R tasks

Task 1: Starting with R

Type in anything below that is displayed after the “>” prompt. Check that the output from the computer is as shown on this sheet. Notice those commands that automatically produce a response and those that do not.

```
> 1:10
[1] 1 2 3 4 5 6 7 8 9 10

> x = 1:10
> x
[1] 1 2 3 4 5 6 7 8 9 10

> y = log(x)
> y

[1] 0.00000 0.69315 1.09861 1.38629 1.60944 1.79176 1.94591 2.07944 2.19722
[10] 2.30259

> x*y
[1] 0.0000 1.3863 3.2958 5.5452 8.0472 10.7506 13.6214 16.6355 19.7750
[10] 23.0259

> z = x*y
> z
[1] 0.0000 1.3863 3.2958 5.5452 8.0472 10.7506 13.6214 16.6355 19.7750
[10] 23.0259
> mean(z)
[1] 10.208

> objects()
[1] "x" "y" "z"

> rm(x,y)
> objects()

[1] "z"
```

Notes: (1) Typing > objects() (a built-in R function with, on this occasion, no supplied arguments) produces a list of all user-defined objects (variables, datasets, functions) in the current workspace. Objects may be selectively removed with the function rm.

(2) Previous command lines may be recalled for editing via the up-arrow and down-arrow keys (try this now!). Long calculations may be interrupted via the <Esc> key.

Task 2: Managing data in R

(a) Small datasets may be typed directly into the R workspace. Try the following example:

```
> marks = c(53, 67, 81, 25, 72, 40)
> marks
[1] 53 67 81 25 72 40
```

Try calculating the mean (average) of this data set by using

```
> mean(marks)
```

Write down your answer

(b) Alternatively, the scan command can be used to enter data. This is how it is done:

After the > prompt, type scores = scan( )
The computer will display 1
Then type in the following numbers separated by spaces and press the enter key after the last one 35 62 81 45 24 46

The computer will then display 7:

This is because pressing the "Enter" key takes you to the expected seventh entry. Now press it again to end the process. This is useful if you are entering data from a list separated by spaces as you can "cut and paste"

Find the mean of these data by using

```
> mean(scores)
```

Write down your answer

(c) Data already stored elsewhere may usually be loaded into the workspace via the function data. As an example, type

```
> data(iris)
```

This gives a famous data set and includes the measurements in centimetres of the variables sepal length and width and petal length and width, for 50 flowers from each of 3 species of iris.

Have a look at the iris data by simply typing

```
>iris
```

Task 3: More on nanaging data in R

If you are not reading this from the computer screen, open the Word document as indicated at the top of the first page. Highlight the results below and copy them to the clipboard (in the usual way for "Word" documents).

21836	25024	25272	26428	24582	22279	27714	28188	25878
26744	25625	24656	21273	25229	22998	27002	25537	24560
30324	26198	27693	22346	30615	26717	30461	26102	18039
21553	19178	26977	25247	24234	17226	22204	25100	22474
26059	21376	27752	26362	22287	22948	27321	30340	20631
25600	28056	28451	26625	24146	25726	23419	23428	21483
25201	24551	20989	27719	27340	24785	27161	24806	24355
22414	32831	21795	25445	23190	30351	23673	30650	22688
27681	22541	21764	21590	25199	30487	21403	23341	25937
26418	20672	23413	27226	26867	19974	27986	21216	21525
22766	26224	22339	29391	22437	28401	26561	24033	25310
26458	29859	20497	24749	25212	22573	22808	27109	26916
21648	21107	25042	31242	29735	23867	30918	21281	31371
29890	23354	28355	28239	24198	22758	27670	25507	31248
21298	22259	28200	21183					

Use the scan command to enter them into an R worksheet under the name “salaries”.
Do this by typing the following commands

> salaries=scan ()

The “egg-timer” will appear, but go ahead anyway.

At this point, paste the results into the page. The results will be seen to be going in, then the package will expect another result and will show:

131.

Since there are no further results, now press "Enter" to return to the usual prompt.

To be sure that you have entered the data properly, type
> salaries
This should display the appropriate results.
A histogram can be obtained using the command
> hist(salaries)

(a) Open a new Microsoft Word document, give it the heading “Salaries Data” and, one by one, copy all the following graphs into this document. Don’t worry if you do not know precisely what each graph shows at the moment.

> hist(salaries)
> hist(salaries, br = 7, col="lightblue", border="pink")
> stem(salaries)
> stem(salaries,scale=2)
> boxplot(salaries)

Note: R graphics

A graph may be resized. It may also be printed via the menu obtained by clicking on it with the right mouse button. A graph may be copied and pasted directly into a Microsoft Word document, Excel spreadsheet, or other Windows application. To copy, use the menu obtained by clicking on the right mouse button in the graph (it is probably best to use the Windows metafile format). To paste, e.g. in Word, again use the right mouse button menu. Note the very different final effects between resizing a graph before and after copying and pasting it.

Alternatively, a graph may be saved to an external file in any one of various formats.

(b) Now type mean(salaries) and max(salaries) into the normal R window and then cut and paste the results into the same document as (a).

Task 4: Exploring (linear) dependence

Obtain and attach the data set "cats" by typing the following:

```
>library(MASS)
>data(cats)
>attach(cats)
```

Have a look at the data by typing cats and also use the ?cats command.

Plot Hwt (Heart Weight) against Bwt (Body Weight). Choose Bwt to be on the horizontal axis.

```
>plot(Hwt~Bwt)
```

Obtain the best straight line through the points using

```
>abline(lm(Hwt~Bwt))
```

So far, the analysis has not looked at all at the gender of the cat.

Type in the R code

```
>boxplot(Hwt~Sex, xlab='gender', ylab='heart weight')
```

which clearly shows that heart weight depends on gender.

(use the subcommand horizontal =T if you want a different orientation).

Similarly, the R code

```
> plot(Hwt~Bwt, xlab='body weight (kg)', ylab='heart weight (g)', type='n')
> points(Hwt[Sex=='M']~Bwt[Sex=='M'])
> points(Hwt[Sex=='F']~Bwt[Sex=='F'], pch=3)
```

produces a plot of heart weight against body weight for male cats (denoted by o) and female cats (denoted by +).

Task 5: Distributions in R

A coin is given 10 independent tosses, each of which lands heads with probability a. Let the random variable N denote the total number of heads obtained. Then

$$p(N = x) = \binom{10}{x} a^x (1-a)^{10-x}$$

Given *a* and any value *(or vector of values!)* of *x*, we can calculate this in R as

```
> dbinom(x,10,a)
```

Set a = 0.3 (not a very fair coin!) and verify that the above equation and the R command agree for x=2.
(i.e. type dbinom(2,10,0.3))

Generate the entire probability function of the random variable N with

```
> pf = dbinom(0:10,10,0.3)
```

Use sum to verify that the sum of these probabilities is 1, and plot a bar chart to show their distribution with

```
> barplot(pf,names=0:10)
```

What are the first, second, and third most likely numbers of heads? Calculate the mean and standard deviation of this distribution with

```
> mn = sum(0:10*pf)
> stdev = sqrt(sum((0:10)^2*pf) - mn^2)
```

(to see them, type in mn and stdev after you have done that).

Verify that the values obtained agree with those obtained from your own knowledge of the binomial distribution (i.e. with the standard formulae for mean and stdev of binomial).

Now, suppose that we are unable to do exact calculations with the Binomial distribution. Simulate 1000 realisations of the random variable N with

```
> heads = rbinom(1000,10,0.3)
```

Draw a histogram of the distribution of heads with hist(heads), or better with

```
> hist(heads, breaks=seq(-1,10))
```

(which fixes a slight glitch with plotting histograms of counts).

Compare it with the bar chart of the probability function obtained earlier. Use the functions mean and sd to calculate also the mean and standard deviation of the distribution of *heads*, and compare these with the mean and standard deviation for the theoretical distribution obtained earlier.