

GLM for count data: Poisson regression

Francisco Rodríguez-Sánchez

<https://frodriguezsanchez.net>

- Gaussian: `lm`

- Gaussian: `lm`
- Binary: `glm (family binomial / quasibinomial)`

- Gaussian: `lm`
- Binary: `glm (family binomial / quasibinomial)`
- Counts: `glm (family poisson / quasipoisson)`

Poisson regression

- Response variable: Counts (0, 1, 2, 3...) - discrete
- Link function: \log

Then

$$\log(N) = a + bx$$

$$N = e^{a+bx}$$

Example dataset: Seedling counts in quadrats

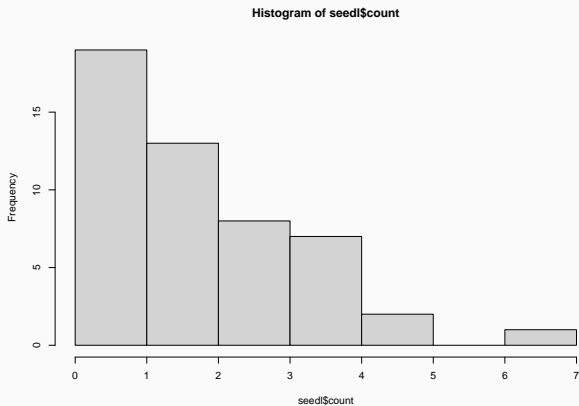
```
seedl <- read.csv("data/seedlings.csv")
```

sample	count	light	area
Min. : 1.00	Min. :0.00	Min. : 2.571	Min. :0.25
1st Qu.:13.25	1st Qu.:1.00	1st Qu.:26.879	1st Qu.:0.25
Median :25.50	Median :2.00	Median :47.493	Median :0.50
Mean :25.50	Mean :2.14	Mean :47.959	Mean :0.62
3rd Qu.:37.75	3rd Qu.:3.00	3rd Qu.:67.522	3rd Qu.:1.00
Max. :50.00	Max. :7.00	Max. :99.135	Max. :1.00

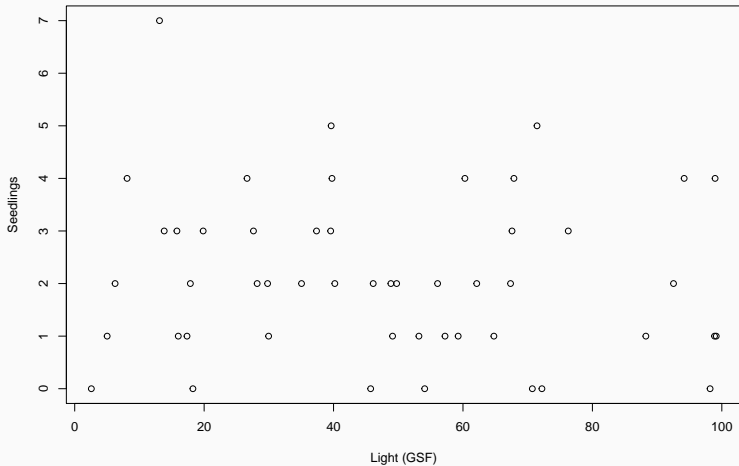
Exploring the data

```
table(seedl$count)
```

0	1	2	3	4	5	7
7	12	13	8	7	2	1



Relationship between Nseedlings and light?



Poisson regression

```
seedl.glm <- glm(count ~ light,  
                 data = seedl,  
                 family = poisson)
```

which corresponds to

```
equatiomatic::extract_eq(seedl.glm)
```

$$\log(E(\text{count})) = \alpha + \beta_1(\text{light})$$

Interpreting Poisson GLM

Call:

```
glm(formula = count ~ light, family = poisson, data = seed1)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.1906	-0.8466	-0.1110	0.5220	2.4577

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.881805	0.188892	4.668	3.04e-06 ***
light	-0.002576	0.003528	-0.730	0.465

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 63.029 on 49 degrees of freedom
Residual deviance: 62.492 on 48 degrees of freedom
AIC: 182.03

Number of Fisher Scoring iterations: 5

Parameter estimates are in log scale!

Parameter estimates (log scale):

```
coef(seedl.glm)[1]
```

(Intercept)

0.881805

We need to back-transform: apply the inverse of the logarithm

```
exp(coef(seedl.glm)[1])
```

(Intercept)

2.415255

```
allEffects(seedl.glm)
```

```
model: count ~ light
```

```
light effect
```

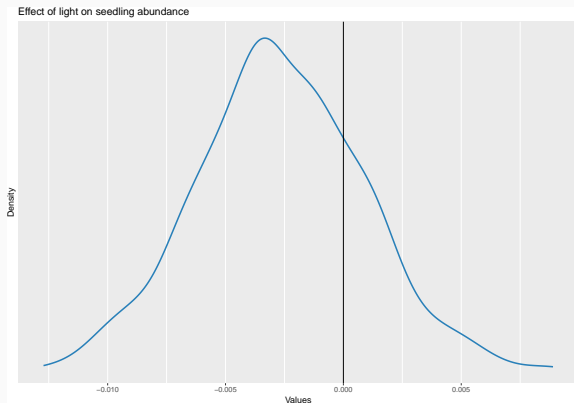
```
light
```

3	30	50	70	100
---	----	----	----	-----

2.396665	2.235657	2.123408	2.016794	1.866826
----------	----------	----------	----------	----------

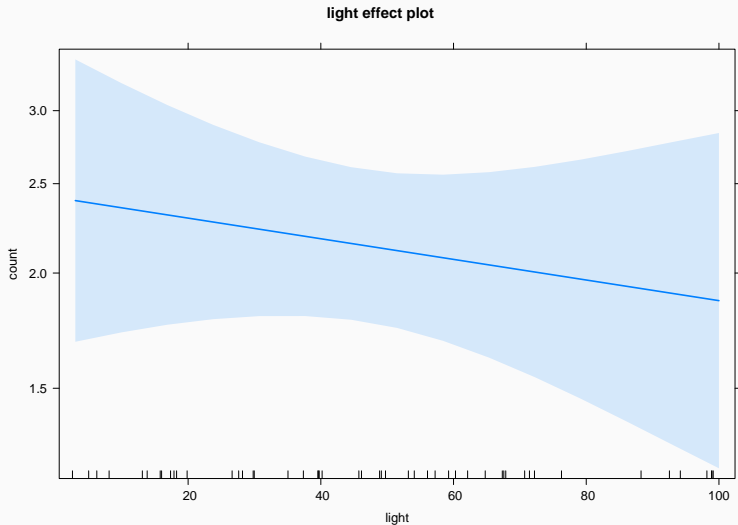
Estimated distribution of the slope parameter

```
library("parameters")  
plot(simulate_parameters(seedl.glm)) +  
  geom_vline(xintercept = 0) +  
  ggtitle("Effect of light on seedling abundance")
```



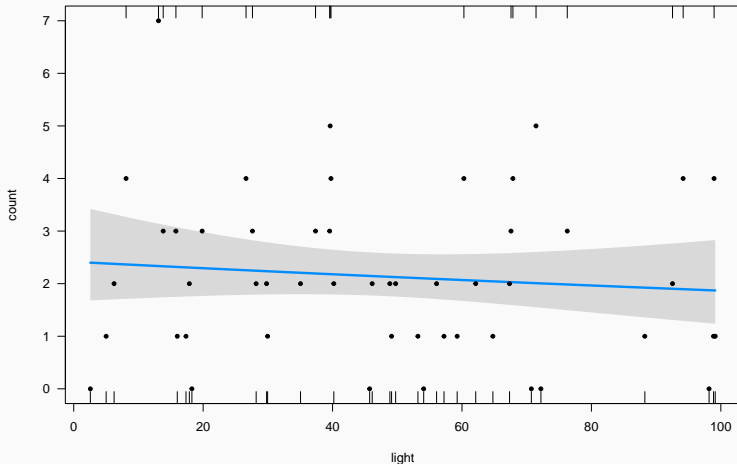
So what's the relationship between Nseedlings and light?

```
plot(allEffects(seedl.glm))
```



Using visreg

```
visreg(seedl.glm, scale = "response", ylim = c(0, 7))  
points(count ~ light, data = seedl, pch = 20)
```



```
library("performance")  
r2(seedl.glm)
```

```
# R2 for Generalized Linear Regression  
Nagelkerke's R2: 0.015
```


Describing the model results

```
library("report")  
report(seedl.glm)
```

We fitted a poisson model (estimated using ML) to predict count with light (formula: count ~ light). The model's explanatory power is very weak (Nagelkerke's $R^2 = 0.01$). The model's intercept, corresponding to light = 0, is at 0.88 (95% CI [0.50, 1.24], $p < .001$). Within this model:

- The effect of light is statistically non-significant and negative (beta = $-2.58e-03$, 95% CI [$-9.57e-03$, $4.28e-03$], $p = 0.465$; Std. beta = -0.07 , 95% CI [-0.27 , 0.12])

Standardized parameters were obtained by fitting the model on a standardized version of the dataset. 95% Confidence Intervals (CIs) and p-values were computed using

Model checking

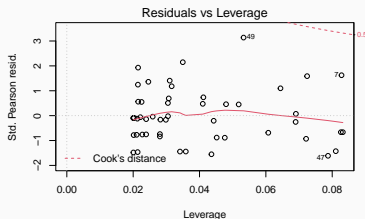
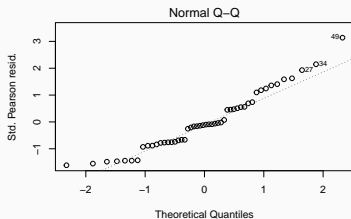
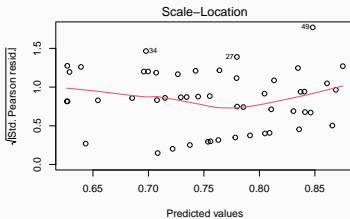
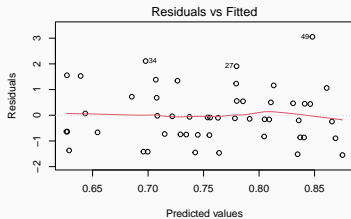
- Linearity (log response \sim predictors)

- Linearity (log response \sim predictors)
- Observations are independent

- Linearity (log response \sim predictors)
- Observations are independent
- Mean = Variance

Checking Poisson GLM

```
plot(seedl.glm)
```



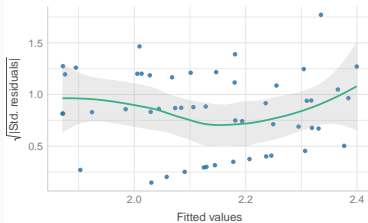
null device

Checking Poisson GLM

```
check_model(seedl.glm)
```

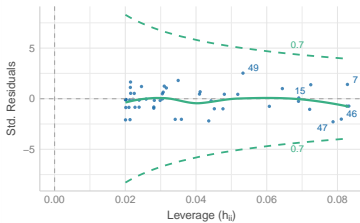
Homogeneity of Variance

Reference line should be flat and horizontal



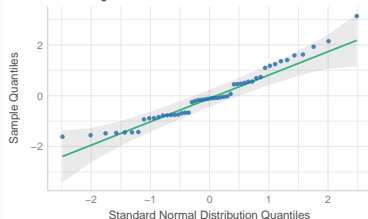
Influential Observations

Points should be inside the contour lines



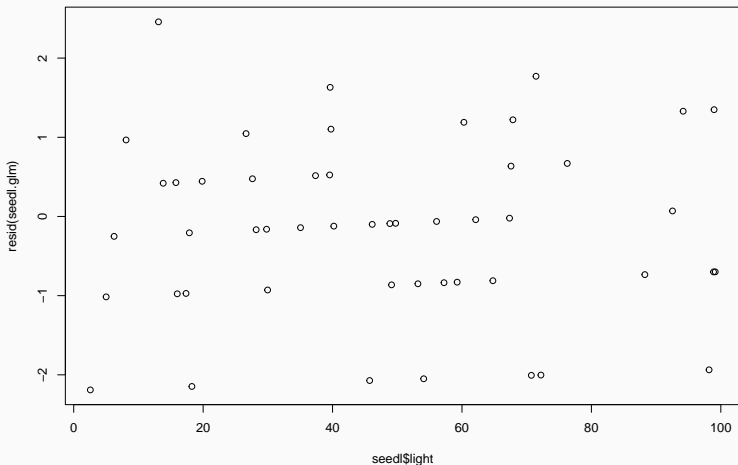
Normality of Residuals

Dots should fall along the line



Is there pattern of residuals along predictor?

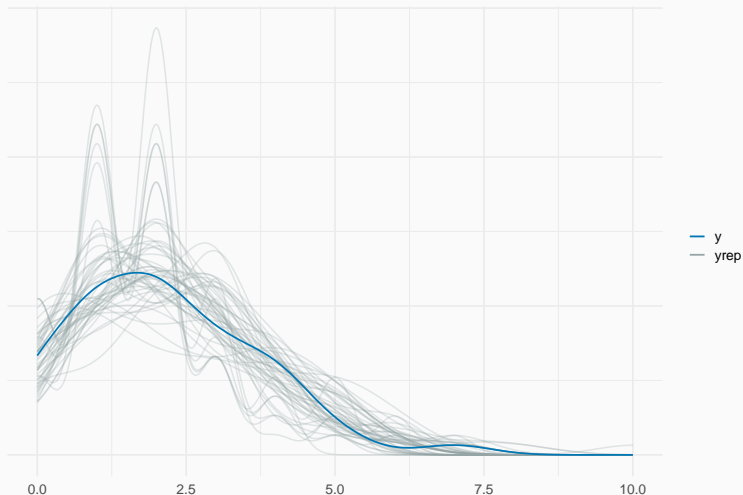
```
plot(seedl$light, resid(seedl.glm))
```



Posterior predictive checking

Simulate data from fitted model (y_{rep}) and compare with observed data (y)

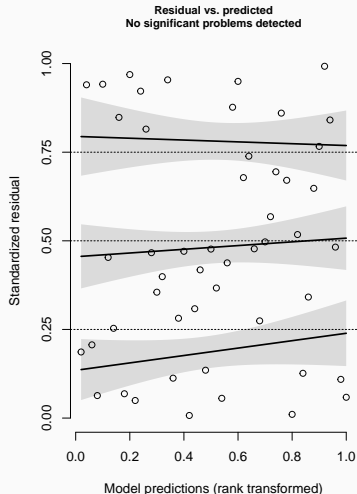
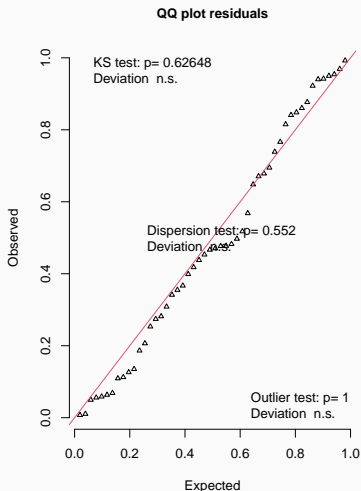
Posterior Predictive Check



Residuals diagnostics with DHARMA

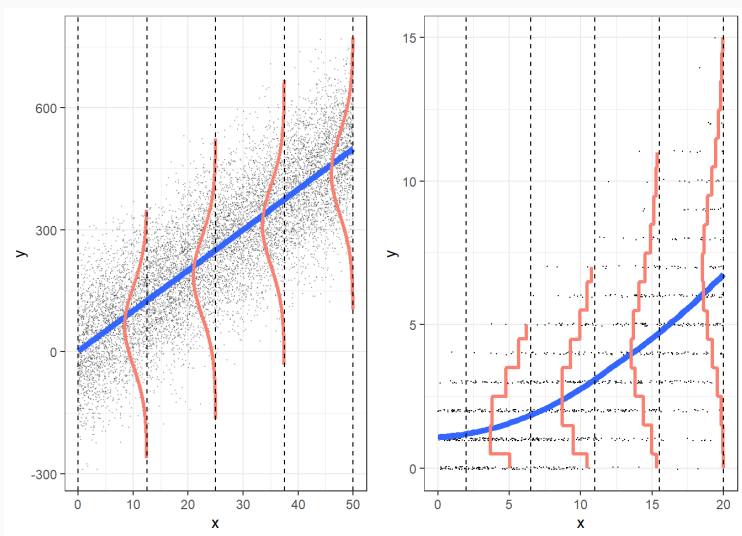
```
simulateResiduals(seedl.glm, plot = TRUE)
```

DHARMA residual diagnostics



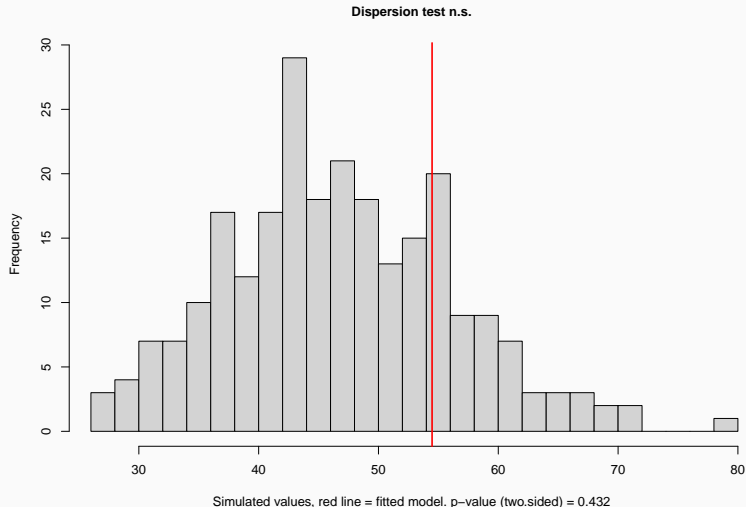
Overdispersion

Poisson GLM assumes mean = variance



Always check overdispersion with count data

```
simres <- simulateResiduals(seedl.glm, refit = TRUE)  
testDispersion(simres)
```



- Use family `quasipoisson`

- Use family `quasipoisson`
- Use negative binomial distribution (`MASS::glm.nb`)

- Use family `quasipoisson`
- Use negative binomial distribution (`MASS::glm.nb`)
- Include observation-level random effect (e.g. see [Harrison 2014](#))

Accounting for overdispersion with family quasipoisson

Call:

```
glm(formula = count ~ light, family = quasipoisson, data = seedl)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.1906	-0.8466	-0.1110	0.5220	2.4577

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.881805	0.201230	4.382	6.37e-05 ***
light	-0.002576	0.003758	-0.685	0.496

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 1.134907)

Null deviance: 63.029 on 49 degrees of freedom
Residual deviance: 62.492 on 48 degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5

Mean estimates do not change after accounting for overdispersion

```
allEffects(seedl.overdisp)
```

```
model: count ~ light
```

```
light effect
```

```
light
```

	3	30	50	70	100
	2.396665	2.235657	2.123408	2.016794	1.866826

```
allEffects(seedl.glm)
```

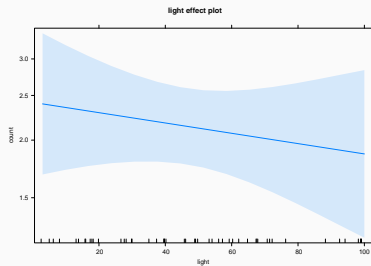
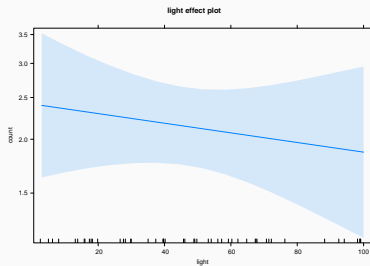
```
model: count ~ light
```

```
light effect
```

```
light
```

	3	30	50	70	100
	2.396665	2.235657	2.123408	2.016794	1.866826

But standard errors may change



Accounting for overdispersion using negative binomial

```
library("MASS")  
seedl.nb <- glm.nb(count ~ light, data = seedl)
```

Call:

```
glm.nb(formula = count ~ light, data = seedl, init.theta = 22.23419419,  
       link = log)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.1349	-0.8162	-0.1061	0.4954	2.2814

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.881996	0.198213	4.450	8.6e-06 ***
light	-0.002580	0.003691	-0.699	0.485

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(22.2342) family taken to be 1)

Null deviance: 58.247 on 49 degrees of freedom
Residual deviance: 57.756 on 48 degrees of freedom

What if survey plots have
different area?

Shall we *standardise* counts dividing by sampling plot area?

Model would be: $\text{count/area} \sim \text{light}$

	sample	count	light	area
1	1	0	70.71854	0.50
2	2	1	88.26021	0.25
3	3	2	67.35133	0.50
4	4	3	67.57850	1.00
5	5	4	26.63098	0.25
6	6	3	15.79433	1.00

J. R. Statist. Soc. A (1993)
156, Part 3, pp. 379–392

Spurious Correlation and the Fallacy of the Ratio Standard Revisited

By RICHARD A. KRONMAL†

<https://doi.org/10.2307/2983064>

Use offset to account for variable sampling effort

```
seedl.offset <- glm(count ~ light,  
                    offset = log(area),  
                    data = seedl,  
                    family = poisson)
```


Note estimates now referred to area units!

Call:

```
glm(formula = count ~ light, family = poisson, data = seedl,  
     offset = log(area))
```

Deviance Residuals:

	Min	1Q	Median	3Q	Max
	-2.9918	-1.0142	0.1673	0.8401	3.8230

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.513185	0.183245	8.258	<2e-16 ***
light	-0.005674	0.003384	-1.677	0.0936 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Note estimates now referred to area units!

```
exp(coef(seedl.offset)[1])
```

(Intercept)

4.541173

Prediction

Predicting number of seedlings given light

```
new.lights <- data.frame(light = c(10, 90))  
predict(seedl.glm, newdata = new.lights, type = "response", se.fit
```

```
$fit
```

	1	2
	2.353841	1.915533

```
$se.fit
```

	1	2
	0.3756992	0.3502446

```
$residual.scale
```

```
[1] 1
```

- Infant mortality \sim GDP

- Infant mortality ~ GDP
- Number of cones consumed by squirrels ([data](#))

- Infant mortality ~ GDP
- Number of cones consumed by squirrels ([data](#))
- Elephant matings ([Poole 1989](#))