week # 2

Bashir,I. (2022)

20/03/2022

An R Introduction

```
1+1
```

```
## [1] 2
```

```
4-2
```

```
## [1] 2
```

```
2*3
```

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

```
6/2
```

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

```
2^3
```

```
## [1] 8
```

You can solve values into variables (objects):

```
a <- 1+1
a
```

```
## [1] 2
```

```
b <- 3*2
b
```

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

```
b/a
```

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

We can also store characters (strings) inside a variable

```
c <- "Aa"
c
```

```
## [1] "Aa"
```

```
d <- TRUE
d
```

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

There are different object classes in R:

```
class(a)
```

```
## [1] "numeric"
```

```
class(b)
```

```
## [1] "numeric"
```

```
class(c)
```

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

```
class(d)
```

```
## [1] "logical"
```

Change a class of an object e.g:

```
d <- as.numeric(d)
d
```

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

```
class(d)
```

```
## [1] "numeric"
```

Working with Vectors

The command c() is used to create vectors:

```
vec1 <- c(0, -1, 1/2, 1000, 2)
vec1
```

```
## [1]    0.0   -1.0    0.5 1000.0    2.0
```

```
vec2 <- c(1:5)
vec2
```

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

Select elements using square brackets:

```
vec1[1]
```

```
## [1] 0
```

operations with vectors:

```
vec1 + vec2
```

```
## [1]    1.0    1.0    3.5 1004.0    7.0
```

```
vec2 * 3
```

```
## [1]  3  6  9 12 15
```

other functions to create specific vectors:

```
vec3 <- seq(from = 2, to = 4, by = 0.5)
vec3
```

```
## [1] 2.0 2.5 3.0 3.5 4.0
```

```
vec4 <- rep(x = 1:3, times = 3)
vec4
```

```
## [1] 1 2 3 1 2 3 1 2 3
```

```
vec5 <- rep(x = 1:3, each = 3)
vec5
```

```
## [1] 1 1 1 2 2 2 3 3 3
```

```
table(vec4)
```

```
## vec4
## 1 2 3
## 3 3 3
```

```
rev(vec5)
```

```
## [1] 3 3 3 2 2 2 1 1 1
```

```
unique(vec5)
```

```
## [1] 1 2 3
```

R Environment

To see the list of elements in the environment is using the function `ls()`:

```
ls()
```

```
## [1] "a"    "b"    "c"    "d"    "vec1" "vec2" "vec3" "vec4" "vec5"
```

```
ls()[5] # this selects the fifth element
```

```
## [1] "vec1"
```

To remove any of the objects, we use the function `rm()`:

```
rm("c") # to remove a specific object
ls()
```

```
## [1] "a"    "b"    "d"    "vec1" "vec2" "vec3" "vec4" "vec5"
```

```
rm(list = c("a","b"))
ls()
```

```
## [1] "d"    "vec1" "vec2" "vec3" "vec4" "vec5"
```

```
rm(list = ls()[-4]) # to remove all expect the forth object
ls()
```

```
## [1] "vec3"
```

```
rm(list = ls()) # remove all objects
ls()
```

```
## character(0)
```

Loading external data

One of the functions to load external data is `read.csv()`:

```
data <- read.csv(file = "data/mouse.csv", header = TRUE)
head(data)
```

```
##   BW M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14
## 1 NA  0  0  0  0  0  0  0  0  0   0   0   0   0   0
## 2 50  1  1  1  1  1  1  1  1  1   1   1   1   1   1
## 3 54  1  1  1  1  1  1  1  1  1   1   1   1   1   0
## 4 49  0  1  1  1  1  1  1  1  1   1   1   1   1   1
## 5 41  0  0  0  0  0  0  0  0  0   0   0   0   0   0
## 6 36  1  1  1  1  1  1  1  1  1   1   1   1   1   1
```

```
tail(data)
```

```
##     BW M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14
## 99  56  0  0  0  0  0  0  0  0  0   0   0   0   0   0
## 100 50  0  0  0  0  0  0  0  0  0   0   0   0   0   0
## 101 45  1  1  1  1  0  0  0  0  0   0   0   0   0   0
## 102 43  1  1  1  0  0  0  0  0  0   0   0   0   0   0
## 103 37  0  0  0  0  0  0  0  0  0   0   0   0   0   1
## 104 35  1  0  0  0  0  0  0  0  0   0   0   0   0   0
```

```
colnames(data)
```

```
##  [1] "BW"  "M1"  "M2"  "M3"  "M4"  "M5"  "M6"  "M7"  "M8"  "M9"  "M10" "M11"
## [13] "M12" "M13" "M14"
```

```
dim(data)
```

```
## [1] 104  15
```

```
str(data)
```

```
## 'data.frame':    104 obs. of  15 variables:
##  $ BW : int  NA 50 54 49 41 36 48 37 55 42 ...
##  $ M1 : int  0 1 1 0 0 1 0 0 1 0 ...
##  $ M2 : int  0 1 1 1 0 1 0 0 1 0 ...
##  $ M3 : int  0 1 1 1 0 1 0 0 1 0 ...
##  $ M4 : int  0 1 1 1 0 1 0 0 1 0 ...
##  $ M5 : int  0 1 1 1 0 1 0 0 1 0 ...
##  $ M6 : int  0 1 1 1 0 1 0 0 1 0 ...
##  $ M7 : int  0 1 1 1 0 1 0 1 1 0 ...
##  $ M8 : int  0 1 1 1 0 1 0 1 1 0 ...
##  $ M9 : int  0 1 1 1 0 1 0 1 1 0 ...
##  $ M10: int  0 1 1 1 0 1 0 1 1 0 ...
##  $ M11: int  0 1 1 1 0 1 0 1 1 0 ...
##  $ M12: int  0 1 1 1 0 1 0 1 1 0 ...
##  $ M13: int  0 1 1 1 0 1 0 1 1 0 ...
##  $ M14: int  0 1 0 1 0 1 0 1 0 0 ...
```

To select a particular set of rows/columns, we use square brackets:

```
data[c(1:3),] # first three rows
```

```
##   BW M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14
## 1 NA  0  0  0  0  0  0  0  0  0   0   0   0   0   0
## 2 50  1  1  1  1  1  1  1  1  1   1   1   1   1   1
## 3 54  1  1  1  1  1  1  1  1  1   1   1   1   1   0
```

```
data[,2] # second column
```

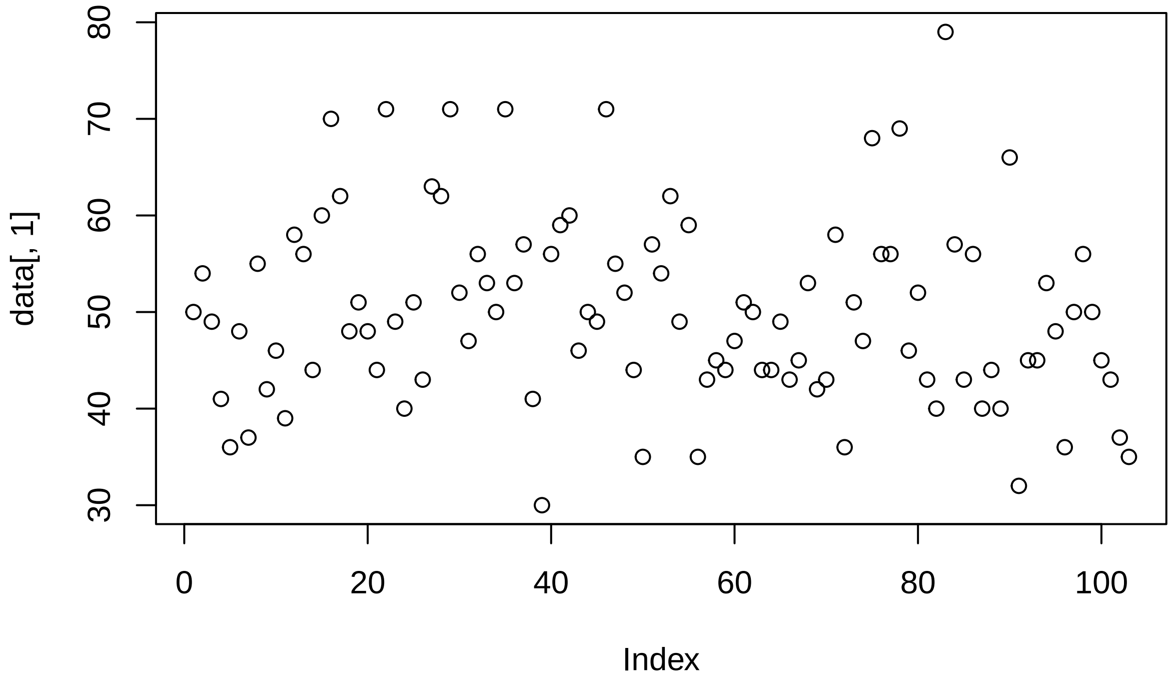
```
##   [1] 0 1 1 0 0 1 0 0 1 0 0 0 1 1 0 1 1 0 0 0 0 0 1 0 0 1 0 1 1 1 1 0 0 0 1 1 0
##  [38] 0 0 0 1 1 0 1 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
##  [75] 1 1 1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 1
```

```
data <- data[-1,]
dim(data)
```

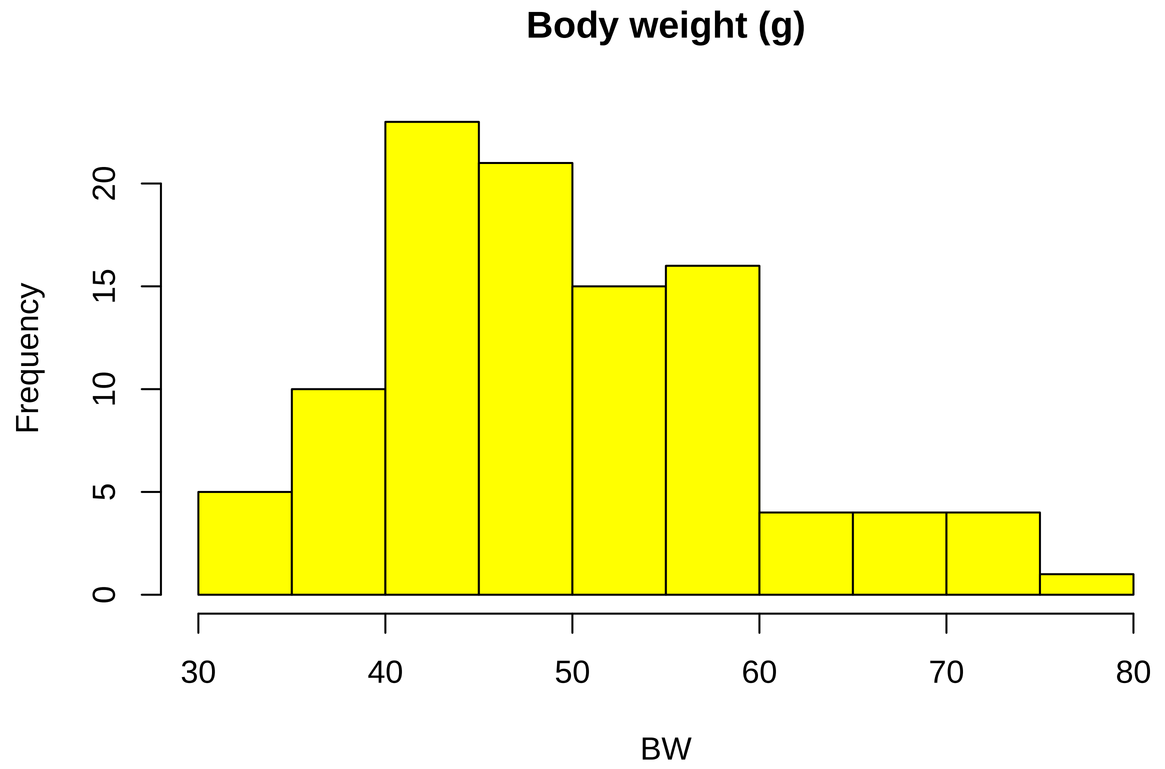
```
## [1] 103  15
```

To plot the BW phenotype:

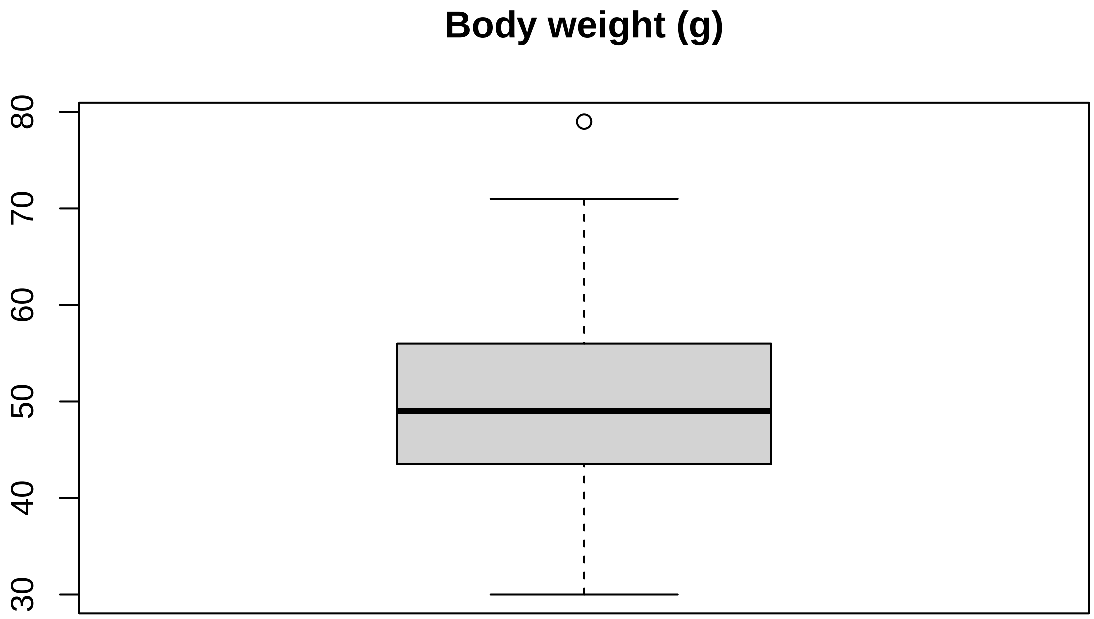
```
plot(data[,1])
```



```
hist(data[,"BW"], main = "Body weight (g)", col = "yellow", xlab = "BW")
```

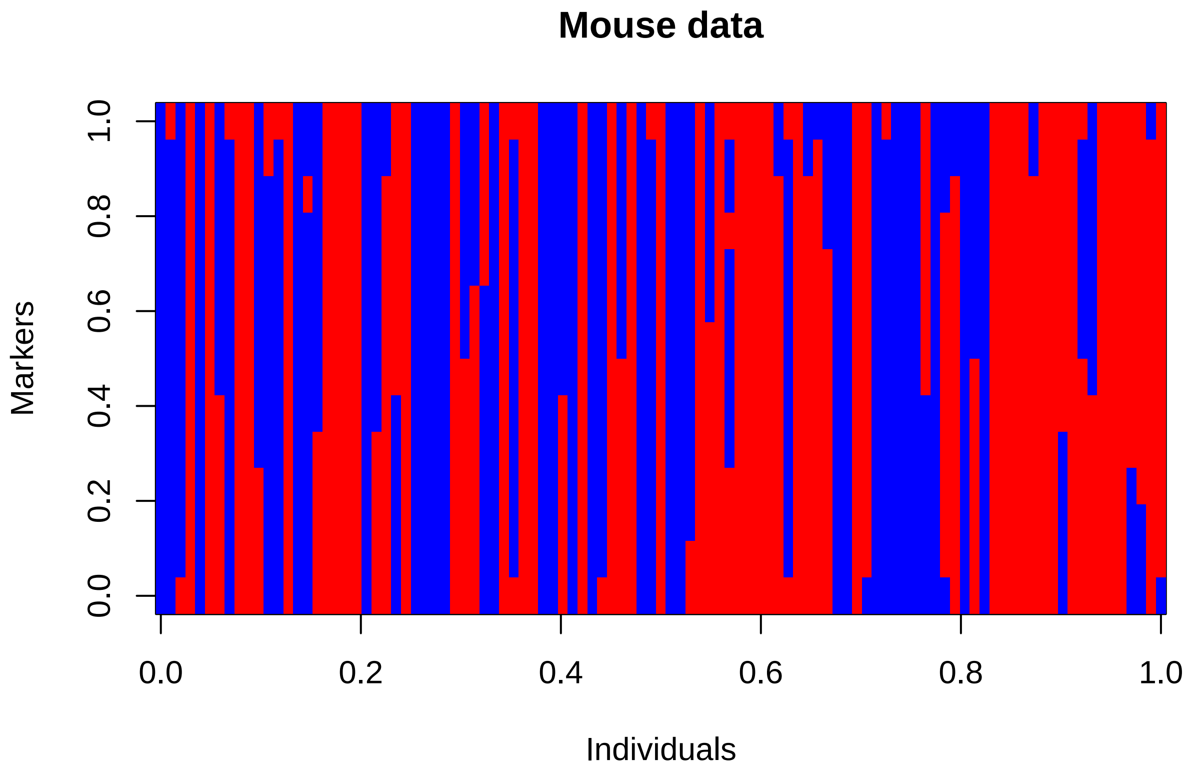


```
boxplot(data$BW, main = "Body weight (g)")
```



To plot information about markers:

```
image(as.matrix(data[,2:15]), col = c("red", "blue"), xlab = "Individuals", ylab = "Markers", main = "Mo
```



Math operations

To run some math operations, we have:

```
sum(data$BW)
```

```
## [1] 5156
```

```
mean(data$BW)
```

```
## [1] 50.05825
```

To run a simple regression:

```
reg1 <- lm(formula = BW ~ M1, data = data)
reg1
```

```
##
## Call:
## lm(formula = BW ~ M1, data = data)
##
## Coefficients:
## (Intercept)           M1
##      47.323        6.873
```

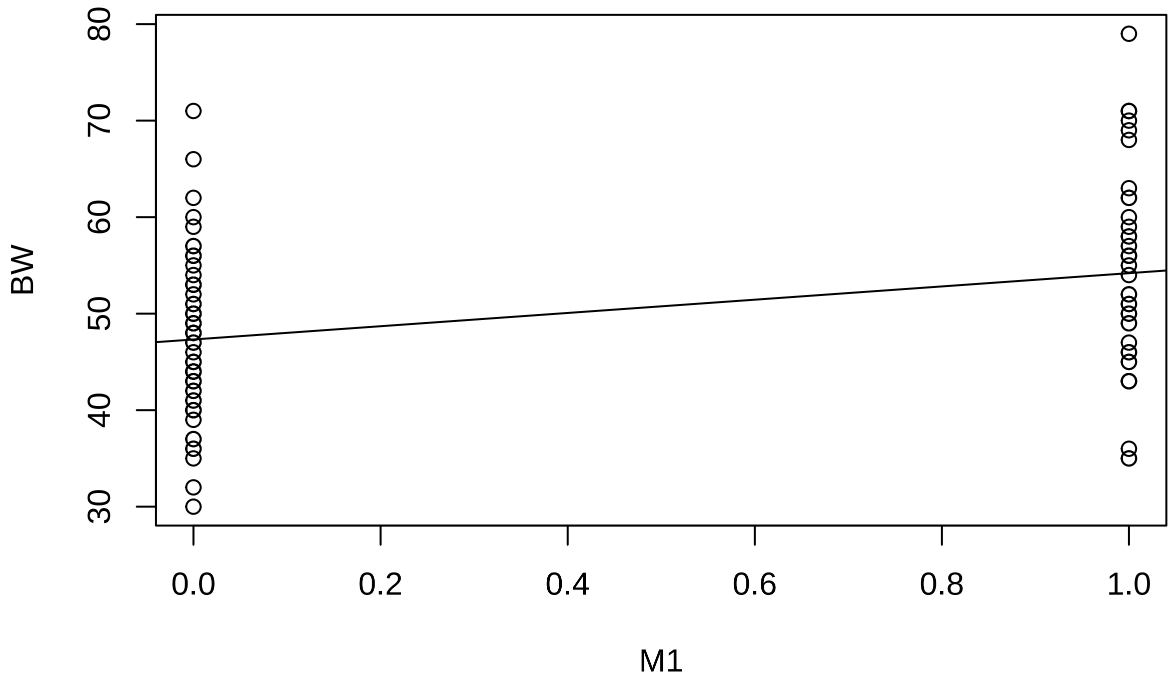
```
summary(reg1)
```

```
##
## Call:
## lm(formula = BW ~ M1, data = data)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.1951  -5.8226  -0.3226   5.6774  24.8049
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   47.323      1.155  40.966  < 2e-16 ***
## M1             6.873      1.831   3.754 0.000291 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.096 on 101 degrees of freedom
## Multiple R-squared:  0.1224, Adjusted R-squared:  0.1137
## F-statistic: 14.09 on 1 and 101 DF,  p-value: 0.0002914
```

A regression plot:

```
plot(BW ~ M1, data = data)
abline(reg1)
```



Programming in R

We create loops in R using `for()`:

```
for(i in 2:15) {
  print(table(data[,i]))
}
```

```
##
##  0  1
## 62 41
##
##  0  1
## 61 42
##
##  0  1
## 60 43
##
##  0  1
## 61 42
##
##  0  1
## 60 43
##
##  0  1
## 59 44
##
##  0  1
## 58 45
##
##  0  1
## 54 49
##
##  0  1
## 53 50
##
##  0  1
## 53 50
##
##  0  1
## 53 50
##
##  0  1
## 52 51
##
##  0  1
## 47 56
##
##  0  1
## 54 49
```

Or using `while()`:

```
i <- 2
while(i < 15) {
  print(table(data[,i]))
  i <- i+1
}
```

```
##
##  0  1
## 62 41
```

```
##
##  0  1
## 61 42
##
##  0  1
## 60 43
##
##  0  1
## 61 42
##
##  0  1
## 60 43
##
##  0  1
## 59 44
##
##  0  1
## 58 45
##
##  0  1
## 54 49
##
##  0  1
## 53 50
##
##  0  1
## 53 50
##
##  0  1
## 53 50
##
##  0  1
## 52 51
##
##  0  1
## 47 56
```

Saving our work:

We can save our work by using `save.image()` or `save()` to be used later:

```
save.image("all.RData") # saves all objects in the environment
save(reg1, file = "reg.RData") # save specific objects
```