Multiple Sampling

Week 9

Packages needed and a Note about Icons

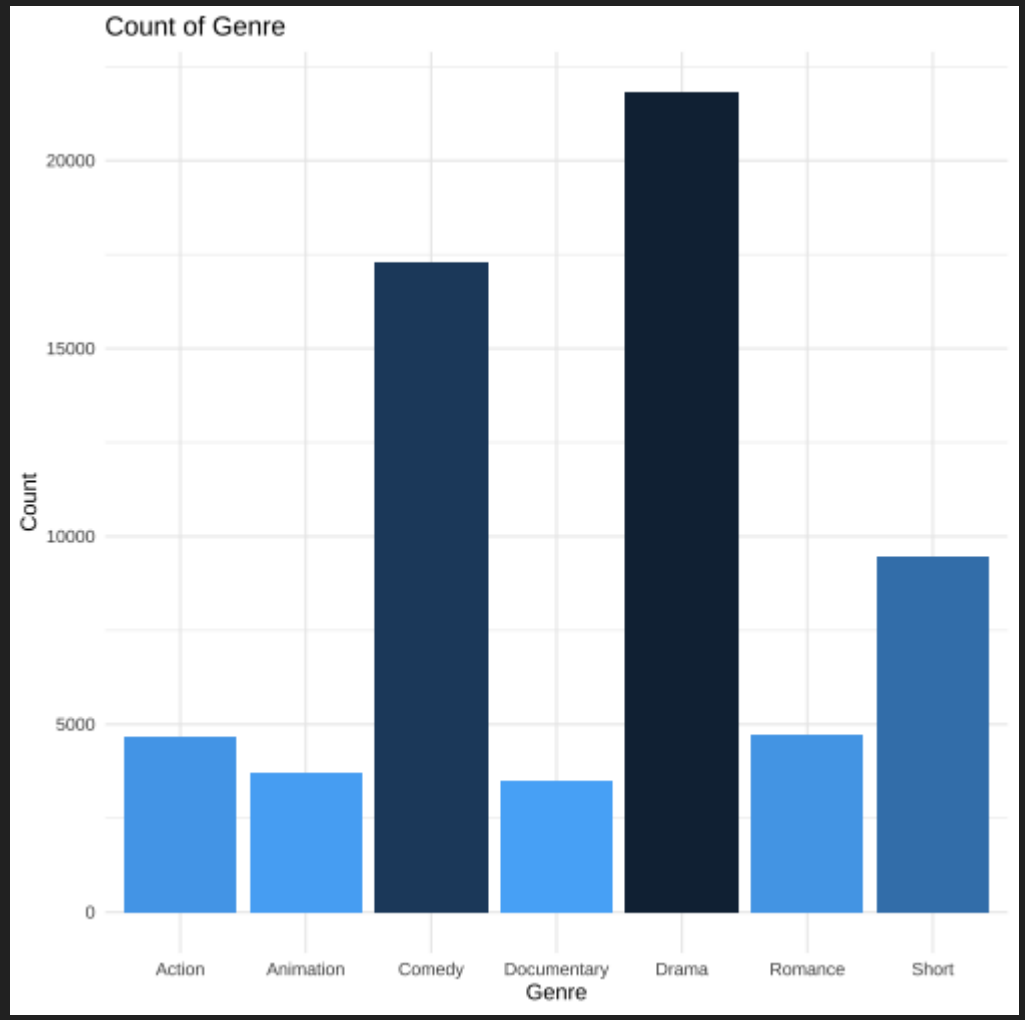
Please load up the following packages. Remember to first install the ones you don't have.

```
library(tidyverse)
library(mosaic)
library(ggplot2movies)
library(viridis)
library(patchwork)
```

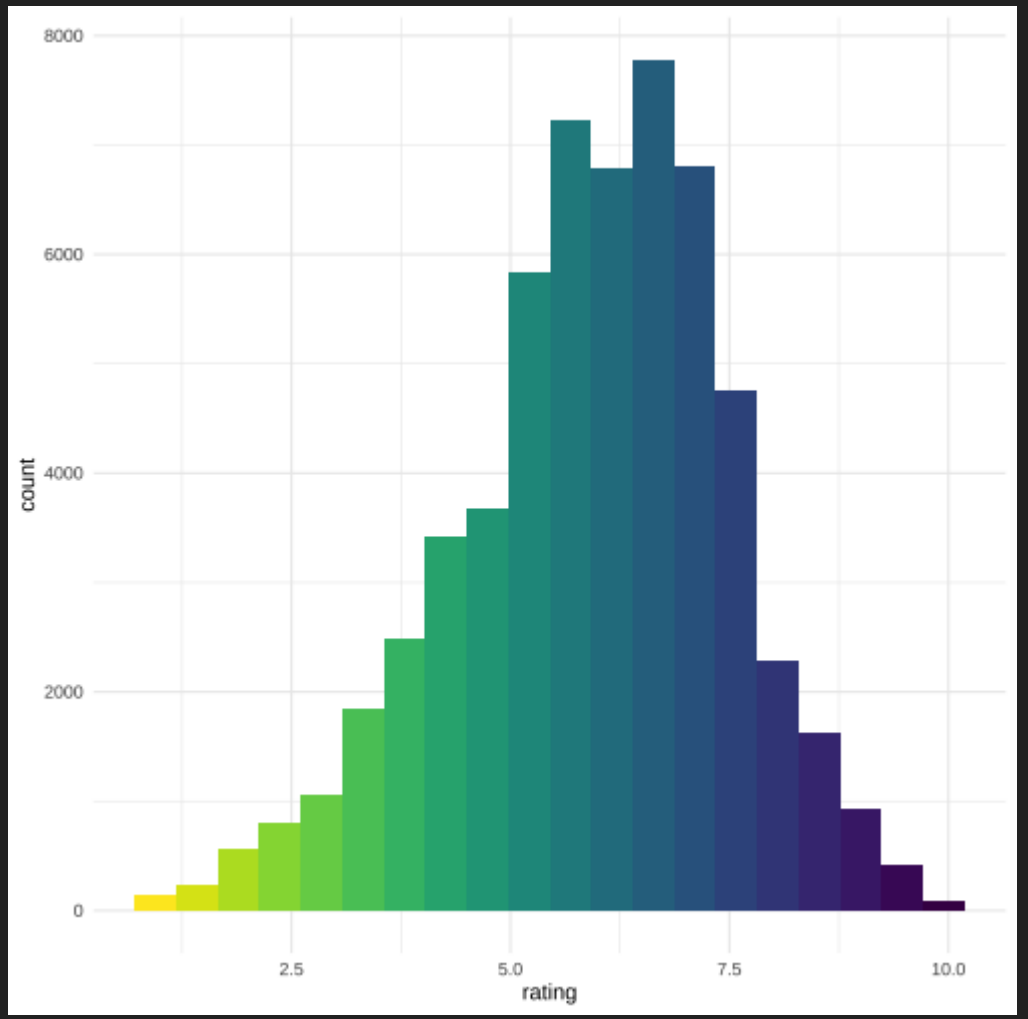
You may come across the following icons. The table below lists what each means.

Icon	Description
	Indicates that an example continues on the following slide.
	Indicates that a section using common syntax has ended.
	Indicates that there is an active hyperlink on the slide.
	Indicates that a section covering a concept has ended.

```r
ggplot2movies::movies %>%
  select(Action, Animation,
         Comedy, Drama,
         Documentary, Romance,
         Short) %>%
  pivot_longer(everything(),
               names_to = "genre") %>%
  group_by(genre) %>%
  tally(value) %>%
ggplot(aes(x = genre,
           y = n,
           fill = -n)) +
  geom_bar(stat='identity',
           show.legend = FALSE) +
  labs(title = "Count of Genre",
       x = "Genre",
       y = "Count") +
  theme_minimal()
```



```
ggplot2movies::movies %>%
  ggplot(aes(x = rating,
             fill = ..x..)) +
  geom_histogram(bins = 20,
                 show.legend = FALSE) +
  scale_fill_viridis(direction = -1) +
  theme_minimal()
```

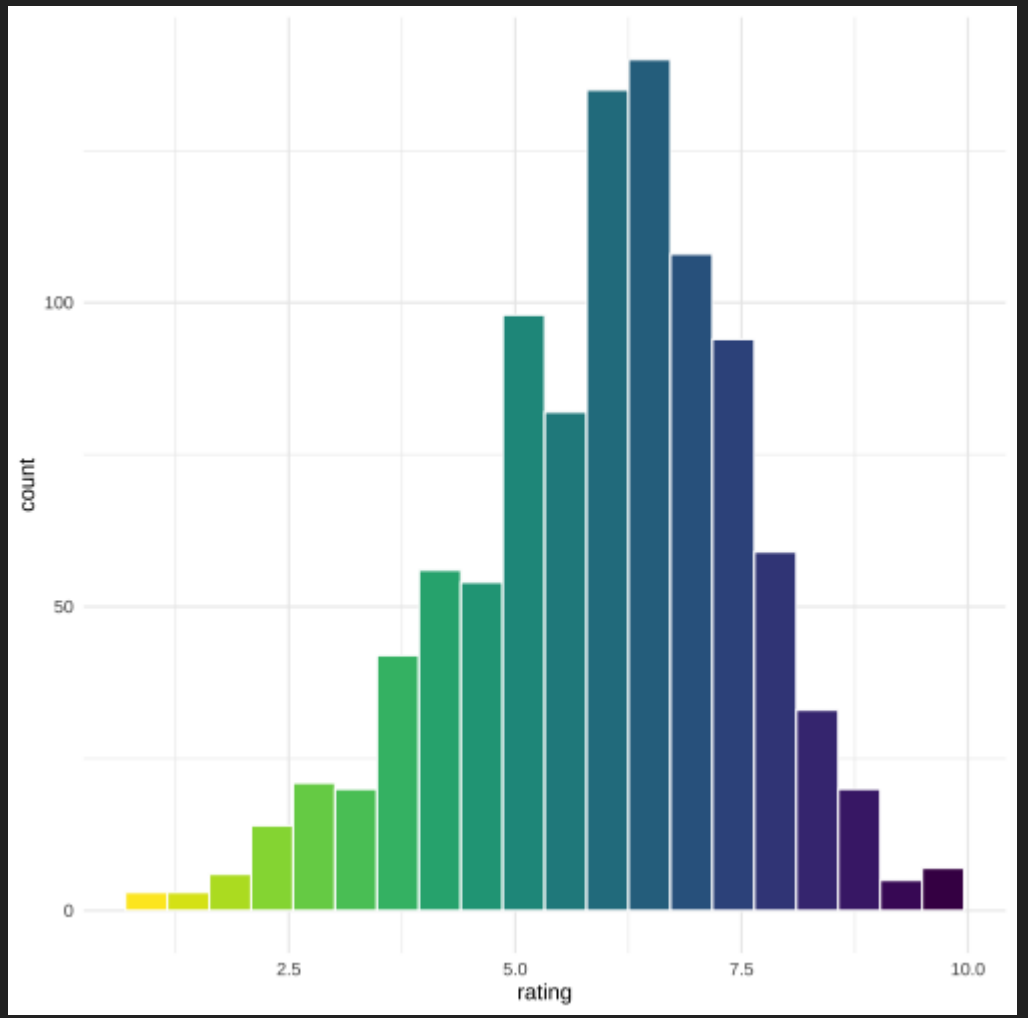


```r
set.seed(123)

ggplot2movies::movies %>%
  sample_n(1000)
```

```
## # A tibble: 1,000 × 24
##    title       year length budget rating votes    r1    r2    r3
##    <chr>      <int>  <int>  <int>  <dbl> <int> <dbl> <dbl> <dbl>
##  1 Thief of …  1952     78 NA         5       8  14.5   0     0
##  2 Yakuza, T…  1975    112 NA         6.9   550   4.5   4.5   4.5
##  3 Aprimi il…  2003     93 NA         4.5    32  14.5   4.5   4.5
##  4 Zendan-e …  2002    106 NA         6.8    52   4.5   0     0
##  5 Lightning…  1994     98 NA         4.8  1020   4.5   4.5   4.5
##  6 Leylasede   1995    100  7  e5      6.6    29   0     0     4.5
##  7 Ojos sini…  1973     81 NA         3      12  34.5   4.5   0
##  8 Another D…  1998    101 NA         6.3  1872   4.5   4.5   4.5
##  9 Sebastian…  1990     88 NA         5.5     7  14.5   0     0
## 10 Shine       1996    105  5.5e6     7.6 12425   4.5   4.5   4.5
## # … with 990 more rows, and 14 more variables: r5 <dbl>, r6 <dbl>,
## #   r7 <dbl>, r8 <dbl>, r9 <dbl>, r10 <dbl>, mpaa <chr>,
## #   Action <int>, Animation <int>, Comedy <int>, Drama <int>,
## #   Documentary <int>, Romance <int>, Short <int>
```

```r
ggplot(movies_sample,
       aes(x = rating,
           fill = ..x..)) +
       geom_histogram(color = "white",
                      bins = 20,
                      show.legend = FALSE)
       scale_fill_viridis(direction = -1) +
       theme_minimal()
```



```
rand_sample <-
  ggplot(movies_sample,
         aes(x = rating,
             fill = ..x..)) +
  geom_histogram(color = "white",
                 bins = 20,
                 show.legend = FALSE) +
  scale_fill_viridis(direction = -1) +
  theme_minimal()
```

```
movies_sample %>%
  summarize(mean = mean(rating))
```

```
## # A tibble: 1 × 1
##    mean
##   <dbl>
## 1  5.97
```

This values is only a single estimation. What you did earlier was top keep sampling from the population, or what is known as **sampling with replacement**.

```
resample(movies_sample) %>%
  arrange(orig.id) %>%
  summarize(mean = mean(rating))
```

```
## # A tibble: 1 × 1
##    mean
##   <dbl>
## 1  5.99
```

But again, this is only one sample mean.

```r
do(10) *
  (resample(ggplot2movies::movies) %>%
     summarize(mean = mean(rating)))
```

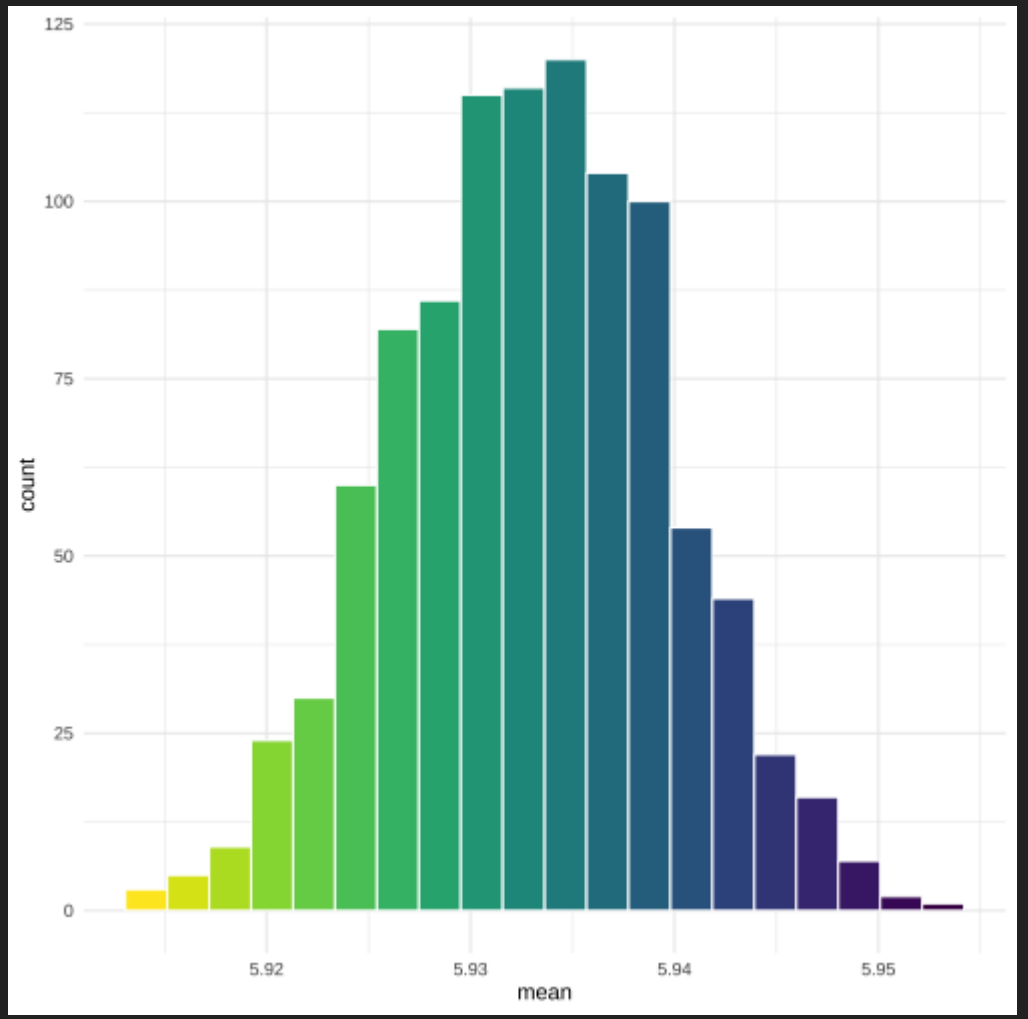
```
##        mean
## 1  5.937487
## 2  5.936739
## 3  5.930409
## 4  5.921472
## 5  5.934427
## 6  5.934444
## 7  5.930503
## 8  5.944598
## 9  5.933162
## 10 5.929152
```

```
do(1000) *
summarize(resample(ggplot2movies::movies)
          mean = mean(rating))
```

```
##          mean
## 1    5.947217
## 2    5.936438
## 3    5.936428
## 4    5.942277
## 5    5.942965
## 6    5.935548
## 7    5.925339
## 8    5.936348
## 9    5.947751
## 10   5.929229
## 11   5.929606
## 12   5.934994
## 13   5.922717
## 14   5.923758
## 15   5.932064
## 16   5.936353
## 17   5.931591
## 18   5.930556
## 19   5.935526
## 20   5.938646
## 21   5.936723
## 22   5.937902
## 23   5.934175
## 24   5.930826
## 25   5.941469
## 26   5.928256
## 27   5.926160
## 28   5.932947
## 29   5.941658
## 30   5.933526
## 31   5.930333
```

```r
not_tiny <-
  do(1000) *
  summarize(resample(ggplot2movies::movies),
            mean = mean(rating))
```

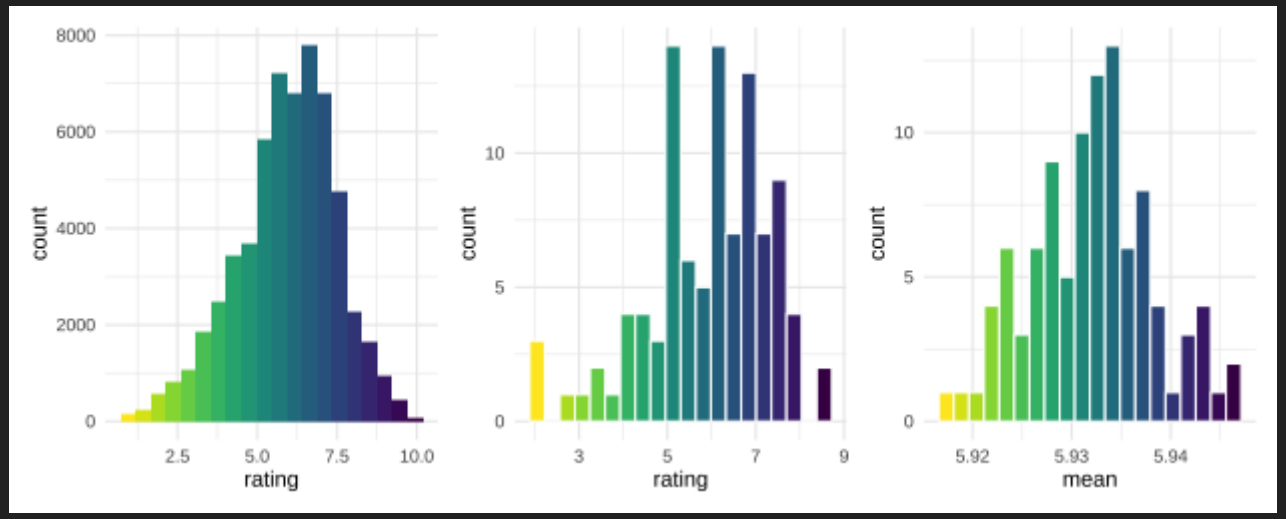
```r
ggplot(not_tiny,
       mapping = aes(x = mean,
                     fill = ..x..)) +
  geom_histogram(bins = 20,
                 color = "white",
                 show.legend = FALSE) +
  scale_fill_viridis(direction = -1) +
  theme_minimal()
```



```
rep_sample <-
  ggplot(data = not_tiny ,
         mapping = aes(x = mean,
                       fill = ..x..)) +
  geom_histogram(bins = 20,
                 color = "white",
                 show.legend = FALSE) +
  scale_fill_viridis(direction = -1) +
  theme_minimal()
```

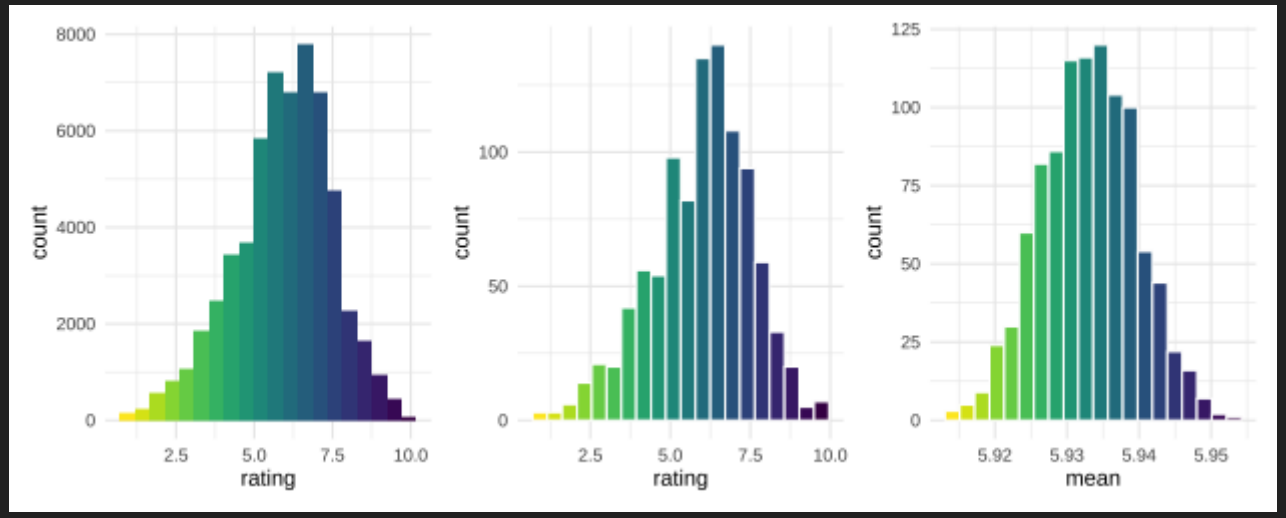
Comparison for $n = 100$

- Left - original population distribution
- Middle - random sample distribution
- Right - repeated sample distribution



Comparison for $n = 1000$

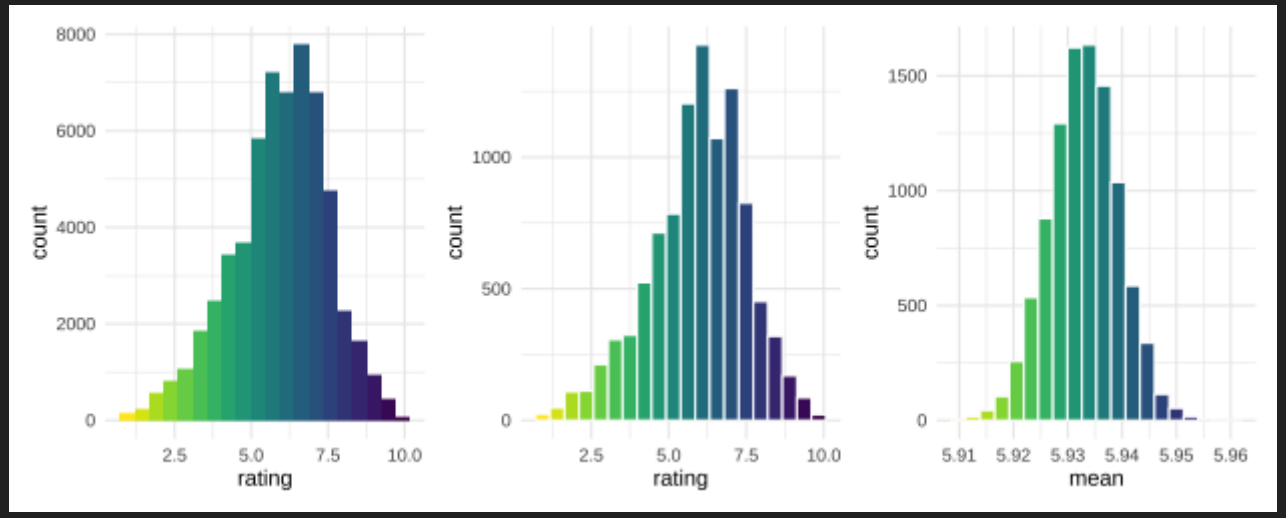
- Left - original population distribution
- Middle - random sample distribution
- Right - repeated sample distribution



The repeated sample distribution does a better job for smaller samples...

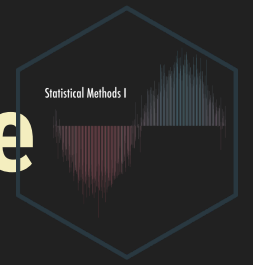
Comparison for $n = 10000$

- Left - original population distribution
- Middle - random sample distribution
- Right - repeated sample distribution



...but oversampling sample means leads to a normal distribution!

Shortcut: Naming and Outputting a Data Frame



There are many approaches that you can take to shorten the number of steps taken to get what you want. While knowing the "long" way gives you the greatest flexibility, if you absolutely know where you're going then knowing shortcuts will make your life easier. Here are two ways to name a data frame and see the output at the same time:

Add parentheses to the entire chunk

```
(way1 <-
    starwars %>%
    head() %>%
    select(name, height, hair_color))
```

```
## # A tibble: 6 × 3
##   name           height hair_color
##   <chr>           <int> <chr>
## 1 Luke Skywalker    172 blond
## 2 C-3PO             167 <NA>
## 3 R2-D2              96 <NA>
## 4 Darth Vader       202 none
## 5 Leia Organa       150 brown
## 6 Owen Lars         178 brown, grey
```

Add a semicolon and name after the chunk

```
way2 <-
    starwars %>%
    head() %>%
    select(name, height, hair_color); way2
```

```
## # A tibble: 6 × 3
##   name           height hair_color
##   <chr>           <int> <chr>
## 1 Luke Skywalker    172 blond
## 2 C-3PO             167 <NA>
## 3 R2-D2              96 <NA>
## 4 Darth Vader       202 none
## 5 Leia Organa       150 brown
## 6 Owen Lars         178 brown, grey
```

Confidence using quantiles

We can now calculate a confidence interval using many options. Let's first isolate the middle 95% of values which corresponds to a 95% confidence interval for the population mean rating.

count: false

```
confint(not_tiny,
        level = 0.95,
        method = "quantile")
```

```
##   name    lower    upper level     method estimate
## 1 mean 5.920575 5.946025  0.95 percentile  5.93285
```

Based on the sample data and bootstrapping techniques, we can be 95% confident that the true mean rating of ALL IMDB ratings is between 5.49 and about 6.13.

Confidence using the standard error

Recall that the **standard error** is the standard deviation of the sampling distribution and is approximated by the bootstrap distribution or the null distribution depending on the context. To do this we can use the same function as before but only by changing the method

count: false

```
confint(not_tiny,
        level = 0.95,
        method = "stderr")
```

```
## Warning: confint: Using df = Inf.

##   name    lower    upper level method estimate margin.of.error
## 1 mean 5.920053 5.945945  0.95 stderr  5.93285      0.01294608
```

The interpretation is virtually the same here.

Thats it!