# Analytics 512 Homework 6


## Arif Ali


# Exercise 2

## Part A

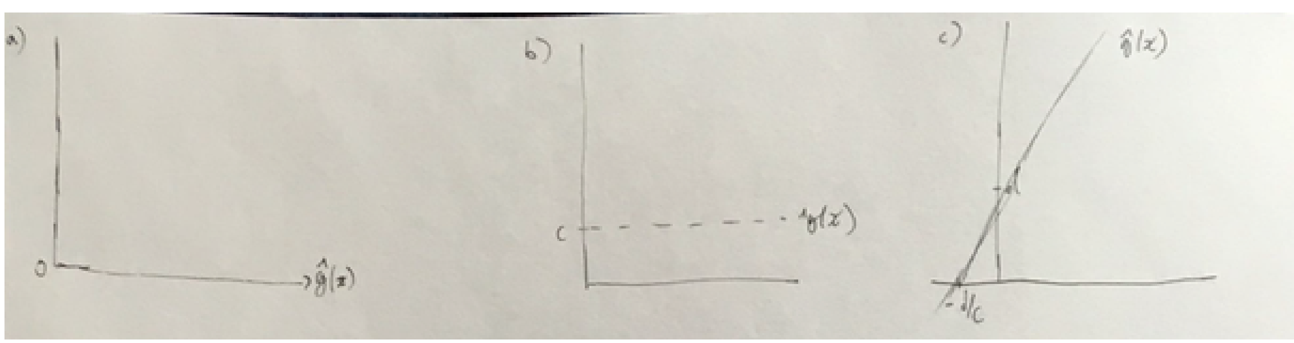
In order for the arg min to be achieved, $\lambda$ would have to be minimized since it is $\infty$, therefore, $g(x) = 0$

## Part B

by the logic in part A, $g'(x) = 0$, so $g(x) = \int g'(x)dx = c$ where c is a constant

## Part C

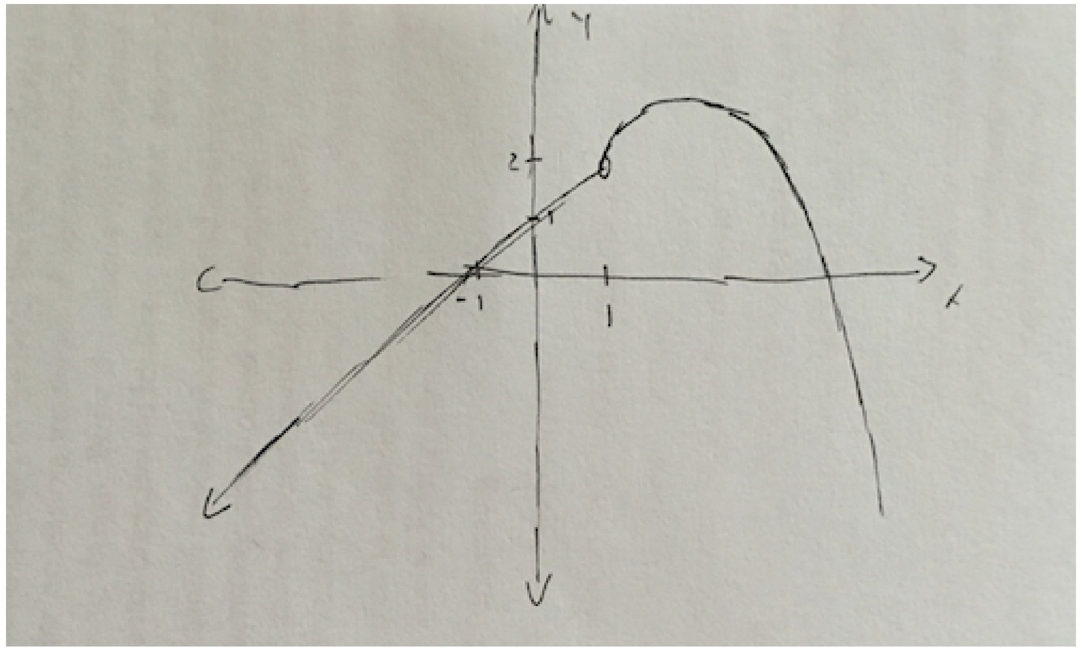
$g''(x) = 0 \implies g'(x) = c \implies g(x) = c * x + d$ where c and d are both constants



# Exercise 3

$$Y = 1 + X - 2(X-1)^2 = 1 + X - 2(X^2 - 2X + 1) = -2X^2 + 5X - 1 \; when \; X \geq 1$$
$$Y = 1 + X \; o.w.$$

Intercepts at (0,1), (-1,0), ($\sqrt{17}/4 + 5/4$,0)




# Exercise 6

## Part A

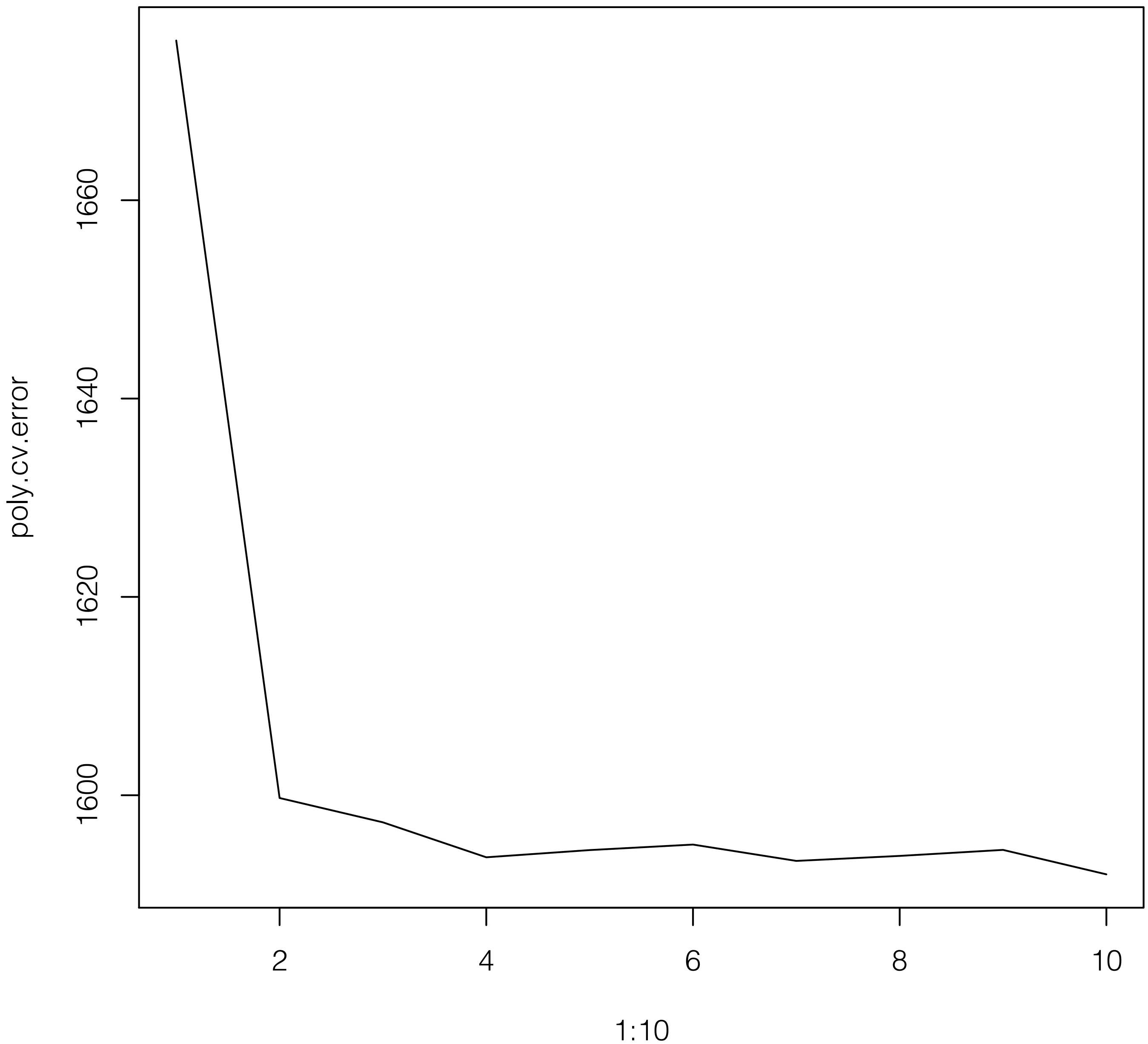
In [1]:
```r
d = 10
library(ISLR)
library(boot)
poly.cv.error = c()
different.d = as.list(1:10)
for(i in 1:d){
    wage.pm = glm(wage~poly(age, i), data = Wage)
    poly.cv.error[i] = cv.glm(Wage, wage.pm, K = 10)$delta[2]
    different.d[[i]] = lm(wage~poly(age, i), data = Wage)
    }
poly.cv.error
which.min(poly.cv.error)
```

Out[1]:

1676.10420087314 1599.72101804354 1597.26734459586 1593.74316802434 1594.46944707439
1595.02556440882 1593.37772701369 1593.88216375312 1594.4884242901 1592.02258842504

Out[1]: 10

In [2]:
```
plot(1:10,poly.cv.error, type = "l")
```



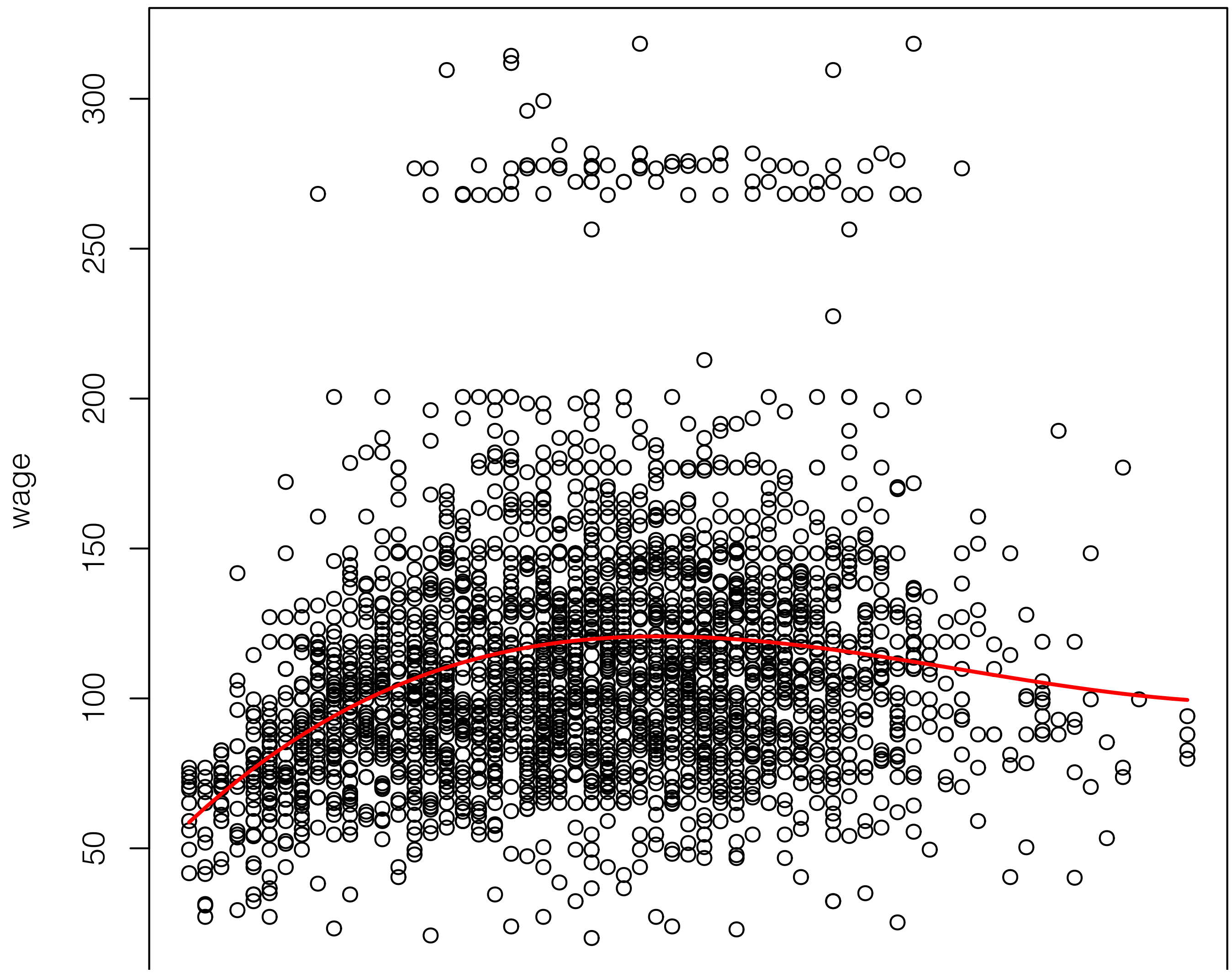
In [3]:
```
print(anova(different.d[[1]], different.d[[2]],different.d[[3]],different.d[[4]],different.d[[5]],
     different.d[[6]], different.d[[7]],different.d[[8]],different.d[[9]],different.d[[10]]))
```

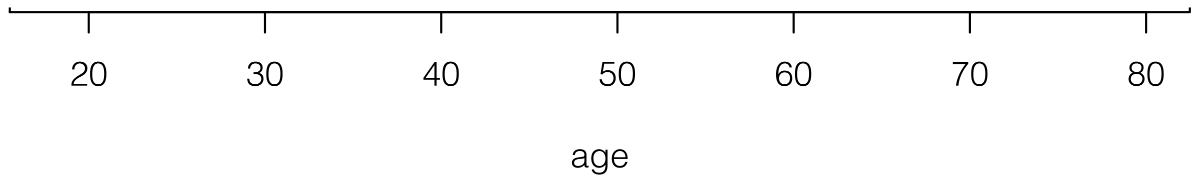
```
Analysis of Variance Table

Model  1: wage ~ poly(age, i)
Model  2: wage ~ poly(age, i)
Model  3: wage ~ poly(age, i)
Model  4: wage ~ poly(age, i)
Model  5: wage ~ poly(age, i)
Model  6: wage ~ poly(age, i)
Model  7: wage ~ poly(age, i)
Model  8: wage ~ poly(age, i)
Model  9: wage ~ poly(age, i)
Model 10: wage ~ poly(age, i)
   Res.Df     RSS Df Sum of Sq        F    Pr(>F)
1    2998 5022216
2    2997 4793430  1    228786 143.7638 < 2.2e-16 ***
3    2996 4777674  1     15756   9.9005  0.001669 **
4    2995 4771604  1      6070   3.8143  0.050909 .
5    2994 4770322  1      1283   0.8059  0.369398
6    2993 4766389  1      3932   2.4709  0.116074
7    2992 4763834  1      2555   1.6057  0.205199
8    2991 4763707  1       127   0.0796  0.777865
9    2990 4756703  1      7004   4.4014  0.035994 *
10   2989 4756701  1         3   0.0017  0.967529
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Via the cross validated method, a polynomial with the degree of 9 was chosen. Under the ANOVA test, the 9th degree polynomial model is considered significiant; however, not as significant as the second or third degree polynomial, so the third degree is probably the most optimal

In [4]:
```
plot(wage ~ age, data = Wage)
agelims = range(Wage$age)
age.grid = seq(from = agelims[1], to = agelims[2])
wage.d3m = lm(wage ~ poly(age, 3), data = Wage)
preds = predict(wage.d3m, newdata = list(age = age.grid))
lines(age.grid, preds, col = "red", lwd = 2)
```



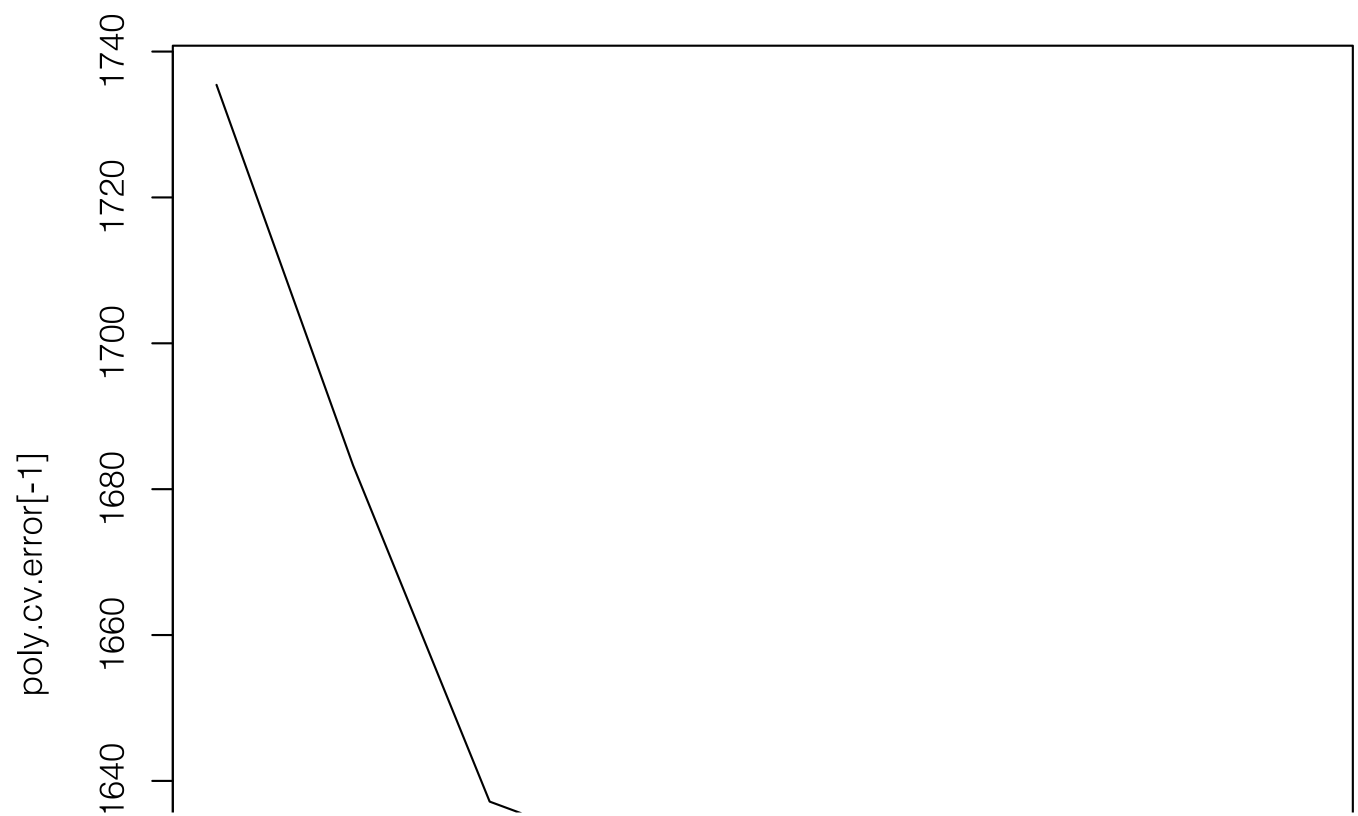


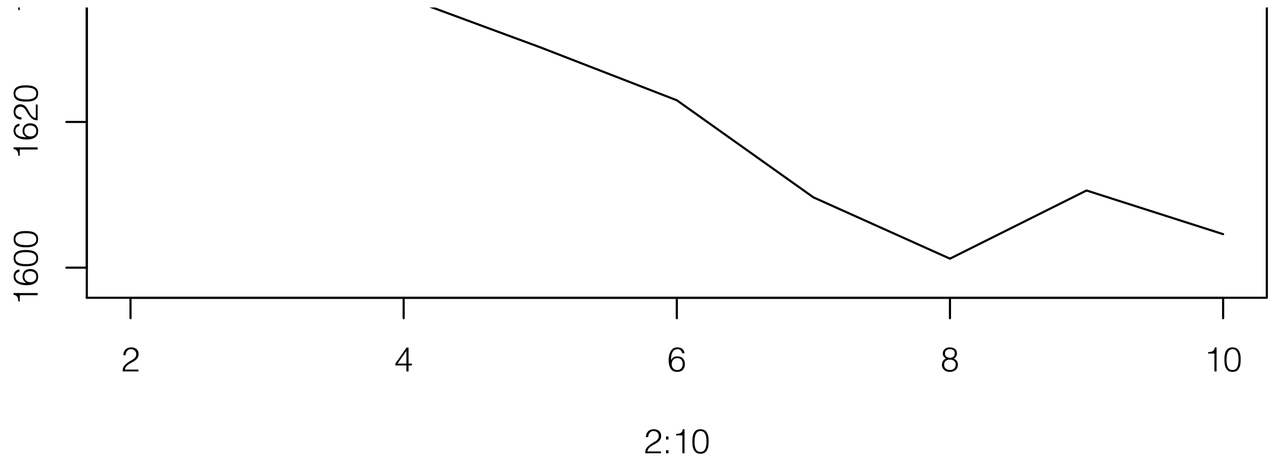

**Part B**

In [5]:
```
d = 10
for(i in 2:d){
    Wage$age_cuts = cut(Wage$age, i)
    wage.pm = glm(wage~age_cuts, data = Wage)
    poly.cv.error[i] = cv.glm(Wage, wage.pm, K = 10)$delta[2]
    }
poly.cv.error[-1]
which.min(poly.cv.error[-1])
plot(2:10,poly.cv.error[-1], type = "l")
```

Out[5]: 1735.43659534234 1683.26493861114 1637.16324930901 1630.24366180349 1622.97659710992 1609.65316531392 1601.23375699219 1610.59146799314 1604.59775913335

Out[5]: 7






Eight cuts results in the lowest MSE from a cross-validated model.

# Exercise 9

## Part A

In [6]:
```
library(MASS)
```

In [7]:
```
nox.nlm = lm(nox~poly(dis,3),data = Boston)
summary(nox.nlm)
```

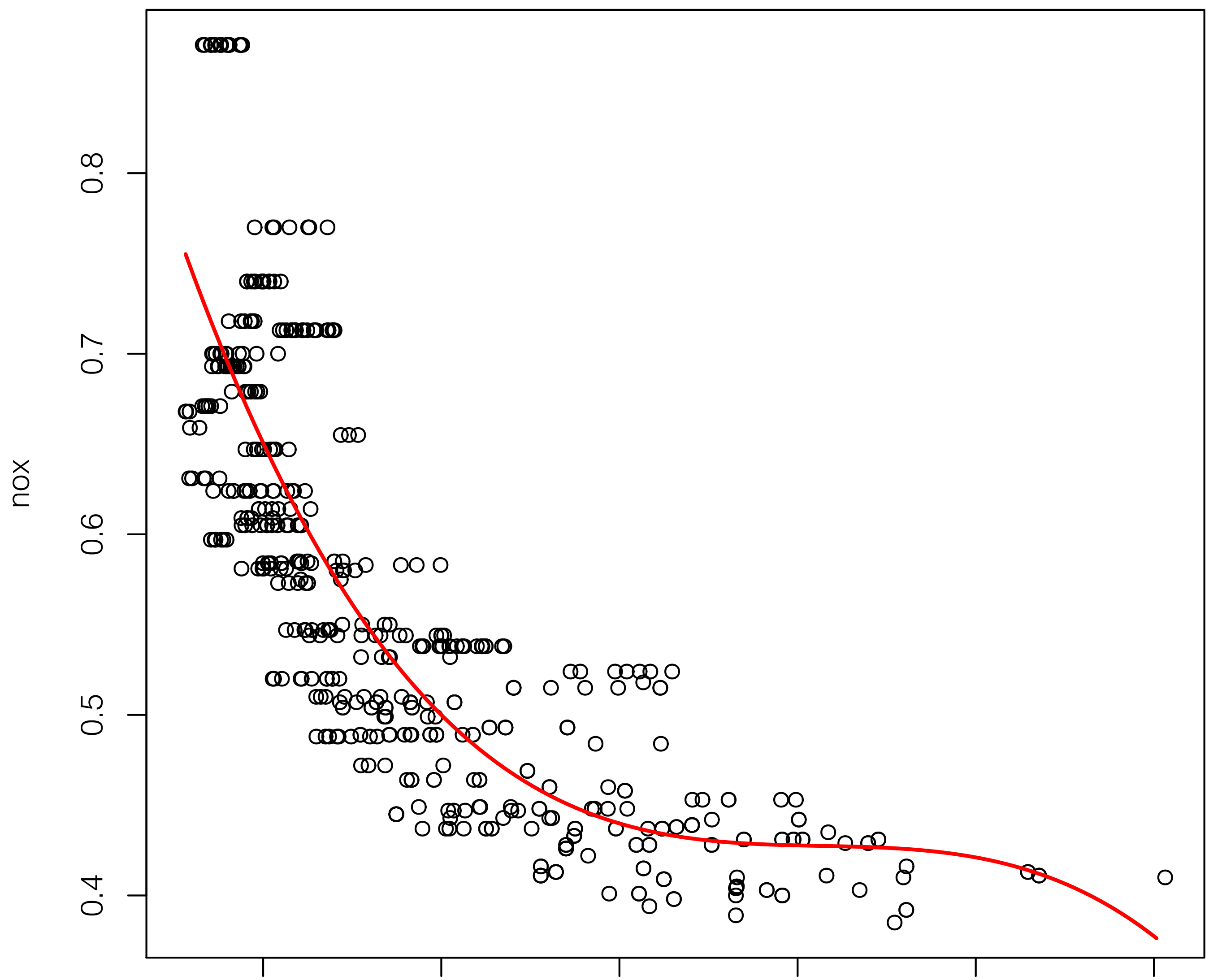
Out[7]:
```
Call:
lm(formula = nox ~ poly(dis, 3), data = Boston)

Residuals:
      Min        1Q    Median        3Q       Max
-0.121130 -0.040619 -0.009738  0.023385  0.194904

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.554695   0.002759 201.021  < 2e-16 ***
poly(dis, 3)1 -2.003096   0.062071 -32.271  < 2e-16 ***
poly(dis, 3)2  0.856330   0.062071  13.796  < 2e-16 ***
poly(dis, 3)3 -0.318049   0.062071  -5.124 4.27e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06207 on 502 degrees of freedom
Multiple R-squared:  0.7148,    Adjusted R-squared:  0.7131
F-statistic: 419.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

In [8]:
```
dislims = range(Boston$dis)
dis.grid = seq(from = dislims[1], to = dislims[2], by = 0.1)
preds = predict(nox.nlm, list(dis = dis.grid))
plot(nox ~ dis, data = Boston)
lines(dis.grid, preds, col = "red", lwd = 2)
```



2 4 6 8 10 12

dis

### Part B

In [9]:
```
for(i in 1:10){
    nox.nlm = lm(nox~poly(dis,i),data = Boston)
    statement = paste("For a polynomial of", i, "The RSS is",sum(nox.nlm$residuals^2), sep = " ")
    print(statement)
    }
```

```
[1] "For a polynomial of 1 The RSS is 2.76856285896928"
[1] "For a polynomial of 2 The RSS is 2.03526186893526"
[1] "For a polynomial of 3 The RSS is 1.93410670717907"
[1] "For a polynomial of 4 The RSS is 1.93298132729859"
[1] "For a polynomial of 5 The RSS is 1.9152899610843"
[1] "For a polynomial of 6 The RSS is 1.87825729850816"
[1] "For a polynomial of 7 The RSS is 1.84948361458298"
[1] "For a polynomial of 8 The RSS is 1.83562968906769"
[1] "For a polynomial of 9 The RSS is 1.8333308044916"
[1] "For a polynomial of 10 The RSS is 1.83217112393138"
```

## Exercise 11

### Part A

In [10]:
```
esp = rnorm(100)
x1 = rnorm(100)
x2 = rnorm(100)
y = x1 + 2*x2 + esp
```

### Part B

In [11]:
```
beta1 = 14
```

### Part C

In [12]:
```
a=y-beta1*x1
beta2=lm(a~x2)$coef[2]
```

### Part D

In [13]:
```
a=y-beta2*x2
beta1=lm(a~x1)$coef[2]
beta0=lm(a~x1)$coef[1]
```

### Part E

In [14]:
```
beta1s = rep(beta1,1000)
beta2s = rep(beta2,1000)
beta0s = rep(beta0,1000)
for(i in 2:1000){
    a=y-beta1*x1
    beta2=lm(a~x2)$coef[2]
    beta2s[i] = beta2
    a=y-beta2*x2
    beta1=lm(a~x1)$coef[2]
    beta1s[i] = beta1
    beta0=lm(a~x1)$coef[1]
    beta0s[i] = beta0
    }
```

In [15]:
```
summary(beta1s)
unique(beta1s)
```

Out[15]:
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.9279  0.9279  0.9279  0.9279  0.9279  0.9289 
```

Out[15]: 0.928854928102986   0.92789650362555   0.927896433350339   0.927896433345186   0.927896433345186

In [16]:
```
summary(beta2s)
unique(beta2s)
```

Out[16]:
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.967   2.078   2.078   2.078   2.078   2.078 
```

Out[16]: 1.96730839472973   2.07797035256653   2.07797846670868   2.07797846730364   2.07797846730368

In [17]:
```
summary(beta0s)
unique(beta0s)
```

Out[17]:
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
0.05372 0.08331 0.08331 0.08328 0.08331 0.08331 
```

Out[17]: 0.0537164601650667   0.08330699333951   0.0833091630263381   0.0833091631854278   0.0833091631854391

## Part F

In [18]:
```
master.nlm = lm(y~x1+x2)
summary(master.nlm)
```

Out[18]:
```
Call:
lm(formula = y ~ x1 + x2)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0989 -0.7849 -0.0257  0.6801  2.7337 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.08331    0.11360   0.733    0.465    
x1           0.92790    0.10638   8.723 7.64e-14 ***
x2           2.07798    0.10517  19.757  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.074 on 97 degrees of freedom
Multiple R-squared:  0.8288,	Adjusted R-squared:  0.8252 
F-statistic: 234.7 on 2 and 97 DF,  p-value: < 2.2e-16
```

The values obtained from part E, are very close to the values obtained by E which the exception of the 0 percentile values for $\beta_0$ and $\beta_2$ and the 100 percentile with respect to $\beta_1$