Getting and Cleaning Data Course

Presented by Johns Hopkins University through Coursera
November 2014

Code Book for Course Project

Written by GitHub user sdbj2063.

Table of Contents

I. Introduction

The course “Getting and Cleaning Data” offered by Johns Hopkins University through Coursera has a culminating project. The student must complete an R script that performs data transformation and creates a text file. Part of the project offering is this code book, along with the script, a readme file and the data file.

II. Documentation Conventions

References to section headings within this document are in double quotes.

File names, directories and **URLs** use a **bold Verdana** sans serif font.

`Screen text` uses the `Lucida Console` monospaced serif font.

III. Assumptions about the Audience

This project code book assumes that the reader has a basic working knowledge of R; using packages; sourcing script files and executing scripts; importing data; organizing data in data frames; doing basic mathematical calculations, such as mean or standard deviation; and writing out data to a text file. Understanding these concepts is helpful when reading this code book, the R script, custom readme and the tidy data set.

This code book assumes the reader only has access to C**odeBook.md**, **run_analysis.R** and **README.md**. The project includes the tidy data set generated by the script, **gcd_tidy_data_set.txt**. The code book also assumes the user has access to the Internet with a connection speed sufficient to download a 58 MB file. The script downloads all pertinent original data and descriptive files from the internet and extracts them before beginning the data transformation process.

IV. Project Requirements

1. List of Deliverables:

1. GitHub repository for this Johns Hopkins project
2. Link to the Github repo, **https://github.com/sdbj2063/gcd**
3. **gcd_tidy_data_set.txt** tidy data set in the GitHub repo
4. **CodeBook.md** code book in the GitHub repo
5. **README.md** in the GitHub repo

6. **run_analysis.R** in the GitHub repo. The file should have limited comments and perform the required data analysis requirements.

2. Data Analysis Requirements:

An R script titled **run_analysis.R** must perform the following actions:

1. Merge the test and training data sets into one data set.
2. Extract the measurements on the mean and standard deviation for each measurement.
3. Use descriptive activity names for the activities.
4. Use human-readable descriptive names for the variables.
5. From the data set created following steps 1-4, create a second, independent tidy data set with the average of each variable for each activity and each subject.

I interpreted #5 to be the mean for each variable containing "mean()" and "std()" in the name, grouped by subject and activity. For more information about the decision process in choosing the variables, please see sections VI.2 and VI.3, "Processing Decisions," "Choosing Variable for Mean and Standard Deviation Calculations" and "Excluded Variables with 'mean' in the Name."

"Appendix A" contains the Johns Hopkins project description.

V. Original Data Sources

1. URLs

This project utilizes the code book and data at the following URLs:

1) The original project description:

   http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

2) The original data for that project:

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

"Appendix B" contains the original project and data description.

2. Licensing Statement from Original Development Team

Use of this dataset in publications must be acknowledged by referencing the following publication [1]

[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.

3. Description of Data from the Original Project

The original researchers derived the data from recording 30 subjects performing a variety of activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) while wearing a Samsung Galaxy II smartphone. The original experiment designers randomly divided the subjects into two groups, 70% for training data and 30% for test data. The volunteers had an age range of 19 to 48. Using the phones' embedded accelerometer and gyroscope, the research team recorded linear acceleration and angular velocity on the X, Y and Z axis at a rate of 50Hz. Included in the acceleration data are gravitational and body motion components. By utilizing the time and frequency domains in the 128 readings for each activity window, the researchers calculated 561 variables of data, including mean and standard deviation.

“Appendix B” contains a more detailed explanation of the data and process.

The data pertaining to this Johns Hopkins project is a collection of 561 attributes of time series data. Each row of data pertains to one subject and one of six activities. There are multiple collections of data points for each subject and activity.

4. Description of Data Sets

The zipped file includes several data sets that the system extracts into subdirectories in the working directory:

/UCI HAR Dataset

/UCI HAR Dataset/test

/UCI HAR Dataset/train

The data files used for this project consist of nine test subjects, **X_test.txt** and **y_test.txt**, and 21 evaluated subjects, **X_train.txt** and **y_train.txt**. The "X" data files contain the calculated data for the variables collected. The "y" data files contain a column with an activity code for each row of data in the corresponding "X" data file. The **subject_test.txt** and **subject_train.txt** data files contain a column with the subject id for each row of data in the corresponding "X" data files.

The "X" data sets have separate column names contained in **features.txt**.

Activity codes in the "X" data sets match one of the six character label names contained in **activity_labels.txt**.

The fully expanded zip file also provides raw data files in subdirectories:

/UCI HAR Dataset/test/InertialSignals

/UCI HAR Dataset/train/InertialSignals

They are not necessary for the purpose of this Johns Hopkins project.

5. How Data Knits Together

The original source data knits together in a Lego block pattern, with rectangles of different sizes forming to create one large rectangle. The diagram in "Appendix C" illustrates how the various data files knit together to form a cohesive data set. It also shows the primary phases in processing the data to reach the tidy data set.

6. Data Files Used with Definitions

The project requires the following data files included in the zipped file downloaded during the script's running process:

Filename	Description
features.txt	CSV file listing all 561 variable names to be used for column names.
activity_labels.txt	CSV file listing the six activity codes with their activity names.
train/X_train.txt	CSV file with 561 variables of data for the training subjects.
train/y_train.txt	CSV file with the activity codes for the **X_train.txt** data.
train/subject_train.txt	CSV file with the subject ids for the training subjects.
test/X_test.txt	CSV file with 561 variables of data for the test subjects.
test/y_test.txt	CSV file with the activity codes for the **X_test.txt** data.
test/subject_test.txt	CSV file with the subject ids for the test subjects.

VI. Processing Decisions

Completing the project data transformations required making several key decisions, including what commands to use, which variables to select, how to name the variables and researching the background of gyroscopes and accelerometers within the context of mobile phones.

1. R Language Libraries

I used two libraries, "utils" and "reshape2," the latter of which the lesson videos covered. To address the mean calculations I used "melt" and "dcast" from the "reshape2" library.

I removed variables as they became obsolete, which improved performance.

2. Choosing Variables for Mean and Standard Deviation Calculations

The Johns Hopkins project required students to calculate the mean() for variables pertaining to the baseline calculations for mean and standard deviation on the raw data sets.

The original data sets had 561 variables with X, Y, and Z measurements for location in space. To meet the project requirements for mean and standard deviation, I selected 33 variables with "mean()" and 33 variables with "std()" in the names.

3. Excluded Variables with “mean” in the Name

A key assumption I made in choosing what variables to use is that the Johns Hopkins professor would want symmetrical results for students to review one another. In other words, it made sense that when reviewing variable names, I identified an equal number of variables with “std()” and “mean()”

One noticeable aspect of the list of 561 variables is that the original researchers made many baseline calculations on the raw data. In addition to mean and standard deviation, they calculated the median absolute deviation (mad()), the largest and smallest values in the array (max() and min()), the signal magnitude area (sma()), and interquartile range (iqr()), among others, for the major variable groups.

Then they calculated the meanFreq(). According to the original data project file **features_info.txt**, "meanFreq(): Weighted average of the frequency components to obtain a mean frequency." The researchers did not explain why they calculated the mean on frequency but not complete the other baseline calculations on frequency, such as standard deviation. This lead me to believe that the meanFreq() calculations were secondary calculations. Furthermore, including these would unbalance the tidy data set as a whole.

The excluded “mean” variables for this Johns Hopkins project are below:

fBodyAcc-meanFreq()-X

fBodyAcc-meanFreq()-Y

fBodyAcc-meanFreq()-Z

fBodyAccJerk-meanFreq()-X

fBodyAccJerk-meanFreq()-Y

fBodyAccJerk-meanFreq()-Z

fBodyGyro-meanFreq()-X

fBodyGyro-meanFreq()-Y

fBodyGyro-meanFreq()-Z

FbodyBodyAccJerkMag-meanFreq()

fBodyBodyGyroMag-meanFreq()

fBodyBodyGyroJerkMag-meanFreq()

Seven angle variables contained "mean." In this context, according to **features_info.txt**, "angle(): Angle between to(sp) vectors." This is not a baseline calculation but a relationship between two data points, one of which is a mean. As such, this is a secondary calculation and does not meet the Johns Hopkins project criteria. The rejected variables follow:

angle(tBodyAccMean,gravity)

angle(tBodyAccJerkMean),gravityMean

angle(tBodyGyroMean,gravityMean)

angle(tBodyGyroJerkMean,gravityMean)

angle(X,gravityMean)

angle(Y,gravityMean)

angle(Z,gravityMean)

4. Background on Gyroscopes, Accelerometers and Original Data

The gyroscope determines orientation in 3D space, while the accelerometer measures non-gravitational acceleration. The gyroscope maintains a position on a 3-axial plane, using gravity to determine which way is down. The accelerometer captures vibrations and generates current signals when the device moves; based upon those signals it calculates acceleration. Ref 1, 2

The original researchers used "low pass Butterworth filter with a corner frequency of 0.3 Hz" to separate the acceleration signal into body and gravity figures. The original development team obtained the frequency domain signals by applying Fast Fourier Transform (FFT) to variables. Ref 3

Using time and the “body linear acceleration and angular velocity,” researchers calculated Jerk signals. Finally, using the Euclidean norm, researchers calculated the magnitude of the XYZ values. Ref 3

The _X, _Y and _Z values in the data indicate the axial measurements. Ref 3. X and Y are planar coordinates, and Z is the altitude, or depth.

5. Human-readable Variable Names

What constitutes a human-readable descriptive name is a question of personal experience. In database development, administrators often use standard abbreviations to represent key database concepts, such as “tbl” for table or “qry” for query. They also use underscores to separate words that are parts of names to make them more readable.

In the R language development environment, most commands or functions have multiple words to make up one function name. Each word begins with a capital letter to ease readability.

I chose to expand key abbreviations to a complete word, capitalize the first letter of each word and substitute underscore for special characters. Some of the variable names are long, but they are more understandable. Word substitutions included the following:

- "t" at the beginning of the variable name became "Time."
- "f" at the beginning of the variable name became "Freq" for frequency.
- "Acc" became "Acceleration."
- "Gyro" became "Gyroscope."
- "Mag" became "Magnitude."

I found two common patterns in the system, "-X()-" and "-X()" where X was either "mean" or "std." Substituting "_" for the punctuation in those four parameters became the final step in the conversion process.

6. Variable Key Words and Their Meanings

I contacted the original developers and asked why they used the phrase “BodyBody” in some variable names. I did not receive a reply.

Key Word Substitute	**Meaning**	**Original Variable Tag**
Time	Time series data.	t
Freq	Frequency domain signals calculated.	f
Body	Body motion component from sensor acceleration signal	Body
BodyBody	Body motion component. The original developers did not provide an explanation for this keyword combination.	BodyBody
Acceleration	Linear acceleration.	Acc
Jerk	Rate of change of acceleration.	Jerk

Key Word Substitute	**Meaning**	**Original Variable Tag**
Gyroscope	Angular velocity.	Gyro
Gravity	Gravitational motion component from sensor acceleration signal.	Gravity
Magnitude	How far the quantity differs from zero.	Mag

References 3, 4, 5

7. List of mean() variables used with explanations

The following variables contain the mean value of the mean() variables grouped by subject and activity.

The table below lists the time measurements.

TimeBodyAcceleration_mean_X, _Y, _Z	## Body acceleration signal.
TimeGravityAcceleration_mean_X, _Y, _Z	## Gravity acceleration signal
TimeBodyAccelerationJerk_mean_X, _Y, _Z	## Body acceleration jerk measurement
TimeBodyGyroscope_mean_X, _Y, _Z	## Body gyroscope baseline measurement
TimeBodyGyroscopeJerk_mean_X, _Y, _Z	## Body angular jerk velocity
TimeBodyAccelerationMagnitude_mean	## Magnitude of its measurement
TimeGravityAccelerationMagnitude_mean	## Magnitude of its measurement
TimeBodyAccelerationJerkMagnitude_mean	## Magnitude of its measurement
TimeBodyGyroscopeMagnitude_mean	## Magnitude of its measurement
TimeBodyGyroscopeJerkMagnitude_mean	## Magnitude of its measurement

The following variables are frequency counterpart measurements to the time measurements above. The original development team obtained the frequency domain signals by applying Fast Fourier Transform (FFT) to variables.

FreqBodyAcceleration_mean_X, _Y, _Z

FreqBodyAccelerationJerk_mean_X, _Y, _Z

FreqBodyGyroscope_mean_X, _Y, _Z

FreqBodyAccelerationMagnitude_mean

FreqBodyBodyAccelerationJerkMagnitude_mean

FreqBodyBodyGyroscopeMagnitude_mean

FreqBodyBodyGyroscopeJerkMagnitude_mean

8. List of std() variables used with explanations

The following variables contain the mean value of the standard deviation--std()--variables grouped by subject and activity.

The table below lists the time measurements.

TimeBodyAcceleration_std_X, _Y, _Z	## Body acceleration signal.
TimeGravityAcceleration_std_X, _Y, _Z	## Gravity acceleration signal
TimeBodyAccelerationJerk_std_X, _Y, _Z	## Body acceleration jerk measurement
TimeBodyGyroscope_std_X, _Y, _Z	## Body gyroscope baseline measurement
TimeBodyGyroscopeJerk_std_X, _Y, _Z	## Body angular jerk velocity
TimeBodyAccelerationMagnitude_std	## Magnitude of its measurement
TimeGravityAccelerationMagnitude_std	## Magnitude of its measurement
TimeBodyAccelerationJerkMagnitude_std	## Magnitude of its measurement
TimeBodyGyroscopeMagnitude_std	## Magnitude of its measurement
TimeBodyGyroscopeJerkMagnitude_std	## Magnitude of its measurement

The following variables are frequency counterpart measurements to the time measurements above. The original development team obtained the frequency domain signals by applying Fast Fourier Transform (FFT) to variables.

FreqBodyAcceleration_std_X, _Y, _Z

FreqBodyAccelerationJerk_std_X, _Y, _Z

FreqBodyGyroscope_std_X, _Y, _Z

FreqBodyAccelerationMagnitude_std

FreqBodyBodyAccelerationJerkMagnitude_std

FreqBodyBodyGyroscopeMagnitude_std

FreqBodyBodyGyroscopeJerkMagnitude_std

VII. Running the Script and Key Decision Points

The following subsections describe the script development environment and provide directions about using the script and key components of the script’s processing. **README.md**, included as part of this Johns Hopkins project, contains a step-by-step description of the script's processing.

1. System Configuration

The script and results were compiled and run on the following system using RStudio.

- Windows 7 home Premium, Service Pack 1
- 64-bit OS
- 1.6 GHz processor

- 4.00 GB RAM

- R version 3.1.1 (2014-07-10) -- "Sock it to Me"
- Copyright (C) 2014 The R Foundation for Statistical Computing
- Platform: x86_64-w64-mingw32/x64 (64-bit)

- R package "utils" 3.1.1
- R package "reshape2" 1.4

2. Executing the Script

When the script executed on the above system configuration, downloading the files took about 2 1/2 minutes depending upon the connection speed. Total run time for the script was less than five minutes.

For the script to run properly, save it to your working directory. After sourcing the script from the command line prompt, type the function name at the command prompt to execute the script:

```
> source(“run_analysis.R”)

> run_analysis()
```

3. Messages Typed to the Screen

When the script executes, the R system generates three messages to the screen automatically.

```
trying URL
'http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognitio
n+Using+Smartphones'

Content type 'text/html; charset=UTF-8' length 200 bytes

opened URL

downloaded 9017 bytes


trying URL
'http://d396qusza40orc.cloudfront.net/getdata%2Fprojectfiles%2FUC
I%20HAR%20Dataset.zip'

Content type 'application/zip' length 62556944 bytes (59.7 Mb)
```

```
opened URL
downloaded 59.7 Mb


Warning message:
In download.file(myurlbook, "human-activity-recognition-using-
smartphones.html") :
  downloaded length 9017 != reported length 200
```

None of these messages has any impact on the script’s execution.

4. Downloading and Extracting Files

The script does not check for pre-existing directory paths or files. Before rerunning the script, the best course of action is to first archive or delete the generated tidy data set, the original set of HTML and Zip files and any associated subdirectories and files.

The script downloads the two files mentioned in Section V.1, “Original Data Sources: URLs,” to the working directory. It creates a directory path and extracts the text files to that path.

5. Transformation of Data

Once the script extracts the data from the zip file, the program imports the necessary data files. Please see Section V.6, “Original Data Sources: Data Files Used with Definitions.” Then the script completes the following transformations:

1) Merges the "subject" data sets, first "test" then "train."

2) Merges the "X" data sets, first "test" then "train."

3) Merges the "Y" data sets, first "test" then "train."

4) Creates human readable variables names.

5) Applies new column names to the big data set.

6) Extracts the variable columns concerning mean() and std().

7) Merges the "Y" super set with the "X" super set.

8) Merges the "subject" super set with the "X-Y" super set.

9) Melts the data into a manageable data frame grouped by subject and activity.

10) Calculates the mean for every variable grouped by subject and activity.

11) Replaces activity codes with activity labels.

11) Exports the tidy data set to text file **gcd_tidy_data_set.txt**.

README.md contains a detailed description of the script processing steps and results.

6. Tidy Data Set Results

The script creates a space-delimited file **gcd_tidy_data_set.txt** in the working directory.

A correct file contains the following rows of data:

- 181x rows, with the first containing the variable, or column, names;

A correct file contains the following columns, left to right, of data:

- 1x column for subject ids ("SubjectCode") containing an integer 1-30 representing the unique of id of the subject;
- 1x column for activity names ("ActivityCode") containing a character string with the name of the activity;
- 33x columns for variables representing the mean--mean()--of measurements; and
- 33x columns for variables representing the standard deviation--std()--of measurements.

For each combination of subject and activity, the tidy data set contains one row of data with the mean of the data points for that subject/activity combination. Thus, each subject has six rows of data, one for each activity.

7. The Tidiness of the Data Set

The swirl() exercise for "Getting and Cleaning Data" on tidy data using tidyr() identified three characteristics of tidy data.

1) Each variable is in a column. This data set meets that requirement by listing each variable in its own column.

2) Each observation is in a row. This data set meets that requirement by listing the mean calculations for each mean() and std() observed data point for each subject/activity combination.

3) Each type of observational unit forms a table. This data set meets that requirement by listing all the means calculated.

The horizontal layout of this report provided a manageable means of visually inspecting the data.

To take this process one step further, I could have created a four-column table with subject, activity, variable name and value. Instead of the numerical variables each forming their own column, each variable name would be a value in the variable column and its subject/activity/variable combination would have one value. In effect, the data would be a three-way intersection (many-to-many) table in a relational diagram.

However, I saw no practical reason for creating one really long, really skinny data set that a user would then have to recombine in order to make sense of the data. For the purposes of enabling student peer evaluators to quickly review the data, the current layout seemed the best approach and closely resembled the original data.

VIII. References

1 Ryan Goodrich, "Accelerometer vs. Gyroscope: What's the Difference?" *Purch*, http://www.livescience.com/40103-accelerometer-vs-gyroscope.html (accessed November 07, 2014).

2 Manisha Kumar, "Difference Between Gyroscope and Accelerometer," *Differences Between*, http://www.differencebetween.net/technology/difference-between-gyroscope-and-accelerometer/ (accessed November 07, 2014).

3 Davide Anguita et al, "features_info.txt" in original project documentation.

4 Richard G. Lyons, "Understanding Digital Signal Processing: Discrete Sequences and Systems," *Informit*, http://www.informit.com/articles/article.aspx?p=1650107&seqNum=2 (accessed November 07, 2014).

5 Vincent T. van Hees et al, "Separating Movement and Gravity Components in an Acceleration Signal and Implications for the Assessment of Human Daily Physical Activity," *PLos One*, National Institutes of Health, http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3634007/ (accessed November 07, 2014).

Appendix A

Coursera Project Description

The description below is quoted from "• Peer Assessments • /Getting and Cleaning Data Course Project," https://class.coursera.org/getdata-009/human_grading/ (accessed November 11, 2014).

The purpose of this project is to demonstrate your ability to collect, work with, and clean a data set. The goal is to prepare tidy data that can be used for later analysis. You will be graded by your peers on a series of yes/no questions related to the project. You will be required to submit: 1) a tidy data set as described below, 2) a link to a Github repository with your script for performing the analysis, and 3) a code book that describes the variables, the data, and any transformations or work that you performed to clean up the data called CodeBook.md. You should also include a README.md in the repo with your scripts. This repo explains how all of the scripts work and how they are connected.

One of the most exciting areas in all of data science right now is wearable computing - see for example this article . Companies like Fitbit, Nike, and Jawbone Up are racing to develop the most advanced algorithms to attract new users. The data linked to from the course website represent data collected from the accelerometers from the Samsung Galaxy S smartphone. A full description is available at the site where the data was obtained:

http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

Here are the data for the project:

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

You should 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.

Good luck!

Appendix B

Original Data Set Description

Davide Anguita et al., “Human Activity Recognition Using Smartphones Data Set,” http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones (accessed November 11, 2014).

Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

The description for “Human Activity Recognition Using Smartphones Data Set” is quoted from the reference, above.

Human Activity Recognition Using Smartphones Data Set

Download: Data Folder, Data Set Description

Abstract: Human Activity Recognition database built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors.

Data Set Characteristics:	Multivariate, Time-Series	**Number of Instances:**	10299	**Area:**	Computer
Attribute Characteristics:	N/A	**Number of Attributes:**	561	**Date Donated**	2012-12-10
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A	**Number of Web Hits:**	125474

Source:

Jorge L. Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto.
Smartlab - Non Linear Complex Systems Laboratory
DITEN - UniversitÃ degli Studi di Genova, Genoa I-16145, Italy.
activityrecognition '@' smartlab.ws
www.smartlab.ws

Data Set Information:

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

Check the README.txt file for further details about this dataset.

Attribute Information:

For each record in the dataset it is provided:
- Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
- Triaxial Angular velocity from the gyroscope.
- A 561-feature vector with time and frequency domain variables.
- Its activity label.
- An identifier of the subject who carried out the experiment.

Relevant Papers:

N/A

Citation Request:

[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

Feature Selection Information

The information below is quoted from **features_info.txt** in the original zipped file.

Feature Selection

=================

The features selected for this database come from the accelerometer and gyroscope 3-axial raw signals tAcc-XYZ and tGyro-XYZ. These time domain signals (prefix 't' to denote time) were captured at a constant rate of 50 Hz. Then they were filtered using a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20 Hz to remove noise. Similarly, the acceleration signal was then separated into body and gravity acceleration signals (tBodyAcc-XYZ and tGravityAcc-XYZ) using another low pass Butterworth filter with a corner frequency of 0.3 Hz.

Subsequently, the body linear acceleration and angular velocity were derived in time to obtain Jerk signals (tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ). Also the magnitude of these three-dimensional signals were calculated using the Euclidean norm (tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag).

Finally a Fast Fourier Transform (FFT) was applied to some of these signals producing fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccJerkMag, fBodyGyroMag, fBodyGyroJerkMag. (Note the 'f' to indicate frequency domain signals).

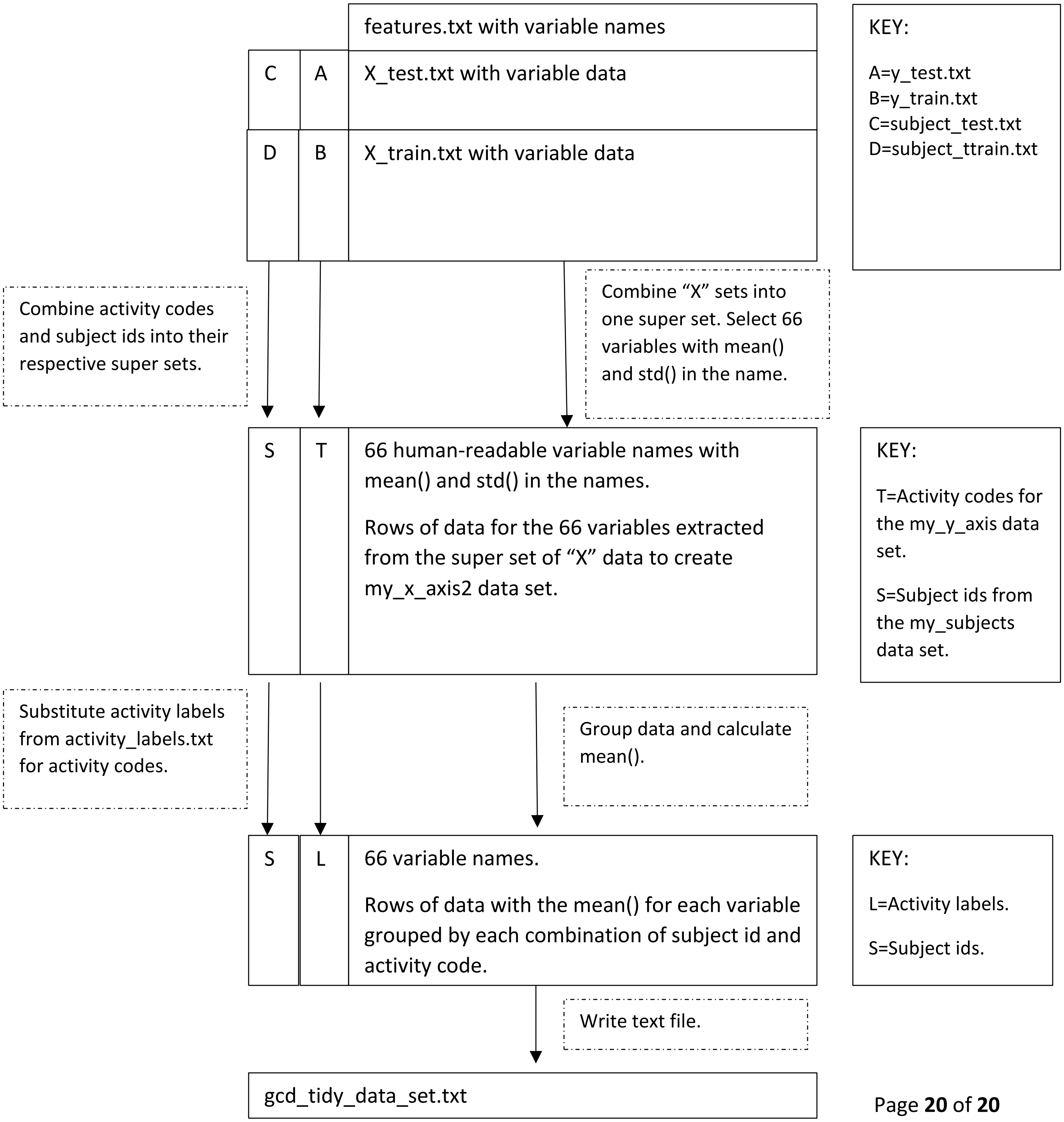
These signals were used to estimate variables of the feature vector for each pattern:

'-XYZ' is used to denote 3-axial signals in the X, Y and Z directions.

Appendix C

How the Original Data Knits Together

The following diagram illustrates the original data and how it connects to each other plus the transformation process it undergoes through the R script.


features.txt with variable names
C
A
X_test.txt with variable data
D
B
X_train.txt with variable data
KEY:
A=y_test.txt
B=y_train.txt
C=subject_test.txt
D=subject_ttrain.txt
Combine activity codes and subject ids into their respective super sets.
Combine "X" sets into one super set. Select 66 variables with mean() and std() in the name.
S
T
66 human-readable variable names with mean() and std() in the names.
Rows of data for the 66 variables extracted from the super set of "X" data to create my_x_axis2 data set.
KEY:
T=Activity codes for the my_y_axis data set.
S=Subject ids from the my_subjects data set.
Substitute activity labels from activity_labels.txt for activity codes.
Group data and calculate mean().
S
L
66 variable names.
Rows of data with the mean() for each variable grouped by each combination of subject id and activity code.
KEY:
L=Activity labels.
S=Subject ids.
Write text file.
gcd_tidy_data_set.txt