Poisson regression

Thomas Achia

6/19/2023

## Poisson regression

Example 1

The Hypothetical problem with the following count data:

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.2.0  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

y<-c(2,3,6,7,8,9,10,12,15) # The count outcome  
x<-c(-1,-1,0,0,0,0,1,1,1)# Explanatory variable  
  
data001<-tibble(x=x,y=y)

We start with an Incorrect analysis: Fitting a simple linear regression, a linear regression model, a General linear mdels

fit<-lm(y~x)  
summary(fit)

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.3636 -0.5454 -0.3636 0.5454 2.6364   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.4545 0.5433 13.721 2.57e-06 \*\*\*  
## x 4.9091 0.7289 6.735 0.000269 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.612 on 7 degrees of freedom  
## Multiple R-squared: 0.8663, Adjusted R-squared: 0.8472   
## F-statistic: 45.36 on 1 and 7 DF, p-value: 0.0002687

The line of best fit is

Employing the model when x=-2 we get

**The correct analysis**: Assume that a Generalized linear model, we fit a GLM assumng the Poisson family with a natural log link function. That is,

fit1<-glm(y~x,family = poisson(link = log))  
summary(fit1)

##   
## Call:  
## glm(formula = y ~ x, family = poisson(link = log))  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.8472 -0.2601 -0.2137 0.5214 0.8788   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 1.8893 0.1421 13.294 < 2e-16 \*\*\*  
## x 0.6698 0.1787 3.748 0.000178 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 18.4206 on 8 degrees of freedom  
## Residual deviance: 2.9387 on 7 degrees of freedom  
## AIC: 41.052  
##   
## Number of Fisher Scoring iterations: 4

In the example,

If x=-2

Example 2. COVID i relation to HIV

Region<-c(1,2,3,4)# Regions  
Region<-as.factor(Region)  
n<-c(100000,50000,250000,10000)# Population at risk  
y<-c(500,300,150,100) # Incident COVID cases  
phiv=c(0.03,0.1,0.2,0.2) # HIV prevalence   
  
datacovid<-tibble(Region=Region,n=n,y=y,phiv=phiv)  
datacovid

## # A tibble: 4 × 4  
## Region n y phiv  
## <fct> <dbl> <dbl> <dbl>  
## 1 1 100000 500 0.03  
## 2 2 50000 300 0.1   
## 3 3 250000 150 0.2   
## 4 4 10000 100 0.2

The question one may ask include:

* Are the region difference in COVID risk
* Is COVID risk associated with HIV positive

Let be the number of COVID in region i and the population in refgion i The COVID positivity rate

We asssume that

fit2<-glm(y~phiv,data = datacovid,family = poisson(link = log),offset = log(n))  
summary(fit2)

##   
## Call:  
## glm(formula = y ~ phiv, family = poisson(link = log), data = datacovid,   
## offset = log(n))  
##   
## Deviance Residuals:   
## 1 2 3 4   
## -3.679 10.545 -9.534 15.771   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.8574 0.0484 -100.36 <2e-16 \*\*\*  
## phiv -9.3598 0.4121 -22.71 <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: 1016.95 on 3 degrees of freedom  
## Residual deviance: 464.36 on 2 degrees of freedom  
## AIC: 497.25  
##   
## Number of Fisher Scoring iterations: 5

An incorrect analysis would be

fit20<-glm(y~phiv,data = datacovid,family = poisson(link = log))  
summary(fit20)

##   
## Call:  
## glm(formula = y ~ phiv, family = poisson(link = log), data = datacovid)  
##   
## Deviance Residuals:   
## 1 2 3 4   
## -0.3297 0.7377 1.9295 -2.5399   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 6.47293 0.05107 126.74 <2e-16 \*\*\*  
## phiv -8.12041 0.44979 -18.05 <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: 363.575 on 3 degrees of freedom  
## Residual deviance: 10.827 on 2 degrees of freedom  
## AIC: 43.716  
##   
## Number of Fisher Scoring iterations: 4

**Example 3**. Time to recovery for TB patients between treatment and placebo

treat<-c("treat","placebo")  
n<-c(15000,10000)  
t<-c(5000,4000)  
y<-c(100,30)  
  
datatb<-tibble(treat=treat,n=n,t=t,y=y)  
datatb

## # A tibble: 2 × 4  
## treat n t y  
## <chr> <dbl> <dbl> <dbl>  
## 1 treat 15000 5000 100  
## 2 placebo 10000 4000 30

fit32<-glm(y~treat,data=datatb,family = poisson(link = log),offset = log(t))  
summary(fit32)

##   
## Call:  
## glm(formula = y ~ treat, family = poisson(link = log), data = datatb,   
## offset = log(t))  
##   
## Deviance Residuals:   
## [1] 0 0  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.8929 0.1826 -26.799 < 2e-16 \*\*\*  
## treattreat 0.9808 0.2082 4.712 2.46e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 2.5760e+01 on 1 degrees of freedom  
## Residual deviance: -1.7764e-15 on 0 degrees of freedom  
## AIC: 15.689  
##   
## Number of Fisher Scoring iterations: 2

Example. Consider the first example

The data:

y<-c(2,3,6,7,8,9,10,12,15) # The count outcome  
x<-c(-1,-1,0,0,0,0,1,1,1)# Explanatory variable  
  
data001<-tibble(x=x,y=y)  
  
# Develop the algorithm  
  
X=cbind(rep(1,9),x)  
X

## x  
## [1,] 1 -1  
## [2,] 1 -1  
## [3,] 1 0  
## [4,] 1 0  
## [5,] 1 0  
## [6,] 1 0  
## [7,] 1 1  
## [8,] 1 1  
## [9,] 1 1

y

## [1] 2 3 6 7 8 9 10 12 15

The first iteration with the initial guess as

beta0<-as.matrix(c(0,0))  
beta0

## [,1]  
## [1,] 0  
## [2,] 0

mu0<-exp(X%\*%beta0)  
  
  
Xtyminusm0<-t(X)%\*%(y-exp(X%\*%beta0))  
Xtyminusm0

## [,1]  
## 63  
## x 31

W0<-diag(exp(as.numeric(X%\*%beta0)))  
  
XtWoXinv<-solve(t(X)%\*%W0%\*%X)  
XtWoXinv

## x  
## 0.11363636 -0.02272727  
## x -0.02272727 0.20454545

beta1=beta0+XtWoXinv%\*%Xtyminusm0  
beta1

## [,1]  
## 6.454545  
## x 4.909091

For the second iteration, with

mu1<-exp(X%\*%beta1)  
  
  
Xtyminusm1<-t(X)%\*%(y-exp(X%\*%beta1))  
Xtyminusm1

## [,1]  
## -260875.7  
## x -258354.6

W1<-diag(exp(as.numeric(X%\*%beta1)))  
  
XtW1Xinv<-solve(t(X)%\*%W1%\*%X)  
XtW1Xinv

## x  
## 0.0003876182 -0.0003875900  
## x -0.0003875900 0.0003914318

beta2=beta1+XtW1Xinv%\*%Xtyminusm1  
beta2

## [,1]  
## 5.470051  
## x 4.893710