Modelos para Datos de Conteo

Carlos Pérez Manzano

Tabla de contenidos

Modelos aplicados en el análisis

Recogemos los modelos estudiados más relevantes del estudio:

1. Modelos lineales generalizados (Poisson y Binomial Negativa)

- model.poiss_full: modelo Poisson incluyendo todas las variables explicativas.
- model.poiss_best: modelo Poisson con variables explicativas X1, X2, X4, X8, X11.
- model.bn_full: modelo de familia Binomial Negativa con todas las variables predictoras.
- model.bn_best: modelo de familia Binomial Negativa incluyendo X1, X2, X4, X11.

2. Modelos aditivos generalizados

- model.gam_full: modelo con todas las variables regresoras.
- model.gam_X11tp: modelo considerando únicamente la variable X11 como predictora y spline de regresión de placa delgada.
- model.gam_X11ts: modelo considerando únicamente la variable X11 como predictora y spline de regresión de placa delgada penalizada.
- model.gam_X11cr: modelo considerando únicamente la variable X11 como predictora y spline cúbico.

3. Modelos mediante regresión polinomial

- model.poly_8: regresión polinomial con X11 como única variable regresora de grado 8.

4. Modelos de regresión a través de splines

- model.spline: regresión a través de splines tomando como nodos el 0 y 1.

Resumen ejecutivo

Introducción

El proyecto consiste en la utilización de distintas técnicas para el ajuste de una variable objetivo que es de tipo conteo. Estas son variables discretas no negativas y representan el número de veces que ocurre un evento en un determinado período de tiempo o espacio. Este tipo de dato requiere de modelos específicos que desarrollaremos. Para la ejemplificación de estos modelos se disponde del conjunto de datos **"data.xlsx"**, del que desconocemos la naturaleza ni el significado de las variables que lo componen.

Para la comparación entre los distintos modelos, nos basaremos principalmente en el criterio AIC, que tiene en cuenta tanto el ajuste como la complejidad del modelo, la desviación residual siempre que sea posible y por último el test Anova.

Análisis exploratorio

En el conjunto de datos contamos con un total de 1999 observaciones de 14 variables. La variable objetivo es nombrada como `var_obj` y las variables regresoras $X1, X2, ... X13$. Todas son continuas, excepto $X13$ que es categórica con 4 niveles.

Como hemos dicho anteriormente, la variable objetivo es de tipo conteo, y se puede ver en este gráfico como se ajustan a las distribuciones teóricas de las distribuciones Poisson y Binomial Negativa.

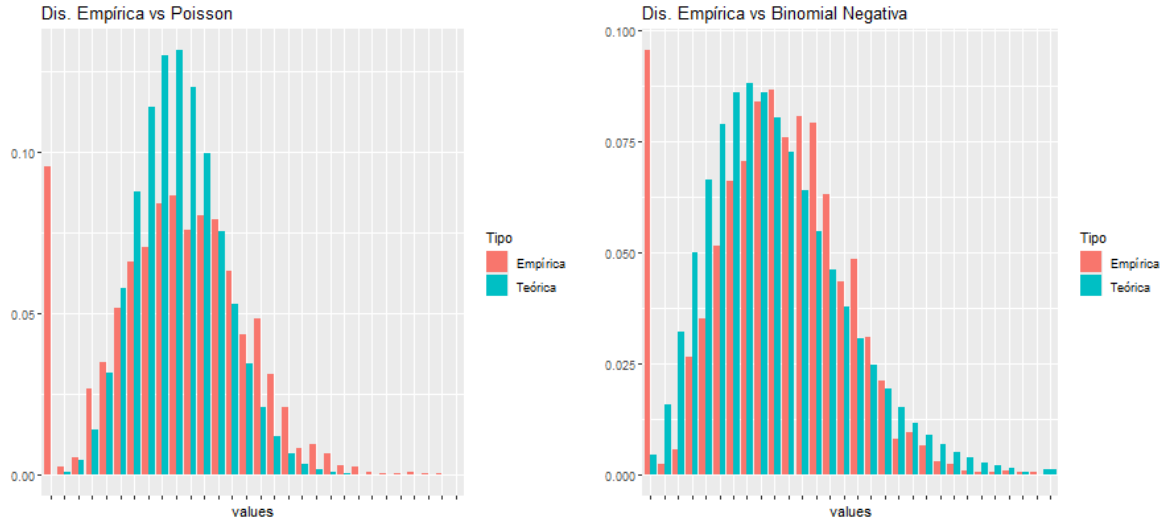


Figura 1: Distribución empírica vs teórica

En cuanto a las variables continuas, se realiza la transformación logarítmica a las variables $X1$ y $X2$ debido a la alta presencia de outliers, renombrándolas como $X1\_trans$ y $X2\_trans$ respectivamente.

Modelos lineales generalizados

Respecto al modelo Poisson, se parte del modelo con todas las variables **model.poiss_full**, y eliminamos variables mediante el criterio de menor AIC, llegando al modelo **model.poiss_best**, que incluye las variables $X1\_trans$, $X2\_trans$, $X4$, $X8$ y $X11$. Para este modelo final, los coeficientes son muy cercanos a 0, exceptuando la variable $X11$ que tiene un valor aproximado de 3.19. Por tanto el modelo estima que por cada unidad adicional de la variable $X11$, el valor de `var_obj` se incrementa en $e^{3.19} \approx 24.3$.

En cuanto al modelo mediante la Binomial Negativa, de igual manera se realiza **model.bn_full** con todas las variables, y eliminando variables llegamos a **model.bn_best**. Llegamos a la conlusión de que no hay diferencias significativas con el modelo Poisson, luego escogeremos este último como representante.

	AIC	Devianza
model.poiss_full	7495.794	118.6009
model.bn_full	7498.059	120.7888

Tabla 1: Tabla de AIC y Devianza añadiendo GAM

Modelos Adivitivos Generalizados

Partimos de **model.gam_full**. el modelo aditivo generalizado con todas las variables y el tipo de spline por defecto, que es spline de regresión de placa delgada. El objetivo es realizar a través de este la selección de variables y posteriormente seleccionar el tipo de spline más adecuado. Con este primer modelo ya podemos observar una disminución considerable en el AIC, con valor de 7420.893 y sobre todo en la devianza, 20.07102. Además, el valor del R^2_{adj} es de 0.997 aproximadamente. Esto nos hace sospechar un posible problema de concurvidad, lo que conduciría a modelos poco eficientes.

Mediante un proceso de eliminación de las variables con mayor concurvidad, llegando al modelo incluyendo las variables $X2\_trans$, $X3$, $X8$, $X9$, $X11$ y $X12$ con este problema ya solucionado. Sin embargo, para todas las variables del modelo se rechazan rotundamente los test individuales de significatividad, excepto el de la variable $X11$. Por ello, se realiza el modelo **model.gam_X11tp**, en el que solo se incluye la variable $X11$ como predictora. Realizando el test anova para compararla con **model.gam_full**, aceptamos la igualdad de modelos. Mostramos en la siguiente tabla los resultados de bondad del modelo.

	AIC	Devianza
model.poiss_full	7495.794	118.6009
model.bn_full	7498.059	120.7888
model.gam_full	7420.893	20.07102
model.gam_X11tp	7400.386	21.53563

Tabla 2: Tabla de AIC y Devianza añadiendo GAM

Seleccionaremos por tanto **model.gam_X11tp** como mejor modelo aditivo generalizado, debido a que tiene un AIC bastante más reducido que **model.gam_full** y la variación en la devianza no es para nada elevada.

Posteriormente, debido a que desconocemos la naturaleza de los datos, comprobamos si los resultados son mejores variando el tipo de spline empleado en el modelo. Sin embargo, las diferencias son mínimas. Se puede consultar el anexo para más detalle.

Modelos de Regresión Polinómica

Llegados a este punto, es claro que es suficiente con considerar la variable $X11$ para la predicción. Realizamos un proceso de elección del mejor grado del polinomio tomando como criterio el test anova, es decir, se toma el grado para el cual no se mejora el modelo tomando como hiperparámetro el grado posterior. Esocgemos los posibles valores entre 1 y 10 y no superiores para evitar sobreajuste. De esta manera, obtenemos **model.poly_8**, tomando grado 8. Se muestra la tabla con los modelos actuales.

	AIC	Devianza
model.poiss_full	7495.794	118.6009
model.bn_full	7498.059	120.7888
model.gam_full	7420.893	20.07102
model.gam_X11tp	7400.386	21.53563
model.gam_X11ts	7400.385	21.53674
model.gam_X11cr	7401.010	21.99967
model.poly_8	670.896	NA

Tabla 3: Tabla de AIC y Devianza añadiendo Regresión Polinómica

Obtenemos un AIC realmente bueno. Veamos un gráfico del ajuste del modelo.

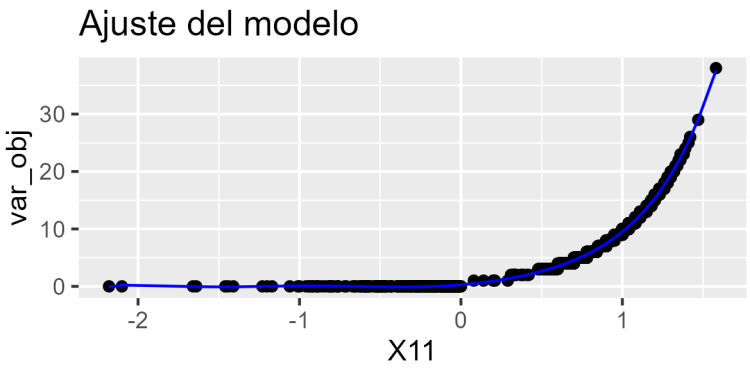


Figura 2: Ajuste del modelo polinómico

Modelos de Regresión mediante Splines

Se realiza un modelo con nodos en el 0 y 1, para captar los cambios de comportamiento entre $X11$ y `var_obj`.

	AIC	Devianza
model.poiss_full	7495.794	118.6009
model.bn_full	7498.059	120.7888
model.gam_full	7420.893	20.07102
model.gam_tp	7400.386	21.53563
model.gam_ts	7400.385	21.53674
model.gam_cr	7401.010	21.99967
model.poly_8	670.896	NA
model.spline	897.5015	NA

Tabla 4: Tabla de AIC y Devianza añadiendo Regresión Polinómica

Obtenemos buenos resultados, sin emabrgo son aparentemente mejores para el regresor polinomial.

Selección final y evaluación

Teniendo en cuenta la exposición de los modelos anteriores el modelo final escogido será el polinómico de grado 8. Por último incluimos una prueba para refutar la existencia de sobreajuste. Dividimos el conjunto de datos en entrenamiento y test, ajustamos el modelo polinomial de grado 8 para el conjunto de entrenamiento y evaluamos en el conjunto test. Mostramos la gráfica del ajuste al conjunto de prueba.

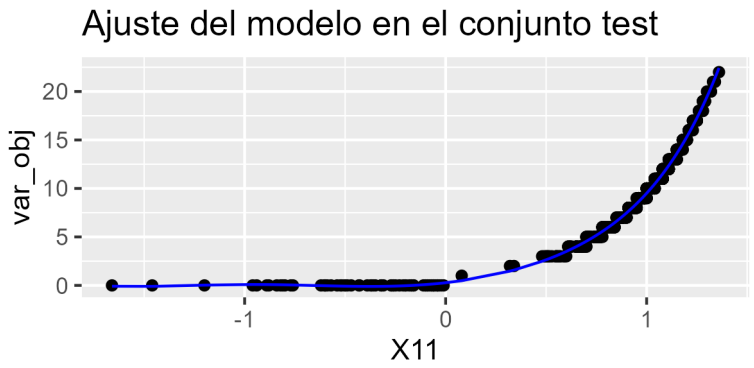


Figura 3: Ajuste en el conjunto test

El ajuste es realmente bueno. Concluimos que la variable $X11$ puede modelar casi por completo a la variable `var_obj`, hay una dependencia clara entre ambas. A continuación se muestra el anexo, donde se puede ver con detalle el desarrollo del proyecto y algunos análisis complementarios.

Anexo

Ánalisis exploratorio

Cargamos en primer lugar los paquetes necesarios.

```
library(openxlsx)
library(usdm)
library(MASS)
library(dplyr)
library(AER)
library(ggplot2)
library(gridExtra)
library(mgcv)
library(gamair)
library(splines)
```

Cargamos el dataset.

```
datos = read.xlsx("data.xlsx")
head(datos)
```

```
  var_obj   X1     X2    X3    X4    X5    X6    X7   X8   X9    X10  X11
  X12
1       7 0.67  28.31  2.47 15.08 15.83 12.18 12.36 0.54 1.92  91.39 0.87
21.95
2       8 0.83  80.95  0.99 15.69 15.85 12.30 12.41 0.69 0.54  82.50 0.94
19.54
3       7 0.53  27.41  7.37 15.23 17.67 19.00 17.88 0.33 0.48  97.31 0.90
24.36
4       9 0.98  58.40  0.28 16.39 15.66 19.03 18.29 0.61 1.24  66.04 0.96
36.54
5       3 0.53   6.38 74.70 12.61 11.77 18.07 15.61 0.05 0.68  43.11 0.53
19.46
6      16 1.71 121.18  3.61 21.03 21.39 18.69 17.62 0.31 1.97 164.62 1.22
17.86
  X13
1   B
2   C
3   B
4   C
5   B
6   D
```

```
sapply(datos, function(x) sum(is.na(x)))
```

```
var_obj      X1      X2      X3      X4      X5      X6      X7      X8
X9
      0       0       0       0       0       0       0       0       0
      0
    X10     X11     X12     X13
      0       0       0       0
```

No tenemos valores nulos en ninguna de las variables.

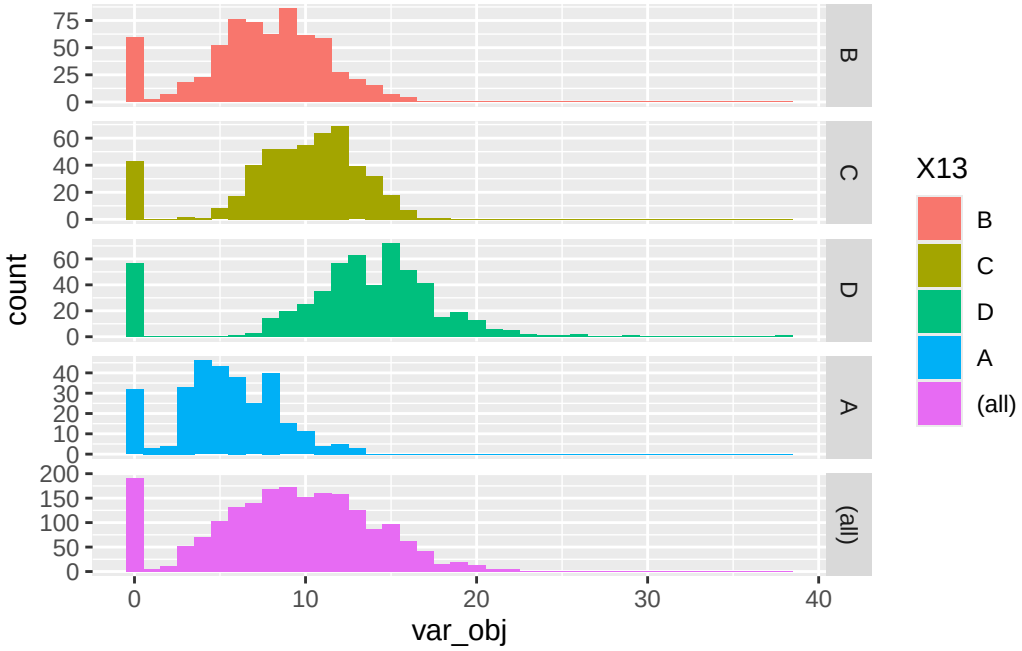
```
summary(datos)
```

```
   var_obj             X1               X2                  X3
 Min.   : 0.000   Min.   :0.0400   Min.   :    0.03   Min.   :  0.000
 1st Qu.: 6.000   1st Qu.:0.5000   1st Qu.:    5.96   1st Qu.:  1.150
 Median : 9.000   Median :0.8300   Median :   24.34   Median :  2.020
 Mean   : 9.117   Mean   :0.9492   Mean   :  173.53   Mean   :  4.621
 3rd Qu.:12.000   3rd Qu.:1.2500   3rd Qu.:   82.09   3rd Qu.:  3.465
 Max.   :38.000   Max.   :5.0300   Max.   :68676.15   Max.   :330.670
      X4               X5               X6               X7
 Min.   :10.61   Min.   : 9.26   Min.   :10.21   Min.   :10.40
 1st Qu.:15.38   1st Qu.:15.46   1st Qu.:15.41   1st Qu.:15.47
 Median :17.52   Median :17.54   Median :17.67   Median :17.53
 Mean   :17.48   Mean   :17.54   Mean   :17.60   Mean   :17.58
 3rd Qu.:19.65   3rd Qu.:19.62   3rd Qu.:19.73   3rd Qu.:19.71
 Max.   :25.16   Max.   :24.67   Max.   :24.36   Max.   :24.73
      X8               X9               X10              X11
 Min.   :0.0000   Min.   :0.0000   Min.   :  2.96   Min.   :-2.1800
 1st Qu.:0.2500   1st Qu.:0.4900   1st Qu.: 68.87   1st Qu.: 0.8100
 Median :0.5000   Median :0.9800   Median : 92.76   Median : 0.9900
 Mean   :0.4981   Mean   :0.9904   Mean   : 93.36   Mean   : 0.8545
 3rd Qu.:0.7500   3rd Qu.:1.5000   3rd Qu.:117.28   3rd Qu.: 1.1100
 Max.   :1.0000   Max.   :2.0000   Max.   :181.18   Max.   : 1.5800
     X12              X13
 Min.   : 0.01   Length:1999
 1st Qu.:13.34   Class :character
 Median :25.67   Mode  :character
 Mean   :25.48
 3rd Qu.:38.00
 Max.   :49.98
```

Todas las variables explicativas son continuas, excepto la variable `X13` que debemos codificar como factor.

```
datos$X13 <- factor(datos$X13, levels = unique(datos$X13))
```

```
ggplot(datos, aes(x = var_obj, fill = X13)) +
  geom_histogram(binwidth=1) +
  facet_grid(X13 ~ ., margins=TRUE, scales="free")
```



Se aprecia como según la clase de `X13`, los valores de `var_obj` se mueven en un rango ligeramente distinto, siendo menores los valores para la clase A y mayores para la clase D.

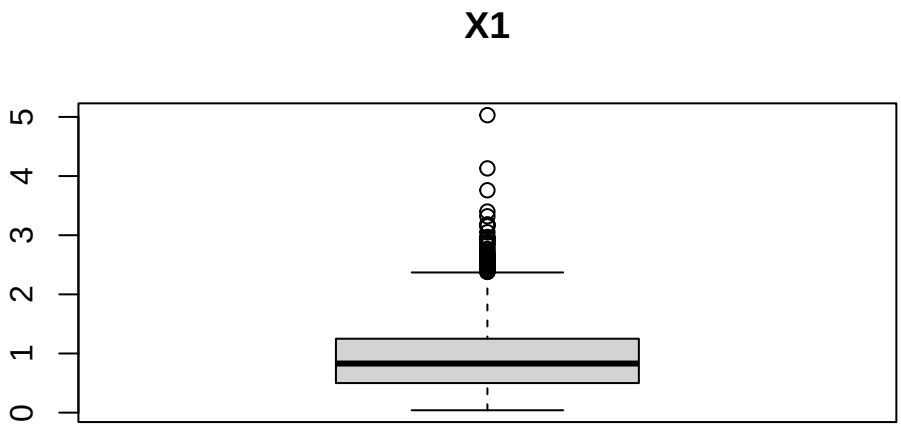
Vamos a detectar las variables continuas que contengan outliers y tomar la transformación logarítmica en los casos en los que sea posible, es decir, si los valores que toma la variable son todos positivos.

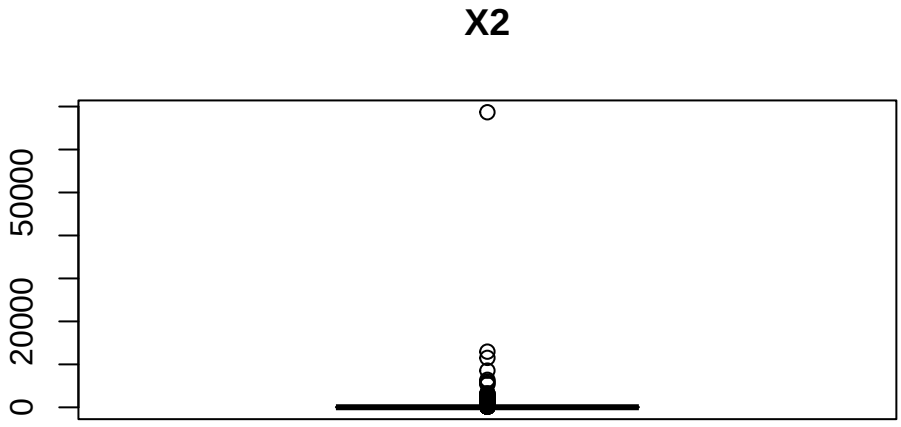
```
varout = rep(FALSE, ncol(datos)-2)
names(varout) = names(datos)[-c(1,ncol(datos))]
for (i in names(varout)){
  varout[i] = length(boxplot(datos[i], plot = FALSE)$out) > 0
}
```

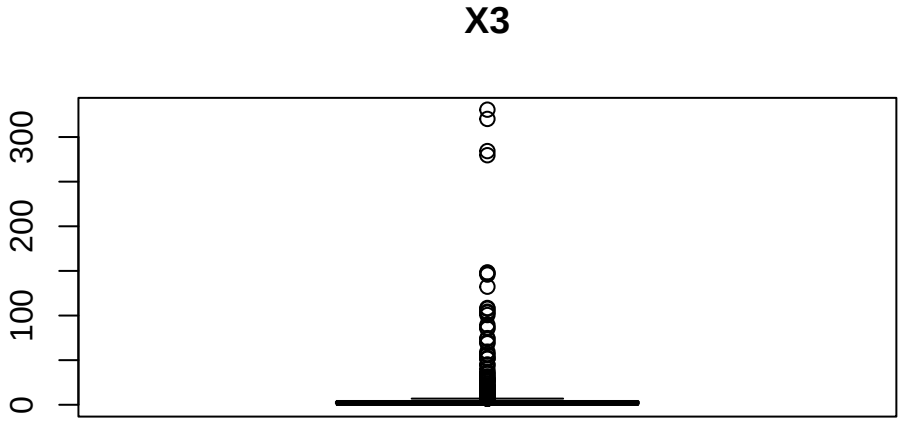
```r
cat("Las variables", paste(names(varout[varout]), collapse = ", "),
 ↪  "contienen outliers\n")
```

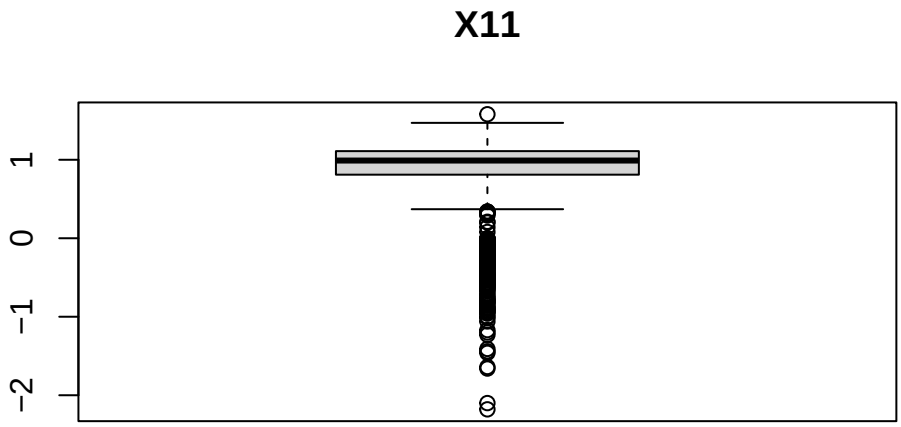
```
Las variables X1, X2, X3, X11 contienen outliers
```

```r
for (i in names(varout[varout])){
  boxplot(datos[i], main = i)
  if (all(datos[i]>0)){
    datos[i] = log(datos[i])
    names(datos)[which(names(datos) == i)] = paste(i , "_trans", sep =
     ↪  "")
  }
}
```



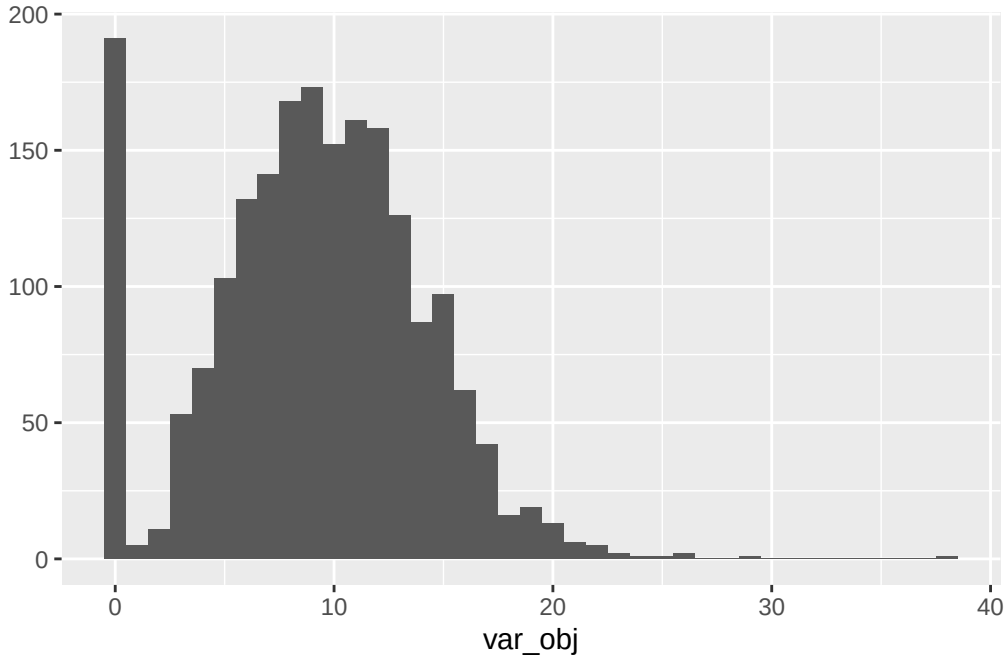

X2
0
20000
50000


X3
0
100
200
300



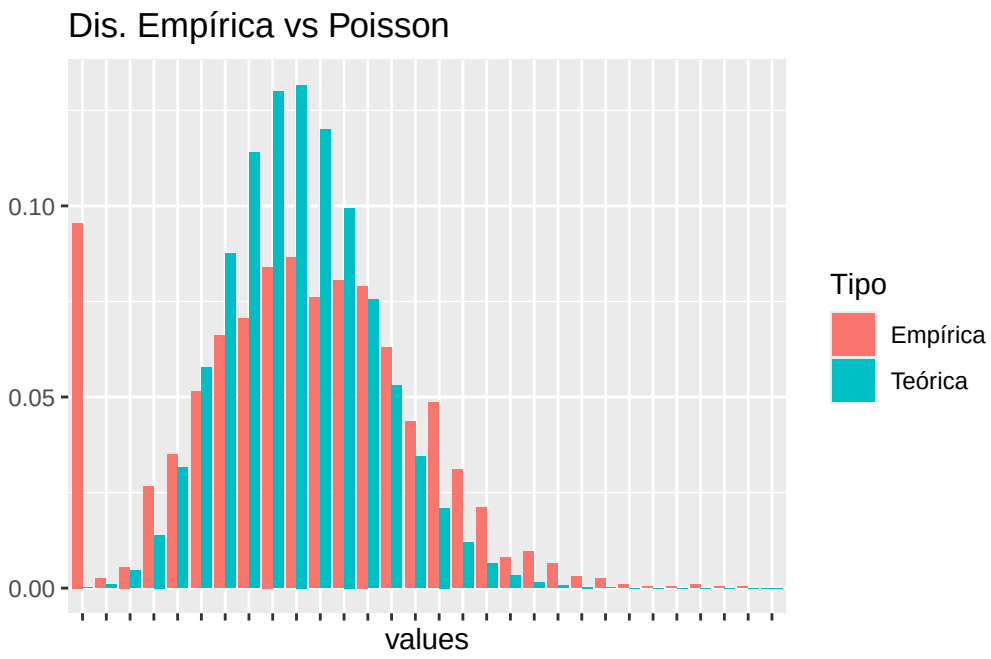
Veamos el histograma de `var_obj`, que es una variable discreta de conteo.

```
ggplot(datos, aes(x = var_obj)) +
  geom_histogram( binwidth = 1) +
  labs(y = "")
```

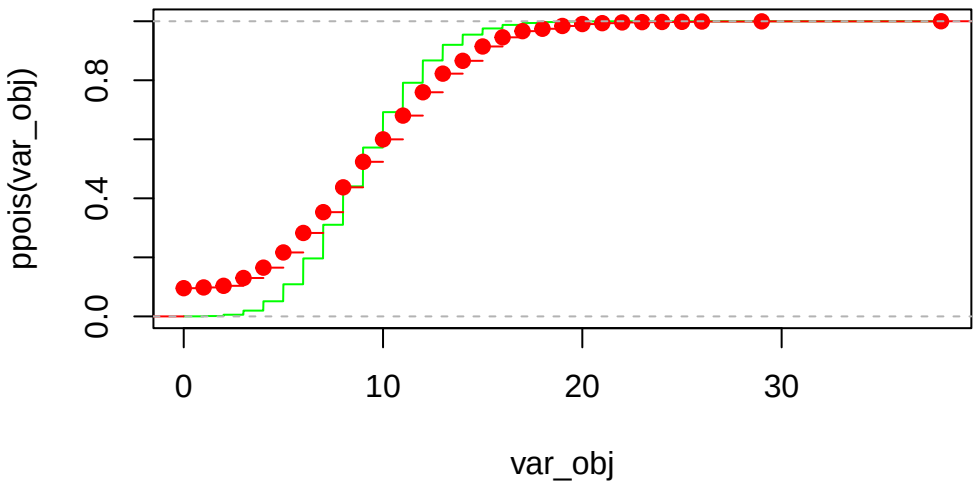


Vamos a compararlo con la función de probabilidad de la Poisson con parámetro λ igual a la media muestral.

```
df1<-data.frame(table(datos$var_obj))
names(df1)<-c("values","Freq")
df1$Tipo<-"Empírica"
df1$Freq<-df1$Freq/sum(df1$Freq)
media<-mean(datos$var_obj)
rango = range(datos$var_obj)[1]:range(datos$var_obj)[2]
df2<-data.frame(values=rango, Freq = dpois(rango,lambda=media))
df2$Tipo<-"Teórica"
df<-rbind(df1,df2)

(grafico1 = ggplot(data=df, aes(x=values, y=Freq, fill=Tipo)) +
    geom_bar(stat="identity", position=position_dodge()) +
    labs(title = "Dis. Empírica vs Poisson",
       y = "") +
    theme(axis.text.x = element_blank()))
```



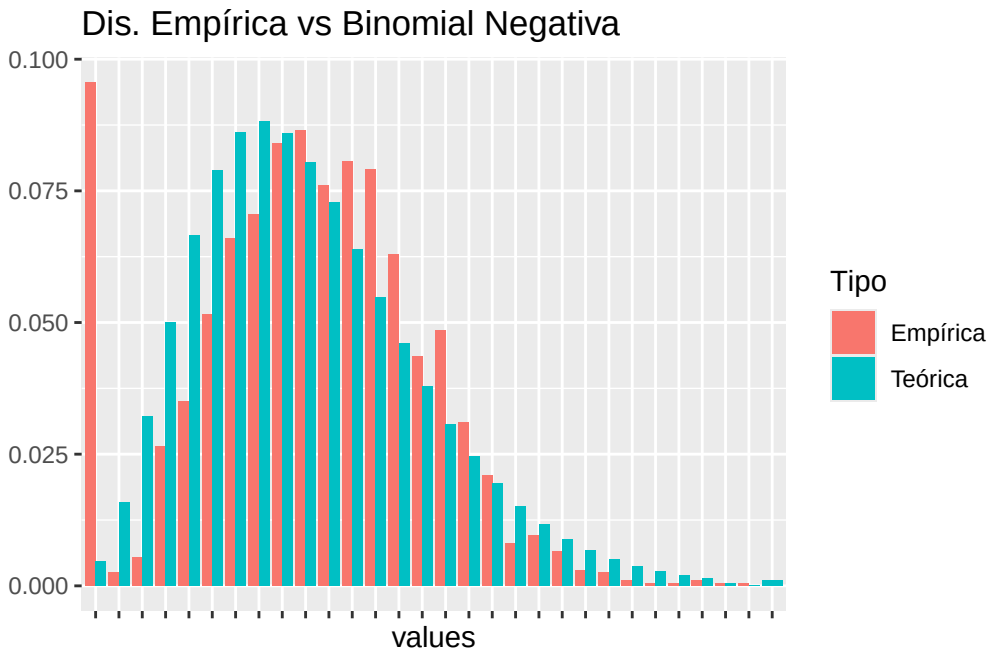

```
xempp <- seq(min(datos$var_obj), max(datos$var_obj), by=0.01)
plot(xempp, ppois(xempp, lambda=media), type="l", col="green",
 ↪  xlab="var_obj",
ylab="ppois(var_obj)")
plot(ecdf(datos$var_obj), col="red",add=TRUE)
```



Comparamos con la Binomial Negativa

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

media <- mean(datos$var_obj)
varianza <- var(datos$var_obj)
size <- media^2 / (varianza - media)  # size = mu^2 / (sigma^2 - mu)
size <- ifelse(size > 0, size, 1)

rango <- range(datos$var_obj)[1]:range(datos$var_obj)[2]

df2 <- data.frame(values = rango, Freq = dnbinom(rango, size = size, mu
↪  = media))
df2$Tipo <- "Teórica"
df <- rbind(df1, df2)

(grafico2 =ggplot(data = df, aes(x = values, y = Freq, fill = Tipo)) +
    geom_bar(stat = "identity", position = position_dodge()) +
    labs(title = "Dis. Empírica vs Binomial Negativa",
```

```
    y = "") +
  theme(axis.text.x = element_blank()))
```



Hay parecidos razonables entre las distribuciones empíricas y las teóricas, tanto para Poisson como Binomial Negativa. Es por ello que parece razonable la adopción de modelos para ambas familias.

Veamos la posibilidad de multicolinealidad en los datos.

```
usdm::vif(datos[,-c(1,14)])
```

```
   Variables      VIF
1   X1_trans 1.645669
2   X2_trans 1.019701
3         X3 1.042166
4         X4 5.655976
5         X5 5.422963
6         X6 5.014344
7         X7 5.537517
8         X8 1.007844
9         X9 1.003463
10       X10 3.771034
```

```
11        X11 1.591202
12        X12 1.002621
```

Tenemos una multicolinealidad moderada, sobre todo debido a la variable `X1_trans`. Sin embargo, como no existe ninguna variable con un *VIF* mayor que 10, no es claro un problema de multicolinealidad.

Regresión de Poisson

Vamos a realizar el modelo de regresión Poisson incluyendo todas las variables explicativas.

```r
model.poiss <- glm(data=datos,var_obj ~ . , family="poisson")
summary(model.poiss)
```

```
Call:
glm(formula = var_obj ~ ., family = "poisson", data = datos)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.509e-01  9.602e-02  -3.654 0.000258 ***
X1_trans    -1.172e-01  3.590e-02  -3.266 0.001090 **
X2_trans    -1.374e-02  4.281e-03  -3.209 0.001333 **
X3           6.989e-05  6.189e-04   0.113 0.910091
X4          -3.320e-02  7.575e-03  -4.382 1.17e-05 ***
X5          -1.365e-03  5.783e-03  -0.236 0.813387
X6          -4.649e-04  5.380e-03  -0.086 0.931135
X7          -1.413e-03  5.664e-03  -0.249 0.802997
X8          -3.979e-02  2.597e-02  -1.532 0.125521
X9          -1.575e-02  1.295e-02  -1.216 0.223899
X10          6.808e-05  4.385e-04   0.155 0.876614
X11          3.268e+00  1.161e-01  28.144  < 2e-16 ***
X12          9.646e-07  5.123e-04   0.002 0.998498
X13C         3.555e-02  2.664e-02   1.334 0.182074
X13D         3.709e-02  3.965e-02   0.935 0.349603
X13A        -5.966e-02  3.793e-02  -1.573 0.115779
---
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:  112.59  on 1983  degrees of freedom
```

```
AIC: 7509.8

Number of Fisher Scoring iterations: 5
```

```r
bondad = data.frame(AIC = AIC(model.poiss), dev = model.poiss$deviance)
rownames(bondad)[1] = "pois.full"
```

El modelo con todas las variables tiene un gran número de variables que son despreciables. La devianza residual del modelo es 112.59 y tiene un AIC de 7509.8.

```r
model.poiss <- update(model.poiss, . ~ . - X3 - X5 - X6 - X7 - X10 -
 ↪  X12)
summary(model.poiss)
```

```
Call:
glm(formula = var_obj ~ X1_trans + X2_trans + X4 + X8 + X9 +
    X11 + X13, family = "poisson", data = datos)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.390007   0.072779  -5.359 8.38e-08 ***
X1_trans    -0.117502   0.035638  -3.297 0.000977 ***
X2_trans    -0.013827   0.004268  -3.239 0.001198 **
X4          -0.033858   0.005727  -5.912 3.38e-09 ***
X8          -0.039905   0.025964  -1.537 0.124302
X9          -0.015867   0.012933  -1.227 0.219885
X11          3.270223   0.115534  28.305  < 2e-16 ***
X13C         0.035354   0.026610   1.329 0.183990
X13D         0.037227   0.039585   0.940 0.347003
X13A        -0.059775   0.037857  -1.579 0.114346
---
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:  113.11  on 1989  degrees of freedom
AIC: 7498.3

Number of Fisher Scoring iterations: 5
```

La varianza residual en este caso es 113.11, lo que indica que ha experimentado un aumento

con respecto al modelo anterior, aunque dicho incremento no es significativo. Por otro lado, debido a la reducción de la complejidad el *AIC* ha reducido su valor a 7498.3.

Veamos el modelo únicamente con las variables regresoras que son significativas en este momento.

```
model.poiss <- glm(data=datos,var_obj ~ X1_trans + X2_trans + X4 + X11 ,
 ↪  family="poisson")
summary(model.poiss)
```

```
Call:
glm(formula = var_obj ~ X1_trans + X2_trans + X4 + X11, family = "poisson",
    data = datos)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.388038   0.068247  -5.686 1.30e-08 ***
X1_trans    -0.068361   0.017077  -4.003 6.25e-05 ***
X2_trans    -0.011540   0.004144  -2.785  0.00536 **
X4          -0.029032   0.005310  -5.468 4.56e-08 ***
X11          3.162082   0.104390  30.291  < 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:  120.79  on 1994  degrees of freedom
AIC: 7496

Number of Fisher Scoring iterations: 4
```

Experimentamos de nuevo algo de incremento en la devianza residual, pero una disminución del *AIC*.

Con idea de tener un modelo lo más interpretable posible, vamos a escoger aquel con un menor *AIC*.

```
model.poiss_full = glm(data = datos, var_obj ~., family = "poisson")
model.poiss_best<-MASS::stepAIC(model.poiss_full, trace = 0)
summary(model.poiss_best)
```

```
Call:
glm(formula = var_obj ~ X1_trans + X2_trans + X4 + X8 + X11,
    family = "poisson", data = datos)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.376358   0.068799  -5.470 4.49e-08 ***
X1_trans    -0.072305   0.017335  -4.171 3.03e-05 ***
X2_trans    -0.012326   0.004184  -2.946  0.00322 **
X4          -0.030298   0.005411  -5.599 2.15e-08 ***
X8          -0.038374   0.025930  -1.480  0.13890
X11          3.193881   0.107615  29.679  < 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: 6754.4  on 1998  degrees of freedom
Residual deviance:  118.6  on 1993  degrees of freedom
AIC: 7495.8

Number of Fisher Scoring iterations: 5
```

```r
bondad = rbind(bondad, data.frame(AIC = AIC(model.poiss_best), dev =
 ↪  model.poiss_best$deviance))
rownames(bondad)[nrow(bondad)] = "pois.red"
```

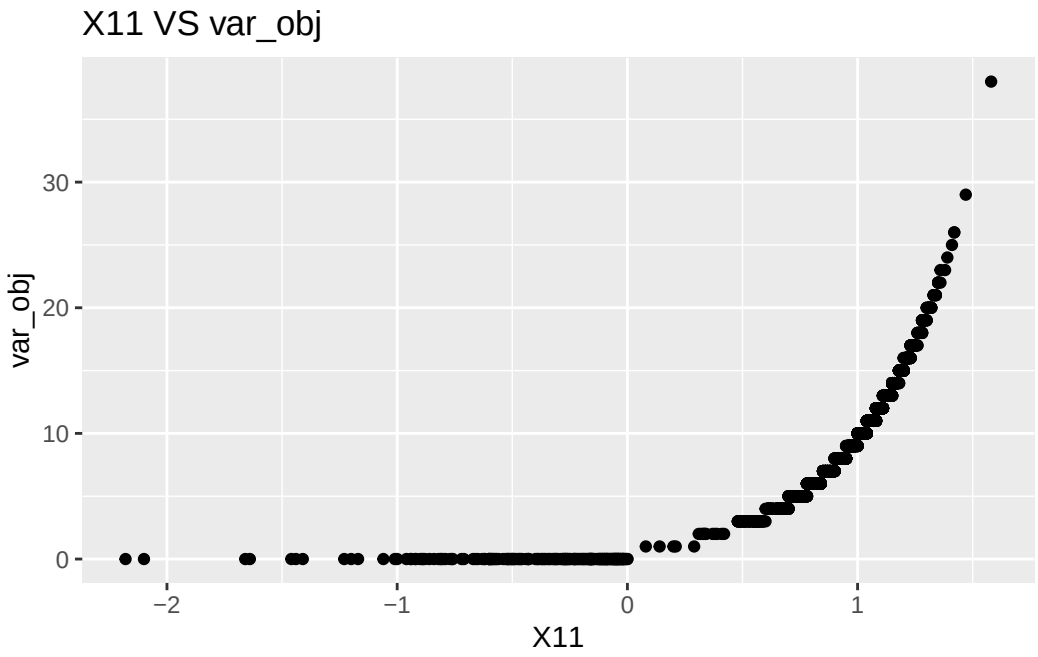
Las variables escogidas para el modelo son `X1_trans`, `X2_trans`, `X4`, `X8` y `X11`. Teniendo en cuenta que

$$E(Y_i \mid x_i) = \exp\left(\beta_0 + \beta_1 X1\_trans_i + \beta_2 X2\_trans_i + \beta_3 X4_i + \beta_4 X8_i + \beta_5 X11_i\right)$$

todas las variables producen una disminución de `var_ob` excepto la variable `X11`, que es la más significativa, siendo el valor del parámetro correspondiente igual a 3.19. Por tanto, por cada unidad adicional en $X11$ el valor de `var_obj` se multiplica por $\exp(3.19) \approx 24.3$.

Representemos gráficamente la variable `X11` frente a `var_obj`.

```r
ggplot(data = datos, aes(x = X11, y = var_obj))+
  geom_point() +
  labs(title = "X11 VS var_obj")
```



Podemos observar un comportamiento particularmente interesante. Y es que, para los valores de `X11` negativos son los casos en los que la variable objetivo toma un valor nulo, mientras que a medida que la variable predictora comienza a tomar valores positivos, el valor de `var_obj` comienza a dispararse.

El modelo de regresión de Poisson tiene como hipótesis la igualdad de media y varianza. En caso de que la varianza sea mayor que la media, es más recomendable emplear un modelo Quasi-Poisson, que supone que la varianza es una función lineal de la media.

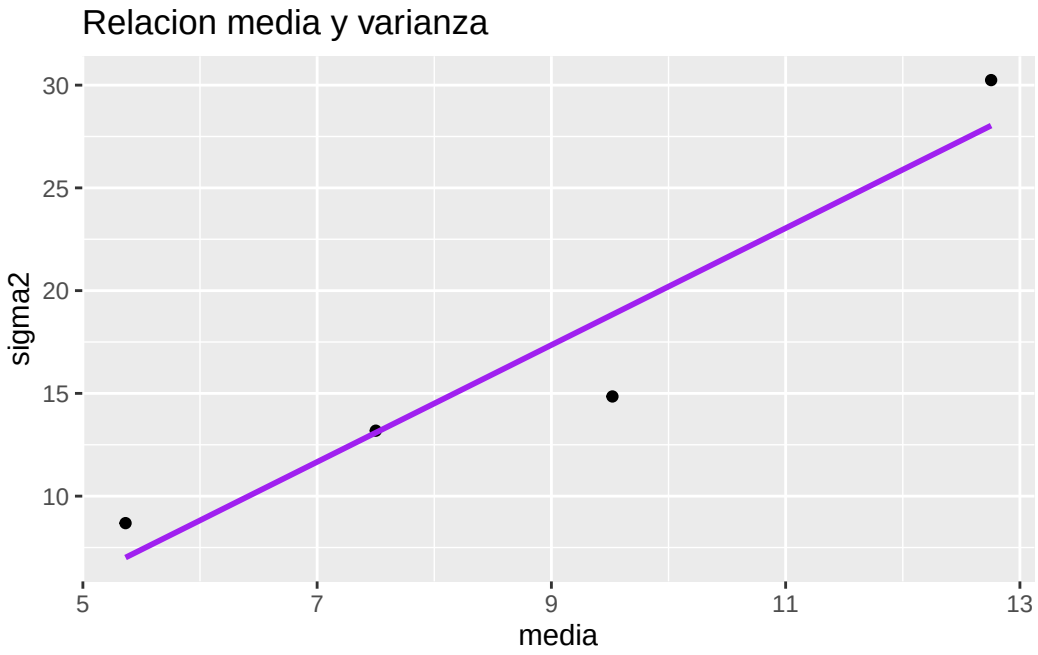
Veamos la relación de la media y varianza según la clase de $X13$.

```
nd = dim(datos)[1]
rel_var_media = datos %>% group_by(X13) %>%
  summarise(media = mean(var_obj), sigma2 = (nd-1)*var(var_obj)/nd)

rel_var_media
```

```
# A tibble: 4 x 3
  X13   media sigma2
  <fct> <dbl>  <dbl>
1 B      7.50  13.2
2 C      9.52  14.9
3 D     12.8   30.2
```

```
4 A      5.36   8.69
```

```
ggplot(data = rel_var_media, aes(x = media, y = sigma2)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "purple") +
  labs(title= "Relacion media y varianza")
```



Realicemos el test de igualdad de media y varianza.

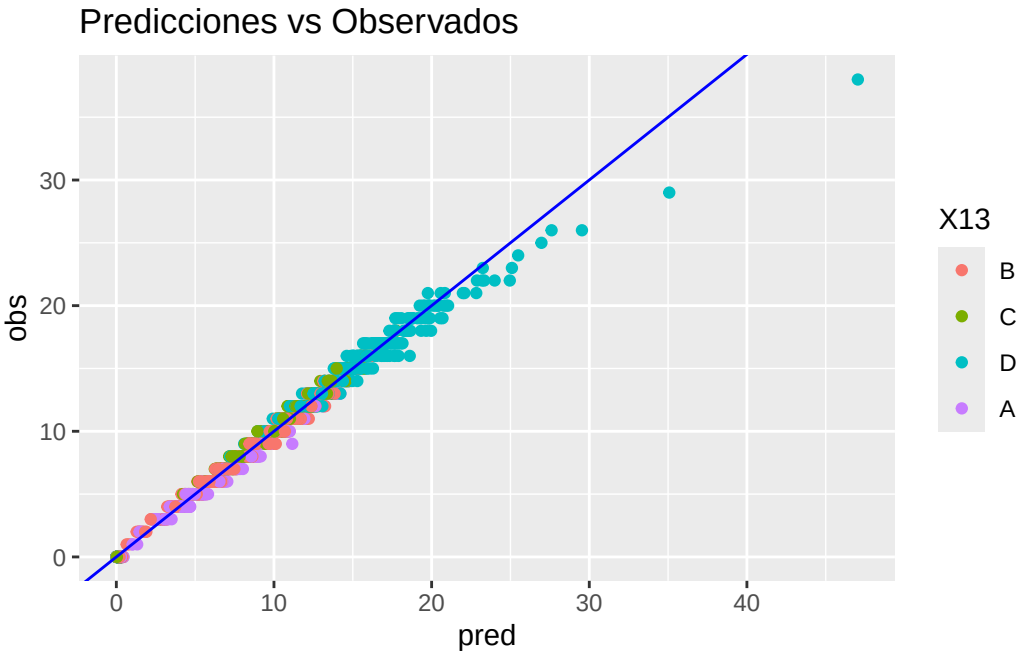
```
dispersiontest(model.poiss_best)
```

```
    Overdispersion test

data:  model.poiss_best
z = -110.93, p-value = 1
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
 0.1283027
```

Aceptamos la igualdad de media y varianza, por tanto es correcto el modelo Poisson.

Veamos una comparación entre las predicciones y los valores observados.

```
predicciones <- predict(model.poiss_best, type = "response")
observados <- datos$var_obj

ggplot(data = data.frame(pred = predicciones, obs = observados, X13 =
↪  datos$X13),
       aes(x = pred, y = obs, color = X13)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "blue") +
  labs(title = "Predicciones vs Observados")
```



MLG Binomial Negativa

Realicemos el modelo para la Binomial Negativa, aunque debido a que hemos aceptado la igualdad de media y varianza no sería necesario.

Realicemos en primer lugar el modelo con todas las variables.

```
model.bn_full = glm.nb(var_obj~., data = datos)
summary(model.bn_full)
```

```
Call:
glm.nb(formula = var_obj ~ ., data = datos, init.theta = 234752.732,
    link = log)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.509e-01  9.602e-02  -3.654 0.000258 ***
X1_trans    -1.172e-01  3.590e-02  -3.266 0.001090 **
X2_trans    -1.374e-02  4.281e-03  -3.209 0.001333 **
X3           6.990e-05  6.189e-04   0.113 0.910084
X4          -3.320e-02  7.575e-03  -4.383 1.17e-05 ***
X5          -1.365e-03  5.784e-03  -0.236 0.813404
X6          -4.648e-04  5.380e-03  -0.086 0.931144
X7          -1.413e-03  5.664e-03  -0.249 0.802989
X8          -3.979e-02  2.597e-02  -1.532 0.125519
X9          -1.575e-02  1.295e-02  -1.216 0.223899
X10          6.808e-05  4.385e-04   0.155 0.876612
X11          3.268e+00  1.161e-01  28.144  < 2e-16 ***
X12          9.596e-07  5.123e-04   0.002 0.998505
X13C         3.555e-02  2.664e-02   1.334 0.182092
X13D         3.709e-02  3.965e-02   0.935 0.349639
X13A        -5.966e-02  3.793e-02  -1.573 0.115791
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(234752.7) family taken to be 1)

    Null deviance: 6754.21  on 1998  degrees of freedom
Residual deviance:  112.59  on 1983  degrees of freedom
AIC: 7511.9

Number of Fisher Scoring iterations: 1

              Theta:  234753
          Std. Err.:  465699
Warning while fitting theta: iteration limit reached

 2 x log-likelihood:  -7477.859
```

Tomemos ahora el modelo con las variables finales del modelo de Poisson.

```r
formula = "X1_trans + X2_trans + X4 + X8 + X11"
formula = as.formula(paste("var_obj~", formula))
model.bn_best = glm.nb(formula, data = datos)
summary(model.bn_best)
```

```
Call:
glm.nb(formula = formula, data = datos, init.theta = 235362.4878,
    link = log)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.376360   0.068800  -5.470 4.49e-08 ***
X1_trans    -0.072307   0.017335  -4.171 3.03e-05 ***
X2_trans    -0.012327   0.004184  -2.946  0.00322 **
X4          -0.030299   0.005411  -5.599 2.15e-08 ***
X8          -0.038375   0.025931  -1.480  0.13890
X11          3.193906   0.107616  29.679  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(235362.5) family taken to be 1)

    Null deviance: 6754.2  on 1998  degrees of freedom
Residual deviance:  118.6  on 1993  degrees of freedom
AIC: 7497.9

Number of Fisher Scoring iterations: 1

              Theta:  235362
          Std. Err.:  468234
Warning while fitting theta: iteration limit reached

 2 x log-likelihood:  -7483.869
```

```r
anova(model.bn_full, model.bn_best)
```

```
Likelihood ratio tests of Negative Binomial Models

Response: var_obj

                                                                      Model
```

```
1                                          X1_trans + X2_trans + X4 + X8 +
X11
2 X1_trans + X2_trans + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12 +
X13
     theta Resid. df    2 x log-lik.   Test    df LR stat.   Pr(Chi)
1 235362.5      1993       -7483.869
2 234752.7      1983       -7477.859 1 vs 2    10 6.010086 0.8144151
```

Aceptamos el test anova, por tanto escogemos el modelo con un menor número de variables.

```
m = update(model.bn_best, . ~. - X8)
anova(model.bn_best, m)
```

```
Likelihood ratio tests of Negative Binomial Models

Response: var_obj
                              Model    theta Resid. df    2 x log-lik.
                              Test
1      X1_trans + X2_trans + X4 + X11 235407.4      1994       -7486.059
2 X1_trans + X2_trans + X4 + X8 + X11 235362.5      1993       -7483.869 1 vs
2
     df LR stat.   Pr(Chi)
1
2     1 2.190386 0.1388745
```

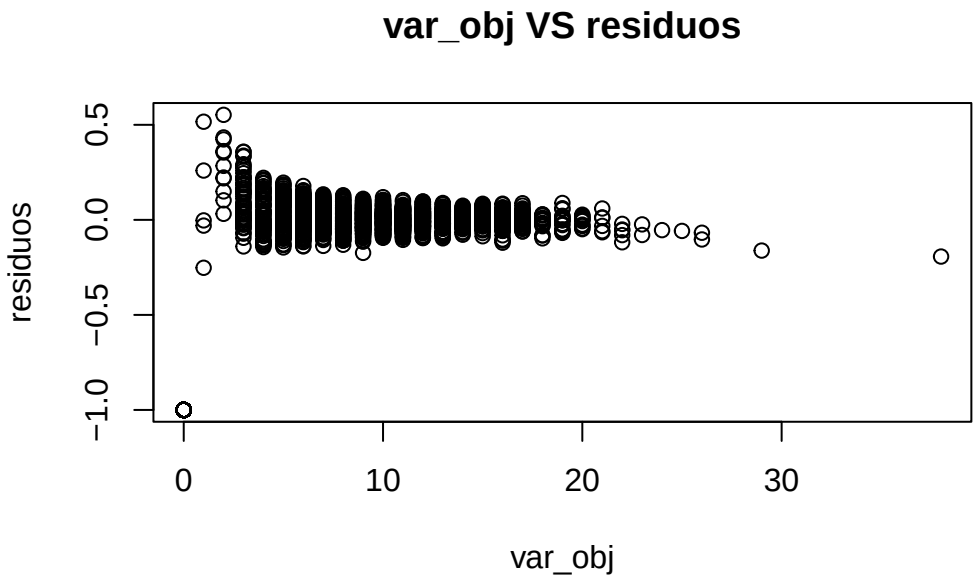
Podemos escoger el modelo eliminando la variable `X8`.

```
model.bn_best = m
summary(model.bn_best)
```

```
Call:
glm.nb(formula = var_obj ~ X1_trans + X2_trans + X4 + X11, data = datos,
    init.theta = 235407.3655, link = log)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.388041   0.068249  -5.686 1.30e-08 ***
X1_trans    -0.068362   0.017078  -4.003 6.25e-05 ***
X2_trans    -0.011540   0.004144  -2.785  0.00536 **
X4          -0.029033   0.005310  -5.468 4.56e-08 ***
X11          3.162105   0.104396  30.290  < 2e-16 ***
---
```

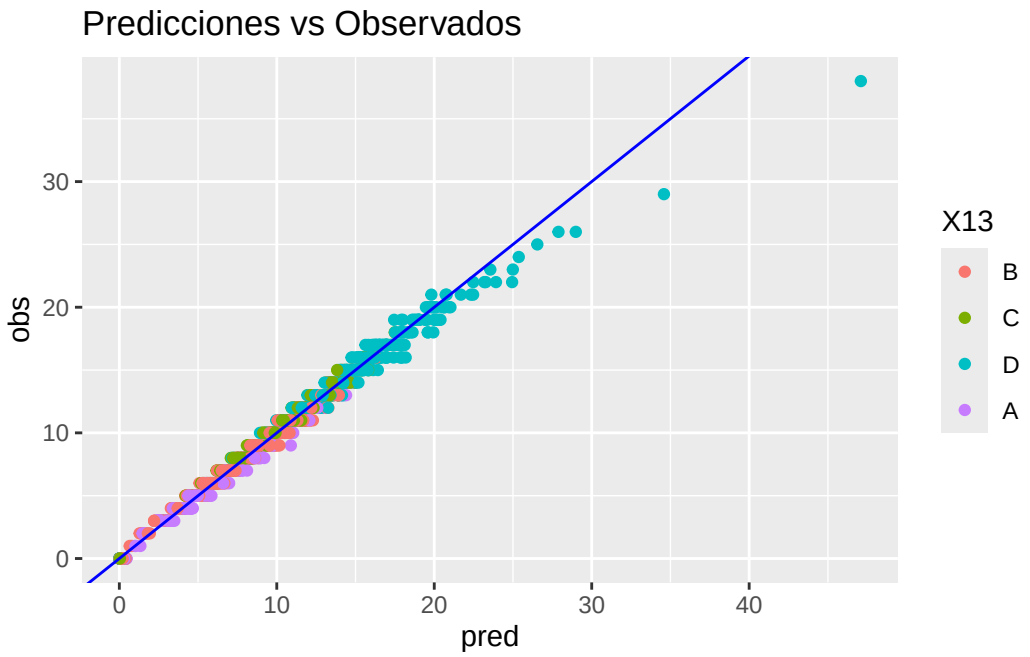
```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(235407.4) family taken to be 1)

    Null deviance: 6754.21  on 1998  degrees of freedom
Residual deviance:  120.79  on 1994  degrees of freedom
AIC: 7498.1

Number of Fisher Scoring iterations: 1

              Theta:  235407
          Std. Err.:  468248
Warning while fitting theta: iteration limit reached

 2 x log-likelihood:  -7486.059
```

```r
plot(data.frame(datos$var_obj,model.bn_best$resid), main = "var_obj VS
 ↪  residuos",
     xlab = "var_obj", ylab = "residuos")
```



Observamos como la varianza de los residuos es mayor conforme los valores de la variable objetivo son menores.

```r
predicciones <- predict(model.bn_best, type = "response")
observados <- datos$var_obj

ggplot(data = data.frame(pred = predicciones, obs = observados, X13 =
 ↪  datos$X13),
       aes(x = pred, y = obs, color = X13)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "blue") +
  labs(title = "Predicciones vs Observados")
```



Comparemos con el modelo de Poisson.

```r
bn = with(model.bn_best, cbind(res.deviance = deviance, df =
 ↪  df.residual,
               AIC = aic, p = pchisq(deviance, df.residual,
                ↪  lower.tail=FALSE)))

pois = with(model.poiss_best, cbind(res.deviance = deviance, df =
 ↪  df.residual,
               AIC = aic, p = pchisq(deviance, df.residual,
                ↪  lower.tail=FALSE)))
```

```
  comp = data.frame(rbind(bn,pois))
  rownames(comp) = c("bin.neg", "poisson")
  comp
```

```
        res.deviance   df      AIC p
bin.neg     120.7888 1994 7498.059 1
poisson     118.6009 1993 7495.794 1
```

```
  X2 <- 2 * (logLik(m) - logLik(model.poiss_best))
  X2
```

```
'log Lik.' -2.265253 (df=6)
```

```
  pchisq(X2, df = 1, lower.tail=FALSE)
```

```
'log Lik.' 1 (df=6)
```

En definitiva, no hay diferencias razonables entre los modelos de Poisson y Binomial Negativa. Escogeremos el modelo de Poisson, pues el correspondiente a la Binomial Negativa añade un parámetro de dispersión que no es necesario y por tanto añade complejidad.

Veamos como maneja los ceros el modelo.

```
  sum(datos$var_obj==0)
```

```
[1] 191
```

```
  sum(round(model.poiss_best$fitted.values)==0)
```

```
[1] 191
```

No va a ser necesario por tanto modelos ideados para excesos de ceros, como ZAP, ZAPNB, ZIP o ZINP.

Modelos aditivos generalizados

Planteamos en primer lugar el modelo incluyendo todas las variables, excluyendo la variable categórica `X13`. Tomamos el tipo de spline por defecto (spline de regresión de placa delgada).

```r
model.gam_fulltp=gam(var_obj~s(X1_trans)+s(X2_trans)+s(X3)+s(X4)+s(X5)+s(X6)+s(X7)+s(X8)
              +s(X9)+s(X10)+s(X11)+s(X12),
              family=poisson,data=datos)
summary(model.gam_fulltp)
```

```
Family: poisson
Link function: log

Formula:
var_obj ~ s(X1_trans) + s(X2_trans) + s(X3) + s(X4) + 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) -0.08183    1.12997  -0.072    0.942

Approximate significance of smooth terms:
              edf Ref.df  Chi.sq p-value
s(X1_trans) 1.000  1.001   0.874   0.350
s(X2_trans) 1.000  1.000   0.848   0.357
s(X3)       1.000  1.000   0.022   0.881
s(X4)       1.000  1.000   0.941   0.332
s(X5)       1.000  1.000   0.003   0.959
s(X6)       1.000  1.000   0.032   0.858
s(X7)       1.000  1.000   0.018   0.893
s(X8)       1.000  1.000   0.090   0.764
s(X9)       1.000  1.000   0.179   0.672
s(X10)      1.000  1.000   0.000   0.988
s(X11)      5.813  6.002 227.433  <2e-16 ***
s(X12)      1.000  1.000   0.002   0.966
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.997   Deviance explained = 99.7%
UBRE = -0.97214  Scale est. = 1         n = 1999
```

```r
bondad = rbind(bondad, data.frame(AIC = AIC(model.gam_fulltp), dev =
 ↪  model.gam_fulltp$deviance))
rownames(bondad)[nrow(bondad)] = "gam.full"
bondad
```

```
               AIC       dev
pois.full 7509.784 112.59043
pois.red  7495.794 118.60095
gam.full  7420.893  20.07102
```

Observemos que solo se rechaza el contraste de significación para la variable `X11`, lo que quiere decir que estas componentes se pueden considerar lineales en el modelo.

La devianza de este modelo es considerablemente menor que los obtenidos anteriormente. El AIC también es bastante menor pese a haber incluido bastante complejidad en el modelo. Notemos también un R^2_{adj} de 0.997, lo que podría venir debido a un problema de concurvidad.

```
concurvity(model.gam_fulltp)
```

```
                 para s(X1_trans) s(X2_trans)      s(X3)     s(X4)     s(X5)
worst    8.534483e-22   0.7500835   0.3007732 0.18777169 0.9186791 0.8588488
observed 8.534483e-22   0.5733909   0.2993984 0.12022539 0.9067218 0.8362430
estimate 8.534483e-22   0.5057328   0.2047236 0.09208301 0.8452130 0.7650285
             s(X6)     s(X7)      s(X8)      s(X9)    s(X10)    s(X11)
worst    0.8559399 0.9186696 0.09773346 0.09395599 0.7660037 0.8898312
observed 0.8312689 0.8517819 0.08500326 0.07820874 0.7588971 0.5061541
estimate 0.7637182 0.8057073 0.07992844 0.07536437 0.6615673 0.7342535
             s(X12)
worst    0.07642063
observed 0.05020144
estimate 0.04972798
```

Procedemos a eliminar del modelo las variables `X4` y `X7` pues presentan los valores más altos.

```
model.gam=gam(var_obj~s(X1_trans)+s(X2_trans)+s(X3)+s(X5)+s(X6)+s(X8)+s(X9)+
               s(X10)+s(X11)+s(X12),
             family=poisson,data=datos)
summary(model.gam)
```

```
Family: poisson
Link function: log

Formula:
var_obj ~ s(X1_trans) + s(X2_trans) + s(X3) + s(X5) + s(X6) +
    s(X8) + s(X9) + s(X10) + s(X11) + s(X12)

Parametric coefficients:
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1286     1.1594  -0.111    0.912

Approximate significance of smooth terms:
              edf Ref.df  Chi.sq p-value
s(X1_trans) 1.000  1.000   0.251   0.616
s(X2_trans) 1.000  1.000   0.237   0.627
s(X3)       1.000  1.000   0.020   0.887
s(X5)       1.000  1.000   0.146   0.703
s(X6)       1.000  1.000   0.020   0.887
s(X8)       1.000  1.000   0.002   0.961
s(X9)       1.000  1.000   0.037   0.847
s(X10)      1.000  1.000   0.057   0.812
s(X11)      5.825  6.009 486.782  <2e-16 ***
s(X12)      1.000  1.000   0.001   0.978
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.997   Deviance explained = 99.7%
UBRE = -0.97369  Scale est. = 1         n = 1999
```

```
AIC(model.gam)
```

```
[1] 7417.796
```

Los resultados son muy similares, y logramos reducir en gran medida el *AIC*. Sigamos estudiando la concurvidad.

```
concurvity(model.gam)
```

```
                para s(X1_trans) s(X2_trans)      s(X3)     s(X5)     s(X6)
worst    7.46883e-22   0.6960512   0.2167315 0.17724966 0.8060251 0.4454782
observed 7.46883e-22   0.5520731   0.2119835 0.10957284 0.7816938 0.2463880
estimate 7.46883e-22   0.4830229   0.1479930 0.07950935 0.7021930 0.2397350
              s(X8)      s(X9)    s(X10)    s(X11)     s(X12)
worst    0.07687468 0.07356114 0.7425739 0.8161682 0.06451535
observed 0.06848868 0.05676968 0.7353194 0.2513440 0.03458145
estimate 0.06461442 0.05577813 0.6383015 0.4455683 0.03517419
```

Eliminamos ahora las variables `X5` y `X10`.

```r
model.gam=gam(var_obj~s(X1_trans)+s(X2_trans)+s(X3)+s(X6)+s(X8)
              +s(X9)++s(X11)+s(X12),
              family=poisson,data=datos)
summary(model.gam)
```

```
Family: poisson
Link function: log

Formula:
var_obj ~ s(X1_trans) + s(X2_trans) + s(X3) + s(X6) + s(X8) +
    s(X9) + +s(X11) + s(X12)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1386     1.1664  -0.119    0.905

Approximate significance of smooth terms:
              edf Ref.df   Chi.sq p-value
s(X1_trans) 1.000  1.000    0.139   0.710
s(X2_trans) 1.000  1.000    0.083   0.773
s(X3)       1.000  1.000    0.036   0.849
s(X6)       1.000  1.000    0.019   0.891
s(X8)       1.000  1.000    0.002   0.968
s(X9)       1.000  1.000    0.016   0.900
s(X11)      5.827  6.011 1130.452  <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 = -0.97551  Scale est. = 1         n = 1999
```

```r
AIC(model.gam)
```

```
[1] 7414.142
```

```r
concurvity(model.gam)
```

```
                para s(X1_trans) s(X2_trans)      s(X3)     s(X6)      s(X8)
worst    7.159814e-22   0.6702219  0.12147483 0.13620676 0.4404886 0.06467247
```

```
observed 7.159814e-22   0.5272180  0.11484996 0.06701720 0.2399647 0.05457660
estimate 7.159814e-22   0.4593052  0.08544005 0.04875849 0.2330751 0.05095532
              s(X9)    s(X11)     s(X12)
worst    0.05409244 0.6782213 0.04553591
observed 0.04315636 0.2432647 0.02509528
estimate 0.04219070 0.3887580 0.02480953
```

Seguimos mejorando el AIC, mientras que el R^2_{adj} y el porcentaje de la devianza explicada sigue invariante. Por ello, continuemos eliminando variables del modelo, en este caso `X1_trans` y `X6`.

```
model.gam=gam(var_obj~s(X2_trans)+s(X3)+s(X8)+s(X9)+s(X11)+s(X12),
              family=poisson,data=datos)
summary(model.gam)
```

```
Family: poisson
Link function: log

Formula:
var_obj ~ s(X2_trans) + s(X3) + s(X8) + s(X9) + s(X11) + s(X12)

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1427     1.1681  -0.122    0.903

Approximate significance of smooth terms:
              edf Ref.df   Chi.sq p-value
s(X2_trans) 1.000  1.000    0.038   0.846
s(X3)       1.000  1.000    0.033   0.855
s(X8)       1.000  1.000    0.009   0.926
s(X9)       1.000  1.000    0.009   0.925
s(X11)      5.828  6.011 2603.257  <2e-16 ***
s(X12)      1.000  1.000    0.000   0.995
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.997   Deviance explained = 99.7%
UBRE = -0.97743  Scale est. = 1         n = 1999
```

```
AIC(model.gam)
```

```
[1] 7410.301
```

```
concurvity(model.gam)
```

```
                 para s(X2_trans)      s(X3)      s(X8)      s(X9)     s(X11)
worst    6.735351e-22  0.06014400 0.12445222 0.03971308 0.03798578 0.16108894
observed 6.735351e-22  0.05302919 0.06315362 0.03432974 0.03147283 0.01763415
estimate 6.735351e-22  0.04251914 0.04291111 0.03188402 0.03074584 0.04411130
             s(X12)
worst    0.03321685
observed 0.01672769
estimate 0.01667042
```

Continuamos mejorando el *AIC* mientras que la bondad de ajuste del modelo sigue siendo idéntica. Hemos solucionado en este punto los problemas de concurvidad, pero se siguen rechazando con firmeza los contrastes individuales sobre las variables. Llegados a este punto, planteamos el modelo que incluye tan solo la variable `X11`, pues parece ser que las relaciones no lineales de esta con la variable objetivo explican casi por completo esta, y el modelo las está capturando.

```
model.gam_X11tp=gam(var_obj~s(X11), family=poisson,data=datos)
summary(model.gam_X11tp)
```

```
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
```

```r
bondad = rbind(bondad, data.frame(AIC = AIC(model.gam_X11tp), dev =
 ↪  model.gam_X11tp$deviance))
rownames(bondad)[nrow(bondad)] = "gam.X11tp"
bondad
```

```
               AIC       dev
pois.full 7509.784 112.59043
pois.red  7495.794 118.60095
gam.full  7420.893  20.07102
gam.X11tp 7400.386  21.53563
```

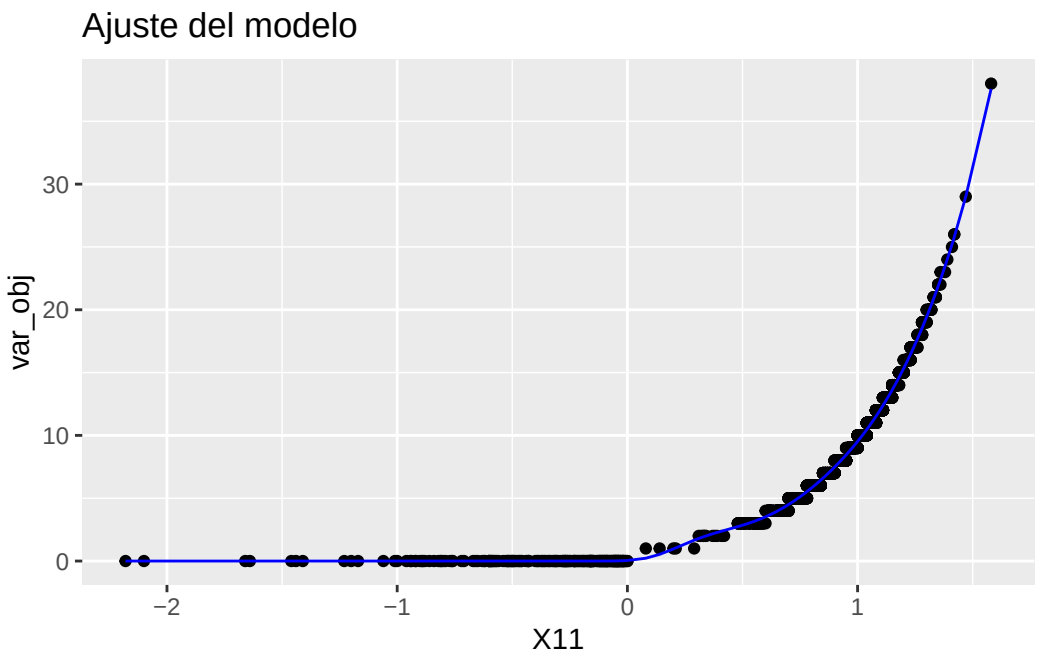
Observamos una disminución clara del AIC y un aumento no muy significante de la devianza residual. Hagamos el test anova para comparar con el modelo con todas las variables.

```r
resultado_anova = anova.gam(model.gam_fulltp, model.gam_X11tp)
diff_dev <- resultado_anova$Deviance[2]
diff_df <- abs(resultado_anova$Df[2])

pchisq(diff_dev, df = diff_df, lower.tail = FALSE)
```

```
[1] 1
```

Se acepta por tanto la igualdad de modelos y por tanto el modelo con tan solo la variable `X11` como regresora es el mejor, debido a su simplicidad.

```r
ggplot(data = datos, aes(x = datos$X11, y = datos$var_obj)) +
  geom_point() +
  geom_line(aes(y = model.gam_X11tp$fitted.values), color = "blue") +
  labs(title = "Ajuste del modelo",
       x = "X11",
       y = "var_obj")
```



Veamos como afecta la inlusión de la variable categórica.

```
model.gam=gam(var_obj~s(X11,by=X13),family=poisson,data=datos)
summary(model.gam)
```

```
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):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 ***
s(X11):X13A 4.448  4.996  273.7  <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
```

```
data.frame(AIC = AIC(model.gam), dev = model.gam$deviance)
```

```
       AIC      dev
1 7426.984 26.33163
```

Mantenemos por tanto el modelo con `X11`.

Veamos ahora el modelo con la variable `X11` modificando el tipo de spline. En primer lugar el spline de regresión de placa delgada penalizada.

```
model.gam_X11ts=gam(var_obj~s(X11,bs="ts"),family=poisson,data=datos)
summary(model.gam_X11ts)
```

```
Family: poisson
Link function: log

Formula:
var_obj ~ s(X11, bs = "ts")

Parametric coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1396     1.1611   -0.12    0.904

Approximate significance of smooth terms:
         edf Ref.df Chi.sq p-value
s(X11) 5.827      9   2789  <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
```

```r
bondad = rbind(bondad, data.frame(AIC = AIC(model.gam_X11ts), dev =
 ↪  model.gam_X11ts$deviance))
rownames(bondad)[nrow(bondad)] = "gam.X11ts"
bondad
```

```
               AIC       dev
pois.full 7509.784 112.59043
pois.red  7495.794 118.60095
gam.full  7420.893  20.07102
gam.X11tp 7400.386  21.53563
gam.X11ts 7400.385  21.53674
```

Usando splines de regresión cúbicos.

```r
model.gam_X11cr=gam(var_obj~s(X11,bs="cr"),family=poisson,data=datos)
summary(model.gam_X11cr)
```

```
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
```

```r
bondad = rbind(bondad, data.frame(AIC = AIC(model.gam_X11cr), dev =
 ↪  model.gam_X11cr$deviance))
rownames(bondad)[nrow(bondad)] = "gam.X11cr"
bondad
```

```
               AIC       dev
pois.full 7509.784 112.59043
pois.red  7495.794 118.60095
gam.full  7420.893  20.07102
gam.X11tp 7400.386  21.53563
gam.X11ts 7400.385  21.53674
gam.X11cr 7401.010  21.99967
```

Los resultados son muy similares.

Regresión Polinómica

Procedamos ahora al ajuste mediante regresiones de tipo polinómicas. Teniendo en cuenta los resultados anteriores, emplearemos `X11` como única variable regresora. Vamos a realizar una búsqueda del mejor modelo según el hiperparámetro del grado, tomando el test anova como criterio de selección.

```r
grados <- 1:10

mejor_grado <- NULL
mejor_modelo <- NULL
mejor_anova_p <- 1

modelo_anterior <- lm(var_obj ~ poly(X11, 1, raw = TRUE), data = datos)

for (g in grados[-1]) {
  modelo_actual <- lm(var_obj ~ poly(X11, g, raw = TRUE), data = datos)

  anova_resultado <- anova(modelo_anterior, modelo_actual)
  p_valor <- anova_resultado$`Pr(>F)`[2]

  cat("Grado:", g, " ---- p-valor:", p_valor, "\n")

  if (p_valor < 0.05) {
    mejor_grado <- g
    mejor_modelo <- modelo_actual
    mejor_anova_p <- p_valor
    modelo_anterior <- modelo_actual
  } else {
    cat("No hay diferencias significativas entre grado", g, "y grado",
     ↪  g-1, "detenemos la búsqueda.\n")
    break
```

```
  }
}
```

```
Grado: 2  ---- p-valor: 0
Grado: 3  ---- p-valor: 0
Grado: 4  ---- p-valor: 0
Grado: 5  ---- p-valor: 6.826215e-171
Grado: 6  ---- p-valor: 2.452581e-59
Grado: 7  ---- p-valor: 0.0009175444
Grado: 8  ---- p-valor: 0.01738858
Grado: 9  ---- p-valor: 0.704594
No hay diferencias significativas entre grado 9 y grado 8 detenemos la
búsqueda.
```

```
cat("\nMejor grado:", mejor_grado, "con p-valor:", mejor_anova_p, "\n")
```

```
Mejor grado: 8 con p-valor: 0.01738858
```

```
summary(mejor_modelo)
```

```
Call:
lm(formula = var_obj ~ poly(X11, g, raw = TRUE), 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, g, raw = TRUE)1  2.289573   0.076981  29.742  < 2e-16 ***
poly(X11, g, raw = TRUE)2  4.039196   0.168242  24.008  < 2e-16 ***
poly(X11, g, raw = TRUE)3  1.485938   0.202233   7.348 2.93e-13 ***
poly(X11, g, raw = TRUE)4 -0.253835   0.259871  -0.977   0.3288
poly(X11, g, raw = TRUE)5  0.932037   0.165077   5.646 1.88e-08 ***
poly(X11, g, raw = TRUE)6  0.791915   0.137634   5.754 1.01e-08 ***
poly(X11, g, raw = TRUE)7  0.006936   0.042952   0.161   0.8717
poly(X11, g, raw = TRUE)8 -0.060741   0.025517  -2.380   0.0174 *
---
```

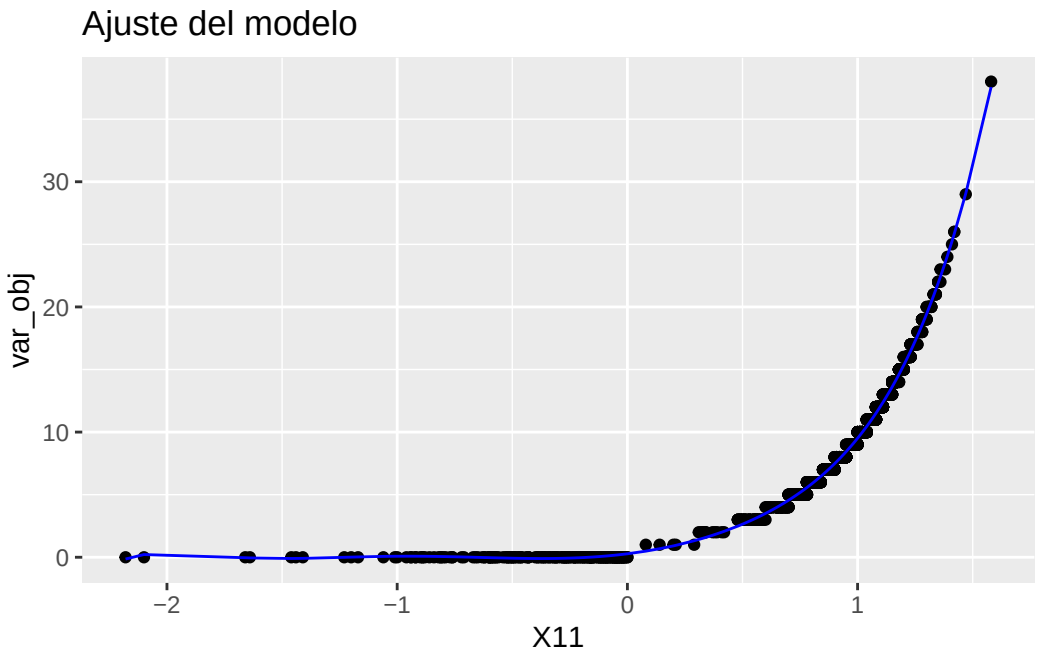
```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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
```

Por tanto, el modelo seleccionado es el de grado 8.

```r
model.poly_8 = mejor_modelo
bondad = rbind(bondad, data.frame(AIC = AIC(model.poly_8), dev = NA))
rownames(bondad)[nrow(bondad)] = "poly"
bondad
```

```
               AIC       dev
pois.full 7509.784 112.59043
pois.red  7495.794 118.60095
gam.full  7420.893  20.07102
gam.X11tp 7400.386  21.53563
gam.X11ts 7400.385  21.53674
gam.X11cr 7401.010  21.99967
poly       670.896        NA
```

```r
ggplot(data = datos, aes(x = datos$X11, y = datos$var_obj)) +
    geom_point() +
    geom_line(aes(y = model.poly_8$fitted.values), color = "blue") +
    labs(title = "Ajuste del modelo",
          x = "X11",
          y = "var_obj")
```



El ajuste es realmente bueno, y tenemos un AIC bastante bajo.

Regresión con Splines

Emplearemos por último regresión con Splines. Tomaremos nodos en el 0 y 1, para tener en cuenta los cambios en la dependencia de `var_ob` y `X11`.

```
model.spline=lm(var_obj~bs(X11,knots=c(0,1)),data=datos)
summary(model.spline)
```

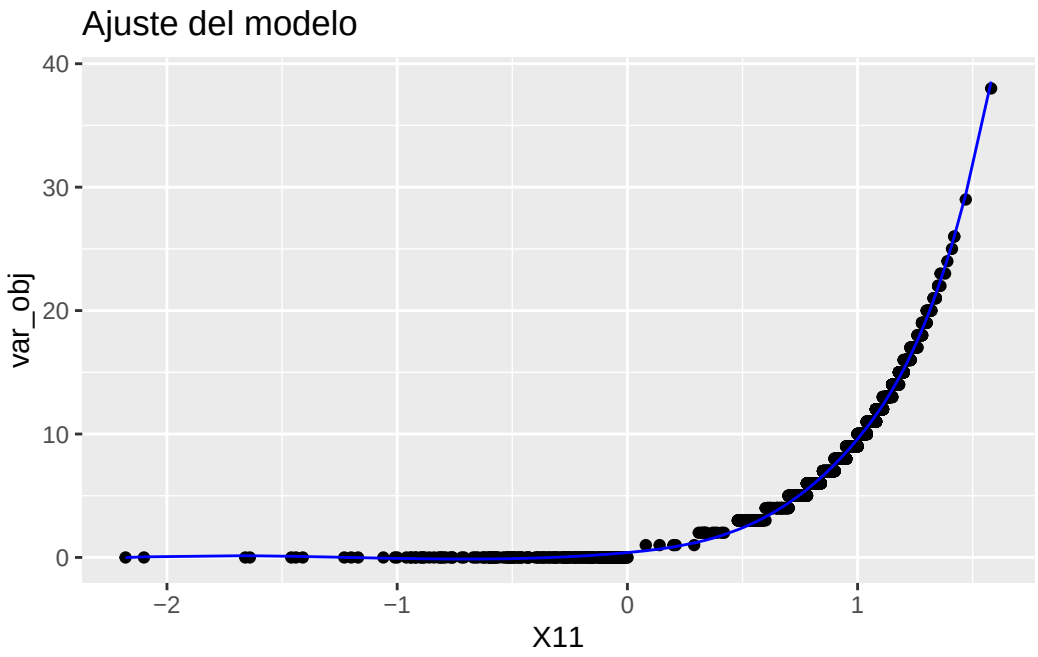
```
Call:
lm(formula = var_obj ~ bs(X11, knots = c(0, 1)), data = datos)

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
```

```r
bondad = rbind(bondad, data.frame(AIC = AIC(model.spline), dev = NA))
rownames(bondad)[nrow(bondad)] = "spline"
bondad
```

```
                AIC       dev
pois.full 7509.7837 112.59043
pois.red  7495.7942 118.60095
gam.full  7420.8926  20.07102
gam.X11tp 7400.3856  21.53563
gam.X11ts 7400.3848  21.53674
gam.X11cr 7401.0105  21.99967
poly       670.8960        NA
spline     897.5015        NA
```

```r
ggplot(data = datos, aes(x = datos$X11, y = datos$var_obj)) +
  geom_point() +
  geom_line(aes(y = model.spline$fitted.values), color = "blue") +
  labs(title = "Ajuste del modelo",
       x = "X11",
       y = "var_obj")
```



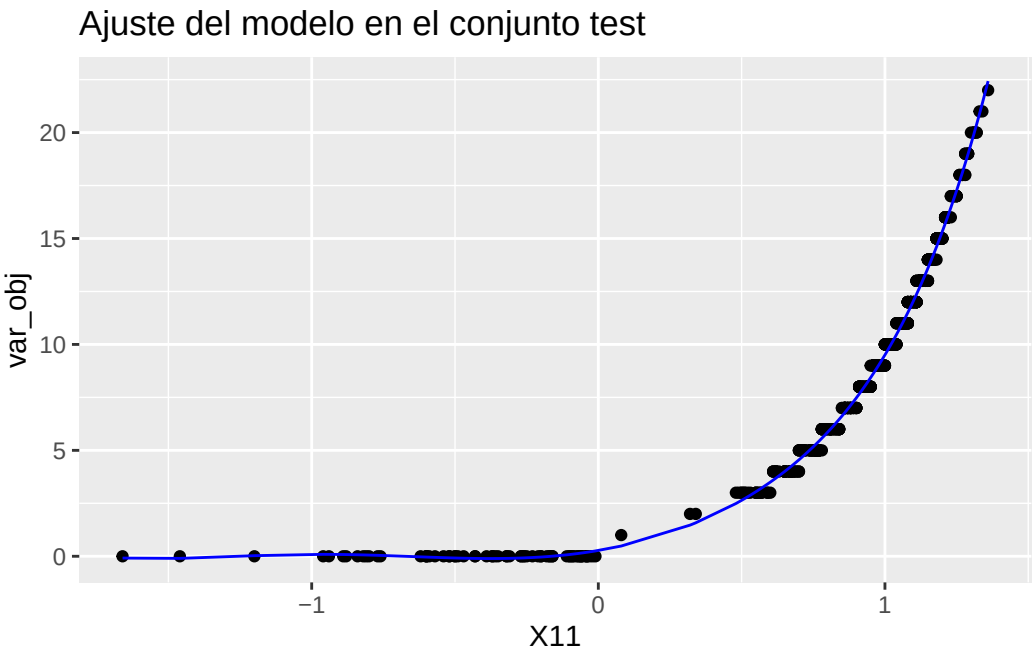
El ajuste es realmente bueno, pero se ha aumentado el AIC con respecto al modelo polinómico.

Elección del modelo final y evaluación de los resultados

El modelo que mejores resultados de bondad ha dado es la regresión polinómica de grado 8. Vamos a realizar una evaluación del ajuste real y el posible sobreajuste.

```
set.seed(12345)
ind = sample(1:nrow(datos), size = 2/3 * nrow(datos))

train = datos[ind, ]
test = datos[-ind, ]

final_model = lm(var_obj~poly(X11, 8, raw = T), data = train)

pred = predict(final_model, newdata = data.frame(X11 = test$X11))

ggplot(data = test, aes(x = X11, y = var_obj)) +
    geom_point() +
    geom_line(aes(y = pred), color = "blue") +
    labs(title = "Ajuste del modelo en el conjunto test",
```

```
    x = "X11",
    y = "var_obj")
```



Concluimos por tanto que el modelo polinomial logra una predicción realmente buena de la variable objetivo `var_obj`.