Stat 897 Fall 2017 Data Analysis Assignment 8

Penn State

Due October 22, 2017

In this assignment we again use the College data found in the ISLR library, with the 600 observation training set (using the rest as the test data).

(a) 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. Please set the seed at 801 once at the beginning before choosing the training set.

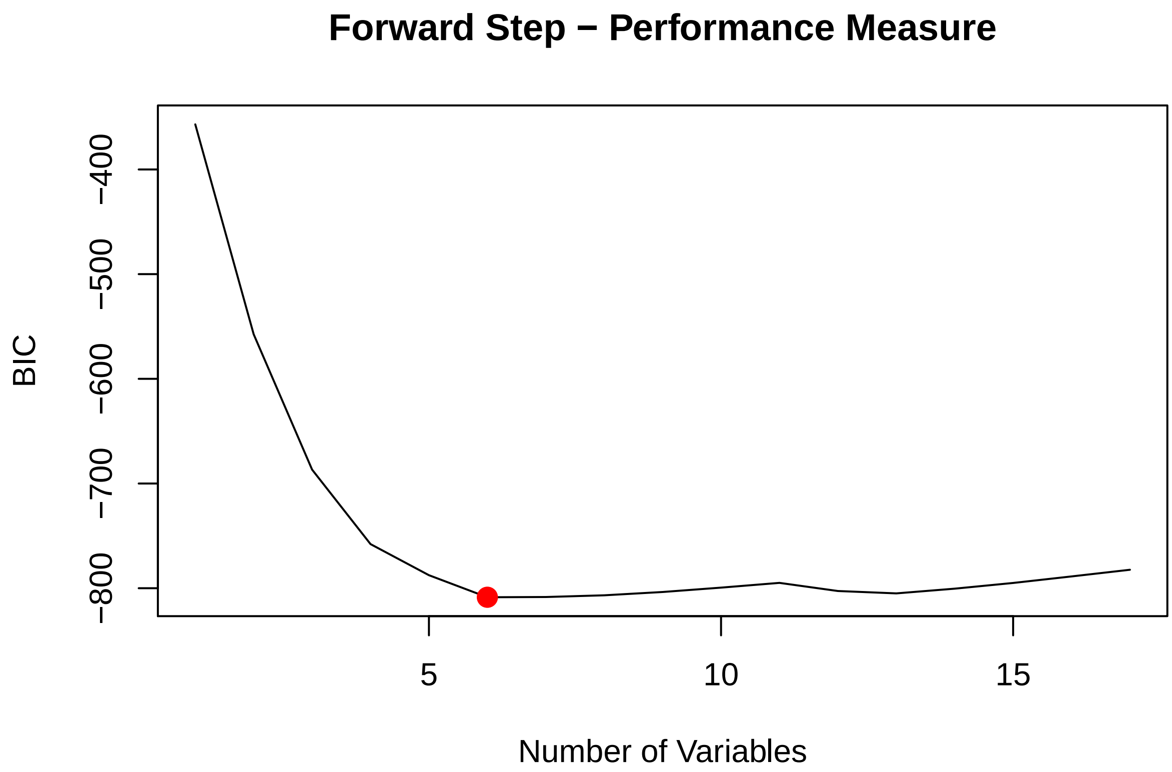
```
library(leaps)
library(ISLR)
#install.packages('ISLR')
library(gam)
```

```
## Loading required package: splines
## Loading required package: foreach
## Loaded gam 1.14-4
```

```
data("College")
attach(College)

set.seed (801)
trainingRows=sample (nrow(College), 600, replace = FALSE)
train = College[trainingRows,]
test = College[-trainingRows,]

regfit.fwd=regsubsets (Outstate~.,data=train, nvmax =17, method='forward')
reg.summary = summary (regfit.fwd)
plot(reg.summary$bic, xlab ="Number of Variables",ylab="BIC", type = 'l', main = 'Forward Step - Perfor
which.min (reg.summary$bic )
```

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

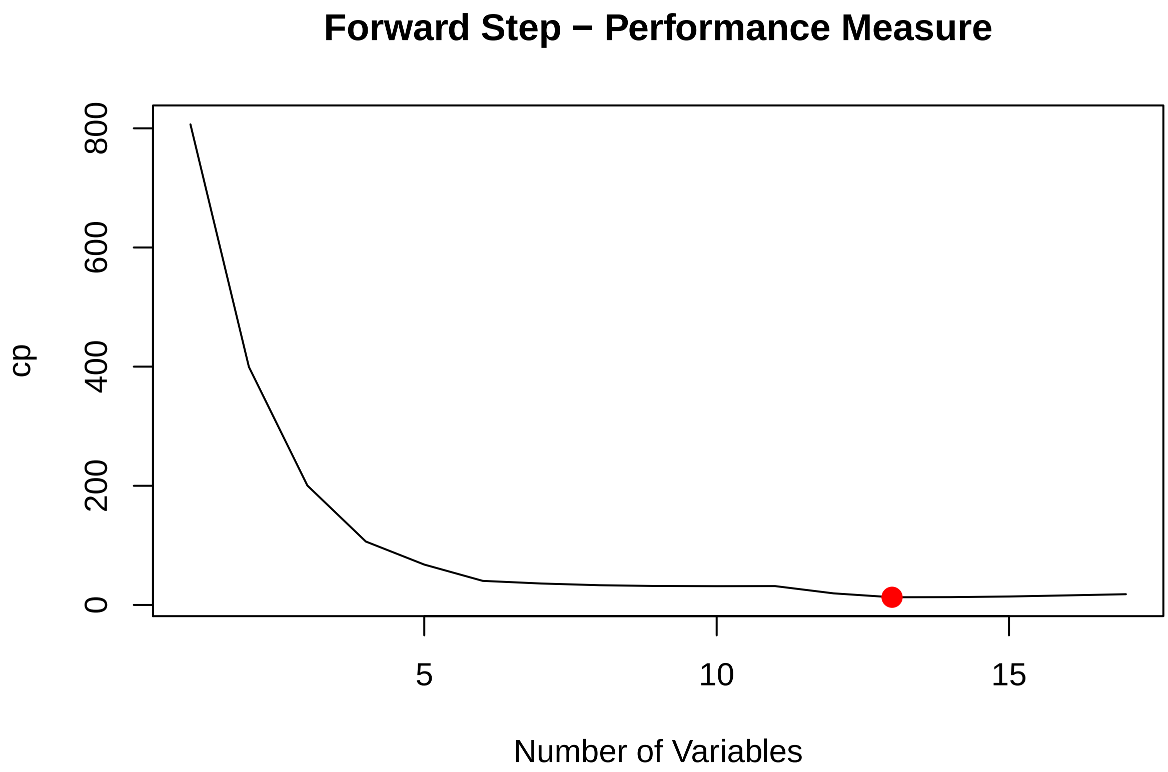
```
points (which.min (reg.summary$bic ), reg.summary$bic[which.min (reg.summary$bic )], col ="red",cex =2,
```




```r
plot(reg.summary$cp, xlab ="Number of Variables",ylab="cp", type = 'l', main = 'Forward Step - Performan
which.min (reg.summary$cp )
```

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

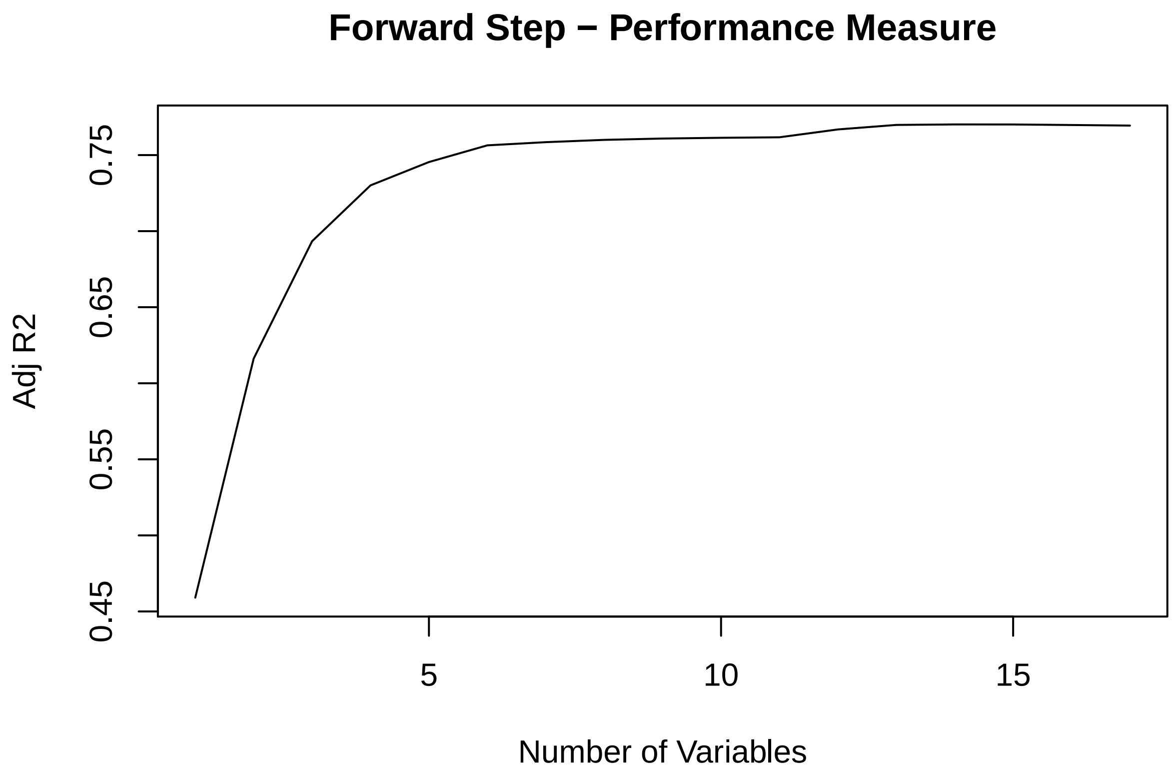
```r
points (which.min (reg.summary$cp ), reg.summary$cp[which.min (reg.summary$cp )], col ="red",cex =2, pch
```



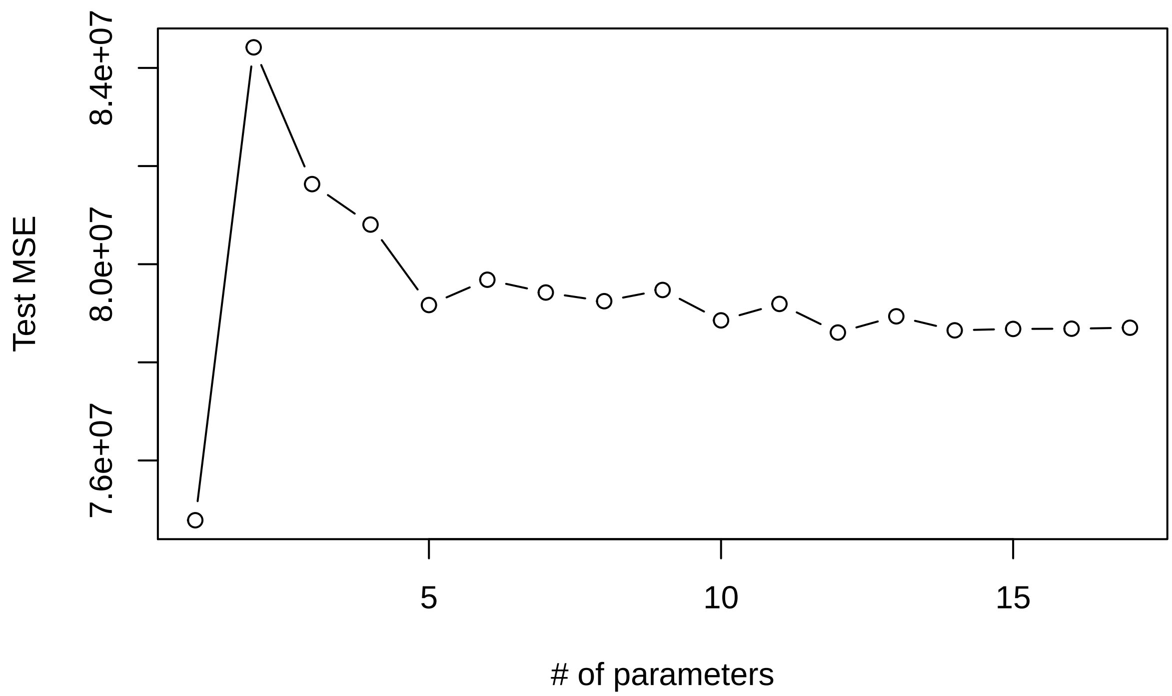

```
plot(reg.summary$adjr2, xlab ="Number of Variables",ylab="Adj R2", type = 'l', main = 'Forward Step - Pe
which.max (reg.summary$adjr2 )
```

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

```
points (which.max (reg.summary$adjr2 ), reg.summary$cp[which.max (reg.summary$adjr2 )], col ="red",cex =
```




```
test.mat=model.matrix (Outstate~.,data=test)
test.val.errors =rep(NA ,17)
for(i in 1:17){
    coefi=coef(regfit.fwd ,id=i)
    pred=test.mat [,names(coefi)] %*% coefi
    test.val.errors [i]= mean(( test$Apps-pred)^2)
}
plot(test.val.errors ,type='b', xlab='# of parameters', ylab='Test MSE')
```



The above indicates that we get the following choice based on the different measures:

BIC - 6 CP - 13 Adjusted R2 - 14

From a visual inspection, it is clear that the elbow in all the plots happens at 6 and therefore it could be a good choice to move forward. Let see the various parameters included in this model

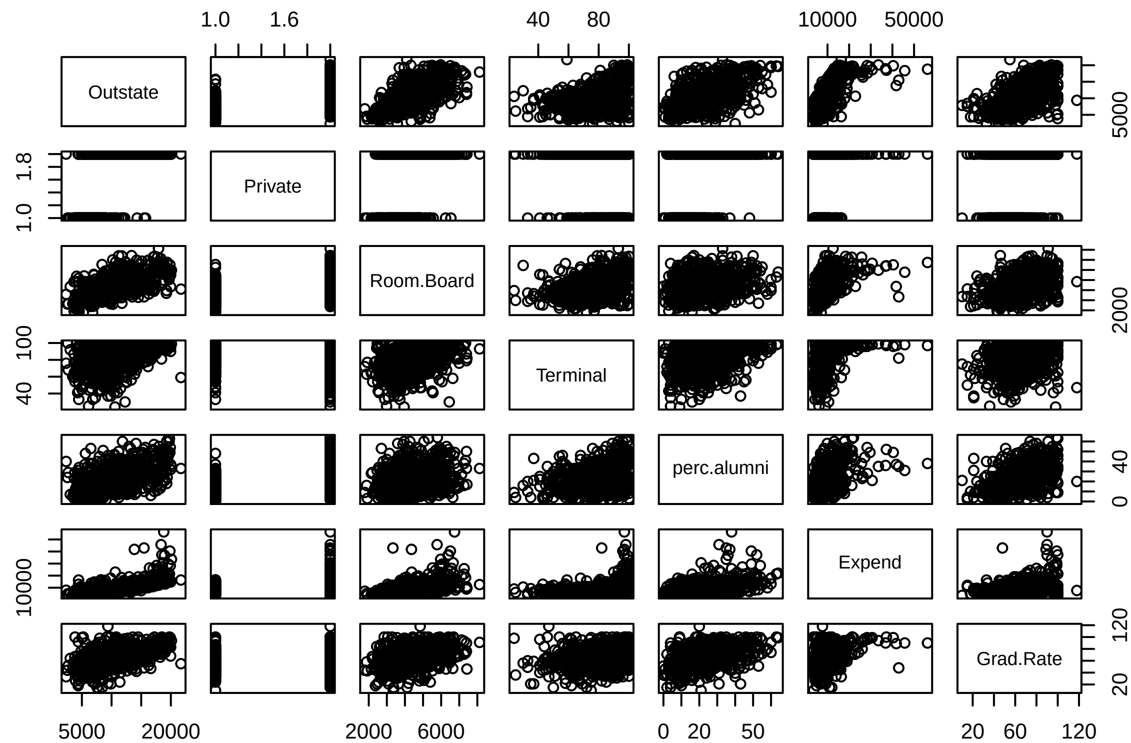
```
names(coef(regfit.fwd ,id=6))
```

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

(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. Use appropriate nonlinear components (e.g. natural splines, step functions) for the variables that need it as in the salary example from the book. Plot the results, and explain your findings. What nonlinear components did you use?

Let's start with a plot

```
#"PrivateYes"  "Room.Board"  "Terminal"    "perc.alumni" "Expend"      "Grad.Rate"
pairs(College[, c("Outstate", "Private", "Room.Board", "Terminal", "perc.alumni", "Expend", "Grad.Rate")
```



It seems like Outstate with Room.Board and perc.alumni appear to be linear while Terminal, Expend and Grad.Rate is non-linear.

Now lets analyze further and compare:

```
fit.rb= lm(Outstate~poly(Room.Board ,5) ,data=train)
coef(summary(fit.rb))
```

```
##                        Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)           10469.803   123.0119 85.1120997 0.000000e+00
## poly(Room.Board, 5)1  64392.552  3013.1645 21.3704068 1.351882e-75
## poly(Room.Board, 5)2  -4022.523  3013.1645 -1.3349827 1.823934e-01
## poly(Room.Board, 5)3  -3075.000  3013.1645 -1.0205217 3.078966e-01
## poly(Room.Board, 5)4  -2694.614  3013.1645 -0.8942804 3.715341e-01
## poly(Room.Board, 5)5  -1396.529  3013.1645 -0.4634760 6.431930e-01
```

```
# The relationship appears to be only linear

fit.t= lm(Outstate~poly(Terminal ,5) ,data=train)
coef(summary(fit.t))
```

```
##                      Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)         10469.803   145.6471 71.8847136 2.991496e-295
## poly(Terminal, 5)1  39574.797  3567.6119 11.0927977  4.078590e-26
## poly(Terminal, 5)2  18568.679  3567.6119  5.2047922  2.679276e-07
## poly(Terminal, 5)3   9619.195  3567.6119  2.6962560  7.211193e-03
## poly(Terminal, 5)4   2499.347  3567.6119  0.7005658  4.838484e-01
## poly(Terminal, 5)5  -2046.429  3567.6119 -0.5736131  5.664467e-01
```

```r
# We have a cubic relationship, we can try ns

fit.pa= lm(Outstate~poly(perc.alumni ,5) ,data=train)
coef(summary(fit.pa))
```

```
##                          Estimate Std. Error     t value      Pr(>|t|)
## (Intercept)           10469.80333   133.9415 78.16696531 8.450476e-315
## poly(perc.alumni, 5)1 56340.89451  3280.8842 17.17247293  5.743245e-54
## poly(perc.alumni, 5)2 -1333.21304  3280.8842 -0.40635785  6.846260e-01
## poly(perc.alumni, 5)3  1838.44687  3280.8842  0.56035105  5.754513e-01
## poly(perc.alumni, 5)4   935.69311  3280.8842  0.28519541  7.755938e-01
## poly(perc.alumni, 5)5    70.14265  3280.8842  0.02137919  9.829504e-01
```

```r
# The relationship appears to be only linear

fit.e= lm(Outstate~poly(Expend ,5) ,data=train)
coef(summary(fit.e))
```

```
##                     Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)        10469.803   104.5294 100.1613265 0.000000e+00
## poly(Expend, 5)1   66364.082  2560.4369  25.9190460 1.159413e-99
## poly(Expend, 5)2  -35092.238  2560.4369 -13.7055663 2.401009e-37
## poly(Expend, 5)3    6038.903  2560.4369   2.3585439 1.866974e-02
## poly(Expend, 5)4    2126.115  2560.4369   0.8303718 4.066622e-01
## poly(Expend, 5)5   -1859.267  2560.4369  -0.7261521 4.680315e-01
```

```r
# We have a cubic relationship, we can try ns

fit.gr= lm(Outstate~poly(Grad.Rate ,5) ,data=train)
coef(summary(fit.gr))
```

```
##                        Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)           10469.803   132.0475 79.2881550 3.747527e-318
## poly(Grad.Rate, 5)1   56192.580  3234.4902 17.3729325  5.686746e-55
## poly(Grad.Rate, 5)2    6408.709  3234.4902  1.9813659  4.801090e-02
## poly(Grad.Rate, 5)3  -11357.065  3234.4902 -3.5112382  4.798610e-04
## poly(Grad.Rate, 5)4   -5046.234  3234.4902 -1.5601329  1.192611e-01
## poly(Grad.Rate, 5)5   -2600.015  3234.4902 -0.8038407  4.218105e-01
```

```r
# We have a cubic relationship, we can try ns

fit.1=gam(Outstate~Room.Board, data=train)
fit.2=gam(Outstate~Room.Board+poly(Terminal,3), data=train)
anova(fit.1, fit.2)
```

```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ Room.Board
## Model 2: Outstate ~ Room.Board + poly(Terminal, 3)
##   Resid. Df Resid. Dev Df  Deviance  Pr(>Chi)
## 1       598 5427868825
## 2       595 4962881502  3 464987323 4.756e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 2nd is better, lets continue
```

```r
fit.3=gam(Outstate~Room.Board+poly(Terminal,3)+perc.alumni, data=train)
anova(fit.1, fit.2, fit.3)
```

```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ Room.Board
## Model 2: Outstate ~ Room.Board + poly(Terminal, 3)
## Model 3: Outstate ~ Room.Board + poly(Terminal, 3) + perc.alumni
##   Resid. Df Resid. Dev Df   Deviance  Pr(>Chi)
## 1       598 5427868825
## 2       595 4962881502  3  464987323 < 2.2e-16 ***
## 3       594 3619103887  1 1343777615 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 3rd is best, lets continue

fit.4=gam(Outstate~Room.Board+poly(Terminal,3)+perc.alumni+poly(Expend,3), data=train)
anova(fit.1, fit.2, fit.3, fit.4)
```

```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ Room.Board
## Model 2: Outstate ~ Room.Board + poly(Terminal, 3)
## Model 3: Outstate ~ Room.Board + poly(Terminal, 3) + perc.alumni
## Model 4: Outstate ~ Room.Board + poly(Terminal, 3) + perc.alumni + poly(Expend,
##     3)
##   Resid. Df Resid. Dev Df   Deviance  Pr(>Chi)
## 1       598 5427868825
## 2       595 4962881502  3  464987323 < 2.2e-16 ***
## 3       594 3619103887  1 1343777615 < 2.2e-16 ***
## 4       591 2715031968  3  904071918 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 4th is best, lets continue

fit.5=gam(Outstate~Room.Board+poly(Terminal,3)+perc.alumni+poly(Expend,3)+poly(Grad.Rate,3), data=train
anova(fit.1, fit.2, fit.3, fit.4, fit.5)
```

```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ Room.Board
## Model 2: Outstate ~ Room.Board + poly(Terminal, 3)
## Model 3: Outstate ~ Room.Board + poly(Terminal, 3) + perc.alumni
## Model 4: Outstate ~ Room.Board + poly(Terminal, 3) + perc.alumni + poly(Expend,
##     3)
## Model 5: Outstate ~ Room.Board + poly(Terminal, 3) + perc.alumni + poly(Expend,
##     3) + poly(Grad.Rate, 3)
##   Resid. Df Resid. Dev Df   Deviance  Pr(>Chi)
## 1       598 5427868825
## 2       595 4962881502  3  464987323 < 2.2e-16 ***
## 3       594 3619103887  1 1343777615 < 2.2e-16 ***
## 4       591 2715031968  3  904071918 < 2.2e-16 ***
## 5       588 2512080880  3  202951088 2.715e-10 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 5th is best, lets continue

fit.6=gam(Outstate~Room.Board+poly(Terminal,3)+perc.alumni+poly(Expend,3)+poly(Grad.Rate,3)+Private, da
anova(fit.1, fit.2, fit.3, fit.4, fit.5, fit.6)
```

```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ Room.Board
## Model 2: Outstate ~ Room.Board + poly(Terminal, 3)
## Model 3: Outstate ~ Room.Board + poly(Terminal, 3) + perc.alumni
## Model 4: Outstate ~ Room.Board + poly(Terminal, 3) + perc.alumni + poly(Expend,
##     3)
## Model 5: Outstate ~ Room.Board + poly(Terminal, 3) + perc.alumni + poly(Expend,
##     3) + poly(Grad.Rate, 3)
## Model 6: Outstate ~ Room.Board + poly(Terminal, 3) + perc.alumni + poly(Expend,
##     3) + poly(Grad.Rate, 3) + Private
##   Resid. Df Resid. Dev Df   Deviance  Pr(>Chi)
## 1       598 5427868825
## 2       595 4962881502  3  464987323 < 2.2e-16 ***
## 3       594 3619103887  1 1343777615 < 2.2e-16 ***
## 4       591 2715031968  3  904071918 < 2.2e-16 ***
## 5       588 2512080880  3  202951088 2.662e-12 ***
## 6       587 2092670759  1  419410121 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

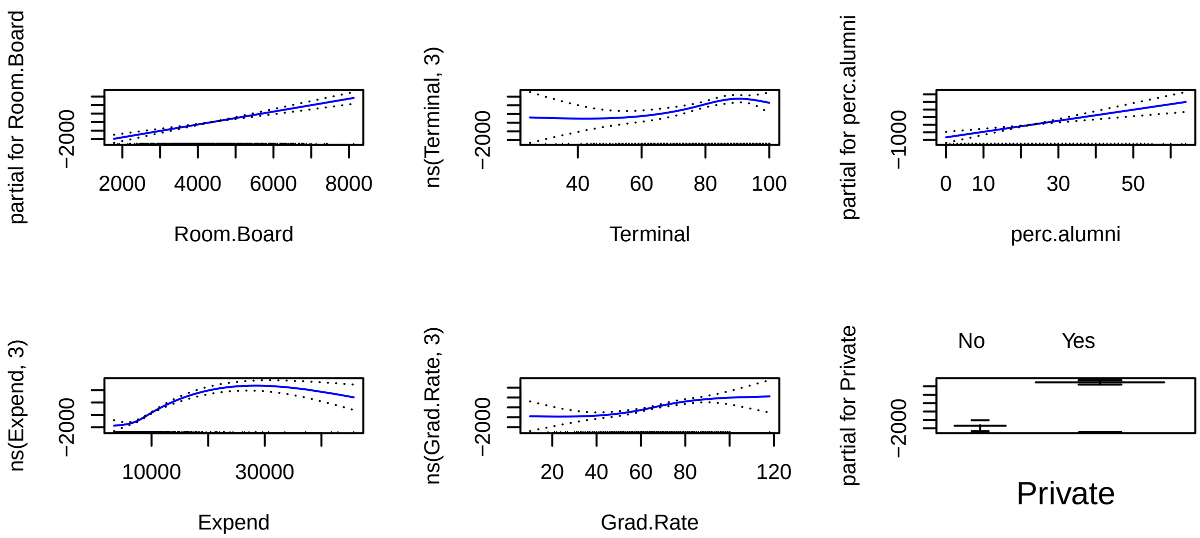
```r
# We will try a model with natural splines
fit.7=gam(Outstate~Room.Board+ns(Terminal,3)+perc.alumni+ns(Expend,3)+ns(Grad.Rate,3)+Private, data=tra


#Since model 6 and 7 are not nested, lets find the MSE on the test data to perform further comparison
pred=predict (fit.6,newdata =test, se=T)
mean((test$Outstate - pred$fit)^2)
```

```
## [1] 3914595
```

```r
pred=predict (fit.7,newdata =test, se=T)
mean((test$Outstate - pred$fit)^2)
```

```
## [1] 3666235
```

```r
# We get lower Test MSE with fit.7 (uses ns)
par(mfrow =c(3,3))
plot(fit.7, se=TRUE ,col ="blue")
```



(c) Evaluate both models obtained from part (a) and (b) on the test set, and explain the results obtained.

Test MSE for model from part (a)

```
coefi=coef(regfit.fwd, id=13)
pred=test.mat [,names(coefi)] %*% coefi
TestMSE_partA = mean(( test$Apps-pred)^2)
```

Test MSE for model from part (b)

```
pred=predict (fit.7 ,newdata =test, se=T)
TestMSE_partB = mean((test$Outstate - pred$fit)^2)

TestMSE_partA / TestMSE_partB
```

```
## [1] 21.53095
```

Test MSE from the non-linear model is a fraction of the test MSE obtained in the part a. Clearly we have a non-linear relationship and it should be used for modeling the relationship.

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

We reviewed all these above individually and found non-linear relationships for the following: Terminal, Expend, Grad.Rate

The result are reproduced here:

```r
#"PrivateYes"  "Room.Board"  "Terminal"    "perc.alumni" "Expend"      "Grad.Rate"
fit.rb= lm(Outstate~poly(Room.Board ,5) ,data=train)
coef(summary(fit.rb))
```

```
##                         Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)            10469.803   123.0119 85.1120997 0.000000e+00
## poly(Room.Board, 5)1 64392.552  3013.1645 21.3704068 1.351882e-75
## poly(Room.Board, 5)2 -4022.523  3013.1645 -1.3349827 1.823934e-01
## poly(Room.Board, 5)3 -3075.000  3013.1645 -1.0205217 3.078966e-01
## poly(Room.Board, 5)4 -2694.614  3013.1645 -0.8942804 3.715341e-01
## poly(Room.Board, 5)5 -1396.529  3013.1645 -0.4634760 6.431930e-01
```

```r
# The relationship appears to be only linear

fit.t= lm(Outstate~poly(Terminal ,5) ,data=train)
coef(summary(fit.t))
```

```
##                       Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)          10469.803   145.6471 71.8847136 2.991496e-295
## poly(Terminal, 5)1 39574.797   3567.6119 11.0927977  4.078590e-26
## poly(Terminal, 5)2 18568.679   3567.6119  5.2047922  2.679276e-07
## poly(Terminal, 5)3  9619.195   3567.6119  2.6962560  7.211193e-03
## poly(Terminal, 5)4  2499.347   3567.6119  0.7005658  4.838484e-01
## poly(Terminal, 5)5 -2046.429   3567.6119 -0.5736131  5.664467e-01
```

```r
# We have a cubic relationship, we can try ns

fit.pa= lm(Outstate~poly(perc.alumni ,5) ,data=train)
coef(summary(fit.pa))
```

```
##                           Estimate Std. Error     t value      Pr(>|t|)
## (Intercept)            10469.80333   133.9415 78.16696531 8.450476e-315
## poly(perc.alumni, 5)1 56340.89451   3280.8842 17.17247293  5.743245e-54
## poly(perc.alumni, 5)2 -1333.21304   3280.8842 -0.40635785  6.846260e-01
## poly(perc.alumni, 5)3  1838.44687   3280.8842  0.56035105  5.754513e-01
## poly(perc.alumni, 5)4   935.69311   3280.8842  0.28519541  7.755938e-01
## poly(perc.alumni, 5)5    70.14265   3280.8842  0.02137919  9.829504e-01
```

```r
# The relationship appears to be only linear

fit.e= lm(Outstate~poly(Expend ,5) ,data=train)
coef(summary(fit.e))
```

```
##                      Estimate Std. Error     t value     Pr(>|t|)
## (Intercept)         10469.803   104.5294 100.1613265 0.000000e+00
## poly(Expend, 5)1   66364.082  2560.4369  25.9190460 1.159413e-99
## poly(Expend, 5)2 -35092.238  2560.4369 -13.7055663 2.401009e-37
## poly(Expend, 5)3    6038.903  2560.4369   2.3585439 1.866974e-02
## poly(Expend, 5)4    2126.115  2560.4369   0.8303718 4.066622e-01
## poly(Expend, 5)5   -1859.267  2560.4369  -0.7261521 4.680315e-01
```

```r
# We have a cubic relationship, we can try ns

fit.gr= lm(Outstate~poly(Grad.Rate ,5) ,data=train)
coef(summary(fit.gr))
```

```
##                         Estimate Std. Error    t value      Pr(>|t|)
```

```
## (Intercept)          10469.803   132.0475 79.2881550 3.747527e-318
## poly(Grad.Rate, 5)1  56192.580  3234.4902 17.3729325  5.686746e-55
## poly(Grad.Rate, 5)2   6408.709  3234.4902  1.9813659  4.801090e-02
## poly(Grad.Rate, 5)3 -11357.065  3234.4902 -3.5112382  4.798610e-04
## poly(Grad.Rate, 5)4  -5046.234  3234.4902 -1.5601329  1.192611e-01
## poly(Grad.Rate, 5)5  -2600.015  3234.4902 -0.8038407  4.218105e-01
```

```
# We have a cubic relationship, we can try ns
```

We can also see in the summary of the GAM model:

```
summary(fit.7)
```

```
##
## Call: gam(formula = Outstate ~ Room.Board + ns(Terminal, 3) + perc.alumni +
##     ns(Expend, 3) + ns(Grad.Rate, 3) + Private, data = train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -7088.12 -1112.63    27.35  1300.92  8416.46
##
## (Dispersion Parameter for gaussian family taken to be 3500160)
##
##     Null Deviance: 9574269519 on 599 degrees of freedom
## Residual Deviance: 2054593869 on 587 degrees of freedom
## AIC: 10758.57
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                   Df     Sum Sq    Mean Sq  F value    Pr(>F)
## Room.Board         1 4146400694 4146400694 1184.632 < 2.2e-16 ***
## ns(Terminal, 3)    3  464901934  154967311   44.274 < 2.2e-16 ***
## perc.alumni        1 1343243697 1343243697  383.766 < 2.2e-16 ***
## ns(Expend, 3)      3  908993330  302997777   86.567 < 2.2e-16 ***
## ns(Grad.Rate, 3)   3  196008440   65336147   18.667 1.412e-11 ***
## Private            1  460127556  460127556  131.459 < 2.2e-16 ***
## Residuals        587 2054593869    3500160
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All the selected terms are significant and shows non-linear relations for: Terminal, Expend, Grad.Rate