Homework 4

Brian Cervantes Alvarez
February 14, 2024
ST552: Statistical Methods

Question 1

Part A

From the model, we can see that "sex" and "income" are significant in the model, with a higher significance for "income". This may imply that depending on your income and what your sex is, they represent strong indicators for the response variable, which is gambling spending.

```
options(scipen = 3)
library(faraway)
data(teengamb)
fullModel <- lm(gamble ~ ., data = teengamb)
summary(fullModel)
```

```
Call:
lm(formula = gamble ~ ., data = teengamb)

Residuals:
    Min      1Q  Median      3Q     Max
-51.082 -11.320  -1.451   9.452  94.252

Coefficients:
             Estimate Std. Error t value  Pr(>|t|)
(Intercept)  22.55565   17.19680   1.312    0.1968
sex         -22.11833    8.21111  -2.694    0.0101 *
status        0.05223    0.28111   0.186    0.8535
income        4.96198    1.02539   4.839 0.0000179 ***
verbal       -2.95949    2.17215  -1.362    0.1803
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22.69 on 42 degrees of freedom
Multiple R-squared:  0.5267,    Adjusted R-squared:  0.4816
F-statistic: 11.69 on 4 and 42 DF,  p-value: 0.000001815
```

Question 1

Part B

The F-test comparing the two models yielded an F-statistic of 4.1338 with a p-value of 0.01177, suggesting that adding the other variables (e.g. "sex") to the model will increase the explanatory power. Hence, it is important to test the other variables and determine which one is the weakest of the bunch. From part a, we can suggest adding "sex" as the second predictor since it was statistically significant when compared "status" and "verbal."

```
# Models
fullModel <- lm(gamble ~ ., data = teengamb)
nullModel <- lm(gamble ~ income, data = teengamb)

# Doing it by hand, version 1
RSSfull <- sum(fullModel$residuals^2)
RSSnull <- sum(nullModel$residuals^2)
n <- as.numeric(nrow(teengamb))
pfull <- 5
pnull <- 2
Fstat <- ((RSSnull - RSSfull)/(pfull - pnull)) / (RSSfull/(n - pfull))
pval <- 1-pf(Fstat, df1 = pfull-pnull, df2 = n-pfull)
# Print the ds frame
ds <- data.frame(
  Description = c("RSS Full", "RSS Null", "Sum of Sq",
                  "n", "Full Parms",
                  "Null Parms", "DF",
                  "F", "[Pr(>F)]"),
  Value = c(RSSfull, RSSnull, RSSnull - RSSfull,
            n, pfull, pnull, pfull - pnull,
            Fstat, pval)
)
ds
```

```
  Description          Value
1    RSS Full 21623.76705490
2    RSS Null 28008.58759816
3   Sum of Sq  6384.82054326
4           n    47.00000000
5  Full Parms     5.00000000
6  Null Parms     2.00000000
7          DF     3.00000000
8           F     4.13376112
9    [Pr(>F)]     0.01177211
```

```r
# Doing it by hand, generalized version 2
K <- matrix(c(0, 1, 0, 0, 0,
              0, 0, 1, 0, 0,
              0, 0, 0, 0, 1), byrow = TRUE, nrow = 3)
betahat <- fullModel$coefficients
X <- model.matrix(fullModel)
m <- 3 # Number of hypotheses
sigmahat2 <- summary(fullModel)$sigma^2
n <- as.numeric(nrow(teengamb)) # Number of observations
Fstat <- (t(K %*% betahat) %*%
            solve(K %*% solve(t(X) %*% X) %*% t(K)) %*%
            (K %*% betahat) / m) / sigmahat2
pval <- 1 - pf(Fstat, df1 = m, df2 = n - length(betahat))
Fstat
```

```
         [,1]
[1,] 4.133761
```

```r
pval
```

```
           [,1]
[1,] 0.01177211
```

```r
# Anova Method to verify
anova(nullModel, fullModel)
```

```
Analysis of Variance Table

Model 1: gamble ~ income
Model 2: gamble ~ sex + status + income + verbal
  Res.Df   RSS Df Sum of Sq      F  Pr(>F)
1     45 28009
2     42 21624  3    6384.8 4.1338 0.01177 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 1

Part C

We can see that income, again, is a very strong predictor for the response variable. This can be some by the p-value, which is highly significant. Oh, the t-value and f-stat have a relationship where if you square the t-statistic then you get your F-statistic. For example,

$$t_{stat} = 5.330 \Rightarrow (5.330)^2 \approx 28.41 = F_{stat}$$

```
summary(nullModel)
```

```
Call:
lm(formula = gamble ~ income, data = teengamb)

Residuals:
    Min      1Q  Median      3Q     Max
-46.020 -11.874  -3.757  11.934 107.120

Coefficients:
            Estimate Std. Error t value   Pr(>|t|)
(Intercept)   -6.325      6.030  -1.049        0.3
income         5.520      1.036   5.330 0.00000305 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.95 on 45 degrees of freedom
Multiple R-squared:  0.387, Adjusted R-squared:  0.3734
F-statistic: 28.41 on 1 and 45 DF,  p-value: 0.000003045
```

Question 2

Part A

None of the individual predictors are statistically significant at a 5% level. However, if we notice the F-test's output, we can see that the model has a p-value of 0.019024, which is significant at the 5% level. This suggests that the combination of predictors significantly predicts the response variable better than a model without these predictors. In other words, there could be multicollinearity affecting the individual significance tests.

```r
library(faraway)
data(punting)

nullModel <- lm(Distance ~ RStr + LStr + RFlex + LFlex, data = punting)
summary(nullModel)
```

```
Call:
lm(formula = Distance ~ RStr + LStr + RFlex + LFlex, data = punting)

Residuals:
    Min      1Q  Median      3Q     Max
-23.941  -8.958  -4.441  13.523  17.016

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -79.6236    65.5935  -1.214    0.259
RStr          0.5116     0.4856   1.054    0.323
LStr         -0.1862     0.5130  -0.363    0.726
RFlex         2.3745     1.4374   1.652    0.137
LFlex        -0.5277     0.8255  -0.639    0.541

Residual standard error: 16.33 on 8 degrees of freedom
Multiple R-squared:  0.7365,    Adjusted R-squared:  0.6047
F-statistic:  5.59 on 4 and 8 DF,  p-value: 0.01902
```

Question 2

Part B

Based on the F-test results, we can conclude that, collectively, these four predictors (RStr, LStr, RFlex, LFlex) do not have a statistically significant relationship to the response variable based on this sample.

```
# Models
fullModel <- lm(Distance ~., data = punting)
nullModel <- lm(Distance ~ RStr + LStr + RFlex + LFlex, data = punting)

# Doing it by hand, version 1
RSSfull <- sum(fullModel$residuals^2)
RSSnull <- sum(nullModel$residuals^2)
n <- as.numeric(nrow(punting))
pfull <- 7
pnull <- 5
Fstat <- ((RSSnull - RSSfull)/(pfull - pnull)) / (RSSfull/(n - pfull))
pval <- 1-pf(Fstat, df1 = pfull-pnull, df2 = n-pfull)
# Print the ds frame
ds <- data.frame(
  Description = c("RSS Full", "RSS Null", "Sum of Sq",
                  "n", "Full Parms",
                  "Null Parms", "DF",
                  "F", "[Pr(>F)]"),
  Value = c(RSSfull, RSSnull, RSSnull - RSSfull,
            n, pfull, pnull, pfull - pnull,
            Fstat, pval)
)
ds
```

```
  Description        Value
1    RSS Full 1500.0212650
2    RSS Null 2132.6407146
3   Sum of Sq  632.6194496
4           n   13.0000000
5  Full Parms    7.0000000
6  Null Parms    5.0000000
7          DF    2.0000000
8           F    1.2652210
9    [Pr(>F)]    0.3479679
```

```r
# Doing it by hand, generalized version 2
K <- matrix(c(0, 1, 0, 0, 0, 0, 0,
              0, 0, 0, 0, 0, 0, 1), byrow = TRUE, nrow = 2)
betahat <- fullModel$coefficients
X <- model.matrix(fullModel)
m <- 2 # Number of hypotheses
sigmahat2 <- summary(fullModel)$sigma^2
n <- nrow(punting) # Number of observations (13)
Fstat <- (t(K %*% betahat) %*%
            solve(K %*% solve(t(X) %*% X) %*% t(K)) %*%
            (K %*% betahat) / m) / sigmahat2
pval <- 1 - pf(Fstat, df1 = m, df2 = n - length(betahat))
Fstat
```

```
         [,1]
[1,] 1.265221
```

```r
pval
```

```
          [,1]
[1,] 0.3479679
```

```r
# Anova Method to verify
anova(nullModel, fullModel)
```

```
Analysis of Variance Table

Model 1: Distance ~ RStr + LStr + RFlex + LFlex
Model 2: Distance ~ Hang + RStr + LStr + RFlex + LFlex + OStr
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1      8 2132.6
2      6 1500.0  2    632.62 1.2652  0.348
```

Question 2

Part C

After running the F-test again, we conclude that the interaction term between RStr and LStr serves a significant purpose in the model. It indicates that the effect of one predictor on the response variable is dependent on the level of the other predictor. This interaction term captures the combined effect of RStr and LStr on Distance that is not simply additive, suggesting a more complex relationship between the variables and the response.

```
# Future Reference, when you create interaction terms,
# they are placed at the end of the vector. In other words, RStr:LStr is
# would be referenced by the K-matrix as [c(0,0,0,0,0,1)]. If I added another
# intereaction term, then you would reference both with the K-matrix as
# K = [c(0,0,0,0,0,1,1), byrow = TRUE, nrow = 1]
fullModel <- lm(Distance ~ RStr + LStr +  RFlex + LFlex + RStr:LStr,
                data = punting)
nullModel <- lm(Distance ~ RStr + LStr + RFlex + LFlex, data = punting)

# ------------------------ Doing it by hand, version 1 ------------------------
RSSfull <- sum(fullModel$residuals^2)
RSSnull <- sum(nullModel$residuals^2)
n <- as.numeric(nrow(punting))
pfull <- 6
pnull <- 5
Fstat <- ((RSSnull - RSSfull)/(pfull - pnull)) / (RSSfull/(n - pfull))
pval <- 1-pf(Fstat, df1 = pfull-pnull, df2 = n-pfull)

ds <- data.frame(
  Description = c("RSS Full", "RSS Null", "Sum of Sq",
                  "n", "Full Parms",
                  "Null Parms", "DF",
                  "F", "[Pr(>F)]"),
  Value = c(RSSfull, RSSnull, RSSnull - RSSfull,
            n, pfull, pnull, pfull - pnull,
            Fstat, pval)
)
ds
```

```
  Description          Value
1    RSS Full  585.173576093
2    RSS Null 2132.640714627
3   Sum of Sq 1547.467138534
4           n   13.000000000
```

```
5  Full Parms    6.000000000
6  Null Parms    5.000000000
7          DF    1.000000000
8           F   18.511208319
9    [Pr(>F)]    0.003555823
```

```r
# ----------------- Doing it by hand, generalized version 2 -----------------
K <- matrix(c(0, 0, 0, 0, 0, 1), byrow = TRUE, nrow = 1)
betahat <- fullModel$coefficients
X <- model.matrix(fullModel)

m <- 1 # Number of hypotheses
sigmahat2 <- summary(fullModel)$sigma^2
n <- nrow(punting) # Number of observations (13)

Fstat <- (t(K %*% betahat) %*%
            solve(K %*% solve(t(X) %*% X) %*% t(K)) %*%
            (K %*% betahat) / m) / sigmahat2
pval <- 1 - pf(Fstat, df1 = m, df2 = n - length(betahat))



print(paste0("F-statistic: ", Fstat))
```

```
[1] "F-statistic: 18.5112083188562"
```

```r
print(paste0("P-value: ", pval))
```

```
[1] "P-value: 0.00355582327050996"
```

```r
# Anova Method for verification *,*
anova(nullModel, fullModel)
```

```
Analysis of Variance Table

Model 1: Distance ~ RStr + LStr + RFlex + LFlex
Model 2: Distance ~ RStr + LStr + RFlex + LFlex + RStr:LStr
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)
1      8 2132.64
2      7  585.17  1    1547.5 18.511 0.003556 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 2

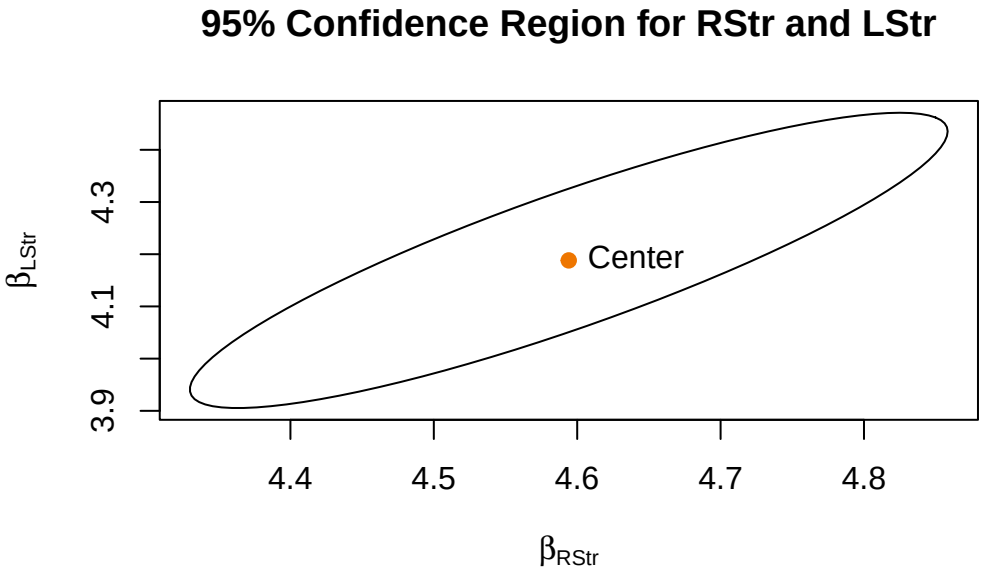
Part D

The F-test from part c showed us that the way $RStr$ and $LStr$ work together really matters for predicting the response. The confidence region we drew for their effects helps us see this more clearly. It doesn't just show us where the true effects of $RStr$ and $LStr$ might lie based on our ds, but it also shows how these two are linked together. This tells us that the interaction between $RStr$ and $LStr$ is a key part of understanding our model.

```
library(ellipse)

betahat <- coef(fullModel)
X <- model.matrix(fullModel)
sigmahat2 <- summary(fullModel)$sigma^2
K <- matrix(c(0, 1, 0, 0, 0, 0,
              0, 0, 1, 0, 0, 0), byrow = TRUE, nrow = 2)
fCrit <- qf(0.95, 2, nrow(punting) - length(betahat))
covM <- K %*% solve(t(X) %*% X) %*% t(K)
covM
```

```
           [,1]       [,2]
[1,] 0.01165468 0.01090222
[2,] 0.01090222 0.01335239
```

```
betaRStr <- betahat["RStr"]
betaLStr <- betahat["LStr"]
seRStr <- sqrt(covM[1, 1])
seLStr <- sqrt(covM[2, 2])
radiusRStr <- sqrt(fCrit) * seRStr
radiusLStr <- sqrt(fCrit) * seLStr
CREllipsiod <- ellipse(covM, centre = c(betaRStr, betaLStr), level = 0.95)

# Ellipse Confidence Region
plot(CREllipsiod, type = 'l', xlab = expression(beta[RStr]), ylab = expression(beta[LStr]),
     main = "95% Confidence Region for RStr and LStr")
points(betaRStr, betaLStr, pch = 19, col = 'darkorange2') # Mark the center
text(betaRStr, betaLStr, labels = "Center", pos = 4)
```


95% Confidence Region for RStr and LStr
Center
β_{LStr}
3.9
4.1
4.3
4.4
4.5
4.6
4.7
4.8
β_{RStr}

Question 2

Part E

```
ds <- punting
ds$TStr <- punting$LStr + punting$RStr

fullModel <- lm(Distance ~ LStr + RStr, data = ds)
nullModel <- lm(Distance ~ TStr, data = ds)
#summary(nullModel)

# ------------------------ Doing it by hand, version 1 ------------------------
RSSfull <- sum(fullModel$residuals^2)
RSSnull <- sum(nullModel$residuals^2)
n <- as.numeric(nrow(punting))
pfull <- 3
pnull <- 2
Fstat <- ((RSSnull - RSSfull)/(pfull - pnull)) / (RSSfull/(n - pfull))
pval <- 1-pf(Fstat, df1 = pfull-pnull, df2 = n-pfull)

ds <- data.frame(
  Description = c("RSS Full", "RSS Null", "Sum of Sq",
                  "n", "Full Parms",
                  "Null Parms", "DF",
                  "F", "[Pr(>F)]"),
  Value = c(RSSfull, RSSnull, RSSnull - RSSfull,
            n, pfull, pnull, pfull - pnull,
            Fstat, pval)
)
ds
```

```
  Description        Value
1    RSS Full 2973.0728200
2    RSS Null 3061.3540404
3   Sum of Sq   88.2812204
4           n   13.0000000
5  Full Parms    3.0000000
6  Null Parms    2.0000000
7          DF    1.0000000
8           F    0.2969360
9    [Pr(>F)]    0.5977516
```

```r
# Anova Method for verification *,*
anova(nullModel, fullModel)
```

```
Analysis of Variance Table

Model 1: Distance ~ TStr
Model 2: Distance ~ LStr + RStr
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     11 3061.3
2     10 2973.1  1    88.281 0.2969 0.5978
```

Question 2

Part F

```
fullModel <- lm(Distance ~ RStr + LStr + RFlex + LFlex, data = punting)
nullModel <- lm(Distance ~ RFlex + LFlex, data = punting)

# ------------------------ Doing it by hand, version 1 ------------------------
RSSfull <- sum(fullModel$residuals^2)
RSSnull <- sum(nullModel$residuals^2)
n <- as.numeric(nrow(punting))
pfull <- 5
pnull <- 3
Fstat <- ((RSSnull - RSSfull)/(pfull - pnull)) / (RSSfull/(n - pfull))
pval <- 1-pf(Fstat, df1 = pfull-pnull, df2 = n-pfull)

ds <- data.frame(
  Description = c("RSS Full", "RSS Null", "Sum of Sq",
                  "n", "Full Parms",
                  "Null Parms", "DF",
                  "F", "[Pr(>F)]"),
  Value = c(RSSfull, RSSnull, RSSnull - RSSfull,
            n, pfull, pnull, pfull - pnull,
            Fstat, pval)
)
ds
```

```
  Description        Value
1    RSS Full 2132.6407146
2    RSS Null 2492.1013123
3   Sum of Sq  359.4605976
4           n   13.0000000
5  Full Parms    5.0000000
6  Null Parms    3.0000000
7          DF    2.0000000
8           F    0.6742075
9    [Pr(>F)]    0.5363003
```

```
# Anova Method for verification *,*
anova(nullModel, fullModel)
```

```
Analysis of Variance Table
```

```
Model 1: Distance ~ RFlex + LFlex
Model 2: Distance ~ RStr + LStr + RFlex + LFlex
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     10 2492.1
2      8 2132.6  2    359.46 0.6742 0.5363
```

Question 2

Part G

```r
ds <- punting

ds$diffStr <- (ds$RStr - ds$LStr)
ds$diffFlex <- (ds$RFlex - ds$LFlex)

# Full model with differences to test symmetry
fullModelSymmetry <- lm(Distance ~ LStr +
                          RStr + LFlex + RFlex + diffStr + diffFlex, data = ds)
# Null model without differences
nullModelSymmetry <- lm(Distance ~ LStr + RStr + LFlex + RFlex, data = ds)
# Compare models
anova(nullModelSymmetry, fullModelSymmetry)
```

```
Analysis of Variance Table

Model 1: Distance ~ LStr + RStr + LFlex + RFlex
Model 2: Distance ~ LStr + RStr + LFlex + RFlex + diffStr + diffFlex
  Res.Df    RSS Df Sum of Sq F Pr(>F)
1      8 2132.6
2      8 2132.6  0         0
```

Question 2

Part H

Comparing models with different response variables using ANOVA is statistically invalid because ANOVA assumes the same response variable across models, making direct comparisons inappropriate. To evaluate predictor significance across models for Distance and Hang, we must assess each model independently, as they analyze distinct aspects of punting performance.

```r
hangModel <- lm(Hang ~ LStr + RStr + LFlex + RFlex, data = punting)
summary(hangModel)
```

```
Call:
lm(formula = Hang ~ LStr + RStr + LFlex + RFlex, data = punting)

Residuals:
     Min       1Q   Median       3Q      Max
-0.36297 -0.13528 -0.07849  0.09938  0.35893

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.225239   1.032784  -0.218    0.833
LStr         0.007697   0.008077   0.953    0.369
RStr         0.005153   0.007645   0.674    0.519
LFlex        0.004614   0.012998   0.355    0.732
RFlex        0.019404   0.022631   0.857    0.416

Residual standard error: 0.2571 on 8 degrees of freedom
Multiple R-squared:  0.8156,    Adjusted R-squared:  0.7235
F-statistic: 8.848 on 4 and 8 DF,  p-value: 0.004925
```

Question 3

Part A

Here are the models being used:

- Acetic:
  - Full model: $\text{taste}_i = \beta_0 + \beta_1 \text{Acetic}_i + \epsilon_i$
  - Reduced model: $\text{taste}_i = \beta_0 + \epsilon_i$
- H2S:
  - Full model: $\text{taste}_i = \beta_0 + \beta_1 \text{Acetic}_i + \beta_2 \text{H2S}_i + \epsilon_i$
  - Reduced model: $\text{taste}_i = \beta_0 + \epsilon_i$
- Lactic:
  - Full model: $\text{taste}_i = \beta_0 + \beta_1 \text{Acetic}_i + \beta_2 \text{H2S}_i + \beta_3 \text{Lactic}_i + \epsilon_i$
  - Reduced model: $\text{taste}_i = \beta_0 + \beta_1 \text{Acetic}_i + \beta_2 \text{H2S}_i + \epsilon_i$

```r
library(faraway)
data(cheddar)
fit <- lm(taste ~ ., data = cheddar)
anova(fit)
```

```
Analysis of Variance Table

Response: taste
          Df  Sum Sq Mean Sq F value     Pr(>F)
Acetic     1 2314.14 2314.14 22.5481 0.00006528 ***
H2S        1 2147.02 2147.02 20.9197  0.0001035 ***
Lactic     1  533.32  533.32  5.1964  0.0310795 *
Residuals 26 2668.41  102.63
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 3

Part B

```r
# Fit the full model
fullModel <- lm(taste ~ Acetic + H2S + Lactic, data = cheddar)

# Fit the reduced models
reducedModelLactic <- lm(taste ~ Acetic + H2S, data = cheddar)
reducedModelH2S <- lm(taste ~ Acetic, data = cheddar)
interceptOnlymodel <- lm(taste ~ 1, data = cheddar)

# Perform ANOVA calls
anova(interceptOnlymodel, reducedModelH2S) # Compares intercept only to Acetic
```

```
Analysis of Variance Table

Model 1: taste ~ 1
Model 2: taste ~ Acetic
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     29 7662.9
2     28 5348.7  1    2314.1 12.114 0.001658 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(reducedModelH2S, reducedModelLactic) # Compares Acetic to Acetic + H2S
```

```
Analysis of Variance Table

Model 1: taste ~ Acetic
Model 2: taste ~ Acetic + H2S
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     28 5348.7
2     27 3201.7  1      2147 18.106 0.0002247 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(reducedModelLactic, fullModel) # Compares Acetic + H2S to Acetic + H2S + Lactic
```

```
Analysis of Variance Table

Model 1: taste ~ Acetic + H2S
```

```
Model 2: taste ~ Acetic + H2S + Lactic
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     27 3201.7
2     26 2668.4  1    533.32 5.1964 0.03108 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 3

Part C

The F-values and p-values do not match between the original ANOVA output and the separate ANOVA calls because the F-statistic's denominator changes. In the original output, it's based on the residual variance from the full model. In the separate calls, it's based on the variance from the reduced models. That's the important distinction when viewing either version.

```
ssEffectAcetic = 2314.1
dfEffectAcetic = 1
rssModelAcetic = 5348.7
dfResidualAcetic = 28

# Calculate the F-statistic for the Acetic comparison using the correct values
fStatAcetic = (ssEffectAcetic / dfEffectAcetic) / (rssModelAcetic / dfResidualAcetic)
fStatAcetic
```

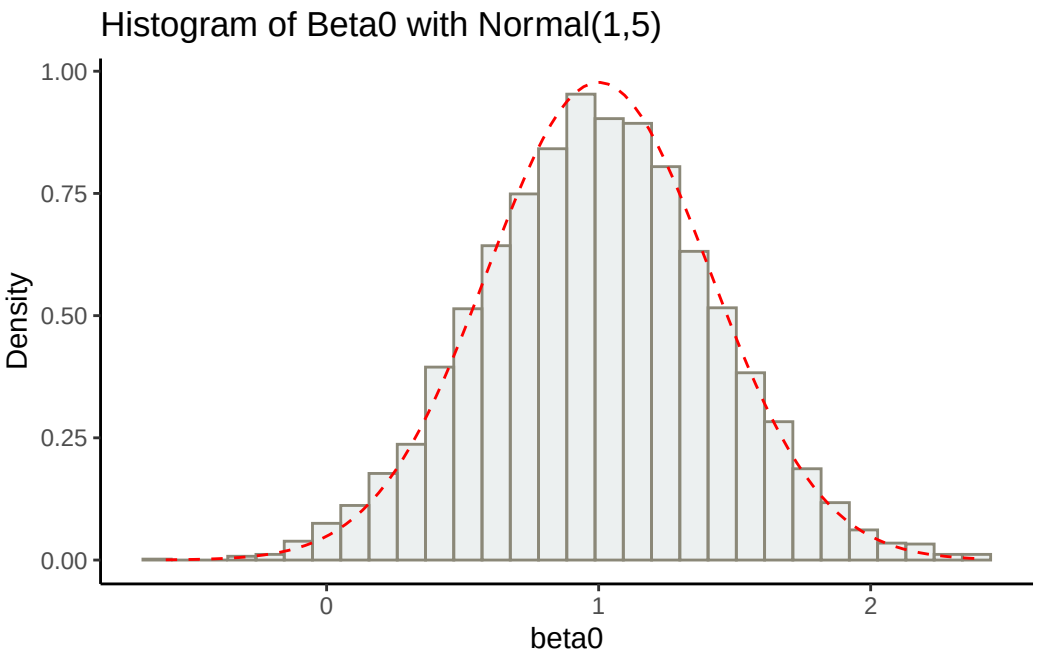
```
[1] 12.11412
```

Question 4

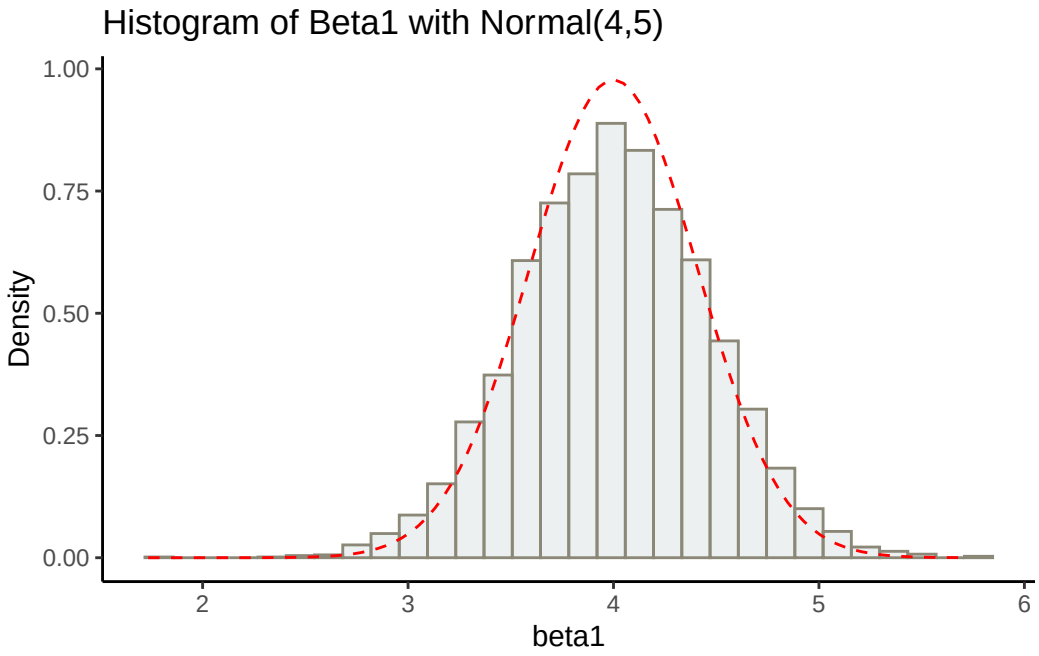
Part A

```r
library(ggplot2)
library(dplyr)
library(tidyr)

ds <- read.csv("HW4simulation.csv")

# Number of iterations
iterations <- 5000
n <- nrow(ds)

# Storage for estimates
betaEstimates <- matrix(NA, nrow = iterations, ncol = 4)
sigmaHat2 <- numeric(iterations)

# Simulation Loop
for(i in 1:iterations) {
  epsilon <- rnorm(n, mean = 0, sd = sqrt(5))
  Y <- 1 + 4 * ds$X1 - 3 * ds$X2 + epsilon
  model <- lm(Y ~ X1 + X2 + X3, data = ds)

  betaEstimates[i,] <- coef(model)
  sigmaHat2[i] <- sum(resid(model)^2) / (n - length(coef(model)))
}
# Data Wrangling
resultsDs <- data.frame(betaEstimates, sigmaHat2 = sigmaHat2)
colnames(resultsDs)[1:4] <- paste0("beta", 0:3)
longDs <- pivot_longer(resultsDs, cols = everything(), names_to = "estimate", values_to = "v


# Plot density plots for beta estimates with theoretical distribution curves overlaid
betaPlots <- function(data, betaType, expectedMean, variance, title) {
  ggplot(data %>% filter(estimate == betaType), aes(x = value)) +
    geom_histogram(aes(y = ..density..), color = "cornsilk4", fill = "azure3", alpha = 0.3,
    stat_function(fun = dnorm, args = list(mean = expectedMean, sd = sqrt(variance)),
                  color = "red", linetype = "dashed") +
    labs(title = title, x = betaType, y = "Density") +
    theme_classic()
}
```

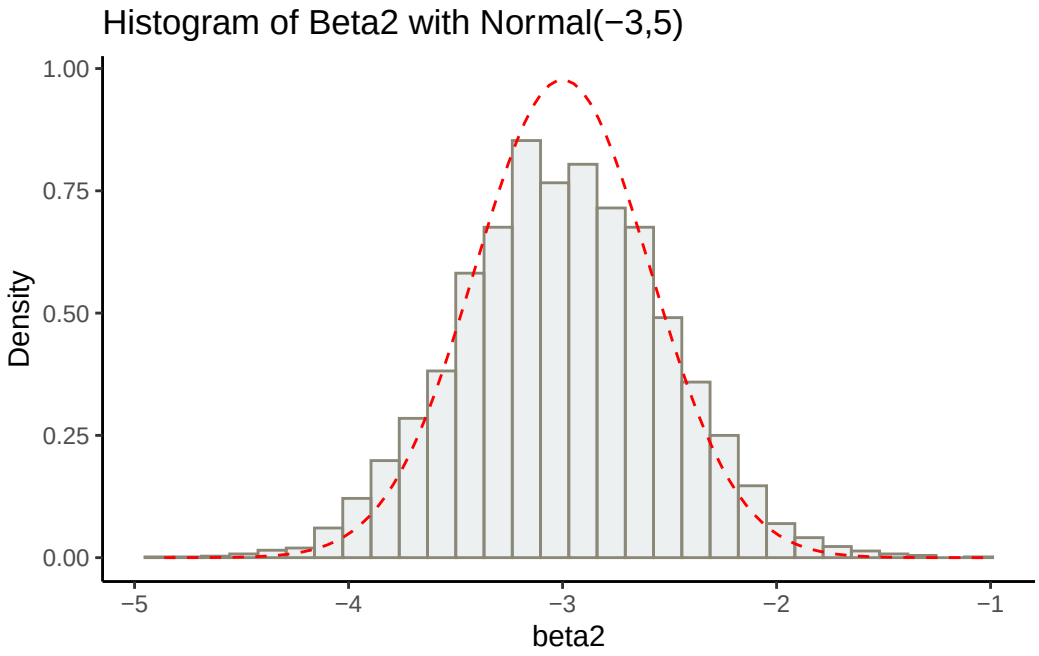
```r
# For sigma^2, assuming the theoretical distribution is a scaled chi-square distribution
# which can be approximated or transformed into a Gamma distribution
sigma2Plot <- function(data, sigmaHat, n, p, variance) {
  # Degrees of freedom for the chi-square distribution
  df <- n - p
  # Scale parameter for the chi-square distribution transformed to Gamma distribution
  scale <- 2 * variance / df

  ggplot(data %>% filter(estimate == sigmaHat), aes(x = value)) +
    geom_density(color = "limegreen", fill = "cyan4", alpha = 0.3) +
    stat_function(fun = dgamma,
                  args = list(shape = df / 2, scale = scale),
                  color = "red",
                  linetype = "dashed") +
    labs(title = "Density plot of sigma^2 with Gamma(13, 5/13)",
         x = "sigma^2", y = "Density") +
    theme_classic()
}

variance = 5/30

betaPlots(longDs, "beta0", 1, variance, "Histogram of Beta0 with Normal(1,5)")
```



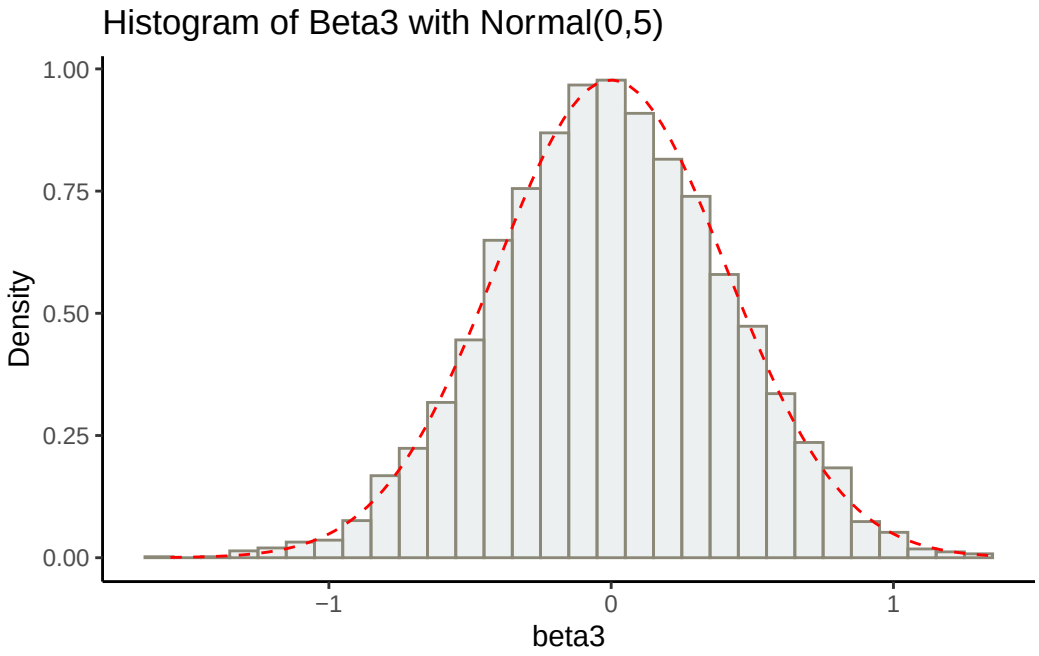

```r
betaPlots(longDs, "beta1", 4, variance, "Histogram of Beta1 with Normal(4,5)")
```



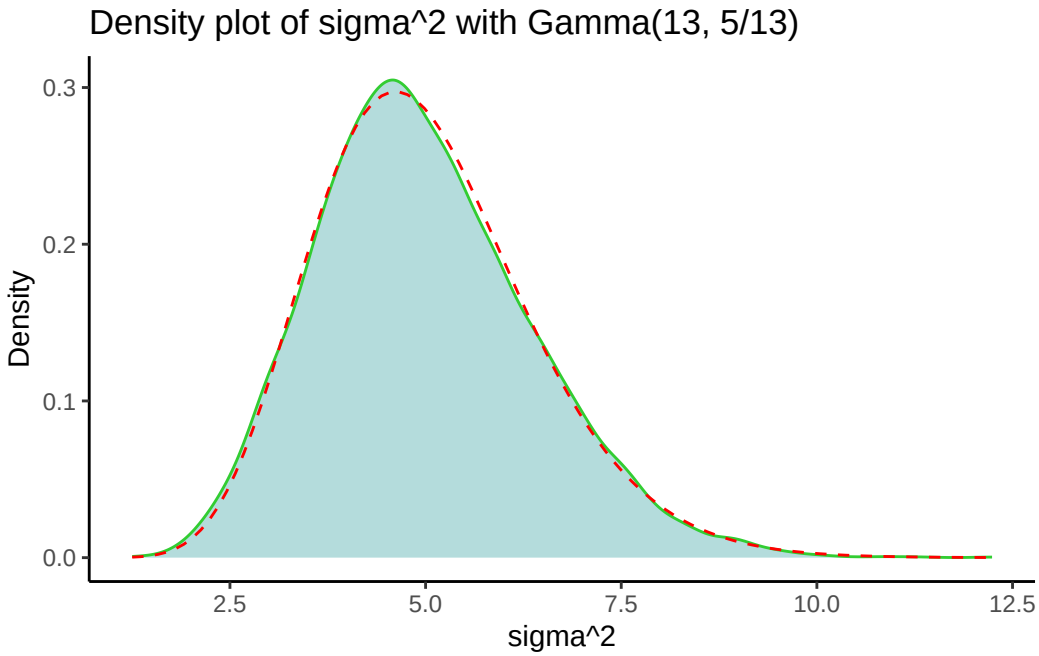

```
betaPlots(longDs, "beta2", -3, variance, "Histogram of Beta2 with Normal(-3,5)")
```




```
betaPlots(longDs, "beta3", 0, variance, "Histogram of Beta3 with Normal(0,5)")
```




```
sigma2Plot(longDs, "sigmaHat2", 30, 4, 5)
```



Theoretical Distributions:

- The coefficient estimates $\hat{\beta}_i$ are normally distributed, i.e., $\hat{\beta}_i \sim N(\beta_i, \sigma^2(X^TX)^{-1})$, where X is the design matrix including a column of 1s for the intercept.
- The estimator $\hat{\sigma}^2$ follows a scaled chi-squared distribution. Specifically, if $\epsilon \sim N(0, \sigma^2 I_n)$, then $\frac{(n-p)\hat{\sigma^2}}{\sigma^2} \sim \chi^2_{n-p}$, where n is the number of observations, p is the number of parameters (including the intercept), and $n-p$ are the degrees of freedom. By rearranging, we get that $\hat{\sigma}^2 \sim \frac{\sigma^2}{n-p}\chi^2_{n-p}$, which can be represented as $Gamma\left(\frac{n-p}{2}, \frac{2\sigma^2}{n-p}\right)$.
- $(\hat{\beta}_1 - \beta_1)/SE(\hat{\beta}_1)$ follows a standard normal distribution, $N(0,1)$, under the null hypothesis that the true coefficient β_1 is as specified.

Question 4

Part B

The histogram of $(\hat{\beta}_1 - \beta_1)/SE(\hat{\beta}_1)$ will be centered around 0 with a standard deviation of 1

```
beta1_estimates <- betaEstimates[, 2]

error_variance <- 5
SE_beta1 <- sqrt(error_variance / n)

# Standardize beta1 estimates
standardized_beta1 <- (beta1_estimates - 4) / SE_beta1  # Assuming beta1 true value is 4

# Plot histogram with theoretical distribution
ggplot(data.frame(standardized_beta1), aes(x = standardized_beta1)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "cornsilk4", color = "black", alpha
  stat_function(fun = dnorm, args = list(mean = 0, sd = 1), color = "red", size = 1, linetyp
  labs(title = "Standardized Beta1 Estimates",
       x = NULL,
       y = "Density") +
  theme_classic()
```