Regresión Poisson

Laura Zapata Prada

Teoria

Para el modelo de regresión poisson la variable respuesta "Y" es un conteo; es decir, se cuentan cuantas unidades experimentales tienen o no tienen una característica de interés de un total.

Ejemplo: Número de defectuosos por unidad de empaque.

El Lambda representaría el número promedio de unidades defectuosas por cada unidad de empaque bajo las condiciones de operación (x1,x2…xk)

Con el lambda estimado puedo calcular el número de unidades defectuosas por cantidad empacadas que yo espero observar en el siguiente empaque que produciré.

$$\log(\lambda) = B0 + B1X1 + B2X2 + BkXk$$

Modelo de Regresión Poisson

Donde λ y los Betas son estimados.

Si todo se queda constante y solo incremento unitariamente la variable X1, implicaría que el log(λ) aumentaría B1 unidades; y si el log(λ) aumenta, se espera que λ también sea incrementado.

Para saber el incremento esperado en las unidades defectuosas por unidades de empaque solo incrementando la variable X1, debería calcular exp(B1) y eso sería el aumento en el λ.

$$\lambda = e^{B0+B1X1+B2X2+BkXk}$$

Donde λ y los Betas son estimados.

Ejercicio en RStudio

Lectura de datos

Iniciamos con la lectura de datos, la cual será guardada con el nombre de "datos" y en este caso es extraído de una ruta en mi ordenador personal.

```
datos<- read.table("C:/Users/acer/Downloads/poissondata.txt",header =
TRUE)
```

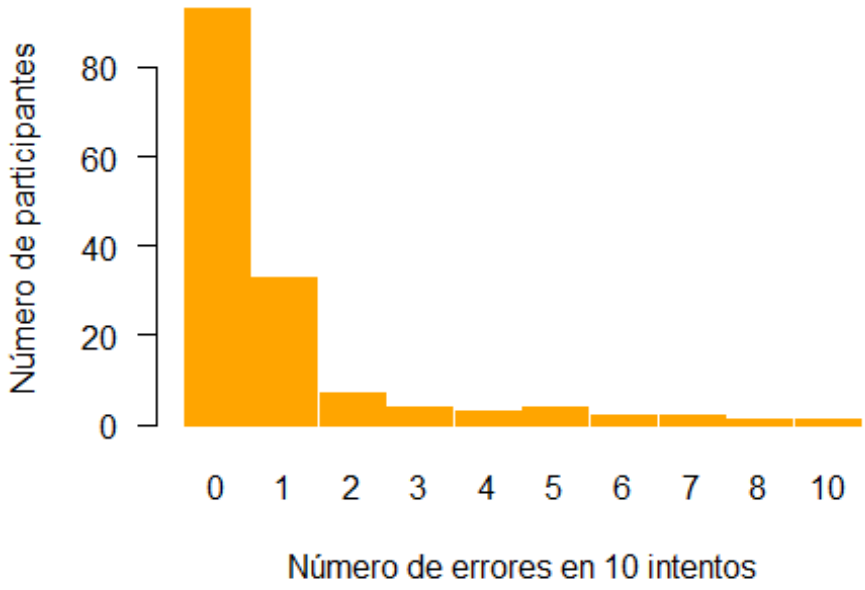
En este ejemplo tendríamos que "Y" es el número de errores en 10 intentos de un test. La información recolectada es de 150 personas.

Análisis inicial de datos

```
#Podemos ver la tabla de frecuencia de los errores.
table(datos$y)

##
##  0  1  2  3  4  5  6  7  8 10
## 93 33  7  4  3  4  2  2  1  1

#A partir de un barplot se puede apreciar visualmente
par(mfrow  = c(1,1))
with(datos, barplot(table(y), las = 1, space = .05,
                col = 'orange', border = 'orange',
                xlab = 'Número de errores en 10 intentos',
                ylab = 'Número de participantes'))
```




```
mean(datos$y)

## [1] 0.9
```

Con este grafico se puede ver la distribución del número de errores en 10 intentos que cometen 150 personas; además, en general se podría afirmar que en promedio las 150 personas tuvieron un error.

Modelo de Regresión Poisson

La función clave es "glm", y debemos seleccionar family = poisson, para que se entienda que calculara la relación con el log(λ).

```
#Creación del modelo
modelo <- glm(y ~ ., data = datos, family = poisson())

# coeficientes estimados
summary(modelo)

##
## Call:
## glm(formula = y ~ ., family = poisson(), data = datos)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4277  -0.7602  -0.3576   0.2688   2.1429
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.6283     0.1542  -4.074 4.62e-05 ***
## x1            0.9460     0.1037   9.125  < 2e-16 ***
## x2           -1.3260     0.1425  -9.304  < 2e-16 ***
## x3male       -0.8869     0.1962  -4.521 6.15e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 348.68  on 149  degrees of freedom
## Residual deviance: 118.44  on 146  degrees of freedom
## AIC: 269.45
##
## Number of Fisher Scoring iterations: 5
```

De acuerdo con el summary se aprecia que x3 es una variable categórica, por lo que RStudio elige el primer nivel de manera alfabética, como female va primero que male, se organiza el modelo en referencia a female.

Al observar que todas las variables tienen un valor p menor a un alpha de 0.05, por lo que son significativas.

De acuerdo con lo obtenido el modelo sería el siguiente:

$log(\lambda)=-0.6283+0.9460x1-1.3260x2-0.8869x3male$

Tener en cuenta que si la persona que hace la tarea es un hombre x3male será igual a 1; en caso de que sea una mujer, x3male será igual a cero, por lo que su coeficiente se eliminaría.

- Al observar los coeficientes (“Estimate”), es posible comprender que, si dejo fijas las variables x2 y x3male, el aumento únicamente en la variable x1, el logit (log(λ)) aumenta en 0.9460.
- Si dejo fijas las variables x1 y x3male, el aumento únicamente en la variable x2, el logit (log(λ)) disminuye en 1.3260.
- Si dejo fijas las variables x1 y x2, y dejo que un hombre haga el test, es decir, x3male=1, el logit (log(λ)) disminuye en 0.8869.

Para ver el cambio directo en λ, debo exponenciar los betas.

```
exp(coefficients(modelo)[-1])
```

```
##        x1        x2    x3male
## 2.5753775 0.2655449 0.4119121
```

- Si se aumenta x1 en una unidad, el número de errores (λ) aumenta en 2.57
- Si se aumenta x2 en una unidad, el número de errores (λ) disminuye y en este caso al ser el resultado menor que uno se recomienda hacer el inverso para tener el análisis más claro.

```
#Para X2
1/0.2655449
```

```
## [1] 3.765841
```

```
#Para x3male
1/0.4119121
```

```
## [1] 2.427702
```

- Si aumentas en una unidad x2, el número de errores (λ) va a disminuir 3.76.
- Si en x3male=1, se tiene que la persona que hace el test es un hombre, el número de errores (λ) va a disminuir 2.42 comparado con las mujeres.

Verificando Sobredispersión

```
## check availability of AER package
if(!require('AER')) install.packages('AER')
```

```
## Loading required package: AER
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival

require(AER)

## testing for overdispersion
dispersiontest(modelo, trafo = 1)

##
##  Overdispersion test
##
## data:  modelo
## z = -0.90813, p-value = 0.8181
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##       alpha
## -0.07419873
```

Recordar: Queremos NO rechazar H0.

En este caso se desea no rechazar la hipótesis nula, debido a que si se rechazara indicaría que existe sobredispersión.

Al obtener un valor p de 0.81 mayor a un alpha de 0.05, no se rechaza H0, por lo que no existe sobredispersión.

Verificando Importancia de Covariables

```
## verificar disponibilidad de lmtest
if(!require(lmtest)) install.packages('lmtest')
require(lmtest)

## comparación de modelos usando la LRT
## ajustamos un modelo simple
nullmodel <- glm(y ~ 1, data = datos, family = poisson())

## ahora comparamos el modelo simple
## con el modelo propuesto usando una LRT
lrtest(nullmodel, modelo)

## Likelihood ratio test
##
## Model 1: y ~ 1
## Model 2: y ~ x1 + x2 + x3
##   #Df  LogLik Df  Chisq Pr(>Chisq)
```

```
## 1   1 -245.84
## 2   4 -130.73  3 230.24  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Recordar: Queremos que se rechace H0.

Las covariables en este ejercicio serian x1, x2 y x3male.

En este caso se obtiene un valor p de 2.2e-16, lo cual es menor a un alpha de 0.05; y esto indicaría que se rechaza la H0, lo cual significa que el modelo que estamos proponiendo (que incluye covariables), es mucho mejor que no tener covariables en el modelo para explicar el número de errores que comete una persona en 10 intentos.

Predicción de modelo

```
## probabilidades estimadas
lambdahat <- predict(modelo, type = 'response')

## ahora incluyamos lambdahat a la base de datos
datos <- data.frame(id = 1:NROW(datos), datos, lambdahat)
```

Se realiza el cálculo de Lambdahat, el cual representaría el número promedio de errores por cada 10 intentos bajo las condiciones de operación (x1,x2…xk) de cada sujeto.

```
#Predicción
# que pasa para las coordenadas (2, -1), (-2, 4) y sexo male?
predict(modelo,
        newdata = data.frame(x1 = c(2, -2),
                             x2 = c(-1, 4),
                             x3 = 'male'),
        type = 'response', se.fit = TRUE)

## $fit
##            1            2
## 5.4890566150 0.0001647494
##
## $se.fit
##           1           2
## 0.827494281 0.000132288
##
## $residual.scale
## [1] 1
```

Ahora se han realizado dos predicciones:

En la primera prediccion con (x1=2, X2=-1 y x3=male), se espera que el lambda sea de 5.489

Y la segunda predicción con (x1=-2, X2=4 y x3=male), se espera que el lambda sea de 0.00016

Número de errores en 10 intentos para 10000 hombres

Para conocer el número de errores, en vez del número promedio (λ) se puede realizar una simulación.

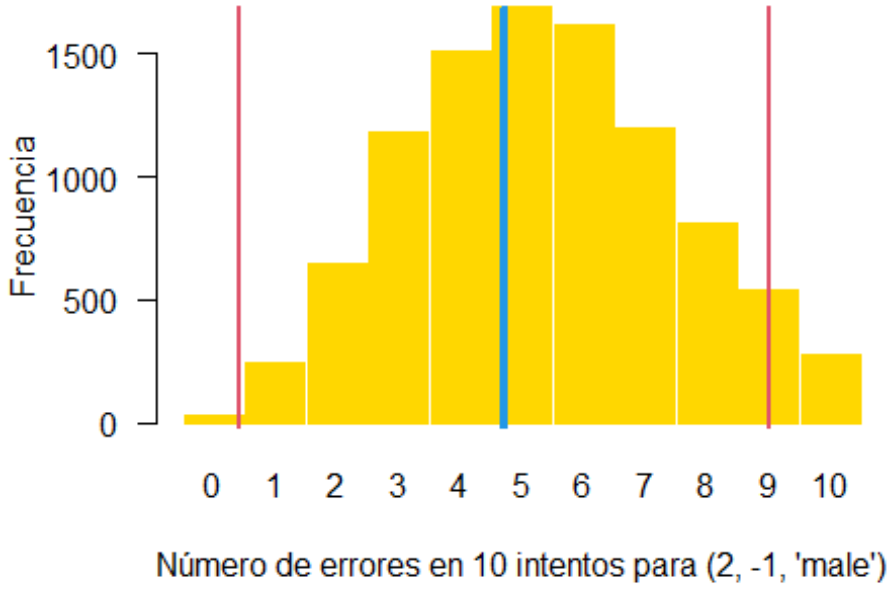
```
#Cuando el punto se proyecta en (2, -1)
lambdahat <- predict(modelo, newdata =
                       data.frame(x1 = 2,
                                  x2 = -1,
                                  x3 = 'male'),
                     type = 'response')
set.seed(3)
yhat <- rpois(10000, lambdahat)
yhat <- yhat[yhat < 11]
#Se pone menor a 11, porque máximo puede tener 10 errores (ya que el test
solo tiene 10 puntos)
(ci <- quantile(yhat, probs = c(0.025, 0.975)))

##  2.5% 97.5%
##     1    10
```

Lo que concluimos es que esa persona, con esas condiciones (x1 = 2, x2 = -1, x3 = 'male') puede cometer entre 1 y 10 errores con un 95% de confianza. En promedio comete 5.48, de acuerdo a la predicción del modelo realizado en el inciso anterior.

Ahora podemos hacer un barplot del número de errores, con esas condiciones:

```
## barplot para el número de errores en (2, -1, 'male')
barplot(table(yhat), xlab = "Número de errores en 10 intentos para (2, -
1, 'male')", las = 1,
        col = 'gold', ylab = 'Frecuencia', border = 'gold', space = .05)
abline(v = ci, col = 2, lwd = 2)
abline(v = lambdahat, col  = 4, lwd = 4)
```



Prueba de bondad de ajuste

Para rectificar que el modelo ajusta bien a los datos.

```
## here we use the deviance
## the null hypothesis is that our model is correctly specified
## we do NOT want to reject
with(modelo, pchisq(deviance, df = df.residual, lower.tail=FALSE))

## [1] 0.9542188
```

Recordar: No queremos rechazar H0.

Al tener un valor p de 0.9542, mayor a un alpha de 0.05, no se rechaza H0, por lo que se afirma que se ajusta bien el modelo.

- Otra forma de hacer prueba de bondad de ajuste:

```
## another option is to use the chi^2 statistic
yreal <- datos$y
lambdahat <- predict(modelo, type = 'response')
chi2 <- sum((yreal - lambdahat)^2/lambdahat)
chi2

## [1] 139.1197

df <- length(yhat) - 1
result <- c(statistic = chi2, df = df, p.value = pchisq(chi2, df = df,
```

```
lower.tail=FALSE))
result

## statistic        df   p.value
##  139.1197 9748.0000    1.0000
```

Recordar: No queremos rechazar H0.

Al tener un valor p de 1.0000, mayor a un alpha de 0.05, no se rechaza H0, por lo que se afirma que se ajusta bien el modelo.