Introduction to Statistical Analysis using R commander

Sarah Vowler and Mark Dunning [*]

Last Document revision: June 8, 2015

Contents

[*] Acknowledgements: Sarah Dawson, Elizabeth Merrall and Deepak Parashar

1 Introduction

"To consult the statistician after an experiment is finished is often merely to ask him to conduct a post mortem examination. He can perhaps say what the experiment died of.". R.A. Fisher, 1938

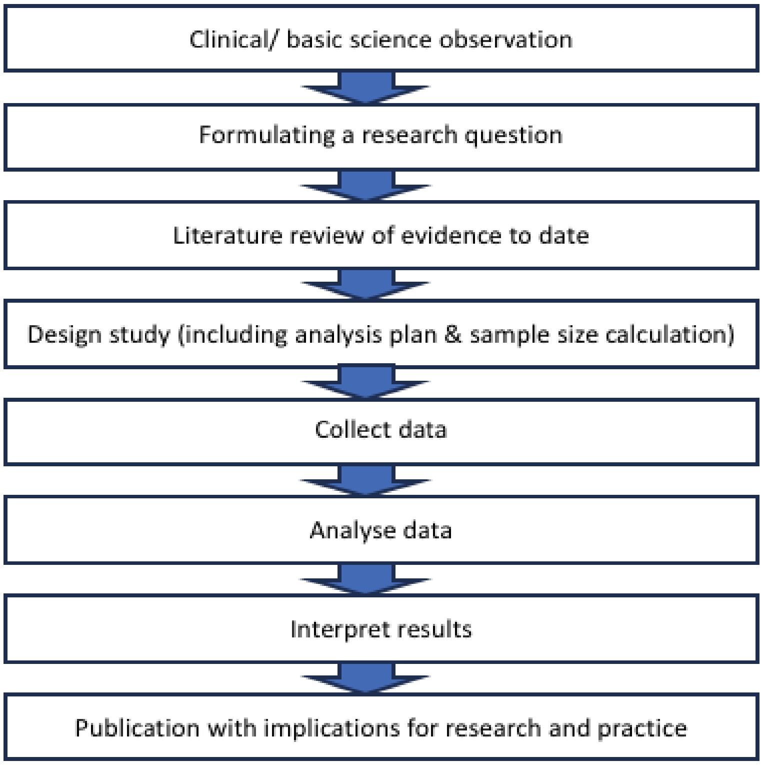
The goals of statistical methods could be summarised as follows:

- drawing conclusions about a population by analysing data on just a sample;
- evaluating the uncertainty in these conclusions; and,
- designing the sampling approach so that valid and accurate conclusions can be made from the data collected.

R Commander is an add-on to R, designed to make statistical analysis in R more accessible to those unfamiliar with R programming. Some understanding of basic statistical tests is required before being able to use R Commander effectively, as it is easy to make mistakes if you don't know precisely what you're doing.

The Bioinformatics Core at Cambridge Research Institute employs statisticians to help researchers at the institute with the statistical aspects of their studies. Whilst we are always happy to do an analysis for you, we are also happy to support you if you choose to run the analysis yourself. Please get in touch if you have any questions or require our support, or come along to our Wednesday afternoon Statistics Clinic.

E-mail address: `cristatsclinic@cruk.cam.ac.uk`


Clinical/ basic science observation
Formulating a research question
Literature review of evidence to date
Design study (including analysis plan & sample size calculation)
Collect data
Analyse data
Interpret results
Publication with implications for research and practice

2 Thinking about your analysis

Statistical tests are used to help us draw conclusions about one or more populations. As populations can be very large, we usually take a sample to collect our data on and perform statistical analyses on this sample data. It is never too early in the research process to start thinking about the statistical methods you will use. At the design stage of a study, it is invaluable to think about the data that your experiment will generate, how that data might be analysed and what size of effects may be detected.

We want to generalise our findings from a sample of observations. Most statistical methods rely on the assumption that each observation is sampled independently of the others. In this context, independence means that one observation has no influence over other observations in the sample, or that one observation gives us no information about other observations within the sample. For example:

- **Suppose you are interested in a measurement taken from each of 20 individuals**. If there is no reason why the measurement for subject 1 should be more related to another measurement in the set of 20 measurements than any other, e.g. no siblings amongst the 20 individuals, only a single measurement from each individual, then in this situation we can say that the 20 measurements are independent of each other.
- **Suppose you are repeating an experiment six times, each time in triplicate**. The 18 measurements are not independent as observations within an experiment may be more similar than observations from separate experiments. The experimental conditions (e.g. temperatures, reagent preparations, extraction date, etc.) may differ between experiments. We could get six independent mean values by calculating the mean of each triplicate.
- **Suppose you are measuring blood pressure before and after a treatment for 30 patients**.

You will not have 60 independent measurements as we have two sets of measurements per patient. Measurements within a patient may be more similar than between patients. In addition, the treatment may affect the blood pressure measurements taken afterwards. Therefore, measurements before and after treatment are not necessarily comparable. However, for each patient, we can calculate the difference between the measurements before and after treatment. The 30 differences are independent and we might test whether these differences are significant using a t test (more later!).

- **Suppose you are measuring protein expression in a cell sample which may be one of five cell-types and collected from one of three mice**. As you might be able to tell by now, these 15 samples would not be independent. The protein expression may depend on the cell-type and which of the three mice the sample was collected from. This example is tricky(!) and how you might handle these data will depend on your research question. Advance planning will certainly help.

The type of data you will get will determine which analyses will be most suitable. Data take two main forms **categorical** or **numerical**.

Categorical observations are allocations of individuals to one of two or more classes or categories. These categories may be **nominal** or **ordinal** (some natural ordering of the categories).

Examples of **nominal** data are: Sex - Male/female; Disease status diseased/non-diseased; Treatment status treated/non-treated.

Examples of **ordinal** data are: Smoking non-smoker/ex- smoker/light smoker/heavy smoker; Stage of breast cancer 0/1/2/3/4; socioeconomic status low/middle/high.

Numerical observations may be **discrete** or **continuous**. Discrete numerical variables are often counts of events whereas continuous numerical variables are mostly measurements of some form.

Examples of **discrete** data are: Number of children; Sequence tag counts (from ChIP-seq); Number of relapses.

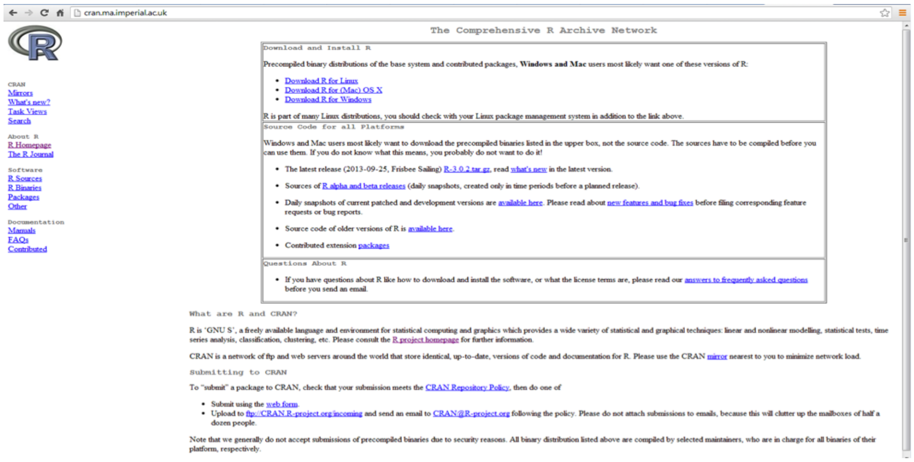
Examples of **continuous** data are: Body Mass Index (BMI); Temperature; Blood pressure; Tumour width.

3 R and R Commander basics

To install R visit www.r-project.org. In the 'Getting Started' box half-way down the page follow the 'download R' link. Scroll down to the UK and select any one of the three links. On the next page choose the appropriate operating system for your computer from the three 'Download R for...' options.

3.1 Installation of R

After clicking on the 'Download R for Windows' link, select 'install R for the first time' on the following page. The version of R used to write this manual is 3.2.0, the version number you download may be



The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2013-09-25, Frisbee Sailing) R-3.0.2.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the R project homepage for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN mirror nearest to you to minimize network load.

Submitting to CRAN

To "submit" a package to CRAN, check that your submission meets the CRAN Repository Policy, then do one of

- Submit using the web form.
- Upload to ftp://CRAN.R-project.org/incoming and send an email to CRAN@R-project.org following the policy. Please do not attach submissions to emails, because this will clutter up the mailboxes of half a dozen people.

Note that we generally do not accept submissions of precompiled binaries due to security reasons. All binary distribution listed above are compiled by selected maintainers, who are in charge for all binaries of their platform, respectively.

Figure 1: R can be downloaded from the R-project web page

different as new versions are released every six months. Following this link will start the installation of R. If you get a security warning select 'Run'. Follow the directions in the install wizard to install R. We haven chosen to run R through the RStudio interface, which you will also need to install.

3.2 Installation of RStudio

To install RStudio visit http://www.rstudio.com/products/RStudio/ and follow the links to download RStudio Desktop for your operating system.

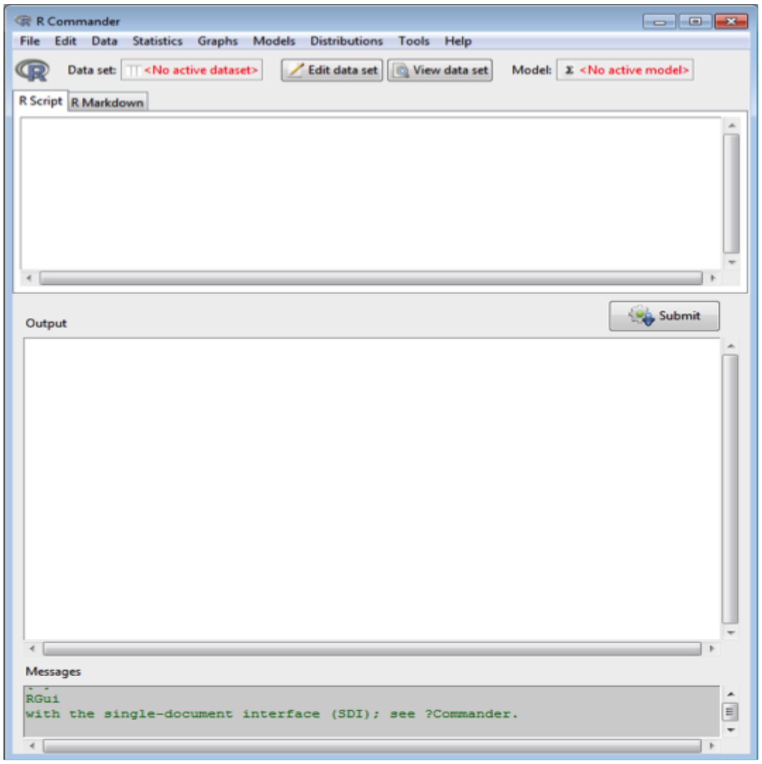
3.3 Installing R commander

The first step is to load RStudio. There should be an icon for this on your Desktop, or in the Start menu.

Once loaded, type the following in the 'Console' in the bottom-left where the 'blinking' cursor is;

```r
install.packages("Rcmdr")
```

and press Return. R Commander should now be downloaded and installed. You only need to install R Commander once. After installing R Commander you need to load the package in order for the R Commander window to open, and you need to do this every time you wish to use the package. In the RStudio console type



```
RGui
with the single-document interface (SDI); see ?Commander.
```

Figure 2: The R commander screen upon startup

```r
library(Rcmdr)
```

and press Return. Once the R Commander package has loaded a new window should appear - this is R Commander. *Note that the first time you try and run R commander it will ask if you wish to install some additional pacakges. We recommend that you do this.*

3.4 Navigating R commander

R Commander has a tool bar across the top, a row of buttons and three panels below.

The tool bar can be used to import data, perform statistical analyses and plot graphs, amongst other things. This is also where you will open and/or save your work.

The row of buttons allow you to visualise and edit your data, as well as switch between different datasets.

The first panel contains your R script. You can either type a script directly into this box yourself, or you can use the tool bar across the top and based on your selections a script will be automatically generated.

The second panel shows the R script along with the output that script generates when run.

The third panel shows any messages that appear as a result of the R script being run. Error messages will appear in red, warning messages in green, and other information in blue. **Note that running R**

commander through RStudio will put all R output into the RStudio console rather than this panel.

3.5 Importing data

There are two main ways to get your data into R Commander:

- 1. **Input by hand**: This method is fine for small datasets but would be time consuming with larger datasets and we must be careful not to make any mistakes whilst typing out the data by hand. To input your own dataset by hand into R Commander, click on 'Data' on the R Commander tool bar, 'New data set...'. You will be prompted to enter a name for your dataset. Enter a name and click 'OK'. The data editor will appear. Click on a cell to enter data. Once you've finished editing the data, close the data editor to return to R Commander.
- 2. **Import from a .txt file**: This method is much easier than inputting the data by hand if your dataset is large. You need to make sure the data is displayed in the correct way in your file before importing it into R Commander. To import a data file into R Commander, click on 'Data' on the R Commander tool bar, 'Import data' and the select either 'from text file, clipboard or URL...'. You will be prompted to enter a name for the dataset you're importing. Enter a name for your dataset and press 'OK'. Find the file location, click on the file you wish to import and click 'Open'.

You may notice that you can also import data from Excel files, however this is often problematic when you save your work to come back to it at a later date, so this method of importing data is best avoided. An Excel file can easily be converted into a .txt file, and vice versa. Please ask a member of the Bioinformatics Core if you're unsure of how to do this.

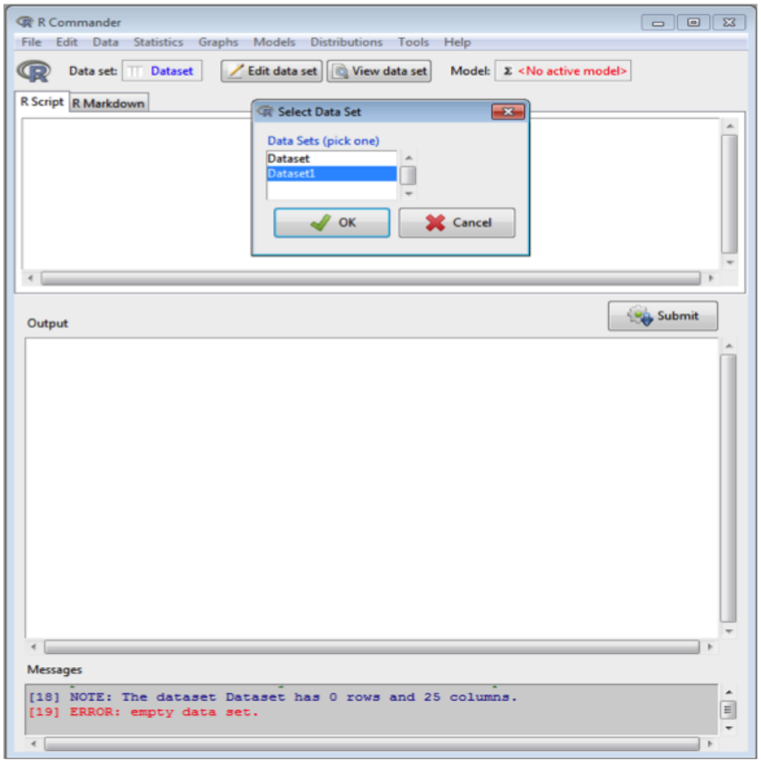
3.6 Creating projects with multiple datasets

R can handle multiple datasets within one project. When you import your data ensure that each dataset has a different name. Note that giving two datasets the same name will cause the one currently being held in R's memory with that name to be overwritten with the new dataset of the same name.

To change between your datasets click on the 'Data set' button within R Commander. This will bring up a list of your datasets allowing you to choose which one you want to work with (Figure 3).

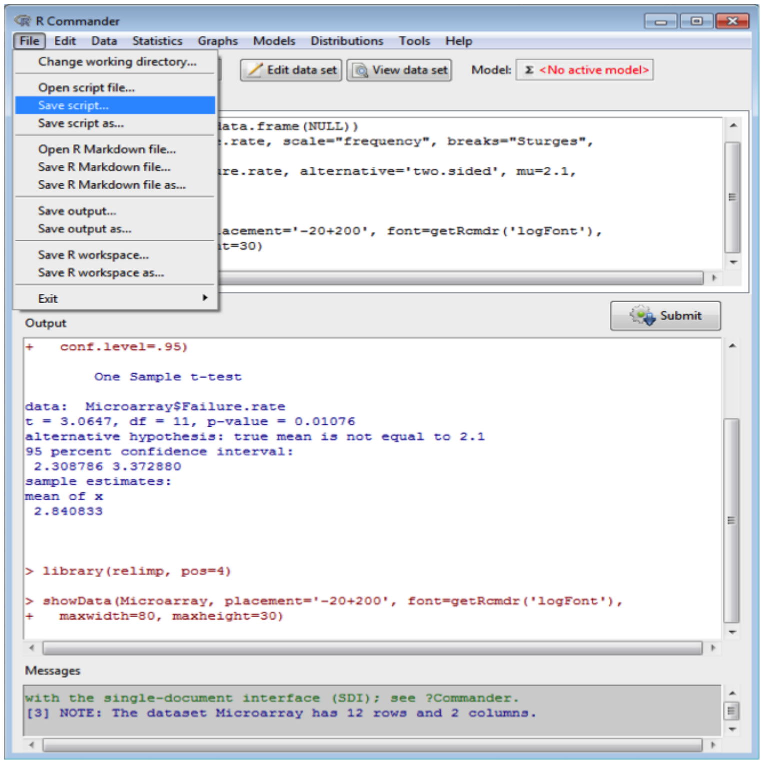
3.7 Saving your work

R Commander has a key advantage over most traditional 'point-and-click' statistical software. When using point-and-click statistical software, although you can save your work it's not usually possible to keep a record of every single step taken. However, in R Commander an R script is generated, creating a record of every step you've taken in your analysis. This is of increasing importance in an era where research is becoming more and more data-driven, and the need for reproducibility at every step of an experiment, including the statistical analysis, has been recognised.



```
[18] NOTE: The dataset Dataset has 0 rows and 25 columns.
[19] ERROR: empty data set.
```

Figure 3: Selecting a dataset to work with in R commander



```
ata.frame(NULL))
.rate, scale="frequency", breaks="Sturges",

re.rate, alternative='two.sided', mu=2.1,



acement='-20+200', font=getRcmdr('logFont'),
t=30)
```

```
+   conf.level=.95)

        One Sample t-test

data:  Microarray$Failure.rate
t = 3.0647, df = 11, p-value = 0.01076
alternative hypothesis: true mean is not equal to 2.1
95 percent confidence interval:
 2.308786 3.372880
sample estimates:
mean of x
 2.840833


> library(relimp, pos=4)

> showData(Microarray, placement='-20+200', font=getRcmdr('logFont'),
+   maxwidth=80, maxheight=30)
```

```
with the single-document interface (SDI); see ?Commander.
[3] NOTE: The dataset Microarray has 12 rows and 2 columns.
```

Figure 4: Saving your R Commander script

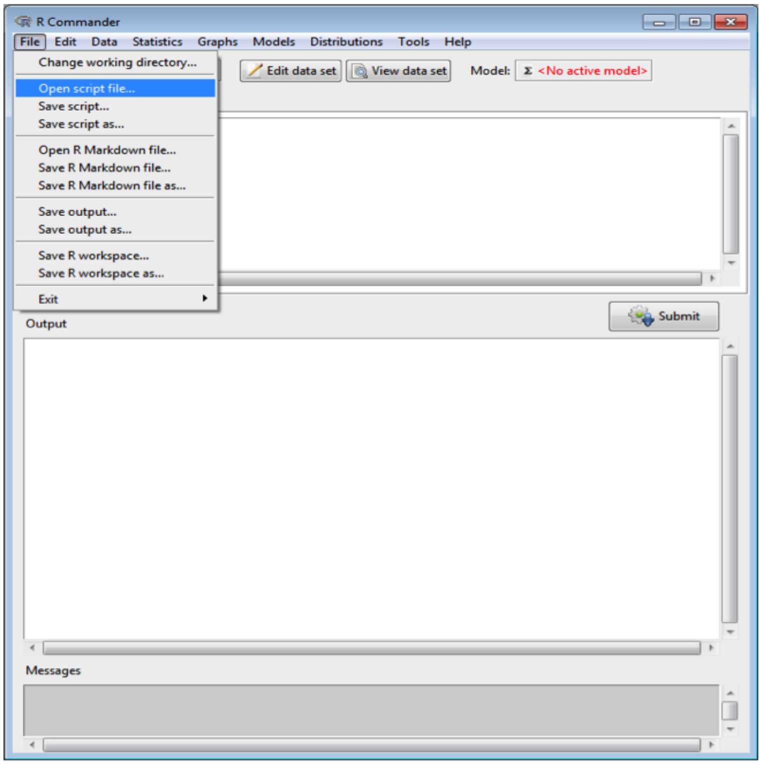


Figure 5: Opening your R Commander script

When saving your work in R Commander, you will usually be saving the R script. By saving the R script you can come back at a later date, run the script again and produce the same set of results. It also records details of all the parameters used in your analysis, so if you later need this information you can refer back to your R script to find it. The only thing the R script does not record fully is when data is typed into the data editor by hand.

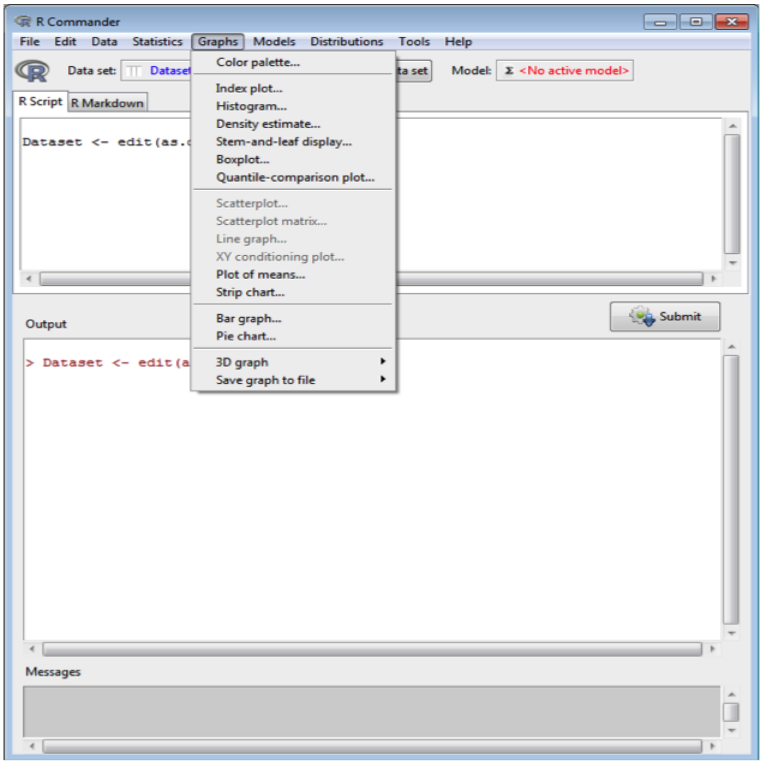
Use: `File` → `Save script...` → Give your file a name (will have file extension ".R") → `Save`

3.8 Opening previous R Commander projects

You may have already done an analysis in R Commander and wish to open it to view the analysis or to continue working on it.

Use: `File` → `Open script file...` → Find your file a name (will have file extension ".R") → `Open`.

Once you've opened your previous R script, press the Submit button below the R Script panel to run the script.



```
Dataset <- edit(as.d
```

```
> Dataset <- edit(a
```

Figure 6: Selecting a graph to draw

3.9 Graphs

R is well known for creating plots of publication quality. Whilst R Commander will help you to create basic plots, most plot types can be customised to more exact specifications using R script. This is beyond the scope of this course, but further training is planned in this area. Some suggestions on how to modify R commander graphs can be found in Natasha Karps' guide; http://cran.r-project.org/doc/contrib/Karp-Rcommander-intro.pdf

R Commander has several graphical options and it will not be possible to discuss them all here. Some of the most useful options will be discussed below. Whichever graph type you choose to use, take a little bit of time to consider whether this is a clear and meaningful representation of your data.

- Scatter plot
- Histogram
- Box plot
- Bar chart

Once a graph has been drawn you cannot edit it from the graph window. If you wish to make changes to a graph (e.g. changing the title or axis labels) you will need to re-create the graph using the point and click menus, making sure to choose the option(s) relevant to the change(s) you want to make.

4 Statistical Tests

The statistical approach used is dependent on the data type. In this document we will describe **t-tests** (including one-sample, independent two-sample and paired two-sample), which can be used when we have one or two groups of **continuous numerical** data, and **contingency tables** (chi-squared test and Fisher's exact test), which can be used when we have two **categorical** variables (which may be ordinal).

4.1 Exploratory analysis

Before conducting the formal analysis of our data, it is always a good idea to run some exploratory checks of the data:

- a) To check that the data has been read in or entered correctly;
- b) To identify any outlying values and if there is reason to question their validity, exclude them or investigate them further;
- c) To see the distribution of the observations and whether the planned analyses are appropriate.

4.2 Summary statistics

Summary statistics give you an initial impression of the data the most common measures of the location and spread of the data are, respectively, the **mean** and **standard deviation**.

The **mean** is the sum of all observations divided by the number of observations.

E.g. No. of facebook friends for 7 colleagues

$311, 345, 270, 310, 243, 5300, 11$

The mean is given by:

$$\bar{X} = \frac{311 + 345 + 270 + 310 + 243 + 5300 + 11}{7} = 970 \qquad (1)$$

The **standard deviation** is the square root of the variance of our observations. The variance is the sum of squared differences between each observation and the mean divided by the number of observations.

$$s.d = \sqrt{\frac{(311 - 970)^2 + (345 - 970)^2 + \ldots (11 - 970)^2}{7}} = 1913 \qquad (2)$$

When the data are skewed by several extreme, but valid, observations (see example histogram below), the **median** and **interquartile range** may be more meaningful summary measures than the mean and standard deviation. With the extreme observation 5300 in our data-set, the median and interquartile range could provide a more suitable summary of our data.

The **median** is the middle observation when the data is ranked in order. When there is an even number of observations, the median is the mean of the two central values.

Therefore, the median of the above data is **310**:

11,243,270 310, 311,345,5300

The **interquartile range** is the difference between the upper and lower quartiles (or 75th and 25th centiles).The quartiles are identified in a similar fashion to the median - the middle observations of the lower and upper halves of the data, as **243** and **345**:

11,243,270 310, 311,345,5300.

Therefore, the interquartile range is $345 - 243 =$ **102**.

4.3 Outliers or missing data?

Summary statistics and exploratory plots can help us identify missing and strange values both of which can have considerable influence on our analyses. Missing values can bias our conclusions unless we can make the assumption that these values are missing completely at random (i.e. independent of observed and unobserved variables). Outlying values should only be excluded if there is reason to question their validity. Therefore, complete and accurate data collection is important.

Lets suppose that upon further investigation, the *5300* Facebook friends observed was erroneous and should have actually been input as *530*. We recalculate the mean and standard deviation as: **289** and **154** respectively.

4.4 Standard error of the mean vs. Standard deviation

The standard deviation is often confused with the standard error of the mean.

The **standard deviation** quantifies the spread or variability of the observations. In Figure 7 can see that the standard deviation of the observations in **Group 1** is **greater** than the standard deviation of the observations in **Group 2**.

The **standard error of the mean** quantifies how precisely we know the true mean.

The standard error of the mean is the standard deviation divided by the square root of the number of observations. In our example data, the standard error of the mean (s.e.m.) is given by:

$se = \frac{sd}{\sqrt{n}} = \frac{154}{\sqrt{7}} = 58$

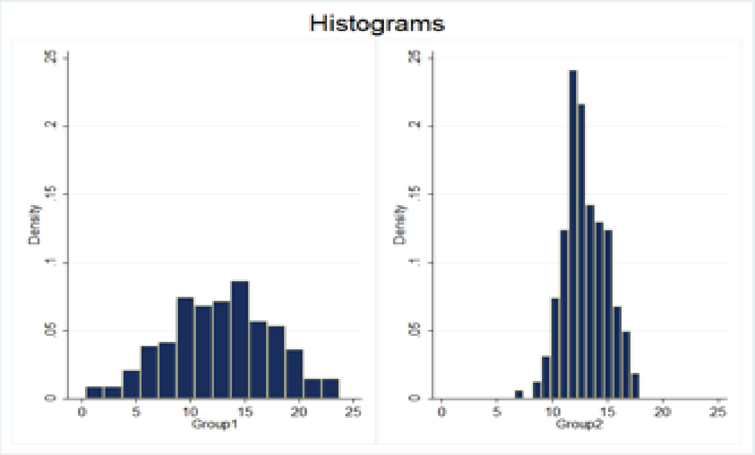
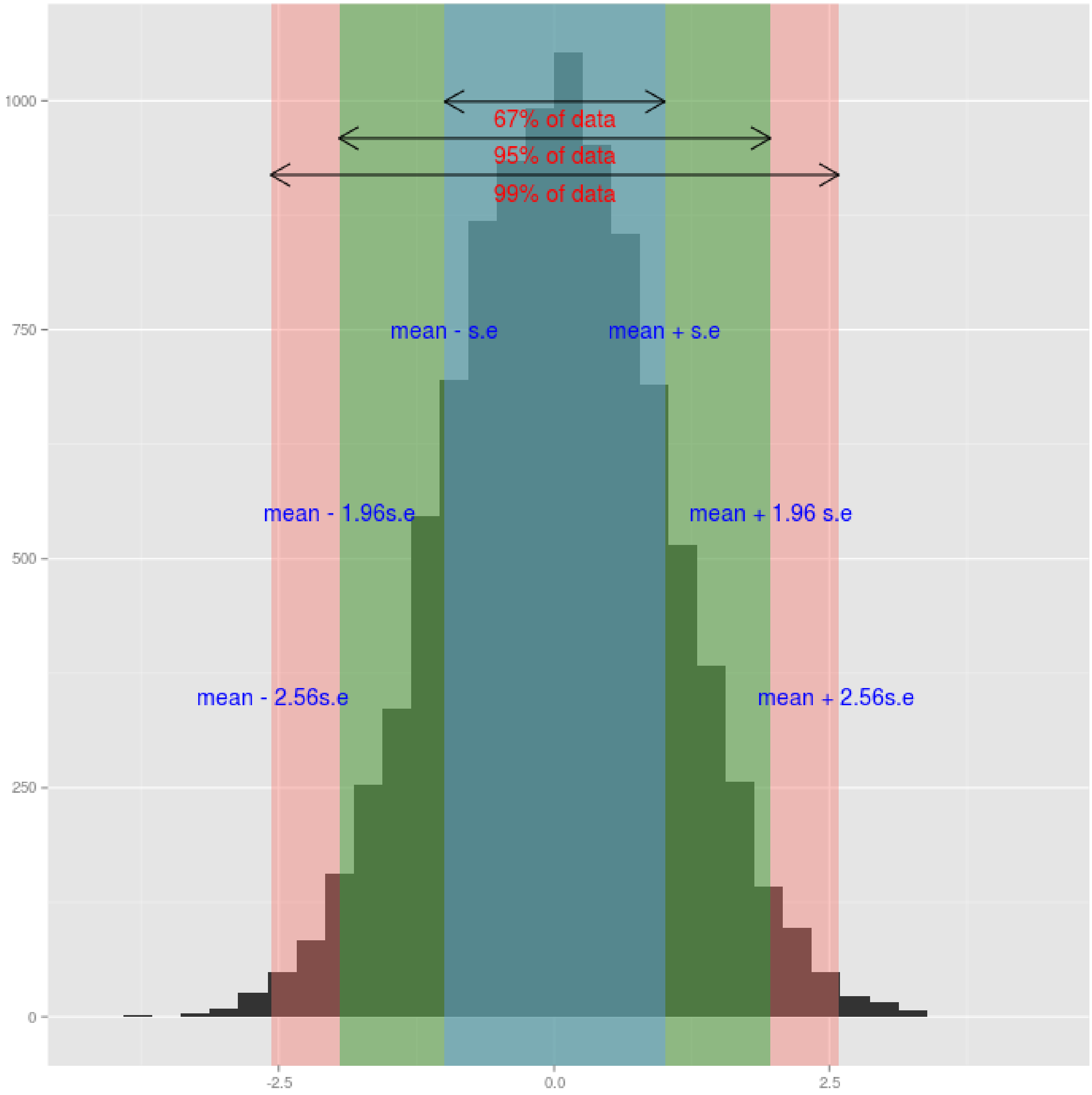


Figure 7: Comparing the standard deviation of two distributions

Increasing our number of observations would not affect the standard deviation but the standard error of the mean would get **smaller**.

We can use the **standard error of the mean** to calculate a **confidence interval** for the mean. The **confidence interval** indicates the uncertainty about the estimate.

The normal distribution plays an important part in confidence interval estimation. Essentially, the frequency distribution of the sample mean is normal, irrespective of the shape of the frequency distribution of the population from which the sample comes. The approximation is better if the population itself is reasonably normal but even if not it gets closer as the sample size increases.



It is most common to calculate a 95% confidence interval as:

$$(mean - 1.96s.e, mean + 1.96s.e) \tag{3}$$

If we were to take many random samples, of equal size, from our population of interest, then the mean estimates of all these samples would be normally distributed. Amongst the 95% confidence intervals of those means, we would expect 95% to contain the true population mean. Hence, for our example above (number of Facebook friends of seven colleagues) the 95% confidence interval for mean number of Facebook friends is:

Mean 289, 95% CI ($289 - (1.96 \times 58), 289 + (1.96 \times 58)$)
Mean 289, 95% CI (175, 402)

The confidence interval is quite wide which indicates that we are quite uncertain of the true mean number of Facebook friends of **all** colleagues.

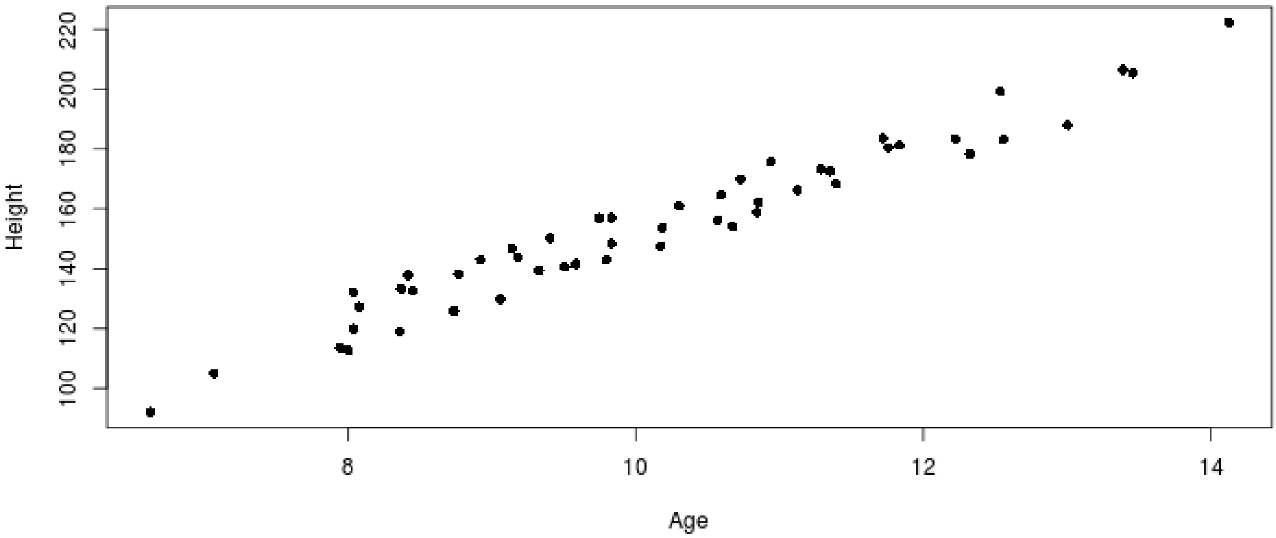
4.5 Graphs

One of the best ways of displaying data is by using a graph. Graphs can make both simple and complex data easier to understand by making it easier to spot trends and patterns. We can use plots to view the distribution of our data (minimum, maximum, mid-point, spread etc) and to ensure that the values in our dataset seem realistic. Many statistical tests rely on the assumption that the data are normally distributed, which can be assessed using histograms or box plots (more later).

In order to draw graphs in R Commander, simply go to the Graphs option in the top menu, and then select the type.

4.5.1 Scatter plots

A scatterplot is an excellent way of displaying the relationship between two continuous variables. For example, a scatterplot can be used to show the relationship between age and height. If there is a fairly clear response variable (height in this case) it should be put on the vertical axis with the explanatory variable on the horizontal axis.

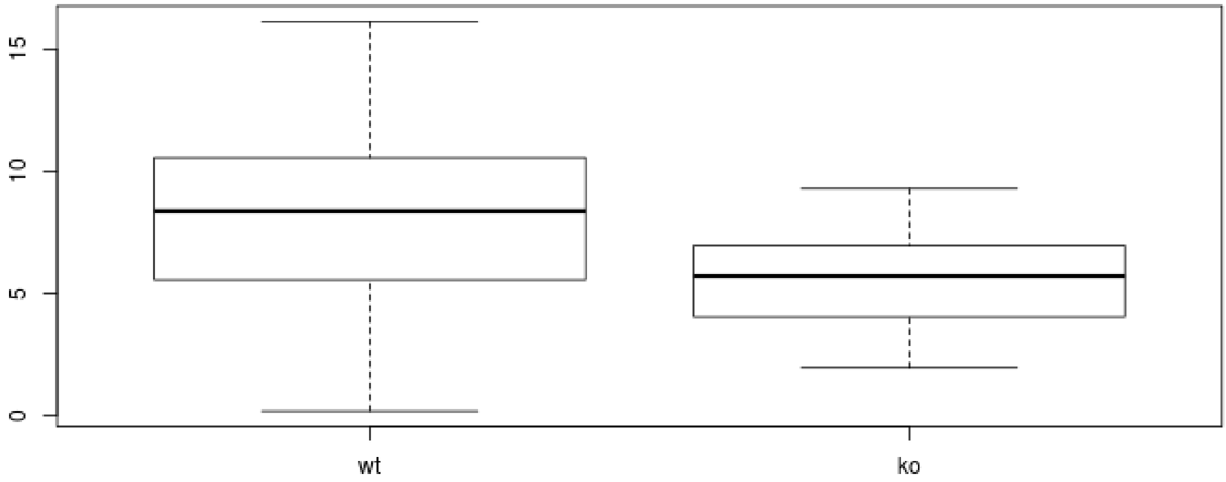


Scatter plots are very useful because they show every point in your dataset. You may be able to spot trends in the data and identify any unusual observations (outliers).

4.5.2 Box plots

A box plot is an excellent way of displaying continuous data when you are interested in the spread of your data. The box of the box plot corresponds to the lower and upper quartiles of the respective observations and the bar within the box, the median. The whiskers of the box plot correspond to the distance between the lower/upper quartile and the **smaller** of: the smaller/largest measurement **OR** 1.5 times the interquartile range.

A disadvantage of the box plot is that you don't see the exact data points. However, box plots are very useful in large datasets where plotting all of the data may give an unclear picture of the shape of your data.



5 Hypothesis testing basic set-up

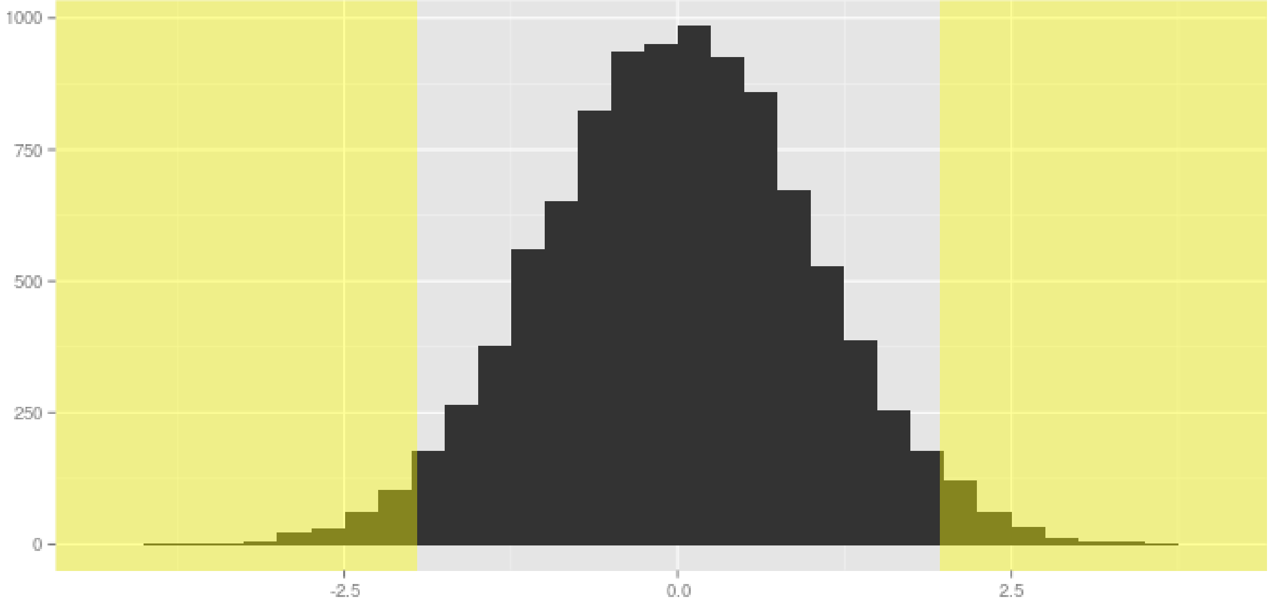
There are four key steps in hypothesis testing:

1. Formulate a **null hypothesis**, H_0. This is the working hypothesis that we wish to disprove.
2. Under the assumption that the **null hypothesis** is true, calculate a **test statistic** from the data.
3. Determine whether the **test statistic** is more extreme than we would expect under the **null hypothesis**, i.e. look at the **p-value**.
4. Reject or do not reject the **null hypothesis**.

As the name suggests, the null hypothesis typically corresponds to a null effect.

For example, there is **no difference** in the measurements in group 1 compared with group 2. A small p-value indicates that the probability of observing such a test statistic as small under the assumption that the null hypothesis is true. If the p-value is below a pre-specified **significance level**, then this is a **significant result** and, we would conclude, there is evidence to reject the null hypothesis.

The **significance level** is most commonly set at 5% and may also be thought of as the **false positive rate**. That is, there is a 5% chance that the null hypothesis is true for data-sets with test statistics corresponding to p-values of less than 0.05 we may wrongly reject the null hypothesis when the null hypothesis is true (false positive).



Equally, we may make **false negative** conclusions from statistical tests. In other words, we may not reject the null hypothesis when the null hypothesis is, in fact, not true. When referring to the false negative rate, statisticians usually refer to **power**, which is 1-false negative rate.

The **power** of a statistical test will depend on:

- The **significance level** - a 5% test of significance will have a greater chance of rejecting the null than a 1% test because the strength of evidence required for rejection is less.
- The **sample size** the larger the sample size, the more accurate our estimates (e.g. of the mean) which means we can differentiate between the null and alternative hypotheses more clearly.
- The **size of the difference or effect** we wish to detect bigger differences (i.e. alternative hypotheses) are easier to detect than smaller differences.
- The **variability**, or standard deviation, of the observations the more variable our observations, the less accurate our estimates which means it is more difficult to differentiate between the null and alternative hypotheses.

	Null hypothesis does not hold	**Null hypothesis holds**
Reject null hypothesis	Correct *True Positive*	Wrong *False positive*
Do no reject null hypothesis	Wrong *False negative*	Correct *True negative*

Table 1: Error definitions

6 T-tests

T-tests can be broken down into two main types: one-sample and two-sample. Both will be discussed below with examples of their applications. The two-sample t-test can also be broken down further into independent and paired t-tests, which will both be discussed below.

6.1 One-sample t-test

A one-sample t-test is the most basic t-test available and should be used if you want to test whether your population mean may be significantly different to a hypothesised mean. As it is rarely possible to measure the whole population, a sample is taken in the hope that it is representative of the wider population.

6.1.1 Example

A microarray supplier claims that their microarray failure rate is 2.1%. Genomics would like to know whether this reflects the failure rates that they've observed over the last 12-months, so they have collected failure rate data on a monthly basis. The data is given in Table 2.

Our **null hypothesis** is that the mean monthly failure rate of the microarrays is **equal** to 2.1%. i.e.

$$\textbf{Mean monthly failure rate} = 2.1\%$$

Our **alternative hypothesis** is that the mean monthly failure rate of the microarrays is **not equal** to 2.1%. i.e.

$$\textbf{Mean monthly failure rate} \neq 2.1\%$$

Month	Monthly failure rate
January	2.90
February	2.99
March	2.48
April	1.48
May	2.71
June	4.17
July	3.74
August	3.04
September	1.23
October	2.72
November	3.23
December	3.40

Table 2: Mean monthly microarray failure rate

To calculate the mean for this sample we add up the 12 values of the monthly failure rate and divide this number by the number of observations in the sample, in this case 12. So the sample mean is:

$$\frac{2.90 + 2.99 + \ldots 3.40}{12} = \frac{34.09}{12} = 2.84 \tag{4}$$

We can see straight away that the sample mean of 2.84% is higher than our hypothesised mean of 2.1%, but we cannot yet say if it is significantly different, that is if the difference is greater than we would expect by chance. This is where the one-sample t-test should be used.

A two-sided test is used when you want to know if the population mean is **different** to a hypothesised value. A one-sided test is used when you want to know if the population mean is either **higher** or **lower** than that hypothesised value and you can justify that observing a difference in one direction would lead to the same action/conclusion as if no difference had been observed.

A two-sided test is always favoured unless there is a strong argument for a one-sided test. In this example, we have not specified in our hypothesis whether we think the mean monthly failure rate of the microarrays is higher or lower than 2.1%, so we will use a two-sided t-test.

The one sample t-test is based on the formula:

$$t_{n-1} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} \tag{5}$$

where;

- $\bar{x}$ is the sample mean
- μ_0 is the hypothesised mean
- s is the sample standard deviation
- n is the sample size

The key assumptions of the one-sample t-test are;

- The observations are independent
- The observations are normally distributed

To perform this test in R Commander, we need to first enter the data. In this example we will enter the data by hand.

Use `Data` → `New data set/..` (Figure 8)

Enter a name for your dataset and click OK. Note that giving two datasets the same name will cause the one currently being held in R's memory with that name to be overwritten with the new dataset of the same name.

(See Figure 9) Click on a cell to enter data into it. Give each column of data a heading by clicking on the 'var' text at the top of the column. At this stage you should also select whether your data is numeric or character. In most cases you should select 'numeric' for continuous data, and 'character'

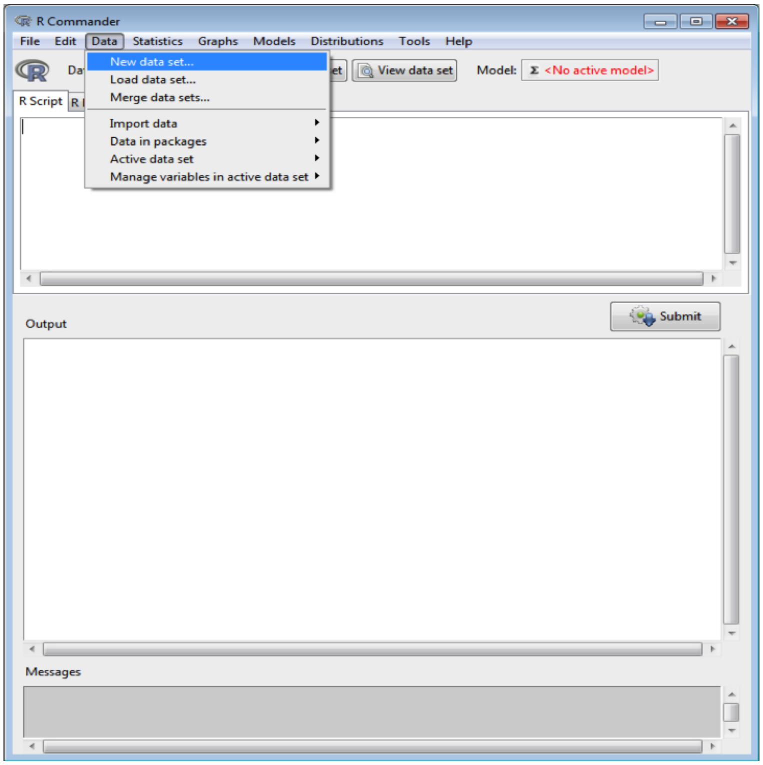


Figure 8: Creating a new dataset

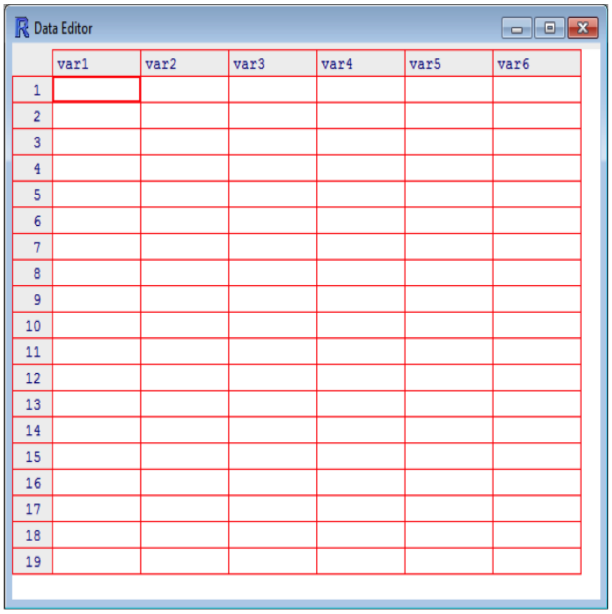
for categorical data. Once you have finished editing the data click on the cross in the top right hand corner of the data editor to close the editor and return to R Commander. Note that you cannot use R Commander whilst the data editor is open.

When you return to R Commander after entering your data there should be some new text in the three R Commander panels. The first panel shows you the R script corresponding to entering your data. The second panel shows the same line of R script, signalling that the line of code has been run. The third panel tells you the name and dimensions of your new dataset.

Before conducting the one sample t-test, we need to check the observations are normally distributed and this can be done by creating a histogram (Figure 10).

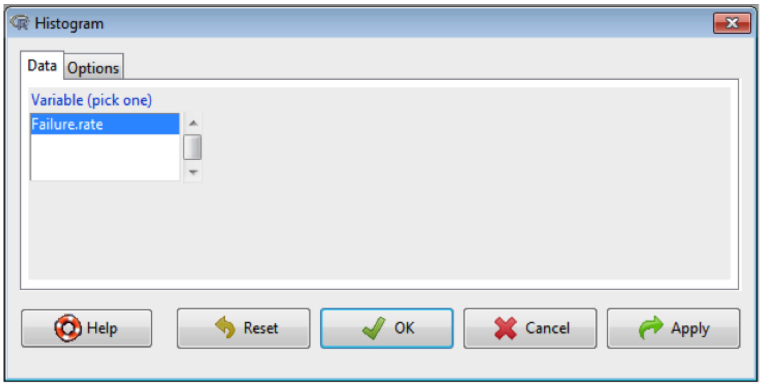
Use `Graphs` → `Histogram...` → `Apply`
By using Apply rather than OK creates your histogram without closing the histogram wizard. If you selected OK instead, your histogram would be produced but the histogram wizard would close. By keeping the histogram wizard open it's easier to go back and make small changes to your graph, e.g. changing axis titles, or adding a main plot title. To make changes to your plot select the Options tab in the histogram wizard. You can change the number of bins (the number of groups the data is split into to create the histogram), the axis labels and the main title for the graph. Click Apply to update your graph. Once you've finished editing your graph click OK.

We want the histogram to be fairly symmetrical with a rough bell shape like that of the plot on the right hand side above. It is often difficult to assess whether data from a small sample are normally



	var1	var2	var3	var4	var5	var6
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						

Figure 9: Creating a dataset by-hand



Variable (pick one)
Failure.rate

Figure 10: Creating a histogram

distributed, as a single observation can distort the shape of the histogram.

Here (Figure 11), the histogram is **fairly symmetrical with a rough bell shape** (it doesn't have to be perfect!) so the normality assumption seems reasonable. Slight deviations from normality are rarely a problem as the t-test is fairly robust.

Now, we can carry out the one-sample t-test to test whether the mean microarray failure rate = 2.1%.

Use `Statistics` → `Means` → `Single-sample t-test..`
The one-sample t-test wizard will open (Figure 13). Insert your hypothesised mean next to Null hypothetical mu - in this example this is 2.1. Keep all other options the same and click on OK. The output from your one sample t-test will appear in the output panel. You will also see that some new text has appeared in the R Script panel corresponding to the t-test and its parameters.

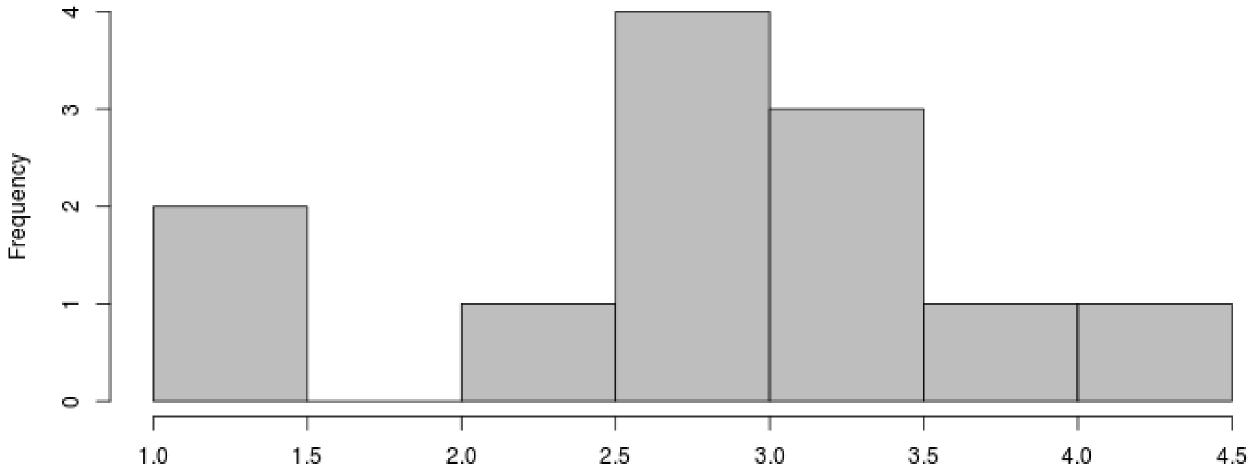
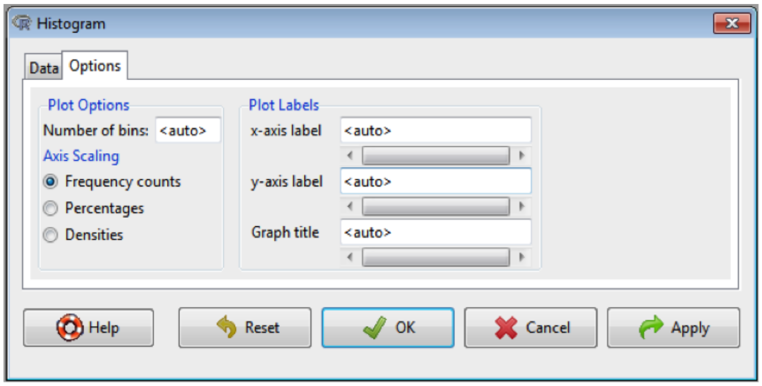


Figure 11: Histogram of the microarray failure data with the default settings



Plot Options

Number of bins: <auto>

Axis Scaling

● Frequency counts

○ Percentages

○ Densities

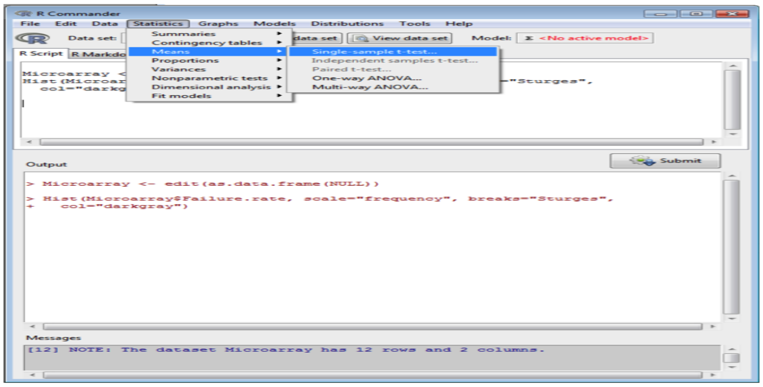
Plot Labels

x-axis label <auto>

y-axis label <auto>

Graph title <auto>

Figure 12: Modifying the histogram



```
Microarray <
Hist(Microar                                                    ="Sturges",
  col="darkg
```

```
> Microarray <- edit(as.data.frame(NULL))
> Hist(Microarray$Failure.rate, scale="frequency", breaks="Sturges",
+   col="darkgray")
```

```
[12] NOTE: The dataset Microarray has 12 rows and 2 columns.
```

Figure 13: Performing a one-sample t-test

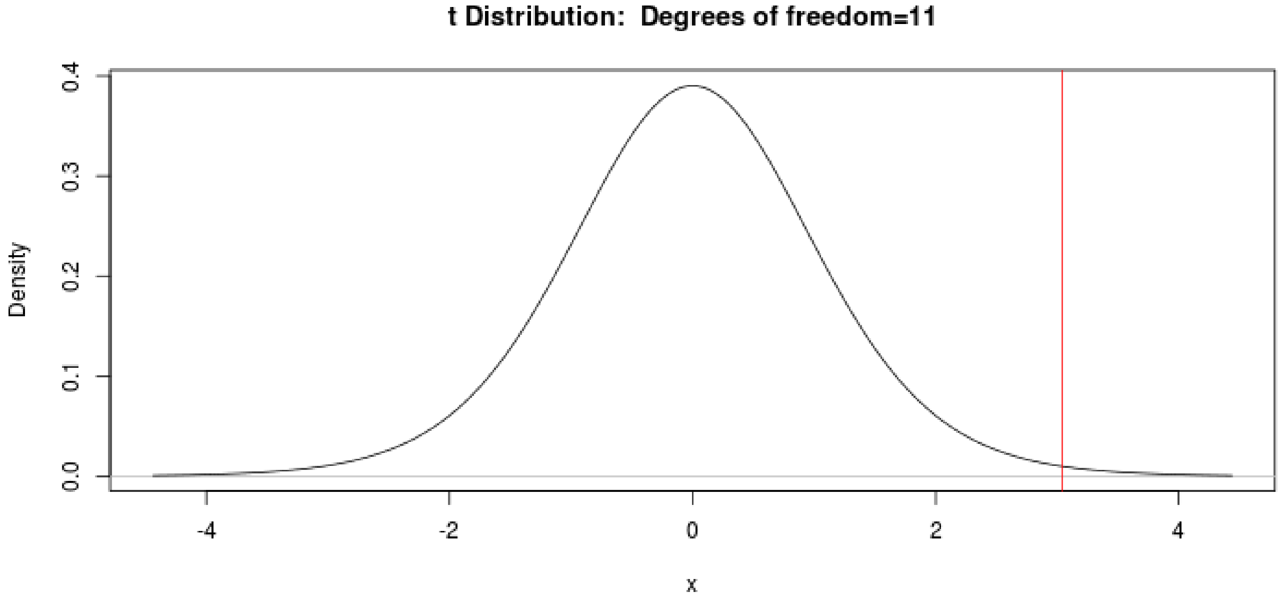


Figure 14: t-distribution with 11 degrees of freedom. The t-statistic 3.07 is indicated

```
##
##  One Sample t-test
##
## data:  microarray[, 2]
## t = 3.0647, df = 11, p-value = 0.01076
## alternative hypothesis: true mean is not equal to 2.1
## 95 percent confidence interval:
##  2.308786 3.372880
## sample estimates:
## mean of x
##  2.840833
```

Under the one sample t test, the t-statistic has been calculated by:

$$t_{n-1} = t_11 = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} = \frac{2.84 - 2.10}{0.84/\sqrt{12}} = \frac{0.74}{0.24} = 3.07 \tag{6}$$

Assuming that the null hypothesis is true, our t-statistic comes from the t distribution with 11 degrees of freedom, shown in the Figure 14.

Under the null hypothesis, that the mean monthly failure rate of the microarrays = 2.1%, we can see that the probability of observing a value of the t-statistic as extreme as 3.07 is very small. This **p-value** is

$$P(T \leq 3.07 | T \geq 3.07) = 0.01 \tag{7}$$

As the p-value of 0.01 is less than 0.05 (5%), there is **evidence to reject the null hypothesis** and conclude that there is evidence to suggest that the failure rate of the microarrays from this supplier is not 2.1%.

6.2 Two-sample t-test

A two-sample t-test should be used if you want to **compare the measurements of two populations**. There are two types of two-sample t-test: independent (unpaired) and paired (dependent).

An independent two-sample t-test is used when the two samples are **independent** of each other, e.g. comparing the mean response of two groups of patients on treatment vs. control in a clinical trial. As the name suggests, a paired two-sample t-test is used when the two samples are paired, e.g. comparing the mean blood pressure of patients before and after treatment (two measurements per patient).

6.2.1 Independent two-sample t-test

Breed A		Breed B	
Subject	Weight	Subject	Weight
1	20.77	21	15.51
2	9.08	22	12.93
3	9.80	23	11.50
4	8.13	24	16.07
5	16.54	25	15.51
6	11.36	26	17.66
7	11.47	27	11.25
8	12.10	28	13.65
9	14.04	29	14.28
10	16.82	30	13.21
11	6.32	31	10.28
12	17.51	32	12.41
13	9.87	33	9.63
14	12.41	34	14.75
15	7.39	35	9.81
16	9.23	36	13.02
17	4.06	37	12.33
18	8.26	38	11.90
19	10.24	39	8.98
20	14.64	40	11.29

Table 3: Weights of 2 breeds (A and B) of 4 week-old male mice

Example: A researcher is interested in the effect of breed on weight in 4 week old male mice. 40 male mice were used, 20 of breed A and 20 of breed B. The data are shown in Table 3.

So, the researcher wants to test the **null hypothesis** that the mean weight of breed A is **equal** to the mean weight of breed A in 4 week-old male mice.

Mean weight of breed A = Mean weight of breed B

Our **alternative hypothesis** is that the mean weight of breed A is **not equal** to the mean weight of breed B in 4 week-old male mice.

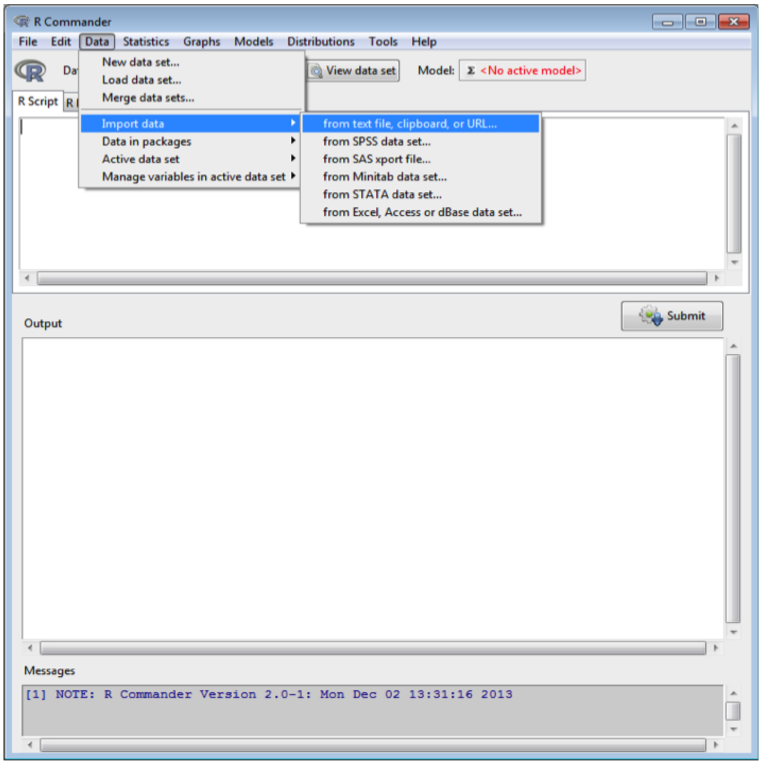
Mean weight of breed A $\neq$ Mean weight of breed B

To perform the independent two-sample t-test, we calculate the following t-statistic from our data:

$$t_{df} = \frac{\bar{X}_A - \bar{X}_B}{s.e(\bar{X}_A - \bar{X}_B)} \qquad (8)$$

where;

- $\bar{X}_A$ is the mean weight of mice in breed A



```
[1] NOTE: R Commander Version 2.0-1: Mon Dec 02 13:31:16 2013
```

Figure 15: Importing data from a text file

- $\bar{X}_B$ is the mean weight of mice in breed B
- $s.e(\bar{X}_A - \bar{X}_B)$ is the standard error of the difference in mean weights
- df is the degrees of freedom and is equal the total number of mice (breed A and breed B) minus the number of independent groups (N -2)

To perform this test in R Commander, start by importing the data from a text file (Figure 15)

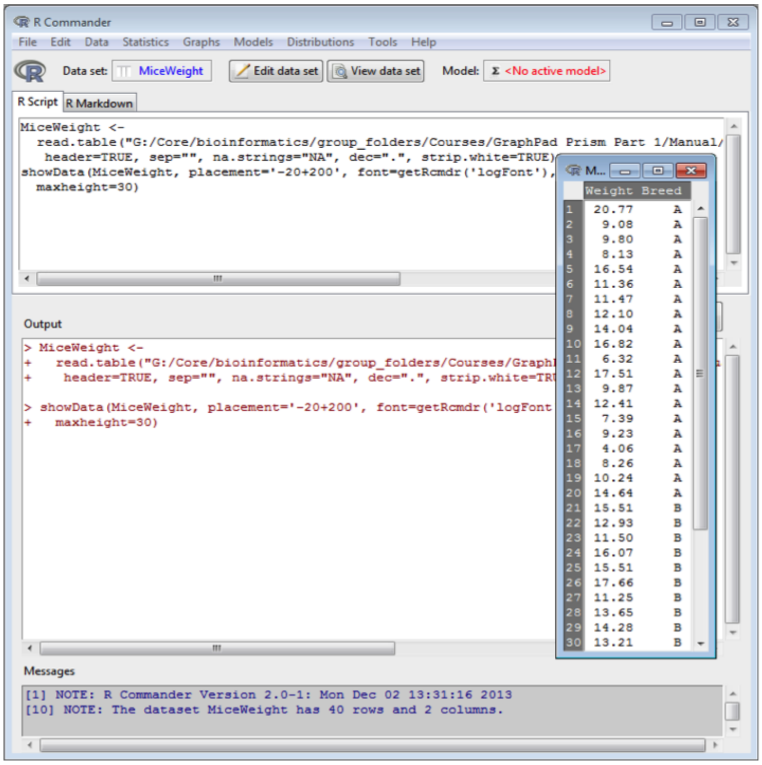
Give the dataset an appropriate name, keep all the other options as they are and click on `OK`.

Next, check your data by clicking on the `View data set button`. This will open an additional window displaying your data (Figure 16) . Ensure that the data has been read in correctly and close the data viewer.

The independent two-sample t-test has similar assumptions to the one-sample t-test:

- The observations are **independent**
- The measurements in each group are **normally distributed**
- The variances of the measurements in the two groups are **equal**

We will come to the third assumption later. To assess the **distribution of the observations** in each of the two groups, we produce histograms just as we did for the one-sample t-test. However when we have two or more independent groups we must first partition the data by group. We have two groups (breed A and breed B), so we must partition our dataset into two subgroups - one for each breed.



```
MiceWeight <-
  read.table("G:/Core/bioinformatics/group_folders/Courses/GraphPad Prism Part 1/Manual/
   header=TRUE, sep="", na.strings="NA", dec=".", strip.white=TRUE)
showData(MiceWeight, placement='-20+200', font=getRcmdr('logFont'),
  maxheight=30)
```

```
> MiceWeight <-
+   read.table("G:/Core/bioinformatics/group_folders/Courses/Graph
+    header=TRUE, sep="", na.strings="NA", dec=".", strip.white=TRU

> showData(MiceWeight, placement='-20+200', font=getRcmdr('logFont
+   maxheight=30)
```

```
   Weight Breed
1   20.77     A
2    9.08     A
3    9.80     A
4    8.13     A
5   16.54     A
6   11.36     A
7   11.47     A
8   12.10     A
9   14.04     A
10  16.82     A
11   6.32     A
12  17.51     A
13   9.87     A
14  12.41     A
15   7.39     A
16   9.23     A
17   4.06     A
18   8.26     A
19  10.24     A
20  14.64     A
21  15.51     B
22  12.93     B
23  11.50     B
24  16.07     B
25  15.51     B
26  17.66     B
27  11.25     B
28  13.65     B
29  14.28     B
30  13.21     B
```

```
[1] NOTE: R Commander Version 2.0-1: Mon Dec 02 13:31:16 2013
[10] NOTE: The dataset MiceWeight has 40 rows and 2 columns.
```

Figure 16: Viewing a dataset

Use `Data` → `Active data set` → `Subset active data set..` (Figure 17)

In this instance, when you partition the data you want to keep both variables (weight and breed) so make sure the `Include all variables` box is ticked. Under `Subset expression`, define the way you want to partition the data. This should take the form of:

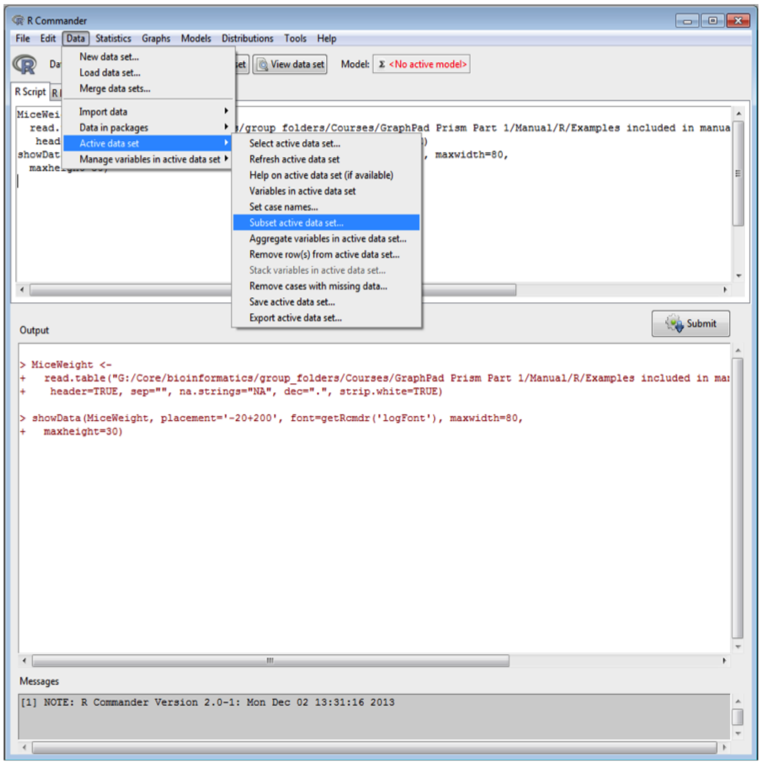
```
variablename == "subsetlevel"
```

In this case our variable name is Breed, and we want to subset the mice that are labelled as breed A. Hence we use `Breed=="A"`. Choose a new name for your subset and click on OK (Figure 18) Repeat this process to create a breed B subset, ensuring you change the active dataset back to `MiceWeight` first.

Refer to the one-sample t-test section for details on how to produce the histograms, ensuring that you produce a histogram for each of your two independent groups. Ensure that your active dataset is the one you wish to create the plot from.

Here, the histogram for each breed is fairly symmetrical with a rough bell shape (Figure 19).

The normality assumption seems reasonable and so we can carry out the independent two-sample t-test. The third assumption of the independent two-sample t-test is that the **data in each of the two**

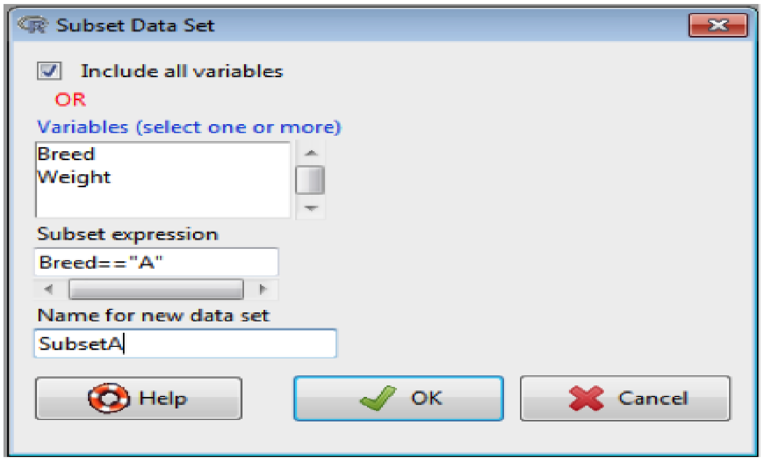

R Commander
File Edit Data Statistics Graphs Models Distributions Tools Help
New data set...
Load data set...
Merge data sets...
Import data
Data in packages
Active data set
Manage variables in active data set
Select active data set...
Refresh active data set
Help on active data set (if available)
Variables in active data set
Set case names...
Subset active data set...
Aggregate variables in active data set...
Remove row(s) from active data set...
Stack variables in active data set...
Remove cases with missing data...
Save active data set...
Export active data set...
View data set
Model: Σ <No active model>
R Script
Submit
Output
Messages

```
> MiceWeight <-
+   read.table("G:/Core/bioinformatics/group_folders/Courses/GraphPad Prism Part 1/Manual/R/Examples included in mai
+    header=TRUE, sep="", na.strings="NA", dec=".", strip.white=TRUE)

> showData(MiceWeight, placement='-20+200', font=getRcmdr('logFont'), maxwidth=80,
+   maxheight=30)
```

```
[1] NOTE: R Commander Version 2.0-1: Mon Dec 02 13:31:16 2013
```

Figure 17: Creating a 'subset'



☑ Include all variables

Subset expression: Breed=="A"

Name for new data set: SubsetA

Figure 18: Creating a subset of the mice weight data

groups should have approximately equal variance. This can be assessed using an **F-test**. For this use

`Statistics` → `Variances` → `Two-variances F-Test` (Figure 20)

On the options tab keep the default options (Figure 21) This gives the result

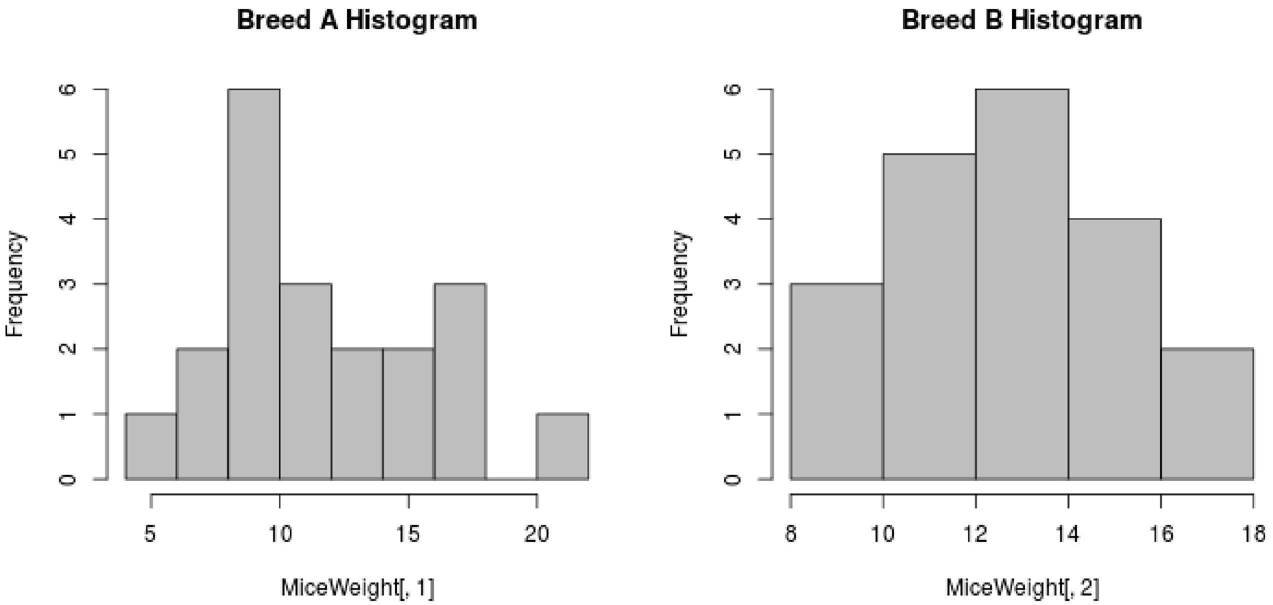
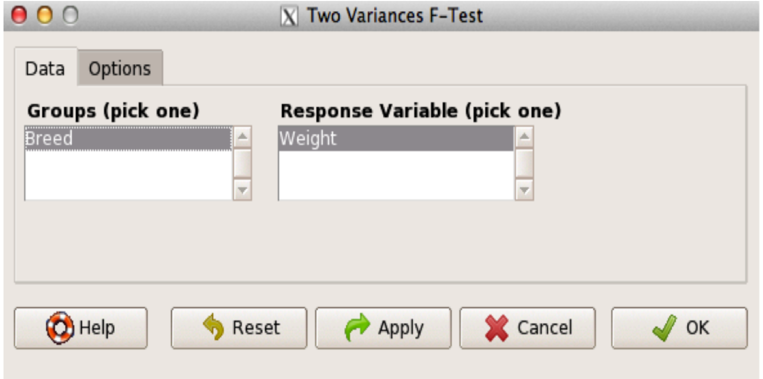


Figure 19: Histogram of Mice Weight data



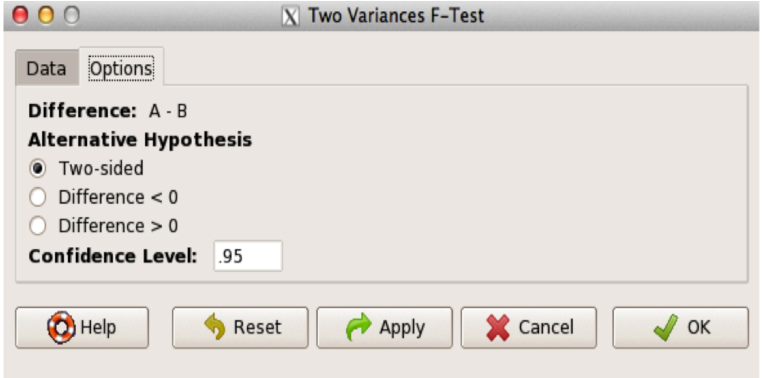
Groups (pick one)

Breed

Response Variable (pick one)

Weight

Figure 20: Assessing the variance of two groups



Difference: A - B

Alternative Hypothesis

● Two-sided

○ Difference < 0

○ Difference > 0

Confidence Level: .95

Figure 21: Options for the two-variances F-test

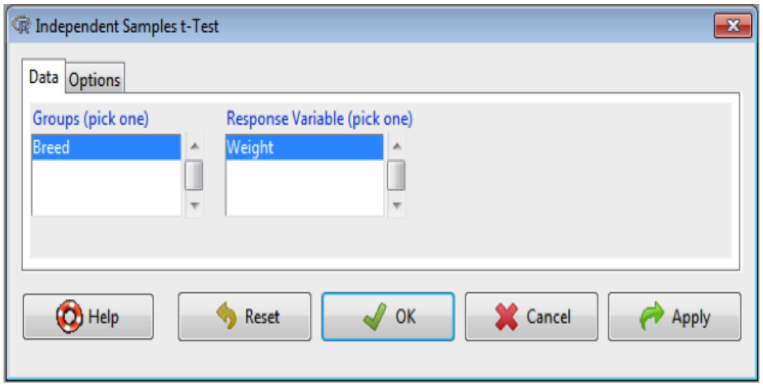


Figure 22: Setting-up the Independent samples t-test

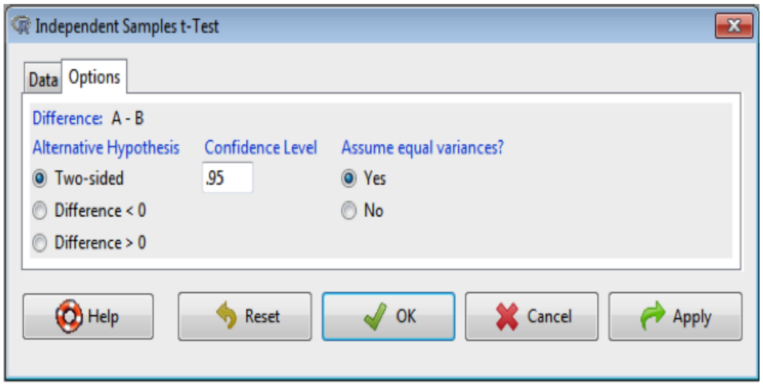
```
##
##  F test to compare two variances
##
## data:  MiceWeight[, 1] and MiceWeight[, 2]
## F = 3.2131, num df = 19, denom df = 19, p-value = 0.01447
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.271785 8.117743
## sample estimates:
## ratio of variances
##           3.213102
```

In this example, the p-value given under the **F test to compare variances** was 0.015 which is strongly significant. This means that we **cannot assume equal variance** between the two different breeds of mice and so, a slightly different formulation of the two-sample t-test is needed. When selecting the two-sample t-test, a **Welch's correction needs to be applied**. You may be tempted to apply the Welch's correction routinely, even in cases where the variance is similar in your two groups. However, this is not recommended because the use of the Welch's correction has a large impact on the degrees of freedom of the test. When the variances are similar, an unpaired t-test *with* a Welch's correction is much less powerful than a standard unpaired t-test *without* the Welch's correction. Use the Welch correction when carrying out the t-test as follows: Firstly, ensure `MiceWeight` is your active dataset.

Use: `Statistics` → `Means` → `Independent samples t-test...`
Your measurement variable should be selected under **Response** Variable and your grouping variable should be selected under **Groups** (Figure 22).

Clicking on the `Options` tab will bring up further useful options (Figure 23). Here you can choose either a two-sided or one-sided test and change the confidence level. You can also choose to apply a Welch correction by selecting `No` under `Assume equal variance?`. Leave the other options on their default settings and click `Apply`.

This yields the result



Difference: A - B

Alternative Hypothesis
● Two-sided
○ Difference < 0
○ Difference > 0

Confidence Level
.95

Assume equal variances?
● Yes
○ No

Figure 23: Options for the two-sample t-test

```
t.test(MiceWeight[,1], MiceWeight[,2])

##
##  Welch Two Sample t-test
##
## data:  MiceWeight[, 1] and MiceWeight[, 2]
## t = -1.2107, df = 29.782, p-value = 0.2355
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.4842299  0.8912299
## sample estimates:
## mean of x mean of y
##   11.5020   12.7985
```

The mean weight of Breed A male mice at 4 weeks old was 11.50g, whilst for Breed B the mean weight was 12.80g. The Welch-corrected t-statistic is given by:

$$t_{29} = \frac{\bar{X}_A - \bar{X}_B}{s.e(\bar{X}_A - \bar{X}_B)} = 1.21 \tag{9}$$

We can see that the t-statistic we observe is consistent with the null hypothesis, that the mean weight of 4 week old male mice is the same for breeds A and B. That is, the probability of observing a t-statistic of 1.21 or more, or -1.21 or less, is quite high:

$$P(T \leq -1.21 | T \geq 1.21) = 0.24 \tag{10}$$

This is not a significant result (p> 0.05), so there is **no evidence of a difference** in the weight of male mice at 4 weeks old between Breeds A and B.

	A	B	Difference
1	1201.33	1155.98	-45.35
2	1029.64	1020.82	-8.82
3	895.57	881.21	-14.36
4	842.14	830.78	-11.36
5	903.07	897.06	-6.01
6	1311.57	1262.73	-48.84
7	833.52	823.06	-10.46
8	1007.66	951.01	-56.65
9	1465.51	1450.98	-14.53
10	967.82	978.15	10.33
11	812.72	778.26	-34.46
12	884.08	823.57	-60.51
13	1358.56	1335.78	-22.78
14	1280.10	1293.91	13.81
15	942.38	925.75	-16.63
16	884.33	891.34	7.01
17	930.09	892.02	-38.07
18	1146.75	1132.80	-13.95
19	881.50	847.78	-33.72
20	1315.22	1337.80	22.58

Table 4: Cellularity at two sites of disesase

6.2.2 Paired two-sample t-test

Example: 20 patients with advanced cancer were studied using MRI imaging. Cellularity was measured for each individual patient by estimating water movement. We want to know whether there is a significant difference in the cellularity between two sites in the body; A and B. The data are shown in Table 4. We want to test the **null hypothesis** that the mean cellularity at site A is equal to the mean cellularity at site B. This is like saying:

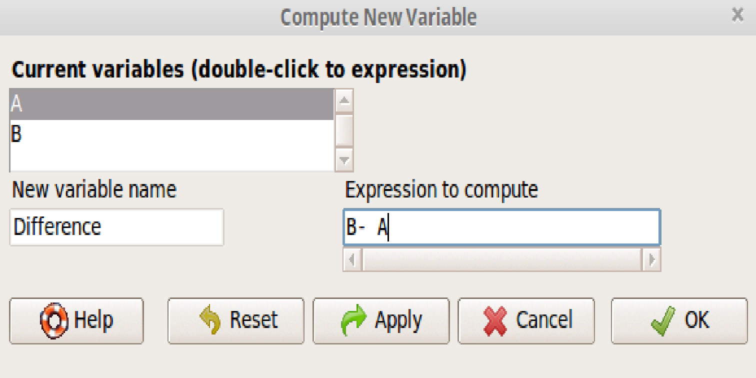
Mean cellularity at site A = mean cellularity at site B

Essentially, this two-sample test corresponds to a formal comparison of the **differences between each pair** of cellularities with 0 (so a one-sample t-test). We could reformulate our null hypothesis as:

Mean difference in cellularities at site A and site B =0

Our **alternative hypothesis** is that the mean cellularity at site A is **not equal** to the mean cellularity at site B. This is like saying:

Mean cellularity at site A $\neq$ Mean cellularity at site B



New variable name: Difference

Expression to compute: B- A

Figure 24: Computing a new variable

To perform the paired two-sample t-test, we calculate the following t-statistic from our data:

$$t_{n-1} = t_{19} = \frac{\bar{X}_{A-B}}{s.e.(\bar{X}_{A-B})} \tag{11}$$

where

- $\bar{X}_{A-B}$ is the mean difference in cellularities between the two sites
- $s.e(\bar{X}_{A-B})$ is the standard error of the mean difference in cellularities

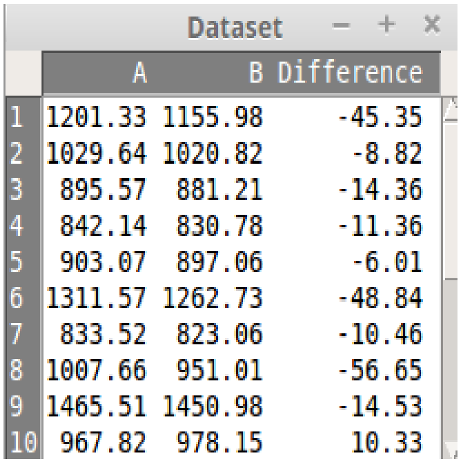
The assumptions of the paired t-test coincide with those of the one-sample t-test:

- The observed **differences** are **independent**
- The observed **differences** are **normally distributed**

Import the data. Compute a new variable to represent the **difference** column (Figure 24)

`Data` → `Manage variables in active data set` → `Compute new variable`
Click on the View data set button to check if the Difference column has been added to the dataset (see Figure 25).

In the calculation of the difference between Site A and Site B column, we need to choose either one as our baseline; this will simply determine whether we calculate A-B or B-A. The results of the paired t-test will be the same either way, but summary statistics such as the mean and confidence intervals will be either positive or negative depending on which column you choose as your baseline, and similarly the histogram with be either on the positive or negative scale (the overall shape will be identical but will be flipped on the vertical axis). In this example, the A column was used as the baseline, so the difference column calculated represents the calculation B-A.



	A	B	Difference
1	1201.33	1155.98	-45.35
2	1029.64	1020.82	-8.82
3	895.57	881.21	-14.36
4	842.14	830.78	-11.36
5	903.07	897.06	-6.01
6	1311.57	1262.73	-48.84
7	833.52	823.06	-10.46
8	1007.66	951.01	-56.65
9	1465.51	1450.98	-14.53
10	967.82	978.15	10.33

Figure 25: Verifying that Difference column has been added to the dataset

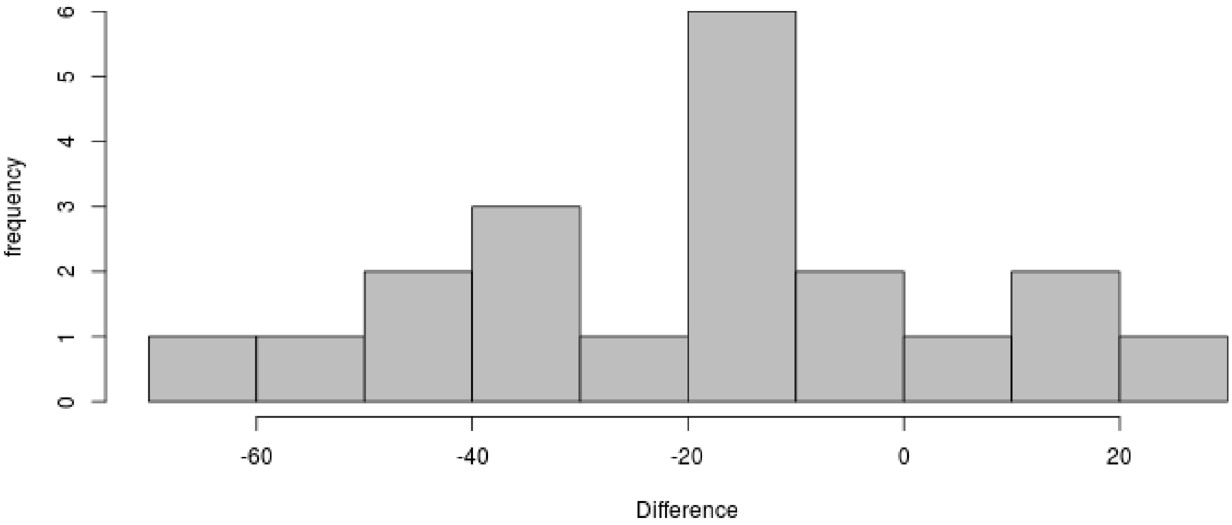
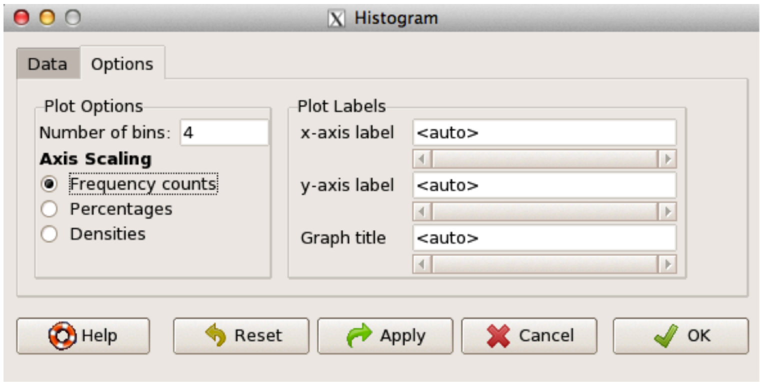


Figure 26: Histogram of cellularity data with default settings

One can then draw a histogram of the paired differences (see Figure 26):

In this example, the histogram probably has too many bars (bins) given the small sample size (just 20 observations). We can change the number of bins on the options tab (Figure 27) so that we have fewer, wider bins. This now produces Figure 28:

Note that the histogram will be flipped on the vertical axis if the difference is calculated as B - A rather than A - B, but this won't impact the end result of the test.



Plot Options
Number of bins: 4
Axis Scaling
● Frequency counts
○ Percentages
○ Densities

Plot Labels
x-axis label <auto>
y-axis label <auto>
Graph title <auto>

Figure 27: Changing the number of bins

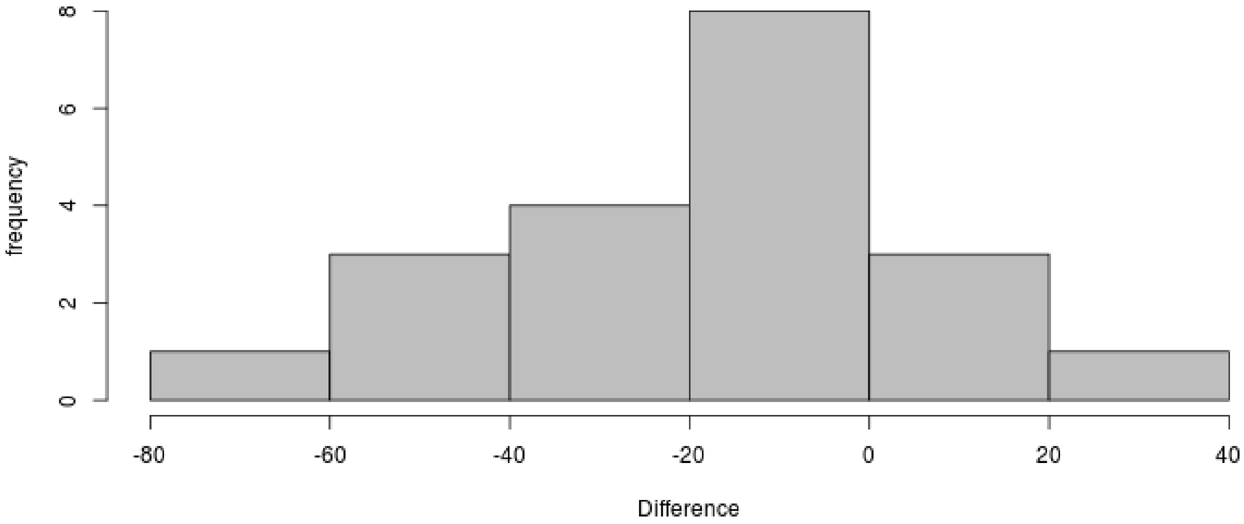


Figure 28: Histogram with user-defined bins

If satisfied with the normality assumption, we can go ahead with the paired two-sample t-test (Figure 29).

```
t.test(cell$B,cell$A,paired=TRUE)

##
##  Paired t-test
##
## data:  cell$B and cell$A
## t = -3.6624, df = 19, p-value = 0.001656
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -30.076046  -8.200954
```

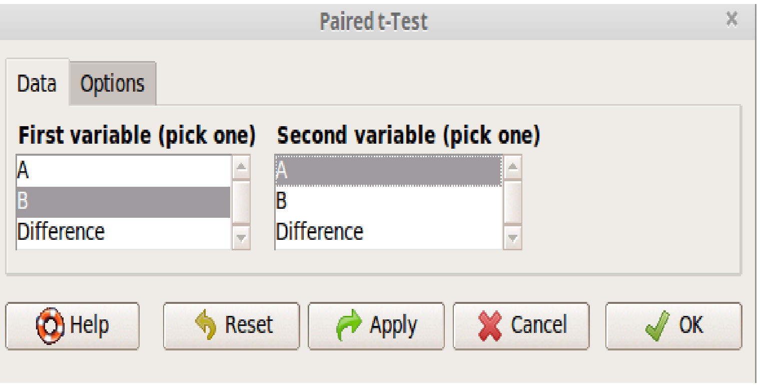


Figure 29: Running the paired-samples t-test

```
## sample estimates:
## mean of the differences
##                -19.1385
```

The mean difference in cellularity between the two sites of disease was 19.14 units. The corresponding t-statistic is:

$$t_{n-1} = t_{19} = \frac{\bar{X}_{A-B}}{s.e(\bar{X}_{A-B})} \tag{12}$$

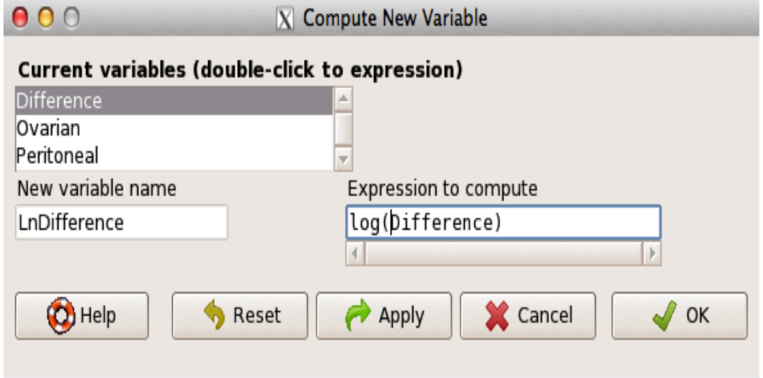
Under the null hypothesis that there is no difference in the cellularities between the two sites of disease, we can see that the probability of observing such a large t-statistic is very small: the p-value is 0.0017.

This is a significant result ($p < 0.05$), so there is **evidence of a difference** in the cellularity between Site A and Site B in patients with advanced cancer.

6.3 What to do if the normality assumption is unreasonable?

There may be instances where normality is hard to determine from histograms, for example, where the sample size is small. In these situations, we may need to draw on the experience of similar sets of measurements. Bland and Altman (2009) observed that *"body size measurements are usually approximately normal, as are the logarithms of many blood concentrations and the square roots of counts."* In other instances, normality may be an unreasonable assumption to make and a t-test is then inappropriate. There are two main options in this circumstance:

- Transformations
- Non-parametric tests



New variable name
LnDifference

Expression to compute
log(Difference)

Figure 30: Apply a log-transformation to data

6.3.1 Transformations

Sometimes, a simple transformation is enough to normalise your data. In biological sciences, a natural log-transformation is often used. Each number in your dataset is replaced by its log and the histogram is produced on this log-transformed data. If normality appears to be a reasonable assumption to make, the t-test is then performed on the log-transformed data. This can be a very simple approach although care must be taken to interpret your results correctly, bearing in mind that you are now working on the log scale. For example, if on the log-scale the mean height in group A is 5.06 and in group B is 5.15, these could be back transformed by exponentiating the values: $e^{5.06} = 157.59$cm and $e^{5.15} =$ 172.43cm. In most cases it is more meaningful to give results which are back-transformed. Data can be log-transformed in R Commander by

`Data` → `Manage variables in active data set` → `Compute new variable`
This can be used when looking at one or two groups of data, regardless of whether observations are independent or paired.

Note: a log-transformation is not suitable if there are a large number of zero's in your dataset.

6.3.2 Non-Parametric tests

As an alternative to transforming your data, or if a transformation fails to normalise your data, there are non-parametric tests available which don't make any assumptions about the distribution of your data. Table 5 shows the non-parametric test than can be used in place of the different types of t-test where the assumptions of normality are unreasonable. Note, however, that the non-parametric tests still have their own set of assumptions.

There are options for these non-parametric tests within R commander:

`Statistics` → `Nonparametric tests`

Parametric test	**Non-parametric equivalent**
One-sample t-test	One-sample Wilcoxon signed rank test
Independent two-sample t-test	Mann-Whitney U test
Paired two-sample t-test	Matched-pairs Wilcoxon signed rank test

Table 5: T-tests and their non-parametric equivalents

7 Tests for categorical variables

When working with categorical variables, we are usually interested in the **frequencies of the different categories** in our sample. To display data for two or more categorical variables, cross-tabulations, or contingency tables, are commonly used - with 2 x 2 tables being the simplest. We can then test whether there is an **association between the row factor and the column factor** by a chi-squared test or a Fishers exact test. Table 6 shows an example of a contingency table.

	Column Factor		
Row factor	C1	C2	**Total**
R1	a	b	a+b
R2	c	d	c +d
Total	a +c	b + d	n = a + b + c + d

Table 6: Example of a contingency table

These tests do not give a measure of effect size; they only give a p-value suggesting whether or not an association exists between the two variables.

7.1 Chi-squared tests

Example: A trial was conducted to assess the effectiveness of a new **treatment** versus a **placebo** in reducing tumour size in patients with ovarian cancer. We want to know whether or not there is an association between treatment group and the **incidence of tumour shrinkage**.

The **null hypothesis** is that there is **no association** between treatment group and tumour shrinkage.

The **alternative hypothesis** is that there is **some association** between treatment group and tumour shrinkage.

	Tumour Shrinkage		
Treatment group	No	Yes	**Total**
Treatment	44	40	84
Placebo	24	16	40
Total	68	56	124

Table 7: Observed frequencies for chi-squared test

The data in Table 7 can be used to calculate the chi-squared statistic. The calculations for the chi-squared test are based on the expected frequency within each entry of the 2 x 2 table.

From Table 7 the expected frequencies can be calculated and are shown in Table 8. The expected frequency for the entry in row i and column j is given by:

$$row_i\,total \times \frac{column_j\,total}{overall\,total} \tag{13}$$

	Tumour Shrinkage		
Treatment group	No	Yes	**Total**
Treatment	46.06	37.94	84
Placebo	21.94	18.06	40
Total	68	56	124

Table 8: Observed frequencies for chi-squared test

As you can see, the values in the shaded part of the two tables of observed and expected frequencies are very similar.The chi-squared test statistic is calculated using the following formula:

$$\chi^2_{rows-1,columns-1} = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{14}$$

where;

- O_{ij} is the observed frequency for a single cell in row i and column j
- E_{ij} is the expected frequency for the same entry in row i and column j
- $rows$ is the number of rows; $columns$ is the number of columns

In the example dataset;

$$\chi^2_1 = \frac{(44-46.06)^2}{46.06} + \frac{(40-37.94)^2}{37.94} + \frac{(24-21.94)^2}{21.94} + \frac{(16-18.06)^2}{18.06} = 0.63 \tag{15}$$

Looking at the relevant chi-squared distribution, with one degree of freedom (Figure 31), we can see that the chi-squared statistic that we observe is **consistent with the null hypothesis** that there is no association between tumour shrinkage and treatment group.

Under the null hypothesis, the probability of observing a chi-squared statistic of 0.63 is

$$P(\chi^2_1 \geq 0.63) = 0.43 > 0.05 \tag{16}$$

Therefore, we do not reject our null hypothesis and conclude that there is no evidence of an association between treatment group and incidence of tumour shrinkage.

To perform the Chi-squared test in R commander, start by setting up an empty contingency data table via

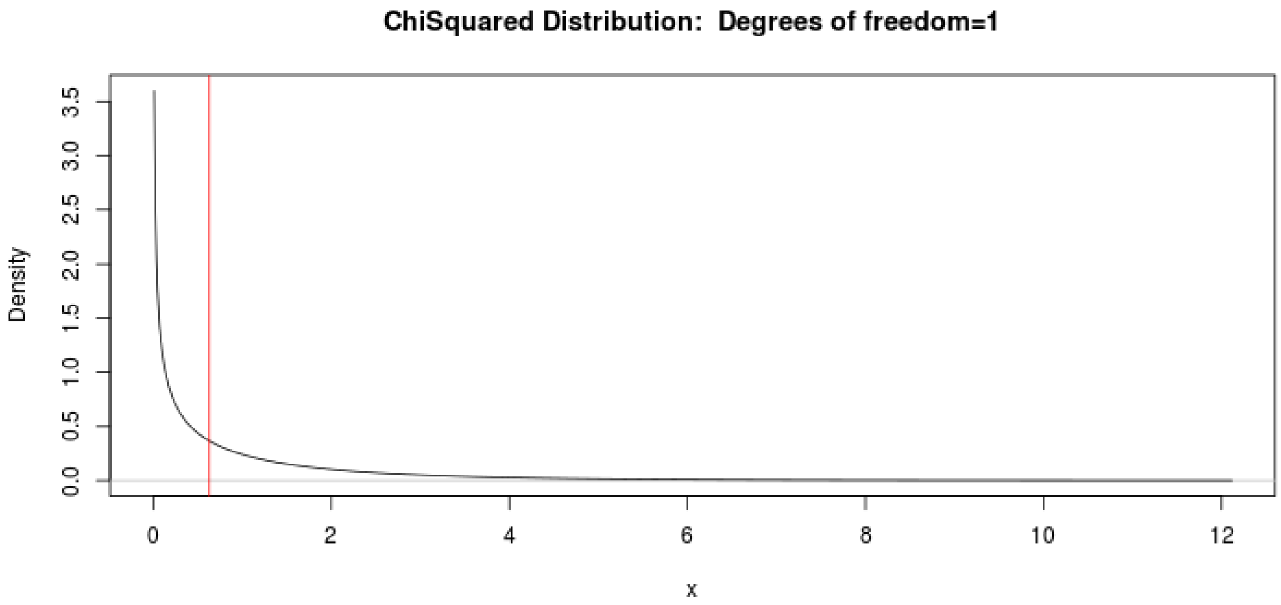
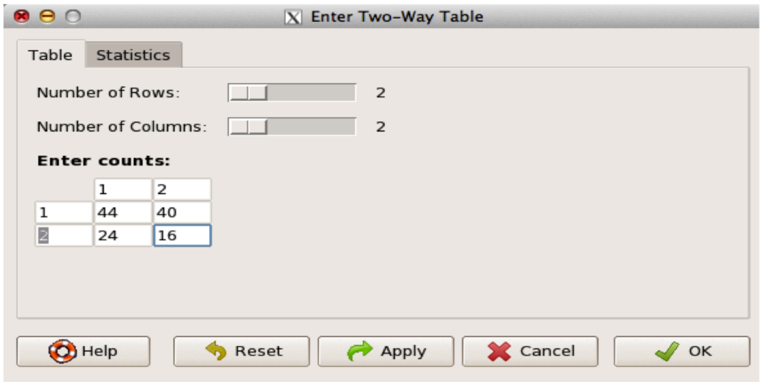


Figure 31: χ^2 distribution with 1 degree of freedom. Test statistic of 0.63 is indicated



Number of Rows: 2

Number of Columns: 2

Enter counts:

	1	2
1	44	40
2	24	16

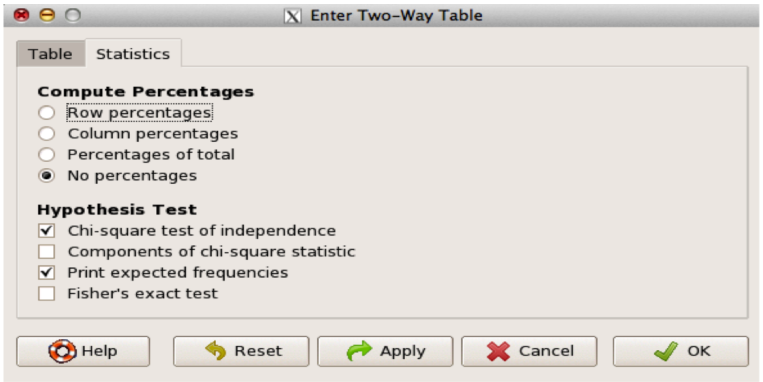
Figure 32: Setting-up a contingency table

`Statistics` → `Contingency table` → `Enter and analyse two-way table`

Input the data in the style of a 2 x 2 table (see Figure 32). In the example before, Treatment and Placebo were placed on rows 1 and 2, whereas No and Yes were placed on the columns 1 and 2. On the statistics tab, there is an option to select the expected frequencies.

```
.Table <- matrix(c(44,40,24,16), 2, 2, byrow=TRUE)
.Test <- chisq.test(.Table, correct=FALSE)
.Test

##
```



Compute Percentages

○ Row percentages
○ Column percentages
○ Percentages of total
● No percentages

Hypothesis Test

☑ Chi-square test of independence
□ Components of chi-square statistic
☑ Print expected frequencies
□ Fisher's exact test

```
##  Pearson's Chi-squared test
##
## data:  .Table
## X-squared = 0.63513, df = 1, p-value = 0.4255
```

```
.Test$expected
```

```
##          [,1]     [,2]
## [1,] 46.06452 37.93548
## [2,] 21.93548 18.06452
```

As we have already seen in our manual calculations, in the section of the output headed **Chi-square**, the probability of observing such a Chi-squared statistic is fairly high: the p-value is 0.43. This is not significant at the 5% level ($p > 0.05$). Hence, there is no evidence of an association between treatment group and tumour shrinkage.

The **chi-squared** test is most suited to large datasets. As a general rule, the chi-squared test is appropriate if **at least 80% of the cells have an expected frequency of 5 or greater**. In addition, none of the cells should have an expected frequency less than 1. If the expected values are very small, categories may be combined (if it makes sense to do so) to create fewer larger categories. Alternatively, Fishers exact test can be used.

7.2 Fisher's exact test

Fisher's exact test can be used in exactly the same way as the Chi-squared test.

Example: Suppose that we use the same example as for the Chi-squared test, but this time we have a **smaller sample size** this time as shown in Table 9

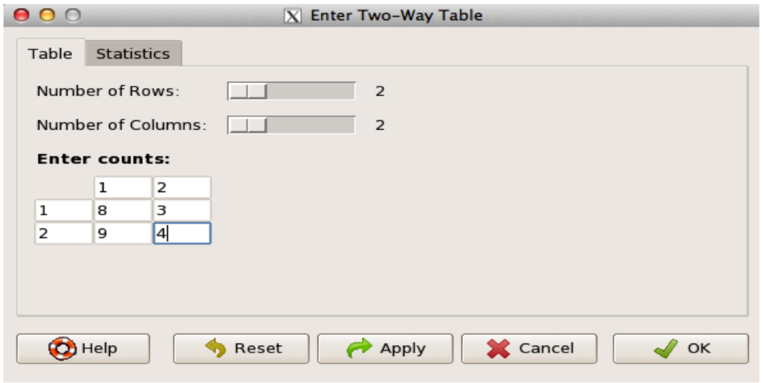
The null and alternative hypotheses are identical to those of the Chi-squared test above.

The **null hypothesis** is that there is **no association** between treatment group and tumour shrinkage.

The **alternative hypothesis** is that there is **some association** between treatment group and tumour shrinkage.

	Tumour Shrinkage		
Treatment group	No	Yes	**Total**
Treatment	8	3	11
Placebo	9	4	13
Total	17	7	24

Table 9: Observed frequencies for chi-squared test



Number of Rows: 2

Number of Columns: 2

Enter counts:

	1	2
1	8	3
2	9	4

In the same way as for the Chi-squared analysis, the expected frequencies can be calculated. These are given in Table 10. Notice that **two of the expected frequencies are less than 5**. As we said earlier if at least 80% of the cells have an expected frequency of 5 or greater, then the Chi-squared test is not appropriate. Instead, a Fisher's exact test can be used.

	Tumour Shrinkage		
Treatment group	No	Yes	**Total**
Treatment	7.79	3.21	11
Placebo	9.21	3.79	13
Total	17	7	24

Table 10: Observed frequencies for chi-squared test

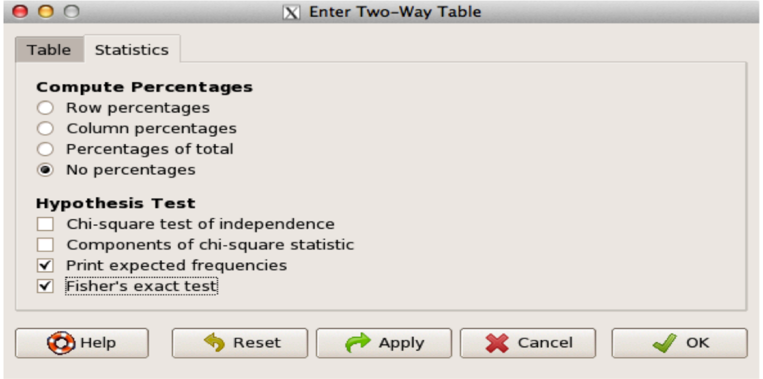
In a new contingency data table input the data in the style of a 2 x 2 table.

On the statistics tab, if we select chi-squared and print expected frequencies then we get

```
.Table <- matrix(c(8,3,9,4), 2, 2, byrow=TRUE)
.Test <- chisq.test(.Table, correct=FALSE)
.Test$expected # Expected Counts

##          [,1]     [,2]
## [1,] 7.791667 3.208333
## [2,] 9.208333 3.791667
```

where we can easily spot the warning regarding low expected frequencies. Therefore, we decide to do a Fishers test instead



Compute Percentages
○ Row percentages
○ Column percentages
○ Percentages of total
● No percentages

Hypothesis Test
□ Chi-square test of independence
□ Components of chi-square statistic
☑ Print expected frequencies
☑ Fisher's exact test

Figure 33: Options for the contingency test

```
fisher.test(.Table)

##
##  Fisher's Exact Test for Count Data
##
## data:  .Table
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##   0.1456912 10.6433317
## sample estimates:
## odds ratio
##   1.176844
```

The output **Fisher's exact test** tells us that the probability of observing such an extreme combination of frequencies is high, our p-value is 1.000 which is clearly greater than 0.05.

In this case, there is no evidence of an association between treatment group and tumour shrinkage.