Homework#5

Arinjay Jain

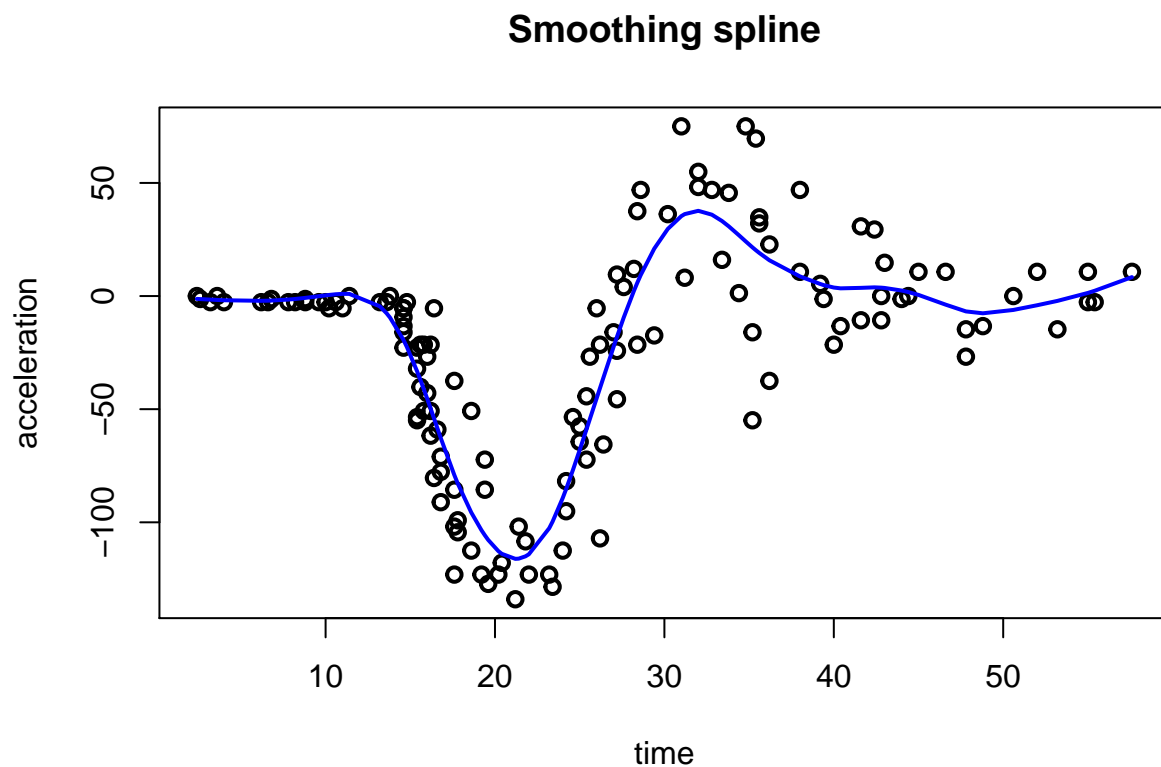### November 2, 2020

```r
library(MASS)
```

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

```
dataset <- mcycle

x <- dataset$times

y <- dataset$accel
```

```r
##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)
```
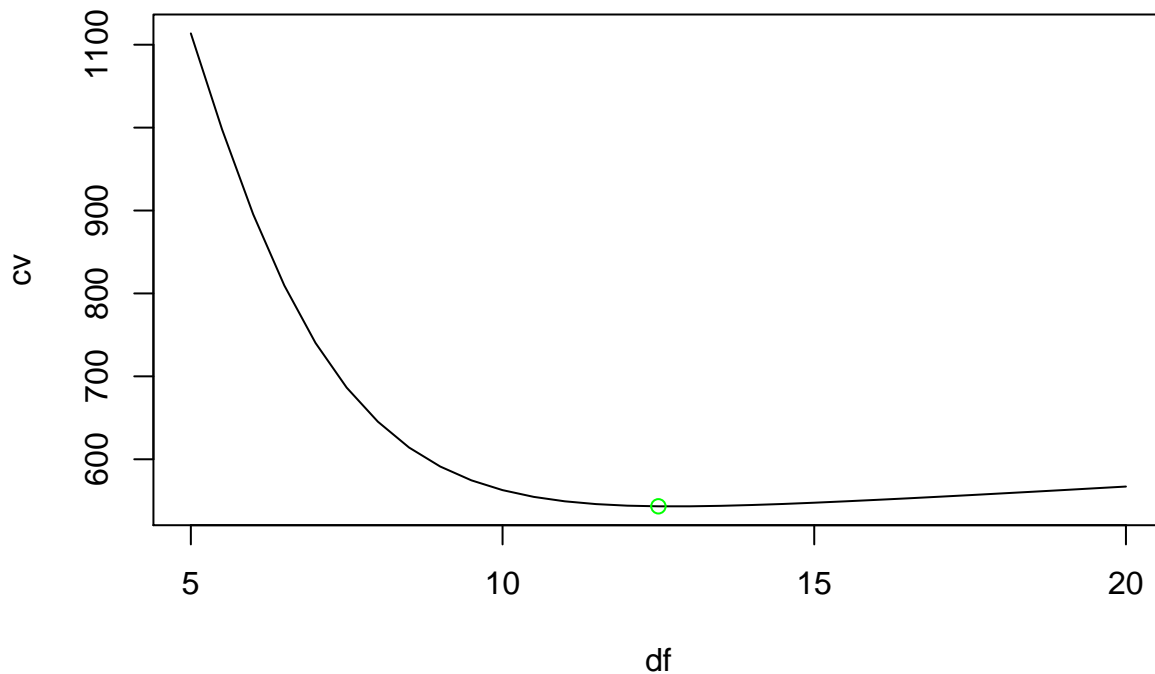
Optimal degr. of freedom with df 0.5 step

```
#
# Using CV to choose the "right" degrees of freedom
#
#n <- length(unique(x))
cv <- numeric(31)
df <- seq(5,20, by = 0.5)
for (i in 1:length(df)) 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: 12.5
```

```
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=12.5, cv = T)
```
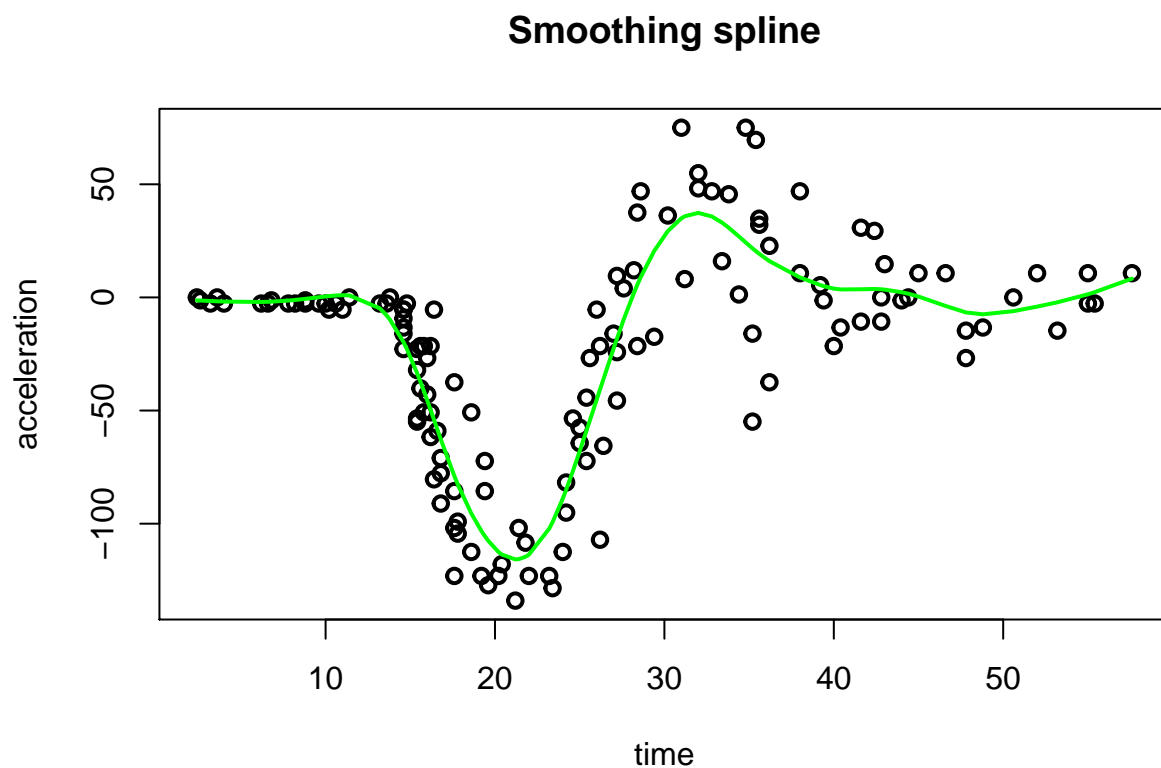
```
## Call:
## smooth.spline(x = x, y = y, df = 12.5, cv = T)
##
## Smoothing Parameter  spar= 0.6534782  lambda= 9.971239e-05 (14 iterations)
```

```
## Equivalent Degrees of Freedom (Df): 12.49818
## Penalized Criterion (RSS): 38370.84
## PRESS(l.o.o. CV): 543.2575
```

Part-A

```r
plot(x,y,lwd=2,xlab='time',ylab='acceleration',main='Smoothing spline')
optimal_fit = smooth.spline(x,y,df = 12.5, cv = T)
lines(optimal_fit$x, optimal_fit$y,col='green',lwd=2)
```


Part-B df = 5, 10, 15

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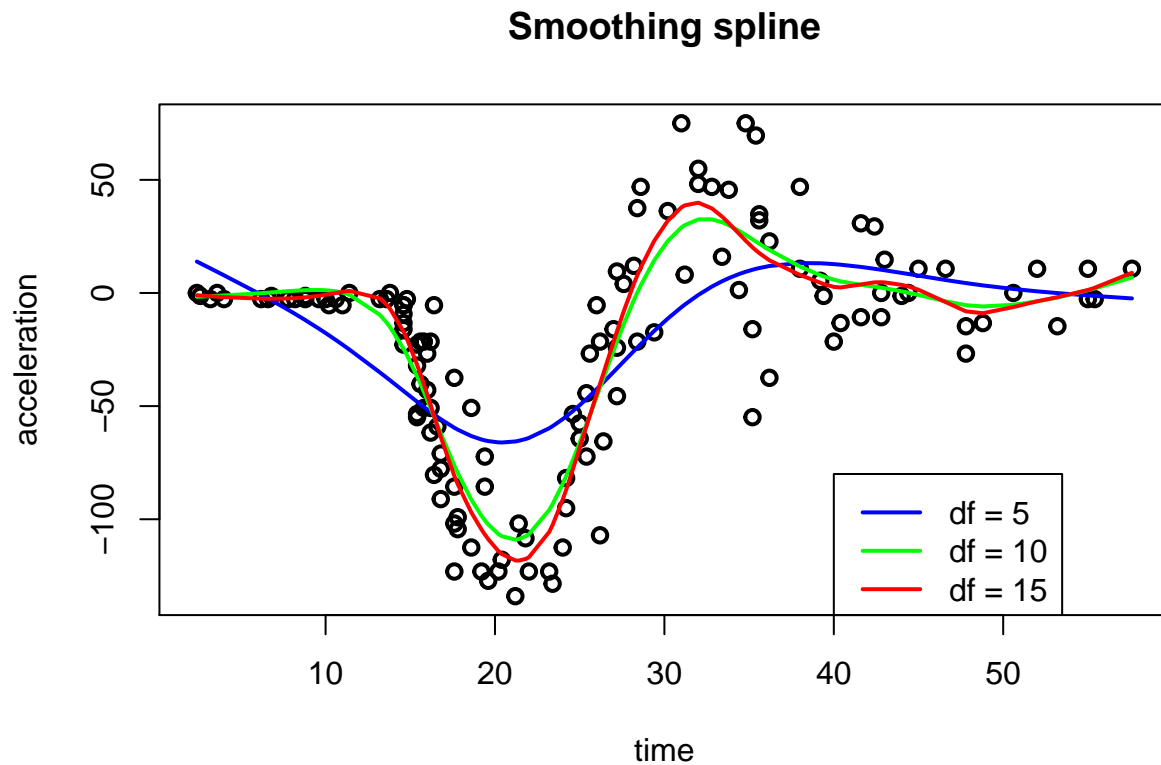
```
lines(df_10_fit$x, df_10_fit$y,col='green',lwd=2)

#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 differ
lines(df, cvs, col='blue')
```

cross validation errors against different df's

cross validation

degr. of freedom