Advances Computational Econometrics. Chapter 1: Introduction. Solution of exercises

Thierry Denoeux

2023-04-07

Exercise 1

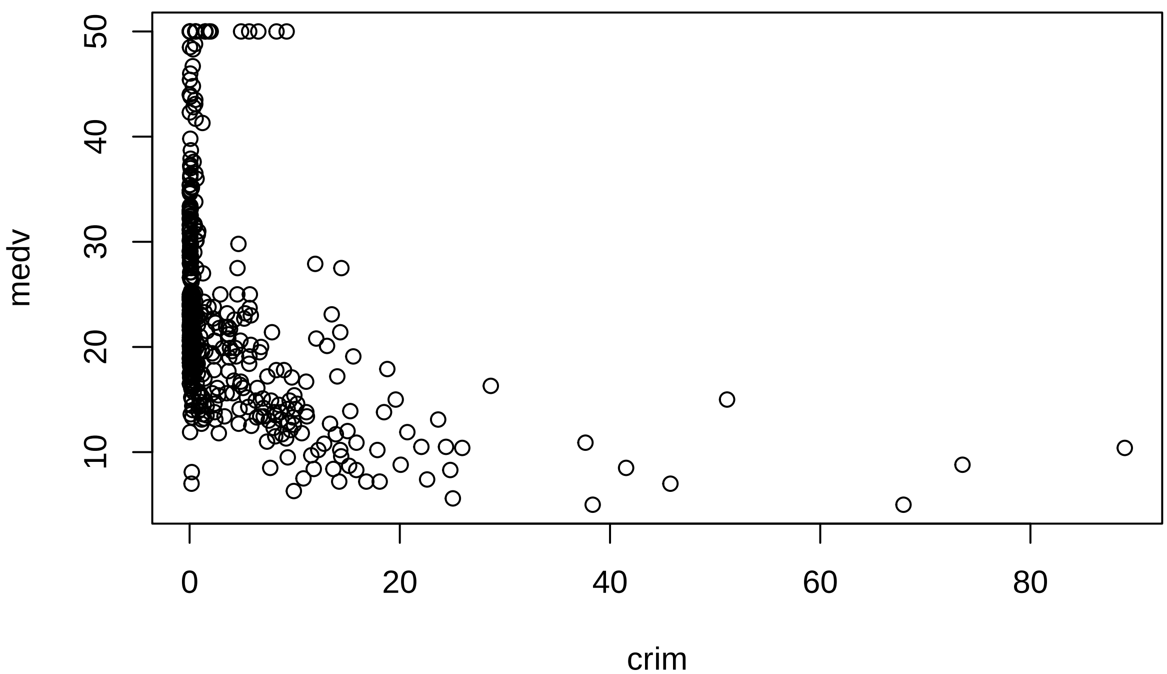
Question 1

Loading the data:

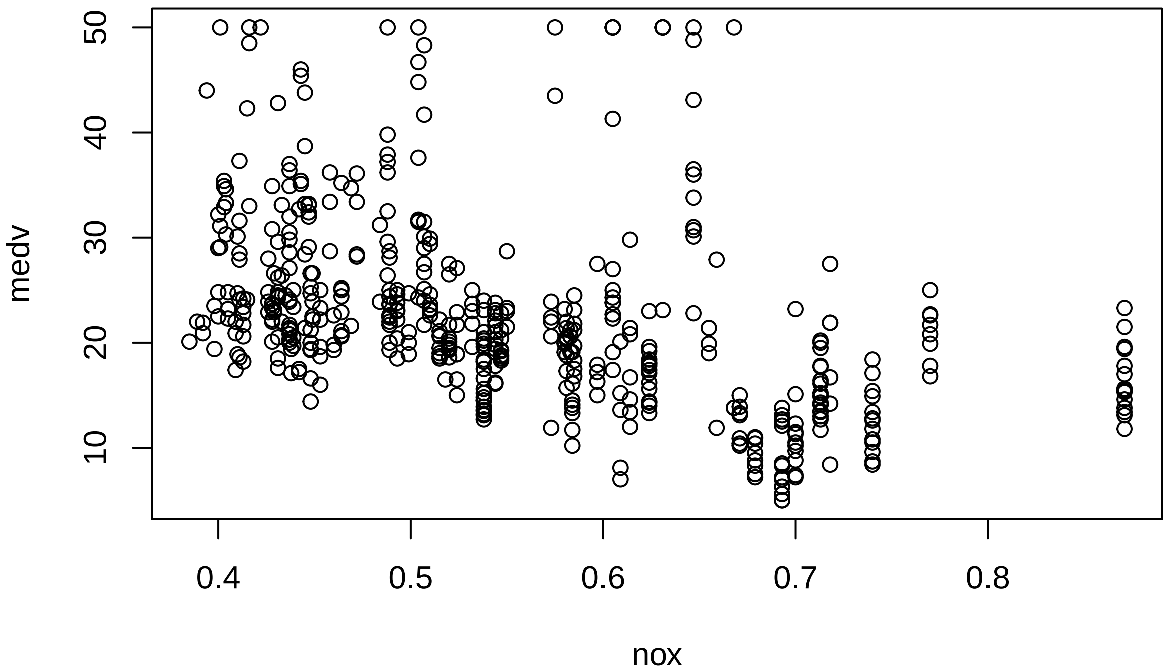
```
library(MASS)
attach(Boston)
```

We plot the response against some of the predictors:

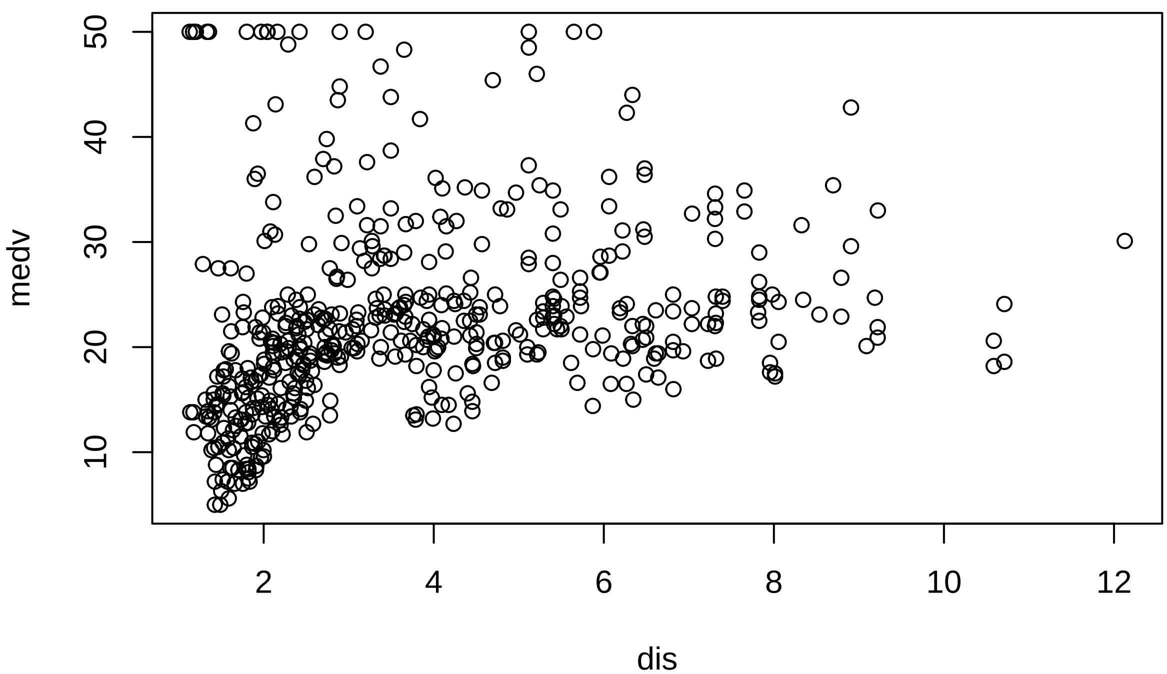
```
plot(crim,medv)
```



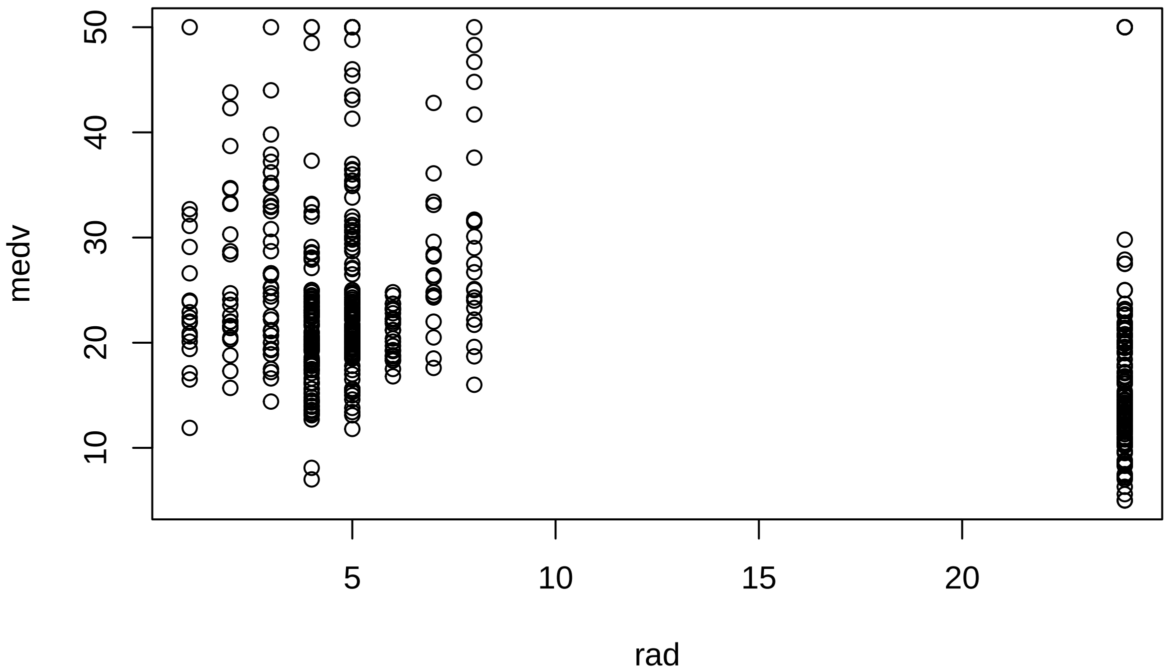

```
plot(nox,medv)
```



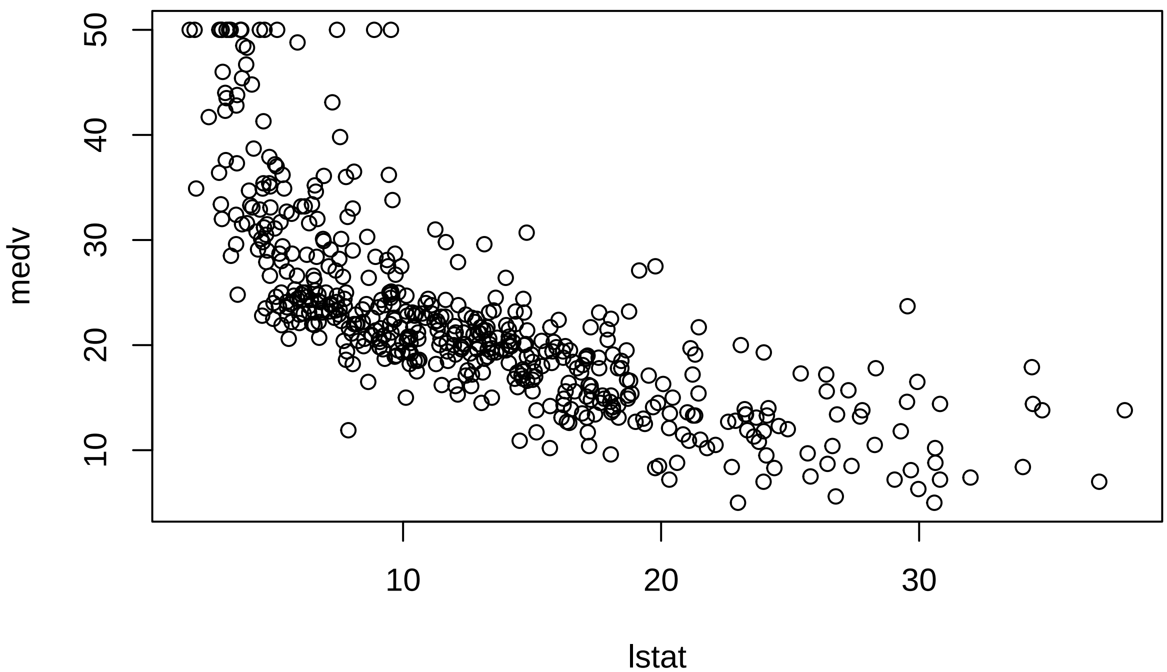

```
plot(dis,medv)
```




```
plot(rad,medv)
```




```
plot(lstat,medv)
```



Question 2

We split the data randomly between a training set (2/3) and a test set (1/3):

```
set.seed(220322)
n<-nrow(Boston)
ntrain<-round(2*n/3)
ntest<-n-ntrain
```

```
train<-sample(n,ntrain)
Boston.train<-Boston[train,]
Boston.test<-Boston[-train,]
```

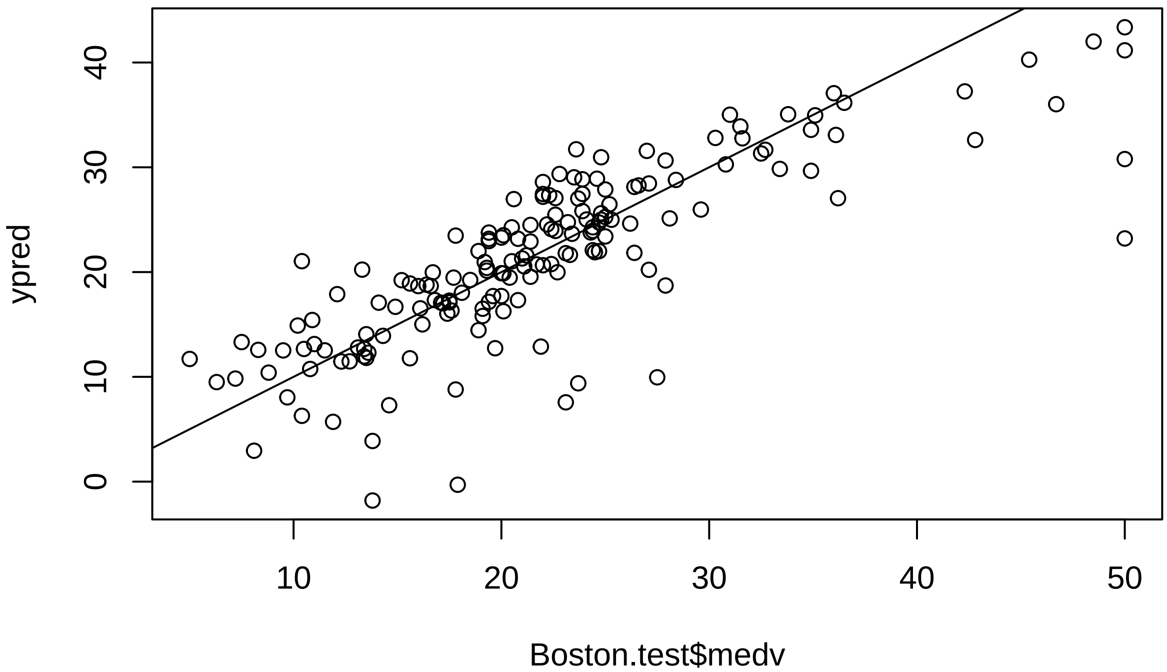
Question 3

We apply linear regression to the data set:

```
fit<-lm(medv~.,data=Boston.train)
ypred<-predict(fit,newdata=Boston.test)
```

We plot the predicted values vs. the observed values of the response, and we display the corresponding mean-squared error on the test set:

```
plot(Boston.test$medv,ypred)
abline(0,1)
```




```
mse.linreg<- mean((Boston.test$medv-ypred)^2)
print(mse.linreg)
```

```
## [1] 28.94119
```

Question 4

We standardize the predictor variables:

```
x.train<-scale(Boston.train[,-14])
x.test<-scale(Boston.test[,-14])
```

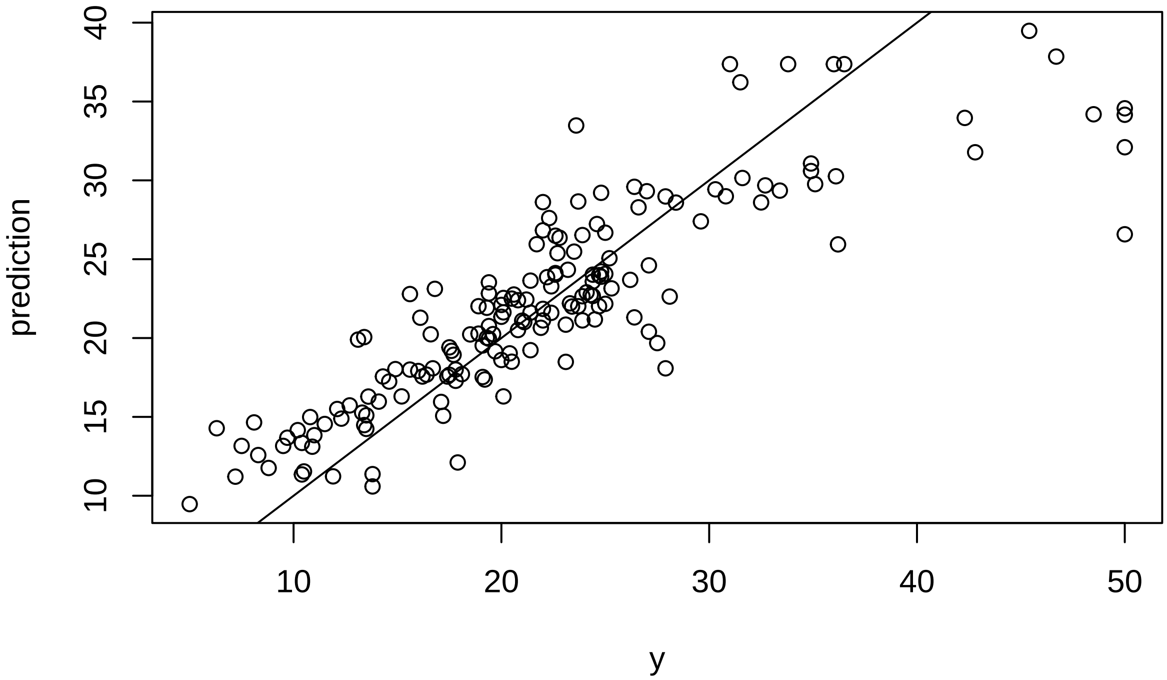
We predict the response for the test set using the kNN regression method with $K = 10$ neighbors, and display the corresponding test MSE:

```
library(FNN)
knnfit<-knn.reg(train=x.train, test = x.test, y=Boston.train$medv, k = 10)
mean((Boston.test$medv-knnfit$pred)^2)
```

```
## [1] 21.81451
```

As before, we plot the predicted response vs. the observed response:

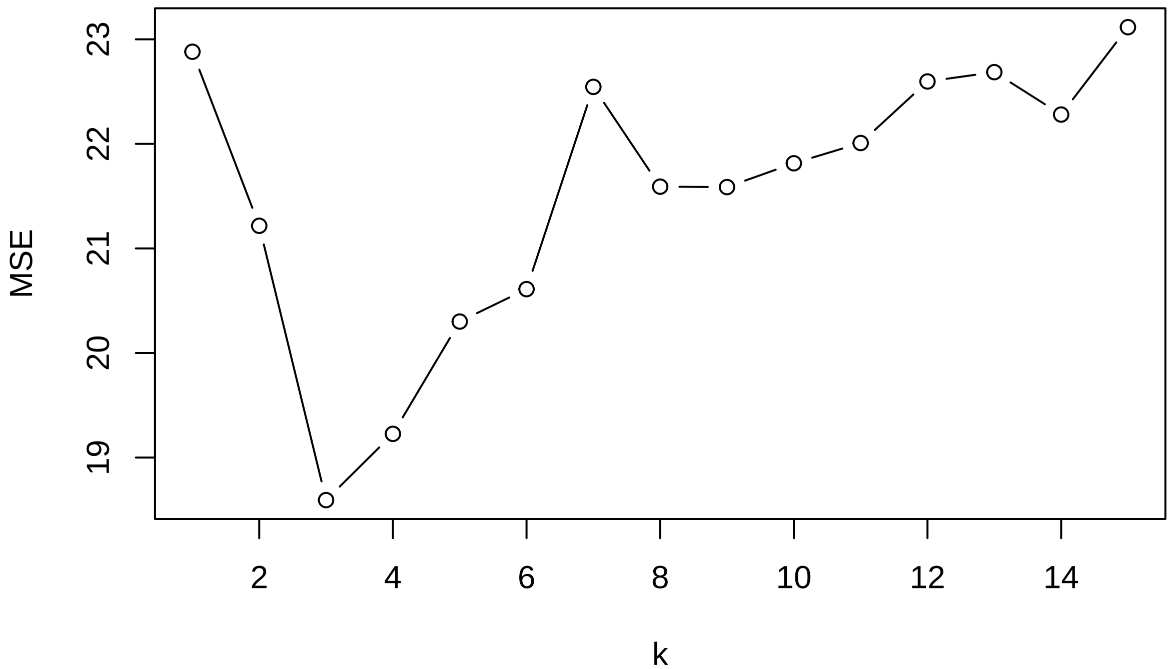
```
plot(Boston.test$medv,knnfit$pred,xlab='y',ylab='prediction')
abline(0,1)
```



Question 5

Plotting the test MSE as a function of the number of neighbors

```
MSE<-rep(0,15)
for(k in 1:15){
  knnfit<-knn.reg(train=x.train, test = x.test, y=Boston.train$medv, k = k)
  MSE[k]<-mean((Boston.test$medv-knnfit$pred)^2)
}
plot(1:15,MSE,type='b',xlab='k',ylab='MSE')
```

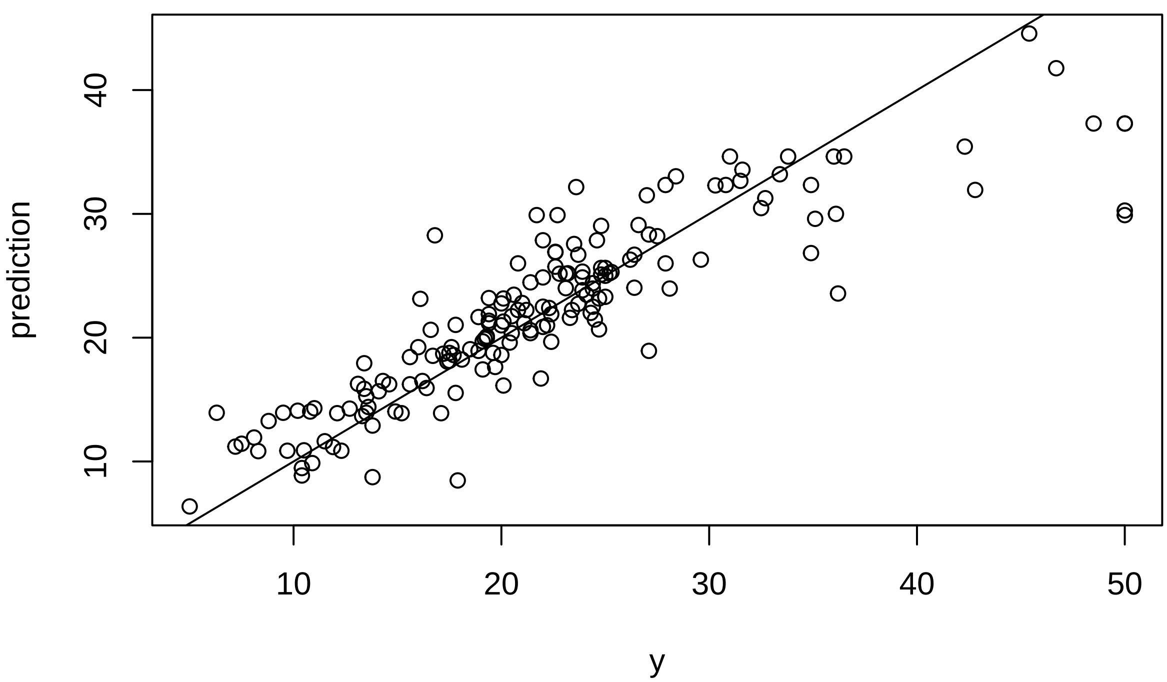


For the optimal number of neighbors, we compute again the test MSE and plot the predicted response vs. the observed response:

```
kopt<-which.min(MSE)
knnfit<-knn.reg(train=x.train, test = x.test, y=Boston.train$medv, k = kopt)
mean((Boston.test$medv-knnfit$pred)^2)
```

```
## [1] 18.59324
```

```
plot(Boston.test$medv,knnfit$pred,xlab='y',ylab='prediction')
abline(0,1)
```



Exercise 2

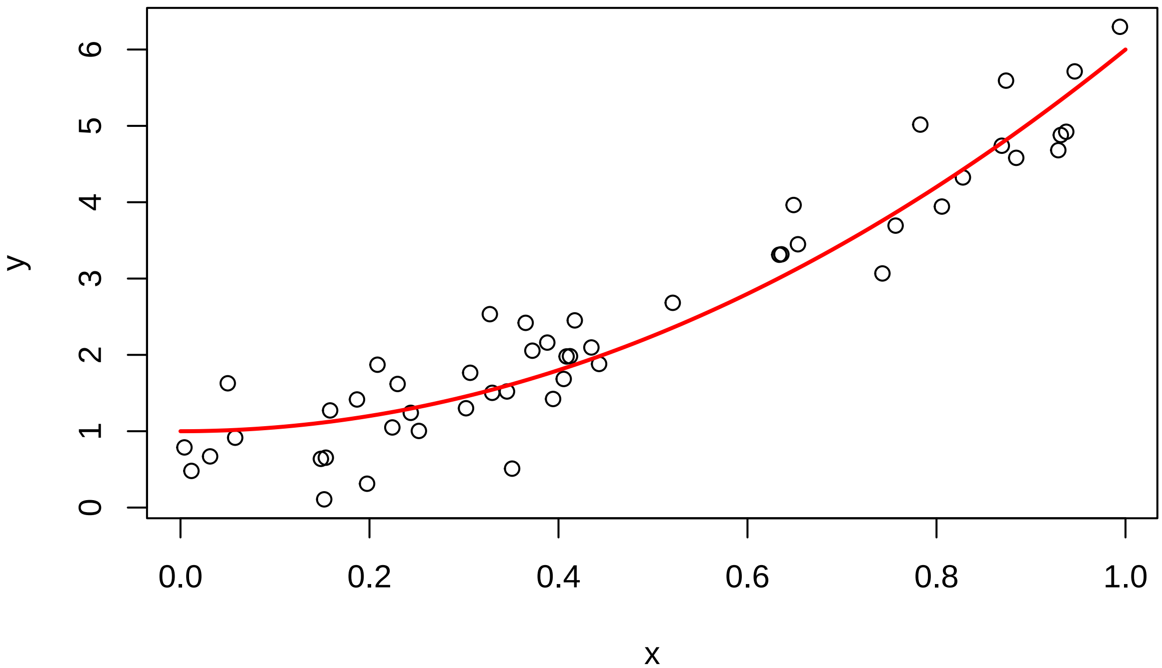
Question 1

See the slides.

Question 2

Generation of a dataset:

```
sig<- 0.5
n<-50
x<-runif(n)
y<-1+5*x^2+sig*rnorm(n)
plot(x,y)
x1<-seq(0,1,0.01)
lines(x1,1+5*x1^2,col="red",lwd=2)
```



We fix $x_0 = 0.5$:

```
x0<-0.5
Ey0<-1+5*x0^2
```

Generation of 10,000 datasets; for each dataset, and each value of K in $\{1, \ldots, 40\}$, estimation of $f(x_0)$:

```
N<-10000 # we generate 10000 learning sets
Kmax<- 40
fhat<-matrix(0,N,Kmax)
y0<-rep(0,N)
for(i in 1:N){
  # learning set generation
  x<-runif(n)
  y<-1+5*x^2+sig*rnorm(n)
```

```
  # generation of one observation of Y
  y0[i]<-Ey0+sig*rnorm(1)
  # Compute the predictions for K=1,...,Kmax
  for(K in 1:Kmax) fhat[i,K]<-knn.reg(train=x, test = x0, y=y, k = K)$pred
}
```

Calculation of the MSE, squared bias and variance for each value of K:

```
error<-rep(0,Kmax)
biais2<-rep(0,Kmax)
variance<-rep(0,Kmax)
for(K in 1:Kmax){
  error[K]<-mean((fhat[,K]-y0)^2)    # MSE
  biais2[K]<-(mean(fhat[,K])-Ey0)^2 # bias^2
  variance[K]<-var(fhat[,K])         # variance
}
```

Plotting the MSE (in blue) and the sum of the squared bias, variance and irreducible error (in red):

```
plot(1:Kmax,error,type="l",ylim=range(error,biais2,variance),col="blue",xlab="K",lwd=2)
lines(1:Kmax,biais2,lty=2,col="green",lwd=2)
lines(1:Kmax,variance,lty=3,lwd=2)
abline(h=sig^2,lty=2,col="cyan",lwd=2)
lines(1:Kmax,biais2+variance+sig^2,col="red",lwd=2)
```