Introduction to R
9th – 11th April 2014

Exercise 1

1. Start R on the networked computers as described in Section 1.3 of the course manual.

2. Identify and explore the different components of the R GUI.

3. Using internet explorer, navigate to the R-project website (http://www.r-project.org) and explore links that catch your eye. Make sure you find the R manuals page and the user contributed documents section, R search and the R wiki. Download any manuals that you think you might find useful (some are listed in the course manual) and save them on your pendrive or network drive.

4. Use the search facility on the R-project website to search for items related to 'introduction to R' or something similar.

5. Start Tinn-R and initiate a new session. Explore the program, noting where commands are entered and the menu structure. Save this session as 'Aberdeen_R_course' (or something similar) on your pendrive or in a folder on your network drive (H:\ in most cases). This will be your main Tinn-R file for the course to which you will add all your commands as you work through the manual and exercises.

6. Using the R help system, find further information on the functions `mean()` and `median()`. Type your commands in Tinn-R and then copy and paste them in the R console. Make sure you annotate your Tinn-R file with lots of explanatory comments (precede your comments with a #). Hint: use either `help(mean)`, `?mean` or `help("mean")`. Examine the different components of the help file for each function.

7. Get a list of all the functions in R that contains the string '`test`'. Find further information on at least one of these functions.

8. Search the R help system for instances of the string '`plot`'. Narrow your search by only searching for '`plot`' in the `car` package.

9. Start R's html help. Follow the link 'search engines and keywords' and search for the term 'linear model'. Navigate through html help exploring the various links further.

10. Identify which packages are currently installed on your system.

11. Determine the current working directory on your computer.

12. Change the working directory to a folder either on your pendrive or network drive.

Exercise 2

1. Use R to determine the area of a circle with a diameter of 10 cm.

2. Find the natural log, $\log_{10}$, square root and the natural antilog of 12.43

3. Use R to find the cube root of 23 x 0.21

4. Use the function `c()` to create a vector called `weight` containing the weight (in kg) of 10 children: 69, 62, 57, 59, 59, 64, 56, 66, 67, 66. Get R to calculate the mean, variance, standard deviation, range of values and the number of values for `weight`. Extract the values of weight for the first five children.

5. Use the function `scan()` to create a vector called `height` containing the height (in cm) of the same 10 children: 112, 102, 83, 84, 99, 90, 77, 112, 133, 112. Produce a summary of these data. Extract the height of the 2nd, 3rd, 9th and 10th child. Also extract all the heights less than or equal to 99 cm.

6. Calculate the body mass index (BMI) of each child and store the results in a variable called `bmi`. The BMI is calculated as weight (in kg) divided by the square of the height (in meters). Hint: you don't need to do this for each child individually.

7. Create a sequence of numbers ranging from 0 to 1 in steps of 0.1

8. Create a sequence from 10 to 1 in 0.5 steps (Hint: you may find it easier to include the `rev()` function).

9. Generate the following sequences:
   i) 123123123
   ii) 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3
   iii) 1 1 1 1 1 2 2 2 2 3 3 3 4 4 5
   iv) 4 sevens, 3 twos, 1 eight and 5 ones

10. Sort the values of the previously created variable `height` into ascending order and assign the sorted vector to a new variable name (such as `height.sorted`). Now sort all heights into descending order and assign the new vector a name of your choice.

11. Create a new vector called `names` with the following names of the 10 children: Alfred, Barbara, James, Jane, John, Judy, Louise, Mary, Ronald, William. Sort the names of the children into ascending order of height. Who is the tallest?

12. Now order the names of the children by descending values of weight. Who is the heaviest?

13. Generate a factor called `treatment` with 3 levels and with each level repeated 10 times.

14. Save your workspace image and history to a convenient folder on your pendrive or network drive. Also, save your Tinn-R file.

15. List all variables in your workspace. Remove the variable `treatment` from the workspace.

16. Exit R.

Exercise 3

1. Download the data file 'whaledata.xls' from the course website. Open this file in Microsoft Excel and save it as a tab delimited file ('whaledata.txt') to a folder on your pendrive or network drive.

2. Use the function `read.table()` with appropriate arguments to import this file into R and assign it a convenient name (such as `whale`).

3. Attach the dataframe to the R workspace and identify the column names and other attributes.

4. List all values in the variable '`time.at.station`'. What was the time at station for the 59th row of the dataframe?

5. Extract all the elements of the first 10 rows of the dataframe. Also, extract rows 9 to 17 of the first 4 columns.

6. Extract all rows with the following criteria
   i) at depths greater than 1200 m
   ii) gradients steeper than 200 degrees
   iii) water noise level of 'low' and in the month of May
   iv) month of October, water noise level of 'low' and gradient greater than the median value of gradient
   v) all rows **except** the first 10 rows and last column
   vi) all rows that do not have a water noise level of medium

7. Obtain a summary of the dataframe `whale`. Do you notice anything potentially problematic in the summary? If not, don't worry and move onto the next question.

8. Extract all rows of the dataframe `whale` with depths greater than 1500 m and with a greater number of whales spotted than average. Can you see the problem?

9. Create a new dataframe (called `new.whale`) to overcome this problem. And extract rows using the criteria in question 8 (remember to detach the old dataframe and attach the new one).

10. Use the `order()` function to sort all rows in the new dataframe in ascending order of depth. Sort all rows by descending order of depth within each level of water noise.

11. Use the `subset()` function to extract all rows in October with a water noise level of low and a gradient greater than 100.

12. Extract the mean number of whales sighted at the three noise levels. You could also extract the mean number of whales sighted at each noise level on each month. Repeat but calculate standard deviation.

13. Detach the dataframe `new.whales` (or whatever you called it). List the variables in the workspace and then remove them all.

14. Load the dataframe `iris` which is included with R. Use `help("iris")` to find out more information about the dataframe. Open the dataframe in R's data editor. Change a couple of values and one or two variable names in the editor. Exit the editor. Re-open the data editor with `iris`, have the changes been saved?

15. Obtain summary statistics of `Sepal.Length` for each species. Extract all rows with a sepal length less than 5 mm and petal width greater than 0.2.

16. Use R help to look up the function `by()`. Use `by()` to obtain mean values of each column for each species.

17. Export the dataframe `iris` as a tab delimited text file to a folder on your pen or network drive. Include column headings but exclude row names. Import this file into MS Excel.

18. List and remove any unwanted variables in the work space.

19. Create a dataframe of the vectors; `names`, `height`, `weight` and `bmi` that you created in Exercise 2 (refer back to your Tinn-R file for the commands). Call the new dataframe something like `children`.

20. Export the dataframe `children` as a comma delimited csv file to a folder on your pen or network drive. Include column names, but exclude row names.

21. Import the file created in 20 back into R using the `read.csv()` function. Include the childrens names as row names (Hint: use the `row.names=` argument).

22. Copy the dataframe `children` to the clipboard and paste into MS Excel.

23. Save the workspace image if you wish and quit R. Save you Tinn-R script.

Exercise 4

1. Download the data file 'squid1.txt' from the course web site and import it into R. Assign the dataframe an appropriate name (i.e. `squid`) and attach it to the workspace. List the names of the variables contained in the dataframe and produce a summary.

2. Produce a scatter plot of the variable `nid.length` versus index. If there are any obvious oultiers identify them using the `identify (nid.length)` function (use `?identify` for further details). Is the outlier an obvious typo (i.e. a zero in the wrong place)? If so change this value in the dataframe and re-plot the variable. Repeat this for the `dig.weight` variable.

3. Use a scatterplot to plot the variable `DML` against `weight`. Is the relationship linear? If not, use an appropriate transformation to obtain an approximate linear relationship (hint: perhaps try a natural log or square root transformation). Use the `rug()` function to add tick marks on the x and y axes to represent the data points.

4. Produce a histogram of the probability density (not frequency) of each of the following variables: `weight`, `DML`, `dig.weight` and `nid.weight` in the same graphic window (remember to use the `par()` function before you plot the four graphs). Are the breaks sensible for each histogram? If not change them to something more suitable. If you feel like it, add a density curve to each histogram.

5. Use the `boxplot()` function to produce a boxplot of `weight` against `maturity.stage`. Use the `notch=T` argument and also include suitable axes labels and give the graph a title. Save the graph as a pdf.

6. Produce a stripchart of the same variables using an appropriate amount of jitter. Include axes labels and a title. Copy and paste the graph into a Microsoft Word document.

7. Use `pairs()` to investigate the relationships between the variables in columns 8 to 13 of the `squid` dataframe. Add a LOWESS smoother in the lower panel and use the `panel.cor()` function to add the absolute correlations in the upper panel (remember, you can get a copy of the `panel.cor` function from the examples in the `?pairs` help file). Copy and paste the graph into Word.

8. Use a conditional plot to investigate the relationship between `ovary.weight` and `weight` conditioned by `maturity.stage`. Use another coplot to investigate the same relationship, but conditioned by `month` and `year` (hint: make sure `year` and `month` are factors). Copy and paste your graphs into Word.

9. Load the `lattice` package and produce an `xyplot` of `ovary.weight` against `weight` for each `month`. Use the `groups=` arguments to identify data points from each year. Also include a key. Compare the graphs produced using `coplot()` and `xyplot()`.

10. Produce a plot of the first 20 plotting symbols using the `pch` argument in `plot()`. Copy your graph to Word for later reference.

11. Also produce a plot of the first 8 line types using the `lty` argument to `plot()`. Copy your graph to Word.

12. Use the `plot()` function to produce a scatterplot of `DML` on the y axis and `ovary.weight` on the x axis. Use a different symbol and colour for each level of `maturity.stage` (make sure `maturity.stage` is a factor). Produce a legend explaining the different colours and symbols and place it in a suitable position on the plot. Format the graph further to make it suitable for inclusion into your thesis (i.e. add axes labels, change the axes scales etc).

Exercise 5

1. Download the data file 'squid2.txt' from the course web site, import it into R and assign it an appropriate name (`squid2` perhaps).

2. You intend to test whether the average `weight` in the sample is equal to 280 or not. First explore whether `weight` is normally distributed using appropriate plots (hint: histogram, Q-Q plot etc) and perform a normality test.

3. On the basis of the results obtained in 2 decide which test is more appropriate for your original question and run it. Is the average `weight` equal to 280?

4. Test whether `weight` was significantly different in the years 1990 and 1991 using a parametric or a non-parametric approach. Justify your choice of test. Note, to do this you have to create a new data frame which is a subset of the original data set with the rows for year 1989 removed.

5. You aim to explore whether a squid's nidamental gland weight (`nid.weight`) can be predicted from nidamental gland length (`nid.length`). Produce a scatter plot of the data and discuss.

6. Fit a simple linear regression model to these data and explore the results using `summary()`. Produce diagnostic plots and explore these. Is the simple linear model appropriate?

7. Improve the model by also including the square of `nid.length` as an explanatory variable (this is also known as polynomial regression).

8. You suspect that variation in squid `DMS` can be explained using the following explanatory variables; `eviscerate, weight, dig.weight, nid.length, ovary.weight` and `year`. Run a multiple linear regression model for this. Which explanatory variables are significant?

9. Rerun the model leaving the non significant variables out. Assess the appropriateness of both models by considering scatterplots and the diagnostic plots. Is either of these models appropriate? How can we tell which model is better?[1]

[1] Note that the original squid data set was analysed using more advanced modelling methods that we cannot cover here, for details see: Smith JM et al., Seasonal patterns of investment in reproductive and somatic tissues in the squid Loligo forbesi, Aquat. Living Resour. 18, 341–351 (2005). You will be able to understand these methods after next week's course.

Exercise 6

1. Create a function to calculate the area of a circle. Test the function by finding the area of a circle with a diameter of 3.4 cm. Can you use it on a vector of data?

2. Write a function to convert Fahrenheit to Centigrade ($°C = (°F - 32) \times 5/9$). Get your function to print out "Fahrenheit : *value* oF" followed by "Centigrade: *value* oC".

3. Create a vector of normally distributed data, of length 100, mean 35 and standard deviation of 15. Write a function to calculate the mean, median, and range of the vector, print these values out with appropriate labels. Also get the function to plot a histogram (as a proportion) of the values and add a density curve.

4. Write a function to calculate the median value of a vector of numbers. Be careful with vectors of an even sample size, as you will have to take the average of the two central numbers (hint: use modulo %%2 to determine whether vector is an odd or an even size). Test your function on vectors with both odd and even sample sizes.

5. You are a population ecologist for the day and wish to investigate the properties of the Ricker model. The Ricker model is

$$N_{t+1} = N_t \exp\left[r\left(1 - \frac{N_t}{K}\right)\right]$$

   Where N_t is the population size at time t, r is the population growth rate and K is the carrying capacity. Write a function to simulate this model so you can conveniently determine the effect of changing r and the initial population size N_0. K is often set to 100 by default, but you want the option of being able to change this with your function. So, you will need a function with the following arguments; `nzero` which sets the initial population size, `r` which will determine the population growth rate, `time` which sets how long the simulation will run for and `K` which we will initially set to 100 by default.