README

Contents

Getting and Cleaning Data Course Project

The purpose of this project is to demonstrate my ability to collect, work with, and clean a data set. The goal is to prepare tidy data that can be used for later analysis. This repo is provided for peer review. Below are the scripts used with explanations on what they do broken up in a temporal sequence from beginning to end.

Package and Library Check

This section is used to ensure the dplyr package is installed and the proper library is loaded for use.

```r
# check to see if the dplyr package is installed and the library is loaded
# if anything is missing, install and load as needed
print("Loading packages and libraries if needed...")
if (!require("dplyr")) {
  install.packages("dplyr")
  library("dplyr")
} else {
  library("dplyr")
```

File Check

Next we check to see if the file is available and the raw data is in the proper folder.

```r
if (!file.exists("UCI HAR Dataset")) {
  # let the user know we are getting the file
  print("Setting up the raw data file(s) now...")

  # variable to hold our file name that we will use for analysis
  dataFileName <- "harData.zip"

  ## Download the file used for our analysis if we don't already have it
  if (!file.exists(dataFileName)){
    fileURL <- "https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip"
    download.file(fileURL, dataFileName, method = "curl")
  }
```

Data Frame Creation

Now we load all of the data into our data frames using read.table since it is more flexible with delimited files that don't use commas.

```
activities <- read.table("UCI HAR Dataset/activity_labels.txt", col.names = c("code", "activity"))
subject_test <- read.table("UCI HAR Dataset/test/subject_test.txt", col.names = "subject")
x_test <- read.table("UCI HAR Dataset/test/X_test.txt", col.names = features$functions)
y_test <- read.table("UCI HAR Dataset/test/y_test.txt", col.names = "code")
subject_train <- read.table("UCI HAR Dataset/train/subject_train.txt", col.names = "subject")
x_train <- read.table("UCI HAR Dataset/train/X_train.txt", col.names = features$functions)
y_train <- read.table("UCI HAR Dataset/train/y_train.txt", col.names = "code")
```

Data Frame Combination

Using our data frames from above, we combine them to make our analysis easier.

```
xBind <- rbind(x_train, x_test)
yBind <- rbind(y_train, y_test)
subjectBind <- rbind(subject_train, subject_test)
combinedData <- cbind(subjectBind, xBind, yBind)
```

Data Clean Up

Now the lengthy process of cleaning up the data begins with a new variable called cleanData. First, we extract out the the mean and std deviation from the combined data.

```
cleanData <- combinedData %>% select(subject, code, contains("mean"), contains("std"))
```

Then we take the activities column and replace the numeric codes with friendly names for the activity that took place.

```
cleanData$code <- activities[cleanData$code,2]
```

Finally, we fix all the column names to make them more readable.

```
names(cleanData)[1] = "Subject"
names(cleanData)[2] = "Activity"
names(cleanData)<-gsub("\\.", "", names(cleanData)) # get rid of all periods in the names
names(cleanData)<-gsub("Acc", "Accelerometer_", names(cleanData))
names(cleanData)<-gsub("Gyro", "Gyroscope_", names(cleanData))
names(cleanData)<-gsub("BodyBody", "Body_", names(cleanData))
names(cleanData)<-gsub("Body", "Body_", names(cleanData))
names(cleanData)<-gsub("Mag", "Magnitude_", names(cleanData))
names(cleanData)<-gsub("Time", "Time_", names(cleanData))
names(cleanData)<-gsub("^t", "Time_", names(cleanData))
names(cleanData)<-gsub("Freq", "Frequency_", names(cleanData))
names(cleanData)<-gsub("^f", "Frequency_", names(cleanData))
names(cleanData)<-gsub("tBody", "Time_Body_", names(cleanData))
names(cleanData)<-gsub("-mean()", "Mean_", names(cleanData), ignore.case = TRUE)
names(cleanData)<-gsub("mean", "Mean_", names(cleanData), ignore.case = TRUE)
names(cleanData)<-gsub("Jerk", "Jerk_", names(cleanData), ignore.case = TRUE)
names(cleanData)<-gsub("-std()", "STD_", names(cleanData), ignore.case = TRUE)
names(cleanData)<-gsub("-freq()", "Frequency_", names(cleanData), ignore.case = TRUE)
names(cleanData)<-gsub("angle", "Angle_", names(cleanData))
names(cleanData)<-gsub("[Gg]ravity", "Gravity_", names(cleanData))
names(cleanData)<-gsub("X", "X", names(cleanData))
```

```r
names(cleanData)<-gsub("Y", "Y", names(cleanData))
names(cleanData)<-gsub("_$", "", names(cleanData)) # clean up any underscores at the end of the text
```

Environment Clean Up

Remove all but the final two variables that will be used for analysis to leave a clean environment for analysis. These lines can be commented out if the variables are desired.

```r
remove(activities)
remove(combinedData)
remove(features)
remove(subject_test)
remove(subject_train)
remove(subjectBind)
remove(x_test)
remove(x_train)
remove(xBind)
remove(y_test)
remove(y_train)
remove(yBind)
```

Variable Averages By Activity and Subject

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

```r
avgCleanData <- cleanData %>%
  group_by(Subject, Activity) %>%
  summarize_all(list(mean))
```

Original Instructions from Coursera

The original instructions for this project can be found on the Coursera website found here: https://www.coursera.org/learn/data-cleaning/peer/FIZtT/getting-and-cleaning-data-course-project