Gov 50: 21. Hypothesis testing

Matthew Blackwell

Harvard University

Roadmap

1. The lady tasting tea
2. Hypothesis tests
3. Hypothesis testing using infer

1/ The lady tasting tea

The lady tasting tea

Your friend asks you to grab a tea with milk for her before meeting up and she says that she prefers tea poured before the milk. You stop by a local tea shop and ask for a tea with milk. When you bring it to her, she complains that it was prepared milk-first.

- You're skeptical that she can tell the difference, so you devise a test:
    - Prepare 8 cups of tea, 4 milk-first, 4 tea-first
    - Present cups to friend in a **random** order
    - Ask friend to pick which 4 of the 8 were milk-first.

Lady Tasting Tea data

Friend picks out all 4 milk-first cups correctly!

```
library(gov50data)
tea
```

```
## # A tibble: 8 x 2
##   truth      guess
##   <chr>      <chr>
## 1 tea-first  tea-first
## 2 milk-first milk-first
## 3 milk-first milk-first
## 4 tea-first  tea-first
## 5 tea-first  tea-first
## 6 milk-first milk-first
## 7 tea-first  tea-first
## 8 milk-first milk-first
```

Thought experiment

Could she have been guessing at random? What would guessing look like?

```
set.seed(02138)
one_guess <- tea |>
  mutate(random_guess = sample(guess))
one_guess
```

```
## # A tibble: 8 x 3
##   truth      guess      random_guess
##   <chr>      <chr>      <chr>
## 1 tea-first  tea-first  milk-first
## 2 milk-first milk-first tea-first
## 3 milk-first milk-first tea-first
## 4 tea-first  tea-first  milk-first
## 5 tea-first  tea-first  tea-first
## 6 milk-first milk-first milk-first
## 7 tea-first  tea-first  tea-first
## 8 milk-first milk-first milk-first
```

4 correct in this random guess!

Another guess

```
another_guess <- tea |>
  mutate(random_guess = sample(guess))
another_guess
```

```
## # A tibble: 8 x 3
##   truth      guess      random_guess
##   <chr>      <chr>      <chr>
## 1 tea-first  tea-first  tea-first
## 2 milk-first milk-first tea-first
## 3 milk-first milk-first milk-first
## 4 tea-first  tea-first  tea-first
## 5 tea-first  tea-first  milk-first
## 6 milk-first milk-first milk-first
## 7 tea-first  tea-first  tea-first
## 8 milk-first milk-first milk-first
```

6 correct in this random guess!

All possible guesses

We could enumerate all possible guesses. “Guessing” would mean choosing one of these at random:

```
##   Cup 1 Cup 2 Cup 3 Cup 4 Cup 5 Cup 6 Cup 7 Cup 8
## 1  milk  milk  milk  milk   tea   tea   tea   tea
## 2  milk  milk  milk   tea  milk   tea   tea   tea
## 3  milk  milk   tea  milk  milk   tea   tea   tea
## 4  milk   tea  milk  milk  milk   tea   tea   tea
## 5   tea  milk  milk  milk  milk   tea   tea   tea
## 6  milk  milk  milk   tea   tea  milk   tea   tea
```

[snip]

```
##    Cup 1 Cup 2 Cup 3 Cup 4 Cup 5 Cup 6 Cup 7 Cup 8
## 65   tea   tea   tea  milk  milk   tea  milk  milk
## 66  milk   tea   tea   tea   tea  milk  milk  milk
## 67   tea  milk   tea   tea   tea  milk  milk  milk
## 68   tea   tea  milk   tea   tea  milk  milk  milk
## 69   tea   tea   tea  milk   tea  milk  milk  milk
## 70   tea   tea   tea   tea  milk  milk  milk  milk
```

Statistical thought experiment

- Statistical thought experiment: how often would she get all 4 correct **if she were guessing randomly**?
  - Only one way to choose all 4 correct cups.
  - But 70 ways of choosing 4 cups among 8.
  - Choosing at random: picking each of these 70 with equal probability.
- Chances of guessing all 4 correct is $\frac{1}{70} \approx 0.014$ or 1.4%.
- → the guessing hypothesis might be implausible.
  - Impossible? No, because of random chance!

2/ Hypothesis tests

Statistical hypothesis testing

- Statistical hypothesis testing is a **thought experiment**.
  - Could our results just be due to random chance?
- What would the world look like **if we knew the truth**?
- Example 1:
  - An analyst claims that 20% of Boston households are in poverty.
  - You take a sample of 900 households and find that 23% of the sample is under the poverty line.
  - Should you conclude that the analyst is wrong?
- Example 2:
  - Trump won 47.5% of the vote in the 2020 election.
  - Last YouGov poll of 1,363 likely voters said 44% planned to vote for Trump.
  - Could the difference between the poll and the outcome be just due to random chance?

Null and alternative hypothesis

- **Null hypothesis**: Some statement about the population parameters.
  - "Devil's advocate" position $\rightsquigarrow$ assumes what you seek to prove wrong.
  - Usually that an observed difference is due to chance.
  - Ex: poll drawn from the same population as all voters.
  - Denoted H_0
- **Alternative hypothesis**: The statement we hope or suspect is true instead of H_0.
  - It is the opposite of the null hypothesis.
  - An observed difference is real, not just due to chance.
  - Ex: polling for Trump is systematically wrong.
  - Denoted H_1 or H_a
- **Probabilistic** proof by contradiction: try to "disprove" the null.

Hypothesis testing example

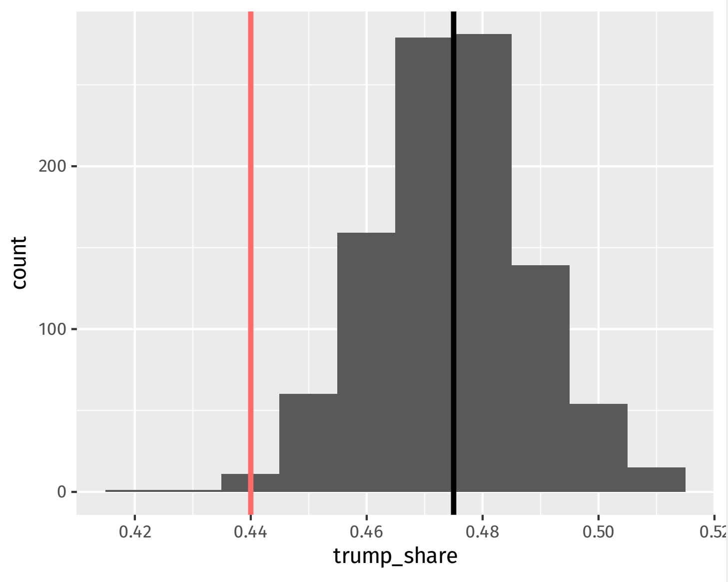
- Are we polling the same population as the actual voters?
    - If so, how likely are we to see polling error this big by chance?
- What is the parameter we want to learn about?
    - True population mean of the surveys, p.
    - Null hypothesis: $H_0 : p = 0.475$ (surveys drawing from same population)
    - Alternative hypothesis: $H_1 : p \neq 0.475$
- Data: poll has $\overline{X} = 0.44$ with $n = 1363$.

Statistical thought experiment

- If the null were true, what should the distribution of the data be?
  - X_i is 1 if respondent i will vote for Trump.
  - Under null, X_i is a coin flip with probability $p = 0.475$ of landing on "Trump".
  - $X_1 + X_2 + \cdots + X_n$ is the number in sample that will vote for Trump.
- We can simulate sums of coin flips using a function called `rbinom()`
- Compare the distribution of proportions under the null to the observed proportion.

```
null_dist <- tibble(
  trump_share = rbinom(n = 1000, size = 1363, prob = 0.475) / 1363
)
ggplot(null_dist, aes(x = trump_share)) +
  geom_histogram(binwidth = 0.01) +
  geom_vline(xintercept = 0.44, color = "indianred1", size = 1.25) +
  geom_vline(xintercept = 0.475, size = 1.25)
```

Simulations of the reference distribution



p-value

p-value

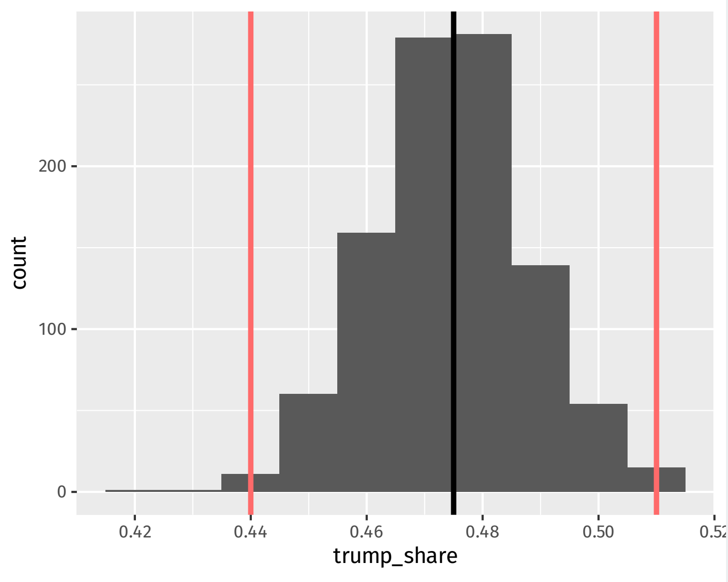
The **p-value** is the probability of observing data as or more extreme as our data if the null hypothesis is true.

- If the null is true, how often would we expect polling errors this big?
    - Smaller p-value $\rightsquigarrow$ stronger evidence against the null
    - **NOT** the probability that the null is true!
- p-values are usually **two-sided**:
    - Observed error of 0.44 - 0.475 = -0.035 under the null.
    - p-value is probability of sample proportions being less than 0.44 **plus**
    - Probability of sample proportions being greater than 0.475 + 0.035 = 0.51.

```
mean(null_dist$trump_share < 0.44) + mean(null_dist$trump_share > 0.51)
```

```
## [1] 0.01
```

Two-sided p-value



One-sided tests

- Sometimes our hypothesis is directional.
  - We only consider evidence against the null from one direction.
- Null: our polls are from the same population as actual voters
  - $H_0 : p = 0.475$
- **One-sided alternative**: polls underestimate Trump support.
  - $H_1 : p < 0.475$
- Makes the p-value one-sided:
  - What's the probability of a random sample underestimating Trump support by as much as we see in the sample?
  - Always smaller than a two-sided p-value.

```
mean(null_dist$trump_share < 0.44)
```

```
## [1] 0.005
```

Rejecting the null

- Tests usually end with a decision to reject the null or not.
- Choose a threshold below which you'll reject the null.
    - **Test level** α: the threshold for a test.
    - Decision rule: "reject the null if the p-value is below α"
    - Otherwise "fail to reject" or "retain", not "accept the null"
- Common (arbitrary) thresholds:
    - $p \geq 0.1$ "not statistically significant"
    - $p < 0.05$ "statistically significant"
    - $p < 0.01$ "highly significant"

Testing errors

- A p-value of 0.05 says that data this extreme would only happen in 5% of repeated samples if the null were true.
  - $\rightsquigarrow$ 5% of the time we'll reject the null when it is actually true.
- Test errors:

	H_0 True	H_0 False
Retain H_0	Awesome!	Type II error
Reject H_0	Type I error	Good stuff!

- Type I error because it's the worst
  - "Convicting" an innocent null hypothesis
- Type II error less serious
  - Missed out on an awesome finding

3/ Hypothesis testing using infer

GSS data from `infer`

```r
library(infer)
gss
```

```
## # A tibble: 500 x 11
##     year   age sex    college   partyid hompop hours income
##    <dbl> <dbl> <fct>  <fct>     <fct>    <dbl> <dbl> <ord>
##  1  2014    36 male   degree    ind          3    50 $25000~
##  2  1994    34 female no degree rep          4    31 $20000~
##  3  1998    24 male   degree    ind          1    40 $25000~
##  4  1996    42 male   no degree ind          4    40 $25000~
##  5  1994    31 male   degree    rep          2    40 $25000~
##  6  1996    32 female no degree rep          4    53 $25000~
##  7  1990    48 female no degree dem          2    32 $25000~
##  8  2016    36 female degree    ind          1    20 $25000~
##  9  2000    30 female degree    rep          5    40 $25000~
## 10  1998    33 female no degree dem          2    40 $15000~
## # i 490 more rows
## # i 3 more variables: class <fct>, finrela <fct>,
## #   weight <dbl>
```

What is the average hours worked?

`dplyr` way:

```
gss |>
  summarize(mean(hours))
```

```
## # A tibble: 1 x 1
##   `mean(hours)`
##           <dbl>
## 1          41.4
```

`infer` way:

```
observed_mean <- gss |>
  specify(response = hours) |>
  calculate(stat = "mean")
observed_mean
```

```
## Response: hours (numeric)
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  41.4
```

Hypothesis test

Could we get a mean this different from 40 hours if that was the true population average of hours worked?

Null and alternative:

$$H_0 : \text{population average hours} = 40$$
$$H_1 : \text{population average hours} \neq 40$$

How do we perform this test using infer? The **bootstrap!**

Specifying the hypotheses

```
gss |>
  specify(response = hours) |>
  hypothesize(null = "point", mu = 40)
```

```
## Response: hours (numeric)
## Null Hypothesis: point
## # A tibble: 500 x 1
##    hours
##    <dbl>
##  1    50
##  2    31
##  3    40
##  4    40
##  5    40
##  6    53
##  7    32
##  8    20
##  9    40
## 10    40
## # i 490 more rows
```

Generating the null distribution

We can use the bootstrap to determine how much variation there will be around 40 in the null distribution.

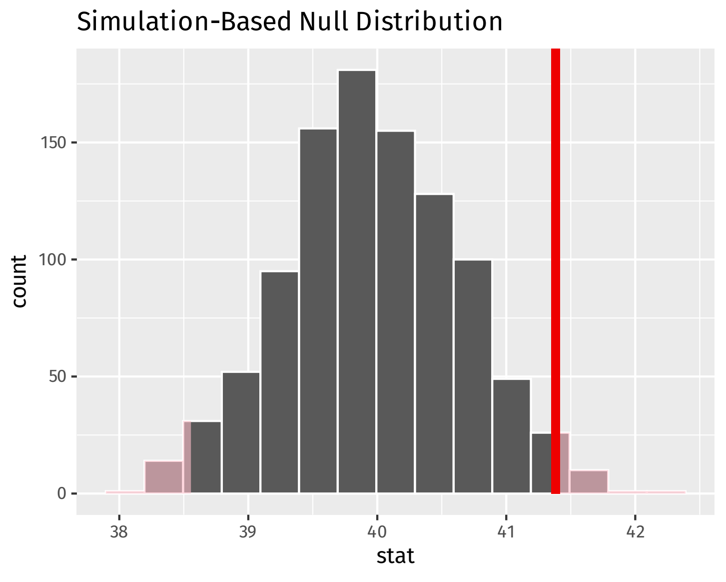
```
null_dist <- gss |>
  specify(response = hours) |>
  hypothesize(null = "point", mu = 40) |>
  generate(reps = 1000, type = "bootstrap") |>
  calculate(stat = "mean")
null_dist
```

```
## Response: hours (numeric)
## Null Hypothesis: point
## # A tibble: 1,000 x 2
##    replicate  stat
##        <int> <dbl>
##  1         1  40.3
##  2         2  39.8
##  3         3  40.0
##  4         4  39.2
##  5         5  40.3
##  6         6  40.2
##  7         7  40.4
```

Visualizing the p-value

We can visualize our bootstrapped null distribution and the p-value as a shaded region:

```
null_dist |>
  visualize() +
  shade_p_value(observed_mean,
                direction = "two-sided")
```


Simulation-Based Null Distribution
count
150
100
50
0
38
39
40
41
42
stat