CLT-based Inference & Inference for Regression

Dr. Maria Tackett

11.21.19

[Click for PDF of slides](#)

Announcements

- [Exam 02](#) due Sunday, November 25 at 11:59p

- **Friday's Lab**: Exam Office Hours

- **Tuesday's class**: Project meeting day

  - At least one member from each group must be present
  - Each group will meet with me or Becky during the class period.

- Statistics Trivia Night **TODAY 7p - 9p** in Old Chem 101

Inference methods based on CLT

Hypotheses

What are the hypotheses for evaluating if Americans, on average, spend more than 3 hours per day relaxing?

$$H_0 : \mu = 3$$

$$H_A : \mu > 3$$

Set up calculations

Summary statistics from the sample:

```
## # A tibble: 1 x 4
##   x_bar   med    sd     n
##   <dbl> <dbl> <dbl> <int>
## 1  3.68     3  2.63  1154
```

Calculating the test statistic

And the CLT says:

$$\bar{x} \sim N\left(mean = \mu, SE = \frac{\sigma}{\sqrt{n}}\right)$$

How many standard errors away from the population mean is the observed sample mean?

Test Statistic

$$t = \frac{\bar{x} - \text{hypothesized } \mu}{s/\sqrt{n}} = \frac{3.68 - 3}{2.63/\sqrt{1154}} = 8.78$$

The sample mean of 3.68 is 8.78 standard errors above the hypothesized mean, 3.

Calculating the p-value

How likely are we to observe a sample mean that is at least as extreme as the observed sample mean, if in fact the null hypothesis is true

P-value

```
df <- 1154 - 1
pt( 8.7876, df, lower.tail = FALSE)
```

```
## [1] 2.720888e-18
```

Given Americans relax three hours, on average, the probability of observing $\bar{x} \geq 3.68$ hours in a sample of 1154, is $2.72 \times 10^{-18} \approx 0.$

Conclusion

- Since the p-value is small, we reject H_0.

- The data provide convincing evidence that Americans, on average, spend more than 3 hours per day relaxing after work.

Confidence interval for a mean

$$point\ estimate \pm critical\ value \times SE$$

```
se <- 2.63 / sqrt(1154)
df <- 1154 - 1
t_star <- qt(0.95, df)

pt_est <- 3.68
round(pt_est + c(-1,1) * t_star * se, 2)
```

```
## [1] 3.55 3.81
```

The 90% confidence interval is 3.55 to 3.81. Interpret this interval in context of the data.

Built-in functionality in R

- There are built in functions for doing some of these tests in R:

- However a learning goal is this course is not to go through an exhaustive list of all CLT based tests and how to implement them

- Instead you should try to understand how these methods are / are not like the simulation based methods we learned about earlier

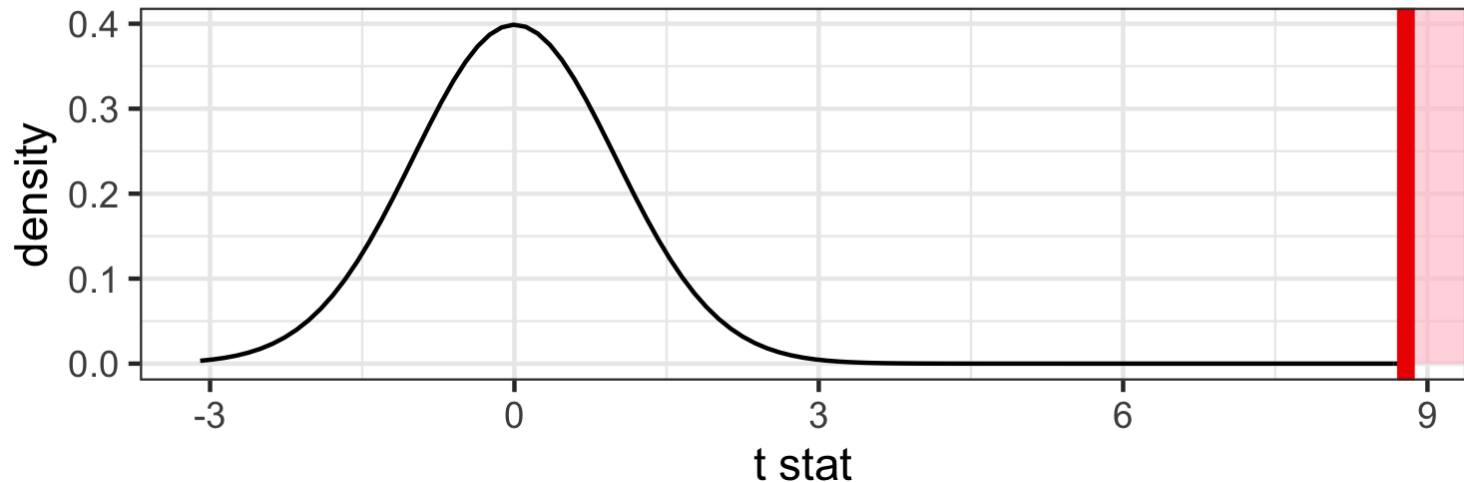What is similar, and what is different, between CLT based test of means vs. simulation based test?

t distribution using `infer`

```
t_null_theor <- gss %>%
  filter(!is.na(hrsrelax)) %>%
  specify(response = hrsrelax) %>%
  hypothesize(null = "point", mu = 3) %>%
  # generate() ## Not used for theoretical
  calculate(stat = "t")
```

t distribution using `infer`

```
visualize(t_null_theor, method = "theoretical") +
  shade_p_value(obs_stat = 8.7876, direction = "greater")
```

```
## Warning: Check to make sure the conditions have been met for the
## theoretical method. {infer} currently does not check these for you.
```

Calculate p-value

```
## Warning: Check to make sure the conditions have been met for the
## theoretical method. {infer} currently does not check these for you.
```


Theoretical t Null Distribution
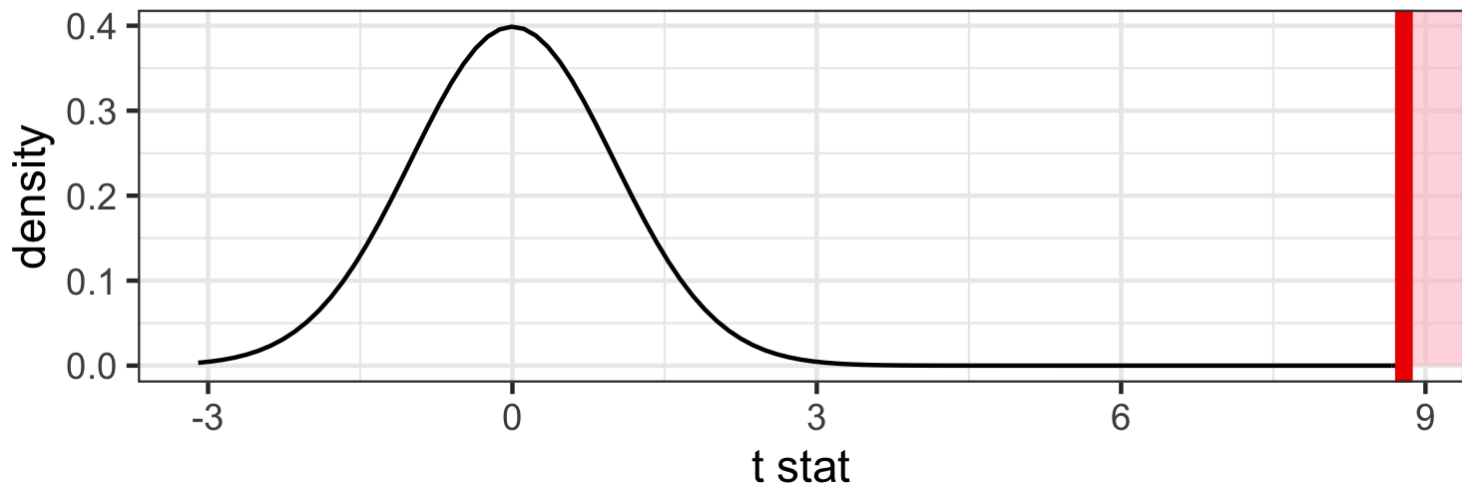
```
df <- 1154 - 1
pt(8.7876, df, lower.tail = FALSE)
```

```
## [1] 2.720888e-18
```

Hypothesis tests in R

```r
# Hypothesis tests
t.test(gss$hrsrelax, mu = 3, alternative = "greater")
```

```
##
##      One Sample t-test
##
## data:  gss$hrsrelax
## t = 8.7876, df = 1153, p-value < 2.2e-16
## alternative hypothesis: true mean is greater than 3
## 95 percent confidence interval:
##  3.552813      Inf
## sample estimates:
## mean of x
##  3.680243
```

Confidence intervals in R

```r
# Confidence intervals
t.test(gss$hrsrelax, conf.level = 0.90)
```

```
##
##      One Sample t-test
##
## data:  gss$hrsrelax
## t = 47.543, df = 1153, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 90 percent confidence interval:
##  3.552813 3.807672
## sample estimates:
## mean of x
##  3.680243
```

Inference for Regression

Riders in Florence, MA

The Pioneer Valley Planning Commission collected data in Florence, MA for 90 days from April 5 to November 15, 2005 using a laser sensor, with breaks in the laser beam recording when a rail-trail user passed the data collection station.

- **hightemp** daily high temperature (in degrees Fahrenheit)
- **volume** estimated number of trail users that day (number of breaks recorded)

```
library(mosaicData)
data(RailTrail)
```

STA 199

Riders in Florence, MA

- **`hightemp`** daily high temperature (in degrees Fahrenheit)
- **`volume`** estimated number of trail users that day (number of breaks recorded)
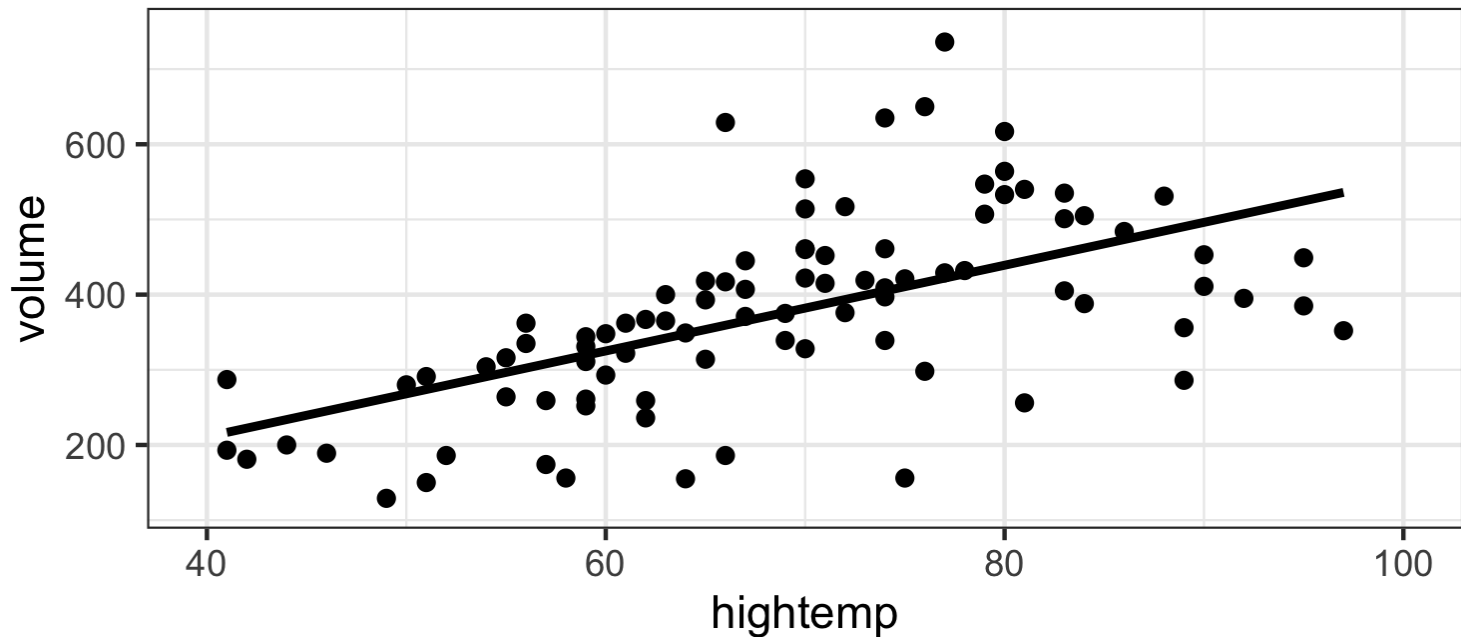
Coefficient interpretation

👤 Interpret the coefficients of the regression model for predicting volume (estimated number of trail users that day) from hightemp (daily high temperature, in F).
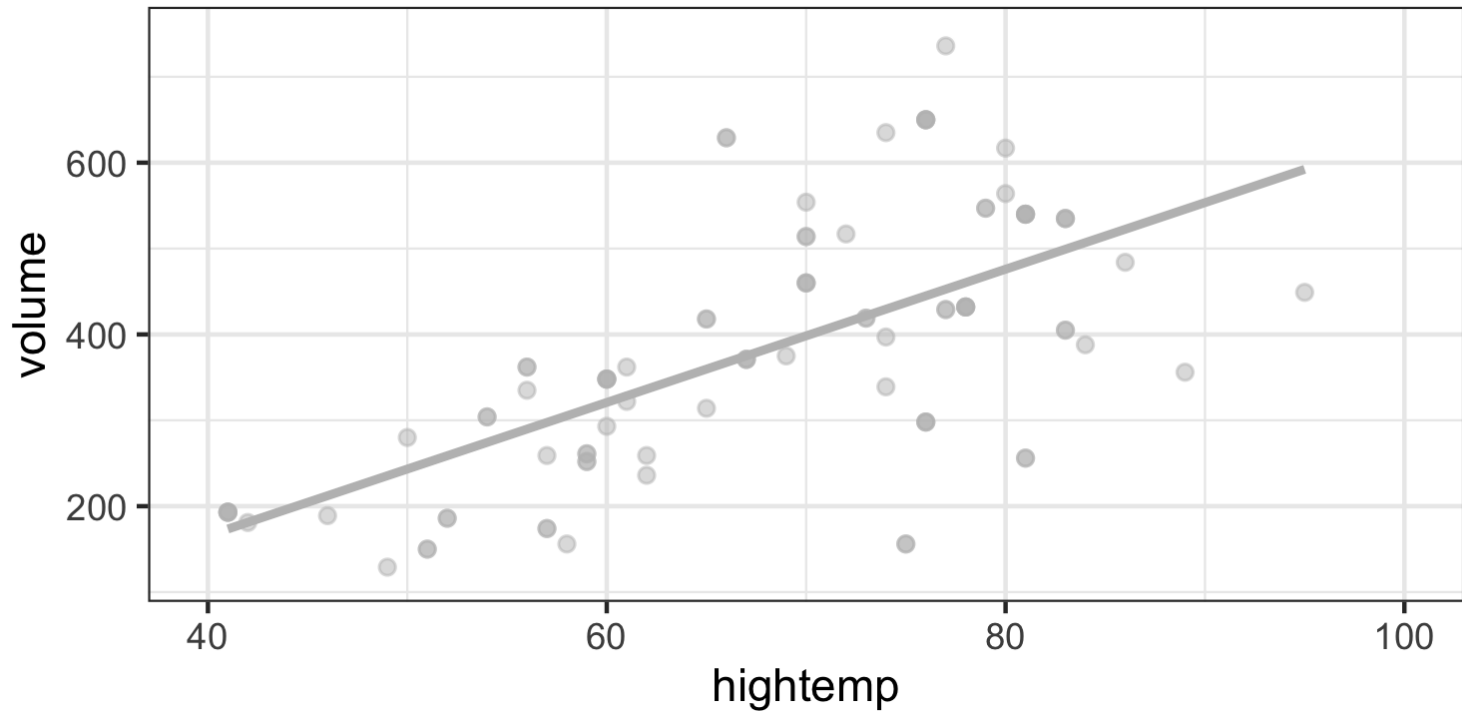
```
m_riders <- lm(volume ~ hightemp, data = RailTrail)
tidy(m_riders) %>%
  select(term, estimate)
```

```
## # A tibble: 2 x 2
##   term          estimate
##   <chr>            <dbl>
## 1 (Intercept)      -17.1
## 2 hightemp          5.70
```

STA 199

Uncertainty around the slope
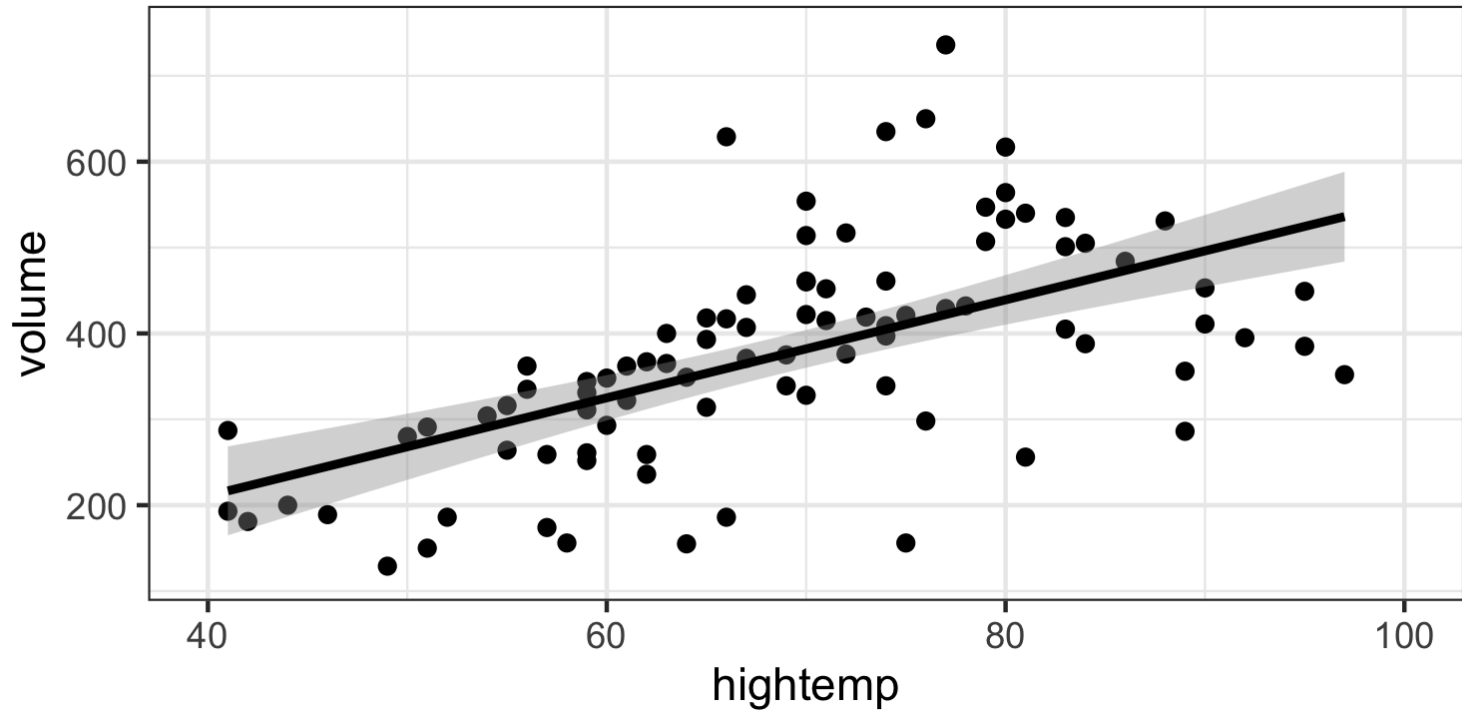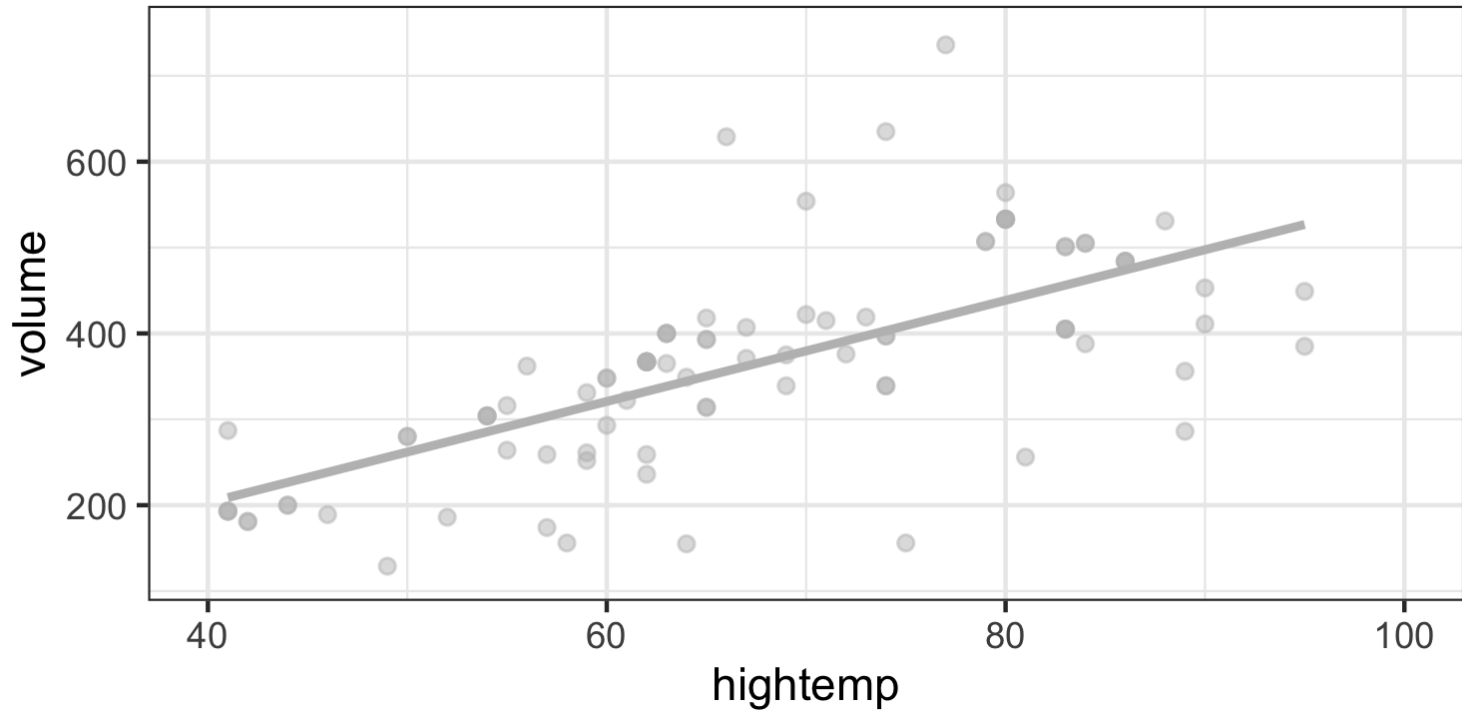
Bootstrapping the data, once
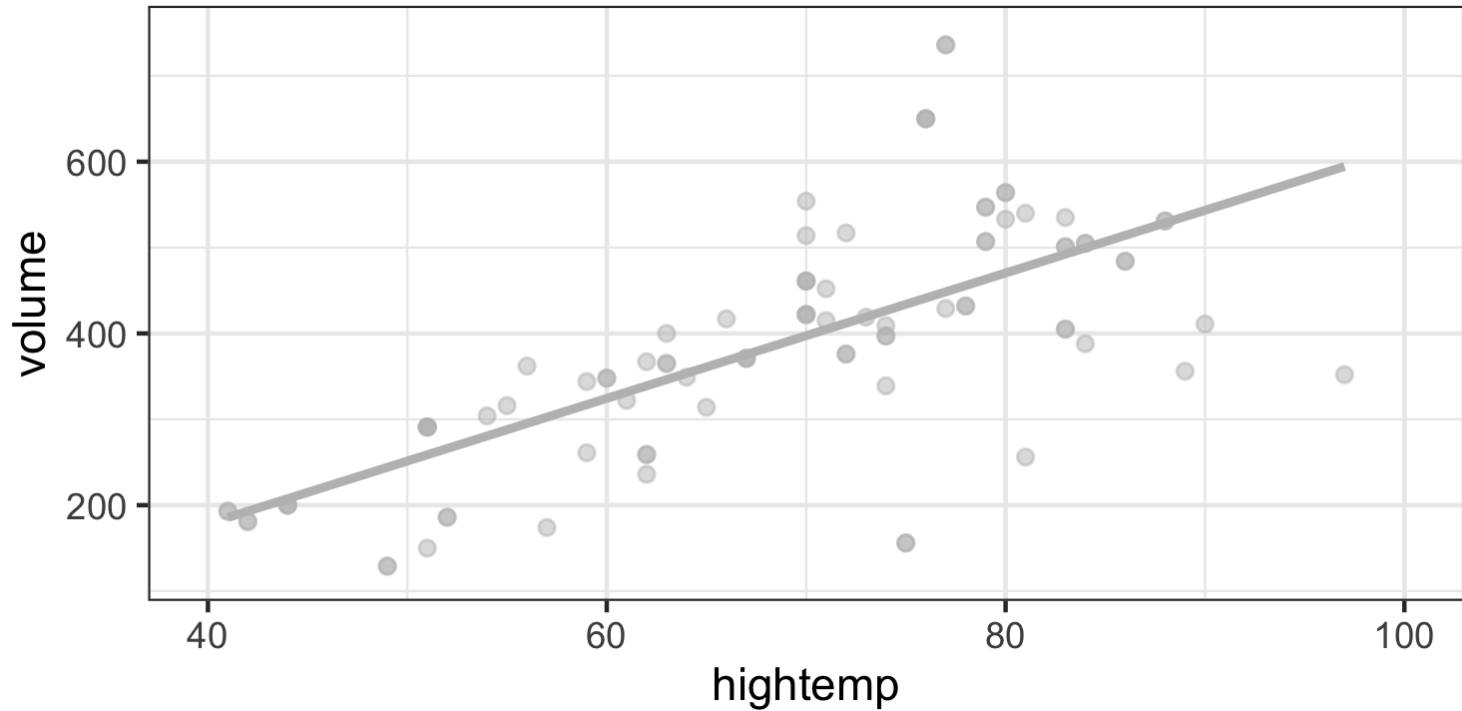


```
## # A tibble: 2 x 2
##   term         estimate
##   <chr>          <dbl>
## 1 (Intercept)  -145.
## 2 hightemp       7.76
```
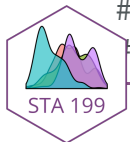
Bootstrapping the data, again


```
## # A tibble: 2 x 2
##   term         estimate
##   <chr>           <dbl>
## 1 (Intercept)     -32.5
## 2 hightemp         5.89
```
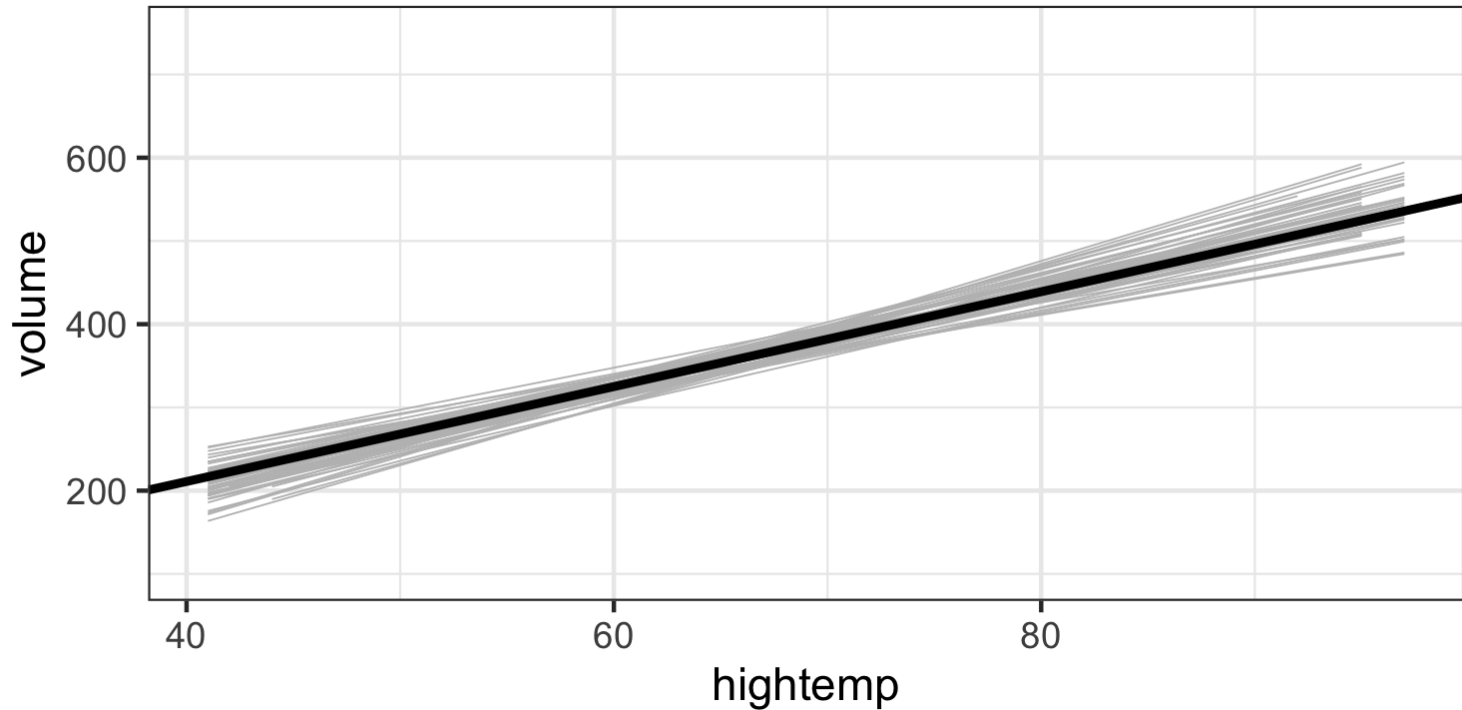
...and again


```
## # A tibble: 2 x 2
##   term         estimate
##   <chr>           <dbl>
## 1 (Intercept)  -114.
## 2 hightemp        7.30
```
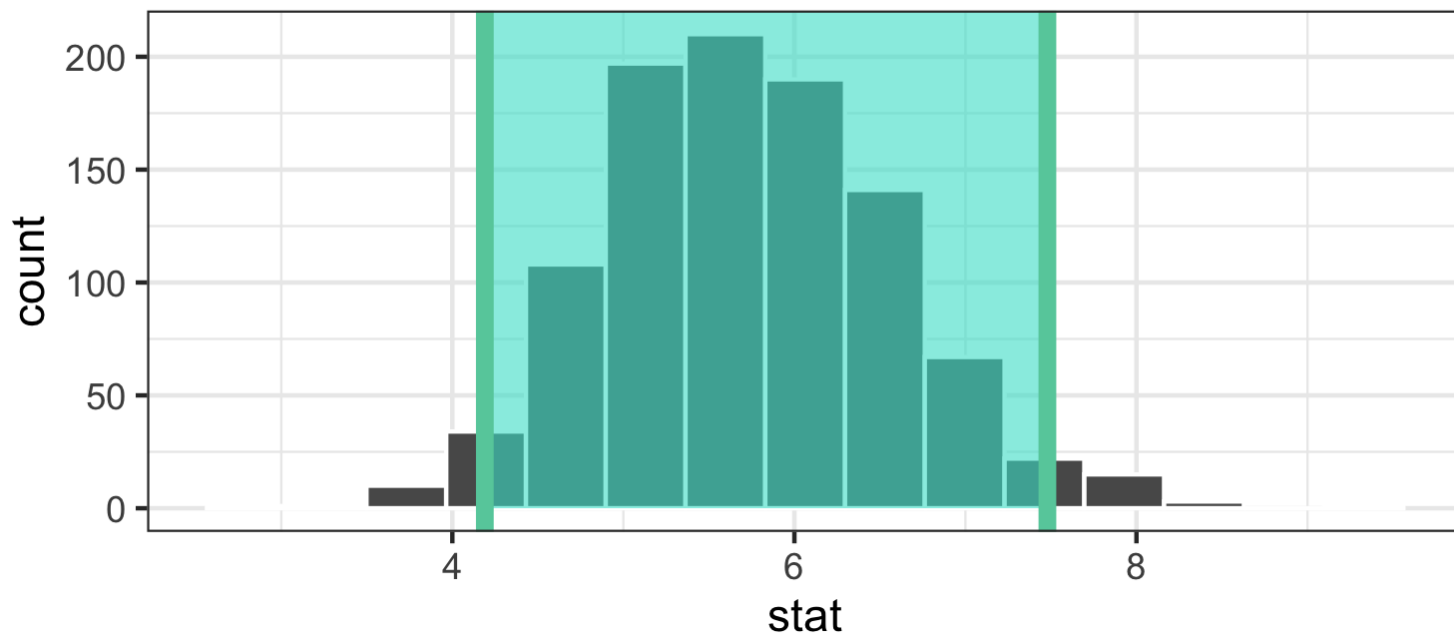
STA 199

Bootstrapping the regression line

STA 199

Bootstrap interval for the slope

```
boot_dist <- RailTrail %>%
  specify(volume ~ hightemp) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "slope")
```


Simulation-Based Null Distribution

Bootstrap interval for the slope

Interpret the bootstrap interval in context of the data.

```
boot_dist %>%
  summarise(l = quantile(stat, 0.025),
            u = quantile(stat, 0.975))
```

```
## # A tibble: 1 x 2
##        l     u
##    <dbl> <dbl>
## 1  4.19  7.48
```
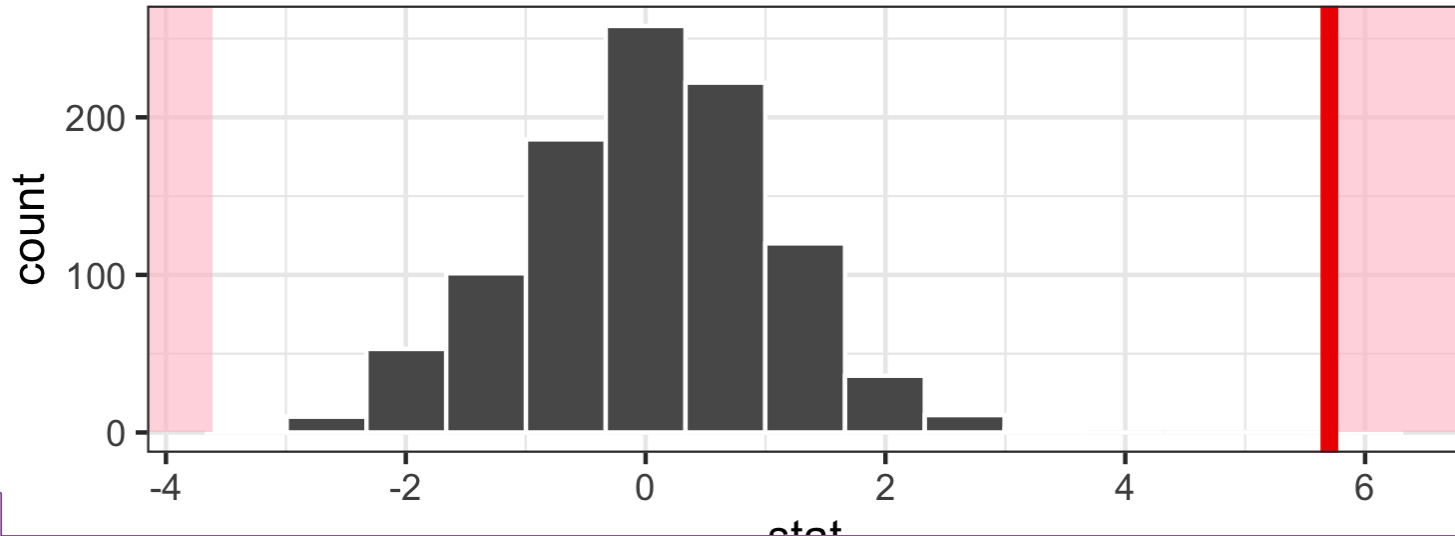
Hypothesis testing for the slope

H_0: No relationship, $\beta_1 = 0$
H_A: There is a relationship, $\beta_1 \neq 0$

```
null_dist <- RailTrail %>%
  specify(response = volume, explanatory = hightemp) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "slope")
```

Simulation-Based Null Distribution

Finding the p-value

```
obs_slope <- tidy(m_riders) %>%
  select(estimate) %>%
  slice(2) %>% pull()

get_p_value(null_dist, obs_slope, direction = "two_sided")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

Hypothesis testing for the slope

… using the Central Limit Theorem

```
tidy(m_riders)
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic      p.value
##   <chr>            <dbl>     <dbl>     <dbl>        <dbl>
## 1 (Intercept)     -17.1      59.4    -0.288 0.774
## 2 hightemp          5.70      0.848    6.72  0.00000000171
```

Conditions for inference for regression

Four conditions:

1. Observations should be independent

2. Residuals should be randomly distributed around 0

3. Residuals should be nearly normally distributed, centered at 0

4. Residuals should have constant variance

STA 199

Checking independence

One consideration might be time series structure of the data. We can check whether one residual depends on the previous by plotting the residuals in the order of data collection.
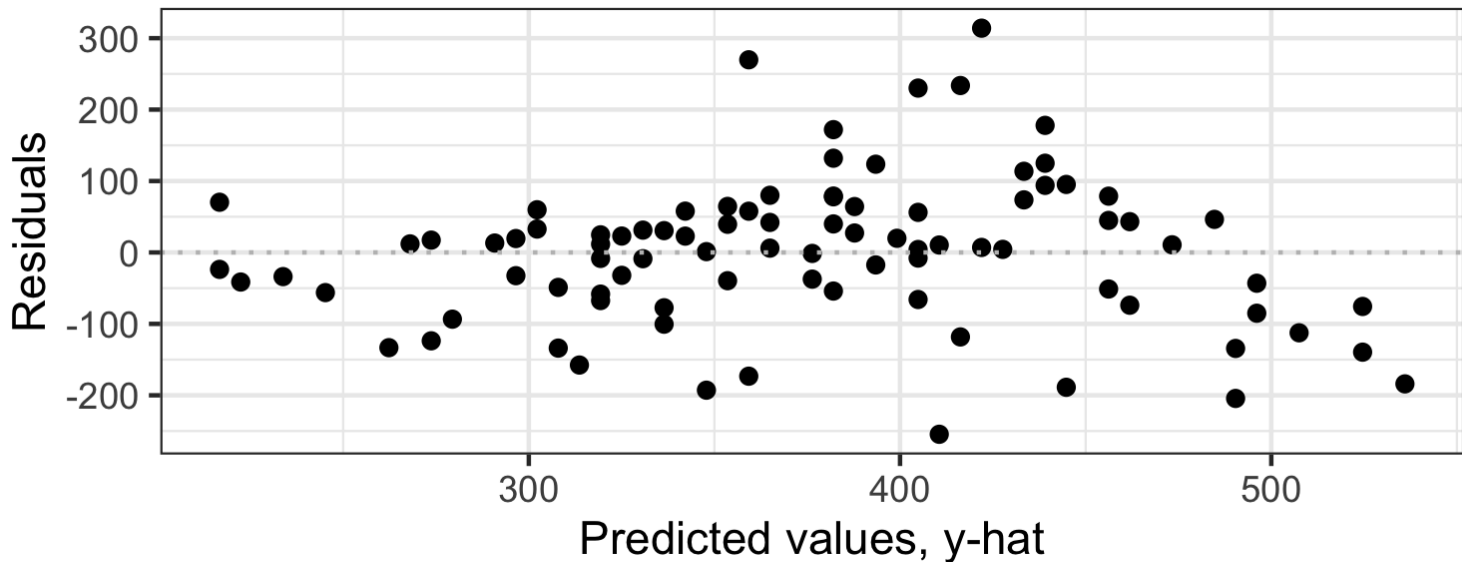
```r
m_riders_aug <- augment(m_riders)
ggplot(data = m_riders_aug, aes(x = 1:nrow(m_riders_aug), y = .resid)) +
  geom_point() +
  labs(x = "Index", y = "Residual")
```
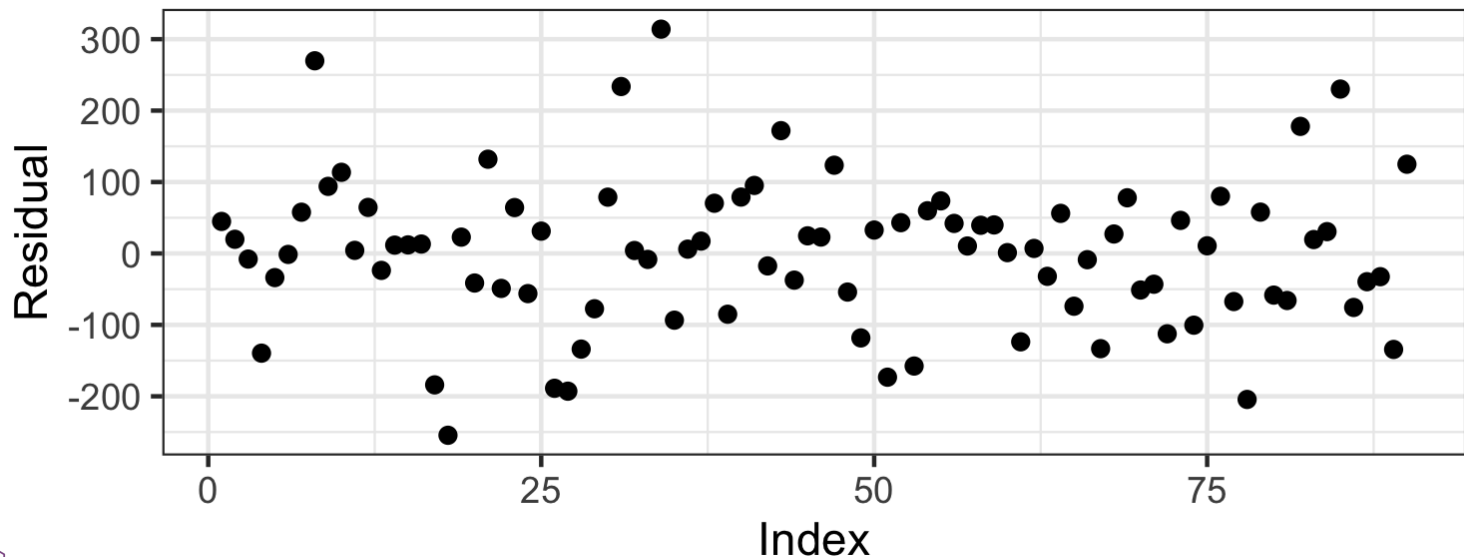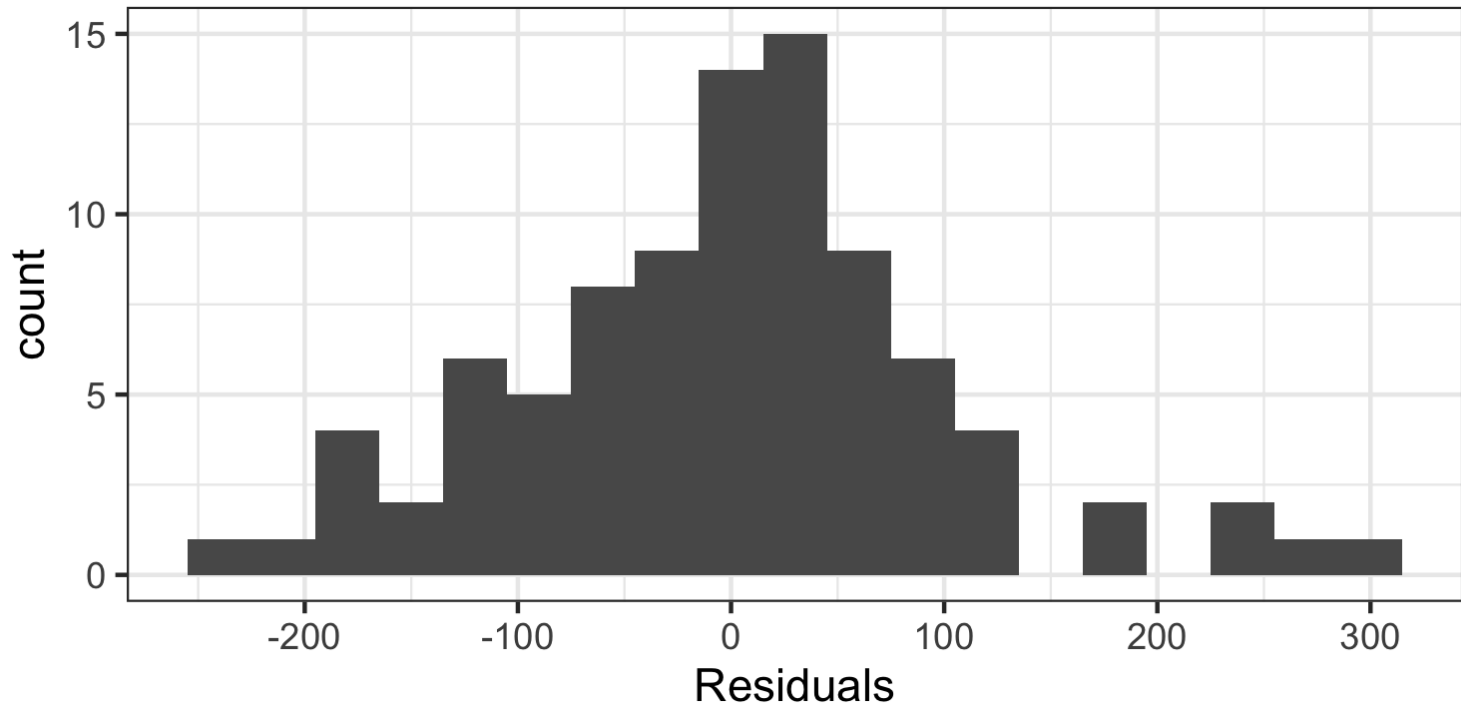
Checking distribution of residuals around 0 and constant variance

```
ggplot(data = m_riders_aug, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, lty = 3, color = "gray") +
  labs(y = "Residuals", x = "Predicted values, y-hat")
```

Checking normality of residuals

```
ggplot(data = m_riders_aug, aes(x = .resid)) +
  geom_histogram(binwidth = 30) +
  labs(x = "Residuals")
```

Thoughts...

- Coefficient p-value

    - If you truly want to know if a coefficient is significantly different from zero (taking the other predictors into account) then use the p-value

    - If you want to know which predictors are important - use model selection