Getting and Cleaning Data Project

Daniel Rosquete

February 18, 2016

Instructions

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:

Data Description

Here are the data for the project:

Data Set

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 dataset with the average of each variable for each activity and each subject.

Getting the Party Started

To achieve the goal, is important to be organized and set all the steps in order, the first step is getting the data

1. Getting the data

1.1 Set the working directory

Watch out! This one is mine, if you would like to use this Script, you should set yours ;-)

```
setwd("C:/Users/Daniel/MachineLearning/Data Science/2 - LimpiandoDataConR/Proyecto/")
```

1.2 Download the important Files

```
fileURL<-"https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip"
download.file(fileURL,destfile="Dataset.zip",method="curl")
```

1.3 Unzip the dataset

```
unzip(zipfile="Dataset.zip")
```

1.4 Include the libraries

```
library(data.table)
library(reshape2)
```

1.5 Checking the Readme

Basically the readme file gives a brief description of the records and the files with their locations. It is really important reading the Readme files in the dataset So, you can understand the architecture.

In this particular case, the file distribution is as follows:

- The features names are in **features.txt**
- The features data are in **X_train.txt** and **X_test.txt**
- The activities are defined in **activity_labels.txt** and are used on **Y_train.txt** and **Y_test.txt**
- The subject data are in **subject_train.txt** and **subject_test.txt**

1.6 Reading the files

Here all the data files will be loaded first, then the label file

1.6.1 Features Data

```
dataFeaturesTrain <- read.table("./UCI HAR Dataset/train/X_train.txt")
dataFeaturesTest  <- read.table("./UCI HAR Dataset/test/X_test.txt")
```

1.6.2 Activity Data

```
dataActivityTrain <- read.table("./UCI HAR Dataset/train/y_train.txt")
dataActivityTest <- read.table("./UCI HAR Dataset/test/y_test.txt")
```

1.6.3 Subject Data

```
dataSubjectTrain <- read.table("./UCI HAR Dataset/train/subject_train.txt")
dataSubjectTest  <- read.table("./UCI HAR Dataset/test/subject_test.txt")
```

1.6.4 Features Names and activity labels

```
#Obtain only the names, that´s why i´m using [,2]
featuresNames <- read.table("./UCI HAR Dataset/features.txt")[,2]
activityLabels <- read.table("./UCI HAR Dataset/activity_labels.txt")[,2]
```

1.7 Extracting the expected measurements

Saving a logical vector where all the positions related to mean and std will be marked. Also saving all the names of the features that I will be using later

```
importantFeatures <- grepl("mean|std",featuresNames)
featuresNames <- grep("mean|std",featuresNames,value = TRUE)
```

2. The first 4 steps and merging

Now that all the relevant data is loaded into R workspace, Let´s work on the merge.

2.1 Keeping the important Data

Deleting all the features that are not relevant for the study

```
dataFeaturesTrain <- dataFeaturesTrain[,importantFeatures]
dataFeaturesTest <- dataFeaturesTest[,importantFeatures]
```

2.2 Assigning Labels to the Activities and more

```
dataActivityTrain[,2] <- activityLabels[dataActivityTrain[,1]]
dataActivityTest[,2] <- activityLabels[dataActivityTest[,1]]
names(dataActivityTrain) <- c("idActivity","activName")
names(dataActivityTest) <- c("idActivity","activName")

#Assigning an ID to the subjects
names(dataSubjectTrain) <- "IDSubject"
names(dataSubjectTest) <- "IDSubject"
#Assigning the names to the Features
names(dataFeaturesTrain) <- featuresNames
names(dataFeaturesTest) <- featuresNames
```

2.3 Binding the Data

Now that the data is well setted and the column names standarized. We can proceed with the binding of the entire dataset

```
trainDataSet<-cbind(as.data.table(dataSubjectTrain),dataActivityTrain,dataFeaturesTrain)
testDataSet<-cbind(as.data.table(dataSubjectTest),dataActivityTest,dataFeaturesTest)
```

2.4 Finally!!! Merging the datasets

```
mergedDataSet<-rbind(trainDataSet,testDataSet)
```

3 Creating a new dataset with the means

I´m using the aggregate because it is simpler to combine the datasets with the mean and order them after that. Also, I´m writing a csv because it is nicer to read

```
tidyData <- aggregate(. ~IDSubject + activName,mergedDataSet,mean)
tidyData <- tidyData[order(tidyData$IDSubject,tidyData$activName),]
write.csv(tidyData,file="tidyData.csv")
```