Data mining: Practice 1

S. Nõmm

[1] Department of Computer Science, Tallinn University of Technology

February 2, 2016

R and R studio

- Check if your computer is running latest version of Java
- You may download R from `https://www.r-project.org/`
- It is advisable to download R studio as well (makes your life easer) `https://www.rstudio.com/products/rstudio/`.
- once R and Rstudio are installed you may try to follow Practice 1 from a course wiki download files with R scripts
  - `Demo1_correlation_regression_otliers.R`
  - `PCA_1.R`
- and data files
  - `demoSetD2.zip` (unzip the file to get demoSetD2.xls
  - `variablesXYZ`

Exercise 1

- Open `Demo1_correlation_regression_otliers.R`. This script demonstrates:
  - Import of the data from .xls file: On this step you will need to add "rJava" and "xlsx". In order to install the packages type in console `install.packages("packageName")`. Once packages are installed in your script add `library(xlsx)` this activates the library.
    `setD<-read.xlsx("C:/Path/fname",1)` reads numeric data from the Sheet 1 into the numeric array setD.
  - Drawing simple plots: `plot(setD[,2],setD[,1])` plots scatter plot whereas the second column of the matrix setD is treated as independent variable and the first column as dependent variable. Note! notation `(setD[,2])` indicates the second column.

Exercise 1 (continued)

- Computing some measures of statistics: `corCoef<-cor(setD[,2],setD[,1])` computes linear correlation coefficient between the first and second columns of the matrix setD.
- Finding coefficients of the linear regression model: `model1<-lm(trainingSetD[,1]~trainingSetD[,2])` builds the model where `trainingSetD[,1]` is the dependent variable and `trainingSetD[,2]` independent. `C=summary((model1)$coefficients)` extracts the values of the coefficient and intercept.
- Finally model validation is performed.
- Each line of the file `Demo1_correlation_regression_otliers.R` is supplied with explanation or comment.

Exercise 2

This exercise illustrates computations necessary to perform PCA (principal component analysis). The data is in native "R" fromat `variablesXYZ` and the script is `PCA_1.R`.

- On the first step we celar the environment as usually.
- Loading native format does not require any external libraries `load(file="C:/Path/fname")`
- We will use some libraries for 3D plotting "sctterplot3D", "car" and "rgl". Instal those packages the same way as in previous example.
- "R" possesses some useful functions like "length" which provide you with the possibility to determine the length of the vector if necessary
- Commented part of the file allows you to position and draw some histograms.

Exercise 2 (continued)

- Followed by computations of correlation coefficients (see previous example) and standard deviations `sx<-sd(x)` computs standard deviation of x
- In many cases it is necessary to center the data (subtract mean).
- `mean_x<-mean(x)` computes the mean value of x the you may subtract it
- `D<-cbind(x,y,z)` combines vectors x, y and z into the matrix $\mathcal{D}$
- `cov_D<-cov(D)` computes covariance matrix of $\mathcal{D}$
- `eig_cov_D<-eigen(cov_D)` computes eigenvalues and eigenvectors
- `rotated_D<-D%*%eig_cov_D$vectors` computes $\mathcal{D}'$

Exercise 2 (continued)

- You may now analyze covariances between the columns of matrix $\mathcal{D}$ and check variances
- `open3d` opens new window for 3D plot
- `scatter3d` plots 3D scatter.