# Data Programming Exam

Please reply to the following questions in an R script called "surname_name.R" (e.g., Mario Rossi will return a file named "rossi_mario.R") and send it by mail to andrea.spano@quantide.com

Please comment your answers using the symbol: #, before the comments (e.g., # my comment).

## Exercise 1

a. Create a vector, named `vec`, containing the following values:
   1, 5, 12, 14, 6, 78, 68, 34, 34, 32, 56, 75

```
vec <- c(1, 5, 12, 14, 6, 78, 68, 34, 34, 32, 56, 75)
```

b. Select the 3-rd element of `vec`.

```
vec[3]
```

```
## [1] 12
```

c. Select all elements of `vec` apart from the 1st.

```
vec[-1]
```

```
##  [1]  5 12 14  6 78 68 34 34 32 56 75
```

## Exercise 2

a. Generate a matrix, named `mat`, with 3 rows and 5 columns containig numbers from 1 to 15.

```
mat <- matrix(1:15, nrow = 3, ncol = 5)
```

b. Select 2-nd and 3-rd rows and 1-st and 3-rd columns of `mat`.

```
mat[c(2,3), c(1,3)]
```

```
##      [,1] [,2]
## [1,]    2    8
## [2,]    3    9
```

## Exercise 3

Given the following list, named `this_list`:

```r
this_list <- list(numbers = c(2,3,5,6,7), letters = c("z", "x", "y", "t"))
this_list
```

```
## $numbers
## [1] 2 3 5 6 7
##
## $letters
## [1] "z" "x" "y" "t"
```

a. Extract the element named `letters` of `this_list` by using the `$` operator.

```r
this_list$letters
```

```
## [1] "z" "x" "y" "t"
```

b. Extract the first element of `this_list` by using double square brackets.

```r
this_list[[1]]
```

```
## [1] 2 3 5 6 7
```

## Exercise 4

a. Generate a data frame, named `df`, corresponding to:

```
df <- data.frame(country = c("Italy", "France", "China", "Japan", "Libya",
                             "Cameroon"),
                 population = c(59801004, 64668129, 1382323332, 126323715,
                                6330159, 23924407),
                 continent = c("Europe", "Europe", "Asia", "Asia",
                               "Africa", "Africa"
                              ), stringsAsFactors = FALSE)
print(df, row.names = FALSE)
```

```
  country population continent
    Italy   59801004    Europe
   France   64668129    Europe
    China 1382323332      Asia
    Japan  126323715      Asia
    Libya    6330159    Africa
 Cameroon   23924407    Africa
```

Use `data.frame()` function and remember to maintain character vectors as they are, specifiyng `stringsAsFactors = FALSE`.

b. Convert `continent` variable of `df` as a factor with levels: “Europe”, “Asia” and “Africa”. Use `factor()` function.

```
df$continent <- factor(df$continent, levels = c("Europe", "Asia", "Africa"))
```

## Exercise 5

a. Import the file *2008.csv* into a data frame named `flights` by using the `read.table()` command. Remember to specify `stringsAsFactors` as `FALSE` in order to avoid importing character columns as factors.
Before importing be sure about:

- column names in the first row
- the field separator
- the decimal separator

```
flights <- read.table(file = "/home/veronica/dev/qtraining/010-rbase/data/2008.csv", header = TRUE, sep
```

This dataset contains information about flight arrival and departure details for all commercial flights within the USA in 2008.

Load `dplyr` library:

```
require(dplyr)
```

b. Convert `flights` data frame to a `tbl_df` using `tbl_df()` function.

```r
flights <- flights %>% tbl_df()
```

c. Starting from `flights` data frame, select `ArrDelay` and `Dest` variables and filter the records for which `ArrDelay` variable is greater than 120.

```r
flights %>% select(ArrDelay, Dest) %>% filter(ArrDelay > 120)
```

```
## Source: local data frame [153,537 x 2]
##
##    ArrDelay   Dest
##       (int) (fctr)
## 1       304    HOU
## 2       210    BUR
## 3       254    IND
## 4       268    SFO
## 5       139    SFO
## 6       176    SFO
## 7       203    SFO
## 8       129    SFO
## 9       131    SFO
## 10      184    SFO
## ..      ...    ...
```

d. Starting from `flights` data frame, compute the mean delay at departure (`DepDelay` variable) groupig by `Origin` variable. Remember to add `na.rm=TRUE option` to mean computation.

```r
flights %>% group_by(Origin) %>% summarise(mean=mean(DepDelay, na.rm=TRUE))
```

```
## Source: local data frame [303 x 2]
##
##    Origin      mean
##    (fctr)     (dbl)
## 1     ABE  8.012224
## 2     ABI  6.880397
## 3     ABQ  7.440111
## 4     ABY 10.962894
## 5     ACK 29.854415
## 6     ACT  5.207400
## 7     ACV 15.202119
## 8     ACY 18.548673
## 9     ADK 13.652174
## 10    ADQ  5.739065
## ..    ...       ...
```

## Exercise 6

Load `mtcars` dataset in this way:

```
data("mtcars")
```

`mtcars` data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

```
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

To achieve more information about `mtcars` dataset type `?mtcars` on R console.

Load `ggplot2` library:

```
require(ggplot2)
```

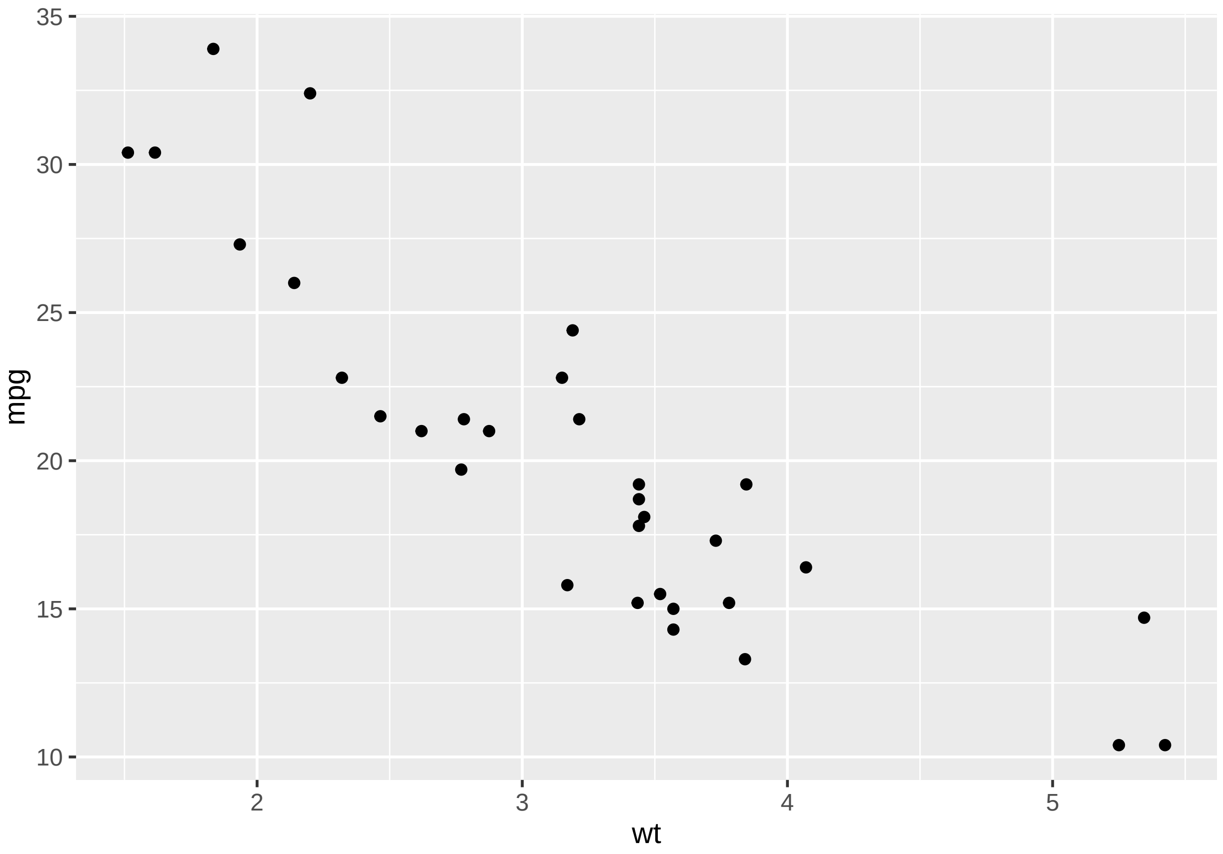
a. Calculate the number of rows and columns of the `mtcars` dataset.

```
dim(mtcars) # or nrow(mtcars) and ncol(mtcars)
```

```
## [1] 32 11
```

b. Build a scatterplot to analyze the relationship between `mpg` and `wt` variables. Use `ggplot()` and `geom_point()` functions.

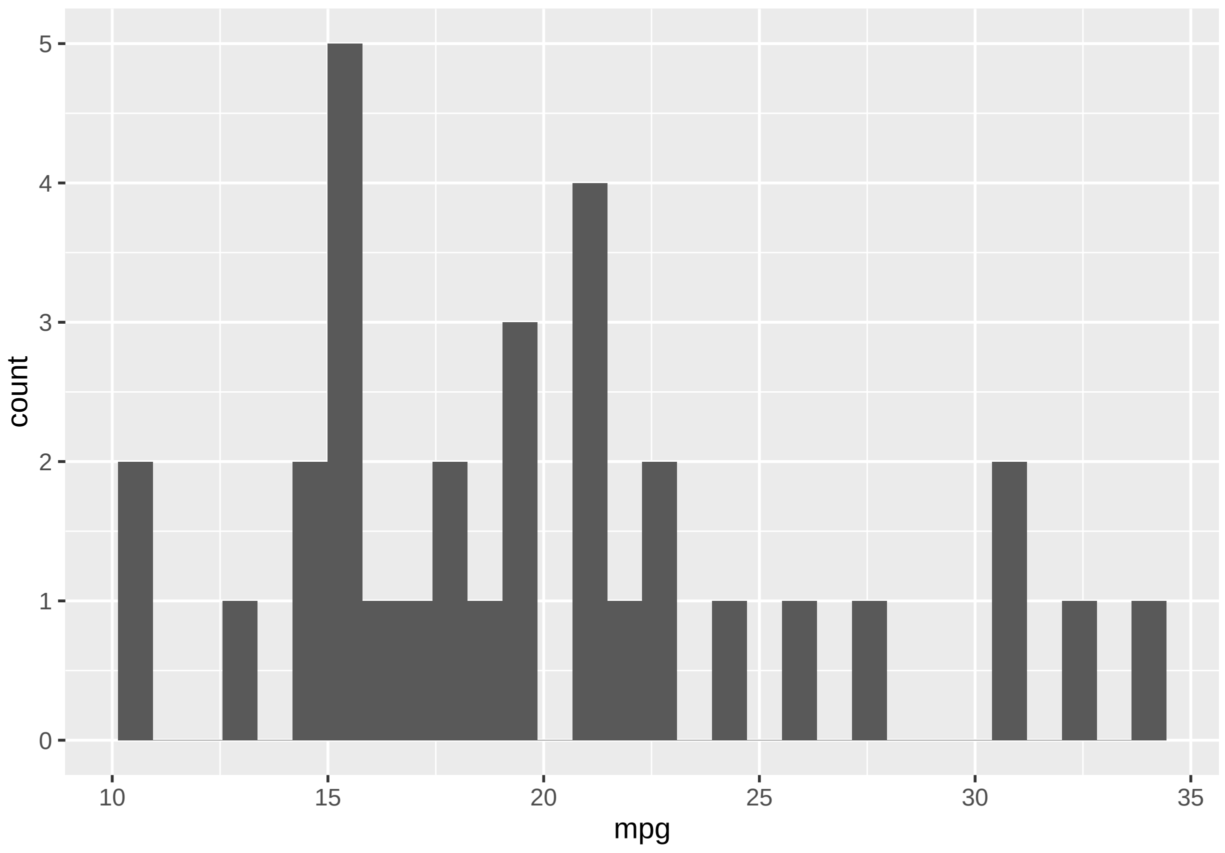
```
ggp <- ggplot(data = mtcars, mapping = aes(x=wt, y=mpg)) +
  geom_point()
print(ggp)
```



c. Represent the distribution of `mpg` variable with an histogram. Use `ggplot()` and `geom_histogram()` functions.

```
ggp <- ggplot(data = mtcars, mapping = aes(x=mpg)) +
  geom_histogram()
print(ggp)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


count
mpg
0
1
2
3
4
5
10
15
20
25
30
35