Data Science 2, HW2

Ekta Chaudhary

20/03/2020

```r
library(tidyverse)
library(caret)
library(ModelMetrics)
library(glmnet)
library(gam)
library(mgcv)
library(splines)
library(pdp)
library(earth)
```

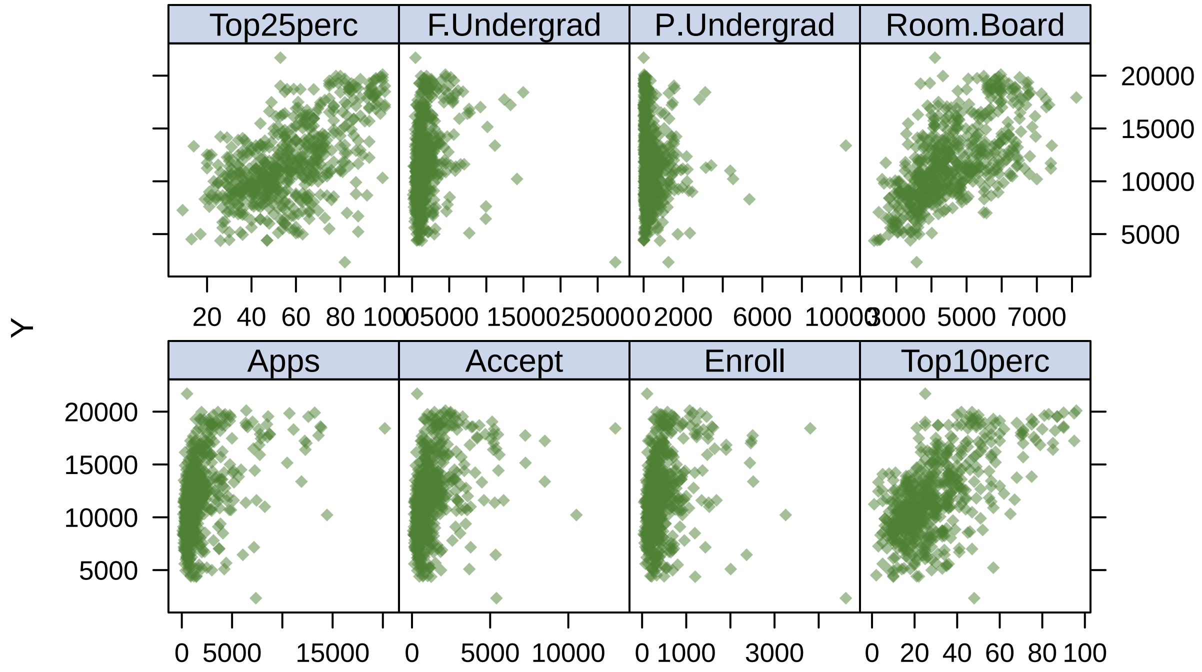
Reading the Datasets

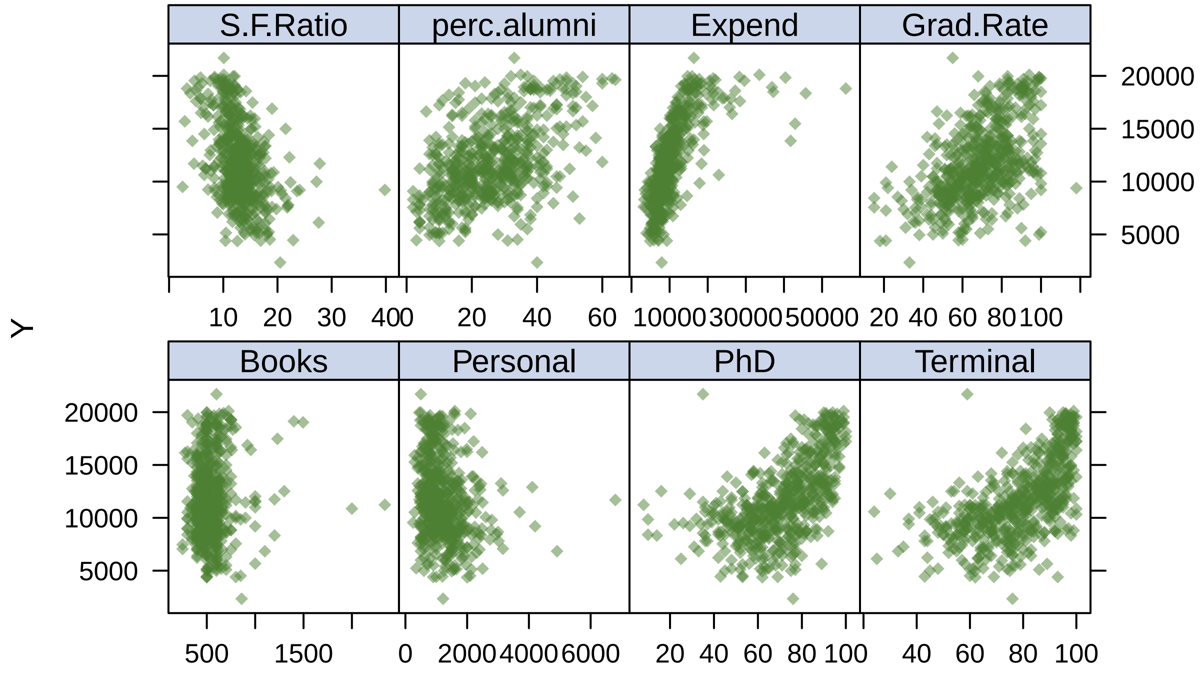
```r
data =
  read_csv('./data/College.csv') %>%
select(-College)
data_1 =
  data[-125,]
data_2 =
  data[125,]
```

```r
x <- model.matrix(Outstate~.,data_1)[,-1]
y <- data_1$Outstate
```

(a) Create scatter plots of response vs. predictors.

```r
theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.3, .5, .2, .5)
theme1$plot.symbol$pch <- 18
theme1$plot.line$col <- rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)
trellis.par.set(theme1)
featurePlot(x, y, plot = "scatter", labels = c("","Y"),
            type = c("p"), layout = c(4, 2))
```


Top25perc
F.Undergrad
P.Undergrad
Room.Board
20000
15000
10000
5000
20 40 60 80 100
0 5000 15000 25000
0 2000 6000 10000
3000 5000 7000
Y
Apps
Accept
Enroll
Top10perc
20000
15000
10000
5000
0 5000 15000
0 5000 10000
0 1000 3000
0 20 40 60 80 100

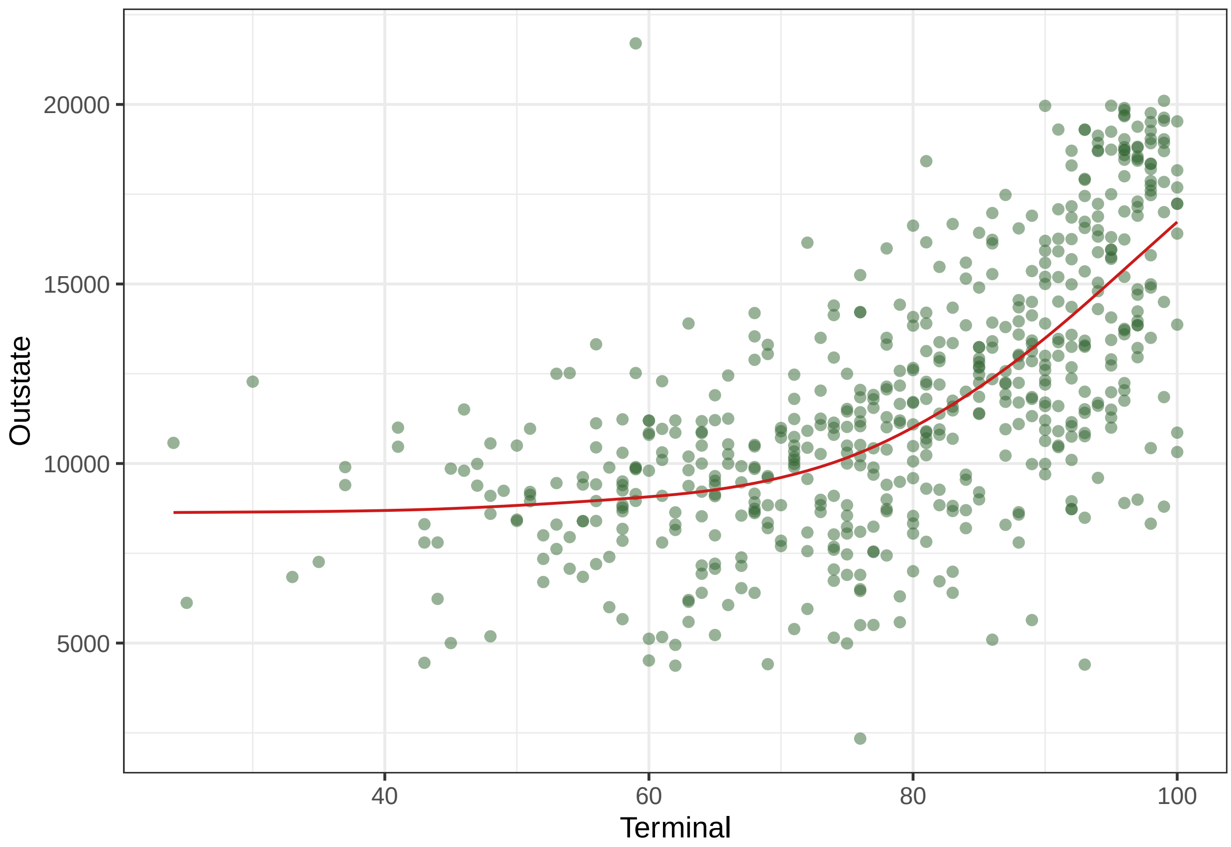

S.F.Ratio
perc.alumni
Expend
Grad.Rate
20000
15000
10000
5000
10 20 30 40
0 20 40 60
10000 30000 50000
20 40 60 80 100
Y
Books
Personal
PhD
Terminal
20000
15000
10000
5000
500 1500
0 2000 4000 6000
20 40 60 80 100
40 60 80 100

b) Fit a smoothing spline model using Terminal as the only predictor of Outstate for a range of degrees of freedom, as well as the degree of freedom obtained by generalized cross- validation, and plot the resulting fits. Describe the results obtained.

```
Terminallims <- range(data_1$Terminal)
Terminal.grid <- seq(from = Terminallims[1],to = Terminallims[2])

fit.ss <- smooth.spline(data_1$Terminal, data_1$Outstate)
fit.ss$df
```

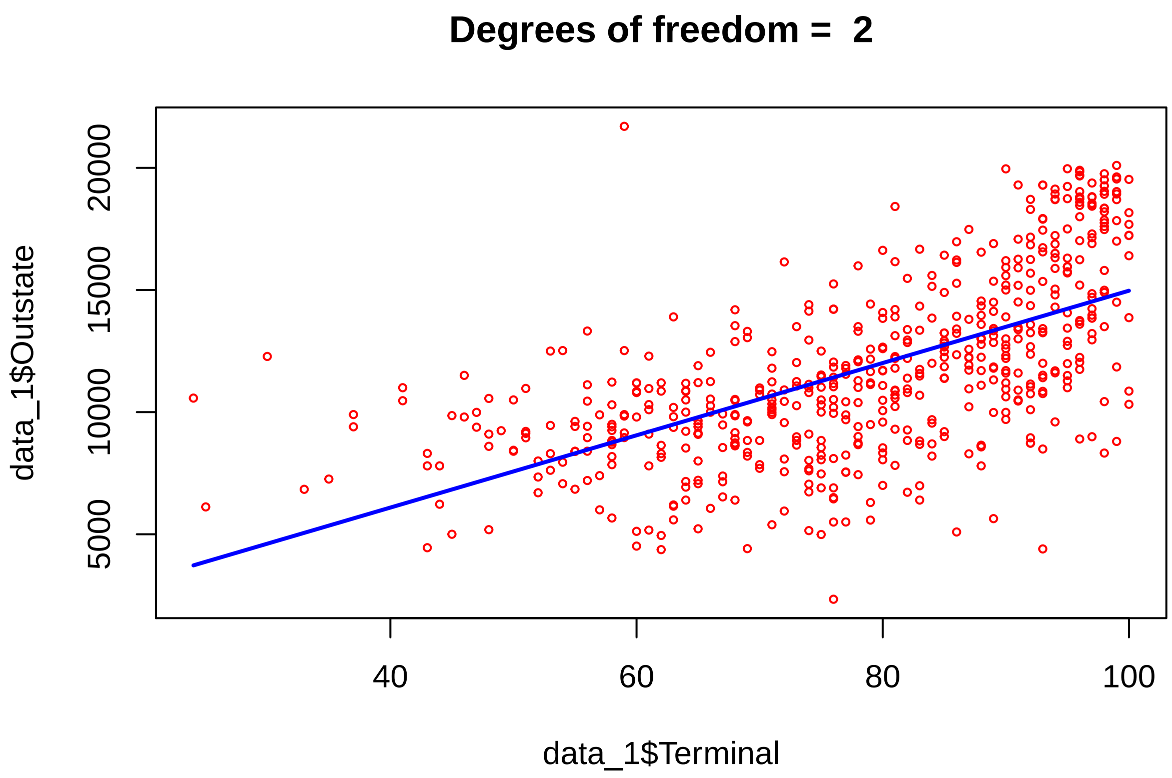
```
## [1] 4.468629
```

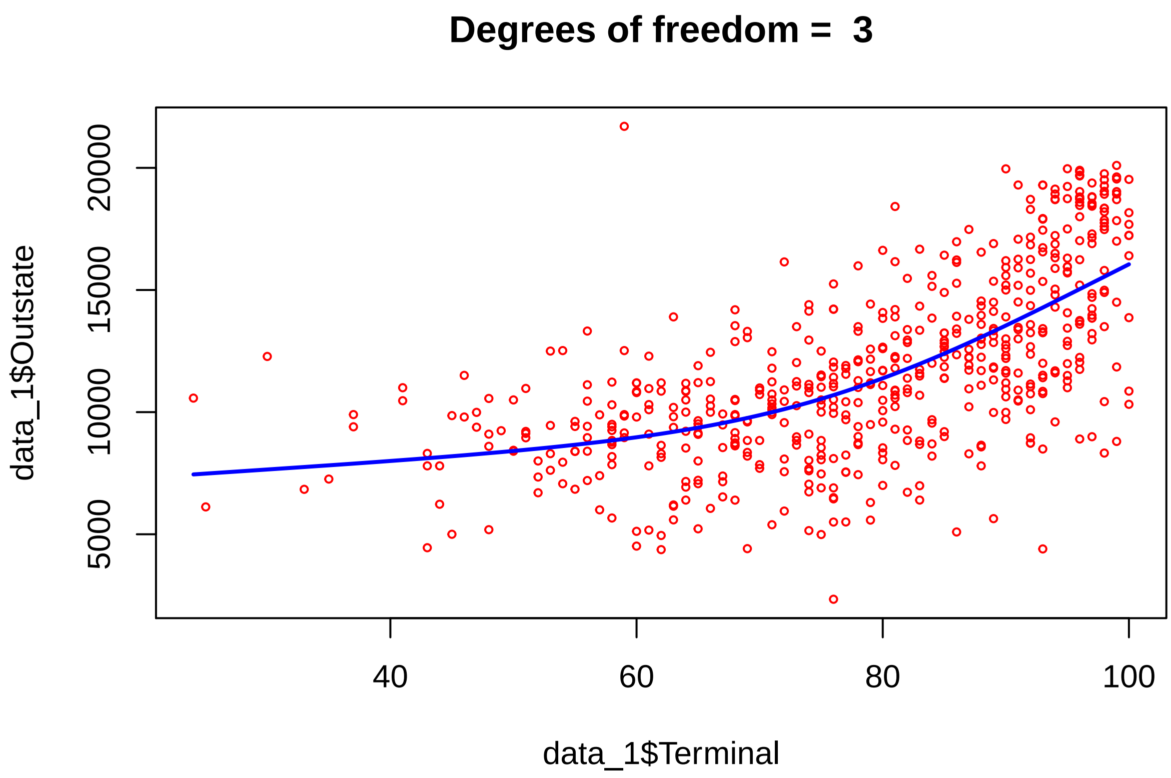
```
pred.ss <- predict(fit.ss,
                   x = Terminal.grid)

pred.ss.df <- data.frame(pred = pred.ss$y,
                         Terminal = Terminal.grid)

p <- ggplot(data = data_1, aes(x = Terminal, y = Outstate)) +
  geom_point(color = rgb(.2, .4, .2, .5))
p + geom_line(aes(x = Terminal, y = pred), data = pred.ss.df,
              color = rgb(.8, .1, .1, 1)) + theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```

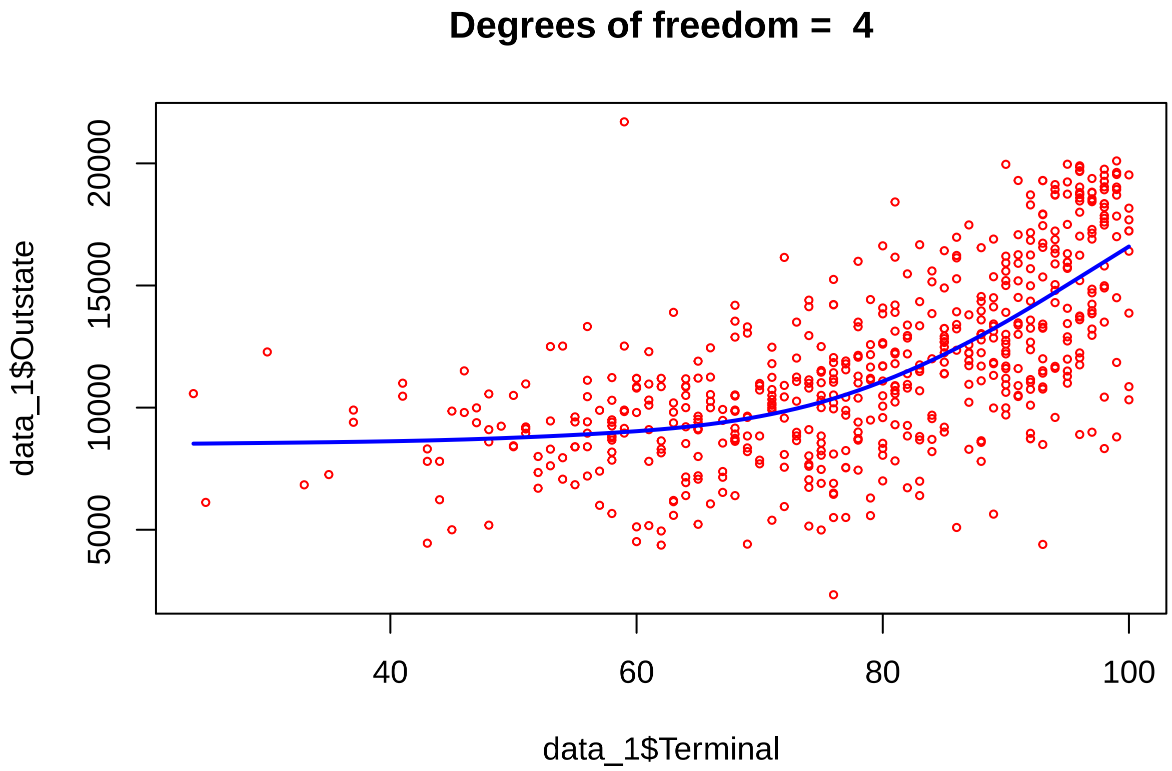


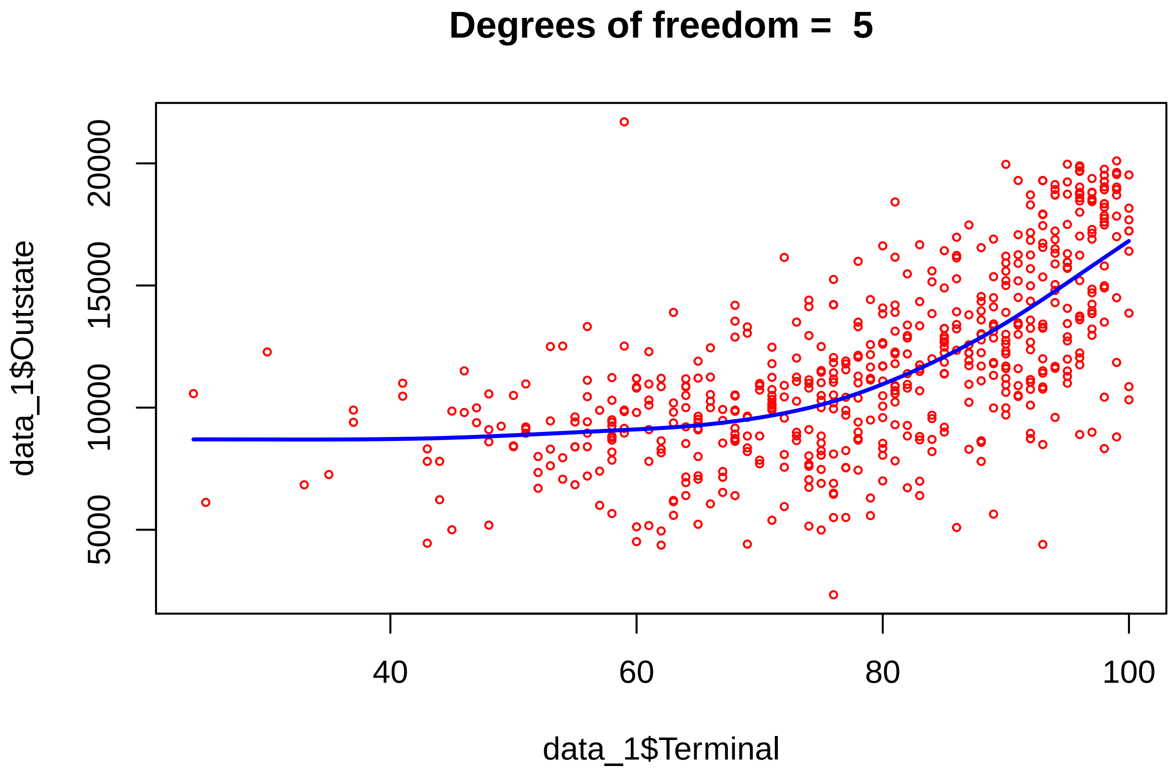
The function `smooth.spline()`is used to fit smoothing spline models. Generalized cross-validation is used to select the degree of freedom.The degree of freedom obtained by generalized cross-validation is 4.468629.

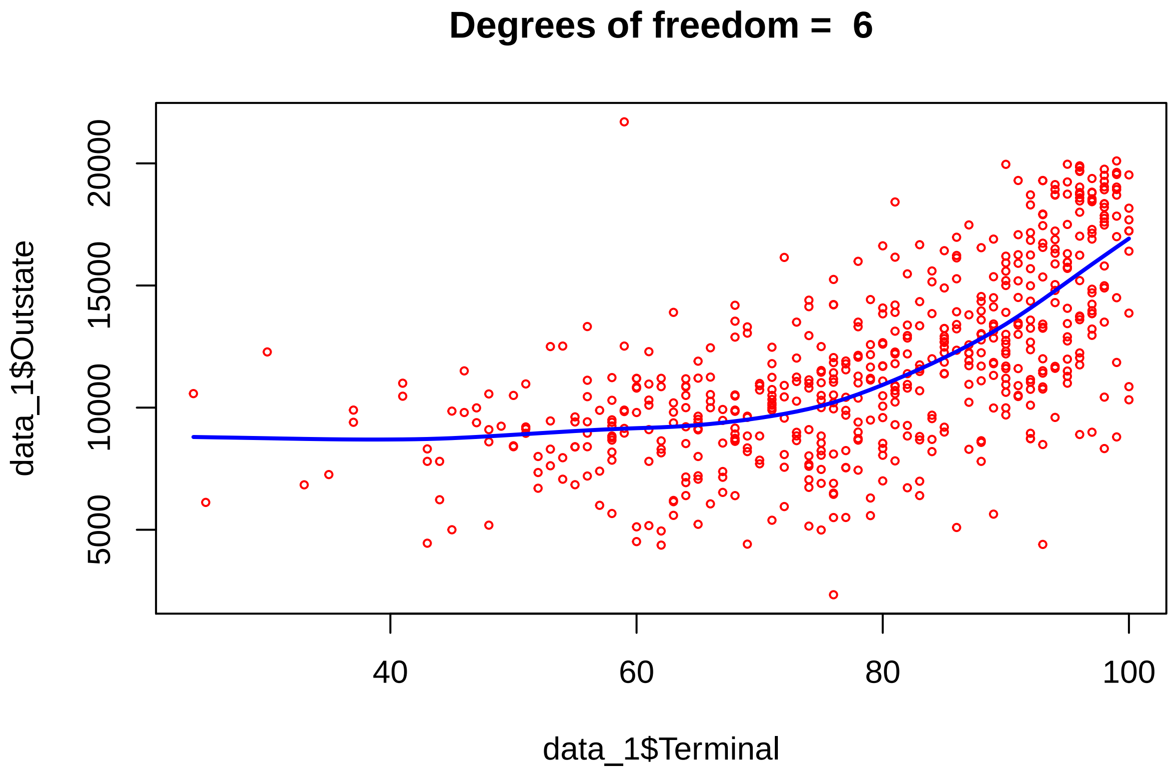
```
for (i in 2:15) {
  fit.ss = smooth.spline(data_1$Terminal, data_1$Outstate, df = i)

  pred.ss <- predict(fit.ss, x = Terminal.grid)

  plot(data_1$Terminal, data_1$Outstate, cex = .5, col = "red")
  title(paste("Degrees of freedom = ", round(fit.ss$df)),  outer = F)
  lines(Terminal.grid, pred.ss$y, lwd = 2, col = "blue")
}
```

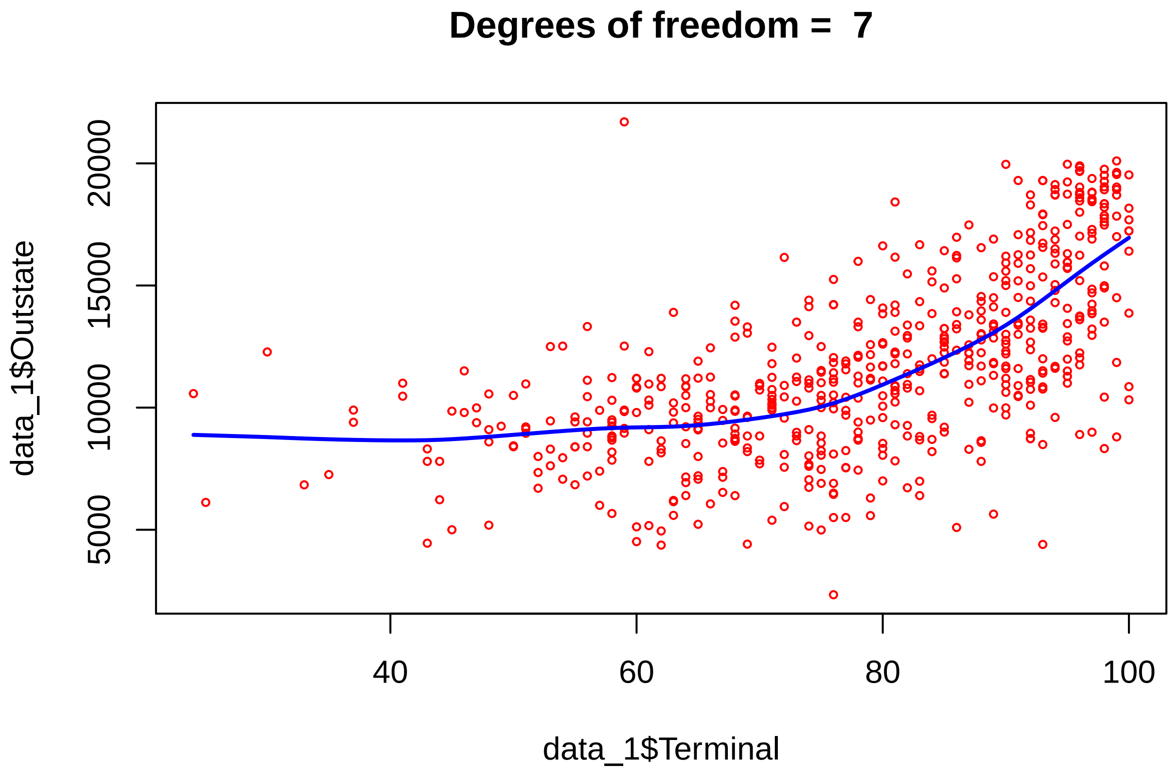

Degrees of freedom = 2
data_1$Outstate
5000
10000
15000
20000
40
60
80
100
data_1$Terminal

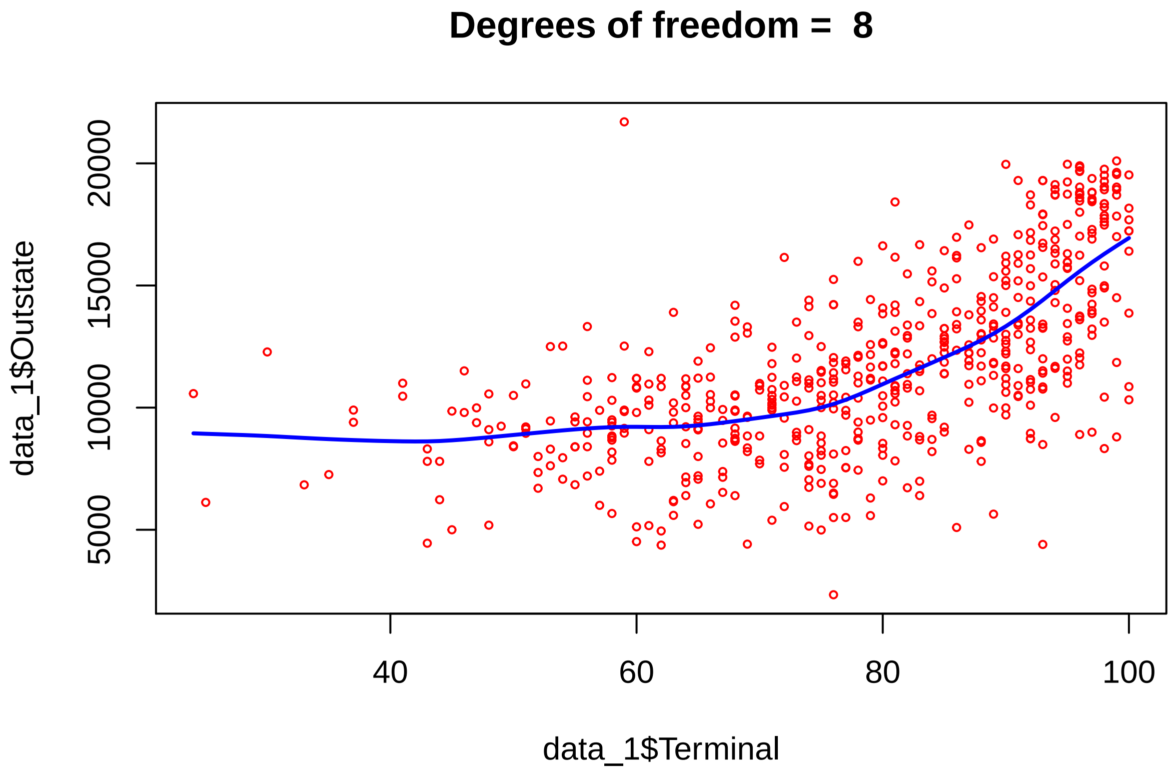

Degrees of freedom = 3
data_1$Outstate
5000
10000
15000
20000
40
60
80
100
data_1$Terminal

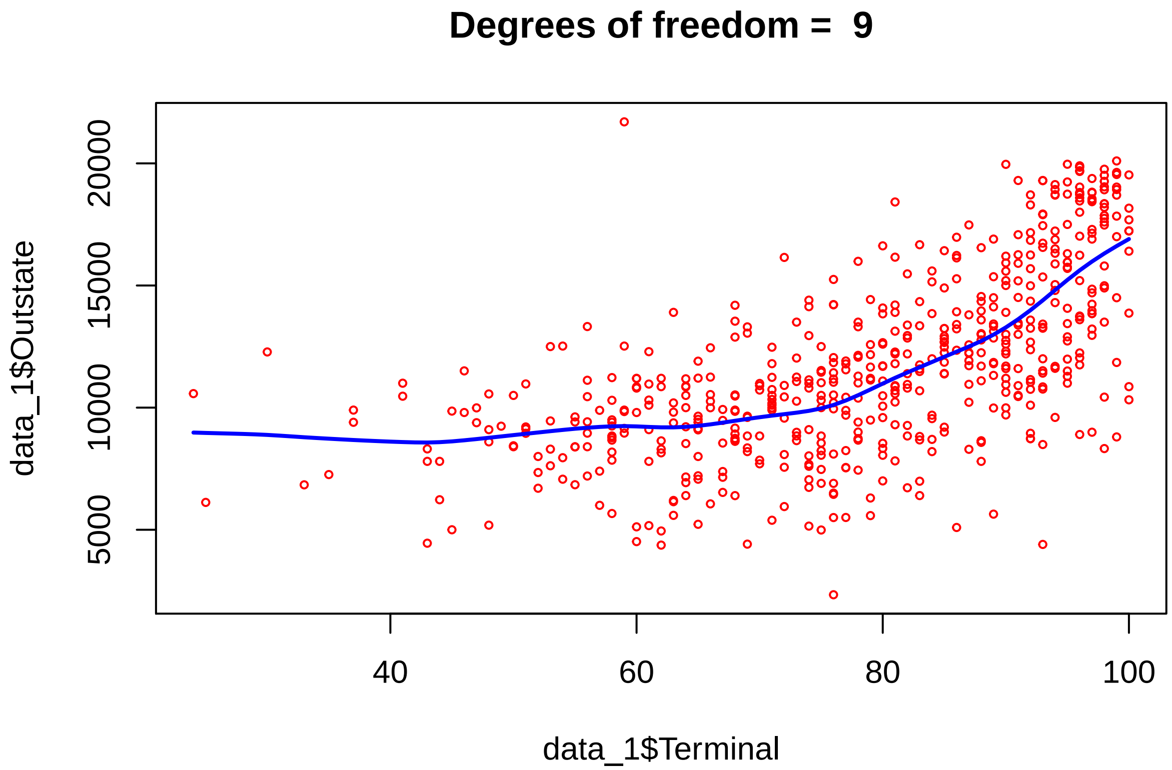

Degrees of freedom = 4
data_1$Outstate
5000
10000
15000
20000
40
60
80
100
data_1$Terminal

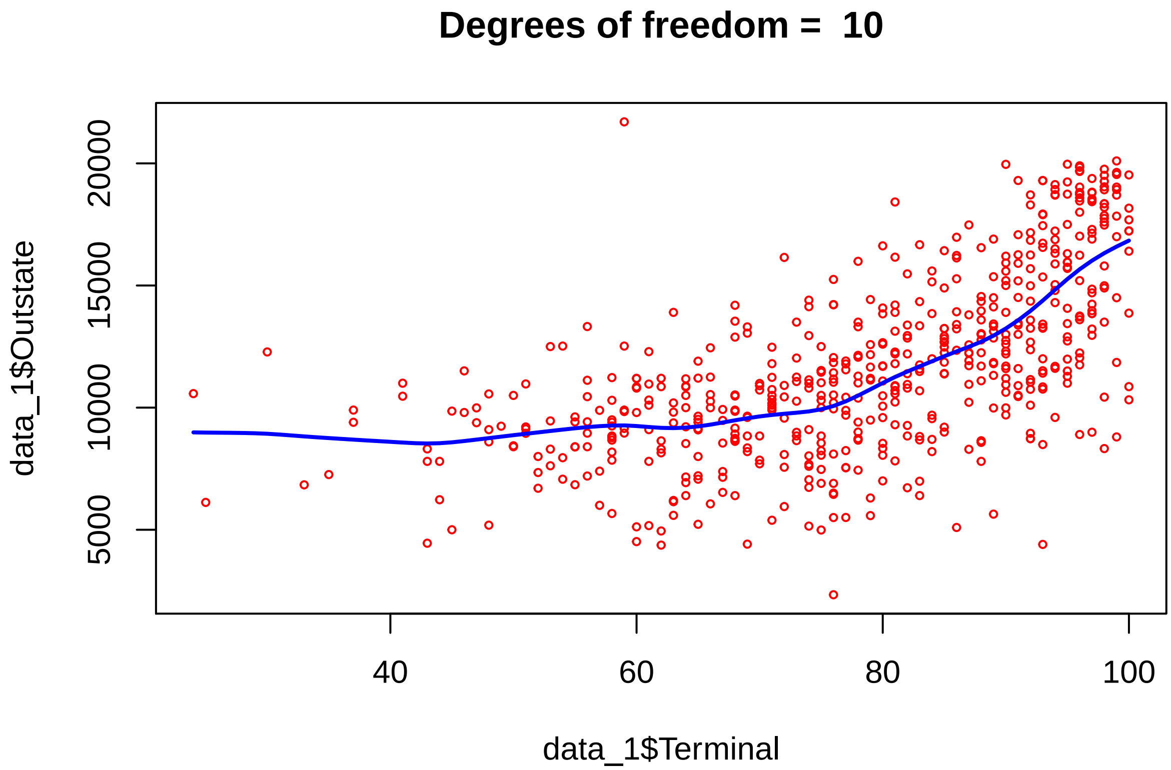

Degrees of freedom = 5
data_1$Outstate
20000
15000
10000
5000
40
60
80
100
data_1$Terminal

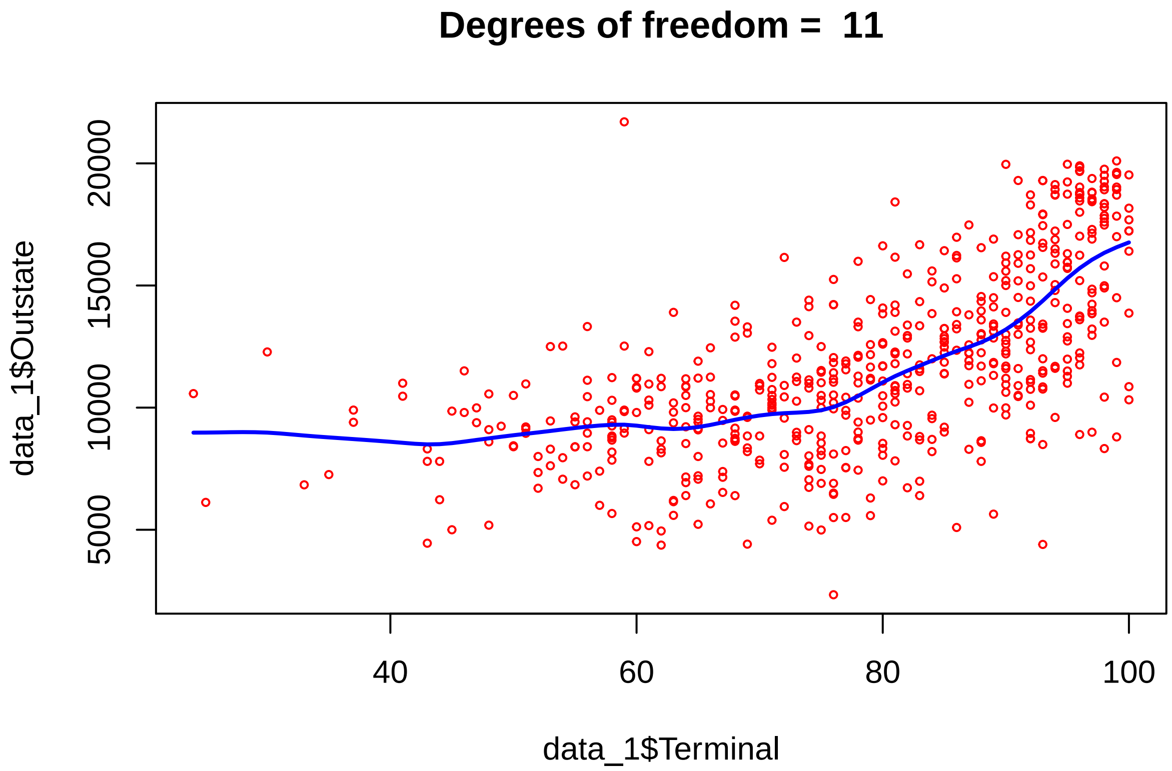

Degrees of freedom = 6
data_1$Outstate
5000
10000
15000
20000
40
60
80
100
data_1$Terminal

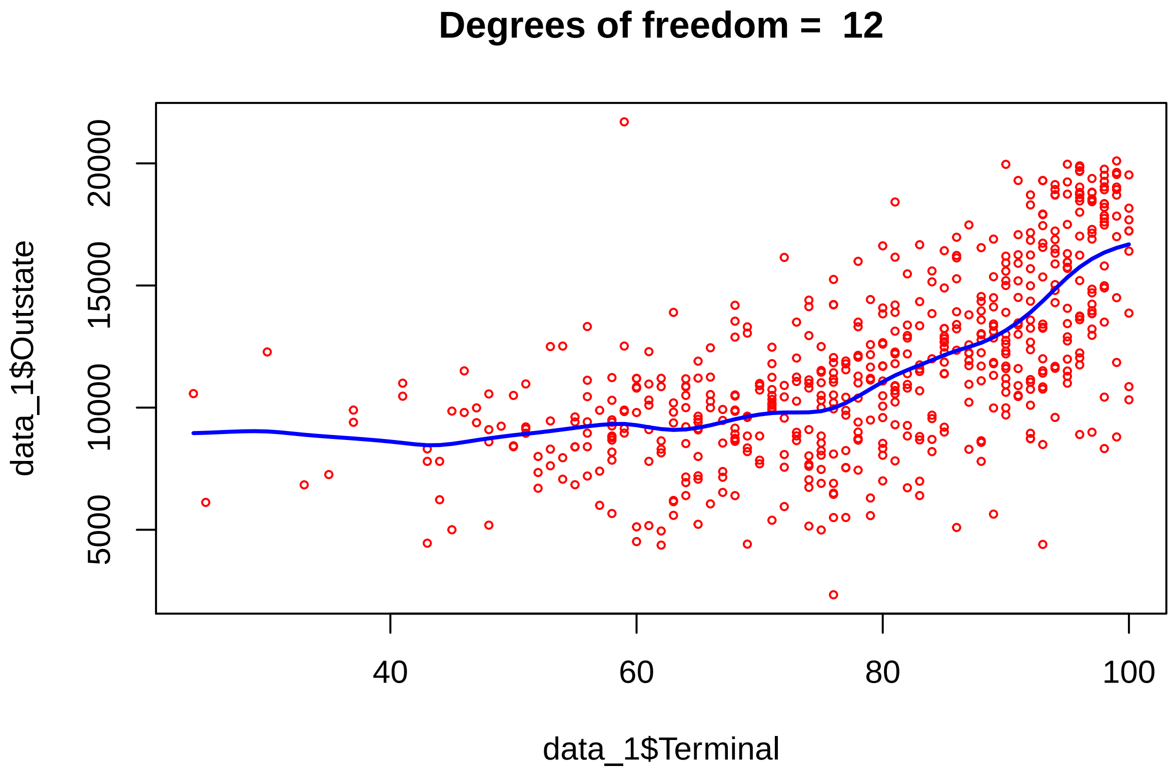

Degrees of freedom = 7
data_1$Outstate
5000
10000
15000
20000
data_1$Terminal
40
60
80
100

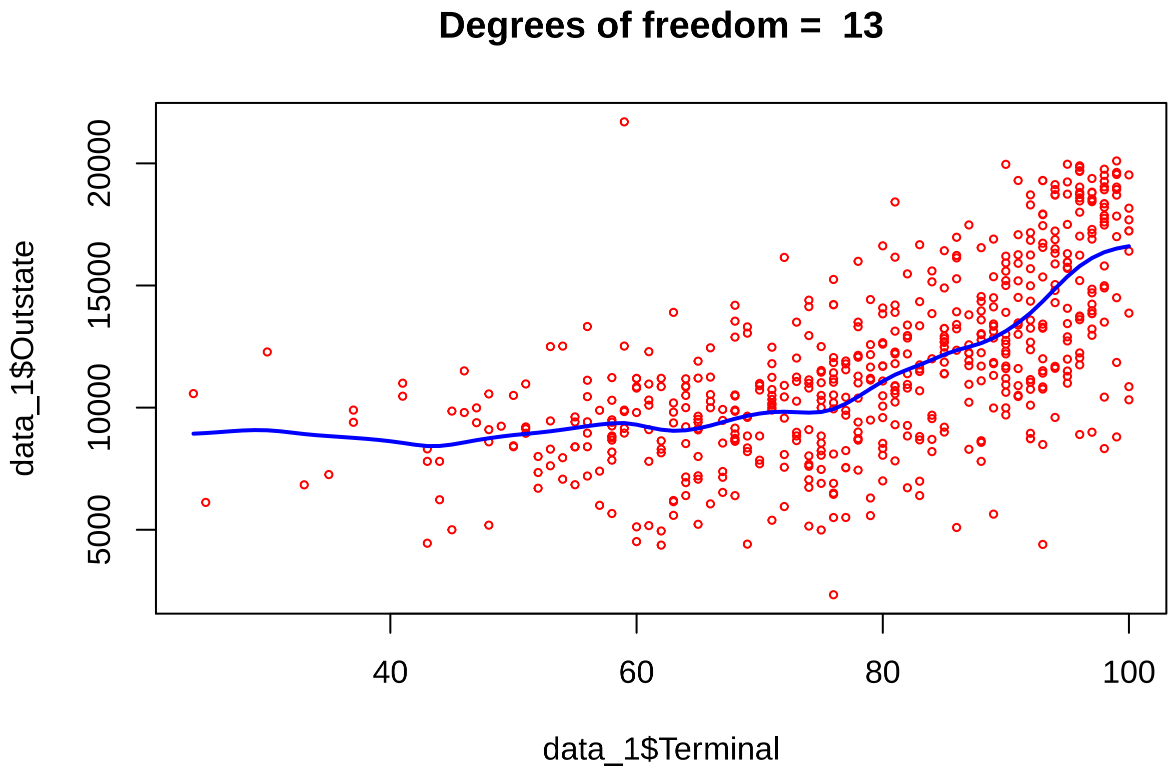

Degrees of freedom = 8
data_1$Outstate
5000
10000
15000
20000
40
60
80
100
data_1$Terminal

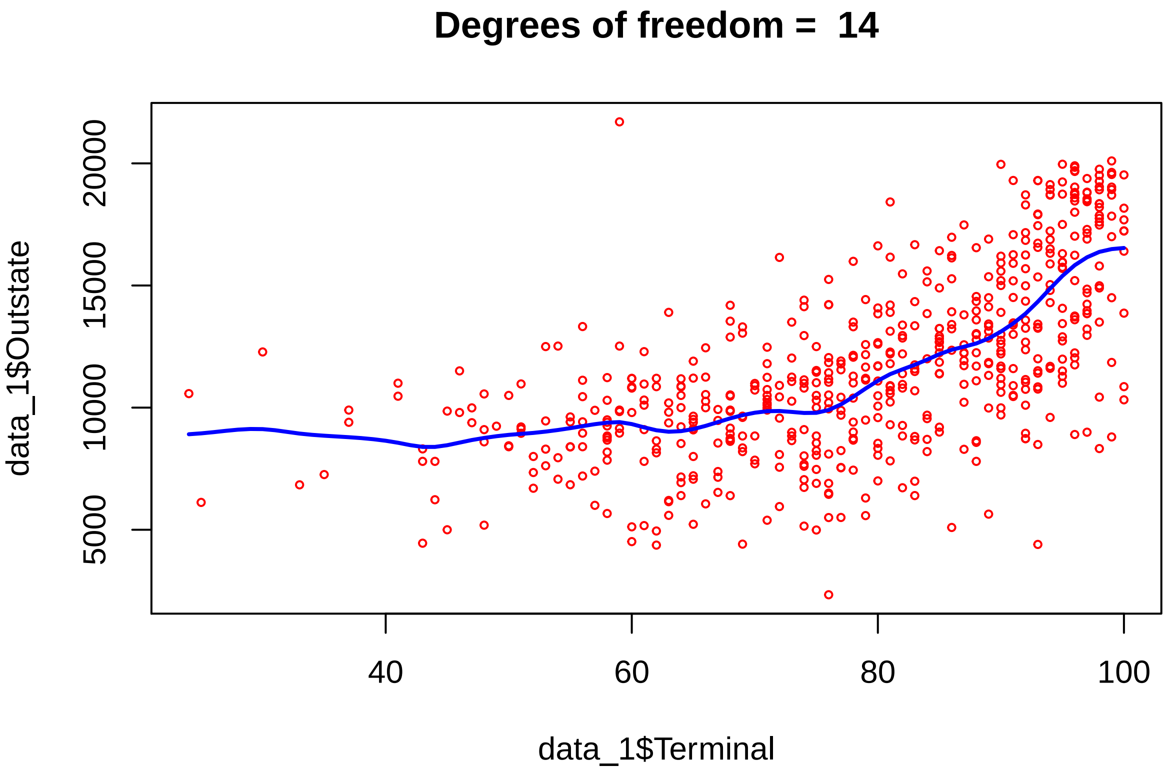

Degrees of freedom = 9
data_1$Outstate
5000
10000
15000
20000
data_1$Terminal
40
60
80
100

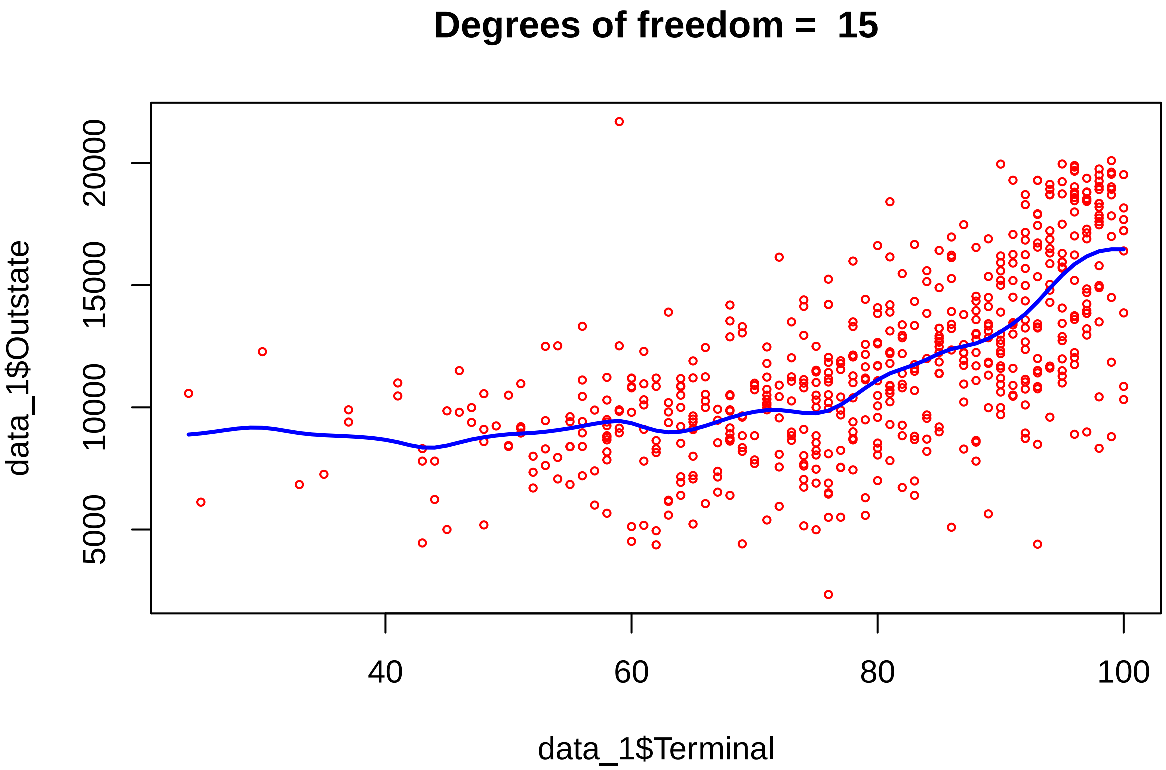

Degrees of freedom = 10
data_1$Outstate
5000
10000
15000
20000
40
60
80
100
data_1$Terminal


Degrees of freedom = 11
data_1$Outstate
5000
10000
15000
20000
40
60
80
100
data_1$Terminal


Degrees of freedom = 12
data_1$Outstate
5000
10000
15000
20000
40
60
80
100
data_1$Terminal


Degrees of freedom = 13
data_1$Outstate
5000
10000
15000
20000
40
60
80
100
data_1$Terminal


Degrees of freedom = 14
data_1$Outstate
5000
10000
15000
20000
40
60
80
100
data_1$Terminal



# I have picked a range of degrees of freedom from 2 to 15. As it can be seen from the plots, when the degree of freedom is 2, the model is linear and when the df increases the model gets wiggly.

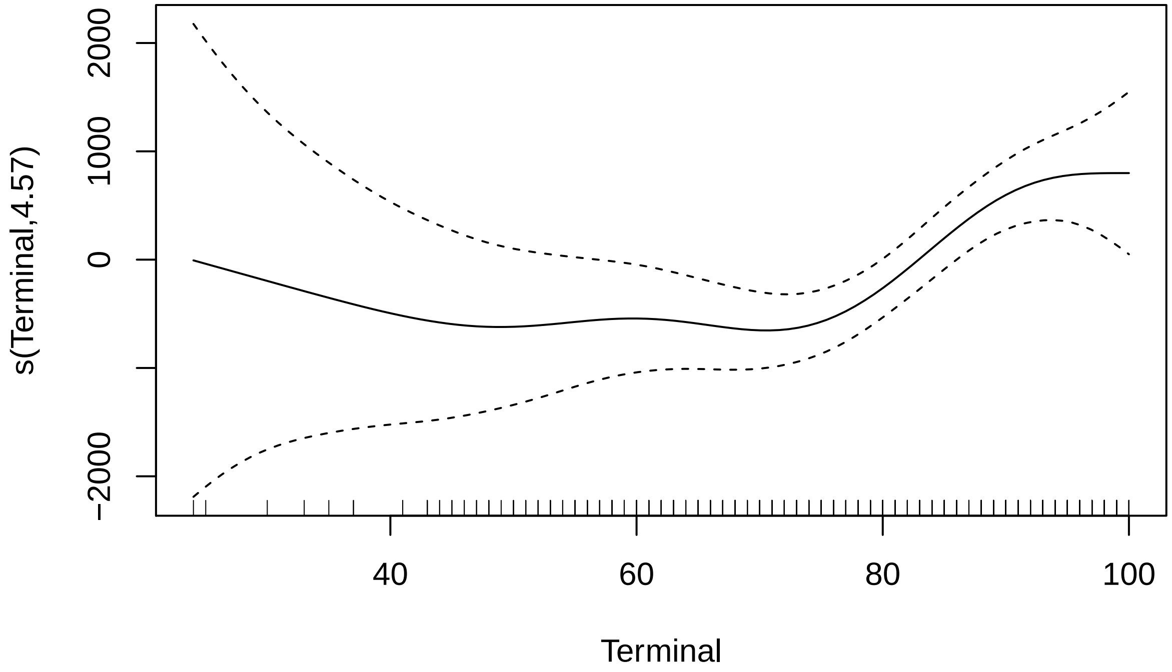
c) Fit a generalized additive model (GAM) using all the predictors. Plot the results and explain your findings.

```
gam.m1 = gam(
  Outstate~ Apps + Accept + Enroll + Top10perc + Top25perc + F.Undergrad + P.Undergrad + Room.Board + Be
  data = data_1)
gam.m2 = gam(
  Outstate~ Apps + Accept + Enroll + Top10perc + Top25perc + F.Undergrad + P.Undergrad + Room.Board + Be
  data = data_1)
gam.m3 = gam(
  Outstate~ Apps + Accept + Enroll + Top10perc + Top25perc + F.Undergrad + P.Undergrad + te(Room.Board)

anova(gam.m1, gam.m2, gam.m3, test = "F")
```

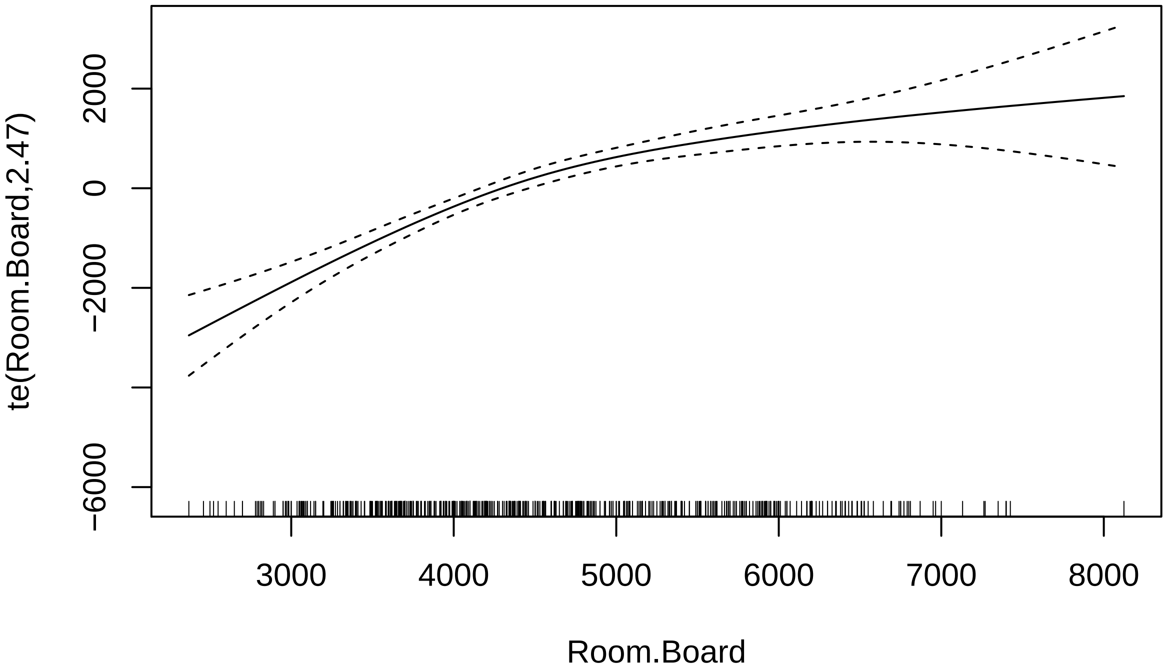
```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ Apps + Accept + Enroll + Top10perc + Top25perc + F.Undergrad +
##     P.Undergrad + Room.Board + Books + Personal + PhD + Terminal +
##     S.F.Ratio + perc.alumni + Expend + Grad.Rate
## Model 2: Outstate ~ Apps + Accept + Enroll + Top10perc + Top25perc + F.Undergrad +
##     P.Undergrad + Room.Board + Books + Personal + PhD + s(Terminal) +
##     S.F.Ratio + perc.alumni + Expend + Grad.Rate
## Model 3: Outstate ~ Apps + Accept + Enroll + Top10perc + Top25perc + F.Undergrad +
##     P.Undergrad + te(Room.Board) + te(Personal) + Books + PhD +
##     s(Terminal) + S.F.Ratio + perc.alumni + Expend + Grad.Rate
##   Resid. Df Resid. Dev     Df Deviance      F    Pr(>F)
## 1    547.00 2092185295
## 2    542.37 2026858216 4.6295 65327078 3.9364  0.002202 **
## 3    537.48 1933201900 4.8882 93656316 5.3448 9.659e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

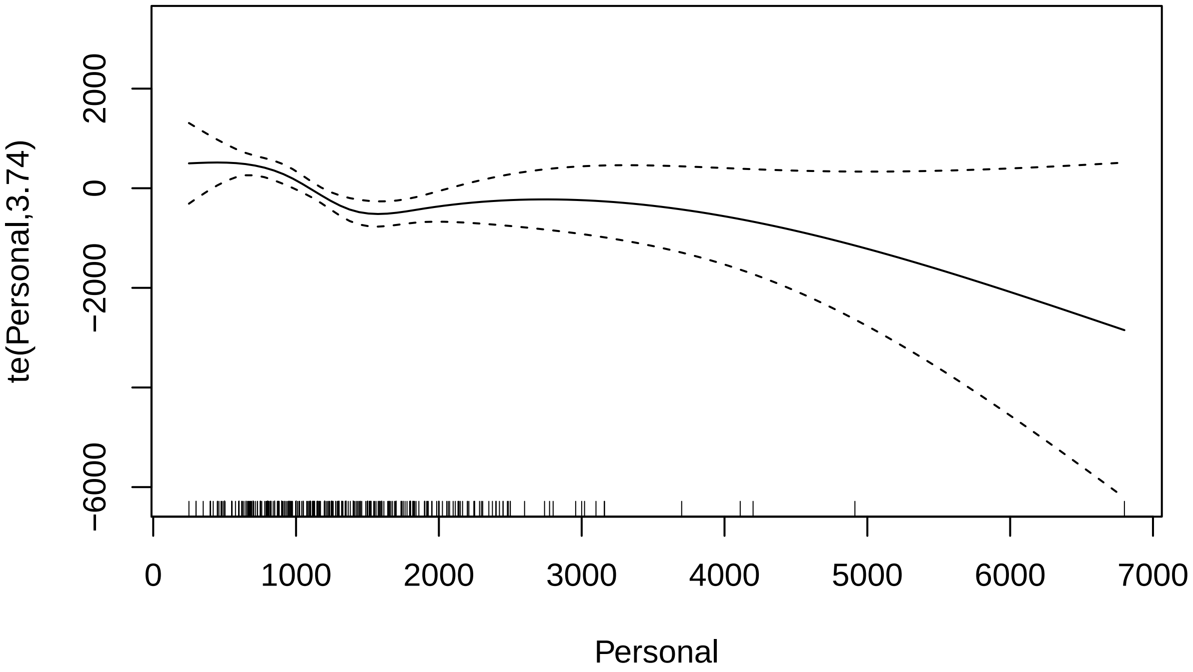
Looking at the p-values from the ANOVA test, Model 3 appears to be the best fitting model.

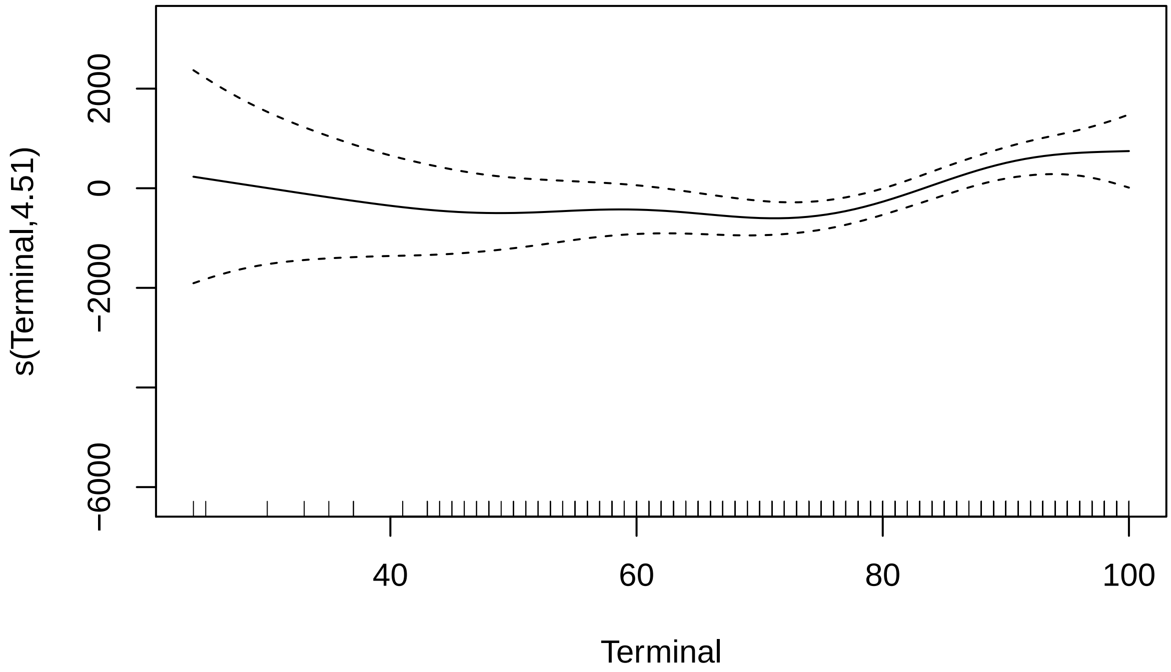
```r
plot(gam.m2)
```



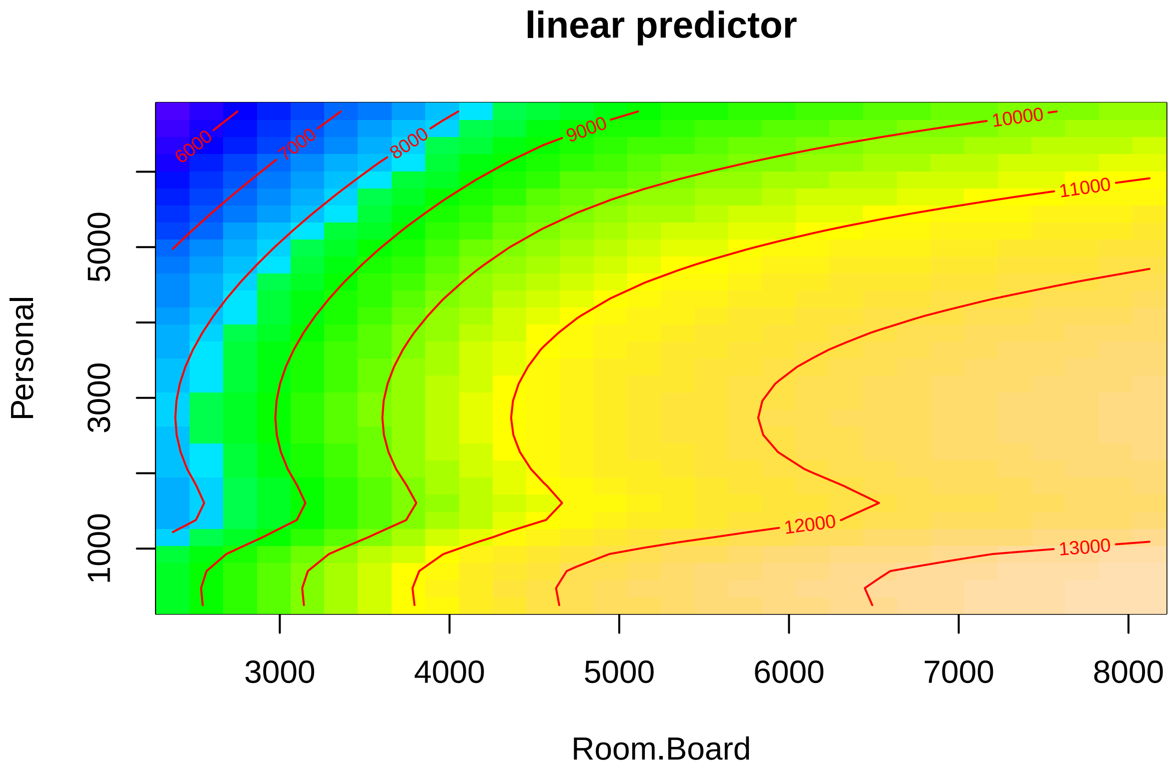
```r
plot(gam.m3)
```


te(Room.Board,2.47)
2000
0
−2000
−6000
3000
4000
5000
6000
7000
8000
Room.Board


te(Personal,3.74)
2000
0
−2000
−6000
0
1000
2000
3000
4000
5000
6000
7000
Personal

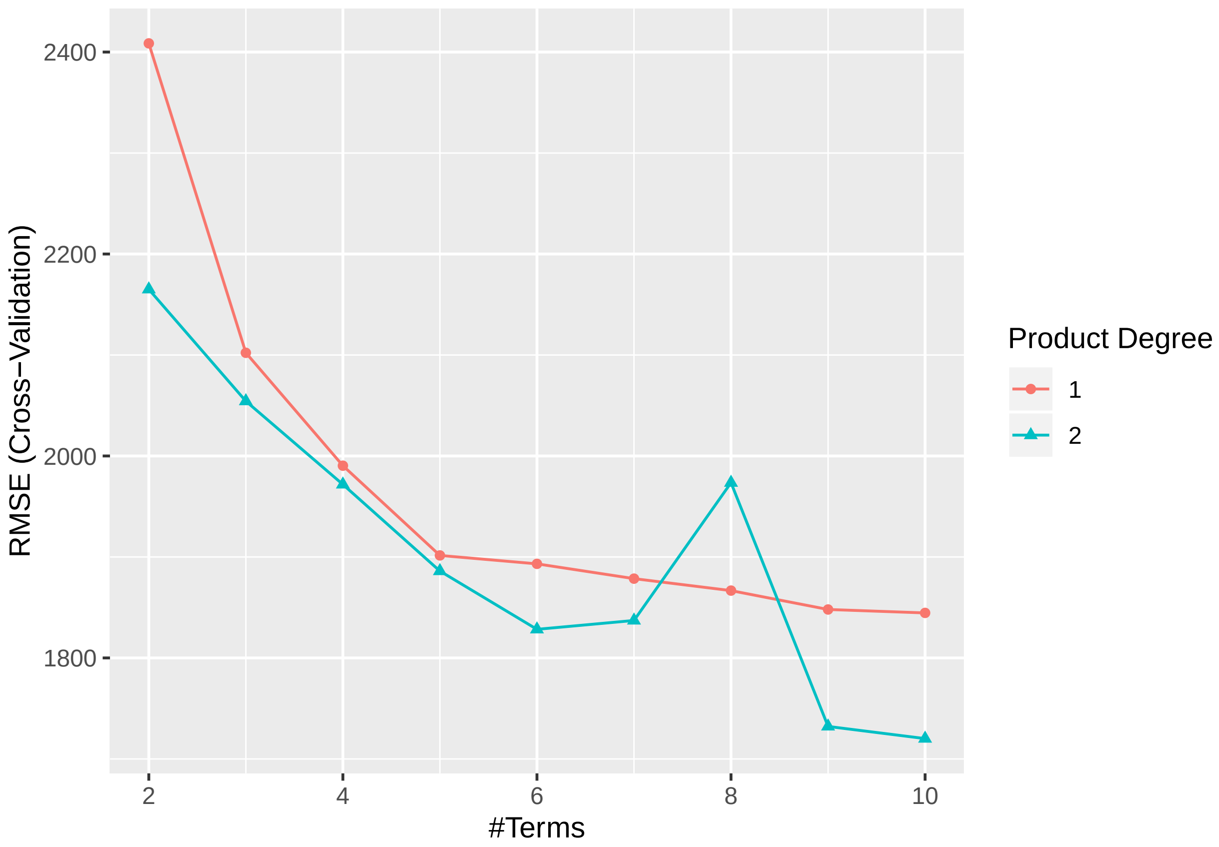


```
vis.gam(gam.m3, view = c("Room.Board","Personal"),plot.type = "contour", color = "topo")
```



(d) Fit a multivariate adaptive regression spline (MARS) model using all the predictors. Report the final model. Present the partial dependence plot of an arbitrary predictor in your final model.

```
ctrl1 <- trainControl(method = "cv", number = 10)
mars_grid <- expand.grid(degree = 1:2,
                         nprune = 2:10)

set.seed(2)
mars.fit <- train(x, y,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)

ggplot(mars.fit)
```



```
mars.fit$bestTune
```

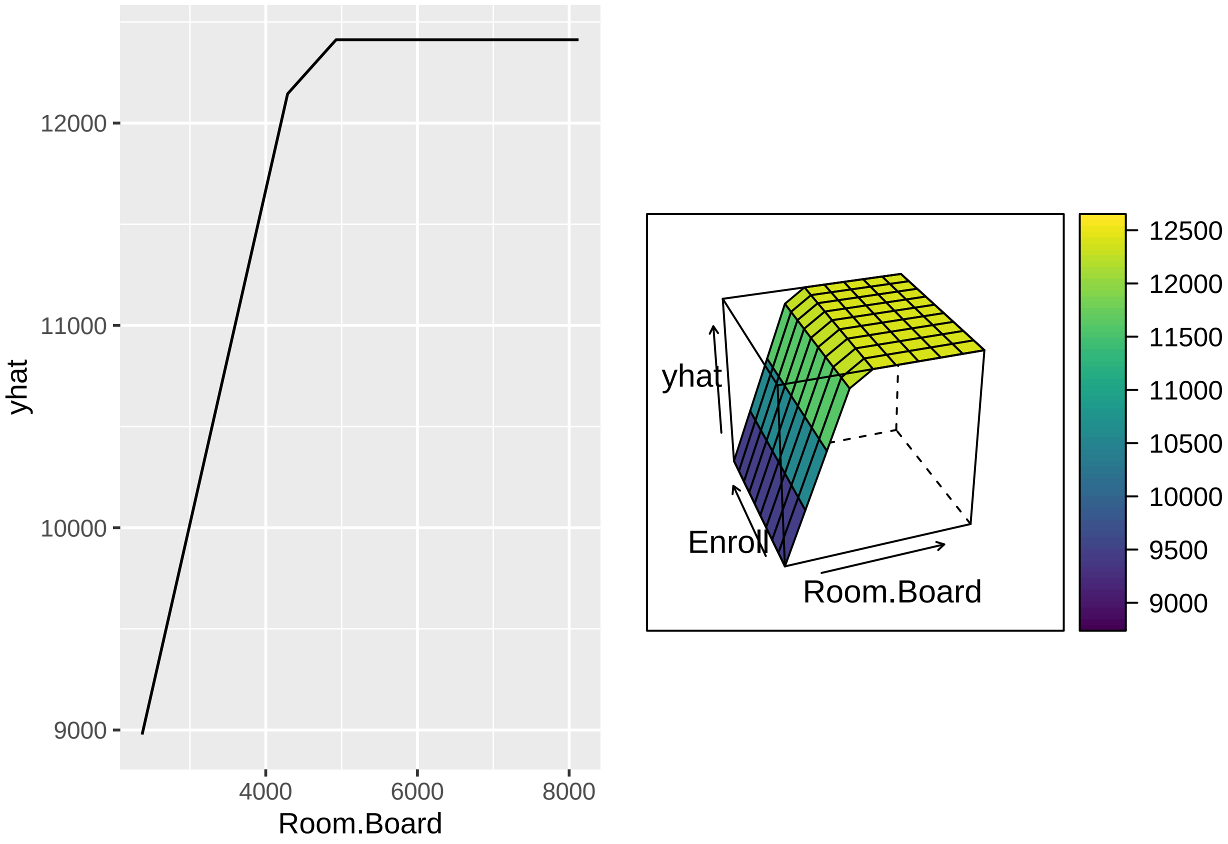
```
##    nprune degree
## 18     10      2
```

```
coef(mars.fit$finalModel)
```

```
##                        (Intercept)                    h(15365-Expend)
##                       1.602840e+04                      -6.313124e-01
##                  h(4450-Room.Board)                  h(perc.alumni-22)
##                      -1.651132e+00                       9.956084e+01
##                  h(22-perc.alumni) h(1546-Accept) * h(perc.alumni-22)
##                      -1.017750e+02                      -1.503182e-01
##                           h(PhD-81)    h(F.Undergrad-1355) * h(PhD-45)
##                       1.173551e+02                      -1.052312e-02
##    h(F.Undergrad-1355) * h(45-PhD)                    h(Accept-2342)
##                      -8.322623e-02                       7.016650e-01
```

Presenting the partial dependence plot of an arbitrary predictor in the final model.

```
p1 <- partial(mars.fit, pred.var = c("Room.Board"), grid.resolution = 10) %>% autoplot()

p2 <- partial(mars.fit, pred.var = c("Room.Board", "Enroll"), grid.resolution = 10) %>%
      plotPartial(levelplot = FALSE, zlab = "yhat", drape = TRUE,
                  screen = list(z = 20, x = -60))

grid.arrange(p1, p2, ncol = 2)
```



(e) Based on the above GAM and MARS models, predict the out-of-state tuition of Columbia University.

```
pred.gam <- predict(gam.m3, newdata = data_2)
pred.mars <- predict(mars.fit, newdata = data_2)
pred.gam
```

```
##        1
## 19249.31
```

```
pred.mars
```

```
##             y
## [1,] 16698.41
```

The predicted out-of-state tuition of Columbia University, based on the GAM model is 19406.71 and based on the MARS model is 16698.41.