705604096_stats101c_hw1

Jade Gregory 705604096

2023-10-11

Question 1

a)

```r
traindat <- read.csv("HW1TRData.csv")
testdat <- read.csv("HW1TSData.csv")
dim(traindat)
```

```
## [1] 700   3
```

```r
dim(testdat)
```

```
## [1] 301   3
```

Our testing data has dimensions 301 rows by 3 columns and our training data has dimensions 700 rows by 3 columns.

b)

```r
model1 <- lm(y ~ x, data = traindat)
model2 <- lm(y ~ poly(x, 2), data = traindat)
model3 <- lm(y ~ poly(x, 3), data = traindat)
model4 <- lm(y ~ poly(x, 4), data = traindat)
model5 <- lm(y ~ poly(x, 5), data = traindat)
```

```r
mse1 <- anova(model1)["Residuals", "Sum Sq"]/700
mse2 <- anova(model2)["Residuals", "Sum Sq"]/700
mse3 <- anova(model3)["Residuals", "Sum Sq"]/700
mse4 <- anova(model4)["Residuals", "Sum Sq"]/700
mse5 <- anova(model5)["Residuals", "Sum Sq"]/700
print(c(mse1, mse2, mse3, mse4, mse5))
```

```
## [1] 5.575408e+22 4.161704e+22 4.027701e+22 4.026535e+22 4.025010e+22
```

The MSE for model 1 is 5.575408e+22. The MSE for model 2 is 4.161704e+22. The MSE for model 3 is 4.027701e+22. The MSE for model 4 is 4.026535e+22. The MSE for model 5 is 4.025010e+22.

```r
r2m1 <- round(summary(model1)$r.squared, 3)
r2m2 <- round(summary(model2)$r.squared, 3)
r2m3 <- round(summary(model3)$r.squared, 3)
r2m4 <- round(summary(model4)$r.squared, 3)
r2m5 <- round(summary(model5)$r.squared, 3)
print(c(r2m1, r2m2, r2m3, r2m4, r2m5))
```

```
## [1] 0.457 0.595 0.608 0.608 0.608
```

The r-squared value for model 1 is 0.457. The r-squared value for model 2 is 0.595. The r-squared value for model 3 is 0.608. The r-squared value for model 4 is 0.608. The r-squared value for model 5 is 0.608.

```r
r2a1 <- summary(model1)$adj.r.squared
r2a2 <- summary(model2)$adj.r.squared
r2a3 <- summary(model3)$adj.r.squared
r2a4 <- summary(model4)$adj.r.squared
r2a5 <- summary(model5)$adj.r.squared
print(c(r2a1, r2a2, r2a3, r2a4, r2a5))
```

```
## [1] 0.4563467 0.5936135 0.6061337 0.6056812 0.6052625
```

The r-squared adjusted value for model 1 is 0.4563467. The r-squared adjusted value for model 2 is 0.5936135. The r-squared adjusted value for model 3 is 0.6061337. The r-squared adjusted value for model 4 is 0.6056812. The r-squared adjusted value for model 5 is 0.6052625.

```r
AIC1 <- extractAIC(model1)
AIC2 <- extractAIC(model2)
AIC3 <- extractAIC(model3)
AIC4 <- extractAIC(model4)
AIC5 <- extractAIC(model5)
print(c(AIC1, AIC2, AIC3, AIC4, AIC5))
```

```
##  [1]     2.00 36666.67     3.00 36463.96     4.00 36443.05     5.00 36444.84
##  [9]     6.00 36446.58
```

The AIC value for model 1 is 36666.67. The AIC value for model 2 is 36463.96. The AIC value for model 3 is 36443.05. The AIC value for model 4 is 36444.84. The AIC value for model 5 is 36446.58.

```r
BIC1 <- extractAIC(model1, k = log(700))
BIC2 <- extractAIC(model2, k = log(700))
BIC3 <- extractAIC(model3, k = log(700))
BIC4 <- extractAIC(model4, k = log(700))
BIC5 <- extractAIC(model5, k = log(700))
print(c(BIC1, BIC2, BIC3, BIC4, BIC5))
```

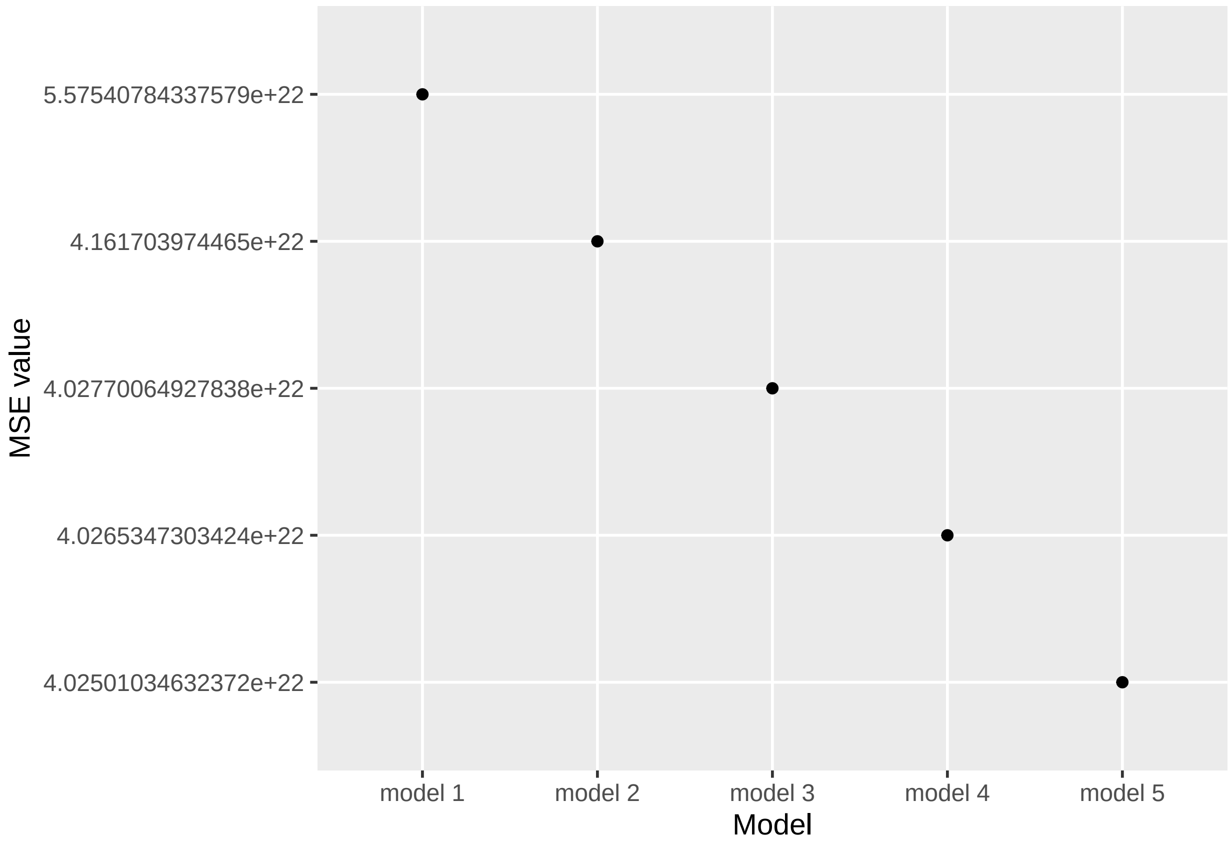
```
##  [1]     2.00 36675.77     3.00 36477.61     4.00 36461.25     5.00 36467.60
##  [9]     6.00 36473.89
```

The BIC value for model 1 is 36675.77. The BIC value for model 2 is 36477.61.The BIC value for model 3 is 36461.25. The BIC value for model 4 is 36467.60. The BIC value for model 5 is 36473.89.

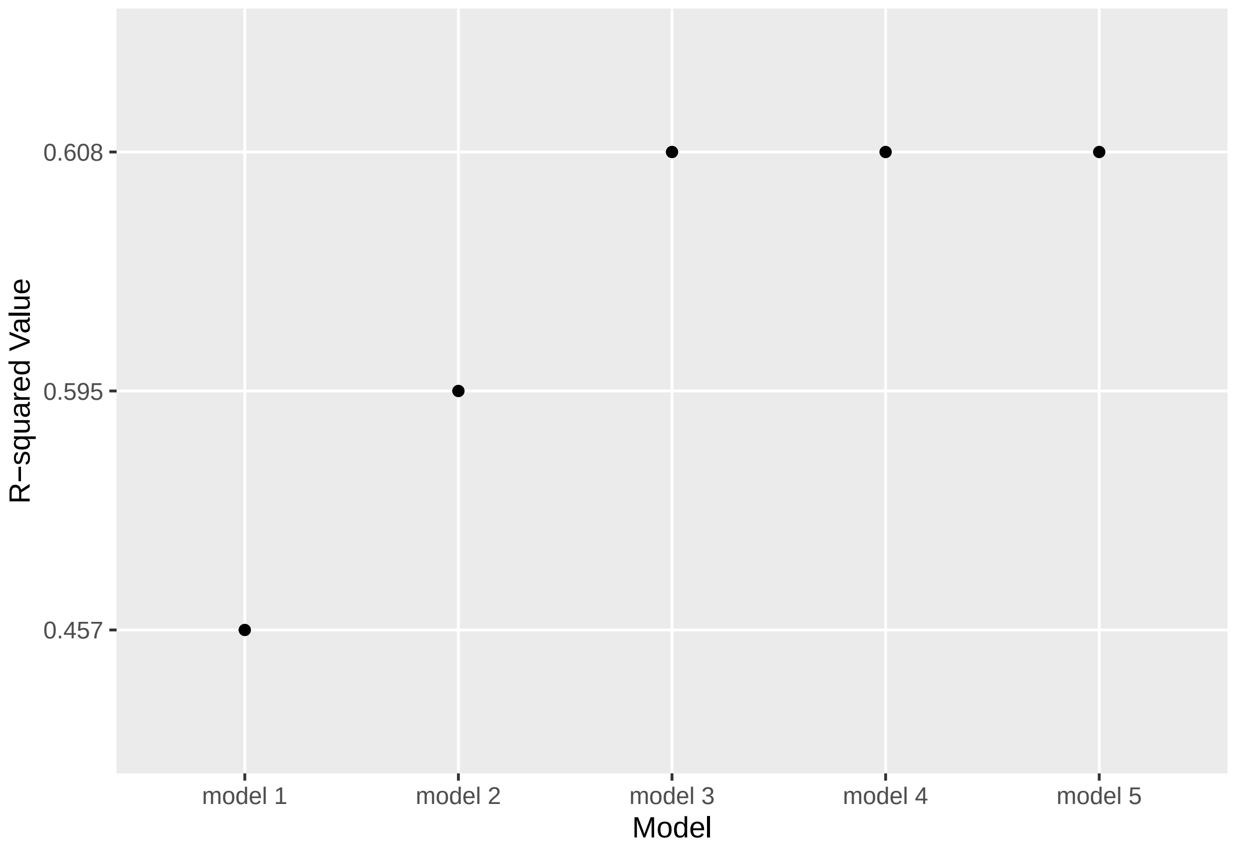
c)

```
#MSE plot
MSEs <- cbind(c("model 1", "model 2", "model 3", "model 4", "model 5"),c(mse1, mse2, mse3, mse4, mse5))
qplot(MSEs[,1], MSEs[,2]) + xlab("Model") + ylab("MSE value")
```

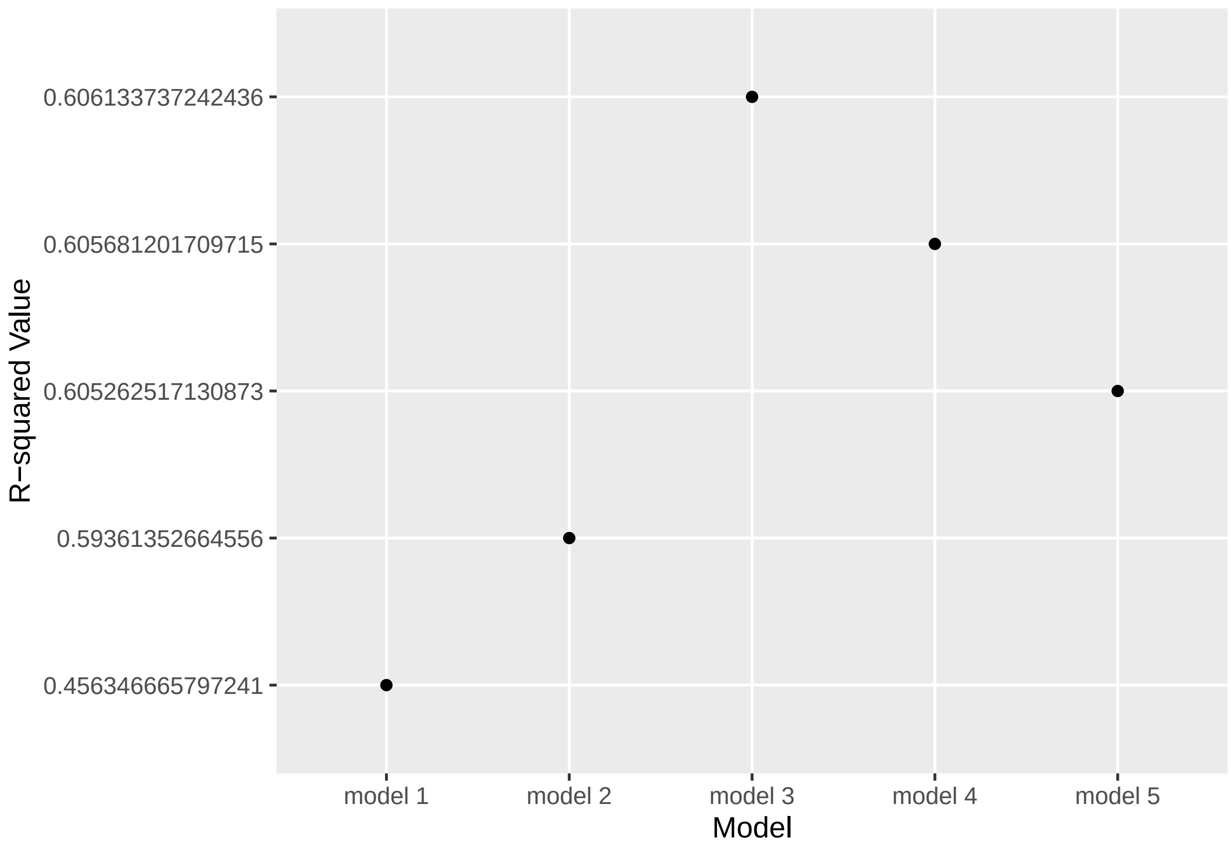
```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



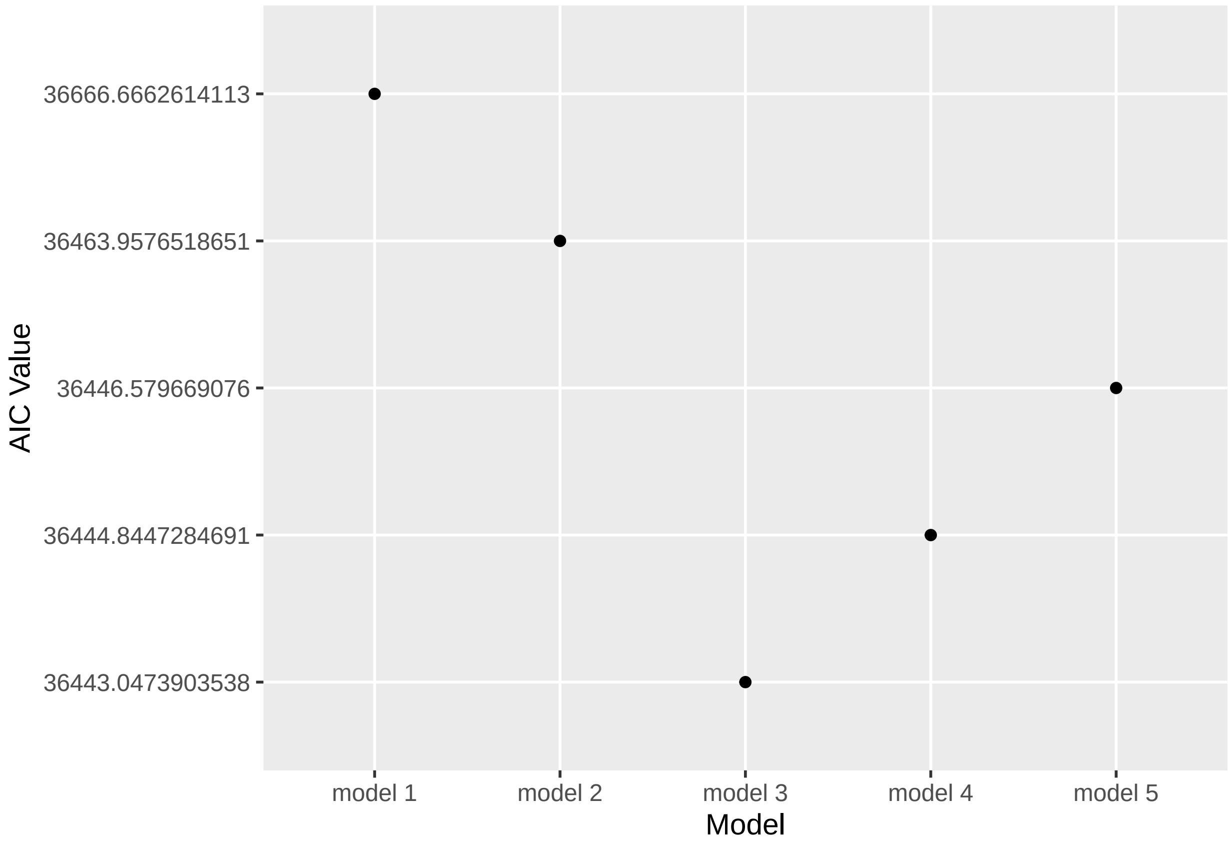
```
#R^2 plot
models <- cbind(c("model 1", "model 2", "model 3", "model 4", "model 5"),c(r2m1, r2m2, r2m3, r2m4, r2m5
qplot(models[,1], models[,2]) + xlab("Model") + ylab("R-squared Value")
```



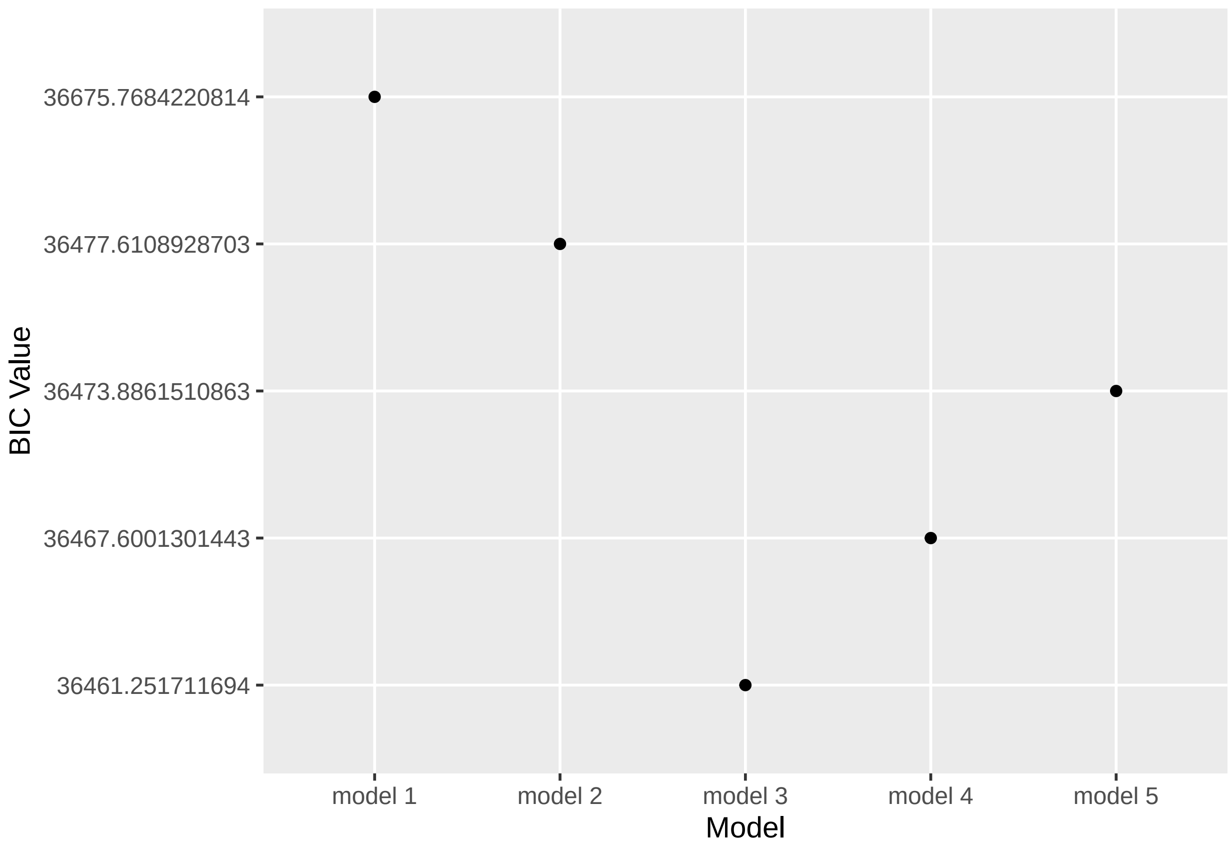
```
#R adj plot
modelsadj <- cbind(c("model 1", "model 2", "model 3", "model 4", "model 5"),c(r2a1, r2a2, r2a3, r2a4, r
qplot(modelsadj[,1], modelsadj[,2]) + xlab("Model") + ylab("R-squared Value")
```



```
#AIC plot
maic <- cbind(c("model 1", "model 2", "model 3", "model 4", "model 5"),c(AIC1[2], AIC2[2], AIC3[2], AIC4
qplot(maic[,1], maic[,2]) + xlab("Model") + ylab("AIC Value")
```



```
#BIC plot
mbic <- cbind(c("model 1", "model 2", "model 3", "model 4", "model 5"),c(BIC1[2], BIC2[2], BIC3[2], BIC4
qplot(mbic[,1], mbic[,2]) + xlab("Model") + ylab("BIC Value")
```



Based off of the models MSE values, model 5 would be the best regression model since it has the lowest MSE value. Based on the models r-squared values, model 3, 4, and 5 are the best fit for our data since they all have the same highest value for their r-squared. The best model based off of the adjusted r-squared values is model 3 since it has the largest adjusted r-squared value. Based off of AIC values, model 3 would be the best model since it has the lowest value. Based off of BIC values, model 3 would also be the best fit model since it has the lowest BIC value.

d)

```
coef(model3)
```

```
##  (Intercept)  poly(x, 3)1  poly(x, 3)2  poly(x, 3)3
## 1.816121e+11 5.732636e+12 3.145779e+12 9.685160e+11
```

```
coef(model4)
```

```
##  (Intercept)  poly(x, 4)1  poly(x, 4)2  poly(x, 4)3  poly(x, 4)4
## 1.816121e+11 5.732636e+12 3.145779e+12 9.685160e+11 9.034065e+10
```

```
coef(model5)
```

```
##   (Intercept)   poly(x, 5)1   poly(x, 5)2   poly(x, 5)3   poly(x, 5)4
##  181612072356 5732636476846 3145779248831  968516017579   90340647285
##   poly(x, 5)5
## -103299022894
```

The partial slopes for model 3 are 5.732636e+12 * poly(x, 3)1, 3.145779e+12 * poly(x, 3)2, and 9.685160e+11 * poly(x, 3)3, The partial slopes for model 4 are 5.732636e+12 * poly(x, 4)1, 3.145779e+12 * poly(x, 4)2, 9.685160e+11 * poly(x, 4)3, and 9.034065e+10 * poly(x, 4)4. The partial slopes for model 5 are 5732636476846 * poly(x, 5)1, 3145779248831 * poly(x, 5)2, 968516017579 * poly(x, 5)3, 90340647285 * poly(x, 5)4, and -103299022894 * poly(x, 5)5.

e)

```
#model 3 model 4 f test
anova(model3, model4)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ poly(x, 3)
## Model 2: y ~ poly(x, 4)
##   Res.Df        RSS Df  Sum of Sq      F Pr(>F)
## 1    696 2.8194e+25
## 2    695 2.8186e+25  1 8.1614e+21 0.2012 0.6539
```

```
#model 3 model 5 f test
anova(model3, model5)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ poly(x, 3)
## Model 2: y ~ poly(x, 5)
##   Res.Df        RSS Df  Sum of Sq      F Pr(>F)
## 1    696 2.8194e+25
## 2    694 2.8175e+25  2 1.8832e+22 0.2319 0.7931
```

```
#model 4 model 5 f test
anova(model4, model5)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ poly(x, 4)
## Model 2: y ~ poly(x, 5)
##   Res.Df        RSS Df  Sum of Sq      F Pr(>F)
## 1    695 2.8186e+25
## 2    694 2.8175e+25  1 1.0671e+22 0.2628 0.6083
```

From our first partial f test that compares model 3 with model 4, we can conclude that model 4 is not statistically significant since it has a p-value of 0.6539 which is greater than our significance level of 0.05. In our partial f test comparing model 3 with model 5, we can conclude that model 5 is not statistically significant since it has a p-value of 0.7931 which is greater than our significance level of 0.05. In our partial f test comparing model 4 with model 5, we can conclude that model 5 is not statistically significant since it has a p-value of 0.6083 which is greater than our significance level of 0.05. From this, I would conclude that model 3 is the best model since model 4 and model 5 did not appear to be statistically significant when comparing to model 3, and it is the simplest model between the three models we have compared.

f)

```r
# generate predicted values
predy1 <- predict(model1, newdata = testdat)
predy2 <- predict(model2, newdata = testdat)
predy3 <- predict(model3, newdata = testdat)
predy4 <- predict(model4, newdata = testdat)
predy5 <- predict(model5, newdata = testdat)

# mse tests
testmse1 <- sum((testdat$y - predy1)^2) / 301
testmse2 <- sum((testdat$y - predy2)^2) / 301
testmse3 <- sum((testdat$y - predy3)^2) / 301
testmse4 <- sum((testdat$y - predy4)^2) / 301
testmse5 <- sum((testdat$y - predy5)^2) / 301
print(c(testmse1, testmse2, testmse3, testmse4, testmse5))
```

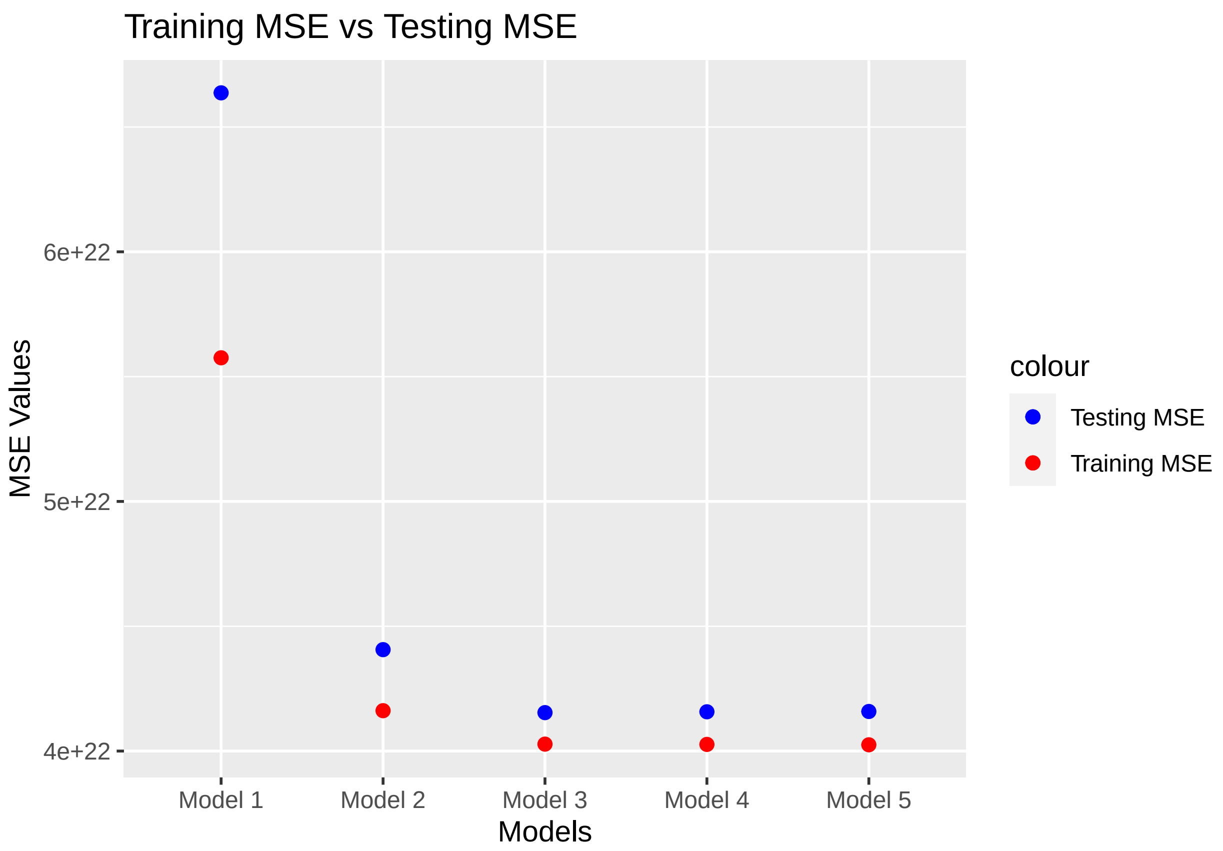
```
## [1] 6.636844e+22 4.406205e+22 4.153921e+22 4.157265e+22 4.158470e+22
```

The MSE for model 1 is 6.636844e+22. The MSE for model 2 is 4.406205e+22. The MSE for model 3 is 4.153921e+22. The MSE for model 4 is 4.157265e+22. The MSE for model 5 is 4.158470e+22.

g)

```r
msedat <- data.frame(Model = c("Model 1", "Model 2", "Model 3", "Model 4", "Model 5"),
                     TrainingMSE = c(mse1, mse2, mse3, mse4, mse5),
                     TestingMSE = c(testmse1, testmse2, testmse3, testmse4, testmse5))

ggplot(msedat, aes(x = Model)) +
  geom_point(aes(y = TrainingMSE, color = "Training MSE"), size = 2) +
  geom_point(aes(y = TestingMSE, color = "Testing MSE"), size = 2) +
  scale_color_manual(values = c("Training MSE" = "red", "Testing MSE" = "blue")) +
  labs(x = "Models", y = "MSE Values") +
  ggtitle("Training MSE vs Testing MSE")
```



Overall, our training MSE values were less than our testing MSEE values for each model. From our answers from part b and part c as well as the MSE values for both data sets, I would say that model 3 was the model used to create the data. This is because when comparing the MSEs, R-squared, R-Squared Adjusted, AIC, and BIC values for each model, model 3 appears to be the best. Also, we determined from the partial f tests that model 3 would be the best model for our data while also being the simplest model of the top models. I think it is safe to conclude that model 3 is the best fit for our data.

Question 2

```r
heartdat <- read.csv("HW1 F2023 Q2 HD Data.csv")
head(heartdat)
```

```
##   Ob    Sex Age ChestPainType RestingBP Cholesterol FastingBS RestingECG MaxHR
## 1  1 Female  37           ASY       110         265        NO        LVH   140
## 2  2 Female  37           ATA       100         226        NO     Normal   130
## 3  3      M  38           NAP       145         292        NO     Normal   130
## 4  4   Male  37            TA       105         220        NO         ST   150
## 5  5   Male  72           ATA       130           0       Yes     Normal    86
## 6  6   Male   5           ASY       135         342        NO     Normal   170
##   ExerciseAngina Oldpeak ST_Slope hypertension ever_married     work_type
## 1              N     2.0     Down           No          Yes       Private
## 2              Y     0.0       Up           No          Yes Self-employed
## 3              N     0.0       Up           No         <NA>          <NA>
## 4              Y     1.4       Up           No          Yes       Private
## 5              N     1.5     Flat           No          Yes       Private
## 6              N     0.0     Flat           No           No      children
##   Residence_type avg_glucose_level  bmi  smoking_status stroke HeartDisease
## 1          Rural             75.18 48.2 formerly smoked     No           No
## 2          Urban             95.08 34.1    never smoked     No          Yes
## 3           <NA>             85.86 27.5            <NA>     No           No
## 4          Rural            107.06 29.9          smokes     No           No
## 5          Urban            234.27 26.9    never smoked     No          Yes
## 6          Rural            100.98 19.0         Unknown     No           No
```

a) Two questions estimating parameters:

1. What is the average bmi value for all females ages 20 to 30?
2. Does avg_glucose_level affect bmi?

Two prediction questions: 1. What is the bmi value for a female of age 20? 2. What is the MaxHR value for a 25 year old female with a restingBP of 100?

b) I will answer the prediction question asking "what is the MaxHR value for a 25 year old female with a restingBP of 100?"

```
lmtest <- lm(MaxHR~ Age + Sex + RestingBP, data = heartdat)
predict(lmtest, newdata = data.frame(Age = 25, Sex = "Female", RestingBP = 100))
```

```
##        1
## 168.5614
```

From my model, the MaxHR of a 25 year old female with a restingBP of 100 is 168.5614. I answered this question by created a model with the Age, Sex, and RestingBP variables modeled against the MaxHR variable. Then I used the predict function to predict the value with my new variable values.

Question 3

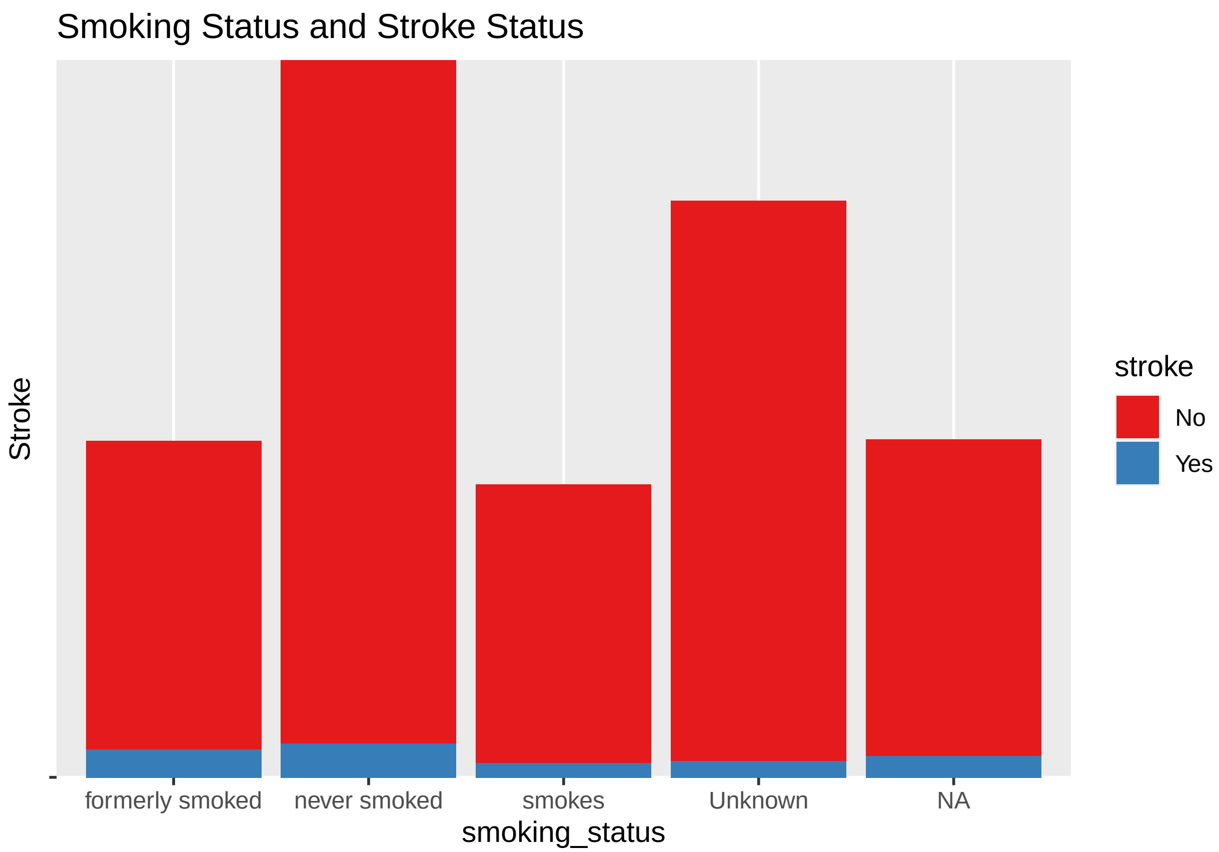
a)

```
conTable <- table(heartdat$smoking_status, heartdat$stroke)
print(conTable)
```

```
##
##                        No  Yes
##   formerly smoked  575   53
##   never smoked    1274   64
##   smokes           520   27
##   Unknown         1045   31
```

b)

```
ggplot(heartdat, aes(fill = stroke, y = "", x = smoking_status)) +
    geom_bar(position = "stack", stat = "identity") +
    labs(title = "Smoking Status and Stroke Status",
         y = "Stroke") +
    scale_fill_brewer(palette = "Set1")
```



c) Based on the bar chart, I do not believe that smoking status and stroke status are correlated. There does not appear to be any correlation between the two variables in the graph. If there is dependence, then it appears that those who have formerly smoked have the highest concentration of stroke victims among their sample population.

d)

```
chisq.test(table(heartdat$stroke, heartdat$smoking_status))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(heartdat$stroke, heartdat$smoking_status)
## X-squared = 26.455, df = 3, p-value = 7.659e-06
```

Because we have a p-value of 7.659e-06 which is less than our significance level of 0.05 meaning we reject our null hypothesis that states the two variables are independent. This means that there is significant evidence to suggest that the variables are dependent.

Question 4

```
a <- 537.1 - 519.8
b <- 3 - 1
```

```r
c <- 215 - 3
d <- b + c
e <- a / b
f <- 519.8 / c
g <- e / f
h <- 0.0158
print(c(a, b, c, d, e, f, g, h))
```

```
## [1]  17.300000   2.000000 212.000000 214.000000   8.650000   2.451887   3.527895
## [8]   0.015800
```

The value of A is 17.3. The value of B is 2. The value of C is 212. The value of D is 214. The value of E is 8.65. The value of F is 2.451887. The value of G is 3.527895. The value of H is 0.0158.