Code Book

Rohit Padebettu

September 5, 2016

Steps in Analysis

1. Load the `dplyr`,`readr` and `tidyr` libraries
2. Read the X.txt, y.txt and subject.txt files in the "test" and "train" directories into respective tibbles
3. Read the features.txt file and activity_labels.txt file in the downloaded file directory into tibbles
4. Merge the train and test datasets for each of X,y and subject files using `bind_rows()` and `bind_cols()` from readr package
5. Grep the features.txt file to get an index for columns with `mean()` and `std()` occurring in them
6. Use the index to subset the columns of the tibble `X` and tibble `feature`
7. Append a column to the tibble `y` by using `inner_join` with the `activity_lables` tibble. This associates descriptive names with each Activity in `y`
8. Give appropriate descriptive names to the columns in `y`,`subject` and use the subsetted `features_filtered` tibble to name the columns in subsetted `X_filt`
9. Combine the `subject`, `y`, `X_filt` tibbles to create a new dataset `DataSetMergedNamed` which now has descriptive variable names and only mean and standard deviation attributes
10. Tidy the dataset by
    - Removing the `ActivityId` attribute
    - melting the data using `gather()` from `dplyr` package to create a dataset with only 4 columns `Subject`,`Activity`,`Feature`,`Value`
    - Group the data set into sub-groups based on Subject, Activity and Feature using `group_by` function
    - Calculate the `mean()` of each such sub_group using the `summarize` function in `dplyr` package
    - Name the column `Average`
11. The resulting dataset `DataTidy` is tidy as each column is a variable, each row is an observation and each observational unit is a value in the table.
12. A summary of `DataTidy` is displayed and result is saved as `TidyData.txt` and `TidyData.csv`

Brief Description of TidyData.txt

The TidyData set has 4 columns given below

```
## [1] "Subject"  "Activity" "Feature"  "Average"
```

```
##   Subject Activity           Feature    Average
## 1       1   LAYING fBodyAcc-mean()-X -0.9390991
## 2       1   LAYING fBodyAcc-mean()-Y -0.8670652
## 3       1   LAYING fBodyAcc-mean()-Z -0.8826669
## 4       1   LAYING  fBodyAcc-std()-X -0.9244374
## 5       1   LAYING  fBodyAcc-std()-Y -0.8336256
```

```
##     Subject                  Activity                 Feature
##  Min.   : 1.0   LAYING            :1980   fBodyAcc-mean()-X:  180
##  1st Qu.: 8.0   SITTING           :1980   fBodyAcc-mean()-Y:  180
##  Median :15.5   STANDING          :1980   fBodyAcc-mean()-Z:  180
```

```
##  Mean   :15.5   WALKING           :1980   fBodyAcc-std()-X :  180
##  3rd Qu.:23.0   WALKING_DOWNSTAIRS:1980   fBodyAcc-std()-Y :  180
##  Max.   :30.0   WALKING_UPSTAIRS  :1980   fBodyAcc-std()-Z :  180
##                                           (Other)          :10800
##     Average
##  Min.   :-0.99767
##  1st Qu.:-0.96205
##  Median :-0.46989
##  Mean   :-0.48436
##  3rd Qu.:-0.07836
##  Max.   : 0.97451
##
```

Subject : Is the ID of the person subject to the test. There were 30 volunteers in the original test

Activity: Is the Activity the subject performed while measured. It is encoded as a factor of 6 levels, listed above

Feature : Is the feature extracted from the original data set. There are 66 such features selected from the original dataset of 561 features.

Average : Is the average value of a particular feature for the subject when performing a specific activity

Feature Selection

(reproduced from original feature_info.txt document)

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:

```
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
```

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

```
mean(): Mean value
std(): Standard deviation
```

Feature List

The entire list of 66 features contained in this data set are as follows

```
##       [,1]
##  [1,] "tBodyAcc-mean()-X"
##  [2,] "tBodyAcc-mean()-Y"
##  [3,] "tBodyAcc-mean()-Z"
##  [4,] "tBodyAcc-std()-X"
##  [5,] "tBodyAcc-std()-Y"
##  [6,] "tBodyAcc-std()-Z"
##  [7,] "tGravityAcc-mean()-X"
##  [8,] "tGravityAcc-mean()-Y"
##  [9,] "tGravityAcc-mean()-Z"
## [10,] "tGravityAcc-std()-X"
## [11,] "tGravityAcc-std()-Y"
## [12,] "tGravityAcc-std()-Z"
## [13,] "tBodyAccJerk-mean()-X"
## [14,] "tBodyAccJerk-mean()-Y"
## [15,] "tBodyAccJerk-mean()-Z"
## [16,] "tBodyAccJerk-std()-X"
## [17,] "tBodyAccJerk-std()-Y"
## [18,] "tBodyAccJerk-std()-Z"
## [19,] "tBodyGyro-mean()-X"
## [20,] "tBodyGyro-mean()-Y"
## [21,] "tBodyGyro-mean()-Z"
## [22,] "tBodyGyro-std()-X"
## [23,] "tBodyGyro-std()-Y"
## [24,] "tBodyGyro-std()-Z"
## [25,] "tBodyGyroJerk-mean()-X"
## [26,] "tBodyGyroJerk-mean()-Y"
## [27,] "tBodyGyroJerk-mean()-Z"
## [28,] "tBodyGyroJerk-std()-X"
## [29,] "tBodyGyroJerk-std()-Y"
## [30,] "tBodyGyroJerk-std()-Z"
## [31,] "tBodyAccMag-mean()"
## [32,] "tBodyAccMag-std()"
## [33,] "tGravityAccMag-mean()"
## [34,] "tGravityAccMag-std()"
## [35,] "tBodyAccJerkMag-mean()"
```

```
## [36,] "tBodyAccJerkMag-std()"
## [37,] "tBodyGyroMag-mean()"
## [38,] "tBodyGyroMag-std()"
## [39,] "tBodyGyroJerkMag-mean()"
## [40,] "tBodyGyroJerkMag-std()"
## [41,] "fBodyAcc-mean()-X"
## [42,] "fBodyAcc-mean()-Y"
## [43,] "fBodyAcc-mean()-Z"
## [44,] "fBodyAcc-std()-X"
## [45,] "fBodyAcc-std()-Y"
## [46,] "fBodyAcc-std()-Z"
## [47,] "fBodyAccJerk-mean()-X"
## [48,] "fBodyAccJerk-mean()-Y"
## [49,] "fBodyAccJerk-mean()-Z"
## [50,] "fBodyAccJerk-std()-X"
## [51,] "fBodyAccJerk-std()-Y"
## [52,] "fBodyAccJerk-std()-Z"
## [53,] "fBodyGyro-mean()-X"
## [54,] "fBodyGyro-mean()-Y"
## [55,] "fBodyGyro-mean()-Z"
## [56,] "fBodyGyro-std()-X"
## [57,] "fBodyGyro-std()-Y"
## [58,] "fBodyGyro-std()-Z"
## [59,] "fBodyAccMag-mean()"
## [60,] "fBodyAccMag-std()"
## [61,] "fBodyBodyAccJerkMag-mean()"
## [62,] "fBodyBodyAccJerkMag-std()"
## [63,] "fBodyBodyGyroMag-mean()"
## [64,] "fBodyBodyGyroMag-std()"
## [65,] "fBodyBodyGyroJerkMag-mean()"
## [66,] "fBodyBodyGyroJerkMag-std()"
```