For me:
On lap top have R open so I can move from PowerPoint to R easily. (not Rcmdr yet - will do with slide 11)
Have the Word document week1lecture.doc open, and reduced, too.
Do not change directory in R till later.

I recommend you keep a copy of these lectures with you in the lab at least for the 1st few weeks as they are very handy notes on using R and Rcmdr.
These lectures will cover:

- Basic principles of programming in R and Rcmdr
- Inputting data
- Processing, plotting and writing output
- Saving data, graphs and output
- Importing files and graphs into Word documents
Organising your work

- On the Turing website you will be given a directory where you can store all of your work. Again we will show you how to access this in your prac.

- Inside it you should make a subdirectory called stat100 and subdivide it weekly so you can find your files easily.

If working on your own computer it is recommended you do the same thing. You will gather a lot of files over the years here for many units so you need to be able to access them quickly and efficiently. The skills you develop here in STAT 100 are meant to be very transferable to all of your work.
Figure 2.1 directory organisation

Note for the titles of the directories:

I have used all lowercase – easier to remember
No spaces between words and numbers
You can add to your stat100 directory weekly
You must have all files associated with a job in the same directory
Before you start

- To work in the Mathematics computer labs you do not need to load R, but you do need to know how to access it. At your 1st prac session you will be walked through this process.

- Please be on time for it and bring notepad and pens. You will build a tool kit over the weeks and once we have shown you how to do something you will be expected to be able to either remember it or look it up.

- Practice helps!
What is R?

- A statistical program available through the Internet under a General Public License i.e. it is free to use.
- Provides an environment in which you can perform statistical analyses and produce graphics.
There is a set of **core functions** that are automatically loaded when you open R and more added when you load Rcmdr.

Others in **libraries** may be needed later and you will be told then how to include them.
The functions are used to

- Read data
- Fit models
- Calculate predictions
- Tabulate and plot results
R requires you to write scripts
- Command lines to tell the computer what to do
- Requires you to be precise with spelling and punctuation
- We will use this sometimes
- Eg 2.1
  ```r
  normal.rv <- rnorm(mean=2, sd=3, n=100)
  print(summary(normal.rv))
  hist(normal.rv)
  ```

To type an arrow use < followed by -

Copy script and paste into R to run.
What do you see in the output?
Summary table of the sample of 100 numbers generated from a normal distribution which has a mean of 2 and a sd of 3. The graph is a histogram of the sample.

Most things you want R to do must be given an internal file name so it can store the information. “normal.rv” is that here. The arrow is generated by typing a greater than sign followed by a dash
Also see that for every opening bracket there must be a closing one.
The placement of commas are important, too.
The computer program and you

- You can see that the computer program has 3 broad components:
  - Read the data
  - Process as required
  - Return the results

- Your job is to:
  - Give it the data
  - Tell it what to do
  - Collect and interpret the output

This computer can only work if you are accurate in your instructions.
R and Rcmdr

- The window that has the R program running in it is called a GUI or graphical user interface.

- To load the Rcmdr package use the Packages tool on the top bar and select Load package.

- Move down the list that appears and click on Rcmdr then OK.

1. GUI is pronounced “gooee”.
Show in R.
Rcmdr

- You now will see a new GUI with a tool bar and 3 windows.
  - Script window
  - Output window
  - Messages window

- Over the semester we will use all of the tool items and manipulate information in both the script and output windows.

- The message window tells you if you have asked R to do something that it can’t do.

- No message is good!
Changing directories

- Your files for a particular job need to be in the same directory and you have to let R know about that.

- On the R GUI use the left hand tool item File, scroll down to Change dir and use the Browse button to find your directory for that week. Select it by clicking OK twice.

Do it

Show by moving around in my directory to find this week’s work – week1.
The Rcmdr script window

- We will use this in 2 ways:
  - See what R is doing after you have used a tool
  - Modify or write commands directly into it

- In the lectures I will show you how to move around Rcmdr using lines like:

  Distributions < Continuous distributions < Normal distributions < sample from normal distribution

- When a box appears asking for you to input information you will either be given it or ways of working it out. (Example 2.1 \( N(2,3^2), n=100 \))

Do it. Get Rcmdr to generate a sample from \( N(2,3) \).
Mean=2, sd=3, no. of samples (rows) = 100, no. of observations per row (columns)=1
Checking our data set

- Once you have an active data set you can see what it looks like.
- Use View data set or Edit data set

You will become more familiar with all of these commands as you work with them.

If things seem to “hang” i.e. not work, then click on the R GUI window. Sometimes files or windows are hidden.
Using a data set

- We find out about our sample
  - Note it is the active data set
  - Obtain a summary of the sample

Statistics < Summaries < numerical summaries

- Select the column name and what information you require.
- You should see the results in the output window.

Do it

Follow on from the N(2,3^2) example.
At this point we don’t need to summarise by groups – for later.
Graphing

- Now we can generate a histogram of our sample using **Graphs < Histogram**

- Again select the column that holds the data and select **densities**.

- To see the graph you will have to click on the R GUI.

Show how.
The Rcmdr output window

- You will see the commands the R has just used repeated in the script and output window.
- As well any calculations etc you have asked for appear here.
- Graphs are presented on the R GUI screen and you may have to click back on that window to see them.

I find it very handy **NOT** to have the R screen maximised so the graphs appear as produced. Note that the output lines can be very long. You do not need to worry about that. As the semester proceeds you will be able to adapt it to make small changes to aid in presentation.
Entering your own script into the script window

- Type it in then highlight it and click the **Submit** button on the bottom right hand side of the window.

```r
normal.rv <- rnorm(mean=2, sd=3, n=100)
summary(normal.rv)
hist(normal.rv)
```

Do it

Use the commands from before. However, see I have removed the **print** command from the second line. 
Rcmdr automatically does this.
Adding command lines

As we have all ready generated a density histogram of our random normal data set we can superimpose a density line on it by typing in

\[
\text{lines(density(normal.rv$obs))}
\]

- Type this line in under the \texttt{hist} line.
- Highlight the line and \textbf{Submit}.

Note the use of the \$ sign to tell R the name of the column which has the data in it. We have told R the file name (normal.rv) and the column name (obs). Also check that every opening bracket “(“ has a closing one “)” to balance it.

Do it.
Saving information - scripts

- **Check you are in the CORRECT DIRECTORY**
  - It is important that you save your work.
  - Not everything may be needed but it is better to have it than want it later and not have it.
  - It is easy to save the script and output windows.
  - Use **File < Save script as** and give it a suitable name like prac1.r **not** the default one.
  - Note that it has a special **.r** extension.
Saving information – output

- Use **File < Save output as**
- Select an appropriate name and use the ending ‘out’ on it as well as the *.txt* extension
  - Eg prac1out.txt
- This is so you can’t accidentally overwrite a data set with your output.
Saving information - graphs

- Graphs should be saved as you produce them.
- When you generate a new graph the old one is normally overwritten.
- **Right** mouse click (RMB) on the graph you have just generated.
- **Save as a metafile** if using Word for compiling records and give it a strategic name eg normalhist.emf

Do it

2. I will show you how to dodge this later or in pracs.
Finished Lecture 2 here.
Lecture 3
In the previous lecture and prac ...

- We looked at
  - setting up R and Rcmdr
  - entering command lines in R
  - Using Rcmdr drop down menus to produce
  - Graphs and numerical summaries
  - Saving graphs, script and output windows.

Lecture 3 starts here.
Today we will ...

- Revise importing text and picture files into Word
- Various ways to enter data
- Importing data files
- Manipulating variables
- Plotting in Rcmdr

1. Necessary for your assignments.
Importing to Word

- All of this information can be imported into a Word document and edited as required.
- Very useful for assignments!
- In your Word document have it set up with a header or footer with your name, job title, page number.
- Then type your heading and the question number etc.
- Then the cursor should be in the spot where you want to insert your file or graph.
- On the Word toolbar use Insert < File and browse to find the one you want. Click insert.

Show how using Week1out.doc

You can delete any information that is not specifically required for that question or exercise. Do not put extraneous material in a report!
Importing a graph

Use Insert < Picture < from file

- Browse to find the one you need.
- Once it is embedded in your document it can be resized by clicking on it.
- A black frame should appear with little black boxes on the corners and centres of sides.
- Hold down the left mouse button on the bottom right box and drag upwards to the left. You will see it shrink.
- Adjust it so you can see the labels etc.

Do it to import the histogram normalhist.emf
Every graph and table needs a heading
- Figure 1 Histogram of sample from $N(2,3^2)$ or
  Table 1 Monthly CO$_2$ concentrations for Muana Loa etc.

Don't forget that nearly every question in an assignment requires some input from you by answering questions or making comments. You can type those in at any time and in any appropriate place inside your word document.
Summary

1. Set up a new directory for the week/assignment
2. Download any files required from STAT100 web page
3. Open R
4. Load Rcmdr
5. Change directory in R
6. Import or create data set = name.txt
7. Work as required
8. Save graphs as you go = metafiles = namegph.emf
9. Save script file = name.r
10. Save output file = nameout.txt
11. Close R and Rcmdr
12. Logout of session

6. An appropriate name with no spaces. Can use dash instead.

8. Remember the metafile is a convenient format if working in Word.

12. Logging out is necessary if you are working on the Maths computers so no one else can access your directory.
1. There are several ways of entering data into Rcmdr for it to work on.
Entering data

- We have seen one way to enter data into Rcmdr and that is have the program generate it. (As in the example normal.rv in your prac.)
- We can also
  - Enter data directly into a file inside Rcmdr.
  - Enter data into a file outside Rcmdr using TINN-R, Notepad or Excel and then import it into Rcmdr.
  - Download data sets from the web - e.g. from the STAT100 page and then import them.

Note you can’t use Word to set up a data file.
Using data files inside Rcmdr

- The same principles apply for setting up data files in all places.

- **Columns** contain variables eg height, weight, gender.

- **Rows** contain the measurements from *each* unit.

- The 1st row contains the column headings used to describe the data.

If we have 2 or more measurements on the same individual item (person, sheep, tree, machine etc) then they go across a row under distinct descriptive column headings.
Entering data

To enter data from the table given

- Use **Data < New data set**
- Give the data set an informative name
- Click on the var1 etc in the top row to enter the column names. Easier to use all lower case and no spaces.
- Enter data by moving down a column.
- Make sure you click outside the last entry.
- When all is entered click the X on the top right to close the file.
Example 2.2 BMI index

Save this data in a file called BMIdata from inside Rcmdr

Table 2.1 BMIdata.txt

<table>
<thead>
<tr>
<th>Height</th>
<th>1.75</th>
<th>1.8</th>
<th>1.65</th>
<th>1.9</th>
<th>1.74</th>
<th>1.91</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>60</td>
<td>72</td>
<td>57</td>
<td>90</td>
<td>95</td>
<td>72</td>
</tr>
</tbody>
</table>

Use **Data < new data set**

Give it a suitable name, bmi

Var1 = height and it is numeric

Var2 = weight and it is numeric

Do it

This is the data set you worked on in the 1st prac session so I’m going to skip some of the hands on with this data set. I’ve left the working in for you for revision.

2. Set up the data set using Rcmdr New Data set facility.

Note that even if data is given to you in a table with numbers entered across a row, when entering it into a table for R use you should enter down the column. The row format is used in print for space saving.
Save your data set

- It is very important that you save your data set immediately you have set it up. If the system crashes you don’t want to have to type it all in again!

- Make sure you are in your own working directory.

- Use **Data < Active data set <Export active data set**
  
  - Select **Write variable names**
  - **Deselect** write row names and quotes around character values
  - Keep missing values as **NA** and separator as **Spaces**

Do it

Some mightn’t have done this so do it. Import the data set
If you have changed your directory to your STAT100/week1 (etc) then this is the default directory that will appear in the Save box.

Save BMIdata
Importing from Excel or other saved data sets

- Use **Data < Import data set**
  - Select the type of file and browse for your file. It should be in the folder you are working from.
  - If using an EXCEL file you will be asked to select the sheet, too.
  - Check the data set box has the named data set you have just imported.
  - If you make any changes to it inside Rcmdr you will need to export it (save it) and then import it again.

Show how by importing **bmi.xls**
Alternative direct data entry

There are times when you may use files that we have directly entered data into so R can read it.

- Type in R or a file that will be sourced (used) by R

```r
Height <- c(1.75,1.8,1.65,1.9,1.74,1.91)
weight <- c(60,72,57,90,95,72)
bmi <- weight/Height^2
BMIdata <- data.frame(weight,Height,bmi)
cat("the mean bmi is\n")
print(mean(BMIdata$bmi))
```

- I have this saved in a TINN-R file called bmi.r
- To source it in R or Rcmdr type `source("bmi.r")`
- Then press enter (R) or highlight and Submit (Rcmdr)

2. Note the use of commas between each number and the open and close brackets.

The bmi line reads as (make the file) bmi equal to weight / square of height.

R takes the 1\textsuperscript{st} weight number and divides it by the 1\textsuperscript{st} height squared and enters it in the 1\textsuperscript{st} place of the new internal file bmi. Then it moves on to the 2\textsuperscript{nd} set of numbers etc.

It is very important then that if you are entering data for individuals they are in the same order!

2.1 I used TINN-R to write the program. We will look at this sometime over the next few weeks in pracs.

2.2 To “source in R” means to import the command file into R and get R to run it. This file must be in the same directory you are working from. Make sure you use double quote marks around the file name.
Calculating new variables

- In the previous example we asked R to calculate the body mass index by dividing weight by height$^2$.
- We can use our saved data table and have Rcmdr do it, too.
- Use

  Data < Manage variables in active data set < compute new variable

You did this in the practical. I’ll do it quickly again for you.
In the box give the new variable a name - bmi.

To compute the expression:
- double LMB click on weight
- Type in the operator /
- Double LMB click on height
- Type ^2
- Click OK

^ is called a ‘carrot’ and means ‘raise (previous thing) to the power of’
It is a good idea to do the next 3 steps
- **View** the data set
- **Save** the data set
- and then **import** it back into Rcmdr to work on it.

Calculate summary statistics of bmi

- **Statistics < Summaries < numerical statistics**
- Select bmi

Do it.
1. Factors

- Sometimes we want to investigate data from different groups
  - like men versus women,
  - Low, Medium or High concentrations of a chemical etc
  - These are called FACTORS

- To do this we enter the factors in a column with our data often as alphabetical labels but sometimes we use numbers 1, 2, 3 etc if no confusion can be made.

We have all ready seen how we can generate new variables from within a data set. Now we will see what else we can do.
This will allow us to produce plots to compare the groups or calculate a variety of statistics to compare them.
Example 2.3 City plants data in sludge.txt

- Column **zinc** is a numeric vector and we can do numerical calculations on it like find the mean.
- Variable **City** is a categorical variable or a factor.
- Column **Block** codes 1 and 2 represent when the samples were taken, January and March. Much easier to type! Currently this is a numeric vector.
- **Rate** could be either numeric or categorical depending on what we want to do.

Categorical = divides data into different categories
Look at the data set from inside Rcmdr
Converting to Factors

- We want to be able to compare the zinc concentrations between the different months so we need to tell R that Block is a factor – a way of dividing the data.
- Import the data and view
  
  Data < import data < from text file
  View data set then close that box

- Convert Block to categorical factor
  
  Data < Manage variables in active data set <
  Convert numeric variables to factors

- do it

2. From sludge.txt – all ready done
3. Do it.
Click on **Block** and for Factor levels select **supply level names**
- Leave **new variable name** as is.
- Click **OK**

- In the new box, type in the names that correspond to the old numbers.
- Click **OK**
- In the script window you can see the line that says what R has done.

Do the conversion.
Do the boxplots.
Plotting in Rcmdr

- Using our previous example, use boxplots to compare the zinc levels in plants in the 3 cities.

**Graphs < Boxplots**

Select Zinc then Plot by groups and select City
OK, OK
Figure 2.2 Boxplot of zinc concentrations by city

Note that when you try this your boxplots will come out clear (white). Wait for a later lecture for ways to modify these.
More plots

- There is quite a range of plots available to describe data depending on the type of data you have.
- Most frequently used ones will be
  - Histogram
  - Boxplot
  - Scatterplot
  - Quantile comparison plot
  - Plot of means

2. We have all ready looked at histograms and boxplots.
   The quantile and plot of means we will meet later.
Scatterplots

- Requires 2 or more pieces of information on the same individual unit

- Using the sludge data graph a scatterplot of zinc against Rate by City.

Graph < Scatterplot

- Select Rate as the X variable
- Select zinc as the Y variable
- Plot by groups => City; OK
- OK
- You may need to left click in a free area of the graph to insert the key depending on the version of R you have.

Lynette McLean

50
Figure 2.3 Scatterplot of zinc against Rate for each City
Scatterplots

This type of plot makes it simpler to suggest how these data could be represented as a statistical model.
Example 2.4 CO2uptake

Using the file CO2uptake.txt from the STA100 web site plot Uptake against Concentration to suggest a model for CO₂ uptake by plants.

1. Import the data
2. Draw a scatterplot
3. Is a pattern visible?

The next 2 examples give you more practice in graphing and manipulating data.

Download the file from the Data sets used in lectures from the STAT100 web page.

Import the file into Rcmdr and draw a scatterplot of the data.
- **Data** < import data < from text file
- **Graph** < scatterplot
- Deselect boxplots and smooth lines

Now change the plotting symbol using `pch=3` inside the scatterplot command line, highlight the entire line and **Submit**.

Do it
Figure 2.4 \( \text{CO}_2 \) uptake by concentration with fitted linear regression line
The systematic effect of CO$_2$ concentration is modelled by the straight line and the random effect will be of a smaller magnitude than systematic differences and accounts for the variation around that line.
Example 2.5 Arsenic concentrations in ground water

- Import arsenic.txt
- Make location a factor
- Draw boxplots of arsenic concentration by location.

do it

1. File is in Data sets used in Lectures – Lecture 3 from STAT100 web page.
Figure 2.4 Arsenic concentration by location
Boxplots

- Good for visual comparison of factored data (i.e. that divided into groups)
- Boxes denote the middle 50% of data and the line inside the box indicates the middle point or median.
- The whiskers denote the 95% range about the median.

More on these when we talk about measures of location later in the unit.