CodeBook.md

EFon

## Analysis of wearable smarthphone data to evaluate 30 volunteers engaged in movements or activities of daily living.

## This analysis will focus on the data collected for the mean and std deviation.

## Initial Setup includes installation of certain packages.

## The data have been separated into training and test folders. The two will be merged. The data also contain non-descriptive variable names which will be made descriptive using the “features.txt” file. Here we find the variables referring to “mean” and “std” for standard deviation. We work with observations from these variables.

## We conclude by preparing a tidyset called “cleanset” and use this to calculate averages of our observations grouped by Subjects(“Persons”) then Activities(“Activ”).

## install.packages(“tidyverse”)

## set working directory to wearfolder or location of downloaded files.

install.packages("tidyverse") # Packages needed for analysis  
  
library(tidyverse)  
library(readr)  
library(dplyr)  
  
#The first part loads the data into r and renames columns appropriately. We collect 6 data frames and combine to 1.  
  
#Question1  
  
X\_train <- read.table("train/X\_train.txt") # From the train folder, load into r "subject\_train.txt", "X\_train.txt", # "y\_train.txt". The same is repeated for the test folder  
  
y\_train <- read.table("train/y\_train.txt") %>%  
  
 rename(Activ = V1) # Load data into r and rename column "Activ"  
  
subject\_train <- read.table("train/subject\_train.txt") %>%  
  
 rename(Persons = V1 )  
  
X\_test <- read.table("test/X\_test.txt")  
  
y\_test <- read.table("test/y\_test.txt") %>%  
  
 rename(Activ = V1)  
  
subject\_test <- read.table("test/subject\_test.txt") %>%  
  
 rename(Persons = V1)  
  
##jtrain <- left\_join(X\_train, Y\_train, by = "V1")  
  
xytrainbind <- cbind(y\_train, X\_train)  
  
xysubTrainbind <- cbind(subject\_train, xytrainbind)  
  
xytestbind <- cbind(y\_test, X\_test)  
  
xysubTestbind <- cbind(subject\_test, xytestbind)  
  
comboset <- rbind(xysubTrainbind, xysubTestbind)  
  
  
## This is the final data frame containing test and training data for all volunteers.  
  
#dim(comboset)

## Next we analyse the descriptive names in file “features.txt”. These will be used to subset the variable containing the mean and standard deviation.

## Here we select every variable containg the word or “mean” or “std” (standard deviation) and we do no separate these depending on the activity. We separate names from “tBodyAcc-mean()-X” to “tBodyAcc Mean X” components.

# Question2  
  
#library(readr)  
  
features\_colnames <- read\_table("features.txt", col\_names = FALSE) %>% # read the data into r  
   
 ### view(features\_colnames)  
   
 separate(col = X1, into = c("Position", "typre", "stat", "axis")) %>%  
   
 # separate features description into several columns to index the "mean" and "std".  
   
 filter(stat == "mean" | stat == "std") # Select only those rows that contain "mean" or "std"  
   
 # # view()  
 #   
 # # xycol <- cbind(ytest, xtest)  
 #   
 # # xysubcol <- cbind(subject\_test, xycol)  
 #   
 # # all\_sets <- rbind(xysubcol, xysubcol2)  
  
  
# We create a vector of numbers which point to the positions of variables "mean" and "std" in the combined dataframe.  
   
featcol <- features\_colnames %>%   
 select(Position) %>%   
 sapply(as.numeric) %>%  
 tbl\_df() %>%  
 mutate(Position = Position + 2) %>%   
 unlist() %>%  
 as.vector()   
  
meanANDstd <- comboset[c(featcol)] #%>%   
  
# The data frame is indexed to select only columns for "mean" and "std".  
   
 #view()  
   
  
  
#Question 3  
  
# set working directory as appropriate  
  
# use comboset from previous analysis  
  
activ\_labels <- read.table("activity\_labels.txt") ## load activity labels into r  
  
actlab <- rename(activ\_labels, Activ = V1) ## rename col1 V1 of activ\_labels to same key as combined dataset,   
 ## comboset, key = "Activ"  
 ##view(actlab)  
  
  
activSet <- right\_join(actlab, comboset, by = "Activ") ## keep all observations of comboset and match to actlab  
 ## combine to one data frame  
 ## view(head(activSet))  
  
# Question 4  
  
## Using features\_colnames  
  
unifeat <- unite(features\_colnames, "namez", typre:axis, sep = " ") ## Join the columns to have a single name  
  
 ## view(unifeat)  
  
 ## comboset %>% select(-(1:2))   
 ## %>% head() %>% view()  
  
meancombo <- meanANDstd   
   
meancombo2 <- tbl\_df(meancombo) ## use a tibble for easier processing  
  
 ## view(head(unifeat))  
   
col\_names <- unifeat %>% ##select the column of unifeat data frame that will become names of variables  
 select(namez) %>%   
 tbl\_df()   
col\_names2 <- as.vector(unlist(col\_names))  
   
  
colnames(meancombo2) <- col\_names2 ## replace variable names with descriptive names  
   
 ## view(head(meancombo2))  
  
## Question 5  
  
# featcol2 <- features\_colnames %>%   
# select(Position) %>%   
# sapply(as.numeric) %>%  
# tbl\_df() %>%  
# ## mutate(Position = Position + 2) %>%   
# unlist() %>%  
# as.vector() ## %>%  
# ## view()  
  
featcol3 <- tbl\_df(featcol)  
featcol4 <- add\_row(featcol3, value = 1:2, .before = TRUE) %>% ## add two rows with values 1&2 to be used  
 unlist() %>% ## for "Persons" & "Activ" variable names  
 as.vector()  
  
meancombine15 <- comboset[c(featcol4)] #%>% ##index the columns for selected variables  
   
 # view()  
   
  
 meancombine19 <- meancombine15  
  
   
col\_names44 <- unifeat %>% ## create new column names vector to replace variable names  
 select(namez) %>%   
 tbl\_df() %>%  
 add\_row(namez = c("Persons", "Activ"), .before = TRUE) %>%  
 unlist() %>%  
 as.vector()  
  
  
  
colnames(meancombine19) <- col\_names44 # change to descriptive variable names  
  
cleanset <- meancombine19 # The final tidy data   
  
 ## view(cleanset)  
  
## Q5b  
  
 lesommer <- cleanset %>% ## group first by subjects then Activities then calculate average for each.  
   
 group\_by(Persons, Activ) %>%  
   
 summarise\_all( mean, na.rm = TRUE) %>%  
   
 view()  
   
 # lesommer2 <- cleanset %>%  
 # group\_by(Persons, Activ) %>%  
 # summarise(  
 # count = n()  
 # ) %>%  
 # view()

## Please see READMe

## Thank You!