2.ForeColor = vbBlack
        lbl_yOverlap.ForeColor = vbBlack
        lbl_accept.ForeColor = vbBlack

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = "39.31"
        txt_sfs.Text = "19.03"
        txt_sfse.Text = ""
        txt_nfse.Text = ""
        txt_nfcl.Text = ""
        txt_sfcl.Text = ""
        txt_nfr1.Text = ""
        txt_nfr2.Text = ""
        txt_sfr1.Text = ""
        txt_sfr2.Text = ""
```

```
        txt_help.Text = "Now we must calculate the Standard Deviation of the readings you've collected! This formula takes a long time to complete if you're doing it by hand so you must be able to complete it using your scientific calculator!" & vbNewLine & "For North Facing: the standard deviation = 39.3135543490481" & vbNewLine & "For South Facing: the standard deviation = 19.0321365648269" & vbNewLine & "Remember, biology likes numbers to be in 2 decimal places so the accepted values would be: 39.31 and 19.03"

    Case 2
    'third help page
        lbl_nfmean.ForeColor = vbBlack
        lbl_sfmean.ForeColor = vbBlack
        txt_nfm.ForeColor = vbBlack
        txt_sfm.ForeColor = vbBlack
        txt_nfs.ForeColor = vbBlack
        txt_sfs.ForeColor = vbBlack
        txt_sfse.ForeColor = vbRed
        txt_nfse.ForeColor = vbRed
        txt_nfcl.ForeColor = vbBlack
        txt_sfcl.ForeColor = vbBlack
        txt_nfr1.ForeColor = vbBlack
        txt_nfr2.ForeColor = vbBlack
        txt_sfr1.ForeColor = vbBlack
        txt_sfr2.ForeColor = vbBlack
        lbl_yOverlap.ForeColor = vbBlack
        lbl_accept.ForeColor = vbBlack

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = "39.31"
        txt_sfs.Text = "19.03"
        txt_sfse.Text = "6.02"
        txt_nfse.Text = "12.43"
        txt_nfcl.Text = ""
        txt_sfcl.Text = ""
        txt_nfr1.Text = ""
        txt_nfr2.Text = ""
        txt_sfr1.Text = ""
        txt_sfr2.Text = ""

        txt_help.Text = "After having calculated the standard deviation, you have to calculate the Standard Error. To do this, you divide the standard deviation by the square root of n. in this case, n = 10." & vbNewLine & "For North Facing:" & vbNewLine & "39.31 / root(10) = 12.4309134821" & vbNewLine & "For South Facing:" & vbNewLine & "19.03 / root(10) = 6.0178143873" & vbNewLine & "Remember, biology likes values to be in 2 decimal places so the accpeted results would be: 12.43 and 6.02"

    Case 3
    'fourth help page
        lbl_nfmean.ForeColor = vbBlack
```

```
        lbl_sfmean.ForeColor = vbBlack
        txt_nfm.ForeColor = vbBlack
        txt_sfm.ForeColor = vbBlack
        txt_nfs.ForeColor = vbBlack
        txt_sfs.ForeColor = vbBlack
        txt_sfse.ForeColor = vbBlack
        txt_nfse.ForeColor = vbBlack
        txt_nfcl.ForeColor = vbRed
        txt_sfcl.ForeColor = vbRed
        txt_nfr1.ForeColor = vbBlack
        txt_nfr2.ForeColor = vbBlack
        txt_sfr1.ForeColor = vbBlack
        txt_sfr2.ForeColor = vbBlack
        lbl_yOverlap.ForeColor = vbBlack
        lbl_accept.ForeColor = vbBlack

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = "39.31"
        txt_sfs.Text = "19.03"
        txt_sfse.Text = "6.02"
        txt_nfse.Text = "12.43"
        txt_nfcl.Text = "24.86"
        txt_sfcl.Text = "12.04"
        txt_nfr1.Text = ""
        txt_nfr2.Text = ""
        txt_sfr1.Text = ""
        txt_sfr2.Text = ""

        txt_help.Text = "For the confidence limits, you simply multiply the
vlaue you got for Standard Error by 2." & vbNewLine & "For North Facing:" &
vbNewLine & "12.43 x 2 = 24.86" & vbNewLine & "For South Facing:" & vbNewLine &
"6.02 x 2 = 12.04"

    Case 4
    'fifth help page
        lbl_nfmean.ForeColor = vbBlack
        lbl_sfmean.ForeColor = vbBlack
        txt_nfm.ForeColor = vbBlack
        txt_sfm.ForeColor = vbBlack
        txt_nfs.ForeColor = vbBlack
        txt_sfs.ForeColor = vbBlack
        txt_sfse.ForeColor = vbBlack
        txt_nfse.ForeColor = vbBlack
        txt_nfcl.ForeColor = vbBlack
        txt_sfcl.ForeColor = vbBlack
        txt_nfr1.ForeColor = vbRed
        txt_nfr2.ForeColor = vbRed
        txt_sfr1.ForeColor = vbRed
        txt_sfr2.ForeColor = vbRed
        lbl_yOverlap.ForeColor = vbBlack
```

```
        lbl_accept.ForeColor = vbBlack

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = "39.31"
        txt_sfs.Text = "19.03"
        txt_sfse.Text = "6.02"
        txt_nfse.Text = "12.43"
        txt_nfcl.Text = "24.86"
        txt_sfcl.Text = "12.04"
        txt_nfr1.Text = "27.14"
        txt_nfr2.Text = "76.86"
        txt_sfr1.Text = "45.96"
        txt_sfr2.Text = "70.04"

        txt_help.Text = "For the range, you just +/- the confidence limits from
the mean" & vbNewLine & "For North Facing:" & vbNewLine & "52 - 24.86 = 27.14"
& vbNewLine & "54 + 24.86 = 76.86" & vbNewLine & "For South Site:" & vbNewLine
& "58 - 12.04 = 45.96" & vbNewLine & "58 + 12.04 = 70.04"

    Case 5
    'sixth help page
        lbl_nfmean.ForeColor = vbBlack
        lbl_sfmean.ForeColor = vbBlack
        txt_nfm.ForeColor = vbBlack
        txt_sfm.ForeColor = vbBlack
        txt_nfs.ForeColor = vbBlack
        txt_sfs.ForeColor = vbBlack
        txt_sfse.ForeColor = vbBlack
        txt_nfse.ForeColor = vbBlack
        txt_nfcl.ForeColor = vbBlack
        txt_sfcl.ForeColor = vbBlack
        txt_nfr1.ForeColor = vbBlack
        txt_nfr2.ForeColor = vbBlack
        txt_sfr1.ForeColor = vbBlack
        txt_sfr2.ForeColor = vbBlack
        lbl_yOverlap.ForeColor = vbRed
        lbl_accept.ForeColor = vbRed

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = "39.31"
        txt_sfs.Text = "19.03"
        txt_sfse.Text = "6.02"
        txt_nfse.Text = "12.43"
        txt_nfcl.Text = "24.86"
        txt_sfcl.Text = "12.04"
        txt_nfr1.Text = "27.14"
        txt_nfr2.Text = "76.86"
        txt_sfr1.Text = "45.96"
        txt_sfr2.Text = "70.04"
```

```
        txt_help.Text = "Now you need to see whether the ranges have overlapped. As you can see from the example, there is an overlap. " & vbNewLine & "27.14 - 45.96 - 70.04 - 76.86" & vbNewLine & "The whole range of the North Facing results overlaps the South Facing results. Because of this overlap, we are forced to admit that we cannot be definitively sure that any difference is significant" & vbNewLine & "Therefore, we must accept the Null Hypothesis"

    End Select
End If
End Sub



Private Sub cmd_back_Click()
Unload me
frm_StandardError.Show
End Sub

Private Sub cmd_prev_Click()
If page <> 0 Then
    page = page - 1
    Select Case page

    Case 0
    'starting help page
        lbl_nfmean.ForeColor = vbRed
        lbl_sfmean.ForeColor = vbRed
        txt_nfm.ForeColor = vbRed
        txt_sfm.ForeColor = vbRed
        txt_nfs.ForeColor = vbBlack
        txt_sfs.ForeColor = vbBlack
        txt_sfse.ForeColor = vbBlack
        txt_nfse.ForeColor = vbBlack
        txt_nfcl.ForeColor = vbBlack
        txt_sfcl.ForeColor = vbBlack
        txt_nfr1.ForeColor = vbBlack
        txt_nfr2.ForeColor = vbBlack
        txt_sfr1.ForeColor = vbBlack
        txt_sfr2.ForeColor = vbBlack
        lbl_yOverlap.ForeColor = vbBlack
        lbl_accept.ForeColor = vbBlack

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = ""
        txt_sfs.Text = ""
        txt_sfse.Text = ""
        txt_nfse.Text = ""
        txt_nfcl.Text = ""
        txt_sfcl.Text = ""
```

```
        txt_nfr1.Text = ""
        txt_nfr2.Text = ""
        txt_sfr1.Text = ""
        txt_sfr2.Text = ""

        txt_help.Text = "The mean % cover is just the mean of your own
readings. In this case, the mean percentage cover of plants at North " &
vbNewLine & "Facing Site was 52. So we put in 52 into the stats table under the
North Facing column" & vbNewLine & "Obviously, you would do the same for the
South Facing mean as well."

    Case 1
    'second help page
        lbl_nfmean.ForeColor = vbBlack
        lbl_sfmean.ForeColor = vbBlack
        txt_nfm.ForeColor = vbBlack
        txt_sfm.ForeColor = vbBlack
        txt_nfs.ForeColor = vbRed
        txt_sfs.ForeColor = vbRed
        txt_sfse.ForeColor = vbBlack
        txt_nfse.ForeColor = vbBlack
        txt_nfcl.ForeColor = vbBlack
        txt_sfcl.ForeColor = vbBlack
        txt_nfr1.ForeColor = vbBlack
        txt_nfr2.ForeColor = vbBlack
        txt_sfr1.ForeColor = vbBlack
        txt_sfr2.ForeColor = vbBlack
        lbl_yOverlap.ForeColor = vbBlack
        lbl_accept.ForeColor = vbBlack

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = "39.31"
        txt_sfs.Text = "19.03"
        txt_sfse.Text = ""
        txt_nfse.Text = ""
        txt_nfcl.Text = ""
        txt_sfcl.Text = ""
        txt_nfr1.Text = ""
        txt_nfr2.Text = ""
        txt_sfr1.Text = ""
        txt_sfr2.Text = ""

        txt_help.Text = "Now we must calculate the Standard Deviation of the
readings you've collected! This formula takes a long time to complete if you're
doing it by hand so you must be able to complete it using your scientific
calculator!" & vbNewLine & "For North Facing: the standard deviation =
39.3135543490481" & vbNewLine & "For South Facing: the standard deviation =
19.0321365648269" & vbNewLine & "Remember, biology likes numbers to be in 2
decimal places so the accepted values would be: 39.31 and 19.03"
```

```
    Case 2
    'third help page
        lbl_nfmean.ForeColor = vbBlack
        lbl_sfmean.ForeColor = vbBlack
        txt_nfm.ForeColor = vbBlack
        txt_sfm.ForeColor = vbBlack
        txt_nfs.ForeColor = vbBlack
        txt_sfs.ForeColor = vbBlack
        txt_sfse.ForeColor = vbRed
        txt_nfse.ForeColor = vbRed
        txt_nfcl.ForeColor = vbBlack
        txt_sfcl.ForeColor = vbBlack
        txt_nfr1.ForeColor = vbBlack
        txt_nfr2.ForeColor = vbBlack
        txt_sfr1.ForeColor = vbBlack
        txt_sfr2.ForeColor = vbBlack
        lbl_yOverlap.ForeColor = vbBlack
        lbl_accept.ForeColor = vbBlack

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = "39.31"
        txt_sfs.Text = "19.03"
        txt_sfse.Text = "6.02"
        txt_nfse.Text = "12.43"
        txt_nfcl.Text = ""
        txt_sfcl.Text = ""
        txt_nfr1.Text = ""
        txt_nfr2.Text = ""
        txt_sfr1.Text = ""
        txt_sfr2.Text = ""

        txt_help.Text = "After having calculated the standard deviation, you
have to calculate the Standard Error. To do this, you divide the standard
deviation by the square root of n. in this case, n = 10." & vbNewLine & "For
North Facing:" & vbNewLine & "39.31 / root(10) = 12.4309134821" & vbNewLine &
"For South Facing:" & vbNewLine & "19.03 / root(10) = 6.0178143873" & vbNewLine
& "Remember, biology likes values to be in 2 decimal places so the accpeted
results would be: 12.43 and 6.02"

    Case 3
    'fourth help page
        lbl_nfmean.ForeColor = vbBlack
        lbl_sfmean.ForeColor = vbBlack
        txt_nfm.ForeColor = vbBlack
        txt_sfm.ForeColor = vbBlack
        txt_nfs.ForeColor = vbBlack
        txt_sfs.ForeColor = vbBlack
        txt_sfse.ForeColor = vbBlack
        txt_nfse.ForeColor = vbBlack
        txt_nfcl.ForeColor = vbRed
```

```
        txt_sfcl.ForeColor = vbRed
        txt_nfr1.ForeColor = vbBlack
        txt_nfr2.ForeColor = vbBlack
        txt_sfr1.ForeColor = vbBlack
        txt_sfr2.ForeColor = vbBlack
        lbl_yOverlap.ForeColor = vbBlack
        lbl_accept.ForeColor = vbBlack

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = "39.31"
        txt_sfs.Text = "19.03"
        txt_sfse.Text = "6.02"
        txt_nfse.Text = "12.43"
        txt_nfcl.Text = "24.86"
        txt_sfcl.Text = "12.04"
        txt_nfr1.Text = ""
        txt_nfr2.Text = ""
        txt_sfr1.Text = ""
        txt_sfr2.Text = ""

        txt_help.Text = "For the confidence limits, you simply multiply the
vlaue you got for Standard Error by 2." & vbNewLine & "For North Facing:" &
vbNewLine & "12.43 x 2 = 24.86" & vbNewLine & "For South Facing:" & vbNewLine &
"6.02 x 2 = 12.04"

    Case 4
    'fifth help page
        lbl_nfmean.ForeColor = vbBlack
        lbl_sfmean.ForeColor = vbBlack
        txt_nfm.ForeColor = vbBlack
        txt_sfm.ForeColor = vbBlack
        txt_nfs.ForeColor = vbBlack
        txt_sfs.ForeColor = vbBlack
        txt_sfse.ForeColor = vbBlack
        txt_nfse.ForeColor = vbBlack
        txt_nfcl.ForeColor = vbBlack
        txt_sfcl.ForeColor = vbBlack
        txt_nfr1.ForeColor = vbRed
        txt_nfr2.ForeColor = vbRed
        txt_sfr1.ForeColor = vbRed
        txt_sfr2.ForeColor = vbRed
        lbl_yOverlap.ForeColor = vbBlack
        lbl_accept.ForeColor = vbBlack

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = "39.31"
        txt_sfs.Text = "19.03"
        txt_sfse.Text = "6.02"
        txt_nfse.Text = "12.43"
```

```
        txt_nfcl.Text = "24.86"
        txt_sfcl.Text = "12.04"
        txt_nfr1.Text = "27.14"
        txt_nfr2.Text = "76.86"
        txt_sfr1.Text = "45.96"
        txt_sfr2.Text = "70.04"

        txt_help.Text = "For the range, you just +/- the confidence limits from
the mean" & vbNewLine & "For North Facing:" & vbNewLine & "52 - 24.86 = 27.14"
& vbNewLine & "54 + 24.86 = 76.86" & vbNewLine & "For South Site:" & vbNewLine
& "58 - 12.04 = 45.96" & vbNewLine & "58 + 12.04 = 70.04"

    Case 5
    'sixth help page
        lbl_nfmean.ForeColor = vbBlack
        lbl_sfmean.ForeColor = vbBlack
        txt_nfm.ForeColor = vbBlack
        txt_sfm.ForeColor = vbBlack
        txt_nfs.ForeColor = vbBlack
        txt_sfs.ForeColor = vbBlack
        txt_sfse.ForeColor = vbBlack
        txt_nfse.ForeColor = vbBlack
        txt_nfcl.ForeColor = vbBlack
        txt_sfcl.ForeColor = vbBlack
        txt_nfr1.ForeColor = vbBlack
        txt_nfr2.ForeColor = vbBlack
        txt_sfr1.ForeColor = vbBlack
        txt_sfr2.ForeColor = vbBlack
        lbl_yOverlap.ForeColor = vbRed
        lbl_accept.ForeColor = vbRed

        txt_nfm.Text = "52"
        txt_sfm.Text = "58"
        txt_nfs.Text = "39.31"
        txt_sfs.Text = "19.03"
        txt_sfse.Text = "6.02"
        txt_nfse.Text = "12.43"
        txt_nfcl.Text = "24.86"
        txt_sfcl.Text = "12.04"
        txt_nfr1.Text = "27.14"
        txt_nfr2.Text = "76.86"
        txt_sfr1.Text = "45.96"
        txt_sfr2.Text = "70.04"

        txt_help.Text = "Now you need to see whether the ranges have
overlapped. As you can see from the example, there is an overlap. " & vbNewLine
& "27.14 - 45.96 - 70.04 - 76.86" & vbNewLine & "The whole range of the North
Facing results overlaps the South Facing results. Because of this overlap, we
are forced to admit that we cannot be definitively sure that any difference is
significant" & vbNewLine & "Therefore, we must accept the Null Hypothesis"
```

```
    End Select
End If
End Sub

Private Sub Form_Load()
page = 0
'intially starts on page 0
lbl_nfmean.ForeColor = vbRed
lbl_sfmean.ForeColor = vbRed
txt_nfm.ForeColor = vbRed
txt_sfm.ForeColor = vbRed
txt_nfs.ForeColor = vbBlack
txt_sfs.ForeColor = vbBlack
txt_sfse.ForeColor = vbBlack
txt_nfse.ForeColor = vbBlack
txt_nfcl.ForeColor = vbBlack
txt_sfcl.ForeColor = vbBlack
txt_nfr1.ForeColor = vbBlack
txt_nfr2.ForeColor = vbBlack
txt_sfr1.ForeColor = vbBlack
txt_sfr2.ForeColor = vbBlack
lbl_yOverlap.ForeColor = vbBlack
lbl_accept.ForeColor = vbBlack

txt_nfm.Text = "52"
txt_sfm.Text = "58"
txt_nfs.Text = ""
txt_sfs.Text = ""
txt_sfse.Text = ""
txt_nfse.Text = ""
txt_nfcl.Text = ""
txt_sfcl.Text = ""
txt_nfr1.Text = ""
txt_nfr2.Text = ""
txt_sfr1.Text = ""
txt_sfr2.Text = ""

txt_help.Text = "The mean % cover is just the mean of your own readings. In
this case, the mean percentage cover of plants at North " & vbNewLine & "Facing
Site was 52. So we put in 52 into the stats table under the North Facing
column" & vbNewLine & "Obviously, you would do the same for the South Facing
mean as well."

End Sub

Private Sub lbl_st_Click()
page = 1
'second page
lbl_nfmean.ForeColor = vbBlack
lbl_sfmean.ForeColor = vbBlack
```

```
txt_nfm.ForeColor = vbBlack
txt_sfm.ForeColor = vbBlack
txt_nfs.ForeColor = vbRed
txt_sfs.ForeColor = vbRed
txt_sfse.ForeColor = vbBlack
txt_nfse.ForeColor = vbBlack
txt_nfcl.ForeColor = vbBlack
txt_sfcl.ForeColor = vbBlack
txt_nfr1.ForeColor = vbBlack
txt_nfr2.ForeColor = vbBlack
txt_sfr1.ForeColor = vbBlack
txt_sfr2.ForeColor = vbBlack
lbl_yOverlap.ForeColor = vbBlack
lbl_accept.ForeColor = vbBlack

txt_nfm.Text = "52"
txt_sfm.Text = "58"
txt_nfs.Text = "39.31"
txt_sfs.Text = "19.03"
txt_sfse.Text = ""
txt_nfse.Text = ""
txt_nfcl.Text = ""
txt_sfcl.Text = ""
txt_nfr1.Text = ""
txt_nfr2.Text = ""
txt_sfr1.Text = ""
txt_sfr2.Text = ""

txt_help.Text = "Now we must calculate the Standard Deviation of the readings
you've collected! This formula takes a long time to complete if you're doing it
by hand so you must be able to complete it using your scientific calculator!" &
vbNewLine & "For North Facing: the standard deviation = 39.3135543490481" &
vbNewLine & "For South Facing: the standard deviation = 19.0321365648269" &
vbNewLine & "Remember, biology likes numbers to be in 2 decimal places so the
accepted values would be: 39.31 and 19.03"

End Sub

Private Sub lbl_se_Click()
page = 2
'third page
lbl_nfmean.ForeColor = vbBlack
lbl_sfmean.ForeColor = vbBlack
txt_nfm.ForeColor = vbBlack
txt_sfm.ForeColor = vbBlack
txt_nfs.ForeColor = vbBlack
txt_sfs.ForeColor = vbBlack
txt_sfse.ForeColor = vbRed
txt_nfse.ForeColor = vbRed
txt_nfcl.ForeColor = vbBlack
txt_sfcl.ForeColor = vbBlack
```

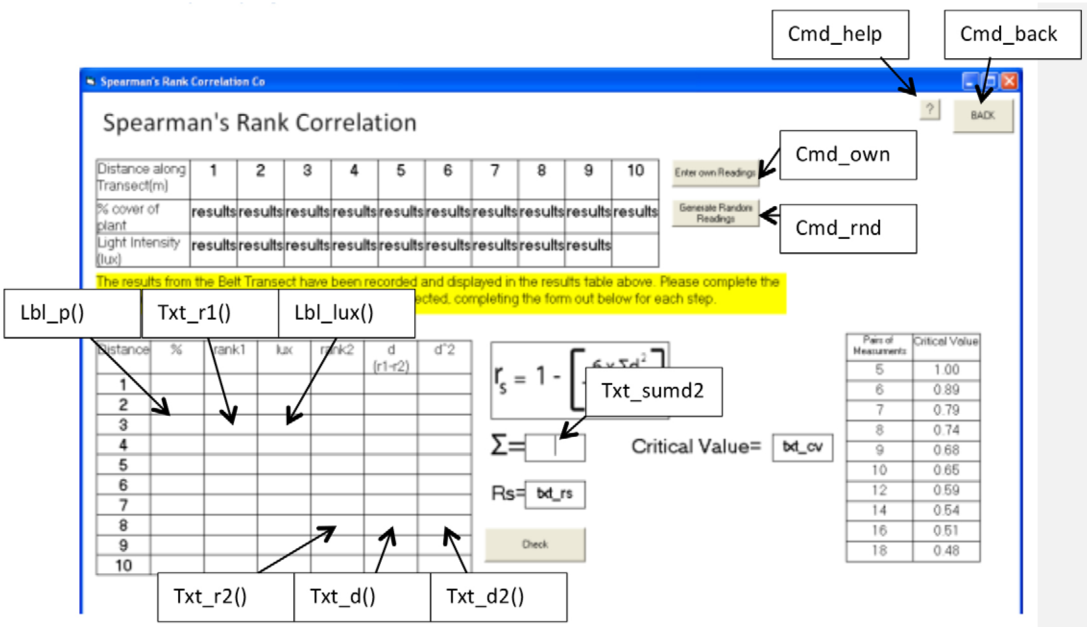
```
txt_nfr1.ForeColor = vbBlack
txt_nfr2.ForeColor = vbBlack
txt_sfr1.ForeColor = vbBlack
txt_sfr2.ForeColor = vbBlack
lbl_yOverlap.ForeColor = vbBlack
lbl_accept.ForeColor = vbBlack

txt_nfm.Text = "52"
txt_sfm.Text = "58"
txt_nfs.Text = "39.31"
txt_sfs.Text = "19.03"
txt_sfse.Text = "6.02"
txt_nfse.Text = "12.43"
txt_nfcl.Text = ""
txt_sfcl.Text = ""
txt_nfr1.Text = ""
txt_nfr2.Text = ""
txt_sfr1.Text = ""
txt_sfr2.Text = ""

txt_help.Text = "After having calculated the standard deviation, you have to
calculate the Standard Error. To do this, you divide the standard deviation by
the square root of n. in this case, n = 10." & vbNewLine & "For North Facing:"
& vbNewLine & "39.31 / root(10) = 12.4309134821" & vbNewLine & "For South
Facing:" & vbNewLine & "19.03 / root(10) = 6.0178143873" & vbNewLine &
"Remember, biology likes values to be in 2 decimal places so the accpeted
results would be: 12.43 and 6.02"

End Sub

Private Sub lbl_cl_Click()
page = 3
'fourth page
lbl_nfmean.ForeColor = vbBlack
lbl_sfmean.ForeColor = vbBlack
txt_nfm.ForeColor = vbBlack
txt_sfm.ForeColor = vbBlack
txt_nfs.ForeColor = vbBlack
txt_sfs.ForeColor = vbBlack
txt_sfse.ForeColor = vbBlack
txt_nfse.ForeColor = vbBlack
txt_nfcl.ForeColor = vbRed
txt_sfcl.ForeColor = vbRed
txt_nfr1.ForeColor = vbBlack
txt_nfr2.ForeColor = vbBlack
txt_sfr1.ForeColor = vbBlack
txt_sfr2.ForeColor = vbBlack
lbl_yOverlap.ForeColor = vbBlack
lbl_accept.ForeColor = vbBlack

txt_nfm.Text = "52"
```

```
txt_sfm.Text = "58"
txt_nfs.Text = "39.31"
txt_sfs.Text = "19.03"
txt_sfse.Text = "6.02"
txt_nfse.Text = "12.43"
txt_nfcl.Text = "24.86"
txt_sfcl.Text = "12.04"
txt_nfr1.Text = ""
txt_nfr2.Text = ""
txt_sfr1.Text = ""
txt_sfr2.Text = ""

txt_help.Text = "For the confidence limits, you simply multiply the vlaue you
got for Standard Error by 2." & vbNewLine & "For North Facing:" & vbNewLine &
"12.43 x 2 = 24.86" & vbNewLine & "For South Facing:" & vbNewLine & "6.02 x 2 =
12.04"

End Sub

Private Sub lbl_range_Click()
page = 4
'fifth page
lbl_nfmean.ForeColor = vbBlack
lbl_sfmean.ForeColor = vbBlack
txt_nfm.ForeColor = vbBlack
txt_sfm.ForeColor = vbBlack
txt_nfs.ForeColor = vbBlack
txt_sfs.ForeColor = vbBlack
txt_sfse.ForeColor = vbBlack
txt_nfse.ForeColor = vbBlack
txt_nfcl.ForeColor = vbBlack
txt_sfcl.ForeColor = vbBlack
txt_nfr1.ForeColor = vbRed
txt_nfr2.ForeColor = vbRed
txt_sfr1.ForeColor = vbRed
txt_sfr2.ForeColor = vbRed
lbl_yOverlap.ForeColor = vbBlack
lbl_accept.ForeColor = vbBlack

txt_nfm.Text = "52"
txt_sfm.Text = "58"
txt_nfs.Text = "39.31"
txt_sfs.Text = "19.03"
txt_sfse.Text = "6.02"
txt_nfse.Text = "12.43"
txt_nfcl.Text = "24.86"
txt_sfcl.Text = "12.04"
txt_nfr1.Text = "27.14"
txt_nfr2.Text = "76.86"
txt_sfr1.Text = "45.96"
txt_sfr2.Text = "70.04"
```

```
txt_help.Text = "For the range, you just +/- the confidence limits from the
mean" & vbNewLine & "For North Facing:" & vbNewLine & "52 - 24.86 = 27.14" &
vbNewLine & "54 + 24.86 = 76.86" & vbNewLine & "For South Site:" & vbNewLine &
"58 - 12.04 = 45.96" & vbNewLine & "58 + 12.04 = 70.04"

End Sub

Private Sub lbl_overlap_Click()
page = 5
'sixth page
lbl_nfmean.ForeColor = vbBlack
lbl_sfmean.ForeColor = vbBlack
txt_nfm.ForeColor = vbBlack
txt_sfm.ForeColor = vbBlack
txt_nfs.ForeColor = vbBlack
txt_sfs.ForeColor = vbBlack
txt_sfse.ForeColor = vbBlack
txt_nfse.ForeColor = vbBlack
txt_nfcl.ForeColor = vbBlack
txt_sfcl.ForeColor = vbBlack
txt_nfr1.ForeColor = vbBlack
txt_nfr2.ForeColor = vbBlack
txt_sfr1.ForeColor = vbBlack
txt_sfr2.ForeColor = vbBlack
lbl_yOverlap.ForeColor = vbRed
lbl_accept.ForeColor = vbRed

txt_nfm.Text = "52"
txt_sfm.Text = "58"
txt_nfs.Text = "39.31"
txt_sfs.Text = "19.03"
txt_sfse.Text = "6.02"
txt_nfse.Text = "12.43"
txt_nfcl.Text = "24.86"
txt_sfcl.Text = "12.04"
txt_nfr1.Text = "27.14"
txt_nfr2.Text = "76.86"
txt_sfr1.Text = "45.96"
txt_sfr2.Text = "70.04"

txt_help.Text = "Now you need to see whether the ranges have overlapped. As you
can see from the example, there is an overlap. " & vbNewLine & "27.14 - 45.96
- 70.04 - 76.86" & vbNewLine & "The whole range of the North Facing results
overlaps the South Facing results. Because of this overlap, we are forced to
admit that we cannot be definitively sure that any difference is significant" &
vbNewLine & "Therefore, we must accept the Null Hypothesis"

End Sub

Private Sub lbl_m_Click()
page = 0
'first page
```

```
lbl_nfmean.ForeColor = vbRed
lbl_sfmean.ForeColor = vbRed
txt_nfm.ForeColor = vbRed
txt_sfm.ForeColor = vbRed
txt_nfs.ForeColor = vbBlack
txt_sfs.ForeColor = vbBlack
txt_sfse.ForeColor = vbBlack
txt_nfse.ForeColor = vbBlack
txt_nfcl.ForeColor = vbBlack
txt_sfcl.ForeColor = vbBlack
txt_nfr1.ForeColor = vbBlack
txt_nfr2.ForeColor = vbBlack
txt_sfr1.ForeColor = vbBlack
txt_sfr2.ForeColor = vbBlack
lbl_yOverlap.ForeColor = vbBlack
lbl_accept.ForeColor = vbBlack

txt_nfm.Text = "52"
txt_sfm.Text = "58"
txt_nfs.Text = ""
txt_sfs.Text = ""
txt_sfse.Text = ""
txt_nfse.Text = ""
txt_nfcl.Text = ""
txt_sfcl.Text = ""
txt_nfr1.Text = ""
txt_nfr2.Text = ""
txt_sfr1.Text = ""
txt_sfr2.Text = ""

txt_help.Text = "The mean % cover is just the mean of your own readings. In
this case, the mean percentage cover of plants at North " & vbNewLine & "Facing
Site was 52. So we put in 52 into the stats table under the North Facing
column" & vbNewLine & "Obviously, you would do the same for the South Facing
mean as well."

End Sub
```

## Frm_Spearmans:




```
Dim percPlant(15) As Integer
Dim lux(15) As Integer
Dim pointer As Double
Dim difference As Double
Dim sumd As Double
Dim spearman As Double
Dim criticalValue As Double
Dim correct As Integer
Dim complete As Boolean
Dim finished As Boolean

Private Sub cmd_help_Click()
frm_spearmans_help.Show
End Sub

Private Sub cmd_own_Click()
'submitting users own data into results table and stats
If stats = False Then
    For I = 1 tO 20
'percentage cover and light intensity
        If IsNumeric(results(I - 1)) = True Then
            sim1.readings(I) = results(I - 1).Text
            results(I - 1).Locked = True
        Else
            MsgBox "Please enter all the data into the table", , "Error"
            Exit Sub
        End If
    Next I
'splitting the 20 values into two seperate variables that store 10
    For I = 1 tO 10
        percPlant(I) = sim1.readings(I)
        lux(I) = sim1.readings(I + 10)
        lbl_p(I).Caption = percPlant(I)
```

```
        lbl_lux(I).Caption = lux(I)
    Next I
    stats = True
End If



End Sub

Private Sub cmd_rnd_Click()
If stats = False Then
'generating random results for the stats and displaying them in the results
table
    For I = 1 tO 10
'percentage cover
        Randomize
        results(I - 1).Text = Int(Rnd(1) * 20) * 5
        sim1.readings(I) = results(I - 1).Text
        results(I - 1).Locked = True
    Next I

    For I = 11 tO 20
'light intensity
        Randomize
        results(I - 1).Text = Int(((14 - 9 + 1) * Rnd + 9) * 1000)
        sim1.readings(I) = results(I - 1).Text
        results(I - 1).Locked = True
    Next I
'seperating 20 values into 2 seperate variables to store 10 each
    For I = 1 tO 10
        percPlant(I) = sim1.readings(I)
        lux(I) = sim1.readings(I + 10)
        lbl_p(I).Caption = percPlant(I)
        lbl_lux(I).Caption = lux(I)
    Next I
    stats = True
End If

End Sub

Private Sub cmd_check_Click()
Call Sorting(percPlant)
Call Sorting(lux)



If complete = False Then
    'comparing the sorted data with the users data - percPlant
    For I = 1 tO 10
        If IsNumeric(txt_r1(I).Text) = True Then
        'if value is an integer or not
            pointer = txt_r1(I).Text
```

```
            If Int(txt_r1(I).Text) = txt_r1(I).Text Then
                pointer = percPlant(pointer)
            Else
                pointer = Int(pointer)
                pointer = percPlant(pointer)
            End If
            If txt_r1(I).Text <> "" Then
                If pointer = lbl_p(I) Then
                    txt_r1(I).BackColor = vbGreen
                    txt_r1(I).Locked = True
                Else
                    txt_r1(I).BackColor = vbRed
                End If
            Else
                MsgBox "Please enter all the ranks for the data", , "Error"
            End If
        Else
            txt_r1(I).BackColor = vbRed
            MsgBox "Only numeric data may be entered into the fields!", ,
"Error"
            Exit Sub
        End If
    Next I
    'correct
    For I = 1 tO 10
        If txt_r1(I).BackColor = vbGreen Then
            correct = correct + 1
        End If
    Next I



    'Comparing the sorted data with the users data - lux
    For I = 1 tO 10
        If IsNumeric(txt_r2(I).Text) = True Then
            pointer = txt_r2(I).Text
            If Int(txt_r2(I).Text) = txt_r2(I).Text Then
                pointer = lux(pointer)
            Else
                pointer = Int(pointer)
                pointer = lux(pointer)
            End If
            If txt_r2(I).Text <> "" Then
                If pointer = lbl_lux(I) Then
                    txt_r2(I).BackColor = vbGreen
                    txt_r2(I).Locked = True
                Else
                    txt_r2(I).BackColor = vbRed
                End If
            Else
                MsgBox "Please enter all the ranks for the data.", , "Error"
            End If
```

```
        Else
            txt_r2(I).BackColor = vbRed
            MsgBox "Only numeric data may be entered into the fields!", , 
"Error"
            Exit Sub
        End If
    Next I

    'correct
    For I = 1 tO 10
        If txt_r2(I).BackColor = vbGreen Then
            correct = correct + 1
        End If
    Next I

    'checking the d values
    For I = 1 tO 10
        If IsNumeric(txt_r1(I).Text) = True And IsNumeric(txt_r2(I).Text) = 
True And IsNumeric(txt_d(I).Text) = True Then
            If txt_r1(I).BackColor = vbGreen And txt_r2(I).BackColor = vbGreen 
Then
                pointer = txt_r1(I)
                difference = pointer
                pointer = txt_r2(I)
                difference = difference - pointer
                If txt_d(I).Text <> "" Then
                    If txt_d(I).Text = difference Then
                        txt_d(I).BackColor = vbGreen
                        txt_d(I).Locked = True
                    Else
                        txt_d(I).BackColor = vbRed
                        Locked = False
                    End If
                Else
                    MsgBox "Please enter the value of d.", , "Error"
                End If
            End If
        Else
            txt_d(I).BackColor = vbRed
            MsgBox "Only numeric data may be entered into the fields!", , 
"Error"
            Exit Sub
        End If
    Next I

    'correct
    For I = 1 tO 10
        If txt_d(I).BackColor = vbGreen Then
            correct = correct + 1
        End If
    Next I
```

```
    'checking the d^2 values
    For I = 1 tO 10
        If IsNumeric(txt_d(I).Text) = True Then
            If txt_d(I).BackColor = vbGreen Then
                difference = txt_d(I)
                difference = difference ^ 2
                If txt_d2(I).Text <> "" Then
                    If IsNumeric(txt_d2(I).Text) = True Then
                        If txt_d2(I) = difference Then
                            txt_d2(I).BackColor = vbGreen
                            txt_d2(I).Locked = True
                        Else
                            txt_d2(I).BackColor = vbRed
                            Locked = False
                        End If
                    Else
                        MsgBox "Only Numerical data may be entered!", , "Error"
                        txt_d2(I).BackColor = vbRed
                        Exit Sub
                    End If
                Else
                    MsgBox "Please enter the d-squared value.", , "Error"
                End If
            End If
        Else
            txt_d2(I).BackColor = vbRed
            MsgBox "Only numeric data may be entered into the fields!", ,
"Error"
            Exit Sub
        End If
    Next I

    'correct
    For I = 1 tO 10
        If txt_d2(I).BackColor = vbGreen Then
            correct = correct + 1
        End If
    Next I

    'sum of d2

    sumd = 0
    For I = 1 tO 10
        If IsNumeric(txt_d2(I).Text) = True Then
            sumd = sumd + Val(txt_d2(I))
        Else
            MsgBox "Only numeric data may be entered into the fields!", ,
"Error"
            Exit Sub
        End If
    Next I
```

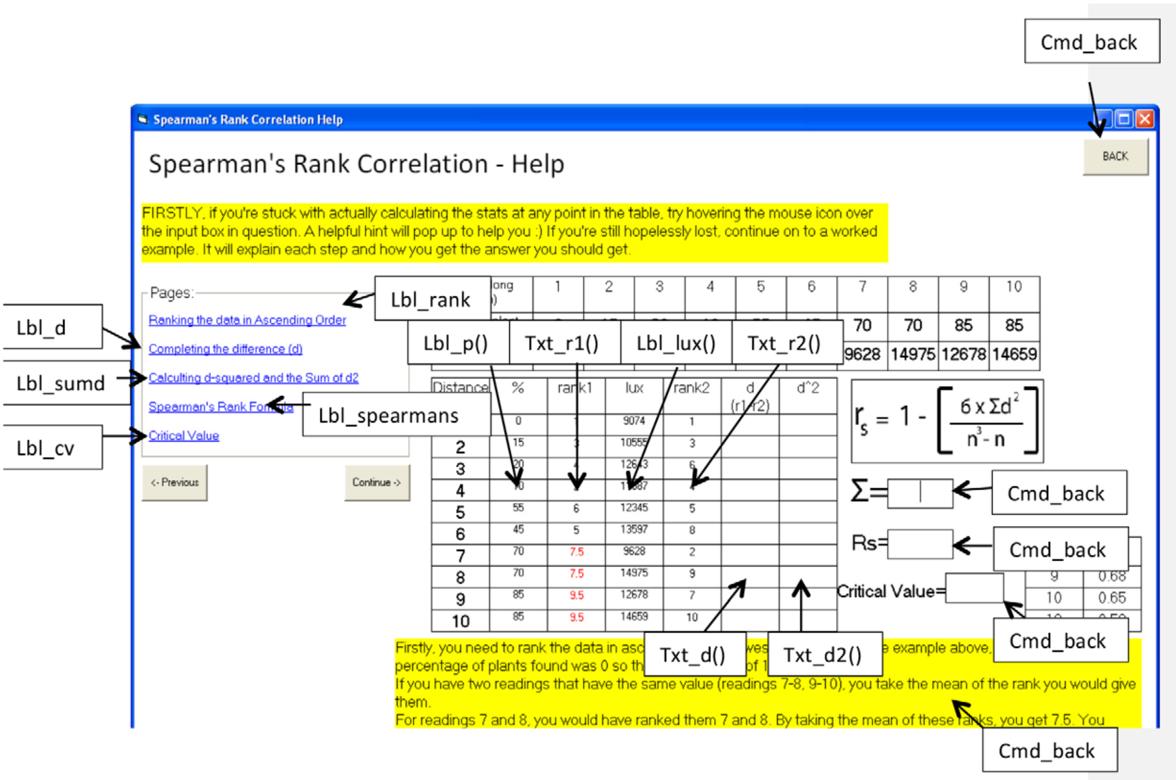
```
If txt_sumd2.Text <> "" Then
    If IsNumeric(txt_sumd2.Text) = True Then
        If txt_sumd2.Text = sumd Then
            txt_sumd2.BackColor = vbGreen
            txt_sumd2.Locked = True
            correct = correct + 1
        Else
            txt_sumd2.BackColor = vbRed
            Locked = False
            correct = 0
        End If
    Else
        txt_sumd2.BackColor = vbRed
        MsgBox "Only numerical data may be entered!", , "Error"
        Exit Sub
    End If
Else
    MsgBox "Please enter the sum of the d-squared values.", , "Error"
End If


'spearman's rank
If txt_sumd2.BackColor = vbGreen Then
    spearman = Round(spearmans(sumd), 2)
    If txt_rs.Text <> "" Then
        If IsNumeric(txt_rs.Text) = True Then
            If txt_rs.Text = spearman Then
                txt_rs.BackColor = vbGreen
                txt_rs.Locked = True
                correct = correct + 1
            Else
                txt_rs.BackColor = vbRed
                correct = 0
                Locked = False
            End If
        Else
            txt_rs.BackColor = vbRed
            MsgBox "Only numerical data may be entered!", , "Error"
            Exit Sub
        End If
    Else
        MsgBox "Please enter the value of Rs.", , "Error"
    End If
End If

'critical value
If txt_cv.Text <> "" Then
    If IsNumeric(txt_cv.Text) = True Then
        If txt_cv.Text = 0.65 Then
            txt_cv.BackColor = vbGreen
```

```
                txt_cv.Locked = True
                correct = correct + 1
            Else
                txt_cv.BackColor = vbRed
                Locked = False
                correct = 0
            End If
        Else
            txt_cv.BackColor = vbRed
            MsgBox "Only numerical data may be entered!", , "Error"
            Exit Sub
        End If
    Else
        MsgBox "Please enter the critical value."
    End If

    'evaluation
    If txt_rs.BackColor = vbGreen And txt_cv.BackColor = vbGreen Then
        criticalValue = 0.65
        If txt_rs.Text < 0 Then
            criticalValue = 0.65 * 2
            criticalValue = 0.65 - criticalValue
            If txt_rs.Text <= criticalValue Then
                frm_less.Visible = True
            Else
                frm_greater.Visible = True
            End If
        ElseIf txt_rs.Text >= criticalValue Then
            frm_less.Visible = True
        Else
            frm_greater.Visible = True
        End If
    End If

    If correct = 43 Then
        complete = True
    Else
        sumd = 0
    End If
End If



End Sub

Private Sub cmd_back_Click()
Unload me
MainMenu.Show
End Sub
```

```
Private Sub Command3_Click()
frm_error.Visible = False
End Sub

Private Sub Form_Load()

For I = 1 tO 10
    percPlant(I) = 0
    lux(I) = 0
Next I

If stats = True Then
    For I = 1 tO 10
        percPlant(I) = sim1.readings(I)
        lux(I) = sim1.readings(I + 10)
    Next I

    'filling in any values that are done automatically

    For I = 1 tO 10
        results(I - 1).Text = sim1.readings(I)
        lbl_p(I).Caption = percPlant(I)
        lbl_lux(I).Caption = lux(I)
    Next I

    For I = 11 tO 20
        results(I - 1).Text = sim1.readings(I)
    Next I
Else
 frm_error.Visible = True
End If

difference = 0
correct = 0
complete = False



End Sub

Public Sub Sorting(ByRef Values As Variant)
'sorting algorithm used on sort the values in ascending order
    Dim varSwap As Variant
    Dim swapped As Boolean
    Do
        swapped = False
        For I = 1 tO 9
```

```
            If Values(I) > Values(I + 1) Then
                varSwap = Values(I)
                Values(I) = Values(I + 1)
                Values(I + 1) = varSwap
                swapped = True
            End If
        Next
    Loop Until Not swapped
End Sub

Function spearmans(sum As Double)
'Function to calculate Spearman's Rank
    Dim numerator As Double
    Dim denominator As Integer
    Dim fraction As Double
    numerator = 6 * sum
    denominator = (10 ^ 3) - 10
    fraction = numerator / denominator
    spearmans = 1 - fraction
End Function
```

## Frm_Spearmans_Help:




```
Dim page As Integer
Dim d(10) As Double
Dim d2(10) As Double

Private Sub cmd_cont_Click()
If page <> 4 Then
```

```
page = page + 1
    Select Case page

    Case 0
    'first page
        For I = 1 tO 10
            txt_d(I).Text = ""
            txt_d2(I).Text = ""
            txt_d(I).ForeColor = vbBlack
            txt_d2(I).ForeColor = vbBlack
        Next I
        txt_sumd2.Text = ""
        txt_sumd2.ForeColor = vbBlack
        txt_rs.Text = ""
        txt_rs.ForeColor = vbBlack
        txt_cv.Text = ""
        txt_cv.ForeColor = vbBlack

        txt_r1(7).ForeColor = vbRed
        txt_r1(8).ForeColor = vbRed
        txt_r1(9).ForeColor = vbRed
        txt_r1(10).ForeColor = vbRed

        txt_help.Text = "Firstly, you need to rank the data in ascending order
(lowest to highest). In the example above, the smallest percentage of plants
found was 0 so this gets the rank of 1." & vbNewLine & "If you have two
readings that have the same value (readings 7-8, 9-10), you take the mean of
the rank you would give them." & vbNewLine & "For readings 7 and 8, you would
have ranked them 7 and 8. By taking the mean of these ranks, you get 7.5. You
would then give this rank to both readings" & vbNewLine & "You do this for both
sets of data!"

    Case 1
    'second page
        For I = 1 tO 10
            txt_d(I).Text = d(I)
            txt_d2(I).Text = ""
            txt_d(I).ForeColor = vbRed
            txt_d2(I).ForeColor = vbBlack
        Next I
        txt_sumd2.Text = ""
        txt_sumd2.ForeColor = vbBlack
        txt_rs.Text = ""
        txt_rs.ForeColor = vbBlack
        txt_cv.Text = ""
        txt_cv.ForeColor = vbBlack
        txt_r1(7).ForeColor = vbBlack
        txt_r1(8).ForeColor = vbBlack
        txt_r1(9).ForeColor = vbBlack
        txt_r1(10).ForeColor = vbBlack
```

```
        txt_help.Text = "Once you have ranked the data, you need to find the
difference between the two sets of ranks. you subtract rank2 from rank1 to get
the difference" & vbNewLine & "For Example: in reading 6, rank1 = 5 and rank2 =
8. the difference (d) = 5-8=-3" & vbNewLine & "You do this even if the answer
you get is negative. This is because the next step will rectify this problem"

    Case 2
    'third page
        For I = 1 tO 10
            txt_d(I).Text = d(I)
            txt_d2(I).Text = d2(I)
            txt_d(I).ForeColor = vbBlack
            txt_d2(I).ForeColor = vbRed
        Next I
        txt_sumd2.Text = "57"
        txt_sumd2.ForeColor = vbRed
        txt_rs.Text = ""
        txt_rs.ForeColor = vbBlack
        txt_cv.Text = ""
        txt_cv.ForeColor = vbBlack
        txt_r1(7).ForeColor = vbBlack
        txt_r1(8).ForeColor = vbBlack
        txt_r1(9).ForeColor = vbBlack
        txt_r1(10).ForeColor = vbBlack

        txt_help.Text = "In this step, we have to Square all the differences we
got in the last step." & vbNewLine & "This will get rid of any negative signs
we may have (as two negatives = a positive)" & vbNewLine & "For example:
reading 6 had a difference of -3. when we square it, we get +9" & vbNewLine &
"Once you have calculated all the d2 results, you need to add them all up. This
is important as we need the answer in the spearman's Rank Formula!"

    Case 3
    'fourth page
        For I = 1 tO 10
            txt_d(I).Text = d(I)
            txt_d2(I).Text = d2(I)
            txt_d(I).ForeColor = vbBlack
            txt_d2(I).ForeColor = vbBlack
        Next I
        txt_sumd2.Text = "57"
        txt_sumd2.ForeColor = vbBlack
        txt_rs.Text = "0.65"
        txt_rs.ForeColor = vbRed
        txt_cv.Text = ""
        txt_cv.ForeColor = vbBlack
        txt_r1(7).ForeColor = vbBlack
        txt_r1(8).ForeColor = vbBlack
        txt_r1(9).ForeColor = vbBlack
        txt_r1(10).ForeColor = vbBlack
```

```
        txt_help.Text = "Next, you have to complete the Spearman's Rank Formula (this will be given to you in the exam)" & vbNewLine & "In the brackets, we have to multiply the sum of d2 by 6. You then divide it by n-cubed - n." & vbNewLine & "In this Example:" & vbNewLine & "6 x 57 = 342" & vbNewLine & "342 / 990 = 0.34545454545..." & vbNewLine & "you then subtract this value from 1. this would equal: 0.6545454545..." & vbNewLine & "In Biology, we like numbers to be in 2 decimal places so the accepted answer would be 0.65"

    Case 4
    'fifth page
        For I = 1 tO 10
            txt_d(I).Text = d(I)
            txt_d2(I).Text = d2(I)
            txt_d(I).ForeColor = vbBlack
            txt_d2(I).ForeColor = vbBlack
        Next I
        txt_sumd2.Text = "57"
        txt_sumd2.ForeColor = vbBlack
        txt_rs.Text = "0.65"
        txt_rs.ForeColor = vbBlack
        txt_cv.Text = "0.65"
        txt_cv.ForeColor = vbRed
        txt_r1(7).ForeColor = vbBlack
        txt_r1(8).ForeColor = vbBlack
        txt_r1(9).ForeColor = vbBlack
        txt_r1(10).ForeColor = vbBlack

        txt_help.Text = "Lastly, we have to find the Critical Value. To do this, you use a data table (which is given to you)." & vbNewLine & "You look at how many pairs of measurements we have. In this case, we have 10." & vbNewLine & "From this, we can see the Critical Value is 0.65." & vbNewLine & "That 's all there is to it! Now try and complete the stats with your own results and see how we interpret these numbers!"

    End Select
End If
End Sub

Private Sub cmd_back_Click()
Unload me
frm_Spearmans.Show
End Sub

Private Sub cmd_prev_Click()
If page <> 0 Then
page = page - 1
    Select Case page

    Case 0
    'first page
        For I = 1 tO 10
```

```
            txt_d(I).Text = ""
            txt_d2(I).Text = ""
            txt_d(I).ForeColor = vbBlack
            txt_d2(I).ForeColor = vbBlack
        Next I
        txt_sumd2.Text = ""
        txt_sumd2.ForeColor = vbBlack
        txt_rs.Text = ""
        txt_rs.ForeColor = vbBlack
        txt_cv.Text = ""
        txt_cv.ForeColor = vbBlack

        txt_r1(7).ForeColor = vbRed
        txt_r1(8).ForeColor = vbRed
        txt_r1(9).ForeColor = vbRed
        txt_r1(10).ForeColor = vbRed

        txt_help.Text = "Firstly, you need to rank the data in ascending order
(lowest to highest). In the example above, the smallest percentage of plants
found was 0 so this gets the rank of 1." & vbNewLine & "If you have two
readings that have the same value (readings 7-8, 9-10), you take the mean of
the rank you would give them." & vbNewLine & "For readings 7 and 8, you would
have ranked them 7 and 8. By taking the mean of these ranks, you get 7.5. You
would then give this rank to both readings" & vbNewLine & "You do this for both
sets of data!"

    Case 1
    'second page
        For I = 1 tO 10
            txt_d(I).Text = d(I)
            txt_d2(I).Text = ""
            txt_d(I).ForeColor = vbRed
            txt_d2(I).ForeColor = vbBlack
        Next I
        txt_sumd2.Text = ""
        txt_sumd2.ForeColor = vbBlack
        txt_rs.Text = ""
        txt_rs.ForeColor = vbBlack
        txt_cv.Text = ""
        txt_cv.ForeColor = vbBlack
        txt_r1(7).ForeColor = vbBlack
        txt_r1(8).ForeColor = vbBlack
        txt_r1(9).ForeColor = vbBlack
        txt_r1(10).ForeColor = vbBlack


        txt_help.Text = "Once you have ranked the data, you need to find the
difference between the two sets of ranks. you subtract rank2 from rank1 to get
the difference" & vbNewLine & "For Example: in reading 6, rank1 = 5 and rank2 =
8. the difference (d) = 5-8=-3" & vbNewLine & "You do this even if the answer
you get is negative. This is because the next step will rectify this problem"
```

```
    Case 2
    'third page
        For I = 1 tO 10
            txt_d(I).Text = d(I)
            txt_d2(I).Text = d2(I)
            txt_d(I).ForeColor = vbBlack
            txt_d2(I).ForeColor = vbRed
        Next I
        txt_sumd2.Text = "57"
        txt_sumd2.ForeColor = vbRed
        txt_rs.Text = ""
        txt_rs.ForeColor = vbBlack
        txt_cv.Text = ""
        txt_cv.ForeColor = vbBlack
        txt_r1(7).ForeColor = vbBlack
        txt_r1(8).ForeColor = vbBlack
        txt_r1(9).ForeColor = vbBlack
        txt_r1(10).ForeColor = vbBlack

        txt_help.Text = "In this step, we have to Square all the differences we
got in the last step." & vbNewLine & "This will get rid of any negative signs
we may have (as two negatives = a positive)" & vbNewLine & "For example:
reading 6 had a difference of -3. when we square it, we get +9" & vbNewLine &
"Once you have calculated all the d2 results, you need to add them all up. This
is important as we need the answer in the spearman's Rank Formula!"

    Case 3
    'fourth page
        For I = 1 tO 10
            txt_d(I).Text = d(I)
            txt_d2(I).Text = d2(I)
            txt_d(I).ForeColor = vbBlack
            txt_d2(I).ForeColor = vbBlack
        Next I
        txt_sumd2.Text = "57"
        txt_sumd2.ForeColor = vbBlack
        txt_rs.Text = "0.65"
        txt_rs.ForeColor = vbRed
        txt_cv.Text = ""
        txt_cv.ForeColor = vbBlack
        txt_r1(7).ForeColor = vbBlack
        txt_r1(8).ForeColor = vbBlack
        txt_r1(9).ForeColor = vbBlack
        txt_r1(10).ForeColor = vbBlack

        txt_help.Text = "Next, you have to complete the Spearman's Rank Formula
(this will be given to you in the exam)" & vbNewLine & "In the brackets, we
have to multiply the sum of d2 by 6. You then divide it by n-cubed - n." &
vbNewLine & "In this Example:" & vbNewLine & "6 x 57 = 342" & vbNewLine & "342
/ 990 = 0.34545454545..." & vbNewLine & "you then subtract this value from 1.
this would equal: 0.6545454545..." & vbNewLine & "In Biology, we like numbers
```

```
to be in 2 decimal places so the accepted answer would be 0.65"

    Case 4
    'fifth page
        For I = 1 tO 10
            txt_d(I).Text = d(I)
            txt_d2(I).Text = d2(I)
            txt_d(I).ForeColor = vbBlack
            txt_d2(I).ForeColor = vbBlack
        Next I
        txt_sumd2.Text = "57"
        txt_sumd2.ForeColor = vbBlack
        txt_rs.Text = "0.65"
        txt_rs.ForeColor = vbBlack
        txt_cv.Text = "0.65"
        txt_cv.ForeColor = vbRed
        txt_r1(7).ForeColor = vbBlack
        txt_r1(8).ForeColor = vbBlack
        txt_r1(9).ForeColor = vbBlack
        txt_r1(10).ForeColor = vbBlack

        txt_help.Text = "Lastly, we have to find the Critical Value. To do
this, you use a data table (which is given to you)." & vbNewLine & "You look at
how many pairs of measurements we have. In this case, we have 10." & vbNewLine
& "From this, we can see the Critical Value is 0.65." & vbNewLine & "That 's
all there is to it! Now try and complete the stats with your own results and
see how we interpret these numbers!"

    End Select
End If
End Sub
Private Sub Form_Load()
page = 0
'first page
For I = 1 tO 10
    d(I) = txt_d(I).Text
    d2(I) = txt_d2(I).Text
Next I

For I = 1 tO 10
    txt_d(I).Text = ""
    txt_d2(I).Text = ""
    txt_d(I).ForeColor = vbBlack
    txt_d2(I).ForeColor = vbBlack
Next I
txt_sumd2.Text = ""
txt_sumd2.ForeColor = vbBlack
txt_rs.Text = ""
txt_rs.ForeColor = vbBlack
txt_cv.Text = ""
txt_cv.ForeColor = vbBlack
```

```
txt_r1(7).ForeColor = vbRed
txt_r1(8).ForeColor = vbRed
txt_r1(9).ForeColor = vbRed
txt_r1(10).ForeColor = vbRed

txt_help.Text = "Firstly, you need to rank the data in ascending order (lowest
to highest). In the example above, the smallest percentage of plants found was
0 so this gets the rank of 1." & vbNewLine & "If you have two readings that
have the same value (readings 7-8, 9-10), you take the mean of the rank you
would give them." & vbNewLine & "For readings 7 and 8, you would have ranked
them 7 and 8. By taking the mean of these ranks, you get 7.5. You would then
give this rank to both readings" & vbNewLine & "You do this for both sets of
data!"

End Sub

Private Sub lbl_d_Click()
page = 1
'second page
For I = 1 tO 10
    txt_d(I).Text = d(I)
    txt_d2(I).Text = ""
    txt_d(I).ForeColor = vbRed
    txt_d2(I).ForeColor = vbBlack
Next I
txt_sumd2.Text = ""
txt_sumd2.ForeColor = vbBlack
txt_rs.Text = ""
txt_rs.ForeColor = vbBlack
txt_cv.Text = ""
txt_cv.ForeColor = vbBlack
txt_r1(7).ForeColor = vbBlack
txt_r1(8).ForeColor = vbBlack
txt_r1(9).ForeColor = vbBlack
txt_r1(10).ForeColor = vbBlack

txt_help.Text = "Once you have ranked the data, you need to find the difference
between the two sets of ranks. you subtract rank2 from rank1 to get the
difference" & vbNewLine & "For Example: in reading 6, rank1 = 5 and rank2 = 8.
the difference (d) = 5-8=-3" & vbNewLine & "You do this even if the answer you
get is negative. This is because the next step will rectify this problem"


End Sub

Private Sub lbl_sumd2_Click()
page = 2
'third page
For I = 1 tO 10
    txt_d(I).Text = d(I)
```

```
    txt_d2(I).Text = d2(I)
    txt_d(I).ForeColor = vbBlack
    txt_d2(I).ForeColor = vbRed
Next I
txt_sumd2.Text = "57"
txt_sumd2.ForeColor = vbRed
txt_rs.Text = ""
txt_rs.ForeColor = vbBlack
txt_cv.Text = ""
txt_cv.ForeColor = vbBlack
txt_r1(7).ForeColor = vbBlack
txt_r1(8).ForeColor = vbBlack
txt_r1(9).ForeColor = vbBlack
txt_r1(10).ForeColor = vbBlack

txt_help.Text = "In this step, we have to Square all the differences we got in
the last step." & vbNewLine & "This will get rid of any negative signs we may
have (as two negatives = a positive)" & vbNewLine & "For example: reading 6 had
a difference of -3. when we square it, we get +9" & vbNewLine & "Once you have
calculated all the d2 results, you need to add them all up. This is important
as we need the answer in the spearman's Rank Formula!"

End Sub

Private Sub lbl_spearmans_Click()
page = 3
'fourth page
For I = 1 tO 10
    txt_d(I).Text = d(I)
    txt_d2(I).Text = d2(I)
    txt_d(I).ForeColor = vbBlack
    txt_d2(I).ForeColor = vbBlack
Next I
txt_sumd2.Text = "57"
txt_sumd2.ForeColor = vbBlack
txt_rs.Text = "0.65"
txt_rs.ForeColor = vbRed
txt_cv.Text = ""
txt_cv.ForeColor = vbBlack
txt_r1(7).ForeColor = vbBlack
txt_r1(8).ForeColor = vbBlack
txt_r1(9).ForeColor = vbBlack
txt_r1(10).ForeColor = vbBlack

txt_help.Text = "Next, you have to complete the Spearman's Rank Formula (this
will be given to you in the exam)" & vbNewLine & "In the brackets, we have to
multiply the sum of d2 by 6. You then divide it by n-cubed - n." & vbNewLine &
"In this Example:" & vbNewLine & "6 x 57 = 342" & vbNewLine & "342 / 990 =
0.34545454545..." & vbNewLine & "you then subtract this value from 1. this
would equal: 0.6545454545..." & vbNewLine & "In Biology, we like numbers to be
in 2 decimal places so the accepted answer would be 0.65"
```

```
End Sub

Private Sub lbl_cv_Click()
page = 4
'fifth page
For I = 1 tO 10
    txt_d(I).Text = d(I)
    txt_d2(I).Text = d2(I)
    txt_d(I).ForeColor = vbBlack
    txt_d2(I).ForeColor = vbBlack
Next I
txt_sumd2.Text = "57"
txt_sumd2.ForeColor = vbBlack
txt_rs.Text = "0.65"
txt_rs.ForeColor = vbBlack
txt_cv.Text = "0.65"
txt_cv.ForeColor = vbRed
txt_r1(7).ForeColor = vbBlack
txt_r1(8).ForeColor = vbBlack
txt_r1(9).ForeColor = vbBlack
txt_r1(10).ForeColor = vbBlack

txt_help.Text = "Lastly, we have to find the Critical Value. To do this, you
use a data table (which is given to you)." & vbNewLine & "You look at how many
pairs of measurements we have. In this case, we have 10." & vbNewLine & "From
this, we can see the Critical Value is 0.65." & vbNewLine & "That 's all there
is to it! Now try and complete the stats with your own results and see how we
interpret these numbers!"

End Sub

Private Sub lbl_rank_Click()
page = 0
'first page
For I = 1 tO 10
    txt_d(I).Text = ""
    txt_d2(I).Text = ""
    txt_d(I).ForeColor = vbBlack
    txt_d2(I).ForeColor = vbBlack
Next I
txt_sumd2.Text = ""
txt_sumd2.ForeColor = vbBlack
txt_rs.Text = ""
txt_rs.ForeColor = vbBlack
txt_cv.Text = ""
txt_cv.ForeColor = vbBlack

txt_r1(7).ForeColor = vbRed
txt_r1(8).ForeColor = vbRed
txt_r1(9).ForeColor = vbRed
txt_r1(10).ForeColor = vbRed
```

```
txt_help.Text = "Firstly, you need to rank the data in ascending order (lowest
to highest). In the example above, the smallest percentage of plants found was
0 so this gets the rank of 1." & vbNewLine & "If you have two readings that
have the same value (readings 7-8, 9-10), you take the mean of the rank you
would give them." & vbNewLine & "For readings 7 and 8, you would have ranked
them 7 and 8. By taking the mean of these ranks, you get 7.5. You would then
give this rank to both readings" & vbNewLine & "You do this for both sets of
data!"

End Sub
```

# Variable Listing:

## Global Variables:

| Variable | Type | Content |
|---|---|---|
| Sim1.readings | Integer * 20 | Stores the values from the simulation/ manually inputted and takes them to the stats |
| stats | boolean | Determines whether the simulation for the stats has been completed or not. |

## Local Variables:

| Variable | Type | Location | Content |
|---|---|---|---|
| TOHTotal | integer | Frm_Chi2 | Stores the total number of plants at Top Of Hill |
| BOHTotal | integer | Frm_Chi2 | Stores the total number of plants at Bottom of Hill |
| TOHobserved | integer | Frm_Chi2 | Stores the observed value for Top of Hill |
| BOHobserved | integer | Frm_Chi2 | Stores the observed value for Bottom of Hill |
| expected | double | Frm_Chi2 | Stores the expected value |
| TOHdifference | double | Frm_Chi2 | Stores the difference for Top of Hill |
| BOHdifference | double | Frm_Chi2 | Stores the difference for Bottom of Hill |
| TOHDifferenceSquared | double | Frm_Chi2 | Stores the difference2 for Top of Hill |
| BOHDifferenceSquared | double | Frm_Chi2 | Stores the difference2 for Bottom of Hill |
| TOHDSoverE | double | Frm_Chi2 | Stores the difference2/E for Top of Hill |
| BOHDSoverE | double | Frm_Chi2 | Stores the difference2/E for Bottom of Hill |
| CHI | double | Frm_Chi2 | Stores the $Chi^2$ result |
| verify1 | boolean | Frm_Chi2 | |
| correct | integer | Frm_Chi2 Frm_Spearmans | When the user inputs a correct answer, the value increases by 1. When the value reaches a certain number, the stats ends. |
| dof | integer | Frm_Chi2 | Stores te value of the Degrees Of Freedom |

| Variable | Type | Location | Content |
|---|---|---|---|
| page | integer | Frm_chi2_Help<br>Frm_spearmans_help<br>Frm_StandardError_help | Used to determine which "page" the user is on, on the help page. Changed the text according to the number |
| Sim_Begin | boolean | Frm_simulation1<br>Frm_simulation2<br>Frm_simulation3 | Is false, means the simulation has not yet started so can load all the boxes and andom numbers needed without inteferring with the simulation in progress |
| TOHrecorder | integer | Frm_simulation1 | Stores how many readings have been taken for Top of Hill |
| BOHrecoder | integer | Frm_simulation1 | Stores how many readings have been taken for Bottom of Hill |
| Rnd_labelX | integer | Frm_simulation1<br>Frm_simulation2 | Stores the random number for the X-Coordinate for the simulation's random numbers |
| Rnd_labelY | integer | Frm_simulation1<br>Frm_simulation2 | Stores the random number for the Y-Coordinate for the simulation's random numbers |
| Random_equationx | integer | Frm_simulation1<br>Frm_simulation2 | Stores the value of the x-position of the box that is to be selected by the user |
| Random_equationy | integer | Frm_simulation1<br>Frm_simulation2<br>Frm_simulation3 | Stores the value of the y-position of the box that is to be selected by the user |
| boxX | integer | Frm_simulation1<br>Frm_simulation2<br>Frm_simulation3 | Stores the value of the x-positionof the selected box |
| boxY | integer | Frm_simulation1<br>Frm_simulation2<br>Frm_simulation3 | Stores the value of the y-position of the selected box |
| Boxcheck | boolean * 100<br>Boolean * 60 | Frm_simulation1<br>Frm_simulation2<br>Frm_simulation3 | Assigned to ever box in the simulation. If false, the box has not been selected. If true, box has been selected |
| RND_Number | Integer * 100<br>Integer * 60 | Frm_simulation1<br>Frm_simulation2<br>Frm_simulation3 | Stores the random number that is given to each box at the start that determines which image is loaded onto the box |

| Variable | Type | Location | Content |
|---|---|---|---|
| Pointer | integer | Frm_simulation1<br>Frm_simulation2<br>Frm_Spearmans | Increases with each readings and is used to assign the readings their position in the global variable |
| Percent | integer | Frm_simulation1<br>Frm_simulation2<br>Frm_simulation3 | Works out the percentage of accuracy in the simulation based on the user's inputted readings to the actual values |
| SimFinished | boolean | Frm_simulation1<br>Frm_simulation2<br>Frm_simulation3 | Once the simulation has been completed, the value will change to True. Only then can you proceed to the stats. |
| NFrecorder | Integer | Frm_simulation2 | Records how many readings have been taken for North Facing |
| SFrecorder | Integer | Frm_simulation2 | Records how many readings have been taken for South Facing |
| selection | integer | Frm_simulation3 | Stores the value of expected index of the selected box |
| recorder | integer | Frm_simulation3 | stores the value of how many readings have been taken |
| lane | integer | Frm_simulation3 | Stores the value of the randomly generated number that tells the user which, of three lanes, they'll be taking readings from |
| LERandomNumber | integer | Frm_simulation3 | Stores the vaue of the randomly generated light intensity value |
| bothReadings | boolean | Frm_simulation3 | If true, both readings for light intensity and percentage cover have been completed. |
| Ppointer | integer | Frm_simulation3 | Used to assign the readings of percentage cover a location in the global variable |
| Lpointer | integer | Frm_simulation3 | Used to assign the readings of light intensity a location in the gobal variable. |

| Variable | Type | Location | Content |
|---|---|---|---|
| percePlant | Integer * 15 | Frm_Spearmans | An array that stores only readings for percentage cover |
| Lux | Integer * 15 | Frm_Spearmans | An array that stores only readings for light intensity |
| Difference | Double | Frm_Spearmans | Stores the value of the difference between the two ranks |
| Sumd | Double | Frm_Spearmans | Stores the value of the sum total of the difference values |
| Spearmans | Double | Frm_Spearmans | Stores the value of the spearmans rank correlation formula |
| CriticalValue | Double | Frm_Spearmans | Stores the value of the critical value |
| Complete | Boolean | Frm_Spearmans | As long as the variable is set to false, the stats will continue. It turns to true when the stats has been completed |
| D | Double * 10 | Frm_Spearmans_help | Records the results in the labels (for the "d" values only) so that they can be changed according to the page number |
| d2 | Double * 10 | Frm_Spearmans_help | Records the results in the labels (for the "d2" values only) so that they can be changed according to the page number |
| NFmean | Double | Frm_StandardError | Stores the value of the mean of the results for North Facing |
| SFmean | Double | Frm_StandardError | Stores the value of the mean of the results for South Facing |
| NFStandardDeviation | Double | Frm_StandardError | Stores the vaue of the standard deviation for the North Facing results |
| NFStandardError | Double | Frm_StandardError | Stores the value of the standard error for the North Facing results |
| SFStandardDeviation | Double | Frm_StandardError | Stores the value of the standard deviation of the South Facing results |

| Variable | Type | Location | Content |
|---|---|---|---|
| SFStandardError | Double | Frm_StandardError | Stores the value of the standard error of the South Facing results |
| ConfidenceLimits | double | Frm_StandardError | Stores the value of the confidence limits |
| Overlap | Boolean | Frm_StandardError | If true, then there is an overlap in the two data sets |
| Yclick | Boolean | Frm_StandardError | If true, the "yes" button has been selected in question to there being an overlap or not |
| Nclick | Boolean | Frm_StandardError | If true, the "no" button has been selected in question to there being an overlap or not |
| Aclick | Boolean | Frm_StandardError | If true, the "accept" button has been selected in question to accepting the null hypothesis or not |
| Rclick | Boolean | Frm_StandardError | If true, the "reject" button has been selected in question to rejecting the null hypothesis or not |
| Range1 | Double | Frm_StandardError | Stores the value of the lower value of the range for North Facing |
| Range2 | Double | Frm_StandardError | Stores the value of the higher value of the range for North Facing |
| Range3 | double | Frm_StandardError | Stores the value of the lower value of the range for South Facing |
| Range4 | Double | Frm_StandardError | Stores the value of the higher value for South Facing |

## Subroutine and Function Listing:

### Main Menu:

**Private Sub cmd_chi2_Click()**
Takes the user to the Chi$^2$ stats page

**Private Sub cmd_simulation1_Click()**
Takes the user to the first simulation

**Private Sub cmd_simulation2_Click()**

Takes the user to the second simulation

**Private Sub cmd_simulation3_Click()**
Takes the user to the third simulation

**Private Sub cmd_spearmans_Click()**
Takes the user to the Spearman's Rank stats page

**Private Sub cmd_standard_Click()**
Takes the user to the Standard Error stats page

**Private Sub Form_Load()**
Initialises the forms by setting their sizes. This is to mainly prevent the forms from being resized at the start of the program. It also sets every value in the global array to 0 and sets the global Boolean to false.

**Simulation 1:**

**Private Sub Form_Load()**
When the form first loads, it initialises all the variables and labels that are changed through the course of the simulation. This is so that the simulation will always begin anew when the form loads and nothing is retained after having left the simulation. It also generates random numbers for each of the 100 boxes in the simulation which then loads an image onto them.

**Private Sub box_Click(Index As Integer)**
This event occurs when one of the 100 boxes has been selected. It checks to see if the correct box has been selected by using equation to work out what their position in the form should be. If the correct box has been selected, a larger image box will appear and the correct image needs to be loaded onto that also.

**Private Sub cmd_NextReading_Click()**
When the reading has been entered by the user and the button pressed, the value gets recorded into the global the teacher (sim1.readings). The counter then increases by one. When it reaches 10, the simulation refreshes itself completely and all new images are loaded onto the text boxes.

**Private Sub cmd_rndnumber_Click()**
Generates two random numbers when pressed that are used to give the coordinates for the box the user is to select

**Private Sub cmd_stats_Click()**
Takes the user to the stats page ($Chi^2$) after the simulation has been completed

**Private Sub cmd_back_Click()**
Takes the user back to the main menu

**$Chi^2$ stats page:**

**Private Sub Form_Load()**
When the form loads, it has to initialise everything first. it loads in the results from the global array into a results table. Afterwards, it works out all of the stats before the user has done them and stores each of the values into a variable that can be used to compare the results the user inputted.

**Private Sub cmd_check_Click()**
This checks everything the user entered. Every box that required an input is checked. If it is correct (the value in the text box matches that of the value of the variable worked out in the form load) the background colour of the box will turn green. If wrong, however, the background colour will change to red. When all the boxes are green and everything is correct, an evaluation box will appear at the end. This explains to the user what the stats mean to the user.

**Private Sub cmd_own_Click()**
If the user came to the stats page without having completed the simulation, the user will need to enter their own data or random data. This button will take the user's self inputted data and transfer to the global array so that it can then be used in the stats. Afterwards, the stats is completed with the new data and stored in separate variables.

**Private Sub cmd_rnd_Click()**
As I mentioned above, if the user has not completed the simulation, they must either generate a random set of results or input their own. If they chose to generate a random set of results, the table of results will be populated with random data (within certain bounds so it the data is somewhat realistic). The new data will then be used in the stats and each result from a calculation assigned to a separate variable.

**Private Sub cmd_help_Click()**
Takes the user to the help page for the $Chi^2$ stats

**Private Sub cmd_back_Click()**
Takes the user back to the main menu page

**Simulation 2:**

**Private Sub Form_Load()**
When the form first loads, it initialises all the variables and labels that are changed through the course of the simulation. This is so that the simulation will always begin anew when the form loads and nothing is retained after having left the simulation. It also generates random numbers for each of the 100 boxes in the simulation which then loads an image onto them.

**Private Sub box_Click(Index As Integer)**
This event oocurs when one of the 100 boxes has been selected. It checks to see if the correct box has been selected by using equation to work out what their position in the form should be. If the correct box has been selected, a larger image box will appear and the correct image needs to be loaded onto that also.

**Private Sub cmd_NextReading_Click()**
When the reading has been entered by the user and the button pressed, the value gets recorded into the global the teacher (sim1.readings). The counter then increases by one. When it reaches 10, the simulation refreshes itself completely and all new images are loaded onto the text boxes.

**Private Sub cmd_rndnumber_Click()**
Generates two random numbers when pressed that are used to give the coordinates for the box the user is to select

**Private Sub cmd_stats_Click()**
Takes the user to the stats page (Standard Error) after the simulation has been completed

**Private Sub cmd_back_Click()**
Takes the user back to the main menu

**Standand Error:**

**Private Sub Form_Load()**
When the form loads, it has to initialise everything first. it loads in the results from the global array into a results table. Afterwards, it works out all of the stats before the user has done them and stores each of the values into a variable that can be used to compare the results the user inputted.

**Function NFStdDev(mean As Double) As Double**
This function takes the mean of the North Facing results and applies the standard deviation formula on to it. This will then return the standard deviation as a single value.
Parameters: takes the mean of the North Facing results
Output: Standard Deviation of the North Facing results

**Function SFStdDev(mean As Double) As Double**
This function takes the mean of the South Facing results and applies the standard deviation formula on to it. This will then return the standard deviation as a single value
Parameters: takes the mean of the South Facing results
Output: Standard Deviation of the South Facing results

**Private Sub lbl_yOverlap_Click()**
If clicked, the background colour of the label will turn yellow. If the "No" button has already been selected, then its background colour will change to white instead.

**Private Sub lbl_nOverlap_Click()**
If clicked, the background colour of the label will turn yellow. If the "Yes" button has already been selected, then its background colour will change to white instead.

**Private Sub lbl_accept_Click()**
If clicked, the background colour of the label will turn yelow. If the "reject" button has already been selected, then its background colour will change to white instead

**Private Sub lbl_reject_Click()**
If clicked, the background colour of the label will turn yellow. If the "accept" button has already been clicked, then its background colour will change to white instead

**Private Sub cmd_check_Click()**
This checks everything the user entered. Every box that required an input is checked. If it is correct (the value in the text box matches that of the value of the variable worked out in the form load) the background colour of the box will turn green. If wrong, however, the background colour will change to red. When all the boxes are green and everything is correct, an evaluation box will appear at the end. This explains to the user what the stats mean to the user.

**Private Sub cmd_own_Click()**
If the user came to the stats page without having compelted the simulation, the user will need to enter their own data or random data. This button will take the user's self inputted data and transfer to the global array so that it can then be used in the stats. Afterwards, the stats is completed with the new data and stored in separate variables.

**Private Sub cmd_rnd_Click()**
As I mentioned above, if the user has not completed the simulation, they must either generate a random set of results or input their own. If they chose to generate a random set of results, the table of results will be populated with random data (within cetain bounds so it the data is somewhat realistic). The new data will then be used in the stats and each result from a calculation assigned to a separate variable.

**Private Sub cmd_help_Click()**
Takes the user to the help page for the Chi$^2$ stats

**Private Sub cmd_back_Click()**
Takes the user back to the main menu page

**Simulation 3:**

**Private Sub Form_Load()**
When the form first loads, it initialises all the variables and labels that are changed through the course of the simulation. This is so that the simulation will always begin anew when the form loads and nothing is retained after having left the simulation. It also generates random numbers for each of the 60 boxes and loads the correct image onto them, according to the random number they were given.

**Private Sub box_Click(Index As Integer)**
This event oocurs when one of the 60 boxes has been selected. It checks to see if the correct box has been selected by using an equation to work out what their position in the form should be. If the correct box has been selected, a larger image box will appear and the correct image needs to be loaded onto that also.

**Private Sub cmd_NextReading_Click()**
When the both readings (for percentage cover and light intensity) have been inputted, the two values will be recorded into the global the teacher (sim1.readings). The counter then inceases by one. When it reaches 10, the simulation ends and no more readings can be taken

**Private Sub tmr_LightIntensity_Timer()**
Generates a random number every 10 seconds, for the light intensity and then displays the value onto the form for the user to take a reading from.

**Private Sub cmd_stats_Click()**
Takes the user to the stats page (Spearmans Rank) after the simulation has been completed

**Private Sub cmd_back_Click()**
Takes the user back to the main menu

**Spearman's Rank:**

**Private Sub Form_Load()**
When the form first loads, it populates the results table and some of the stats table with the two sets of values in the global array. It then fills the two arrays (percPlant and lux) with the corresponding values from the global array.

**Private Sub cmd_check_Click()**
This will check everything that requires the user to enter data. If the answer in the text box matches the correct, expected answer, the background colour of the box will turn green. If it's wrong, however, it will turn red. If all the values are correct, an evaluation text box will appear, telling the user what the stats has shown them.

**Public Sub Sorting(ByRef Values As Variant)**
This event sorts the two arrays (percPlant and lux) into ascending order. This is because the stats requires the data to be sorted into ascending order.

Parameters: the program will take the arrays, percPlant and lux, and will sort them into ascending order
Output: will have sorted both arrays in ascending order so that they can be compared with the users sorted data

**Function spearmans(sum As Double)**
This function takes the value of the sum total of the differences in ranks and applies it to the Spearman's Rank formula.
Parameters: receives the vaue of the sum of the differences
Output: returns the Spearman's Rank value of the data

**Private Sub cmd_own_Click()**
If the user came to the stats page without having compelted the simulation, the user will need to enter their own data or random data. This button will take the user's self inputted data and transfer to the global array so that it can then be used in the stats. Afterwards, the stats is completed with the new data and stored in separate variables.

**Private Sub cmd_rnd_Click()**
As I mentioned above, if the user has not completed the simulation, they must either generate a random set of results or input their own. If they chose to generate a random set of results, the table of results will be populated with random data (within cetain bounds so it the data is somewhat realistic). The new data will then be used in the stats and each result from a calculation assigned to a separate variable.

**Private Sub cmd_help_Click()**
Takes the user to the help page for the $Chi^2$ stats

**Private Sub cmd_back_Click()**
Takes the user back to the main menu page

**$Chi^2$ Help:**

**Private Sub Form_Load()**
Displayes the first page of the help page upon the loading of the form

**Private Sub cmd_cont_Click()**
Increments the page by one and takes the user through the help form step by step.

**Private Sub cmd_prev_Click()**
Decreases the page by one and allows the user to go back through the help form step by step.

**Private Sub lbl_o_Click()**
Immediately takes the user to the first page of the help form

**Private Sub lbl_e_Click()**
Immediately takes the user to the second page of the help form

**Private Sub lbl_oe_Click()**
Immediately takes the user to the third page of the help form

**Private Sub lbl_oe2_Click()**
Immediately takes the user to the fourth page of the help form

**Private Sub lbl_oe2e_Click()**
Immediately takes the user to the fifth page of the help form

**Private Sub lbl_chi_Click()**
Immediately takes the user to the sixth page of the help form

**Private Sub lbl_dofcv_Click()**
Immediately takes the user to the seventh of the help form

**Standard Error Help:**

**Private Sub Form_Load()**
Starts at the very first help page and initialises everything so navigating the help form will be easy and no problems should occur

**Private Sub cmd_cont_Click()**
Increments the page by one so the user can follow the guide in a step by step sort of way

**Private Sub cmd_prev_Click()**
Decreases the page by one so the user can backtrack through the guide in a step by step sort of way

**Private Sub lbl_m_Click()**
Takes the user to the first page of the help form

**Private Sub lbl_st_Click()**
Takes the user to the second page of the help form

**Private Sub lbl_se_Click()**
Takes the user to the third page of the help form

**Private Sub lbl_cl_Click()**
Takes the user to the fourth page of the help form

**Private Sub lbl_range_Click()**
Takes the user to the fifth page of the help form

**Private Sub lbl_overlap_Click()**
Takes the user to the sixth page of the help form

**Private Sub cmd_back_Click()**
Takes the user back to the main menu

**Spearman's Rank Help:**

**Private Sub Form_Load()**
Loads the first page for the user and initialises all the text boxes and labels so that there are no errors whilst the user is navigating

**Private Sub cmd_cont_Click()**
Increments the page by one so the user can go through the guide in a step by step sort of way

**Private Sub cmd_prev_Click()**
Decreases the page by one so the user can backtrack through the guide in a step by step sort of way

**Private Sub lbl_rank_Click()**
Immediately takes the user to the first page in the help form

**Private Sub lbl_d_Click()**
Immediately takes the user to the second page in the help form

**Private Sub lbl_sumd2_Click()**
Immediately takes the user to the third page in the help form

**Private Sub lbl_spearmans_Click()**
Immediately the user to the fourth page in the help form

**Private Sub lbl_cv_Click()**
Immediately takes the user to the fifth page in the help form

**Private Sub cmd_back_Click()**
Takes the user back to the main menu

# Evaluation:

## Success of Objectives:

The following list of processes is the user requirements that were asked for in the interview process and the observation of current practise and documents. I will evaluate each of these processes to make sure that what I set out to achieve was what was achieved. To do this I will use the following criteria:

- How easy the system was to use – This asks how much help I have put into the system to make it user friendly.
- The reliability of the system – This asks whether there are still bugs present in each process.
- The effectiveness of the system – This asks whether the program works effectively and quickly.

### The Simulations:

One problem with the simulations (I think anyway) is that they may not be completely clear for a new user to understand right off the bat. I'm making the assumption that the teacher will have gone through the use of the program with the students beforehand. Also, the teacher will have been instructed in its use as well. Nonetheless, I feel it could appear rather confusing for a new user to understand and the help button may not be clear at a first glance. This is perhaps a limitation for all the simulations. To amend this possible error, I might add an introductory help page at the start of each simulation. This way, the user can understand the program from the beginning without any frustrations. Of course, it would be unfair to ask every user to read through the page before they can proceed so there will be an exit button for those that have used the program before. Aside from this issue, everything is clearly labelled, so that the user knows what it is that they're being shown or what they're meant to type in. There's also tooltips on key text boxes and labels which the user can get to by hovering over the text box or label in question. When the user enters a reading, the counter will increase by 1. This is shown as a fraction where 10 is the denominator and the number of readings is the numerator. When the label changes to 10/10, the simulation, understandably, will stop and no more readings can be taken. The user can also return to the main menu if they so wish and can go to the stats page after the simulation have been completed

To keep the simulations reliable and valid, only numerical data can be entered and data needs to be present to be entered. This prevents the user from trying to enter nothing into the program or by trying to enter data that isn't numerical. This would cause an error message to pop up and stop the program from running. To prevent this, I have ensured appropriate validation procedures have been put in. every box that the user enteres data into will be checked for being a numerical value and if it isn't just a blank space. If one of the two constraints is not met, an error message will pop up on the screen, telling the user that the data the entered is not numerical or that no data has been entered. The simulation will not proceed until an acceptable value has been inputted

For the most part, the simulation is effective and quick. The image box appears instantly when the user selects the correct one and the data is accepted into the program instantly also. However, when the program is assigning a number to each image box and then loading an image based on that number, the program freezes momentarily. This also happenes for simulation 1 and 2 when 10 readings have been taken and the grid needs to refresh itself with new data. This is merely because the program has to randomly generate 100 numbers and then load 100 images onto an array of 100 boxes. The program can't perform the task very quickly because of the size it has to deal with. This will be a problem that will only be fixed if the college's hardware is upgraded so the computers can perform this task much faster. However, I could help by changing the mouse symbol to the hourglass icon. This would indicate to the user that the program is doing something and that they'll need to wait for a little but for the process to be completed. The implementation of the "Back" button allows the user to return to the main menu whenever they wish. This makes the program more effective because it means the user can leave the simulation whenever they want; in case they chose the wrong simulation or don't want to continue on to the stats section of the program.

**Transferring Data to the Stats and adding it into a Table of Results**

I can't expect every user to have to record the result they took and when they then go into the stats section, copy out the results they recorded. This would be ludicrous and honestly, a waste of time and would render the point of the program nearly useless. Thus, I have made this automatic for the user, they need only concentrate on the simulation/stats itself and not on the results they obtained. I needed to show the user that the results were being taken of course so I included a counter that counts up with every reading that is taken. Also, I added a text box at the bottom that shows the readings the user took against the actual values they were taking readings from. This is all to allow the user to see that the program is recording their results correctly and also to show where they may have gone wrong. When the user enters the stats, from the simulation, the results will automatically be generated into a table of results and, depending on the stats, some of the stats table will be filled in also. This is done as soon as the user loads the stats from up so they can get straight to the stats without having to wait.

In terms of reliability, I have encountered no problems with transferring the readings and they have always been correct. During my development testing and alpha testing I made sure to test everything and the results definitely carried across without a problem.

I think this is rather effective insofar as it is very quick. As soon as the form loads, the results have been added into the results table. It records the users entered data and not the actual data. This is important as the user won't have a way of knowing the exact number when they go out into the field to do their field work.

**Statistical Tests:**

The screen is somewhat crowded with boxes and tables because I wanted to break down the stats in a way that made sense and could be followed easily by anyone. However, I did include a small paragraph of writing beneath the table of results that gives the user a rough idea of what they're meant to do. Not only do I want to teach the user how to work out the stats, I want the user to be able to think about what they're meant to do to solve the problem. This is why I gave little direction. However, if the user is genuinely stuck, the help button on the top of the form will open up a form that will give a step by step guide of the stats using example data. All the tables and text boxes are correctly and precisely labelled so that the user knows what they're meant to enter into that specific text box. Another feature to help the user is that I include tool tips on almost all of the text boxes and labels. This will allow the user to get additional help on what they're required to enter. Such things include descriptions of what formula the user should use, or what a formula may mean. Other things tell the user that the answer the program will accept needs to be to 2.d.p. Lastly, if the user has come straight from the main menu, they won't have any results to do the stats on. To fix this problem, I added a small error box that will pop up in this instance. It will tell the user to either enter their own data into the text box or to generate a set of random (but believable) results to work with. The user cannot proceed with the stats until this has been completed. This task can be accomplished by entering data into the results table and then selecting the input own data button or by selecting the "generate random numbers" button, which fills in the table for you. The user's own data also goes through validation checks to ensure that no empty spaces have been left and that all the data is numerical. If it isn't an error message will pop up, telling the user to complete the table properly.

The form seems reliable in the sense that everything is checked and validated. When the "check" button is pressed by the user, the program will look through every text box, in order, and check if it's correct. If checks to make sure that the data the user entered is not blank and consists of only numerical data. If not, an error message will pop up and tell the user what the problem was. The text box with the incorrect value in it will also change colour to red and will end the checking procedure. This contrasts against the green so the mistake can easily be spotted by the user. Also, the stopping of the checking procedure means no more text boxes are checked after the mistake is found. This means the boxes will remain white so the user can spot the mistake more easily.

The stats forms are effective because they are instant. When you click on the "check" button, the results are displayed instantly and the evaluation text box appears instantly also, depending on whether you got it all right of course.

**Non-Linearity:**

I tried to make sure that the user didn't feel restricted in any way and that they had freedom of choice in what they did in the program. With this in mind, I have included in a "back" button on every form that will take the user back to the main menu when pressed. The only time this does not occur is when the user is on the help page. The back button will instead take the user back to the stats that they were currently on. Also, the user does not have to complete the simulation to access the stats, nor does the user need to go to the stats after the simulation has been completed. The only limitation I can think of is that you can't access every form from every other form. However, this isn't much of an issue I believe because of the project itself. You would have no need of accessing the Spearman's Rank form if you're currently on the simulation 1 form. Because of this, I think it is acceptable that the user can return to the main menu every time.

The navigation is very reliable in that it takes the user to the correct form. Each button is correctly labelled also, which also helps to reduce any confusion the user may have. The navigation is also instant in that it will take you to the form instantly and there is little to no lag. The only exception to this would be the simulations. However, this is not because of the navigation acting slowly but rather that the loading of the images for the boxes slows the process down.

I think the navigation is effective because it is very short and simple. The user can easily identify which simulation goes with which stats test and there isn't any unnecessary space and boxes that gets in the way and confuses things. The only potential limitation is that I don't have many confirmation boxes of sorts that ask if the user would like to proceed. The user may accidentally click on the "back" button during the stats when they actually wanted to click on the help button. They'll now have lost their readings and the stats that they may have already completed.

## Email from End User:

Thanks for all the effort you have put into making this system for us, and the minor changes you made during the Beta Testing I did.

The program has been installed on the computers and we are glad to say that it is fully functional. The department are very pleased with the program and have gone through it a few times to familiarise themselves with it. We have already begun teaching and some of the A2 students tried the program and from what I have heard, it was well received.

There are some things that have been picked up on by the students that we feel should be addressed to make the program easier to handle.

Whilst we thought that the Simulations of the Quadrat Sampling and the Belt Transect would be ok as they were, some students were a little baffled when it came to doing the simulation. The main problem was with the Interrupted Belt Transect. We think that it would be best to add the help page – like you have for the statistics pages.

Perhaps adding a feature to allow the user to go directly to the help pages would be beneficial, instead of relying on the student to find the help page themselves.

These are the only alterations we would like seen in the program at the time being. However, I will notify you of any problems or alterations we would like seen in the future.

[redacted] thank you for all your effort you have put in to making this program for us and I hope to see you later in the year with the updated version.

## Response to Email:

The user email that was sent to me indicates that they are quite happy with how the progam has turned out. However, upon using the program, there were some alterations that he would like to be added.

The system was implemented over the Christmas Break, ready for use when the college term started once more. During the time, the teachers were able to use the program themselves and see how it worked and how it could be a benefit to them as a teaching aid.

The problems they identified were:

1. No help page for the simulations. This issue was brought up during beta testing but disregegarded as not a necessity. However, upon using the sysem, students have shown difficulty in using the simulation straight away.

Whilst we thought that the Simulations of the Quadrat Sampling and the Belt Transect would be ok as they were, some students were a little baffled when it came to doing the simulation. The main problem was with the Interrupted Belt Transect. We think that it would be best to add the help page – like you have for the statistics pages.

This is a simple solution and something I thought was likely to come up when the program was put in full use. Making the help page will be simple as I will only have to describe how to complete – much like I did for the Beta Testing of my end user.

I will include diagrams and annotations as well but it won't be like the stats, insofar that it won't include a step by step guide. I think this would be a fairly pointless exercise as the students only need to be told how to work the simulation once. The rest of the results follow the same pattern.

2. Adding a feature which would allow the user to go straight to the help page in some way. My end user thinks it may be somewhat unclear as to where the help is, if the student isn't looking.

Perhaps adding a feature to allow the user to go directly to the help pages would be beneficial, instead of relying on the student to find the help page themselves.

This is a simple addition also but one I think I should have implemented from the beginning. To do this, I would change the "Back" button on the help forms that already exist to going back to the main menu and then add another button that will take the user to the appropariate page. I will then include extra buttons on the main menu to incorporate these help pages as well, allowing the user to access them directly from when they first open up the program. Like I mentioned in my evaluation above, I think adding a way of showing the help page, or telling the user where it is located, at the start of the page may be a useful addition. This way, they know ehere they can access it, even if they don't think they will need it straight away.

## Desirable Extensions:

Because of time constraints, I had to shorten the program down quite a bit from what I had originally envisioned.

A feature I left out was the file handling section of the data. This would include a save/ edit/ load and delete button for data that the user has either collected from the simulation or created on their own. The process would be fairly simple. I would create another global variable that is an array with dimesnsions of 20. I would then create another form called, "Results Table," or something similar, where the user can go to review their data they received from the simulation, or to create their own. In this form, they will have the options to save the data, delete it, edit it or to load data.

Saving the data would be easy as the data that needs to be stored is already in the correct order. I would simply need to store it serially into a file. The data that would be saved would be the 20 different results that are contained within this new global array. I would create a loop that would cycle through the array and store each value into the next available location within the file to be created. The user will also be able to name the file to be created so that they can easily find it if they would like to edit the data or delete it later on.

Loading the data would be completed by allowing the user to search for the file that they want to load. When they have a file they want to load, the results table on the form will be cleared of all existing data and the text boxes will load the data from within the file. Just like it was with saving, the data can be accessed serially. This means that I can load the data in the file into the global array and then from that array, I can loop through and populate all the text boxes with the data.

Editing the data would be completed by simply making the results table full of text boxes. This will allow the user then to edit the data to what they want. They can then press "save" to save the data they edited or go directly to the corresponding statistical section.

Deleting previously saved data would require the user to have first loaded the data. Once they have completed this action, they will be allowed to delete the data. When the delete button has been pressed, it will clear every space of the file that was loaded so that it will be deleted. The data that has already been loaded into the results table will also be cleared.

I would also need to include a notification method that will pop up when the user clicks on any of the buttons. This is to make sure that they are certain with the action they want to perform but also to prevent the possibility that they pressed the button by accident. Especially with the delete and save buttons. The technician might not be very pleased if he finds the same file created a hundred different times on the network. With the delete button, it would look very unprofessional if the teacher comes into class, thinking they have data already saved and can simply load it up when they get into class, and the data isn't there and they have to recreate it.

Something that I spoke often about throughout the creation of the project was that I would change the graphics of the simulation so that it looked more lifelike and wasn't made up of a bunch of red dots on a green background. Alas, I have still not completed this mundane task. I want the graphics to be somewhat realistic and to achieve this, I had planned on having 3 or so variations of the same graphic. I thought a way to achieve this would be to have the files all have the same name but all use a different file extension. I would then use another random number generator ranging from 1 to 3. Using the two generated numbers (for the name of the file and then for the extension), they would pick out the correct image and load it into the simulation. The potential limitation of this is that each file is indeed a picture and therefore has a fairly high file size. Having so many large files could cause the program to be somewhat slower. Also, the simulation startup and when it refreshed would take longer than it already does. Just so that people don't think the program has crashed, I will change the mouse icon to the hour glass. At least this way, the user knows the program is doing something and to wait.

# User Guide:

# Ecology and Field Work Program

## Minimum System Requirements:

The minimum system requirements have been worked out from the minimum system requirements required to run Windows XP operating system. This is the Operating system that is currently installed on the College's computers. Even if the college upgrades their computers and the operating system they use, the program will still run because it is design to run on a slower system.

### Minimum software requirements:

The minimum software requirements for this program are Windows XP.

### Minimum hardware requirements:

- Pentium 233-megahertz (MHz) processor or faster.
- At least 64 megabytes (MB) of RAM
- At least 1.5 gigabytes (GB) of available space on the hard disk
- CD-ROM or DVD-ROM drive
- Keyboard and a Mouse or some other compatible pointing device
- Video adapter and monitor with Super VGA (800 x 600)
- Sound card
- Speakers or headphones

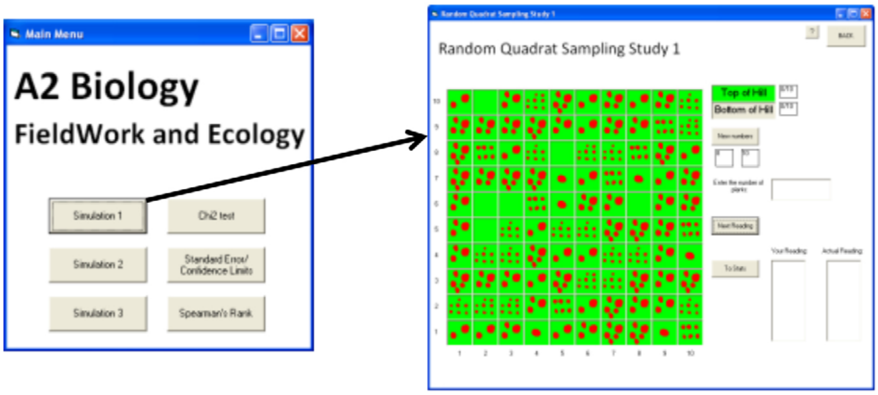
## Installation Guide:

1. Insert the CD-ROM into the disc drive
2. Go to "Start" -> "My Computer" and double click on the CD-ROM drive icon. The disc drive menu should now open up. Double click on the icon that is called "Setup.exe" and the setup launch wizard should pop up on the screen.
3. Follow the instructions that are displayed on the screen.
4. The installation will finish and you'll be given the option of adding a shortcut icon to the desktop so you can have quick access to the program. After this, the installation will be complete and you can now use the program.

# Operating Instructions:

## Navigation:

This program uses a menu based system. This means that clicking on a button will open up another page according to the button that you pressed.

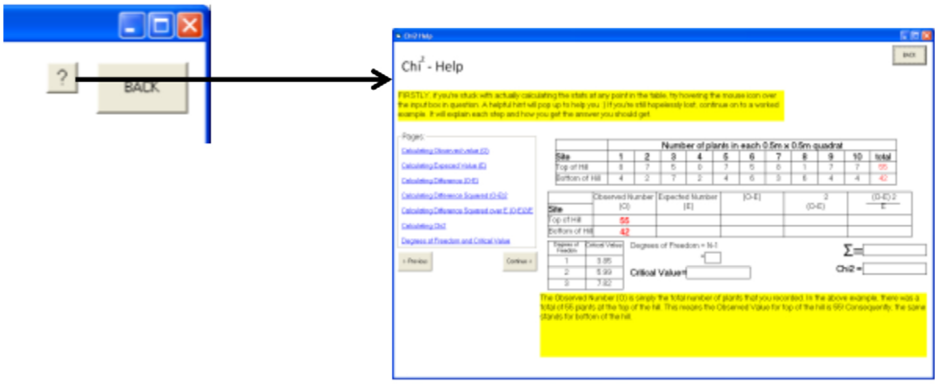



Some of the forms will require you to finish the form you're currently on first. For example, the simulation needs to be completed before you can go to the stat's using incomplete data.

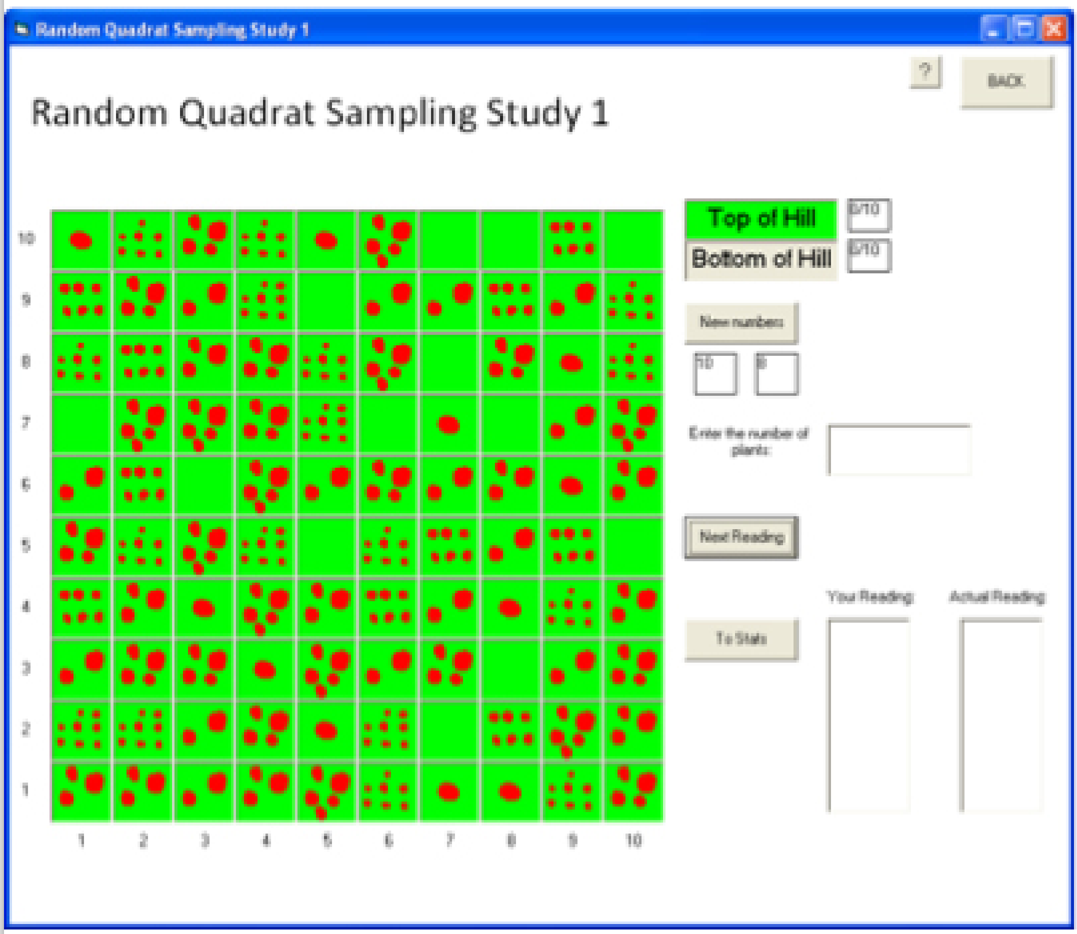

## How to use the Forms:

Most of the forms will have on screen help. Hovering over certain text boxes will display a "tool tip." This will be a small info box that will pop up that will give you a little extra information if you require it.

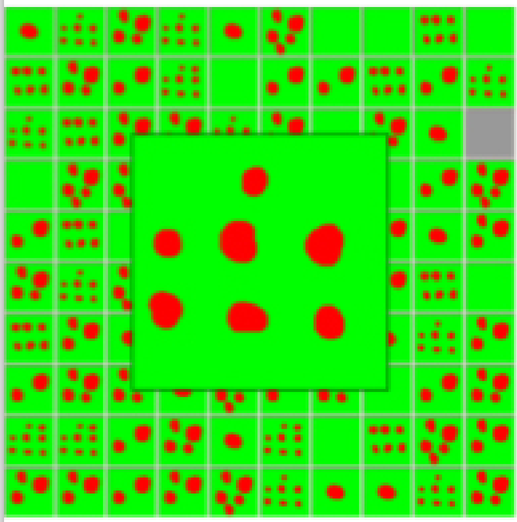

I have also included a button on the top right hand corner (displayed as a "?" image) that will take you to a help page. These help pages will guide you through the form, using a worked example.



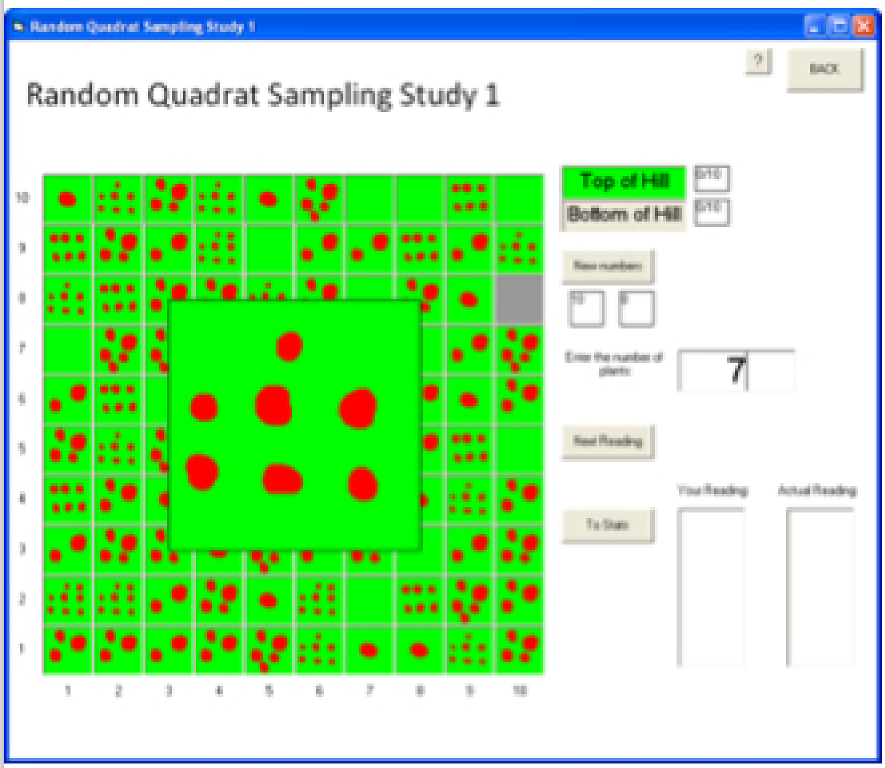

## Simulation 1 and 2:



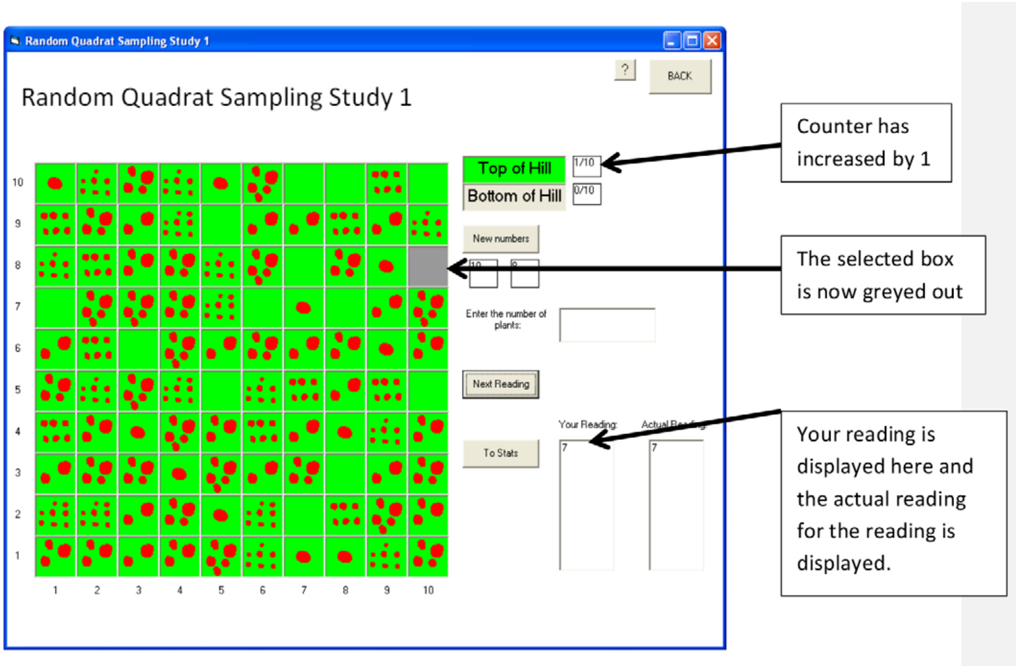

The numbers you have been given to begin with are 10 and 8. These are the coordinates for the reading you will take. If you look at the coordinate (10, 8), you can see the number of plants in the box is 7. When you click on the box, a large image box will appear. This will display the same image as the box you just selected. This is to help those who may have a visual impairment.



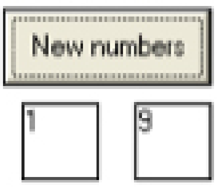
As you can see, a larger image box has appeared in the centre of the grid, displaying the same image as the one you selected.




The next step is to enter the value into the text box that you can see on the image box. You can submit the reading by clicking on the button below, labelled "Next Reading."




The next step is to generate a new set of coordinates. If you get the same coordinates, you can click on the button again to generate a new set of coordinates to use.

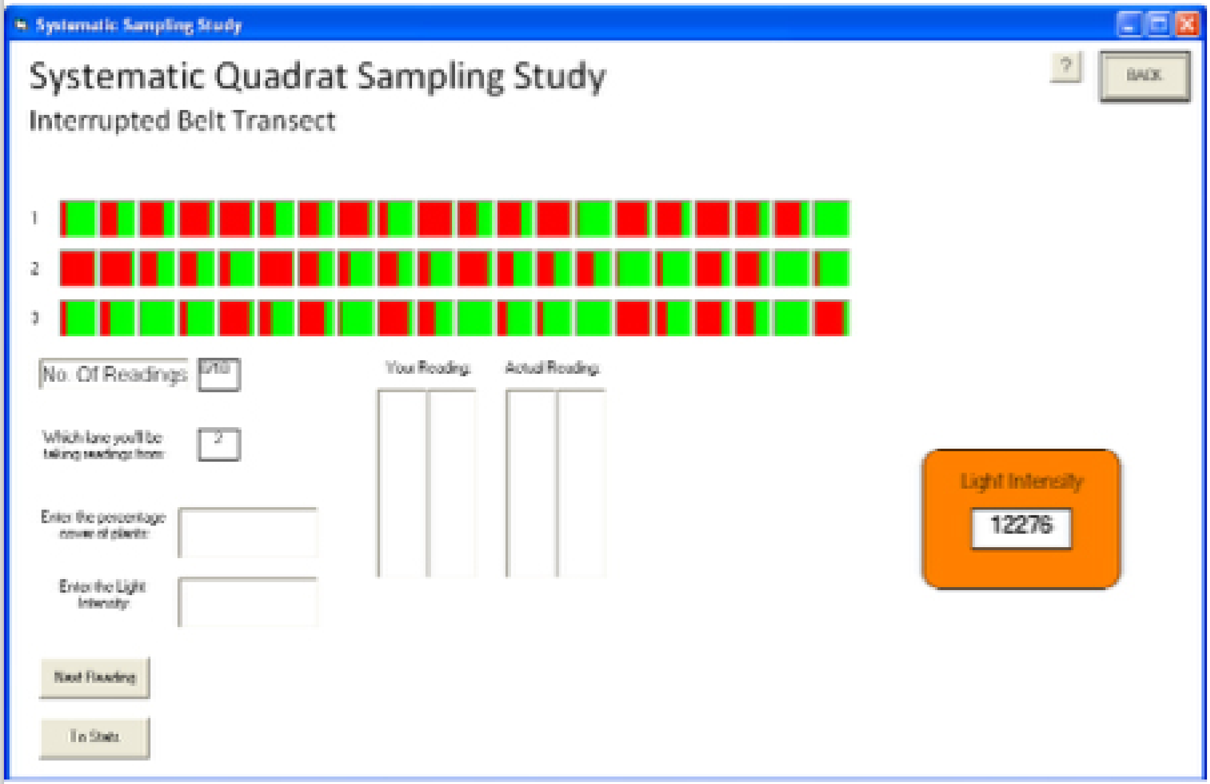



As you can see, a new pair of numbers has been generated. Once you have completed the first 10 readings, the simulation will refresh itself. This process may take a little while but it will only take a few seconds of waiting.

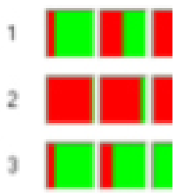
Once the simulation has been completed (10 readings for both sites), you can then proceed to the stat's form.


As mentioned above, select the "To Stats" button towards the bottom of the page. And according to which simulation you are doing, the corresponding stats page will open up.

**Simulation 3:**



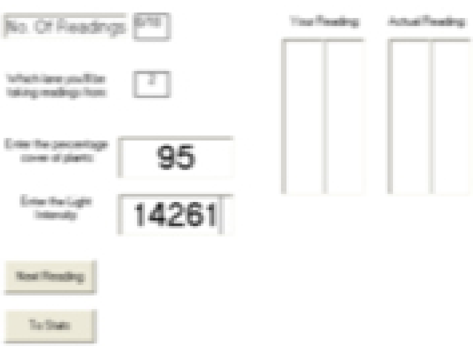

Firstly, you have been given the number 2 for the lane that you will be taking the readings from. As it may suggest, you will only be able to take readings from the second lane of boxes. As the simulation is that of an interrupted belt transect, you will take a reading from every other box. Therefore, you'll start on the second box across.



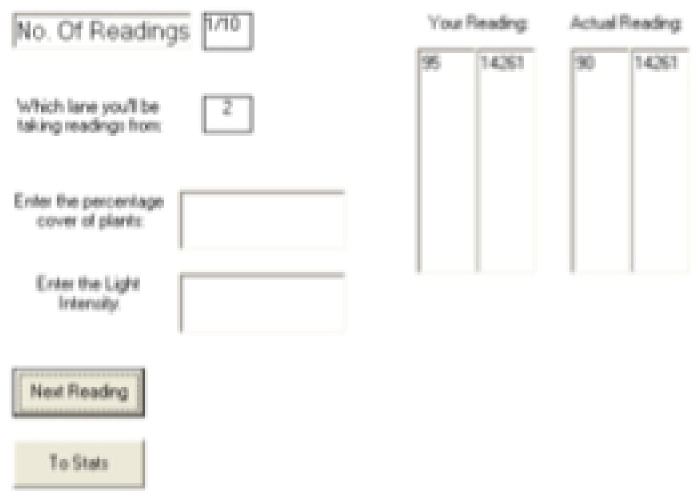
As you can see, the image second box, on lane 2, has a very high percentage of plant cover (shown by the red zone). When you select this box, a larger version will appear to the right of the transect. This box will also display the same image as the one that you selected. As I mentioned previously, this is to help those who may suffer a visual impairment.


As you can see, the large image box has displayed the same image and the image we had selected previously has been "greyed" out. This is to prevent you from selecting the same image again and potentially ruining the simulation.

You will also notice that, underneath the large image box is a smaller box labelled "Light Intensity." The value in this box constantly changes as it is, as the name would suggest, trying to simulate the varying level of light intensity. The value displayed will not change when you're trying to enter the value in to the text boxes on the left however. For obvious reasons, it would be frustrating to enter a value and have it change before, meaning you would get it wrong.

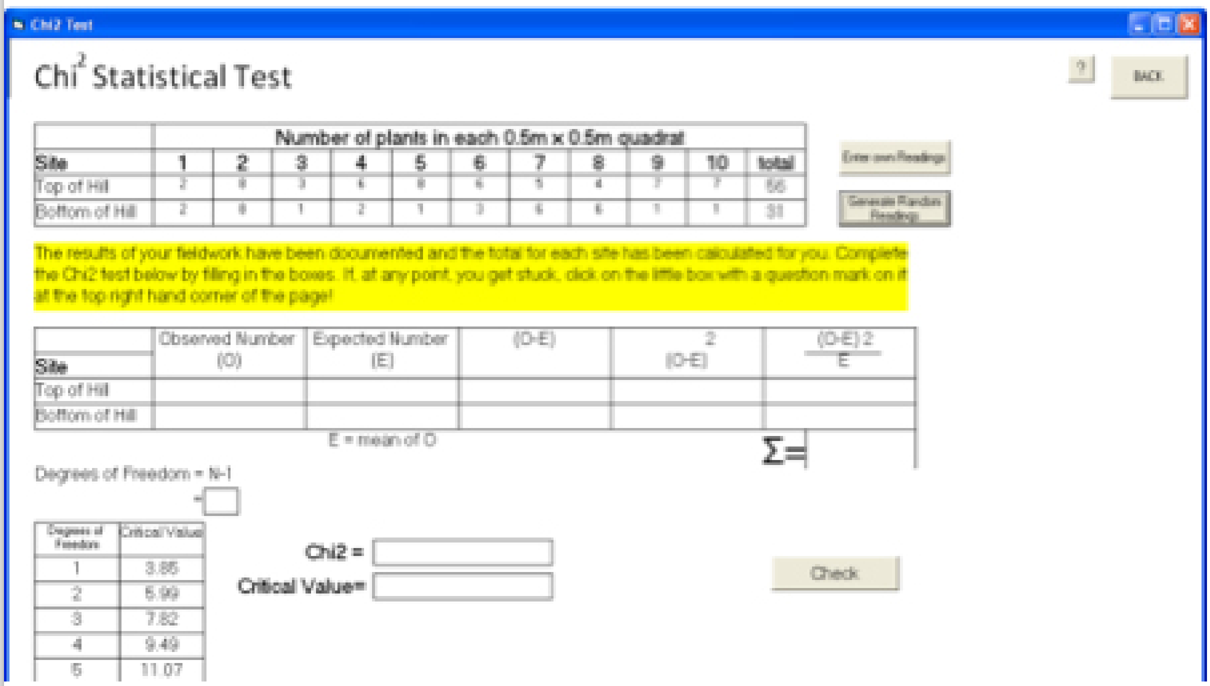



You will need to enter the values you want to record in the two text boxes (as I have shown to the right). They will be labelled appropriately so you know which text box to enter the data into. You will then need to press the button labelled "Next Reading." This will submit your readings for use later. You will also notice the 4 boxes slightly to the right of these. The 2 on the leftmost side will display the value that you will have just entered. The value that will be displayed on the rightmost side is the actual value of the text box. Whilst it may not have been entirely necessary when counting numbers on Simulation 1, percentages are trickier to calculate so this feature will allow you to see how close you were to the actual value.




As you can see, the value entered was displayed on the left and the actual values were displayed on the right. You may also have noticed the counter has increased by 1. It now reads 1/10. Once the counter reaches 10/10, the simulation will be completed and you will be able to press the button labelled "To Stat's." As the name suggests, this will take you to the stat's section.

## Chi²:



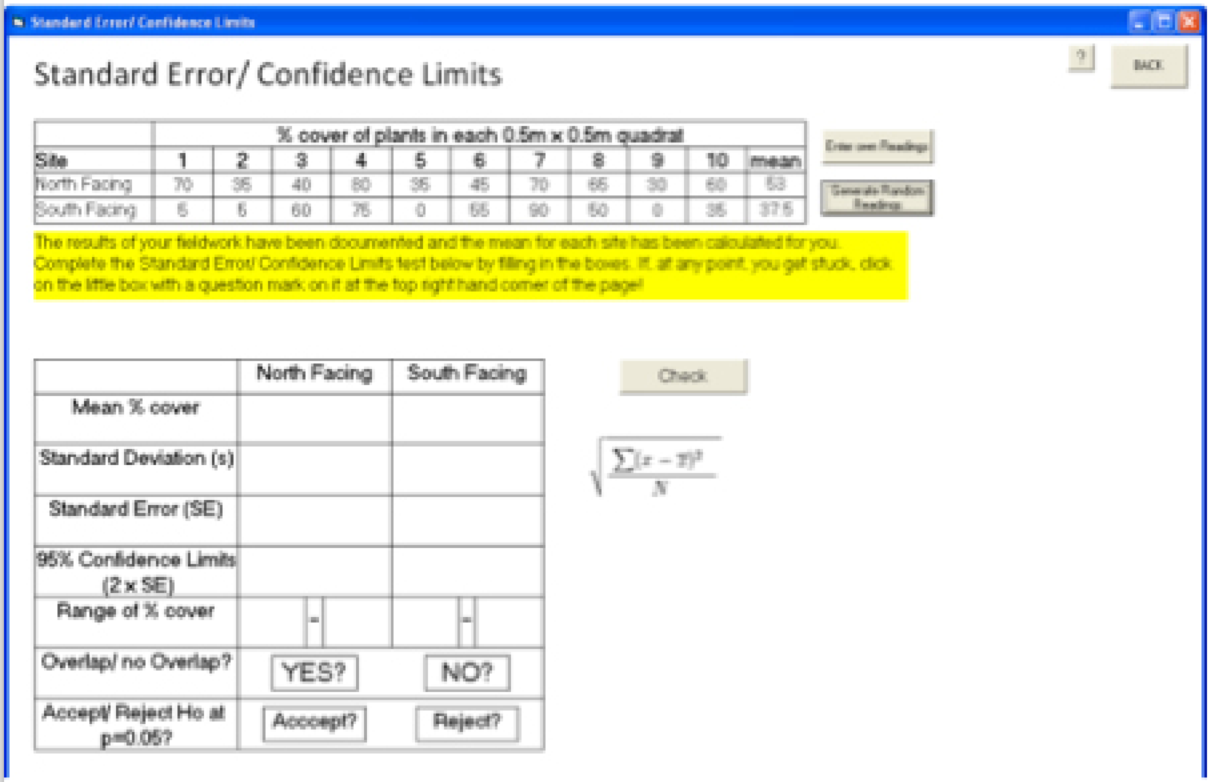

When you enter the Chi² stat's test, this is what you will be presented with. There is a table of results at the top of the page. This results table will show you your results from the simulation (if you completed the simulation beforehand). If you did not however, you will find two buttons on the side of the results table. If you want to enter your own data into the table, you want to press the top button. It will then take your readings and allow you to complete the stat's with them. However, you can generate a set of completely random results be pressing the button on the bottom. This will also fill in the table for you.

The next thing you will want to complete is the stat's table in the centre of the page. I have broken the steps down and laid it out in the same why as the hand-outs that the students will be provided with. If you get stuck with what to do, there is a help button on the top of the page. It will look like a question mark (?). This will take you through a guided example.

When you complete all the text boxes, click on the "Check" button to see how you did. If you got everything correct, all the text boxes will turn green. If you got something wrong however, the text box with the mistake will turn red. If it remains red, make sure that you have entered the value to 2.d.p.

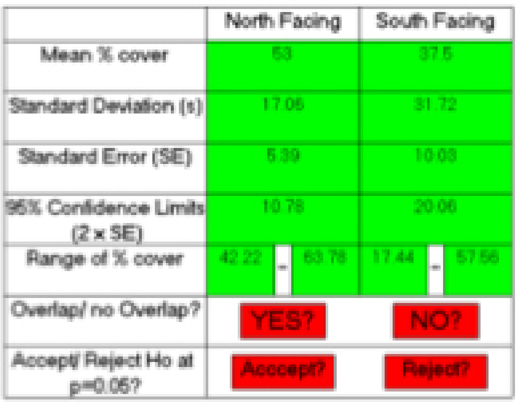
If you have got everything correct, an evaluation text box will appear at the bottom of the page. This will explain what the stat's has shown you and uses the correct vocabulary and jargon that students will be expected to use when in the exam.

### Standard Error/ 95% Confidence Limits:




This is the page for Standard Error. It's very similar to the Chi² as to avoid too much confusion. Like a said previously, the results table will be filled in for you after having completed the simulation. However, if you have not completed the simulation, you can chose to enter your own data and then submit them by pressing the top button or generate a random set of results by selecting to bottom button.

For the range, there are two text boxes in one of the spaces provided. This is to enter the two numbers in a separate text box. The lower value of the range will go on the left and the upper value will go on the right.

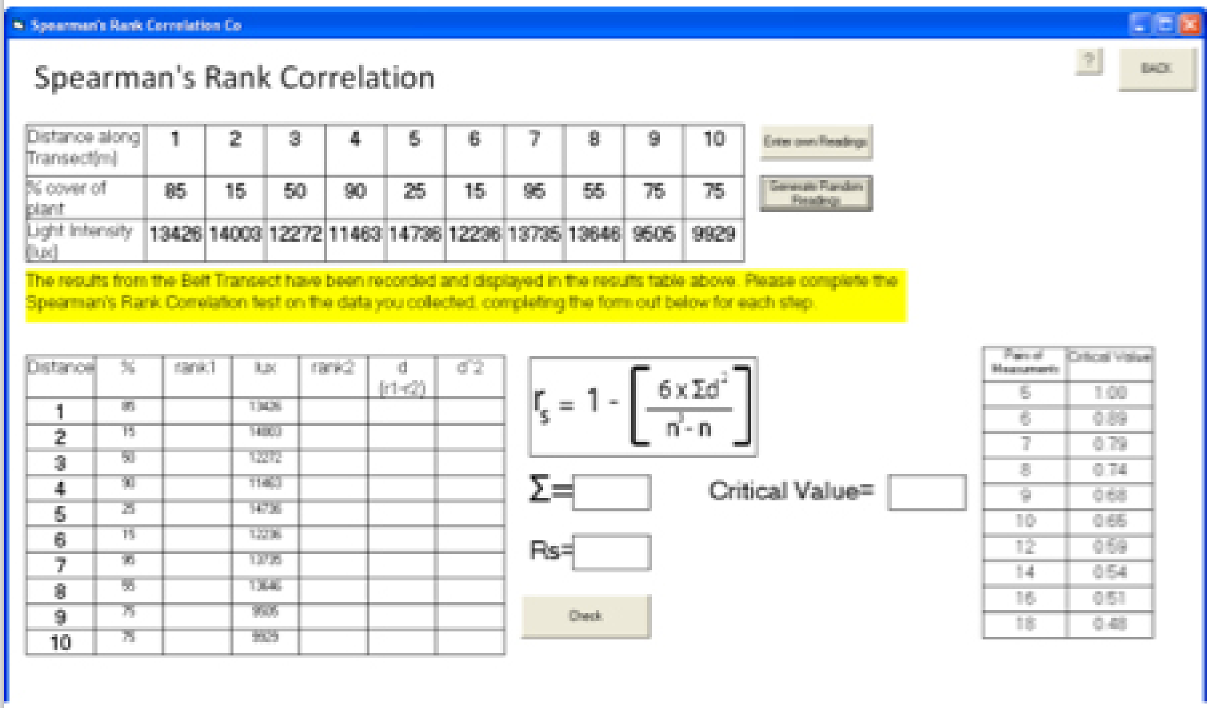


As you can see by this example, the lower end of the range is in the left text box and the upper value is in the right text box.

When everything is correct, the stat's will be complete and the evaluation box will pop up on the side of the page.

There's also a help button on the top right of the page that will open up a step by step guide of the stat's with example data.

## Spearman's Rank Correlation Coefficient:



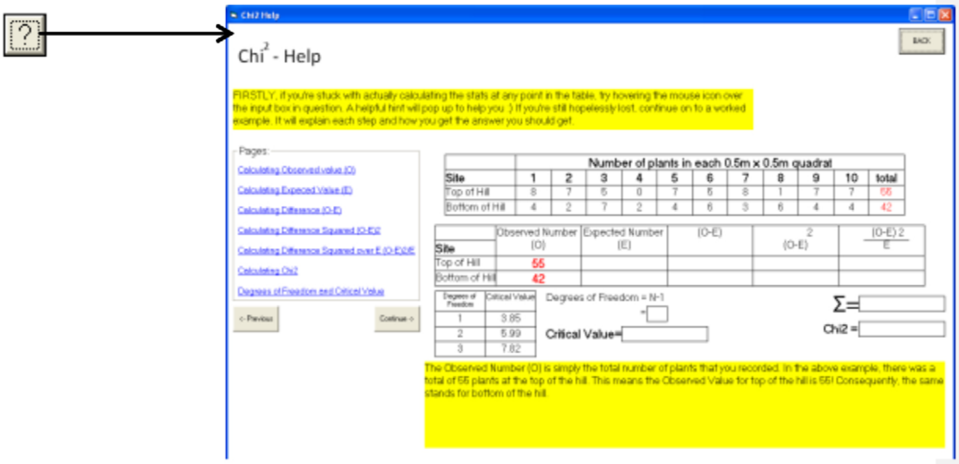

Fundamentally the same layout but there's a lot more going on in this stat's section.

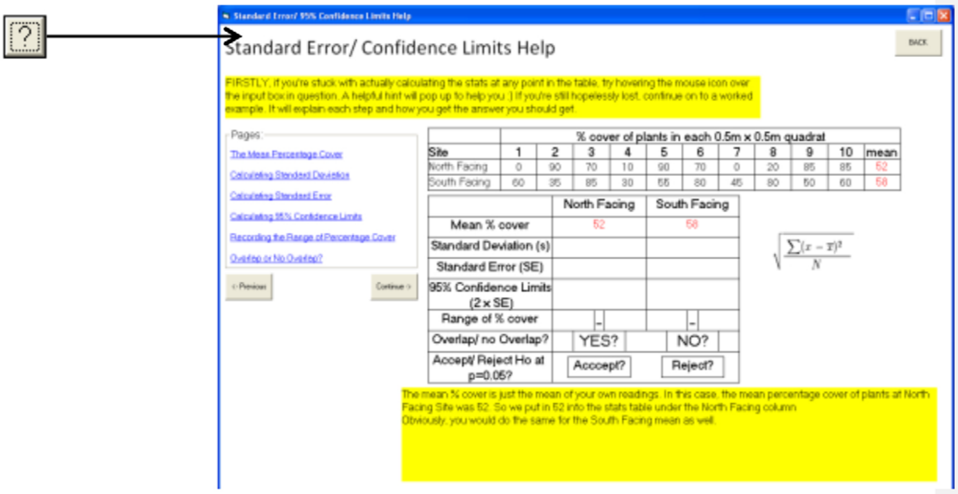
Some of the results have been entered into the stat's table to begin with because they're simply repeating the results. Firstly, you need to rank the data in ascending order. You do this by adding the rank into the box next to the reading. You will need to complete each box for all 10 readings for this stat's section and so takes a little longer. Once the stat's has been completed and everything is correct (and green), the evaluation box will pop up at the bottom of the page.

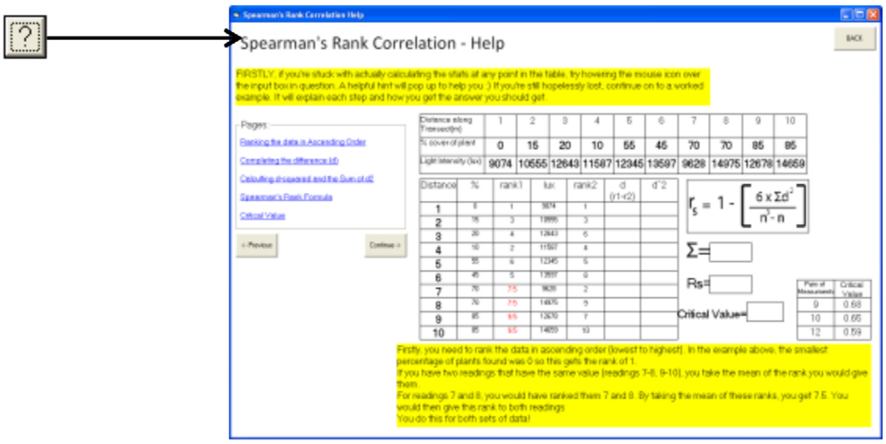
Like the other stat's sections, you can press the help button at the top of the page to go through a step by step guide of the stats, using example data

**Help Page:**

When you click on the help button, a help page will pop up.

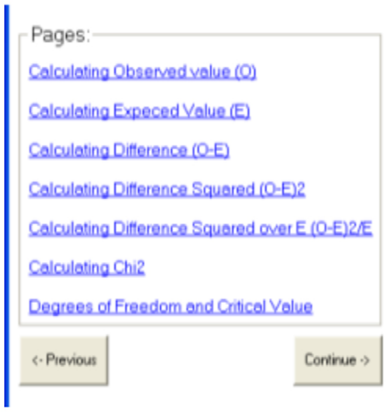


When you click on the help button the on the stats page, a help page will open up. Points of interests are highlighted in red and explained in the Yellow text box at the bottom. You can navigate the tutorial using either the "Previous" or "Continue" button. This will move you forward or back a page, allowing you to go through the guide step by step.

You can also go to a specific page by clicking on the blue text above the buttons.

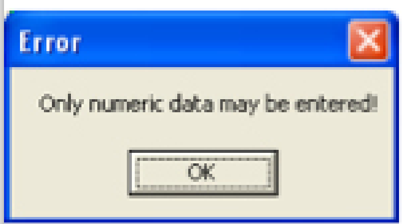



As you can see, each link is accurately labelled so you won't be confused about knowing which link to select for which section.

## Error Messages:

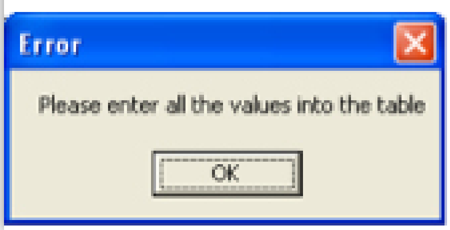
You may encounter some error messages when using the program. They are all fairly straightforward and describe to you what the error is.

During the simulation, trying to enter no data (as in you click the next reading button but haven't entered any data) or by entering data that is not numerical (contains symbols or letters for example) will bring up an error message.




This will be the message displayed to you. Make sure that what you have entered is valid for the program to enter and you won't get this error message.

Also, when entering your own data into the stats, make sure that you fill in every box in the results table before pressing the submit problem. The values all need to be numerical also. If not, this error message will appear. Simply rectify the problem and the error message will not come up.



## FAQ – Frequently Asked Questions:

1. Why is there not on screen help on all pages?

- This is because some pages are fairly self-explanatory. The stats pages all have help because they can be very confusing.

2. I'm entering the data but it still says it's wrong (stats)

- Depending on what you're trying to enter, make sure that the data is to 2.d.p (2 decimal places). Anything over will not be accepted. If a value does need to be shortened to 2.d.p, remember to round up or down and not just take the first two decimal numbers.
- If the problem still persists, there may be something wrong with the method you're using. Follow the stats guide for help if you must to see how things have been worked out and rounded.

3. I can't get the large image box to pop up for Simulation 3

- Simulation 3 is simulating an "interrupted belt transect." This means you don't start at the first box in the lane you have been given but on the one after. Interrupted basically means you are taking values at every other point. Therefore you start at the second box and not the first.

The small print

We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: www.ocr.org.uk/expression-of-interest



OCR acknowledges the use of the following content:
Square down and Square up: alexwhite/Shutterstock.com

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications:
resources.feedback@ocr.org.uk

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.



**ocr.org.uk/alevelreform**

OCR customer contact centre

**General qualifications**
Telephone 01223 553998
Facsimile 01223 552627
Email general.qualifications@ocr.org.uk



A DIVISION OF
CAMBRIDGE ASSESSMENT

ISO 9001

001