705604096_stats101c_hw6

Jade Gregory

2023-11-29

Question 1

```
College <- read.csv("CollegeF23.csv")
head(College)
```

```
##   Private Apps Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad
## 1     Yes  996    866    377        29        58        1411          72
## 2     Yes  548    428    167        18        46         618         113
## 3     Yes  450    430    125        20        46         488          43
## 4     Yes 1470   1199    425        21        76        1820         558
## 5     Yes 1465    810    313        71        95        1088          16
## 6     Yes  417    349    137        60        89         510          63
##   Outstate Room.Board Books Personal PhD Terminal S.F.Ratio perc.alumni
## 1    12065       3615   430      685  62       78      12.5          41
## 2     8958       3670   300     1000  53       59      15.3          26
## 3     9950       3920   300     1300  76       76      11.8          25
## 4    11040       4840   400      900  89       92      13.3          28
## 5    18165       6750   500     1200 100      100      12.3          49
## 6    12960       5450   450      875  92       97       7.7          37
##   Grad.Rate Expend
## 1        80   8596
## 2        64   9798
## 3        47   9466
## 4        94   8118
## 5        89  17449
## 6        59  19016
```

```
set.seed(1128)
index =sample(nrow(College), 2100,replace = FALSE)
C.train=College[index,]
C.test=College[-index,]
dim(C.train)
```

```
## [1] 2100   18
```

```
dim(C.test)
```

```
## [1] 900  18
```

a)

```
# least squares regression
collegelm <- lm(Expend ~ ., data = C.train)
summary(collegelm)
```

```
##
## Call:
## lm(formula = Expend ~ ., data = C.train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8379  -1503   -332    828  31950
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7153.67500  731.06947   9.785  < 2e-16 ***
## PrivateYes  -731.99977  258.64810  -2.830  0.00470 **
## Apps           0.57249    0.06315   9.066  < 2e-16 ***
## Accept        -0.74134    0.13195  -5.618 2.19e-08 ***
## Enroll         0.55167    0.36918   1.494  0.13525
## Top10perc    134.12375   10.07403  13.314  < 2e-16 ***
## Top25perc    -65.57782    7.78280  -8.426  < 2e-16 ***
## F.Undergrad   -0.06479    0.06644  -0.975  0.32961
## P.Undergrad    0.11799    0.05375   2.195  0.02826 *
## Outstate       0.53687    0.03323  16.154  < 2e-16 ***
## Room.Board     0.05500    0.09058   0.607  0.54381
## Books          1.06872    0.46273   2.310  0.02101 *
## Personal       0.24907    0.11315   2.201  0.02783 *
## PhD           -1.32911    8.51188  -0.156  0.87593
## Terminal      25.09413    9.64050   2.603  0.00931 **
## S.F.Ratio   -375.66703   23.76434 -15.808  < 2e-16 ***
## perc.alumni    9.63200    7.32182   1.316  0.18848
## Grad.Rate    -16.40873    5.39817  -3.040  0.00240 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3128 on 2082 degrees of freedom
## Multiple R-squared:  0.6629, Adjusted R-squared:  0.6602
## F-statistic: 240.8 on 17 and 2082 DF,  p-value: < 2.2e-16
```

```
C.trainres <- residuals(collegelm)
mse_train <- mean(C.trainres^2)
test_pred <- predict(collegelm, newdata = C.test, type = "response")
C.testres <- C.test$Expend - test_pred
mse_test <- mean(C.testres^2)
print(c(mse_train, mse_test))
```

```
## [1] 9701589 8779286
```

The MSE for the training data is 9701589 and the MSE for the testing data is 8779286.

b)

```r
# ridge regression
set.seed(12345)
i <- seq(10, -2, length = 100)
lambda.v <- 10^i
x_rid <- model.matrix(Expend ~. , data = C.train)
y_rid <- C.train$Expend
x_rid_tes <- model.matrix(Expend ~., data = C.test)
y_rid_tes <- C.test$Expend
model.ridge <- glmnet(x_rid, y_rid, alpha = 0, lambda = lambda.v)

cv.output <- cv.glmnet(x_rid , y_rid, alpha = 0)
bestlamb.cv <- cv.output$lambda.min

glmtrapred <- predict(model.ridge, s = bestlamb.cv, newx = x_rid)
glmtrares <- C.train$Expend - glmtrapred
mseglmtra <- mean(glmtrares^2)
glmtespred <- predict(model.ridge, s = bestlamb.cv, newx = x_rid_tes)
glmtesres <- C.test$Expend - glmtespred
mseglmtes <- mean(glmtesres^2)
print(c(mseglmtra, mseglmtes))
```

```
## [1] 9964969 9118191
```

The MSE for the training data is 9964969 and the MSE for the testing data is 9118191.

c)

```r
# lasso regression
model.lasso <- glmnet(x_rid, y_rid, alpha = 1, lambda = lambda.v)

lassocv.output <- cv.glmnet(x_rid, y_rid, alpha = 1)
bestlamb.lasso <- lassocv.output$lambda.min

lasstrapred <- predict(model.lasso, s = bestlamb.lasso, newx = x_rid)
lasstrares <- C.train$Expend - lasstrapred
mselasstra <- mean(lasstrares^2)
lasstespred <- predict(model.lasso, s = bestlamb.lasso, newx = x_rid_tes)
lasstesres <- C.test$Expend - lasstespred
mselasstes <- mean(lasstesres^2)
print(c(mselasstra, mselasstes))
```

```
## [1] 9733074 8855893
```

```r
predict(model.lasso, s = bestlamb.lasso, type = "coefficients")
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept) 7137.48748055
## (Intercept)    .
## PrivateYes  -636.01874410
## Apps           0.46581608
```

```
## Accept        -0.49076489
## Enroll         .
## Top10perc    133.64625605
## Top25perc    -60.10919112
## F.Undergrad    .
## P.Undergrad    0.11294494
## Outstate       0.52112456
## Room.Board     0.04597488
## Books          1.04266515
## Personal       0.23057297
## PhD            .
## Terminal      21.37349788
## S.F.Ratio   -377.31658825
## perc.alumni    8.68193524
## Grad.Rate    -14.10126204
```

The MSE of the training data is 9721518 and the MSE of the testing data is 8832009. 15 of our coefficients are nonzero.

Question 2

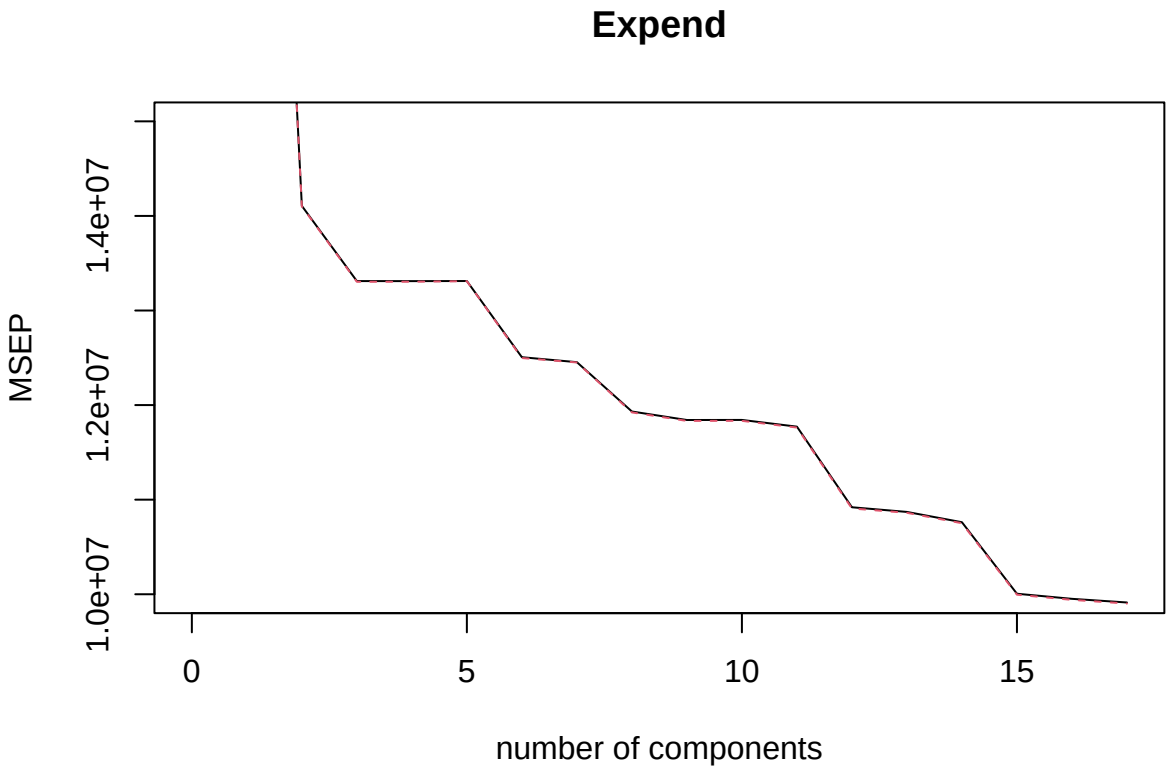
a)

```
library(pls)
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
# PCR model
pcr.fit <- pcr(Expend ~., data = C.train, scale = TRUE, validation = "CV")
validationplot(pcr.fit, val.type = "MSEP", ylim = c(10000000, 15000000))
```




```
summary(pcr.fit)
```

```
## Data:    X dimension: 2100 17
##  Y dimension: 2100 1
## Fit method: svdpc
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            5367     5062     3756     3648     3648     3649     3536
## adjCV         5367     5063     3755     3647     3647     3648     3535
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        3529     3454     3441      3441      3431      3304      3297
## adjCV     3529     3453     3440      3440      3430      3303      3296
##        14 comps  15 comps  16 comps  17 comps
## CV         3280      3163      3155      3148
## adjCV      3279      3161      3153      3147
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X         30.55    58.97    65.78    71.36    76.44    81.18    85.07    88.63
## Expend    12.36    51.39    54.14    54.22    54.24    57.02    57.35    59.09
##         9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X         91.52     93.86     95.92     97.21     98.24     99.10     99.67
```

```
## Expend    59.46     59.51     59.80     62.75     63.03     63.37     65.90
##         16 comps  17 comps
## X          99.87    100.00
## Expend     66.12     66.29
```

```
out.pc <- princomp(x_rid)
summary(out.pc)
```

```
## Importance of components:
##                              Comp.1       Comp.2       Comp.3       Comp.4
## Standard deviation     6347.4938875 4079.5335604 1.702739e+03 1.289495e+03
## Proportion of Variance    0.6407756    0.2646808 4.611027e-02 2.644483e-02
## Cumulative Proportion     0.6407756    0.9054563 9.515666e-01 9.780114e-01
##                              Comp.5       Comp.6       Comp.7       Comp.8
## Standard deviation     773.48092750 6.314750e+02 5.711460e+02 1.878487e+02
## Proportion of Variance   0.00951482 6.341816e-03 5.187948e-03 5.611999e-04
## Cumulative Proportion    0.98752624 9.938681e-01 9.990560e-01 9.996172e-01
##                              Comp.9      Comp.10      Comp.11      Comp.12
## Standard deviation     1.520001e+02 2.109143e+01 1.413151e+01 1.266975e+01
## Proportion of Variance 3.674423e-04 7.074790e-06 3.175986e-06 2.552922e-06
## Cumulative Proportion  9.999846e-01 9.999917e-01 9.999949e-01 9.999975e-01
##                             Comp.13      Comp.14      Comp.15      Comp.16
## Standard deviation     9.206952e+00 6.256677e+00 5.265531e+00 2.902533e+00
## Proportion of Variance 1.348134e-06 6.225709e-07 4.409463e-07 1.339848e-07
## Cumulative Proportion  9.999988e-01 9.999994e-01 9.999999e-01 1.000000e+00
##                             Comp.17 Comp.18
## Standard deviation     2.638862e-01       0
## Proportion of Variance 1.107477e-09       0
## Cumulative Proportion  1.000000e+00       1
```

```
# We cannot reduce the number of PCs in our model since the lowest MSE is with 17 PCs
pcrtrares <- residuals(pcr.fit)
msepcrtra <- mean(pcrtrares^2)
pcrtespred <- predict(pcr.fit, newdata = C.test, type = "response")
pcrtesres <- C.test$Expend - pcrtespred
msepcrtes <- mean(pcrtesres^2)
print(c(msepcrtra, msepcrtes))
```

```
## [1] 12425394 11471031
```

```
povmat <- matrix(c(0.6407756, 0.2646808, 4.611027e-02, 2.644483e-02, 0.00951482, 6.341816e-03, 5.187948
rownames(povmat) <- "Proportion of Variance"
colnames(povmat) <- c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8", "PC9", "PC10", "PC11", "PC
povmat
```

```
##                              PC1       PC2        PC3        PC4        PC5
## Proportion of Variance 0.6407756 0.2646808 0.04611027 0.02644483 0.00951482
##                               PC6         PC7          PC8          PC9
## Proportion of Variance 0.006341816 0.005187948 0.0005611999 0.0003674423
##                              PC10         PC11         PC12         PC13
## Proportion of Variance 7.07479e-06 3.175986e-06 2.552922e-06 1.348134e-06
##                               PC14         PC15         PC16         PC17
## Proportion of Variance 6.225709e-07 4.409463e-07 1.339848e-07 1.107477e-09
```

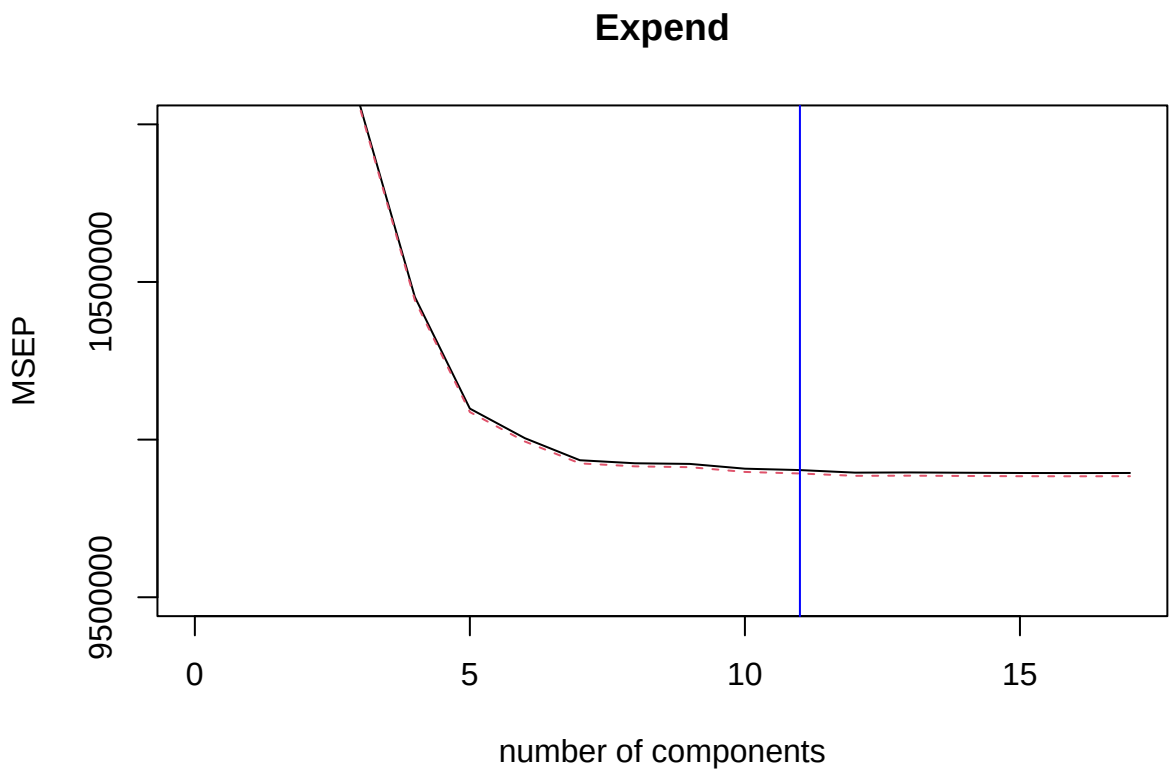
Our MSE for our training data is 12425394, and our MSE for our testing data is 11471031. We have 17 M principal components since the MSE was not found to be smaller at any other PC value. The proportion of variance explained by PC 1 is 0.6407756. The proportion of variance explained by each PC is displayed in the matrix.

b)

```
pls.fit <- plsr(Expend ~., data = C.train, scale = TRUE, validation = "CV")
summary(pls.fit)
```

```
## Data:    X dimension: 2100 17
##  Y dimension: 2100 1
## Fit method: kernelpls
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            5367     3635     3394     3326     3233     3178     3163
## adjCV         5367     3635     3393     3324     3231     3176     3161
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        3152     3150     3150      3148      3147      3146      3146
## adjCV     3150     3149     3148      3146      3145      3144      3144
##        14 comps  15 comps  16 comps  17 comps
## CV         3146      3146      3145      3145
## adjCV      3144      3144      3144      3144
##
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X         28.86     40.2    63.85    68.25    71.35    74.54    76.59    80.49
## Expend    54.39     60.5    62.20    64.37    65.59    65.92    66.13    66.18
##         9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X         83.66     85.49     87.69     90.93     93.69     96.49     97.83
## Expend    66.21     66.25     66.28     66.28     66.29     66.29     66.29
##         16 comps  17 comps
## X          99.10    100.00
## Expend     66.29     66.29
```

```
validationplot(pls.fit, val.type = "MSEP", ylim = c(9500000, 11000000))
abline(v = 11, col = "blue")
```




```
x_rid_pls <- model.matrix(Expend ~ . - Personal - PhD - Terminal - S.F.Ratio - perc.alumni - Grad.Rate,
pls.11pc <- plsr(Expend ~., data = C.train, scale = TRUE, ncomp = 11)
summary(pls.11pc)
```

```
## Data:    X dimension: 2100 17
##  Y dimension: 2100 1
## Fit method: kernelpls
## Number of components considered: 11
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X         28.86     40.2    63.85    68.25    71.35    74.54    76.59    80.49
## Expend    54.39     60.5    62.20    64.37    65.59    65.92    66.13    66.18
##         9 comps  10 comps  11 comps
## X         83.66     85.49     87.69
## Expend    66.21     66.25     66.28
```

```
out.11pc <- princomp(x_rid_pls)
summary(out.11pc)
```

```
## Importance of components:
##                             Comp.1       Comp.2       Comp.3       Comp.4
## Standard deviation     6345.0733531 4075.6649662 1.701075e+03 1.286690e+03
## Proportion of Variance    0.6449704    0.2661114 4.635677e-02 2.652249e-02
## Cumulative Proportion     0.6449704    0.9110818 9.574386e-01 9.839611e-01
```

```
##                              Comp.5       Comp.6       Comp.7       Comp.8
## Standard deviation     7.730296e+02 5.866868e+02 1.878399e+02 1.540788e+02
## Proportion of Variance 9.573237e-03 5.514157e-03 5.652522e-04 3.803226e-04
## Cumulative Proportion  9.935343e-01 9.990485e-01 9.996137e-01 9.999940e-01
##                              Comp.9      Comp.10      Comp.11 Comp.12
## Standard deviation     1.840898e+01 5.798258e+00 2.805830e-01       0
## Proportion of Variance 5.429078e-06 5.385942e-07 1.261216e-09       0
## Cumulative Proportion  9.999995e-01 1.000000e+00 1.000000e+00       1
```

```
# calculating MSE
plstrares <- residuals(pls.11pc)
mseplstra <- mean(plstrares^2)
plstespred <- predict(pls.11pc, newdata = C.test, type = "response")
plstesres <- C.test$Expend - plstespred
mseplstes <- mean(plstesres^2)
print(c(mseplstra, mseplstes))
```

```
## [1] 10360532  9508265
```

```
# Proportion of variance explained martix
povmat2 <- matrix(c(0.6449704, 0.2661114, 4.635677e-02, 2.652249e-02, 9.573237e-03,  5.514157e-03, 5.65
rownames(povmat2) <- "Proportion of Variance"
colnames(povmat2) <- c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8", "PC9", "PC10", "PC11")
povmat2
```

```
##                               PC1       PC2        PC3        PC4         PC5
## Proportion of Variance 0.6449704 0.2661114 0.04635677 0.02652249 0.009573237
##                                PC6          PC7          PC8          PC9
## Proportion of Variance 0.005514157 0.0005652522 0.0003803226 5.429078e-06
##                               PC10         PC11
## Proportion of Variance 5.385942e-07 1.261216e-09
```

Our MSE for our training data is 10360532 and our MSE for our testing data is 9508265. We were able to use 11 components based off of our cross validation. The proportion of variance explained by each PC is displayed in the matrix.

Question 3

a)

```
# Backwards Stepwise Selection
baic <- step(collegelm, direction = "backward", data = C.train, k = log(nrow(C.train)))
```

```
## Start:  AIC=33922.07
## Expend ~ Private + Apps + Accept + Enroll + Top10perc + Top25perc +
##     F.Undergrad + P.Undergrad + Outstate + Room.Board + Books +
##     Personal + PhD + Terminal + S.F.Ratio + perc.alumni + Grad.Rate
##
##               Df  Sum of Sq        RSS   AIC
## - PhD          1     238589 2.0374e+10 33914
## - Room.Board   1    3607325 2.0377e+10 33915
```

```
## - F.Undergrad  1    9304872 2.0383e+10 33915
## - perc.alumni  1   16934627 2.0390e+10 33916
## - Enroll       1   21850647 2.0395e+10 33917
## - P.Undergrad  1   47156806 2.0420e+10 33919
## - Personal     1   47415973 2.0421e+10 33919
## - Books        1   52197583 2.0426e+10 33920
## - Terminal     1   66301971 2.0440e+10 33921
## <none>                     2.0373e+10 33922
## - Private      1   78376293 2.0452e+10 33922
## - Grad.Rate    1   90414485 2.0464e+10 33924
## - Accept       1  308872405 2.0682e+10 33946
## - Top25perc    1  694742298 2.1068e+10 33985
## - Apps         1  804338526 2.1178e+10 33996
## - Top10perc    1 1734547490 2.2108e+10 34086
## - S.F.Ratio    1 2445322167 2.2819e+10 34152
## - Outstate     1 2553683143 2.2927e+10 34162
##
## Step:  AIC=33914.45
## Expend ~ Private + Apps + Accept + Enroll + Top10perc + Top25perc +
##     F.Undergrad + P.Undergrad + Outstate + Room.Board + Books +
##     Personal + Terminal + S.F.Ratio + perc.alumni + Grad.Rate
##
##               Df  Sum of Sq        RSS   AIC
## - Room.Board   1    3683488 2.0377e+10 33907
## - F.Undergrad  1    9237666 2.0383e+10 33908
## - perc.alumni  1   16965123 2.0391e+10 33909
## - Enroll       1   21936690 2.0396e+10 33909
## - P.Undergrad  1   46934228 2.0421e+10 33912
## - Personal     1   47621253 2.0421e+10 33912
## - Books        1   56435502 2.0430e+10 33913
## <none>                     2.0374e+10 33914
## - Private      1   78209154 2.0452e+10 33915
## - Grad.Rate    1   91140573 2.0465e+10 33916
## - Terminal     1  132151531 2.0506e+10 33920
## - Accept       1  312102351 2.0686e+10 33939
## - Top25perc    1  696038538 2.1070e+10 33977
## - Apps         1  806742915 2.1180e+10 33988
## - Top10perc    1 1806420136 2.2180e+10 34085
## - S.F.Ratio    1 2454871209 2.2828e+10 34146
## - Outstate     1 2554706116 2.2928e+10 34155
##
## Step:  AIC=33907.18
## Expend ~ Private + Apps + Accept + Enroll + Top10perc + Top25perc +
##     F.Undergrad + P.Undergrad + Outstate + Books + Personal +
##     Terminal + S.F.Ratio + perc.alumni + Grad.Rate
##
##               Df  Sum of Sq        RSS   AIC
## - F.Undergrad  1    9110249 2.0386e+10 33900
## - perc.alumni  1   15036912 2.0392e+10 33901
## - Enroll       1   20911844 2.0398e+10 33902
## - Personal     1   46187285 2.0423e+10 33904
## - P.Undergrad  1   48508264 2.0426e+10 33905
## - Books        1   59589146 2.0437e+10 33906
## <none>                     2.0377e+10 33907
```

```
## - Private      1   75487802 2.0453e+10 33907
## - Grad.Rate    1   88075353 2.0465e+10 33909
## - Terminal     1  145137083 2.0522e+10 33914
## - Accept       1  313705457 2.0691e+10 33932
## - Top25perc    1  700503713 2.1078e+10 33971
## - Apps         1  833146256 2.1210e+10 33984
## - Top10perc    1 1808132929 2.2185e+10 34078
## - S.F.Ratio    1 2457317729 2.2835e+10 34139
## - Outstate     1 3026946773 2.3404e+10 34190
##
## Step:  AIC=33900.47
## Expend ~ Private + Apps + Accept + Enroll + Top10perc + Top25perc +
##     P.Undergrad + Outstate + Books + Personal + Terminal + S.F.Ratio +
##     perc.alumni + Grad.Rate
##
##               Df  Sum of Sq        RSS   AIC
## - Enroll       1   12652972 2.0399e+10 33894
## - perc.alumni  1   17113459 2.0403e+10 33895
## - P.Undergrad  1   41411971 2.0428e+10 33897
## - Personal     1   43691190 2.0430e+10 33897
## - Books        1   59906019 2.0446e+10 33899
## - Private      1   68940886 2.0455e+10 33900
## <none>                      2.0386e+10 33900
## - Grad.Rate    1   87166434 2.0474e+10 33902
## - Terminal     1  144039076 2.0530e+10 33908
## - Accept       1  316347463 2.0703e+10 33925
## - Top25perc    1  709476024 2.1096e+10 33965
## - Apps         1  833319355 2.1220e+10 33977
## - Top10perc    1 1808836445 2.2195e+10 34071
## - S.F.Ratio    1 2495286117 2.2882e+10 34135
## - Outstate     1 3041856779 2.3428e+10 34185
##
## Step:  AIC=33894.12
## Expend ~ Private + Apps + Accept + Top10perc + Top25perc + P.Undergrad +
##     Outstate + Books + Personal + Terminal + S.F.Ratio + perc.alumni +
##     Grad.Rate
##
##               Df  Sum of Sq        RSS   AIC
## - perc.alumni  1   19084398 2.0418e+10 33888
## - Personal     1   47779084 2.0447e+10 33891
## - Books        1   60082287 2.0459e+10 33893
## - P.Undergrad  1   60598618 2.0460e+10 33893
## <none>                      2.0399e+10 33894
## - Private      1   82655351 2.0482e+10 33895
## - Grad.Rate    1   86616780 2.0486e+10 33895
## - Terminal     1  141191226 2.0540e+10 33901
## - Accept       1  454604785 2.0854e+10 33933
## - Top25perc    1  728923466 2.1128e+10 33960
## - Apps         1  835693823 2.1235e+10 33971
## - Top10perc    1 2047348860 2.2446e+10 34087
## - S.F.Ratio    1 2485155442 2.2884e+10 34128
## - Outstate     1 3084342505 2.3483e+10 34182
##
## Step:  AIC=33888.44
```

```
## Expend ~ Private + Apps + Accept + Top10perc + Top25perc + P.Undergrad +
##     Outstate + Books + Personal + Terminal + S.F.Ratio + Grad.Rate
##
##               Df  Sum of Sq        RSS   AIC
## - Personal     1   42953851 2.0461e+10 33885
## - Books        1   58164117 2.0476e+10 33887
## - P.Undergrad  1   62542587 2.0481e+10 33887
## <none>                      2.0418e+10 33888
## - Grad.Rate    1   75468662 2.0494e+10 33889
## - Private      1   79335526 2.0497e+10 33889
## - Terminal     1  147881452 2.0566e+10 33896
## - Accept       1  464799478 2.0883e+10 33928
## - Top25perc    1  723591516 2.1142e+10 33954
## - Apps         1  828584187 2.1247e+10 33964
## - Top10perc    1 2105242624 2.2523e+10 34087
## - S.F.Ratio    1 2539567897 2.2958e+10 34127
## - Outstate     1 3226851280 2.3645e+10 34189
##
## Step:  AIC=33885.2
## Expend ~ Private + Apps + Accept + Top10perc + Top25perc + P.Undergrad +
##     Outstate + Books + Terminal + S.F.Ratio + Grad.Rate
##
##               Df  Sum of Sq        RSS   AIC
## <none>                      2.0461e+10 33885
## - Books        1   75977207 2.0537e+10 33885
## - Grad.Rate    1   81989130 2.0543e+10 33886
## - Private      1   83970225 2.0545e+10 33886
## - P.Undergrad  1   85302539 2.0546e+10 33886
## - Terminal     1  147129462 2.0608e+10 33893
## - Accept       1  462187498 2.0923e+10 33924
## - Top25perc    1  735544147 2.1197e+10 33952
## - Apps         1  832164041 2.1293e+10 33961
## - Top10perc    1 2137866366 2.2599e+10 34086
## - S.F.Ratio    1 2599628239 2.3061e+10 34129
## - Outstate     1 3193972990 2.3655e+10 34182
```

```
bictrares <- residuals(baic)
msebictra <- mean(bictrares^2)
bictespred <- predict(baic, newdata = C.test, type = "response")
bictesres <- C.test$Expend - bictespred
msebictes <- mean(bictesres^2)
print(c(msebictra, msebictes))
```

```
## [1] 9743362 8828745
```

Our satisfactory model from our BIC is Expend ~ Private + Apps + Accept + Top10perc + Top25perc + P.Undergrad + Outstate + Books + Terminal + S.F.Ratio + Grad.Rate. The MSE for our training data is 9743362 and the MSE for our testing data is 8828745.

b)

```r
library(gam)
```

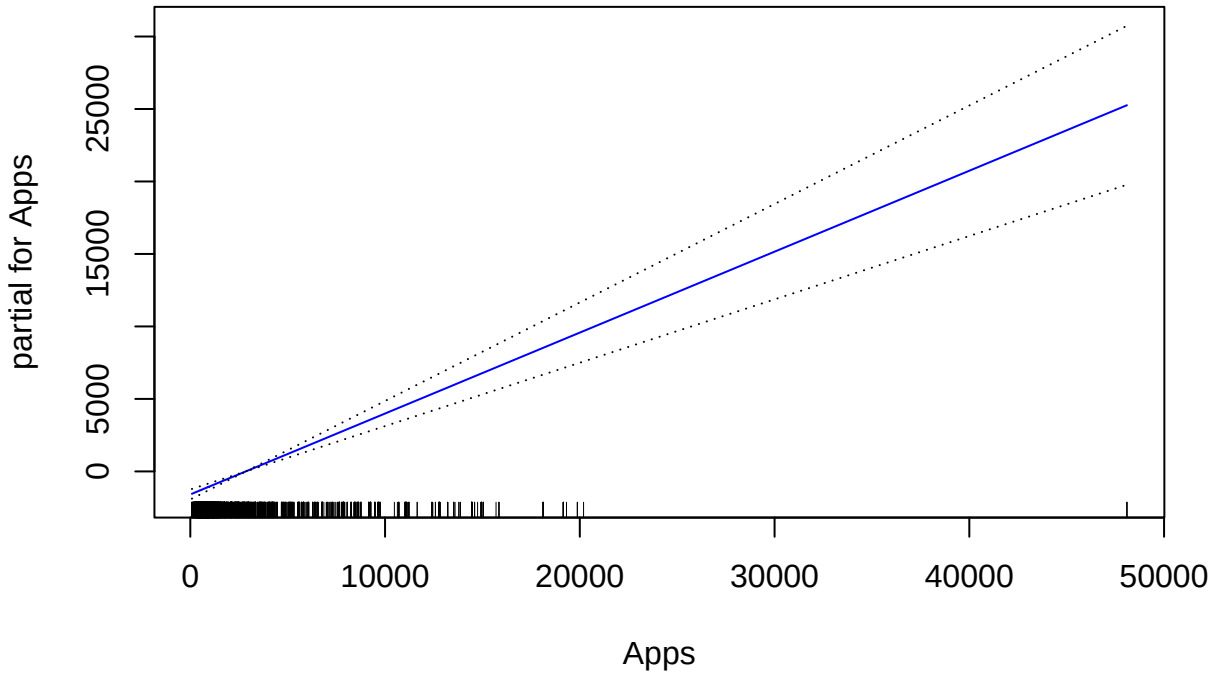
```
## Loading required package: splines
```

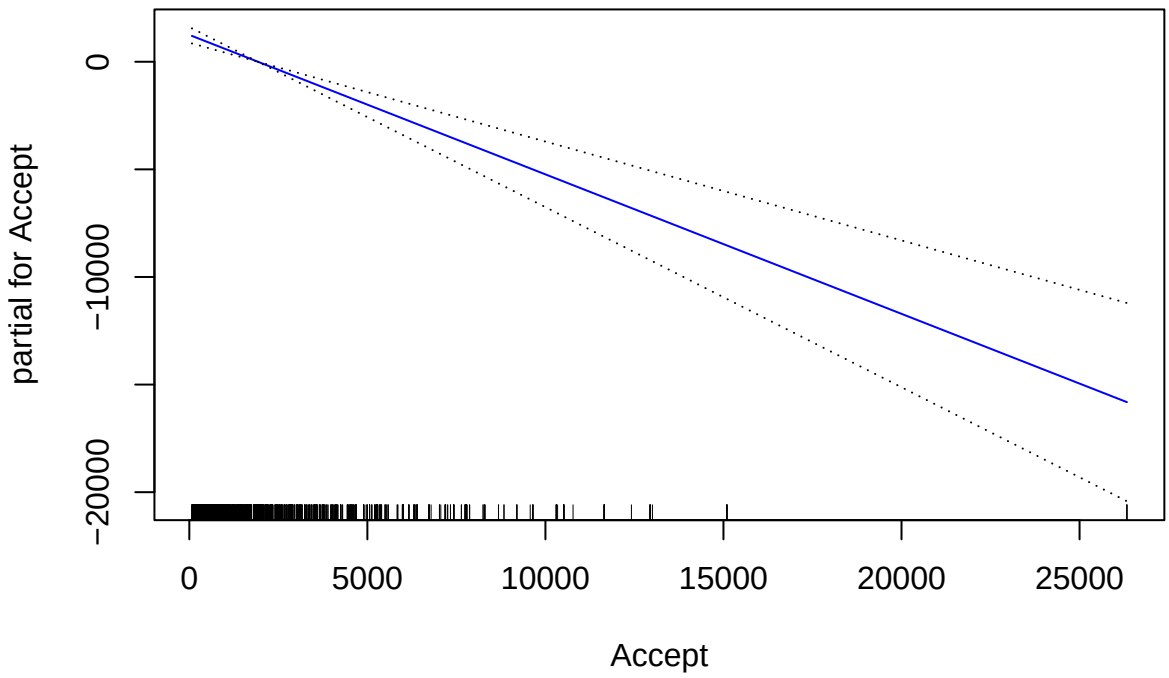
```
## Loading required package: foreach
```

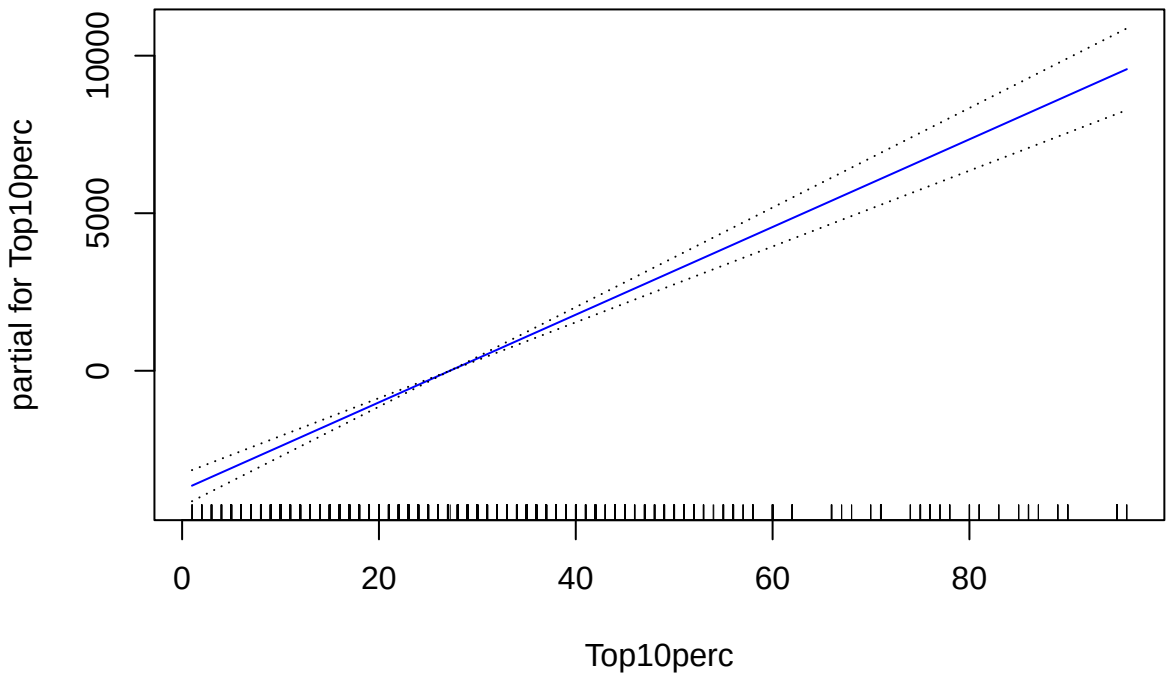
```
## Loaded gam 1.22-2
```

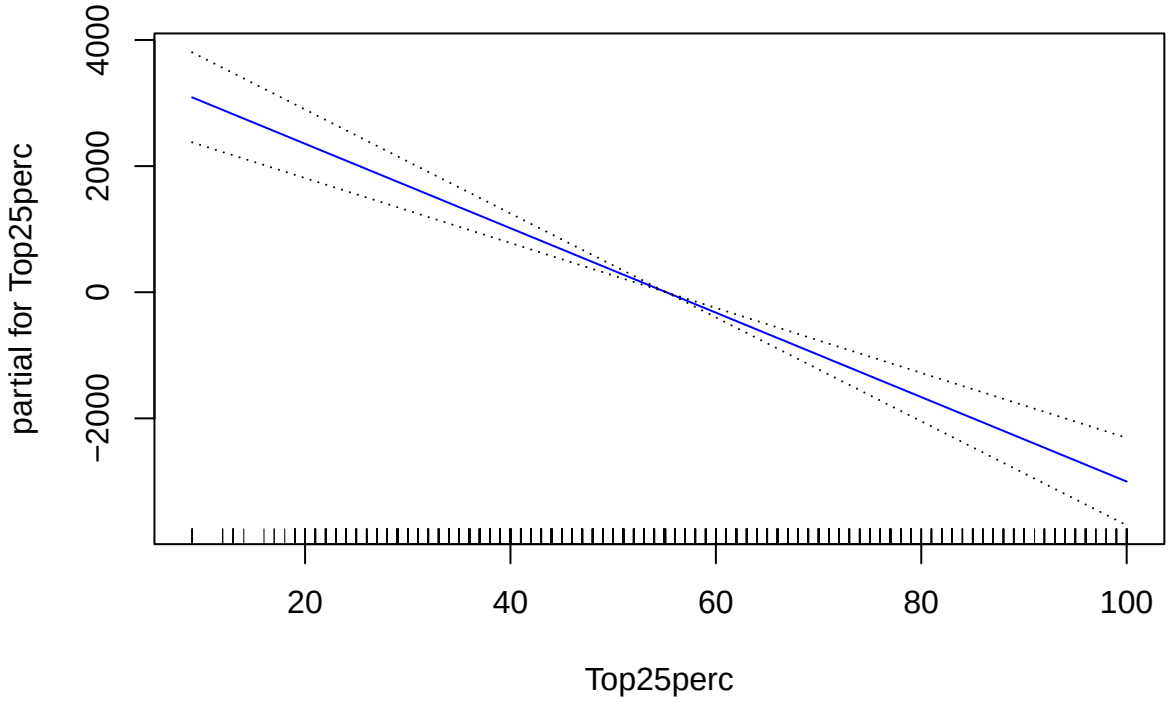
```r
# GAM model
traingam <- gam(Expend ~ Private + Apps + Accept + Top10perc + Top25perc + P.Undergrad + Outstate + Bool
plot(traingam, se = TRUE, col = "blue")
```

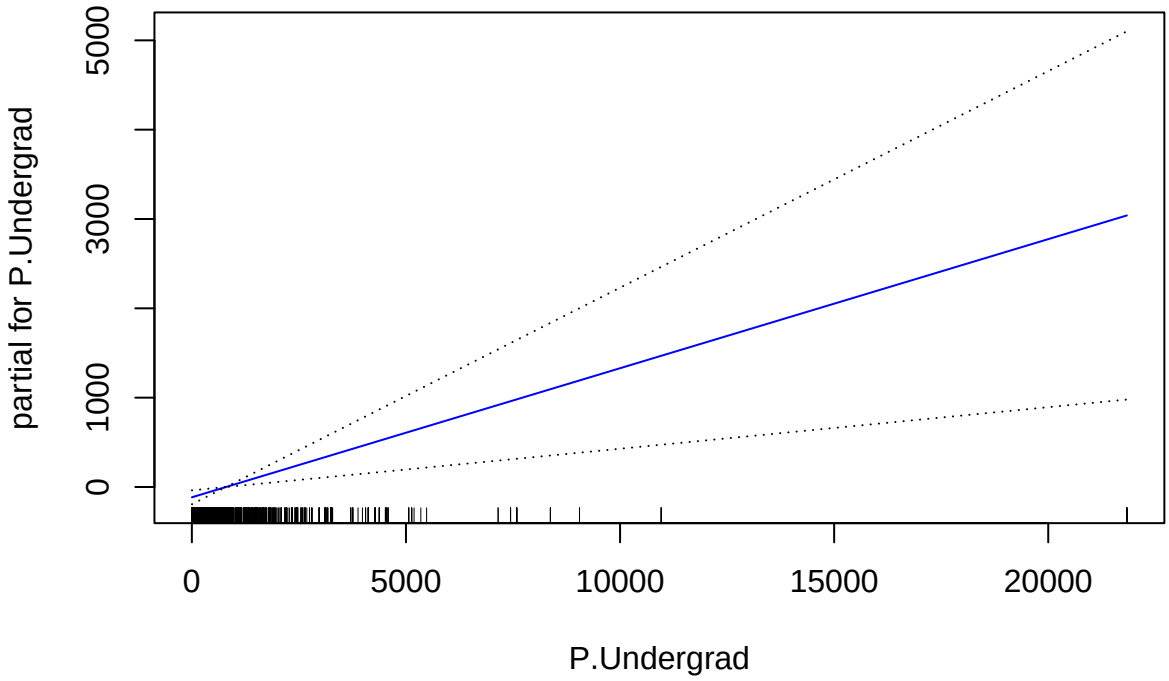
```
## Warning in gplot.default(x = c("Yes", "Yes", "Yes", "Yes", "Yes", "No", : The
## "x" component of "partial for Private" has class "character"; no gplot()
## methods available
```

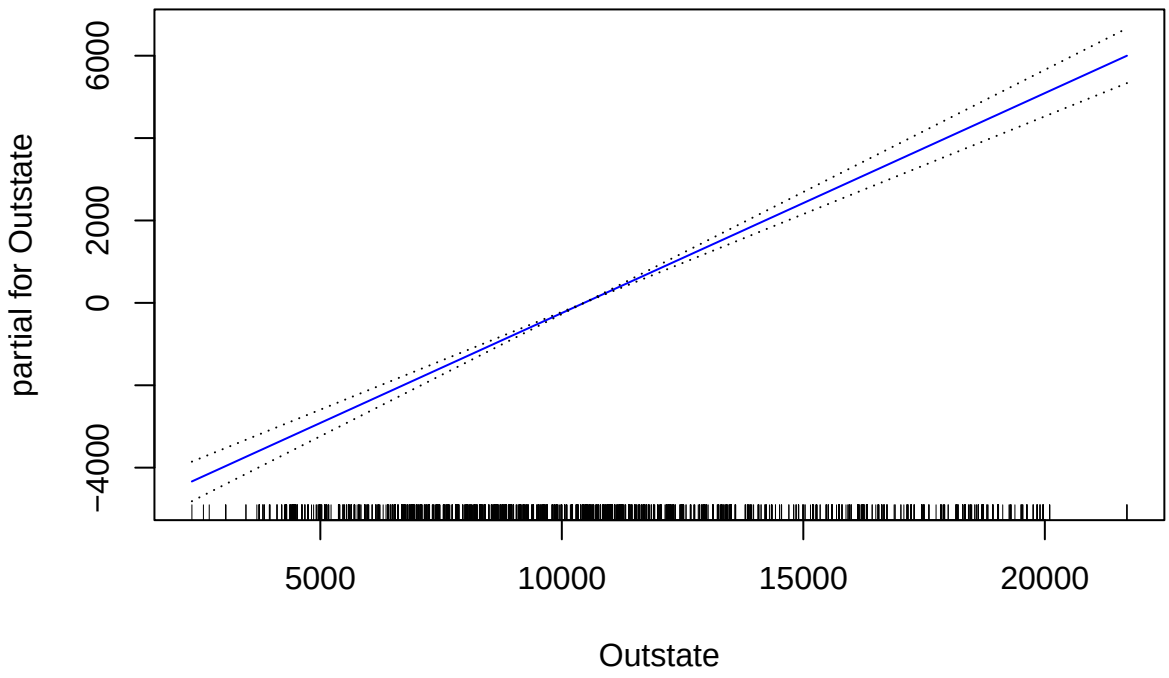


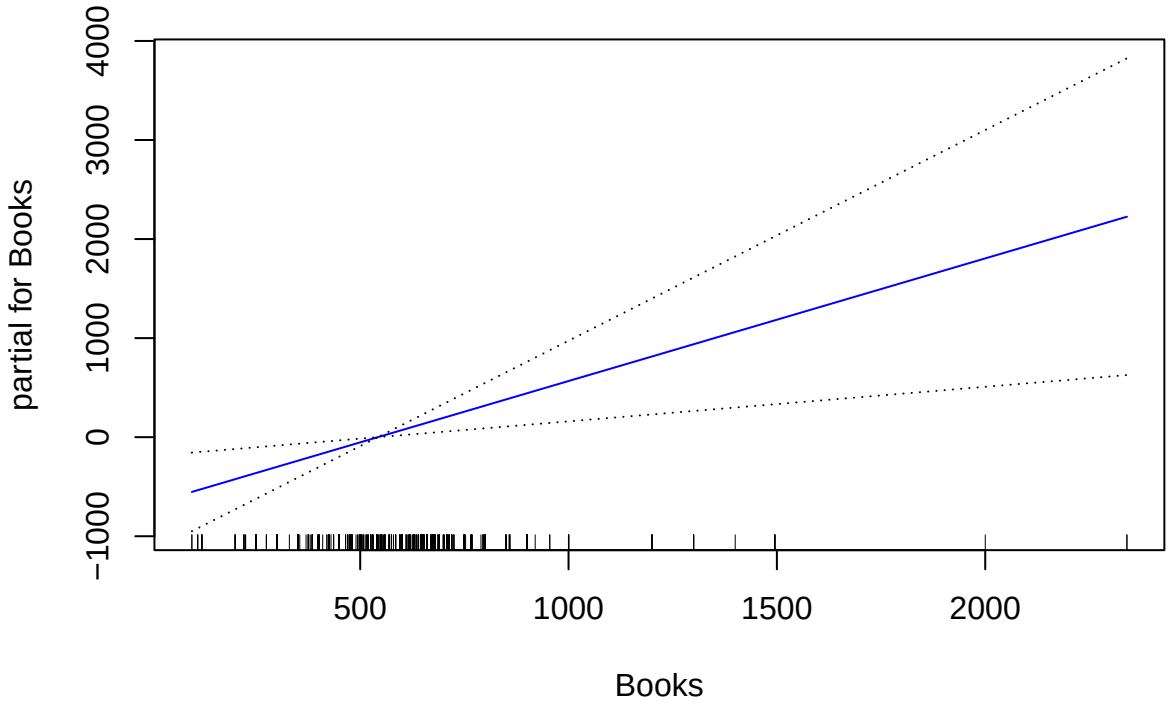

partial for Accept
0
−10000
−20000
0
5000
10000
15000
20000
25000
Accept

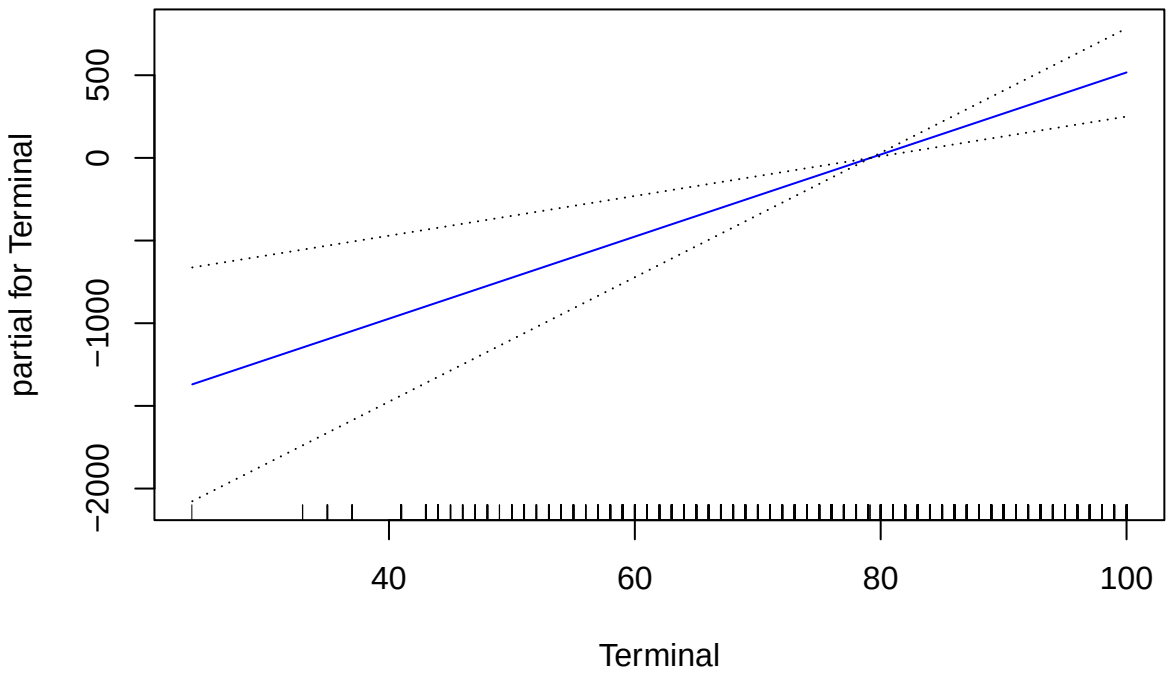

partial for Top10perc
0
5000
10000
0
20
40
60
80
Top10perc

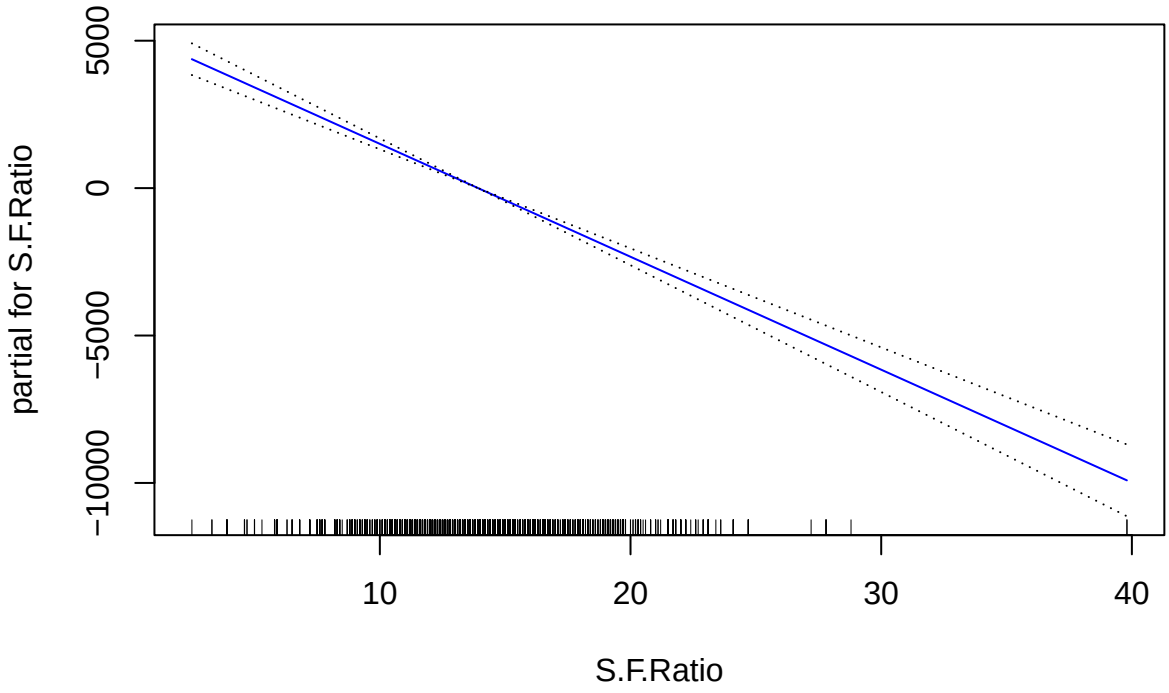

partial for Top25perc
4000
2000
0
−2000
20
40
60
80
100
Top25perc

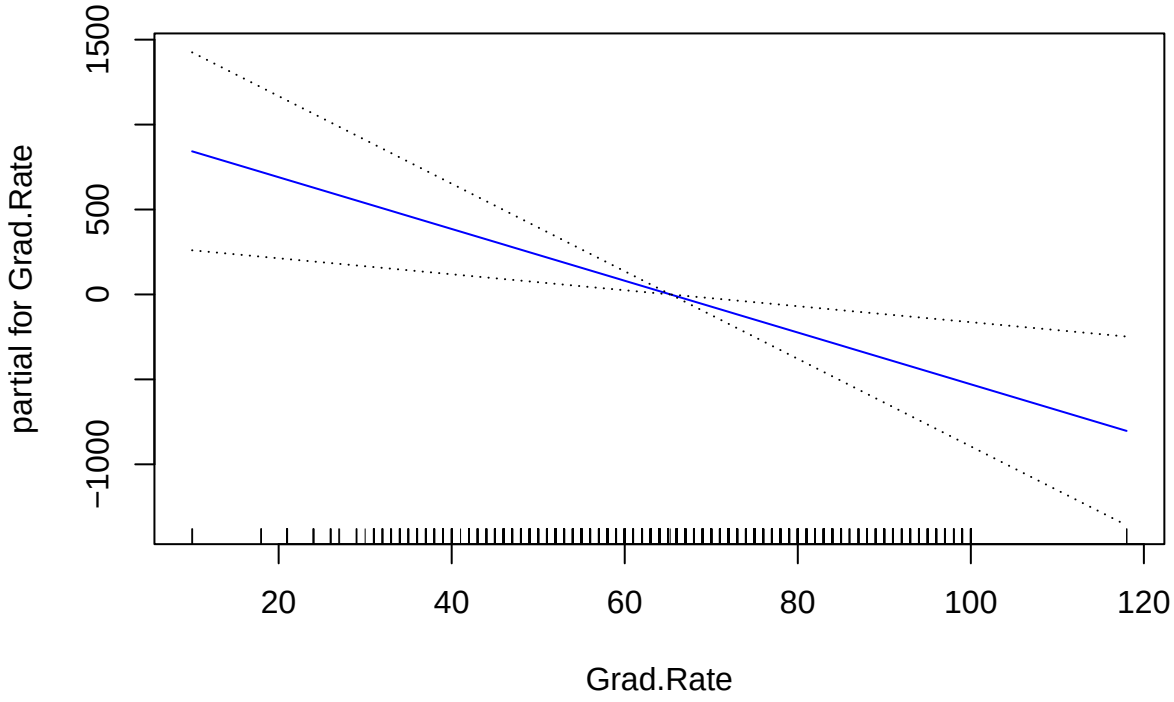

partial for P.Undergrad
0
1000
3000
5000
0
5000
10000
15000
20000
P.Undergrad


partial for Outstate
−4000
0
2000
6000
5000
10000
15000
20000
Outstate


partial for Books
−1000
0
1000
2000
3000
4000
500
1000
1500
2000
Books


partial for Terminal
500
0
−1000
−2000
40
60
80
100
Terminal


partial for S.F.Ratio
5000
0
−5000
−10000
10
20
30
40
S.F.Ratio



All of our predictors have a linear relationship with the response as shown by their smooth plots. We can also argue that all of the predictors in our model are significant because none of them have confidence intervals that are horizontal across the graph.

c)

```
gamtrares <- residuals(traingam)
msegamtra <- mean(gamtrares^2)
gamtespred <- predict(traingam, newdata = C.test, type = "response")
gamtesres <- C.test$Expend - gamtespred
msegamtes <- mean(gamtesres^2)
print(c(msegamtra, msegamtes))
```

```
## [1] 9743362 8828745
```

Our MSE value for our testing data is 8828745. This is reduced from our MSE value for our training data which is 9743362. This means that our model holds well amongst our testing data. d)

```
summary(traingam)
```

```
##
## Call: gam(formula = Expend ~ Private + Apps + Accept + Top10perc +
##     Top25perc + P.Undergrad + Outstate + Books + Terminal + S.F.Ratio +
##     Grad.Rate, data = C.train)
```

```
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -8615.3 -1505.0  -326.0   771.6 31872.7
##
## (Dispersion Parameter for gaussian family taken to be 9799358)
##
##     Null Deviance: 60437676793 on 2099 degrees of freedom
## Residual Deviance: 20461059549 on 2088 degrees of freedom
## AIC: 39778.94
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##               Df     Sum Sq    Mean Sq   F value    Pr(>F)
## Private        1 3.6201e+09 3.6201e+09  369.4185 < 2.2e-16 ***
## Apps           1 9.8480e+09 9.8480e+09 1004.9593 < 2.2e-16 ***
## Accept         1 4.0817e+09 4.0817e+09  416.5305 < 2.2e-16 ***
## Top10perc      1 1.1768e+10 1.1768e+10 1200.9440 < 2.2e-16 ***
## Top25perc      1 8.0223e+08 8.0223e+08   81.8661 < 2.2e-16 ***
## P.Undergrad    1 1.7661e+08 1.7661e+08   18.0225 2.279e-05 ***
## Outstate       1 6.6477e+09 6.6477e+09  678.3828 < 2.2e-16 ***
## Books          1 1.2445e+08 1.2445e+08   12.7000 0.0003739 ***
## Terminal       1 1.1062e+08 1.1062e+08   11.2884 0.0007939 ***
## S.F.Ratio      1 2.7148e+09 2.7148e+09  277.0349 < 2.2e-16 ***
## Grad.Rate      1 8.1989e+07 8.1989e+07    8.3668 0.0038610 **
## Residuals   2088 2.0461e+10 9.7994e+06
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No, from our output we do not see that any of the predictors have a strong non-linear relationship with our response.

Question 4

The testing MSE for the Least Squares full model using lm is 8779286. The testing MSE for the ridge model with the best lambda is 9118191. The testing MSE for the lasso model with the best lambda is 8832009. The testing MSE for the PCR model is 11471031. The testing MSE for the PLS model is 9508265. The testing MSE for the stepwise backward regression using BIC is 8828745. The testing MSE for the GAM is 8828745. We can see that there is not a huge amount of variation among our testing MSE values. Our PCR model does the worst, with the highest MSE of 11471031. Our least squares full model using does the best, with the smallest MSE value of 8779286. From this, we can say that our least squares full model has the least amount of error.

Question 5

a)

```
births <- read.csv("better2000births.csv")
birth <- na.omit(births)
head(birth)
```

```
##   Gender Premie weight Apgar1 Fage Mage Feduc Meduc TotPreg Visits   Marital
```

```
## 1   Male     No    124      8   31   25    13    14       1     13   Married
## 2 Female     No    177      8   36   26     9    12       2     11 Unmarried
## 3   Male     No    107      3   30   16    12     8       2     10 Unmarried
## 4 Female     No    144      6   33   37    12    14       2     12 Unmarried
## 5   Male     No    117      9   36   33    10    16       2     19   Married
## 6 Female     No     98      4   31   29    14    16       3     20   Married
##   Racemom Racedad Hispmom Hispdad Gained     Habit MomPriorCond BirthDef
## 1   White   White NotHisp NotHisp     40 NonSmoker         None     None
## 2   White   White Mexican Mexican     20 NonSmoker         None     None
## 3   White Unknown Mexican Unknown     70 NonSmoker At Least One     None
## 4   White   White NotHisp NotHisp     50 NonSmoker         None     None
## 5   White   Black NotHisp NotHisp     40 NonSmoker At Least One     None
## 6   White   White NotHisp NotHisp     21 NonSmoker         None     None
##      DelivComp BirthComp
## 1 At Least One      None
## 2 At Least One      None
## 3 At Least One      None
## 4 At Least One      None
## 5         None      None
## 6         None      None
```

```
dim(birth)
```

```
## [1] 1998   21
```

```
set.seed(1128)
birth$Gender <- as.factor(birth$Gender)
birth$Premie <- as.factor(birth$Premie)
birth$Marital <- as.factor(birth$Marital)
birth$Racemom <- as.factor(birth$Racemom)
birth$Racedad <- as.factor(birth$Racedad)
birth$Hispmom <- as.factor(birth$Hispmom)
birth$Hispdad <- as.factor(birth$Hispdad)
birth$Habit <- as.factor(birth$Habit)
birth$MomPriorCond <- as.factor(birth$MomPriorCond)
birth$BirthDef <- as.factor(birth$BirthDef)
birth$DelivComp <- as.factor(birth$DelivComp)
birth$BirthComp <- as.factor(birth$BirthComp)
s.train.i <- sample(1:nrow(birth), 1000, replace = FALSE)
length(s.train.i)
```

```
## [1] 1000
```

```
b.train <- birth[s.train.i,]
dim(b.train)
```

```
## [1] 1000   21
```

```
b.test <- birth[-s.train.i,]
dim(b.test)
```

```
## [1] 998  21
```

```r
library(tree)
```

```r
tree.birth <- tree(Premie ~., data = b.train)
summary(tree.birth)
```

```
##
## Classification tree:
## tree(formula = Premie ~ ., data = b.train)
## Variables actually used in tree construction:
## [1] "weight"    "Feduc"     "Apgar1"    "Fage"      "BirthComp"
## Number of terminal nodes:  10
## Residual mean deviance:  0.2457 = 243.2 / 990
## Misclassification error rate: 0.051 = 51 / 1000
```

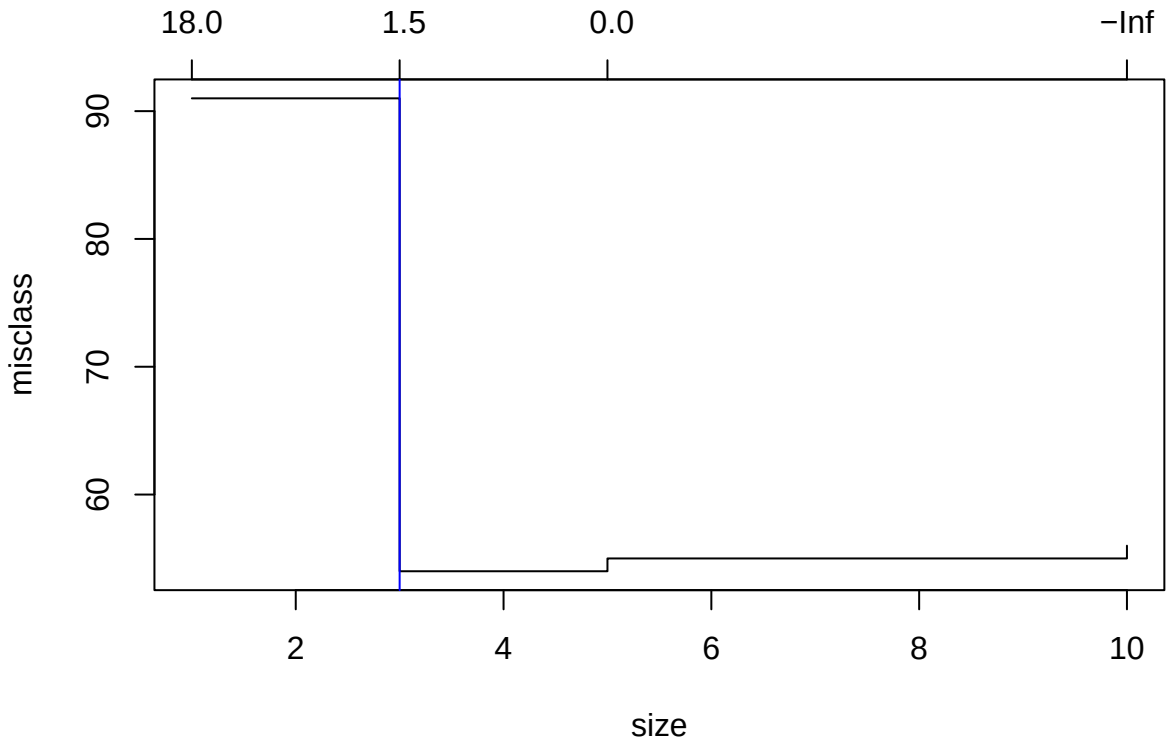
```r
treetespred <- predict(tree.birth, newdata = b.test, type = "class")
mean(treetespred != b.test$Premie)
```

```
## [1] 0.06212425
```

Our testing misclassification rate is 0.06212425.

b)

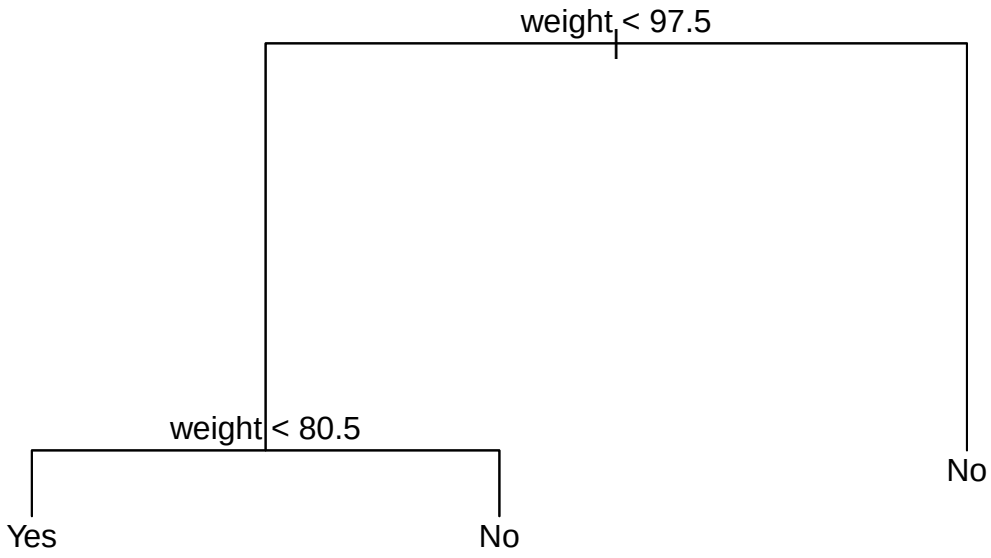
```r
cv.tree.birth <- cv.tree(tree.birth, FUN = prune.misclass)
plot(cv.tree.birth)
abline(v = 3, col = "blue")
```




```
par(mfrow = c(1, 2))
plot(cv.tree.birth$size, cv.tree.birth$dev, type = "b")
plot(cv.tree.birth$k, cv.tree.birth$dev, type = "b")
```

We can see that the best tree will have 3 nodes.

```
prune.tree.birth <- prune.misclass(tree.birth, best = 3)
plot(prune.tree.birth)
text(prune.tree.birth, pretty = 0)
```




```
summary(prune.tree.birth)
```

```
##
## Classification tree:
## snip.tree(tree = tree.birth, nodes = c(4L, 5L, 3L))
## Variables actually used in tree construction:
## [1] "weight"
## Number of terminal nodes:  3
## Residual mean deviance:  0.322 = 321.1 / 997
## Misclassification error rate: 0.054 = 54 / 1000
```

c) Our tree was pruned down to only 3 nodes. The only variable actually used in the tree is "weight", which was decided by our code. This means that smoking is not in our model as a potential cause of premature births. It does not include many factors that affect premature births, only weight.

d) From our output, we can see that our misclassification rate is 5.4%. This means our model is more accurate than the 9% misclassification rate given in the question.