BIOSTAT 285 Spring 2020 Homework 2

Due by 11:00 PM, 05/14/2020

Nan Liu

Remark. For **Computational Part**, please complete your answer in the **RMarkdown** file and summit the generated PDF and RMD files.

Computational Part

1. ([ISL] 4.11, *25 pt*) In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the `Auto` data set. Write a data analysis report addressing the following problems.

(a) Create a binary variable, `mpg01`, that contains a 1 if `mpg` contains a value above its median, and a 0 if `mpg` contains a value below its median.

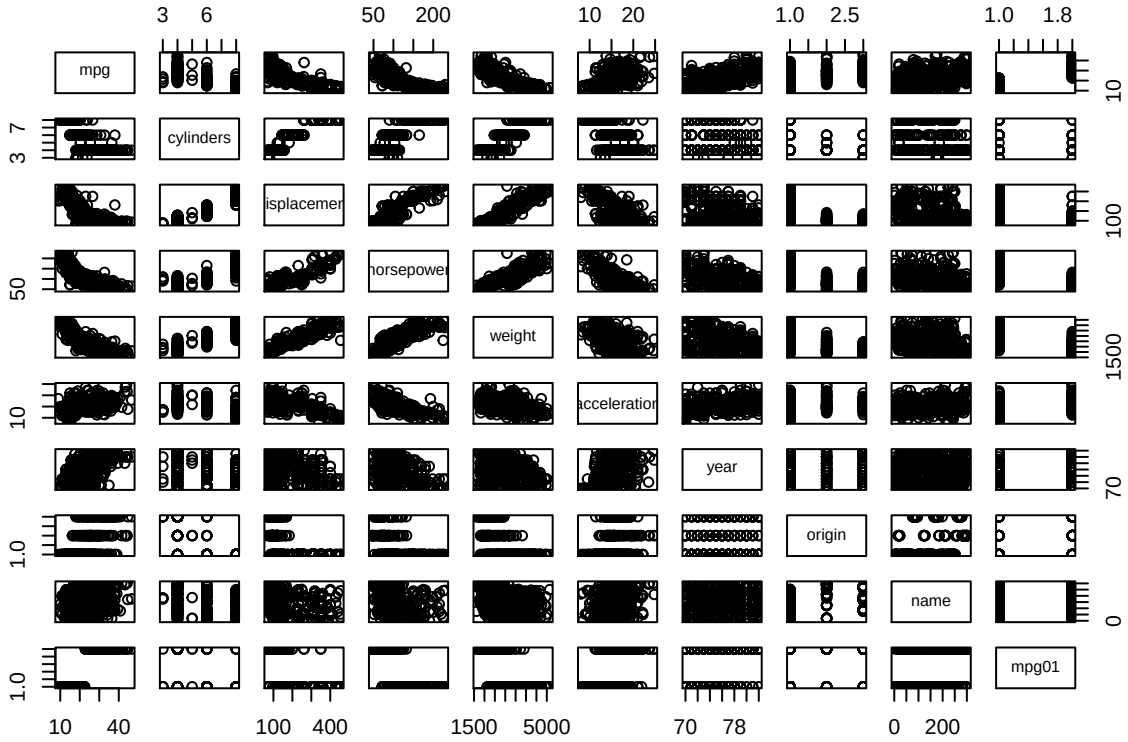
```
library(ISLR)
data("Auto")
```

```
set.seed(123)
Auto <- Auto %>%
  mutate (mpg01 = factor(ifelse(mpg > median(mpg), 1, 0)))
median(Auto$mpg)
```

```
## [1] 22.75
```

(b) Explore the data graphically in order to investigate the association between `mgp01` and the other features. Which of the other features seem most likely to be useful in predicting `mpg01`? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

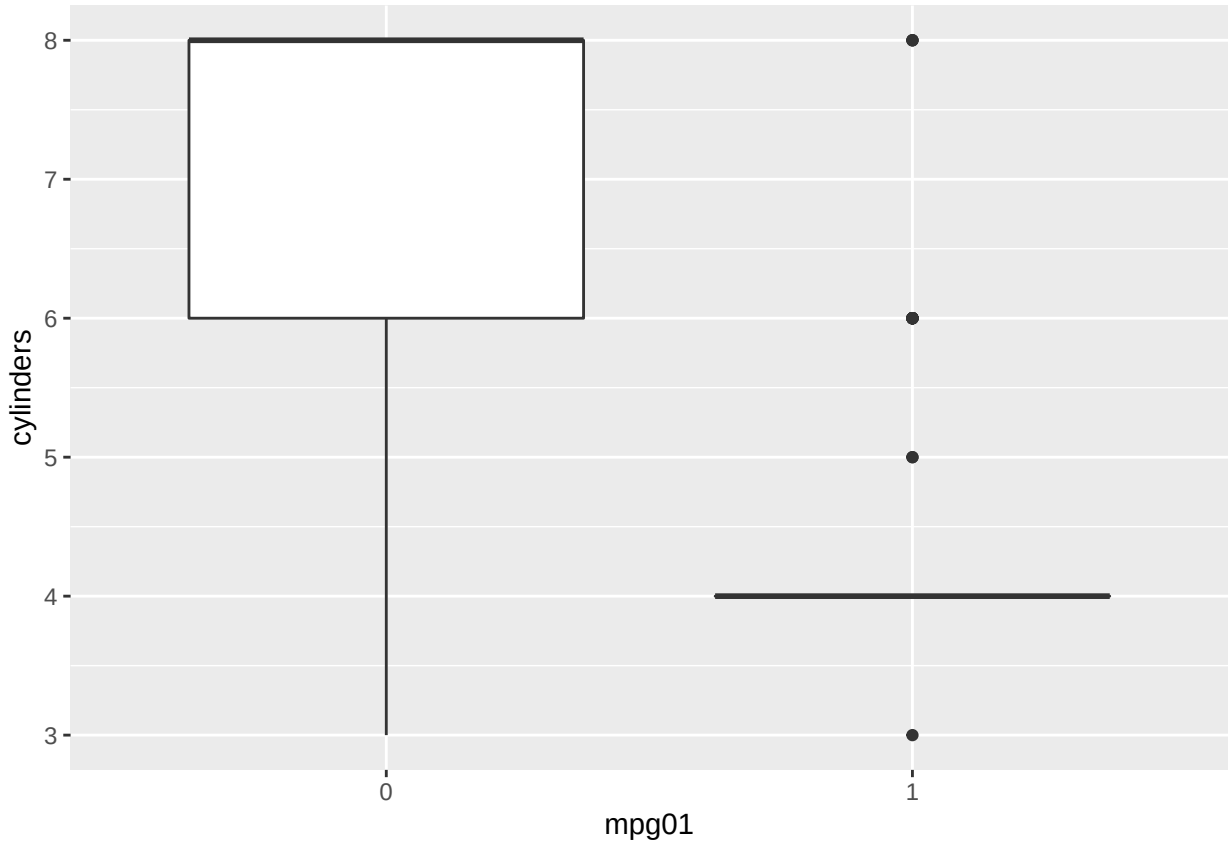
```
#scatter plots
pairs(Auto)
```



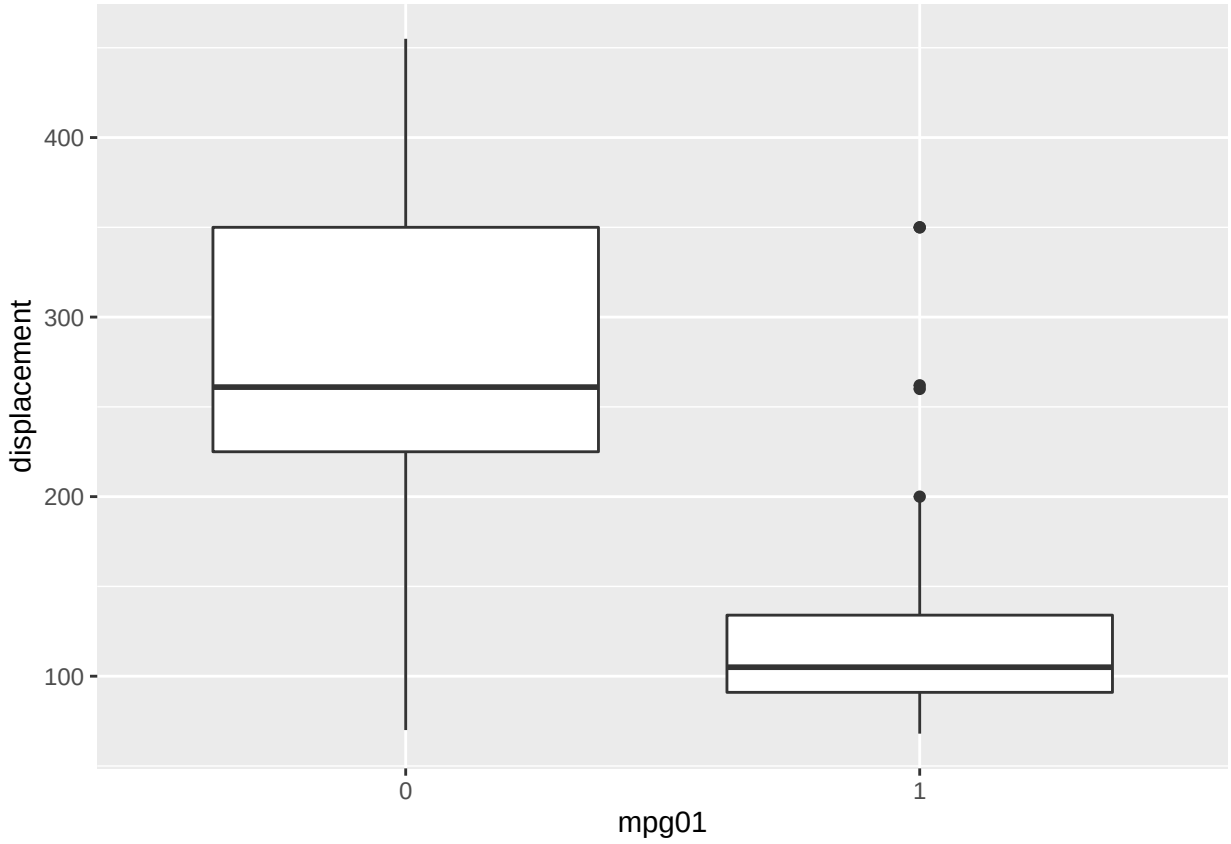
From the scatter plot, we assume 'cylinders', 'displacement', 'horsepower' and 'weight' seem mostly likely to be useful in predicting `mpg01`.

Now let's display the boxplot

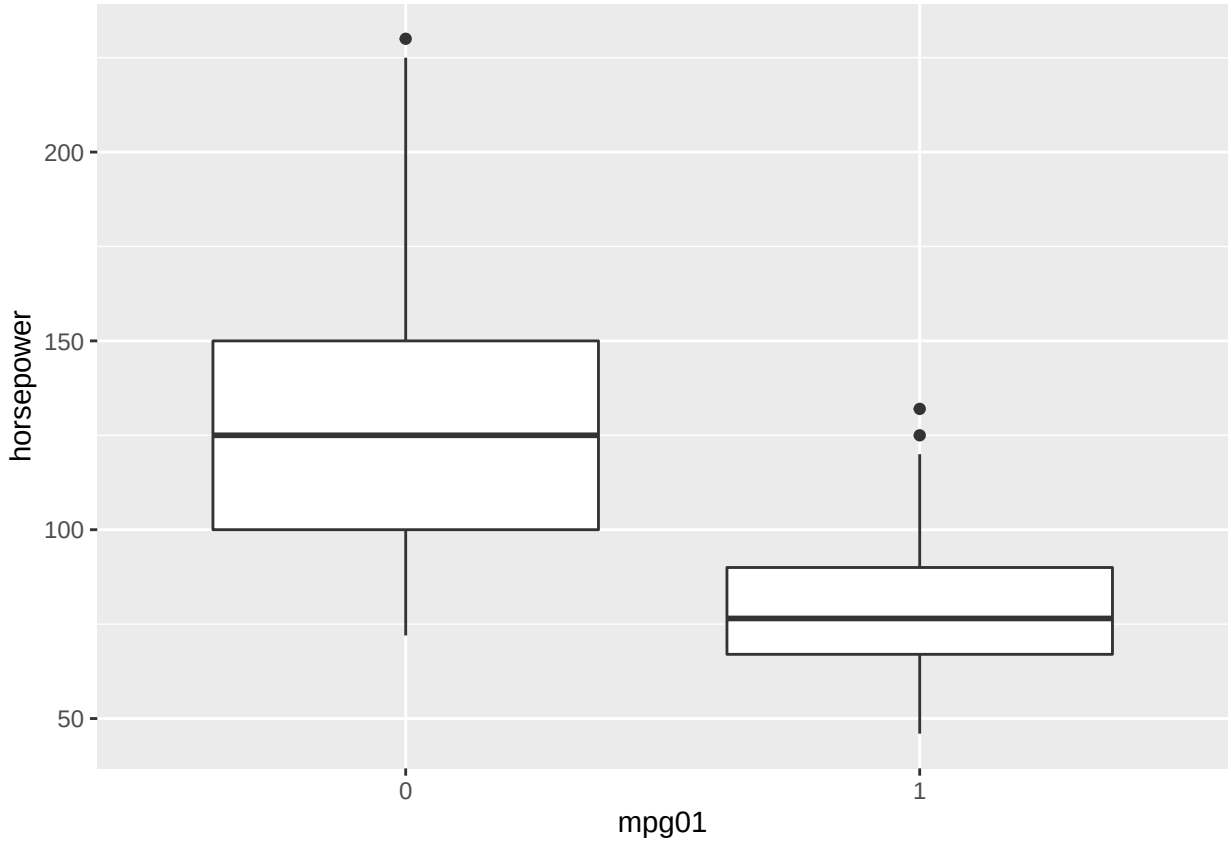
```
#Boxplots
ggplot(Auto, aes(mpg01,cylinders)) + geom_boxplot()
```



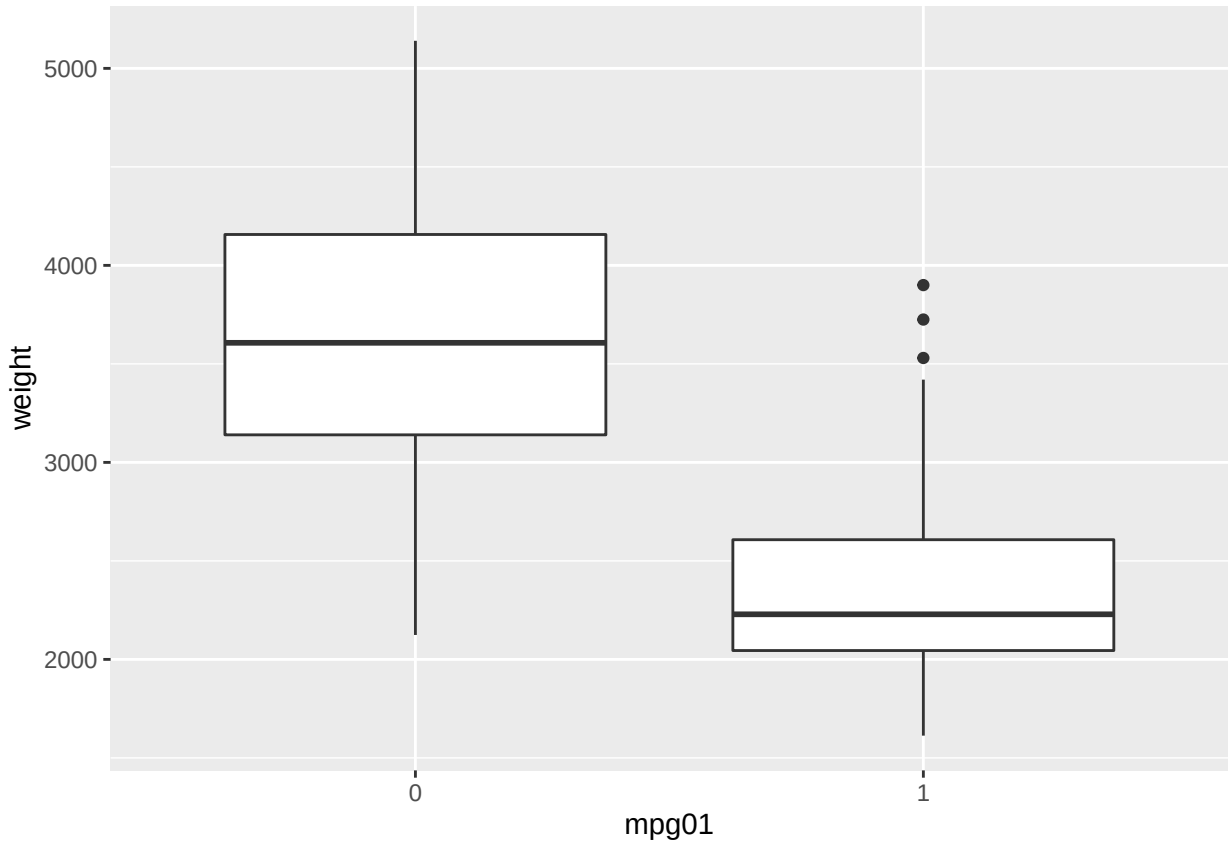
```
ggplot(Auto, aes(mpg01,displacement)) + geom_boxplot()
```



```
ggplot(Auto, aes(mpg01, horsepower)) + geom_boxplot()
```



```
ggplot(Auto, aes(mpg01,weight)) + geom_boxplot()
```



(c) Split the data into a training set and a test set.

```
#split data in to 70% traing set and 30% test set
set.seed(123)
s <- sample(nrow(Auto), floor(nrow(Auto)*0.7), replace = F)
training <- Auto[s,]
test <- Auto[-s,]
```

(d) Perform LDA on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (b). What is the test error of the model obtained?

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

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

```r
lda.fit <- lda(mpg01 ~ cylinders + displacement + horsepower + weight, data = training)

#test error
lda.pred <- predict(lda.fit, test)
mean(lda.pred$class != test$mpg01)
```

```
## [1] 0.1101695
```

The test error is about 11%.

(e) Perform QDA on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (b). What is the test error of the model obtained?

```r
qda.fit <- qda(mpg01 ~ cylinders + displacement + horsepower + weight, data = training)

#test error
qda.class <- predict(qda.fit, test)$class
mean(qda.class != test$mpg01)
```

```
## [1] 0.1016949
```

The test error is about 10%

(f) Perform logistic regression on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (b). What is the test error of the model obtained?

```r
glm.fit <- glm(mpg01 ~ cylinders + displacement + horsepower + weight,
               data = training, family=binomial)
#test error
prob <- predict(glm.fit, test, type = "response")
glm.pred <- ifelse(prob > 0.5, 1, 0)
mean(glm.pred != test$mpg01)
```

```
## [1] 0.1101695
```

The test error is about 11%

Report

From the scatter plot, we assume 'cylinders', 'displacement', 'horsepower' and 'weight' seem mostly likely to be useful in predicting `mpg01`. The box plots also agree with this variable selection. Then we split our data into 70% traing set and 30% test set. In order to predict `mpg01`, we perform LDA, QDA and logistic regression on the training set using the variables 'cylinders', 'displacement', 'horsepower' and 'weight'. The test error from LDA method, QDA method and logistic regression method are all about 11%.

2. (New York Times Covid-19 State Data, *35 pt*) Download the data file `covid-19-state-level-data.csv`, then answer the following questions using splines.

(a) Create subsets of the data for States New York, California, and Washington. Print out the dimension and first 6 observations of each state subset.

```r
#import csv file
covid <- read.csv("covid-19-state-level-data.csv")
#subset new york
ny <- filter(covid, state == "New York")
dim(ny)
```

```
## [1] 58  5
```

```r
head(ny,6)
```

```
##         date    state fips cases deaths
## 1 2020-03-01 New York   36     1      0
## 2 2020-03-02 New York   36     1      0
## 3 2020-03-03 New York   36     2      0
## 4 2020-03-04 New York   36    11      0
## 5 2020-03-05 New York   36    22      0
## 6 2020-03-06 New York   36    44      0
```

```r
#california
ca <- filter(covid, state == "California")
dim(ca)
```

```
## [1] 94  5
```

```r
head(ca,6)
```

```
##         date      state fips cases deaths
## 1 2020-01-25 California    6     1      0
## 2 2020-01-26 California    6     2      0
## 3 2020-01-27 California    6     2      0
## 4 2020-01-28 California    6     2      0
## 5 2020-01-29 California    6     2      0
## 6 2020-01-30 California    6     2      0
```

```r
#washington
wa <- filter(covid, state == "Washington")
dim(wa)
```

```
## [1] 98  5
```

```r
head(wa,6)
```

```
##         date      state fips cases deaths
## 1 2020-01-21 Washington   53     1      0
## 2 2020-01-22 Washington   53     1      0
## 3 2020-01-23 Washington   53     1      0
## 4 2020-01-24 Washington   53     1      0
## 5 2020-01-25 Washington   53     1      0
## 6 2020-01-26 Washington   53     1      0
```

(b) Fit both cubic splines (using `bs`) and natural cubic splines (using `ns`) on each subset, with the number of cases as response, and the number of days since first case as predictor. Find the optimal degrees of freedom (`df`) for the basis.

```
#generate the number of days since first case
ny <- ny %>%
  mutate(days = as.numeric(as.Date(as.character(date))-
                  as.Date(as.character("2020-03-01"))))
ca <- ca %>%
  mutate(days = as.numeric(as.Date(as.character(date))-
                  as.Date(as.character("2020-01-25"))))
wa <- wa %>%
  mutate(days = as.numeric(as.Date(as.character(date))-
                  as.Date(as.character("2020-01-21"))))
```

```
#find the optimal degree of freedom
#split 80% training, 20% test
set.seed(6)
s <- sample(nrow(ny), floor(nrow(ny)*0.8), replace = F)
nytraining <- ny[s,]
nytest <- ny[-s,]

library(splines)
#bs
se <- c()
for (i in 3:40){
fit <- lm(cases ~ bs(days, df = i, knots = NULL,
                        degree = 3, intercept = FALSE), data = nytraining)
pred <- predict(fit, nytest)
se[i] <- mean((pred - nytest$cases)^2)
}
which.min(se)
```

```
## [1] 39
```

```
#ns
se <- c()
for (i in 3:40){
fit <- lm(cases ~ ns(days, df = i, knots = NULL,
                      intercept = FALSE), data = nytraining)
pred <- predict(fit, nytest)
se[i] <- mean((pred - nytest$cases)^2)
}
which.min(se)
```

```
## [1] 37
```

The optimal degree of freedom for New York is 39 when using cubic splines and 37 when using natural cubic splines.

```
#split 80% training, 20% test
set.seed(8)
```

```
s <- sample(nrow(ca), floor(nrow(ca)*0.8), replace = F)
catraining <- ca[s,]
catest <- ca[-s,]

#bs df = 37
se <- c()
for (i in 3:40){
fit <- lm(cases ~ bs(days, df = i, knots = NULL, degree = 3,
                     intercept = FALSE), data = catraining)
pred <- predict(fit, catest)
se[i] <- mean((pred - catest$cases)^2)
}
which.min(se)
```

```
## [1] 37
```

```
#ns df = 29
se <- c()
for (i in 3:40){
fit <- lm(cases ~ ns(days, df = i, knots = NULL,
                     intercept = FALSE), data = catraining)
pred <- predict(fit, catest)
se[i] <- mean((pred - catest$cases)^2)
}
which.min(se)
```

```
## [1] 29
```

The optimal degree of freedom for California is 37 when using cubic splines and 29 when using natural cubic splines.

```
#split 80% training, 20% test
set.seed(8)
s <- sample(nrow(wa), floor(nrow(wa)*0.8), replace = F)
watraining <- wa[s,]
watest <- wa[-s,]

#bs df = 38
se <- c()
for (i in 3:40){
fit <- lm(cases ~ bs(days, df = i, knots = NULL, degree = 3,
                     intercept = FALSE), data = watraining)
pred <- predict(fit, watest)
se[i] <- mean((pred - watest$cases)^2)
}
which.min(se)
```

```
## [1] 38
```

```
#ns df = 36
se <- c()
```

```
for (i in 3:40){
fit <- lm(cases ~ ns(days, df = i, knots = NULL,
                    intercept = FALSE), data = watraining)
pred <- predict(fit, watest)
se[i] <- mean((pred - watest$cases)^2)
}
which.min(se)
```

```
## [1] 36
```

The optimal degree of freedom for Washington is 38 when using cubic splines and 36 when using natural cubic splines.

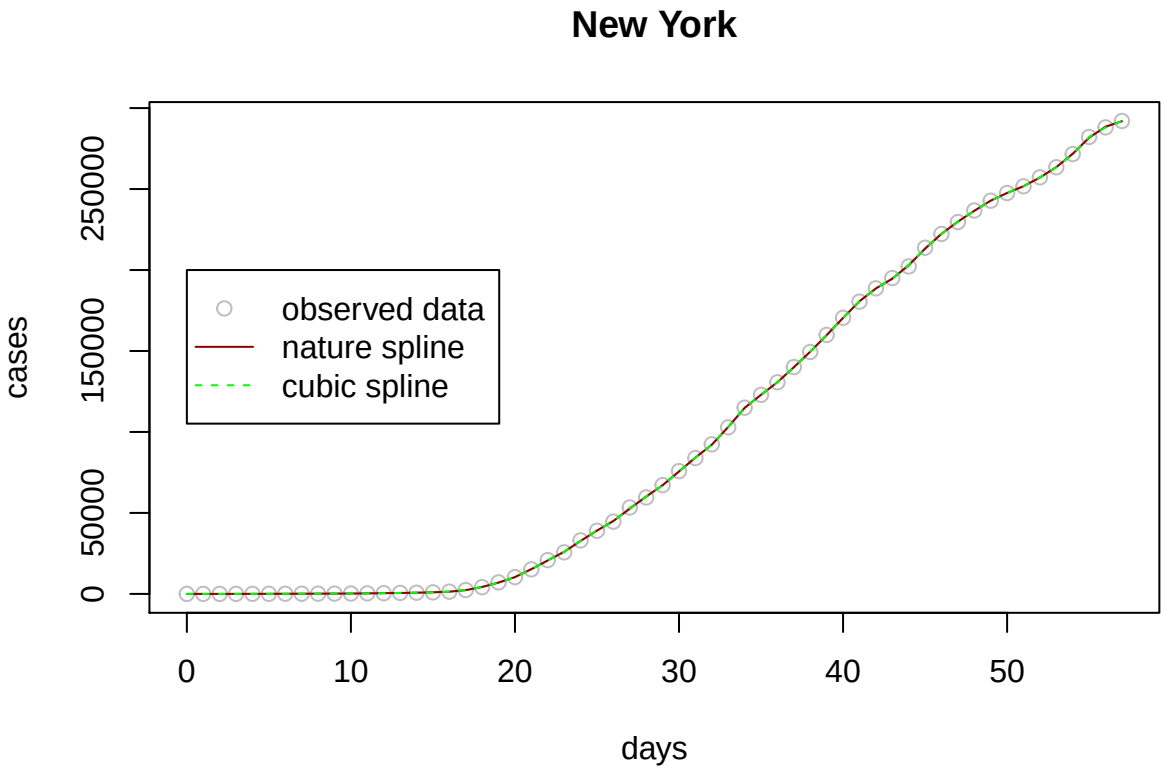
```
#ny
ny.bs <- bs(ny$days, df = 39, knots = NULL,
            degree = 3, intercept = FALSE)
ny.ns <- ns(ny$days, df = 37, knots = NULL,
            intercept = FALSE)
#ca
ca.bs <- bs(ca$days, df = 37, knots = NULL,
            degree = 3, intercept = FALSE)
ca.ns <- ns(ca$days, df = 29, knots = NULL,
            intercept = FALSE)
#washington
wa.bs <- bs(wa$days, df = 38, knots = NULL,
            degree = 3, intercept = FALSE)
wa.ns <- ns(wa$days, df = 36, knots = NULL,
            intercept = FALSE)
```

(c) Generate a plot of three panels, each corresponding to one state. Each panel plots the observed data and the fitted splines (different colors for cubic spline and NCS). Compare and commment on the two types of splines.

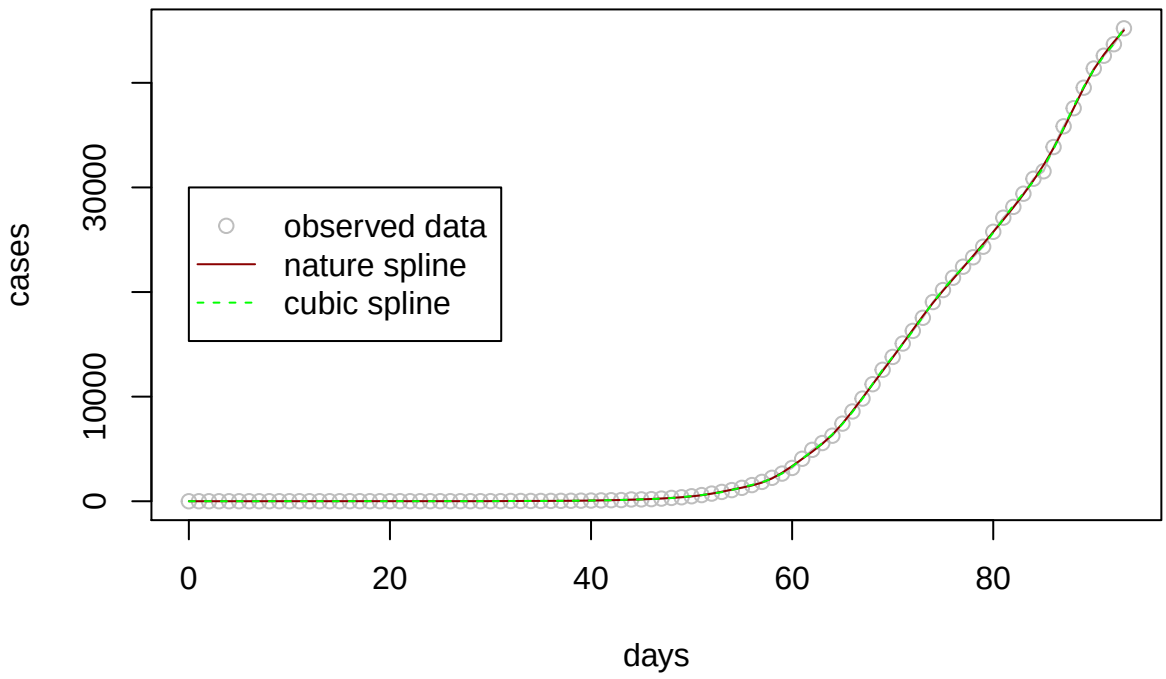
```
ny.bs.fit <- lm(cases ~ bs(days, df = 39, knots = NULL, degree = 3), data = ny)
prednybs <- predict(ny.bs.fit,data = ny)


ny.ns.fit <- lm(cases ~ ns(days, df = 37, knots = NULL,
            intercept = FALSE), data = ny)
prednyns <- predict(ny.ns.fit, data = ny)
```

```
plot(ny$days,ny$cases, type = "o",
     xlab = "days",
     ylab = "cases",
     col = "gray")
lines(ny$days,prednyns, type = "l",
     xlab = "days",
     ylab = "cases",
     col = "dark red")
lines(ny$days,prednybs, xlab = NULL, ylab = NULL,
      lty = 2, type = "l",col = "green", title("New York"))
legend(0,200000,legend=c("observed data","nature spline","cubic spline"),
```

```
       col=c("gray","dark red","green"), pch = c(1, NA, NA),
        lty=c(NA,1,2), ncol=1)
```

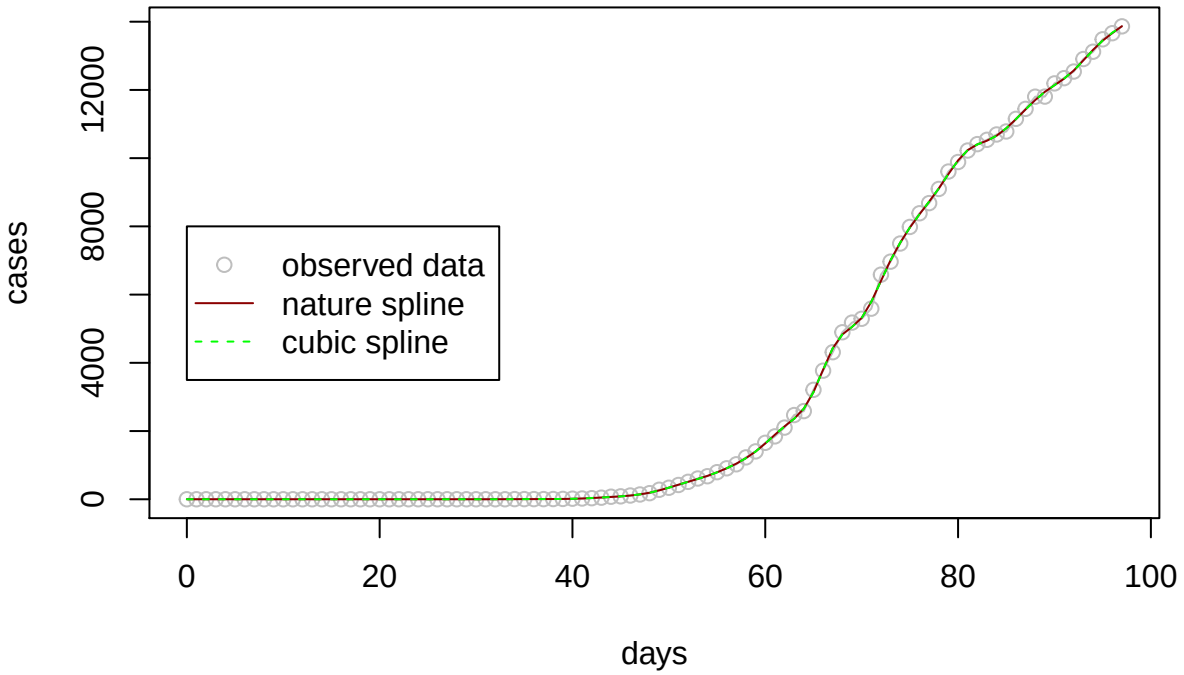



```
#new form nearly the same
ca.bs.fit <- lm(cases ~ bs(days, df = 37, knots = NULL,
            degree = 3, intercept = FALSE), data = ca)
predcabs <- predict(ca.bs.fit,data = ca)
ca.ns.fit <- lm(cases ~ ns(days, df = 29, knots = NULL,
            intercept = FALSE), data = ca)
predcans <- predict(ca.ns.fit, data = ca)
plot(ca$days,ca$cases, type = "o",
     xlab = "days",
     ylab = "cases",
     col = "gray")
lines(ca$days,predcans, type = "l",
     xlab = "days",
     ylab = "cases",
     col = "dark red")
lines(ca$days,predcabs, xlab = NULL, ylab = NULL,
      lty = 2, type = "l",col = "green", title("California"))
legend(0,30000,legend=c("observed data","nature spline","cubic spline"),
       col=c("gray","dark red","green"), pch = c(1,NA, NA),
        lty=c(NA, 1,2), ncol=1)
```




```
wa.bs.fit <- lm(cases ~ bs(days, knots = NULL, df = 38,
                           degree = 3, intercept = FALSE), data = wa)
predwabs <- predict(wa.bs.fit,data = wa)
wa.ns.fit <- lm(cases ~ ns(days, df = 36, knots = NULL,
            intercept = FALSE), data = wa)
predwans <- predict(wa.ns.fit, data = wa)
```

```
plot(wa$days,wa$cases, type = "o",
     xlab = "days",
     ylab = "cases",
     col = "gray")
lines(wa$days,predwans, type = "l",
     xlab = "days",
     ylab = "cases",
     col = "dark red")
lines(wa$days,predwabs, xlab = NULL, ylab = NULL,
      lty = 2, type = "l",col = "green", title("Washington"))
legend(0,8000,legend=c("observed data","nature spline","cubic spline"),
       col=c("gray","dark red","green"), pch = c(1,NA, NA),
        lty=c(NA,1,2), ncol=1)
```



If we specify the degree of freedom to be the optimal degree of freedom, the two lines nearly coincide. The cubic splines and natural cubic splines result in similar fitted values on these three subsets. From the plot we know the cubic splines and natural cubic splines can both describe the model pretty well on the 3 subsets.

(d) Use the fitted splines from Washington and California to predict the number of cases in New York State. Which prediction is better? Comment on why/why not the prediction is good.

```
#washington bs
predwabs <- predict(wa.bs.fit, ny)
mean((predwabs - ny$cases)^2)
```

```
## [1] 20334956650
```

```
#washington ns
predwans <- predict(wa.ns.fit, ny)
mean((predwans - ny$cases)^2)
```

```
## [1] 20334956635
```

```
#ca bs
predcabs <- predict(ca.bs.fit, newdata = ny)
mean((predcabs - ny$cases)^2)
```

```
## [1] 20297165754
```

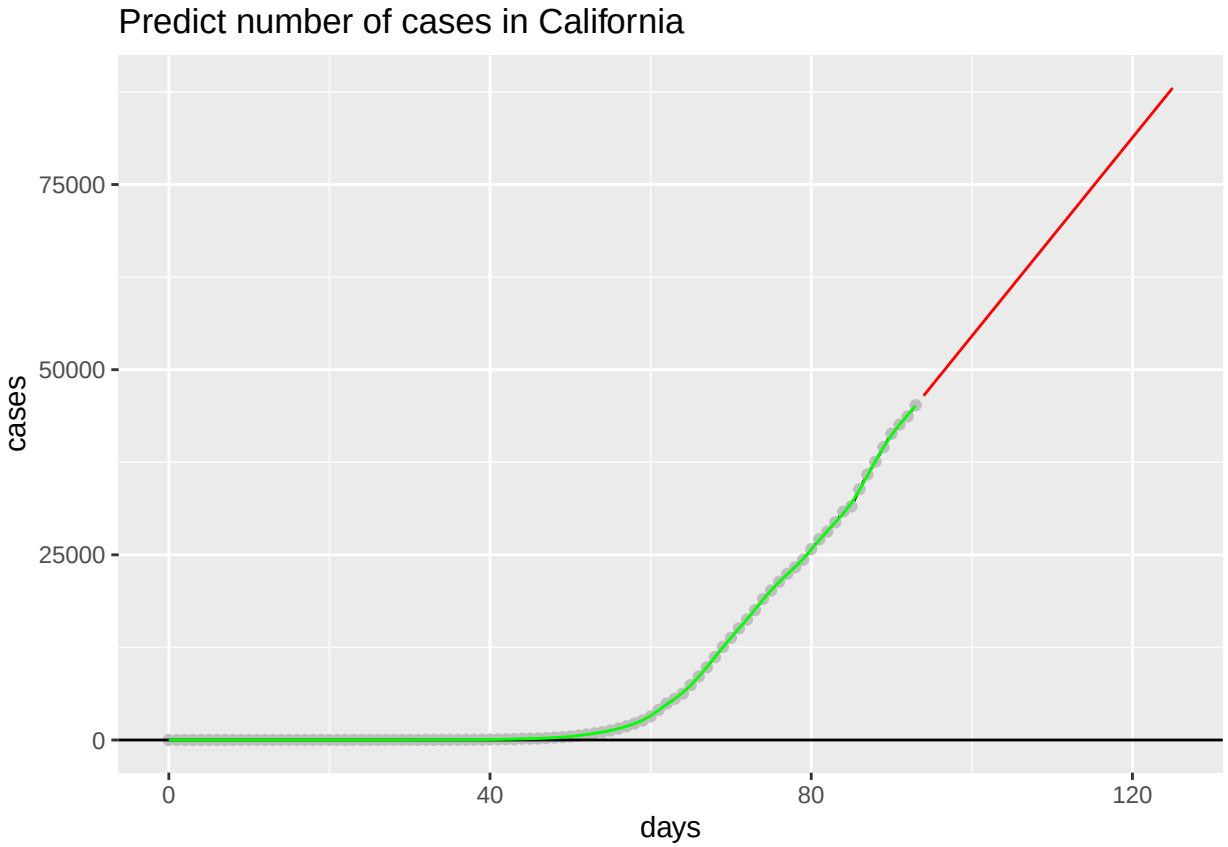
```r
#ca ns
predcans<- predict(ca.ns.fit, newdata = ny)
mean((predcans - ny$cases)^2)
```

```
## [1] 20297826747
```

Both cubic splines and natural cubic splines result in quite large and similar test error. The predictions are not good because New York has far more cases than those in California and Washington. Also, the growth rate of cases in New York is much higher than those of the other 2 states.

(e) Now use Smoothing Spline (`smooth.spline`) to fit the California data, and predict its number of cases in the next 30 days. Plot the observed data, the fitted line, and its extrapolation in a single plot. Comment on the prediction you get.

```r
## develop a model
smooth.fit <- smooth.spline(x =ca[,"days"],
                            y = ca[,"cases"], w = NULL, cv = FALSE)
cafit <- data.frame(days = ca$days)
cafitcases <- predict(smooth.fit, x = ca$days)
cafit$cases <- cafitcases$y

capred<- data.frame(days = 94:125)
predcases <- predict(smooth.fit, x = c(94:125))
capred$cases <- predcases$y

## plot the data
p1 <- ggplot(ca, aes(x = days, y=cases)) +
  geom_line() +
  geom_point(colour = "gray") +
  geom_hline(aes(yintercept=0)) +
  geom_line(color="red", data=capred) +
  geom_line(color="green", data=cafit) +
  labs(title = "Predict number of cases in California")
print(p1)
```



The gray dots are observed data, the green line is fitted line and the red line is extrapolation. In the first 50 days since the first case, the increasing rate of the number of cases is quite low. After 50 days since the first case, the number of cases increases very fast. The increasing speed are nearly the same until Day 125 .