Step by step description of run_analysis.R

P. A. Ortiz Otalvaro

20 August 2019

1. Load packages

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.1
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.6.1
```

2. Read data

2.1. Test and train data

Test and train data are read and stored in separated data frames: `test_x` and `train_x`. A new column is added to each data frame to identify its original subset.

```
test_x <- read.table("./UCI HAR Dataset/test/X_test.txt", header = FALSE, stringsAsFactors = FALSE)
train_x <- read.table("./UCI HAR Dataset/train/X_train.txt", header = FALSE, stringsAsFactors = FALSE)
test_x <- dplyr::mutate(test_x, subset = "test")
train_x <- dplyr::mutate(train_x, subset = "train")
```

2.2 Activity data

In files test_y.txt and train_y.txt there are all the label numbers of the activities corresponding to each observation in the test and train sets respectively. These values are read in independent data frames: `test_activi_label` and `train_activi_label`. These are data frames with only one column. The column name "ActivityLabel" is assign to them:

```
test_activi_label <- read.table("./UCI HAR Dataset/test/y_test.txt", header = FALSE, stringsAsFactors =
train_activi_label <- read.table("./UCI HAR Dataset/train/y_train.txt", header = FALSE, stringsAsFactor
colnames(test_activi_label) <- "ActivityLabel"
colnames(train_activi_label) <- "ActivityLabel"
```

Furthermore, the activity names corresponding to the activity label numbers can be found in file activity_labesl.txt. This correspondance is stored in data frame `activity_names`:

```
activity_names <- read.table("./UCI HAR Dataset/activity_labels.txt", header = FALSE, stringsAsFactors =
colnames(activity_names) <- c("ActivityLabel", "ActivityName")
```

2.3 Subject data

The last data set to read is the one giving the subject ID corresponding to each observation in train and test data sets. This is found in files subject_test.txt and subject_train.txt. They are read and stored in data frames `test_subject` and `train_subject`.

```
test_subject <- read.table("./UCI HAR Dataset/test/subject_test.txt", header = FALSE, stringsAsFactors =
colnames(test_subject) <- "subject"
train_subject <- read.table("./UCI HAR Dataset/train/subject_train.txt", header = FALSE, stringsAsFactor
colnames(train_subject) <- "subject"
```

3. Merge datasets

All data is separated in train and test subsets. Before concatenating all of data, all train and test subsets are merged:

- `har_dataset` : test and train datasets concatenated (aggregated) one on top of the other.
- `activity_labels`: labels of test and train datasets concatenated (aggregated) one on top of the other.
- `subject`: subjects corresponding to test and train datasets concatenated (aggregated) one on top of the other.

After the datasets are merged, names corresponding to each feature are extracted from file features.txt and assigned to `har_dataset` columns.

```
har_dataset <- dplyr::bind_rows(test_x, train_x)
activity_labels <- dplyr::bind_rows(test_activi_label, train_activi_label)
subject <- dplyr::bind_rows(test_subject, train_subject)

col_names <- read.delim("./UCI HAR Dataset/features.txt", header = FALSE, stringsAsFactor = FALSE, sep =
colnames(har_dataset) <- make.names(t(col_names[2]), unique = TRUE)
```

Now, each observation in the `har_dataset` can be assigned its activity label, activity name and subject:

```
har_dataset <- dplyr::bind_cols(activity_labels, har_dataset)

har_dataset <- dplyr::right_join(x = activity_names, y = har_dataset, by = "ActivityLabel")

har_dataset <- dplyr::bind_cols(subject, har_dataset)
```

4. Extract tidy dataset

Extract columns containing mean or standard deviation of each observation (`selected_features`), save this is a new data frame: `har_subset`. And write it to file titanic_clean.csv.

```
selected_features <- colnames(har_dataset)[grepl("mean|std", colnames(har_dataset))]
har_subset <- har_dataset[selected_features]
data.table::fwrite(har_subset, file = "har_clean.csv" )
```