CS 5565

LAB6

(Moving Beyond Linearity)

Shubhabrata Mukherjee

Id: 16201097

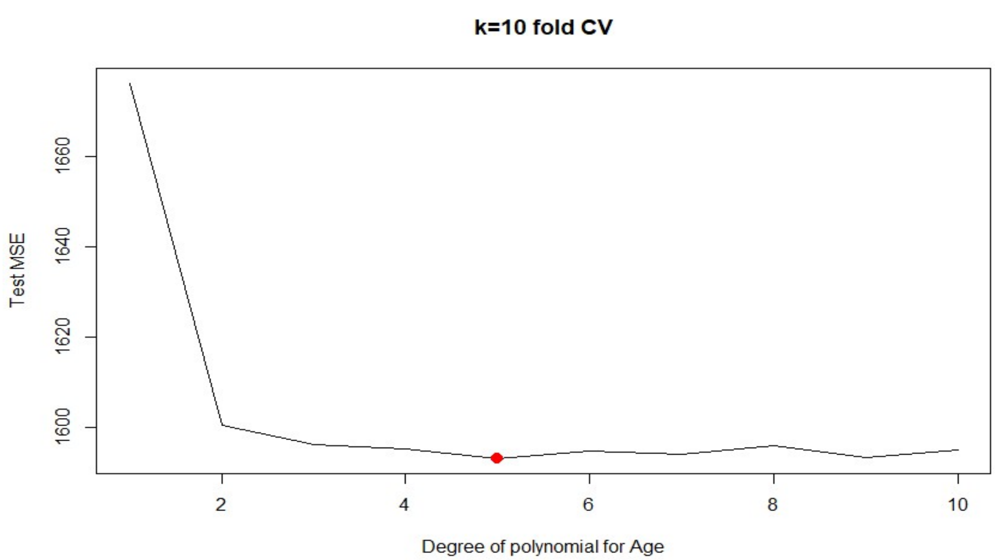
2)

By doing various polynomial regression for Wage using age we get:

```
C:/Users/shubh/Downloads/Shubh_myR_proj/
> setwd("C:/Users/shubh/Downloads/Shubh_myR_proj")
> library(ISLR)
> attach(Wage)
> names(Wage)
 [1] "year"       "age"        "maritl"     "race"       "education"  "region"     "jobclass"   "health"     "health_ins"
[10] "logwage"    "wage"
>
> polyfit.2= lm(wage~poly(age ,2) ,data=Wage)
> polyfit.3= lm(wage~poly(age ,3) ,data=Wage)
> polyfit.4= lm(wage~poly(age ,4) ,data=Wage)
> polyfit.5= lm(wage~poly(age ,5) ,data=Wage)
> polyfit.6= lm(wage~poly(age ,6) ,data=Wage)
```

Then using K = 10 fold cross validation we get:

```
Console   Terminal   Jobs
E:/shubh_projects/
> d=10
> # for K-fold cross valiadtion
> k=10
> set.seed(2)
> cv.error=rep(0,limit)
> for (i in 1:d)
+ {
+   polyfit = glm(wage~poly(age,i),data=Wage)
+   cv.error[i] = cv.glm(Wage, polyfit, K=k)$delta[1]
+ }
>
> plot(1:d,cv.error,xlab="Degree of polynomial for Age", ylab="Test MSE",type="l",main="k=10 fold CV")
> points(which.min(cv.error),cv.error[which.min(cv.error)],col="red",cex=2, pch=20)
```



So, we get best results for 5th degree of polynomial.

And then comparing the results using ANOVA we get:

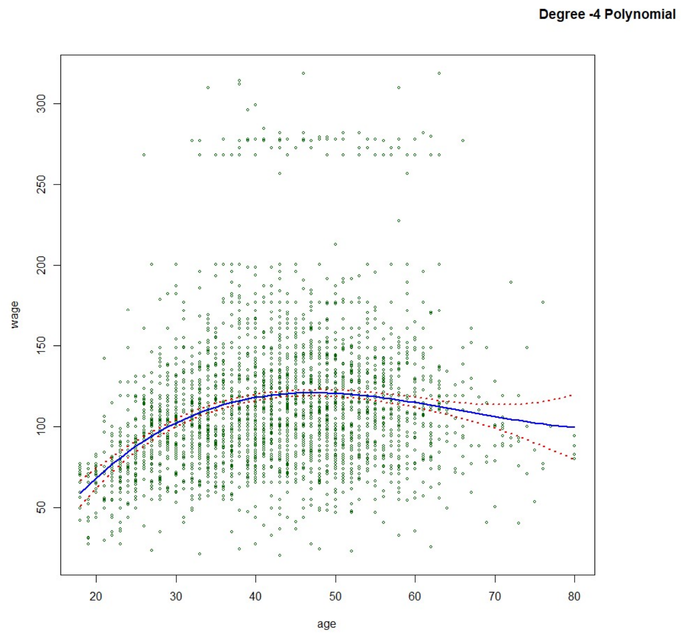
```
C:/Users/shubh/Downloads/Shubh_myR_proj/
> anova(polyfit.2, polyfit.3, polyfit.4, polyfit.5, polyfit.6)
Analysis of Variance Table

Model 1: wage ~ poly(age, 2)
Model 2: wage ~ poly(age, 3)
Model 3: wage ~ poly(age, 4)
Model 4: wage ~ poly(age, 5)
Model 5: wage ~ poly(age, 6)
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)
1   2997 4793430
2   2996 4777674  1   15755.7 9.8936 0.001675 **
3   2995 4771604  1    6070.2 3.8117 0.050989 .
4   2994 4770322  1    1282.6 0.8054 0.369565
5   2993 4766389  1    3932.3 2.4692 0.116201
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

We see that, either a cubic or quartic polynomial appear to provide a good fit to the data, but lower or higher order models are not giving better results than that.

Now making the predictions and plot we get:

For degree of polynomial 3:

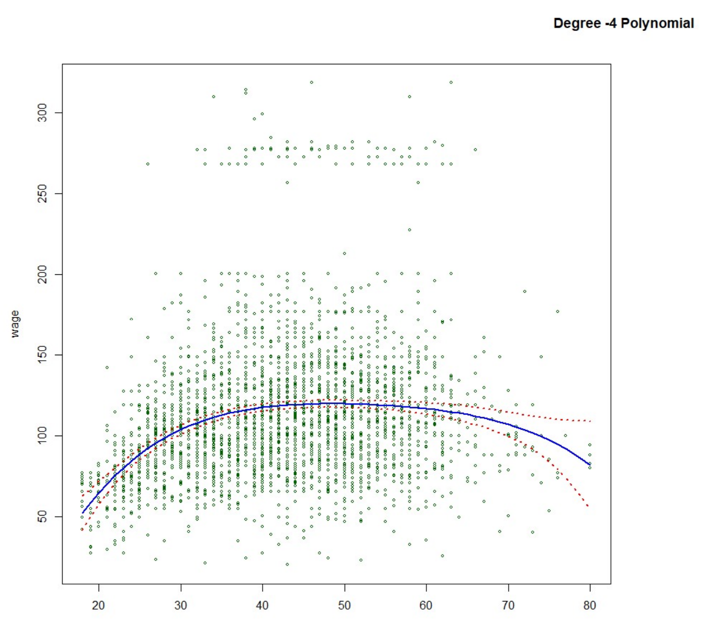
```
> fit= lm(wage~poly(age ,3) ,data=Wage)
> agelims =range(age)
> age.grid=seq (from=agelims [1], to=agelims [2])
> preds=predict (fit ,newdata =list(age=age.grid),se=TRUE)
> se.bands=cbind(preds$fit +2* preds$se.fit ,preds$fit -2* preds$se.fit)
>
> #Plotting the poly
>
> par(mfrow =c(1,2) ,mar=c(4.5 ,4.5 ,1 ,1) ,oma=c(0,0,4,0))
> plot(age ,wage ,xlim=agelims ,cex =.5, col =" darkgreen ")
> title (" Degree -4 Polynomial ",outer =T)
> lines(age.grid ,preds$fit ,lwd =2, col =" blue")
> matlines (age.grid ,se.bands ,lwd =2, col =" red",lty =3)
```



For degree of polynomial 4:

C:/Users/shubh/Downloads/Shubh_myR_proj/

```
> fit= lm(wage~poly(age ,4) ,data=Wage)
> agelims =range(age)
> age.grid=seq (from=agelims [1], to=agelims [2])
> preds=predict (fit ,newdata =list(age=age.grid),se=TRUE)
> se.bands=cbind(preds$fit +2* preds$se.fit ,preds$fit -2* preds$se.fit)
>
> #Plotting the poly
>
> par(mfrow =c(1,2) ,mar=c(4.5 ,4.5 ,1 ,1) ,oma=c(0,0,4,0))
> plot(age ,wage ,xlim=agelims ,cex =.5, col =" darkgreen ")
> title (" Degree -4 Polynomial ",outer =T)
> lines(age.grid ,preds$fit ,lwd =2, col =" blue")
> matlines (age.grid ,se.bands ,lwd =2, col =" red",lty =3)
```



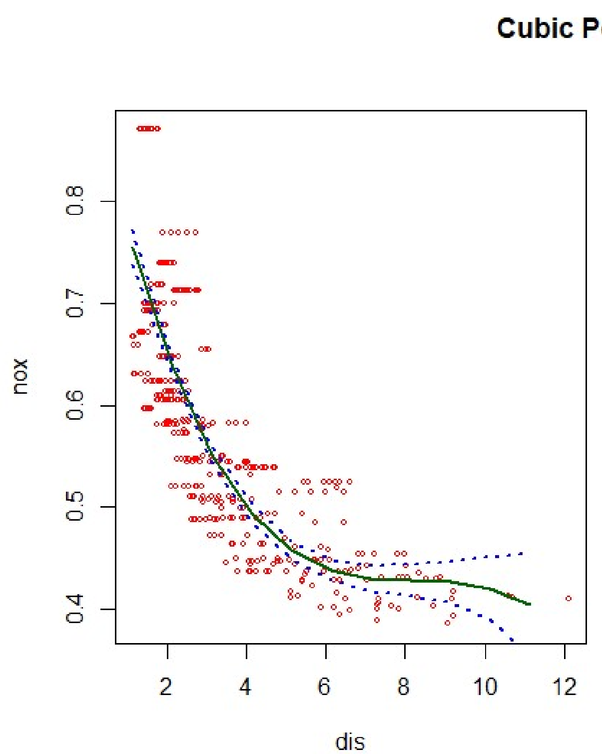
3) By doing various polynomial regression for Boston using age we get:

```
E:/shubh_projects/
> library(MASS)
> attach(Boston)
> setwd("E:/shubh_projects")
> polyfit.3= lm(nox~poly(dis,3) ,data=Boston)
> summary(polyfit.3)

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
```

Now after performing prediction and plotting the data we get:

```
E:/shubh_projects/
> fit= lm(nox~poly(dis,3) ,data=Boston)
> dislims =range(dis)
> dis.grid=seq (from=dislims [1], to=dislims [2])
> preds=predict (fit ,newdata =list(dis=dis.grid),se=TRUE)
> se.bands=cbind(preds$fit +2* preds$se.fit ,preds$fit -2* preds$se.fit)
> par(mfrow =c(1,2) ,mar=c(4.5 ,4.5 ,1 ,1) ,oma=c(0,0,4,0))
> plot(dis ,nox ,xlim=dislims ,cex =.5, col =" red ")
> title (" Cubic Polynomial ",outer =T)
> lines(dis.grid ,preds$fit ,lwd =2, col =" darkgreen")
> matlines (dis.grid ,se.bands ,lwd =2, col =" blue",lty =3)
> 
```

The plot is given below:



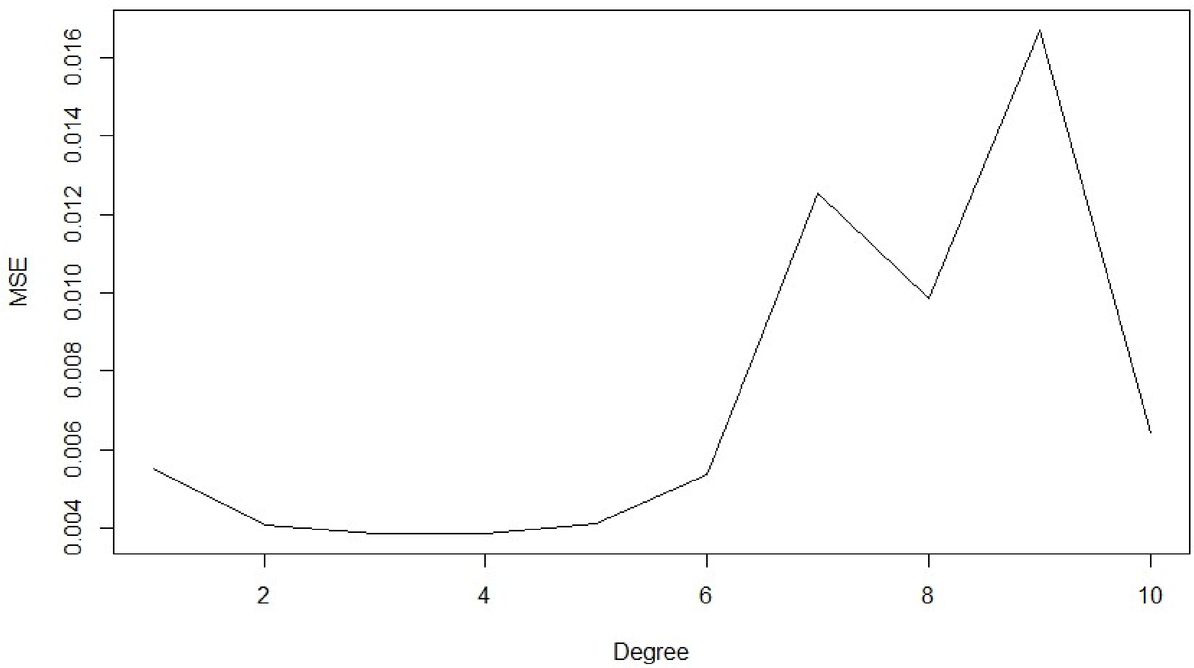
b) Now plotting polynomial (1 – 10) we get:

```
E:/shubh_projects/
> fit.1 = lm(nox~dis,data=Boston)
> polyfit.2= lm(nox~poly(dis,2) ,data=Boston)
> polyfit.3= lm(nox~poly(dis,3) ,data=Boston)
> polyfit.4= lm(nox~poly(dis,4) ,data=Boston)
> polyfit.5= lm(nox~poly(dis,5) ,data=Boston)
> polyfit.6= lm(nox~poly(dis,6) ,data=Boston)
> polyfit.7= lm(nox~poly(dis,7) ,data=Boston)
> polyfit.8= lm(nox~poly(dis,8) ,data=Boston)
> polyfit.9= lm(nox~poly(dis,9) ,data=Boston)
> polyfit.10= lm(nox~poly(dis,10) ,data=Boston)
> anova(fit.1,polyfit.2, polyfit.3, polyfit.4, polyfit.5, polyfit.6,polyfit.7,polyfit.8,polyfit.9,polyfit.10)
Analysis of Variance Table

Model  1: nox ~ dis
Model  2: nox ~ poly(dis, 2)
Model  3: nox ~ poly(dis, 3)
Model  4: nox ~ poly(dis, 4)
Model  5: nox ~ poly(dis, 5)
Model  6: nox ~ poly(dis, 6)
Model  7: nox ~ poly(dis, 7)
Model  8: nox ~ poly(dis, 8)
Model  9: nox ~ poly(dis, 9)
Model 10: nox ~ poly(dis, 10)
   Res.Df    RSS Df Sum of Sq        F    Pr(>F)
1     504 2.7686
2     503 2.0353  1   0.73330 198.1169 < 2.2e-16 ***
3     502 1.9341  1   0.10116  27.3292 2.535e-07 ***
4     501 1.9330  1   0.00113   0.3040  0.581606
5     500 1.9153  1   0.01769   4.7797  0.029265 *
6     499 1.8783  1   0.03703  10.0052  0.001657 **
7     498 1.8495  1   0.02877   7.7738  0.005505 **
8     497 1.8356  1   0.01385   3.7429  0.053601 .
9     496 1.8333  1   0.00230   0.6211  0.431019
10    495 1.8322  1   0.00116   0.3133  0.575908
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

RSS ‘s are: 2.7686, 2.0353,1.9341,1.933,1.9153,1.8783,1.8495,1.8356,1.8333,1.8322

c) Performing cross validation we get:

```
E:/shubh_projects/
> library(boot)
> inter=rep(NA, 10)
> for (i in 1:10) {
+   polyfit = glm(nox ~ poly(dis, i), data = Boston)
+   inter[i] = cv.glm(Boston, fit, K = 10)$delta[1]
+ }
> plot(1:10, inter, xlab = "Degree", ylab = "MSE", type = "l")
```



So, from the plot we can validate, optimal degree for polynomial is 4, as for 4 we get the lowest MSE.

d)

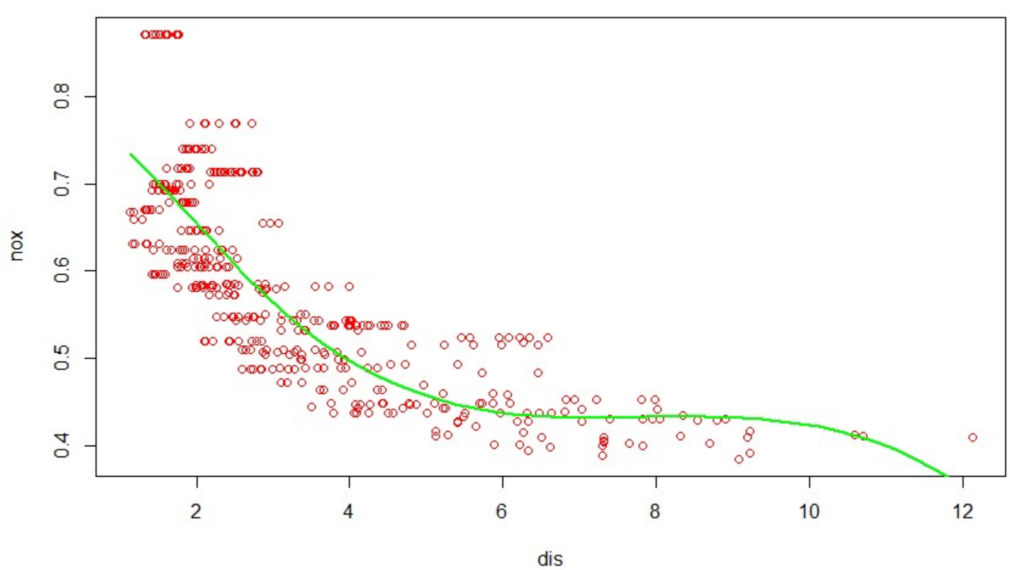
Using the bs() function we fit a regression spline to predict nox using dis

E:/shubh_projects/

```
> library(splines)
> spfit=lm(nox ~ bs(dis, df=4), data = Boston)
> summary(spfit)

Call:
lm(formula = nox ~ bs(dis, df = 4), data = Boston)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.124622 -0.039259 -0.008514  0.020850  0.193891 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)        0.73447    0.01460  50.306  < 2e-16 ***
bs(dis, df = 4)1 -0.05810    0.02186  -2.658  0.00812 ** 
bs(dis, df = 4)2 -0.46356    0.02366 -19.596  < 2e-16 ***
bs(dis, df = 4)3 -0.19979    0.04311  -4.634 4.58e-06 ***
bs(dis, df = 4)4 -0.38881    0.04551  -8.544  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06195 on 501 degrees of freedom
Multiple R-squared:  0.7164,	Adjusted R-squared:  0.7142 
F-statistic: 316.5 on 4 and 501 DF,  p-value: < 2.2e-16
```

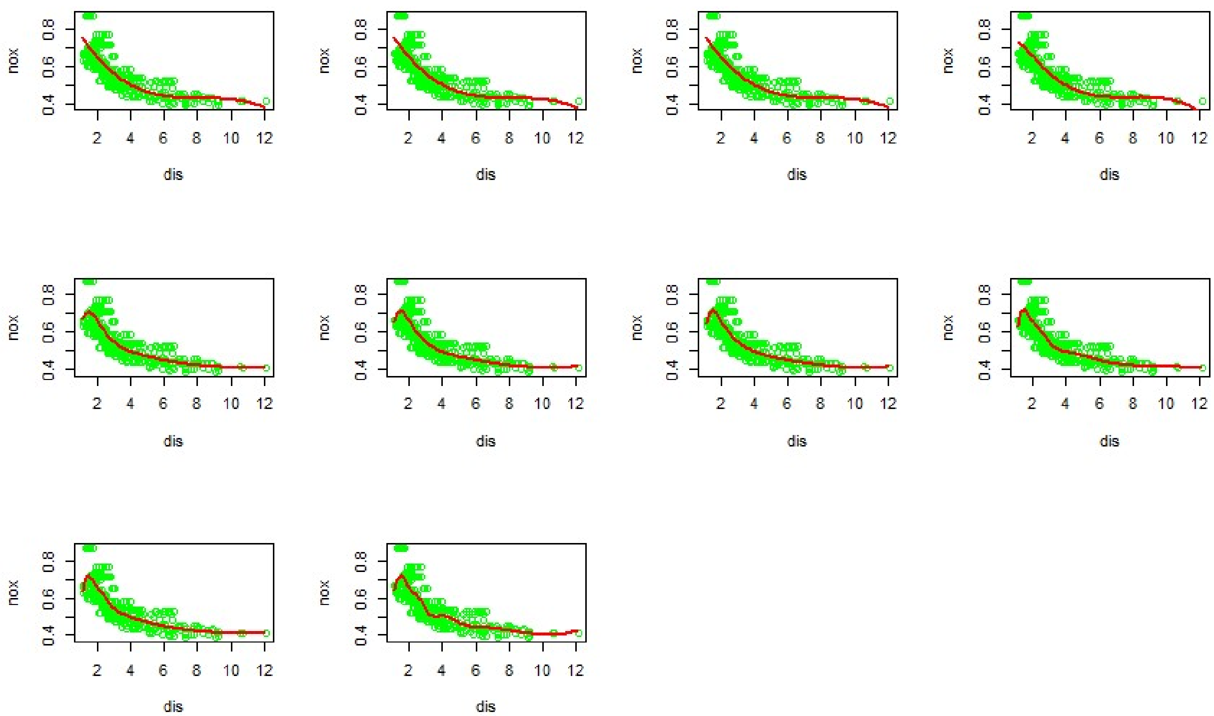
E:/shubh_projects/

```
> limdis = range(Boston$dis)
> #limdis
> x = seq(from = limdis[1], to = limdis[2], by = .25)
> sp.pred = predict(spfit, data.frame(dis = x))
> plot(nox ~ dis, data = Boston, col = "red")
> lines(x, sp.pred, col = "green", lwd = 2)
```



e) Now fitting a regression spline for a range of degrees of freedom we get:

```
E:/shubh_projects/
> par(mfrow=c(3,4))
> rss = rep(0,10)
> for(i in 1:10){
+   spfit=lm(nox ~ bs(dis, i), data = Boston)
+   limdis=range(Boston$dis)
+   x=seq(from = limdis[1], to = limdis[2], by = 0.1)
+   sp.pred=predict(spfit, data.frame(dis = x))
+   plot(nox ~ dis, data = Boston, col = "green")
+   lines(x, sp.pred, col = "red", lwd = 2)
+   rss[i] = sum(spfit$residuals^2)
+ }
Warning messages:
1: In bs(dis, i) : 'df' was too small; have used 3
2: In bs(dis, i) : 'df' was too small; have used 3
> rss
 [1] 1.934107 1.934107 1.934107 1.922775 1.840173 1.833966 1.829884 1.816995 1.825653 1.792535
```

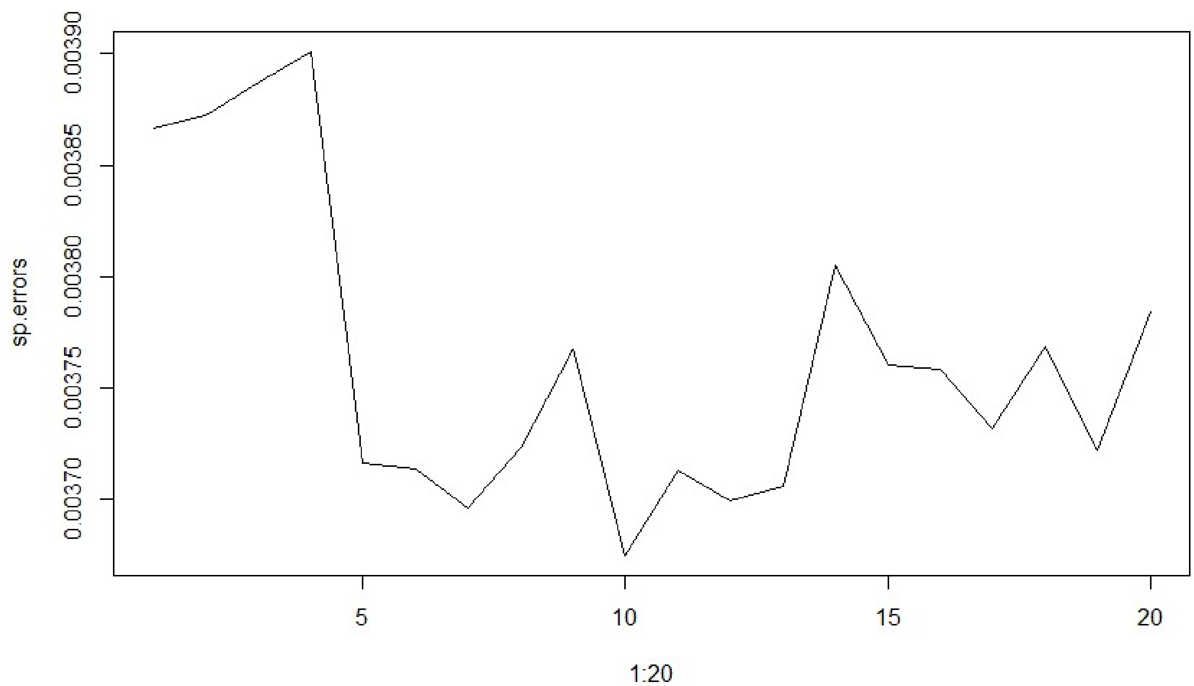


So, from the RSS value we can see that in general RSS decreases when the polynomial degree gets higher. However, the 9th result gives a higher value. Also we see a warning message that,

In 1 and 2 ‘df’ was too small, so it have automatically used 3.

f) Performing the cross-validation or another approach in order to select the best degrees of freedom for a regression spline we get below results:

```
E:/shubh_projects/
> sp.errors = sapply(1:20, function(i){
+   spfit = glm(nox ~ bs(dis, df = i), data=Boston)
+   return(cv.glm(Boston, spfit, K = 10)$delta[2])
+ })
There were 50 or more warnings (use warnings() to see the first 50)
>
> plot(1:20, sp.errors,type = "l")
>
```



We get the lowest error when df=10.