

Practice Session 5

In this practice section we will introduce the bootstrap distribution, bootstrap confidence interval, and hypothesis testing. You may use the functions: `sample()`, `do_it` to create the bootstrap distribution.

Part 1: Confidence interval concept

Practice 1.a: True or False/ Confidence interval interpretation

A catalog sales company promises to deliver orders placed on the Internet within 3 days. Follow-up calls to a few randomly selected customers show that a 95% confidence interval for the proportion of all orders that arrive on time is $85\% \pm 5\%$.

- 1.) Between 80% and 90% of all orders arrive on time.
- 2.) Ninety five percent of all random samples of customers will show that 85% of orders arrive on time.
- 3.) The interval between 80% and 90% gives a plausible range of values for where the true population parameter lies since 95% of intervals created will contain the population proportion.

Answers:

- 1.) Not correct. This statement implies certainty. There is no level of confidence in the statement.
- 2.) Not correct. Different samples will give different results. Many fewer than 95% of samples are expected to have exactly 88% on-time orders.
- 3.) Correct.

Practice 1.b: True or False/ confidence level, Sample size, and confidence interval

Several factors are involved in the creation of a confidence interval. Among them are the sample size, the level of confidence, and the margin of error.

- 1.) For a given sample size, higher confidence means a larger margin of error.
- 2.) For a specified confidence level, smaller samples provide smaller margins of error.
- 3.) For a fixed margin of error, larger samples provide greater confidence.

Answers:

- 1.) True. For a given sample size, higher confidence means a larger margin of error.
- 2.) False. Smaller samples lead to larger standard errors, which lead to larger margins of error.
- 3.) True. Larger samples are less variable, which makes us more confident that a given confidence interval succeeds in catching the population proportion.

Part 2: Construct Bootstrap Distribution

Here's the clever idea: We don't have the population, but we have a sample. Probably the sample is similar to the population in many ways. So let's sample from our sample. We'll call it **bootstrapping**. We want samples **the same size** as our original sample, so we will need to **sample with replacement**. This means that we may pick some members of the population more than once and others not at all. We'll replicate this many times.

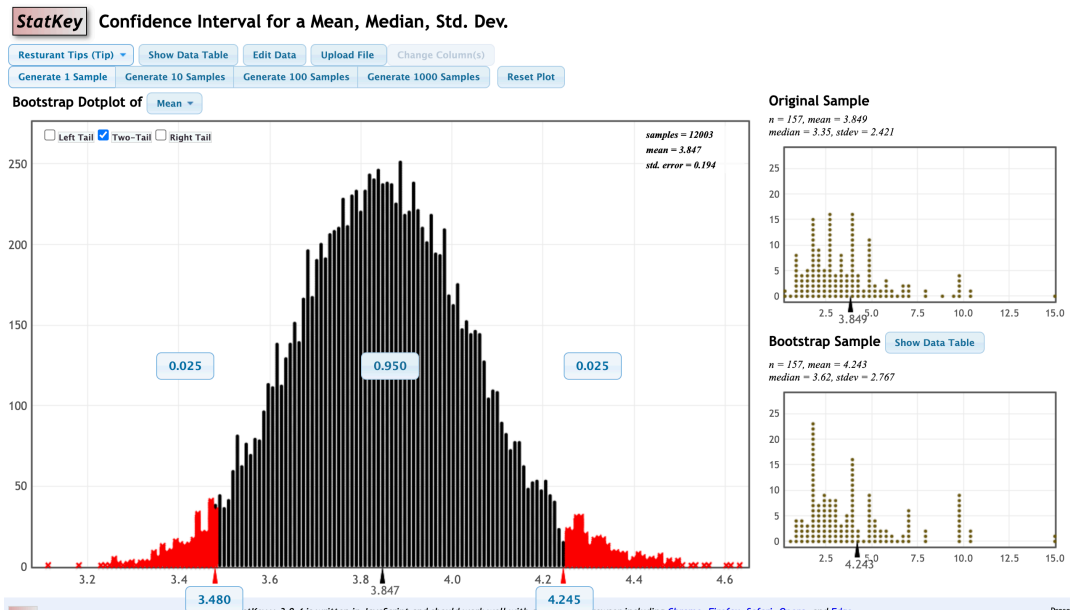
Generating a Bootstrap Distribution:

- Generate bootstrap samples by **sampling with replacement** from the original sample, using the **same sample size**.
- Compute the **statistic of interest** (called a bootstrap statistic), for each of the bootstrap samples.
- Collect the **samples statistics** for many bootstrap samples to create a **bootstrap distribution**.

Example:

Using StatKey [link](https://www.lock5stat.com/StatKey/), try to play with the creation of the bootstrap distribution from different data. The following picture shows the website and an example of bootstrap CI for the variable `tip` from the data `Restaurant tips`.

https://www.lock5stat.com/StatKey/bootstrap_1_quant/bootstrap_1_quant.html



Practice 2:

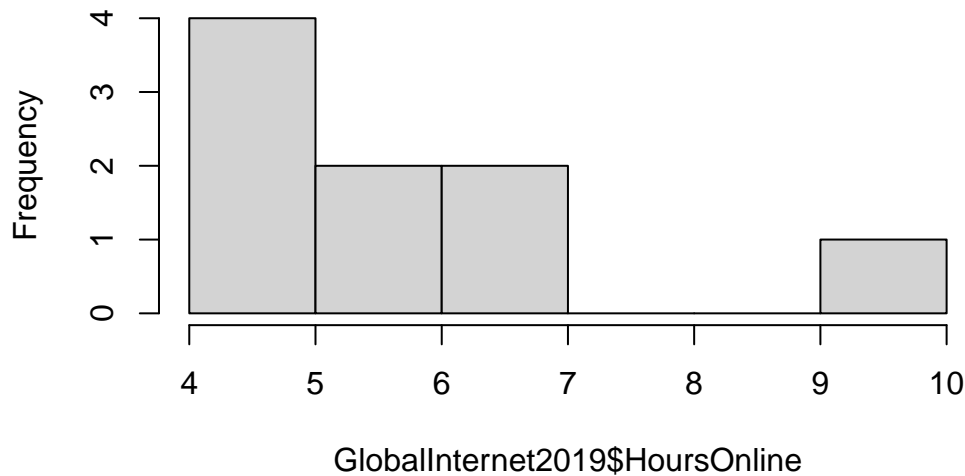
The data GlobalInternet2019 provide the average number of hours spent online for countries.

- 1) First, create histogram of HoursOnline. What is the sample size of the data ?

```
library(Lock5Data)
library(SDS1000)
data(GlobalInternet2019)

n <- length(GlobalInternet2019$HoursOnline)
hist(GlobalInternet2019$HoursOnline)
```

Histogram of GlobalInternet2019\$HoursOnline



- 2) Second, create a one bootstrap sample from HoursOnline . You might use the functions sample.

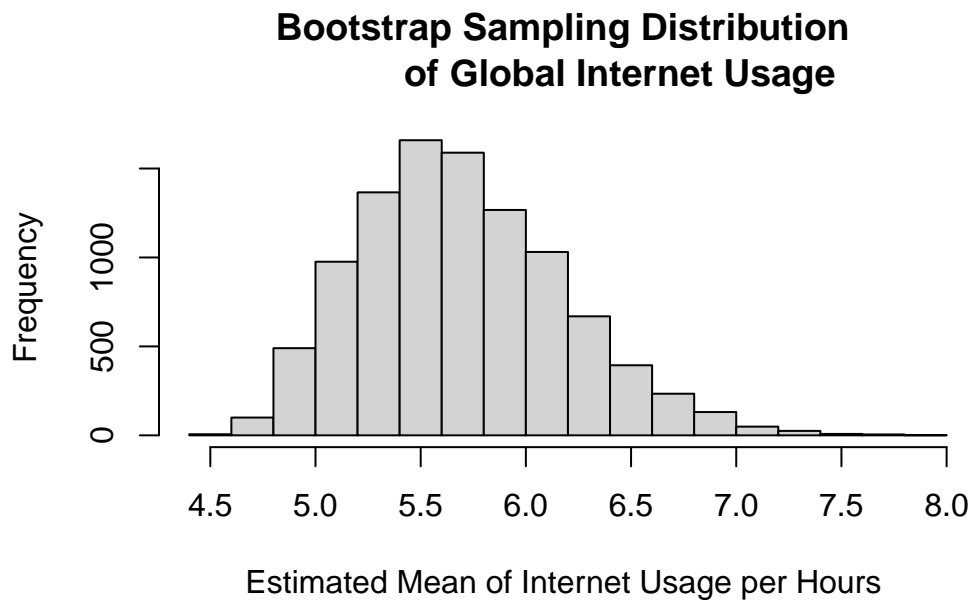
```
one_sample <- sample(GlobalInternet2019$HoursOnline,  
                      size = n, replace = T)
```

- 3) Third, you might replicate this one sample 10000 times with replacement using the function do_it() to create a bootstrap sampling ditribution

```
boot_dist <- do_it(10000) * {  
  one_sample = sample(GlobalInternet2019$HoursOnline, size = n, replace = T)  
  mean( one_sample)  
}
```

- 4) Fourth, create a histogram for the bootstrap distribution of the variable HoursOnline

```
hist(boot_dist, main = "Bootstrap Sampling Distribution  
of Global Internet Usage ",  
     xlab = "Estimated Mean of Internet Usage per Hours ")
```



5) congrats!, you created bootstrap sampling distribution from one sample !

Part 3: Construct Bootstrap Confidence Interval

The steps to Construct Bootstrap Confidence Interval are:

1. Compute the statistic from the original sample.
2. Create a bootstrap distribution by resampling from the sample.
 - same size samples as the original sample
 - with replacement
 - compute the statistic for each sample. The distribution of these statistics is the bootstrap distribution.
3. Estimate the standard error SE by computing the standard deviation of the bootstrap distribution.
4. 95% CI is: $statistic \pm 2 * SE$

Practice 3:

From the `GlobalInternet2019` bootstrap sampling distribution you have created in the previous question, create a 95% CI for the the sample mean of `Distance`.

1.) Calculate the mean of HoursOnline from your original sample

```
x_bar <- mean(GlobalInternet2019$HoursOnline )  
  
x_bar
```

```
[1] 5.693333
```

2.) Calculate the standard error of our bootstrap sampling distribution of the HoursOnline

```
boot_se <- sd(boot_dist )  
  
boot_se
```

```
[1] 0.4869169
```

3.) Calculate the 95% CI, which is based on the formula: $statistic \pm 2 * SE$

```
#CI_lower <- x_bar - 2*boot_se  
#CI_upper <- x_bar + 2*boot_se  
  
CI95_lower_upper <- x_bar+2*boot_se*c(-1,1)  
  
CI95_lower_upper
```

```
[1] 4.719500 6.667167
```

4) Calculate the confidence interval of 99% using the formula

```
# 99\% has different critical value than 95%  
# 0.99+(1-0.99)/2  
  
cv_99 <- qnorm(0.995)  
  
CI99_lower_upper <- x_bar+(cv_99)*boot_se*c(-1,1)  
  
CI99_lower_upper
```

```
[1] 4.439119 6.947548
```

Part 4: Create Bootstrap Confidence Interval with Percentiles

Practice 4:

- 1) Let create a 95% , 90% CI of the mean HoursOnline in R, using **the percentiles** which they can be calculated using `quantile` function.

```
CI_95_lower_upper <- quantile(boot_dist, c(0.025, 0.975))
CI_95_lower_upper
```

```
      2.5%      97.5%
4.888889 6.763333
```

```
CI_90_lower_upper<- quantile(boot_dist, c(0.05, 0.95))
CI_90_lower_upper
```

```
      5%      95%
4.976667 6.567833
```

Part 5: Hypothesis testing

Statistical Tests:

A statistical test is used to determine whether results from a sample are convincing enough to allow us to conclude something about the population.

We have two competing claims about the population, the **null hypothesis**, denoted by H_0 , and the **alternative hypothesis**, denoted by H_a .

Practice 5:

State the null and alternative hypotheses for the statistical test described:

- 1.) Testing to see if there is evidence that a mean is less than 50.
- 2.) Testing to see if there is evidence that a proportion is greater than 0.3.
- 3.) Testing to see if there is evidence that the mean of group A is not the same as the mean of group B.
- 4.) Testing to see if there is evidence that the correlation between two variables is positive.

Answers:

1. $H_0 : \mu = 50$ vs $H_a : \mu < 50$
2. $H_0 : \pi = 0.3$ vs $H_a : \pi > 0.3$
3. $H_0 : \mu_A = \mu_B$ vs $H_a : \mu_A \neq \mu_B$
4. $H_0 : \rho = 0$ vs $H_a : \rho > 0$