Fitopatometria |> R

- Juan Edwards, Juan Paredes y Bruno Pugliese

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 básicas

```
# Setup
if (!require("pacman")) install.packages("pacman")
pacman::p_load(tidyverse, rio)
```

1.1 Tipos de variables

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

Empecemos simulando datos de incidencia

```
# 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

```
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)
```

```r
rug(binomial2)

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

inc <- binomial2/10

# Media poblacional = np
10*0.3

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

Disribución 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)
```

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.

```
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.

```
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

```
# load("data/data.RData")
olivo <- rio::import("https://raw.githubusercontent.com/epifito/fitopatometria-r/main/data
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))
```

2 Métricas compuestas

```
# if (!require("pacman")) install.packages("pacman")
pacman::p_load(tidyverse, epifitter, emmeans, performance, ggResidpanel, multcomp, multcom
theme_set(theme_light())
```

2.1 AUC

Area bajo la curva de progreso de la enfermedad

Reproducción de: APS-AUDPC

```
epi <- tibble(
  time = c(1,2,3,4),
  dis = c(1,2,3,10))
```

```
epi %>%
  ggplot()+
  aes(x=time, y = dis)+
  geom_point()+
  geom_line()
```

Area bajo la curva del progreso de la enfermedad (ABC) - Absoluta

```
abc_1  <- with(epi,
                 AUDPC(time = time,
                  y = dis,
                  y_proportion = FALSE,
                  type = "absolute"))
abc_1
```

ABC standarizada

```r
sabc1 <- abc_1/(4-1)
sabc1
```

Aplicación a un caso real

Reproducción de: APS Stripe rust

```r
## Set up vector for Madras AUDPC Chart
dat <- tibble(
  dai = c(0,10,20,30,40,50,60,70,80,90,100),
  sev_68 = c(0,0,0,0,3,20,50,80, 90, 100, 100),
  sev_69 = c( 0,0,0,0,0,0,0,3,6,30,70)
)
dat
```

```r
dat %>%
  ggplot()+
  aes(x = dai) +
  geom_line(aes(y = sev_68), col="red")+
  geom_line(aes(y = sev_69), col="blue")
```

```r
# Epidemia de 1968
with(dat,
     AUDPC(time = dai,
      y = sev_68,
      y_proportion = FALSE,
      type = "absolute"))
```

```r
# Epidemia de 1969
with(dat,
     AUDPC(time = dai,
      y = sev_69,
      y_proportion = FALSE,
      type = "absolute"))
```

```r
# un poco de coding
dat %>%
  pivot_longer(
    cols= c(sev_68, sev_69),
    names_to = "epidemia",
```

```
    values_to = "sev",
    names_prefix = "sev_")-> dat_long
```

```
dat_long %>%
  ggplot()+
  aes(x = dai, y = sev, col=epidemia) +
  geom_line()
```

```
dat_long %>%
  group_by(epidemia) %>%
  summarise(abc=AUDPC(time = dai,
                      y = sev,
                      y_proportion = FALSE,
                      type = "absolute"))
```

2.2 DSI

```
# poroto  <-  rio::import("data/poroto.csv")
poroto <- read.csv("https://raw.githubusercontent.com/epifito/fitopatometria-r/main/data/p
```

Disease severity index (Indice de severidad)

- Poroto/sclerotinia

Dataset de formato wide, que incluye 3 variables descriptivas y 4 variables respuesta.

```
poroto %>%
  mutate(diseased = class_1 + class_2 + class_3 + class_4)  %>%
  mutate(inc_p = diseased/n) %>%
  # mutate(dsi_eq1 = (1*class_1+2*class_2+3*class_3+4*class_4)/(n*4) *100) %>%
  # mutate(dsi_eq2 = (1*class_1+2*class_2+3*class_3+4*class_4)/n) %>%
  mutate(dsi = (.13*class_1 +.375*class_2 + .625*class_3 + .875*class_4)/n*100) %>%
  mutate(dsi_p = dsi/100) %>%
  mutate_at(vars(trt, rep), as.factor) -> poroto_dsi
poroto_dsi
```

```r
poroto_dsi %>%
  ggplot() +
  aes(x=trt, y =dsi) +
  geom_point(alpha=.3)
```

Model fitting

```r
mod1 <- lm(dsi ~ trt, data = poroto_dsi)
```

```r
resid_panel(mod1, plots = c("resid", "qq"))
check_heteroscedasticity(mod1)
check_normality(mod1)
cld(emmeans(mod1, ~ trt, type = "response"))
```

```r
mod2 <- lm(sqrt(dsi) ~ trt, data = poroto_dsi)
```

```r
resid_panel(mod2, plots = c("resid", "qq"))
check_heteroscedasticity(mod2)
check_normality(mod2)
cld(emmeans(mod2, ~ trt, type = "response"))
```

```r
asin_tran <- make.tran("asin.sqrt", 100)
mod3 <- with(asin_tran,
             lm(linkfun(dsi) ~ trt, data = poroto_dsi)
             )
```

```r
resid_panel(mod3, plots = c("resid", "qq"))
check_heteroscedasticity(mod3)
check_normality(mod3)
cld(emmeans(mod3, ~ trt, type = "response"))
```

```r
mod4 = lm(log(dsi_p/(1-dsi_p)) ~ trt, data = poroto_dsi)
```

```r
resid_panel(mod4, plots = c("resid", "qq"))
check_heteroscedasticity(mod4)
check_normality(mod4)
cld(emmeans(mod4, ~trt,
```

```
        tran = "logit",
        type = "response"))
```

```
compare_performance(mod1, mod2, mod3, mod4)
```

2.3 Enfermedades que inducen senescencia

Ensayo de fungicidas en cebada (trt=3). DBCA con 4 rep. Evaluaciones de sev media a los 0, 9, 20 y 29 dias desde aplicado. Estimacion de AF activa (lo que no es senescencia) La senescencia no cuenta para ninguna enfermedad, ya que es imposible distinguir su causa.

```
cebada_raw <- read.csv("https://raw.githubusercontent.com/epifito/fitopatometria-r/main/da
```

Hacemos un cebada long solo con fines graficos, entonces no creamos `cebada_long`.

```
cebada_long <- cebada_raw %>%
  pivot_longer(
    cols = c("verdor", "e1_sev", "e2_sev"),
    names_to = "var",
    values_to = "val") %>%
  mutate(var = factor(var),
         var = fct_relevel(var, "verdor"))

cebada_long %>%
  ggplot()+
  aes(x=dias, y=val, col = var)+
  facet_wrap("trt")+
  geom_point(alpha=0.3) +
  stat_summary(fun=mean, geom="line",
               size=0.7, alpha=.5,
               aes(col=var, group=var)) +
  scale_color_manual(
    labels = c("AF",  "Sev mancha en red (%)", "Sev escaldadura (%)"),
    values = c("green", "red", "blue")
  ) +
  theme_bw()+
  labs(title = "Evolución área foliar",
       y = "%", x = "Días desde aplicado",
       col = "")
```

Ahora calculamos el AF sana (%), restando al AF activa, la severidad media de mancha en red y escaldadura.

```
cebada <- cebada_raw %>%
  mutate(af_sana = verdor - e1_sev - e2_sev) %>%
  mutate(sev_tot = e1_sev + e2_sev)
cebada
```

Finalmente calculamos el AUC del AF sana (LAI)

```
cebada %>%
  group_by(trt, rep) %>%
  summarize(auc = AUDPC(time = dias,
                        y = af_sana,
                        y_proportion = FALSE,
                        type = "absolute")) -> cebada_auc
```

```
cebada_auc <- cebada_auc %>%
  mutate_at(vars(trt, rep), as.factor)
```

```
cebada_auc %>%
  ggplot()+
  aes(y=auc, x=trt, col=rep)+
  geom_point()
```

3 GLM binomial

3.1 Demos

- Dist. Normal

```
set.seed(1)
x <- rnorm(n=100,    # sample size
           mean=10,  # mean of sample
           sd=3      # standard deviation of sample
          )
head(x)
mean(x)
sd(x)

# predictor linear
mu = 3 + 2*x
plot(x, mu)

# generamos el componente aleatorio con distribucion normal de los errores
# set.seed(1)
y  <- mu + rnorm(100, 0, 3)

plot(x,y)
```

- LM - repaso

```
mod1 <- lm(y~x)
plot_model(mod1, type='pred', show.data=T, ci.lvl = NA)
summary(mod1)
```

De este modelo entendemos que cuando x=0, y=2.89 y por cada aumento unitario de x, y aumenta 1.99 unidades

y = 2.89 + 1.99 * x

Ahora veamos el mismo ajuste usando “glm”

```
mod1.1 <- glm(y~x, family = gaussian)
plot_model(mod1.1, type='pred', show.data=T, ci.lvl = NA)
summary(mod1.1)
```

y = 2.89 + 1.99 * x

Exactamente los mismos coeficientes que mod1

```
gaussian()
binomial()
poisson()
```

- Dist. Binomial

Entendamos la naturaleza binaria de la incidencia.

Imaginemos que estamos entrando en un campo de soja y queremos estimar la incidencia de una enfermedad foliar X

Una estacion (unidad) de muestreo de tamaño 30

```
set.seed(1)

bin_1 <- rbinom(
  1,        # numero de observaciones o simulaciones (estaciones de muestreo)
  size=30,  # numero de ensayos (n)
  p=0.1     # probabilidad de exito (p)
)
bin_1
bin_1/30    # 1 valor de incidencia de estacion de muestreo

# 0.066 -> 6,6% incidencia media del lote

# gghist(bin_1, e=30*0.1, m = mean(bin_1))
```

10 estaciones de muestreo de tamaño 30

```
set.seed(1)

bin_2 <- rbinom(
  10,       # numero de observaciones
```

```
  size=30,  # numero de ensayos (n)
  p=0.1     # probabilidad de exito (p)
)
bin_2       # muestra compuesta de 10 estaciones de muestreo con n=30
bin_2/30    # 10 valores de incidencia de n=30

# gghist(bin_2, e=30*0.1, m=mean(bin_2))

mean(bin_2/30)
# 0.11 -> 11% incidencia media del lote
```

100 estaciones de muestreo de tamaño 30

```
set.seed(1)
bin_3 <- rbinom(
  100,    # numero de observaciones
  size=30,  # numero de ensayos
  p=0.1     # probabilidad de exito
)
bin_3

# gghist(bin_3, e=30*0.1, m=mean(bin_3))
bin_3/30 # 100 valores de incidencia de n=30
mean(bin_3/30)
# 0.101 -> 10.1% incidencia media del lote
```

```
# media = np
media = 30*0.1
media

# varianza  = np(1-p)
varianza =  30*0.1*(1-0.1)
varianza

# sd = sqrt(np(1-p))
sd =  sqrt(30*0.1*(1-0.1))
sd
# sd_field =  1.643168
```

Variable aleatoria X que es distribuida $X \sim binomial(n, p)$ con media $\mu = np$ y varianza $\sigma 2 = np(1-p)$, siendo X el conteo de eventos exitosos en n ensayos Bernoulli idénticos e independientes con probabilidad de éxito p constante.

3.2 DBCA

```
phom_raw <- import("https://raw.githubusercontent.com/juanchiem/glm_webinar/main/data/phom
# phom_raw <- rio::import("data/phomopsis.csv") %>% tibble
```

Efecto de tratamientos de fungicidas sobre tizon foliar por Phomopsis en frutilla (Madden et al. 2002)

- Patógeno: *Phomopsis obscurans*
- Diseño en bloques completos aleatorizados (RCBD)
- Cuatro bloques (bk, j = 1, ..., 4)
- Ocho tratamientos: control no tratado + 7 fungicidas (trt, i = 1, ..., 8) aleatorizados dentro de cada bloque
- Variable respuesta (Y): Numero de foliolos enfermos
- n Tamaño de la muestra
- Incidencia por parcela = y/n
- Acondicionamiento

```
phom_raw

# Factorizamos nuestros variebles independientes (predictoras) y calculamos la incidencia

phom_dat <- phom_raw %>%
  mutate_at(vars(trt, bk), as.factor) %>%
  mutate(inc=y/n) %>%
  arrange(trt)
phom_dat
```

- Visualización

```
phom_dat %>%
  ggplot() +
  aes(x=trt, y = inc) +
  geom_boxplot(alpha=.5, width = .2) +
  geom_point(alpha=.7) +
  labs(x="Tratamientos", y="Incidencia (proporción)")
```

- Modelos mixtos

Efecto fijo al tratamiento y aleatorio a los bloques

- LM

```
# pacman::p_load(lmerTest)
mod_phom_LM <- lmer(inc ~ trt + (1|bk),
                    data=phom_dat)
performance::check_homogeneity(mod_phom_LM)
performance::check_normality(mod_phom_LM)
```

```
car::Anova(mod_phom_LM, type="III")
summary(mod_phom_LM)
```

Podriamos avanzar con el modelo, hacia la estimacion de medias predichas por el mismo

{emmeans, multcomp}

```
em_phom_LM <- emmeans(mod_phom_LM, ~ trt, type="response")
em_phom_LM
# comparaciones multiples
res_phom_LM <- cld(em_phom_LM, Letters = letters, alpha = .05, type = "response")

knitr::kable(res_phom_LM)

plot_model(mod_phom_LM, type='pred', show.data=T)
```

Interpretacion de coeficientes:

Ahora que tenemos los predichos de cada tratamiento podemos interpretar los coeficientes.

Recordemos que trt 1 es el nivel de referencia (orden arbitrario alfabético, se puede cambiar), y el resto de trat se suman a este para la estimacion de su media:

```
knitr::kable(res_phom_LM)
summary(mod_phom_LM)
c_t1 = 0.4366667
c_t2 = 0.4366667 +(-0.15667)
c_t2
c_t3 = 0.4366667 +(-0.29000)
c_t3
```

- GLM

(Anecdotico)

```
mod_phom_LM2 <- glmer(inc ~ trt + (1|bk),
                      family = gaussian("identity"),
                      data=phom_dat)
summary(mod_phom_LM2)
summary(mod_phom_LM)
```

Opción 1: variable original (éxitos y fracasos) agrupados

```
mod_phom_GLM1 <- glmer(
  cbind(y, n-y) ~ trt + (1|bk), # matriz de exitos y fracasos
  family="binomial",
  data=phom_dat)
summary(mod_phom_GLM1)
```

Opción 2: proporcion de exitos / total muestra (incidencia)

```
mod_phom_GLM2 <- glmer(inc ~ trt + (1|bk),
               family="binomial",
               weights = n, # pesos o tamaño de muestra
               data=phom_dat)
summary(mod_phom_GLM2)
```

```
tab_model(mod_phom_GLM1, mod_phom_GLM2)
```

- Diagnósticos

https://stats.stackexchange.com/questions/185491/diagnostics-for-generalized-linear-mixed-models-specifically-residuals

{DHARMa}

https://cran.r-project.org/web/packages/DHARMa/vignettes/DHARMa.html#goodness-of-fit-tests-on-the-scaled-residuals

```
testOutliers(mod_phom_GLM2)
testDispersion(mod_phom_GLM2)
```

Medias predichas por el modelo ajustado y comparaciones multiples

```
em_phom_GLM <- emmeans(mod_phom_GLM2, ~ trt, type="response")
res_phom_GLM <- cld(em_phom_GLM, Letters = letters, alpha = .05, type = "response")
knitr::kable(res_phom_GLM)
```

- Interpretacion de coeficientes

trt 1 es el nivel de referencia, y el resto de trat se suman a este para la estimacion de su media:

Pero ahora en escala de log ODDS

```
summary(mod_phom_GLM2)
knitr::kable(res_phom_GLM)

LO_t1 = -0.2585
p_t1 = 0.4357251

odds_t1 = 0.4357251 / (1-0.4357251)
odds_t1

log(odds_t1)
LO_t1
# volver a slide 15
```

Resto de tratamientos

```
p_t2 = 0.2763751
odds_t2 = 0.2763751 / (1-0.2763751)

OR_t2_t1 = odds_t2/odds_t1

OR_t2_t1 #  0.4946108

log(OR_t2_t1) # chequear con summary
summary(mod_phom_GLM2)
log(OR_t2_t1) # chequear con summary

# similar al LM?
log(odds_t2) - log(odds_t1)
```

```r
# t2 directamente del summary
exp(-0.7040)
OR_t2_t1
tab_model(mod_phom_GLM2)
(1-0.4946029) * 100
# 50.53971
```

> La chance de un foliolo de frutilla presentar sintoma de phomopsis disminuye un 50% cuando se aplica el tratamiento 2 respecto al control sin tratar

```r
knitr::kable(res_phom_LM)
knitr::kable(res_phom_GLM)
```

• Los errores estándar estimados (SE) son todos incorrectos (por definición), deben ser funciones de la media para datos binomiales

• Los SE incorrectos darán pruebas incorrectas de significación para los efectos del tratamiento y conducirán a conclusiones incorrectas

3.3 Reg. Logistica

- Data maracuya:
- geno: genotipos de maracuyá (Passiflora edulis) (A y B)
- bk: bloque (area homogenea dentro del campo que incluye hileras de genotipo A y B) - Efecto aleatorio
- days: dias desde la inoculacion (DDI) con el virus CABMV (Cowpea aphid-borne mosaic virus)
- n_plants: nro de plantas evaluadas dentro de cada parcela
- dis_plants: plantas con sintomas del CABMV
- y = inc_prop (dis_plants/n_plants)
- plot: unidad experimental (parcelas=bloque:geno)

```r
raw <- rio::import("https://raw.githubusercontent.com/juanchiem/glm_webinar/main/data/mara
# raw <- rio::import("data/maracuya.csv") %>% tibble

dat <- raw %>%
```

```
  mutate_at(vars(geno, bk), as.factor) %>%
  mutate(inc_prop=dis_plants/n_plants,
         plot = interaction(bk,geno))  # %>%

dat %>%
    ggplot() +
    aes(x=days, y=inc_prop, col=geno, shape=bk)+
    geom_point()+
    geom_line(aes(group=interaction(bk,geno)))
```

Filtramos el dataset completo para subsets menores

```
# solo una evaluación a los 60 dias
dat60 <- dat %>%
  filter(days %in% c(60))

# solo una evaluación a los 90 dias
dat90 <- dat %>%
  filter(days %in% c(90))

# Dos evaluaciones: a los 60 y 90 dias
dat60_90 <- dat %>%
  filter(days %in% c(60, 90)) # %>%
  # mutate_at(vars(days), as.factor)
```

3.3.1 Single-point assessment

- 60 d

```
dat60
dat60 %>%
  ggplot() +
  aes(x=geno, y=inc_prop) +
  geom_jitter(alpha=.5, width=.02)
```

```
# mod1 <- glmer(
#   cbind(dis_plants, n_plants-dis_plants) ~ geno + (1|bk),
#   family="binomial",
#   data=dat60)
```

```
mod1 <- glmer(
  inc_prop ~ geno + (1|bk), # bloque como efecto aleatorio
  weights=n_plants,
  family="binomial",
  data=dat60)

car::Anova(mod1)
summary(mod1)
```

```
tab_model(mod1)
plot_model(mod1, type='pred', show.data=T,   bpe.color ="red")
```

Otro gran aliado es el paquete “emmeans” quien nos devuelve las estimaciones en proporcion ahorrandonos muchos calculos manuales

```
em1 <- emmeans(mod1, ~ geno, type="response")
res1 <- cld(em1, Letters = letters, alpha = .05, type = "response")
knitr::kable(res1)
```

Interpretacion de coef y medidas de efecto

```
# lo que nos da el emmeans
p_A = 0.1066667
p_B =  0.0933333

odds_A = p_A/(1-p_A)
odds_B = p_B/(1-p_B)

# lo que nos da el tab_model
OR_B_A = odds_B/odds_A
OR_B_A

# lo que nos da el summary
log_OR_B_A = log(OR_B_A)
log_OR_B_A
summary(mod1)
```

- 90 d

```r
dat90

dat90 %>%
  ggplot() +
  aes(x=geno, y=inc_prop) +
  geom_point()

# mod1 <- glmer(
#   cbind(dis_plants, n_plants-dis_plants) ~ geno + (1|bk),
#   family="binomial",
#   data=dat60)
```

```r
mod2 <- glmer(
  inc_prop ~ geno + (1|bk),
  weights=n_plants,
  family="binomial",
  data=dat90)

# boundary (singular) fit: see help('isSingular') puede deberse al bajo numero de bk
```

```r
car::Anova(mod2)
summary(mod2)
```

Vemos que ahora si, el geno tiene efecto significativo sobre la incidencia de la enfermedad

```r
tab_model(mod2)
```

podemos decir que la chance de presentar la enfermedad del genotipo B es 71% (1 - 0.29 = 0.71 * 100) menor en relacion al geno A

```r
plot_model(mod2, type='pred', show.data=T)
```

```r
em2 <- emmeans(mod2, ~ geno, type="response")
res2 <- cld(em2, Letters = letters, alpha = .05, type = "response")
knitr::kable(res2)
```

3.3.2 Multiple-point assessment

Incluyendo una interaccion

- 60 y 90 d

```
dat60_90
```

```
dat60_90 %>%
  ggplot() +
  aes(x=days, y=inc_prop, col=geno, shape=bk)+
  geom_point()
```

```
# debido a las mediciones repetidas en el tiempo agregamos efecto aleatorio sobre la parce

mod3 <- glmer(inc_prop ~ geno * days +
                (1|bk) + (1|bk:geno),
              weights=n_plants,
              family="binomial",
              data=dat60_90)

car::Anova(mod3)
```

(anecdotico: days como factor con 2 niveles)

```
car::Anova(glmer(inc_prop ~ geno * factor(days) +
                (1|bk) + (1|bk:geno),
              weights=n_plants,
              family="binomial",
              data=dat60_90))
```

Removiendo la interaccion, dejando como efectos simples geno y dias

```
mod3.1 <- glmer(inc_prop ~ geno + days +
                  (1|bk) + (1|bk:geno),
              weights=n_plants,
              family="binomial",
              data=dat60_90)
```

```
anova(mod3, mod3.1, test = "Chisq")
AIC(mod3, mod3.1)
```

El modelo conteniendo la interaccion (geno * days) es mejor (p=0.0231, AIC=58.34804)

(anecdótico: asignacion de parcela explicitamente)

```
mod3_ <- glmer(inc_prop ~ geno * days +
                 (1|bk) + (1|plot),
               weights=n_plants,
               family="binomial",
               data=dat60_90)
AIC(mod3, mod3_)
```

```
summary(mod3)
```

log odds A = -5.33987 + 0.05358 days log odds B = (-5.33987 + 2.04105) + (0.0535-0.036) days

```
tab_model(mod3)
```

days = 1.06 » por cada dia acumulado desde la inoculacion el geno A tiene una chance de aumentar 1.06 veces la probabilidad de aparicion de sintomas (aumento de la incidencia) y es significativo (IC: 1.03–1.08, p= <0.001)

geno [B] * days = 0.96 »> la chance de aumentar la incidencia por cada dia desde la inoculacion en el geno B es 4% menor respecto al geno A (IC: 0.93 – 0.99, p=0.023)

```
plot_model(mod3,
           terms = c("days", "geno"),
           type='pred', show.data=T)

em3 <- emmeans(mod3, ~ geno|days, type="response")

res3 <- cld(em3, Letters = letters, alpha = .05, type = "response")
knitr::kable(res3)
```

confirmamos lo visto anteriormente: - a los 60 dias no hubo diferencias en la incidencia del virus, pero si a los 90 ddi

3.3.3 Serie full

```
head(dat)
dat %>%
  ggplot() +
  aes(x=days, y=inc_prop, col=geno, shape=bk)+
  geom_point()
```

```
# debido a las mediciones repetidas en el tiempo agregamos efecto aleatorio sobre la parce

mod_serie <- glmer(inc_prop ~ geno * days +
                     (1|bk) + (1|bk:geno),
                   weights=n_plants,
                   family="binomial",
                   data=dat)
```

```
# Sacamos el efecto del genotipo
mod_serie0 <- glmer(inc_prop ~ days +
    (1|bk) + (1|bk:geno),
               weights=n_plants,
               family="binomial",
               data=dat)

mod_serie1 <- glmer(inc_prop ~ days + geno +
    (1|bk) + (1|bk:geno),
               weights=n_plants,
               family="binomial",
               data=dat)
```

- Seleccion de modelo

```
anova(mod_serie0, mod_serie1, mod_serie, test = "Chisq")
AIC(mod_serie0, mod_serie1, mod_serie)
```

mod_serie: df=6 y AIC=299.4276

Diagnósticos

```
testOutliers(mod_serie)
testDispersion(mod_serie)
```

```
summary(mod_serie)

tab_model(mod_serie)
plot_model(mod_serie,
           terms = c("days", "geno"),
           type='pred', show.data=T)
```

Pred. lineal geno A = -3.18 + 0.024 * days (dias significativo para el geno A, ya que conforme transcurren los dias la incidencia aumenta)

Pred. lineal geno B = (-3.275 + 0.147) + (0.025 + 0.99) * days

Interaccion significativa: las curvas son diferentes, deben ajustarse una por genotipo

- Predicción

{ggeffects}

Curva completa

```
ggpredict(mod_serie, c( "days", "geno"))
```

Genotipo A - intervalo 100 a 110 ddi

```
ggpredict(mod_serie,
          terms = "days [100:110]",
          condition = c(geno = c("A")))
```

Ambos genotipos para el DDI=100

```
ggpredict(mod_serie,
          terms = "geno",
          condition = c(days = "100"))
```

Referencias

Madden, L., Turechek, W., and Nita, M. 2002. Evaluation of generalized linear mixed models for analyzing disease incidence data obtained in designed experiments. Plant Disease. 86:316–325.