

prueba_ecocbo_2

```
library(devtools)
library(ggplot2)
library(tibble)
library(dplyr)
library(tidyr)
load_all()
```

Single factor

Pretratamiento de los datos

```
DataPrep <- prep_data(epiDat,
                      type = "counts",
                      Sest.method = "average",
                      cases = 5,
                      N = 100,
                      M = 3,
                      n = 18,
                      m = 3,
                      k = 50,
                      transformation = "square root",
                      method = "bray",
                      dummy = FALSE,
                      useParallel = TRUE,
                      model = "single.factor")
```

| | 0%

```
Warning: ignoring unrecognized snow option(s): progress
```

```
head(DataPrep$Results)
```

```
dat.sim k n pseudoFH0 pseudoFHa      MSR
[1,]     1 1 2 1.0390855 1.7075969 0.2452660
[2,]     1 2 2 1.2867822 1.9374069 0.2514253
[3,]     1 3 2 0.5453476 1.3536560 0.2640905
[4,]     1 4 2 0.7240018 0.9379929 0.3172073
[5,]     1 5 2 0.9068334 2.0581435 0.2232367
[6,]     1 6 2 1.4401572 0.9356663 0.3301834
```

Componentes de variación

```
CompVar <- scompvar(DataPrep)
CompVar
```

```
Source Est.var.comp
1 Residual    0.2879503
```

```
# Este valor debe ser igual al máximo de los promedios de cada simulación
CompVarCalculado <- DataPrep$Results |>
  as.data.frame() |>
  group_by(dat.sim) |>
  summarise(MSR = mean(MSR)) |>
  summarise(CompVarCalculado = max(MSR))
CompVarCalculado
```

```
# A tibble: 1 x 1
  CompVarCalculado
  #> <dbl>
  1        0.286
```

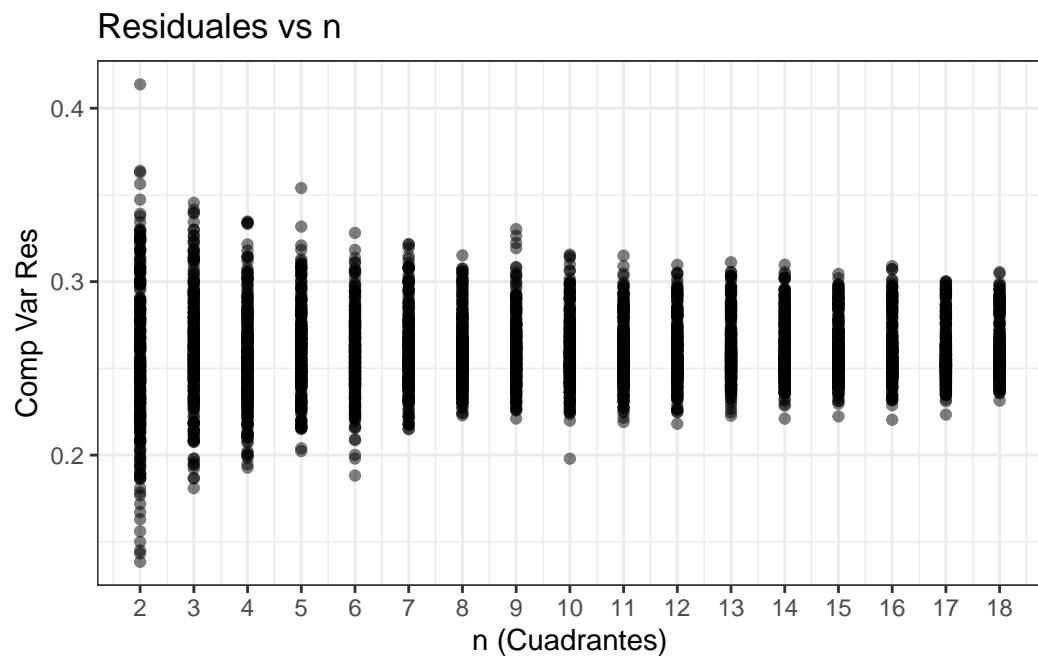
Comportamiento de cuadrados medios

Se puede observar como la estimación de componente de variación de los residuales se va haciendo más específica conforme aumenta el número de muestras.

```

ggplot(DataPrep$Results, aes(x = n, y = MSR)) +
  geom_point(alpha = 0.5) +
  scale_x_continuous(breaks = seq(2, 18, 1)) +
  labs(
    title = "Residuales vs n",
    x = "n (Cuadrantes)",
    y = "Comp Var Res",
    color = "m (Sitios)"
  ) +
  theme_bw()

```



Nested symmetric

Pretratamiento de los datos

```

DataPrep <- prep_data(SSP::epibionts,
                        type = "counts",
                        Sest.method = "average",
                        cases = 5,
                        N = 100,
                        M = 30,

```

```
n = 18,  
m = 15,  
k = 20,  
transformation = "square root",  
method = "bray",  
dummy = FALSE,  
useParallel = TRUE,  
model = "nested.symmetric")
```

```
| | 0%  
| |  
| | 1%  
| |= 1%  
| |  
| |= 2%  
| |  
| == 2%  
| |  
| == 3%  
| |  
| == 4%  
| |  
| === 4%  
| |  
| === 5%  
| |  
| ===== 5%  
| |  
| ===== 6%  
| |  
| ===== 6%  
| |  
| ===== 7%  
| |  
| ===== 8%  
| |  
| ===== 8%  
| |  
| ===== 9%
```

=====	9%
=====	10%
=====	11%
=====	11%
=====	12%
=====	12%
=====	13%
=====	14%
=====	14%
=====	15%
=====	15%
=====	16%
=====	16%
=====	17%
=====	18%
=====	18%
=====	19%
=====	19%
=====	20%
=====	21%
=====	21%

=====	22%
=====	22%
=====	23%
=====	24%
=====	24%
=====	25%
=====	25%
=====	26%
=====	26%
=====	27%
=====	28%
=====	28%
=====	29%
=====	29%
=====	30%
=====	31%
=====	31%
=====	32%
=====	32%
=====	33%
=====	34%
=====	34%

===== 	35%
===== 	35%
===== 	36%
===== 	36%
===== 	37%
===== 	38%
===== 	38%
===== 	39%
===== 	39%
===== 	40%
===== 	41%
===== 	41%
===== 	42%
===== 	42%
===== 	43%
===== 	44%
===== 	44%
===== 	45%
===== 	45%
===== 	46%
===== 	46%

===== 	47%
===== 	48%
===== 	48%
===== 	49%
===== 	49%
===== 	50%
===== 	51%
===== 	51%
===== 	52%
===== 	52%
===== 	53%
===== 	54%
===== 	54%
===== 	55%
===== 	55%
===== 	56%
===== 	56%
===== 	57%
===== 	58%
===== 	58%
===== 	59%
===== 	59%

=====	60%
=====	61%
=====	61%
=====	62%
=====	62%
=====	63%
=====	64%
=====	64%
=====	65%
=====	65%
=====	66%
=====	66%
=====	67%
=====	68%
=====	68%
=====	69%
=====	69%
=====	70%
=====	71%
=====	71%
=====	72%

===== 	72%
===== 	73%
===== 	74%
===== 	74%
===== 	75%
===== 	75%
===== 	76%
===== 	76%
===== 	77%
===== 	78%
===== 	78%
===== 	79%
===== 	79%
===== 	80%
===== 	81%
===== 	81%
===== 	82%
===== 	82%
===== 	83%
===== 	84%
===== 	84%
===== 	85%

=====	85%
=====	86%
=====	86%
=====	87%
=====	88%
=====	88%
=====	89%
=====	89%
=====	90%
=====	91%
=====	91%
=====	92%
=====	92%
=====	93%
=====	94%
=====	94%
=====	95%
=====	95%
=====	96%
=====	96%
=====	97%

```

|=====
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|===== | 98%
|
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|===== | 98%
|
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|===== | 99%
|
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|===== | 99%
|
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|=====
|===== | 100%

```

```
head(DataPrep$Results)
```

	dat.sim	k	m	n	pseudoFHO	pseudoFHa	MSB(A)	MSR
[1,]	1	1	2	2	18.00874	17.88066	0.017787514	0.01793359
[2,]	1	2	2	2	40.63247	29.30312	0.014780596	0.01818012
[3,]	1	3	2	2	34.61613	59.14762	0.009825905	0.01772639
[4,]	1	4	2	2	31.60036	41.42452	0.012126045	0.01529915
[5,]	1	5	2	2	20.99560	12.47240	0.018846358	0.01964909
[6,]	1	6	2	2	32.33960	30.50729	0.013572206	0.01909605

Componentes de variación

```
CompVar <- scompvar(DataPrep)
CompVar
```

Source	Est.var.comp
1 B(A)	0.04899581
2 Residual	0.12309687

```
# Este valor debe ser igual al máximo de los promedios de cada simulación
scompvarCalc <- tribble(~name, ~CompVar,
                         "B(A)", NA,
                         "Res", NA)

CompVarResCalculado <- DataPrep$Results |>
  as.data.frame() |>
  filter(m == 15 & n == 18) |>
  group_by(dat.sim) |>
  summarise(MSR = mean(MSR)) |>
```

```

summarise(CompVarCalculado = max(MSR))

CompVarBACalculado <- DataPrep$Results |>
  as.data.frame() |>
  filter(m == 15 & n == 18) |>
  group_by(dat.sim) |>
  summarise(MSBA = mean(`MSB(A)`)) |>
  summarise(CompVarCalculado = max(MSBA))
CompVarBACalculado <- (CompVarBACalculado - CompVarResCalculado) / 18

scompvarCalc[1,2] <- CompVarBACalculado
scompvarCalc[2,2] <- CompVarResCalculado

scompvarCalc

# A tibble: 2 x 2
  name   CompVar
  <chr>   <dbl>
1 B(A)    0.0490
2 Res     0.123

```

Comportamiento de cuadrados medios

En los cálculos correspondientes a `scompvar` y demás, la función tomará por default los valores máximos de `n` y `m` para aprovechar el mayor número de grados de libertad posibles. En estas gráficas se incluyen los comportamientos para todos los valores calculados de `n` y `m` para poder observar la operación de las función `prep_data`.

En las primeras dos gráficas se observa el *funnel* para el cálculo de componente de variación residual con respecto al número de muestras `n`, y un comportamiento homogéneo para el cálculo de cuadrados medios anidados $B(A)$ con respecto al número de muestras `n`.

```

# se añade columna de componentes de variación para componente anidado
Resultados <- as.data.frame(DataPrep$Results) |>
  mutate(CompVarBA = (`MSB(A)` - MSR) / n)

p1 <- ggplot(Resultados, aes(x = n, y = MSR, color = factor(m))) +
  geom_point() +
  # geom_jitter(height=0) +
  scale_x_continuous(breaks = seq(2, 18, 1)) +
  # scale_y_continuous(breaks = seq(0.0, 1, 0.2)) +

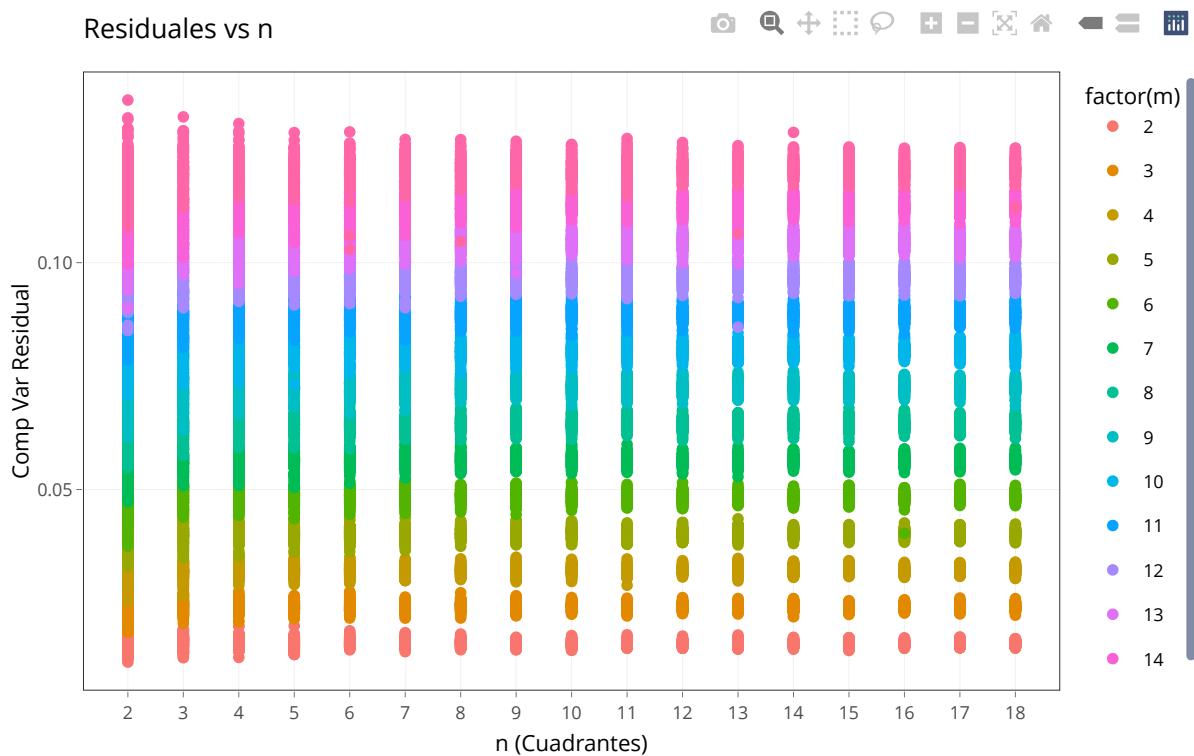
```

```

  labs(
    title = "Residuales vs n",
    x = "n (Cuadrantes)",
    y = "Comp Var Residual",
  ) +
  theme_bw()

p1 |>
  plotly::ggplotly()

```

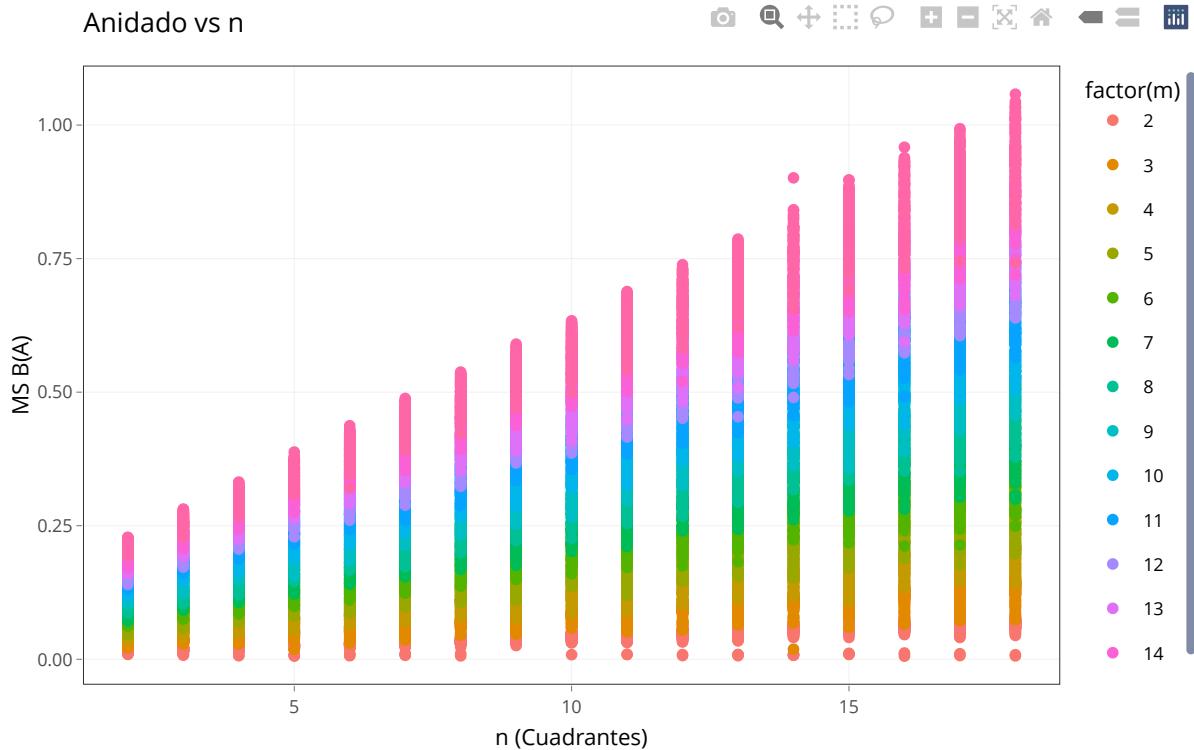


```

p2 <- ggplot(Resultados, aes(x = n, y = `MSB(A)`), color = factor(m))+
  geom_point()+
  # scale_y_continuous(breaks = seq(0.0, 1, 0.2)) +
  labs(
    title = "Anidado vs n",
    x = "n (Cuadrantes)",
    y = "MS B(A)",
  ) +
  theme_bw()

```

```
p2 |>
  plotly::ggplotly()
```

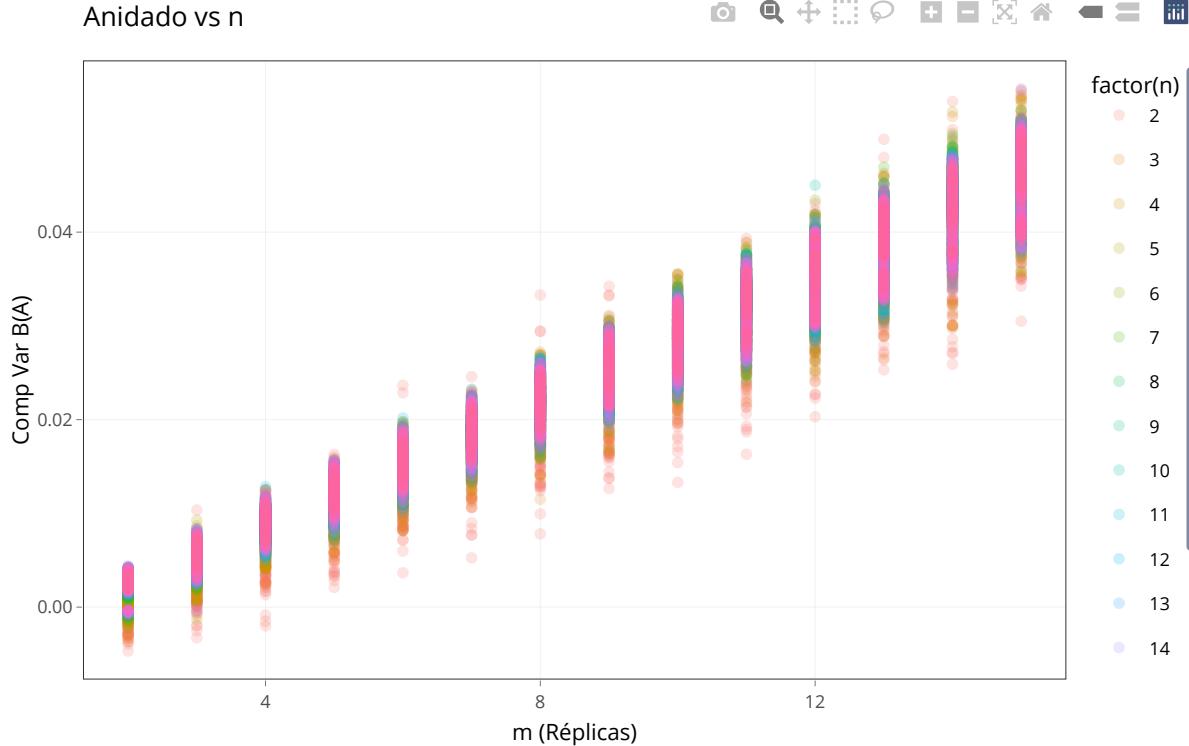


A continuación se observan los gráficos para el componente de variación anidado $B(A)$ que tiene forma de *funnel* al igual que el componente de variación residual.

Esto se debe a que el cálculo del componente de variación anidado incluye una división con un denominador que se va haciendo progresivamente mayor, por lo que el resultado tendrá que ser más pequeño cada vez. La última gráfica presenta esta matemática de manera visual, se resta $MS_{B(A)} - MS_R$ y el resultado se divide por un n cada vez mayor, por lo que el componente de variación se va concentrando en valores cada vez más cercanos a cero.

```
p3 <- ggplot(Resultados, aes(x = m, y = CompVarBA, color = factor(n)))+
  geom_point(alpha = 0.2) +
  labs(
    title = "Anidado vs n",
    x = "m (Réplicas)",
    y = "Comp Var B(A)"
  ) +
  theme_bw()
```

```
plotly::ggplotly(p3)
```



```
Resultados |>
```

```
pivot_longer(cols=c("CompVarBA", `MSB(A)` , "MSR"),
             names_to="CompVar",
             values_to="Value") |>
  ggplot(aes(x = n, y = Value, color = CompVar))+
  geom_point(alpha = 0.5)+  
  scale_color_manual(values = c("orange", "chocolate", "aquamarine"))+
  labs(title = "Anidado vs n",
       x = "n (Cuadrantes)",
       y = "Componentes",
       color = "Componentes")+
  theme_bw()
```

Anidado vs n

