Trabajo de Evaluación: Regresión de conteo y GAM.

María Marín Cerdá

Máster en Data Science y Big Data

Aprendizaje Estadístico y Modelización

Introducción

En el presente trabajo se ha realizado un estudio de los datos .Ejercicio de regresión de conteo y gam.xlsxçon el fin de encontrar un modelo adecuado para explicar la variable objetivo de manera efectiva y precisa. El análisis comenzó con la exploración de la distribución empírica de la variable objetivo, lo que permitió determinar los enfoques más apropiados para el modelado. A partir de esta evaluación inicial, se seleccionaron y probaron diversos modelos: regresión de conteo de Poisson, modelos aditivos generalizados, regresión polinómica y regresión con splines. Cada uno de estos modelos fue evaluado cuidadosamente para determinar su capacidad para ajustar los datos de manera eficiente y proporcionar una interpretación clara y precisa de la relación entre las variables independientes y la variable objetivo. A lo largo del trabajo, se buscará identificar el modelo que ofrezca el mejor rendimiento en términos de precisión, capacidad de generalización y adecuación al comportamiento observado en los datos. A continuación, se expondrán las conclusiones de los distintos resultados obtenidos al aplicar en R el código aportado en el anexo. Destacar que en todos los contrastes se ha considerado una significación de $\alpha = 0.05$.

Resumen ejecutivo

Variable objetivo

El dataset cuenta con 1999 observaciones de 14 variables, la variable objetivo y 13 variables predictoras.

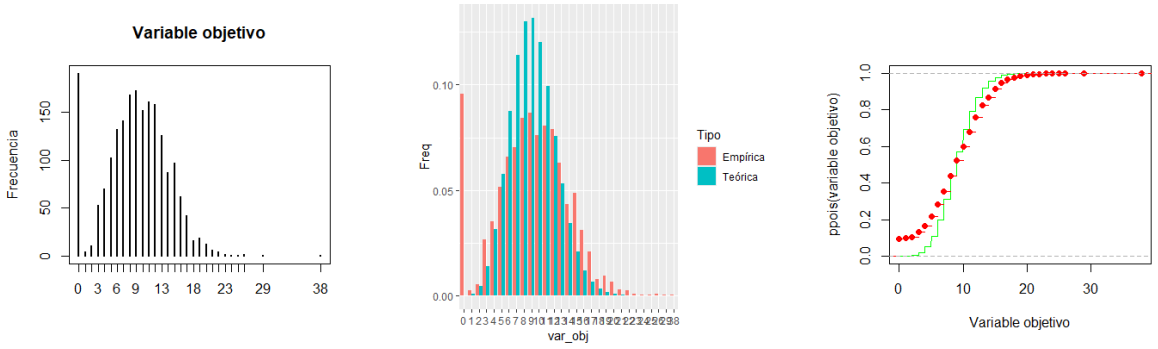


Figura 1: 1.Gráfico de barras 2-3 .Comparación distribución empírica y teórica de una Poisson

Observamos que la variable objetivo toma valores discretos y no negativos. Esto sugiere que podemos estar ante una variable de conteo. El análisis gráfico sugiere que la distribución empírica de la variable objetivo guarda una razonable concordancia con una distribución de Poisson, por lo que se tendrá en cuenta a la hora de construir los distintos modelos.

Modelos Lineales

Comenzamos con un modelo lineal generalizado de la familia Poisson con todas las variables. Se ejecuta un test de dispersión en el que se acepta la hipótesis de igualdad de media y varianza, lo que apoya a la idea de que la variable objetivo siga una distribución Poisson. El modelo de regresión con todas las variables obtiene una devianza explicada del 98.27 %. Se obtiene un buen resultado. No obstante, se verifica que no existe multicolinealidad entre las variables para asegurar que los resultados obtenidos no han sido alterados. Tomando un nivel de significación del 0.05 las variables significativas del modelo son X_1, X_4 y X_{11}, por lo que se plantea un modelo de regresión lineal generalizado con dichas variables. La devianza residual aumenta levemente

pero se consigue un menor valor de AIC. Tras ejecutar el test ANOVA, concluimos que no hay evidencias significativas de explicabilidad entre el modelo con todas las variables y el modelo más simple, por lo que nos quedaremos con el modelo que utiliza las variables X_1, X_4 y X_{11}.

Modelos Aditivos Generalizados

Comenzamos realizando un primer modelo considerando todas las variables. El único término suave significativo que obtenemos es $s(X_{11})$. Se obtiene una devianza del 99.699 % y un R^2_{adj} del 0.9968. Se estudió la concurvidad del modelo (una extensión de la multicolinealidad para modelos aditivos) porque puede afectar seriamente la interpretación de los efectos no lineales de las variables explicativas. Para el siguiente modelo se eliminaros las variables X_1 y X_4 ya que presentaron un coeficiente de concurvidad elevado. También se descartó la variable categórica X_{13}, ya que no era sigificativa y no es relevante para el estudio de la concurvidad.Volvimos a estudiar la concurvidad del nuevo modelo resultante y obtuvimos resultados similares. En esta ocasión descartamos las variables X_6 y X_7. Reiterando este proceso llegamos a un modeo sin concurvidad considerando únicamente X_2, X_3, X_8 , X_9 y X_{11}. No obstante, el último modelo seguía considerando como única variable significativa X_{11}. Por esa razón se construye el modelo GAM, que en el anexo se denota como **gm5**, con la variable X_{11}. En este modelo se obtuvo un AIC de 7400.386 mientras que en el modelo con todas las variables se obtuvo un AIC de 7427.414. La devianza explicada que se ubtuvo en el modelo **gm5** fue de 99.68 % y un R^2_{adj} de 0.9967. Tras realizar el test ANOVA, confirmamos que todos los modelos realizados considerando más variables son modelos anidados de **gm5**. Dado que desconocemos la naturaleza de las variables, realizamos otros modelos GAM cambiando el tipo de spline. En el siguiente modelo dado (denotado por **gm6**), se considera el spline cúbico de regresión, donde se obtuvo resultados muy similares.

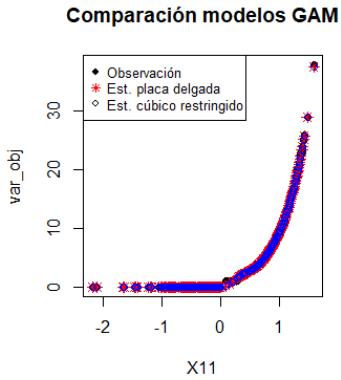


Figura 2: Comparación de dos modelos GAM

Otro modelo que se ha considerado, es aquel incorpora un término suavizado para X_{11}, permitiendo que su efecto varíe en función de la variable categórica X_{13}. Este modelo busca

capturar posibles relaciones no lineales y explorar cómo el impacto de X_{11} cambia según los valores X_{13}.

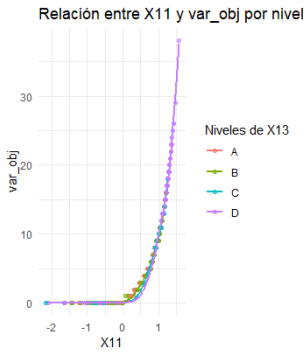


Figura 3: Relación de la variable X_{11} y variable objetivo diferenciada por niveles de X_{13}

El gráfico muestra que no hay señales de una interacción significativa entre X_{11} y X_{13} pues no hay una separación clara entre las curvas de cada categoría. Al realizar el modelo, observamos que se mantuvo el valor de R^2_{adj} y creió el AIC. Concluimos por tanto que el mejor

Modelos	gm	gm2	gm3	gm4	gm5	gm6	gm7
AIC	7427.414	7418.143	7414.175	7410.329	7400.386	7401.010	7428.984
R^2_{adj}	0.9968845	0.9967409	0.996372	0.996319	0.996316	0.996309	0.996765
Devianza explicada	0.9969545	0.9968473	0.9968424	0.9968200	0.9968116	0.9967429	0.9961016

Tabla 1: Tabla resumen: Modelos GAM.

modelo aditivo generalizado es aquel que considera únicamente la variable X_{11} (**gm5**). Para descartar el sobreajuste, dividimos los datos en conjunto de entrenamiento y conjunto test. Obtuvimos buenos resultados del modelo, pues considerando como datos de entrada el conjunto de entrenamiento las predicciones del conjunto test fueron muy acertadas.

Regresión polinómica

Como concluimos que con la variable X_{11} es suficiente para explicar la variable objetivo, los modelos de regresión polinómica han sido construidos considerando dicha variable variando el grado del molinomio. Comenzamos por grado 6 y observamos que todos los contrastes individuales fueron rechazados. Fuimos aumentando el grado hasta obtener contrastes individuales que no se rechazaban. Tras aplicar el test ANOVA concluimos que el último modelo que tenía cambios significativos, era el de grado 8, denotado como **regpoly2**. Todos los valores de R^2_{adj} son muy similares por lo que tomaremos aquel con menor AIC. Al igual que en el apartado anterior comprobamos que no había sobreajuste.

Modelo	regpoly	regpoly2	regpoly3
AIC	683.6137	670.896	672.7515
R^2_{adj}	0.9963178	0.9966325	0.9963462

Tabla 2: Tabla resumen: Regresión polinómica

Regresión con splines

Razonando de manera análoga al apartado anterior, solo se consideró la variable X_{11}. En el primer modelo se consideró los parámetros que vienen por defecto, pero obtuvimos resultados bastantes mejorables. Tras una representación gráfica, se contempló que variando los nodos considerados en el modelo podríamos tener mejores resultados. Tomamos el modelo de regresión

Modelo	regspline1	regspline2
AIC	3578.041	897.5015
R^2_{adj}	0.9855	0.9962

Tabla 3: Tabla resumen: Regresión con splines

cuyos nodos han sido introducidos manualmente, con **regspline2**. Al igual que en los modelos anteriores se comprobó que no había sobreajuste.

Selección final

Una vez visto todos los modelos y seleccionando los mejores de cada tipo, vamos a decidir con qué modelo nos quedaremos. El Modelo Aditivo Generalizado aporta mejores resultados que el Modelo Lineal Generalizado ya que este último presenta un AIC mayor y una devianza explicada menor. El modelo GAM da muy buenos resultados, sin embargo hemos obtenido modelos en los que el AIC disminuye y el R^2_{adj} se mantiene. En la regresión polinómica el AIC es de 670.896 y en la regresión con splines, 897.5015. En ambos modelos el R^2_{adj} toma un valor de 0.996. Concluimos por tanto que el modelo que mejor explica a la variable objetivo disminuyendo su complejidad es el modelo de regresión polinómica de grado 8.

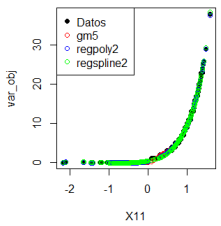


Figura 4: Comparación entre los datos y las estimaciones

---

output: pdf_document

Anexo

Procedemos a la carga de datos.

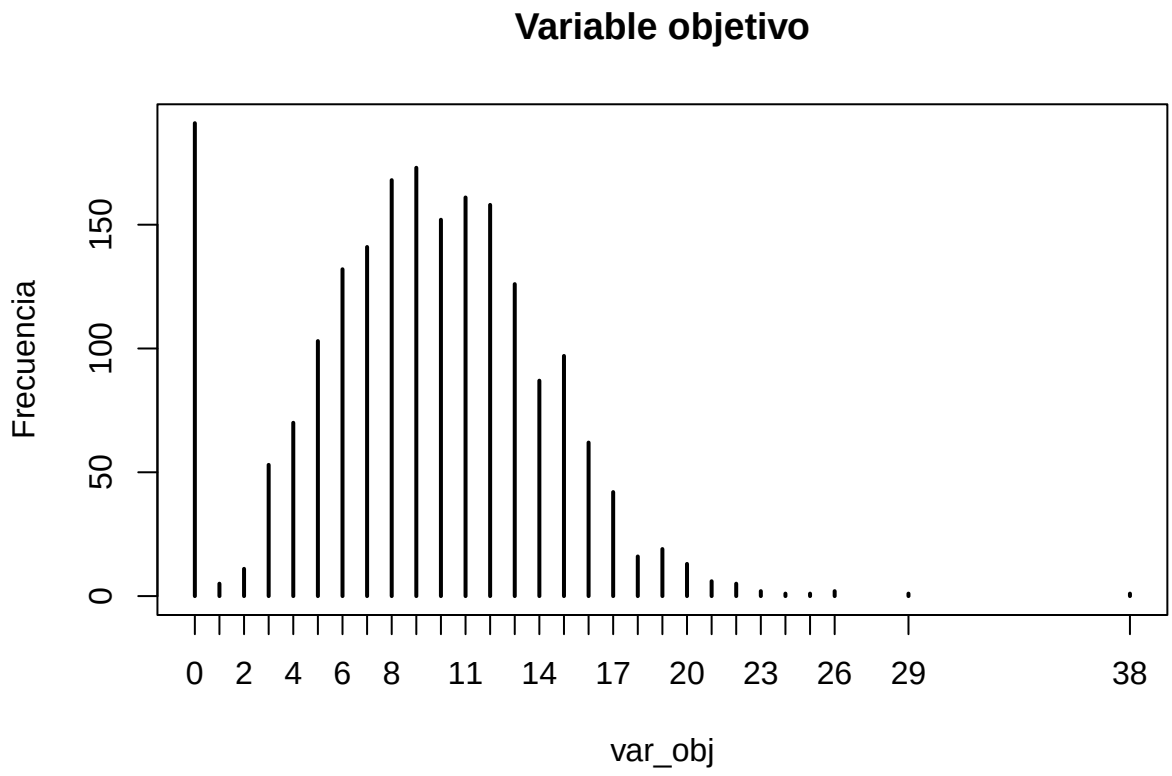
```
#install.packages("readxl")
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.4.3
```

```
datos<-read_excel("Ejercicio regresión de conteo y gam.xlsx")
datos$X13<-as.factor(datos$X13)
attach(datos)
```

Veamos la distribución de la variable objetivo.

```
plot(table(var_obj),ylab = "Frecuencia",main="Variable objetivo")
```



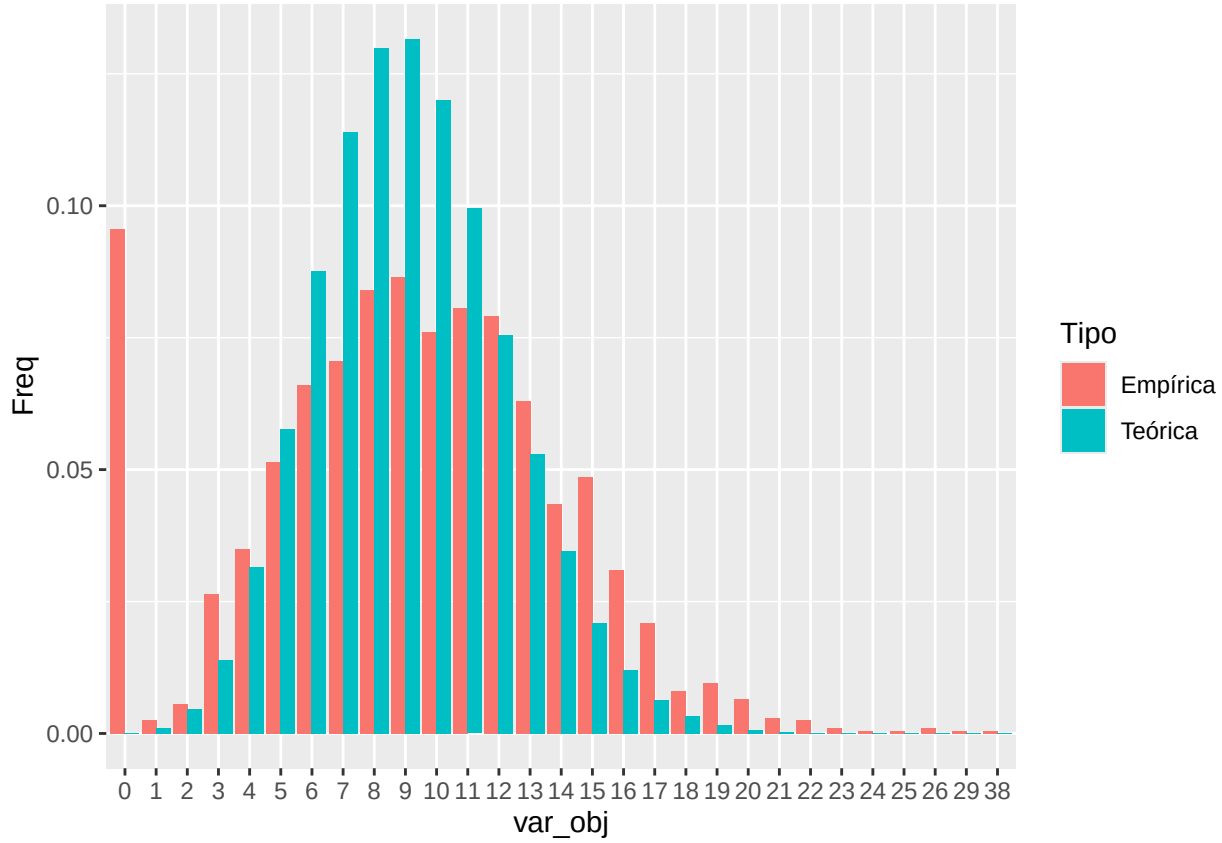
Observamos que la variable objetivo es discreta y no negativa. Por ello, nos podemos plantear una regresión con datos de conteo. Comparemos la distribución empírica de la variable objetivo con la función de distribución teórica de una Poisson.

```
df1<-data.frame(table(datos$var_obj))
names(df1)<-c("var_obj","Freq")
df1$Tipo<-"Empírica"
df1$Freq<-df1$Freq/sum(df1$Freq)
```

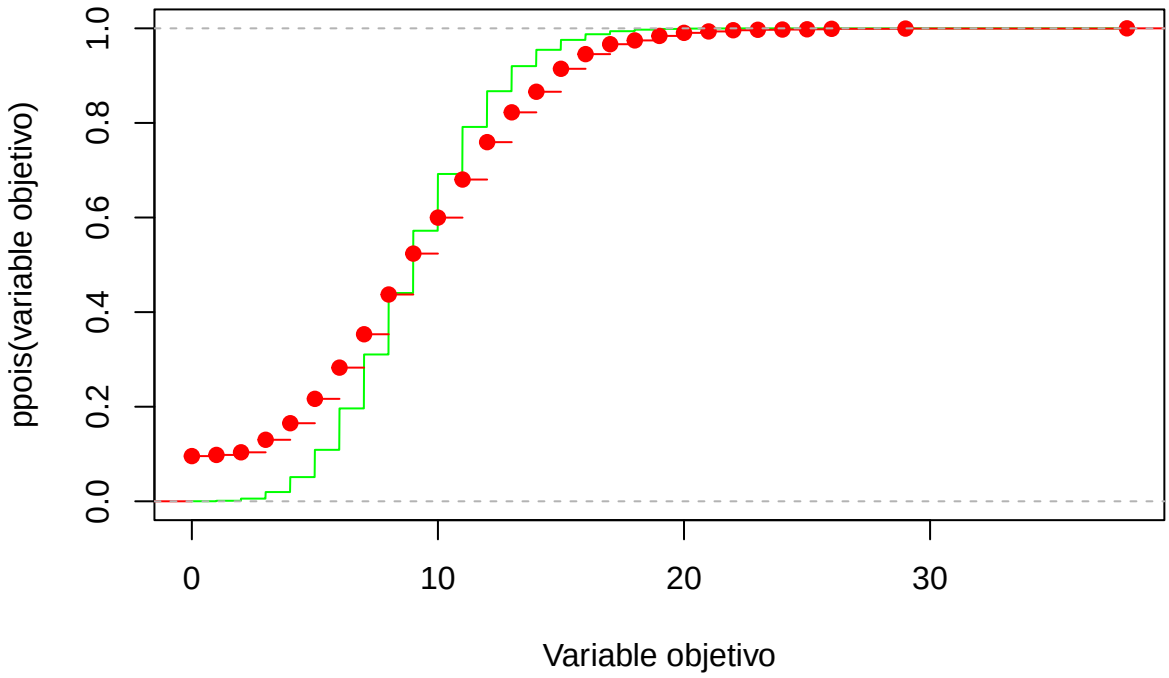
```r
mvar_obj<-mean(datos$var_obj)

valores<-sort(unique(datos$var_obj))
df2<-data.frame(var_obj=valores,dpois(valores,lambda=mvar_obj))
names(df2)[2]<-"Freq"
df2$Tipo<-"Teórica"
df<-rbind(df1,df2)

library(ggplot2)
ggplot(data=df, aes(x=var_obj, y=Freq, fill=Tipo)) +
  geom_bar(stat="identity", position=position_dodge())
```




```r
xempp <- seq(min(datos$var_obj), max(datos$var_obj), by=0.01)
plot(xempp, ppois(xempp, lambda=mvar_obj), type="l", col="green", xlab="Variable objetivo",
     ylab="ppois(variable objetivo)")
plot(ecdf(datos$var_obj), col="red",add=TRUE)
```



El análisis gráfico sugiere que la distribución empírica de la variable objetivo guarda una razonable concordancia con una distribución de Poisson, lo que indica que este modelo podría ser apropiado para la regresión.

Modelo Lineal Generalizado

Comenzaremos analizando el modelo con todas las variables.

```
regre_todas<-glm(var_obj~.,family = "poisson",data=datos)
#install.packages("AER")
library(AER)
```

```
## Warning: package 'AER' was built under R version 4.4.3
## Cargando paquete requerido: car
## Cargando paquete requerido: carData
## Cargando paquete requerido: lmtest
## Warning: package 'lmtest' was built under R version 4.4.2
## Cargando paquete requerido: zoo
## Warning: package 'zoo' was built under R version 4.4.2
##
## Adjuntando el paquete: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Cargando paquete requerido: sandwich
```

```
## Warning: package 'sandwich' was built under R version 4.4.2
```

```
## Cargando paquete requerido: survival
```

```
dispersiontest(regre_todas)
```

```
##
##  Overdispersion test
##
## data:  regre_todas
## z = -112.77, p-value = 1
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##  0.1257553
```

La función `dispersiontest` del paquete de AER se utiliza para realizar una prueba de dispersión del modelo de Poisson. Se acepta que la media y la varianza coinciden.

La devianza mide cuánto se aleja un modelo de la perfección. Matemáticamente , la devianza se definde como

$$D = 2 \times (logL_{modelosaturado} - logL_{modeloajustado})$$

donde el modelo saturado predice cada punto perfectamente. Por ello a mayor devianza, menor capacidad de predicción posee el modelo estimado. La devianza nula mide qué tan bien el modelo más simple (solo con el intercepto) explica la variable respuesta. Sirve como punto de referencia para evaluar si añadir predictores mejora la capacidad explicativa del modelo. La devianza residual es la cantidad de variabilidad que el modelo no ha conseguido explicar. Un valor muy bajo de devianza residual sugiere que el modelo ajusta bastante bien los datos. La devianza explicada mide qué proporción de la devianza total ha sido explicada por el modelo.

$$D_{expl} = \frac{\text{Devianza nula - Devianza residual}}{\text{Devianza nula}}$$

```
summary(regre_todas)
```

```
##
## Call:
## glm(formula = var_obj ~ ., family = "poisson", data = datos)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.773e-01  8.768e-02  -3.163 0.001563 **
## X1          -8.068e-02  2.300e-02  -3.508 0.000452 ***
## X2          -4.329e-06  4.230e-06  -1.023 0.306110
## X3           1.112e-04  6.197e-04   0.179 0.857576
## X4          -3.066e-02  7.399e-03  -4.144 3.42e-05 ***
## X5           3.641e-06  5.787e-03   0.001 0.999498
## X6          -9.099e-04  5.363e-03  -0.170 0.865274
## X7          -1.124e-03  5.658e-03  -0.199 0.842517
## X8          -3.301e-02  2.578e-02  -1.280 0.200408
## X9          -1.309e-02  1.290e-02  -1.015 0.310028
## X10          4.115e-05  4.383e-04   0.094 0.925196
## X11          3.190e+00  1.067e-01  29.904  < 2e-16 ***
## X12         -9.061e-06  5.126e-04  -0.018 0.985897
## X13B         1.260e-03  2.994e-02   0.042 0.966433
## X13C         1.051e-02  3.326e-02   0.316 0.751904
```

```
## X13D         1.535e-02  4.259e-02   0.360 0.718571
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 6754.41  on 1998  degrees of freedom
## Residual deviance:  116.24  on 1983  degrees of freedom
## AIC: 7513.4
##
## Number of Fisher Scoring iterations: 5
```

```
regre_todas$null.deviance
```

```
## [1] 6754.412
```

```
regre_todas$deviance
```

```
## [1] 116.2448
```

```
(dev_expl<-(regre_todas$null.deviance-regre_todas$deviance)/regre_todas$null.deviance)
```

```
## [1] 0.9827898
```

Observamos que la devianza residual es bastante baja, por lo que la devianza explicada es bastante alta. Sin embargo hay numerosas variables que el modelo considera insignificativas. Comprobemos si hay multicolinealidad entre las variables y están alterando los resultados del modelo.

```
library(car)
vif(regre_todas)
```

```
##         GVIF Df GVIF^(1/(2*Df))
## X1  4.397948  1        2.097129
## X2  1.029181  1        1.014486
## X3  1.037793  1        1.018721
## X4  7.465503  1        2.732307
## X5  4.850713  1        2.202433
## X6  4.099120  1        2.024628
## X7  4.102873  1        2.025555
## X8  1.035395  1        1.017544
## X9  1.025555  1        1.012697
## X10 3.417603  1        1.848676
## X11 5.780358  1        2.404237
## X12 1.004210  1        1.002103
## X13 3.672582  3        1.242115
```

Observamos que no hay ninguna variable que presente un Factor de Inflación de la Varianza (VIF) superior a 10, por lo que destacaremos multicolinealidad. Tomando un nivel de significación de 0.05, las variables significativas del modelo son X_1, X_4 y X_{11}. Estudiemos el modelo de regresión con dichas variables.

```
reg<-glm(var_obj~X1+X4+X11,family = "poisson",data=datos)
reg$deviance
```

```
## [1] 120.9635
```

```
summary(reg)
```

```
##
## Call:
```

```
## glm(formula = var_obj ~ X1 + X4 + X11, family = "poisson", data = datos)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.341622   0.056573  -6.039 1.55e-09 ***
## X1          -0.067787   0.015533  -4.364 1.28e-05 ***
## X4          -0.027677   0.005096  -5.431 5.61e-08 ***
## X11          3.135077   0.100352  31.241  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 6754.41  on 1998  degrees of freedom
## Residual deviance:  120.96  on 1995  degrees of freedom
## AIC: 7494.2
##
## Number of Fisher Scoring iterations: 4
```

La devianza residual ha aumentado levemente en este modelo. No obstante hemos conseguido disminuir el AIC y considerando un modelo más simple, por lo que podríamos considerar que este segundo modelo es mejor que el modelo que considera todas las variables.

```
regre_todas$aic
```

```
## [1] 7513.438
```

```
reg$aic
```

```
## [1] 7494.157
```

Calculemos si la diferencia de devianza entre los dos modelos es estadísticamente significativa.

```
anova(regre_todas,reg)
```

```
## Analysis of Deviance Table
##
## Model 1: var_obj ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 +
##     X11 + X12 + X13
## Model 2: var_obj ~ X1 + X4 + X11
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      1983     116.25
## 2      1995     120.96 -12  -4.7187   0.9667
```

No hay evidencias para rechazar la hipótesis nula, es decir, la diferencia entre el modelo `regre_todas` y `reg` no es estadísticamente significativa. Es por tanto, que frente al modelo de todas las variables, se considera mejor el modelo con las variables X_1, X_4 y X_{11} al ser más simple y lograr la misma explicabilidad de la variable objetivo. Veamos ahora si considerar una relación no lineal aporta mejores resultados.

Modelos aditivos generalizados

Plantearemos el primer modelo utilizando todas las variables.

```
#install.packages("gamair")
library(gamair)
```

```
## Warning: package 'gamair' was built under R version 4.4.3
```

```
library(mgcv)
```

```
## Cargando paquete requerido: nlme
```

```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
gm<-gam(var_obj~s(X1)+s(X2)+s(X3)+s(X4)+s(X5)+s(X6)+s(X7)+s(X8)+s(X9)+s(X10)+s(X11)+s(X12)+factor(X13),
```

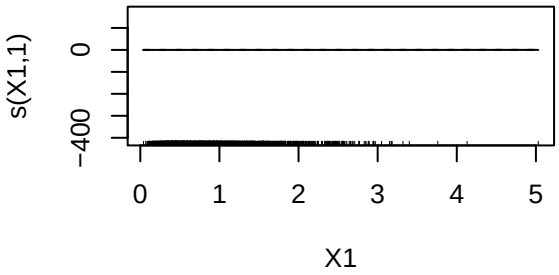
Por defecto, la función `gam()` de la librería mgcv utiliza splines suavizado penalizado y utilizando spline de regresión de placas delgadas como base. Toma `scale=1`. Dado que hemos asumido igualdad de media y varianza podemos dejar dicho valor por defecto.

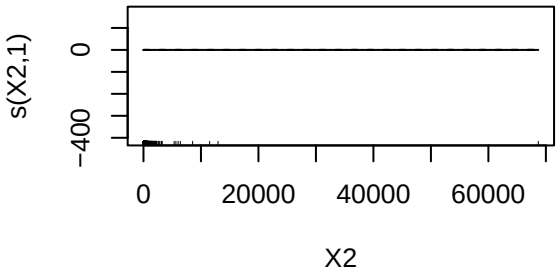
```
summary(gm)
```

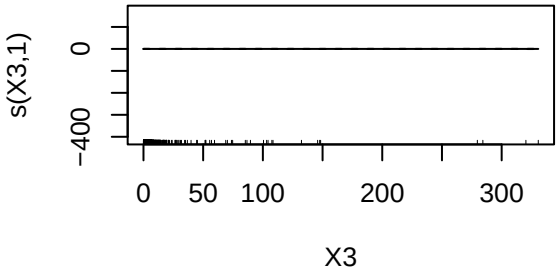
```
##
## Family: poisson
## Link function: log
##
## Formula:
## var_obj ~ s(X1) + s(X2) + s(X3) + s(X4) + s(X5) + s(X6) + s(X7) +
##     s(X8) + s(X9) + s(X10) + s(X11) + s(X12) + factor(X13)
##
## Parametric coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.137455   1.167482  -0.118    0.906
## factor(X13)B -0.002579   0.032256  -0.080    0.936
## factor(X13)C -0.006410   0.040413  -0.159    0.874
## factor(X13)D -0.009254   0.053960  -0.172    0.864
##
## Approximate significance of smooth terms:
##          edf Ref.df  Chi.sq p-value
## s(X1)  1.000   1.00   0.359   0.549
## s(X2)  1.000   1.00   0.019   0.890
## s(X3)  1.000   1.00   0.019   0.892
## s(X4)  1.000   1.00   0.554   0.457
## s(X5)  1.000   1.00   0.000   0.999
## s(X6)  1.000   1.00   0.008   0.928
## s(X7)  1.000   1.00   0.003   0.955
## s(X8)  1.000   1.00   0.022   0.883
## s(X9)  1.000   1.00   0.112   0.738
## s(X10) 1.000   1.00   0.002   0.969
## s(X11) 5.825   6.01 276.207  <2e-16 ***
## s(X12) 1.000   1.00   0.000   0.988
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.997   Deviance explained = 99.7%
## UBRE = -0.96887  Scale est. = 1         n = 1999
```

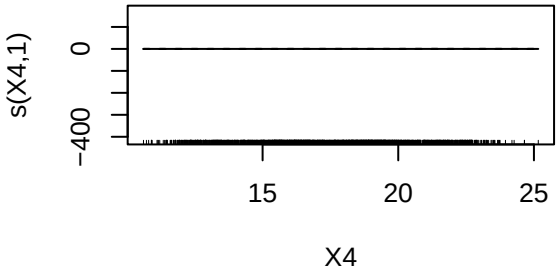
Observamos que el único término suave significativo es $s(X_{11})$ con un `edf` de 5.825, indicando una relación no lineal compleja.

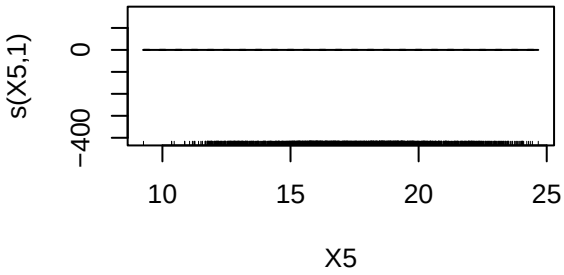
```
par(mfrow=(c(2,2)))
plot(gm)
```

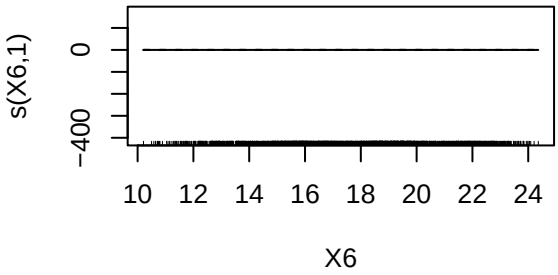

s(X1,1)
0
−400
0
1
2
3
4
5
X1

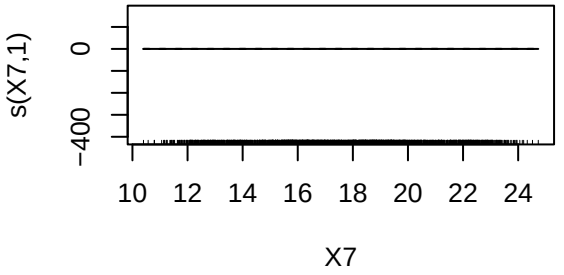

s(X2,1)
0
−400
0
20000
40000
60000
X2

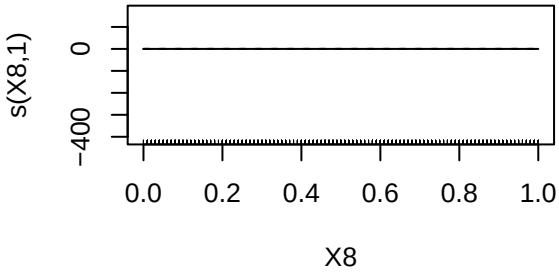

s(X3,1)
0
−400
0
50
100
200
300
X3

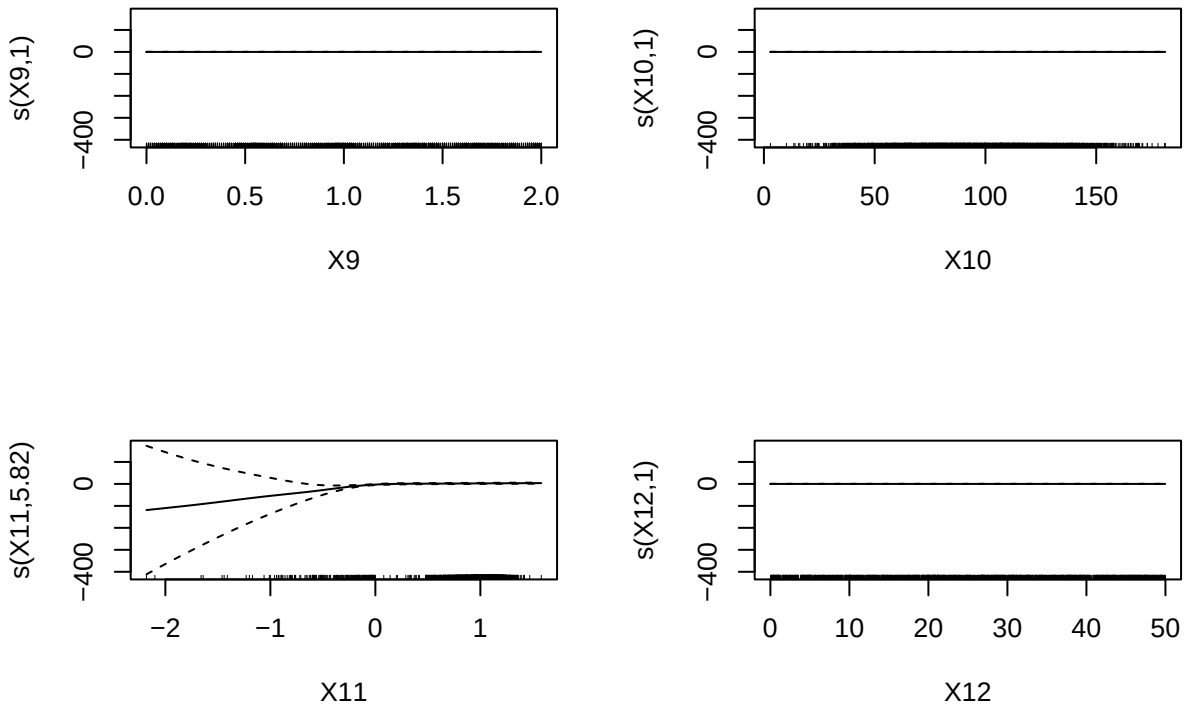

s(X4,1)
0
−400
15
20
25
X4


s(X5,1)
0
−400
10
15
20
25
X5


s(X6,1)
0
−400
10
12
14
16
18
20
22
24
X6


s(X7,1)
0
−400
10
12
14
16
18
20
22
24
X7


s(X8,1)
0
−400
0.0
0.2
0.4
0.6
0.8
1.0
X8




```
summary(gm)$r.sq
```

```
## [1] 0.9968845
```

Efectivamente para el resto de variables considera un efecto lineal (edf = 1), aunque no significativo. Procedamos a estudiar la concurvidad de las variables. La concurvidad en los modelos aditivos generalizados se refiere a una situación en la que uno o más términos suaves en el modelo pueden ser aproximados por otros términos suaves del mismo modelo. Es similar a la colinealidad en los modelos lineales, pero se aplica a términos suaves en lugar de términos lineales. Dado que hemos obtenido un valor R^2_{adj} muy alto, podremos estar ante una situación de concurvidad.

```
concurvity(gm)
```

```
##                para     s(X1)      s(X2)      s(X3)     s(X4)     s(X5)
## worst    0.9774198 0.9499666 0.23495349 0.18468796 0.9169471 0.8582916
## observed 0.9774198 0.8994437 0.05385902 0.11820432 0.9038765 0.8362356
## estimate 0.9774198 0.8074372 0.04882562 0.08872764 0.8429966 0.7651736
##               s(X6)     s(X7)      s(X8)      s(X9)    s(X10)    s(X11)
## worst    0.8562658 0.9188748 0.08753637 0.08533654 0.7652554 0.8857384
## observed 0.8318084 0.8518701 0.07860007 0.06684323 0.7577366 0.5028948
## estimate 0.7641446 0.8056411 0.07508976 0.06542273 0.6602227 0.7332216
##               s(X12)
## worst    0.08084530
## observed 0.05002921
## estimate 0.05002506
```

Un valor cercano a 1 indica concurvidad, mientras que un valor cercano a 0 indica que no existe tal problema. `worst` muestra el peor caso de concurvidad posible para cada término suave. `observed` muestra la concurvidad

observada y `estimated` la estimación. Observamos que hay problemas con varias variables. Comenzaremos eliminando las variables X_1 y X_4. Además eliminaremos la variable X_{13} ya que no se considera significativa.

```
gm2<-gam(var_obj~s(X2)+s(X3)+s(X5)+s(X6)+s(X7)+s(X8)+s(X9)+s(X10)+s(X11)+s(X12),family="poisson",data=da
summary(gm2)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## var_obj ~ s(X2) + s(X3) + s(X5) + s(X6) + s(X7) + s(X8) + s(X9) +
##     s(X10) + s(X11) + s(X12)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1427     1.1677  -0.122    0.903
##
## Approximate significance of smooth terms:
##          edf Ref.df  Chi.sq p-value
## s(X2)  1.000  1.000   0.003   0.958
## s(X3)  1.000  1.000   0.021   0.886
## s(X5)  1.000  1.000   0.053   0.818
## s(X6)  1.000  1.000   0.022   0.881
## s(X7)  1.000  1.000   0.008   0.928
## s(X8)  1.000  1.000   0.004   0.950
## s(X9)  1.000  1.000   0.016   0.898
## s(X10) 1.000  1.000   0.028   0.868
## s(X11) 5.827  6.011 914.317  <2e-16 ***
## s(X12) 1.000  1.000   0.000   0.988
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.997   Deviance explained = 99.7%
## UBRE = -0.97351  Scale est. = 1         n = 1999
```

```
(aic1=AIC(gm))
```

```
## [1] 7427.414
```

```
(aic2=AIC(gm2))
```

```
## [1] 7418.143
```

Hemos obtenido un modelo similar, en el que solo considera como término suave significativo $s(X_{11})$. No obstante, hemos conseguido reducir levemente el AIC al considerar menos variables. Estudiemos la concurvidad de este modelo.

```
concurvity(gm2)
```

```
##                  para      s(X2)      s(X3)     s(X5)     s(X6)     s(X7)
## worst    4.334447e-20 0.12992144 0.17877539 0.7997013 0.8550167 0.9174399
## observed 4.334447e-20 0.03428476 0.10982675 0.7765985 0.8298343 0.8507232
## estimate 4.334447e-20 0.03208764 0.07962028 0.6951953 0.7612835 0.8037094
##               s(X8)      s(X9)    s(X10)    s(X11)     s(X12)
## worst    0.06187265 0.06740811 0.7394179 0.8786842 0.06881729
## observed 0.05397982 0.04621717 0.7320848 0.4961959 0.03715868
```

```
## estimate 0.05269712 0.04627011 0.6355352 0.6891527 0.03760309
```

Sigue habiendo problemas de concurvidad. Sigamos reduciendo el número de variables. Descartemos ahora las variables X_7 y X_6

```
gm3<-gam(var_obj~s(X2)+s(X3)+s(X5)+s(X8)+s(X9)+s(X10)+s(X11)+s(X12),family="poisson",data=datos )
summary(gm3)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## var_obj ~ s(X2) + s(X3) + s(X5) + s(X8) + s(X9) + s(X10) + s(X11) +
##     s(X12)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1396     1.1657   -0.12    0.905
##
## Approximate significance of smooth terms:
##          edf Ref.df  Chi.sq p-value
## s(X2)  1.000  1.000   0.003   0.954
## s(X3)  1.000  1.000   0.021   0.886
## s(X5)  1.000  1.000   0.053   0.818
## s(X8)  1.000  1.000   0.004   0.948
## s(X9)  1.000  1.000   0.016   0.900
## s(X10) 1.000  1.000   0.027   0.870
## s(X11) 5.827  6.011 914.851  <2e-16 ***
## s(X12) 1.000  1.000   0.000   0.994
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.997   Deviance explained = 99.7%
## UBRE = -0.9755  Scale est. = 1         n = 1999
```

```
(aic3=AIC(gm3))
```

```
## [1] 7414.175
```

```
concurvity(gm3)
```

```
##                  para      s(X2)      s(X3)     s(X5)      s(X8)      s(X9)
## worst    4.296278e-20 0.11900571 0.16839882 0.7965262 0.04766792 0.05270884
## observed 4.296278e-20 0.02683089 0.10260517 0.7732515 0.04334792 0.03893141
## estimate 4.296278e-20 0.02462001 0.07105554 0.6915145 0.04171582 0.03893472
##             s(X10)     s(X11)     s(X12)
## worst    0.7358473 0.70772631 0.05221251
## observed 0.7282632 0.02683951 0.02705136
## estimate 0.6305713 0.17772440 0.02775259
```

Sigamos reduciendo el número de variables. Descartemos ahora las variables X_5 y X_{10}

```
gm4<-gam(var_obj~s(X2)+s(X3)+s(X8)+s(X9)+s(X11)+s(X12),family="poisson",data=datos )
summary(gm4)
```

```
##
## Family: poisson
```

```
## Link function: log
##
## Formula:
## var_obj ~ s(X2) + s(X3) + s(X8) + s(X9) + s(X11) + s(X12)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1435     1.1687  -0.123    0.902
##
## Approximate significance of smooth terms:
##          edf Ref.df   Chi.sq p-value
## s(X2)  1.000  1.000    0.008   0.928
## s(X3)  1.000  1.000    0.032   0.859
## s(X8)  1.000  1.000    0.011   0.917
## s(X9)  1.000  1.000    0.009   0.925
## s(X11) 5.828  6.012 2681.302  <2e-16 ***
## s(X12) 1.000  1.000    0.000   0.998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.997   Deviance explained = 99.7%
## UBRE = -0.97742  Scale est. = 1         n = 1999
```

```
(aic4=AIC(gm4))
```

```
## [1] 7410.329
```

```
concurvity(gm4)
```

```
##                  para      s(X2)      s(X3)      s(X8)      s(X9)     s(X11)
## worst    4.234098e-20 0.04737622 0.12691622 0.03663153 0.03984509 0.14536174
## observed 4.234098e-20 0.01459780 0.06117484 0.02778071 0.02574637 0.01627891
## estimate 4.234098e-20 0.01411253 0.04126586 0.02673002 0.02594065 0.04142713
##              s(X12)
## worst    0.03245871
## observed 0.01704738
## estimate 0.01702878
```

Hemos conseguido eliminar los problemas de concurvidad. Sin embargo seguimos considerando variables en las que no rechazamos los contrastes individuales. Construyamos un modelo con la única variable significativa.

```
gm5<-gam(var_obj~s(X11),family = poisson, data = datos)
summary(gm5)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## var_obj ~ s(X11)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.1462     1.1695  -0.125      0.9
##
## Approximate significance of smooth terms:
##          edf Ref.df Chi.sq p-value
```

```
## s(X11) 5.828  6.012   2785  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.997   Deviance explained = 99.7%
## UBRE = -0.9824  Scale est. = 1         n = 1999
```

```
(aic5=AIC(gm5))
```

```
## [1] 7400.386
```

Veamos una tabla resumen de los resultados obtenidos.

```
modelos<-c("gm","gm2","gm3","gm4","gm5")
aic=c(aic1,aic2,aic3,aic4,aic5)

r_adj=c(summary(gm)$r.sq,summary(gm2)$r.sq,summary(gm3)$r.sq,summary(gm4)$r.sq,summary(gm5)$r.sq)
dev_r=c(gm$deviance,gm2$deviance,gm3$deviance,gm4$deviance,gm5$deviance)
dev_exp=c(summary(gm)$dev.expl,summary(gm2)$dev.expl,summary(gm3)$dev.expl,summary(gm4)$dev.expl,summary

(summary_table <- data.frame(
  Model = modelos,
  Adjusted_R2 = r_adj,
  AIC = aic,
  Deviance_Residual = dev_r,
  Deviance_Explained=dev_exp
))
```

```
##   Model Adjusted_R2      AIC Deviance_Residual Deviance_Explained
## 1    gm   0.9968845 7427.414          20.57077          0.9969545
## 2   gm2   0.9967409 7418.143          21.29475          0.9968473
## 3   gm3   0.9967372 7414.175          21.32754          0.9968424
## 4   gm4   0.9967319 7410.329          21.47924          0.9968200
## 5   gm5   0.9967316 7400.386          21.53563          0.9968116
```

Todos los modelos obtienen un R^2_{adj} similar, al igual que los valores para la devianza explicada. Es por ello que podemos considerar que el mejor modelo es el que considera únicamente la variable X_{11}, pues es el que mejor AIC posee. Hagamos el test ANOVA para confirmar que los modelos anteriores son realmente modelos anidados a éste último.

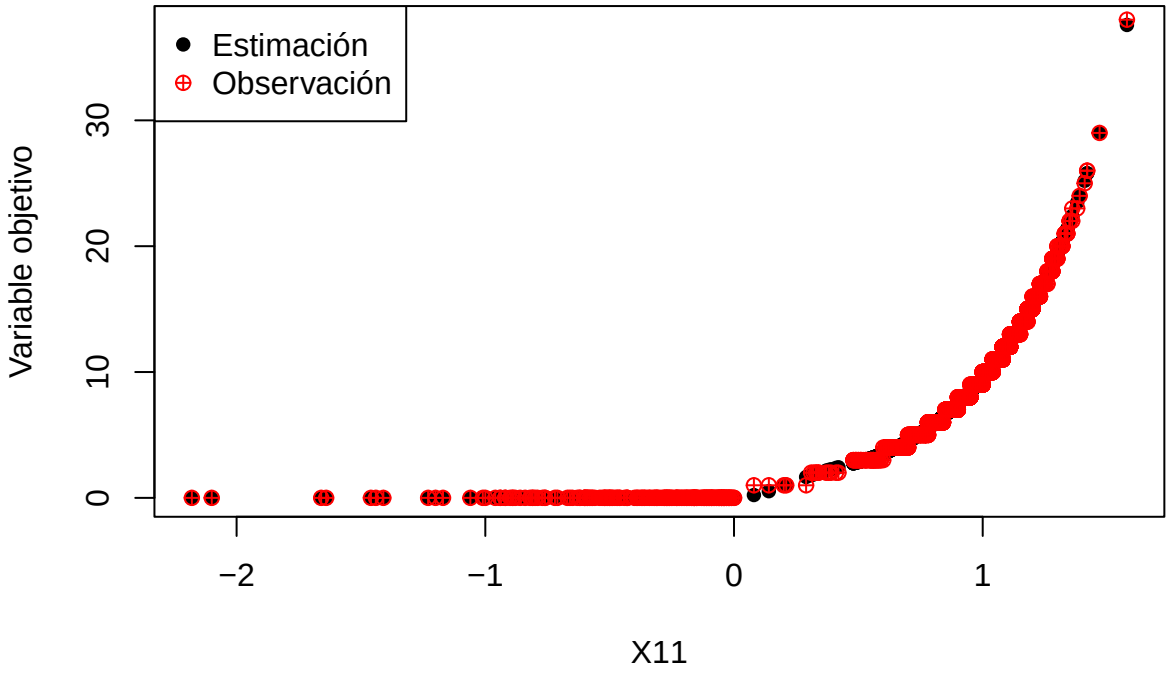
```
anova.gam(gm,gm2,gm3,gm4,gm5)
```

```
## Analysis of Deviance Table
##
## Model 1: var_obj ~ s(X1) + s(X2) + s(X3) + s(X4) + s(X5) + s(X6) + s(X7) +
##     s(X8) + s(X9) + s(X10) + s(X11) + s(X12) + factor(X13)
## Model 2: var_obj ~ s(X2) + s(X3) + s(X5) + s(X6) + s(X7) + s(X8) + s(X9) +
##     s(X10) + s(X11) + s(X12)
## Model 3: var_obj ~ s(X2) + s(X3) + s(X5) + s(X8) + s(X9) + s(X10) + s(X11) +
##     s(X12)
## Model 4: var_obj ~ s(X2) + s(X3) + s(X8) + s(X9) + s(X11) + s(X12)
## Model 5: var_obj ~ s(X11)
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1      1978     20.571
## 2      1983     21.295 -4.9985 -0.72398   0.9816
## 3      1985     21.328 -2.0003 -0.03279   0.9837
## 4      1987     21.479 -1.9993 -0.15170   0.9269
```

```
## 5      1992     21.536 -5.0004 -0.05639   1.0000
```

Ninguno de los términos eliminados en los modelos sucesivos resulta en un cambio significativo en la devianza. Esto sugiere que los términos eliminados no contribuyen significativamente al ajuste del modelo.El modelo más simple parece ser suficiente para explicar la variabilidad en los datos.

```
par(mfrow=c(1,1))
plot(X11,gm5$fitted.values,col="black",pch=16,ylab="Variable objetivo")
points(X11,var_obj,col="red",pch=10)
legend("topleft", legend = c("Estimación", "Observación"), col = c("black", "red"), pch = c(16,10))
```



Veamos ahora otros modelos estudiando únicamente la variable X_{11} pero cambiando el tipo de spline.

```
gm6<-gam(var_obj~s(X11,bs="cr"),family=poisson,data=datos)
summary(gm6)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## var_obj ~ s(X11, bs = "cr")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.5810     0.7352    0.79    0.429
##
## Approximate significance of smooth terms:
```

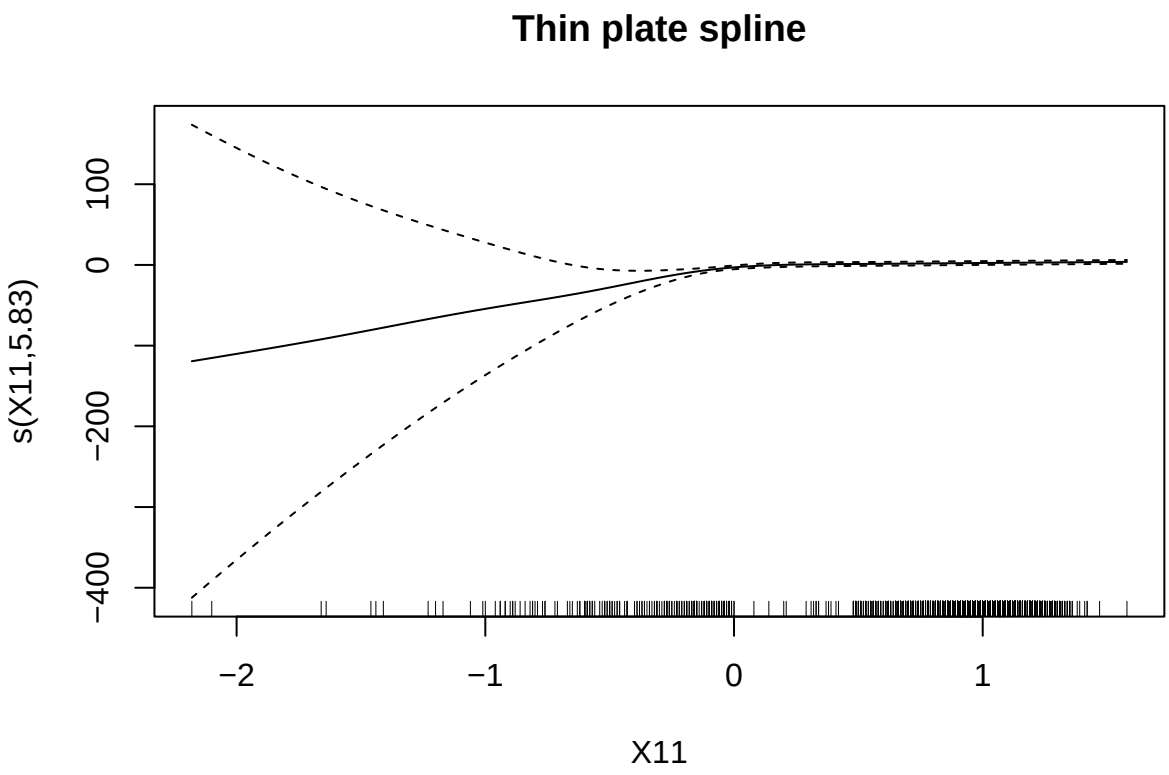
```
##          edf Ref.df Chi.sq p-value
## s(X11) 5.909  5.996   2779  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.997   Deviance explained = 99.7%
## UBRE = -0.98208  Scale est. = 1         n = 1999
```

```
(aic6=AIC(gm6))
```

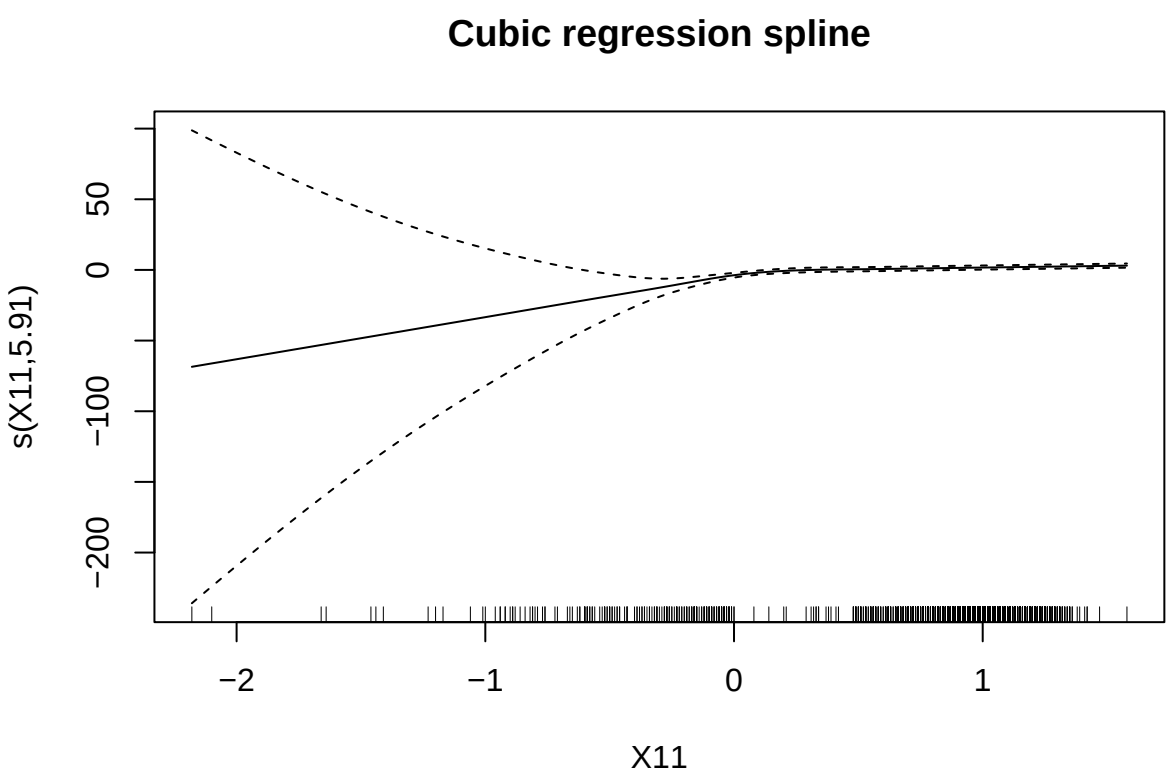
```
## [1] 7401.01
```

Obtenemos resultados muy similares.

```
plot(gm5)
title("Thin plate spline")
```

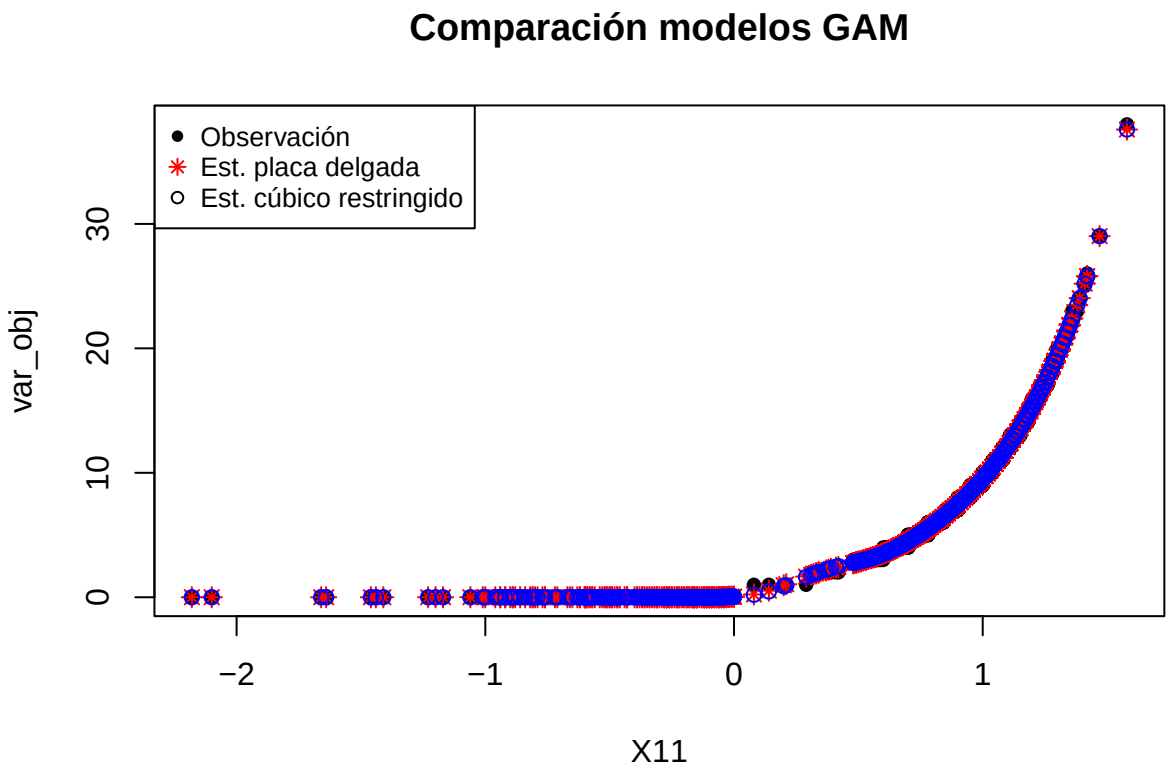



```
plot(gm6)
title("Cubic regression spline")
```



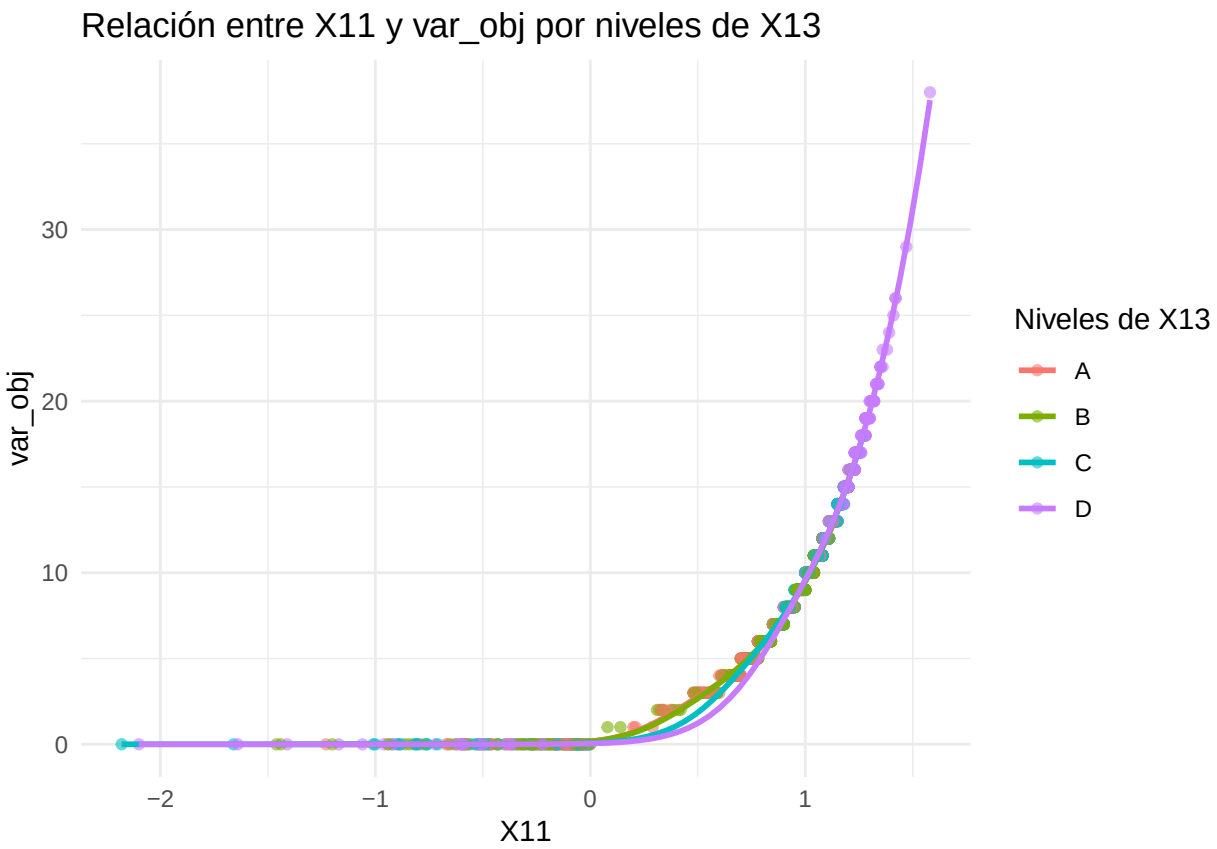
El spline de placa delgada es más flexible, como se puede observar en las fluctuaciones más notables al inicio del rango de X_{11}. En el spline cúbico restringido muestra un efecto más contenido, con un suavizado más rígido y lineal hacia el final del rango de X_{11}. Los intervalos de confianza son más amplios en el spline de placa delgada, especialmente en los extremos del rango de X_{11}. Esto sugiere que el modelo permite mayor flexibilidad, pero a costa de mayor incertidumbre en las estimaciones. No obstante, debido a los resultados tan similares, nos quedaremos con `gm5`.

```
plot(X11,var_obj,col="black",pch=16)
points(X11,gm5$fitted.values,col="red",pch=8)
points(X11,gm6$fitted.values,col="blue",pch=1)
legend("topleft", legend = c("Observación","Est. placa delgada","Est. cúbico restringido"), col = c("bla
title("Comparación modelos GAM")
```



En el modelo GAM con todas las variables vimos que la variable X_{13} no era significativa para el modelo. Sin embargo, como teníamos problemas de concurvidad, vamos a comprobar si verdaderamente la variable categórica no es significativa o los resultados anteriores se vieron alterados.

```
ggplot(data=datos,aes(x=X11,y=var_obj,color=X13))+
  geom_point(alpha=0.6)+
  geom_smooth(method = "gam", formula = y ~ s(x),method.args = list(family = poisson), se = FALSE) +
  labs(
    title = "Relación entre X11 y var_obj por niveles de X13",
    x = "X11",
    y = "var_obj",
    color = "Niveles de X13"
  ) +
  theme_minimal()
```




```
  theme(legend.position = "left")
```

```
## List of 1
##  $ legend.position: chr "left"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

No hay señales de una interacción significativa entre X_{11} y X_{13} pues no hay una separación clara entre las curvas de cada categoría. No obstante, hay una cierta separación cuando la variable X_{11} toma valores entre 0 y 1. En dicha franja era donde observábamos un mayor error en el modelo. Procedamos a construir el modelo para contrastar este resultado.

```
gm7<-gam(var_obj~s(X11,by=X13),family = poisson,data=datos)
summary(gm7)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## var_obj ~ s(X11, by = X13)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.3331     0.1215   10.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Approximate significance of smooth terms:
##               edf Ref.df Chi.sq p-value
## s(X11):X13A 4.448  4.996  273.7  <2e-16 ***
## s(X11):X13B 5.018  5.511  615.8  <2e-16 ***
## s(X11):X13C 3.705  4.197  360.9  <2e-16 ***
## s(X11):X13D 3.559  3.980  516.6  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.997   Deviance explained = 99.6%
## UBRE = -0.96909  Scale est. = 1         n = 1999
```

```
(aic7=AIC(gm7))
```

```
## [1] 7426.984
```

A pesar de la significación de los términos suaves individuales, el AIC aumenta y el R^2_{adj} se mantiene.

```
nuevas_filas<-data.frame(Model=c("gm6","gm7"),Adjusted_R2=c(summary(gm6)$r.sq,summary(gm7)$r.sq),
                         AIC=c(aic6,aic7),Deviance_Residual=c(gm5$deviance,gm6$deviance),
                         Deviance_Explained=c(summary(gm6)$dev.expl,summary(gm7)$dev.expl))

(summary_table<-rbind(summary_table,nuevas_filas))
```

```
##   Model Adjusted_R2      AIC Deviance_Residual Deviance_Explained
## 1    gm   0.9968845 7427.414          20.57077          0.9969545
## 2   gm2   0.9967409 7418.143          21.29475          0.9968473
## 3   gm3   0.9967372 7414.175          21.32754          0.9968424
## 4   gm4   0.9967319 7410.329          21.47924          0.9968200
## 5   gm5   0.9967316 7400.386          21.53563          0.9968116
## 6   gm6   0.9967309 7401.010          21.53563          0.9967429
## 7   gm7   0.9966765 7426.984          21.99967          0.9961016
```

Concluimos por tanto que el mejor modelo aditivo generalizado es aquel que considera únicamente la variable X_{11}

Vamos a estudiar si existe sobreajuste. Para ello dividamos los datos en conjunto test y conjunto de entrenamiento.

```
set.seed(456)
indices<-sample(1:nrow(datos),round(0.7*nrow(datos),0))
train<-datos[indices,]
test<-datos[-indices,]
```

Construyamos el modelo con los datos de entrenamiento y verifiquemos su bondad con los datos test.

```
gm5_test<-gam(var_obj~s(X11),family = poisson, data = train)
summary(gm5_test)$r.sq
```

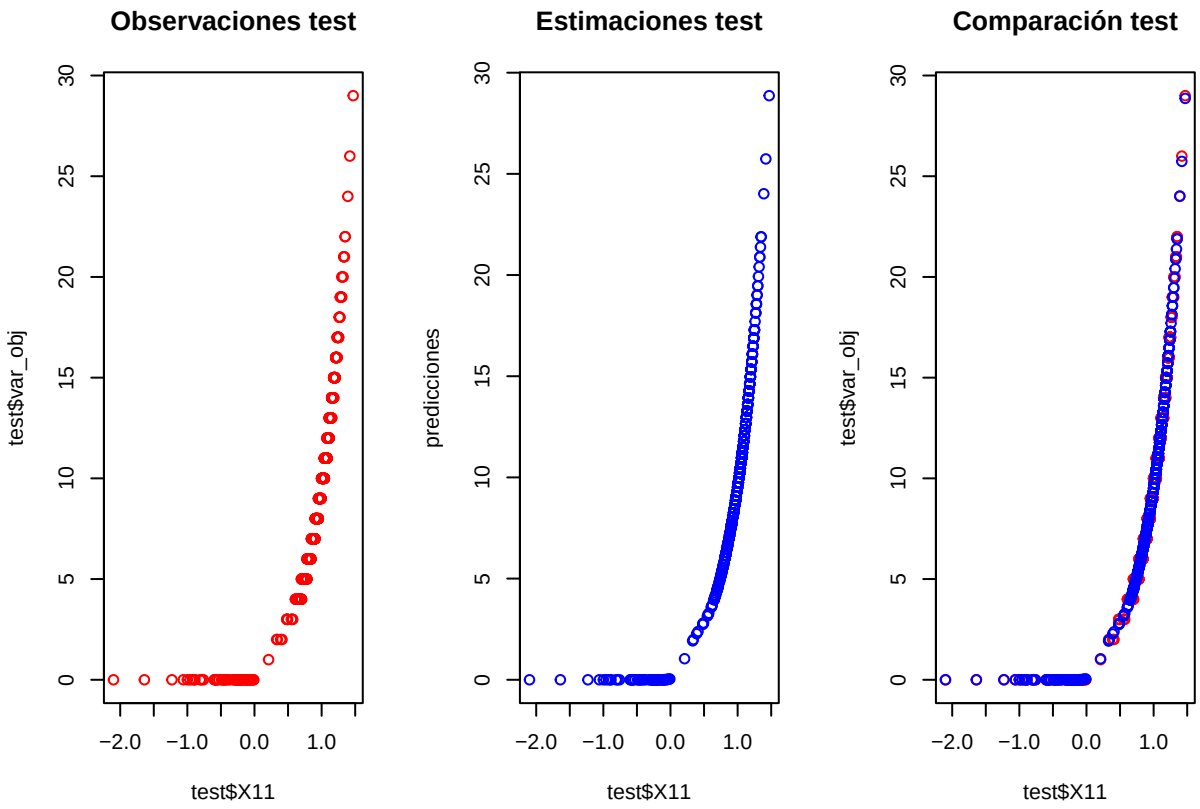
```
## [1] 0.9965993
```

```
predicciones<-predict.gam(gm5_test,newdata=test[,-1],type="response")
(err1<-mean( (test$var_obj -predicciones)^2))
```

```
## [1] 0.07633689
```

```
par(mfrow=c(1,3))
plot(test$X11,test$var_obj,col="red",main="Observaciones test")
plot(test$X11,predicciones,col="blue",main="Estimaciones test")
```

```
plot(test$X11,test$var_obj,col="red",main="Comparación test")
points(test$X11,predicciones,col="blue",main="Estimaciones test")
```

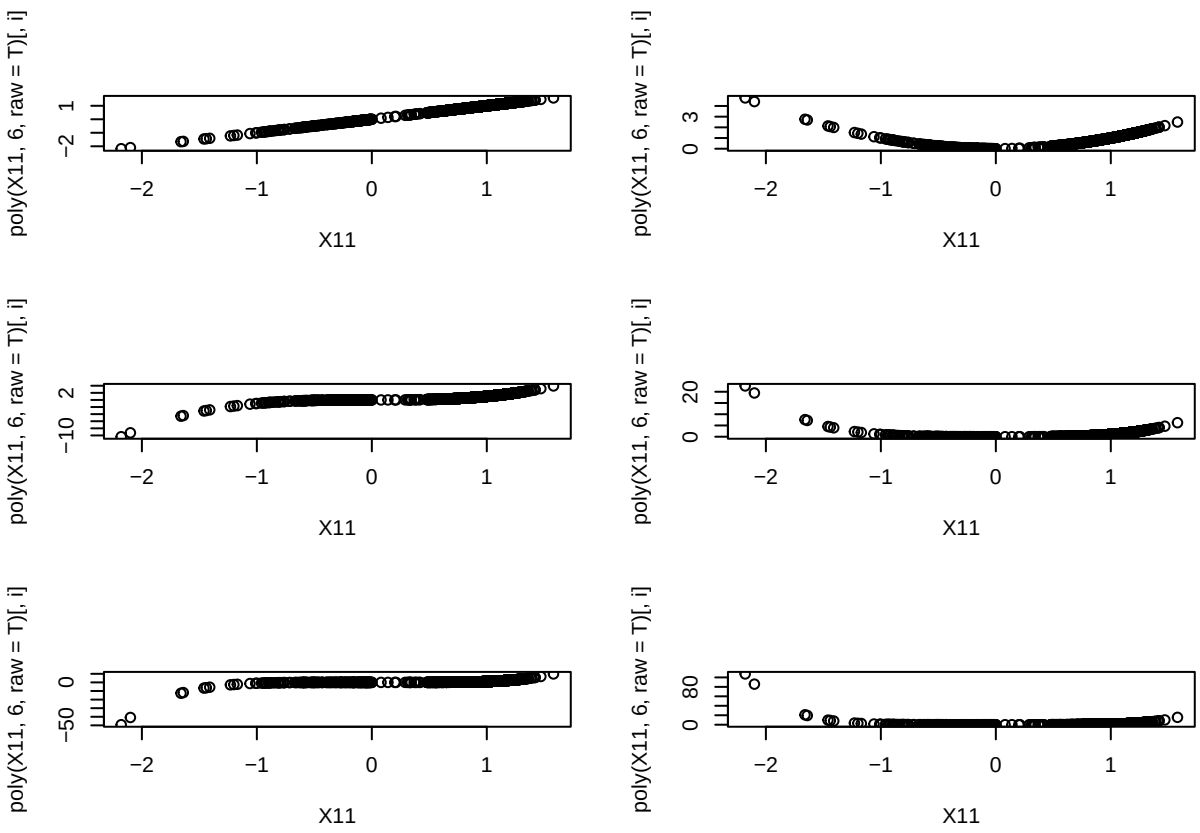


Observamos un buen ajuste en el conjunto test, por lo que rechazamos el sobreajuste.

Regresión polinómica

Vamos a estudiar distintos modelos de regresión polinómica considerando la variable X_{11}.

```
par(mfrow=c(3,2))
for(i in 1:6) {plot(X11,poly(X11,6,raw=T)[,i])}
```




```
regpoly<-lm(var_obj~poly(X11,6,raw=T),data=datos)
summary(regpoly)
```

```
##
## Call:
## lm(formula = var_obj ~ poly(X11, 6, raw = T), data = datos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68532 -0.24649  0.01932  0.22577  0.66345
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.32825    0.02522  13.018  < 2e-16 ***
## poly(X11, 6, raw = T)1  2.46142    0.04762  51.688  < 2e-16 ***
## poly(X11, 6, raw = T)2  3.56930    0.10174  35.083  < 2e-16 ***
## poly(X11, 6, raw = T)3  1.12678    0.07641  14.746  < 2e-16 ***
## poly(X11, 6, raw = T)4  0.59212    0.08824   6.711 2.52e-11 ***
## poly(X11, 6, raw = T)5  1.06539    0.03074  34.658  < 2e-16 ***
## poly(X11, 6, raw = T)6  0.36466    0.02171  16.800  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2864 on 1992 degrees of freedom
## Multiple R-squared:  0.9966, Adjusted R-squared:  0.9966
## F-statistic: 9.783e+04 on 6 and 1992 DF,  p-value: < 2.2e-16
```

```
AIC(regpoly)
```

```
## [1] 683.6137
```

Hemos rechazado todos los contrastes individuales y hemos obtenido un gran ajuste. Sigamos aumentando el grado para ver si obtenemos mejores resultados. Más adelante, comprobaremos si existe sobreajuste.

```
regpoly2<-lm(var_obj~poly(X11,8,raw=T),data=datos)
summary(regpoly2)
```

```
##
## Call:
## lm(formula = var_obj ~ poly(X11, 8, raw = T), data = datos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6860 -0.2423  0.0231  0.2163  0.6692
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.277955   0.028163   9.869  < 2e-16 ***
## poly(X11, 8, raw = T)1  2.289573   0.076981  29.742  < 2e-16 ***
## poly(X11, 8, raw = T)2  4.039196   0.168242  24.008  < 2e-16 ***
## poly(X11, 8, raw = T)3  1.485938   0.202233   7.348 2.93e-13 ***
## poly(X11, 8, raw = T)4 -0.253835   0.259871  -0.977   0.3288
## poly(X11, 8, raw = T)5  0.932037   0.165077   5.646 1.88e-08 ***
## poly(X11, 8, raw = T)6  0.791915   0.137634   5.754 1.01e-08 ***
## poly(X11, 8, raw = T)7  0.006936   0.042952   0.161   0.8717
## poly(X11, 8, raw = T)8 -0.060741   0.025517  -2.380   0.0174 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2854 on 1990 degrees of freedom
## Multiple R-squared:  0.9966, Adjusted R-squared:  0.9966
## F-statistic: 7.392e+04 on 8 and 1990 DF,  p-value: < 2.2e-16
```

```
AIC(regpoly2)
```

```
## [1] 670.896
```

Ahora no se rechazan todos los contrastes individuales y hemos conseguido disminuir el AIC. Construiremos un modelo con un grado más y realizaremos el test de ANOVA para ver con qué modelo nos quedamos.

```
regpoly3<-lm(var_obj~poly(X11,9,raw=T),data=datos)
summary(regpoly3)
```

```
##
## Call:
## lm(formula = var_obj ~ poly(X11, 9, raw = T), data = datos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68400 -0.24342  0.02205  0.21733  0.66954
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.282625   0.030743   9.193  < 2e-16 ***
```

```
## poly(X11, 9, raw = T)1  2.315630   0.103202  22.438  < 2e-16 ***
## poly(X11, 9, raw = T)2  3.995777   0.203542  19.631  < 2e-16 ***
## poly(X11, 9, raw = T)3  1.383112   0.338311   4.088 4.52e-05 ***
## poly(X11, 9, raw = T)4 -0.151684   0.374350  -0.405  0.68538
## poly(X11, 9, raw = T)5  1.051384   0.355427   2.958  0.00313 **
## poly(X11, 9, raw = T)6  0.716380   0.242145   2.958  0.00313 **
## poly(X11, 9, raw = T)7 -0.049630   0.155241  -0.320  0.74923
## poly(X11, 9, raw = T)8 -0.043902   0.051219  -0.857  0.39147
## poly(X11, 9, raw = T)9  0.009883   0.026065   0.379  0.70459
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2855 on 1989 degrees of freedom
## Multiple R-squared:  0.9966, Adjusted R-squared:  0.9966
## F-statistic: 6.567e+04 on 9 and 1989 DF,  p-value: < 2.2e-16
```

```
AIC(regpoly3)
```

```
## [1] 672.7515
```

```
anova(regpoly,regpoly2,regpoly3)
```

```
## Analysis of Variance Table
##
## Model 1: var_obj ~ poly(X11, 6, raw = T)
## Model 2: var_obj ~ poly(X11, 8, raw = T)
## Model 3: var_obj ~ poly(X11, 9, raw = T)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1992 163.45
## 2   1990 162.09  2   1.36122 8.3525 0.0002442 ***
## 3   1989 162.08  1   0.01172 0.1438 0.7045940
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Al aumentar a una regresión polinómica de grado 9 hemos aumentado el AIC. Por ello, y apoyándonos en los resultados del test ANOVA, tomaremos la regresión de grado 8.

```
(summary(regpoly2)$adj.r.squared)
```

```
## [1] 0.9966325
```

Construyamos el modelo con los datos de entrenamiento y verifiquemos su bondad con los datos test.

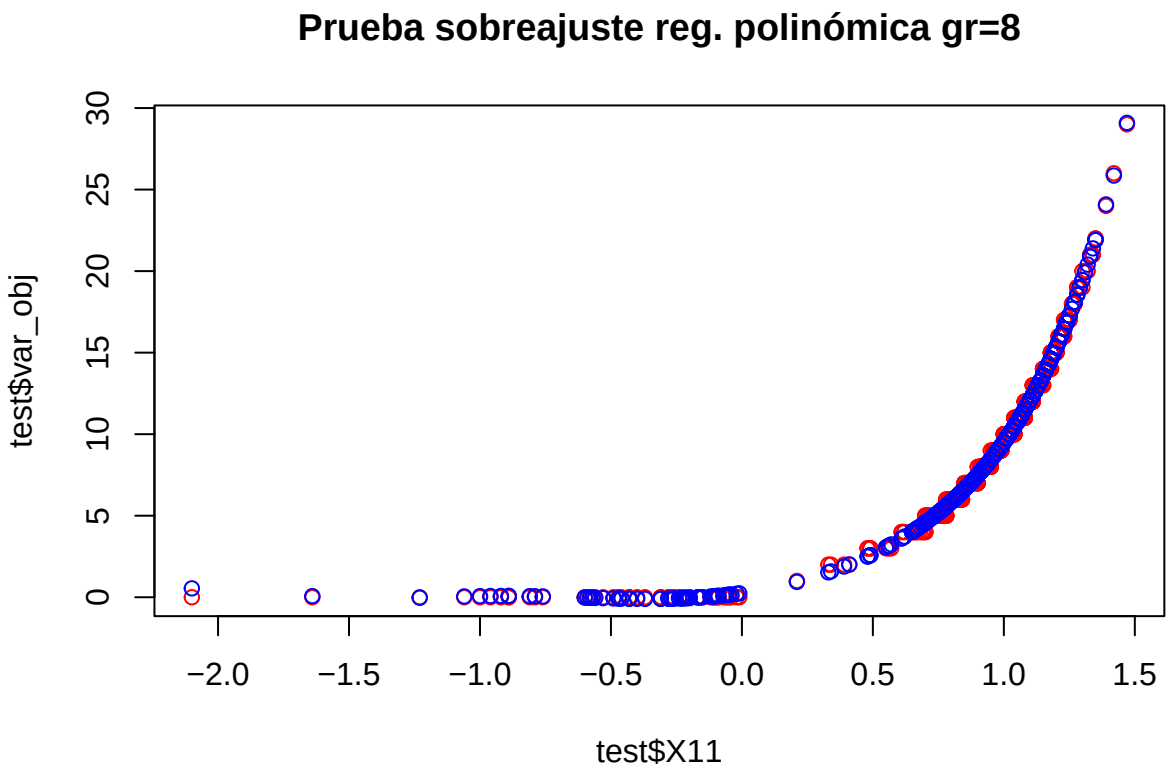
```
regpoly_entreno<-lm(var_obj~poly(X11,8,raw=T),data=train)
summary(regpoly_entreno)$adj.r.squared
```

```
## [1] 0.9965492
```

```
predicciones<-predict(regpoly_entreno,test[,-1])
(err2<-mean( (test$var_obj -predicciones)^2))
```

```
## [1] 0.08016036
```

```
par(mfrow=c(1,1))
plot(test$X11,test$var_obj,col="red",)
points(test$X11,predicciones,col="blue")
title("Prueba sobreajuste reg. polinómica gr=8")
```



Observamos que el modelo realiza un ajuste muy bueno considerando el conjunto de entrenamiento y prediciendo el conjunto test. Podemos descartar el sobreajuste.

Regresión con splines

```
library(splines)
```

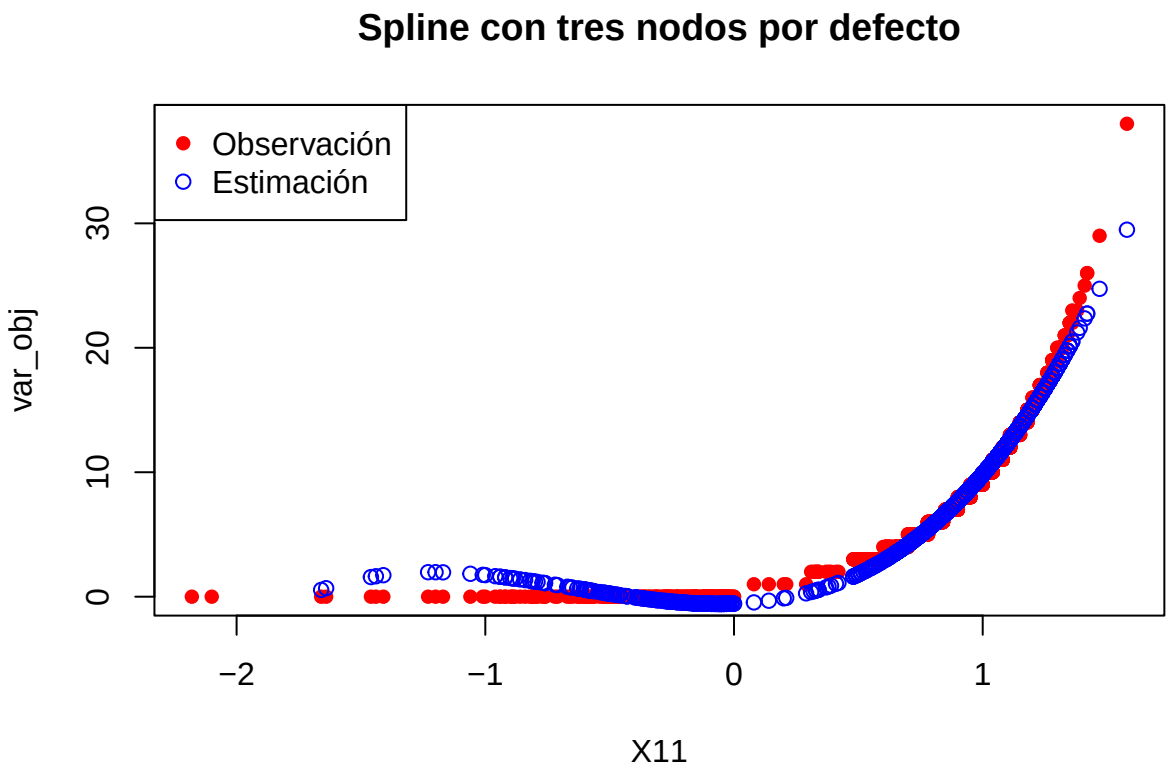
Este paquete proporciona funciones para trabajar con splines utilizando la base B-spline y la base spline cúbico natural. Por defecto la función tomará tres nodos.

```
reg_spline1<-lm(var_obj~bs(X11),data=datos)
summary(reg_spline1)
```

```
##
## Call:
## lm(formula = var_obj ~ bs(X11), data = datos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9660 -0.3266 -0.0603  0.2775  8.5115
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -6.3369     0.3518  -18.01   <2e-16 ***
## bs(X11)1     25.4443     0.6225   40.87   <2e-16 ***
## bs(X11)2    -21.2865     0.3109  -68.48   <2e-16 ***
## bs(X11)3     35.8255     0.3869   92.58   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5913 on 1995 degrees of freedom
## Multiple R-squared:  0.9856, Adjusted R-squared:  0.9855
## F-statistic: 4.541e+04 on 3 and 1995 DF,  p-value: < 2.2e-16
```

```
plot(X11,var_obj,col="red",pch=16)
points(X11,reg_spline1$fitted.values,col="blue",pch=1)
legend("topleft",legend = c("Observación","Estimación"),col=c("red","blue"),pch=c(16,1))
title("Spline con tres nodos por defecto")
```



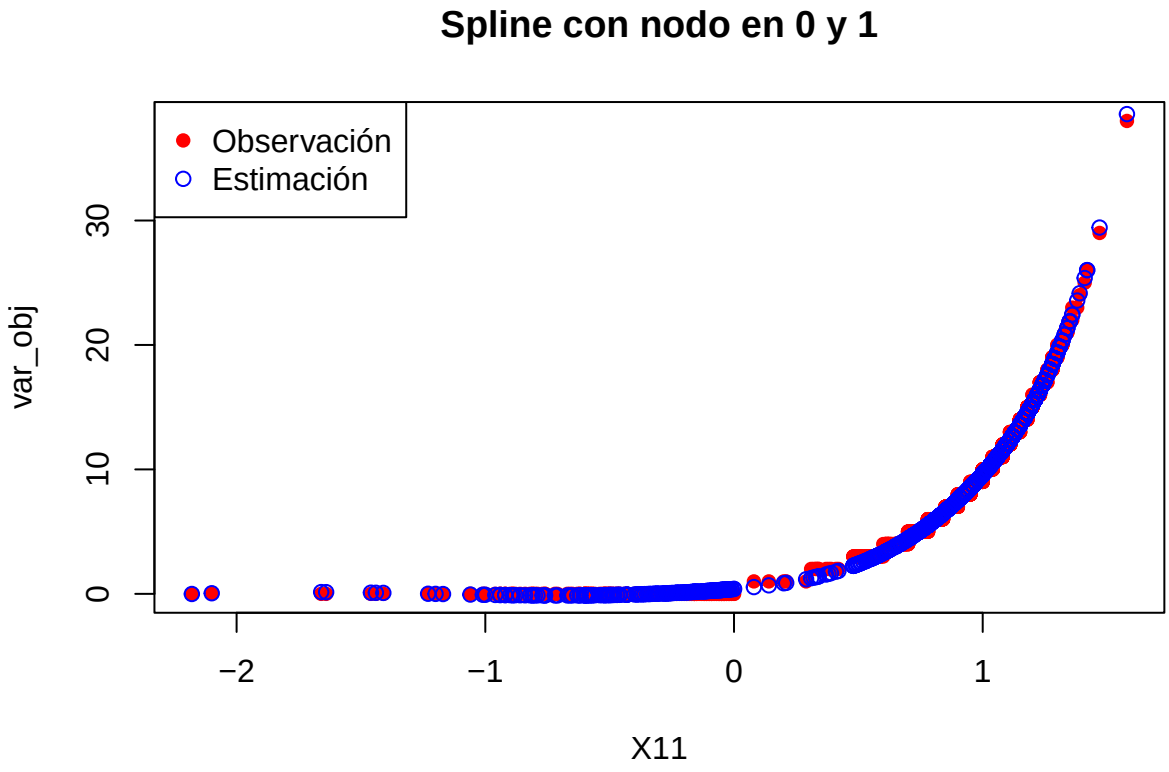
Hasta ahora, gráficamente se puede observar que es el peor modelo que hemos obtenido. Sin embargo se muestra un valor alto de R^2_{adj}. Vamos a indicar los puntos donde vemos un mayor cambio en la variable objetivo según los valores de $X_1 1$.

```
reg_spline2<-lm(var_obj~bs(X11,knots=c(0,1)),data=datos)
summary(reg_spline2)
```

```
##
## Call:
## lm(formula = var_obj ~ bs(X11, knots = c(0, 1)), data = datos)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59957 -0.25499  0.01421  0.19580  0.75153
##
## Coefficients:
```

```
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -0.01202    0.21021  -0.057   0.9544
## bs(X11, knots = c(0, 1))1   0.61682    0.31942   1.931   0.0536 .
## bs(X11, knots = c(0, 1))2 -1.50880    0.20201  -7.469  1.2e-13 ***
## bs(X11, knots = c(0, 1))3  2.77707    0.22110  12.560  < 2e-16 ***
## bs(X11, knots = c(0, 1))4 20.44460    0.20859  98.014  < 2e-16 ***
## bs(X11, knots = c(0, 1))5 38.56377    0.28904 133.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3023 on 1993 degrees of freedom
## Multiple R-squared:  0.9962, Adjusted R-squared:  0.9962
## F-statistic: 1.054e+05 on 5 and 1993 DF,  p-value: < 2.2e-16
```

```
plot(X11,var_obj,col="red",pch=16)
points(X11,reg_spline2$fitted.values,col="blue",pch=1)
legend("topleft",legend = c("Observación","Estimación"),col=c("red","blue"),pch=c(16,1))
title("Spline con nodo en 0 y 1")
```



Efectivamente, vemos la importancia de tomar los nodos correctos.

```
(aic_sp1=AIC(reg_spline1))
```

```
## [1] 3578.041
```

```
(aic_sp2=AIC(reg_spline2))
```

```
## [1] 897.5015
```

Hemos conseguido disminuir considerablemente el AIC.

Veamos si existe sobreajuste.

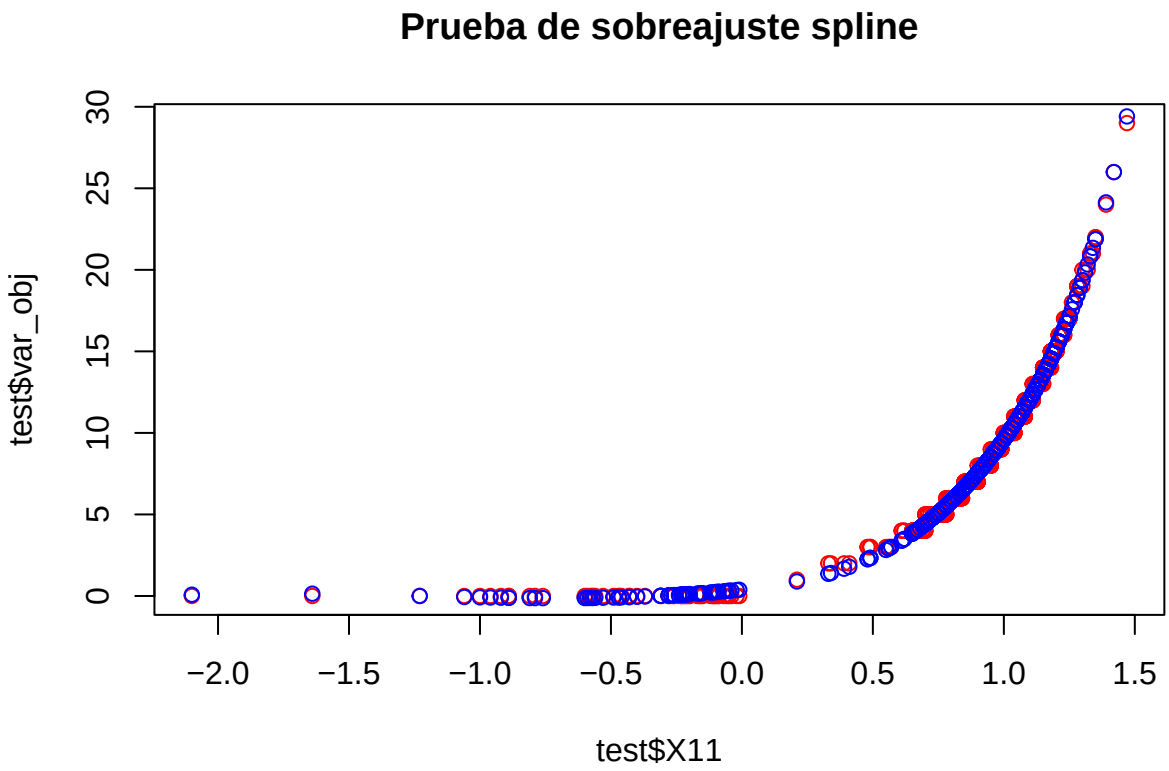
```
regspline_entreno<-lm(var_obj~bs(X11,knots=c(0,1)),data=train)
summary(regspline_entreno)$adj.r.squared
```

```
## [1] 0.99608
```

```
predicciones<-predict(regspline_entreno,test[,-1])
(err3<-mean( (test$var_obj -predicciones)^2))
```

```
## [1] 0.08677656
```

```
par(mfrow=c(1,1))
plot(test$X11,test$var_obj,col="red",)
points(test$X11,predicciones,col="blue")
title("Prueba de sobreajuste spline")
```



Observamos que el modelo realiza un ajuste muy bueno considerando el conjunto de entrenamiento y prediciendo el conjunto test. Podemos descartar el sobreajuste.