Homework4

Nam Jun Lee

12/03/2021

Q1. Question 7 of Chapter 7 of the ISLR book. (Page 299).

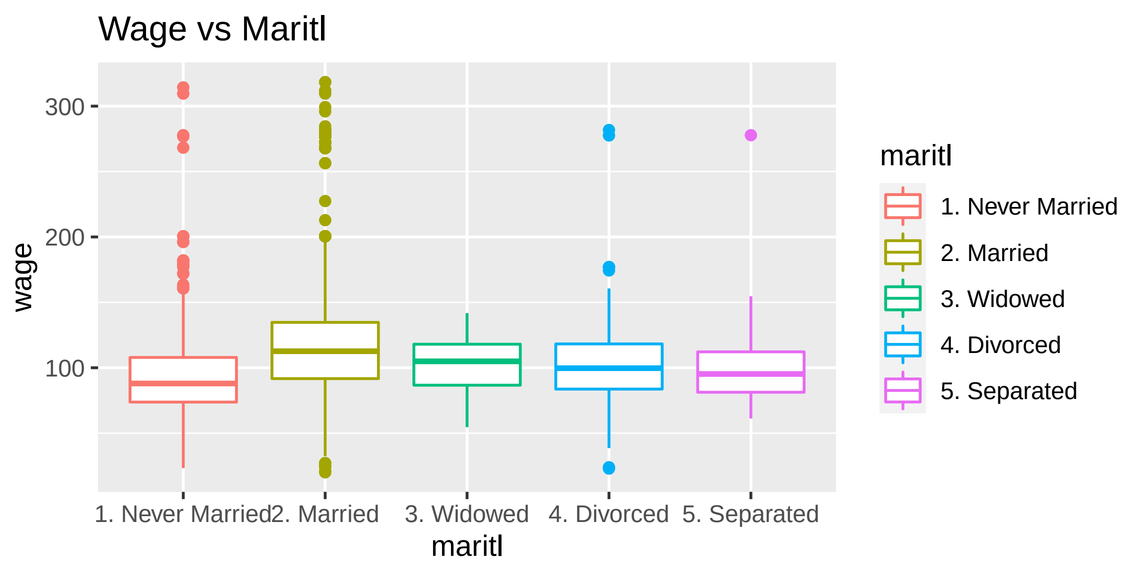
The Wage data set contains a number of other features not explored in this chapter, such as marital status (maritl), job class (jobclass), and others. Explore the relationships between some of these other predictors and wage, and use non-linear fitting techniques in order to fit flexible models to the data. Create plots of the results obtained, and write a summary of your findings.

```r
# Wage dataset into Wage variable
Wage <- ISLR::Wage
# summary of Wage dataset
summary(Wage)
```

```
##       year           age                     maritl           race     
##  Min.   :2003   Min.   :18.00   1. Never Married: 648   1. White:2480  
##  1st Qu.:2004   1st Qu.:33.75   2. Married      :2074   2. Black: 293  
##  Median :2006   Median :42.00   3. Widowed      :  19   3. Asian: 190  
##  Mean   :2006   Mean   :42.41   4. Divorced     : 204   4. Other:  37  
##  3rd Qu.:2008   3rd Qu.:51.00   5. Separated    :  55                  
##  Max.   :2009   Max.   :80.00                                          
##                                                                        
##               education                     region               jobclass   
##  1. < HS Grad      :268   2. Middle Atlantic   :3000   1. Industrial :1544  
##  2. HS Grad        :971   1. New England       :   0   2. Information:1456  
##  3. Some College   :650   3. East North Central:   0                        
##  4. College Grad   :685   4. West North Central:   0                        
##  5. Advanced Degree:426   5. South Atlantic    :   0                        
##                           6. East South Central:   0                        
##                           (Other)              :   0                        
##             health      health_ins      logwage           wage       
##  1. <=Good     : 858   1. Yes:2083   Min.   :3.000   Min.   : 20.09  
##  2. >=Very Good:2142   2. No : 917   1st Qu.:4.447   1st Qu.: 85.38  
##                                      Median :4.653   Median :104.92  
##                                      Mean   :4.654   Mean   :111.70  
##                                      3rd Qu.:4.857   3rd Qu.:128.68  
##                                      Max.   :5.763   Max.   :318.34  
## 
```

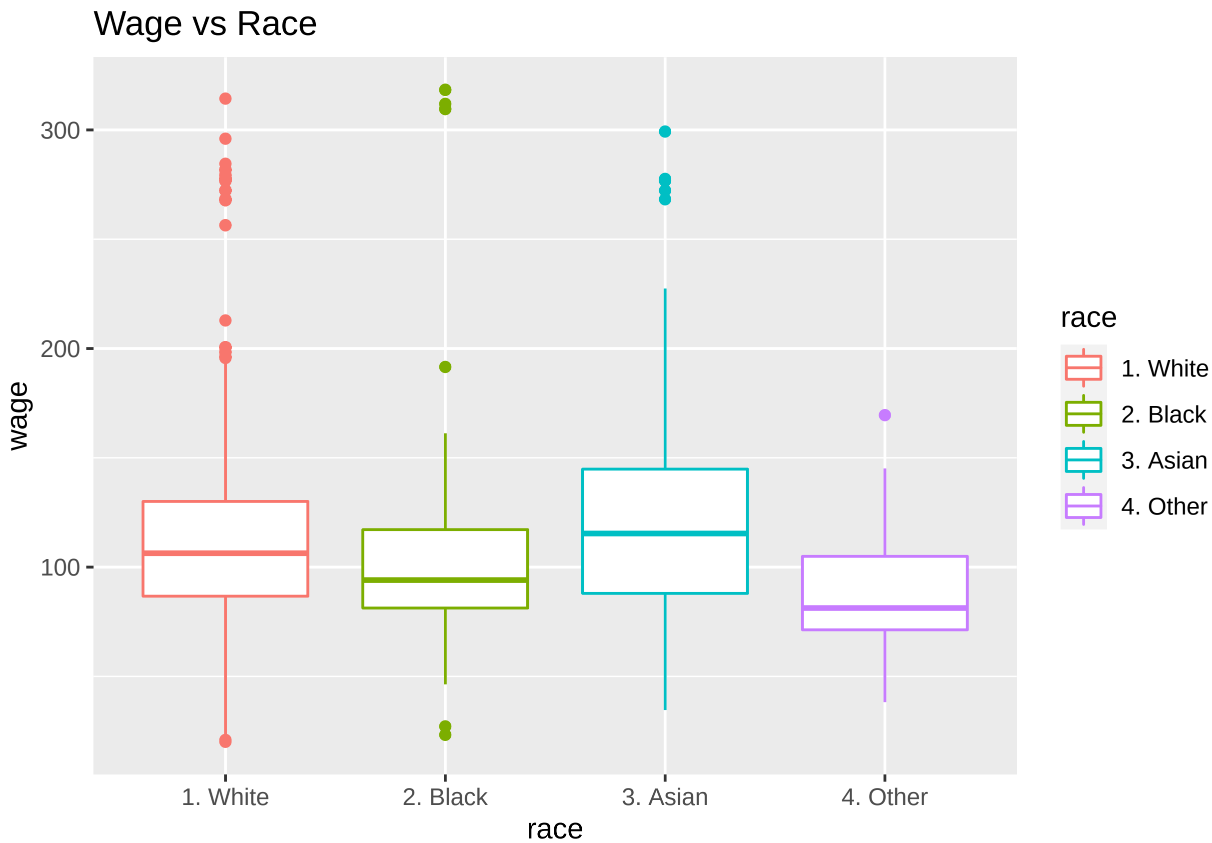
When checking this wage dataset, it can be seen that categorical variables are maritl, race, education, region, job class, health, and health_ins. In addition, the region variable is divided into six categories, but considering that there are 3,000 middle atlantic alone, it is better to exclude this variable.

```
# box plot maritl variable
ggplot(data = Wage, aes(x = maritl, y = wage, color = maritl)) + geom_boxplot() +
    ggtitle("Wage vs Maritl")
```



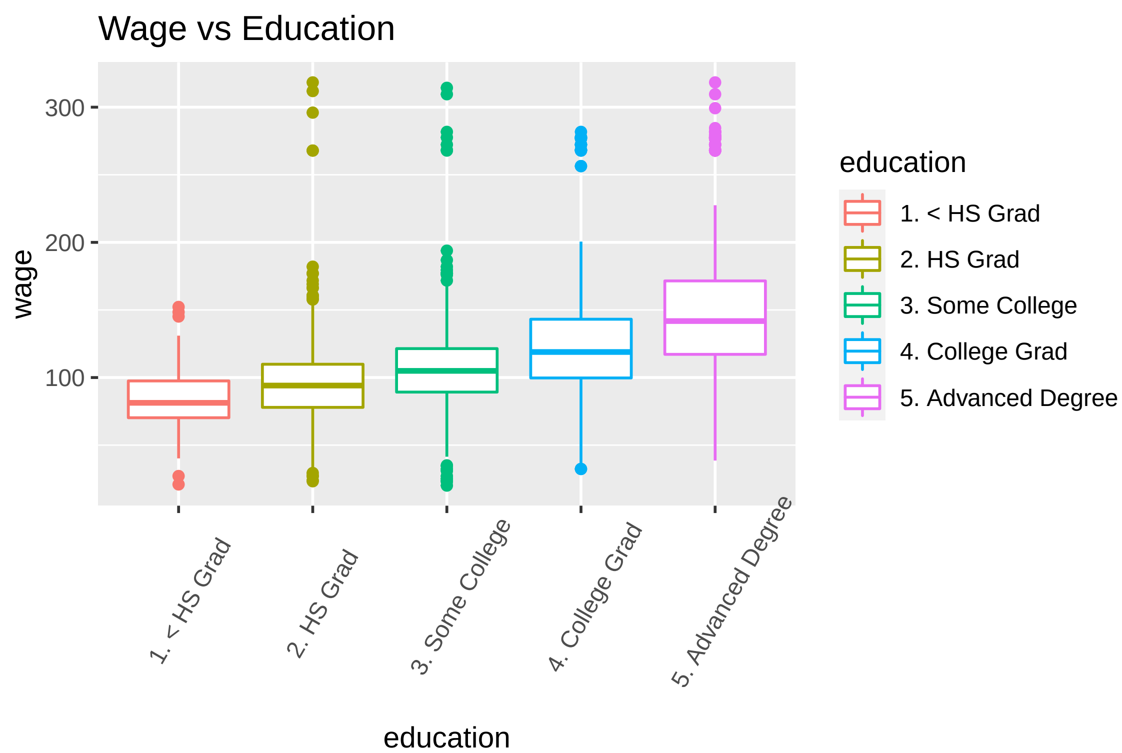
When visualizing the maritl variable for wage, it can be seen that the married category is the highest and that outliers exist.

```
# box plot race variable
ggplot(data = Wage, aes(x = race, y = wage, color = race)) + geom_boxplot() + ggtitle("Wage vs Race")
```



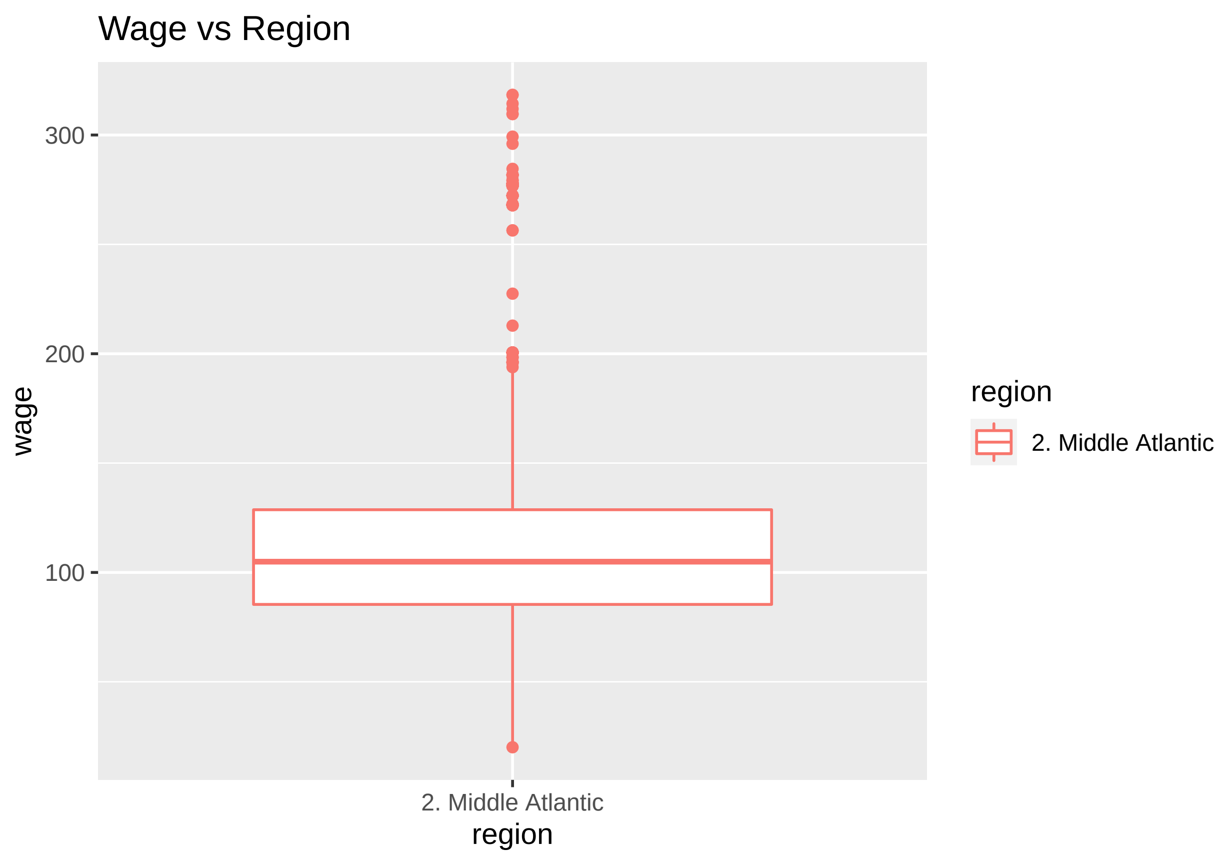
When visualizing the race variable for wage, it can be seen that the asian category is the highest and that outliers exist.

```
# boxplot eductaion variable
ggplot(data = Wage, aes(x = education, y = wage, color = education)) + geom_boxplot() +
    ggtitle("Wage vs Education") + theme(axis.text.x = element_text(angle = 60, hjust = 0.35,
    vjust = 0.5))
```



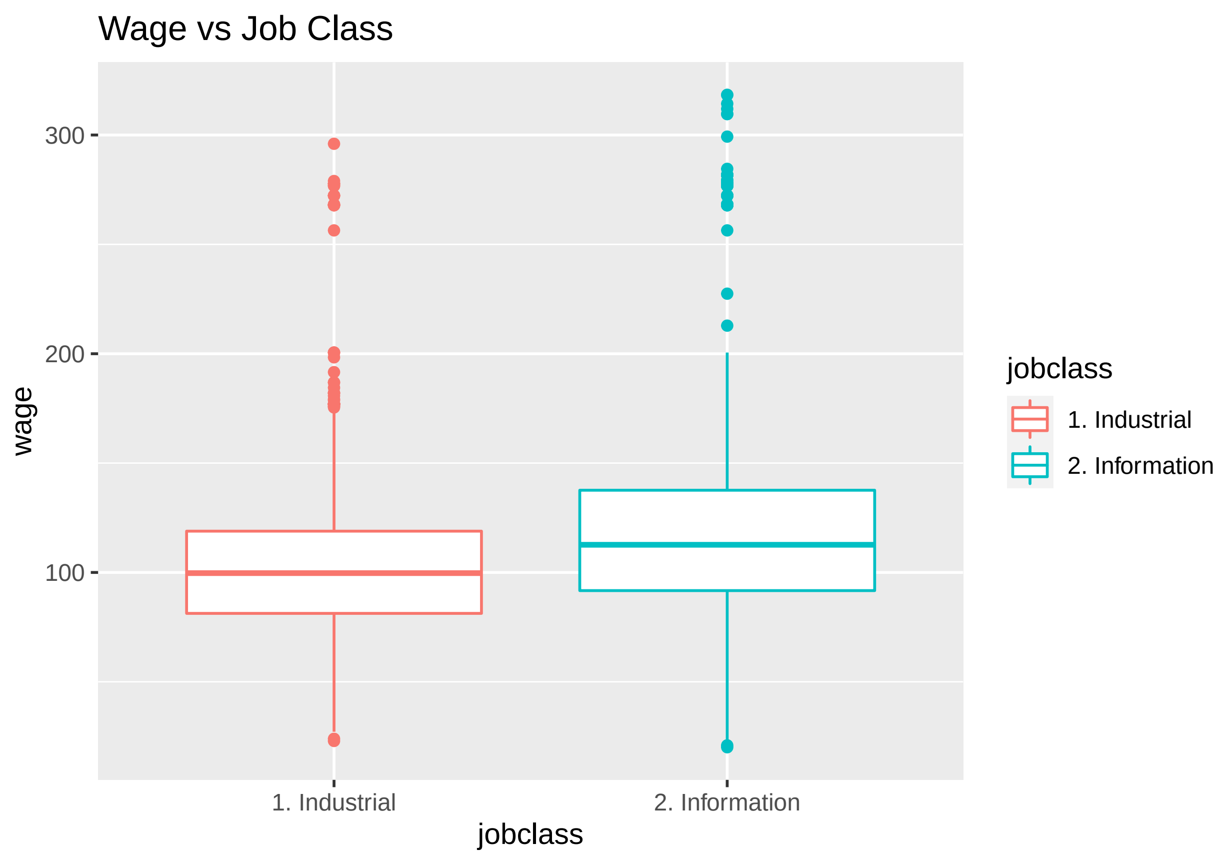
When visualizing the education variable for wage, it can be seen that the advanced degree category is the highest and that outliers exist.

```
# boxplot region variable
ggplot(data = Wage, aes(x = region, y = wage, color = region)) + geom_boxplot() +
    ggtitle("Wage vs Region")
```



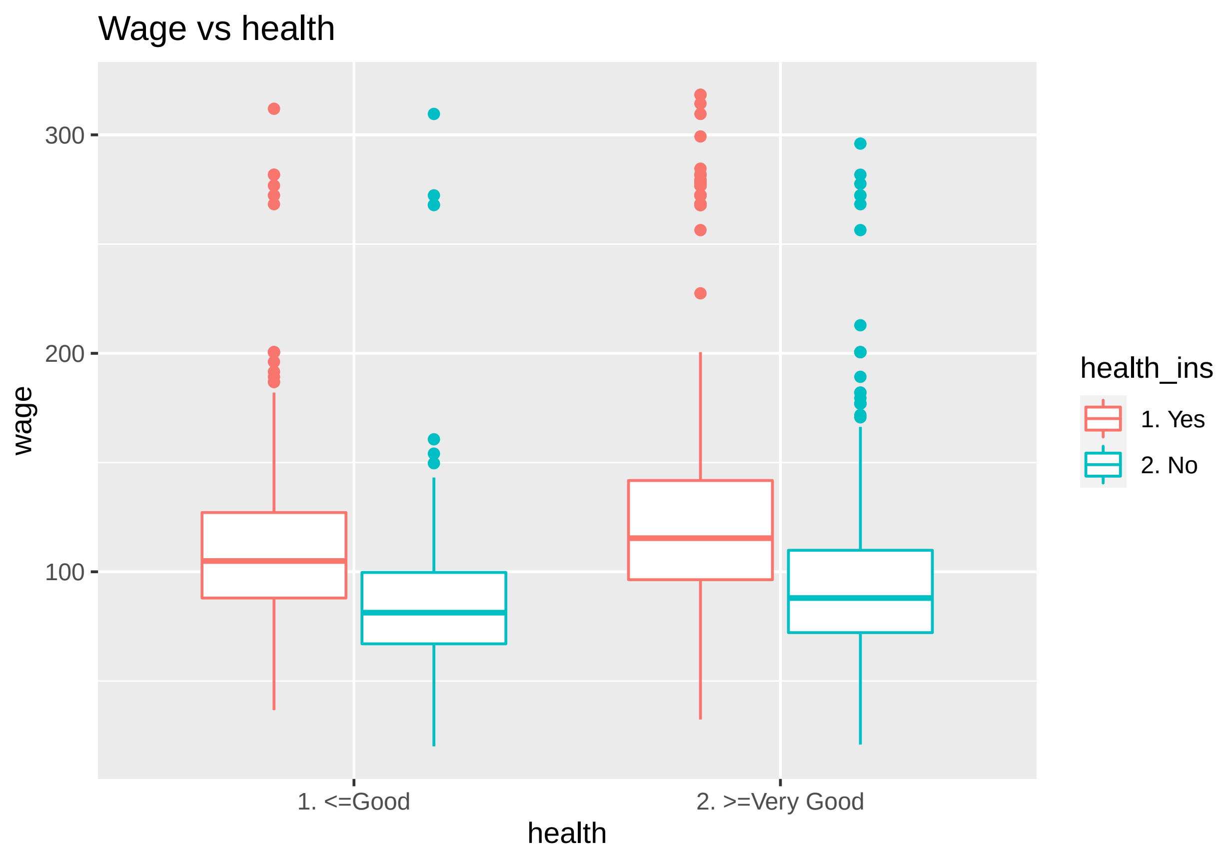
Visualizing the regional variable for wages shows that this variable is not suitable for modeling, given that only Middle Atlantic exists.

```
# box plot job class variable
ggplot(data = Wage, aes(x = jobclass, y = wage, color = jobclass)) + geom_boxplot() +
    ggtitle("Wage vs Job Class")
```



When visualizing the job class variable for wage, it can be seen that the information category is more higher than industrial and that outliers exist.

```
# box plot health and health insurance variables
ggplot(data = Wage, aes(x = health, y = wage, color = health_ins)) + geom_boxplot() +
    ggtitle("Wage vs health")
```



When visualizing the health and health insurance variables for wage, It can be seen that having insurance is higher in wages than not having insurance and that outliers exist.

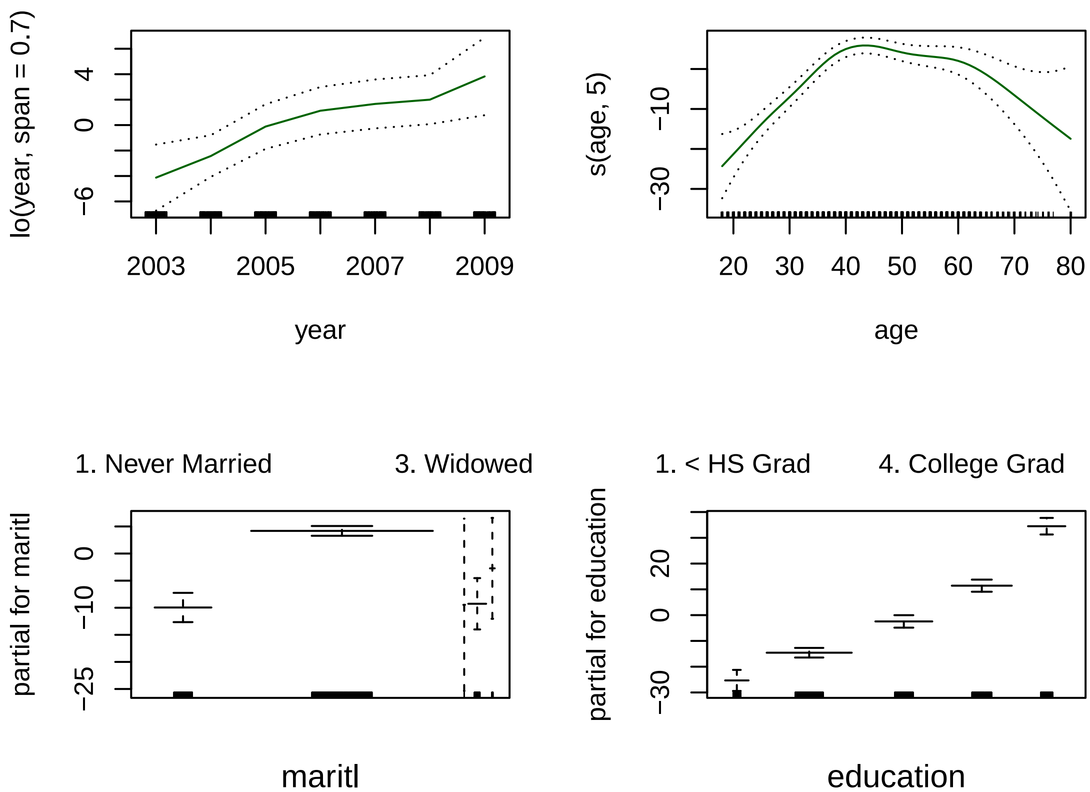
```
# fit a GAM
gam.m1 <- gam(wage ~ lo(year, span = 0.7) + s(age, 5), data = Wage)
gam.m2 <- gam(wage ~ lo(year, span = 0.7) + education, data = Wage)
gam.m3 <- gam(wage ~ lo(year, span = 0.7) + maritl, data = Wage)
gam.m4 <- gam(wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + education + jobclass,
    data = Wage)
gam.m5 <- gam(wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + race + education,
    data = Wage)
gam.m6 <- gam(wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + race + education +
    jobclass, data = Wage)
gam.m7 <- gam(wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + race + education +
    health, data = Wage)
gam.m8 <- gam(wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + race + education +
    health_ins, data = Wage)
gam.m9 <- gam(wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + race + education +
    health_ins + health + jobclass, data = Wage)
# compare the models
anova(gam.m1, gam.m2, gam.m3, gam.m4, gam.m5, gam.m6, gam.m7, gam.m8, gam.m9, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: wage ~ lo(year, span = 0.7) + s(age, 5)
## Model 2: wage ~ lo(year, span = 0.7) + education
```

```
## Model 3: wage ~ lo(year, span = 0.7) + maritl
## Model 4: wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + education +
##     jobclass
## Model 5: wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + race + education
## Model 6: wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + race + education +
##     jobclass
## Model 7: wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + race + education +
##     health
## Model 8: wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + race + education +
##     health_ins
## Model 9: wage ~ lo(year, span = 0.7) + s(age, 5) + maritl + race + education +
##     health_ins + health + jobclass
##   Resid. Df Resid. Dev       Df Deviance       F    Pr(>F)
## 1    2991.1    4739791
## 2    2992.1    3977177 -0.99896   762615
## 3    2992.1    4831543  0.00000  -854367
## 4    2982.1    3583675  9.99896  1247869 109.711 < 2.2e-16 ***
## 5    2980.1    3588839  2.00000    -5164
## 6    2979.1    3572897  1.00000    15942  14.015 0.0001848 ***
## 7    2979.1    3557066  0.00000    15830
## 8    2979.1    3417969  0.00000   139097
## 9    2977.1    3386543  2.00000    31427  13.814 1.068e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When the model is fitted and the model is compared, the **fourth model** is judged to be the best fit. Also, the evidence that there is a nonlinear relationship with the response is **age**.

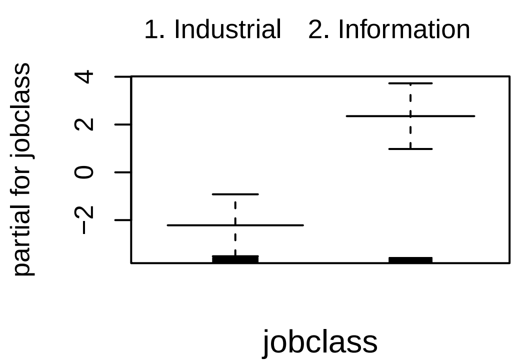
```r
# split plot 2, 2
par(mfrow = c(2, 2))
# show best model plots
plot(gam.m4, se = TRUE, col = "darkgreen")
```




```r
# summary model
summary(gam.m4)
```

```
##
## Call: gam(formula = wage ~ lo(year, span = 0.7) + s(age, 5) + maritl +
##     education + jobclass, data = Wage)
## Deviance Residuals:
##      Min       1Q   Median      3Q     Max
## -108.684  -19.523   -2.588  13.788 213.278
##
## (Dispersion Parameter for gaussian family taken to be 1201.729)
##
##     Null Deviance: 5222086 on 2999 degrees of freedom
## Residual Deviance: 3583675 on 2982.099 degrees of freedom
## AIC: 29808.03
##
## Number of Local Scoring Iterations: 1
##
## Anova for Parametric Effects
##                          Df  Sum Sq Mean Sq F value    Pr(>F)
## lo(year, span = 0.7)    1.0   26181   26181  21.786 3.182e-06 ***
## s(age, 5)               1.0  195235  195235 162.462 < 2.2e-16 ***
```

```
## maritl                  4.0  157721   39430  32.811 < 2.2e-16 ***
## education               4.0 1038115  259529 215.963 < 2.2e-16 ***
## jobclass                1.0   14059   14059  11.699 0.0006338 ***
## Residuals            2982.1 3583675    1202
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                      Npar Df  Npar F     Pr(F)
## (Intercept)
## lo(year, span = 0.7)     1.9  0.7287    0.4762
## s(age, 5)                4.0 18.4105 5.995e-15 ***
## maritl
## education
## jobclass
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



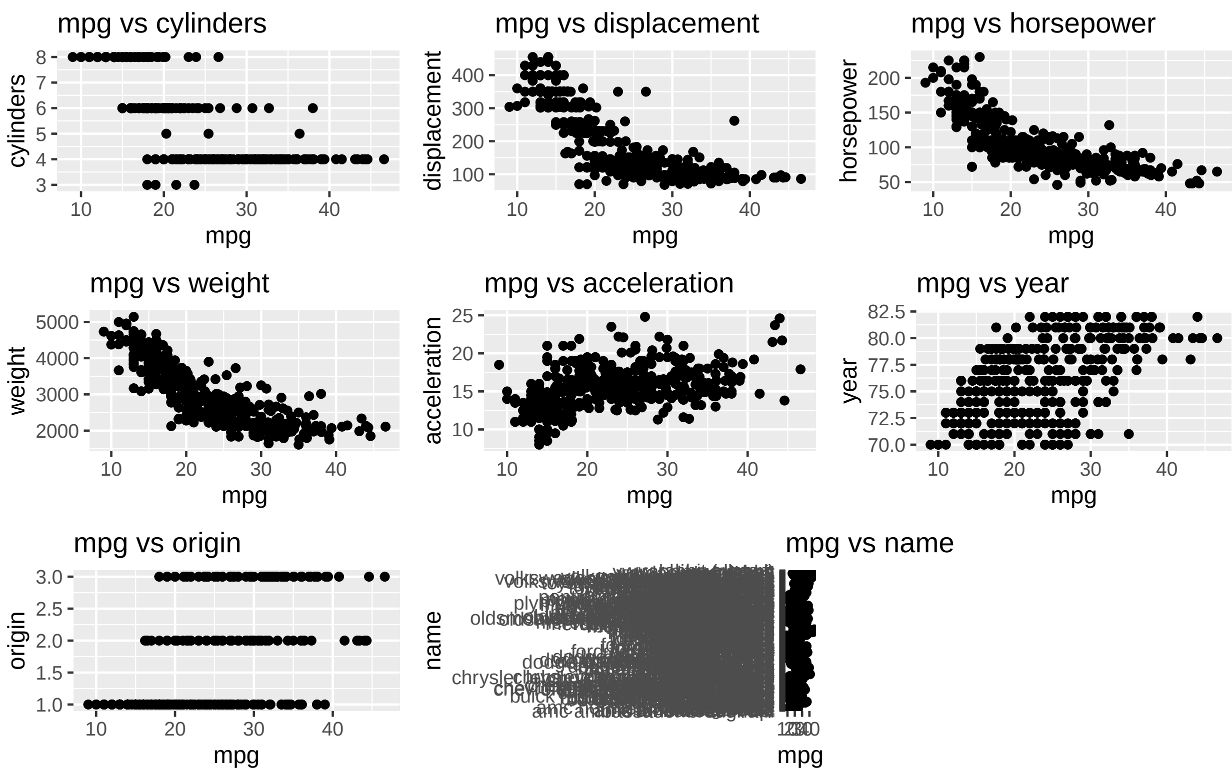
When checking the results of the fourth model in the figure, it can be seen that the **higher the education level**, the higher the wage, the higher the wage for **married people**, and the higher the wage for those in their **40s**. It can also be seen that **information** wages are higher than industries.

Q2. Question 8 of Chapter 7 of the ISLR book. (Page 299).

```r
# Auto dataset into Auto variable
Auto <- ISLR::Auto
```

Fit some of the non-linear models investigated in this chapter to the Auto data set. Is there evidence for non-linear relationships in this data set? Create some informative plots to justify your answer.

```r
# show all variables using plots (mpg) mpg vs cylinders
p1 <- ggplot(Auto, aes(x = mpg, y = cylinders)) + geom_point() + ggtitle("mpg vs cylinders")
# mpg vs displacement
p2 <- ggplot(Auto, aes(x = mpg, y = displacement)) + geom_point() + ggtitle("mpg vs displacement")
# mpg vs horsepower
p3 <- ggplot(Auto, aes(x = mpg, y = horsepower)) + geom_point() + ggtitle("mpg vs horsepower")
# mpg vs weight
p4 <- ggplot(Auto, aes(x = mpg, y = weight)) + geom_point() + ggtitle("mpg vs weight")
# mpg vs acceleration
p5 <- ggplot(Auto, aes(x = mpg, y = acceleration)) + geom_point() + ggtitle("mpg vs acceleration")
# mpg vs year
p6 <- ggplot(Auto, aes(x = mpg, y = year)) + geom_point() + ggtitle("mpg vs year")
# mpg vs origin
p7 <- ggplot(Auto, aes(x = mpg, y = origin)) + geom_point() + ggtitle("mpg vs origin")
# mpg vs name
p8 <- ggplot(Auto, aes(x = mpg, y = name)) + geom_point() + ggtitle("mpg vs name")
# shows several graphs on the screen
grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, nrow = 3, ncol = 3)
```



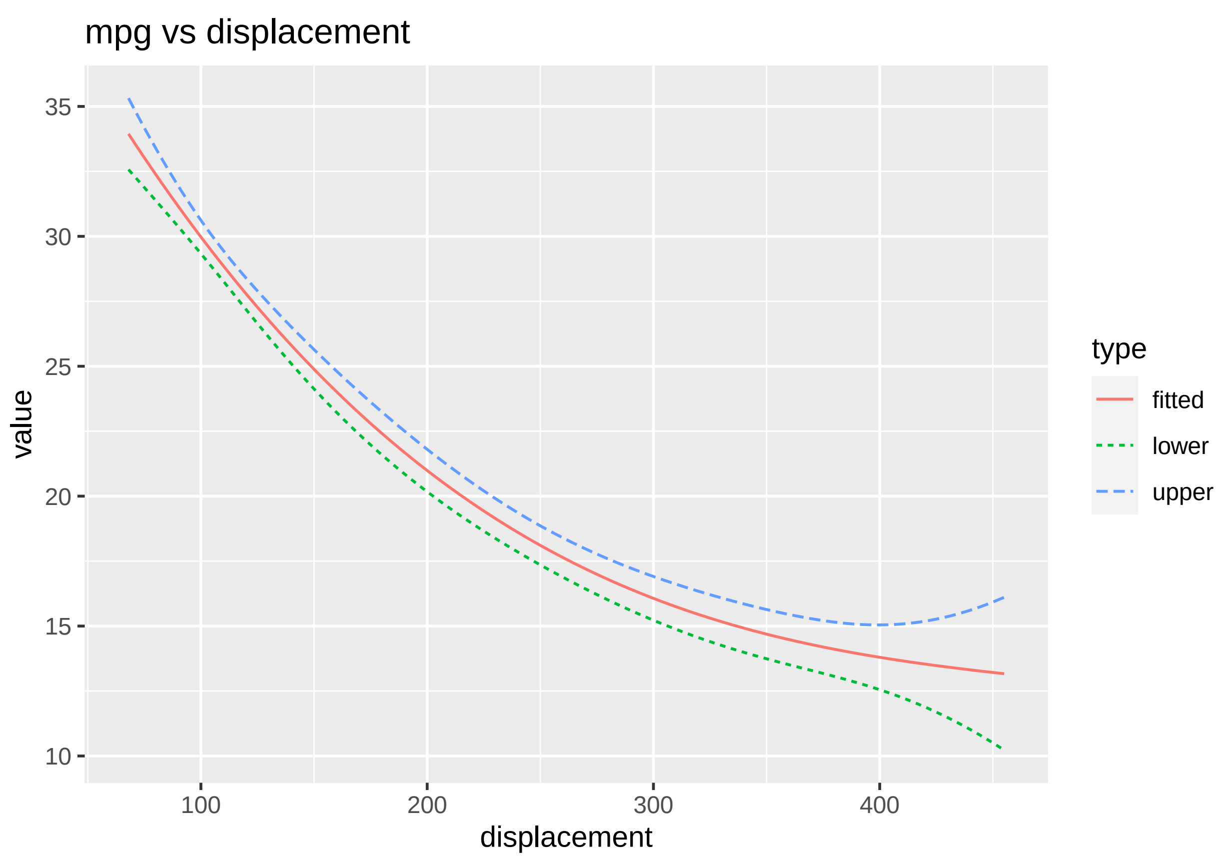
When checking all variables for mpg, it can be seen that the **displacement, horsepower, weight, acceleration** variables are nonlinear. Therefore, I will fit these four variables into a flexible model.

```r
# polynomial regression
fit.disp = lm(mpg~poly(displacement,3), data=Auto)
# summary of fit.disp
summary(fit.disp)
```

```
##
## Call:
## lm(formula = mpg ~ poly(displacement, 3), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.6791  -2.3900  -0.2987   2.1156  20.4528
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             23.4459     0.2205 106.350  < 2e-16 ***
## poly(displacement, 3)1 -124.2585     4.3649 -28.468  < 2e-16 ***
## poly(displacement, 3)2   31.0895     4.3649   7.123 5.18e-12 ***
## poly(displacement, 3)3   -4.4655     4.3649  -1.023    0.307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.365 on 388 degrees of freedom
## Multiple R-squared:  0.6896, Adjusted R-squared:  0.6872
## F-statistic: 287.4 on 3 and 388 DF,  p-value: < 2.2e-16
```

```
# make predictions
displims = range(Auto$displacement)
disp.grid = seq(displims[1],displims[2],length.out = 200)
preds.disp = predict(fit.disp, newdata = list(displacement=disp.grid),se=T)
# make 95% confidence intervals for predictions
fitted = preds.disp$fit
lower = fitted-2*preds.disp$se.fit
upper=fitted+2*preds.disp$se.fit
#visualize the model + confidence intervals (displacement)
df=data.frame(value=c(fitted,lower,upper), displacement=rep(disp.grid,3),
              type=c(rep("fitted",length(disp.grid)),
                     rep("lower",length(disp.grid)),

                     rep("upper",length(disp.grid))))
ggplot(data=df, aes(x=displacement,y=value,color=type, linetype= type))+geom_line() + ggtitle("mpg vs di
```



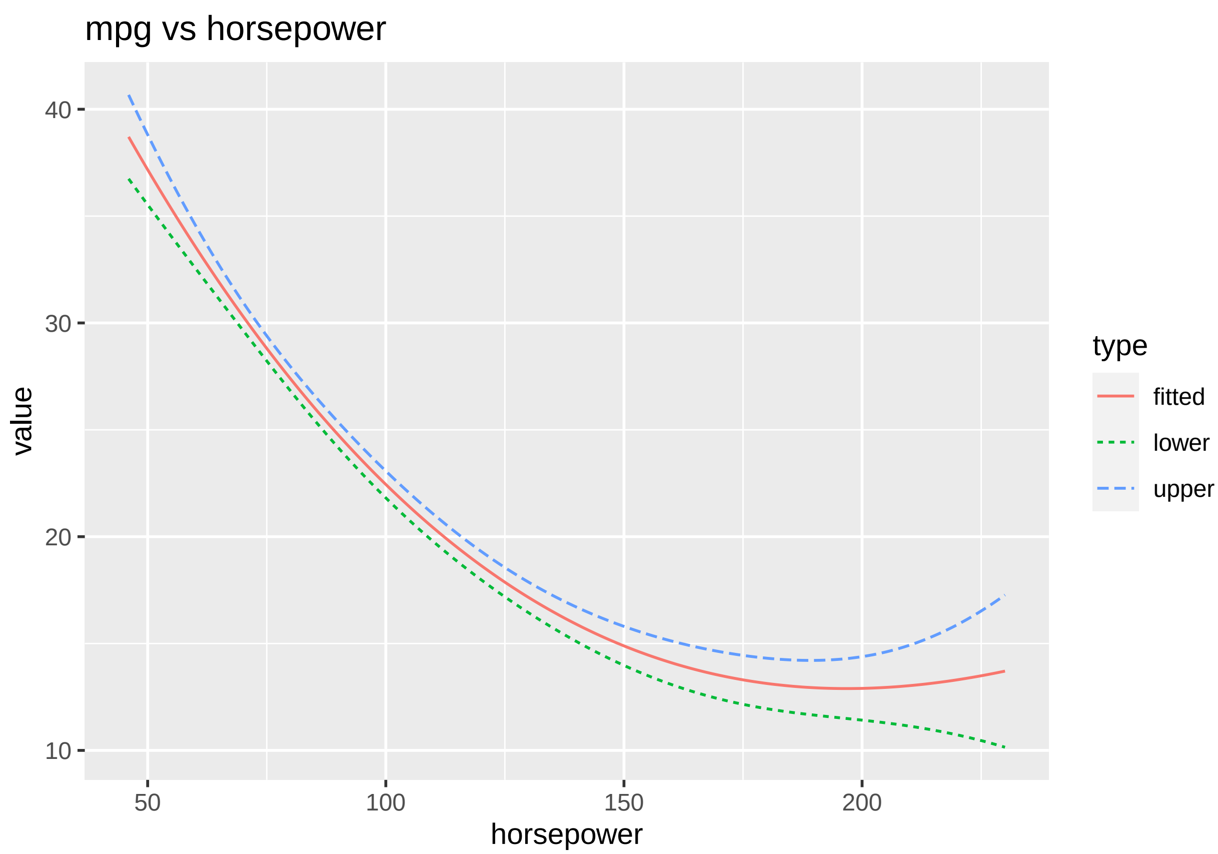

```
# polynomial regression
fit.horse = lm(mpg~poly(horsepower,3), data=Auto)
# summary horsepower
summary(fit.horse)
```

```
##
## Call:
## lm(formula = mpg ~ poly(horsepower, 3), data = Auto)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.7039  -2.4491  -0.1519   2.2035  15.8159
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             23.446      0.221 106.105   <2e-16 ***
## poly(horsepower, 3)1 -120.138      4.375 -27.460   <2e-16 ***
## poly(horsepower, 3)2   44.090      4.375  10.078   <2e-16 ***
## poly(horsepower, 3)3   -3.949      4.375  -0.903    0.367
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.375 on 388 degrees of freedom
## Multiple R-squared:  0.6882, Adjusted R-squared:  0.6858
## F-statistic: 285.5 on 3 and 388 DF,  p-value: < 2.2e-16
```

```r
# make predictions
hlims = range(Auto$horsepower)
h.grid = seq(hlims[1],hlims[2],length.out = 200)
preds.h = predict(fit.horse, newdata = list(horsepower=h.grid),se=T)
# make 95% confidence intervals for predictions
fitted = preds.h$fit
lower = fitted-2*preds.h$se.fit
upper=fitted+2*preds.h$se.fit
# visualize the model and confidence intervals (horsepower)
df1=data.frame(value=c(fitted,lower,upper), horsepower=rep(h.grid,3),
               type=c(rep("fitted",length(h.grid)),
                     rep("lower",length(h.grid)),

                     rep("upper",length(h.grid))))
ggplot(data=df1, aes(x=horsepower,y=value,color=type, linetype= type))+geom_line() +
  ggtitle("mpg vs horsepower")
```

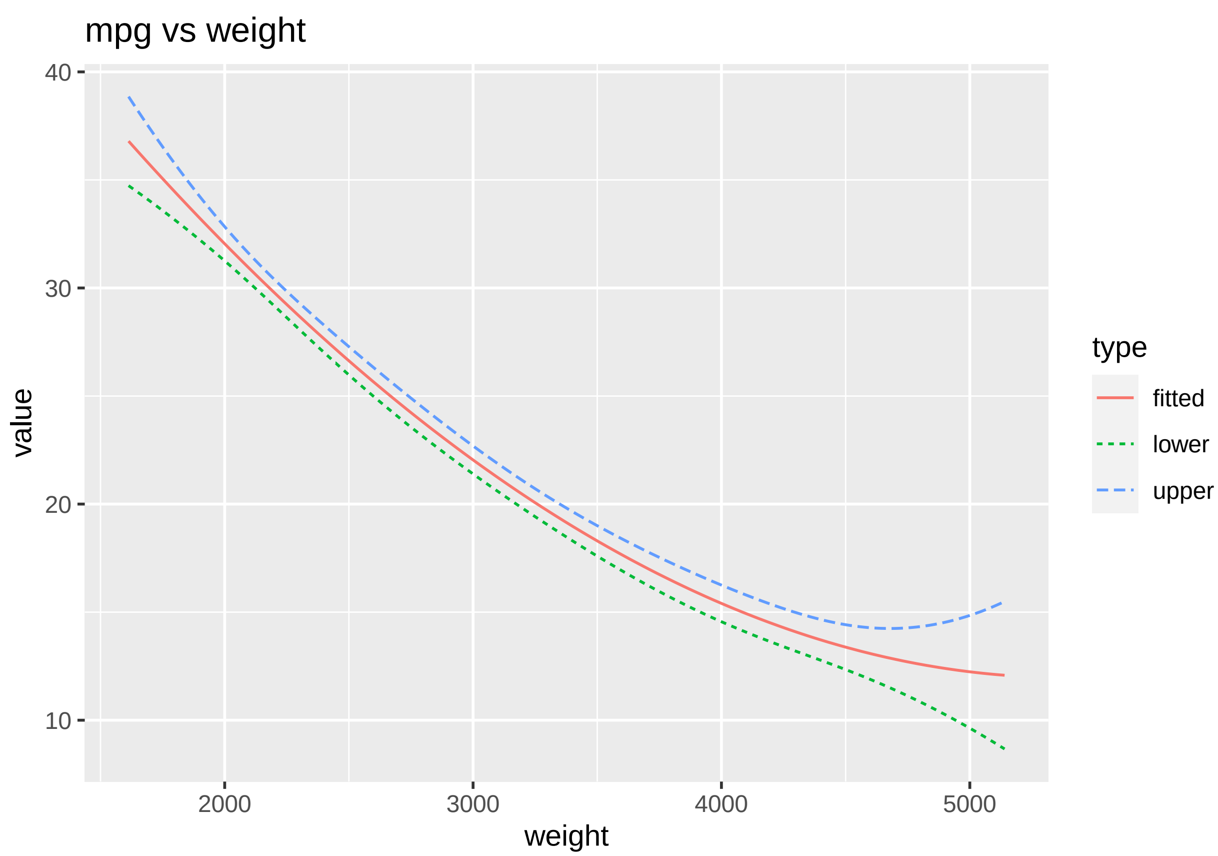



```
# polynomial regression
fit.w = lm(mpg~poly(weight,3), data=Auto)
# summary weight
summary(fit.w)
```

```
##
## Call:
## lm(formula = mpg ~ poly(weight, 3), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.6259  -2.7080  -0.3552   1.8385  16.0816
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          23.4459     0.2112 111.008  < 2e-16 ***
## poly(weight, 3)1 -128.4436     4.1817 -30.716  < 2e-16 ***
## poly(weight, 3)2   23.1589     4.1817   5.538 5.65e-08 ***
## poly(weight, 3)3    0.2204     4.1817   0.053    0.958
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.182 on 388 degrees of freedom
## Multiple R-squared:  0.7151, Adjusted R-squared:  0.7129
## F-statistic: 324.7 on 3 and 388 DF,  p-value: < 2.2e-16
```

```r
# make predictions
wlims = range(Auto$weight)
w.grid = seq(wlims[1],wlims[2],length.out = 200)
preds.w = predict(fit.w, newdata = list(weight=w.grid),se=T)
# make 95% confidence intervals for predictions
fitted = preds.w$fit
lower = fitted-2*preds.w$se.fit
upper=fitted+2*preds.w$se.fit
# viusalize the model and confidence intervals (weight)
df2=data.frame(value=c(fitted,lower,upper), weight=rep(w.grid,3),
              type=c(rep("fitted",length(w.grid)),
                     rep("lower",length(w.grid)),

                     rep("upper",length(w.grid))))
ggplot(data=df2, aes(x=weight,y=value,color=type, linetype= type))+geom_line() +
  ggtitle("mpg vs weight")
```



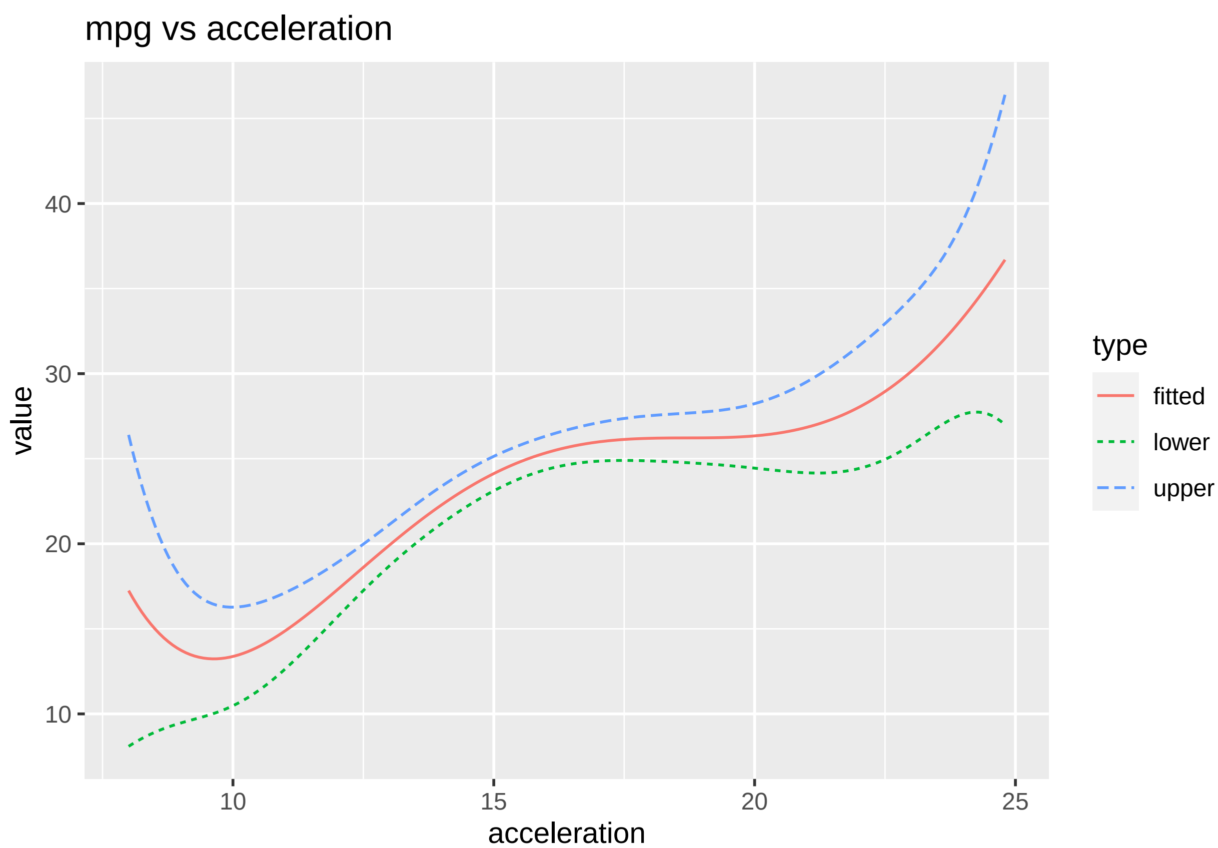

```r
# polynomial regression
fit.acc = lm(mpg~poly(acceleration,5), data=Auto)
# summary acceleration
summary(fit.acc)
```

```
##
## Call:
## lm(formula = mpg ~ poly(acceleration, 5), data = Auto)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.2209  -5.2976  -0.9565   4.7597  22.7506
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              23.4459     0.3516  66.689  < 2e-16 ***
## poly(acceleration, 5)1  65.3340     6.9608   9.386  < 2e-16 ***
## poly(acceleration, 5)2 -18.7482     6.9608  -2.693  0.00738 **
## poly(acceleration, 5)3   6.0643     6.9608   0.871  0.38418
## poly(acceleration, 5)4  20.7577     6.9608   2.982  0.00304 **
## poly(acceleration, 5)5  -5.3550     6.9608  -0.769  0.44218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.961 on 386 degrees of freedom
## Multiple R-squared:  0.2148, Adjusted R-squared:  0.2046
## F-statistic: 21.12 on 5 and 386 DF,  p-value: < 2.2e-16
```

```r
# make predictions
alims = range(Auto$acceleration)
a.grid = seq(alims[1],alims[2],length.out = 200)
preds.a = predict(fit.acc, newdata = list(acceleration=a.grid),se=T)
# make 95% confidence intervals for predictions
fitted = preds.a$fit
lower = fitted-2*preds.a$se.fit
upper=fitted+2*preds.a$se.fit
# visualize the model and confidence intervals (acceleration)
df3=data.frame(value=c(fitted,lower,upper), acceleration=rep(a.grid,3),
              type=c(rep("fitted",length(a.grid)),

                     rep("lower",length(a.grid)),
                     rep("upper",length(a.grid))))
ggplot(data=df3, aes(x=acceleration,y=value,color=type, linetype= type)) + geom_line() +
  ggtitle("mpg vs acceleration")
```




```r
# fit model using game
gam1 = gam(mpg ~ displacement + weight + acceleration + horsepower, data = Auto)
gam2 = gam(mpg ~ s(displacement, 3) + s(horsepower, 3) + s(weight, 3) + s(acceleration,
    5), data = Auto)
gam3 = gam(mpg ~ s(displacement, 5) + s(weight, 5) + s(acceleration, 5), data = Auto)
# compare models
anova(gam1, gam2, gam3, test = "F")
```

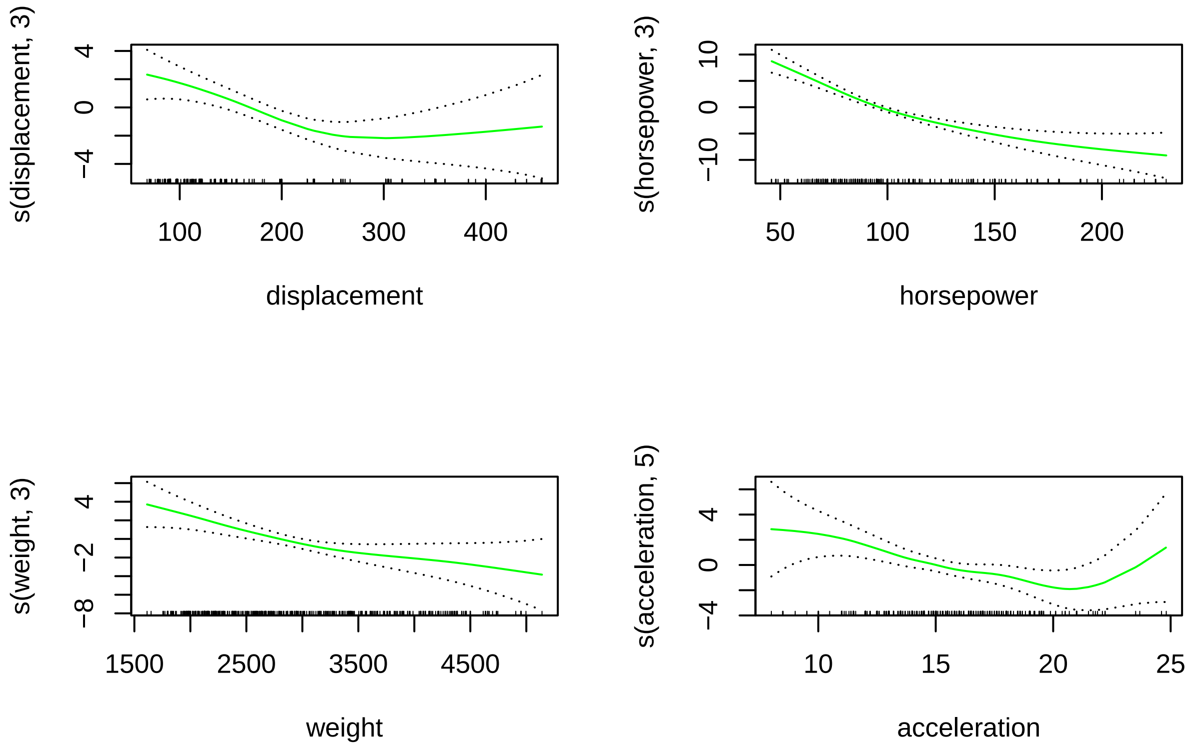
```
## Analysis of Deviance Table
##
## Model 1: mpg ~ displacement + weight + acceleration + horsepower
## Model 2: mpg ~ s(displacement, 3) + s(horsepower, 3) + s(weight, 3) +
##     s(acceleration, 5)
## Model 3: mpg ~ s(displacement, 5) + s(weight, 5) + s(acceleration, 5)
##   Resid. Df Resid. Dev       Df Deviance      F    Pr(>F)
## 1       387     6979.4
## 2       377     5489.6 10.00005  1489.78 9.1701 1.345e-13 ***
## 3       376     6108.5  0.99979  -618.84
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As a result of evaluating the modeling through these four variables, it can be seen that the **second model** is the most suitable.

```r
# summary best model
summary(gam2)
```

```
##
## Call: gam(formula = mpg ~ s(displacement, 3) + s(horsepower, 3) + s(weight,
##     3) + s(acceleration, 5), data = Auto)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4460  -2.2362  -0.3621   1.8938  16.0566
##
## (Dispersion Parameter for gaussian family taken to be 14.5614)
##
##     Null Deviance: 23818.99 on 391 degrees of freedom
## Residual Deviance: 5489.634 on 376.9999 degrees of freedom
## AIC: 2179.075
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                     Df  Sum Sq Mean Sq   F value    Pr(>F)
## s(displacement, 3)   1 15752.3 15752.3 1081.7879 < 2.2e-16 ***
## s(horsepower, 3)     1   841.9   841.9   57.8174 2.312e-13 ***
## s(weight, 3)         1   361.0   361.0   24.7882 9.756e-07 ***
## s(acceleration, 5)   1   131.0   131.0    8.9971  0.002884 **
## Residuals          377  5489.6    14.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                    Npar Df  Npar F     Pr(F)
## (Intercept)
## s(displacement, 3)       2  6.9997  0.001036 **
## s(horsepower, 3)         2 15.4745 3.479e-07 ***
## s(weight, 3)             2  2.5683  0.078005 .
## s(acceleration, 5)       4  2.0989  0.080354 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# visualize the best model
par(mfrow = c(2, 2))
plot(gam2, se = TRUE, col = "green")
```



Visualizing the second model shows that the higher the displacement and weight, the larger the mpg and the higher the acceleration, the larger the mpg.
As a result of summarizing the model, it can be seen that the displacement, horsepower are significant because the p-value value is lower than 0.05. On the contrary, weight and acceleration are not significant because p value more than 0.05 and there is no evidence of nonlinear effects, so it can be seen that there is a linear effect.
Thus, evidence of a non-linear relationship with the response are **displacement** and **horsepower**.

Q3. Question 9 of Chapter 7 of the ISLR book. (Page 299).

This question uses the variables dis (the weighted mean of distances to five Boston employment centers) and nox (nitrogen oxides concentration in parts per 10 million) from the Boston data. We will treat dis as the predictor and nox as the response.

```
# Boston dataset into bt
bt <- MASS::Boston
```

a. Use the poly() function to fit a cubic polynomial regression to predict nox using dis. Report the regression output, and plot the resulting data and polynomial fits.

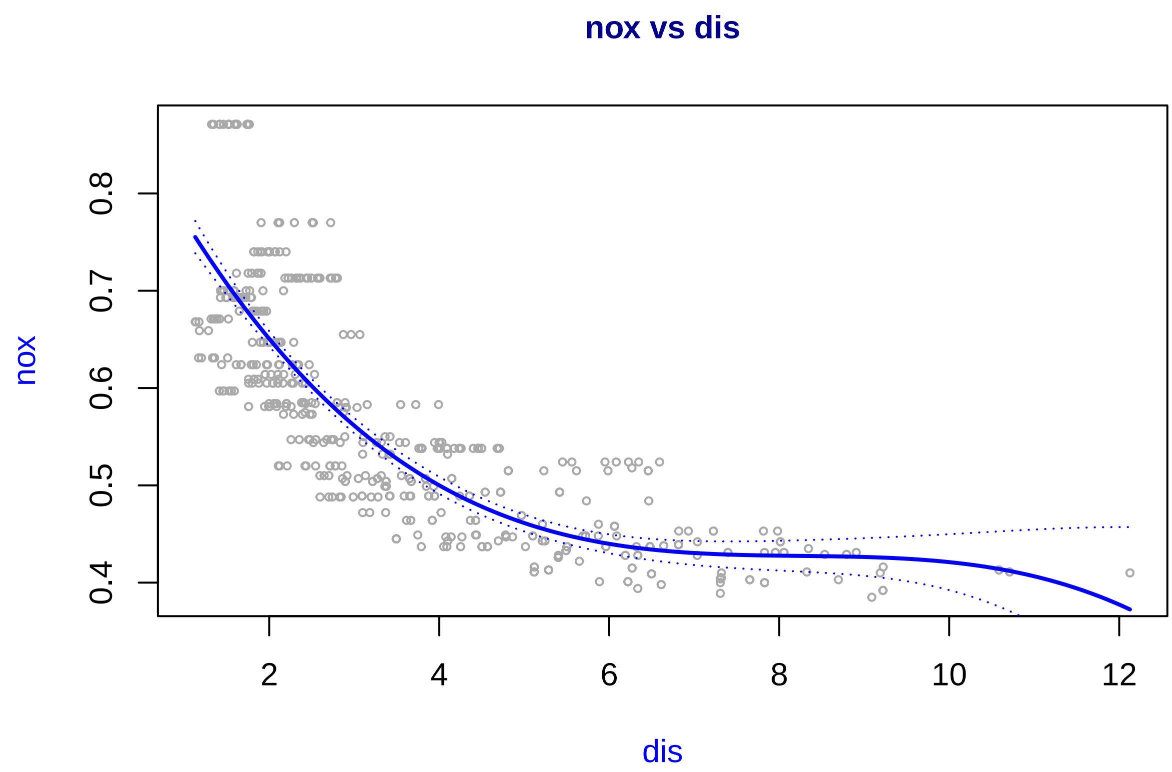
```
# polynomial regression
fitbt = lm(nox ~ poly(dis, 3), data = bt)
# summary predict nox using dis
summary(fitbt)
```

```
##
## Call:
## lm(formula = nox ~ poly(dis, 3), data = bt)
##
## 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
```

```
# make predictions
dislims = range(bt$dis)
dislims
```

```
## [1]  1.1296 12.1265
```

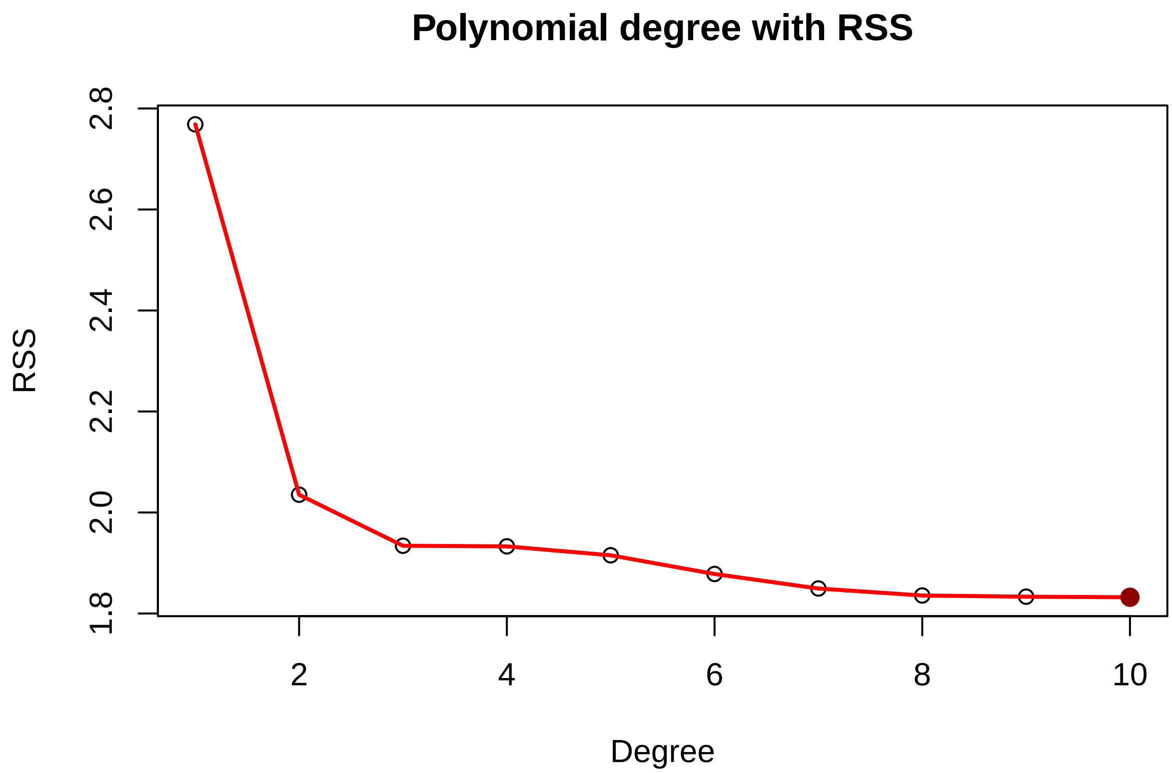
```
dis.grid = seq(dislims[1], dislims[2], length.out = 200)
predsbt = predict(fitbt, newdata = list(dis = dis.grid), se = T)
# make 95% confidence intervals for predictions
fitted = predsbt$fit
lower = fitted - 2 * predsbt$se.fit
upper = fitted + 2 * predsbt$se.fit
bands = cbind(upper, lower)
# visualize the model and confidence intervals
plot(x = bt$dis, y = bt$nox, xlim = dislims, cex = 0.5, col = "darkgrey", xlab = "dis",
    ylab = "nox", col.lab = "blue")
lines(dis.grid, fitted, lwd = 2, col = "blue")
matlines(dis.grid, bands, lwd = 1, col = "blue", lty = 3)
title(main = "nox vs dis", col.main = "darkblue", cex.main = 1)
```



As a result of spline suitability, it can be concluded that it is significant that most of the terms fit well, but there is a limit to the tail.

b. Plot the polynomial fits for a range of different polynomial degrees (say, from 1 to 10), and report the associated residual sum of squares.

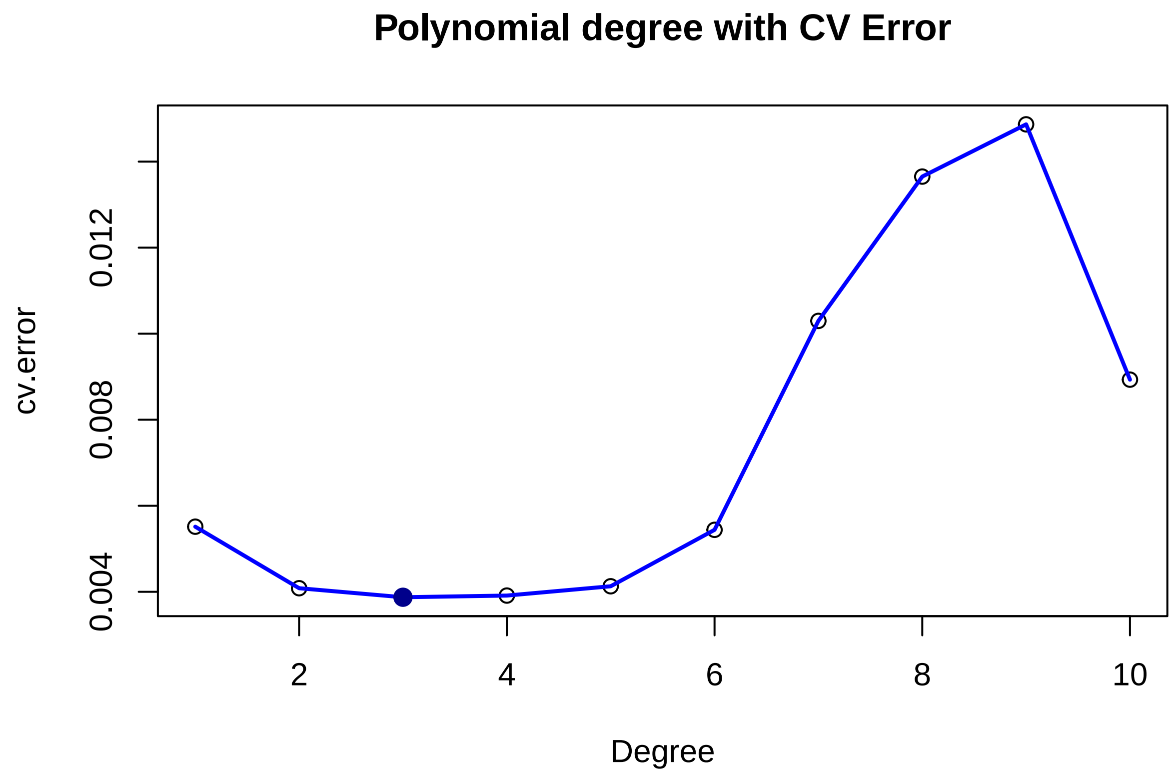
```
# set range 1 to 10
ran0 <- 1:10
# set rss polynomial degrees
rss1 <- rep(0, 10)
# polynomial fits from 1 to 10
for (i in 1:10) {
    fitPoly10 <- lm(nox ~ poly(dis, i), data = bt)
    rss1[i] <- sum(fitPoly10$residuals^2)  # compute rss
}
# show plot
plot(ran0, rss1, xlab = "Degree", ylab = "RSS")
lines(ran0, rss1, lwd = 2, col = "red")
title(main = "Polynomial degree with RSS")
points(which.min(rss1), rss1[which.min(rss1)], col = "darkred", pch = 20, cex = 1.8)
```



It can be seen that **10** degree is the minimum and It can be seen that RSS decreases as flexibility increases with polynomial degree.

c. Perform cross-validation or another approach to select the optimal degree for the polynomial, and explain your results.

```
# set seed
set.seed(123)
# set range 1 to 10
ran <- 1:10
# set mse 1 to 10
cv.error1 <- rep(0, 10)
for (i in 1:10) {
    fitPoly = glm(nox ~ poly(dis, i), data = bt)
    cv.error1[i] = cv.glm(bt, fitPoly, K = 10)$delta[1]  # compute mse
}
# show plot
plot(ran, cv.error1, xlab = "Degree", ylab = "cv.error")
lines(ran, cv.error1, lwd = 2, col = "blue")
title(main = "Polynomial degree with CV Error")
points(which.min(cv.error1), cv.error1[which.min(cv.error1)], col = "darkblue", pch = 20,
    cex = 1.8)
```



As a result of dividing the data into 10 segments for cross-validation, it can be seen that the selected fit is a spline with **3** degree of freedom.

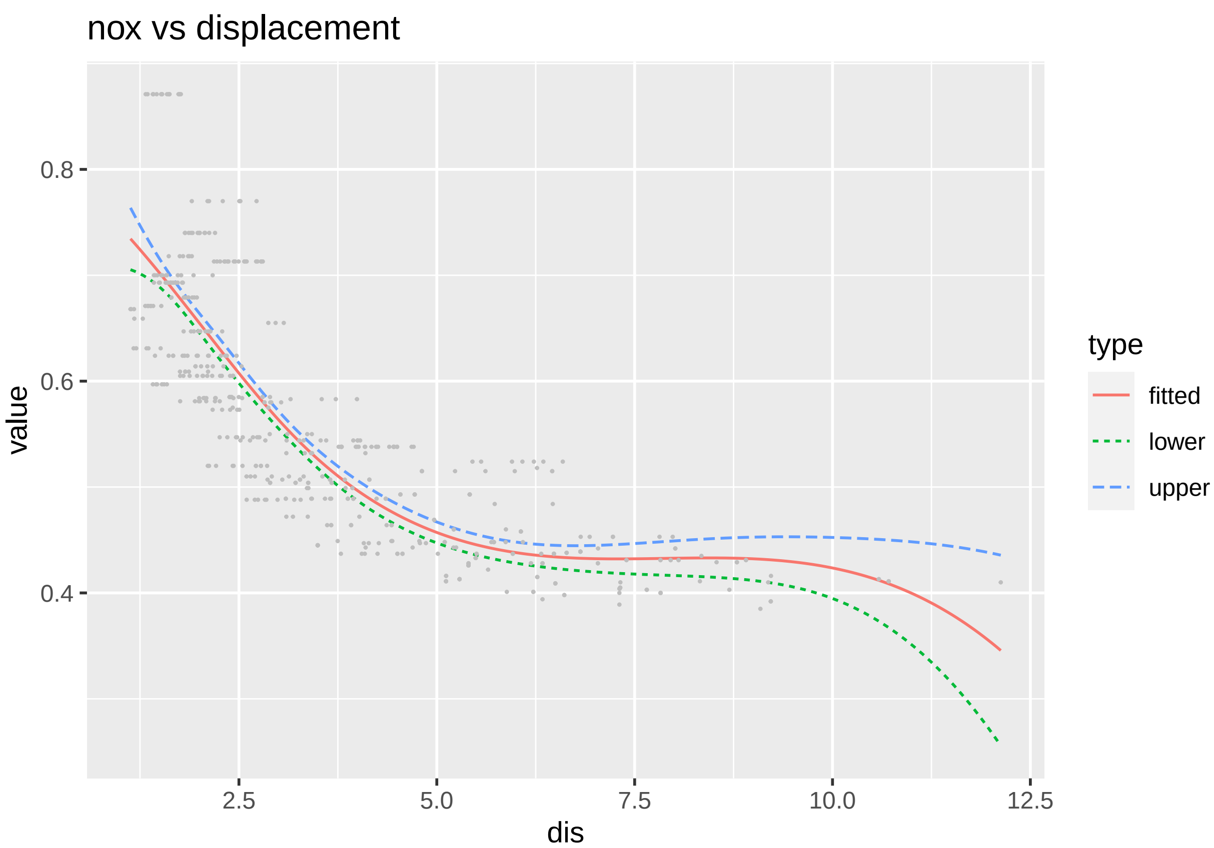
d. Use the bs() function to fit a regression spline to predict nox using dis. Report the output for the fit using four degrees of freedom. How did you choose the knots? Plot the resulting fit.

```
# bs() function to fit a regression spline four degrees of freedom
fitbs = lm(nox ~ bs(dis, df = 4), data = bt)
# show knots
attr(bs(bt$dis, df = 4), "knots")
```

```
##     50%
## 3.20745
```

```
# make 95% confidence intervals for predictions
dislims1 = range(bt$dis)
dis.grid1 = seq(dislims1[1], dislims1[2], length.out = 200)
predsDis = predict(fitbs, newdata = list(dis = dis.grid1), se = T)
fittedDis = predsDis$fit
lowerDis = fittedDis - 2 * predsDis$se.fit
upperDis = fittedDis + 2 * predsDis$se.fit
# visualize the model and confidence intervals
dfDis <- data.frame(value = c(fittedDis, lowerDis, upperDis), dis = rep(dis.grid,
```

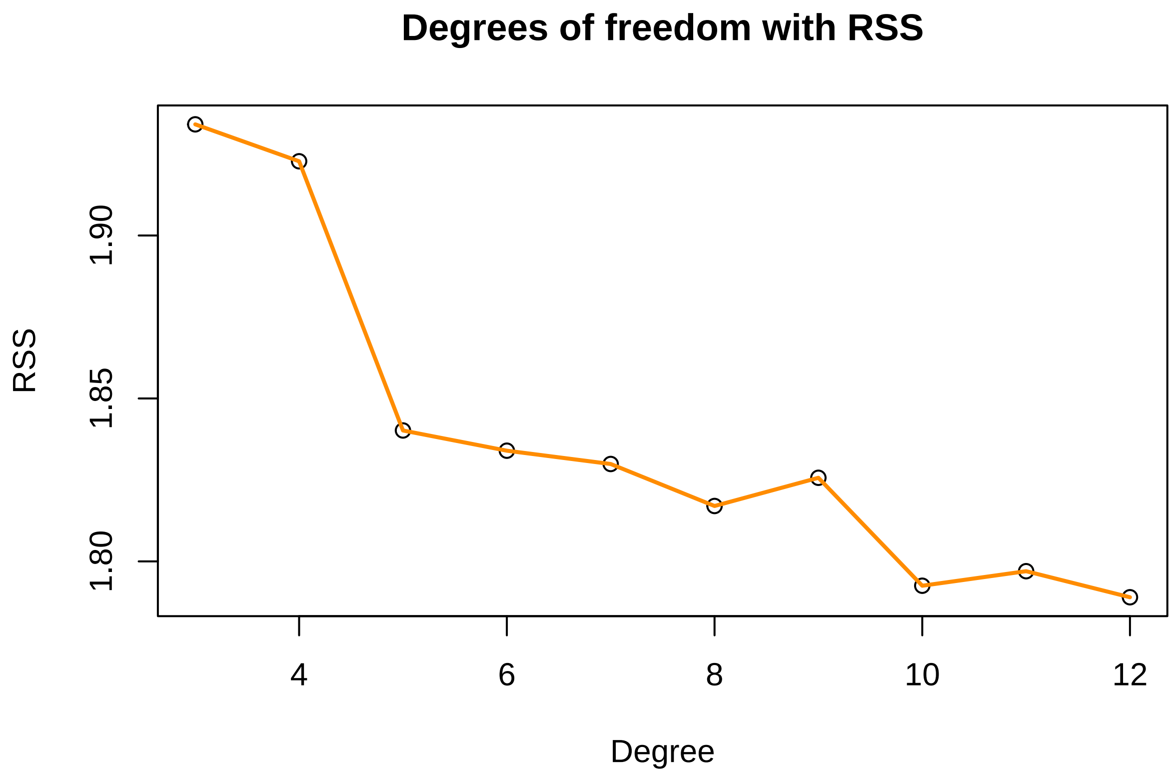
```
    3), type = c(rep("fitted", length(dis.grid1)), rep("lower", length(dis.grid1)),
    rep("upper", length(dis.grid1))))
ggplot() + geom_line(data = dfDis, aes(x = dis, y = value, color = type, linetype = type)) +
    geom_point(data = bt, aes(x = dis, y = nox), size = 0.1, colour = "grey") + ggtitle("nox vs displace
```



As a result of predicting nox using dis by fitting the regression spline using degree of freedom of 4 degree, **it can be seen that once the knots are 1, all terms of spline fit are significant**.

e. Now fit a regression spline for a range of degrees of freedom, and plot the resulting fits and report the resulting RSS. Describe the results obtained.

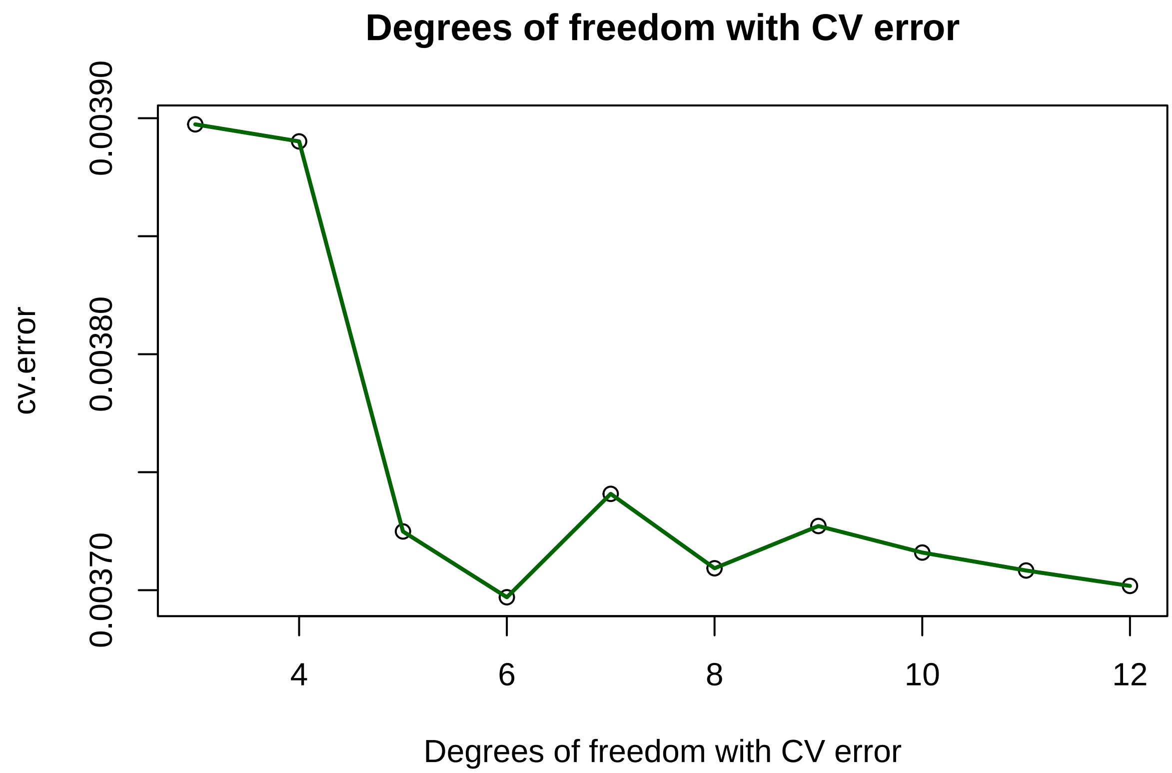
```
# set range 3 to 12
ran1 <- 3:12
# set rss range 1 to 10
rss1 <- rep(0, 10)
for (i in 3:12) {
    fitSpline <- lm(nox ~ bs(dis, df = i), data = bt)
    rss1[i - 2] <- sum(fitSpline$residuals^2)  # compute rss
}
# show plot
plot(ran1, rss1, xlab = "Degree", ylab = "RSS")
lines(ran1, rss1, lwd = 2, col = "darkorange")
title(main = "Degrees of freedom with RSS")
```



When the degree of freedom is set from 3 to 12, it can be seen that the RSS of degrees of freedom **12** is the lowest, and whenever additional degrees of freedom are allowed, the trend is not simple.

f. Perform cross-validation or another approach in order to select the best degrees of freedom for a regression spline on this data. Describe your results.

```
# set seed
set.seed(1)
# set range 3 to 12
ran2 <- 3:12
# set mse range 1 to 10 and regression spline
cv.error2 <- rep(0, 10)
for (i in 3:12) {
    fit.cross.spline = glm(nox ~ bs(dis, df = i), data = bt)
    cv.error2[i - 2] = cv.glm(bt, fit.cross.spline, K = 10)$delta[1]  # compute MSE
}
# show plot
plot(ran2, cv.error2, xlab = "Degrees of freedom with CV error", ylab = "cv.error")
lines(ran2, cv.error2, lwd = 2, col = "darkgreen")
title(main = "Degrees of freedom with CV error")
```



As a result of checking 10 degrees of freedom from 3 to 12 through repeated cross-validation, it can be seen that the selected fit is a spline with **6** degree of freedom.

Q4. Question 10 of Chapter 7 of the ISLR book. (Page 300).

This question relates to the College data set.

```
# College dataset into coll
coll <- ISLR::College
```

(a) Split the data into a training set and a test set. Using out-of-state tuition as the response and the other variables as the predictors, perform forward stepwise selection on the training set in order to identify a satisfactory model that uses just a subset of the predictors.

```
# set seed
set.seed(1)
# train range
train = sample(length(coll$Outstate), length(coll$Outstate)/2)
# set train
coll.train <- coll[train, ]
```

```r
# set test
coll.test <- coll[-train, ]
# perform forward stepwise selection on training
regfit.fwd = regsubsets(Outstate ~ ., data = coll.train, nvmax = 19, method = "forward")
# summary regfit.fwd into reg.summary
reg.summary = summary(regfit.fwd)
# find min show the all plots
which.min(reg.summary$cp)
```

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

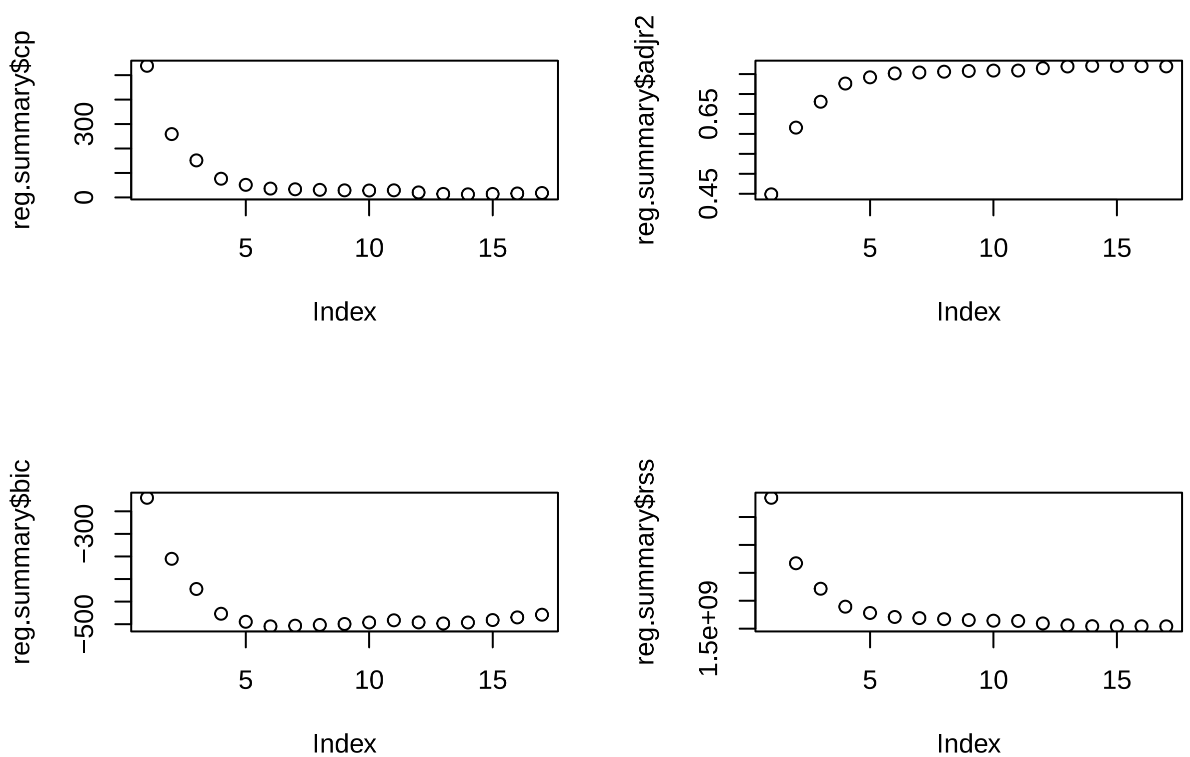
```r
which.max(reg.summary$adjr2)
```

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

```r
which.min(reg.summary$bic)
```

```
## [1] 6
```

```r
par(mfrow = c(2, 2))
plot(reg.summary$cp)
plot(reg.summary$adjr2)
plot(reg.summary$bic)
plot(reg.summary$rss)
```



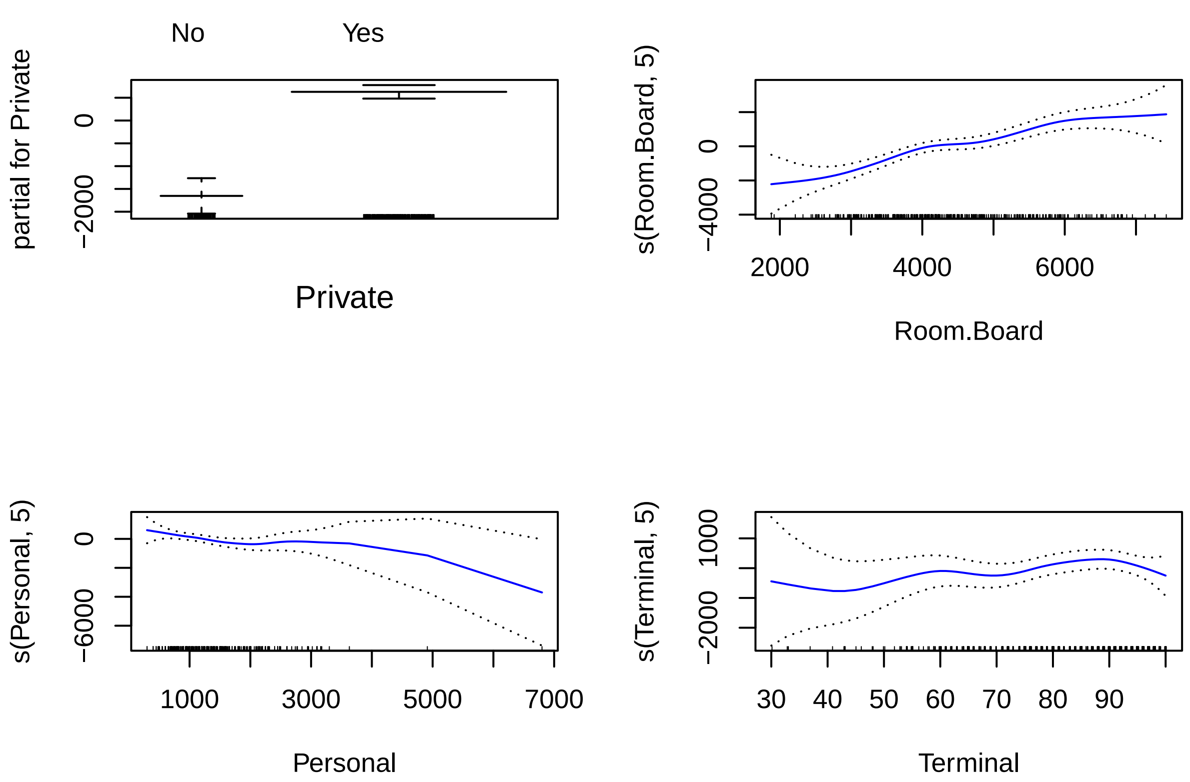
```r
# set model (min)
model1 <- coef(regfit.fwd, 6)
# check fit model names
names(model1)
```

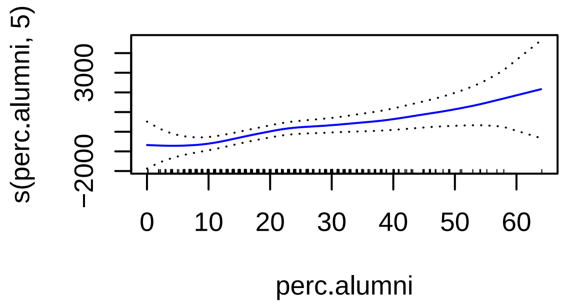
```
## [1] "(Intercept)" "PrivateYes"  "Room.Board"  "Terminal"    "perc.alumni"
## [6] "Expend"      "Grad.Rate"
```

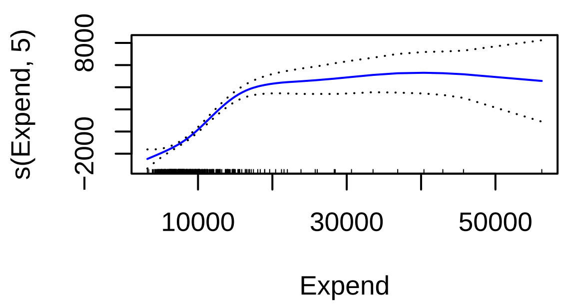
After dividing the data into training sets and test sets, as a result of forming a train with a forward stepwise selection, the optimal is set to **6** because the minimum of cp is 14, the maximum of adjr2 is 14, and the minimum of bic is 6.

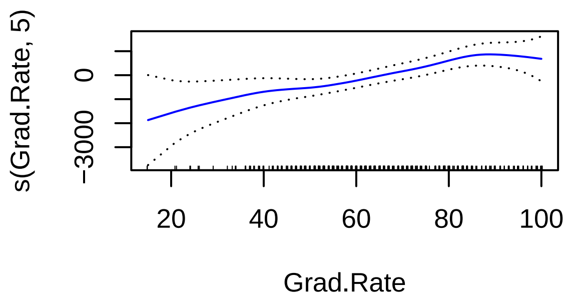
b. Fit a GAM on the training data, using out-of-state tuition as the response and the features selected in the previous step as the predictors. Plot the results, and explain your findings.

```r
# fit gam on the training data
gam.coll <- gam(Outstate ~ Private + s(Room.Board, 5) + s(Personal, 5) + s(Terminal,
    5) + s(perc.alumni, 5) + s(Expend, 5) + s(Grad.Rate, 5), data = coll.train)
# show plot the results
par(mfrow = c(2, 2))
plot(gam.coll, se = T, col = "blue")
```









As a result of fitting the GAM to the training data using the function selected in the previous step, clear evidence of the nonlinear effect of **Expend** can be seen.

c. Evaluate the model obtained on the test set, and explain the results obtained.

```
predGam <- predict(gam.coll, coll.test)
# compute err
err <- mean((coll.test$Outstate - predGam)^2)
# compute tss
tss <- mean((coll.test$Outstate - mean(coll.test$Outstate))^2)
# compute rss
rss <- 1 - err/tss
# show RSS
rss
```

```
## [1] 0.7650887
```

As a result of evaluating the model obtained from the test set, **it can be seen that R-square is well generalized to 0.7650887 (76.51 %)**.

d. For which variables, if any, is there evidence of a non-linear relationship with the response?

```r
# summary model
summary(gam.coll)
```

```
##
## Call: gam(formula = Outstate ~ Private + s(Room.Board, 5) + s(Personal,
##     5) + s(Terminal, 5) + s(perc.alumni, 5) + s(Expend, 5) +
##     s(Grad.Rate, 5), data = coll.train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -7144.18 -1059.38   -25.56  1234.44  6550.66
##
## (Dispersion Parameter for gaussian family taken to be 3615058)
##
##     Null Deviance: 6989966760 on 387 degrees of freedom
## Residual Deviance: 1286958128 on 355.9993 degrees of freedom
## AIC: 6992.74
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                    Df     Sum Sq    Mean Sq F value    Pr(>F)
## Private             1 1787717294 1787717294 494.520 < 2.2e-16 ***
## s(Room.Board, 5)    1 1620702516 1620702516 448.320 < 2.2e-16 ***
## s(Personal, 5)      1   77159748   77159748  21.344 5.373e-06 ***
## s(Terminal, 5)      1  267587898  267587898  74.020 2.508e-16 ***
## s(perc.alumni, 5)   1  308555955  308555955  85.353 < 2.2e-16 ***
## s(Expend, 5)        1  652820998  652820998 180.584 < 2.2e-16 ***
## s(Grad.Rate, 5)     1   73124483   73124483  20.228 9.317e-06 ***
## Residuals         356 1286958128    3615058
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                   Npar Df  Npar F     Pr(F)
## (Intercept)
## Private
## s(Room.Board, 5)        4  1.9151    0.1074
## s(Personal, 5)          4  0.9645    0.4270
## s(Terminal, 5)          4  1.6283    0.1665
## s(perc.alumni, 5)       4  0.4603    0.7649
## s(Expend, 5)            4 21.4769 6.661e-16 ***
## s(Grad.Rate, 5)         4  0.7352    0.5685
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As a result of summarizing the model, it can be seen that the Expand is significant because the p-value value is very lower than 0.05. On the contrary, Personal, Terminal, per.alumni, Grad.Rate, Room.Board are not significant and there is no evidence of nonlinear effects, so it can be seen that there is a linear effect.
Thus, evidence of a non-linear relationship with the response is **Expand**.