HW03 - Multiple Regression Analysis

Stat 159, Fall 2016, Prof. Sanchez

Multiple Regression Analysis

The purpose of this assignment is to extend the scope of the previous HW. In addition to keep applying regression analysis in R—using `lm()`—you will also write some functions as well as their unit tests.

Your mission consists of reproducing the analysis from Section 3.2 (pages 71 to 82), from the book "An Introduction to Statistical Learning" (by James et al):

http://www-bcf.usc.edu/~gareth/ISL/

The data set is in the `Advertising.csv` file available here:

http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv

The main analysis involves carrying out a **multiple linear regression** with predictor variables `TV`, `Radio`, `Newspaper`, and the response variable `Sales`. The ultimate output will be a report replicating the following results from Chapter 3:

- Table 3.3 (page 72): Coefficient estimates of simple regression models: `Sales` on `TV`, `Sales` on `Radio`, and `Sales` on `Newspaper`. The book only shows two tables (those of `Radio` and `Newspaper`) but you should also include the table for `TV`.
- Table 3.4 (page 74): Coefficient estimates of the least squares model.
- Table 3.5 (page 75): Correlation matrix.
- Table 3.6 (page 76): RSE, R^2 and F-statistic of the least squares model.

The way you are going to work with the `.Rmd` file is similar to the previous HW. You will use this file mainly to write the narrative of the report. The R code for most of the analysis will be written in separate `.R` script files, with the main outputs being generated outside the `.Rmd` file. In addition, you will have to write some functions, and their corresponding tests.

Functions and Tests

A derived goal of this assignment is to practice writing functions and tests—not just simply writing code scripts. You are going to handle all the code inside a dedicated folder `code/` with three subdirectories, and one extra file `test-that.R`:

```
code/
  functions/
  scripts/
  tests/
  test-that.R
```

In order to carry out the regression analysis, you will keep using the function `lm()`. However, this time you can only use the `summary()` function to obtain the table of coefficients (for Table 3.4). To compute values for RSE, R^2 and F-statistic, you must write the following functions (note that they all take an object of class `"lm"` as input):

Residual Sum of Squares. Write a function `residual_sum_squares()` to calculate the RSS (residual sum of squares). This function should take the `"lm"` object as input, and the output is the RSS. See the formula of eq. 3.16 (page 69).

Total Sum of Squares. Write a function `total_sum_squares()` to calculate the TSS (total sum of squares). This function takes the `"lm"` object as input, and it returns the TSS. See description right below the formula of eq 3.17 (page 70).

R-squared. Write a function `r_squared()` to calculate the R^2 (coefficient of determination) This function takes the `"lm"` object as input, and it returns the R^2. See formula of eq 3.17 (page 70).

F-statistic. Write a function `f_statistic()` to calculate F-statistic. This function takes the `"lm"` object as input, and it returns the F-statistic. See formula of eq 3.23 (page 75).

Residual Standard Error. Write a function `residual_std_error()` to calculate the RSE (residual standard error). This function takes the `"lm"` object as input, and it returns the RSE. See formula of eq 3.25 (page 80).

Write your functions in an .R file called `"regression-functions.R"`. All the functions must be well documented, this means that you should include, for each function, descriptions about:

- what the function does
- what is the expected input
- what is the returned output

To write the unit tests, you will have to learn about the R package `"testhat"` (by Hadley Wickham). Some resources are:

`testthat`: Get Started with Testing (by Hadley Wickham)

https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf

Example of unit testing R code with testthat (by John Cook)

http://www.johndcook.com/blog/2013/06/12/example-of-unit-testing-r-code-with-testthat/

Testing chapter of R packages (by Hadley Wickham)

http://r-pkgs.had.co.nz/tests.html

Running Tests

For testing purposes, use the data `mtcars` (that comes in R), and compute a multiple regression of `mpg` on `disp` and `hp`.

```r
# multiple regression
reg <- lm(mpg ~ disp + hp, data = mtcars)

# summary of 'reg'
regsum <- summary(reg)
```

You can use the following reference values to compare the output of your functions, and test whether you get the right computations:

```r
# RSS
sum(reg$residuals^2)
```

```
## [1] 283.4934
```

```r
# TSS
sum((mtcars$mpg - mean(mtcars$mpg))^2)
```

```
## [1] 1126.047
```

```r
# RSE
regsum$sigma
```

```
## [1] 3.126601
```

```r
# R2
regsum$r.squared
```

```
## [1] 0.7482402
```

```r
# F-statistic
regsum$fstatistic[1]
```

```
##    value
## 43.09458
```

We are giving you freedom on the number-and-type of contexts, tests, and expectations of your unit tests. Like many other things in programming, writing unit tests is a practice that takes time to become skilled. Take this HW as the opportunity to "break the ice" with such tests. More info about running tests with `"testthat"` in lab07 of the course's github repo.

File Structure

The complete file-structure for this assignment is the following:

```
stat159-fall2016-hw03/
    .gitignore
    README.md
    LICENSE
    Makefile
    session-info.txt                      # produced by session-info-script.R
    code/
      README.md
      test-that.R
      functions/
        regression-functions.R
      scripts/
        eda-script.R
        regression-script.R
        session-info-script.R
      tests/
        test-regression.R
    data/
      README.md
      Advertising.csv
      eda-output.txt                       # produced by eda-script.R
      correlation-matrix.RData             # produced by eda-script.R
      regression.RData                     # produced by regression-script.R
    images/
        histogram-sales.png                 # produced by eda-script.R
        histogram-tv.png                    # produced by eda-script.R
        histogram-radio.png                 # produced by eda-script.R
        histogram-newspaper.png             # produced by eda-script.R
        scatterplot-matrix.png              # produced by eda-script.R
        scatterplot-tv-sales.png            # produced by regression-script.R
        scatterplot-radio-sales.png         # produced by regression-script.R
        scatterplot-newspaper-sales.png     # produced by regression-script.R
        residual-plot.png                   # produced by regression-script.R
        scale-location-plot.png             # produced by regression-script.R
        normal-qq-plot.png                  # produced by regression-script.R
    report/
        report.Rmd
        report.pdf
```

Files

- Code scripts:
  - `eda-script.R` reads in the `Advertising.csv` data set, and computes summary statistics, histograms for all the variables (`TV`, `Radio`, `Newspaper`, and `Sales`), matrix of correlations among all variables, and scatterplot-matrix (pairwise scatterplots). The summary statistics (clearly labeled) and the matrix of correlations, will be saved in a file `eda-output.txt`. In addition to including the correlation matrix in `eda-output.txt`, save it also in binary format `correlation-matrix.RData`. In turn, each exploratory chart is saved in PNG format.
  - `regression-script.R` reads in the `Advertising.csv` data set, and computes a `"regression"` object—via `lm()`—as well as the summary of such regression object—via `summary()`. These objects are saved in the file `regression.RData`. This script also produces the three diagnostics plots `residual-plot.png`, `scale-location-plot.png`, and `normal-qq-plot.png` (see `help(plot.lm)` for more info).
  - `session-info-script.R` is a script that includes `library()` calls to ALL the packages that you use for your project, as well as the output of the function `sessionInfo()`. Export the output via `sink()` to the file `session-info.txt` (this file is at the project's top level directory).
- Data Files:
  - `Advertising.csv` is the main data set. This file is downloaded from `http://www-bcf.usc.edu/~gareth/ISL/Advertising.csv`.
  - `eda-output.txt` is a text file containing the summary statistics of all the variables, and the correlation matrix. This file should be produced via `sink()` from the `eda-script.R` file.
  - `correlation-matrix.RData` is an R's binary format file containing the matrix of correlations among all variables. This file should be produced via `save()` from the `eda-script.R` file.
  - `regression.RData` is an R's binary format file containing the regression objects obtained when running `regression-script.R`. This file should be produced via `save()` from the `regression-script.R` file.
- Image files:
  - histograms of all the variables, produced from `eda-script.R`.
  - scatterplot-matrix of all variables (pairwise) produced from `eda-script.R`.
  - scatterplots for all individual simple regressions, with the corresponding fitted regression line. These files are an output of `regression-script.R`.
  - Plot of residuals against fitted values (from the multiple regression).
  - Scale-Location plot of $\sqrt{|residuals|}$ against fitted values (from the multiple regression).
  - Normal Q-Q plot (from the multiple regression).

  - The last three plots can be obtained with `plot.lm()`, and they will also be an output of `regression-script.R`.

- Report files:
  - `report.Rmd` is the source Rmd document used to generate the pdf report (you can generate an html report if you don't have LaTeX). `report.Rmd` sources the functions in `regression-functions.R`, reads in the objects of `correlation-matrix.RData` and `regression.RData`, and produces the tables and regression indices.
  - `report.pdf` is the generated pdf file from the Rmd document. Alternatively you could have a `report.html` file. Or even better, you can try to produce both types of output: pdf and html. It is not mandatory to have both types of files, but you can practice writing both targets in your `Makefile`.

- Licenses:
  - This project involves producing software content (R code), as well as media content (narrative, and images).
  - Choose a creative commons license for the media content (the legend of this license can be included in the main `README.md` file):
  - https://creativecommons.org/choose/
  - Choose one of the open source licenses for the code content (the text of this license is usually included in a separate file `LICENSE`):
  - https://opensource.org/licenses/category

- README files:
  - The main README file shoul have information about the title of the project, a brief description of what the project is about, another description of the file structure, instructions for a "competent" user on how to recreate your analysis and report, information about the licenses (for media-content and for software-content), as well as the author info.
  - In addition to the main readme, folders `data/` and `code/` should also contain their own readme files with a brief description of the files in such directories.

- `.gitignore` file:
  - use this file to include files that you don't want Git to track in your project. Typical examples to include in `.gitignore` are files such as `.Rhistory`, `.DS_Store` (for Mac OS), and secondary output files (e.g. LaTeX secondary files).

- Session Information:
  - The text file `session-info.txt` will be at the top level of the project. This file is generated by `session-info-script.R`, and it will serve as a file for reference (i.e. documentation) purposes: info about your the version of R, your platform, operating system, and used R packages. In future assignments we'll see how to include the information about the versions of Git, Make, LaTeX, etc.

Report

You should write a PDF report. This is the last assignment in which you can use an HTML if you don't have LaTeX (reports in future assignments will have to be made in PDF, which involves havin a LaTeX installation). Your report should be in the form of a paper with the following sections:

- Abstract
- Introduction
- Data
- Methodology
- Results
- Conclusions

Address those questions listed in page 75:

1. Is at least one of the predictors useful in predicting the response?
2. Do all predictors help to explain the response, or is only a subset of the predictors useful?
3. How well does the model fit the data?
4. How accurate is the prediction?

Your report must contain replicates of the listed tables. You can optionally include figures of the charts. Make sure all tables (and images) have captions.

Avoid hard coding numeric values in your Rmd file. Instead, use **inline code**. Likewise, do NOT use pandoc to generate the pdf. Use the the function `rednder()` in the package rmarkdown (via the command Rsript).

Makefile targets

Your `Makefile` should have the following Phony targets:

- `data:` will download the file `Advertising.csv` to the folder `data/`
- `tests:` will run the unit tests of your functions (i.e. executes the code in `test-that.R`)
- `eda:` will perform the exploratory data analysis (i.e. executes the code in `eda-script.R`)
- `regression:` will perform the series of regression analyses (i.e. executes the code in `regression-script.R`)
- `report:` will generate `report.pdf` (or `report.html`)
- `clean:` will delete the generated report (pdf and/or html)
- `all:` will be associated to phony targets `eda`, `regression`, and `report`

Based on the phony targets, you can add as many other targets as you consider convenient.

Grading

What you need to "turn in" is basically the **public** github repository of this project—don't use a private repo (keep in mind *Open Science*).

If you have minimal experience with github, we recommend you to create a new repository for this HW. In this way, you can practice creating remote repos, and adding them to your local repository. If you are already familiar with github, you can choose to handle this HW in one of your existing repositories.

In addition to checking whether you meet all the listed project requirements, for this and subsequent assignments, we will evaluate the following core competencies of your reports:

- Computation: Perform computations correctly.
- Analysis: Carry out analysis appropriate for data and context.
- Synthesis: Identify key features of the analysis, and interpret results.
- Visual presentation: communicate findings graphically clearly.
- Verbal: communicate findings in writing clearly, precisely and concisely.

Miscelanea

- Commit soon and often. We don't want to see that you have less than 10 commits.
- Write good commit messages. Ideally, commit messages should be unique (don't use the same message for two different commits). You should develop a naming style for different types of commits.
- Try experimenting with git branches.
- Use comments in your `Makefile`.
- Declare variables in your `Makefile`.
- Use automatic variables in your `Makefile`.
- Do NOT use absolute path names for the files (this breaks reproducibility).
- Don't underestimate README files: use them for your benefit; these are the files that you will look at months later when you come back to the project trying to remember the things you did (and why you did them).
- Avoid hard coding numeric values in your Rmd file that depend on the original data or on derived results from the analaysis. Instead, use **inline code**.
- You can discuss the various tasks of the HW with other students, but you must write your own code.