

# Poisson Regression

A Short Course on Data Analysis Using R Software

Wan Nor Arifin

Unit of Biostatistics and Research Methodology,  
Universiti Sains Malaysia.

*Email: wnarifin@usm.my*

*Website: wnarifin.github.io*



©Wan Nor Arifin under the Creative Commons Attribution-ShareAlike 4.0 International License.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Preliminaries</b>	<b>2</b>
2.1	Load libraries . . . . .	2
<b>3</b>	<b>Simple Poisson regression models</b>	<b>2</b>
3.1	Count data . . . . .	2
3.1.1	X categorical . . . . .	2
3.1.2	X numerical . . . . .	4
3.2	Rate data . . . . .	7
<b>4</b>	<b>Multiple Poisson regression model</b>	<b>9</b>
4.1	Univariable . . . . .	10
4.2	Multivariable . . . . .	12
4.3	Interaction . . . . .	14
4.4	Final model . . . . .	14
	<b>References</b>	<b>15</b>

## 1 Introduction

Multiple Poisson Regression for count is given as

$$\ln E(Y|\mathbf{X}) = \ln \mu = \beta_0 + \beta_1 X_1 + \cdots + \beta_{p-1} X_{p-1} = \beta_0 + \sum \beta_{p-1} X_{p-1}$$

where the  $\mathbf{X}$  (in bold) denotes a collection of Xs.  $p$  is the number of estimated parameters.

Multiple Poisson Regression for rate with offset<sup>1</sup> is given as

$$\ln E(Y|\mathbf{X}) = \ln a(\mathbf{X}) + \beta_0 + \sum \beta_{p-1} X_{p-1}$$

The rate ratio, RR is

$$RR = e^{\beta_{p-1}}$$

---

<sup>1</sup>the ln of the denominator/person-years,  $a(\mathbf{X})$

## 2 Preliminaries

### 2.1 Load libraries

```
library(epiDisplay)
library(car)
```

## 3 Simple Poisson regression models

### 3.1 Count data

#### 3.1.1 X categorical

```
# - UKaccident.csv is modified from builtin data Seatbelts
acc = read.csv("UKaccident.csv")
#- driverskilled: number of death
#- law: before seatbelt law = 0, after law = 1
str(acc)
```

```
## 'data.frame': 122 obs. of 2 variables:
## $ driverskilled: int 107 97 102 87 119 106 110 106 107 125 ...
## $ law : int 0 0 0 0 0 0 0 0 0 0 ...
```

```
head(acc); tail(acc)
```

```
## driverskilled law
## 1 107 0
## 2 97 0
## 3 102 0
## 4 87 0
## 5 119 0
## 6 106 0

## driverskilled law
## 117 81 1
## 118 84 1
## 119 87 1
## 120 90 1
## 121 79 1
## 122 96 1
```

```
# - some descriptives
tapply(acc$driverskilled, acc$law, sum) # total death before vs after
```

```
## 0 1
## 11826 1294
```

```
table(acc$law) # num of observations before vs after
```

```
##
## 0 1
## 107 15
```

```

# - mean count, manually
11826/107 # 110.5234, count before law

## [1] 110.5234
1294/15 # 86.26667, count after law

## [1] 86.26667
model.acc = glm(driverskilled ~ law, data = acc, family = poisson)
summary(model.acc) # significant p based on Wald test

##
## Call:
## glm(formula = driverskilled ~ law, family = poisson, data = acc)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.16127 -0.72398  0.04531  0.77308  1.89182
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.705227   0.009196  511.681  <2e-16 ***
## law          -0.247784   0.029281  -8.462   <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: 219.17  on 121  degrees of freedom
## Residual deviance: 142.64  on 120  degrees of freedom
## AIC: 940.7
##
## Number of Fisher Scoring iterations: 4
# - to get CI
cbind(coef(model.acc), confint(model.acc))

## Waiting for profiling to be done...
##              2.5 %    97.5 %
## (Intercept)  4.7052269  4.6871495  4.7231960
## law          -0.2477837 -0.3056189 -0.1908312
# - ln(count) = 4.71 - 0.25*LAW
4.71 - 0.25 # = 4.46

## [1] 4.46
exp(4.71) # 111.0522, count before law

## [1] 111.0522
exp(4.46) # 86.48751, count after law

## [1] 86.48751
# - Model fit
poisgof(model.acc) # fit well, based on chi-square test on the residual deviance

```

```

## $results
## [1] "Goodness-of-fit test for Poisson assumption"
##
## $chisq
## [1] 142.6436
##
## $df
## [1] 120
##
## $p.value
## [1] 0.07764771

# - Diagnostics
# - standardized residuals
sr = rstandard(model.acc)
sr[abs(sr) > 1.96]

##          4          54          55          91          113
## -2.335861 -3.176147 -2.857937 -2.647896 -3.098644

# - predicted count vs fitted values
fitted.acc = model.acc$fitted
data.frame(acc, fitted.acc)[names(sr[abs(sr) > 1.96]),] # look at the discrepancies

##      driverskilled law fitted.acc
## 4                87  0 110.52336
## 54                79  0 110.52336
## 55                82  0 110.52336
## 91                84  0 110.52336
## 113               60  1  86.26667

# Summary with RR
idr.display(model.acc) # easier, also view LR test

##
## Poisson regression predicting driverskilled
##
##              IDR(95%CI)      P(Wald's test) P(LR-test)
## law: 1 vs 0  0.78 (0.74,0.83) < 0.001      < 0.001
##
## Log-likelihood = -468.3481
## No. of observations = 122
## AIC value = 940.6963

```

### 3.1.2 X numerical

```

# - Data from https://stats.idre.ucla.edu/stat/data/poisson_sim.csv
aw = read.csv("poisson_sim.csv")
head(aw); tail(aw)

##   id num_awards prog math
## 1  45           0   3   41
## 2 108           0   1   41
## 3  15           0   3   44
## 4  67           0   3   42
## 5 153           0   3   40

```

```

## 6 51      0  1  42
##      id num_awards prog math
## 195 61      1  2  60
## 196 100     2  2  71
## 197 143     2  3  75
## 198 68      1  2  71
## 199 57      0  2  72
## 200 132     3  2  73
str(aw)

## 'data.frame':  200 obs. of  4 variables:
## $ id      : int  45 108 15 67 153 51 164 133 2 53 ...
## $ num_awards: int  0 0 0 0 0 0 0 0 0 0 ...
## $ prog     : int  3 1 3 3 3 1 3 3 3 3 ...
## $ math     : int  41 41 44 42 40 42 46 40 33 46 ...
##- num_awards: The number of awards earned by students at one high school.
##- math: the score on their final exam in math.
model.aw = glm(num_awards ~ math, data = aw, family = poisson)
summary(model.aw) # math sig.

##
## Call:
## glm(formula = num_awards ~ math, family = poisson, data = aw)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1853  -0.9070  -0.6001   0.3246   2.9529
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.333532   0.591261  -9.021  <2e-16 ***
## math         0.086166   0.009679   8.902  <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: 287.67  on 199  degrees of freedom
## Residual deviance: 204.02  on 198  degrees of freedom
## AIC: 384.08
##
## Number of Fisher Scoring iterations: 6
cbind(coef(model.aw), confint(model.aw))

## Waiting for profiling to be done...
##
##              2.5 %   97.5 %
## (Intercept) -5.3335321 -6.52038334 -4.200322
## math         0.0861656  0.06737466  0.105356
poisgof(model.aw) # fit well

## $results
## [1] "Goodness-of-fit test for Poisson assumption"

```

```

##
## $chisq
## [1] 204.0213
##
## $df
## [1] 198
##
## $p.value
## [1] 0.3695697
sr = rstandard(model.aw)
sr[abs(sr) > 1.96]

##          54          120          122          150          157          164          172          181          199
## 2.740294 1.975409 2.015236 2.112331 2.963862 2.253872 2.112331 2.451774 -2.241058
aw_ = data.frame(aw[c(4,2)], predicted = model.aw$fitted); head(aw_); tail(aw_)

##  math num_awards predicted
## 1   41             0 0.1651762
## 2   41             0 0.1651762
## 3   44             0 0.2139002
## 4   42             0 0.1800399
## 5   40             0 0.1515396
## 6   42             0 0.1800399

##  math num_awards predicted
## 195  60             1 0.8490848
## 196  71             2 2.1907094
## 197  75             2 3.0922155
## 198  71             1 2.1907094
## 199  72             0 2.3878444
## 200  73             3 2.6027189
aw_[names(sr[abs(sr) > 1.96]),] # look at the discrepancies

##  math num_awards predicted
## 54   50             3 0.3587060
## 120  49             2 0.3290921
## 122  58             3 0.7146750
## 150  57             3 0.6556731
## 157  61             5 0.9254913
## 164  62             4 1.0087733
## 172  57             3 0.6556731
## 181  69             6 1.8439209
## 199  72             0 2.3878444

# 1 unit increase in math score
idr.display(model.aw)

##
## Poisson regression predicting num_awards
##
##              IDR(95%CI)      P(Wald's test) P(LR-test)
## math (cont. var.) 1.09 (1.07,1.11) < 0.001      < 0.001
##
## Log-likelihood = -190.0381
## No. of observations = 200

```

```
## AIC value = 384.0762
# 10 unit increase in math score? Manually...
b1 = coef(model.aw)[[2]]*10
b1.ll = confint(model.aw)[[2]]*10

## Waiting for profiling to be done...
b1.ul = confint(model.aw)[[4]]*10

## Waiting for profiling to be done...
exp(cbind("Math RR" = b1, "95% LL" = b1.ll, "95% UL" = b1.ul))

##      Math RR    95% LL    95% UL
## [1,] 2.367077 1.961573 2.867842
```

### 3.2 Rate data

```
# - data in Fleiss et al 2003
" Table 12.1
  cigar.day person.yrs cases      rate      pred
1      0.0      1421      0 0.000000000 0.000793326
2      5.2      927      0 0.000000000 0.001170787
3     11.2      988      2 0.002024291 0.001834458
4     15.9      849      2 0.002355713 0.002607843
5     20.4     1567      9 0.005743459 0.003652195
6     27.4     1409     10 0.007097232 0.006167215
7     40.8      556      7 0.012589928 0.016813428
"
```

```
## [1] " Table 12.1\n  cigar.day person.yrs cases      rate      pred\n1      0.0      1421      0 0.000000000 0.000793326\n2      5.2      927      0 0.000000000 0.001170787\n3     11.2      988      2 0.002024291 0.001834458\n4     15.9      849      2 0.002355713 0.002607843\n5     20.4     1567      9 0.005743459 0.003652195\n6     27.4     1409     10 0.007097232 0.006167215\n7     40.8      556      7 0.012589928 0.016813428\n"
```

```
##  cigar.day person.yrs cases
## 1      0.0      1421      0
## 2      5.2      927      0
## 3     11.2      988      2
## 4     15.9      849      2
## 5     20.4     1567      9
## 6     27.4     1409     10
## 7     40.8      556      7
```

```
cig$rate = cig$cases/cig$person.yrs; cig
```

```
##  cigar.day person.yrs cases      rate
## 1      0.0      1421      0 0.000000000
## 2      5.2      927      0 0.000000000
## 3     11.2      988      2 0.002024291
## 4     15.9      849      2 0.002355713
## 5     20.4     1567      9 0.005743459
## 6     27.4     1409     10 0.007097232
## 7     40.8      556      7 0.012589928
```

```

model.cig = glm(cases ~ cigar.day, offset = log(person.yrs),
               data = cig, family = "poisson")
# - it includes offset variable
summary(model.cig)

##
## Call:
## glm(formula = cases ~ cigar.day, family = "poisson", data = cig,
##      offset = log(person.yrs))
##
## Deviance Residuals:
##      1       2       3       4       5       6       7
## -1.5015 -1.4733  0.1370 -0.1463  1.2630  0.4340 -0.8041
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.13928    0.45402 -15.725 < 2e-16 ***
## cigar.day    0.07485    0.01564  4.786 1.7e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 30.9017 on 6 degrees of freedom
## Residual deviance: 6.8956 on 5 degrees of freedom
## AIC: 28.141
##
## Number of Fisher Scoring iterations: 5

```

```
poisgof(model.cig)
```

```

## $results
## [1] "Goodness-of-fit test for Poisson assumption"
##
## $chisq
## [1] 6.895581
##
## $df
## [1] 5
##
## $p.value
## [1] 0.2285227

```

```
cig$pred = model.cig$fitted/cig$person.yrs; cig
```

```

##   cigar.day person.yrs cases      rate      pred
## 1      0.0      1421      0 0.000000000 0.000793326
## 2      5.2       927      0 0.000000000 0.001170787
## 3     11.2       988      2 0.002024291 0.001834458
## 4     15.9       849      2 0.002355713 0.002607843
## 5     20.4      1567      9 0.005743459 0.003652195
## 6     27.4      1409     10 0.007097232 0.006167215
## 7     40.8       556      7 0.012589928 0.016813428

```

```
idr.display(model.cig) # interpret?
```



```
##
## Poisson regression predicting cases with offset = log(person.yrs)
##
##                IDR(95%CI)          P(Wald's test) P(LR-test)
## cigar.day (cont. var.) 1.08 (1.05,1.11) < 0.001      < 0.001
##
## Log-likelihood = -12.0707
## No. of observations = 7
## AIC value = 28.1413

# - 5 cigar/day
exp(coef(model.cig)[[2]]*5) # interpret?

## [1] 1.453868

# - 10 cigar/day
exp(coef(model.cig)[[2]]*10) # interpret?

## [1] 2.113733
```

## 4 Multiple Poisson regression model

```
# - Again, data from https://stats.idre.ucla.edu/stat/data/poisson_sim.csv
aw = read.csv("poisson_sim.csv")
str(aw)

## 'data.frame': 200 obs. of 4 variables:
## $ id : int 45 108 15 67 153 51 164 133 2 53 ...
## $ num_awards: int 0 0 0 0 0 0 0 0 0 0 ...
## $ prog : int 3 1 3 3 3 1 3 3 3 3 ...
## $ math : int 41 41 44 42 40 42 46 40 33 46 ...

head(aw); tail(aw)

## id num_awards prog math
## 1 45 0 3 41
## 2 108 0 1 41
## 3 15 0 3 44
## 4 67 0 3 42
## 5 153 0 3 40
## 6 51 0 1 42

## id num_awards prog math
## 195 61 1 2 60
## 196 100 2 2 71
## 197 143 2 3 75
## 198 68 1 2 71
## 199 57 0 2 72
## 200 132 3 2 73

#- num_awards: The number of awards earned by students at one high school.
#- prog: 1 = General, 2 = Academic, 3 = Vocational
#- math: the score on their final exam in math.
#- factor prog & save as a new variable prog1
aw$prog1 = factor(aw$prog, levels = 1:3, labels = c("General", "Academic", "Vocational"))
```

```
str(aw)
```

```
## 'data.frame': 200 obs. of 5 variables:
## $ id : int 45 108 15 67 153 51 164 133 2 53 ...
## $ num_awards: int 0 0 0 0 0 0 0 0 0 0 ...
## $ prog : int 3 1 3 3 3 1 3 3 3 3 ...
## $ math : int 41 41 44 42 40 42 46 40 33 46 ...
## $ prog1 : Factor w/ 3 levels "General","Academic",...: 3 1 3 3 3 1 3 3 3 3 ...
```

```
head(aw); tail(aw)
```

```
## id num_awards prog math prog1
## 1 45 0 3 41 Vocational
## 2 108 0 1 41 General
## 3 15 0 3 44 Vocational
## 4 67 0 3 42 Vocational
## 5 153 0 3 40 Vocational
## 6 51 0 1 42 General

## id num_awards prog math prog1
## 195 61 1 2 60 Academic
## 196 100 2 2 71 Academic
## 197 143 2 3 75 Vocational
## 198 68 1 2 71 Academic
## 199 57 0 2 72 Academic
## 200 132 3 2 73 Academic
```

## 4.1 Univariable

```
# - Math
model.aw.u1 = glm(num_awards ~ math, data = aw, family = poisson)
summary(model.aw.u1) # Math sig.
```

```
##
## Call:
## glm(formula = num_awards ~ math, family = poisson, data = aw)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -2.1853 -0.9070 -0.6001 0.3246 2.9529
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.333532 0.591261 -9.021 <2e-16 ***
## math 0.086166 0.009679 8.902 <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: 287.67 on 199 degrees of freedom
## Residual deviance: 204.02 on 198 degrees of freedom
## AIC: 384.08
##
```

```

## Number of Fisher Scoring iterations: 6
# - Prog
model.aw.u2 = glm(num_awards ~ prog1, data = aw, family = poisson)
summary(model.aw.u2) # Vocational vs General not sig. -> Combine

##
## Call:
## glm(formula = num_awards ~ prog1, family = poisson, data = aw)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4142  -0.6928  -0.6325   0.0000   3.3913
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.6094     0.3333  -4.828 1.38e-06 ***
## prog1Academic     1.6094     0.3473   4.634 3.59e-06 ***
## prog1Vocational   0.1823     0.4410   0.413  0.679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 234.46  on 197  degrees of freedom
## AIC: 416.51
##
## Number of Fisher Scoring iterations: 6
aw$prog2 = recode(aw$prog1, "c('General', 'Vocational') = 'General & Vocational'")
levels(aw$prog2)

## [1] "Academic"          "General & Vocational"
# - Prog2: General & Vocational vs Academic
model.aw.u2a = glm(num_awards ~ prog2, data = aw, family = poisson)
summary(model.aw.u2a)

##
## Call:
## glm(formula = num_awards ~ prog2, family = poisson, data = aw)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4142  -0.6649  -0.6649   0.0000   3.3913
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    4.352e-16  9.759e-02   0.000      1
## prog2General & Vocational -1.509e+00  2.390e-01  -6.314 2.72e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##

```

```
## Null deviance: 287.67 on 199 degrees of freedom
## Residual deviance: 234.63 on 198 degrees of freedom
## AIC: 414.69
##
## Number of Fisher Scoring iterations: 6
```

```
table(No_Award = aw$num_awards, aw$prog2)
```

```
##
## No_Award Academic General & Vocational
##      0      48      76
##      1      32      17
##      2      11       2
##      3       9       0
##      4       2       0
##      5       2       0
##      6       1       0
```

```
tapply(aw$num_awards, aw$prog2, sum)
```

```
##           Academic General & Vocational
##           105                21
```

## 4.2 Multivariable

```
model.aw.m1 = glm(num_awards ~ math + prog2, data = aw, family = poisson)
summary(model.aw.m1) # both vars sig.
```

```
##
## Call:
## glm(formula = num_awards ~ math + prog2, family = poisson, data = aw)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2020  -0.8346  -0.5115   0.2589   2.6793
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.15050    0.66781  -6.215 5.13e-10 ***
## math             0.06995    0.01068   6.548 5.83e-11 ***
## prog2General & Vocational -0.89129    0.25662  -3.473 0.000514 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 287.67 on 199 degrees of freedom
## Residual deviance: 190.16 on 197 degrees of freedom
## AIC: 372.22
##
## Number of Fisher Scoring iterations: 6
```

```
poisgof(model.aw.m1) # good fit
```

```
## $results
```

```

## [1] "Goodness-of-fit test for Poisson assumption"
##
## $chisq
## [1] 190.1611
##
## $df
## [1] 197
##
## $p.value
## [1] 0.6235879
idr.display(model.aw.m1)

##
## Poisson regression predicting num_awards
##
##                                crude IDR(95%CI)  adj. IDR(95%CI)
## math (cont. var.)                1.09 (1.07,1.11)  1.07 (1.05,1.1)
##
## prog2: General & Vocational vs Academic 0.22 (0.14,0.35)  0.41 (0.25,0.68)
##
##                                P(Wald's test) P(LR-test)
## math (cont. var.)                < 0.001      < 0.001
##
## prog2: General & Vocational vs Academic < 0.001      < 0.001
##
## Log-likelihood = -183.108
## No. of observations = 200
## AIC value = 372.216
AIC(model.aw.u1, model.aw.u2a, model.aw.m1)

##                df      AIC
## model.aw.u1    2 384.0762
## model.aw.u2a  2 414.6871
## model.aw.m1   3 372.2160
# - diagnostics
sr = rstandard(model.aw.m1)
sr[abs(sr) > 1.96]

##          54          154          157          164          181          191          199
## 2.372000  1.996023  2.693894  2.014175  2.342797 -2.013339 -2.261164
aw$pred = model.aw.m1$fitted
aw_diag = data.frame(num_of_awards = aw$num_awards, pred_awards = round(aw$pred, 1))
aw_diag[names(sr[abs(sr) > 1.96]), ] # look at the discrepancies

##      num_of_awards  pred_awards
## 54                3            0.5
## 154               2            0.3
## 157               5            1.1
## 164               4            1.2
## 181               6            2.0
## 191               0            2.0
## 199               0            2.4

```

```

# - model fit: scaled Pearson chi-square statistic
quasi = summary(glm(num_awards ~ math + prog2, data = aw, family = quasipoisson))
quasi$dispersion # dispersion parameter = scaled Pearson chi-square statistic

## [1] 1.08969
# - closer to 1, better.

```

### 4.3 Interaction

```

model.aw.i1 = glm(num_awards ~ math + prog2 + math*prog2, data = aw, family = poisson)
summary(model.aw.i1) # interaction term not sig.

```

```

##
## Call:
## glm(formula = num_awards ~ math + prog2 + math * prog2, family = poisson,
##      data = aw)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2295  -0.8162  -0.5377   0.2528   2.6826
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.30286    0.74810  -5.752 8.83e-09 ***
## math             0.07241    0.01196   6.053 1.42e-09 ***
## prog2General & Vocational -0.19552    1.50706  -0.130  0.897
## math:prog2General & Vocational -0.01277    0.02742  -0.466  0.641
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 287.67  on 199  degrees of freedom
## Residual deviance: 189.94  on 196  degrees of freedom
## AIC: 374
##
## Number of Fisher Scoring iterations: 6

```

```

AIC(model.aw.m1, model.aw.i1) # increase in AIC, M1 is better

```

```

##           df      AIC
## model.aw.m1  3 372.2160
## model.aw.i1  4 373.9965

```

### 4.4 Final model

```

# - Accept model.aw.m1
idr.display(model.aw.m1)

```

```

##
## Poisson regression predicting num_awards
##

```

```

##                               crude IDR(95%CI)  adj. IDR(95%CI)
## math (cont. var.)             1.09 (1.07,1.11)  1.07 (1.05,1.1)
##
## prog2: General & Vocational vs Academic 0.22 (0.14,0.35)  0.41 (0.25,0.68)
##
##                               P(Wald's test) P(LR-test)
## math (cont. var.)             < 0.001         < 0.001
##
## prog2: General & Vocational vs Academic < 0.001         < 0.001
##
## Log-likelihood = -183.108
## No. of observations = 200
## AIC value = 372.216

b1 = coef(model.aw.m1)[[2]]*10
b1.ll = confint(model.aw.m1)[[2]]*10

## Waiting for profiling to be done...

b1.ul = confint(model.aw.m1)[[5]]*10

## Waiting for profiling to be done...

exp(cbind("Math RR" = b1, "95% LL" = b1.ll, "95% UL" = b1.ul))

##      Math RR  95% LL  95% UL
## [1,] 2.012665 1.63494 2.485884

```

## References

- Chongsuvivatwong, V. (2018). *EpiDisplay: Epidemiological data display package*. Retrieved from <https://CRAN.R-project.org/package=epiDisplay>
- Fox, J., Weisberg, S., & Price, B. (2018). *Car: Companion to applied regression*. Retrieved from <https://CRAN.R-project.org/package=car>
- Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). *Applied linear statistical model (5th ed.)*. Singapore: McGraw-Hill Education (Asia).