run\_analysis.R

reetu

2020-04-10

setwd("C:\\Rfiles\\GettingAndCleaningData")  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

filename <- "Coursera\_DS3\_Final.zip"  
  
# Checking if archieve already exists.  
if (!file.exists(filename)){  
 fileURL <- "https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip"  
 download.file(fileURL, filename, method="curl")  
}   
  
# Checking if folder exists  
if (!file.exists("UCI HAR Dataset")) {   
 unzip(filename)   
}  
  
features <- read.table("UCI HAR Dataset/features.txt", col.names = c("n","functions"))  
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")  
  
# 1. Merges the training and the test sets to create one data set.  
x<-rbind(x\_test,x\_train)  
y<-rbind(y\_test,y\_train)  
sub<-rbind(subject\_test,subject\_train)  
merged<-cbind(x,y,sub)  
  
  
# 2. Extracts only the measurements on the mean and   
# standard deviation for each measurement.  
tidyset<-merged %>% select(subject, code, contains("mean"), contains("std"))  
  
  
# 3. Uses descriptive activity names to name the activities in the data set  
tidyset$code<-activities[merged$code,2]  
  
# 4. Appropriately labels the data set with descriptive variable names.  
names(tidyset)[2] = "activity"  
names(tidyset)<-gsub("Acc", "Accelerometer", names(tidyset))  
names(tidyset)<-gsub("Gyro", "Gyroscope", names(tidyset))  
names(tidyset)<-gsub("BodyBody", "Body", names(tidyset))  
names(tidyset)<-gsub("Mag", "Magnitude", names(tidyset))  
names(tidyset)<-gsub("^t", "Time", names(tidyset))  
names(tidyset)<-gsub("^f", "Frequency", names(tidyset))  
names(tidyset)<-gsub("tBody", "TimeBody", names(tidyset))  
names(tidyset)<-gsub("-mean()", "Mean", names(tidyset), ignore.case = TRUE)  
names(tidyset)<-gsub("-std()", "STD", names(tidyset), ignore.case = TRUE)  
names(tidyset)<-gsub("-freq()", "Frequency", names(tidyset), ignore.case = TRUE)  
names(tidyset)<-gsub("angle", "Angle", names(tidyset))  
names(tidyset)<-gsub("gravity", "Gravity", names(tidyset))  
  
# 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.  
  
FinalData <- tidyset %>%  
 group\_by(subject, activity) %>%  
 summarise\_all(funs(mean))

## Warning: funs() is soft deprecated as of dplyr 0.8.0  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## This warning is displayed once per session.

write.table(FinalData, "FinalData.txt", row.name=FALSE)