

Meta-análisis de correlaciones y meta-regresión en R: Guía práctica

Juan David Leongómez 

26 enero, 2023

EvoCo: Laboratorio de Evolución y Comportamiento Humano, Facultad de Psicología, Universidad El Bosque, Bogotá, Colombia. Email: jleongomez@unbosque.edu.co. Web: jleongomez.info.

Resumen

El meta-análisis es un método ampliamente utilizado para sintetizar los datos de diferentes estudios. Sin embargo, a menudo estudiantes, profesionales e investigadores carecen de conocimientos prácticos para llevar a cabo un meta-análisis. Esta guía presenta una serie de herramientas para realizar meta-análisis de correlaciones en R, ilustradas a partir de ejemplos. Incluye desde análisis simples y su interpretación hasta análisis con moderadores (meta-regresión), usando los paquetes `metafor` (Viechtbauer, 2010) y `metaviz` (Kossmeier et al., 2020). También incluye explicaciones para la transformación de coeficientes r de Pearson a z de Fisher (y viceversa), creación de gráficos de bosque (*forest plots*) y gráficos de embudo (*funnel plots*), análisis de heterogeneidad y diagnósticos de influencia, así como estrategias para detectar posibles sesgos de publicación utilizando el paquete `weightr` (Coburn & Veeva, 2019), y para determinar el poder estadístico de un meta-análisis utilizando `metameta` (Quintana, 2022).



Nota: Esta guía está inspirada en [este video](#) de Quintana (2021), pero contiene pasos adicionales y alternativos, citas a fuentes primarias, e información complementaria más detallada. Asume un manejo básico de R, comprensión de correlaciones y regresiones, y un entendimiento de los principios del meta-análisis. De ser necesario, recomiendo ver [este video](#) introductorio al meta-análisis en *jamovi* (Leongómez, 2021) que publiqué en mi canal de YouTube *Investigación Abierta*.

Cita esta guía como _____

Leongómez, J. D. (2022). Meta-análisis de correlaciones y meta-regresión en R: Guía práctica. Zenodo. <https://doi.org/10.5281/zenodo.5640182>

Índice

1. Base de datos de ejemplo	3
1.1. Variables de la base de datos	4
2. Preparación de los datos: transformación de coeficientes <i>r</i> de Pearson a <i>z</i> de Fisher	5
3. Hacer el meta-análisis	6
3.1. Interpretación de los resultados del meta-análisis	6
3.1.1. Alternativa: Reportar el estimado como <i>r</i> de Pearson en vez de <i>z</i> de Fisher	8
3.2. Más información sobre heterogeneidad	9
3.3. Diagnóstico de influencia	10
3.4. <i>Forest plot</i> (diagrama de bosque)	12
3.5. <i>Funnel plot</i> (diagrama de embudo) y sesgo de estudios pequeños	16
3.5.1. <i>Funnel plot</i>	17
3.5.2. Regresión de Egger	21
3.6. Sesgo de publicación (<i>publication bias</i>)	21
3.6.1. Método <i>trim and fill</i> (recorte y relleno)	21
3.6.2. Estimación del modelo de función de peso	23
3.7. Poder estadístico del meta-análisis	25
3.7.1. Instalación de metameta	25
3.7.2. Análisis de poder	25
3.7.3. Visualización del análisis de poder (<i>Firepower plot</i>)	26
3.8. TODAS LAS COMBINACIONES POSIBLES	28
3.8.1. Combinatorial meta-analysis	28
4. Meta-análisis de correlación con moderador (meta-regresión)	31
4.1. Ejemplo 1: Moderación de una variable continua (edad promedio de los participantes)	31
4.1.1. Más información e interpretación de la moderación	32
4.1.1.1. <i>Meta-Analytic Scatter Plot</i> (Gráfico de dispersión meta-analítico)	33
4.1.2. <i>Forest plot</i> y <i>funnel plot</i>	34
4.2. Ejemplo 2: Moderación de una variable categórica (controles usados en cada estudio meta-analizado)	36
4.2.1. Más información e interpretación de la moderación	37
4.2.1.1. <i>Meta-Analytic Scatter Plot</i> (Gráfico de dispersión meta-analítico)	37
4.2.2. <i>Forest plot</i> y <i>funnel plot</i>	40
5. Referencias	41
APÉNDICES	42
A. Alternativas a metafor	42
B. Citas y referencias de paquetes de R	42
C. Paquetes de R y versiones usados para crear este documento (para garantizar reproducibilidad)	43

1. Base de datos de ejemplo

Para los ejemplos usados en ésta guía, usaré la base de datos `dat.molloy2014`, tomada de Molloy et al. (2013). Esta base de datos viene incluida con el paquete `metafor` de R.

Básicamente, Molloy et al. (2013) estudiaron si existe una asociación entre la concienciación (*consciousness*¹) y la adherencia a la medicación. En otras palabras, ¿las personas más *concienciadas* tienden a cumplir más con la medicación prescrita?

Primero, debemos cargar los principales paquetes que usaré a lo largo de esta guía: `metafor` (Viechtbauer, 2010) y `metaviz` (Kossmeier et al., 2020) para hacer e ilustrar los resultados del meta-análisis, así como `dplyr` (Wickham et al., 2021) y `forcats` (Wickham, 2022) para manipular y organizar la base de datos.

```
library(metafor)
library(metaviz)
library(dplyr)
library(forcats)
```

Como ya hemos cargado el paquete `metafor`, ya podemos cargar la base de datos `dat.molloy2014`. En éste caso, para poder *llamarla* cuando sea necesario, la asignaré a un objeto que sencillamente llamaré `dat`.

```
dat <- get(data(dat.molloy2014))
```

Tras asignar la base de datos a este objeto (`dat`), la base de datos se puede ver en la consola de R sencillamente usando como comando el nombre que le dimos al objeto al que lo asignamos (en este caso, `dat`).

```
dat
```

Consola de R: Output 1

	authors	year	ni	ri	controls	design	a_measure
1	Axelsson et al.	2009	109	0.187	none	cross-sectional	self-report
2	Axelsson et al.	2011	749	0.162	none	cross-sectional	self-report
3	Bruce et al.	2010	55	0.340	none	prospective	other
4	Christensen et al.	1999	107	0.320	none	cross-sectional	self-report
5	Christensen & Smith	1995	72	0.270	none	prospective	other
6	Cohen et al.	2004	65	0.000	none	prospective	other
7	Dobbels et al.	2005	174	0.175	none	cross-sectional	self-report
8	Ediger et al.	2007	326	0.050	multiple	prospective	self-report
9	Insel et al.	2006	58	0.260	none	prospective	other
10	Jerant et al.	2011	771	0.010	multiple	prospective	other
11	Moran et al.	1997	56	-0.090	multiple	prospective	other
12	O'Cleirigh et al.	2007	91	0.370	none	prospective	self-report
13	Penedo et al.	2003	116	0.000	none	cross-sectional	self-report
14	Quine et al.	2012	537	0.150	none	prospective	self-report
15	Stilley et al.	2004	158	0.240	none	prospective	other
16	Wiebe & Christensen	1997	65	0.040	none	prospective	other
					c_measure	meanage	quality
1	other		22.00		1		
2	NEO		53.59		1		
3	NEO		43.36		2		
4	other		41.70		1		
5	NEO		46.39		2		
6	NEO		41.20		2		
7	NEO		52.30		1		
8	NEO		41.00		3		
9	other		77.00		2		
10	NEO		78.60		3		
11	NEO		57.20		2		
12	NEO		37.90		2		
13	NEO		39.20		1		
14	other		69.00		2		
15	NEO		46.20		3		
16	NEO		56.00		1		

¹Para una definición detallada del concepto de concienciación, ver John & Srivastava (1999).

1.1. Variables de la base de datos

Por favor mira la base de datos, y familiarízate con sus columnas. ¿Qué información contiene cada variable? Explicaré cada una a continuación.

Primero, hay unas columnas que contienen información absolutamente necesaria para poder hacer un meta-análisis:

- **authors**: los autores de cada estudio a meta-analizar
- **year**: año de publicación de cada estudio
- **ni**: tamaño de muestra de cada estudio
- **ri**: coeficiente de correlación de Pearson de cada estudio

Esas son las variables más importantes y son necesarias para hacer un meta-análisis de correlación simple: alguna columna o columnas con información que permita identificar cada estudio (en este caso **authors** y **year**), una una columna con el tamaño de muestra de cada estudio (en este caso, **ni**), y finalmente columna con los coeficientes de correlación (en este caso la columna **ri**). Si tu meta-análisis no requiere moderadores, con una estructura como esta sería suficiente.

Después hay unas columnas que contienen variables específicas para este este ejemplo, y que son una serie de posibles moderadores que agrupan características de los estudios a meta-analizar:

- **controls**: cantidad de variables controladas (ninguna o múltiples)
- **design**: si se utilizó un diseño transversal o prospectivo
- **a_measure**: tipo de medida de adherencia (autoinforme u otro tipo de medida más “objetivo”)
- **c_measure**: tipo de medida de concienciación (si se midió con alguna versión del inventario de personalidad NEO o con alguna otra escala)
- **meanage**: edad promedio de la muestra
- **quality**: calidad metodológica (la calidad metodológica fue calificada por los autores en una escala de 1 a 4, donde las puntuaciones más altas indican una mayor calidad; para información respecto a cómo se obtuvo esta puntuación, ver el artículo original de Molloy et al. (2013)).

Por supuesto, la salida de la consola no es la más clara, pero para una versión más legible, se puede usar la función **View**, y el nombre de la base de datos o tabla como argumento (en este caso **View(dat)**). Sin embargo, de aquí en adelante mostraré algunas tablas en un formato de impresión (y no como sale en la consola de R), para que sean más fáciles de leer.

Voy a volver a cargar la base de datos (sobrescribiendo el objeto **dat**), para organizarla un poco mejor. Primero, agregaré una nueva columna llamada **study_id**, en la que numeraré los estudios del 1 al 16, lo que será útil para identificar cada estudio en algunas gráficas y análisis. Despues, reorganizaré las columnas para que **study_id** sea la primera, en vez de la última columna. Finalmente, las columnas de variables que son factores (**controls**, **design**, **a_measure**, **c_measure** y **quality**), pueden ser definidas como tal, para evitar pasos adicionales más adelante. En este caso, .

```
dat <- get(data(dat.molloy2014)) %>%
  mutate(study_id = 1:16) %>% #crear columna study_id y agregar número del 1 al 16
  select(study_id, authors:quality) %>% #mover study_id como primera columna
  mutate_at(c("controls", # Transformar variables en factores
            "design",
            "a_measure",
            "c_measure",
            "quality"),
            as.factor) %>%
  mutate(controls = fct_relevel(controls, "none", "multiple"))
```

Con esto, la base de datos tiene ahora la siguiente estructura (Tabla 1):

Tabla 1*Estructura de la base de datos con estudios numerados*

study_id	authors	year	ni	ri	controls	design	a_measure	c_measure	meanage	quality
1	Axelsson et al.	2009	109	0.187	none	cross-sectional	self-report	other	22.00	1
2	Axelsson et al.	2011	749	0.162	none	cross-sectional	self-report	NEO	53.59	1
3	Bruce et al.	2010	55	0.340	none	prospective	other	NEO	43.36	2
4	Christensen et al.	1999	107	0.320	none	cross-sectional	self-report	other	41.70	1
5	Christensen & Smith	1995	72	0.270	none	prospective	other	NEO	46.39	2
6	Cohen et al.	2004	65	0.000	none	prospective	other	NEO	41.20	2
7	Dobbels et al.	2005	174	0.175	none	cross-sectional	self-report	NEO	52.30	1
8	Ediger et al.	2007	326	0.050	multiple	prospective	self-report	NEO	41.00	3
9	Insel et al.	2006	58	0.260	none	prospective	other	other	77.00	2
10	Jerant et al.	2011	771	0.010	multiple	prospective	other	NEO	78.60	3
11	Moran et al.	1997	56	-0.090	multiple	prospective	other	NEO	57.20	2
12	O'Clirigh et al.	2007	91	0.370	none	prospective	self-report	NEO	37.90	2
13	Penedo et al.	2003	116	0.000	none	cross-sectional	self-report	NEO	39.20	1
14	Quine et al.	2012	537	0.150	none	prospective	self-report	other	69.00	2
15	Stilley et al.	2004	158	0.240	none	prospective	other	NEO	46.20	3
16	Wiebe & Christensen	1997	65	0.040	none	prospective	other	NEO	56.00	1

Nota: Datos tomados de Molloy et al. (2013).

2. Preparación de los datos: transformación de coeficientes r de Pearson a z de Fisher

Los coeficientes de Pearson no se distribuyen normalmente, lo que podría llevar a calcular varianzas incorrectas, especialmente cuando se trata de correlaciones con tamaños de muestra pequeños. Por esto, lo mejor es transformar los coeficientes r de Pearson a z de Fisher (Fisher, 1915), que no tienen este problema².

Para transformar los coeficientes r de Pearson a coeficientes z de Fisher, usaré la función `escalc` del paquete `metafor`. Los argumentos que requiere esta función, además del tipo de transformación a realizar (en este caso `measure = "ZCOR"`), son los coeficientes de correlación (`ri`), el tamaño de muestra de cada correlación (`ni`), y la base de datos que contiene estos valores (`data`). En nuestro caso, las columnas donde están estos valores, tienen los mismos nombres (`ri`, `ni`). En este ejemplo, asignaré el resultado de esta función al mismo objeto `dat`, que contiene la base de datos, para sobrescribirlo y no crear objetos adicionales.

```
dat <- escalc(measure = "ZCOR", ri = ri, ni = ni, data = dat)
```

Esta función agrega dos nuevas variables: `yi`, que es el tamaño de efecto (en valores z de Fisher), y `vi` que es la varianza (Tabla 2).

²Para decirlo de manera más precisa, el problema es que la distribución del coeficiente de correlación de Pearson (r), es muy sesgada cuando se trata de variables altamente correlacionadas (positiva, o negativamente). Esto dificulta la estimación de los intervalos de confianza y por tanto la aplicación de las pruebas de significación para coeficientes r . La transformación de Fisher de valores r a z —que es la [tangente hiperbólica inversa](#) de r —resuelve este problema, pues los coeficientes z tienen una distribución aproximadamente normal, y una varianza estable a lo largo de diferentes valores posibles de r (para una demostración en español, ver Sánchez-Bruno & Borges del Rosal, 2005).

Tabla 2*Estructura de la base de datos, con transformación de los r de Pearson a z de Fisher*

study_id	authors	year	ni	ri	controls	design	a_measure	c_measure	meanage	quality	yi	vi
1	Axelsson et al.	2009	109	0.187	none	cross-sectional	self-report	other	22.00	1	0.1892266	0.0094340
2	Axelsson et al.	2011	749	0.162	none	cross-sectional	self-report	NEO	53.59	1	0.1634399	0.0013405
3	Bruce et al.	2010	55	0.340	none	prospective	other	NEO	43.36	2	0.3540925	0.0192308
4	Christensen et al.	1999	107	0.320	none	cross-sectional	self-report	other	41.70	1	0.3316471	0.0096154
5	Christensen & Smith	1995	72	0.270	none	prospective	other	NEO	46.39	2	0.2768638	0.0144928
6	Cohen et al.	2004	65	0.000	none	prospective	other	NEO	41.20	2	0.0000000	0.0161290
7	Dobbels et al.	2005	174	0.175	none	cross-sectional	self-report	NEO	52.30	1	0.1768200	0.0058480
8	Ediger et al.	2007	326	0.050	multiple	prospective	self-report	NEO	41.00	3	0.0500417	0.0030960
9	Insel et al.	2006	58	0.260	none	prospective	other	other	77.00	2	0.2661084	0.0181818
10	Jerant et al.	2011	771	0.010	multiple	prospective	other	NEO	78.60	3	0.0100003	0.0013021
11	Moran et al.	1997	56	-0.090	multiple	prospective	other	NEO	57.20	2	-0.0902442	0.0188679
12	O'Clirigh et al.	2007	91	0.370	none	prospective	self-report	NEO	37.90	2	0.3884231	0.0113636
13	Penedo et al.	2003	116	0.000	none	cross-sectional	self-report	NEO	39.20	1	0.0000000	0.0088496
14	Quine et al.	2012	537	0.150	none	prospective	self-report	other	69.00	2	0.1511404	0.0018727
15	Stilley et al.	2004	158	0.240	none	prospective	other	NEO	46.20	3	0.2447741	0.0064516
16	Wiebe & Christensen	1997	65	0.040	none	prospective	other	NEO	56.00	1	0.0400214	0.0161290

Nota: Las nuevas columnas creadas usando la función `escalc` (yi como tamaño de efecto y vi como varianza) están resaltadas en naranja

3. Hacer el meta-análisis

Para hacer el meta-análisis, usaré la función `rma` del paquete `metafor`. Esta función requiere especificar los tamaños de efecto (yi) y varianzas (vi) de los estudios a meta-analizar. En nuestro caso, las columnas donde están estos valores, tienen los mismos nombres (yi, vi). Asignaré los resultados del meta-análisis a un nuevo objeto llamado `res`.

```
res <- rma(yi = yi, vi = vi, data = dat)
```

Los resultados, son los siguientes:

```
res
```

Consola de R: Output 2

```
Random-Effects Model (k = 16; tau^2 estimator: REML)

tau^2 (estimated amount of total heterogeneity): 0.0081 (SE = 0.0055)
tau (square root of estimated tau^2 value):      0.0901
I^2 (total heterogeneity / total variability):   61.73%
H^2 (total variability / sampling variability):  2.61

Test for Heterogeneity:
Q(df = 15) = 38.1595, p-val = 0.0009

Model Results:

estimate      se     zval    pval   ci.lb   ci.ub
 0.1499  0.0316  4.7501 <.0001  0.0881  0.2118 ***

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

3.1. Interpretación de los resultados del meta-análisis

Vamos a analizar estos resultados de la consola de R por partes:

Primero, nos confirman que ajustamos un modelo con efectos aleatorios (`Random-Effects Model`), a partir de 16 estudios (`k = 16`), y que para estimar τ^2 (tau cuadrado) usamos el método de **máxima verosimilitud restringida**³ (`tau^2 estimator: REML`), que se designa como *REML* por sus siglas en inglés .

³Hay varios métodos disponibles como estimador, además de **máxima verosimilitud restringida** (REML). Sin embargo, si tienes dudas, REML es una buena opción. Cada método tiene ventajas y desventajas que, si tienes interés en mirar, están descritas en la documentación de la función `rma`.

Posteriormente, nos provee los valores de una serie de estimadores de heterogeneidad:

- τ^2 : tau^2 (estimated amount of total heterogeneity): 0.0081 (SE = 0.0055)
- τ : tau (square root of estimated tau^2 value): 0.0901
- I^2 : I^2 (total heterogeneity / total variability): 61.73%, y
- H^2 : H^2 (total variability / sampling variability): 2.61

La tercera parte, reporta otra prueba de heterogeneidad, usando el estadístico Q :

- Test for Heterogeneity:

$Q(df = 15) = 38.1595$, p-val = 0.0009

De todos estos, los más comúnmente reportados son τ^2 , τ , I^2 y Q . Cada una de estas medidas tiene ventajas y desventajas, por lo cual tiene sentido reportarlas todas.

- I^2 : tiene la ventaja de ser sencillo de interpretar, pues hay criterios generales para heterogeneidad baja, moderada y alta (típicamente 25%, 50%, and 75%, respectivamente). Sin embargo, es muy sensible a los tamaños de muestra de los estudios meta-analizados (por ejemplo, si en tu meta-análisis hay estudios con tamaños de muestra muy grandes, esto va a sesgar tu I^2).
- Q : aunque no es sensible al tamaño de muestra, es sensible al número de estudios meta-analizados. Tiene la ventaja de ser un test de hipótesis, y como tal, puede ser interpretado a partir de su valor p .
- τ^2 : no tiene problemas de sensibilidad a los tamaños de muestra o número de estudios meta-analizados, pero es más difícil de interpretar. τ^2 es una estimación de la varianza de los tamaños de los efectos reales entre los estudios meta-analizados. Se usa, principalmente, para asignar pesos a cada estudio (para más información, ver [Borenstein et al., 2009](#)).

En nuestro caso, el estadístico Q sugiere que hay una heterogeneidad significativa en los estudios meta-analizados ($p = 0.0009$). I^2 , sugiere una heterogeneidad moderada, lo que quiere decir que se estima que más de la mitad (61.73%) de la varianza se deriva de diferencias en los tamaños de efecto.

Más abajo, el *output* de la consola de R nos muestra los resultados de nuestro meta-análisis (**Model results**); en otras palabras, ¿cuál es el tamaño de efecto de la asociación entre concienciación (*conscientiousness*) y la adherencia a la medicación, según nuestro meta-análisis?

Esta parte nos provee varios resultados:

- **Estimate (0.1499)**: estimado de la correlación entre concienciación y adherencia a la medicación, en este caso en valores z de Fisher, pues así transformamos los coeficientes de cada estudio
- **se (0.0316)**: error estándar del estimado de la asociación (en valores z de Fisher)
- **zval (4.7501)**: estadístico Z (mayúscula) que comprueba la media de una distribución. No se debe confundir con la transformación de coeficientes de correlación a z de Fisher (minúscula); este estadístico no nos provee una estimación de la asociación entre las variables correlacionadas, sino, de manera similar a una prueba t , nos sugiere si nuestra media (para el caso, el resultado de nuestro meta-análisis), se diferencia de 0 (o una correlación nula). Cuando Z es mayor a 1.96 (o menor a -1.96), nuestro resultado está en el 5% extremo de la distribución Z y sería significativo con un α tradicional de 0.05 (dos colas)
- **pval (<.0001)**: valor p de la correlación meta-analizada
- **ci.lb (0.0881)**: límite inferior del intervalo de confianza (*confidence interval lower bound*) de la correlación meta-analizada (en valores z de Fisher)
- **ci.ub (0.2118)**: límite superior del intervalo de confianza (*confidence interval upper bound*) de la correlación meta-analizada (en valores z de Fisher)
- Nivel de significación (***)**: representación con asteriscos (o un punto) del nivel de significación**
- **Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '**: Clave para interpretar los niveles de significación. Aunque puede parecer complejo, básicamente, quiere decir que tres asteriscos (***) representan un valor p entre 0 y 0.001 (lo que comúnmente se representa como $p < .001$); dos asteriscos (**) un valor

p entre 0.001 y 0.01 ($p < .01$); un asterisco (*) un valor p entre 0.01 y 0.05 ($p < .05$); un punto (.) un valor p entre 0.05 y 0.01 ($p < .1$, que ya no es significativo); y si no hay ningún símbolo, un valor p entre 0.1 y 1 ($p > .1$, no significativo)

En este caso, el meta-análisis nos sugiere que en efecto existe una asociación positiva entre concienciación y adherencia a la medicación (coeficiente de correlación transformado en z de Fisher = .1499), con un error estándar de 0.0316. Así mismo, sugiere que esa asociación es significativa ($Z = 4.7501, p < .0001$), y nos muestra el intervalo de confianza del 95% (o IC 95%); los intervalos de confianza (en este caso del 95%) lo que estiman es que, si hiciéramos 100 muestras independientes, 95 de éstas contendrían una asociación que estaría entre los límites inferior ($z = .0881$) y superior ($z = .2118$) de estos intervalos de confianza.

Esto se podría resumir, por ejemplo, como:

El resultado meta-analizado sugiere que existe una asociación positiva entre concienciación y adherencia a la medicación ($z \pm se = 0.15 \pm 0.032$, IC 95% [0.09, 0.21]; $Z = 4.75, p < .0001$). Sin embargo, es importante considerar que hay heterogeneidad entre los estudios meta-analizados ($\tau^2 \pm se = 0.0081 \pm 0.0055; \tau = 0.0901; Q(15) = 38.16, p < .001; I^2 = 61.7\%$).

3.1.1. Alternativa: Reportar el estimado como r de Pearson en vez de z de Fisher

Para reportar la correlación, si prefieres reportar coeficientes r de Pearson en vez de la transformación a z de Fisher, puedes transformar los valores z de Fisher de vuelta a r de Pearson. Para esto existen múltiples opciones en R, incluyendo simplemente usar la función `tanh`, que calcula la tangente hiperbólica, o la función `fisherz2r` del paquete `psych` (Revelle, 2021). Por ejemplo, para transformar el estimado de nuestro meta-análisis a r de Pearson, solo debo usar alguna de esas funciones, y agregar el valor z (0.1499 en nuestro caso) como único argumento:

```
tanh(0.1499)
```

Consola de R: Output 3

```
[1] 0.1487872
```

```
library(psych)
fisherz2r(0.1499)
```

Consola de R: Output 4

```
[1] 0.1487872
```

Cualquiera de estas opciones nos da un valor de $r = 0.1487872$, muy similar al obtenido ($z = 0.1499$). Esto se debe a que, para coeficientes r de Pearson entre -0.4 y 0.4, la transformación a valores z de Fisher produce resultados muy similares (Fig. 1).

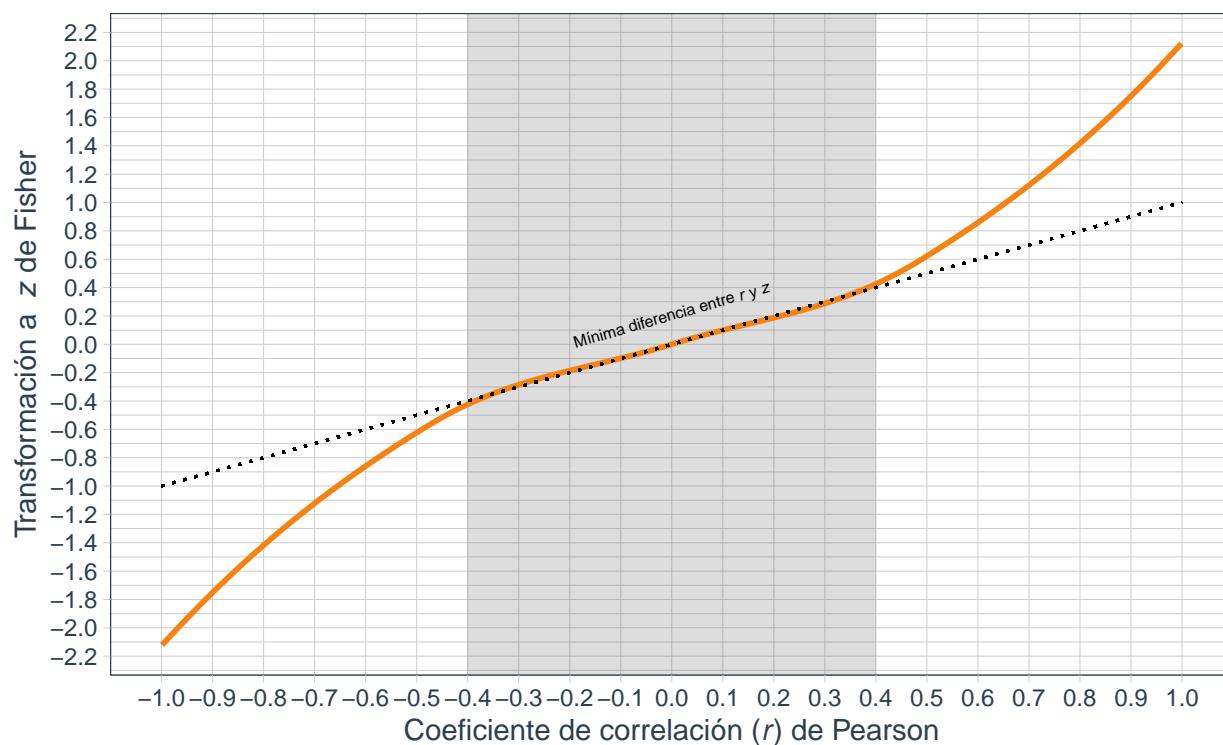


Figura 1. Asociación entre coeficientes de correlación r de Pearson (eje X), y su transformación a z de Fisher (eje Y). La línea naranja representa la asociación entre valores r y z ; como referencia, la línea negra punteada representa igualdad entre ejes ($y = x$). Como se puede ver, cuando r está aproximadamente entre -0.4 y 0.4 (rectángulo gris), los valores r y z son casi idénticos. Para valores más extremos, el valor de z se aleja progresivamente del valor de r .

Por supuesto, si decides reportar los resultados de tu meta-análisis en coeficientes r de Pearson, siempre puedes hacer lo mismo con el error estándar y los límites del intervalo de confianza del 95 % (todos valores en z de Fisher, pues fue el tamaño de efecto que meta-analizamos).

3.2. Más información sobre heterogeneidad

Es importante tener en cuenta que la heterogeneidad no es un *supuesto* que se deba cumplir al hacer un meta-análisis, y por ende una heterogeneidad moderada o alta no invalida sus resultados. Sencillamente es información útil que se debe reportar y tener en cuenta al interpretar el resultado de un meta-análisis. En este caso, la presencia de heterogeneidad sugiere que los estudios meta-analizados varían y no suelen reportar resultados similares.

De manera general, y para decirlo de manera más técnica, la presencia de heterogeneidad estadística es indicativa de una variación entre los estudios en la magnitud y la dirección de la estimación del efecto estudiado (para más información y ejemplos, ver Sedgwick, 2015).

Por esto, reportar información detallada acerca de la heterogeneidad de los estudios meta-analizados es siempre útil. De hecho, además de reportar los estadísticos τ^2 , τ , I^2 y Q (como expliqué en la sección 3.1), podemos fácilmente calcular los intervalos de confianza para τ^2 , τ , e I^2 (además de H^2 , que no he usado) con la función `confint`.

```
confint(res)
```

Consola de R: Output 5

	estimate	ci.lb	ci.ub
τ^2	0.0081	0.0017	0.0378
τ	0.0901	0.0412	0.1944
$I^2(%)$	61.7324	25.2799	88.2545
H^2	2.6132	1.3383	8.5139

Para τ^2 , el hecho de que los límites del intervalo de confianza no crucen el 0 (es decir, no hay un límite negativo y otro positivo; en nuestro caso, ambos son positivos: IC 95 % [0.0017, 0.0378]), también sugiere que hay

heterogeneidad entre los estudios que meta-analizamos.

Estos intervalos de confianza que también pueden ser reportados junto a sus correspondientes estadísticos. Entonces, esto se podría resumir, por ejemplo, como:

El resultado meta-analizado sugiere que en efecto existe una asociación positiva entre concienciación y adherencia a la medicación ($z \pm se = 0.15 \pm 0.032$, IC 95% [0.09, 0.21]; $Z = 4.75$, $p < .0001$). Sin embargo, es importante considerar que hay heterogeneidad entre los estudios meta-analizados ($\tau^2 \pm se = 0.0081 \pm 0.0055$, IC 95% [0.0017, 0.0378]; $\tau = 0.0901$, IC 95% [0.0412, 0.1944]; $Q(15) = 38.16$, $p < .001$; $I^2 = 61.7\%$, IC 95% [25.2799%, 88.2545%]).

3.3. Diagnóstico de influencia

Otro aspecto importante de un meta-análisis, es determinar si alguno(s) de los estudios meta-analizados es(son) particularmente influyente(s) en nuestro resultado⁴. Para esto, se suele usar la técnica conocida como *leave-one-out*, que se refiere al resultado del meta-análisis cuando se excluye cada estudio; al estimar cómo y cuánto cambia el resultado del modelo de meta-análisis al excluir cada estudio, podemos estimar su influencia en el resultado. Dicho de otra manera, si al excluir un estudio el resultado cambia mucho, sabemos que ese estudio tiene gran influencia en el meta-análisis y sería mejor excluirlo.

Para hacer un análisis de influencia, podemos usar la función `influence` del paquete `metafor`, cuyo resultado, en este caso, asignaré a un objeto llamado `inf.res`.

```
inf.res <- influence(res)
```

Dado que lo asigné a un objeto (`inf.res`), para ver el resultado, debo usar como comando el nombre que le di al objeto.

```
inf.res
```

Consola de R: Output 6

	rstudent	dffits	cook.d	cov.r	tau2.del	QE.del	hat	weight	dfbs	inf
1	0.2918	0.0485	0.0025	1.1331	0.0091	37.7109	0.0568	5.6776	0.0481	
2	0.1196	-0.0031	0.0000	1.2595	0.0100	36.7672	0.1054	10.5396	-0.0032	
3	1.2740	0.2595	0.0660	0.9942	0.0075	35.3930	0.0364	3.6432	0.2623	
4	1.4711	0.3946	0.1439	0.9544	0.0068	33.5886	0.0562	5.6195	0.3994	
5	0.8622	0.1838	0.0339	1.0505	0.0082	36.5396	0.0441	4.4069	0.1837	
6	-0.9795	-0.2121	0.0455	1.0639	0.0084	37.1703	0.0411	4.1094	-0.2112	
7	0.2177	0.0296	0.0010	1.1740	0.0094	37.6797	0.0714	7.1362	0.0296	
8	-0.9774	-0.3120	0.1001	1.1215	0.0084	36.1484	0.0889	8.8886	-0.3128	
9	0.7264	0.1392	0.0195	1.0561	0.0083	37.0495	0.0379	3.7886	0.1387	
10	-1.8667	-0.5861	0.2198	0.8502	0.0047	25.0661	0.1058	10.5826	-0.5430	
11	-1.4985	-0.2771	0.0756	1.0073	0.0077	35.6617	0.0369	3.6922	-0.2791	
12	1.8776	0.4918	0.2148	0.8819	0.0059	31.9021	0.0511	5.1150	0.5059	
13	-1.1892	-0.2939	0.0859	1.0550	0.0080	36.3291	0.0587	5.8732	-0.2941	
14	-0.0020	-0.0423	0.0021	1.2524	0.0100	37.7339	0.0998	9.9778	-0.0434	
15	0.8066	0.2126	0.0459	1.0907	0.0083	35.8385	0.0684	6.8403	0.2125	
16	-0.7160	-0.1656	0.0280	1.0853	0.0087	37.7017	0.0411	4.1094	-0.1642	

Esto me muestra gran cantidad de información de cada estudio (en este caso, lo presento como una tabla sin formato, tal cual se ve en la consola de R). Las columnas que incluye son:

- **rstudent**: residuo estandarizados externamente. Esto **no** corresponde al coeficiente de correlación de cada estudio, sino a la diferencia entre el tamaño de efecto observado en cada estudio, y la predicción de dicho valor cuando dicho estudio se elimina del meta-análisis
- **dffits**: diferencia de ajuste(s) (en inglés *difference in fit(s)*). Es una medida diagnóstica de la influencia de cada punto en una regresión (en este caso, cada estudio en un meta-análisis) propuesta originalmente por Belsley et al. (1980)

⁴Por ejemplo, si estuviésemos meta-analizando 20 estudios, de los cuales 19 tienen un n de 100, pero el otro tiene un n de 10.000, éste último tendrá una influencia enorme en nuestro resultado. Sería preocupante que nuestro meta-análisis sea dependiente principalmente de un único estudio.

- **cook.d**: Distancia de Cook (en inglés, *Cook's distance*). Es otra medida diagnóstica de la influencia, propuesta originalmente por Cook (1977). Es conceptualmente idéntica a DFFITS (**dffits**), y he hecho existir una fórmula para convertir una medida en la otra (Henry et al., 2003). Aunque no hay un acuerdo absoluto respecto, comúnmente se asume que valores mayores a 1 ($D_i > 1$), representan puntos altamente influyentes en un modelo (Cook & Weisberg, 1982), que probablemente deban ser excluidos
- **cov.r**: relación (o proporción) de covarianza. Es el determinante de la matriz de varianza-covarianza de las estimaciones de los parámetros basadas en el conjunto de datos, cuando cada estudio se elimina del meta-análisis, dividido por el mismo determinante de la matriz de varianza-covarianza cuando se incluyen todos los estudios. Básicamente, un valor menor a 1 indica que la eliminación de ese estudio produce estimaciones más precisas de los coeficientes del modelo
- **tau2.del**: es la heterogeneidad (residual) τ^2 cuando se elimina cada estudio
- **QE.del**: similar al resultado anterior, este se refiere al estadístico Q obtenido cuando se excluye cada estudio
- **hat**: los valores *hat* (h) son una medida estandarizada de la distancia de cada punto a la media de la variable predictora. Mientras valores cercanos a 0 indican que no hay un valor preocupante, valores cercanos a 1, aunque no indican directamente alta influencia de algún punto, ciertamente nos indican que vale la pena investigar más. Los valores *hat* están abiertos a la interpretación, pero un valor de corte que es común es el doble de la media de los todos valores *hat* (\bar{h}); cualquier estudio con un valor mayor debe ser examinado con cuidado
- **weight**: peso asignado a cada estudio
- **dfbs**: el valor de **dfbs** (o *DFBETAS*) indica cuántas desviaciones estándar cambia el coeficiente estimado después de excluir cada estudio del modelo de meta-análisis
- **inf**: por suerte, esta columna **inf** resume esta información por nosotros. Cualquier estudio que se considere influyente, teniendo en cuenta la diferencia de ajuste, la distancia de Cook, los valores *hat* o *DFBETAS*, será señalado acá como influyente, usando asteriscos.

Aunque hay mucha información, lo más importante ahora es mirar la última columna, llamada **inf**. Si ahí aparecieran asteriscos para algún estudio meta-analizado (que no es nuestro caso), sugeriría que ese estudio es particularmente influyente y podría ser necesario eliminarlo del meta-análisis.

Por último, podemos también ver ésta información que tenemos guardada en el objeto **inf.res**, de manera gráfica, usando la función **plot** (Fig. 2).

```
plot(inf.res)
```

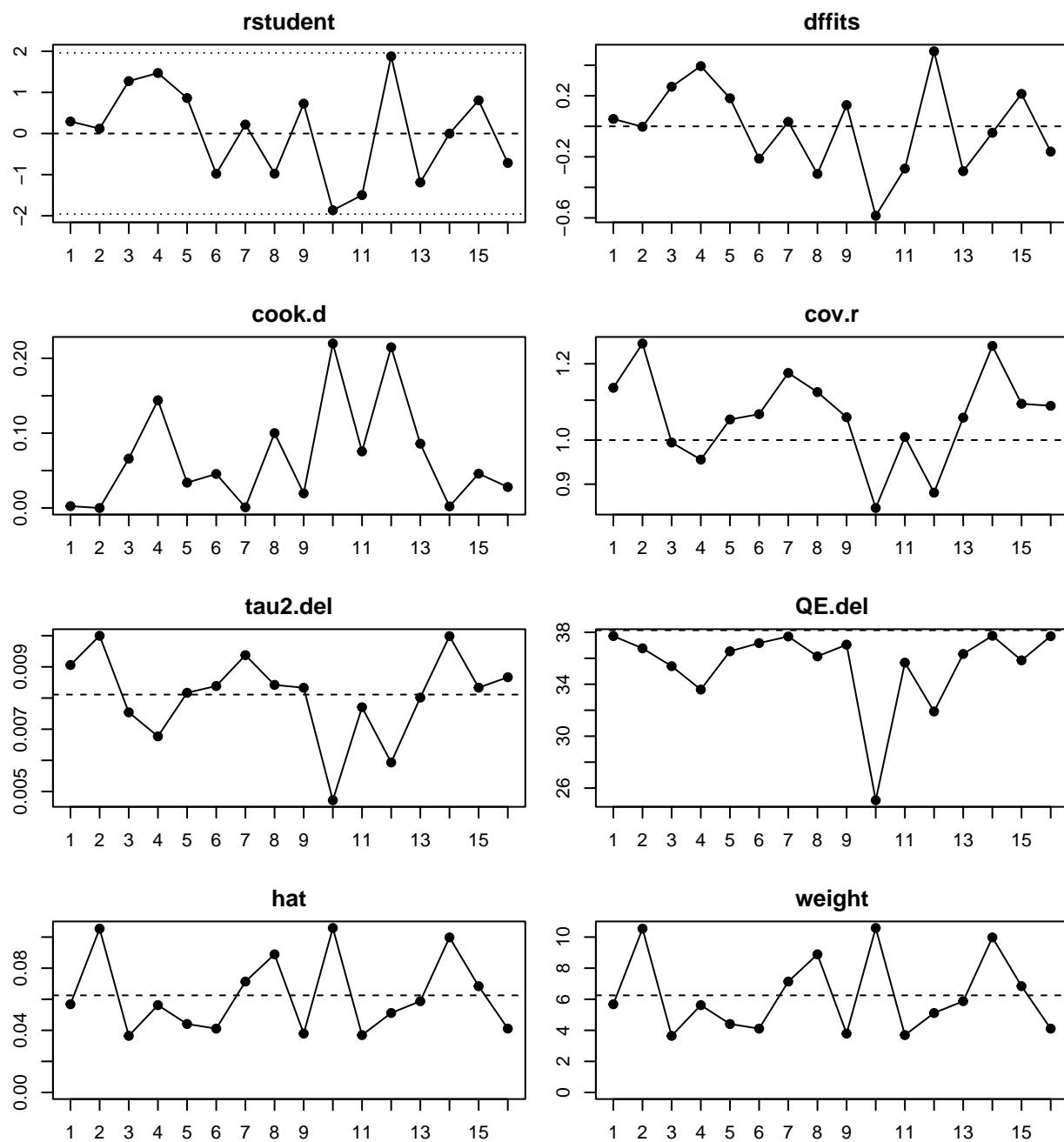


Figura 2. Diagnóstico de influencia. Estudios particularmente influyentes serían representados con un punto rojo. Los números 1 a 16 en el eje X representan cada estudio, como lo definimos en columna `study_id` de la Tabla 1. En este caso, no hay ningún estudio que se considere demasiado influyente, por lo que este análisis sugiere que podemos estar tranquilos con nuestro meta-análisis.

3.4. Forest plot (diagrama de bosque)

Para hacer un diagrama de bosque (*forest plot*) con `metafor` resumiendo nuestro meta-análisis, solo tenemos que usar la función `forest`, usando como argumento el objeto al que asignamos los resultados de nuestro meta-análisis (`res`; esto produce la Fig. 3).

```
forest(res)
```

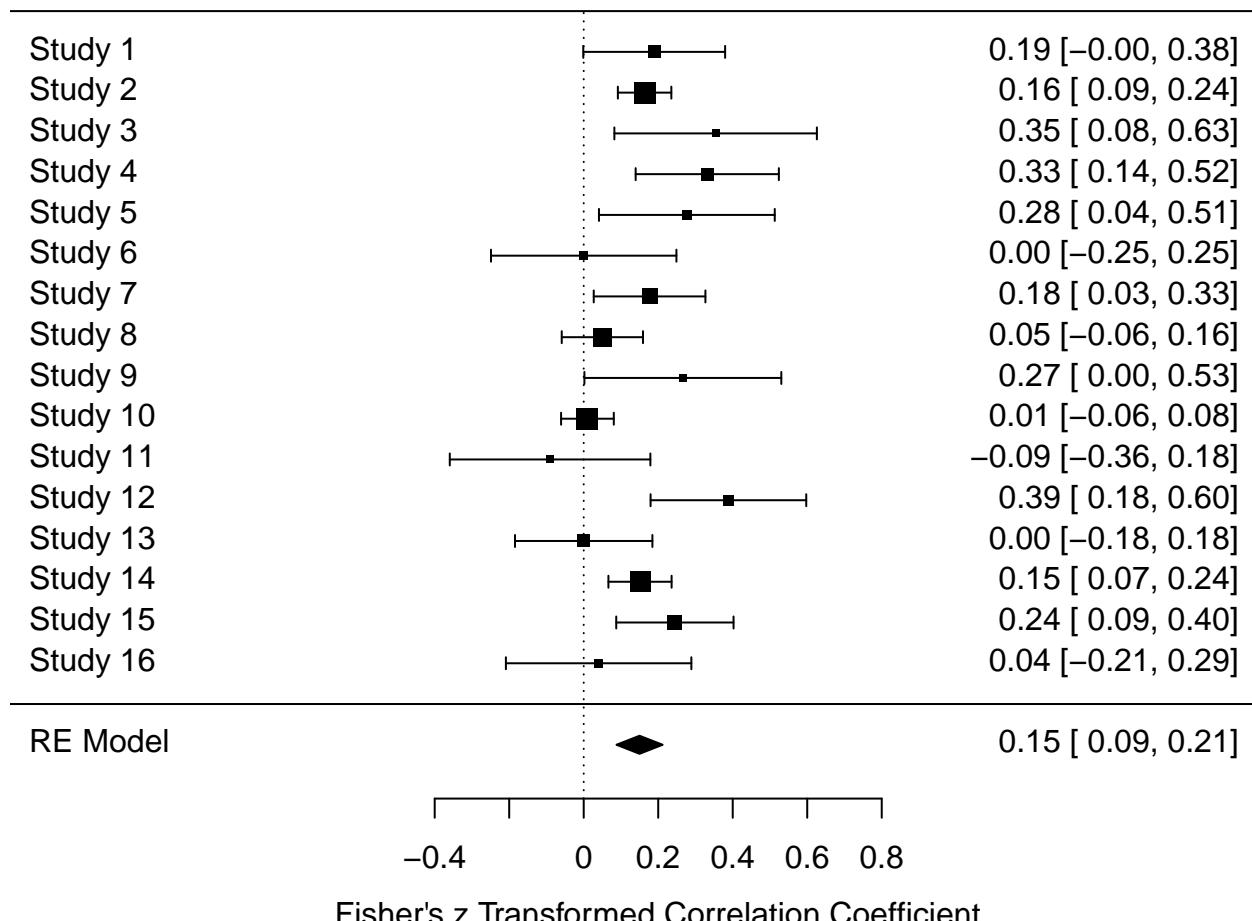


Figura 3. Forest plot básico de `metafor`. Para cada estudio meta-analizado, tenemos el efecto (correlación, en este caso en valores z de Fisher), así como sus intervalos de confianza entre paréntesis cuadrados. Esta misma información está representada gráficamente, con los cuadrados representando el efecto de cada estudio así como sus intervalos de confianza como barras de error, y el tamaño de muestra representado por el tamaño del cuadrado. Bajo estos resultados, tenemos nuestro meta-análisis, con el mismo formato en texto, pero representando el efecto y sus intervalos de confianza con un diamante.

Como se puede ver en las Figuras 3, 4 y 5 (que son versiones del mismo *forest plot*), no es una sorpresa que el análisis nos sugiera bastante heterogeneidad; las correlaciones encontradas entre los diferentes estudios varían mucho (están entre -0.09 y 0.37), y aunque son positivas en la mayoría de los casos (en algunos claramente positivas), en otros casos son prácticamente 0 o incluso negativas.

Para una versión más completa y anotada, también usando el *plot* básico de `metafor`, pero agregando encabezados de cada columna en español, nombres de los estudios meta-analizados⁵ así como una columna con los pesos dados a cada estudio, y detalles del modelo final⁶, podemos hacer algo como esto:

```
# forest plot con anotaciones adicionales
forest(res, cex = 0.75, xlim = c(-1.6, 1.6),
       slab = paste(dat$authors, dat$year, sep = ", "),
       showweights = TRUE,
       xlab = "Coeficiente de correlación transformado en z de Fisher",
       digits = c(2,3L),
       mlab = bquote(paste("Modelo EA: Q(", .(res$k - res$p), ") = ",
                           .(formatC(res$QE, digits = 2, format = "f")),
                           ", p ", .(scales::pvalue(res$pval)), "; ", I^2, " = ",
                           .(formatC(res$I2, digits = 1, format = "f")), "%")))
```

⁵En este caso, y dado que tenemos la lista de autores y años de publicación en columnas separadas, pegando las columnas `authors` y `year` separadas por una coma y un espacio: `paste(dat$authors, dat$year, sep = ", ")` como argumento `slab`.

⁶Estas opciones están explicadas [acá](#).

```
# agregar encabezados a las columnas (valores de X y Y deben ser ajustados)
op <- par(cex = 0.8, font = 2)
text(x = -1.6, y = 18, labels = "Autor(es), Año", pos = 4)
text(x = 0, y = 18, labels = "Efecto e IC", pos = 4)
text(x = 1, y = 18, labels = "Peso", pos = 2)
text(x = 1.6, y = 18, labels = "Corr. [95% IC]", pos = 2)
```

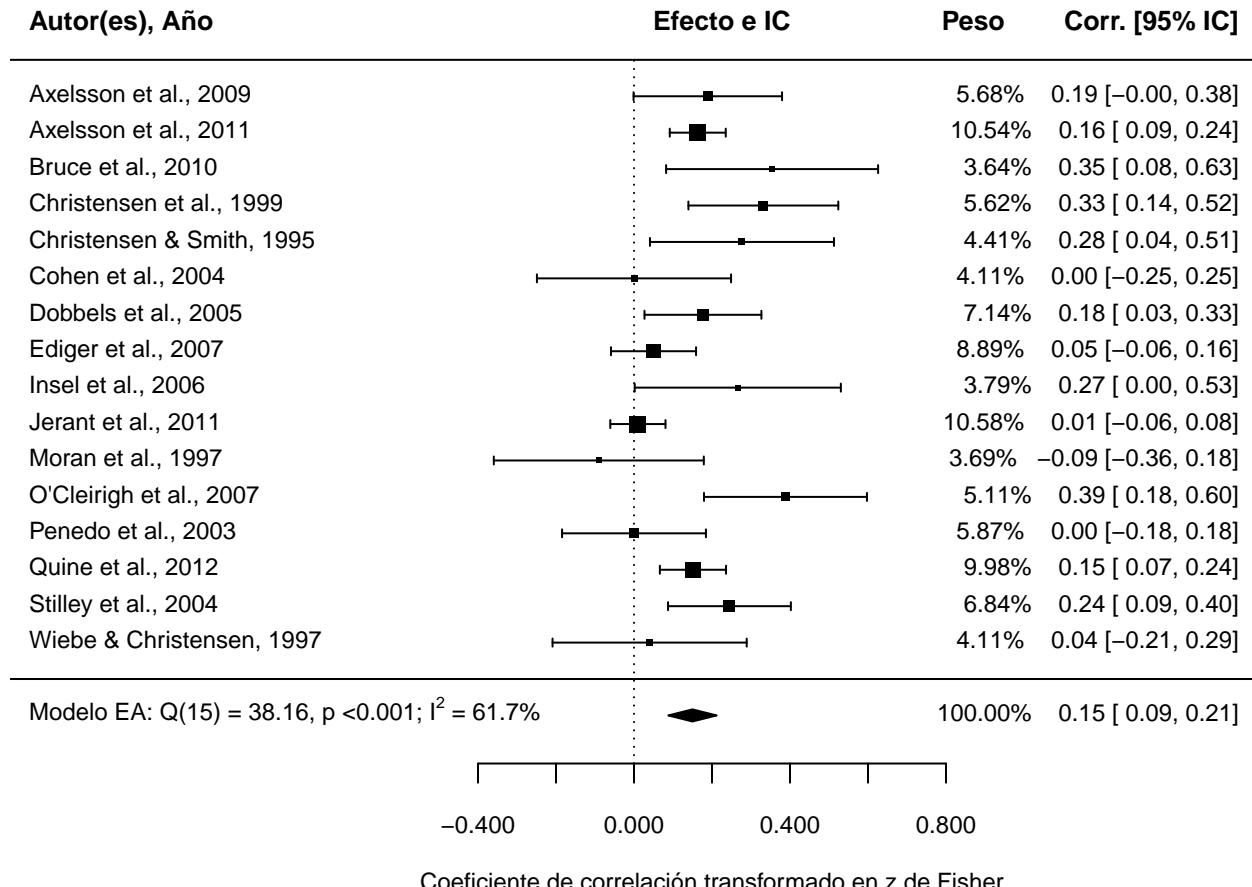


Figura 4. Forest plot anotado, creado con `metafor`. En esta versión agregué algunos encabezados en español, así como estadísticos generales del modelo de meta-análisis. Modelo EA se refiere al modelo meta-analizado, de efectos aleatorios.

O, para un *forest plot* incluso más sofisticado, se puede usar la función `viz_forest` del paquete `metaviz`.

```
# A. Variante "classic" (no tiene que ser definida, pues es la opción por defecto)
viz_forest(res,
           study_labels = paste(dat$authors, dat$year, sep = ", "),
           xlab = "Correlación",
           annotate_CI = TRUE,
           summary_label = "Resumen",
           text_size = 2.6,
           x_trans_function = tanh)

# B. Variante "thick"
viz_forest(res,
           study_labels = paste(dat$authors, dat$year, sep = ", "),
           xlab = "Correlación",
           variant = "thick",
           col = "Greens",
           annotate_CI = TRUE,
```

```

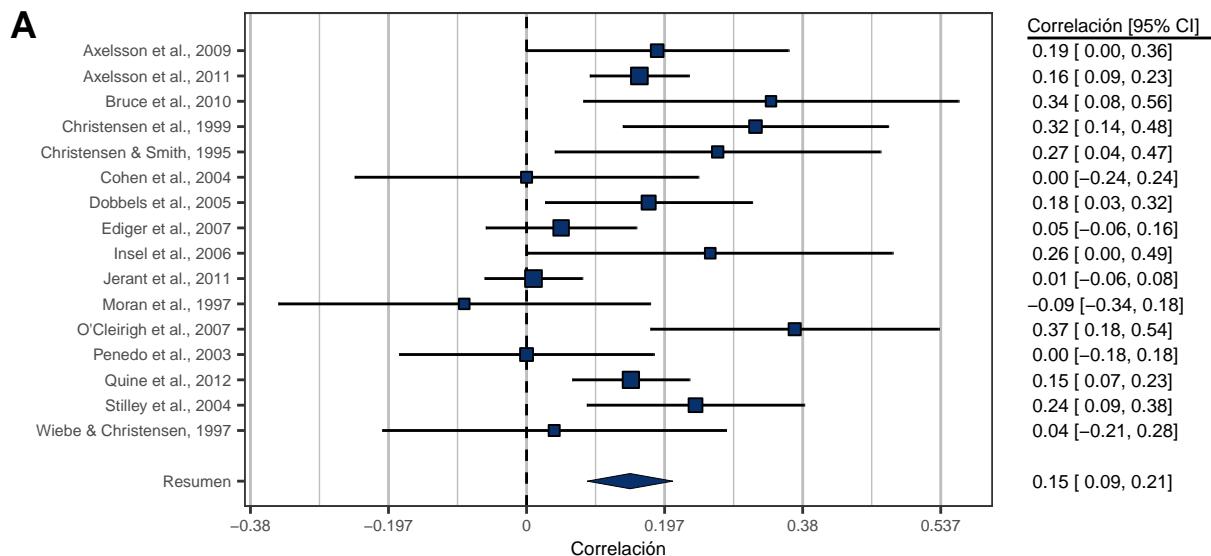
summary_label = "Resumen",
text_size = 2.6,
x_trans_function = tanh)

# C. Variante "rain"
viz_forest(res,
study_labels = paste(dat$authors, dat$year, sep = ", "),
xlab = "Correlación",
variant = "rain",
col = "Oranges",
annotate_CI = TRUE,
summary_label = "Resumen",
text_size = 2.6,
x_trans_function = tanh)

```

Con el código anterior genero las siguientes tres versiones del mismo *forest plot* (Fig. 5) usando diferentes variantes y escalas de colores, y transformando de vuelta los coeficientes de z de Fisher a r de Pearson (con el argumento `x_trans_function = tanh`).

Por supuesto, es cuestión de gusto cuál usar.



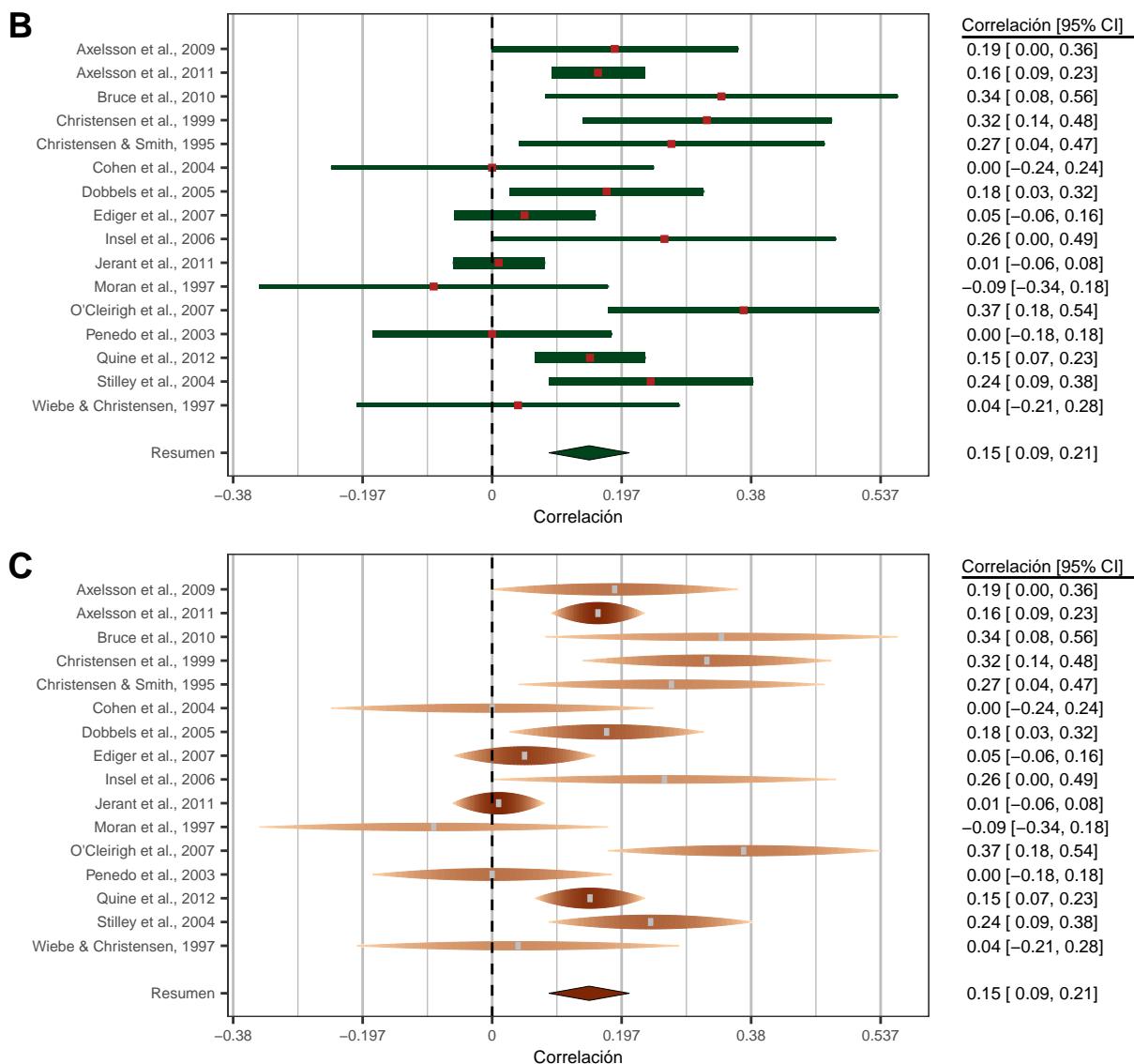


Figura 5. Variantes de *forest plots* creados con `metaviz`. **A.** Variante clásica (opción por defecto). **B.** Variante “thick” y escala de colores “Greens”. **C.** Variante “rain” y escala de colores “Oranges”.

3.5. *Funnel plot* (diagrama de embudo) y sesgo de estudios pequeños

En este punto, es en donde más errores se cometan. Las pruebas más comunes para evaluar sesgos de publicación, son la evaluación de la asimetría en el *funnel plot* (diagrama de embudo), y la regresión (o test) de Egger (Egger et al., 1997).

Sin embargo, el principal error que la mayoría de los investigadores (meta-analistas) cometan, es que simplemente basándose en éstos métodos, concluyen que un meta-análisis tiene (o no) riesgo de sufrir de un sesgo de publicación⁷. Sin embargo, estos métodos, no son pruebas exclusivas de sesgo de publicación, sino de sesgo de estudios de tamaño muestral pequeño (ver e.g. Schwarzer et al., 2015b), que pueden incluir sesgo de publicación, pero no se centran exclusivamente en éste.

A pesar de esto, tanto la regresión de Egger como el *funnel plot*, son interesantes dado que el sesgo de estudios pequeños es importante.

⁷Por supuesto, yo mismo soy culpable de haber hecho esto en el pasado, incluyendo algunos comentarios en mi video sobre meta-análisis en *jamovi* (Leongómez, 2021).

3.5.1. Funnel plot

Para crear un *funnel plot* con `metafor`, de nuestro meta-análisis, solo tenemos que usar la función `funnel`, usando como argumento el objeto al que asignamos los resultados de nuestro meta-análisis (`res`). Con esto, generamos la Figura 6.

```
funnel(res)
```

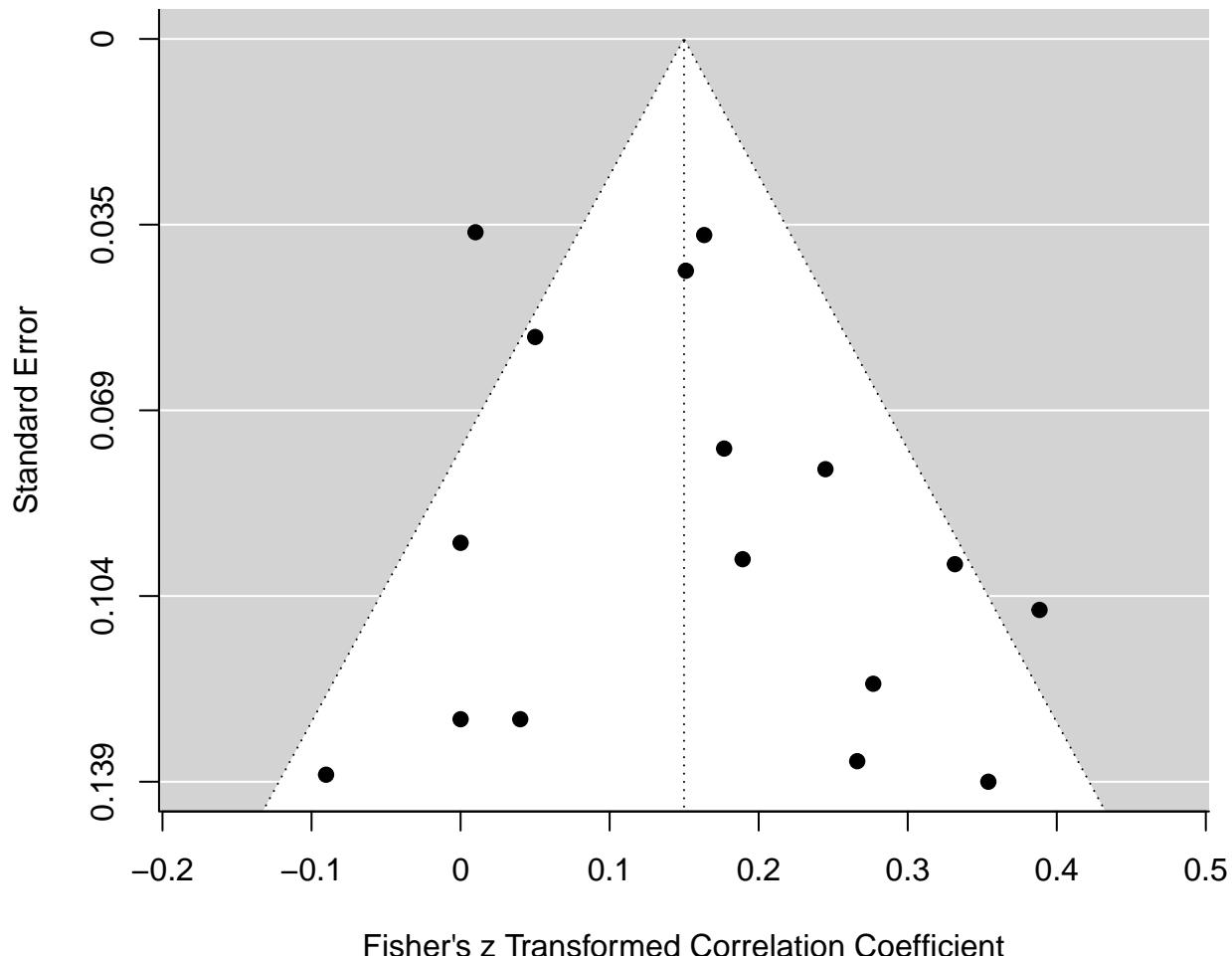


Figura 6. Funnel plot básico de `metafor`. Para cada estudio meta-analizado, tenemos el efecto (correlación, en este caso en valores z de Fisher) en el eje X , así como su error estándar en el eje Y . La línea punteada vertical representa el efecto meta-analizado que hemos encontrado, así que podemos ver los estudios que encontraron un efecto mayor (derecha) o menor (izquierda) que éste. A primera vista no parece haber mucha asimetría, pero es importante tener en cuenta que es un análisis muy subjetivo.

O, si queremos cambiar los títulos de los ejes, por ejemplo escribiéndolos en español, podemos hacerlo agregando los argumentos `xlab` (para el eje X) y/o `ylab` (para el eje Y), como se ve en la Figura 7.

```
funnel(res,
       xlab = "Coeficiente de correlación transformado en z de Fisher",
       ylab = "Error estándar")
```

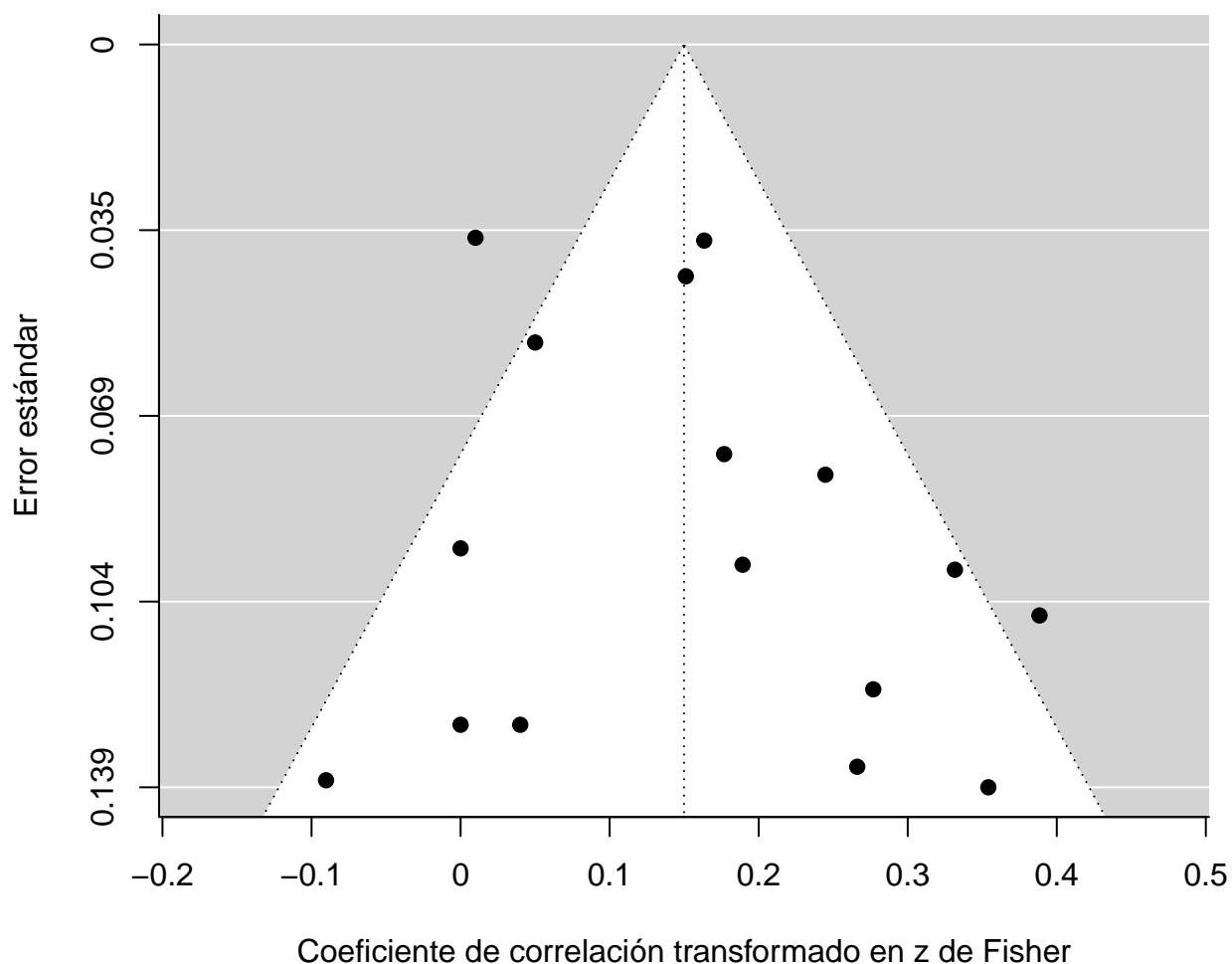


Figura 7. Funnel plot básico de `metafor`, con títulos de ejes en español. Para cada estudio meta-analizado, tenemos el efecto (correlación, en este caso en valores z de Fisher) en el eje X , así como su error estándar en el eje Y . La línea punteada vertical representa el efecto meta-analizado que hemos encontrado, así que podemos ver los estudios que encontraron un efecto mayor (derecha) o menor (izquierda) que éste.

De nuevo, para una versión más sofisticada, se puede usar el paquete `metaviz`, usando la función `viz_funnel`. Hay muchas opciones, pero como ejemplo, usaré la versión por defecto, agregando solo la línea de la regresión de Egger (`egger = TRUE`; ver sección 3.5.2, más adelante), transformando los tamaños de efecto de regreso a r de Pearson (`x_trans_function = tanh`), y con los títulos de los ejes en español (Fig. 8).

```
viz_funnel(res,
           egger = TRUE,
           x_trans_function = tanh,
           ylab = "Error estándar",
           xlab = "Coeficiente de correlación")
```

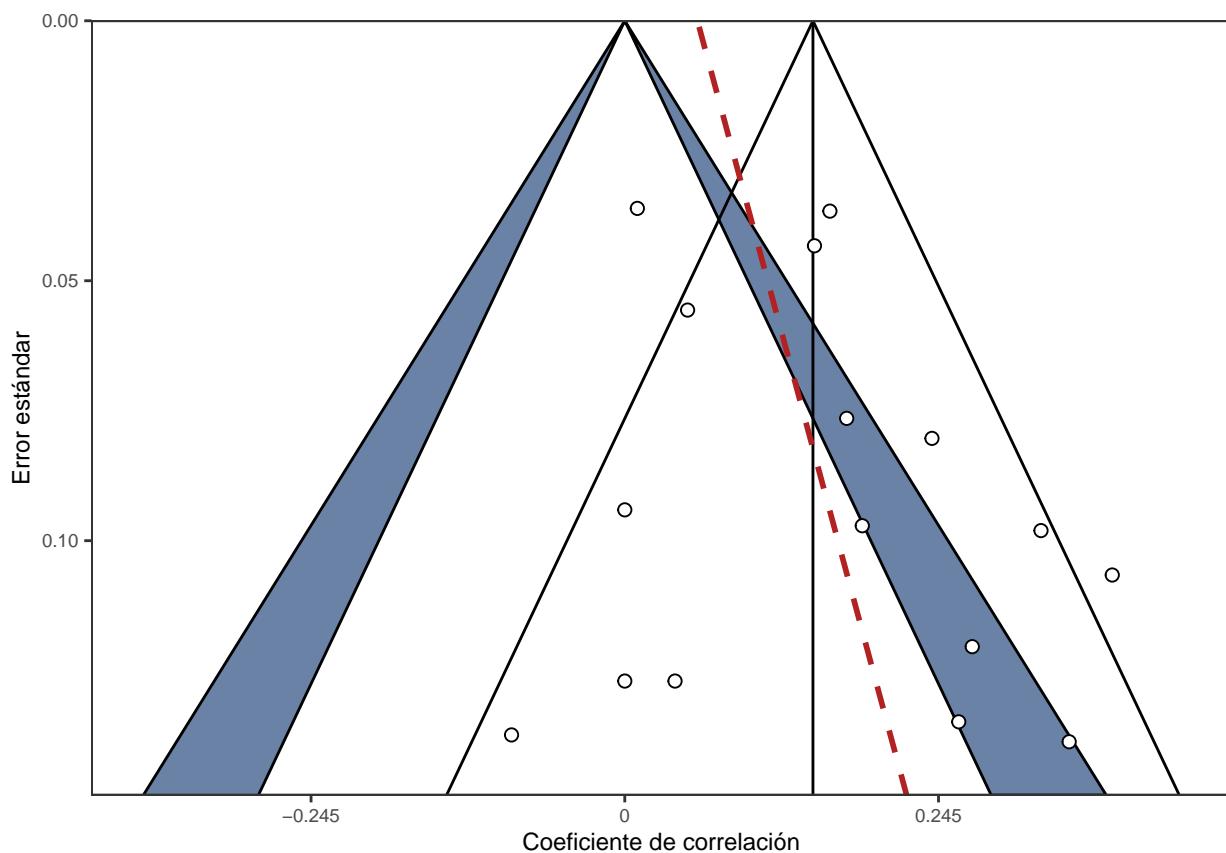


Figura 8. *Funnel plot* creado con `metaviz`. En azul, se representa el área donde estudios, según su error (y su tamaño de muestra), tendrían un efecto significativo al 5% (i.e. $p > 0.05$), y fuera de ésta, donde tendrían un efecto significativo al 1% (i.e. $p > 0.01$). La línea negra vertical representa el efecto meta-analizado, y el triángulo a partir de su inicio, el área donde se ubican los estudios que no se diferencian significativamente del resultado del meta-análisis. La línea roja punteada, representa la regresión de Egger.

Alternativamente, el paquete `metaviz` tiene la función `viz_sunset`, que permite además mostrar el poder estadístico (o potencia) de los estudios meta-analizados para detectar un efecto de interés mediante una prueba de Wald de dos colas. De ser necesario, para entender las bases del poder estadístico, recomiendo ver [esta serie de videos](#) (Leongómez, 2020b) y/o, para mayor profundidad, leer [esta guía](#) (Leongómez, 2020a) que publiqué anteriormente.

A continuación, muestro dos versiones de *funnel plots* creados con la función `viz_sunset` (Fig. 9). En ambos casos, agregué el efecto *real* estimado a partir del meta-análisis (`contours = TRUE`), y transformé los tamaños de efecto de regreso a r de Pearson (`x_trans_function = tanh`).

```
# A. Escala de poder discreta
viz_sunset(res,
            contours = TRUE,
            x_trans_function = tanh)

# B. Escala de poder continua
viz_sunset(res,
            contours = TRUE,
            x_trans_function = tanh,
            power_contours = "continuous")
```

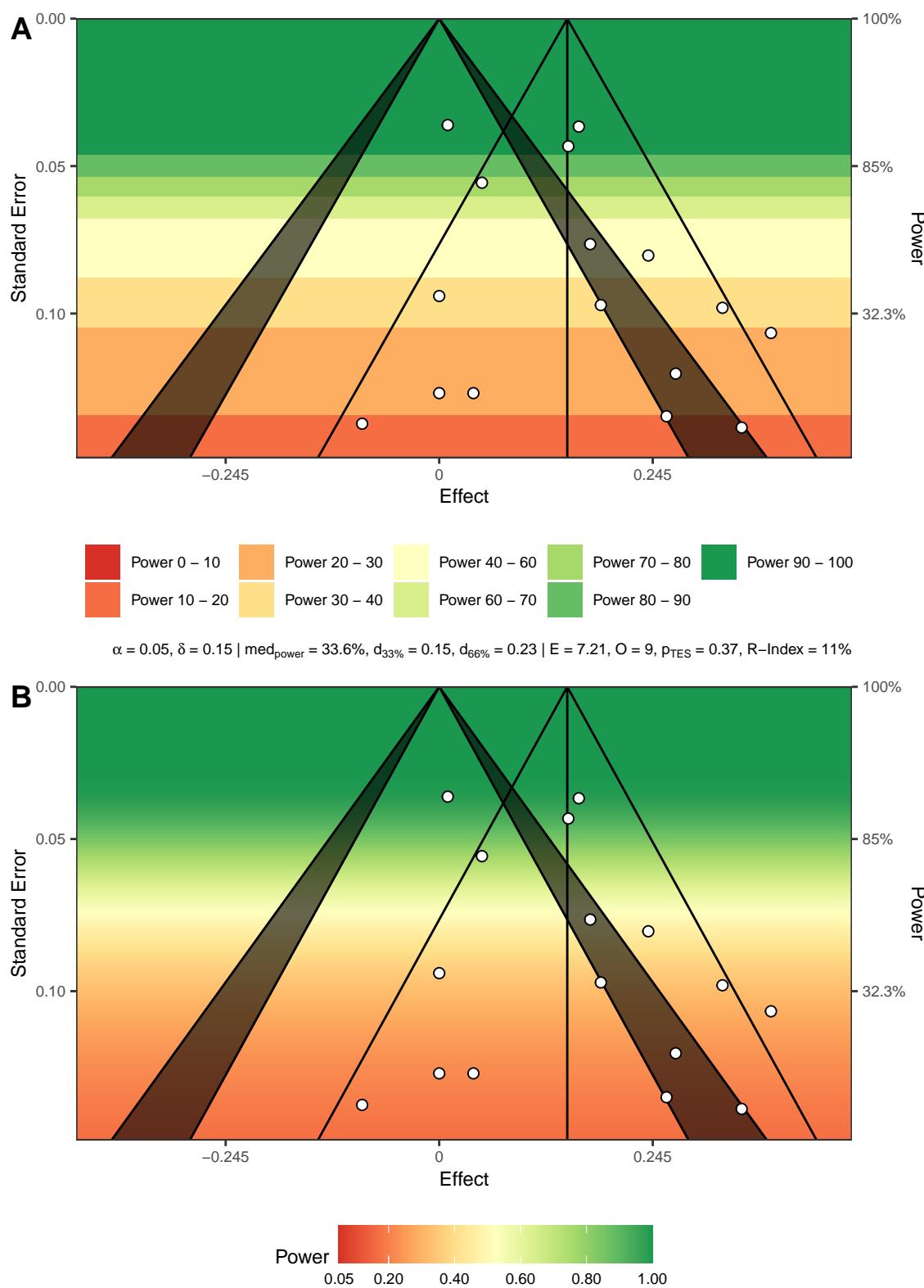


Figura 9. Dos versiones de funnel plot creados con [metaviz](#), usando la función `viz-sunset`, que estima el poder de cada estudio para detectar un efecto de interés. **A.** Poder representado por bandas discretas de color. **B.** Poder representado de manera continua en una escala de color. En ambos casos, y tal como en la Fig. 8, el efecto real está representado como una línea vertical, y el triángulo a partir de su inicio representa el área donde se ubican los estudios que no se diferencian significativamente del resultado del meta-análisis.

3.5.2. Regresión de Egger

Para hacer una prueba formal de sesgo de estudios pequeños (Schwarzer et al., 2015b; Sterne et al., 2000), podemos hacer una prueba o regresión de Egger (Egger et al., 1997). En `metafor`, esto se hace con la función `regtest`, de nuevo usando como argumento el objeto al que asignamos el resultado de nuestro meta-análisis (`res`).

```
regtest(res)
```

Como se puede ver, la prueba de Egger no muestra un resultado significativo ($z = 1.0216$, $p = 0.3070$).

Consola de R: Output 7

```
Regression Test for Funnel Plot Asymmetry

Model: mixed-effects meta-regression model
Predictor: standard error

Test for Funnel Plot Asymmetry: z = 1.0216, p = 0.3070
Limit Estimate (as sei -> 0): b = 0.0790 (CI: -0.0686, 0.2266)
```

Con base en esto, y la inspección visual subjetiva del *funnel plot*, muchos investigadores concluyen que no hay sesgo de publicación. Sin embargo, como mencioné antes, estas pruebas no se centran en el sesgo de publicación sino en el sesgo de estudios pequeños. En otras palabras, con base en esto, lo único que podemos concluir correctamente, es que no hay sesgo de estudios pequeños (más adelante, en la sección 3.6, explicaré cómo evaluar si hay sesgo de publicación).

3.6. Sesgo de publicación (*publication bias*)

Existen diferentes opciones para estimar el sesgo de publicación de un meta-análisis. En esta sección, me centraré en dos: el método de recorte y relleno (*trim and fill*), y la estimación del modelo de función de peso.

3.6.1. Método *trim and fill* (recorte y relleno)

El método de recorte y relleno (*trim and fill*) es una técnica no paramétrica para aumentar datos (Duval & Tweedie, 2000a, 2000b; ver también Duval, 2005). Este método puede utilizarse para estimar y ajustar el número y los resultados de los estudios que faltan en un meta-análisis, con base en el escrutinio de un lado de un diagrama de embudo (*funnel plot*) para estimar la asimetría que se supone que se debe al sesgo de publicación.

Básicamente, el método aumenta los datos observados para que el gráfico de embudo sea más simétrico, y vuelve a calcular la estimación del meta-análisis con base en los datos completos.

Aunque muy útil, es importante tener en cuenta que método de recorte y relleno (*trim and fill*) tiene algunas limitaciones que deben considerarse:

- No debe considerarse como una forma de obtener una estimación más “válida” del efecto o resultado global (a partir del meta-análisis), sino como una forma de examinar qué tan sensibles son los resultados a un mecanismo de selección particular (es decir, una forma particular de sesgo de publicación)
- Es importante tener en cuenta que este método no se puede usar en modelos con moderadores, como los que explico en la sección 4

En `metafor`, el método de recorte y relleno (*trim and fill*) se hace simplemente con la función `trimfill`, de nuevo usando como argumento el objeto al que asignamos el resultado de nuestro meta-análisis (`res`). En este caso, asignaré el resultado de esta función a un objeto que llamaré `tf`.

```
tf <- trimfill(res)
tf
```

Consola de R: Output 8

```
Estimated number of missing studies on the left side: 2 (SE = 2.7118)

Random-Effects Model (k = 18; tau^2 estimator: REML)
```

```

tau^2 (estimated amount of total heterogeneity): 0.0112 (SE = 0.0066)
tau (square root of estimated tau^2 value):         0.1061
I^2 (total heterogeneity / total variability):   67.50%
H^2 (total variability / sampling variability):  3.08

Test for Heterogeneity:
Q(df = 17) = 46.3990, p-val = 0.0002

Model Results:

estimate      se     zval    pval   ci.lb   ci.ub
 0.1288  0.0333  3.8628  0.0001  0.0635  0.1942 ***

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

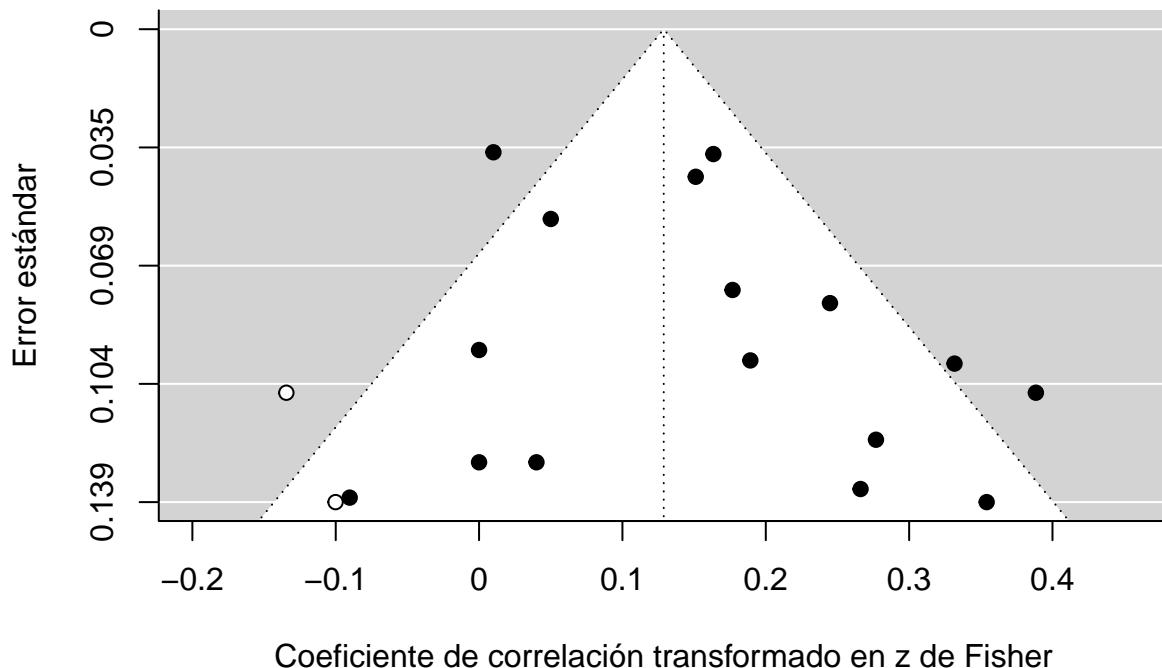
```

Para crear un *funnel plot* con `metafor` del meta-análisis con el método de recorte y relleno (*trim and fill*), solo tenemos que usar la función `funnel`, usando como argumento el objeto al que asignamos los resultados del método de recorte y relleno (`tf`). Con esto, generamos la Figura 10.

```

funnel(tf,
       xlab = "Coeficiente de correlación transformado en z de Fisher",
       ylab = "Error estándar")

```



Coeficiente de correlación transformado en z de Fisher

Figura 10. *Funnel plot* básico de `metafor` usando el método de recorte y relleno (*trim and fill*). En negro los estudios meta-analizados; en blanco, los estudios *rellenados*.

De nuevo, alternativamente podemos usar la función `viz_funnel` del paquete `metaviz`, para crear este *funnel plot* representando los estudios *rellenados* con el método de recorte y relleno. Sin embargo, para esto usaremos como argumento el meta-análisis original (`res`), pero agregando los argumentos `trim_and_fill = TRUE` y `trim_and_fill_side = "left"` ("left" dado que sabemos que los estudios *faltantes* están a la izquierda del *funnel plot*).

```

viz_funnel(res,
            contours_col = "Oranges",
            trim_and_fill = TRUE,
            trim_and_fill_side = "left", #IMPORTANTE

```

```

egger = TRUE,
x_trans_function = tanh,
ylab = "Error estándar",
xlab = "Coeficiente de correlación") +
geom_vline(xintercept = 0, linetype = "dotted")

```

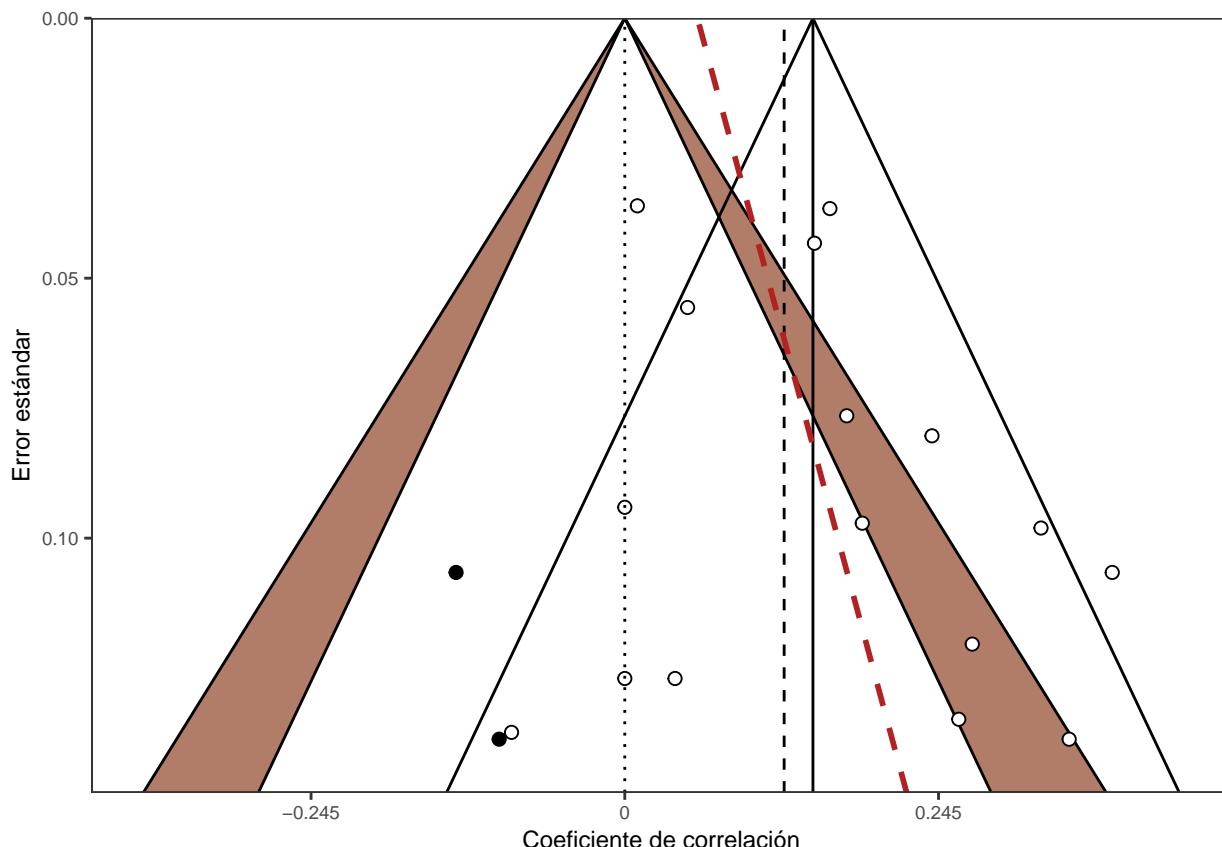


Figura 11. Funnel plot creado con `metaviz` usando el método de recorte y relleno (*trim and fill*). En negro los estudios meta-analizados; en blanco, los estudios *rellenados*.

El resultado nos muestra que el análisis aumentó dos estudios hipotéticos a la izquierda. Es decir, sugiere que hay *cierto* nivel de sesgo de publicación. De hecho, la estimación del efecto al aumentar estos estudios hipotéticos para que la distribución sea realmente simétrica, disminuye a 0.1288, algo menor que el 0.1499 que nos sugería el análisis inicial.

Si quieras profundizar en el método de recorte y relleno (*trim and fill*), su interpretación, así como sus fortalezas y limitaciones, te recomiendo leer el artículo de Shi & Lin (2019) al respecto.

3.6.2. Estimación del modelo de función de peso

Para determinar el sesgo de publicación, se puede usar la función `weightfunct` del paquete `weightr` (Coburn & Vevea, 2019), que nos permite “estimar tanto el modelo de función de peso para el sesgo de publicación que se publicó originalmente en Vevea & Hedges (1995) como la versión modificada presentada en Vevea & Woods (2005)”, como se describe en la [documentación](#) de la función `weightfunct`.

```
library(weightr)
```

En este caso, usaré esta función, asignando el resultado a un objeto que llamaré `wf`.

```
wf <- weightfunct(effect = dat$yi, v = dat$vi, table = TRUE)
wf
```

Consola de R: Output 9

Unadjusted Model (k = 16):

```
tau^2 (estimated amount of total heterogeneity): 0.0070 (SE = 0.0051)
tau (square root of estimated tau^2 value):  0.0834
```

Test for Heterogeneity:

```
Q(df = 15) = 38.1595, p-val = 0.001436053
```

Model Results:

	estimate	std.error	z-stat	p-val	ci.lb	ci.ub
Intercept	0.1486	0.03073	4.835	1.3288e-06	0.08837	0.2088

Adjusted Model (k = 16):

```
tau^2 (estimated amount of total heterogeneity): 0.0056 (SE = 0.0045)
tau (square root of estimated tau^2 value):  0.0750
```

Test for Heterogeneity:

```
Q(df = 15) = 38.1595, p-val = 0.001436053
```

Model Results:

	estimate	std.error	z-stat	p-val	ci.lb	ci.ub
Intercept	0.09153	0.04464	2.050	0.040341	0.00403	0.1790
0.025 < p < 1	0.24121	0.20122	1.199	0.230626	-0.15317	0.6356

Likelihood Ratio Test:

```
X^2(df = 1) = 2.98493, p-val = 0.084043
```

Number of Effect Sizes per Interval:

	Frequency
p-values <0.025	9
0.025 < p-values < 1	7

El modelo tradicional nos da un estimado muy similar (pero no idéntico) al del estudio original (0.1486), dado que usa un método ligeramente diferente.

Como antes, también nos da valores de heterogeneidad τ^2 , τ y Q .

Pero lo más importante, es que nos da los resultado del meta-análisis, ajustando los pesos dados a cada efecto, de cada estudio meta-analizado.

Lo que esta función hace es lo que se conoce como *selection models* (modelos de selección). Básicamente, da más peso a ciertos tamaños de efecto. La realidad de la literatura científica es que es más probable que algunos estudios sean publicados, dependiendo de sus valores p (Coburn & Vevea, 2015; para una descripción sencilla y general, ver por ejemplo “[Acerca del sesgo de publicación en literatura sobre antidepresivos,](#)” 2008) publicados que estudios con $p < 0,05$.

La función `weightfunct` incrementa el peso de estudios que tienen menos probabilidad de ser publicados, y reduce el peso de estudios con mayor probabilidad de ser publicados. Por esto, al usar ésta técnica, estás asumiendo que de hecho, en el efecto que tratas de encontrar en tu meta-análisis, de hecho hay un sesgo de publicación, lo que a menudo es una suposición bastante justa.

Al usar ésta técnica, tenemos un resultado bastante distinto. Mientras que el meta-análisis original nos daba como resultado un efecto de ~ 0.15 , esta técnica nos estima un efecto de ~ 0.09 . Básicamente, ha *encogido* nuestro tamaño de efecto.

Al final el *Likelihood ratio test* (algo así como “Prueba de cociente de probabilidades”), que evalúa la bondad del ajuste de dos modelos estadísticos que compiten entre sí basándose en la relación de su verosimilitud. En este caso, comparando el modelo original, con este modelo con pesos ajustados.

Este resultado nos da una tendencia no descartable ($p\text{-val} = 0.084043$, lo que es < 0.10 ; significativa si asumimos un análisis de una cola), que nos da evidencia de que en efecto hay un sesgo de publicación, a pesar de que el *funnel*

plot (Figs. 6, 7 8 y 9) y la regresión de Egger (sección 3.5.2), sugerían lo contrario.

3.7. Poder estadístico del meta-análisis

En esta sección explicaré cómo hacer un análisis de poder de un meta-análisis; la idea de ésto es saber si nuestro meta-análisis tiene un poder suficiente para detectar el efecto meta-analizado (en nuestro caso 0.15 para el meta-análisis original “*res*”, o 0.09 el meta-análisis con pesos ajustados “*wf*”). Para este ejemplo, asumiré que el efecto *real* es el encontrado en nuestro análisis original (0.15), pues este efecto es mayor. Si no tuviésemos el poder suficiente para detectar confiablemente ese efecto, menos lo tendríamos para un efecto menor, como el detectado en nuestro meta-análisis con pesos ajustados.

Para hacer esto, usaré el paquete **metameta** (Quintana, 2022, *in press*), que permite calcular y visualizar el poder estadístico de un meta-análisis para detectar un rango de posibles efectos *reales*.

3.7.1. Instalación de **metameta**

El paquete **metameta** se debe instalar desde GitHub⁸ ya que, al día de hoy, no está aún disponible en CRAN.

Para esto, debemos tener instalado el paquete **devtools**, y usar la función **install_github** que nos permite instalar paquetes directamente desde GitHub.

```
#se debe tener instalado el paquete devtools
library(devtools)
install_github("dsquintana/metameta")
```

3.7.2. Análisis de poder

Una vez instalado, podemos cargar el paquete.

```
library(metameta)
```

Como datos, necesitamos no solamente los tamaños de efecto a meta-analizar (*r* de Pearson transformado a *z* de Fisher), sino además los intervalos de confianza, tal como fueron reportados en varios de nuestros *Forest plots*.

En este caso, voy a asumir un *efecto real* de *r* = 0,15, tal como en nuestro meta-análisis original. Sin embargo, el *efecto real* no es algo que podamos saber (es, de hecho, lo que queremos acercarnos a conocer a través del meta-análisis), así que la función **mapower_ul** del paquete **metameta** calcula el poder de cada meta-análisis para un rango de posibles efectos reales.

```
dat.power <- summary(dat) %>%
  select(yi, ci.lb, ci.ub) %>%
  rename(lower = ci.lb, upper = ci.ub)

power <- mapower_ul(dat = dat.power, observed_es = 0.15, name = "Molloy et al. 2014")

power_list <- list(power$power_median_dat)
power_dat <- power$dat

power_list
```

Consola de R: Output 10

```
[1]
power_es_observed      es01      es02      es03      es04      es05      es06
1          0 0.1762492 0.5356629 0.8673657 0.9837594 0.9992182 0.9999858
           es07 es08 es09 es1 es_observed meta_analysis_name
1 0.9999999    1    1    1  0.3364197 Molloy et al. 2014
```

⁸GitHub es un repositorio abierto para proyectos de código abierto en el que, entre otras cosas, suelen estar alojados todos los paquetes de R incluso en versiones de desarrollador. Por supuesto, a diferencia de CRAN, GitHub es un repositorio de código general, no específico para paquetes de R.

```
power_dat
```

Consola de R: Output 11

	yi	lower	upper	sei	power_es_observed	power_es01
1	0.18922664	-0.001141888	0.37959517	0.09712680	0.3390835	0.1774836
2	0.16343992	0.091680539	0.23519930	0.03661193	0.9837035	0.7797623
3	0.35409253	0.082294427	0.62589063	0.13867250	0.1910727	0.1113715
4	0.33164711	0.139456828	0.52383739	0.09805627	0.3337559	0.1750148
5	0.27686382	0.040911886	0.51281576	0.12038364	0.2383028	0.1320187
6	0.00000000	-0.248915675	0.24891567	0.12699779	0.2188783	0.1234921
7	0.17682002	0.026937828	0.32670221	0.07647051	0.5006730	0.2576556
8	0.05004173	-0.059013584	0.15909704	0.05564047	0.7691106	0.4354589
9	0.26610841	0.001826917	0.53038990	0.13483749	0.1994161	0.1149996
10	0.01000033	-0.060723775	0.08072444	0.03608373	0.9859910	0.7914240
11	-0.09024419	-0.359465946	0.17897757	0.13735804	0.1938536	0.1125800
12	0.38842310	0.179490237	0.59735596	0.10659840	0.2905754	0.1553001
13	0.00000000	-0.184377902	0.18437790	0.09407036	0.3575910	0.1861296
14	0.15114044	0.066324499	0.23595637	0.04327344	0.9340134	0.6371865
15	0.24477411	0.087346017	0.40220221	0.08032046	0.4632371	0.2379950
16	0.04002135	-0.208894321	0.28893703	0.12699779	0.2188783	0.1234921
	power_es02	power_es03	power_es04	power_es05	power_es06	power_es07
1	0.5395394	0.8705053	0.9845502	0.9992836	0.9999877	0.9999999
2	0.9997697	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
3	0.3026621	0.5806100	0.8223954	0.9500825	0.9910281	0.9989921
4	0.5317864	0.8642261	0.9829686	0.9991528	0.9999840	0.9999999
5	0.3827659	0.7026651	0.9135189	0.9858616	0.9987531	0.9999421
6	0.3502733	0.6562694	0.8829174	0.9759855	0.9971497	0.9998088
7	0.7439048	0.9751837	0.9994638	0.9999977	1.0000000	1.0000000
8	0.9489275	0.9997002	0.9999999	1.0000000	1.0000000	1.0000000
9	0.3170763	0.6044851	0.8429297	0.9597856	0.9936099	0.9993842
10	0.9998300	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
11	0.3074782	0.5886809	0.8294859	0.9535367	0.9919839	0.9991443
12	0.4666848	0.8035418	0.9634686	0.9968384	0.9998781	0.9999980
13	0.5659845	0.8904902	0.9890522	0.9996035	0.9999950	1.0000000
14	0.9961139	0.9999997	1.0000000	1.0000000	1.0000000	1.0000000
15	0.7019697	0.9620572	0.9987365	0.9999900	1.0000000	1.0000000
16	0.3502733	0.6562694	0.8829174	0.9759855	0.9971497	0.9998088
	power_es09	power_es1				
1	1.0000000	1.0000000				
2	1.0000000	1.0000000				
3	0.9999971	0.9999999				
4	1.0000000	1.0000000				
5	1.0000000	1.0000000				
6	0.9999999	1.0000000				
7	1.0000000	1.0000000				
8	1.0000000	1.0000000				
9	0.9999988	1.0000000				
10	1.0000000	1.0000000				
11	0.9999978	0.9999999				
12	1.0000000	1.0000000				
13	1.0000000	1.0000000				
14	1.0000000	1.0000000				
15	1.0000000	1.0000000				
16	0.9999999	1.0000000				

3.7.3. Visualización del análisis de poder (*Firepower plot*)

Según esto, nuestro meta-análisis solo tiene un poder estadístico suficiente para detectar de manera confiable efectos mayores a ~0.3, lo que está muy por encima de nuestras estimaciones del efecto real (0.15 en nuestro meta-análisis original, 0.09 en nuestro meta-análisis con pesos ajustados).

```
power.plot <- firepower(power_list)
```

Para ver el *fireplot* que creamos, y ya que lo asigné a un objeto que llamé `power.plot`, debo correr el objeto para ver el resultado (Fig. 12).

power.plot

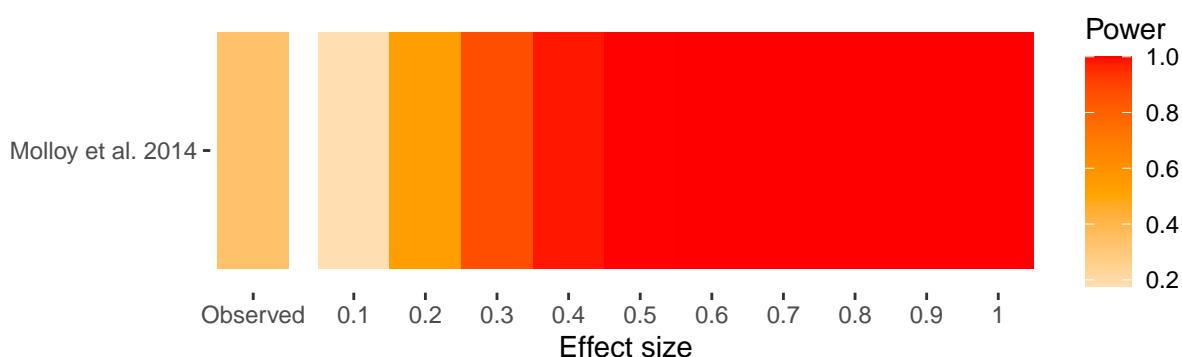


Figura 12. Fireplot básico de [metameta](#), para un análisis de poder de nuestro meta-análisis. *Observed* hace referencia al tamaño de efecto observado en nuestro meta-análisis original; en este caso, 0.15.

Si queremos cambiar los títulos a español, y ya que el objeto `power.plot` contiene dos elementos (`dat` y `fp_plot`, que es propiamente la gráfica). Éste último elemento es de clase `ggplot`, por lo que podemos usar funciones de `ggplot2` para cambiar, por ejemplo, el título del eje *X* a “Tamaño de efecto”, el título de la leyenda a “Poder”, y el efecto observado de “*Observed*” a “Observado”⁹ (Fig. 13). Por ejemplo:

```
power.plot$fp_plot +
  xlab("Tamaño de efecto") +
  guides(fill = guide_legend(title = "Poder",
                             reverse = TRUE)) +
  scale_x_discrete(labels = c("es_observed" = "Observado",
                             "es01" = 0.1,
                             "es02" = 0.2,
                             "es03" = 0.3,
                             "es04" = 0.4,
                             "es05" = 0.5,
                             "es06" = 0.6,
                             "es07" = 0.7,
                             "es08" = 0.8,
                             "es09" = 0.9,
                             "es1" = 1))
```

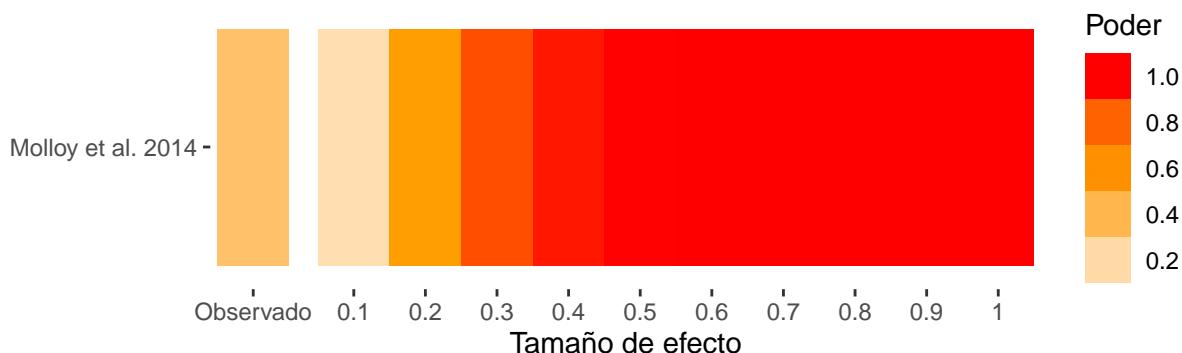


Figura 13. Fireplot básico de [metameta](#), para un análisis de poder de nuestro meta-análisis, con el texto traducido a español y con la leyenda en una escala discreta para facilitar su lectura. *Observado* hace referencia al tamaño de efecto observado en nuestro meta-análisis original (en este caso, 0.15).

⁹Para cambiar el título del eje *X* usé la función `xlab`; para el título de la leyenda la función `guides` (opción `fill = guide_legend`); y para los valores del eje *X*, la función `scale_x_discrete`.

3.8. TODAS LAS COMBINACIONES POSIBLES

Ver acá <https://www.dsquintana.blog/combinatorial-meta-analysis/>

```
l_one <- leave1out(res)
l_one
```

Consola de R: Output 12

	estimate	se	zval	pval	ci.lb	ci.ub	Q	Qp	tau2	I2
1	0.1483	0.0336	4.4157	0.0000	0.0825	0.2142	37.7109	0.0006	0.0091	64.9684
2	0.1500	0.0354	4.2355	0.0000	0.0806	0.2194	36.7672	0.0008	0.0100	62.2717
3	0.1418	0.0315	4.5064	0.0000	0.0801	0.2035	35.3930	0.0013	0.0075	61.1741
4	0.1379	0.0308	4.4739	0.0000	0.0775	0.1984	33.5886	0.0024	0.0068	58.1050
5	0.1441	0.0323	4.4550	0.0000	0.0807	0.2075	36.5396	0.0009	0.0082	62.8995
6	0.1566	0.0326	4.8119	0.0000	0.0928	0.2205	37.1703	0.0007	0.0084	63.5769
7	0.1489	0.0342	4.3555	0.0000	0.0819	0.2160	37.6797	0.0006	0.0094	65.1901
8	0.1599	0.0334	4.7841	0.0000	0.0944	0.2254	36.1484	0.0010	0.0084	61.4011
9	0.1455	0.0324	4.4863	0.0000	0.0819	0.2091	37.0495	0.0007	0.0083	63.4803
10	0.1647	0.0291	5.6601	0.0000	0.1077	0.2218	25.0661	0.0339	0.0047	43.6587
11	0.1586	0.0317	5.0069	0.0000	0.0965	0.2207	35.6617	0.0012	0.0077	61.6732
12	0.1353	0.0296	4.5647	0.0000	0.0772	0.1934	31.9021	0.0041	0.0059	55.0102
13	0.1592	0.0324	4.9100	0.0000	0.0956	0.2227	36.3291	0.0009	0.0080	62.0683
14	0.1514	0.0353	4.2858	0.0000	0.0821	0.2206	37.7339	0.0006	0.0100	63.6942
15	0.1432	0.0330	4.3432	0.0000	0.0786	0.2078	35.8385	0.0011	0.0083	62.6114
16	0.1552	0.0329	4.7203	0.0000	0.0908	0.2196	37.7017	0.0006	0.0087	64.3368
			H2							
1	2.8546									
2	2.6505									
3	2.5756									
4	2.3869									
5	2.6954									
6	2.7455									
7	2.8727									
8	2.5907									
9	2.7382									
10	1.7749									
11	2.6091									
12	2.2227									
13	2.6363									
14	2.7544									
15	2.6746									
16	2.8040									

3.8.1. Combinatorial meta-analysis

The number of analyses to be conducted can be calculated using the formula $2^k - 1$, with k representing the number of studies. So for a meta-analysis with 16 studies, you will perform 65535 meta-analyses, that is $2^{16} - 1$.

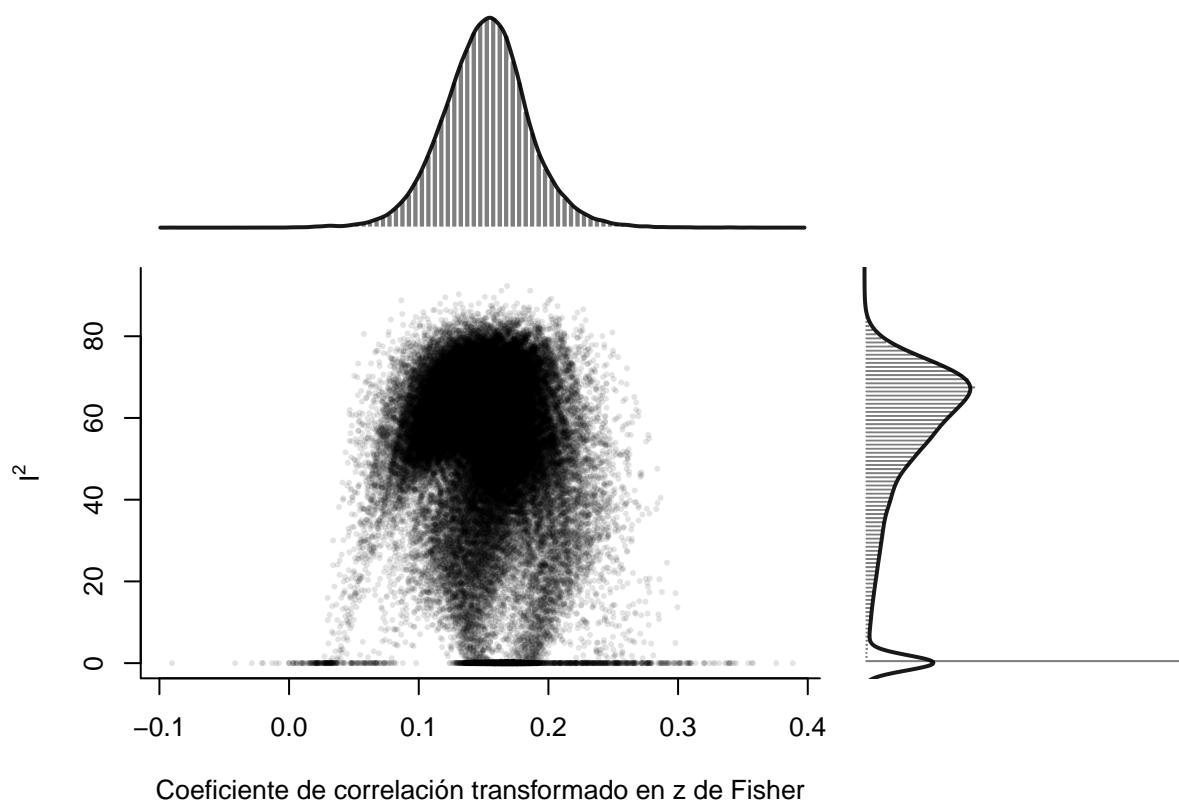
To visualise this enormous number of summary effect size estimates, we'll put together a Graphical Display of Study Heterogeneity (GOSH) plot, using the following script, but before running the script, I should warn you that this could take some time to create 65535 models (it took about 10 minutes on my 2015 MacBook Pro).

```
gp <- gosh(res)
```

Fitting 65535 models (based on all possible subsets).

Here's a visualisation of the summary effects of 65535 models (horizontal axis), plotted against the model's heterogeneity (vertical axis).

```
plot(gp, breaks = 100,
      labels = c("Coeficiente de correlación transformado en z de Fisher",
                expression(paste("I"^-2))))
```



By extracting the list of effect sizes, it's straightforward to calculate the mean effect size, which is 0.15.

```
mgosh <- gp$res
mean_est <- mean(mgosh$estimate, na.rm = TRUE)
```

The effect sizes range from about 0.05 to 0.25, demonstrating how the choice of included studies influences the summary effect size estimate.

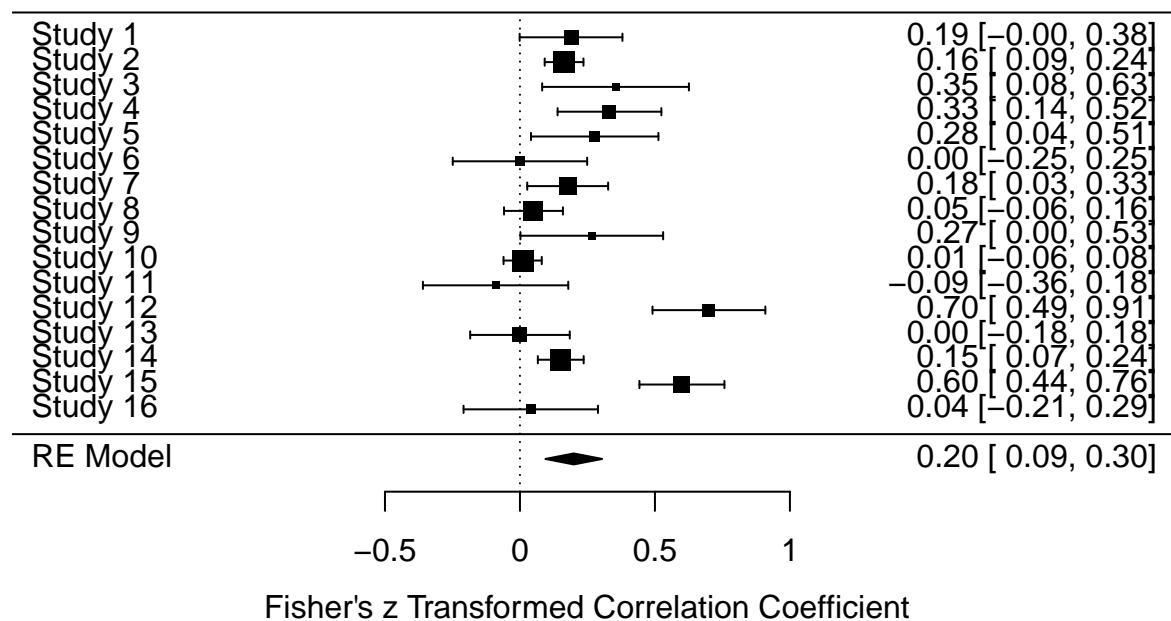
Now let's see what happens when we add in some artificial outliers, using this script:

```
dat_mo <- get(data(dat.molloy2014))
dat_mo <- escalc(measure = "ZCOR",
                 ri = ri, ni = ni, data = dat_mo)

dat_mo[12,11] = 0.7
dat_mo[15,11] = 0.6

res_mo <- rma.uni(yi, vi, data=dat_mo)

forest(res_mo)
```



You can see the introduced outliers in the forest plot (studies 12 and 15).

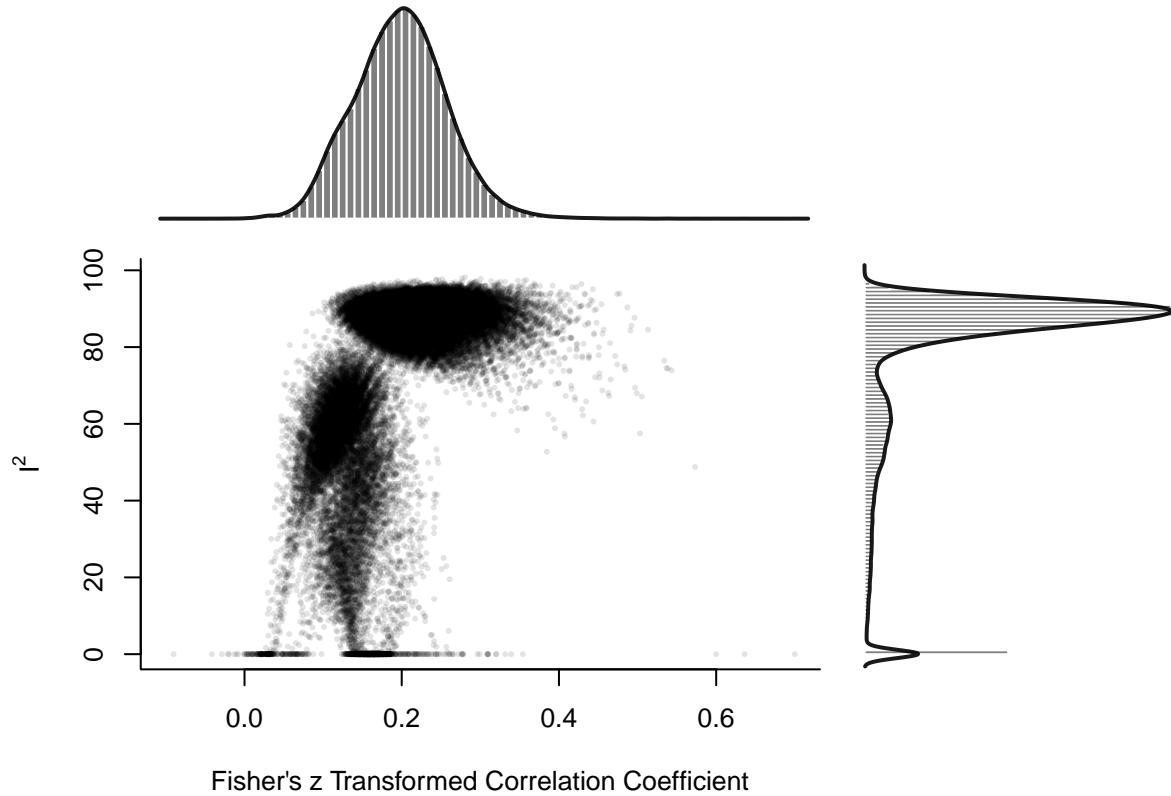
Now let's run the combinatorial meta-analysis and create a GOSH plot:

```
gp_mo <- gosh(res_mo)
```

Fitting 65535 models (based on all possible subsets).

Y el plot

```
plot(gp_mo, breaks = 100)
```



There is a clear cluster of large effect sizes with high heterogeneity in the top right of the plot. We can specifically

highlight the meta-analysis models that included the potential outlier studies:

```
plot(gp_mo, out = 12, breaks = 100,
     labels = c("Coeficiente de correlación transformado en z de Fisher",
               expression(paste("I"^(2))))
```

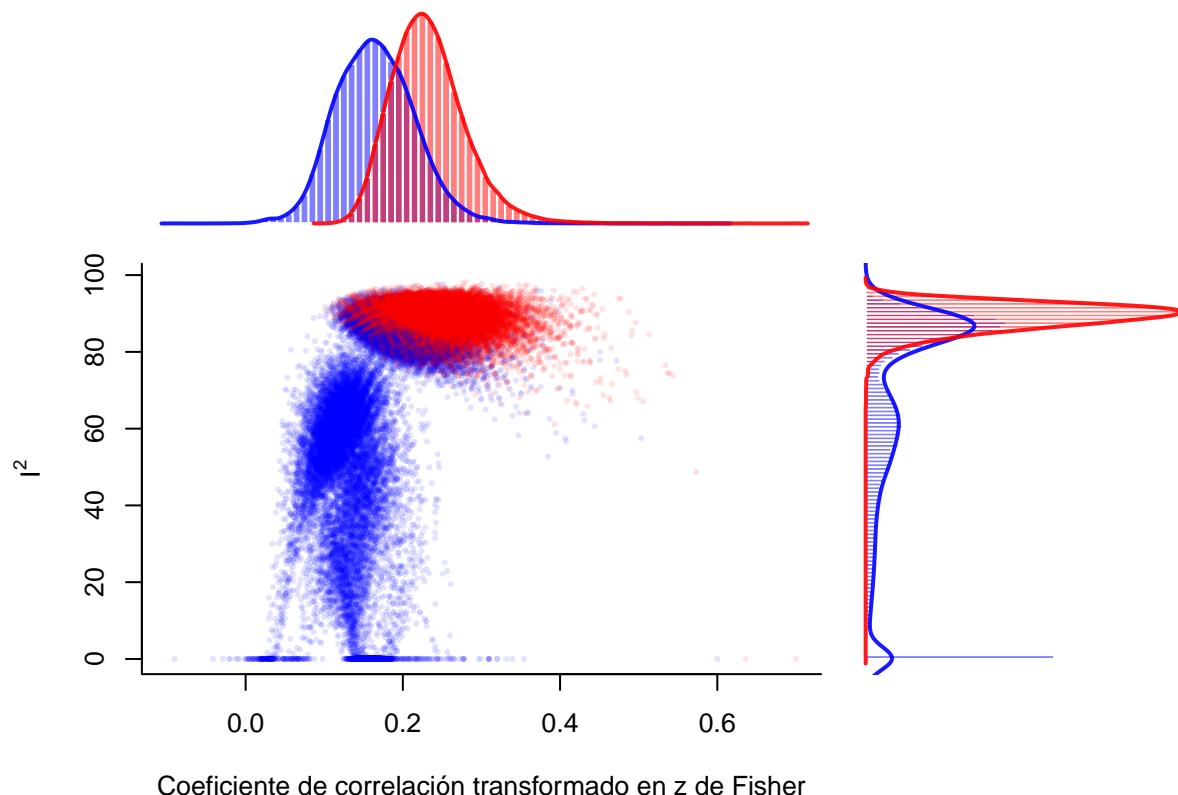


Figura 14. Se puede ver en las distribuciones del eje X (distribuciones arriba) cómo los meta análisis que incluyen ese estudio tienden a arrojar estimados más fuertes para la asociación, sesgando el resultado hacia valores más extremos.

There's a noticeable shift in the distribution of meta-analysis model summary effect size estimates that include study 12.

Ultimately, combinatorial meta-analysis paired with GOSH plots demonstrate how your analysis choices influence your meta-analysis outcomes and conclusions.

4. Meta-análisis de correlación con moderador (meta-regresión)

Incluir moderadores en nuestro meta-análisis (i.e. hacer una meta-regresión), es una fusión de principios de regresión meta-analíticos y lineales, que sirva para explorar la heterogeneidad. Con esto, podemos saber si existe una asociación lineal entre el resultado de nuestro meta-análisis y una o más covariables.

La meta-regresión desempeña un papel fundamental en la consideración de los efectos de las covariables, especialmente en presencia de variables categóricas que pueden utilizarse para el análisis de subgrupos.

4.1. Ejemplo 1: Moderación de una variable continua (edad promedio de los participantes)

Primero, y como ejemplo, vamos a ver si la edad promedio de los participantes de un estudio (en nuestros datos, `meanage`) modera el resultado. Esto es importante, pues hay una enorme variación entre las edades medias de los participantes de los diferentes estudios¹⁰, lo que podría moderar (afectar) la asociación entre concienciación

¹⁰De hecho, mientras que en el estudio de Axelsson et al. (2009) la edad promedio fue de 22, en el estudio de Jerant et al. (2011) la edad promedio fue de 78.6.

(*conscientiousness*) y adherencia a la medicación prescrita.

Para esto, de nuevo podemos usar la función `rma` de paquete `metafor` y de la misma manera que en la sección 3, pero agregando nuestra variable moderadora (`meanage`) al argumento `mods`. En este caso voy a asignar a un objeto llamado `res.modage`, para diferenciarlo del objeto `res` al que asigné el meta-análisis básico, sin moderadores.

```
res.modage <- rma(yi = yi, vi = vi, mods = ~meanage, data = dat)
```

Los resultados, son los siguientes:

```
res.modage
```

Consola de R: Output 13

```
Mixed-Effects Model (k = 16; tau^2 estimator: REML)

tau^2 (estimated amount of residual heterogeneity): 0.0072 (SE = 0.0054)
tau (square root of estimated tau^2 value): 0.0846
I^2 (residual heterogeneity / unaccounted variability): 56.50%
H^2 (unaccounted variability / sampling variability): 2.30
R^2 (amount of heterogeneity accounted for): 11.76%

Test for Residual Heterogeneity:
QE(df = 14) = 30.9050, p-val = 0.0057

Test of Moderators (coefficient 2):
QM(df = 1) = 1.4286, p-val = 0.2320

Model Results:

estimate      se      zval     pval    ci.lb    ci.ub
intrcpt   0.2741  0.1090  2.5147  0.0119  0.0605  0.4877  *
meanage   -0.0024  0.0020 -1.1952  0.2320 -0.0063  0.0015

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

Los resultados, que tienen la misma organización que los del análisis sin moderadores (sección 3). Este resultado nos muestra que, a pesar de la gran diferencia de edad entre estudios, la edad no tiene un efecto significativo, como se puede ver en la sección “Test of Moderators (coefficient 2)” (al final nos muestra el valor *p* como “*p-val* = 0.2320”), así como los resultados para el efecto de `meanage` en la tabla `Model Results` (donde nos da el mismo resultado: “0.2320”).

4.1.1. Más información e interpretación de la moderación

Para más información, podemos *predecir* el efecto (en este caso, la correlación entre la concienciación y la adherencia a la medicación), a diferentes edades, usando la función `predict`.

En este ejemplo, como argumentos de esta función incluiré el objeto que contiene el meta-análisis con la edad como variable moderadora (`res.modage`), y los los valores para los cuales quiero saber el coeficiente de correlación predicho (argumento `newmods`); en este ejemplo voy a usar “`seq(20, 80, by = 10)`”, lo que produce la secuencia de números de 20 a 80, cada 10 (es decir: **20, 30, 40, 50, 60, 70, 80**), que serán las edades para las cuales obtendré el coeficiente de correlación estimado por el modelo¹¹.

Adicionalmente, voy a convertir esta tabla a un objeto clase `data.frame` (`as.data.frame()`), crearé una nueva variable llamada `meanage` que contiene las edades para las cuales he calculado la predicción (`mutate(meanageage = seq(20, 80, by = 10))`¹²), y reorganizado el orden de las columnas (`select(7, 1:6)`). Asignaré esta tabla a un objeto llamado `pred.res.modage`.

¹¹Escogí edades entre 20 y 80 como referencia, pues están cerca al rango de valores de las edades promedio de los estudios meta-analizados (22 a 78.6), y con edades cada 10 años, para obtener un número manejable pero informativo de predicciones.

¹²Para agregar los valores de la variable moderadora para los cuales se hace la predicción, alternativamente se puede agregar el argumento `addx = TRUE` a la función `predict` (en este caso, por ejemplo, `predict(res.modage, newmods = seq(20, 80, by = 10), addx = TRUE)`). Sin embargo, para evitar crear columnas adicionales y tener una tabla en un formato más sencillo y claro, he decidido crear la columna `meanage` manualmente. Si usara el argumento `addx = TRUE`, crearía las columnas `X.intcpt` con el intercepto, y `X.meanagecon` los valores de `meanage`.

```
# Calcular efecto ajustado para diferentes edades
pred.res.modage <- predict(res.modage, newmods = seq(20, 80, by = 10)) %>%
  as.data.frame() %>%
  mutate_all(~round(., 3)) %>%
  mutate(meanage = seq(20, 80, by = 10)) %>%
  select(7, 1:6)
# Ver la tabla
pred.res.modage
```

Consola de R: Output 14

	meanage	pred	se	ci.lb	ci.ub	pi.lb	pi.ub
1	20	0.226	0.071	0.086	0.366	0.009	0.443
2	30	0.202	0.054	0.097	0.307	0.006	0.398
3	40	0.178	0.039	0.102	0.254	-0.005	0.360
4	50	0.154	0.031	0.094	0.214	-0.022	0.330
5	60	0.130	0.034	0.063	0.197	-0.049	0.309
6	70	0.106	0.047	0.013	0.198	-0.084	0.296
7	80	0.082	0.064	-0.043	0.207	-0.126	0.290

Con esto, el objeto `pred.res.modage`, al que he asignado esta predicción, tiene las siguientes columnas:

- **meanage**: edad promedio de los participantes de los estudios meta-analizados
- **pred**: valor del efecto predicho (en este caso, coeficiente de correlación transformado en z de Fisher), para cada edad promedio
- **se**: error estándar (en inglés *standard error*) del efecto predicho
- **ci.lb**: límite inferior del intervalo de confianza del 95 % (en inglés *confidence interval lower bound*) del efecto predicho
- **ci.ub**: límite superior del intervalo de confianza del 95 % (en inglés *confidence interval upper bound*) del efecto predicho
- **pi.lb**: límite inferior del intervalo de predicción (en inglés *prediction interval lower bound*), y
- **pi.ub**: límite superior del intervalo de predicción (en inglés *prediction interval upper bound*)

Las columnas `age` (edad) y `pred` (efecto predicho), nos muestran que, para estudios donde los participantes son en promedio más jóvenes, se esperaría una correlación más fuerte; por ejemplo, mientras que para estudios con personas de 20 años de edad en promedio se esperaría una correlación de $z = 0.226$, para estudios participantes con una edad promedio de 80 años, el modelo estima un efecto de $z = 0.082$. En otras palabras, en nuestra muestra de estudios, la asociación entre concienciación y adherencia a la medicación tiende a reducirse ligeramente con la edad promedio de la muestra.

Sin embargo, este efecto moderador de la edad no es significativo, por lo que no tenemos evidencia suficiente que nos sugiera que es real, más allá de nuestra muestra de estudios.

4.1.1.1. Meta-Analytic Scatter Plot (Gráfico de dispersión meta-analítico) Alternativamente, cuando tenemos un modelo con moderador, también podemos ver la asociación entre la variable moderadora, y el efecto de cada estudio meta-analizado, a modo de regresión. La función `regplot` hace precisamente esto (Fig. 15).

```
regplot(res.modage,
        ylab = "Coeficiente de correlación transformado en z de Fisher",
        xlab = "Edad promedio del estudio")
```

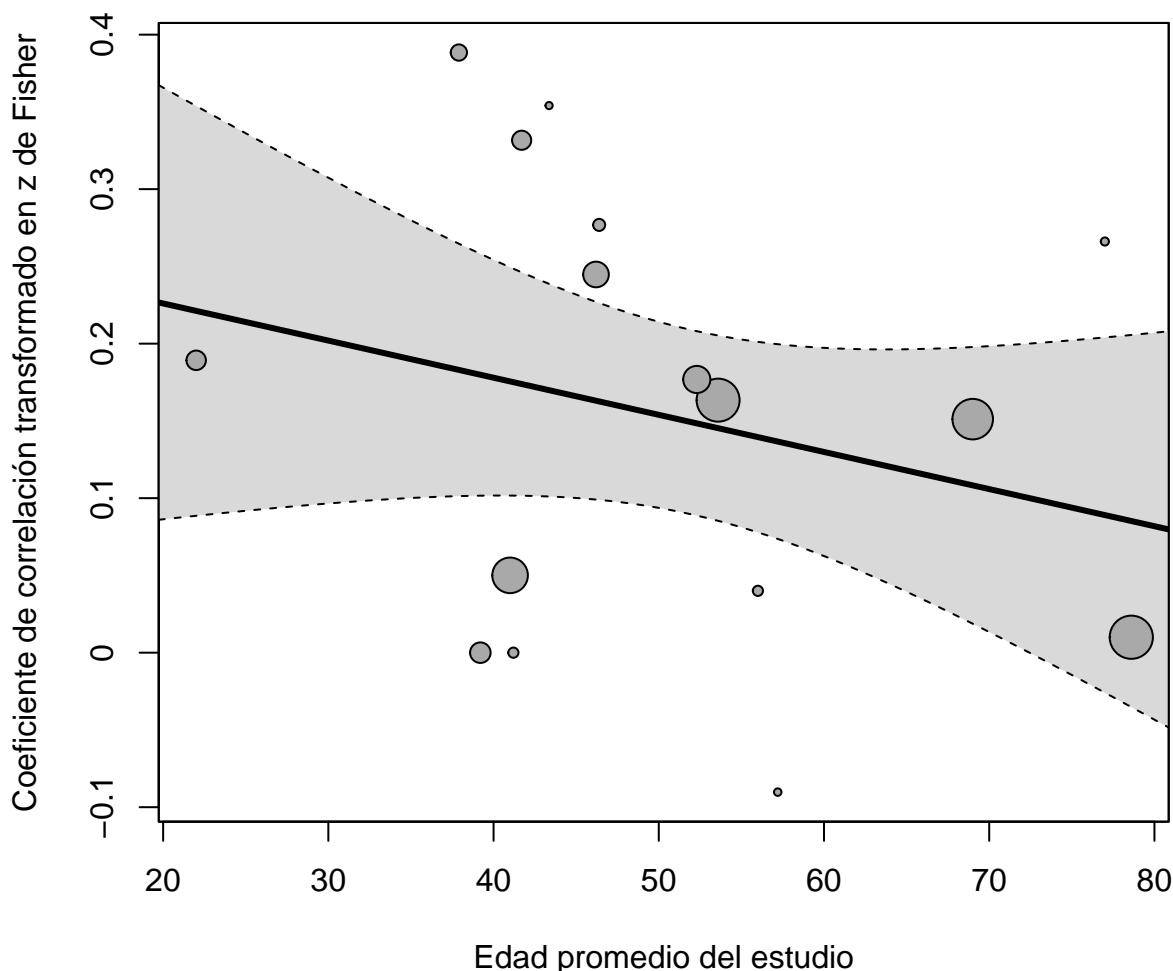


Figura 15. Gráfico de dispersión meta-analítico (*Meta-Analytic Scatter Plot*). El tamaño de los puntos es proporcional al peso que recibieron los estudios en el meta-análisis (puntos más grandes para los estudios con más peso, pues tienen un tamaño de muestra mayor y con un menor error estimado). La línea negra representa el efecto previsto en función del predictor (en este caso `meanage`, edad promedio), que por supuesto coincide con las predicciones del objeto `pred.res.modage` (Output 11, en la sección 4.1.1); la banda gris demilimata por líneas punteadas representa el intervalo de confianza del 95 %.

4.1.2. Forest plot y funnel plot

Por supuesto, de estos resultados también puedo crear *forest plots* y *funnel plots*, siguiendo los ejemplos y código de la sección 3. Para el *forest plot*, hago a continuación un ejemplo anotado y mejorado¹³ (Fig. 16, con un código similar al usado como ejemplo en la Fig. 4). Sin embargo, es importante tener en cuenta que esta opción no creará un resumen del meta-análisis, ya que no tenemos un solo efecto *real* como producto del meta-análisis.

```
# forest plot con anotaciones adicionales
forest(res.modage, cex = 0.75, xlim = c(-1.6, 1.6),
       slab = paste(dat$authors, dat$year, sep = ", "),
       showweights = TRUE,
       xlab = "Coeficiente de correlación transformado en z de Fisher",
       digits = c(2,3L))
# agregar encabezados a las columnas (valores de X y Y deben ser ajustados)
par(cex = 0.8, font = 2)
text(x = -1.6, y = 18, labels = "Autor(es), Año", pos = 4)
text(x = 0, y = 18, labels = "Efecto e IC", pos = 4)
```

¹³La función `viz_forest` del paquete `metaviz` no puede crear un *forest plot* de un meta-análisis ajustado con `metafor` cuando este modelo contiene variables moderadoras continuas, o cuando tiene como variable moderadora más de una variable categórica. En otras palabras, solo podrá hacer el *forest plot* si nuestro meta-análisis no tiene moderadores, o tiene un único moderador categórico. Como en este ejemplo nuestro moderador es una variable continua, no es posible usar esta función.

```
text(x = 1, y = 18, labels = "Peso", pos = 2)
text(x = 1.6, y = 18, labels = "Corr. [95% IC]", pos = 2)
```

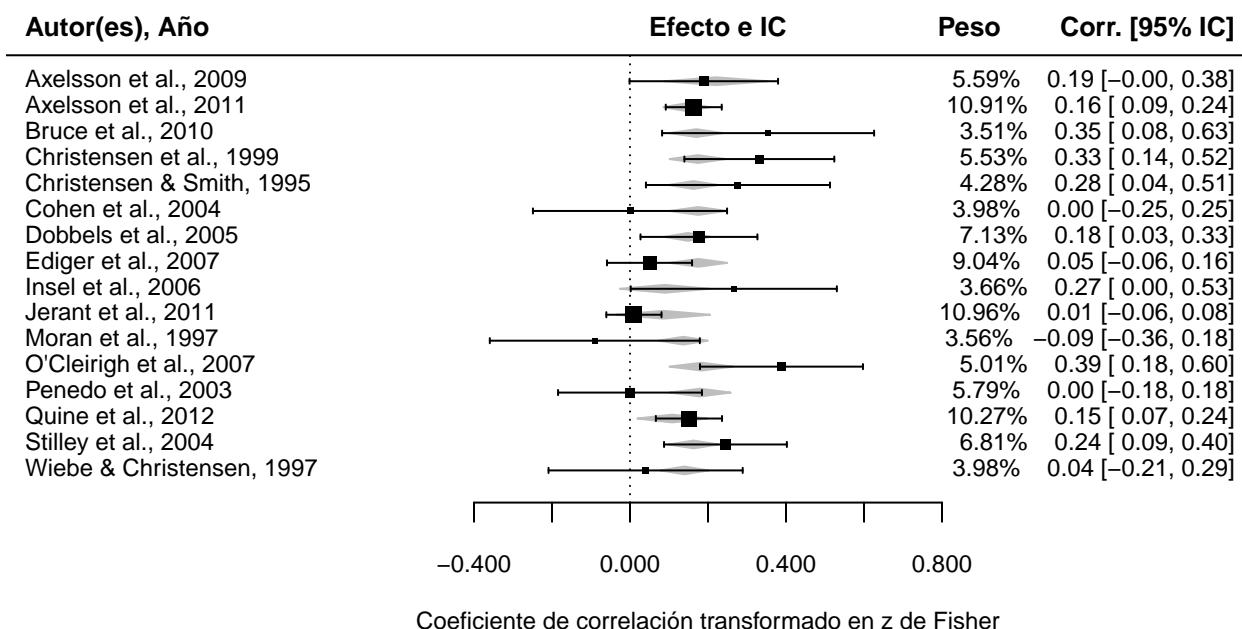


Figura 16. Forest plot básico de `metafor`, para un meta-análisis incluyendo la edad promedio de los participantes como moderador. En la ilustración gráfica, además de los efectos originales, se puede ver el efecto de cada estudio estimado cuando se incluye el moderador como polígonos (diamantes) de color gris. Sin embargo, ya no obtenemos una fila al final representando el efecto promediado del meta-análisis, ya que no tenemos un solo efecto.

De manera similar, podemos obtener un *funnel plot* de nuestro meta-análisis con moderador. Es importante tener en cuenta que tanto el paquete `weightr` como el paquete `metafor` tienen funciones llamadas `funnel`. Dado que cargamos el paquete `weightr` después, R por defecto intentará intentará crear el *funnel plot* con la función `funnel` del paquete `weightr`, en vez de la función `funnel` del paquete `metafor`. Para evitar este error, tenemos dos opciones: podemos pedirle a R explícitamente que use la función `funnel` del paquete `metafor` con el comando `metafor::funnel`, o podemos pedirle a R que elimine el paquete de la memoria con la función `detach("package:metameta", unload = TRUE)`. En este caso, usará la primera opción.

Este *funnel plot*, a diferencia los los anteriores, nos mostrará los valores residuales de cada estudio (es decir, qué tanto se alejan del resultado de nuestro meta-análisis, que tiene un valor residual de 0; Fig. 6), en vez de los coeficientes de correlación (transformados a *z* de Fisher).

```
metafor::funnel(res.modage,
                xlab = "Valor residual",
                ylab = "Error estándar")
```

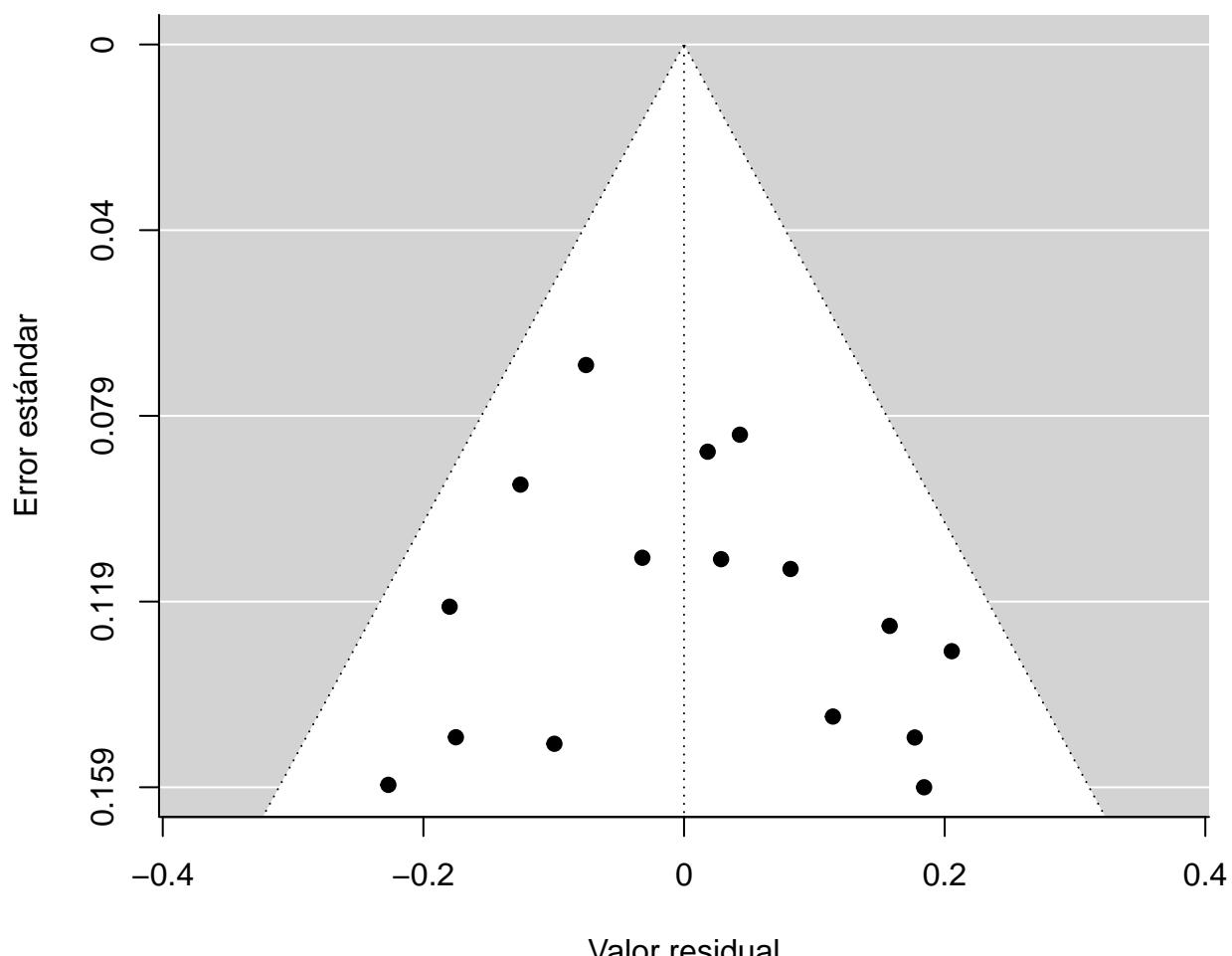


Figura 17. Funnel plot básico de `metafor`, para un meta-análisis incluyendo la edad promedio de los participantes como moderador, y con títulos de los ejes en español. La línea punteada vertical representa el efecto meta-analizado que hemos encontrado, así que podemos ver los estudios que encontraron un efecto mayor (derecha de la línea punteada) o menor (izquierda) de éste. **Nota:** Para evitar confusión entre las funciones `funnel` de los paquetes `weightr` y `metafor`, en este caso he usado el comando `metafor::funnel` para pedirle a R explícitamente que use la función `funnel` del paquete `metafor`.

4.2. Ejemplo 2: Moderación de una variable categórica (controles usados en cada estudio meta-analizado)

Como segundo ejemplo, voy a mirar si el hecho de que los estudios tengan variables que fueron controladas, modera la asociación entre concienciación (*conscientiousness*) y adherencia a la medicación prescrita. Siguiendo los mismos pasos, voy hacer éste análisis, pero voy a asignar este meta-análisis a un objeto llamado `res.contr`.

```
res.contr <- rma(yi = yi, vi = vi, mods = ~controls, data = dat)
res.contr
```

Consola de R: Output 15

```
Mixed-Effects Model (k = 16; tau^2 estimator: REML)

tau^2 (estimated amount of residual heterogeneity): 0.0000 (SE = 0.0015)
tau (square root of estimated tau^2 value): 0.0002
I^2 (residual heterogeneity / unaccounted variability): 0.00%
H^2 (unaccounted variability / sampling variability): 1.00
R^2 (amount of heterogeneity accounted for): 100.00%

Test for Residual Heterogeneity:
QE(df = 14) = 18.0370, p-val = 0.2051
```

```

Test of Moderators (coefficient 2):
QM(df = 1) = 20.1221, p-val < .0001

Model Results:

            estimate      se     zval   pval    ci.lb    ci.ub
intrcpt      0.1788  0.0208  8.6044 <.0001   0.1380   0.2195 ***

controlsmultiple -0.1621  0.0361 -4.4858 <.0001  -0.2329  -0.0913 ***

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

```

En éste caso, a diferencia del ejemplo de moderación anteriores, la variable moderadora (`controls`) sí tiene un efecto significativo, como se puede ver en la columna `pval` para el efecto de `controlsnone` (<0.001), y en los asteriscos que aparecen al final de esa fila (***)�.

4.2.1. Más información e interpretación de la moderación

```

pred.res.contr <- predict(res.contr, newmods = c(0, 1)) %>%
  as.data.frame() %>%
  mutate_all(~round(., 3)) %>%
  mutate(controls = levels(dat$controls)) %>%
  rename(yi = pred) %>%
  select(7, 1:6)
pred.res.contr

```

Consola de R: Output 16

	controls	yi	se	ci.lb	ci.ub	pi.lb	pi.ub
1	none	0.179	0.021	0.138	0.219	0.138	0.219
2	multiple	0.017	0.030	-0.041	0.075	-0.041	0.075

4.2.1.1. Meta-Analytic Scatter Plot (Gráfico de dispersión meta-analítico) Al igual que en el ejemplo de meta-análisis con moderación de una variable continua (sección @ref(ex.mod1)), es posible hacer un gráfico de dispersión meta-analítico (*meta-analytic scatter plot*) usando la función `regplot` del paquete `metafor`. Sin embargo, dado que la variable moderadora es categórica, el modelo genera variables *dummy* asignando valores de 0 y 1 a los niveles de esta variable (en este caso, 0 = ningún control; 1 = múltiples controles). Por esto, en el ejemplo a continuación agregué a la descripción del eje *X* esta información.

```

regplot(res.contr,
        ylab = "Coeficiente de correlación transformado en z de Fisher",
        xlab = "Controles (0 = ninguno; 1 = múltiples)")

```

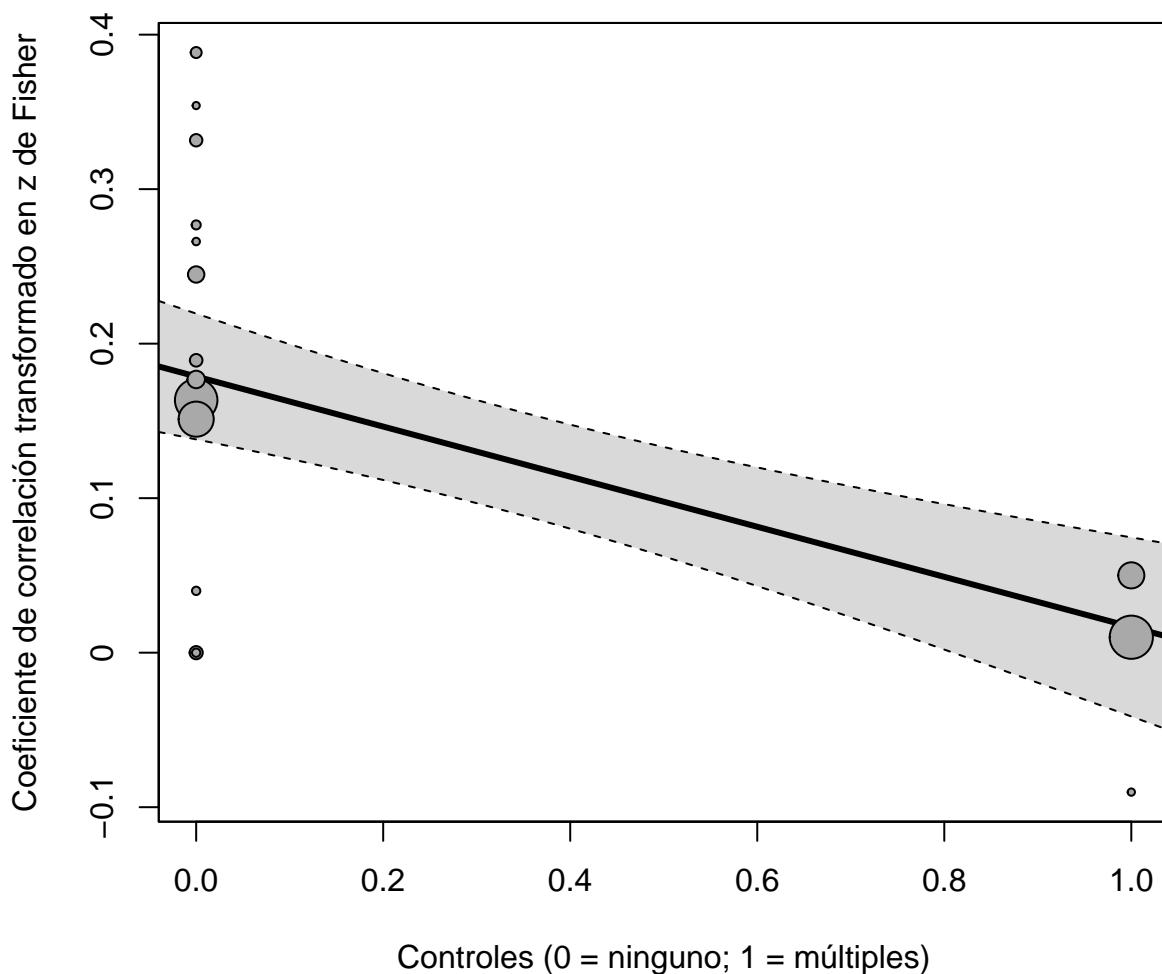


Figura 18. Gráfico de dispersión meta-analítico (*Meta-Analytic Scatter Plot*) básico de `metafor` creado con la función `regplot`. El tamaño de los puntos es proporcional al peso que recibieron los estudios en el meta-análisis (puntos más grandes para los estudios con más peso, pues tienen un tamaño de muestra mayor y con un menor error estimado). La línea negra representa el efecto previsto en función del predictor (en este caso `controls`, controles). La banda gris demilimata por líneas punteadas representa el intervalo de confianza del 95 %. Dado que la variable moderadora es categórica, el modelo genera variables *dummy* asignando valores de 0 y 1 a los niveles de esta variable (en este caso, 0 = ningúun control; 1 = múltiples controles), tal y como se describe en el eje *X*. Para una versión más apropiada, ver Fig. 19.

Aunque la figura 18 presenta información correcta, tal vez no es la más adecuada ni la más clara cuando se trata de representar meta-análisis con moderadores categóricos. Por ejemplo, a pesar de que nuestro moderador es un factor (controles) con dos niveles (ninguno, múltiples), en el eje *X* se representa como una variable continua con valores entre 0 y 1, que además tiene valores intermedios (0.2, 0.4, 0.6, 0.8) que en nuestro caso no tienen sentido.

Sin embargo, de manera un poco más *artesanal*, es posible crear una versión más adecuada usando, por ejemplo, `ggplot2`. El código siguiente permite crear una figura bastante sofisticada usando éste paquete.

```
# Definir base de datos, así como ejes X (controls) y Y (yi)
ggplot(dat, aes(x = controls, y = yi)) +
  # Agregar puntos para cada estudio, con tamaño y color según tamaño de muestra (ni)
  geom_point(aes(size = ni, color = ni),
             alpha = 0.5) +
  # Definir escala de colores
  scale_colour_gradient(low = "red",
                        high = "blue") +
  # Definir rango de tamaño de los puntos
  scale_size_continuous(range = c(1, 10)) +
  # Combinar leyendas de colores y tamaños de puntos en una sola leyenda
```

```

guides(color = guide_legend(),
       size = guide_legend()) +
# Traducir etiquetas del eje X en español
scale_x_discrete(labels=c("none" = "Ninguno",
                           "multiple" = "Múltiples")) +
# Cambiar títulos de ejes a español
labs(x = "Controles",
     y = "Coeficiente de correlación \ntransformado en z de Fisher") +
# Agregar barras de error para cada categoría, con base en predicción hecha en sección 4.2.1
geom_errorbar(data = pred.res.contr,
               mapping = aes(ymin = ci.lb, ymax = ci.ub),
               width = 0.1,
               color = "black") +
# Agregar puntos blancos representando predicción para cada categoría hecha en sección 4.2.1
geom_point(data = pred.res.contr,
            shape = 21, size = 3,
            color = "black", fill = "white") +
# Cambiar tptitulo de leyenda
labs(color = "Tamaño de \nmuestra",
     size = "Tamaño de \nmuestra")

```

El código anterior puede verse algo confuso para quien no haya usado `ggplot2` antes. Por esto, he agregado anotaciones, función por función, para ayudar a su interpretación¹⁴. Este código produce la figura 19.

¹⁴Debido a sus enormes y numerosas posibilidades, `ggplot2` es quizás la opción más poderosa para hacer gráficos estadísticos en R; por esto explicar sus bases supera por mucho el alcance de esta guía, y posiblemente requeriría múltiples tutoriales independientes. Sin embargo, hay gran cantidad de opciones disponibles en internet. Por ejemplo, puedes leer este [tutorial](#) (López Peñaranda, 2019) o ver este [video](#) (Datademia, 2018).

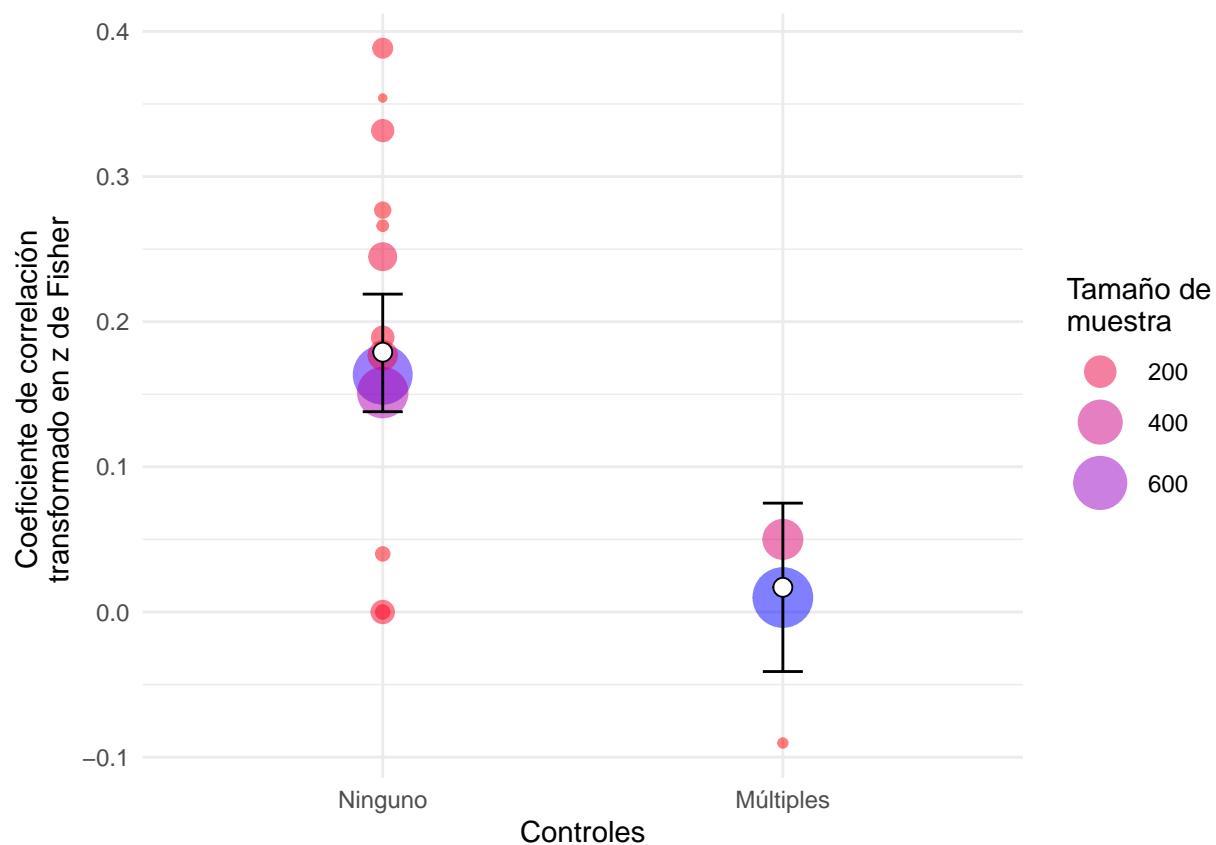


Figura 19. Gráfico de dispersión meta-analítico (*Meta-Analytic Scatter Plot*) creado manualmente con [ggplot2](#) para hacer una mejor representación de un moderador categórico. Los puntos de colores representan el coeficiente de correlación en función de la presencia o ausencia de controles. El tamaño de los puntos es proporcional al tamaño de muestra de los estudios incluidos en el meta-análisis (puntos más grandes y azules para los estudios con mayor tamaño de muestra). Los puntos blancos superpuestos representan el efecto estimado para cada categoría, y las barras de error representan los intervalos de confianza del 95 %.

4.2.2. Forest plot y funnel plot

Por supuesto, *forest plots* y *funnel plots* pueden ser creados, tal y como describí en la sección 4.1.2.

5. Referencias

- Acerca del sesgo de publicación en literatura sobre antidepresivos. (2008). *Revista Colombiana de Psiquiatría*, 37(1), 8–10. <http://ref.scielo.org/sp52pr>
- Balduzzi, S., Rücker, G., & Schwarzer, G. (2019). How to perform a meta-analysis with R: A practical tutorial. *Evidence-Based Mental Health*, 22, 153–160. <https://doi.org/10.1136/ebmental-2019-300117>
- Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. John Wiley & Sons, Inc. <https://doi.org/10.1002/0471725153>
- Borenstein, M., Hedges, L. V., Higgins, J. P. T., & Rothstein, H. R. (2009). Identifying and Quantifying Heterogeneity. In *Introduction to Meta-Analysis* (pp. 107–125). Wiley. <https://doi.org/10.1002/9780470743386.ch16>
- Coburn, K. M., & Vevea, J. L. (2015). Publication bias as a function of study characteristics. *Psychological Methods*, 20(3), 310–330. <https://doi.org/10.1037/met0000046>
- Coburn, K. M., & Vevea, J. L. (2019). *Weightr: Estimating weight-function models for publication bias*. <https://CRAN.R-project.org/package=weightr>
- Cook, R. D. (1977). Detection of Influential Observation in Linear Regression. *Technometrics*, 19(1), 15–18. <https://doi.org/10.2307/1268249>
- Cook, R. D., & Weisberg, S. (1982). *Residuals and Influence in Regression*. New York: Chapman and Hall. <https://hdl.handle.net/11299/37076>
- Datademia. (2018). *Aprende Ggplot2 en R* [video]. YouTube. https://youtu.be/BUzTAr_QqKs.
- Duval, S. (2005). The Trim and Fill Method. In H. R. Rothstein, A. J. Sutton, & M. Borenstein (Eds.), *Publication Bias in Meta-Analysis* (pp. 127–144). John Wiley & Sons. <https://doi.org/10.1002/0470870168.ch8>
- Duval, S., & Tweedie, R. (2000a). A Nonparametric “Trim and Fill” Method of Accounting for Publication Bias in Meta-Analysis. *Journal of the American Statistical Association*, 95(449), 89–98. <https://doi.org/10.1080/01621459.2000.10473905>
- Duval, S., & Tweedie, R. (2000b). Trim and Fill: A Simple Funnel-Plot-Based Method of Testing and Adjusting for Publication Bias in Meta-Analysis. *Biometrics*, 56(2), 455–463. <https://doi.org/10.1111/j.0006-341X.2000.00455.x>
- Egger, M., Smith, G. D., Schneider, M., & Minder, C. (1997). Bias in Meta-Analysis Detected by a Simple, Graphical Test. *BMJ*, 315(7109), 629–634. <https://doi.org/10.1136/bmj.315.7109.629>
- Fisher, R. A. (1915). Frequency Distribution of the Values of the Correlation Coefficient in Samples from an Indefinitely Large Population. *Biometrika*, 10(4), 507–521. <https://doi.org/10.2307/2331838>
- Harrer, M., Cuijpers, P., A, F. T., & Ebert, D. D. (2021). *Doing Meta-Analysis With R: A Hands-On Guide* (1st ed.). Chapman & Hall/CRC Press. https://bookdown.org/MathiasHarrer/Doing_Meta_Analysis_in_R/
- Harrer, M., Cuijpers, P., Furukawa, T., & Ebert, D. D. (2019). *dmetar: Companion R Package For The Guide 'Doing Meta-Analysis in R'*. <http://dmetar.protectlab.org/>
- Henry, N. W., Cohen, J., & Cohen, P. (2003). *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. (3rd ed.). Lawrence Erlbaum Associates. <https://doi.org/10.4324/9780203774441>
- John, O. P., & Srivastava, S. (1999). The Big Five Trait taxonomy: History, measurement, and theoretical perspectives. In L. A. Pervin & O. P. John (Eds.), *Handbook of personality: Theory and research*, 2nd ed (pp. 102–138). Guilford Press. http://jenni.uchicago.edu/econ-psych-trait/John_Srivastava_1995_big5.pdf
- Kossmeier, M., Tran, U. S., & Voracek, M. (2020). *metaviz: Forest Plots, Funnel Plots, and Visual Funnel Plot Inference for Meta-Analysis*. <https://CRAN.R-project.org/package=metaviz>
- Leongómez, J. D. (2021). *Hacer meta-análisis en jamovi es muy fácil* [video]. YouTube. https://youtu.be/ntBbkOn9D_o.
- Leongómez, J. D. (2020a). *Análisis de poder estadístico y cálculo de tamaño de muestra en R: Guía práctica*. Zenodo. <https://doi.org/10.5281/zenodo.3988776>
- Leongómez, J. D. (2020b). *Poder estadístico y tamaño de muestra en R* [serie de videos]. YouTube. <https://youtu.be/PLHk7UNt35ccVdyHqnQ6oXVYA6JBNFrE1x>.
- López Peñaranda, A. (2019). *Tutorial de ggplot2*. RPubs. <https://rpubs.com/anlope10/562981>.
- Mollov, G. J., O’Carroll, R. E., & Ferguson, E. (2013). Conscientiousness and medication adherence: A meta-analysis. *Annals of Behavioral Medicine*, 47(1), 92–101. <https://doi.org/10.1007/s12160-013-9524-4>
- Quintana, D. S. (2021). *How to Perform a Meta-Analysis in R* [video]. YouTube. <https://youtu.be/lH4VZMTEZSc>.
- Quintana, D. S. (2022). *Metameta: A suite of tools to re-evaluate published meta-analyses*. <https://github.com/dsqquintana/metameta>
- Quintana, D. S. (in press). A guide for calculating study-level statistical power for meta-analyses. *Advances in Methods and Practices in Psychological Science*. <https://doi.org/10.31219/osf.io/js79t>
- Revelle, W. (2021). *Psych: Procedures for psychological, psychometric, and personality research*. Northwestern Uni-

- versity. <https://CRAN.R-project.org/package=psych>
- Sánchez-Bruno, A., & Borges del Rosal, Á. (2005). Transformación Z de Fisher para la determinación de intervalos de confianza del coeficiente de correlación de Pearson. *Psicothema*, 17(1), 148–153. <https://www.redalyc.org/articulo.oa?id=72717124>
- Schwarzer, G., Carpenter, J. R., & Rücker, G. (2015a). *Meta-Analysis with R*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-21416-0>
- Schwarzer, G., Carpenter, J. R., & Rücker, G. (2015b). Small-Study Effects in Meta-Analysis. In G. Schwarzer, J. R. Carpenter, & G. Rücker (Eds.), *Meta-Analysis with R* (pp. 107–141). Springer International Publishing. https://doi.org/10.1007/978-3-319-21416-0_5
- Sedgwick, P. (2015). Meta-analyses: What is heterogeneity? *BMJ*, 350, h1435. <https://doi.org/10.1136/bmj.h1435>
- Shi, L., & Lin, L. (2019). The trim-and-fill method for publication bias: Practical guidelines and recommendations based on a large database of meta-analyses. *Medicine*, 98(23), e15987. <https://doi.org/10.1097/MD.00000000000015987>
- Simonsohn, U., Nelson, L. D., & Simmons, J. P. (2014). P-Curve and Effect Size: Correcting for Publication Bias Using Only Significant Results. *Perspectives on Psychological Science*, 9(6), 666–681. <https://doi.org/10.1177/1745691614553988>
- Sterne, J. A. C., Gavaghan, D., & Egger, M. (2000). Publication and related bias in meta-analysis. *Journal of Clinical Epidemiology*, 53(11), 1119–1129. [https://doi.org/10.1016/S0895-4356\(00\)00242-0](https://doi.org/10.1016/S0895-4356(00)00242-0)
- Vevea, J. L., & Hedges, L. V. (1995). A general linear model for estimating effect size in the presence of publication bias. *Psychometrika*, 60(3), 419–435. <https://doi.org/10.1007/BF02294384>
- Vevea, J. L., & Woods, C. M. (2005). Publication bias in research synthesis: Sensitivity analysis using a priori weight functions. *Psychological Methods*, 10(4), 428–443. <https://doi.org/10.1037/1082-989X.10.4.428>
- Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software*, 36(3). <https://doi.org/10.18637/jss.v036.i03>
- Wickham, H. (2022). *Forcats: Tools for working with categorical variables (factors)*. <https://CRAN.R-project.org/package=forcats>
- Wickham, H., François, R., Henry, L., & Müller, K. (2021). *Dplyr: A grammar of data manipulation*. <https://CRAN.R-project.org/package=dplyr>

APÉNDICES

A. Alternativas a `metafor`

Acá he usado principalmente una ruta para hacer meta-análisis basada en el paquete `metafor`, acompañado de `metaviz` para visualizaciones, `weightr` para ajustar pesos y detectar sesgos de publicación, y `metameta` para estimar el poder estadístico de un meta-análisis.

Sin embargo, existen rutas alternativas para realizar meta-análisis en R. El libro *Doing meta-analysis with R: a hands-on guide* (Harrer et al., 2021) se acompaña del paquete `dmetar` (Harrer et al., 2019), que contiene opciones para hacer meta-análisis tanto a partir de `metafor`, como a partir de `meta` (Balduzzi et al., 2019; Schwarzer et al., 2015a).

De manera importante, los objetos generados por `meta` al realizar un meta-análisis permiten hacer otros análisis como *risk of bias* (riesgo de sesgo), inferencia multi-modelo, detección de *outliers* (valores atípicos), y *p-curve* o curva de valores *p* (Simonsohn et al., 2014), así como opciones para hacer gráficos distintos. Para una guía resumida y concreta (en inglés) de estas opciones, recomiendo ver el sitio web del paquete `dmetar`, y en especial la página *Get Started*.

B. Citas y referencias de paquetes de R

Por supuesto, los paquetes de R que usemos deben ser citados. Una manera fácil de encontrar la cita que los autores de un paquete quieren que usemos, es la función `citation` en R. Simplemente debemos usar esta función, agregando como argumento el nombre del paquete que queremos citar entre comillas. Esto nos dará la referencia en un formato estándar, así como como en un formato BibTeX que puede ser usado en documentos L^AT_EX, o por muchos gestores de referencia (alternativamente nos permite saber los campos como autores, título y demás, si vamos a crear las citas y referencias manualmente).

Por ejemplo, en ésta guía usé `dplyr` (Wickham et al., 2021) para manipular los datos, y usando la función `citation`, obtengo esta información:

```
citation("dplyr")
```

Consola de R: Output 17

To cite package 'dplyr' in publications use:

```
Wickham H, François R, Henry L, Müller K (2022). _dplyr: A Grammar of Data Manipulation_. R package version 1.0.10, <https://CRAN.R-project.org/package=dplyr>.
```

A BibTeX entry for LaTeX users is

```
@Manual{,
  title = {dplyr: A Grammar of Data Manipulation},
  author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller},
  year = {2022},
  note = {R package version 1.0.10},
  url = {https://CRAN.R-project.org/package=dplyr},
```

C. Paquetes de R y versiones usados para crear este documento (para garantizar reproducibilidad)

R version 4.2.2 (2022-10-31 ucrt)

Platform: x86_64-w64-mingw32/x64 (64-bit)

attached base packages: *stats, graphics, grDevices, utils, datasets, methods and base*

other attached packages: *pander(v.0.6.5), metameta(v.0.1.1), weightr(v.2.0.2), scales(v.1.2.1), metaviz(v.0.3.1), tidyquant(v.1.0.6), quantmod(v.0.4.20), TTR(v.0.24.3), PerformanceAnalytics(v.2.0.4), xts(v.0.12.2), zoo(v.1.8-11), lubridate(v.1.9.0), timechange(v.0.2.0), psych(v.2.2.9), kableExtra(v.1.3.4), ggpubr(v.0.5.0), forcats(v.0.5.2), stringr(v.1.5.0), dplyr(v.1.0.10), purrr(v.1.0.1), readr(v.2.1.3), tidyverse(v.1.2.1), tibble(v.3.1.8), ggplot2(v.3.4.0), tidyverse(v.1.3.2), metafor(v.3.8-1), metadat(v.1.2-0), Matrix(v.1.5-3), robumeta(v.2.0) and knitr(v.1.41)*

loaded via a namespace (and not attached): *nlme(v.3.1-160), fs(v.1.5.2), RColorBrewer(v.1.1-3), webshot(v.0.5.4), httr(v.1.4.4), tools(v.4.2.2), backports(v.1.4.1), utf8(v.1.2.2), R6(v.2.5.1), mgcv(v.1.8-41), DBI(v.1.1.3), colorspace(v.2.0-3), withr(v.2.5.0), gridExtra(v.2.3), tidyselect(v.1.2.0), mnormt(v.2.1.1), curl(v.5.0.0), compiler(v.4.2.2), cli(v.3.6.0), rvest(v.1.0.3), xml2(v.1.3.3), labeling(v.0.4.2), bookdown(v.0.31), quadprog(v.1.5-8), systemfonts(v.1.0.4), digest(v.0.6.31), rmarkdown(v.2.19), svglite(v.2.1.1), pkgconfig(v.2.0.3), htmltools(v.0.5.4), highr(v.0.10), dbplyr(v.2.3.0), fastmap(v.1.1.0), ggthemes(v.4.2.4), rlang(v.1.0.6), readxl(v.1.4.1), rstudioapi(v.0.14), farver(v.2.1.1), generics(v.0.1.3), jsonlite(v.1.8.4), car(v.3.1-1), googlesheets4(v.1.0.1), magrittr(v.2.0.3), Rcpp(v.1.0.9), Quandl(v.2.11.0), munsell(v.0.5.0), fansi(v.1.0.3), abind(v.1.4-5), lifecycle(v.1.0.3), stringi(v.1.7.12), yaml(v.2.3.6), carData(v.3.0-5), mathjaxr(v.1.6-0), grid(v.4.2.2), parallel(v.4.2.2), crayon(v.1.5.2), lattice(v.0.20-45), cowplot(v.1.1.1), splines(v.4.2.2), haven(v.2.5.1), hms(v.1.1.2), pillar(v.1.8.1), ggsignif(v.0.6.4), reprex(v.2.0.2), glue(v.1.6.2), evaluate(v.0.19), modelr(v.0.1.10), vctrs(v.0.5.1), tzdb(v.0.3.0), cellranger(v.1.1.0), gtable(v.0.3.1), assertthat(v.0.2.1), xfun(v.0.36), broom(v.1.0.2), rstatix(v.0.7.1), googledrive(v.2.0.0), viridisLite(v.0.4.1), garble(v.1.2.1) and ellipsis(v.0.3.2)*