Tidy Data Project

The purpose of this project is to take a large messy dataset (http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones) and to put together a tidy dataframe which is read-able and contains some pertinent info.

About the dataset: "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." So the goal of this is simply to merge the two datasets, and report just the mean and std, for all the individual subjects but also an overall mean for the respective activities.

To run this code, it is best to clone the repo and then just run this file.

This file lists all the features

```
#Get the feature list which represents the values in the dataset
features <- read.csv("UCI HAR Dataset/features.txt", header = FALSE, sep = "\n");
head(features)
```

```
##                    V1
## 1 1 tBodyAcc-mean()-X
## 2 2 tBodyAcc-mean()-Y
## 3 3 tBodyAcc-mean()-Z
## 4  4 tBodyAcc-std()-X
## 5  5 tBodyAcc-std()-Y
## 6  6 tBodyAcc-std()-Z
```

```
#Read in the test data and assign the columns to the features
test <- read.csv("UCI HAR Dataset/test/X_test.txt", sep = "", header = FALSE);
head(test[,1:6])
```

```
##          V1          V2          V3         V4         V5         V6
## 1 0.2571778 -0.02328523 -0.01465376 -0.9384040 -0.9200908 -0.6676833
## 2 0.2860267 -0.01316336 -0.11908252 -0.9754147 -0.9674579 -0.9449582
## 3 0.2754848 -0.02605042 -0.11815167 -0.9938190 -0.9699255 -0.9627480
## 4 0.2702982 -0.03261387 -0.11752018 -0.9947428 -0.9732676 -0.9670907
## 5 0.2748330 -0.02784779 -0.12952716 -0.9938525 -0.9674455 -0.9782950
## 6 0.2792199 -0.01862040 -0.11390197 -0.9944552 -0.9704169 -0.9653163
```

```
colnames(test) <- features[,1];
head(test[,1:6])
```

```
##   1 tBodyAcc-mean()-X 2 tBodyAcc-mean()-Y 3 tBodyAcc-mean()-Z
## 1           0.2571778         -0.02328523         -0.01465376
## 2           0.2860267         -0.01316336         -0.11908252
## 3           0.2754848         -0.02605042         -0.11815167
## 4           0.2702982         -0.03261387         -0.11752018
## 5           0.2748330         -0.02784779         -0.12952716
## 6           0.2792199         -0.01862040         -0.11390197
##   4 tBodyAcc-std()-X 5 tBodyAcc-std()-Y 6 tBodyAcc-std()-Z
## 1         -0.9384040         -0.9200908         -0.6676833
## 2         -0.9754147         -0.9674579         -0.9449582
## 3         -0.9938190         -0.9699255         -0.9627480
## 4         -0.9947428         -0.9732676         -0.9670907
## 5         -0.9938525         -0.9674455         -0.9782950
## 6         -0.9944552         -0.9704169         -0.9653163
```

```
#Add columns for the subject and activity names.
actName <- read.csv("UCI HAR Dataset/test/y_test.txt", sep = "\n", header = FALSE);
subject <- read.csv("UCI HAR Dataset/test/subject_test.txt", sep = "\n", header = FALSE);
test[,"Activity Names"] <- actName;
test[,"Subject"] <- subject;
```

```
#Read in the train data and assign the columns to the features
train <- read.csv("UCI HAR Dataset/train/X_train.txt", sep = "", header = FALSE);
colnames(train) <- features[,1];
actName <- read.csv("UCI HAR Dataset/train/y_train.txt", sep = "\n", header = FALSE);
subject <- read.csv("UCI HAR Dataset/train/subject_train.txt", sep = "\n", header = FALSE);
train[,"Activity Names"] <- actName;
train[,"Subject"] <- subject;
```

```
#Now must merge the two data sets
mergedData <- merge(test, train, all = TRUE);

meanAndStd <- grepl("mean|std|Activity Names|Subject", names(mergedData), fixed = FALSE);

filteredData <- mergedData[,meanAndStd];

head(filteredData[,1:6])
```

```
##   1 tBodyAcc-mean()-X 2 tBodyAcc-mean()-Y 3 tBodyAcc-mean()-Z
## 1          -1.0000000        0.1775221600          0.54393929
## 2          -0.8723954        0.1546078000          0.33075342
## 3          -0.8538482        0.2053651600         -0.11634455
## 4          -0.5920043        0.1469832700          0.05256077
## 5          -0.5210621       -0.0001832748          0.10661589
## 6          -0.5038227       -0.5942073800          0.26480435
##   4 tBodyAcc-std()-X 5 tBodyAcc-std()-Y 6 tBodyAcc-std()-Z
## 1        -0.10078555        -0.12621085         0.35953802
## 2        -0.06063940        -0.30688449         0.06857798
## 3        -0.05714196        -0.07570973        -0.30853842
## 4        -0.42436336        -0.22019354        -0.70417659
## 5        -0.34411877        -0.49510932        -0.46239833
## 6        -0.70340175         0.67248690        -0.46498511
```

```
#Replace the activity numbers with the actual name
activities <- as.character(filteredData[,'Activity Names']);
activities <- gsub("1", "Walking", activities);
activities <- gsub("2", "Walking Upstairs", activities);
activities <- gsub("3", "Walking Downstairs", activities);
activities <- gsub("4", "Sitting", activities);
activities <- gsub("5", "Standing", activities);
activities <- gsub("6", "Laying", activities);
filteredData[,"Activity Names"] <- activities;
```

```
#Now we have filtered data, must make second dataset
tidyData <- aggregate(filteredData[,1:78], by = list(filteredData$`Activity Names`), mean);
tidyDataSupplement <- aggregate(filteredData[,1:78], by = list(filteredData$`Subject`), mean);
mergedTidyData <- merge(tidyData, tidyDataSupplement, all = TRUE);
colnames(mergedTidyData)[1] <- "Subject/Activity";

tidyDataSorted <- mergedTidyData[order(as.numeric(mergedTidyData$`Subject/Activity`)),];
```

```
## Warning in order(as.numeric(mergedTidyData$`Subject/Activity`)): NAs introduced
## by coercion
```

```
write.table(tidyDataSorted, file = "tidyData.csv", row.names = FALSE, sep = ",");
```