Smoothing and regression splines

Biel Caballero Vergés, Svenja Menzenbach and Kleber Enrique Reyes Illescas

2023-11-29

Load data

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 spline estimated with the R function smooth.splines and choosing the smoothing parameter by Generalized Cross Validation.

```
sm.sp <- smooth.spline(x = instant, y = cnt,
                         cv = FALSE, all.knots = FALSE)
sm.sp
```

```
## Call:
## smooth.spline(x = instant, y = cnt, cv = FALSE, all.knots = FALSE)
##
## Smoothing Parameter  spar= 0.24802  lambda= 1.005038e-07 (11 iterations)
## Equivalent Degrees of Freedom (Df): 93.34091
## Penalized Criterion (RSS): 470358491
## GCV: 845608.3
```

```
# Number of knots
sm.sp$fit$nk-2
```

```
## [1] 134
```

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

The value of λ is 1.005038e-07.

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

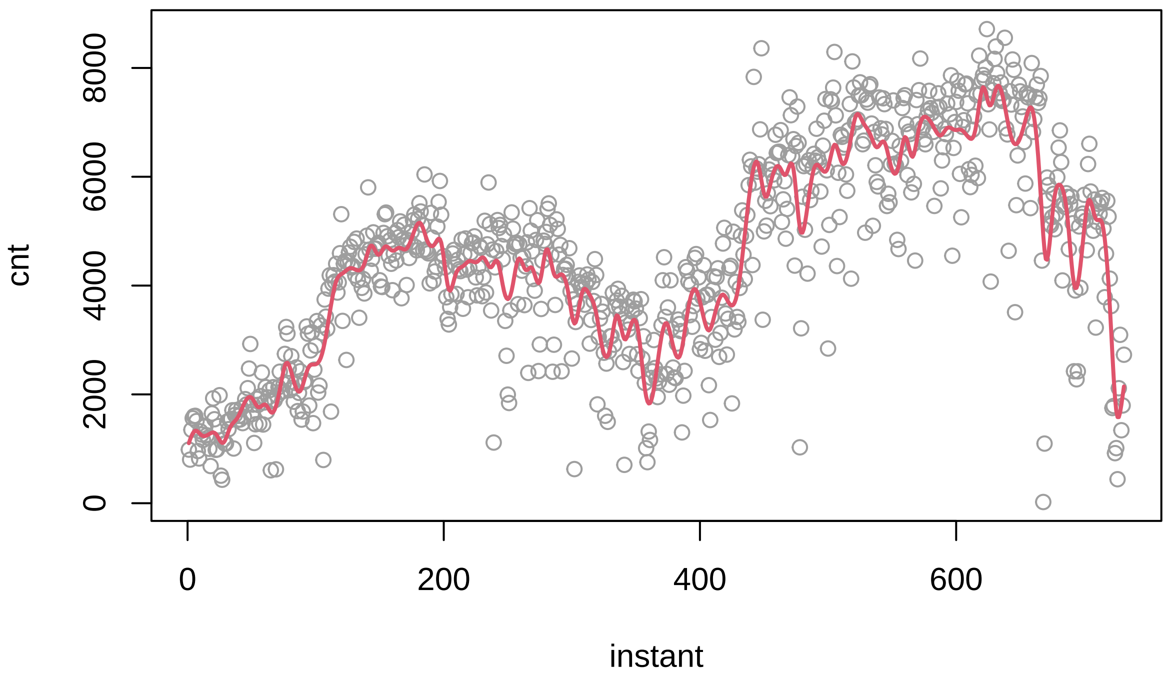
The corresponding equivalent number of degrees of freedom (df) is 93.34091.

c) How many knots have been used?

We have used 134 knots.

d) Give a graphic with the scatter plot and the estimated regression function $\hat{m}$(instant).

```r
plot(instant, cnt, col=8)
#abline(v=sm.sp$fit$min+sm.sp$fit$knot*sm.sp$fit$range, col=8, lty=2)
lines(sm.sp, col=2, lwd=2)
```

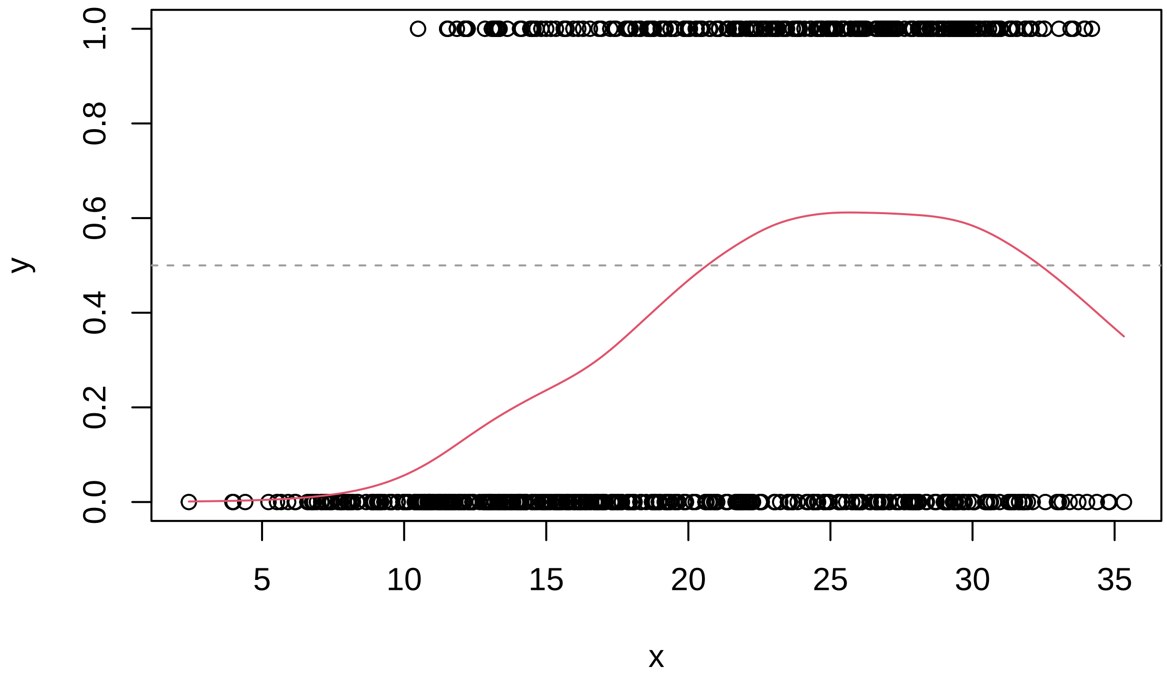


2. The script IRWLS logistic regression.R includes the definition of the function logistic.IRWLS.splines performing nonparametric logistic regression using splines with a IRWLS procedure. The basic syntax is the following: logistic.IRWLS.splines(x=. . . , y=. . . , x.new=. . . , df=. . . , plts=TRUE) 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 equivalent 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")
cnt.5000 <- as.numeric(cnt >= 5000)
```

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
# Sort data according to x
x <- temp
y <- cnt.5000
sx <- sort(x,index.return =TRUE)
x <- sx$x
y <- y[sx$ix]

IRWLS.sp <- logistic.IRWLS.splines(x=x, y=y, x.new = x, df=6)

plot(x, y)
lines(x,IRWLS.sp$fitted.values,col=2)
abline(h=0.5, col=8, lty=2)
```



```r
x.05 <- x[as.numeric(IRWLS.sp$predicted.values >= 0.5) == 1]
x.min <- min(x.05)
x.max <- max(x.05)
print(sprintf("min: %f,  max: %f", x.min, x.max))
```

```
## [1] "min: 20.739153,  max: 32.355847"
```

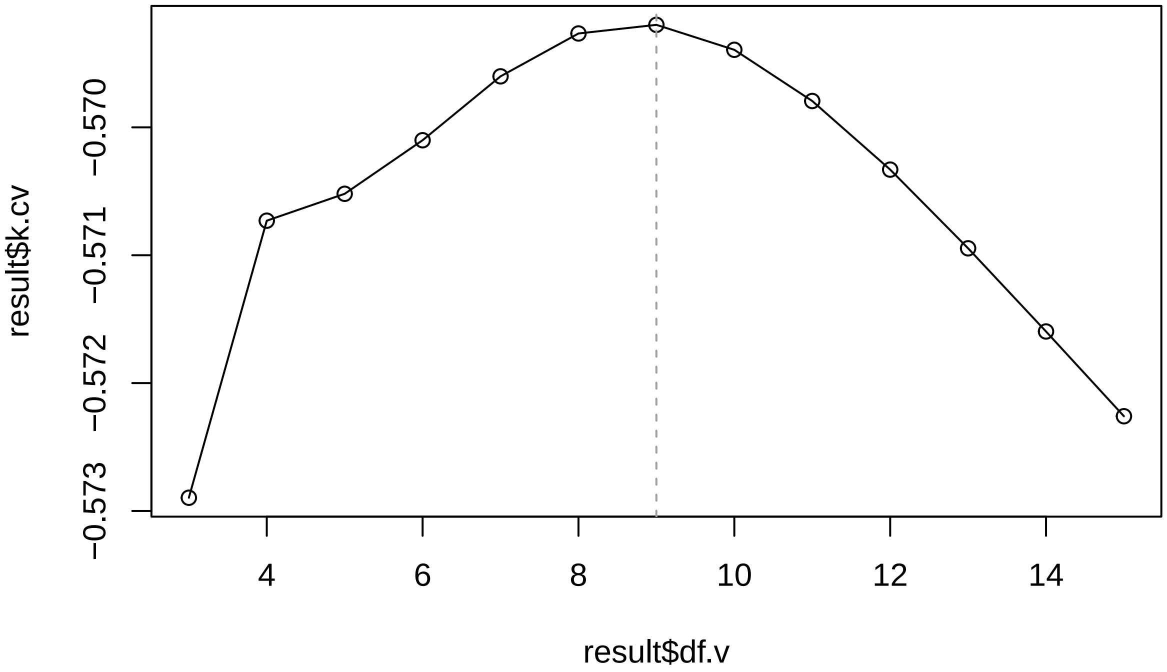
Looking at the returned prediction, the temperatures between 21.7° and 32.4° have Pr(cnt >= 5000|temp) larger than 0.5. We can also check it looking at the plot.

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.

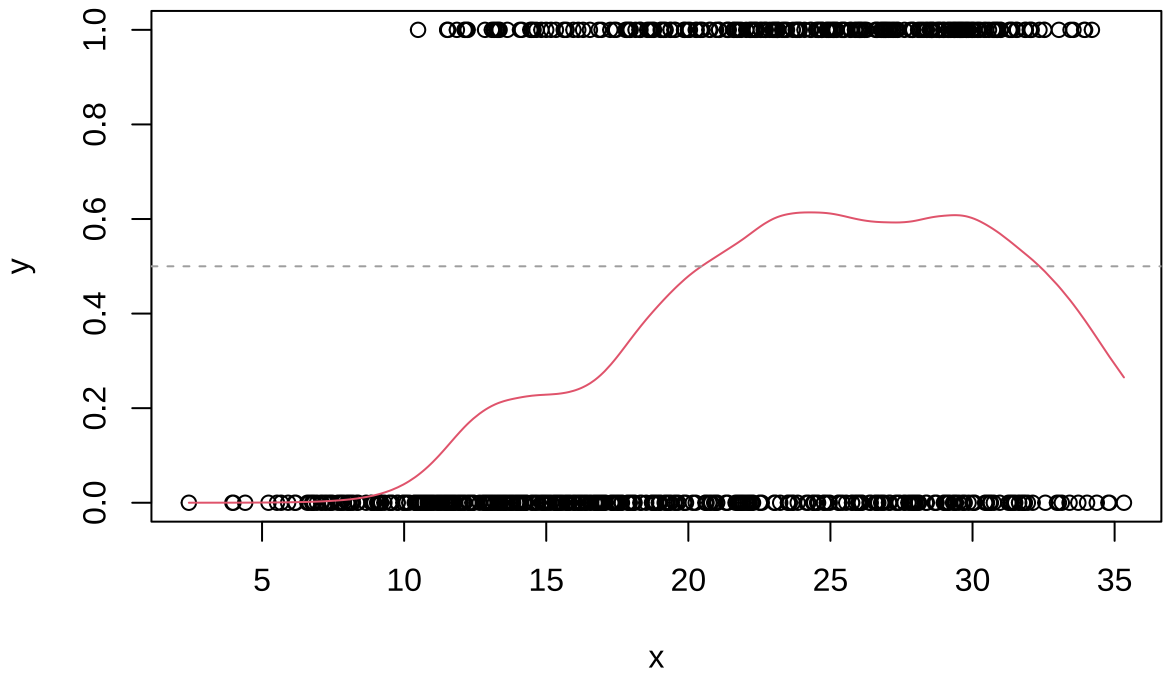
```
k.fold.ll.cv <- function(x,y,df,k=5){
    n <- length(x)
    Ik <- floor((0:(n-1))/(n/k))+1
    cum_sum <- 0

    for (i in (1:k)){
        y.i <- y[Ik==i]
        pred <- logistic.IRWLS.splines(x[Ik!=i], y[Ik!=i],
                                       x.new=x[Ik==i], df=df)$predicted.values
        #cum_sum <- cum_sum + sum((y.i-pred)^2)
        cum_sum <- cum_sum + sum(y.i*log(pred/(1-pred)) + log(1-pred))
    }
    k.cv <- cum_sum/n
    return(k.cv)
}
```

```
df.k.fold.ll.cv <- function(x,y,df.v,k=5){
    n <- length(x)
    perm <- sample(1:n)
    xperm <- x[perm]
    yperm <- y[perm]

    k.cv <- df.v*0
    for (i in (1:length(df.v))){
        df <- df.v[i]
        k.cv[i] <- k.fold.ll.cv(x=xperm, y=yperm, df, k)
    }
    return(list(k=k,df.v=df.v,k.cv=k.cv))
}
```

```
df.v <- 3:15
result <- df.k.fold.ll.cv(x,y,df.v)

selected_df <- result$df.v[which.max(result$k.cv)]

plot(result$df.v, result$k.cv)
lines(result$df.v, result$k.cv)
abline(v = selected_df, col="8", lty=2)
```



```
IRWLS.sp <- logistic.IRWLS.splines(x=x, y=y, x.new = x, df=selected_df)

plot(x, y)
lines(x,IRWLS.sp$fitted.values,col=2)
abline(h=0.5, col=8, lty=2)
```



1.0
0.8
0.6
0.4
0.2
0.0
y
5
10
15
20
25
30
35
x