Fitopatometria |> R

Epifito

Table of contents

Bienvenid@s

"The cornerstone of epidemic analysis" — Campbell and Neher, 1994

Objetivos

Familiarizar al alumnado con herramientas (paquetes) del software R para manipular datos fitopatométricos.

Destinatarios

Agrónomos, Biólogos, Biotecnologos, y áreas afines a la fitopatología, con conocimientos básicos sobre R.

Uso

Este manual web es un compendio de códigos que se utilizaran a lo largo del curso. Se sugiere tener descargados los mismos previamente a la clase para poder ir reproduciendo simultaneamente.

1 Métricas

1.1 Tipos de variables

```r
# obtener todos el mismo set de numeros aleatorios
set.seed(0)
n=100
```

Empecemos simulando datos de incidencia

```r
# nivel de individuo
binomial1 <- rbinom(
  n=n,      # number of observations: sample size
  size=1,   # number of trials
  p=0.3     # probability of success
)
binomial1
hist(binomial1)
abline(v=mean(binomial1), col="red")
```

Ahora con submuestreo

```r
binomial2 <- rbinom(
  n=n,      # number of observations: sample size
  size=10,  # number of trials: sub-sample
  p=0.3     # probability of success
)
binomial2
hist(binomial2)
rug(binomial2)

abline(v=mean(binomial2), col="red")

inc <- binomial2/10
```

```r
# Media poblacional = np
10*0.3

# Varianza = np(1-p)
10*0.3*0.7
sqrt(10*0.3*0.7)
```

Disribucion beta para proporciones, limitada entre 0 y 1:

- incidencia (o prevalencia) de la enfermedad a nivel de muestra o población: proporción de individuos enfermos.
- Severidad: expresada como proporción del área del órgano afectado.

```r
beta1.5 <- rbeta(n = n, shape1 = 1, shape2 = 8)
beta5.1 <- rbeta(n = n, shape1 = 5, shape2 = 1)

hist(beta1.5)
rug(beta1.5)
abline(v=mean(beta1.5), col="red")
hist(beta5.1)
rug(beta5.1)
abline(v=mean(beta5.1), col="red")
```

Compilamos en un data frame / tibble ...

```r
dis_data <- tibble(inc, sev_cond= beta1.5) |>
  rowid_to_column("sample_id")
dis_data

dis_data %>%
  mutate(
    inc_percen = inc*100,
    sev_percen = sev_cond*100,
    sev_media = sev_percen*inc)

dat
```

Variables continuas reales. Ej. Tamaño de lesión, longitud de raiz, etc.

Generalmente se describen mediante la distribución normal. Sin embargo, esta incluye valores negativos. En este caso podemos simular datos con la distribución gamma, que no puede tomar valores negativos.

```r
tam_les <- rgamma(n = n,
                  shape = 10, # valor medio
                  scale = 1)

hist(tam_les)
rug(tam_les)
abline(v=mean(tam_les), col="red")
```

Esclerotos de sclerotinia por capítulo de girasol o nemaotodes por g de raiz son ejemplos de variables de conteos (variables discretas positivas o enteros). Estos pueden ser representados por una distribución de Poisson.

```r
conteos <- rpois(n = n,
                 lambda = 20 # media
                 )
hist(conteos)
rug(conteos)
abline(v=mean(conteos), col="red")
```

1.2 Manipulacion

Ahora veamos una manipulacion multi-nivel de un muestreo multi-regional e inter-anual. Para eso carguemos el dataset Olivo/bacteriosis

```r
load("data/data.RData")
olivo |>  view()
```

dataset formato “wide” (planilla de campo) con 30 columnas de sev por arbol individual [datos simulados]

2) Re-estructuracion —

Pasamos de formato wide a long para hacer operaciones por grupos. Ojo: No siempre debe hacerse este paso aunque nos habilita a group_by()+ summarise() # le pedimos que apile las columnas conteniendo a las plantas 1 a 30 # el nombre de las columnas las apile en una columna llamada “tree” # la observaciones de severidad las apile en una columna llamada sev # el producto de este re-arreglo se llamará “oli_long”

```
olivo |>
  pivot_longer(cols = `1`:`30`,
         names_to = "tree",
         values_to = "sev") -> oli_long
```

Chequeamos cuántos árboles fueron evaluados en cada año/región/lote:

```
oli_long
```

Chequeamos cuantos arboles se evaluaron por campo

```
oli_long |>
  group_by(year, loc, farm) |>
  summarise(n= sum(!is.na(sev))) |>
  pivot_wider(names_from=year,
              values_from = n)
```

Imprimimos los 30 árboles de un mismo lote

```
oli_long |>
  arrange(loc, year) |>
  print(n=30)
```

- Incidencia

(nivel lote - evolución interanual)

Probamos el artilugio matemático que nos permitirá calcular la proporción de árboles enfermos

```
muestra1 <- c(0,1)
mean(muestra1)
```

```
muestra2 <- c(0,0,0,0,1)
mean(muestra2)
```

```
muestra3 <- c(1,1,1,1,1,1,1,1,0,0)
mean(muestra3)
```

Ahora si, aplicaremos el artilugio a nuestros datos.

Tip: pueden ir seleccionando por lineas para ir probando el codigo antes de ejecutarlo por completo (seleccionar hasta antes de cada pipe, sino quedará abierta la sentencia)

```
 oli_long |>
  mutate(diseased = sev>0) |>
  group_by(year, loc, farm) |>
  summarise(inc = mean(diseased, na.rm=TRUE)*100) |>
  ungroup |>
  arrange(loc, year) -> oli_inc
```

Damos print a "oli_inc"

```
oli_inc
```

Graficamos oli_inc (una de las posibilidades)

```
oli_inc |>
  ggplot()+
  # aes(x=factor(year), y=inc) +
  aes(x=factor(year), y=inc, color=factor(farm)) +
  geom_point() +
  # geom_line() +
  geom_line(aes(group=farm)) +
  facet_grid(. ~ loc)
```

- Prevalencia

Nivel región - evolución interanual

```
oli_inc |>
  mutate(diseased_farm = inc>0) |>
  group_by(year, loc) |>
  summarise(prev = mean(diseased_farm, na.rm=TRUE)*100) |>
  ungroup |>
  arrange(loc,year) -> oli_prev
```

```
oli_prev
```

Plot de oli_prev

```
oli_prev |>
  ggplot()+
  aes(x=factor(year), y=prev, color=factor(loc)) +
  geom_point() +
  geom_line(aes(group=loc))
```

- Severidad

Calculamos ambas severidades vistas en la introducción teórica

NOTA: en el teórico la sev_cond daba "NaN" en aquellos casos en que todos los arboles tenian sev=0, y en el filtrado `sev[which(sev > 0)]` el vector quedaba vacío.

```
oli_long |>
  group_by(year, loc, farm) |>
  summarise(
    sev_media = mean(sev, na.rm=TRUE),
    sev_cond =mean(sev[which(sev > 0)])) |>
  ungroup |>
  mutate_all(~replace(., is.nan(.), 0)) |>
  arrange(loc, year) -> oli_sev
oli_sev
```

Print `oli_sev`

```
oli_sev
```

Plot `oli_sev`

- Aprovechamos a usar una función muy eficiente que puede resultar una gran aliada en nuestro trabajo cotidiano: `stat_summary()`

```
oli_sev |>
  ggplot()+
  aes(x=loc, y =sev_media)+
  geom_point(alpha=.3)+
  facet_wrap("year")+
  stat_summary(fun = mean, geom = "crossbar", col="blue")+
  stat_summary(aes(label=..y.. |>  round(1)),
               fun=mean,
               geom="text", size=4, vjust = -0.5)  +
  scale_x_discrete(guide = guide_axis(n.dodge = 2))
```

Referencias