

Exploración de la heterogeneidad

Tamara Ricardo

Análisis de moderadores

El análisis de moderadores permite explorar fuentes de heterogeneidad entre los estudios incluidos en un meta-análisis. Se puede realizar mediante la inclusión de una variable independiente categórica (*análisis de subgrupos*) o numérica (*metarregresión*).

Este procedimiento contribuye a evaluar hipótesis sobre variaciones en la magnitud del efecto entre estudios y a interpretar diferencias observadas en los resultados. Sin embargo, para evitar sesgos de selección, las variables utilizadas como moderadores deben definirse de antemano, durante la extracción de datos relevantes para la revisión sistemática.

El análisis de moderadores consta de dos etapas:

- Estimación del efecto **dentro de cada subgrupo**.
- Prueba estadística para evaluar diferencias **entre subgrupos**.

Análisis de subgrupos

Para evaluar el efecto de una variable categórica, se puede utilizar el argumento `subgroup` al ajustar un modelo de meta-análisis. A continuación, realizaremos un análisis de subgrupos sobre el modelo de prevalencia ajustado previamente, considerando el país (`country`) como moderador.

Comenzaremos cargando los paquetes necesarios:

```
# Carga de paquetes
library(meta)      # Modelos de meta-análisis
library(scico)     # Paletas de colores accesibles
library(janitor)   # Tablas de frecuencia
library(tidyverse) # Manejo de datos

# Paleta colorblind-friendly
pal <- scico(n = 4, palette = "buda")
```

Cargamos los datos y exploramos su estructura:

```
# Cargar datos
datos_prev <- dat.crisafulli2020

# Inspeccionar estructura de los datos
glimpse(datos_prev)
```

```

Rows: 26
Columns: 7
$ study   <chr> "Brooks (1977)", "Danieli (1977)", "Takeshita (1977)", "Drummo...
$ pubyear <int> 1977, 1977, 1977, 1979, 1980, 1980, 1981, 1982, 1983, 1983, 19...
$ country <fct> UK, IT, JP, NZ, AU, IT, IT, CA, FR, IT, DE, IT, JP, CA, NO, IT...
$ from    <int> 1953, 1952, 1956, NA, 1960, 1952, 1955, 1950, 1978, 1969, 1977...
$ to      <int> 1968, 1972, 1970, NA, 1971, 1972, 1974, 1979, 1978, 1980, 1984...
$ cases   <int> 47, 66, 19, 2, 99, 105, 73, 110, 12, 156, 48, 76, 50, 5, 16, 2...
$ total   <int> 177413, 234396, 91157, 10000, 532302, 371698, 301283, 420374, ...

```

Usaremos la función `tabyl()` del paquete `janitor` [1] para generar una tabla de frecuencias de los niveles de la variable `country`:

```

tabyl(datos_prev, country) |> # Genera tabla de frecuencia
  arrange(-n) |>              # Ordena por frecuencia
  adorn_pct_formatting()      # Proporciones a porcentajes

```

country	n	percent
IT	6	23.1%
CA	3	11.5%
UK	3	11.5%
JP	2	7.7%
AU	1	3.8%
BE	1	3.8%
CY	1	3.8%
DE	1	3.8%
DK	1	3.8%
EE	1	3.8%
FR	1	3.8%
NL	1	3.8%
NO	1	3.8%
NZ	1	3.8%
SI	1	3.8%
US	1	3.8%

Dado que la mayoría de los estudios provienen de Italia y la frecuencia en otros países es baja, creamos una variable dicotómica `pais_cat`:

```

datos_prev <- datos_prev |>
  mutate(pais_cat = if_else(
    country == "IT", # Condición
    "Italia",        # Valor si la condición se cumple
    "Otro/s" )       # Valor si la condición no se cumple
  )

```

Ajustamos el modelo de meta-análisis para proporciones, incorporando la variable `pais_cat` como moderador:

```
# Ajuste modelo
mod_sg <- metaprop(
  event = cases,          # Casos observados
  n = total,              # Tamaño de la muestra
  studlab = study,        # Identificador del estudio
  data = datos_prev,      # Conjunto de datos
  sm = "PLOGIT",          # Transformación logit
  common = FALSE,         # Omitir modelo de efectos fijos
  random = TRUE,          # Modelo de efectos aleatorios
  pscale = 100000,        # Escala a casos/100 000 habitantes
  subgroup = pais_cat     # Moderador categórico
)

# Resumen del modelo ajustado
mod_sg
```

```
Number of studies: k = 26
Number of observations: o = 6831388
Number of events: e = 1545

              events          95%-CI
Random effects model 22.2342 [20.6056; 23.9915]

Quantifying heterogeneity (with 95%-CIs):
tau^2 = 0.0126; tau = 0.1121; I^2 = 33.2% [0.0%; 58.6%]; H = 1.22 [1.00; 1.55]

Test of heterogeneity:
      Q d.f. p-value
Wald 37.41  25  0.0527
LRT  39.01  25  0.0368

Results for subgroups (random effects model):
      k  events          95%-CI tau^2   tau    Q  I^2
pais_cat = Otro/s  20 20.9400 [19.0685; 22.9951] 0.0122 0.1104 26.02 27.0%
pais_cat = Italia   6 25.0380 [22.5505; 27.7999] 0.0033 0.0577  5.97 16.2%

Test for subgroup differences (random effects model):
      Q d.f. p-value
Between groups 6.22   1  0.0126

Details of meta-analysis methods:
- Random intercept logistic regression model
- Maximum-likelihood estimator for tau^2
- Calculation of I^2 based on Q
```

- Logit transformation
- Events per 100000 observations

La salida del modelo incluye dos secciones adicionales:

- **Results for subgroups (random effects model):** muestra los resultados para cada categoría del moderador, incluyendo:
 - Identificador de las categorías del moderador.
 - k: número de estudios en la categoría/subgrupo.
 - events: prevalencia en la categoría/subgrupo.
 - 95%-CI: intervalo de confianza al 95% para la prevalencia en la categoría/subgrupo.
 - tau²: varianza dentro de la categoría/subgrupo.
 - tau: desvío estándar en la categoría/subgrupo.
 - Q: estadístico Q de Cochran para la categoría/subgrupo.
 - I²: porcentaje de heterogeneidad observado en la categoría/subgrupo.
- **Test for subgroup differences (random effects model):** prueba de hipótesis para detectar diferencias significativas entre subgrupos.

En este ejemplo, observamos una prevalencia significativamente mayor en Italia en comparación con otros países ($p = 0,013$).

Podemos visualizar los resultados del análisis de moderadores incorporando los siguiente argumentos a la función `forest()`:

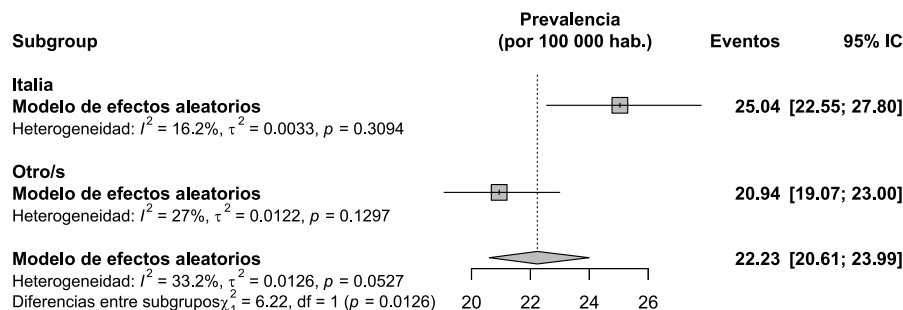
- `layout = "subgroup"`: muestra los estimadores de efecto para cada subgrupo y el estimador global, omitiendo los resultados de los estudios individuales.
- `sort.subgroup`: ordena alfabeticamente las categorías de la variable moderadora.
- `calwidth.subgroup`: ajusta el ancho del forest plot para que se muestren correctamente las etiquetas de las categorías/subgrupos.
- `calwidth.tests`: ajusta el ancho del forest plot para que se muestren correctamente las etiquetas del test de hipótesis de diferencias en las categorías/subgrupos.
- `print.subgroup.name`: muestra la etiqueta de la variable moderadora delante de cada categoría (TRUE, por defecto) o lo oculta (FALSE).
- `label.test.subgroup.common`: etiqueta para los resultados del test para diferencias entre subgrupos en el modelo de efectos fijos.
- `label.test.subgroup.random`: etiqueta para los resultados del test para diferencias entre subgrupos en el modelo de efectos aleatorios.

En nuestro ejemplo:

```

forest(
  mod_sg,
  layout = "subgroup",
  sort.subgroup = TRUE,
  calcwidth.subgroup = TRUE,
  calcwidth.tests = TRUE,
  print.subgroup.name = FALSE,
  smlab = "Prevalencia \n (por 100 000 hab.)",
  rightlabs = c("Eventos", "95% IC"),
  hetlab = "Heterogeneidad: ",
  text.random = "Modelo de efectos aleatorios",
  label.test.subgroup.random = "Diferencias entre subgrupos"
)

```



A diferencia de modelos sin moderadores, `col.diamond` modifica el color del estimador global y de los estimadores para cada subgrupo. En cambio, `col.square` no tiene efecto sobre los colores del forest plot.

Metarregresión

Para evaluar el efecto de una variable continua, como el año de publicación (`pubyear`), primero ajustamos el modelo de meta-análisis:

```

# Ajuste modelo
mod_prev <- metaprop(
  event = cases,          # Casos observados
  n = total,              # Tamaño de la muestra
  studlab = study,        # Identificador del estudio
  data = datos_prev,      # Conjunto de datos
  sm = "PLOGIT",          # Transformación logit
  common = FALSE,         # Omitir modelo de efectos fijos
  random = TRUE,          # Modelo de efectos aleatorios
  pscale = 100000,        # Escala a casos/100 000 habitantes
)

```

Al modelo anterior le aplicamos la función `metareg()`, incluyendo `pubyear` como moderador en el argumento `formula`:

```
# Ajustar modelo de metarregresión
mod_year <- metareg(mod_prev,
                    formula = ~ pubyear)

# Resumen del modelo ajustado
summary(mod_year)
```

Mixed-Effects Model (k = 26; tau^2 estimator: ML)

logLik	deviance	AIC	BIC	AICc
-75.9315	14.5514	157.8629	161.6372	158.9538

tau^2 (estimated amount of residual heterogeneity): 0.0067
 tau (square root of estimated tau^2 value): 0.0816
 I^2 (residual heterogeneity / unaccounted variability): 26.61%
 H^2 (unaccounted variability / sampling variability): 1.36

Tests for Residual Heterogeneity:

Wld(df = 24) = 28.8391, p-val = 0.2262
 LRT(df = 24) = 30.5162, p-val = 0.1682

Test of Moderators (coefficient 2):

QM(df = 1) = 7.1208, p-val = 0.0076

Model Results:

	estimate	se	zval	pval	ci.lb	ci.ub	
intrcpt	8.5560	6.3550	1.3463	0.1782	-3.8996	21.0116	
pubyear	-0.0085	0.0032	-2.6685	0.0076	-0.0148	-0.0023	**

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

Los resultados muestran un efecto estadísticamente significativo del año de publicación sobre la prevalencia del evento ($p = 0,008$).

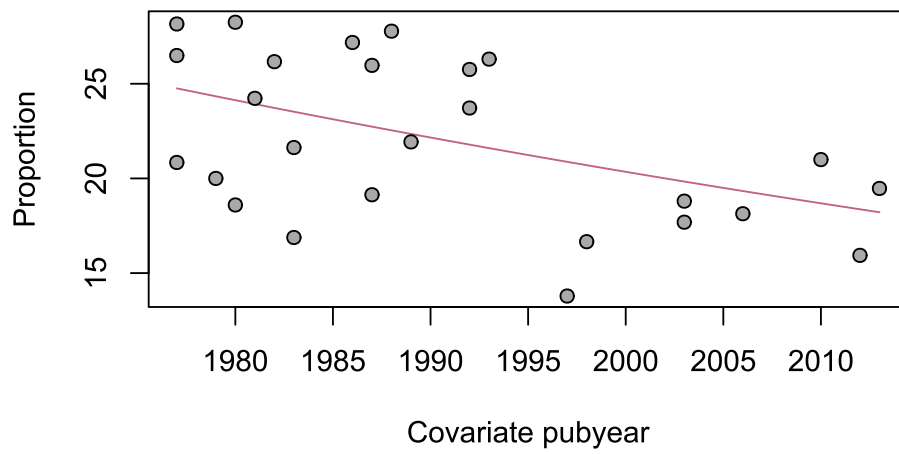
Los resultados de la metarregresión se visualizan usando *bubble plots*. Estos gráficos representan en el eje *X* el valor del moderador continuo y en el eje *Y* el efecto estimado para cada estudio. Cada burbuja representa un estudio individual, y su tamaño es proporcional al peso del estudio (generalmente inversamente proporcional a la varianza). Además, el gráfico incluye una línea de regresión con su intervalo de confianza del 95%, permitiendo visualizar la tendencia general.

La función `bubble()` genera este gráfico de forma rápida:

```

bubble(
  mod_year,
  cex = "common",      # Escala de los símbolos
  col.line = pal[2]     # Color de la línea de regresión
)

```

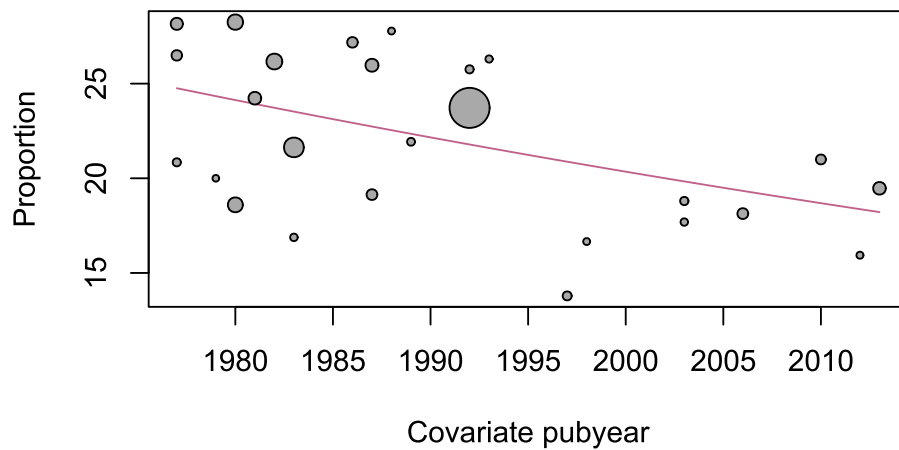


Para ajustar el tamaño de las burbujas según el peso de cada estudio, se puede modificar el argumento `cex` utilizando la función `rescale()` del paquete `plotrix`:

```

bubble(
  mod_year,
  cex = plotrix::rescale(
    x = 1 / mod_year$vi,
    newrange = c(0.5, 3)), # Ajusta la escala de los símbolos
  col.line = pal[2]
)

```



Visualización avanzada con paquete orchaRd

Este contenido es opcional y está dirigido a quienes tengan conocimientos más avanzados de R y deseen generar visualizaciones más atractivas y listas para la presentación en informes o artículos científicos.

El paquete `orchaRd` [2] permite crear visualizaciones con mayores opciones de personalización y compatibles con `ggplot2`. Sin embargo, dado que `orchaRd` no permite utilizar modelos ajustados por el paquete `meta`, es necesario reajustar el modelo utilizando `metafor`.

Cargamos los paquetes requeridos:

```
# Carga paquetes
library(orchaRd)
library(metafor)
```

Calculamos los estimadores de efecto individuales utilizando la función `escalc()`:

```
dat_forest <- escalc(
  measure = "PLO",      # estimador de efecto
  xi = cases,           # número de casos
  ni = total,           # tamaño muestral
  data = datos_prev     # conjunto de datos
)
```

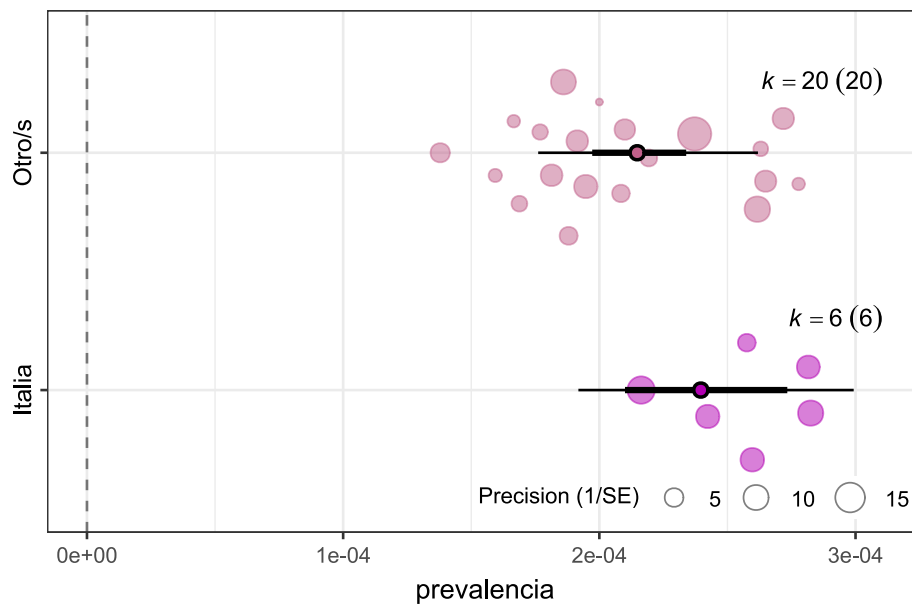
Ajustamos el modelo de meta-análisis con moderadores utilizando `rma.mv()`:


```
mod_forest <- rma.mv(
  yi = yi,                                # estimador de efecto
  V = vi,                                # error estándar
  mods = ~ factor(pais_cat) + pubyear,    # moderadores
  random = ~ 1|study,                    # modelo de efectos aleatorios
  data = dat_forest                      # conjunto de datos
)
```

Generamos el forest plot con la función `orchard_plot()`, definiendo los siguientes argumentos:

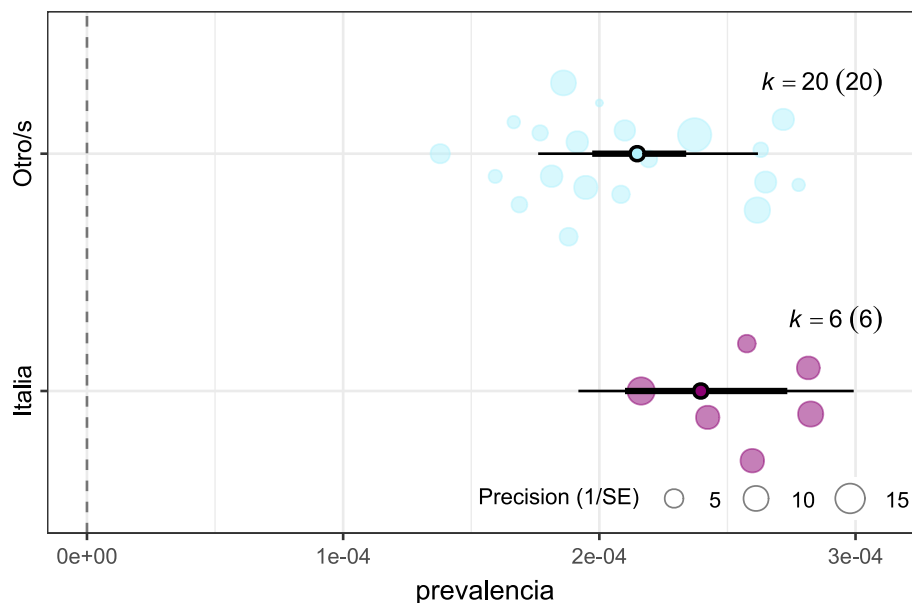
- `group`: identificador de estudio.
- `mod`: nombre del moderador (1 para estimador de efecto global).
- `xlab`: nombre del estimador de efecto.
- `transf`: mostrar los datos en escala original (opcional).

```
# Forest plot por subgrupos
orchard_plot(
  mod_forest,
  group = "study",
  mod = "pais_cat",
  xlab = "prevalencia",
  transf = "invlogit"
) +
# Paleta personalizada
scale_color_manual(values = pal) + # color de borde
scale_fill_manual(values = pal)    # color de relleno
```



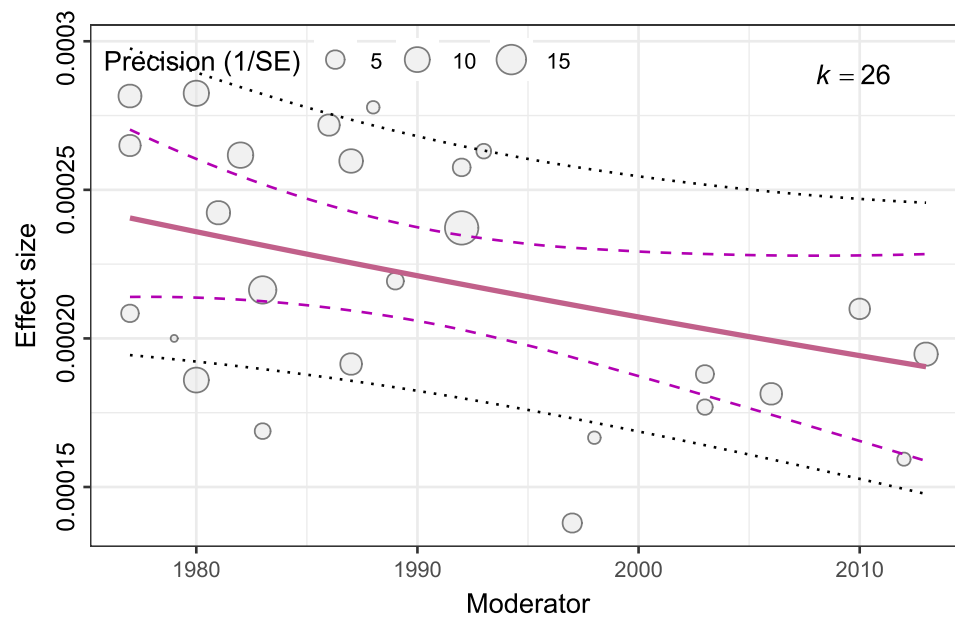
Con el argumento `scale_fill_scico_d()` podemos usar cualquiera de las paletas *colorblind-friendly* incluidas en *scico*:

```
# Forest plot por subgrupos
orchard_plot(
  mod_forest,
  group = "study",
  mod = "pais_cat",
  xlab = "prevalencia",
  transfm = "invlogit"
) +
# Paleta colorblind-friendly
scale_color_scico_d(palette = "hawaii") + # color de borde
scale_fill_scico_d(palette = "hawaii")    # color de relleno
```



Generamos el bubble plot para el moderador continuo (pubyear):

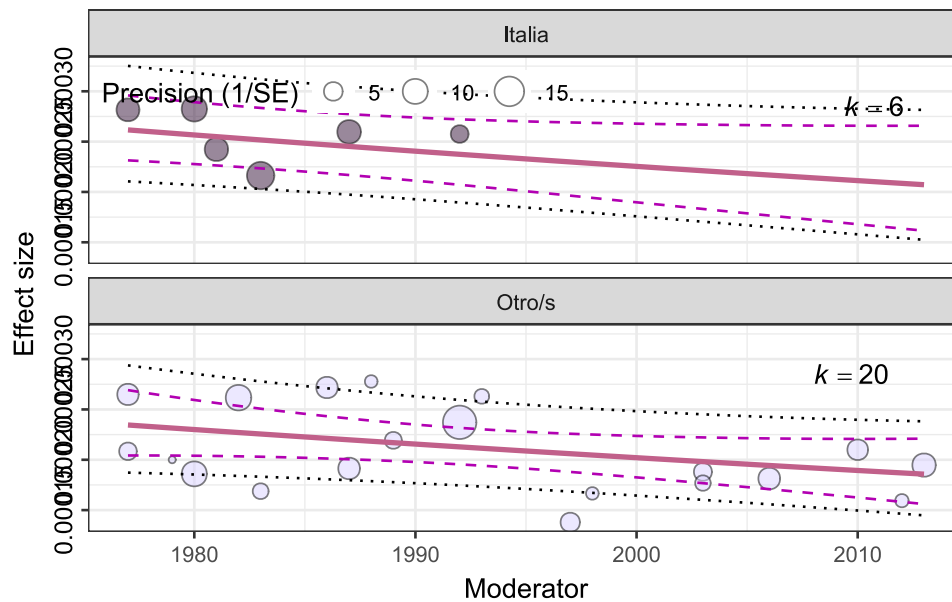
```
bubble_plot(
  mod_forest,          # modelo ajustado con metafor
  transfm = "invlogit", # muestra resultados como proporciones
  mod = "pubyear",      # moderador continuo
  group = "study",      # identificador de estudio
  est.col = pal[2],     # color de la línea de regresión
  ci.col = pal[1],      # color de las líneas del 95% IC
)
```



Si quisiera generar un bubble plot para cada nivel de `pais_cat`, puedo usar el argumento `by`:

```
bubble_plot(
  mod_forest,           # modelo ajustado con metafor
  transfm = "invlogit", # muestra resultados como proporciones
  mod = "pubyear",      # moderador continuo
  group = "study",      # identificador de estudio
  est.col = pal[2],     # color de la línea de regresión
  ci.col = pal[1],      # color de las líneas del 95% IC
  by = "pais_cat"
) +

# Paleta colorblind-friendly
scale_color_scico_d(palette = "glasgow") + # color de borde
scale_fill_scico_d(palette = "glasgow")    # color de relleno
```



Sesgo de publicación y análisis de sensibilidad

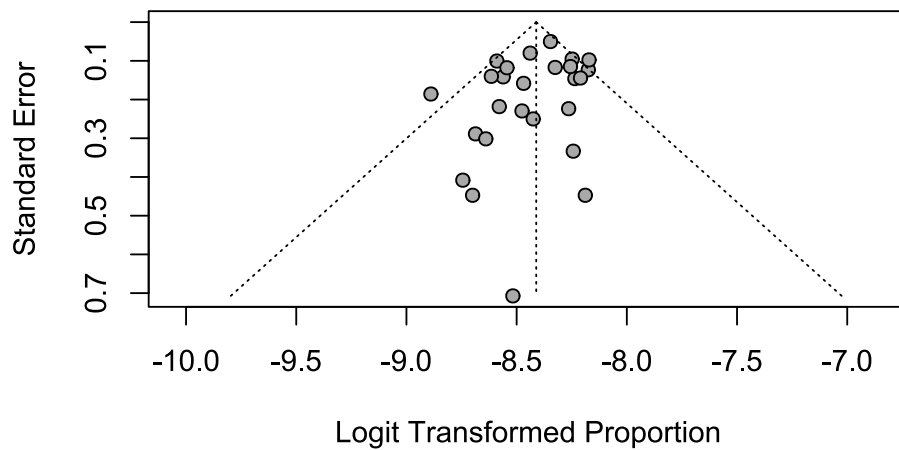
El sesgo de publicación (*publication bias*) se refiere a la tendencia de publicar con mayor frecuencia estudios con resultados positivos, estadísticamente significativos o con grandes tamaños del efecto. Esto puede distorsionar la síntesis de la evidencia en un meta-análisis, produciendo una sobreestimación del efecto global. Por ello, es esencial evaluar y ajustar el sesgo de publicación para garantizar que los resultados sean lo más precisos y representativos posible.

Funnel plot

El funnel plot o gráfico de embudo representa en el eje *Y* el tamaño muestral o la precisión de los estudios, y en el eje *X* el estimador de efecto para cada estudio. En ausencia de sesgo de publicación, se espera una distribución simétrica que forme un patrón similar a un embudo; de lo contrario, se observa asimetría.

Para generar un funnel plot, se utiliza la función `funnel()`:

```
funnel(mod_prev)
```



Test de Egger

El test de Egger evalúa la simetría del funnel plot mediante una regresión lineal entre el estimador de efecto y su error estándar. Un p -valor menor a 0,05 sugiere la presencia de sesgo de publicación.

Este test se implementa con la función `metabias()`:

```
metabias(
  mod_prev,          # modelo de meta-análisis
  method.bias = "Egger" # aplica test de Egger (opción por defecto)
)
```

Linear regression test of funnel plot asymmetry

Test result: $t = -1.45$, $df = 24$, $p\text{-value} = 0.1603$

Bias estimate: -0.6649 (SE = 0.4589)

Details:

- multiplicative residual heterogeneity variance ($\tau^2 = 1.4335$)
- predictor: standard error
- weight: inverse variance
- reference: Egger et al. (1997), BMJ

Test de Begg

El test de Begg utiliza la correlación de rangos para evaluar la relación entre el tamaño del efecto y el error estándar. Un p -valor menor que 0,05 indica la presencia de sesgo de publicación.

Se implementa añadiendo el argumento `method.bias = "Begg"` a la función `metabias()`:

```
metabias(
  mod_prev,          # modelo de meta-análisis
  method.bias = "Begg" # aplica test de Begg
)
```

Rank correlation test of funnel plot asymmetry

Test result: $z = -0.90$, $p\text{-value} = 0.3662$
 Bias estimate: -41.0000 (SE = 45.3689)

Reference: Begg & Mazumdar (1993), Biometrics

Trim-and-fill

El método *trim-and-fill* estima el número de estudios faltantes debido al sesgo de publicación y ajusta la media global en consecuencia, incorporando estudios hipotéticos. Esto ayuda a corregir la estimación del efecto global.

Se utiliza la función `trimfill()`:

```
trimfill(mod_prev)
```

Number of studies: $k = 32$ (with 6 added studies)

	events	95%-CI
Random effects model	23.4496	[21.7062; 25.3329]

Quantifying heterogeneity (with 95%-CIs):

$\tau^2 = 0.0181$; $\tau = 0.1344$; $I^2 = 42.4\%$ [12.2%; 62.2%]; $H = 1.32$ [1.07; 1.63]

Test of heterogeneity:

Q	d.f.	p-value
53.82	31	0.0067

Details of meta-analysis methods:

- Inverse variance method
- Maximum-likelihood estimator for τ^2
- Calculation of I^2 based on Q
- Trim-and-fill method to adjust for funnel plot asymmetry (L-estimator)
- Logit transformation
- Events per 100000 observations

Generalmente, se recomienda utilizar dos o más métodos para evaluar el sesgo de publicación y obtener una visión más completa de su impacto en el meta-análisis. En el ejemplo anterior, no hubiera sido necesario realizar el *trim-and-fill* ya que el funnel plot tiene una forma bastante simétrica y el p -valor de los test de Egger y Begg fue menor a 0,05.

Análisis de sensibilidad

El análisis de sensibilidad se utiliza para evaluar la robustez de los resultados del meta-análisis. Una estrategia común es reajustar el modelo excluyendo estudios con menor tamaño muestral o de menor calidad, y comparar el estimador de efecto y su 95% IC con el modelo original. Otra opción es realizar un análisis *leave-one-out*, en el que se remueve un estudio a la vez y se observa la variación del efecto global.

La función `metainf()` permite realizar el análisis de influencia (*leave-one-out*):

```
metainf(  
  mod_prev,          # modelo de meta-análisis  
  pooled = "random"  # modelo de efectos aleatorios  
)
```

Este análisis ayuda a determinar si algún estudio en particular influye de forma excesiva en los resultados del meta-análisis. Al igual que en los modelos anteriores, los resultados pueden visualizarse usando forest plots.

Note

Hasta aquí hemos cubierto los aspectos básicos para evaluar fuentes de heterogeneidad en modelos de meta-análisis. Quienes tengan interés en profundizar en las medidas de heterogeneidad y su aplicación, recomendamos consultar el artículo de M. J. Bown and A. J. Sutton [3] y los capítulos 7, 8 y 9 de M. Harrer, P. Cuijpers, T. A. Furukawa, and D. D. Ebert [4].

Bibliography

- [1] S. Firke, “janitor: Simple Tools for Examining and Cleaning Dirty Data,” 2024, [Online]. Available: <https://cran.r-project.org/package=janitor>
- [2] S. Nakagawa *et al.*, “orchaRd 2.0: An R package for visualizing meta-analyses with orchard plots,” vol. 12, pp. 4–12, 2023, doi: <https://doi.org/10.32942/X2QC7K>.
- [3] M. J. Bown and A. J. Sutton, “Quality control in systematic reviews and meta-analyses,” *European Journal of Vascular and Endovascular Surgery*, vol. 40, no. 5, pp. 669–677, 2010, doi: 10.1016/j.ejvs.2010.07.011.
- [4] M. Harrer, P. Cuijpers, T. A. Furukawa, and D. D. Ebert, *Doing Meta-Analysis With R: A Hands-On Guide*. Boca Raton, FL and London: Chapman & Hall/CRC Press, 2021. [Online]. Available: <https://www.routledge.com/Doing-Meta-Analysis-with-R-A-Hands-On-Guide/Harrer-Cuijpers-Furukawa-Ebert/p/book/9780367610074>