Problem Set 10 Solutions

MATH E-156: Mathematical Statistics

```
load( "Problem Set 10 R Objects.Rdata")
```

Problem 1

This problem is a little bit unusual, maybe even a little bit weird. Nonethelss, like many weird things it's also kind of cool and interesting, and it will give you some practice with the vectorized operations that we've been working with this week.

In module 2, we explored the fundamental equation for sums of squares:

$$\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{ij}-\overline{x}_{..})^2 \;=\; J\cdot\sum_{i=1}^{I}(\overline{x}_{i.}-\overline{x}_{..})^2+\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{ij}-\overline{x}_{i.})^2$$

I emphasized that this is an algebraic identity, and it holds for all possible values of the $\{x_{ij}\}$, although it does require that all the groups have the same number of observations. In addition, there are no distributional assumptions.

To test these claims, we're going to generate three sets of random values from three totally different probability distributions:

- First, we'll generate 30 observations from a normal distribution with a mean of 100 and a variance of 100.
- Next, we'll generate 30 observations from an exponential distribution with a rate parameter of 0.04.
- Finally, we'll generate 30 observations from a binomial distribution with 20 trials and a probability of success of 0.7.

I'll do this for you; note that I'm including a call to `set.seed()` so that everyone has the same data for the TAs to grade:

```
set.seed( 1 )

group.1.data <-
    rnorm( 30, mean = 100, sd = 10 )

group.2.data <-
    rexp( 30, rate = 0.04 )

group.3.data <-
    rbinom( 30, size = 20, prob = 0.7)
```

If you want to experiment with this problem, you are welcome to change the value for the `set.seed()` or even get rid of it completely, but when you submit your homework you should use `set.seed(1)` just for the sake of uniformity.

Part (a): Total Sum of Squares

Combine the data in the 3 group vectors into one aggregate vector, and calculate the grand mean of the data using the sample mean of the aggregate data vector. Then calculate the total sum of squares using this grand mean. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

```
aggregate.data.vector <-
    c(
        group.1.data,
        group.2.data,
        group.3.data
    )

grand.mean <-
    mean( aggregate.data.vector )

total.sum.of.squares <-
    sum( (aggregate.data.vector - grand.mean)^2 )

cat( "Total Sum of Squares:",
     round( total.sum.of.squares, 5 ) )
```

```
## Total Sum of Squares: 148521.4
```

Part (b): Treatment sum of squares

For each of the three groups, calculate the sample mean. Then calculate the sum of squares for the treatments. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

```
group.1.mean <-
    mean( group.1.data )


group.2.mean <-
    mean( group.2.data )


group.3.mean <-
    mean( group.3.data )

group.mean.vector <-
    c(
        group.1.mean,
        group.2.mean,
        group.3.mean
    )

treatment.sum.of.squares <-
    30 *
```

```
    sum(
        (group.mean.vector - grand.mean)^2
    )
```

Part (c): Error sum of squares

Calculate the error sum of squares for this data. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

```
error.sum.of.squares <-
    sum(
        sum( (group.1.data - group.1.mean)^2 ),
        sum( (group.2.data - group.2.mean)^2 ),
        sum( (group.3.data - group.3.mean)^2 )
    )
```

```
cat( "Error sum of squares:",
     round( error.sum.of.squares, 5 ) )
```

```
## Error sum of squares: 11104.87
```

Part (d): Conclusion

Calculate the sum of the treatment sum of squares and the error sum of squares from parts (b) and (c). Report your result using a `cat()` statement, rounding to 5 decimal places. How does this value compare with the total sum of squares that you calculated in part (a)?

Solution

```
sum.of.sums.of.squares <-
    treatment.sum.of.squares + error.sum.of.squares

cat( "Sum of sums of squares:",
     round( sum.of.sums.of.squares, 5 ) )
```

```
## Sum of sums of squares: 148521.4
```

End of problem 1

Problem 2: Treatment Sum of Squares, Part 1

Now we will explore the sampling distribution of the random variable U:

$$U = \frac{SSTr}{\sigma^2} = \frac{J \cdot \sum_{i=1}^{I} (\overline{x}_{i\cdot} - \overline{x}_{\cdot\cdot})^2}{\sigma^2} \sim \chi^2(I-1)$$

In this problem, we will consider the case where the null hypothesis is true, so that all the group-specific population expected values are equal.

Consider this experimental protocol:

- We have three groups, so $I = 3$.
- Each group is normally distributed, with an expected value of 25 and a variance of 10.
- We draw a sample of size $J = 15$ from each group.

Here are some variables for you:

```
number.of.groups <- 3

group.sample.size <- 15
```

Part (a): Degrees of freedom

For this experimental protocol, what are the degrees of freedom for the random variable U? Report your answer with one or two sentences.

Solution

```
degrees.of.freedom <-
    number.of.groups - 1
```

Part (b): Simulation

Now we're going to simulate the sampling distribution of the random variable U.

For each simulation replication:

- First, draw three random samples from the three groups.
- Next, calculate the treatment sum of squares for this random data.
- Divide the random treatment sum of squares by the common population variance.
- Store this value in the outcome vector.

When the simulation is complete, the outcome vector will be populated with random values of the random variable U.

There's nothing to report for this part, but write your code clearly so the TAs can understand what you're doing.

Solution

```r
population.mean <- 25

population.variance <- 10

number.of.groups <- 3

group.sample.size <- 15

number.of.replications <- 10000

outcome.vector <- numeric( number.of.replications )

for( replication.index in 1:number.of.replications ) {

    group.1.sample.mean <-
        mean(
            rnorm(
                group.sample.size,
                mean = population.mean,
                sd = sqrt( population.variance )
            )
        )

    group.2.sample.mean <-
        mean(
            rnorm(
                group.sample.size,
                mean = population.mean,
                sd = sqrt( population.variance )
            )
        )

    group.3.sample.mean <-
        mean(
            rnorm(
                group.sample.size,
                mean = population.mean,
                sd = sqrt( population.variance )
            )
        )

    group.mean.vector <-
        c(
            group.1.sample.mean,
            group.2.sample.mean,
            group.3.sample.mean
        )

    grand.mean <-
        mean( group.mean.vector )

    treatment.sum.of.squares <-
        group.sample.size *
```

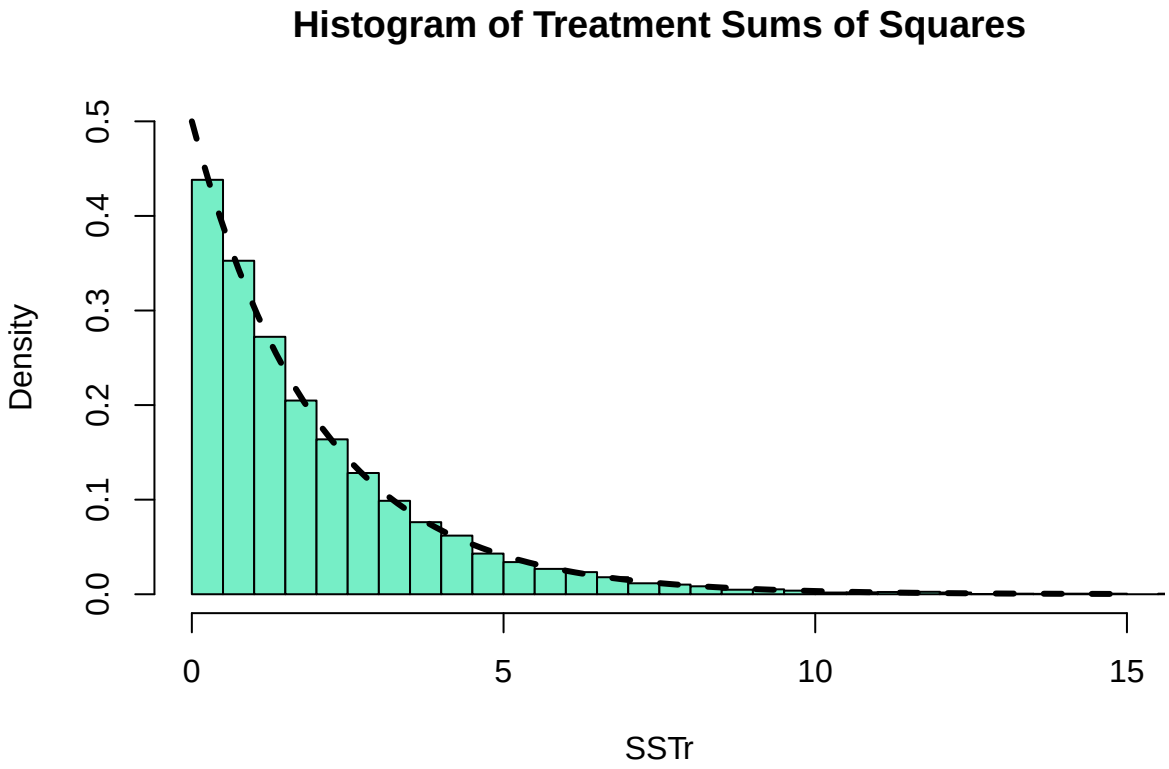
```
        sum( (group.mean.vector - grand.mean)^2 )

    outcome.vector[ replication.index ] <-
        treatment.sum.of.squares /
        population.variance
}
```

Part (c): Histogram

Construct a histogram of the random values in the outcome vector you generated in part (b). Then superimpose the density curve for a chi-squared distribution using the degrees of freedom from part (a). How well does the density curve fit the observed data?

Solution

```
hist(
    outcome.vector,
    prob = TRUE,
    xlim = c(0, 15),
    ylim = c(0, 0.5),
    main = "Histogram of Treatment Sums of Squares",
    xlab = "SSTr",
    ylab = "Density",
    col = "aquamarine2",
    breaks = 50
)

curve(
    dchisq(x, df = degrees.of.freedom),
    lty = "dashed",
    lwd = 3,
    col = "black",
    add = TRUE
)
```


Histogram of Treatment Sums of Squares
Density
0.0
0.1
0.2
0.3
0.4
0.5
0
5
10
15
SSTr

End of problem 2

Problem 3: Treatment Sum of Squares, Part 2

We continue our exploration of the sampling distribution of the random variable U:

$$U = \frac{SSTr}{\sigma^2} = \frac{J \cdot \sum_{i=1}^{I} (\overline{x}_{i\cdot} - \overline{x}_{\cdot\cdot})^2}{\sigma^2} \sim \chi^2(I-1)$$

In this problem, we will consider the case where the null hypothesis is **not** true, so that one of the group-specific population expected values is different from the others.

Consider this experimental protocol:

- We have three groups, so $I = 3$.
- The first two groups are normally distributed, each with an expected value of 25 and a common variance of 10.
- The third group is normally distributed, again with a common variance of 10, but now the expected value of this group is 27.
- We draw a sample of size $J = 15$ from each group.

Part (a): Simulation

Now we're going to simulate the sampling distribution of the random variable U.

For each simulation replication:

- First, draw three random samples from the three groups.
- Next, calculate the treatment sum of squares for this random data.
- Divide the random treatment sum of squares by the common population variance.
- Store this value in the outcome vector.

When the simulation is complete, the outcome vector will be populated with random values of the random variable U.

There's nothing to report for this part, but write your code clearly so the TAs can understand what you're doing.

Solution

```
population.mean <- 25

population.variance <- 10

number.of.groups <- 3

group.sample.size <- 15

number.of.replications <- 10000

outcome.vector <- numeric( number.of.replications )
```

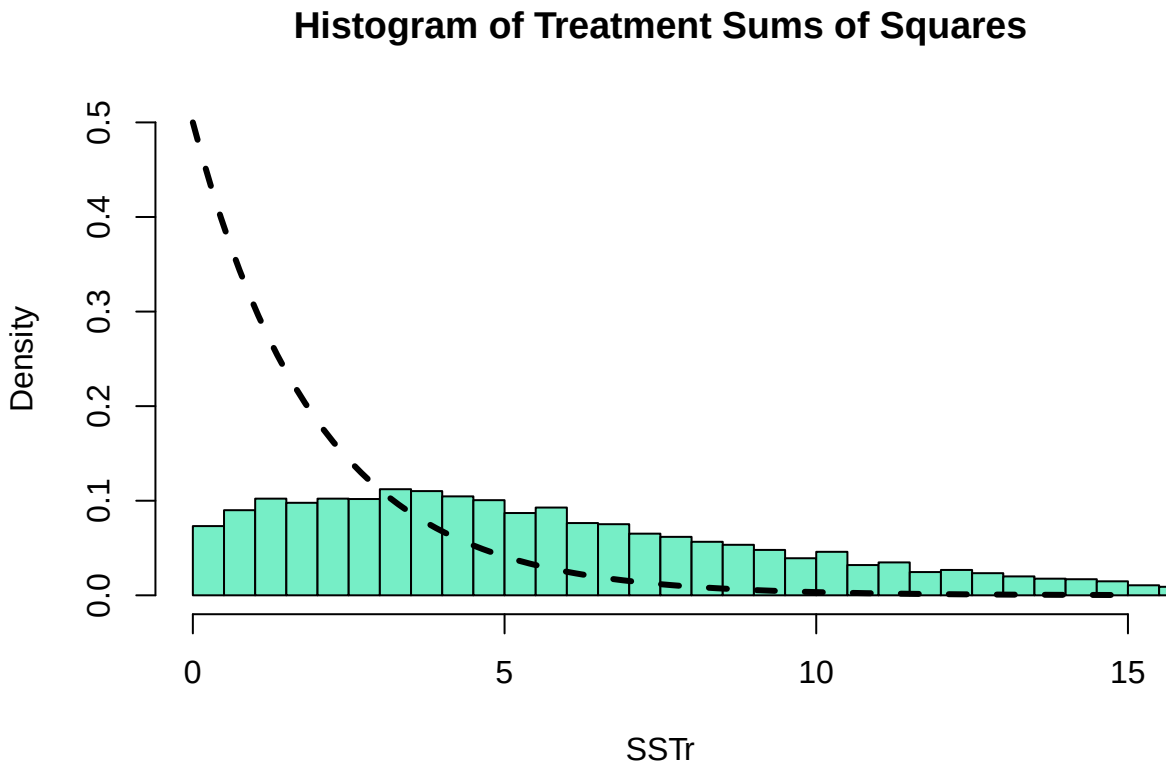
```
for( replication.index in 1:number.of.replications ) {

    group.1.sample.mean <-
        mean(
            rnorm(
                group.sample.size,
                mean = population.mean,
                sd = sqrt( population.variance )
            )
        )

    group.2.sample.mean <-
        mean(
            rnorm(
                group.sample.size,
                mean = population.mean,
                sd = sqrt( population.variance )
            )
        )

    group.3.sample.mean <-
        mean(
            rnorm(
                group.sample.size,
                mean = population.mean + 2,
                sd = sqrt( population.variance )
            )
        )

    group.mean.vector <-
        c(
            group.1.sample.mean,
            group.2.sample.mean,
            group.3.sample.mean
        )

    grand.mean <-
        mean( group.mean.vector )

    treatment.sum.of.squares <-
        group.sample.size *
        sum( (group.mean.vector - grand.mean)^2 )

    outcome.vector[ replication.index ] <-
        treatment.sum.of.squares /
        population.variance
}
```

Part (b): Histogram

Construct a histogram of the random values in the outcome vector you generated in part (a). Then superimpose the density curve for a chi-squared distribution using the degrees of freedom that you determined in problem 2. How well does the density curve fit the observed data?

Solution

```
hist(
    outcome.vector,
    prob = TRUE,
    xlim = c(0, 15),
    ylim = c(0, 0.5),
    main = "Histogram of Treatment Sums of Squares",
    xlab = "SSTr",
    ylab = "Density",
    col = "aquamarine2",
    breaks = 50
)

curve(
    dchisq(x, df = number.of.groups - 1),
    lty = "dashed",
    lwd = 3,
    col = "black",
    add = TRUE
)
```



End of problem 3

Problem 4: Error Sum of Squares, Part 1

Now we will explore the sampling distribution of the random variable V:

$$V = \frac{SSE}{\sigma^2} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{ij} - \overline{x}_{i\cdot})^2}{\sigma^2} \sim \chi^2(I \cdot (J-1))$$

In this problem, we will consider the case where the null hypothesis is true, so that all the group-specific population expected values are equal.

Consider this experimental protocol:

- We have three groups, so $I = 3$.
- Each group is normally distributed, with an expected value of 25 and a variance of 10.
- We draw a sample of size $J = 15$ from each group.

Part (a): Degrees of freedom

For this experimental protocol, what are the degrees of freedom for the random variable V? Report your answer with one or two sentences.

Solution

```
degrees.of.freedom <-
    number.of.groups * (group.sample.size - 1)

degrees.of.freedom
```

```
## [1] 42
```

Part (b): Simulation

Now we're going to simulate the sampling distribution of the random variable V.

For each simulation replication:

- First, draw three random samples from the three groups.
- Next, calculate the error sum of squares for this random data.
- Divide the random error sum of squares by the common population variance.
- Store this value in the outcome vector.

When the simulation is complete, the outcome vector will be populated with random values of the random variable U.

There's nothing to report for this part, but write your code clearly so the TAs can understand what you're doing.

Solution

```
population.mean <- 25

population.variance <- 10

number.of.groups <- 3

group.sample.size <- 15

number.of.replications <- 10000

outcome.vector <- numeric( number.of.replications )

for( replication.index in 1:number.of.replications ) {

    group.1.random.sample <-
        rnorm(
            group.sample.size,
            mean = population.mean,
            sd = sqrt( population.variance )
        )

    group.1.sample.mean <-
        mean( group.1.random.sample )

    group.2.random.sample <-
        rnorm(
            group.sample.size,
            mean = population.mean,
            sd = sqrt( population.variance )
        )

    group.2.sample.mean <-
        mean( group.2.random.sample )

    group.3.random.sample <-
        rnorm(
            group.sample.size,
            mean = population.mean,
            sd = sqrt( population.variance )
        )

    group.3.sample.mean <-
        mean( group.3.random.sample )

    error.sum.of.squares <-
        sum(
            (group.1.random.sample - group.1.sample.mean)^2,
            (group.2.random.sample - group.2.sample.mean)^2,
            (group.3.random.sample - group.3.sample.mean)^2
        )

    outcome.vector[ replication.index ] <-
        error.sum.of.squares /
```

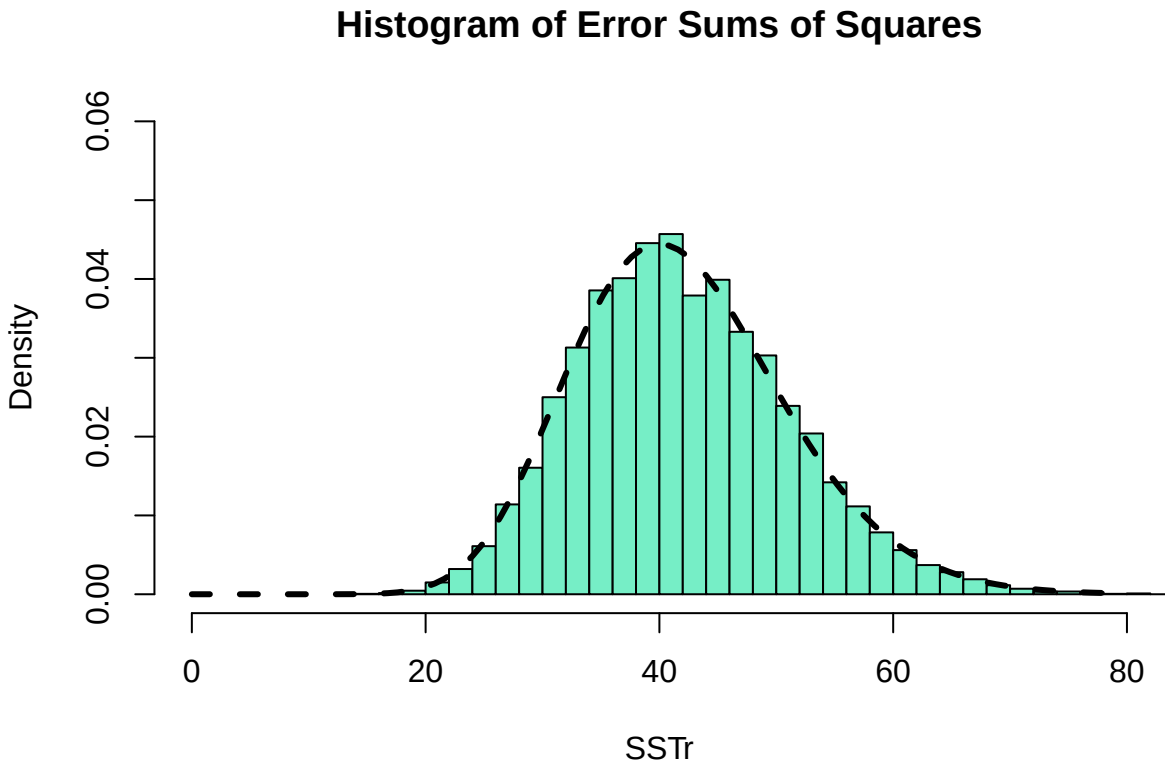
```
        population.variance
}
```

Part (c): Histogram

Construct a histogram of the random values in the outcome vector you generated in part (b). Then superimpose the density curve for a chi-squared distribution using the degrees of freedom from part (a). How well does the density curve fit the observed data?

Solution

```
hist(
    outcome.vector,
    prob = TRUE,
    xlim = c(0, 80),
    ylim = c(0, 0.06),
    main = "Histogram of Error Sums of Squares",
    xlab = "SSTr",
    ylab = "Density",
    col = "aquamarine2",
    breaks = 50
)

curve(
    dchisq(x, df = degrees.of.freedom),
    lty = "dashed",
    lwd = 3,
    col = "black",
    add = TRUE
)
```


Histogram of Error Sums of Squares
Density
0.00
0.02
0.04
0.06
0
20
40
60
80
SSTr

End of problem 4

Problem 5: Error Sum of Squares, Part 2

Now we will explore the sampling distribution of the random variable V:

$$V = \frac{SSE}{\sigma^2} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}(x_{ij} - \overline{x}_{i\cdot})^2}{\sigma^2} \sim \chi^2(I \cdot (J-1))$$

In this problem, we will consider the case where the null hypothesis is **not** true, so that one of the group-specific population expected values is different from the others.

Consider this experimental protocol:

- We have three groups, so $I = 3$.
- The first two groups are normally distributed, each with an expected value of 25 and a common variance of 10.
- The third group is normally distributed, again with a common variance of 10, but now the expected value of this group is 27.
- We draw a sample of size $J = 15$ from each group.

Part (a): Simulation

Now we're going to simulate the sampling distribution of the random variable V.

For each simulation replication:

- First, draw three random samples from the three groups.
- Next, calculate the error sum of squares for this random data.
- Divide the random error sum of squares by the common population variance.
- Store this value in the outcome vector.

When the simulation is complete, the outcome vector will be populated with random values of the random variable U.

There's nothing to report for this part, but write your code clearly so the TAs can understand what you're doing.

Solution

```
population.mean <- 25

population.variance <- 10

number.of.groups <- 3

group.sample.size <- 15

number.of.replications <- 10000

outcome.vector <- numeric( number.of.replications )
```

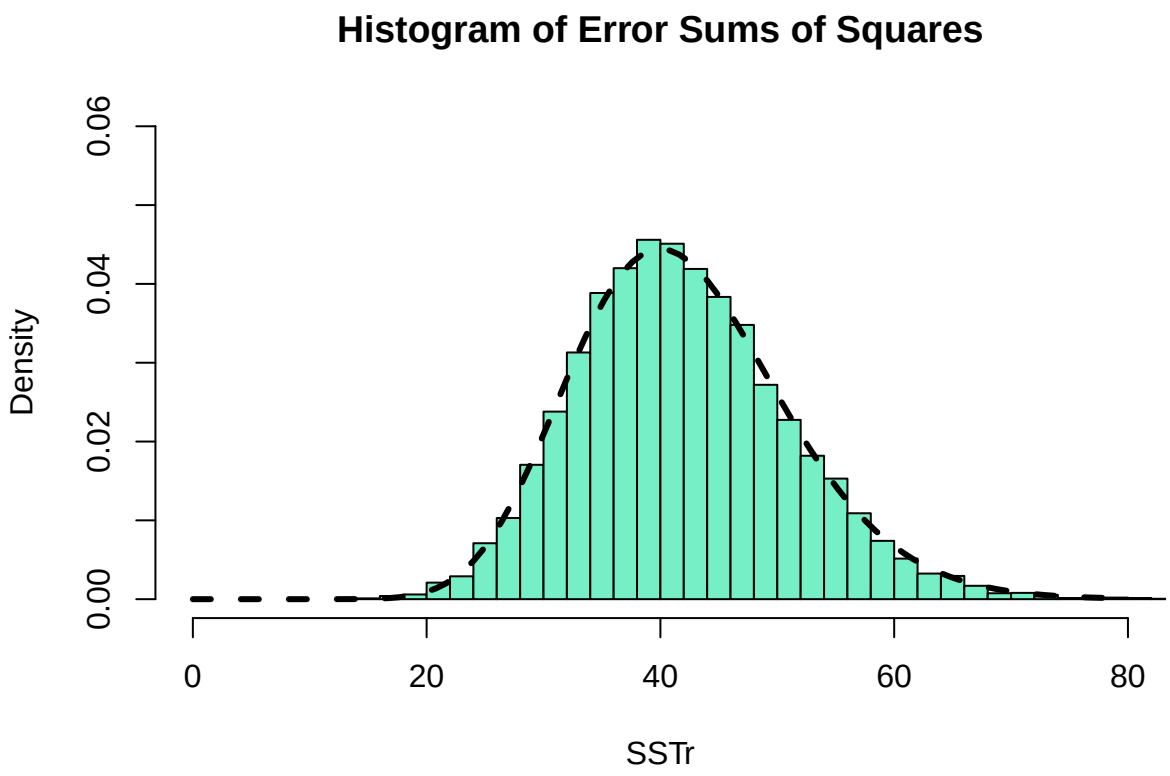
```
for( replication.index in 1:number.of.replications ) {

    group.1.random.sample <-
        rnorm(
            group.sample.size,
            mean = population.mean,
            sd = sqrt( population.variance )
        )

    group.1.sample.mean <-
        mean( group.1.random.sample )

    group.2.random.sample <-
        rnorm(
            group.sample.size,
            mean = population.mean,
            sd = sqrt( population.variance )
        )

    group.2.sample.mean <-
        mean( group.2.random.sample )

    group.3.random.sample <-
        rnorm(
            group.sample.size,
            mean = population.mean + 2,
            sd = sqrt( population.variance )
        )

    group.3.sample.mean <-
        mean( group.3.random.sample )

    error.sum.of.squares <-
        sum(
            (group.1.random.sample - group.1.sample.mean)^2,
            (group.2.random.sample - group.2.sample.mean)^2,
            (group.3.random.sample - group.3.sample.mean)^2
        )

    outcome.vector[ replication.index ] <-
        error.sum.of.squares /
        population.variance
}
```

Part (b): Histogram

Construct a histogram of the random values in the outcome vector you generated in part (a). Then superimpose the density curve for a chi-squared distribution using the degrees of freedom that you determined in problem 4. How well does the density curve fit the observed data?

Solution

```r
hist(
    outcome.vector,
    prob = TRUE,
    xlim = c(0, 80),
    ylim = c(0, 0.06),
    main = "Histogram of Error Sums of Squares",
    xlab = "SSTr",
    ylab = "Density",
    col = "aquamarine2",
    breaks = 50
)

curve(
    dchisq(x, df = degrees.of.freedom),
    lty = "dashed",
    lwd = 3,
    col = "black",
    add = TRUE
)
```



End of problem 5

Problem 6: Putting It All Together

Now we will explore the sampling distribution of the random variable V:

$$F = \frac{MSTr}{MSE}$$

In this problem, we will consider the case where the null hypothesis is true, so that all the group-specific population expected values are equal.

Consider this experimental protocol:

- We have three groups, so $I = 3$.
- Each group is normally distributed, with an expected value of 25 and a variance of 10.
- We draw a sample of size $J = 15$ from each group.

Part (a): Numerator degrees of freedom

For this experimental protocol, what are the numerator degrees of freedom for the test statistic F? Report your answer with one or two sentences.

Solution

```
numerator.degrees.of.freedom <-
    number.of.groups  - 1

numerator.degrees.of.freedom
```

```
## [1] 2
```

Part (b): Denominator degrees of freedom

For this experimental protocol, what are the denominator degrees of freedom for the test statistic F? Report your answer with one or two sentences.

Solution

```
denominator.degrees.of.freedom <-
    number.of.groups * (group.sample.size - 1)

denominator.degrees.of.freedom
```

```
## [1] 42
```

Part (c): Simulation

Now we're going to simulate the sampling distribution of the random variable F.

For each simulation replication:

- First, draw three random samples from the three groups.

- Next, calculate the F statistic for this random data.
- Store this value in the outcome vector.

When the simulation is complete, the outcome vector will be populated with random values of the test statistic F.

There's nothing to report for this part, but write your code clearly so the TAs can understand what you're doing.

Solution

```
population.mean <- 25

population.variance <- 10

number.of.groups <- 3

group.sample.size <- 15

number.of.replications <- 10000

outcome.vector <- numeric( number.of.replications )

for( replication.index in 1:number.of.replications ) {

    group.1.random.sample <-
        rnorm(
            group.sample.size,
            mean = population.mean,
            sd = sqrt( population.variance )
        )

    group.1.sample.mean <-
        mean( group.1.random.sample )

    group.1.sample.variance <-
        var( group.1.random.sample )

    group.2.random.sample <-
        rnorm(
            group.sample.size,
            mean = population.mean,
            sd = sqrt( population.variance )
        )

    group.2.sample.mean <-
        mean( group.2.random.sample )

    group.2.sample.variance <-
        var( group.2.random.sample )


    group.3.random.sample <-
        rnorm(
```

```
            group.sample.size,
            mean = population.mean,
            sd = sqrt( population.variance )
        )

    group.3.sample.mean <-
        mean( group.3.random.sample )

    group.3.sample.variance <-
        var( group.3.random.sample )


    group.mean.vector <-
        c(
            group.1.sample.mean,
            group.2.sample.mean,
            group.3.sample.mean
        )

    group.variance.vector <-
        c(
            group.1.sample.variance,
            group.2.sample.variance,
            group.3.sample.variance
        )

    mean.square.for.treatments <-
        var( group.mean.vector )

    mean.square.for.errors <-
        mean( group.variance.vector )

    f.statistic <-
        group.sample.size * mean.square.for.treatments /
        mean.square.for.errors

    outcome.vector[ replication.index ] <-
        f.statistic
}
```

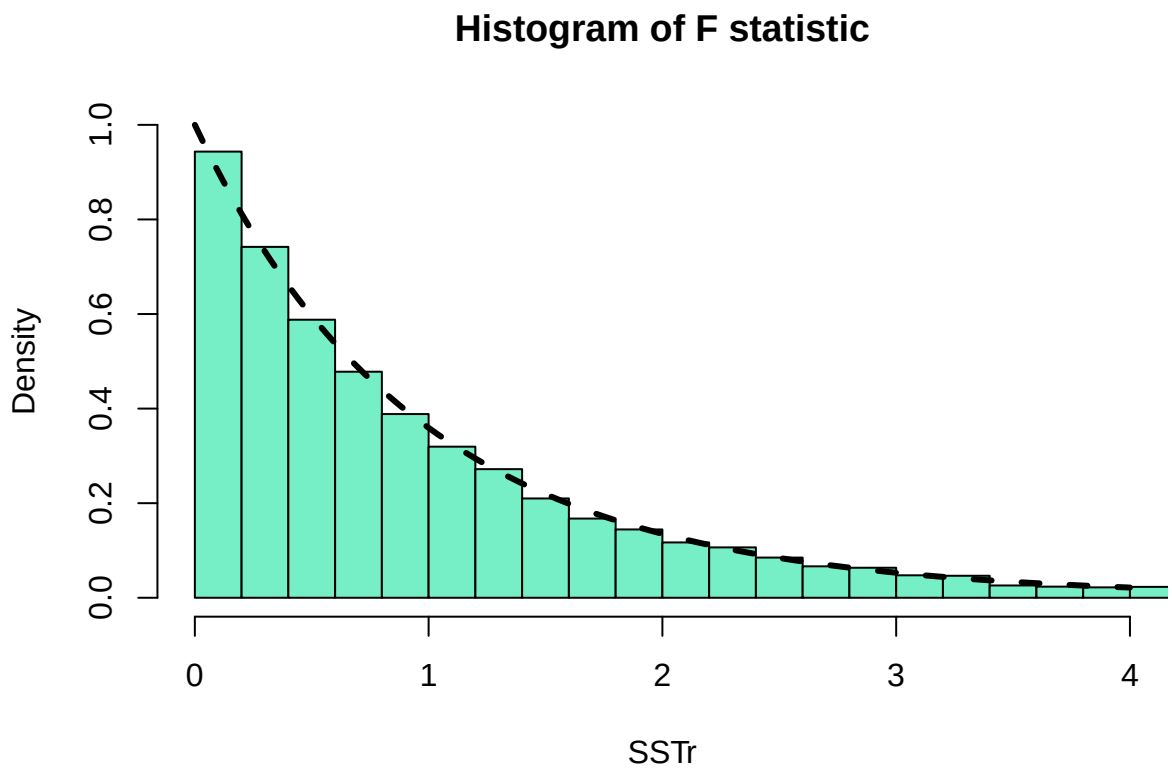
Part (c): Histogram

Construct a histogram of the random values in the outcome vector you generated in part (b). Then superimpose the density curve for an F distribution using the numerator and denominator degrees of freedom from parts (a) and (b). How well does the density curve fit the observed data?

Solution

```
hist(
    outcome.vector,
    prob = TRUE,
    xlim = c(0, 4),
    ylim = c(0, 1),
```

```
    main = "Histogram of F statistic",
    xlab = "SSTr",
    ylab = "Density",
    col = "aquamarine2",
    breaks = 50
)

curve(
    df(
        x,
        df1 = numerator.degrees.of.freedom,
        df2 = denominator.degrees.of.freedom
    ),
    lty = "dashed",
    lwd = 3,
    col = "black",
    add = TRUE
)
```



End of problem 6

Problem 7: Constructing the Test

In this problem and the next, we will conduct an ANOVA test on a set of data.

This data consists of three groups, each of which has 40 observations.

Here are some variables for you:

```
number.of.groups <- 3

group.sample.size <- 40
```

The test should be calibrated so that the probability of incorrectly rejecting the null hypothesis when it is true is 10%.

Part (a): Significance level

Determine the significance level of this hypothesis test. Report your result with one or two sentences.

Solution

The significance level of the test is the Type I error probability, and by definition this is the probability of incorrectly rejecting the null hypothesis when it is true. Thus, the significance level of this test is 10%.

```
significance.level <- 0.10
```

Part (b): Numerator degrees of freedom

Calculate the numerator degrees of freedom for the F test. Report your result using a `cat()` statement.

Solution

```
numerator.degrees.of.freedom <-
    number.of.groups - 1

cat( "Numerator degrees of freedom:",
     numerator.degrees.of.freedom )
```

```
## Numerator degrees of freedom: 2
```

Part (c): Denominator degrees of freedom

Calculate the denominator degrees of freedom for the F test. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

```
denominator.degrees.of.freedom <-
    number.of.groups * (group.sample.size - 1)

cat( "Denominator degrees of freedom:",
     denominator.degrees.of.freedom )
```

```
## Denominator degrees of freedom: 117
```

Part (d): Critical value

Calculate the critical value for this hypothesis test. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

```
critical.value <-
    qf(
        significance.level,
        df1 = numerator.degrees.of.freedom,
        df2 = denominator.degrees.of.freedom,
        lower.tail = FALSE
    )

cat( "Critical value:",
     round( critical.value, 5 ) )
```

```
## Critical value: 2.3485
```

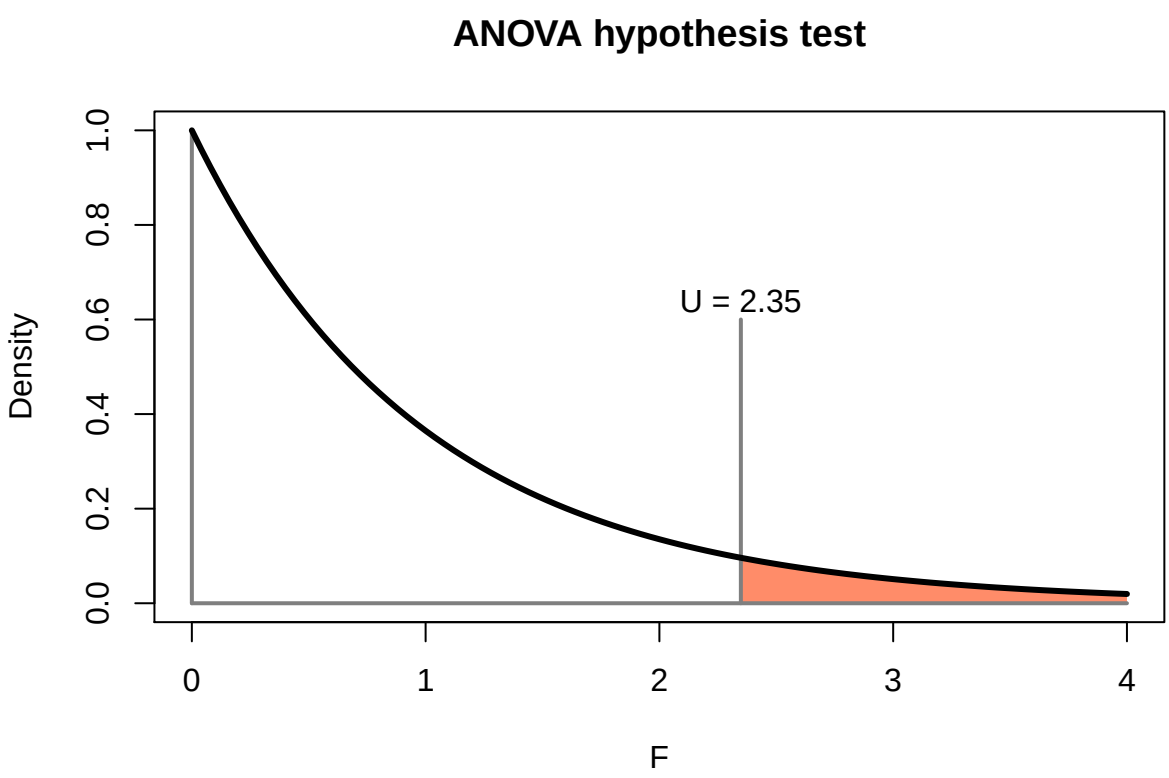
Part (e): Visualizing the F test

Draw a graph of this hypothesis test. Draw the density curve of the appropriate F test, then shade under the rejection region and use a vertical line with text annotation to indicate the critical value.

Solution

```
plot(
    x = NULL,
    xlim = c(0, 4),
    ylim = c(0, 1),
    main = "ANOVA hypothesis test",
    xlab = "F",
    ylab = "Density"
)

shade.under.f.density.curve(
    initial.x = critical.value,
    final.x = 4,
    df1 = numerator.degrees.of.freedom,
    df2 = denominator.degrees.of.freedom,
    fill.color = "salmon1"
)


segments(
    x0 = 0, y0 = 0,
    x1 = 4, y1 = 0,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)
```

```r
segments(
    x0 = 0, y0 = 0,
    x1 = 0, y1 = 1,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)


segments(
    x0 = critical.value, y0 = 0,
    x1 = critical.value, y1 = 0.6,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)


text(
    x = critical.value,
    y = 0.64,
    labels =
        paste( "U =", round( critical.value, 2 ) )
)


curve(
    df(
        x,
        df1 = numerator.degrees.of.freedom,
        df2 = denominator.degrees.of.freedom
    ),
    lty = "solid",
    lwd = 3,
    col = "black",
    add = TRUE
)
```


ANOVA hypothesis test
Density
0.0
0.2
0.4
0.6
0.8
1.0
U = 2.35
0
1
2
3
4
F

End of problem 7

Problem 8

The data for this problem is contained in 3 vectors:

- The vector `group.a.data.vector` contains the data for Group A.
- The vector `group.b.data.vector` contains the data for Group B.
- The vector `group.c.data.vector` contains the data for Group C.

Each vector consists of 40 observations.

Here are some variables for you:

```
number.of.groups <- 3

group.sample.size <- 40
```

Part (a)

Calculate the sample means of each of `group.a.data.vector`, `group.b.data.vector`, and `group.c.data.vector`. Report each sample mean using a separate `cat()` statement, rounding to 5 decimal places.

Solution

```
group.a.sample.mean <-
    mean( group.a.data.vector )

cat( "Group A sample mean:",
     round( group.a.sample.mean, 5 ) )
```

```
## Group A sample mean: 41.08325
```

```
group.b.sample.mean <-
    mean( group.b.data.vector )

cat( "Group B sample mean:",
     round( group.b.sample.mean, 5 ) )
```

```
## Group B sample mean: 41.99004
```

```
group.c.sample.mean <-
    mean( group.c.data.vector )

cat( "Group C sample mean:",
     round( group.c.sample.mean, 5 ) )
```

```
## Group C sample mean: 40.27619
```

Part (b): Calculating the grand mean

Combine the data in these three separate group vectors into one aggregate data vector. Then calculate the grand mean for this data. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

```
aggregate.data.vector <-
    c(
        group.a.data.vector,
        group.b.data.vector,
        group.c.data.vector
    )
```

Now we can calculate the grand mean:

```
grand.mean <-
    mean( aggregate.data.vector )

cat( "Grand mean:",
     round( grand.mean, 5 ) )
```

```
## Grand mean: 41.1165
```

Part (c): Numerator of the test statistic

Calculate the numerator of the test statistic. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

First, let's construct a vector consisting of the group sample means:

```
group.mean.vector <-
    c(
        group.a.sample.mean,
        group.b.sample.mean,
        group.c.sample.mean
    )
```

Now we can calculate the sum of squares of the treatments:

```
treatment.sum.of.squares <-
    group.sample.size *
    sum(
        (group.mean.vector - grand.mean)^2
    )

cat( "Treatment sum of squares:",
     round( treatment.sum.of.squares, 5 ) )
```

```
## Treatment sum of squares: 58.81173
```

Now we can calculate the mean square for treatments:

```
mean.square.for.treatments <-
    treatment.sum.of.squares /
    (number.of.groups - 1)

cat( "Mean square for treatments:",
     round( mean.square.for.treatments, 5 ) )
```

```
## Mean square for treatments: 29.40586
```

The other approach is to calculate the sample variance of the group means, and multiply by the group sample size.

```
mean.square.for.treatments <-
    group.sample.size *
    var( group.mean.vector )

cat( "Mean square for treatments:",
     round( mean.square.for.treatments, 5 ) )
```

```
## Mean square for treatments: 29.40586
```

Part (d): Denominator of the test statistic

Calculate the denominator of the test statistic.

Solution

You can also construct a vector of the group sample variances, and then take the mean:

```
group.sample.variance.vector <-
    c(
        var( group.a.data.vector ),
        var( group.b.data.vector ),
        var( group.c.data.vector )
    )

mean.square.for.error <-
    mean( group.sample.variance.vector )

cat( "Mean square for error:",
     round( mean.square.for.error, 5 ) )
```

```
## Mean square for error: 14.33206
```

Part (e): Calculating the test statistic

Calculate the value of the test statistic. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

```
f.statistic <-
    mean.square.for.treatments /
    mean.square.for.error

cat( "F statistic:",
     round( f.statistic, 5 ) )
```

```
## F statistic: 2.05175
```

Part (f): Conducting the hypothesis test

Using the critical value you calculated in problem 7 and the observed value of the test statistic, conduct a test of the ANOVA null hypothesis $H_0 : \mu_1 = \mu_2 = \mu_3$. Report your conclusions using a few sentences.

Solution

The observed value of the test statistic is $F = 2.05$, and this is not greater than the critical value. Thus, we do not reject the null hypothesis, and we conclude that this data does not constitute evidence against the ANOVA null hypothesis.

Part (g): Visualizing the test

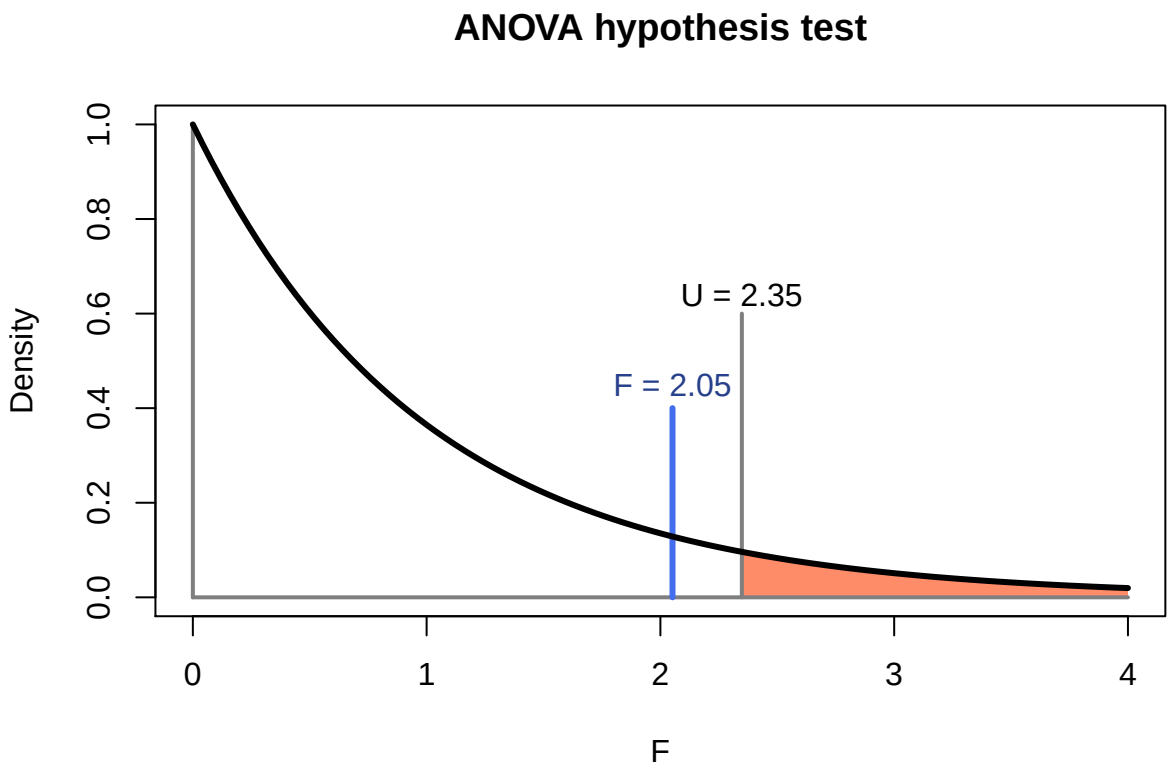
Copy your code from problem 7, part (e). Then add in a vertical line representing the observed test statistic, and annotate it with text.

Solution

```
plot(
    x = NULL,
    xlim = c(0, 4),
    ylim = c(0, 1),
    main = "ANOVA hypothesis test",
    xlab = "F",
    ylab = "Density"
)

shade.under.f.density.curve(
    initial.x = critical.value,
    final.x = 4,
    df1 = numerator.degrees.of.freedom,
    df2 = denominator.degrees.of.freedom,
    fill.color = "salmon1"
)


segments(
    x0 = 0, y0 = 0,
    x1 = 4, y1 = 0,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)
```

```
segments(
    x0 = 0, y0 = 0,
    x1 = 0, y1 = 1,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)


segments(
    x0 = critical.value, y0 = 0,
    x1 = critical.value, y1 = 0.6,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)


text(
    x = critical.value,
    y = 0.64,
    labels =
        paste( "U =", round( critical.value, 2 ) )
)


segments(
    x0 = f.statistic, y0 = 0,
    x1 = f.statistic, y1 = 0.4,
    lty = "solid",
    lwd = 3,
    col = "royalblue2"
)

text(
    x = f.statistic,
    y = 0.45,
    labels =
        paste( "F =", round(f.statistic, 2) ),
    col = "royalblue4"
)


curve(
    df(
        x,
        df1 = numerator.degrees.of.freedom,
        df2 = denominator.degrees.of.freedom
    ),
    lty = "solid",
    lwd = 3,
    col = "black",
    add = TRUE
```

```
)
```



Part (h): Calculating a p-value

Calculate a p-value for this observed data. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

```
p.value <-
    pf(
        f.statistic,
        df1 = numerator.degrees.of.freedom,
        df2 = denominator.degrees.of.freedom,
        lower.tail = FALSE
    )

cat( "p-value:",
     round( p.value, 5 ) )
```

```
## p-value: 0.13311
```

Part (i): Using the built-in R functions

Construct a vector of group identifiers. Then use this to construct a linear model using the `lm()` function, and display the results of this model using the `anova()` function. How do the results in the ANOVA table compare with your previous calculations?

Solution

```
group.id.vector <-
    c(
        rep( "Group A", group.sample.size ),
        rep( "Group B", group.sample.size ),
        rep( "Group C", group.sample.size )
    )
```

```
linear.model <-
    lm(
        aggregate.data.vector ~ group.id.vector
    )
```

```
anova( linear.model )
```

```
## Analysis of Variance Table
##
## Response: aggregate.data.vector
##                   Df  Sum Sq Mean Sq F value Pr(>F)
## group.id.vector    2   58.81  29.406  2.0518 0.1331
## Residuals        117 1676.85  14.332
```