# 1. GENERAL INSTRUCTIONS

The following sequence of function calls allows accomplishing all tasks required by the project:

1) *check_if_exists()*
2) *original_df <- readData("UCI HAR Dataset", FALSE)*
3) *subset_df <- extractMeanStdColumns(original_df)*
4) *summarized_df <- summarizedData(subset_df)*

All these functions have to be run in the indicated sequence order to obtain the file "output.txt" as it has been uploaded in the coursera web page. Also notice that the input for the "extractMeanStdColumns()" function has to be the output of "readData()" function and that the input for "summarizedData()" has to be the output of "extractMeanStdColumns()" function. Instead of creating an overall function containing this sequence with the correct inputs, I prefer to call each function separately from the command line in order to have access to outputs from readData() function and from extractMeanStdColumns(). Additionally, in this way, final user is provided with the flexibility to decide whether include the Inertial Signals or not and, in case she does, she can access to this data as an output from the readData() function.

If the UCI HAR Dataset directory does not exist in the local computer, an internet connection is needed for the *check_if_exists()* function to programmatically download the source files and create the UCI HAR Data set folder in the local working directory.

# 2. STRUCTURE OF “run_analysis.R” FILE

All functions are defined in the attached “run_analysis.R” and described, at an high level, in this read me file.

Besides the 4 above mentioned main functions, the “run_analysis.R” also contains other 3 auxiliary functions: “firstLevelData()”, “readInertialData()” and “replaceActivityName()”. The first 2 are called by the “readData()” function. The third one is called by the “firstLevelData()”. The resulting flow in the function stack is as follow:

1. check_if_exixts()
2. readData()
   - 2.1. firstLevelData()
     - 2.1.1. replaceActivityName()
   - 2.2. readInertialData()
3. extractMeanStdColumns()
4. summarizedData()

The below scheme gives a high level description of what each of the above functions does. More details for each function are provided later in this read me file.

| **Function name** | **Function description** |
|---|---|
| check_if_exixts() | Checks if all necessary data is stored in the local computer and creates the necessary environment in case it is needed |
| readData() | Reads data from the UCI HAR Dataset directory and build the overall dataframe according to the requirements specified in the arguments (meaning that data from the Inertial Signals folder are loaded only upon request) |
| firstLevelData() | Builds a dataframe containing the subject, activity and main sensorial data (meaning not the Inertial Signal data) either from test or from train folder depending on the argument settings. |
| readInertialData() | Columnwise attaches data from Inertial Signal folder to the input dataframe and names columns in a descriptive way |
| replaceActivityName() | Replaces numeric activity indicators with descriptive string labels according to the table in the |

| | |
|---|---|
| | “features.txt” file |
| extractMeanStdColumns() | Extracts from the overall dataframe those columns containing data on either the mean or standard deviation of the variables of interest. |
| summarizedData() | Collapse the input dataframe into a summarizing 30x6 dataframe grouped by subject and carried out activity calculating the mean and standard deviation of each variable of input dataframe |

# 3. TASKS REQUIRED BY THE PROJECT

## 3.1 Preliminary step: “check if it exists”

The *“chech_if_exists()”* function checks whether the "UCI HAR Dataset” directory exists in the working directory.

If the directory exists, the function just quits without doing anything since it assumes that the existing “UCI HAR Dataset” directory already contains all the files and sub directories as when recently downloaded and unzipped.

If the "UCI HAR Dataset” does not exist, the *“chech_if_exists()”* downloads the source zipped file from the link provided in the project instructions:
“https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip”
Once downloading is completed, the function unzips the downloaded file in the working directory and creates the “UCI HAR DATASET” folder with all files and subfolders needed for the project.

## 3.2 Merges the training and the test sets to create one data set.

The overall data frame is build by calling the function *“readData()”* as follows:

*original_df-readData(main_folder, inertial)*

where both arguments are optional:

-*main_folder* by default is set to "UCI HAR Dataset”, that is to say the folder name containing all the files and subfolders needed for the project
-*inertial* is an logical argument that by default is set to FALSE and indicates whether files within sub folders “Inertial Signals” have to be loaded.

The output of *“readData()”* is a data frame containing the 10.299 observations from both “train” and “test” data “rbinded” together. As explained in the next, the number of variables (columns) of the output dataframe depends on the “inertial” argument setting.

Regardless the inertial argument setting, in the output data frame there will be 2 columns named “subject” and “activity” which respectively identify both the person being observed and the activity carried out by the observed person. These data are loaded from the files

- “subject_train.txt”,
- “y_train.txt”,

and

- “subject_test.txt”
- “y_test.txt”

stored in the “train” and “test” subdirectories of “UCI HAR Dataset” main directory.

In order to track if a given row comes from a train or a test subject, the column “train” is added to the resulting dataframe: in those rows with data from a “test” subject, this field is set to 0; in those rows with data from a “train” subject, this field is set to 1.

By leaving the “inertial” argument to FALSE, the “*readData()”* function will load exclusively subject, activity and basic sensorial information leaving out the Inertial Signals files. The output will be a 10299 observations dataframe described by 564 columns corresponding to the 3 above mentioned columns (“subject”, “activity” and “train”) and the 561 columns contained in each of “X” files in “train” and “test” directories.

If inertial argument is TRUE, the “*readData()”* function will build a dataset that will also include data from Inertial Signals files. This option is available for those who need to work on Inertial Signals. It is recommend to use this option only onto powerful computers since the output is a gigantic 10299x1716 dataframe.
Of the 1716 columns, 564 are the same as in the inertial argument= FALSE case. The additional 1152 columns are the 128 columns of each of the 9 files stored in the Inertial Signals subfolders. This 128x9=1152 columns are “cbinded” to the dataframe.

The column names for the basic sensorial data are the same as in the “feature.txt” files. On the other hand, the column names for the Inertial Signals are built as the combination of the source file name and a numeric vector 1:128. In other words, the 128 columns from (for instance) the “body_acc_x_test.txt” are cbinded to the output dataframe and named “body_acc_x.1”, “body_acc_x.2”…. “body_acc_x.128”. The same applies to all other files from Inertial Signals. It is worth to notice that data from “test” and “train” subjects are distinguished through the column “train” which in one case is set to 0 and in the other case is set to 1.

## 3.3 Extracts only the measurements on the mean and standard deviation for each measurement.

The subsetting operation is performed by calling the function:

*subset_df <- extractMeanStdColumns(original_df)*

where “*original_df*” is the output dataframe from the previously described “*readData()*” function.

The “*extractMeanStdColumns*()” function creates a new dataframe extracting from the original one the 66 columns containing mean and standard deviation information.
Please, notice that the “…meanFreq()” columns are not extracted since I believe are not required.
The output dataframe will also keep the “subject”, “activity” and “train” columns from the original data frame.

As a result, the output “*subset_df “* will be a 10.229 x 69 data frame

## 3.4 Uses descriptive activity names to name the activities in the data set

This task is accomplished through the *“replaceActivityName()”* function called while running the *“firstLevelData()”* function which in turn is a function called by *“readData()”* function.

The *“replaceActivityName()”* function, replaces numeric values appearing into the “activity” column of the main dataframe with descriptive labels according to the “activity_labels.txt” file.

## 3.5 Appropriately labels the data set with descriptive activity names.

Descriptive labels are given to each column in the building phase of the dataframe.
More in details, while running the *“readData()”* function, basic sensory columns are named accordingly to the *“features.txt”* file.
If Inertial Signals data are included, the additional 1152 columns are named through a quite descriptive codification which is the result of the combination of the file name where data is stored and the numeric vector with values 1:128.

The 128 values of each observation from (for instance) "body_acc_x" are stored in the output dataframe in 128 columns named "body_acc_x.1", "body_acc_x.2".... "body_acc_x.128"

## 3.6 Creates a second, independent tidy data set with the average of each variable for each activity and each subject.

This task is accomplished by calling the function:

*summarized_df* <- *summarizedData(subset_df)*

where "*subset_df*" is likely to be the subset dataframe as created through the "*extractMeanStdColumns()"* function described in point 3. But actually, this function can be also applied to the complete dataframe created through the "*readData()"* function described in point 1.

For each column of the input dataframe, the function aggregates data according to all activity (6) levels and all subjects (30). Therefore the output dataframe will contain 30x6=180 observations. While the original 10299 are collapsed into the 180 rows, the mean and the standard deviation of each column of the input dataframe are calculated and stored into the output dataframe.