HARUS Codebook

Elizabeth Lee

12/18/2021

Codebook for the HARUS Averages Data Set

"HARUS" is an acronym of "Human Activity Recognition Using Smartphones", which is the name of the source dataset used in this analysis. All credit for the creation of the source data set belongs to:

```
Human Activity Recognition Using Smartphones Dataset
Version 1.0

Jorge L. Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto.
Smartlab - Non Linear Complex Systems Laboratory
DITEN - Universit‡ degli Studi di Genova.
Via Opera Pia 11A, I-16145, Genoa, Italy.
activityrecognition@smartlab.ws
www.smartlab.ws
```

The full description and original dataset are available at:

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

The data used in this project were downloaded from:

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

on 16 December 2021 at 16:05 America/Los_Angeles time. Processing was performed in RStudio, Version 1.4.1106, with R version 4.1.2, running on MacOS 10.16.0.

The Human Activity Recognition Using Smartphones Dataset, as described by its authors in the dataset's README file, records the results of experiments which captured sensor signals from the embedded accelerometer and gyroscope of smartphones worn by the volunteers while performing specific activities. THe experiment defined six activities, recorded in the file activity_labels.txt; each volunteer performed each of the six activities multiple times, for a total of 10,299 observations.

These raw data were processed by the authors into 561 components of motion, referred to as "features".

```
library(data.table)
library(tidyr)

tibble(fread("./Data/features.txt"))
```

```
## # A tibble: 561 x 2
##       V1 V2
##    <int> <chr>
##  1     1 tBodyAcc-mean()-X
##  2     2 tBodyAcc-mean()-Y
##  3     3 tBodyAcc-mean()-Z
##  4     4 tBodyAcc-std()-X
##  5     5 tBodyAcc-std()-Y
```

```
##  6     6 tBodyAcc-std()-Z
##  7     7 tBodyAcc-mad()-X
##  8     8 tBodyAcc-mad()-Y
##  9     9 tBodyAcc-mad()-Z
## 10    10 tBodyAcc-max()-X
## # ... with 551 more rows
```

These feature measurements are divided in two text files, the training set, X_train.txt, and the test set, X_test.txt; each row of these files represents the observation of one experiment, that is, the features of one activity performed one time by one volunteer.

Accompanying each of these data files are text files containing the subject identifiers (subject_train.txt, subject_test.txt) and activity identifiers (y_train.txt, y_test.txt) that correspond to the rows of their respective feature data files.

The HARUS Averages data set is derived only from the processed data (X_train.txt and X_test.txt) and accompanying subject and activity identifier files. The dataset consists of

```
- a Readme file
- this codebook
- an R script file, run_analysis.R
- a data file, HARUS Averages.csv
```

run_analysis.R is the script used to process the source data and generate the HARUS Averages.csv file.

HARUS Averages.csv is a comma-separated text file with headers which contains the averages of the mean and standard deviation of time and frequency features of the combined train and test data sets.

The data consist of four variables:

- **SubjectID**: *integer*, range 1:30
  - Meaning: unique identifiers of the 30 volunteers who participated in the Human Activity Recognition Using Smartphones experiments
  - Source: subject_test.txt, subject_train.txt
  - Processing: none
- **Activity**: *factor* with 6 levels
  - Meaning: unique identifiers of the six activities performed during the experiments
  - Source: activity_labels.txt; y_test.txt, y_train.txt
  - Processing: the factor variable is based on activity_labels.txt; see note 1; the values of the Activity variable were read from the y_*.txt files unmodified

```
  Source                 Factor: activity
                         Level   Label
  1 WALKING                1     Walking
  2 WALKING_UPSTAIRS       2     WalkingUpstairs
  3 WALKING_DOWNSTAIRS     3     WalkingDownstairs
  4 SITTING                4     Sitting
  5 STANDING               5     Standing
  6 LAYING                 6     Lying
```

- **Feature**: *character*, n = 66
  - Meaning: the names of selected features
  - Source: features.txt
  - Processing: the selected features are the means and standard deviations of time and frequency variables calculated from acceleration and gravity signals by the original authors; see note 2
- **Average**: calculated *double*, range [-1,1]
  - Meaning: each value represents the average of the values selected features, for all repetitions of an activity by one subject
  - Source: X_test.txt, X_train.txt
  - Processing: see Processing, below

Note 1: The source activity labels were reformatted slightly for readability in defining the activity factor. In addition, the label LAYING was changed to the correct English verb for the activity performed, Lying. The mistake of confusing the verbs to lay and to lie is common even among native English speakers, but given that this is a scientific study characterizing motion during specific activities, the distinction is relevant.

```
to lay: to put [something] down, especially gently or carefully

to lie: to be in or assume a horizontal or resting position on a
        supporting surfact

        (source: Macintosh Dictionary app, version 2.3.0)
```

The action of lying down which was performed by the volunteers is not at all the same as the action of laying something down. In respect for the scientific rigor of the experiment's authors, the correct terminology should be used.

Note 2: The names assigned to features by the authors of the Human Activity Recognition Using Smartphones experiments are necessarily somewhat cryptic, given the challenge of devising unique descriptive names that are still workably short for 561 measurements. The original feature names were read from the source file and modified slightly by regex substitution for readability by: - removing () and - characters - standardizing the use of case - expanding the initial "t" and "f" that identify time and frequency variables to "time" and "freq" - removing what appeared to be a corruption in the original data of several frequency names where the string "Body" was duplicated:

```
    ex. fBodyBodyAccJerkMag-mean() -> freqBodyAccJerkMagMean
```

Processing

Initial examination of the source dataset showed that the data are clean and straightforward to work with.

```
test <- fread("./Data/test/X_test.txt")
train <- fread("./Data/train/X_train.txt")
dim(test); any(is.na(test))
```

```
## [1] 2947  561
```

```
## [1] FALSE
```

```
dim(train); any(is.na(train))
```

```
## [1] 7352  561
```

```
## [1] FALSE
```

The source dataset's README file states that there are 561 features, and that each row of the X_*.txt files is a feature vector. The dimensions of the data files confirm that each row has 561 columns, as expected, and a check for missing values detects none.

```
features <- fread("./Data/features.txt")
head(features, n = 4L)
```

```
##    V1               V2
## 1:  1 tBodyAcc-mean()-X
## 2:  2 tBodyAcc-mean()-Y
## 3:  3 tBodyAcc-mean()-Z
## 4:  4  tBodyAcc-std()-X
```

```
dim(features)
```

```
## [1] 561   2
```

```
any(is.na(features))
```

```
## [1] FALSE
```

As expected, there are 561 features, which identify the values in the columns of the numeric data files; no values are missing.

```
testNames <- names(test)
trainNames <- names(train)
head(testNames, n = 4L)
```

```
## [1] "V1" "V2" "V3" "V4"
```

```
head(trainNames, n = 4L)
```

```
## [1] "V1" "V2" "V3" "V4"
```

```
identical(testNames, trainNames)
```

```
## [1] TRUE
```

A check of the column names of the read-in test and train data files shows that default column names have been generated, and that these are identical between the two files. The two data tables can simply be clipped together using rbind().

However, these are fairly large data tables and we are only interested in a subset of the features they contain, namely, the means and standard deviations of the following variables, described in this exceprt from the features_info.txt file of the source dataset:

> 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.
>
> tBodyAcc-XYZ
> tGravityAcc-XYZ
> tBodyAccJerk-XYZ
> tBodyGyro-XYZ
> tBodyGyroJerk-XYZ
> tBodyAccMag
> tGravityAccMag
> tBodyAccJerkMag
> tBodyGyroMag
> tBodyGyroJerkMag
> fBodyAcc-XYZ
> fBodyAccJerk-XYZ
> fBodyGyro-XYZ
> fBodyAccMag
> fBodyAccJerkMag
> fBodyGyroMag
> fBodyGyroJerkMag

The following regular expression is used to select the desired features and their corresponding row numbers.

```
tmp <- grep("mean[^A-Z,a-z]()|std[^A-Z,a-z]()", features$V2, value = TRUE)
avg_std <- features[features$V2 %in% tmp, ]
head(avg_std, n = 4L)
```

```
##    V1               V2
## 1:  1 tBodyAcc-mean()-X
## 2:  2 tBodyAcc-mean()-Y
## 3:  3 tBodyAcc-mean()-Z
## 4:  4  tBodyAcc-std()-X
```

The resulting set of features contains 66 variables.

```
nrow(avg_std)
```

```
## [1] 66
```

```
avg_std$V1
```

```
##  [1]   1   2   3   4   5   6  41  42  43  44  45  46  81  82  83  84  85  86 121
## [20] 122 123 124 125 126 161 162 163 164 165 166 201 202 214 215 227 228 240 241
## [39] 253 254 266 267 268 269 270 271 345 346 347 348 349 350 424 425 426 427 428
## [58] 429 503 504 516 517 529 530 542 543
```

The first column, V1, of the selected features data table contains the original row numbers of the full feature set, which are passed to *fread()* via the *select* argument to read in only the desired columns. The second column, V2, are passed via the *col.names* argument to provide the column headers of the imported data tables for the test and train data.

Before rbinding the test and train data tables, it is necessary to attach the subject and activity identifiers for each row of the respective tables. Examination of the subject_*.txt and y_*.txt files demonstrates that each contains a single vector whose length matches the row count of the respective data file. Therefore these files can simply be attached to the lefthand side of the appropriate data table using cbind(). Before cbinding the activity identifiers, the data is converted from integers to a factor defined to match the activity labels in the original dataset (activity_labels.txt), as described above.

```
testSubject <- fread("./Data/test/subject_test.txt")
dim(testSubject); any(is.na(testSubject))
```

```
## [1] 2947    1
```

```
## [1] FALSE
```

```
testActivity <- fread("./Data/test/y_test.txt")
dim(testActivity); any(is.na(testActivity))
```

```
## [1] 2947    1
```

```
## [1] FALSE
```

```
trainSubject <- fread("./Data/train/subject_train.txt")
dim(trainSubject); any(is.na(trainSubject))
```

```
## [1] 7352    1
```

```
## [1] FALSE
```

```
trainActivity <- fread("./Data/train/y_train.txt")
dim(trainActivity); any(is.na(trainActivity))
```

```
## [1] 7352    1
```

```
## [1] FALSE
```

Calculation of the averages for each feature, by activity and subject is a straightforward matter of splitting the combined data table first on subject, then on activity, and calculating the means of all the features in each resulting group.

The output data table has 68 columns: the 2 identifier columns, SubjectID and Activity, and 66 feature columns. This is messy, because the dataset actually contains only four variables:

- SubjectID (OK)
- Activity (OK)
- Feature, spread across the column names
- Average, spread across the column values

The pivot_longer() function is used to reshape the data table by storing the features (column names) in a variable named "Feature", and the values (the averages) in a variable named "Average". The resulting data table now conforms to Hadley Wickham's principles for tidy data:

1. Each variable forms a column
2. Each observation forms a row
3. Each type of observational unit forms a table

```
tibble(read.csv("./HARUS Averages.csv"))
```

```
## # A tibble: 11,880 x 4
##    SubjectID Activity Feature             Average
##        <int> <chr>    <chr>                 <dbl>
##  1         1 Walking  timeBodyAccMeanX      0.277
##  2         1 Walking  timeBodyAccMeanY     -0.0174
##  3         1 Walking  timeBodyAccMeanZ     -0.111
##  4         1 Walking  timeBodyAccStdX      -0.284
##  5         1 Walking  timeBodyAccStdY       0.114
##  6         1 Walking  timeBodyAccStdZ      -0.260
##  7         1 Walking  timeGravityAccMeanX   0.935
##  8         1 Walking  timeGravityAccMeanY -0.282
##  9         1 Walking  timeGravityAccMeanZ -0.0681
## 10         1 Walking  timeGravityAccStdX   -0.977
## # ... with 11,870 more rows
```