README

Samuel Bozzi Baco

29/02/2020

General Explanation

This document explains the analysis for Samsung Accelerometer data, asked to complete the activity for final project of Coursera's Johns Hopkins Getting and Cleaning Data.

The code run_analysis has seven (7) sections, being the section "0" related to opening files. In this section both *subjects* and *activities* variables receive names to not be confounded with other variables during the analysis. Also, specific objects for variables names (*acLabels* and *featNames*) are created.

```
names(yTrain) <- "activity"
```

```
names(subjTrain) <- "subject"
```

Section 1: Merging Datasets

The section 1 has the code to merge both *test* and *train* data sets. The results are stored at *dfComplete* dataframe.

```
dfComplete <- rbind(train, test)
```

Section 2: Extracting only Mean and Standard Deviation variables

The section 2 look for patterns related to Mean and Standard deviation at variable names (stored at vector *featNames*) and select only those, storing the result at *dfSelected* dataframe.

```
meanVar <- grep("mean\\(\\)", featNames$V2)

sdVar <- grep("std\\(\\)", featNames$V2)

dfSelected <- dfComplete[ ,c(meanVar, sdVar, 562:563)]
```

Section 3: Renaming activities

The section 3 renames all values at *activites* variable using the function *mapvalues* from *plyr* package.

```
dfSelected$activity <- plyr::mapvalues(dfSelected$activity ,
                                       from = sort(unique(dfSelected$activity)),
                                       to = acLabels$V2)
```

Section 4: Column names

The section 4 renames all the variables on *dfSelected* dataframe with original names found at databook supplied.

```
colnames(dfSelected)[colnames(dfSelected) %in%
                           c(names(dfSelected[ ,-c(67:68)]))] <- featNames$V2[c(meanVar, sdVar)]
```

Section 5: Average by subjects and activities

The section 5 calculates the averages by subjects and activities, using dyplr *pipes* and functions. First, the dataframe is grouped by subjects and activities, than all averages are calculated.

```
average <- dfSelected %>%
        group_by(subject, activity) %>%
        summarise_all(mean)
```

Section 6: Write dataframe

The section 6 has the code to write the dataframe as a *.txt* file.

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