Take-Home Test 3 R Work

Alex Burgos

1.

```
algebra = read.delim("algebra.txt", header = T, sep = ",")
attach(algebra)
```

a.

```
control = algebra$control
n = length(control)
a = 0.05

stdDev = sd(control)

lowerChi = qchisq(a / 2, df = n - 1)
upperChi = qchisq(1 - (a / 2), df = n - 1)

lowerBound = sqrt((n - 1) * (stdDev^2 / upperChi))
upperBound = sqrt((n - 1) * (stdDev^2 / lowerChi))

cat("95% Confidence Interval for the control group: (",
    round(lowerBound, 4), ", ",
    round(upperBound, 4), ")\n")
```

```
95% Confidence Interval for the control group: ( 17.9991 ,  27.4643 )
```

b.

```r
exp = na.omit(algebra$experimental)
n = length(exp)
a = 0.05

stdDev = sd(exp)

lowerChi = qchisq(a / 2, df = n - 1)
upperChi = qchisq(1 - (a / 2), df = n - 1)

lowerBound = sqrt((n - 1) * (stdDev^2 / upperChi))
upperBound = sqrt((n - 1) * (stdDev^2 / lowerChi))

cat("95% Confidence Interval for the experimental group: (",
    round(lowerBound, 4), ", ",
    round(upperBound, 4), ")\n")
```

```
95% Confidence Interval for the experimental group: ( 12.2281 ,  21.7861 )
```

c.

```r
F = var.test(control, exp, alternative = "greater")
cat("F-statistic:", round(F$statistic, 4), "\n")
```

```
F-statistic: 1.9275
```

```r
cat("Degrees of freedom:", F$parameter[1], "and", F$parameter[2], "\n")
```

```
Degrees of freedom: 44 and 24
```

```r
cat("p-value:", round(F$p.value, 4), "\n")
```

```
p-value: 0.0436
```

2.

a.

```r
# Given data
x1 = 69
n1 = 21000
x2 = 113
n2 = 22000

proportion1 <- x1 / n1
proportion2 <- x2 / n2

stdError <- sqrt((proportion1 * (1 - proportion1)) / n1 + (proportion2 * (1 - proportion2)) /

z <- qnorm(1 - 0.01 / 2)

lowerBound <- (proportion1 - proportion2) - z * stdError
upperBound <- (proportion1 - proportion2) + z * stdError

cat("99% Confidence Interval for the difference in proportions: (",
    round(lowerBound, 4), ", ",
    round(upperBound, 4), ")\n")
```

```
99% Confidence Interval for the difference in proportions: ( -0.0035 ,  -2e-04 )
```

```r
#b
propsCombined = (x1 + x2) / (n1 + n2)

nullStdError = sqrt(propsCombined * (1 - propsCombined) * (1 / n1 + 1 / n2))

z = (proportion2 - proportion1) / nullStdError

p = 1 - pnorm(z)

cat("Test Statistic (z):", round(z, 4), "\n")
```

```
Test Statistic (z): 2.9548
```

```r
cat("p-value:", round(p, 4), "\n")
```

```
p-value: 0.0016
```

3.

```
batteries = read.delim("battery.txt", header = T, sep = ",")

Brand = batteries$Brand
Hours = batteries$Hours

# a
brandMean = tapply(Hours, Brand, mean)
brandStdDev = tapply(Hours, Brand, sd)
brandN = tapply(Hours, Brand, length)

brandMean
```

```
        A         B         C         D
 7.675000 10.425000  8.791667  5.900000
```

```
brandStdDev
```

```
       A        B        C        D
1.429112 2.198642 2.352062 1.149879
```

```
brandN
```

```
 A  B  C  D
10 10 12 10
```

```
# b
criticalValue = qf(0.95, 3, 38)
criticalValue
```

```
[1] 2.851741
```

```
anova = aov(Hours ~ Brand, data = batteries)
summary(anova)
```

```
            Df Sum Sq Mean Sq F value  Pr(>F)
Brand        3  109.3   36.44   10.29 4.3e-05 ***
Residuals   38  134.6    3.54
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.

```
data <- matrix(c(739, 70, 44, 853,
                 492, 130, 38, 660,
                 198, 143, 27, 368,
                 51,  84, 14, 149,
                 41, 114, 17, 172),
               nrow = 5, byrow = TRUE)

rownames(data) = c("Under 50", "50-179", "180-499", "500-999", "1000 and over")
colnames(data) = c("Full Owner", "Part Owner", "Tenant", "Total")

data = data[, -4]

chiSqr = chisq.test(data)

cat("Chi-Square Test Results:\n")
```

```
Chi-Square Test Results:
```

```
print(chiSqr)
```

```

    Pearson's Chi-squared test

data:  data
X-squared = 451.97, df = 8, p-value < 2.2e-16
```

```
criticalValue = qchisq(0.95, 8)
round(criticalValue, 4)
```

```
[1] 15.5073
```

5.

```r
observed = c(21, 28, 27, 40)
totalSeries = sum(observed)

probabilities = c(0.125, 0.25, 0.3125, 0.3125)

expected = probabilities * totalSeries

chiSqr = chisq.test(observed, p = probabilities, rescale.p = TRUE)
print(chiSqr)
```

```
    Chi-squared test for given probabilities

data:  observed
X-squared = 5.6966, df = 3, p-value = 0.1273
```

```r
criticalValue5 = qchisq(0.95, 3)
round(criticalValue5, 4)
```

```
[1] 7.8147
```

```r
criticalValue10 = qchisq(0.90, 3)
round(criticalValue10, 4)
```

```
[1] 6.2514
```