Assignment 7
Smoothing and regression splines: Bikes in Washington

Víctor Duque, Geraldo Gariza, Jan Leyva and Andreu Meca
Universitat Politècnica de Catalunya

28th of March, 2021

---

1. Consider the nonparametric regression of cnt as a function of instant. Estimate the regression function m(instant) of cnt as a function of instant using a cubic regression splines estimated with the R function smooth.splines and choosing the smoothing parameter by Generalized Cross Validation

```
sm_sp <- smooth.spline(x = (bikes$instant), y = (bikes$cnt),
                        cv = TRUE, all.knots = FALSE)
```

a) Which is the value of the chosen penalty parameter λ?

The penalty parameter lamda is: 8.7334916×10^{-8}

b) Which is the corresponding equivalent number of degrees of freedom df?

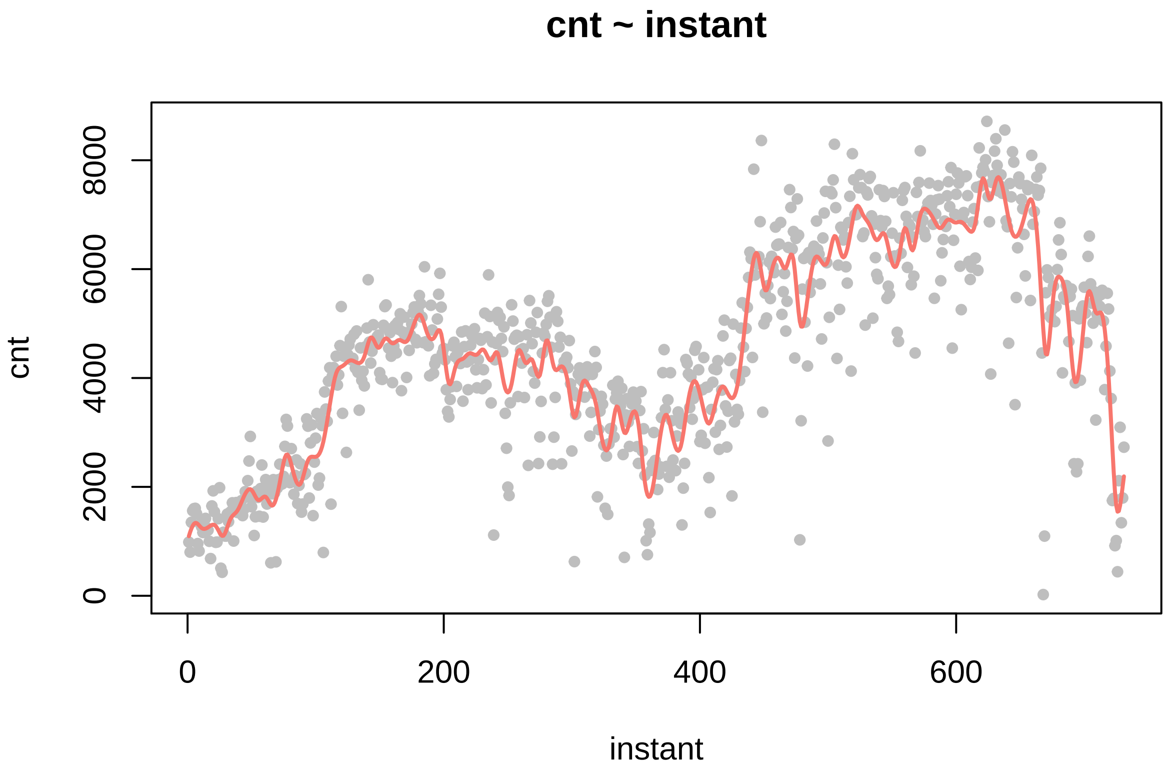
Degress of freedom (Df): 95.6803912

c) How many knots have been used?

Number of knots: 134

d) Give a graphic with the scatter plot and the estimated regression function m(instant)

```
plot(x = (bikes$instant), y = (bikes$cnt), col = "grey", pch = 20,
     xlab = "instant", ylab = "cnt", main = "cnt ~ instant")
lines(sm_sp,col="#F8766D", lwd=2)
```



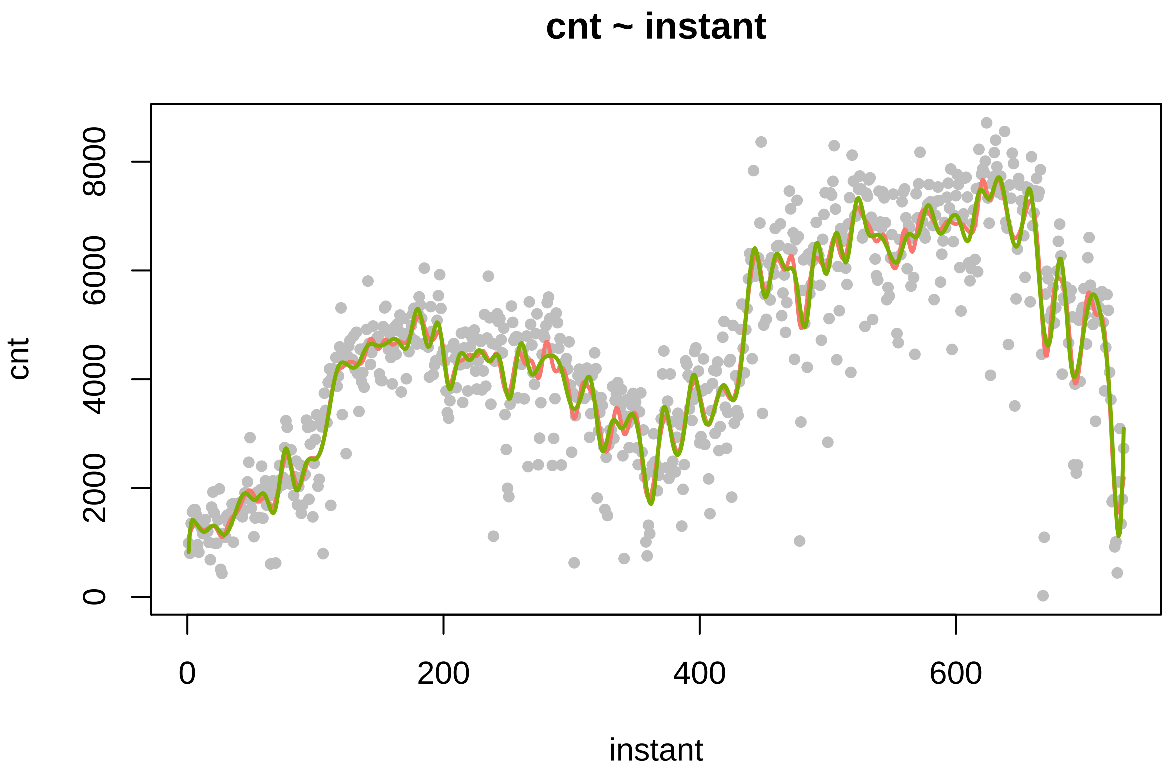
As we can see the model fit the data extremely good, it seems to be very sensitive to the extrem points which may cause the overfitting.

e) (Optional) Estimate now m(instant) by unpenalized regression splines combining the R functions bs and lm

```
my_knots <- quantile(bikes$instant,
                     ((1:(as.numeric(sm_sp$df)-4))-.5)/(as.numeric(sm_sp$df)-4))
lm_fit <- lm(bikes$cnt ~ bs(bikes$instant,knots = my_knots))
```

f) (Optional) Give a graphic with the scatter plot and the two estimated regression functions

```r
plot(x = (bikes$instant), y = (bikes$cnt), col = "grey", pch = 20,
     xlab = "instant", ylab = "cnt", main = "cnt ~ instant")
lines(sm_sp,col="#F8766D",lwd=2)
lines(x = bikes$instant, y = lm_fit$fitted.values, col="#7CAE00",lwd=2)
```



Without further research it seems that the second model fits very similar to the previuos one. So it is possible that we have again an overfiting too.

2. The script IRWLS logistic regression.R includes the definition of the function logistic.IRWLS.splines performing nonparametric logistic regression using splines with a IRWLS procedure. The basic syntax is the following:

   logistic.IRWLS.splines(x=. . . , y=. . . , x.new=. . . , df=. . . , plts=TRUE)

   where the arguments are the explanatory variable x, the 0-1 response variable y, the vector x.new of new values of variable x where we want to predict the probability of y being 1 given that x is equal to x.new, the eauivalent number of parameters (or model degrees of freedom) df, and the logical plts indicating if plots are desired or not. Define a new variable cnt.5000 taking the value 1 for days such that the number of total rental bikes is larger than or equal to 5000, on 0 otherwise.

```r
source("IRWLS_logistic_regression.R")
attach(bikes)
```

```r
# Data:
x <- temp
y <- cnt

sx <- sort(x,index.return =TRUE)
x <- sx$x
y <- y[sx$ix]
```

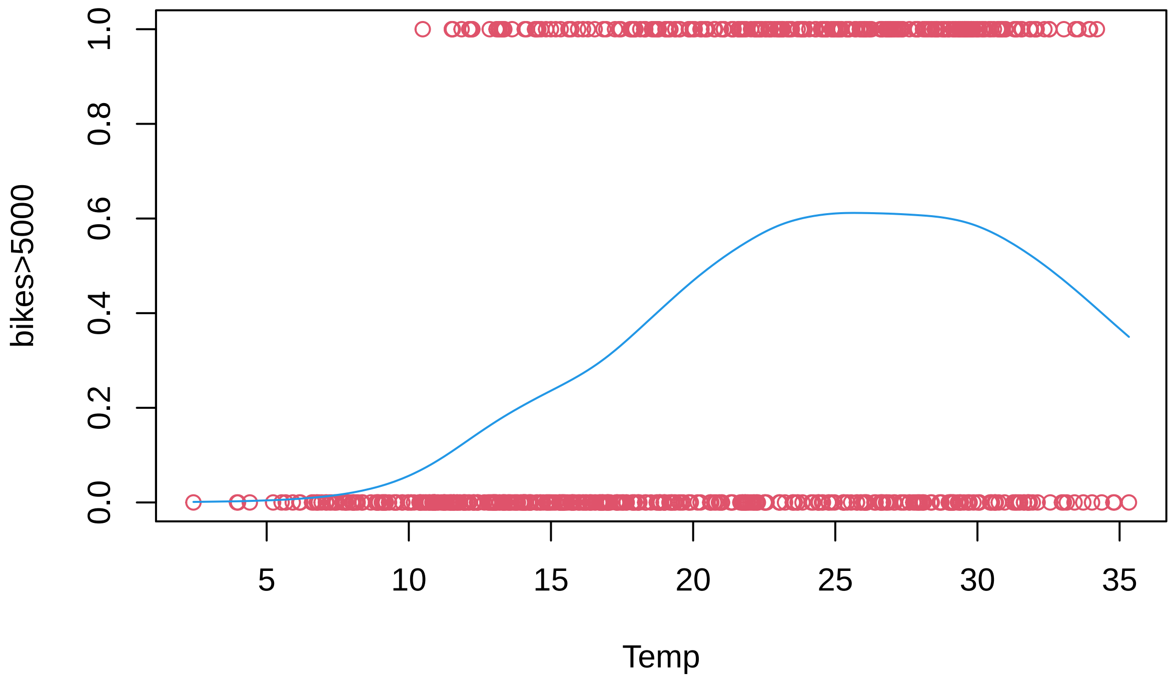
a) Use the function logistic.IRWLS.splines to fit the non-parametric binary regression cnt.5000 as a function of the temperature, using df=6. In which range of temperatures is Pr(cnt >= 5000|temp) larger than 0.5?

```r
cnt.5000 <- c()
for(i in 1:nrow(bikes)){
  cnt.5000[i] <- ifelse(y[i] > 5000, 1, 0)
}
```

```r
splines1<-logistic.IRWLS.splines(y=cnt.5000, x=x, df=6)
```

Plot:

```
plot(x,cnt.5000,col=2,xlab="Temp",ylab="bikes>5000")
lines(x,splines1$fitted.values,col=4)
```



```
range.temp<-x[which(splines1$predicted.values>0.5)]
summary(range.temp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   20.74   23.70   26.67   26.55   29.34   32.36
```

```
pred_logistic <-ifelse(splines1$predicted.values>0.5, 1, 0)
good_fit <-length(which(pred_logistic == cnt.5000))
```

Good predicted by the model: 68.8098495

```
confusion_matrix<-confusionMatrix(data =
    as.factor(cnt.5000), as.factor(pred_logistic))
confusion_matrix$table
```

```
##           Reference
## Prediction   0   1
##          0 303 142
##          1  86 200
```

```
min(range.temp)
```

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

```
max(range.temp)
```

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

The minium value that the model predict 1's is 20.739 and the maxium is 32.35.

b) Choose the parameter df by k-fold log-likelihood cross validation with k = 5 and using df.v = 3:15 as the set of possible values for df.

```
loglik.CV <- function(k, df){

  fold <- sample(1:k, length(x), replace = T)

  for(i in 1:k){
    pred_values <- logistic.IRWLS.splines(y=cnt.5000[fold!=i], x=x[fold!=i],
      x.new = x[fold==i], df=df)$predicted.values
    if(i == 1){
      pred <- as.data.frame(pred_values)
    }else{
      pred <- rbind(pred, as.data.frame(pred_values))
    }
  }

  return((sum(log(exp(-pred$pred_values)*
                    (pred$pred_values^x)
                  /factorial(x))))/length(pred))
}



loog_lik_CV <- function(k, df.v){
  loglik_res <- rep(NA, 12)

    for(df in df.v){
      loglik_res[df-2] <- loglik.CV(k = 5, df = df)
    }

return(cat('The degree of freedom is:',
             df.v[which.min(loglik_res)]))
}
```

```
loog_lik_CV(k = 5, df = 3:15)
```

```
## The degree of freedom is: 15
```

The degrees of freedom chosen by k-fold log-likelihood cross validation is 15.

```
good_log_spliness<-logistic.IRWLS.splines(y=cnt.5000, x=x,
                                           df=15, plt = F)
pred_logistic <-ifelse(good_log_spliness$predicted.values>0.5, 1, 0)
good_fit <-length(which(pred_logistic == cnt.5000))

cat('The goodness of fit is:',good_fit/(length(cnt.5000))*100)
```

```
## The goodness of fit is: 69.08345
```

As we can see using 15 degrees of freedom we have a better performance of the function, increasing the goodness of fit.