codeBook

Project description

The purpose of this project is to demonstrate your ability to collect, work with, and clean a data set. The goal is to prepare tidy data that can be used for later analysis. You will be graded by your peers on a series of yes/no questions related to the project. You will be required to submit: 1) a tidy data set as described below, 2) a link to a Github repository with your script for performing the analysis, and 3) a code book that describes the variables, the data, and any transformations or work that you performed to clean up the data called CodeBook.md. You should also include a README.md in the repo with your scripts. This repo explains how all of the scripts work and how they are connected.

One of the most exciting areas in all of data science right now is wearable computing - see for example this article . Companies like Fitbit, Nike, and Jawbone Up are racing to develop the most advanced algorithms to attract new users. The data linked to from the course website represent data collected from the accelerometers from the Samsung Galaxy S smartphone. A full description is available at the site where the data was obtained:

http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

Here are the data for the project:

https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip

Project Assigments

You should create one R script called run_analysis.R that does the following.

- Merges the training and the test sets to create one data set.
- Extracts only the measurements on the mean and standard deviation for each measurement.
- Uses descriptive activity names to name the activities in the data set
- Appropriately labels the data set with descriptive variable names.
- From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject.

Solution

Load libraries

```
library(dplyr)
library(plyr)
```

Load and unzip file

```
url <- "https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip"
if(!file.exists("./data")) dir.create("./data")
if(!file.exists("./data/Week4-Project-Data.zip")) download.file(url, destfile="./data/Week4-Project-Data
if(!file.exists("./data/UCI HAR Dataset")) unzip(zipfile = "./data/Week4-Project-Data.zip", exdir = "./
```

Extract data frames

```
feature <- read.table("./data/UCI HAR Dataset/features.txt", col.names = c("id_feature","obs"))
activity <- read.table("./data/UCI HAR Dataset/activity_labels.txt", col.names = c("id_activity", "acti

testX <- read.table("./data/UCI HAR Dataset/test/X_test.txt", col.names = feature$obs)
testY<- read.table("./data/UCI HAR Dataset/test/y_test.txt", col.names = "id")

trainX <- read.table("./data/UCI HAR Dataset/train/X_train.txt", col.names = feature$obs)
trainY <- read.table("./data/UCI HAR Dataset/train/y_train.txt", col.names = "id")

testSubject <- read.table("./data/UCI HAR Dataset/test/subject_test.txt", col.names = "id_subject")
trainSubject <- read.table("./data/UCI HAR Dataset/train/subject_train.txt", col.names = "id_subject")
```

Merge test and train data frames

```
dataX <- rbind(testX, trainX)
dataY <- rbind(testY, trainY)
dataSubject <- rbind(testSubject, trainSubject)
```

Assignment 1: Merges test and train sets to subject identification to create one tidy data set

```
dataMerged <- cbind(dataX, dataY, dataSubject)
```

Assignment 2: Extracts only the measurements on the mean and standard deviation for each measurement

```
dataMeanStd <- select(dataMerged, id_subject, id, contains("mean"), contains("std"))
```

Assingment 3: Uses descriptive activity names to name the activities in the data set

```
dataMeanStd$id <- activity[dataMeanStd$id, 2]
```

Assingment 4: Appropriately labels the data set with descriptive variable names

Acc to Accelerometer

```
names(dataMeanStd)<-gsub("Acc", "Accelerometer", names(dataMeanStd))
```

Gyro to Gyroscope

```
names(dataMeanStd)<-gsub("Gyro", "Gyroscope", names(dataMeanStd))
```

Mag to Magnitude

```
names(dataMeanStd)<-gsub("Mag", "Magnitude", names(dataMeanStd))
```

t to Time

```
names(dataMeanStd)<-gsub("^t", "Time", names(dataMeanStd))
names(dataMeanStd)<-gsub("\\.t", "Time", names(dataMeanStd))
```

f to Frequency

```
names(dataMeanStd)<-gsub("^f", "Frequency", names(dataMeanStd))
```

-mean to Mean

```
names(dataMeanStd)<-gsub(".mean()", "Mean", names(dataMeanStd))
```

-std to Std

```
names(dataMeanStd)<-gsub(".std()", "Std", names(dataMeanStd))
```

id to Activity

```
names(dataMeanStd)<-gsub("^id$", "Activity", names(dataMeanStd))
```

id_subject to Subject

```
names(dataMeanStd)<-gsub("^id_subject$", "Subject", names(dataMeanStd))
```

angle to Angle

```
names(dataMeanStd)<-gsub("angle", "Angle", names(dataMeanStd))
```

BodyBody to Body

```
names(dataMeanStd)<-gsub("BodyBody", "Body", names(dataMeanStd))
```

Remove Dots

```
names(dataMeanStd)<-gsub("\\.", "", names(dataMeanStd))
```

gravity to Gravity

```
names(dataMeanStd)<-gsub("gravity", "Gravity", names(dataMeanStd))
```

Assingment 5: From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject

```
dataMergedAverage<-aggregate(. ~Subject + Activity, dataMeanStd, mean)
dataMergedAverage<-dataMergedAverage[order(dataMergedAverage$Subject,dataMergedAverage$Activity),]
write.table(dataMergedAverage, "./data/finalAverageData.txt", row.name=FALSE)
```