Section 001 CDS 101 Introduction to Computational and Data Sciences

Class 23: Inference and simulations II

April 17, 2018

General

Annoucements

- Reading for Thursday's class: *Nature* News Feature article, "Scientific method: Statistical errors" by R. Nuzzo
    - Slack responses for this are different from the standard procedure, you are responding to two prompts after you complete the reading
    - **Students that write a full and thoughtful response that addresses both prompts will receive both a question and an answer credit.** See the posted reading 14 assignment for details.
- Homework 4 to be posted today or tomorrow, due on Friday, April 27th by 11:59pm

`infer` review

Case study dataset: gender discrimination

- Last time, we reviewed the following dataset from a study on gender discrimination in hiring practices:

Case study dataset: gender discrimination

- Last time, we reviewed the following dataset from a study on gender discrimination in hiring practices:

```
applicants_data <- data_frame(
    sex = c(rep("Male", 24), rep("Female", 24)),
    outcome = c(
      rep("Promoted", 21),
      rep("Not Promoted", 3),
      rep("Promoted", 14),
      rep("Not Promoted", 10)))

experiment_result <- (21/24) - (14/24)
```

Case study dataset: gender discrimination

- Last time, we reviewed the following dataset from a study on gender discrimination in hiring practices:

```r
applicants_data <- data_frame(
    sex = c(rep("Male", 24), rep("Female", 24)),
    outcome = c(
      rep("Promoted", 21),
      rep("Not Promoted", 3),
      rep("Promoted", 14),
      rep("Not Promoted", 10)))

experiment_result <- (21/24) - (14/24)
```

- The result was that, of the 48 candidates reviewed, 29.2% more men than women were recommended for promotion, all else being equal.

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution
- We used the `infer` package to do this:

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution
- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution

- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution
- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

- In `specify(outcome ~ sex, success = "Promoted")`, the first part `outcome ~ sex` is a formula where the lefthand variable `outcome` is the response and the righthand variable `sex` is explanatory.

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution
- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution
- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

- In `hypothesize(null = "independence")`, we specify that we will simulate what will happen if `outcome` and `sex` were independent.

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution
- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution

- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

- In `generate(reps = 10000, type = "permute")`, we specify that we will run 10,000 simulations by permuting the `outcome` and `sex` columns

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution

- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

- In `generate(reps = 10000, type = "permute")`, we specify that we will run 10,000 simulations by permuting the `outcome` and `sex` columns

- To permute, we randomly shuffle the data in the `outcome` column, and then randomly shuffle the data in the `sex` column

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution

- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution

- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

- Using `calculate(stat = "diff in props", order = c("Male", "Female"))` means, after each simulation, compute:

$$\frac{\text{Promoted Men}}{\text{Total Men}} - \frac{\text{Promoted Men}}{\text{Total Men}}$$

Using `infer` to build the null distribution

- To use this dataset to test the hypothesis that "women were less likely than men to be hired", we needed to generate a null distribution

- We used the `infer` package to do this:

```
simulation_results <- applicants_data %>%
  specify(outcome ~ sex, success = "Promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 10000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Male", "Female"))
```

- Using `calculate(stat = "diff in props", order = c("Male", "Female"))` means, after each simulation, compute:

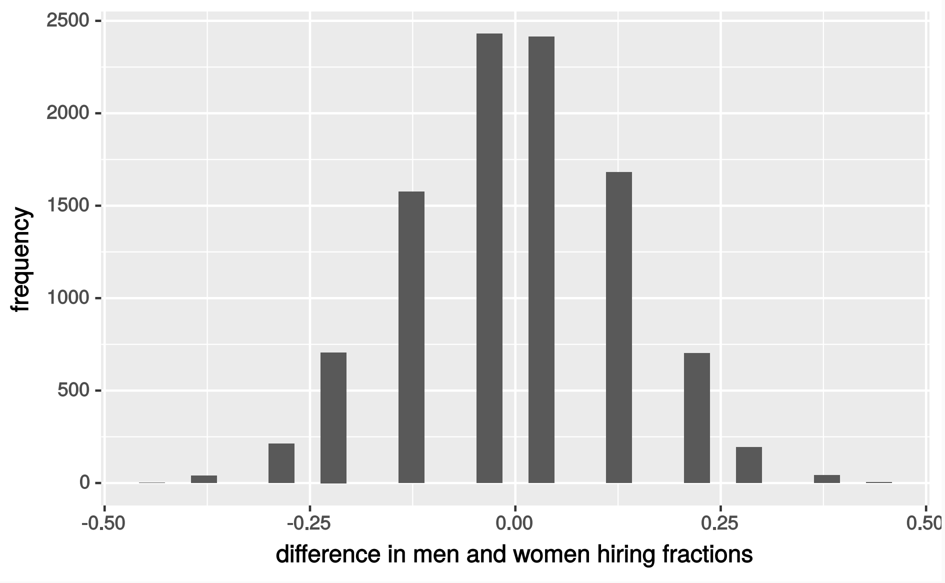
$$\frac{\text{Promoted Men}}{\text{Total Men}} - \frac{\text{Promoted Men}}{\text{Total Men}}$$

Note that this is exactly how `experiment_result` was calculated.

Null distribution

- After running the simulation, we obtain a null distribution:

```
simulation_results %>%
  ggplot() +
  geom_histogram(mapping = aes(x = stat)) +
  labs(
    x = "difference in men and women hiring fractions",
    y = "frequency")
```



One-sided hypothesis test

- To compute the p-value for a one-sided test, we need to figure out what fraction of the data is more extreme than the experimental result

One-sided hypothesis test

- To compute the p-value for a one-sided test, we need to figure out what fraction of the data is more extreme than the experimental result
- In other words, how much of the simulated data represents an outcome where the difference in the men and women hiring fractions is **larger** than 0.292?

One-sided hypothesis test

- To compute the *p*-value for a one-sided test, we need to figure out what fraction of the data is more extreme than the experimental result
- In other words, how much of the simulated data represents an outcome where the difference in the men and women hiring fractions is **larger** than 0.292?
- We can answer this by filtering the data to only keep the more extreme results, counting the remaining rows, and dividing by 10,000 (the number of simulations)

```
simulation_results %>%
  select(stat) %>%
  filter(stat > experiment_result) %>%
  count() %>%
  transmute(pvalue = n / 10000)
```

One-sided hypothesis test

- To compute the *p*-value for a one-sided test, we need to figure out what fraction of the data is more extreme than the experimental result
- In other words, how much of the simulated data represents an outcome where the difference in the men and women hiring fractions is **larger** than 0.292?
- We can answer this by filtering the data to only keep the more extreme results, counting the remaining rows, and dividing by 10,000 (the number of simulations)

```
simulation_results %>%
  select(stat) %>%
  filter(stat > experiment_result) %>%
  count() %>%
  transmute(pvalue = n / 10000)
```

pvalue
0.0046

Two-sided hypothesis test

- Computing the p-value for a two-sided test is similar to the one-sided test

Two-sided hypothesis test

- Computing the *p*-value for a two-sided test is similar to the one-sided test
- We still need to count the outcomes where the difference in the men and women hiring fractions is **larger** than 0.292, but we also need count the data in the opposite extreme, where the bias would be towards hiring more women than men

Two-sided hypothesis test

- Computing the p-value for a two-sided test is similar to the one-sided test
- We still need to count the outcomes where the difference in the men and women hiring fractions is **larger** than 0.292, but we also need count the data in the opposite extreme, where the bias would be towards hiring more women than men
- This corresponds to a difference in hiring fractions **less than** -0.292

Two-sided hypothesis test

- Computing the *p*-value for a two-sided test is similar to the one-sided test
- We still need to count the outcomes where the difference in the men and women hiring fractions is **larger** than 0.292, but we also need count the data in the opposite extreme, where the bias would be towards hiring more women than men
- This corresponds to a difference in hiring fractions **less than** -0.292
- As before, we can filter the data to just keep these extreme outcomes, then divide the remaining rows and divide by 10,000

```
simulation_results %>%
  select(stat) %>%
  filter(stat > experiment_result | stat < -experiment_result) %>%
  count() %>%
  transmute(pvalue = n / 10000)
```

Two-sided hypothesis test

- Computing the *p*-value for a two-sided test is similar to the one-sided test
- We still need to count the outcomes where the difference in the men and women hiring fractions is **larger** than 0.292, but we also need count the data in the opposite extreme, where the bias would be towards hiring more women than men
- This corresponds to a difference in hiring fractions **less than** -0.292
- As before, we can filter the data to just keep these extreme outcomes, then divide the remaining rows and divide by 10,000

```
simulation_results %>%
  select(stat) %>%
  filter(stat > experiment_result | stat < -experiment_result) %>%
  count() %>%
  transmute(pvalue = n / 10000)
```

pvalue
0.0088

`infer` activity

Mythbusters Statistics

An experiment conducted by the *MythBusters*, a science entertainment TV program that aired on the Discovery Channel, tested if a person can be subconsciously influenced into yawning if another person near them yawns.

Mythbusters Statistics

An experiment conducted by the *MythBusters*, a science entertainment TV program that aired on the Discovery Channel, tested if a person can be subconsciously influenced into yawning if another person near them yawns.

- 50 people were randomly assigned to two groups

Mythbusters Statistics

An experiment conducted by the *MythBusters*, a science entertainment TV program that aired on the Discovery Channel, tested if a person can be subconsciously influenced into yawning if another person near them yawns.

- 50 people were randomly assigned to two groups
  - **Treatment group**: 34 people where a person near them yawned
  - **Control group**: 16 people where there wasn't a person yawning near them

Mythbusters Statistics

An experiment conducted by the *MythBusters*, a science entertainment TV program that aired on the Discovery Channel, tested if a person can be subconsciously influenced into yawning if another person near them yawns.

- 50 people were randomly assigned to two groups
    - **Treatment group**: 34 people where a person near them yawned
    - **Control group**: 16 people where there wasn't a person yawning near them
- The full activity is available at the following Github Classroom link: https://classroom.github.com/a/o-KntOw5