Problem Section 6

Bootstrapped Intervals

Learning Outcomes

The problems are designed to build conceptual understanding and problem-solving skills. The emphasis is on learning to find, evaluate and build confidence. The specific tasks include:

- Implement a parametric and non-parametric bootstrap scheme

- Calculate bootstrap confidence intervals

- Back up and support work with relevant explanations

Instructions

- Please submit your completed file in CANVAS. There is no pre-groupwork or peer review this week.

Exercises

1. (Parametric bootstrap) Consider the following set of 25 numbers which are independently drawn from a normal distribution with mean 0 and variance $\sigma_0^2 = 1$.

```
set.seed(1225)

x <- rnorm(25, mean = 0, sd = 1)

x
```

```
##  [1]  1.373345216 -1.321717373 -1.130430171  0.174318722  1.572396621
##  [6] -1.033720544  0.528722173  0.433138549  0.359519501 -1.127509844
## [11]  0.572745921  1.164382652  1.058067005  0.195217444  0.156435394
## [16] -0.445555920 -0.930436079  1.185193882 -0.610439478 -0.068488257
## [21]  0.565900170 -0.322861038 -0.203976031  0.002425486  0.280508433
```

In this problem, we are interested in bootstrapping the sampling distribution of the method of moments estimator of σ_0^2, namely:

$$\hat{\sigma}_0^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

Since we know the data have been drawn from a normal distribution, we will use the parametric bootstrap method to accomplish this.

a. First, let's calculate the observed value of the estimator $\hat{\sigma}_0^2$ for this sample and save it in a variable called `obs_sigma2hat`. Report its value using inline code rounded to four digits.

```
n <- length(x)

obs_sigma2hat <- var(x)*(n-1)/n; obs_sigma2hat
```

```
## [1] 0.6548503
```

The observed value is 0.6549

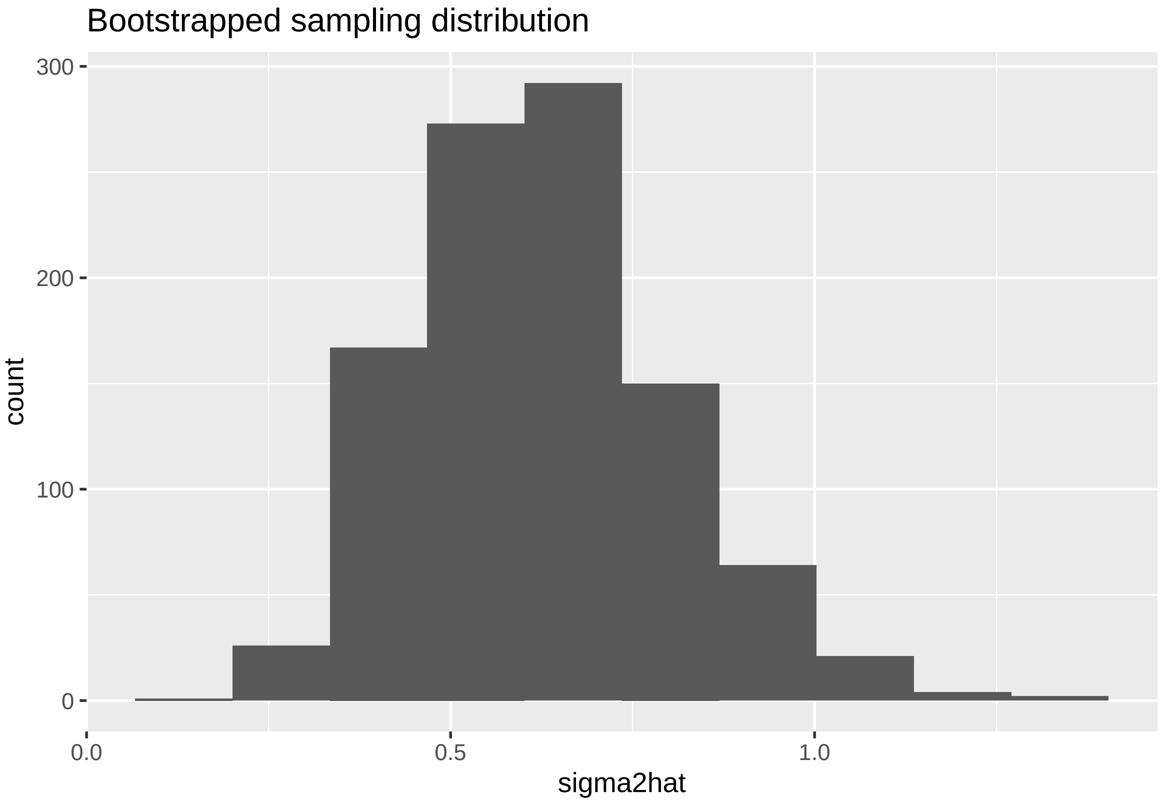
b. Now implement a parametric bootstrap scheme to approximate the sampling distribution of $\hat{\sigma}_0^2$. Generate $B = 1,000$ bootstrapped estimates and save them in a data frame called `boot_df`. For reproducibility, let's use 1234 as the random number seed. Scan the first few rows to look at it. (Consult slides 8 - 13 in the slide deck 6.1 - 6.2.)

```
set.seed(1234)

B <- 1000
n <- length(x)

boot_df <- tibble(
  sigma2hat = replicate(n = B,
                     expr = {
                       xstar = rnorm(n=25, mean =0, sd = sqrt(obs_sigma2hat))
                       var(xstar)*(n-1)/n
                     })
  )

boot_df
```

```
## # A tibble: 1,000 x 1
##    sigma2hat
##        <dbl>
##  1     0.536
##  2     0.411
##  3     0.863
##  4     0.511
##  5     0.423
##  6     0.528
##  7     0.583
##  8     1.22
##  9     0.477
## 10     0.726
## # i 990 more rows
```

c. Create a histogram of the bootstrapped estimates. Comment on the shape.

```
#bins = log2(1000)

ggplot(data = boot_df,
   mapping = aes(x = sigma2hat)) +
  geom_histogram(bins = 10 ) +
  labs(title = "Bootstrapped sampling distribution")
```



The bootstrapped sampling distribution is skewed to the right.

d. Calculate the average and standard deviation of the bootstrapped estimates. Calculate the bootstrap estimate of the bias in the estimator.

```
boot_df_summary <- boot_df %>% summarise(
               avg_est = mean(sigma2hat),
               se_est = sd(sigma2hat),
               boot_bias = avg_est - obs_sigma2hat)

boot_df_summary
```

```
## # A tibble: 1 x 3
##   avg_est se_est boot_bias
##     <dbl>  <dbl>     <dbl>
## 1   0.624  0.174   -0.0305
```

e. Revisit your answer from part a and bias correct your estimate in light of your answer in d. Call the bias corrected estimate as `unbias_obs_sigma2hat`

Since $\hat{\sigma}_0^2$ has a negative bias, we subtract it from our observed estimate. This has the effect of making the observed estimate slightly larger which helps counteract the fact that it underestimates the true value.

```
unbias_obs_sigma2hat <- obs_sigma2hat - boot_df_summary$boot_bias
unbias_obs_sigma2hat
```

```
## [1] 0.6853244
```

2. (Non-parametric bootstrap) Consider the following set of 20 numbers which were independently drawn from some (unknown) distribution with mean μ_0 and variance σ_0^2.

```
x  = c(3.56, 0.69, 0.10, 1.84, 3.93,
        1.25, 0.18, 1.13, 0.27, 1.21,
        0.50, 0.67, 0.01, 0.61, 0.82,
        1.70, 0.39, 0.11, 1.20, 0.72)
```

In this problem we are interested in constructing a confidence interval estimate for σ_0^2 using the method of moments estimator

$$\hat{\sigma}_0^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

a. Calculate the observed value of the estimator $\hat{\sigma}_0^2$ for this sample and save it in a variable called `obs_sigma2hat`. Report its value using inline code rounded to four digits.

```
# calculate obs_sigma2hat for the observed sample
    n <- length(x)

    obs_sigma2hat <- var(x)*(n-1)/n; obs_sigma2hat
```

```
## [1] 1.066775
```

Now we wish to construct the sampling distribution of $\hat{\sigma}_0^2$. Since we do not know the actual distribution that these data were sampled from, we will use the non-parametric bootstrap method this time to help us create an approximate sampling distribution.

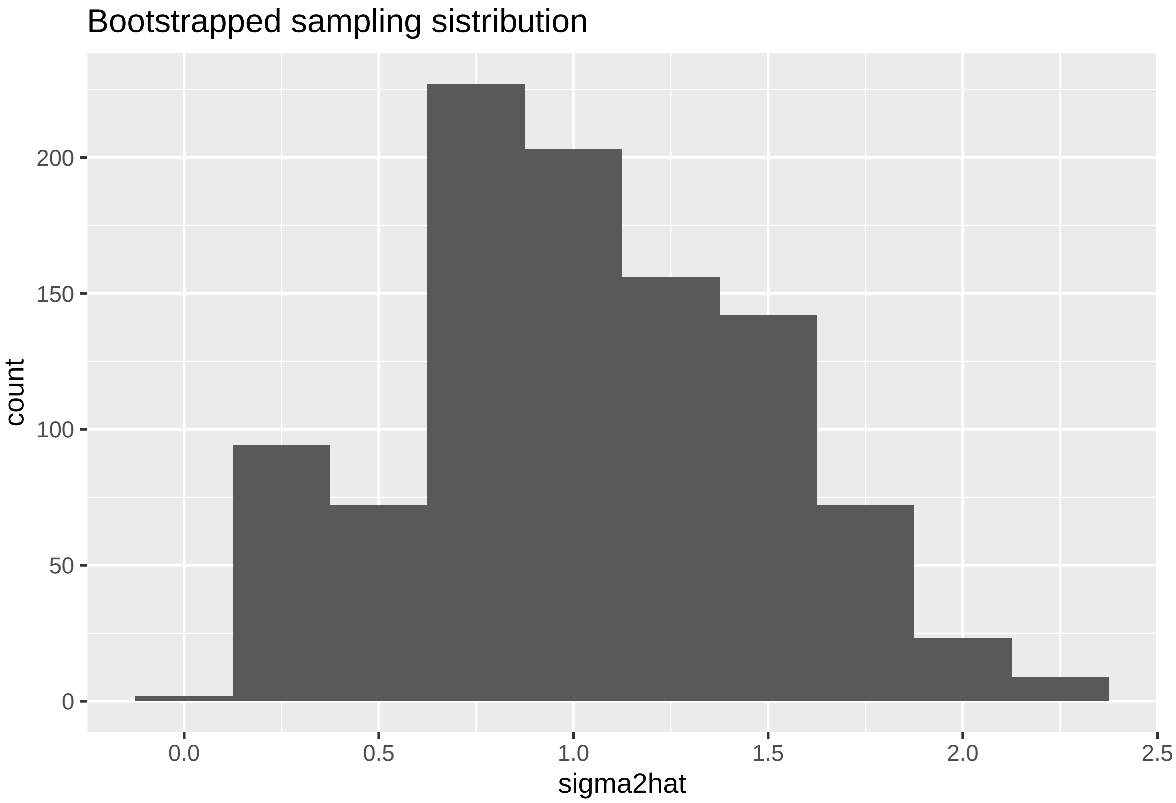
b. Follow the steps below to construct the bootstrapped sampling distribution of $\hat{\sigma}_0^2$. You can look at pages 16 - 19 of the slidedeck labeled "6.1 - 6.2")

   - Take B random samples of size 20 each from the observed sample data with replacement and calculate the value of the bootstrapped estimate - call it sigma2hat - for each resample. Save the estimates in a data frame called `boot_df`.
   - Make a histogram of the bootstrapped estimates. Examine the shape. Calculate the mean and standard deviation of the bootstrapped estimates.
   - If the bootstrap results indicate evidence of bias in the estimator, adjust your point estimate from part a accordingly to calculate a bias-corrected version.

```
set.seed(1415151)

B = 1000
n = length(x)

#Step 1:
boot_df <- tibble(
   sigma2hat = replicate(n = B,
                   expr = {
                     xstar = sample(x, size = n, replace = TRUE)
                     var(xstar )*(n-1)/n
                      } #end expr
                       ) #end replicate
                        )  #end tibble

boot_df %>% slice_head(n=5)
```

```
## # A tibble: 5 x 1
##   sigma2hat
##       <dbl>
## 1     0.808
## 2     1.29
## 3     0.214
## 4     1.82
## 5     1.06
```

```
#Step 2: Create a histogram

ggplot(data=boot_df,
   mapping = aes(x = sigma2hat)) +
  geom_histogram(binwidth = 0.25) +  #10  bins = 0.25 binwidth
  labs(title="Bootstrapped sampling sistribution")
```



```
#Calculate average and sd of bootstrapped estimates and also bias
boot_df_summary <- boot_df %>%
                 summarise( avg_est = mean(sigma2hat),  #expected value
                            sd_est = sd(sigma2hat),  #standard error
                            boot_bias = avg_est - obs_sigma2hat)

boot_df_summary
```

```
## # A tibble: 1 x 3
##   avg_est sd_est boot_bias
##     <dbl>  <dbl>     <dbl>
## 1    1.03  0.456   -0.0373
```

Again, we will see a negative bias in our estimator. So we can bias correct by subtracting it from the observed value of the estimate which will essentially inflate the estimate a bit.

```
unbias_obs_sigma2hat <- obs_sigma2hat - boot_df_summary$boot_bias
unbias_obs_sigma2hat
```

```
## [1] 1.104122
```

c. Calculate a 90% bootstrap percentile interval estimate for σ_0^2 and report it (rounded to 4 digits) in a sentence using inline code.

```
#calculate 90% bootstrap percentile interval by finding the
# 5th percentile and 95th percentile of the bootstrapped sampling dist

boot_df %>% summarize(
       lower = quantile(sigma2hat, p = 0.05),
       upper = quantile(sigma2hat,p = 0.95)
)
```

```
## # A tibble: 1 x 2
##   lower upper
##   <dbl> <dbl>
## 1 0.261  1.80
```