

CAPÍTULO 4

Aplicaciones de Modelos Lineales Generalizado (GLM) para datos de conteo

José Bruno Malaquias, Jéssica Karina da Silva Pachú, Filipe Lemos Jacques, Paulo Eduardo Degrande, Milton Fernando Cabezas Guerrero

Resumen

En esta última sección, presentamos dos ejemplos de datos que son de naturaleza discreta. Para ello, utilizamos la prueba de ajuste de Modelos Lineales Generalizados (GLM) con las siguientes distribuciones: Poisson, cuasi-Poisson y Binomial Negativo. Una de las bases de datos tiene un solo factor, mientras que la segunda base de datos está formada por un bifactorial. Usaremos los siguientes paquetes R: [readxl](#), [hnp](#), [MASS](#), [multcomp](#) y [Rmisc](#). Los comandos a ejecutar en R se muestran en color azul.

Palabras claves: desviación; variable discreta; número de insectos; contrastes; glm.

1. Importación de base de datos

La base de datos a trabajar es la ocurrencia de *Scaptocoris castanea* Perty, 1830 (Hemiptera: Cydnidae) en diferentes cultivos. Para acceder a la base de datos haga [clic aquí](#), la misma que también se expone en el apéndice de este capítulo.

- *Host* es la planta huésped;
- *Rep* corresponde al bloque (repetición) y
- *y* es la variable de respuesta que se estudió como ocurrencia de insectos.

Antes de iniciar el análisis debemos ejecutar el comando: `rm(list=ls(all=TRUE))` que sirve para borrar la memoria del programa y evitar errores.

Después de cambiar nuestro directorio (seleccione la carpeta que contiene los archivos que usaremos): **Session -> Set working Directory -> Choose Directory**

Podemos ver los nombres de todos los archivos contenidos en la carpeta seleccionada: **list.files()**

Luego debemos usar el comando para cargar el paquete que lee la hoja de Excel, que es el paquete readxl [1]: **require(readxl)**

Luego de cargar el paquete, necesitamos leer el archivo, para eso usaremos la línea de comando:

```
data<-read_excel("Datos-Abundancia.xlsx", sheet = 1)
```

- data es el nombre que se le da al dataframe (puede poner el nombre que desee);
- read_excel es el comando para leer el archivo de Excel;
- Abundancia de datos es el nombre de su archivo dentro de la carpeta seleccionada;
- xlsx es la extensión del archivo con el que está trabajando;
- sheet = 1 hace referencia a qué pestaña de su archivo de Excel desea analizar.

2. Análisis descriptivo

El primer paso es la producción de un análisis descriptivo, para ello usaremos el paquete **Rmisc** [2], y para cargarlo usaremos el comando: **require(Rmisc)**.

Luego con la función **summarySE** obtendremos los valores medios, desviación estándar, error estándar e intervalos de confianza asociados a la media, para eso debemos ejecutar la línea de comando:

```
(descriptiva<- summarySE(data, measurevar="y", groupvars=c("Host")))
```

El resultado del análisis descriptivo se muestra a continuación:

	Host	N	y	sd	se	ci
1	Algodão	10	194.8	53.1722568	16.8145440	38.0371411
2	Brachiaria	10	17.5	6.7535999	2.1356758	4.8312343
3	Crotalaria	10	25.5	19.9568980	6.3109253	14.2763048
4	Girassol	10	0.6	0.6992059	0.2211083	0.5001818
5	Guandu	10	1.9	1.7288403	0.5467073	1.2367379
6	Milho	10	366.2	98.5391067	31.1608016	70.4906305
7	Mucuna Preta	10	1.6	1.8378732	0.5811865	1.3147353
8	Pousio	10	0.2	0.4216370	0.1333333	0.3016210
9	Soja	10	58.3	29.4016250	9.2976102	21.0326555
10	Sorgo	10	1.6	2.0110804	0.6359595	1.4386403

El siguiente paso es elegir el modelo que mejor se adapte a este conjunto de datos.

3. Prueba de ajuste del modelo

Cuando la variable es independiente, debemos verificar si son factores, para eso usamos el comando:

```
is.factor(data$Host)
```

La respuesta para la verificación debe ser: **TRUE**

Si la respuesta es: **FALSE**, es necesario convertir las variables independientes en factores, para eso usamos la función “as.factor”:

```
data$Host<-as.factor(data$Host)
```

A continuación, compruebe si la conversión se realizó correctamente:

```
is.factor(data$Host)
```

Con una conversión exitosa, podemos analizar cuál de los 4 modelos se ajusta mejor a los datos, para eso usaremos los modelos:

model1<-glm(y~Host+Rep, data=data): Modelo gaussiano (no es necesario probar este modelo ya que se aplica a variables continuas.

model2<-glm(y~Host+Rep, family="poisson", data=data): Modelo Poisson.

model3<-glm(y~Host+Rep, family="quasipoisson", data=data): Modelo quasi-Poisson.

Algunos modelos necesitan paquetes específicos (como model4, modelo binomial negativo). Para ello es necesario cargar el paquete **MASS** [3] para trabajar con la función **glm.nb**

Require(MASS)

`model4<-glm.nb(y~Host+Rep, data=data)`: Modelo Binomial Negativo.

Luego, es necesario cargar el paquete “**hnp**” para analizar la calidad de ajuste de los modelos a los datos y usamos: `require(hnp)`. El paquete `hnp` fue desarrollado por Moral *et al.* [4], para más detalles sobre media normal con envelope simulado, ver Demetrius [5] y Demetrius *et al.* [6].

Para ver los gráficos en otra ventana, use el comando: `dev.new()`

Para agrupar los 4 gráficos en una cuadrícula de 2 x 2: `par(mfrow=c(2,2))`

Para probar el ajuste de los diferentes modelos usaremos diferentes líneas de comando:

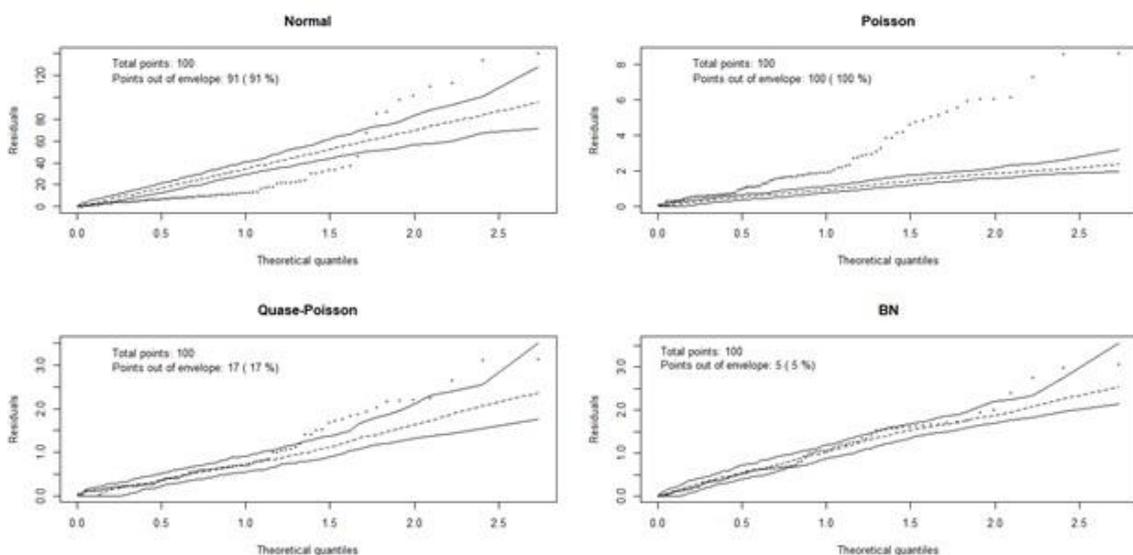
`hnp(model1, print.on="T", main="Normal")`: Prueba de bondad de ajuste del modelo Normal

`hnp(model2, print.on="T", main="Poisson")`: Prueba de bondad de ajuste del modelo Poisson

`hnp(model3, print.on="T", main="QuasePoisson")`: Prueba de bondad de ajuste del modelo quase-Poisson

`hnp(model4, print.on="T", main="BN")`: Prueba de bondad de ajuste del modelo Binomial Negativo

Este es el resultado de ajustar los modelos a los datos de conteo:



Se puede observar que el modelo **Binomial Negativo (BN)** fue el modelo que mejor se ajustó a los datos de conteo de *S. castanea*, ya que solo el 5% de los puntos estaban fuera del envelope. Entonces esta será el modelo seleccionado.

El siguiente paso es ejecutar el comando `anova` con la prueba F para mostrar el análisis de *desviación* del modelo seleccionado. Para ello usaremos el comando:

```
anova(model4, test="F")
```

```
Analysis of Deviance Table
```

```
Model: Negative Binomial(3.3323), link: log
```

```
Response: y
```

```
Terms added sequentially (first to last)
```

	Df	Deviance	Resid.	Df	Resid. Dev	F	Pr(>F)
NULL				99	1189.58		
Host	9	1071.87		90	117.71	119.0965	<2e-16 ***
Rep	9	3.54		81	114.17	0.3935	0.9389

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

De acuerdo con el análisis de *desviación*, es posible verificar que existe evidencia de una diferencia significativa entre las plantas hospedantes (factor *Host*), ya que $Pr(> F)$ es menor a 0.05 (5%).

Ahora comparemos los tratamientos usando el paquete “**multcomp**” [7], para eso ejecutaremos la línea de comando: `require(multcomp)`

Para visualizar las diferencias usaremos la función `glht` y el método Tukey, para eso es necesario ejecutar el comando:

```
(Comparaciones <- summary(glht(model4, linfct = mcp(Host = "Tukey"))))
```

```
Comparaciones <- summary(Comparaciones, test = univariate())
```

A continuación, imprimiremos los resultados expresando las letras de cada tratamiento:

```
(Letras.mP<-cld(Comparaciones, level=0.05, Letters= c(LETTERS, letters), decreasing=TRUE))
```

De la prueba anterior, obtendremos como resultado las comparaciones. Aquellas con las mismas letras se consideran significativamente iguales:

Algodão "B"	Brachiaria "D"	Crotalaria "D"	Girassol "FG"	Guandu "E"	Milho "A"
Mucuna Preta "EF"	Pousio "G"	Soja "C"	Sorgo "EF"		

4. Importación de base de datos - Ejemplo 2

Ejemplo 2 – Datos de conteo en un ensayo en arreglo factorial que contabilizó el número de larvas de *Spodoptera frugiperda* (Lepidoptera: Noctuidae) en diferentes plantas hospedantes (factor 1) en tres estadios larvales (factor 2). Para obtener la base de datos y el script, haga [clic aquí](#)

Para leer los archivos de Excel, necesitamos el paquete "readxl" [1], para eso usamos el comando: `library(readxl)`.

Para leer la base de datos, use el comando:

```
df <- read_excel("DatosConteo.xlsx", sheet = 1)
```

Para ver el encabezado: `head(df, n=2)`

Antes de proceder con el análisis es necesario comprobar si las variables son factores, para ello usamos el comando:

`is.factor(df$Estadio)`: Preguntar si el Estadio es un factor

`is.factor(df$Bloque)`: Preguntar si el Bloque es un factor

`is.factor(df$Cultivo)`: Preguntar si el Cultivo es un factor

Si la respuesta es "FALSE", es necesario convertir las variables.

`df$Estadio<-as.factor(df$Estadio)`: Conversión de la variable Estadio en un factor

`df$Bloque<-as.factor(df$Bloque)`: Conversión de la variable Bloque en un factor

`df$Cultura<-as.factor(df$Cultivo)`: Conversión de la variable Cultivo en un factor

Después de la conversión es necesario comprobar si funcionó.

is.factor(df\$Estadio): Preguntar de nuevo si el Estadio es un factor

is.factor(df\$Bloque): Preguntar de nuevo si el Bloque es un factor

is.factor(df\$Cultivo): Preguntar de nuevo si el Cultivo es un factor

Si la respuesta es "TRUE", podemos continuar con el análisis.

NOTA: si tiene dudas sobre los pasos descritos anteriormente, consulte los capítulos anteriores.

5. Prueba de ajuste

Probemos el ajuste de los modelos de Poisson, Quasi-Poisson y Binomial negativo:

```
Modelo.poisson<-glm(SfrugTot~Estadio*Cultivo+Bloque, family="poisson", data=df)
```

```
Modelo.qpoisson<-glm(SfrugTot~Estadio*Cultivo+Bloque, family="quasipoisson", data=df)
```

library(MASS) – recuerde que necesita cargar el paquete MASS [2] para probar el modelo binomial negativo.

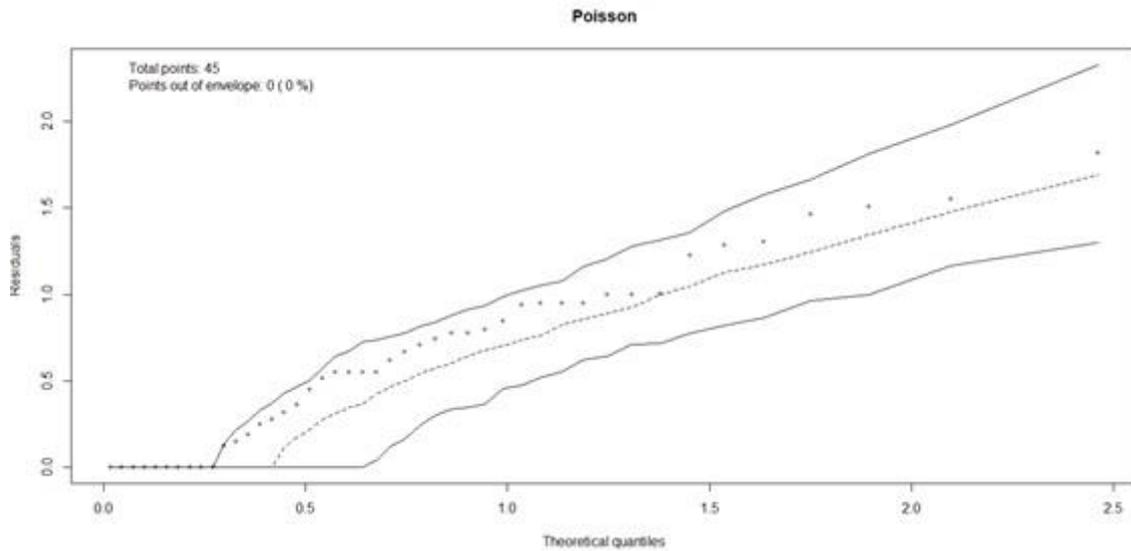
```
Modelo.BN<-glm.nb(SfrugTot~Estadio*Cultivo+Bloque, data=df)
```

Ahora probemos si el modelo se ajusta a los datos, para eso ejecutaremos el paquete hnp [3] con el siguiente comando:

```
require(hnp)
```

```
hnp(Modelo.poisson, print.on="T", main="Poisson")
```

Es posible observar que el modelo de Poisson se ajustó a los datos, ya que ninguno de los puntos estaba fuera de las líneas, por lo que usaremos este modelo.



Después de elegir el modelo, podemos ejecutar el comando ANOVA. Como el modelo de Poisson se ajusta a los datos, usaremos la prueba de chi-cuadrado (Chisq):

```
anova(Modelo.poisson, test="Chisq")
```

Este es el resumen del análisis de *desviación*. Se puede observar que hubo una interacción significativa entre los factores Estadio y Cultivo.

Tabla de análisis de *desviación* - Model: poisson, link: log

Respuesta: SfrugTot

```

Analysis of Deviance Table

Model: poisson, link: log
Response: sfrugTot

Terms added sequentially (first to last)

              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                44      89.170
Estadio              2   4.9117      42      84.259  0.085792 .
Cultura              2  18.6056      40      65.653 9.117e-05 ***
Bloco                4  13.4147      36      52.238 0.009418 **
Estadio:Cultura      4  24.2078      32      28.031 7.257e-05 ***
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

6. Fraccionamiento de la interacción

Como la interacción fue significativa, realizamos el desdoblamiento de la interacción y para ello necesitamos conocer los niveles de cada factor. Para esto usaremos la función `sqldf` del paquete `sqldf` [8]

`sqldf::sqldf("select distinct Estadio from df")`: Verifica los niveles de los factores del Estadio

`sqldf::sqldf("select distinct Cultivo from df")`: Verifica los niveles del factor de Cultivo

- Para el factor Estadio tenemos: V5, V7 y V8.
- En cuanto a Cultivo tenemos: Milhoconv; CornBt y Urochloa

Primero comparemos las etapas dentro de cada cultivo.

Primero, crearemos un subconjunto dentro de cada cultivo:

```
EstadiosMC<-subset(df, df$Cultivo=="Maizoconv")
```

```
EstadiosMBt<-subset(df, df$Cultivo=="MaizBt")
```

```
EstadiosU<-subset(df, df$Cultivo=="Urochloa")
```

Entonces necesitaremos crear un modelo para el cultivo.

```
Modelo.EstadiosMC<-glm(SfrugTot~Estadio+Bloque, family= "poisson",
data=EstadiosMC)
```

```
Modelo.EstadiosMBt<-glm(SfrugTot~Estadio+Bloque, family= "poisson",
data=EstadiosMBt)
```

```
Modelo.EstadiosU<-glm(SfrugTot~Estadio+Bloque, family= "poisson",
data=EstadiosU)
```

Para realizar las comparaciones necesitaremos el paquete "**multcomp**" [7], para ello ejecutaremos la línea de comando (cargará o instalará el paquete si es necesario):

```
if(!require(multcomp)){install.packages("multcomp")}
```

Modelo para datos de maíz convencional:

```
Comp.EstadiosMC<-
summary(glht(Modelo.EstadiosMC,linfct=mcp(Estadio="Tukey")))
```

Modelo para datos de maíz Bt:

```
Comp.EstadiosMBt<-summary(glht(Modelo.EstadiosMBt,
linfct=mcp(Estadio="Tukey")))
```

Modelo para datos de Urochloa:

```
Comp.EstadiosU<-summary(glht(Modelo.EstadiosU,
linfct=mcp(Estadio="Tukey")))
```

Resultados de comparaciones en forma de probabilidad:

```
(Comp.EstadiosMC<-summary(Comp.EstadiosMC, test = univariate()))
```

Linear Hypotheses:

	Estimate	Std. Error	z value	Pr(> z)
v7 - v5 == 0	-1.3863	0.7906	-1.754	0.0795 .
v8 - v5 == 0	-0.9808	0.6770	-1.449	0.1474
v8 - v7 == 0	0.4055	0.9129	0.444	0.6569

signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(univariate p values reported)

```
(Comp.EstadiosMBt<-summary(Comp.EstadiosMBt, test = univariate()))
```

Linear Hypotheses:

	Estimate	Std. Error	z value	Pr(> z)
V7 - V5 == 0	2.472e+01	8.153e+04	0	1
V8 - V5 == 0	3.203e-09	1.153e+05	0	1
V8 - V7 == 0	-2.472e+01	8.153e+04	0	1

(Univariate p values reported)

(Comp.EstadiosU<-summary(Comp.EstadiosU, test = univariate()))

Linear Hypotheses:

	Estimate	Std. Error	z value	Pr(> z)
V7 - V5 == 0	1.7918	1.0801	1.659	0.0971 .
V8 - V5 == 0	2.8332	1.0290	2.753	0.0059 **
V8 - V7 == 0	1.0415	0.4749	2.193	0.0283 *

 signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
 (Univariate p values reported)

De los resultados de las comparaciones en forma de probabilidad, podemos ver que hubo una diferencia significativa solo en la tercera comparación en "V8 - V5" y "V8 - V7".

Para visualizarlo con mayor claridad, es posible asignar letras a las diferencias significativas, con los siguientes comandos:

(Letras.EstadiosMC<-cld(Comp.EstadiosMC, level=0.05, Letters=c(LETTERS, letters), decreasing=TRUE))

V5	V7	V8
"A"	"A"	"A"

(Letras.EstadiosMBt<-cld(Comp.EstadiosMBt, level=0.05, Letters=c(LETTERS, letters), decreasing=TRUE))

V5	V7	V8
"A"	"A"	"A"

(Letras.EstadiosU<-cld(Comp.EstadiosU, level=0.05, Letters=c(LETTERS, letters), decreasing=TRUE))

V5	V7	V8
"B"	"B"	"A"

Asignando letras podemos confirmar la diferencia estadística que se produce en la última comparación.

Ahora comparemos los cultivos dentro de cada Estadio.

Haremos un subconjunto dentro de cada Estadio:

```
CultivosV5<-subset(df, df$Estadio=="V5")
```

```
CultivosV7<-subset(df, df$Estadio=="V7")
```

```
CultivosV8<-subset(df, df$Estadio=="V8")
```

Posteriormente, necesitaremos crear un modelo para cada Estadio.

```
Modelo.V5<-glm(SfrugTot~Cultivo+Bloque, family= "poisson", data=CultivosV5)
```

```
Modelo.V7<-glm(SfrugTot~Cultivo+Bloque, family= "poisson", data=CultivosV7)
```

```
Modelo.V8<-glm(SfrugTot~Cultivo+Bloque, family= "poisson", data=CultivosV8)
```

Realicemos las comparaciones utilizando el mismo método que el anterior, solo contrastando el factor "Cultivo".

```
Comp.CultivosV5<-summary(glht(Modelo.V5, linfct=mcp(Cultivo="Tukey")))
```

```
Comp.CultivosV7<-summary(glht(Modelo.V7, linfct=mcp(Cultivo="Tukey")))
```

```
Comp.CultivosV8<-summary(glht(Modelo.V8, linfct=mcp(Cultivo="Tukey")))
```

Resultados de comparaciones en forma de probabilidad:

```
(Comp.CultivosV5<-summary(Comp.CultivosV5, test = univariate()))
```

Linear Hypotheses:

	Estimate	Std. Error	z value	Pr(> z)
Milhoconv - MilhoBt == 0	22.425	15878.588	0.001	0.9989
Urochloa - MilhoBt == 0	20.345	15878.588	0.001	0.9990
Urochloa - Milhoconv == 0	-2.079	1.061	-1.961	0.0499 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Univariate p values reported)

```
(Comp.CultivosV7<-summary(Comp.CultivosV7, test = univariate()))
```

Linear Hypotheses:

	Estimate	Std. Error	z value	Pr(> z)
Milhoconv - MilhoBt == 0	-0.4055	0.9129	-0.444	0.657
Urochloa - MilhoBt == 0	0.6931	0.7071	0.980	0.327
Urochloa - Milhoconv == 0	1.0986	0.8165	1.346	0.178

(Univariate p values reported)

```
(Comp.CultivosV8<-summary(Comp.CultivosV8, test = univariate()))
```

Linear Hypotheses:

	Estimate	Std. Error	z	value	Pr(> z)
Milhoconv - MilhoBt == 0	2.234e+01	2.485e+04	0.001	0.99928	
Urochloa - MilhoBt == 0	2.407e+01	2.485e+04	0.001	0.99923	
Urochloa - Milhoconv == 0	1.735e+00	6.262e-01	2.770	0.00561	**

 signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
 (Univariate p values reported)

De los resultados de las comparaciones en forma de probabilidad, podemos ver que hubo una diferencia significativa solo en la primera comparación en "Urochloa - Maizconv" y en la última comparación en "Urochloa - Maizconv".

Para visualizarlo con mayor claridad, es posible asignar letras a las diferencias significativas, con los siguientes comandos:

(Letras.CultivosV5<-cld(Comp.CultivosV5, level=0.05, Letters= c(letters, letters), decreasing=TRUE))

MaizBt	Maizconv	Urochloa
"ab"	"a"	"b"

(Letras.CultivosV7<-cld(Comp.CultivosV5, level=0.05, Letters= c(letters, letters), decreasing=TRUE))

MaizBt	Maizconv	Urochloa
"a"	"a"	"a"

(Letras.CultivosV8<-cld(Comp.CultivosV5, level=0.05, Letters= c(letters, letters), decreasing=TRUE))

MaizBt	Maizconv	Urochloa
"ab"	"a"	"b"

Con la asignación de letras podemos confirmar la diferencia estadística que se produce en la última comparación.

Ahora realicemos un análisis descriptivo, para eso necesitamos el paquete "Rmisc" [2]. Ejecutemos la línea de comando que vemos a continuación, el paquete se instalará, si es necesario.

if(!require(Rmisc)){install.packages("Rmisc")}: usemos el paquete "Rmisc"

Luego, hacemos un "summary" para ver un resumen del análisis:

```
summarySE(df, measurevar = "SfrugTot", groupvars = c("Estadio",
"Cultivo"))
```

```
Estadio  Cultura N  sfrugTot      sd      se      ci
V5      MilhoBt  5      0.0  0.0000000  0.0000000  0.0000000
V5      Milhoconv 5      1.6  1.1401754  0.5099020  1.4157148
V5      Urochloa  5      0.2  0.4472136  0.2000000  0.5552890
V7      MilhoBt  5      0.6  1.3416408  0.6000000  1.6658671
V7      Milhoconv 5      0.4  0.8944272  0.4000000  1.1105780
V7      Urochloa  5      1.2  0.8366600  0.3741657  1.0388506
V8      MilhoBt  5      0.0  0.0000000  0.0000000  0.0000000
V8      Milhoconv 5      0.6  0.5477226  0.2449490  0.6800874
V8      Urochloa  5      3.4  2.7018512  1.2083046  3.3547914
```

Los valores que no tienen variabilidad no se pueden considerar en las comparaciones. Los valores obtenidos del análisis descriptivo se pueden colocar en una tabla de la siguiente manera:

Cultivo	Estadios		
	V5	V7	V8
Maíz Bt	0.00±0.00 *	0.60±0.60 a A	0.00±0.00 *
Maíz convencional	1.60±0.50 a A	0.40±0.40 a A	0.60±0.24 a A
Urochloa	0.20±0.20 b B	1.20±0.37 a B	3.40±1.20 b A

Las letras minúsculas comparan los cultivos dentro del factor Estadio (comparación dentro de las columnas), mientras que las letras mayúsculas comparan los Estadios dentro de cada cultivo (comparación dentro de las filas). * Dado que no hubo variabilidad en este tratamiento, no se considerará en las comparaciones.

Anexo - Base de datos 1

Host	Rep	y
Brachiaria	1	13
Brachiaria	2	20
Brachiaria	3	24
Brachiaria	4	17
Brachiaria	5	22
Brachiaria	6	2
Brachiaria	7	17
Brachiaria	8	15
Brachiaria	9	26
Brachiaria	10	19
Crotalaria	1	36
Crotalaria	2	50
Crotalaria	3	1
Crotalaria	4	34
Crotalaria	5	2
Crotalaria	6	48
Crotalaria	7	6
Crotalaria	8	45
Crotalaria	9	27
Crotalaria	10	6
Algodón	1	216
Algodón	2	180
Algodón	3	197
Algodón	4	124
Algodón	5	180
Algodón	6	281
Algodón	7	203
Algodón	8	100
Algodón	9	228
Algodón	10	239
Soya	1	29
Soya	2	4
Soya	3	61
Soya	4	50
Soya	5	34
Soya	6	77
Soya	7	84
Soya	8	70
Soya	9	103
Soya	10	71
Maíz	1	222
Maíz	2	218

Maíz	3	373
Maíz	4	400
Maíz	5	478
Maíz	6	454
Maíz	7	400
Maíz	8	263
Maíz	9	472
Maíz	10	382
Gandul	1	2
Gandul	2	2
Gandul	3	4
Gandul	4	0
Gandul	5	2
Gandul	6	0
Gandul	7	5
Gandul	8	3
Gandul	9	0
Gandul	10	1
Mucuna Negra	1	0
Mucuna Negra	2	0
Mucuna Negra	3	0
Mucuna Negra	4	2
Mucuna Negra	5	4
Mucuna Negra	6	0
Mucuna Negra	7	1
Mucuna Negra	8	1
Mucuna Negra	9	5
Mucuna Negra	10	3
Sorgo	1	2
Sorgo	2	0
Sorgo	3	5
Sorgo	4	4
Sorgo	5	0
Sorgo	6	4
Sorgo	7	0
Sorgo	8	0
Sorgo	9	1
Sorgo	10	0
Girasol	1	2
Girasol	2	0
Girasol	3	1
Girasol	4	0
Girasol	5	1
Girasol	6	0
Girasol	7	0
Girasol	8	0

Girasol	9	1
Girasol	10	1
Barbecho	1	0
Barbecho	2	0
Barbecho	3	1
Barbecho	4	0
Barbecho	5	1
Barbecho	6	0
Barbecho	7	0
Barbecho	8	0
Barbecho	9	0
Barbecho	10	0

Anexo – Banco de Datos 2

Estadio	Cultivo	Bloque	SfrugTot
V5	Maizconv	1	2
V5	Maizconv	2	0
V5	Maizconv	3	1
V5	Maizconv	4	2
V5	Maizconv	5	3
V5	MaizBt	1	0
V5	MaizBt	2	0
V5	MaizBt	3	0
V5	MaizBt	4	0
V5	MaizBt	5	0
V5	Urochloa	1	1
V5	Urochloa	2	0
V5	Urochloa	3	0
V5	Urochloa	4	0
V5	Urochloa	5	0
V7	Maizconv	1	0
V7	Maizconv	2	0
V7	Maizconv	3	0
V7	Maizconv	4	0
V7	Maizconv	5	2
V7	MaizBt	1	3
V7	MaizBt	2	0
V7	MaizBt	3	0
V7	MaizBt	4	0
V7	MaizBt	5	0
V7	Urochloa	1	1
V7	Urochloa	2	2

V7	Urochloa	3	1
V7	Urochloa	4	2
V7	Urochloa	5	0
V8	Maizconv	1	1
V8	Maizconv	2	1
V8	Maizconv	3	0
V8	Maizconv	4	0
V8	Maizconv	5	1
V8	MaizBt	1	0
V8	MaizBt	2	0
V8	MaizBt	3	0
V8	MaizBt	4	0
V8	MaizBt	5	0
V8	Urochloa	1	7
V8	Urochloa	2	3
V8	Urochloa	3	0
V8	Urochloa	4	2
V8	Urochloa	5	5

7. Referencias de paquetes usados

- [1] Wickham H., Bryan J. readxl: Read Excel Files. R package version 1.3.1. 2019. <https://CRAN.R-project.org/package=readxl>
- [2] Hope, R.M. Rmisc: Rmisc: Ryan Miscellaneous. R package version 1.5. <https://CRAN.R-project.org/package=Rmisc>. 2013.
- [3] Venables W.N., Ripley B. D. Modern Applied Statistics with S. Fourth Edition. Springer, New York. 2002.
- [4] Moral, R.A., Hinde, J., Demétrio, C.G.B. Half-normal plots and overdispersed models in R: The hnp package. Journal of Statistical Software, v. 81, n. 10, 2017.
- [5] Demétrio, C.G.B. Modelos lineares generalizados em experimentação agrônômica. USP/ESALQ, 2001.
- [6] Demétrio, C.G., Hinde, J., Moral, R.A. Models for overdispersed data in entomology. In: Ecological modelling applied to entomology. Springer, Cham, 2014. p. 219-259.
- [7] Hothorn, T., Bretz F., Westfall P. Simultaneous inference in general parametric models. Biometrical Journal 50(3), 346--363. 2008.
- [8] Grothendieck, G. sqldf: Manipulate R Data Frames Using SQL. R package version 0.4-11. <https://CRAN.R-project.org/package=sqldf>. 2017.