Getting and Cleaning Data - Assignment

Myron Keith Gibert Jr

July 13, 2020

Contents

Correspondence

Please address any questions to Myron Keith Gibert Jr at mkgibertjr@msn.com. Code for this project is stored in a GitHub repository.

Introduction

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. I was graded by my peers on a series of yes/no questions related to the project. I was required to submit: 1) a tidy data set as described below, 2) a link to a Github repository with my script for performing the analysis, and 3) a code book that describes the variables, the data, and any transformations or work that I performed to clean up the data called CodeBook.md. I also included a README.md in the Github repository (repo) with my 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

I created an R script called run_analysis.R that does the following.

1. Merges the training and the test sets to create one data set.
2. Extracts only the measurements on the mean and standard deviation for each measurement.
3. Uses descriptive activity names to name the activities in the data set
4. Appropriately labels the data set with descriptive variable names.
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.

Set Parameters

```
#set the directory that contains the files
zipfile <- "ASN_rcleaning_data.zip"
#set output directory? Default: outputdir <- "assignment1outputs"
outputdir <- "assignmentoutputs"
#Overwrite contents of the output directory? Default: deleteoutputs <- FALSE
deleteoutputs <- TRUE
#Delete specdata/ directory after completing the analysis? Default: deletespec <- TRUE
deletespec <- TRUE
```

Debug

The debug chunk will prevent the script from running if any of the dependent variables for this analysis do not exist. This should prevent the program from erroring out after a long runtime without producing any results due to a missing variable. If modifying the input .csv and .xlsx files, it is important to leave all header information and column names intact, as the program uses this information to extract relevant data. Columns are intuitively labeled to end user convenience.

```
if (dir.exists(outputdir) && deleteoutputs == FALSE ){
  stop("Your output directory already exists!  Please delete/move
       this folder from your working directory.  Alternatively, you
       can set 'deleteoutputs' to TRUE to auto-delete this folder
       for every run. You may also choose an alternative output
       directory.")
}else{
  unlink(outputdir,recursive = TRUE)
}

if (!exists("outputdir")){
stop("outputdir variable is not defined.  Please ensure that all
     parameters in the r parameters chunk are defined.")
}

if (!exists("deleteoutputs")){
```

```
stop("deleteoutputs variable is not defined.  Please ensure that all
     parameters in the r parameters chunk are defined.")
}

if (!exists("deletespec")){
stop("deletespec variable is not defined.  Please ensure that all
     parameters in the r parameters chunk are defined.")
}

if (!dir.exists(outputdir)){dir.create(outputdir)}
```

Data

The zip file containing the data for this assignment can be downloaded here:

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

I have renamed the file to "ASN_rcleaning_data.zip" for organization.

Unzipping the data

For this programming assignment I needed to unzip this file and create the directory 'ASN_rcleaning_data'. Once I unzipped the zip file, I did not make any modifications to the files in the 'ASN_rcleaning_data' directory.

```
if (!file.exists(zipfile)){
  fileURL <- "https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip "
  download.file(fileURL, zipfile, method="curl")
}
if(!dir.exists("UCI HAR Dataset")){
  unzip(zipfile)
}
```

Reading in the data

```
activity_labels <- read.table("UCI HAR Dataset/activity_labels.txt")
features <- read.table("UCI HAR Dataset/features.txt")

activity_labels[,2] <- as.character(activity_labels[,2])
features[,2] <- as.character(features[,2])

X_train <- read.table("UCI HAR Dataset/train/X_train.txt")
y_train <- read.table("UCI HAR Dataset/train/y_train.txt")
subject_train <- read.table("UCI HAR Dataset/train/subject_train.txt")

X_test <- read.table("UCI HAR Dataset/test/X_test.txt")
y_test <- read.table("UCI HAR Dataset/test/y_test.txt")
subject_test <- read.table("UCI HAR Dataset/test/subject_test.txt")
```

Alternative: Reading in the data

```
i = 1

filenames.uci <-  list.files(path = "UCI HAR Dataset",pattern = "\\.txt$")

for(i in 1:length(filenames.uci)){
  tryCatch({
    print(filenames.uci[i])
    varname <- gsub('.txt', '', paste(filenames.uci[i]))
    datafile <- paste("UCI HAR Dataset",filenames.uci[i],sep="/")
    assign(paste(varname), read.table(datafile))
  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")})
}
rm(varname)
rm(datafile)

filenames.test <- list.files(path = paste("UCI HAR Dataset","test",sep="/"),pattern = "\\.txt$")

for(i in 1:length(filenames.test)){
  tryCatch({
    print(filenames.test[i])
    varname <- gsub('.txt', '', paste(filenames.test[i]))
    datafile <- paste("UCI HAR Dataset","test",filenames.test[i],sep="/")
    assign(paste(varname), read.table(datafile))
  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")})
}
rm(varname)
rm(datafile)

filenames.train <- list.files(path = paste("UCI HAR Dataset","train",sep="/"),pattern = "\\.txt$")

for(i in 1:length(filenames.train)){
  tryCatch({
    print(filenames.train[i])
    varname <- gsub('.txt', '', paste(filenames.train[i]))
    datafile <- paste("UCI HAR Dataset","train",filenames.train[i],sep="/")
    assign(paste(varname), read.table(datafile))
  }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")})
}
rm(varname)
rm(datafile)
```

Step 1: Merge training and test sets to create one data set.

```
train <- cbind(subject_train,y_train,X_train)

test <- cbind(subject_test,y_test,X_test)

allData <- rbind(train,test)
colnames(allData) <- c("subject","activity",features[,2])
```

Step 2: Use descriptive activity names to name the activities in the data set

```
# group the activity column of dataSet, re-name lable of levels with activity_levels, and apply it to d
allData$activity <- factor(allData$activity, levels = activity_labels[,1], labels = activity_labels[,2])
allData$subject <- as.factor(allData$subject)
```

Step 3: Extract only the mean and standard deviation for each measurement

```
MeanStdOnly <- grep("mean()|std()", colnames(allData))
subject_act <- grep("subject|activity", colnames(allData))
dataSet <- allData[,c(subject_act,MeanStdOnly)]
```

Step 4: Appropriately labels the data set with descriptive activity names

```
# Create vector of "Clean" feature names by getting rid of "()" apply to the dataSet to rename labels.
CleanFeatureNames <- sapply(features[, 2], function(x) {gsub("[()]", "",x)})
names(dataSet) <- c("subject","activity",CleanFeatureNames[MeanStdOnly])
```

Step 5: Creates a second, tidy data set with the average of each variable for each activity and each subject.

```
# melt data to tall skinny data and cast means. Finally write the tidy data to the working directory as
baseData <- melt(dataSet,(id.vars=c("subject","activity")))
secondDataSet <- dcast(baseData, subject + activity ~ variable, mean)
names(secondDataSet)[-c(1:2)] <- paste("[mean of]" , names(secondDataSet)[-c(1:2)] )
write.csv(secondDataSet, paste(outputdir,"/tidy_data.csv",sep=""),row.names=FALSE)
```

Cleanup

This final command removes the unzipped "UCI HAR Dataset" directory if the deletespec variable is set to TRUE. This reduces the overall storage burden of this project by removing the files that we no longer need access to. The zipped file remains in the working directory, so "UCI HAR Dataset" will be recreated whenever I use the command in line 100 (unzip) if it is deleted during this step.

```
if(deletespec == TRUE){unlink("UCI HAR Dataset",recursive = TRUE)}
```