# Codebook for Getting and Cleaning Data project

Greg Main  
17 August 2015

# Project Description

Demonstrate the ability to collect, work with, and clean a data set. The goal is to prepare tidy data that can be used for later analysis.

# Raw data

### Collection of the raw data

The data set used was Human Activity Recognition Using Smartphones Data set (Version 1.0) obtained from the UCI Machine Learning Repository

### Raw database Abstract

Human Activity Recognition database built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors.

### Raw database collection

The data was collected from experiments 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 data set 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.

### Raw data - signals recorded

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. Similarly, the acceleration signal was then separated into body and gravity acceleration signals (tBodyAcc-XYZ and tGravityAcc-XYZ) using another low pass Butterworth filter with a corner frequency of 0.3 Hz.

Subsequently, 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. (Note the 'f' to indicate frequency domain signals).

These signals were used to estimate variables of the feature vector for each pattern (-XYZ' is used to denote 3-axial signals in the X, Y and Z directions:

* tBodyAcc-XYZ
* tGravityAcc-XYZ
* tBodyAccJerk-XYZ
* tBodyGyro-XYZ
* tBodyGyroJerk-XYZ
* tBodyAccMag
* tGravityAccMag
* tBodyAccJerkMag
* tBodyGyroMag
* tBodyGyroJerkMag
* fBodyAcc-XYZ
* fBodyAccJerk-XYZ
* fBodyGyro-XYZ
* fBodyAccMag
* fBodyAccJerkMag
* fBodyGyroMag
* fBodyGyroJerkMag

### Variables estimated from the signals

The set of variables that were estimated from these signals are:

* mean(): Mean value
* std(): Standard deviation
* mad(): Median absolute deviation
* max(): Largest value in array
* min(): Smallest value in array
* sma(): Signal magnitude area
* energy(): Energy measure. Sum of the squares divided by the number of values.
* iqr(): Interquartile range
* entropy(): Signal entropy
* arCoeff(): Autorregresion coefficients with Burg order equal to 4
* correlation(): correlation coefficient between two signals
* maxInds(): index of the frequency component with largest magnitude
* meanFreq(): Weighted average of the frequency components to obtain a mean frequency
* skewness(): skewness of the frequency domain signal
* kurtosis(): kurtosis of the frequency domain signal
* bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window.
* angle(): Angle between to vectors.

Additional vectors obtained by averaging the signals in a signal window sample. These are used on the angle() variable:

* gravityMean
* tBodyAccMean
* tBodyAccJerkMean
* tBodyGyroMean
* tBodyGyroJerkMean

The complete list of variables of each feature vector is available in 'features.txt'

No unit of measures is reported as all features were normalized and bounded within [-1,1].

### Raw dataset files

The raw data set included the following files

* 'README.txt' - overview of the files and collection process
* 'features\_info.txt': Shows information about the variables used on the feature vector.
* 'features.txt': List of all features (or variables) collected.
* 'activity\_labels.txt': Description of the activity that can be linked to a numeric reference in the class labels (dataset x\_test and y\_test).
* 'train/X\_train.txt': Training data set.
* 'train/y\_train.txt': Training labels.
* 'test/X\_test.txt': Test data set.
* 'test/y\_test.txt': Test labels.
* 'train/subject\_train.txt': Each row identifies the subject who performed the activity for each window sample. Its range is from 1 to 30.
* 'test/subject\_test.txt': Each row identifies the subject who performed the activity for each window sample. Its range is from 1 to 30.

# Creating the tidy datafile

The raw data is downloaded and transformed into a tidy data set by running the run\_analysis.R script

### Downloading the raw datasets

1. The data set is downloaded from the following link <https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Data> set.zip
2. The zip archive is unzipped to a folder ./data/UCI HAR Data set
3. The script then sets the working directory to ./data

### Reading in the datasets

The script reads in the following datasets from the zip archive (see above file list for the descriptions of each file)

1. Features and activity column label files

* features.txt (read in as features)
* activity\_labels.txt (read in as activity\_labels)

1. Data, activity labels and subject ID data for Test data set

* X\_test.txt (read in as x\_test)
* y\_test.txt (read in as y\_test)
* subject\_test. txt (read in as subject\_test)

1. Data, activity labels and subject ID data for Train data set

* X\_train.txt (read in as x\_train)
* y\_train.txt (read in as y\_train)
* subject\_train.txt (read in as subject\_train)

### Adding column names

1. Manually add identifying names to columns in the features dataset (features\_id and features\_name)
2. Manually add identifying names to columns in the activity\_labels dataset (activity\_id, activity\_name)
3. Manually add identifying names to columns in the activity and subject ID datasets for test datasets (number 2 above)

* for y\_test add a column name activity\_id
* for subject\_test add a column name subject\_id

1. Manually add identifying names to columns in the activity and subject ID datasets for train datasets (number 3 above)

* for y\_train add a column name activity\_id
* for subject\_train add a column name subject\_id

1. Using the list of names included in the features dataset (containing 561 entries) and use this to label the 561 columns of data in the x\_test and x\_train datasets

### Merging the training and test datasets

Combine the 6 datasets as follows:

1. Combine horizontally (using cbind) the three data sets for test (x\_test + y\_test + subject\_test)
2. Combine horizontally (using cbind) the three data sets for train (x\_train + y\_train + subject\_train)

The result from above is 2 datasets of 563 columns and 2947 and 7352 respectively.

1. Combine vertically these two subsequent test and train datasets created above (using rbind) to form a merged data set of 10299 rows and 563 columns

### Extract only the mean and standard deviation variables

From the merged data set extract a trimmed data set with only the mean (variables with labels that contain "mean") and standard deviation (variables with labels that contain "std").

### Cleaning the data

1. Merge the trimmed data set with the activity\_name file to bring into the trimmed dataset the activity names (instead of the existing numerical references)
2. Remove the activity\_id column (as this is now redundant given it is named in the above step)
3. Re-label the column headings by replacing abbreviations as follows:

* re-label references to t and f with the words time and frequency
* re-label the columns to make the titles more readable in plain language (including Acc to Accelerometer, Gyro to Gyroscope, Mag to Magnitude and BodyBody to Body)
* Remove the () marks after the mean and std indicators and change mean and std to Mean and StdDev respectively

### Description of the variables in the tidy file (sensor\_data\_subset.txt)

1. The initial tidy data set prior to the creation of the final submitted file (finaldata) contained 10299 rows and 68 columns of data, and included the folowing:

* an activity column (activity\_name): WALKING, WALKING\_UPSTAIRS, WALKING\_DOWNSTAIRS, SITTING, STANDING, LAYING
* an identifier of the subject who carried out the experiment (subject\_id): numbered 1 to 30
* 66 columns (feature vector) with time and frequency domain signal variables (all numeric)

1. The final tidy data set (alldatamean) comprises the average for each activity and each subject and has been outputted to sensor\_data\_subset.txt. This contains 180 rows and 68 columns of data, comprising:

* an activity column (activity\_name): WALKING, WALKING\_UPSTAIRS, WALKING\_DOWNSTAIRS, SITTING, STANDING, LAYING
* an identifier of the subject who carried out the experiment (subject\_id): numbered 1 to 30
* 66 columns (feature vector) with time and frequency domain signal variables (numeric)