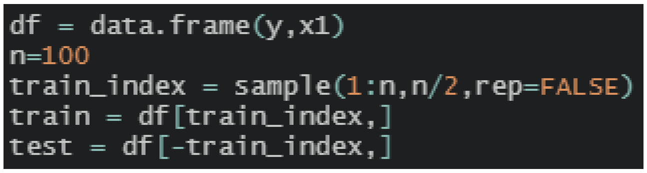
Neha Maddali

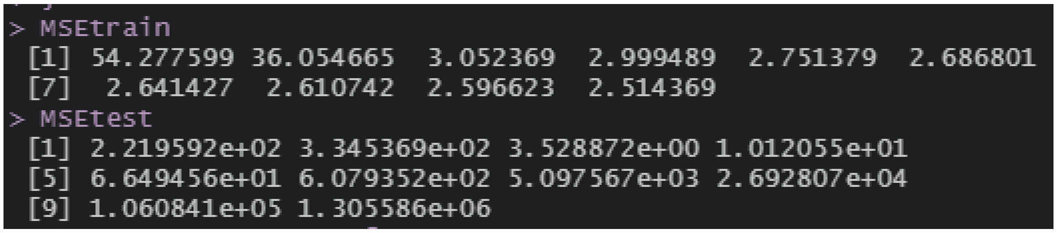
**Problem 1:**

a. x1 = rnorm(100)
b. error = rnorm(100,0,2)
c. B0 = 2, B1 = 3, B2 = 4, B3 = 5
d.



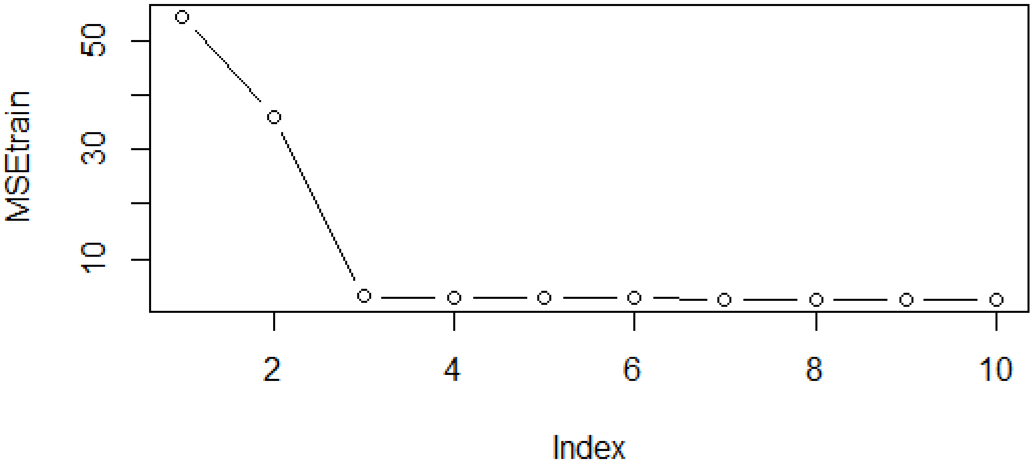
```
df = data.frame(y,x1)
n=100
train_index = sample(1:n,n/2,rep=FALSE)
train = df[train_index,]
test = df[-train_index,]
```

e.

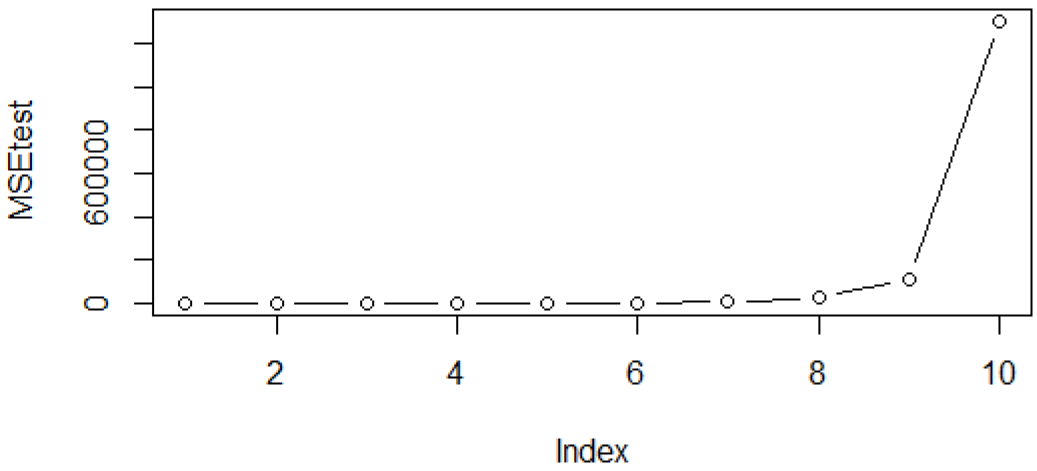


```
> MSEtrain
 [1] 54.277599 36.054665  3.052369  2.999489  2.751379  2.686801
 [7]  2.641427  2.610742  2.596623  2.514369
> MSEtest
 [1] 2.219592e+02 3.345369e+02 3.528872e+00 1.012055e+01
 [5] 6.649456e+01 6.079352e+02 5.097567e+03 2.692807e+04
 [9] 1.060841e+05 1.305586e+06
```

f.

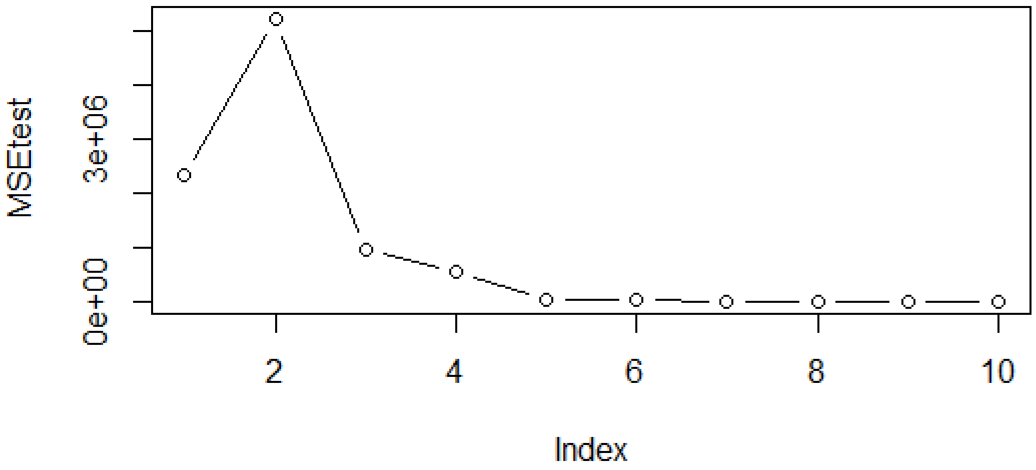


The training MSE starts high, then curves down until it reaches a plateau-like limit. The difference between model 1 and 2 is very large.



g.

Model 3 has the lowest test MSE. The test MSEs start low but increase drastically towards the end. Based on the bias-variance tradeoff, as the model becomes more fit or even overfitted, the test MSE increases as the variance is increasing.

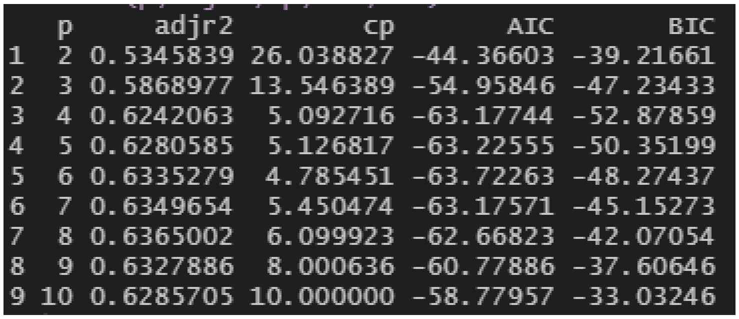


h.

This new test MSE curve decreases as the models become more complicated. The true model is complex so this is expected. As the fitted models become more complex, the test MSE decreases. There is an abnormal spike but this could be due to randomness within the data.

**Problem 2:**

a.

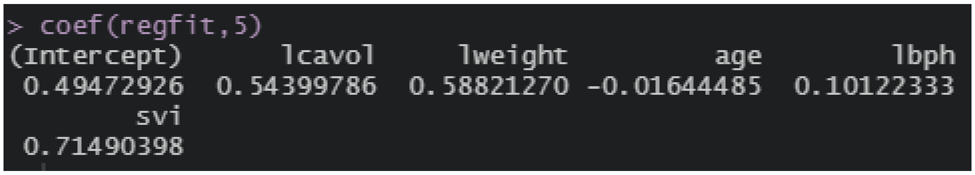


```
   p     adjr2        cp       AIC       BIC
1  2 0.5345839 26.038827 -44.36603 -39.21661
2  3 0.5868977 13.546389 -54.95846 -47.23433
3  4 0.6242063  5.092716 -63.17744 -52.87859
4  5 0.6280585  5.126817 -63.22555 -50.35199
5  6 0.6335279  4.785451 -63.72263 -48.27437
6  7 0.6349654  5.450474 -63.17571 -45.15273
7  8 0.6365002  6.099923 -62.66823 -42.07054
8  9 0.6327886  8.000636 -60.77886 -37.60646
9 10 0.6285705 10.000000 -58.77957 -33.03246
```

Smallest AIC: M5
Smallest BIC: M3
Largest adjusted R^2: M7
Smallest Mallow’s CP: M5

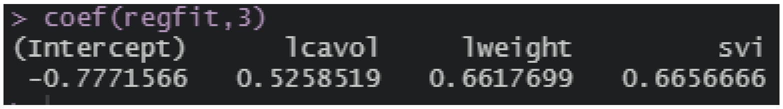
Yes, these lead to different models. I have chosen Model 5 as the best model because it has followed 2/4 of the criteria.

Model 5: lpsa = 0.49472926 + 0.54399786*lcavol + 0.58821270*lweight + -0.01644485*age + 0.10122333*lbph + 0.71490398*svi



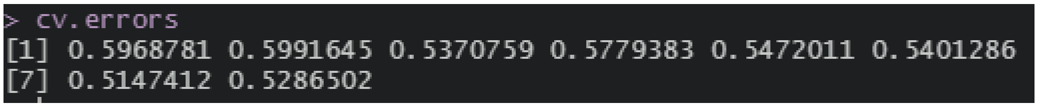
```
> coef(regfit,5)
(Intercept)      lcavol     lweight         age        lbph
 0.49472926  0.54399786  0.58821270 -0.01644485  0.10122333
        svi
 0.71490398
```

b. I have chosen Model 3 which has the lowest test MSE. The final model is
lpsa = -0.7771566 + 0.5258519*lcavol + 0.6617699*lweight + 0.6656666*svi



```
> coef(regfit,3)
(Intercept)      lcavol     lweight         svi
 -0.7771566   0.5258519   0.6617699   0.6656666
```

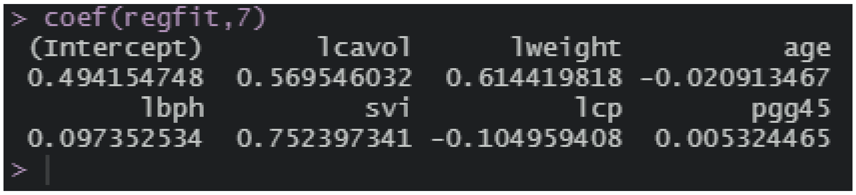
c. Part i)



```
> cv.errors
[1] 0.5968781 0.5991645 0.5370759 0.5779383 0.5472011 0.5401286
[7] 0.5147412 0.5286502
```

Model 7 has the smallest CV error.

Part ii)



```
> coef(regfit,7)
 (Intercept)       lcavol      lweight          age
 0.494154748  0.569546032  0.614419818 -0.020913467
        lbph          svi          lcp        pgg45
 0.097352534  0.752397341 -0.104959408  0.005324465
>
```

lpsa = 0.494154748 + 0.569546032*lcavol + 0.614419818*lweight + -0.020913467*age + 0.097352534*lbph + 0.752397341*svi + -0.104959408*lcp + 0.005324465*pgg45

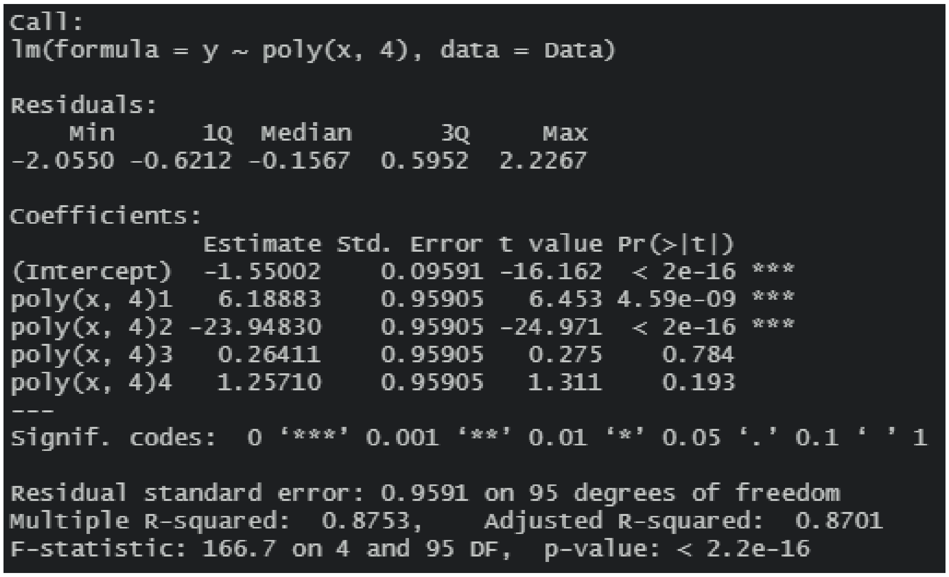
**Problem 3:**

a. The k-fold cross validation is implemented by taking the n number of observations and randomly splitting them into k non-overlapping groups. Each of these divided groups acts as a validation set and the rest is the training set. The test error is then estimated by averaging the k resulting MSE estimates.

b. i) disadvantages of the validation set approach relative to k-fold cross validation is that the estimate of the test error rate can be highly variable depending on which observations are included in the training/validation set. The validation set error rate may tend to overestimate the test error rate for the model fit on the entire data set.
ii) LOOCV is computationally intense since the model must be fit n times. But, LOOCV has higher variance and lower bias than k-fold.

c. error = rnorm(100,0,1)
y = x - 2 * x^2 + error

d. LOOCV error for M1 = 7.288162
   LOOCV error for M2 = 0.9374236
   LOOCV error for M3 = 0.9566218
   LOOCV error for M4 = 0.9539049
e. LOOCV error for M1 = 7.288162
   LOOCV error for M2 = 0.9374236
   LOOCV error for M3 = 0.9566218
   LOOCV error for M4 = 0.9539049
   The results are identical to the results from part d because LOOCV evaluates n folds of a single observation.
f. M2 has the lowest LOOCV error and this could be because the relation between x and y is quadratic in our given model of the simulated data set.



```
Call:
lm(formula = y ~ poly(x, 4), data = Data)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0550 -0.6212 -0.1567  0.5952  2.2267 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -1.55002    0.09591 -16.162  < 2e-16 ***
poly(x, 4)1   6.18883    0.95905   6.453 4.59e-09 ***
poly(x, 4)2 -23.94830    0.95905 -24.971  < 2e-16 ***
poly(x, 4)3   0.26411    0.95905   0.275    0.784    
poly(x, 4)4   1.25710    0.95905   1.311    0.193    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9591 on 95 degrees of freedom
Multiple R-squared:  0.8753,	Adjusted R-squared:  0.8701 
F-statistic: 166.7 on 4 and 95 DF,  p-value: < 2.2e-16
```

g. Looking at the p-values, the linear and quadratic terms are statistically significant. The cubic and 4th degree terms are not statistically significant. Thus, the results of the cross-validation agree with the conclusion as the minimum LOOCV error was for the quadratic model.

**Problem 4:**

a. True. AIC assumes that the true model is not in the candidate pool and tries to mimic it. BIC will eventually lead to a true model if it’s in the candidate pool and if n is large enough. Even then, all the criteria use a different formula and weigh the factors differently. AIC, BIC, adjusted R^2 and Mallow’s CP can lead to different final models, but it’s also up to the data scientist’s understanding of the data as well.
b. False. Model 4 and Model 3 share the predictor X4. However M3 is not nested in M4. If there are trash predictors in M3 and good predictors in M4, then the $RSS_{M3} \geq RSS_{M4}$ holds. If there are trash predictors in M4 and good predictors in M3, then the $RSS_{M4} \geq RSS_{M3}$.
c. True. M2 is nested in M4, so adding predictors to M4 means it can do no worse than M2. So $RSS_{M2} \geq RSS_{M4}$.