Clean Data Project ***Read-Me***

B Baillargeon

Thursday, April 23, 2015

Project Synopsis

Students in the Johns Hopkins Bloomberg School of Public Heath MOOC(*Massive Open Online Course*) *Getting and Cleaning Data, getData-013*, must demonstrate an ability to collect, work with, and clean a data set. **The goal is to prepare tidy data that can be used for later analysis.**

We accessed the data derived from the UC Irvine data set "Human Activity Recognition." Unzipping this file creates the original file structure of the research text files, including the descriptions of the measured variables and subjects.

Files delivered with this project

I'm creating a repo from my RStudio 'clean data project' project, which has the reference data set and its structure in its zipped-up directory structure, rooted at `\UCI HAR Dataset\`. Also, I've the **output** `\Tidydata\` folder, containing a time-stamp of the most recent pass, and the `TidySubjActAverages.txt` dataset per the instructions. *This readme* and 2 *codebook* files are at this the working directory level of the repo/ project as *SubjectActivityAverages.Rmd* and *SubjectActivityAverages_info.Rmd*.
Note that the ultimate output goal is a dataset which has a `mean()` calculated over the many readings of *each* of the targeted 66 mean-and-standard-deviation variables' columns. These *accumulators* for these observations are organized to produce **one** *arithmetic* mean (*average*) value, per Subject in the research, and within that Subject's observations, grouped by the six Activities the researchers tracked.

I.e., One record per Subject "A" per the activity (e.g.,"Standing") with all 66 variables' means calculated and in that same record. In one example of Subject 10, there are 54 separate readings for each of these 66 variables, for each of the six Activities. These **54** records become **one** that carries the average of those 54 records' values, for that *Subject* performing that *Activity*.

Each of the 30 Subjects will have 6 records -1 each for each of the 6 Activities -and this should give us 180 records/ observations, and it does. 180 observations of 68 variables (Subject, Activity, and 66 Mean()/Std() averages).

Relevant reference files

The original file structure included several sub-directories and text files. In the structure below,
* the *italicized* are "informative" or structural and not actual research finding data;
* the **bolded** are ***relevant*** files that have data we are assessing in the scope of this project; and
* all others are not used in the scope of this project

\ **UCI HAR Dataset**(dir)
. . . \ **test**(dir)
. . . . . . \ Inertial Signals(dir) *none are used*
. . . . body_acc+ *+-several entries per*
. . . . body_gyro+

... ... total_acc+
... ... **subject_test.txt — tells us which subject is recorded** per observation
... ... **X_test.txt — massive list of observations across 561 columns(variables)**
... ... **y_test.txt — index of the activity the subject is performing** per observation
... \ **train**(dir)
... ... \ Inertial Signals(dir) *none are used*
... ... body_acc+ *+-several entries per*
... ... body_gyro+
... ... total_acc+
... ... **subject_train.txt — tells us which subject is recorded** per observation
... ... **X_train.txt — massive list of observations across 561 columns(variables)**
... ... **y_train.txt — index of the activity the subject is performing** per-observation
... *README.txt — describes all other files*
... *features.txt — names the 561 column variables*
... *features_info.txt — summarizes the 561 variables and their content*
... **activity_labels.txt — simple list mapping the activity index to the activity description**

Columns containing the targeted `mean()` and `std()` (*standard deviation*) variables

Within the large **X_test.txt** and **X_train.txt** files of **561** columns, we are directed to extract only the `mean()` and standard deviation `std()` readings. These readings are spread across **66** columns, as identified in the *features.txt* file.
These columns, when combined with the

- **subject_train.txt** \ **subject_test.txt** (*subject: who is doing*) and the
- **y_train.txt** \ **y_test.txt** (*activity: what they did*)

yield a **wide** data frame of the mean, std, activity, and subject ID for the observations. The last step of making the dataset 'tidy' will "melt" -narrow -the **561** measures' columns down to ***2***: *one* tracking the column variable name, and the *second* tracking its value. After working the data set to generate the per-subject-per-activity averages(mean) of each of the 66 unique `mean()` and `std()` columns for each subject-activity pairing, I will recast these back out to realize substantially reduced summary.

Time Related `mean`, `std` Variables	**X_...txt** Column #s (40)	Notes
tBodyAcc-_____*()*-[X|Y|Z]	*1:6*	**1-3** _____*()* is `mean()` on X, Y, & Z axes
...	...	**4-6** _____*()* is `std()` on X, Y, & Z axes
tGravityAcc-_____*()*-[X|Y|Z]	*41:46*	
tBodyAccJerk-_____*()*-[X|Y|Z]	*81:86*	
tBodyGyro-_____*()*-[X|Y|Z]	*121:126*	
tBodyGyroJerk-_____*()*-[X|Y|Z]	*161:166*	
tBodyAccMag-_____*()*	201 *&* 202	_____*()* is `mean()` & `std()`, respectively
tGravityAccMag-_____*()*	214 *&* 215	
tBodyAccJerkMag-_____*()*	227 *&* 228	
tBodyGyroMag-_____*()*	240 *&* 241	
tBodyGyroJerkMag-_____*()*	253 *&* 254	

Frequency Related `mean`, `std` Variables	**X_...txt** Column #s (26)	Notes
fBodyAcc-_____*()*-[X|Y|Z]	*266:271*	**1-3** _____*()* is `mean()` on X, Y, & Z axes
...	...	**4-6** _____*()* is `std()` on X, Y, & Z axes
fBodyAccJerk-_____*()*-[X|Y|Z]	*345:350*	
fBodyGyro-_____*()*-[X|Y|Z]	*424:429*	
fBodyAccMag-_____*()*	503 *&* 504	_____*()* is `mean()` & `std()`, respectively
fBodyAccJerkMag-_____*()*	516 *&* 517	
fBodyGyroMag-_____*()*	529 *&* 530	
fBodyGyroJerkMag-_____*()*	542 *&* 543	

Extracting these 66 values into a data frame, then assigning relevant names for these *unnamed columns* as guided by the original data set names above (from the *features_info.txt* file) gives us a useful data.frame. I chose to rename the mid-process/ working `data.frame` column variables with *mean* and *std* as the *trailing* element of the names, not embedded, and *removed underscores & parenthesis*; this helps readability and the reshape flow.

Creating Tidy Data

My process for pulling the data from the several files and finally generating the 180x68 tidy dataset:

- performed for both the *train* and the *test* datasets:
  - `read.table()` to get the ___.txt data into data.tables
  - `subset via DT[]` the codebook showed the 33 mean and 33 st-dev indices, this made the working set of 66 numeric columns
  - `read.fortran()` to get the fixed-position data for the activity labels
  - `cbind()` bind the subject, the activity descriptive label columns (both **as factors**) to the numeric data.tables
  - `col.names()` to assign workable, consistent names to data.tables

- `dplyr::arrange()` to order the records by Subject then by Activity
- `merge()` to bring test and train datasets into one
- `reshape2::melt()` to drive the 66 columns into 1 'variable' column, and their values in to the 'value' column; very narrow and very long.
- `group_by()` to get all records ordered by Subject then by Activity
- `dcast()` to recast the molten data.table to reflect the formula `Subject + Activity + Mean_StdDev_Var ~ .` and creating the aggregation of these using `mean`
- `dcast()` a 2nd time to re-distribute the columns back to a compact form via the formula `Subject + Activity ~ Mean_StdDev_Var`
- `write.table()` to drive the resulting 180x68 table to the file required by the project instructions.
