Lab 2 - Basics in R, part 2

Systems modelling and data analysis
2017/2018

1 Reading and writing data files

1. Run RStudio
2. Set your Working Directory using the setwd() command.
3. Download, extract and then open the .csv file using the read.csv() function from the file car-speeds.csv.zip. Display the first six rows of the carSpeeds.The file car-speeds.csv contains information on the speeds at which cars of different colors were clocked in 45 mph zones in the four-corners states in the US.

```
        download.file("http://home.agh.edu.pl/~mmd/_media/dydaktyka/
as-is/car-speeds.csv.zip", "car-speeds.csv.zip")
unzip("car-speeds.csv.zip")

carSpeeds <- read.csv(file='./car-speeds.csv')
head(carSpeeds)
```

4. Replace the 'Blue' color with 'Green' in the $Color column using ifelse() function. What happened? Why? Note your answer.

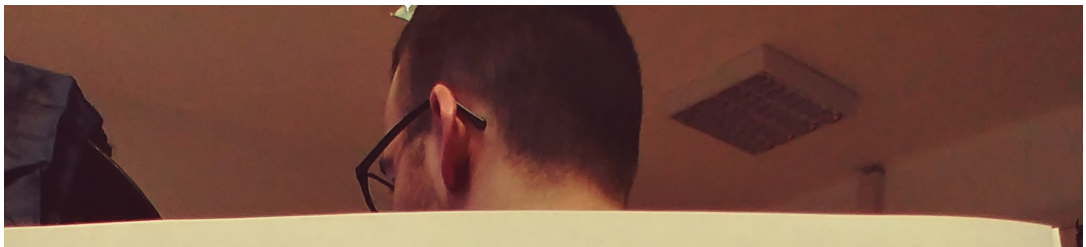
```
carSpeeds$Color <- ifelse(carSpeeds$Color=='Blue', 'Green',
                          carSpeeds$Color)
carSpeeds$Color
```

5. Now load the dataset, but set the StringsAsFactor argument to false. Then, use the ifelse() function again to replace the 'Blue' color with 'Green' in the $Color column. What happened this time and why?

```
carSpeeds <- read.csv(file='./car-speeds.csv', stringsAsFactors=FALSE)
```

6. Use the as.is argument to import colors of cars as strings and the names of the states as factors. To do that, use the as.is argument.

```
carSpeeds <- read.csv(file='./car-speeds.csv', as.is = 1)

carSpeeds$Color <- ifelse(carSpeeds$Color=='Blue', 'Green',
                          carSpeeds$Color)
carSpeeds$Color

carSpeeds$State <- ifelse(carSpeeds$State=='Arizona','Ohio',
                          carSpeeds$State)
carSpeeds$State
```



7. Check for the unique values in the $Color column of the dataset to see if there are any whitespaces. Use the unique() function.

```
unique(carSpeeds$Color)
```

8. Import the data using the strip.white argument, then check again the unique values in the $Color column. Any changes? What would happen if we did not set the sep argument in this example?

```
carSpeeds <- read.csv(file='./car-speeds.csv',stringsAsFactors=FALSE,
             strip.white=TRUE,sep=',')
```

9. Export the dataset with replaced 'Blue' with 'Green' in the $Color column by using the write.csv() function. Note that this function requires a minimum of two arguments: the data to be saved and the name of the output file.

```
write.csv(carSpeeds, file='./car-speeds-cleaned.csv')
```

Open this new file and see what it contains.

10. Correct the output file by setting row.names argument to false.

```
write.csv(carSpeeds, file='data/car-speeds-cleaned.csv',
          row.names=FALSE)
```

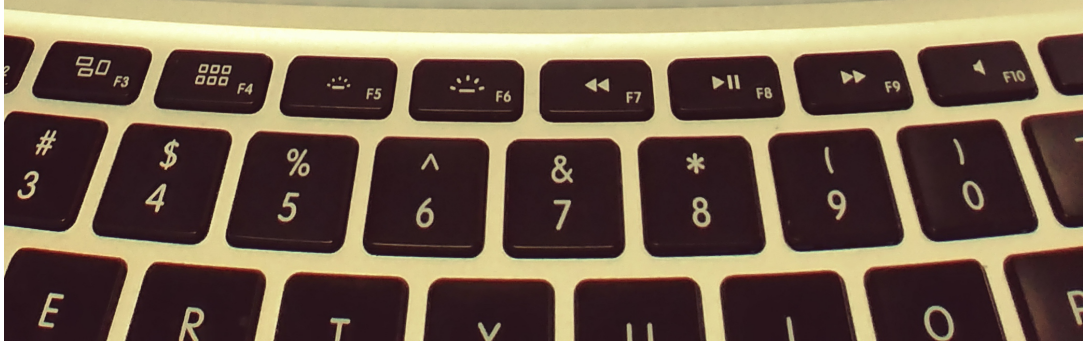
11. Replace the speed in the 3rd row of carSpeeds with NA value by using an index. Then write the new .csv file with setting the NA value to -9999 .

```
carSpeeds$Speed[3]<-NA
head(carSpeeds)

write.csv(carSpeeds, file='data/car-speeds-cleaned.csv',
          row.names=FALSE, na= '-9999')
```

2 Vectorized operations

1. Create two vectors: a and b with values 1:10 and then add pairs of numbers contained in this two vectors.

```
a <- 1:10
b <- 1:10

res <- a + b
res
```



2. Create a vector c with 1:5 values, then add this vector to vector a. What happened and why?

```
c <- 1:5

res2 <- a + c
res2
```

3. Multiply every element of vector a by 5.

```
d <- 5
a * d
```

4. Try to add a vector a and e. Notice that this time the length of the longer object is not a multiple of the shorter object length.

```
e <- 1:7
a + e
```

3 Subsetting data

1. Create a vector animal as in the example below. Then subset this vector with first three and then last three characters.

```
animal <- c("m", "o", "n", "k", "e", "y")
animal[1:3]
animal[4:6]
```

2. Answer the questions:

   (a) If the first four characters are selected using the slice animal[1:4], how can we obtain the first four characters in reverse order?

   (b) What is animal[-1]? What is animal[-4]? Given those answers, explain what animal[-1:-4] does.

   (c) Use a slice of animal to create a new character vector that spells the word "eon", i.e. c("e", "o", "n").

3. Download, extract and then read the .csv file into 'dat' variable from the file inflammation-01.csv.zip.

```
download.file("http://home.agh.edu.pl/~mmd/_media/dydaktyka/
as-is/inflammation-01.csv.zip", "inflammation-01.csv.zip")
unzip("inflammation-01.csv.zip")

dat <- read.csv(file = "./inflammation-01.csv", header = FALSE)
```

4. Check what type and shape of thing 'dat' is (use class() and dim() functions).
5. Pick column 10 and 20 from rows 1, 3, 5.

```
dat[c(1, 3, 5), c(10, 20)]
```

6. Select the first ten columns of values for the first four rows.

```
dat[1:4, 1:10]
```

7. Select the first ten columns of rows 5 to 10.

```
dat[5:10, 1:10]
```

8. Select all rows from column 16-18.

```
dat[, 16:18]
```

9. Check the maximum inflammation value for patient 1 and patient 2. The check the minimum inflammation on day 7.

```
patient_1 <- dat[1, ]
max(patient_1)

max(dat[2, ])

min(dat[, 7])
```

4 Control structures

1. Analyze how this for loop works:

```
v <- c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j")
for(i in v) {
print(i)
}
for(i in 1:10) {
print(v[i])
}
for(i in seq_along(v)) {
print(v[i])
}
```

2. Analyze how this while loop works:

```
i <- 1
while(i <= 10) {
print(i)
i <- i + 1
}
```

3. Analyze how below repeat loop works. Pay attention to the if-else, next and break structures.

```
i <- 1
repeat {
          i <- i + 1
          if(i == 11) {
                break;
          }
     else if(i == 5) {
     next;
     }
     else {
     print(i)
     }
}
```

5 Exercise

1. Suppose you want to determine the maximum inflammation for patient 5 across days three to seven. To do this you would extract the relevant subset from the data frame and calculate the maximum value. Which of the following lines of R code gives the correct answer?

   (a) max(dat[5,])

   (b) max(dat[3:7, 5])

   (c) max(dat[5, 3:7])

   (d) max(dat[5, 3, 7])

2. Using the inflammation data frame dat from above: Let's pretend there was something wrong with the instrument on the first five days for every second patient (#2, 4, 6, etc.), which resulted in the measurements being twice as large as they should be.

   (a) Write a vector containing each affected patient (hint: ? seq)

   (b) Create a new data frame with in which you halve the first five days' values in only those patients

   (c) Print out the corrected data frame to check that your code has fixed the problem