

Assignment 8

Anant Vishwakarma

5/25/2022

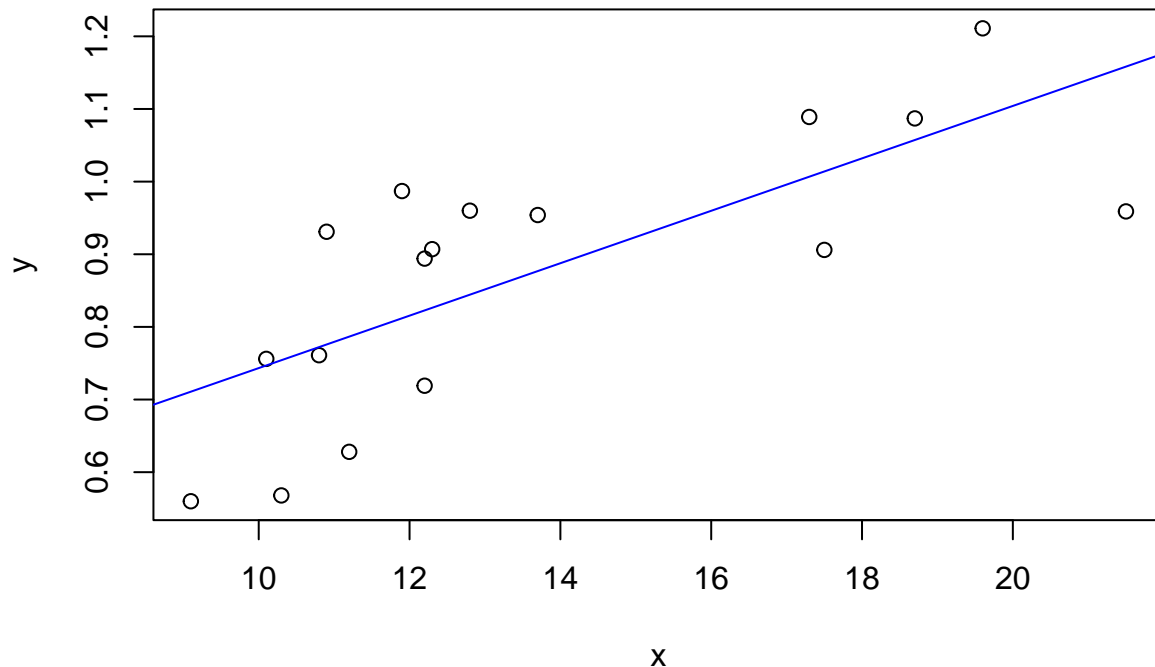
In the file `boneDensity.csv`, you can find data consisting of a measurement of bone density (y) and age in years (x) for each of 17 children and young adults. Imagine that you will be responsible for predicting peoples' bone densities from their ages. How accurate (measure this with MSPE!) would you estimate predictions based on linear regression will be? How accurate would you estimate predictions based on a kernel regression model with bandwidth 1.5 will be? Which of the two is better? Finally, does the answer to this change if you use larger bandwidths for the kernel regression?

We are given a data file with the bone densities of 17 children and young adults. The task is to check the accuracy of predicting peoples' bone densities from their ages.

This is a part of the data set for reference:

| ## | | x | y |
|----|---|------|-------|
| ## | 1 | 12.3 | 0.907 |
| ## | 2 | 9.1 | 0.560 |
| ## | 3 | 17.5 | 0.906 |
| ## | 4 | 19.6 | 1.211 |
| ## | 5 | 12.8 | 0.960 |
| ## | 6 | 12.2 | 0.894 |

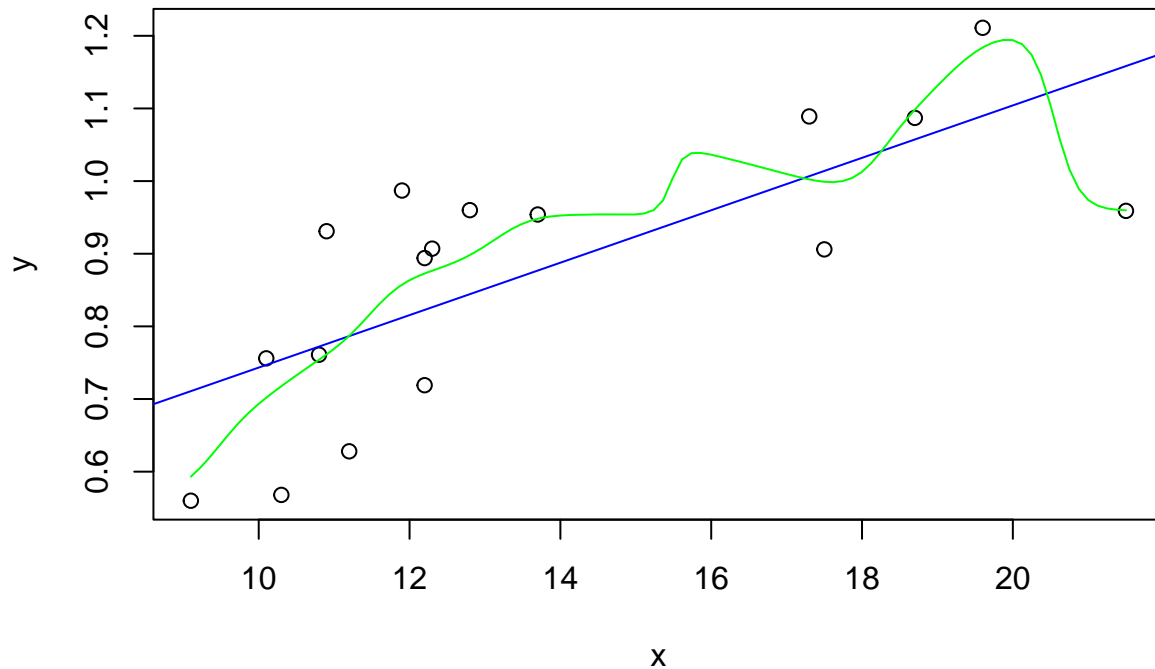
By using the leave one out linear one out linear regression method we can test out the prediction accuracy. We can plot this data as:



```
## [1] 0.02091536
```

When predicting people's bone densities from the given data set using the linear regression, the estimated prediction comes out to be 0.0209.

This can also be done using the leave one out kernel regression method. The kernel error obtained with bandwidth = 1.5 is:



```
## [1] 0.02263895
```

Let's now see what happens if we change the bandwidth. The kernel error obtained with a higher bandwidth (in this case 3) is :

```
## [1] 0.0204717
```

The kernel error, when the bandwidth is increased to 3 turn out to be 0.0247.

Discussion of Results

When predicting people's bone densities from the given data set using the kernel regression model, the estimated prediction comes out to be 0.2263. Between the error measured by linear regression and the the error measured by kernel regression the error value for linear model is lower. Hence the linear model seems to be better in this case.

When the bandwidth is increased the kernel error becomes lower. The bandwidth is a tuning parameter i.e it controls the smoothness of the non linear fit. The higher the bandwidth the smoother it fits. Thus the kernel error decreases with a higher bandwidth. With the higher bandwidth (now at 3) the kernel error comes out to be lower than the linear error.

Appendix

```
knitr::opts_chunk$set(echo = FALSE)
knitr::opts_chunk$set(echo = FALSE)
bones <- read.csv("boneDensity.csv")
head(bones)

knitr::opts_chunk$set(echo = FALSE)
#the linear leave one out function
oneOut <- function(row, dataSet){
  testingData <- dataSet[row,]
  trainingData <- dataSet[-row,]
  trainedLM <- lm(y~x, trainingData)
  testingLMPrediction <- predict(trainedLM, testingData)
  (testingLMPrediction - testingData$y)^2
}

plot(bones)
fittedLM <- lm(y~x, data=bones)
abline(fittedLM, col="blue")

errors <- NULL
n <- nrow(bones)
for (i in 1:n){
  errors[i] <- oneOut(i, bones)
}
linearError <- mean(errors)
linearError

knitr::opts_chunk$set(echo = FALSE)
plot(bones)
fittedLM <- lm(y~x, data=bones)
abline(fittedLM, col="blue")

kernelFit <- with(bones, ksmooth(x, y,
kernel = "normal",
bandwidth = 1.5))
lines(kernelFit, col="green")

#the kernel leave one out function
kernelOneOut <- function(row, dataSet, bw){
  testingData <- dataSet[row,]
  trainingData <- dataSet[-row,]
  kernelPrediction <- with(trainingData,
                           ksmooth(x, y,
                                   kernel = "normal",
                                   bandwidth = bw,
                                   x.points = testingData$x))
  (kernelPrediction$y - testingData$y)^2
}

errors2 <- NULL
```

```

n <- nrow(bones)
for (i in 1:n){
  errors2[i] <- kernelOneOut(i, bones, bw=1.5)
}
kernelError <-mean(errors2)
kernelError

knitr::opts_chunk$set(echo = FALSE)
errors2 <- NULL
n <- nrow(bones)
for (i in 1:n){
  errors2[i] <- kernelOneOut(i, bones, bw=3)
}
kernelError2 <-mean(errors2)
kernelError2

```