Homework #5 ARINJAY JAIN

A20447307.

Problem - 1

Ex 5.1 Show that the truncated power basis functions in (5.3) represent a basis for a cubic spline with the two knots as indicated.

Solution ⇒

from equation 5.3

$h_1(x) = 1$, $h_3(x) = x^2$, $h_5(x) = (x - \xi_1)^3_+$

$h_2(x) = x$ $h_4(x) = x^3$, $h_6(x) = (x - \xi_2)^3_+$

Total regions = 3
Parameter per region = 4
total knots = 2
Constraints per knots = 6

Now we have two knots 'ξ_1' & 'ξ_2' of a cubic spline which is cubic in the region that are

$x < \xi_1$ $\xi_1 < x < \xi_2$ $x > \xi_2$

we can write $f(x)$ as a linear combination of basis function given by eq 5.3.

$$f(x) = \sum_{m=1}^{6} \beta_m h_m(x)$$

we will see that $f(x)$ is continous at knots ξ_1 and ξ_2 and that its first and secound derivatives are also continuous at the knots.

let define 3 regions. for $f(x)$

$$\hat{f}(x) = \begin{cases} a_1 h_1(x) + b_1 h_2(x) + c_1 h_3(x)^2 + d_1 h_4(x)^3 & x \le \xi_1 \\ a_2 h_1(x) + b_2 h_2(x) + c_2 h_3(x)^2 + d_2 h_4(x)^3 & \xi_1 \le x \le \xi_2 \\ a_3 h_1(x) + b_3 h_2(x) + c_3 h_3(x)^2 + d_3 h_4(x)^3 & x \ge \xi_2 \end{cases}$$

Now solving $h_5(x)$ and $h_6(x)$ for region 1 $(x \le \xi_1)$, region 2 $(\xi_1 \le x \le \xi_2)$ and region 3 $(x \ge \xi_2)$

Let solve for region 1

$h_5(x)$ and $h_6(x)$ will not conturibute, becave

$$h_5(x) = (x - \xi_1)^3_+ = \max(x - \xi_1, 0)^3$$
$$= 0$$

$$h_5(x) = 0$$

$$h_6 = (x - \xi_2)^3_+ = \max(x - \xi_2, 0)$$
$$= 0$$

So in region 1 $x \leq \xi_1$, so $h_5(x) = 0$ & $h_6(x) = 0$

In region 2

$h_6(x)$ will not contribute, but $h_5(x)$ will contribute because in this region $x \geq \xi_1$ so $h_5(x) \neq 0$

In region 3

both $h_5(x)$ & $h_6(x)$ will contribute because $\xi_1 \leq x \leq \xi_2$

so $h_5(x) \neq 0$ & $h_6(x) \neq 0$

Assuming the cofficient of $h_5(x)$ is γ.

Now, considering the region 2 in truncated basis function from equation (4) we have

$$a_1 h_1(x) + b_2 h_2(x) + c_1 h_3(x)^2 + d_1 h_4(x)^3 + \gamma h_5(x)^3$$

$$= a_1 + b_1 x + c_1 x^2 + d_1 x^3 + \gamma(x - \xi_1)^3$$

$$= a_1 + b_1 x + c_1 x^2 + d_1 x^3 + \gamma(x^3 - \xi_1^3 - 3x^2\xi_1 + 3x\xi_1^2)$$

$$= (a_1 - \gamma\xi_1^3) + (b_1 + 3\gamma\xi_1^2) + (c_1 - 3\gamma\xi_1)x^2$$

$$+ (d_1 + \gamma)x^3$$

Note eq(1) is used to get the truncated basis function of region 2, and it should be working fine when constraints are included, if truncated basis function is equivalent to spline basis.

Now we need to compare eq(1') with the equation for region 2.

$$a_2 = a_1 - \gamma \xi_1^3$$

$$\Rightarrow a_1 - a_2 = \gamma \xi_1^3$$

$$b_2 = b_1 + 3\gamma \xi_1^2$$

$$b_1 - b_2 = -3\gamma \xi_1^2$$

$$c_2 = c_1 - 3\gamma \xi_1$$

$$\Rightarrow c_1 - c_2 = 3\gamma \xi_1$$

$$d_2 = d_1 + \gamma$$

$$\Rightarrow d_1 - d_2 = -\gamma$$

Now substituting these values in Constraint eq(2) we have:

$$a_1 + b_1 \xi_1 + c_1 \xi_1^2 + d_1 \xi_1^3 = a_2 + b_2 \xi_1 + c_2 \xi_1^2 + d_2 \xi_1^3$$

$$\Rightarrow (a_1 - a_2) + (b_1 - b_2)\xi_1 + (c_1 - c_2)\xi_1^2 + (d_1 - d_2)\xi_1^3 = 0$$

$$\Rightarrow \gamma \xi_1^3 - 3\gamma \xi_1^3 + 3\gamma \xi_1^3 - \gamma \xi_1^3 = 0$$

therefore satisfying the four conditions that cause the basis expansion to match $f(x)$ in region 2 is equivalent to satisfying the boundary constraint of the cubic spline.

Same we can do for region 3.

Therefore the truncated power basis function in 5.3 represent a basis for a cubic spline with ~~too~~ the two knots.

ARINJAY JAIN
A20447307

Homework #5

Q3 Ex. 5.14 Derive the constraints on the α_j in the thin plate spline expansion (5.39) to guarantee that the penalty $J(f)$ is finite. How see else could one ensure that the penalty was finite?

Solution. ⇒

equation 5.39

$$f(x) = \beta_0 + \beta^T x \sum_{j=1}^{N} \alpha_j h_j(x);$$

where

$$h_j(x) = \|x - x_j\|^2 \log \|x - x_j\|.$$

to find the coefficients we can use

$$\min_f \sum_{i=1}^{N} \{ y_i - f(x_i) \}^2 + \lambda J[f]$$

where

$$J[f] = \iint_{R^2} \left[\left(\frac{\partial^2 f(x)}{\partial x_1^2} \right)^2 + 2\left(\frac{\partial^2 f(x)}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f(x)}{\partial x_2^2} \right)^2 \right] \partial x_1 \partial x_2$$

Let redefine the $f(x)$

$$f(x) = \beta_0 + \beta^T x \sum_{j=1}^{N} \alpha_j (x - x_j)^2 \log (x - x_j)$$

derivation rule

$$\frac{\partial}{\partial x}(f(x)\,g(x)) = \frac{\partial}{\partial x} f(x)\cdot g(x) + \frac{\partial}{\partial x} g(x)\cdot f(x).$$

$$f'(x) = \frac{\partial f(x)}{\partial x} = 0 + \beta^T + \sum_{j=1}^{N} \alpha_j \left[2(x - x_j)\log(x - x_j) + (x - x_j)^2 \cdot \frac{1}{(x - x_j)} \right]$$

$$f'(x) = \beta^T + \sum_{j=1}^{N} \alpha_j \left[2(x - x_j)\log(x - x_j) + (x - x_j) \right]$$

$$f''(x) = 0 + \sum_{j=1}^{N} \alpha_j \left[2(1 + \log(x - x_j)) + 1 \right]$$

$$f''(x) = \sum_{j=1}^{N} \alpha_j \left[3 + 2\log(x - x_j) \right]$$

$$f''(x) = 3\sum_{j=1}^{N} \alpha_j + 2\sum_{j=1}^{N} \alpha_j \log(x - x_j)$$

the thin plate spline, always attach the finite conditions of $J(f)$

$$\sum_j \alpha_j = 0 \quad \text{and} \quad \sum_j \alpha_j x_j = 0$$

It is a extension of linear constraints of the natural cubic spline and for any cubic splines g. then $\int_{-\infty}^{\infty} g(x)'' dx$ is infinite unless g is a natural cubic s.

$$g_{jk}(X) = h_{1j}(X_1) \cdot h_{2k}(X_2) \quad j = 1, \dots M_1, \quad k = 1 \dots M_2$$

then $\int_{-\infty}^{\infty} g(x)'' dx$ is infinite unless g is a natural cubic spline.

~~After spend~~

Now we fix x_2 and then using the natural cubic spline on x_1 we obtain the coefficient vector θ_1. Because the solutions of natural cubic spline and the thin plate spline are both unique. Thus we can present $\alpha = A\theta_1$ with A is a non-~~sig~~ singular matrix.

$$\alpha = A\theta_1 \qquad \therefore A = \text{non-singular matrix.}$$

Applying linear constraints of the natural cubic spline on θ_1 we can derive linear constraints of the thin plate spline on x_1. we can do the same with X_2 we can derive the over all conditions of the thin plate spline.

$$h_j(x_i) = \|x_i - x_j\|^2 \log \|x_i - x_j\|$$

$$\frac{\partial h_j(x)}{\partial x_1} = 2(x_{i1} - x_{j1}) \log \|x_i - x_j\| + \sqrt{2}(x_{i1} - x_{j1})$$

$$\frac{\partial^2 h_j(x)}{\partial x_1^2} = 2 \log \|x_i - x_j\| + 2(x_{i1} - x_{j1}) \sqrt{2} \frac{(x_{i1} - x_{j1})}{\|x_i - x_j\|^2}$$

$$\frac{\partial h_j(x)}{\partial x_2} = 2(x_{i2} - x_{j2}) \log \|x_i - x_j\| + \sqrt{2}(x_{i2} - x_{j2})$$

$$\frac{\partial^2 h_j(x)}{\partial x_2^2} = 2 \log \|x_i - x_j\| + 2(x_{i2} - x_{j2}) \sqrt{2} \frac{(x_{i2} - x_{j2})}{\|x_i - x_j\|^2}$$

$$\frac{\partial^2 h_j(x)}{\partial x_1 \partial x_2} = \sqrt{2}(x_{i1} - x_{j1}) \, 2(x_{i2} - x_{j2})$$

So from the above equations we can conclude that the constraints on α_j are

$$\sum_{j=1}^{N} \alpha_j < \infty$$

$$\sum_{j=1}^{N} \alpha_j (x_i - x_j)^2 < \infty$$

$$\sum_{j=1}^{N} \alpha_j (x_i - x_j) \log \|x_i - x_j\| < \infty$$

we could also use smoothing spline to ensure that the penalty is finite..

Homework#5

Arinjay Jain

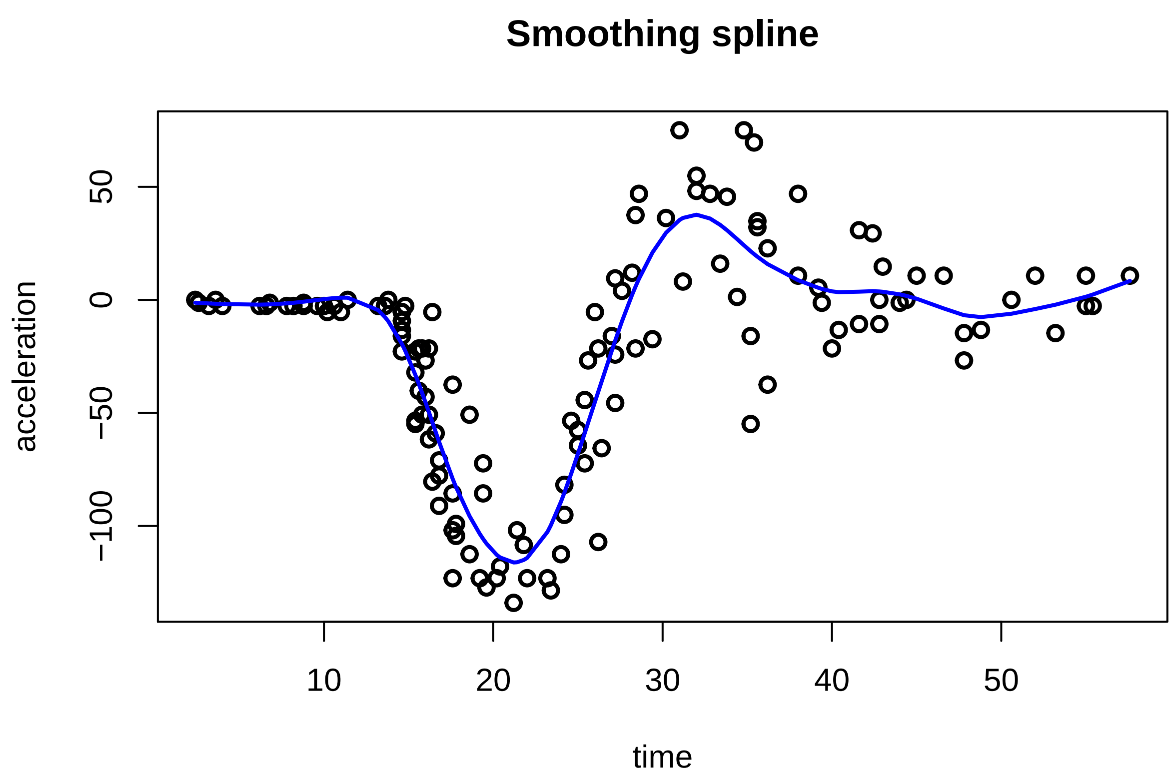
November 2, 2020

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.6.3
```

```
dataset <- mcycle

x <- dataset$times

y <- dataset$accel
```

```
##Using Smoothing spline to fit the data
plot(x,y,lwd=2,xlab='time',ylab='acceleration',main='Smoothing spline')
out = smooth.spline(x,y, cv = T)
lines(out$x, out$y,col='blue',lwd=2)
```

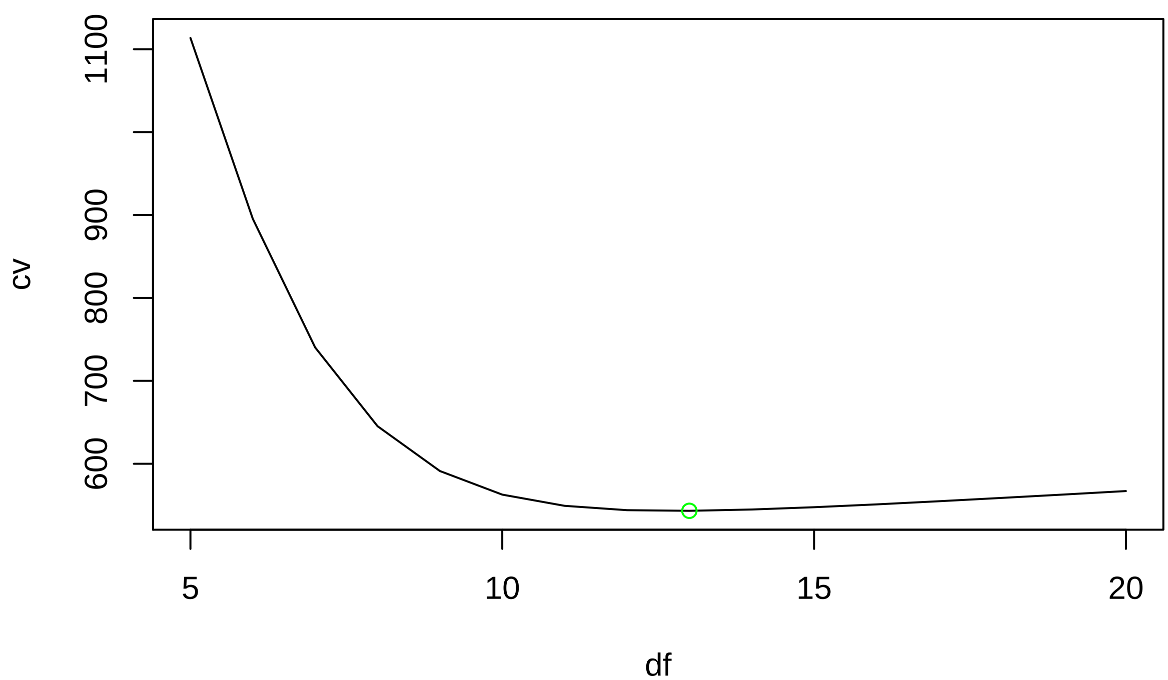


Optimal degr. of freedom

```
#
# Using CV to choose the "right" degrees of freedom
#
#n <- length(unique(x))
cv <- numeric(16)
df <- seq(5,20)
for (i in 1:16) cv[i] <- smooth.spline(x,y,df=df[i], cv = T)$cv.crit
plot(df,cv ,type="l")
cat("optimal degr. of freedom:",df[which.min(cv)]) # optimal degr. of freedom
```

```
## optimal degr. of freedom: 13
```

```
points(df[which.min(cv)], min(cv), col = "green")
```



##What is the lambda and cross-validation error of the best fit?

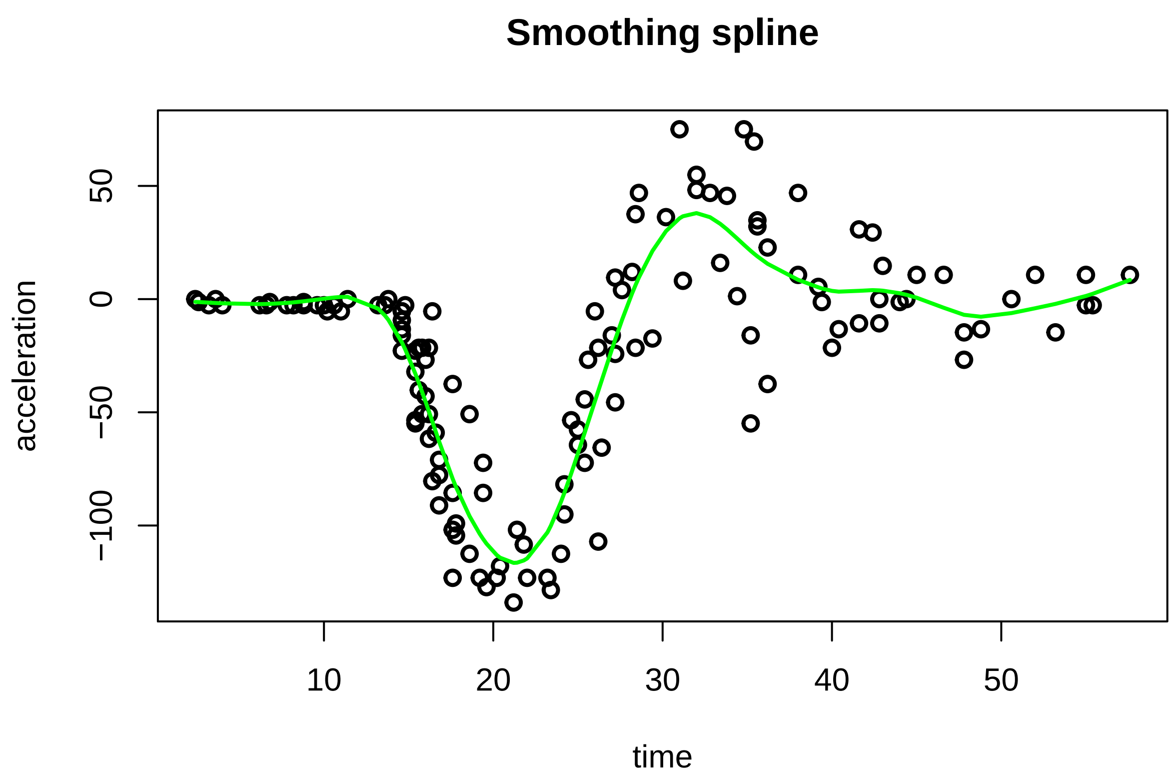
```
smooth.spline(x,y,df=13, cv = T)
```

```
## Call:
## smooth.spline(x = x, y = y, df = 13, cv = T)
##
## Smoothing Parameter  spar= 0.6429168  lambda= 8.355342e-05 (14 iterations)
```

```
## Equivalent Degrees of Freedom (Df): 13.00167
## Penalized Criterion (RSS): 37953.44
## PRESS(l.o.o. CV): 543.2597
```

Part-A

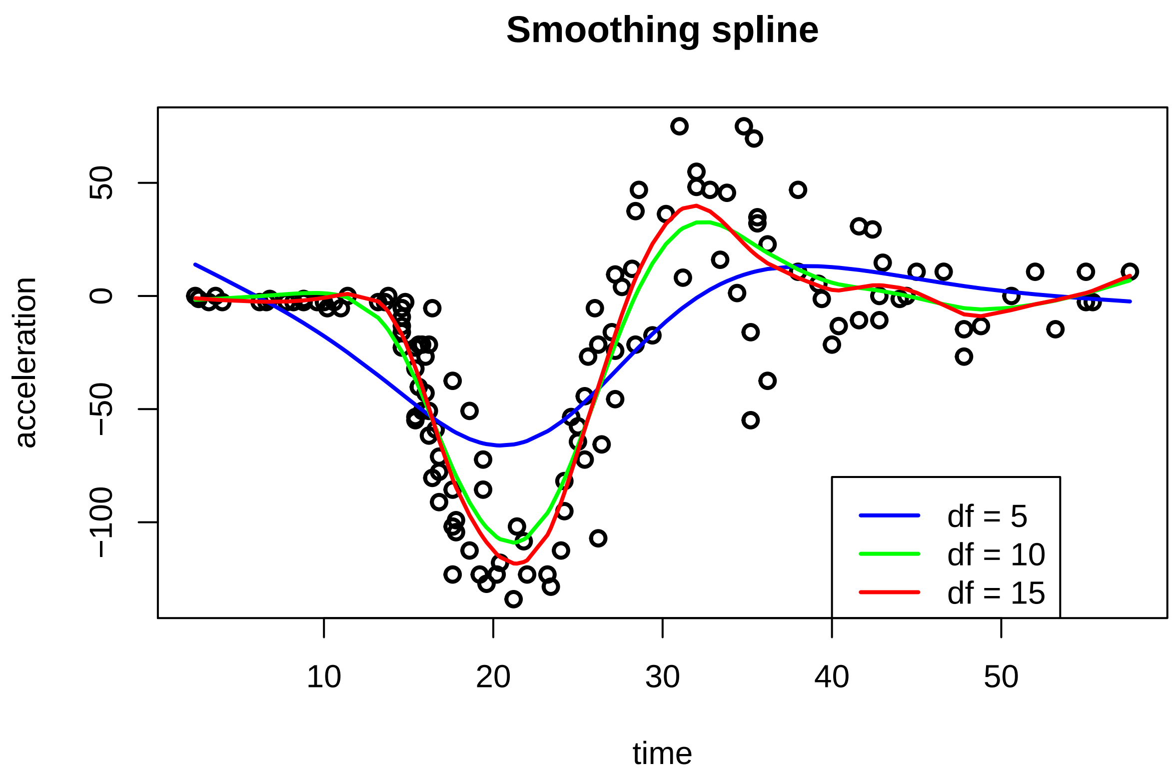
```
plot(x,y,lwd=2,xlab='time',ylab='acceleration',main='Smoothing spline')
optimal_fit = smooth.spline(x,y,df = 13, cv = T)
lines(optimal_fit$x, optimal_fit$y,col='green',lwd=2)
```



Part-B df = 5, 10, 15

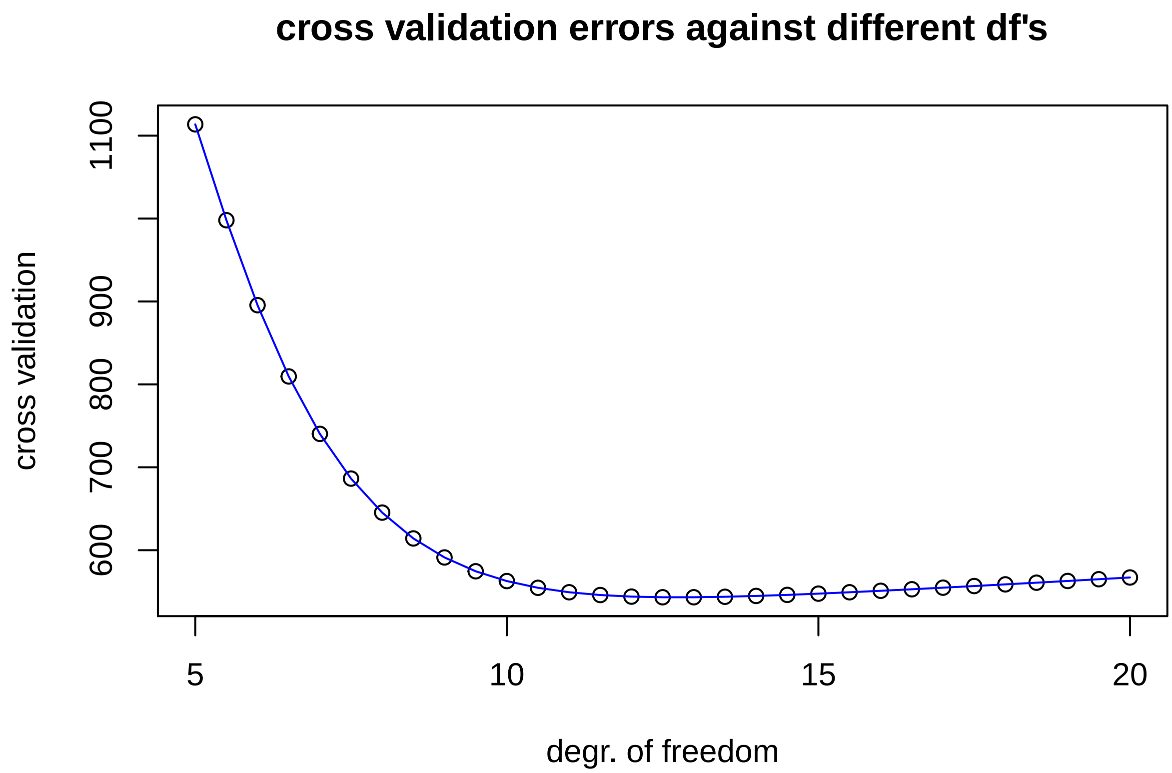
```
plot(x,y,lwd=2,xlab='time',ylab='acceleration',main='Smoothing spline')

#df = 5
df_5_fit = smooth.spline(x,y,df = 5, cv = T)
lines(df_5_fit$x, df_5_fit$y,col='blue',lwd=2)

#df = 10
df_10_fit = smooth.spline(x,y,df = 10, cv = T)
```

```
lines(df_10_fit$x, df_10_fit$y,col='green',lwd=2)

#df = 15
df_15_fit = smooth.spline(x,y,df = 15, cv = T)
lines(df_15_fit$x, df_15_fit$y,col='red',lwd=2)

legend(40,-80,legend=c("df = 5", "df = 10", "df = 15"),
col=c("blue", "green", "red"),lwd=2)
```



Part-C

```
cvs <- c()
df <- seq(5,20, by = 0.5)
for (i in 1:length(df)) cvs <- append(cvs,smooth.spline(x,y,df=df[i], cv = T)$cv)
plot(df,cvs,xlab='degr. of freedom',ylab='cross validation',main="cross validation errors against diffe
lines(df, cvs, col='blue')
```


cross validation errors against different df's
cross validation
1100
900
800
700
600
5
10
15
20
degr. of freedom