Mini Project 2

of

Stat 4360

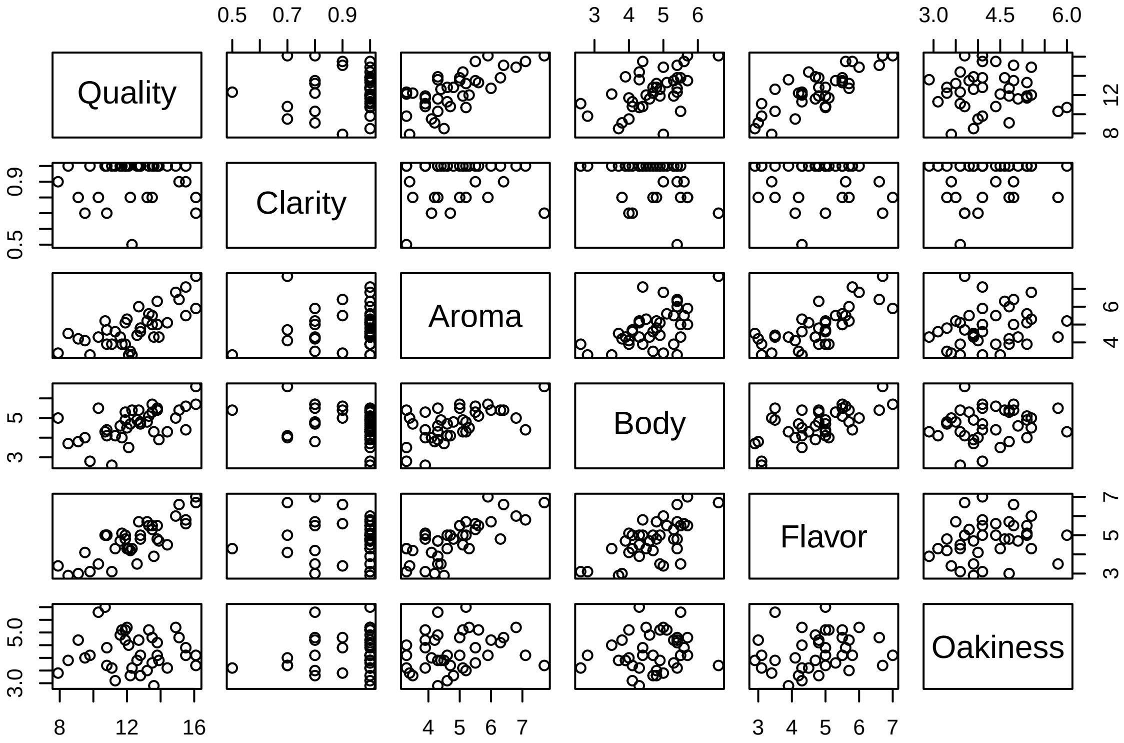
```
setwd("/Users/vannguyen/Downloads")
wine <- read.table("wine.txt", header = TRUE, sep = "\t")


## Question 1(a)
# Treat Region as a factor
wine$Region <- as.factor(wine$Region)

# Quick overview
str(wine)
```

```
## 'data.frame':    38 obs. of  7 variables:
##  $ Clarity : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Aroma   : num  3.3 4.4 3.9 3.9 5.6 4.6 4.8 5.3 4.3 4.3 ...
##  $ Body    : num  2.8 4.9 5.3 2.6 5.1 4.7 4.8 4.5 4.3 3.9 ...
##  $ Flavor  : num  3.1 3.5 4.8 3.1 5.5 5 4.8 4.3 3.9 4.7 ...
##  $ Oakiness: num  4.1 3.9 4.7 3.6 5.1 4.1 3.3 5.2 2.9 3.9 ...
##  $ Quality : num  9.8 12.6 11.9 11.1 13.3 12.8 12.8 12 13.6 13.9 ...
##  $ Region  : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 3 1 ...
```

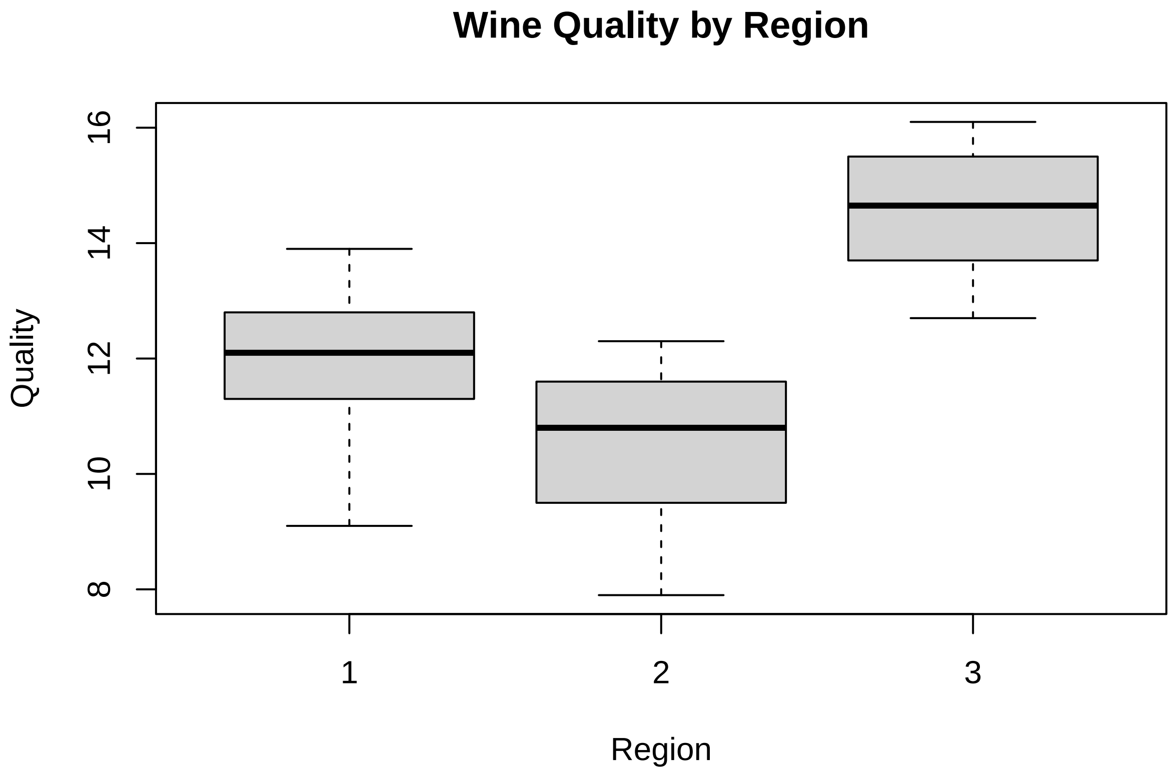
```
summary(wine)
```

```
##     Clarity           Aroma            Body           Flavor     
##  Min.   :0.5000   Min.   :3.300   Min.   :2.600   Min.   :2.900  
##  1st Qu.:0.8250   1st Qu.:4.125   1st Qu.:4.150   1st Qu.:4.225  
##  Median :1.0000   Median :4.650   Median :4.750   Median :4.800  
##  Mean   :0.9237   Mean   :4.847   Mean   :4.684   Mean   :4.768  
##  3rd Qu.:1.0000   3rd Qu.:5.450   3rd Qu.:5.375   3rd Qu.:5.500  
##  Max.   :1.0000   Max.   :7.700   Max.   :6.600   Max.   :7.000  
##     Oakiness        Quality      Region
##  Min.   :2.900   Min.   : 7.90   1:17  
##  1st Qu.:3.700   1st Qu.:11.15   2: 9  
##  Median :4.100   Median :12.45   3:12  
##  Mean   :4.255   Mean   :12.44         
##  3rd Qu.:4.775   3rd Qu.:13.75         
##  Max.   :6.000   Max.   :16.10         
```

```
# Pairwise scatterplots to see relationships
pairs(wine[, c("Quality","Clarity","Aroma","Body","Flavor","Oakiness")])
```



```
# Boxplot of Quality by Region
boxplot(Quality ~ Region, data = wine,
        main = "Wine Quality by Region",
        xlab = "Region", ylab = "Quality")
```



```
## Question 1(b)
# Simple linear regressions of Quality on each predictor

# Clarity
fit_clarity <- lm(Quality ~ Clarity, data = wine)
summary(fit_clarity)
```

```
##
## Call:
## lm(formula = Quality ~ Clarity, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5257 -1.3227  0.0947  1.2773  3.7681
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.0034     2.5610   4.687 3.89e-05 ***
## Clarity       0.4692     2.7486   0.171    0.865
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.073 on 36 degrees of freedom
## Multiple R-squared:  0.0008089,  Adjusted R-squared:  -0.02695
## F-statistic: 0.02914 on 1 and 36 DF,  p-value: 0.8654
```

```r
# Aroma
fit_aroma <- lm(Quality ~ Aroma, data = wine)
summary(fit_aroma)
```

```
##
## Call:
## lm(formula = Quality ~ Aroma, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4726 -0.8574 -0.0091  0.8346  2.2563
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.9583     1.1050   5.392 4.51e-06 ***
## Aroma         1.3365     0.2226   6.004 6.87e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.466 on 36 degrees of freedom
## Multiple R-squared:  0.5003, Adjusted R-squared:  0.4864
## F-statistic: 36.04 on 1 and 36 DF,  p-value: 6.871e-07
```

```r
# Body
fit_body <- lm(Quality ~ Body, data = wine)
summary(fit_body)
```

```
##
## Call:
## lm(formula = Quality ~ Body, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9669 -0.8386  0.0620  1.2204  3.4502
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.0580     1.6441   3.685 0.000748 ***
## Body          1.3618     0.3458   3.938 0.000361 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.734 on 36 degrees of freedom
## Multiple R-squared:  0.3011, Adjusted R-squared:  0.2817
## F-statistic: 15.51 on 1 and 36 DF,  p-value: 0.0003612
```

```r
# Flavor
fit_flavor <- lm(Quality ~ Flavor, data = wine)
summary(fit_flavor)
```

```
##
## Call:
```

```
## lm(formula = Quality ~ Flavor, data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.38583 -0.72226 -0.00756  0.62006  2.52822
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.9414     0.9911   4.986 1.57e-05 ***
## Flavor        1.5719     0.2033   7.732 3.68e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.271 on 36 degrees of freedom
## Multiple R-squared:  0.6242, Adjusted R-squared:  0.6137
## F-statistic: 59.79 on 1 and 36 DF,  p-value: 3.683e-09
```

```r
# Oakiness
fit_oakiness <- lm(Quality ~ Oakiness, data = wine)
summary(fit_oakiness)
```

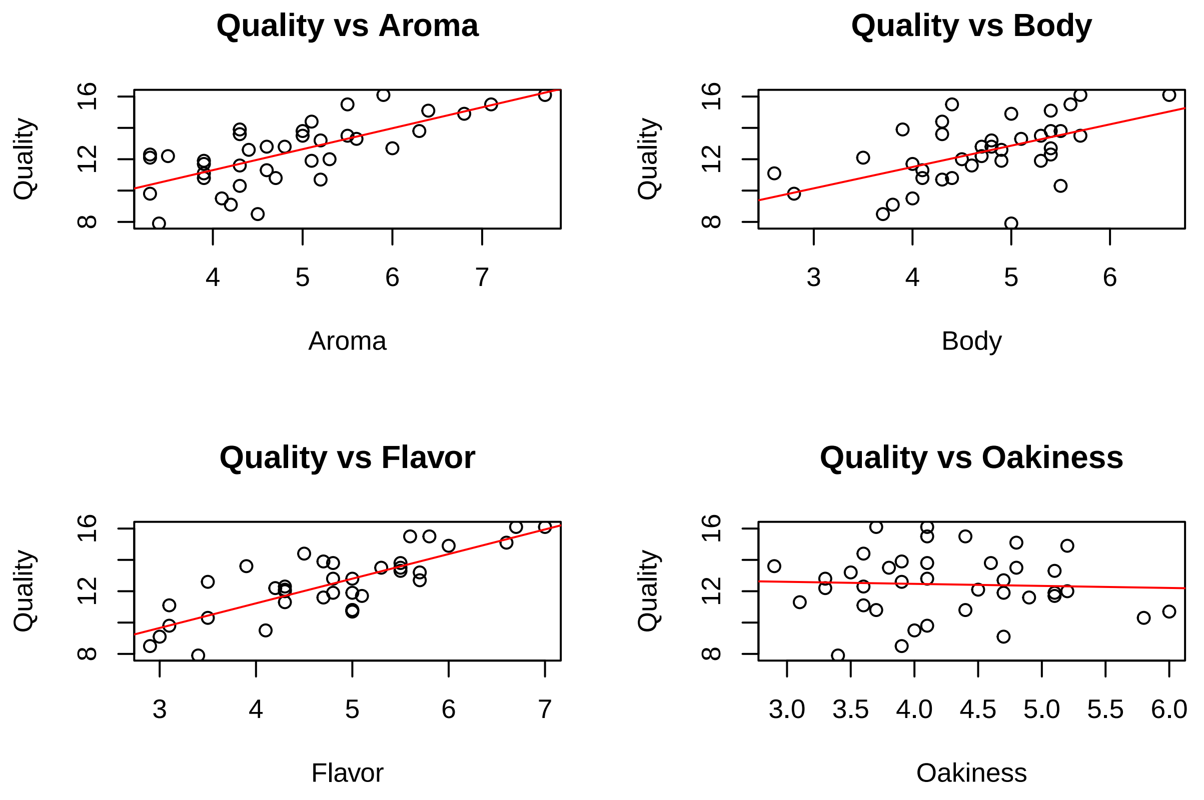
```
##
## Call:
## lm(formula = Quality ~ Oakiness, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6483 -1.3886 -0.0527  1.2907  3.6429
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.9916     1.9918   6.522  1.4e-07 ***
## Oakiness     -0.1304     0.4614  -0.283    0.779
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.071 on 36 degrees of freedom
## Multiple R-squared:  0.002213,   Adjusted R-squared:  -0.0255
## F-statistic: 0.07984 on 1 and 36 DF,  p-value: 0.7791
```

```r
# Region (qualitative predictor)
fit_region <- lm(Quality ~ Region, data = wine)
summary(fit_region)
```

```
##
## Call:
## lm(formula = Quality ~ Region, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8765 -0.8532  0.2395  0.9167  1.9235
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.9765     0.3180  37.662  < 2e-16 ***
## Region2      -1.5320     0.5405  -2.834  0.00757 **
## Region3       2.6069     0.4944   5.273 7.01e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.311 on 35 degrees of freedom
## Multiple R-squared:  0.6113, Adjusted R-squared:  0.5891
## F-statistic: 27.52 on 2 and 35 DF,  p-value: 6.587e-08
```

```r
# Scatterplots with regression lines
par(mfrow=c(2,2))
plot(Quality ~ Aroma, data=wine, main="Quality vs Aroma")
abline(lm(Quality ~ Aroma, data=wine), col="red")

plot(Quality ~ Body, data=wine, main="Quality vs Body")
abline(lm(Quality ~ Body, data=wine), col="red")

plot(Quality ~ Flavor, data=wine, main="Quality vs Flavor")
abline(lm(Quality ~ Flavor, data=wine), col="red")

plot(Quality ~ Oakiness, data=wine, main="Quality vs Oakiness")
abline(lm(Quality ~ Oakiness, data=wine), col="red")
```



```r
## Question 1(c)
# Multiple regression with all predictors
fit_all <- lm(Quality ~ Clarity + Aroma + Body + Flavor + Oakiness + Region, data = wine)
summary(fit_all)
```

```
##
## Call:
## lm(formula = Quality ~ Clarity + Aroma + Body + Flavor + Oakiness +
##     Region, data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.80824 -0.58413 -0.02081  0.48627  1.70909
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.81437    1.96944   3.968 0.000417 ***
## Clarity      0.01705    1.45627   0.012 0.990736
## Aroma        0.08901    0.25250   0.353 0.726908
## Body         0.07967    0.26772   0.298 0.768062
## Flavor       1.11723    0.24026   4.650 6.25e-05 ***
## Oakiness    -0.34644    0.23301  -1.487 0.147503
## Region2     -1.51285    0.39227  -3.857 0.000565 ***
## Region3      0.97259    0.51017   1.906 0.066218 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.9154 on 30 degrees of freedom
## Multiple R-squared:  0.8376, Adjusted R-squared:  0.7997
## F-statistic:  22.1 on 7 and 30 DF,  p-value: 3.295e-10
```

```r
## Question 1(d)
# Reduced model
fit_reduced <- lm(Quality ~ Flavor + Region, data = wine)
summary(fit_reduced)
```

```
##
## Call:
## lm(formula = Quality ~ Flavor + Region, data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.97630 -0.58844  0.02184  0.51572  1.94232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.0943     0.7912   8.967 1.76e-10 ***
## Flavor        1.1155     0.1738   6.417 2.49e-07 ***
## Region2      -1.5335     0.3688  -4.158 0.000205 ***
## Region3       1.2234     0.4003   3.056 0.004346 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Residual standard error: 0.8946 on 34 degrees of freedom
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8087
## F-statistic: 53.13 on 3 and 34 DF,  p-value: 6.358e-13
```

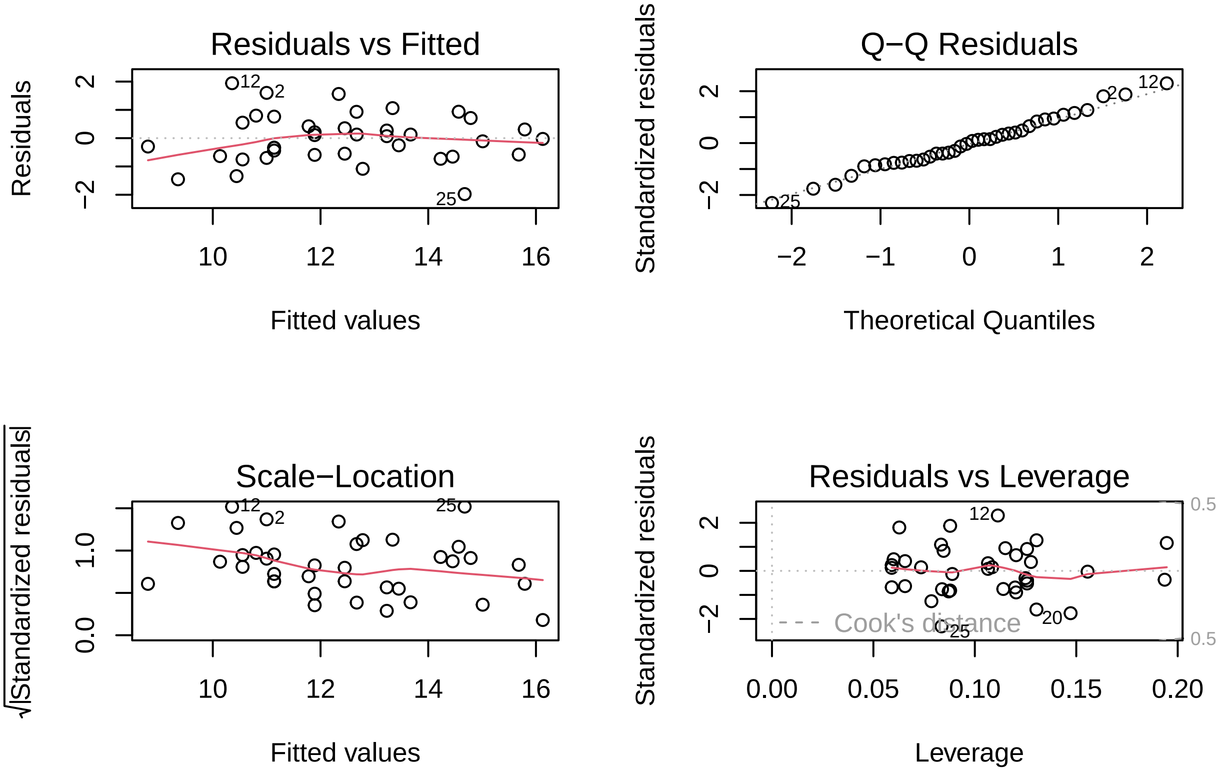
```
# Interaction check
fit_interaction <- lm(Quality ~ Flavor * Region, data = wine)
summary(fit_interaction)
```

```
##
## Call:
## lm(formula = Quality ~ Flavor * Region, data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.94964 -0.58463  0.04393  0.49607  1.97295
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.7311     1.1292   5.961 1.22e-06 ***
## Flavor          1.1985     0.2532   4.733 4.31e-05 ***
## Region2        -2.8942     2.1183  -1.366    0.181
## Region3         3.3833     2.0153   1.679    0.103
## Flavor:Region2  0.3108     0.4766   0.652    0.519
## Flavor:Region3 -0.4029     0.3878  -1.039    0.307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8914 on 32 degrees of freedom
## Multiple R-squared:  0.8357, Adjusted R-squared:  0.8101
## F-statistic: 32.56 on 5 and 32 DF,  p-value: 1.179e-11
```

```
anova(fit_reduced, fit_interaction)
```

```
## Analysis of Variance Table
##
## Model 1: Quality ~ Flavor + Region
## Model 2: Quality ~ Flavor * Region
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     34 27.213
## 2     32 25.429  2    1.7845 1.1229 0.3378
```

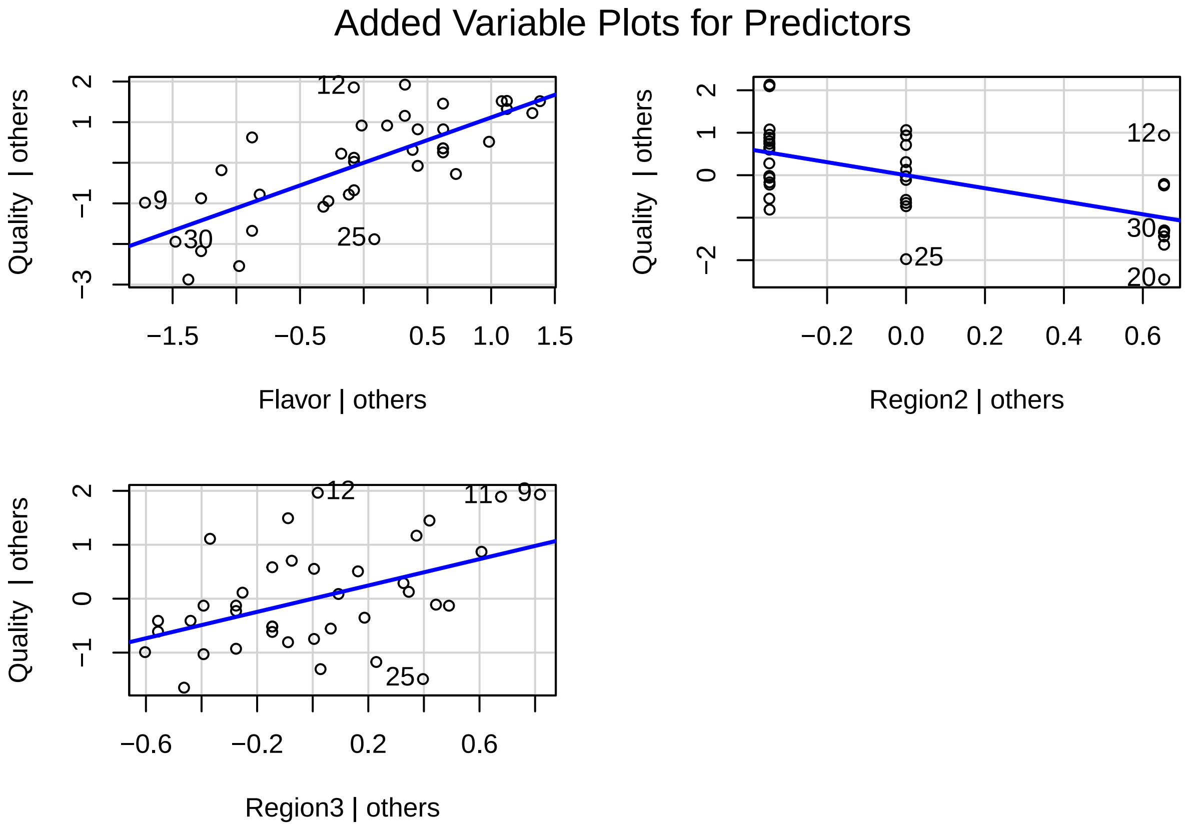
```
# Residual diagnostics
par(mfrow=c(2,2))
plot(fit_reduced)
```



```r
# Added-variable plots
library(car)
```

```
## Loading required package: carData
```

```r
avPlots(fit_reduced, main="Added Variable Plots for Predictors")
```



```r
## Question 1(f)

# Use the reduced model from part (d)
fit_reduced <- lm(Quality ~ Flavor + Region, data = wine)

# Mean Flavor (from dataset)
mean_flavor <- mean(wine$Flavor)

# Create new data for Region 1 with Flavor = mean
new_obs <- data.frame(Flavor = mean_flavor, Region = factor("1", levels = c("1","2","3")))

# Prediction and confidence intervals
predict(fit_reduced, newdata = new_obs,
        interval = "confidence", level = 0.95)
```

```
##        fit      lwr     upr
## 1 12.41371 11.95152 12.8759
```

```r
predict(fit_reduced, newdata = new_obs,
        interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 12.41371 10.53775 14.28967
```

```
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(MASS)
library(pROC)
```

```
## Type ’citation("pROC")’ for a citation.
```

```
##
## Attaching package: ’pROC’
```

```
## The following objects are masked from ’package:stats’:
##
##     cov, smooth, var
```

```
setwd("/Users/vannguyen/Downloads")
diabetes <- read.csv("diabetes.csv")

## Question 2(a)
# Quick structure and summary
str(diabetes)
```

```
## ’data.frame’:    2000 obs. of  9 variables:
##  $ Pregnancies             : int  2 0 0 0 1 0 4 8 2 2 ...
##  $ Glucose                 : int  138 84 145 135 139 173 99 194 83 89 ...
##  $ BloodPressure           : int  62 82 0 68 62 78 72 80 65 90 ...
##  $ SkinThickness           : int  35 31 0 42 41 32 17 0 28 30 ...
##  $ Insulin                 : int  0 125 0 250 480 265 0 0 66 0 ...
##  $ BMI                     : num  33.6 38.2 44.2 42.3 40.7 46.5 25.6 26.1 36.8 33.5 ...
##  $ DiabetesPedigreeFunction: num  0.127 0.233 0.63 0.365 0.536 ...
##  $ Age                     : int  47 23 31 24 21 58 28 67 24 42 ...
##  $ Outcome                 : int  1 0 1 1 0 0 0 0 0 0 ...
```

```
summary(diabetes)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 63.50   1st Qu.:  0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median : 23.00
##  Mean   : 3.704   Mean   :121.2   Mean   : 69.15   Mean   : 20.93
##  3rd Qu.: 6.000   3rd Qu.:141.0   3rd Qu.: 80.00   3rd Qu.: 32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :110.00
##     Insulin            BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.00   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.00   1st Qu.:27.38   1st Qu.:0.2440           1st Qu.:24.00
##  Median : 40.00   Median :32.30   Median :0.3760           Median :29.00
##  Mean   : 80.25   Mean   :32.19   Mean   :0.4709           Mean   :33.09
##  3rd Qu.:130.00   3rd Qu.:36.80   3rd Qu.:0.6240           3rd Qu.:40.00
##  Max.   :744.00   Max.   :80.60   Max.   :2.4200           Max.   :81.00
##     Outcome
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.342
##  3rd Qu.:1.000
##  Max.   :1.000
```

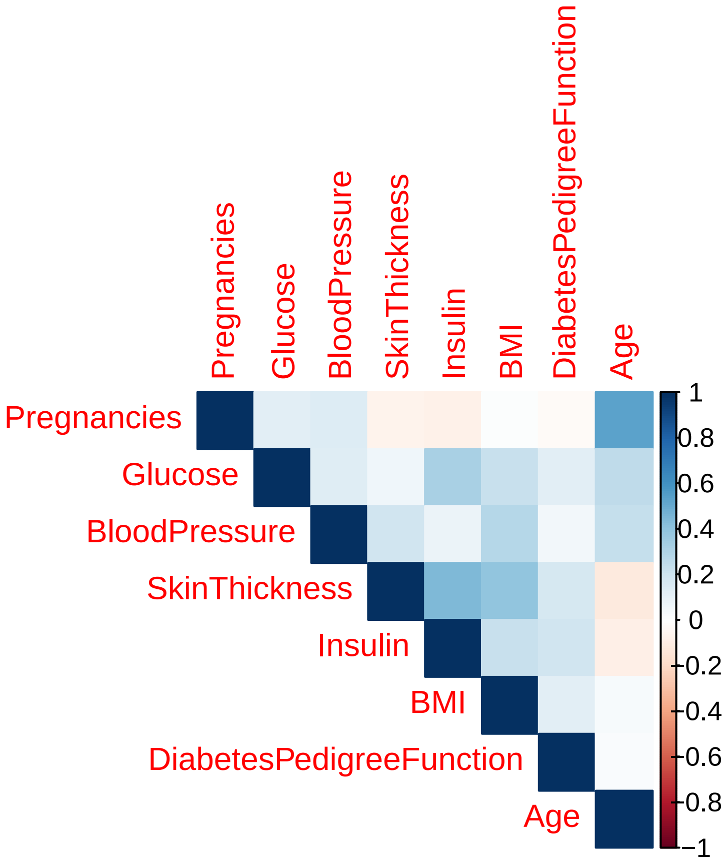
```
# Check distribution of the response
table(diabetes$Outcome)
```

```
##
##    0    1
## 1316  684
```

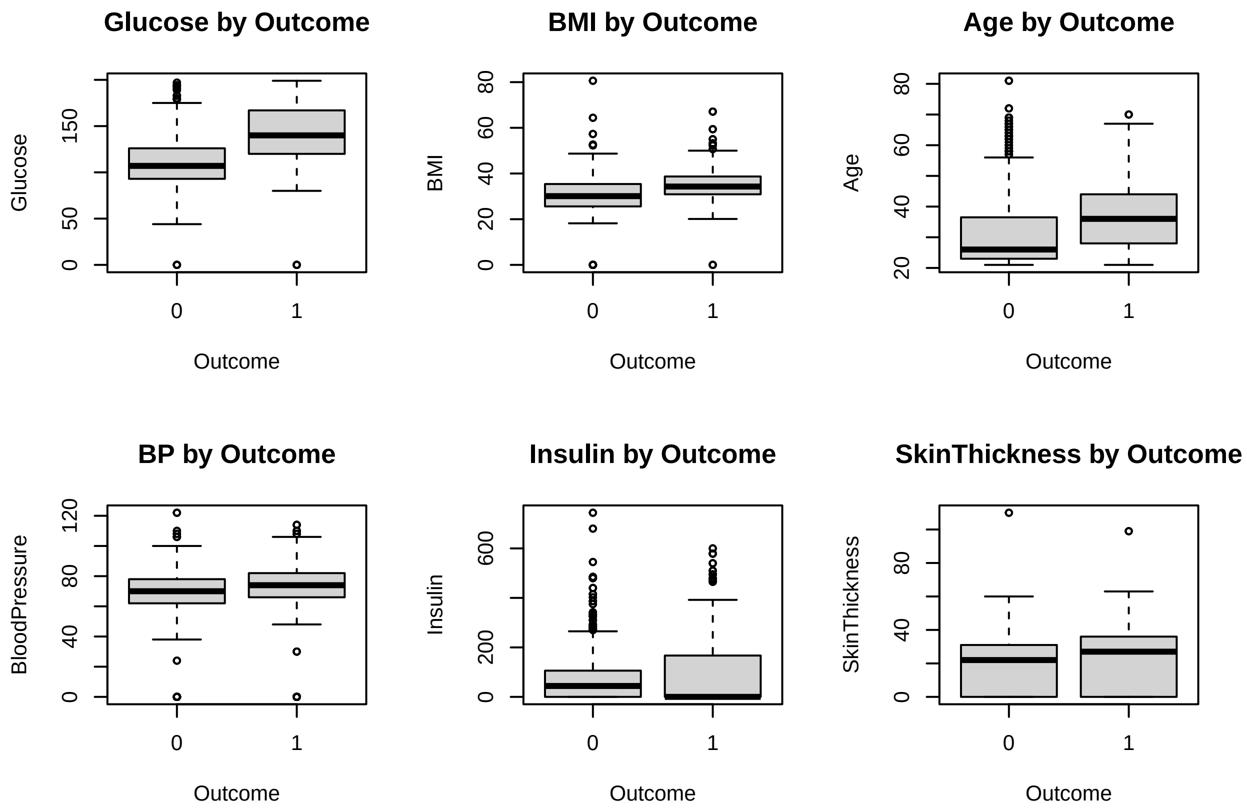
```
# Means by Outcome (to see group differences)
aggregate(. ~ Outcome, data = diabetes, mean)
```

```
##   Outcome Pregnancies  Glucose BloodPressure SkinThickness  Insulin      BMI
## 1       0    3.168693 110.5866      68.09498      20.05243 70.56383 30.56748
## 2       1    4.732456 141.5687      71.16667      22.63304 98.89766 35.32047
##   DiabetesPedigreeFunction      Age
## 1                0.4346763 31.08131
## 2                0.5406813 36.95614
```

```
# Correlation matrix of numeric predictors
corrplot(cor(diabetes[,-9]), method="color", type="upper")
```



```r
# Boxplots of key predictors by Outcome
par(mfrow=c(2,3))
boxplot(Glucose ~ Outcome, data=diabetes, main="Glucose by Outcome")
boxplot(BMI ~ Outcome, data=diabetes, main="BMI by Outcome")
boxplot(Age ~ Outcome, data=diabetes, main="Age by Outcome")
boxplot(BloodPressure ~ Outcome, data=diabetes, main="BP by Outcome")
boxplot(Insulin ~ Outcome, data=diabetes, main="Insulin by Outcome")
boxplot(SkinThickness ~ Outcome, data=diabetes, main="SkinThickness by Outcome")
```



```r
## Question 2(b)
# LDA model
fit_lda <- lda(Outcome ~ ., data=diabetes)

# Predict with LDA
pred_lda <- predict(fit_lda)

# Classify using 0.5 cutoff
lda_class <- ifelse(pred_lda$posterior[,2] > 0.5, 1, 0)

# Confusion matrix
table(Predicted = lda_class, Actual = diabetes$Outcome)
```

```
##          Actual
## Predicted    0    1
##         0 1174  298
##         1  142  386
```

```r
# Misclassification rate
mean(lda_class != diabetes$Outcome)
```

```
## [1] 0.22
```

```
# Sensitivity and Specificity
sensitivity <- sum(lda_class==1 & diabetes$Outcome==1) / sum(diabetes$Outcome==1)
specificity <- sum(lda_class==0 & diabetes$Outcome==0) / sum(diabetes$Outcome==0)
sensitivity; specificity
```

```
## [1] 0.5643275
```

```
## [1] 0.8920973
```

```
# ROC for LDA
roc_obj <- roc(diabetes$Outcome, pred_lda$posterior[,2], direction="<")
```

```
## Setting levels: control = 0, case = 1
```

```
plot(roc_obj, legacy.axes=TRUE, col="blue", lwd=2,
     main="ROC Curve - LDA")
auc(roc_obj)
```

```
## Area under the curve: 0.8369
```

```
## Question 2(c)
# QDA model
fit_qda <- qda(Outcome ~ ., data=diabetes)

# Predict with QDA
pred_qda <- predict(fit_qda)

# Classify with 0.5 cutoff
qda_class <- ifelse(pred_qda$posterior[,2] > 0.5, 1, 0)

# Confusion matrix
table(Predicted = qda_class, Actual = diabetes$Outcome)
```

```
##          Actual
## Predicted    0    1
##         0 1135  290
##         1  181  394
```

```
# Misclassification rate
mean(qda_class != diabetes$Outcome)
```

```
## [1] 0.2355
```

```
# Sensitivity and Specificity
sensitivity_qda <- sum(qda_class==1 & diabetes$Outcome==1) / sum(diabetes$Outcome==1)
specificity_qda <- sum(qda_class==0 & diabetes$Outcome==0) / sum(diabetes$Outcome==0)
sensitivity_qda; specificity_qda
```

```
## [1] 0.5760234
```

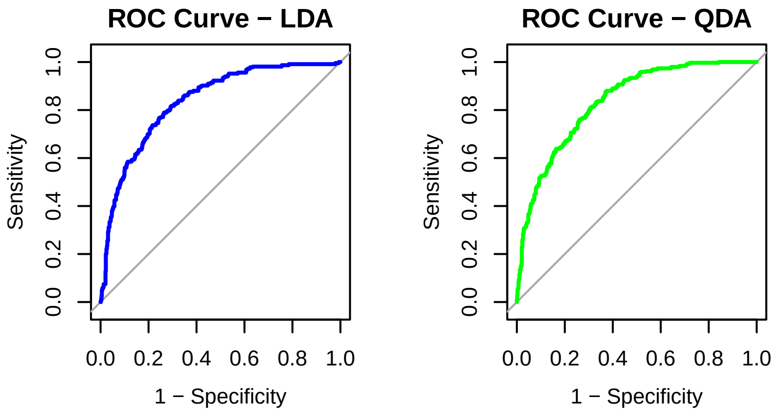
```
## [1] 0.862462
```

```
# ROC curve for QDA
roc_qda <- roc(diabetes$Outcome, pred_qda$posterior[,2], direction="<")
```

```
## Setting levels: control = 0, case = 1
```

```
plot(roc_qda, legacy.axes=TRUE, col="green", lwd=2,
     main="ROC Curve - QDA")
auc(roc_qda)
```

```
## Area under the curve: 0.8354
```



```
## Bonus Question (a)
p <- 10
sigma <- 1
mu <- rep(1, p)
N <- 1000

# Storage
mu_mle <- matrix(0, nrow=N, ncol=p)
mu_js  <- matrix(0, nrow=N, ncol=p)

# Simulation
for(i in 1:N){
  Y <- MASS::mvrnorm(1, mu, sigma^2 * diag(p))
  # MLE
  mu_mle[i,] <- Y
  # James-Stein shrinkage
  shrink <- 1 - ( (p-2)*sigma^2 ) / sum(Y^2)
  mu_js[i,] <- shrink * Y
}

# Compute bias and risk
bias_mle <- norm(colMeans(mu_mle) - mu, type="2")
bias_js  <- norm(colMeans(mu_js)  - mu, type="2")

risk_mle <- mean(rowSums((mu_mle - matrix(mu, nrow=N, ncol=p, byrow=TRUE))^2))
risk_js  <- mean(rowSums((mu_js  - matrix(mu, nrow=N, ncol=p, byrow=TRUE))^2))

bias_mle; bias_js
```

```
## [1] 0.09452887
```

```
## [1] 1.283602
```

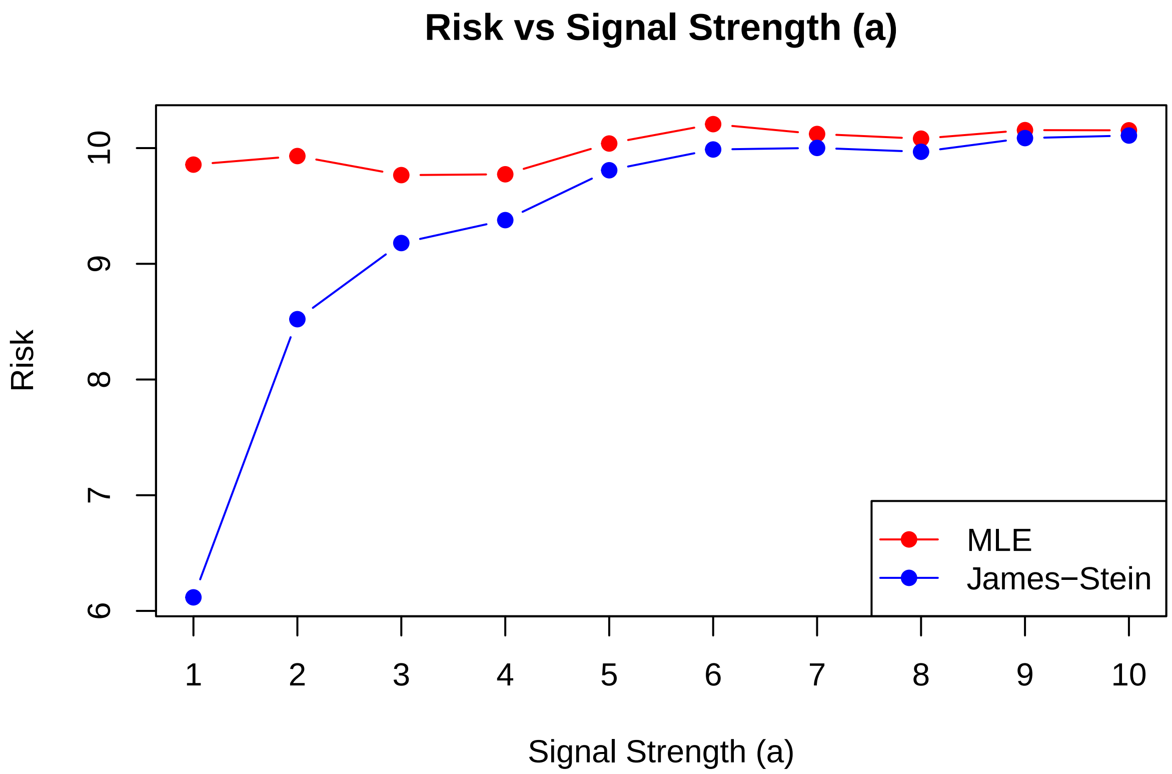
```
risk_mle; risk_js
```

```
## [1] 9.843847
```

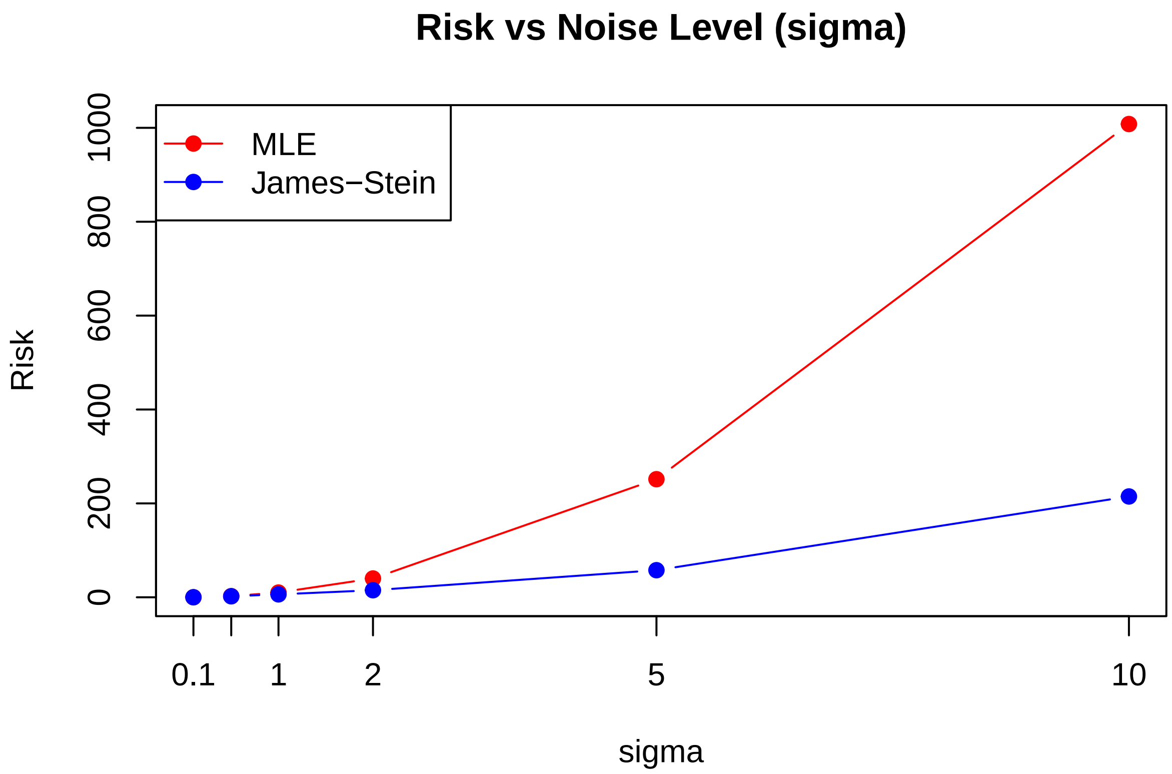
```
## [1] 6.150852
```

```
## Bonus Question (b)
# Risk vs Signal Strength (a)
a.values <- 1:10
risk_mle_a <- numeric(length(a.values))
risk_js_a  <- numeric(length(a.values))

for(k in 1:length(a.values)){
  a <- a.values[k]
  mu <- rep(a, p)    # mean vector changes with a

  mu_mle <- matrix(0, nrow=N, ncol=p)
  mu_js  <- matrix(0, nrow=N, ncol=p)

  for(i in 1:N){
    Y <- MASS::mvrnorm(1, mu, sigma^2 * diag(p))
    mu_mle[i,] <- Y
    shrink <- 1 - ((p-2)*sigma^2) / sum(Y^2)
    mu_js[i,] <- shrink * Y
  }

  # record risk
  risk_mle_a[k] <- mean(rowSums((mu_mle - mu)^2))
  risk_js_a[k]  <- mean(rowSums((mu_js  - mu)^2))
}
```

```r
# Plot
plot(a.values, risk_mle_a, type="b", col="red", pch=19,
     ylim=range(c(risk_mle_a, risk_js_a)),
     xlab="Signal Strength (a)", ylab="Risk",
     main="Risk vs Signal Strength (a)", xaxt="n")
lines(a.values, risk_js_a, type="b", col="blue", pch=19)
legend("bottomright", legend=c("MLE", "James-Stein"),
       col=c("red","blue"), pch=19, lty=1)
axis(1, at=a.values, labels=a.values)
```



```r
## Bonus Question (c)
# Risk vs Noise Level (sigma)
sigma.values <- c(0.1, 0.5, 1, 2, 5, 10)
risk_mle_s <- numeric(length(sigma.values))
risk_js_s  <- numeric(length(sigma.values))
mu <- rep(1, p)    # reset mean vector

for(k in 1:length(sigma.values)){
  sigma <- sigma.values[k]

  mu_mle <- matrix(0, nrow=N, ncol=p)
  mu_js  <- matrix(0, nrow=N, ncol=p)

  for(i in 1:N){
    Y <- MASS::mvrnorm(1, mu, sigma^2 * diag(p))
    mu_mle[i,] <- Y
    shrink <- 1 - ((p-2)*sigma^2) / sum(Y^2)
    mu_js[i,] <- shrink * Y
  }

  # record risk
  risk_mle_s[k] <- mean(rowSums((mu_mle - mu)^2))
  risk_js_s[k]  <- mean(rowSums((mu_js  - mu)^2))
}

# Plot
plot(sigma.values, risk_mle_s, type="b", col="red", pch=19,
     ylim=range(c(risk_mle_s, risk_js_s)),
     xlab="sigma", ylab="Risk",
     main="Risk vs Noise Level (sigma)", xaxt="n")
lines(sigma.values, risk_js_s, type="b", col="blue", pch=19)
legend("topleft", legend=c("MLE", "James-Stein"),
       col=c("red","blue"), pch=19, lty=1)
axis(1, at=sigma.values, labels=sigma.values)
```


Risk vs Noise Level (sigma)
MLE
James−Stein
Risk
sigma