A6 - Regresión Poisson

Héctor Hibran Tapia Fernández - A01661114

2024-10-29

Trabajaremos con el paquete dataset, que incluye la base de datos warpbreaks, que contiene datos del hilo (yarn) para identificar cuáles variables predictoras afectan la ruptura de urdimbre.

```
data = warpbreaks
head(data,10)
```

```
##    breaks wool tension
## 1      26    A       L
## 2      30    A       L
## 3      54    A       L
## 4      25    A       L
## 5      70    A       L
## 6      52    A       L
## 7      51    A       L
## 8      26    A       L
## 9      67    A       L
## 10     18    A       M
```

Este conjunto de datos indica cuántas roturas de urdimbre ocurrieron para diferentes tipos de telares por telar, por longitud fija de hilo:

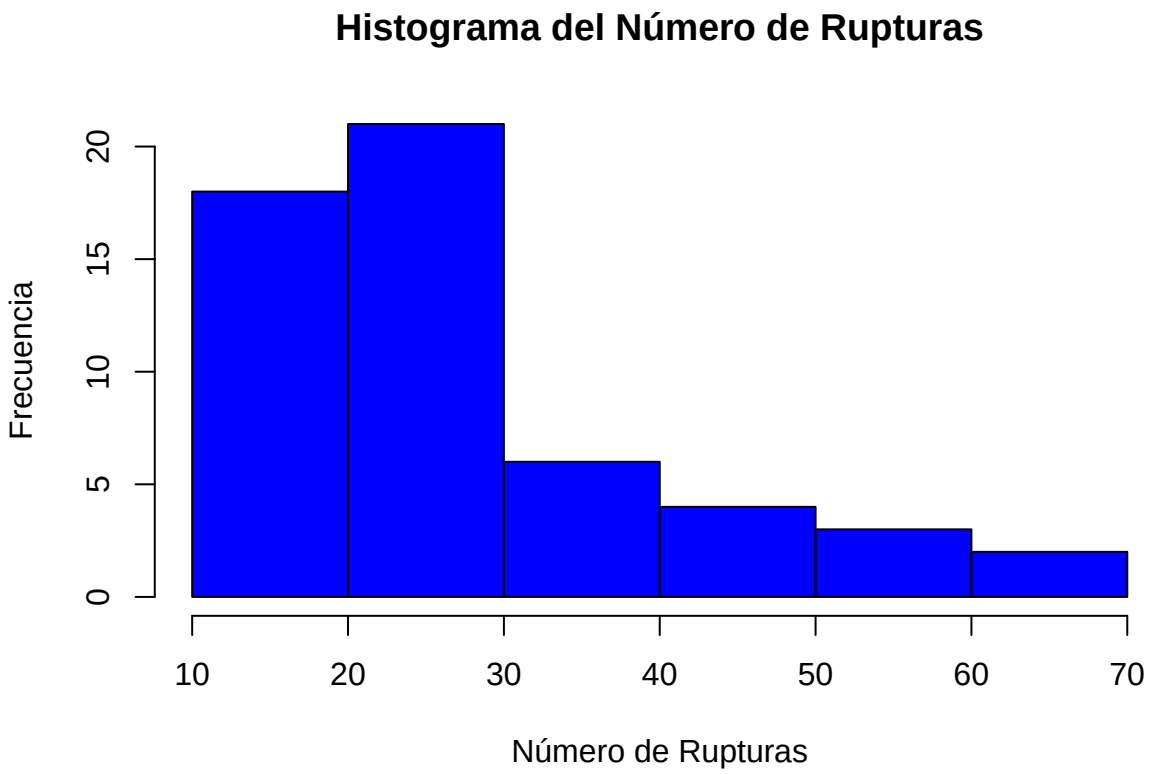
- breaks: número de rupturas
- wool: tipo de lana (A o B)
- tensión: el nivel de tensión (L, M, H)

Sigue el siguiente procedimiento de análisis:

I. Análisis Descriptivo

- Histograma del número de rupturas

```
hist(data$breaks, main = "Histograma del Número de Rupturas", xlab = "Número de Rupturas", ylab = "Frecu
```



- Obtén la media y la varianza de la variable dependiente

```
mean_breaks = mean(data$breaks)
var_breaks = var(data$breaks)

mean_breaks
```

```
## [1] 28.14815
```

```
var_breaks
```

```
## [1] 174.2041
```

- Interpreta en el contexto de una Regresión Poisson

El número de rupturas es una variable que se ajusta a un modelo donde se asume que la varianza es aproximadamente igual a la media. Lo que implica que las rupturas de urdimbre pueden modelarse en función de variables predictoras, como el tipo de lana y la tensión, las cuales impactan la frecuencia de rupturas. Este modelo es adecuado si el número de rupturas sigue una distribución de Poisson.

II. Ajusta dos modelos de Regresión Poisson

- Ajusta el modelo de regresión Poisson sin interacción

```
poisson_model_no_interaction = glm(breaks ~ wool + tension, data = data, family = poisson(link = "log"))
summary(poisson_model_no_interaction)
```

```
##
## Call:
## glm(formula = breaks ~ wool + tension, family = poisson(link = "log"),
##     data = data)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6871  -1.6503  -0.4269   1.1902   4.2616
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.69196    0.04541  81.302  < 2e-16 ***
## woolB       -0.20599    0.05157  -3.994 6.49e-05 ***
## tensionM    -0.32132    0.06027  -5.332 9.73e-08 ***
## tensionH    -0.51849    0.06396  -8.107 5.21e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 297.37  on 53  degrees of freedom
## Residual deviance: 210.39  on 50  degrees of freedom
## AIC: 493.06
##
## Number of Fisher Scoring iterations: 4
```

- Ajusta el modelo de regresión Poisson con interacción

```
poisson_model_interaction = glm(breaks ~ wool * tension, data = data, family = poisson(link = "log"))
summary(poisson_model_interaction)
```

```
##
## Call:
## glm(formula = breaks ~ wool * tension, family = poisson(link = "log"),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3383  -1.4844  -0.1291   1.1725   3.5153
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.79674    0.04994  76.030  < 2e-16 ***
## woolB          -0.45663    0.08019  -5.694 1.24e-08 ***
## tensionM       -0.61868    0.08440  -7.330 2.30e-13 ***
## tensionH       -0.59580    0.08378  -7.112 1.15e-12 ***
## woolB:tensionM  0.63818    0.12215   5.224 1.75e-07 ***
## woolB:tensionH  0.18836    0.12990   1.450    0.147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 297.37  on 53  degrees of freedom
## Residual deviance: 182.31  on 48  degrees of freedom
## AIC: 468.97
##
## Number of Fisher Scoring iterations: 4
```

III. Selección del modelo

Para seleccionar el modelo se toma en cuenta:

- Desviación residual: es la suma del cuadrado de los residuos estandarizados que se obtienen bajo el modelo. Con los grados de libertad se realiza una prueba de χ^2 para significancia del modelo.

```
residual_deviance_no_interaction = deviance(poisson_model_no_interaction)
df_no_interaction = df.residual(poisson_model_no_interaction)
p_value_no_interaction = pchisq(residual_deviance_no_interaction, df_no_interaction, lower.tail = FALSE)

residual_deviance_no_interaction
```

```
## [1] 210.3919
```

```
df_no_interaction
```

```
## [1] 50
```

```
p_value_no_interaction
```

```
## [1] 1.44606e-21
```

```
residual_deviance_interaction = deviance(poisson_model_interaction)
df_interaction = df.residual(poisson_model_interaction)
p_value_interaction = pchisq(residual_deviance_interaction, df_interaction, lower.tail = FALSE)

residual_deviance_interaction
```

```
## [1] 182.3051
```

```
df_interaction
```

```
## [1] 48
```

```
p_value_interaction
```

```
## [1] 1.582538e-17
```

- AIC: Criterio de Aikaike

```
AIC_no_interaction = AIC(poisson_model_no_interaction)
AIC_interaction = AIC(poisson_model_interaction)

AIC_no_interaction
```

```
## [1] 493.056
```

```
AIC_interaction
```

```
## [1] 468.9692
```

En este caso cabe recalcar que un menor AIC indica un mejor modelo, por lo tanto **el mejor modelo es el modelo con interacción.**

- Comparación entre los coeficientes y los errores estándar de de ambos modelos.

```
coefficients_no_interaction = summary(poisson_model_no_interaction)$coefficients
coefficients_interaction = summary(poisson_model_interaction)$coefficients

coefficients_no_interaction
```

```
##               Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)  3.6919631 0.04541069 81.301626 0.000000e+00
```

```
## woolB       -0.2059884 0.05157117 -3.994256 6.489775e-05
## tensionM    -0.3213204 0.06026580 -5.331721 9.728642e-08
## tensionH    -0.5184885 0.06395944 -8.106520 5.209021e-16
```

```
coefficients_interaction
```

```
##                  Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)     3.7967368 0.04993753 76.029734 0.000000e+00
## woolB          -0.4566272 0.08019202 -5.694172 1.239721e-08
## tensionM       -0.6186830 0.08440012 -7.330357 2.295399e-13
## tensionH       -0.5957987 0.08377723 -7.111702 1.146202e-12
## woolB:tensionM  0.6381768 0.12215312  5.224400 1.747203e-07
## woolB:tensionH  0.1883632 0.12989529  1.450115 1.470263e-01
```

Desviación residual (Prueba de χ^2)

- Si el modelo nulo explica a los datos, entonces la desviación nula será pequeña. Lo mismo ocurre con la Desviación residual . Puesto que es de suponer que el modelo contiene variables significativas, lo que importa que es la desviación residual del modelo sea suficientemente pequeño.

```
poisson_model = glm(breaks ~ wool * tension, data = data, family = poisson(link = "log"))
S = summary(poisson_model)
gl = S$null.deviance - S$df.residual
valor_frontera <- qchisq(0.05, gl, lower.tail = FALSE)
cat("Valor frontera de la zona de rechazo =", valor_frontera, "\n")
```

```
## Valor frontera de la zona de rechazo = 287.2075
```

```
dr = S$deviance
cat("Estadístico de prueba (Desviación residual) =", dr, "\n")
```

```
## Estadístico de prueba (Desviación residual) = 182.3051
```

```
vp = 1 - pchisq(dr, gl)
cat("Valor p =", vp, "\n")
```

```
## Valor p = 0.999506
```

- La prueba de χ^2 mide qué tan lejano está del cero la desviación residual del modelo. Entre más lejos esté del cero, el modelo será un buen modelo, entre más cerca, el modelo será un mal modelo que explicará poco la variabilidad de los datos. Su modelo supone:
  - H0: Deviance = 0
  - H1: Deviance > 0
  - gl = gl_desviación residual (n-(p+1))

```
S = summary(poisson_model_no_interaction)
gl = S$df.null - S$df.residual
valor_frontera <- qchisq(0.05, gl)
cat("Valor frontera de la zona de rechazo =", valor_frontera, "\n")
```

```
## Valor frontera de la zona de rechazo = 0.3518463
```

```
dr = S$deviance
cat("Estadístico de prueba =", dr, "\n")
```

```
## Estadístico de prueba = 210.3919
```

```
vp = 1 - pchisq(dr, gl)
cat("Valor p =", vp, "\n")
```

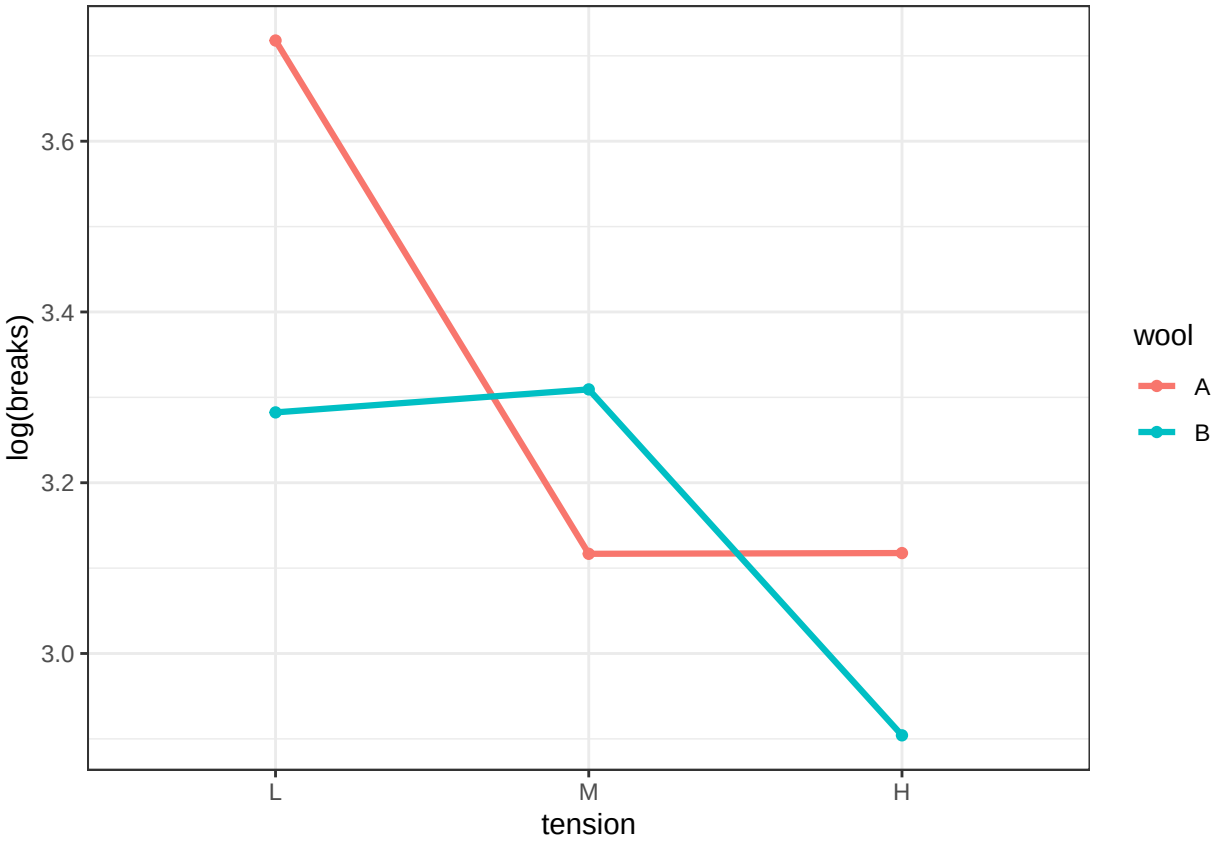
```
## Valor p = 0
```

Interpreta los coeficientes de ambos modelos.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
ggplot(data, aes(x = tension, y = log(breaks), group = wool, color = wool)) +
  stat_summary(fun = mean, geom = "point") +
  stat_summary(fun = mean, geom = "line", lwd = 1.1) +
  theme_bw() +
  theme(panel.border = element_rect(fill = "transparent"))
```



Como se observa las líneas no son paralelas y exhiben comportamientos opuestos lo que nos indica de una interacción clara en el gráfico entre wool y tension, el modelo con interacción es el mejor, esto y el AIC, lo soportan.

IV. Evaluación de los supuestos

Los supuestos principales que se deben cumplir son:

Independencia: haz la misma prueba de independencia que usaste en los modelos lineales.

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

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

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

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

```
dwtest(poisson_model_no_interaction, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  poisson_model_no_interaction
## DW = 2.0332, p-value = 0.7791
## alternative hypothesis: true autocorrelation is not 0
```

```
dwtest(poisson_model_interaction, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  poisson_model_interaction
## DW = 2.2376, p-value = 0.8499
## alternative hypothesis: true autocorrelation is not 0
```

Sobredispersión de los residuos. La sobredispersión de los residuos indicará que el modelo no cumple con el supuesto de que la media es igual a la varianza de los residuos. Para probarla se usa la prueba posgof, que es una prueba χ^2 con gl = grados de libertad residual. La desviación estándar se compara con los grados de libertad de la desviación residual, no deben ser muy diferentes. Esto indicará una sobredispersión de los residuos:

H0: No hay una sobredispersión del modelo H1: Hay una sobredispersión del modelo

Usaremos el modelo con interacción

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

```
## Loading required package: foreign
```

```
## Loading required package: survival
```

```
## Loading required package: MASS
```

```
## Loading required package: nnet
```

```
##
## Attaching package: 'epiDisplay'
```

```
## The following object is masked from 'package:lmtest':
##
##     lrtest
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
poisgof(poisson_model_interaction)
```

```
## $results
## [1] "Goodness-of-fit test for Poisson assumption"
##
## $chisq
## [1] 182.3051
##
## $df
## [1] 48
##
## $p.value
## [1] 1.582538e-17
```

```
poisson.model3 <- glm(breaks ~ wool * tension, data = data, family = quasipoisson(link = "log"))
summary(poisson.model3)
```

```
##
## Call:
## glm(formula = breaks ~ wool * tension, family = quasipoisson(link = "log"),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3383  -1.4844  -0.1291   1.1725   3.5153
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.79674    0.09688  39.189  < 2e-16 ***
## woolB          -0.45663    0.15558  -2.935 0.005105 **
## tensionM       -0.61868    0.16374  -3.778 0.000436 ***
## tensionH       -0.59580    0.16253  -3.666 0.000616 ***
## woolB:tensionM  0.63818    0.23699   2.693 0.009727 **
## woolB:tensionH  0.18836    0.25201   0.747 0.458436
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 3.76389)
##
##     Null deviance: 297.37  on 53  degrees of freedom
## Residual deviance: 182.31  on 48  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

```
bnm <- glm.nb(breaks ~ wool * tension, data = data, control = glm.control(maxit = 1000))
summary(bnm)
```

```
##
## Call:
## glm.nb(formula = breaks ~ wool * tension, data = data, control = glm.control(maxit = 1000),
##     init.theta = 12.08216462, link = log)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.09611  -0.89383  -0.07212   0.65270   1.80646
##
```

```
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.7967     0.1081  35.116  < 2e-16 ***
## woolB           -0.4566     0.1576  -2.898 0.003753 **
## tensionM        -0.6187     0.1597  -3.873 0.000107 ***
## tensionH        -0.5958     0.1594  -3.738 0.000186 ***
## woolB:tensionM   0.6382     0.2274   2.807 0.005008 **
## woolB:tensionH   0.1884     0.2316   0.813 0.416123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(12.0822) family taken to be 1)
##
##     Null deviance: 86.759  on 53  degrees of freedom
## Residual deviance: 53.506  on 48  degrees of freedom
## AIC: 405.12
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  12.08
##           Std. Err.:  3.30
##
##  2 x log-likelihood:  -391.125
```

V. Define cuál es tu mejor modelo

En base a lo anterior, el modelo **binomial negativa con interacción es el mejor modelo** para estos datos, debido a que:

- Maneja adecuadamente la sobredispersión sin violar los supuestos de independencia de los residuos.
- Tiene el AIC más bajo, tiene un mejor ajuste en comparación con el modelo cuasi-Poisson y el modelo de Poisson.