ESI 4606 Analytics I - Foundations of Data Science
Homework 4
Due: October 19st (11:00AM), 2022

Problem 1 (1.5 points)

This question involves using R to perform the multiple linear regression using the "Auto" data from R library of "ISLR".

(a) Fit a multiple linear regression model to predict "mpg" using all other variables except "name" as the predictors. With significance level of 0.05, for which of the predictors can you reject the null hypothesis $H_0 : \beta_j = 0$?
Code:

```
#Check the predictors:
names(Auto)
#Remove the last predictor "name":
data_1 <- Auto[,c(-dim(Auto)[2])]
model_mlr <- lm(mpg~., data = data_1)
summary(model_mlr)
```

Summaries of the models are in the **Appendix**.
Conclusions: For the predictors displacement, weight, year and origin you can reject the null hypothesis.

(b) Based on answers of (a), re-fit a smaller model until all predictors in the model are significant based on significance level of 0.05. Provide an interpretation of each coefficient in the model.

```
model_mlr_1 <- lm(mpg~displacement+weight+year+origin, data = data_1)
summary(model_mlr_1) #displacement is not significant
#exclude displacement
model_mlr_2 <- lm(mpg~+weight+year+origin, data = data_1)
summary(model_mlr_2)
```

(c) Perform the log transformation of the response variable "mpg". Based on the transformed "mpg", fit a multiple linear regression model using all other variables except "name" as the predictors. With significance level of 0.05, for which of the predictors can you reject the null hypothesis $H_0 : \beta_j = 0$?

```
model_mlr_log <- lm(log(mpg)~., data = data_1)
summary(model_mlr_log)
```

Conclusions: For all predictors other than "acceleration" can we reject the null hyperthesis.

(d) Based on answers of (c), re-fit a smaller model until all predictors in the model are significant based on significance level of 0.05. How well do the final models in (d) and (b) fit the data?

```
data_2 <- data_1[,c(-6)]
model_mlr_log_1 <- lm(log(mpg)~., data = data_2)
summary(model_mlr_log_1)
```

Conclusions: With log(mpg), Residual standard error: 0.119 , Multiple R-squared: 0.8795, Adjusted R-squared: 0.8776
With original mpg, Residual standard error: 3.348, Multiple R-squared: 0.8175, Adjusted R-squared: 0.816
The model in part (d) fits better as it gets higher R-squared and lower RSE.

Note: **To get full points, include R codes in the appendix sections**

Problem 2 (1.5 points)

75% of the light aircraft that disappear while in flight in a certain country are subsequently discovered. Of the aircraft that are discovered, 65% have an emergency locator, whereas 95% of the aircraft not discovered do not have such a locator. Suppose a light aircraft has disappeared.

(a) What is the probability that it has an emergency locator?
Solution:
Let the event **Discovered** be **A**, **Not Discovered** be **ND**, **With Locator** be **L**, **Without Locator** be **NL**.
From the problem description, we have $P(D) = 0.75, P(ND) = 0.25, P(L|D) = 0.65, P(NL|D) = 0.35, P(NL|ND) = 0.95, P(L|ND) = 0.05$.
So $P(L) = P(L|D)P(D) + P(L|ND)P(ND) = 0.65 * 0.75 + 0.05 * 0.25 = 0.5$.

(b) If it has an emergency locator, what is the probability that it will not be discovered?

Solution:$P(ND|L) = \frac{P(L|ND)P(ND)}{P(L)} = \frac{0.05*0.25}{0.5} = 0.025$

(c) If it does not have an emergency locator, what is the probability that it will be discovered?
Solution:$P(D|NL) = \frac{P(NL|D)P(D)}{P(NL)} = \frac{0.35*0.75}{0.5} = 0.525$

Problem 3 (2 points)

Questions in this problem should be answered using the data set files of training data "HM4-train-2022.txt" and test data "HM4-test-2022.txt". After importing the data, please change response variable "y" in each data set as a factor object.

(a) Using "y" as response variable and all other variables as predictors, fit the logistic regression model based on **training data**. Compute the confusion matrix and the mis-classification error rate for **test data**.

```
train <- read.table("HM4-train-2022.txt")
train$y <- as.factor(train$y)
test <- read.table("HM4-test-2022.txt")
test$y <- as.factor(test$y)
model.logist <- glm(y~.,data=train,family=binomial)
pred.logist <- predict(model.logist,newdata=test,type="response")
pred.logist.class <- rep("A",nrow(test))
pred.logist.class[pred.logist>0.5]="B"
table(pred.logist.class,test$y)
mean(pred.logist.class!=test$y)
```

Table 1: Confusion matrix of prediction results using logistic regression model

	Actual class A	Actual class B
Predicted class A	47	5
Predicted class B	3	45

Misclassification error rate is 0.08.
(b) Repeat (a) using LDA.

```
library(MASS)
model.lda <- lda(y~.,data=train)
pred.lda <- predict(model.lda,newdata=test)
table(pred.lda$class,test$y)
mean(pred.lda$class!=test$y)
```

Misclassification error rate is 0.1.

Table 2: Confusion matrix of prediction results using LDA model

	Actual class A	Actual class B
Predicted class A	43	3
Predicted class B	7	47

(c) Repeat (a) using QDA.

```
model.qda <- qda(y~.,data=train)
pred.qda <- predict(model.qda,newdata=test)
table(pred.qda$class,test$y)
mean(pred.qda$class!=test$y)
```

Table 3: Confusion matrix of prediction results using LDA model

	Actual class A	Actual class B
Predicted class A	45	6
Predicted class B	5	44

Misclassification error rate is 0.11.
(d) Repeat (a) using KNN with K=2 (Note: set your random seed as 2022).

```
library(class)
set.seed(2022)
pred.knn <- knn(train[,1:2],test[,1:2],train[,3],k=2)
table(pred.knn,test$y)
mean(pred.knn!=test$y)
```

Table 4: Confusion matrix of prediction results using LDA model

	Actual class A	Actual class B
Predicted class A	41	10
Predicted class B	9	40

Misclassification error rate is 0.19.
(e) Which of these methods appears to provide the best prediction results? Why?
Logistic regression appears to provide the best predictions with the lowest misclassification error rate.
Note: **To get full points, include R codes in the appendix sections**

Appendix:

```
# summary(model_mlr)
Call:
lm(formula = mpg ~ ., data = data_1)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.5903 -2.1565 -0.1169  1.8690 13.0604 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
cylinders     -0.493376   0.323282  -1.526  0.12780    
displacement   0.019896   0.007515   2.647  0.00844 ** 
horsepower    -0.016951   0.013787  -1.230  0.21963    
weight        -0.006474   0.000652  -9.929  < 2e-16 ***
acceleration   0.080576   0.098845   0.815  0.41548    
year           0.750773   0.050973  14.729  < 2e-16 ***
origin         1.426141   0.278136   5.127 4.67e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.328 on 384 degrees of freedom
Multiple R-squared:  0.8215,	Adjusted R-squared:  0.8182 
F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16

# summary(model_mlr_1)

Call:
lm(formula = mpg ~ displacement + weight + year + origin, data = data_1)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.8102 -2.1129 -0.0388  1.7725 13.2085 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -1.861e+01  4.028e+00  -4.620 5.25e-06 ***
displacement  5.588e-03  4.768e-03   1.172    0.242    
weight       -6.575e-03  5.571e-04 -11.802  < 2e-16 ***
year          7.714e-01  4.981e-02  15.486  < 2e-16 ***
origin        1.226e+00  2.670e-01   4.593 5.92e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.346 on 387 degrees of freedom
Multiple R-squared:  0.8181,	Adjusted R-squared:  0.8162 
F-statistic: 435.1 on 4 and 387 DF,  p-value: < 2.2e-16

# summary(model_mlr_2)

Call:
```

```
lm(formula = mpg ~ +weight + year + origin, data = data_1)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.9440 -2.0948 -0.0389  1.7255 13.2722 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.805e+01  4.001e+00  -4.510 8.60e-06 ***
weight      -5.994e-03  2.541e-04 -23.588  < 2e-16 ***
year         7.571e-01  4.832e-02  15.668  < 2e-16 ***
origin       1.150e+00  2.591e-01   4.439 1.18e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.348 on 388 degrees of freedom
Multiple R-squared:  0.8175,	Adjusted R-squared:  0.816 
F-statistic: 579.2 on 3 and 388 DF,  p-value: < 2.2e-16

# summary(model_mlr_log)

Call:
lm(formula = log(mpg) ~ ., data = data_1)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.40955 -0.06533  0.00079  0.06785  0.33925 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   1.751e+00  1.662e-01  10.533  < 2e-16 ***
cylinders    -2.795e-02  1.157e-02  -2.415  0.01619 *  
displacement  6.362e-04  2.690e-04   2.365  0.01852 *  
horsepower   -1.475e-03  4.935e-04  -2.989  0.00298 ** 
weight       -2.551e-04  2.334e-05 -10.931  < 2e-16 ***
acceleration -1.348e-03  3.538e-03  -0.381  0.70339    
year          2.958e-02  1.824e-03  16.211  < 2e-16 ***
origin        4.071e-02  9.955e-03   4.089 5.28e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1191 on 384 degrees of freedom
Multiple R-squared:  0.8795,	Adjusted R-squared:  0.8773 
F-statistic: 400.4 on 7 and 384 DF,  p-value: < 2.2e-16

# summary(model_mlr_log_1)

Call:
lm(formula = log(mpg) ~ ., data = data_2)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.40671 -0.06629  0.00104  0.06899  0.33853 
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   1.723e+00  1.493e-01  11.539  < 2e-16 ***
cylinders    -2.772e-02  1.154e-02  -2.402  0.01679 *  
displacement  6.466e-04  2.673e-04   2.419  0.01601 *  
horsepower   -1.359e-03  3.876e-04  -3.505  0.00051 ***
weight       -2.594e-04  2.044e-05 -12.693  < 2e-16 ***
year          2.963e-02  1.817e-03  16.309  < 2e-16 ***
origin        4.067e-02  9.944e-03   4.090 5.25e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.119 on 385 degrees of freedom
Multiple R-squared:  0.8795,	Adjusted R-squared:  0.8776 
F-statistic: 468.2 on 6 and 385 DF,  p-value: < 2.2e-16
```