# Lab 9

*Attila Lazar*

*09.12.2020*

## 1)

### data

We load the dataset, oder it by the temperature and define a train and a test set.

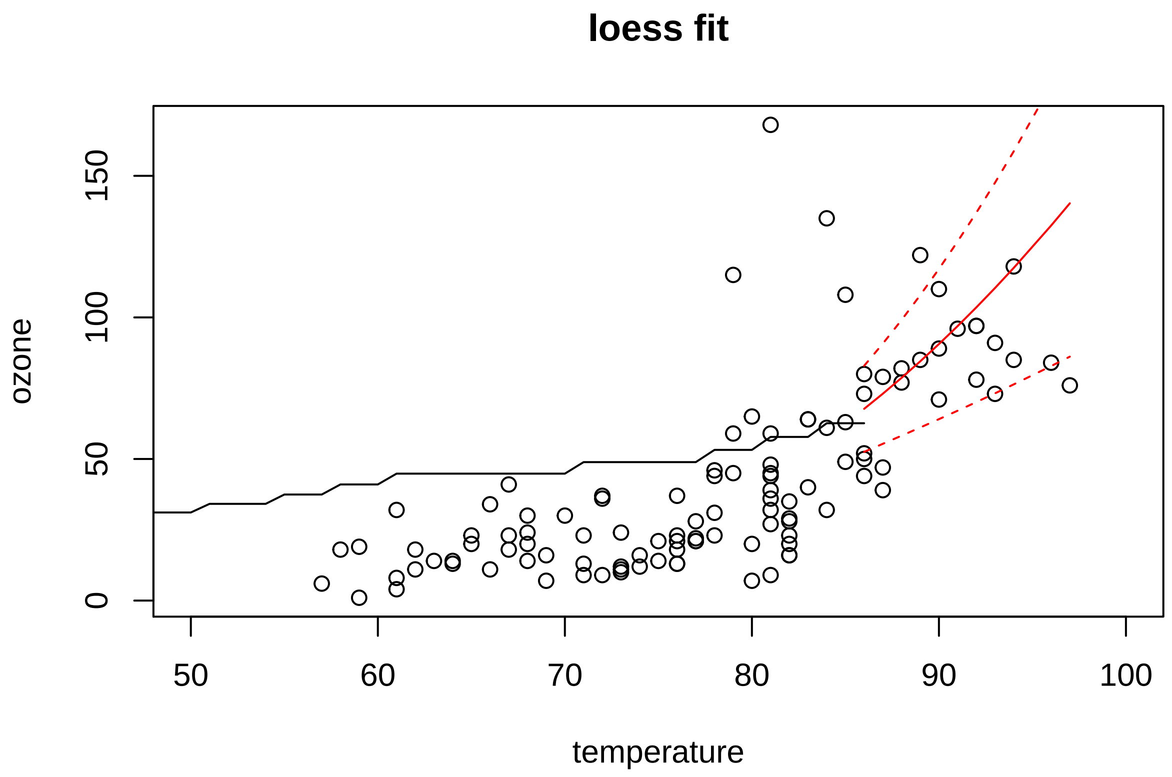
```
data(environmental, package="lattice")
str(environmental)
```

```
## 'data.frame':    111 obs. of  4 variables:
##  $ ozone      : num  41 36 12 18 23 19 8 16 11 14 ...
##  $ radiation  : num  190 118 149 313 299 99 19 256 290 274 ...
##  $ temperature: num  67 72 74 62 65 59 61 69 66 68 ...
##  $ wind       : num  7.4 8 12.6 11.5 8.6 13.8 20.1 9.7 9.2 10.9 ...
```

```
data <- environmental[order(environmental$temperature),]
train <- which(data$temperature <= 85)
test <- (1:nrow(data)) [-train]
```

We fit a model using *loess*. With red lines we show the extrapolation for values over 85 °F

```
plot(data$temperature, data$ozone, xlim=c(50, 100), main="loess fit", xlab="temperature", ylab="ozone")

smooth <- loess(ozone~temperature, data = data, subset = train, control = loess.control(surface = "dire
lines(smooth$fitted)

pred1 <- predict(smooth, data[test,'temperature'], se = TRUE)
lines(data[test,'temperature'], pred1$fit, col = 'red')
lines(data[test,'temperature'], pred1$fit + 2 * pred1$se.fit, lty = 2, col = 'red')
lines(data[test,'temperature'], pred1$fit - 2 * pred1$se.fit, lty = 2, col = 'red')
```



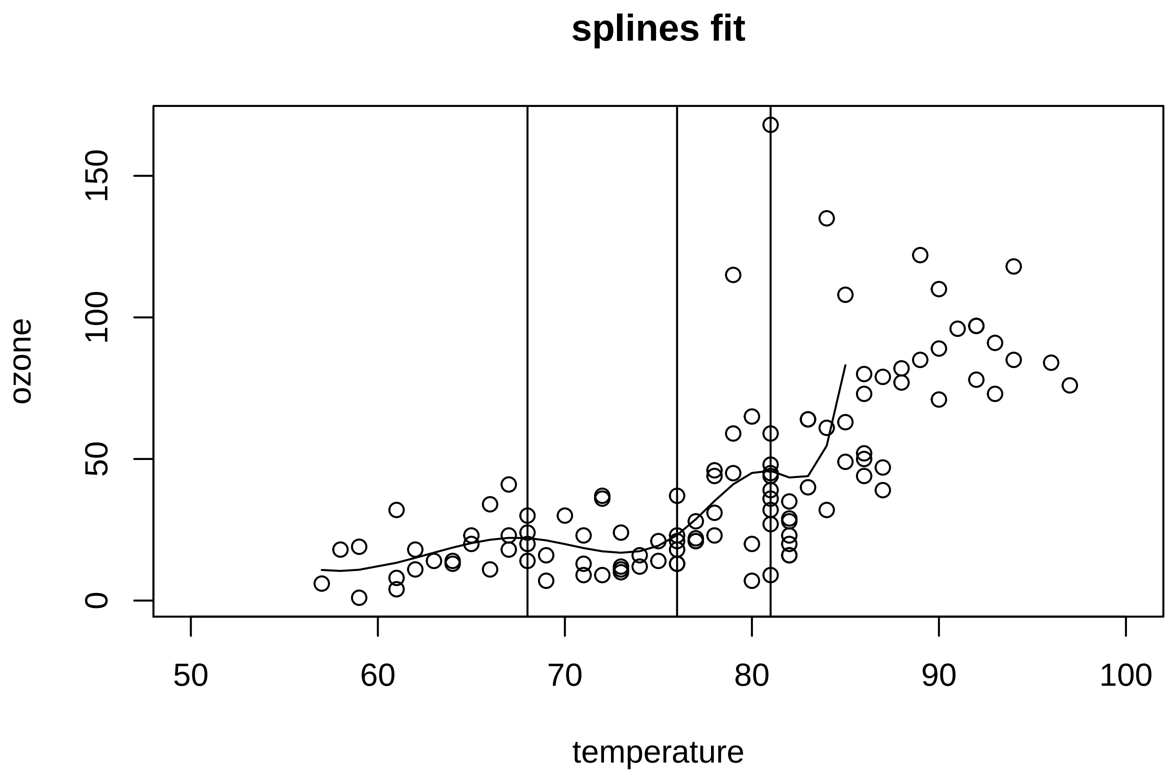
# 2)

We define a function for computing the basis functions

```r
lecturespl <- function(x, nknots=2, M=4){
    # nknots ... number of knots -> placed at regular quantiles
    # M ... M-1 is the degree of the polynomial
    n <- length(x)
    # X will not get an intercept column
    X <- matrix(NA,nrow=n,ncol=(M-1)+nknots)
    for (i in 1:(M-1)){ X[,i] <- x^i }
    # now the basis functions for the constraints:
    quant <- seq(0,1,1/(nknots+1))[c(2:(nknots+1))]
    qu <- quantile(x,quant)
    for (i in M:(M+nknots-1)){
      X[,i] <- ifelse(x-qu[i-M+1]<0,0,(x-qu[i-M+1])^(M-1))
    }
    list(X=X,quantiles=quant,xquantiles=qu)
}
```

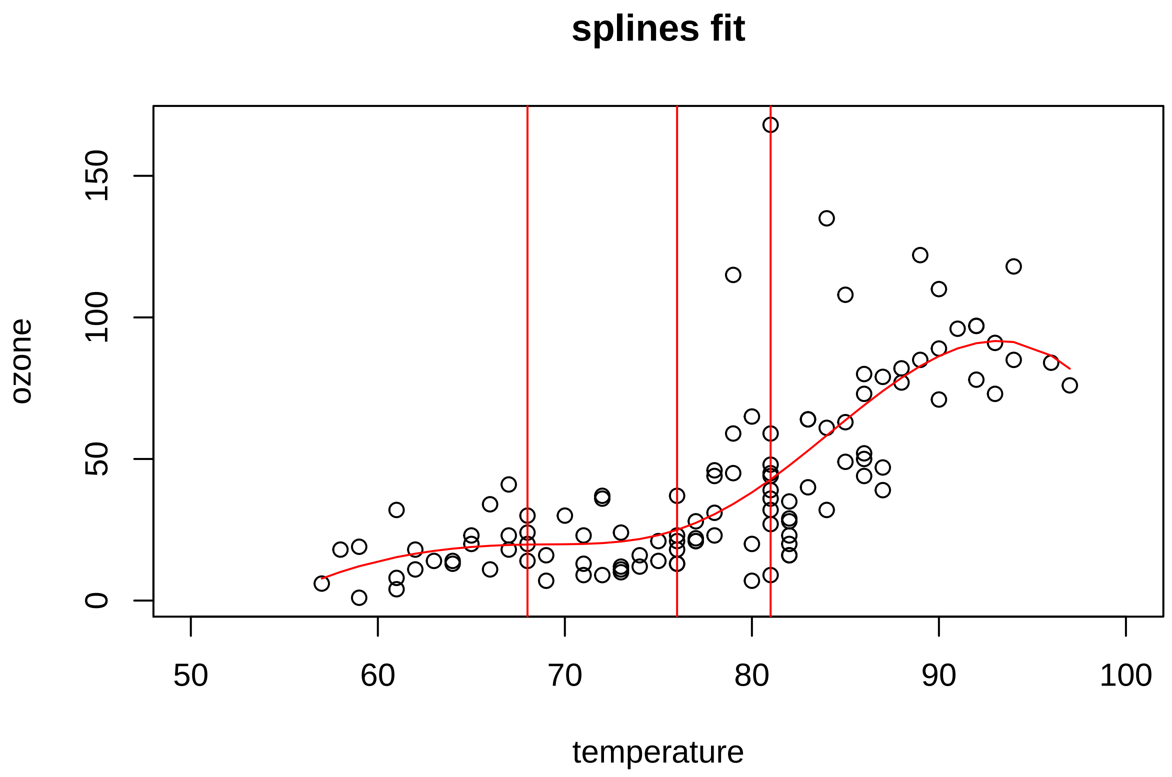
We fit a linear model to the basis functions with 3 knots. The vertical lines show the knots at the quantiles

```r
plot(data$temperature, data$ozone, xlim=c(50, 100), main="splines fit", xlab="temperature", ylab="ozone

spl1 <- lecturespl(data[train, 'temperature'], nknots=3, M=4)
ozone <- data$ozone[train]
lm1 <- lm(ozone~spl1$X)
lines(data$temperature[train],predict(lm1, newdata=data.frame(data$temperature[train])))
abline(v=spl1$xquantiles)
```



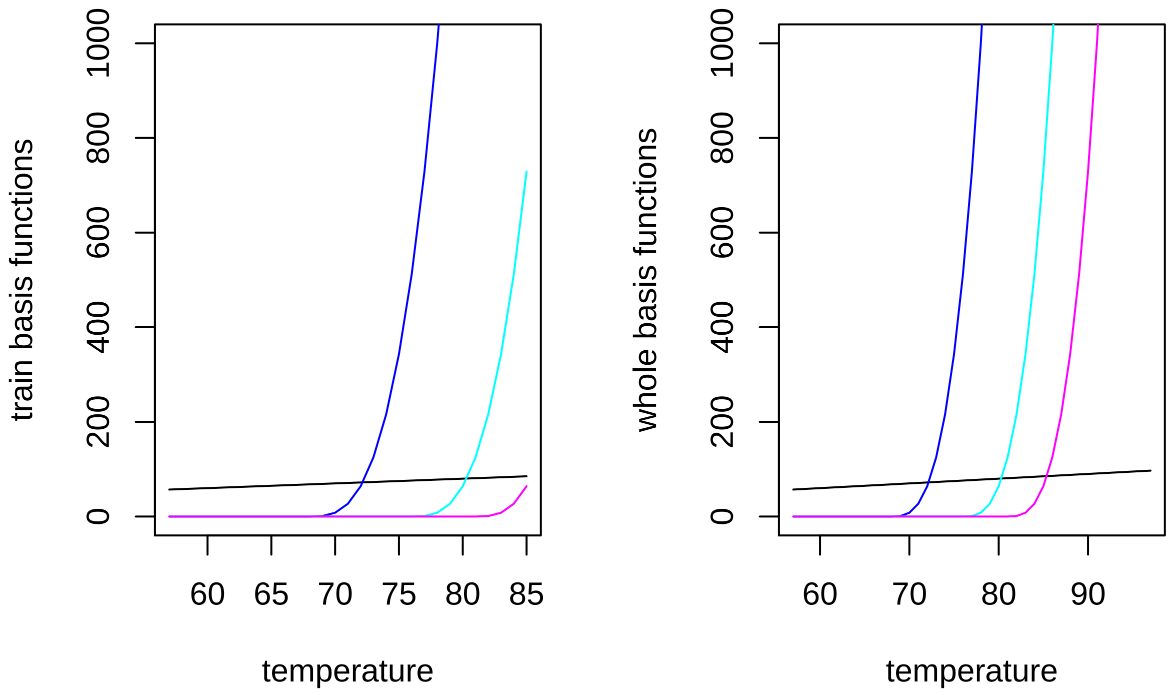
We reconstruct the basis functions using the whole data but the same knots as before.

```
extraspl <- function(x, knots, M=4){
  # nknots ... number of knots -> placed at regular quantiles
  nknots <- length(knots)
  # M ... M-1 is the degree of the polynomial
  n <- length(x)
  # X will not get an intercept column
  X <- matrix(NA,nrow=n,ncol=(M-1)+nknots)
  for (i in 1:(M-1)){ X[,i] <- x^i }
  # now the basis functions for the constraints:
  #quant <- seq(0,1,1/(nknots+1))[c(2:(nknots+1))]
  qu <- knots
  for (i in M:(M+nknots-1)){
    X[,i] <- ifelse(x-qu[i-M+1]<0,0,(x-qu[i-M+1])^(M-1))
  }
  list(X=X,xquantiles=qu)
}


spl3 <- extraspl(data[, 'temperature'], knots=spl1$xquantiles, M=4)

ozone <- data$ozone
lm3 <- lm(ozone~spl3$X)
plot(data$temperature, data$ozone, xlim=c(50, 100), main="splines fit", xlab="temperature", ylab="ozone
lines(data$temperature, predict(lm3, newdata=data.frame(data$temperature)),col='red')
abline(v=spl1$xquantiles, col='red')
```



We plot the basis functions for the train and for the whole dataset

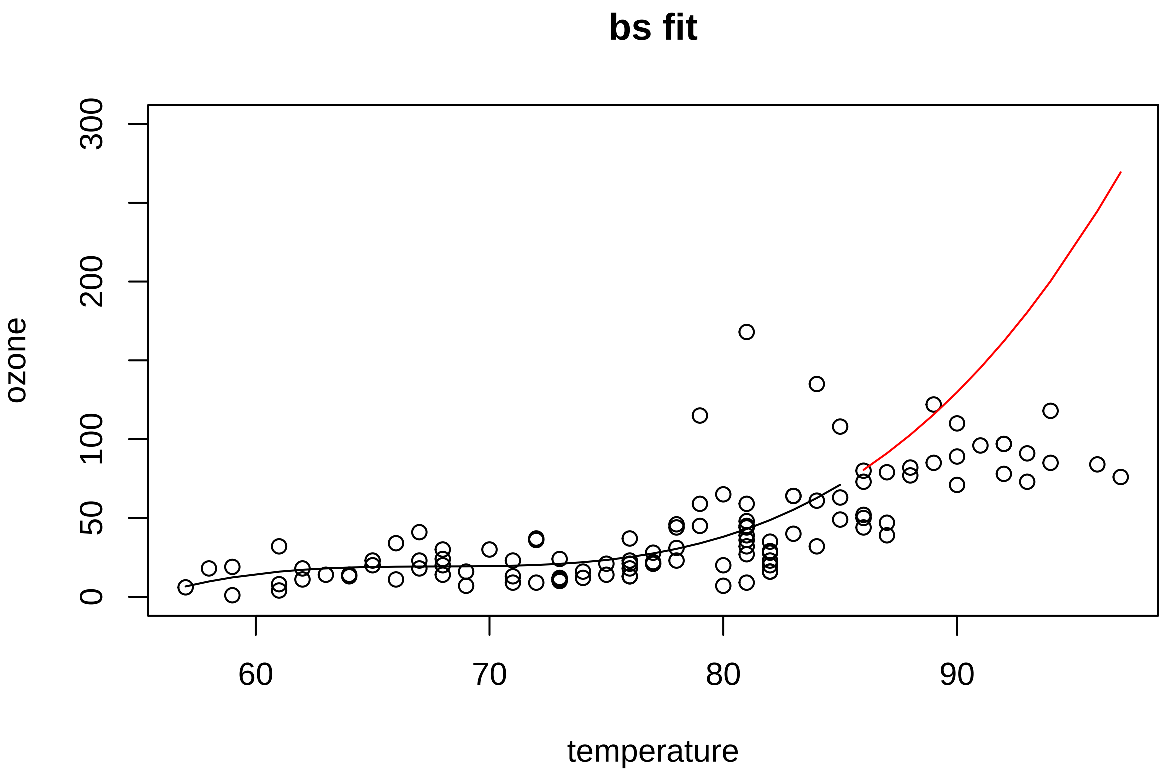
```
par(mfrow=c(1,2))
matplot(data$temperature[train], spl1$X, type="l",lty=1, ylim=c(0,1000), xlab="temperature", ylab="trai
matplot(data$temperature, spl3$X, type="l",lty=1, ylim=c(0,1000), xlab="temperature", ylab="whole basis
```



## 3)

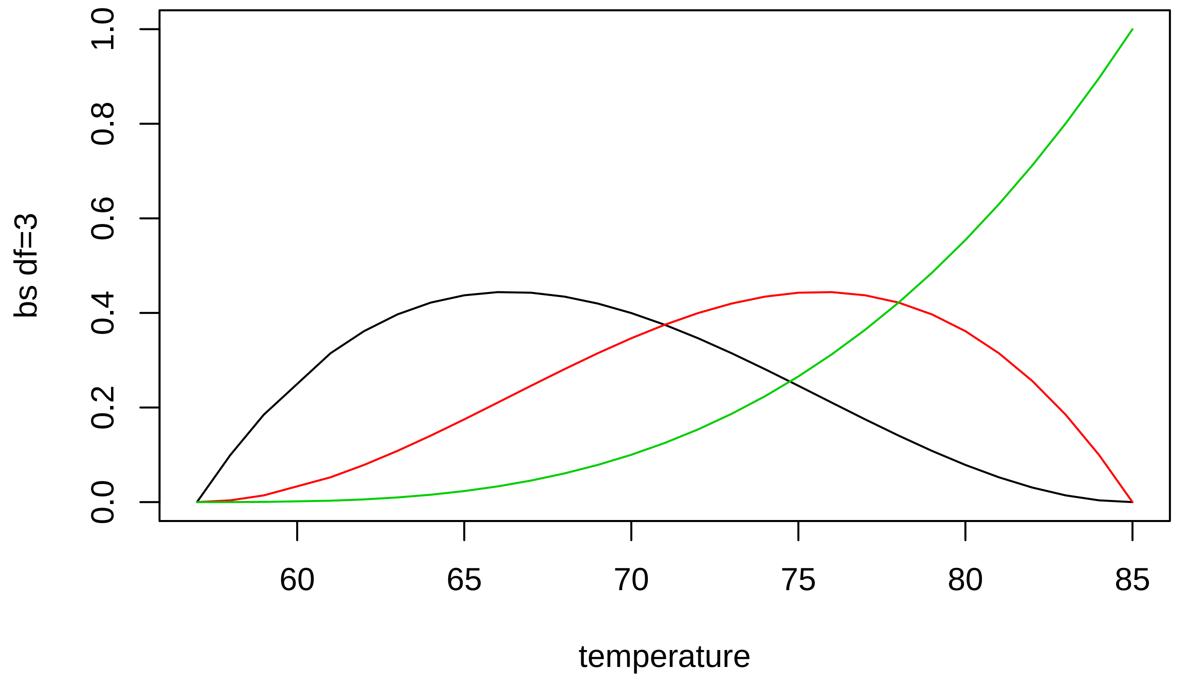
We use B-splines to fit a linear model.

```r
library(splines)


bs1 <- lm(ozone ~ bs(temperature, df=3), data=data, subset=train)

plot(data$temperature, data$ozone, ylim=c(0,300), main="bs fit", xlab="temperature", ylab="ozone")
lines(data$temperature[train], predict.lm(bs1, data.frame(temperature=data$temperature[train])))
lines(data$temperature[test], predict.lm(bs1, data.frame(temperature=data$temperature[test])), col="red
```



The basis functions for df=3

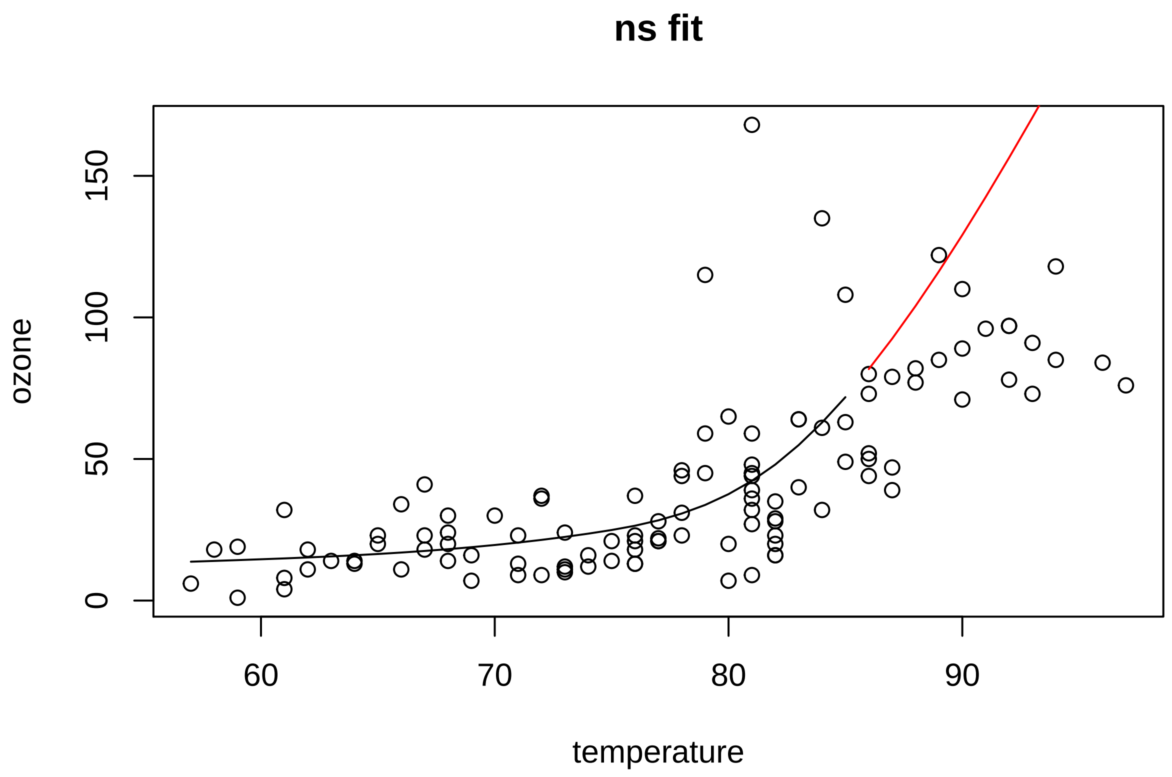
```r
matplot(data$temperature[train], bs(data$temperature[train], df=3), type="l",lty=1, xlab="temperature",
```



## 4)

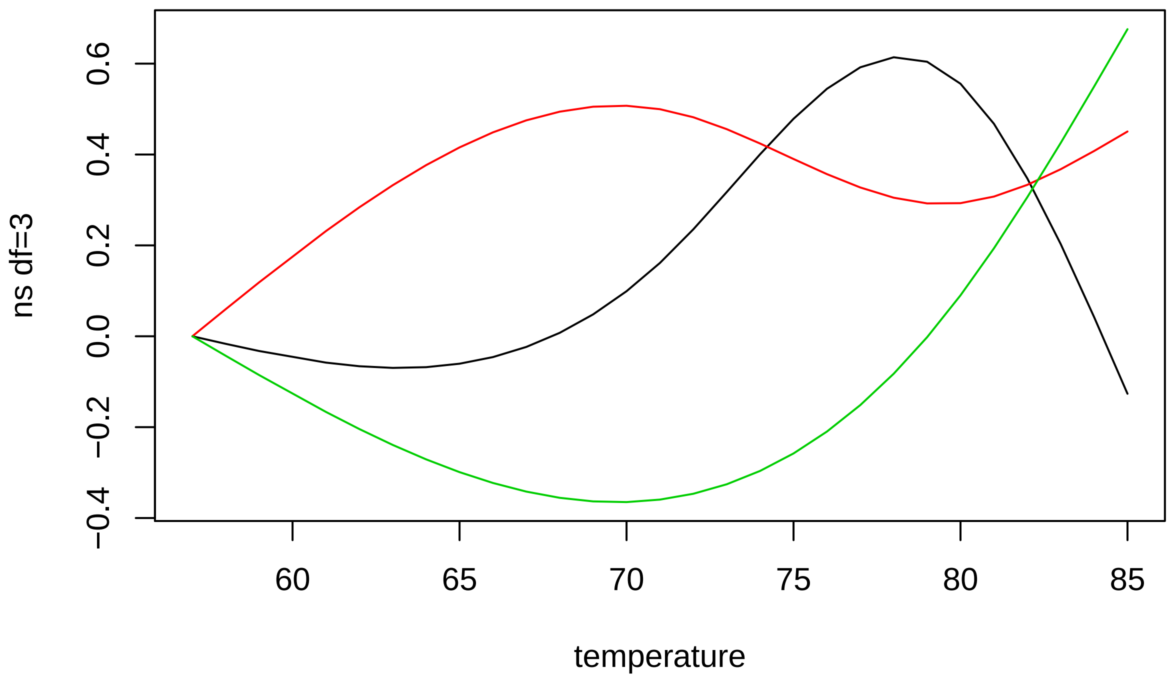
We use natural cubic splines with df=3

```
ns1 <- lm(ozone ~ ns(temperature, df=3), data=data, subset=train)
plot(data$temperature, data$ozone, main="ns fit", xlab="temperature", ylab="ozone")
lines(data$temperature[train], predict.lm(ns1))
lines(data$temperature[test], predict.lm(ns1, data.frame(temperature=data$temperature[test])), col="red
```



The corresponding basis functions

```r
matplot(data$temperature[train], ns(data$temperature[train], df=3), type="l",lty=1, xlab="temperature",
```


ns df=3
temperature