

Code Book

Loading the Files and completing the data sets

Loading the Libraries that are going to be used:

```
library(dplyr) #Use for manipulating the data
```

Reading the data:

```
Xtest  <- tbl_df(read.table(file = "test/X_test.txt"))
Xtrain <- tbl_df(read.table(file = "train/X_train.txt"))

Ytest <- read.table("test/y_test.txt")
Ytrain <- read.table("train/y_train.txt")
SubjectTest <- read.table("test/subject_test.txt")
SubjectTrain <- read.table("train/subject_train.txt")
Features <- read.table("features.txt")
ActLabel <- read.table("activity_labels.txt")
```

The code bellow show how the test and train data set are completed with other information from the folder.

```
colnames(Xtest) <- Features[,2]##Add the Features for Test
colnames(Xtrain)<- Features[,2]##Add the Features for Train

Xtest$Activity <- Ytest[,1]  ##Activity ID for Test
Xtrain$Activity <- Ytrain[,1] ##Activity ID for Train
Xtest$Subject <- SubjectTest[,1] ##Subject ID for Test
Xtrain$Subject <- SubjectTrain[,1] ##Subject ID for Train
```

Assignment 1: Merging the data sets

For the first assignment we are supposed to merge the two data sets: Xtest and Xtrain

If there are duplicated variables, they are removed.

```
merge <- tbl_df(rbind(Xtest,Xtrain))
if(sum(duplicated(colnames(merge)))>0){
  merge <- merge[,!duplicated(colnames(merge))] ##Remove duplicated Variables
}
```

Assignment 2: Extract the variable of the mean and standart deviation

This chunk of code goes though the variable names and only select the ones that have mean or std in their names.

```
Xbar <- merge[,grep("[Mm][e][a][n]",colnames(merge))]#Gets all the variables that have mean
Sbar <- merge[,grep("[Ss][t][d]",colnames(merge))]#Gets all the variables that have Std
```

Assignment 3: Names the types of activities

Create an inside function in `sapply` to name and create factors for the Activity Variable

```
## [1] STANDING STANDING STANDING STANDING STANDING STANDING
## 6 Levels: LAYING SITTING STANDING WALKING ... WALKING_UPSTAIRS
```

Assignment 4: Give readable names for the variables

By the file `features_info`, there is some abbreviations

f: Frequency; t: Time ; Mag: Magnitude; Acc: Accelerator; Gyro: Gyroscope

```
names(merge) <- gsub("^f", "Frequency", names(merge))
names(merge) <- gsub("^t", "Time", names(merge))
names(merge) <- gsub("Mag", "MAgnitude", names(merge))
names(merge) <- gsub("Acc", "Accelerator", names(merge))
names(merge) <- gsub("Gyro", "Gyroscope", names(merge))
```

The participants are identified with the following code

```
merge$Subject <- as.factor(sapply(X = merge$Subject,
                                function(x){x <- paste("Participant", x, sep = " ")}))
head(merge$Subject)
```

```
## [1] Participant 2 Participant 2 Participant 2 Participant 2 Participant 2
## [6] Participant 2
## 30 Levels: Participant 1 Participant 10 Participant 11 ... Participant 9
```

Assignment 5: Create a Tidy Data with the mean by Activity and Subject

The functions `group_by` and `summarize_each` from the `dplyr` package are used to improve the process

```
merge <- group_by(merge, Subject, Activity)
TidyData <- summarize_each(merge, funs(mean))
write.table(TidyData, file = "Tidy.txt", row.names = FALSE)
```

That is the process taken for tidying the data.