2 A Natural Cubic Spline with K Knots is represented by K basis functions:

$$N_1(x) = 1, \quad N_2(x) = x, \quad N_{j+2}(x) = d_j(x) - d_{K-1}(x)$$

for $j = 1, \ldots, K-2$

Where

$$d_j(x) = \frac{((x-\xi_j)^+)^3 - ((x-\xi_K)^+)^3}{\xi_K - \xi_j}$$

$$\therefore \quad N_3(x) = x^2$$

$$\therefore \quad N_4(x) = x^3$$

The proposed function to solve this problem is a linear combination of K basis function.

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \sum_{j=1}^{K} b_j((x-\xi_j)^+)^3 \quad — \text{①}$$

$$((x-\xi_j)^+)^3 = \begin{cases} (x-\xi_j)^3, & \text{if } x \geq \xi_j \\ 0 & \text{if } x < \xi_j \end{cases}$$

is the truncated power function &

$$0 \leq \xi_1 < \xi_2 < \ldots < \xi_K$$

ξ_1 are order fixed values & should be pre-defined, they are called Knots & K is the number of Knots.

Next, let's see What is the first & last spline for this ordinary cubic spline.

→ first spline

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \quad \text{———} \; \textcircled{2}$$

since, Within this interval all $x < \xi_1$ then $\sum_{j=1}^{1} b_1((x-\xi)^+)^3$ will be zero.

→ last spline i.e When $x > \xi_K$ then

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \sum_{j=1}^{K} b_j ((x-\xi_j)^+)^3 \quad \text{———} \; \textcircled{3}$$

equation ③ & equation ① are same except $x > \xi_K$

Now for natural spline i.e we force ② & ③ linear

For ②, We Want it to be linear, so $a_2 = 0$ & $a_3 = 0$

If We want $f''(x)$ always be zero, We need a_2 & a_3 be zero

∴ the first spline will be

$$f(x) = a_0 + a_1 x$$

∴ the last spline will be

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + b_1 (x - \xi_1)^3 + b_2 (x - \xi_2)^3 + b_3 (x - \xi)^3 + \dots + b_K (x - \xi_k)^3$$

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \sum_{j=1}^{K} b_j (x-\xi)^3$$

$$= a_0 + a_1x + a_2x^2 + x^3\sum_{j=1}^{K} b_j - 3x^2\sum_{j=1}^{K} b_j\xi_j$$

$$+ 3x\sum_{j=1}^{K} b_j\xi_j^2 - \sum_{j=1}^{K} b_j\xi_j^3 \quad — \quad ④$$

Therefore, in order to be linear, We need

$$a_2 = a_3 = 0$$

$$\sum_{j=1}^{K} b_j = 0$$

$$\sum_{j=1}^{K} b_j\xi_j = 0$$

These restrict will leave ③ & ④ as

$$f(x) = a_0 + a_1x + 3x\sum_{j=1}^{K} b_j\xi_j^2 - \sum_{j=1}^{K} b_j\xi_j^2$$

Which is in terms of x

$$f''(x) = 0.$$

Hence proved.

Assignment 5

Code ▾

1. The data set asthma comes from a study of the relationship between childhood asthma and exposure to air pollution from concentrated animal feeding operations (CAFOs). For this problem, treat asthma (Yes/No) as following a binomial distribution given exposure, and use a smooth version of logistic regression for (a)-(c) below.

a. Choose a criterion by which to select the smoothing parameter. Plot this criterion versus the smoothing parameter and choose the optimal value for use in (b) and (c).

Hide

```
df1 = data.frame(asthma)
library(stringr)
df1$Asthma = str_replace(df1$Asthma,"Yes",'1')
df1$Asthma = str_replace(df1$Asthma,"No",'0')
df1<-df1[order(df1$Exposure),]
attach(df1)
A<-as.numeric(as.character(df1$Asthma))
E<-df1$Exposure
fitl$deviance
```

```
[1] 388.4937
```

Hide

```
library(locfit)
```

```
locfit 1.5-9.4   2020-03-24
```

Hide

```
n <- length(A)
k <- seq(10,n,by=10)    #h = k/n
cv <- mcr <- gcv <- rep(0,length(k))
for (i in 1:length(k)){
  fit <- locfit(A~lp(E,nn=k[i]/n,deg=1),family="binomial",ev=dat(cv=TRUE))
  pihat <- fitted(fit)
  yhat <- (pihat > .5)*1
  mcr[i] <- mean(abs(A-yhat))
  cv[i] <- (-2*(sum(log(pihat[A==1 ]))+sum(log(1-pihat[A==0 ])))/n)
  #  gcv[i] <- gcv(fit)["gcv"]
  gcv[i]<- (-2*n*fit$dp["lk"]/(n-fit$dp["df2"])^2)
}
```

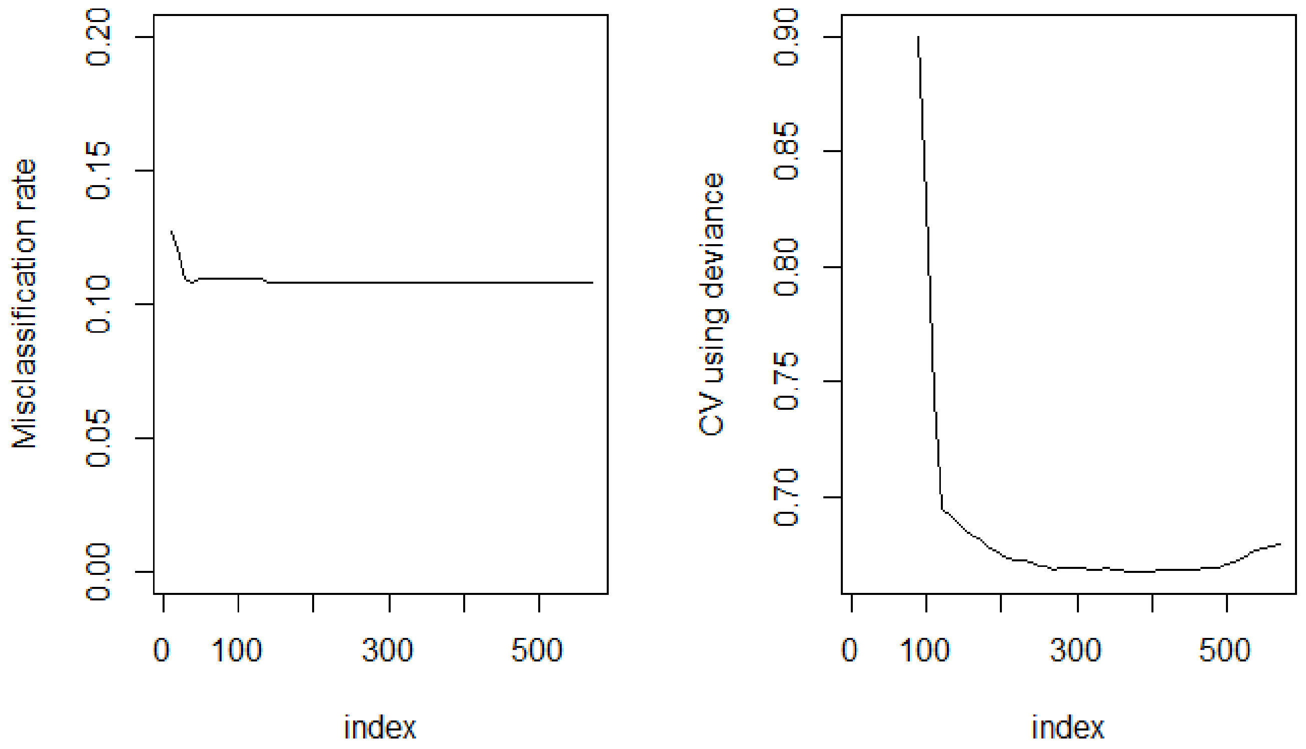
```
procv: no points with non-zero weight
```

Hide

```
par(mfrow=c(1,2))
plot(k,mcr,type="l",xlab = 'index',ylab="Misclassification rate", ylim=c(.0,.2))
plot(k,cv,type="l",xlab = 'index',ylab="CV using deviance")
```

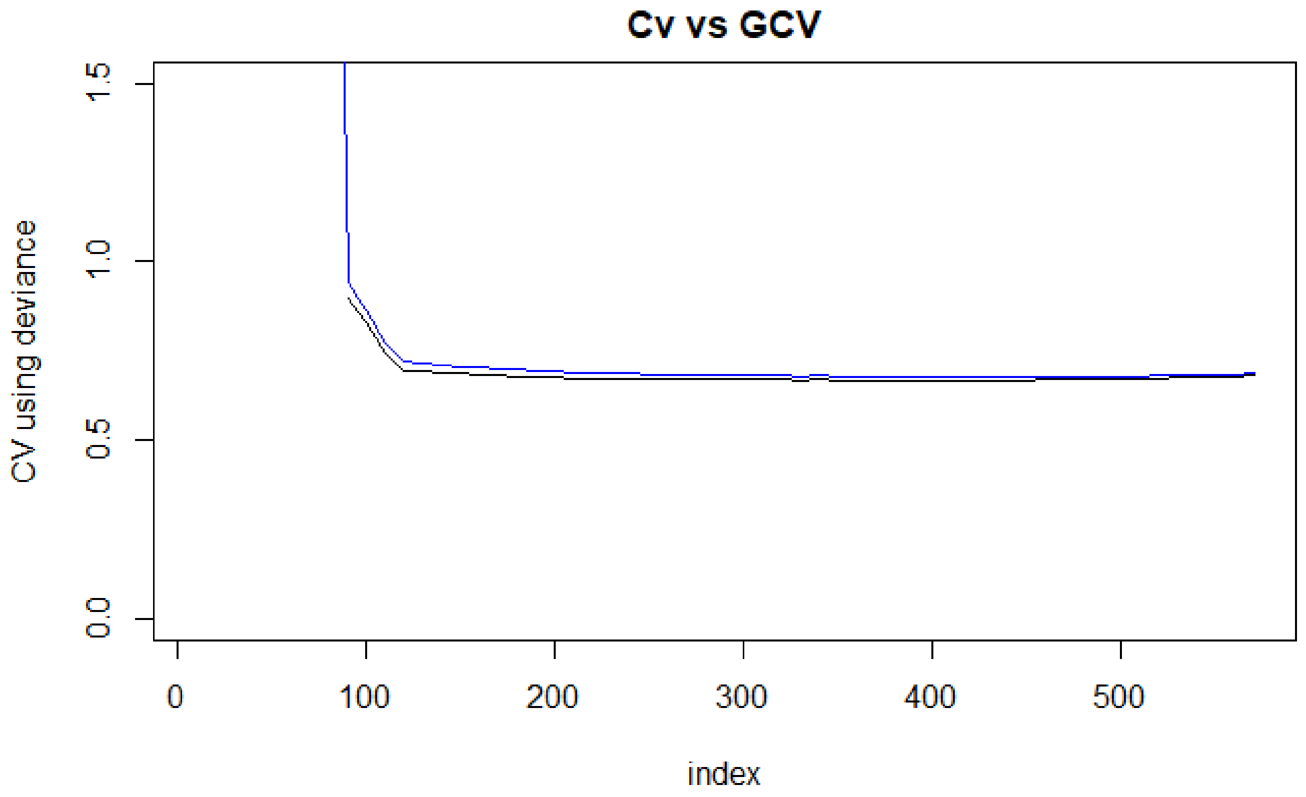
Hide

```
par(mfrow=c(1,1))
```



Hide

```
plot(k,cv,type="l",xlab = 'index',ylab="CV using deviance",main="Cv vs GCV",ylim=c(0.0,1.5))
lines(k,gcv,type="l",xlab = 'index',ylab="GCV", col='blue')
```



b. Plot a smooth curve estimating the relationship between exposure and the log-odds of developing asthma.

Hide

```
h_cv<-k[which.min(cv)]
which.min(cv)
```

```
[1] 38
```

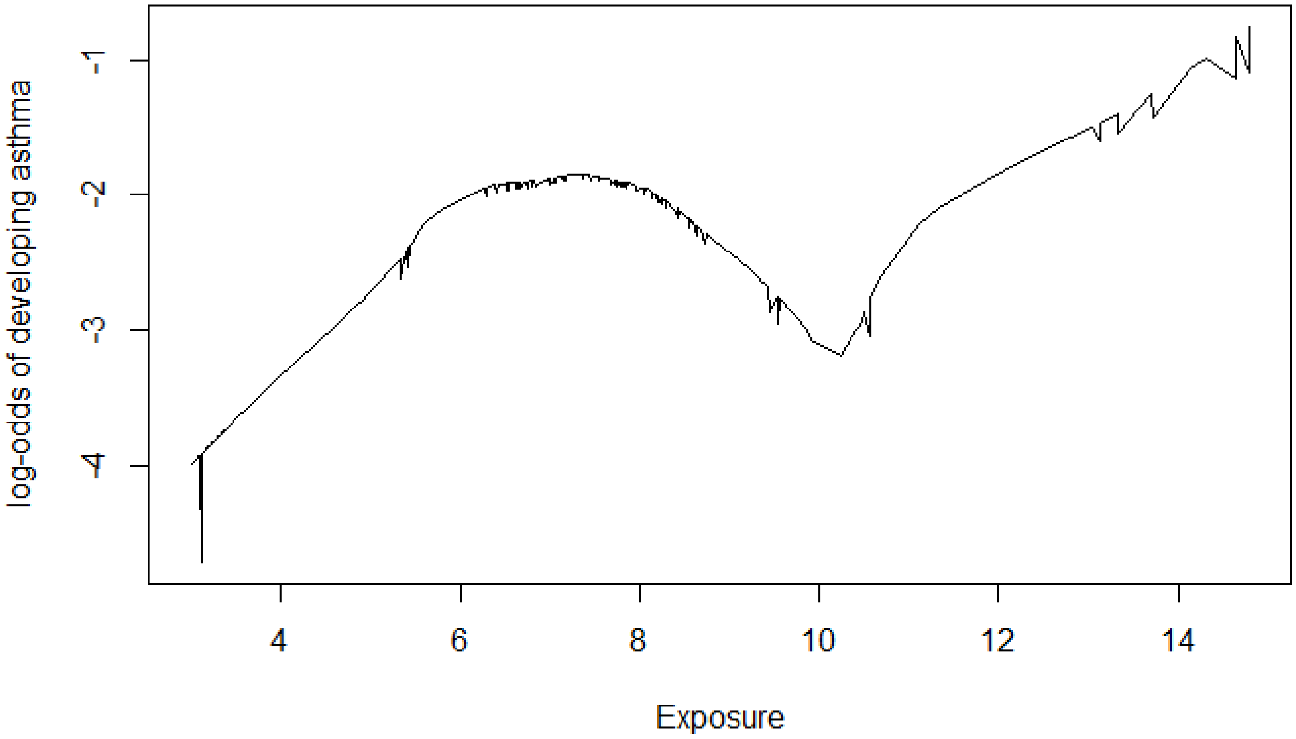
Hide

```
h_gcv<-k[which.min(gcv)]
which.min(gcv)
```

```
[1] 45
```

Hide

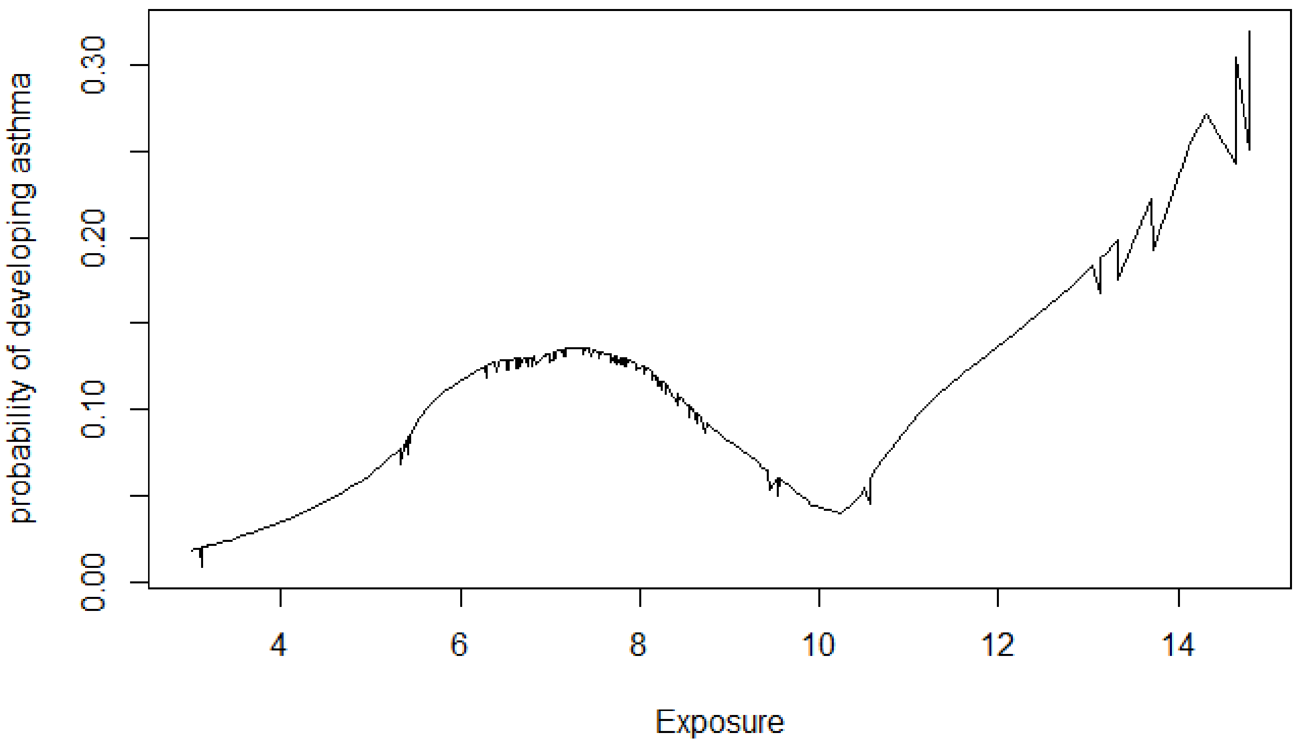
```
fit <-locfit(A~lp(E,nn=h_cv/n,deg = 1),family="binomial",ev=dat(cv=TRUE))
pihat <- fitted(fit)
plot(E,log(pihat/(1-pihat)),type = "l",xlab = 'Exposure',ylab ='log-odds of developing asthma')
```



c. Plot a smooth curve estimating the relationship between exposure and the probability of developing asthma, with confidence bands.

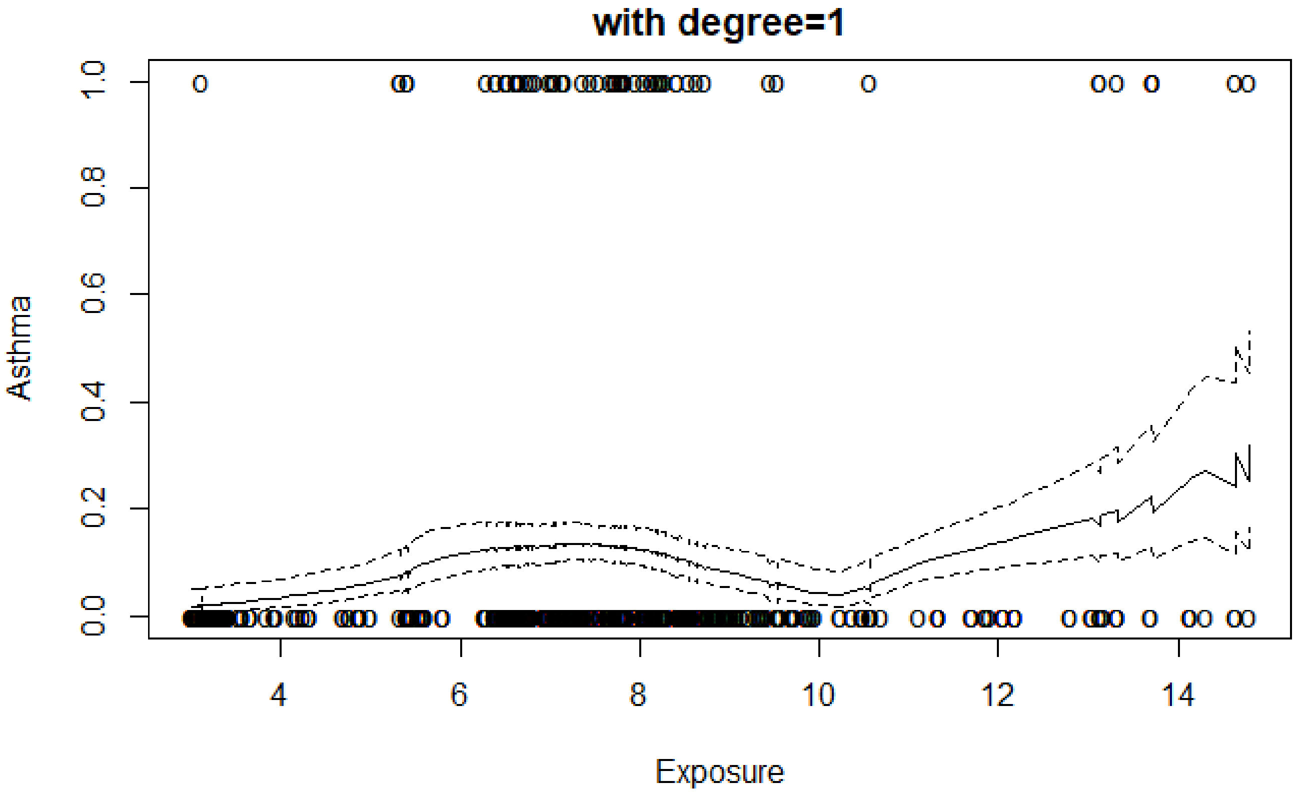
Hide

```
plot(E,fitted(fit),type="l",xlab='Exposure',ylab='probability of developing asthma')
```



Hide

```
plot(fit,band='local',get.data = T,main='with degree=1',xlab = 'Exposure',ylab='Asthma')
```



d. Prepare an ANOVA table (or rather, an analysis of deviance table) testing the sequence of models: Null ⊂ Linear ⊂ Nonlinear

Hide

```
fit1<-lm(A~1,family="binomial")
```

```
In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
 extra argument 怮牠family怮牫 will be disregarded
```

Hide

```
fit2<-lm(A~E,family="binomial")
```

```
In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
 extra argument 怮牠family怮牫 will be disregarded
```

Hide

```
fit3 <- ksmooth(A,E,kernel="normal",bandwidth=30)
anova(lm(fit3$y~fit3$x))
```

```
ANOVA F-tests on an essentially perfect fit are unreliable
```

```
Analysis of Variance Table

Response: fit3$y
           Df     Sum Sq    Mean Sq   F value    Pr(>F)
fit3$x      1 1.5087e-05 1.5087e-05 851385799 < 2.2e-16 ***
Residuals 573 0.0000e+00 0.0000e+00
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hide

```
anova(fit1,fit2)
```

```
Analysis of Variance Table

Model 1: A ~ 1
Model 2: A ~ E
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1    574 55.315
2    573 54.845  1   0.46997 4.9101 0.02709 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

3. The library ISLR contains a dataset called Wage with information about wages and demographic factors (age, gender, health, education, etc). Fit three splines to estimate the relationship between age and wage: natural base, B-spline and smooth-splines.

Hide

```
library(ISLR)
```

```

Attaching package: 㤼㸰ISLR㤼㸱

The following object is masked _by_ 㤼㸰.GlobalEnv㤼㸱:

    Wage
```

Hide

```
names(Wage)
```

```
 [1] "year"       "age"        "maritl"     "race"       "education"  "region"     "jobclass"  
 [8] "health"     "health_ins" "logwage"    "wage"      
```

Hide

```
str(Wage)
```

```
'data.frame':	3000 obs. of  11 variables:
 $ year      : int  2006 2003 2004 2009 2004 2004 2008 2006 2004 2006 ...
 $ age       : int  18 18 18 18 18 18 18 18 18 18 ...
 $ maritl    : Factor w/ 5 levels "1. Never Married",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ race      : Factor w/ 4 levels "1. White","2. Black",..: 1 2 1 1 1 1 1 1 1 1 ...
 $ education : Factor w/ 5 levels "1. < HS Grad",..: 1 2 1 1 1 2 2 1 2 3 ...
 $ region    : Factor w/ 9 levels "1. New England",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ jobclass  : Factor w/ 2 levels "1. Industrial",..: 1 1 1 1 1 1 1 1 1 2 ...
 $ health    : Factor w/ 2 levels "1. <=Good","2. >=Very Good": 1 2 2 1 1 2 2 2 2 2 ...
 $ health_ins: Factor w/ 2 levels "1. Yes","2. No": 2 1 2 2 1 2 2 2 1 2 ...
 $ logwage   : num  4.32 4.26 3.73 3.9 4.24 ...
 $ wage      : num  75 70.5 41.7 49.6 69.6 ...
```

Hide

```
Wage<-Wage[order(Wage$age),]
head(Wage)
```

	y…	…	maritl	race	education	region	jobclass
	<int>	<int>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>
231655	2006	18	1. Never Married	1. White	1. < HS Grad	2. Middle Atlantic	1. Industria
157244	2003	18	1. Never Married	2. Black	2. HS Grad	2. Middle Atlantic	1. Industria
83515	2004	18	1. Never Married	1. White	1. < HS Grad	2. Middle Atlantic	1. Industria
453584	2009	18	1. Never Married	1. White	1. < HS Grad	2. Middle Atlantic	1. Industria
85657	2004	18	1. Never Married	1. White	1. < HS Grad	2. Middle Atlantic	1. Industria
87661	2004	18	1. Never Married	1. White	2. HS Grad	2. Middle Atlantic	1. Industria

6 rows | 1-8 of 11 columns

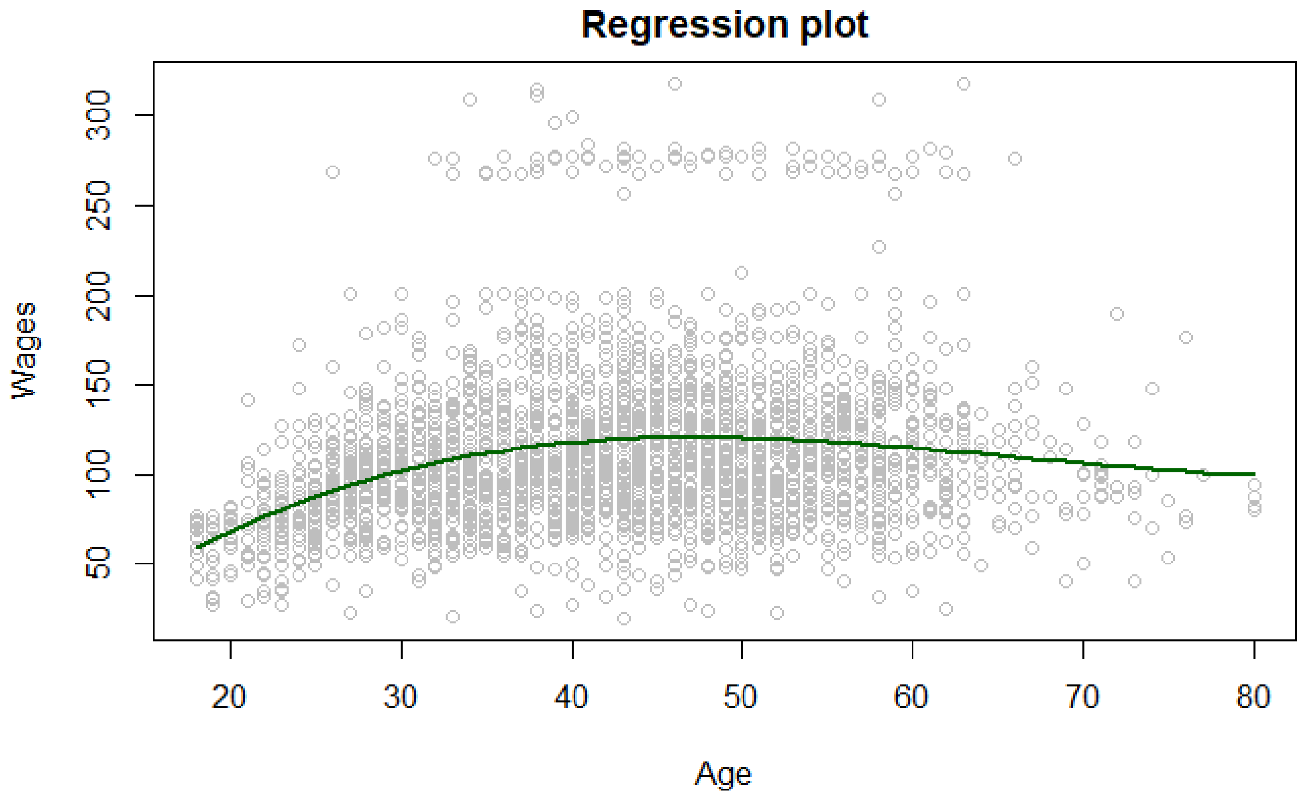
Hide

```
attach(Wage)
```

a. State the criterion by which to select the smoothing parameter and use it to select. Plot this criterion versus the smoothing parameter and choose the optimal value for use in (b) and (c).

Hide

```
#Plotting the Regression Line to the scatterplot
library(splines)
fit<-lm(Wage$wage~bs(Wage$age))
plot(Wage$age,Wage$wage,col="grey",xlab="Age",ylab="Wages",main = "Regression plot")
lines(Wage$age,predict(fit),col="darkgreen",lwd=2,type="l")
```



Hide

```
lm.fit=lm(wage~age,data=Wage)
coef(lm.fit)
```

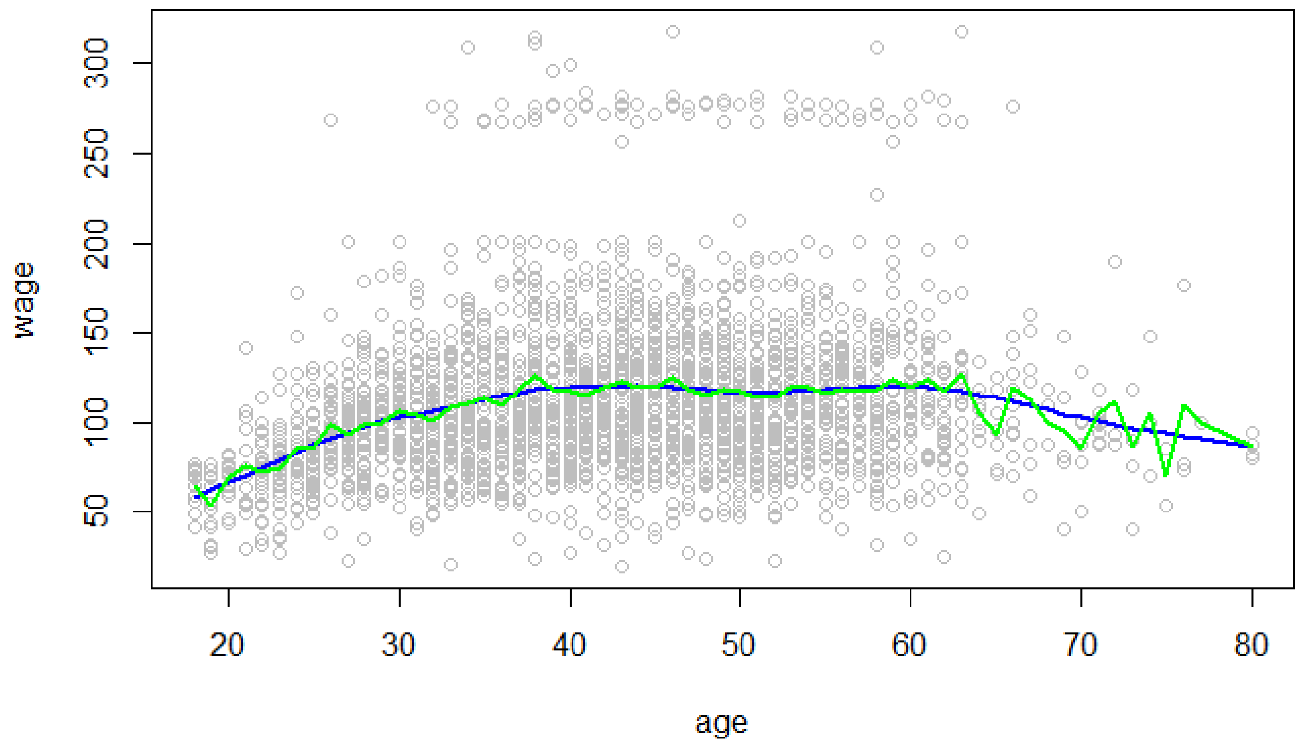
```
(Intercept)        age
 81.7047354   0.7072759
```

Hide

```
plot(age,wage,col = 'grey')
lines(smooth.spline(age,wage,df=10),lwd=2,col="blue")
```

Hide

```
lines(smooth.spline(age,wage,df=61),lwd=2,col="green")
```



Hide

```
NA
NA
```

b. Plot a smooth curve estimating the relationship between age and the wages.

Hide

```
plot(age,wage,col='grey')
spline_fit_10=smooth.spline(Wage$age,Wage$wage,df=10)
spline_fit_CV=smooth.spline(Wage$age,Wage$wage,cv=TRUE)
```

```
cross-validation with non-unique 'x' values seems doubtful
```

Hide

```
spline_fit_CV$df
```

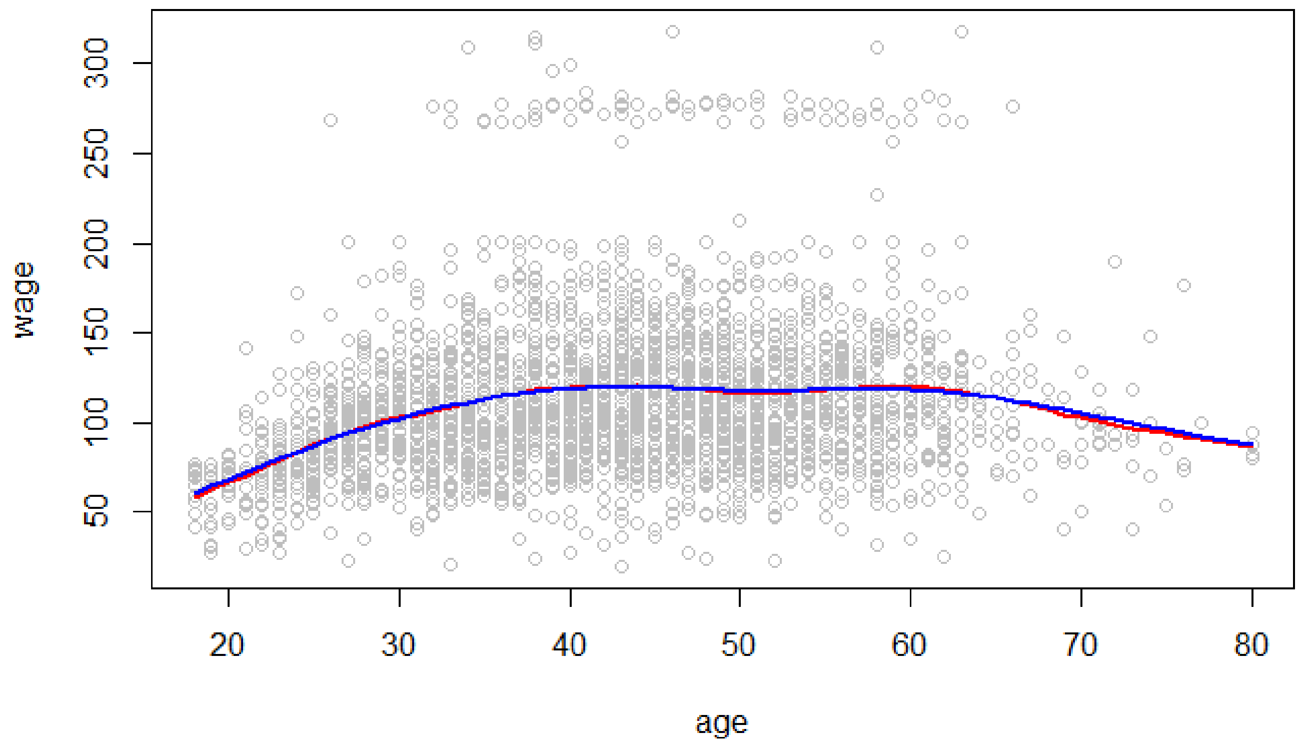
```
[1] 6.794596
```

Hide

```
lines(spline_fit_10$x,spline_fit_10$y,lwd=2,col='red')
```

Hide

```
lines(spline_fit_CV$x,spline_fit_CV$y,lwd=2,col='blue')
```



c. Compare the three models.

Hide

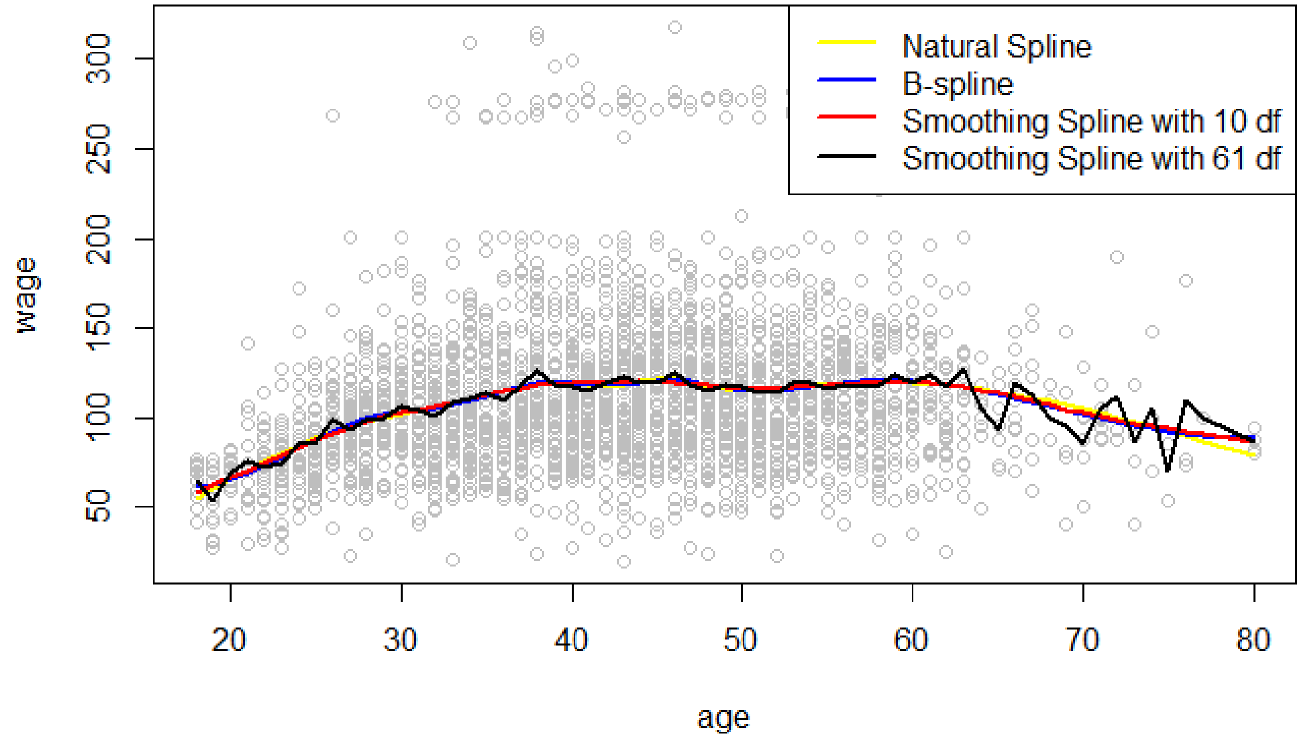
```
plot(age,wage,col='gray')
#Natural spline model
fit1<-lm(wage~ns(age,df=10))
lines(age,predict(fit1),lwd =2,col='yellow')
```

Hide

```
#B-spline model
fit2<-lm(wage~bs(age,df=10))
lines(age,predict(fit2),lwd =2,col='blue')
#Smoothing spline model
#plot(age,wage,col='gray')
lines(smooth.spline(age,wage,df=10),lwd =2,col='Red')
```

Hide

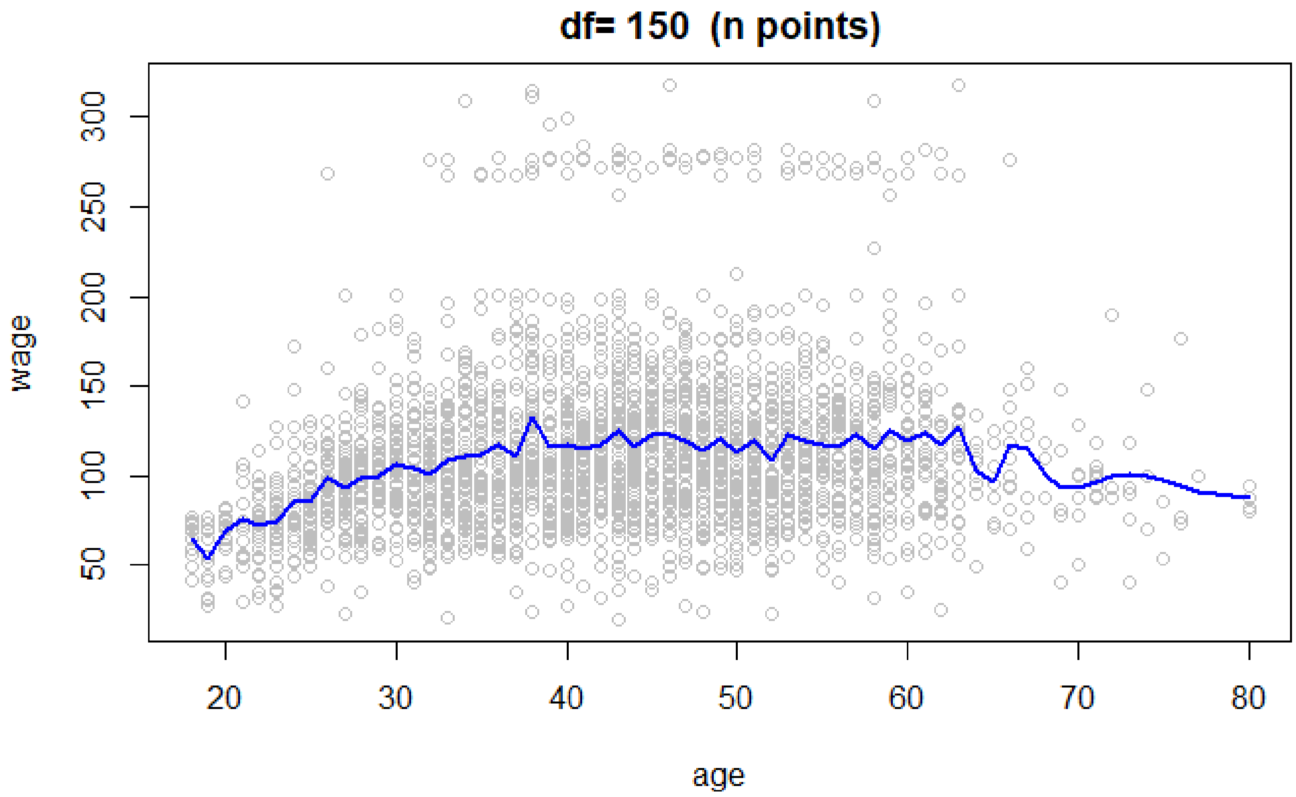
```
lines(smooth.spline(age,wage,df=61),lwd =2,col='black')
legend("topright",c("Natural Spline","B-spline","Smoothing Spline with 10 df","Smoothing Spline with 61 df"),col=c("yellow"
,"blue","red","black"),lwd=2)
```



d. Use B-splines to create 4 plots corresponding to splines with degrees of freedom = p × number of points, for the following p’s: p = 0.05, 0.5, 0.75, 0.98.

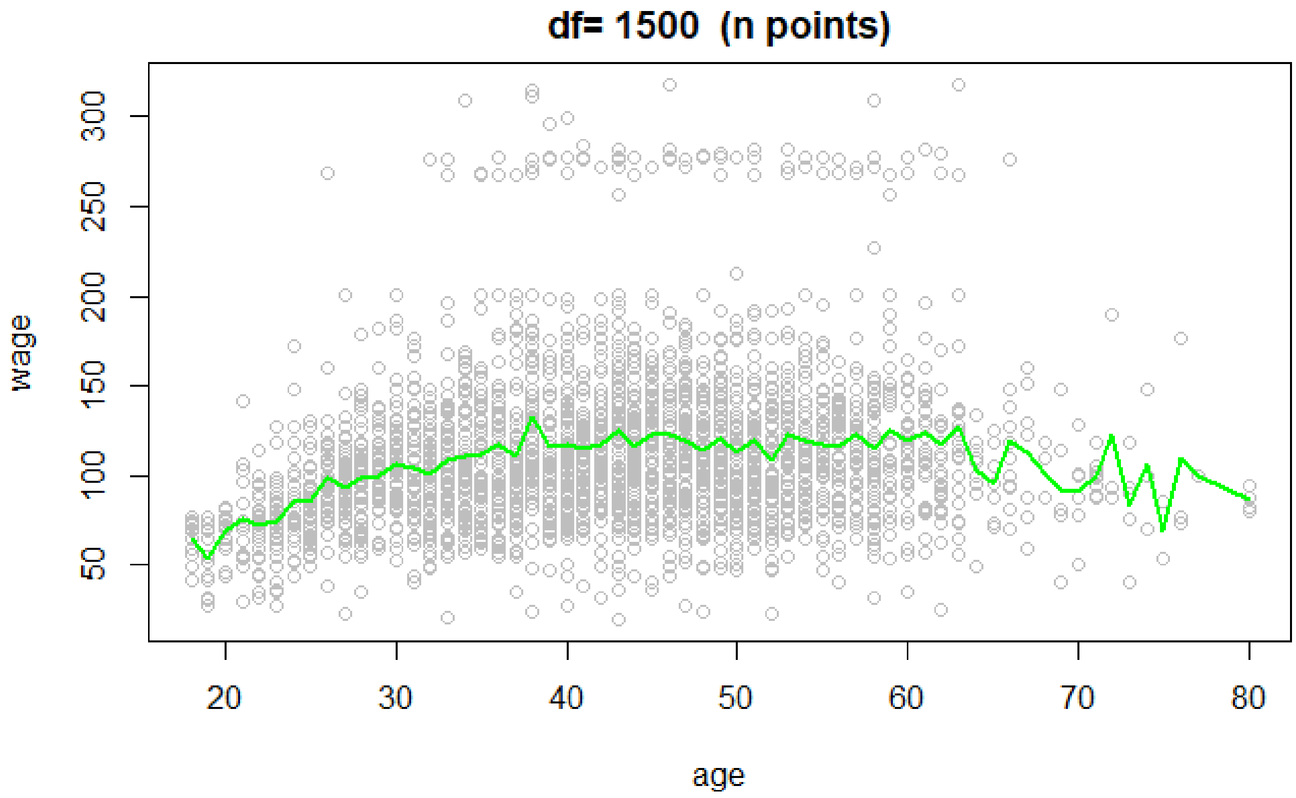
Hide

```
n=length(age)
df1=floor(.05*n)
#plot(age,wage,col="grey")
fit3<-lm(wage~bs(age,df=df1))
plot(age,wage,col="grey",main=paste('df=',df1,' (n points)'))
lines(age,predict(fit3),lwd =2,col='blue')
```


df= 150 (n points)

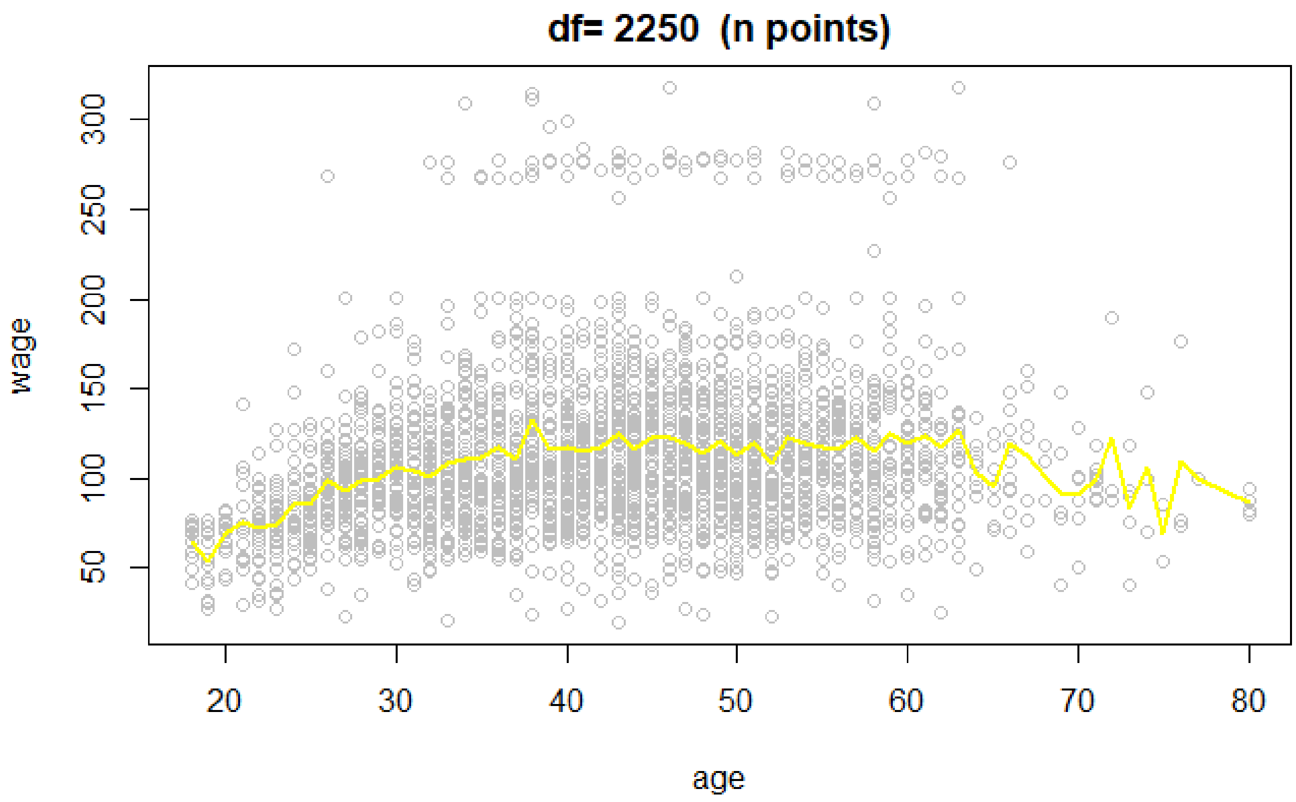
Hide

```
df2 = floor(.5*n)
fit4<-lm(wage~bs(age,df=df2))
plot(age,wage,col="grey",main=paste('df=',df2,' (n points)'))
lines(age,predict(fit4),lwd =2,col='green')
```


df= 1500 (n points)

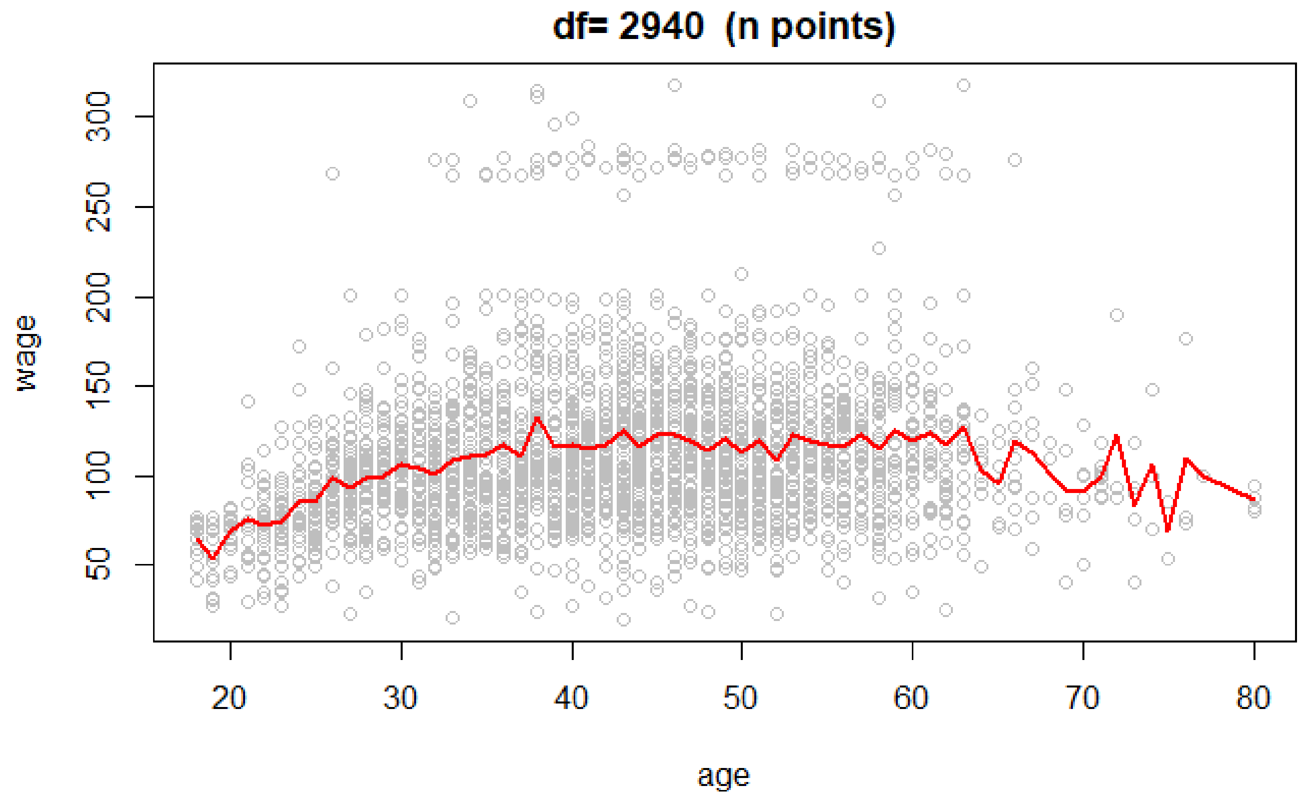
Hide

```
df3 = floor(.75*n)
fit5<-lm(wage~bs(age,df=df3))
plot(age,wage,col="grey",main=paste('df=',df3,' (n points)'))
lines(age,predict(fit5),lwd =2,col='yellow')
```



Hide

```
df4=floor(.98*n)
fit6<-lm(wage~bs(age,df=df4))
plot(age,wage,col="grey",main=paste('df=',df4,' (n points)'))
lines(age,predict(fit6),lwd =2,col='red')
```



4 (a) Write the set of truncated spline basis functions for representing a cubic spline function with three knots inside [0, 1].

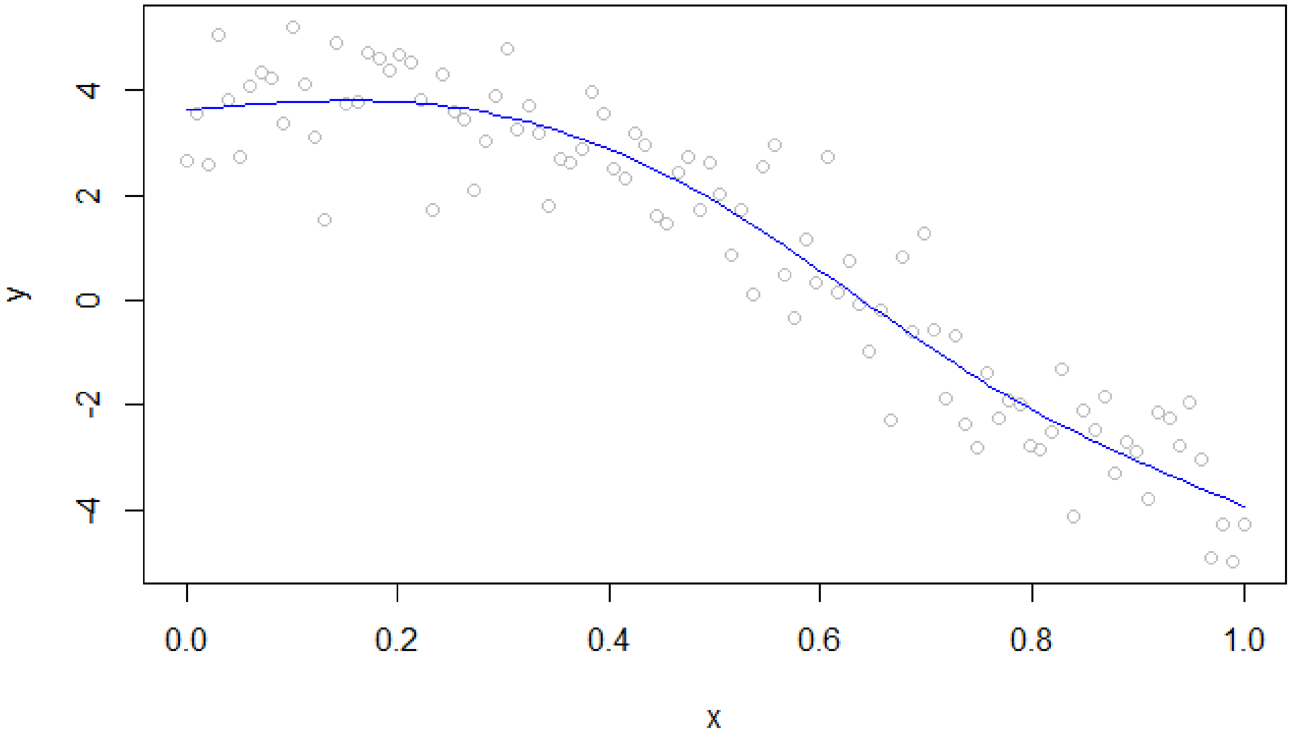
Hide

```
##
library(splines)
X <- seq(0, 1, length=100)
knots <- c(0.25, 0.50, 0.75)
n=length(x)
set.seed(1); err=1
y<- (2.3*cos(3.0*x) + 1.2*sin( 4.5*x) + cos(1.92*x)  + rnorm(n, 0, err))
```

b. For x ∈ [0, 1] and equally spaced knots, plot the above basis functions (exclude the intercept term).

Hide

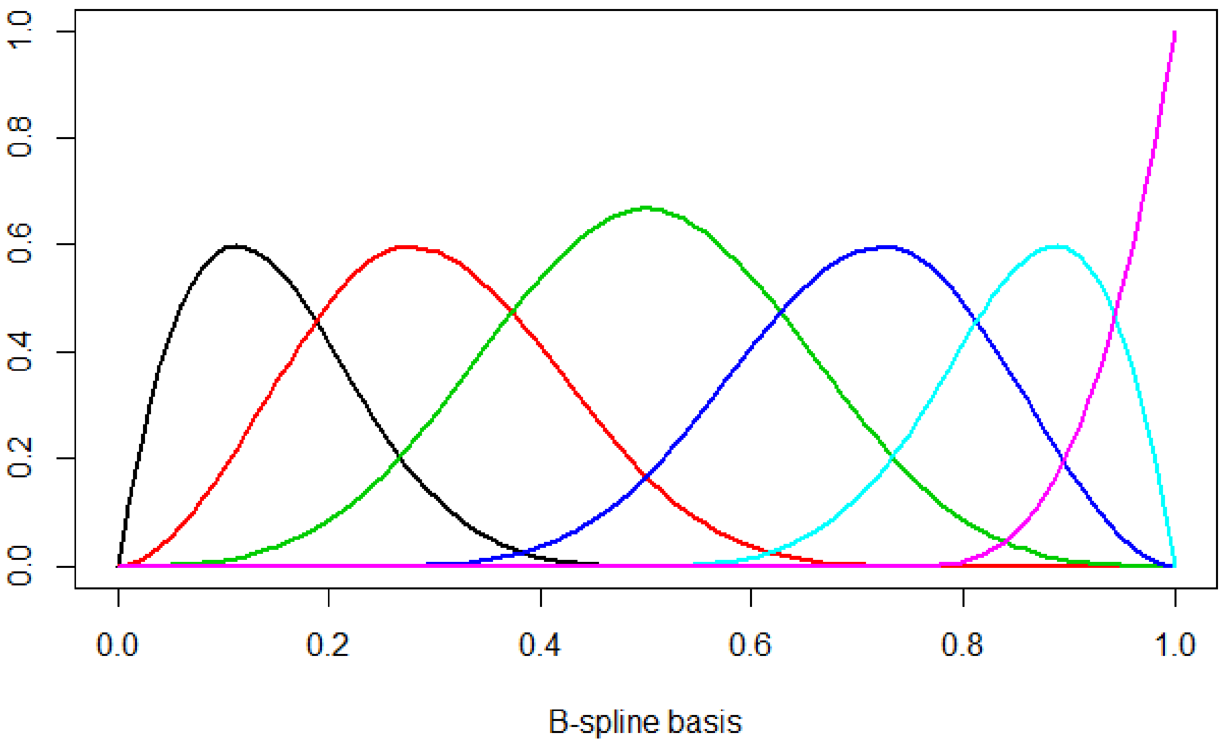
```
plot(x,y, col="grey")
lines(x, predict(lm(y~ns(x, df=10,knots=c(0.25, 0.50, 0.75),intercept=FALSE))), col='blue')
```



c. For the same range and knots as in (b), plot the B–spline basis functions, again excluding the intercept term. You may use the splines package to construct them for you.

Hide

```
library(splines)
spl <- bs(x,df=10,knots=c(0.25, 0.50, 0.75),intercept=FALSE)
plot(spl[,1]~x, ylim=c(0,max(spl)), type='l', lwd=2, col=1,
    xlab="B-spline basis", ylab="")
for (j in 2:ncol(spl)) lines(spl[,j]~x, lwd=2, col=j)
```


0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
B-spline basis