STA6703 SML HW7

Christopher Marais and Macy Somoskey

Chapter 7

Problem 1

a.)

$$f_1(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

b.)

$$\begin{aligned}
f_2(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)^3 \\
&= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x^3 - 3x^2\xi + 3x\xi^2 - \xi^3) \\
&= (\beta_0 - \beta_4 \xi^3) + x(\beta_1 + 3\beta_4\xi^2) + x^2(\beta_2 - 3\beta_4\xi) + x^3(\beta_3 + \beta_4)
\end{aligned}$$

c.)

$$f_1(\xi) = \beta_0 + \beta_1 \xi + \beta_2 \xi^2 + \beta_3 \xi^3$$

$$\begin{aligned}
f_2(\xi) &= (\beta_0 - \beta_4 \xi^3) + \xi(\beta_1 + 3\beta_4\xi^2) + \xi^2(\beta_2 - 3\beta_4\xi) + \xi^3(\beta_3 + \beta_4) \\
&= \beta_0 - \beta_4\xi^3 + \beta_1\xi + 3\beta_4\xi^3 + \beta_2\xi^2 - 3\beta_4\xi^3 + \beta_3\xi^3 + \beta_4\xi^3 \\
&= \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3 = f_1(\xi)
\end{aligned}$$

d.)

$$f_1'(\xi) = \beta_1 + 2\beta_2\xi + 3\beta_3\xi^2$$

$$\begin{aligned}
f_2'(\xi) &= (\beta_1 + 3\beta_4\xi^2) + 2\xi(\beta_2 - 3\beta_4\xi) + 3\xi^2(\beta_3 + \beta_4) \\
&= \beta_1 + 3\beta_4\xi^2 + 2\beta_2\xi - 6\beta_4\xi^2 + 3\beta_3\xi^2 + 3\beta_4\xi^2 \\
&= \beta_1 + 2\beta_2\xi + 3\beta_3\xi^2 = f_1'(\xi)
\end{aligned}$$

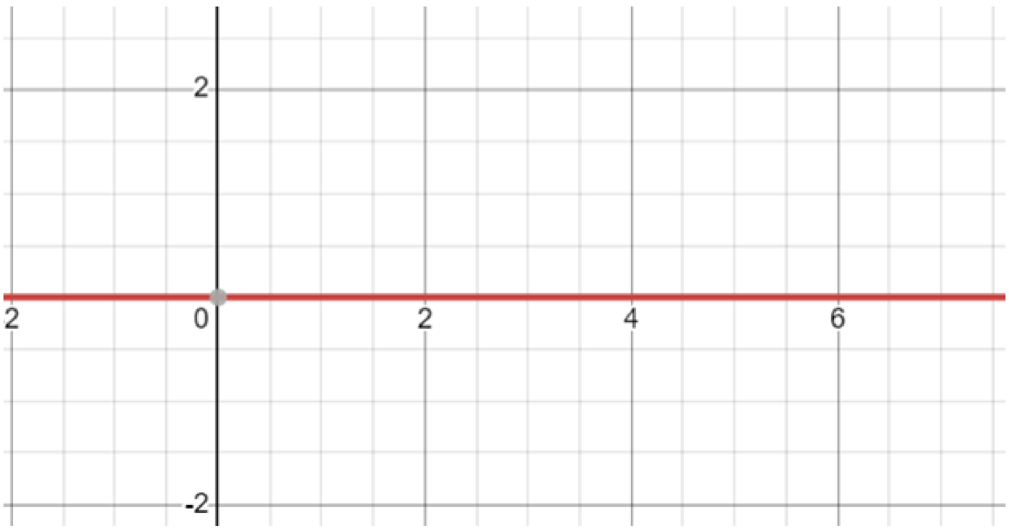
e.)

$$f_1''(\xi) = 2\beta_2 + 6\beta_3\xi$$

$$\begin{aligned} f_2''(\xi) &= 2(\beta_2 - 3\beta_4\xi) + 6\xi(\beta_3 + \beta_4) \\ &= 2\beta_2 - 6\beta_4\xi + 6\beta_3\xi + 6\beta_4\xi \\ &= 2\beta_2 + 6\beta_3\xi^2 = f_1''(\xi) \end{aligned}$$

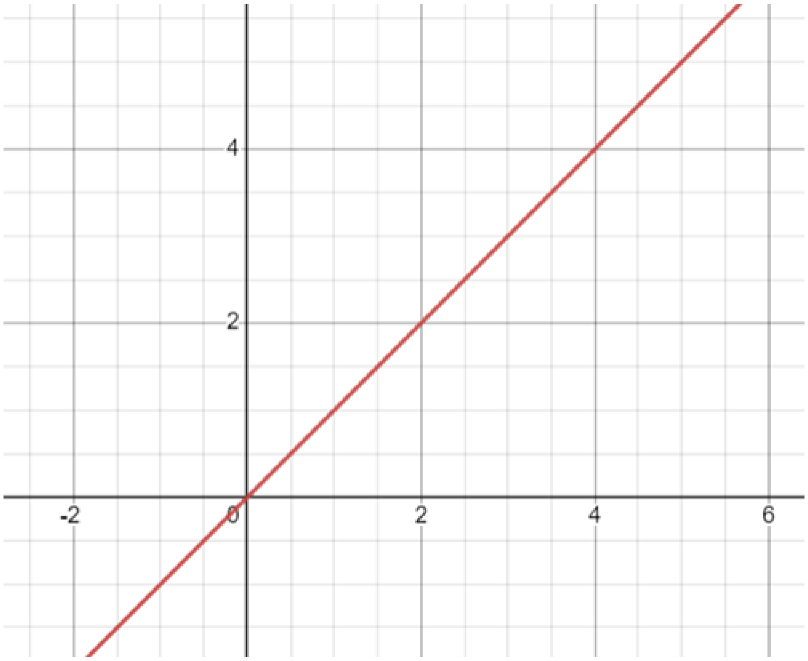
Problem 2

a.)



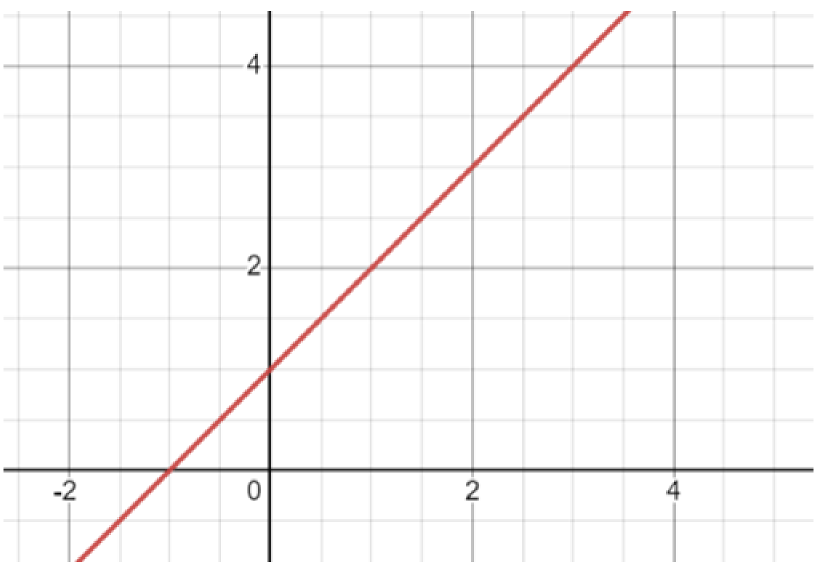
Lambda handles over fitting and controls the wigglyness of the function. In this case g hat = 0 due to the large smoothing parameter. g_(0)(x) -> 0

b.)



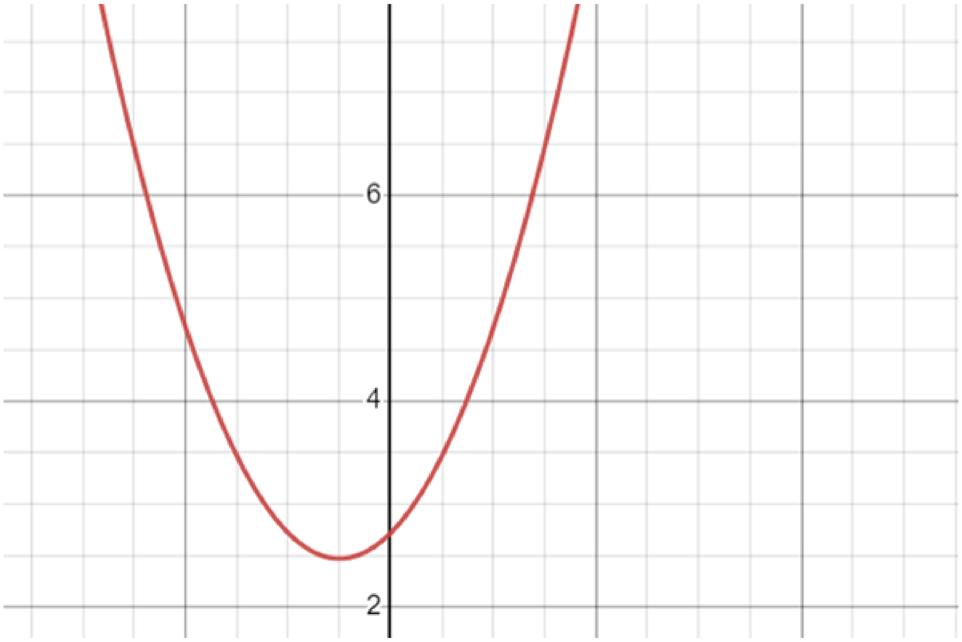
g hat = cx+d where c and d are constants due to the large smoothing parameter. g_(2)(x) -> 0

c.)



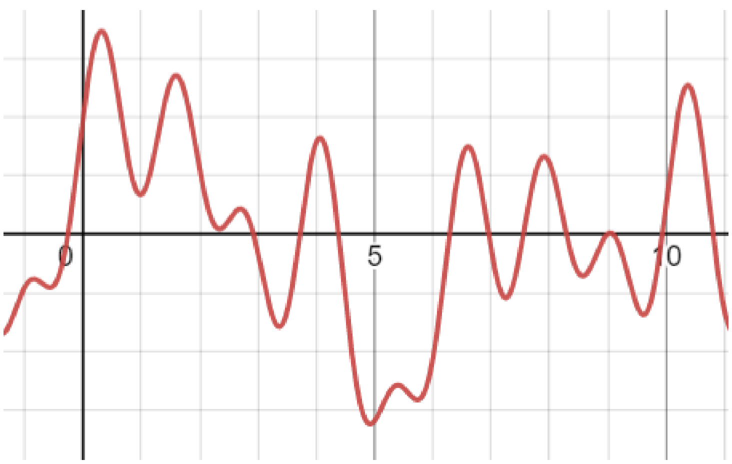
g hat = cx2+dx+e where c, d, and e are constants due to the large smoothing parameter. g_(3)(x) -> 0

d.)



The smaller lambda, the more wiggly the function and g hat interpolates yi when Lambda = 0. No smoothing occurs.

e.)



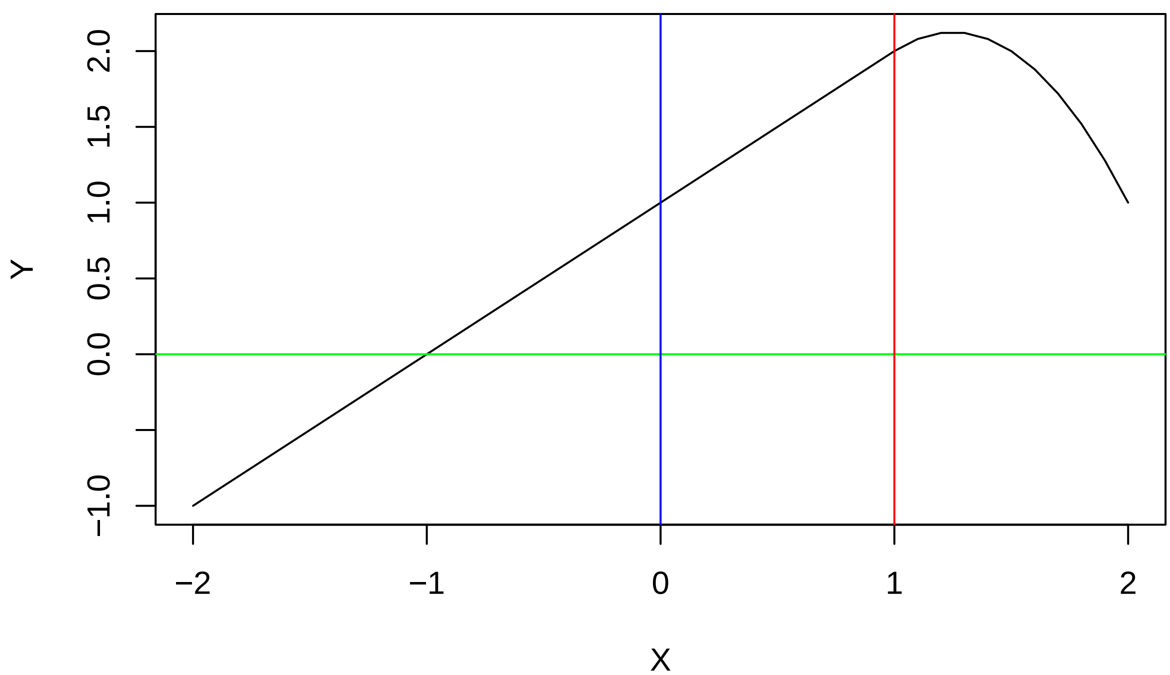
The smaller lambda, the more wiggly the function and g hat interpolates yi when Lambda = 0. No smoothing occurs.

Problem 3

```
X = seq(-2,2,0.1)
Y = rep(NA,length(X))
for (i in 1:length(X)){
  if (X[i]<1){
    Y[i] = 1 + 1*X[i]
```

```
  }
  else{
    Y[i] = 1 + 1*X[i] - 2*(X[i]-1)^2
  }
}

plot(X,Y,type='l') +
abline(h=0, col = "green") +
abline(v=0, col = "blue") +
abline(v = 1, col = "red")
```




```
## integer(0)
```

As long s x<1 the line is linear with a slope of 1. For 1<x the line is quadratic.

Problem 6

a.)

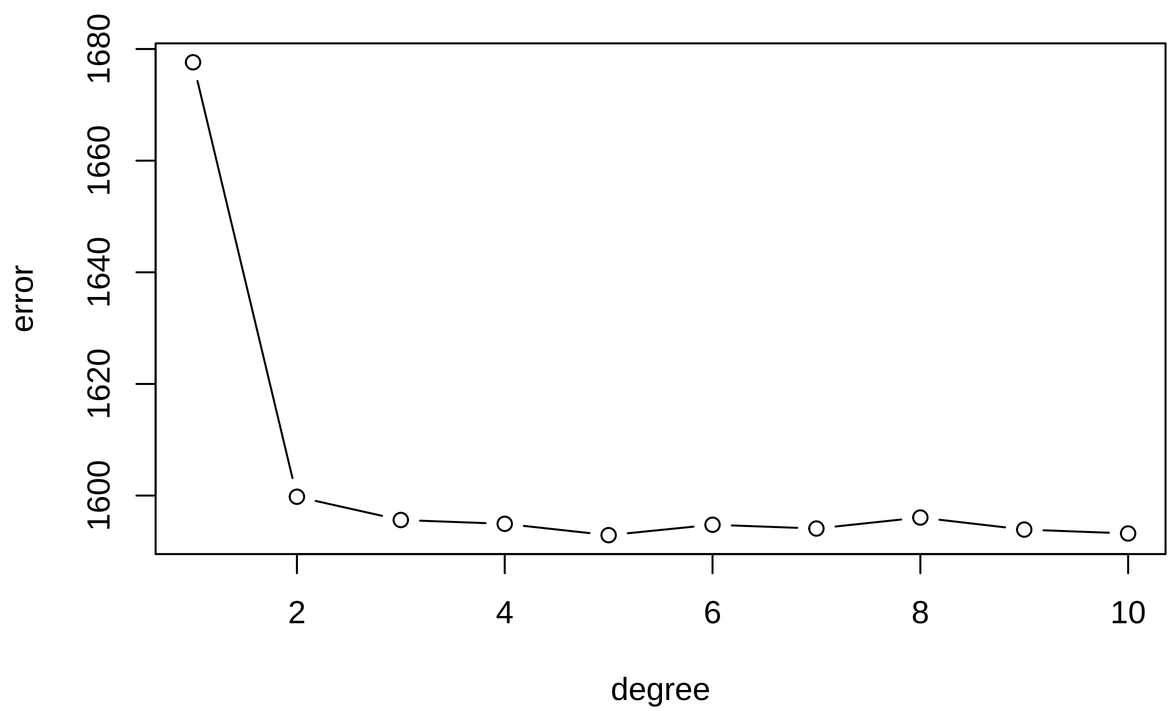
```
library(ISLR)
library(boot)
```

```
set.seed(0)
cv.error.10 = rep(0,10)
for (i in 1:10) {
  glm.fit=glm(wage~poly(age,i),data=Wage)
  cv.error.10[i]=cv.glm(Wage,glm.fit,K=10)$delta[1]
}

cv.error.10
```

```
##  [1] 1677.625 1599.790 1595.625 1594.940 1592.908 1594.775 1594.096 1596.071
##  [9] 1593.908 1593.211
```

```
plot(cv.error.10, type="b", xlab="degree", ylab="error")
```




```
lm.fit = glm(wage~poly(age,4),data=Wage)
summary(lm.fit)
```

```
##
## Call:
## glm(formula = wage ~ poly(age, 4), data = Wage)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -98.707  -24.626   -4.993   15.217  203.693
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     111.7036     0.7287 153.283  < 2e-16 ***
## poly(age, 4)1  447.0679    39.9148  11.201  < 2e-16 ***
## poly(age, 4)2 -478.3158    39.9148 -11.983  < 2e-16 ***
## poly(age, 4)3  125.5217    39.9148   3.145  0.00168 **
## poly(age, 4)4  -77.9112    39.9148  -1.952  0.05104 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1593.19)
##
##     Null deviance: 5222086  on 2999  degrees of freedom
## Residual deviance: 4771604  on 2995  degrees of freedom
## AIC: 30641
##
## Number of Fisher Scoring iterations: 2
```

From the Cross validation performance we can see that the performance does not improve much when the degree is increased from 4.

```r
fit.1 = lm(wage~age ,data=Wage)
fit.2 = lm(wage~poly(age ,2) ,data=Wage)
fit.3 = lm(wage~poly(age ,3) ,data=Wage)
fit.4 = lm(wage~poly(age ,4) ,data=Wage)
fit.5 = lm(wage~poly(age ,5) ,data=Wage)
anova(fit.1,fit.2,fit.3,fit.4,fit.5)
```

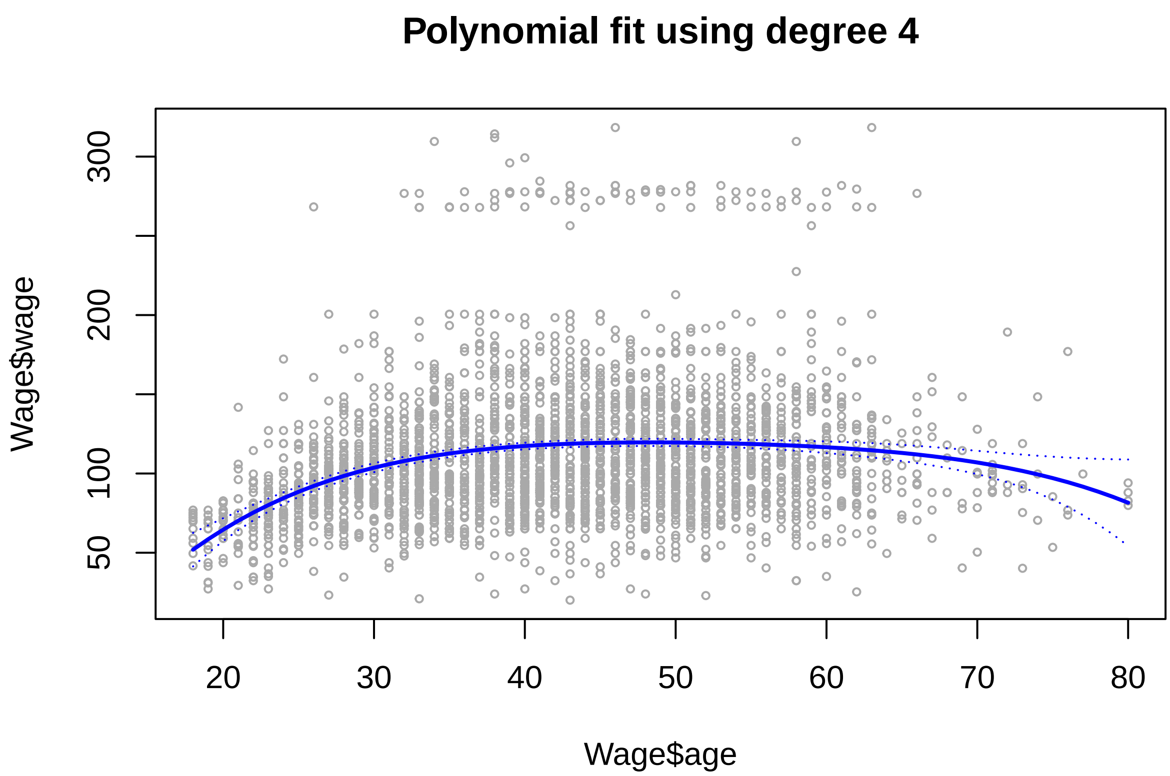
```
## Analysis of Variance Table
##
## Model 1: wage ~ age
## Model 2: wage ~ poly(age, 2)
## Model 3: wage ~ poly(age, 3)
## Model 4: wage ~ poly(age, 4)
## Model 5: wage ~ poly(age, 5)
##   Res.Df     RSS Df Sum of Sq        F    Pr(>F)
## 1   2998 5022216
## 2   2997 4793430  1    228786 143.5931 < 2.2e-16 ***
## 3   2996 4777674  1     15756   9.8888  0.001679 **
## 4   2995 4771604  1      6070   3.8098  0.051046 .
## 5   2994 4770322  1      1283   0.8050  0.369682
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The comparison of model 1 and 2 to the model 4 shows p-values of statistical significance below 0.05. Models 3 almost has a p-values of statistical significance when comapred to model 4. Model 5 has a high p-value when compared to model 4 that is not statistically significant. This indicates that a cubic or quadratic model has the best fit to the data. These results are the same as waht was gained through polynomial regression.

```
agelims=range(Wage$age)
age.grid=seq(from=agelims[1],to=agelims[2])

preds=predict(lm.fit,newdata=list(age=age.grid),se=TRUE)
se.bands=cbind(preds$fit+2*preds$se.fit,preds$fit-2*preds$se.fit)

plot(Wage$age,Wage$wage,xlim=agelims,cex=.5,col="darkgrey")
title("Polynomial fit using degree 4")
lines(age.grid,preds$fit,lwd=2,col="blue")
matlines(age.grid,se.bands,lwd =1,col="blue",lty =3)
```



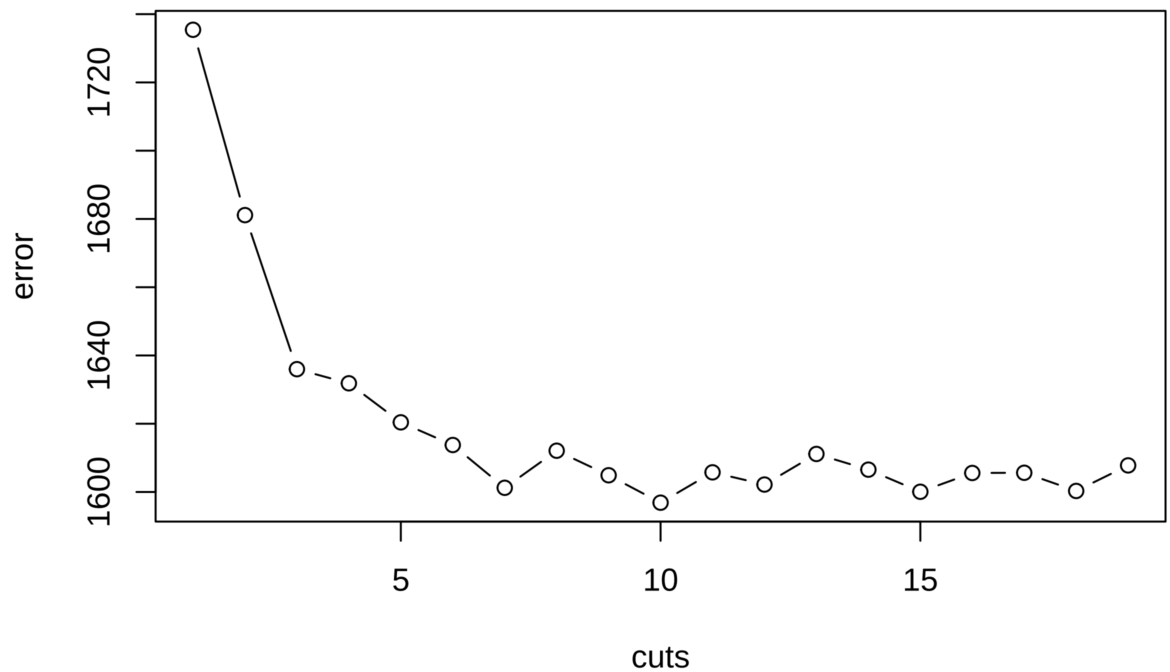
b.)

```
set.seed(0)
cv.error.20 = rep(NA,19)
for (i in 2:20) {
  Wage$age.cut = cut(Wage$age,i)
  step.fit=glm(wage~age.cut,data=Wage)
  cv.error.20[i-1]=cv.glm(Wage,step.fit,K=10)$delta[1]
}
```

```
cv.error.20
```

```
##  [1] 1735.421 1681.123 1635.985 1631.846 1620.407 1613.770 1601.228 1612.097
##  [9] 1604.901 1596.871 1605.760 1602.182 1611.143 1606.552 1600.069 1605.572
## [17] 1605.633 1600.308 1607.792
```

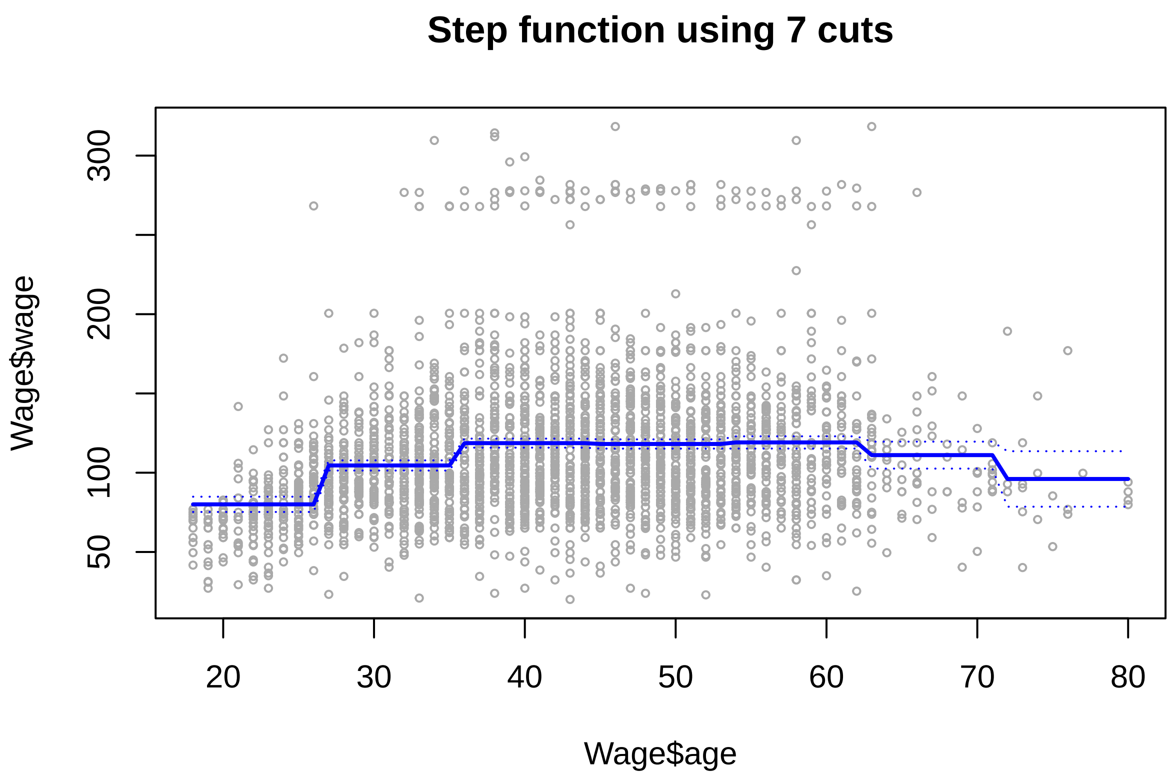
```
plot(cv.error.20,type='b',ylab="error",xlab='cuts')
```



From the data we can see that after 7 cuts in the step function that we see little improvement in performance.

```
step.fit = glm(wage~cut(age,7), data=Wage)
preds2=predict(step.fit,newdata=list(age=age.grid), se=T)
se.bands2=cbind(preds2$fit+2*preds2$se.fit,preds2$fit-2*preds2$se.fit)
plot(Wage$age,Wage$wage,xlim=agelims,cex=.5,col="darkgrey")
title("Step function using 7 cuts")
lines(age.grid,preds2$fit,lwd=2,col="blue")
matlines(age.grid,se.bands2,lwd =1,col="blue",lty =3)
```

Step function using 7 cuts



Problem 10 ### a.)

```
library(caTools)

set.seed(0)
college_data = College
college_sample = sample.split(college_data$Outstate, SplitRatio = 0.80)
college_train = subset(college_data, college_sample==TRUE)
college_test = subset(college_data, college_sample==FALSE)
```

```
library(leaps)

fit.fwd = regsubsets(Outstate~., data=college_train, nvmax=17, method="forward")
fit.summary = summary(fit.fwd)

which.min(fit.summary$cp)
```

```
## [1] 13
```

```
which.min(fit.summary$bic)
```

```
## [1] 12
```

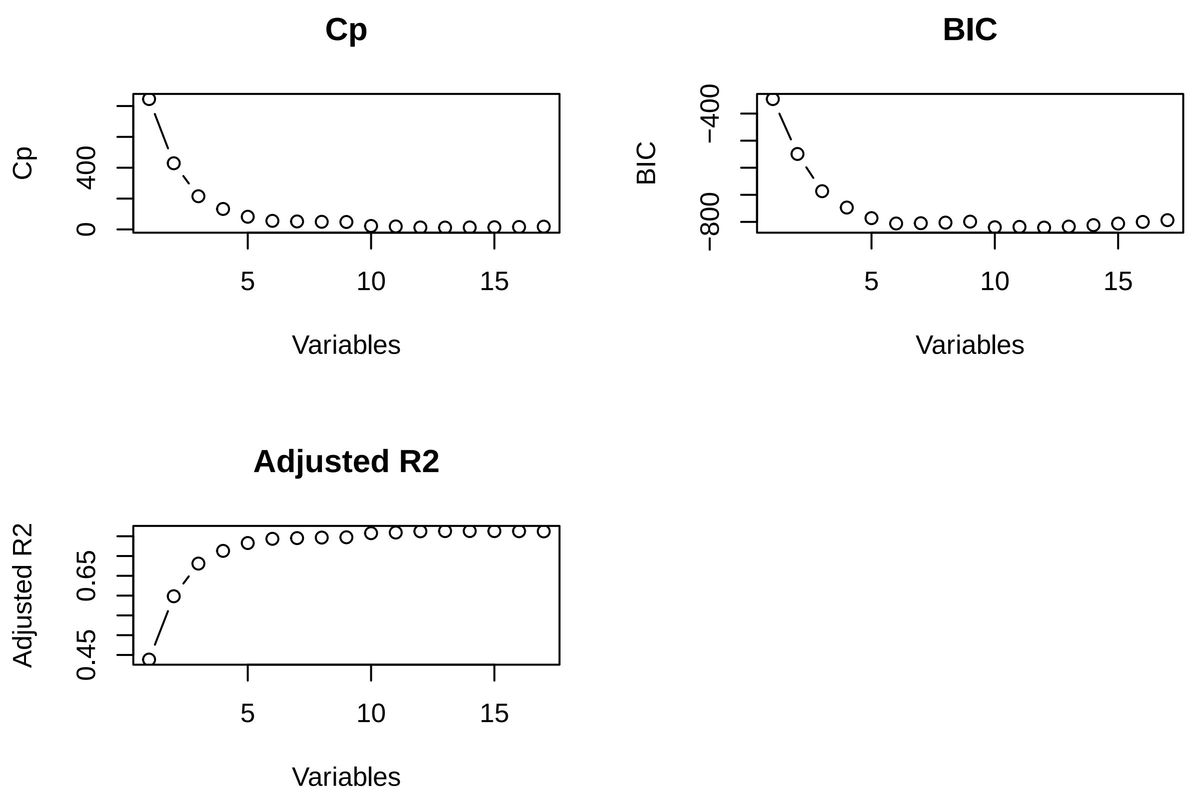
```
which.max(fit.summary$adjr2)
```

```
## [1] 14
```

```
par(mfrow=c(2,2))
plot(1:17, fit.summary$cp,xlab="Variables",ylab="Cp",main="Cp", type='b')
plot(1:17, fit.summary$bic,xlab="Variables",ylab="BIC",main="BIC", type='b')
plot(1:17, fit.summary$adjr2,xlab="Variables",ylab="Adjusted R2",main="Adjusted R2", type='b')

coef(fit.fwd,6)
```

```
##   (Intercept)    PrivateYes    Room.Board      Terminal   perc.alumni
## -4410.6502563  2925.5538789     0.9695901    43.7622879    42.6904751
##        Expend     Grad.Rate
##     0.2092328    30.1696002
```



From the Cp, BIC, and Adjusted R squared we see that they are all in agreement that after 6 variables are included the model does not improve dramatically. Therefore 6 variables is ideal as it is lower complexity with a high relative performance.

b.)

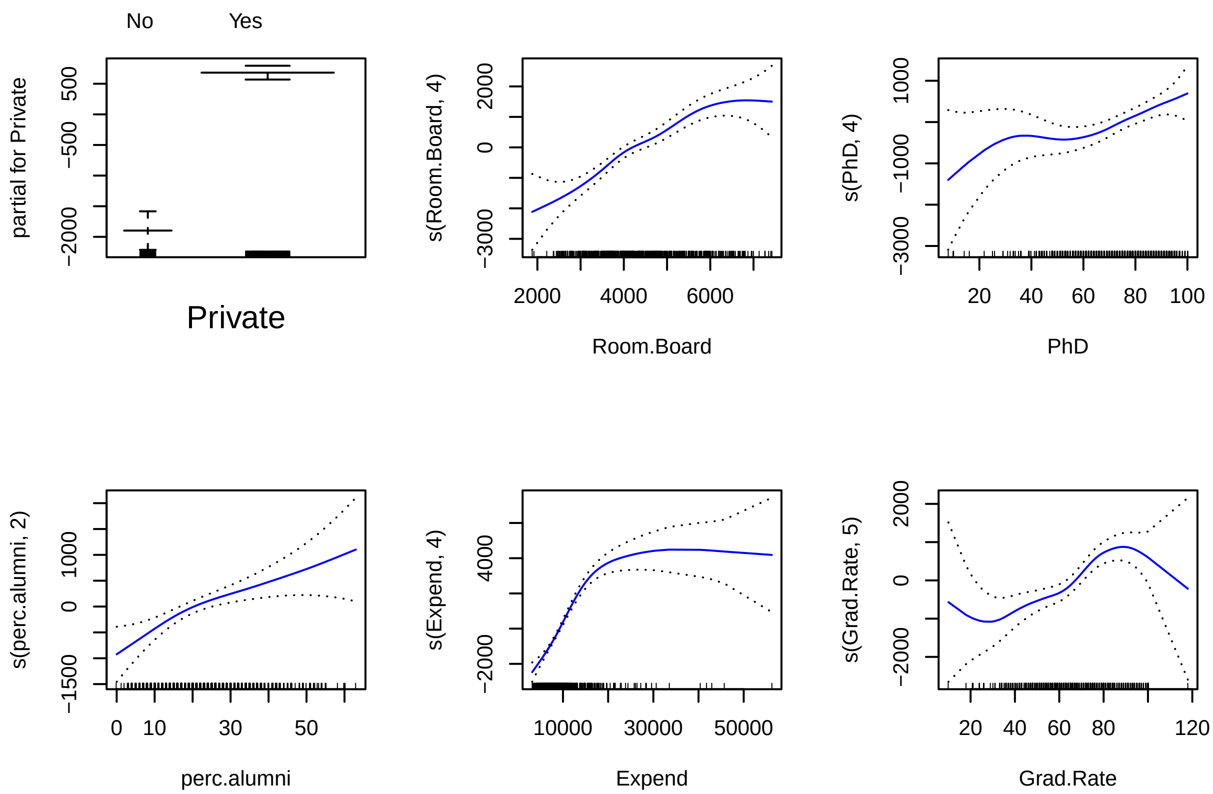
```r
library(gam)
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
## Loaded gam 1.20.2
```

```r
gam.m1 = gam(Outstate~Private+
               s(Room.Board,4)+
               s(PhD,4)+
               s(perc.alumni,2)+
               s(Expend,4)+
               s(Grad.Rate,5), data=college_train)
par(mfrow=c(2,3))
plot(gam.m1, col="blue", se=T)
```



When all other variables are kept constant the state tuition seems to increase as room costs or the proportion of donating alumni increases.

c.)

```r
preds = predict(gam.m1,newdata = college_test)
mse = mean((college_test$Outstate - preds)^2)
print(mse)
```

```
## [1] 3394714
```

The graduation rate seems to have a non-linear relationship with the outstate variable.

d.)

```r
gam.m2 = gam(Outstate~Private+s(Room.Board,4)+s(PhD,4)+s(perc.alumni,2)+s(Expend,4), data=college
gam.m3 = gam(Outstate~Private+s(Room.Board,4)+s(PhD,4)+s(perc.alumni,2)+s(Expend,4)+Grad.Rate, da
gam.m4 = gam(Outstate~Private+s(Room.Board,4)+s(PhD,4)+s(perc.alumni,2)+s(Expend,4)+s(Grad.Rate,4
anova(gam.m2,gam.m3,gam.m4,gam.m1, test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ Private + s(Room.Board, 4) + s(PhD, 4) + s(perc.alumni,
##     2) + s(Expend, 4)
## Model 2: Outstate ~ Private + s(Room.Board, 4) + s(PhD, 4) + s(perc.alumni,
##     2) + s(Expend, 4) + Grad.Rate
## Model 3: Outstate ~ Private + s(Room.Board, 4) + s(PhD, 4) + s(perc.alumni,
##     2) + s(Expend, 4) + s(Grad.Rate, 4)
## Model 4: Outstate ~ Private + s(Room.Board, 4) + s(PhD, 4) + s(perc.alumni,
##     2) + s(Expend, 4) + s(Grad.Rate, 5)
##   Resid. Df Resid. Dev      Df  Deviance       F   Pr(>F)
## 1       605 2247543765
## 2       604 2135125410 1.00000 112418355 32.1493 2.22e-08 ***
## 3       601 2105614375 2.99983  29511034  2.8133  0.03864 *
## 4       600 2098054662 0.99984   7559714  2.1623  0.14196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the Anova we can see that a Generalized Additive Model with a non-linear spline of degree 4 that also includes the Grad.rate variable is required to produce the response.