R Notebook

```
#Setup
insurance = read.csv("insurance.csv")
data = insurance
head(data)
```

```
##   age gender    bmi children smoker    region   charges
## 1  19 female 27.900        0    yes southwest 16884.924
## 2  18   male 33.770        1     no southeast  1725.552
## 3  28   male 33.000        3     no southeast  4449.462
## 4  33   male 22.705        0     no northwest 21984.471
## 5  32   male 28.880        0     no northwest  3866.855
## 6  31 female 25.740        0     no southeast  3756.622
```

```
data$gender = as.factor(data$gender)
data$smoker = as.factor(data$smoker)
data$region = as.factor(data$region)
str(data)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ gender  : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
```

Problem 1: Insurance Data

a.

Done in setup

b.

```
fit = lm(charges ~ age + bmi + gender, data = data)
summary(fit)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi + gender, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14974  -7073  -5072   6953  47348
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6986.82    1761.04  -3.967 7.65e-05 ***
## age           243.19      22.28  10.917  < 2e-16 ***
## bmi           327.54      51.37   6.377 2.49e-10 ***
## gendermale   1344.46     622.66   2.159    0.031 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11370 on 1334 degrees of freedom
## Multiple R-squared:  0.1203, Adjusted R-squared:  0.1183
## F-statistic: 60.78 on 3 and 1334 DF,  p-value: < 2.2e-16
```

So we can see that age and bmi is statistically significant to charges as their p values is below 0.05.

c.

```
data$gender <- relevel(data$gender, ref = "female")
male1 = lm(charges ~ age + bmi + gender, data = data)
summary(male1)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi + gender, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14974  -7073  -5072   6953  47348
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6986.82    1761.04  -3.967 7.65e-05 ***
## age           243.19      22.28  10.917  < 2e-16 ***
## bmi           327.54      51.37   6.377 2.49e-10 ***
## gendermale   1344.46     622.66   2.159    0.031 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11370 on 1334 degrees of freedom
## Multiple R-squared:  0.1203, Adjusted R-squared:  0.1183
## F-statistic: 60.78 on 3 and 1334 DF,  p-value: < 2.2e-16
```

$Y = -6986.82 + 243.19(x1) + 327.54(x2) + 1344.46(x3) + e$

```r
data$gender <- relevel(data$gender, ref = "male")
female1 = lm(charges ~ age + bmi + gender, data = data)
summary(female1)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi + gender, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14974  -7073  -5072   6953  47348
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5642.35    1779.00  -3.172  0.00155 **
## age            243.19      22.28  10.917  < 2e-16 ***
## bmi            327.54      51.37   6.377 2.49e-10 ***
## genderfemale -1344.46     622.66  -2.159  0.03101 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11370 on 1334 degrees of freedom
## Multiple R-squared:  0.1203, Adjusted R-squared:  0.1183
## F-statistic: 60.78 on 3 and 1334 DF,  p-value: < 2.2e-16
```

$$y = -5642.35 + 243.19(x1) + 327.54(x2) + -1344.46(x3) + e$$

d.

```r
males = data[data$gender=="male",]
fit_males = lm(charges ~ age + bmi, data = males)
summary(fit_males)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi, data = males)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -16895  -7984  -5567   7904  47424
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8012.79    2619.78  -3.059  0.00231 **
## age           238.63      33.70   7.081 3.61e-12 ***
## bmi           409.87      77.11   5.315 1.45e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12210 on 673 degrees of freedom
## Multiple R-squared:  0.1168, Adjusted R-squared:  0.1142
## F-statistic:  44.5 on 2 and 673 DF,  p-value: < 2.2e-16
```

```
summary(fit_males$fitted.values)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2824   10767   13739   13957   17257   25936
```

```
females = data[data$gender=="female",]
fit_females = lm(charges ~ age + bmi, data = females)
summary(fit_females)
```

```
##
## Call:
## lm(formula = charges ~ age + bmi, data = females)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -10063  -5959  -4587  -1521  46239
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4515.22    2286.14  -1.975 0.048681 *
## age           246.92      29.03   8.504  < 2e-16 ***
## bmi           241.32      67.49   3.576 0.000375 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10440 on 659 degrees of freedom
## Multiple R-squared:  0.1224, Adjusted R-squared:  0.1197
## F-statistic: 45.94 on 2 and 659 DF,  p-value: < 2.2e-16
```

Male regression coefficients: 2619.78

Female regression coefficients: 2286.14

e. Compare your results in part (d) to part (c). Is the model you obtained for males only in part (c) the same as fit males? What about for females? Explain in plain language to your classmate why these two approaches will not give the same results.

The first model shows that there is some difference in the intercept (the cost of insurance) for males and females. While the second model better explain the difference in cost between males and females by showing how the age and bmi of either genders plays into the cost of their insurance.

f. The model from part (b) has a significant F-test statistic, which tells us the overall model is jointly significant and at least one of the regression coefficients is significantly different from zero. However, the R2 is quite low. Are these results contradictory? Explain.

```
fit = lm(charges ~ age + bmi + gender, data = data)
summary(fit)
```

```
##
```

```
## Call:
## lm(formula = charges ~ age + bmi + gender, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14974  -7073  -5072   6953  47348
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5642.35    1779.00  -3.172  0.00155 **
## age            243.19      22.28  10.917  < 2e-16 ***
## bmi            327.54      51.37   6.377 2.49e-10 ***
## genderfemale -1344.46     622.66  -2.159  0.03101 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11370 on 1334 degrees of freedom
## Multiple R-squared:  0.1203, Adjusted R-squared:  0.1183
## F-statistic: 60.78 on 3 and 1334 DF,  p-value: < 2.2e-16
```

The results are not contradictory. F stat explains the overall how well our model fits the model with no independent variables. R^2 explains how well our model fits the data. In this senario, high f stat and low r square mean we reject the null hypothesis that the coefficient are 0, but hte variance of the residual is very large.

Problem 2: Predictions in the presence of multicollinearity

a. Is multicollinearity a problem for making accurate predictions? If you're unsure, make an educated guess based on what we have learned in class.

Multidisciplinary undermines the statically significance of an independent variable, yet it does not effect the predictive accuracy of a model. So the key take away for a multilinearity problem is the the RSS will have a high error rate.

b.

```
set.seed(42)
x1 = runif(100)
x2 = 0.8*x1 + rnorm(100,0,0.1)

cor(x1, x2)
```

```
## [1] 0.9404249
```

```
y = 3 + 2*x1 + 4*x2
y
```

```
##   [1] 7.885762 7.559257 5.118217 7.575487 6.372981 5.809919 7.101975 3.736199
##   [9] 5.219124 6.780290 5.233363 6.813476 8.093025 4.888125 5.113006 8.409093
##  [17] 8.221117 4.026337 5.838276 6.202082 7.283716 3.685218 8.391644 7.541265
```

```
##  [25] 3.211544 5.906300 5.336330 7.895345 4.969932 6.907310 7.440580 7.320655
##  [33] 5.053539 6.514524 2.542800 7.575962 2.951282 4.006724 8.087666 6.509958
##  [41] 5.530555 5.075543 3.454781 8.618852 4.800791 7.635081 7.163630 5.744204
##  [49] 8.081019 6.479240 5.214208 5.220991 4.670841 7.819795 2.935760 6.935942
##  [57] 6.352937 3.841634 4.432935 5.722612 6.503121 8.153879 6.755456 5.744053
##  [65] 6.753947 3.832331 4.205630 8.387181 6.059819 4.305735 2.626092 3.142317
##  [73] 4.175085 5.094217 4.025805 6.569347 2.795532 4.142677 5.185021 3.079973
##  [81] 6.251389 3.623956 4.866972 6.806442 7.610224 5.492118 4.168330 3.948498
##  [89] 3.257291 4.566148 6.436175 2.646171 3.906690 7.840000 7.647805 7.262645
##  [97] 4.539577 5.505062 7.147413 5.797081
```

c.

```
y_train = y[1:50]
x1_train = x1[1:50]
x2_train = x2[1:50]

train = data.frame (y = y_train, x1 = x1_train, x2 = x2_train)
training = lm(y ~ x1+x2, data = train)
train_sum = summary(training)
```

```
## Warning in summary.lm(training): essentially perfect fit: summary may be
## unreliable
```

```
mean(train_sum$residuals^2)
```

```
## [1] 3.688194e-30
```

d.

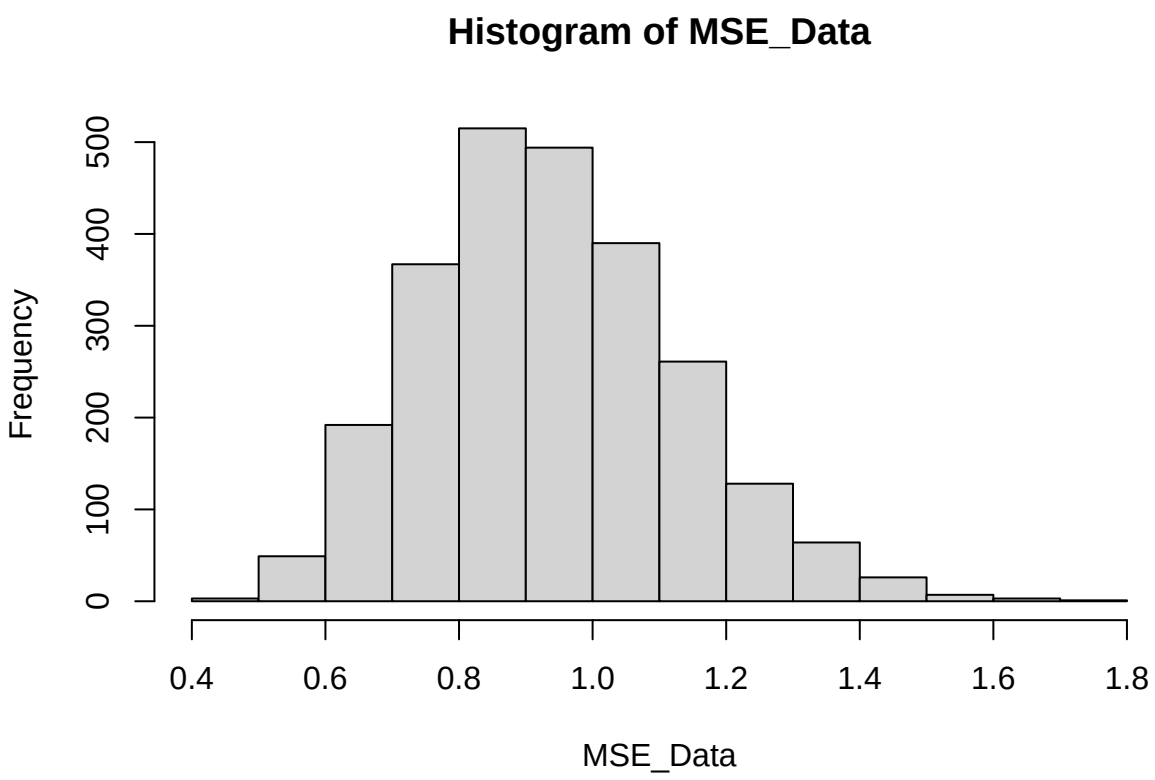
```
B = 2500
n = 50
MSE_Data = rep(NA, B)
for(i in 1:B){

  x1 = runif(100)
  x2 = 0.8*x1 + rnorm(100,0,0.1)
  error = rnorm(n,0,1)
  y = 3 + 2*x1 + 4*x2 + error

  y_train = y[1:50]
  x1_train = x1[1:50]
  x2_train = x2[1:50]

  train = data.frame (y = y_train, x1 = x1_train, x2 = x2_train)
  training = lm(y ~ x1+x2, data = train)
  MSE_Data[i] =  mean(training$residuals^2)
}

hist(MSE_Data)
```




```
mean(MSE_Data)
```

```
## [1] 0.9386311
```

e.

```
set.seed(24)
x1 = runif(100)
x2 = rnorm(100,0,1)

cor(x1, x2)
```

```
## [1] 0.03316596
```

```
y = 3 + 2*x1 + 4*x2
y
```

```
##   [1]  5.69175815 -0.84472197  7.48717341 11.12141580  3.54051046  5.65988651
##   [7]  1.17307887  9.24063192  8.80094612  5.75337485 10.01807845  5.37427602
##  [13]  2.58204435  9.62022941  5.92640757 -0.31164930 -1.55928643 -0.60195840
##  [19]  9.40029291 -2.99290062  3.76306912  0.56006753  4.34804286  1.96490768
##  [25]  3.14208801 -1.04331328 -0.69025700  5.09927772  3.76765347  3.09743413
```

```
##  [31]  1.35405327  5.16798490 -2.95590819 -1.24805422  6.32037342  7.65228516
##  [37] -3.51354724  7.30812403  4.42304141  5.34616426  3.00969625 -1.93007813
##  [43]  0.29178380  5.28390855  2.22775707  3.70597060  7.16508731 -1.50586572
##  [49]  7.69727708  8.51410399  4.25759636 15.85234639  6.31296062  4.71303686
##  [55]  1.86035894  1.77115283  1.86458140  1.35902895  4.30975860  0.99368524
##  [61] 11.91536507  5.34507357  4.72799791  1.20425540 -0.48292990 -6.38030091
##  [67] -0.71574872  1.33964640 -0.82012175  4.78146628  7.33408647  3.50175317
##  [73]  0.97140775  2.95808732  5.34786089  6.03415335  2.98101883  2.51721379
##  [79]  2.34896458  0.71344801 -5.81612111  3.03905924  4.99940063  9.55923976
##  [85]  6.09765396 -2.77525169  0.96447369 10.19043285  0.76603959  6.19004590
##  [91]  6.31330490  4.44997924  5.22864413  4.49918380  4.58709456 -0.04490678
##  [97]  6.05210019  7.42846601 -2.08398613  4.77174773
```

f.

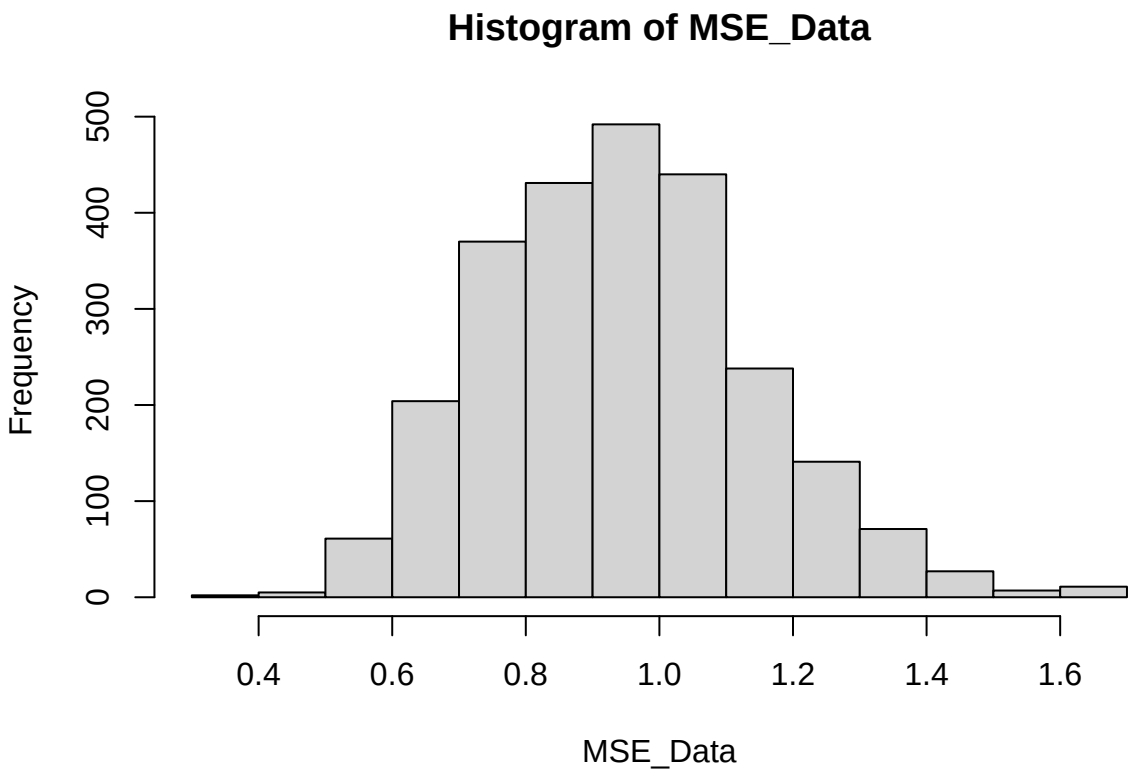
```
B = 2500
n = 50
MSE_Data = rep(NA, B)
for(i in 1:B){

  x1 = runif(100)
  x2 = 0.8*x1 + rnorm(100,0,0.1)
  error = rnorm(n,0,1)
  y = 3 + 2*x1 + 4*x2 + error

  y_train = y[1:50]
  x1_train = x1[1:50]
  x2_train = x2[1:50]

  train = data.frame (y = y_train, x1 = x1_train, x2 = x2_train)
  training = lm(y ~ x1+x2, data = train)
  MSE_Data[i] =  mean(training$residuals^2)
}

hist(MSE_Data)
```




```
mean(MSE_Data)
```

```
## [1] 0.9436961
```

g.

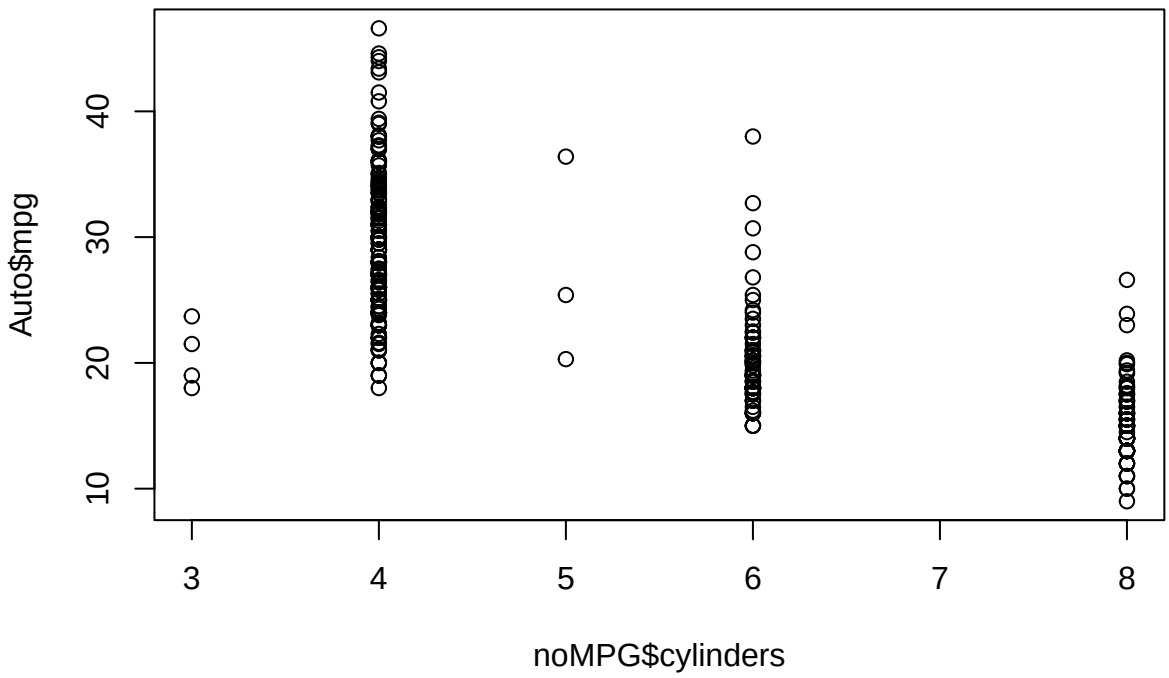
Based on our observation from both models, we can confidingly concluded that multicollinearity does not effect the accuracy of our models as the MSE is nearly identical.

Problem 3: Model Diagnostics

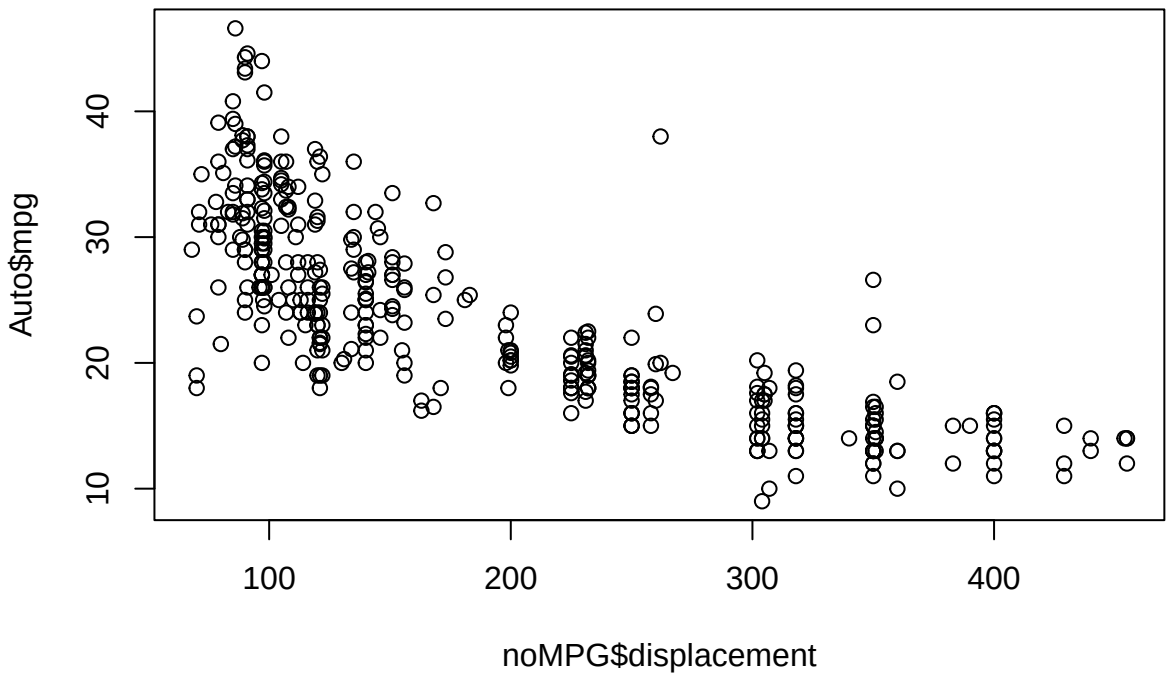
```
library(ISLR2)
```

a.

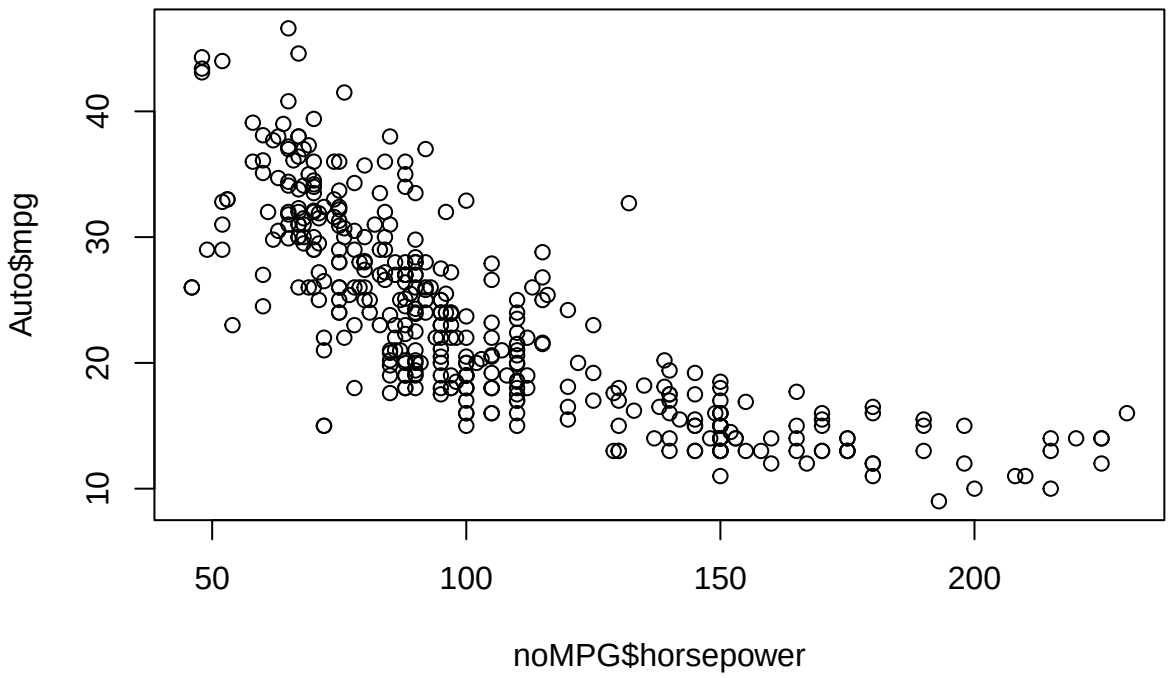
```
noMPG = Auto[, !names(Auto) %in% c("mpg", "name")]
plot(noMPG$cylinders, Auto$mpg)
```



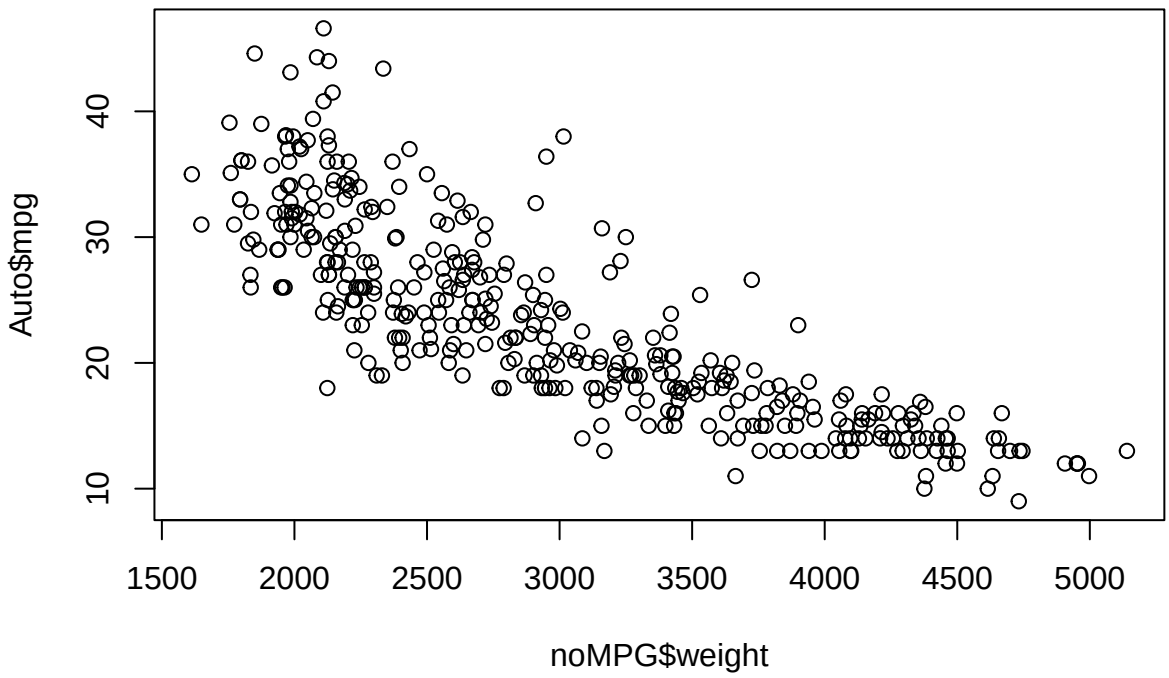

```
plot(noMPG$displacement, Auto$mpg)
```



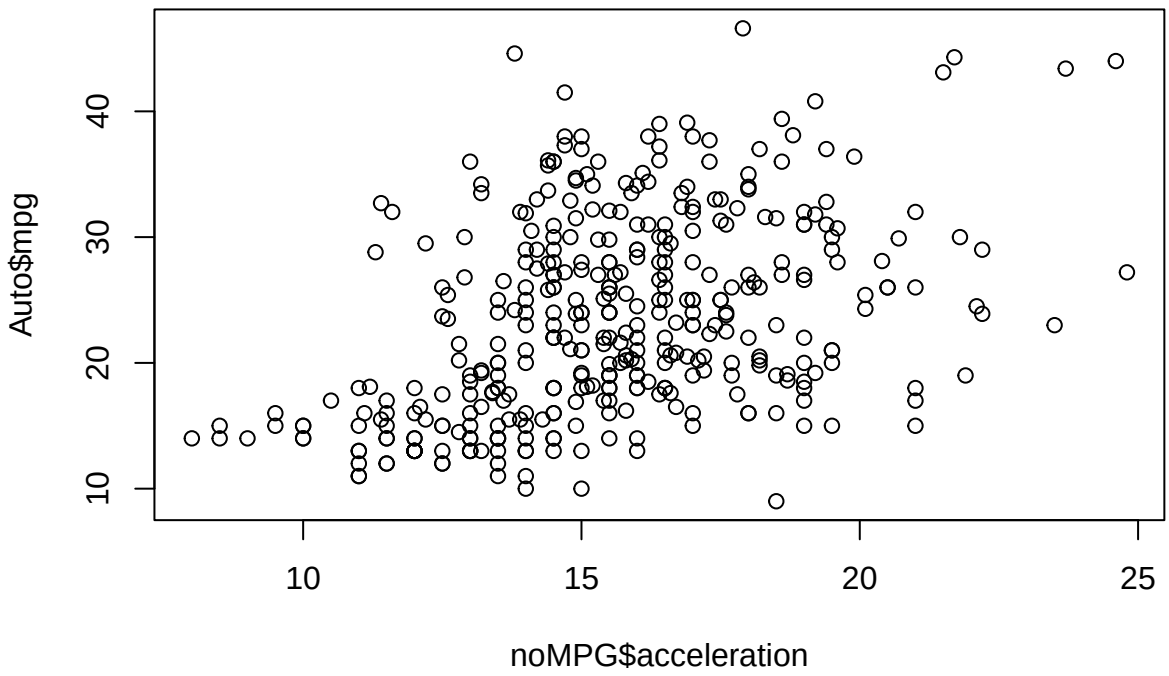

```
plot(noMPG$horsepower, Auto$mpg)
```



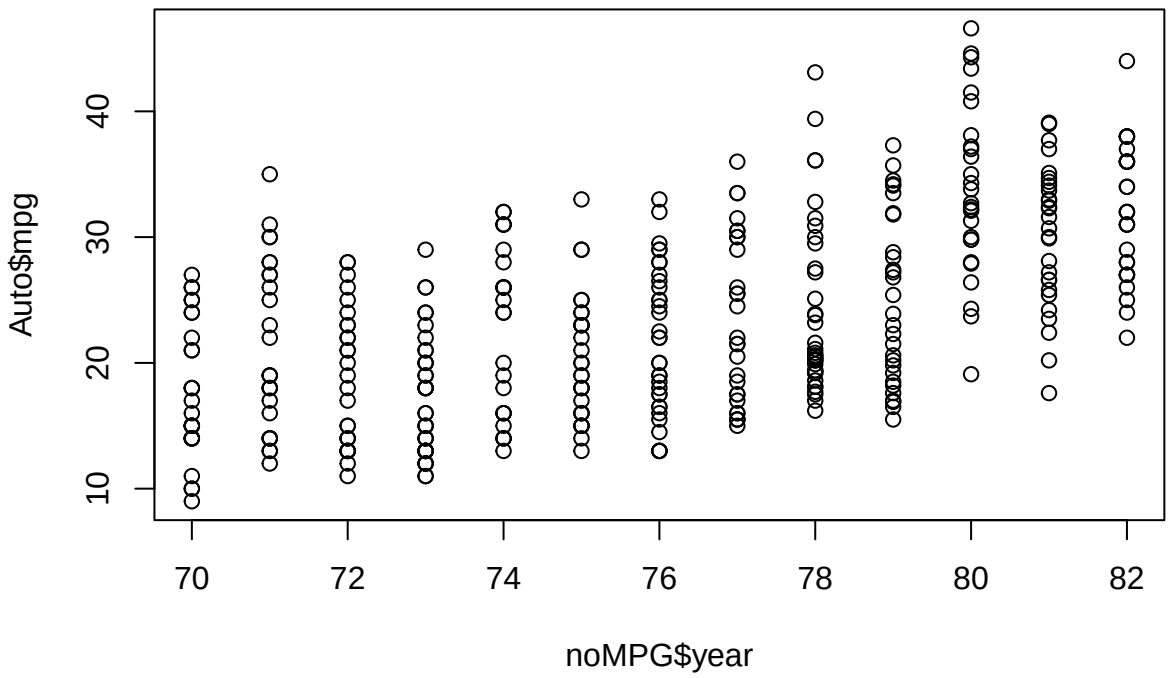

```
plot(noMPG$weight, Auto$mpg)
```



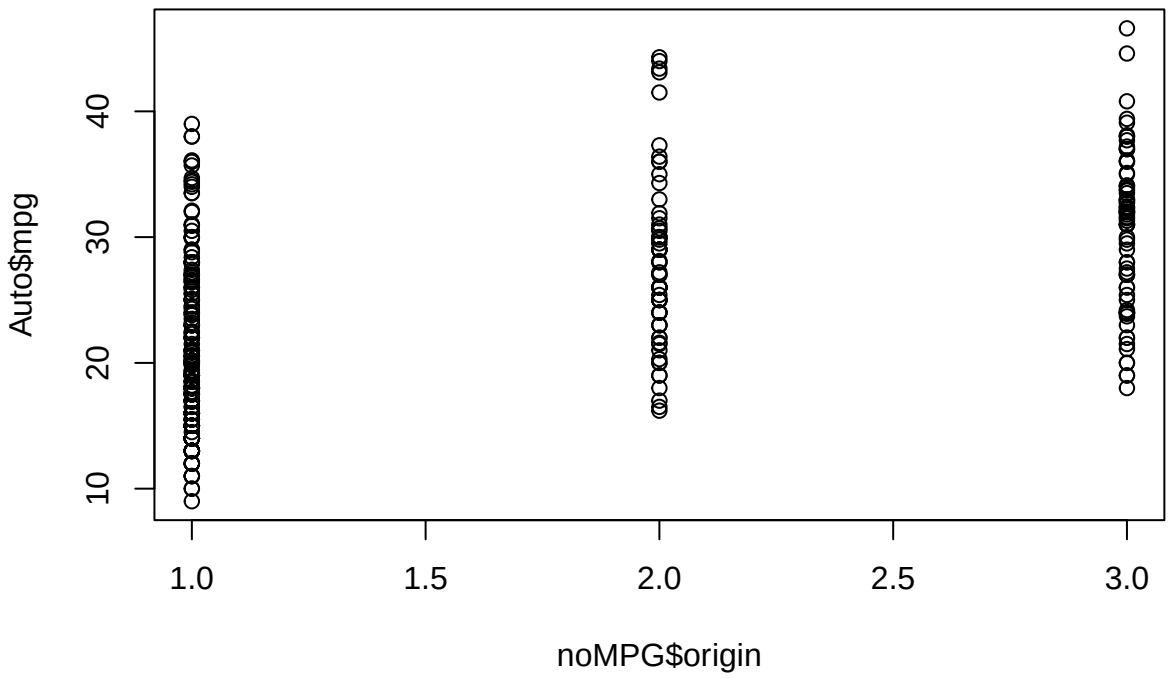

```
plot(noMPG$acceleration, Auto$mpg)
```




```
plot(noMPG$year, Auto$mpg)
```




```
plot(noMPG$origin, Auto$mpg)
```



I can see the horsepower and weight are the only non linear relationships with mpg.

b.

```
model3 = lm(Auto$mpg ~ ., data = noMPG)
summary(model3)
```

```
##
## Call:
## lm(formula = Auto$mpg ~ ., data = noMPG)
##
## 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
```

Based on our model, I can see that displacement, weight, year, and origin has a statistically significant association to mpg as their P values are incredibly small. Small p values mean that there is a very small percentage of the data in these variables that are considered random. So these predictors values are statistically significant.

c.

Yes, look at B.

d.

Based off our model, we can say that for every 1 increase in year, the mpg will increase by 0.75.

e.

No, based on what we had demonstrated in problem 2, we can safely concluded that multicollinearity is not an issue as it does not effect the accuracy of the model.

f.

```
summary(model3)
```

```
##
## Call:
## lm(formula = Auto$mpg ~ ., data = noMPG)
##
## 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
```

```
model_log = lm(Auto$mpg~log(cylinders) + log(displacement) + log(horsepower) + log(weight) + log(accele
summary(model_log)
```

```
##
## Call:
## lm(formula = Auto$mpg ~ log(cylinders) + log(displacement) +
##     log(horsepower) + log(weight) + log(acceleration) + log(year),
##     data = noMPG)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5641 -1.7873 -0.0611  1.5810 13.2714
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -62.413     17.650  -3.536 0.000456 ***
## log(cylinders)       2.750      1.626   1.691 0.091585 .
## log(displacement)   -3.406      1.355  -2.513 0.012371 *
## log(horsepower)     -6.386      1.563  -4.085 5.36e-05 ***
## log(weight)        -11.905      2.240  -5.316 1.80e-07 ***
## log(acceleration)   -5.326      1.622  -3.283 0.001119 **
## log(year)           54.825      3.595  15.250  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.103 on 385 degrees of freedom
## Multiple R-squared:  0.8444, Adjusted R-squared:  0.8419
## F-statistic: 348.1 on 6 and 385 DF,  p-value: < 2.2e-16
```

Based on our models, we can see that linearity holds as there are less statistically significant values in the logamatic problems model compare to a linear model.

g.

```
summary(model3)
```

```
##
## Call:
## lm(formula = Auto$mpg ~ ., data = noMPG)
##
## 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
```

```
model_log = lm(Auto$mpg~log(cylinders) + log(displacement) + log(horsepower) + log(weight) + log(accele
summary(model_log)
```

```
##
## Call:
## lm(formula = Auto$mpg ~ log(cylinders) + log(displacement) +
##     log(horsepower) + log(weight) + log(acceleration) + log(year),
##     data = noMPG)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5641 -1.7873 -0.0611  1.5810 13.2714
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -62.413     17.650  -3.536 0.000456 ***
## log(cylinders)       2.750      1.626   1.691 0.091585 .
## log(displacement)   -3.406      1.355  -2.513 0.012371 *
## log(horsepower)     -6.386      1.563  -4.085 5.36e-05 ***
## log(weight)        -11.905      2.240  -5.316 1.80e-07 ***
## log(acceleration)   -5.326      1.622  -3.283 0.001119 **
## log(year)           54.825      3.595  15.250  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.103 on 385 degrees of freedom
## Multiple R-squared:  0.8444, Adjusted R-squared:  0.8419
## F-statistic: 348.1 on 6 and 385 DF,  p-value: < 2.2e-16
```

Let's run a log transformation and see if we can generate more p values than linear regression and a lower R^2. After generating the model, we can see that log transformation is less effective than a linear model. The linear model has more variables that is statistically significant and it has a smaller adjusted R^2.

Problem 4: Wrapping up Multiple Linear Regression

a. Write out the population multiple linear regression model.

y = B0 + B1x2 + B2x2 ... Bkxk + e

b. Conceptually, how do we obtain the least square estimates for a multiple linear regression model?

Our goal is to find coefficient estimates B hat 0 and B hat 1 such that the linear model fits the data well. In other words, we want to find the line to be as closed as possible to the data points.

c. Are these least square estimates trustworthy? How do we know? Explain any key concepts in plain language.

Least squares estimators cannot really be fully trusted as they are simply estimators and do not represent the true value of the data. Due to the nature that the model could be flawed, the LSE will only carry that bias.

d.What is our estimate for E(Y) for specific values of X? Clearly define any quantities. How do we quantify any uncertainty about our estimate for E(Y)?

E(y) is the True prediction prediction error in a model. The TPE is equal to the Training error + the test error. E(y) produced the true error of our model based on our predictors.

e. What is our prediction for Y for specific values of X? Clearly define any quantities. How do we quantify any uncertainty about our prediction?

Yhat is the outcome generated by our combination of predictors (X) by our model. We assume taht there is some relationship between Y and X, our goal is the estimate (learn) the function f, using a set of training data (Yhat = fhat(x)). Where fhat represents the resulting prediction for y.

f. How can we evaluate how good our model is at prediction? Explain what the bias-variance tradeoff tells us about model behavior.

We use MSE to evaluate the training and test MSE. The lower the MSE, the less error there is our model compare to the actual data.

g. What is statistical inference and why is it useful in the context of linear regression models?

Use stats to to generate statistical metrics there the user utilized logic to determine the ultimate result.

h. What are 3 potential issues that may arise with our multiple linear regression model? For each of these issues, explain 1. why the issue can cause problems and 2. what can be done to resolve the issue.

Non-linearity * The reason for this problem is one of the assumptions involved in linear regression. It is the assumption for linearity, which states that the relation between the predictor and response is linear. If the

actual relation between response and the predictor is not linear, then all the conclusion we draw becomes null and void. Also, the accuracy of the model may drop significantly.

correlation of error terms * A principal assumption of the linear model is that the error terms are uncorrelated. The "uncorrelated" terms indicated that the sign of error for one observation is independent of others. The correlation among error terms may occur due to several factors. For instance, if we are observing the weight and height of people. The correlation in error may occur due to the diet they consume, the exercise they do, environmental factors, or they are members of the same family. What happens to the model when errors are correlated? If the error terms are correlated then the standard error in the model coefficients gets underestimated. As a result, confidence and prediction intervals will be narrower than they should be.

non constant variance of error terms * The source of this problem is also an assumption. The assumption is that the error term has a constant variance, also referred to as Homcedacity. Generally, that is not the case. We can often identify a non-constant variance in errors, or heteroscedasticity, from the presence of funnel shape in residual plots. In Fig.2, the funnel represents that the error terms have non-constant variance.