

GVPT622 Lab Session

Hypothesis Testing

Ryan Bookstein

October 30, 2026

University of Maryland

- ▶ Load the packages we'll use today and set the seed:

```
library(readxl)
library(tidyverse)
library(skimr)
library(janitor)
library(infer)
library(scales)

set.seed(1234)
```

- ▶ Last lab: Were women members more likely than not to be Democrats ($p = 0.5$)?

- ▶ Last lab: Were women members more likely than not to be Democrats ($p = 0.5$)?
- ▶ Today: comparing two groups. Is the difference between them real, or could random chance have produced it?

- ▶ Last lab: Were women members more likely than not to be Democrats ($p = 0.5$)?
- ▶ Today: comparing two groups. Is the difference between them real, or could random chance have produced it?
- ▶ Today:

- ▶ Last lab: Were women members more likely than not to be Democrats ($p = 0.5$)?
- ▶ Today: comparing two groups. Is the difference between them real, or could random chance have produced it?
- ▶ Today:
 - ▷ “There is only one test”: a simulation-based difference of means

- ▶ Last lab: Were women members more likely than not to be Democrats ($p = 0.5$)?
- ▶ Today: comparing two groups. Is the difference between them real, or could random chance have produced it?
- ▶ Today:
 - ▷ “There is only one test”: a simulation-based difference of means
 - ▷ Two-sample t -tests (equal variances and Welch)

- ▶ Last lab: Were women members more likely than not to be Democrats ($p = 0.5$)?
- ▶ Today: comparing two groups. Is the difference between them real, or could random chance have produced it?
- ▶ Today:
 - ▷ “There is only one test”: a simulation-based difference of means
 - ▷ Two-sample t -tests (equal variances and Welch)
 - ▷ Difference of proportions

The Question

Does the Majority Make More Law?

- ▶ The majority party controls the committees, the Rules Committee, and the floor schedule

Does the Majority Make More Law?

- ▶ The majority party controls the committees, the Rules Committee, and the floor schedule
- ▶ So, on average, do majority-party members get more of their bills signed into law than minority-party members?

Does the Majority Make More Law?

- ▶ The majority party controls the committees, the Rules Committee, and the floor schedule
- ▶ So, on average, do majority-party members get more of their bills signed into law than minority-party members?
- ▶ From the probability lab: $P(\text{at least one law} \mid \text{majority}) \approx 55\%$. Today we ask whether a gap like that could be noise

Does the Majority Make More Law?

- ▶ The majority party controls the committees, the Rules Committee, and the floor schedule
- ▶ So, on average, do majority-party members get more of their bills signed into law than minority-party members?
- ▶ From the probability lab: $P(\text{at least one law} \mid \text{majority}) \approx 55\%$. Today we ask whether a gap like that could be noise
- ▶ From lecture: treat the two groups as two populations with means μ_1 and μ_2 . Is μ_1 different from μ_2 ?

Does the Majority Make More Law?

- ▶ The majority party controls the committees, the Rules Committee, and the floor schedule
- ▶ So, on average, do majority-party members get more of their bills signed into law than minority-party members?
- ▶ From the probability lab: $P(\text{at least one law} \mid \text{majority}) \approx 55\%$. Today we ask whether a gap like that could be noise
- ▶ From lecture: treat the two groups as two populations with means μ_1 and μ_2 . Is μ_1 different from μ_2 ?
- ▶ Data: the CEL House data, 118th Congress (2023–2024), 448 members

Preparing the Data

```
cel <- read_excel("CELHouse93to118.xlsx", guess_max = 20000)

house118 <- cel |>
  rename(
    congress = "Congress number",
    name = "Legislator name, as given in THOMAS",
    majority = "1 = majority party member",
    laws = "Total bills becoming law (inc in other leg vehicles)"
  ) |>
  filter(congress == 118) |>
  mutate(status = if_else(majority == 1, "Majority", "Minority"),
         status = factor(status, levels = c("Majority", "Minority"))) |>
  select(name, status, laws)
```

- **status** is a factor with Majority first. The order matters later: R subtracts the second level from the first

```
house118
```

```
# A tibble: 448 x 3
  name          status    laws
  <chr>         <fct>    <dbl>
1 Adams, Alma   Minority    0
2 Aderholt, Robert Majority    2
3 Aguilar, Pete Minority    2
4 Alford, Mark  Majority    0
5 Allen, Rick   Majority    2
# i 443 more rows
```

The Shape of the Outcome

```
skim(house118, laws)
```

```
-- Variable type: numeric -----  
  skim_variable n_missing complete_rate  mean    sd p0 p25 p50 p75 p100  
1 laws          0              1 0.737 1.24  0  0  0  1  12
```

- ▶ The average member got 0.74 bills into law. The median got **zero**

The Shape of the Outcome

```
skim(house118, laws)
```

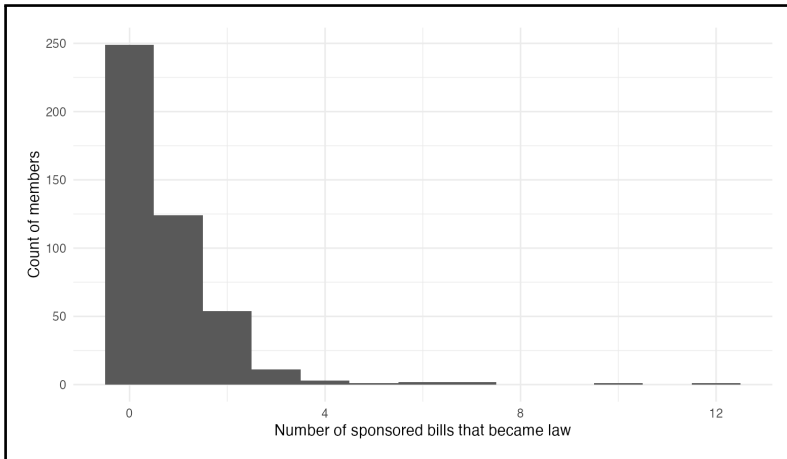
```
-- Variable type: numeric -----  
  skim_variable n_missing complete_rate  mean    sd p0 p25 p50 p75 p100  
1 laws                0              1 0.737 1.24  0  0  0  1  12
```

- ▶ The average member got 0.74 bills into law. The median got **zero**
- ▶ One member got 12. This is a heavily right-skewed count

The Shape of the Outcome

```
ggplot(house118, aes(x = laws)) +  
  geom_histogram(binwidth = 1) +  
  theme_minimal() +  
  labs(x = "Number of sponsored bills that became law",  
       y = "Count of members")
```

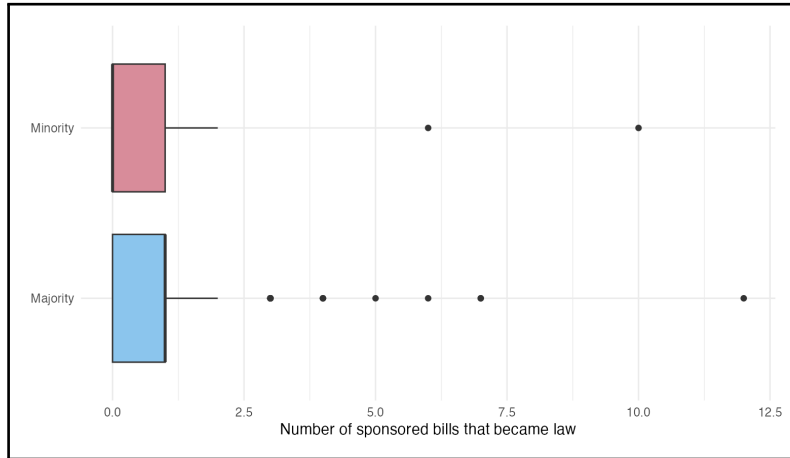
The Shape of the Outcome



249 of 448 members got no bills into law. A handful got five or more.

```
ggplot(house118, aes(x = laws, y = status, fill = status)) +  
  geom_boxplot() +  
  theme_minimal() +  
  theme(legend.position = "none") +  
  scale_fill_discrete(type = c("#8bc5ed", "#D88C9A")) +  
  labs(x = "Number of sponsored bills that became law",  
       y = NULL)
```

Majority vs. Minority



Both medians are 0, so the boxes look alike. The difference is in the tail.

Group Averages

```
laws_avg <- house118 |>
  group_by(status) |>
  summarise(n_members = n(),
            avg_laws = mean(laws)) |>
  mutate(diff_means = avg_laws - lead(avg_laws))
```

laws_avg

```
# A tibble: 2 x 4
  status    n_members avg_laws diff_means
<fct>      <int>    <dbl>    <dbl>
1 Majority     228    0.961    0.456
2 Minority     220    0.505     NA
```

- Majority members averaged 0.96 laws; minority members 0.50

Group Averages

```
laws_avg <- house118 |>
  group_by(status) |>
  summarise(n_members = n(),
            avg_laws = mean(laws)) |>
  mutate(diff_means = avg_laws - lead(avg_laws))
```

laws_avg

```
# A tibble: 2 x 4
  status    n_members avg_laws diff_means
<fct>      <int>    <dbl>    <dbl>
1 Majority     228    0.961    0.456
2 Minority     220    0.505     NA
```

- ▶ Majority members averaged 0.96 laws; minority members 0.50
- ▶ Is that gap real, or random error?

There Is Only One Test

1. Compute the test statistic in your sample

Allen Downey's Five Steps

1. Compute the test statistic in your sample
2. Define the null hypothesis: a model of a world with no effect

Allen Downey's Five Steps

1. Compute the test statistic in your sample
2. Define the null hypothesis: a model of a world with no effect
3. Use that model to simulate many samples from the null world

Allen Downey's Five Steps

1. Compute the test statistic in your sample
2. Define the null hypothesis: a model of a world with no effect
3. Use that model to simulate many samples from the null world
4. Compute the test statistic in each simulated sample

Allen Downey's Five Steps

1. Compute the test statistic in your sample
2. Define the null hypothesis: a model of a world with no effect
3. Use that model to simulate many samples from the null world
4. Compute the test statistic in each simulated sample
5. Find the proportion of simulated statistics at least as extreme as yours. If it's tiny, your sample would be very unlikely in a world with no effect

Step 1: The Test Statistic

```
diff_means <- laws_avg |>  
  drop_na(diff_means) |>  
  pull(diff_means)
```

```
diff_means
```

```
[1] 0.4559809
```

- ▶ On average, majority members got 0.46 more bills into law than minority members

Step 1: The Test Statistic

```
diff_means <- laws_avg |>
  drop_na(diff_means) |>
  pull(diff_means)

diff_means
```

```
[1] 0.4559809
```

- ▶ On average, majority members got 0.46 more bills into law than minority members
- ▶ Majority – Minority, so a positive number means the majority made more law

Step 2: The Null Hypothesis

- ▶ H_a : majority members make more law, $\mu_{maj} > \mu_{min}$

Step 2: The Null Hypothesis

- ▶ H_a : majority members make more law, $\mu_{maj} > \mu_{min}$
- ▶ H_0 : no difference in means, $\mu_{maj} - \mu_{min} = 0$

Step 2: The Null Hypothesis

- ▶ H_a : majority members make more law, $\mu_{maj} > \mu_{min}$
- ▶ H_0 : no difference in means, $\mu_{maj} - \mu_{min} = 0$
- ▶ Three ways to model a world with no difference:

Step 2: The Null Hypothesis

- ▶ H_a : majority members make more law, $\mu_{maj} > \mu_{min}$
- ▶ H_0 : no difference in means, $\mu_{maj} - \mu_{min} = 0$
- ▶ Three ways to model a world with no difference:
 - ▶ Parametric: pool the groups, find the pooled center and spread, and draw from that distribution

Step 2: The Null Hypothesis

- ▶ H_a : majority members make more law, $\mu_{maj} > \mu_{min}$
- ▶ H_0 : no difference in means, $\mu_{maj} - \mu_{min} = 0$
- ▶ Three ways to model a world with no difference:
 - ▷ Parametric: pool the groups, find the pooled center and spread, and draw from that distribution
 - ▷ Resampling: pool the groups and draw from them with replacement

Step 2: The Null Hypothesis

- ▶ H_a : majority members make more law, $\mu_{maj} > \mu_{min}$
- ▶ H_0 : no difference in means, $\mu_{maj} - \mu_{min} = 0$
- ▶ Three ways to model a world with no difference:
 - ▷ Parametric: pool the groups, find the pooled center and spread, and draw from that distribution
 - ▷ Resampling: pool the groups and draw from them with replacement
 - ▷ Permutation: shuffle the group labels at random (resampling without replacement)

Step 2: The Null Hypothesis

- ▶ H_a : majority members make more law, $\mu_{maj} > \mu_{min}$
- ▶ H_0 : no difference in means, $\mu_{maj} - \mu_{min} = 0$
- ▶ Three ways to model a world with no difference:
 - ▷ Parametric: pool the groups, find the pooled center and spread, and draw from that distribution
 - ▷ Resampling: pool the groups and draw from them with replacement
 - ▷ Permutation: shuffle the group labels at random (resampling without replacement)
- ▶ We'll use permutation. The t -test makes a parametric choice for you, behind the scenes

Step 3: Simulate the Null World

```
laws_null <- house118 |>
  specify(laws ~ status) |>
  hypothesize(null = "independence") |>
  generate(reps = 5000, type = "permute")
```

- ▶ Each replicate keeps every member's **laws** but shuffles who is labeled Majority and Minority

Step 3: Simulate the Null World

```
laws_null <- house118 |>  
  specify(laws ~ status) |>  
  hypothesize(null = "independence") |>  
  generate(reps = 5000, type = "permute")
```

- ▶ Each replicate keeps every member's **laws** but shuffles who is labeled Majority and Minority
- ▶ If party status didn't matter, any shuffle is as plausible as the real labels

Step 3: Simulate the Null World

```
laws_null
```

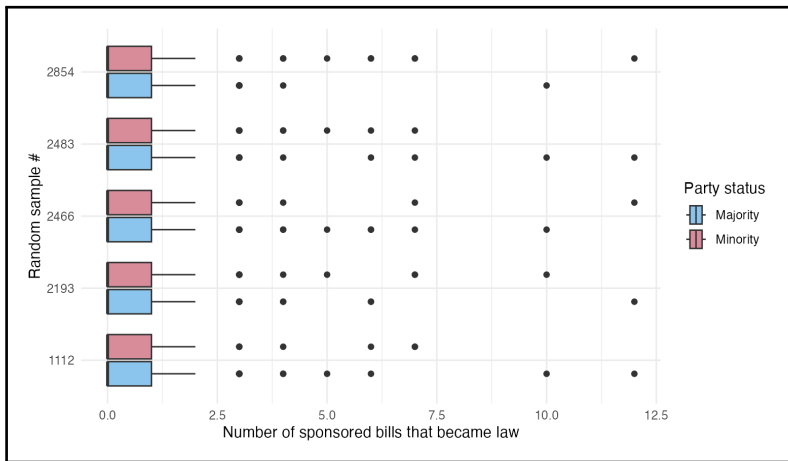
```
Response: laws (numeric)
Explanatory: status (factor)
Null Hypothesis: indepe...
# A tibble: 2,240,000 x 3
# Groups:   replicate [5,000]
   laws status    replicate
   <dbl> <fct>         <int>
1      0 Minority         1
2      0 Majority         1
3      0 Minority         1
4      0 Majority         1
# i 2,239,996 more rows
```

448 members $\times$ 5,000 replicates = 2,240,000 rows.

```
random_5 <- sample(1:5000, 5)

laws_null |>
  filter(replicate %in% random_5) |>
  ggplot(aes(x = laws, y = factor(replicate), fill = status)) +
  geom_boxplot() +
  theme_minimal() +
  labs(x = "Number of sponsored bills that became law",
       y = "Random sample #",
       fill = "Party status") +
  scale_fill_discrete(type = c("#8bc5ed", "#D88C9A"))
```

Five Null Worlds



Within each null world the groups differ a little, but only by chance: who drew the 10- and 12-law members moves around.

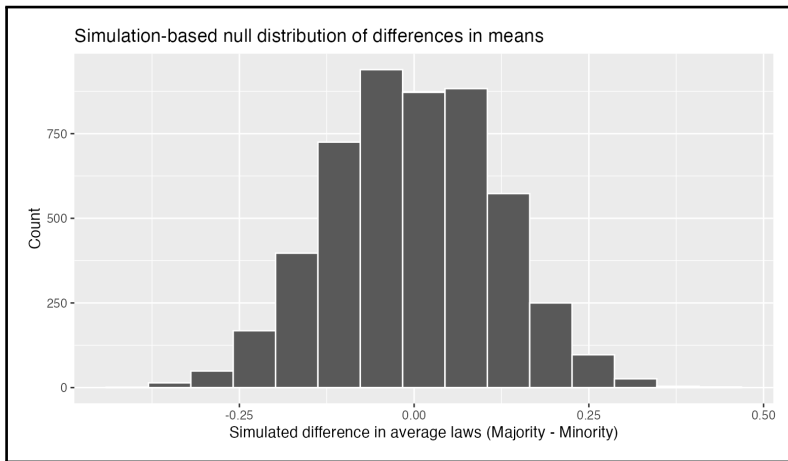
Step 4: The Statistic in Each Null World

```
laws_diffs_null <- laws_null |>
  calculate("diff in means", order = c("Majority", "Minority"))

visualize(laws_diffs_null) +
  labs(x = "Simulated difference in average laws (Majority - Minority)",
       y = "Count",
       title = "Simulation-based null distribution of differences in
               means")
```

► **order** sets which group is subtracted: Majority – Minority, matching step 1

Step 4: The Statistic in Each Null World



5,000 differences from worlds where party status doesn't matter. Centered on 0; the only thing creating them is random noise.

Step 5: Where Do Null Worlds Land?

```
null_interval <- get_confidence_interval(laws_diffs_null)
null_interval
```

```
# A tibble: 1 x 2
  lower_ci upper_ci
  <dbl>    <dbl>
1  -0.232    0.233
```

- ▶ The middle 95% of null-world differences fall between -0.23 and 0.23 laws

Step 5: Where Do Null Worlds Land?

```
null_interval <- get_confidence_interval(laws_diffs_null)
null_interval
```

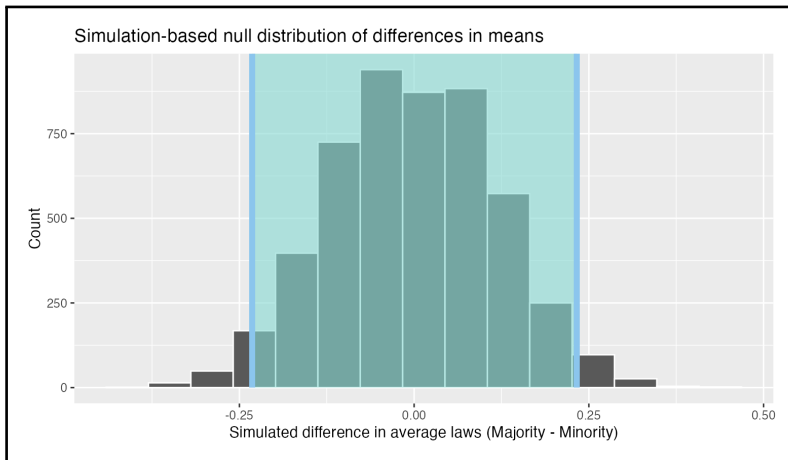
```
# A tibble: 1 x 2
  lower_ci upper_ci
  <dbl>    <dbl>
1  -0.232    0.233
```

- ▶ The middle 95% of null-world differences fall between -0.23 and 0.23 laws
- ▶ This is the interval for the null distribution, not for the true difference. We'll call it the 95% null interval

Step 5: Where Do Null Worlds Land?

```
visualize(laws_diffs_null) +  
  shade_confidence_interval(null_interval, color = "#8bc5ed", fill =  
    "#85d9d2") +  
  labs(x = "Simulated difference in average laws (Majority - Minority)",  
        y = "Count",  
        title = "Simulation-based null distribution of differences in  
means")
```

Step 5: Where Do Null Worlds Land?



The shaded band is the 95% null interval: where 95% of differences land when party status doesn't matter.

Where Do Those Numbers Come From?

```
quantile(laws_diffs_null$stat, probs = c(0.025, 0.975))
```

2.5%	97.5%
-0.2317384	0.2326954

- ▶ The 2.5th and 97.5th percentiles of the 5,000 simulated differences. No new formula: just find where 95% of them fall

Where Do Those Numbers Come From?

```
quantile(laws_diffs_null$stat, probs = c(0.025, 0.975))
```

2.5%	97.5%
-0.2317384	0.2326954

- ▶ The 2.5th and 97.5th percentiles of the 5,000 simulated differences. No new formula: just find where 95% of them fall
- ▶ That's a two-tailed interval: 125 simulated differences below -0.23 and 125 above 0.23

A Directional Hypothesis

```
quantile(laws_diffs_null$stat, probs = 0.95)
```

```
95%  
0.1880383
```

- ▶ Our H_a is directional: the majority makes more law

A Directional Hypothesis

```
quantile(laws_diffs_null$stat, probs = 0.95)
```

```
95%  
0.1880383
```

- ▶ Our H_a is directional: the majority makes more law
- ▶ So only big positive differences count against H_0 . All 5% goes in the upper tail

A Directional Hypothesis

```
quantile(laws_diffs_null$stat, probs = 0.95)
```

```
95%  
0.1880383
```

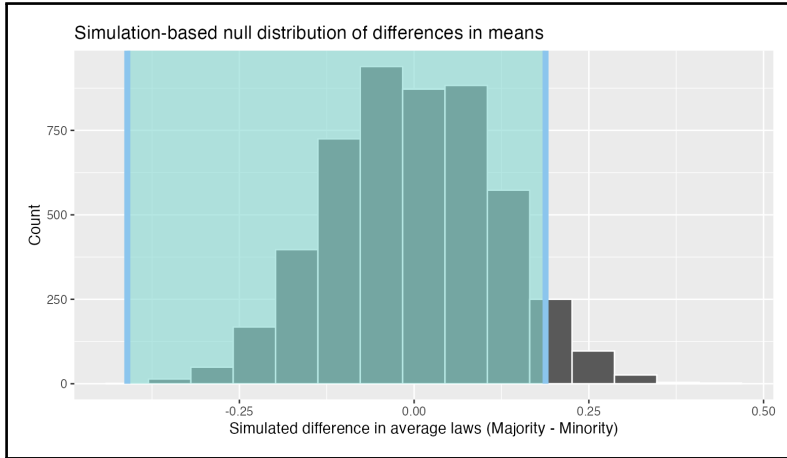
- ▶ Our H_a is directional: the majority makes more law
- ▶ So only big positive differences count against H_0 . All 5% goes in the upper tail
- ▶ The cutoff drops from 0.23 to 0.19: a one-tailed test rejects sooner, but only in the direction you predicted

A Directional Hypothesis

```
visualize(laws_diffs_null) +  
  shade_confidence_interval(quantile(laws_diffs_null$stat, probs = c(0,  
    0.95)),  
                             color = "#8bc5ed", fill = "#85d9d2") +  
  labs(x = "Simulated difference in average laws (Majority - Minority)",  
        y = "Count",  
        title = "Simulation-based null distribution of differences in  
        means")
```

`probs = c(0, 0.95)` shades from the minimum up to the 95th percentile, leaving the top 5% unshaded.

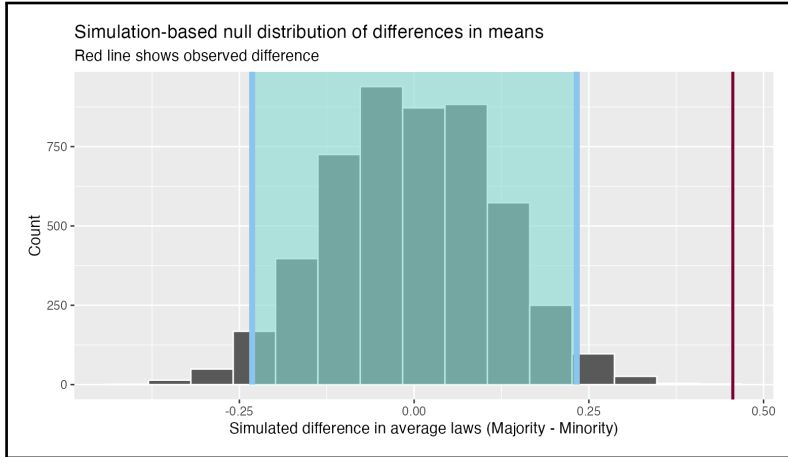
A Directional Hypothesis



One-tailed: only differences above 0.19 would lead us to reject H_0 .

```
visualize(laws_diffs_null) +  
  shade_confidence_interval(null_interval, color = "#8bc5ed", fill =  
    "#85d9d2") +  
  geom_vline(xintercept = diff_means, linewidth = 1, color = "#77002c")  
  +  
  labs(x = "Simulated difference in average laws (Majority - Minority)",  
    y = "Count",  
    title = "Simulation-based null distribution of differences in  
    means",  
    subtitle = "Red line shows observed difference")
```

Adding Our Sample



Our observed difference (0.46, red) is past every one of the 5,000 null-world differences.

The p-value

```
laws_diffs_null |>  
  get_p_value(obs_stat = diff_means, direction = "greater") |>  
  mutate(p_value_clean = pvalue(p_value))
```

```
# A tibble: 1 x 2  
  p_value p_value_clean  
    <dbl> <chr>  
1      0 <0.001
```

- ▶ **direction = "greater"**: the share of null differences ≥ 0.456 . (“less” and “two-sided” also work)

The p-value

```
laws_diffs_null |>
  get_p_value(obs_stat = diff_means, direction = "greater") |>
  mutate(p_value_clean = pvalue(p_value))
```

```
# A tibble: 1 x 2
  p_value p_value_clean
  <dbl> <chr>
1      0 <0.001
```

- ▶ **direction = "greater"**: the share of null differences ≥ 0.456 . (“less” and “two-sided” also work)
- ▶ None of the 5,000 got there. That isn't $p = 0$, it's $p < 1/5,000$. `infer` warns you about this; `pvalue()` from **scales** reports it as <0.001

The p-value

```
laws_diffs_null |>
  get_p_value(obs_stat = diff_means, direction = "greater") |>
  mutate(p_value_clean = pvalue(p_value))
```

```
# A tibble: 1 x 2
  p_value p_value_clean
  <dbl> <chr>
1      0 <0.001
```

- ▶ `direction = "greater"`: the share of null differences ≥ 0.456 . (“less” and “two-sided” also work)
- ▶ None of the 5,000 got there. That isn't $p = 0$, it's $p < 1/5,000$. `infer` warns you about this; `pvalue()` from `scales` reports it as `<0.001`
- ▶ We reject the null of no difference

Two-Sample t-Tests

- ▶ The theory-based route: how many standard errors apart are the two group means?

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

- ▶ The theory-based route: how many standard errors apart are the two group means?

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

- ▶ From lecture: the variance of a difference is the sum of the two variances. The denominator is the standard error of the difference

- ▶ The theory-based route: how many standard errors apart are the two group means?

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

- ▶ From lecture: the variance of a difference is the sum of the two variances. The denominator is the standard error of the difference
- ▶ Large $|t| \rightarrow$ a difference that's big relative to the noise $\rightarrow$ small p-value

- ▶ The theory-based route: how many standard errors apart are the two group means?

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

- ▶ From lecture: the variance of a difference is the sum of the two variances. The denominator is the standard error of the difference
- ▶ Large $|t| \rightarrow$ a difference that's big relative to the noise $\rightarrow$ small p-value
- ▶ The null model is buried in the formula: under H_0 , t follows a t -distribution with some degrees of freedom

Equal Variances (As in the Notes)

```
t_test_laws <- t.test(laws ~ status, var.equal = T, data = house118)
t_test_laws
```

Two Sample t-test

data: laws by status

t = 3.9673, df = 446, p-value = 8.466e-05

alternative hypothesis: true difference in means between group Majority
and group Minority is not equal to 0

95 percent confidence interval:

0.2301025 0.6818592

sample estimates:

mean in group Majority	mean in group Minority
0.9605263	0.5045455

`var.equal = T` pools the variances (not quite the last slide's formula). $df = n_1 + n_2 - 2 = 446$.

Welch's t-Test (R's Default)

```
t.test(laws ~ status, data = house118)
```

Welch Two Sample t-test

data: laws by status

t = 3.9897, df = 413.62, p-value = 7.825e-05

alternative hypothesis: true difference in means between group Majority
and group Minority is not equal to 0

95 percent confidence interval:

0.2313200 0.6806418

sample estimates:

mean in group Majority mean in group Minority

0.9605263

0.5045455

Welch uses the formula on the last slide and doesn't assume equal spreads (here, SDs of 1.39 vs. 1.00).

Software estimates df; by hand, lecture's conservative rule uses the smaller of $n_1 - 1$ and $n_2 - 1 = 219$.

- ▶ $t \approx 4$: the means are about four standard errors apart

- ▶ $t \approx 4$: the means are about four standard errors apart
- ▶ $p \approx 0.00008$ is two-sided ("not equal to 0"). For our directional H_a , halve it, or add `alternative = "greater"`

- ▶ $t \approx 4$: the means are about four standard errors apart
- ▶ $p \approx 0.00008$ is two-sided (“not equal to 0”). For our directional H_a , halve it, or add `alternative = "greater"`
- ▶ 95% CI for $\mu_{maj} - \mu_{min}$: 0.23 to 0.68 laws. This one is about the true difference, and it excludes 0

- ▶ $t \approx 4$: the means are about four standard errors apart
- ▶ $p \approx 0.00008$ is two-sided ("not equal to 0"). For our directional H_a , halve it, or add `alternative = "greater"`
- ▶ 95% CI for $\mu_{maj} - \mu_{min}$: 0.23 to 0.68 laws. This one is about the true difference, and it excludes 0
- ▶ Same answer as the simulation: reject H_0 . The two t -tests barely differ because the groups are large and similar in size

Difference of Proportions

- The probability lab's question: did a member get **at least one** bill into law?

```
house118 <- house118 |>
  mutate(any_law = if_else(laws > 0, "Yes", "No"))

tabyl(house118, status, any_law) |>
  adorn_percentages() |>
  adorn_pct_formatting(digits = 1) |>
  adorn_ns()
```

status	No	Yes
Majority	45.2% (103)	54.8% (125)
Minority	66.4% (146)	33.6% (74)

Same Test, New Statistic

```
diff_props <- house118 |>
  specify(any_low ~ status, success = "Yes") |>
  calculate("diff in props", order = c("Majority", "Minority"))

diff_props
```

```
# A tibble: 1 x 1
  stat
  <dbl>
1 0.212
```

(Output trimmed to the table.) `success = "Yes"` tells infer which outcome to count. 54.8% – 33.6% = 21.2 points.

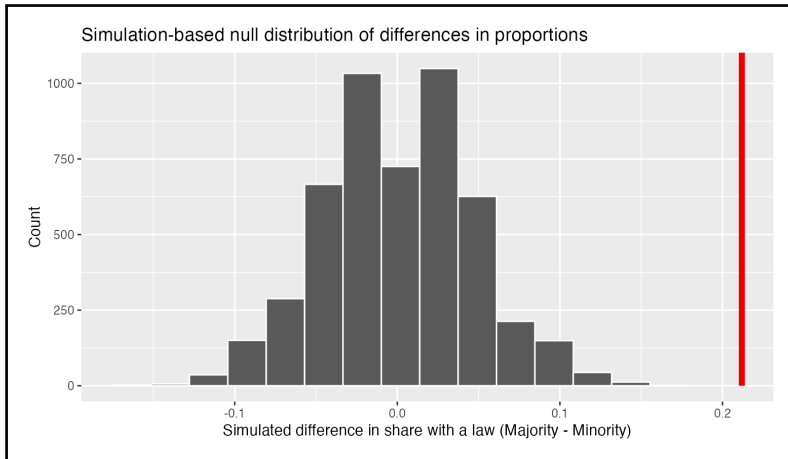
Same Five Steps

```
props_null <- house118 |>
  specify(any_law ~ status, success = "Yes") |>
  hypothesize(null = "independence") |>
  generate(reps = 5000, type = "permute") |>
  calculate("diff in props", order = c("Majority", "Minority"))

visualize(props_null) +
  shade_p_value(obs_stat = diff_props, direction = "greater") +
  labs(x = "Simulated difference in share with a law (Majority -
    Minority)",
    y = "Count",
    title = "Simulation-based null distribution of differences in
    proportions")
```

`shade_p_value()` draws the observed statistic and shades the tail it's tested against.

Same Five Steps



The red line is our observed 21.2-point gap. No null world got there, so there is no tail to shade.

```
props_null |>
  get_p_value(obs_stat = diff_props, direction = "greater") |>
  mutate(p_value_clean = pvalue(p_value))
```

```
# A tibble: 1 x 2
  p_value p_value_clean
  <dbl> <chr>
1      0 <0.001
```

- ▶ Again $p < 1/5,000$, reported as **<0.001**. Reject H_0 : a 21-point gap would be very unlikely by chance alone

```
law_counts <- house118 |>
  group_by(status) |>
  summarise(n_members = n(),
            n_low = sum(any_low == "Yes"),
            prop_low = mean(any_low == "Yes"))
```

```
law_counts
```

```
# A tibble: 2 x 4
  status    n_members n_low prop_low
  <fct>      <int> <int>    <dbl>
1 Majority     228   125    0.548
2 Minority     220    74    0.336
```

```
p_pooled <- sum(law_counts$n_law) / sum(law_counts$n_members)
se_null <- sqrt(p_pooled * (1 - p_pooled) *
               (1 / law_counts$n_members[1] + 1 /
                law_counts$n_members[2]))
z <- (law_counts$prop_law[1] - law_counts$prop_law[2]) / se_null
z

pnorm(z, lower.tail = FALSE)
```

```
[1] 4.512171
[1] 3.208367e-06
```

- If H_0 ($p_1 = p_2$) is true, both groups share one proportion. Estimate it by **pooling**:
 $\hat{p} = 199/448$

```

p_pooled <- sum(law_counts$n_law) / sum(law_counts$n_members)
se_null <- sqrt(p_pooled * (1 - p_pooled) *
               (1 / law_counts$n_members[1] + 1 /
                law_counts$n_members[2]))
z <- (law_counts$prop_law[1] - law_counts$prop_law[2]) / se_null
z

pnorm(z, lower.tail = FALSE)

```

```

[1] 4.512171
[1] 3.208367e-06

```

- ▶ If H_0 ($p_1 = p_2$) is true, both groups share one proportion. Estimate it by **pooling**: $\hat{p} = 199/448$
- ▶ $z = (\hat{p}_1 - \hat{p}_2)/SE_{H_0} = 4.5$. **lower.tail = FALSE** gives $P(Z \geq z)$, our one-tailed p-value

Checking With `prop.test()`

```
prop.test(x = law_counts$n_law, n = law_counts$n_members,  
          alternative = "greater", correct = FALSE)
```

```
2-sample test for equality of proportions without continuity  
correction
```

```
data:  law_counts$n_law out of law_counts$n_members  
X-squared = 20.36, df = 1, p-value = 3.208e-06  
alternative hypothesis: greater  
95 percent confidence interval:  
 0.1364886 1.0000000  
sample estimates:  
  prop 1    prop 2  
0.5482456 0.3363636
```

Same p-value. `prop.test()` reports z^2 ($4.512^2 = 20.36$), a chi-square statistic: more next week.

- ▶ Every test agrees: a gap this large would almost never show up if party status made no difference. We reject H_0

- ▶ Every test agrees: a gap this large would almost never show up if party status made no difference. We reject H_0
- ▶ Statistically significant isn't the same as why. Majority members also chair every committee and hold the leadership posts (probability lab: is it the majority, or something that comes with it?)

- ▶ Every test agrees: a gap this large would almost never show up if party status made no difference. We reject H_0
- ▶ Statistically significant isn't the same as why. Majority members also chair every committee and hold the leadership posts (probability lab: is it the majority, or something that comes with it?)
- ▶ And is 118th-Congress data a “sample”? Think of it as one draw from the process that produces Congresses

- ▶ There is only one test: compute a statistic, build a null world, simulate it, and ask how often the null world matches your sample

- ▶ There is only one test: compute a statistic, build a null world, simulate it, and ask how often the null world matches your sample
- ▶ Permutation builds the null world by shuffling group labels. `infer: specify()`
→ `hypothesize()` → `generate()` → `calculate()`

- ▶ There is only one test: compute a statistic, build a null world, simulate it, and ask how often the null world matches your sample
- ▶ Permutation builds the null world by shuffling group labels. `infer: specify()`
→ `hypothesize()` → `generate()` → `calculate()`
- ▶ One-tailed tests put all of α in the direction you predicted

- ▶ There is only one test: compute a statistic, build a null world, simulate it, and ask how often the null world matches your sample
- ▶ Permutation builds the null world by shuffling group labels. `infer: specify()`
→ `hypothesize()` → `generate()` → `calculate()`
- ▶ One-tailed tests put all of α in the direction you predicted
- ▶ The t -test does the same thing with a formula. Welch is R's default; `var.equal = T` pools the variances

- ▶ There is only one test: compute a statistic, build a null world, simulate it, and ask how often the null world matches your sample
- ▶ Permutation builds the null world by shuffling group labels. `infer: specify()`
→ `hypothesize()` → `generate()` → `calculate()`
- ▶ One-tailed tests put all of α in the direction you predicted
- ▶ The t -test does the same thing with a formula. Welch is R's default; `var.equal = T` pools the variances
- ▶ Proportions work the same way: `diff in props`, a pooled z -test, or `prop.test()`

Center for Effective Lawmaking. 2025. "House Legislative Effectiveness Scores, 93rd–118th Congresses." <https://thelawmakers.org>.

Volden, Craig, and Alan E. Wiseman. 2014. *Legislative Effectiveness in the United States Congress: The Lawmakers*. New York: Cambridge University Press.

Questions?