Code Book of Final project for the Getting and Cleaning Data Course of Johns Hopkins University

Libraries Used in run_analysis.R

1. Dplyr: a really useful library which allows us to merge and summarise data frame between other good functions.

Variables

In general, each variable defined in tidy data set have the following structure:

tBodyAcc_mean_X

in regular expressions looks like

#DomainSignal#Variable_#Measure(_#Axis)?

Domain Signal	Variable	Measure	Axis
- t : time domain signal - f: frequency domain signal	- BodyAcc: Body Acceleration Signal - GravityAcc: Gravity Acceleration Signal - BodyAccJerk: Body Acceleration Jerk Signal - GravityAccJerk: Gravity Acceleration Jerk Signal - BodyGyro: Body Gyroscope signal - BodyGyroJerk: Body Gyroscope Jerk signal - BodyAccMag: Body Acceleration Magnitude (with the Euclidean norm) - GravityAccMag: Gravity Acceleration Magnitude (with the Euclidean norm) - BodyGyroMag: Body Gyroscope Magnitude (with the Euclidean norm) - BodyGyroJerkMag: Body Gyroscope Jerk Magnitude (with the Euclidean norm)	- mean: average in variable - std: deviation standard in variable	X, Y, Z: possibilities in axials

Comparison Table

Raw data set	Tidy data set	Raw data set	Tidy data set

variable	variable	variable	variable
tBodyAcc_mean()_X	tBodyAcc_mean_X	tBodyAcc_std()_Y	tBodyAcc_std_Y
tBodyAcc_mean()_Y	tBodyAcc_mean_Y	tBodyAcc_std()_Z	tBodyAcc_std_Z
tBodyAcc_mean()_Z	tBodyAcc_mean_Z	tGravityAcc_std()_X	tGravityAcc_std_X
tGravityAcc_mean()_X	tGravityAcc_mean_X	tGravityAcc_std()_Y	tGravityAcc_std_Y
tGravityAcc_mean()_Y	tGravityAcc_mean_Y	tGravityAcc_std()_Z	tGravityAcc_std_Z
tGravityAcc_mean()_Z	tGravityAcc_mean_Z	tBodyAccJerk_std()_X	tBodyAccJerk_std_X
tBodyAccJerk_mean()_X	tBodyAccJerk_mean_X	tBodyAccJerk_std()_Y	tBodyAccJerk_std_Y
tBodyAccJerk_mean()_Y	tBodyAccJerk_mean_Y	tBodyAccJerk_std()_Z	tBodyAccJerk_std_Z
tBodyAccJerk_mean()_Z	tBodyAccJerk_mean_Z	tBodyGyro_std()_X	tBodyGyro_std_X
tBodyGyro_mean()_X	tBodyGyro_mean_X	tBodyGyro_std()_Y	tBodyGyro_std_Y
tBodyGyro_mean()_Y	tBodyGyro_mean_Y	tBodyGyro_std()_Z	tBodyGyro_std_Z
tBodyGyro_mean()_Z	tBodyGyro_mean_Z	tBodyGyroJerk_std()_X	tBodyGyroJerk_std_X
tBodyGyroJerk_mean()_X	tBodyGyroJerk_mean_X	tBodyGyroJerk_std()_Y	tBodyGyroJerk_std_Y
tBodyGyroJerk_mean()_Y	tBodyGyroJerk_mean_Y	tBodyGyroJerk_std()_Z	tBodyGyroJerk_std_Z
tBodyGyroJerk_mean()_Z	tBodyGyroJerk_mean_Z	tBodyAccMag_std()	tBodyAccMag_std
tBodyAccMag_mean()	tBodyAccMag_mean	tGravityAccMag_std()	tGravityAccMag_std
tGravityAccMag_mean()	tGravityAccMag_mean	tBodyAccJerkMag_std()	tBodyAccJerkMag_std
tBodyAccJerkMag_mean()	tBodyAccJerkMag_mean	tBodyGyroMag_std()	tBodyGyroMag_std
tBodyGyroMag_mean()	tBodyGyroMag_mean	tBodyGyroJerkMag_std()	tBodyGyroJerkMag_std
tBodyGyroJerkMag_mean()	tBodyGyroJerkMag_mean	fBodyAcc_std()_X	fBodyAcc_std_X
fBodyAcc_mean()_X	fBodyAcc_mean_X	fBodyAcc_std()_Y	fBodyAcc_std_Y
fBodyAcc_mean()_Y	fBodyAcc_mean_Y	fBodyAcc_std()_Z	fBodyAcc_std_Z
fBodyAcc_mean()_Z	fBodyAcc_mean_Z	fBodyAccJerk_std()_X	fBodyAccJerk_std_X
fBodyAccJerk_mean()_X	fBodyAccJerk_mean_X	fBodyAccJerk_std()_Y	fBodyAccJerk_std_Y
fBodyAccJerk_mean()_Y	fBodyAccJerk_mean_Y	fBodyAccJerk_std()_Z	fBodyAccJerk_std_Z
fBodyAccJerk_mean()_Z	fBodyAccJerk_mean_Z	fBodyGyro_std()_X	fBodyGyro_std_X

fBodyGyro_mean_X	fBodyGyro_mean_X	fBodyGyro_std()_Y	fBodyGyro_std_Y
fBodyGyro_mean()_Y	fBodyGyro_mean_Y	fBodyGyro_std()_Z	fBodyGyro_std_Z
fBodyGyro_mean()_Z	fBodyGyro_mean_Z	fBodyAccMag_std()	fBodyAccMag_std
fBodyAccMag_mean()	fBodyAccMag_mean	fBodyBodyAccJerkMag_std()	fBodyAccJerkMag_std
fBodyBodyAccJerkMag_mean()	fBodyAccJerkMag_mean	fBodyBodyGyroMag_std()	fBodyGyroMag_std
fBodyBodyGyroMag_mean()	fBodyGyroMag_mean	fBodyBodyGyroJerkMag_std()	fBodyGyroJerkMag_std
fBodyBodyGyroJerkMag_mean()	fBodyGyroJerkMag_mean	label from the file “activity_label.txt” based on data set in y	activity
tBodyAcc_std_X	tBodyAcc_std_X		

Analysis

This project is splitted in 5 sections:

1. **PREPARE ENVIRONMENT**: These sections create directories where files will be stored and download and install packages used along the project.
2. **GETTING THE DATA**: In this section, all data is downloaded from the internet and stored in folders.
3. **INITIALIZING PATH VARIABLES**: In this section, path variables pointing to data are created. This makes it easier to know where data is stored.
4. **LOADING DATA**: This section loads all data from path variables creating data frame variables in local.
5. **DATA WRANGLING**: The last but no least section covers all questions requested by the course project. Data frames are merged, splitted, and filters in order to get 2 data frames, one data frame for mean a std variable and another for averaging the previous data frame based on activities.

Prepare Environment

In this section, the package “dplyer” is installed and one folder called backup is created to stored the zip data download later.

Getting the data

In this section, the data is downloaded and uncompressed. Later, the name folder is changed from “UCI HAR Dataset” to “data”.

Initializing path variables

Here, some path variables are defined in order to know where the data is stored. The variables are used later to load the data from the text files. These variables are pathData, pathActivity, pathFeatures, pathTest, pathInertialTest, and so on.

Loading Data

In this section, all the data is loaded in variables such as df_activity and df_Xtest as the data frame structure. Additionally, some variables are loaded but not used later such as df_testtotacc_x and df_testtotacc_y, just in case anyone wants to use them in a future work. Finally, some characters such as "(" and ")" are deleted in features variables, and the character "-" is replaced by "-".

Data Wrangling

The last section focuses on developing all the requests. For this purpose, data in test and train folders are merged into "df_x" and "df_y". Also, labels in df_x are setted with "df_features", which is da*ta frame with all the variables name from *features.txt*. Additionally, files in Initial Signal are merged, but they are not necessary being merged. After that, in "df_X_mean" and "df_X_std" is stored all data about mean and std measures with the help of "grep" function. In "df_y" is merged with "df_features" to get all names for each activity. Finally, with the help of "dplyer" package, the entire data set "df_total" (which includes *df_X_mean*, *df_X_std*, "df_y") is grouped by activity and later each column is reduced based on activity too.

Some facts in raw data

- Some variables such as fBodyBodyAccJerkMag_mean have a typo. They have to be in the form fBodyAccJerkMag_mean() without double Body.
- Files in the Initial Signal folder are not really necessary for this project.
- Some variables are defined with meanFreq, these variables are not considered in our final data set. Only variables with mean without Freq.