Homework 4

```
knitr::opts_chunk$set(echo = TRUE)
library(ISLR2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(repr)
library(splines)
library(tree)
library(randomForest)
```

```
## randomForest 4.7-1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(gam)
```

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

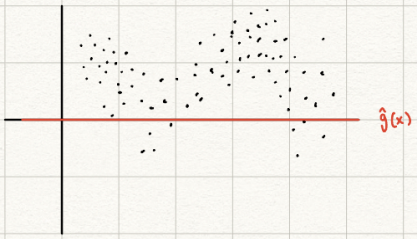
```
## Loaded gam 1.20.1
```

```
attach(Wage)
attach(Auto)
```

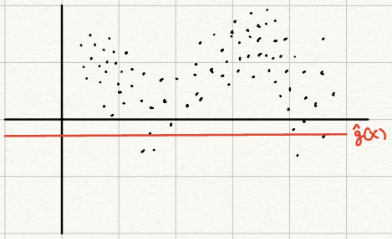
```
## The following object is masked from Wage:
##
##     year
```

1. $\hat{g} = \underset{g}{\operatorname{argmin}} \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int [g^{(m)}(x)]^2 \, dx \right)$

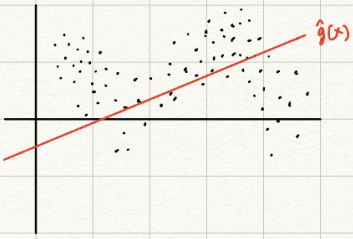
a) $\lambda = \infty$, $m = 0$



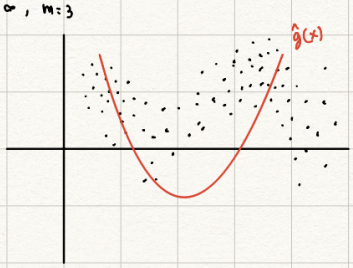
b) $\lambda = \infty$, $m = 1$



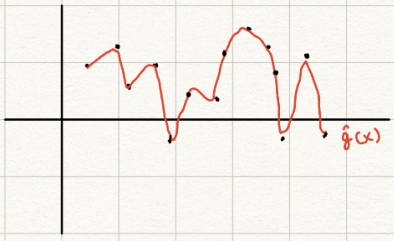
c) $\lambda = 0$, $m = 2$



d) $\lambda = \infty$, $m = 3$



e) $\lambda = 0$, $m = 3$

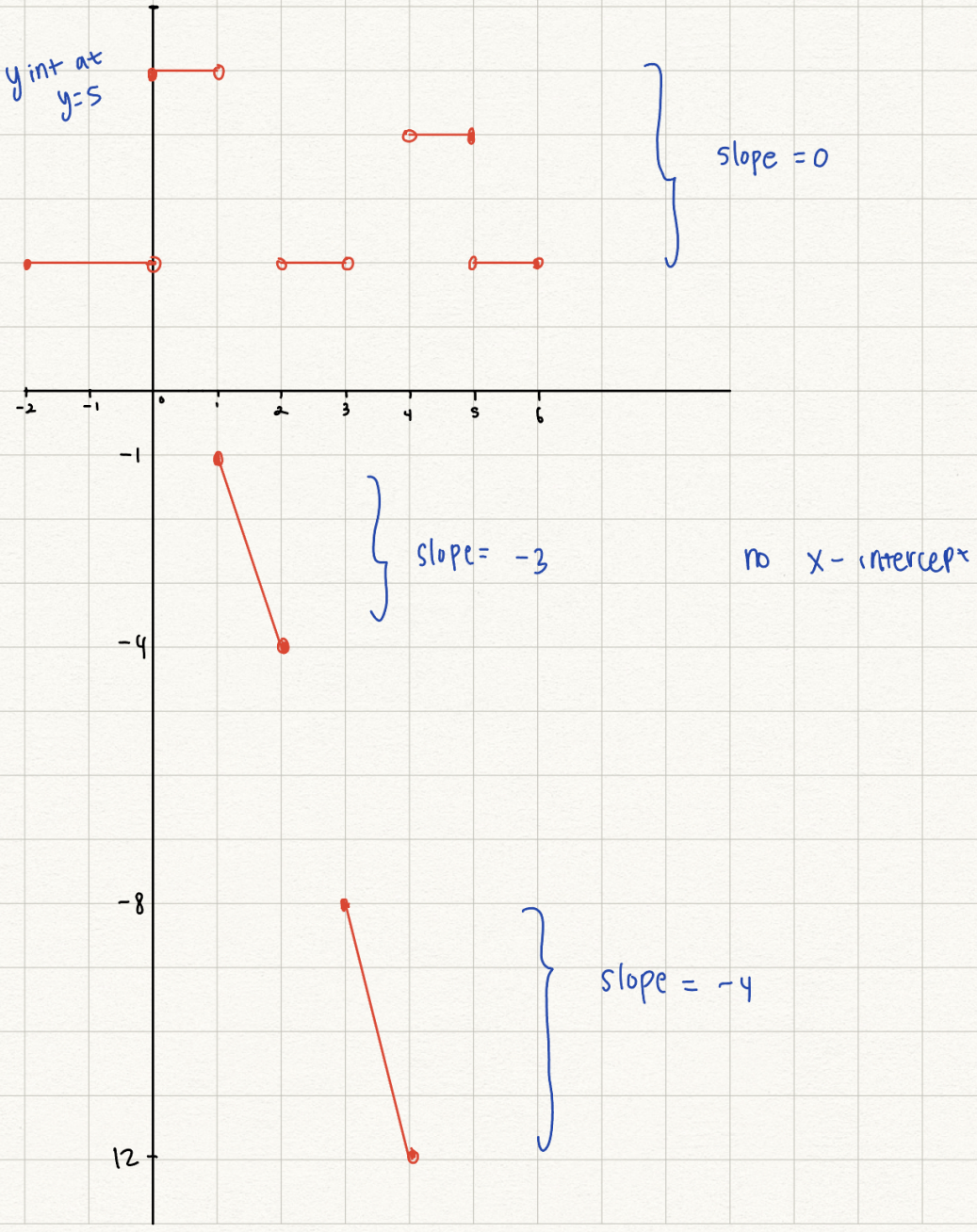


2. $b_1(x) = I(0 \le x \le 2) - (x+1) I(1 \le x \le 2)$

$b_2(x) = (2x-2) I(3 \le x \le 4) - I(4 < x \le 5)$

$Y = \beta_0 + \beta_1 b_1(x) + \beta_2 b_2(x) + \epsilon$

$\hat{\beta}_0 = 2 \quad \hat{\beta}_1 = 3 \quad \hat{\beta}_2 = -2$



3. Prove

$$f(x) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 (X - \psi)^3$$

is a cubic spline with knot at ψ

A) prove that it's cubic

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \hat{\beta}_3 x^3 + \hat{\beta}_4 (x - \psi)^3$$

$$= \begin{cases} \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \hat{\beta}_3 x^3 & \text{if } x < \psi \\ \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \underbrace{\hat{\beta}_3 x^3 + \hat{\beta}_4 (x - \psi)^3}_{\text{cubic + cubic = cubic}} & \text{if } x \geq \psi \end{cases}$$

$\therefore$ this is a piecewise cubic

B) prove it's continuous

$$f_1(\psi) = \hat{\beta}_0 + \hat{\beta}_1 \psi + \hat{\beta}_2 \psi^2 + \hat{\beta}_3 \psi^3$$

$$f_2(\psi) = \hat{\beta}_0 + \hat{\beta}_1 \psi + \hat{\beta}_2 \psi^2 + \hat{\beta}_3 \psi^3 + \hat{\beta}_4 (\psi - \psi)^3 = \hat{\beta}_0 + \hat{\beta}_1 \psi + \hat{\beta}_2 \psi^2 + \hat{\beta}_3 \psi^3$$

Here, we can see that $f_1(\psi) = f_2(\psi)$ which means that the function is continuous at the knot ψ

C) prove it's continuous at 1° & 2° derivative

$$f_1'(\psi) = \hat{\beta}_1 + 2\hat{\beta}_2 \psi + 3\hat{\beta}_3 \psi^2$$

$$f_2'(\psi) = \hat{\beta}_1 + 2\hat{\beta}_2 \psi + 3\hat{\beta}_3 \psi^2 + 3\hat{\beta}_4 (\psi - \psi)^2 = \hat{\beta}_1 + 2\hat{\beta}_2 \psi + 3\hat{\beta}_3 \psi^2$$

$$f_1''(\psi) = 2\hat{\beta}_2 + 6\hat{\beta}_3 \psi$$

$$f_2''(\psi) = 2\hat{\beta}_2 + 6\hat{\beta}_3 \psi + 6\hat{\beta}_4 (\psi - \psi) = 2\hat{\beta}_2 + 6\hat{\beta}_3 \psi$$

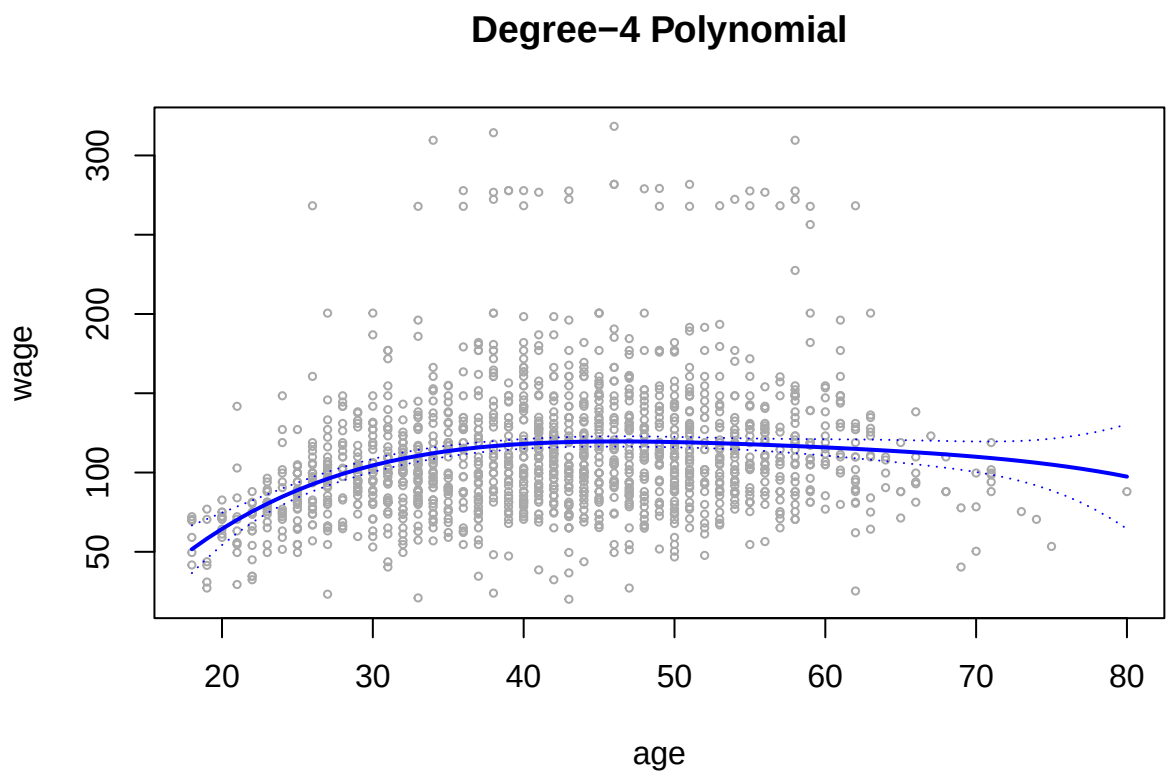
Here, we can see that $f_1'(\psi) = f_2'(\psi)$ & $f_1''(\psi) = f_2''(\psi)$ which means that the function is continuous at the knot ψ

```
set.seed(435)

train <- sample(1:nrow(Wage), nrow(Wage)/2)
```

```r
wage.train <- Wage[train,]
wage.test <- Wage[-train,]

# Part 4a (polynomial)
(fit <- lm(wage ~ poly(age, 4), data=Wage, subset=train))
```

```
##
## Call:
## lm(formula = wage ~ poly(age, 4), data = Wage, subset = train)
##
## Coefficients:
##   (Intercept)  poly(age, 4)1  poly(age, 4)2  poly(age, 4)3  poly(age, 4)4
##        112.00         437.34        -460.41         173.72         -60.97
```

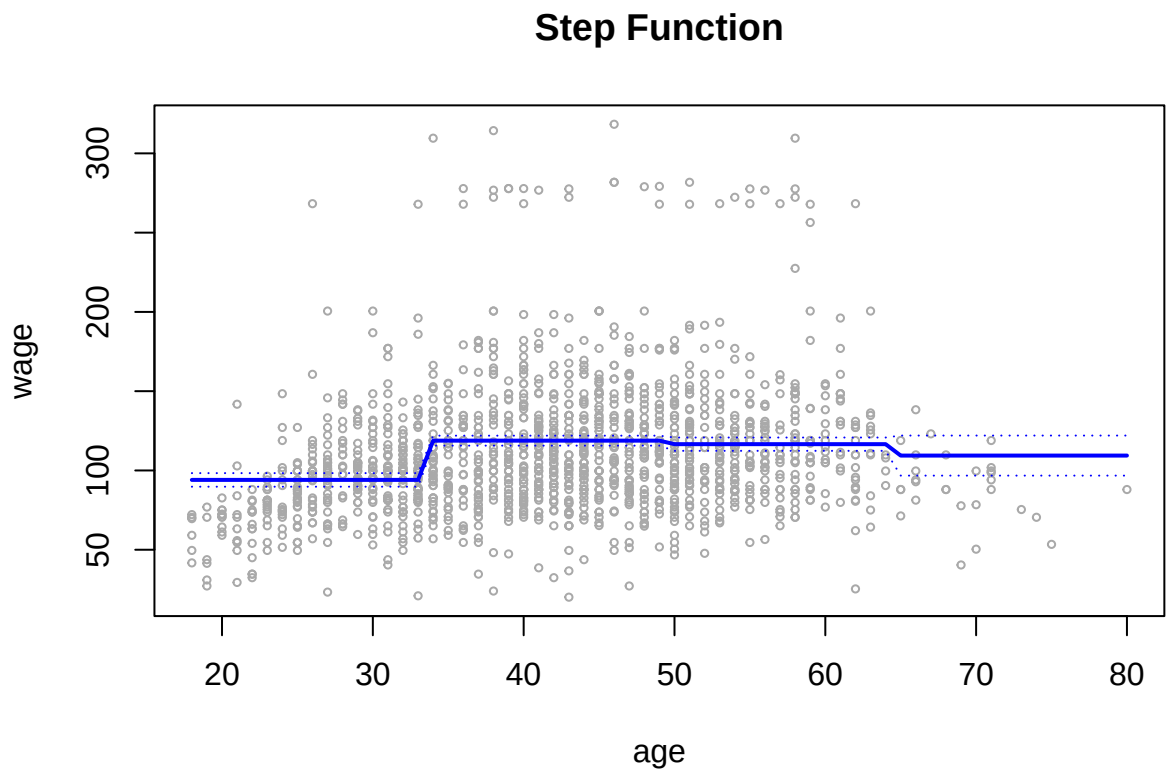
```r
agelims <- range(age)
age.grid <- seq(from = agelims[1], to = agelims[2])
preds_poly <- predict(fit, newdata=list(age=age.grid), se=TRUE)
se.bands <- cbind(preds_poly$fit + 2 * preds_poly$se.fit,
                  preds_poly$fit - 2 * preds_poly$se.fit)
options(repr.plot.width=12, repr.plot.height=6)
plot(wage.test$age, wage.test$wage, xlim=agelims, cex=.5, col="darkgrey",
     xlab="age", ylab="wage")
title("Degree-4 Polynomial")
lines(age.grid, preds_poly$fit, lwd=2, col="blue")
matlines(age.grid, se.bands, lwd=1, col="blue", lty=3)
```



```r
# Part 4b (step function)
(fit <- lm(wage ~ cut(age, 4), data=Wage, subset=train))
```

```
##
## Call:
## lm(formula = wage ~ cut(age, 4), data = Wage, subset = train)
##
## Coefficients:
##            (Intercept)    cut(age, 4)(33.5,49]    cut(age, 4)(49,64.5]
##                  94.04                   24.77                   22.54
## cut(age, 4)(64.5,80.1]
##                  15.34
```

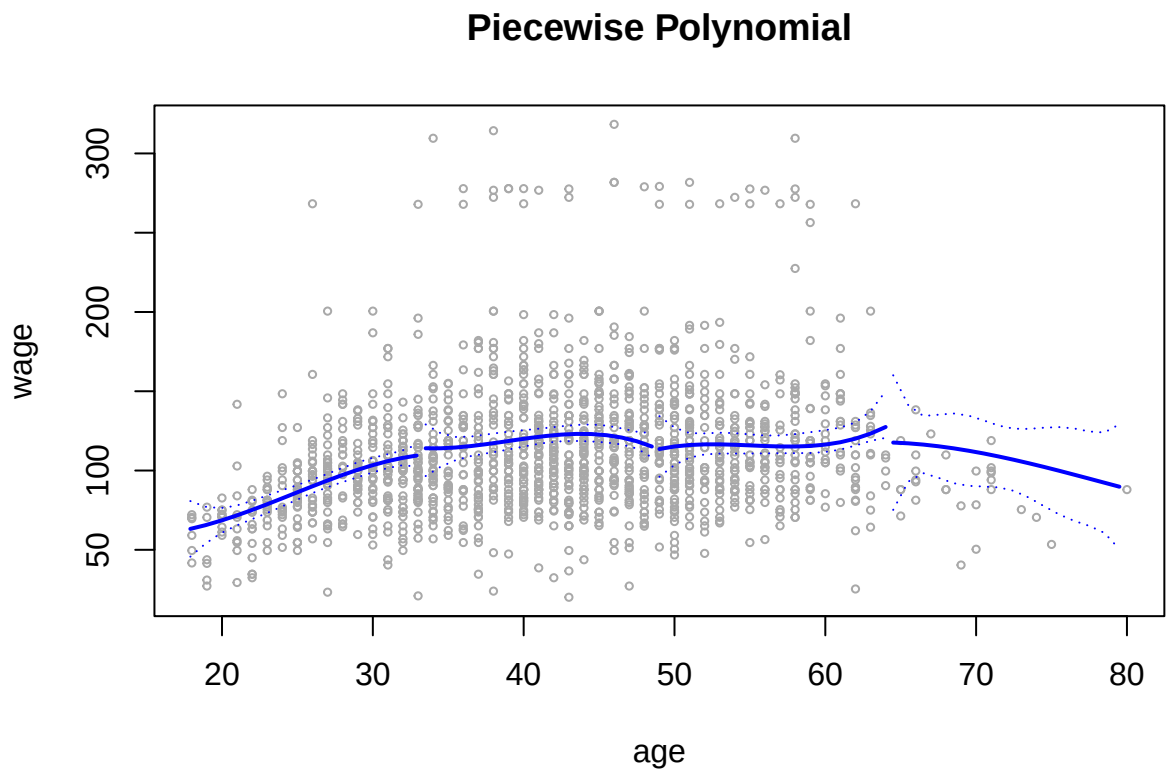
```r
preds_poly <- predict(fit, newdata=list(age=age.grid), se=TRUE)
se.bands <- cbind(preds_poly$fit + 2 * preds_poly$se.fit,
                  preds_poly$fit - 2 * preds_poly$se.fit)
options(repr.plot.width=12, repr.plot.height=6)
plot(wage.test$age, wage.test$wage, xlim=agelims, cex=.5, col="darkgrey",
     xlab="age", ylab="wage")
title("Step Function")
lines(age.grid, preds_poly$fit, lwd=2, col="blue")
matlines(age.grid, se.bands, lwd=1, col="blue", lty=3)
```



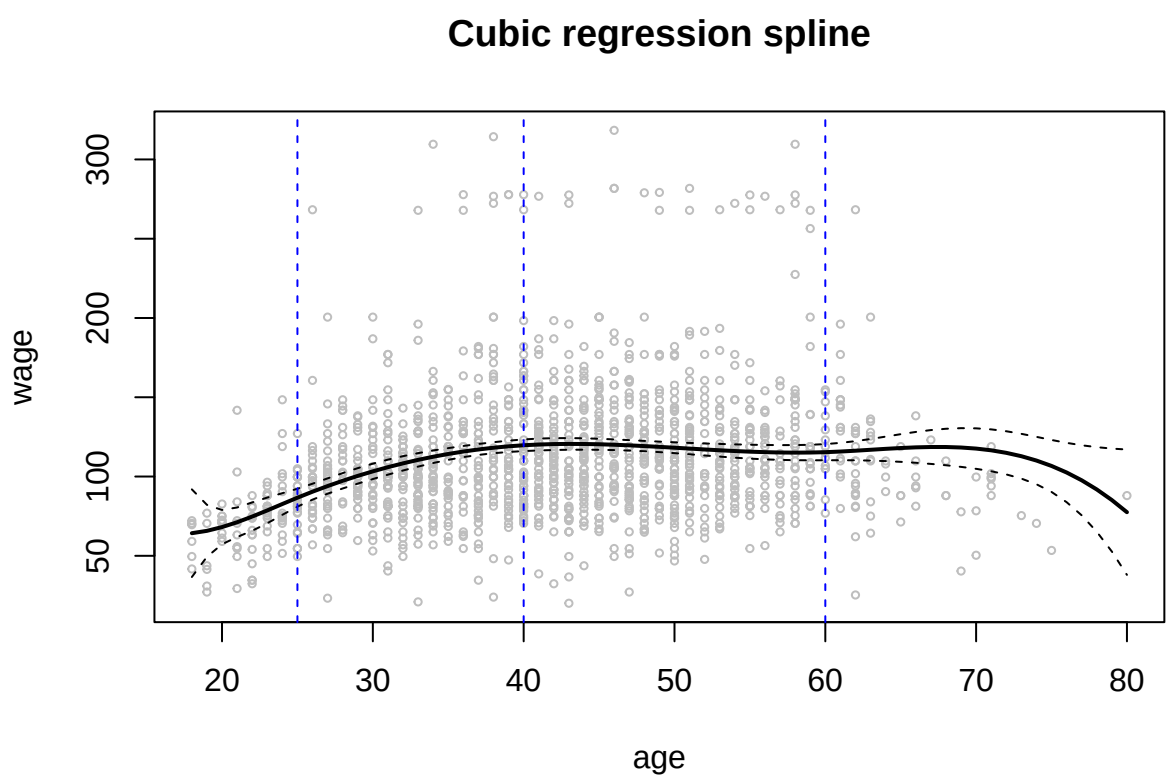
```r
# Part 4c (piecewise polynomials)
table(cut(wage.train$age, 4))
```

```
##
## (17.9,33.5]   (33.5,49]   (49,64.5] (64.5,80.1]
##         375         692         390          43
```

```r
fit1 <- lm(wage ~ poly(age, 3), data = wage.train[wage.train$age <= 33.5,])
fit2 <- lm(wage ~ poly(age, 3),
           data = wage.train[wage.train$age > 33.5 & wage.train$age <= 49,])
fit3 <- lm(wage ~ poly(age, 3),
           data = wage.train[wage.train$age > 49 & wage.train$age <= 64.5,])
fit4 <- lm(wage ~ poly(age, 3),
           data = wage.train[wage.train$age > 64.5,])

age.grid1 <- seq(from = 17.9, to = 33.5)
age.grid2 <- seq(from = 33.5, to = 49)
age.grid3 <- seq(from = 49, to = 64.5)
age.grid4 <- seq(from = 64.5, to = 80.1)

pred1 <- predict(fit1, newdata=list(age=age.grid1), se=TRUE)
pred2 <- predict(fit2, newdata=list(age=age.grid2), se=TRUE)
pred3 <- predict(fit3, newdata=list(age=age.grid3), se=TRUE)
pred4 <- predict(fit4, newdata=list(age=age.grid4), se=TRUE)

bands1 <- cbind(pred1$fit + 2 * pred1$se.fit, pred1$fit - 2 * pred1$se.fit)
bands2 <- cbind(pred2$fit + 2 * pred2$se.fit, pred2$fit - 2 * pred1$se.fit)
bands3 <- cbind(pred3$fit + 2 * pred3$se.fit, pred3$fit - 2 * pred1$se.fit)
bands4 <- cbind(pred4$fit + 2 * pred4$se.fit, pred4$fit - 2 * pred4$se.fit)

options(repr.plot.width=12, repr.plot.height=6)
plot(wage.test$age, wage.test$wage, xlim=agelims, cex=.5, col="darkgrey",
     xlab="age", ylab="wage")
title("Piecewise Polynomial")
matlines(age.grid1, bands1, lwd=1, col="blue", lty=3)
matlines(age.grid2, bands2, lwd=1, col="blue", lty=3)
matlines(age.grid3, bands3, lwd=1, col="blue", lty=3)
matlines(age.grid4, bands4, lwd=1, col="blue", lty=3)
lines(age.grid1, pred1$fit, lwd=2, col="blue")
lines(age.grid2, pred2$fit, lwd=2, col="blue")
lines(age.grid3, pred3$fit, lwd=2, col="blue")
lines(age.grid4, pred4$fit, lwd=2, col="blue")
```



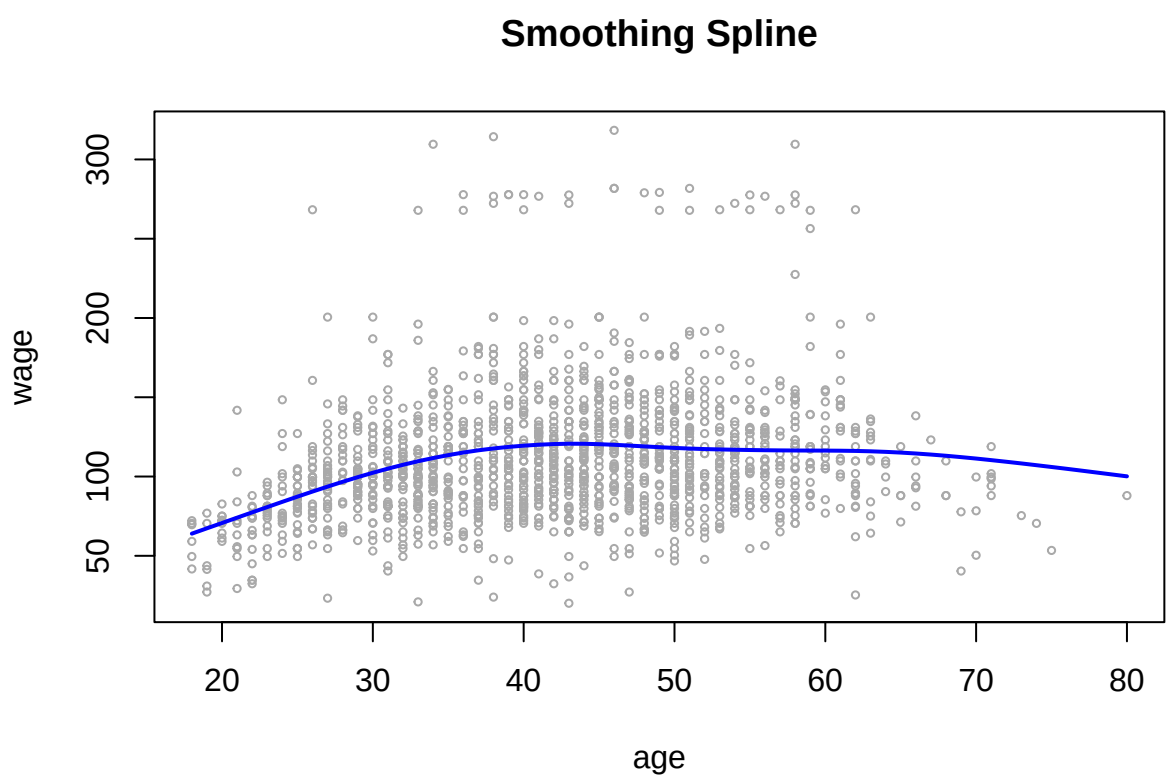

```
# Part 4d (cubic spline)
fit <- lm(wage ~ bs(age, knots = c(25, 40, 60)), data = Wage, subset = train)
pred_bs <- predict(fit, newdata = list(age = age.grid), se = T)

plot(wage.test$age, wage.test$wage, col="gray", xlab="age", ylab="wage",
     main="Cubic regression spline", cex=.5)
lines(age.grid, pred_bs$fit, lwd=2)
lines(age.grid, pred_bs$fit + 2 * pred_bs$se, lty="dashed")
lines(age.grid, pred_bs$fit - 2 * pred_bs$se, lty="dashed")
abline(v=25, lty="dashed", col="blue")
abline(v=40, lty="dashed", col="blue")
abline(v=60, lty="dashed", col="blue")
```




```
# Part 4e (smoothing spline)
(fit <- smooth.spline(wage.train$age, wage.train$wage, cv = TRUE))
```

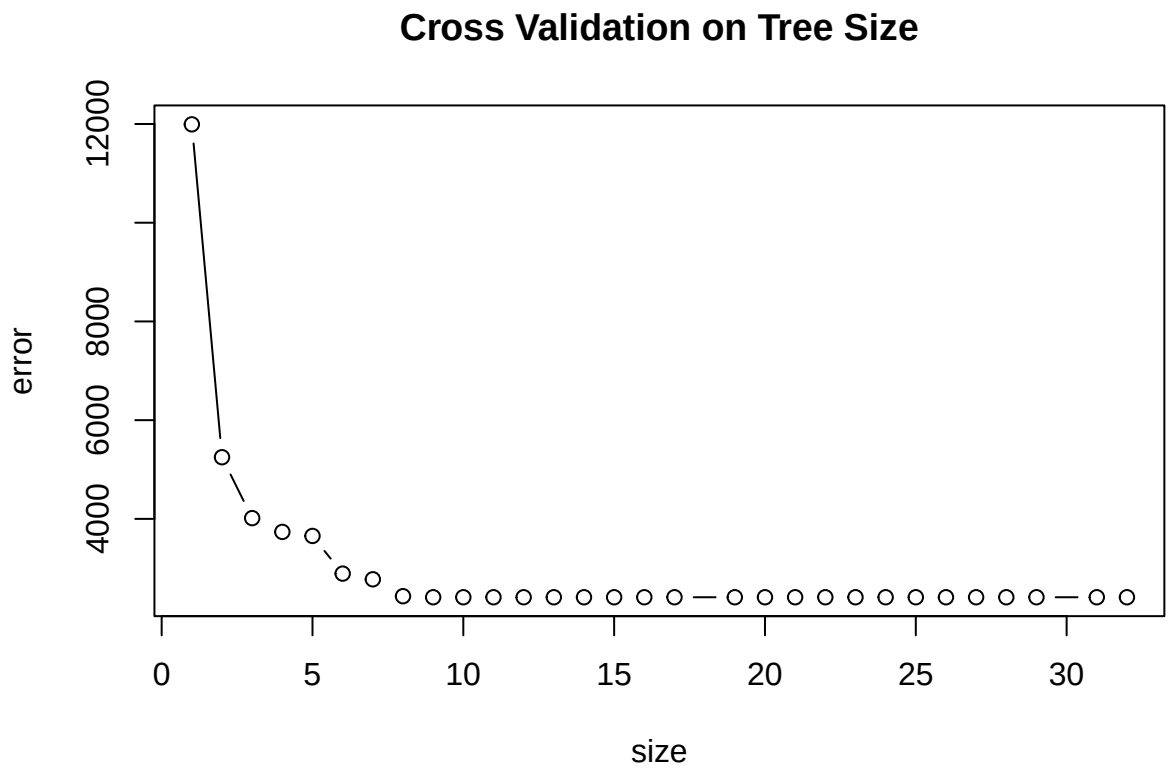
```
## Call:
## smooth.spline(x = wage.train$age, y = wage.train$wage, cv = TRUE)
##
## Smoothing Parameter  spar= 0.7623456  lambda= 0.04008168 (11 iterations)
## Equivalent Degrees of Freedom (Df): 5.461871
## Penalized Criterion (RSS): 86822.37
## PRESS(l.o.o. CV): 1672.657
```

```
plot(wage.test$age, wage.test$wage, xlim = agelims, cex=.5, col="darkgrey",
     xlab="age", ylab="wage", main="Smoothing Spline")
lines(fit, col="blue", lwd=2)
```

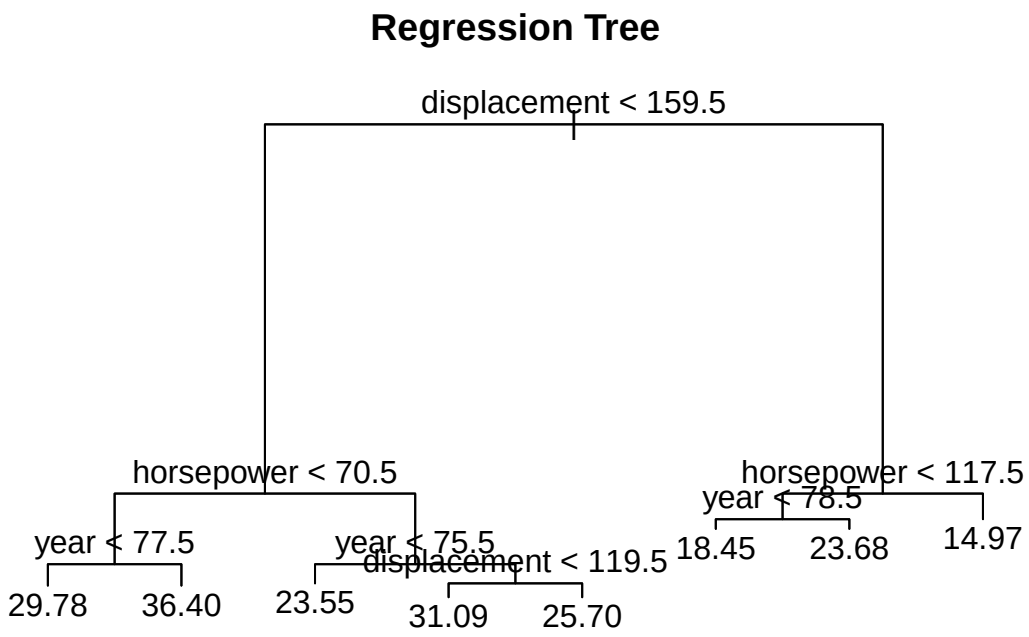


I believe that the smoothing spline is the best model because it considers the edge cases pretty well compared to the other models. The piecewise polynomial and the step functions model are hard to understand or interpret because the lines weren't smooth on the knots (we can see that there are cuts on the knots). However, we can see that the models are relatively similar in terms of its shape. It first increases as age increases, then it decreases as the age passes 40.

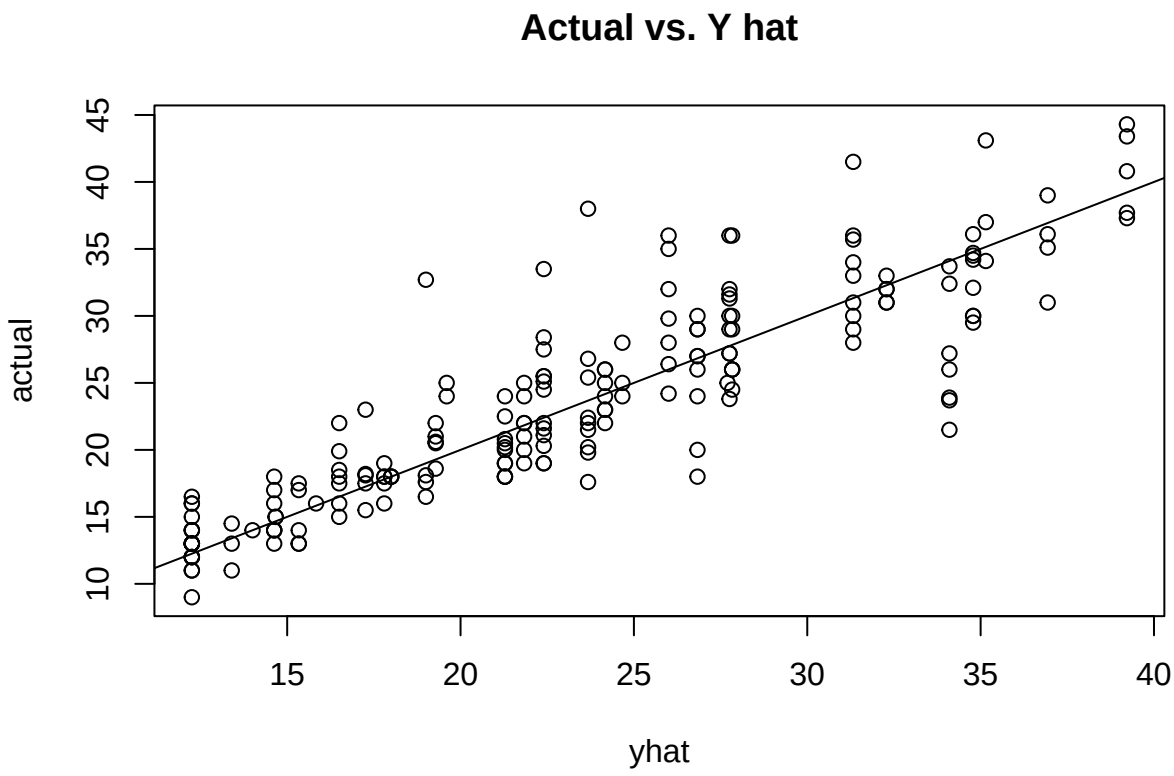
```
# Part 5a
set.seed(435)
Auto <- select(Auto, c(1:8))
train <- sample(1 : nrow(Auto), nrow(Auto) / 2)
tree.auto <- tree(mpg ~ ., Auto, subset = train,
                  control = tree.control(nobs = length(train), mindev = 0))

cv.auto <- cv.tree(tree.auto)
plot(cv.auto$size, cv.auto$dev, type="b", xlab="size", ylab="error",
     main="Cross Validation on Tree Size")
```




```
prune.auto <- prune.tree(tree.auto, best = 8)
plot(prune.auto)
title("Regression Tree")
text(prune.auto, pretty = 0)
```




```
yhat <- predict(tree.auto, newdata = Auto[-train, ])
plot(yhat, Auto[-train, "mpg"],
ylab="actual", main="Actual vs. Y hat")
abline(0, 1)
```

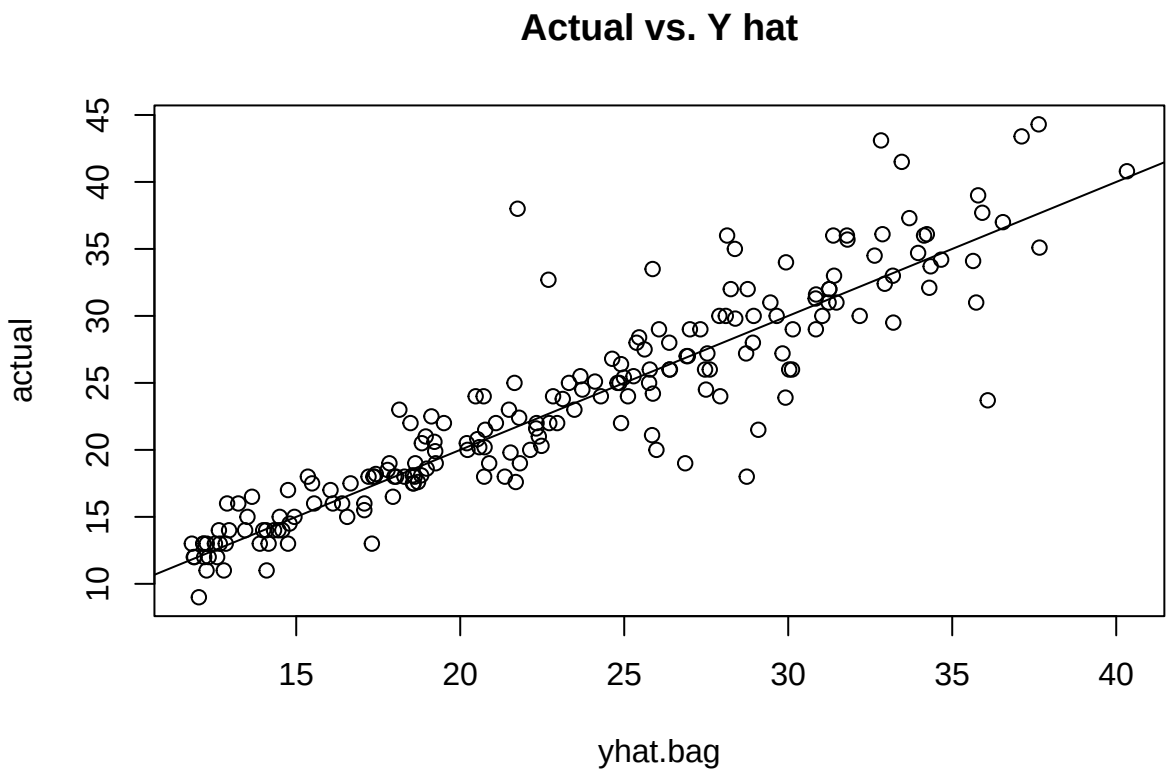



```
mean((yhat - Auto[-train, "mpg"])^2)
```

```
## [1] 13.62
```

As we can see, our predicted y values are linear with respect to the actual values of mpg, and it has an MSE of 13.62. This means that the model we use predicted the proper regions. We can also see that from our cross validation graph that there is a sharp decline from 1 to 2, then it keeps on decreasing since then. We can also see from the regression tree that displacement plays an important role as a predictor to determine mpg.

```
# Part 5b
set.seed(435)
bag.auto <- randomForest(mpg ~., data = Auto,
                         subset=train, mtry=7, importance = TRUE)
yhat.bag <- predict(bag.auto, newdata = Auto[-train, ])
auto.test <- Auto[-train, "mpg"]

plot(yhat.bag, auto.test, ylab="actual", main="Actual vs. Y hat")
abline(0, 1)
```

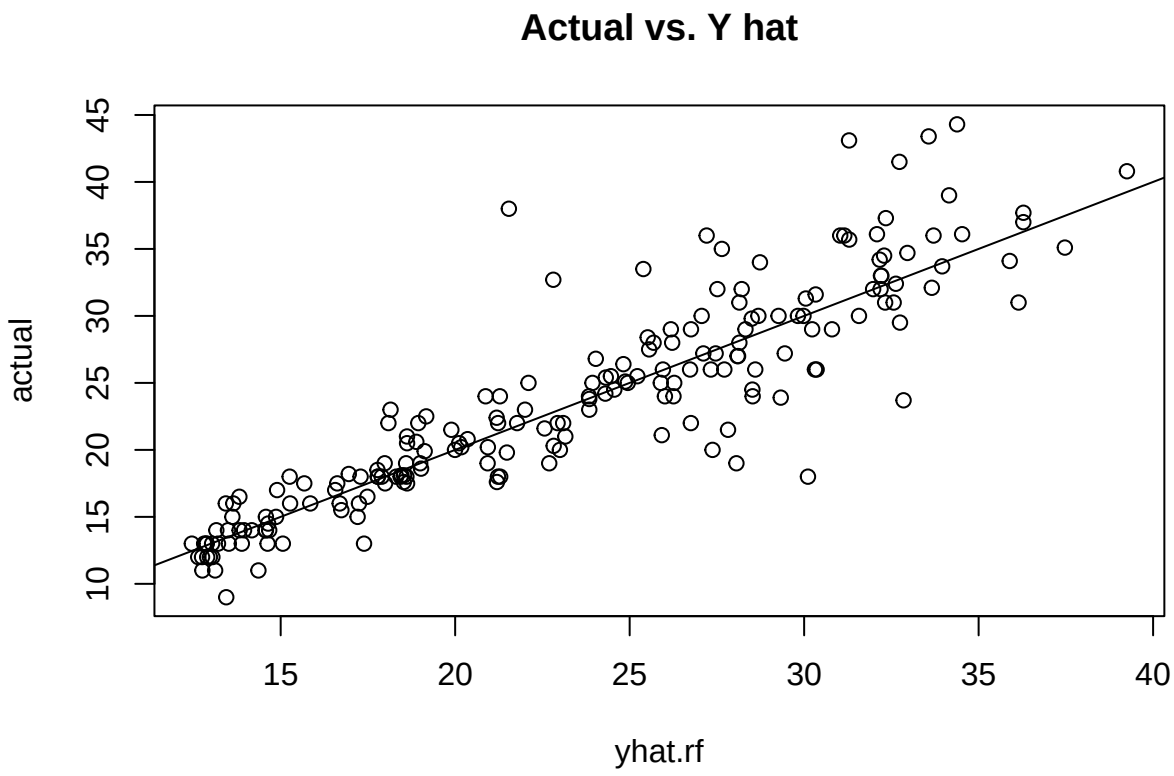



```
paste("The MSE is", mean((yhat.bag - auto.test)^2))
```

```
## [1] "The MSE is 9.75746069785654"
```

Since we are doing bagging and we have 7 predictors, our tuning parameter is m = 7 (m=p)

```
# Part 5c
set.seed(435)
rf.auto <- randomForest(mpg ~., data = Auto,
                        subset=train, mtry=7/3, importance = TRUE)
yhat.rf <- predict(rf.auto, newdata = Auto[-train, ])
auto.test <- Auto[-train, "mpg"]

plot(yhat.rf, auto.test, ylab="actual", main="Actual vs. Y hat")
abline(0, 1)
```

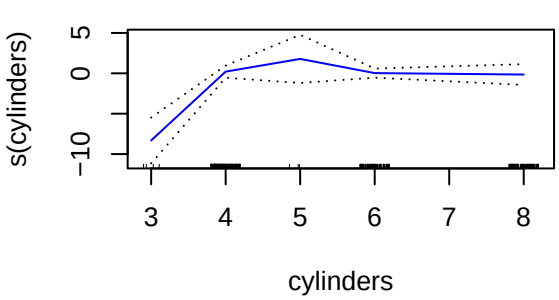


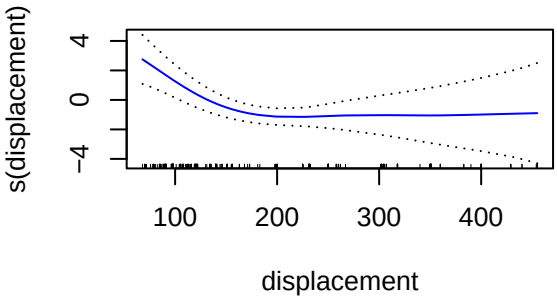

```
paste("The MSE is", mean((yhat.rf - auto.test)^2))
```

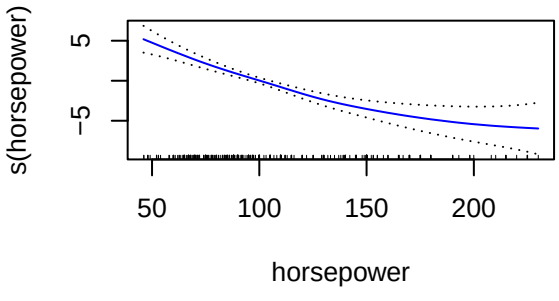
```
## [1] "The MSE is 11.3559812524429"
```

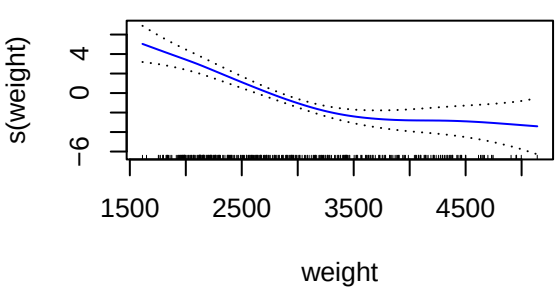
Our default tuning parameter is m = 7/3 since with regression trees, m = p/3

```
# part 5d
gam.m7 <- gam(mpg ~ s(cylinders) + s(displacement) + s(horsepower) + s(weight)
              + s(acceleration) + s(year) + origin, data = Auto)
par(mfrow=c(2, 2))
plot(gam.m7, se=TRUE, col="blue")
```



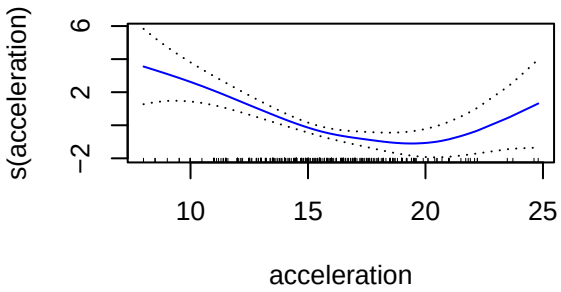


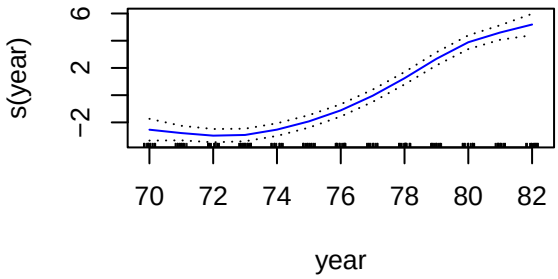


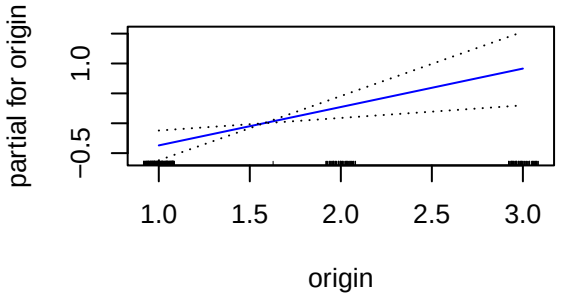



```
auto.test <- Auto[-train, "mpg"]

yhat <- predict(gam.m7, newdata = Auto[-train, ])
paste("MSE is", mean((yhat - auto.test)^2))
```

```
## [1] "MSE is 6.70467987738433"
```







We can see that the MSE is pretty small because we are fitting individual smoothing splines for each variable. It is useful as we can see how each variable interacts with mpg.

5e. In my opinion, I still think that the first model (model in part a) is the best in terms of its interpretability. It is easier to understand as we can easily decide the values for each datapoints. Although the other model has lower MSEs, they are harder to interpret.