STAT 420: Homework 03

Fall 2020, D. Unger

Due: Tuesday, September 29 by 11:30 PM CT

Contents

Directions

Students are encouraged to work together on homework. However, sharing, copying or providing any part of a homework solution or code is an infraction of the University's rules on Academic Integrity. Any violation will be punished as severely as possible.

Final submissions must be uploaded to our Compass 2g site on the Homework page. No email, hardcopy, or late submissions will be accepted.

- Your assignment must be submitted through the submission link on **Compass 2g.** You are required to attach one `.zip` file, named `hw03_yourNetID.zip`, which contains:
  - Your RMarkdown file which should be saved as `hw03_yourNetID.Rmd`. For example `hw03_dunger.Rmd`.
  - The result of knitting your RMarkdown file as `hw03_yourNetID.html`. For example `hw03_dunger.html`.
  - Any raw data supplied by me for the assignment. For example `nutrition.csv`.
- Your resulting `.html` file will be considered a "report" which is the material that will determine the majority of your grade. Be sure to visibly include all `R` code and output that is relevant to answering the exercises. (You do not need to include irrelevant code you tried that resulted in error or did not answer the question correctly.)
- You are granted an unlimited number of submissions, but only the last submission *before* the deadline will be viewed and graded.
- If you use this `.Rmd` file as a template, be sure to remove the directions section. Consider removing `eval = FALSE` from any code chunks provided in the template, if you would like to run that code as part of your assignment.

- Your `.Rmd` file should be written such that, if it is placed in a folder with any data your are asked to import, it will knit properly without modification.
- Unless otherwise stated, you may use `R` for each of the exercises.
- Be sure to read each exercise carefully!
- Include your Name and NetID in the final document, not only in your filenames.

Assignment

Exercise 1 (Using `lm`)

For this exercise we will use the `faithful` dataset. This is a default dataset in `R`, so there is no need to load it. You should use `?faithful` to learn about the background of this dataset.

(a) Suppose we would like to predict the duration of an eruption of the Old Faithful geyser in Yellowstone National Park based on the waiting time before an eruption. Fit a simple linear model in `R` that accomplishes this task. Store the results in a variable called `faithful_model`. Output the result of calling `summary()` on `faithful_model`.

(b) Output only the estimated regression coefficients. Interpret β_0 and $\hat{\beta}_1$ in the *context of the problem*. Be aware that only one of those is an estimate.

(c) Use your model to predict the duration of an eruption based on a waiting time of **80** minutes. Do you feel confident in this prediction? Briefly explain.

(d) Use your model to predict the duration of an eruption based on a waiting time of **120** minutes. Do you feel confident in this prediction? Briefly explain.

(e) Calculate the RSS for this model.

(f) Create a scatterplot of the data and add the fitted regression line. Make sure your plot is well labeled and is somewhat visually appealing.

(g) Report the value of R^2 for the model. Do so directly. Do not simply copy and paste the value from the full output in the console after running `summary()` in part **(a)**.

Exercise 2 (Writing Functions)

This exercise is a continuation of Exercise 1.

(a) Write a function called `get_sd_est` that calculates an estimate of σ in one of two ways depending on input to the function. The function should take two arguments as input:

- `model_resid` - A vector of residual values from a fitted model.
- `mle` - A logical (`TRUE` / `FALSE`) variable which defaults to `FALSE`.

The function should return a single value:

- s_e if `mle` is set to `FALSE`.
- $\hat{\sigma}$ if `mle` is set to `TRUE`.

(b) Run the function `get_sd_est` on the residuals from the model in Exercise 1, with `mle` set to `FALSE`.

(c) Run the function `get_sd_est` on the residuals from the model in Exercise 1, with `mle` set to `TRUE`.

(d) To check your work, output `summary(faithful_model)$sigma`. It should match at least one of **(b)** or **(c)**.

Exercise 3 (Simulating SLR)

Consider the model

$$Y_i = 3 - 7x_i + \epsilon_i$$

with

$$\epsilon_i \sim N(\mu = 0, \sigma^2 = 4)$$

where $\beta_0 = 3$ and $\beta_1 = -7$.

Before answering the following parts, set a seed value equal to **your** birthday, as was done in the previous assignment.

```
birthday = 18760613
set.seed(birthday)
```

(a) Use `R` to simulate `n = 50` observations from the above model. For the remainder of this exercise, use the following "known" values of x.

```
x = runif(n = 50, 0, 10)
```

You may use the `sim_slr` function provided in the text. Store the data frame this function returns in a variable of your choice. Note that this function calls y `response` and x `predictor`.

(b) Fit a model to your simulated data. Report the estimated coefficients. Are they close to what you would expect? Briefly explain.

(c) Plot the data you simulated in part **(a)**. Add the regression line from part **(b)**. Hint: Keep the two commands in the same chunk, so `R` knows what plot to add the line to when knitting your `.Rmd` file.

(d) Use `R` to repeat the process of simulating `n = 50` observations from the above model 2000 times. Each time fit a SLR model to the data and store the value of $\hat{\beta}_1$ in a variable called `beta_hat_1`. Some hints:

- Use a `for` loop.
- Create `beta_hat_1` before writing the `for` loop. Make it a vector of length 2000 where each element is `0`.
- Inside the body of the `for` loop, simulate new y data each time. Use a variable to temporarily store this data together with the known x data as a data frame.
- After simulating the data, use `lm()` to fit a regression. Use a variable to temporarily store this output.
- Use the `coef()` function and `[]` to extract the correct estimated coefficient.
- Use `beta_hat_1[i]` to store in elements of `beta_hat_1`.
- See the notes on Distribution of a Sample Mean for some inspiration.

You can do this differently if you like. Use of these hints is not required.

(e) Report the mean and standard deviation of `beta_hat_1`. Do either of these look familiar?

(f) Plot a histogram of `beta_hat_1`. Comment on the shape of this histogram.

Exercise 4 (Be a Skeptic)

Consider the model

$$Y_i = 10 + 0x_i + \epsilon_i$$

with

$$\epsilon_i \sim N(\mu = 0, \sigma^2 = 1)$$

where $\beta_0 = 10$ and $\beta_1 = 0$.

Before answering the following parts, set a seed value equal to **your** birthday, as was done in the previous assignment.

```
birthday = 18760613
set.seed(birthday)
```

(a) Use `R` to repeat the process of simulating `n = 25` observations from the above model 1500 times. For the remainder of this exercise, use the following "known" values of x.

```
x = runif(n = 25, 0, 10)
```

Each time fit a SLR model to the data and store the value of $\hat{\beta}_1$ in a variable called `beta_hat_1`. You may use the `sim_slr` function provided in the text. Hint: Yes $\beta_1 = 0$.

(b) Plot a histogram of `beta_hat_1`. Comment on the shape of this histogram.

(c) Import the data in `skeptic.csv` and fit a SLR model. The variable names in `skeptic.csv` follow the same convention as those returned by `sim_slr()`. Extract the fitted coefficient for β_1.

(d) Re-plot the histogram from **(b)**. Now add a vertical red line at the value of $\hat{\beta}_1$ in part **(c)**. To do so, you'll need to use `abline(v = c, col = "red")` where `c` is your value.

(e) Your value of $\hat{\beta}_1$ in **(c)** should be positive. What proportion of the `beta_hat_1` values are larger than your $\hat{\beta}_1$? Return this proportion, as well as this proportion multiplied by `2`.

(f) Based on your histogram and part **(e)**, do you think the `skeptic.csv` data could have been generated by the model given above? Briefly explain.

Exercise 5 (Comparing Models)

For this exercise we will use the data stored in `goalies.csv`. It contains career data for all 716 players in the history of the National Hockey League to play goaltender through the 2014-2015 season. The variables in the dataset are:

- `Player` - NHL Player Name
- `First` - First year of NHL career
- `Last` - Last year of NHL career
- `GP` - Games Played
- `GS` - Games Started
- `W` - Wins
- `L` - Losses
- `TOL` - Ties/Overtime/Shootout Losses

- `GA` - Goals Against
- `SA` - Shots Against
- `SV` - Saves
- `SV_PCT` - Save Percentage
- `GAA` - Goals Against Average
- `SO` - Shutouts
- `MIN` - Minutes
- `G` - Goals (that the player recorded, not opponents)
- `A` - Assists (that the player recorded, not opponents)
- `PTS` - Points (that the player recorded, not opponents)
- `PIM` - Penalties in Minutes

For this exercise we will define the "Root Mean Square Error" of a model as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}.$$

(a) Fit a model with "wins"" as the response and "minutes" as the predictor. Calculate the RMSE of this model. Also provide a scatterplot with the fitted regression line.

(b) Fit a model with "wins"" as the response and "goals against" as the predictor. Calculate the RMSE of this model. Also provide a scatterplot with the fitted regression line.

(c) Fit a model with "wins"" as the response and "shutouts" as the predictor. Calculate the RMSE of this model. Also provide a scatterplot with the fitted regression line.

(d) Based on the previous three models, which of the three predictors used is most helpful for predicting wins? Briefly explain.