Local_poison_regression

Gregoire Gasparini, Aurora Hofman, Beatriu Tort

24 de marzo de 2020

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(splines)
```

Loading and reading data:

```
load("bikes.Washington.Rdata")
```

```
cnt <- bikes$cnt
instant <- bikes$instant
```

Excersice 1 ——————————————————————————————-

Estimate function using smooth.spline:

```
m<- smooth.spline(x= instant, y= cnt) #uses GCV by default.
```

```
#Penalty parameter:
m$lambda
```

```
## [1] 1.005038e-07
```

```
#Correstponding degrees of freedom:
m$df
```

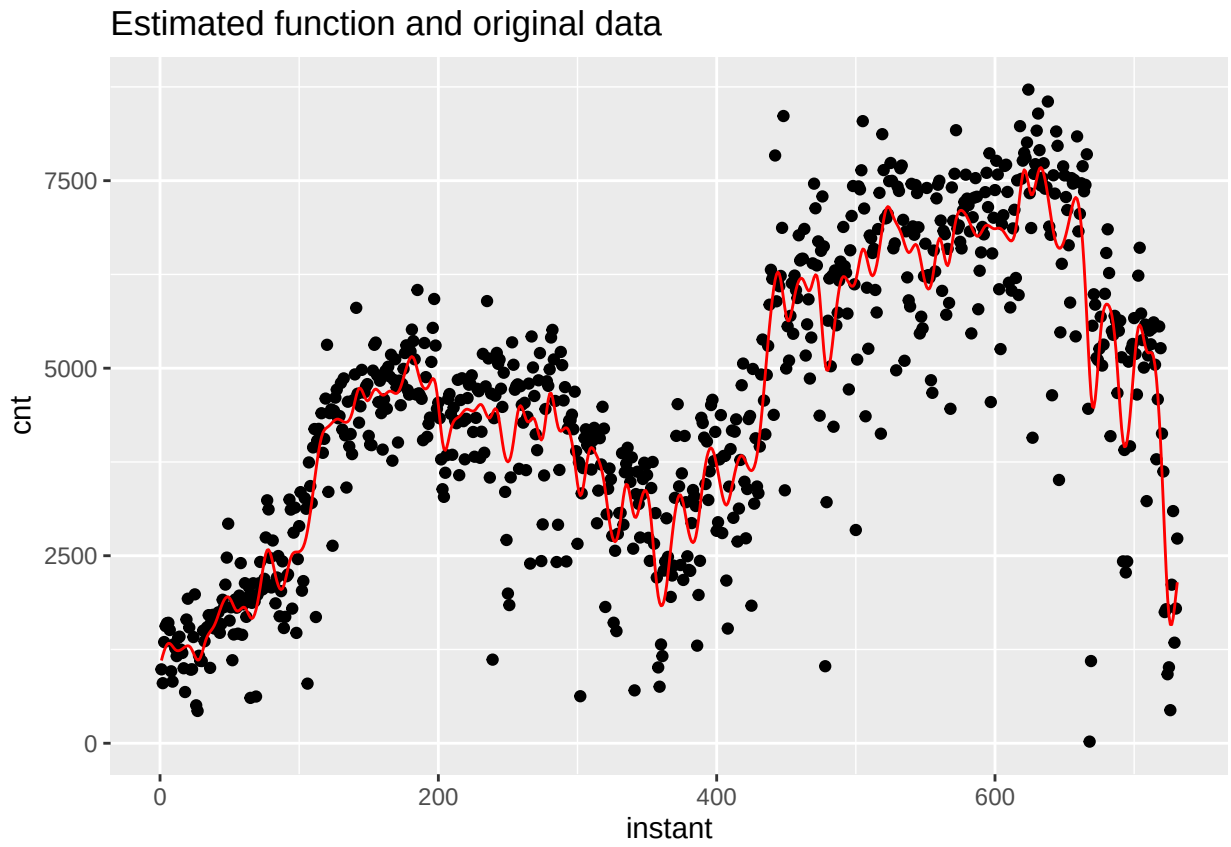
```
## [1] 93.34091
```

```
#Number of knots:
length(m$fit$knot)
```

```
## [1] 140
```

```
df<- tibble(instant, cnt, m$x, m$y)

g<- ggplot(data = df, aes(instant, cnt)) + geom_point() +
  geom_line(aes(m$x, m$y), colour = "red") +
  ggtitle("Estimated function and original data")
g
```

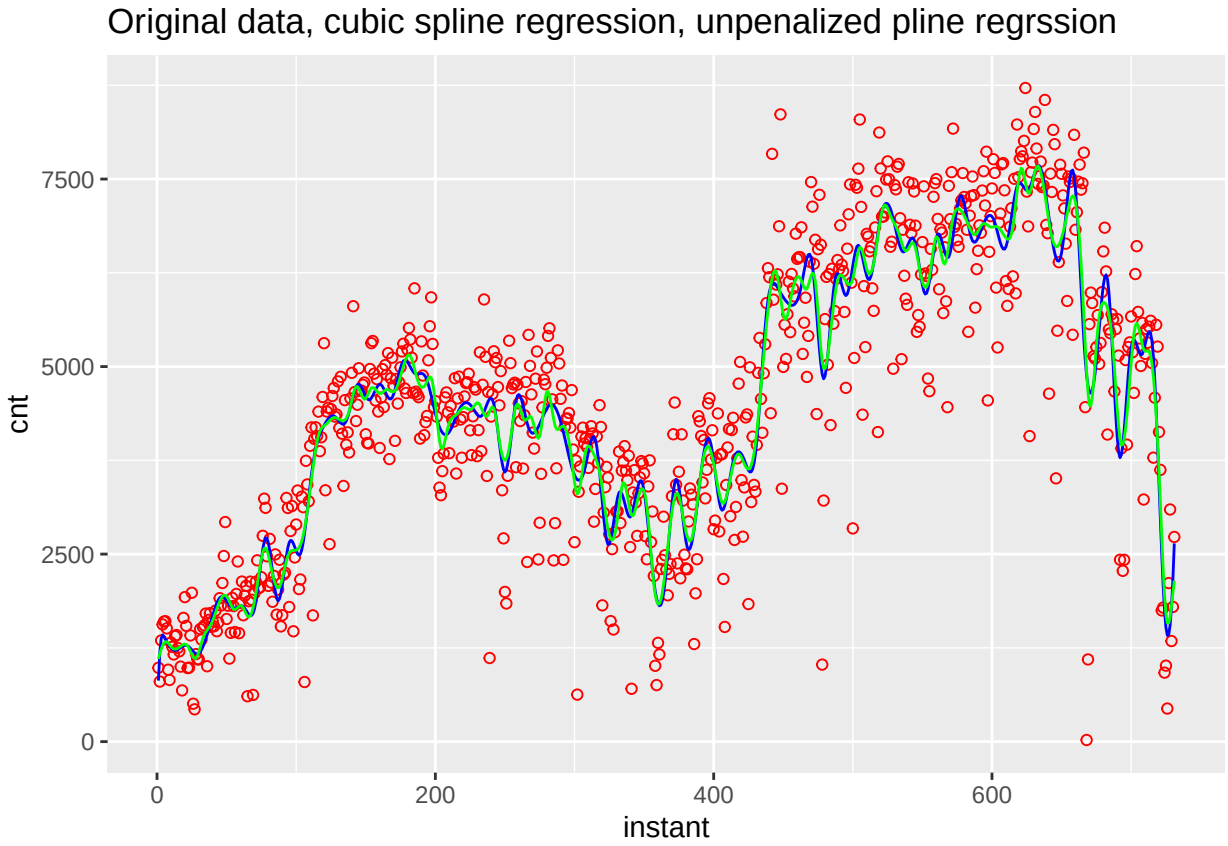


Estimation using R bs and lm

```
n.knots <- m$df - 4
my.knots <- quantile(instant, ((1:n.knots)-.5)/n.knots)

m_bs <- bs(instant, knots = my.knots, intercept=T)
m_lm <- lm(cnt~m_bs-1)
```

```
df_tot <- tibble(instant, cnt, m_lm$fitted.values, m$y)
g_lm <- ggplot(data = df_tot, aes(instant, cnt)) +
  geom_point(colour = "red", shape= 1) + geom_line(aes(instant, m_lm$fitted.values), color = "blue") +
  geom_line(aes(instant, m$y), colour = "green") +
  ggtitle("Original data, cubic spline regression, unpenalized pline regrssion")
g_lm
```



Here is the color code :

- Red : initial data
- Green : smooth.spline() regression
- Blue : bs & lm regression

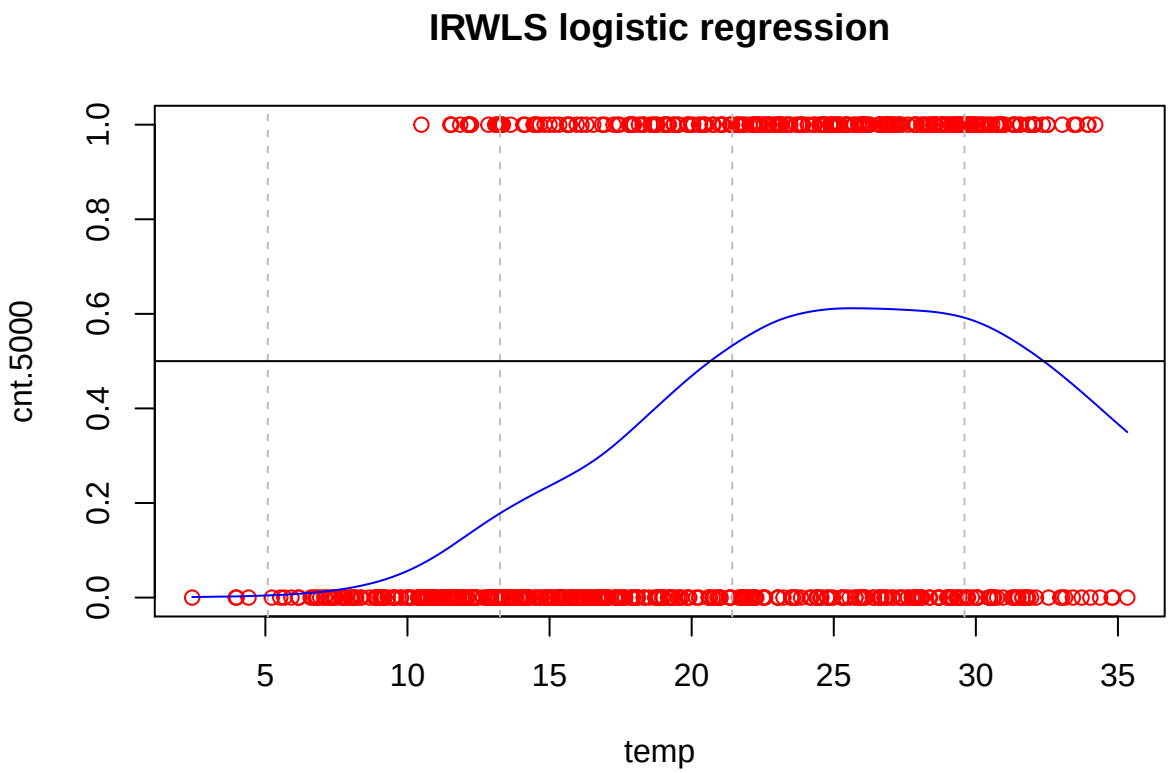
Excersice 2

```
source('IRWLS_logistic_regression.R')

#Building the vector cnt.5000
cnt.5000 <- bikes$cnt > 5000
for (k in 1:length(cnt))
{
  if (cnt.5000[k] == TRUE) {cnt.5000[k] = 1} else {cnt.5000[k] = 0}
}
#First, We sort the data according to the explanatory variable.
x <- bikes$temp
y <- cnt.5000
sx <- sort(x,index.return =TRUE)
x <- sx$x
y <- y[sx$ix]

#Now we can fit the model
```

```r
my.spline.glm <-
  logistic.IRWLS.splines(x,y,
                         df=6, #df = 6 as demanded in instructions
                         all.knots=FALSE, plts = FALSE)
#Plot
plot(x,y,col=2,xlab="temp",ylab="cnt.5000",main = "IRWLS logistic regression")
abline(v=my.knots,lty=2,col="grey")
lines(x,my.spline.glm$fitted.values,col=4)
abline(h=0.5)
```




```r
probability = my.spline.glm$fitted.values

prob_above_50 <- (probability > 0.5)
indeces<-which(prob_above_50)
range<- c(x[indeces[1]], x[indeces[length(indeces)]])
range
```

```
## [1] 20.73915 32.35585
```

The temperatures for which the chance of more than 5000 bikes are rented is 20.74 to 32.36.