Final Exam Solutions

Code

MATH E-156: Mathematical Statistics

Hide

```
rm( list = ls() )
```

Be sure to load in these objects before you begin:

Hide

```
load( "Final Exam R Objects.Rdata" )
```

Problem 1: Sample Size Calculation

Gazelle wants to perform a one-sample test on the population mean. She wants to use a null hypothesis of the form $H_0 : \mu = 125$, and based on her experience she believes that the true mean is $\mu_A = 128$. She knows that the true population variance is $\sigma^2 = 65$, and she knows that the population is normally distributed. She will perform a two-sided test at the $\alpha = 0.05$ significance level.

Part (a): Calculating the sample size

Determine the sample size n that Gazelle needs to achieve 90% power.

Solution

Hide

```
mu_o.1 <- 125
mu_a.1 <- 128
pop.var.1 <- 65
pop.sd.1 <- sqrt(pop.var.1)
alpha.1 <- 0.05
beta.1 <- 1 - 0.9

sample.size.1 <- ceiling((
  (pop.sd.1 * (qnorm( 1 - alpha.1 / 2) - qnorm( beta.1))) /
    (mu_a.1 - mu_o.1)
  ) ^ 2 )

cat("Sample size:", sample.size.1)
```

```
Sample size: 76
```

Part (b): Variance of the sample mean

Using the sample size you calculated in part (a) and the information in the problem statement, calculate the variance of the sample mean. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
var.sample.mean.1 <- pop.var.1 / sample.size.1

cat("Variance of sample mean:", round(var.sample.mean.1,5))
```

```
Variance of sample mean: 0.85526
```

Part (c): Critical Values

Given the sample size you calculated in part (a) and the information in the problem statement, calculate L and U, the lower and upper critical values. Report each value using a separate `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
sample.var.1 <- pop.var.1 / sample.size.1
sample.sd.1 <- sqrt(sample.var.1)

lower.critical.value.1 <- qnorm(alpha.1 / 2,
                                mean = mu_o.1,
                                sd = sample.sd.1)

cat("Lower Critical Value:", round(lower.critical.value.1,5))
```

```
Lower Critical Value: 123.1874
```

Hide

```
upper.critical.value.1 <- qnorm(alpha.1 / 2,
                                mean = mu_o.1,
                                sd = sample.sd.1,
                                lower.tail = FALSE)

cat("\nUpper Critical Value:", round(upper.critical.value.1,5))
```

```

Upper Critical Value: 126.8126
```

Part (d): Null Hypothesis Graph

Draw a graph of the density curve for the sample distribution of the sample mean under the null hypothesis. Include the lower and upper critical values L and U, and shade under the curve for the rejection region.

Solution

Hide

```
plot(x=NULL,
     xlim = c(120, 130),
     ylim = c(0, 0.5),
     main = "Sampling Distribution Under the Null Hypothesis",
     xlab = "Sample Mean",
     ylab = "Density")

shade.under.normal.density.curve(initial.x = 120,
                                 final.x = lower.critical.value.1,
                                 curve.mean = mu_o.1,
                                 curve.sd = sample.sd.1,
                                 fill.color = "azure2")
```

Hide

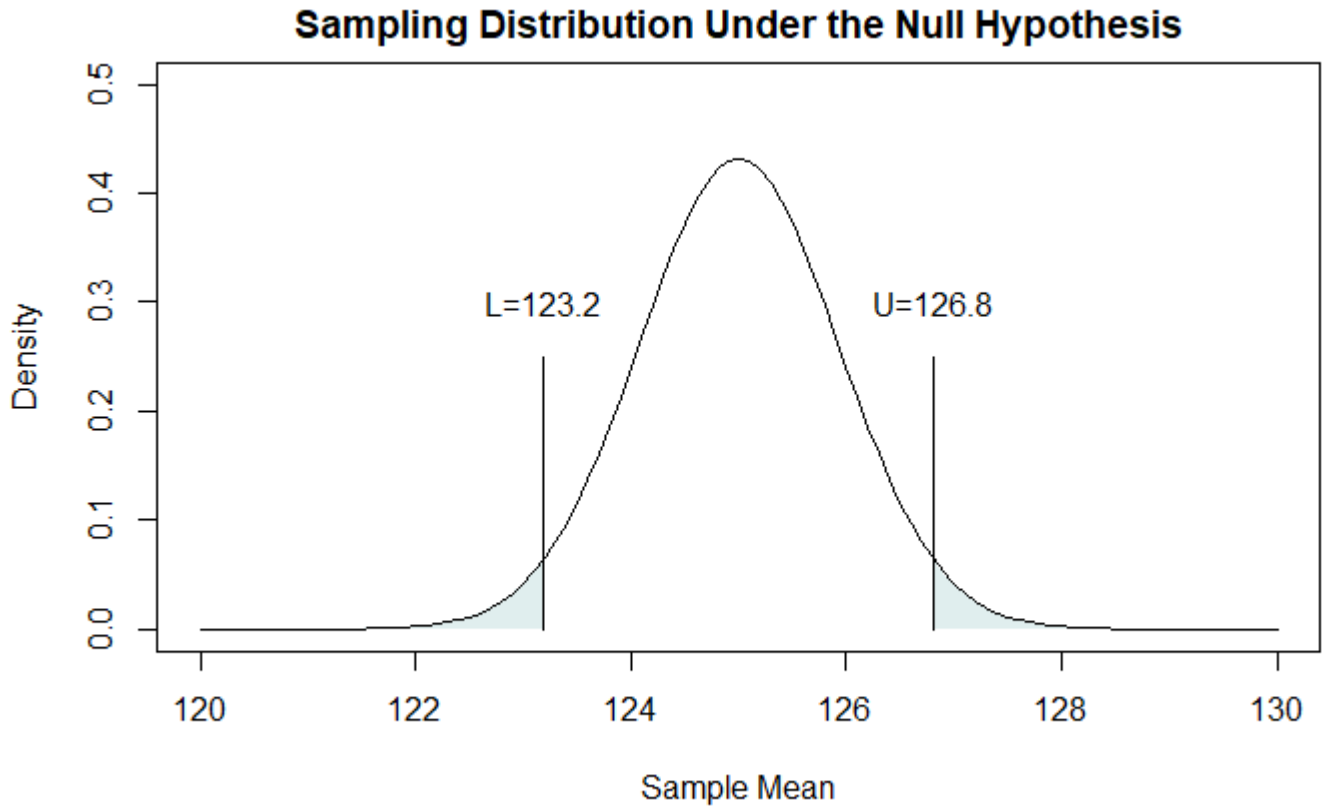
```

shade.under.normal.density.curve(initial.x = upper.critical.value.1,
                                 final.x = 130,
                                 curve.mean = mu_o.1,
                                 curve.sd = sample.sd.1,
                                 fill.color = "azure2")

curve(dnorm(x,
            mean = mu_o.1,
            sd = sample.sd.1),
      add = TRUE)
```

Hide

```

segments(lower.critical.value.1, 0,
         lower.critical.value.1, .25)

text(lower.critical.value.1, .3,
     paste0("L=", round(lower.critical.value.1,1)))
```

Hide

```

segments(upper.critical.value.1, 0,
         upper.critical.value.1, .25)

text(upper.critical.value.1, .3,
     paste0("U=", round(upper.critical.value.1,1)))
```



Part (e): Alternative Hypothesis Graph

Using your results from parts (a) through (d), construct a graph of the sampling distribution of the sample mean under the alternative hypothesis. Include the density curve for the sampling distribution of the sample mean under the null hypothesis, using a gray color. Indicate the lower and upper critical values L and U using a vertical line with a text annotation. Shade the region under the alternative density curve corresponding to a Type II error. Then, using a different color, shade the region under the alternative density curve corresponding to the statistical power. Finally, use text annotations to label the two shaded regions.

Solution

Hide

```
plot(x=NULL,
     xlim = c(120, 135),
     ylim = c(0, 0.5),
     main = "Sampling Distribution Under the Alternative Hypothesis",
     xlab = "Sample Mean",
     ylab = "Density")

shade.under.normal.density.curve(initial.x = 120,
                                 final.x = lower.critical.value.1,
                                 curve.mean = mu_o.1,
                                 curve.sd = sample.sd.1,
                                 fill.color = "azure2")
```

Hide

```
shade.under.normal.density.curve(initial.x = 120,
                                 final.x = upper.critical.value.1,
                                 curve.mean = mu_a.1,
                                 curve.sd = sample.sd.1,
                                 fill.color = "darkturquoise")

shade.under.normal.density.curve(initial.x = upper.critical.value.1,
                                 final.x = 140,
                                 curve.mean = mu_a.1,
                                 curve.sd = sample.sd.1,
                                 fill.color = "blue")
```

Hide

```
curve(dnorm(x,
            mean = mu_o.1,
            sd = sample.sd.1),
      add = TRUE,
      col = "gray70")

curve(dnorm(x,
            mean = mu_a.1,
            sd = sample.sd.1),
      add = TRUE)
```

Hide

```
segments(lower.critical.value.1, 0,
         lower.critical.value.1, .45)

text(lower.critical.value.1, .47,
     paste0("L=", round(lower.critical.value.1,1)))
```

Hide

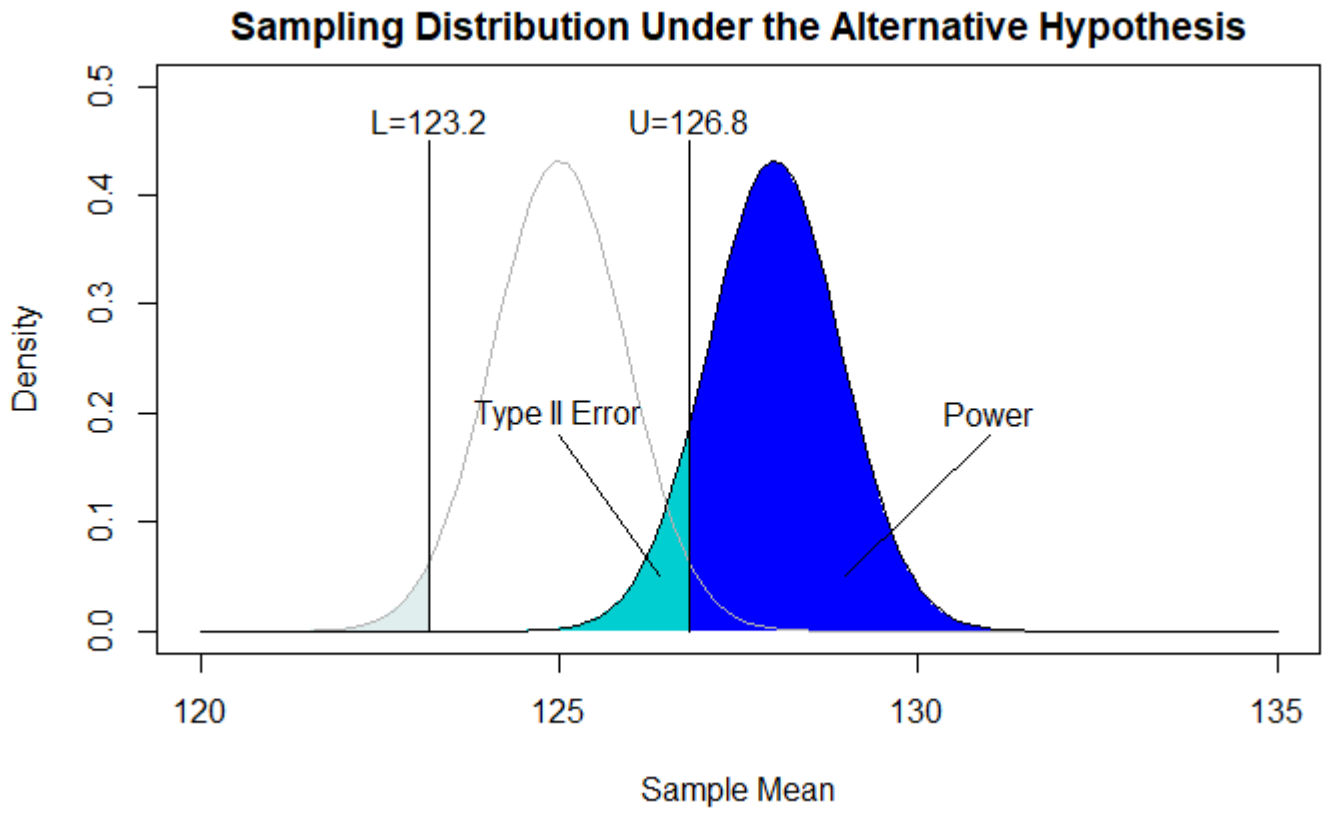
```
segments(upper.critical.value.1, 0,
         upper.critical.value.1, .45)

text(upper.critical.value.1, .47,
     paste0("U=", round(upper.critical.value.1,1)))
```

Hide

```
text(125, .2,
     "Type II Error")

segments(125, .18,
         126.4, 0.05)
```

Hide

```
text(131, .2,
     "Power")

segments(131, .18,
         129, 0.05)
```



Part (f): Type II error probability

Calculate the Type II error probability, using any method of your choice. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
type.II.error.1 <- pnorm(upper.critical.value.1,
                         mean = mu_a.1,
                         sd = sample.sd.1)

cat("Type II Error Probability:", round(type.II.error.1,5))
```

```
Type II Error Probability: 0.09958
```

Part (g): Type II error probability

Now let's check to make sure that the probability of a type II error is at most 5%.

- For each simulation replication:
    - Draw a random sample from the distribution under the alternative hypothesis using the sample size that you calculated in part (a) of problem 6.
    - Calculate the sample mean of the random sample.
    - Store the sample mean in the `outcome.vector`.

At the end of the simulation, the `outcome.vector` will be populated with random sample means drawn from the sampling distribution under the alternative hypothesis. Then use the sample proportion of the elements of `outcome.vector` that are less than the upper critical value U to estimate the type II error probability. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Note: I am proceeding with this problem assuming there is a typo in part g. It should read the "probability of a type II error is at most 10%."

Hide

```
outcome.vector.1 <- numeric()

for (i in 1:10000){

  sample.1 <- rnorm(sample.size.1,
                    mean = mu_a.1,
                    sd = pop.sd.1)

  outcome.vector.1[i] <- mean(sample.1)
}

sim.type.II.error.rate.1 <- mean(outcome.vector.1 < upper.critical.value.1)

cat("Simulated Type II Error Rate:", round(sim.type.II.error.rate.1, 5))
```

```
Simulated Type II Error Rate: 0.098
```

The simulated error rate matches the theoretical closely.

End of problem 1

Problem 2: The One-Sample t-Test

The data in `problem.2.data` comes from a normally distributed population with an unknown expected value, and we will perform a two-sided test of the null hypothesis $H_0 : \mu = 200$ using a one-sample t-test with a significance level of $\alpha = 0.05$.

Hide

```
mu_o.2 <- 200
alpha.2 <- 0.05
```

Part (a): Sample size

How many observations are in the vector `problem.2.data` ? Save this value in a variable, and report your answer with a `cat()` statement.

Solution

Hide

```
n.2 <- length(problem.2.data)
cat("Observations:", n.2)
```

```
Observations: 22
```

Part (b): Degrees of freedom

What are the appropriate degrees of freedom for this test? Save this value in a variable, and report your result using a `cat()` statement.

Solution

Hide

```
dof.2 <- n.2 - 1
cat("Degrees of Freedom:", dof.2)
```

```
Degrees of Freedom: 21
```

Part (c): Lower critical value

We wish to perform our test using a significance level of $\alpha = 0.05$. Calculate lower critical value L for this test. Store this value in a variable, and report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
lower.critical.value.2 <- qt(alpha.2 / 2,
                             dof.2)

cat("Lower Critical Value:", round(lower.critical.value.2, 5))
```

```
Lower Critical Value: -2.07961
```

Part (d): Upper critical value

We wish to perform our test using a significance level of $\alpha = 0.05$. Calculate the upper critical value U for this test. Store this value in a variable, and report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
upper.critical.value.2 <- qt(alpha.2 / 2,
                             dof.2,
                             lower.tail = FALSE)

cat("Upper Critical Value:", round(upper.critical.value.2, 5))
```

```
Upper Critical Value: 2.07961
```

Part (e): Graphing the density curve

Draw a diagram showing the density curve for the sampling distribution of the studentized t statistic for this data. Indicate the lower and upper critical values with a vertical bar, and annotate these with text. Shade underneath the curve for the rejection region.

Solution

Hide

```
plot(x=NULL,
     xlim = c(-4, 4),
     ylim = c(0, 0.5),
     main = "Sampling Distribution Using the Studentized t Statistic",
     xlab = "t Statistic",
     ylab = "Density")

shade.under.t.density.curve(initial.x = -10,
                            final.x = lower.critical.value.2,
                            degrees.of.freedom = dof.2,
                            fill.color = "azure2")
```

Hide

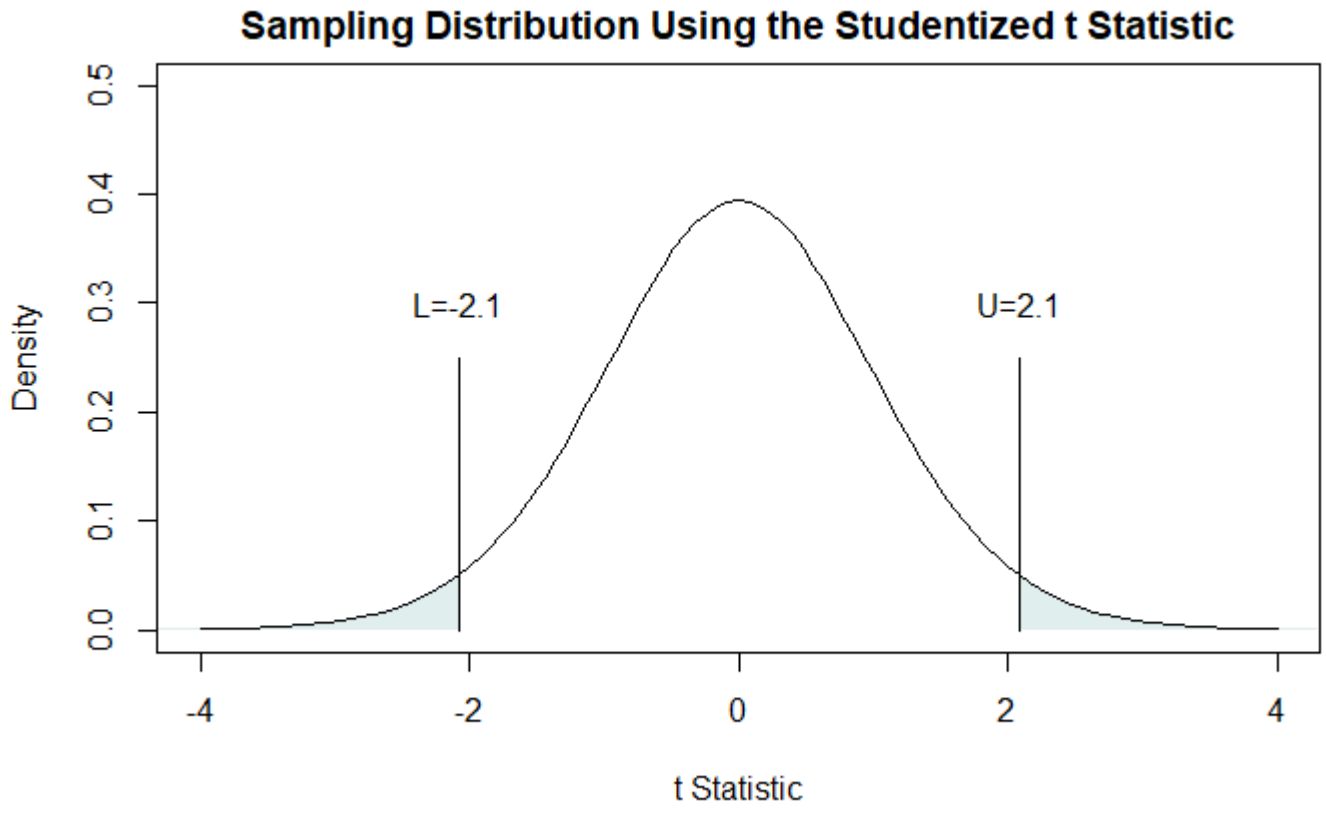
```
shade.under.t.density.curve(initial.x = upper.critical.value.2,
                                final.x = 10,
                                degrees.of.freedom = dof.2,
                                fill.color = "azure2")

curve(dt(x,
         df = dof.2),
      add = TRUE)
```

Hide

```
segments(lower.critical.value.2, 0,
         lower.critical.value.2, .25)

text(lower.critical.value.2, .3,
     paste0("L=", round(lower.critical.value.2,1)))
```

Hide

```
segments(upper.critical.value.2, 0,
         upper.critical.value.2, .25)

text(upper.critical.value.2, .3,
     paste0("U=", round(upper.critical.value.2,1)))
```



Part (f): Sample mean

What is the sample mean of `problem.2.data`? Store this value in a variable, and report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
sample.mean.2 <- mean(problem.2.data)
cat("Sample mean:", round(sample.mean.2, 5))
```

```
Sample mean: 54.39012
```

Part (g): Sample variance

What is the sample variance of `problem.2.data`? Store this value in a variable, and report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
sample.var.2 <- var(problem.2.data)
sample.sd.2 <- sqrt(sample.var.2)

cat("Sample variance:", round(sample.var.2, 5))
```

```
Sample variance: 153.2477
```

Part (h): Calculating the t-statistic

Calculate the one-sample t statistic for this data, using the null hypothesis value of $\mu_0 = 200$. Save this value in a variable, and report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
t.score.2 <- (sample.mean.2 - mu_o.2) / (sample.sd.2 /sqrt( n.2))
cat("t score:", round(t.score.2, 5))
```

```
t score: -55.17027
```

Part (i): Performing the test

Using the lower and upper critical values you calculate in parts (c) and (d), perform a two-sided test of the null hypothesis $H_0 : \mu = 200$ at the $\alpha = 0.05$ significance level.

Solution

The t score is significantly below the lower critical value therefore I reject the null hypothesis at the 0.05 significance level. With 95% confidence, mu is not equal to 200.

Part (j): Visualizing the hypothesis test

Copy your graph from part (e). Then add in a vertical line indicating the observed t statistic, and annotate it with text.

Solution

Hide

```
plot(x=NULL,
     xlim = c(-60, 4),
     ylim = c(0, 0.5),
     main = "Sampling Distribution Using the Studentized t Statistic",
     xlab = "t statistic",
     ylab = "Density")

shade.under.t.density.curve(initial.x = -10,
                            final.x = lower.critical.value.2,
                            degrees.of.freedom = dof.2,
                            fill.color = "azure2")
```

Hide

```
shade.under.t.density.curve(initial.x = upper.critical.value.2,
                                final.x = 10,
                                degrees.of.freedom = dof.2,
                                fill.color = "azure2")

curve(dt(x,
         df = dof.2),
      add = TRUE)
```

Hide

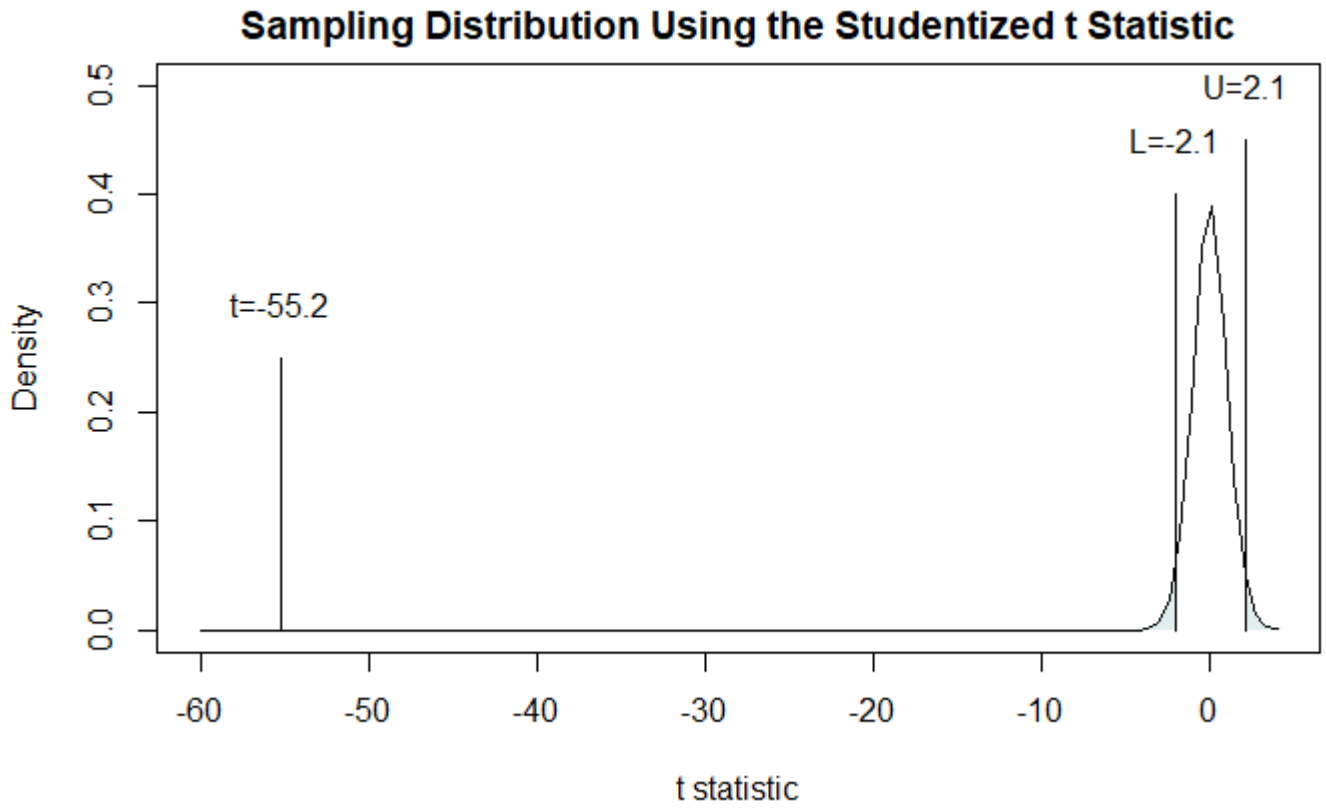
```
segments(lower.critical.value.2, 0,
         lower.critical.value.2, .4)

text(lower.critical.value.2, .45,
     paste0("L=", round(lower.critical.value.2,1)))
```

Hide

```
segments(upper.critical.value.2, 0,
         upper.critical.value.2, .45)

text(upper.critical.value.2, .5,
     paste0("U=", round(upper.critical.value.2,1)))
```

Hide

```
segments(t.score.2, 0,
         t.score.2, .25)

text(t.score.2, .3,
     paste0("t=", round(t.score.2,1)))
```



Part (k): Confidence interval

Construct a two-sided 95% confidence interval for the true population expected value. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
lower.ci.2 <- sample.mean.2 +
  qt( alpha.2 / 2, df = dof.2) *
  sqrt( sample.var.2 / n.2)
cat("Lower Confidence Interval End Point:", round(lower.ci.2, 5))
```

```
Lower Confidence Interval End Point: 48.90143
```

Hide

```
upper.ci.2 <- sample.mean.2 +
  qt( alpha.2 / 2, df = dof.2, lower.tail = FALSE) *
  sqrt( sample.var.2 / n.2)
cat("Lower Confidence Interval End Point:", round(upper.ci.2, 5))
```

```
Lower Confidence Interval End Point: 59.87881
```

The null hyposethsis of mu=200 is outside of the confidence interval therefore I again reject the null hypothesis.

Part (l): p-value

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

Solution

Hide

```
p.value.2 <- 2 * pt(t.score.2, dof.2)

cat("p value:", round(p.value.2, 5))
```

```
p value: 0
```

The p-value is so small that the software is rounding it to 0 when I specify to round to 5 decimal places. I again, reject the null hypothesis because the p value is less than alpha.

Part (m): Built-in R function

Conduct the hypothesis test using the built-in R function `t.test`. How do the results of this analysis compare with your work in the previous parts of this problem?

Solution

Hide

```
t.test(x = problem.2.data,
       mu = mu_o.2,
       conf.level = 1 - alpha.2)
```

```
    One Sample t-test

data:  problem.2.data
t = -55.17, df = 21, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 200
95 percent confidence interval:
 48.90143 59.87881
sample estimates:
mean of x
 54.39012
```

The results of the t.test function match my calculations for the t score, degrees of freedom, p-value, confidence interval end points and mean of the sample perfectly.

End of problem 2

Problem 3: Two-Sample t-Test

The two vectors `problem.3.group.1.data` and `problem.3.group.2.data` contain data from two populations with equal variances. In this problem, we will construct and perform a two-sample t-test to compare the population means, using a two-tailed test with a significance level of $\alpha = 0.05$.

Hide

```
alpha.3 <- 0.05
```

Part (a): Sample sizes

Determine the sample size of the data for Group 1, save it in a variable, and report it using a `cat()` statement. Then determine the sample size of the data for Group 2, save it in a variable, and report it using a `cat()` statement.

Solution

Hide

```
n.group.1.3 <- length(problem.3.group.1.data)
cat("Group 1 sample size:", n.group.1.3)
```

```
Group 1 sample size: 33
```

Hide

```
n.group.2.3 <- length(problem.3.group.2.data)
cat("\nGroup 2 sample size:", n.group.2.3)
```

```

Group 2 sample size: 37
```

Part (b): Degrees of freedom

Calculate the appropriate degrees of freedom for a two-sample t-test. Store this value in a variable, and report it using a `cat()` statement.

Solution

Hide

```
dof.group.1.3 <- n.group.1.3 - 1
dof.group.2.3 <- n.group.2.3 - 1
total.dof.3 <- dof.group.1.3 + dof.group.2.3

cat("Degrees of freedom:", total.dof.3)
```

```
Degrees of freedom: 68
```

Part (c): Lower critical value

Using a significance level of $\alpha = 0.05$, calculate L, the lower critical value for the test. Store this in a variable, and report it using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
lower.critical.value.3 <- qt( alpha.3 / 2, total.dof.3)
cat("Lower critical value:", round(lower.critical.value.3, 5))
```

```
Lower critical value: -1.99547
```

Part (d): Upper critical value

Using a significance level of $\alpha = 0.05$, calculate U, the upper critical value for the test. Store this in a variable, and report it using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
upper.critical.value.3 <- qt( alpha.3 / 2, total.dof.3, lower.tail = FALSE)
cat("Upper critical value:", round(upper.critical.value.3, 5))
```

```
Upper critical value: 1.99547
```

Part (e): Visualizing the sampling distribution

Draw a diagram showing the density curve for the sampling distribution of the studentized two-sample t statistic under the null hypothesis. Indicate the lower and upper critical values with a vertical bar, and annotate these with test. Shade underneath the curve for the rejection region.

Solution

Hide

```
plot(x=NULL,
     xlim = c(-4, 4),
     ylim = c(0, 0.5),
     main = "Sampling Distribution Using the Studentized t Statistic",
     xlab = "t statistic",
     ylab = "Density")

shade.under.t.density.curve(initial.x = -10,
                            final.x = lower.critical.value.3,
                            degrees.of.freedom = total.dof.3,
                            fill.color = "azure2")
```

Hide

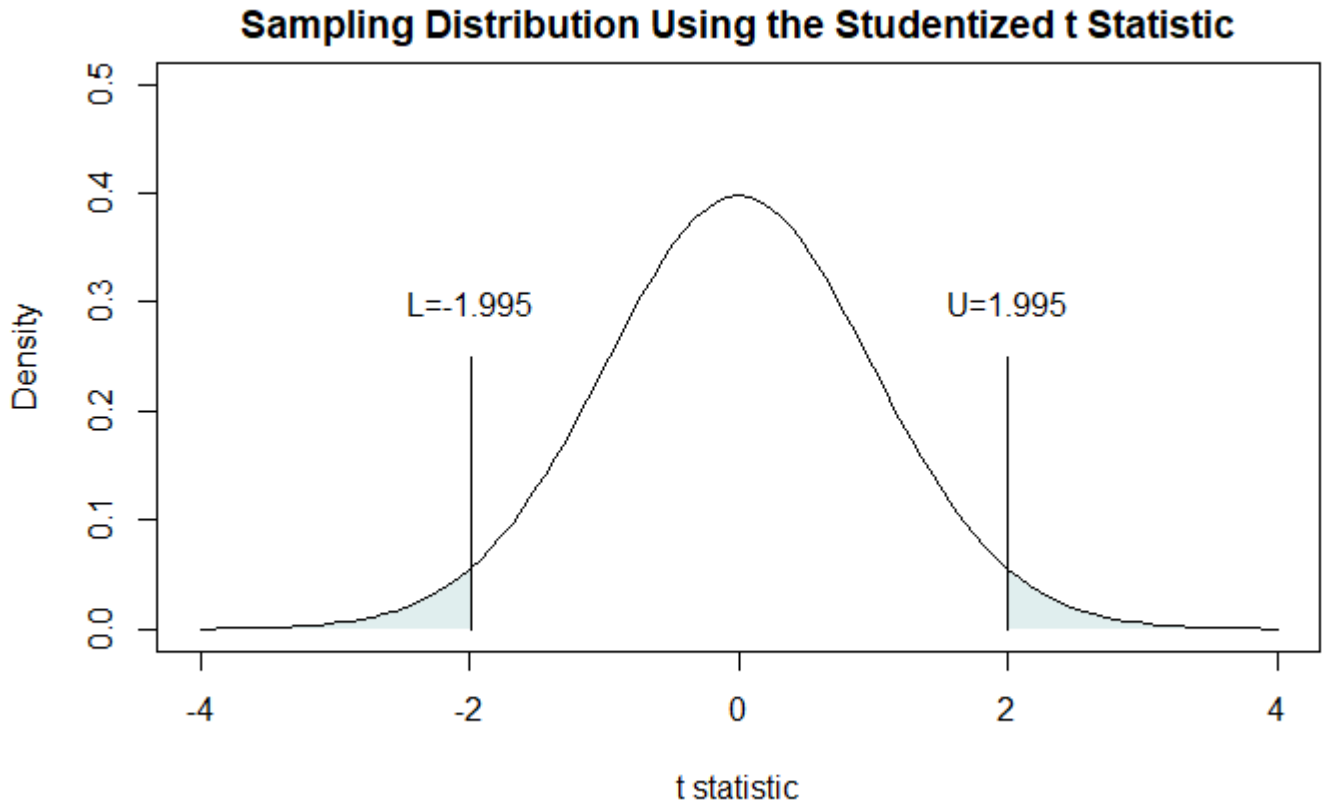
```
shade.under.t.density.curve(initial.x = upper.critical.value.3,
                                final.x = 10,
                                degrees.of.freedom = total.dof.3,
                                fill.color = "azure2")

curve(dt(x,
         df = total.dof.3),
      add = TRUE)
```

Hide

```
segments(lower.critical.value.3, 0,
         lower.critical.value.3, .25)

text(lower.critical.value.3, .3,
     paste0("L=", round(lower.critical.value.3,3)))
```

Hide

```
segments(upper.critical.value.3, 0,
         upper.critical.value.3, .25)

text(upper.critical.value.3, .3,
     paste0("U=", round(upper.critical.value.3,3)))
```



Part (f): Sample means

Calculate the sample mean for Group 1. Store this value in a variable, and report it using a `cat()` statement. Then calculate the sample mean for Group 2. Store this value in a variable, and report it using a `cat()` statement.

Solution

Hide

```
sample.mean.group.1.3 <- mean(problem.3.group.1.data)
cat("Sample mean for group 1:", round(sample.mean.group.1.3, 5))
```

```
Sample mean for group 1: 427.8966
```

Hide

```
sample.mean.group.2.3 <- mean(problem.3.group.2.data)
cat("\nSample mean for group 2:", round(sample.mean.group.2.3, 5))
```

```

Sample mean for group 2: 432.1282
```

Part (g): Sample variances

Calculate the sample variance for Group 1. Store this value in a variable, and report it using a `cat()` statement. Then calculate the sample variance for Group 2. Store this value in a variable, and report it using a `cat()` statement.

Solution

Hide

```
sample.var.group.1.3 <- var(problem.3.group.1.data)
sample.sd.group.1.3 <- sqrt(sample.var.group.1.3)
cat("Sample variance for group 1:", round(sample.var.group.1.3, 5))
```

```
Sample variance for group 1: 44.9493
```

Hide

```
sample.var.group.2.3 <- var(problem.3.group.2.data)
sample.sd.group.2.3 <- sqrt(sample.var.group.2.3)
cat("\nSample variance for group 2:", round(sample.var.group.2.3, 5))
```

```

Sample variance for group 2: 59.32295
```

Part (h): Pooled estimate of the variance

Calculate S_p, the pooled estimate of the common population variance. Store this value in a variable, and report it using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
weight.1.3 <- dof.group.1.3 / total.dof.3
weight.2.3 <- dof.group.2.3 / total.dof.3

pooled.var.3 <- weight.1.3 * sample.var.group.1.3 +
  weight.2.3 * sample.var.group.2.3

cat("Pooled Estimate of the Commom Population Variance:", round(pooled.var.3, 5))
```

```
Pooled Estimate of the Commom Population Variance: 52.55888
```

Part (i): t statistic

Calculate the t statistic for this data. Store this in a variable, and report your result using a `cat()` statement.

Solution

Hide

```
t.score.3 <- (sample.mean.group.1.3 - sample.mean.group.2.3) /
  sqrt( pooled.var.3 * (1 / n.group.1.3 + 1 / n.group.2.3))

cat("t statistic:", round(t.score.3, 5))
```

```
t statistic: -2.43777
```

Part (j): Visualizing the test statistic

Copy the code for the graph of the density curve of the test statistic under the null hypothesis from part (e). Then add a vertical line to indicate the observed t statistic, and annotate it with text.

Solution

Hide

```
plot(x=NULL,
     xlim = c(-4, 4),
     ylim = c(0, 0.5),
     main = "Sampling Distribution Using the Studentized t Statistic",
     xlab = "t statistic",
     ylab = "Density")

shade.under.t.density.curve(initial.x = -10,
                            final.x = lower.critical.value.3,
                            degrees.of.freedom = total.dof.3,
                            fill.color = "azure2")
```

Hide

```

shade.under.t.density.curve(initial.x = upper.critical.value.3,
                                final.x = 10,
                                degrees.of.freedom = total.dof.3,
                                fill.color = "azure2")

curve(dt(x,
         df = total.dof.3),
      add = TRUE)
```

Hide

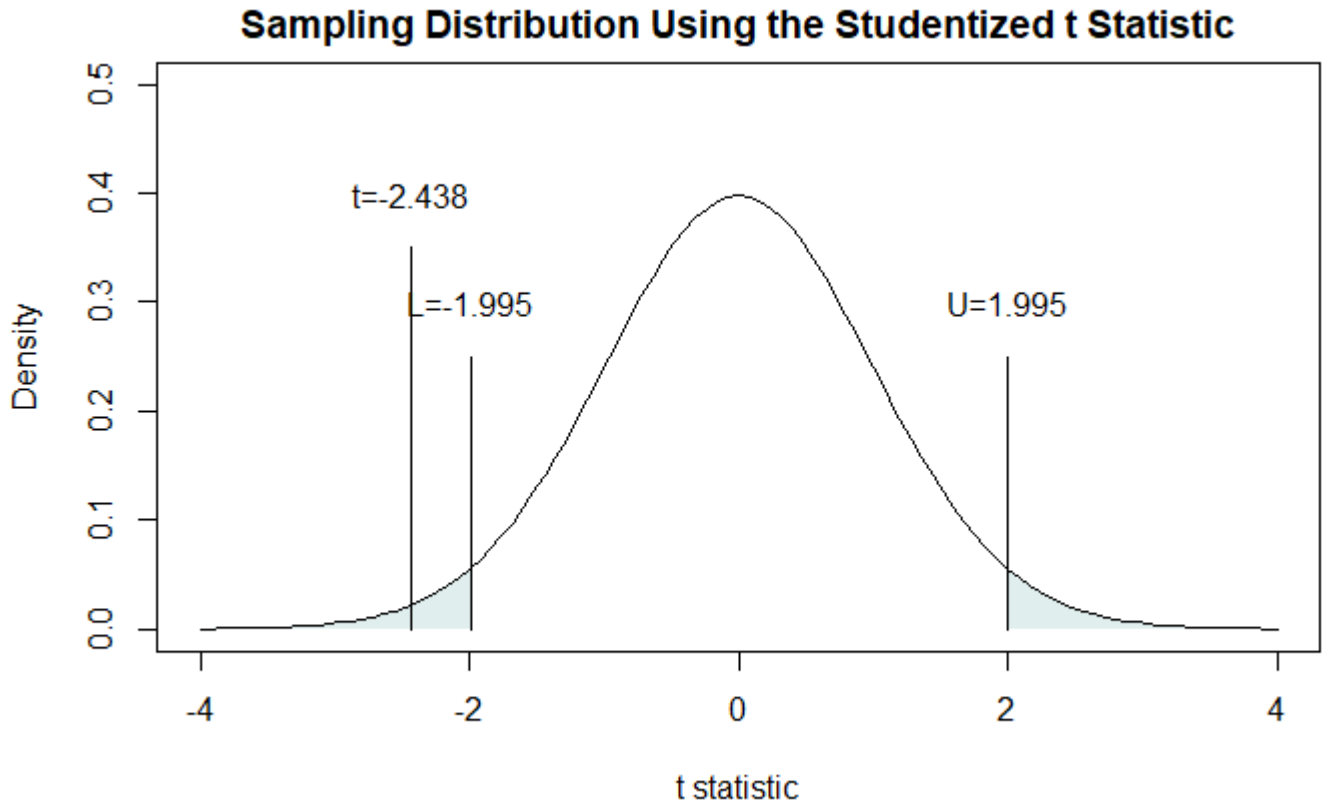
```

segments(lower.critical.value.3, 0,
         lower.critical.value.3, .25)

text(lower.critical.value.3, .3,
     paste0("L=", round(lower.critical.value.3,3)))
```

Hide

```
segments(upper.critical.value.3, 0,
         upper.critical.value.3, .25)

text(upper.critical.value.3, .3,
     paste0("U=", round(upper.critical.value.3,3)))
```

Hide

```
segments(t.score.3, 0,
         t.score.3, .35)

text(t.score.3, .4,
     paste0("t=", round(t.score.3,3)))
```



Part (k): Conducting the test

Using the critical values you calculated in parts (c) and (d), perform a two-sided test of the null hypothesis $h_0 : \mu_1 = \mu_2$ using a significance level of $\alpha = 0.05$. Does this data constitute strong evidence against the null hypothesis? Explain your answer with one or two sentences.

Solution

The t score is lower than the lower critical value therefore I reject the null hypothesis at the 0.05 significance level. The means of the groups are likely not equal.

Part (l): Confidence interval

Calculate the lower endpoint of the 95% confidence interval for the difference of the true population expected values. Report your result using a `cat()` statement, rounding to 5 decimal places. Then calculate the upper endpoint of the 95% confidence interval for the difference of the true population expected values. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
lower.ci.3 <- (sample.mean.group.1.3 - sample.mean.group.2.3) +
  qt( alpha.3 / 2,
      total.dof.3) *
  sqrt( pooled.var.3 * (1/n.group.1.3 + 1/n.group.2.3))
cat("Lower confidence interval end point:", round(lower.ci.3, 5))
```

```
Lower confidence interval end point: -7.69547
```

Hide

```
upper.ci.3 <- (sample.mean.group.1.3 - sample.mean.group.2.3) +
  qt( alpha.3 / 2,
      total.dof.3,
      lower.tail = FALSE) *
  sqrt( pooled.var.3 * (1/n.group.1.3 + 1/n.group.2.3))
cat("\nUpper confidence interval end point:", round(upper.ci.3, 5))
```

```

Upper confidence interval end point: -0.76777
```

The confidence interval does not encompass 0 therefore I again reject the null hypothesis.

Part (m): p-value

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

Solution

Hide

```
p.value.3 <- 2 * pt(t.score.3, total.dof.3)
cat("p-value:", round(p.value.3, 5))
```

```
p-value: 0.0174
```

The p value is smaller than alpha therefore I again reject the null hypothesis.

Part (n): Built-in R function

Now use the built-in R function `t.test()` to conduct the two-sample t-test. How do the results of this analysis compare with your previous work?

Solution

Hide

```
t.test(problem.3.group.1.data, problem.3.group.2.data, var.equal = TRUE)
```

```
    Two Sample t-test

data:  problem.3.group.1.data and problem.3.group.2.data
t = -2.4378, df = 68, p-value = 0.0174
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.695474 -0.767772
sample estimates:
mean of x mean of y
 427.8966  432.1282
```

The values I calculated for the t score, degrees of freedome, p-value, upper and lower confidence interval endpoints, and sample means match the t.test() method exactly.

End of problem 3

Problem 4: Simulating the T statistic

Recall the definition of the test statistic for the two-sample t-test:

$$T = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{S_p^2 \cdot \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

The pooled variance estimator S_p^2 is:

$$S_p^2 = \left(\frac{(n_1 - 1)}{n_1 + n_2 - 2} \cdot S_1^2\right) + \left(\frac{(n_2 - 1)}{n_1 + n_2 - 2} \cdot S_2^2\right)$$

You can also write the pooled variance estimator as:

$$S_p^2 = \frac{(n_1 - 1) \cdot S_1^2 + (n_2 - 1) \cdot S_2^2}{n_1 + n_2 - 2}$$

Given all the assumptions for the two-sample t-test, we know that this test statistic T has a t-distribution with $n_1 + n_2 - 2$ degrees of freedom, but we never actually did any simulation experiments to verify that claim.

Part (a): Constructing the simulation

In this problem, we will construct a simulation of the two-sample t-test test statistic T under the null hypothesis, and show that it has a t distribution with the appropriate degrees of freedom.

For each simulation replication:

- Generate a random sample of size $n_1 = 5$ observations from a normal distribution with an expected value of $\mu_1 = 45$ and a variance of $\sigma^2 = 30$.
- Calculate the sample mean and sample variance of this random sample.
- Next, generate a second random sample of size $n_2 = 6$ observations from a normal distribution with an expected value of $\mu_2 = 45$ and a variance of $\sigma^2 = 30$.
- Calculate the sample mean and sample variance of this second random sample.
- Calculate the pooled sample variance estimate S_p^2 using the sample variances of the first and second samples.
- Calculate the two-sample t statistic for this simulation replication, and store this value in the outcome vector.

At the end of this simulation, the outcome vector should be populated with random two-sample t-statistics.

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

Solution

Hide

```
n.1.4 <- 5
n.2.4 <- 6

mu.4 <- 45

var.4 <- 30
sd.4 <- sqrt(var.4)

t.scores.4 <- numeric()

for ( j in 1:10000) {
  #step 1:
  sample.1.4 <- rnorm(n.1.4, mean = mu.4, sd = sd.4)

  #step 2:
  sample.1.mean.4 <- mean(sample.1.4)
  sample.1.var.4 <- var(sample.1.4)

  #step 3:
  sample.2.4 <- rnorm(n.2.4, mean = mu.4, sd = sd.4)

  #step 4:
  sample.2.mean.4 <- mean(sample.2.4)
  sample.2.var.4 <- var(sample.2.4)

  #step 5:
  pooled.var.4 <- (((n.1.4 - 1) * var.4) + ((n.2.4 - 1) * var.4)) /
    (n.1.4 + n.2.4 -2)

  #step 6:
  t.scores.4[j] <- (sample.1.mean.4 - sample.2.mean.4) /
    sqrt( pooled.var.4 * (1/n.1.4 + 1/n.2.4))
}
```

Part (b): Degrees of freedom

What is the appropriate value of the degrees of freedom for the t distribution for the t statistics you generated in part (a)? Report your answer with one or two sentences.

Solution

Hide

```
dof.group.1.4 <- n.1.4 - 1
dof.group.2.4 <- n.2.4 - 1
total.dof.4 <- dof.group.1.4 + dof.group.2.4
cat("Degrees of freedom:", total.dof.4)
```

```
Degrees of freedom: 9
```

The degrees of freedom for this t distribution is the sum of the degrees of freedom of each group. Each group's degrees of freedom is its length minus 1. Therefore DOF = (5-1) + (6-1) = 9.

Part (c): Sample mean

Calculate the sample mean of the values in the outcome vector from your simulation. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
sim.t.score.4 <- mean(t.scores.4)
cat("Simulated t score:", round(sim.t.score.4, 5))
```

```
Simulated t score: -0.02174
```

Part (d): Sample variance

Calculate the sample variance of the values in the outcome vector from your simulation. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
sim.t.score.var.4 <- var(t.scores.4)
cat("Variance of simulated t score:", round(sim.t.score.var.4, 5))
```

```
Variance of simulated t score: 0.98167
```

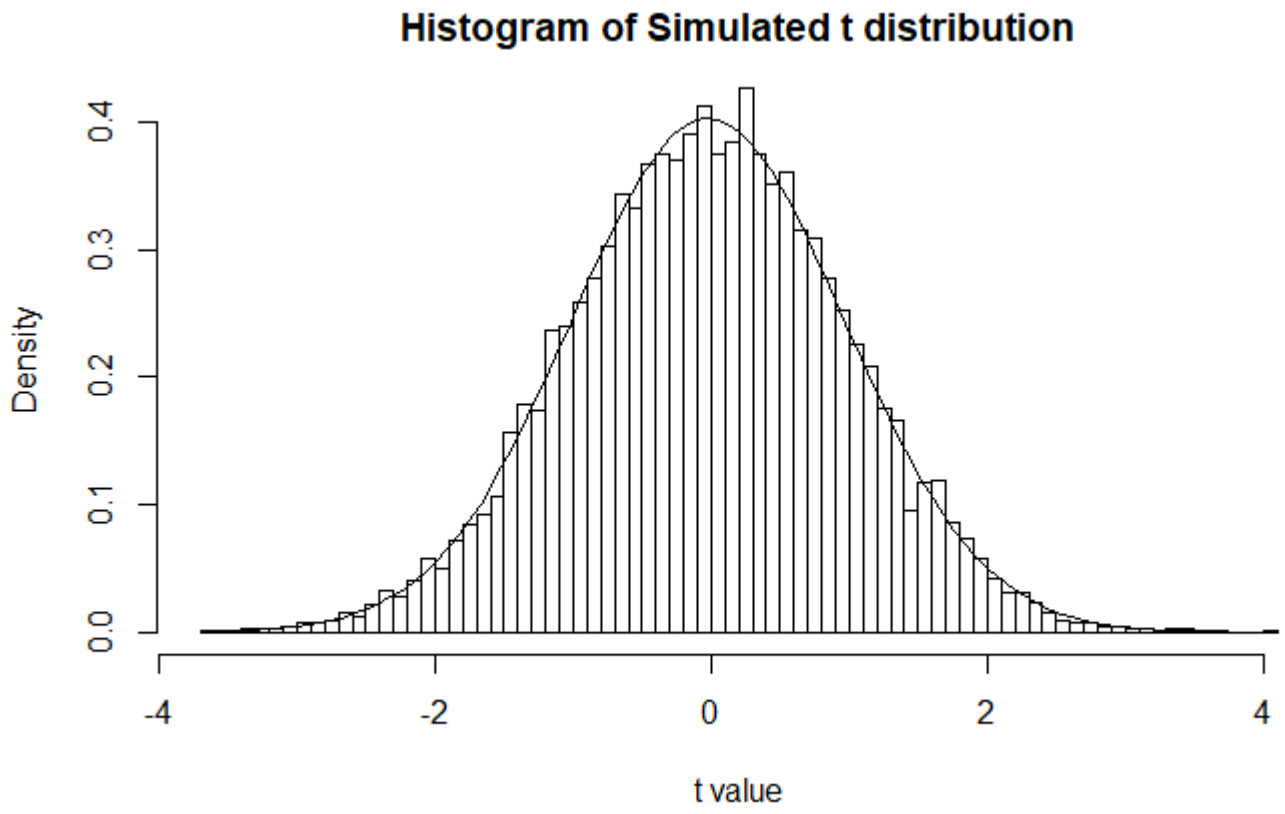
Part (e): Visualizing the normal approximation

Construct a histogram of the values in the outcome vector from your simulation in part (a). The superimpose a normal density curve, estimating the population expected value and variance by using the sample mean and sample variance from parts (c) and (d), respectively. How well does this normal density curve fit the histogram of random t statistics in the outcome vector?

Solution

Hide

```
hist(t.scores.4,
     probability = TRUE,
     main = "Histogram of Simulated t distribution",
     xlab = "t value",
     ylab = "Density",
     breaks = 100)

curve(
  dnorm(x,
        mean = sim.t.score.4,
        sd = sqrt(sim.t.score.var.4)),
      add = TRUE)
```



The histogram fits the normal curve nicely.

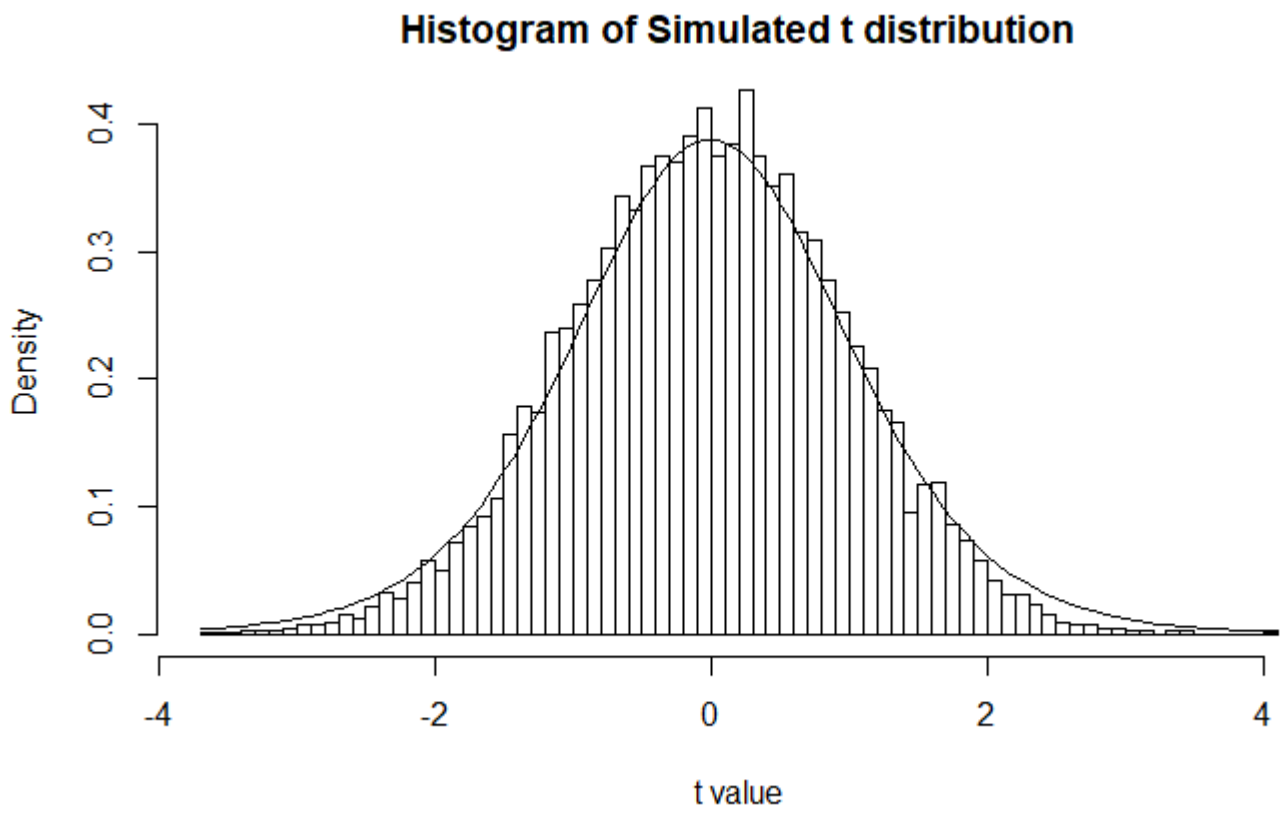
Part (f): Visualizing the t-distribution

Construct a histogram of the values in the outcome vector from your simulation in part (a). The superimpose a t density curve, using the degrees of freedom that you calculated in part (b). How well does this t density curve fit the histogram of random t statistics in the outcome vector?

Solution

Hide

```
hist(t.scores.4,
     probability = TRUE,
     main = "Histogram of Simulated t distribution",
     xlab = "t value",
     ylab = "Density",
     breaks = 100)

curve(
  dt(x,
     df = total.dof.4),
  add = TRUE)
```



The curve does not fit this histogram as well. The curve is lower than the histogram near the center and higher and wider on the ends.

End of problem 4

Problem 5: The One-Sample Variance Test

Now we're going to perform a one-sample test on a variance. The data for this problem is contained in the vector `problem.5.data` .

Part (a): Sample size

How many observations are contained in `problem.5.data` ? Save this value in a variable, and report your result using a `cat()` statement.

Solution

Hide

```
n.5 <- length(problem.5.data)
cat("Observations (n):", n.5)
```

```
Observations (n): 107
```

Part (b): Degrees of freedom

The data in `problem.5.data` comes from a normally distributed population with an unknown variance, and we will perform a two-sided test on the population variance. What are the appropriate degrees of freedom for this test? Save this value in a variable, and report your result using a `cat()` statement.

Solution

Hide

```
dof.5 <- n.5 - 1
cat("Degrees of freedom:", dof.5)
```

```
Degrees of freedom: 106
```

Part (c): Lower critical value

Suppose we wish to perform our two-sided test using the standardized variance test statistic, with a significance level of $\alpha = 0.05$. Calculate the lower critical value L for this test. Store this value in a variable, and report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
alpha.5 <- 0.05

lower.critical.value.5 <- qchisq( alpha.5 / 2,
                                  dof.5)
cat("Lower critical value:", round(lower.critical.value.5, 5))
```

```
Lower critical value: 79.40127
```

Part (d): Upper critical value

Calculate the upper critical value U for this test using a significance level of $\alpha = 0.05$. Store this value in a variable, and report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
upper.critical.value.5 <- qchisq( alpha.5 / 2,
                              dof.5,
                              lower.tail = FALSE)
cat("Upper critical value:", round(upper.critical.value.5, 5))
```

```
Upper critical value: 136.3822
```

Part (e): Graphing the sampling distribution

Draw a diagram showing the density curve for the sampling distribution of the standardized variance test statistic for this data. Indicate the lower and upper critical values with a vertical bar, and annotate these with test. Shade underneath the curve for the rejection region.

Solution

Hide

```
plot(x=NULL,
     xlim = c(0,200),
     ylim = c(0, .03),
     main = "Sampling Distribution of One Sample Test for Variance",
     xlab = "Test Statistic",
     ylab = "Density")

shade.under.chisq.density.curve(initial.x = 0,
                                final.x = lower.critical.value.5,
                                degrees.of.freedom = dof.5,
                                fill.color = "firebrick1")
```

Hide

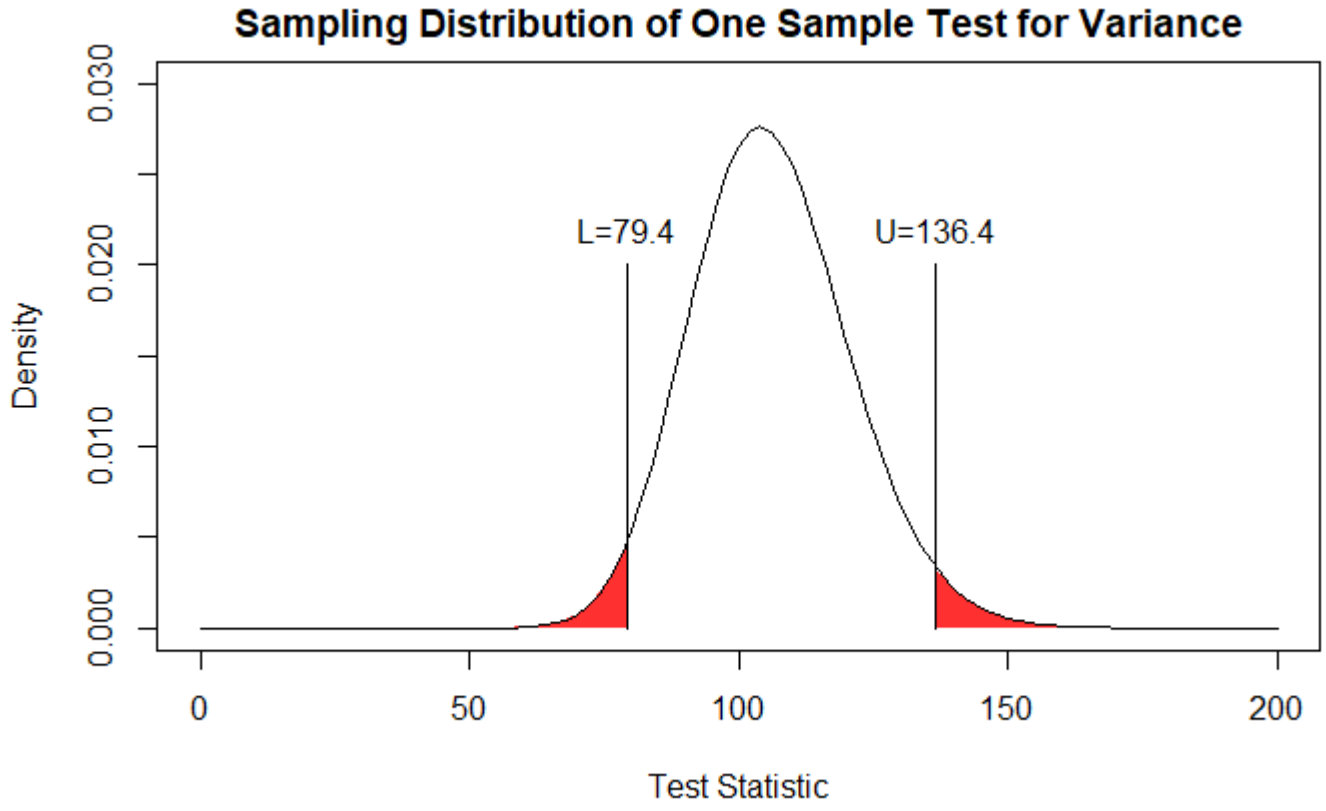
```
shade.under.chisq.density.curve(initial.x = upper.critical.value.5,
                                final.x = 200,
                                degrees.of.freedom = dof.5,
                                fill.color = "firebrick1")

curve(dchisq(x, df = dof.5),
      add = TRUE)
```

Hide

```
segments(lower.critical.value.5, 0,
         lower.critical.value.5, 0.02)

text(lower.critical.value.5, 0.022,
     paste0("L=", round(lower.critical.value.5, 1)))
```

Hide

```
segments(upper.critical.value.5, 0,
         upper.critical.value.5, 0.02)

text(upper.critical.value.5, 0.022,
     paste0("U=", round(upper.critical.value.5, 1)))
```



Part (f): Sample variance

What is the sample variance of the data in `problem.5.data` ? Store this value in a variable, and report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
sample.var.5 <- var(problem.5.data)
cat("Sample variance:", round(sample.var.5, 5))
```

```
Sample variance: 520.104
```

Part (g): Calculating the test statistic

Calculate the standardized variance test statistic for this data, using the null hypothesis value of $\sigma^2 = 630$. Save this value in a variable, and report your result using a `cat()` statement.

Solution

Hide

```
var_o.5 <- 630
test.statistic.5 <- dof.5 * sample.var.5 / var_o.5
cat("Test statistic:", round(test.statistic.5, 5))
```

```
Test statistic: 87.50956
```

Part (h): Perform the hypothesis test

Using the lower and critical values you calculated in parts (c) and (d), perform a two-sided test of the null hypothesis $H_0 : \mu = 200$ at the $\alpha = 0.05$ significance level. Report your conclusion with a few sentences.

Solution

The test statistic falls between the upper and lower critical values therefore I fail to reject the null hypothesis at the 0.05 significance level. The variance may truly be 630.

Part (i): Visualizing the test statistic

Copy your graph from part (e). Then add in a vertical line indicating the observed standardized variance test statistic, and annotate it with text.

Solution

Hide

```
plot(x=NULL,
     xlim = c(0,200),
     ylim = c(0, .03),
     main = "Sampling Distribution of One Sample Test for Variance",
     xlab = "Test Statistic",
     ylab = "Density")

shade.under.chisq.density.curve(initial.x = 0,
                                final.x = lower.critical.value.5,
                                degrees.of.freedom = dof.5,
                                fill.color = "firebrick1")
```

Hide

```
shade.under.chisq.density.curve(initial.x = upper.critical.value.5,
                                final.x = 200,
                                degrees.of.freedom = dof.5,
                                fill.color = "firebrick1")

curve(dchisq(x, df = dof.5),
      add = TRUE)
```

Hide

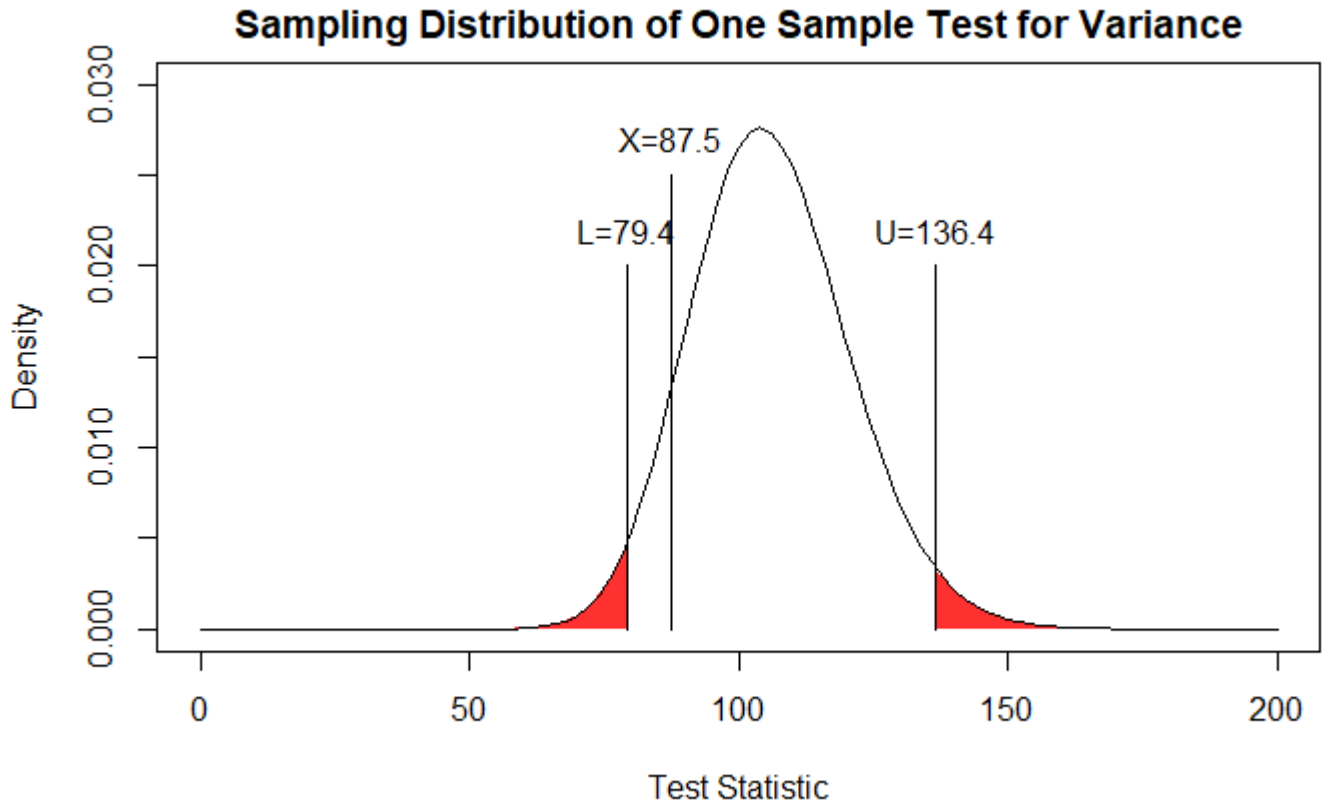
```
segments(lower.critical.value.5, 0,
         lower.critical.value.5, 0.02)

text(lower.critical.value.5, 0.022,
     paste0("L=", round(lower.critical.value.5, 1)))
```

Hide

```
segments(upper.critical.value.5, 0,
         upper.critical.value.5, 0.02)

text(upper.critical.value.5, 0.022,
     paste0("U=", round(upper.critical.value.5, 1)))
```

Hide

```
segments(test.statistic.5, 0,
         test.statistic.5, 0.025)

text(test.statistic.5, 0.027,
     paste0("X=", round(test.statistic.5, 1)))
```



Part (j): Confidence interval

Construct a two-sided 95% confidence interval for the true population variance. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
lower.ci.5 <- dof.5 * sample.var.5 /
  qchisq( 1 - alpha.5 / 2, dof.5)

upper.ci.5 <- dof.5 * sample.var.5 /
  qchisq( alpha.5 / 2, dof.5)

cat("Lower confidence interval endpoint:", round(lower.ci.5, 5))
```

```
Lower confidence interval endpoint: 404.2392
```

Hide

```
cat("\nUpper confidence interval endpoint:", round(upper.ci.5, 5))
```

```

Upper confidence interval endpoint: 694.3342
```

The variance value for the null hypothesis is within the confidence interval therefore I again fail to reject the null hypothesis.

Part (k): p-value

Calculate the p-value for this test statistic. Report your result using a `cat() statement, rounding to 5 decimal places. How does this relate to your answer for part (h)?

Solution

Hide

```
p.value.5 <- 2 * pchisq(test.statistic.5, dof.5, lower.tail = FALSE)
cat("p-value:", round(p.value.5, 5))
```

```
p-value: 1.80839
```

End of problem 5

Problem 6: Two-Sample Test on Variances

This problem is concerned with the two-sample test on variances. We will be using a two-tailed test, with a significance level of $\alpha = 0.05$.

Hide

```
alpha.6 <- 0.05
```

Part (a): Sample sizes

Determine the sample size of the data for Group 1, save it in a variable, and report it using a `cat()` statement. Then determine the sample size of the data for Group 2, save it in a variable, and report it using a `cat()` statement.

Solution

Hide

```
n.1.6 <- length(problem.6.group.1.data)
cat("Group 1 sample size:", n.1.6)
```

```
Group 1 sample size: 31
```

Hide

```
n.2.6 <- length(problem.6.group.2.data)
cat("\nGroup 2 sample size:", n.2.6)
```

```

Group 2 sample size: 47
```

Part (b): Degrees of freedom

Calculate the appropriate numerator and denominator degrees of freedom for a two-sample test on variances. Store these values in variables, and report each one separately using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
numerator.degrees.of.freedom <- n.1.6 - 1
cat("Numerator degrees of freedom:", numerator.degrees.of.freedom)
```

```
Numerator degrees of freedom: 30
```

Hide

```
denominator.degrees.of.freedom <- n.2.6 - 1
cat("\nDenominator degrees of freedom:", denominator.degrees.of.freedom)
```

```
Denominator degrees of freedom: 46
```

Part (c): Lower critical value

Using a significance level of $\alpha = 0.05$, calculate L, the lower critical value for the test. Store this value in a variable, and report it using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
lower.critical.value.6 <- qf( alpha.6 / 2,
                              df1 = numerator.degrees.of.freedom,
                              df2 = denominator.degrees.of.freedom)

cat("Lower critical value:", round(lower.critical.value.6, 5))
```

```
Lower critical value: 0.50443
```

Part (d): Upper critical value

Using a significance level of $\alpha = 0.05$, calculate U, the upper critical value for the test. Store this value in a variable, and report it using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
upper.critical.value.6 <- qf( alpha.6 / 2,
                              df1 = numerator.degrees.of.freedom,
                              df2 = denominator.degrees.of.freedom,
                              lower.tail = FALSE)

cat("Upper critical value:", round(upper.critical.value.6, 5))
```

```
Upper critical value: 1.89259
```

Part (e): Graphing the sampling distribution

Draw a diagram for the sampling distribution of the F statistic under the null hypothesis. Shade under the density curve for the rejection region, and indicate the lower and upper critical values using a vertical line annotated with text.

Solution

Hide

```
plot(x=NULL,
     xlim = c(0,3),
     ylim = c(0,1.5),
     main = "Hypothesis Test for Two Variances Using the F Distribution",
     xlab = "F",
     ylab = "Density")

shade.under.f.density.curve(0, lower.critical.value.6,
                            df1 = numerator.degrees.of.freedom,
                            df2 = denominator.degrees.of.freedom,
                            fill.color = "firebrick1")
```

Hide

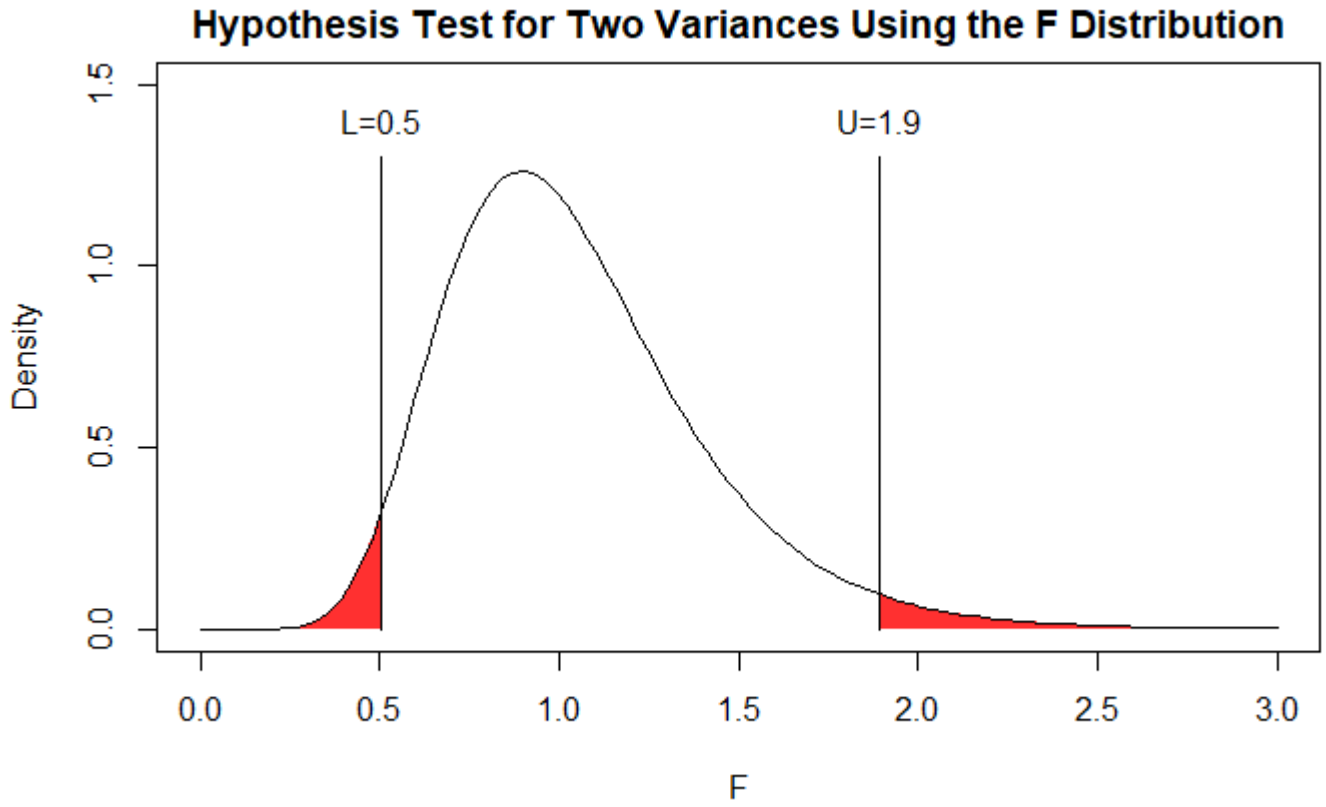
```
shade.under.f.density.curve(upper.critical.value.6, 3,
                            df1 = numerator.degrees.of.freedom,
                            df2 = denominator.degrees.of.freedom,
                            fill.color = "firebrick1")

curve(df(x,
         df1 = numerator.degrees.of.freedom,
         df2 = denominator.degrees.of.freedom),
      add = TRUE)
```

Hide

```
segments(lower.critical.value.6, 0,
         lower.critical.value.6, 1.3)

text(lower.critical.value.6, 1.4,
     paste0("L=", round(lower.critical.value.6, 1)))
```

Hide

```
segments(upper.critical.value.6, 0,
         upper.critical.value.6, 1.3)

text(upper.critical.value.6, 1.4,
     paste0("U=", round(upper.critical.value.6, 1)))
```



Part (f): Sample variances

Calculate the sample variance for Group 1. Store this value in a variable, and report it using a `cat()` statement, rounding to 5 decimal places.

Then calculate the sample variance for Group 2. Store this value in a variable, and report it using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
sample.var.1.6 <- var(problem.6.group.1.data)
cat("Sample variance group 1:", round(sample.var.1.6, 5))
```

```
Sample variance group 1: 128.4409
```

Hide

```
sample.var.2.6 <- var(problem.6.group.2.data)
cat("\nSample variance group 2:", round(sample.var.2.6, 5))
```

```

Sample variance group 2: 60.62073
```

Part (g): F statistic

Calculate the F statistic for this observed data under the null hypothesis $H_0 : \sigma_1^2 = \sigma_2^2$. Store this value in a variable, and report it using a `cat()` statement.

Solution

Hide

```
f.statistic.6 <- sample.var.1.6 / sample.var.2.6
cat("F statistic:", round(f.statistic.6, 5))
```

```
F statistic: 2.11876
```

Part (h): Visualizing the F statistic

Copy the code from part (e) for the graph of the density curve of the test statistic under the null hypothesis. Then add a vertical line to indicate the observed F statistic, and annotate it with text.

Solution

Hide

```
plot(x=NULL,
     xlim = c(0,3),
     ylim = c(0,1.5),
     main = "Hypothesis Test for Two Variances Using the F Distribution",
     xlab = "F",
     ylab = "Density")

shade.under.f.density.curve(0, lower.critical.value.6,
                            df1 = numerator.degrees.of.freedom,
                            df2 = denominator.degrees.of.freedom,
                            fill.color = "firebrick1")
```

Hide

```

shade.under.f.density.curve(upper.critical.value.6, 3,
                            df1 = numerator.degrees.of.freedom,
                            df2 = denominator.degrees.of.freedom,
                            fill.color = "firebrick1")

curve(df(x,
         df1 = numerator.degrees.of.freedom,
         df2 = denominator.degrees.of.freedom),
      add = TRUE)
```

Hide

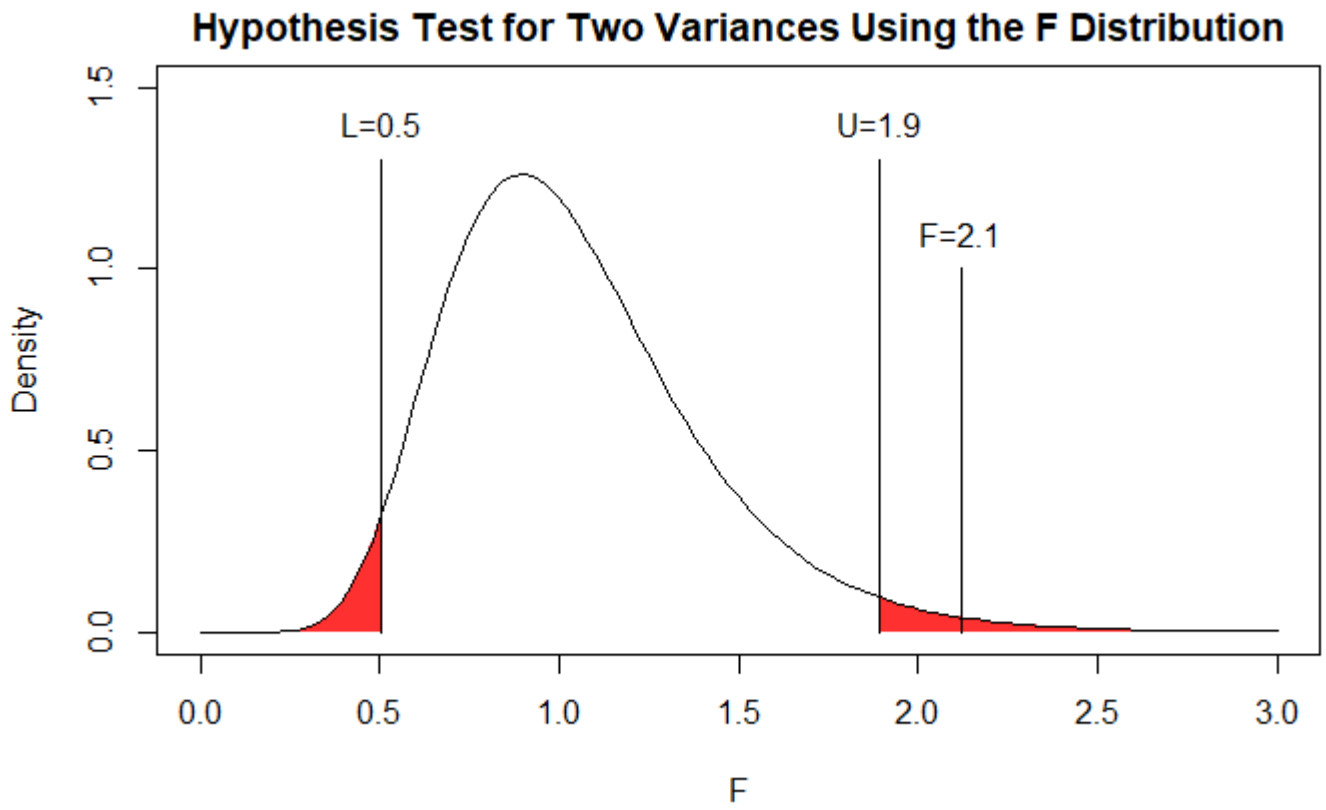
```
segments(lower.critical.value.6, 0,
         lower.critical.value.6, 1.3)

text(lower.critical.value.6, 1.4,
     paste0("L=", round(lower.critical.value.6, 1)))
```

Hide

```
segments(upper.critical.value.6, 0,
         upper.critical.value.6, 1.3)

text(upper.critical.value.6, 1.4,
     paste0("U=", round(upper.critical.value.6, 1)))
```

Hide

```
segments(f.statistic.6, 0,
         f.statistic.6, 1)

text(f.statistic.6, 1.1,
     paste0("F=", round(f.statistic.6, 1)))
```



The test statistic is higher than the upper critical value therefore I reject the null hypothesis.

Part (i): p-value

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

Solution

Hide

```
p.value.6 <- 2 * pf( f.statistic.6,
                     df1 = numerator.degrees.of.freedom,
                     df2 = denominator.degrees.of.freedom,
                     lower.tail = FALSE)
cat("p value:", round(p.value.6, 5))
```

```
p value: 0.0212
```

The p value is smaller than alpha therefore I again reject the null hypothesis.

Part (j): Built-in R function

Now use the built-in R function `var.test()` to conduct the two-sample test on variances. How do the results of this analysis compare with your previous work?

Solution

Hide

```
var.test(problem.6.group.1.data,
         problem.6.group.2.data,
         conf.level = 1 - alpha.6)
```

```

    F test to compare two variances

data:  problem.6.group.1.data and problem.6.group.2.data
F = 2.1188, num df = 30, denom df = 46, p-value = 0.0212
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 1.119505 4.200279
sample estimates:
ratio of variances
          2.118762
```

My calculated values mathc the F statistic, numerator and denominotor degress of freedome, and p-value exactly.

End of problem 6

Problem 7: Analysis of Variance

In this problem, we will perform an analysis of variance.

The data for this problem is contained in 3 vectors:

- The vector `problem.7.group.1.data` contains the data for Group 1.
- The vector `problem.7.group.2.data` contains the data for Group 2.
- The vector `problem.7.group.3.data` contains the data for Group 3.

Each vector consists of 25 observations.

Here are some variables for you:

Hide

```
number.of.groups <- 3

group.sample.size <- 25
```

Part (a): Significance level

We want to design this experiment so that it will have a Type I error rate of 10%. Determine the significance level of this hypothesis test. Report your result with one or two sentences.

Solution

Hide

```
alpha.7 <- 0.1
```

The significance level is simply the the decimal of the type I error rate percentage, so 0.1.

Part (b): Numerator degrees of freedom

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

Solution

Hide

```
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

Hide

```
denominator.degrees.of.freedom <- number.of.groups * (group.sample.size - 1)
cat("Denominator degrees of freedom:", denominator.degrees.of.freedom)
```

```
Denominator degrees of freedom: 72
```

Part (d): Critical value

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

Solution

Hide

```
upper.critical.value.7 <- qf(alpha.7,
                             df1 = numerator.degrees.of.freedom,
                             df2 = denominator.degrees.of.freedom,
                             lower.tail = FALSE)
cat("Upper critical value:", round(upper.critical.value.7, 5))
```

```
Upper critical value: 2.37782
```

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

Hide

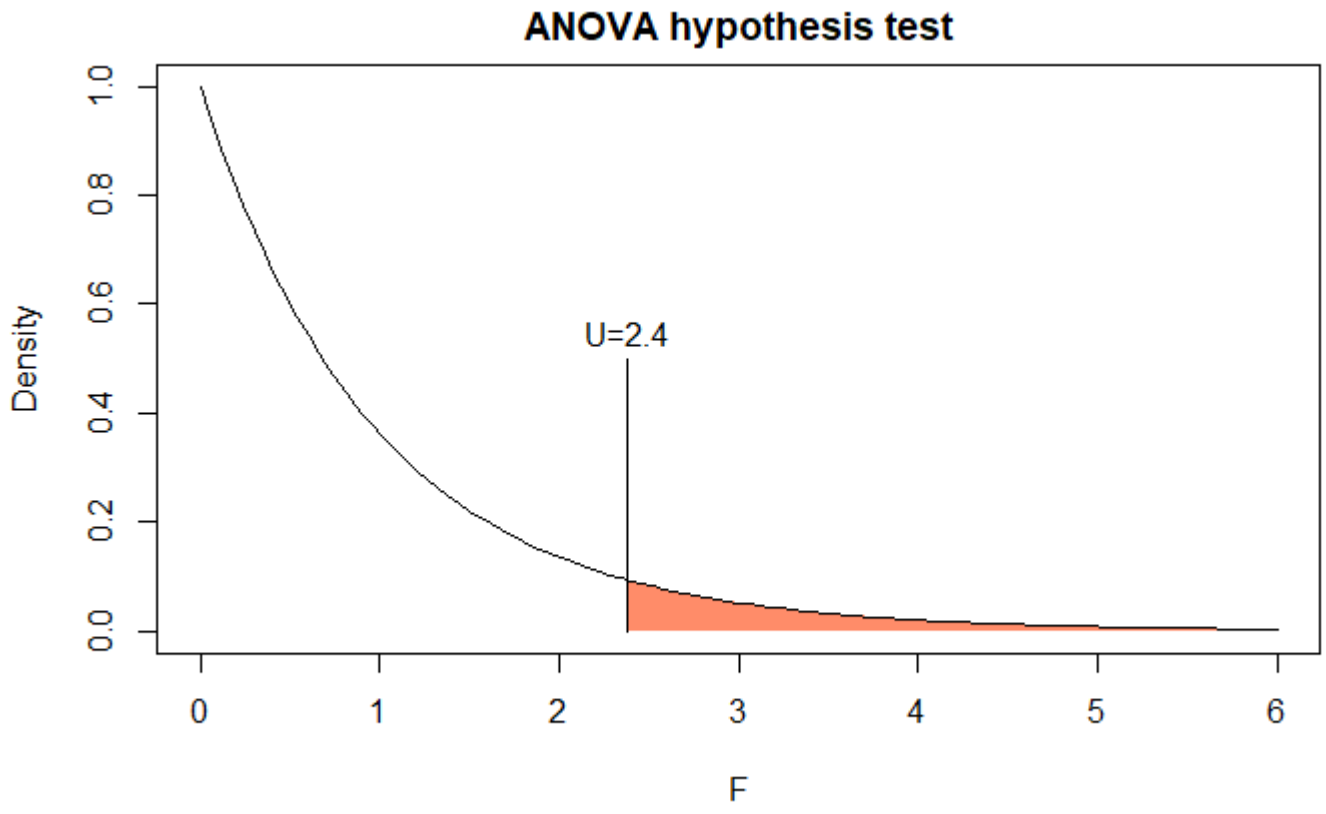
```
plot(
    x = NULL,
    xlim = c(0, 6),
    ylim = c(0, 1),
    main = "ANOVA hypothesis test",
    xlab = "F",
    ylab = "Density"
)

shade.under.f.density.curve(
    initial.x = upper.critical.value.7,
    final.x = 6,
    df1 = numerator.degrees.of.freedom,
    df2 = denominator.degrees.of.freedom,
    fill.color = "salmon1"
)
```

Hide

```
curve(
    df(
        x,
        df1 = numerator.degrees.of.freedom,
        df2 = denominator.degrees.of.freedom
    ),
    add = TRUE
)

segments(upper.critical.value.7, 0,
         upper.critical.value.7, .5)
```

Hide

```
text(upper.critical.value.7, .55,
     paste0("U=", round(upper.critical.value.7, 1)))
```



Part (f): Sample means

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

Hide

```
sample.mean.1.7 <- mean(problem.7.group.1.data)
cat("Sample mean for group 1:", round(sample.mean.1.7, 5))
```

```
Sample mean for group 1: 59.24825
```

Hide

```
sample.mean.2.7 <- mean(problem.7.group.2.data)
cat("\nSample mean for group 2:", round(sample.mean.2.7, 5))
```

```

Sample mean for group 2: 58.0937
```

Hide

```
sample.mean.3.7 <- mean(problem.7.group.3.data)
cat("\nSample mean for group 3:", round(sample.mean.3.7, 5))
```

```

Sample mean for group 3: 55.17868
```

Part (g): 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

Hide

```
aggregate.data.7 <- c(problem.7.group.1.data,
                      problem.7.group.2.data,
                      problem.7.group.3.data)
grand.mean.7 <- mean(aggregate.data.7)
cat("Grand mean:", round(grand.mean.7, 5))
```

```
Grand mean: 57.50688
```

Part (h): 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

Hide

```
aggregate.means.7 <- c(sample.mean.1.7, sample.mean.2.7, sample.mean.3.7)
mean.square.treatments.7 <- group.sample.size * var(aggregate.means.7)
cat("Numerator test statistic:", round(mean.square.treatments.7, 5))
```

```
Numerator test statistic: 109.9656
```

Part (i): Denominator of the test statistic

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

Solution

Hide

```
aggregate.var.7 <- c(
  var(problem.7.group.1.data),
  var(problem.7.group.2.data),
  var(problem.7.group.3.data)
)

mean.square.errors.7 <- mean(aggregate.var.7)
cat("Denominator test statistic:", round(mean.square.errors.7, 5))
```

```
Denominator test statistic: 44.20057
```

Part (j): Calculating the test statistic

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

Solution

Hide

```
f.statistic.7 <- mean.square.treatments.7 / mean.square.errors.7
cat("F statistic:", round(f.statistic.7, 5))
```

```
F statistic: 2.48788
```

Part (k): Conducting the hypothesis test

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 test statistic is larger than the upper critical value therefore I reject the null hypothesis at the .1 significance level. The group means are likely not equal.

Part (I): Visualizing the test statistic

Copy your code from part (e) for the graph of the sampling distribution. Then add in a vertical line representing the observed test statistic, and annotate it with text.

Solution

Hide

```
plot(
    x = NULL,
    xlim = c(0, 6),
    ylim = c(0, 1),
    main = "ANOVA hypothesis test",
    xlab = "F",
    ylab = "Density"
)

shade.under.f.density.curve(
    initial.x = upper.critical.value.7,
    final.x = 6,
    df1 = numerator.degrees.of.freedom,
    df2 = denominator.degrees.of.freedom,
    fill.color = "salmon1"
)
```

Hide

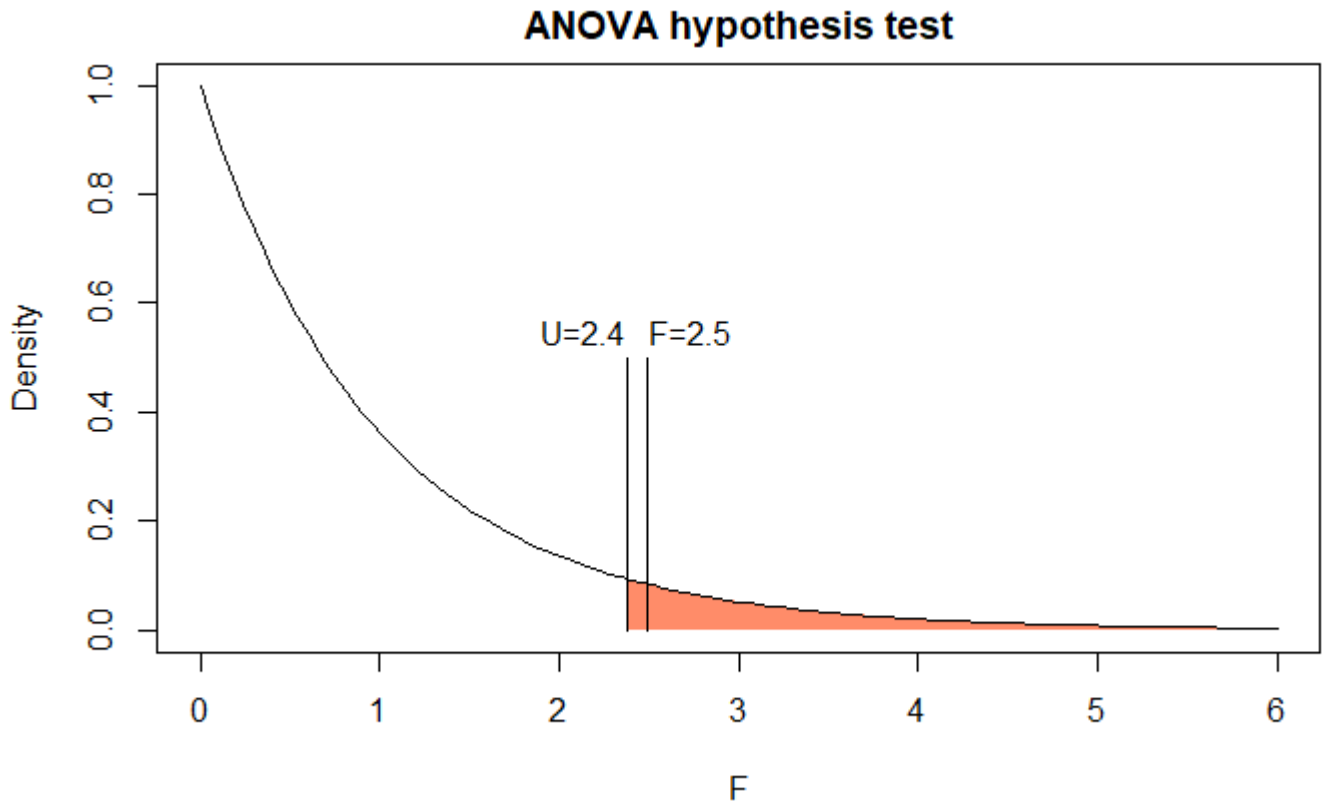
```


curve(
    df(
        x,
        df1 = numerator.degrees.of.freedom,
        df2 = denominator.degrees.of.freedom
    ),
    add = TRUE
)

segments(upper.critical.value.7, 0,
         upper.critical.value.7, .5)
```

Hide

```


text(upper.critical.value.7-.25, .55,
     paste0("U=", round(upper.critical.value.7, 1)))

segments(f.statistic.7, 0,
         f.statistic.7, .5)
```

Hide

```
text(f.statistic.7+.25, .55,
     paste0("F=", round(f.statistic.7, 1)))
```



Part (m): 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

Hide

```
p.value.7 <- pf(f.statistic.7,
                df1 = numerator.degrees.of.freedom,
                df2 = denominator.degrees.of.freedom)
cat("p value:", round(p.value.7,5))
```

```
p value: 0.9098
```

The p value is smaller than the significance level therefore I again reject the null hypothesis.

Part (n): 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

Hide

```
group.id.vector.7 <- c(
  rep("Group_1", times = group.sample.size),
  rep("Group_2", times = group.sample.size),
  rep("Group_3", times = group.sample.size)
)

anova.linear.model <- lm(aggregate.data.7 ~ group.id.vector.7)

anova(anova.linear.model)
```

```
Analysis of Variance Table

Response: aggregate.data.7
                  Df Sum Sq Mean Sq F value Pr(>F)
group.id.vector.7  2  219.9 109.966  2.4879 0.0902 .
Residuals         72 3182.4  44.201
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

My calculations match the results exactly for the degrees of freedom, test statistic and p-value.

End of problem 7

Problem 8: Simple Linear Regression

The vectors `problem.8.x.data` and `problem.8.y.data` contain a random sample of values from the simple linear regression model.

Part (a): x sample mean

Calculate the sample mean of the values in `problem.8.x.data`. Store your result in a variable, and report it using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
sample.mean.x.8 <- mean(problem.8.x.data)
cat("Sample mean of x:", round(sample.mean.x.8, 5))
```

```
Sample mean of x: 13
```

Part (b): y sample mean

Calculate the sample mean of the values in `problem.8.y.data`. Store your result in a variable, and report it using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
sample.mean.y.8 <- mean(problem.8.y.data)
cat("Sample mean of y:", round(sample.mean.y.8))
```

```
Sample mean of y: 30
```

Part (c): Calculating S_{xy}

Calculate S_{xy} for this data. Report your result using a `cat()` statement, rounding to 5 decimal places.

Solution

Hide

```
s_xy.8 <- sum(
           (problem.8.x.data - sample.mean.x.8) *
           (problem.8.y.data - sample.mean.y.8))
cat("S_xy:", round(s_xy.8, 5))
```

```
S_xy: -172.2081
```

Part (d): Calculating S_{xx}

Calculate S_{xx} for this data. Report your result using a `cat()` statement, rounding to 5 decimal places.

Hide

```
s_xx.8 <- sum(
            (problem.8.x.data - sample.mean.x.8) ^ 2)
cat("S_xx:", round(s_xx.8, 5))
```

```
S_xx: 140
```

Part (e): Estimating the slope coefficient

Calculate $\widehat{\beta}$, the estimate of the slope parameter. Report your result using a `cat()` statement.

Solution

Hide

```
beta.8 <- s_xy.8 / s_xx.8
cat("Slope parameter beta:", round(beta.8, 5))
```

```
Slope parameter beta: -1.23006
```

Part (f): Estimating the y-intercept coefficient

Calculate $\widehat{\alpha}$, the estimate of the y-intercept parameter. Report your result using a `cat()` statement.

Solution

Hide

```
alpha.8 <- sample.mean.y.8 - beta.8 * sample.mean.x.8
cat("y-intercept parameter alpha:", round(alpha.8, 5))
```

```
y-intercept parameter alpha: 46.31052
```

Part (g): Using the `lm()` function

Use the built-in R function `lm()` to estimate the linear model using `problem.8.x.data` and `problem.8.y.data`. Then report the y-intercept parameter estimate and the slope parameter estimate using a separate `cat()` statement for each, rounding to 5 decimal places. How do these values compare with your results in parts (e) and (f)?

Solution

Hide

```
model <- lm(problem.8.y.data ~ problem.8.x.data)
model
```

```

Call:
lm(formula = problem.8.y.data ~ problem.8.x.data)

Coefficients:
     (Intercept)  problem.8.x.data
           46.31             -1.23
```

Hide

```
cat("y-intercept:", round(model$coefficients[[1]], 5))
```

```
y-intercept: 46.31052
```

Hide

```
cat("\nslope:", round(model$coefficients[[2]], 5))
```

```

slope: -1.23006
```

The slope and intercept values I calculated are extremely close to the values determined by the lm function.

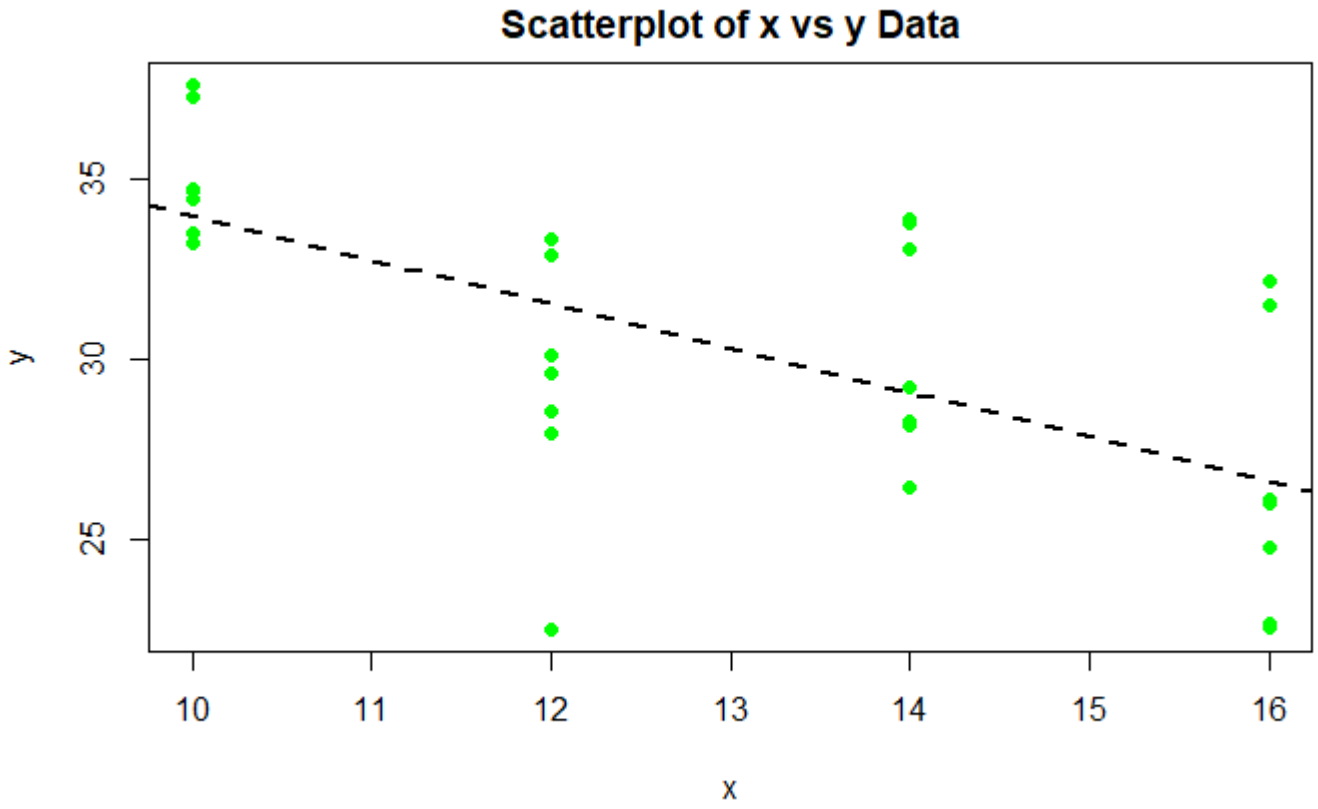
Part (h): Scatterplot

Construct a scatterplot for the values in `problem.8.x.data` and `problem.8.y.data`. Then draw the least-squares linear regression line.

Solution

Hide

```
plot(problem.8.y.data ~ problem.8.x.data,
     main = "Scatterplot of x vs y Data",
     xlab = "x",
     ylab = "y",
     pch = 19,
     cex = 1,
     col = "green")
abline(model,
       lwd = 2,
       lty = "dashed")
```


Scatterplot of x vs y Data
y
35
30
25
10
11
12
13
14
15
16
x