Exercise 1

Open Source Tools for Intelligent Systems
Summer semester 2018

The exercise should be submitted by 11/8 before 23:59. Submit an R script file "Ex1.R", which includes both the necessary code and the verbal answers as comments (using # at the beginning of the line).

Q1.

a) Write a function called **mean_and_var(x)** that calculates the mean and variance (var) of x, where x can be a numeric vector or matrix with at least two elements (use **stopifnot** function to abort the function if this is not the case). Besides x, the function can receive a second parameter called use.sum, which is set to TRUE by default. If use.sum is FALSE then the function should calculate the mean and var using only **for** loops. If use.sum is TRUE the function should do the calculation using base R **sum()** function. The function should return a list object with two components named "mean" and "var". You can use base R **mean()** and **var()** functions to test your function and make sure it returns the right results. Note that for a matrix x, **var(x)** returns a covariance matrix not the variance, so if x is a matrix you should convert it to a vector first and then call var or use **var(as.vector(x))**

b) Compare the time it takes for the function to compute the mean and variance of a 1000x10000 matrix when use.sum is set to FALSE and when it is set to TRUE. Use the **system.time()** function which measures the run time of the code given to it (compare the results of **system.time(mean_and_var(x,use.sum=F))["elapsed"]** to **system.time(mean_and_var(x,use.sum=T))["elapsed"]** where x is the same 1000x10000 matrix). How much faster did the function run using **sum()** compared to using **for** loops?

Q2.

a) Create a matrix with 200 rows and 5 columns filled with random numbers from a normal distribution with mean=5 and var=100. (What is the corresponding standard deviation sd?) Use **rnorm()** to generate the random numbers. Use **apply()** to calculate the mean and var of each column in the matrix (you can use the function **mean_and_var()** you created in the previous question as the function for **apply()** or use **apply()** twice - with the base R **mean()** and **var()** functions).

b) Repeat what you did in (a) for a matrix with 20000 rows and 5 columns. What can you see from the results of the two sections?

c) What can you do in order to be able to compare your results for this question with the results of your classmate?

Q3.

a) Load the file '2017 World Happiness Report.csv' into a data frame (if you are interested, see https://www.kaggle.com/unsdsn/world-happiness or http://worldhappiness.report/ to read more about this report). How many observations are in the data? How many variables? Remove columns (3, 5, 6, 13) corresponding to variables ("Happiness.Rank", "Whisker.high", "Whisker.low", "Dystopia Residual") from the data frame. Rename the variables ("Happiness.score", "Economy..GDP.per.Capita.", "Health..Life.Expectancy.", "Trust..Government.Corruption.") to ("Happiness", "Economy", "Health", "Trust") using **colnames()**. In the modified data frame what are the variables and what is the type of each one? How many different Regions are in the data? (You can use the **levels()** function to answer this). What are the min/max/mean Happiness score in the data?

b) Using ggplot2, create a scatter plot of each of the last 6 variables in the data frame against the Happiness variable (6 scatter plots altogether). Set the color of the points according to their Region. Add a regression line to each plot using **geom_smooth** with the method parameter set to **'lm'**. What can you tell from the plots?

c) Plot again the scatter plot the Economy vs. Happiness with the colors of the points according to their Region. This time add to the plot a regression line for each of the Regions (using one call of **geom_smooth**). Make sure the confidence intervals for the regression lines are not shown.

d) Using ggplot2, create a bar plot showing the number of countries in each Region. Have the bars displayed horizontally instead of vertically. Why does flipping the coordinates improves the plot in this case?

e) Using ggplot2, create a box plot showing the distribution of Happiness for each Region. Have the boxplot displayed horizontally and ordered according to the mean Happiness value for each Region. Which Region has the most variability in Happiness? Which Region has the least variability? What is the problem with the answer to this last question?

f) (* Optional / Bonus) Install and load the package **GGally**. Call the **ggpairs()** function with a subset of the data frame that includes only the last 7 variables. What does this function plot? What does it show on the diagonal? Above the diagonal? Below the diagonal? What can you learn from this plot that you couldn't tell from the plots you created in section (b)?