Getting and cleaning Data Course Project

Hariharan

This PDF contains the code and output generated for Programming assignment in the course Find the Programming_Asignment.Rmd in the same folder as this file to interact with the code and make changes for a better learning experience The link to the final Github repository used for this project submission can be found here https://github.com/hari9-9/Getting-and-cleaning-data

place the required files in the same directoriy as the .R file

Loading Data

```
#loading the activity data
activity_test<-read.table("Y_test.txt",header=FALSE)
activity_train<-read.table("Y_train.txt",header=FALSE)

#loading the subject data
sub_train<-read.table("subject_train.txt",header = FALSE)
sub_test<-read.table("subject_test.txt",header = FALSE)

#loading features data
features_test<-read.table("X_test.txt",header = FALSE)
features_train<-read.table("X_train.txt",header = FALSE)
```

merging test and train data by row using rbind()

```
subject_merged<-rbind(sub_train,sub_test)
activity_merged<-rbind(activity_train,activity_test)
features_merged<-rbind(features_train,features_test)
```

Setting names

```
#setting names
names(subject_merged)<-c("subject")
names(activity_merged)<-c("activity")
features_name<-read.table("features.txt",header = FALSE)
head(features_name)
```

```
##   V1               V2
## 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
```

```
names(features_merged)<- features_name$V2
```

Viewing Dataset

```
head(subject_merged)
```

```
##   subject
## 1       1
## 2       1
## 3       1
## 4       1
## 5       1
## 6       1
```

```
head(activity_merged)
```

```
##   activity
## 1        5
## 2        5
## 3        5
## 4        5
## 5        5
## 6        5
```

```
#head(features_merged)
```

merge all the data

```
data_combine<-cbind(subject_merged,activity_merged)
complete_data<-cbind(data_combine,features_merged)
#head(complete_data)
```

filter by names for mean and standard deviation columns

```
subdataFeaturesNames<-features_name$V2[grep("mean\\(\\)|std\\(\\)", features_name$V2)]
selectedNames<-c(as.character(subdataFeaturesNames), "subject", "activity" )
filtered_data<-subset(complete_data,select=selectedNames)
```

labeling activities

```
filtered_data$activity <- factor(filtered_data$activity, labels= c("WALKING", "WALKING_UPSTAIRS", "WALK
#head(filtered_data)
```

Setting descreptive variable names

```
names(filtered_data)<-gsub("^t", "time", names(filtered_data))
names(filtered_data)<-gsub("^f", "frequency", names(filtered_data))
names(filtered_data)<-gsub("Acc", "Accelerometer", names(filtered_data))
names(filtered_data)<-gsub("Gyro", "Gyroscope", names(filtered_data))
names(filtered_data)<-gsub("Mag", "Magnitude", names(filtered_data))
names(filtered_data)<-gsub("BodyBody", "Body", names(filtered_data))
names(filtered_data)
```

```
##  [1] "timeBodyAccelerometer-mean()-X"
##  [2] "timeBodyAccelerometer-mean()-Y"
##  [3] "timeBodyAccelerometer-mean()-Z"
##  [4] "timeBodyAccelerometer-std()-X"
##  [5] "timeBodyAccelerometer-std()-Y"
##  [6] "timeBodyAccelerometer-std()-Z"
##  [7] "timeGravityAccelerometer-mean()-X"
##  [8] "timeGravityAccelerometer-mean()-Y"
##  [9] "timeGravityAccelerometer-mean()-Z"
## [10] "timeGravityAccelerometer-std()-X"
## [11] "timeGravityAccelerometer-std()-Y"
## [12] "timeGravityAccelerometer-std()-Z"
## [13] "timeBodyAccelerometerJerk-mean()-X"
## [14] "timeBodyAccelerometerJerk-mean()-Y"
## [15] "timeBodyAccelerometerJerk-mean()-Z"
## [16] "timeBodyAccelerometerJerk-std()-X"
## [17] "timeBodyAccelerometerJerk-std()-Y"
## [18] "timeBodyAccelerometerJerk-std()-Z"
## [19] "timeBodyGyroscope-mean()-X"
## [20] "timeBodyGyroscope-mean()-Y"
## [21] "timeBodyGyroscope-mean()-Z"
## [22] "timeBodyGyroscope-std()-X"
## [23] "timeBodyGyroscope-std()-Y"
## [24] "timeBodyGyroscope-std()-Z"
## [25] "timeBodyGyroscopeJerk-mean()-X"
## [26] "timeBodyGyroscopeJerk-mean()-Y"
## [27] "timeBodyGyroscopeJerk-mean()-Z"
## [28] "timeBodyGyroscopeJerk-std()-X"
```

```
## [29] "timeBodyGyroscopeJerk-std()-Y"
## [30] "timeBodyGyroscopeJerk-std()-Z"
## [31] "timeBodyAccelerometerMagnitude-mean()"
## [32] "timeBodyAccelerometerMagnitude-std()"
## [33] "timeGravityAccelerometerMagnitude-mean()"
## [34] "timeGravityAccelerometerMagnitude-std()"
## [35] "timeBodyAccelerometerJerkMagnitude-mean()"
## [36] "timeBodyAccelerometerJerkMagnitude-std()"
## [37] "timeBodyGyroscopeMagnitude-mean()"
## [38] "timeBodyGyroscopeMagnitude-std()"
## [39] "timeBodyGyroscopeJerkMagnitude-mean()"
## [40] "timeBodyGyroscopeJerkMagnitude-std()"
## [41] "frequencyBodyAccelerometer-mean()-X"
## [42] "frequencyBodyAccelerometer-mean()-Y"
## [43] "frequencyBodyAccelerometer-mean()-Z"
## [44] "frequencyBodyAccelerometer-std()-X"
## [45] "frequencyBodyAccelerometer-std()-Y"
## [46] "frequencyBodyAccelerometer-std()-Z"
## [47] "frequencyBodyAccelerometerJerk-mean()-X"
## [48] "frequencyBodyAccelerometerJerk-mean()-Y"
## [49] "frequencyBodyAccelerometerJerk-mean()-Z"
## [50] "frequencyBodyAccelerometerJerk-std()-X"
## [51] "frequencyBodyAccelerometerJerk-std()-Y"
## [52] "frequencyBodyAccelerometerJerk-std()-Z"
## [53] "frequencyBodyGyroscope-mean()-X"
## [54] "frequencyBodyGyroscope-mean()-Y"
## [55] "frequencyBodyGyroscope-mean()-Z"
## [56] "frequencyBodyGyroscope-std()-X"
## [57] "frequencyBodyGyroscope-std()-Y"
## [58] "frequencyBodyGyroscope-std()-Z"
## [59] "frequencyBodyAccelerometerMagnitude-mean()"
## [60] "frequencyBodyAccelerometerMagnitude-std()"
## [61] "frequencyBodyAccelerometerJerkMagnitude-mean()"
## [62] "frequencyBodyAccelerometerJerkMagnitude-std()"
## [63] "frequencyBodyGyroscopeMagnitude-mean()"
## [64] "frequencyBodyGyroscopeMagnitude-std()"
## [65] "frequencyBodyGyroscopeJerkMagnitude-mean()"
## [66] "frequencyBodyGyroscopeJerkMagnitude-std()"
## [67] "subject"
## [68] "activity"
```

Forming independent dataset to get aggregate of each subject during each activity

```
library(plyr)
Data_agg<-aggregate(. ~subject + activity, filtered_data, mean)
Data_agg<-Data_agg[order(Data_agg$subject,Data_agg$activity),]
```

Storing results in a textfile

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