Getting and Clening Data course project

Przemysław Skowron

8/7/2020

This document describes in details run_analysis.R script created as final project for the Getting and Cleaning Data course.

Prerequisites and assumptions

It is assumed that the source data package is downloaded from the location indicated in the instruction and zip file is extracted in the ~/R folder (creating new folder name ~/R/UCI HAR Dataset with all content inside). The script will terminate if any file / folder is missing or there is lack of perimssions to read/write file.

Script flow

The run_analysis.R script executes the following steps in order to create output dataset:

1. **Sets R working directory** to "~/R/UCI HAR Dataset".
2. **Loads dplyr package** using library() function.
3. **Merges 2 datasets (train and test)** using bind_rows function from dplyr package. This is performed for main variable data, vector of subjects and vector of activities. After merge is completed test and train datasets are removed using rm() function to clean up memory. Only "combined" datasets / vestors are retained in the memory.
4. **Extracts mean and standard deviation variables**:
   - Read vector of variable labels from the features.txt file;
   - Identify variables containing mean() or std() character strings; and their position in the list of variables using grepl and filter functions;
   - Then select those variables from combined_data and overwrite combined_data dataset.
5. **Transforms variable names into more descriptive ones** using mutate, sub and gsub functions as well as regular expressions:
   - Replaces "t" and "f" prefixes with "time" and "freq";
   - Uppercase first letter in "mean" and "std" strings;
   - Removes parentheses and dash characters (replace them with "").
   - Finally names columns in combined_data with transformed variable names.
6. **Replaces activity numbers with descriptive names** in the combined_activities vector:
   - First loads activity labels from activity_labels.txt file;
   - Then uses left_join function to assign corresponding activity label to each activity id in combined_activities;
   - Finally, adds those labels as new variable in 1st column position (activityName) to our main dataset (combined_data) using mutate() function with parameter .before = 1.
7. **Adds subjectId as new variable** at the beginning of combined_data dataset.

8. **Computes average of each variable** for each activity and each subject:
    - Uses group_by function to group combined_data dataset by activityName and subjectId;
    - Uses summarize_all function to compute mean function on all non-grouped variables;
    - Finally, overwrites combined_data dataset;
9. **Saves the result** (combined_data dataframe) to output_data.txt file.
10. **Reset working directory** to old path.