PSY 504: Advanced Statistics

Poisson (🐟) Regression and Negative Binomial Regression

Jason Geller, Ph.D. (he/him/his)

Princeton University

Updated:2023-03-04


RICHARD TERMINE

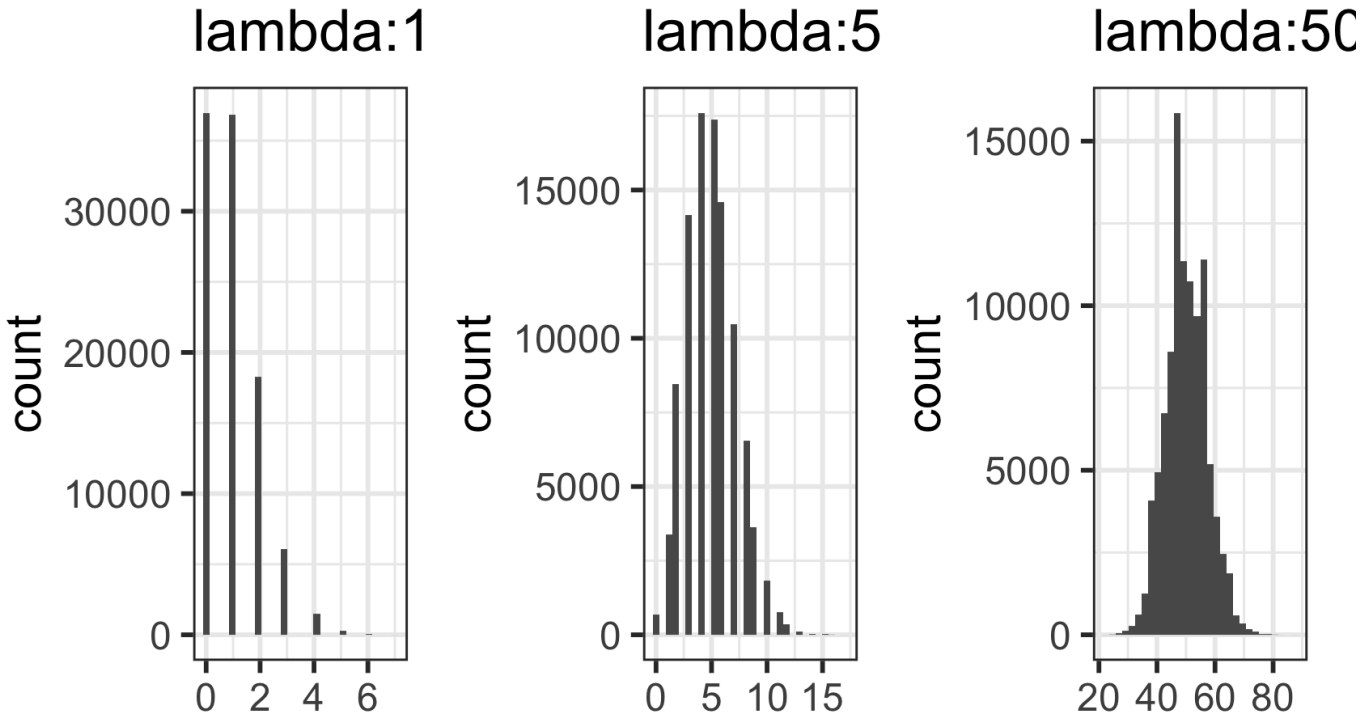
Poisson distribution

Let Y be the number of events in a given unit of time or space. Then Y can be modeled using a **Poisson distribution**

$$P(Y = y|\lambda) = \frac{e^{-\lambda}\lambda^y}{y!} \quad y = 0, 1, 2, \ldots, \infty$$

Features

- $E(Y) = Var(Y) = \lambda$ (*just the mean number of events*)
- The distribution is typically skewed right, particularly if λ is small
- The distribution becomes more symmetric as λ increases
  - If λ is sufficiently large, it can be approximated using a normal distribution



	Mean	Variance
lambda = 1	0.99351	0.9902178
lambda = 5	4.99367	4.9865798
lambda = 50	49.99288	49.8962682

Examples

The annual number of earthquakes registering at least 2.5 on the Richter Scale and having an epicenter within 40 miles of downtown Memphis follows a Poisson distribution with mean 6.5. **What is the probability there will be 3 or fewer such earthquakes next year?**

$$P(Y = y|\lambda) = \frac{e^{-6.5}10^0}{0!} + \frac{e^{-6.5}10^1}{1!} + \frac{e^{-6.5}10^2}{2!} + \frac{e^{-6.5}10^3}{3!}$$

```
a=(exp(-6.5) * 6.5^0) / factorial(0)
b=(exp(-6.5) * 6.5^1) / factorial(1)
c=(exp(-6.5) * 6.5^2) / factorial(2)
d=(exp(-6.5) * 6.5^3) / factorial(3)

ppois(3, 6.5)
```

```
## [1] 0.1118496
```

Examples

- Exact count
  - Let's say you read, on average, 10 pages an hour. **What is the probability you will read 8 pages in an hour?**

$$P(Y = y|\lambda) = \frac{e^{-10}10^8}{8 * 7 * 6 * 5 * 4 * 3 * 2 * 1}$$

```
prob <- (exp(-10) * 10^8) / factorial(8)

dpois(x=8, lambda=10)
```

```
## [1] 0.112599
```

```
prob
```

```
## [1] 0.112599
```

Poisson regression

Preferential viewing task

- The data: Viewing behavior to emotional faces

Preferential viewing task

Response:

- Number of fixations (visits) to each face
- Predictors:
  - **`Emotion`**: Anger vs. Happy
  - **`Group`**: Control vs. Stuttering

The data

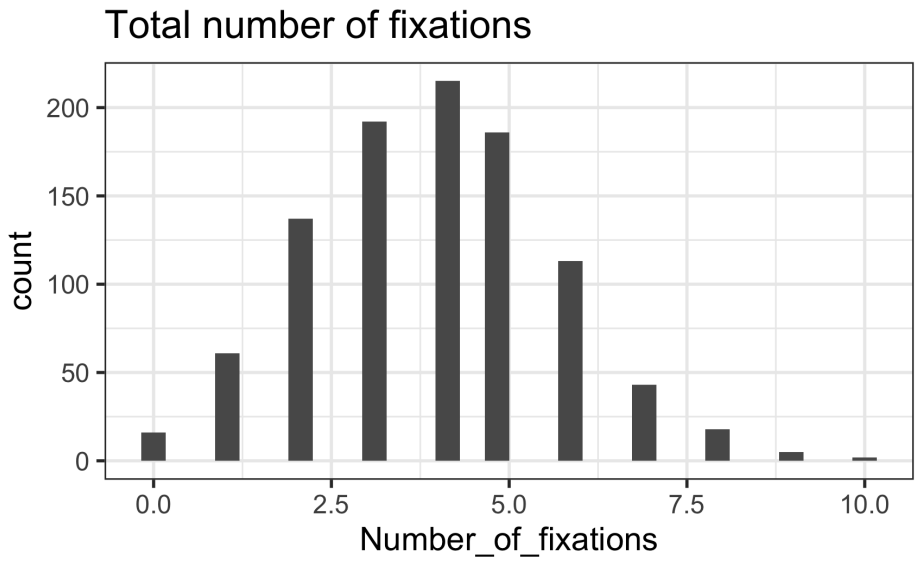
```
#####PUT ON GIT
hh_data <- read_csv(here::here("data", "tobii_aoi_study1.csv"))
```

Package

```
library(tidyverse)
library(performance) # check model
library(lme4) # glmer and glmer.nb
library(emmeans) # marginal means and contrasts
library(ggeffects) # viz
library(broom.mixed) #lme4 tidy
#library(MASS) # glm.nb
```

```
## # A tibble: 6 × 4
##   ID    Number_of_fixations emotion Group
##   <chr>               <dbl> <chr>   <chr>
## 1 AB001                   4 Anger   C
## 2 AB001                   1 Anger   C
## 3 AB001                   3 Happy   C
```

Response variable



mean	var	ratio
3.936	3.091	1.273

Why the least-squares model doesn't work

The goal is to model λ, the expected number of fixations on faces, as a function of the predictors (covariates)

We might be tempted to try a linear model

$$\lambda_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

This model won't work because...

- It could produce negative values of λ for certain values of the predictors
- The equal variance assumption required to conduct inference for linear regression is violated.

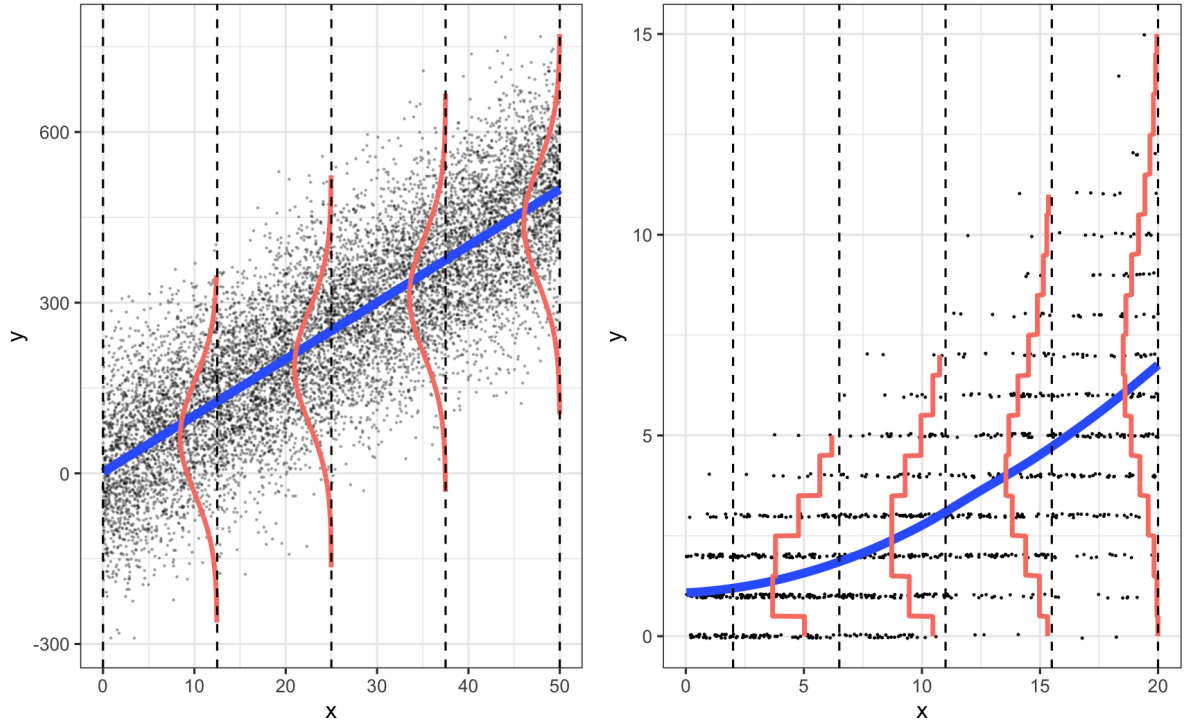
Poisson regression model

If $Y_i \sim Poisson$ with $\lambda = \lambda_i$ for the given values $x_{i1}, \ldots, x_{ip}$, then

$$\log(\lambda_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

- Each observation can have a different value of λ based on its value of the predictors $x_1, \ldots, x_p$
- λ determines the mean and variance, so we don't need to estimate a separate error term

Poisson vs. multiple linear regression



Regression models: Linear regression (left) and Poisson regression (right).

From BMLR Figure 4.1

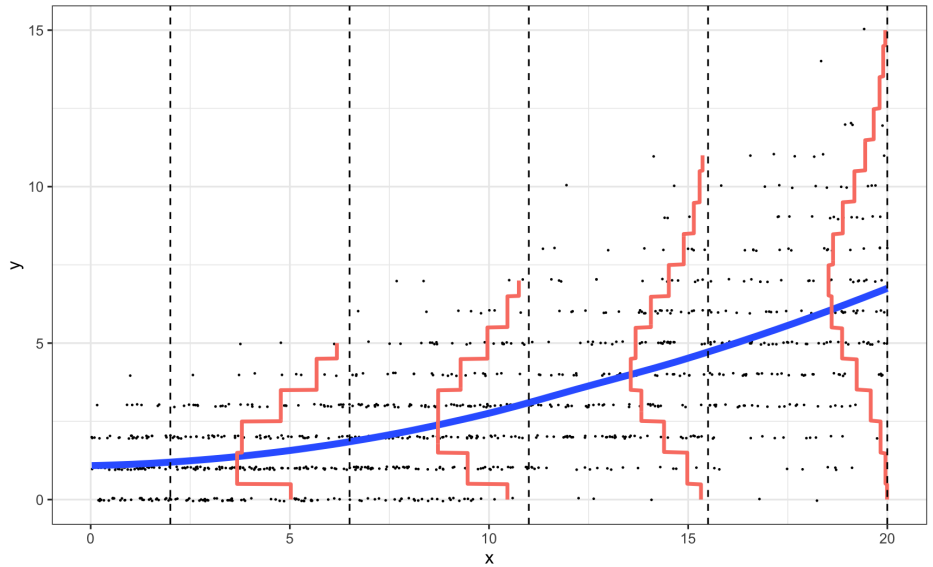
Assumptions for Poisson regression

Poisson response: The response variable is a count per unit of time or space, described by a Poisson distribution, at each level of the predictor(s)

Independence: The observations must be independent of one another

Linearity: The log of the mean rate, $\log(\lambda)$, must be a linear function of the predictor(s)

Mean = Variance: The mean must equal the variance



Poisson regression: Fitting and Interpretation

Poisson regression: Fitting and Interpretation

- **glm**

```
model_glm <- glm(Number_of_fixations ~ emotion+ Group, data = hh_data, family = po
```

- **glmer**

```
library(lme4)
# fit poisson model# change family to poisson
# tidy summary
# repeated measures poisson
#dummy coded
model1 <- glmer(Number_of_fixations ~ emotion+ Group + (1|ID), data = hh_data, fam
# tidy summary
#contrast coded (0.5, -0.5)
model1_cont <- glmer(Number_of_fixations ~ emotion+ Group + (1|ID), data = hh_data
# tidy summary
```

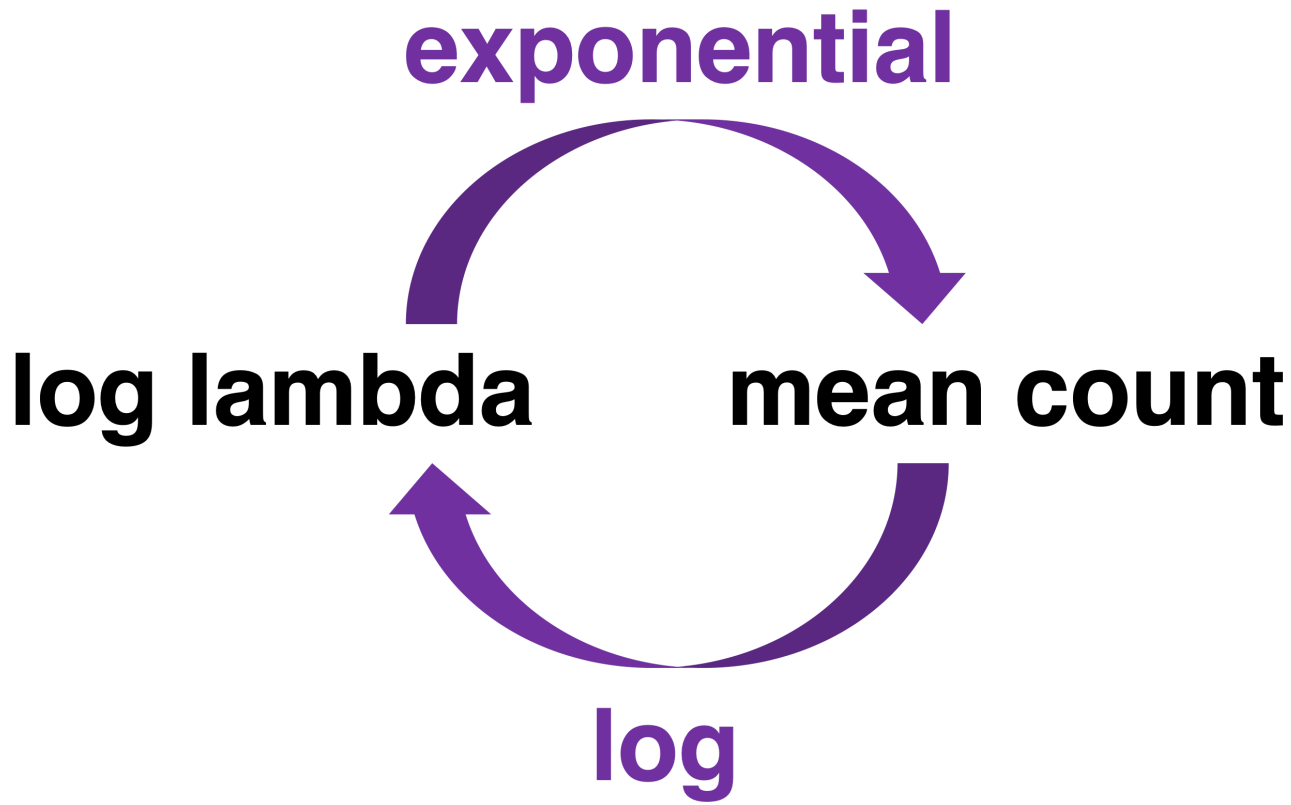
Poisson regression: Fitting and interpretation

```
tidy(model1_cont, exponentiate = FALSE, conf.int =TRUE) %>% kable(digits = 3, forma
```

effect	group	term	estimate	std.error	statistic	p.value	conf.low	conf.high
fixed	NA	(Intercept)	1.352	0.035	38.761	0.000	1.283	1.420
fixed	NA	emotion	-0.085	0.032	-2.648	0.008	-0.148	-0.022
fixed	NA	Group	0.085	0.070	1.228	0.220	-0.051	0.222
ran_pars	ID	sd__(Intercept)	0.170	NA	NA	NA	NA	NA

Poisson regression: Fitting and interpretation

- Mean count rather than log count more interpretable



Poisson regression: Fitting and interpretation

- Incidence rate ratios (IRR)
  - The IRR for a one-unit change in x_i is exp(b_i)
  - The coefficient tells you how changes in X affect the rate at which Y occurs

effect	group	term	estimate	std.error	statistic	p.value	conf.low	conf.high
fixed	NA	(Intercept)	3.864	0.135	38.761	0.000	3.608	4.137
fixed	NA	emotion	0.919	0.029	-2.648	0.008	0.863	0.978
fixed	NA	Group	1.089	0.076	1.228	0.220	0.950	1.249
ran_pars	ID	sd__(Intercept)	0.170	NA	NA	NA	NA	NA

Poisson regression: Fitting and interpretation

effect	group	term	estimate	std.error	statistic	p.value	conf.low	conf.high
fixed	NA	(Intercept)	1.352	0.035	38.761	0.000	1.283	1.420
fixed	NA	emotion	-0.085	0.032	-2.648	0.008	-0.148	-0.022
fixed	NA	Group	0.085	0.070	1.228	0.220	-0.051	0.222
ran_pars	ID	sd__(Intercept)	0.170	NA	NA	NA	NA	NA

- exp($\alpha) = Overall mean count

Poisson regression: Fitting and interpretation

Log count

term	estimate
(Intercept)	1.352
emotion	-0.085
Group	0.085
sd__(Intercept)	0.170

Exp

term	estimate
(Intercept)	3.864
emotion	0.919
Group	1.089
sd__(Intercept)	0.170

- -.08 = Angry faces have -.14 fewer log fixations than neutral faces, or
    - exp(-.08) = 0.923x that of happy faces

Main effect: `emotion`

- Marginal mean counts

```
# get rate for emotion
emmeans(model1, "emotion", type="response") %>%
  kable(digits = 3, format = "markdown")
```

emotion	rate	SE	df	asymp.LCL	asymp.UCL
Anger	3.703	0.143	Inf	3.433	3.995
Happy	4.031	0.154	Inf	3.741	4.344

Poisson regression: Fitting and interpretation

Log count

term	estimate
(Intercept)	1.352
emotion	-0.085
Group	0.085
sd__(Intercept)	0.170

Exp

term	estimate
(Intercept)	3.864
emotion	0.919
Group	1.089
sd__(Intercept)	0.170

- .085 = Stuttering group has .085 more log fixations than Control group, or
    - exp(0.85) = 1.089x that of Control group

Main effect: `Group`

- Marginal mean counts

```
emmeans(model1, "Group",  type="response") %>%
  kable(digits = 3, format = "markdown")
```

Group	rate	SE	df	asymp.LCL	asymp.UCL
C	3.702	0.192	Inf	3.344	4.098
S	4.032	0.188	Inf	3.680	4.418

Full model

- LRT test for more complex models

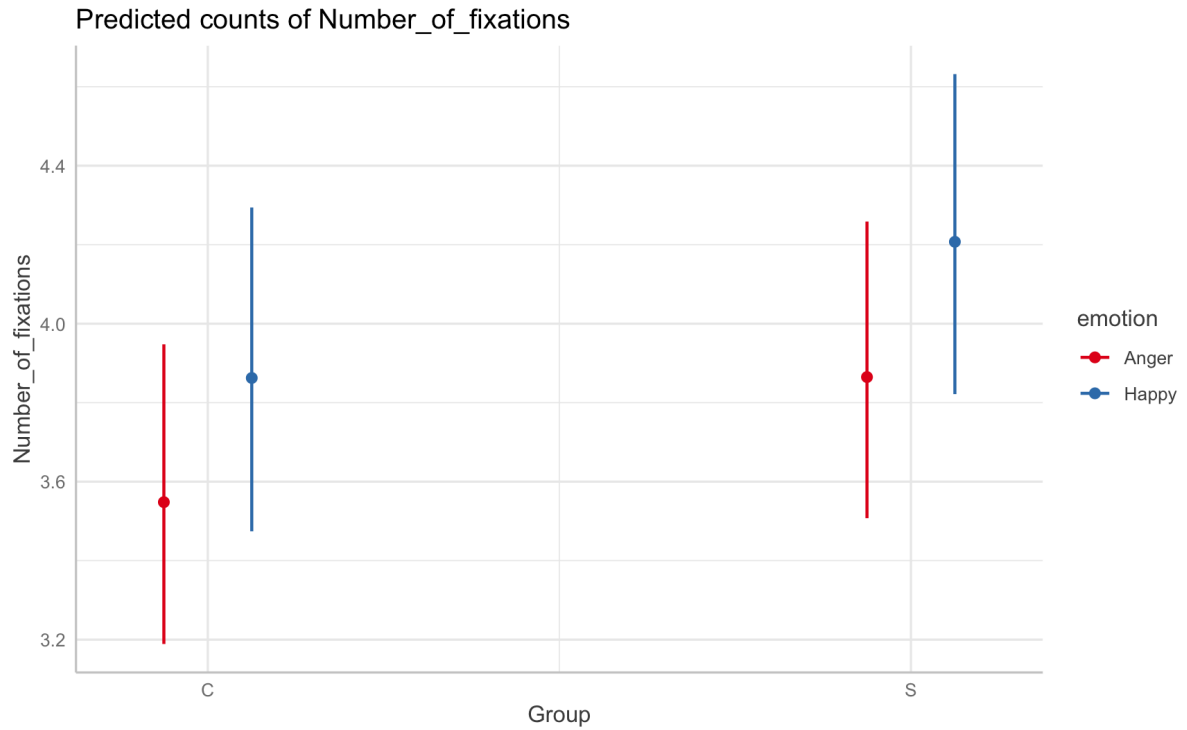
```
#overall model
mod_1 <- car::Anova(model1)

mod_1 %>% kable(digits = 3, format = "markdown")
```

	Chisq	Df	Pr(>Chisq)
emotion	7.013	1	0.008
Group	1.507	1	0.220

Visualizing poisson regression

- Used expected/predicted values

```
ggemmeans(model1, terms=c("Group", "emotion")) %>%
  plot()
```



Model 2: Add interaction

```
model2 <- glmer(Number_of_fixations ~ emotion*Group+ (1|ID), data = hh_data_contra

tidy(model2) %>%
    kable(digits = 3, format = "markdown")
```

effect	group	term	estimate	std.error	statistic	p.value
fixed	NA	(Intercept)	1.351	0.035	38.725	0.000
fixed	NA	emotion	-0.095	0.032	-2.927	0.003
fixed	NA	Group	0.089	0.070	1.277	0.202
fixed	NA	emotion:Group	0.148	0.065	2.281	0.023
ran_pars	ID	sd__(Intercept)	0.170	NA	NA	NA

Add `emotion*Group` to the model?

- ✅ Conduct a drop-in-deviance LR test

```
anova(model1_cont, model2, test="chisq")
```

```
## Data: hh_data_contrast
## Models:
## model1_cont: Number_of_fixations ~ emotion + Group + (1 | ID)
## model2: Number_of_fixations ~ emotion * Group + (1 | ID)
##             npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## model1_cont    4 3866.7 3886.3 -1929.4   3858.7
## model2         5 3863.5 3888.0 -1926.8   3853.5 5.1974  1    0.02262 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interaction: `Group*emotion`

- Simple effects test done on the *response* (count)

```
emmeans(model1, specs=c("Group", "emotion"), regrid = "response") %>% pairs(., by=
      kable(digits = 3, format = "markdown")
```

contrast	Group	estimate	SE	df	z.ratio	p.value
Anger - Happy	C	-0.314	0.12	Inf	-2.627	0.009
Anger - Happy	S	-0.342	0.13	Inf	-2.631	0.009

Goodness-of-fit

Pearson residuals

We can calculate two types of residuals for Poisson regression: Pearson residuals and deviance residuals

$$\text{Pearson residual}_i = \frac{\text{observed} - \text{predicted}}{\text{std. error}} = \frac{y_i - \hat{\lambda}_i}{\sqrt{\hat{\lambda}_i}}$$

- Similar interpretation as standardized residuals from linear regression
- Expect most to fall between -2 and 2
- Used to calculate overdispersion parameter

Deviance residuals

The **deviance residual** indicates how much the observed data deviates from the fitted model

$$\text{deviance residual}_i = \text{sign}(y_i - \hat{\lambda}_i)\sqrt{2\left[y_i \log\left(\frac{y_i}{\hat{\lambda}_i}\right) - (y_i - \hat{\lambda}_i)\right]}$$

where

$$\text{sign}(y_i - \hat{\lambda}_i) = \begin{cases} 1 & \text{if } (y_i - \hat{\lambda}_i) > 0 \\ -1 & \text{if } (y_i - \hat{\lambda}_i) < 0 \\ 0 & \text{if } (y_i - \hat{\lambda}_i) = 0 \end{cases}$$

Goodness-of-fit

- **Goal**: Use the (residual) deviance to assess how much the predicted values differ from the observed values. Recall

$$(\text{deviance}) = \sum_{i=1}^{n} (\text{deviance residual})_i^2$$

- If the model sufficiently fits the data, then :

$$\text{deviance} \sim \chi^2_{df}$$

where df is the model's residual degrees of freedom

Model 1: Goodness-of-fit calculations

```
# tidy function glance
dev_mod1 <- glance(model1)
dev_mod1$deviance
```

```
## [1] 706.6215
```

```
dev_mod1$df.residual
```

```
## [1] 984
```

```
pchisq(dev_mod1$deviance,df= dev_mod1$df.residual, lower.tail = FALSE)
```

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

The probability of observing a deviance greater than 706.6 is ≈ 1, so there is no evidence of lack-of-fit.

Lack-of-fit

There are a few potential reasons for lack-of-fit:

- Missing important interactions or higher-order terms
- Missing important variables (perhaps this means a more comprehensive data set is required)
- There could be extreme observations causing the deviance to be larger than expected (assess based on the residual plots)
- There could be a problem with the Poisson model
    - May need more flexibility in the model to handle **overdispersion**

Overdispersion

Overdispersion: There is more variability in the response than what is implied by the Poisson model

Overall

mean	var
3.936	3.091

by Emotion

emotion	mean	var	ratio
Anger	3.769	3.200	1.178
Happy	4.103	2.933	1.399

by Group

Group	mean	var	ratio
C	3.768	3.337	1.129
S	4.076	2.850	1.430

Testing for overdispersion

- **Easystats**

```
check_overdispersion(model1)
```

```
## # Overdispersion test
##
##        dispersion ratio =   0.651
##   Pearson's Chi-Squared = 640.180
##                 p-value =       1
```

Why overdispersion matters

- If there is overdispersion, then there is more variation in the response than what's implied by a Poisson model. This means:

❌ The standard errors of the model coefficients are artificially small

❌ The p-values are artificially small

❌ This could lead to models that are more complex than what is needed

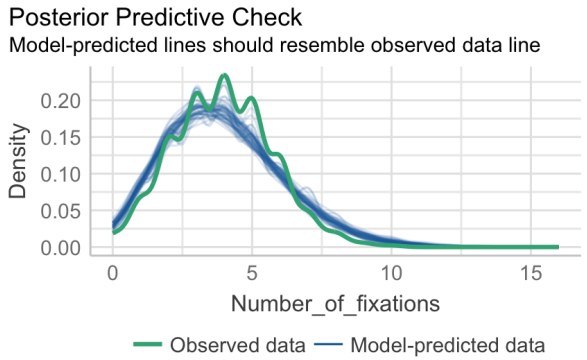
Why overdispersion matters

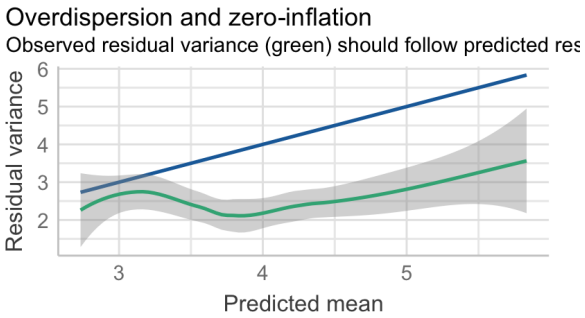
We can take overdispersion into account by:

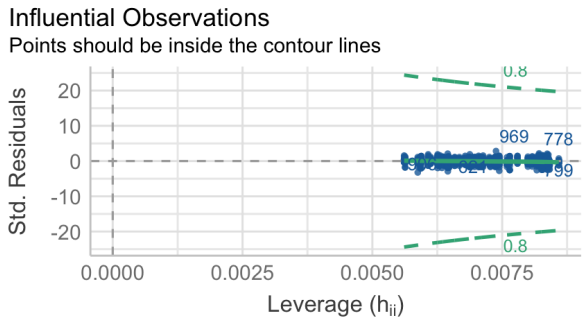
- Inflating standard errors by multiplying them by a dispersion factor
- Using a negative-binomial regression model

Assumptions

```
##
performance::check_model(model2, check = c("pp_check", "outliers", "overdispersion
```







Negative binomial regression model

Negative binomial regression model

Another approach to handle overdispersion is to use a **negative binomial regression model**

- Basically a poisson model, but allowing for a dispersion parameter *r*

$$Var(Y) = \mu + \frac{\mu^2}{r}$$

- Makes the counts more dispersed than with a single parameter

Running negative binomial

- **`glmer.nb`** (nested data)

```
#use to run neg binomial

m.nb <- glmer.nb(Number_of_fixations ~ emotion*Group + (1|ID), data=hh_data_contra

tidy(m.nb, conf.int =TRUE) %>%
  kable(digits = 3, format = "markdown")
```

effect	group	term	estimate	std.error	statistic	p.value	conf.low	conf.high
fixed	NA	(Intercept)	1.351	0.035	39.044	0.000	1.283	1.419
fixed	NA	emotion	-0.095	0.032	-2.938	0.003	-0.158	-0.032
fixed	NA	Group	0.089	0.067	1.319	0.187	-0.043	0.221
fixed	NA	emotion:Group	0.148	0.063	2.340	0.019	0.024	0.271

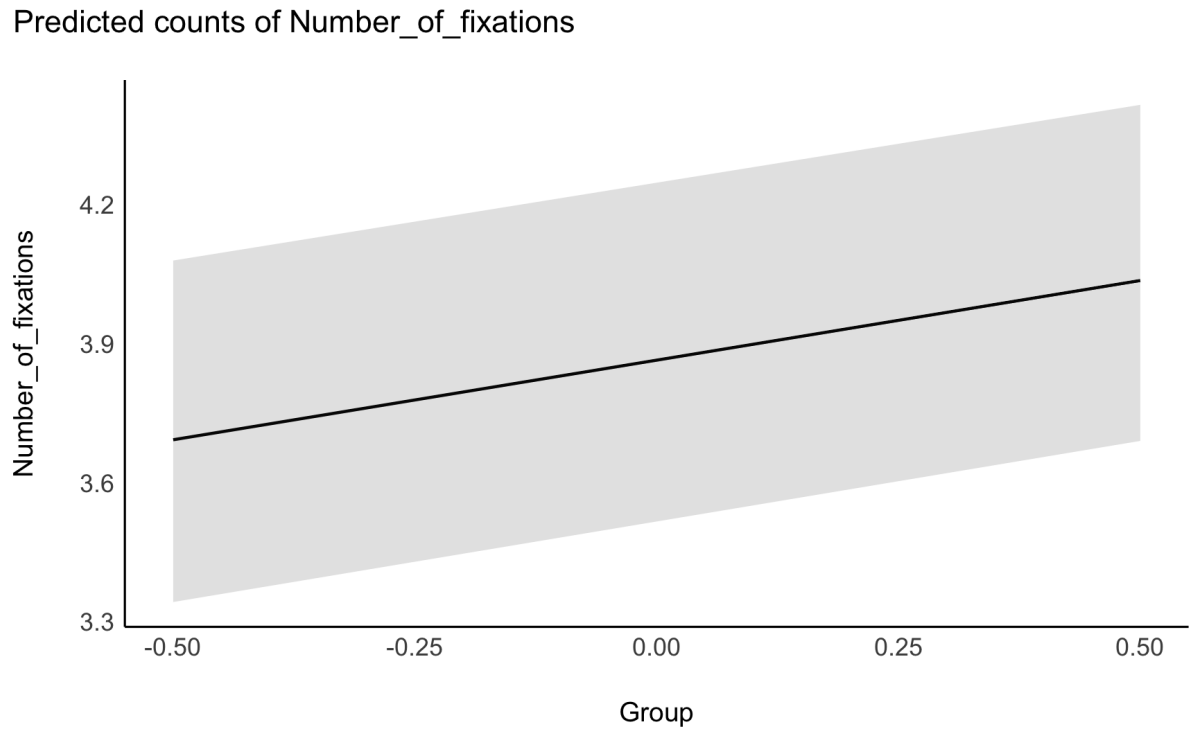
Running negative binomial

- **glm.nb** (non-nested/between-subjects)

```
library(MASS)
#use to run neg binomial
m.nb <- glm.nb(Number_of_fixations ~ emotion*Group, data=hh_data_contrast)

tidy(m.nb) %>%
  kable(digits = 3, format = "markdown")
```

Visualize negative binomial

- Same as Poisson (show expected counts)



Effect sizes

- RecountD
  - Cohen's *d*

Reporting a poisson regression

- State your hypothesis, statistical test, its link function, and justify the use of a poisson regression
    - We hypothesized that years that had more water main breaks would have fewer renewable projects approved. The number of renewable projects approved by the City of Toronto represented a count of rare events, which violated the normality assumption required for traditional regression. Thus, a poisson regression with a log link function was used to predict the number of renewable projects in Toronto in a given year using R 4.0.4 (R Core Team, 2020). Prior to the analysis, the number of water main breaks was mean centered. Furthermore, the number of water main breaks was divided by 100 to improve interpretation of the slopes. Effect sizes that approximate Cohen's d were calculated using the RCountD Shiny App (Coxe, 2018).

Reporting a Poisson Regression

- State the full model and your results
    - The number of renewable projects in a given year were modelled as a function of water main breaks in the same year. As shown in Figure 1, this analysis revealed that years with more water main breaks had fewer renewable projects approved, b = -0.15, SE = 0.06, z(11) = -2.56, p = 0.01, d = -0.27

Reporting a negative binomial regression

- State your hypothesis, statistical test, its link function, and justify the use of a negative binomial regression
    - We hypothesized that people with more opportunity for conflict (i.e., who had more social interactions) would report more interpersonal conflicts over 10 days. The number of conflicts during the 10 days represented frequency counts, which violated the normality assumption required for traditional regression, but these counts were also zero-inflated as most people did not have any interpersonal conflicts during this time. Thus, a negative binomial regression was used to predict the number of interpersonal conflicts using the MASS package (Venables & Ripley, 2002) in R 4.0.4 (R Core Team, 2020). Effect sizes that approximate Cohen's d were calculated using the RCountD Shiny App (Coxe, 2018).

Reporting a negative binomial regression

- State the full model and your results
    - The number of interpersonal conflicts were modelled as a function of social interactions and typical mood, and the covariates of age, sex, student status, public transit use, alcohol use, and average daily mood. As shown in Figure 1, this analysis revealed that people who had more social interactions were slightly more likely to have interpersonal conflicts, b = 0.06, SE = 0.01, z(52) = 5.16, p < 0.01, d = .05. The estimates for the full model are provided in Table 1.

Underdisperion

- Variance < mean
    - performance::check_zeroinflation()
- Use zero-inflated poisson (**`pscl`** package)
- Use **`brms`**

Wenesday

- Watch videos
- Kabacoff, R. I. (2022). *R* in Action* Chapter 13