Peer-graded Assignment: Getting and Cleaning Data Course Project

*Nunno Nugroho*

Instructions

Review criteria

* The submitted data set is tidy.
* The Github repo contains the required scripts.
* GitHub contains a code book that modifies and updates the available codebooks with the data to indicate all the variables and summaries calculated, along with units, and any other relevant information.
* The README that explains the analysis files is clear and understandable.
* The work submitted for this project is the work of the student who submitted it.

Getting and Cleaning Data Course Project

The purpose of this project is to demonstrate your ability to collect, work with, and clean a data set. The goal is to prepare tidy data that can be used for later analysis. You will be graded by your peers on a series of yes/no questions related to the project.

You will be required to submit:

1. A tidy data set as described below,
2. A link to a Github repository with your script for performing the analysis,
3. A code book that describes the variables, the data, and any transformations or work that you performed to clean up the data called CodeBook.md. You should also include a README.md in the repo with your scripts. This repo explains how all of the scripts work and how they are connected.

Dataset

One of the most exciting areas in all of data science right now is wearable computing - see for example this article . Companies like Fitbit, Nike, and Jawbone Up are racing to develop the most advanced algorithms to attract new users. The data linked to from the course website represent data collected from the accelerometers from the Samsung Galaxy S smartphone. A full description is available at the site where the data was obtained:

<http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

Here are the data for the project:

<https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip>

You should create one R script called *run\_analysis.R* that does the following.

1. Merges the training and the test sets to create one data set.
2. Extracts only the measurements on the mean and standard deviation for each measurement.
3. Uses descriptive activity names to name the activities in the data set
4. Appropriately labels the data set with descriptive variable names.
5. From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject.

Good luck!

Here we go!

Preparation Stage

Loading required packages

Hide

library(dplyr)

Download the dataset

Hide

filename <- "Coursera\_DS3\_Final.zip"

# Checking if archieve already exists.

if (!file.exists(filename)){

fileURL <- "https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip"

download.file(fileURL, filename, method="curl")

}

# Checking if folder exists

if (!file.exists("UCI HAR Dataset")) {

unzip(filename)

}

Assigning all data frames

Hide

features <- read.table("UCI HAR Dataset/features.txt", col.names = c("n","functions"))

activities <- read.table("UCI HAR Dataset/activity\_labels.txt", col.names = c("code", "activity"))

subject\_test <- read.table("UCI HAR Dataset/test/subject\_test.txt", col.names = "subject")

x\_test <- read.table("UCI HAR Dataset/test/X\_test.txt", col.names = features$functions)

y\_test <- read.table("UCI HAR Dataset/test/y\_test.txt", col.names = "code")

subject\_train <- read.table("UCI HAR Dataset/train/subject\_train.txt", col.names = "subject")

x\_train <- read.table("UCI HAR Dataset/train/X\_train.txt", col.names = features$functions)

y\_train <- read.table("UCI HAR Dataset/train/y\_train.txt", col.names = "code")

The “Go for it!” Stage

“You should create one R script called *run\_analysis.R* that does the following:”

**Step 1: Merges the training and the test sets to create one data set.**

Hide

X <- rbind(x\_train, x\_test)

Y <- rbind(y\_train, y\_test)

Subject <- rbind(subject\_train, subject\_test)

Merged\_Data <- cbind(Subject, Y, X)

**Step 2: Extracts only the measurements on the mean and standard deviation for each measurement.**

Hide

TidyData <- Merged\_Data %>% select(subject, code, contains("mean"), contains("std"))

**Step 3: Uses descriptive activity names to name the activities in the data set.**

Hide

TidyData$code <- activities[TidyData$code, 2]

**Step 4: Appropriately labels the data set with descriptive variable names.**

Hide

names(TidyData)[2] = "activity"

names(TidyData)<-gsub("Acc", "Accelerometer", names(TidyData))

names(TidyData)<-gsub("Gyro", "Gyroscope", names(TidyData))

names(TidyData)<-gsub("BodyBody", "Body", names(TidyData))

names(TidyData)<-gsub("Mag", "Magnitude", names(TidyData))

names(TidyData)<-gsub("^t", "Time", names(TidyData))

names(TidyData)<-gsub("^f", "Frequency", names(TidyData))

names(TidyData)<-gsub("tBody", "TimeBody", names(TidyData))

names(TidyData)<-gsub("-mean()", "Mean", names(TidyData), ignore.case = TRUE)

names(TidyData)<-gsub("-std()", "STD", names(TidyData), ignore.case = TRUE)

names(TidyData)<-gsub("-freq()", "Frequency", names(TidyData), ignore.case = TRUE)

names(TidyData)<-gsub("angle", "Angle", names(TidyData))

names(TidyData)<-gsub("gravity", "Gravity", names(TidyData))

**Step 5: From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject.**

Hide

FinalData <- TidyData %>%

group\_by(subject, activity) %>%

summarise\_all(funs(mean))

write.table(FinalData, "FinalData.txt", row.name=FALSE)

Final Check Stage

Checking variable names

Hide

str(FinalData)

Classes grouped\_df, tbl\_df, tbl and 'data.frame': 180 obs. of 88 variables:

$ subject : int 1 1 1 1 1 1 2 2 2 2 ...

$ activity : Factor w/ 6 levels "LAYING","SITTING",..: 1 2 3 4 5 6 1 2 3 4 ...

$ TimeBodyAccelerometer.mean...X : num 0.222 0.261 0.279 0.277 0.289 ...

$ TimeBodyAccelerometer.mean...Y : num -0.04051 -0.00131 -0.01614 -0.01738 -0.00992 ...

$ TimeBodyAccelerometer.mean...Z : num -0.113 -0.105 -0.111 -0.111 -0.108 ...

$ TimeGravityAccelerometer.mean...X : num -0.249 0.832 0.943 0.935 0.932 ...

$ TimeGravityAccelerometer.mean...Y : num 0.706 0.204 -0.273 -0.282 -0.267 ...

$ TimeGravityAccelerometer.mean...Z : num 0.4458 0.332 0.0135 -0.0681 -0.0621 ...

$ TimeBodyAccelerometerJerk.mean...X : num 0.0811 0.0775 0.0754 0.074 0.0542 ...

$ TimeBodyAccelerometerJerk.mean...Y : num 0.003838 -0.000619 0.007976 0.028272 0.02965 ...

$ TimeBodyAccelerometerJerk.mean...Z : num 0.01083 -0.00337 -0.00369 -0.00417 -0.01097 ...

$ TimeBodyGyroscope.mean...X : num -0.0166 -0.0454 -0.024 -0.0418 -0.0351 ...

$ TimeBodyGyroscope.mean...Y : num -0.0645 -0.0919 -0.0594 -0.0695 -0.0909 ...

$ TimeBodyGyroscope.mean...Z : num 0.1487 0.0629 0.0748 0.0849 0.0901 ...

$ TimeBodyGyroscopeJerk.mean...X : num -0.1073 -0.0937 -0.0996 -0.09 -0.074 ...

$ TimeBodyGyroscopeJerk.mean...Y : num -0.0415 -0.0402 -0.0441 -0.0398 -0.044 ...

$ TimeBodyGyroscopeJerk.mean...Z : num -0.0741 -0.0467 -0.049 -0.0461 -0.027 ...

$ TimeBodyAccelerometerMagnitude.mean.. : num -0.8419 -0.9485 -0.9843 -0.137 0.0272 ...

$ TimeGravityAccelerometerMagnitude.mean.. : num -0.8419 -0.9485 -0.9843 -0.137 0.0272 ...

$ TimeBodyAccelerometerJerkMagnitude.mean.. : num -0.9544 -0.9874 -0.9924 -0.1414 -0.0894 ...

$ TimeBodyGyroscopeMagnitude.mean.. : num -0.8748 -0.9309 -0.9765 -0.161 -0.0757 ...

$ TimeBodyGyroscopeJerkMagnitude.mean.. : num -0.963 -0.992 -0.995 -0.299 -0.295 ...

$ FrequencyBodyAccelerometer.mean...X : num -0.9391 -0.9796 -0.9952 -0.2028 0.0382 ...

$ FrequencyBodyAccelerometer.mean...Y : num -0.86707 -0.94408 -0.97707 0.08971 0.00155 ...

$ FrequencyBodyAccelerometer.mean...Z : num -0.883 -0.959 -0.985 -0.332 -0.226 ...

$ FrequencyBodyAccelerometer.meanFreq...X : num -0.1588 -0.0495 0.0865 -0.2075 -0.3074 ...

$ FrequencyBodyAccelerometer.meanFreq...Y : num 0.0975 0.0759 0.1175 0.1131 0.0632 ...

$ FrequencyBodyAccelerometer.meanFreq...Z : num 0.0894 0.2388 0.2449 0.0497 0.2943 ...

$ FrequencyBodyAccelerometerJerk.mean...X : num -0.9571 -0.9866 -0.9946 -0.1705 -0.0277 ...

$ FrequencyBodyAccelerometerJerk.mean...Y : num -0.9225 -0.9816 -0.9854 -0.0352 -0.1287 ...

$ FrequencyBodyAccelerometerJerk.mean...Z : num -0.948 -0.986 -0.991 -0.469 -0.288 ...

$ FrequencyBodyAccelerometerJerk.meanFreq...X : num 0.132 0.257 0.314 -0.209 -0.253 ...

$ FrequencyBodyAccelerometerJerk.meanFreq...Y : num 0.0245 0.0475 0.0392 -0.3862 -0.3376 ...

$ FrequencyBodyAccelerometerJerk.meanFreq...Z : num 0.02439 0.09239 0.13858 -0.18553 0.00937 ...

$ FrequencyBodyGyroscope.mean...X : num -0.85 -0.976 -0.986 -0.339 -0.352 ...

$ FrequencyBodyGyroscope.mean...Y : num -0.9522 -0.9758 -0.989 -0.1031 -0.0557 ...

$ FrequencyBodyGyroscope.mean...Z : num -0.9093 -0.9513 -0.9808 -0.2559 -0.0319 ...

$ FrequencyBodyGyroscope.meanFreq...X : num -0.00355 0.18915 -0.12029 0.01478 -0.10045 ...

$ FrequencyBodyGyroscope.meanFreq...Y : num -0.0915 0.0631 -0.0447 -0.0658 0.0826 ...

$ FrequencyBodyGyroscope.meanFreq...Z : num 0.010458 -0.029784 0.100608 0.000773 -0.075676 ...

$ FrequencyBodyAccelerometerMagnitude.mean.. : num -0.8618 -0.9478 -0.9854 -0.1286 0.0966 ...

$ FrequencyBodyAccelerometerMagnitude.meanFreq.. : num 0.0864 0.2367 0.2846 0.1906 0.1192 ...

$ FrequencyBodyAccelerometerJerkMagnitude.mean.. : num -0.9333 -0.9853 -0.9925 -0.0571 0.0262 ...

$ FrequencyBodyAccelerometerJerkMagnitude.meanFreq..: num 0.2664 0.3519 0.4222 0.0938 0.0765 ...

$ FrequencyBodyGyroscopeMagnitude.mean.. : num -0.862 -0.958 -0.985 -0.199 -0.186 ...

$ FrequencyBodyGyroscopeMagnitude.meanFreq.. : num -0.139775 -0.000262 -0.028606 0.268844 0.349614 ...

$ FrequencyBodyGyroscopeJerkMagnitude.mean.. : num -0.942 -0.99 -0.995 -0.319 -0.282 ...

$ FrequencyBodyGyroscopeJerkMagnitude.meanFreq.. : num 0.176 0.185 0.334 0.191 0.19 ...

$ Angle.TimeBodyAccelerometerMean.Gravity. : num 0.021366 0.027442 -0.000222 0.060454 -0.002695 ...

$ Angle.TimeBodyAccelerometerJerkMean..GravityMean. : num 0.00306 0.02971 0.02196 -0.00793 0.08993 ...

$ Angle.TimeBodyGyroscopeMean.GravityMean. : num -0.00167 0.0677 -0.03379 0.01306 0.06334 ...

$ Angle.TimeBodyGyroscopeJerkMean.GravityMean. : num 0.0844 -0.0649 -0.0279 -0.0187 -0.04 ...

$ Angle.X.GravityMean. : num 0.427 -0.591 -0.743 -0.729 -0.744 ...

$ Angle.Y.GravityMean. : num -0.5203 -0.0605 0.2702 0.277 0.2672 ...

$ Angle.Z.GravityMean. : num -0.3524 -0.218 0.0123 0.0689 0.065 ...

$ TimeBodyAccelerometer.std...X : num -0.928 -0.977 -0.996 -0.284 0.03 ...

$ TimeBodyAccelerometer.std...Y : num -0.8368 -0.9226 -0.9732 0.1145 -0.0319 ...

$ TimeBodyAccelerometer.std...Z : num -0.826 -0.94 -0.98 -0.26 -0.23 ...

$ TimeGravityAccelerometer.std...X : num -0.897 -0.968 -0.994 -0.977 -0.951 ...

$ TimeGravityAccelerometer.std...Y : num -0.908 -0.936 -0.981 -0.971 -0.937 ...

$ TimeGravityAccelerometer.std...Z : num -0.852 -0.949 -0.976 -0.948 -0.896 ...

$ TimeBodyAccelerometerJerk.std...X : num -0.9585 -0.9864 -0.9946 -0.1136 -0.0123 ...

$ TimeBodyAccelerometerJerk.std...Y : num -0.924 -0.981 -0.986 0.067 -0.102 ...

$ TimeBodyAccelerometerJerk.std...Z : num -0.955 -0.988 -0.992 -0.503 -0.346 ...

$ TimeBodyGyroscope.std...X : num -0.874 -0.977 -0.987 -0.474 -0.458 ...

$ TimeBodyGyroscope.std...Y : num -0.9511 -0.9665 -0.9877 -0.0546 -0.1263 ...

$ TimeBodyGyroscope.std...Z : num -0.908 -0.941 -0.981 -0.344 -0.125 ...

$ TimeBodyGyroscopeJerk.std...X : num -0.919 -0.992 -0.993 -0.207 -0.487 ...

$ TimeBodyGyroscopeJerk.std...Y : num -0.968 -0.99 -0.995 -0.304 -0.239 ...

$ TimeBodyGyroscopeJerk.std...Z : num -0.958 -0.988 -0.992 -0.404 -0.269 ...

$ TimeBodyAccelerometerMagnitude.std.. : num -0.7951 -0.9271 -0.9819 -0.2197 0.0199 ...

$ TimeGravityAccelerometerMagnitude.std.. : num -0.7951 -0.9271 -0.9819 -0.2197 0.0199 ...

$ TimeBodyAccelerometerJerkMagnitude.std.. : num -0.9282 -0.9841 -0.9931 -0.0745 -0.0258 ...

$ TimeBodyGyroscopeMagnitude.std.. : num -0.819 -0.935 -0.979 -0.187 -0.226 ...

$ TimeBodyGyroscopeJerkMagnitude.std.. : num -0.936 -0.988 -0.995 -0.325 -0.307 ...

$ FrequencyBodyAccelerometer.std...X : num -0.9244 -0.9764 -0.996 -0.3191 0.0243 ...

$ FrequencyBodyAccelerometer.std...Y : num -0.834 -0.917 -0.972 0.056 -0.113 ...

$ FrequencyBodyAccelerometer.std...Z : num -0.813 -0.934 -0.978 -0.28 -0.298 ...

$ FrequencyBodyAccelerometerJerk.std...X : num -0.9642 -0.9875 -0.9951 -0.1336 -0.0863 ...

$ FrequencyBodyAccelerometerJerk.std...Y : num -0.932 -0.983 -0.987 0.107 -0.135 ...

$ FrequencyBodyAccelerometerJerk.std...Z : num -0.961 -0.988 -0.992 -0.535 -0.402 ...

$ FrequencyBodyGyroscope.std...X : num -0.882 -0.978 -0.987 -0.517 -0.495 ...

$ FrequencyBodyGyroscope.std...Y : num -0.9512 -0.9623 -0.9871 -0.0335 -0.1814 ...

$ FrequencyBodyGyroscope.std...Z : num -0.917 -0.944 -0.982 -0.437 -0.238 ...

$ FrequencyBodyAccelerometerMagnitude.std.. : num -0.798 -0.928 -0.982 -0.398 -0.187 ...

$ FrequencyBodyAccelerometerJerkMagnitude.std.. : num -0.922 -0.982 -0.993 -0.103 -0.104 ...

$ FrequencyBodyGyroscopeMagnitude.std.. : num -0.824 -0.932 -0.978 -0.321 -0.398 ...

$ FrequencyBodyGyroscopeJerkMagnitude.std.. : num -0.933 -0.987 -0.995 -0.382 -0.392 ...

- attr(\*, "vars")= chr "subject"

- attr(\*, "drop")= logi TRUE

Take a look at final data

Hide

FinalData

|  |
| --- |
|  |

| **subject**  <int> | **activity**  <fctr> | **TimeBodyAccelerometer.mean...X**  <dbl> | **TimeBodyAccelerometer.mean...Y**  <dbl> |  |
| --- | --- | --- | --- | --- |
| 1 | LAYING | 0.2215982 | -0.040513953 |  |
| 1 | SITTING | 0.2612376 | -0.001308288 |  |
| 1 | STANDING | 0.2789176 | -0.016137590 |  |
| 1 | WALKING | 0.2773308 | -0.017383819 |  |
| 1 | WALKING\_DOWNSTAIRS | 0.2891883 | -0.009918505 |  |
| 1 | WALKING\_UPSTAIRS | 0.2554617 | -0.023953149 |  |
| 2 | LAYING | 0.2813734 | -0.018158740 |  |
| 2 | SITTING | 0.2770874 | -0.015687994 |  |
| 2 | STANDING | 0.2779115 | -0.018420827 |  |
| 2 | WALKING | 0.2764266 | -0.018594920 |  |

Next

123456

...

18

Previous

1-10 of 180 rows | 1-4 of 88 columns

Submission

Code Book

The run\_analysis.R script performs the data preparation and then followed by the 5 steps required as described in the course project’s definition.

1. **Download the dataset**
   * Dataset downloaded and extracted under the folder called UCI HAR Dataset
2. **Assign each data to variables**
   * features <- features.txt : 561 rows, 2 columns  
     *The features selected for this database come from the accelerometer and gyroscope 3-axial raw signals tAcc-XYZ and tGyro-XYZ.*
   * activities <- activity\_labels.txt : 6 rows, 2 columns  
     *List of activities performed when the corresponding measurements were taken and its codes (labels)*
   * subject\_test <- test/subject\_test.txt : 2947 rows, 1 column  
     *contains test data of 9/30 volunteer test subjects being observed*
   * x\_test <- test/X\_test.txt : 2947 rows, 561 columns  
     *contains recorded features test data*
   * y\_test <- test/y\_test.txt : 2947 rows, 1 columns  
     *contains test data of activities’code labels*
   * subject\_train <- test/subject\_train.txt : 7352 rows, 1 column  
     *contains train data of 21/30 volunteer subjects being observed*
   * x\_train <- test/X\_train.txt : 7352 rows, 561 columns  
     *contains recorded features train data*
   * y\_train <- test/y\_train.txt : 7352 rows, 1 columns  
     *contains train data of activities’code labels*
3. **Merges the training and the test sets to create one data set**
   * X (10299 rows, 561 columns) is created by merging x\_train and x\_test using **rbind()** function
   * Y (10299 rows, 1 column) is created by merging y\_train and y\_test using **rbind()** function
   * Subject (10299 rows, 1 column) is created by merging subject\_train and subject\_test using **rbind()** function
   * Merged\_Data (10299 rows, 563 column) is created by merging Subject, Y and X using **cbind()** function
4. **Extracts only the measurements on the mean and standard deviation for each measurement**
   * TidyData (10299 rows, 88 columns) is created by subsetting Merged\_Data, selecting only columns: subject, code and the measurements on the mean and *standard deviation* (std) for each measurement
5. **Uses descriptive activity names to name the activities in the data set**
   * Entire numbers in code column of the TidyData replaced with corresponding activity taken from second column of the activities variable
6. **Appropriately labels the data set with descriptive variable names**
   * code column in TidyData renamed into activities
   * All Acc in column’s name replaced by Accelerometer
   * All Gyro in column’s name replaced by Gyroscope
   * All BodyBody in column’s name replaced by Body
   * All Mag in column’s name replaced by Magnitude
   * All start with character f in column’s name replaced by Frequency
   * All start with character t in column’s name replaced by Time
7. **From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject**
   * FinalData (180 rows, 88 columns) is created by sumarizing TidyData taking the means of each variable for each activity and each subject, after groupped by subject and activity.
   * Export FinalData into FinalData.txt file.

README

Peer-graded Assignment: Getting and Cleaning Data Course Project

This repository is a **Nunno Nugroho** submission for Getting and Cleaning Data course project. It has the instructions on how to run analysis on Human Activity recognition dataset.

Dataset

[Human Activity Recognition Using Smartphones](http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones)

Files

* CodeBook.md a code book that describes the variables, the data, and any transformations or work that I performed to clean up the data
* run\_analysis.R performs the data preparation and then followed by the 5 steps required as described in the course project’s definition:
  + Merges the training and the test sets to create one data set.
  + Extracts only the measurements on the mean and standard deviation for each measurement.
  + Uses descriptive activity names to name the activities in the data set
  + Appropriately labels the data set with descriptive variable names.
  + From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject.
* FinalData.txt is the exported final data after going through all the sequences described above.

Review criteria

* The submitted data set is tidy. 
* The Github repo contains the required scripts. 
* GitHub contains a code book that modifies and updates the available codebooks with the data to indicate all the variables and summaries calculated, along with units, and any other relevant information. 
* The README that explains the analysis files is clear and understandable. 
* The work submitted for this project is the work of the student who submitted it. 

**END**