STAT 206 Lab 8

Xin Feng(Vanessa)

11/23/2019

Due Monday, November 25, 5:00 PM

General instructions for labs: Labs must be completed as a pdf file. Give the commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file. Each answer must be supported by written statements as well as any code used.

Agenda: Fit polynomial regression models to the electricity usage data, use K-fold cross-validation to automatically select degree of the polynomial

Polynomial regression

The polynomial regression model posits that a response variable Y and explanatory variable X are related by the equation.

$$Y = \sum_{j=0}^{d} \beta_j X^j + \epsilon \, .$$

The number d is called the degree of the polynomial. Polynomial regression reduces to linear regression when $d = 1$. Its flexibility and complexity increase as d increases. The cases $d = 2$ and $d = 3$ are usually referred to as quadratic and cubic. The polynomial regression model can be expressed as a $d + 1$ parameter linear model by considering $(X_0, X_1, X_2, \ldots, X_d)$ as explanatory variables. This is done by `poly()` and can be combined with `lm()` to fit a polynomial regression model. In the following example, we fit a degree-3 polynomial, or cubic, regression model using variables y and x in the dataframe `df`.

```
degree <- 3
obj <- lm(y ~ poly(x, degree), data = df)
```

'electemp' dataset

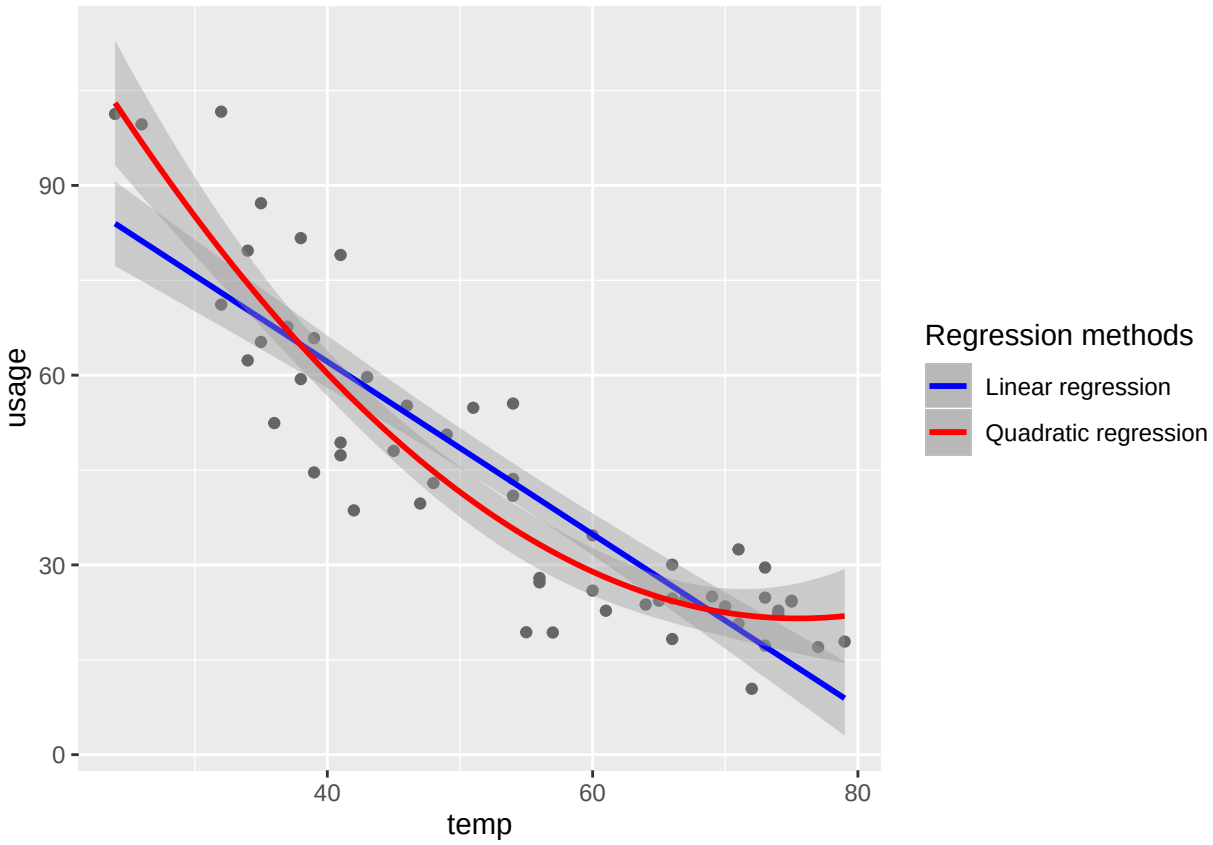
The 'electemp' dataset has 55 observations on monthly electricity usage and average temperature for a house in Westchester County, New York

```
url <- 'http://www.faculty.ucr.edu/~jflegal/electemp.txt'
electemp <- read.table(url)
```

1. Create a scatterplot of `temp` and `usage` with `ggplot2` that includes the least squares fits of a linear and quadratic regression models. You should also include a legend on the plot.

```
library(ggplot2)
# summary(electemp)
# View(electemp)
x <- electemp$temp
y <- electemp$usage
p <- ggplot(data=electemp, aes(temp, usage)) + geom_point(color="grey40")
```

```
p1 <- geom_smooth(method='lm', formula=y~x, aes(color="Linear regression"))
p2 <- geom_smooth(method='lm', formula=y~poly(x,2),
                  aes(color="Quadratic regression"))
p3 <- scale_color_manual(name="Regression methods", values=c("blue", "red"))
p+p1+p2+p3
```



2. Does the linear or quadratic model fit the data better?

```
# According the picture, I think the quadratic regression model fit the data better.
fit1 <- lm(y~x)
fit2 <- lm(y~poly(x,2))
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.305  -8.163   0.559   7.723  28.611
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 116.71619    5.56494   20.97   <2e-16 ***
## x            -1.36461    0.09941  -13.73   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.35 on 53 degrees of freedom
## Multiple R-squared:  0.7805, Adjusted R-squared:  0.7763
## F-statistic: 188.4 on 1 and 53 DF,  p-value: < 2.2e-16
```

```
summary(fit2)
```

```
##
## Call:
## lm(formula = y ~ poly(x, 2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.8593  -5.6813   0.3756   5.4650  21.9701
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   43.275      1.291  33.522  < 2e-16 ***
## poly(x, 2)1 -155.866      9.574 -16.280  < 2e-16 ***
## poly(x, 2)2   45.464      9.574   4.749 1.65e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.574 on 52 degrees of freedom
## Multiple R-squared:  0.8469, Adjusted R-squared:  0.841
## F-statistic: 143.8 on 2 and 52 DF,  p-value: < 2.2e-16
```

```
# R square value in Model II greater than Model I, therefore, the quadratic regression
# model is better.
```

3. Write a function `cv_poly()` that performs K-fold cross-validation to estimate the mean squared prediction error (MSPE) of polynomial regression. It takes vectors x and y containing observations of the explanatory and response variables, a vector degree of the degrees of polynomial models to fit, and a number K indicating the number of folds for cross-validation. It returns a $K \times D$ matrix, where K is the number of folds and D is the number of different degree models that are being fit. The entries of the matrix are the MSPE for each fold and degree polynomial model being fit.

```
library(stats)
mspe_func <- function(test_data, train_data, degree){
  x <- train_data$temp
  y <- train_data$usage
  fit <- lm(y~poly(x, degree))
  mspe <- mean((test_data$usage - predict(fit, data.frame(x=test_data$temp),
                                          interval="prediction"))^2)
  return(mspe)
}
mspe_func(electemp[1:5,],electemp[-1:-5,],2)
```

```
## [1] 330.4533
```

```r
cv_ploy <- function(X, Y, degrees, K){
  D <- length(degrees)
  N <- length(y)
  mspe <- matrix(nrow=K, ncol=D)
  index <- append(seq(1, N, N%/%K),N)
  if(K==1){
    return(FALSE)
  }
  for(i in 1:K){
    test_data <- data.frame(temp=X[index[i]:index[i+1]],usage=y[index[i]:index[i+1]])
    train_data <- data.frame(temp=X[-index[i]:-index[i+1]], usage=Y[-index[i]:-index[i+1]])
    mspe[i,] <- sapply(degrees,mspe_func,test_data=test_data,train_data=train_data)
  }
  rownames(mspe) <- paste("K=",1:K)
  colnames(mspe) <- paste("D=",1:D)
  return(mspe)
}
# cv_ploy(x,y,c(1,2,3),5)
```

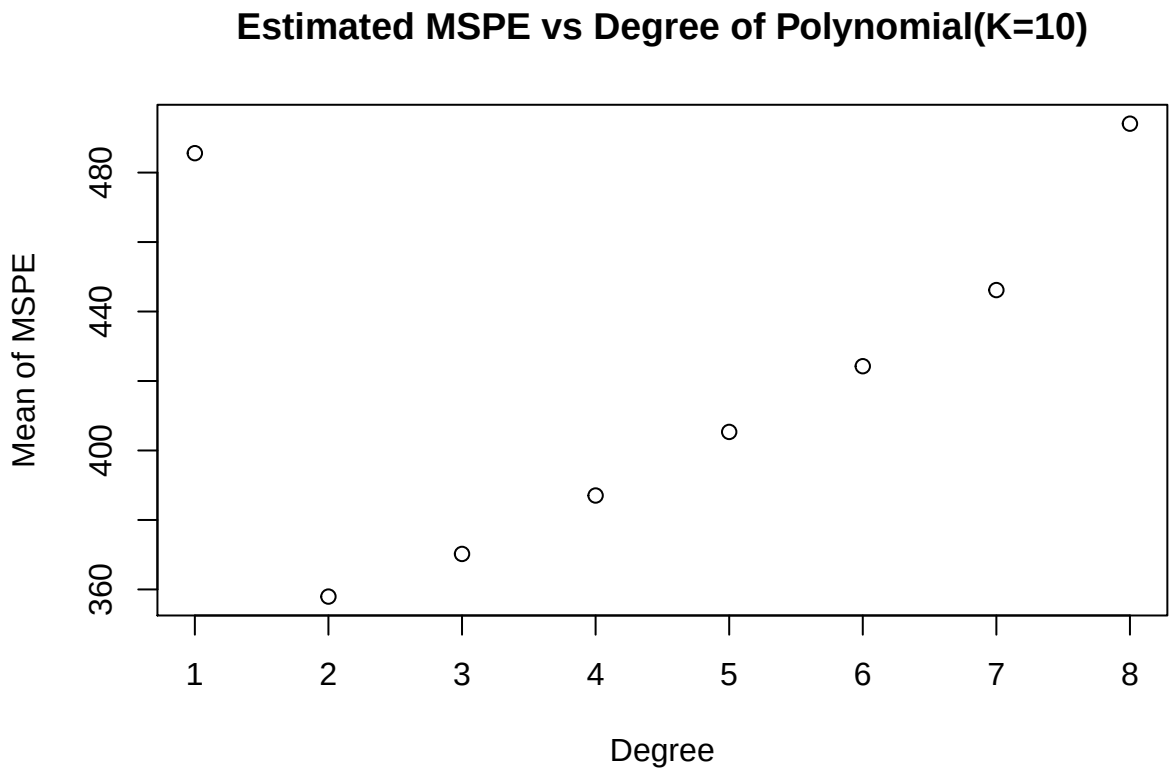
4. Use `cv_poly()` to estimate the MSPE of polynomial regression on the electricity usage data by $K = 10$-fold cross-validation for $d = 1, 2, \ldots, 8$. Note that `cv_poly()` should return a matrix, call it `cv_error` with K rows corresponding to the K different validation sets.

```r
cv_error <- cv_ploy(x,y,1:8,10)
cv_error
```

```
##           D= 1     D= 2     D= 3     D= 4     D= 5     D= 6     D= 7
## K= 1  510.0480 336.0236 352.3436 363.2993 375.6440 384.1873 434.1448
## K= 2  449.0503 354.7674 362.9353 379.5203 391.5261 382.8805 390.0512
## K= 3  440.2172 306.7165 318.0286 329.3274 341.1198 339.9319 348.0664
## K= 4  560.4133 419.1204 428.7571 437.9073 449.1311 447.1916 453.3499
## K= 5  445.0084 316.3668 328.2978 341.2822 351.8650 357.1007 365.0274
## K= 6  496.9553 345.2328 361.9651 377.0986 387.5467 396.1900 419.8860
## K= 7  436.3103 336.2360 344.6037 355.9013 365.5278 360.7220 369.6830
## K= 8  489.8844 365.9372 375.2708 385.9300 396.0813 391.0578 394.0609
## K= 9  584.2882 470.4722 479.4105 507.6763 540.0431 565.2875 566.0518
## K= 10 443.6740 328.5723 350.5345 392.3764 454.9589 618.0285 721.4314
##           D= 8
## K= 1  579.2443
## K= 2  397.0741
## K= 3  361.2547
## K= 4  458.6127
## K= 5  379.7506
## K= 6  424.3059
## K= 7  379.4651
## K= 8  398.7198
## K= 9  600.9630
## K= 10 961.3316
```

5. Plot the estimated MSPE (by averaging across the K folds) versus degree of the polynomial. What degree polynomial would you select according to cross-validation?

```
mspe_degree <- colMeans(cv_error)
plot(mspe_degree, xlab="Degree", ylab="Mean of MSPE",
     main="Estimated MSPE vs Degree of Polynomial(K=10)")
```




```
# I select degree=2, which has a least MSPE.
```

6. Repeat the preceding problem for $K = 5$ and leave-one-out cross-validation ($K = n$). What do you notice about the time it takes to compute the cross-validation? How do the results change with K?

```
t1 <- Sys.time()
cv_error_2 <- cv_ploy(x,y,1:8,10)
Sys.time()-t1
```

```
## Time difference of 0.126087 secs
```

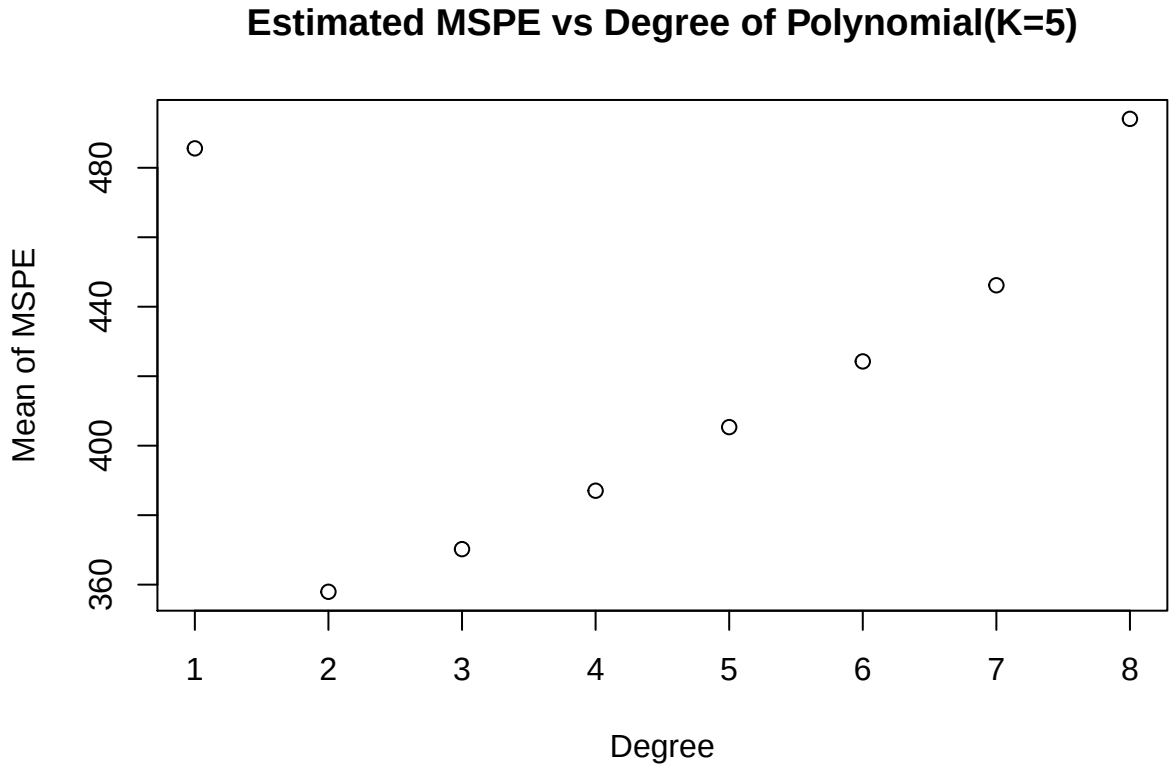
```
t2 <- Sys.time()
cv_error_3 <- cv_ploy(x,y,1:8,length(x))
Sys.time()-t2
```

```
## Time difference of 0.6035891 secs
```

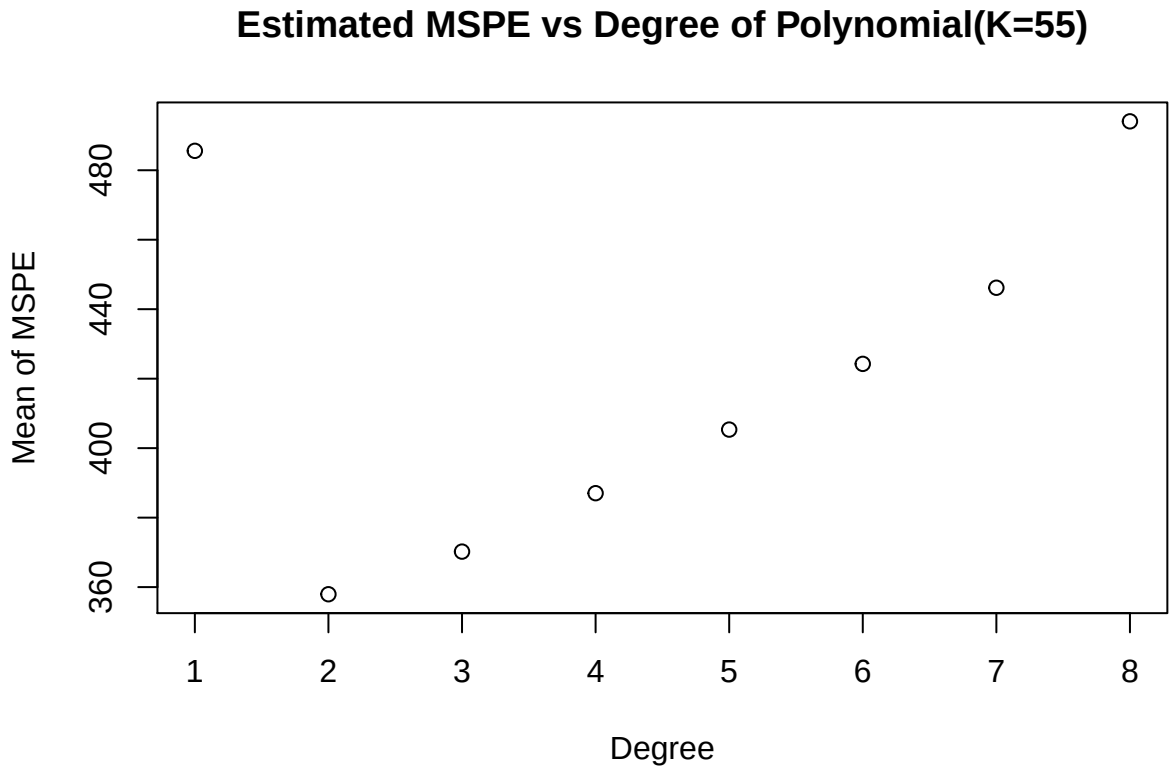
```
# K=n takes much more time than K=5(n=55>5)
# The greater of K, the more time it takes.
```

7. Plot the estimated MSPE versus degree of the polynomial. What degree polynomial would you select according to cross-validation? Are there differences between $K = 5$, $K = 10$, and leave-one-out estimates of MSPE?

```r
mspe_degree_2 <- colMeans(cv_error_2)
plot(mspe_degree, xlab="Degree", ylab="Mean of MSPE",
     main="Estimated MSPE vs Degree of Polynomial(K=5)")
```




```r
# Select degree=2, which has a least MSPE.
mspe_degree_3 <- colMeans(cv_error_3)
plot(mspe_degree, xlab="Degree", ylab="Mean of MSPE",
     main="Estimated MSPE vs Degree of Polynomial(K=55)")
```




```
# Select degree=2, which has a least MSPE.
# The results are the same whatever K is.
mspe_degree_2
```

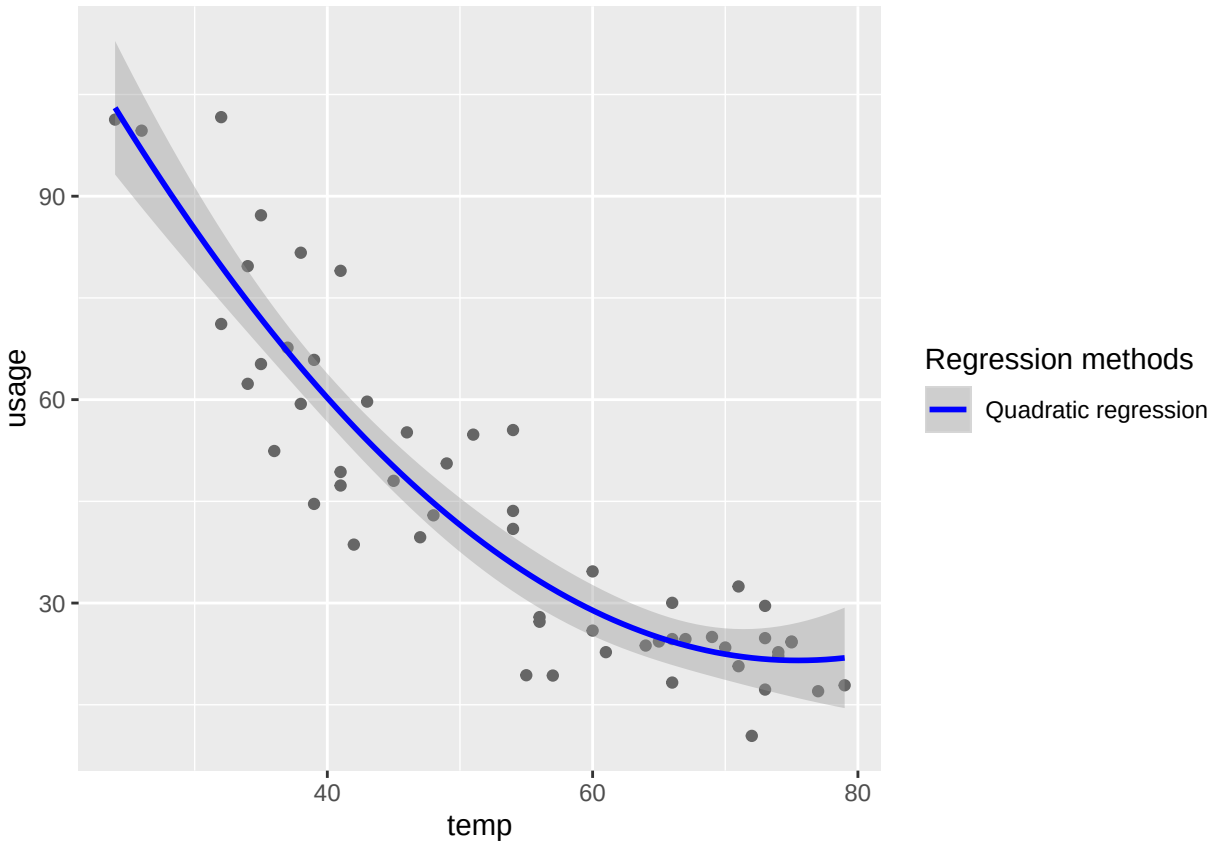
```
##     D= 1     D= 2     D= 3     D= 4     D= 5     D= 6     D= 7     D= 8
## 485.5849 357.9445 370.2147 387.0319 405.3444 424.2578 446.1753 494.0722
```

```
mspe_degree_3
```

```
##     D= 1     D= 2     D= 3     D= 4     D= 5     D= 6     D= 7     D= 8
## 506.3399 365.7356 380.2428 399.6180 426.4269 447.5321 497.0503 647.1114
```

8. Reproduce your first plot and add a layer showing the polynomial regression model selected by cross-validation by modifying the following code.

```
p + p2 + p3
```


usage
90
60
30
40
60
80
temp
Regression methods
Quadratic regression