Getting and cleaning Data:Project

Created by Krish Mahajan

Basic settings

```
echo = TRUE  # Always make code visible
options(scipen = 1)  # Turn off scientific notations for numbers
```

Merges the training and the test sets to create one data set.

```
trainData <- read.table("./data/train/X_train.txt")
dim(trainData)

## [1] 7352  561

trainLabel <- read.table("./data/train/y_train.txt")
table(trainLabel)

## trainLabel
##    1    2    3    4    5    6 
## 1226 1073  986 1286 1374 1407

trainSubject <- read.table("./data/train/subject_train.txt")
dim(trainSubject)

## [1] 7352    1

#test data
testData <- read.table("./data/test/X_test.txt")
dim(testData)

## [1] 2947  561

testLabel <- read.table("./data/test/y_test.txt")
table(testLabel)

## testLabel
##   1   2   3   4   5   6 
## 496 471 420 491 532 537

testSubject <- read.table("./data/test/subject_test.txt")

### Joining test & training Data
joinData <- rbind(trainData, testData)
dim(joinData)

## [1] 10299   561

joinLabel <- rbind(trainLabel, testLabel)
dim(joinLabel)
```

```
## [1] 10299     1
```

```
joinSubject <- rbind(trainSubject, testSubject)
dim(joinSubject)
```

```
## [1] 10299     1
```

Step2. Extracts only the measurements on the mean and standard

```
features <- read.table("./data/features.txt")
dim(features)
```

```
## [1] 561   2
```

```
meanStdIndices <- grep("mean\\(\\)|std\\(\\)", features[, 2])
length(meanStdIndices)
```

```
## [1] 66
```

```
joinData <- joinData[, meanStdIndices]
dim(joinData)
```

```
## [1] 10299    66
```

```
names(joinData) <- gsub("\\(\\)", "", features[meanStdIndices, 2]) # remove
"()"
names(joinData) <- gsub("mean", "Mean", names(joinData)) # capitalize M
names(joinData) <- gsub("std", "Std", names(joinData)) # capitalize S
names(joinData) <- gsub("-", "", names(joinData)) # remove "-" in column
names
names(joinData)
```

```
##  [1] "tBodyAccMeanX"            "tBodyAccMeanY"
##  [3] "tBodyAccMeanZ"            "tBodyAccStdX"
##  [5] "tBodyAccStdY"             "tBodyAccStdZ"
##  [7] "tGravityAccMeanX"         "tGravityAccMeanY"
##  [9] "tGravityAccMeanZ"         "tGravityAccStdX"
## [11] "tGravityAccStdY"          "tGravityAccStdZ"
## [13] "tBodyAccJerkMeanX"        "tBodyAccJerkMeanY"
## [15] "tBodyAccJerkMeanZ"        "tBodyAccJerkStdX"
## [17] "tBodyAccJerkStdY"         "tBodyAccJerkStdZ"
## [19] "tBodyGyroMeanX"           "tBodyGyroMeanY"
## [21] "tBodyGyroMeanZ"           "tBodyGyroStdX"
## [23] "tBodyGyroStdY"            "tBodyGyroStdZ"
## [25] "tBodyGyroJerkMeanX"       "tBodyGyroJerkMeanY"
## [27] "tBodyGyroJerkMeanZ"       "tBodyGyroJerkStdX"
## [29] "tBodyGyroJerkStdY"        "tBodyGyroJerkStdZ"
## [31] "tBodyAccMagMean"          "tBodyAccMagStd"
## [33] "tGravityAccMagMean"       "tGravityAccMagStd"
## [35] "tBodyAccJerkMagMean"      "tBodyAccJerkMagStd"
## [37] "tBodyGyroMagMean"         "tBodyGyroMagStd"
## [39] "tBodyGyroJerkMagMean"     "tBodyGyroJerkMagStd"
## [41] "fBodyAccMeanX"            "fBodyAccMeanY"
## [43] "fBodyAccMeanZ"            "fBodyAccStdX"
## [45] "fBodyAccStdY"             "fBodyAccStdZ"
## [47] "fBodyAccJerkMeanX"        "fBodyAccJerkMeanY"
## [49] "fBodyAccJerkMeanZ"        "fBodyAccJerkStdX"
## [51] "fBodyAccJerkStdY"         "fBodyAccJerkStdZ"
## [53] "fBodyGyroMeanX"           "fBodyGyroMeanY"
## [55] "fBodyGyroMeanZ"           "fBodyGyroStdX"
## [57] "fBodyGyroStdY"            "fBodyGyroStdZ"
## [59] "fBodyAccMagMean"          "fBodyAccMagStd"
## [61] "fBodyBodyAccJerkMagMean"  "fBodyBodyAccJerkMagStd"
## [63] "fBodyBodyGyroMagMean"     "fBodyBodyGyroMagStd"
## [65] "fBodyBodyGyroJerkMagMean" "fBodyBodyGyroJerkMagStd"
```

Step3. Uses descriptive activity names to name the activities in

```
activity <- read.table("./data/activity_labels.txt")
head(activity)
```

```
##   V1                 V2
## 1  1            WALKING
## 2  2   WALKING_UPSTAIRS
## 3  3 WALKING_DOWNSTAIRS
## 4  4            SITTING
## 5  5           STANDING
## 6  6             LAYING
```

```
dim(activity)
```

```
## [1] 6 2
```

```
activity[, 2] <- tolower(gsub("_", "", activity[, 2])) # remove '-' and lower
substr(activity[2, 2], 8, 8) <- toupper(substr(activity[2, 2], 8, 8))
#codingConventions
substr(activity[3, 2], 8, 8) <- toupper(substr(activity[3, 2], 8, 8))
#codingConventions

activityLabel <- activity[joinLabel[, 1], 2] #interesting step
joinLabel[, 1] <- activityLabel
names(joinLabel) <- "activity"
```

Step4. Appropriately labels the data set with descriptive activity

```
names(joinSubject) <- "subject"
cleanedData <- cbind(joinSubject, joinLabel, joinData)
dim(cleanedData)
```

```
## [1] 10299    68
```

```
write.table(cleanedData, "merged_data.txt") # write out the 1st dataset
```

Step5. Creates a second, independent tidy data set with the average of each variable for each activity and each subject.

```r
subjectLen <- length(table(joinSubject))
activityLen <- dim(activity)[1]
columnLen <- dim(cleanedData)[2]
result <- matrix(NA, nrow=subjectLen*activityLen, ncol=columnLen)
result <- as.data.frame(result)
colnames(result) <- colnames(cleanedData)
row <- 1
for(i in 1:subjectLen) {
    for(j in 1:activityLen) {
        result[row, 1] <- sort(unique(joinSubject)[, 1])[i]
        result[row, 2] <- activity[j, 2]
        bool1 <- i == cleanedData$subject
        bool2 <- activity[j, 2] == cleanedData$activity
        result[row, 3:columnLen] <- colMeans(cleanedData[bool1&bool2,
3:columnLen])
        row <- row + 1
    }
}
write.table(result, "data_with_means.txt") # write out the 2nd dataset
```