# Code Book

*Derrick Williams*

*11/17/2019*

## CodeBook

### Coursera Final Project guidelines

1. The submitted data set is tidy.
2. The Github repo contains the required scripts.
3. 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.
4. The README that explains the analysis files is clear and understandable.
5. The work submitted for this project is the work of the student who submitted it.

### Downloading appropriate packages

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

`dplyr` will be used later on in the cleaning process.

### Designating working directory

Determining if the folder desitnation exists and if not to create one.

### Data download and unzip

```
# String variables for file download
url <-
  "http://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip"
download.file(url, destfile = "./data/UIC.zip", method = "curl")
UICfolder <- unzip("./data/UIC.zip")
```

`UICfolder <- unzip("./data/UIC.zip")` had 1 column and 28 rows of .txt file, i.e. the `UICfoler` contained 28 different flat text files.

```
#figuring out what files to read using regular expression
trainingRows <- grep("(.)*train(.)*", UICfolder, ignore.case = TRUE)
testRows <- grep("(.)*test(.)*", UICfolder, ignore.case = TRUE)
```

Both the `trainingRows` and `testingRows` search with regular expression resulted in 12 hits. Whereas, output from the above mentioned datasets lead to the understanding that only the last 3 (10-12) were important for this analysis.

## Saving values for to plug into the `read.table` function

```
# Update test and training data based on text file description
subjectTrain <- UICfolder[trainingRows[10]]
Xtrain <- UICfolder[trainingRows[11]]
Ytrain <- UICfolder[trainingRows[12]]
subjecttest <- UICfolder[testRows[10]]
Xtest <- UICfolder[testRows[11]]
Ytest <- UICfolder[testRows[12]]
```

Using the finding from the regular expression to prepare for reading the text file while minimizing the chances for typos.

## Reading text files of designated dataset and renaming column names

```
activity_labels <- read.table("UCI HAR Dataset/activity_labels.txt",
                              col.names = c("activityId","activityType"))
features <- read.table("UCI HAR Dataset/features.txt", col.names = c("n","functions"))
subject_test <- read.table(subjecttest, col.names = "subId")
subject_train <- read.table(subjectTrain, col.names = "subId")
x_test <- read.table(Xtest, col.names = features$functions)
x_train <- read.table(Xtrain, col.names = features$functions)
y_test <- read.table(Ytest, col.names = "activityId")
y_train <- read.table(Ytrain, col.names = "activityId")
```

Table 1: size of tables from cooresponding text files

| File Name | # of obs. | # of variables |
|---|---|---|
| activity_labels | 6 | 2 |
| features | 561 | 2 |
| subject_test | 2947 | 1 |
| subject_train | 7352 | 1 |
| x_test | 2947 | 561 |
| x_train | 7352 | 561 |
| y_test | 2947 | 1 |
| y_train | 7352 | 1 |

## Merging Data

```
subjectdata <- rbind(subject_train,subject_test)
x_data <- rbind(x_train,x_test)
y_data <- rbind(y_test,y_train)
finaldata <- cbind(subjectdata,x_data,y_data)
```

`rbind` was used to combine `subject_train` with `subject_test`, `x_train` with `x_test`, and `y_test` with `y_train` while `cbind` was used to combine the priviously merged datasets `subjectdata` , `x_data`, and `y_data`.

Table 2: size of tables after merging the files

| File Name | # of obs. | # of variables |
|---|---|---|
| `subjectdata` | 10299 | 1 |
| `x_data` | 10299 | 561 |
| `y_data` | 10299 | 1 |
| `finaldata` | 10299 | 563 |

## Selecting appropriate columns with `dplyr`

```
preTidyData <- finaldata %>% select(subId, activityId, contains("mean"), contains("std"))
```

`preTidyData` was derived from selecting the column names that contained "mean" and "std", which resulted in a dataset with 10299 obs. and 88 variables

## Cleaning up the column names

```
names(preTidyData) <- gsub("\\(|\\)", "", names(preTidyData), perl  = TRUE)
```

## Changing column names to more descriptive titles

```
names(preTidyData) <- gsub("Acc", "acceleration", names(preTidyData))
names(preTidyData) <- gsub("^t", "time", names(preTidyData))
names(preTidyData) <- gsub("^f", "frequency", names(preTidyData))
names(preTidyData) <- gsub("BodyBody", "body", names(preTidyData))
names(preTidyData) <- gsub("mean", "mean", names(preTidyData))
names(preTidyData) <- gsub("Freq", "frequency", names(preTidyData))
names(preTidyData) <- gsub("Mag", "magnitude", names(preTidyData))
```

## Grouping and sumarizing the data

```
TidyData <- preTidyData %>%
  group_by(subId, activityId) %>%
  summarise_all(funs(mean))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

This two step procedure lead to `Tidydata` having 180 obs. and 88 variables

```
names(TidyData)
```

```
##  [1] "subId"
##  [2] "activityId"
##  [3] "timeBodyacceleration.mean...X"
##  [4] "timeBodyacceleration.mean...Y"
##  [5] "timeBodyacceleration.mean...Z"
##  [6] "timeGravityacceleration.mean...X"
##  [7] "timeGravityacceleration.mean...Y"
##  [8] "timeGravityacceleration.mean...Z"
##  [9] "timeBodyaccelerationJerk.mean...X"
## [10] "timeBodyaccelerationJerk.mean...Y"
## [11] "timeBodyaccelerationJerk.mean...Z"
## [12] "timeBodyGyro.mean...X"
## [13] "timeBodyGyro.mean...Y"
## [14] "timeBodyGyro.mean...Z"
## [15] "timeBodyGyroJerk.mean...X"
## [16] "timeBodyGyroJerk.mean...Y"
## [17] "timeBodyGyroJerk.mean...Z"
## [18] "timeBodyaccelerationmagnitude.mean.."
## [19] "timeGravityaccelerationmagnitude.mean.."
## [20] "timeBodyaccelerationJerkmagnitude.mean.."
## [21] "timeBodyGyromagnitude.mean.."
## [22] "timeBodyGyroJerkmagnitude.mean.."
## [23] "frequencyBodyacceleration.mean...X"
## [24] "frequencyBodyacceleration.mean...Y"
## [25] "frequencyBodyacceleration.mean...Z"
## [26] "frequencyBodyacceleration.meanfrequency...X"
## [27] "frequencyBodyacceleration.meanfrequency...Y"
## [28] "frequencyBodyacceleration.meanfrequency...Z"
## [29] "frequencyBodyaccelerationJerk.mean...X"
## [30] "frequencyBodyaccelerationJerk.mean...Y"
## [31] "frequencyBodyaccelerationJerk.mean...Z"
```

```
## [32] "frequencyBodyaccelerationJerk.meanfrequency...X"
## [33] "frequencyBodyaccelerationJerk.meanfrequency...Y"
## [34] "frequencyBodyaccelerationJerk.meanfrequency...Z"
## [35] "frequencyBodyGyro.mean...X"
## [36] "frequencyBodyGyro.mean...Y"
## [37] "frequencyBodyGyro.mean...Z"
## [38] "frequencyBodyGyro.meanfrequency...X"
## [39] "frequencyBodyGyro.meanfrequency...Y"
## [40] "frequencyBodyGyro.meanfrequency...Z"
## [41] "frequencyBodyaccelerationmagnitude.mean.."
## [42] "frequencyBodyaccelerationmagnitude.meanfrequency.."
## [43] "frequencybodyaccelerationJerkmagnitude.mean.."
## [44] "frequencybodyaccelerationJerkmagnitude.meanfrequency.."
## [45] "frequencybodyGyromagnitude.mean.."
## [46] "frequencybodyGyromagnitude.meanfrequency.."
## [47] "frequencybodyGyroJerkmagnitude.mean.."
## [48] "frequencybodyGyroJerkmagnitude.meanfrequency.."
## [49] "angle.tBodyaccelerationMean.gravity."
## [50] "angle.tBodyaccelerationJerkMean..gravityMean."
## [51] "angle.tBodyGyroMean.gravityMean."
## [52] "angle.tBodyGyroJerkMean.gravityMean."
## [53] "angle.X.gravityMean."
## [54] "angle.Y.gravityMean."
## [55] "angle.Z.gravityMean."
## [56] "timeBodyacceleration.std...X"
## [57] "timeBodyacceleration.std...Y"
## [58] "timeBodyacceleration.std...Z"
## [59] "timeGravityacceleration.std...X"
## [60] "timeGravityacceleration.std...Y"
## [61] "timeGravityacceleration.std...Z"
## [62] "timeBodyaccelerationJerk.std...X"
## [63] "timeBodyaccelerationJerk.std...Y"
## [64] "timeBodyaccelerationJerk.std...Z"
## [65] "timeBodyGyro.std...X"
## [66] "timeBodyGyro.std...Y"
## [67] "timeBodyGyro.std...Z"
## [68] "timeBodyGyroJerk.std...X"
## [69] "timeBodyGyroJerk.std...Y"
## [70] "timeBodyGyroJerk.std...Z"
## [71] "timeBodyaccelerationmagnitude.std.."
## [72] "timeGravityaccelerationmagnitude.std.."
## [73] "timeBodyaccelerationJerkmagnitude.std.."
## [74] "timeBodyGyromagnitude.std.."
## [75] "timeBodyGyroJerkmagnitude.std.."
## [76] "frequencyBodyacceleration.std...X"
## [77] "frequencyBodyacceleration.std...Y"
## [78] "frequencyBodyacceleration.std...Z"
## [79] "frequencyBodyaccelerationJerk.std...X"
## [80] "frequencyBodyaccelerationJerk.std...Y"
## [81] "frequencyBodyaccelerationJerk.std...Z"
## [82] "frequencyBodyGyro.std...X"
## [83] "frequencyBodyGyro.std...Y"
## [84] "frequencyBodyGyro.std...Z"
## [85] "frequencyBodyaccelerationmagnitude.std.."
```

```
## [86] "frequencybodyaccelerationJerkmagnitude.std.."
## [87] "frequencybodyGyromagnitude.std.."
## [88] "frequencybodyGyroJerkmagnitude.std.."
```

### Writing text file

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