run\_analysis.R

s\_burt

Sat Oct 25 14:23:31 2014

##\*\*\*\*\*\*\*\*\*\*\*\*Acknowledgements\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
#[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. International Workshop of Ambient Assisted Living (IWAAL 2012). Vitoria-Gasteiz, Spain. Dec 2012  
#This dataset is distributed AS-IS and no responsibility implied or explicit can be addressed to the authors or their institutions for its use or misuse. Any commercial use is prohibited.  
#Jorge L. Reyes-Ortiz, Alessandro Ghio, Luca Oneto, Davide Anguita. November 2012.  
#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
# Created by Simon Burt as part of the Coursea Getting & Cleaning Data coursework - 24/10/2014  
  
#For testing clearing workspace  
#Add code to clear any previous envirmoment varibles set  
#rm(list = ls(all = TRUE))   
  
##Install pylyr package if required  
if(!is.element("plyr", installed.packages()[,1])){  
 print("Installing plyr packages")  
 install.packages("plyr")  
}  
##load pylr package  
library(plyr)  
  
##Install data.table package if required  
if(!is.element("data.table", installed.packages()[,1])){  
 print("Installing data.table packages")  
 install.packages("data.table")  
}  
##Load data.table package  
library(data.table)  
  
## Download and extract the data, if it hasn't been done already.  
if(!file.exists('UCI HAR Dataset')){  
 url <- paste0('https://d396qusza40orc.cloudfront.net/',  
 'getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip')  
 download.file(url, destfile = 'dataset.zip', method='curl')  
 unzip('dataset.zip')  
}  
  
##Get, load and combine trainig data  
training = read.csv("UCI HAR Dataset/train/X\_train.txt", sep="",colClasses = "numeric", header=FALSE)  
training[,562] = read.csv("UCI HAR Dataset/train/Y\_train.txt",colClasses = "numeric", sep="", header=FALSE)  
training[,563] = read.csv("UCI HAR Dataset/train/subject\_train.txt",colClasses = "numeric", sep="", header=FALSE)  
  
##Get, load and combine testing data  
testing = read.csv("UCI HAR Dataset/test/X\_test.txt", sep="",colClasses = "numeric", header=FALSE)  
testing[,562] = read.csv("UCI HAR Dataset/test/Y\_test.txt", sep="",colClasses = "numeric", header=FALSE)  
testing[,563] = read.csv("UCI HAR Dataset/test/subject\_test.txt",colClasses = "numeric", sep="", header=FALSE)  
  
## Merge training and test sets together  
activityData = rbind(training, testing)  
  
##Get & load Activity Labels data  
activityLabels = read.csv("UCI HAR Dataset/activity\_labels.txt", sep="", header=FALSE, stringsAsFactors=FALSE)  
# Make activity labels more readable  
activityLabels = mutate(activityLabels,V2=tolower(c(V2)))  
activityLabels$V2 <- gsub("\_", " ", activityLabels$V2)  
  
## Get headers for activity data from features.txt  
feature\_header<-read.csv("UCI HAR Dataset/features.txt", sep="",header=FALSE, stringsAsFactors=FALSE)  
#Add columns for Activity and Subject  
feature\_header <- c(feature\_header$V2, "Activity", "Subject")  
  
## Add new column names to activity data  
colnames(activityData) <- c(feature\_header)  
feature\_header <- cbind(names(activityData))  
column.table <- data.frame(feature\_header)  
  
## Extract required columns  
# Extract mean columns  
column.table$extract <- grepl("-mean", column.table$feature\_header)|grepl("Mean", column.table$feature\_header)  
# Extract std columns  
column.table$extract <- column.table$extract | grepl("-std", column.table$feature\_header)  
# Extract Activity & Subject columns  
column.table$extract <- column.table$extract | grepl("Activity", column.table$feature\_header)|grepl("Subject", column.table$feature\_header)  
column.table$extract[1:2] <- TRUE  
# Use the above extracted columns to get only columns to needed from activity data  
activityData <- activityData[,column.table$extract]  
  
## Tidy column heading to be more readable  
names(activityData) <- gsub("mean", "Mean ", names(activityData))  
names(activityData) <- gsub("std", "Std", names(activityData))  
names(activityData) <- gsub("-", " ", names(activityData))  
names(activityData) <- gsub("[()]", "", names(activityData))  
names(activityData) <- gsub('^t', 'time.', names(activityData))  
names(activityData) <- gsub('^f', 'freq.', names(activityData))  
names(activityData) <- gsub('Acc', 'Acceleration', names(activityData))  
names(activityData) <- gsub('Mag', 'Magnitude', names(activityData))  
  
## Change Activity column data to more meaniful descriptors based on tidied activity labels  
currentActivity = 1  
for( currentActivityLabel in activityLabels$V2) {  
 activityData$Activity[activityData$Activity==currentActivity] = activityLabels$V2[activityLabels$V1==currentActivity]  
 currentActivity <- currentActivity + 1  
 }  
  
##Save finshed tidy data as file  
write.table(activityData, "tidy.txt", sep="\t")  
  
## Work out average of each variable for each activity and each subject  
tidymean = aggregate(activityData,by=list(Activity = activityData$Activity, Subject=activityData$Subject), FUN=mean, na.rm=TRUE)

## Warning in mean.default(X[[1L]], ...): argument is not numeric or logical:  
## returning NA

## Warning in mean.default(X[[2L]], ...): argument is not numeric or logical:  
## returning NA

## Warning in mean.default(X[[3L]], ...): argument is not numeric or logical:  
## returning NA

## Warning in mean.default(X[[4L]], ...): argument is not numeric or logical:  
## returning NA

## Warning in mean.default(X[[5L]], ...): argument is not numeric or logical:  
## returning NA

## Warning in mean.default(X[[6L]], ...): argument is not numeric or logical:  
## returning NA

## Warning in mean.default(X[[7L]], ...): argument is not numeric or logical:  
## returning NA

## Warning in mean.default(X[[8L]], ...): argument is not numeric or logical:  
## returning NA

## Warning in mean.default(X[[9L]], ...): argument is not numeric or logical:  
## returning NA

## Warning in mean.default(X[[10L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[11L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[12L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[13L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[14L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[15L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[16L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[17L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[18L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[19L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[20L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[21L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[22L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[23L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[24L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[25L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[26L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[27L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[28L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[29L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[30L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[31L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[32L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[33L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[34L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[35L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[36L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[37L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[38L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[39L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[40L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[41L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[42L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[43L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[44L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[45L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[46L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[47L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[48L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[49L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[50L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[51L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[52L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[53L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[54L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[55L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[56L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[57L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[58L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[59L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[60L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[61L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[62L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[63L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[64L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[65L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[66L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[67L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[68L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[69L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[70L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[71L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[72L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[73L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[74L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[75L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[76L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[77L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[78L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[79L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[80L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[81L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[82L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[83L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[84L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[85L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[86L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[87L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[88L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[89L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[90L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[91L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[92L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[93L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[94L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[95L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[96L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[97L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[98L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[99L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[100L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[101L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[102L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[103L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[104L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[105L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[106L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[107L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[108L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[109L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[110L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[111L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[112L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[113L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[114L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[115L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[116L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[117L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[118L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[119L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[120L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[121L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[122L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[123L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[124L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[125L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[126L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[127L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[128L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[129L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[130L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[131L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[132L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[133L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[134L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[135L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[136L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[137L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[138L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[139L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[140L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[141L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[142L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[143L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[144L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[145L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[146L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[147L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[148L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[149L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[150L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[151L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[152L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[153L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[154L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[155L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[156L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[157L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[158L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[159L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[160L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[161L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[162L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[163L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[164L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[165L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[166L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[167L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[168L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[169L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[170L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[171L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[172L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[173L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[174L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[175L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[176L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[177L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[178L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[179L]], ...): argument is not numeric or  
## logical: returning NA

## Warning in mean.default(X[[180L]], ...): argument is not numeric or  
## logical: returning NA

#Save averaged data as file  
write.table(tidymean, "tidy\_average.txt", sep="\t")