Final Project

Helen Moses, Helen Cross, and Hazel DeHarpporte

3/09/2022

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

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

```
library(mosaic)
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:purrr':
##
##     cross
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
```

```
##
##     max, mean, min, prod, range, sample, sum
```

```
library(infer)
```

```
##
## Attaching package: 'infer'
```

```
## The following objects are masked from 'package:mosaic':
##
##     prop_test, t_test
```

```
library(patchwork)
library(forcats)
```

```
ultra_subset <- read.csv("ultra_subset.csv")
```

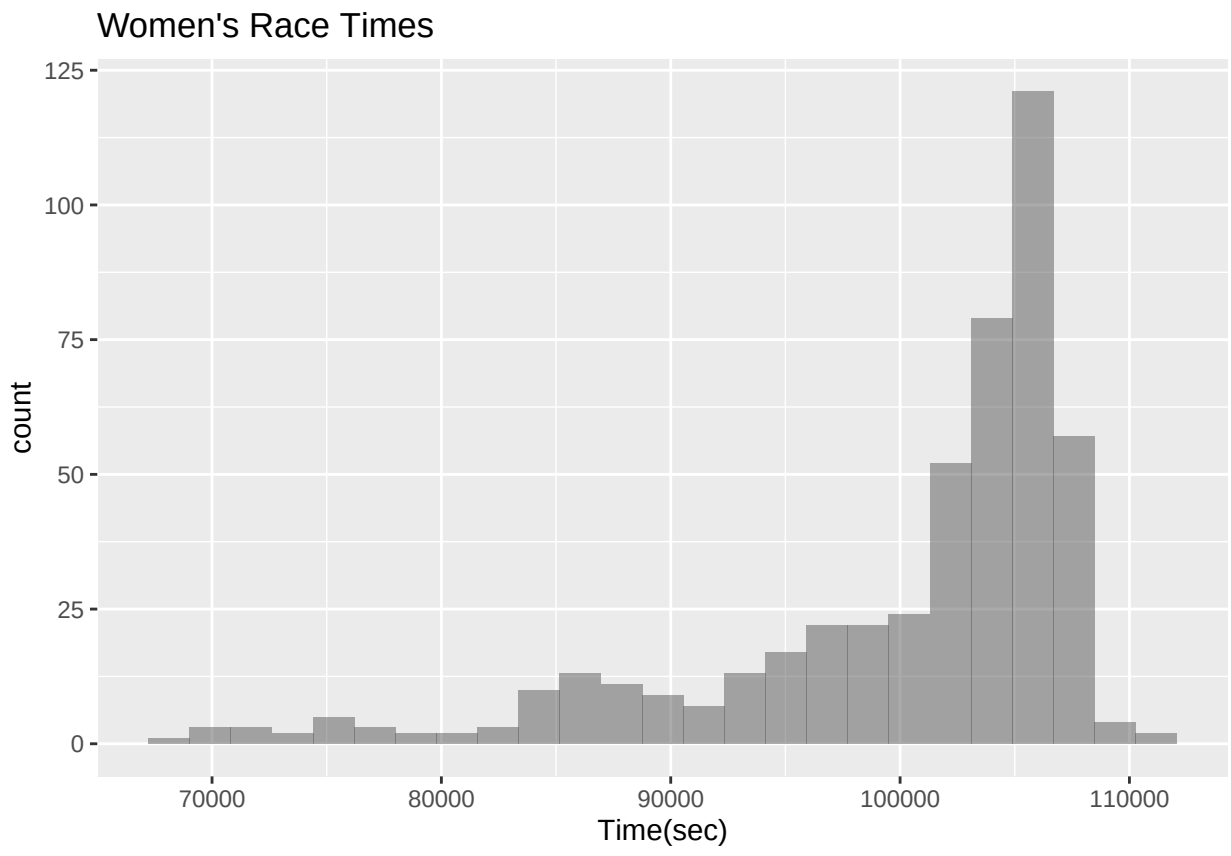
1.

First we had to divide the data set into a set with only women data and a set with only men data. Here is the code we used to do that:

```
Women <- filter(ultra_subset,  gender== "W")
Men <- filter(ultra_subset, gender== "M")
```

Once we got the two data sets, we were able to analyze only men data and only women data. Our first step in analysis was to plot histograms of the race time distributions to get an idea of what the data we were working with looked like. We also found the favstats to get the mean of the women's race time and men's race times along with a couple other important statistics.

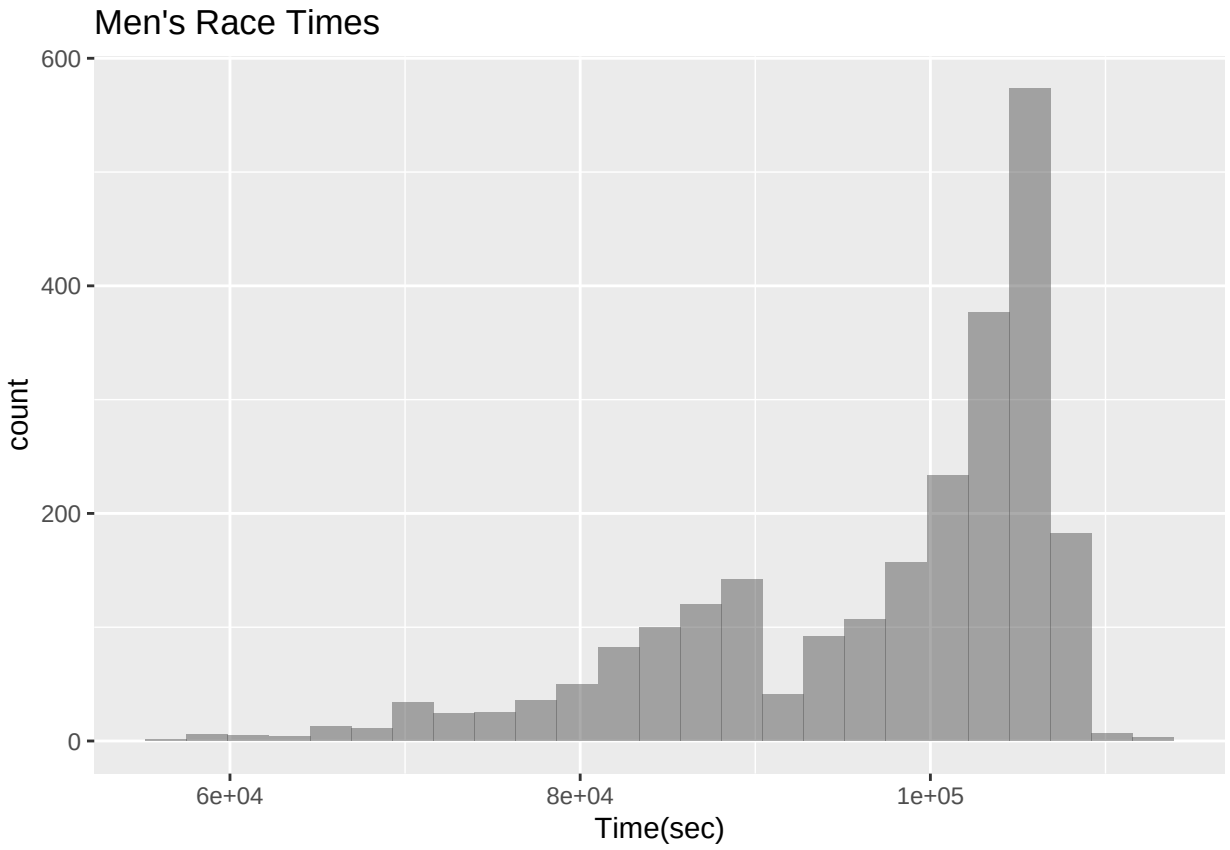
```
gf_histogram(~time_in_seconds, data = Women, na.rm = TRUE, title = "Women's Race Times", xlab = "Time(s
```



```
favstats(~time_in_seconds, data = Women)
```

```
##    min      Q1 median     Q3    max     mean       sd   n missing
##  68427 97205.5 103517 105846 111459 100309.5 8339.321 487     129
```

```
gf_histogram(~time_in_seconds, data = Men, na.rm = TRUE, title = "Men's Race Times", xlab = "Time(sec)"
```




```
favstats(~time_in_seconds, data = Men)
```

```
##    min    Q1 median     Q3    max     mean       sd    n missing
##  57117 89281 101602 105248 113478 97153.04 10404.22 2425     421
```

2.

Then we had to make sure that R recognized the variable "race_year_id" as a catergorical variable, not a numerical variable so we used the following code:

```
Women$race_year_id <- as.factor(Women$race_year_id)
```

```
Men$race_year_id<- as.factor(Men$race_year_id)
```

We also needed to relabel the "race_year_id" so that the plot would be easier to follow for people who were not as familiar with the data. To do this, we used the below code:

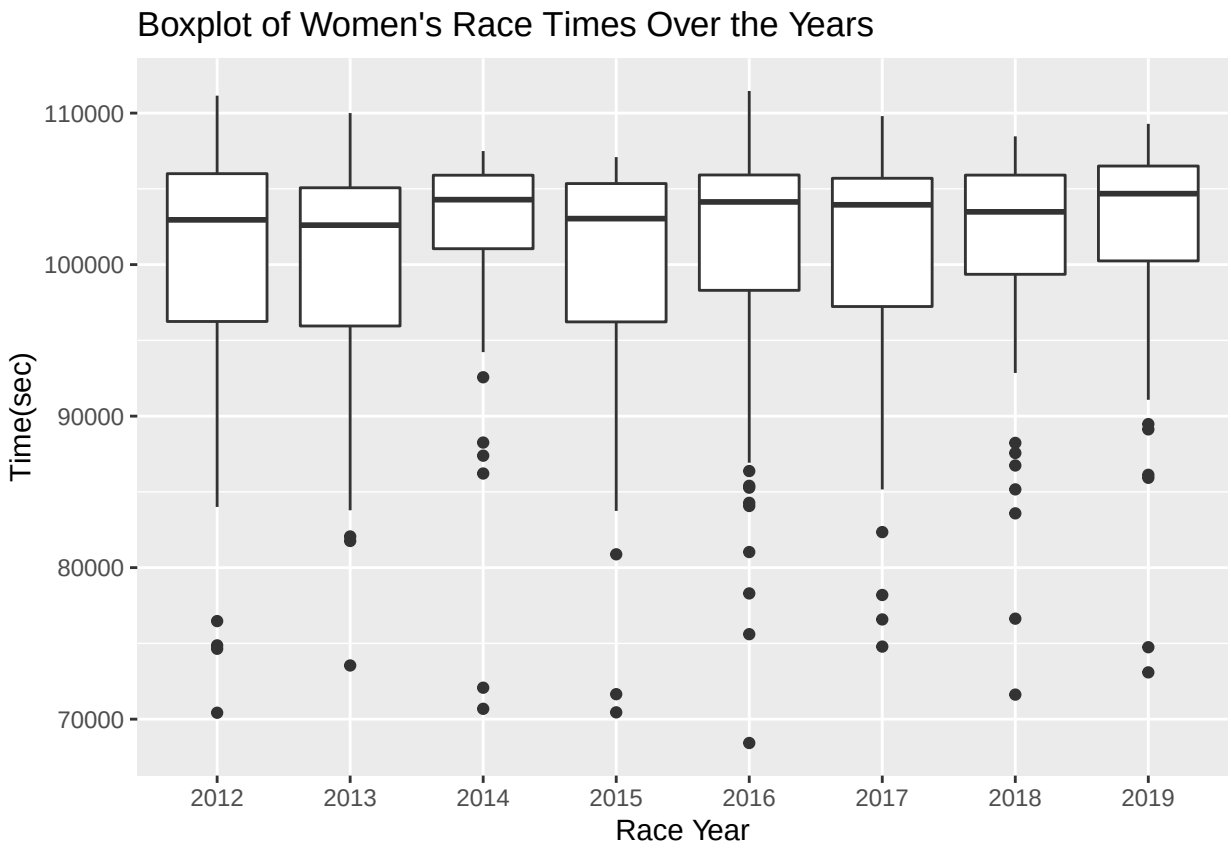
```
Women$race_year_id <- fct_recode(
  Women$race_year_id,
  "2012" = "2917",
  "2013" = "4294",
  "2014" = "7113",
  "2015" = "10852",
```

```
  "2016" = "15375",
  "2017" = "25331",
  "2018" = "36235",
  "2019" = "41269"
)
```

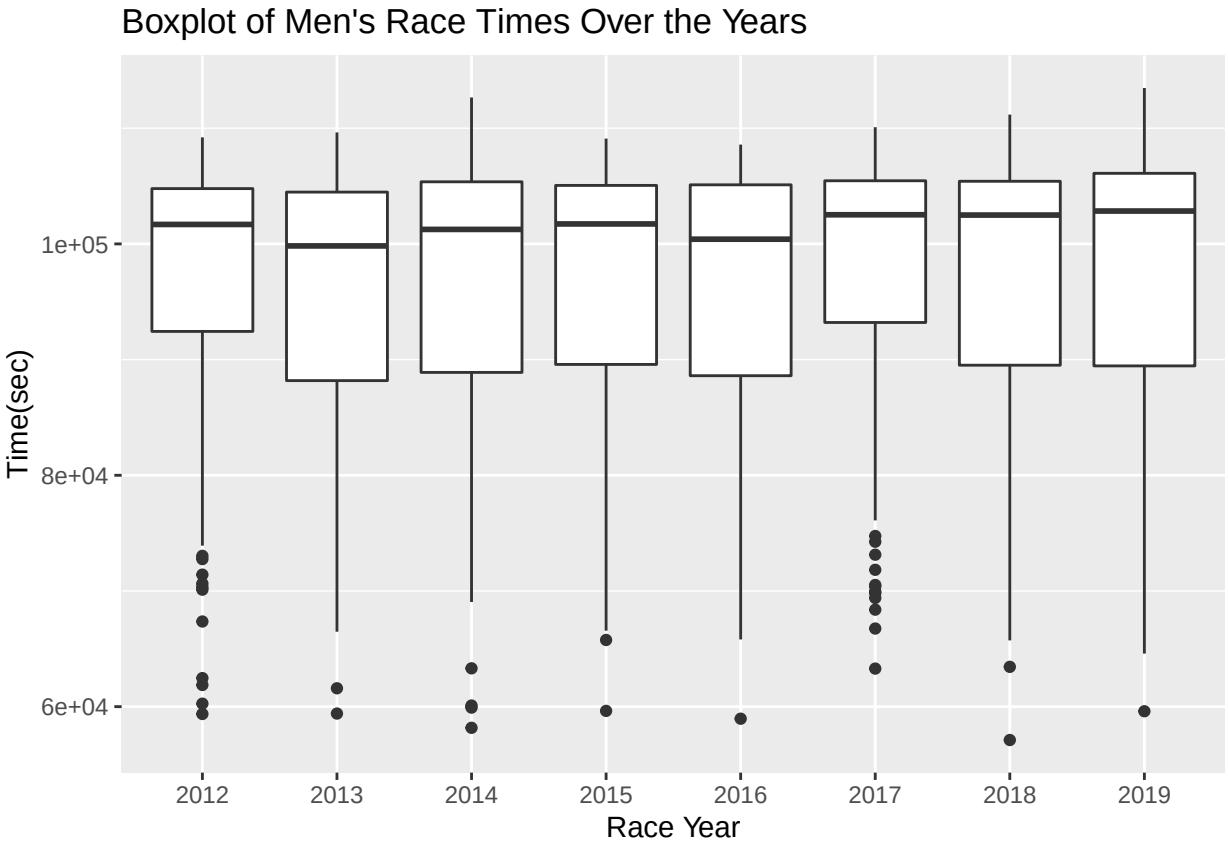
```
Men$race_year_id <- fct_recode(
  Men$race_year_id,
  "2012" = "2917",
  "2013" = "4294",
  "2014" = "7113",
  "2015" = "10852",
  "2016" = "15375",
  "2017" = "25331",
  "2018" = "36235",
  "2019" = "41269"
)
```

After we made those two changes to how R interprets the variables in "race_year_id", we could use the following code to plot our side by side boxplots. The boxplots were necessary to allow us to visualize if there was a difference in race times between 2012 and 2019.

```
gf_boxplot(time_in_seconds ~ race_year_id, data = Women, na.rm = TRUE, title = "Boxplot of Women's Race
```




```
gf_boxplot(time_in_seconds ~ race_year_id, data = Men, na.rm = TRUE, title = "Boxplot of Men's Race Tim
```



We also needed some numerical values for our analysis which we found with the following code:

```
favstats(time_in_seconds ~ race_year_id, data = Women)
```

```
##   race_year_id   min       Q1 median     Q3    max      mean       sd  n
## 1         2012 70423  96246.0 102954 106002 111150  98948.09 9433.536 56
## 2         2013 73544  95948.0 102603 105072 110001  99799.53 7349.056 81
## 3         2014 70684 101049.0 104287 105900 107495 101116.25 8077.906 51
## 4         2015 70449  96219.5 103035 105350 107096  99309.80 8710.995 59
## 5         2016 68427  98302.0 104136 105916 111459 100077.09 9454.094 65
## 6         2017 74789  97233.0 103946 105697 109802  99855.31 8845.775 45
## 7         2018 71620  99359.5 103482 105908 108464 101177.25 7911.871 59
## 8         2019 73087 100245.0 104683 106506 109290 101996.01 7220.482 71
##   missing
## 1       0
## 2       0
## 3       0
## 4       0
## 5       0
## 6       0
## 7       0
## 8     129
```

```
favstats(time_in_seconds ~ race_year_id, data = Men)
```

```
##   race_year_id   min       Q1   median       Q3    max     mean       sd   n
## 1         2012 59367 92437.00 101683.0 104783.0 109212 97556.86 10011.65 307
## 2         2013 59403 88184.50  99827.5 104480.2 109638 96160.42 10223.86 416
```

```
## 3         2014 58172 88894.00 101254.0 105370.0 112661 97104.83 10695.87 305
## 4         2015 59634 89583.00 101727.5 105058.0 109106 97097.75 10229.95 254
## 5         2016 58959 88608.75 100409.0 105108.2 108588 96619.07 10510.83 276
## 6         2017 63291 93209.75 102520.5 105459.5 110090 98100.85 10003.19 242
## 7         2018 57117 89516.00 102494.0 105416.5 111181 97299.98 10771.85 319
## 8         2019 59604 89456.75 102844.0 106103.5 113478 97770.14 10683.40 306
##   missing
## 1       0
## 2       0
## 3       0
## 4       0
## 5       0
## 6       0
## 7       0
## 8     421
```

3.

We needed a way to determine if the above results were statistically significant, so we decided to use ANOVA. Our steps to finding a p-value can be seen through the following lines of code.

```
aov(time_in_seconds ~ race_year_id, data = Women)
```

```
## Call:
##    aov(formula = time_in_seconds ~ race_year_id, data = Women)
##
## Terms:
##                 race_year_id   Residuals
## Sum of Squares     476184740 33322334485
## Deg. of Freedom            7         479
##
## Residual standard error: 8340.651
## Estimated effects may be unbalanced
## 129 observations deleted due to missingness
```

```
Women %>%
  specify(time_in_seconds ~ race_year_id) %>%
  calculate(stat = "F") %>%
  pull()
```

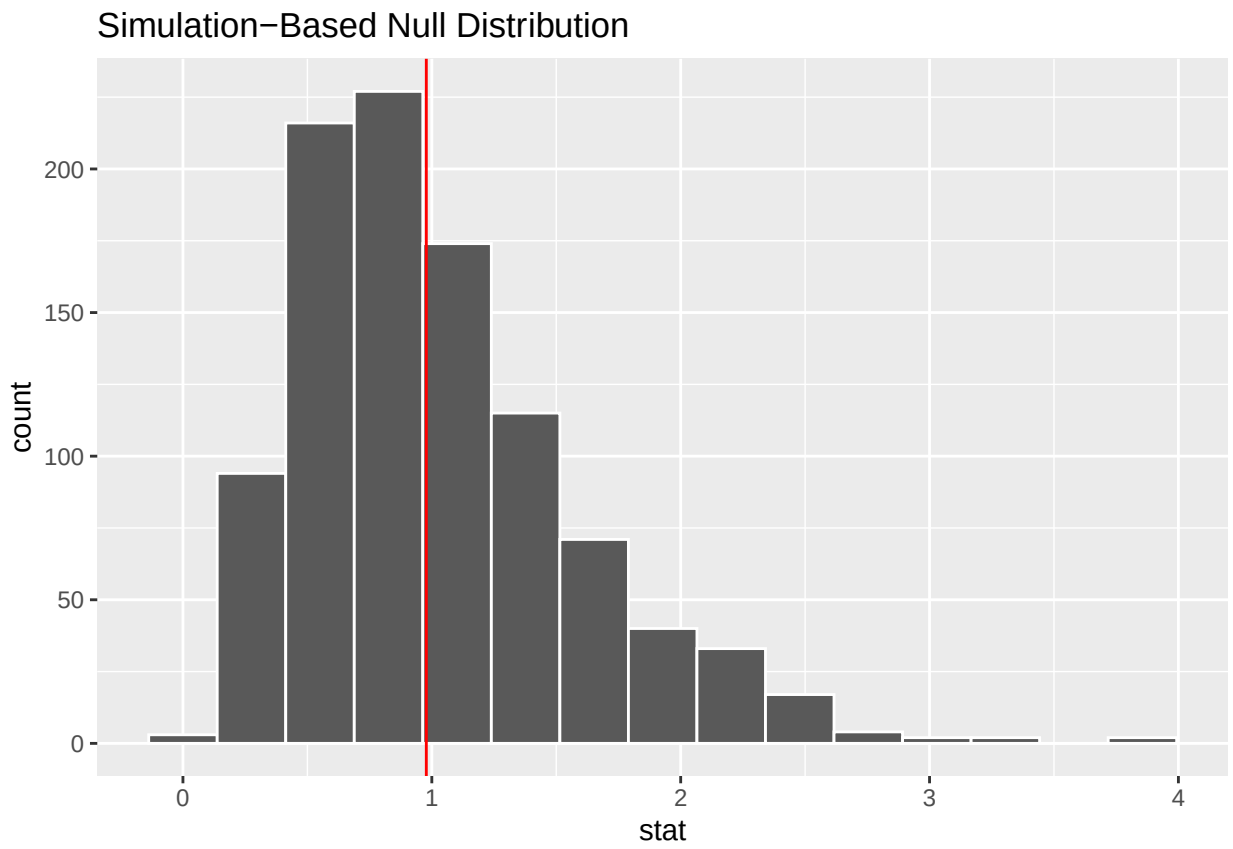
```
## Warning: Removed 129 rows containing missing values.
```

```
## [1] 0.9778619
```

```
rand_dsn <- Women %>%
  specify(time_in_seconds ~ race_year_id) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "F")
```

```
## Warning: Removed 129 rows containing missing values.
```

```
visualize (rand_dsn) %>%
gf_vline(xintercept = ~.97786, color = "red")
```




```
get_pvalue(rand_dsn, obs_stat = .97786, direction = "greater")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1    0.45
```

```
aov(time_in_seconds ~ race_year_id, data = Men)
```

```
## Call:
##    aov(formula = time_in_seconds ~ race_year_id, data = Men)
##
## Terms:
##                 race_year_id    Residuals
## Sum of Squares     880942314 261511515878
## Deg. of Freedom            7         2417
##
## Residual standard error: 10401.77
## Estimated effects may be unbalanced
## 421 observations deleted due to missingness
```

```
Men %>%
  specify(time_in_seconds ~ race_year_id) %>%
  calculate(stat = "F") %>%
  pull()
```

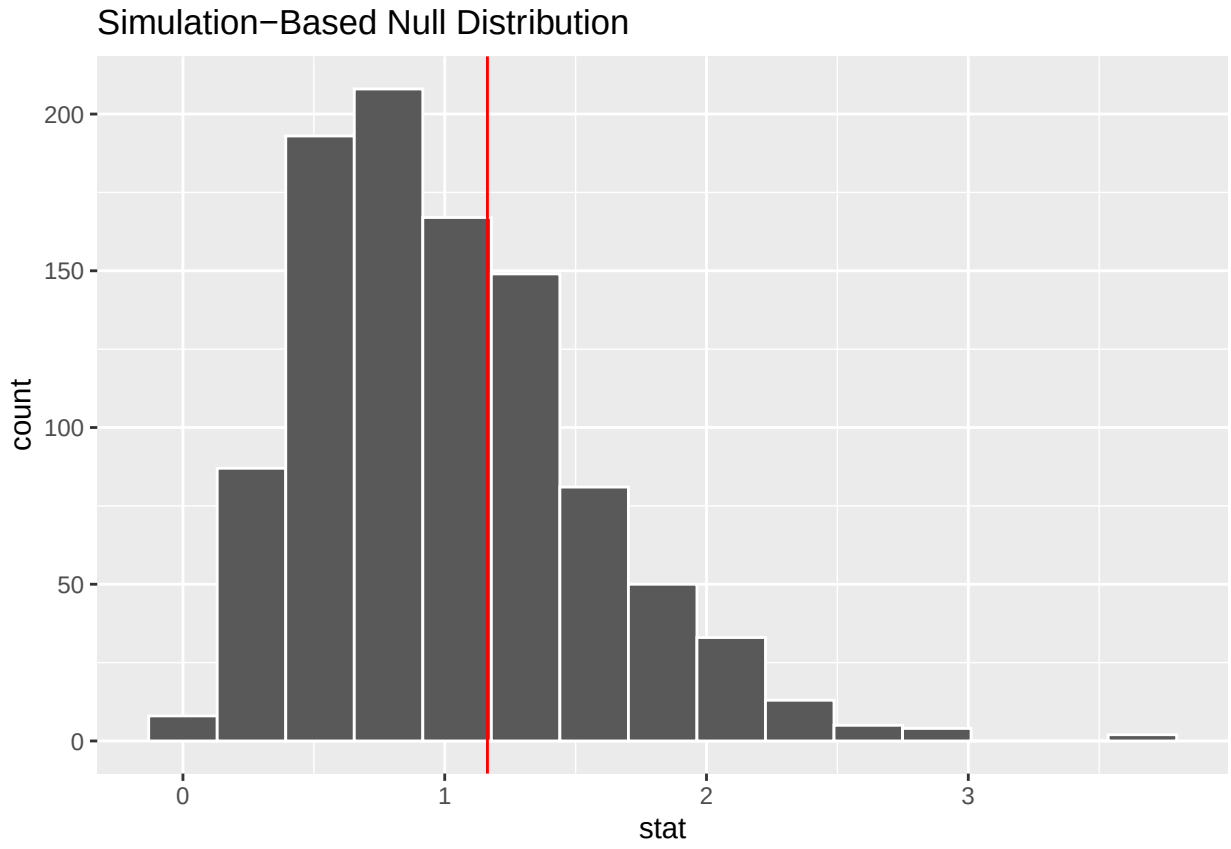
```
## Warning: Removed 421 rows containing missing values.
```

```
## [1] 1.163149
```

```
rand_dsn2 <- Men %>%
  specify(time_in_seconds ~ race_year_id) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "F")
```

```
## Warning: Removed 421 rows containing missing values.
```

```
visualize (rand_dsn2) %>%
gf_vline(xintercept = ~1.163149, color = "red")
```




```
get_pvalue(rand_dsn2, obs_stat = 1.163149, direction = "greater")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.346
```

4.

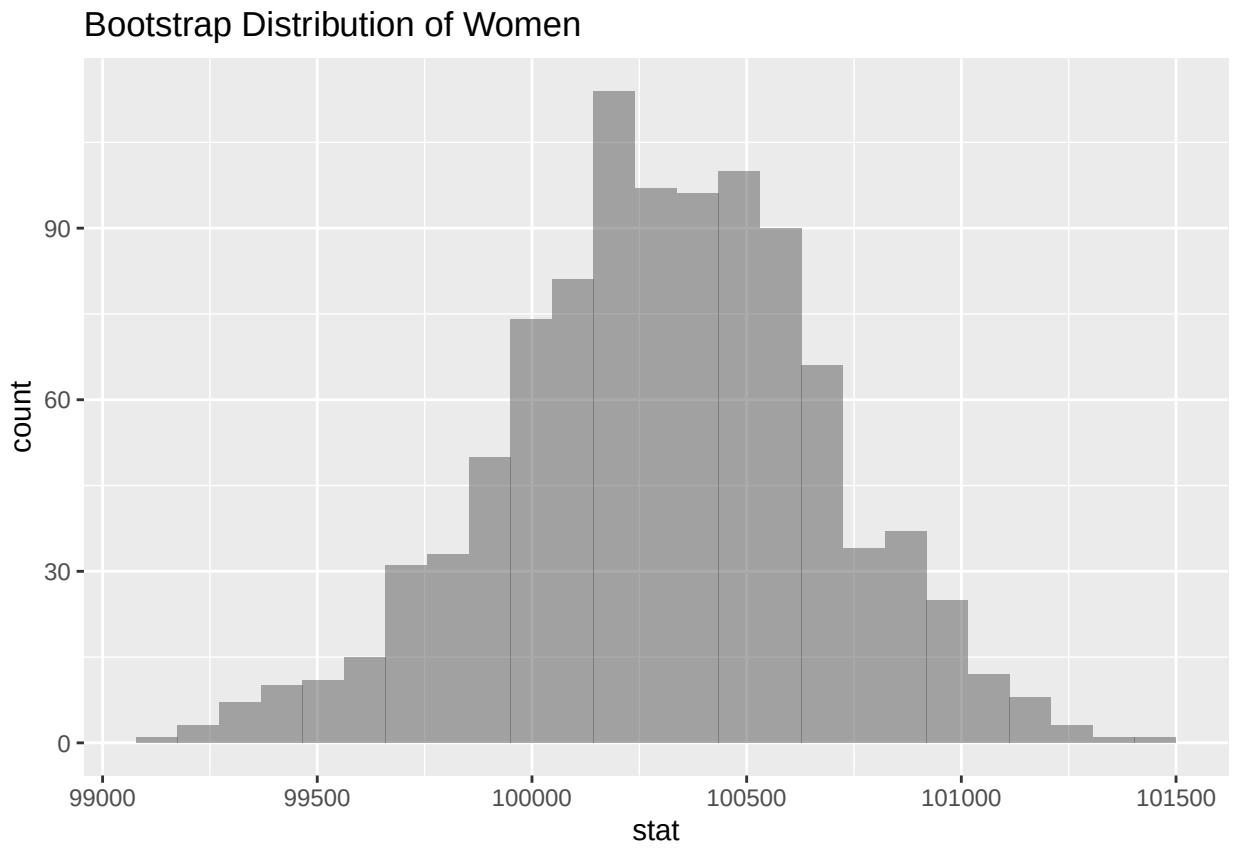
Once we determined there was no evidence to support a statistically discernible difference between mean race times over the years we studies, we thought it would be informative to find a confidence interval for the mean race time over all of the years. The code for this is seen below:

```
boot_dsn <- Women %>%
  specify(response = time_in_seconds) %>%
  generate(reps = 1000, type = "bootstrap") %>%
```

```
  calculate(stat = "mean")
```

```
## Warning: Removed 129 rows containing missing values.
```

```
gf_histogram(~stat, title = "Bootstrap Distribution of Women", data = boot_dsn)
```



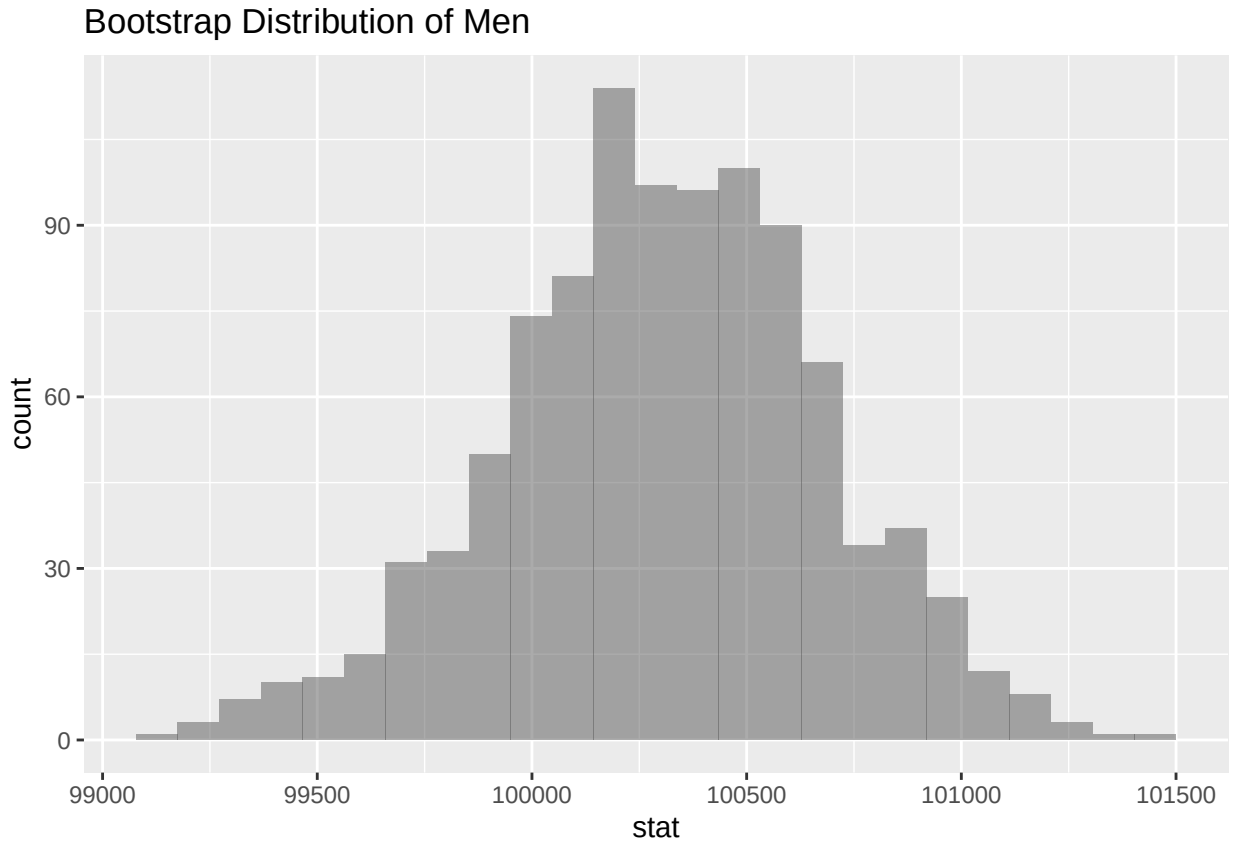

```
get_ci(boot_dsn)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   99495.  101015.
```

```
boot_dsn2 <- Men %>%
  specify(response = time_in_seconds) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "mean")
```

```
## Warning: Removed 421 rows containing missing values.
```

```
gf_histogram(~stat, title = "Bootstrap Distribution of Men", data = boot_dsn)
```




```
get_ci(boot_dsn2)
```

```
## # A tibble: 1 x 2
##   lower_ci upper_ci
##      <dbl>    <dbl>
## 1   96754.   97579.
```