AEM - Tema 5 - Modelo de Regresión a través de Splines. Trabajo de evaluación.

Jerónimo Carranza Carranza

21 de febrero de 2017

Contents

1 Auto

1) Con el fichero Auto de la librería ISLR:

a) Seleccionar los vehículos con mpg>=13

Proponer un modelo que identifique qué variables influyen en la nueva variable de conteo: m_13=round(mpg-13).

```
library(ISLR)
str(Auto)
```

```
## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 2
```

La variable origin hace referencia a un factor, por lo que se formula como tal:

```
Auto <- within(Auto, {
  origin <- factor(origin, levels=1:3,
                   labels=c("American", "European", "Japanese"))
})
```

Cálculo de la nueva variable entera m_13 y resumen de datos en *dataAuto*:

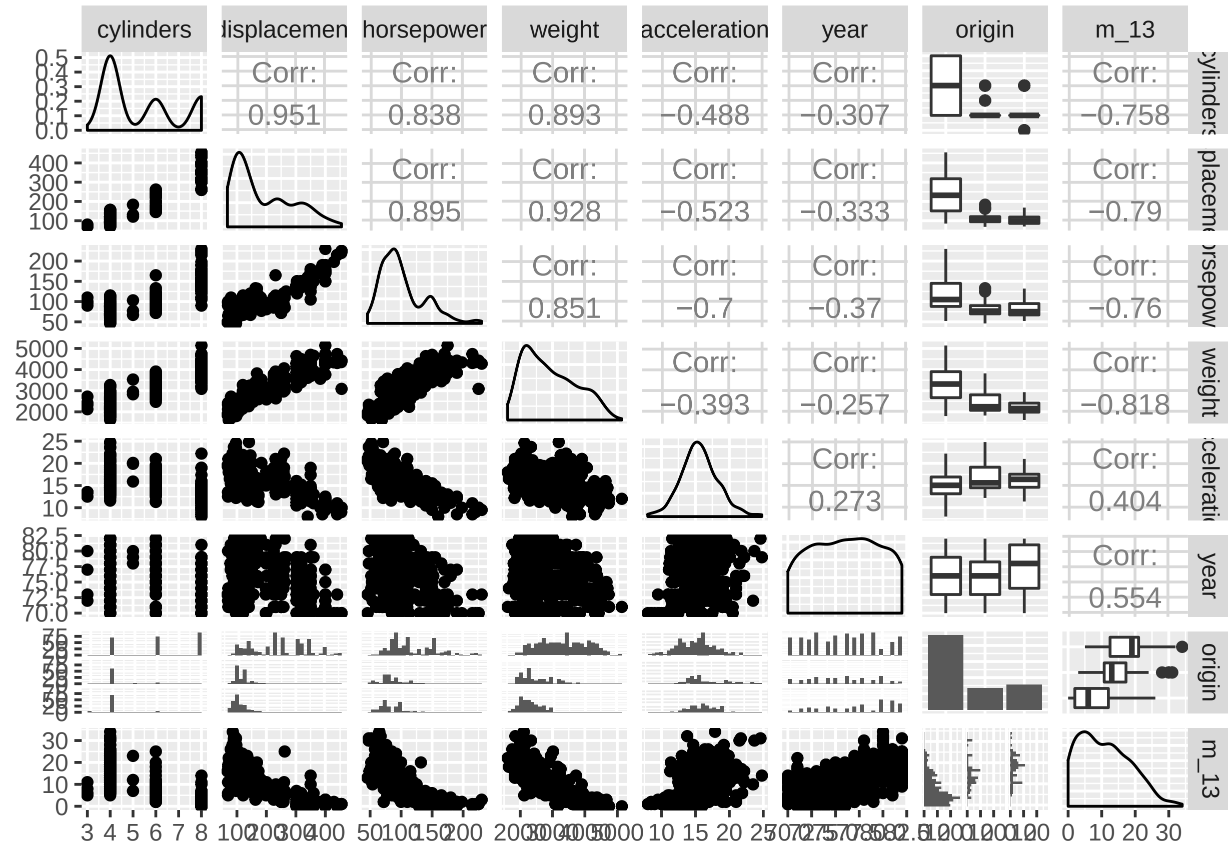
```
dataAuto = Auto[I(Auto$mpg>=13),]
dataAuto = data.frame(dataAuto,m_13=round(dataAuto$mpg-13))
attach(dataAuto)
str(dataAuto)
```

```
## 'data.frame':    379 obs. of  10 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : Factor w/ 3 levels "American","European",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 2
##  $ m_13        : num  5 2 5 3 4 2 1 1 1 2 ...
```

```r
summary(dataAuto)
```

```
##       mpg          cylinders      displacement     horsepower   
##  Min.   :13.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0  
##  1st Qu.:18.00   1st Qu.:4.000   1st Qu.: 99.5   1st Qu.: 75.0  
##  Median :23.00   Median :4.000   Median :141.0   Median : 92.0  
##  Mean   :23.87   Mean   :5.385   Mean   :188.3   Mean   :101.5  
##  3rd Qu.:29.25   3rd Qu.:6.000   3rd Qu.:258.0   3rd Qu.:115.0  
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0  
##                                                                 
##      weight      acceleration        year           origin    
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   American:232  
##  1st Qu.:2220   1st Qu.:14.00   1st Qu.:73.00   European: 68  
##  Median :2745   Median :15.50   Median :76.00   Japanese: 79  
##  Mean   :2921   Mean   :15.63   Mean   :76.12                 
##  3rd Qu.:3512   3rd Qu.:17.20   3rd Qu.:79.00                 
##  Max.   :5140   Max.   :24.80   Max.   :82.00                 
##                                                               
##                 name          m_13      
##  amc matador       :  5   Min.   : 0.00  
##  ford pinto        :  5   1st Qu.: 5.00  
##  toyota corolla    :  5   Median :10.00  
##  amc gremlin       :  4   Mean   :10.85  
##  amc hornet        :  4   3rd Qu.:16.00  
##  chevrolet chevette:  4   Max.   :34.00  
##  (Other)           :352
```

Exploración gráfica de relaciones por pares:



No se han incluido las variables mpg (1) y name (9), la primera porque de ella depende funcionalmente m_13, y name, porque prácticamente es un etiqueta del caso.

Se observan algunas aparentes relaciones negativas intensas entre m_13 y: cylinders (-.758), displacement(-.79), horsepower(-.76), weight(-.818) que a su vez forma un grupo muy correlacionado entre sí positivamente. Por otro lado m_13 se relaciona positivamente, con menor intensidad, con acceleration (.404) y year (.554) y en orden decreciente para "American", "European" y "Japanese" en cuanto a origin.

Modelo lineal general con todos las variables potencialmente relacionadas:

```r
dataAuto.glm01 = glm (m_13 ~ cylinders + displacement +
                      horsepower + weight + acceleration +
                      year + origin,
                    data = dataAuto,
                    family = 'poisson')

summary(dataAuto.glm01)
```

```
##
## Call:
## glm(formula = m_13 ~ cylinders + displacement + horsepower +
##     weight + acceleration + year + origin, family = "poisson",
##     data = dataAuto)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0553  -0.6566  -0.0408   0.4888   3.5582
```

```
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -8.381e-01  4.591e-01  -1.826  0.06791 .
## cylinders       -1.419e-02  3.905e-02  -0.363  0.71641
## displacement    -1.220e-03  1.011e-03  -1.207  0.22725
## horsepower      -4.964e-03  1.804e-03  -2.751  0.00594 **
## weight          -5.192e-04  8.385e-05  -6.193 5.91e-10 ***
## acceleration     3.952e-03  9.383e-03   0.421  0.67365
## year             6.799e-02  4.825e-03  14.091  < 2e-16 ***
## originEuropean  8.189e-02  4.773e-02   1.716  0.08623 .
## originJapanese  3.219e-02  4.481e-02   0.718  0.47254
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2218.51  on 378  degrees of freedom
## Residual deviance:  336.77  on 370  degrees of freedom
## AIC: 1798.1
##
## Number of Fisher Scoring iterations: 5
```

Aparecen como significativas, para $\alpha = 0.05$, las variables: *horsepower*, *weight* y *year*.

Considerando exclusivamente las variables que aparecen como significativas en el modelo *dataAuto.glm01* se formula el modelo *dataAuto.glm02*:

```
dataAuto.glm02 = glm (m_13 ~ horsepower + weight + year,
                    data = dataAuto,
                    family = 'poisson')

summary.glm(dataAuto.glm02)
```

```
##
## Call:
## glm(formula = m_13 ~ horsepower + weight + year, family = "poisson",
##     data = dataAuto)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2847  -0.6696  -0.1176   0.4693   3.2591
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.4835743  0.3672210  -1.317    0.188
## horsepower  -0.0066108  0.0012038  -5.492 3.98e-08 ***
## weight      -0.0006371  0.0000471 -13.528  < 2e-16 ***
## year         0.0672117  0.0045366  14.815  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 2218.51  on 378  degrees of freedom
## Residual deviance:  349.78  on 375  degrees of freedom
## AIC: 1801.1
##
## Number of Fisher Scoring iterations: 4
```

```r
anova(dataAuto.glm01,dataAuto.glm02)
```

```
## Analysis of Deviance Table
##
## Model 1: m_13 ~ cylinders + displacement + horsepower + weight + acceleration +
##     year + origin
## Model 2: m_13 ~ horsepower + weight + year
##   Resid. Df Resid. Dev Df Deviance
## 1       370     336.77
## 2       375     349.78 -5  -13.014
```

Se puede apreciar que aunque el modelo 01 marque exclusivamente como significativas (para el α considerado) sólo tres variables, las otras tienen cierta influencia que hace que se incremente tanto la deviación como el indice AIC cuando se eliminan todas ellas, señalando con ello que no se compensa globlamente la reducción de complejidad del modelo.

Considerando la eliminación de variables una a una comprobamos el efecto singular de cada una de ellas:

```r
dataAuto.glm_cylinders = update(dataAuto.glm01, . ~ . -cylinders)
dataAuto.glm_displacement = update(dataAuto.glm01, . ~ . -displacement)
dataAuto.glm_acceleration = update(dataAuto.glm01, . ~ . -acceleration)
dataAuto.glm_origin = update(dataAuto.glm01, . ~ . -origin)
dataAuto.glm_horsepower = update(dataAuto.glm01, . ~ . -horsepower)
dataAuto.glm_weight = update(dataAuto.glm01, . ~ . -weight)
dataAuto.glm_year = update(dataAuto.glm01, . ~ . -year)

anova(dataAuto.glm01, dataAuto.glm_cylinders, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: m_13 ~ cylinders + displacement + horsepower + weight + acceleration +
##     year + origin
## Model 2: m_13 ~ displacement + horsepower + weight + acceleration + year +
##     origin
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       370     336.77
## 2       371     336.90 -1 -0.13212   0.7162
```

```r
anova(dataAuto.glm01, dataAuto.glm_displacement, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: m_13 ~ cylinders + displacement + horsepower + weight + acceleration +
##     year + origin
## Model 2: m_13 ~ cylinders + horsepower + weight + acceleration + year +
##     origin
```

```
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       370     336.77
## 2       371     338.23 -1  -1.4614   0.2267
```

```r
anova(dataAuto.glm01, dataAuto.glm_acceleration, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: m_13 ~ cylinders + displacement + horsepower + weight + acceleration +
##     year + origin
## Model 2: m_13 ~ cylinders + displacement + horsepower + weight + year +
##     origin
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       370     336.77
## 2       371     336.95 -1 -0.17708   0.6739
```

```r
anova(dataAuto.glm01, dataAuto.glm_origin, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: m_13 ~ cylinders + displacement + horsepower + weight + acceleration +
##     year + origin
## Model 2: m_13 ~ cylinders + displacement + horsepower + weight + acceleration +
##     year
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       370     336.77
## 2       372     339.77 -2  -3.0021   0.2229
```

```r
anova(dataAuto.glm01, dataAuto.glm_horsepower, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: m_13 ~ cylinders + displacement + horsepower + weight + acceleration +
##     year + origin
## Model 2: m_13 ~ cylinders + displacement + weight + acceleration + year +
##     origin
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       370     336.77
## 2       371     344.43 -1  -7.6622 0.005639 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(dataAuto.glm01, dataAuto.glm_weight, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: m_13 ~ cylinders + displacement + horsepower + weight + acceleration +
##     year + origin
## Model 2: m_13 ~ cylinders + displacement + horsepower + acceleration +
##     year + origin
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
```

```
## 1       370     336.77
## 2       371     374.27 -1  -37.506 9.114e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(dataAuto.glm01, dataAuto.glm_year, test = 'Chisq')
```

```
## Analysis of Deviance Table
##
## Model 1: m_13 ~ cylinders + displacement + horsepower + weight + acceleration +
##     year + origin
## Model 2: m_13 ~ cylinders + displacement + horsepower + weight + acceleration +
##     origin
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       370     336.77
## 2       371     539.00 -1  -202.23 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Los resultados confirman que son las tres variables señaladas anteriormente: *horsepower*, *weight* y *year*; las únicas influyentes de forma significativa ($\alpha = 0.05$) en *m_13*.

```r
detach(dataAuto)
```

2 College

2) Con el fichero College de la librería ISRL:

Proponer un modelo gam para la variable Grad.Rate eligiendo la función que considere adecuada para cada variable predictora.

```r
library(ISLR)
attach(College)
str(College)
```

```
## 'data.frame':    777 obs. of  18 variables:
##  $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

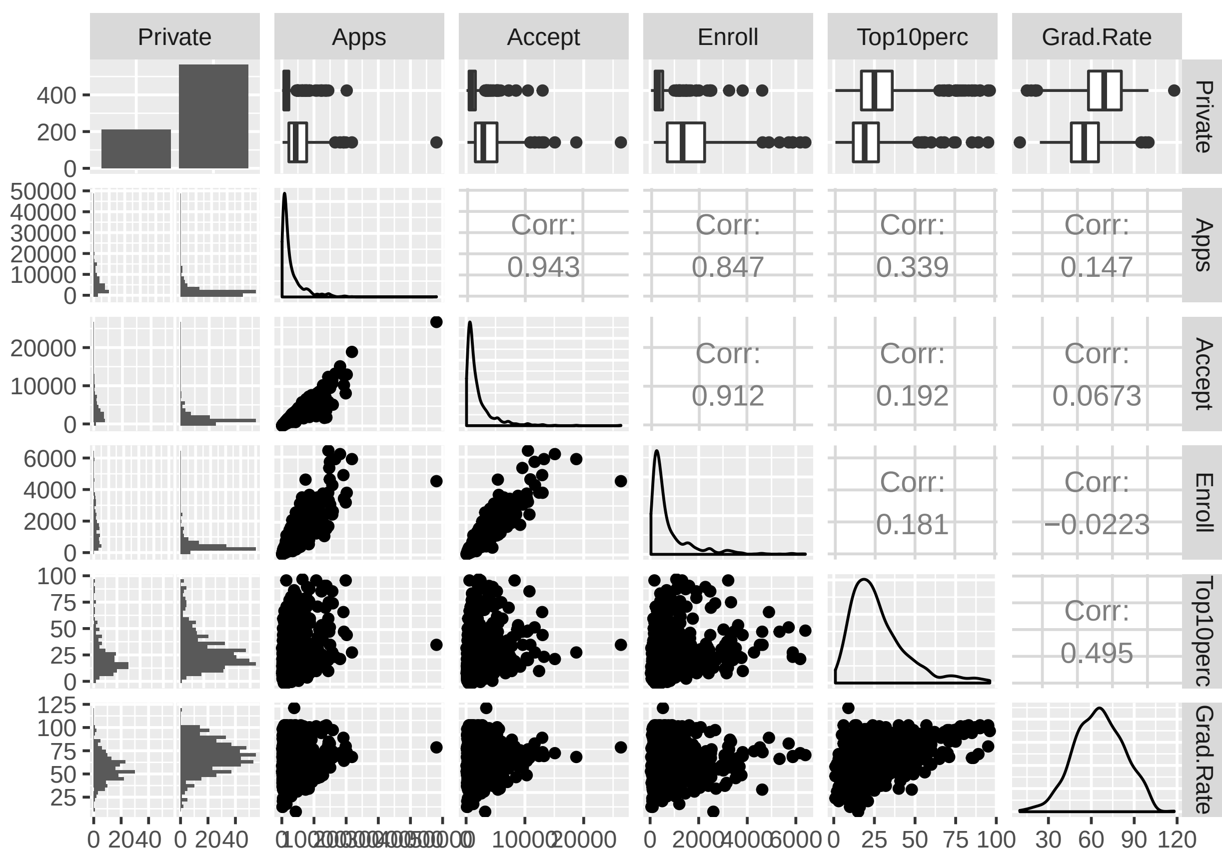
```r
summary(College)
```

```
##  Private        Apps           Accept          Enroll       Top10perc    
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00  
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00  
##            Median : 1558   Median : 1110   Median : 434   Median :23.00  
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56  
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00  
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00  
##    Top25perc      F.Undergrad     P.Undergrad         Outstate    
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340  
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320  
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990  
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441  
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925  
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700  
##    Room.Board       Books           Personal         PhD        
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00  
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00  
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00  
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66  
```

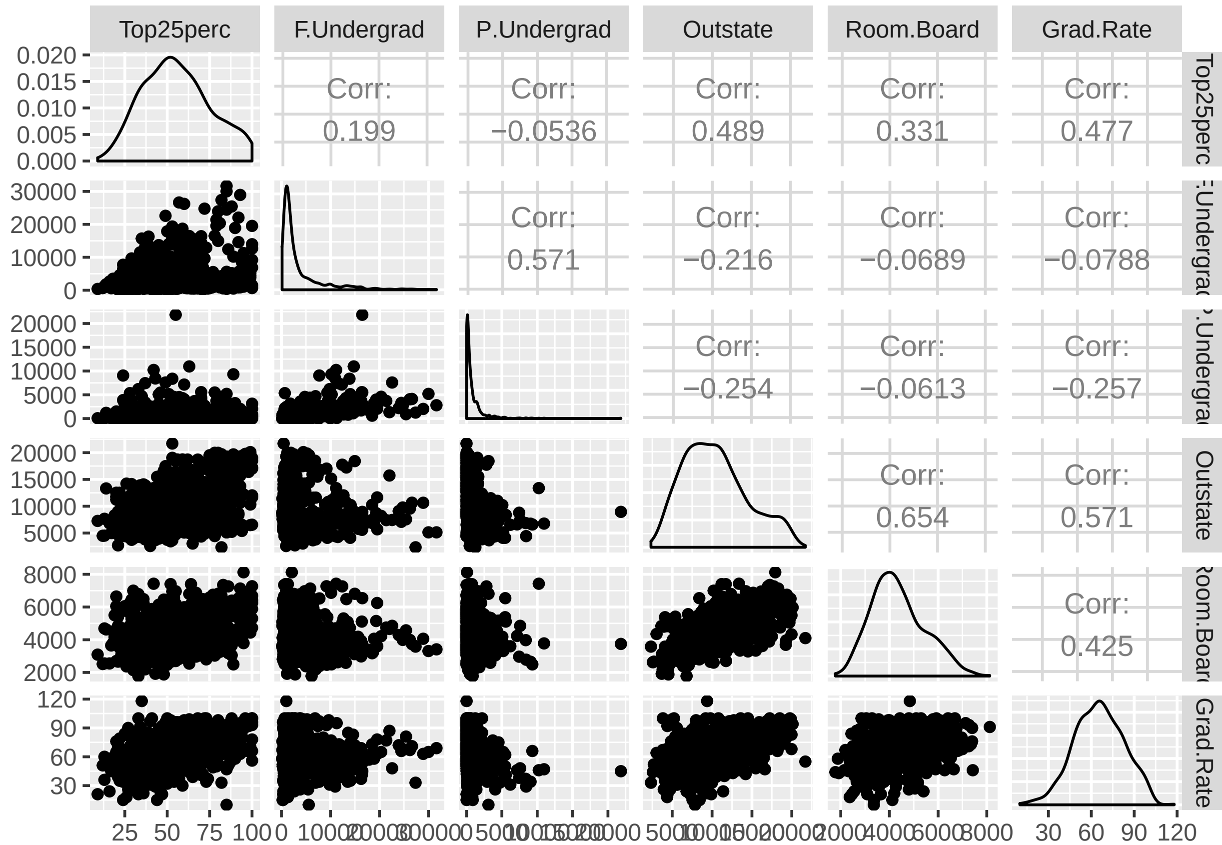
```
##   3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##   Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##      Terminal       S.F.Ratio      perc.alumni        Expend
##   Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##   1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##   Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##   Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##   3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##   Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##     Grad.Rate
##   Min.   : 10.00
##   1st Qu.: 53.00
##   Median : 65.00
##   Mean   : 65.46
##   3rd Qu.: 78.00
##   Max.   :118.00
```

Exploración gráfica:

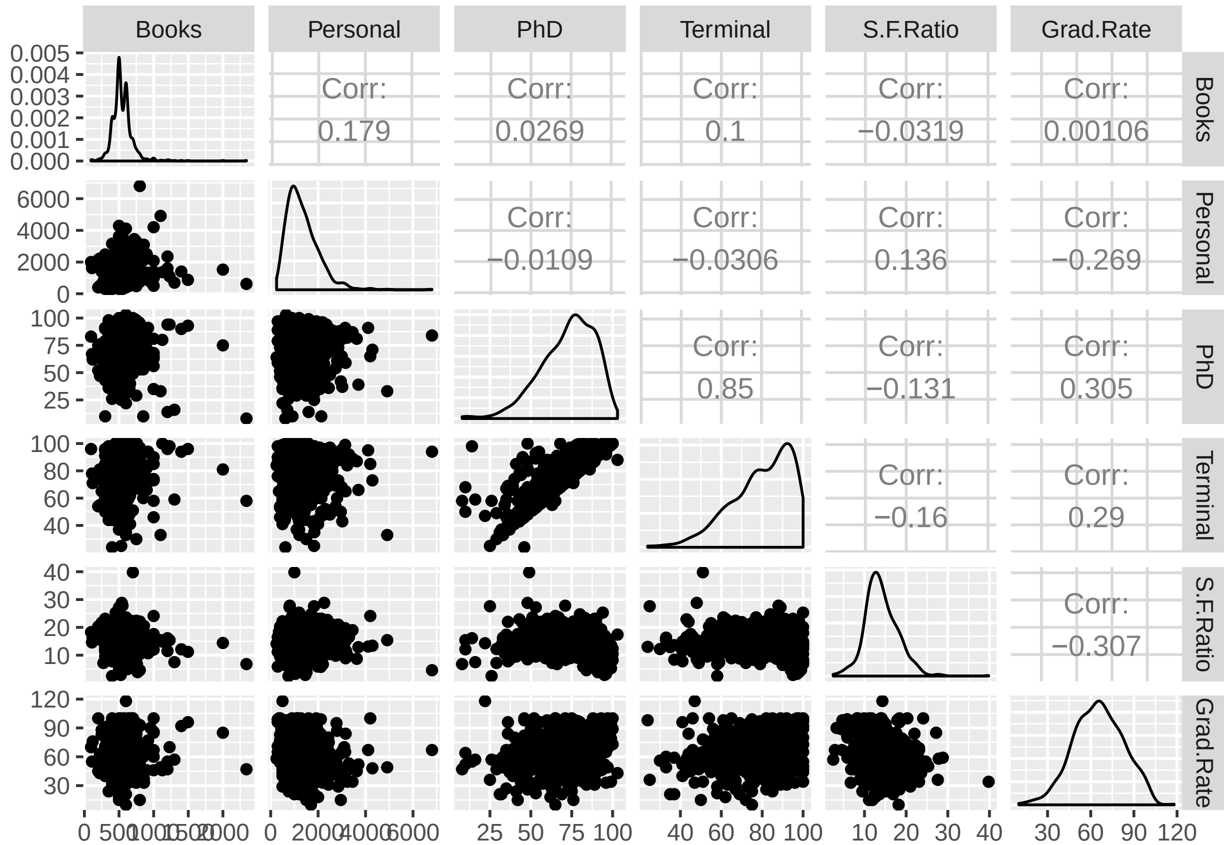
```r
library(GGally)
ggpairs(College[,c(1:5,18)])
```



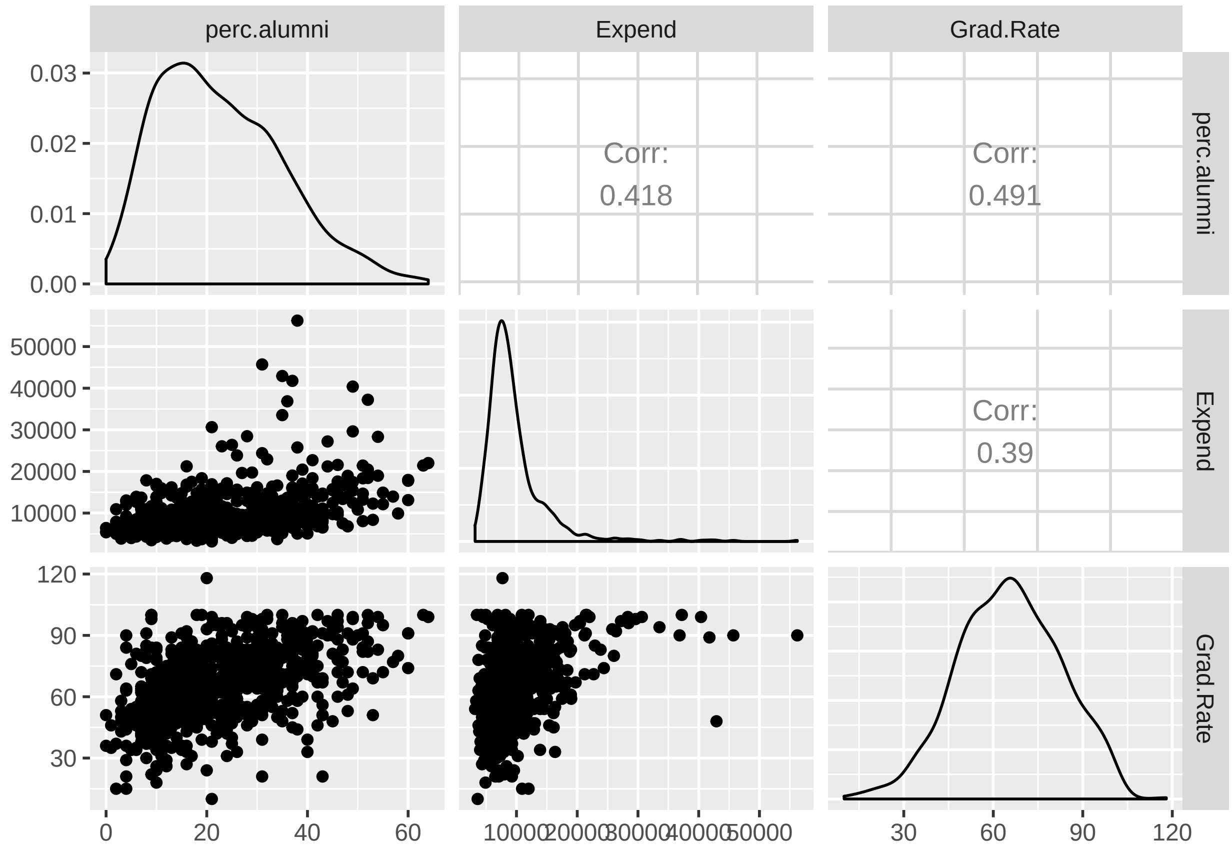

```r
ggpairs(College[,c(6:10,18)])
```




```
ggpairs(College[,c(11:15,18)])
```

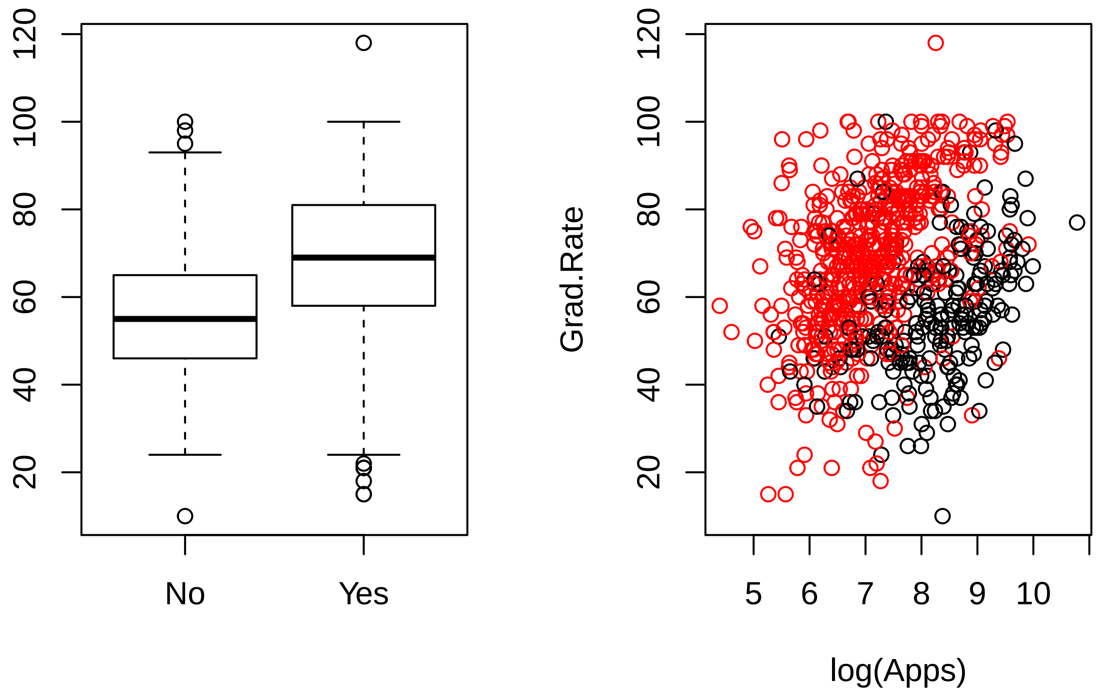



```r
ggpairs(College[,c(16:17,18)])
```

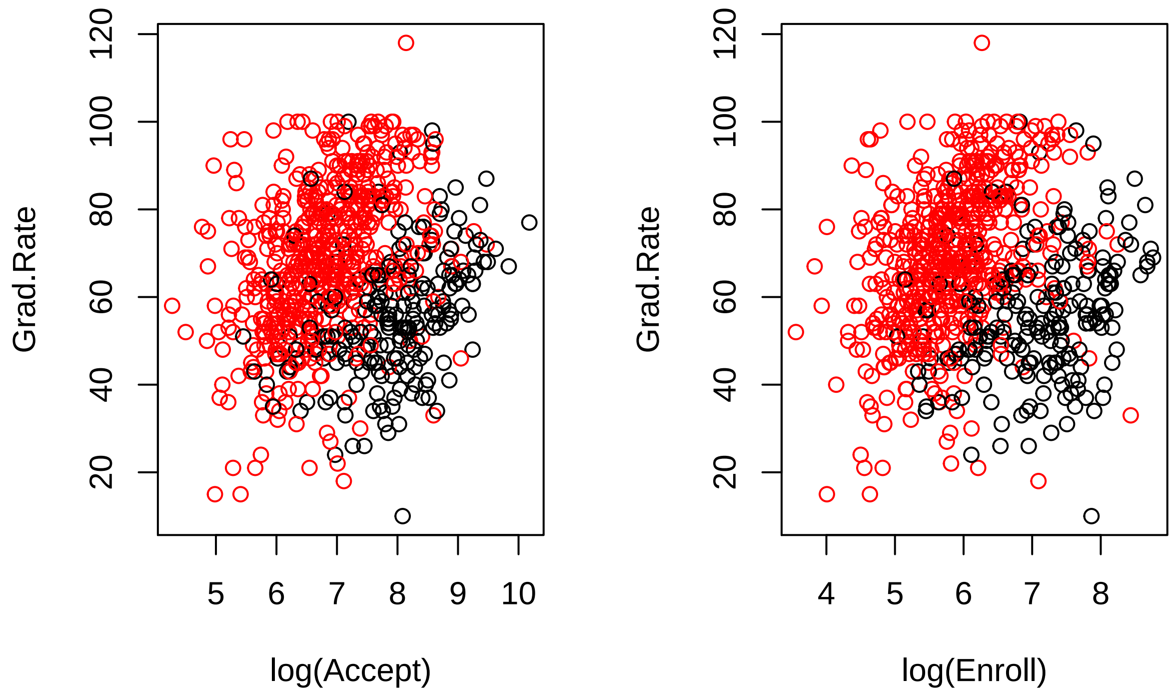


Algunos ajustes visuales preliminares sobre las variables predictoras:

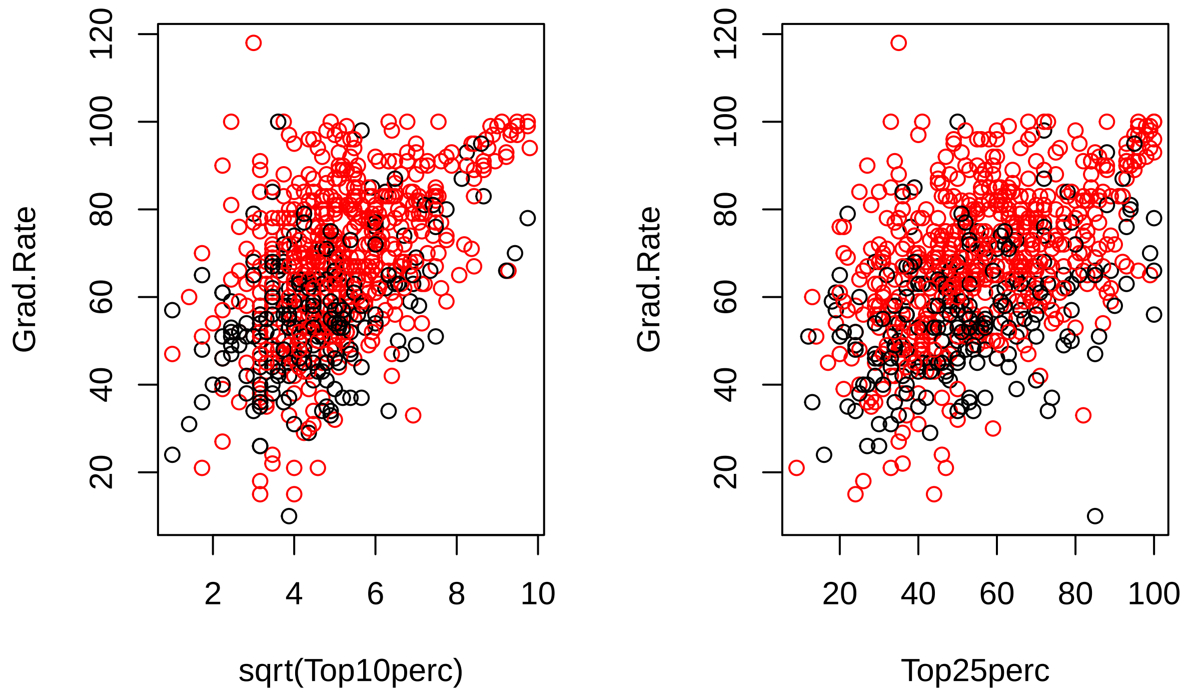
```
par(mfcol=c(1,2))
plot(Private, Grad.Rate)
plot(log(Apps),Grad.Rate,col=Private)
```



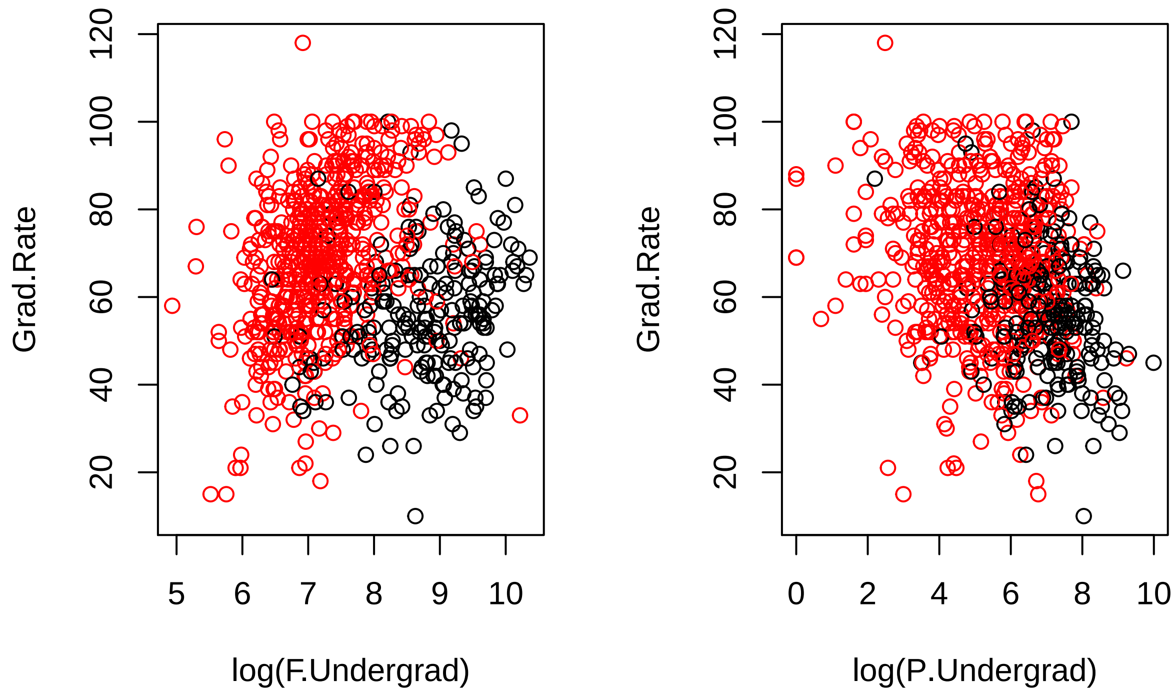

```r
plot(log(Accept),Grad.Rate,col=Private)
plot(log(Enroll),Grad.Rate,col=Private)
```



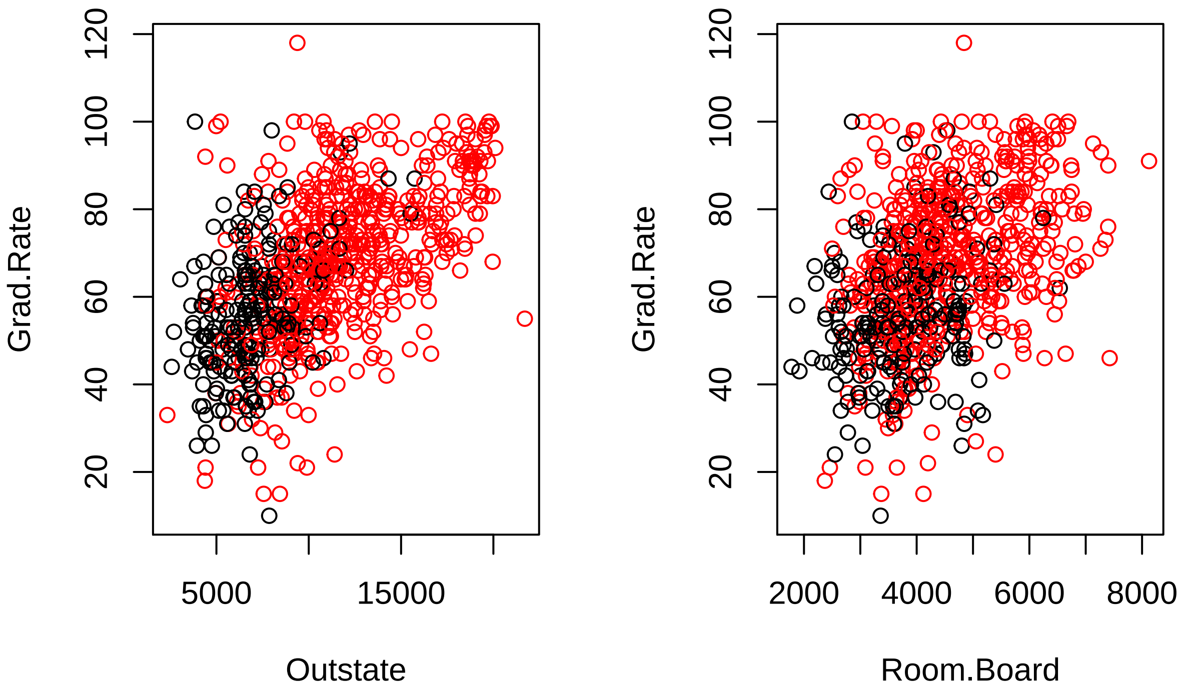

```r
plot(sqrt(Top10perc),Grad.Rate,col=Private)
plot(Top25perc,Grad.Rate,col=Private)
```



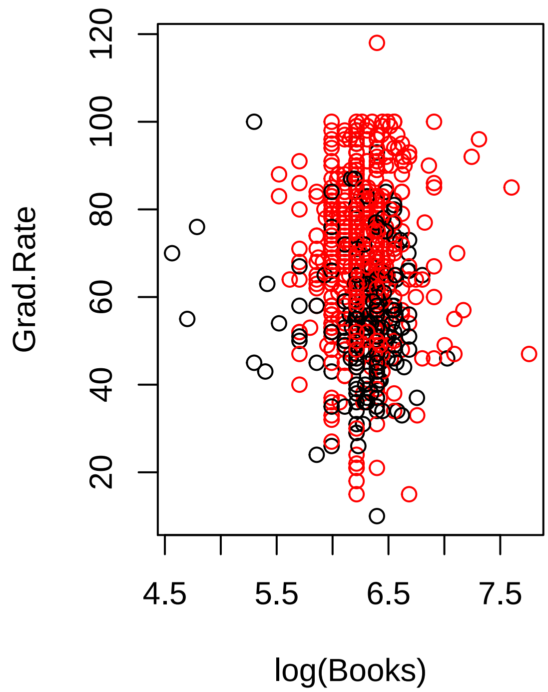

```
plot(log(F.Undergrad),Grad.Rate,col=Private)
plot(log(P.Undergrad),Grad.Rate,col=Private)
```

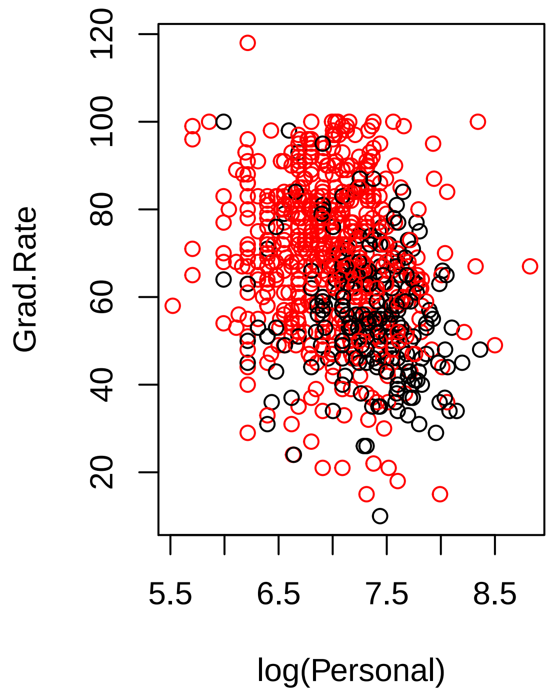



```r
plot(Outstate,Grad.Rate,col=Private)
plot(Room.Board,Grad.Rate,col=Private)
```

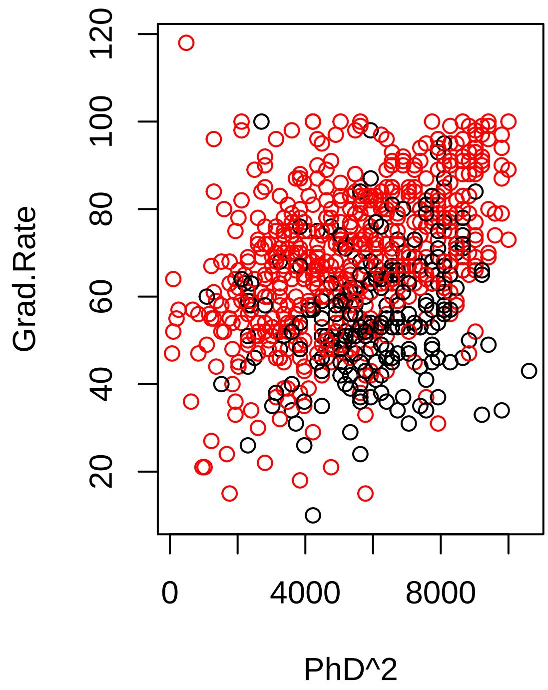


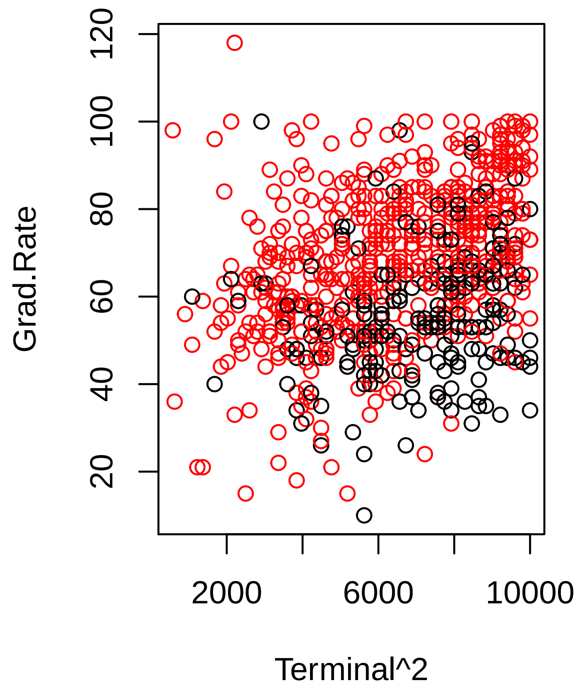

```
plot(log(Books),Grad.Rate,col=Private)
plot(log(Personal),Grad.Rate,col=Private)
```



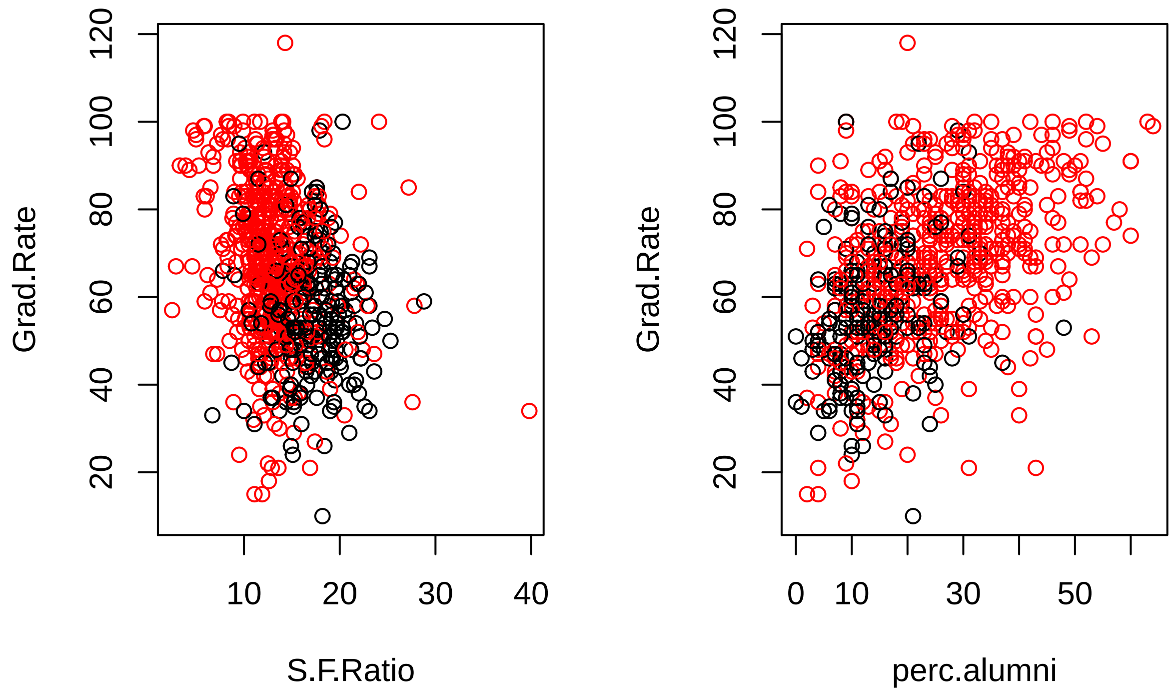



```
plot(PhD**2,Grad.Rate,col=Private)
plot(Terminal**2,Grad.Rate,col=Private)
```

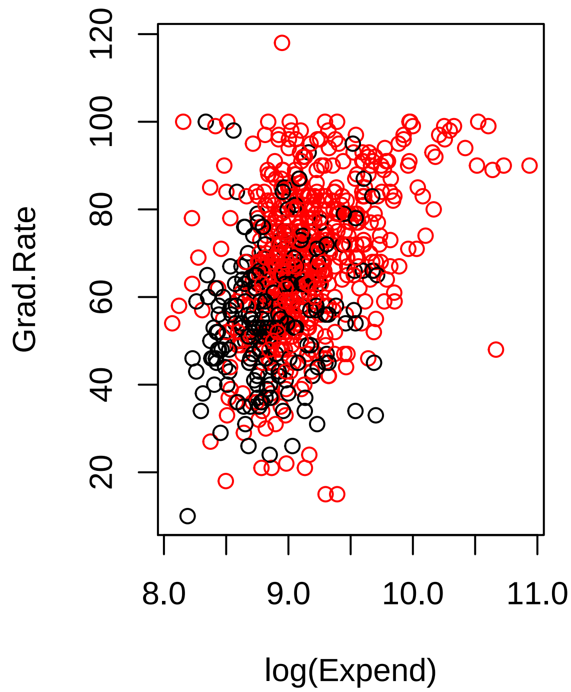





```
plot(S.F.Ratio,Grad.Rate,col=Private)
plot(perc.alumni,Grad.Rate,col=Private)
```




```
plot(log(Expend),Grad.Rate,col=Private)
par(mfcol=c(1,1))
```




```r
library(gam)
```

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

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

```
## Loaded gam 1.14
```

```r
Grad.Rate.gam_01 = gam(Grad.Rate ~ ., data=College)
summary(Grad.Rate.gam_01)
```

```
##
## Call: gam(formula = Grad.Rate ~ ., data = College)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -53.897  -7.132  -0.292   7.213  54.056
##
## (Dispersion Parameter for gaussian family taken to be 162.4439)
##
##     Null Deviance: 228977.2 on 776 degrees of freedom
## Residual Deviance: 123294.9 on 759 degrees of freedom
## AIC: 6180.007
##
## Number of Local Scoring Iterations: 2
```

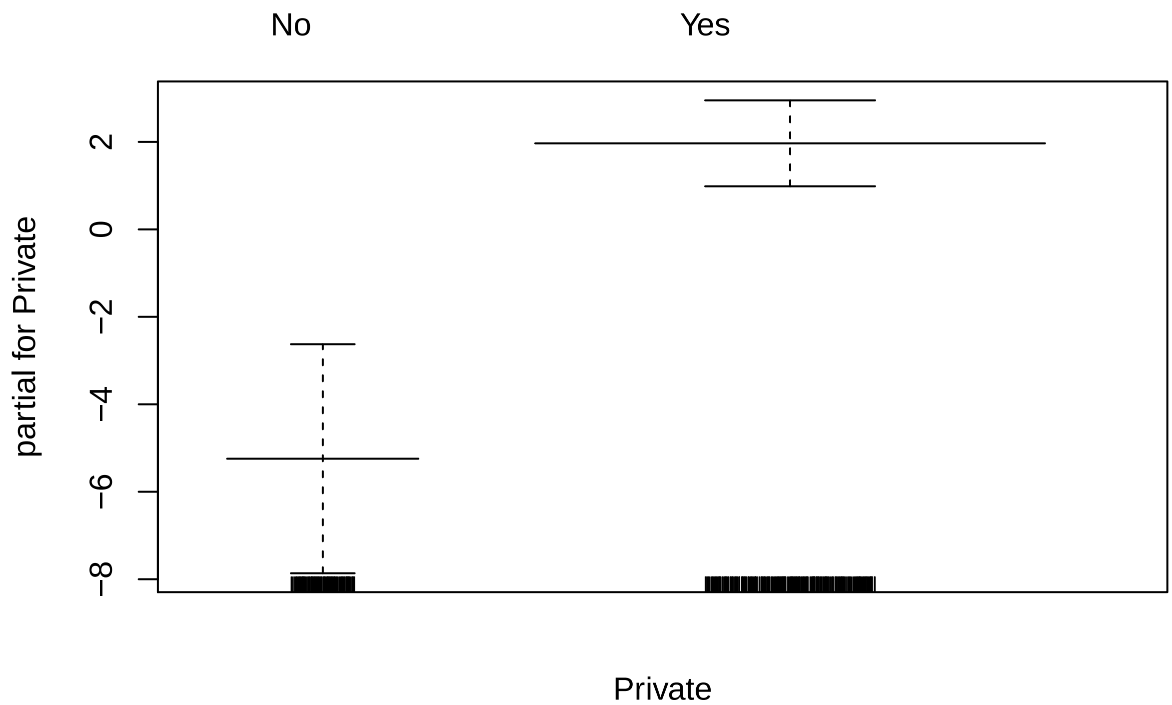
```
##
## Anova for Parametric Effects
##              Df Sum Sq Mean Sq  F value    Pr(>F)
## Private       1  25876 25875.6 159.2894 < 2.2e-16 ***
## Apps          1  24007 24006.9 147.7856 < 2.2e-16 ***
## Accept        1   3160  3159.8  19.4515 1.181e-05 ***
## Enroll        1    247   246.9   1.5198 0.2180317
## Top10perc     1  24486 24486.1 150.7359 < 2.2e-16 ***
## Top25perc     1   2089  2089.2  12.8611 0.0003570 ***
## F.Undergrad   1   2195  2195.2  13.5138 0.0002535 ***
## P.Undergrad   1   2731  2730.8  16.8104 4.576e-05 ***
## Outstate      1  10483 10483.2  64.5345 3.620e-15 ***
## Room.Board    1    897   897.5   5.5247 0.0190047 *
## Books         1    634   633.6   3.9006 0.0486314 *
## Personal      1   1559  1559.1   9.5979 0.0020199 **
## PhD           1    217   216.6   1.3335 0.2485450
## Terminal      1    166   165.6   1.0195 0.3129548
## S.F.Ratio     1    282   281.8   1.7350 0.1881724
## perc.alumni   1   5230  5230.1  32.1966 1.982e-08 ***
## Expend        1   1424  1424.2   8.7671 0.0031628 **
## Residuals   759 123295   162.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

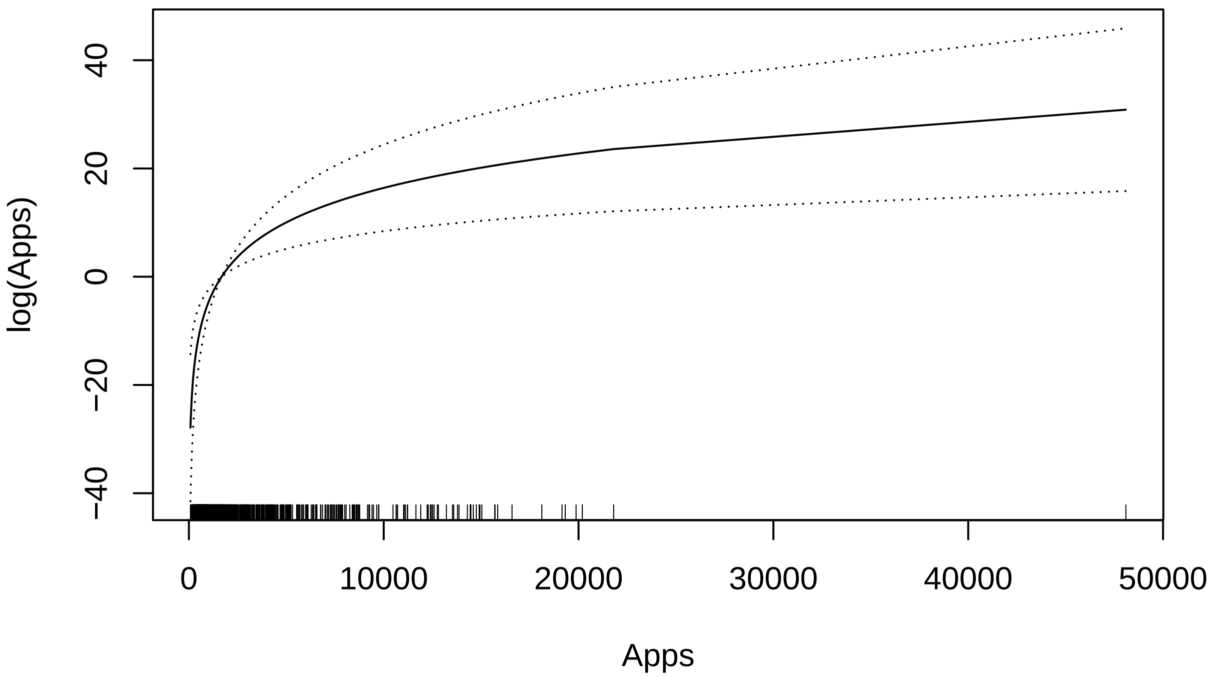
```r
#plot(Grad.Rate.gam_01,se=TRUE)

Grad.Rate.gam_02 = gam(Grad.Rate ~ Private + log(Apps) + log(Accept) +
                         log(Enroll) + sqrt(Top10perc) + Top25perc +
                         log(F.Undergrad) + log(P.Undergrad) +
                         Outstate + Room.Board + log(Books) +
                         log(Personal) + s(PhD,4) + s(Terminal,4) +
                         S.F.Ratio + perc.alumni + log(Expend),
                       data=College)
summary(Grad.Rate.gam_02)
```

```
##
## Call: gam(formula = Grad.Rate ~ Private + log(Apps) + log(Accept) +
##     log(Enroll) + sqrt(Top10perc) + Top25perc + log(F.Undergrad) +
##     log(P.Undergrad) + Outstate + Room.Board + log(Books) + log(Personal) +
##     s(PhD, 4) + s(Terminal, 4) + S.F.Ratio + perc.alumni + log(Expend),
##     data = College)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -57.5766  -7.3754  -0.3216   7.2068  50.4543
##
## (Dispersion Parameter for gaussian family taken to be 159.2514)
##
##     Null Deviance: 228977.2 on 776 degrees of freedom
## Residual Deviance: 119916.3 on 753.0003 degrees of freedom
## AIC: 6170.418
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
```

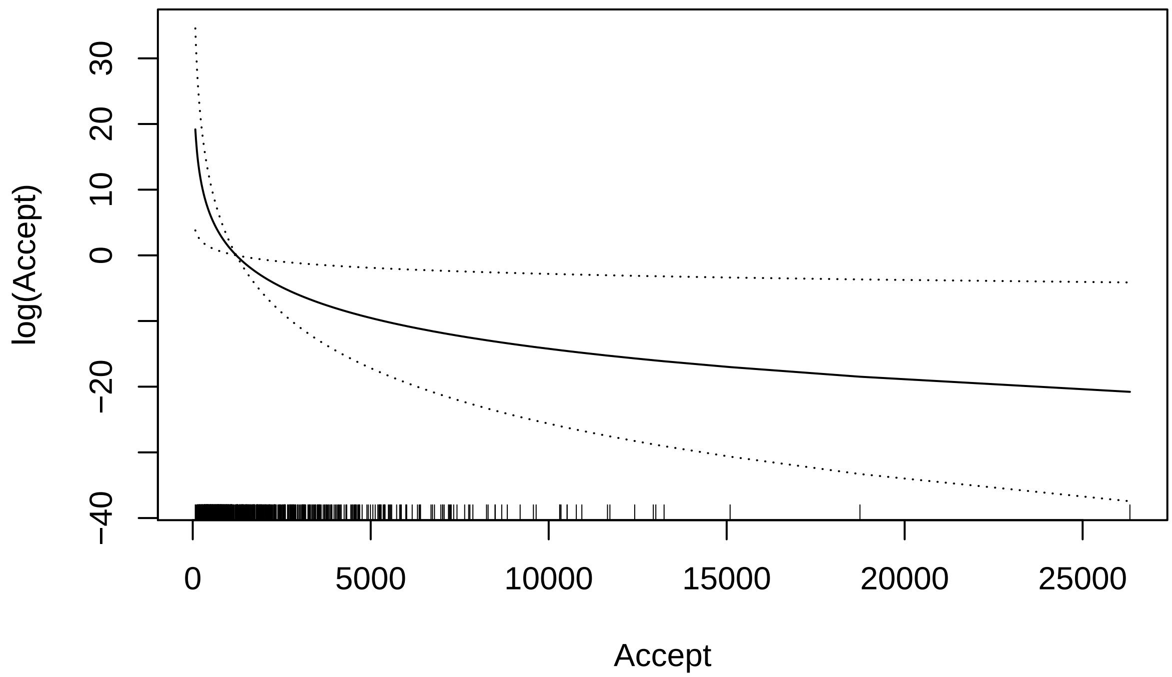
```
##                   Df Sum Sq Mean Sq  F value    Pr(>F)    
## Private            1  25970   25970 163.0768 < 2.2e-16 ***
## log(Apps)          1  38290   38290 240.4398 < 2.2e-16 ***
## log(Accept)        1   5125    5125  32.1821 2.002e-08 ***
## log(Enroll)        1    457     457   2.8697 0.0906777 .  
## sqrt(Top10perc)    1  14474   14474  90.8906 < 2.2e-16 ***
## Top25perc          1    943     943   5.9207 0.0151955 *  
## log(F.Undergrad)   1    471     471   2.9565 0.0859432 .  
## log(P.Undergrad)   1    941     941   5.9088 0.0152975 *  
## Outstate           1   7183    7183  45.1022 3.685e-11 ***
## Room.Board         1    564     564   3.5426 0.0601956 .  
## log(Books)         1   2131    2131  13.3812 0.0002719 ***
## log(Personal)      1   2297    2297  14.4244 0.0001577 ***
## s(PhD, 4)          1      8       8   0.0513 0.8208567    
## s(Terminal, 4)     1    248     248   1.5565 0.2125597    
## S.F.Ratio          1     37      37   0.2329 0.6295352    
## perc.alumni        1   5819    5819  36.5417 2.350e-09 ***
## log(Expend)        1   3557    3557  22.3326 2.736e-06 ***
## Residuals        753 119916     159                       
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Anova for Nonparametric Effects
##                  Npar Df Npar F  Pr(F)
## (Intercept)                           
## Private                               
## log(Apps)                             
## log(Accept)                           
## log(Enroll)                           
## sqrt(Top10perc)                       
## Top25perc                             
## log(F.Undergrad)                      
## log(P.Undergrad)                      
## Outstate                              
## Room.Board                            
## log(Books)                            
## log(Personal)                         
## s(PhD, 4)              3 1.4126 0.2378
## s(Terminal, 4)         3 1.3433 0.2591
## S.F.Ratio                             
## perc.alumni                           
## log(Expend)
```

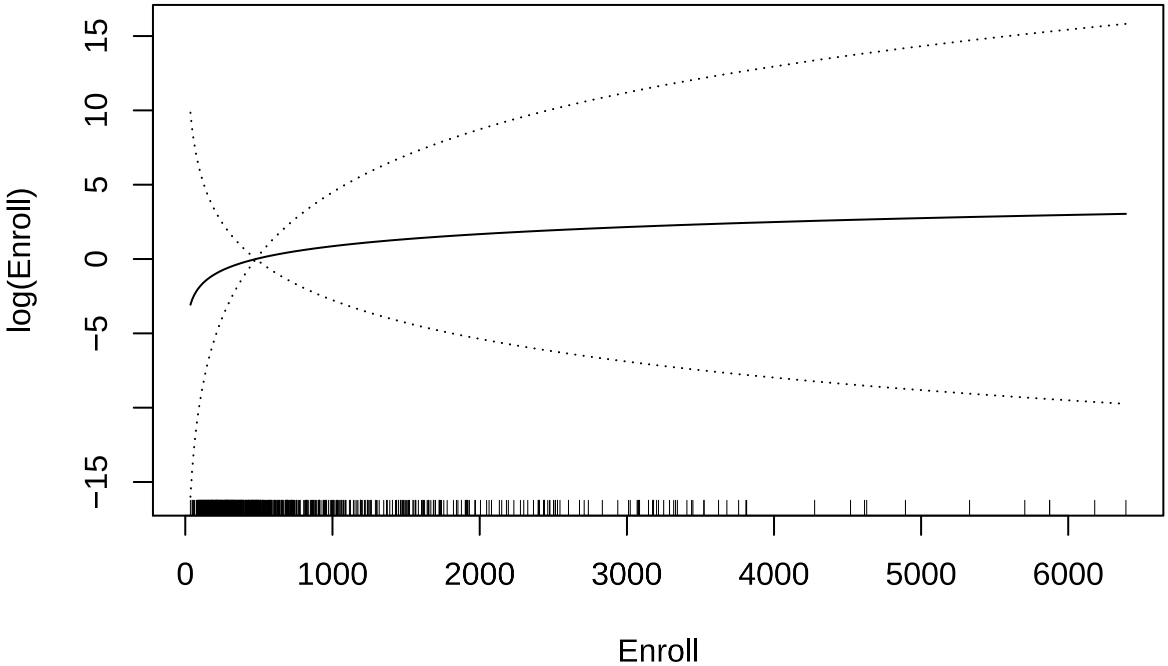
```r
plot(Grad.Rate.gam_02,se=TRUE)
```

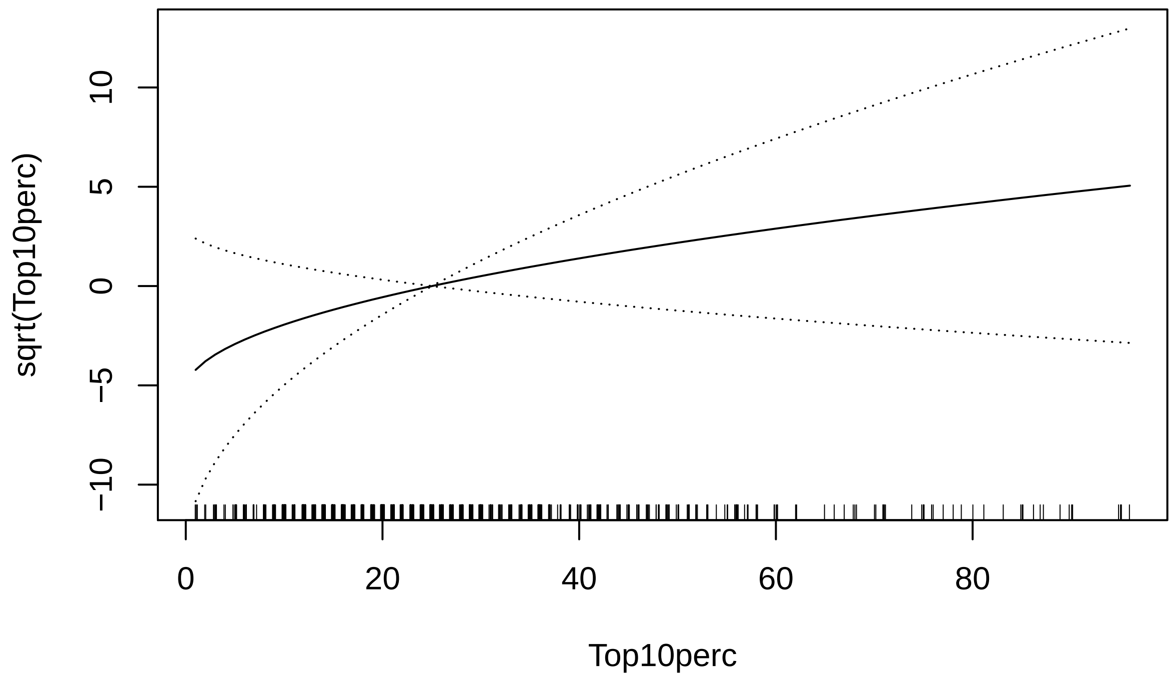

No
Yes
2
0
−2
−4
−6
−8
partial for Private
Private



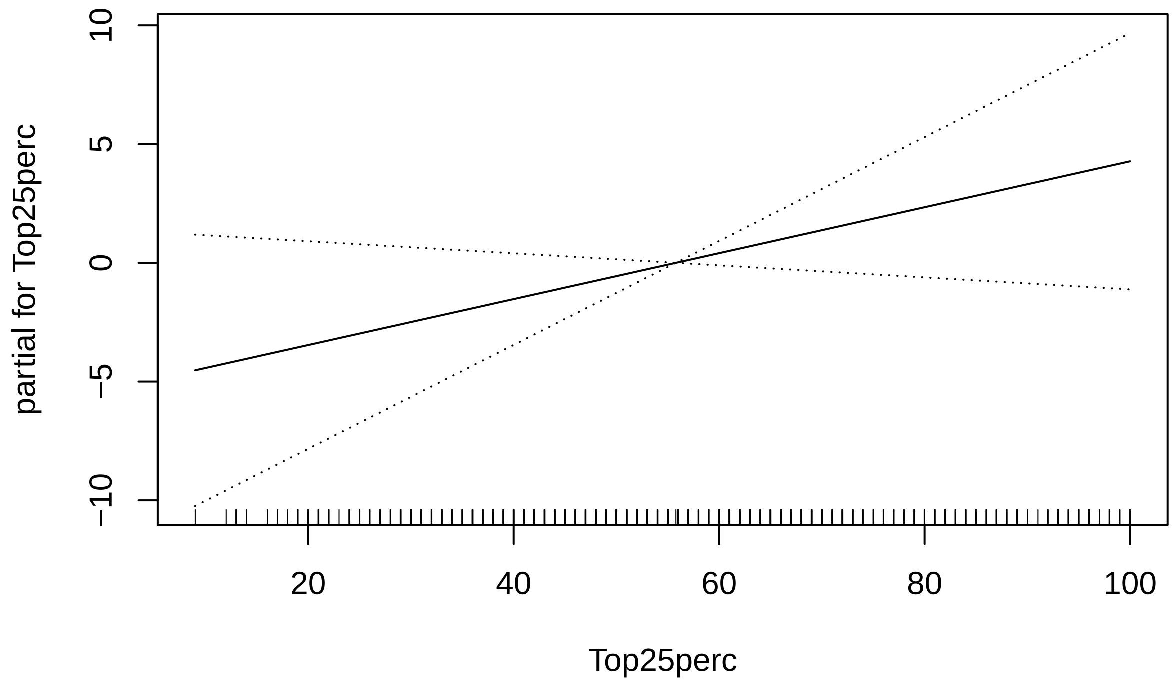
log(Apps)
40
20
0
−20
−40
0
10000
20000
30000
40000
50000
Apps

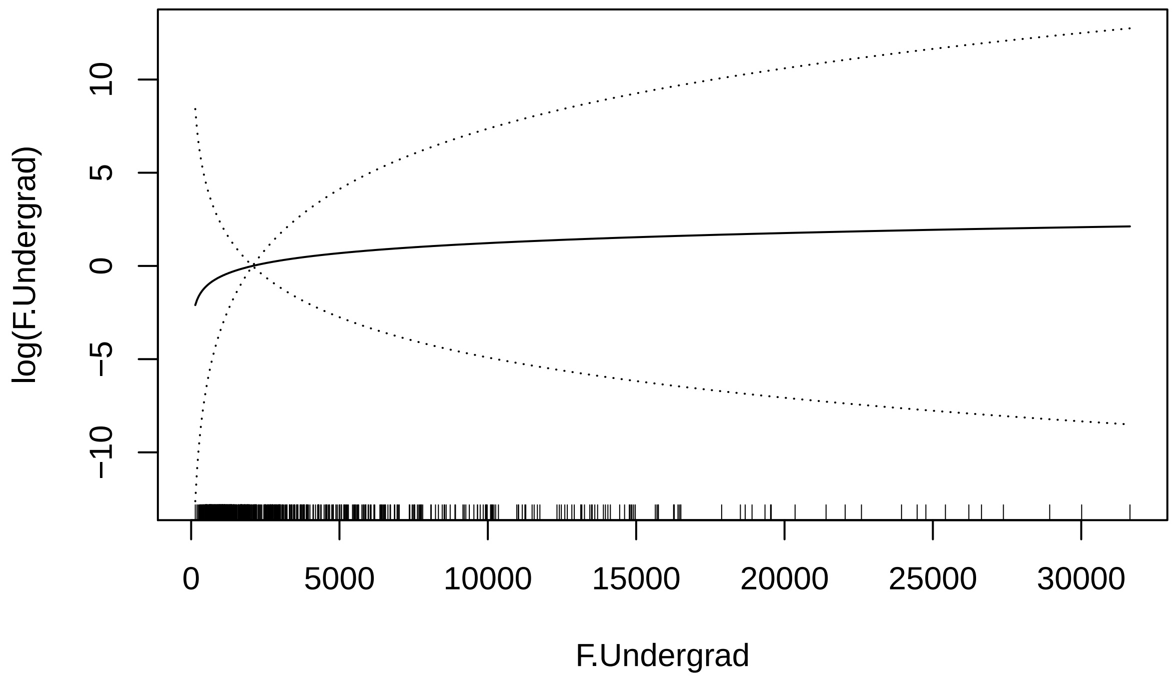

log(Accept)
30
20
10
0
−20
−40
0
5000
10000
15000
20000
25000
Accept


log(Enroll)
15
10
5
0
−5
−15
0
1000
2000
3000
4000
5000
6000
Enroll

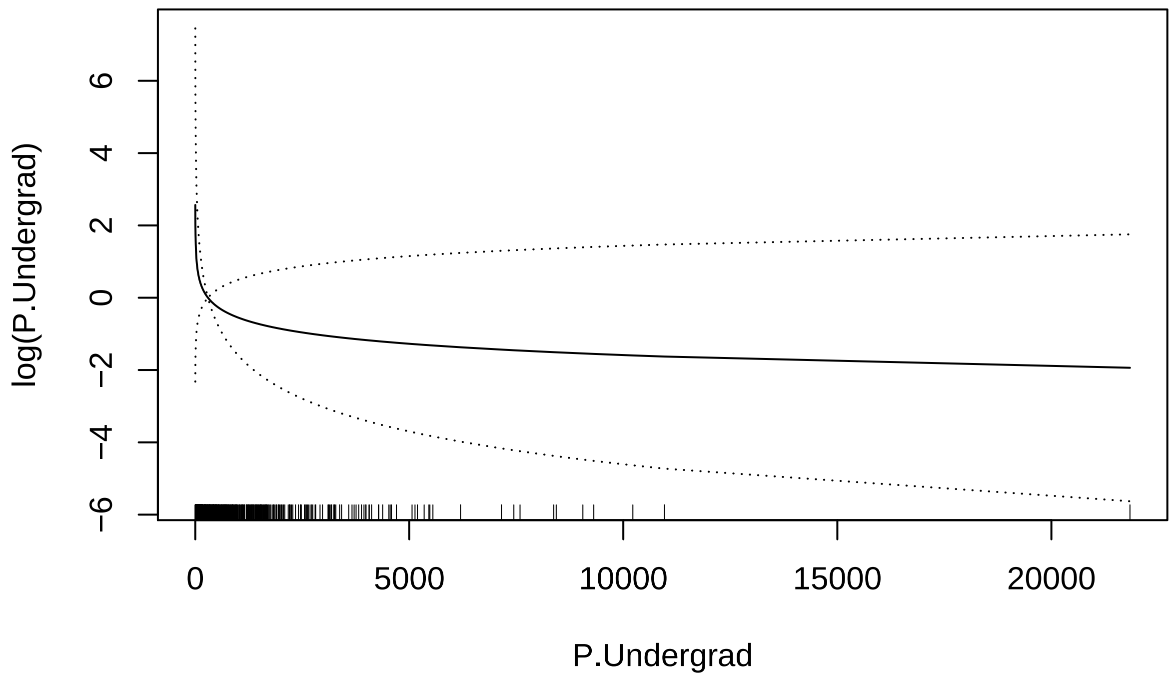


sqrt(Top10perc)
10
5
0
−5
−10
0
20
40
60
80
Top10perc

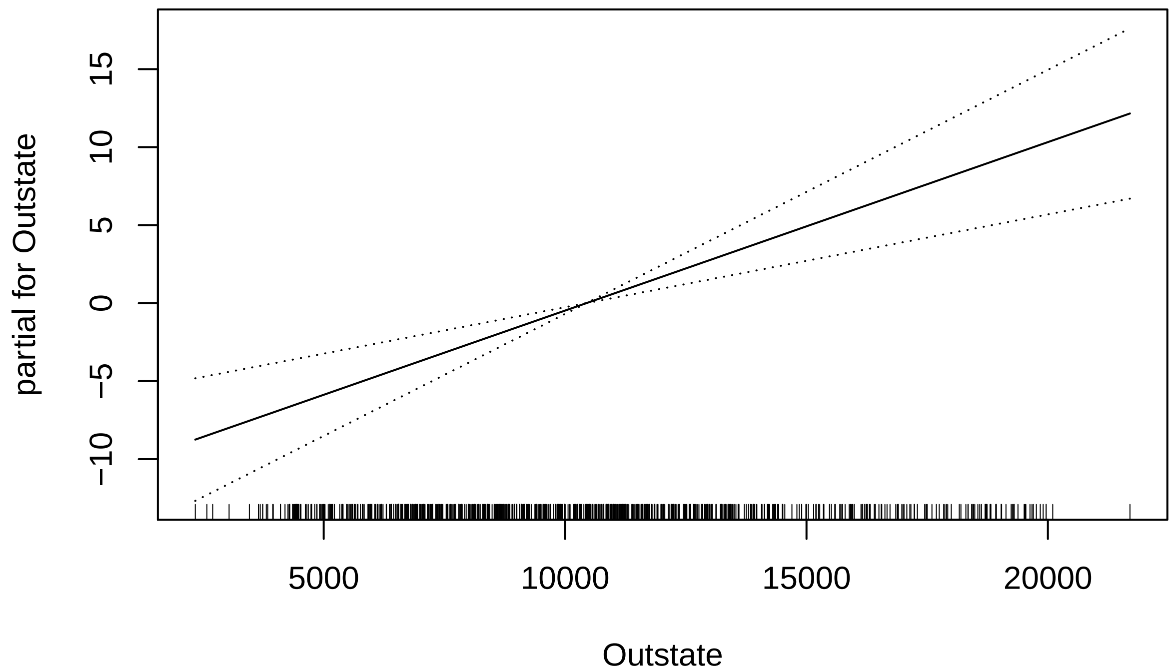

partial for Top25perc
10
5
0
−5
−10
20
40
60
80
100
Top25perc

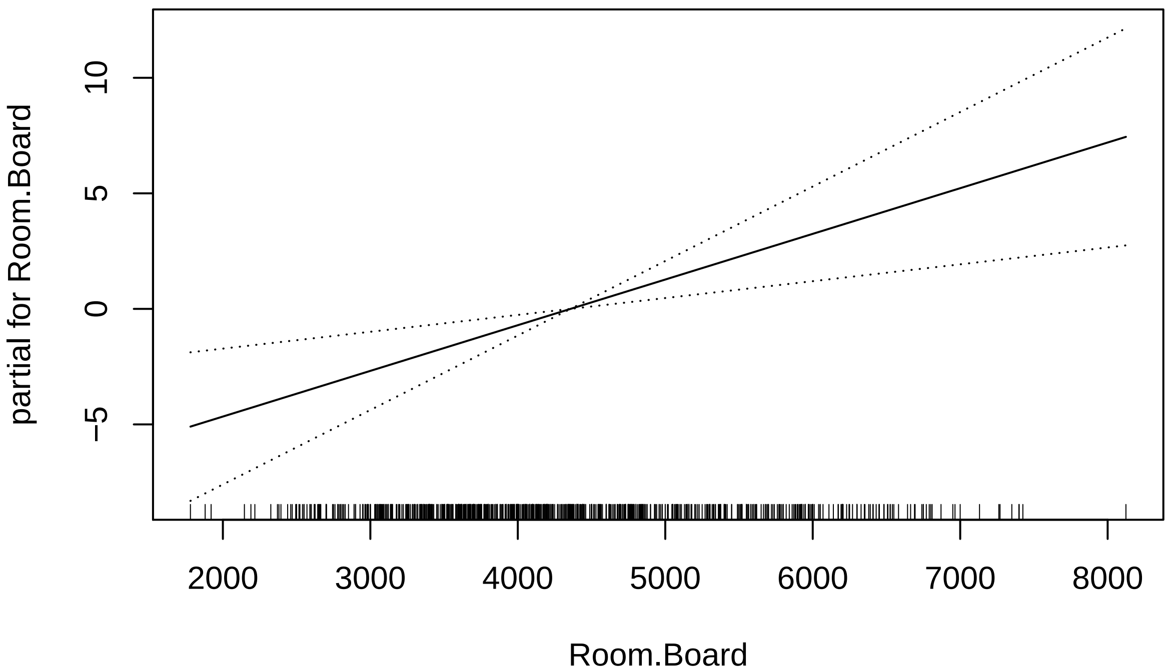


log(F.Undergrad)
10
5
0
−5
−10
0
5000
10000
15000
20000
25000
30000
F.Undergrad

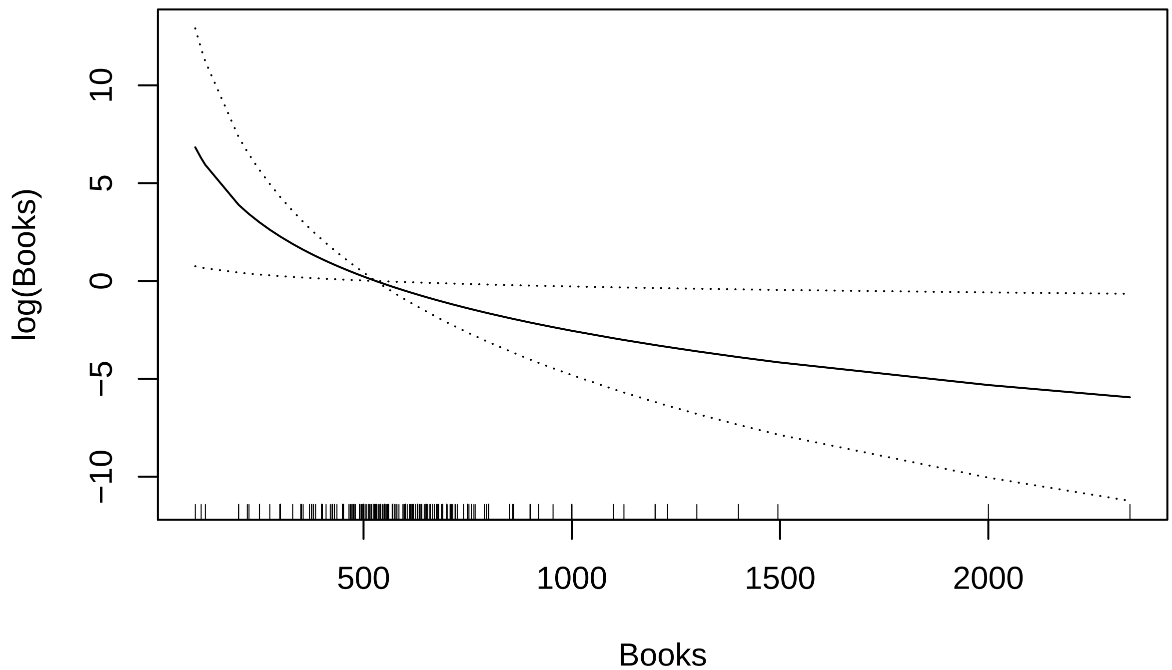


log(P.Undergrad)
6
4
2
0
−2
−4
−6
0
5000
10000
15000
20000
P.Undergrad

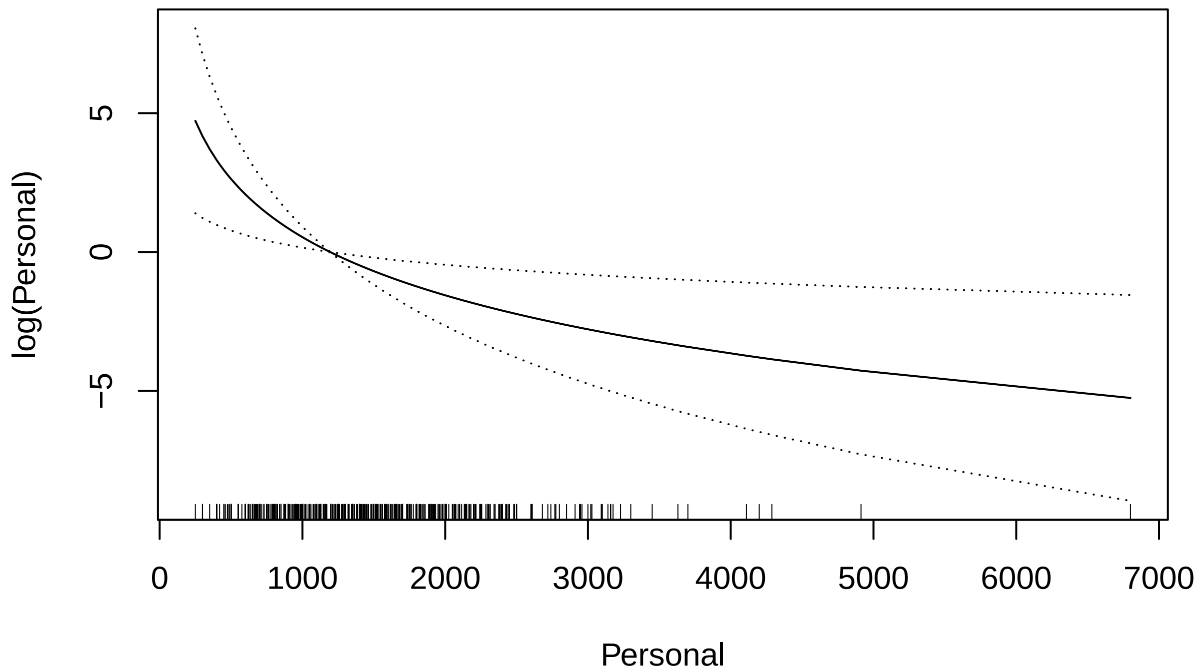

partial for Outstate
15
10
5
0
−5
−10
5000
10000
15000
20000
Outstate

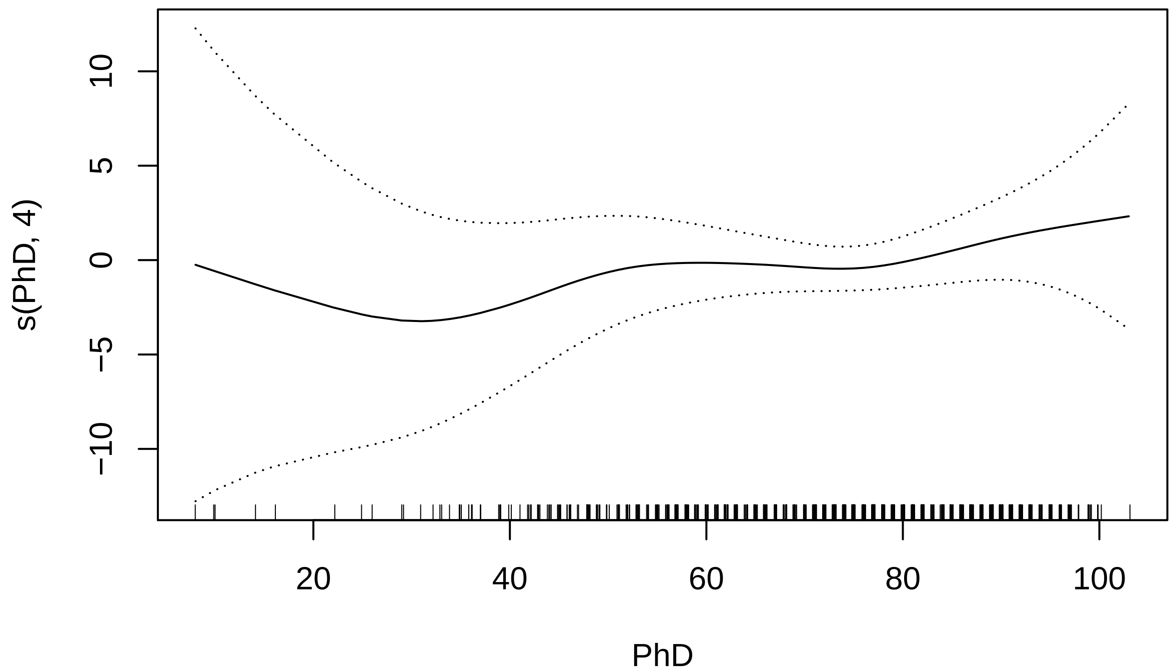


partial for Room.Board
10
5
0
−5
2000
3000
4000
5000
6000
7000
8000
Room.Board

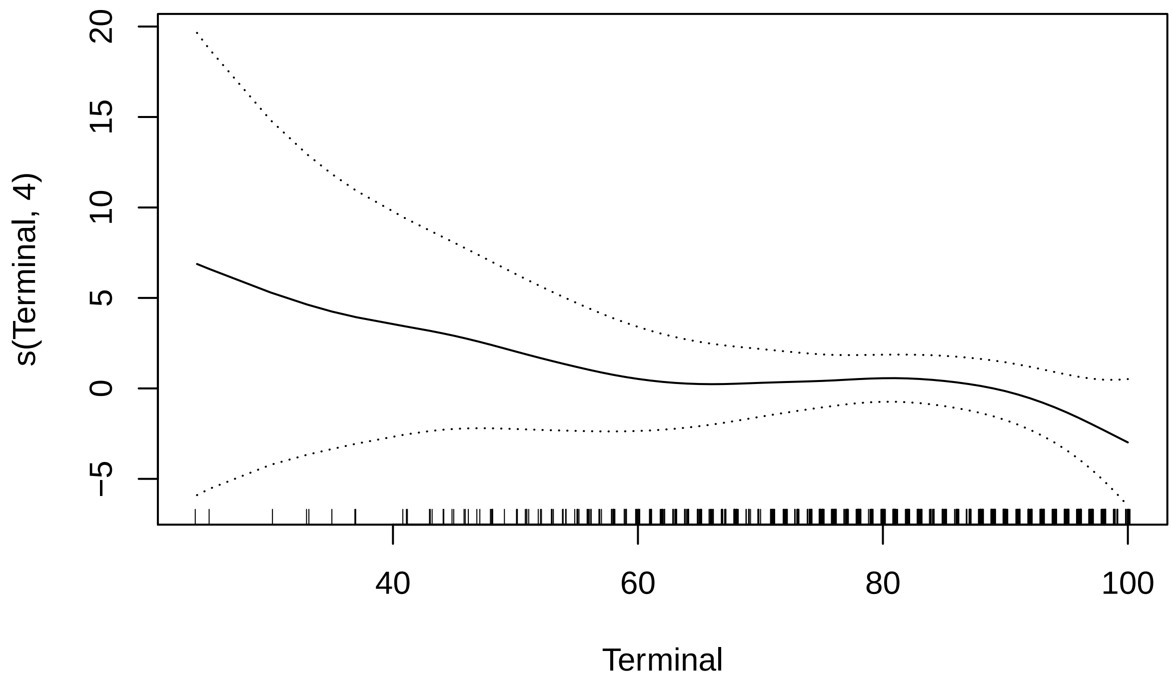


log(Books)
10
5
0
−5
−10
500
1000
1500
2000
Books

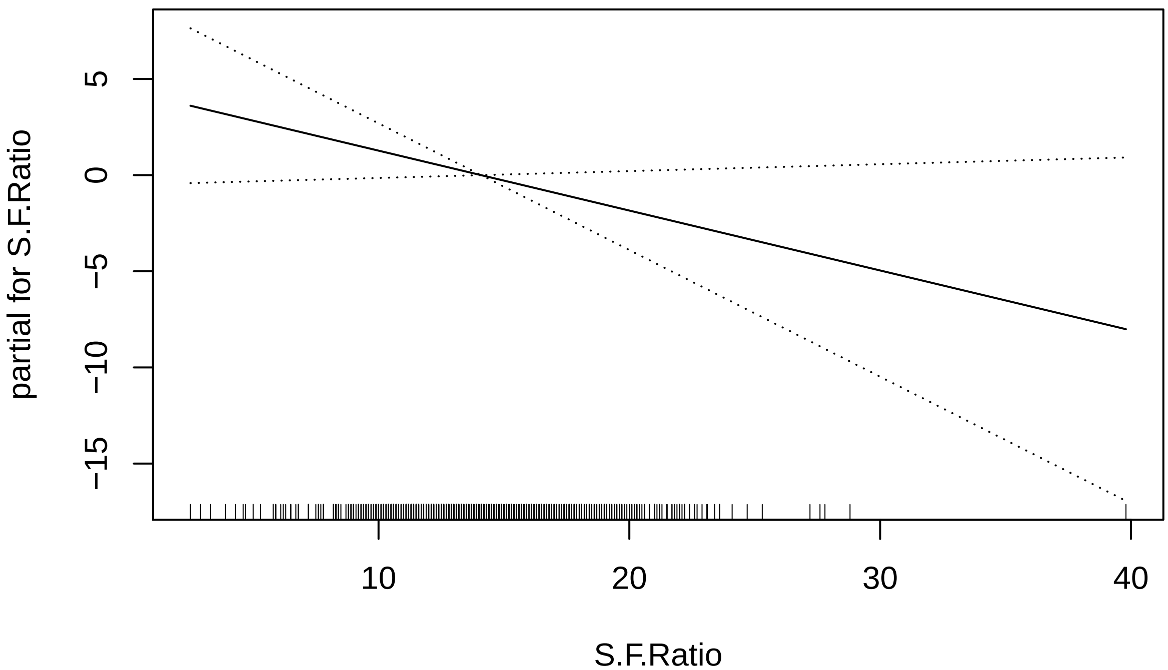

log(Personal)
5
0
−5
0
1000
2000
3000
4000
5000
6000
7000
Personal



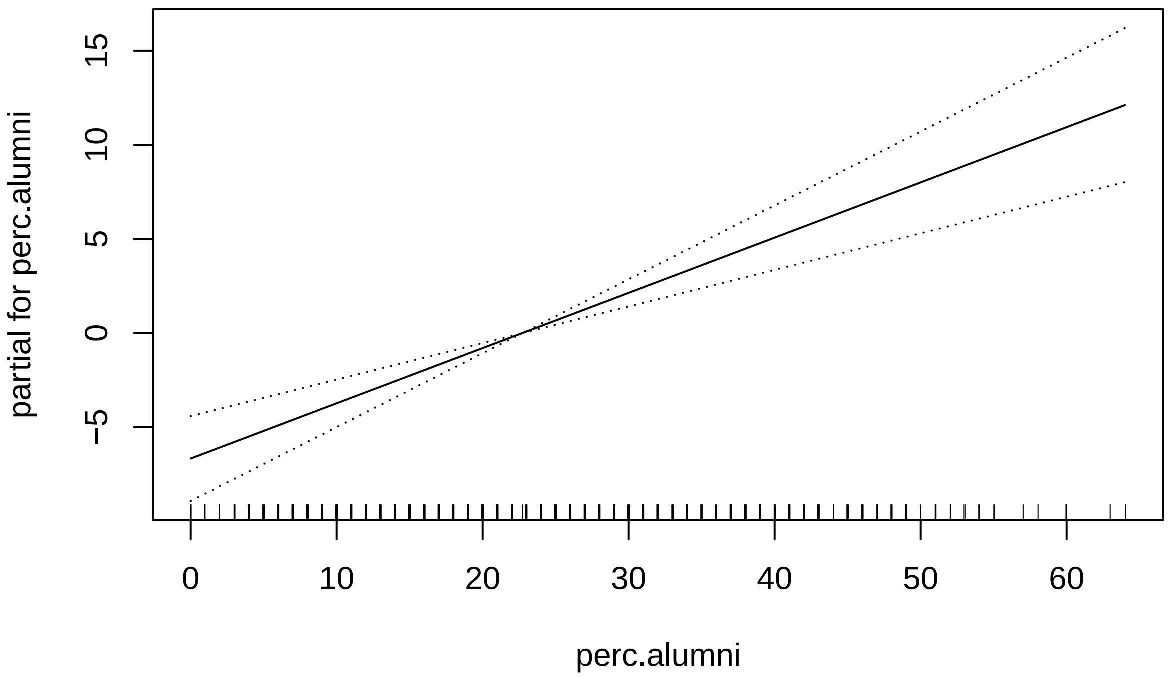
s(PhD, 4)
10
5
0
−5
−10
20
40
60
80
100
PhD

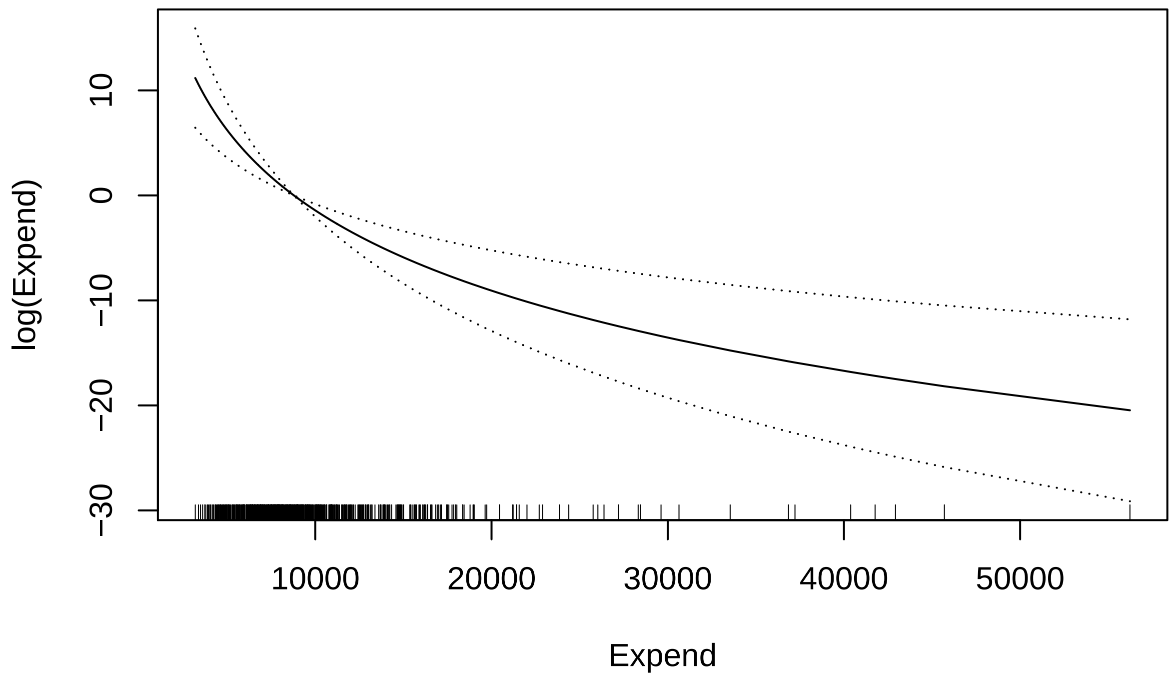


s(Terminal, 4)
20
15
10
5
0
−5
40
60
80
100
Terminal

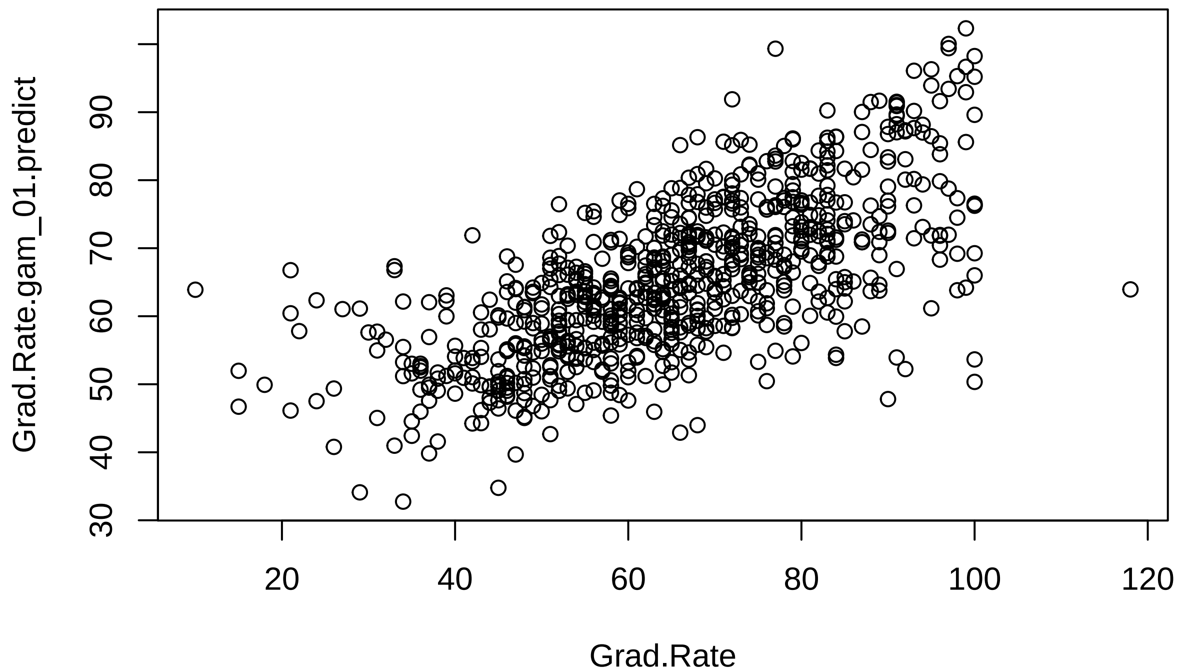


partial for S.F.Ratio
5
0
−5
−10
−15
10
20
30
40
S.F.Ratio


partial for perc.alumni
perc.alumni
−5
0
5
10
15
0
10
20
30
40
50
60



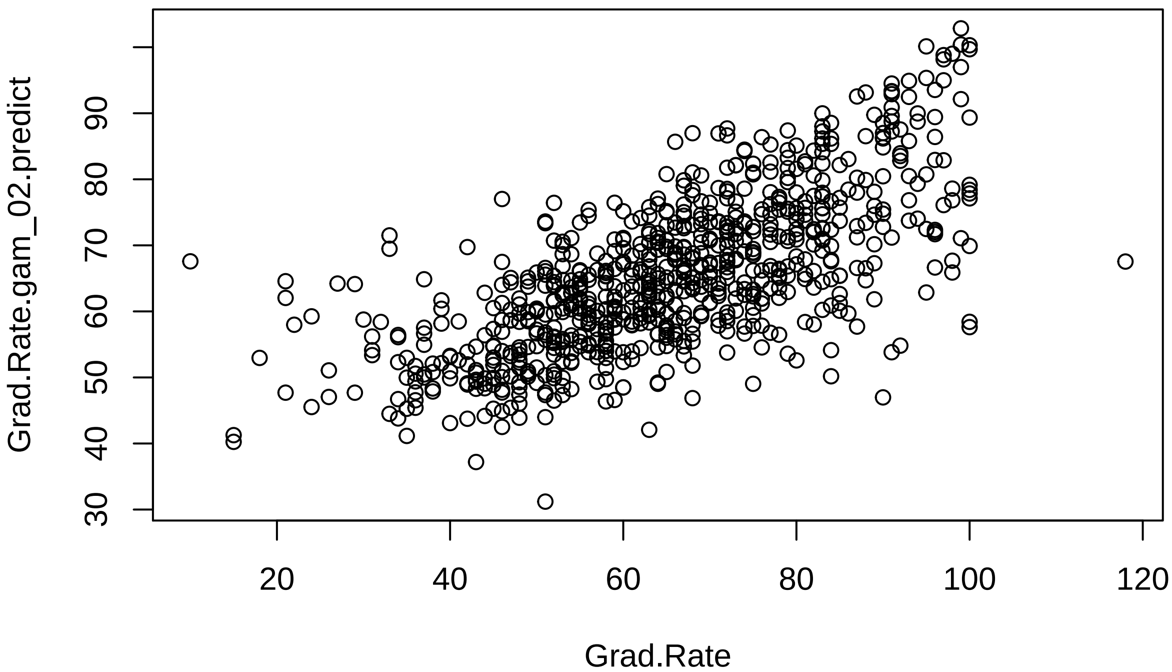

```r
Grad.Rate.gam_01.predict = predict(Grad.Rate.gam_01, data=College)
Grad.Rate.gam_01.resid= Grad.Rate.gam_01.predict - Grad.Rate
plot(Grad.Rate,Grad.Rate.gam_01.predict)
```




```
(Grad.Rate.gam_01.ECM = mean(Grad.Rate.gam_01.resid**2))
```

```
## [1] 158.6807
```

```
Grad.Rate.gam_02.predict = predict(Grad.Rate.gam_02, data=College)
Grad.Rate.gam_02.resid= Grad.Rate.gam_02.predict - Grad.Rate
plot(Grad.Rate,Grad.Rate.gam_02.predict)
```




```
(Grad.Rate.gam_02.ECM = mean(Grad.Rate.gam_02.resid**2))
```

```
## [1] 154.3325
```