Script7

Yosune Miquelajauregui

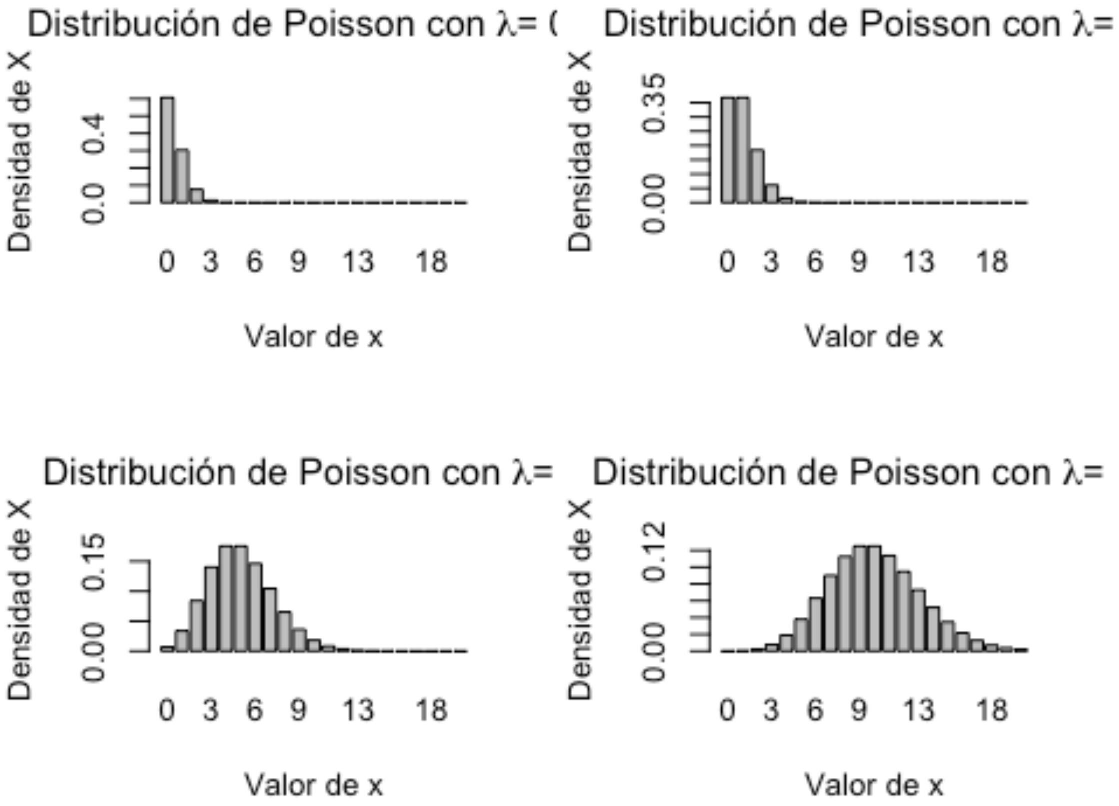
31/12/2017

Modelos lineales generalizados

Regresión Poisson

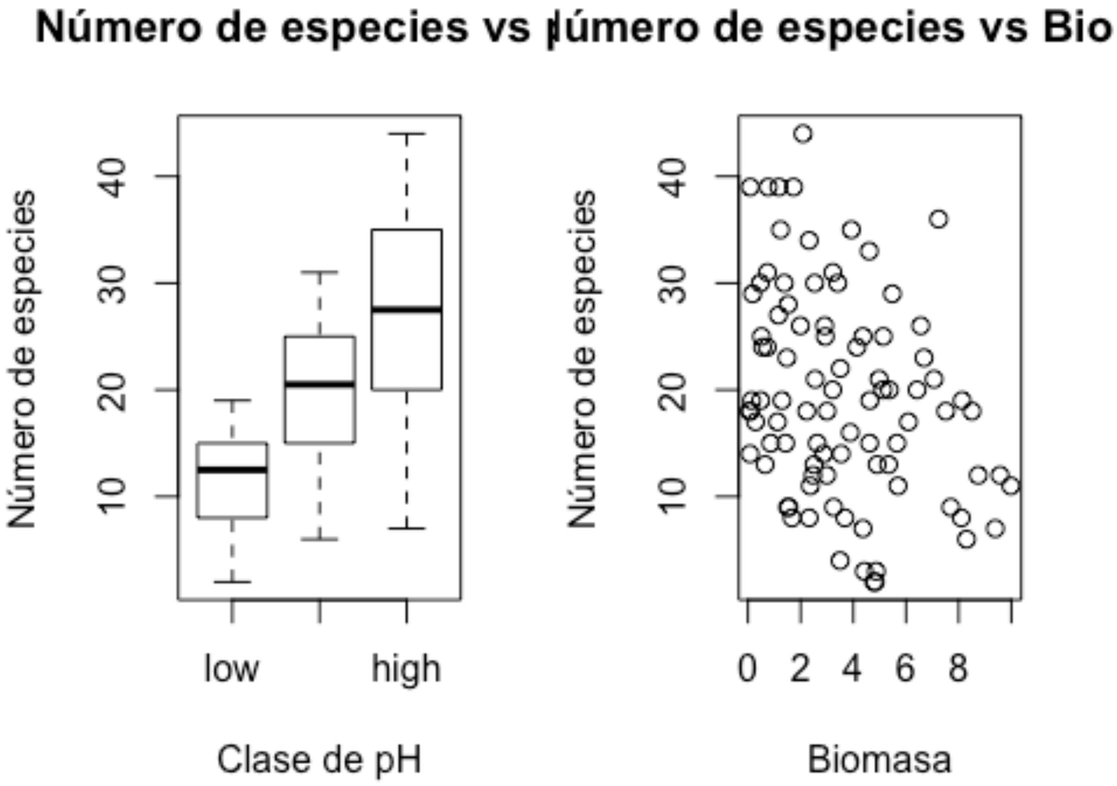
Función de densidad para valores que siguen una distribución Poisson con media dada por lamba

```
x1<-dpois(0:20, lambda=0.5)
x2<-dpois(0:20, lambda=1)
x3<-dpois(0:20, lambda=5)
x4<-dpois(0:20, lambda=10)
par(mfrow=c(2,2))
barplot(x1, main=expression(paste("Distribución de Poisson con ", lambda,
"= 0.5")), names.arg=0:20, ylab="Densidad de X", xlab="Valor de x")
barplot(x2, main=expression(paste("Distribución de Poisson con ", lambda,
"= 1")), names.arg=0:20, ylab="Densidad de X", xlab="Valor de x")
barplot(x3, main=expression(paste("Distribución de Poisson con ", lambda,
"= 5")), names.arg=0:20, ylab="Densidad de X", xlab="Valor de x")
barplot(x4, main=expression(paste("Distribución de Poisson con ", lambda,
"= 10")), names.arg=0:20, ylab="Densidad de X", xlab="Valor de x")
```



Regresión Poisson. Datos del libro Crawley

```
especies<-read.table("Species.txt", header=TRUE)
str(especies)

## 'data.frame':    90 obs. of  3 variables:
##  $ pH     : Factor w/ 3 levels "high","low","mid": 1 1 1 1 1 1 1 1 1 1
...
##  $ Biomasa: num  0.469 1.731 2.09 3.926 4.367 ...
##  $ Especie: int  30 39 44 35 25 29 23 18 19 12 ...

especies$pH <- factor(especies$pH , levels = c("low", "mid", "high"))
spp<-split(especies$Especie, especies$pH)
bio<-split(especies$Biomasa, especies$pH)
par(mfrow=c(1,2))
plot(especies$Especie~especies$pH, ylab="Número de especies", xlab="Clase
de pH", main="Número de especies vs pH")
plot(especies$Especie~especies$Biomasa, ylab="Número de especies",
xlab="Biomasa", main="Número de especies vs Biomasa")
```



Ajustar un modelo de regresión Poisson

```
mod1<-glm(Especie~Biomasa+pH+Biomasa:pH, family=poisson(link = log),
data=especies)
names(mod1)

##  [1] "coefficients"      "residuals"         "fitted.values"
##  [4] "effects"           "R"                 "rank"
##  [7] "qr"                "family"            "linear.predictors"
## [10] "deviance"          "aic"               "null.deviance"
## [13] "iter"              "weights"           "prior.weights"
## [16] "df.residual"       "df.null"           "y"
## [19] "converged"         "boundary"          "model"
## [22] "call"              "formula"           "terms"
## [25] "data"              "offset"            "control"
## [28] "method"            "contrasts"         "xlevels"

summary(mod1)

##
## Call:
## glm(formula = Especie ~ Biomasa + pH + Biomasa:pH, family =
poisson(link = log),
##     data = especies)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4978  -0.7485  -0.0402   0.5575   3.2297
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     2.95255    0.08240  35.833  < 2e-16 ***
## Biomasa        -0.26216    0.03803  -6.893 5.47e-12 ***
## pHmid           0.48411    0.10723   4.515 6.34e-06 ***
## pHhigh          0.81557    0.10284   7.931 2.18e-15 ***
## Biomasa:pHmid   0.12314    0.04270   2.884 0.003927 **
## Biomasa:pHhigh  0.15503    0.04003   3.873 0.000108 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 452.346  on 89  degrees of freedom
## Residual deviance:  83.201  on 84  degrees of freedom
## AIC: 514.39
##
## Number of Fisher Scoring iterations: 4
```

Podemos resumir la información con una tabla análoga de ANOVA que lidie con variables dicretas (número de especies)

```
anova(mod1, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: Especie
##
## Terms added sequentially (first to last)
##
##
##            Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                          89     452.35
## Biomasa     1   44.673        88     407.67 2.328e-11 ***
## pH          2  308.431        86      99.24 < 2.2e-16 ***
## Biomasa:pH  2   16.040        84      83.20 0.0003288 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Análisis de devianza

```
mod2<-glm(Especie~Biomasa+pH, family=poisson, data=especies)
mod3<-glm(Especie~Biomasa, family=poisson, data=especies)
anova(mod2, mod1, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Especie ~ Biomasa + pH
## Model 2: Especie ~ Biomasa + pH + Biomasa:pH
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        86     99.242
## 2        84     83.201  2    16.04 0.0003288 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
LR<--2*(logLik(mod2)[1]-logLik(mod1)[1])
df<-df.residual(mod2)-df.residual(mod1)
pchisq(LR,df,lower.tail=F)
```

```
## [1] 0.0003287607
```

Devianza residual

```
dev.residual1<-2*sum(especies$Especie*log(especies$Especie/mod1$fitted)-
(especies$Especie-mod1$fitted))
mod1$deviance
```

```
## [1] 83.20114
```

```
mod2$deviance
```

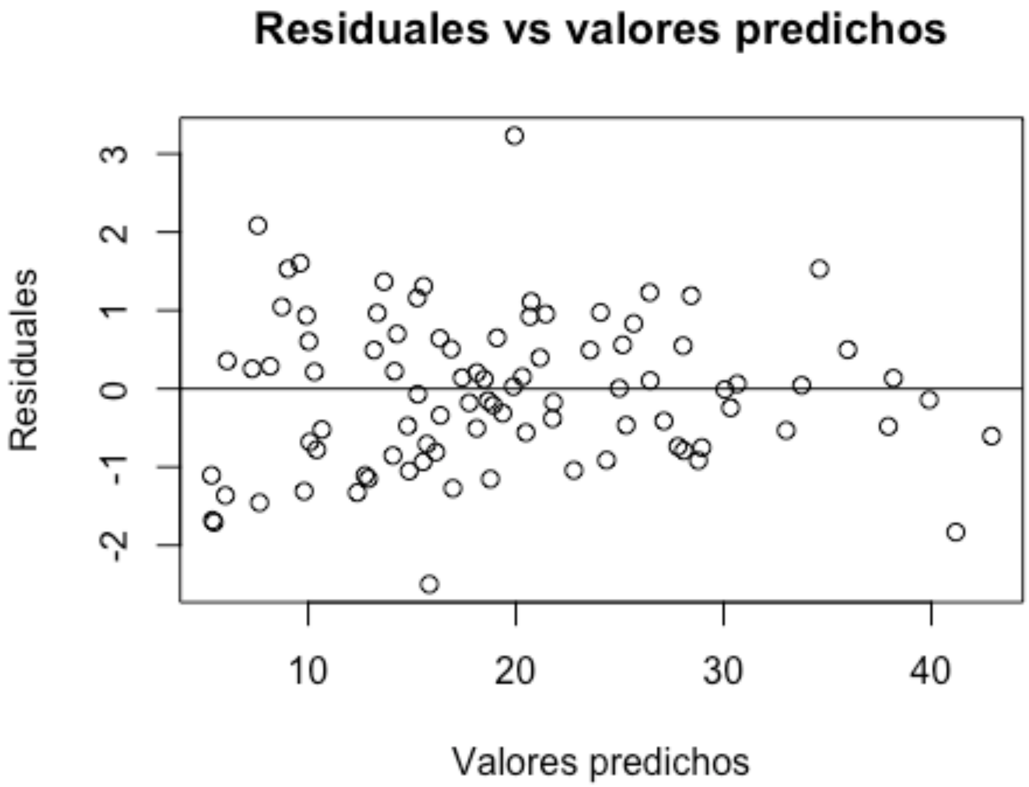
```
## [1] 99.2415
```

```
residual.dev2<-2*sum(especies$Especie*log(especies$Especie/mod3$fitted)-
(especies$Especie-mod3$fitted))
```

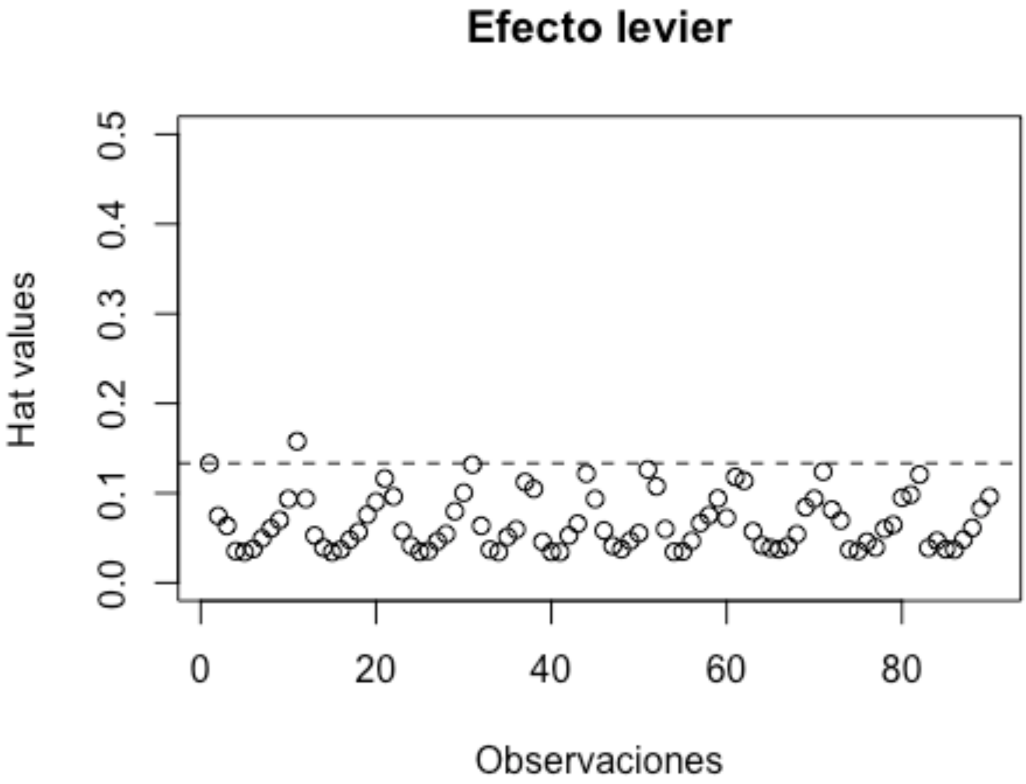
Verificar supuestos

```
residuals(mod1, type="response")
```

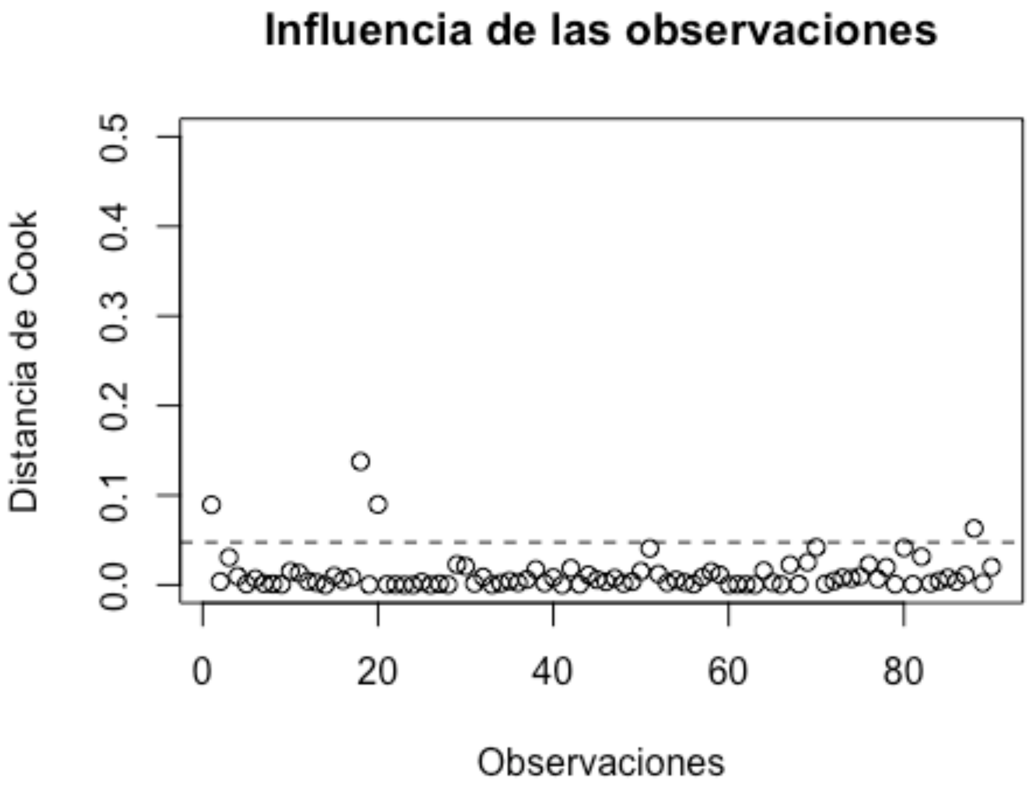
```
##            1            2            3            4            5
## -11.17568365   3.02962044   9.38642086   6.56687473  -2.12104570
##            6            7            8            9           10
##   4.93301929   1.84249672  -1.36373626   0.88170028  -3.52828937
##           11           12           13           14           15
##  -3.89865196  -2.92486021  -3.01184170  -0.05507520   6.56243292
##           16           17           18           19           20
##  -4.36338032   4.55992223  16.06886644   0.58849504  -8.83262593
##           21           22           23           24           25
##  -0.89221934   0.82852195   0.24819521   0.34312577  -3.79958067
##           26           27           28           29           30
##   0.02740658  -1.76150805   0.68833709  -4.96536739  -3.86156982
##           31           32           33           34           35
##  -1.33269497   4.33164743  -0.80219122  -2.48222560  -2.71589053
##           36           37           38           39           40
##  -1.78813331  -1.65674603  -4.78015784   2.43019563   5.26670199
##           41           42           43           44           45
##   0.12417538   5.44552367   0.84631309  -2.08031387   2.95403747 
##           46           47           48           49           50 
##   2.86191891  -4.79058101  -2.11103182   2.66229074   4.74763238 
##           51           52           53           54           55 
##  -3.80389778  -3.95034583  -2.30885776   4.32979131   2.90700288 
##           56           57           58           59           60 
##  -1.35724563  -3.07864101   3.67557501  -4.08305508   0.53920339 
##           61           62           63           64           65 
##  -0.65500122   0.52869799  -0.27435072   5.35185887   1.97340098 
##           66           67           68           69           70 
##   0.70002350   5.38540597   0.83432381  -3.01663552  -3.46970582 
##           71           72           73           74           75 
##  -0.90447189   2.12264452  -3.14241797  -3.74078784  -4.35819106 
##           76           77           78           79           80 
##   4.97367900   3.08204281  -3.65705655   0.69809237  -3.39744137 
##           81           82           83           84           85 
##  -0.75379420  -4.76923995   1.83024026   2.72169583  -3.90003785 
##           86           87           88           89           90 
##  -2.41016343   3.26616805   6.42400054   0.89960748  -2.34258662

plot(residuals(mod1)~fitted(mod1), ylab="Residuales", xlab="Valores
predichos", main="Residuales vs valores predichos")
abline(h=0)
```



```
plot(hatvalues(mod1), ylim=c(0,0.5), ylab="Hat values",
xlab="Observaciones", main="Efecto levier")
abline(h=6*2/90, lty=2) # hat = k/n, donde k es el número de parámetros,
n es el tamaño muestral
```



```
plot(cooks.distance(mod1), ylab="Distancia de Cook",
xlab="Observaciones", ylim=c(0,0.5), main="Influencia de las
observaciones")
abline(h=4/(90-6), lty=2)
```



```
chisq<-sum(((especies$Especie-mod1$fitted)^2)/mod1$fitted)
c_hat<-chisq/mod1$df.residual
```

Interpretar output del mod1

```
##Correr an´álisis separado para cada nivel de pH
low<-especies[especies$pH=="low", ]
mid<-especies[especies$pH=="mid", ]
high<-especies[especies$pH=="high", ]
mod_low<-glm(Especie~Biomasa, family=poisson, data=low)
mod_mid<-glm(Especie~Biomasa, family=poisson, data=mid)
mod_high<-glm(Especie~Biomasa, family=poisson, data=high)
##comparar output de los tres modelos
mod_low

##
## Call:  glm(formula = Especie ~ Biomasa, family = poisson, data = low)
##
## Coefficients:
## (Intercept)      Biomasa
##      2.9526      -0.2622
##
## Degrees of Freedom: 29 Total (i.e. Null);  28 Residual
## Null Deviance:       83.77 
## Residual Deviance: 32.72     AIC: 161.1
```

```
mod_mid
```

```
## 
## Call:  glm(formula = Especie ~ Biomasa, family = poisson, data = mid)
## 
## Coefficients:
## (Intercept)      Biomasa  
##       3.437       -0.139  
## 
## Degrees of Freedom: 29 Total (i.e. Null);  28 Residual
## Null Deviance:       71.43 
## Residual Deviance: 16.54     AIC: 163.8
```

```
mod_high
```

```
## 
## Call:  glm(formula = Especie ~ Biomasa, family = poisson, data = high)
## 
## Coefficients:
## (Intercept)      Biomasa  
##      3.7681      -0.1071  
## 
## Degrees of Freedom: 29 Total (i.e. Null);  28 Residual
## Null Deviance:       109.9 
## Residual Deviance: 33.94     AIC: 189.5
```

```
mod1
```

```
## 
## Call:  glm(formula = Especie ~ Biomasa + pH + Biomasa:pH, family = 
poisson(link = log), 
##     data = especies)
## 
## Coefficients:
##    (Intercept)         Biomasa            pHmid           pHhigh  
##         2.9526         -0.2622           0.4841           0.8156  
##  Biomasa:pHmid  Biomasa:pHhigh  
##         0.1231          0.1550  
## 
## Degrees of Freedom: 89 Total (i.e. Null);  84 Residual
## Null Deviance:       452.3 
## Residual Deviance: 83.2  AIC: 514.4
```

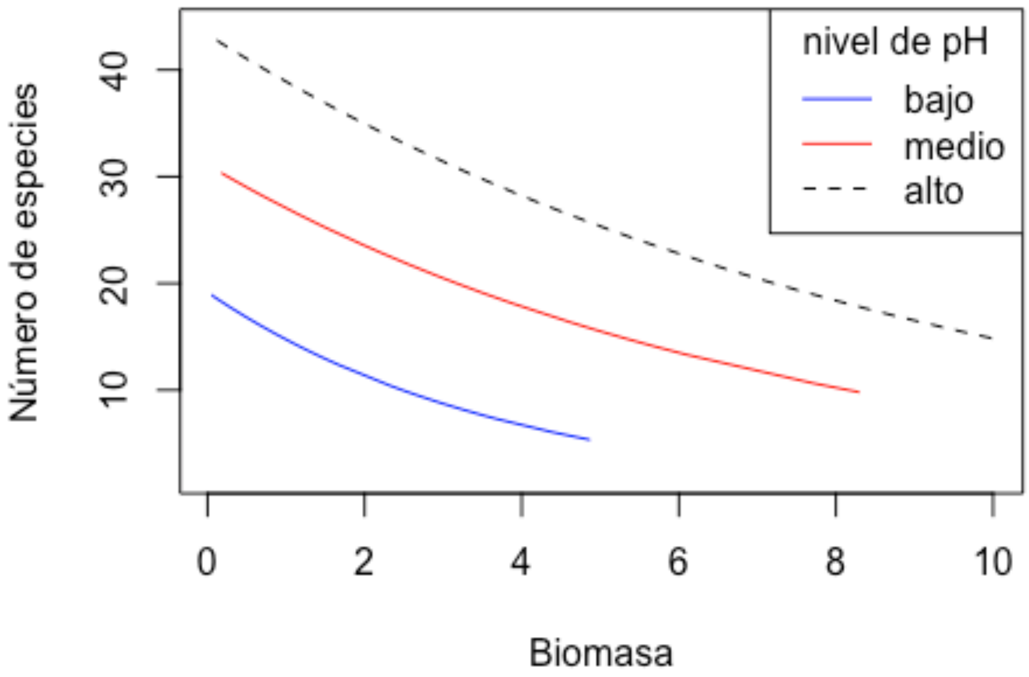
Crear hoja dedatos con los valores predichos para cada nivel de pH

```
fit_low<-data.frame(fit=fitted(mod_low), low)
fit_low<-fit_low[order(fit_low$fit), ]
fit_mid<-data.frame(fit=fitted(mod_mid), mid)
fit_mid<-fit_mid[order(fit_mid$fit), ]
```

```
fit_high<-data.frame(fit=fitted(mod_high), high)
fit_high<-fit_high[order(fit_high$fit), ]
```

Graficar

```
plot(1~1, ylab="Número de especies", xlab="Biomasa", type="n",
ylim=range(especies$Especie), xlim=range(especies$Biomasa))

## Warning in plot.formula(1 ~ 1, ylab = "Número de especies", xlab =
## "Biomasa", : the formula '1 ~ 1' is treated as '1 ~ 1'

lines(fit_low$fit~fit_low$Biomasa, lty="solid", col="blue")
lines(fit_mid$fit~fit_mid$Biomasa, lty="solid", col="red")
lines(fit_high$fit~fit_high$Biomasa, lty="dashed", col="black")
legend(x="topright", lty=c("solid", "solid", "dashed"), col=c("blue",
"red", "black"), legend=c("bajo", "medio", "alto"), title="nivel de pH")
```



Precitor lineal y valores predichos

```
especies [1,]     #primera observación

##      pH   Biomasa Especie
## 1 high 0.4692972      30

mod1$linear.predictors[1:5]
```

```
##        1        2        3        4        5 
## 3.717848 3.582696 3.544246 3.347555 3.300310
```

```
mod1$fitted.values[1:5]
```

```
##        1        2        3        4        5 
## 41.17568 35.97038 34.61358 28.43313 27.12105
```

```
model.matrix(mod1) #extraer matriz
```

```
##    (Intercept)    Biomasa pHmid pHhigh Biomasa:pHmid Biomasa:pHhigh
## 1            1 0.46929722     0      1     0.0000000     0.46929722
## 2            1 1.73087043     0      1     0.0000000     1.73087043
## 3            1 2.08977848     0      1     0.0000000     2.08977848
## 4            1 3.92578714     0      1     0.0000000     3.92578714
## 5            1 4.36679265     0      1     0.0000000     4.36679265
## 6            1 5.48197468     0      1     0.0000000     5.48197468
## 7            1 6.68468591     0      1     0.0000000     6.68468591
## 8            1 7.51165063     0      1     0.0000000     7.51165063
## 9            1 8.13220251     0      1     0.0000000     8.13220251
## 10           1 9.57212864     0      1     0.0000000     9.57212864
## 11           1 0.08665367     0      1     0.0000000     0.08665367
## 12           1 1.23697390     0      1     0.0000000     1.23697390
## 13           1 2.53204324     0      1     0.0000000     2.53204324
## 14           1 3.40794153     0      1     0.0000000     3.40794153
## 15           1 4.60504596     0      1     0.0000000     4.60504596
## 16           1 5.36771709     0      1     0.0000000     5.36771709
## 17           1 6.56084215     0      1     0.0000000     6.56084215
## 18           1 7.24206214     0      1     0.0000000     7.24206214
## 19           1 8.50363299     0      1     0.0000000     8.50363299
## 20           1 9.39095342     0      1     0.0000000     9.39095342
## 21           1 0.76488801     0      1     0.0000000     0.76488801
## 22           1 1.17647020     0      1     0.0000000     1.17647020
## 23           1 2.32512082     0      1     0.0000000     2.32512082
## 24           1 3.22288207     0      1     0.0000000     3.22288207
## 25           1 4.13612930     0      1     0.0000000     4.13612930
## 26           1 5.13717652     0      1     0.0000000     5.13717652
## 27           1 6.42193811     0      1     0.0000000     6.42193811
## 28           1 7.06552638     0      1     0.0000000     7.06552638
## 29           1 8.74592918     0      1     0.0000000     8.74592918
## 30           1 9.98177013     0      1     0.0000000     9.98177013
## 31           1 0.17576270     1      0     0.1757627     0.00000000
## 32           1 1.37677830     1      0     1.3767783     0.00000000
## 33           1 2.55104256     1      0     2.5510426     0.00000000
## 34           1 3.00027434     1      0     3.0002743     0.00000000
## 35           1 4.90562386     1      0     4.9056239     0.00000000
## 36           1 5.34330542     1      0     5.3433054     0.00000000
## 37           1 7.70000000     1      0     7.7000000     0.00000000
## 38           1 0.55368893     1      0     0.5536889     0.00000000
## 39           1 1.99029644     1      0     1.9902964     0.00000000
## 40           1 2.91263671     1      0     2.9126367     0.00000000
## 41           1 3.21645133     1      0     3.2164513     0.00000000
## 42           1 4.97988468     1      0     4.9798847     0.00000000
## 43           1 5.65872290     1      0     5.6587229     0.00000000
## 44           1 8.10000000     1      0     8.1000000     0.00000000
## 45           1 0.73956986     1      0     0.7395699     0.00000000
## 46           1 1.52693420     1      0     1.5269342     0.00000000
## 47           1 2.23212239     1      0     2.2321224     0.00000000
## 48           1 3.88528818     1      0     3.8852882     0.00000000
## 49           1 4.62650541     1      0     4.6265054     0.00000000
## 50           1 5.12096844     1      0     5.1209684     0.00000000
## 51           1 8.30000000     1      0     8.3000000     0.00000000
## 52           1 0.51127858     1      0     0.5112786     0.00000000
## 53           1 1.47823269     1      0     1.4782327     0.00000000
## 54           1 2.93455800     1      0     2.9345580     0.00000000
## 55           1 3.50548891     1      0     3.5054889     0.00000000
## 56           1 4.61790914     1      0     4.6179091     0.00000000
## 57           1 5.69696382     1      0     5.6969638     0.00000000
## 58           1 6.09301688     1      0     6.0930169     0.00000000
## 59           1 0.73006280     1      0     0.7300628     0.00000000
## 60           1 1.15806838     1      0     1.1580684     0.00000000
## 61           1 0.10084790     0      0     0.0000000     0.00000000
## 62           1 0.13859609     0      0     0.0000000     0.00000000
## 63           1 0.86351508     0      0     0.0000000     0.00000000
## 64           1 1.29291903     0      0     0.0000000     0.00000000
## 65           1 2.46916355     0      0     0.0000000     0.00000000
## 66           1 2.36655309     0      0     0.0000000     0.00000000
## 67           1 2.62921708     0      0     0.0000000     0.00000000
## 68           1 3.25228652     0      0     0.0000000     0.00000000
## 69           1 4.41727619     0      0     0.0000000     0.00000000
## 70           1 4.78081039     0      0     0.0000000     0.00000000
## 71           1 0.05017529     0      0     0.0000000     0.00000000
## 72           1 0.48283691     0      0     0.0000000     0.00000000
## 73           1 0.65266714     0      0     0.0000000     0.00000000
## 74           1 1.55533656     0      0     0.0000000     0.00000000
## 75           1 1.67163820     0      0     0.0000000     0.00000000
## 76           1 2.87005390     0      0     0.0000000     0.00000000
## 77           1 2.51072052     0      0     0.0000000     0.00000000
## 78           1 3.49760385     0      0     0.0000000     0.00000000
## 79           1 3.67876186     0      0     0.0000000     0.00000000
## 80           1 4.83154245     0      0     0.0000000     0.00000000
## 81           1 0.28972266     0      0     0.0000000     0.00000000
## 82           1 0.07756009     0      0     0.0000000     0.00000000
## 83           1 1.42902041     0      0     0.0000000     0.00000000
## 84           1 1.12074092     0      0     0.0000000     0.00000000
## 85           1 1.50795384     0      0     0.0000000     0.00000000
## 86           1 2.32596318     0      0     0.0000000     0.00000000
## 87           1 2.99570582     0      0     0.0000000     0.00000000
## 88           1 3.53819909     0      0     0.0000000     0.00000000
## 89           1 4.36454121     0      0     0.0000000     0.00000000
## 90           1 4.87050789     0      0     0.0000000     0.00000000
## attr(,"assign")
## [1] 0 1 2 2 3 3
## attr(,"contrasts")
## attr(,"contrasts")$pH
## [1] "contr.treatment"
```

```
coef(mod1)
```

```
##    (Intercept)        Biomasa          pHmid         pHhigh 
Biomasa:pHmid 
##      2.9525524     -0.2621580      0.4841087      0.8155712 
0.1231362 
## Biomasa:pHhigh 
##      0.1550282
```

```
especies [50,]
```

```
##     pH  Biomasa Especie
## 50 mid 5.120968      20
```

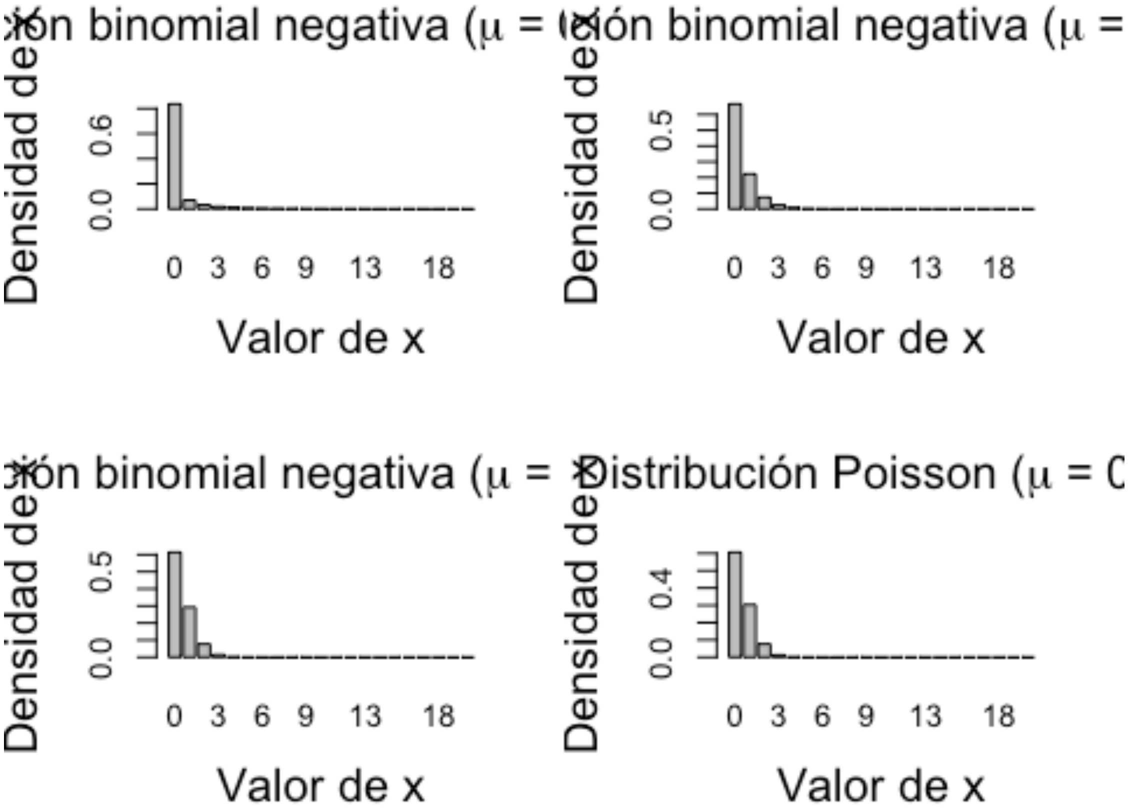
```
linpred<-model.matrix(mod1)[50,]%*%coef(mod1) #calcular valor predicho de 
la observaci´ón 50
exp(linpred) #calcular valor predicho misma escala que la observación 
original
```

```
##          [,1]
## [1,] 15.25237
```

Regresión Binomial negativa

Función de densidad para valores que siguen una distribución binomial negativa con parámetro de dispersión size=k

```
y1<-dnbinom(x=0:20, mu=0.5, size=0.1)
y2<-dnbinom(x=0:20, mu=0.5, size=1)
y3<-dnbinom(x=0:20, mu=0.5, size=10)
y4<-dpois(x=0:20, lambda=0.5)
layout(mat=matrix(1:4, nrow=2, ncol=2, byrow=TRUE))
barplot(y1, main=expression(paste("Distribución binomial negativa (", mu, 
" = 0.5, ", italic(k), " = 0.1)")), names.arg=0:20, ylab="Densidad de x", 
xlab="Valor de x", cex.lab=1.5, cex.main=1.5)
barplot(y2, main=expression(paste("Distribución binomial negativa (", mu, 
" = 0.5, ", italic(k), " = 1)")), names.arg=0:20, ylab="Densidad de x", 
xlab="Valor de x", cex.lab=1.5, cex.main=1.5)
barplot(y3, main=expression(paste("Distribución binomial negativa (", mu, 
" = 0.5, ", italic(k), " = 10)")), names.arg=0:20, ylab="Densidad de x", 
xlab="Valor de x", cex.lab=1.5, cex.main=1.5)
barplot(y4, main=expression(paste("Distribución Poisson (", mu, " = 
0.5)")), names.arg=0:20, ylab="Densidad de x", xlab="Valor de x", 
cex.lab=1.5, cex.main=1.5)
```

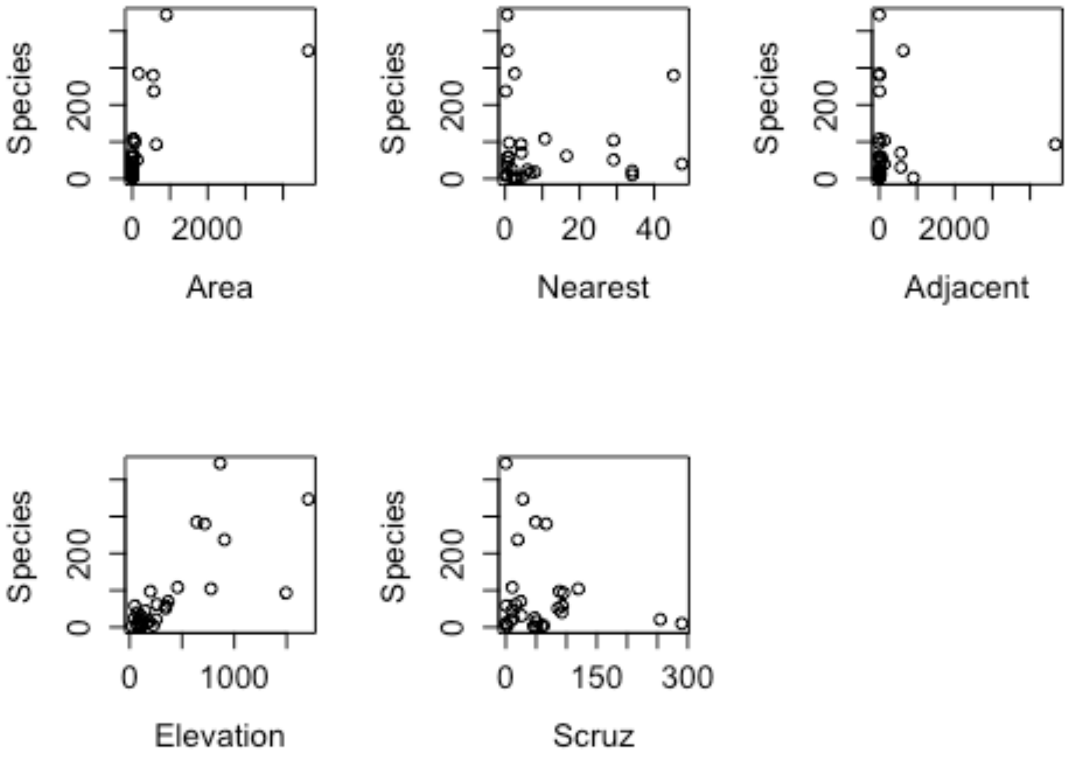


Cargar paquete faraway que contiene los datos gala

```
library(faraway)
data(gala)
attach(gala)
```

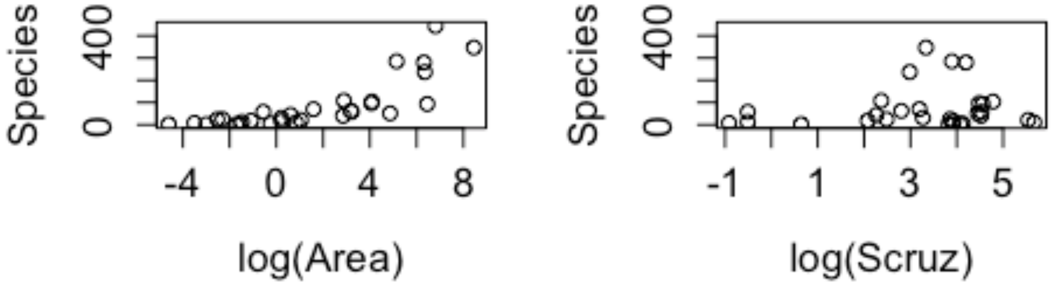
Checar relaciones entre variables

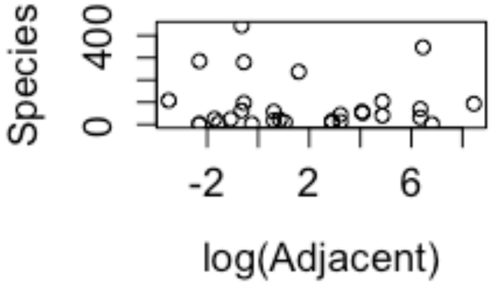
```
layout(mat=matrix(1:6, nrow=2, ncol=3))
plot(Species~Area, cex.lab=1.3, cex.axis=1.3)
plot(Species~Elevation, cex.lab=1.3, cex.axis=1.3)
plot(Species~Nearest, cex.lab=1.3, cex.axis=1.3)
plot(Species~Scruz, cex.lab=1.3, cex.axis=1.3)
plot(Species~Adjacent, cex.lab=1.3, cex.axis=1.3)
```



Transformar a logaritmo para mejorar distribuci´ón de observaciones

```
layout(mat=matrix(1:4, nrow=2, ncol=2))
plot(Species~log(Area), cex.lab=1.3, cex.axis=1.3)
plot(Species~log(Adjacent), cex.lab=1.3, cex.axis=1.3)
plot(Species~log(Scruz), cex.lab=1.3, cex.axis=1.3)
```





Linealizar relaciones

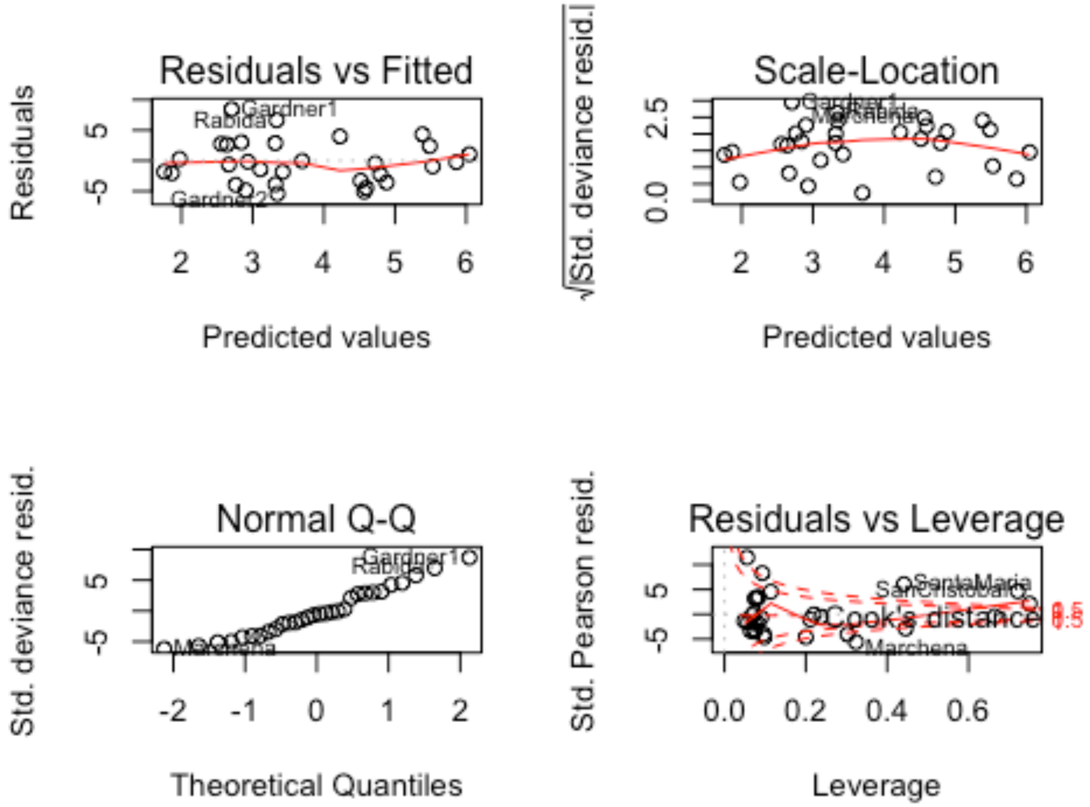
```
gala$log_Area<-log(gala$Area)
gala$log_Adjacent<-log(gala$Adjacent)
gala$log_Scruz<-log(gala$Scruz+1)
```

Correr modelo

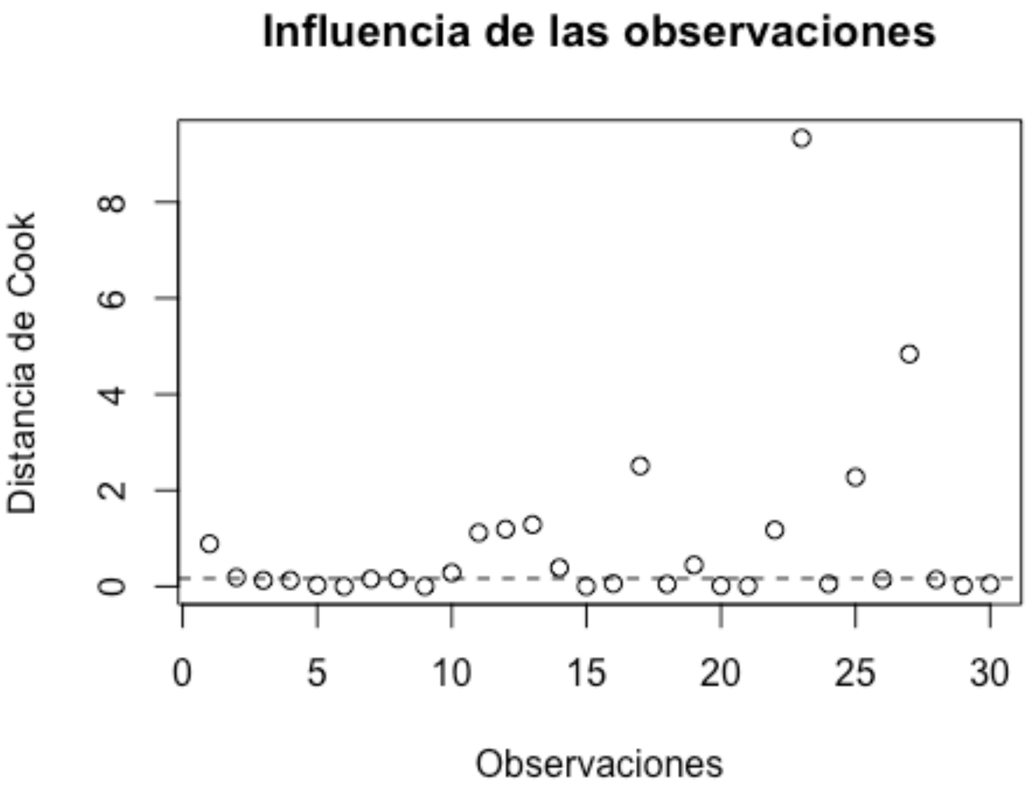
```
gala1<-glm(Species~log_Area+Elevation+Nearest+log_Adjacent+log_Scruz,
data=gala, family=poisson)
summary(gala1)

##
## Call:
## glm(formula = Species ~ log_Area + Elevation + Nearest + log_Adjacent
+
##     log_Scruz, family = poisson, data = gala)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.4341  -3.0204  -0.5416   2.5770   8.4359
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.505e+00  6.368e-02  55.048  < 2e-16 ***
## log_Area      3.575e-01  1.615e-02  22.135  < 2e-16 ***
## Elevation     5.663e-05  1.044e-04   0.543  0.58732
## Nearest      -3.431e-03  1.674e-03  -2.050  0.04038 *
## log_Adjacent -9.130e-02  8.760e-03 -10.423  < 2e-16 ***
## log_Scruz    -5.003e-02  1.535e-02  -3.259  0.00112 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3510.73  on 29  degrees of freedom
## Residual deviance:  361.17  on 24  degrees of freedom
## AIC: 534
##
## Number of Fisher Scoring iterations: 5
```

Verificar supuestos

```
layout(mat=matrix(1:4, nrow=2, ncol=2))
plot(gala1)
```



```
plot(cooks.distance(gala1), ylab="Distancia de Cook",
xlab="Observaciones", main="Influencia de las observaciones")
abline(h=4/(30-6), lty=2)
```



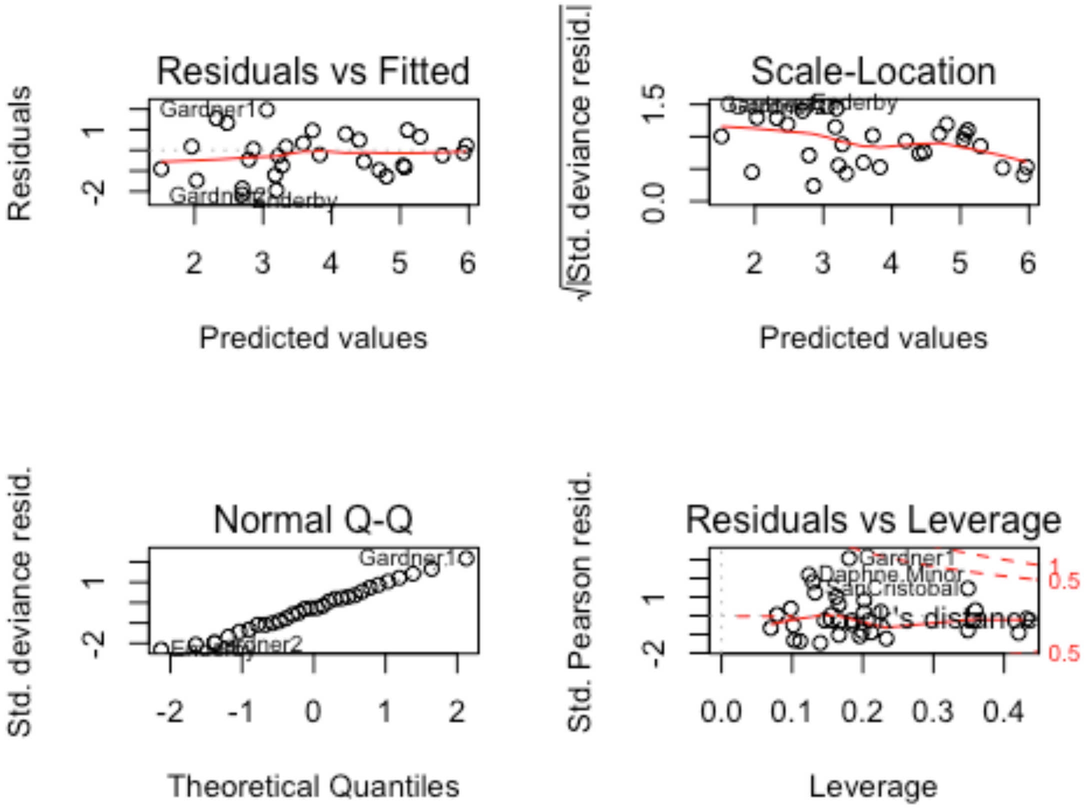
Chi-square/c_hat. Verificar sobredispersi´ón

```
chisq<-sum(((gala$Species-gala1$fitted)^2)/gala1$fitted)
c.hat<-chisq/gala1$df.residual
```

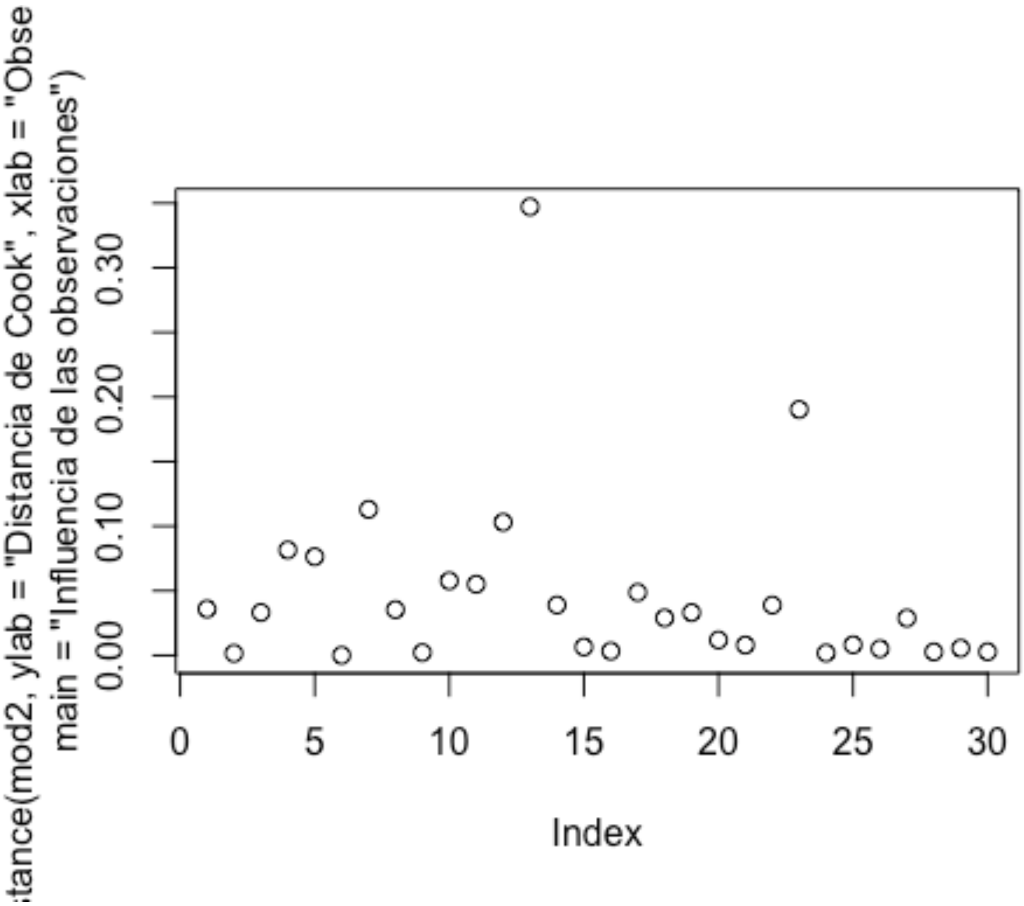
Intentar ajustear un modelo de regresión binomial

```
library(MASS)
mod2<-glm.nb(Species~log_Area+Elevation+Nearest+log_Adjacent+log_Scruz,
data=gala)
summary(mod2)

##
## Call:
## glm.nb(formula = Species ~ log_Area + Elevation + Nearest +
log_Adjacent +
##     log_Scruz, data = gala, init.theta = 2.938490105, link = log)
##
## Deviance Residuals:
##     Min        1Q    Median       3Q       Max
## -2.1466  -0.8867  -0.2338   0.4581   1.9671  
## 
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)    
## (Intercept)    3.5621341  0.2971453  11.988  < 2e-16 ***
## log_Area       0.4215363  0.0660078   6.386  1.7e-10 ***
## Elevation     -0.0005481  0.0005335  -1.027    0.304    
## Nearest       -0.0137115  0.0102400  -1.339    0.181    
## log_Adjacent -0.0250659  0.0376654  -0.665    0.506    
## log_Scruz     -0.0277083  0.0878747  -0.315    0.753    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Negative Binomial(2.9385) family taken to be
1)
## 
##     Null deviance: 149.161  on 29  degrees of freedom
## Residual deviance:  32.782  on 24  degrees of freedom
## AIC: 287.9
## 
## Number of Fisher Scoring iterations: 1
## 
## 
##                Theta:  2.938 
##            Std. Err.:  0.866 
## 
##  2 x log-likelihood:  -273.896
```

```
layout(mat=matrix(1:4, nrow=2))
plot(mod2)
```



```
plot(cooks.distance(mod2, ylab="Distancia de Cook", xlab="Observaciones",
main="Influencia de las observaciones"))
```



Correr modelo con glm y parámetro de dispersión obtenido en mod2

```
mod2b<-glm(Species~log_Area+Elevation+Nearest+log_Adjacent+log_Scruz,
data=gala,family=negative.binomial(theta=2.938))

logLik(mod2)

## 'log Lik.' -136.9479 (df=7)

#theta es un parámetro adicional
logLik(mod2b)

## 'log Lik.' -136.9479 (df=6)
```