Week 2 - Homework

STAT 420, Summer 2017, Dalpiaz

Directions

- Be sure to remove this section if you use this `.Rmd` file as a template.
- You may leave the questions in your final document.

Exercise 1 (Using `lm`)

For this exercise we will use the `cats` dataset from the `MASS` package. You should use `?cats` to learn about the background of this dataset.

(a) Suppose we would like to understand the size of a cat's heart based on the body weight of a cat. Fit a simple linear model in `R` that accomplishes this task. Store the results in a variable called `cat_model`. Output the result of calling `summary()` on `cat_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 heart weight of a cat that weights **3.3** kg. Do you feel confident in this prediction? Briefly explain.

(d) Use your model to predict the heart weight of a cat that weights **1.5** kg. Do you feel confident in this prediction? Briefly explain.

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

(f) 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(cat_model)$sigma`. It should match at least one of **(b)** or **(c)**.

Exercise 3 (Simulating SLR)

Consider the model

$$Y_i = -4 + 2x_i + \epsilon_i$$

with

$$\epsilon_i \sim N(\mu = 0, \sigma^2 = 6.25)$$

where $\beta_0 = -4$ and $\beta_1 = 2$.

This exercise relies heavily on generating random observations. To make this reproducible we will set a seed for the randomization. Alter the following code to make `birthday` store your birthday in the format: `yyyymmdd`. For example, William Gosset, better known as *Student*, was born on June 13, 1876, so he would use:

```
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)** as well as the line for the true model. Hint: Keep all plotting commands in the same chunk.

(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:

- Consider 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 exercise.

```
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

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

(a) Fit three SLR models, each with "wins" as the reponse. For the predictor, use "minutes", "goals against", and "shutouts" respectively. For each, calculate RMSE and R^2. Arrange the results in a markdown table, with a row for each model. Suggestion: Create a data frame that stores the results, then investigate the `kable()` function from the `knitr` package.

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

Exercise 00 (SLR without Intercept)

This exercise will *not* be graded and is simply provided for your information. No credit will be given for the completion of this exercise. Give it a try now, and be sure to read the solutions later.

Sometimes it can be reasonable to assume that β_0 should be 0. That is, the line should pass through the point $(0,0)$. For example, if a car is traveling 0 miles per hour, its stopping distance should be 0! (Unlike what we saw in the book.)

We can simply define a model without an intercept,

$$Y_i = \beta x_i + \epsilon_i.$$

(a) In the **Least Squares Approach** section of the text you saw the calculus behind the derivation of the regression estimates, and then we performed the calculation for the `cars` dataset using `R`. Here you need to do, but not show, the derivation for the slope only model. You should then use that derivation of $\hat{\beta}$ to write a function that performs the calculation for the estimate you derived.

In summary, use the method of least squares to derive an estimate for β using data points (x_i, y_i) for $i = 1, 2, \ldots n$. Simply put, find the value of β to minimize the function

$$f(\beta) = \sum_{i=1}^{n}(y_i - \beta x_i)^2.$$

Then, write a function `get_beta_no_int` that takes input:

- `x` - A predictor variable
- `y` - A response variable

The function should then output the $\hat{\beta}$ you derived for a given set of data.

(b) Write your derivation in your `.Rmd` file using TeX. Or write your derivation by hand, scan or photograph your work, and insert it into the `.Rmd` as an image. See the RMarkdown documentation for working with images.

(c) Test your function on the `cats` data using body weight as `x` and heart weight as `y`. What is the estimate for β for this data?

(d) Check your work in `R`. The following syntax can be used to fit a model without an intercept:

```
lm(response ~ 0 + predictor, data = dataset)
```

Use this to fit a model to the `cat` data without an intercept. Output the coefficient of the fitted model. It should match your answer to **(c)**.