Purpose
The objective of this project is to prepare a tidy data set for subsequent analysis. The information below provides background on the HAR Dataset and an overview of the processing performed to prepare the tidy data.

Contents Overview
This following files are included in the repo at https://github.com/nxd1/UCI-HAR-data-analysis:

1. *README.md – overview of repo*
2. *README.pdf – this file*
3. *UCI_HAR_tidy_data. txt* – file containing tidy data formatted as per the code book. Though not required in the repo, this file may be of interest as a way to verify the script's output.
4. *codebook.pdf – description of the variables and values in the tidy data set resulting from this analysis*
5. *run_analysis.R – R script for processing the raw data and creating the tidy data set. An overview of this script is given below.*

Analysis Steps
To recreate the tidy data for this project:

1. Download and extract the HAR Dataset zip archive (see codebook) into a subfolder named "UCI HAR Dataset" in the working directory.
2. Copy "run_analysis.R" to the working directory and source the file. See below for an description of what this script does.
3. Invoke cleanHARdata() in run_analysis.R, which creates/overwrites the file "*UCI_HAR_tidy_data. txt"* in the working directory.

The above was run 4 times, confirming intermediate and final results visually and programmatically in Microsoft Excel, and verifying consistent results each time.

This analysis was performed on a workstation running Windows 7 Enterprise SP1 (64 bit), Intel Core i5 - 4200U CPU @1.6 GHz with 16GB RAM.

R.Version() information:

$platform	[1] "x86_64-w64-mingw32"
$version.string	[1] "R version 3.1.1 (2014-07-10)"
$nickname	[1] "Sock it to Me"

Overview run_analysis.R
This script converts the raw data provided for this project, the "HAR Dataset", into tidy data[1] for further analysis. The entry point for this conversion is the function cleanHARdata(), which performs the following steps:

1. Merges the training and the test sets (of measurements)
2. Extracts only the measurements on the mean and standard deviation for each measurement
3. Appropriately labels the data set with descriptive variable names

[1] See http://vita.had.co.nz/papers/tidy-data.pdf for more information on data tidying.

4. Merges subject and activity data columns with the measurement data
5. Uses descriptive activity names to name the activities in the data set
6. Creates an independent tidy data set with the average of each variable for each activity and each subject.

Intermediate processing steps:

Merged measurements		Merged subjects		Merged activities		Data set labels	subject	activity
training measurements (7,352 x 561)	+	training subjects (7,352 x 1)	+	training activities (7,352 x 1)	===>	Extracted measurements (10,299 x 66)	subject ID's	activity ID's
test measurements (2,947 x 561)		test subjects (2,947 x 1)		test activities (2,947 x 1)				

Final data set layout:

subject	activity	measurement	mean
subject ID's	activity descriptions	measurement labels	values

NOTE: Final dataset length is 11,880 rows (30 subjects*6 activities*66 measurements), plus the header.

The following summarizes the processing steps. More detailed comments may be found in run_analysis.R.

Step 1 - Merges the training and the test sets (of measurements) to create one data set

The training and test sets consist of 7,352 and 2,947 records, respectively, of the processed smartphone data for each of six activities. Each row in these files consists of 561 time and frequency domain measurements from the processed accelerometer and gyroscope time series. This is explained further in the documentation accompanying the HAR Dataset (refer to feature_info.txt in the archive).

The script reads the set of measurements from the training and test sets into data frames, and appends the test to the training data.

Step 2 - Extracts only the measurements on the mean and standard deviation for each measurement

As per the HAR Dataset documentation (feature_info.txt), the variables selected for extraction in this step are only those containing "mean()" or "std()" in the variable name. These variables reflect 2 of the 17 different computations estimated in the data set, as per the HAR Dataset documentation.

Based on the data set description and column names, there are 66 variables to which mean or standard deviation computations are applied:

Triaxial Variables	Single Variables
1. tBodyAcc-XYZ	1. tBodyAccMag
2. tGravityAcc-XYZ	2. tGravityAccMag
3. tBodyAccJerk-XYZ	3. tBodyAccJerkMag
4. tBodyGyro-XYZ	4. tBodyGyroMag
5. tBodyGyroJerk-XYZ	5. tBodyGyroJerkMag
6. fBodyAcc-XYZ	6. fBodyAccMag
7. fBodyAccJerk-XYZ	7. fBodyAccJerkMag
8. fBodyGyro-XYZ	8. fBodyGyroMag
	9. fBodyGyroJerkMag
24 total variables (8*3)	
Total selected for the two measurements: 66 ((24 + 9) * 2)	

From the above 33 variables (24 triaxial/9 single) the 17 measures indicated in the HAR Dataset documentation produce the 561 features.

NOTE: This script specifically excludes processing a number of variables that include 'mean' in the name. The excluded measurements reflect either weighted average calculations (e.g., meanFreq()) or calculations of the angle between vectors (e.g., angle(tBodyAccMean, gravity)). Therefore, these measurements were not considered relevant for this analysis.

The script reads the variable names from a file, converts these to lowercase, and creates a feature vector based on those variables that contain "mean()" or "std()" in the name. The feature vector consists of the column index for those variables, which is then used to subset the merged data set from Step 1.

Step 3 - Appropriately labels the data set with descriptive variable names

The script transforms the original variable names for the selected measurements into appropriate labels using pattern matching. Given the length of the variable names in this analysis, a slightly different convention from the lecture is used in naming variables to improve legibility and traceability. Periods, for example, replace dashes and whitespace in the original name.

It was also important for this analysis, since a number of original variables are dropped, to trace easily to the original columns that remain. Thus, variable names are prefixed with a unique ID (vNNN), where NNN is the original column number from the raw data set.

Example:

Original name (lowercase)	Label after conversion
tbodyacc-mean()-x	v001.tbodyacc.mean.x
tbodyacc-mean()-y	v002.tbodyacc.mean.y
tbodyacc-mean()-z	v003.tbodyacc.mean.z

Step 4 - Merges subject and activity data columns with the measurement data

The HAR Dataset includes separate files for training and test data that identify the subject and activity for the corresponding measurements. The subject and activity files are as follows:

File name	Rows/Records	Content Description
subject_train.txt	7,352	Identifiers for 21 different subjects in the training data
subject_test.txt	2,947	Identifiers for 9 different subjects in the test data
y_train.txt	7,352	Identifiers for the 6 activities in the training data
y_test.txt	2,947	Identifiers for the 6 activities in the test data

The script reads the subject identifiers from the training and test files, and appends the test to the training subject data. The combined subject data is then merged, column-wise, with the measurement data in Step 1 following the same order.

The script also reads the activity identifiers from the training and test files, and appends the test to the activity training data. Similarly, the combined activity data is then merged, column-wise, with the measurement data in Step 1 and the subject data.

Step 5 - Uses descriptive activity names to name the activities in the data set

The script reads the file "features.txt" containing the descriptive names for the activities and their identifiers. A look-up is made on activity identifier to update the data table with the corresponding descriptive name (e.g., 1 → "WALKING")

Step 6 - Creates an independent tidy data set with the average of each variable for each activity and each subject

The script creates a "long" data table containing subject, activity name, measurement, and the calculated average for each mean and standard deviation measurement, and for each subject/activity combination. Another option for the final tidy data was the "wide" format. This was discarded in favor of closer adherence to the interpretation that tidy data consists of one column for each variable (Third Normal Form), which in this case is simply the average of a number of measurements. If necessary for further analysis, the averages for "mean" and "std" could be subset, based on the name of the variable in the measurement column.

Each of the 66 corresponding variable names from Step 2 is folded (melted) into a column for each subject and activity. This yields a total of 11,880 rows (30 subjects * 6 activities * 66 measurements) for the tidy data set. The average is then calculated in the data table by subject, activity, and measurement.

As a final step, the script outputs a space delimited file named "UCI_HAR_tidy_data.txt". A description of the variables and values in the tidy data set can be found in the codebook. This file can be read into R using the command:

```
tidy <- read.table(file = "UCI_HAR_tidy_data.txt", header = TRUE)
```

The resulting output is:

```
> head(tidy,12)
```

	subject	activity	measure	mean
1	1	WALKING	v001.tbodyacc.mean.x	0.2773308
2	1	WALKING_UPSTAIRS	v001.tbodyacc.mean.x	0.2554617
3	1	WALKING_DOWNSTAIRS	v001.tbodyacc.mean.x	0.2891883
4	1	SITTING	v001.tbodyacc.mean.x	0.2612376
5	1	STANDING	v001.tbodyacc.mean.x	0.2789176
6	1	LAYING	v001.tbodyacc.mean.x	0.2215982
7	2	WALKING	v001.tbodyacc.mean.x	0.2764266
8	2	WALKING_UPSTAIRS	v001.tbodyacc.mean.x	0.2471648
9	2	WALKING_DOWNSTAIRS	v001.tbodyacc.mean.x	0.2776153
10	2	SITTING	v001.tbodyacc.mean.x	0.2770874
11	2	STANDING	v001.tbodyacc.mean.x	0.2779115
12	2	LAYING	v001.tbodyacc.mean.x	0.2813734

```
> tail(tidy, 12)
```

	subject	activity	measure	mean
11869	29	WALKING	v543.fbodybodygyrojerkmag.std	-0.6186677
11870	29	WALKING_UPSTAIRS	v543.fbodybodygyrojerkmag.std	-0.7564642
11871	29	WALKING_DOWNSTAIRS	v543.fbodybodygyrojerkmag.std	-0.6266760
11872	29	SITTING	v543.fbodybodygyrojerkmag.std	-0.9947420
11873	29	STANDING	v543.fbodybodygyrojerkmag.std	-0.9915168
11874	29	LAYING	v543.fbodybodygyrojerkmag.std	-0.9975852
11875	30	WALKING	v543.fbodybodygyrojerkmag.std	-0.5785800
11876	30	WALKING_UPSTAIRS	v543.fbodybodygyrojerkmag.std	-0.7913494
11877	30	WALKING_DOWNSTAIRS	v543.fbodybodygyrojerkmag.std	-0.6455039
11878	30	SITTING	v543.fbodybodygyrojerkmag.std	-0.9909464
11879	30	STANDING	v543.fbodybodygyrojerkmag.std	-0.9550086
11880	30	LAYING	v543.fbodybodygyrojerkmag.std	-0.9754815