Q1. Save Answer

Q2. You fit the following linear model on a dataset of mammal species. Each row includes lifespan (in years), weight (in pounds), and diet (carnivore, omnivore, or herbivore).

$$\hat{\text{lifespan}} = 11.5 + 2.0 \cdot \text{weight} + 1.5 \cdot \text{carnivore} - 1.2 \cdot \text{omnivore}$$

Q2.1 Interpret an Indicator pt. 1

According to this model, if carnivore A weighs 5 pounds more than carnivore B, what is the difference in predicted lifespans?

- The coefficient for weight is 2, so all else equal, a carnivore weighing 5 more pounds than another carnivore is expected to live 5*2 = 10 years longer

Q2.2 Interpret an Indicator pt. 2

According to this model, if an herbivore and a carnivore weigh the same, what is the absolute value of the difference in predicted lifespans?

- Being a carnivore would add its coefficient of 1.5 years to that animal's lifespans, and with their weights being the same, the absolute value of the difference would be 1.5 years

Q2.3-Q2.6 You decide to change the regression to the following:

$$\hat{\text{lifespan}} = \beta_0 + \beta_1 \cdot \text{weight} + \beta_2 \cdot \text{carnivore} + \beta_3 \cdot \text{herbivore}$$

Q2.3 What would the resulting coefficient β_0 be? (hint: for any mammal you can think of, the predicted lifespans should be the same as the original model)

- Setting the two equations above equal to each other in terms of life span would yield:

  $11.5 + 2.0 \cdot \text{weight} + 1.5 \cdot \text{carnivore} - 1.2 \cdot \text{omnivore} = \beta_0 + \beta_1 \cdot \text{weight} + \beta_2 \cdot \text{carnivore} + \beta_3 \cdot \text{herbivore}$

- ... which we can also write as:

  $$11.5 + 2.0 \cdot \text{weight} + 1.5 \cdot \text{carnivore} - 1.2 \cdot \text{omnivore} + 0 \cdot \text{herbivore} =$$

  $$\beta_0 + \beta_1 \cdot \text{weight} + \beta_2 \cdot \text{carnivore} + 0 \cdot \text{omnivore} + \beta_3 \cdot \text{herbivore}$$

- Setting weight, herbivore, and carnivore to 0, while setting omnivore to 1, we would get:

  $$11.5 - 1.2 \cdot 1 = \beta_0 = 10.3$$

- Thus, β_0 is equal to 10.3

Q2.4 What would the resulting coefficient β_1 be?

- Since weight is unchanged between models, β_1 is also 2.0

Q2.5 What would the resulting coefficient β_2 be?

- Using the same process as previously, we get $\beta_2 = 2.7$ if we set carnivore to 1 everything else to 0:

  $$11.5 + 1.5 \cdot 1 = 10.3 + \beta_2 \cdot 1$$

Q2.6 What would the resulting coefficient β_3 be?

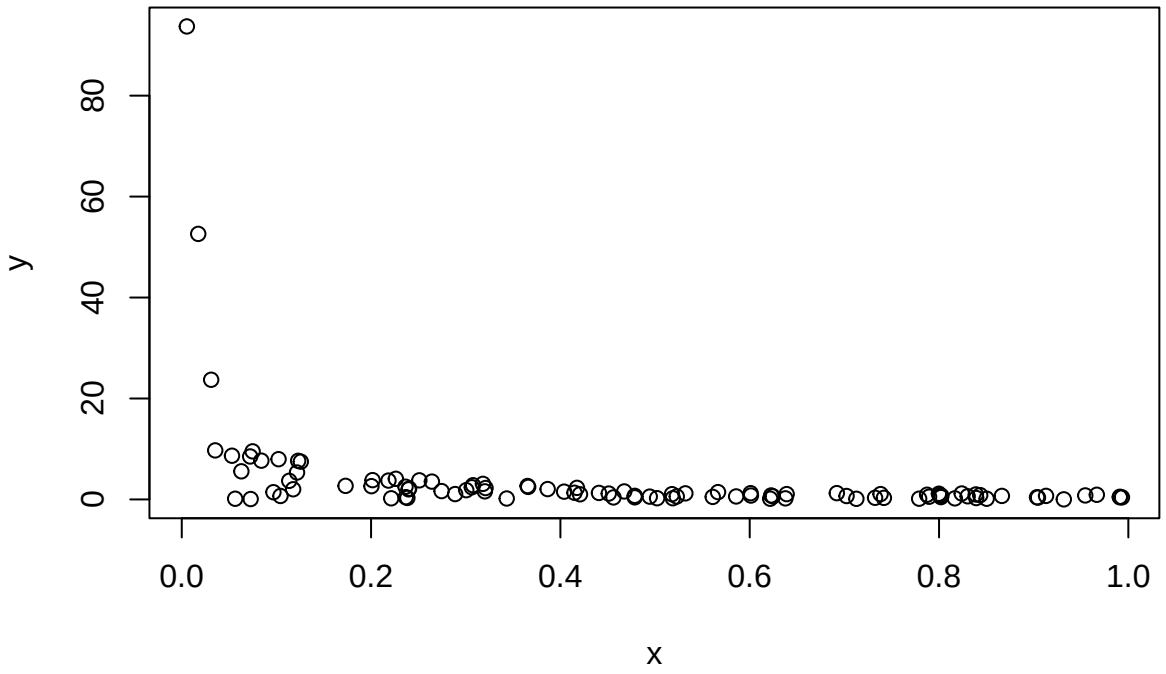
- Using the same process as previously, we get $\beta_3 = 1.2$ if we set herbivore to 1 everything else to 0:

  $$11.5 = 10.3 + \beta_3 \cdot 1$$

Q3 Regression Coefficients are Random Variables

The following function simulates an iid sample from a certain joint distribution.

```r
rmystery <- function(n){
  x = runif(n)
  y = runif(n, min=0, max = 1/x)
  data.frame(x=x,y=y)
}
plot(rmystery(100))
```



Q3.1 Set up an "experiment" for `rmystery`

Create a function, called experiment_m, which draws 100 datapoints from the mystery distribution above, fits a regression of y on x, then returns the slope coefficient.

Run your function a few times and notice that you get different values for the slope each time. Remind yourself that to a statistician, slope coefficients are random variables. We always pretend that data comes from a joint distribution, so if we could magically rewind time and get new data, we'd get new slope coefficients.

Use the visualization trick to approximate the sampling distribution of the slope. Run your experiment function 1000 times and plot a histogram of your result. What do you notice about the sampling distribution?

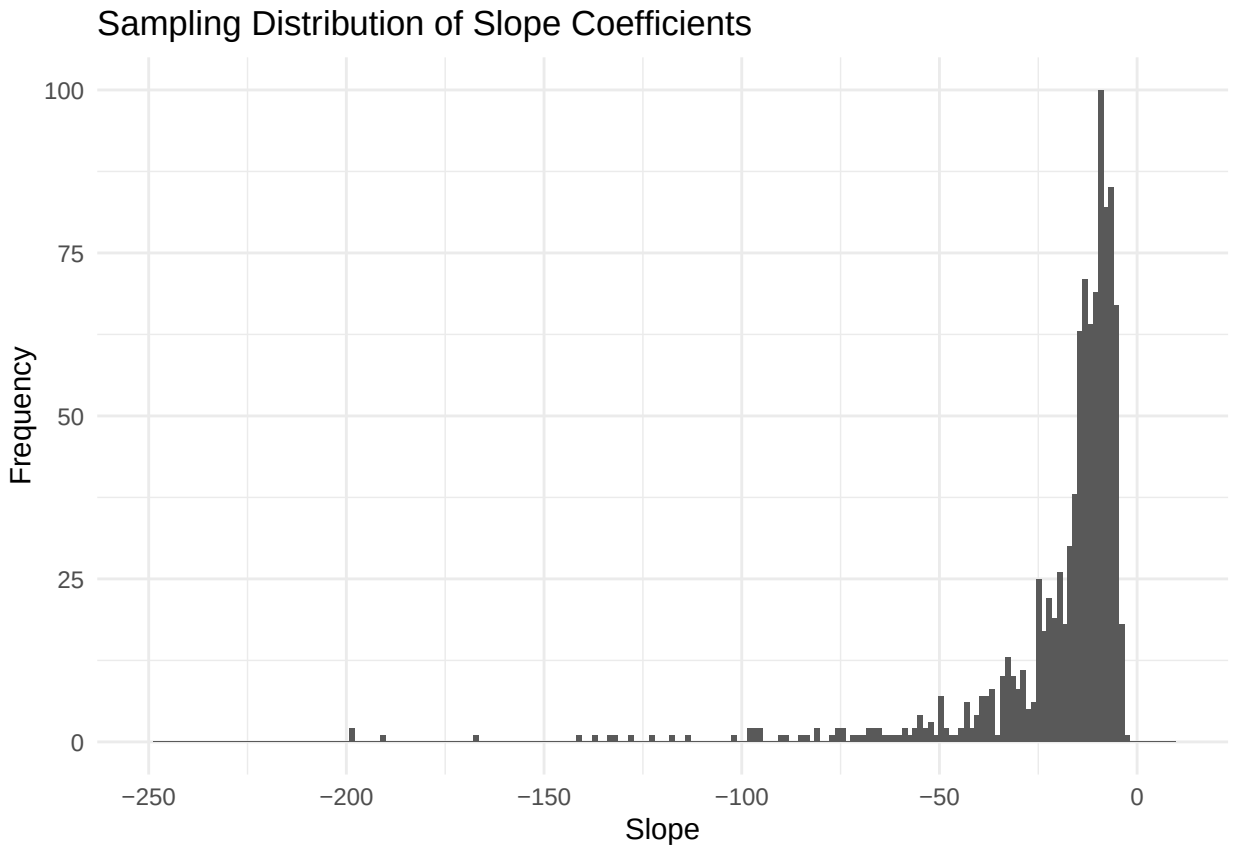
```r
experiment_m <- function(){
  model_a <- lm(y ~ x, data = rmystery(100))
  return(coef(model_a)['x'])
}

slope_data <- data.frame(slopes = replicate(1000, experiment_m()))

slope_data %>%
  ggplot(aes(x = slopes)) +
  geom_histogram(bins = 200) +
  xlim(-250, 10) +
  labs(title = "Sampling Distribution of Slope Coefficients", x = "Slope", y = "Frequency") +
  theme_minimal()
```

```
## Warning: Removed 16 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```



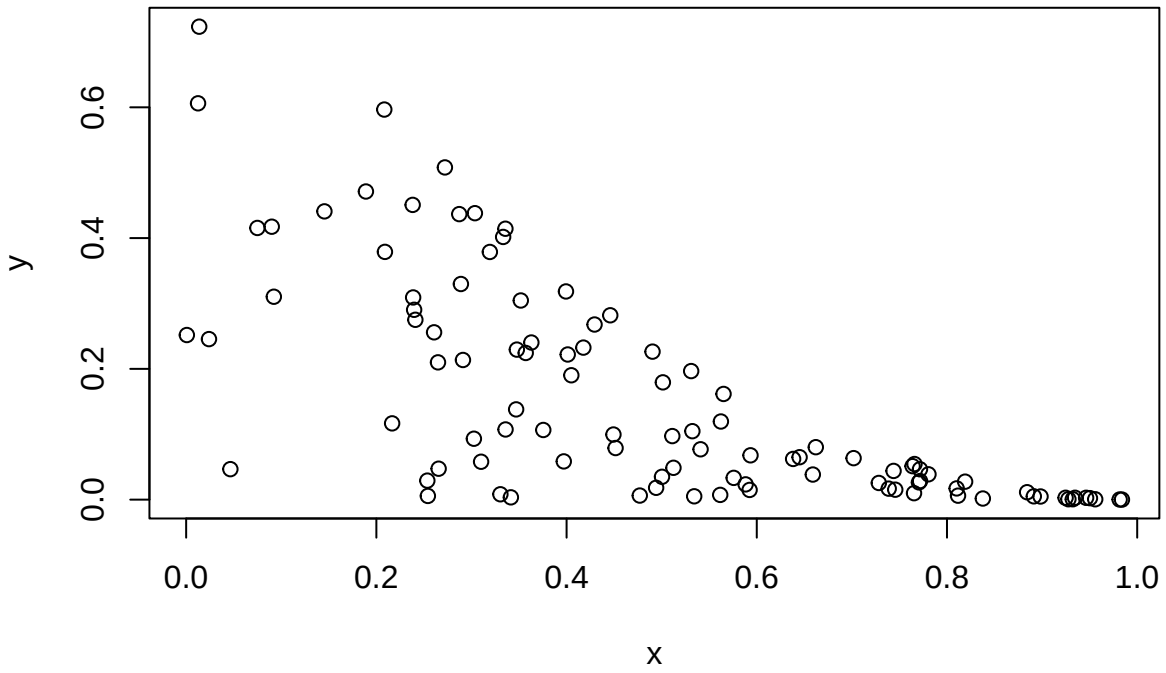
Thus, the sampling distribution looks extremely left-skewed.

Q3.2 Explain your results for `rmystery`

- The sampling distribution does not appear to be normal at all. Generating the histogram of the results shows that the sampling distribution is extremely left-skewed, as when x gets closer to 0, y will explode towards infinity. Evaluating the large sample assumptions of IID and unique BLP, it does appear that it is at least IID, as the model is independently picking from the same distribution for every draw. A unique BLP, however, does not exist due to the nature of the fat tail (leptokurtotic), which means the variance is unbounded and not finite.
- The large sample assumptions include IID and a unique BLP.
- A unique blp probably does not exist due to the nature of the fat tails (leptokurtotic)

Q3.3 You have a new function renigma as defined below.

```
renigma <- function(n){
  x = runif(n)
  y = runif(n, min=0, max = (1-x)^2)
  data.frame(x=x,y=y)
}
plot(renigma(100))
```



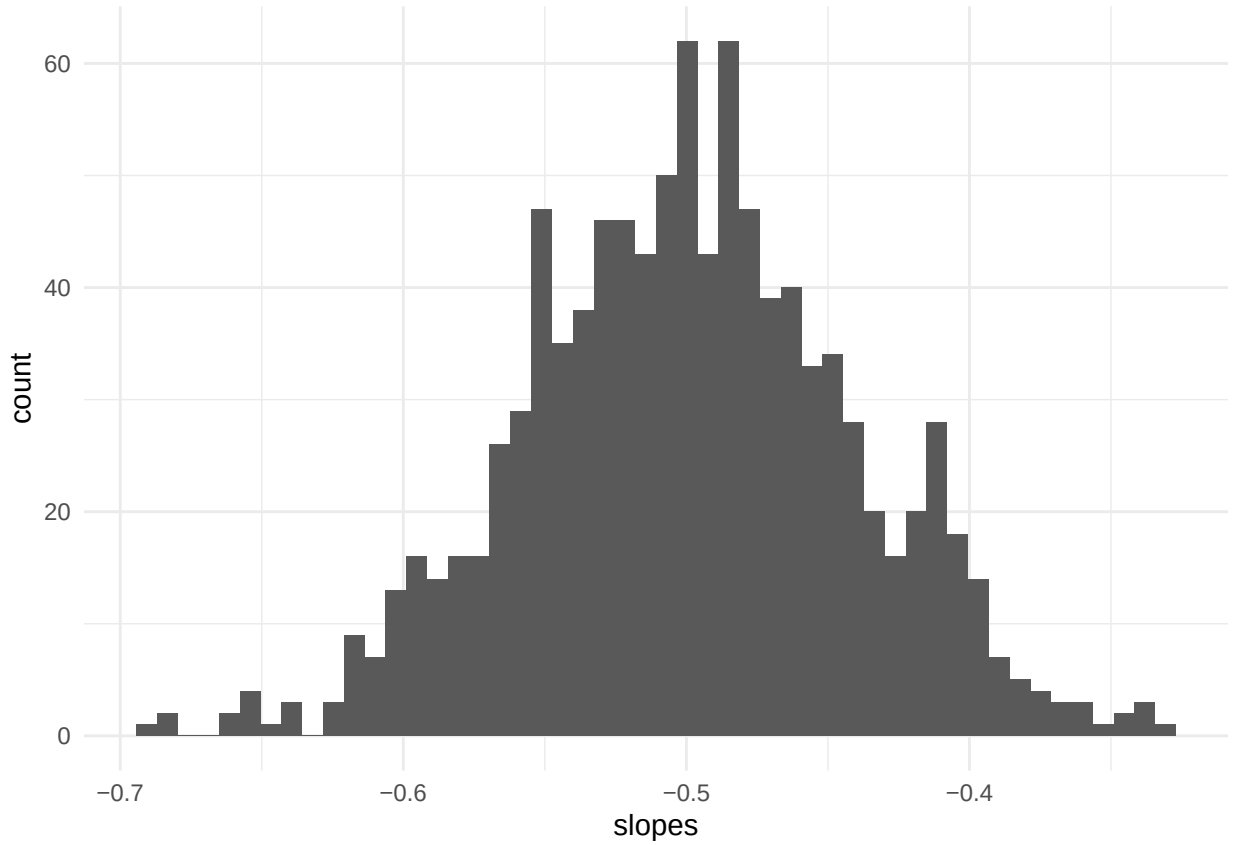
Create a function, called experiment_e, which draws 100 datapoints from the enigma distribution above, fits a regression of y on x, then returns the slope coefficient.

Run your experiment function 1000 times and plot a histogram of your result. What do you notice about the sampling distribution?

```
experiment_e <- function(){
  model_b <- lm(y ~ x, data = renigma(100))
  return(coef(model_b)['x'])
}

slope_data_2 <- data.frame(slopes = replicate(1000, experiment_e()))

slope_data_2 %>%
  ggplot(aes(x = slopes)) +
  geom_histogram(bins = 50)
```




```
  labs(title = "Sampling Distribution of Enigma Slope Coefficients", x = "Slope", y = "Frequency") +
  theme_minimal()
```

```
## NULL
```

Thus, this sampling distribution looks approximately normal.

Q3.4 Explain your results for `renigma`

- This sampling distribution appears to be relatively normal. Generating the histogram of results shows that the sampling distribution is roughly normal without the heavy tails that existed previously (as y will never shoot up to infinity). Evaluating the large sample assumptions of IID and unique BLP, it does appear that this is IID, as the model is independently picking from the same distribution for every draw. A unique BLP also likely exists because 1) the cov[Xi, Xj] and cov[Xi, Y] is finite, and 2) there is no perfect collinearity.