Modelos lineales generalizados con R

Jaime Isaac y Gerson Rivera

2021-04-28

Contents

Chapter 1

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.org/tinytex/.

Chapter 2

Introduction

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 2. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter **??**.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 2.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 2.1.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2021) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

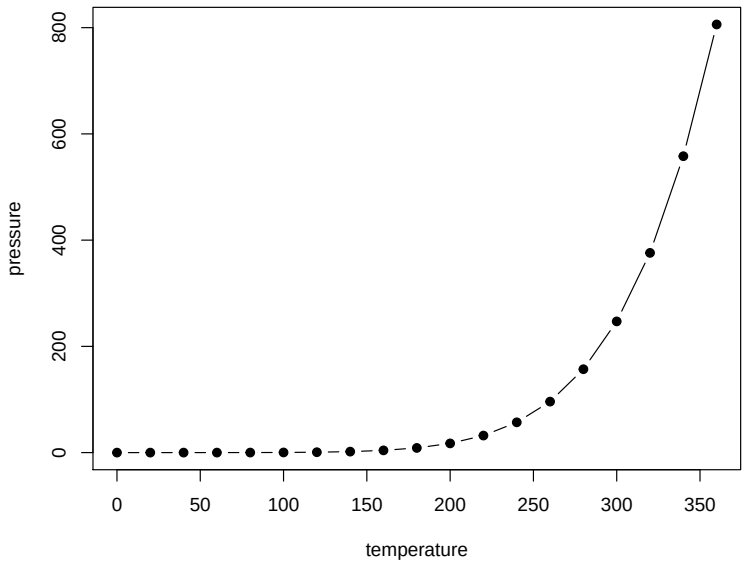


Figure 2.1: Here is a nice figure!

Table 2.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 3

Regresión Logística 1

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
require(MASS)
```

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

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
require(mosaic)
```

```
## Loading required package: mosaic
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affecte
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following objects are masked from 'package:psych':
##
##     logit, rescale
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:purrr':
##
##     cross
```

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

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
data(Pima.tr)
head(Pima.tr)
```

```
##   npreg glu bp skin  bmi   ped age type
## 1     5  86 68   28 30.2 0.364  24   No
## 2     7 195 70   33 25.1 0.163  55  Yes
## 3     5  77 82   41 35.8 0.156  35   No
## 4     0 165 76   43 47.9 0.259  26   No
## 5     0 107 60   25 26.4 0.133  23   No
## 6     5  97 76   27 35.6 0.378  52  Yes
```

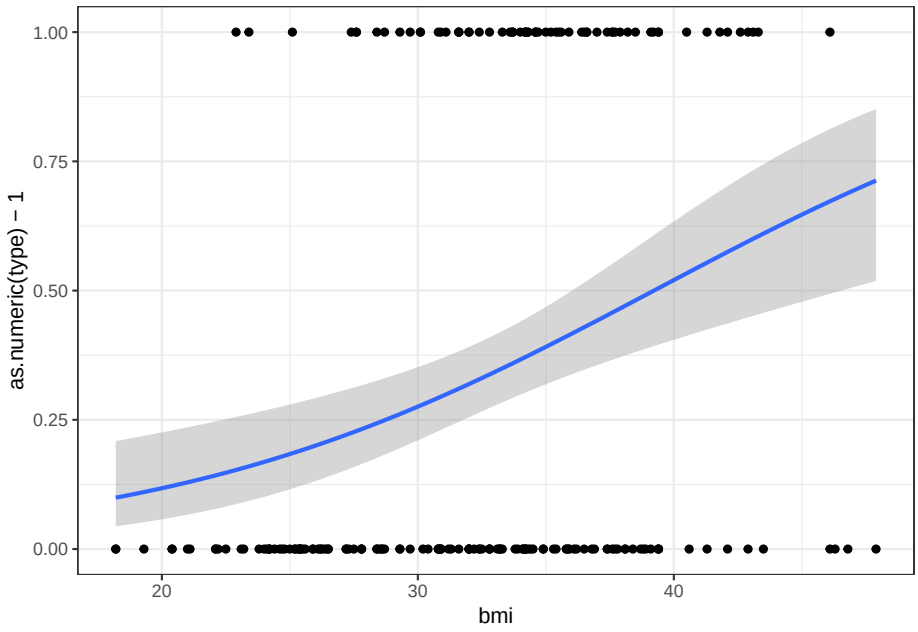
```
attach(Pima.tr)
```

Chapter 4

REGRESIÓN LOGÍSTICA SIMPLE

```
#y=as.numeric(type)-1 is needed for the plot
ggplot(Pima.tr, aes(x=bmi, y=as.numeric(type)-1)) + geom_point() +
geom_smooth(method="glm",
method.args=list(family="binomial"(link=logit)), se=TRUE) +
theme_bw()
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
diabetes.model <- glm(type~bmi,data=Pima.tr,family="binomial"(link=logit))
summary(diabetes.model)
```

```
##
## Call:
## glm(formula = type ~ bmi, family = binomial(link = logit), data = Pima.tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5797  -0.9235  -0.6541   1.2506   1.9377
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.11156    0.92806  -4.430 9.41e-06 ***
## bmi          0.10482    0.02738   3.829 0.000129 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 239.97  on 198  degrees of freedom
## AIC: 243.97
##
## Number of Fisher Scoring iterations: 4
```

Si solo observa el valor p, el resultado no sorprendente es que hay un relación entre el índice de masa corporal y la diabetes.

Profundizando en la salida con más detalle, nuestra ecuación de regresión logística es:

$$ln(\frac{\hat{p}}{1-\hat{p}}) = -4.11156 + 0.10482X$$

Suponga que queremos hacer una predicción para una mujer en esta población con un índice de masa corporal de $X = 30$. Sustituya la ecuación para obtener su logit

$$ln(\frac{\hat{p}}{1-\hat{p}}) = -4.11156 + 0.10482(30) = -0.96696$$

Observe que su logit (log odds-ratio) es negativo. Esto será cierto siempre que la probabilidad predicha $\hat{p} < 0.5$, por lo que en este escenario usted querría un logit negativo. Cuando $\hat{p} > 0.5$, el logit será positivo, y si $\hat{p} = 0.5$ entonces el logit es::

$$ln(\frac{0.5}{1-0.5}) = ln(1) = 0$$

Probablemente preferiria una probabilidad o un porcentaje en lugar de un logit.Tome la funcion logit inversa para obtener esto.

$$\hat{p} = \frac{exp(-0.96696)}{1 + exp(-0.96696)} = 0.275$$

Estamos pronosticando un 27.5% de probabilidad de diabetes tipo II cuando el índice de masa corporal es igual a $X = 30$.

Prestando atención al parámetro de "pendiente" β_1, su estimación es 0.10482. Eso es positivo, lo que significa que bmi está asociado positivamente con el evento, que tiene diabetes tipo II. Por cada aumento de 1 unidad en X (es decir, alguien gana peso suficiente para que el bmi suba en 1), el aumento previsto en el logit es 0.10482.

Si esto no significa mucho para usted, entonces podemos exponencializar la pendiente para convertir el registro de la proporción de log probabilidades en solo la proporción de probabilidades.

$$exp(\hat{\beta_1}) = e^{\hat{\beta_1}} = e^{0.10428} = 1.11$$

Entonces, la razón de posibilidades es 1.11. Esto significa que por cada aumento de 1 en el bmi, la probabilidad de tener diabetes tipo II aumenta en un 11%. Si la razón de posibilidades era exactamente 1, eso indicaría una probabilidad

igual (es decir, la variable no estaría asociada con la evento), y las razones de probabilidad por debajo de 1 indican que la probabilidad disminuye a medida que la variable aumenta. Si hubo un ejercicio de variable en el conjunto de datos que fue asociado con tener diabetes, esperaríamos que su β fuera negativo, por lo que el La razón de posibilidades estaría entre 0 y 1.

Si quisiéramos observar el impacto de un aumento de 10 puntos en el bmi ($\Delta x = 10$)

$exp(\Delta x\hat{\beta}_1) = exp(10 \times 0.10428) = e^{1.0428} = 2.85$
Las probabilidades(odss) de tener diabetes casi se triplicarían si el IMC aumenta en 10 unidades.

Chapter 5

Test de Wald para regresión logística

R de forma predeterminada utiliza la prueba de Wald en la tabla de resumen para un lineal generalizado modelo. Repitamos esa tabla

```
summary(diabetes.model)
```

```
##
## Call:
## glm(formula = type ~ bmi, family = binomial(link = logit), data = Pima.tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5797  -0.9235  -0.6541   1.2506   1.9377
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.11156    0.92806  -4.430 9.41e-06 ***
## bmi          0.10482    0.02738   3.829 0.000129 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 239.97  on 198  degrees of freedom
## AIC: 243.97
##
## Number of Fisher Scoring iterations: 4
```

R informa el estadístico de Wald z, que es la raíz cuadrada de la prueba de Wald χ^2 discutido en el capítulo 21. Esto se basa en el hecho matemático de que si al cuadrado de la distribución normal estándar Z, se obtiene una distribución chi-cuadrado con $df = 1$.

Las hipótesis son:

$$H_0 : \beta_1 = 0 \qquad \text{vs.} \qquad H_1 : \beta_1 \neq 0 \tag{5.1}$$

o en términos de odds-ratio θ

$$H_0 : \theta_1 = 0 \qquad \text{versus} \qquad H_1 : \theta_1 \neq 0 \tag{5.2}$$

Observe que el estadístico z dado es la estimación dividida por el error estándar y el p-valor se basa en la distribución normal estándar.

$$z = \frac{\hat{\beta}_1}{S_{\hat{\beta}_1}} = \frac{0.10482}{0.02738} = 3.849$$

Existe una relación significativa entre bmi y diabetes tipo II:

Wald $z = 3.829$,$p = 0.000129$

Algunos paquetes de software darán la prueba de chi-cuadrado de Wald en su lugar, que es solo nuestra estadística al cuadrado.

$$\chi^2 = \left(\frac{\hat{\beta}_1}{S_{\beta_1}}\right)^2 = \left(\frac{0.10482}{0.02738}\right)^2 = 14.656$$

Dado que esta estadística es chi-cuadrado con $df = 1$, el valor de p es:

```
1-pchisq(14.656,df=1)
```

```
## [1] 0.0001290233
```

El intervalo de confianza de Wald para β_1 se calcula de manera similar a muchos otros intervalos de confianza que hemos visto

$$\beta_1 \pm z(S_{\beta_1})$$

Para nuestro al 95% de confianza:

$$0.10482 \pm 1.96 \times 0.02738$$

$$0.10482 \pm 0.05366$$

$$(0.05062, 0.15794)$$

Exponencia este intervalo para obtener un intervalo de confianza para la razón de posibilidades θ

$$(e^{0.05062}, e^{0.05062})$$

Observe que el IC completo para β_1 está por encima de cero y, de manera equivalente, el IC completo para θ es por encima de uno. Esto indica una relación significativa.

Chapter 6

Prueba de razón de verosimilitud.

La inferencia para modelos lineales generalizados también se puede realizar con una razón de verosimilitud.

Esto implicará el análisis de la tabla de deviance.

```
library(car)
```

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

```
##
## Attaching package: 'car'
```

```
## The following objects are masked from 'package:mosaic':
##
##     deltaMethod, logit
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
Anova(diabetes.model,type="II",test="LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: type
##     LR Chisq Df Pr(>Chisq)
## bmi   16.445  1  5.008e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Leyendo de la salida, vemos que $\chi^2 = 16.445$ con $df = 1$, $p < 0.0001$.

Observe que el estadístico de la prueba de chi cuadrado NO es igual al chi cuadrado de Wald prueba calculada anteriormente.

$$\chi^2 = -2ln\left(\frac{\mathcal{L}_{\mathcal{R}}}{\mathcal{L}_{\mathcal{F}}}\right)$$

$$\chi^2 = -2\left(ln\mathcal{L}_{\mathcal{R}} - ln\mathcal{L}_{\mathcal{F}}\right)$$

Hasta ahora, solo hemos ajustado el tipo de modelo completo type ~ bmi. Ajustemos el modelo reducido escriba ~ 1 (es decir, un modelo de solo intercepción o "nulo") y calculamos las probabilidades logarítmicas y la deviance con R.

```
diabetes.null <- glm(type~1,data=Pima.tr,family="binomial"(link=logit))
logLik(diabetes.null)
```

```
## 'log Lik.' -128.2071 (df=1)
```

```
logLik(diabetes.model)
```

```
## 'log Lik.' -119.9846 (df=2)
```

```
diff <- logLik(diabetes.null)[1] - logLik(diabetes.model)[1]
chisq.LRT <- -2*diff
chisq.LRT
```

```
## [1] 16.44499
```

```
pval.LRT <- 1-pchisq(chisq.LRT,df=1)
pval.LRT
```

```
## [1] 5.008242e-05
```

Observe que obtenemos la misma estadística de prueba proporcionada por el comando Anova. También mire nuevamente la parte inferior del resumen.

```
summary(diabetes.model)
```

```
##
## Call:
## glm(formula = type ~ bmi, family = binomial(link = logit), data = Pima.tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5797  -0.9235  -0.6541   1.2506   1.9377
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.11156    0.92806  -4.430 9.41e-06 ***
## bmi          0.10482    0.02738   3.829 0.000129 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 239.97  on 198  degrees of freedom
## AIC: 243.97
##
## Number of Fisher Scoring iterations: 4
```

Observe que la diferencia de la desviación nula (256.41) y la desviación residual (239.97) es 16.44, nuestro estadístico de prueba de chi-cuadrado con $199-198 = 1df$ El La desviación nula es -2 veces la forma logarítmica del modelo reducido, mientras que la la desviación es -2 veces la probabilidad logarítmica del modelo completo. Esto es análogo a el concepto de "SS extra" de las pruebas F parciales.

Chapter 7

REGRESIÓN LOGÍSTICA MÚLTIPLE

Veamos cómo ajustar varios modelos de regresión logística a un conjunto de datos. Tomemos el conjunto de datos Pima.tr y cree una variable categórica Mom donde una mujer con npreg> 0 se clasifica como madre (sin tener en cuenta la posibilidad de que algunos los embarazos pueden no haber resultado en un nacimiento vivo).

```
Pima.tr <- Pima.tr %>%
mutate(Mom=ifelse(npreg==0,"No","Yes"))
xtabs(~type+Mom,data=Pima.tr)
```

```
##      Mom
## type   No Yes
##   No   16 116
##   Yes  12  56
```

Calculemos la razón de posibilidades(odss-ratio) de la tabla :

$$OR = \frac{\frac{a}{b}}{\frac{c}{d}} = \frac{\frac{16}{116}}{\frac{12}{256}} = 0.6437$$

Tomaremos el recíproco para facilitar la interpretación, $\frac{1}{OR} = \frac{1}{0.6437} = 1.5536$

Las probabilidades de tener diabetes tipo II son 1.55 veces mayores para las no madres que para las madres. Observe que aproximadamente el 43% de las no madres y el
33% de las madres tienen el Tipo II diabetes.

Primero, encuentre un intervalo de confianza para el logaritmo de la razón de posibilidades(logit)

$$ln(OR) \pm z\sqrt{\frac{1}{a}+\frac{1}{b}+\frac{1}{c}+\frac{1}{d}}$$

$$ln(1.5536) \pm (1.96)\sqrt{\frac{1}{16}+\frac{1}{116}+\frac{1}{12}+\frac{1}{56}}$$

$$(0.4406 \pm 0.8316)$$

$$(-0.3730, 1.2542)$$

Este intervalo de confianza contiene 0, por lo que la variable Mom no es un predictor significativo de la diabetes tipo II. Si prefiere el IC en términos de razón de posibilidades, exponencial.

$$(e^{-0.3730}, e^{1.2542})$$

$$(0.6887, 3.5050)$$

El IC incluye el valor 1 (que indica que no hay efecto para la razón de posibilidades).

Usemos R para ajustar el tipo de type~Mom

```
mod0<- glm(type~1,data=Pima.tr,family="binomial"(link=logit))
mod1<- glm(type~Mom,data=Pima.tr,family="binomial"(link=logit))
summary(mod1)
```

```
##
## Call:
## glm(formula = type ~ Mom, family = binomial(link = logit), data = Pima.tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0579  -0.8876  -0.8876   1.4981   1.4981
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.2877     0.3819  -0.753    0.451
## MomYes       -0.4406     0.4151  -1.061    0.289
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 255.31  on 198  degrees of freedom
## AIC: 259.31
##
## Number of Fisher Scoring iterations: 4
```

```
Anova(mod1,test="LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: type
##     LR Chisq Df Pr(>Chisq)
## Mom   1.1056  1     0.2931
```

```
confint(mod1)
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %    97.5 %
## (Intercept) -1.058545 0.4561131
## MomYes      -1.250578 0.3907974
```

Vemos que Mom no es significativa con la prueba de Wald o la razón de verosimilitud prueba. El intervalo de confianza se calcula con una fórmula más compleja que dado aquí, por lo que los resultados no son idénticos. Los signos son opuestos porque mi La tabla usó **No** como un éxito y el ajuste glm de R usó **Sí** como un éxito. Hagamos una regresión logística múltiple y hagamos una predicción. Usaré 3 predictores bmi, age y Mom.

```
mod2 <- glm(type~bmi+age+Mom,data=Pima.tr,family="binomial"(link=logit))
summary(mod2)
```

```
##
## Call:
## glm(formula = type ~ bmi + age + Mom, family = binomial(link = logit),
##     data = Pima.tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8477  -0.8045  -0.4907   0.9983   2.3009
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.76192    1.23679  -4.659 3.18e-06 ***
## bmi          0.09703    0.03011   3.223  0.00127 **
## age          0.07830    0.01628   4.809 1.52e-06 ***
## MomYes      -0.83471    0.48052  -1.737  0.08237 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 212.96  on 196  degrees of freedom
## AIC: 220.96
##
## Number of Fisher Scoring iterations: 4
```

Observe que el bmi y la edad son predictores significativos de bmi con valores pendiente positivas. Lo que indica un mayor riesgo a medida que aumenta el bmi o la edad. Mom no es significativo en $\alpha = 0.05$, pero tiene una pendiente negativa que indica que las madres fueron menos probabilidades de ser diabéticas que las no madres.

Si una mujer tiene 40 años, un bmi de 28 y es madre, calculemos su probabilidad de diabetes tipo II.

$$ln\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) = -5.76192 + 0.09703(28) + 0.07830(40) - 0.83471(1) = -0.74779$$

Su logit negativo indica menos del 50% de posibilidades de diabetes. Tomando el logit inverso:

$$\frac{e^{-0.74779}}{1+e^{-0.74779}} = 0.321$$

La probabilidad es de aproximadamente el 32%. Ahora hazlo para una persona con las mismas estadísticas, excepto que no sea madre.

$$ln\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) = -5.76192 + 0.09703(28) + 0.07830(40) - 0.83471(0) = 0.08692$$

Ahora el logit es positivo, por lo que la probabilidad será superior al 50%.

$$\frac{e^{0.08692}}{1+e^{0.08692}} = 0.522$$

Una prueba de razón de verosimilitud que compara el modelo 1 (solo con mamá) y el modelo 2 (con bmi, age y Mom) tendrá $df = 2$ con dos parámetros adicionales, y tendríamos esperar que el segundo modelo sea una mejora significativa.

```
anova(mod1,mod2,test="LR")
```

```
## Analysis of Deviance Table
##
## Model 1: type ~ Mom
## Model 2: type ~ bmi + age + Mom
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       198     255.31
## 2       196     212.96  2   42.353 6.355e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vemos que hay una mejora significativa, con $\chi^2 = 42.353$, $df = 2$ y $p = 0.0001$.

Tal vez no deberíamos haber creado una variable categórica como Mom, pero solo usar npreg. Encajaré un tercer tipo de type ~ bmi + age + npreg.

```
mod3 <- glm(type~bmi+age+npreg,data=Pima.tr,family="binomial"(link=logit))
summary(mod3)
```

```
##
## Call:
## glm(formula = type ~ bmi + age + npreg, family = binomial(link = logit),
##     data = Pima.tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7413  -0.8235  -0.4918   0.9773   2.2382
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.44462    1.18360  -5.445 5.18e-08 ***
## bmi          0.10761    0.02986   3.604 0.000313 ***
## age          0.05937    0.01817   3.267 0.001086 **
## npreg        0.06508    0.05720   1.138 0.255226
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 214.62  on 196  degrees of freedom
## AIC: 222.62
##
## Number of Fisher Scoring iterations: 4
```

Algo extraño, ya que npreg no es significativo, pero el signo de su estimación es positivo, en lugar de negativo para Mom. Puede haber alguna explicación médica. No soy conciente de.

Por último, suponga que quisiera comparar los modelos 2 y 3. No están anidados, por lo que necesita usar AIC en su lugar. Crearé una tabla AIC para los cuatro modelos (incluidos el modelo nulo).

```
require(bbmle)
```

```
## Loading required package: bbmle
```

```
## Loading required package: stats4
```

```
##
## Attaching package: 'bbmle'
```

```
## The following object is masked from 'package:dplyr':
##
##     slice
```

```
AICtab(mod0,mod1,mod2,mod3,base=TRUE,delta=TRUE,weights=TRUE,sort=TRUE)
```

```
##      AIC   dAIC  df weight
## mod2 221.0   0.0 4  0.7
## mod3 222.6   1.7 4  0.3
## mod0 258.4  37.5 1  <0.001
## mod1 259.3  38.4 2  <0.001
```

Al AIC parece gustarle un poco más el Model 2 que el Model 3, aunque no hasta cierto punto eso se consideraría sustancial. El modelo 0 y el modelo 1 son muy débiles, con $\Delta_i > 10$ y pesos diminutos Akaike $w_i < 0.001$.

```
predict(object = mod2, newdata = data.frame(bmi =30.3 ,age=27,Mom="No"))
```

```
##          1
## -0.7077035
```

Chapter 8

Regresión de Poisson

8.1 ¿Qué son los modelos de regresión de Poisson?

Los modelos de *regresión de Poisson* se utilizan mejor para modelar eventos en los que se cuentan los resultados. O, más específicamente, contar datos: datos discretos con valores enteros no negativos que cuentan algo, como la cantidad de veces que ocurre un evento durante un período de tiempo determinado o la cantidad de personas en la fila en la tienda de comestibles.

Los datos de recuento también se pueden expresar como datos de tasa, ya que el número de veces que ocurre un evento dentro de un período de tiempo se puede expresar como una cuenta sin procesar (es decir, "En un día, comemos tres comidas") o como una tasa ("Comemos a una tasa de 0,125 comidas por hora").

La regresión de Poisson nos ayuda a analizar tanto los datos de recuento como los datos de tasa al permitirnos determinar qué variables explicativas (valores X) tienen un efecto en una variable de respuesta dada (valor Y, el recuento o una tasa). Por ejemplo, una tienda de comestibles podría aplicar la regresión de Poisson para comprender y predecir mejor la cantidad de personas en una línea.

8.2 ¿En qué se diferencia la distribución de Poisson de la distribución normal?

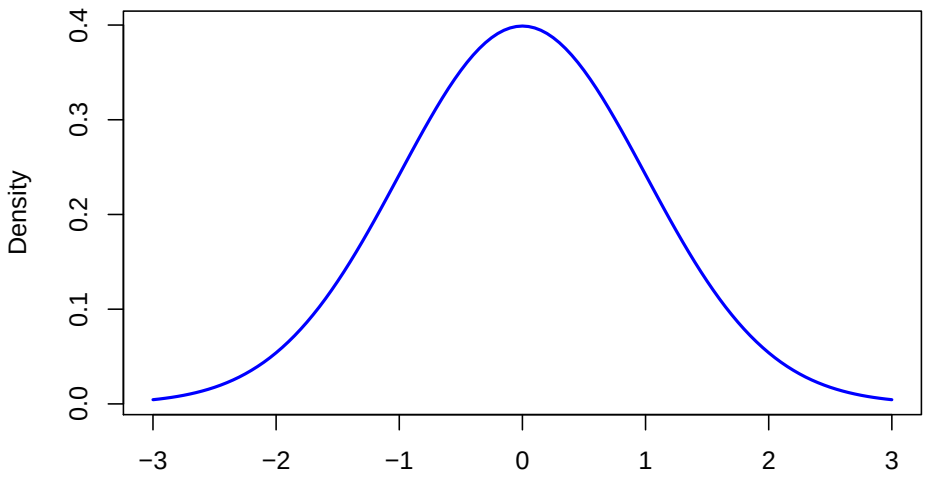
La distribución de Poisson se usa más comúnmente para encontrar la probabilidad de que ocurran eventos dentro de un intervalo de tiempo dado. Dado que

estamos hablando de un recuento, con la distribución de Poisson, el resultado debe ser 0 o superior; no es posible que un evento ocurra un número negativo de veces. Por otro lado, la distribución normal es una distribución continua para una variable continua y podría resultar en un valor positivo o negativo:

Distribución Poisson	Distribución Normal
Se utiliza para contar datos o tasa(razón) de datos.	Usada para variables contínuas
Sesgada según los valores de lambda	Curva en forma de campana que es simétrica alrededor de la media
Varianza igual que la media	La varianza y la media son parámetros diferentes; media, mediana y moda son iguales

Podemos generar una distribución normal en R así:

```
# create a sequence -3 to +3 with .05 increments
xseq<-seq(-3, 3, .05)
# generate a Probability Density Function
densities <- dnorm(xseq, 0, 1)
# plot the graph
plot(xseq, densities, col = "blue", xlab = "", ylab = "Density", type = "l", lwd = 2)
```




```
# col: changes the color of line
# 'xlab' and 'ylab' are labels for x and y axis respectively
# type: defines the type of plot. 'l' gives a line graph
# lwd: defines line width
```

En R, **dnorm (secuencia, media, std.dev)** se usa para trazar la función de densidad de probabilidad (PDF) de una distribución normal.

Para comprender la distribución de Poisson, considere el siguiente problema del libro de texto Tutorial R de Chi Yau:

Si hay 12 automóviles que cruzan un puente por minuto en promedio, ¿cuál es la probabilidad de que diecisiete o más automóviles crucen el puente en un minuto dado?

Aquí, la cantidad promedio de automóviles que cruzan un puente por minuto es $\mu = 12$.

`ppois(q, u, lower.tail = TRUE)` es una función R que da la probabilidad de que una variable aleatoria sea menor o igual a un valor.

Tenemos que encontrar la probabilidad de tener diecisiete o más autos, por lo que usaremos `lower.trail = FALSE` y estableceremos q en 16:

```
ppois(16, 12, lower.tail = FALSE)
```

```
## [1] 0.101291
```

```
# lower.tail = logical; if TRUE (default) then probabilities are P[X < = x], otherwise, P[X > x].
```

Para obtener un porcentaje, simplemente necesitamos multiplicar esta salida por 100. Ahora tenemos la respuesta a nuestra pregunta: hay una probabilidad del 10.1% de tener 17 o más autos cruzando el puente en cualquier minuto en particular.

8.3 Modelos de regresión de Poisson y GLM(Generalized Linear MOdels)

Los modelos lineales generalizados son modelos en los que las variables de respuesta siguen una distribución diferente a la distribución normal. Eso contrasta con los modelos de regresión lineal, en los que las variables de respuesta siguen una distribución normal. Esto se debe a que los modelos lineales generalizados tienen variables de respuesta que son categóricas, como Sí, No; o Grupo A, Grupo B y, por lo tanto, no van de ∞ a ∞. Por tanto, la relación entre la respuesta y las variables predictoras puede no ser lineal. En GLM

$$y_i = \alpha + \beta_1 X_{1i} + \beta_2 X_{2i} + ... + \beta_p X_{pi} + e_i,\ i = 1, 2, ..., n$$

La variable de respuesta y_i se modela mediante una función lineal de variables predictoras y algún término de error.

Un **modelo de regresión de Poisson** es un modelo lineal generalizado (GLM) que se utiliza para modelar datos de recuento y tablas de contingencia. La salida Y (recuento) es un valor que sigue la distribución de Poisson. Asume el logaritmo de los valores esperados (media) que pueden modelarse en forma lineal mediante algunos parámetros desconocidos.

Para transformar la relación no lineal en forma lineal, se utiliza una función de enlace que es el logaritmo de la regresión de Poisson. Por esa razón, un modelo de regresión de Poisson también se denomina modelo log-lineal. La forma matemática general del modelo de regresión de Poisson es:

$$log(y) = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

donde,

- y es la variable respuesta
- α y β: son coeficientes numéricos, α es la intersección, a veces α también está representada por β_0, es lo mismo.
- x es la variable predictora/explicativa

Los coeficientes se calculan utilizando métodos como la **Estimación de máxima verosimilitud (MLE)** o la cuasi-verosimilitud máxima.

Considere una ecuación con una variable predictora y una variable de respuesta:

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

Esto es equivalente a,

$$y = e^{\alpha+\beta X}$$

Nota: En los modelos de regresión de Poisson, las variables predictoras o explicativas pueden tener una combinación de valores numéricos o categóricos.

Una de las características más importantes para la distribución de Poisson y la regresión de Poisson es la equidispersión, lo que significa que la media y la varianza de la distribución son iguales.

La varianza mide la dispersión de los datos. Es el "promedio de las diferencias al cuadrado de la media". La varianza (Var) es igual a 0 si todos los valores son idénticos. Cuanto mayor sea la diferencia entre los valores, mayor será la varianza. La media es el promedio de valores de un conjunto de datos. El promedio es la suma de los valores dividida por el número de valores.

Digamos que la media (μ) se denota por $E(X)$

$$E(X) = \mu$$

Para la regresión de Poisson, la media y la varianza se relacionan como:

$$var(X) = \sigma^2 E(X)$$

Donde σ^2 es el parámetro de Dispersión.

Dado que $var(X) = E(X)$ (varianza = media) debe ser válida para que el modelo de Poisson se ajuste completamente, σ^2 debe ser igual a 1.

Cuando la varianza es mayor que la media, eso se denomina sobredispersión y es mayor que 1. Si es menor que 1, se conoce como subdispersión.

8.4 Modelado de regresión de Poisson utilizando datos de recuento.

En R, el comando `glm ()`se usa para modelar modelos lineales generalizados. Aquí está la estructura general de `glm ()`:

```
glm(formula, family = familytype(link = ""), data,...)
```

formula La fórmula es una representación simbólica de cómo se modela para ajustar.

family La familia indica la elección de las funciones de varianza y enlace. Hay varias opciones de familia, incluidas Poisson y Logistic.

datos Los datos son el conjunto de datos que se utilizará.

`glm()` ofrece ocho opciones para la familia con las siguientes funciones de enlace predeterminadas:

Family	**Default Link Function**
binomial	(link = “logit”)
gaussian	(link = “identity”)
Gamma	(link = “inverse”)
inverse.gaussian	(link = $frac{1}{mu^2}$)
poisson	
	(link = “log”)
quasi	(link = “identity”, variance = “constant”)
quasibinomial	
	(link = “logit”)
quasipoisson	(link = “log”)

8.4.1 ¡Empecemos a modelar!

Vamos a modelar la regresión de Poisson relacionada con la frecuencia con la que el hilo se rompe durante el tejido. Estos datos se encuentran en el paquete **datasets** en R, por lo que lo primero que debemos hacer es instalar el paquete usando `install.packages("datasets")` y cargar la biblioteca con la librería `library(datasets)`:

```
# install.packages("datasets")
library(datasets) # include library datasets after installation
```

El paquete**datasets**incluye toneladas de conjuntos de datos, por lo que debemos seleccionar específicamente nuestros datos de hilo(yarn). Consultando la documentación del paquete, podemos ver que se llama **warpbreaks**, así que almacenémoslo como un objeto.

```
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
```

Echemos un vistaso a los datos:

```
columns<-names(data) # Extract column names from dataframe
columns # show columns
```

```
## [1] "breaks"  "wool"    "tension"
```

¿Qué hay en nuestros datos?

Este conjunto de datos analiza cuántas roturas de urdimbre ocurrieron para diferentes tipos de telares por telar, por longitud fija de hilo. Podemos leer más detalles sobre este conjunto de datos en la documentación aquí, pero aquí están las tres columnas que veremos y a qué se refiere cada una:

Variable	Tipo	Descripción
breaks	numérica	número de roturas
wool	factor	El tipo de lana(A o B)
tension	factor	El nivel de tensión (L, M, H)

Hay medidas en 9 telares de cada uno de los seis tipos de deformación, para un total de 54 entradas en el conjunto de datos.

Veamos cómo se estructuran los datos mediante el comando`ls.str ():`

```
ls.str(warpbreaks)
```

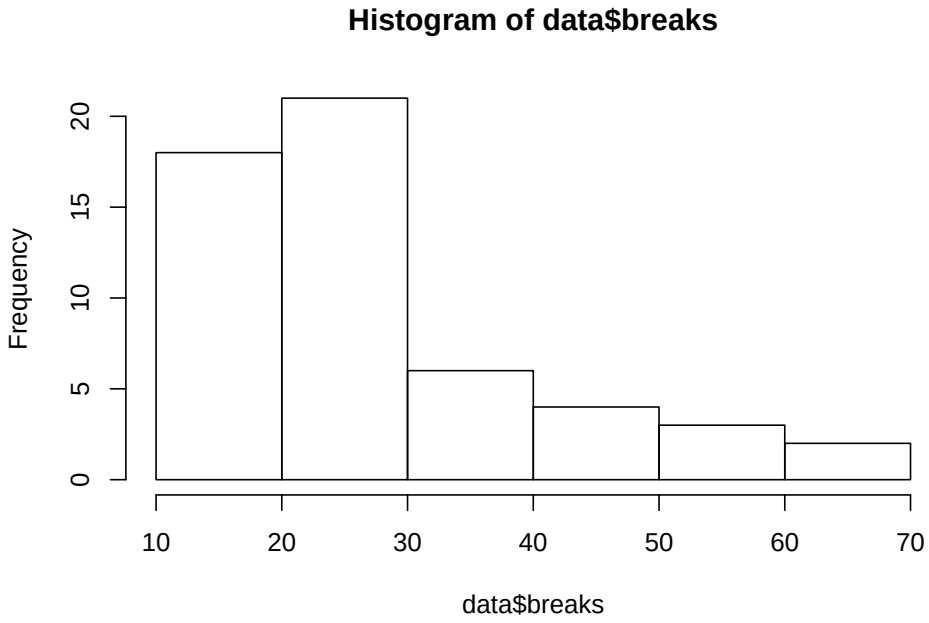
```
## breaks :  num [1:54] 26 30 54 25 70 52 51 26 67 18 ...
## tension :  Factor w/ 3 levels "L","M","H": 1 1 1 1 1 1 1 1 1 2 ...
## wool :  Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
```

De lo anterior, podemos ver tanto los tipos como los niveles presentes en los datos. Lea esto para aprender un poco más sobre los factores en R.

Ahora trabajaremos con el marco de datos. Recuerde, con un modelo de distribución de Poisson estamos tratando de averiguar cómo algunas variables predictoras afectan una variable de respuesta. Aquí, `breaks` es la variable de respuesta y `wool` y `tension` son variables predictoras.

Podemos ver que la variable dependiente breaks de datos contínuos creando un histograma:

```
hist(data$breaks)
```



Claramente, los datos no tienen la forma de una curva de campana como en una distribución normal.

Veamos la media `mean()`y la varianza `var ()`de la variable dependiente:

```
mean(data$breaks) # calculate mean
```

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

```
var(data$breaks) # calculate variance
```

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

La varianza es mucho mayor que la media, lo que sugiere que tendremos una dispersión excesiva en el modelo.

Ajustemos el modelo de Poisson usando el comando `glm ()`.

```
# model poisson regression using glm()
poisson.model<-glm(breaks ~ wool + tension, data, family = poisson(link = "log"))
summary(poisson.model)
```

```
##
## 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
```

`summary ()` es una función genérica que se utiliza para producir resúmenes de los resultados de varias funciones de ajuste de modelos.

8.5 Interpretación del modelo de Poisson

Nos acaban de dar mucha información, ahora necesitamos interpretarla. La primera columna llamada Estimación son los valores de los coeficientes de α(intersección), β_1 y así sucesivamente. A continuación se muestra la interpretación de las estimaciones de los parámetros:

- $exp(\alpha)$= efecto sobre la media μ, cuando $X = 0$
- $exp(\beta)$ = con cada unidad de aumento en X, la variable predictora tiene un efecto multiplicativo de $exp(\beta)$ sobre la media de Y, es decir μ.
- Si $\beta = 0$, entonces $exp(\beta) = 1$, y el recuento esperado es $exp(\alpha)$ y, Y y X no están relacionados.
- Si $\beta > 0$, entonces $exp(\beta) > 1$, y el recuento esperado es $exp(\beta)$ veces mayor que cuando $X = 0$

- Si $\beta < 0$, entonces $exp(\beta) < 1$, y el recuento esperado es $exp(\beta)$ veces menor que cuando $X = 0$

Si`family = poisson` se mantiene en `glm ()`, estos parámetros se calculan utilizando la estimación de máxima verosimilitud MLE.

R trata las variables categóricas como variables ficticias. Las variables categóricas, también llamadas variables indicadoras, se convierten en variables ficticias asignando a los niveles de la variable alguna representación numérica. La regla general es que si hay k categorías en una variable factorial, la salida de `glm ()` tendrá k-1 categorías con 1 restante como categoría base.

Podemos ver en el resumen anterior que para la lana, "A" se ha hecho la base y no se muestra en el resumen. De manera similar, para la tensión, "L" se ha convertido en la categoría base.

Para ver qué variables explicativas tienen un efecto sobre la variable de respuesta, veremos los valores p. Si la p es menor que 0.05 entonces, la variable tiene un efecto sobre la variable de respuesta. En el resumen anterior, podemos ver que todos los valores de p son menores a 0.05, por lo que ambas variables explicativas (lana y tensión) tienen un efecto significativo en las roturas. Observe cómo la salida de R usó *** al final de cada variable. El número de estrellas significa significancia.

Antes de comenzar a interpretar los resultados, verifiquemos si el modelo tiene una dispersión excesiva o insuficiente. Si la desviación residual es mayor que los grados de libertad, entonces existe una dispersión excesiva. Esto significa que las estimaciones son correctas, pero los errores estándar (desviación estándar) son incorrectos y el modelo no los tiene en cuenta.

La desviación nula muestra qué tan bien se predice la variable de respuesta mediante un modelo que incluye solo el intercepto (gran media) mientras que el residual con la inclusión de variables independientes. Arriba, podemos ver que la suma de 3 (53-50 = 3) variables independientes disminuyó la desviación de 297.37 a 210.39. Una mayor diferencia de valores significa un mal ajuste.

Entonces, para tener un error estándar más correcto, podemos usar un modelo cuasi-poisson:

```
poisson.model2<-glm(breaks ~ wool + tension, data = data, family = quasipoisson(link =
summary(poisson.model2)
```

```
##
## Call:
## glm(formula = breaks ~ wool + tension, family = quasipoisson(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 t value Pr(>|t|)
## (Intercept)  3.69196    0.09374  39.384  < 2e-16 ***
## woolB       -0.20599    0.10646  -1.935 0.058673 .
## tensionM    -0.32132    0.12441  -2.583 0.012775 *
## tensionH    -0.51849    0.13203  -3.927 0.000264 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 4.261537)
##
##     Null deviance: 297.37  on 53  degrees of freedom
## Residual deviance: 210.39  on 50  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

8.6 Comparando los modelos:

Ahora que tenemos dos modelos diferentes, comparémoslos para ver cuál es mejor. Primero, instalaremos la biblioteca `arm` porque contiene una función que necesitamos:

```
# install.packages("arm")

# load library arm that contains the function se.coef()
library(arm)
```

```
## Loading required package: lme4
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

```
##
## Attaching package: 'lme4'
```

```
## The following object is masked from 'package:mosaic':
##
##     factorize

##
## arm (Version 1.11-2, built: 2020-7-27)

## Working directory is /home/jaime/Escritorio/git/github/para-glm-curso/bookglm

##
## Attaching package: 'arm'

## The following object is masked from 'package:car':
##
##     logit

## The following objects are masked from 'package:mosaic':
##
##     logit, rescale

## The following objects are masked from 'package:psych':
##
##     logit, rescale, sim
```

Ahora usaremos esa funciónse.coef () para extraer los coeficientes de cada modelo, y luego usaremoscbind ()para combinar esos valores extraídos en un solo marco de datos para poder compararlos.

```
 #extract coefficients from first model using 'coef()'
coef1 = coef(poisson.model)

# extract coefficients from second model
coef2 = coef(poisson.model2)

# extract standard errors from first model using 'se.coef()'
se.coef1 = se.coef(poisson.model)

# extract standard errors from second model
se.coef2 = se.coef(poisson.model2)

# use 'cbind()' to combine values into one dataframe
models.both<-cbind(coef1, se.coef1, coef2, se.coef2, exponent = exp(coef1))

# show dataframe
models.both
```

```
##                  coef1   se.coef1      coef2   se.coef2   exponent
## (Intercept)  3.6919631 0.04541069  3.6919631 0.09374352 40.1235380
## woolB       -0.2059884 0.05157117 -0.2059884 0.10646089  0.8138425
## tensionM    -0.3213204 0.06026580 -0.3213204 0.12440965  0.7251908
## tensionH    -0.5184885 0.06395944 -0.5184885 0.13203462  0.5954198
```

En el resultado anterior, podemos ver que los coeficientes son los mismos, pero los errores estándar son diferentes.

Teniendo en cuenta estos puntos, veamos la estimación de la lana. Su valor es -0,2059884 y el exponente de -0,2059884 es 0,8138425.

```
1-0.8138425
```

```
## [1] 0.1861575
```

Esto muestra que cambiar de lana tipo A a lana tipo B da como resultado una disminución en las roturas de 0.8138425 veces la intersección, porque la estimación -0.2059884 es negativa. Otra forma de decir esto es que si cambiamos el tipo de lana de A a B, el número de roturas caerá en un 18.6% asumiendo que todas las demás variables son iguales.

Chapter 9

Predecir a partir del modelo

Una vez que se crea el modelo, podemos usar `predict (model, data, type)`para predecir resultados usando nuevos marcos de datos que contienen datos distintos a los de entrenamiento. Veamos un ejemplo.

```
# make a dataframe with new data
newdata = data.frame(wool = "B", tension = "M")

# use 'predict()' to run model on new data

predict(poisson.model2, newdata = newdata, type = "response")
```

```
##        1
## 23.68056
```

Nuestro modelo predice que habrá aproximadamente 24 roturas con lana tipo B y nivel de tensión M.

Chapter 10

Visualización de hallazgos usando jtools

Cuando comparte su análisis con otras personas, las tablas a menudo no son la mejor manera de captar la atención de las personas. Los diagramas y gráficos ayudan a las personas a comprender sus hallazgos más rápidamente. La forma más popular de visualizar datos en R es probablemente `ggplot2` (que se enseña en el curso de visualización de datos de Dataquest), también usaremos un paquete de R increíble llamado `jtools`que incluye herramientas para resumir y visualizar específicamente modelos de regresión. Usemos`jtools` para visualizar `poisson.model2`

```
#install.packages("jtools")

# you may be asked to install 'broom' and 'ggstance' packages as well
#install.packages("broom")
#install.packages("ggstance")
```

`jtools`proporciona `plot_summs ()` y `plot_coefs ()` para visualizar el resumen del modelo y también nos permite comparar diferentes modelos con `ggplot2`.

.

```
# Include jtools library
library(jtools)
```

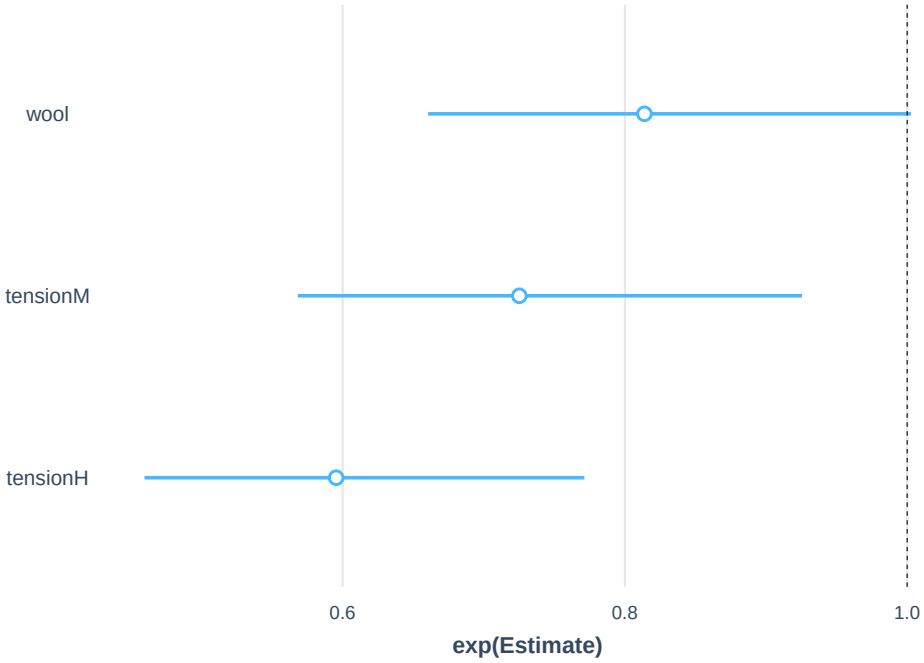
```
##
## Attaching package: 'jtools'
```

```
## The following object is masked from 'package:arm':
##
##     standardize
```

```
# plot regression coefficients for poisson.model2
plot_summs(poisson.model2, scale = TRUE, exp = TRUE)
```

```
## Note: Pseudo-R2 for quasibinomial/quasipoisson families is calculated by
## refitting the fitted and null models as binomial/poisson.
```

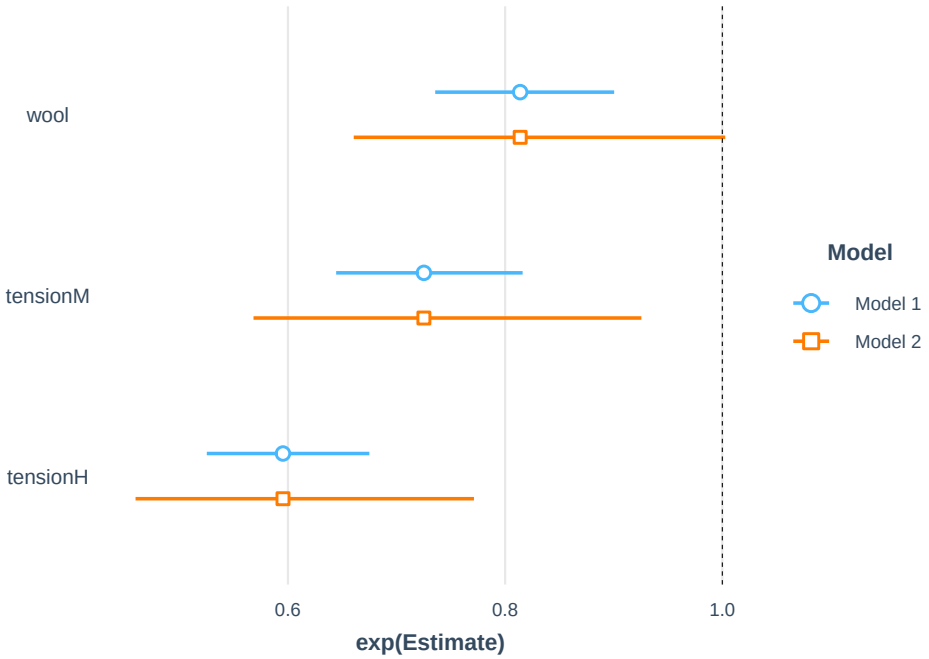
```
## Loading required namespace: broom.mixed
```




```
# plot regression coefficients for poisson.model2 and poisson.model
plot_summs(poisson.model, poisson.model2, scale = TRUE, exp = TRUE)
```

```
## Note: Pseudo-R2 for quasibinomial/quasipoisson families is calculated by
## refitting the fitted and null models as binomial/poisson.
```

```
## Loading required namespace: broom.mixed
## Loading required namespace: broom.mixed
```



En el código anterior, `plot_summs (poisson.model2, scale = TRUE, exp = TRUE)`traza el segundo modelo usando la familia cuasi-poisson en `glm`.

El primer argumento en `plot_summs ()`es el modelo de regresión que se utilizará, puede ser uno o más de uno.

scale ayuda con el problema de las diferentes escalas de las variables.`exp` se establece en `TRUE` porque para la regresión de Poisson es más probable que nos interesen los valores exponenciales de las estimaciones en lugar de los lineales.

Puede encontrar más detalles sobre jtools y plot_summs () aquí en la documentación.

También podemos visualizar la interacción entre variables predictoras.`jtools` proporciona diferentes funciones para diferentes tipos de variables. Por ejemplo, si todas las variables son categóricas, podríamos usar`cat_plot ()`para comprender mejor las interacciones entre ellas. Para variables continuas, se usa `interact_plot ()`.

En los datos de warpbreaks tenemos variables predictoras categóricas, por lo que usaremos `cat_plot ()`para visualizar la interacción entre ellas, dándole argumentos que especifiquen qué modelo nos gustaría usar, la variable predictora que estamos viendo y la otra variable predictora con la que se combina para producir el resultado.

```
library(broom)
library(jtools)
```

```
library(ggstance)
#interact_plot(poisson.model2,pred = wool,modx =tension)
# argument 1: regression model
# pred: The categorical variable that will appear on x-axis
# modx: Moderator variable that has an effect in combination to pred on outcome
```

```
library(jtools)
#install.packages("broom")
#install.packages("ggstance")
#cat_plot(poisson.model2,pred = wool,modx = tension)
# argument 1: regression model
# pred: The categorical variable that will appear on x-axis
# modx: Moderator variable that has an effect in combination to pred on outcom
```

```
#cat_plot(poisson.model2, pred=tension, modx = wool, #geom = "line")
```

Chapter 11

Modelado de regresión de Poisson utilizando datos de tasas

Hasta ahora, en este tutorial, hemos modelado datos de recuento, pero también podemos modelar datos de tasa que predicen el número de recuentos durante un período de tiempo o agrupación. La fórmula para modelar datos de tasa viene dada por:

$$\log(\frac{X}{n}) = \beta_0 + \sum \beta_i X_i$$

esto es equivalente a:

$$\log(X) - \log(n) = \beta_0 + \sum \beta_i X_i$$

$$\log(X) = \log(n) + \beta_0 + \sum \beta_i X_i$$

Por lo tanto, los datos de tasa se pueden modelar incluyendo el término log(n) con un coeficiente de 1.

Esto se denomina compensación. Este desplazamiento se modela con `offset()`en R.

Usemos otro conjunto de datos llamado **eba1977** del paquete **ISwR** para modelar el modelo de regresión de Poisson para datos de tasas. Primero, instalaremos el paquete:

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

Ahora, echemos un vistazo a algunos detalles sobre los datos e imprimamos las primeras diez filas para tener una idea de lo que incluye el conjunto de datos.

```
data(eba1977)
cancer.data = eba1977
cancer.data[1:10, ]
```

```
##          city   age  pop cases
## 1  Fredericia 40-54 3059    11
## 2     Horsens 40-54 2879    13
## 3     Kolding 40-54 3142     4
## 4       Vejle 40-54 2520     5
## 5  Fredericia 55-59  800    11
## 6     Horsens 55-59 1083     6
## 7     Kolding 55-59 1050     8
## 8       Vejle 55-59  878     7
## 9  Fredericia 60-64  710    11
## 10    Horsens 60-64  923    15
```

```
# Description
# Lung cancer incidence in four Danish cities 1968-1971
# Description:
# This data set contains counts of incident lung cancer cases and
# population size in four neighbouring Danish cities by age group.
# Format:
# A data frame with 24 observations on the following 4 variables:
# city a factor with levels Fredericia, Horsens, Kolding, and Vejle.
# age a factor with levels 40-54, 55-59, 60-64, 65-69,70-74, and 75+.
# pop a numeric vector, number of inhabitants.
# cases a numeric vector, number of lung cancer cases.
```

Para modelar datos de tasa, usamos $\frac{X}{n}$ donde X es el evento que sucederá y n es la agrupación. En este ejemplo, X = casos (el evento es un caso de cáncer) y n = pop (la población es la agrupación).

Como en la fórmula anterior, los datos de tasa se contabilizan mediante log (n) y en estos datos n es la población, por lo que primero encontraremos el log de la población. Podemos modelar para casos / población de la siguiente manera:

```
# find the log(n) of each value in 'pop' column. It is the third column

logpop = log(cancer.data[ ,3])

# add the log values to the dataframe using 'cbind()'
```

```
new.cancer.data = cbind(cancer.data, logpop)

# display new dataframe
new.cancer.data
```

```
##          city   age  pop cases   logpop
## 1  Fredericia 40-54 3059    11 8.025843
## 2     Horsens 40-54 2879    13 7.965198
## 3     Kolding 40-54 3142     4 8.052615
## 4       Vejle 40-54 2520     5 7.832014
## 5  Fredericia 55-59  800    11 6.684612
## 6     Horsens 55-59 1083     6 6.987490
## 7     Kolding 55-59 1050     8 6.956545
## 8       Vejle 55-59  878     7 6.777647
## 9  Fredericia 60-64  710    11 6.565265
## 10    Horsens 60-64  923    15 6.827629
## 11    Kolding 60-64  895     7 6.796824
## 12      Vejle 60-64  839    10 6.732211
## 13 Fredericia 65-69  581    10 6.364751
## 14    Horsens 65-69  834    10 6.726233
## 15    Kolding 65-69  702    11 6.553933
## 16      Vejle 65-69  631    14 6.447306
## 17 Fredericia 70-74  509    11 6.232448
## 18    Horsens 70-74  634    12 6.452049
## 19    Kolding 70-74  535     9 6.282267
## 20      Vejle 70-74  539     8 6.289716
## 21 Fredericia   75+  605    10 6.405228
## 22    Horsens   75+  782     2 6.661855
## 23    Kolding   75+  659    12 6.490724
## 24      Vejle   75+  619     7 6.428105
```

Ahora, modelemos los datos de la tasa con **offset()**

```
poisson.model.rate<-glm(cases ~ city + age+ offset(logpop), family = poisson(link = "log"), data

#display summary
summary(poisson.model.rate)
```

```
##
## Call:
## glm(formula = cases ~ city + age + offset(logpop), family = poisson(link = "log"),
##     data = cancer.data)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -2.63573  -0.67296  -0.03436   0.37258   1.85267
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.6321     0.2003 -28.125  < 2e-16 ***
## cityHorsens  -0.3301     0.1815  -1.818   0.0690 .
## cityKolding  -0.3715     0.1878  -1.978   0.0479 *
## cityVejle    -0.2723     0.1879  -1.450   0.1472
## age55-59      1.1010     0.2483   4.434 9.23e-06 ***
## age60-64      1.5186     0.2316   6.556 5.53e-11 ***
## age65-69      1.7677     0.2294   7.704 1.31e-14 ***
## age70-74      1.8569     0.2353   7.891 3.00e-15 ***
## age75+        1.4197     0.2503   5.672 1.41e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 129.908  on 23  degrees of freedom
## Residual deviance:  23.447  on 15  degrees of freedom
## AIC: 137.84
##
## Number of Fisher Scoring iterations: 5
```

En este conjunto de datos, podemos ver que la desviación residual está cerca de los grados de libertad y el parámetro de dispersión es 1.5 (23.447 / 15) que es pequeño, por lo que el modelo es un buen ajuste.

Usamos `fitted(model)` para devolver valores ajustados por el modelo. Devuelve resultados utilizando los datos de entrenamiento sobre los que se construye el modelo. Hagamos un intento:

```
fitted(poisson.model.rate)
```

```
##         1         2         3         4         5         6         7         8
## 10.954812  7.411803  7.760169  6.873215  8.615485  8.384458  7.798635  7.201421
##         9        10        11        12        13        14        15        16
## 11.609373 10.849479 10.092831 10.448316 12.187276 12.576313 10.155638 10.080773
##        17        18        19        20        21        22        23        24
## 11.672630 10.451942  8.461440  9.413988  8.960422  8.326004  6.731286  6.982287
```

Usando este modelo, podemos predecir el número de casos por 1000 habitantes para un nuevo conjunto de datos, usando la función `predict (),` de manera muy similar a como lo hicimos para nuestro modelo de conteo de datos anteriormente:

```r
# create a test dataframe containing new values of variables
test.data = data.frame(city = "Kolding", age = "40-54", pop = 1000, logpop = log(1000))

# predict outcomes (responses) using 'predict()'
predicted.value<-predict(poisson.model.rate, test.data, type = "response")

# show predicted value
predicted.value
```

```
##        1
## 2.469818
```

Entonces, para la ciudad de Kolding entre las personas en el grupo de edad de 40 a 54 años, podríamos esperar aproximadamente 2 o 3 casos de cáncer de pulmón por cada 1000 personas.

Chapter 12

Conclusión

Los modelos de regresión de Poisson tienen una gran importancia en las predicciones econométricas y del mundo real. En este tutorial, hemos aprendido sobre la distribución de Poisson, los modelos lineales generalizados y los modelos de regresión de Poisson.

También aprendimos cómo implementar modelos de regresión de Poisson para datos de recuento y tasa en R usando glm (), y cómo ajustar los datos al modelo para predecir un nuevo conjunto de datos. Además, analizamos cómo obtener errores estándar más precisos en glm () usando quasipoisson y vimos algunas de las posibilidades disponibles para la visualización con jtools

Chapter 13

Variables Binarias y regresión Logística.

13.1 Distribuciones de probabilidad..

En este capítulo consideramos modelos lineales generalizados en los que las variables de resultado se miden en una escala binaria. Por ejemplo, las respuestas pueden estar vivas o muertas, presentes o ausentes. El éxito y el fracaso se utilizan como términos genéricos de las dos categorías. Primero, definimos la variable aleatoria binaria

$$Z = \begin{cases} 1 & \text{si el resultado es un éxito} \\ 0 & \text{si el resultado es un fracaso} \end{cases}$$

con probabilidades $\Pr(Z = 1) = \pi$ y $\Pr(Z = 0) = 1 - \pi$, que es la distribución de Bernoulli B(PS Si hay n tales variables aleatorias $Z_1, \ldots, Z_n$, que son independientes con $\Pr\left(Z_j = 1\right) = \pi_j$, entonces su probabilidad conjunta es

$$\prod_{j=1}^{n} \pi_j^{z_j} \left(1 - \pi_j\right)^{1-z_j} = \exp\left[\sum_{j=1}^{n} z_j \log\left(\frac{\pi_j}{1-\pi_j}\right) + \sum_{j=1}^{n} \log\left(1 - \pi_j\right)\right] \tag{13.1}$$

que es un miembro de la familia exponencial.

A continuación, para el caso en el que los π_j son todos iguales, podemos definir

$$Y = \sum_{j=1}^{n} Z_j$$

de modo que Y es el número de éxitos en n "ensayos". La variable aleatoria Y tiene la distribución $\text{Bin}(n, \pi)$

$$\Pr(Y = y) = \left(\begin{array}{c} n \\ y \end{array} \right) \pi^y (1 - \pi)^{n-y}, \quad y = 0, 1, \ldots, n \tag{13.2}$$

Finalmente, consideramos el caso general de N variables aleatorias independientes $Y_1, Y_2, \ldots, Y_N$ correspondientes al número de éxitos en N diferentes subgrupos o estratos (en la siguiente tabla). Si $Y_i \sim \text{Bin}(n_i, \pi_i)$, la función de probabilidad de registro es

$$\begin{array}{l} l(\pi_1, \ldots, \pi_N; y_1, \ldots, y_N) \\ = \sum_{i=1}^{N} \left[y_i \log\left(\frac{\pi_i}{1-\pi_i}\right) + n_i \log(1-\pi_i) + \log \left(\begin{array}{c} n_i \\ y_i \end{array} \right) \right] \end{array}$$

poner aqui una tabla como imagen dobson

13.2 Modelos lineales generalizados.

Queremos describir la proporción de éxitos, $P_i = \dfrac{Y_i}{n_i}$, en cada subgrupo en términos de niveles de factores y otras variables explicativas que caracterizan al subgrupo. Como $\text{E}(Y_i) = n_i \pi_i$ y así $\text{E}(P_i) = \pi_i$, modelamos las probabilidades π_i como

$$g(\pi_i) = \mathbf{x}_i^T \beta$$

donde $\mathbf{x}_i$ es un vector de variables explicativas(variables ficticias para niveles de factor y valores medidos para covariables), β es un vector de parámetros y g es un función de enlace. El caso más simple es el modelo lineal.

$$\pi = \mathbf{x}^T \beta$$

Esto se usa en algunas aplicaciones prácticas, pero tiene la desventaja de que aunque π es una probabilidad, los valores ajustados $\mathbf{x}^T \mathbf{b}$ pueden ser menores que cero o mayores que uno.

Para garantizar que π esté restringido al intervalo $[0, 1]$, a menudo se modela utilizando una distribución de probabilidad acumulativa

$$\pi = \int_{-\infty}^{t} f(s) ds$$

donde $f(s) \geqslant 0$ y $\int_{-\infty}^{\infty}(s) ds = 1$. La función de densidad de probabilidad $f(s)$ se denomina distribución de tolerancia.

13.3 Modelos de respuesta a la dosis.

Históricamente, uno de los primeros usos de modelos de regresión para datos binomiales fue para los resultados de bioensayos (Finney 1973). Las respuestas fueron las proporciones o porcentajes de "éxitos"; por ejemplo, la proporción de animales de experimentación muertos por distintos niveles de dosis de una

sustancia tóxica. A veces, estos datos se denominan respuestas cuánticas. El objetivo es describir la probabilidad de "éxito", π, en función de la dosis, x; por ejemplo, $g(\pi) = \beta_1 + \beta_2 x$. Si la distribución de tolerancia $f(s)$ es la distribución uniforme en el intervalo $[c_1, c_2]$

$$f(s) = \begin{cases} \frac{1}{c_2 - c_1} & \text{if} c_1 \leqslant s \leqslant c_2 \\ 0 & \text{de lo contrario} \end{cases}$$

entonces π es acumulativo

$$\pi = \int_{c_1}^{x} f(s)ds = \frac{x - c_1}{c_2 - c_1} \quad \text{para} \quad c_1 \leqslant x \leqslant c_2$$

(ver figura 7.1). Esta ecuación tiene la forma $\pi = \beta_1 + \beta_2 x$, donde

$$\beta_1 = \frac{-c_1}{c_2 - c_1} \quad \text{y} \quad \beta_2 = \frac{1}{c_2 - c_1}$$

Chapter 14

Final Words

We have finished a nice book.

Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2021). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.22.