Basic Statistics and Hypothesis Testing in R

Prof. Eric A. Suess

November 28, 2018

If you want to learn about Statistics using base R a nice website is the Quick-R website, see Statistics > t-tests

These are some example of basic statistics and hypothesis testing in R. Most of the code here is from base R.

We will use the *mtcars* data set.

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------------------ t

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ---------------------------------------------------------------------------------- tidyve
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
```

```
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

Summary Statistics

```r
mtcars %>% summarize(mpg_mean = mean(mpg), mpg_sd = sd(mpg))
```

```
##   mpg_mean   mpg_sd
## 1 20.09062 6.026948
```

Subsets and statistics.

```r
mtcars %>% group_by(vs) %>%
  summarize(mpg_mean = mean(mpg), mpg_sd = sd(mpg))
```

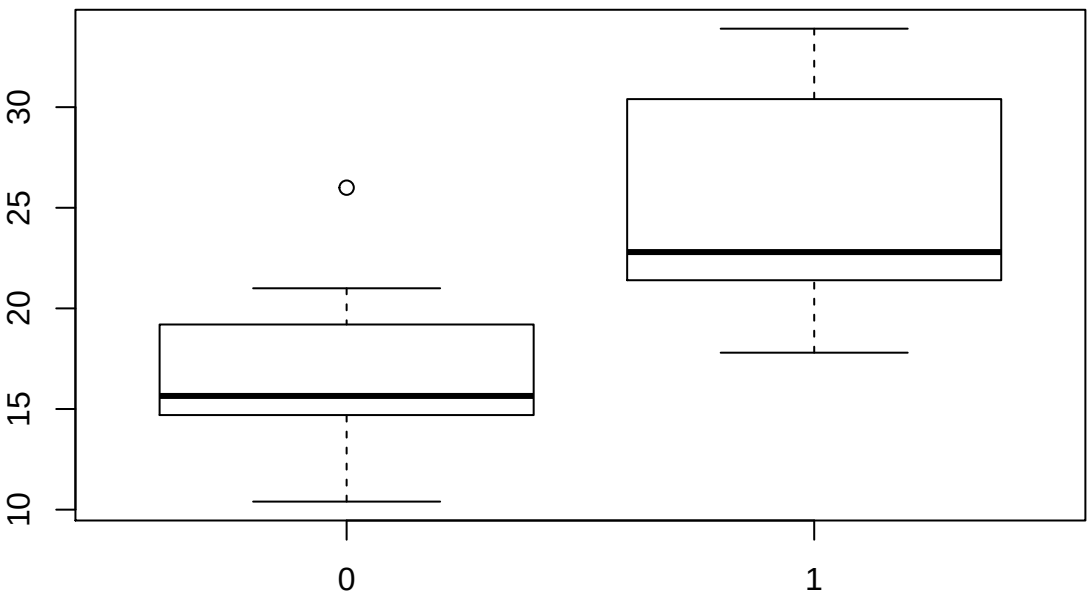
```
## # A tibble: 2 x 3
##      vs mpg_mean mpg_sd
##   <dbl>    <dbl>  <dbl>
## 1     0     16.6   3.86
## 2     1     24.6   5.38
```

Note that the t.test function does not work well with the tidyverse. There is a new package called *infer* that works with the tidyverse. And if you are interested chech out the *broom* package.

I like using the formula interface when doing hypothesis testing.

t test

```r
?t.test

with(mtcars, boxplot(mpg ~ vs))
```




```r
output1 <- with(mtcars, t.test(mpg ~vs))

output1
```

```
##
##  Welch Two Sample t-test
##
## data:  mpg by vs
## t = -4.6671, df = 22.716, p-value = 0.0001098
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -11.462508  -4.418445
## sample estimates:
## mean in group 0 mean in group 1
##        16.61667        24.55714
```

```r
summary(output1)
```

```
##             Length Class  Mode
## statistic   1      -none- numeric
## parameter   1      -none- numeric
## p.value     1      -none- numeric
## conf.int    2      -none- numeric
## estimate    2      -none- numeric
## null.value  1      -none- numeric
## alternative 1      -none- character
## method      1      -none- character
## data.name   1      -none- character
```

```r
output1$statistic
```

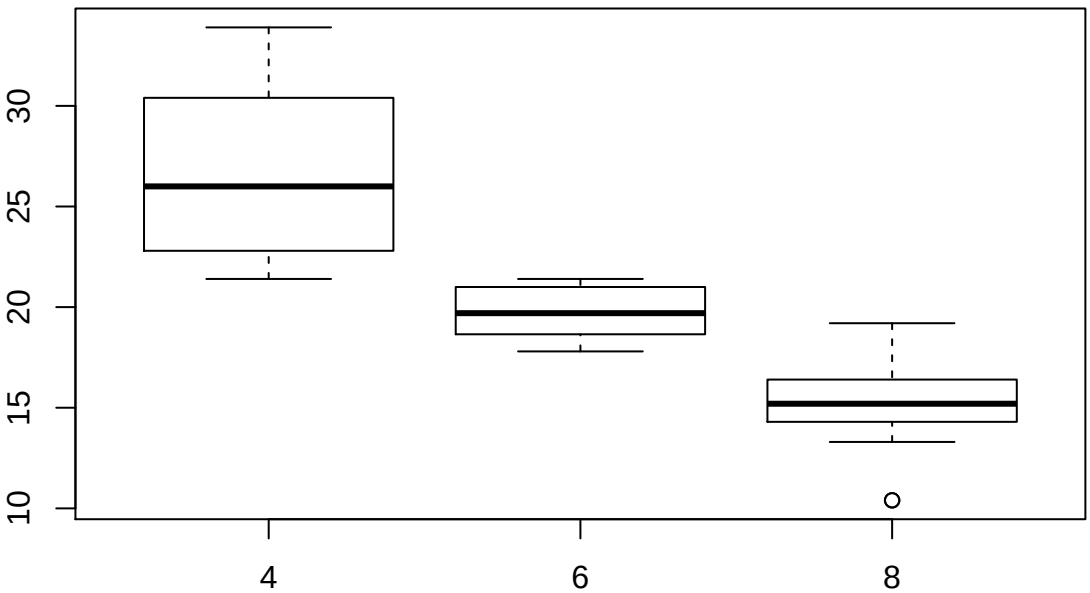
```
##         t
## -4.667053
```

```r
output1$p.value
```

```
## [1] 0.0001098368
```

ANOVA

```
?aov

with(mtcars, boxplot(mpg ~cyl))
```




```
output2 <- with(mtcars, aov(mpg ~ cyl))

output2
```

```
## Call:
##    aov(formula = mpg ~ cyl)
##
## Terms:
##                       cyl Residuals
## Sum of Squares  817.7130  308.3342
## Deg. of Freedom        1        30
##
## Residual standard error: 3.205902
## Estimated effects may be unbalanced
```

```
summary(output2)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## cyl          1  817.7   817.7   79.56 6.11e-10 ***
## Residuals   30  308.3    10.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Linear Regression

```
?lm

attach(mtcars)
```

```
## The following object is masked from package:ggplot2:
##
##     mpg
```

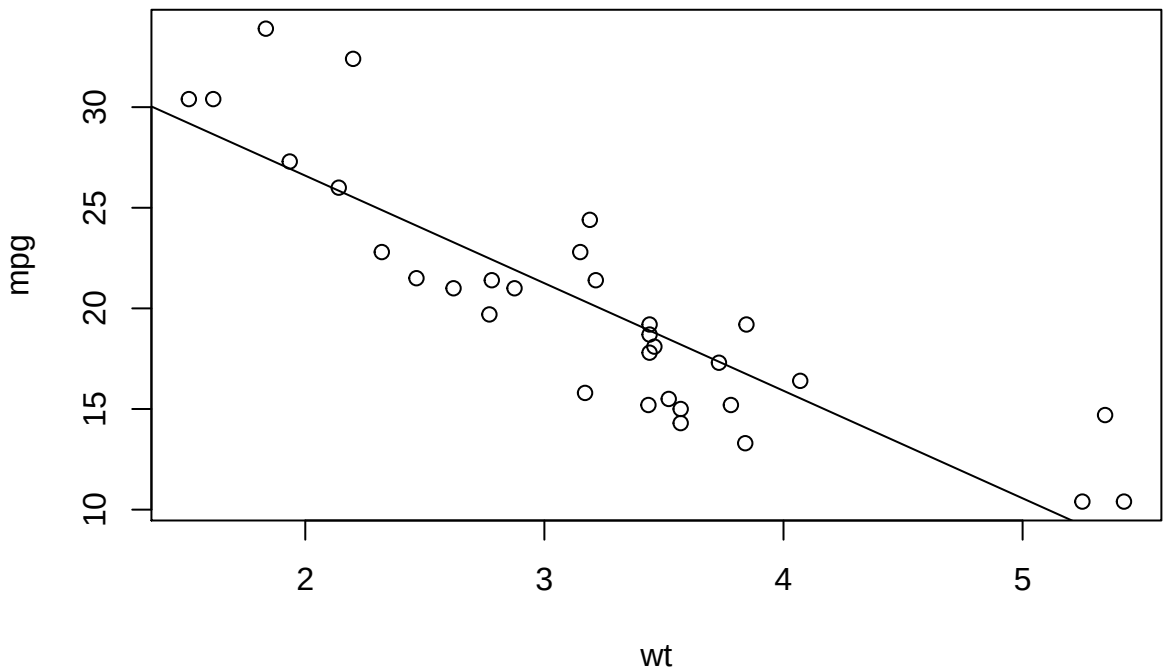
```r
plot(mpg ~ wt)

output3 <-lm(mpg ~ wt)

output3
```

```
##
## Call:
## lm(formula = mpg ~ wt)
##
## Coefficients:
## (Intercept)           wt
##      37.285       -5.344
```

```r
summary(output3)
```

```
##
## Call:
## lm(formula = mpg ~ wt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5432 -2.3647 -0.1252  1.4096  6.8727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
## wt           -5.3445     0.5591  -9.559 1.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.046 on 30 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7446
## F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

```r
plot(mpg ~ wt)
abline(lm(mpg ~ wt))
```

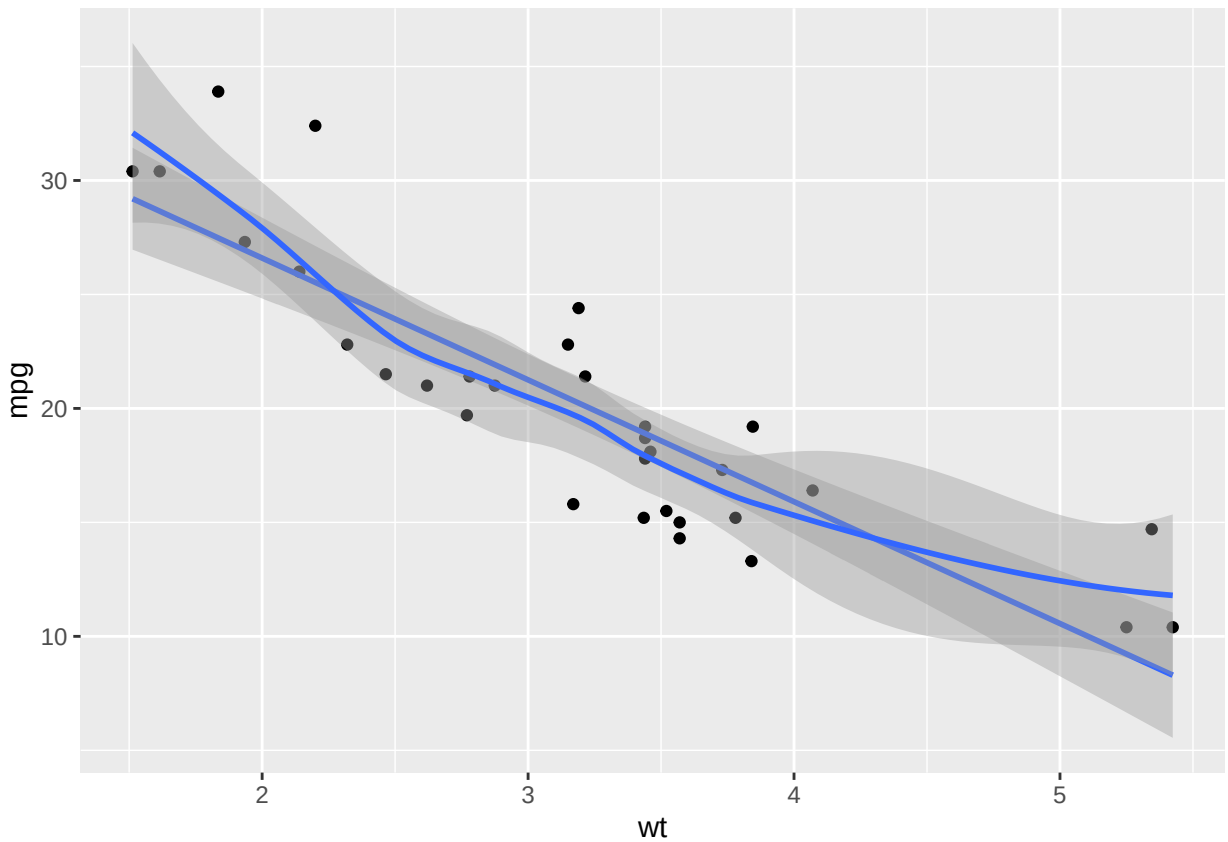



```
detach(mtcars)
```

Using ggplot

```
mtcars %>% ggplot(aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method=lm) +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



If you want to learn Hypothesis Testing using modern R code check out the book moderndive. See Chapter 10. The authors of this book are working on a new package called infer R package.

```
library(infer)
```

The two sample t test example from the website.

```
library(nycflights13)
library(dplyr)
library(stringr)
library(infer)

set.seed(2017)
fli_small <- flights %>%
  sample_n(size = 500) %>%
  mutate(half_year = case_when(
    between(month, 1, 6) ~ "h1",
    between(month, 7, 12) ~ "h2"
  )) %>%
  mutate(day_hour = case_when(
    between(hour, 1, 12) ~ "morning",
    between(hour, 13, 24) ~ "not morning"
  )) %>%
  select(arr_delay, dep_delay, half_year,
         day_hour, origin, carrier)
```

```
obs_t <- fli_small %>%
  specify(arr_delay ~ half_year) %>%
  calculate(stat = "t", order = c("h1", "h2"))
```

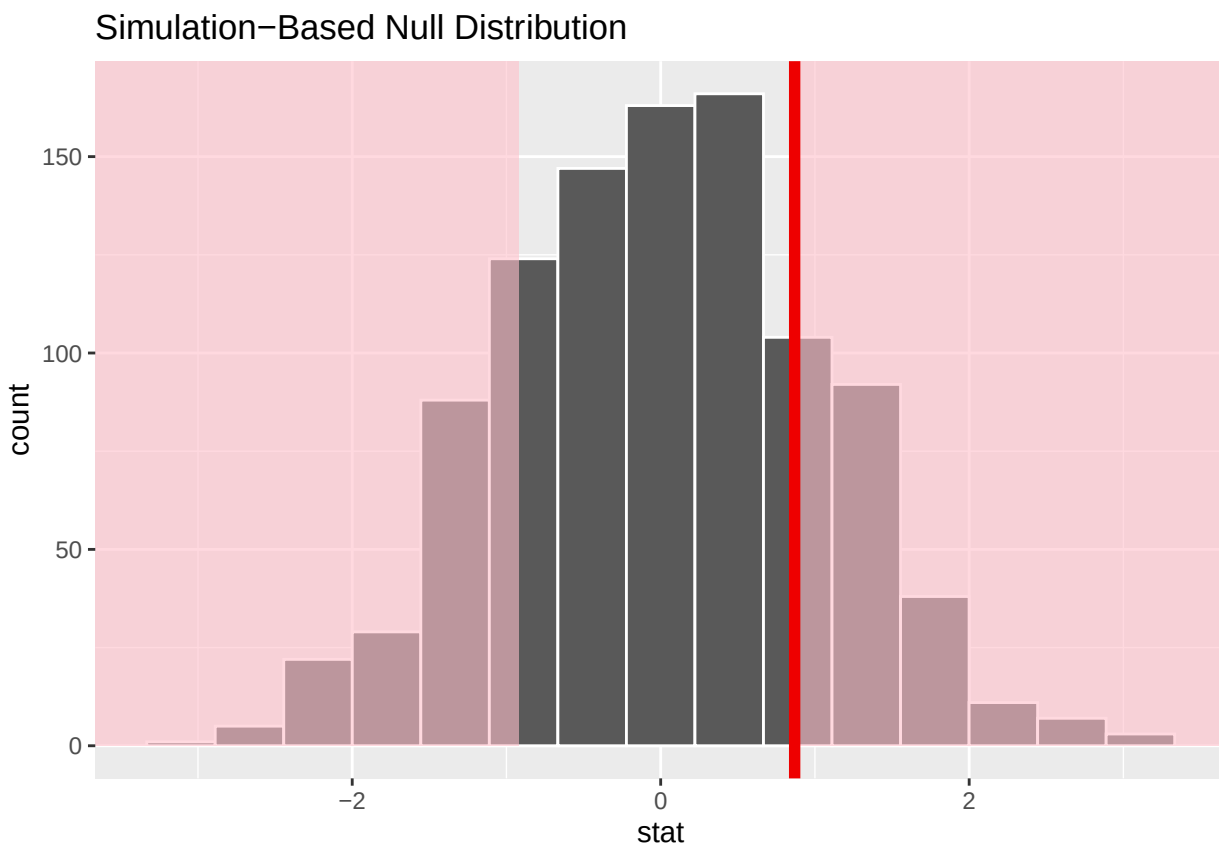
```
## Warning: Removed 15 rows containing missing values.
```

```
obs_t <- fli_small %>%
  t_stat(formula = arr_delay ~ half_year, order = c("h1", "h2"))
```

```
t_null_perm <- fli_small %>%
  # alt: response = arr_delay, explanatory = half_year
  specify(arr_delay ~ half_year) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "t", order = c("h1", "h2"))
```

```
## Warning: Removed 15 rows containing missing values.
```

```
visualize(t_null_perm) +
  shade_p_value(obs_stat = obs_t, direction = "two_sided")
```



Randomized p-value

```
t_null_perm %>%
  get_p_value(obs_stat = obs_t, direction = "two_sided")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.408
```

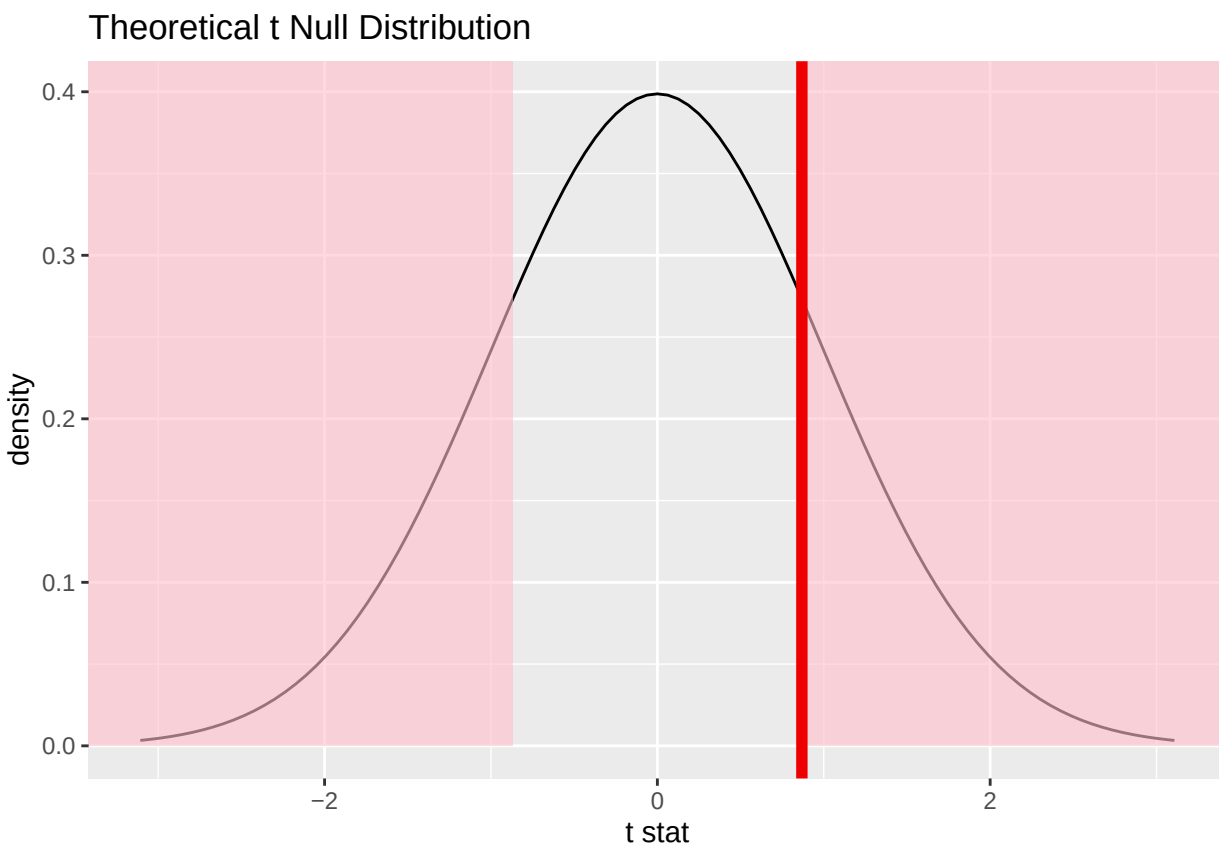
Theoretical p-value

```r
t_null_theor <- fli_small %>%
  # alt: response = arr_delay, explanatory = half_year
  specify(arr_delay ~ half_year) %>%
  hypothesize(null = "independence") %>%
  # generate() ## Not used for theoretical
  calculate(stat = "t", order = c("h1", "h2"))
```

```
## Warning: Removed 15 rows containing missing values.
```

```r
visualize(t_null_theor, method = "theoretical") +
  shade_p_value(obs_stat = obs_t, direction = "two_sided")
```

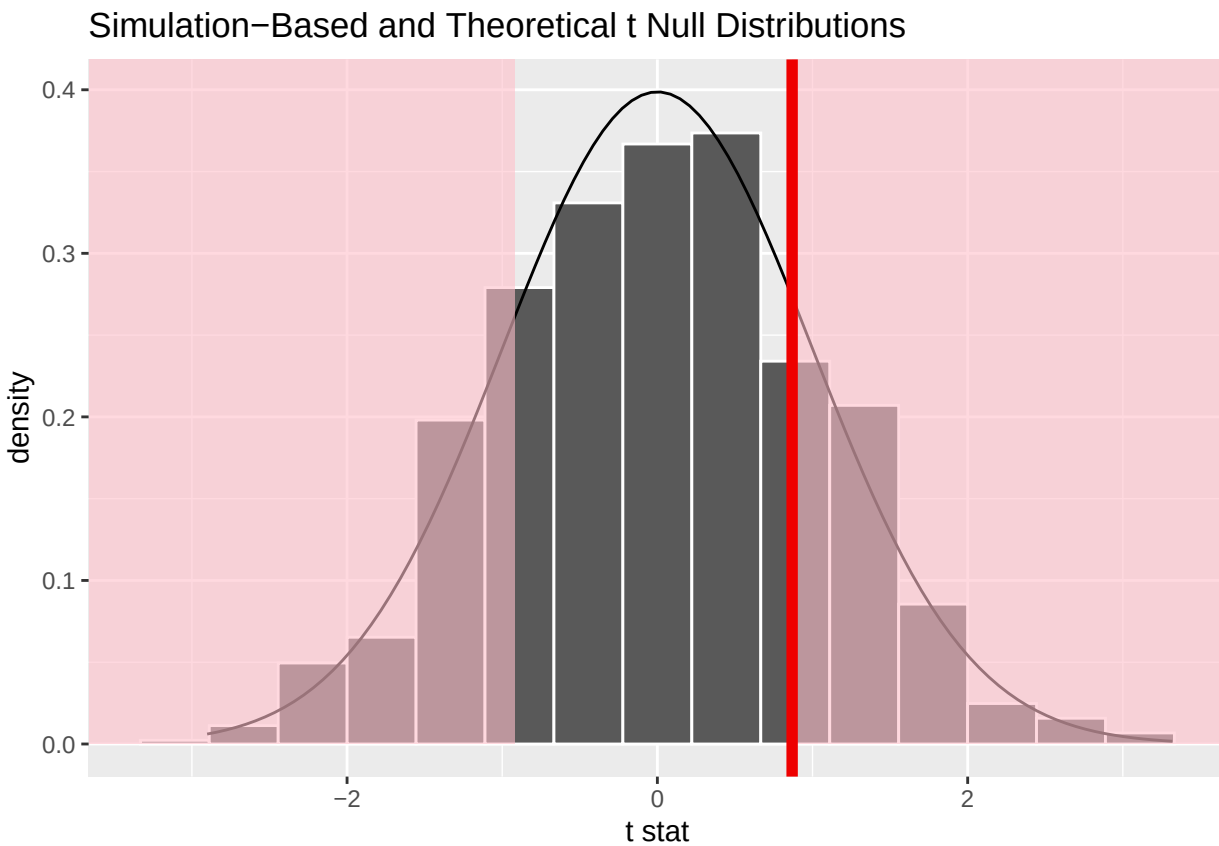
```
## Warning: Check to make sure the conditions have been met for the
## theoretical method. {infer} currently does not check these for you.
```



Overlay

```r
visualize(t_null_perm, method = "both") +
  shade_p_value(obs_stat = obs_t, direction = "two_sided")
```

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



Compute the Theoretical p-value

```
fli_small %>%
  t_test(formula = arr_delay ~ half_year,
         alternative = "two_sided",
         order = c("h1", "h2")) %>%
  dplyr::pull(p_value)
```

```
## [1] 0.3855325
```