MSc in Modern Epidemiology: November 2007

Introduction to Statistics in R

Alex Lewin

Dept of Epidemiology and Public Health, Imperial College

Contents

1 Basic Commands in R

When you open R, you will be faced with a window where you type commands to R to perform calculations, plot data etc.
For example,

```
> 2 + 2
```

```
[1] 4
```

```
> hist(rnorm(1000))
```

The commands above do something (add 2+2 or draw a plot) but otherwise leave no trace. In R, we can also save objects in a workspace, so they can be used again. For example,

```
> x <- rnorm(1000)
```

This defines an object called `x`, which is an array of 1000 numbers generated from a Normal distribution. This object is stored in a file called a workspace, so you can use it again.

```
> hist(x)
```

To see what objects you have in your workspace, type

```
> ls()
```

```
[1] "x"
```

To look at individual objects, use print, or simply type the name of the object:

```
> y <- 2:4
> ls()
```

```
[1] "x" "y"
```

```
> print(y)
```

```
[1] 2 3 4
```

```
> y
```

```
[1] 2 3 4
```

Important: notice that whenever you use a command (eg. `hist` or `print`) you must always use round brackets.

Question 1.1 Vectors and Matrices

Sets of data can be stored as single objects. For example, here is a small set of peoples' heights (in inches), stored in a vector:

```
> heights <- c(56, 65, 59, 57, 74)
```

Two-dimensional data can be stored in a matrix. For example, here is a table displaying cross-tabulated data on bronchitis in infancy and respiratory symptoms in later life:

	bronchitis at 5	No bronchitis at 5
cough at 14	26	44
no cough at 14	247	1002

a) Enter the data from this table (row by row) as a vector called `bronch`, and then do

```
> ls()
```

You should see that you have an object called `bronch` in your workspace. Use the print function to look at it.

b) Type the following command to turn it into a matrix.

```
> bronch <- matrix(bronch, ncol = 2, byrow = TRUE)
> print(bronch)
```

What do the arguments `ncol` and `byrow` mean? (Get help for the function `matrix` by typing `?matrix`. There you will be able to read about `ncol` and `byrow`.)

Another way to make a matrix is using `cbind` to join vectors together. Suppose we have weights as well as heights. The heights and weights can be combined a matrix.

```
> weights <- c(159, 107, 130, 230, 201)
> measures1 <- cbind(heights, weights)
```

c) Make the height and weight vectors, and combine them into a matrix.

To access individual elements of vectors and matrices, use square brackets:

```
> heights[4]

[1] 57

> measures1[4, 1]

[1] 57
```

Question 1.2 Lists

Vectors can also be joined together in a list. This is a much more flexible format than a matrix, as the vectors may be of different length.

a) To combine the weights and heights into a list, do this:

```
> measures2 <- list(heights = heights, weights = weights)
```

To access individual vectors of the matrix, you can either use **double** square brackets, or use the dollar sign and the vector name.

```
> measures2[[1]]
```

```
[1] 56 65 59 57 74
```

```
> measures2$heights
```

```
[1] 56 65 59 57 74
```

b) How can you access just the 4th height in this data set?

Question 1.3 Data Frames

Data frames are probably the most important storage method for data used in epidemiology and public health. A data frame is a special sort of list, one where each vector is the same length, and refers to data taken on the same set of individuals.

A data frame can be made like this:

```
> measures3 <- data.frame(heights, weights)
```

Elements of a data frame can be accessed either using square brackets like a **matrix**, or using the dollar sign like a **list**.

a) Obtain the 3rd weight in the data frame `measures3`, firstly using square brackets, secondly using the dollar sign.

Question 1.4 Data types and missing data

All data looked at so far has been **numerical**.

Sometimes you will need to use **character** data. This is enclosed in quotation marks. For example to add a title "My data" to a plot, you would use

```
> title("My data")
```

You may have a vector of names, eg

```
> days <- c("Mon", "Tues", "Wed")
```

a) Select the 2nd element of the vector `days`.

Very often in epidemiology and public health you will have **categorical** data. For example smoker/non-smoker or country of birth. This type of data is called a **factor** in R. The categories are called **levels**.

This can be entered in R as character, and converted to factor. Here is the smoking data for the people with heights and weights above.

```
> smoke <- c("N", "Y", "Y", "N", "N")
> smoke2 <- as.factor(smoke)
```

b) Look at the two versions of the smoking data. Note that you can see at a glance what levels are in the data (this is useful for large datasets).

c) You can also get the levels without looking at the whole data set (try this):

```
> levels(smoke2)
```

d) A common problem you will encounter is **missing data**. R uses a special code for this: `NA`. For example, suppose we did not know the weight of the 4th person. Instead of a number there would be NA.

Type the following commands

```
> weights <- c(159, 107, 130, NA, 201)
> is.na(weights)
```

What does `is.na` do?

Question 1.5 Functions and Help

Most things that you do in R (eg. plot data, sum some numbers, list the objects in your workspace) you do using a **function**. Most functions have **arguments** (which go inside the round brackets) and a **value**.

For example, in the command

```
> bronch <- matrix(bronch, ncol = 2, byrow = TRUE)
```

Function	matrix
Argument 1	Data to put into `matrix`
Argument 2	No. columns
Argument 3	`byrow`
Value	The resulting matrix object, here called `bronch`

Some arguments are compulsory (eg. here you must give the function some data to put in the matrix). Others are optional (`ncol` and `byrow`). The optional arguments have **default values**.

You can find out all the possible arguments and values with the `help` function, or with a question mark in front of the name of the function you are interested in.

```
> help(matrix)
> ?matrix
```

a) Look at the help for `data.frame`. What are the possible optional arguments?

b) What is the default value for the argument `check.rows`?

Question 1.6 Miscellaneous useful stuff

You can do arithmetic on vectors, eg.

```
> heights + weights

[1] 215 172 189  NA 275
```

The arithmetic operators are

Addition	+
Subtraction	−
Multiplication	*
Division	/
Raise to power	∧ or ∗∗

Another useful construct is

```
> 0:4

[1] 0 1 2 3 4

> 6:8

[1] 6 7 8
```

Here are some useful functions: seq, rep, sum, sort. Use them to answer the rest of the questions in this section. You can look in their help files to find out what they do. Note: at the bottom of each help file you will find some examples of how to use the function.

a) Make a vector called `x` which looks like this:

```
[1] 0 1 2 3 4 5
```

b) Find the sum of the elements in `x`.

c) Make a vector which looks like this:

```
[1] 3 3 3 3 3 3
```

d) Make a vector which looks like this:

```
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

e) Make a vector called `y` which looks like this:

```
[1] 0 1 2 3 4 5 0 1 2 3 4 5
```

f) Sort the values of `y`

Question 1.7 Indexing

It is often useful to extract just a few elements of a vector. For example:

```
> heights[c(2, 4)]

[1] 65 57

> heights[1:3]

[1] 56 65 59
```

You can also select on some condition, eg. values less than a certain threshold:

```
> index <- heights < 58
> heights[index]

[1] 56 57
```

To find elements equal to a certain value, use the `==` symbol.

a) Sum the last 3 elements of the heights vector.

b) Find the weights which are over 140.

c)

```
> smoke <- c("N", "Y", "Y", "N", "N")
```

Find the number of people who are smokers using the `==` symbol.

Saving your work

There are two important ways you need to save your work in R.

The first is to save the workspace (the objects you see when you type `ls()` in R). You can do this by selecting Save Workspace ... in the File menu. This will bring up a window where you will see a file called .RData. This is a hidden file, which contains all the objects you see when you type `ls()`. You can choose which directory to save this in.

Later you can open the workspace by selecting Load Workspace ... in the File menu, or by double clicking on it in Windows Explorer.

The second part of your work you need to save is the commands you are typing into R. The simplest way to do this is to save them in a text file, eg. in WordPad. Save them just as you would type them into R.

Later if you need to repeat commands you can use a function called `source` to read the whole file into R (so all commands in the file will be performed). If you want to perform just some of the commands, either copy and paste them into R, or **comment out** commands you don't want to run (this is done by putting a `#` in front).

The argument to `source` is the name of the file, something like this:

```
> source("mycommands.R")
> source("P:\\Teaching\\MSc-EPH-R-module\\mycommands.R")
> # this command is not run
```

2 Reading and looking at data

Text files - Matrices and Arrays

If your data is all of the same type (eg. a matrix of real numbers), it can be read very quickly using the `scan` function. Usually you just give the file name:

```
> x <- scan("test-data.txt")
> length(x)

[1] 266

> x[1:5]

[1] 115 140 130 103 137
```

Text files - Data Frames

Very often you will use data sets consisting of different types of variables (categorical, numerical etc.) each measured on a set of individuals. This will be presented in a table format (columns of variables, one row per record). This can be read into R using `read.table`.

```
> bp.data <- read.table("BP-reduced-data.txt", header = T, sep = "\t")
```

The argument `header=T` means that the first line of the file is taken to be the column (variable) names. The argument `sep="\t"` means that the columns are separated by tabs (the default is columns separated by any white space).

The object containing the data (here called `bp.data`) is called a **data frame**.

Excel and SPSS files

The best way to get data from these formats into R is to save the file (from Excel or SPSS) as either a text (tab delimitted) file (ending .txt) or a comma separated variable file (ending .csv). Then this file can be read into R using

either `read.table` or `read.csv`.

Using comma separated data and `read.csv` is usually the most reliable. This function is used in the same way as `read.table`.

Some Excel files can be read in using `read.table`. You can try opening a file with a text editor such as Wordpad. If you are able to do this, you should be able to open the file in R with `read.table`.

Question 2.1

The data set used in this session is part of a data set from GlaxoSmithKline, Toronto, Ontario, Canada. The full data set can be found here:
http://www.math.yorku.ca/Who/Faculty/Ng/ssc2003/BPMain.htm

The data set you will use contains 500 observations (subjects) and 6 variables. Of the 500 subjects, 250 had low blood pressure and 250 had high blood pressure (i.e. hypertension). The variables consist of one response variable (systolic blood pressure) and 5 clinical covariates:

sbp	systolic blood pressure
gender	F/M
smoke	N/Y
height	in inches
alcohol	1 = low intake, 2 = medium, 3 = high
bmi	body mass index

a) This data is contained in a file called "BP-reduced-data.txt". Read this data set into R using `read.table`. Call the data object `bp.data`.

When you have read data into a data frame like this, it is useful to check the dimensions (using the R function `dim`) to make sure it has all been read in properly. Do this now - **remember you must use round brackets with functions**.

What are the dimensions of this data set?

b) Another useful function is `names`. Try this on `bp.data`. What does this function do?

c) When you are analysing data from a data frame, you will need to be able to access each variable separately (for all individuals together). You can do this using a dollar sign, eg.

```
> bp.data$height
```

This gives a vector of all the heights in the data set.

How can you access the height of the 412th person?

d) Instead of using the dollar sign, you can use double square brackets:

```
> bp.data[[4]]
```

How can you access the height of the 412th person using square brackets?

In the rest of the module, the dollar sign will be used as I think it is easier. However, you may come across square brackets as well.

e) What are the possible values taken by the variable `smoke` in the BP data set? Use the function `unique`.

f) What is the range of SBP?

g) What is the sum of the first 10 heights in the data set?

Question 2.2

You should have a file called "ONSsmoking.csv". This data will be used in a later practical session. See if you can read this data in using the function `read.csv`.

a) What are the dimensions of this data set?

b) What are the possible values taken by the variable `NumChild`?

3 Plotting data and summary statistics

Useful functions for summary statistics are: `mean, median, sd, var, range, quantile`. They are used in the usual way. Plotting functions are introduced within the questions below.

Question 3.1 Plotting categorical variables

a) Categorical data can be conveniently displayed in tables, eg.

```
> table(bp.data$gender)
```

How many men are there in the study?

How many non-smokers are there in the study?

b) Categorical data can be cross-tabulated using the `table` function as well, for example:

```
> table(bp.data$alcohol, bp.data$smoke)
```

How many male non-smokers are there in the study?

c) You can also plot a bar plot of categorical data:

```
> barplot(table(bp.data$gender))
```

(Notice that you have to use `barplot` on the tabulated data, not on the original vector of data).

Try plotting a barplot of the cross-tabulation of alcohol and smoking. What happens?

Question 3.2 Plotting continuous variables

a) Plot a histogram of the BMI variable from the BP data set (use the function `hist`). Approximately how many people are in the lowest bin used in the histogram?

Now re-do the histogram plot, but use the argument `nclass=20` in the `hist` function. What effect does this have on the plot?

If you want to see both plots at the same time, you can use this command *before* plotting:

```
> par(mfrow = c(1, 2))
```

What does this command do?

b) Give the mean and standard deviation of the BMI.

c) Another way to display the distribution of continuous data is with a box plot. This shows the median, the inter-quartile range and outlying data points.

```
> boxplot(bp.data$height)
```

Estimate the median height by reading from the plot.

Now get the median directly by using the R function median.

d) Use the `quantile` function to obtain the inter-quartile range.

Question 3.3 Plotting continuous v. categorical variables

The R function `boxplot` gives a useful way to quickly compare the distributions of a continuous variable amongst different groups. For instance, if you want to compare the height distributions for men and women, you could do this:

```
> boxplot(bp.data$height ~ bp.data$gender)
```

Make a similar plot comparing heights of the smokers and non-smokers.

It is also possible to obtain a plot containing four separate box plots, for the four combinations of male/female and smoker/non-smoker. See if you can do this. Which category of person is shortest on average?

Question 3.4 Plotting continuous v. continuous variables

a) To compare two continuous variables you can simply use the `plot` function. Try plotting the BMI versus height from the BP data set (height on the x-axis and BMI on the y-axis).

You can add straight lines to plots with `abline`. Use this function to add a 2 lines to your plot: a vertical line at 75 inches, and a horizontal line at BMI=20.

How many people have BMI less than 20 and are taller than 75 inches?

b) Your plot shows just height and BMI, but each point has other data associated with it. One way to look at individual points is to use the identify function. Type the following command in R.

```
> identify(bp.data$height,bp.data$bmi,labels=bp.data$gender)
```

When you have typed this, you will be able to click on the plot, and where you click the nearest point will be labelled with the gender of the person at that point. You can label as many points as you want; when you want to stop, press ESCAPE.

Label some of the points in the different sections of the plot defined by the straight lines. Can you see a pattern in where men and women tend to lie in the plot?

c) You can see the pattern of men and women in the plot all at once, by overplotting some points in different colours. First define an index of the men:

```
> index <- bp.data$gender == "M"
```

Look at this so you understand what this index is. You can then plot just these points in red:

```
> points(bp.data$height[index], bp.data$bmi[index], col = "red",
+     pch = 3)
```

Now plot the points for women in blue.

Question 3.5 Labelling plots

A title can be added to plots with the function `title`. The labels for the x and y-axes can be changed with the arguments `xlab` and `ylab` in the call to plot.

Re-do some of your plots with you own axis labels and add titles.

4 Manipulating Data

You should have a file called "ONSsmoking.csv". This file contains data from a survey by the Office of National Statistics, carried out in November 2005.
From the ONS website: "The ONS Omnibus Survey is a regular, multi-purpose survey. Each month's questionnaire consists of two elements: core questions, covering demographic information, are asked each month together with non-core questions that vary from month to month. The non-core questions for this month was Smoking (Module 130): this module was asked on behalf of the Department of Health."

The data in this file are the answers to the questions in the survey, for 1151 people. In order to get information in a format useful for answering questions, you will have to manipulate the data you read in from the file.

Question 4.1 Workplace restrictions on smoking

One interesting question that can be investigated using this data set is the effect of restrictions on smoking in the workplace. This data set comes from 2005, when there was no general ban on smoking in public places. However, different workplaces had different rules on smoking, so the different levels of restriction can be compared with different levels of smoking.

The table below is a simple way to display this comparison. In this question you will learn how to produce a table like this starting from the survey data.

	Smoking banned	In designated areas	No restrictions
Current smoker	65	73	13
Ex smoker	63	54	11
Never smoked	194	113	19
	322	240	43

a) First read in the data (call the data frame `ONSsmoking`). You can use one of the commands given in section 2. What are the dimensions of the data frame? How many variables are there?

b) The survey did not ask people directly if they were a current smoker, ex-smoker or had never smoked. The following is an extract from the information

about the questions asked in this survey.

```
Pos. = 72   Variable = M130_2   Variable label = Do you smoke cigarettes at all nowadays
This variable is  numeric, the SPSS measurement level is scale.
SPSS user missing values = 8  and 9
    Value label information for M130_2
    Value = 1   Label = Yes
    Value = 2   Label = No
    Value = 8   Label = Refused
    Value = 9   Label = Don't know

Pos. = 76   Variable = M130_6   Variable label = Ever smoked cigarettes regularly?
This variable is  numeric, the SPSS measurement level is scale.
SPSS user missing values = 8  and 9
    Value label information for M130_6
    Value = 1   Label = Yes
    Value = 2   Label = No
    Value = 8   Label = Refused
    Value = 9   Label = Don't know
```

You can look just at these variables by typing

```
> ONSsmoking$M130_2
> ONSsmoking$M130_6
```

What are the possible values taken by these variables? (Use the `unique` function).

c) You will use these two variables to make a new variable called `smoking.status`, which is 1 for a current smoker, 2 for an ex smoker and 3 for someone who has never smoked.

First make a new object in your R workspace called `current.smoke`:

```
> current.smoke <- ONSsmoking$M130_2
> ls()
> print(current.smoke)
```

It is usual to code variables as 1 for yes, 0 for no. So you need to change all the 2's to 0's. First define an index:

```
> index.current.no <- current.smoke == 2
```

What values does this index take?

You can use this index to find all the places where current.smoke has a 2, and turn all those into 0:

```
> current.smoke[index.current.no] <- 0
> print(current.smoke)
```

When you have done this, look at

```
> sum(current.smoke)
```

What does the result of this mean?

d) Go through the same type of process to define a variable called `ever.smoke`, using the variable `M130_6` from the data frame.

e) There is an important extra step needed for this variable: you will have noticed that there are many missing values (NA) in this variable. This is because the question about ever smoking was not asked if the person had already said "yes" to current smoking.

Change all the NA values into 1 (for yes). You will need the `is.na` function.

f) Now you need to make a variable coding 1 for an ex-smoker, 0 for not an ex-smoker. Ex-smokers are people who answered "yes" to ever smoking, but "no" to currently smoking.

```
> ex.smoke <- current.smoke == 0 & ever.smoke == 1
> ex.smoke <- as.numeric(ex.smoke)
```

How many ex smokers are there?

g) In a similar way, make a variable called `never.smoke`, coding 1 for someone who has never smoked, 0 for someone who has smoked at some time.

Look at the first 10 values of `current.smoke`, `ex.smoke` and `never.smoke` (these are the values of these variables for the first 10 people questioned). You should see that each person has a 1 for exactly one of these quantities.

h) At last you can make the variabled called `smoking.status`, which is 1 for a current smoker, 2 for an ex smoker and 3 for someone who has never smoked. Given the 3 variables you have just defined, this is straightforward:

```
> smoke.status <- current.smoke + 2 * ex.smoke + 3 * never.smoke
```

Now type this command:

```
> table(smoke.status)
```

What do the results of this command mean?

i) So now you have the variable on smoking status. To make the table above, you also need the data on restrictions on smoking at work. Luckily this was asked directly in the questionnaire:

```
Pos. = 169  Variable = M130_35  Variable label = What restrictions are there at your wor
This variable is  numeric, the SPSS measurement level is scale.
SPSS user missing values = 8  and 9
    Value label information for M130_35
    Value = 1   Label = No smoking at all on the premises
    Value = 2   Label = Smoking only allowed in designated smoking rooms or areas
    Value = 3   Label = No restrictions at all
    Value = 4   Label = Don t work in a building with other people
    Value = 8   Label = Refused
    Value = 9   Label = Don't know
```

Put this variable into an object called `smoke.restrict`. What are the values that this variable takes?

j) Finally, to obtain the cross-tabulation of smoking status versus workplace smoking restrictions, use the table function again, but on both variables this time:

```
> table(smoke.status, smoke.restrict)
```

How many people are included in this table? (Use the `sum` function, don't count by hand.) Why are there fewer than 1151 people?

If you want to exclude the people who don't work in a building with other people (`smoke.restrict` = 4), you can change the 4's to NA's in `smoke.restrict` and re-do the table.

Question 4.2 Smoking versus age

Now you will look at smoking status for different age categories. The table below shows smoking status versus age group.

	16-24	25-44	45-64	65+
Current smoker	30	135	87	28
Ex smoker	12	62	108	115
Never smoked	59	206	170	139

Here are the details for the age variable:

```
Pos. = 15   Variable = RESPAGE  Variable label = Age of Respondent
This variable is  numeric, the SPSS measurement level is scale.
SPSS user missing values = 998  and 999
    Value label information for RESPAGE
    Value = 998 Label = Refusal
    Value = 999 Label = Don't Know
```

a) Put this variable into an objects called `age`. What is the range of ages in the data?

b) For cross-tabulating the data, it is best to use age groups instead of the actual ages, as in the table shown above. Make a vector of the boundaries of the age groups called `age.breaks`. Note that your boundaries must extend beyond the range of ages.

c) To make the age groups, use the `cut` function.

```
> age.gp <- cut(age, breaks = age.breaks)
```

Look at the first 10 values of `age`, and the first 10 values of `age.gp`. What type of data is `age.gp`?

Note: you will see that the groups are represented by notation something like this: `(15,25]`. The round bracket means the boundary is not included in the group; the square bracket means the boundary is included. So `(15,25]` means the group includes the values 16,...,25. The groups can be changed from `(]` to `[)` by using the argument `right=FALSE` in `cut`.

d) Finally, cross-tabulate age group with smoking status. Your table should look like the one above.

5 Correlation

Correlation is defined as

$$Cor(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X) * Var(Y)}} \tag{1}$$

It measures the strength of assocation between X and Y. It is similar to the Covariance that you learnt about in the Introduction to Statistics module, but it is scaled by the Variances of X and Y, so that the value of the correlation is always between -1 and 1. This enables you to compare correlation coefficients from different data sets.

The sample correlation coefficient is an estimate of the above quantity. It is defined as

$$r(X,Y) = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2 \sum_i (Y_i - \bar{Y})^2}} \tag{2}$$

The sample correlation coefficient also always takes values between -1 and 1.

In this section you will use simulated data to gain an understanding of how the correlation changes for data with differing degrees of association.

The reason for using simulated data rather than real data is that you can control the association in your simulated data.

How to simulate correlated data

In order to investigate the behaviour of the correlation coefficient, you will need to simulate 2 arrays of data, which can be compared to each other.
We will call the 2 variables x and y. To simulate unrelated x and y, you could simulate as before:

```
> x <- rnorm(1000)
> y <- rnorm(1000)
```

But here you need to simulate related variables. An example of a linear relation can be written as:

$$y = 3 + 2x + \epsilon \tag{3}$$

This equation means the y is approximately twice x. The symbol ϵ ('epsilon') stands for the differences between y and $2x$.
To generate data like this (using the Normal distribution), do this:

```
> x <- rnorm(1000)
> y <- rnorm(1000, mean = 3 + 2 * x)
```

The default value for the mean in `rnorm` is zero, so if we didn't specifying the mean for y, there would be no relation between x and y.

Question 5.1

Here you will investigate how the correlation changes depending on the linear relation between x and y.

a) Simulate data of the form

$$y = a + bx + \epsilon \tag{4}$$

for several different values of b (keep a fixed, and keep the number of points generated as 1000). Each time, plot y versus x, and calculate the correlation between them:

```
> plot(x, y)
> cor(x, y)
```

How does the value of the correlation change as you change b? Look at how the plots change too. Include some negative values for b.

b) Now do the same, but keep b fixed and change a.

Question 5.2

The correlation also changes depending on the amount of variability in the data. This is determined by the standard deviation parameter in the Normal distribution (in Question 1 we only changed the mean). To change the standard deviation, use for example

```
> x <- rnorm(1000, sd = 1)
> y <- rnorm(1000, mean = 3 + 2 * x, sd = 5)
```

The default value of sd is 1.

a) Investigate how the correlation changes for different values of standard deviation. First keep sd for x fixed and change the sd for y, then vice versa. (Keep the mean of y the same throughout.)

b) Repeat, but now keep the sd equal for x and y.

It is very important to look at the plots here to understand why the correlation is changing.

Question 5.3

The correlation coefficient is designed to measure association between variables which are linearly related. It is possible to have other relations which are interesting, but not detected by the correlation coefficient. For example, try plotting these variables:

```
> x <- rnorm(1000)
> y <- rnorm(1000, mean = x^2)
```

and calculate their correlation.

6 The Normal Distribution

The Normal distribution is often used in statistics to aproximate the distribution of continuous variables. In this section you will plot histograms of data and see how the shapes compare with the Normal distribution.

The `histogram` function

The function `hist` not only plots the histogram, but also returns several quantities relating to the calculation needed to obtain the plot. To see these quantities, assign the results of the histogram function to an *object*:

```
> h <- hist(bp.data$sbp)
> names(h)
```

```
[1] "breaks"      "counts"      "intensities" "density"     "mids"
[6] "xname"       "equidist"
```

The variables can be accessed individually using the `$` character, for example:

```
> h$breaks

 [1]  60  80 100 120 140 160 180 200 220 240
```

The variable `breaks` contains the limits of the bins used for the histogram. You should be able to match these up with the x-axis of the histogram plot. The variable `counts` contains the number of observations in that bin.

Question 6.1

a) Plot a histogram of the systolic blood pressure (SBP) from the Blood Pressure data set. Use the plot to estimate the proportion of people with SBP greater than 180. Then use the breaks and counts output from the histogram function to calculate this proportion exactly.

b) Now we will try to plot a Normal curve over the histogram of SBP. The density of the Normal distribution is obtained with the function `dnorm`. First have a look at the help for this function. What are the possible arguments to this function?

c) The default `mean` and `sd` parameters used in `dnorm` are 0 and 1. In order to plot a distribution suitable for the SBP, you need to use the mean and sd of SBP. You can do this by

```
> mu.sbp <- mean(bp.data$sbp)
> sig.sbp <- sd(bp.data$sbp)
> plot(bp.data$sbp, dnorm(bp.data$sbp, mean = mu.sbp, sd = sig.sbp))
```

If you compare this plot to your histogram of SBP, you will see that though the x-axes are the same, the y-axes are different. This is because the histogram shows the frequency (counts) in each bin, not the probability. You can make the histogram show probabilities by using the argument `freq=F`.

Now plot the histogram of SBP with the Normal curve superimposed (hint: plot the histogram first, then use `points` in place of `plot` for the Normal curve).

d) The area under the Normal curve can be used to approximate the probability of a given range of SBP. Use the `pnorm` function to calculate the probability of someone having SBP greater than 180 (using the mean and sd calculated in **c)**). You should find this probability is reasonably close to the proportion calculated in **a)**.

e) Plot a histogram of Height from the BP data set, with a suitable Normal curve superimposed. Does this curve look like it would give a good approximation to the empirical distribution?

f) You can directly compare an empirical distribution with a theoretical one using the `qqnorm` function (NOT the `qnorm` function!). The function draws a quantile-quantile plot, which plots the empirical quantiles of the data against the quantiles from a Normal distribution. The plot is close to a 45-degree straight line if the empirical distribution is close to a Normal.

Try this for both the SBP and Height variables from the BP data set.

7 More practice with probability distributions

This section includes some of the questions you did in the "Introduction to Statistics" Module. You will see that using R makes it easier to obtain probabilities than working them out by hand.

R functions for probability distributions

There are 4 functions related to each probability distribution in R. For example, if X is a random variable from a Binomial distribution:

- `dbinom(x)` gives the probability that $X = x$
- `pbinom(q)` gives the cumulative probability that $X <= q$
- `qbinom(p)` gives the quantile q corresponding to cumulative probability p, i.e. gives q where $p = Prob(X <= q)$.
- `rbinom(n)` simulates n Binomial random variables

All probability distributions have 4 functions named similarly (starting with d, p, q and r), eg. `dgeom, dpois, dnorm, dgamma`. For continuous distributions d- gives the probability density function.

Question 7.1 Binomial distribution

It is found that 25% of people exposed to a particular cancerogen become ill with cancer. Among four people with equal exposure to the cancerogen,

a) Show that the probabilities that 0, 1, 2, 3 and 4 children become ill are 81/256, 108/256, 54/256, 12/256 and 1/256 respectively (use the `dbinom` function).

b) What is the probability that at least one child becomes ill? (Use the `pbinom` function.)

Question 7.2 Geometric distribution

Assume that a woman has 49.9 % chance of conceiving a girl in each of her pregnancies. She is very keen on getting a girl.

a) What is the probability that she has to wait till her third pregnancy? Use the `dgeom` function.

b) What is the probability that she does not conceive a girl in her first two pregnancies? Use the `dgeom` or `pgeom` function.

Question 7.3 Poisson distribution

Assume that the number of colds caught by a London inhabitant in the months from November to February is Poisson distributed with parameter 2. What is the probability that the inhabitant will escape having colds altogether in the period November to February? (Use the `ppois` function.)

Question 7.4 Normal distribution

Assume that among diabetics the fasting blood level of glucose is approximately normally distributed with a mean of 105 mg per 100 ml and an SD of 9 mg per 100 ml.

a) Plot the density function for this distribution. You can do this by first defining x as a vector of data in the range of the glucose levels:

```
> x <- seq(60, 150)
```

Do this and print x so you understand what this function does. Then you can use this command to plot the Normal density:

```
> plot(x, dnorm(x, mean = 105, sd = 9), type = "l")
```

What does the argument `type="l"` mean?

b) What proportion of diabetics have levels between 90 and 125 mg per 100 ml? This quantity is represented by the area under the Normal curve between 90 and 125. To get a broad idea, you can mark this range on the plot:

```
> abline(v = 90, lty = 2)
```

Make another line for 125.

c) You can see from your plot that most of the area under the Normal curve is contained between the two vertical lines. Therefore you should expect the proportion of diabetics with levels between 90 and 125 mg per 100 ml to be high.

Use the `pnorm` function to calculate this proportion (hint: you will have to use the function twice).

d) What level cuts off the lower 10 % of diabetics? Here use the `qnorm` function (not `qqnorm`).

Add another vertical line to your plot to mark this 10 % quantile.

8 Hypothesis Testing and Confidence Intervals

In this section you will go through the hypothesis testing questions from the Introduction to Statistics module. You will see that once you have chosen which test to use, it is straightforward to perform the test. The R functions for hypothesis testing can also be used to provide confidence intervals on the quantities of interest.

Good statistical analysis consists of determining an appropriate test to use on a given data set.

Question 8.1 Bronchitis Example

Researchers wanted to know to what extent children with bronchitis in infancy get more respiratory symptoms in later life than others. The following data was collected

cough at 14	bronchitis at 5	No bronchitis at 5	total
yes	26	44	70
no	247	1002	1249
total	273	1046	1319

a) First, you will aim to answer the question: do the proportions of children coughing differ significantly in the two groups (those with bronchitis at five versus those without bronchitis at five)?

You can do this by using the function `prop.test`, which performs a hypothesis test for comparing two proportions.

```
> prop.test(x = c(26, 44), n = c(273, 1046), correct = F)
```

State the null and alternative hypotheses used here.

Look at the output from the test. What is the p-value for the test?

Have a look at the help file for `prop.test`. What quantity is the confidence interval given for?

b) An alternative way to do this is to do the Chi-squared test, using the `chisq.test` function:

```
> obs.table <- matrix(c(26, 44, 247, 1002), ncol = 2)
> chisq.test(obs.table, correct = F)
```

What is the p-value given here?

Question 8.2 Tumour Growth Example (10 patients)

Suppose that the data below refer to paired observations from 10 patients with tumors. Suppose the growth of each patients's tumors are recorded after one month (x), the patients are then given the drug, and the growth of the tumors are recorded again after two months (y). Is the drug effective? Assume the data are Normal.

Control (x)	Treatment (y)
7	4
10	6
9	10
8	8
7	5
6	3
8	10
9	8
12	8
13	10

Here you are comparing two groups of continuous values, so you can use a t-test. Since the data is paired, you must use this form of the test:

```
> x <- c(7, 10, 9, 8, 7, 6, 8, 9, 12, 13)
> y <- c(4, 6, 10, 8, 5, 3, 10, 8, 8, 10)
> t.test(x, y, paired = TRUE)
```

a) What is the value of the t-statistic for this data?

b) What is the p-value for this test? Would you consider that the drug is effective?

c) Give the confidence interval for the mean difference in growth before and after.

Question 8.3 Tumour Growth Example (20 patients)

Consider the data below and assume that the measurements are tunor growth (in mm) of 20 patients with tumors that have been randomly assigned to a treatment group (T) and a control group (C). Assume that the growth in each group is normally distributed. Is the drug effective?

Control (x)	Treatment (y)
7	4
10	6
9	10
8	8
7	5
6	3
8	10
9	8
12	8
13	10

This problem is similar to the previous one, but since now you are comparing two different groups of people, the data is unpaired. Therefore you need the unpaired t-test:

```
> x <- c(7, 10, 9, 8, 7, 6, 8, 9, 12, 13)
> y <- c(4, 6, 10, 8, 5, 3, 10, 8, 8, 10)
> t.test(x, y, paired = FALSE)
```

a) What is the p-value for this test? Would you consider that the drug is effective in this case?

b) Give the confidence interval for the mean difference in growth before and after.

Question 8.4 Ecological Study

In an ecological study comparing 154 cases (Ca) and 178 controls (Co), the investigators notice that 27 cases and 14 controls have been exposed to a toxin in their workplace.

a) Do the percentages in the two groups differ significantly? Perform an appropriate hypothesis test.

b) State the p-value from your test.

c) What is the confidence interval on the difference in proportions?

Question 8.5 Clairvoyance I

A clairvoyant presents you with 4 cards and asks you to remember one. She pretends to be able to know which card you have in your mind. Out of 10 attempts, she guesses the correct card 5 times. Does she have a gift?

a) The null hypothesis is that the clairvoyant guess randomly. What is the probability of her guessing successfully if the null hypothesis is true?

b) What is the alternative hypothesis?

c) Use the function `binom.test` to perform a Binomial test. (Look in the help file to guide you in the syntax.)

d) What is the p-value from your test? Do you think the clairvoyant is gifted?

Question 8.6 Clairvoyance II

More telepathy! A person is asked to think of a number between 1 and 5. The telepathist must then guess this number. The table below gives the results of 1000 attempts by the clairvoyant (1,2,..,5 are the numbers picked and I, II, ..,, V the numbers guessed).

	I	II	III	IV	V	TOTAL
1	54	42	39	44	39	218
2	26	48	39	33	41	187
3	40	32	46	40	38	196
4	38	35	40	49	42	204
5	34	38	39	37	47	195
TOTAL	192	195	203	203	207	1000

a) Test the hypothesis that the clairvoyant does no better than chance, using a Chi-squared test. You can read in the data from the file "Clairvoyance2.txt", using the `scan` function (then convert the read data into a matrix).

b) What is the Chi-squared statistic from this test? Does the test give evidence of telepathy?

Question 8.7 Workplace smoking restrictions

Recall the data on workplace smoking restrictions from Section 4.

	Smoking banned	In designated areas	No restrictions
Current smoker	65	73	13
Ex smoker	63	54	11
Never smoked	194	113	19
	322	240	43

You can now perform a suitable hypothesis test on this data, to find out if there is a statistically significant association of smoking status with restrictions on workplace smoking.

9 Extended questions

Question 9.1 Smoking survey data

The ONS survey also included questions on attitudes to increasing restrictions on smoking in general. In particular, there was a question on how far the respondent agreed with smoking restrictions at work:

```
Pos. = 161  Variable = M130_34a Variable label = Agree to smoking restrictions at work
This variable is  numeric, the SPSS measurement level is scale.
SPSS user missing values = 8  and 9
    Value label information for M130_34a
    Value = 1   Label = Agree strongly
    Value = 2   Label = Agree
    Value = 3   Label = Neither agree nor disagree/don t mind
    Value = 4   Label = Disagree
    Value = 5   Label = Disagree strongly
    Value = 8   Label = Refused
    Value = 9   Label = Don't know
```

a) Cross-tabulate this variable with the variable on smoking status you made in Section 4.

b) Suppose you are interested in whether extent to which people agree with restrictions on smoking depends on their own smoking status. State a suitable null hypothesis to test.

c) Carry out a Chi-squared test on this data. In order for the approximation used in the Chi-squared test to be valid, you will have to group some of the categories together.

What conclusion would you draw from this test?

d) The survey includes more detailed information for some respondents, on how many cigarettes are typically smoked per day.

```
Pos. = 74   Variable = M130_4   Variable label = How many cigarettes a day on weekdays?
This variable is  numeric, the SPSS measurement level is scale.
SPSS user missing values = 998 and 999
    Value label information for M130_4
    Value = 998 Label = Refused
    Value = 999 Label = Don't know
```

Look at the distribution of this variable. There are peaks in the distribution at some values. Why is this?

e) Think of a grouping for the number of cigarettes per day. Plot a barplot to look at the distribution of the grouped variable. Does this plot show a smooth

distribution, with one peak, or are there still more than one large peak? You may have to group the data even further.

f) When you are happy with the grouped variable for number of cigarettes per day, perform a suitable hypothesis test to investigate possible association between number of cigarettes per day and agreement with restrictions on smoking in the workplace. You may find you have to group your data even further.

What conclusion do you draw from this test?

Question 9.2 Blood pressure data

In this question you will compare the systolic blood pressure (SBP) for smokers and non-smokers.

a) First plot histograms of SBP separately for smokers and non-smokers. Do you think the SBP is on average different for smokers and non-smokers?

b) Try separate boxplots of SBP for smokers and non-smokers. What do you think now?

c) You can now perform a t-test to see if there is a significant difference in SBP for smokers and non-smokers. In order for this test to be valid, the SBP must be reasonably close to a Normal distribution. Do you think it is?

d) Carry out the t-test. What is the null hypothesis?

What conclusion do you draw from this test?

e) Carry out a t-test comparing SBP for men and women. What conclusion do you draw?

Useful functions

Reading data

read.table
read.csv
scan
matrix
cbind
list
data.frame

Looking at data

ls
print
dim
length
names
unique
sort
sum
range
table
levels
is.na

Plotting

plot
points
lines
abline
hist
boxplot
barplot
table
identify
title

Summary statistics

mean
median
sd
var
range
quantile
cor

Miscellaneous

cut
seq
rep

Probability distributions

rnorm, dnorm, pnorm, qnorm
rbinom, dbinom, pbinom, qbinom
rpois, dpois, ppois, qpois

Hypothesis tests

chisq.test
prop.test
binom.test
t.test

References

Reference for ONS data:
Office for National Statistics. Social Survey Division, ONS Omnibus Survey, Smoking Module, October and November, 2005 [computer file]. Colchester, Essex: UK Data Archive [distributor], September 2007. SN: 5707.

Useful reference book on R:
Introductory Statistics with R, Peter Dalgaard, Springer.

Website on R:
http://cran.r-project.org/ → Manuals → An Introduction to R

Acknowledgements

The examples in Sections 7 and 8 were given by Sylvia Richardson, with modifications by Anne-Mette Hein and Natalia Bochkina.