This is the READ ME file that accompanies my run.analysis.R files for the Week 4 Assignment from the Getting and Cleaning data course on Coursera. It explains the code in the run.analysis.R script and my rational.

**Getting Started**

Libraries necessary for this code are dplyr and tidyr

I downloaded and unzipped the UCI HAR data prior to script execution.

The script calls the necessary libraries and sets the working directory to the directory containing the UCI HAR Dataset folder.

Within the UCI HAR Dataset folder is the following (the files in bold are used in the run.analysis.R script):

README.txt – a test file that explains the UCL HAR experiments and data files contained within the UCL HAR Dataset folder

features\_info.txt – a text file that explains the features data gathered in the experiment

**features.txt** – a text file that lists the features and their corresponding numbers in the X\_test.txt and X\_train.txt data files

**activity\_labels.txt** – a text file that lists the 6 activities performed by the test subjects and their corresponding numbers in the X\_test.txt and X\_train.txt data files

test folder

Inertial Signals folder – The raw data signals, this data was not used for this analysis

**subject\_test.txt** – a text file that lists the subjects who performed each of the trials (rows) in the X\_test.txt dataset

**X\_test.txt** – a text file that contains the unlabeled test data

**y\_test.txt** – a text file that contains numbers corresponding to the activities the subjects performed in each experiment (row) of the X\_test.txt dataset

train folder

Inertial Signals folder – The raw data signals, this data was not used for this analysis

**subject\_train.txt** – a text file that lists the subjects who performed each of the trials (rows) in the X\_train.txt dataset

**X\_train.txt** – a text file that contains the unlabeled training data

**y\_train.txt** – a text file that contains numbers corresponding to the activities the subjects performed in each experiment (row) of the X\_train.txt dataset

**Input data (test subjects)**

xtest is a table with 2947 rows, 561 columns, and 1,653,267 observations

ytest is a list of 2947 variables that correspond to the activities being performed

subtest is a list of 2947 variables that correspond to the subjects

feat is a list with 561 variables that correspond features described in the features\_info.txt file

Based on the info above, I surmised that the data contained in xtest is the feature data from 2947 measurements. I therefore changed the column names of xtest to correspond to the features (colnames(xtest) <- feat) and used cbind to combine the xtest, ytest, and subtest data to create the test\_data table, which has 2947 rows and 564 columns. With this command, I also created a column at the beginning of the data frame with the test type (“Test Subject”). This will be useful when merging the two data sets.

**Input data (training subjects)**

xtrain is a table with 7352 rows, 561 columns, and 4,124,472 observations

ytrain is a list of 7352 variables that correspond to the activities being performed

subtrain is a list of 7352 variables that correspond to the subjects

I repeated the process detailed above with the training data, creating the train\_data data frame, which has 7352 rows and 564 columns. For this data.frame, the test type column is labeled “Training Subject”.

**Merging files**

Since both test\_data and train\_data have the same columns, the two data sets can be easily merged into one large data frame (merged\_data), with 10299 rows and the same 564 columns.

The data type, subject, position, and all the mean and standard deviation measurements were extracted using the dplyr ‘select’ function, to create the merged\_data\_mnst data frame with 10,299 rows and 69 columns. Since selecting for matches(‘mean|std’) retained the MeanFreq and angle means, I excluded these using -matches(‘Freq|angle’).

**Cleaning Merged Data**

The Position column contains numbers that correspond to the Subject position for the test and trial runs according to the activity\_labels.txt file. In order to change these numbers to the actual positions, I used a series of ifelse statements, for example, if the position is 1, replaced the variable with “Walking”, ect…

**Creating the Second Data Set of the Means**

I used the aggregate function to split the data by Type, Subject, and Position, calculate the mean of each column, and recombine the data. Since the Type, Subject, and Position columns are not numeric, NAs are returned (which produces a bunch of warnings. Aggregate also produces a column for each grouping function (Group.1, Group.2, Group.3), with the corresponding Type, Subject, and Position. The resulting table (mdmst\_means) is 180 rows by 72 columns.

**Cleaning the Second Dataset**

Since the Type, Subject, and Position columns contain only NAs, resulting in a cleaner table (mdmst\_meanscl) of 180 row and 69 columns. I further tidied up the data by arranging the data by Test Type (Group.1) and Subject Number (Group.2), creating mdmst\_meanscl2. In order to do this, I had to change the class of the Group.2 column of data from “factor” to “character” and then “numeric”. Going straight from “factor” to “numeric” caused the subjects to be renumbered, messing up the data. Next, I changed the column names to make them easier to understand. I renamed columns 1-3, “Date Type”, “Subject Number”, and “Position”. From columns 4-69, I removed the numbers are the beginning of the names and the “()” where applicable.

Lastly, I used write.table to save the cleaned second dataset in a text file as tidy\_data\_final.

When tidy\_data\_final is read with read.table (header = TRUE) the spaces and “-“ in the column names are changed to “.”s.

**References**

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

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

https://www.rstudio.com/wp-content/uploads/2016/09/RegExCheatsheet.pdf