Assignment 6: Who busts the Mythbusters?

FirstName LastName

2022-07-12

Exercise 1

```
View(experimental_data)
```

1. By looking at three variables for this experiment, 'group' variable is the explanatory variable and 'yawn' is the response variable.
2. The value that has 'yes' in the response variable should be classified as success.

Exercise 2

Review the results table in the About the dataset section again. What quantity should we use or compute in order to build the null distribution? 1. We should use or compute "The average difference in fraction_yawned between the treatment and control groups" in order to build the null distribution. The reason is that when we compare with "yawn", it has different totals to compare which means control and treatment group have different range which makes hard to accurately compare. However, when we use "fraction_yawned", control and treatment have same totals to comapare which helps to bring more accurate result.

Exercise 3

```
specify(yawn ~ group, success = "yes")
```

Exercise 4

```
hypothesize(null = "independence")
```

1. We chose independence, because we have explanatory and response variables refered and do not have estimated number (point) to compare.

Exercise 5

```
generate(reps = 10000,  type = "permute")
```

If we want to run 10000 (ten thousand) simulations to create the null distribution for this dataset, what should I replace the placeholders with? Replace the placeholders in the answer file with their correct values. (Hint: what values did they use in the lecture videos?) - simulte x - certain probabilities oof happening does not exist - bootstrap x - comparing confidence intervals, relationship between different columns - does not exist - permute type o - generating null-hypothesis ## Exercise 6

```
calculate(stat = "diff in props",  order = c("control", "treatment"))
```

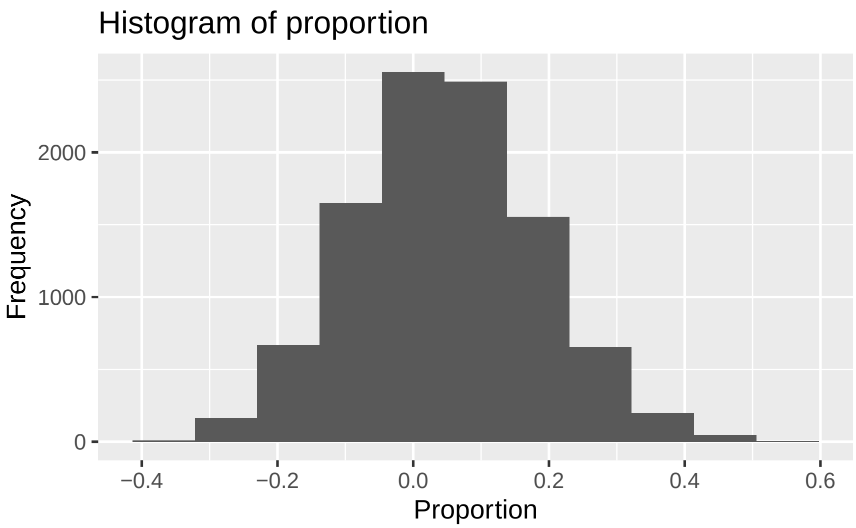
Based on your response to question 2 and the above information, replace the placeholders with their correct values. ## Exercise 7

```
yawn_null <- experimental_data %>%
  specify(yawn ~ group, success = "yes") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000,  type = "permute") %>%
  calculate(stat = "diff in props",  order = c("Control", "Treatment"))
```

Exercise 8

1. Histogram

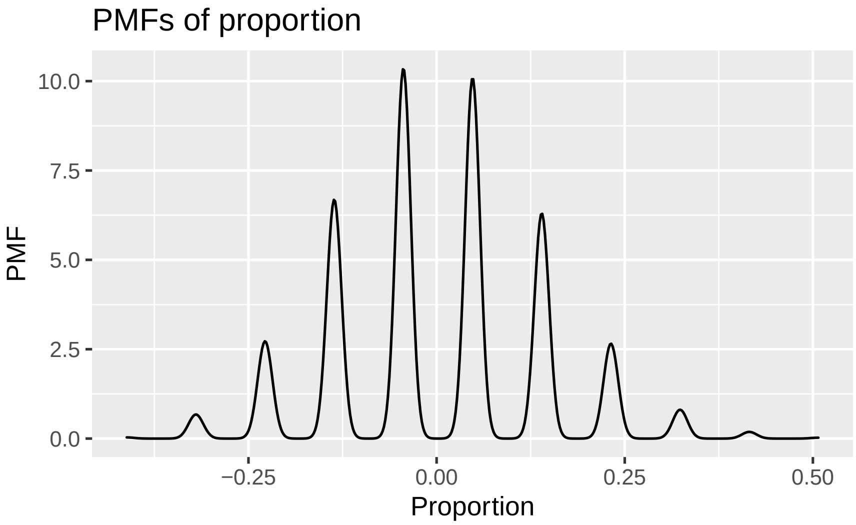
```
ggplot(data = yawn_null) +
  geom_histogram(
    mapping = aes(x = stat),
    bins = 11
  ) +
  labs(title = "Histogram of proportion",
    x = "Proportion",
    y = "Frequency"
  )
```



2. Density plot

```
ggplot(data = yawn_null) +
  geom_density(
    mapping = aes(x = stat),
    bins = 11
  ) +
  labs(title = "PMFs of proportion",
    x = "Proportion",
    y = "PMF"
  )
```

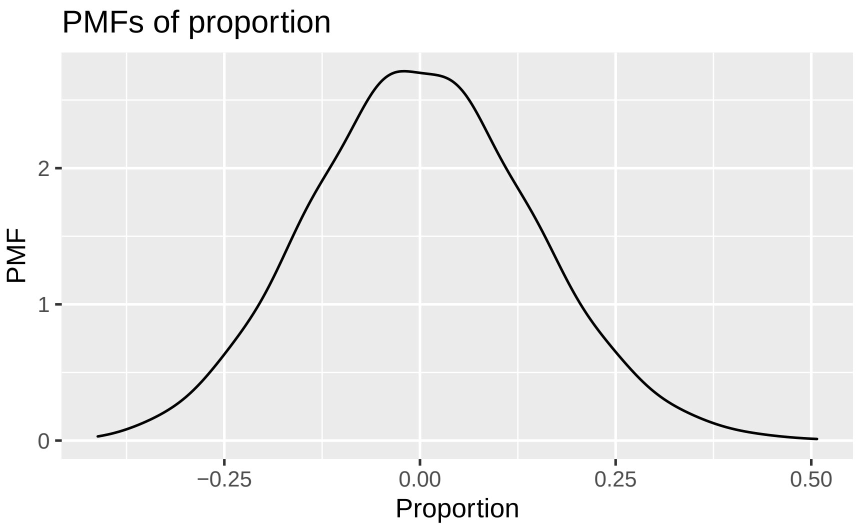
```
## Warning: Ignoring unknown parameters: bins
```



3. Do you see how the density plot is jagged? We can remove this by changing how the density curve is smoothed. Create a new density plot using your code from part (ii) but add the the adjust parameter to the geom_density() function as in this example:

```
ggplot(data = yawn_null) +
  geom_density(
    mapping = aes(x = stat, y= ..density..), adjust = 5,
    bins = 11
  ) +
  labs(title = "PMFs of proportion",
    x = "Proportion",
    y = "PMF"
  )
```

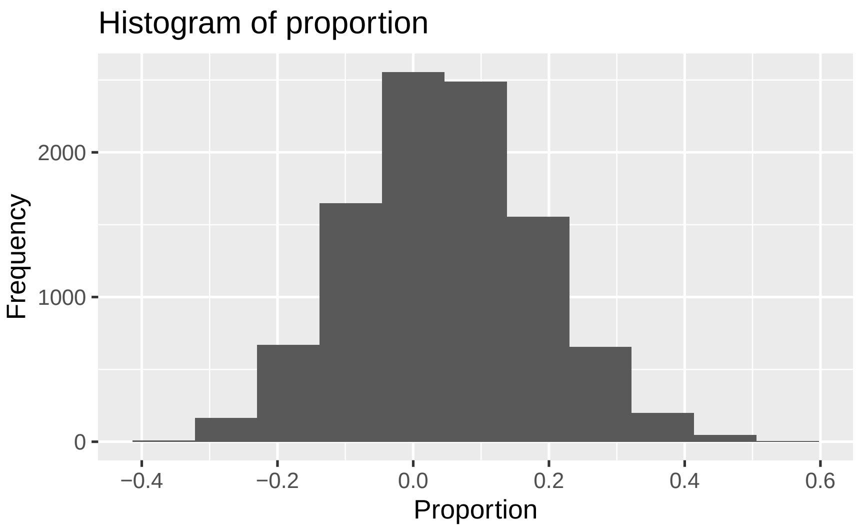
```
## Warning: Ignoring unknown parameters: bins
```



4. Where is the center of each distribution? Does this make sense given what you know about the data (i.e. would you expect the difference of proportions to be centered on this value)?

Exercise 9

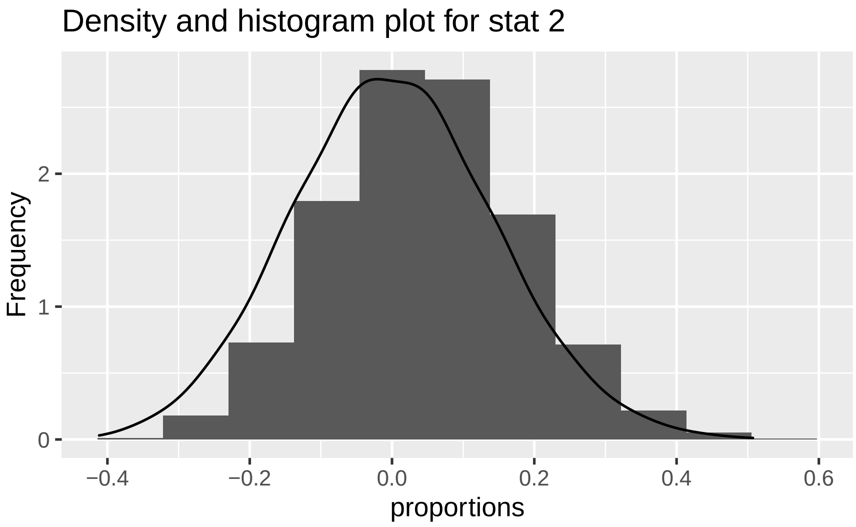
```
ggplot(data = yawn_null) +
  geom_histogram(
    mapping = aes(x = stat),
    bins = 11
  ) +
  labs(title = "Histogram of proportion",
    x = "Proportion",
    y = "Frequency"
  )
```



Problem) y-axis of PMF graph and histogram doesn't match.

Exercise 10

```
yawn_null %>%
ggplot +
geom_histogram(mapping = aes(x=stat, y=..density..), bins = 11) +
geom_density(mapping= aes(x=stat), adjust = 5) +
labs(title = "Density and histogram plot for stat 2", x= "proportions", y = "Frequency")
```



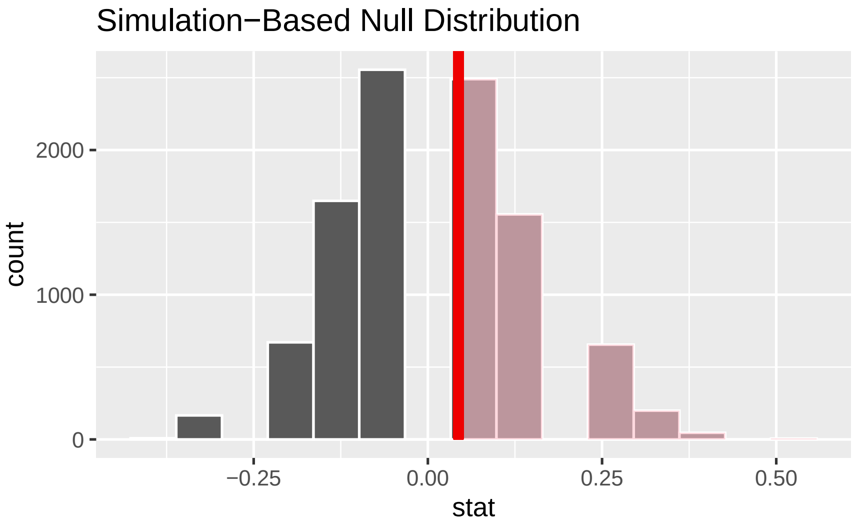
Exercise 11

```
yawn_obs_stat <- experimental_data %>%
specify(yawn ~ group, success = "yes") %>%
calculate(stat = "diff in props", order = c("Treatment", "Control"))
```

```
yawn_null %>%
get_p_value(obs_stat = yawn_obs_stat, direction = "left")
```

p_value
0.5049

```
yawn_null %>%
visualize() + shade_p_value(obs_stat = yawn_obs_stat, direction = "right")
```


Simulation−Based Null Distribution
count
stat