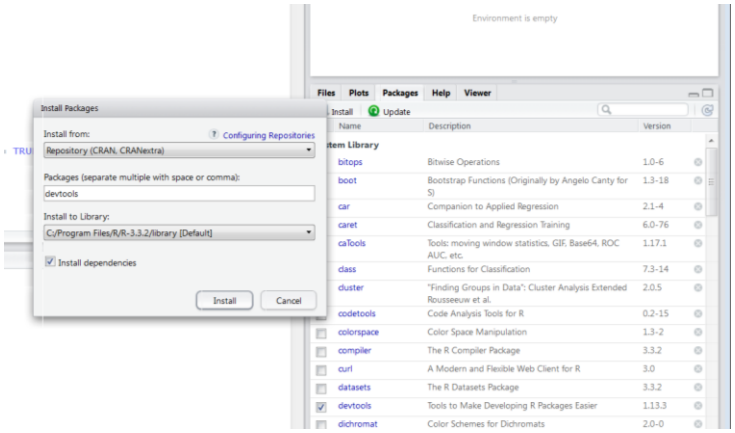
TOWARDS DATA MINING

EXERCISE 7. NORMALIZATION, TRANSFORMATIONS, DATA REDUCTION AND DISTRIBUTIONS

In this exercise, finish the following tasks. Then, answer the Exercise 7 questions in Moodle and submit your quiz. In this exercise we will use R, but you can find similar tools for Matlab as well. First, set your working directory, where you saved the required material from Moodle (in RStudio select Session -> Set Working Directory -> Choose Directory… and find your folder and click Select Folder).



1. **In this task, you study the effect of normalization with principal components analysis.** Open the script *sript_PCA_E7.R* from the Files and run the first row of the code.
   Explore the matrix X with

   ```
   summary(X)
   ```

   and prepare to answer a question about the content of the data. Then run the rest of the script.

   Explore, what the code did do. Note, that the script produces a plot and prints some results for the PCA. Pay attention to the importance of each principal component by observing the *Proportion of Variance* and *Cumulative Proportion* for PC1-4.

   The original scipt was run with a raw data. Now, modify the script by adding normalisation for the data matrix *X*. You can do it separately with preferred method or with "TRUE" additions to centering and scaling in principal component calculation in the beginning of the script on line 2

   ```
   x.pca<-prcomp(X,center=TRUE,scale=TRUE)
   ```

   which performs standardization by moving the data to center and by scaling it with standard deviation. Run the line and the following printing and plotting lines for PCA results.

   Again, analyse the new result, and especially, how the standardization improved the results of the principal component analysis.

2. **In this task, you study the effect of data reduction.** First, you'll train a neural network model with the original data and then compare the model results to one trained with the data that has treated with PCA to reduce the number of variables.

   Open the script *script_NN_E7.R* and observe it. What normalization method has been used in this task.? How does it differ from the method used in the previous task? Run the code. While running, observe the script further. The neural network model *model_nn* is trained with the normalized data and then the predicted values are produced. The model performance is analysed with cor() that calculates the correlation of the predicted value to the original one, which will be printed after the training. You will need this value later.

   Note that in real life, a more robust method would be the calculation of Root Mean Squared Error (RMSE)! Correlation was selected only, because, it is easy to interpret. You should not use correlation for model selection!

   Study the input data in *X* with str(X). As you can see, all of the data is not continuous, but instead, you have some variables that have only values 0/1. As you remember from the lecture, PCA can be performed only for continuous variables and thus, variables containing only 0/1 values have been left out at the next step. Remember to check the result of the *model_nn*!

   Next, open the script *script_NNPCA_E7.R* and add the numbers of columns with continuous variables to c() on line 11 (only them will be selected to the PCA). Then, run the code. While running, observe the script. The amount of continuous variables in this dataset is 7, but in the script, we select only the first PC to the neural network input together with the two-class 0/1-variables. What can you say about the performance of the model? Plot the results in a scatter plot:

   ```
   plot(y_spca.predict,y)
   ```

   Compare the range of the original value *y* and the predicted value. Because, we scaled *y* for Neural Network training, the predicted value should be scaled back to the original scale, when utilizing the predictions (especially important with RMSE). That can be done with

   ```
   y_pca.predict<- (y_spca.predict*(maxy-miny))+miny
   ```

   Write summary(x.pca) to the command line in order to observe the summary of the PCA, and especially, the *Cumulative Proportion* of the variance. Try different values for *N* in the script and try to figure out, how many principal components you need in order to achieve the performance you had in the beginning with the original data *X*.

   Choose the best number of Principal Components you would select considering both the efficiency in data reduction and the accuracy of the model. Plot the scatter plot of the results and attach the plot to your report together with the comments, why this model is the best. Note, that there is not only one correct answer, but the reasons that motivated your selection are more important.