

# **Modelos estadísticos II: Modelos lineales generalizados**

Técnicas estadísticas avanzadas para la conservación de la biodiversidad - Universidad de Huelva

---

David García Callejas

01/2021

# Modelos lineales generalizados

- Hasta ahora: modelos lineales con variable respuesta **continua** y residuos **normales**

```
gupp <- read.csv(here::here("datasets",  
                           "chap17f9_1GuppyColdDeath.csv"))  
head(gupp)
```

##	fish	exposureDurationMin	mortality
## 1	1	3	1
## 2	2	3	1
## 3	3	3	1
## 4	4	3	1
## 5	5	3	1
## 6	6	3	1

# Modelos lineales generalizados

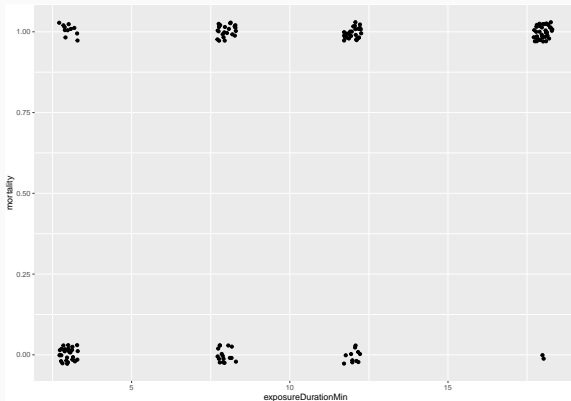
- Hasta ahora: modelos lineales con variable respuesta **continua** y residuos **normales**
- ¿podemos modelar variables con respuestas discretas? Por ejemplo, mortalidad de peces en función de tiempo de exposición a temperaturas de 5°C:

```
gupp <- read.csv(here::here("datasets",  
                           "chap17f9_1GuppyColdDeath.csv"))  
head(gupp)
```

##	fish	exposureDurationMin	mortality
## 1	1	3	1
## 2	2	3	1
## 3	3	3	1
## 4	4	3	1
## 5	5	3	1
## 6	6	3	1

# Modelos lineales generalizados

```
ggplot(gupp, aes(x = exposureDurationMin, y = mortality)) +  
  geom_point(position = position_jitter(width = .3, height = .03))
```



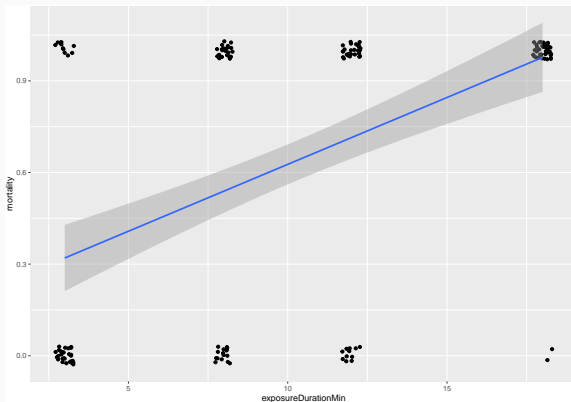
- ¿Podemos aplicar una regresión lineal a estos datos?
- ¿la relación entre  $X$  e  $Y$  es lineal?

# Modelos lineales generalizados

- ¿Podemos aplicar una regresión lineal a estos datos?
- ¿la relación entre  $X$  e  $Y$  es lineal?
- ¿esperamos que los residuos sean normales?

# Modelos lineales generalizados

```
lmgupp <- lm(mortality ~ exposureDurationMin, data = gupp)
ggplot(gupp, aes(x = exposureDurationMin, y = mortality)) +
  geom_point(position = position_jitter(width = .3, height = .03)) +
  geom_smooth(method = "lm")
```



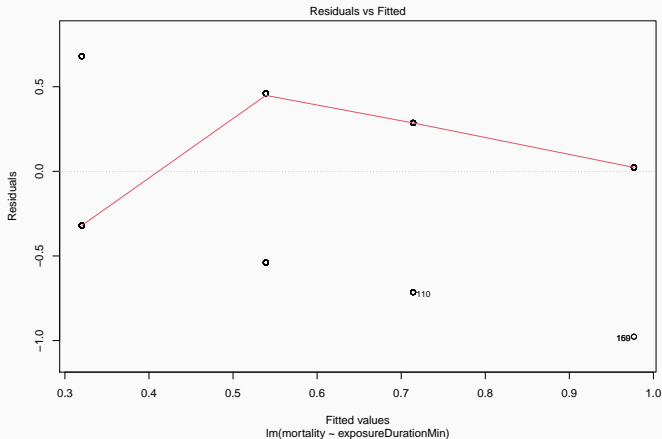
- Para valores muy bajos o muy altos de exposición, la mortalidad es  $< 0$  o  $> 1$



- Para valores muy bajos o muy altos de exposición, la mortalidad es  $< 0$  o  $> 1$
- ¿y los residuos?

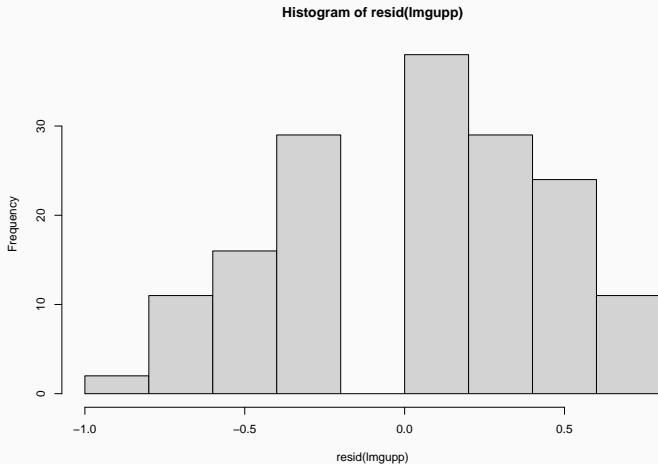
# Modelos lineales generalizados

- Para valores muy bajos o muy altos de exposición, la mortalidad es  $< 0$  o  $> 1$
- ¿y los residuos?



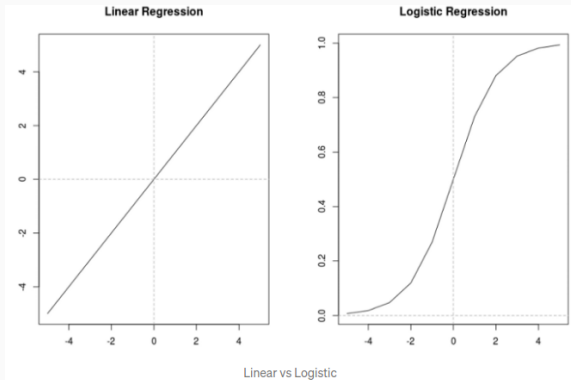
# Modelos lineales generalizados

- Para valores muy bajos o muy altos de exposición, la mortalidad es  $< 0$  o  $> 1$
- ¿y los residuos?

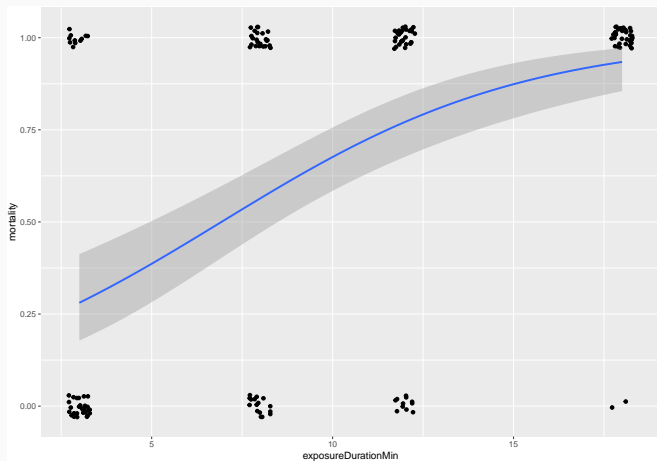


# Modelos lineales generalizados

En este caso, queremos modelar la probabilidad de mortalidad en función del tiempo de exposición a temperaturas bajas, con una función limitada entre 0 y 1



# Modelos lineales generalizados



# Modelos lineales generalizados

Este tipo de modelos, que permiten modelar respuestas *no normales*, se llaman **Modelos lineales generalizados** (*Generalized Linear Models*, GLM).

Tienen tres componentes:

- Distribución estadística de la variable respuesta

# Modelos lineales generalizados

Este tipo de modelos, que permiten modelar respuestas *no normales*, se llaman **Modelos lineales generalizados** (*Generalized Linear Models*, GLM).

Tienen tres componentes:

- Distribución estadística de la variable respuesta
- Variables predictoras

# Modelos lineales generalizados

Este tipo de modelos, que permiten modelar respuestas *no normales*, se llaman **Modelos lineales generalizados** (*Generalized Linear Models*, GLM).

Tienen tres componentes:

- Distribución estadística de la variable respuesta
- Variables predictoras
- Función de enlace



- En un modelo con variable respuesta **binaria**, la distribución es la distribución **binomial**.

- En un modelo con variable respuesta **binaria**, la distribución es la distribución **binomial**.
- Las variables predictoras son equivalentes a un modelo lineal.

- En un modelo con variable respuesta **binaria**, la distribución es la distribución **binomial**.
- Las variables predictoras son equivalentes a un modelo lineal.
- La función de enlace nos permite modelar nuestra respuesta  $a + b \cdot x_i$  en el intervalo  $[0, 1]$ , en vez de que tome cualquier valor entre  $[-\infty, \infty]$

# Modelos lineales generalizados

- Función de enlace

Usamos la función logística:

$$Pr(mortalidad_i) = \frac{e^{a+bx_i}}{1 + e^{a+bx_i}}$$

- La función enlace se aplica a la variable respuesta, por lo que reordenamos la ecuación previa:

$$\begin{aligned} Pr(mortalidad_i) &= p_i = g(a + bx_i) \\ g^{-1}(p_i) &= a + bx_i \end{aligned} \tag{1}$$

La función inversa de la logística se llama “logit”. Esta, por fin, es nuestra función de enlace:

$$\text{logit}(p_i) = a + bx_i$$

De esta manera, para cualquier valor de  $a$ ,  $b$ ,  $x_i$ , la respuesta estará acotada entre  $[0, 1]$ .

**Función de enlace:** Transforma la estimación del modelo para que se ajuste a la distribución de la variable respuesta.

# Modelos lineales generalizados

- Ya tenemos todos los ingredientes para ajustar nuestro primer GLM

```
glm1 <- glm(mortality ~ exposureDurationMin,  
            data = gupp,  
            family = "binomial")
```

que se corresponde con

$$\text{logit}(\text{Pr}(\text{mortalidad}_i)) = a + b \cdot \text{exposure}_i$$

# Modelos lineales generalizados

```
summary(glm1)
```

```
##
## Call:
## glm(formula = mortality ~ exposureDurationMin, family = "binomial",
##      data = gupp)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3332  -0.8115   0.3688   0.7206   1.5943
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.66081    0.40651  -4.086 4.40e-05 ***
## exposureDurationMin  0.23971    0.04245   5.646 1.64e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 209.55  on 159  degrees of freedom
## Residual deviance: 164.69  on 158  degrees of freedom
## AIC: 168.69
##
## Number of Fisher Scoring iterations: 4
```



# Modelos lineales generalizados

```
coef(glm1)
```

```
##          (Intercept) exposureDurationMin  
##          -1.6608075           0.2397113
```

**Estos coeficientes están en escala logit.** No se pueden interpretar como probabilidades de manera directa, sino que debemos “deshacer” la función de enlace para recuperar probabilidades estándar. La función inversa de la logit es la función logística, que se aplica en R con el comando `plogis`.

# Modelos lineales generalizados

Por ejemplo, si queremos saber la probabilidad de mortalidad de un pez en condiciones basales, sin exposición a temperaturas de 5°C, el modelo sería:

$$\begin{aligned} \text{logit}(y_i) &= a + b \cdot 0 = a \\ y_i &= \text{plogis}(a) \end{aligned} \tag{2}$$

En R:

```
a <- coef(glm1)[1]
plogis(a)
```

```
## (Intercept)
## 0.1596536
```

# Modelos lineales generalizados

O si queremos saber la probabilidad de mortalidad de un pez tras 12 minutos de exposición:

$$\text{logit}(y_i) = a + b \cdot 12$$

$$y_i = \text{plogis}(a + b \cdot 12)$$

```
a <- coef(glm1)[1]; b <- coef(glm1)[2]
plogis(a + b*12)
```

```
## (Intercept)
## 0.7713109
```

Si el modelo es apropiado, esta probabilidad debe ser similar a las probabilidades obtenidas directamente de los datos:

```
sum(gupp$mortality[gupp$exposureDurationMin == 12]) /
nrow(gupp[gupp$exposureDurationMin == 12,])
```

```
## [1] 0.725
```

# Modelos lineales generalizados

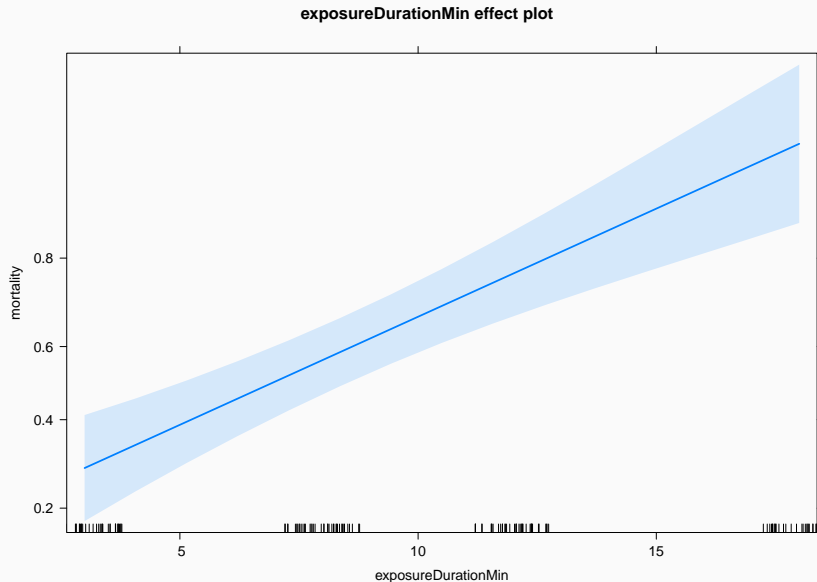
Interpretar resultados: El paquete `effects` da los coeficientes en probabilidades

```
library(effects)
allEffects(glm1)
```

```
## model: mortality ~ exposureDurationMin
##
## exposureDurationMin effect
## exposureDurationMin
##          3          6.8          10          14          18
## 0.2805624 0.4923079 0.6761874 0.8449003 0.9342568
```

# Modelos lineales generalizados

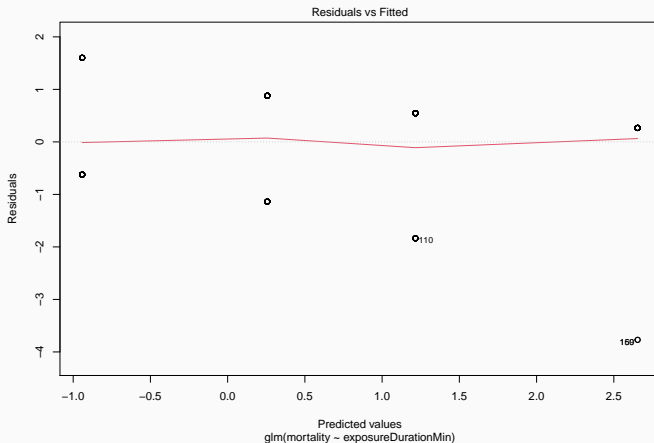
```
plot(allEffects(glm1))
```



# Modelos lineales generalizados

- Comprobación de los residuos del modelo

```
plot(glm1)
```

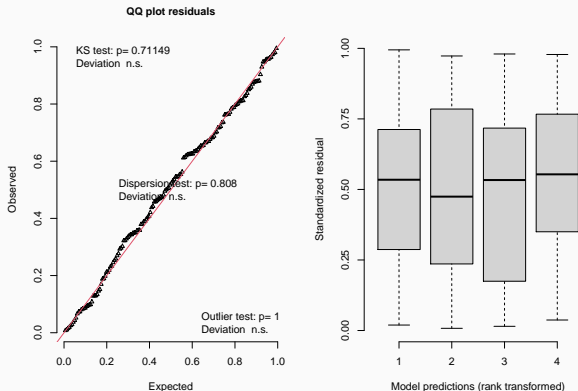


# Modelos lineales generalizados

- Comprobación de los residuos del modelo: paquete DHARMa

```
library(DHARMa)
simulateResiduals(glm1, plot = TRUE)
```

DHARMa residual diagnostics



Pasos para generar GLMs:

- Análisis exploratorio: Visualización de los datos



Pasos para generar GLMs:

- Análisis exploratorio: Visualización de los datos
- Ajuste del modelo (cuidado con el argumento “family”!)

Pasos para generar GLMs:

- Análisis exploratorio: Visualización de los datos
- Ajuste del modelo (cuidado con el argumento “family”!)
- Comprobación: `summary`, residuos (e.g. con DHARMA)

Pasos para generar GLMs:

- Análisis exploratorio: Visualización de los datos
- Ajuste del modelo (cuidado con el argumento “family”!)
- Comprobación: `summary`, residuos (e.g. con DHARMA)
- Transformar coeficientes (e.g. con `allEffects`)

Pasos para generar GLMs:

- Análisis exploratorio: Visualización de los datos
- Ajuste del modelo (cuidado con el argumento “family”!)
- Comprobación: `summary`, residuos (e.g. con DHARMA)
- Transformar coeficientes (e.g. con `allEffects`)
- Visualizar modelo (e.g. con `allEffects` o `visreg`)

# Modelos lineales generalizados

Los modelos de regresión logística se pueden aplicar también a datos de proporciones

```
gupp.prop <- gupp %>%  
  group_by(exposureDurationMin) %>%  
  summarise(alive = sum(mortality == 0),  
            dead = sum(mortality == 1))
```

```
gupp.prop
```

```
## # A tibble: 4 x 3  
##   exposureDurationMin alive  dead  
##           <int> <int> <int>  
## 1             3     29    11  
## 2             8     16    24  
## 3            12     11    29  
## 4            18      2    38
```

# Modelos lineales generalizados

Ajustamos el modelo usando `cbind(positivos, negativos)` como variable respuesta. En este caso, la probabilidad es de mortalidad, por lo que nuestro “positivo” es el número de muertes.

```
glm.prop <- glm(cbind(dead,alive) ~ exposureDurationMin,  
               data = gupp.prop,  
               family = "binomial")
```

```
coef(glm1)
```

```
##           (Intercept) exposureDurationMin  
##           -1.6608075           0.2397113
```

```
coef(glm.prop)
```

```
##           (Intercept) exposureDurationMin  
##           -1.6608075           0.2397113
```

# Modelos lineales generalizados

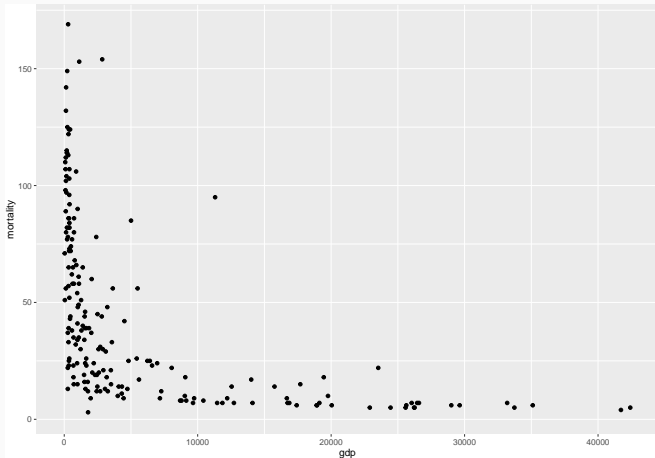
- Otro ejemplo con datos de proporciones

```
gdp <- read.csv(here::here("datasets",  
                           "UN_GDP_infantmortality.csv"))  
head(gdp)
```

##	country	mortality	gdp
## 1	Afghanistan	154	2848
## 2	Albania	32	863
## 3	Algeria	44	1531
## 4	American.Samoa	11	NA
## 5	Andorra	NA	NA
## 6	Angola	124	355

# Modelos lineales generalizados

```
ggplot(gdp, aes(x = gdp, y = mortality)) +  
  geom_point()
```





# Modelos lineales generalizados

```
gdp.glm <- glm(cbind(mortality, 1000 - mortality) ~ gdp,  
               data = gdp, family = binomial)
```

# Modelos lineales generalizados

```
summary(gdp.glm)
```

```
##
## Call:
## glm(formula = cbind(mortality, 1000 - mortality) ~ gdp, family = binomial,
##      data = gdp)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -9.2230  -3.5163  -0.5697   2.4284  13.5849
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.657e+00  1.311e-02 -202.76  <2e-16 ***
## gdp          -1.279e-04  3.458e-06  -36.98  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 6430.2  on 192  degrees of freedom
## Residual deviance: 3530.2  on 191  degrees of freedom
##      (14 observations deleted due to missingness)
## AIC: 4525.8
##
## Number of Fisher Scoring iterations: 5
```

# Modelos lineales generalizados

Coeficientes:

```
allEffects(gdp.glm)
```

```
## model: cbind(mortality, 1000 - mortality) ~ gdp
```

```
##
```

```
## gdp effect
```

```
## gdp
```

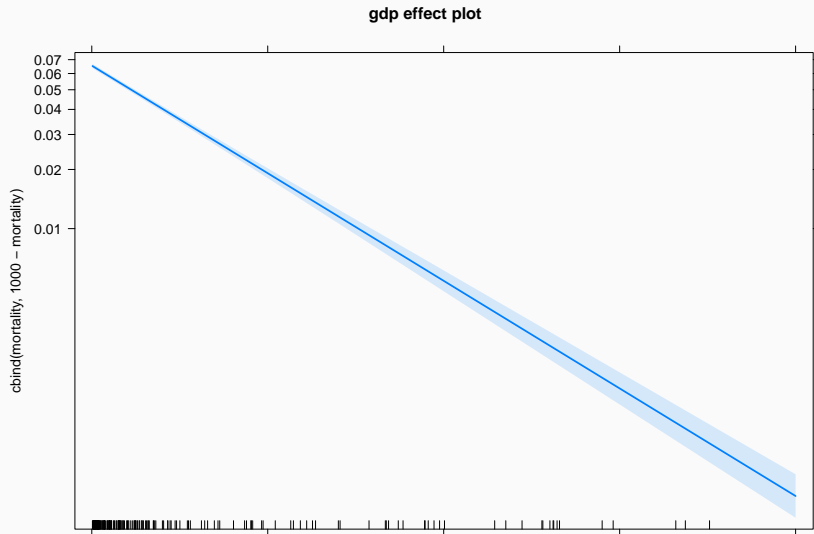
```
##           40           10000           20000           30000           40000
```

```
## 0.0652177296 0.0191438829 0.0054028095 0.0015096074 0.0004206154
```

# Modelos lineales generalizados

Visualización del modelo:

```
plot(allEffects(gdp.glm))
```

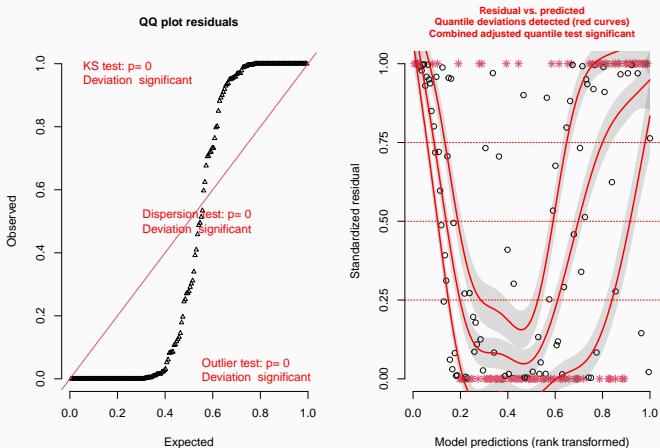


# Modelos lineales generalizados

Residuos:

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

DHARMA residual diagnostics



# Modelos lineales generalizados

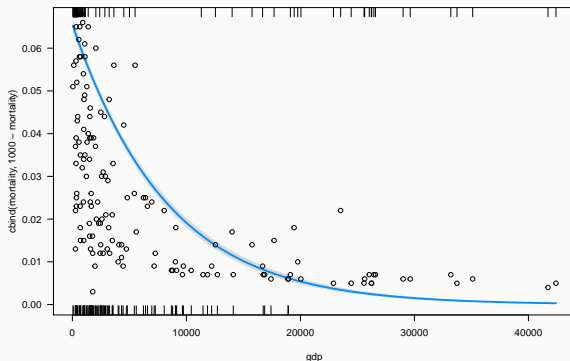
Welcome to the real world!



# Modelos lineales generalizados

Este patrón en los residuos indica **sobredispersión**. Los datos están más dispersos de lo que esperaríamos según el modelo. En este caso, para un gdp determinado, hay una variación muy grande en mortalidad infantil.

```
visreg(gdp.glm, scale = "response")  
points(mortality/1000 ~ gdp, data = gdp)
```



# Modelos lineales generalizados

Podemos comprobar la sobredispersión (o infradispersión) de manera explícita con DHARMA:

```
simres <- simulateResiduals(gdp.glm, refit = TRUE)
testDispersion(simres, plot = FALSE)
```

```
##
```

```
## DHARMA nonparametric dispersion test via mean deviance residual fit
```

```
## vs. simulated-refitted
```

```
##
```

```
## data: simres
```

```
## dispersion = 21, p-value < 2.2e-16
```

```
## alternative hypothesis: two.sided
```



# Modelos lineales generalizados

La sobredispersión se puede tratar explícitamente escogiendo otra distribución para la variable respuesta. En este caso, la distribución *quasibinomial* ayuda a modelar esta varianza extra

```
gdp.glm.qb <- glm(cbind(mortality, 1000 - mortality) ~ gdp,  
                  data = gdp, family = quasibinomial)
```

# Modelos lineales generalizados

- Los valores medios de los coeficientes se mantienen con respecto al modelo binomial

```
allEffects(gdp.glm)
```

```
## model: cbind(mortality, 1000 - mortality) ~ gdp
##
## gdp effect
## gdp
##           40           10000           20000           30000           40000
## 0.0652177296 0.0191438829 0.0054028095 0.0015096074 0.0004206154
```

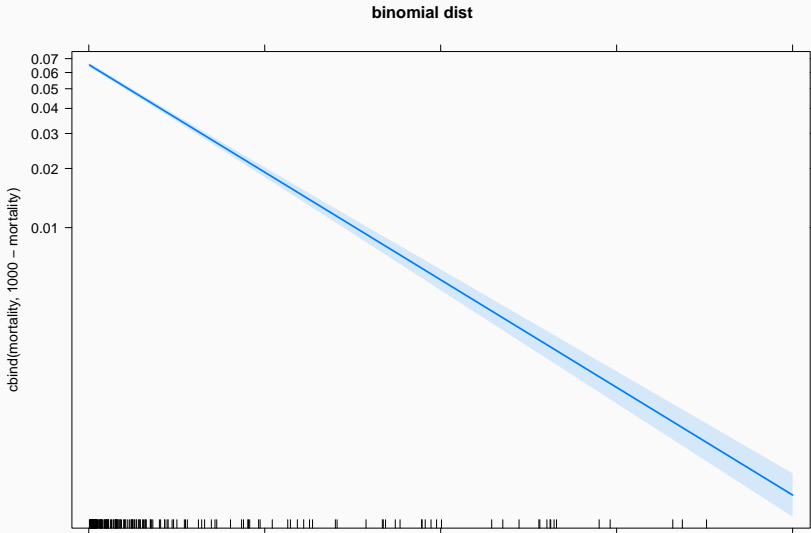
```
allEffects(gdp.glm.qb)
```

```
## model: cbind(mortality, 1000 - mortality) ~ gdp
##
## gdp effect
## gdp
##           40           10000           20000           30000           40000
## 0.0652177296 0.0191438829 0.0054028095 0.0015096074 0.0004206154 39
```

# Modelos lineales generalizados

- Pero los errores asociados sí varían

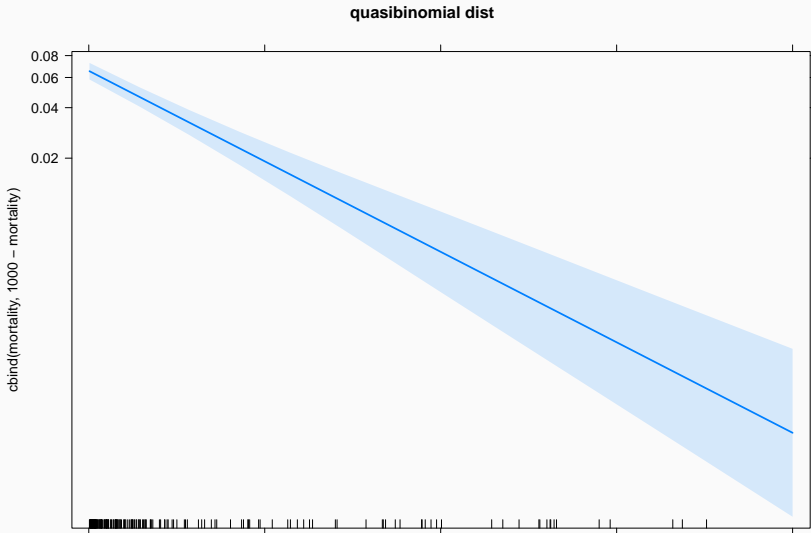
```
plot(allEffects(gdp.glm),main = "binomial dist")
```



# Modelos lineales generalizados

- Pero los errores asociados sí varían

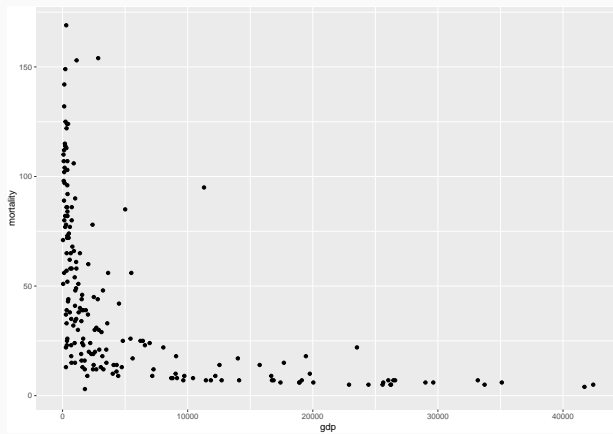
```
plot(allEffects(gdp.glm.qb),main = "quasibinomial dist")
```



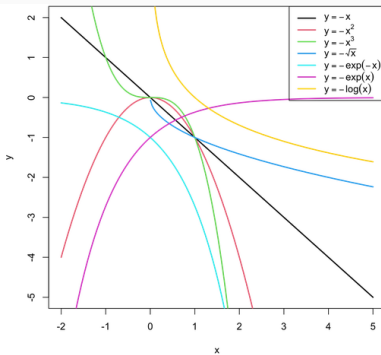
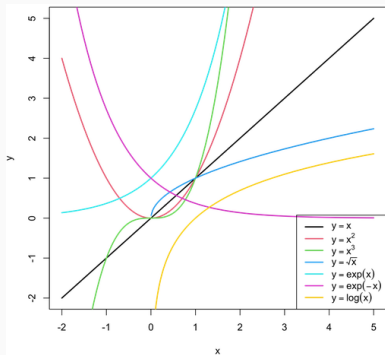
# Modelos lineales generalizados

Más allá de la solución concreta, este ejemplo nos ayuda a pensar en la forma de las relaciones entre variables. No todas las relaciones son de naturaleza lineal

```
ggplot(gdp, aes(x = gdp, y = mortality)) +  
  geom_point()
```



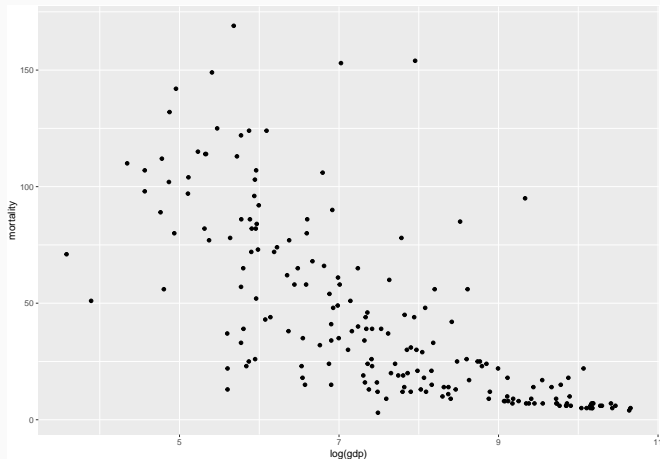
# Modelos lineales generalizados



# Modelos lineales generalizados

A veces es conveniente transformar la variable respuesta para acercarnos a una relación lineal

```
ggplot(gdp, aes(x = log(gdp), y = mortality)) +  
  geom_point()
```



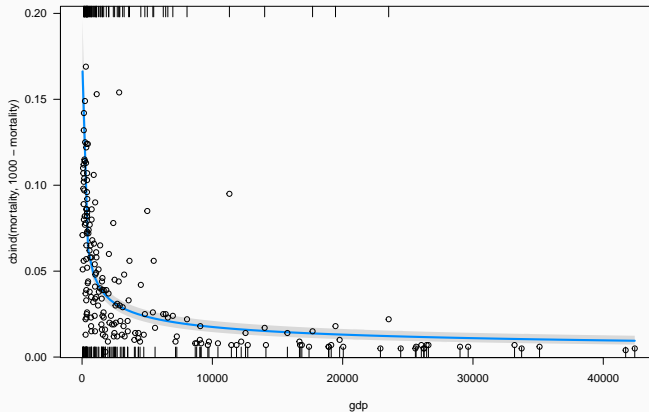
# Modelos lineales generalizados

```
gdp.glm.log <- glm(cbind(mortality, 1000 - mortality) ~ log(gdp),  
  data = gdp, family = quasibinomial)
```



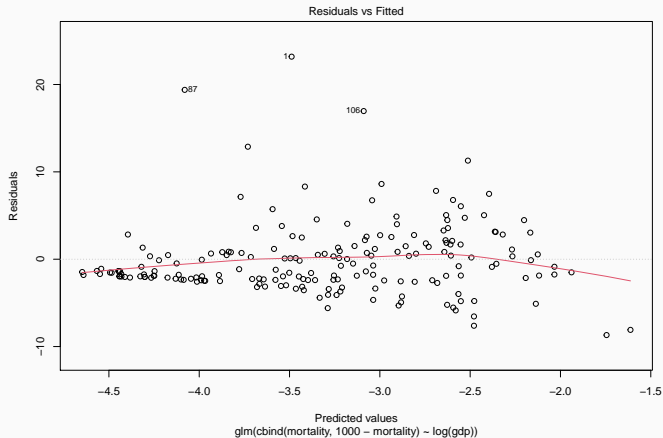
# Modelos lineales generalizados

```
visreg(gdp.glm.log, scale = "response")  
points(mortality/1000 ~ gdp, data = gdp)
```



# Modelos lineales generalizados

```
plot(gdp.glm.log)
```



Este último modelo sigue sin ser ideal, pero con datos reales, a veces no es fácil llegar a modelos *perfectos*

- Ya conocemos la distribución normal  $Y \sim N(\mu, \sigma^2)$ , que es una distribución continua, y la binomial, que es una distribución discreta. Hay muchas otras distribuciones que podemos considerar para modelar datos ecológicos.

- Ya conocemos la distribución normal  $Y \sim N(\mu, \sigma^2)$ , que es una distribución continua, y la binomial, que es una distribución discreta. Hay muchas otras distribuciones que podemos considerar para modelar datos ecológicos.
- Uno de los tipos de datos más comunes que nos encontraremos son datos de conteos

# Modelos lineales generalizados

- Ya conocemos la distribución normal  $Y \sim N(\mu, \sigma^2)$ , que es una distribución continua, y la binomial, que es una distribución discreta. Hay muchas otras distribuciones que podemos considerar para modelar datos ecológicos.
- Uno de los tipos de datos más comunes que nos encontraremos son datos de conteos

```
seedlings <- read.csv(here::here("datasets", "seedlings.csv"))  
head(seedlings)
```

```
##   X count row col   light area  
## 1 1     0   1   1 70.71854 0.50  
## 2 2     1   1   2 88.26021 0.25  
## 3 3     2   1   3 67.35133 0.50  
## 4 4     3   1   4 67.57850 1.00  
## 5 5     4   1   5 26.63098 0.25  
## 6 6     3   1   6 15.79433 1.00
```

- Los datos de conteos son datos discretos en el intervalo  $[0, 1, \dots, n]$
- Estos datos se modelan con la distribución de **Poisson**: una distribución discreta que expresa la probabilidad de un número de eventos ocurriendo en un intervalo fijo (espacial o temporal), suponiendo que estos eventos ocurren con una tasa media constante, y de manera independiente entre eventos.

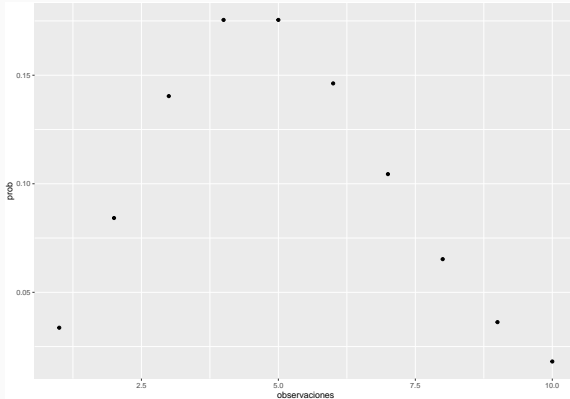
Por ejemplo, pensad en alguien que salga a menudo al campo a observar aves, y anote el tiempo que tarda entre cada observación.

- Podemos asumir que la tasa media de observaciones por hora es constante, porque nuestro observador siempre va a la misma zona y en la misma época del año.
- Nuestro observador, fijándose en sus notas, concluye que, de media, observa 5 aves por hora.
- Podemos preguntarnos cuál es la probabilidad de observar un número  $X$  de aves por hora, dada esta tasa media. Esto es justamente lo que nos dice la distribución de Poisson

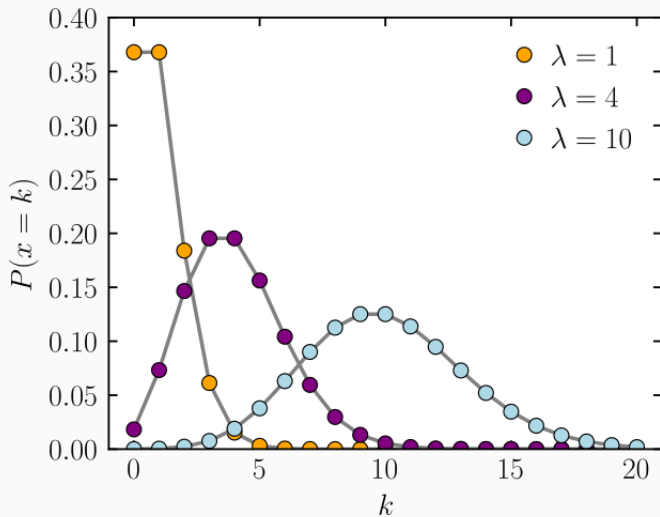


# Modelos lineales generalizados

```
prob.poisson <- dpois(x = 1:10, lambda = 5)
prob.df <- data.frame(observaciones = 1:10,
                      prob = prob.poisson)
ggplot(prob.df, aes(x = observaciones, y = prob)) +
  geom_point()
```



# Modelos lineales generalizados



En el ejemplo del dataset `seedlings.csv`, queremos modelar *el número de seedlings observados* en función de varios parámetros. Usando un modelo lineal generalizado con distribución Poisson, asumimos:

- que el número medio de seedlings observado es constante
- que las observaciones son independientes: observar un seedling en un punto determinado no influye en cualquier otra observación

Estas asunciones pueden parecer muy restrictivas, pero son necesarias para hacer inferencia.

Para ajustar un GLM Poisson, necesitamos los mismos ingredientes que para el GLM binomial:

- Distribución de la variable respuesta: **Poisson**

Para ajustar un GLM Poisson, necesitamos los mismos ingredientes que para el GLM binomial:

- Distribución de la variable respuesta: **Poisson**
- Variables independientes (numéricas o categóricas)

Para ajustar un GLM Poisson, necesitamos los mismos ingredientes que para el GLM binomial:

- Distribución de la variable respuesta: **Poisson**
- Variables independientes (numéricas o categóricas)
- Función de enlace: La más común es el **logaritmo** ¿por qué? la media debe ser *positiva*!

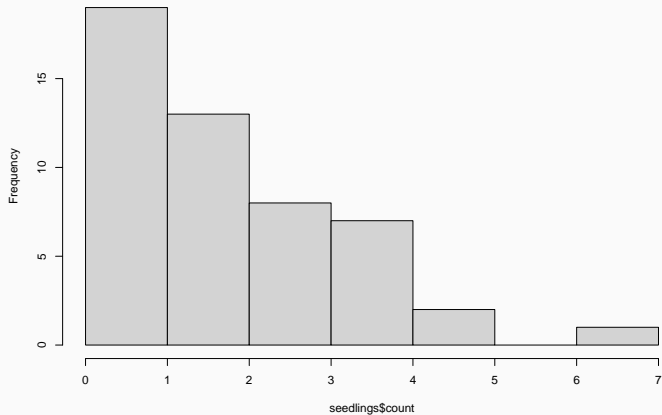
- Función de enlace para GLM Poisson:

$$\ln(\mu_i) = a + b \cdot x_i \Leftrightarrow \mu_i = e^{a+b \cdot x_i}$$

# Modelos lineales generalizados

```
hist(seedlings$count)
```

Histogram of seedlings\$count





# Modelos lineales generalizados

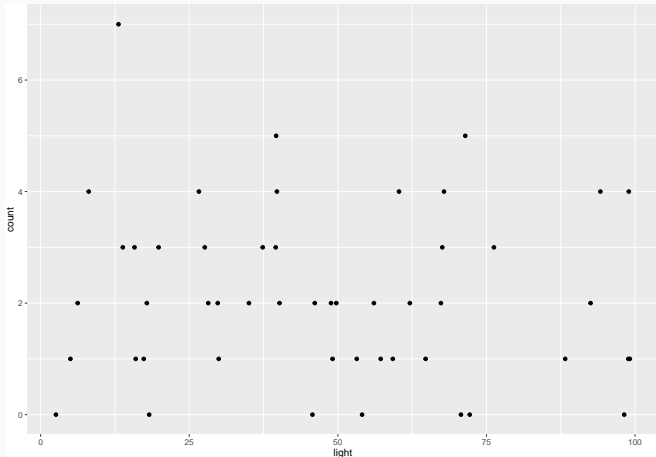
```
head(seedlings)
```

```
##      X count row col      light area
## 1 1      0   1   1 70.71854 0.50
## 2 2      1   1   2 88.26021 0.25
## 3 3      2   1   3 67.35133 0.50
## 4 4      3   1   4 67.57850 1.00
## 5 5      4   1   5 26.63098 0.25
## 6 6      3   1   6 15.79433 1.00
```

¿Hay relación entre el número de seedlings y la radiación solar?

# Modelos lineales generalizados

```
ggplot(seedlings, aes(x = light, y = count)) +  
  geom_point()
```



# Modelos lineales generalizados

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

# Modelos lineales generalizados

```
summary(seed1.glm)
```

```
##
## Call:
## glm(formula = count ~ light, family = poisson, data = seedlings)
##
## 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
```

# Modelos lineales generalizados

- Al igual que con el modelo binomial, aquí tenemos que transformar los coeficientes para interpretarlos. Los que devuelve el modelo están en escala logarítmica

```
coef(seed1.glm)
```

```
## (Intercept)      light  
## 0.881805022 -0.002575656
```

- Les aplicamos el inverso del logaritmo: la exponencial

```
exp(coef(seed1.glm))
```

```
## (Intercept)      light  
## 2.4152554 0.9974277
```

El número medio de seedlings observados es 2.4152554. Cada incremento de una unidad en radiación tiene un efecto multiplicativo de 0.9974277 sobre la media.

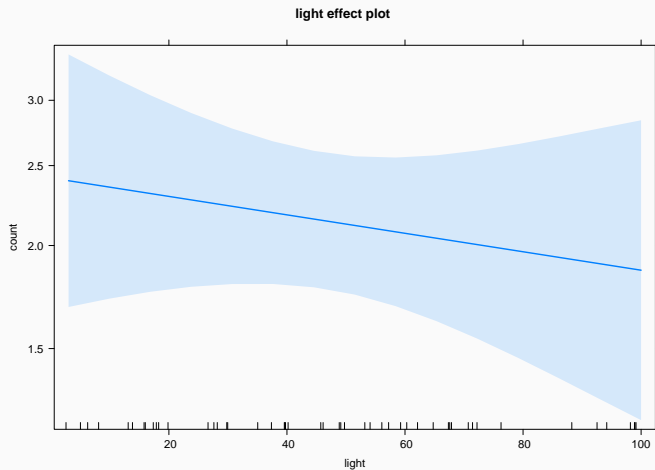
# Modelos lineales generalizados

```
summary(allEffects(seed1.glm))
```

```
## model: count ~ light
##
## light effect
## light
##      3      30      50      70     100
## 2.396665 2.235657 2.123408 2.016794 1.866826
##
## Lower 95 Percent Confidence Limits
## light
##      3      30      50      70     100
## 1.684579 1.795202 1.753373 1.567785 1.228247
##
## Upper 95 Percent Confidence Limits
## light
##      3      30      50      70     100
## 3.409754 2.784179 2.571535 2.594398 2.837408
```

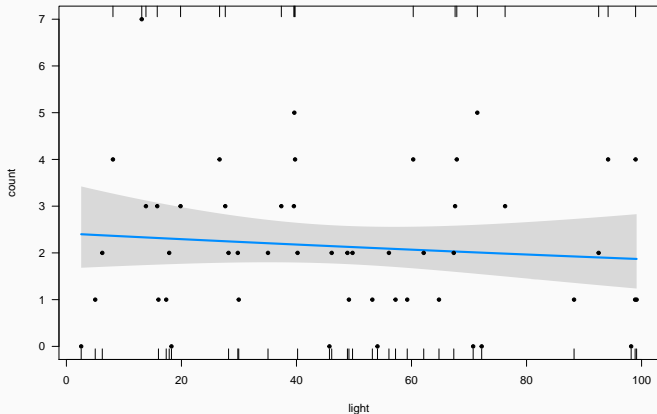
# Modelos lineales generalizados

```
plot(allEffects(seed1.glm))
```



# Modelos lineales generalizados

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



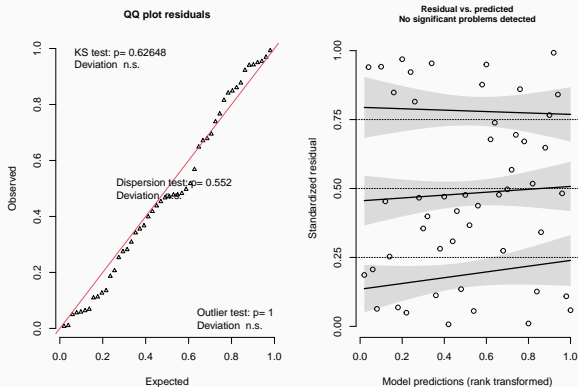


# Modelos lineales generalizados

- Como siempre... comprobación de residuos

```
DHARMA::simulateResiduals(seed1.glm, plot = TRUE)
```

DHARMA residual diagnostics



# Modelos lineales generalizados

- El modelo es razonable, y nos indica que *no* hay diferencias estadísticas entre la cantidad de seedlings observados con diferentes niveles de radiación.
- En datos de conteos, la sobredispersión es bastante común. Podemos asegurarnos de que no es el caso, de nuevo usando el paquete DHARMA

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

```
##
```

```
## DHARMA nonparametric dispersion test via mean deviance residual fit
```

```
## vs. simulated-refitted
```

```
##
```

```
## data:  simres
```

```
## dispersion = 1.1655, p-value = 0.432
```

```
## alternative hypothesis: two.sided
```

- Al igual que para el modelo binomial, si observamos sobredispersión en nuestro modelo podemos modificar la distribución. Para un modelo de conteos, dos opciones son la `quasipoisson` y la binomial negativa (esta última es más robusta, pero necesita la función `glm.nb` del paquete MASS)

# Modelos lineales generalizados

```
head(seedlings)
```

```
##      X count row col      light area
## 1 1      0   1   1 70.71854 0.50
## 2 2      1   1   2 88.26021 0.25
## 3 3      2   1   3 67.35133 0.50
## 4 4      3   1   4 67.57850 1.00
## 5 5      4   1   5 26.63098 0.25
## 6 6      3   1   6 15.79433 1.00
```

- ¿y si las observaciones fueron tomadas en plots de diferentes áreas?  
Por definición, en plots de áreas más grandes (e.g.  $1\text{m}^2$ ) será más probable observar seedlings que en plots más pequeños (e.g.  $0.25\text{m}^2$ ).
- Otro ejemplo: número de casos de cáncer en cuatro ciudades danesas. Cada ciudad tiene un tamaño diferente, por lo que tiene sentido modelar la media de casos observados *en función del tamaño de la población*.

```
cancer.data <- read.csv(here::here("datasets","cancer_data.csv"))
head(cancer.data)
```

- Seedlings:

```
seedl.offset <- glm(count ~ light,  
                    data = seedlings,  
                    offset = seedlings$area)
```

# Modelos lineales generalizados

Los coeficientes ahora vienen referidos *por unidad de área*

```
exp(coef(seed1.offset))
```

```
## (Intercept)      light  
##    6.6098971    0.9923444
```

Cancer:

- distribuciones continuas y discretas



- distribuciones continuas y discretas
- likelihood (WS p814)

- distribuciones continuas y discretas
- likelihood (WS p814)
- esquema general: distribución de residuos, fórmula, función de enlace

- distribuciones continuas y discretas
- likelihood (WS p814)
- esquema general: distribución de residuos, fórmula, función de enlace
- regresión logística (WS p701)

- distribuciones continuas y discretas
- likelihood (WS p814)
- esquema general: distribución de residuos, fórmula, función de enlace
- regresión logística (WS p701)
- regresión de conteos (poisson, negbin)

- distribuciones continuas y discretas
- likelihood (WS p814)
- esquema general: distribución de residuos, fórmula, función de enlace
- regresión logística (WS p701)
- regresión de conteos (poisson, negbin)
- selección de modelos (AIC)