Code Book

Wednesday June 10 2015

Project Description

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, and 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.

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.

Study design and data processing

Collection of the raw data

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING UPSTAIRS, WALKING DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

Notes on the original (raw) data

The features selected for this database come from the accelerometer and gyroscope 3-axial raw signals tAcc-XYZ and tGyro-XYZ. These time domain signals (prefix 't' to denote time) were captured at a constant rate of 50 Hz. Then they were filtered using a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20 Hz to remove noise. The acceleration signal was then separated into body and gravity acceleration signals (tBodyAcc-XYZ and tGravityAcc-XYZ)

The body linear acceleration and angular velocity were derived in time to obtain Jerk signals (tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ). Also the magnitude of these three-dimensional signals were calculated using the Euclidean norm (tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag).

Finally a Fast Fourier Transform (FFT) was applied to some of these signals producing fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccJerkMag, fBodyGyroMag, fBodyGyroJerkMag.

Creating the tidy datafile

Guide to create the tidy data file

The run_analysis.R script was created with the idea in mind to be self sufficient, to download and extract the necessary files from the zipped archive. However if the archive already exists in the current working directory the script will use the existing file skipping the download step. As a result the zip file does not need to be extracted prior to running the run analysis script. As part of the processing of the main data from the files it will create a file data.txt which is a monolithic file of all the data combined prior to the extraction for the tidy set this is saved simply to save time and re-processing should the script need to be re-run it may be deleted at the users discretion. The final output of the script is a file named tidydataset.txt as described above in the project section.

Cleaning of the data

Files used from the archive:

File Name	File Description
UCI HAR Dataset/activity_labels.txt	Activity description (training & testing)
UCI HAR Dataset/features.txt	Variable Names (training & testing)
UCI HAR Dataset/train/subject_train.txt	Subject Id's (training)
UCI HAR Dataset/train/X_train.txt	Observation estimates (training)
UCI HAR Dataset/train/y_train.txt	Activity Id's (training)
UCI HAR Dataset/test/subject_test.txt	Subject Id's (testing)
UCI HAR Dataset/test/X_test.txt	Observation estimates (testing)
UCI HAR Dataset/test/y_test.txt	Activity Id's (testing)

further information on the processing steps can be found in the README.md

---

Description of the variables in the tidydataset.txt file

```
## 'data.frame':    180 obs. of  68 variables:
##  $ subjectid                              : int  1 1 1 1 1 1 2 2 2 2 ...
```

```
##  $ activity                                  : Factor w/ 6 levels "LAYING","SITTING",..: 1 2 3 4 5 6
##  $ timebodyaccelerometerMEANx                : num  0.222 0.261 0.279 0.277 0.289 ...
##  $ timebodyaccelerometerMEANy                : num  -0.04051 -0.00131 -0.01614 -0.01738 -0.00992 ...
##  $ timebodyaccelerometerMEANz                : num  -0.113 -0.105 -0.111 -0.111 -0.108 ...
##  $ timebodyaccelerometerSDx                  : num  -0.928 -0.977 -0.996 -0.284 0.03 ...
##  $ timebodyaccelerometerSDy                  : num  -0.8368 -0.9226 -0.9732 0.1145 -0.0319 ...
##  $ timebodyaccelerometerSDz                  : num  -0.826 -0.94 -0.98 -0.26 -0.23 ...
##  $ timegravityaccelerometerMEANx             : num  -0.249 0.832 0.943 0.935 0.932 ...
##  $ timegravityaccelerometerMEANy             : num  0.706 0.204 -0.273 -0.282 -0.267 ...
##  $ timegravityaccelerometerMEANz             : num  0.4458 0.332 0.0135 -0.0681 -0.0621 ...
##  $ timegravityaccelerometerSDx               : num  -0.897 -0.968 -0.994 -0.977 -0.951 ...
##  $ timegravityaccelerometerSDy               : num  -0.908 -0.936 -0.981 -0.971 -0.937 ...
##  $ timegravityaccelerometerSDz               : num  -0.852 -0.949 -0.976 -0.948 -0.896 ...
##  $ timebodyaccelerometerjerkMEANx            : num  0.0811 0.0775 0.0754 0.074 0.0542 ...
##  $ timebodyaccelerometerjerkMEANy            : num  0.003838 -0.000619 0.007976 0.028272 0.02965 ...
##  $ timebodyaccelerometerjerkMEANz            : num  0.01083 -0.00337 -0.00369 -0.00417 -0.01097 ...
##  $ timebodyaccelerometerjerkSDx              : num  -0.9585 -0.9864 -0.9946 -0.1136 -0.0123 ...
##  $ timebodyaccelerometerjerkSDy              : num  -0.924 -0.981 -0.986 0.067 -0.102 ...
##  $ timebodyaccelerometerjerkSDz              : num  -0.955 -0.988 -0.992 -0.503 -0.346 ...
##  $ timebodygyroscopeMEANx                    : num  -0.0166 -0.0454 -0.024 -0.0418 -0.0351 ...
##  $ timebodygyroscopeMEANy                    : num  -0.0645 -0.0919 -0.0594 -0.0695 -0.0909 ...
##  $ timebodygyroscopeMEANz                    : num  0.1487 0.0629 0.0748 0.0849 0.0901 ...
##  $ timebodygyroscopeSDx                      : num  -0.874 -0.977 -0.987 -0.474 -0.458 ...
##  $ timebodygyroscopeSDy                      : num  -0.9511 -0.9665 -0.9877 -0.0546 -0.1263 ...
##  $ timebodygyroscopeSDz                      : num  -0.908 -0.941 -0.981 -0.344 -0.125 ...
##  $ timebodygyroscopejerkMEANx                : num  -0.1073 -0.0937 -0.0996 -0.09 -0.074 ...
##  $ timebodygyroscopejerkMEANy                : num  -0.0415 -0.0402 -0.0441 -0.0398 -0.044 ...
##  $ timebodygyroscopejerkMEANz                : num  -0.0741 -0.0467 -0.049 -0.0461 -0.027 ...
##  $ timebodygyroscopejerkSDx                  : num  -0.919 -0.992 -0.993 -0.207 -0.487 ...
##  $ timebodygyroscopejerkSDy                  : num  -0.968 -0.99 -0.995 -0.304 -0.239 ...
##  $ timebodygyroscopejerkSDz                  : num  -0.958 -0.988 -0.992 -0.404 -0.269 ...
##  $ timebodyaccelerometermagnitudeMEAN        : num  -0.8419 -0.9485 -0.9843 -0.137 0.0272 ...
##  $ timebodyaccelerometermagnitudeSD          : num  -0.7951 -0.9271 -0.9819 -0.2197 0.0199 ...
##  $ timegravityaccelerometermagnitudeMEAN     : num  -0.8419 -0.9485 -0.9843 -0.137 0.0272 ...
##  $ timegravityaccelerometermagnitudeSD       : num  -0.7951 -0.9271 -0.9819 -0.2197 0.0199 ...
##  $ timebodyaccelerometerjerkmagnitudeMEAN    : num  -0.9544 -0.9874 -0.9924 -0.1414 -0.0894 ...
##  $ timebodyaccelerometerjerkmagnitudeSD      : num  -0.9282 -0.9841 -0.9931 -0.0745 -0.0258 ...
##  $ timebodygyroscopemagnitudeMEAN            : num  -0.8748 -0.9309 -0.9765 -0.161 -0.0757 ...
##  $ timebodygyroscopemagnitudeSD              : num  -0.819 -0.935 -0.979 -0.187 -0.226 ...
##  $ timebodygyroscopejerkmagnitudeMEAN        : num  -0.963 -0.992 -0.995 -0.299 -0.295 ...
##  $ timebodygyroscopejerkmagnitudeSD          : num  -0.936 -0.988 -0.995 -0.325 -0.307 ...
##  $ frequencybodyaccelerometerMEANx           : num  -0.9391 -0.9796 -0.9952 -0.2028 0.0382 ...
##  $ frequencybodyaccelerometerMEANy           : num  -0.86707 -0.94408 -0.97707 0.08971 0.00155 ...
##  $ frequencybodyaccelerometerMEANz           : num  -0.883 -0.959 -0.985 -0.332 -0.226 ...
##  $ frequencybodyaccelerometerSDx             : num  -0.9244 -0.9764 -0.996 -0.3191 0.0243 ...
##  $ frequencybodyaccelerometerSDy             : num  -0.834 -0.917 -0.972 0.056 -0.113 ...
##  $ frequencybodyaccelerometerSDz             : num  -0.813 -0.934 -0.978 -0.28 -0.298 ...
##  $ frequencybodyaccelerometerjerkMEANx       : num  -0.9571 -0.9866 -0.9946 -0.1705 -0.0277 ...
##  $ frequencybodyaccelerometerjerkMEANy       : num  -0.9225 -0.9816 -0.9854 -0.0352 -0.1287 ...
##  $ frequencybodyaccelerometerjerkMEANz       : num  -0.948 -0.986 -0.991 -0.469 -0.288 ...
##  $ frequencybodyaccelerometerjerkSDx         : num  -0.9642 -0.9875 -0.9951 -0.1336 -0.0863 ...
##  $ frequencybodyaccelerometerjerkSDy         : num  -0.932 -0.983 -0.987 0.107 -0.135 ...
##  $ frequencybodyaccelerometerjerkSDz         : num  -0.961 -0.988 -0.992 -0.535 -0.402 ...
##  $ frequencybodygyroscopeMEANx               : num  -0.85 -0.976 -0.986 -0.339 -0.352 ...
```

```
##  $ frequencybodygyroscopeMEANy                : num  -0.9522 -0.9758 -0.989 -0.1031 -0.0557 ...
##  $ frequencybodygyroscopeMEANz                : num  -0.9093 -0.9513 -0.9808 -0.2559 -0.0319 ...
##  $ frequencybodygyroscopeSDx                  : num  -0.882 -0.978 -0.987 -0.517 -0.495 ...
##  $ frequencybodygyroscopeSDy                  : num  -0.9512 -0.9623 -0.9871 -0.0335 -0.1814 ...
##  $ frequencybodygyroscopeSDz                  : num  -0.917 -0.944 -0.982 -0.437 -0.238 ...
##  $ frequencybodyaccelerometermagnitudeMEAN    : num  -0.8618 -0.9478 -0.9854 -0.1286 0.0966 ...
##  $ frequencybodyaccelerometermagnitudeSD      : num  -0.798 -0.928 -0.982 -0.398 -0.187 ...
##  $ frequencybodyaccelerometerjerkmagnitudeMEAN: num  -0.9333 -0.9853 -0.9925 -0.0571 0.0262 ...
##  $ frequencybodyaccelerometerjerkmagnitudeSD  : num  -0.922 -0.982 -0.993 -0.103 -0.104 ...
##  $ frequencybodygyroscopemagnitudeMEAN        : num  -0.862 -0.958 -0.985 -0.199 -0.186 ...
##  $ frequencybodygyroscopemagnitudeSD          : num  -0.824 -0.932 -0.978 -0.321 -0.398 ...
##  $ frequencybodygyroscopejerkmagnitudeMEAN    : num  -0.942 -0.99 -0.995 -0.319 -0.282 ...
##  $ frequencybodygyroscopejerkmagnitudeSD      : num  -0.933 -0.987 -0.995 -0.382 -0.392 ...
```

```
## NULL
```

Abbreviations and units of measurement for the above variables

- Leading "time" or "frequency" denotes a time or frequency domain measurement.
- body = Movement of the body.
- gravity = Acceleration of gravity.
- accelerometer = Accelerometer measurement in g's.
- gyroscope = Gryroscope measurement in g/sec.
- jerk = Sudden movement measured in rad/sec/sec.
- magnitude = Magnitude of movement.
- MEAN[X,Y,Z] = Mean calculated and axis.
- SD[X,Y,Z] = Standard deviation calculated and axis.

This tidy data set is a set of variables for each activity and subject. The original data (10299 instances) was grouped by 30 subjects and 6 activities then for each of the 66 variables of mean and standard deviation an average was calculated. The dataset (see above) contains 33 columns of mean variables and 33 columns of standard deviation variables, the first row is the header information.