Exercise 7

Tobias Raidl, 11717659

2023-12-12

Contents

```r
library(gclus)
```

```
## Warning: package 'gclus' was built under R version 4.3.2
```

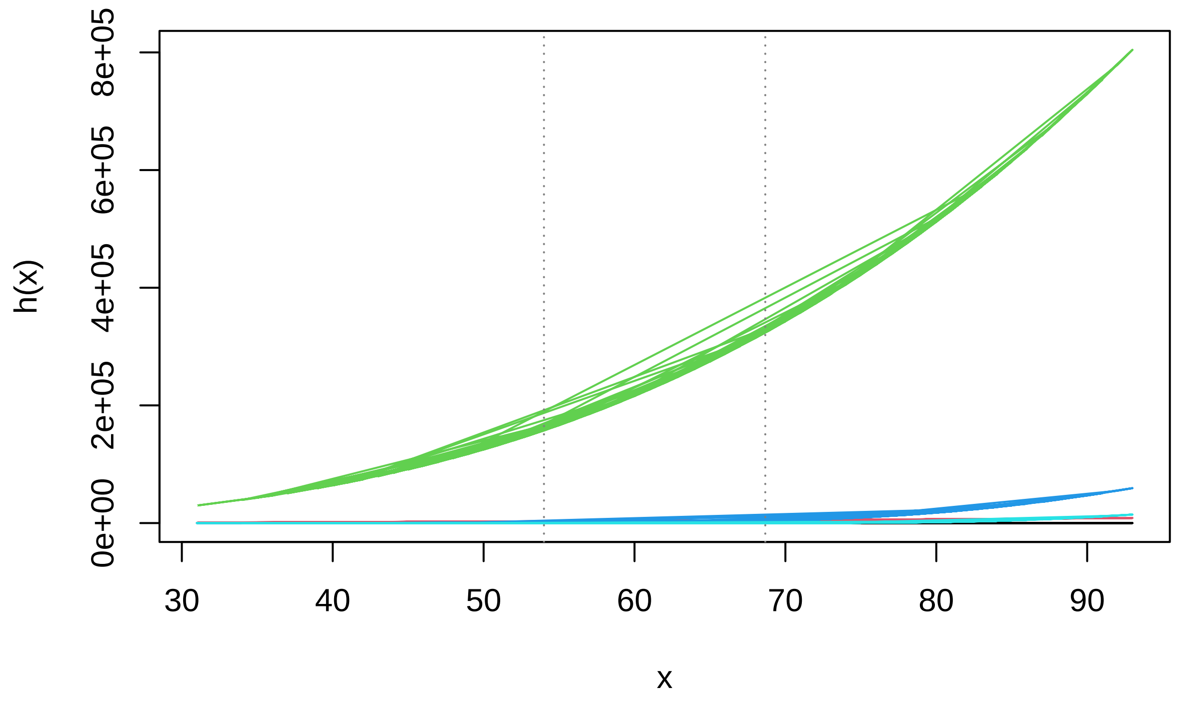
```
## Loading required package: cluster
```

```r
data(ozone)

sample <- sample(c(TRUE, FALSE), nrow(ozone), replace=TRUE, prob=c(0.7,0.3))
train  <- ozone[sample, ]
test   <- ozone[!sample, ]
```

```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)
}
```

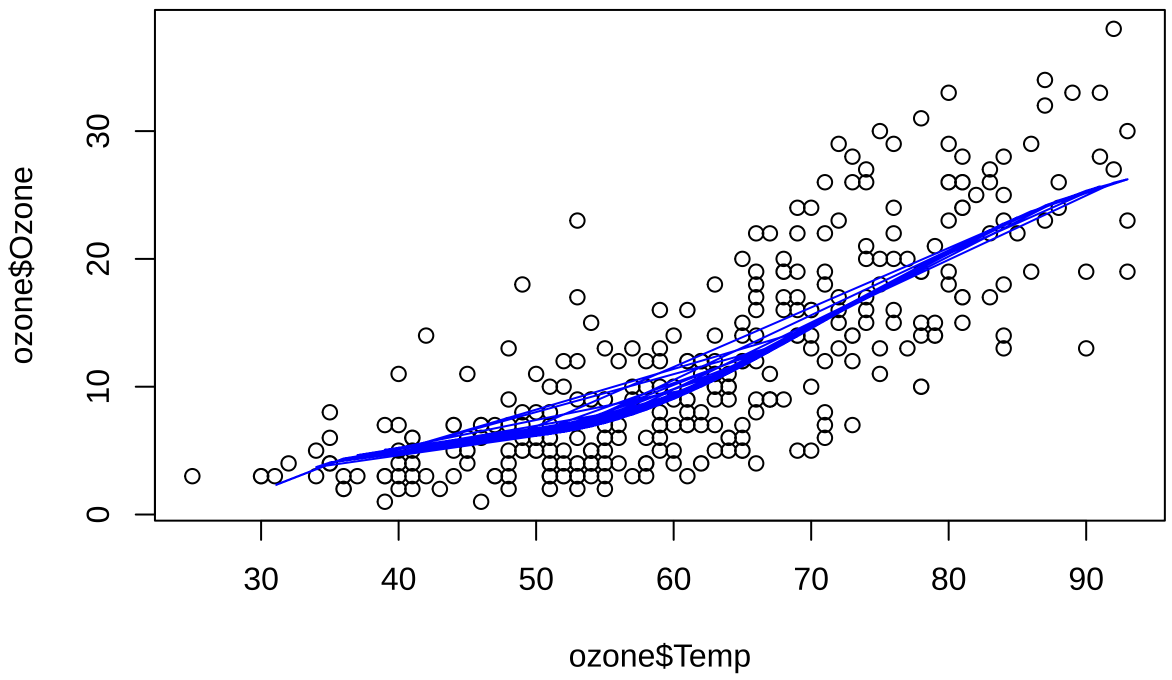
1

```r
plotspl <- function(splobj, ...){
  matplot(train$Temp, splobj$X,type="l",lty=1,
  xlab="x",ylab="h(x)", ...)
  abline(v=splobj$xquantiles,lty=3,col=gray(0.5))
}

spl = lecturespl(train$Temp)
plotspl(spl)
```



2

```
spl_train = data.frame(Ozone=train$Ozone, spl=spl$X)
spl_test = data.frame(spl=lecturespl(test$Temp)$X)

model = lm(Ozone~., data=spl_train)

plot(ozone$Temp, ozone$Ozone)
lines(train$Temp, predict(model, newdata=spl_train), col="blue")
```

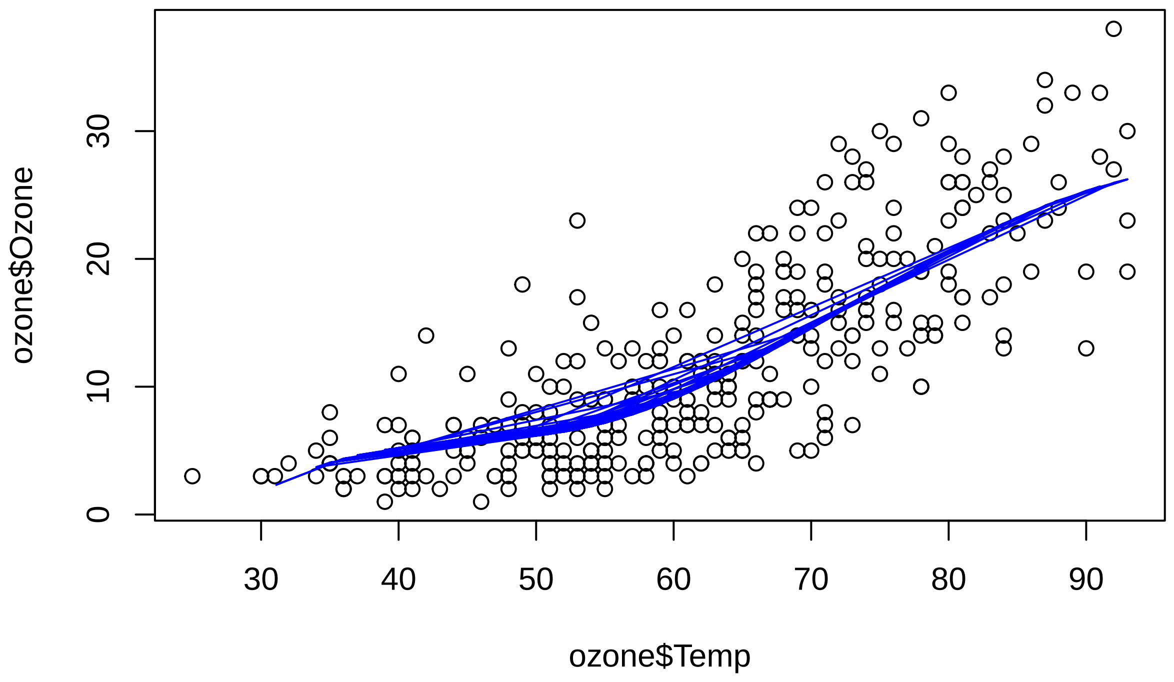


3

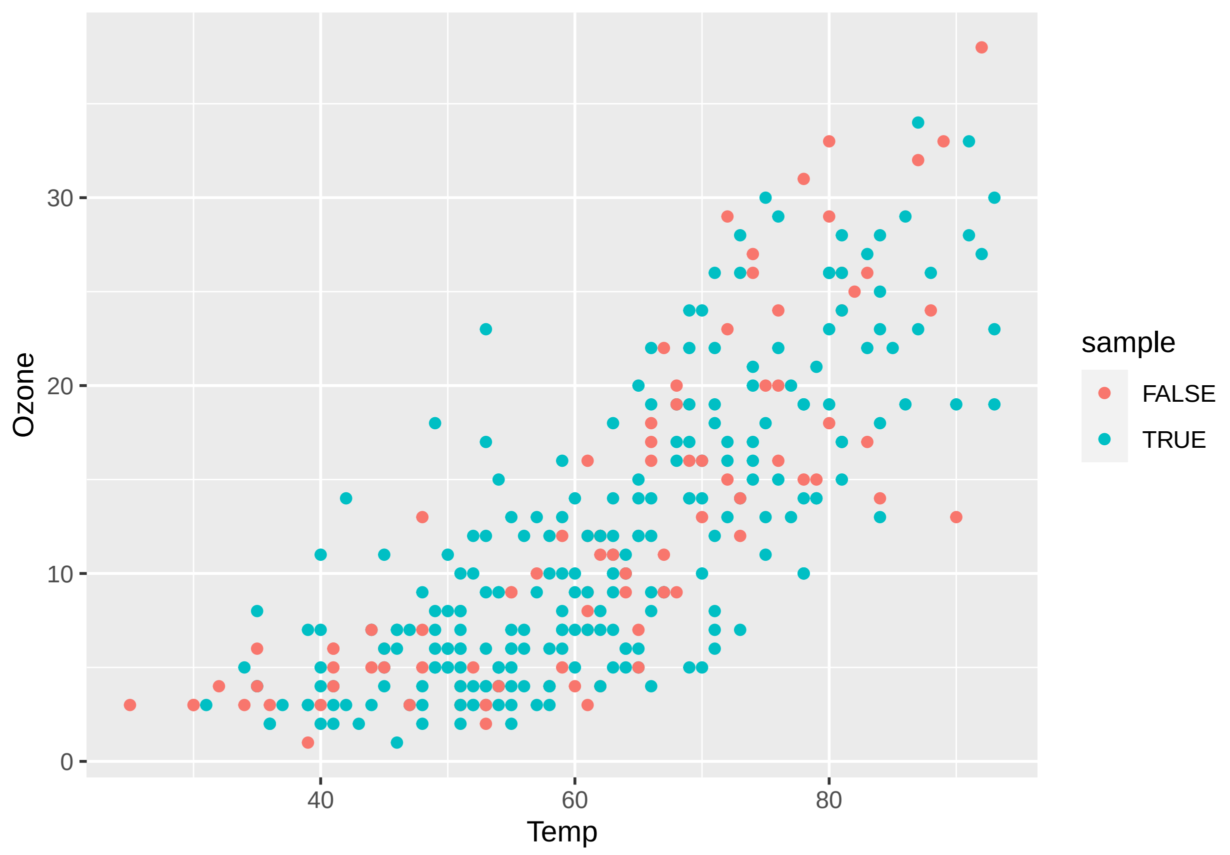
```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

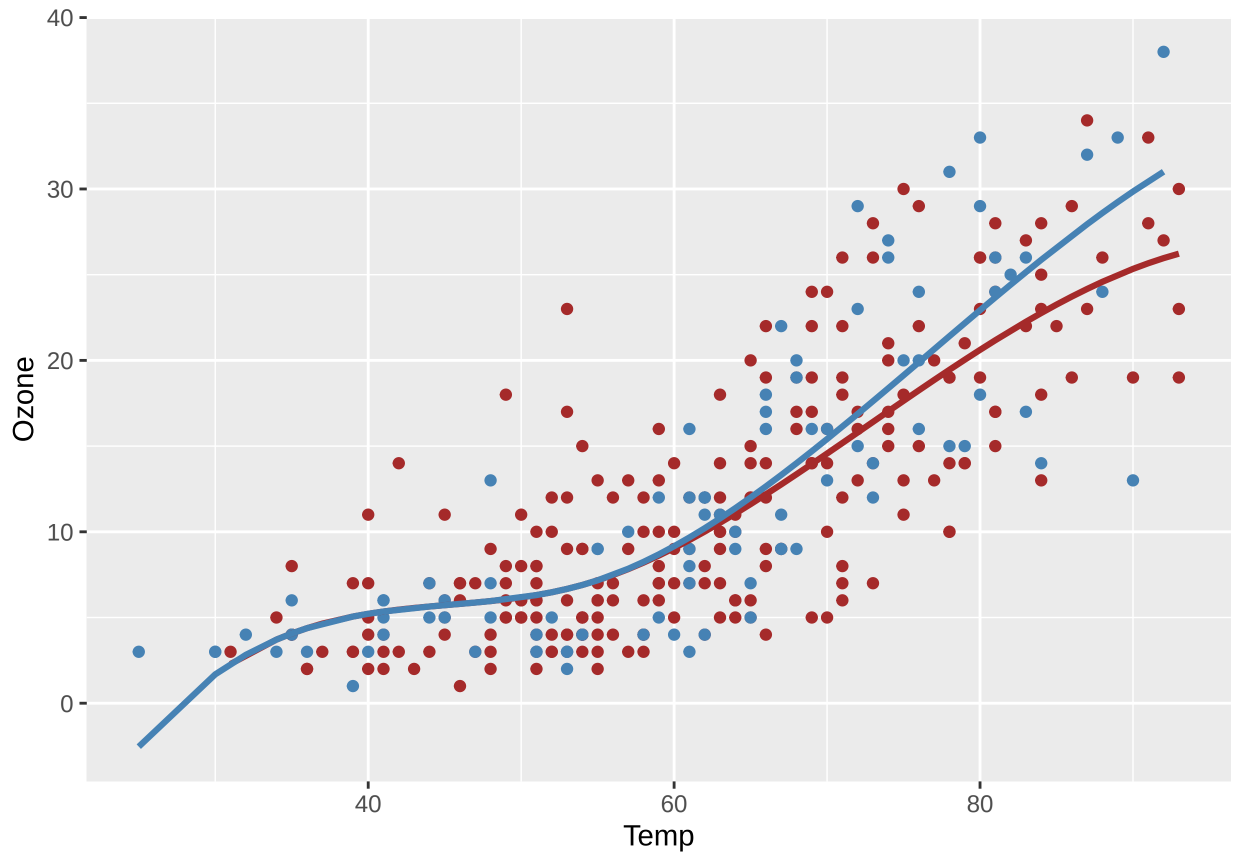
```r
plot(ozone$Temp, ozone$Ozone)
lines(train$Temp,predict(model, newdata=spl_train), col="blue")
```



```r
ggplot(data=ozone, aes(x=Temp, y=Ozone, col=sample),legend) +
  geom_point()
```



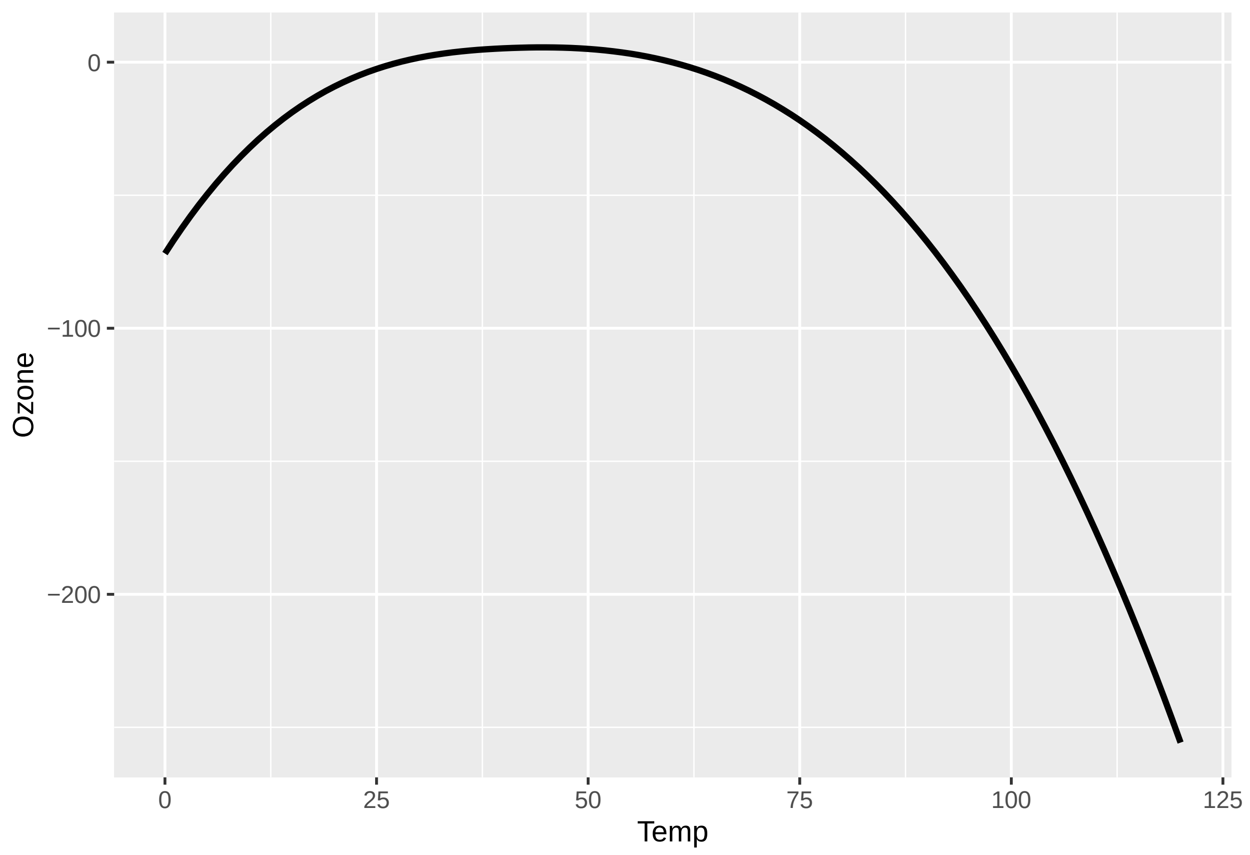
```
ggplot() +
  geom_point(data=train, aes(x=Temp, y=Ozone), col="brown") +
  geom_point(data=test, aes(x=Temp, y=Ozone), col="steelblue") +
  geom_line(data=spl_train, aes(x=train$Temp, y=predict(model, newdata=spl_train)),
  ↪  col="brown", lwd=1.1) +
  geom_line(data=spl_test, aes(x=test$Temp, y=predict(model, newdata=spl_test)),
  ↪  col="steelblue", lwd=1.1)
```



4

Generate new temperature data with seq(0,120). Thus, we extend the range of the explanatory variable. Use the model derived above to obtain predictions, and present those in the plot from 3. (by first extending the x-range)

```r
aug_temp = seq(0,120)
spl_aug = data.frame(spl=lecturespl(aug_temp)$X)

ggplot() +
  geom_line(data=spl_aug, aes(x=aug_temp, y=predict(model, newdata=spl_aug)), lwd=1.1) +
  labs(x="Temp", y="Ozone")
```



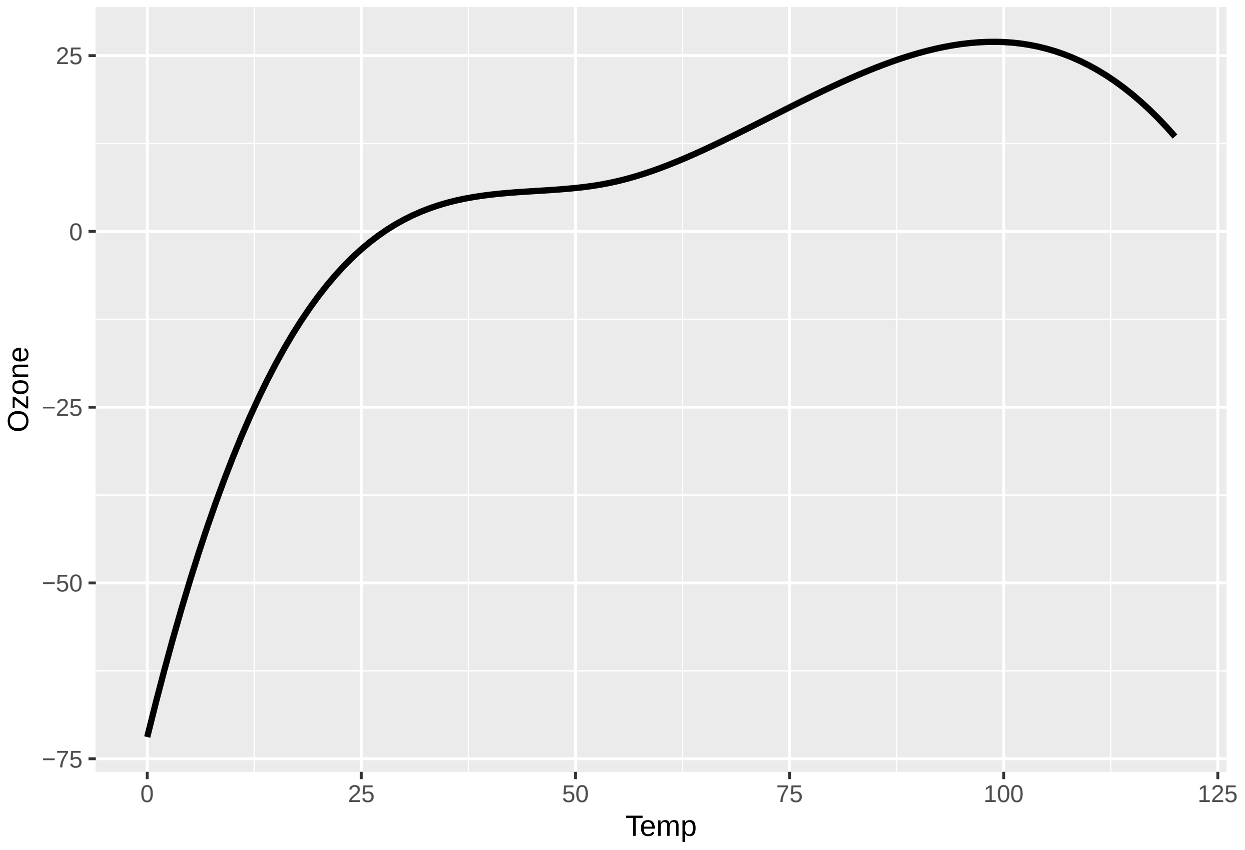
This does not seem right…

5

You might realize that the predictions from 4. are non-sense. The problem is that lecturespl() constructs the knots on the quantiles of the input variable. However, we should use the knots that have been constructed for the training data. Thus, modify the function accordingly, compute the predictions again, and visualize the new results.

```r
lecturespl_modified <- 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(train$Temp,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)
}
```

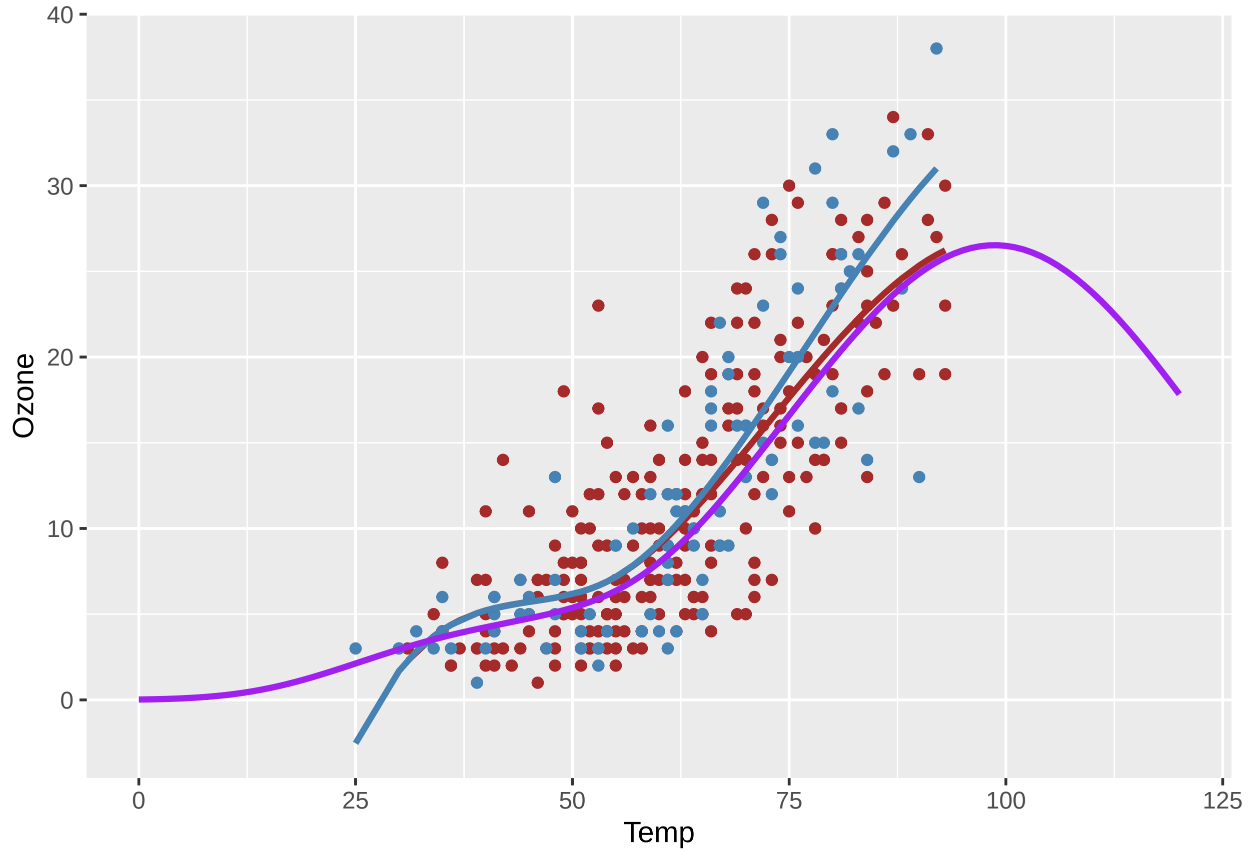
```
aug_temp = seq(0,120)
spl_aug = data.frame(spl=lecturespl_modified(aug_temp)$X)

ggplot() +
  geom_line(data=spl_aug, aes(x=aug_temp, y=predict(model, newdata=spl_aug)), lwd=1.1) +
  labs(x="Temp", y="Ozone")
```



6

Another problem: Your predictions for low temperatures might be negative. However, negative ozone concentrations are very rare in practice. Forcing the (smooth) predictions to be non-negative might be quite complicated. An easy way out is to use the log-transformed response inside lm(), and exp() for the resulting predictions. Thus, do the analyses again, and present all results in one plot (training and test data, and the predictions for training, test and new x data).

```
log_model = lm(log(Ozone)~., data=spl_train)

ggplot() +
  geom_point(data=train, aes(x=Temp, y=Ozone), col="brown") +
  geom_point(data=test, aes(x=Temp, y=Ozone), col="steelblue") +
  geom_line(data=spl_train, aes(x=train$Temp, y=predict(model, newdata=spl_train)),
  ↪  col="brown", lwd=1.1) +
  geom_line(data=spl_test, aes(x=test$Temp, y=predict(model, newdata=spl_test)),
  ↪  col="steelblue", lwd=1.1) +
  geom_line(data=spl_aug, aes(x=aug_temp, y=exp(predict(log_model, newdata=spl_aug))),
  ↪  col="purple", lwd=1.1)
```


Ozone
Temp
0
10
20
30
40
0
25
50
75
100
125