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

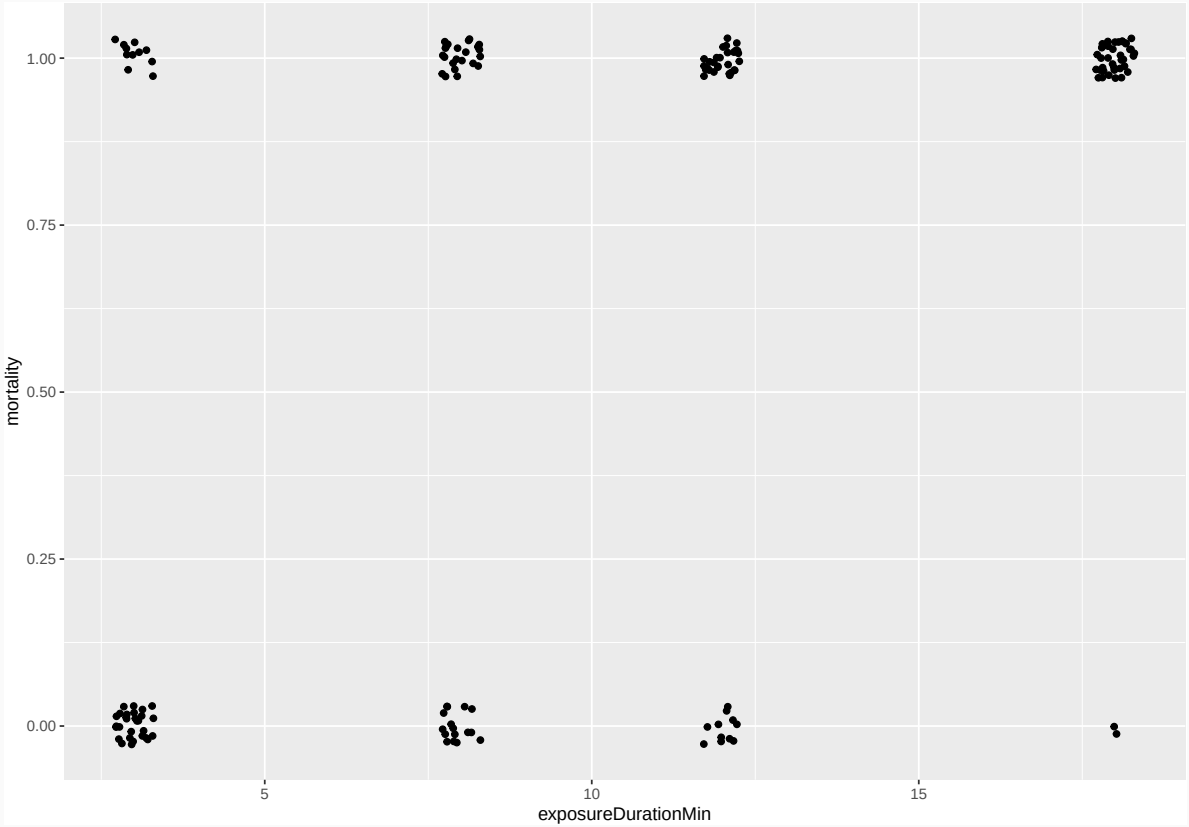
- Hasta ahora: modelos lineales con variable respuesta **continua** y residuos **normales**
- ¿podemos modelar variables con respuestas discretas? Por ejemplo, mortalidad de peces en función de tiempo de exposición a temperaturas de 5ºC:

```
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))
```



Modelos lineales generalizados

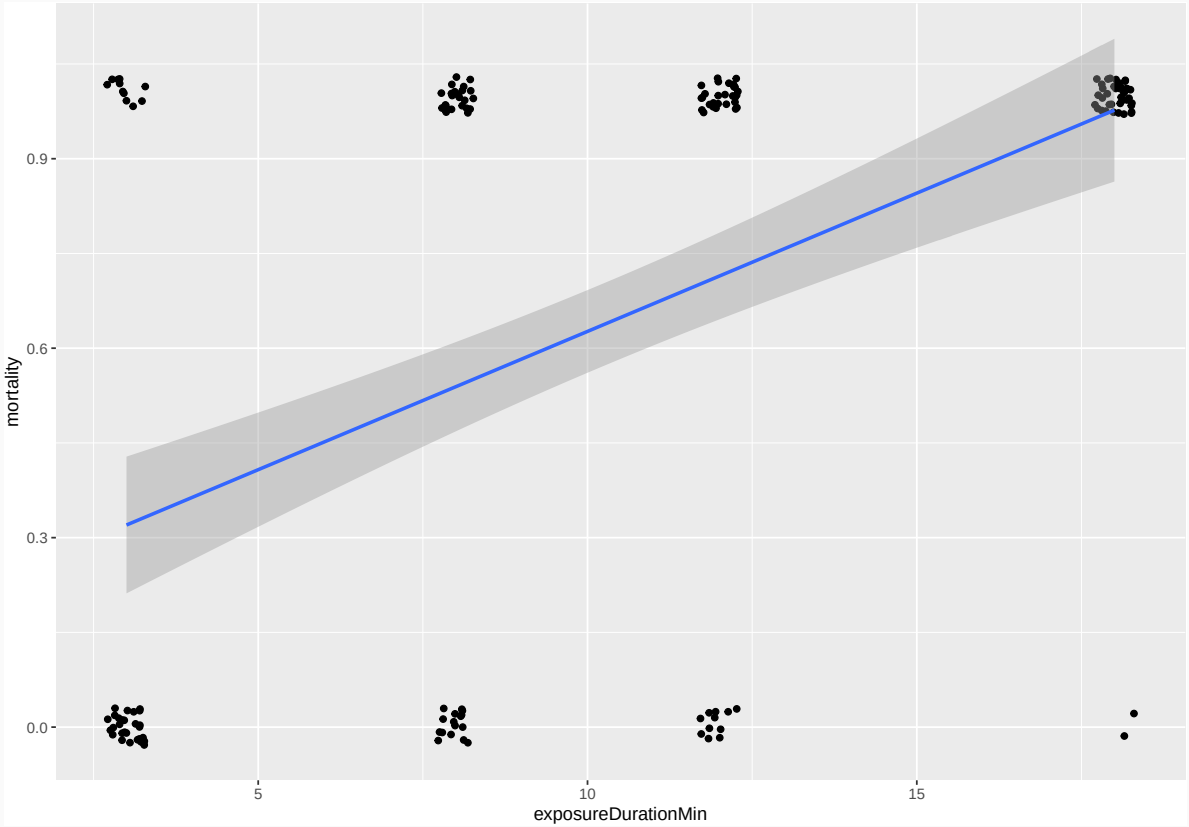
- ¿Podemos aplicar una regresión lineal a estos datos?
- ¿la relación entre X e Y es lineal?

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")
```



Modelos lineales generalizados

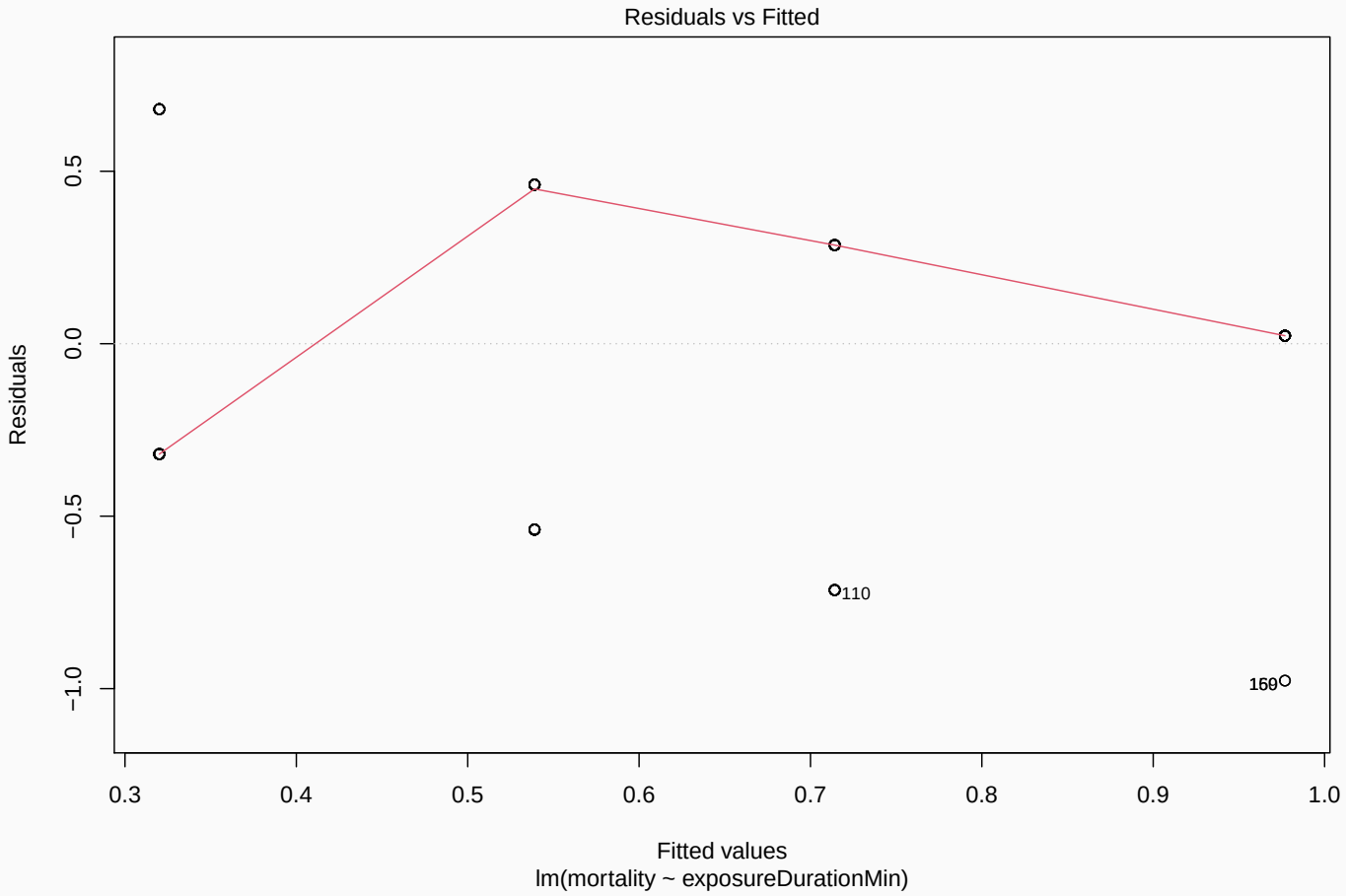
- Para valores muy bajos o muy altos de exposición, la mortalidad es $<$ **0** o $>$ **1**

Modelos lineales generalizados

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

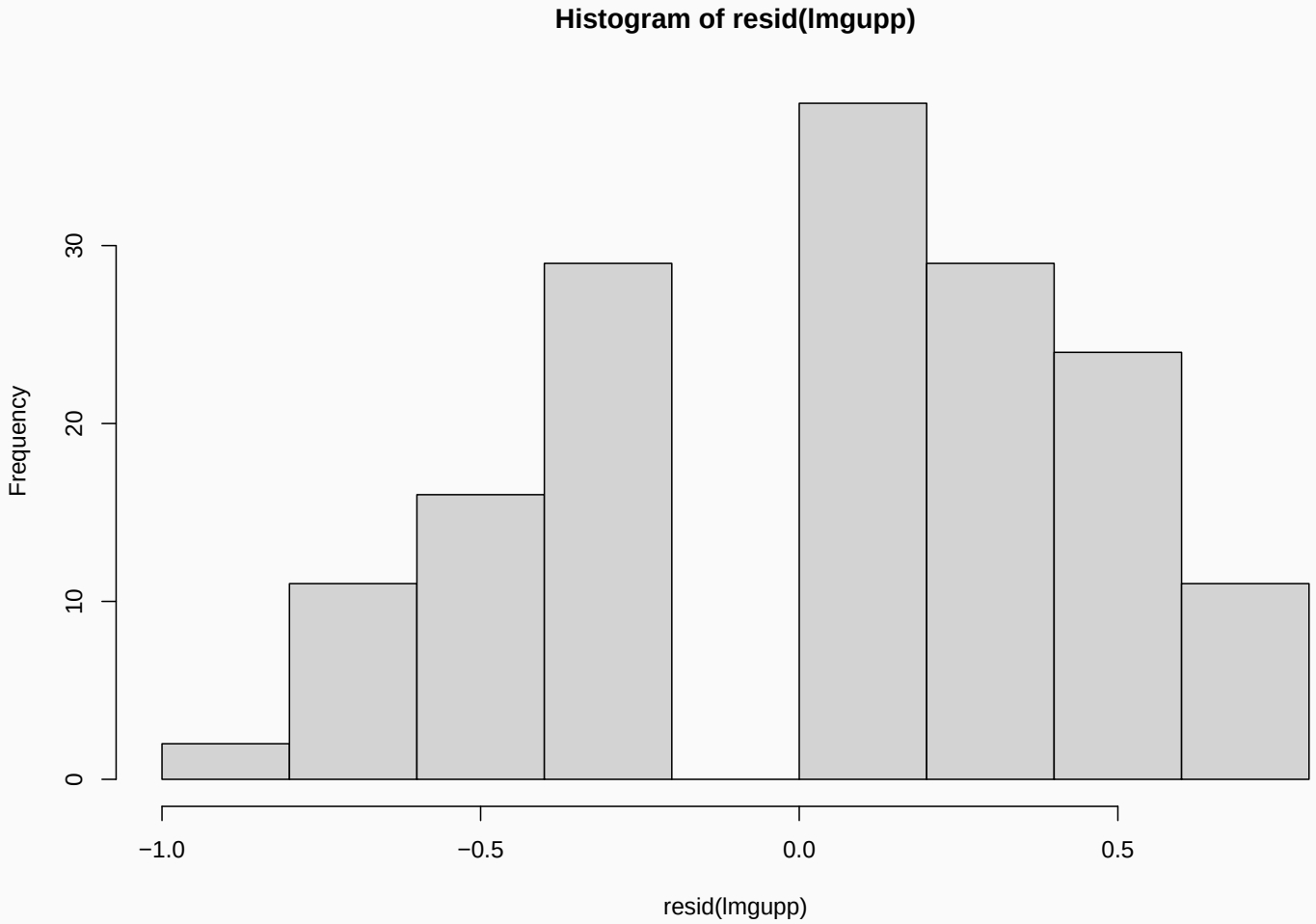
Modelos lineales generalizados

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



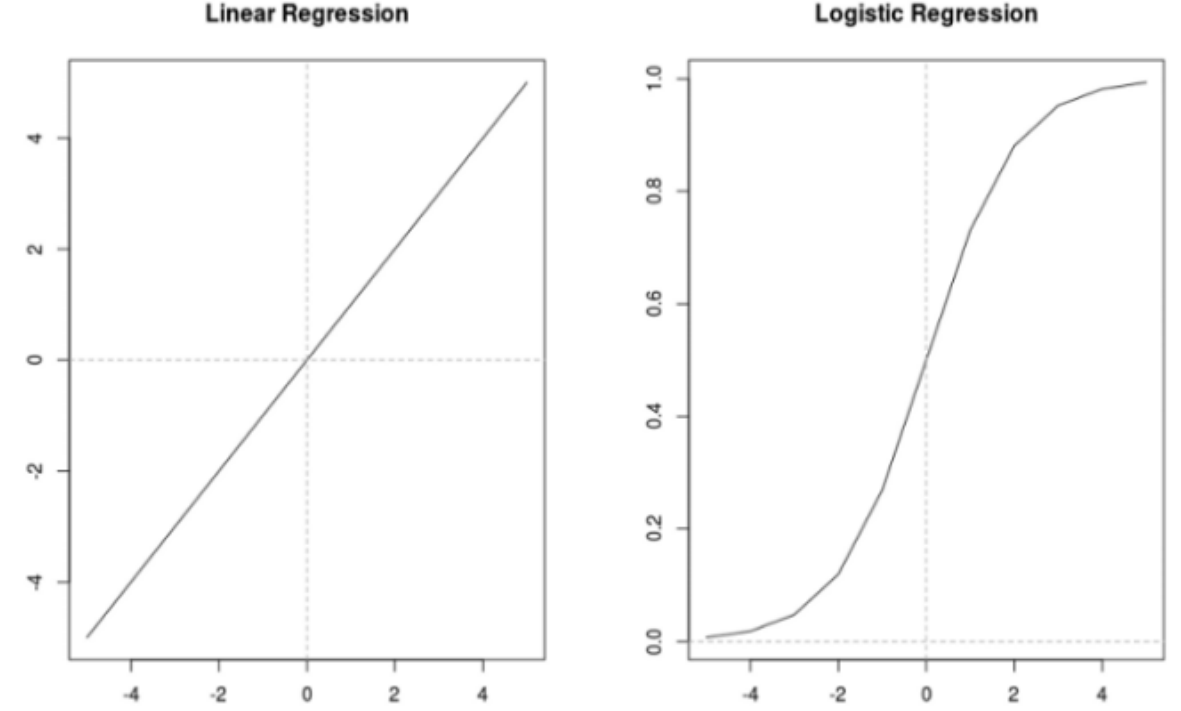
Modelos lineales generalizados

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

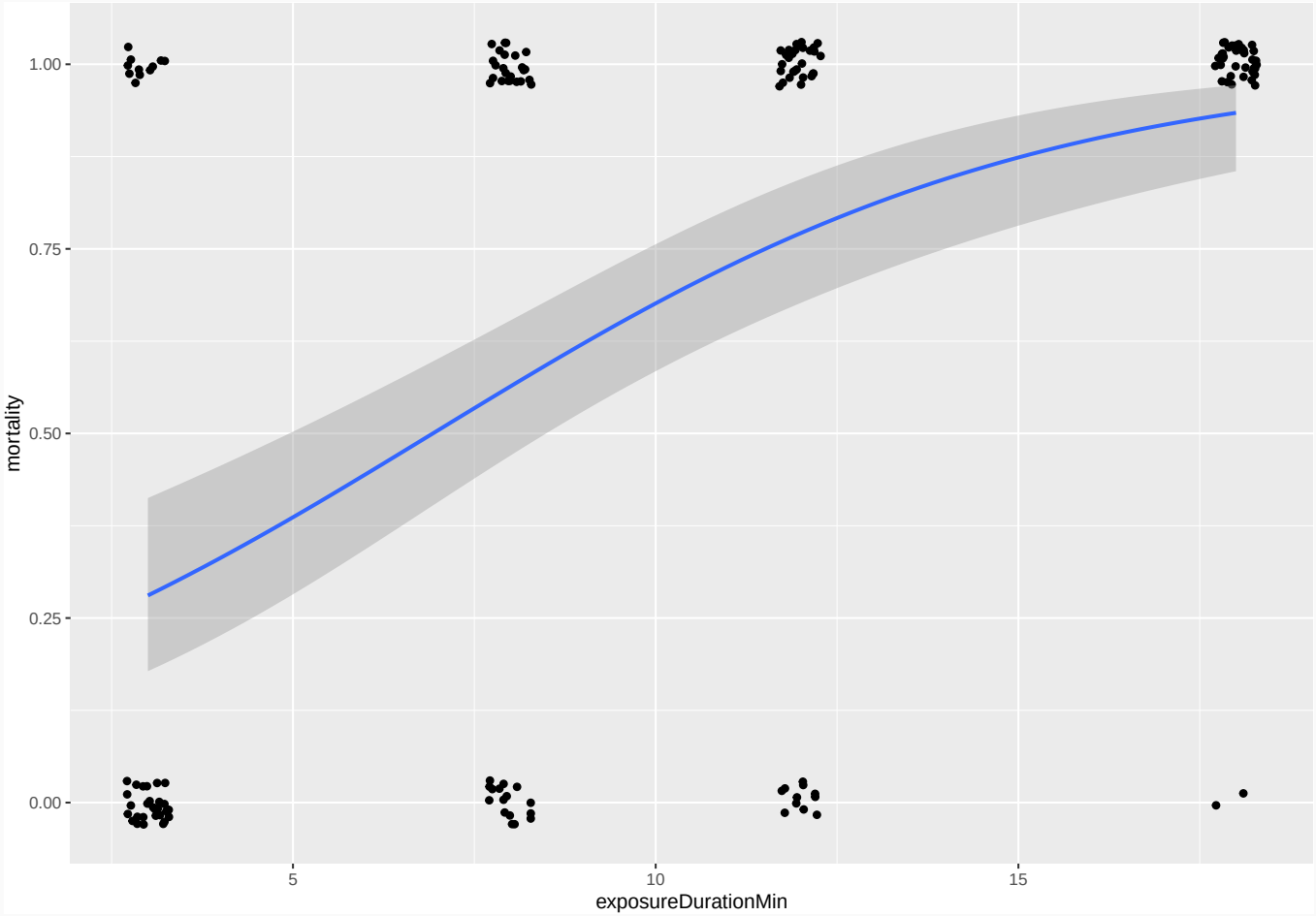


Modelos lineales generalizados

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

Modelos lineales generalizados

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

Modelos lineales generalizados

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

Modelos lineales generalizados

- 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:

$$Pr(mortalidad_i) = p_i = g(a + bx_i)$$
$$g^{-1}(p_i) = a + bx_i \quad (1)$$

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

$$logit(p_i) = a + bx_i$$

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

Modelos lineales generalizados

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

$$logit(Pr(mortalidad_i)) = a + b \cdot exposure_i$$

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 5ºC, el modelo sería:

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

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:

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

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

```
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

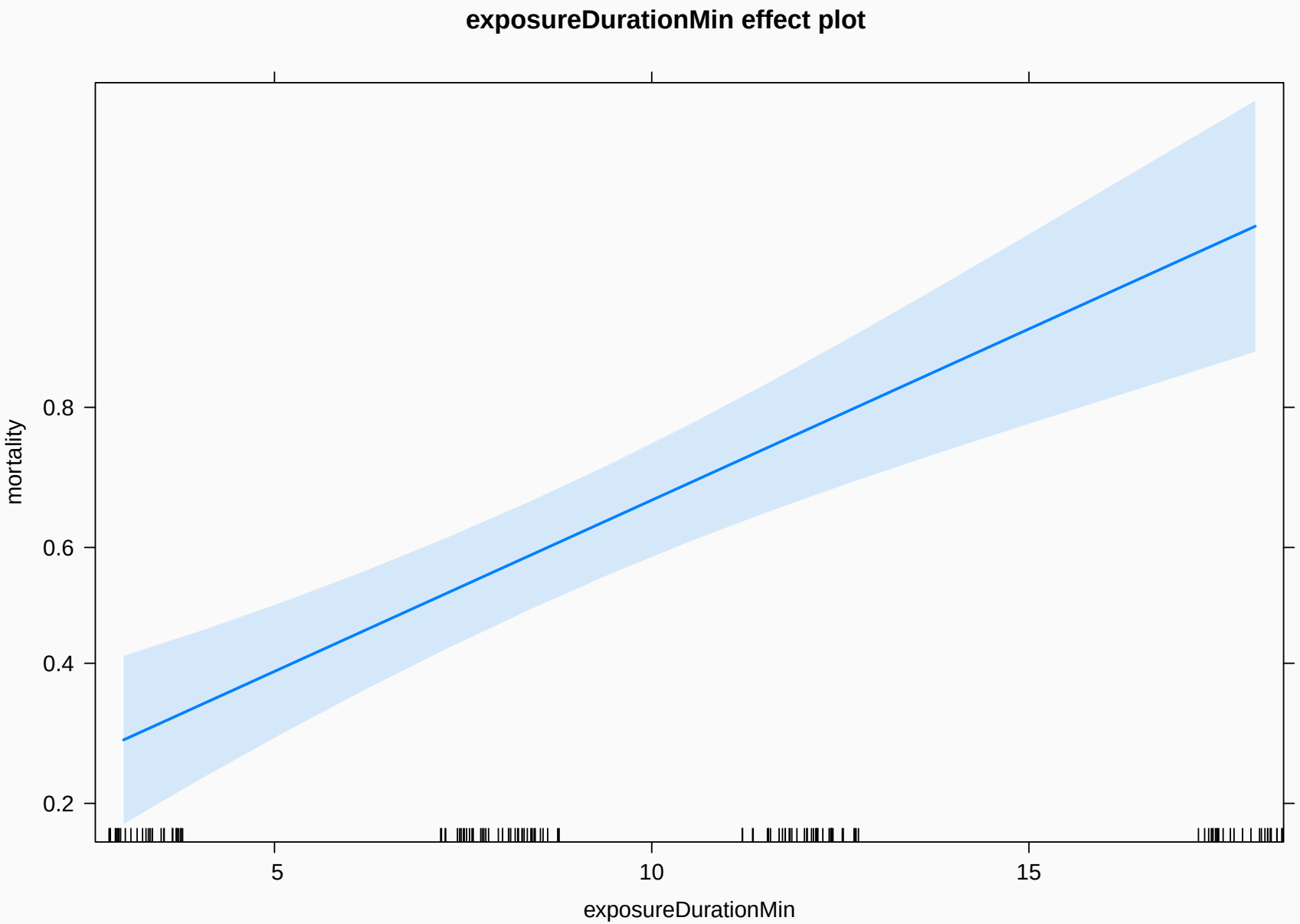
Interpretar resultados: El paquete `effects` da los coeficientes en probabilidades

```
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

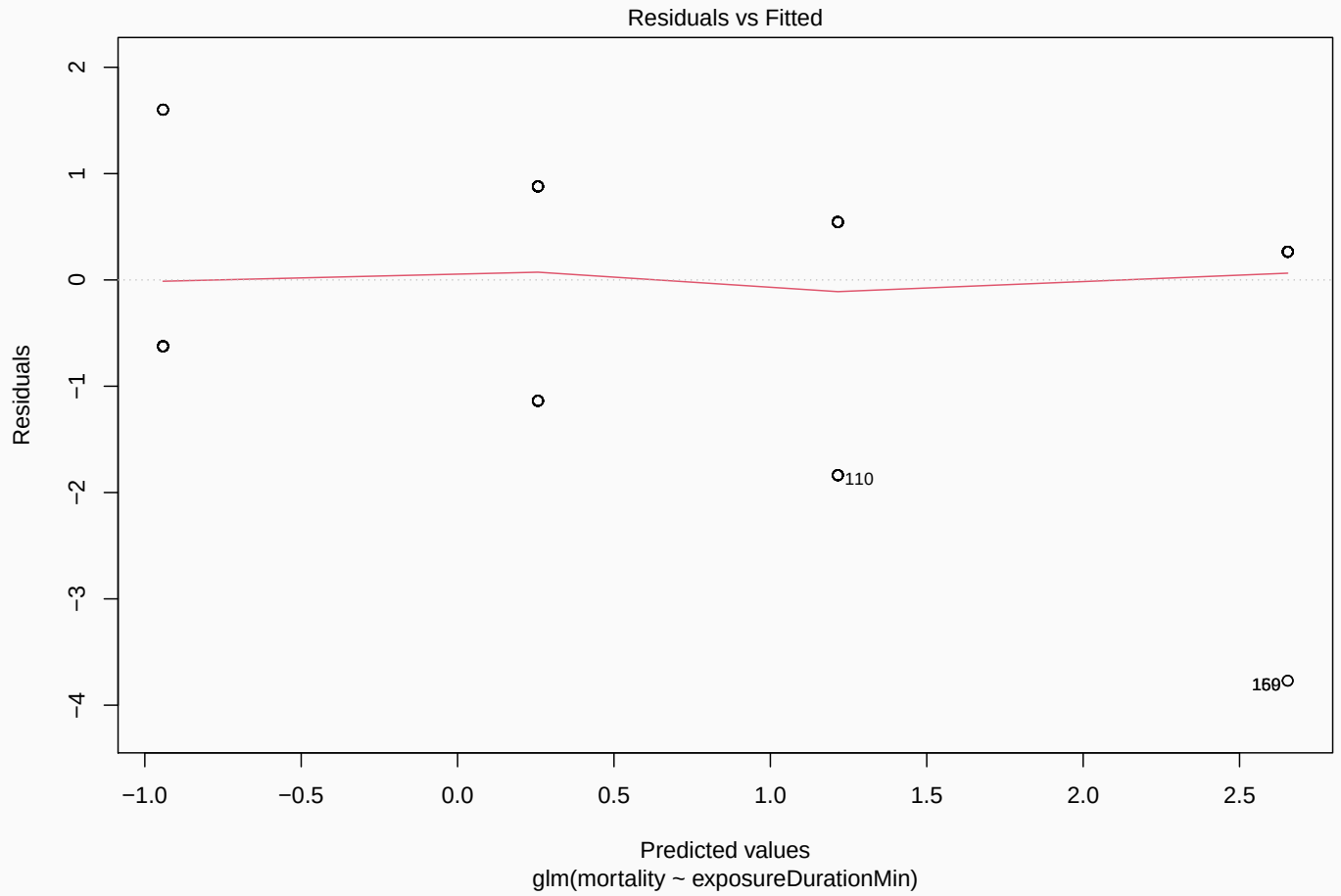
```
plot(allEffects(glm1))
```



Modelos lineales generalizados

- Comprobación de los residuos del modelo

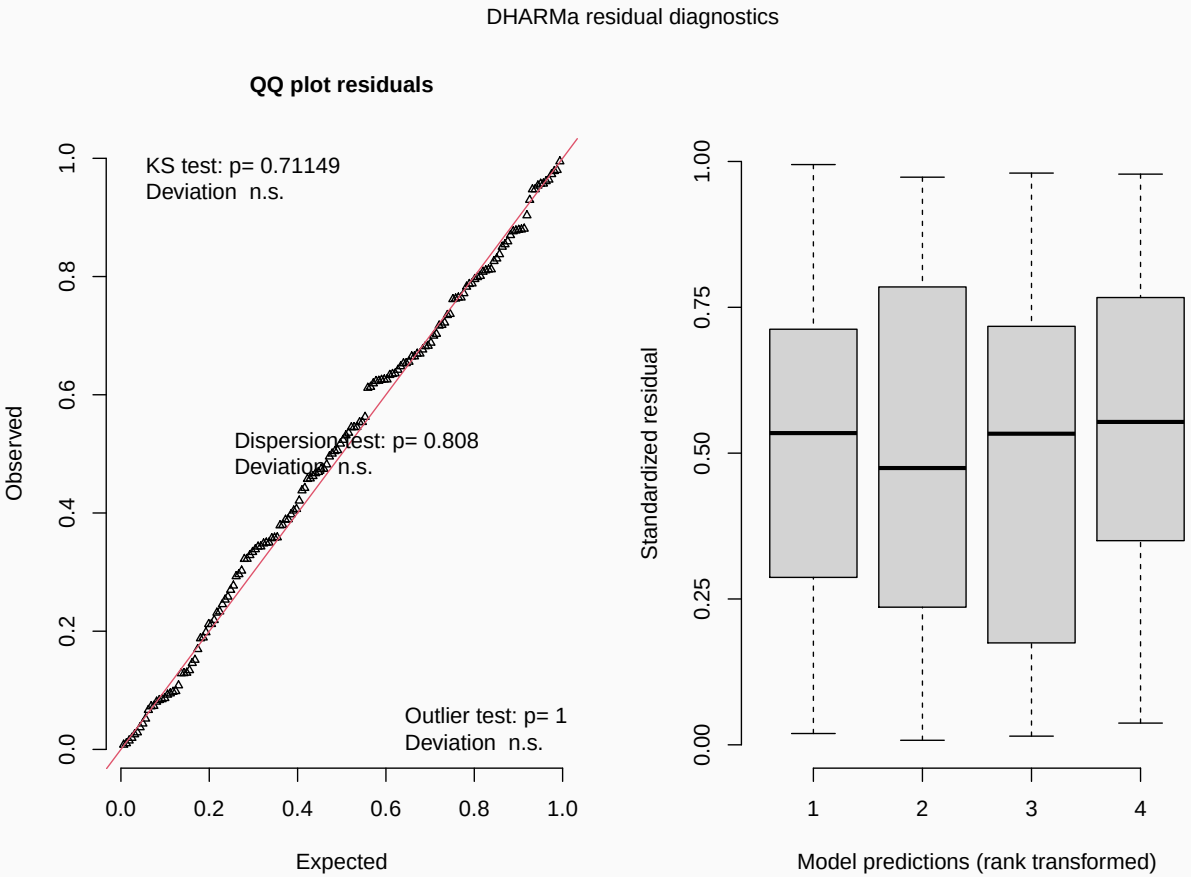
```
plot(glm1)
```



Modelos lineales generalizados

- Comprobación de los residuos del modelo: paquete `DHARMa`

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



Modelos lineales generalizados

Pasos para generar GLMs:

- Análisis exploratorio: Visualización de los datos

Modelos lineales generalizados

Pasos para generar GLMs:

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

Modelos lineales generalizados

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`)

Modelos lineales generalizados

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`)

Modelos lineales generalizados

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

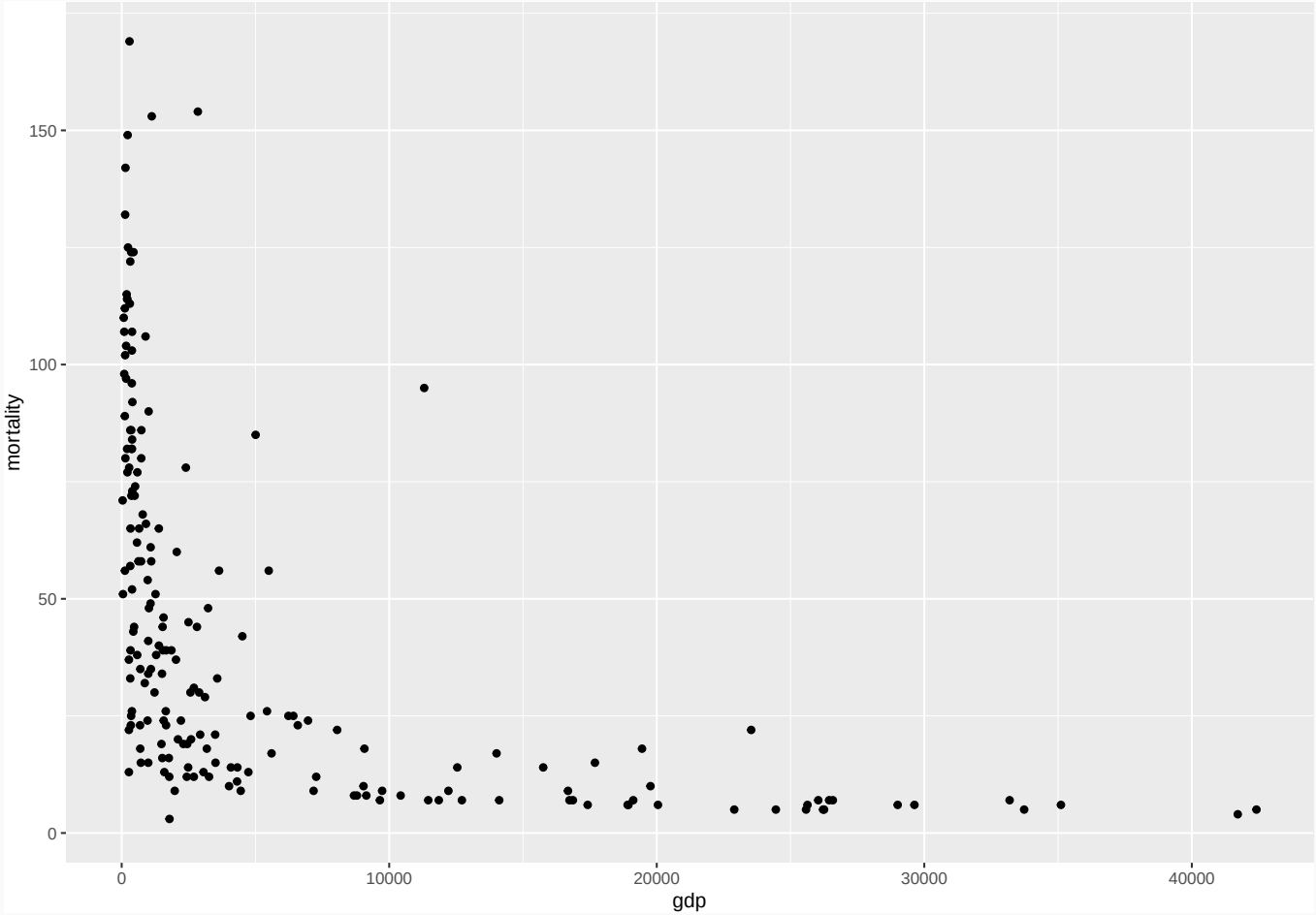
- Otro ejemplo con datos de proporciones

```
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

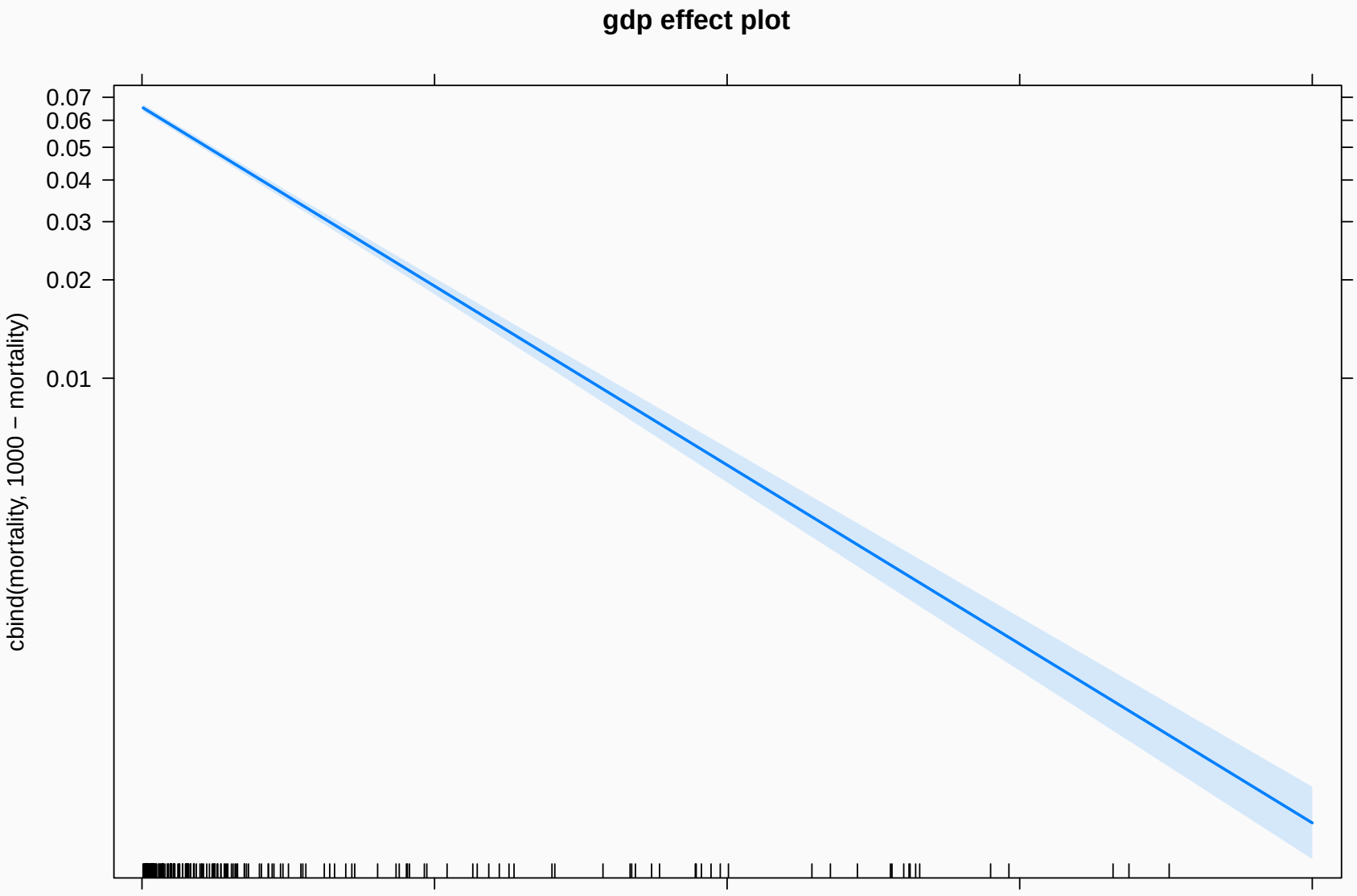
Coeficientes:

```
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
```

Visualización del modelo:

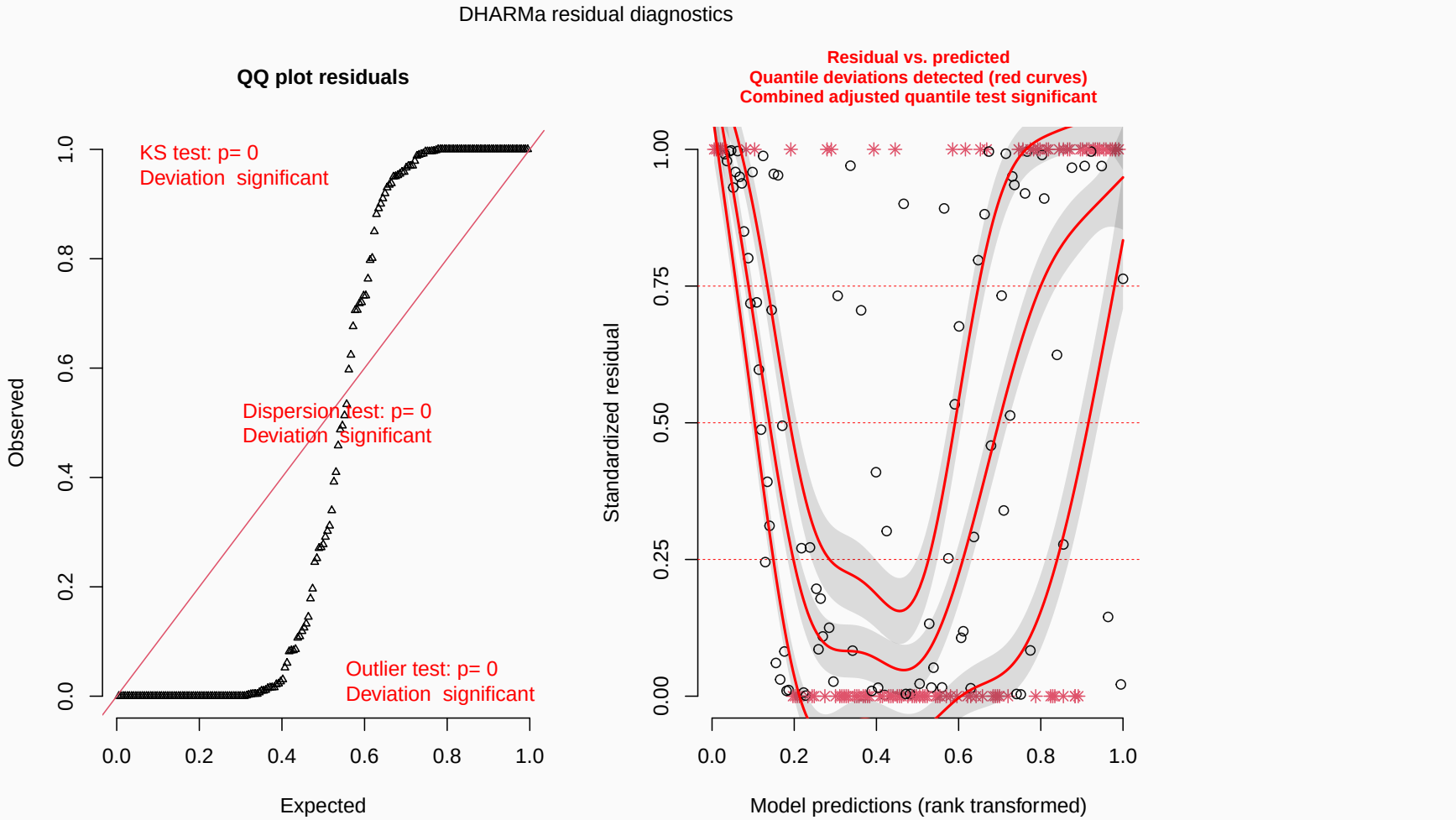
```
plot(allEffects(gdp.glm))
```



Modelos lineales generalizados

Residuos:

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



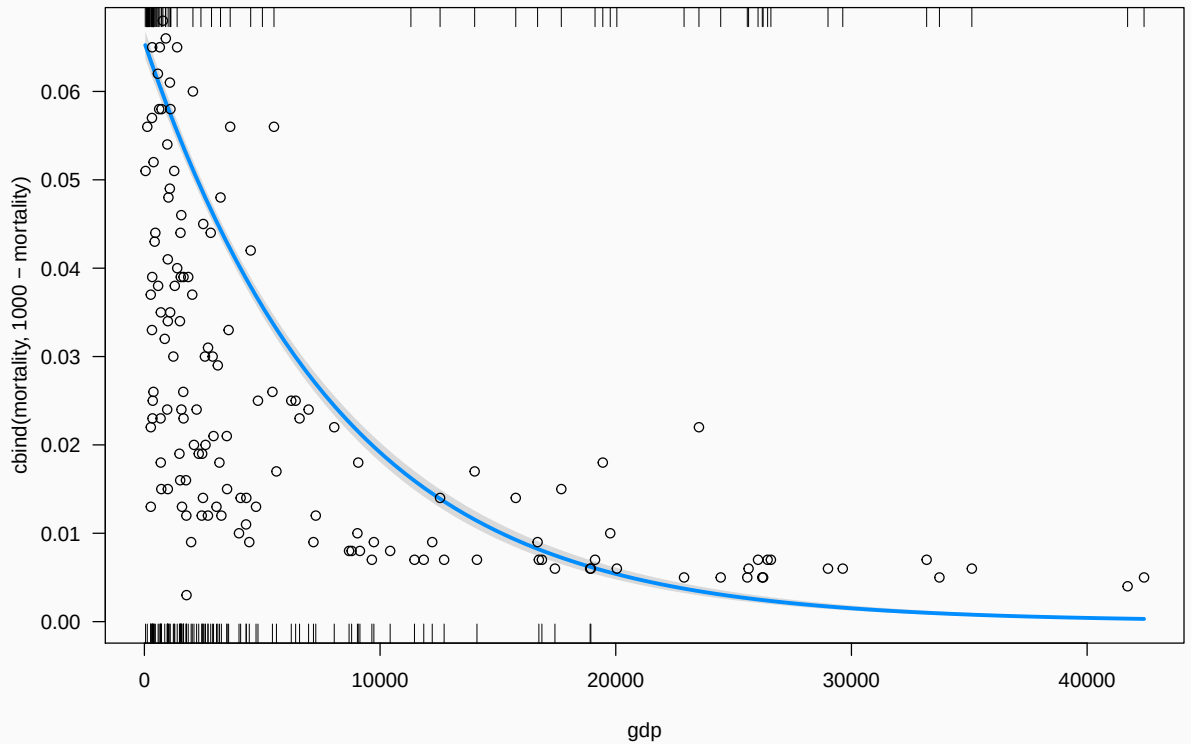
Modelos lineales generalizados

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)
```



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 explicitamente 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
## 0.065217729 0.019143882 0.005402809 0.001509607 0.000420615
```

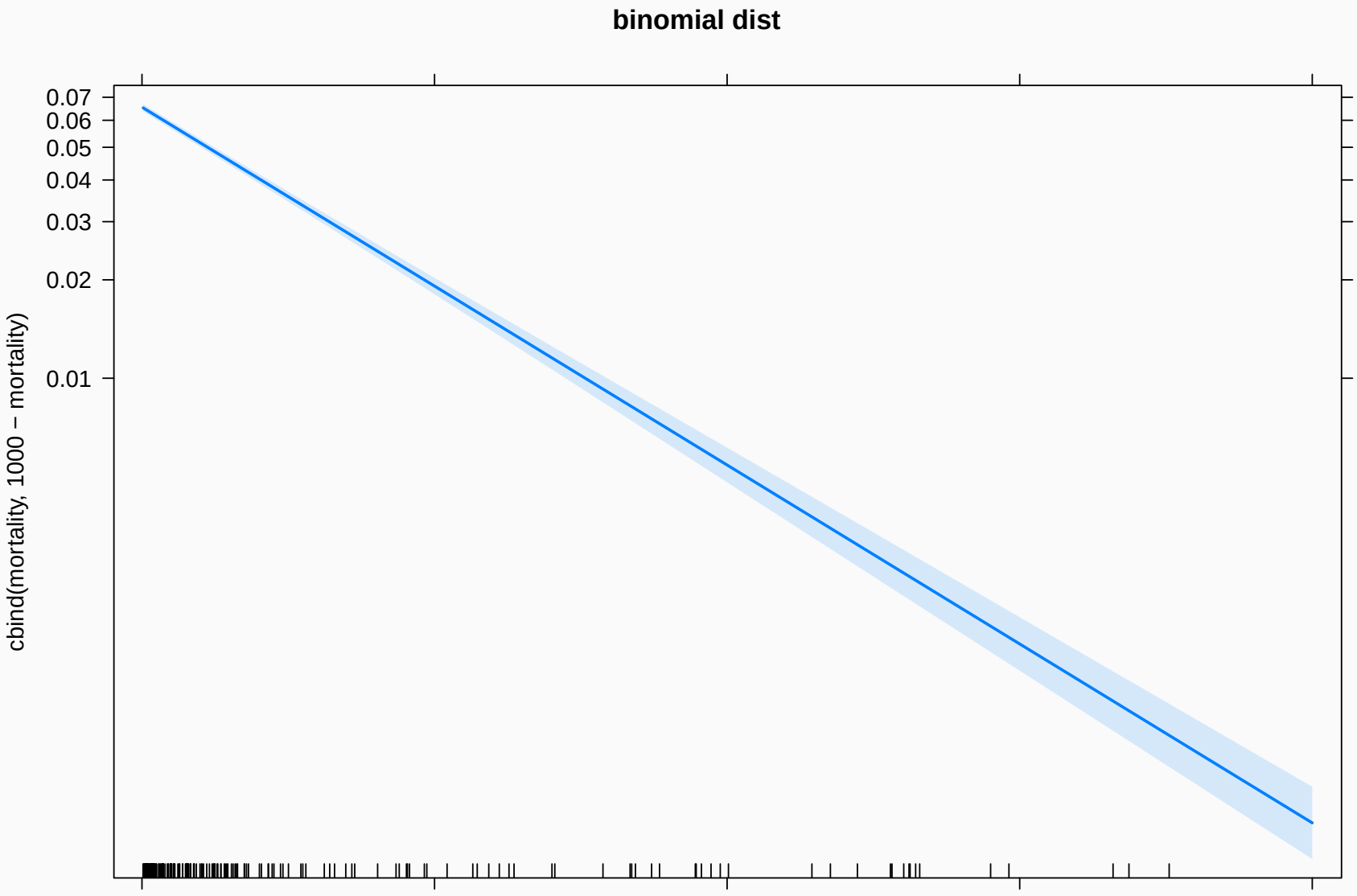
```
allEffects(gdp.glm.qb)
```

```
##  model: cbind(mortality, 1000 - mortality) ~ gdp
##
##  gdp effect
## gdp
##          40       10000       20000       30000       40000
## 0.065217729 0.019143882 0.005402809 0.001509607 0.000420615
```

Modelos lineales generalizados

- Pero los errores asociados sí varían

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



Modelos lineales generalizados

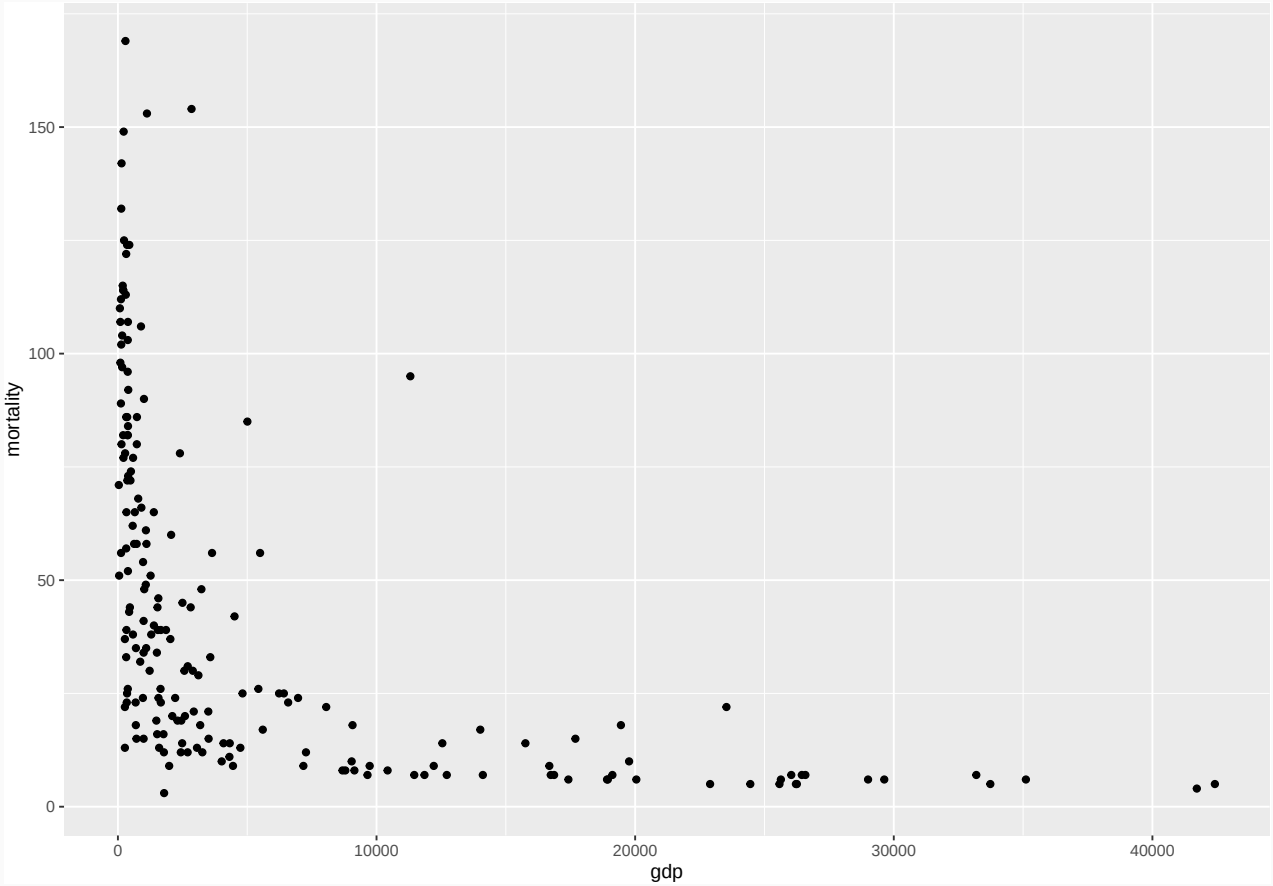
- Pero los errores asociados sí varían

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

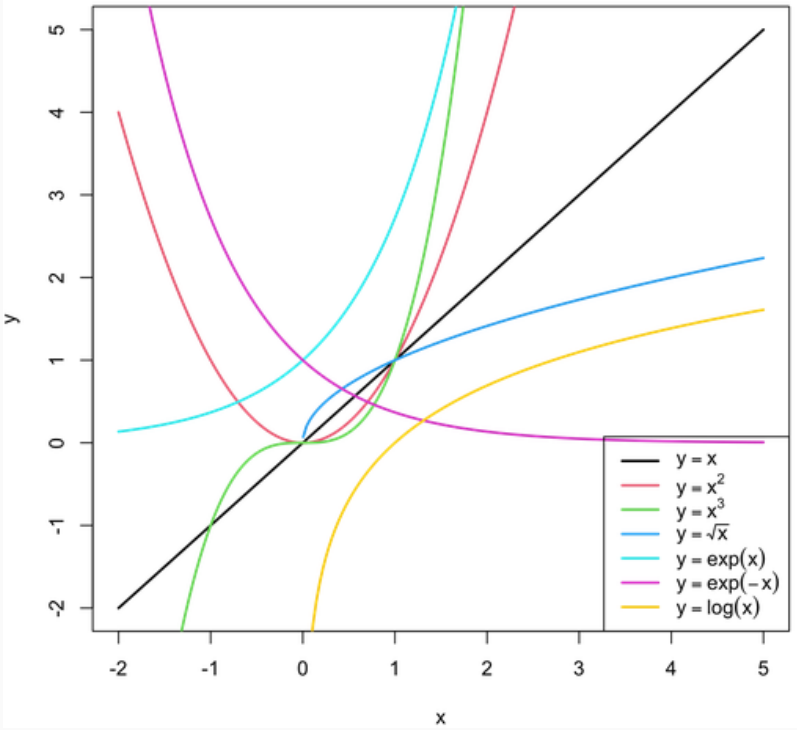
Modelos lineales generalizados

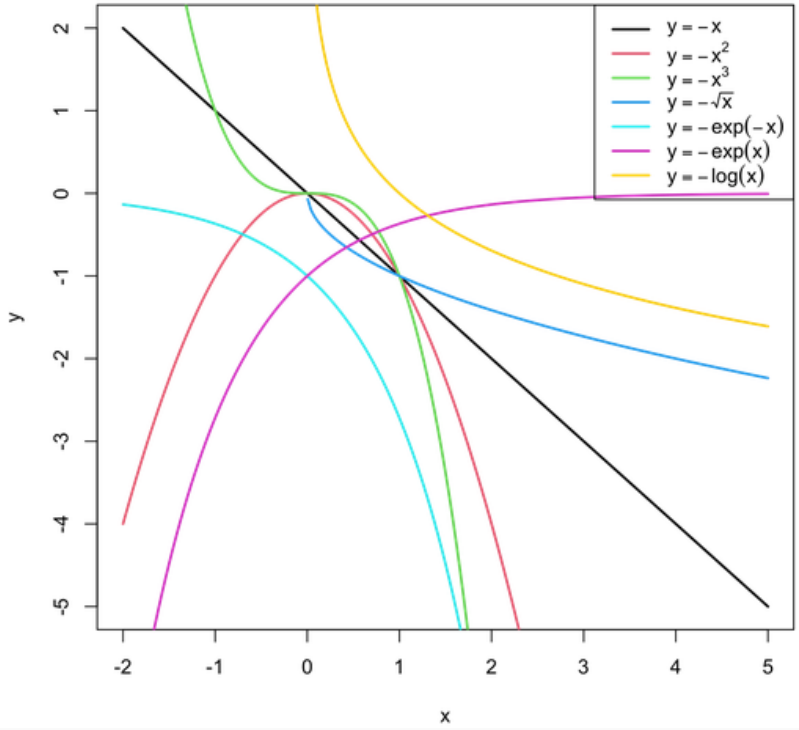
Más allá de la solución concreta, este ejemplo nos ayuda a pensar en la forma de las relaciones entre variables. No todas las relaciones son de naturaleza lineal

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



Modelos lineales generalizados

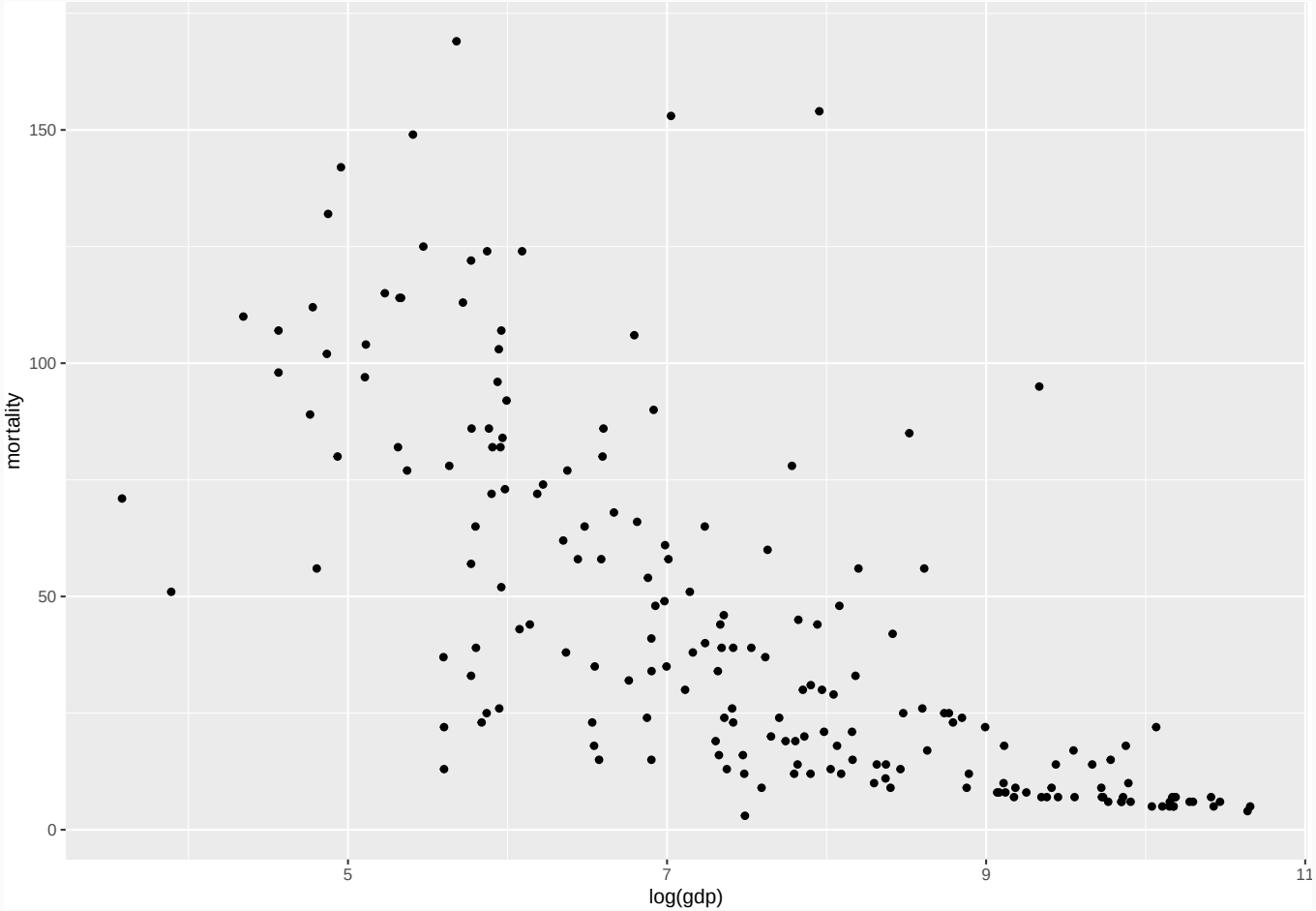




Modelos lineales generalizados

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

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

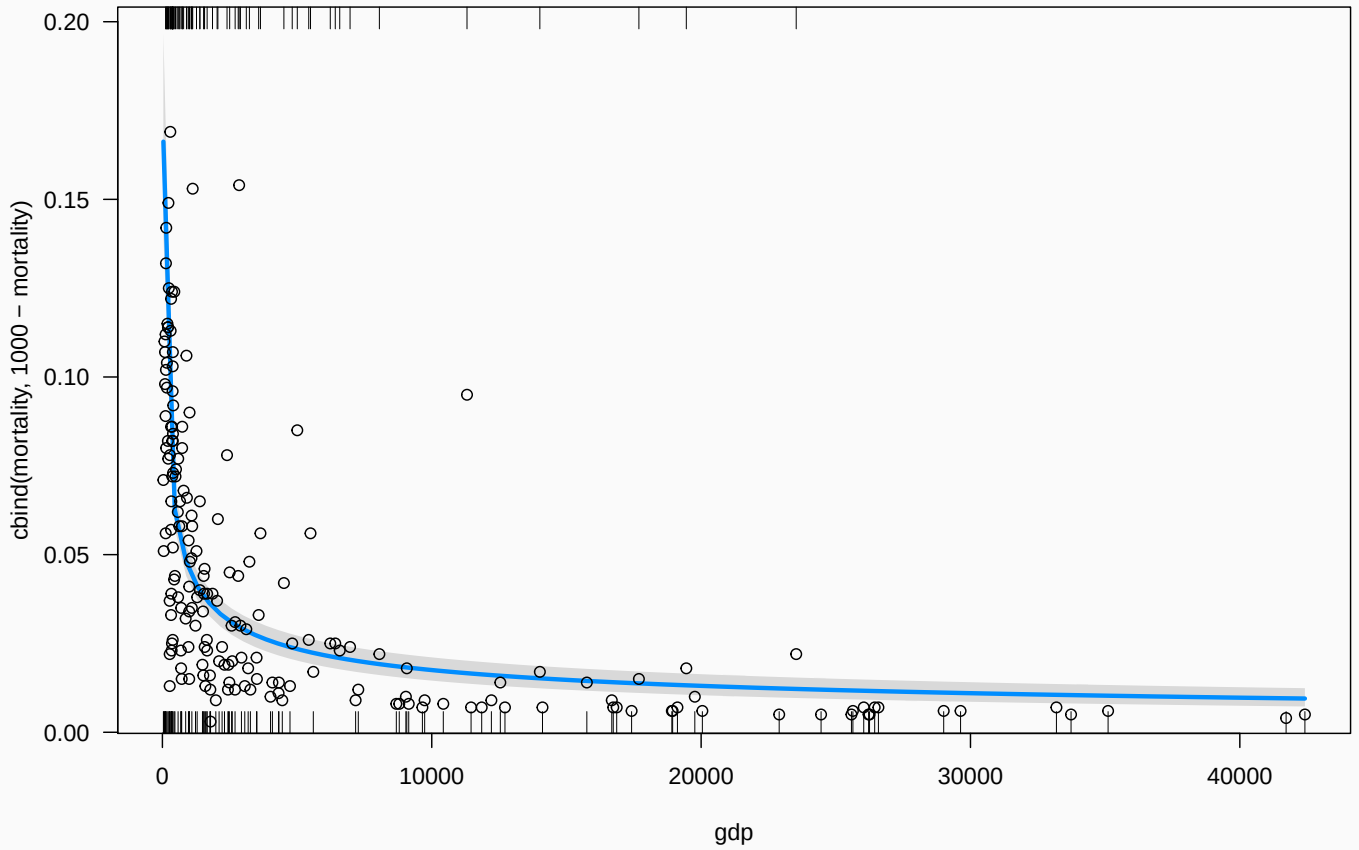


Modelos lineales generalizados

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

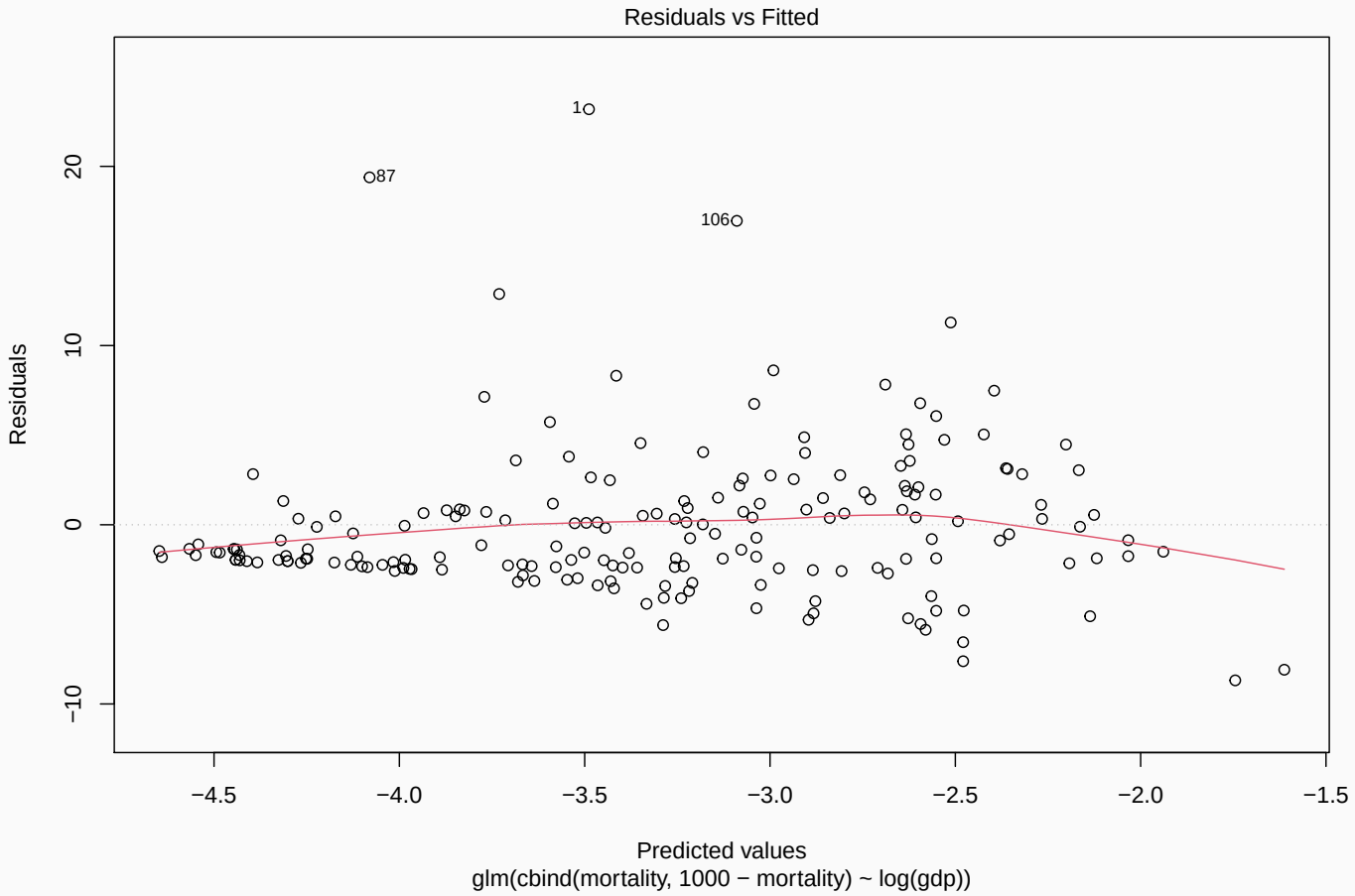
Modelos lineales generalizados

```
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*

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.

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

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
```

Modelos lineales generalizados

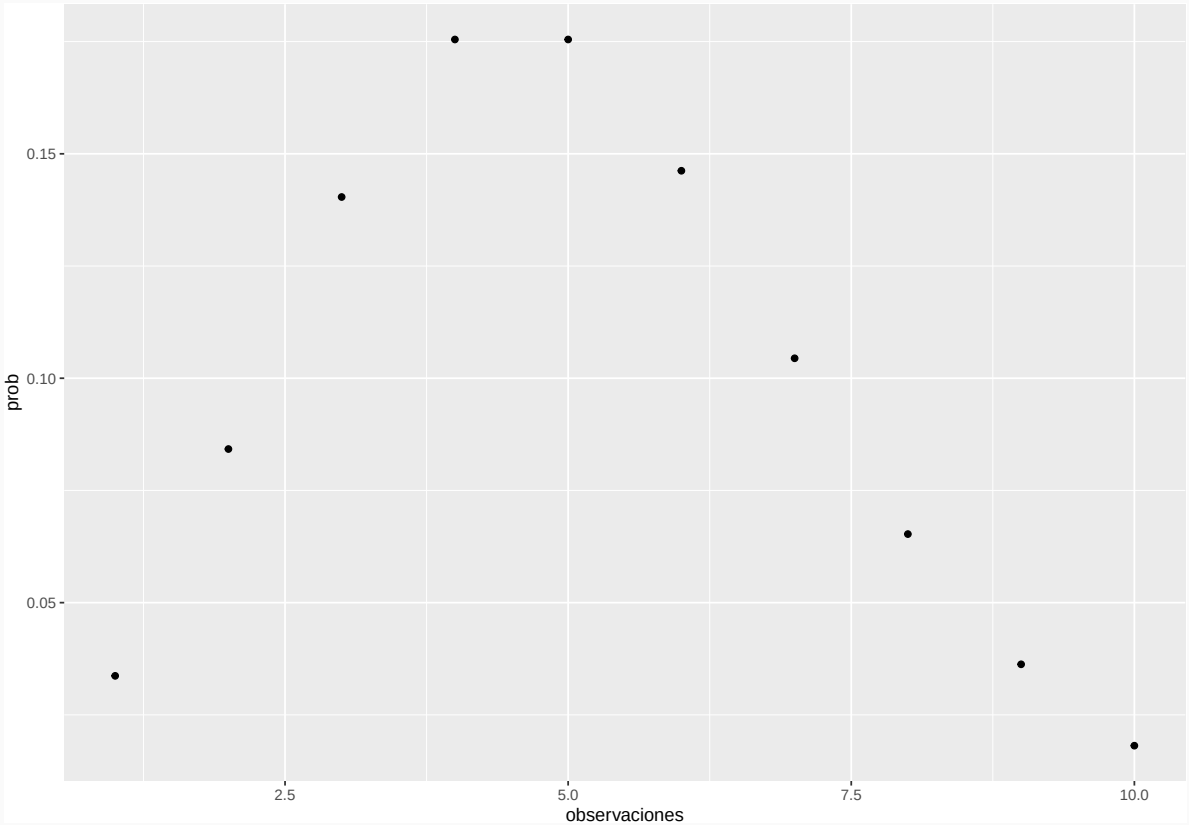
- Los datos de conteos son datos discretos en el intervalo $[0, 1, .., n]$
- Estos datos se modelan con la distribución de **Poisson**: una distribución discreta que expresa la probabilidad de un número de eventos ocurriendo en un intervalo fijo (espacial o temporal), suponiendo que estos eventos ocurren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

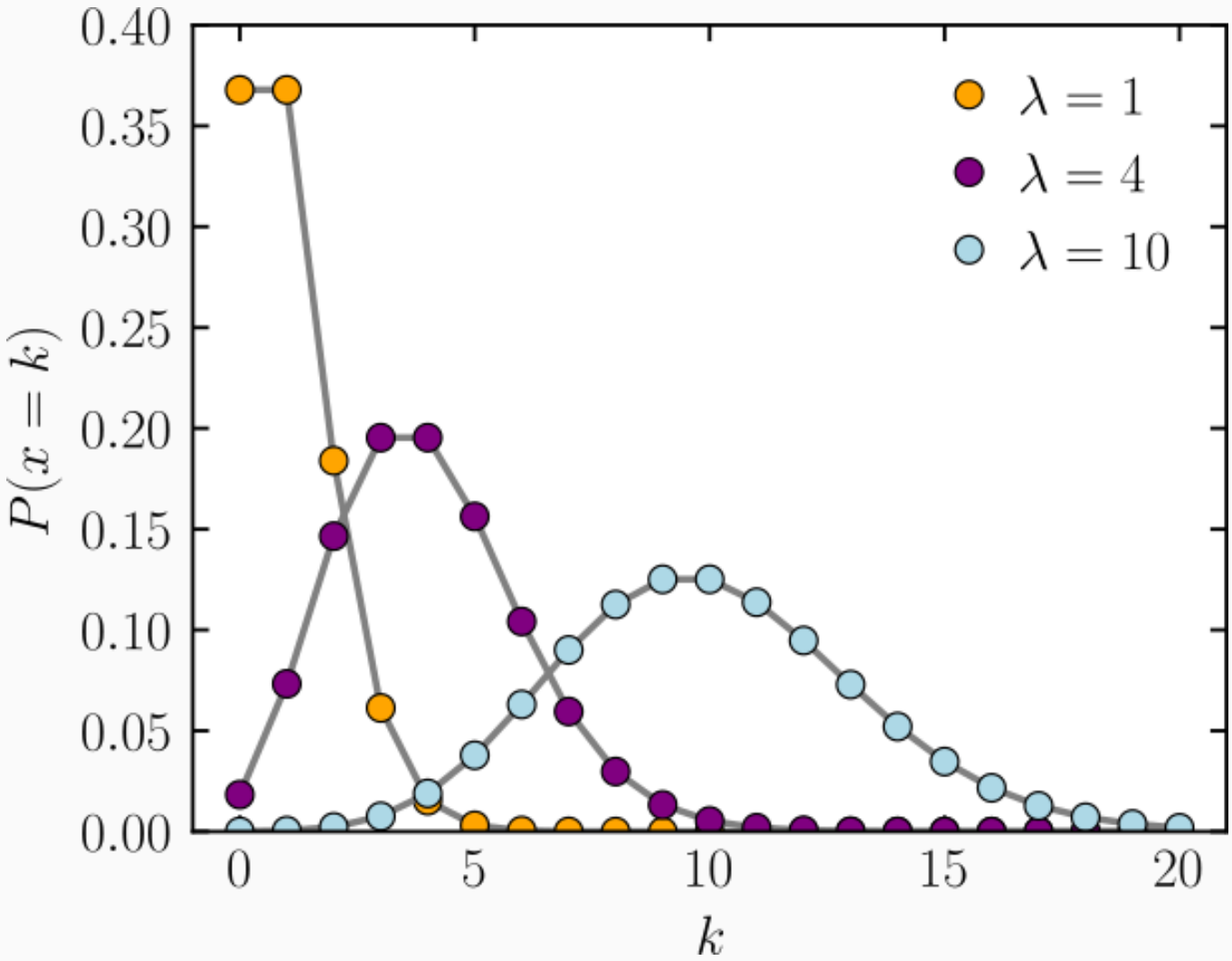
- Podemos asumir que la tasa media de observaciones por hora es constante, porque nuestro observador siempre va a la misma zona y en la misma época del año.
- Nuestro observador, fijándose en sus notas, concluye que, de media, observa 5 aves por hora.
- Podemos preguntarnos cuál es la probabilidad de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



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*!

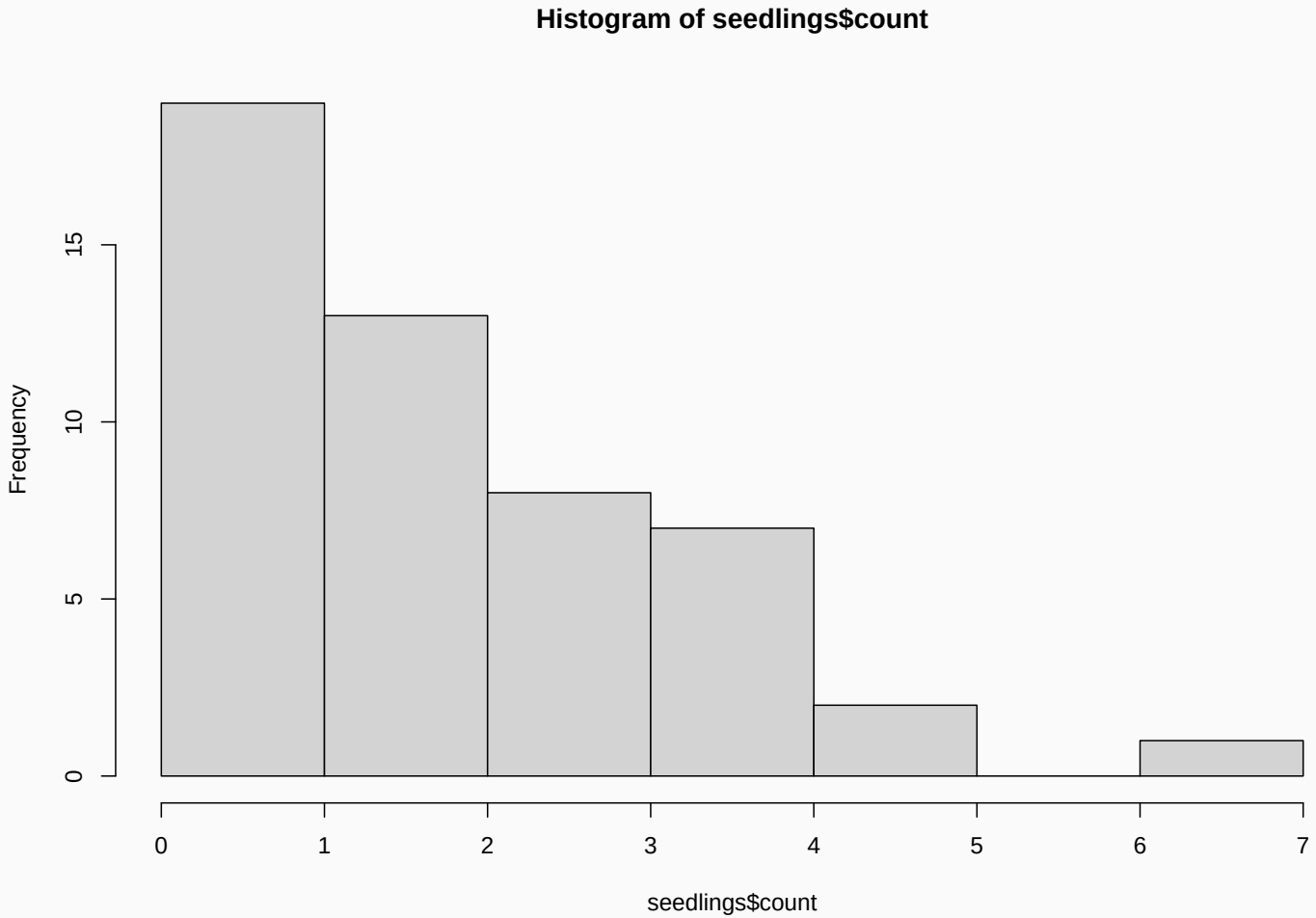
Modelos lineales generalizados

- 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)
```



Modelos lineales generalizados

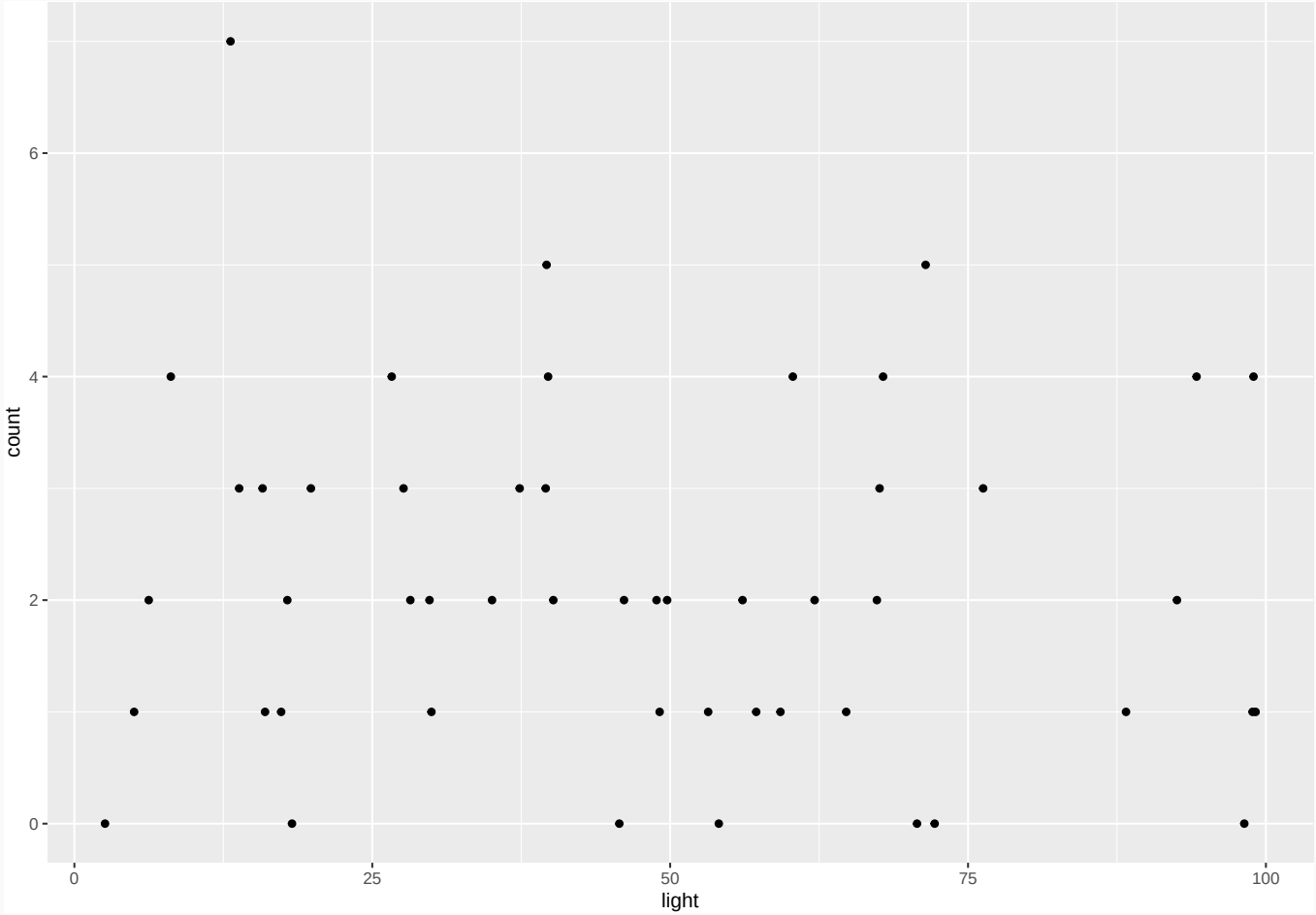
```
head(seedlings)
```

```
##   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(seedl.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(seedl.glm)
```

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

- Les aplicamos el inverso del logaritmo: la exponencial

```
exp(coef(seedl.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.

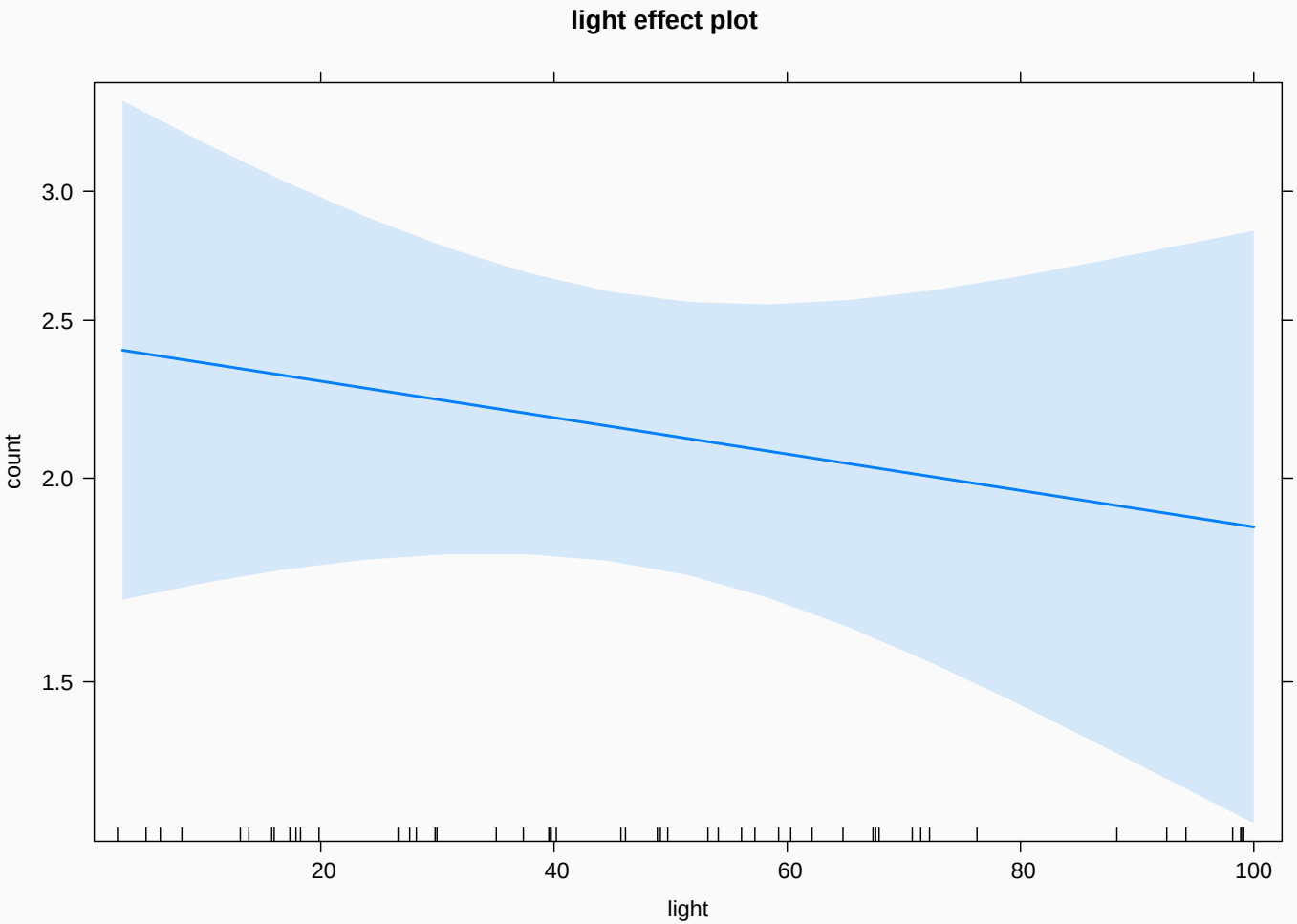
Modelos lineales generalizados

```
summary(allEffects(seedl.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
```

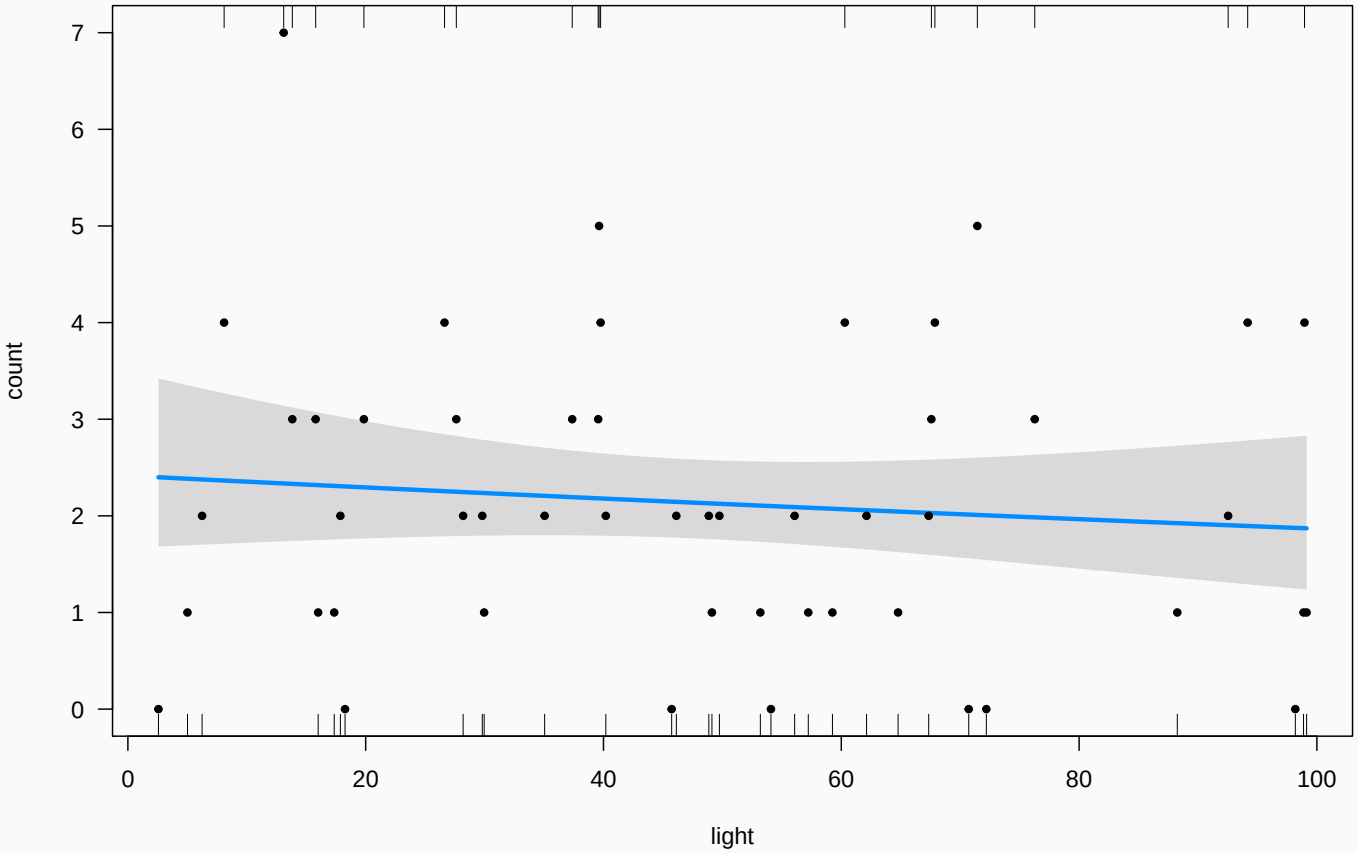
Modelos lineales generalizados

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



Modelos lineales generalizados

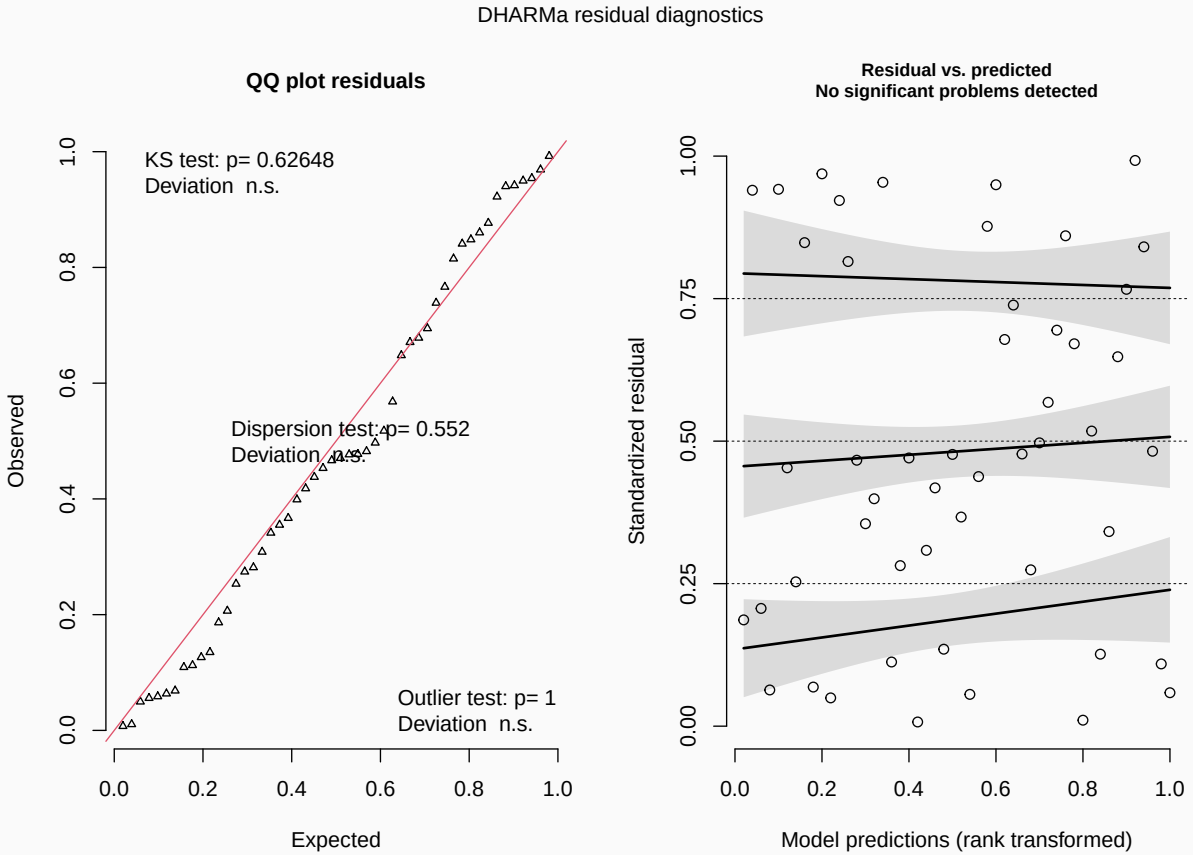
```
visreg(seedl.glm, scale = "response", ylim = c(0, 7))
points(count ~ light, data = seedlings, pch = 20)
```



Modelos lineales generalizados

- Como siempre... comprobación de residuos

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



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`)

```
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 areas? Por definición, en plots de áreas más grandes (e.g. $1m^2$) será más probable observar seedlings que en plots más pequeños (e.g. $0.25m^2$). Queremos modelar *seedlings observados por unidad de área*.

Modelos lineales generalizados

- El factor tamaño se tiene en cuenta con el argumento `offset`

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

- ¿Porqué `log(area)`?

En vez de tener

$$log(\mu_i) = a + b \cdot x_i$$

ahora tenemos

$$log(\frac{\mu_i}{a_i}) = a + b \cdot x_i$$

lo que, reordenando, nos deja

$$log(\mu_i) = log(a_i) + a + b \cdot x_i$$

Los coeficientes ahora vienen referidos *por unidad de área*

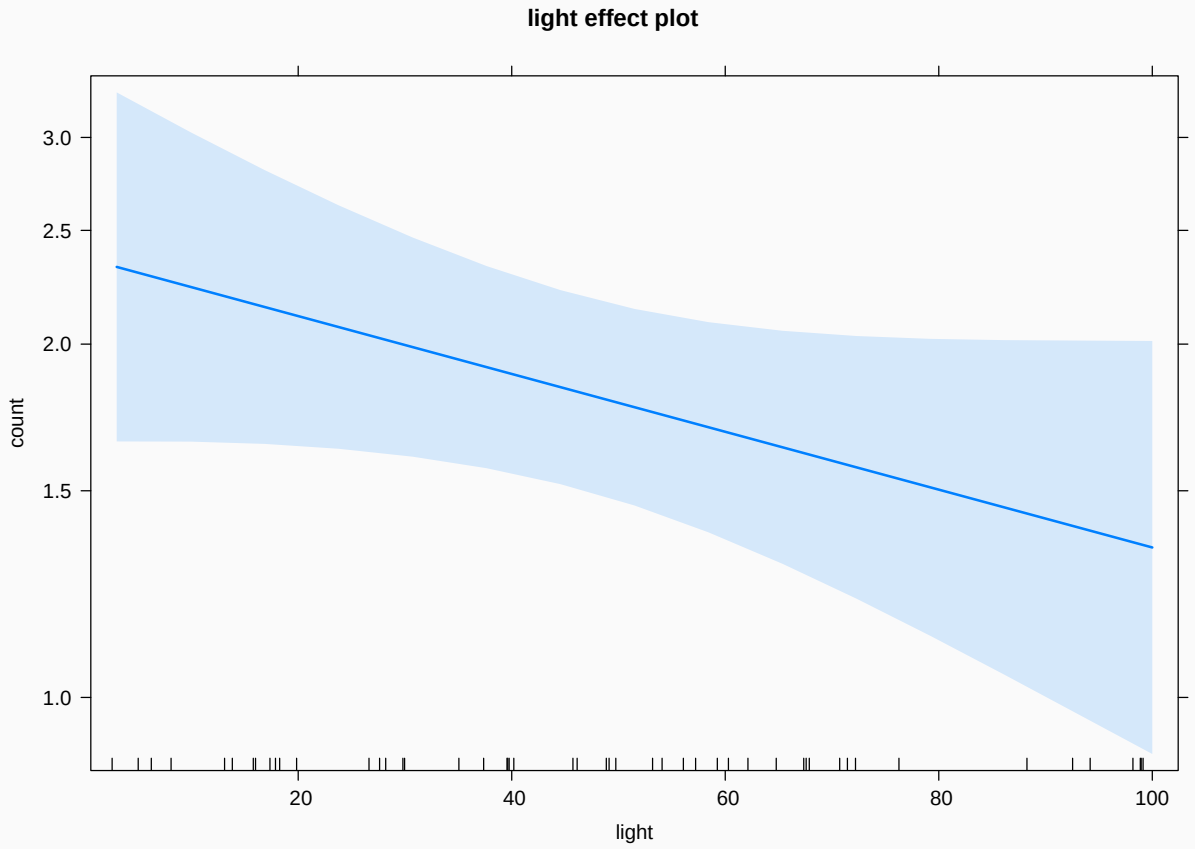
```
exp(coef(seedl.offset))
```

```
## (Intercept)       light
##   4.5411732   0.9943416
```

Modelos lineales generalizados

- Las figuras de `allEffects` se generan sin transformar los coeficientes de vuelta, cuidado

```
plot(allEffects(seedl.offset))
```



Modelos lineales generalizados

- Otro ejemplo: número de casos de cáncer en cuatro ciudades danesas por grupos de edad, durante cuatro años. ¿Varía el número de casos observados por grupos de edad?. 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)
```

```
##         city   age  pop cases
## 1 Fredericia 40-54 3059    11
## 2    Horsens 40-54 2879    13
## 3    Kolding 40-54 3142     4
## 4      Vejle 40-54 2520     5
## 5 Fredericia 55-59  800    11
## 6    Horsens 55-59 1083     6
```

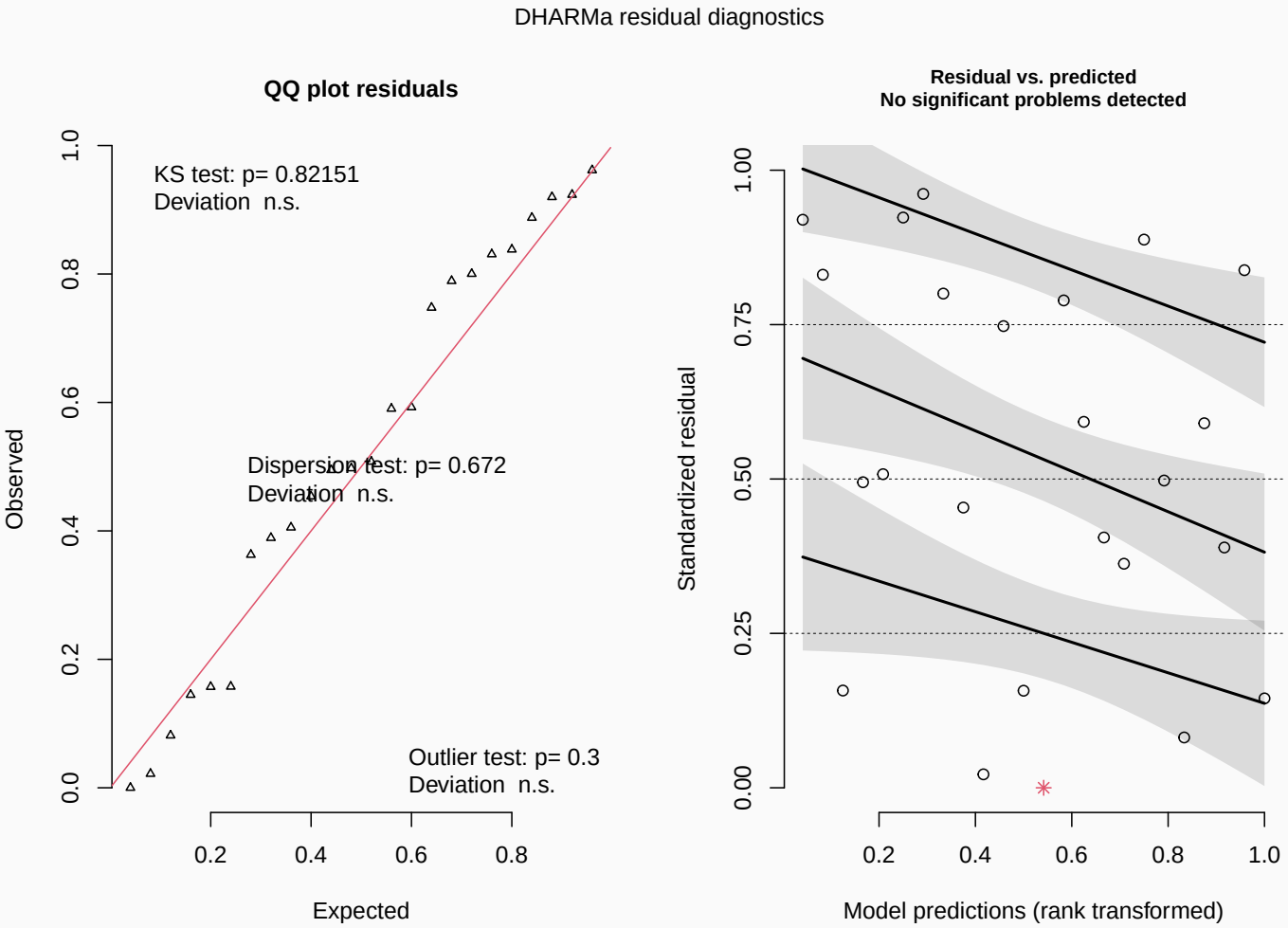
- Recordamos usar `family = poisson` y dar el logaritmo de la población como offset. En este caso, lo modificamos ligeramente para obtener casos *por año y por 10000 habitantes*. Recordad que los datos son de un periodo de cuatro años, por lo que

$$\frac{\mu_i * 10000/4}{pop_i} = \frac{\mu_i}{pop_i/2500}$$

```
canc.glm <- glm(cases ~ age,
                offset = log(pop/2500),
                data = cancer.data,
                family = poisson)
```

Modelos lineales generalizados

```
DHARMa::simulateResiduals(canc.glm, plot = TRUE)
```



- La variable respuesta es categórica. Recordad la interpretación

```
exp(coef(canc.glm))
```

```
## (Intercept)    age55-59    age60-64    age65-69    age70-74      age75+
##    7.112069    2.951584    4.489204    5.756252    6.342177    4.088919
```

El `intercept` (a) nos dice el número de casos por año y 10000 habitantes en la categoría de referencia (40-54 años).

Los coeficientes de los grupos de edad superiores van referidos a su **variación** con respecto al grupo de referencia. Por ejemplo, personas de 60-64 años tienen, en media, 4.4892045 casos más de cáncer por año y 10000 habitantes que el grupo de 40-54 años.

¿Son estadísticamente significativas estas diferencias?

```
summary(canc.glm)
```

```
##
## Call:
## glm(formula = cases ~ age, family = poisson, data = cancer.data,
##     offset = log(pop/2500))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8520  -0.6424  -0.1067   0.7853   1.5468
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.9618     0.1741  11.270  < 2e-16 ***
## age55-59      1.0823     0.2481   4.363 1.29e-05 ***
## age60-64      1.5017     0.2314   6.489 8.66e-11 ***
## age65-69      1.7503     0.2292   7.637 2.22e-14 ***
## age70-74      1.8472     0.2352   7.855 4.00e-15 ***
## age75+        1.4083     0.2501   5.630 1.80e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

¿Son estadísticamente significativas estas diferencias?

```
exp(confint(canc.glm))
```

```
##                  2.5 %    97.5 %
## (Intercept) 4.953216  9.821960
## age55-59    1.809509  4.808601
## age60-64    2.859539  7.111934
## age65-69    3.685433  9.084775
## age70-74    4.005487 10.110915
## age75+      2.495075  6.684026
```

Otras consideraciones

- ¿R^2 para GLM? No es tan sencillo como para modelos lineales. Existen aproximaciones, pero en general, no son fácilmente interpretables como un buen o mal ajuste del modelo.

Otras consideraciones

- ¿R^2 para GLM? No es tan sencillo como para modelos lineales. Existen aproximaciones, pero en general, no son fácilmente interpretables como un buen o mal ajuste del modelo.
- ¿otras distribuciones? Gamma (datos continuos, positivos, y asimétricos), Beta (continua, acotada entre 0 y 1)...

Otras consideraciones

- ¿R^2 para GLM? No es tan sencillo como para modelos lineales. Existen aproximaciones, pero en general, no son fácilmente interpretables como un buen o mal ajuste del modelo.
- ¿otras distribuciones? Gamma (datos continuos, positivos, y asimétricos), Beta (continua, acotada entre 0 y 1)...
- El ajuste en GLMs no se calcula por el método de mínimos cuadrados. Se calcula por el método de máxima verosimilitud (maximum likelihood). No lo estudiaremos, pero si necesitáis profundizar en el ajuste de modelos complejos, es el concepto más importante que hay que entender. Un buen sitio para empezar es el capítulo 20 de Whitlock & Schluter.

Resumen

- Los modelos lineales generalizados (GLMs) nos permiten modelar datos que no siguen una distribución normal

Resumen

- Los modelos lineales generalizados (GLMs) nos permiten modelar datos que no siguen una distribución normal
- Las distribuciones más comunes son la binomial y la poisson, ambas discretas

Resumen

- Los modelos lineales generalizados (GLMs) nos permiten modelar datos que no siguen una distribución normal
- Las distribuciones más comunes son la binomial y la poisson, ambas discretas
- Los GLM necesitan además una función de enlace, para transformar las estimaciones a la distribución elegida

Resumen

- Los modelos lineales generalizados (GLMs) nos permiten modelar datos que no siguen una distribución normal
- Las distribuciones más comunes son la binomial y la poisson, ambas discretas
- Los GLM necesitan además una función de enlace, para transformar las estimaciones a la distribución elegida
- Para interpretar los coeficientes de un GLM, por tanto, hay que deshacer la función de enlace

Resumen

- Los modelos lineales generalizados (GLMs) nos permiten modelar datos que no siguen una distribución normal
- Las distribuciones más comunes son la binomial y la poisson, ambas discretas
- Los GLM necesitan además una función de enlace, para transformar las estimaciones a la distribución elegida
- Para interpretar los coeficientes de un GLM, por tanto, hay que deshacer la función de enlace
- El proceso de ajuste de un GLM es similar al de un modelo lineal: Visualización, ajuste, comprobación de residuos, interpretación

Resumen

- Los modelos lineales generalizados (GLMs) nos permiten modelar datos que no siguen una distribución normal
- Las distribuciones más comunes son la binomial y la poisson, ambas discretas
- Los GLM necesitan además una función de enlace, para transformar las estimaciones a la distribución elegida
- Para interpretar los coeficientes de un GLM, por tanto, hay que deshacer la función de enlace
- El proceso de ajuste de un GLM es similar al de un modelo lineal: Visualización, ajuste, comprobación de residuos, interpretación
- Los GLM aceptan todo tipo de combinaciones de predictores: categóricos, numéricos, o interacciones entre ellos.

Recetario de R

- función para ajustar GLM: `glm(formula, data = datos, family = distribucion)`
- comprobación de residuos: `DHARMa::simulateResiduals(modelo, plot = TRUE)`
- comprobación de sobredispersion (también con DHARMa): `testDispersion(simulateResiduals(modelo))`
- coeficientes: transformar con `plogis` (binomial) o con `exp` (poisson). ¡Cuidado con la interpretación!