Problem Section 8

Empirical P-values

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:

- Find an empirical P-value

Instructions

- Please submit your completed file in CANVAS

Exercises

1. Suppose the following sample - denoted by $X_1, X_2, \ldots, X_n$ - are drawn from a $Norm(0, \sigma_0^2)$ distribution:

```
x <- c(-0.58319935, -1.36090219,  0.38663763, -1.54365592,
       0.87083945, -0.69187830, 0.45898841, -2.82556635,
       0.01777137, -0.62753863 , 0.54611381, -1.39731591,
       -1.72584231,  0.91371529,  0.18096064, -0.53063107,
       -0.76604739, -1.97107704, 0.56394712,  1.13707563)
```

Suppose we want to test $H_0 : \sigma_0^2 = 1$ versus $\sigma_0^2 > 1$ using the sample variance

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2.$$

as our test statistic Note: S^2 is an unbiased estimator for σ_0^2.

a. Calculate the observed value for S^2 and save it in a variable called obs_s2.

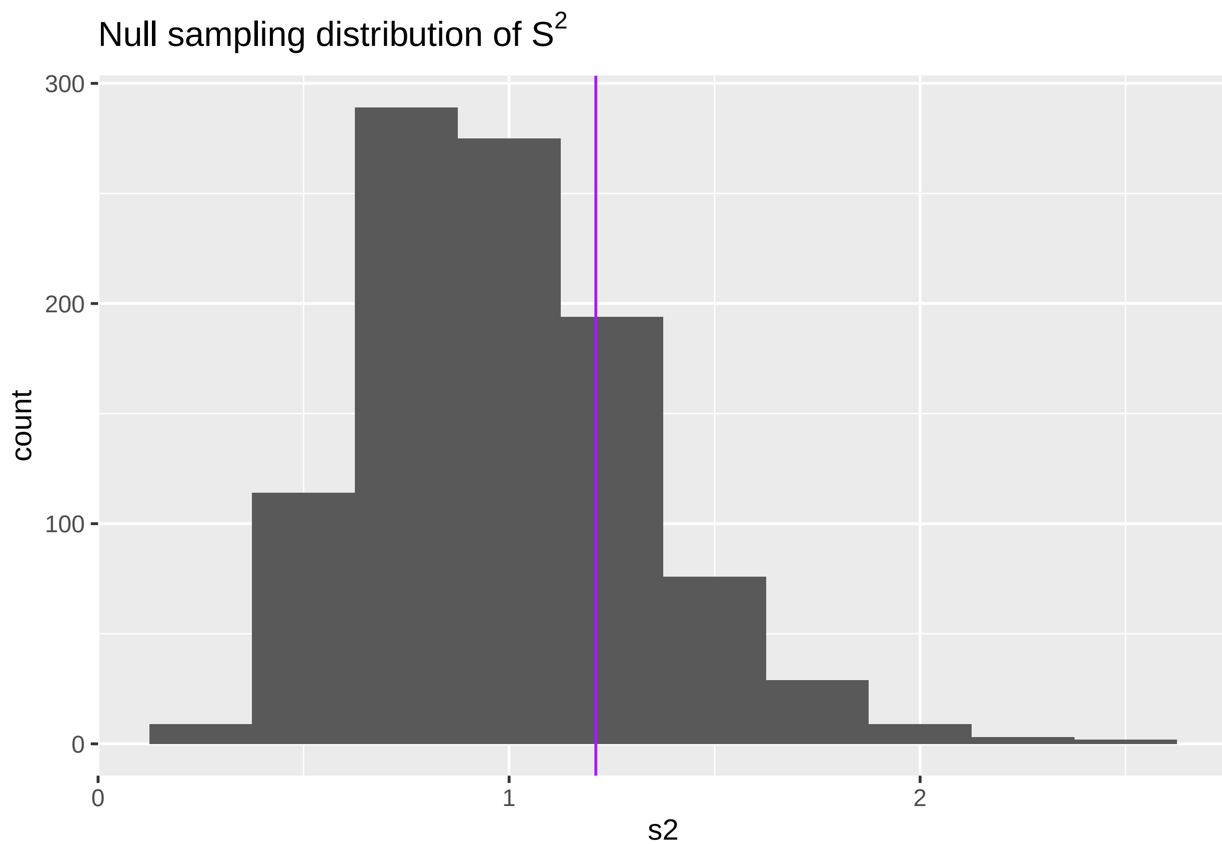
```
obs_s2 <- var(x)
```

b. Simulate a large number - B- of values from the null sampling distribution of the estimator S^2. Make a histogram of these values, label it properly and mark the observed value from part a with a purple vertical line.

   Some tips: Create a dataframe called `null_dist` which has a column $s2$ with the values for S^2 when data are simulated from the null hypothesis.

```
set.seed(2626)
nobs <- length(x)                          #sample size
B <- 1000
sigma2_null <- 1

#write code to simulate values of S2 from the null hypothesis
```

```r
null_dist <- tibble(
    s2 = replicate(n = B,
                   expr = var(rnorm(n = nobs,
                                    mean = 0,
                                    sd = sqrt(sigma2_null)) ) )
   )

# write code to make a histogram of null dist of S2.
#don't forget to play with binwidth

ggplot(data = null_dist,
       mapping = aes(x = s2)) +
       geom_histogram(binwidth = 0.25) +  #binwidth = 2.5/log2(1000) +
       geom_vline( xintercept = obs_s2,
                   color = "purple")+
       labs(title = expression(paste("Null sampling distribution of ", S^2)))
```



c. Calculate the empirical P-value. What would you conclude at any reasonable α level?

```r
sum(null_dist$s2 >= obs_s2)/B
```

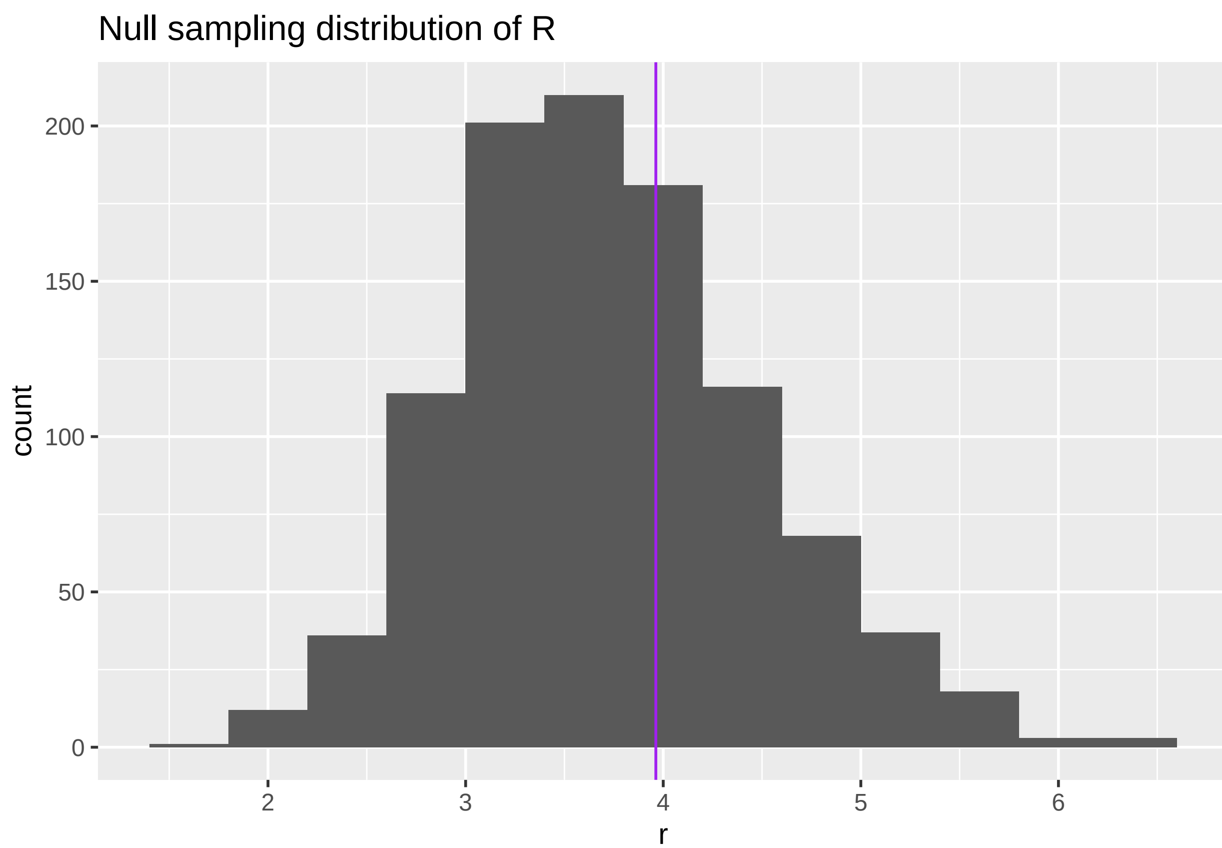
```
## [1] 0.226
```

For any reasonable α level, say 0.05, 0.01, we would fail to reject the null hypothesis since the P-value is larger than α.

For those who want to stretch. This will also be covered in section on Monday.

d. Suppose we now decide to use the Range $R = X_{max} - X_{min}$ as our test statistic. Fill in the blanks in the code below to repeat parts a, b, c using the statistic R. We are now going to write code for b flexibly - using `lapply` - so we can calculate multiple statistics for each iteration - s^2 and r - from the same simulated data and return a dataframe with their values.

```
set.seed(2626)

B = 1000

#calculate observed value of R
obs_r = diff(range(x))

#create a list with B small data frames in it each
#containing a s2 and r calculated for a data set simulated
#under the null hypothesis

null_dist_list <- lapply( 1:B, FUN = function(i){
                          sim_x = rnorm(n = nobs,
                                        mean = 0,
                                        sd = sqrt(sigma2_null))
                          data.frame( s2 = var(sim_x),
                                      r = diff( range(sim_x) ) ) })
#use map_dfr to bind the mini dataframes by row
#into one larger dataframe

null_dist <- map_dfr(null_dist_list, bind_rows)

# write code to make a histogram of null dist of R.
#don't forget to play with binwidth
ggplot(data = null_dist,
       mapping = aes(x = r)) +
       geom_histogram(binwidth = 0.4) +  #binwidth = 4/log2(1000) +
       geom_vline( xintercept = obs_r,
                   color = "purple")+
       labs(title = "Null sampling distribution of R")
```




```
#calculate empirical Pvalue for test based on R as test statistic

sum(null_dist$r >= obs_r)/B
```

```
## [1] 0.35
```

We will now switch gears and think about conducting the test at a level $\alpha = 0.05$. Of course, neither S^2 nor R are large enough for rejection at this level. What sort of value would we need to see to be able to reject at this level? Let's find out.

e. Suppose we decide to use a level $\alpha = 0.05$ test. What values of S^2 will you reject H_0 for? Same question for R.

   Hint: find the 95th percentile of the null values for each statistic. Think about why.

   Since the P-value is of the form $P(T \geq t)$ where T is the test statitic with observed value t, it will be equal to 0.05 when our observed test statistic is the 95th percentile. Therefore, we will reject at a $\alpha = 0.05$ level if we observe a value that is equal to the 95th percentile or larger.

```
critical_values <- null_dist %>%
                summarise(s2_crit = quantile(s2, p = 0.95),
                          r_crit = quantile(r, p = 0.95) )
critical_values
```

```
##    s2_crit   r_crit
## 1 1.578237 5.085938
```

f. For the decision rule corresponding to the level $\alpha = 0.05$ test from part e, compare the Type II error probabilities for S^2 and R when $\sigma_0^2 = 3$. Remember, the Type II error rate is the probability that we fail to reject H_0 when H_1 is true. How do they do?

Hint: You will first need to simulate data under the alternative and then calculate the Type II error rate. No scaffolded code provided.

```
set.seed(151)
sigma2_alt <- 3

alt_dist_list <- lapply( 1:B, FUN = function(i){
                      sim_x = rnorm(n = nobs,
                                    mean = 0,
                                    sd = sqrt(sigma2_alt))
                      data.frame( s2 = var(sim_x),
                                  r = diff( range(sim_x) ) ) })
 #use map_dfr to bind the mini dataframes by row
 #into one larger dataframe

 alt_dist <- map_dfr(alt_dist_list, bind_rows)


 #calculate Type II error for test based on R as test statistic and also S2

 sum(alt_dist$s2 < critical_values$s2_crit)/B
```

```
## [1] 0.046
```

```
 sum(alt_dist$r < critical_values$r_crit)/B
```

```
## [1] 0.134
```