# Homework 1 - Data Mining

*Anna Takacs*

*9/23/2019*

## Exercise 1

*In what ways does a matrix differ from a array in R? Use R code to create one of each.*

The difference between matrices and arrays is that arrays can hold multidimensional rectangular data. Rectangular means that for each dimension, the lenth of the data need to be the same (e.g. each column has the same length). Matrices are only two dimensional items.

The following is a three dimensional matrix: [1]

```
## , , First
##
##       egy kettő három
## one     1     4     7
## two     2     5     8
## three   3     6     9
##
## , , Second
##
##       egy kettő három
## one    10    13    16
## two    11    14    17
## three  12    15    18
##
## , , Third
##
##       egy kettő három
## one    19     1     4
## two    20     2     5
## three  21     3     6
```

And a two dimensional matrix:

```
##       one two three
## egy     1   4     7
## kettő   2   5     8
## három   3   6     9
```

## Exercise 2

Simplify and make the following code consistent with the Hadley Wickham R style guide.

[1] Columns are named in Hungarian.

```r
a <- c(5, -2, 3, -4, 1, 2)
b <- a*-1
b [b > 0]
```

```
## [1] 2 4
```

```r
num = 1
mycondition <- round(runif(1, 0, 1))
if( mycondition ) {num <- num + 1
print(paste("num =", num, sep = " "))} else {print("false")}
```

```
## [1] "false"
```

Firstly, create an integer sequence from 1 to 50. To see if a number if even, use the module function. And lastly, subset the sequence of integers depending on whether they are divisible by 2.

```r
y <- 1:50
even <- y %% 2 == 0
y = y [even]
```

The following code creates a matrix and the mean of the values in the matrix divided by the number of elements it has.

```r
x <- matrix(c(23, 34, 35, 6, 87, 39, 21, 14, 99), nrow = 3)
df <- as.data.frame(x)
names(df) <- c("percentage_score_on_reading_test",
 "percentage.score.on.math.test",
  "percentage-score-on-writing-test")
my_mean <- function(x) {sum(x) / length(x)}
```

## Exercise 3

*Look at the spredsheet-like representation of the data and write an R code that extracts the data for Ohio on the variables 'population' and 'frost' in three different ways.*

1st way of extracting:

```r
state.x77["Ohio", c("Population", "Frost")]
```

```
## Population      Frost
##      10735        124
```

2nd way of extracting:

```r
state.x77["Ohio", -c(2:6, 8)]
```

```
## Population      Frost
##      10735        124
```

3rd way of subtracting information:

I transform the tibble into a data frame. In this way, I can tell R STudio to extract the values from the 35th row and 1st and 7th columns.

```
class(as.data.frame(state.x77))
```

```
## [1] "data.frame"
```

```
state.x77[35, c(1, 7)]
```

```
## Population      Frost
##      10735        124
```

## Exercise 4

*Replace the Wind variable with windspeed measured in kilometers per hour.*

```
# I multiply the value of the wind column in the dataset by 1.609344 so that the values are expressed i
airquality$Wind <- airquality$Wind * 1.609344
# To be able to distingues between the new and old variables, I rename the new km/h variable to windspe
names(airquality) <- c("Ozone", "Solar.R", "Windspeed", "Temp", "Month", "Day")
print(head(airquality))
```

```
##   Ozone Solar.R Windspeed Temp Month Day
## 1    41     190  11.90915   67     5   1
## 2    36     118  12.87475   72     5   2
## 3    12     149  20.27773   74     5   3
## 4    18     313  18.50746   62     5   4
## 5    NA      NA  23.01362   56     5   5
## 6    28      NA  23.97923   66     5   6
```

## Exercise 5

I brings the data.frame called turnout into R's memory and observe its values.

```
##   year    VEP    VAP total ANES felons noncit overseas osvoters
## 1 1980 159635 164445 86515   71    802   5756     1803       NA
## 2 1982 160467 166028 67616   60    960   6641     1982       NA
## 3 1984 167702 173995 92653   74   1165   7482     2361       NA
## 4 1986 170396 177922 64991   53   1367   8362     2216       NA
## 5 1988 173579 181955 91595   70   1594   9280     2257       NA
## 6 1990 176629 186159 67859   47   1901  10239     2659       NA
```

To calculate the turnout rate based on the voting age population, I add the voting age population and the number of eligible oversees voters and multiply it with the estimated turnout rate to get a value for the turnout rate. I repeat the same process with the voting eligible population. I print out the values for both for the corresponding years.

```
##  [1] 11803608 10080600 13050344  9547314 12894840  8874446 14489700
##  [8] 11059272 14783595 10829000 15589880 13563740 17263246 18395832
```

```
##  [1] 11462098  9746940 12584662  9148436 12308520  8426536 13655550
##  [8] 10351712 13785758 10054564 14400564 12504780 15965565 17026308
```

I observe that the turnout rate is lower when I use the voting eligible value, which is intuitive coming from the fact that not everyone is eligible to vote who are above the voting age, but everyone who is eligible to who must be above the voting age.

This concludes the solutions for homework 1 for the Data Mining class.