**Instruction List** for run\_analysis.R assignment

Project title: run\_analysis.R

Author: Jeffrey Gonlin

Date: 4.April.2016

Summary: This code has been written in as part of the Coursera Data Cleaning course, week 4 assignment. See also: “Codebook run\_analysis.R assignment” The code below is taken straight out of the R file “run\_analysis.R” and contains a considerable volume of ##commetary which should shed some light on my mental ramblings as I sketched out the code.

## Coursera Data Week4 Assignment

## WD should be set to "Desktop", where I've established a folder "Week4 Assignment",

## i.e., "C:/Users/Jeff/Desktop/Week4 Assignment"

setwd("C:/Users/Jeff/Desktop/Week4 Assignment")

library(dplyr)

library(tidyr)

library(R.utils)

library(data.table)

library(plyr)

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

download.file(fileUrl, destfile = "./Dataset.zip")

Sys.Date() #2016-04-02 as of test run

datasetunzip <- unzip("./Dataset.zip",list=TRUE)

str(datasetunzip)

View(datasetunzip)

## Zip.file has been downloaded and structure examined / viewed. At this point:

## **TO PRESERVE THE ORIGINAL DATA**: Go to the desktop / WD 'Assignment' folder and open "Dataset.zip"

## **I WOULD** **DO THIS MANUALLY:**

## **COPY the "UCI HAR** **Dataset"** folder but take no other action with it.

## "UCI HAR Dataset" contains all we need / more than we need.

## **GO UP ONE LEVEL AND PASTE IT. THIS IS THE FILE I WILL WORK WITH.**

## Both "Dataset.zip" and "UCI HAR Dataset" should be found in: "C:/Users/Jeff/Desktop/Week4 Assignment"

## The original "Dataset.zip" download will remain untouched as original source data downloaded 2016-04-02.

## Examining "UCI HAR Dataset":

## "ReadMe" and "features\_info" text files provide details of methodology, sources of data, etc.

## and are best opened in a text program.

## Not all files are of interest. Two folders pertaining to "Inertial Signals" can be ignored.

## They contain original 128 readings sourced from subject cell phones, from which the measurements that interst us were derived.

## See below for subject, test and train files of interest. The "features" and "features\_info" files will also find use.

subject\_test <- read.table("./UCI HAR Dataset/test/subject\_test.txt")

str(subject\_test)

distinct(subject\_test) ## 9 distinct test subjects by number, incl: 2,4,9,10,12,13,18,20,24.

subject\_train <- read.table("./UCI HAR Dataset/train/subject\_train.txt")

str(subject\_train)

distinct(subject\_train) # 21 distinct, different subjects; along with the 9 in "test", this gives us 30.

## I'll now load and look at the X\_test and X\_train files.

X\_test <- read.table("./UCI HAR Dataset/test/X\_test.txt", sep = "")

str(X\_test)

## dim + 2947 x 561 variables

X\_train <- read.table("./UCI HAR Dataset/train/X\_train.txt", sep = "")

str(X\_train)

## dim + 7352 x 561 variables

## Now for the y\_test and y\_train files

y\_test <- read.table("./UCI HAR Dataset/test/y\_test.txt", sep = "")

str(y\_test)

## dim = 2947 x 1

distinct(y\_test)

## 6 distinct values. Looks like the activity, coded

y\_train <- read.table("./UCI HAR Dataset/train/y\_train.txt", sep = "")

str(y\_train)

## dim = 7352 x 1

distinct(y\_train)

## 6 distinct values; activity by code

## Observations:

## 1) d.f's contain no indexes,

## 2) colnames are "V1, V2 ..." so need replaced with descriptive names,

## 3) Implicit assumption: the test and train data are in the same row order as subject files and 'y' activity files.

## Let's combine 'subject', 'X' and 'y' files and call them "subjects", "activities" and "feature\_vectors" (in keeping with ReadMe).

subjects <- bind\_rows(subject\_test, subject\_train)

activities <- bind\_rows(y\_test, y\_train)

feature\_vectors <- bind\_rows(X\_test, X\_train)

## set up file to decode progressively the activities and substitute descriptive text.

activity\_labels <- read.table("./UCI HAR Dataset/activity\_labels.txt")

print(activity\_labels)

## V1 V2

## 1 WALKING

## 2 WALKING\_UPSTAIRS

## 3 WALKING\_DOWNSTAIRS

## 4 SITTING

## 5 STANDING

## 6 LAYING

activities1 <- data.table(gsub(1, "walking", activities$V1))

activities2 <- data.table(gsub(2, "walkingupstairs", activities1$V1))

activities3 <- data.table(gsub(3, "wakingdownstairs", activities2$V1))

activities4 <- data.table(gsub(4, "sitting", activities3$V1))

activities5 <- data.table(gsub(5, "standing", activities4$V1))

activities6 <- data.table(gsub(6, "laying", activities5$V1))

rm(activities, activities1, activities2, activities3, activities4, activities5, activity\_labels) # Not needed

## Now add an index, give proper colnames and merge 'subjects' and 'activities6' files

subjects <- mutate(subjects, index = seq\_along(V1))

activities6 <- mutate(activities6, index = seq\_along(V1))

colnames(subjects) <- c("subject", "index")

colnames(activities6) <- c("activity", "index")

subjects\_activities <- merge(subjects, activities6)

## Look at the lables in "features".

## It contains various computed stats (means, std dev, abs dev, etc.) mentioned in "features\_info.txt".

features <- read.table("./UCI HAR Dataset/features.txt", sep = "")

str(features)

## dim = 561 obs x 2 variables.

## These entries correspond to the 561 colname variables (V1, V2, etc.) in 'feature\_vectors'.

## Although not "tidy" in the sense of being all lower case, etc., I find this format more intelligible.

## Use them as is to add the col names to 'feature\_vectors'.

featuresV2<- c(as.character(features$V2))

feature\_vectors <- setNames(feature\_vectors, featuresV2)

## Extract variables relating to: mean(): Mean value and std(): Standard deviation

## Can I get mean and std variables in 'feature\_vectors' by matching in the colnames? Yes.

feature\_vectors\_meanlabels <- data.frame(select\_vars(colnames(feature\_vectors), matches(".mean.")))

feature\_vectors\_stdlabels <- data.frame(select\_vars(colnames(feature\_vectors), matches(".std.")))

## From quick visual check looks like it selected appropriately, however recalling "feature\_info.txt",

## I don't believe the "Angle" vectors (e.g., "Angle(X, gravityMean")) should be included.

## Double check for "Angle" entries and select out.

feature\_vectors\_meanlabels\_unwanted <- data.frame(select\_vars(rownames(feature\_vectors\_meanlabels), matches("^Angle.")))

## NOTE: It is not clear to me whether other measurements, relating to the X/Y/Z dimensions, are required

## or simply measurements further derived from the X/Y/Z statistics, so I have extracted and included them.

## I would want to discuss this with the engineers and project designers to confirm. If not needed,

## they can more easily be eliminated than brought back in...

## 7 such measurements will be eliminated, afterwhich the 'unwanted' file will be removed.

feature\_vectors\_meanlabels <- slice(feature\_vectors\_meanlabels, -c(47:53))

rm(feature\_vectors\_meanlabels\_unwanted)

## Retain the rownames as they will be needed in column selection

rownames(feature\_vectors\_meanlabels) <- feature\_vectors\_meanlabels[ ,1]

## Enforce a common colname on both 'label' files so they can be combined.

colnames(feature\_vectors\_meanlabels) <- c("mean\_std\_labels")

colnames(feature\_vectors\_stdlabels) <- c("mean\_std\_labels")

## Combine and use as a vector to select columns in 'feature\_vectors'; then remove uneeded 'meanlabels' and 'stdlabels' files.

feature\_vectors\_mean\_std\_labels <- rbind(feature\_vectors\_meanlabels, feature\_vectors\_stdlabels)

rm(feature\_vectors\_meanlabels, feature\_vectors\_stdlabels)

feature\_vectorsDT <- as.data.table(feature\_vectors)

## Now to select columns from "feature\_vectors" using the meanlabels and stdlabels

feature\_vectors\_mean\_std\_cols <- feature\_vectorsDT[,c(rownames(feature\_vectors\_mean\_std\_labels)), with=FALSE]

## Now combine 'subjects\_activities' with 'feature\_vectors\_mean\_std\_cols' to get one single complete dataset.

feature\_vectors\_mean\_std\_cols <- mutate(feature\_vectors\_mean\_std\_cols, index=seq\_len(10299))

one\_data\_set <- merge(subjects\_activities, feature\_vectors\_mean\_std\_cols)

one\_data\_set <- as.data.table(one\_data\_set)

View(one\_data\_set)

## 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.

tidy\_data\_set <- one\_data\_set

## Fist get rid of the index column

tidy\_data\_set <- tidy\_data\_set[,index:= NULL]

## Now group and summarize the data using ddply:

tidy\_data\_grouped\_averaged <- ddply(tidy\_data\_set, .(subject, activity), colwise(mean))

## Let's also clean up a few files!!

rm(activities6, subjects, subjects\_activities, subject\_train, subject\_test, X\_test, X\_train, y\_test, y\_train, features)

View(tidy\_data\_grouped\_averaged)

write.table(tidy\_data\_grouped\_averaged, file = "C:/Users/Jeff/Desktop/Week4 Assignment/tidy\_data\_grouped\_averaged\_saved", sep="\t", row.name=FALSE)