Chapter Summary of ...

Your Name Here

Last modified on January 13, 2024 08:23:30 Eastern Standard Time

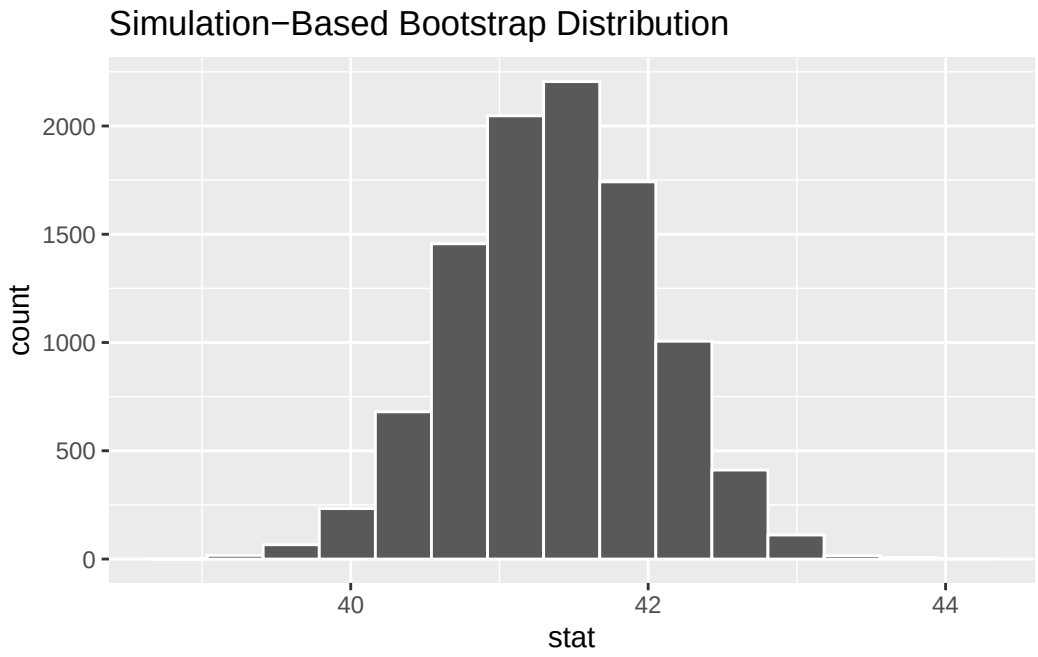
Bootstrapping to estimate a single parameter

- Bootstrapping works with complicated parameters.
- Bootstrap estimate of a parameter is no better than the "standard" estimate of the parameter.
- We bootstrap to get an idea of the "standard error" - bootstrap standard error/deviation.
- Bootstrapping is taking repeated samples of the same size as the original sample with replacement.

Bootstrapping code

- Can use the `infer` package (hides all the work)
- Type `?infer` at the R prompt for help file.
- Should read Getting to Know `infer`

```
# Code
library(infer)
gss %>%
  specify(response = hours) %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "mean") -> bs_dist
visualize(bs_dist) # visualize the bootstrap distribution
```

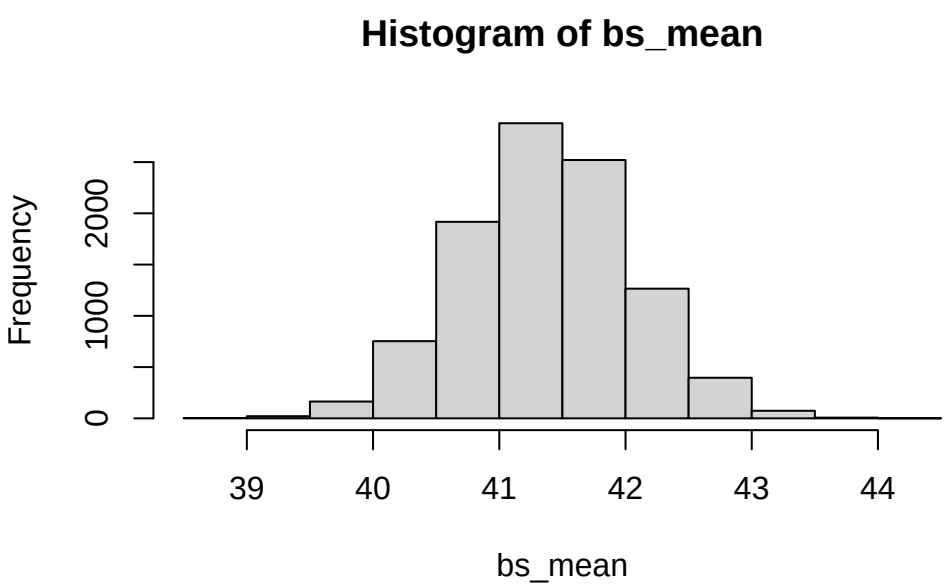



```
# Compute a 90% Bootstrap Percentile CI
get_confidence_interval(bs_dist, level = 0.90, type = "percentile")
```

```
# A tibble: 1 x 2
  lower_ci upper_ci
     <dbl>    <dbl>
1     40.3     42.5
```

- Can use a basic `for()` loop to generate bootstrap samples.

```
library(infer)
B <- 10^4
bs_mean <- numeric(B)
for(i in 1:B){
  bss <- sample(gss$hours, size = 500, replace = TRUE)
  bs_mean[i] <- mean(bss)
}
hist(bs_mean)
```




```
quantile(bs_mean, probs = c(0.05, 0.95))
```

```
     5%     95%
40.2799 42.4801
```

Testing a hypothesis with bootstrapping

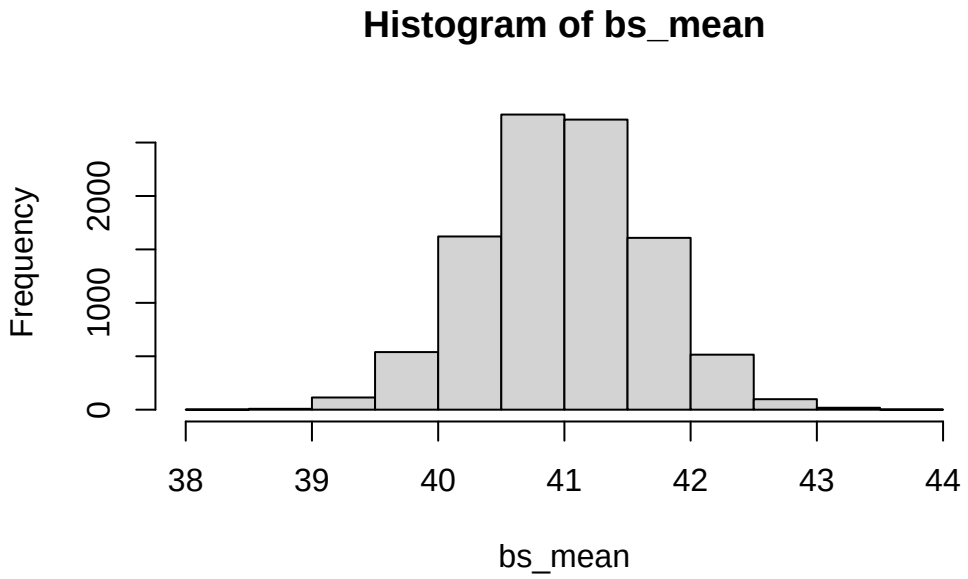
- Must make the bootstrap distribution conform to the null hypothesis (Suppose we want to test $H_0 : \mu = 41$ versus $H_A : \mu > 41$)

```
library(infer)
mean(gss$hours) # In order for the null to be true,
```

```
[1] 41.382
```

```
# need to subtract 0.382 from every value in hours.
B <- 10^4
bs_mean <- numeric(B)
for(i in 1:B){
```

```
  bss <- sample(gss$hours, size = 500, replace = TRUE) - 0.382
  bs_mean[i] <- mean(bss)
}
hist(bs_mean)
```




```
pvalue <- mean(bs_mean >= mean(gss$hours))
pvalue
```

```
[1] 0.2802
```

```
library(infer)
gss %>%
  specify(response = hours) %>%
  hypothesize(null = "point", mu = 41) %>%
  generate(reps = 10000, type = "bootstrap") %>%
  calculate(stat = "mean") -> boot_test
get_p_value(boot_test, mean(gss$hours), direction = "right")
```

```
# A tibble: 1 x 1
```

```
  p_value
    <dbl>
1   0.282
```