Tema 3: Modelos lineales generalizados

Componentes de un modelo generalizado lineal (GLM)

Un modelo lineal generalizado tiene tres componentes básicos:

- **Componente aleatoria**: Identifica la variable respuesta y su distribución de probabilidad.
- **Componente sistemática**: Especifica las variables explicativas (independientes o predictoras) utilizadas en la función predictora lineal.
- **Función link**: Es una función del valor esperado de Y, $E(Y)$, como una combinación lineal de las variables predictoras.

Componente aleatoria

La componente aleatoria de un GLM consiste en una variable aleatoria Y con observaciones independientes $(y_1, \ldots, y_N)$.

En muchas aplicaciones, las observaciones de Y son binarias y se identifican como *éxito* y *fracaso*. Aunque de modo más general, cada Y_i indica el número de éxitos de entre un número fijo de ensayos, y se modeliza como una distribución binomial.

En otras ocasiones cada observación es un recuento, con lo que se puede asignar a Y una distribución de Poisson o una distribución binomial negativa. Finalmente, si las observaciones son continuas se puede asumir para Y una distribución normal.

Todos estos modelos se pueden incluir dentro de la llamada familia *exponencial* de distribuciones

$$f\left(y_i|\theta_i\right) = a\left(\theta_i\right) \cdot b\left(y_i\right) \cdot \exp\left[y_i Q\left(\theta_i\right)\right],$$

de modo que $Q(\theta)$ recibe el nombre de *parámetro natural*.

Componente Sistemática

La componente sistemática de un GLM especifica las variables explicativas, que entran en forma de efectos fijos en un modelo lineal, es decir, las variables x_j se relacionan mediante

$$\alpha + \beta_1 x_1 + \cdots + \beta_k x_k$$

Esta combinación lineal de variables explicativas se denomina *predictor lineal*.

Alternativamente, se puede expresar como un vector $(\eta_1, \ldots, \eta_N)$ tal que

$$\eta_i = \sum_j \beta_j x_{ij}$$

donde x_{ij} es el valor del j-ésimo predictor en el i-ésimo individuo, e $i = 1, \ldots, N$. El término independiente α se obtendría con esta notación haciendo que todos los x_{ij} sean igual a 1 para todos los i.

En cualquier caso, se pueden considerar variables que estén basadas en otras variables como $x_3 = x_1 x_2$ ó $x_3 = x_2^2$, para modelizar interacciones entre variables o efectos curvilíneos de x_2.

Función link

Se denota el valor esperado de Y como $\mu = E(Y)$, entonces la función *link* especifica una función $g(\cdot)$ que relaciona μ con el predictor lineal como

$$g(\mu) = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k$$

Así, la función link $g(\cdot)$ relaciona las componentes aleatoria y sistemática.

De este modo, para $i = 1, \ldots, N$,

$$\mu_i = E(Y_i)$$

$$\eta_i = g(\mu_i) = \sum_j \beta_j x_{ij}$$

La función g más simple es $g(\mu) = \mu$, esto es, la identidad que da lugar al modelo de regresión lineal clásico

$$\mu = E(Y) = \alpha + \beta_1 x_1 + \cdots + \beta_k x_k$$

Los modelos de regresión lineal típicos para respuestas continuas son un caso particular de los GLM. Estos modelos generalizan la regresión ordinaria de dos modos: permitiendo que Y tenga distribuciones diferentes a la normal y, por otro lado, incluyendo distintas funciones link de la media. Esto resulta bastante útil para datos categóricos.

Los modelos GLM permiten la unificación de una amplia variedad de métodos estadísticos como la regresión, los modelos *ANOVA* y los modelos de datos categóricos. En realidad se usa el mismo algoritmo para obtener los estimadores de máxima verosimilitud en todos los casos. Este algoritmo es la base del *procedimiento* `GENMOD` de SAS y de la función `glm` de R.

Modelos lineales Generalizados para datos binarios

En muchos casos las respuestas tienen solo dos categorías del tipo `sí/no` de modo que se puede definir una variable Y que tome dos posibles valores 1 (*éxito*) y 0 (*fracaso*), es decir $Y \sim Bin\,(1, \pi)$. En este caso

$$\begin{aligned} f\,(y|\pi) &= \pi^y (1-\pi)^{1-y} \\ &= (1-\pi)\left(\frac{\pi}{1-\pi}\right)^y \\ &= (1-\pi)\exp\left[y\log\left(\frac{\pi}{1-\pi}\right)\right] \end{aligned}$$

con $y = 0, 1$.

El parámetro natural es

$$Q(\pi) = \log\left(\frac{\pi}{1-\pi}\right) = \text{logit}(\pi)$$

En este caso

$$E(Y) = P\,(Y = 1) = \pi(x)$$

dependiente de p variables explicativas $\mathbf{x} = (x_1, \ldots, x_p)$ y

$$Var(Y) = \pi(x)(1 - \pi(x))$$

En respuestas binarias, un modelo análogo al de regresión lineal es

$$\pi(x) = \alpha + \beta x$$

que se denomina modelo de probabilidad lineal, ya que la probabilidad de éxito cambia linealmente con respecto a x.

El parámetro β representa el cambio en probabilidad por unidad de x. Este modelo es un GLM con un componente aleatorio binomial y con función link igual a la identidad.

Sin embargo, este modelo tiene el problema de que aunque las probabilidades deben estar entre 0 y 1, el modelo puede predecir a veces valores $\pi(x) > 1$ y $\pi(x) < 0$.

Ejemplo

Se tiene la siguiente tabla donde se eligen varios niveles de ronquidos y se ponen en relación con una enfermedad cardíaca. Se toman como puntuaciones relativas de ronquidos los valores $\{0, 2, 4, 5\}$.

	Enfermedad Cardiaca		
Ronquido	*SI*	*NO*	*Proporción de SI*
Nunca	24	1355	0.017
Ocasional	35	603	0.055
Casi cada noche	21	192	0.099
Cada noche	30	224	0.118

```
# Fijamos los niveles de manera ordinal
roncas = c(0, 2, 4, 5)
prop.SI = c(24/(1355+24), 35/(35+603), 21/(21+192), 30/(30+224))

irls = glm(prop.SI ~ roncas)
summary(irls)$coefficients
```

Se obtiene

```
             Estimate   Std. Error  t value     Pr(>|t|)
(Intercept) 0.01631222 0.0015861606 10.28409 0.0093231257
roncas      0.02033780 0.0004729017 43.00639 0.0005402341
```

Es decir, el modelo que se obtiene es

$$\widehat{\pi} = 0{,}0163 + 0{,}0203x$$

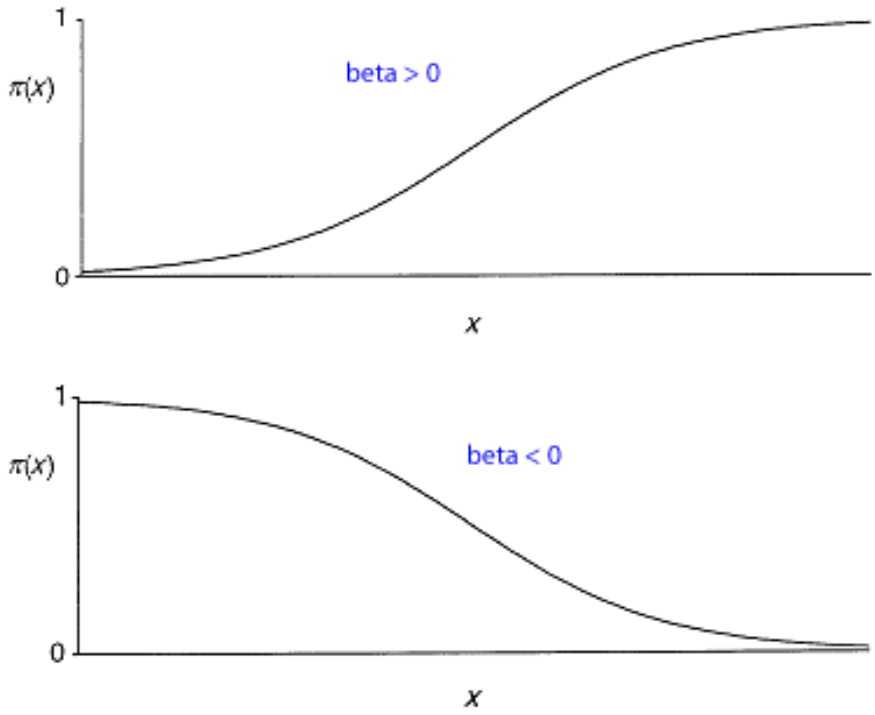
Por ejemplo, para gente que no ronca ($x = 0$) la probabilidad estimada de enfermedad cardíaca sería

$$\widehat{\pi} = 0{,}0163$$

Regresión Logística

Normalmente las relaciones entre $\pi(x)$ y x son **no lineales**, de modo que el cambio en x tiene menor impacto cuando π está cerca de 0 ó de 1 que cuando π está más cerca de la mitad del rango.

La relación habitualmente tiene forma de curva en forma sigmoidal, como puede verse en la gráfica siguiente.



La función matemática que modeliza esta forma es

$$\pi(x) = \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)},$$

que se denomina la función logística de la que se derivan los modelos de regresión logística:

$$\begin{aligned} 1 - \pi(x) &= 1 - \frac{\exp(\alpha + \beta x)}{1 + \exp(\alpha + \beta x)} = \\ & \frac{1}{1 + \exp(\alpha + \beta x)} \end{aligned}$$

por lo que

$$\begin{aligned} \frac{\pi(x)}{1 - \pi(x)} &= \exp(\alpha + \beta x) \Longrightarrow \\ \log\left(\frac{\pi(x)}{1 - \pi(x)}\right) &= \alpha + \beta x \end{aligned}$$

La función link $\log\left(\frac{\pi}{1-\pi}\right)$ de π se denomina función *logit*, de modo que así se asegura que no exista ningún problema estructural respecto al posible rango de valores de π.

El parámetro β determina el rango y la velocidad de incremento o decremento de la curva.

En el ejemplo de los niveles de ronquidos, los resultados que se obtendrían son

```
roncas = c(0, 2, 4, 5)
logit.irls <- glm(cbind(SI=c(24, 35, 21, 30), NO=c(1355,
603, 192, 224)) ~ roncas, family=binomial(link=logit))

summary(logit.irls)$coefficients
```

Se obtiene

```
             Estimate Std. Error    z value      Pr(>|z|)
(Intercept) -3.8662481 0.16621436 -23.260614 1.110885e-119
roncas       0.3973366 0.05001066   7.945039  1.941304e-15
```

De modo que

$$logit\left(\widehat{\pi}(x)\right) = -3{,}87 + 0{,}40x$$

Como $\widehat{\beta} = 0{,}40 > 0$ la probabilidad de ataque cardíaco aumenta cuando los niveles de ronquidos se incrementan.

Regresión Probit

Otro modelo en el que se pueden considerar curvas en forma de **S** son los modelos *probit*. Una idea natural es que

$$\pi(x) = F(x),$$

siendo F una función de distribución. Cuando X es una v.a. continua, la función de distribución como función de x, tiene forma de **S**. Esto sugiere una clase de modelos de dependencia para modelos binarios

Como caso particular se puede considerar el link *probit* que transforma probabilidades en valores estándar de la función de distribución normal, $F(x) = \Phi(x)$.

$$\begin{aligned} \pi(x) &= \Phi(\alpha + \beta x) \\ \Phi^{-1}(\pi(x)) &= \alpha + \beta x \end{aligned}$$

Así, se define probit $\equiv \Phi^{-1}$.

Por ejemplo,

$$\begin{aligned} probit(0{,}05) &= -1{,}645 \\ probit(0{,}975) &= 1{,}96 \end{aligned}$$

En el ejemplo anterior, sobre el sueño y la enfermedad cardíaca, se programa en R como

```
roncas = c(0, 2, 4, 5)
probit.irls = glm(cbind(SI=c(24, 35, 21, 30), NO=c(1355,
603, 192, 224)) ~ roncas, family=binomial(link=probit))

summary(probit.irls)$coefficients
```

Se obtiene

```
             Estimate Std. Error   z value      Pr(>|z|)
(Intercept) -2.0605516 0.07016673 -29.36651 1.471042e-189
roncas       0.1877705 0.02348052   7.99686  1.276323e-15
```

Es decir,

$$probit(\widehat{\pi}(x)) = -2{,}061 + 0{,}188x$$

Con un nivel de ronquido $x = 0$, el probit es igual a

$$-2{,}061 + 0{,}188(0) = -2{,}06$$

Así, para $\pi(0)$ la función de distribución (probabilidad de la cola izquierda) de la normal estándar en $z = -2{,}06$, es igual a 0,020. Esto equivale a la probabilidad de tener una enfermedad cardiaca.

Por otro lado, para un nivel $x = 5$, el probit es igual a

$$-2{,}061 + 0{,}188(5) = -1{,}12$$

que corresponde a una probabilidad de enfermedad cardíaca de 0,131

En la práctica, tanto los modelos *probit* como *logit* producen ajustes similares.

Modelos GLM para recuentos

En muchos casos las variables respuesta son recuentos, y en muchas ocasiones los recuentos aparecen al resumir tablas de contingencia.

En el modelo más simple se asume que el componente aleatorio Y sigue una distribución de Poisson. Esta distribución es unimodal y su propiedad más destacada es que la media y la varianza coinciden

$$E(Y) = Var(Y) = \mu$$

de modo que cuando el número de recuentos es mayor en media, también tienden a tener mayor variabilidad.

En el modelo GLM se usa habitualmente el logaritmo de la media para la función link, de modo que el modelo *log-lineal* con una variable explicativa X se puede expresar como

$$\log(\mu) = \alpha + \beta x$$

de modo que

$$\mu = \exp[\alpha + \beta x] = e^{\alpha}\left(e^{\beta}\right)^{x}$$

Ejemplo

Entre los cangrejos cacerola

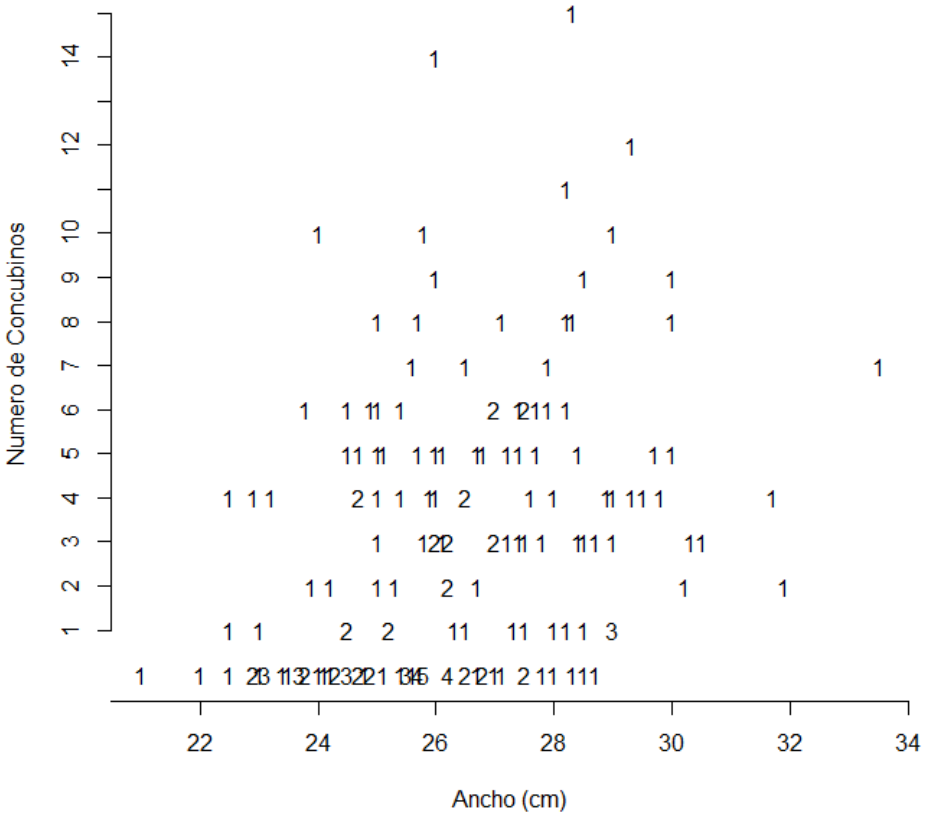
```
http://en.wikipedia.org/wiki/Horseshoe_crab
```

se sabe que cada hembra tiene un macho en su nido, pero puede tener más machos *concubinos*. Se considera que la variable respuesta es el número de concubinos y las variables explicativas son: *color*, *estado de la espina central*, *peso* y *anchura del caparazón*. En un primer análisis solo consideramos la anchura del caparazón como variable explicativa

```
tabla = read.csv("http://www.hofroe.net/stat557/data/crab.txt",
header=T, sep="\t")
dimnames(tabla)[[2]] = c("color","spine","width","satell","weight")
names(tabla)

plot.tabla = aggregate(rep(1, nrow(tabla)),
list(Sa=tabla$satell, W=tabla$width), sum)
```

```
plot(y=plot.tabla$Sa, x=plot.tabla$W, xlab="Ancho (cm)",
ylab="Numero de Concubinos", bty="L", axes=F, type="n")
axis(2, at=1:15)
axis(1, at=seq(20, 34, 2))
text(y=plot.tabla$Sa, x=plot.tabla$W, labels=plot.tabla$x)
```

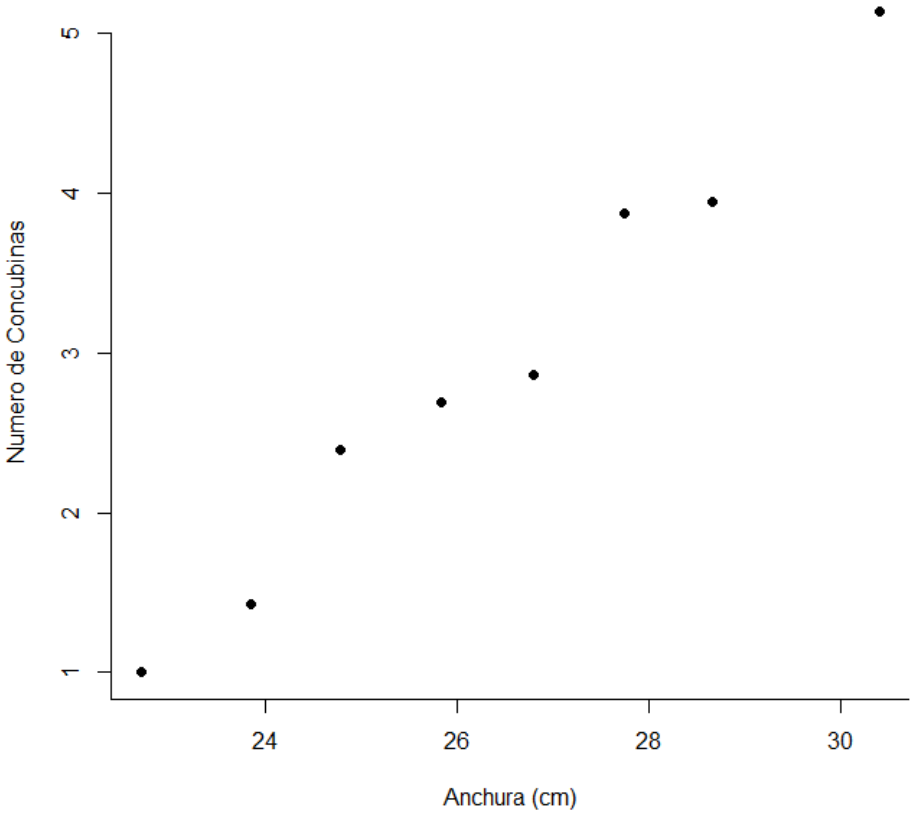



```
# Discretizamos la anchura del caparazon
tabla$W.fac = cut(tabla$width, breaks=c(0, seq(23.25, 29.25), Inf))

# Calculamos el numero medio de concubinos para cada
# categoria segun la anchura del caparazon.
plot.y = aggregate(tabla$satell, by=list(W=tabla$W.fac), mean)$x

# Determinamos la media de la anchura del caparazon por categoria
plot.x = aggregate(tabla$width, by=list(W=tabla$W.fac), mean)$x

# Representamos las medias de anchura y la media
# del numero de concubinos
plot(x=plot.x, y=plot.y, ylab="Numero de Concubinas",
xlab="Anchura (cm)", bty="L", axes=F, type="p", pch=16)
axis(2, at=0:5)
axis(1, at=seq(20, 34, 2))
```



Se puede ajustar un modelo GLM de Poisson.

```
log.fit = glm(satell ~ width, family=poisson, data=tabla)

summary(log.fit)
```

Se obtiene

```
Call:
glm(formula = satell ~ width, family = poisson(link = log), data = tabla)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8526  -1.9884  -0.4933   1.0970   4.9221

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.30476    0.54224  -6.095  1.1e-09 ***
width        0.16405    0.01997   8.216  < 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: 632.79  on 172  degrees of freedom
Residual deviance: 567.88  on 171  degrees of freedom
AIC: 927.18

Number of Fisher Scoring iterations: 6
```

En un modelo de Poisson la *deviance* se calcula como $-2\log(p(y))$.

La `Null deviance` es la desviación para el modelo que no depende de ninguna variable. La `Residual deviance` es la diferencia entre la desviación del modelo que no depende de ninguna variable menos la correspondiente al modelo que incluye a la variable `width`. La diferencia entre ambas se distribuye como una distribución chi-cuadrado con 1 grado de libertad y permite contrastar si el coeficiente de `width` puede considerarse nulo.

```
log.fit$null.deviance - log.fit$deviance
```

```
[1] 64.91309
```

```
1-pchisq(64.91309, 1)
```

```
[1] 7.771561e-16
```

Se puede rechazar claramente la hipótesis nula. Hay un aportación significativa de la anchura del caparazón.

Se pueden ver los atributos del objeto de clase `glm`, y obtener cada uno de ellos.

```
attributes(log.fit)
```

```
$names
"coefficients" "residuals" "fitted.values" "effects"
"R" "rank" "qr" "family" "linear.predictors"
"deviance" "aic" "null.deviance" "iter" "weights"
"prior.weights" "df.residual" "df.null" "y" "converged"
"boundary" "model" "call" "formula" "terms" "data"
"offset" "control" "method" "contrasts" "xlevels"

$class
[1] "glm" "lm"
```

```
# los valores esperados vienen dados por
log.fit$fitted.values
# o
fitted(log.fit)
```

```
       1        2        3        4        5 ...
3.810341 1.471463 2.612781 2.145904 2.612781 ...
```

```
# Para obtener los residuos:
log.fit$residuals
# o
resid(log.fit)
# o
residuals(log.fit)
```

```
        1           2           3           4           5  ...
 1.86768502 -1.71549591  3.08028134 -2.07166794  0.79535899 ...
```

```
# Para obtener los coeficientes
log.fit$coefficients
# o
coef(log.fit)
# o
coefficients(log.fit)
```

```
(Intercept)       width
 -3.3047572   0.1640451
```

```
confint(log.fit)      # Intervalos de confianza
```

```
                2.5 %     97.5 %
(Intercept) -4.3662326 -2.2406858
width        0.1247244  0.2029871
```

Se puede predecir la media de la respuesta para el valor de `width` que queramos con `predict`. Por ejemplo, para una anchura igual a 26.3:

```
predict.glm(log.fit, type="response", newdata=data.frame(width=26.3))
```

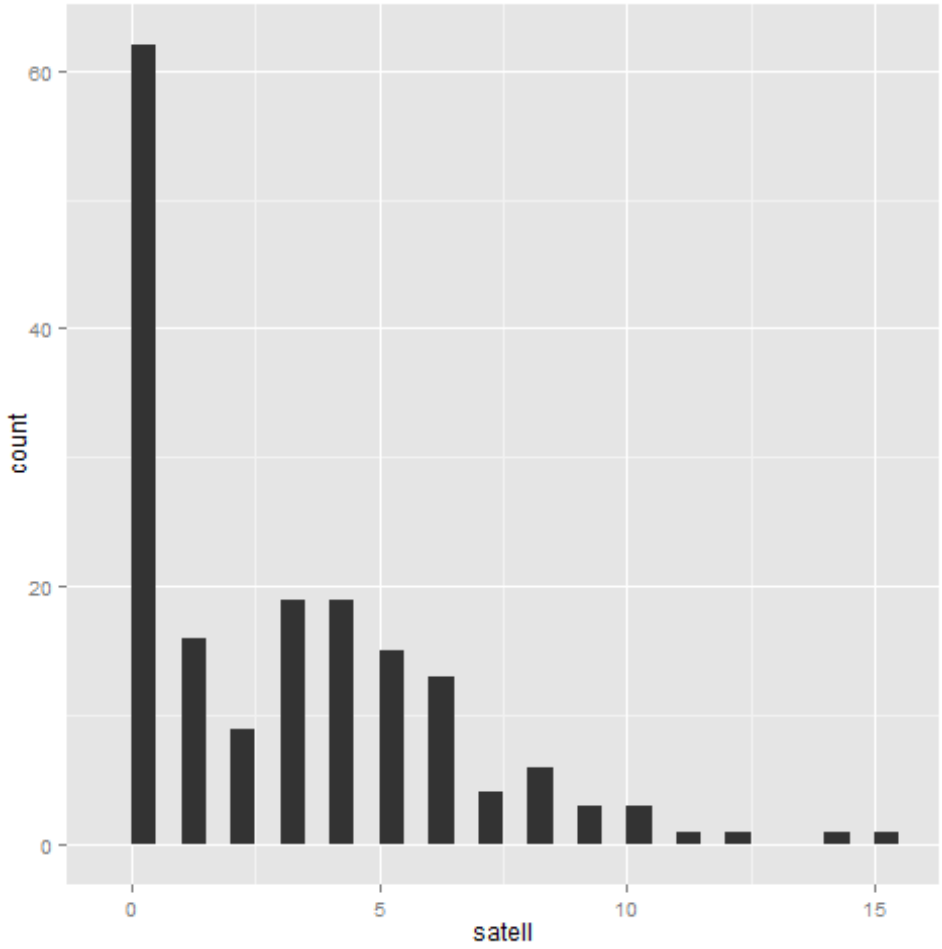
```
[1] 2.744581
```

Sobredispersión en GLM Poisson

En una distribución de Poisson, la media y la varianza son iguales, pero cuando trabajamos con recuentos reales no suele ser cierta esta hipótesis.

Con frecuencia la varianza es mayor que la media. A esto se le llama **sobredispersión** (*over-dispersed*).

```
library(ggplot2)
ggplot(tabla, aes(satell)) + geom_histogram()
```



Habitualmente esta situación se debe a la existencia de heterogeneidad entre las observaciones. Esto se puede interpretar como una mezcla o mixtura de distribuciones de Poisson. No es un problema cuando Y tiene una distribución normal porque la normal tiene un parámetro específico que modeliza la variabilidad.

```r
summary.glm(log.fit.over)$dispersion

log.fit.over = glm(satell ~ width,
family=quasipoisson(link=log), data=tabla)

confint(log.fit.over)

# Se puede comparar con los intervalos del modelo de Poisson
confint(log.fit)
```

```
# Intervalos del modelo de QuasiPoisson
                2.5 %     97.5 %
(Intercept) -4.3662326 -2.2406858
width        0.1247244  0.2029871

# dispersion
3.182205

# Intervalos originales del modelo de Poisson
                 2.5 %     97.5 %
(Intercept) -5.19648768 -1.4047821
width        0.09363759  0.2332483
```

Aquí la varianza es algo más de tres veces la media. La estimación del parámetro de dispersión no es más que la suma de los residuos dividida entre sus grados de libertad.

GLM binomiales negativos

Si una v.a. Y se distribuye como una binomial negativa, entonces la función de probabilidad es

$$P\left(y|k,\mu\right) = \frac{\Gamma(y+k)}{\Gamma(k)\Gamma(y+1)}\left(\frac{k}{\mu+k}\right)^{k}\left(1-\frac{k}{\mu+k}\right)^{y}$$

con $y = 0, 1, 2, \ldots$ donde k y μ son los parámetros de la distribución.

Se tiene que

$$\begin{aligned} E(Y) &= \mu \\ Var(Y) &= \mu + \frac{\mu^2}{k} \end{aligned}$$

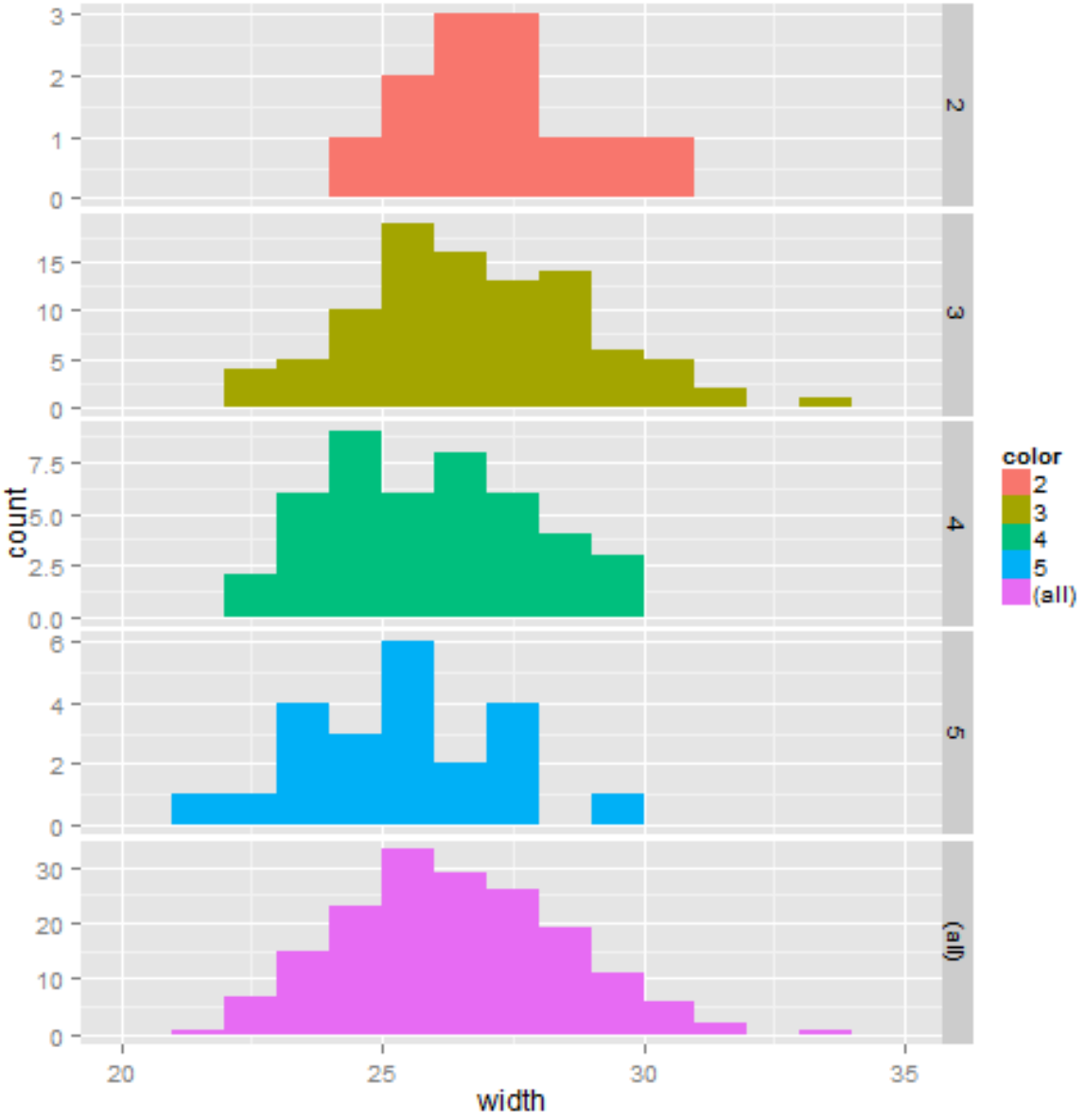
El parámetro $\frac{1}{k}$ es un parámetro de dispersión, de modo que si $\frac{1}{k} \to 0$ entonces $Var(Y) \to \mu$ y la distribución binomial negativa converge a una distribución de Poisson.

Por otro lado, para un valor fijo de k esta distribución pertenece a la familia exponencial natural, de modo que se puede definir un modelo GLM binomial negativo. En general, se usa una función link de tipo logaritmo.

La regresión binomial negativa se puede utilizar para datos sobredispersos de recuentos, es decir cuando la varianza condicional es mayor que la media condicional. Se puede considerar como una generalización de la regresión de Poisson, ya que tiene su misma estructura de medias y además un parámetro adicional para el modelo de sobredispersión. Si la distribución condicional de la variable observada es más dispersa, los intervalos de confianza para la regresión binomial negativa es probable que sean más estrechos que los correspondientes a un modelo de regresión de Poisson.

```
require(ggplot2)
require(MASS)

png(file="NegBinom.png",pointsize=8)
ggplot(tabla, aes(width, fill=color)) + geom_histogram(binwidth=1) +
  facet_grid(color ~ ., margins=TRUE, scales="free")
dev.off()
```


count
width
color
2
3
4
5
(all)
2
3
4
5
(all)
0
1
2
3
0
5
10
15
0.0
2.5
5.0
7.5
0
2
4
6
0
10
20
30
20
25
30
35

```r
summary(m1 = glm.nb(satell ~ width, data=tabla))
```

```
Call:
glm.nb(formula = satell ~ width, data = tabla, init.theta = 0.90456808,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7798  -1.4110  -0.2502   0.4770   2.0177

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.05251    1.17143  -3.459 0.000541 ***
width        0.19207    0.04406   4.360  1.3e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.9046) family taken to be 1)

    Null deviance: 213.05  on 172  degrees of freedom
Residual deviance: 195.81  on 171  degrees of freedom
AIC: 757.29

Number of Fisher Scoring iterations: 1

              Theta:  0.905
          Std. Err.:  0.161

 2 x log-likelihood:  -751.291
```

```r
(est <- cbind(Estimate = coef(m1), confint(m1)))
```

```
              Estimate      2.5 %     97.5 %
(Intercept) -4.0525101 -6.5631403 -1.6033251
width        0.1920732  0.1001437  0.2869604
```

```r
exp(est)
```

```
             Estimate       2.5 %    97.5 %
(Intercept) 0.0173787 0.001411446 0.2012263
width       1.2117592 1.105329741 1.3323714
```

De modo que

$$\log\left(\widehat{\mu}\right) = -4{,}05 + 0{,}192x$$

Ejemplo de UCLA con SAS

Referencia completa:

```
http://www.ats.ucla.edu/stat/sas/dae/poissonreg.htm
```

Se toman los datos de

```
https://stats.idre.ucla.edu/wp-content/uploads/2016/02/poisson_sim.sas7bdat
```

En el ejemplo, `num_awards` es una variable resultado que indica el número de premios que ganan los estudiantes de secundaria en un año, `math` es una variable continua que representa las notas de los estudiantes en matemáticas y `prog` es una variable categórica con tres niveles que indican el nivel del programa. Se codifica como **1** = *General*, **2** = *Academic* y **3** = *Vocational*.

```
OPTIONS nodate ls=75;
/* Programa para SAS University */
/* ODS listing file='/folders/myfolders/sale.lst'; */
TITLE 'Ejemplo Premios Estudiantes';
LIBNAME eso "/folders/myfolders";

/* El fichero 'poisson_sim.sas7bdat' tiene que estar situado
en el directorio c:\folders\myfolders */

DATA eso.poisson_sim;
SET eso.poisson_sim;
RUN;

PROC means data=eso.poisson_sim n mean var min max;
  var num_awards math;
RUN;

PROC freq data=eso.poisson_sim;
tables num_awards / plots=freqplot;
run;

PROC means data=eso.poisson_sim mean var;
  class prog;
  var num_awards;
RUN;
```

```
PROC freq data=eso.poisson_sim;
  tables prog;
RUN;

PROC genmod data=eso.poisson_sim;
  class prog / param=glm;
  model num_awards = prog math / type3 dist=poisson;
RUN;
/* ODS listing close; */
```

Ejemplo Premios Estudiantes

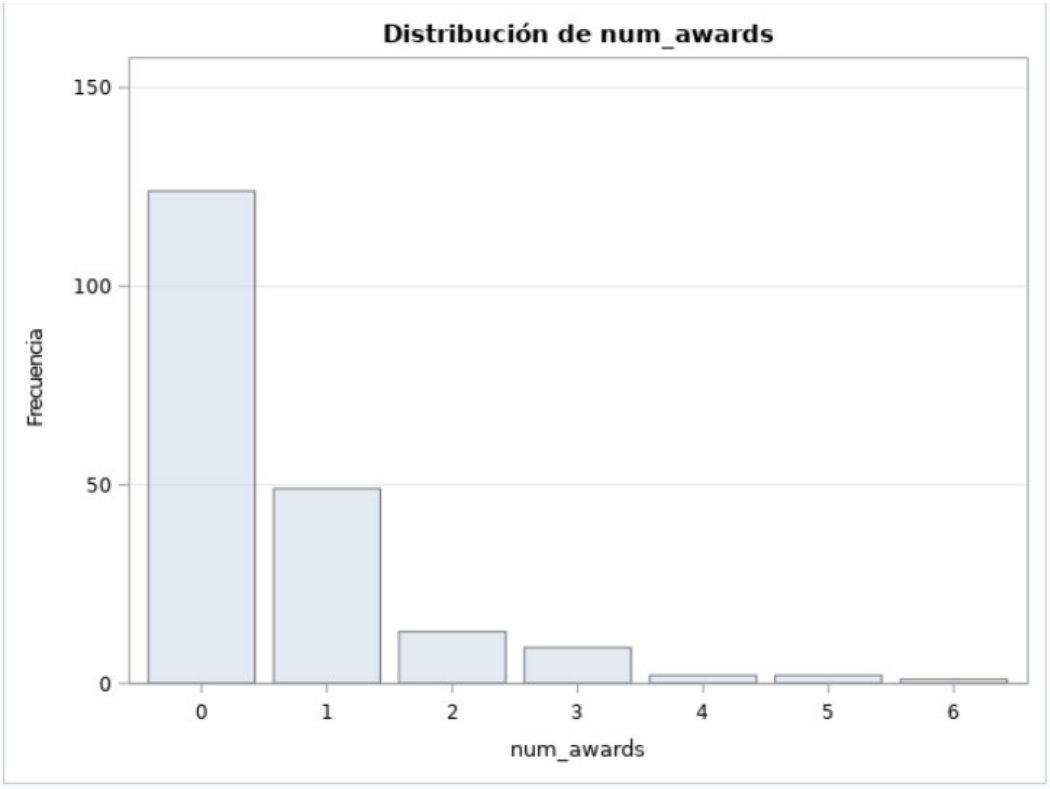
Procedimiento MEANS

Variable	Etiqueta	N	Media	Varianza	Mínimo	Máximo
num_awards		200	0.6300000	1.1086432	0	6.0000000
math	math score	200	52.6450000	87.7678141	33.0000000	75.0000000

Ejemplo Premios Estudiantes

Procedimiento FREQ

num_awards	Frecuencia	Porcentaje	Frecuencia acumulada	Porcentaje acumulado
0	124	62.00	124	62.00
1	49	24.50	173	86.50
2	13	6.50	186	93.00
3	9	4.50	195	97.50
4	2	1.00	197	98.50
5	2	1.00	199	99.50
6	1	0.50	200	100.00


Distribución de num_awards
150
100
50
0
Frecuencia
0
1
2
3
4
5
6
num_awards

Ejemplo Premios Estudiantes

Procedimiento MEANS

Analysis Variable : num_awards			
type of program	N Obs	Media	Varianza
1	45	0.2000000	0.1636364
2	105	1.0000000	1.6346154
3	50	0.2400000	0.2677551

Ejemplo Premios Estudiantes

Procedimiento GENMOD

Información del modelo	
Conjunto de datos	ESO.POISSON_SIM
Distribución	Poisson
Función de vínculo	Log
Variable dependiente	num_awards

N.º observaciones leídas	200
N.º observaciones usadas	200

Información del nivel de clase		
Clase	Niveles	Valores
prog	3	1 2 3

Criterio para evaluar bondad de ajuste			
Criterio	DF	Valor	Valor/DF
Desviación	196	189.4496	0.9666
Desviación escalada	196	189.4496	0.9666
Chi-cuadrado de Pearson	196	212.1437	1.0824
Pearson X2 escalado	196	212.1437	1.0824
Verosimilitud log		-135.1052	
Verosimilitud log completa		-182.7523	
AIC (mejor más pequeño)		373.5045	
AICC (mejor más pequeño)		373.7096	
BIC (mejor más pequeño)		386.6978	

Análisis de estimadores de parámetro de verosimilitud máxima							
Parámetro		DF	Estimación	Error estándar	Límites de confianza de Wald al 95%		Chi-cuadrado de Wald	Pr > ChiSq
Intercept		1	-4.8773	0.6282	-6.1085	-3.6461	60.28	<.0001
prog	1	1	-0.3698	0.4411	-1.2343	0.4947	0.70	0.4018
prog	2	1	0.7140	0.3200	0.0868	1.3413	4.98	0.0257
prog	3	0	0.0000	0.0000	0.0000	0.0000	.	.
math		1	0.0702	0.0106	0.0494	0.0909	43.81	<.0001
Escala		0	1.0000	0.0000	1.0000	1.0000		

Note: The scale parameter was held fixed.

Estadísticos LR para análisis de tipo 3			
Origen	DF	Chi-cuadrado	Pr > ChiSq
prog	2	14.57	0.0007
math	1	45.01	<.0001

Para SAS estándar:

```
OPTIONS nodate ls=75 formchar='|----|+|---+=|-/\<>*';
/* Fijo el directorio de trabajo */
x 'cd "c:\DondeSea"';
LIBNAME eso 'c:\DondeSea';
/* El fichero 'poisson_sim.sas7bdat' tiene que estar situado
   en el directorio c:\DondeSea */
/* Se graban los resultados en un fichero rtf o en uno pdf */
/* ODS pdf file='cosa.pdf' style=minimal startpage=no; */
   ODS rtf file='cosa.rtf' style=minimal startpage=no;

DATA eso.poisson_sim;
SET eso.poisson_sim;
RUN;

PROC means data=eso.poisson_sim n mean var min max;
  var num_awards math;
RUN;

PROC freq data=eso.poisson_sim;
tables num_awards / plots=freqplot;
RUN;

PROC means data=eso.poisson_sim mean var;
  class prog;
  var num_awards;
RUN;

PROC freq data=eso.poisson_sim;
  tables prog;
RUN;
```

```
PROC genmod data=eso.poisson_sim;
  class prog / param=glm;
  model num_awards = prog math / type3 dist=poisson;
RUN;

ODS rtf close;
/* ODS pdf close; */
```

Se obtiene la información básica del modelo y los estadísticos de *goodness-of-fit* como `log likelihood`, `AIC`, y `BIC`. Después, se obtienen los coeficientes de la regresión de Poisson de las variables y sus estadísticos. El coeficiente para `math` is 0.07, esto significa que el incremento esperado en logaritmos del recuento por cada unidad de incremento en `math` es de 0.07.

Para la variable categórica `prog` se muestran los coeficientes que relacionan los niveles 1 y 2 respecto al nivel 3. La variable `prog(2)` es la diferencia esperada en el logaritmo de los recuentos entre el grupo 2 (`prog=2`) y el grupo de referencia (`prog=3`). Así, el logaritmo esperado del recuento para el nivel 2 de `prog` es 0.714 mayor que el esperado para el nivel 3 de `prog`. De modo similar se obtiene que el logaritmo esperado del recuento para el nivel 1 de `prog` es 0.3698 menor que el esperado para el nivel 3 de `prog`.

En la tabla `Type 3` se muestra que prog en conjuto es significativo. Se contrastan las hipótesis de que son cero los estimadores: (nivel 1 vs. nivel 3 y nivel 2 vs. nivel 3).

Ejemplo sobre cáncer con SAS

Se considera un estudio de 400 pacientes con *melanoma* maligno. Se considera como variables de interés dónde aparece el tumor y el tipo celular del mismo. Se asume que los recuentos del tumor se distribuyen como una Poisson y se trata de comprobar si están influidos por el lugar de aparición y el tipo de tumor. Se tiene la siguiente tabla:

	Zona Tumor			
Tipo Tumor	*Cabeza-cuello*	*Tronco*	*Extremidades*	Total
Melanoma de Hutchinson	22	2	10	34
Melanoma superficial	16	54	115	185
Nodulos	19	33	73	125
Indeterminado	11	17	28	56
Total	68	106	226	400

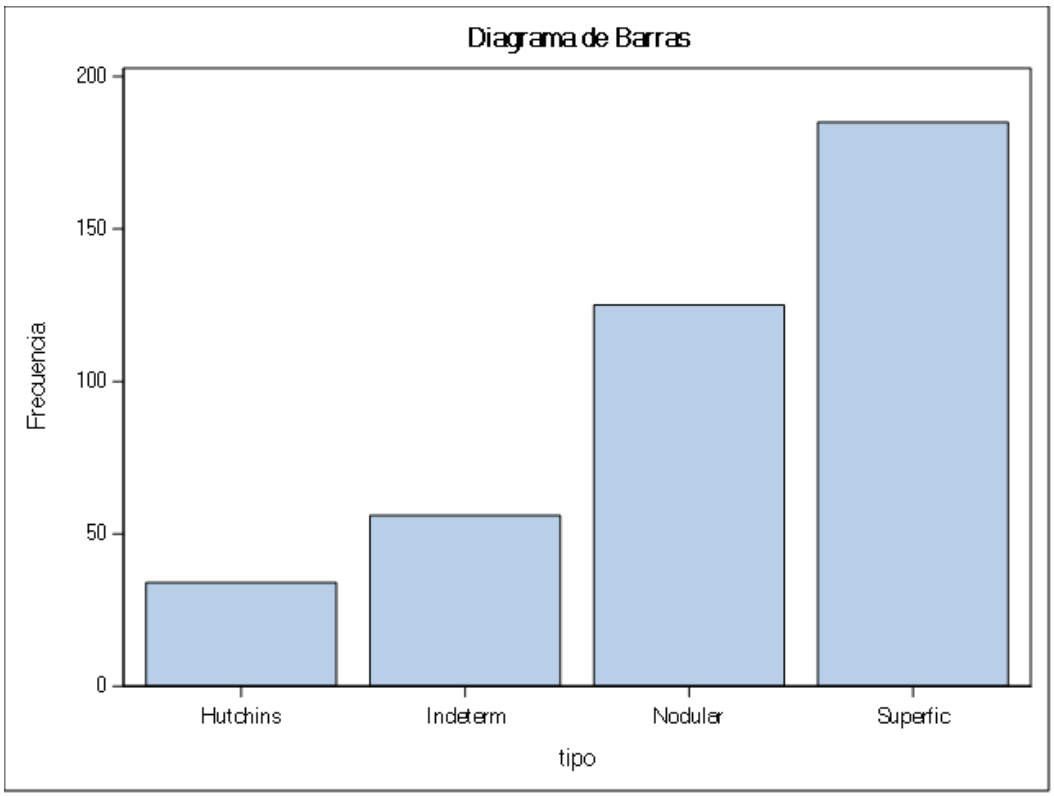
El programa en SAS de regresión de Poisson es

```
OPTIONS nodate ls=75;
/* Programa para SAS University */
/* ODS listing file='/folders/myfolders/sale.lst'; */
DATA melanomas;
INPUT tipo $ sitio $ recuento;

DATALINES;
Hutchinson Cabeza&Cuello 22
Hutchinson Tronco 2
Hutchinson Extremidades 10
Superficial Cabeza&Cuello 16
Superficial Tronco 54
Superficial Extremidades 115
Nodular Cabeza&Cuello 19
Nodular Tronco 33
Nodular Extremidades 73
Indeterminado Cabeza&Cuello 11
Indeterminado Tronco 17
Indeterminado Extremidades 28
;
RUN;
```

```
PROC SGPLOT DATA=melanomas;
TITLE 'Diagrama de Barras';
      VBAR tipo / freq=recuento;
RUN;

PROC genmod data=melanomas;
  class tipo sitio / param=glm;
  model recuento=tipo sitio / type3 dist=poisson;
RUN;
/* ODS listing close; */
```



Información del nivel de clase		
Clase	Niveles	Valores
tipo	4	Hutchins Indeterm Nodular Superfic
sitio	3	Cabeza&C Extremid Tronco

Criterio para evaluar bondad de ajuste			
Criterio	DF	Valor	Valor/DF
Desviación	6	51.7950	8.6325
Desviación escalada	6	51.7950	8.6325
Chi-cuadrado de Pearson	6	65.8129	10.9688
Pearson X2 escalado	6	65.8129	10.9688
Verosimilitud log		1124.3272	
Verosimilitud log completa		-55.4532	
AIC (mejor más pequeño)		122.9064	
AICC (mejor más pequeño)		139.7064	
BIC (mejor más pequeño)		125.8159	

Algoritmo convergido.

Análisis de estimadores de parámetro de verosimilitud máxima							
Parámetro		DF	Estimación	Error estándar	Límites de confianza de Wald al 95%		Chi-cuadrado de Wald	Pr > ChiSq
Intercept		1	3.8923	0.1111	3.6746	4.1100	1227.80	<.0001
tipo	Hutchins	1	-1.6940	0.1866	-2.0597	-1.3283	82.42	<.0001
tipo	Indeterm	1	-1.1950	0.1525	-1.4939	-0.8961	61.39	<.0001
tipo	Nodular	1	-0.3920	0.1158	-0.6190	-0.1651	11.47	0.0007
tipo	Superfic	0	0.0000	0.0000	0.0000	0.0000	.	.
sitio	Cabeza&C	1	-0.4439	0.1554	-0.7485	-0.1394	8.16	0.0043
sitio	Extremid	1	0.7571	0.1177	0.5264	0.9878	41.36	<.0001
sitio	Tronco	0	0.0000	0.0000	0.0000	0.0000	.	.
Escala		0	1.0000	0.0000	1.0000	1.0000		

Note: The scale parameter was held fixed.

Estadísticos LR para análisis de tipo 3			
Origen	DF	Chi-cuadrado	Pr > ChiSq
tipo	3	145.11	<.0001
sitio	2	98.30	<.0001

Para SAS estándar

```
OPTIONS nodate ls=75 formchar='|----|+|---+=|-/\<>*';
/* Fijo el directorio de trabajo */
x 'cd "c:\DondeSea"';
/* Se graban los resultados en un fichero rtf o en uno pdf */
/* ODS pdf file='cosa.pdf' style=minimal startpage=no; */
   ODS rtf file='cosa.rtf' style=minimal startpage=no;

DATA melanomas;
INPUT tipo $ sitio $ recuento;

DATALINES;
Hutchinson Cabeza&Cuello 22
Hutchinson Tronco 2
Hutchinson Extremidades 10
Superficial Cabeza&Cuello 16
Superficial Tronco 54
Superficial Extremidades 115
Nodular Cabeza&Cuello 19
Nodular Tronco 33
Nodular Extremidades 73
Indeterminado Cabeza&Cuello 11
Indeterminado Tronco 17
Indeterminado Extremidades 28
;
RUN;

PROC GCHART DATA=melanomas;
      VBAR tipo / sumvar=recuento descending;
RUN;

PROC genmod data=melanomas;
  class tipo sitio / param=glm;
  model recuento=tipo sitio / type3 dist=poisson;
RUN;

ODS rtf close;
/* ODS pdf close; */
```