STAT 510: Homework 05

David Dalpiaz

Due: Monday, February 28, 11:59 PM

Exercise 1 (Simulating a Sampling Distribution)

Perform three simulation studies:

1. $n = 5$
2. $n = 25$
3. $n = 50$

For each sample size, generate data from an exponential distribution with a rate parameter of 2. (That is, a mean of 0.5.)

$$X_1, \ldots, X_n \sim \text{Exp}(\lambda = 2)$$

We are interested in the sampling distribution of

$$\hat{\theta} = \bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

For each study, perform at least 1000 simulations, each time storing the value of the estimator. Plot a histogram of the simulated estimator values and overlay the approximate large sample distribution of the estimator, that is

$$\bar{X} \approx N\left(\mathbb{E}[\mathbb{X}], \frac{\mathbb{V}[X]}{n}\right)$$

Your final answer should be three side-by-side histograms, each with an overlay of a density.

- Note: This exercise is hinting at how large n needs to be to approximate the sampling distribution of the sample mean with a normal distribution as suggested by our discussions of the CLT.

Solution

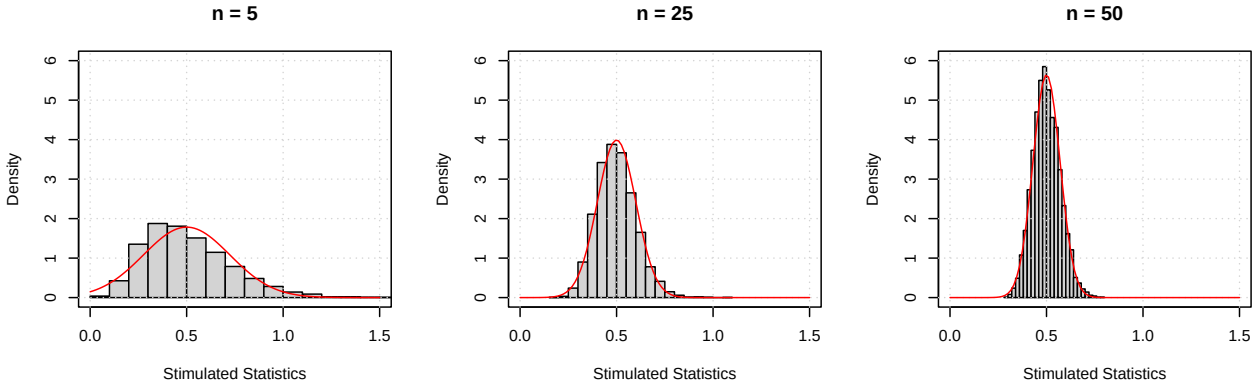
```
set.seed(42)
sim_stats_05 = replicate(n = 5000, mean(rexp(n = 05, rate = 2)))
sim_stats_25 = replicate(n = 5000, mean(rexp(n = 25, rate = 2)))
sim_stats_50 = replicate(n = 5000, mean(rexp(n = 50, rate = 2)))
```

```
make_histogram = function(data, title, n) {
  hist(data, probability = TRUE, main = title, breaks = 25,
       xlab = "Stimulated Statistics", ylim = c(0, 6), xlim = c(0, 1.5))
  box()
  grid()
```

```
  curve(dnorm(x, mean = 0.5, sd = sqrt(0.25 / n)), add = TRUE, col = "red")
}
```

```
par(mfrow = c(1, 3))

make_histogram(sim_stats_05, title = "n = 5",  n =  5)
make_histogram(sim_stats_25, title = "n = 25", n = 25)
make_histogram(sim_stats_50, title = "n = 50", n = 50)
```



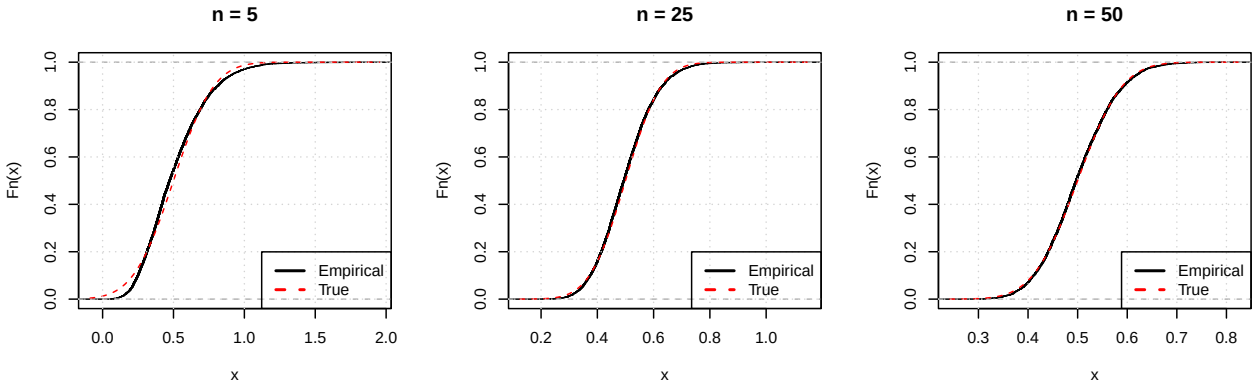
Exercise 2 (How Large is Large?)

Return to your results from Exercise 1. Instead of plotting histograms with a density, plot the empirical CDF and the approximate normal CDF for each of the simulation studies. Based on these results, comment on which of these values of n you would feel comfortable using the normal distribution as an approximation for the true sampling distribution.

Solution

```
make_cdf_plot = function(data, title, n) {
  plot(ecdf(data), main = title)
  curve(pnorm(x, mean = 0.5, sd = sqrt(0.25 / n)), add = TRUE,
        col = "red", lty = 2)
  grid()
  legend("bottomright", legend = c("Empirical", "True"),
         lty = c(1, 2), col = c("black", "red"), lwd = 2)
}
```

```
par(mfrow = c(1, 3))
make_cdf_plot(sim_stats_05, title = "n = 5",  n =  5)
make_cdf_plot(sim_stats_25, title = "n = 25", n = 25)
make_cdf_plot(sim_stats_50, title = "n = 50", n = 50)
```



It is still somewhat hard to say for sure, but it appears that a sample size of 5 is too small in this situation. With a sample size of 25, it seems reasonable to use the approximation.

Exercise 3 (Distribution of a Bootstrap Sample)

Let $X_1, X_2, \ldots, X_n$ be distinct observations, that is, no ties. Let $X_1^\star, X_2^\star, \ldots, X_n^\star$ denote a bootstrap sample and let

$$\bar{X}_n^\star = \frac{1}{n}\sum_{i=1}^{n} X_i^\star.$$

Find:

- $\mathbb{E}\left[\bar{X}_n^\star \mid X_1, X_2, \ldots, X_n\right]$
- $\mathbb{V}\left[\bar{X}_n^\star \mid X_1, X_2, \ldots, X_n\right]$
- $\mathbb{E}\left[\bar{X}_n^\star\right]$
- $\mathbb{V}\left[\bar{X}_n^\star\right]$

Solution

For ease of notation, define

$$\mathbb{E}[X] = \mu, \quad \mathbb{V}[X] = \sigma^2$$

Next, given $X_1, X_2, \ldots, X_n$, the distribution of $X^\star$ is uniform over $X_1, X_2, \ldots, X_n$.

Then, by definition of expected value, we have

$$\mathbb{E}[X^\star \mid X_1, X_2, \ldots, X_n] = \frac{1}{n}X_1 + \ldots + \frac{1}{n}X_n = \bar{X}$$

$$\mathbb{E}\left[\bar{X}^\star \mid X_1, X_2, \ldots, X_n\right] = \frac{1}{n} \cdot n \cdot \bar{X} = \boxed{\bar{X}}$$

$$\mathbb{E}\left[\bar{X}^\star\right] = \mathbb{E}\left[\mathbb{E}\left[\bar{X}^\star \mid X_1, X_2, \ldots, X_n\right]\right] = \mathbb{E}[\bar{X}] = \boxed{\mu}$$

Note that the first expectation is over the randomness of the sample, while the second is over the randomness of the re-sampling.

Moving on the variance, again, by definition we have

$$\mathbb{V}[X^\star \mid X_1, X_2, \ldots, X_n] = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2$$

$$\mathbb{V}\left[\bar{X}^\star \mid X_1, X_2, \ldots, X_n\right] = \boxed{\frac{1}{n^2}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2} = \boxed{\frac{n-1}{n^2}S^2}$$

$$\begin{aligned}
\mathbb{V}\left[\bar{X}^\star\right] &= \mathbb{E}\left[\mathbb{V}\left[\bar{X}^\star \mid X_1, X_2, \ldots, X_n\right]\right] + \mathbb{V}\left[\mathbb{E}\left[\bar{X}^\star \mid X_1, X_2, \ldots, X_n\right]\right] \\
&= \mathbb{E}\left[\frac{n-1}{n^2}S^2\right] + \mathbb{V}\left[\bar{X}\right] \\
&= \frac{n-1}{n^2}\sigma^2 + \frac{\sigma^2}{n} \\
&= \boxed{\frac{\sigma^2}{n} \cdot \left(2 - \frac{1}{n}\right)}
\end{aligned}$$

Exercise 4 (Professor Salaries)

The following code loads data about Professor salaries. (Check the documentation for details.) We will be interested in the `salary` variable.

```
salaries = carData::Salaries
```

Define $\theta = T(F) = q_{0.25}$. Create a 95% confidence interval for the 25th percentile of professor salaries using each of the three bootstrap interval methods: Normal, Pivotal, Percentile.

Use as least 2000 bootstrap samples for each interval.

- Note 1: To obtain a "plug-in" estimate for q_p you may simply use the default arguments to R's `quantile()` function.
- Note 2: These salaries are a few years old, but for **Tenure Track** faculty. Not all of your instructors fall into this category.
- Fun Fact: Illinois is a state institution, so salary information is public. We leave it as an exercise to the read to find this data.

Solution

```
# helper function for plug-in estimator of variance
calc_var = function(x) {
  mean((x - mean(x)) ^ 2)
}
```

```
theta_hat = quantile(salaries$salary, probs = 0.25)
```

```
boot_reps = replicate(
  n = 2000,
  quantile(sample(salaries$salary, replace = TRUE), probs = 0.25)
)
```

```
se = sqrt(calc_var(boot_reps))
```

```
norm_int = theta_hat + c(-1, 1) * qnorm(0.975) * se
pivo_int = 2 * theta_hat - quantile(boot_reps, c(0.975, 0.025))
perc_int = quantile(boot_reps, c(0.025, 0.975))
```

```r
results = data.frame(
  "Normal" = norm_int,
  "Pivotal" = pivo_int,
  "Percentile" = perc_int,
  row.names = c("Lower", "Upper")
)
```

```r
knitr::kable(results)
```

	Normal	Pivotal	Percentile
Lower	88380.5	88582	88600
Upper	93619.5	93400	93418

Exercise 5 (How Long Will You Survive Cancer?)

For this exercise we will use the `Melanoma` data from the `MASS` package.

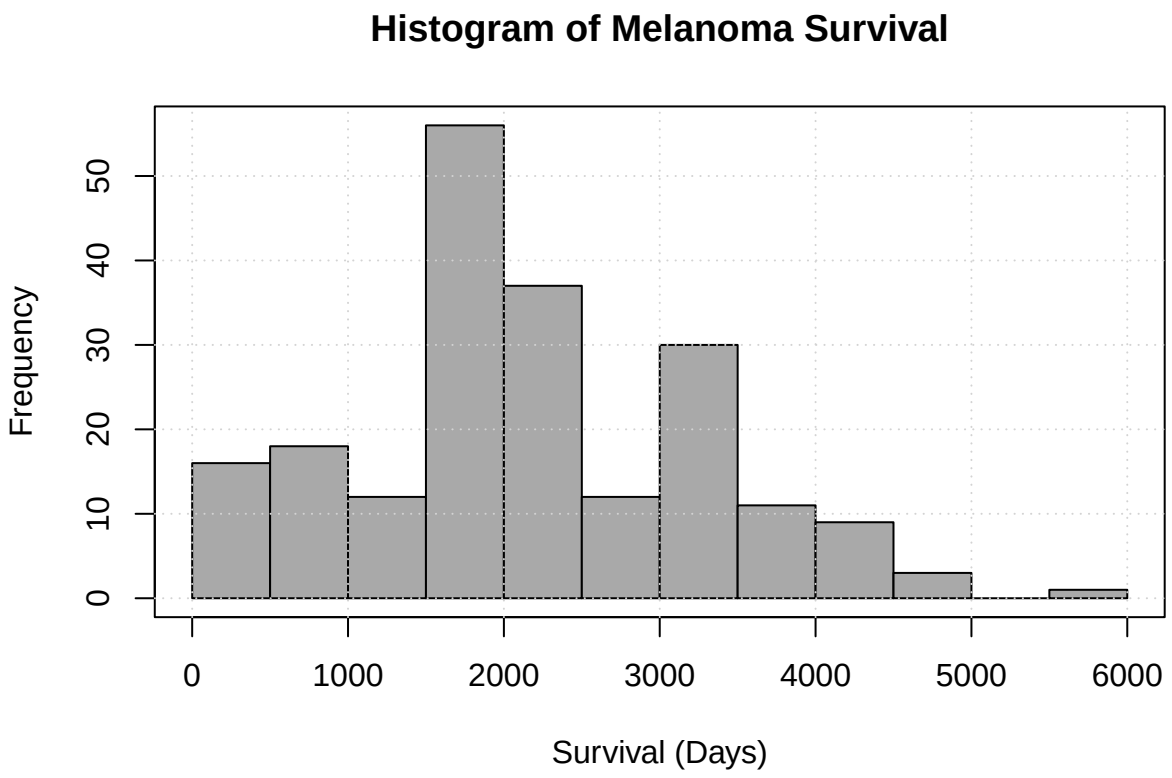
```r
head(MASS::Melanoma)
```

```
##   time status sex age year thickness ulcer
## 1   10      3   1  76 1972      6.76     1
## 2   30      3   1  56 1968      0.65     0
## 3   35      2   1  41 1977      1.34     0
## 4   99      3   0  71 1968      2.90     0
## 5  185      1   1  52 1965     12.08     1
## 6  204      1   1  28 1971      4.84     1
```

We'll focus on the `time` variable which is survival time in days.

```r
mel_survive = MASS::Melanoma$time
```

```r
hist(mel_survive, col = "darkgrey",
     xlab = "Survival (Days)", main = "Histogram of Melanoma Survival")
box()
grid()
```



Let X be the survival time in **years** and define

$$\theta = T(F) = P(X > 5).$$

Create a 95% percentile bootstrap confidence interval for θ, the probability of surviving longer that 5 years. Use at least 20000 bootstrap samples. Also plot a histogram of the bootstrap replicates and overlay the large-sample approximate **estimated** sampling distribution.

Solution:

```
# generate bootstrap replicates
boot_reps = replicate(
  n = 20000,
  mean(sample(mel_survive, replace = TRUE) > (5 * 365))
)
```

```
# calculate estimate given data
theta_hat = mean(mel_survive > (5 * 365))
```

```
# calculate interval
quantile(boot_reps, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.5268293 0.6634146
```

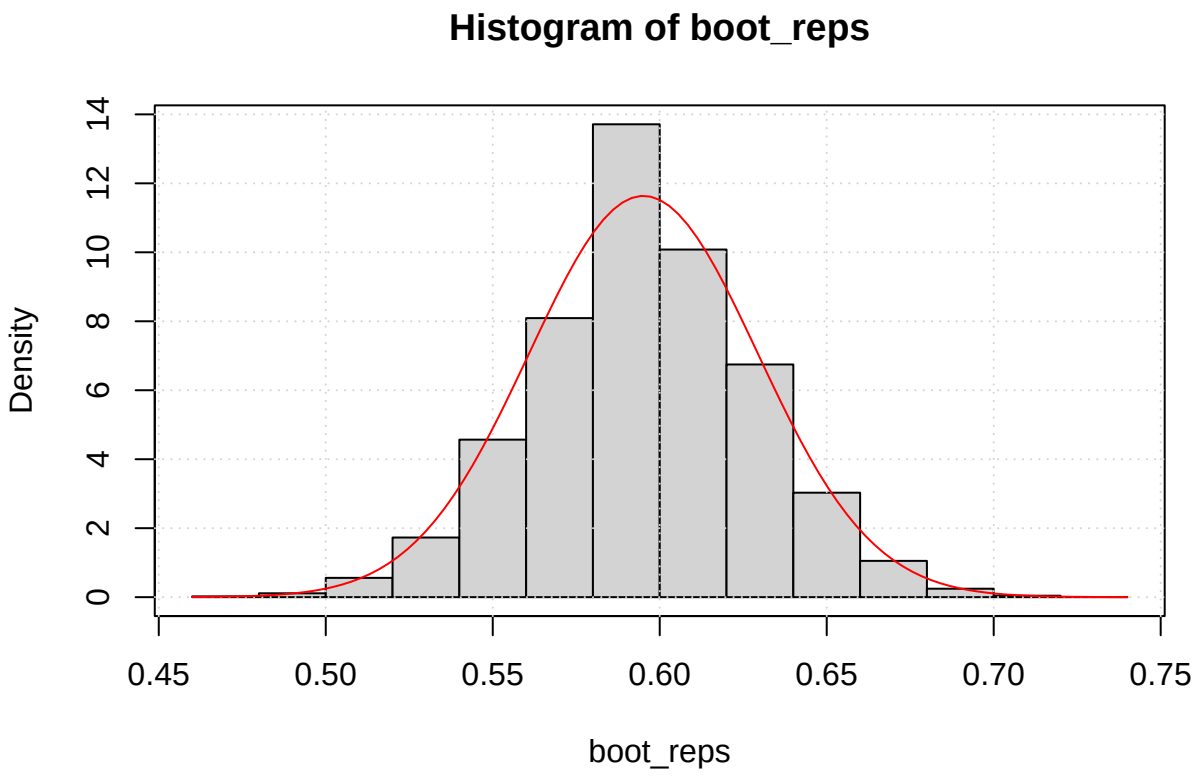
```
# calculate estimate of sd of sampling distribution
sd_theta_hat_hat = sqrt(theta_hat * (1 - theta_hat) / length(mel_survive))
```

Note that with a large sample we have

$$\hat{\theta} \approx N(\theta, \theta(1-\theta)/n).$$

We estimate this distribution by plugging in $\hat{\theta}$ for the unkown θ values.

```
# plot histogram
hist(boot_reps, probability = TRUE)
box()
grid()
curve(dnorm(x, mean = theta_hat, sd = sd_theta_hat_hat), add = TRUE, col = "red")
```



Exercise 6 (Deflategate)

On January 18, 2015, Clete Blakeman measured the pressure in pounds per square inch (PSI) of 15 footballs during halftime of the AFC Championship game. Of these footballs, 11 were a sample from the New England Patriots. The remaining 4 were a sample from the Indianapolis Colts. The data follows:

```
pats  = c(11.50, 10.85, 11.15, 10.70, 11.10, 11.60, 11.85, 11.10, 10.95, 10.50, 10.90)
colts = c(12.70, 12.75, 12.50, 12.55)
```

Use the percentile method to create a 95% confidence interval for the difference in medians of the pressure of the Patriot's and Colt's footballs. Use at least 2000 bootstrap samples.

- Note 1: These sample sizes are probably too small.
- Note 2: This is not a rigorous enough analysis to discredit the Patriots. However, disliking the Patriots is totally normal and acceptable! For actual details, see the Wells Report. (Be aware that the report contains text messages that use some not so pleasant language.)
- Note 3: This is not "tidy" data, but for this example, it is much easy to work with.

Solution

```
create_boot_rep = function() {
  boot_samp_pats = sample(pats, replace = TRUE)
  boot_samp_colts = sample(colts, replace = TRUE)
```

```
  median(boot_samp_pats) - median(boot_samp_colts)
}
```

```
boot_reps = replicate(n = 5000, create_boot_rep())
```

```
quantile(boot_reps, probs = c(0.025, 0.975))
```

```
##   2.5%  97.5%
## -1.825 -1.125
```

Exercise 7 (Rank Correlation)

The following loads the `airquality` data and then removes any missing data.

```
aq = na.omit(airquality)
```

Use the percentile method to create a 90% confidence interval for the Spearman rank correlation between `Ozone` and `Wind`. Use at least 2000 bootstrap samples.

Solution

```
create_boot_rep = function() {
  aq_boot_samp = dplyr::sample_n(aq, size = nrow(aq), replace = TRUE)
  cor(aq_boot_samp$Ozone, aq_boot_samp$Wind, method = "spearman")
}
```

```
boot_reps = replicate(n = 5000, create_boot_rep())
```

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

```
##         5%        95%
## -0.6992082 -0.4885093
```

Exercise 8 (Bootstrap Replicates and the Sampling Distribution)

The following code generates data.

```
set.seed(42)
some_data = rnorm(n = 100, mean = 5, sd = 1)
```

Suppose that we were interested in estimating $\theta = e^{\mu}$ and wanted to consider the estimator

$$\hat{\theta} = e^{\bar{X}}.$$

Generate 2000 (or more) bootstrap replicates of this estimator. Plot a histogram of these bootstrap replicates and overlay the **true** sampling distribution of $\hat{\theta}$.

Hint: Note that the distribution of $\bar{X}$ is normal, thus the distribution of $\hat{\theta}$ is a well known distribution that we have seen before. (Consider returning to the Homework 02 solutions.) The `dlnorm` function might be worth looking into.

Solution

First, note that

$$\bar{X} \sim N(\mu = 5, \sigma^2 = 1/n).$$

Thus, we also know that $\hat{\theta}$ follows a log-normal distribution with the same parameters.

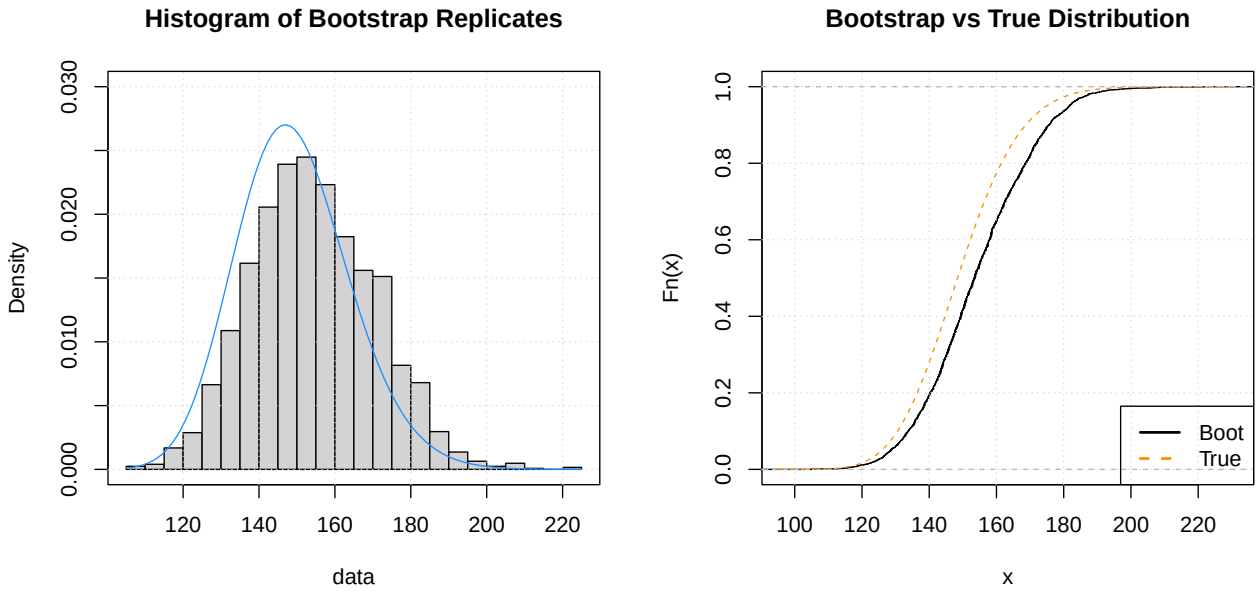
```
boot_reps = replicate(n = 2500, exp(mean(sample(some_data, replace = TRUE))))
```

```
# calculate sample size
n = length(some_data)
```

```
make_histogram = function(data) {
  hist(data, probability = TRUE, ylim = c(0, 0.03), breaks = 25,
       main = "Histogram of Bootstrap Replicates")
  box()
  grid()
  curve(dlnorm(x, meanlog = 5, sdlog = sqrt(1 / 100)),
        add = TRUE, col = "dodgerblue")
}
```

```
make_cdf_plot = function(data) {
  plot(ecdf(data), main = "Bootstrap vs True Distribution")
  curve(plnorm(x, meanlog = 5, sdlog = sqrt(1 / 100)), add = TRUE,
        col = "darkorange", lty = 2)
  grid()
  legend("bottomright", legend = c("Boot", "True"),
         lty = c(1, 2), col = c("black", "darkorange"), lwd = 2)
}
```

```
# setup plotting
par(mfrow = c(1, 2))
make_histogram(boot_reps)
make_cdf_plot(boot_reps)
```



Exercise 9 (The Bootstrap is Not Magic)

Based on Exercise 8, you might not yet be convinced that the empirical distribution of bootstrap replicates is a good estimate of the true sampling distribution.

Repeat Exercise 9 three additional times with the data provided below. For each, plot a histogram of these bootstrap replicates and overlay the **true** sampling distribution of $\hat{\theta}$. Additionally, plot the empirical CDF of

the bootstrap replicates as well as the CDF of the **true** sampling distribution of $\hat{\theta}$.

Your answer should be a total of six plots:

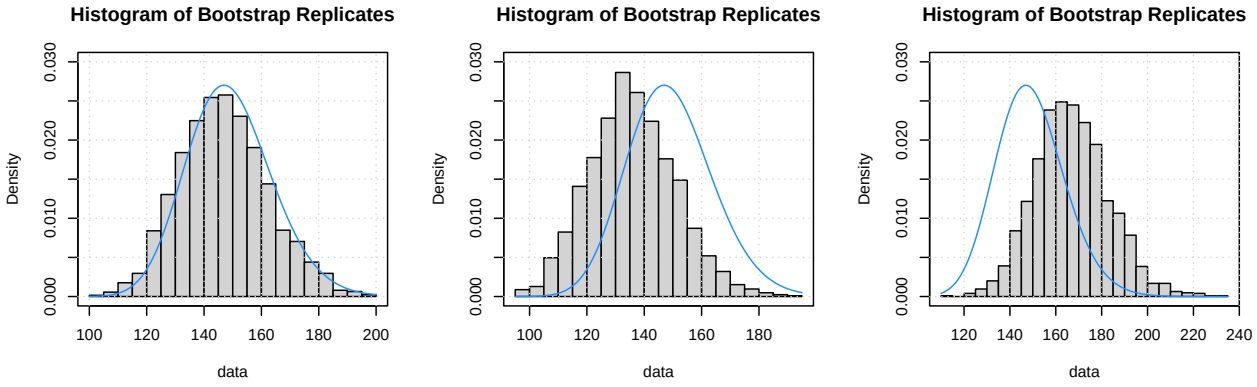
- Three histograms with a density, preferably side-by-side.
- Three plots, each with two CDFs, preferably side-by-side.

```
set.seed(17)
data_1 = rnorm(n = 100, mean = 5, sd = 1)
data_2 = rnorm(n = 100, mean = 5, sd = 1)
data_3 = rnorm(n = 100, mean = 5, sd = 1)
```

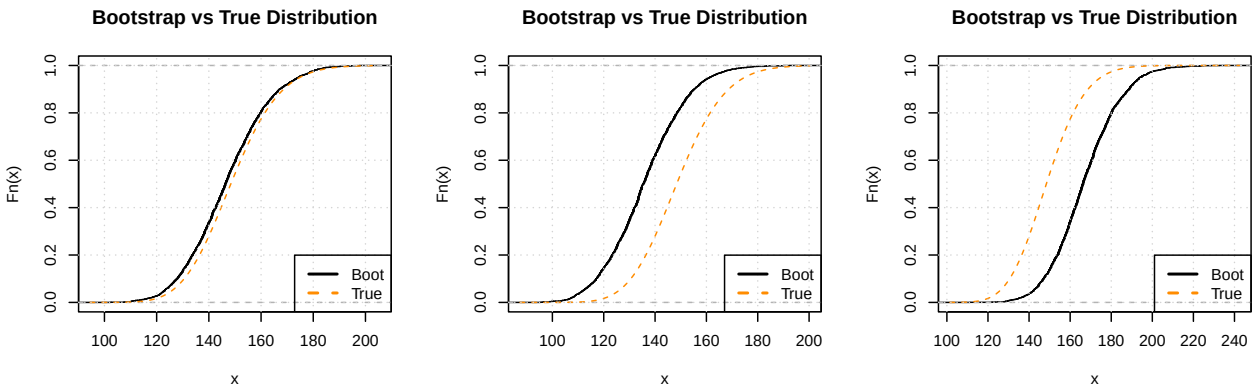
Solution

```
boot_reps_1 = replicate(n = 2500, exp(mean(sample(data_1, replace = TRUE))))
boot_reps_2 = replicate(n = 2500, exp(mean(sample(data_2, replace = TRUE))))
boot_reps_3 = replicate(n = 2500, exp(mean(sample(data_3, replace = TRUE))))
```

```
par(mfrow = c(1, 3))
make_histogram(boot_reps_1)
make_histogram(boot_reps_2)
make_histogram(boot_reps_3)
```




```
par(mfrow = c(1, 3))
make_cdf_plot(boot_reps_1)
make_cdf_plot(boot_reps_2)
make_cdf_plot(boot_reps_3)
```



Exercise 10 (Bootstrap Coverage)

Perform a simulation study to assess the coverage of the three bootstrap confidence interval methods we have discussed: Normal, Pivotal, Percentile

Let $n = 50$ and

$$T(F) = \int \frac{(x - \mu)^3}{\sigma^3} dF(x).$$

Generate $Y_1, Y_2, \dots Y_n \sim N(0, 1)$ and set $X_i = e^{Y_i}$ for $i = 1, 2, \dots n$. With this sample $X_1, \dots X_n$ construct a 95% confidence interval for $T(F)$ using each of the three methods. Use at least 500 bootstrap samples for each, but more is better.

Repeat this process at least 1000 times. Use the results to assess the coverage of the three interval types.

Solution

```
# helper function for plug-in estimator of skewness
calc_skew = function(x) {
  mean((x - mean(x)) ^ 3) / (sqrt(calc_var(x))) ^ 3
}

# helper function for plug-in estimator of variance
calc_var = function(x) {
  mean((x - mean(x)) ^ 2)
}
```

```
check_in_intervals = function(truth) {

  y = rnorm(50)
  x = exp(y)

  theta_hat = calc_skew(x)

  boot_replicates = replicate(n = 500, calc_skew(sample(x, replace = TRUE)))
  se = sqrt(calc_var(boot_replicates))

  norm_int = theta_hat + c(-1, 1) * qnorm(0.975) * se
  pivo_int = 2 * theta_hat - quantile(boot_replicates, c(0.975, 0.025))
  perc_int = quantile(boot_replicates, c(0.025, 0.975))

  c(
    in_norm = norm_int[1] < truth & truth < norm_int[2],
    in_pivo = unname(pivo_int[1] < truth & truth < pivo_int[2]),
    in_perc = unname(perc_int[1] < truth & truth < perc_int[2])
  )
}
```

```
truth = (exp(1) + 2) * sqrt(exp(1) - 1)
results = replicate(n = 1000, check_in_intervals(truth = truth))
rowMeans(results)
```

```
## in_norm in_pivo in_perc
##   0.108   0.150   0.022
```

Exercise 11 (More Bootstrap Coverage)

Repeat the above exercise, but also report the average length of the three interval types in addition to their coverage. This time use random samples of size $n = 25$ from a t distribution with 3 degrees of freedom. That is

$$X_1, \dots X_n \sim t_3$$

Let

$$\theta = T(F) = (q_{0.75} - q_{0.25})/1.34$$

To obtain a "plug-in" estimate for q_p you may simply use the default arguments to R's `quantile()` function.

Solution

```
# helper function to calculate statistic
calc_stat = function(x) {
  diff(quantile(x, probs = c(0.25, 0.75))) / 1.34
}
```

```
check_in_intervals = function(truth) {

  x = rt(n = 25, df = 3)

  theta_hat = calc_stat(x)

  boot_replicates = replicate(n = 500, calc_stat(sample(x, replace = TRUE)))
  se = sqrt(calc_var(boot_replicates))

  norm_int = theta_hat + c(-1, 1) * qnorm(0.975) * se
  pivo_int = 2 * theta_hat - quantile(boot_replicates, c(0.975, 0.025))
  perc_int = quantile(boot_replicates, c(0.025, 0.975))

  c(
    in_norm = norm_int[1] < truth & truth < norm_int[2],
    in_pivo = unname(pivo_int[1] < truth & truth < pivo_int[2]),
    in_perc = unname(perc_int[1] < truth & truth < perc_int[2]),
    len_norm = unname(diff(norm_int)),
    len_pivo = unname(diff(pivo_int)),
    len_perc = unname(diff(perc_int))
  )
}
```

```
truth = diff(qt(p = c(0.25, 0.75), df = 3)) / 1.34
results = replicate(n = 1000, check_in_intervals(truth = truth))
results = rowMeans(results)
```

```
# coverage
results[1:3]
```

```
## in_norm in_pivo in_perc
##   0.950   0.839   0.970
```

```r
# length
results[4:6]
```

```
## len_norm len_pivo len_perc
## 1.337449 1.299417 1.299417
```