CodeBook

Stephen Parton

26 January 2019

Introduction

This is the CodeBook for the following course assignment:

Coursera/ John Hopkins University Data Getting and Cleaning Course - Wk4 Assignment

The R Markdown script ('CodeBook.Rmd', stored in this repo) runs the 'run_analysis.R' script then prepares an updated CodeBook based on then current data. It therefore does not need to be run unless changes are made to the source data or R code.

The output document is stored in the repo as CodeBook.md, CodeBook.pdf, and CodeBook.html.

This document is intended to satisfy Review Criteria 3 of the assignment. A Rmd file is being used so as to automate the process for ease of update, but also as Review Criteria 3 seems to suggest it should be done that way by stating:

"GitHub contains a code book that modifies and updates the available codebooks with the data to indicate all the variables and summaries calculated, along with units, and any other relevant information."

This is also the reason I included multiple CodeBooks (ie the reference to 'codebooks', in plural).

Run 'run_analysis.R' script

The R script 'run_analysis.R' runs the analysis, and is contained in the CookBook.Rmd file, although not in CookBook.md output file.

'all_data' Dataframe Description

This dataframe is the full 'tidied' version of the data.

Structure is summarised below:

	Length	Class	Mode
subject_id	339867	-none-	numeric
activity	339867	factor	numeric
date_time_id	339867	-none-	numeric
data_set	339867	-none-	character
signal	339867	factor	numeric
mean	339867	-none-	numeric
std_dev	339867	-none-	numeric

For information, the first 6 rows of data frame are as follows:

subject_id	activity	date_time_id	data_set	signal	mean	std_dev
1	LAYING	1	train	fBodyAcc-X	-0.9263140	-0.9103749
1	LAYING	1	train	fBodyAcc-Y	-0.8869485	-0.9056274
1	LAYING	1	train	fBodyAcc-Z	-0.9287921	-0.8810446

subject_id	activity	date_time_id	data_set	signal	mean	std_dev
1	LAYING	1	train	fBodyAccJerk-X	-0.9491758	-0.9502229
1	LAYING	1	train	fBodyAccJerk-Y	-0.8980041	-0.9036754
1	LAYING	1	train	fBodyAccJerk-Z	-0.9729599	-0.9824606

The above tables clearly show that the 'tidied' data is structured in a 'long/ narrow' format. The rationale for this approach (as opposed to a potentially much wider structure) is explained in more detail in the README file for this repo.

Transformations - run_analysis.R

The 'run_analysis.R' script handles all data loading, processing and output, as described in the following paragraphs:

1. Load Packages

```
# load packages
library(tidyverse)
library(data.table)
```

- tidyverse is used for transformations
- data.table is used for data loads (fread) which also allows for selection of columns required so all of the data sets do not need to be loaded (only mean and standard deviation variables required).

2. Read Activity Labels and features

```
# read data common to test and train
activity_labels <- fread("activity_labels.txt")
features <- fread("features.txt",stringsAsFactors = FALSE,select = 2 )

# extract feature names and indices to allow selective load of test and training data
feature_names_index <- grep("mean(\\()|std",features$V2,value = FALSE)
feature_names <- grep("mean(\\()|std",features$V2,value = TRUE)
```

The 'feature_names_index' and 'feature_names' are extracted for just 'mean' and 'std' variables so they can then be used to limit the load of the dataset. This is done as the Assignment requires only mean and std data to be included in the final tidied dataset.

3. Read Test and Training Datasets as well as labels

```
# read test data - only those cols required
subject_test <- fread("subject_test.txt")
x_test <-fread("x_test.txt", select = feature_names_index)
y_test <- fread("y_test.txt")

# read training data - only those cols required
subject_train <- fread("subject_train.txt")
x_train <-fread("x_train.txt", select = feature_names_index)
y_train <- fread("y_train.txt")
```

- 'x_test' and 'x_train' text contain data for test and training data sets while y_test and y_train contain activity labels (ie a 1-6 number for each row in the test and train datasets respectively).
- 'subject_test' and 'subject_train' contain subject labels (a 1-30 number for each row in the test and train datasets respectively)

The 'select' option in the fread function limits the columns selected to the mean and std measures as per specified in previous step.

4. Add column names to test and train datasets

```
# add colnames
names(x_test) <- feature_names
names(x_train) <- feature_names

# add test/train identifiers and x,y
test <- cbind(subject_id = subject_test$V1, data_set = "test",
              label_id = y_test$V1, x_test)
train <- cbind(subject_id = subject_train$V1, data_set = "train",
               label_id =y_train$V1, x_train)
```

An additional column has also been added to identify which data source row came from which dataset ('test' or 'train'). This was not specified in Assignment requirements but helps to maintain an audit trail back to source data, and does not impede further analysis.

5. Combine test and train datasets and further tidying as follows

```
# combine train and test datasets,
# join to activity labels,
# reorder cols and
# drop 'label_id' column (now we have included activity labels instead)
all_data <- rbind(test,train) %>%
        inner_join(activity_labels, by=c("label_id" = "V1")) %>%
        select(subject_id,activity = V2, everything()) %>%
        select(-label_id)

# add a proxy sequence for the date_time_id, within subject,data set type, activity
#  - not specifically requested but does no harm and leaves an audit trail back to original data
all_data <- all_data %>%
        group_by(data_set,subject_id,activity) %>%
        mutate(date_time_id = row_number())%>%
        select(subject_id, activity ,date_time_id,everything()) %>%
        ungroup()
```

The date/time identifier is also not specifically required but does also help with audit trail.

Data is now in one 'all_data' dataset

6. Gather data into Required Columns and Tidy Names

```r
# gather and tidy
all_data <- all_data %>%
        gather(key = signal, value = value, c(-subject_id, -activity, -date_time_id, -data_set)) %>%
        mutate(signal = str_remove(signal,"[//(][//)]")) %>%
        extract(col = "signal",into = "metric",regex = "(mean|std)",remove = FALSE)

# fix variable names by removing mean and std description from signal names, add factors
all_data$signal = as.factor(str_replace_all(all_data$signal,c("-mean"="","-std"="")))
all_data$activity = as.factor(all_data$activity)

# put mean and std in separate columns, these being the measures of the various signals
all_data <- all_data %>%
        spread(metric,value) %>%
        rename(std_dev = std)
```

This step structures the data into the selected 'tidy' data structure, and removes reference to mean and std in the 'signal' descriptions column and renames 'std' column to 'std_dev' for more clarity

As discussed in the 'Tidy Data' structure section in the README.md file:

- The 'signals' have not been parsed further as the variables are considered to be the mean and standard deviation (std_dev), while the observations are the signals. This structure then already meets 'tidy data' requirements (although a wider structure will also probably meet those requirements, depending on next stage analysis requirements)
- the signals are as defined. I am not a subject matter expert so have no reason to break the signals' names into their component parts,for example by extacting 'f' and 't' components into separate columns for the mean and std_dev measures. Doing so, would also introduce NULL components, as 'f' and 't' signals are not identical.
- the Summary Analysis required can be easily accomplished with the data in this structure, so there is no reason to complicate the code further. I may have reconsidered if I knew what any further analysis might be, so could perhaps judge if another structure is more appropriate. I do not have that information however.

7. Prepare Summary Analysis and write output to file

```r
# summarise data - as requested and as averages of mean and std_dev within subject,activity and signal.
summary_data <- all_data %>%
        group_by(signal,activity,subject_id) %>%
        summarise(average_mean = mean(mean),average_std_dev = mean(std_dev))

# write to file as txt
 write_tsv(all_data,"all_data.txt")
 write_tsv(summary_data,"summary_data.txt")
```

Column Specifications - 'all_data'

1. Subject_id

30 subjects were used in the original experiment. They are only identified by a number (integer) ranging from 1-30 for both test and training data, subsequently joined. We have no more information about the subjects, so assumed to be just an id field.

2. Activity

6 Activities were defined in the original experiment, and joined to the output in the 'tidy' dataframe as factors, as follows:

Activity
LAYING
SITTING
STANDING
WALKING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS

3. Date/Time ID

This field was added to maintain an audit trail to the source data. It comprises an int sequence number within Activity, Subject and Source dataset. It is not required for this Assignment, but perhaps might be useful in subsequent analyses.

The original exeriment documentation provided has very little focus on date/time or any ordering by date/time. The focus of the observations seems to be more on the derived signals from the activites undertaken by each subject. So this field may well be superfluous to future analytical requirements.

4. Dataset

- 'Test' v 'train', included as a factor.
- included for audit trail purposes, but maybe not required for future analysis.

5. Signal Observation

Unique Signals are as follows:

Signals
fBodyAcc-X
fBodyAcc-Y
fBodyAcc-Z
fBodyAccJerk-X
fBodyAccJerk-Y
fBodyAccJerk-Z
fBodyAccMag
fBodyBodyAccJerkMag
fBodyBodyGyroJerkMag
fBodyBodyGyroMag
fBodyGyro-X
fBodyGyro-Y
fBodyGyro-Z
tBodyAcc-X
tBodyAcc-Y
tBodyAcc-Z
tBodyAccJerk-X
tBodyAccJerk-Y
tBodyAccJerk-Z

Signals
tBodyAccJerkMag
tBodyAccMag
tBodyGyro-X
tBodyGyro-Y
tBodyGyro-Z
tBodyGyroJerk-X
tBodyGyroJerk-Y
tBodyGyroJerk-Z
tBodyGyroJerkMag
tBodyGyroMag
tGravityAcc-X
tGravityAcc-Y
tGravityAcc-Z
tGravityAccMag

As discussed in the 'Tidy Stucture' section of the 'README' file, the above signals could be broken down further:

- f (frequency) v t (time) signals
- 'X' v 'Y' v 'Z' directional signals
- the residual 'Body...' component (and maybe even Acc v Gyro etc separately)

The experiment does not do this but refers to all the above as Signals, so without a good reason to further process data, I have retained all the above as the observations.

The exact meaning of these signals is beyond the scope of this paper, and they are not described in detail in the source data files in any case.

'Features_info.txt' file does have some additional information about the process of their derivation.

6/7. Mean and Std_Dev

The experiment 'features_info.txt' file states:

"The set of variables that were estimated from these signals are:

- *mean(): Mean value*
- *std(): Standard deviation ....* "

The above list includes just the two numerical variables the assignment required to be included. The full set of variables numbers 17 in total, as detailed in the 'features_info.txt' file in the source data files.

Note further that I have only selected the mean() and std() components of the source datasets, so not included 'MeanFreq' for example. This might perhaps required further clarification of future analytical requirements