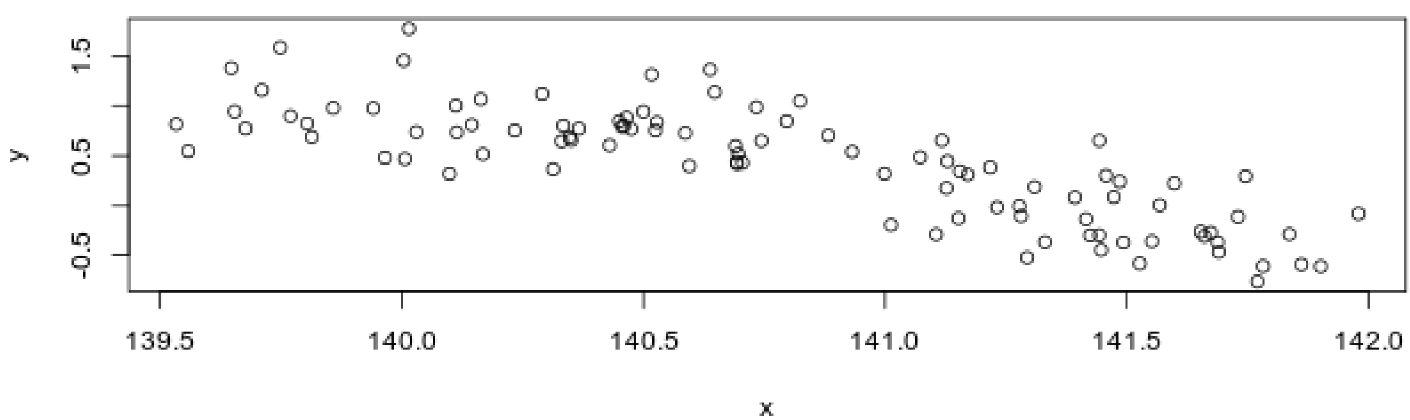
Chapter 7

Lab

Polynomial Functions and Cut Points

```
load(paste0(here::here(), "/ISLR/7.R.RData"))

plot(x, y)
```




```
fit <- lm(y ~ x)
fit2 <- lm(y ~ 1 + x + I(x^2))
```

```
wage <- data.table(ISLR::Wage)
```

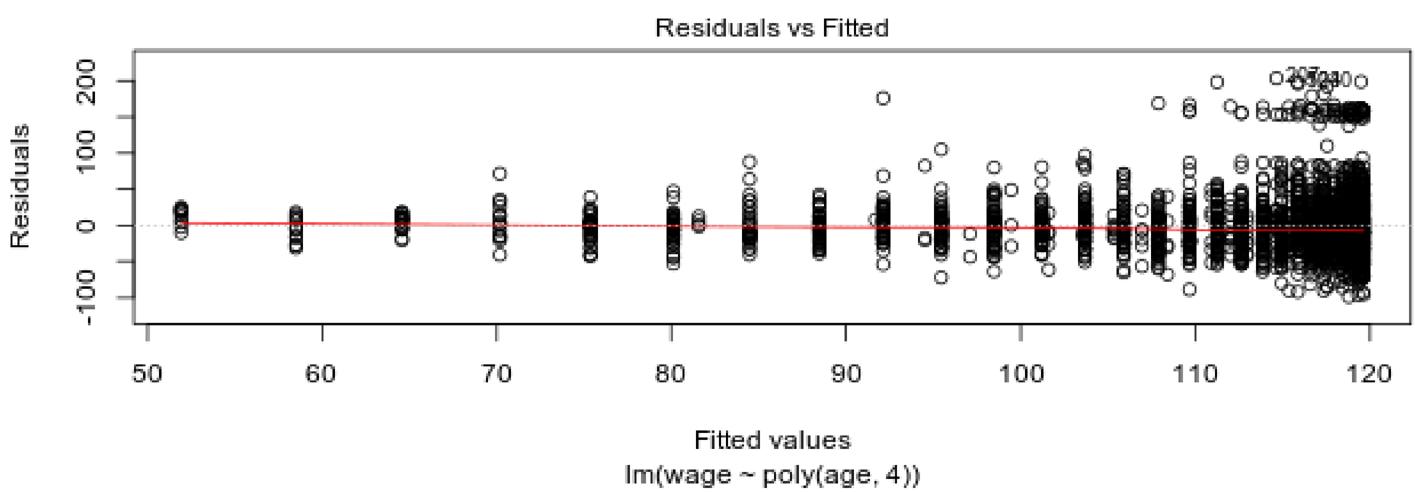
Polynomial Regression and Step Functions

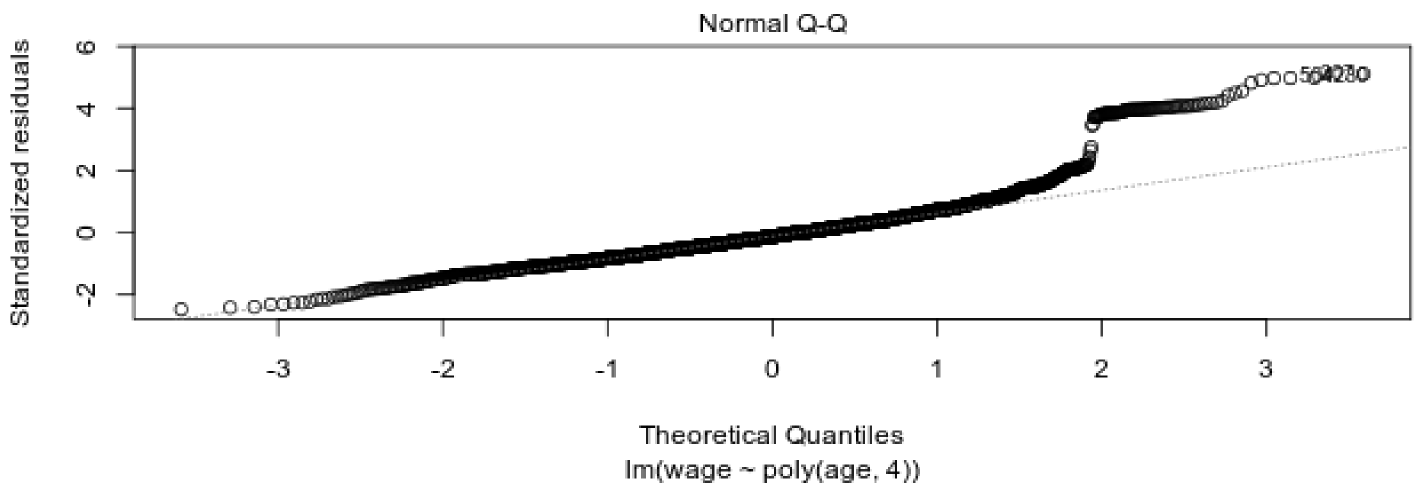
```
fit <- lm(wage ~ poly(age, 4), data = wage)

summary(fit)
```

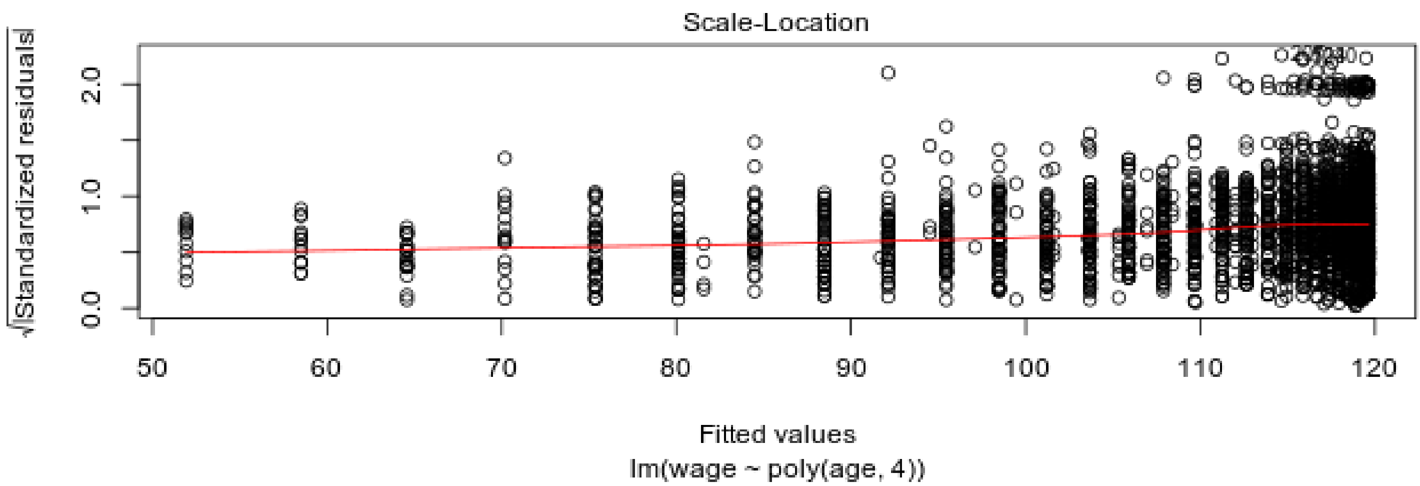
```
Call:
lm(formula = wage ~ poly(age, 4), data = wage)

Residuals:
```

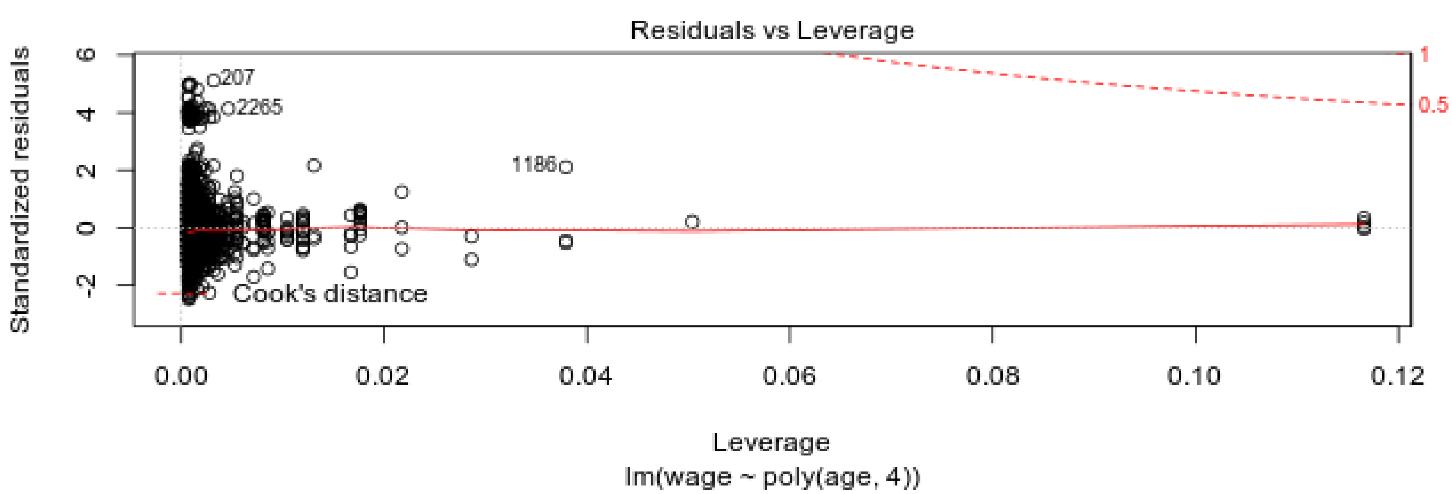
```
    Min      1Q  Median      3Q     Max 
-98.707 -24.626  -4.993  15.217 203.693 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    111.7036     0.7287 153.283  < 2e-16 ***
poly(age, 4)1  447.0679    39.9148  11.201  < 2e-16 ***
poly(age, 4)2 -478.3158    39.9148 -11.983  < 2e-16 ***
poly(age, 4)3  125.5217    39.9148   3.145  0.00168 ** 
poly(age, 4)4  -77.9112    39.9148  -1.952  0.05104 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 39.91 on 2995 degrees of freedom
Multiple R-squared:  0.08626,	Adjusted R-squared:  0.08504 
F-statistic: 70.69 on 4 and 2995 DF,  p-value: < 2.2e-16
```

```
plot(fit)
```




Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(wage ~ poly(age, 4))


Scale-Location
√|Standardized residuals|
Fitted values
lm(wage ~ poly(age, 4))


Residuals vs Leverage
Standardized residuals
207
2265
1186
Cook's distance
1
0.5
Leverage
lm(wage ~ poly(age, 4))

```r
coef(summary(fit))
```

```
                 Estimate Std. Error    t value     Pr(>|t|)
(Intercept)    111.70361  0.7287409 153.283015 0.000000e+00
poly(age, 4)1  447.06785 39.9147851  11.200558 1.484604e-28
poly(age, 4)2 -478.31581 39.9147851 -11.983424 2.355831e-32
poly(age, 4)3  125.52169 39.9147851   3.144742 1.678622e-03
poly(age, 4)4  -77.91118 39.9147851  -1.951938 5.103865e-02
```

```r
fit2 <- lm(wage ~ poly(age, 4, raw = T), data = wage)
coef(summary(fit2))
```

```
                            Estimate   Std. Error   t value     Pr(>|t|)
(Intercept)            -1.841542e+02 6.004038e+01 -3.067172 0.0021802539
poly(age, 4, raw = T)1  2.124552e+01 5.886748e+00  3.609042 0.0003123618
poly(age, 4, raw = T)2 -5.638593e-01 2.061083e-01 -2.735743 0.0062606446
poly(age, 4, raw = T)3  6.810688e-03 3.065931e-03  2.221409 0.0263977518
poly(age, 4, raw = T)4 -3.203830e-05 1.641359e-05 -1.951938 0.0510386498
```

Alternative:

```r
fit2a <- lm(wage ~ age + I(age^2) + I(age^3) + I(age^4), data = wage)
coef(summary(fit2a))
```

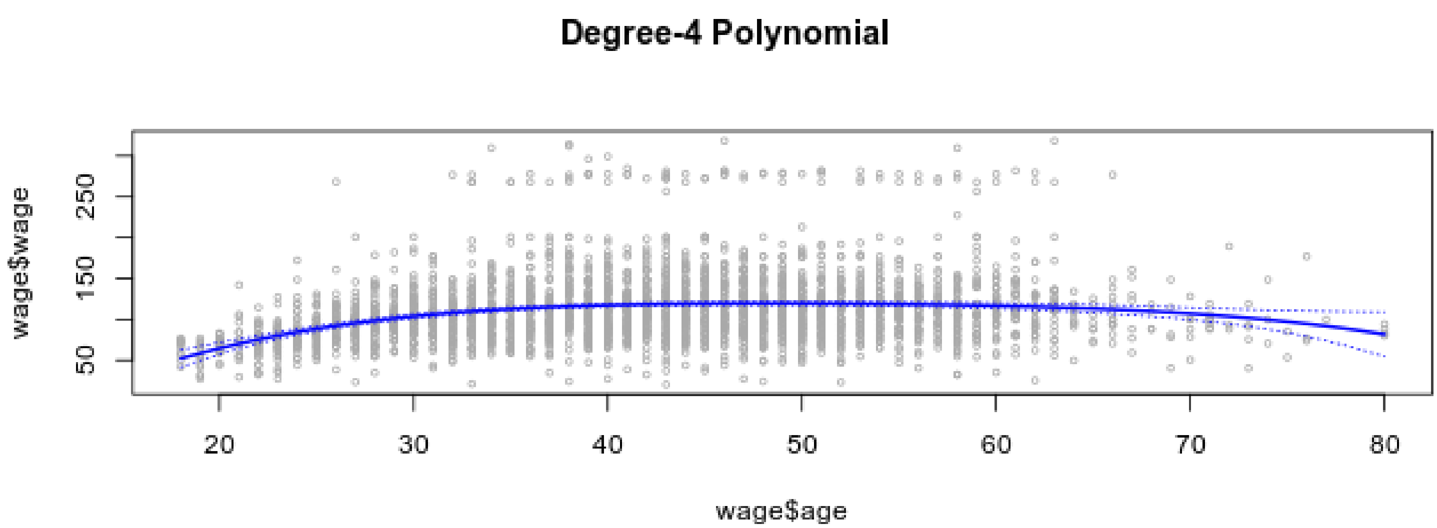
```
                 Estimate   Std. Error   t value     Pr(>|t|)
(Intercept) -1.841542e+02 6.004038e+01 -3.067172 0.0021802539
age          2.124552e+01 5.886748e+00  3.609042 0.0003123618
I(age^2)    -5.638593e-01 2.061083e-01 -2.735743 0.0062606446
I(age^3)     6.810688e-03 3.065931e-03  2.221409 0.0263977518
I(age^4)    -3.203830e-05 1.641359e-05 -1.951938 0.0510386498
```

```r
fit2b <- lm(wage ~ cbind(age, age^2, age^3, age^4), data = wage)
coef(fit2b)
```

```
                      (Intercept) cbind(age, age^2, age^3, age^4)age
                    -1.841542e+02                       2.124552e+01
   cbind(age, age^2, age^3, age^4)    cbind(age, age^2, age^3, age^4)
                    -5.638593e-01                       6.810688e-03
   cbind(age, age^2, age^3, age^4)
                    -3.203830e-05
```

```r
agelims <- range(wage$age)
age.grid <- seq(from = agelims[1], to = agelims[2])

pred <- predict(fit, newdata = list(age = age.grid), se = T)

se.bands <- cbind(pred$fit + 2*pred$se.fit, pred$fit - 2*pred$se.fit)
```

```r
par(mfrow = c(1, 1), mar = c(4.5, 4.5, 1, 1), oma = c(0, 0, 4, 0))
plot(wage$age, wage$wage, xlim = agelims, cex = .5, col = "darkgrey")
title("Degree-4 Polynomial", outer = T)
lines(age.grid, pred$fit, lwd = 2, col = "blue")
matlines(age.grid, se.bands, lwd = 1, col = "blue", lty = 3)
```




```r
pred2 <- predict(fit2, newdata = list(age = age.grid), se = T)
max(abs(pred$fit - pred2$fit))
```

```
[1] 7.81597e-11
```

```r
fit1 <- lm(wage ~ age, data = wage)
fit2 <- lm(wage ~ poly(age, 2), data = wage)
fit3 <- lm(wage ~ poly(age, 3), data = wage)
fit4 <- lm(wage ~ poly(age, 4), data = wage)
fit5 <- lm(wage ~ poly(age, 5), data = wage)

anova(fit1, fit2, fit3, fit4, fit5)
```

```
Analysis of Variance Table

Model 1: wage ~ age
Model 2: wage ~ poly(age, 2)
Model 3: wage ~ poly(age, 3)
Model 4: wage ~ poly(age, 4)
Model 5: wage ~ poly(age, 5)
  Res.Df     RSS Df Sum of Sq        F    Pr(>F)
1   2998 5022216
2   2997 4793430  1    228786 143.5931 < 2.2e-16 ***
3   2996 4777674  1     15756   9.8888  0.001679 **
```

```
4   2995 4771604  1      6070   3.8098  0.051046 .
5   2994 4770322  1      1283   0.8050  0.369682
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coef(summary(fit5))
```

```
                  Estimate Std. Error     t value     Pr(>|t|)
(Intercept)    111.70361  0.7287647 153.2780243 0.000000e+00
poly(age, 5)1  447.06785 39.9160847  11.2001930 1.491111e-28
poly(age, 5)2 -478.31581 39.9160847 -11.9830341 2.367734e-32
poly(age, 5)3  125.52169 39.9160847   3.1446392 1.679213e-03
poly(age, 5)4  -77.91118 39.9160847  -1.9518743 5.104623e-02
poly(age, 5)5  -35.81289 39.9160847  -0.8972045 3.696820e-01
```

```r
fit1 <- lm(wage ~ education + age, data = wage)
fit2 <- lm(wage ~ education + poly(age, 2), data = wage)
fit3 <- lm(wage ~ education + poly(age, 3), data = wage)

anova(fit1, fit2, fit3)
```

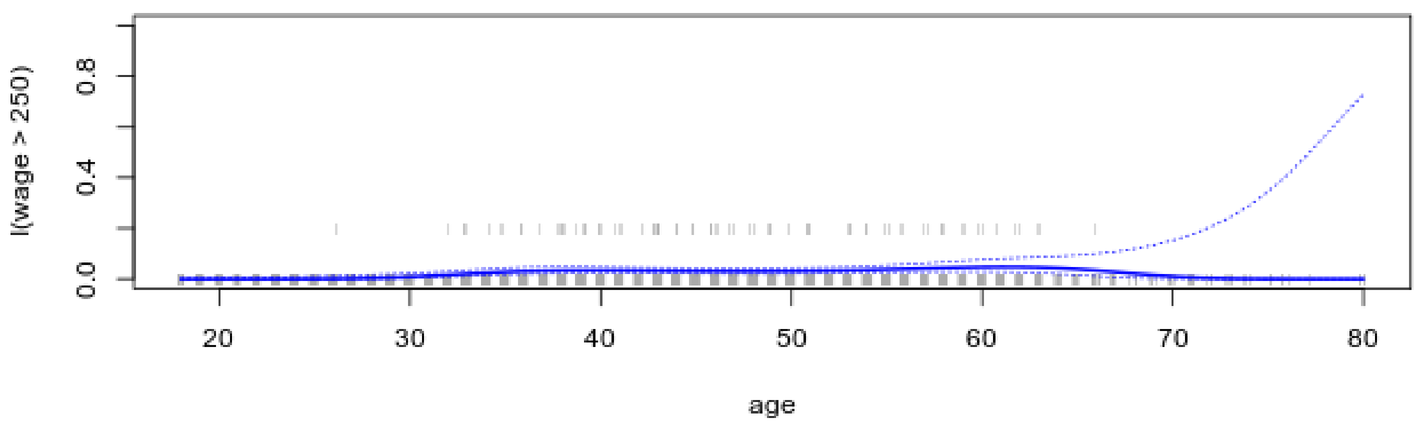
```
Analysis of Variance Table

Model 1: wage ~ education + age
Model 2: wage ~ education + poly(age, 2)
Model 3: wage ~ education + poly(age, 3)
  Res.Df     RSS Df Sum of Sq        F Pr(>F)
1   2994 3867992
2   2993 3725395  1    142597 114.6969 <2e-16 ***
3   2992 3719809  1      5587   4.4936 0.0341 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
fit <- glm(I(wage > 250) ~ poly(age, 4), data = wage, family = "binomial")

pred <- predict(fit, newdata = list(age = age.grid), se = T)

pfit <- exp(pred$fit) / (1 + exp(pred$fit))

se.bands.logit <- cbind(pred$fit + 2 * pred$se.fit, pred$fit - 2*pred$se.fit)

se.bands <- exp(se.bands.logit) / (1 + exp(se.bands.logit))
```

Alternatively:

```r
pred <- predict(fit, newdata = list(age = age.grid), type = "response", se = T)
```

```r
with(wage, {
  plot(age, I(wage > 250), xlim = agelims, type = "n")
  points(jitter(age), I((wage > 250)/5), cex = .5, pch = "|", col = "darkgrey")
  lines(age.grid, pfit, lwd = 2, col = "blue")
  matlines(age.grid, se.bands, lwd = 1, col = "blue", lty = 3)
})
```




```r
table(cut(wage$age, 4))
```

```
(17.9,33.5]   (33.5,49]   (49,64.5] (64.5,80.1]
        750        1399         779          72
```

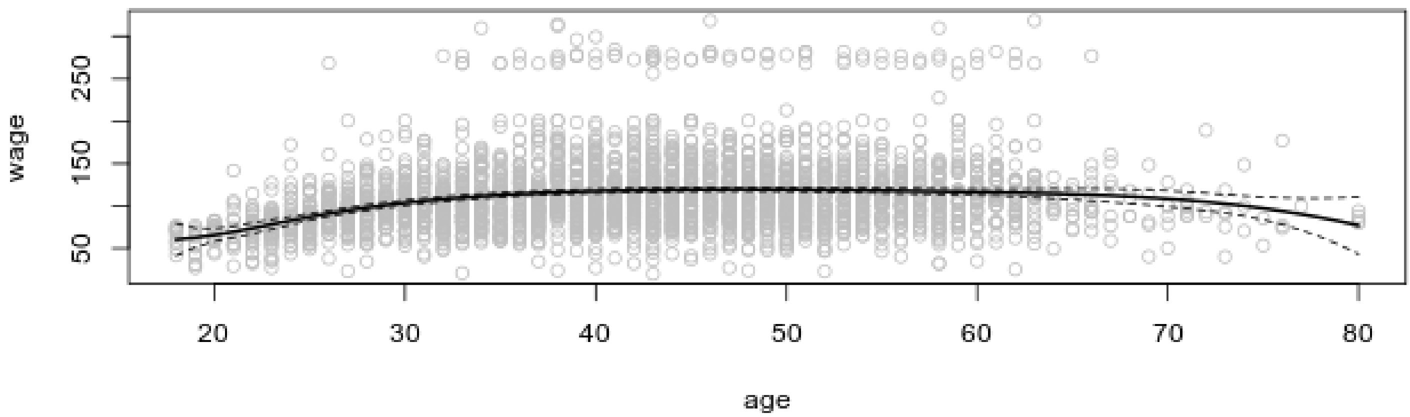
```r
fit <- lm(wage ~ cut(age, 4), data = wage)
coef(summary(fit))
```

```
                        Estimate Std. Error   t value     Pr(>|t|)
(Intercept)            94.158392   1.476069 63.789970 0.000000e+00
cut(age, 4)(33.5,49]   24.053491   1.829431 13.148074 1.982315e-38
cut(age, 4)(49,64.5]   23.664559   2.067958 11.443444 1.040750e-29
cut(age, 4)(64.5,80.1]  7.640592   4.987424  1.531972 1.256350e-01
```

Splines

```r
fit <- lm(wage ~ bs(age, knots = c(25, 40, 60)), data = wage)

pred <- predict(fit, newdata = list(age = age.grid), se = T)

with(wage, {
```

```
   plot(age, wage, col = "gray")
   lines(age.grid, pred$fit, lwd=2)
   lines(age.grid, pred$fit+2*pred$se.fit, lty="dashed")
   lines(age.grid, pred$fit-2*pred$se.fit, lty="dashed")
})
```




```
dim(bs(wage$age, knots = c(25, 40, 60)))
```

```
[1] 3000    6
```

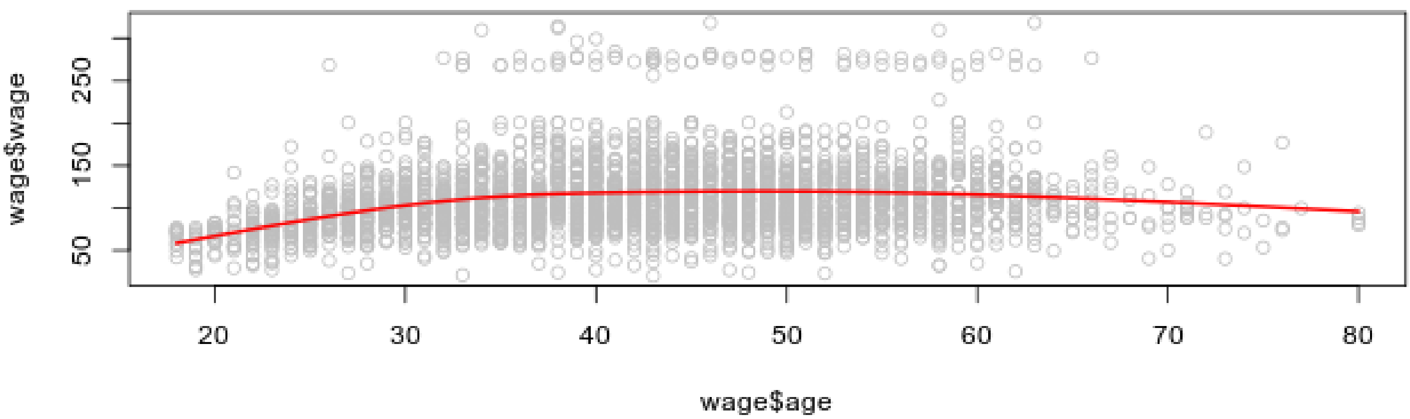
```
dim(bs(wage$age, df = 6))
```

```
[1] 3000    6
```

```
attr(bs(wage$age, df = 6), "knots")
```

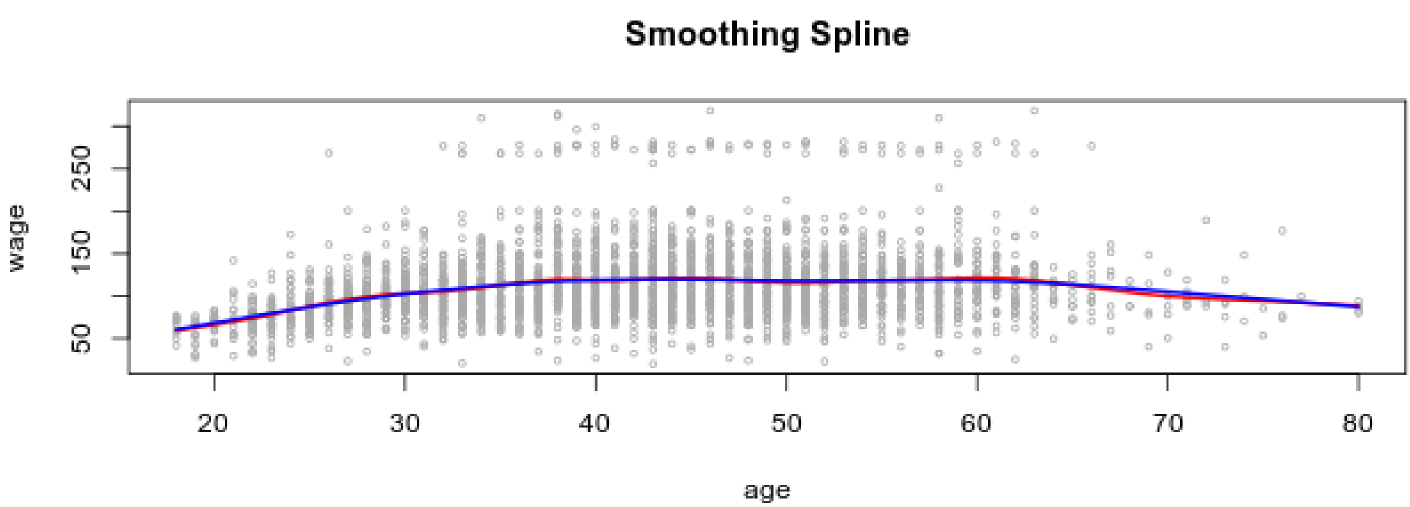
```
  25%   50%   75% 
33.75 42.00 51.00 
```

```
fit2 <- lm(wage ~ ns(age, df = 4), data = wage)
pred2 <- predict(fit2, newdata =  list(age = age.grid), se = T)
par(mfrow=c(1,1))
plot(wage$age, wage$wage, col = "gray")
lines(age.grid, pred2$fit, col = "red", lwd = 2)
```

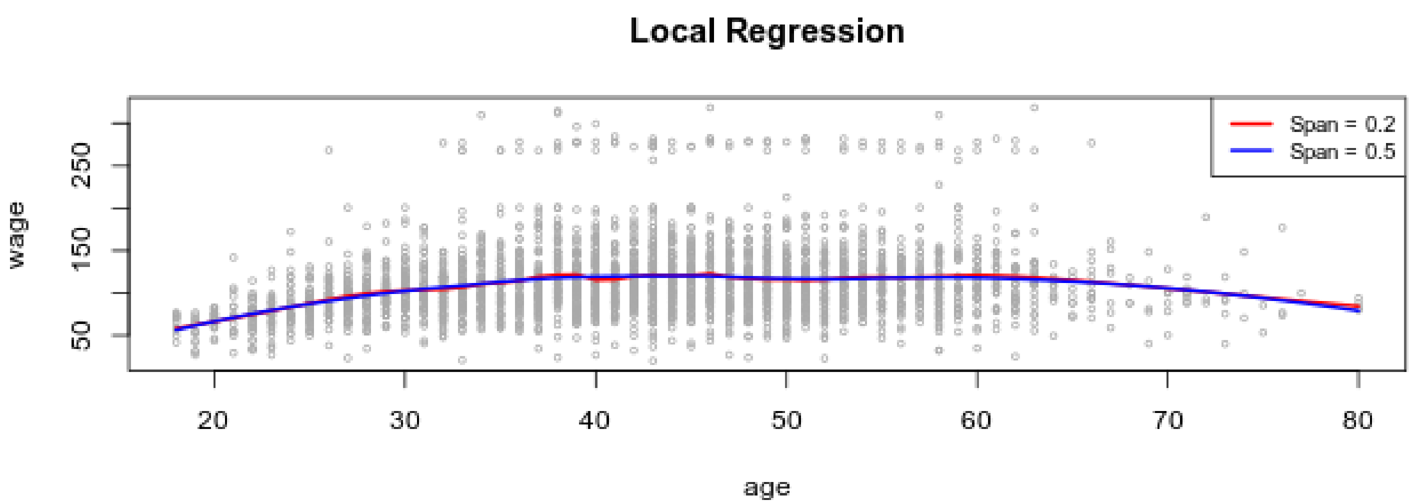



```
with(wage,{
   plot(age, wage, xlim = agelims, cex = .5, col = "darkgrey")
   title("Smoothing Spline")
   fit <- smooth.spline(age, wage, df = 16)
   fit2 <- smooth.spline(age, wage, cv = T)

   lines(fit, col = "red", lwd = 2)
   lines(fit2, col = "blue", lwd = 2)
})
```

```
Warning in smooth.spline(age, wage, cv = T): cross-validation with non-unique
'x' values seems doubtful
```



```r
with(wage, {
  plot(age, wage, xlim = agelims, cex = .5, col = "darkgrey")
  title("Local Regression")
  fit <- loess(wage ~ age, span = .2)
  fit2 <- loess(wage ~ age, span = .5)
  lines(age.grid, predict(fit, data.frame(age = age.grid)), col = "red", lwd = 2)
  lines(age.grid, predict(fit2, data.frame(age = age.grid)), col = "blue", lwd = 2)
  legend("topright", legend = c("Span = 0.2", "Span = 0.5"), col = c("red", "blue"), lty = 1,
})
```

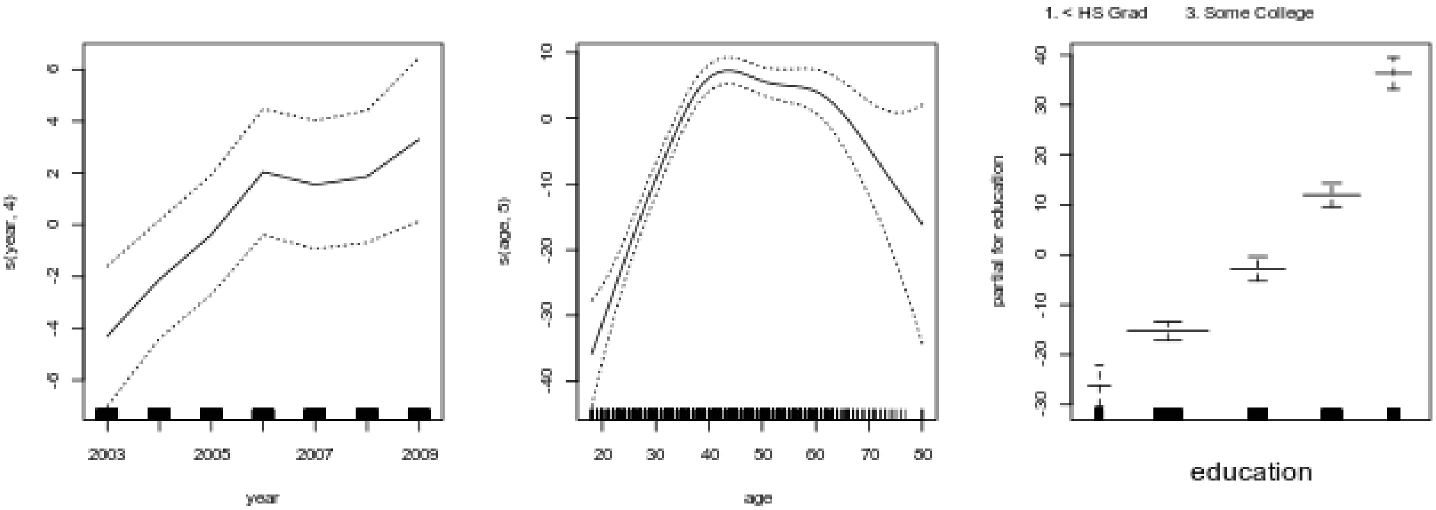


GAMs

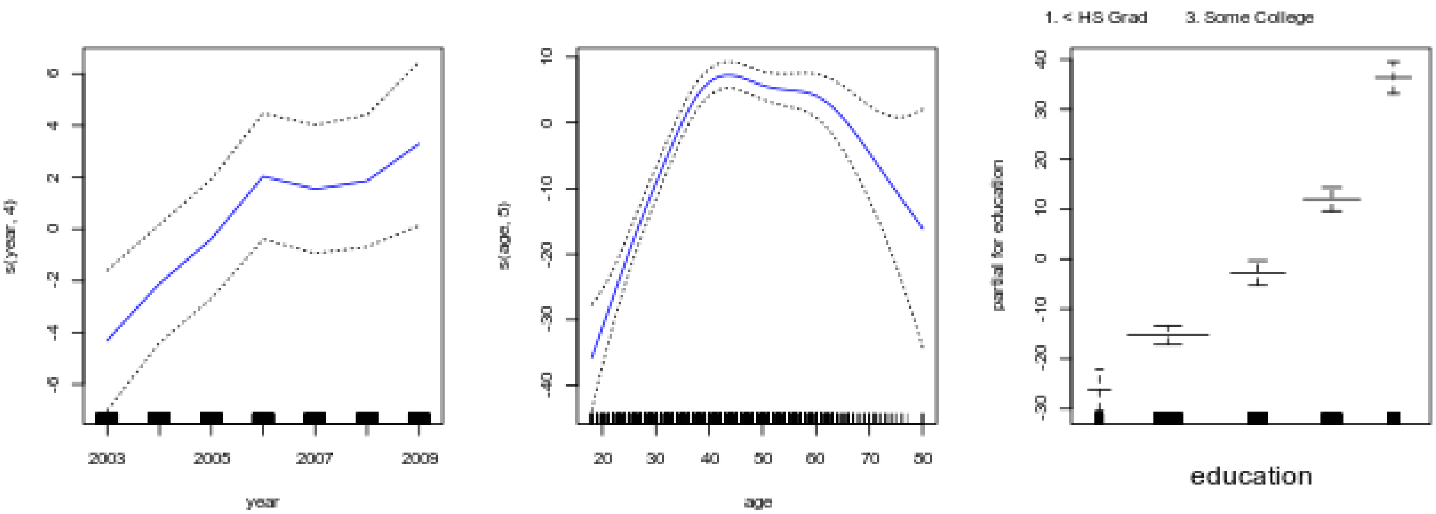
```r
gam1 <- lm(wage ~ ns(year, 4) + ns(age, 5) + education, data = wage)

gam.m3 <- gam(wage ~ s(year, 4) + s(age, 5) + education, data = wage)
```

```
Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
ignored
```

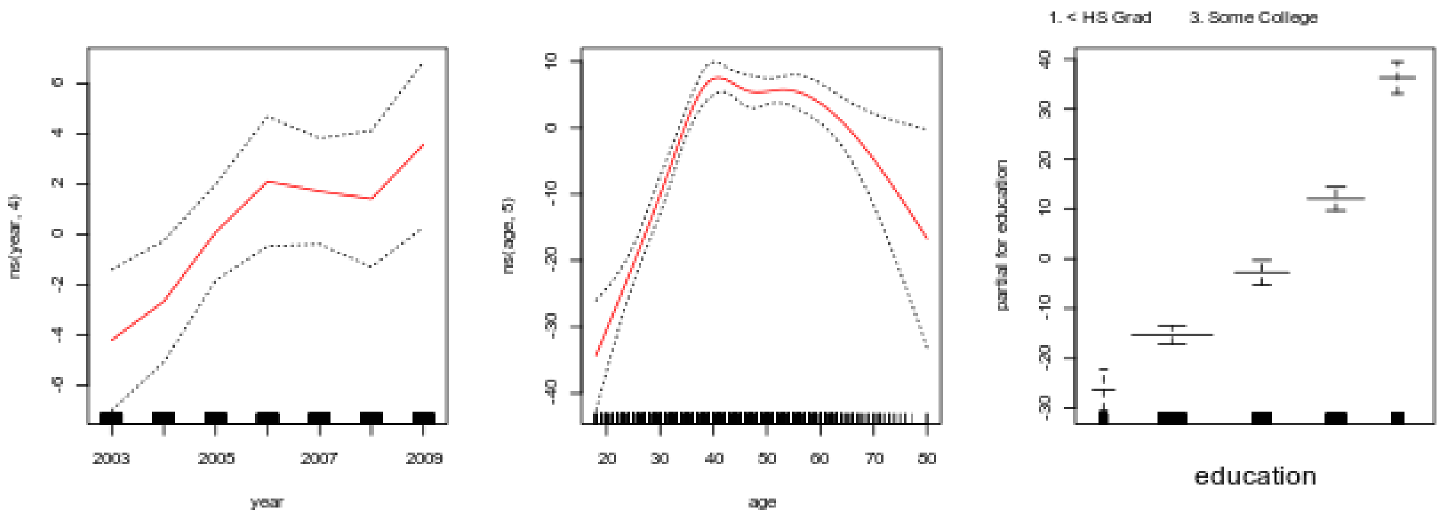
```r
par(mfrow = c(1, 3))
plot.Gam(gam.m3, se = T)
```




```r
par(mfrow = c(1, 3))
plot(gam.m3, se = T, col = "blue")
```




```r
plot.Gam(gam1, se = T, col = "red")
```




```r
gam.m1 <- gam(wage ~ s(age, 5) + education, data = wage)
```

```
Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
ignored
```

```r
gam.m2 <- gam(wage ~ year + s(age, 5) + education, data = wage)
```

```
Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
ignored
```

```r
anova(gam.m1, gam.m2, gam.m3)
```

```
Analysis of Deviance Table

Model 1: wage ~ s(age, 5) + education
Model 2: wage ~ year + s(age, 5) + education
Model 3: wage ~ s(year, 4) + s(age, 5) + education
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      2990    3711731
2      2989    3693842  1  17889.2 0.0001419 ***
3      2986    3689770  3   4071.1 0.3483897
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(gam.m3)
```

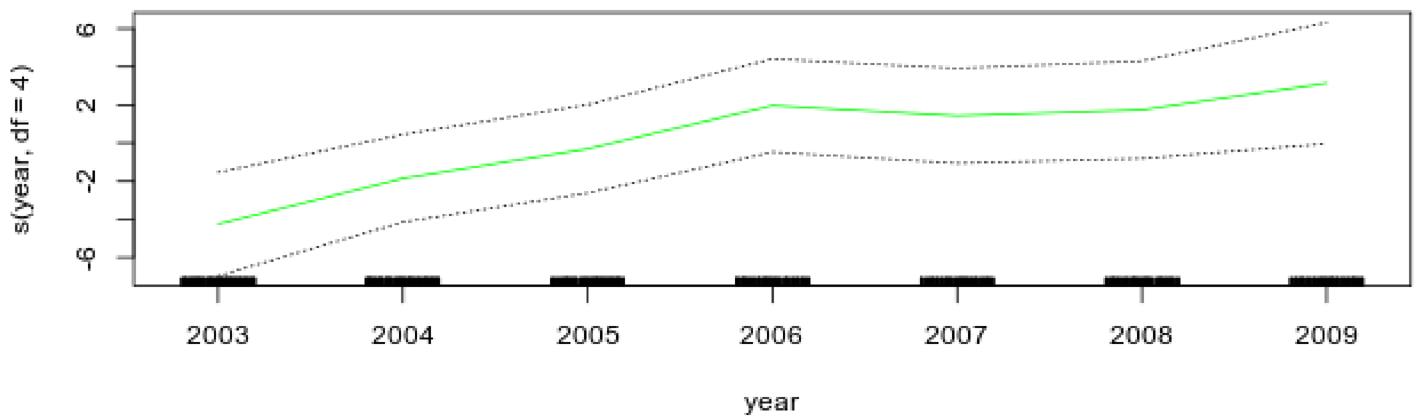
```

Call: gam(formula = wage ~ s(year, 4) + s(age, 5) + education, data = wage)
Deviance Residuals:
    Min      1Q  Median      3Q     Max
-119.43  -19.70   -3.33   14.17  213.48
```

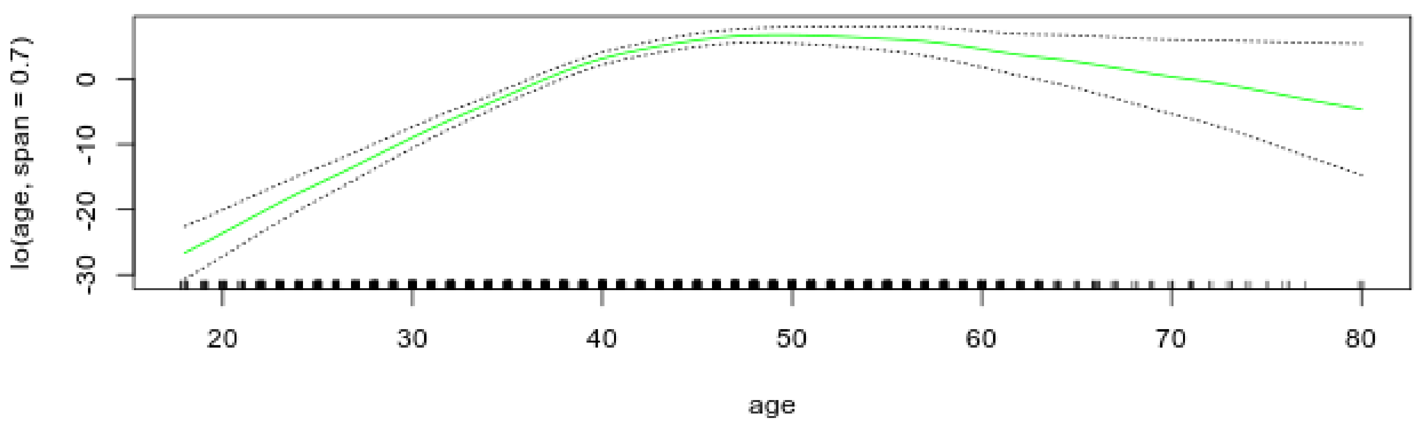
```
(Dispersion Parameter for gaussian family taken to be 1235.69)

    Null Deviance: 5222086 on 2999 degrees of freedom
Residual Deviance: 3689770 on 2986 degrees of freedom
AIC: 29887.75

Number of Local Scoring Iterations: 2

Anova for Parametric Effects
             Df  Sum Sq Mean Sq F value    Pr(>F)
s(year, 4)    1   27162   27162  21.981 2.877e-06 ***
s(age, 5)     1  195338  195338 158.081 < 2.2e-16 ***
education     4 1069726  267432 216.423 < 2.2e-16 ***
Residuals  2986 3689770    1236
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Anova for Nonparametric Effects
            Npar Df Npar F  Pr(F)
(Intercept)
s(year, 4)        3  1.086 0.3537
s(age, 5)         4 32.380 <2e-16 ***
education
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

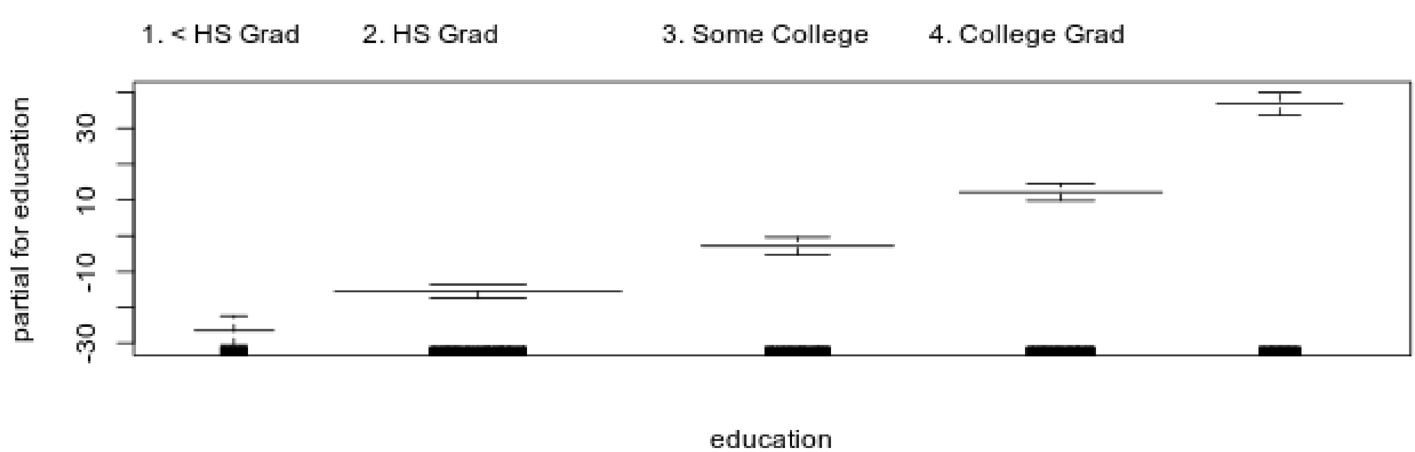
```r
pred <- predict(gam.m2, newdata = wage)

gam.lo <- gam(wage ~ s(year, df = 4) + lo(age, span = 0.7) + education, data = wage)
```

```
Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
ignored
```

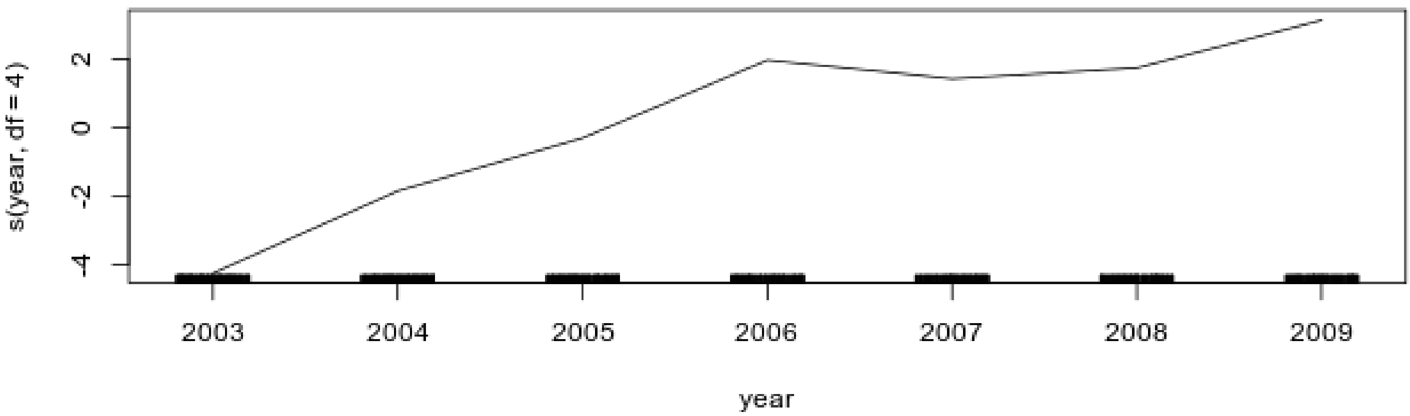
```r
plot.Gam(gam.lo, se = T, col = "green")
```

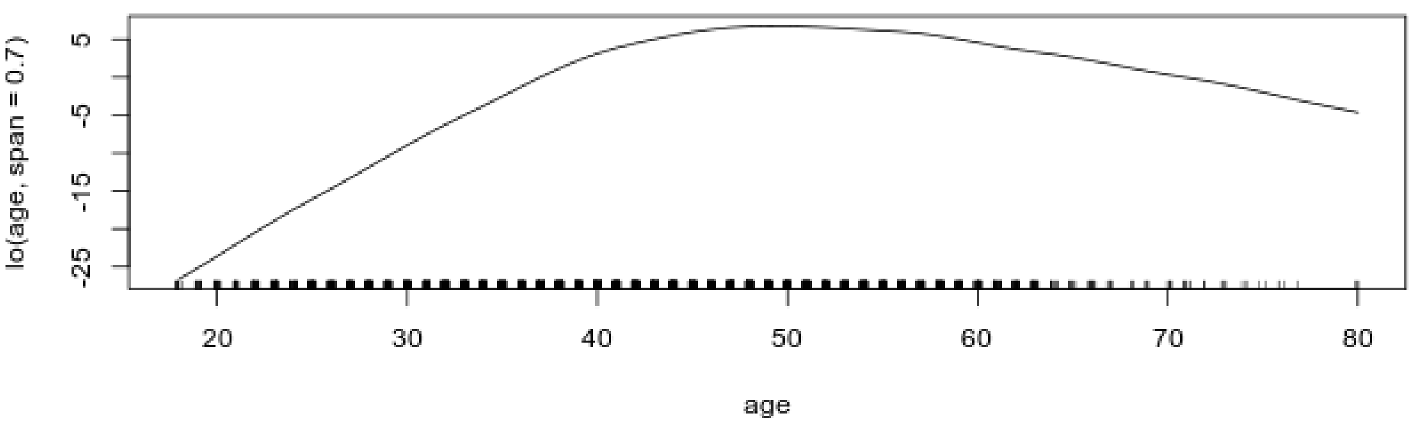

s(year, df = 4)
6
2
-2
-6
2003
2004
2005
2006
2007
2008
2009
year

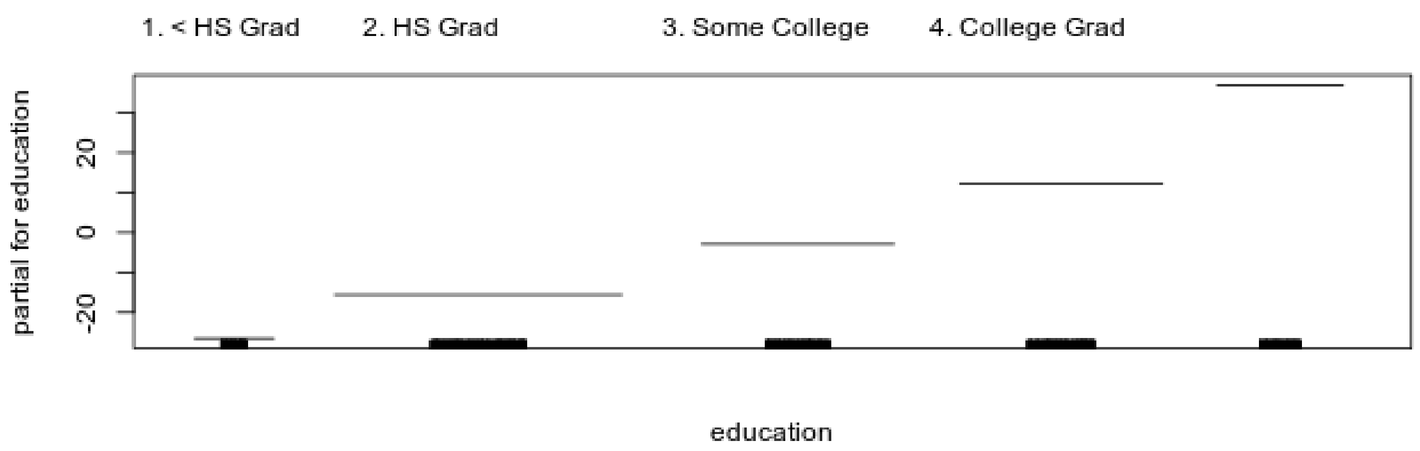

lo(age, span = 0.7)
0
-10
-20
-30
20
30
40
50
60
70
80
age


1. < HS Grad
2. HS Grad
3. Some College
4. College Grad
partial for education
30
10
-10
-30
education

```r
plot(gam.lo)
```


s(year, df = 4)
-4
-2
0
2
2003
2004
2005
2006
2007
2008
2009
year

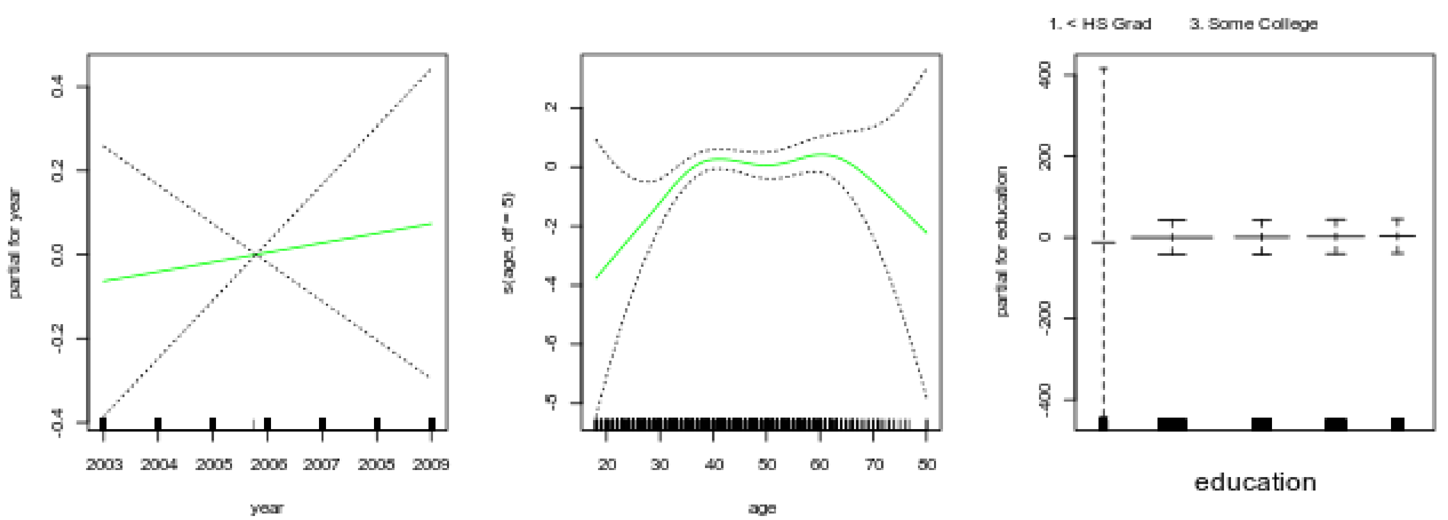

lo(age, span = 0.7)
-25
-15
-5
5
20
30
40
50
60
70
80
age




```
gam.lr <- gam(I(wage > 250) ~ year + s(age, df = 5) + education, family = binomial, data = wage
```

```
Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
ignored
```

```
par(mfrow = c(1, 3))
plot(gam.lr, se = T, col = "green")
```




```
table(wage$education, I(wage$wage > 250))
```

```
                     FALSE TRUE
  1. < HS Grad         268    0
  2. HS Grad           966    5
  3. Some College      643    7
```

```
  4. College Grad        663    22
  5. Advanced Degree     381    45
```

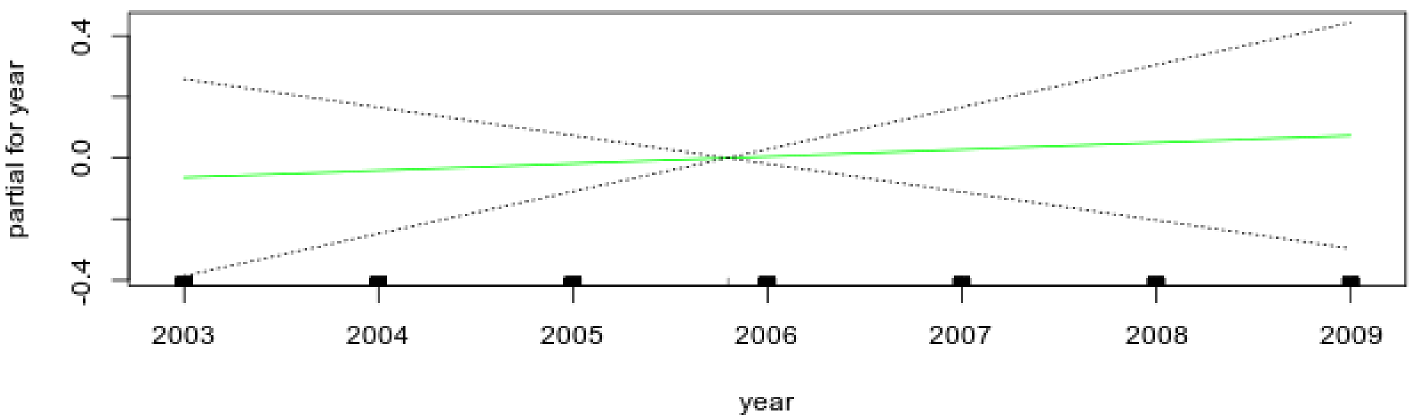
```r
levels(wage$education)
```

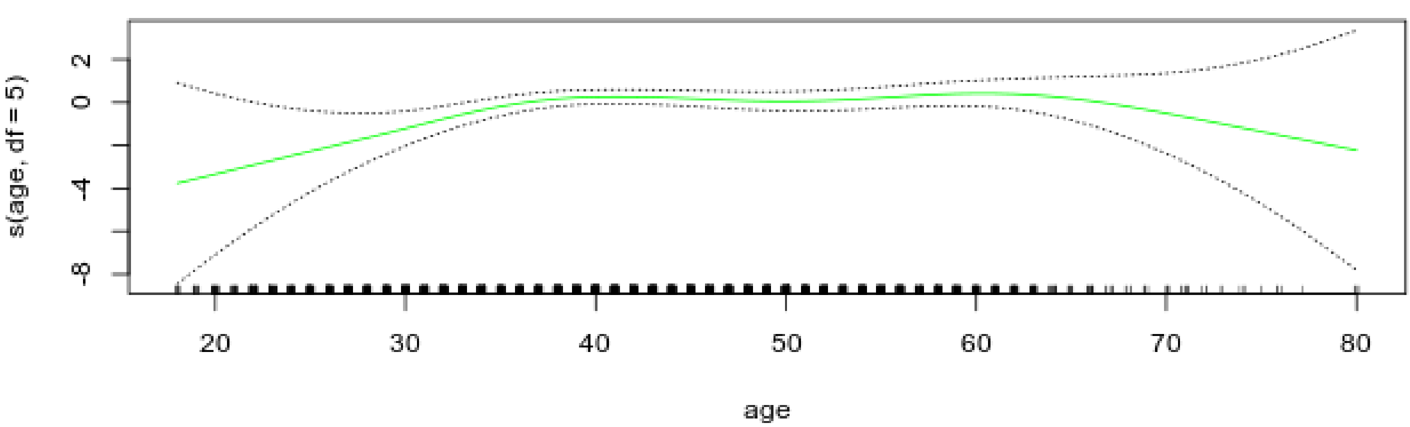
```
[1] "1. < HS Grad"       "2. HS Grad"         "3. Some College"
[4] "4. College Grad"    "5. Advanced Degree"
```

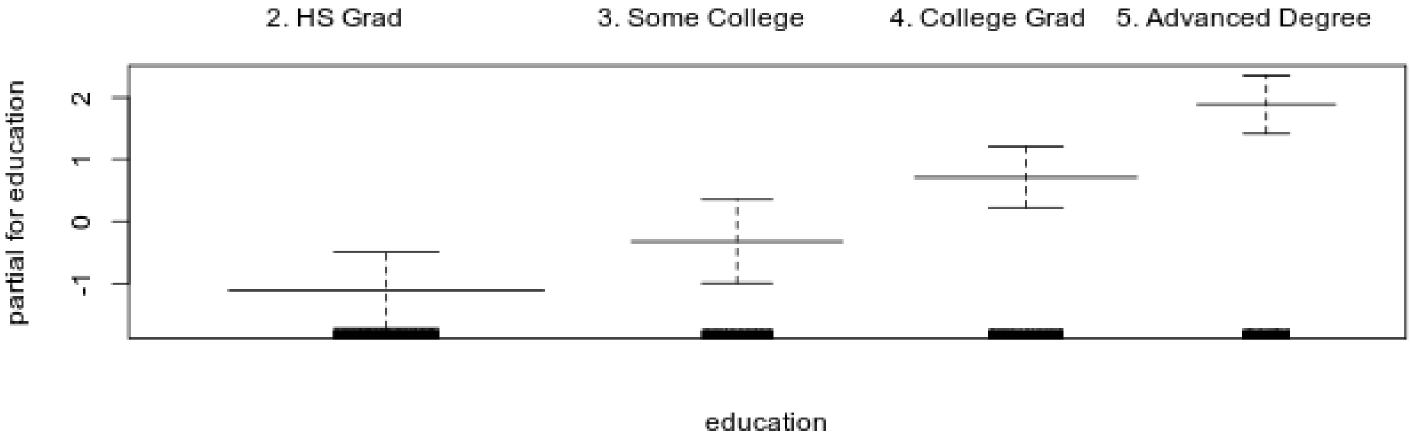
```r
gam.lr.s <- gam(I(wage > 250) ~ year + s(age, df = 5) + education, family = binomial, data = wa
```

```
Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
ignored
```

```r
plot(gam.lr.s, se = T, col = "green")
```







Applied

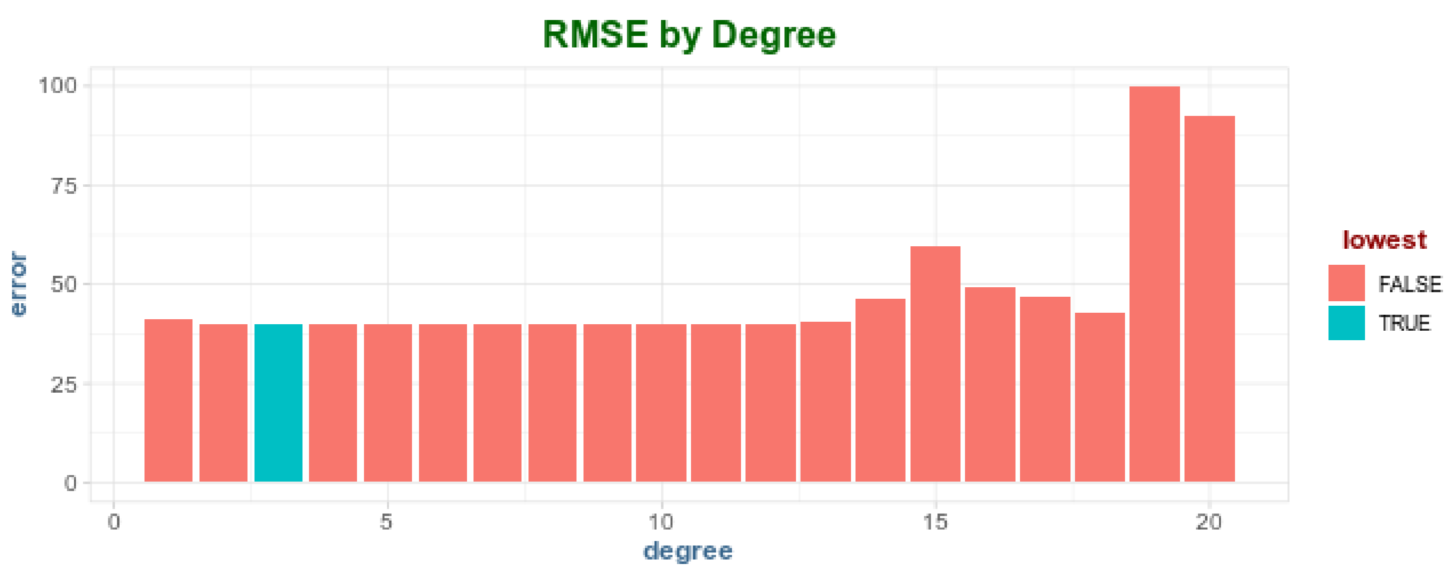
In this exercise, you will further analyze the wage data set considered throughout this chapter.

```
test.size <- .7
index <- sample(nrow(wage), nrow(wage) * test.size, replace = F)

train <- wage[index]
test <- wage[!index]
```

a.) Perform polynomial regression to predict wage using age. Use cross-validation to select the optimal degree d for the polynomial. What degree was chosen, and how does this compare to the result of hypothesis testing using ANOVA? Make a plot of the fit obtained.

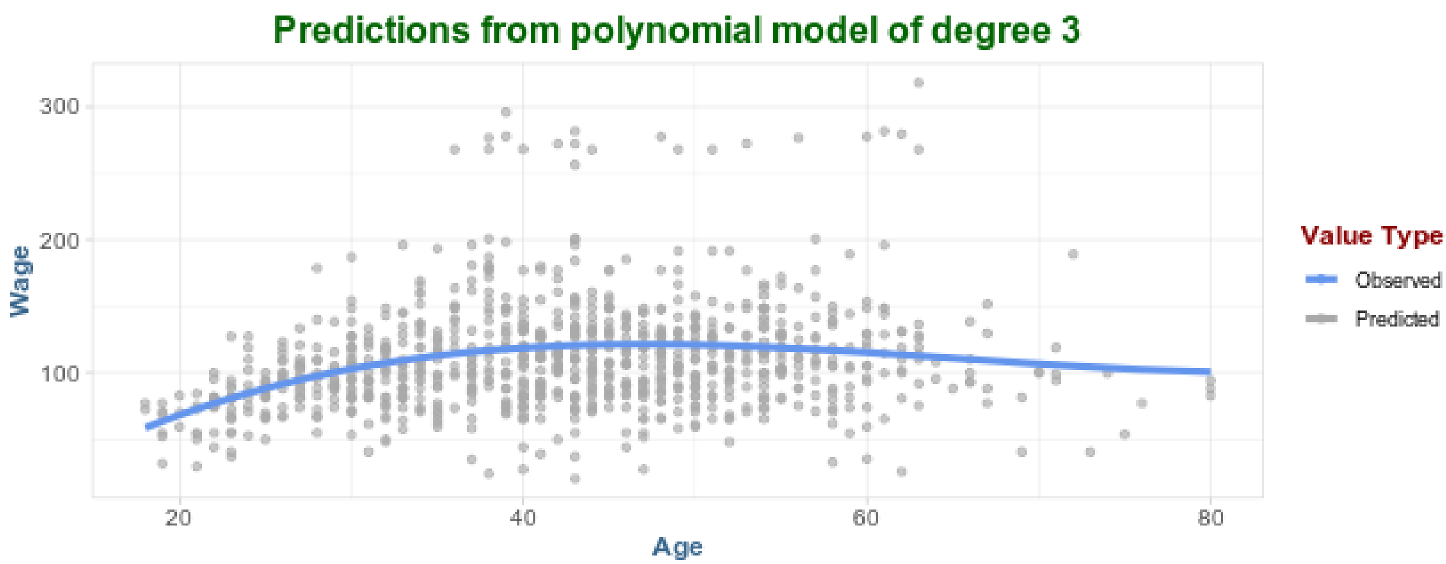
```
degree <- 20; folds = 10
cv.errors <- numeric(degree)

fold.size <- nrow(train) / folds

for(deg in 1:degree)
{
   # 10 fold cv
   errors <- numeric(folds)
   for(fold in 1:folds)
   {
      holdout <- seq((fold - 1) * fold.size, fold * fold.size)

      cv.train <- train[!holdout]
      cv.test <- train[holdout]
```

```
      fit <- lm(wage ~ poly(age, deg), data = cv.train)

      pred <- predict(fit, newdata = cv.test, type = "response")

      errors[fold] <- sqrt(mean((cv.test$wage - pred)^2))
   }
   cv.errors[deg] <- mean(errors)
}

lowest.error <- which.min(cv.errors)

cv.results <- data.table(degree = 1:degree, error = cv.errors)[, lowest := degree == lowest.err

ggplot(cv.results, aes(degree, error, fill = lowest)) +
   geom_bar(stat = "identity") +
   labs(title = "RMSE by Degree")
```




```
model <- lm(wage ~ poly(age, lowest.error), data = train)

test %>%
    mutate(predictions = predict(model, test)) %>%
    ggplot(aes(age, wage, col = 'darkgrey')) +
    geom_point(alpha = .65) +
    geom_line(aes(age, predictions, col = 'cornflowerblue'), size = 1.5) +
    scale_color_manual(name = 'Value Type',
                       labels = c('Observed', 'Predicted'),
                       values = c('cornflowerblue', 'darkgrey' )) +
    labs(x = 'Age', y = 'Wage',
```

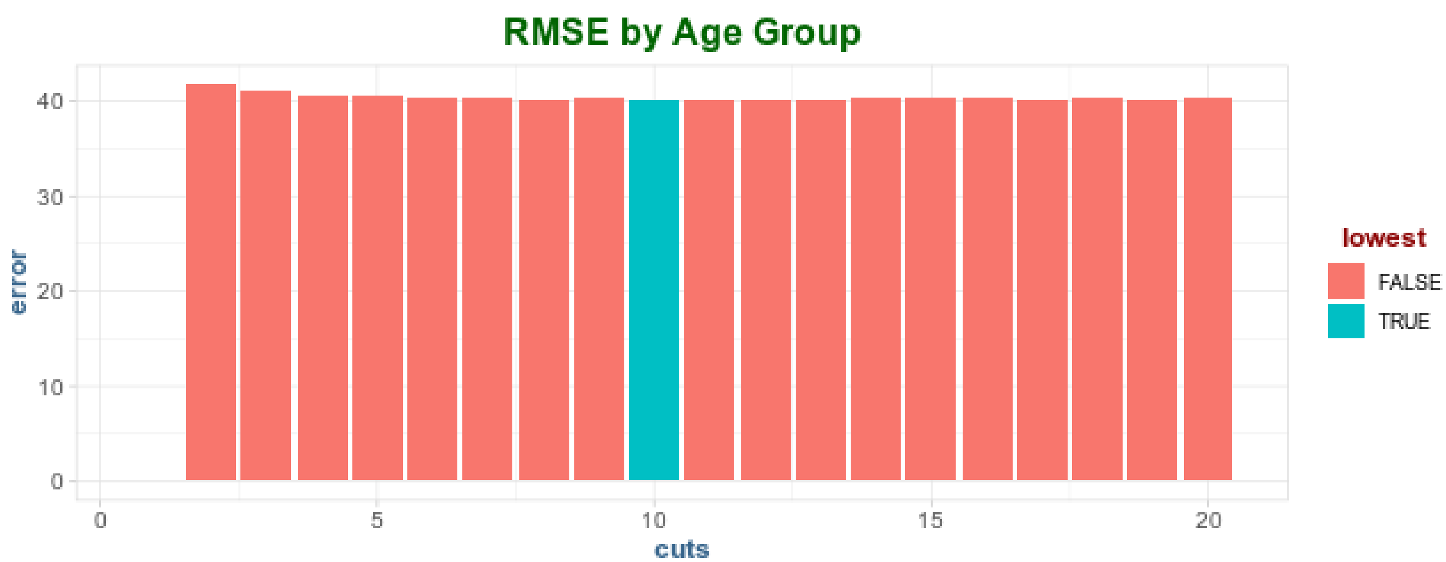
```
        title = paste0('Predictions from polynomial model of degree ', lowest.error))
```



b.) Fit a step function to predict wage using age, and perform cross-validation to choose the optimal number of cuts. Make a plot of the fit obtained.

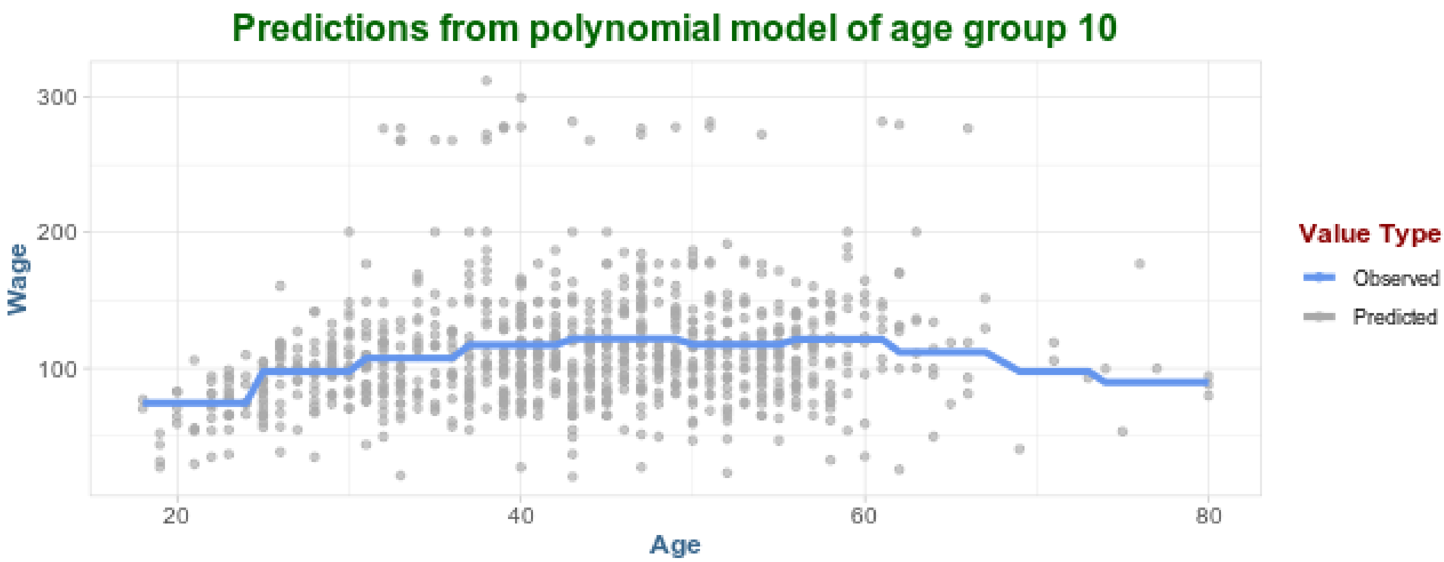
```
cuts <- 20; folds = 10
cv.errors <- numeric(degree)

fold.size <- nrow(train) / folds

for(cuts in 2:cuts)
{
   # 10 fold cv
   errors <- numeric(folds)

   # apply cut here so CV train/test have same levels
   train$AgeGroup <- cut(train$age, cuts)

   for(fold in 1:folds)
   {
      holdout <- seq((fold - 1) * fold.size, fold * fold.size)

      cv.train <- train[!holdout]
      cv.test <- train[holdout]

      fit <- lm(wage ~ I(AgeGroup), data = cv.train)

      pred <- predict(fit, newdata = cv.test, type = "response")
```

```
        errors[fold] <- sqrt(mean((cv.test$wage - pred)^2))
    }

    cv.errors[cuts] <- mean(errors)
}

lowest.error <- which.min(cv.errors[cv.errors != 0])

cv.results <- data.table(cuts = 1:cuts, error = cv.errors)[, lowest := cuts == lowest.error]

ggplot(cv.results, aes(cuts, error, fill = lowest)) +
    geom_bar(stat = "identity") +
    labs(title = "RMSE by Age Group")
```




```
wage.grouped <- wage
wage.grouped$AgeGroup <- cut(wage.grouped$age, lowest.error)

test.size <- .7
index <- sample(nrow(wage), nrow(wage) * test.size, replace = F)

train <- wage.grouped[index]
test <- wage.grouped[!index]

model <- lm(wage ~ I(AgeGroup), data = train)

test %>%
     mutate(predictions = predict(model, test)) %>%
     ggplot(aes(age, wage, col = 'darkgrey')) +
     geom_point(alpha = .65) +
```

```r
    geom_line(aes(age, predictions, col = 'cornflowerblue'), size = 1.5) +
    scale_color_manual(name = 'Value Type',
                       labels = c('Observed', 'Predicted'),
                       values = c('cornflowerblue', 'darkgrey' )) +
    labs(x = 'Age', y = 'Wage',
         title = paste0('Predictions from polynomial model of age group ', lowest.error))
```

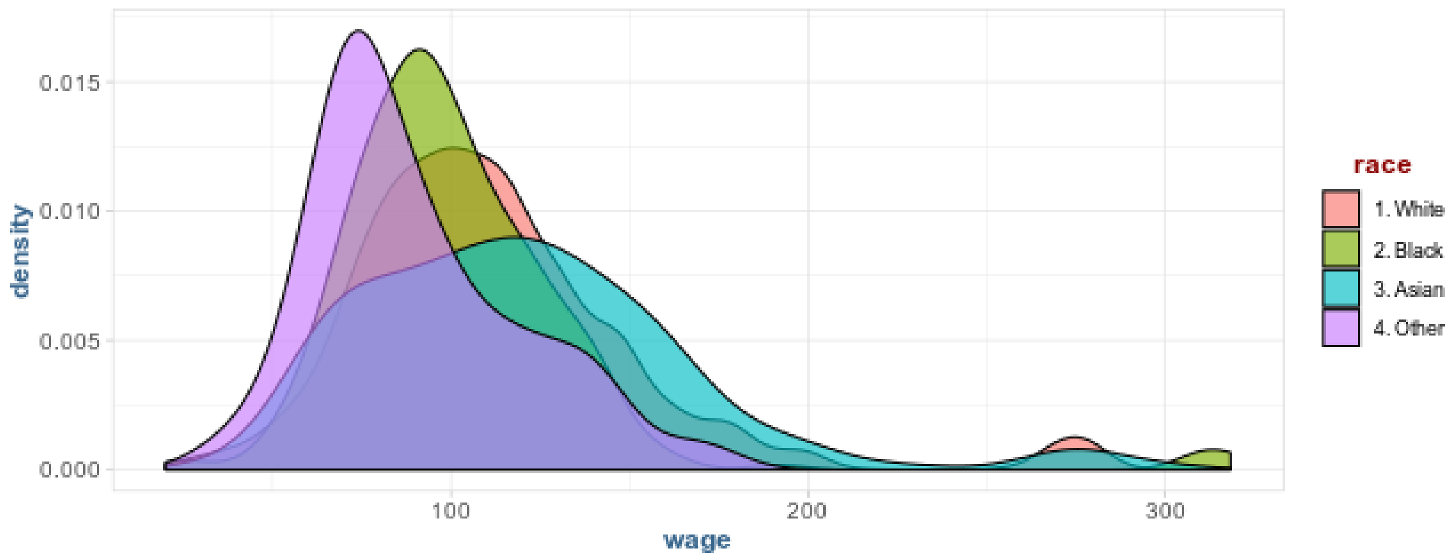


The wage data set contains a number of other features not explored in this chapter, such as marital status (*marit1*), job class (*jobclass*), and others. Explore the relationships between some of these other predictors and wage, and use non-linear fitting techniques in order to fit flexible models to the data. Create plots of the results obtained, and write a summary of your findings.

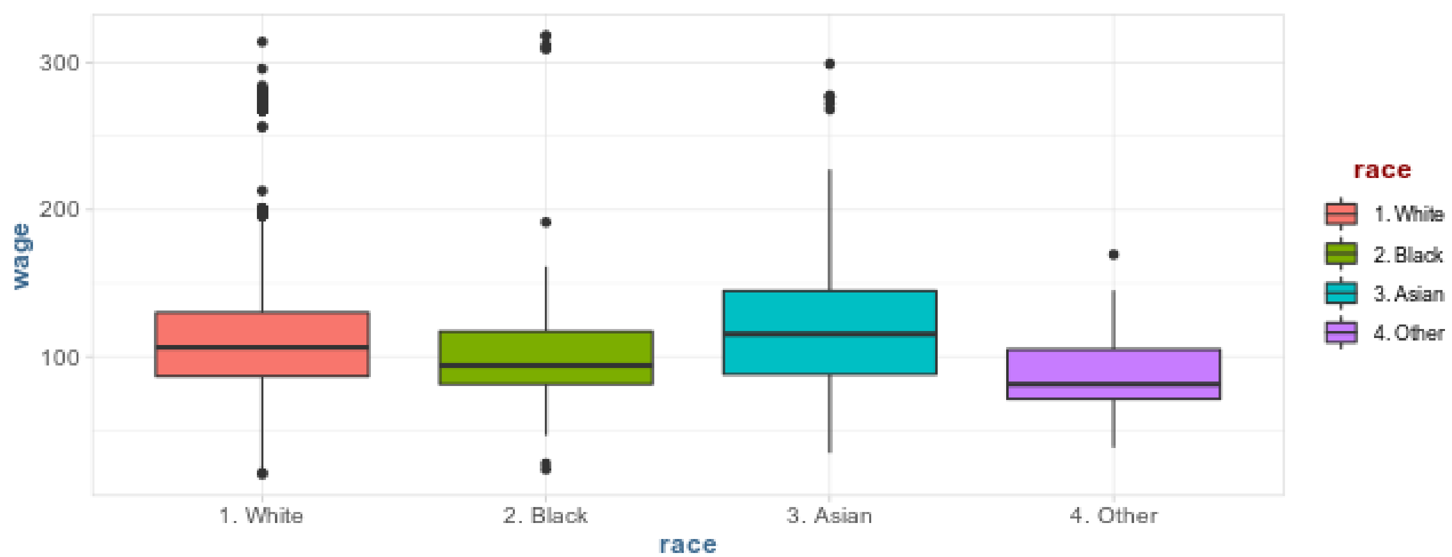
```r
head(wage)
```

```
   year age            maritl     race       education             region
1: 2006  18 1. Never Married 1. White    1. < HS Grad 2. Middle Atlantic
2: 2004  24 1. Never Married 1. White 4. College Grad 2. Middle Atlantic
3: 2003  45        2. Married 1. White 3. Some College 2. Middle Atlantic
4: 2003  43        2. Married 3. Asian 4. College Grad 2. Middle Atlantic
5: 2005  50       4. Divorced 1. White      2. HS Grad 2. Middle Atlantic
6: 2008  54        2. Married 1. White 4. College Grad 2. Middle Atlantic
         jobclass         health health_ins  logwage      wage
1:  1. Industrial      1. <=Good      2. No 4.318063  75.04315
2: 2. Information 2. >=Very Good      2. No 4.255273  70.47602
3:  1. Industrial      1. <=Good     1. Yes 4.875061 130.98218
4: 2. Information 2. >=Very Good     1. Yes 5.041393 154.68529
5: 2. Information      1. <=Good     1. Yes 4.318063  75.04315
6: 2. Information 2. >=Very Good     1. Yes 4.845098 127.11574
```

```r
ggplot(wage, aes(wage, group = race, fill = race)) +
   geom_density(alpha = .65)
```

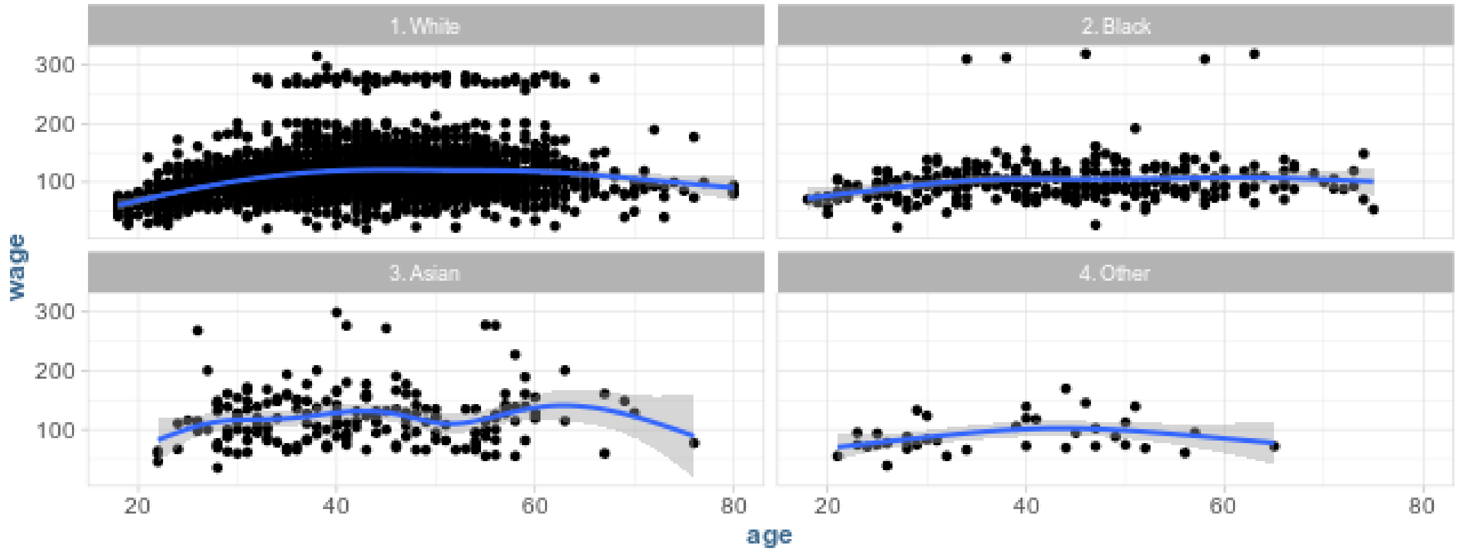



```
ggplot(wage, aes(race, wage, fill = race)) +
  geom_boxplot()
```

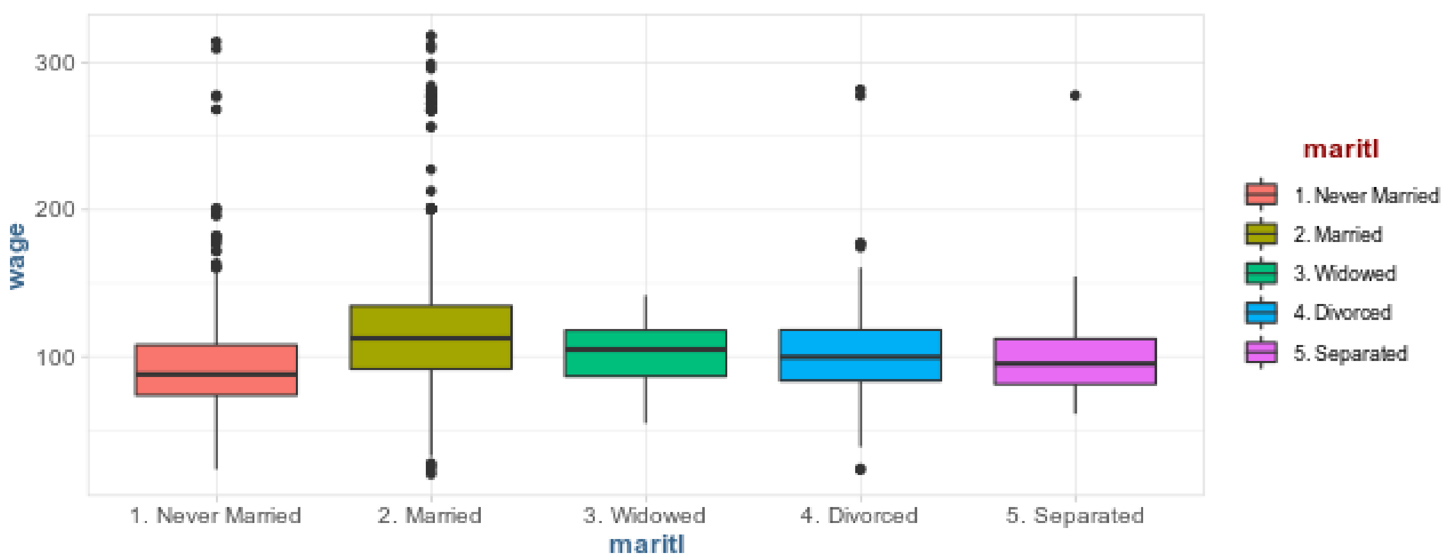



```
ggplot(wage, aes(age, wage)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~race)
```

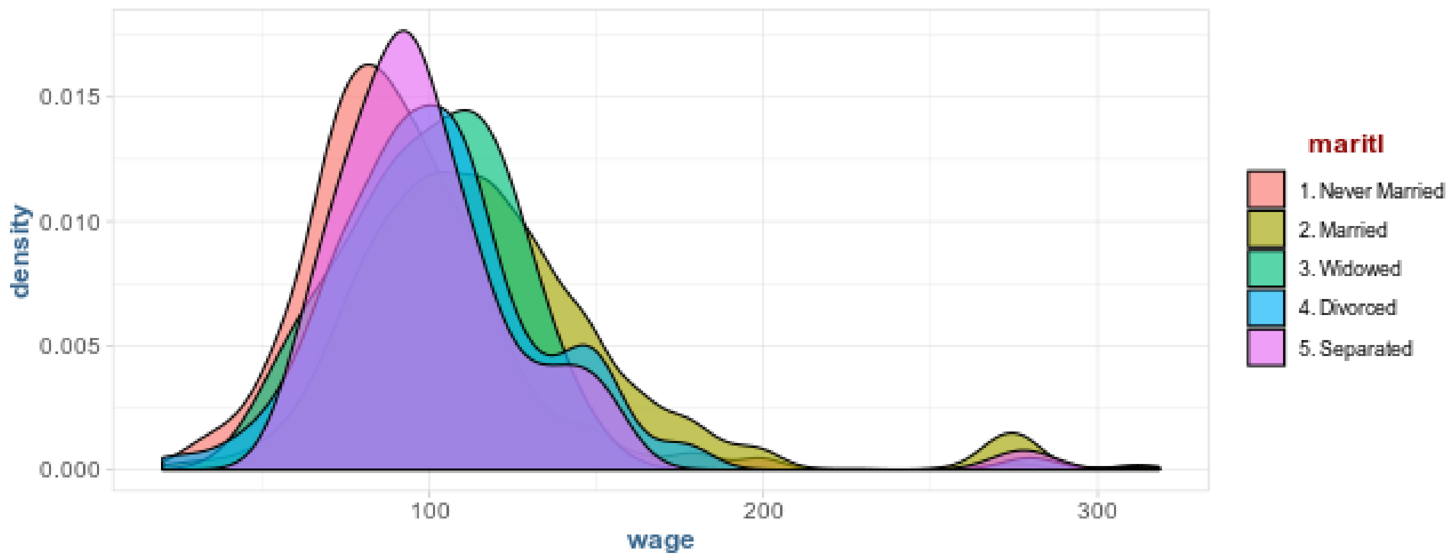
```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```r
ggplot(wage, aes(maritl, wage, fill = maritl)) +
   geom_boxplot()
```

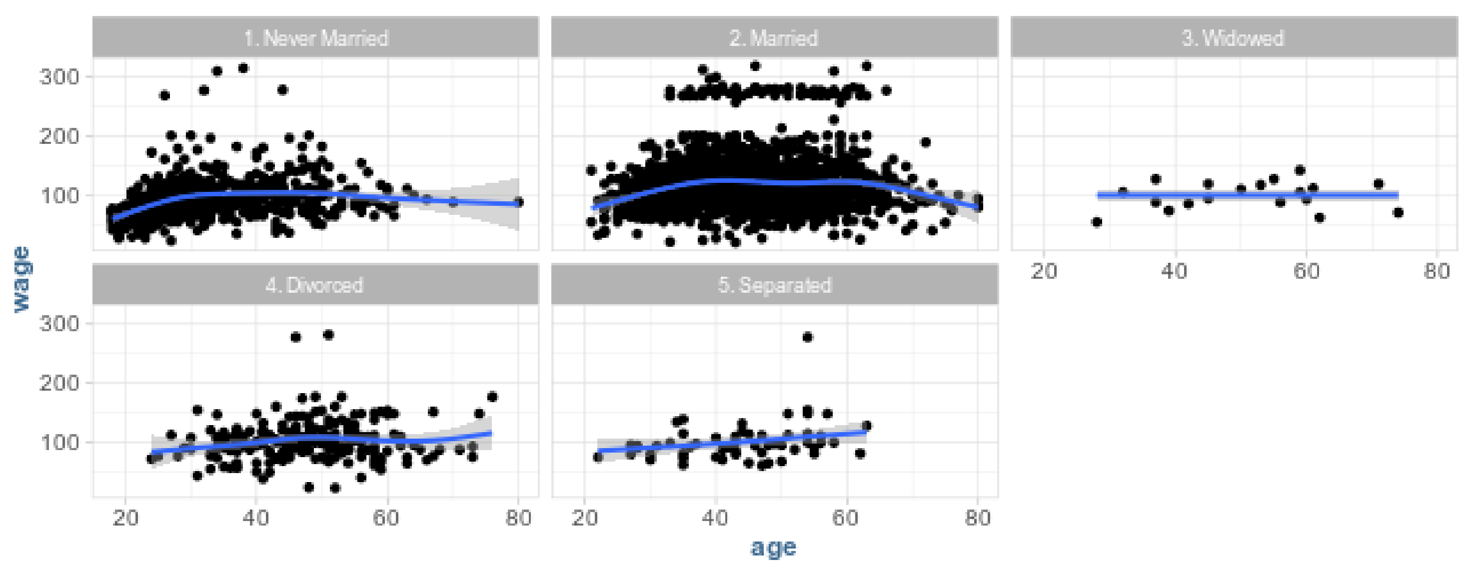



```r
ggplot(wage, aes(wage, fill = maritl)) +
   geom_density(alpha = .65)
```

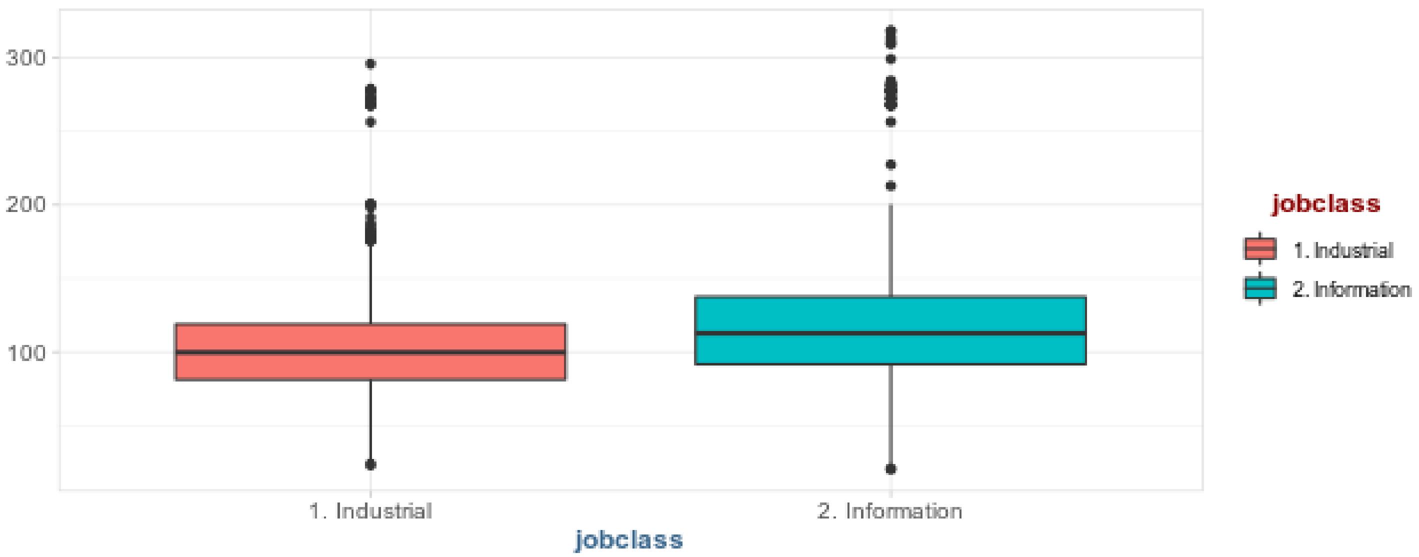



```r
ggplot(wage, aes(age, wage)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~maritl)
```

```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```r
ggplot(wage, aes(jobclass, wage, fill = jobclass)) +
  geom_boxplot()
```

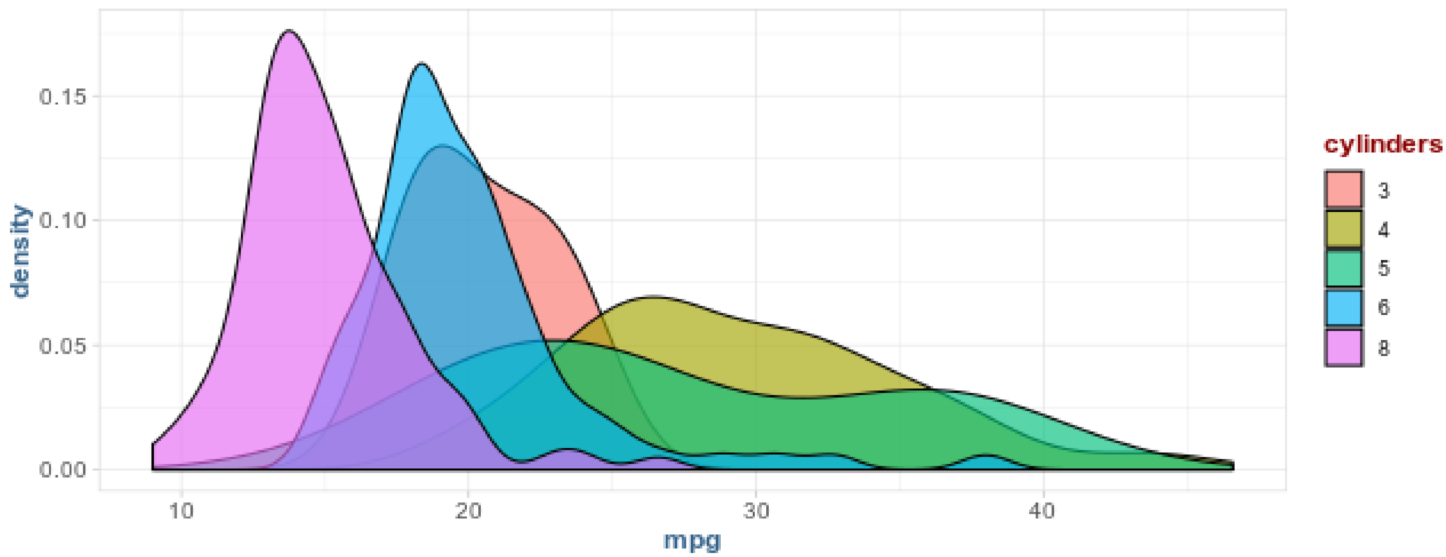


Fit some non-linear models investigated in this chapter to the **Auto** data set. Is there evidence for non-linear relationships in this data set? Create some informative plots to justify your answer.

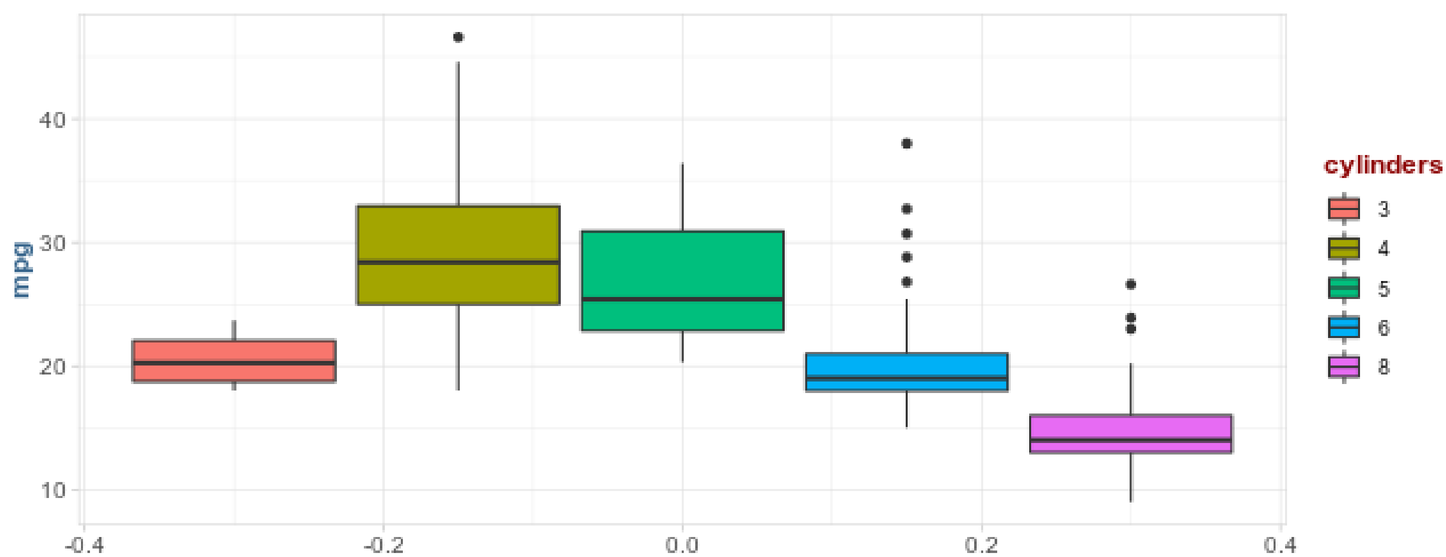
```
auto <- as.data.table(ISLR::Auto)

auto$cylinders <- as.factor(auto$cylinders)

head(auto)
```

```
   mpg cylinders displacement horsepower weight acceleration year origin
1:  18         8          307        130   3504         12.0   70      1
2:  15         8          350        165   3693         11.5   70      1
3:  18         8          318        150   3436         11.0   70      1
4:  16         8          304        150   3433         12.0   70      1
5:  17         8          302        140   3449         10.5   70      1
6:  15         8          429        198   4341         10.0   70      1
                        name
1: chevrolet chevelle malibu
2:         buick skylark 320
3:        plymouth satellite
4:             amc rebel sst
5:               ford torino
6:          ford galaxie 500
```

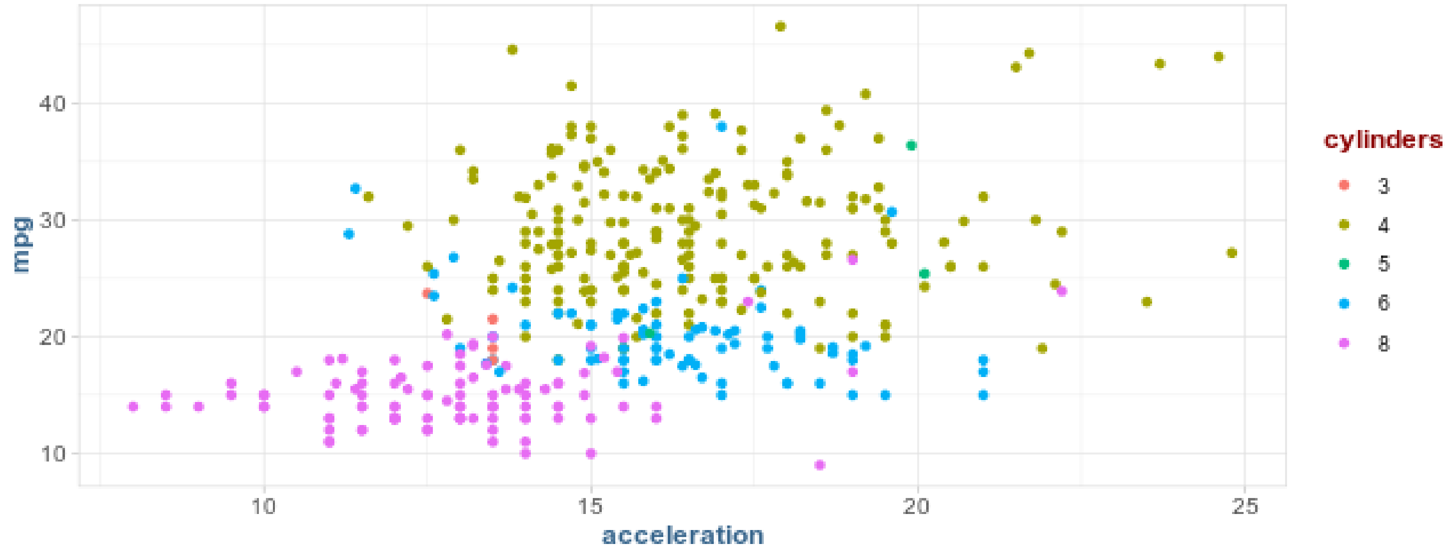
```
ggplot(auto, aes(mpg, group = cylinders, fill = cylinders)) +
   geom_density(alpha = .65)
```




```
ggplot(auto, aes(y = mpg, group = cylinders, fill = cylinders)) +
  geom_boxplot()
```

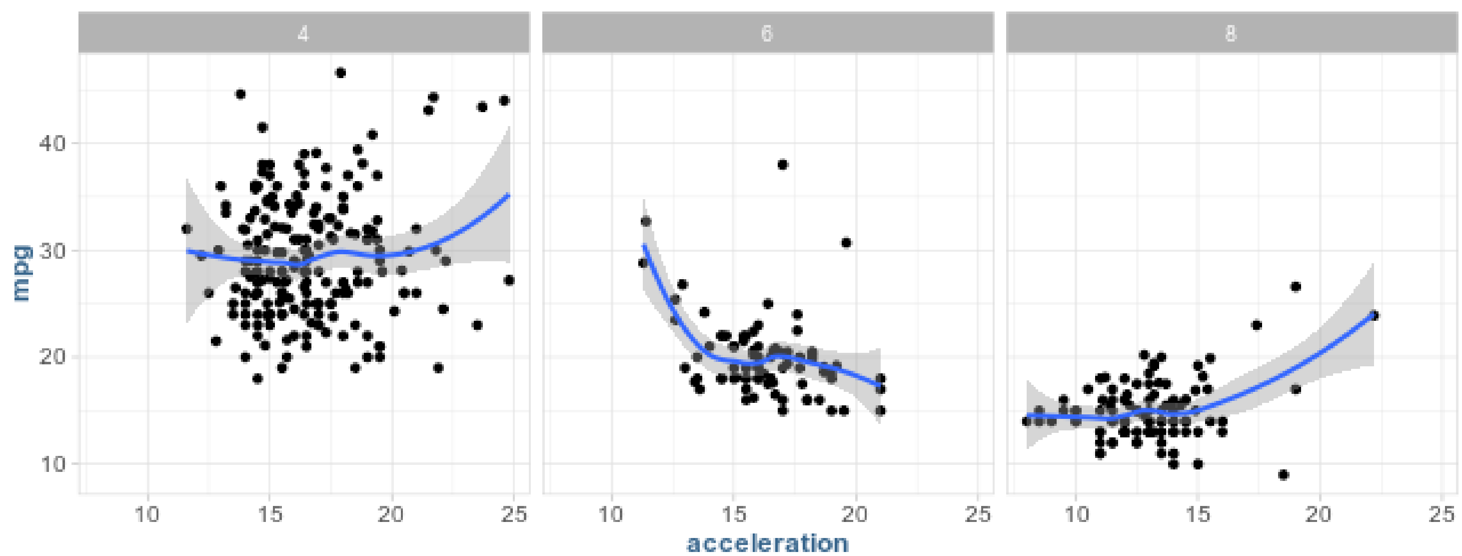



```
ggplot(auto, aes(acceleration, mpg, col = cylinders)) +
  geom_point()
```



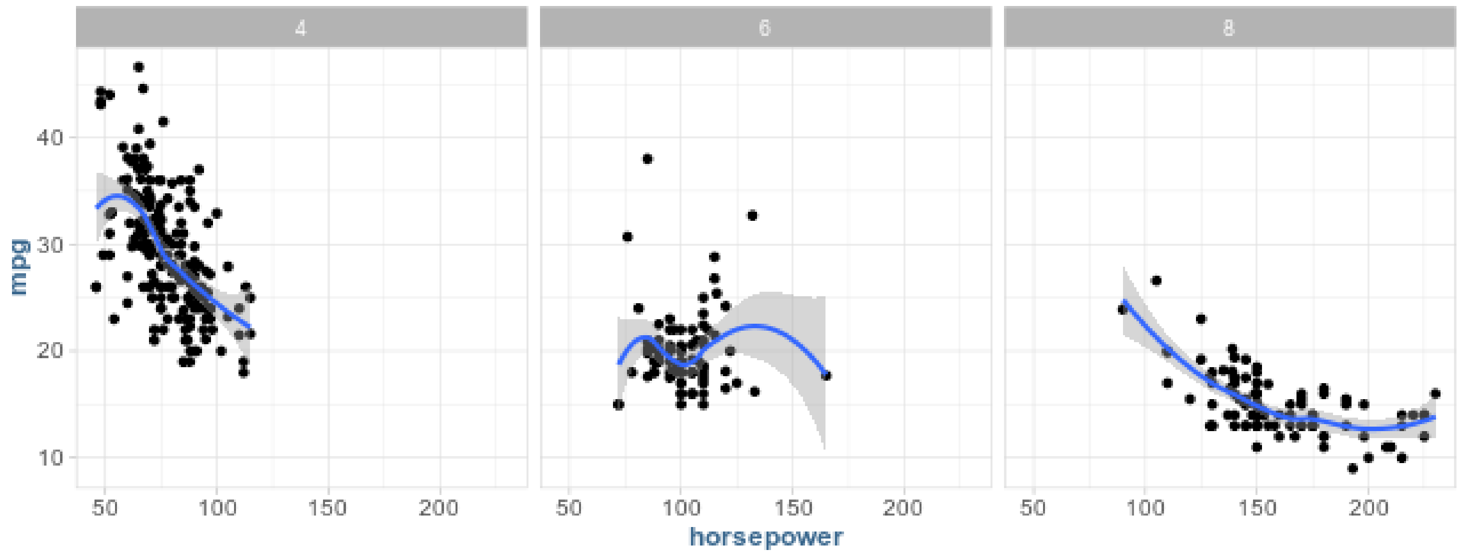

```
ggplot(auto[cylinders %in% c(4, 6, 8)], aes(acceleration, mpg)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~cylinders)
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

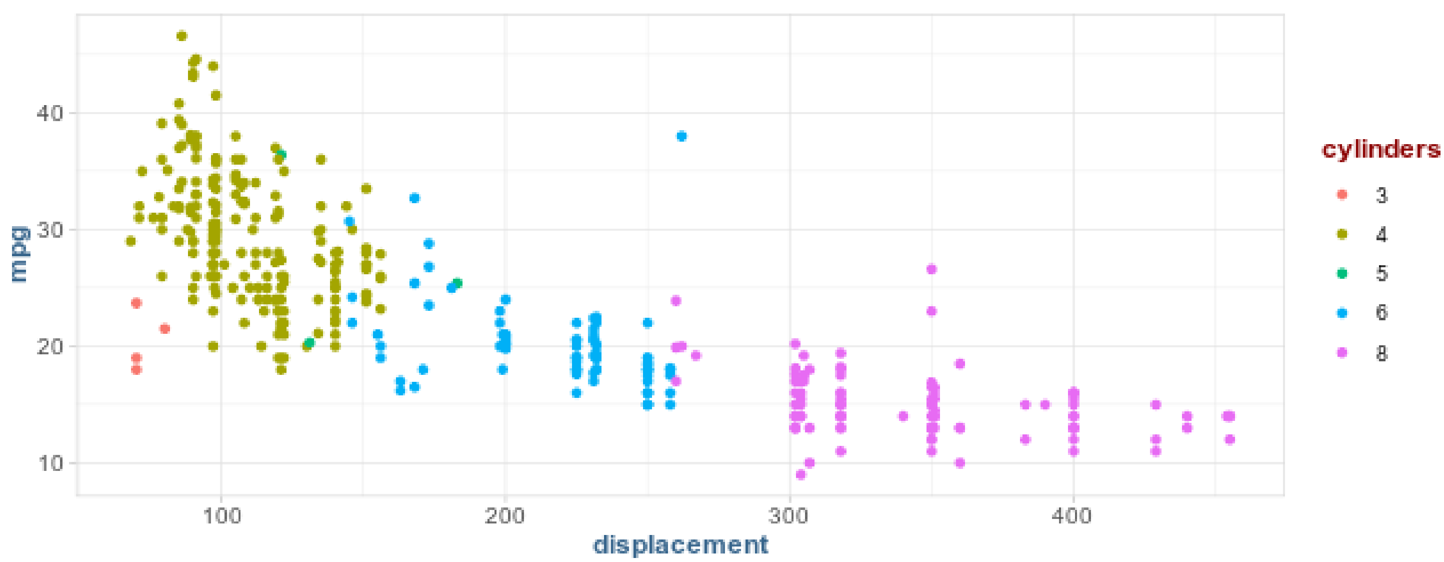



```
ggplot(auto[cylinders %in% c(4, 6, 8)], aes(horsepower, mpg)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~cylinders)
```

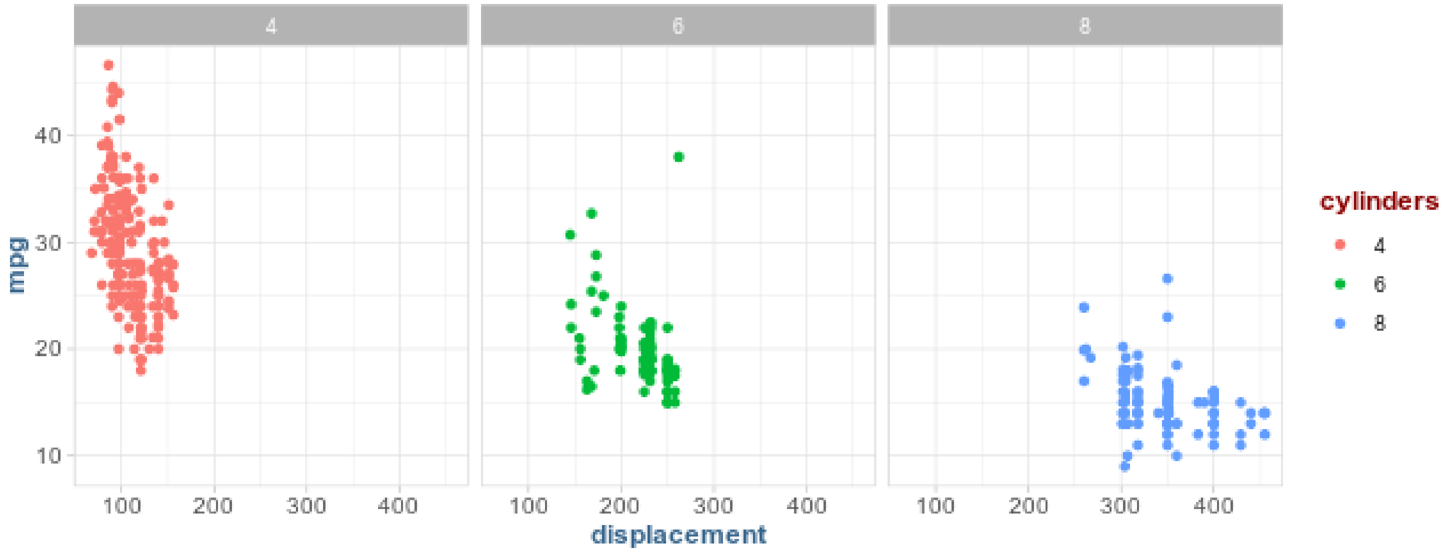
```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(auto, aes(displacement, mpg, col = cylinders)) +
   geom_point()
```




```
ggplot(auto[cylinders %in% c(4, 6, 8)], aes(displacement, mpg, col = cylinders)) +
   geom_point() +
   facet_wrap(~cylinders)
```




```r
# suggests quadratic and quintic are better than linear

with(auto, {
   fit1 <- lm(mpg ~ horsepower + cylinders)
   fit2 <- lm(mpg ~ poly(horsepower, 2) + cylinders)
   fit3 <- lm(mpg ~ poly(horsepower, 3) + cylinders)
   fit4 <- lm(mpg ~ poly(horsepower, 4) + cylinders)
   fit5 <- lm(mpg ~ poly(horsepower, 5) + cylinders)

   anova(fit1, fit2, fit3, fit4, fit5)
})
```

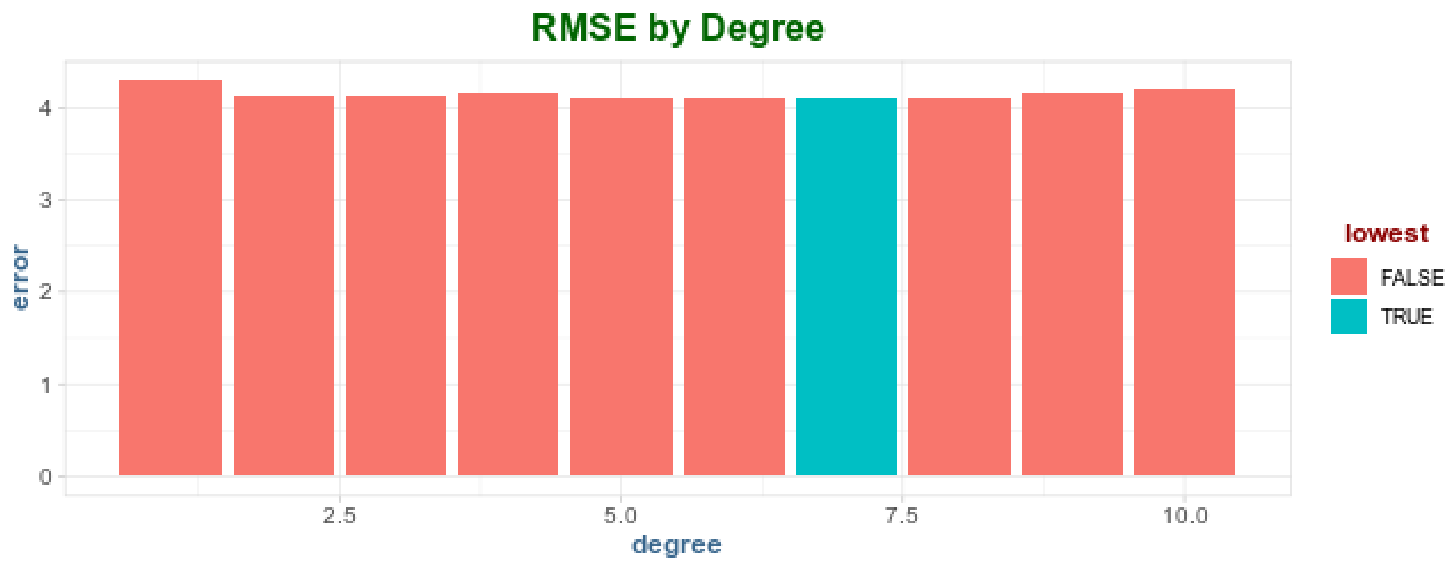
```
Analysis of Variance Table

Model 1: mpg ~ horsepower + cylinders
Model 2: mpg ~ poly(horsepower, 2) + cylinders
Model 3: mpg ~ poly(horsepower, 3) + cylinders
Model 4: mpg ~ poly(horsepower, 4) + cylinders
Model 5: mpg ~ poly(horsepower, 5) + cylinders
  Res.Df    RSS Df Sum of Sq       F    Pr(>F)
1    386 7036.7
2    385 6315.5  1    721.12 44.8041 7.787e-11 ***
3    384 6279.0  1     36.53  2.2698  0.132746
4    383 6265.4  1     13.63  0.8470  0.357982
5    382 6148.3  1    117.12  7.2768  0.007295 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
n <- nrow(auto)
```

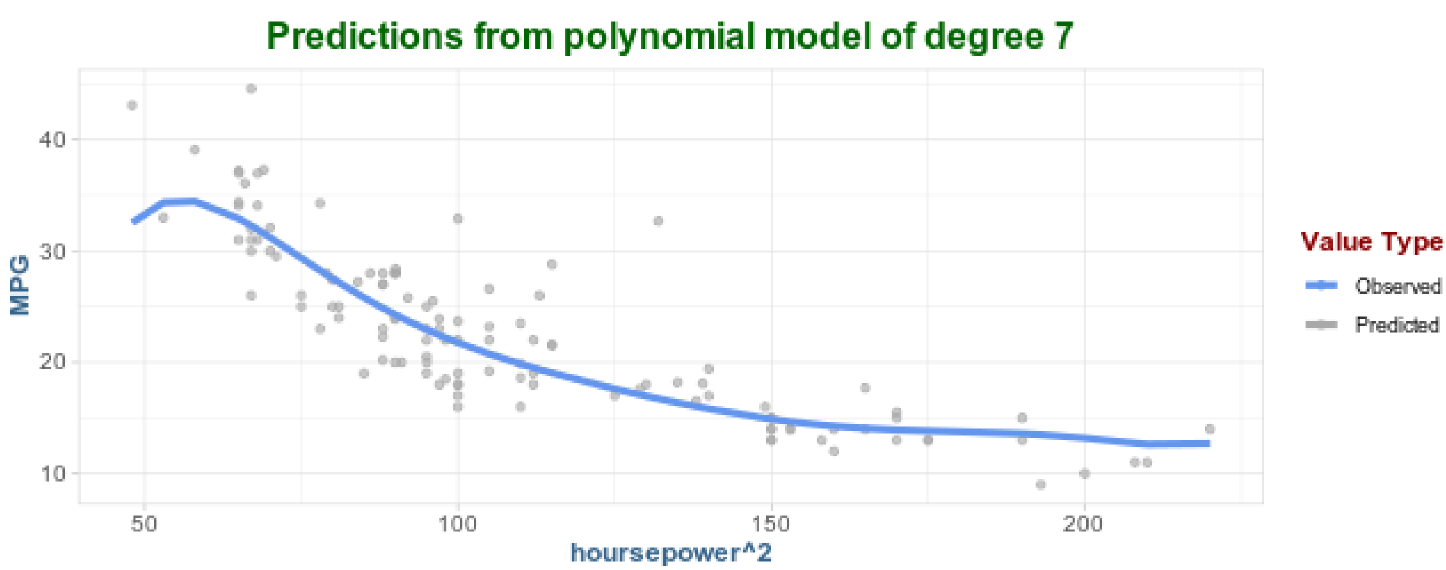
```
index <- sample(n, n * .7, replace = F)

train <- auto[index]
test <- auto[!index]

n; nrow(train); nrow(test)
```

```
[1] 392

[1] 274

[1] 118
```

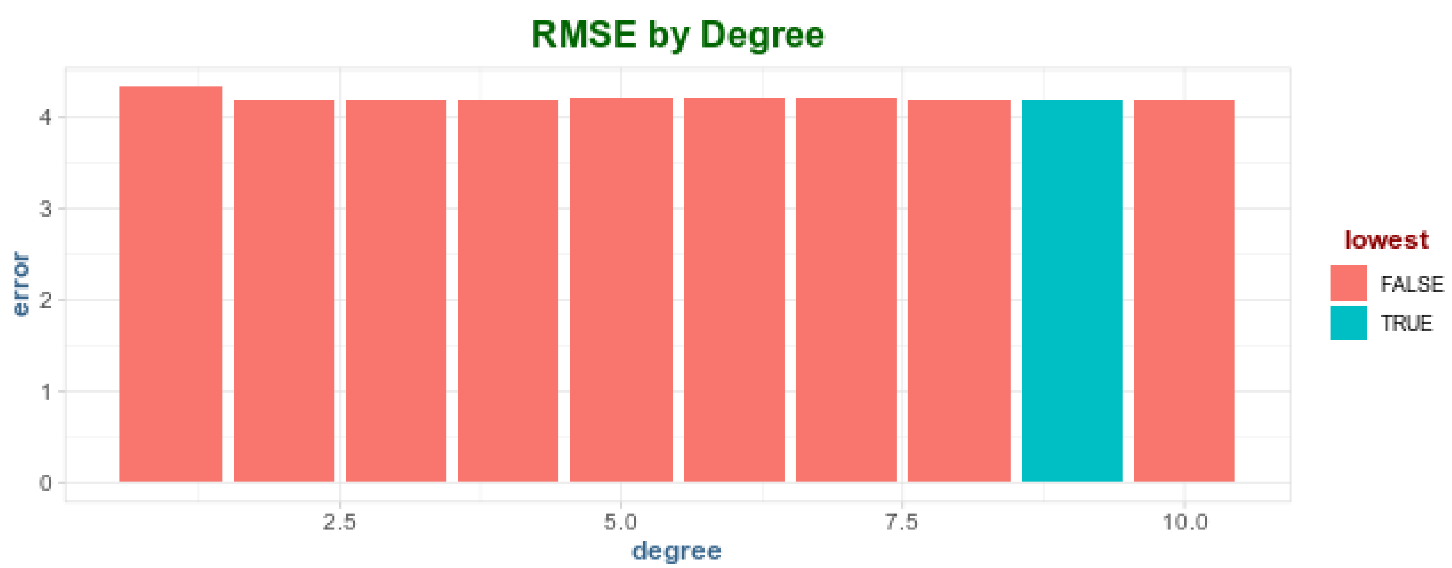
```
degree <- 10; folds = 10
cv.errors <- numeric(degree)

fold.size <- nrow(train) / folds

for(deg in 1:degree)
{
   # 10 fold cv
   errors <- numeric(folds)
   for(fold in 1:folds)
   {
      holdout <- seq((fold - 1) * fold.size, fold * fold.size)

      cv.train <- train[!holdout]
      cv.test <- train[holdout]

      fit <- lm(mpg ~ poly(horsepower, deg) + cylinders, data = cv.train)

      pred <- predict(fit, newdata = cv.test, type = "response")

      errors[fold] <- sqrt(mean((cv.test$mpg - pred)^2))
   }
   cv.errors[deg] <- mean(errors)
}

lowest.error <- which.min(cv.errors)

cv.results <- data.table(degree = 1:degree, error = cv.errors)[, lowest := degree == lowest.err

ggplot(cv.results, aes(degree, error, fill = lowest)) +
   geom_bar(stat = "identity") +
   labs(title = "RMSE by Degree")
```



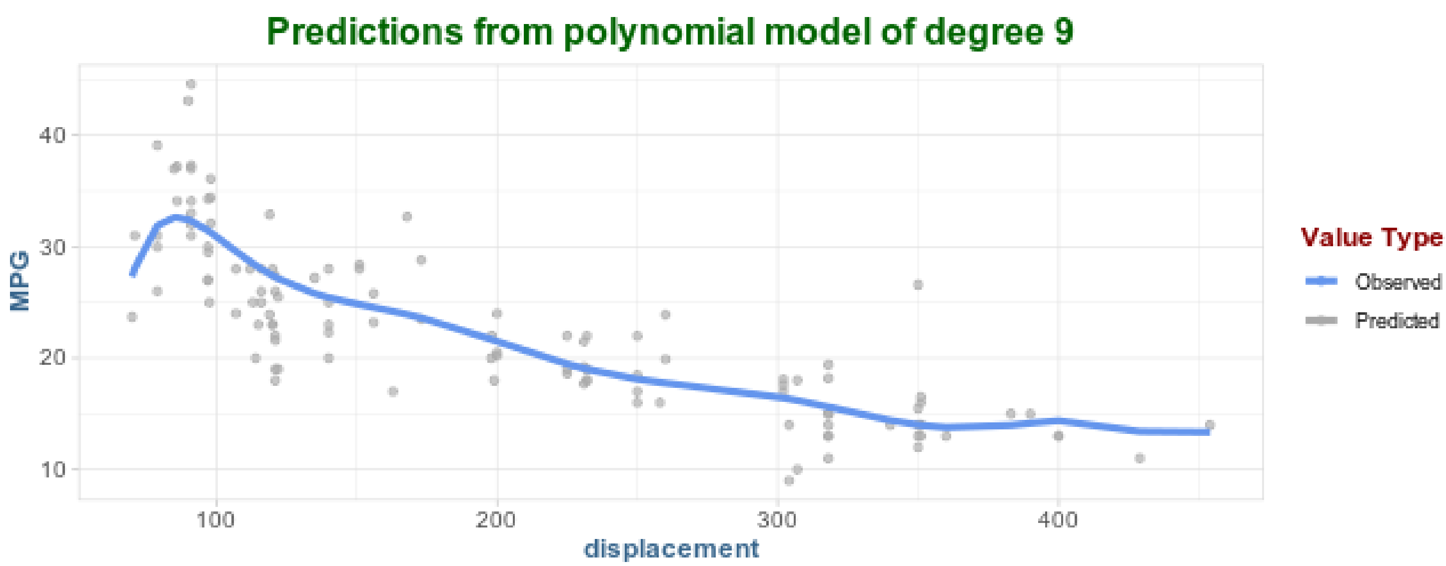

```
model <- lm(mpg ~ poly(horsepower, lowest.error), data = train)

test %>%
    mutate(predictions = predict(model, test)) %>%
    ggplot(aes(horsepower, mpg, col = 'darkgrey')) +
    geom_point(alpha = .65) +
    geom_line(aes(horsepower, predictions, col = 'cornflowerblue'), size = 1.5) +
    scale_color_manual(name = 'Value Type',
                        labels = c('Observed', 'Predicted'),
                        values = c('cornflowerblue', 'darkgrey' )) +
    labs(x = 'hoursepower^2', y = 'MPG',
         title = paste0('Predictions from polynomial model of degree ', lowest.error))
```



```
for(deg in 1:degree)
{
  # 10 fold cv
  errors <- numeric(folds)
  for(fold in 1:folds)
  {
    holdout <- seq((fold - 1) * fold.size, fold * fold.size)

    cv.train <- train[!holdout]
    cv.test <- train[holdout]

    fit <- lm(mpg ~ poly(displacement, deg) + cylinders, data = cv.train)

    pred <- predict(fit, newdata = cv.test, type = "response")

    errors[fold] <- sqrt(mean((cv.test$mpg - pred)^2))
  }
  cv.errors[deg] <- mean(errors)
}

lowest.error <- which.min(cv.errors)

cv.results <- data.table(degree = 1:degree, error = cv.errors)[, lowest := degree == lowest.err

ggplot(cv.results, aes(degree, error, fill = lowest)) +
  geom_bar(stat = "identity") +
  labs(title = "RMSE by Degree")
```



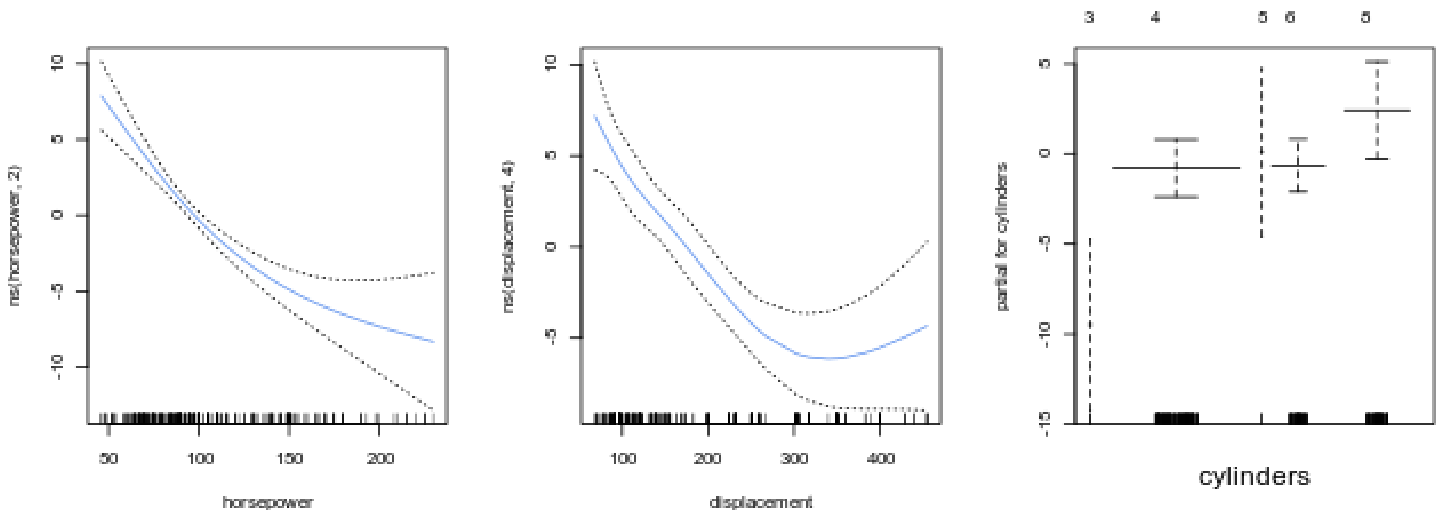
```
model <- lm(mpg ~ poly(displacement, lowest.error), data = train)

test %>%
    mutate(predictions = predict(model, test)) %>%
    ggplot(aes(displacement, mpg, col = 'darkgrey')) +
    geom_point(alpha = .65) +
    geom_line(aes(displacement, predictions, col = 'cornflowerblue'), size = 1.5) +
    scale_color_manual(name = 'Value Type',
                       labels = c('Observed', 'Predicted'),
                       values = c('cornflowerblue', 'darkgrey' )) +
    labs(x = 'displacement', y = 'MPG',
        title = paste0('Predictions from polynomial model of degree ', lowest.error))
```




```
polyfit <- gam(mpg ~ ns(horsepower, 2) + ns(displacement, 4) + cylinders, data = auto)
```

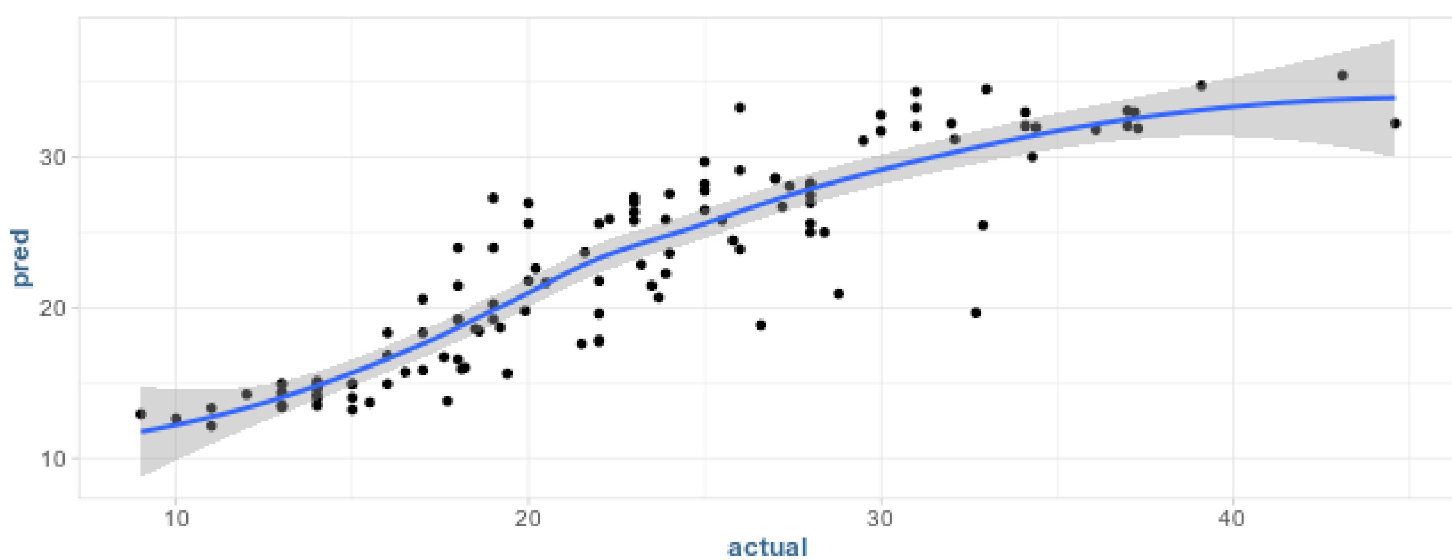
```
Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
ignored
```

```
par(mfrow = c(1, 3))
plot(polyfit, se = T, col = "cornflowerblue")
```




```
pred <- data.table(actual = test$mpg, pred = predict(polyfit, test, type = "response"))

ggplot(pred, aes(actual, pred)) +
   geom_point() +
   geom_smooth()
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Boston

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

```r
boston <- data.table(Boston)
```

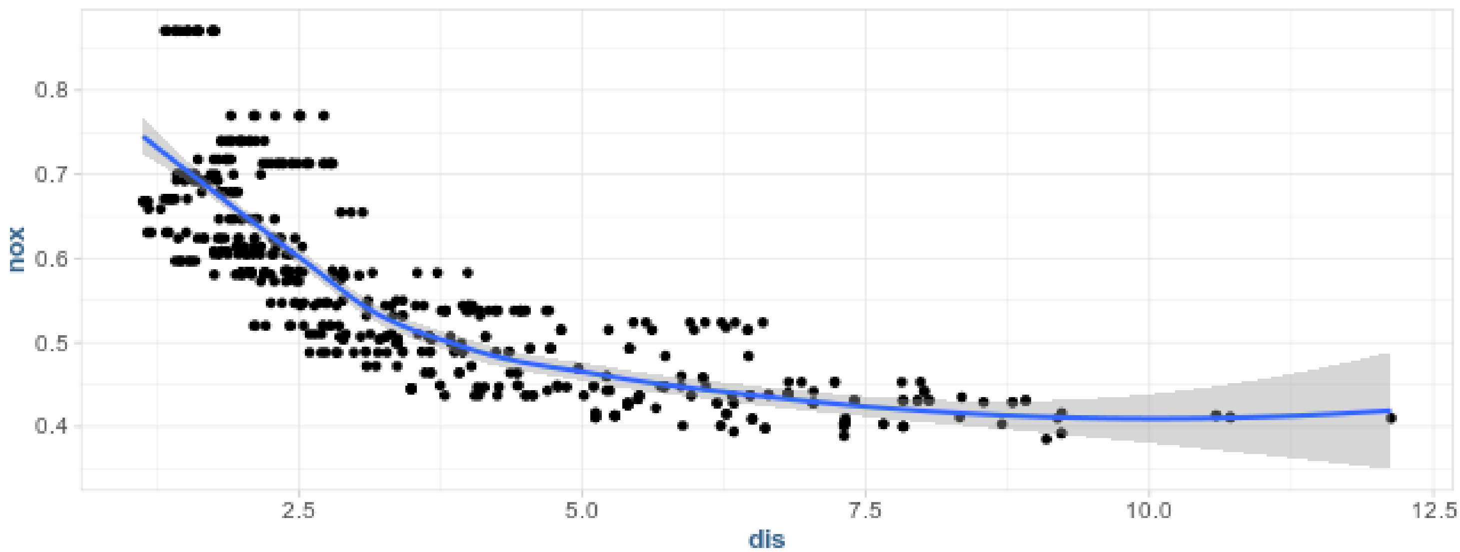
a.) Use the poly() function to fit a cubic polynomial regression to predict nox using dis. Report the regression output, and plot the resulting data using polynomial fits.

```r
fit1 <- lm(nox ~ poly(dis, 3), data = boston)
summary(fit)
```

```

Call:
lm(formula = mpg ~ poly(displacement, deg) + cylinders, data = cv.train)

Residuals:
     Min       1Q   Median       3Q      Max 
-11.0564  -2.7236  -0.5241   1.8734  19.4901 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)                 13.523      3.579   3.779 0.000201 ***
poly(displacement, deg)1   -65.515     22.024  -2.975 0.003245 ** 
poly(displacement, deg)2    17.758      7.846   2.263 0.024540 *  
poly(displacement, deg)3    -9.295      7.468  -1.245 0.214533    
poly(displacement, deg)4     9.029      6.062   1.490 0.137711    
poly(displacement, deg)5    -3.525      6.206  -0.568 0.570539    
poly(displacement, deg)6    -1.657      5.025  -0.330 0.741882    
poly(displacement, deg)7     4.332      5.242   0.827 0.409341    
poly(displacement, deg)8    -8.134      4.845  -1.679 0.094543 .  
poly(displacement, deg)9     7.837      4.631   1.692 0.091966 .  
poly(displacement, deg)10   -4.071      4.661  -0.873 0.383300    
cylinders4                  12.605      3.393   3.716 0.000254 ***
cylinders5                   9.209      4.649   1.981 0.048816 *  
cylinders6                   8.320      3.953   2.105 0.036401 *  
cylinders8                   6.954      4.920   1.413 0.158896    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.222 on 231 degrees of freedom
Multiple R-squared:  0.7323,    Adjusted R-squared:  0.7161 
F-statistic: 45.14 on 14 and 231 DF,  p-value: < 2.2e-16
```

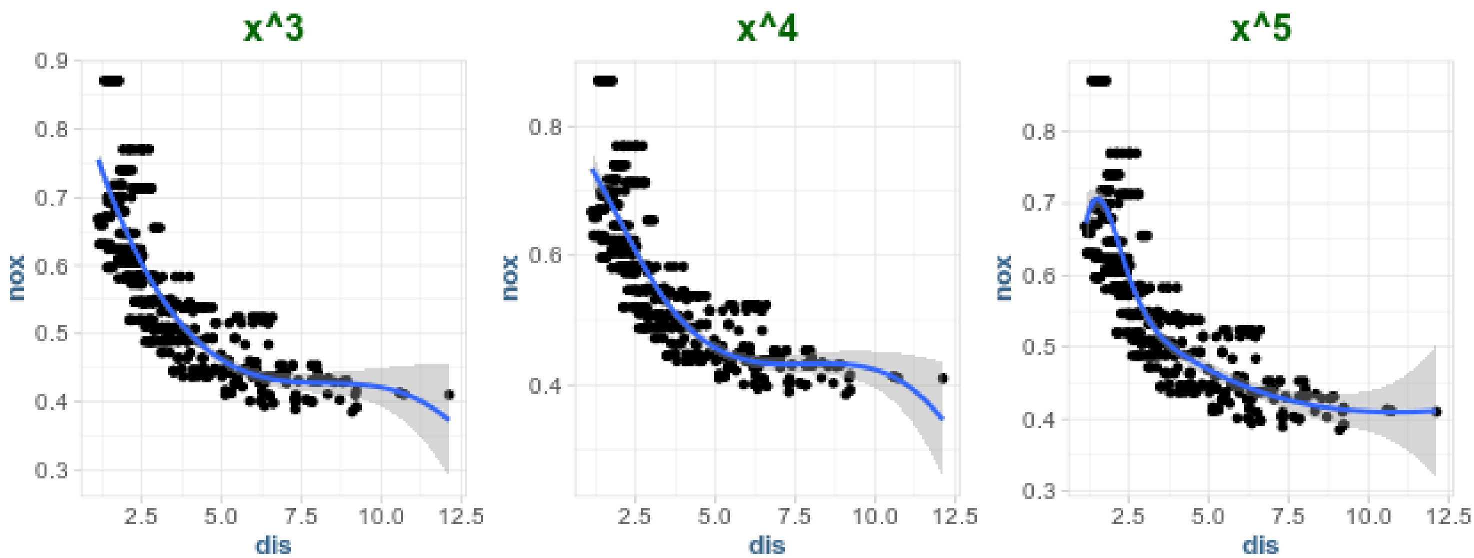
```r
ggplot(boston, aes(dis, nox)) +
   geom_point() +
   geom_smooth()
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

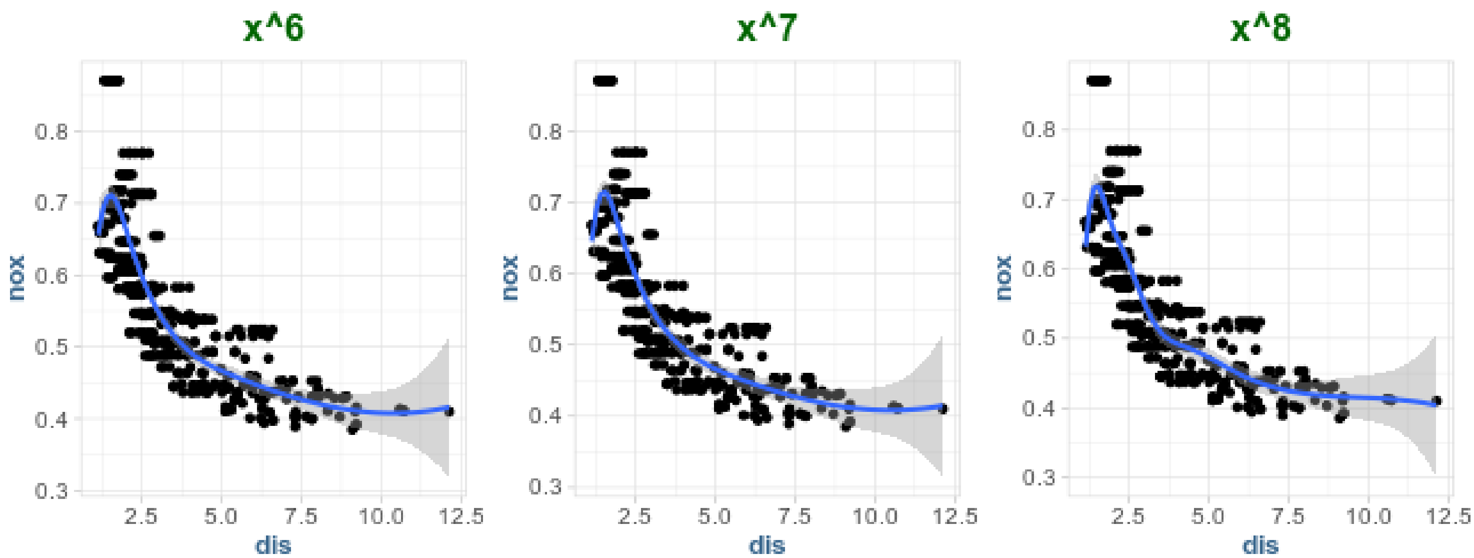


Plot the polynomial fits for a range of different polynomial degrees (say, 1 to 10), and report the associated RSS.

```
p1 <- ggplot(boston, aes(dis, nox)) +
   geom_point() +
   geom_smooth(method = lm, formula = y ~ splines::bs(x, 3)) +
   labs(title = "x^3")

p2 <- ggplot(boston, aes(dis, nox)) +
   geom_point() +
   geom_smooth(method = lm, formula = y ~ splines::bs(x, 4)) +
   labs(title = "x^4")

p3 <- ggplot(boston, aes(dis, nox)) +
   geom_point() +
   geom_smooth(method = lm, formula = y ~ splines::bs(x, 5)) +
   labs(title = "x^5")

grid.arrange(p1, p2, p3, nrow = 1)
```




```r
p4 <- ggplot(boston, aes(dis, nox)) +
   geom_point() +
   geom_smooth(method = lm, formula = y ~ splines::bs(x, 6)) +
   labs(title = "x^6")

p5 <- ggplot(boston, aes(dis, nox)) +
   geom_point() +
   geom_smooth(method = lm, formula = y ~ splines::bs(x, 7)) +
   labs(title = "x^7")

p6 <- ggplot(boston, aes(dis, nox)) +
   geom_point() +
   geom_smooth(method = lm, formula = y ~ splines::bs(x, 8)) +
   labs(title = "x^8")

grid.arrange(p4, p5, p6, nrow = 1)
```



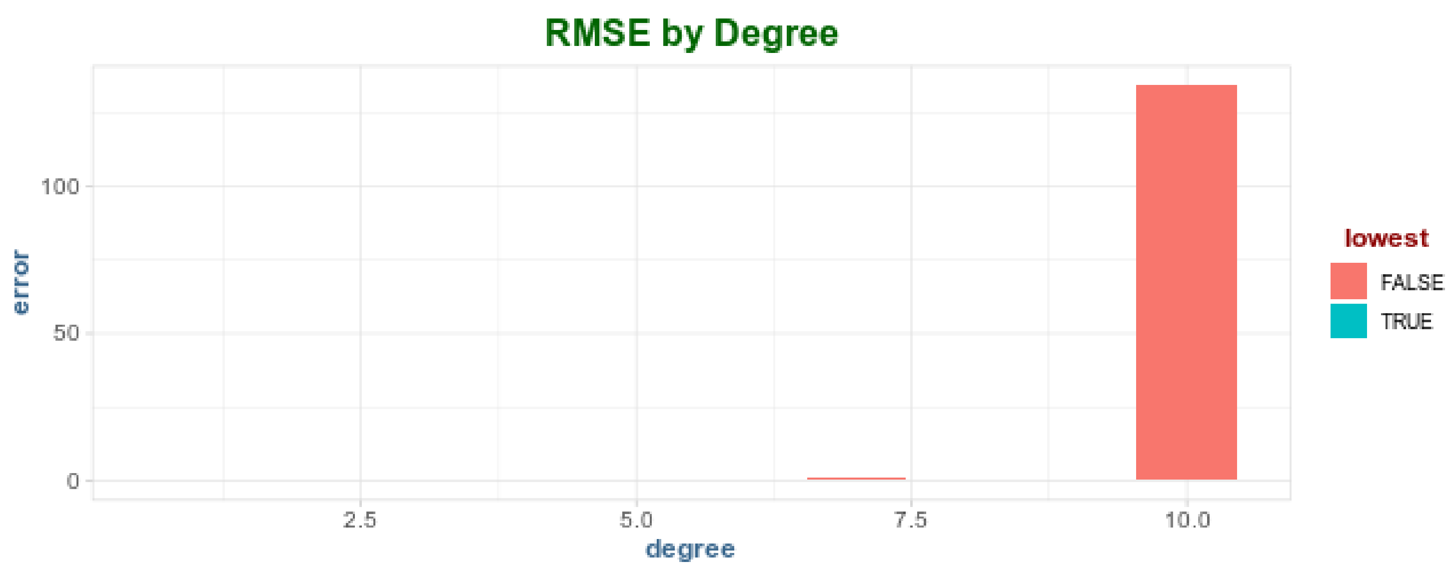
Perform cross-validation or another approach to select the optimal degree for the polynomial, and explain your results.

```
n <- nrow(boston)

index <- sample(n, n * .7, replace = F)

train <- boston[index]
test <- boston[!index]

n; nrow(train); nrow(test)
```

```
[1] 506

[1] 354

[1] 152
```

```
degree <- 10; folds = 10
cv.errors <- numeric(degree)

fold.size <- nrow(train) / folds

for(deg in 1:degree)
{
   # 10 fold cv
   errors <- numeric(folds)
   for(fold in 1:folds)
   {
      holdout <- seq((fold - 1) * fold.size, fold * fold.size)

      cv.train <- train[!holdout]
```

```
      cv.test <- train[holdout]

      fit <- lm(nox ~ poly(dis, deg), data = cv.train)

      pred <- predict(fit, newdata = cv.test, type = "response")

      errors[fold] <- sum( (cv.test$nox - pred)^2 ) # RSS
   }
   cv.errors[deg] <- mean(errors)
}

lowest.error <- which.min(cv.errors)

cv.results <- data.table(degree = 1:degree, error = cv.errors)[, lowest := degree == lowest.err

ggplot(cv.results, aes(degree, error, fill = lowest)) +
   geom_bar(stat = "identity") +
   labs(title = "RMSE by Degree")
```



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

```
n <- nrow(boston)

index <- sample(n, n * .7, replace = F)

train <- boston[index]
test <- boston[!index]
```

```r
n; nrow(train); nrow(test)
```

```
[1] 506

[1] 354

[1] 152
```

```r
range(boston$dis)
```

```
[1]  1.1296 12.1265
```

```r
knots <- seq(1, 15, 1); folds = 10
cv.errors <- numeric(length(knots))

fold.size <- nrow(train) / folds

index <- 1

for(index in 1:length(knots))
{
   # 10 fold cv
   errors <- numeric(folds)
   for(fold in 1:folds)
   {
      holdout <- seq((fold - 1) * fold.size, fold * fold.size)

      cv.train <- train[!holdout]
      cv.test <- train[holdout]

      fit <- lm(nox ~ bs(dis, knots[index]), data = cv.train)

      pred <- predict(fit, newdata = cv.test, type = "response")

      errors[fold] <- sqrt(mean( (cv.test$nox - pred)^2 )) # RMSE
   }
   cv.errors[index] <- mean(errors)
}
```

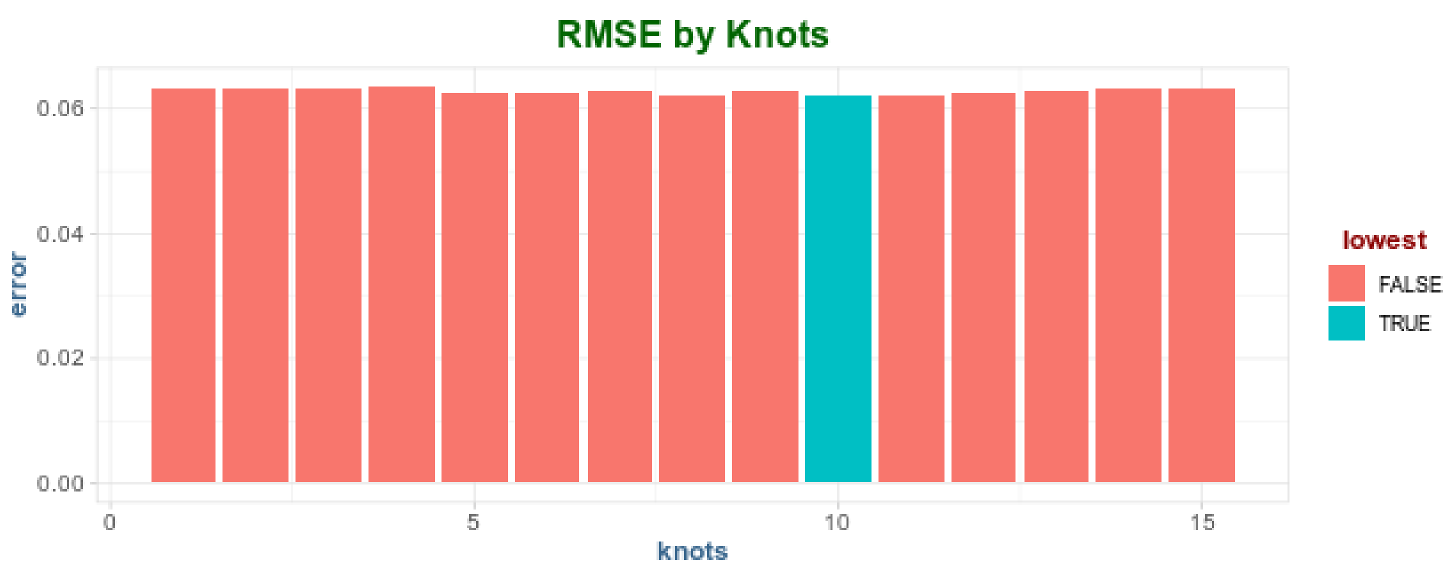
```
Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.1742, :
some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3
Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.137, :
some 'x' values beyond boundary knots may cause ill-conditioned bases
Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3
Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.1742, :
some 'x' values beyond boundary knots may cause ill-conditioned bases
Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3
Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.137, :
some 'x' values beyond boundary knots may cause ill-conditioned bases
Warning in bs(dis, knots[index]): 'df' was too small; have used 3

Warning in bs(dis, knots[index]): 'df' was too small; have used 3
Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.1742, :
some 'x' values beyond boundary knots may cause ill-conditioned bases
Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.137, :
some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
Warning in bs(dis, degree = 3L, knots = c(`50%` = 3.2157), Boundary.knots =
c(1.1742, : some 'x' values beyond boundary knots may cause ill-conditioned
bases

Warning in bs(dis, degree = 3L, knots = c(`50%` = 3.2439), Boundary.knots =
c(1.137, : some 'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`33.33333%` = 2.38933333333333, : some
'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`33.33333%` = 2.36326666666667, : some
'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`25%` = 2.10225, `50%` = 3.2157, :
some 'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`25%` = 2.100175, `50%` = 3.2439, :
some 'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`20%` = 1.94712, `40%` = 2.54414, :
some 'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`20%` = 1.96742, `40%` = 2.54414, :
some 'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`16.66667%` = 1.81911666666667, : some
'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`16.66667%` = 1.85738333333333, : some
'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`14.28571%` = 1.77345714285714, : some
'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`14.28571%` = 1.78912857142857, : some
'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`12.5%` = 1.738975, `25%` = 2.10225, :
some 'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`12.5%` = 1.75315, `25%` = 2.100175, :
some 'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`11.11111%` = 1.66862222222222, : some
'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`11.11111%` = 1.66862222222222, : some
'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`10%` = 1.6202, `20%` = 1.94712, :
some 'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`10%` = 1.61281, `20%` = 1.96742, :
```

```
some 'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`9.090909%` = 1.59121818181818, : some
'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`9.090909%` = 1.59121818181818, : some
'x' values beyond boundary knots may cause ill-conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`8.333333%` = 1.58415, `16.66667%`
= 1.81911666666667, : some 'x' values beyond boundary knots may cause ill-
conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`8.333333%` = 1.58415, `16.66667%`
= 1.85738333333333, : some 'x' values beyond boundary knots may cause ill-
conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`7.692308%` = 1.5411, `15.38462%`
= 1.79335384615385, : some 'x' values beyond boundary knots may cause ill-
conditioned bases

Warning in bs(dis, degree = 3L, knots = c(`7.692308%` = 1.5411, `15.38462%`
= 1.81286153846154, : some 'x' values beyond boundary knots may cause ill-
conditioned bases
```

```r
lowest.error <- knots[which.min(cv.errors)]

cv.results <- data.table(knots = knots, error = cv.errors)[, lowest := knots == lowest.error]

ggplot(cv.results, aes(knots, error, fill = lowest)) +
  geom_bar(stat = "identity") +
  labs(title = "RMSE by Knots")
```



Use CV to select the best degrees of freedom for a regression spline.

```
n <- nrow(boston)

index <- sample(n, n * .7, replace = F)

train <- boston[index]
test <- boston[!index]

n; nrow(train); nrow(test)
```

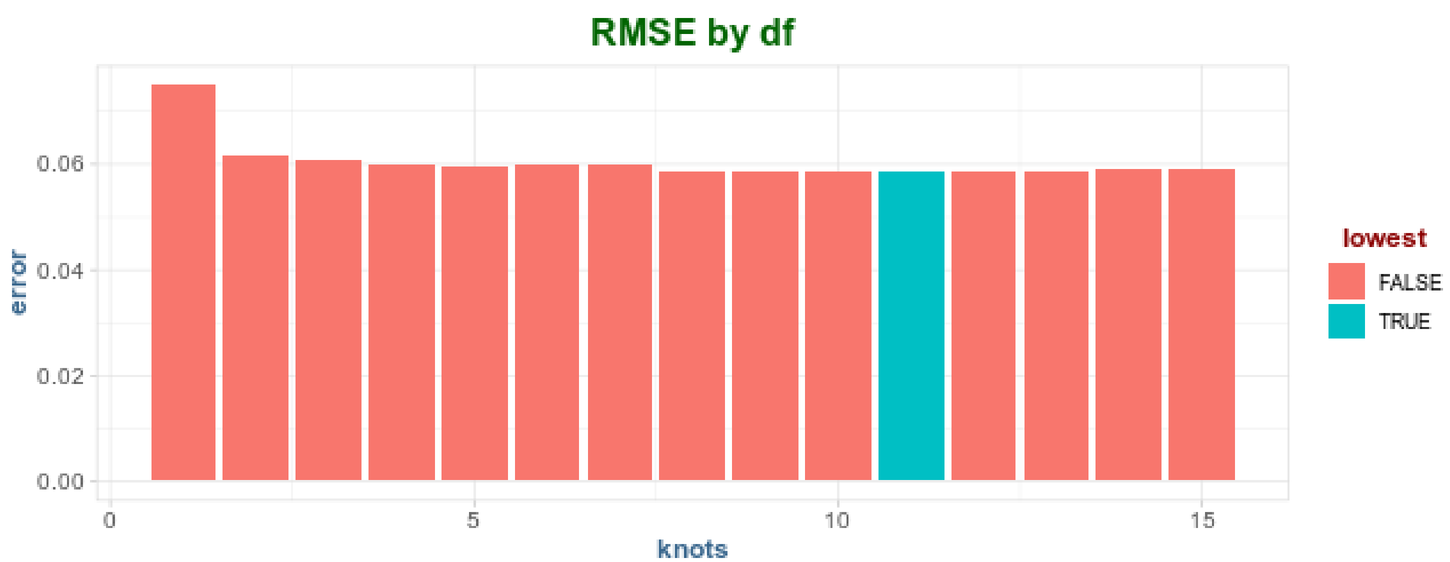
```
[1] 506
[1] 354
[1] 152
```

```
range(boston$dis)
```

```
[1]  1.1296 12.1265
```

```
df <- seq(1, 15, 1); folds = 10
cv.errors <- numeric(length(knots))

fold.size <- nrow(train) / folds

index <- 1

for(index in 1:length(df))
{
   # 10 fold cv
   errors <- numeric(folds)
   for(fold in 1:folds)
   {
      holdout <- seq((fold - 1) * fold.size, fold * fold.size)

      cv.train <- train[!holdout]
      cv.test <- train[holdout]

      fit <- lm(nox ~ ns(dis, df[index]), data = cv.train)

      pred <- predict(fit, newdata = cv.test, type = "response")

      errors[fold] <- sqrt(mean( (cv.test$nox - pred)^2 )) # RMSE
   }
   cv.errors[index] <- mean(errors)
}

lowest.error <- knots[which.min(cv.errors)]
```

```r
cv.results <- data.table(df = df, error = cv.errors)[, lowest := df == lowest.error]

ggplot(cv.results, aes(knots, error, fill = lowest)) +
  geom_bar(stat = "identity") +
  labs(title = "RMSE by df")
```



College

This question uses the college data set.

```r
college <- data.table(ISLR::College)
```

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

```r
n <- nrow(college)

index <- sample(n, n * .7)

train <- college[index]
test <- college[!index]

null <- lm(Outstate ~ 1, data = train)
full <- formula(lm(Outstate ~ ., data = train))

forward.fit <- step(null, direction = 'forward', scope = full)
```

```
Start:  AIC=9028.78
```

```
Outstate ~ 1

              Df  Sum of Sq        RSS    AIC
+ Expend       1 4213155176 4791586422 8688.2
+ Room.Board   1 4078005696 4926735902 8703.3
+ Top10perc    1 2980184288 6024557310 8812.5
+ Grad.Rate    1 2908873529 6095868070 8818.9
+ Private      1 2809833131 6194908467 8827.7
+ S.F.Ratio    1 2809827156 6194914442 8827.7
+ perc.alumni  1 2714855146 6289886452 8835.9
+ Top25perc    1 2219630256 6785111342 8877.1
+ Terminal     1 1481040616 7523700982 8933.2
+ PhD          1 1437674069 7567067529 8936.3
+ Personal     1  842465949 8162275649 8977.4
+ P.Undergrad  1  571583570 8433158028 8995.2
+ F.Undergrad  1  414184055 8590557543 9005.2
+ Enroll       1  221245502 8783496096 9017.3
<none>                      9004741598 9028.8
+ Apps         1   22372393 8982369205 9029.4
+ Books        1   16677116 8988064482 9029.8
+ Accept       1    8047581 8996694017 9030.3

Step:  AIC=8688.21
Outstate ~ Expend

              Df  Sum of Sq        RSS    AIC
+ Private      1 1437049207 3354537215 8496.6
+ Room.Board   1 1270464333 3521122089 8522.9
+ Grad.Rate    1 1001548153 3790038269 8562.9
+ perc.alumni  1  712294984 4079291438 8602.8
+ F.Undergrad  1  523397243 4268189179 8627.4
+ Personal     1  502061156 4289525267 8630.1
+ Enroll       1  402616899 4388969523 8642.6
+ P.Undergrad  1  370522364 4421064058 8646.5
+ S.F.Ratio    1  317385803 4474200619 8653.0
+ Top10perc    1  204662300 4586924122 8666.5
+ Top25perc    1  203550148 4588036274 8666.6
+ Apps         1  179611877 4611974545 8669.5
+ Accept       1  126659701 4664926722 8675.7
+ Terminal     1  104760428 4686825994 8678.2
+ PhD          1   65044267 4726542155 8682.8
<none>                      4791586422 8688.2
+ Books        1    2080852 4789505571 8690.0

Step:  AIC=8496.6
```

```
Outstate ~ Expend + Private

              Df Sum of Sq        RSS    AIC
+ Room.Board   1 699029618 2655507598 8371.7
+ Grad.Rate    1 526400821 2828136395 8405.9
+ Terminal     1 444843322 2909693893 8421.4
+ PhD          1 423277798 2931259418 8425.4
+ Top25perc    1 255101556 3099435660 8455.7
+ perc.alumni  1 216317857 3138219358 8462.4
+ Top10perc    1 202973174 3151564042 8464.7
+ Personal     1 148165333 3206371882 8474.1
+ Accept       1 105908920 3248628296 8481.2
+ Apps         1  59107844 3295429372 8489.0
+ S.F.Ratio    1  17458186 3337079029 8495.8
+ Enroll       1  15590834 3338946382 8496.1
<none>                     3354537215 8496.6
+ F.Undergrad  1   4333520 3350203696 8497.9
+ P.Undergrad  1   4212212 3350325004 8497.9
+ Books        1    545096 3353992119 8498.5

Step:  AIC=8371.72
Outstate ~ Expend + Private + Room.Board

              Df Sum of Sq        RSS    AIC
+ Grad.Rate    1 277779194 2377728403 8313.7
+ perc.alumni  1 220841936 2434665662 8326.6
+ PhD          1 192545611 2462961986 8332.8
+ Terminal     1 189382346 2466125251 8333.5
+ Top25perc    1 145620051 2509887547 8343.1
+ Top10perc    1 129736201 2525771397 8346.5
+ Personal     1  83566619 2571940978 8356.4
+ S.F.Ratio    1  18154720 2637352878 8370.0
+ P.Undergrad  1  17673648 2637833950 8370.1
+ Accept       1  16217617 2639289981 8370.4
<none>                     2655507598 8371.7
+ Books        1   7803895 2647703702 8372.1
+ Apps         1   1804222 2653703376 8373.3
+ F.Undergrad  1   1503743 2654003854 8373.4
+ Enroll       1    557747 2654949850 8373.6

Step:  AIC=8313.72
Outstate ~ Expend + Private + Room.Board + Grad.Rate

              Df Sum of Sq        RSS    AIC
+ Terminal     1 132988820 2244739584 8284.5
```

```
+ PhD          1 127365178 2250363226 8285.8
+ perc.alumni  1  99615903 2278112501 8292.5
+ Top25perc    1  42992467 2334735936 8305.8
+ Personal     1  37015977 2340712426 8307.2
+ Top10perc    1  32700229 2345028174 8308.2
+ S.F.Ratio    1  16121368 2361607036 8312.0
<none>                     2377728403 8313.7
+ F.Undergrad  1   7522092 2370206311 8314.0
+ Books        1   5401533 2372326871 8314.5
+ Apps         1   4434091 2373294312 8314.7
+ Enroll       1   1965603 2375762800 8315.3
+ Accept       1   1824522 2375903882 8315.3
+ P.Undergrad  1   1552795 2376175608 8315.4

Step:  AIC=8284.47
Outstate ~ Expend + Private + Room.Board + Grad.Rate + Terminal

              Df Sum of Sq        RSS    AIC
+ perc.alumni  1  61752092 2182987492 8271.3
+ Personal     1  29952427 2214787156 8279.2
+ F.Undergrad  1  21895847 2222843736 8281.1
+ S.F.Ratio    1  13558761 2231180823 8283.2
+ PhD          1  13307614 2231431970 8283.2
+ Books        1  10581501 2234158082 8283.9
+ Top25perc    1   9742517 2234997067 8284.1
+ Enroll       1   9554156 2235185428 8284.2
+ Apps         1   9297627 2235441957 8284.2
<none>                     2244739584 8284.5
+ Top10perc    1   8165769 2236573814 8284.5
+ P.Undergrad  1   4106236 2240633348 8285.5
+ Accept       1     13078 2244726506 8286.5

Step:  AIC=8271.32
Outstate ~ Expend + Private + Room.Board + Grad.Rate + Terminal +
    perc.alumni

              Df Sum of Sq        RSS    AIC
+ Personal     1  19631683 2163355809 8268.4
+ F.Undergrad  1  15184466 2167803025 8269.5
+ PhD          1  11289351 2171698141 8270.5
+ S.F.Ratio    1   9549610 2173437882 8270.9
+ Books        1   8356889 2174630603 8271.2
<none>                     2182987492 8271.3
+ Enroll       1   6014071 2176973421 8271.8
+ Top25perc    1   3880074 2179107417 8272.4
```

```
+ Apps          1   3805967 2179181524 8272.4
+ Top10perc     1   2537241 2180450250 8272.7
+ P.Undergrad   1   2236820 2180750672 8272.8
+ Accept        1   1070498 2181916994 8273.1

Step:  AIC=8268.41
Outstate ~ Expend + Private + Room.Board + Grad.Rate + Terminal + 
    perc.alumni + Personal

              Df Sum of Sq        RSS    AIC
+ PhD          1  11503314 2151852495 8267.5
+ S.F.Ratio    1  10242520 2153113288 8267.8
+ F.Undergrad  1   9023273 2154332535 8268.1
<none>                     2163355809 8268.4
+ Top25perc    1   5607237 2157748572 8269.0
+ Top10perc    1   3948904 2159406904 8269.4
+ Books        1   3268063 2160087746 8269.6
+ Enroll       1   2753922 2160601887 8269.7
+ Accept       1   2334787 2161021022 8269.8
+ Apps         1   2189308 2161166501 8269.9
+ P.Undergrad  1    368680 2162987129 8270.3

Step:  AIC=8267.52
Outstate ~ Expend + Private + Room.Board + Grad.Rate + Terminal + 
    perc.alumni + Personal + PhD

              Df Sum of Sq        RSS    AIC
+ F.Undergrad  1  11718192 2140134303 8266.6
+ S.F.Ratio    1  10679814 2141172681 8266.8
<none>                     2151852495 8267.5
+ Enroll       1   4175531 2147676964 8268.5
+ Books        1   4048670 2147803825 8268.5
+ Top25perc    1   3273251 2148579244 8268.7
+ Apps         1   3174095 2148678400 8268.7
+ Top10perc    1   1761213 2150091282 8269.1
+ Accept       1   1490249 2150362246 8269.1
+ P.Undergrad  1    644040 2151208455 8269.4

Step:  AIC=8266.55
Outstate ~ Expend + Private + Room.Board + Grad.Rate + Terminal + 
    perc.alumni + Personal + PhD + F.Undergrad

              Df Sum of Sq        RSS    AIC
+ Accept       1  47088320 2093045984 8256.5
+ Enroll       1  15189070 2124945233 8264.7
```

```
+ S.F.Ratio    1   8553763 2131580540 8266.4
<none>                     2140134303 8266.6
+ Top25perc    1   6551799 2133582504 8266.9
+ Top10perc    1   3592035 2136542268 8267.6
+ Books        1   2989416 2137144888 8267.8
+ Apps         1   1112757 2139021546 8268.3
+ P.Undergrad  1    256687 2139877617 8268.5

Step:  AIC=8256.47
Outstate ~ Expend + Private + Room.Board + Grad.Rate + Terminal +
    perc.alumni + Personal + PhD + F.Undergrad + Accept

              Df Sum of Sq        RSS    AIC
+ Apps         1  79164108 2013881875 8237.5
+ S.F.Ratio    1   8647407 2084398576 8256.2
<none>                     2093045984 8256.5
+ Top25perc    1   6459335 2086586648 8256.8
+ Top10perc    1   5213842 2087832141 8257.1
+ Books        1   3293406 2089752578 8257.6
+ P.Undergrad  1    750194 2092295790 8258.3
+ Enroll       1    393724 2092652260 8258.4

Step:  AIC=8237.54
Outstate ~ Expend + Private + Room.Board + Grad.Rate + Terminal +
    perc.alumni + Personal + PhD + F.Undergrad + Accept + Apps

              Df Sum of Sq        RSS    AIC
+ Top10perc    1  27837286 1986044589 8232.0
+ Top25perc    1  19246648 1994635227 8234.3
<none>                     2013881875 8237.5
+ S.F.Ratio    1   7060053 2006821823 8237.6
+ Books        1   1730518 2012151357 8239.1
+ P.Undergrad  1   1232892 2012648983 8239.2
+ Enroll       1      5790 2013876085 8239.5

Step:  AIC=8231.98
Outstate ~ Expend + Private + Room.Board + Grad.Rate + Terminal +
    perc.alumni + Personal + PhD + F.Undergrad + Accept + Apps +
    Top10perc

              Df Sum of Sq        RSS    AIC
<none>                     1986044589 8232.0
+ S.F.Ratio    1   6156026 1979888563 8232.3
+ P.Undergrad  1   4348847 1981695742 8232.8
+ Books        1   3126503 1982918086 8233.1
```

```
+ Enroll       1    594092 1985450497 8233.8
+ Top25perc    1     76044 1985968546 8234.0
```

```r
summary(forward.fit)
```

```

Call:
lm(formula = Outstate ~ Expend + Private + Room.Board + Grad.Rate +
    Terminal + perc.alumni + Personal + PhD + F.Undergrad + Accept +
    Apps + Top10perc, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-8133.3 -1223.2   -73.9  1207.6  9912.1

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.789e+03  6.627e+02  -4.209 3.02e-05 ***
Expend       2.544e-01  2.388e-02  10.653  < 2e-16 ***
PrivateYes   2.241e+03  2.864e+02   7.826 2.76e-14 ***
Room.Board   8.694e-01  9.942e-02   8.744  < 2e-16 ***
Grad.Rate    3.063e+01  6.213e+00   4.930 1.10e-06 ***
Terminal     1.690e+01  1.114e+01   1.517  0.12980
perc.alumni  2.849e+01  8.763e+00   3.251  0.00122 **
Personal    -1.737e-01  1.329e-01  -1.307  0.19178
PhD          1.586e+01  1.086e+01   1.460  0.14491
F.Undergrad -1.986e-01  4.103e-02  -4.840 1.71e-06 ***
Accept       8.925e-01  1.412e-01   6.322 5.47e-10 ***
Apps        -4.081e-01  7.831e-02  -5.212 2.68e-07 ***
Top10perc    2.118e+01  7.770e+00   2.726  0.00663 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1936 on 530 degrees of freedom
Multiple R-squared:  0.7794,    Adjusted R-squared:  0.7745
F-statistic: 156.1 on 12 and 530 DF,  p-value: < 2.2e-16
```

```r
best.model <- formula(forward.fit)

coef(forward.fit)
```

```
  (Intercept)        Expend     PrivateYes     Room.Board      Grad.Rate
-2789.2134066     0.2543600  2241.2104060      0.8693687     30.6319664
     Terminal   perc.alumni       Personal            PhD    F.Undergrad
   16.9040841    28.4859406     -0.1736896     15.8615774     -0.1985515
       Accept          Apps      Top10perc
```

```
     0.8925125     -0.4081407     21.1770241
```

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.

```r
fit <- gam(Outstate ~ Private + s(Room.Board, df = 2) + s(PhD, df = 2) + s(perc.alumni, df = 2)
```

```
Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts argument
ignored
```

```r
par(mfrow = c(2, 3))
plot.Gam(fit, se = T, col = "blue")
```