Chapter 2

Summarising and plotting data

Contents

2.1 Introduction

We start with a data set, in the form of spreadsheet consisting of a fairly long list of numbers. In this chapter, we consider the following:

1. how to **summarise** the data: how to represent the data with a small set of values that describe the main features;

2. how to plot the data, in a way that makes it easy to identify and understand the information represented by the data;

3. how to do the above using R.

Summarising and plotting data to identify the main/interesting features is sometimes referred to as **exploratory data analysis**. In practice, it's often best to start a data analysis with suitable plots, as you can learn things about your data quickly this way, and plots usually provide the most information. However, some plots use numerical summaries of the data referred to in (1), so we will study summarising data numerically first. In the second case study presented in this chapter, where there are large number of potential plots that could be produced, we will see how calculating numerical summaries can help identify *which* plots will be most useful.

2.2 Case study: life expectancy in 181 countries

The file `life-expectancy.csv` (available on MOLE) contains data on life expectancy for 181 countries, produced by the World Health Organisation (WHO)[1].

How life expectancy has been estimated

In a given year, mortality rates are estimated for each age group (e.g. out of N 60-year-olds alive at the start of 2015, d died in that year.) Life expectancy for a given year is reported as the expected lifetime of an individual born in that year, assuming the mortality rates observed in that year do not change over the individual's lifetime. An average is reported for males and females, and the data set contain life expectancies calculated in 2000 and 2015 (columns `life2000` and `life2015` in the spreadsheet).

Regional groupings

Each country is classified under one of six regional groupings used by the WHO: Africa, Americas, South-East Asia, Europe, Eastern Mediterranean, and Western Pacific. A map of the regional groupings is available at http://www.who.int/about/regions/en/.

Objectives for the data analysis

Looking directly at the data (e.g. by opening the .csv file in Excel), we can see life expectancy is different in different countries, and has changed over time, but it's hard to make sense of the data just from looking at all the numbers otherwise. Some things we might like to know are:

[1] http://apps.who.int/gho/data/node.main.688, accessed 22/11/16

1. What was a 'typical' life expectancy in 2015?
2. How much variation is there in life expectancy between countries?
3. Was there an 'overall' change in life expectancy from 2000 to 2015?
4. Do any countries stand out as having relatively high or low life expectancies?
5. Is life expectancy 'generally' worse/better in some regions than in others?

In the following sections, we consider informal or **exploratory** methods we can use to answer these questions.

2.3 Importing data into R: `csv` and `.xlsx` files

The data are in "comma separated variables (csv)" format, so we can use the command `read.csv` to get the data into R. (The file `life-expectancy.csv` will need to be in your working directory. You can change the working directory in RStudio by going to `Session` > `Set Working Directory`).

```
lifeExp <- read.csv("life-expectancy.csv")
```

The data are now stored in R in an object called `lifeExp`. All commands and names in R are case-sensitive: R won't recognise the name `lifeexp`.

Importing Excel `.xlsx` files

If your data is an Excel spreadsheet in `.xlsx` format, you can either save it in Excel as a `.csv` file, or you can use the `readxl` package. If you have not done so already, you will first need to install the `readxl` package.

```
install.packages("readxl")
```

Then, for example, to import a file `spreadsheet.xlsx`, you would use the commands

```
library(readxl)
mydata <- read_excel("spreadsheet.xlsx")
```

2.4 Data frames in R

`lifeExp` is a type of "object" known as a `data frame`. Data frames are the main way of working with data sets in R. The data are arranged in the data frame as they were in the .csv file, with one row per country, and one column per variable. Typing `lifeExp` (and pressing return) in the R console will display the entire data set. The display may not be helpful if the data set is large, so instead, to see the first few rows of the data frame only, use the `head` command:

```
head(lifeExp)
```

```
##                Country                 Region life2000 life2015
## 1          Afghanistan Eastern Mediterranean     54.8     60.5
## 2              Albania                 Europe     72.6     77.8
## 3              Algeria                 Africa     71.3     75.6
## 4               Angola                 Africa     45.3     52.4
## 5 Antigua and Barbuda               Americas     73.6     76.4
## 6            Argentina               Americas     74.1     76.3
```

Extracting columns/observations from data frames

R won't yet recognise the column names. For example, if we wanted to extract the 2015 life expectancy for Albania (element 2 of the column called `life2015`), we might try

```
life2015[2]
```

```
## Error in eval(expr, envir, enclos):  object 'life2015' not found
```

but we just get an error message. We can either use the $ operator (where the syntax is *data-frame-name$column-name*)

```
lifeExp$life2015[2]
```

```
## [1] 77.8
```

or the `attach` command:

```
attach(lifeExp)
life2015[2]
```

```
## [1] 77.8
```

You only need to use the `attach` command once per session, so this can be convenient if you want to do lots of operations in R with the `life2015` variable.

Extracting rows from data frames

You can extract a subset of rows from a dataframe in various ways. For example, to extract row number 100:

```
lifeExp[100, ]
```

```
##      Country          Region life2000 life2015
## 100 Malaysia Western Pacific     72.4       75
```

To extract row numbers 51-60:

```
lifeExp[51:60, ]
```

```
##              Country                Region life2000 life2015
## 51             Egypt Eastern Mediterranean     68.8     70.9
## 52       El Salvador              Americas     69.0     73.5
## 53 Equatorial Guinea                Africa     52.7     58.2
## 54           Eritrea                Africa     45.3     64.7
## 55           Estonia                Europe     70.8     77.6
## 56          Ethiopia                Africa     51.2     64.8
## 57              Fiji       Western Pacific     67.7     69.9
## 58           Finland                Europe     77.5     81.1
## 59            France                Europe     78.8     82.4
## 60             Gabon                Africa     60.1     66.0
```

To extract rows corresponding to Western Pacific countries only:

```
lifeExp[Region=="Western Pacific", ]
```

```
##                                Country          Region life2000 life2015
## 8                            Australia Western Pacific     79.5     82.8
## 24                  Brunei Darussalam Western Pacific     74.4     77.7
## 29                           Cambodia Western Pacific     57.7     68.7
## 35                              China Western Pacific     71.7     76.1
## 57                               Fiji Western Pacific     67.7     69.9
## 83                              Japan Western Pacific     81.1     83.7
## 87                           Kiribati Western Pacific     64.1     66.3
## 90  Lao People's Democratic Republic Western Pacific     58.1     65.7
## 100                          Malaysia Western Pacific     72.4     75.0
## 107 Micronesia (Federated States of) Western Pacific     67.0     69.4
## 108                          Mongolia Western Pacific     62.8     68.8
## 116                       New Zealand Western Pacific     78.6     81.6
## 124                  Papua New Guinea Western Pacific     58.9     62.9
## 127                       Philippines Western Pacific     66.8     68.5
## 131                  Republic of Korea Western Pacific     76.0     82.3
## 138                             Samoa Western Pacific     70.2     74.0
## 145                         Singapore Western Pacific     78.3     83.1
## 148                   Solomon Islands Western Pacific     65.8     69.2
## 164                             Tonga Western Pacific     71.6     73.5
## 176                           Vanuatu Western Pacific     69.0     72.0
## 178                          Viet Nam Western Pacific     73.4     76.0
```

2.5 Data terminology

2.5.1 Quantitative and qualitative variables

- "Life expectancy" for 2000 and 2015 are **quantitative** variables: a 'quantity of something' that can either be a continuous or a discrete numerical value. A quantitative variable

may be measured on a **ratio** scale or an **interval** scale:

  - "life expectancy" is measured on a **ratio** scale: both ratios and differences are meaningful, and zero represents "no life expectancy". The difference between 50 years and 40 years is the same as the difference between 60 years and 50 years, and 40 years is "twice as much" life expectancy as 20 years.
  - "time of day" on a 12-hour clock is measured on an **interval** scale: differences are meaningful, but ratios are not, and zero does not represent "no time of day". The differences between 4pm and 3pm, and 8pm and 7pm represent the same elapsed time, but 2pm does not represent "twice as much time of day" as 1pm.

- "Region" is a **qualitative** variable: a non-numerical description of something. We sometimes use the term **factor** variable. Somewhat confusingly, factor/qualitative variables may be represented numerically in a data set, e.g., in a medical data set: "0" for "non-smoker and "1" for "smoker", but the choices of 0 and 1 are arbitrary, and have no particular relevance in this context. Qualitative variables can be **nominal** or **ordinal**.
  - **ordinal** variables can be ordered in some sense. For example, smoking status could be classified under the **levels** 'non-smoker', 'light smoker', 'heavy smoker', with each level representing increasing use of cigarettes.
  - **nominal** variables have no ordering. "Region" is a nominal variable: there is no meaningful order to the WHO's six regional groupings.

2.5.2 Univariate and multivariate variables

- Life expectancy in 2015 is a **univariate** variable: there is only one number per country;
- Overall, we have **multivariate** data on life expectancy: for each country, we have more than one value (life expectancy in 2000 and 2015). (The case of two values is referred to as **bivariate**)

2.6 Numerical summaries of univariate data

We start by analysing the 2015 data. Define y_i to be the life expectancy in country i, in years, for $i = 1, \ldots, N$ with $N = 181$. The first three values are

```
life2015[1:3]

## [1] 60.5 77.8 75.6
```

So we have $y_1 = 60.5$, $y_2 = 77.8$, $y_3 = 75.6$ and so on.

2.6.1 Mean and variance

To describe a typical value, a **measure of central tendency**, we could use the **mean**, which we define as

$$\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i.$$

To describe variation in the data, we can use the variance, which we define as

$$s_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2.$$

As this is in squared units, it can be helpful to consider the standard deviation, which we define as

$$s_y = \sqrt{s_y^2}.$$

We can calculate these in R as follows:

```
mean(life2015)

## [1] 71.20829

var(life2015)

## [1] 63.72499

sd(life2015)

## [1] 7.982793
```

and so we have (to 2 d.p.)

$$\bar{y} = 71.21, \qquad s_y^2 = 63.72, \qquad s_y = 7.98.$$

Hence, starting with 181 values, we can describe some important features of the data with just two numbers: the mean, to indicate a 'typical' life expectancy, and the standard deviation, to indicate how the life expectancies for the different countries vary around the mean.

2.6.2 Quantiles/percentiles

We have used the mean to describe a 'typical' value. Another choice would be the median: a value such that half the observations are below, and half are above. The median is the 0.5 quantile, or 50th percentile. More generally, the α quantile, or $100 \times \alpha$ percentile is, informally, a value such that, approximately, $100 \times \alpha\%$ of the observations are below the value, with the remainder above. In R, we use the `quantile` command. (We can also use the `median` command to get the median).

```
median(life2015)

## [1] 73.3

quantile(life2015, probs = c(0.025, 0.5, 0.975))

##  2.5%   50% 97.5%
## 53.50 73.30 82.75
```

Hence:

- the median/50th percentile/0.5 quantile is 73.3 years: about half the countries have life expectancies below 73.3 years and half above;
- the 2.5th percentile/0.025 quantile is 53.5 years: about 2.5% of the countries have life expectancies below 53.5 years and 97.5% above;
- the 97.5th percentile/0.975 quantile is 82.75 years: about 97.5% of the countries have life expectancies below 82.75 years and 2.5% above;
- about 95% of the countries have life expectancies between 53.5 years and 82.75 years.

Note that the median is slightly larger than the mean, for reasons that we'll see when we plot the data later on.

Exact calculation of quantiles

The precise calculation of a quantile is a little complicated, and different software packages implement minor variations. For reference, the method used in R as follows (you will not be expected to calculate quantiles by hand, so you won't need to implement this procedure for yourself).
For illustration, we calculate the $p = 0.3$ quantile of the $n = 9$ values 29, 13, 92, 7, 31, 51, 44, 80, 13, defined in a vector `x`. Denote this quantile by $x_{0.3}$.

```
x <- c(29, 13, 92, 7, 31, 51, 44, 80, 13)
```

1. Arrange the values in increasing order

   ```
   (x.ord <- sort(x))

   ## [1]  7 13 13 29 31 44 51 80 92
   ```

   The ordered values are sometimes referred to as the **order statistics**: the first order statistic $x_{(1)}$ is 7, the second and third order statistics $x_{(2)}$ and $x_{(3)}$ are both 13, the fourth order statistic $x_{(4)}$ is 29 and so on.

2. The jth order statistic gives the exact pth quantile for $p = (j-1)/(n-1)$

j	1	2	3	4	5	6	7	8	9
$x_{(j)}$	7	13	13	29	31	44	51	80	92
quantile	0	$\frac{1}{8}$	$\frac{2}{8}$	$\frac{3}{8}$	$\frac{4}{8}$	$\frac{5}{8}$	$\frac{6}{8}$	$\frac{7}{8}$	1

   So the median is 31, the 0.25 quantile is 13 and so on.

3. For any other quantile, we interpolate linearly between the order statistics, using the two closest quantiles that we have. So, to get the 0.3 quantile, we interpolate linearly between the points

   $$(0.25, 13) \text{ and } (0.375, 29)$$

   This is illustrated in Figure 2.1.

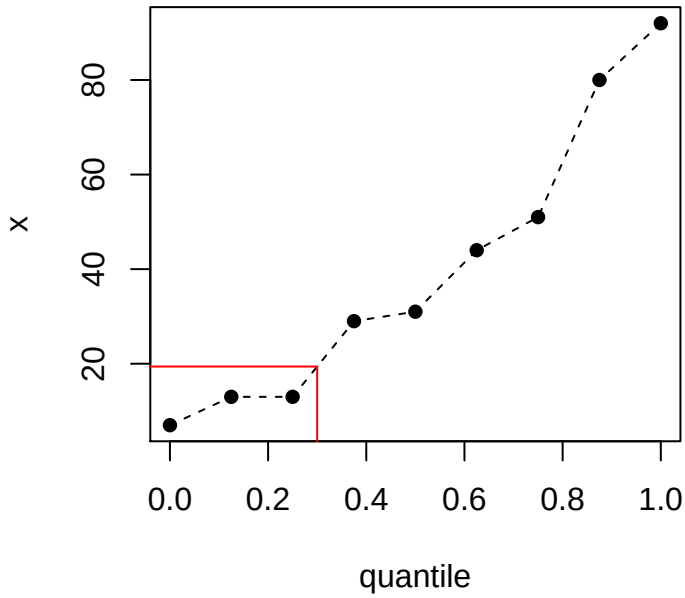


Figure 2.1: To obtain the 0.3 quantile, we interpolate linearly between the two closest available quantiles (the 0.25 and 0.375 quantiles).

To determine the point $(0.3, x_{0.3})$. We have

$$x_{0.3} = 13 + \frac{0.3 - 0.25}{0.375 - 0.25} \times (29 - 13) = 19.4 \tag{2.1}$$

To verify that R gives the same result directly

```
quantile(x, 0.3)

##  30%
## 19.4
```

2.6.3 Inter-quartile range

The 0.25, 0.5 and 0.75 quantiles are also known as the quartiles, and the difference between the 0.75 and 0.25 quantiles is known as the inter-quartile range (IQR). Hence the IQR gives a 'central' interval containing (approximately) 50% of the observations. We can obtain quantiles in R as follows.

```
quantile(life2015, probs = c(0.25, 0.75))

##  25%  75%
## 65.7 76.7

IQR(life2015)

## [1] 11
```

2.6.4 The `summary` command in R

We can also use the `summary` command to obtain the quartiles and mean (and minimum and maximum values):

```
summary(life2015)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   50.10   65.70   73.30   71.21   76.70   83.70
```

The gap between minimum and maximum is quite striking here (but note that not all countries were included in the 2015 data). Which countries are these? We can extract the rows as follows.

```
which.min(life2015)

## [1] 144

lifeExp[144, ]

##          Country Region life2000 life2015
## 144 Sierra Leone Africa       39     50.1
```

or more concisely

```
lifeExp[which.max(life2015), ]

##    Country          Region life2000 life2015
## 83   Japan Western Pacific     81.1     83.7
```

Summarising data by group

Having defined these various summaries of the data, we can calculate them separately for countries within each region, and hence compare life expectancies between the regions. The following command tells R to apply the function `summary` separately to each group of `life2015` values, where the groups are defined by `Region`:

```
by(life2015, Region, summary)

## Region: Africa
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   50.10   58.25   61.40   61.55   64.75   75.60
## ------------------------------------------------------------
## Region: Americas
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   63.50   73.50   75.00   74.74   76.40   82.20
## ------------------------------------------------------------
```

```
## Region: Eastern Mediterranean
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   55.00   65.70   74.10   70.68   75.30   78.20
## ------------------------------------------------------------
## Region: Europe
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   66.30   74.47   77.70   77.42   81.42   83.40
## ------------------------------------------------------------
## Region: South-East Asia
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   66.60   68.70   69.80   71.09   73.35   78.50
## ------------------------------------------------------------
## Region: Western Pacific
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   62.90   68.80   73.50   73.68   77.70   83.70
```

We can look at the values in the output, but there are further R commands we can use to make inspection of the data a little easier. For example, if we wanted to compare mean life expectancies in the six regions, we could first calculate means for each region (assigning to a variable `meanExp`)

```
meanExp <- by(life2015, Region, mean)
```

then, for example, find the regions with the smallest and largest mean life expectancies:

```
which.min(meanExp)

## Africa
##      1

which.max(meanExp)

## Europe
##      4
```

or make a simple table with the mean life expectancies arranged in ascending order, using the `sort` command to arrange the values, and the `cbind` command to display the values in a column format:

```
cbind(sort(meanExp))

##                             [,1]
## Africa                  61.55106
## Eastern Mediterranean   70.68095
## South-East Asia         71.09091
## Western Pacific         73.67619
## Americas                74.73636
## Europe                  77.41667
```

The precision looks rather spurious here, so we could choose to round the values to three significant figures:

```
signif(cbind(sort(meanExp)), 3)
```

```
##                          [,1]
## Africa                 61.6
## Eastern Mediterranean 70.7
## South-East Asia        71.1
## Western Pacific        73.7
## Americas               74.7
## Europe                 77.4
```

This shows the differences between the regions clearly. (The results are not surprising, but the gap between Africa and Europe is still striking, nevertheless.)

2.7 Numerical summaries of bivariate data: covariance and correlation

With bivariate quantitative variables, we may be interested in relationships between the variables, for example, whether it's possible to predict the value of one variable given the other. There are more complex methods for investigating this, but a starting point is to consider their **correlation**, which is defined via **covariance**.

2.7.1 Covariance

If we define z_i to be the life expectancy in 2000 of country i, we can define the covariance between life expectancy in 2000 and 2015 to be

$$s_{yz} = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})(z_i - \bar{z}).$$

In R, we use the command `cov`:

```
cov(life2000, life2015)
```

```
## [1] 77.35256
```

and so $s_{yz} = 77.35$ to 2 d.p.

2.7.2 Pearson's correlation coefficient

Covariances aren't very informative on their own, as they will depend on the scale of measurement of the variables. **Correlation coefficients** are scale independent. There are different versions of the correlation coefficient. **Pearson's** correlation coefficient is defined to be

$$r_{yz} = \frac{s_{yz}}{s_y s_z},$$

with

$$s_y = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2}, \qquad s_z = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(z_i - \bar{z})^2} \tag{2.2}$$

In R, we use the command `cor`

```
cor(life2000, life2015)

## [1] 0.9506997
```

and so $r_{xy} = 0.95$ to 2 d.p.
Pearson's correlation coefficient measures the strength of the *linear* association between the two variables, and is bounded between -1 and 1. A positive correlation implies that as one quantity increases, the other is expected to increase, and a negative correlation implies that as one quantity increases, the other is expected to *decrease*. A correlation of 0.95 is very high, and tells us that if we know the life expectancy in 2000, we should be able to predict the life expectancy in 2015 fairly accurately, and vice versa.

Scale independence of correlation coefficients

To illustrate the point of scale independence, consider measuring life expectancy in months, rather than years. This will change the covariance, but not the correlation, as we see in the following.

```
life2015months <- 12 * life2015
life2000months <- 12 * life2000
cov(life2015months, life2000months)

## [1] 11138.77

cor(life2015months, life2000months)

## [1] 0.9506997
```

2.7.3 Spearman's correlation coefficient

An alternative to Pearson's correlation coefficient is **Spearman's** correlation coefficient. Spearman's correlation coefficient is defined to be **Pearson's** correlation coefficient calculated on the **ranks** of the variables.
For example, suppose we have the following data

i	1	2	3	4	5	6
y_i	68	2	40	20	85	97
z_i	73	26	37	1	63	68

We first calculate the ranks of the observations:

i	1	2	3	4	5	6
rank(y_i)	4	1	3	2	5	6
rank(z_i)	6	2	3	1	4	5

We then calculate Pearson's correlation coefficient on the ranks, as if we have six bivariate observations (4, 6), (1, 2), ... (6, 5). In R, we just include an extra argument in the `cor` command:

```
y <- c(68, 2, 40, 20, 85, 97)
z <- c(73, 26, 37, 1, 63, 68)
cor(y, z, method = 'spearman')

## [1] 0.7714286
```

If we don't specify a `method`, the default is to use Pearson's. We can obtain the rankings in R with the command `rank`. Compare the above with

```
rank(y)

## [1] 4 1 3 2 5 6

rank(z)

## [1] 6 2 3 1 4 5

cor(rank(y), rank(z))

## [1] 0.7714286
```

To help interpret and visualise correlation coefficients, four examples are plotted in Figure 2.2.

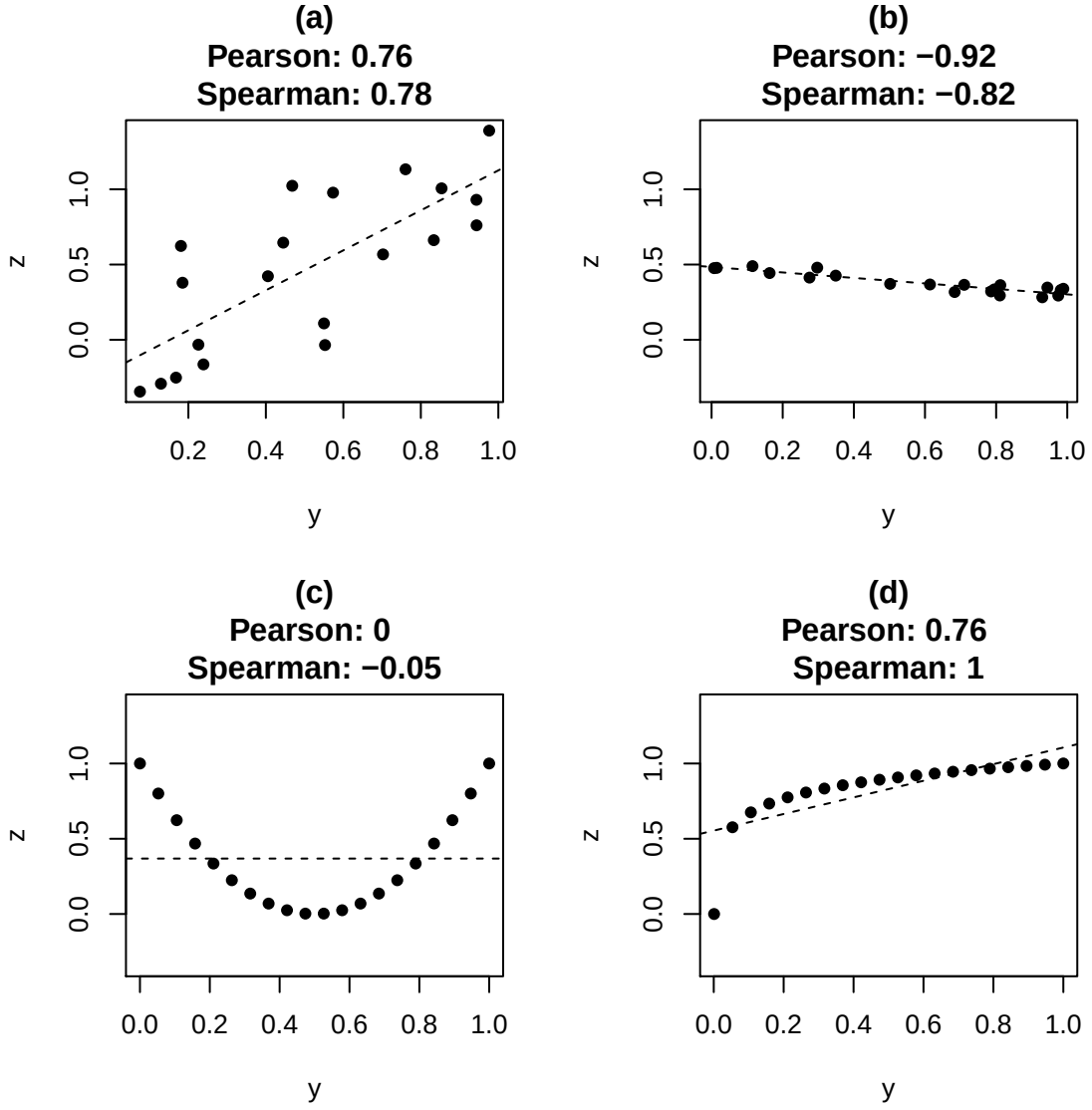


Figure 2.2: Correlation coefficients and linear trends. Comparing (a) and (b), the (absolute value of the) gradient of the linear trend is larger in (a), but the correlation is stronger in (b): it is easier to predict z given y in this case. In (c), y and z are clearly related, but the Pearson correlation is 0: there is no *linear* trend. In (d), the relationship between z and y is monotone, but nonlinear, and the Spearman correlation is greater than Pearson's.

In practice, there is little value in calculating a correlation coefficient if it is easy to plot the data: the picture will tell the story better than the number. However, if we are working with a large data set with many quantitative variables, the correlation coefficient gives a method (though not an infallible one) of automatically detecting related variables.

2.8 Plotting data and distributions

One of the most informative ways to display univariate data visually is to plot its **distribution**. By this, we mean displaying the relative frequencies with which different values have occurred, or the relative frequencies with which different (non-overlapping) intervals of values have occurred.

2.8.1 Plotting a univariate distribution with a histogram

A set of univariate observations is often best plotted using a histogram. A histogram is a bar chart, where the area of each bar is proportional to the number of observations lying in the interval indicated by the bar. Each interval is known as a **bin**. If the bars are all of equal width (which is recommended), then the height of the bar is normally either the number of observations in the corresponding bin, or the proportion of the total number that lie in that bin.
Use the command `hist` to draw a histogram.

```
hist(life2015, xlab = "life expectancy in 2015 (years)",
     main = "histogram of life expectancy \n in  181 countries ")
```

Figure 2.3: A histogram clearly displays the main features of the data. We can see the range of values that have occured, and in what relative frequency.

The bin boundaries and counts can be obtained as follows.

```
h <- hist(life2015, plot = FALSE)
h$breaks

## [1] 50 55 60 65 70 75 80 85

h$counts

## [1]  8 14 21 28 46 36 28
```

So the bins are [50, 55], (55, 60], ..., (80, 85] and the counts in each bin are 8, 14, ..., 28, so 8 countries had life expectancies in the interval [50, 55] years, 14 countries had life expectancies in the interval (55, 60] years and so on.

The main choice we have to make when plotting a histogram is what bin width to use. In R, this can be done by specifying the endpoints of the bins with the argument `breaks`.

```
par(mfrow = c(2,1))
hist(life2015, xlab = "life expectancy in 2015 (years)",
     main = "(a) histogram of life expectancy \n in 181 countries ",
     breaks = c(45, 65, 85))
hist(life2015, xlab = "life expectancy in 2015 (years)",
     main = "(b) histogram of life expectancy \n in 181 countries ",
     breaks = seq(from  = 50, to = 85, by = 0.1))
```

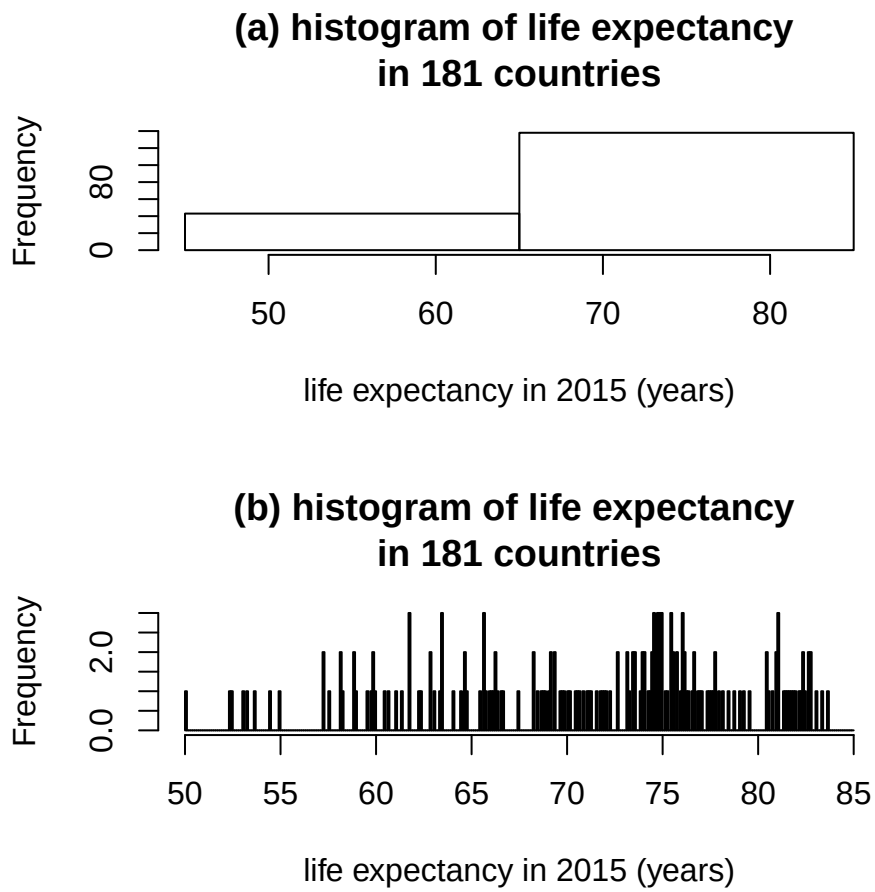


Figure 2.4: In plot (a), we only have two bins, and much of the detail in the distribution is lost. In plot (b), the bins are very narrow, and no bin contains more than three observations. Again, it's harder to see the shape of the distribution clearly.

2.8.2 Skewed distributions

If the shape of the distribution is not symmetric, we say that it is **skewed**. We will draw the histogram again, with the mean and median indicated. For symmetric distributions, the mean and median distributions should be approximately the same. For **negatively skewed** distributions, the mean will be lower than the median (as we have here), and vice versa for **positively skewed** distributions.

```
hist(life2015, xlab = "life expectancy in 2015 (years)",
     main = "histogram of life expectancy \n in  181 countries ")
abline(v = mean(life2015), col = "red", lwd = 2)
abline(v = median(life2015), col = "blue", lwd = 2)
```

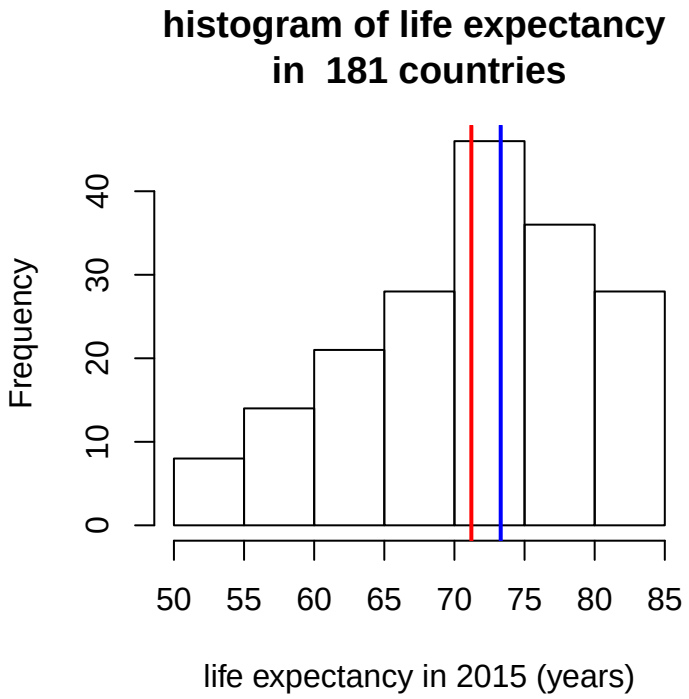


Figure 2.5: The histogram of life expectancies with the mean indicated as the red line, and the median as the blue line. For skewed distributions, the mean and median can be noticeably different.

2.8.3 Box plots for comparing multiple distributions

A box plot can also be used to plot a distribution. For a single distribution, a histogram is usually preferred, but if the aim is to compare multiple distributions, box plots make it easier to show distributions alongside each other.
Box plots can be drawn horizontally or vertically. An illustration (and comparison with a histogram) is given below.

- The thick line shows the median value;

- the left and right sides (or top and bottom, if drawn vertically) of the box show the quartiles (25th and 75th percentiles);
- the dashed lines (the 'whiskers') extend to the most extreme observed values that are no more than 1.5 × the inter-quartile range from the edge of the box;
- individual observations not covered by the whiskers are sometimes indicated as points on the plot, referred to as **outliers** (but there are no such observations here).

```
par(mfrow = c(2, 1))
hist(life2015, xlab = "life expectancy in 2015 (years)",
     main = "histogram of life expectancy \n in 181 countries ")
boxplot(life2015, horizontal = TRUE,
       xlab = "life expectancy in 2015 (years)",
     main = "boxplot of life expectancy \n in 181 countries ")
```

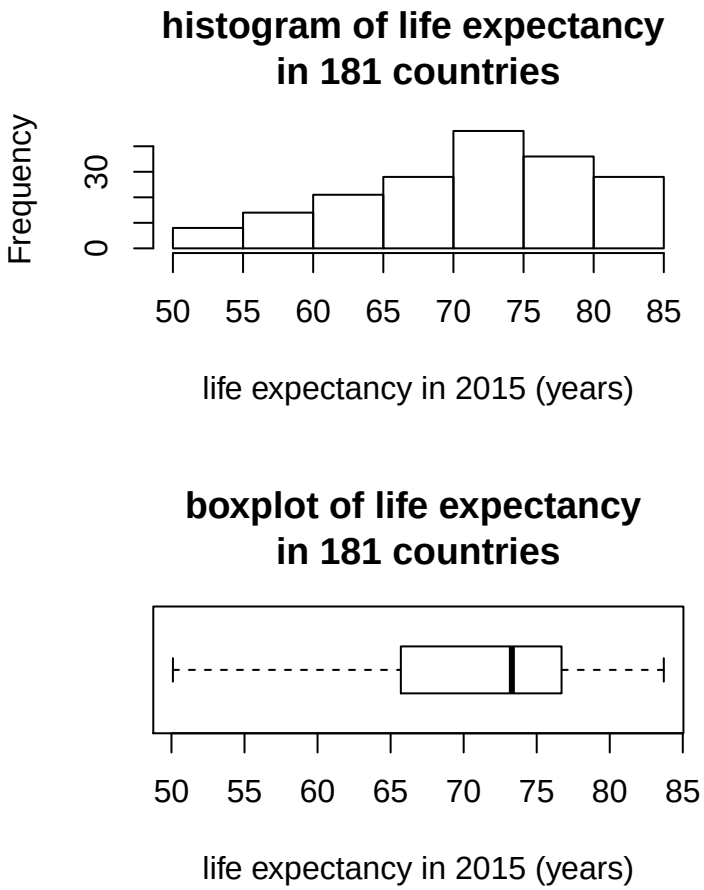


Figure 2.6: A boxplot is another way to show a distribution. For a single distribution, a histogram is usually more informative.

If we want to compare life expectancies across the six regions, we can plot six box plots alongside each other.

```
par(mar=c(10, 4.1, 4.1, 2.1)) # increase the margin below the x-axis
plot(Region, life2015, las = 2,
     ylab = "life expectancy in 2015 (years)")
```

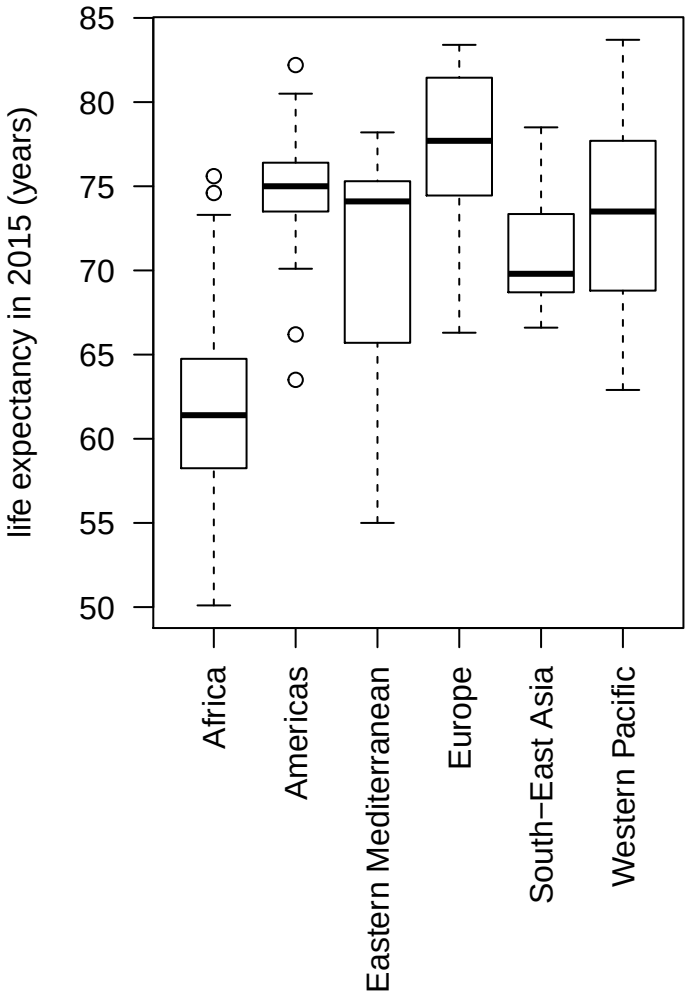


Figure 2.7: Boxplots are good for comparing multiple distributions in a single plot. We also note some outliers (the circles) in Africa and the Americas.

Note that there are two ways to produce a box plot in R. Because `Region` is a `factor` variable, the command `plot` will automatically select a box plot.

```
plot(Region, life2015)
```

There is also a specific box plot command, with a slightly different syntax:

```
boxplot(life2015 ~ Region)
```

2.8.4 Scatter plots for bivariate observations

We have two observations per country: life expectancies in 2000 and 2015. A simple scatter plot is a good way to see how life expectancy has changed:

```
plot(life2000, life2015)
```

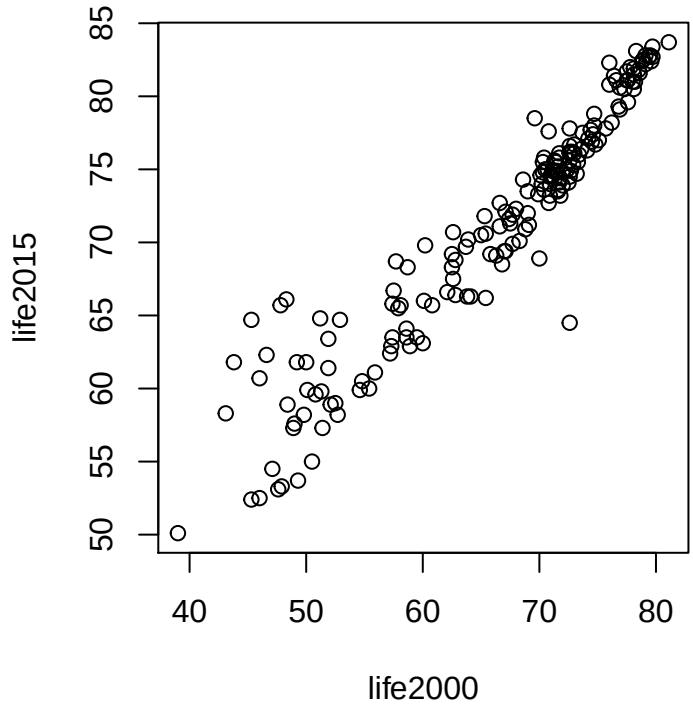


Figure 2.8: A simple scatterplot produced with the `plot` command in R. You will almost always need to add extra arguments/commands to give more descriptive axes labels etc. and make the plot more informative.

Firstly, we must make the axes labels more informative, and give the units: **always specify the units of measurement in any plot**. Additionally, we might

- use the same range on each axis;
- use different colours for the six regions
- add the line $y = x$, corresponding to no change in life expectancy from 2000 to 2015.

```
plot(life2000, life2015, pch = 16, col = Region,
     xlim = c(35, 85),
     ylim = c(35, 85),
     xlab = "life expectancy in 2000 (years)",
     ylab = "life expectancy in 2015 (years)")
abline(0, 1, lty = 2)
legend("bottomright", legend = levels(Region),
       col = 1:6, pch = rep(16, 6))
```

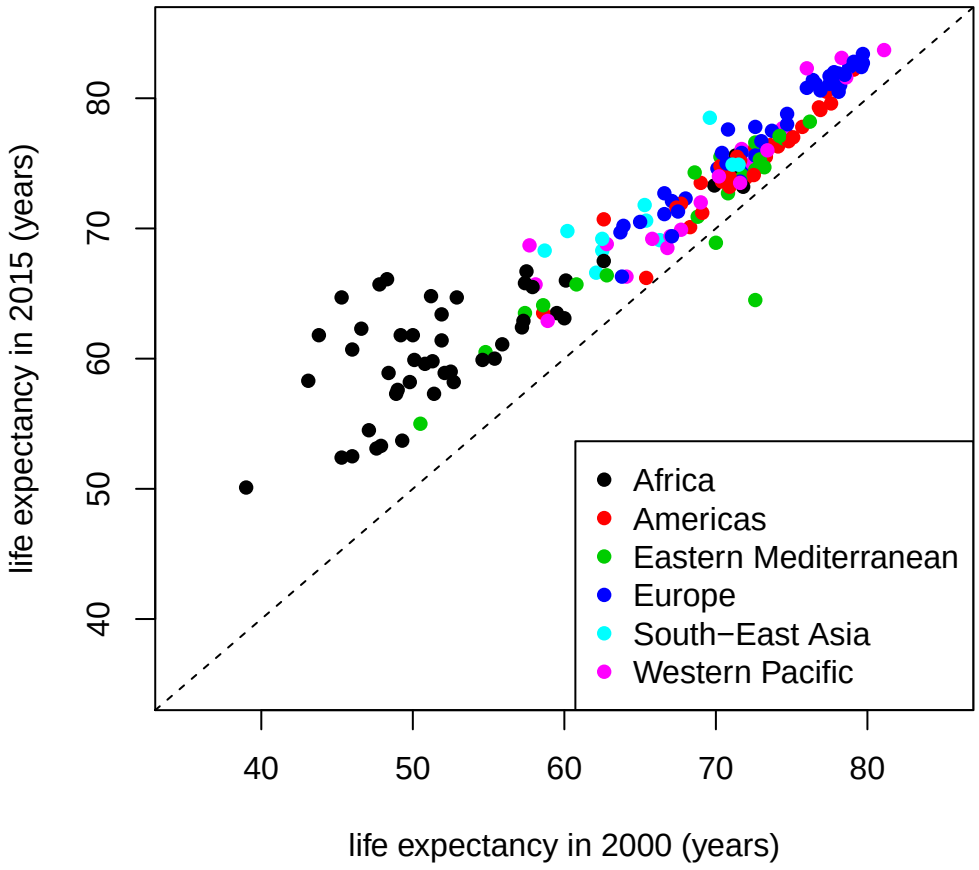


Figure 2.9: A scatterplot to compare life expectancies in 2015 and 2000. The dashed line draws attention to the fact that life expectancy has decreased in only two countries.

It is particularly striking from the plot that there there are only two countries where life expectancy has decreased (and in one case substantially). It is only this scatter plot that has revealed this: their life expectancies were not 'unusual' in either 2000 or 2015, and so looking at each variable in isolation would not have highlighted anything.
We can find out which two countries these are.

```
Country[life2015 < life2000]

## [1] Iraq                  Syrian Arab Republic
## 181 Levels: Afghanistan Albania Algeria Angola ... Zimbabwe
```

and then inspect the corresponding values

```
lifeExp[Country=="Iraq", ]

##    Country                Region life2000 life2015
## 78    Iraq Eastern Mediterranean       70     68.9

lifeExp[Country=="Syrian Arab Republic", ]

##                   Country                Region life2000 life2015
## 159 Syrian Arab Republic Eastern Mediterranean     72.6     64.5
```

(Directly, we could have done)

```
lifeExp[life2015 < life2000, ]

##                   Country                Region life2000 life2015
## 78                   Iraq Eastern Mediterranean     70.0     68.9
## 159 Syrian Arab Republic Eastern Mediterranean     72.6     64.5
```

You can also select points directly in the graphics window, and get R to add the country label.

```
identify(life2000, life2015, Country, n = 1)
```

2.9 Case study: diagnosing breast cancer

A study of breast diagnosis (and prognosis) based on the work of Prof. Olvi L. Mangasarian and Dr. William H. Wolberg is described at http://pages.cs.wisc.edu/~olvi/uwmp/cancer.html. The aim is to diagnose breast cancer based on a technique known as Fine Needle Aspiration (FNA). An FNA sample is taken from a breast mass, and then examined on a microscopic slide. Various characteristics of the cell nuclei are then recorded (30 in total). The aim is then to determine whether the breast mass is cancerous or not ("malignant" or "benign"), based on these measurements only.
To design a diagnostic tool, a data set of 569 samples has been obtained, where the 30 measurements are obtained for each sample, together with the *true* status of the breast mass (malignant or benign), obtained from a more invasive surgical procedure. The task is then to investigate whether the true status can be determined from the 30 measurements *only* (which would mean patients could be diagnosed without surgery)
The R script `breastcancer.R` illustrates how a simple diagnostic tool can be constructed, based on the exploratory methods studied so far. (More advanced statistical methods can give more accurate diagnostic tools.)

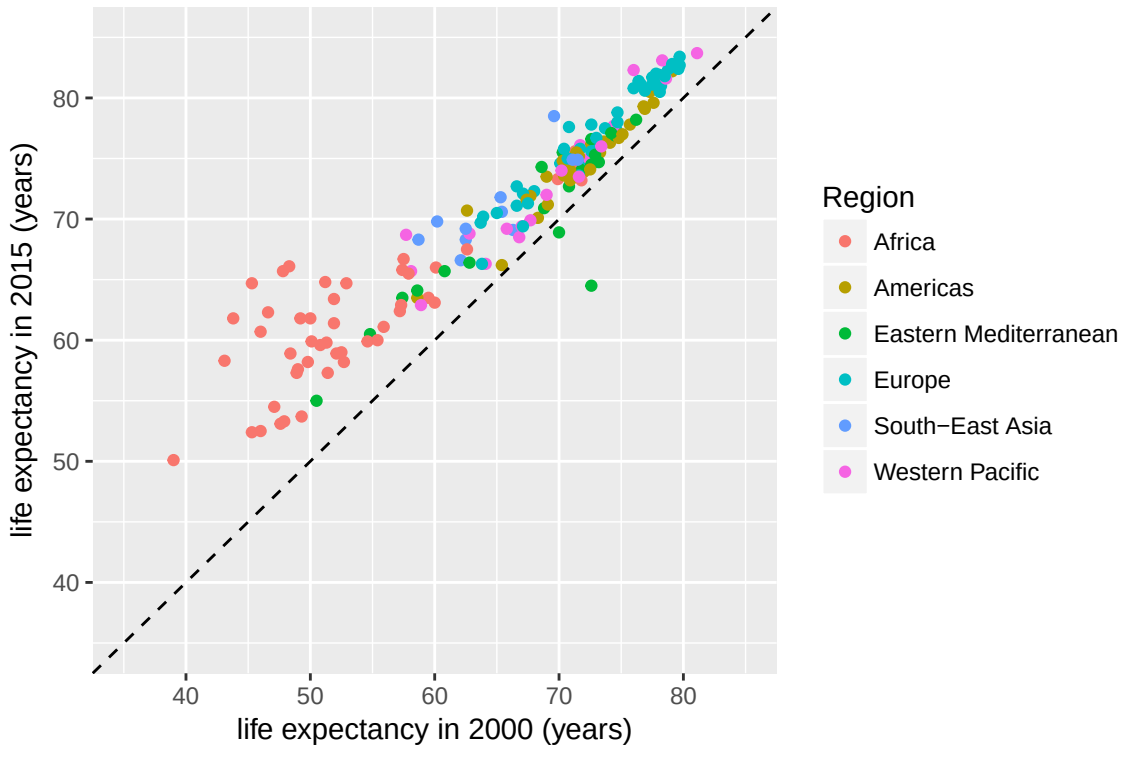
2.10 Graphics in R: advanced topics

The following two topics are not part of the syllabus for MAS113, but may of interest.

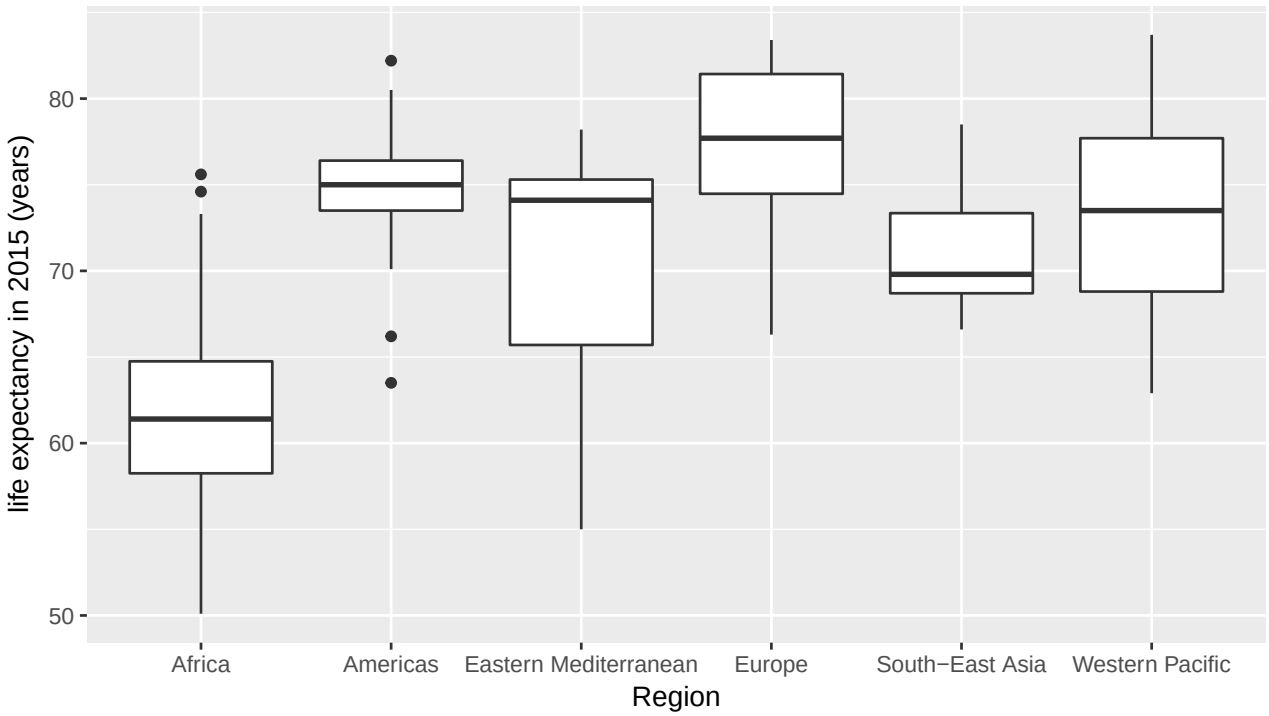
2.10.1 ggplot2

The graphics we have been using in R are known as *base graphics* (they are what you get when you download and install R). A more advanced system is `ggplot2` and has become popular with many R users (solutions to problems posted online about plotting are often given using ggplot2 graphics). An introduction is available on the MOLE. Below are some examples of plots produced with ggplot2. Note that handling of legends is a little easier.

```
library(ggplot2)
ggplot(data = lifeExp, aes(x = life2000,
                           y = life2015,
                           colour = Region)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed" ) +
  xlim(c(35, 85)) +
  ylim(c(35, 85)) +
  labs(x = "life expectancy in 2000 (years)",
       y = "life expectancy in 2015 (years)")
```



```
ggplot(data = lifeExp, aes(x = Region,
                           y = life2015)) +
  geom_boxplot() +
  labs(y = "life expectancy in 2015 (years)")
```



2.10.2 Plotly

Plotly is an online tool for data analysis and visualisation, and can also be used within languages such as R and Python. The code below will produce a scatter plot, in which you can 'hover' over points to see the country labels, amongst other things.

```
library(plotly)
plot_ly(lifeExp, x = ~life2000, y =~life2015,
        type = 'scatter',
        mode = 'markers',
        hoverinfo= 'text',
        color = ~Region,
        text = ~Country)
```