==========
Coursera: Getting and Cleaning Data - Week 4 Assignment
==========

Raw data for the assignment were obtained here:
https://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUCI%20HAR%20Dataset.zip

A full description of how the raw data were obtained is here:
http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

The "Thoughtful Bloke" blog was referenced in completion of this assignment:
https://thoughtfulbloke.wordpress.com/2015/09/09/getting-and-cleaning-the-assignment/

==========
The tasks required for the assignment were:
"create one 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."

(see https://www.coursera.org/learn/data-cleaning/peer/FIZtT/getting-and-cleaning-data-course-project)

==========
The document "run_analysis.R" contains the R source code to complete the assigned tasks. It was created in Studio Version 0.99.902 on a Mac running OS X El Capitan. To run the script in R, use the code:

source(“path/run_analysis.R”) where path is the path to the directory where the file is stored

==========

Why are the data produced by the R script “tidy”? In short, because they meet the criteria described in week 1 of the Coursera “Getting and Cleaning Data” course. That is, there is one variable per column. There is one observation per row. NOTE: the data are in “long form” with multiple rows per subject. However, because each row represents a single observation of each subject, this configuration still meets the criterion of one observation per row. The dataset includes an ID variable for matching with other tables. Finally, the dataset includes 1 “kind” of variable (i.e., variables relating to the accelerometer tests used to obtain the raw data).

==========

The dataset includes the following files:

1. Week-4-assignment-read-me.rtf
2. Codebook.txt
3. tidyData.txt

==========

The R script is annotated to provide details about what the steps that were taken and what the code does. The script includes code to load the necessary packages. NOTE: the plyr package is loaded only at the end of the script when it is needed, because it masks functions from the dplyr package that are used in the early steps of the script.

Task 1 - Merge the training and the test sets to create one data set

The first task is divided into four steps: 1) read in the datasets and View them, 2) add labels to datasets, 3) combine test data and training data, respectively, and 4) merge test and training data. NOTE: step 2 of this process also achieves the requirements of Task 4 for the assignment. The command

colnames(trainData) <- features[,2]
is the command that assigns “descriptive variable names” as required for Task 4 of the assignment.

Task 2 - Extract only the measurements on the mean and standard deviation for each measurement
This code uses the grepl() function and subsetting to create a new dataset containing only variables with names containing the string sequences “mean” or “std”, as well as the variables “ID” and “testtype”.

Task 3 - Use descriptive activity names to name the activities in the data set
Descriptive activity names are provided by merging the “activityLabels” and “meanStdData” data frames, matching by the numeric “testtype” variable found in both datasets.

Task 4 - Appropriately label the data set with descriptive variable names
This was accomplished in Step 2 under Task 1. Please see above for details.

Task 5 - From the data set in step 4, create a second, independent tidy data set with the average of each variable for each activity and each subject
NOTE: This task uses the ddply() function from the plyr package. If you do not have the plyr package, you will need to install it. You can do so by removing the “#” from the command line:
#install.packages(“plyr”)
in the run_analysis.R script.
Code for this task takes the dataset tidyData and uses the ddply() function to group the data first by testtype, then by ID (subject), and then provide means for each variable by subject (ID) for each testtype. The resulting dataset is saved as a text file using the command:
write.table(tidyData, "tidy_data.txt", row.name=FALSE)

To read the data into R, you can use the code read.table(“path/tidy_data.txt”, header = TRUE) where “path” is the path to the directory where the file is stored