Chapter 1: Interactive Notebook for Students

Ram Gopal, Dan Philps, and Tillman Weyde

Summer 2022

Contents

Simple Computations

Create a new script file. Let us begin with simple computations in R using `*`,`/`,`sqrt()`,`exp()`, and `log()` operators and functions.

```
25*77
```

```
## [1] 1925
```

```
25/5
```

```
## [1] 5
```

```
sqrt(77)
```

```
## [1] 8.775
```

```
exp(2.5)
```

```
## [1] 12.18
```

```
log(55)
```

```
## [1] 4.007
```

Creating Vectors

To create a vector, the appropriate syntax is `v = c()`. An example is:

```
v = c(1,2,4,63,7,5)
```

To create a vector of a sequence, we use `seq()`.

```r
vec1 = seq(1,5)
vec1
```

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

```r
vec2 = seq(2,55,by = 3)
vec2
```

```
##  [1]  2  5  8 11 14 17 20 23 26 29 32 35 38 41 44 47 50 53
```

The option `by` in the sequence function allows you to increment by any value.

To create a vector that repeats a number, character, or a vector of numbers or characters, we use `rep()`. Characters must be in quotations.

```r
vec1 = rep(1,10)
vec1
```

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

```r
vec2 = rep(c(3,4,5),3)
vec2
```

```
## [1] 3 4 5 3 4 5 3 4 5
```

```r
vec3 = rep("c",5)
vec3
```

```
## [1] "c" "c" "c" "c" "c"
```

The second element in parenthesis is the number of times you want the first element to be repeated.

To find the sum of a vector, we use the command `sum()`.

```r
sum(vec1)
```

```
## [1] 10
```

The following are other useful vector based functions in R.

`min(x)` - minimum value of vector x
`max(x)` - maximum value of vector x
`mean(x)` - mean value of vector x
`median(x)` - median value of vector x
`quantile(x, p)` - pth quantile of vector x
`sd(x)` - standard deviation of vector x
`var(x)` - variance of vector x
`IQR(x)` - Inter Quartile Range (IQR) of vector x
`diff(range(x))` - total range of vector x

Let us create three vectors to store information about weight, height, and gender. since gender is not numeric data, we use quotation marks when creating the vector.

Note that initially, the gender vector is character. This can be seen using the `class()` command. The gender vector must be a factor (or more commonly refered to as categorical data) in order to be plotted, and thus we write `factor(gender)` in the parenthesis of the `plot()` command.

```r
weight = c(60,72,57,90,95,72)
height = c(1.75,1.8,1.65,1.9,1.74,1.91)
gender = c("m","f","m","f","f","m")
gender
```

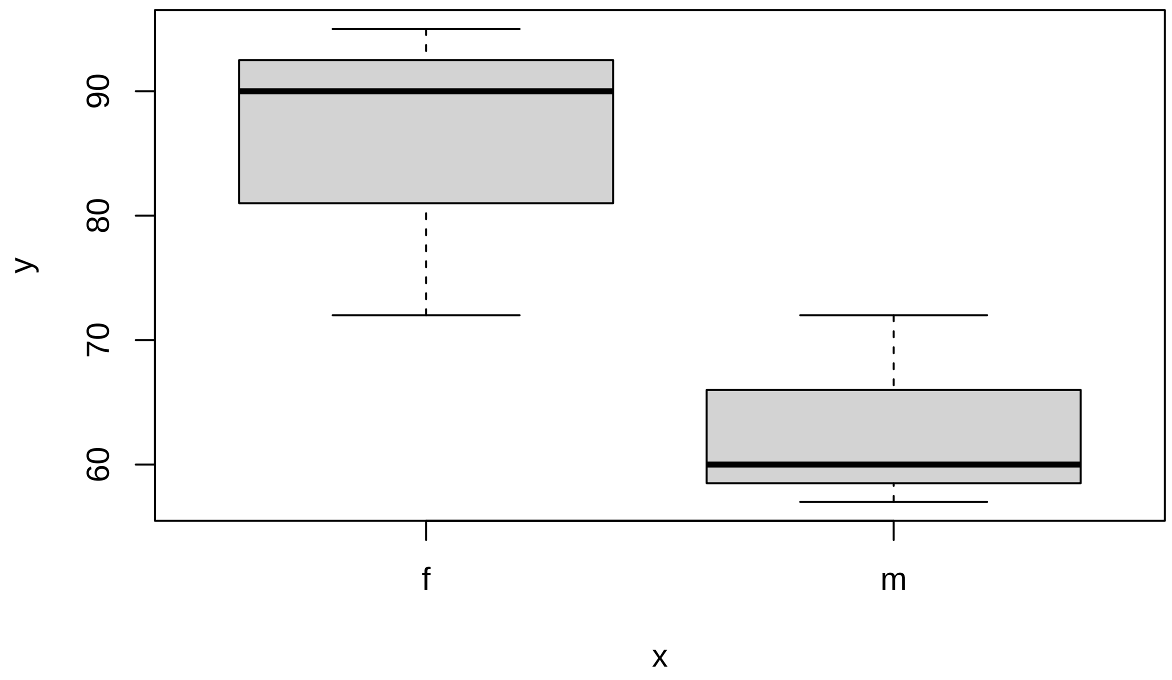
```
## [1] "m" "f" "m" "f" "f" "m"
```

```
class(gender)
```

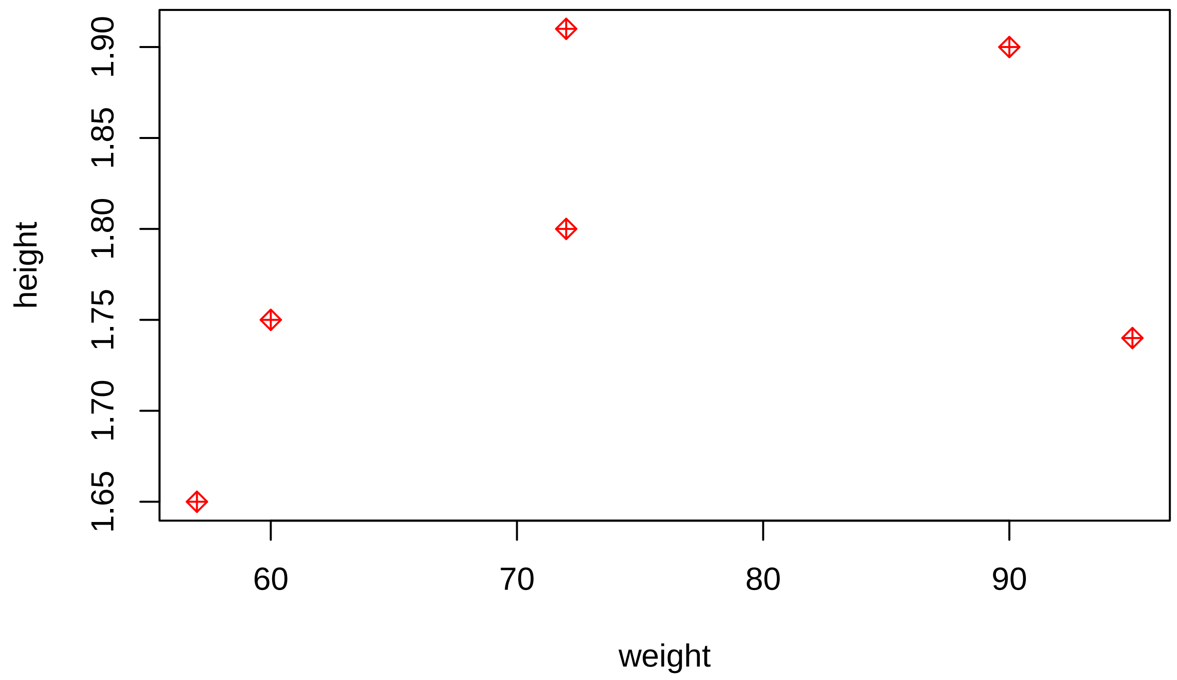
```
## [1] "character"
```

Using the `plot()` command, we can plot these vectors as scatterplots or boxplots. The order within the parenthesis is the variable you want on the x axis, then the variable you want on the y axis. `col` can be used to change the color of the dots in a scatterplot, and `pch` changes the design of the dots.

```
plot(factor(gender),weight)
```



```
plot(weight,height,col = "red",pch=9)
```



Creating Data Frames

A data frame is a two dimensional variable. To create a data frame called **ghw** using these three vectors, we use the command `data.frame()`. `View()` is a key command that allows you to see the data frame as a table in a separate tab.

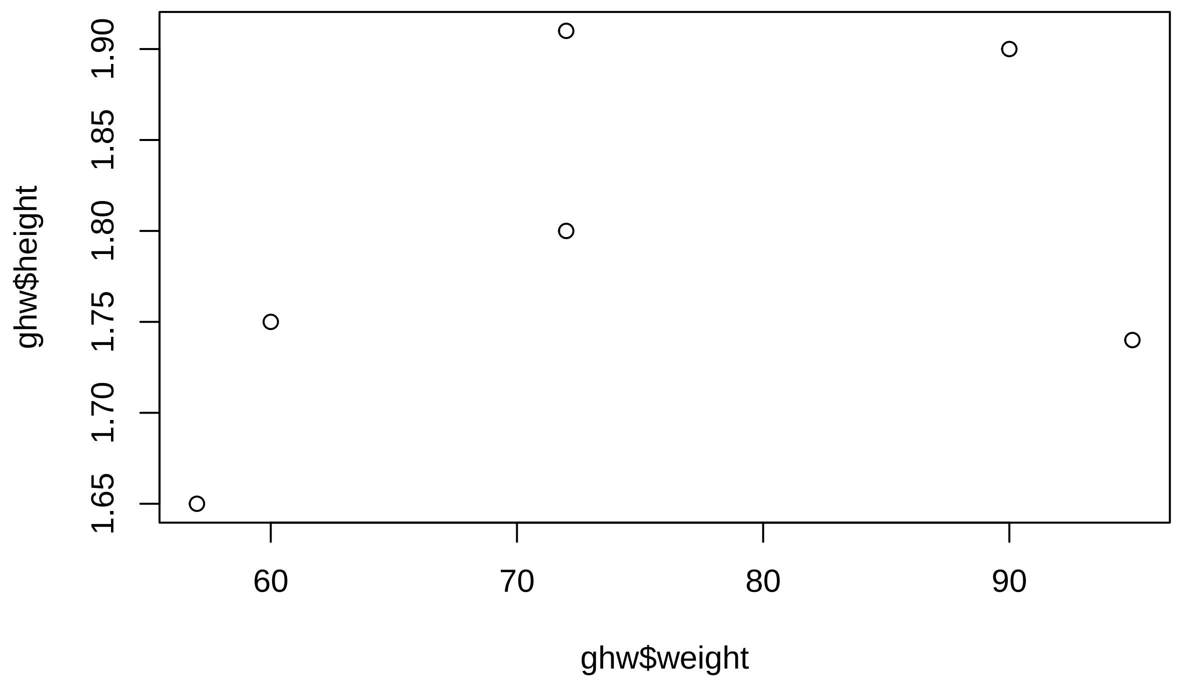
```
ghw=data.frame(gender,height,weight)
ghw
```

```
##   gender height weight
## 1      m   1.75     60
## 2      f   1.80     72
## 3      m   1.65     57
## 4      f   1.90     90
## 5      f   1.74     95
## 6      m   1.91     72
```

```
#View(ghw)
```

Now that the data frame is created, if you want to plot two variables from the data frame with the `plot()` command, you must specify both the data frame and the variable from the data frame. The appropriate syntax is, for example, `ghw$weight`, `ghw$height`, and `ghw$gender`.

```
plot(ghw$weight,ghw$height)
```



Information about Data Frames

If you need a quick summary of the structure or contents of a data frame, there are a few helpful commands. `summary()` gives you the min, max, mean, median, and 1st and 3rd quartiles of numerical data, and the length, class, and mode of categorical data. `dim()` gives you the number of rows and columns of your data frame. `str()` gives you the first few elements of each variable. `head()` gives you the first few rows of the data frame.

```
summary(ghw)
```

```
##     gender              height          weight
##  Length:6           Min.   :1.65   Min.   :57.0
##  Class :character   1st Qu.:1.74   1st Qu.:63.0
##  Mode  :character   Median :1.77   Median :72.0
##                     Mean   :1.79   Mean   :74.3
##                     3rd Qu.:1.88   3rd Qu.:85.5
##                     Max.   :1.91   Max.   :95.0
```

```
dim(ghw)
```

```
## [1] 6 3
```

```
str(ghw)
```

```
## 'data.frame':    6 obs. of  3 variables:
##  $ gender: chr  "m" "f" "m" "f" ...
##  $ height: num  1.75 1.8 1.65 1.9 1.74 1.91
##  $ weight: num  60 72 57 90 95 72
```

```r
head(ghw)
```

```
##   gender height weight
## 1      m   1.75     60
## 2      f   1.80     72
## 3      m   1.65     57
## 4      f   1.90     90
## 5      f   1.74     95
## 6      m   1.91     72
```

To see the full data frame in another tab, use the `View()` command.

```r
#View(ghw)
```

Adding Columns

You can add columns to a data frame based on the data from existing columns. Let's calcuate the Body Mass Index which is defined as:

$$bmi = \frac{weight}{height^2}$$

```r
ghw$bmi = ghw$weight/ghw$height^2
knitr::kable(ghw)
```

gender	height	weight	bmi
m	1.75	60	19.59
f	1.80	72	22.22
m	1.65	57	20.94
f	1.90	90	24.93
f	1.74	95	31.38
m	1.91	72	19.74

Statistics – A First Look

Formally speaking, statistics is a branch of mathematics that transforms data into useful information for decision makers. There are two major areas of statistics – descriptive statistics and inferential statistics. As the name indicates, the objective of descriptive statistics is to describe and understand features of a given set of data. The much broader area of inferential statistics aims to study a sample of data to infer insights on the larger population from which the sample was drawn. Descriptive statistics allow us to detect outliers (i.e., underperforming sales executives), to compare different stock returns in relation to the oil price, using correlations, and more. Let's begin by computing the arithmetic mean of two groups of data.

```r
g1 = c(34, 49, 64, 38, 60, 78, 67, 36, 19, 37)
g2 = c(77, 75, 78, 41, 51, 20, 61, 73, 76, 38)

mean(g1)
```

```
## [1] 48.2
```

```r
mean(g2)
```

```
## [1] 59
```

We can conclude that the average age of individuals in the second group is higher than in the first group. A first glimpse of insights on our data! The mean as a measure of central tendency, however, is susceptible to extreme values, also termed outliers. Let us see an example of this. Consider the following two groups of data:

```
g1a = c(34, 49, 64, 38, 60, 78, 67, 36, 19, 37)
g2a = c(34, 49, 64, 38, 60, 400, 67, 36, 19, 37)
mean(g1a)
```

```
## [1] 48.2
```

```
mean(g2a)
```

```
## [1] 80.4
```

A single outlier value in our data nearly doubles the value of the mean, which highlights the great impact of outliers on it. A more robust measure to outliers is the median. The median is simply the "middle value" such that half the data has values less than or equal to the median, and half larger than or equal to it. Let us compute the median of the two groups.

```
median(g1a)
```

```
## [1] 43.5
```

```
median(g2a)
```

```
## [1] 43.5
```

Let us compute the values of the variance for the groups g1 and g2.

```
var(g1)
```

```
## [1] 340.4
```

```
var(g2)
```

```
## [1] 420
```

In addition to a larger mean, group 2 has a higher variance than g1 indicating that the values in the second group are more spread out than in the first group. The code below computes the standard deviation, and we can verify that it is indeed the square root of the variance.

```
sd(g1)
```

```
## [1] 18.45
```

```
sd(g2)
```

```
## [1] 20.49
```

The following code computes the coefficient of variation for groups 1 and 2:

```
sd(g1)*100/mean(g1)
```

```
## [1] 38.28
```

```
sd(g2)*100/mean(g2)
```

```
## [1] 34.74
```

The z-score indicates how many standard deviations a data point is away from the mean. It is a key business analytic that can be used for anything from identifying mis-sized widgets on a factory line to determining historic extremes in valuations of credit spreads.

```
(g1-mean(g1))/sd(g1)
```

```
##  [1] -0.76965  0.04336  0.85637 -0.55285  0.63957  1.61518  1.01897 -0.66125
##  [9] -1.58266 -0.60705
```

```
scale(g1)
```

```
##           [,1]
##  [1,] -0.76965
##  [2,]  0.04336
##  [3,]  0.85637
##  [4,] -0.55285
##  [5,]  0.63957
##  [6,]  1.61518
##  [7,]  1.01897
##  [8,] -0.66125
##  [9,] -1.58266
## [10,] -0.60705
## attr(,"scaled:center")
## [1] 48.2
## attr(,"scaled:scale")
## [1] 18.45
```

```
g1a = c(34, 49, 64, 38, 60, 78, 67, 36, 19, 37)
g2a = c(34, 49, 64, 38, 60, 400, 67, 36, 19, 37)
scale(g1a)[,1]
```

```
##  [1] -0.76965  0.04336  0.85637 -0.55285  0.63957  1.61518  1.01897 -0.66125
##  [9] -1.58266 -0.60705
```

```
scale(g2a)[,1]
```

```
##  [1] -0.4095 -0.2771 -0.1447 -0.3742 -0.1800  2.8204 -0.1183 -0.3918 -0.5418
## [10] -0.3830
```

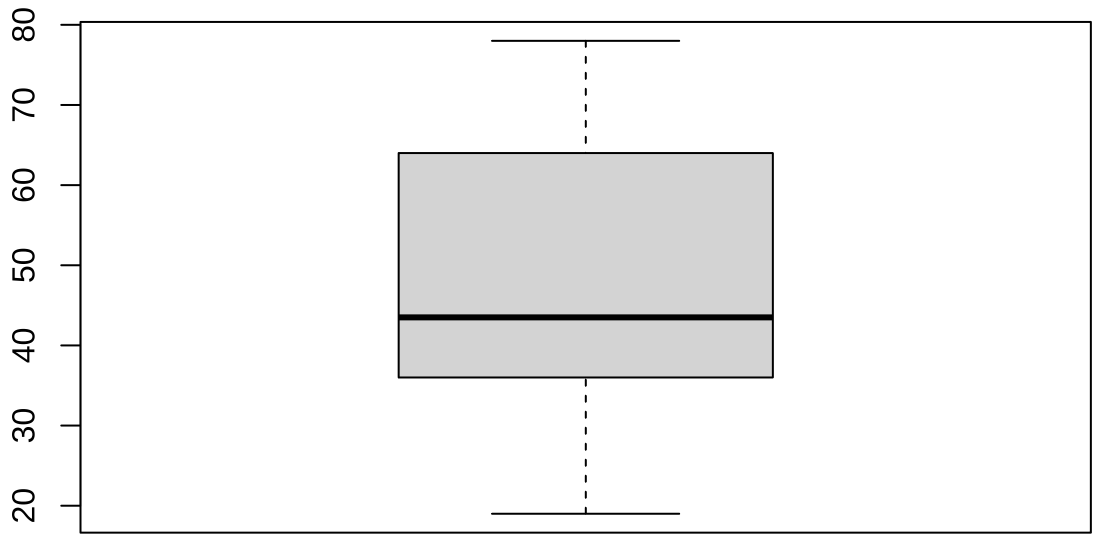
The sixth observation in g2a has a z-score of 2.82 suggesting a possible outlier. Z-scores are a simple, easy tool to detect possible outliers in our data.

An extension of the median is the measure quartile, and it illustrates how the data values are distributed.

```
quantile(g1)
```

```
##    0%   25%   50%   75%  100% 
## 19.00 36.25 43.50 63.00 78.00
```

```
boxplot(g1)
```



Let us code the computation of covariance and correlation between height and weight.

```
cov(ghw$weight,ghw$height)
```

```
## [1] 0.6773
```

```
cor(ghw$weight,ghw$height)
```

```
## [1] 0.4379
```

Some Useful Resources

1. The R reference card is very useful if you want to look up the basic syntax. You are strongly recommended to download and keep it handy as you work to get comfortable with coding in R. R Reference Card
2. There are a number of other cheatsheets for specific packages and functionalities. You can find a list of them at Cheatsheets

3. Quick-R