```
> #HW12
> if (FALSE)
+ {"
+ Use 5-fold cross validation to decide the degree of polynomial to use for a regression of speed on distance needed to stop
from the cars data set.  See details below and the Rcode you will need.  This R program, HR12.R is also loaded into Canvas
under Files Loaded after Spring Break.  Just hand in the text output with a sentence explaining your choice of degree - no
graphics files are necessary.
+ cars is a data frame with 50 observations on 2 variables
+ speed - speed(mph)
+ dist - stopping distance measured in feet
+ "}
> ##-------------------------------------------------------------##
> #Program to help you
> library(faraway)
> library(caret) #install this package if needed
>
> set.seed(13245) #use this seed
> head(cars,1L)
  speed dist
1     4    2
> attach(cars) #n=50
The following objects are masked from cars (pos = 3):

    dist, speed

The following objects are masked from cars (pos = 4):

    dist, speed

> # sorting dataset by distance for graphing purposes
> cars <- cars[order(dist),]
> cars
   speed dist
1      4    2
3      7    4
2      4   10
6      9   10
12    12   14
5      8   16
10    11   17
7     10   18
13    12   20
24    15   20
4      7   22
14    12   24
8     10   26
16    13   26
20    14   26
25    15   26
11    11   28
```

Degree	RMSE	R2	MAE
1	3.190934	0.647672	2.627375
2	3.03327	0.66217	2.468283
3	3.052608	0.686928	2.541086
4	4.399993	0.673987	3.032101

```
15    12   28
27    16   32
29    17   32
39    20   32
9     10   34
17    13   34
18    13   34
21    14   36
36    19   36
28    16   40
30    17   40
32    18   42
19    13   46
37    19   46
40    20   48
31    17   50
41    20   52
26    15   54
45    23   54
33    18   56
42    20   56
22    14   60
43    20   64
44    22   66
38    19   68
46    24   70
34    18   76
23    14   80
35    18   84
50    25   85
47    24   92
48    24   93
49    24  120
>
> windows(7,7)
> #save graph(s) in pdf
> pdf(file="C:/Users/jmard/OneDrive/Desktop/Computing and Graphics in Applied Statistics2020/Homework/HW12_Figures.pdf")
> plot(x=cars$dist,y=cars$speed)
>
> ##-------------------------------------------------------------##
> #The researcher is interested in predicting speed based on knowing stopping distance
> #fit a polynomial to the data - use degree 1, 2, 3, or 4?
> #use cross-validation since overfitting is a concern
> #ASSIGNMENT: use 5-fold cross validation to obtain the choice of degree 1, 2, 3, or 4
>
> #Here is the r-code for a polynomial of degree 4 and plotting the fitted curve
> #You can use the code below and just repeat for a polynomials of degree 1, 2, and 3
> #Fit a polynomial of degree 4
> poly4<- lm(speed~dist+I(dist^2)+I(dist^3)+I(dist^4), data=cars)
> summary(poly4) #summary of results from fitting a polynomial of degree 4
```

```
Call:
lm(formula = speed ~ dist + I(dist^2) + I(dist^3) + I(dist^4),
    data = cars)

Residuals:
    Min      1Q  Median      3Q     Max
-6.8557 -1.9194  0.2788  2.0023  5.5300

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.430e+00  2.364e+00   1.451    0.154
dist         4.806e-01  2.521e-01   1.906    0.063 .
I(dist^2)   -4.909e-03  8.633e-03  -0.569    0.572
I(dist^3)    2.151e-05  1.109e-04   0.194    0.847
I(dist^4)   -1.494e-08  4.657e-07  -0.032    0.975
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.917 on 45 degrees of freedom
Multiple R-squared:  0.7206,    Adjusted R-squared:  0.6957
F-statistic: 29.01 on 4 and 45 DF,  p-value: 5.984e-12

> plot(x=cars$dist,y=cars$speed)
> lines(x=cars$dist,y=poly4$fitted, type="l", col="red")
> #Compute the cross-validation metrics for degree 4
> # Define training control
> train.control <- trainControl(method = "cv", number = 5)
> # Train the model
> CVpoly4 <- train(speed~dist+I(dist^2)+I(dist^3)+I(dist^4),data = cars, method = "lm",
+ trControl = train.control)
> # Summarize the results
> print(CVpoly4)
Linear Regression

50 samples
 1 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 41, 40, 39, 40, 40
Resampling results:

  RMSE      Rsquared   MAE
  4.399993  0.6739873  3.032101

Tuning parameter 'intercept' was held constant at a value of TRUE
> ##
>
> #Fit a polynomial of degree 3
```

```
> poly3<- lm(speed~dist+I(dist^2)+I(dist^3), data=cars)
> summary(poly3) #summary of results from fitting a polynomial of degree 3

Call:
lm(formula = speed ~ dist + I(dist^2) + I(dist^3), data = cars)

Residuals:
   Min     1Q Median     3Q    Max 
-6.846 -1.917  0.278  2.006  5.535 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.478e+00  1.809e+00   1.923 0.060705 .  
dist         4.736e-01  1.241e-01   3.817 0.000402 ***
I(dist^2)   -4.644e-03  2.428e-03  -1.913 0.062046 .  
I(dist^3)    1.798e-05  1.372e-05   1.310 0.196685    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.885 on 46 degrees of freedom
Multiple R-squared:  0.7206,    Adjusted R-squared:  0.7023 
F-statistic: 39.54 on 3 and 46 DF,  p-value: 8.652e-13

> plot(x=cars$dist,y=cars$speed)
> lines(x=cars$dist,y=poly3$fitted, type="l", col="red")
> #Compute the cross-validation metrics for degree 3
> # Define training control
> train.control <- trainControl(method = "cv", number = 5)
> # Train the model
> CVpoly3 <- train(speed~dist+I(dist^2)+I(dist^3),data = cars, method = "lm",
+ trControl = train.control)
> # Summarize the results
> print(CVpoly3)
Linear Regression 

50 samples
 1 predictor

No pre-processing
Resampling: Cross-Validated (5 fold) 
Summary of sample sizes: 40, 41, 38, 41, 40 
Resampling results:

  RMSE      Rsquared   MAE     
  3.052608  0.6869276  2.541086

Tuning parameter 'intercept' was held constant at a value of TRUE
> ##
> 
> #Fit a polynomial of degree 2
```

```
> poly2<- lm(speed~dist+I(dist^2), data=cars)
> summary(poly2) #summary of results from fitting a polynomial of degree 2

Call:
lm(formula = speed ~ dist + I(dist^2), data = cars)

Residuals:
   Min     1Q Median     3Q    Max 
-7.559 -1.722  0.473  1.932  5.942 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.1439610  1.2954573   3.971 0.000244 ***
dist         0.3274544  0.0547392   5.982 2.86e-07 ***
I(dist^2)   -0.0015284  0.0004939  -3.095 0.003316 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.907 on 47 degrees of freedom
Multiple R-squared:  0.7101,    Adjusted R-squared:  0.6978 
F-statistic: 57.57 on 2 and 47 DF,  p-value: 2.299e-13

> plot(x=cars$dist,y=cars$speed)
> lines(x=cars$dist,y=poly2$fitted, type="l", col="red")
> #Compute the cross-validation metrics for degree 2
> # Define training control
> train.control <- trainControl(method = "cv", number = 5)
> # Train the model
> CVpoly2 <- train(speed~dist+I(dist^2),data = cars, method = "lm",
+ trControl = train.control)
> # Summarize the results
> print(CVpoly2)
Linear Regression 

50 samples
 1 predictor

No pre-processing
Resampling: Cross-Validated (5 fold) 
Summary of sample sizes: 41, 40, 40, 40, 39 
Resampling results:

  RMSE     Rsquared  MAE     
  3.03327  0.66217   2.468283

Tuning parameter 'intercept' was held constant at a value of TRUE
> ##
> 
> #Fit a polynomial of degree 1 - straight line model
> poly1<- lm(speed~dist, data=cars)
```

```
> summary(poly1) #summary of results from fitting a polynomial of degree 1

Call:
lm(formula = speed ~ dist, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.5293 -2.1550  0.3615  2.4377  6.4179 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.28391    0.87438   9.474 1.44e-12 ***
dist         0.16557    0.01749   9.464 1.49e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.156 on 48 degrees of freedom
Multiple R-squared:  0.6511,    Adjusted R-squared:  0.6438 
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12

> plot(x=cars$dist,y=cars$speed)
> lines(x=cars$dist,y=poly1$fitted, type="l", col="red")
> #Compute the cross-validation metrics for degree 1
> # Define training control
> train.control <- trainControl(method = "cv", number = 5)
> # Train the model
> CVpoly1 <- train(speed~dist,data = cars, method = "lm",
+ trControl = train.control)
> # Summarize the results
> print(CVpoly1)
Linear Regression 

50 samples
 1 predictor

No pre-processing
Resampling: Cross-Validated (5 fold) 
Summary of sample sizes: 40, 41, 40, 40, 39 
Resampling results:

  RMSE      Rsquared  MAE     
  3.190934  0.647672  2.627375

Tuning parameter 'intercept' was held constant at a value of TRUE
> ##
> 
> 
> ##-------------------------------------------------------------##
> dev.off()
windows 
```

2

>