

Meta-análisis de correlaciones en R:

Guía práctica

Juan David Leongómez 

26 febrero, 2022

Laboratorio de Análisis del Comportamiento Humano (LACH), Facultad de Psicología, Universidad El Bosque, Bogotá, Colombia. Email: jleongomez@unbosque.edu.co. Web: jdleongomez.info.

Descripción

Este documento contiene todo el código explicaciones básicas, paso a paso, para hacer un meta-análisis de coeficientes de correlación en R, usando los paquetes [metafor](#) ([Viechtbauer, 2010](#)) y [metaviz](#) ([Kossmeier et al., 2020](#)). Está principalmente basado en [este video](#), creado por Daniel S. Quintana ([2021](#)), pero contiene citas a fuentes primarias, además de información que he agregado.

Esta guía asume un manejo básico de R, así como una comprensión básica del meta-análisis. Sin embargo, de ser necesario y como introducción, recomiendo ver [este video introductorio](#) al meta-análisis en *jamovi* ([Leongómez, 2021](#)) que publiqué anteriormente en mi canal de YouTube [Investigación Abierta](#).



Cita esta guía como

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

Índice

1. Base de datos de ejemplo	3
2. Transformación de r de Pearson a z de Fisher	4
3. Hacer el meta-análisis	5
3.1. Más información sobre heterogeneidad	6
3.2. Diagnóstico de influencia	6
3.3. <i>Forest plot</i> (diagrama de bosque)	8
3.4. <i>Funnel plot</i> (diagrama de embudo) y sesgo de estudios pequeños	11
3.4.1. <i>Funnel plot</i>	11
3.4.2. Regresión de Egger	15
4. Meta-análisis de correlación con moderador	15
4.1. Ejemplo 1: Moderación de la edad promedio de los participantes	15
4.1.1. <i>Forest plot</i> y <i>funnel plot</i>	16
4.2. Ejemplo 2: Moderación de la calidad de los estudios meta-analizados	17
4.3. Ejemplo 3: Moderación de los controles usados en cada estudio meta-analizado	18
5. Sesgo de publicación (<i>publication bias</i>)	19
6. Poder estadístico del meta-análisis	20
6.1. Instalación de <code>metameta</code>	20
6.2. Análisis de poder	20
6.3. Visualización del análisis de poder (<i>Firepower plot</i>)	21
APÉNDICE 1: Alternativas a <code>metafor</code>	23
Referencias	23

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 diligencia (*conscientiousness*) y la adherencia a la medicación. En otras palabras, ¿las personas más diligentes son más propensas a cumplir con la medicación prescrita?

Primero, primero, debemos cargar los paquetes que usaremos, incluyendo `{metafor}` y `{metaviz}` para hacer meta-análisis, así como `{dplyr}` para manipular y organizar la base de datos.

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

Una vez cargado el paquete `{metafor}`, ya puedo cargar la base de datos `dat.molloy2014`. En éste caso, para poder llamarla cuando sea necesario, la asignaré a un objeto llamado `dat`.

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

Tras asignar la base de datos a este objeto (`dat`), podemos verla en la consola de R sencillamente usando el comando `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				

Por supuesto, la salida de la consola no es la más clara, así que de aquí en adelante mostraré la mayoría de las tablas en un formato de impresión, más fácil 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. A continuación, reorganizaré las columnas para que `study_id` sea la primera, en vez de la última columna.

```
dat <- get(data(dat.molloy2014)) %>%
  mutate(study_id = 1:16) %>% #agregar columna study_id
  select(study_id, authors:quality) #mover study_id como primera columna
```

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'Cleirigh 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).

Por supuesto, la columna **authors** tiene los autores de cada estudio a meta-analizar, y la columna **year** el año de publicación.

La columna **ri** contiene los coeficientes de correlación de Pearson (la columna **ni** contiene los tamaños de muestra de cada estudio).

Adicionalmente, en este ejemplo tenemos una serie de moderadores:

- **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)
- **c_measure:** tipo de medida de diligencia (NEO u otra)
- **meanage:** edad promedio de la muestra
- **quality:** calidad metodológica

2. Transformación de r de Pearson a z de Fisher

Dado que los coeficientes de Pearson (columna **ri**) no tienen una distribución normal, esto podría llevar a calcular varianzas incorrectas, especialmente cuando se trata de correlaciones con tamaños de muestra pequeños. Por esto, vamos a transformar los coeficientes r de Pearson a z de Fisher, que no tienen este problema. Usaré la función **escalc** del paquete **metafor**.

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

Esto ha creado dos nuevas variables en nuestra tabla: **yi**, que es el tamaño de efecto, y **vi** que es la varianza.

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'Cleirigh 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, usaremos la función `rma` del paquete `metafor`, para el que tenemos que especificar los tamaños de efecto (`yi`) y varianzas (`vi`) de los estudios a meta-analizar. En este caso, las columnas donde tenemos estos valores, tienen los mismos nombres (`yi`, `vi`). Asignaré los resultados del meta-análisis a un 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
```

Primero, nos confirma 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**¹ (`tau^2 estimator: REML`), que se designa como *REML* por sus siglas en inglés.

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

- 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`

¹Hay 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`.

- 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 una prueba de heterogeneidad, usando el estadístico Q :

- Test for Heterogeneity:
 $Q(df = 15) = 38.1595$, $p\text{-val} = 0.0009$

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 más de la mitad (61.73 %) de la varianza se estima que se deriva de diferencias en los tamaños de efecto.

Por último, tenemos los resultados del modelo de meta-análisis (**Model results**). Nos provee un estimado de la asociación positiva entre diligencia y adherencia a la medicación (0.1499 ± 0.0316), lo que equivale a un valor z de 4.7501, y sugiere que esa asociación es significativa ($p < .0001$). Así mismo, nos provee los límites inferior (0.0881) y superior (0.2118) de los intervalos de confianza.

3.1. Más información sobre heterogeneidad

Además de reportar los estadísticos τ^2 , τ , I^2 y Q , podemos fácilmente calcular los intervalos de confianza para τ^2 , τ , e I^2 con la función `confint`, que también pueden ser reportado junto a estos estadísticos.

```
confint(res)
```

Consola de R: Output 3

	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 el τ^2 , el hecho de que los intervalos de confianza no crucen el 0 (en nuestro caso 0.0017 — 0.0378), sugiere que de hecho también que hay heterogeneidad entre los estudios que meta-analizamos.

3.2. 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 resultado². Para esto, podemos usar la función `influence`, cuyo resultado en este caso asignaré a un objeto llamado `inf.res`.

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

Ya que lo asigné a un objeto (`inf.res`), para ver el resultado, tengo que correr este objeto para ver su resultado.

²Por 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 tu meta-análisis sea dependiente de un único estudio.

inf.res

Consola de R: Output 4

```

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, una tabla sin formato). Sin embargo, lo más importante ahora es mirar la última columna, llamada `inf`. Si ahí aparecieran asteriscos (que no es nuestro caso), sugeriría que ese estudio es particularmente influyente.

Por último, podemos también ver esta información que tenemos guardada en el objeto `inf.res`, de manera gráfica, usando la función `plot`.

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

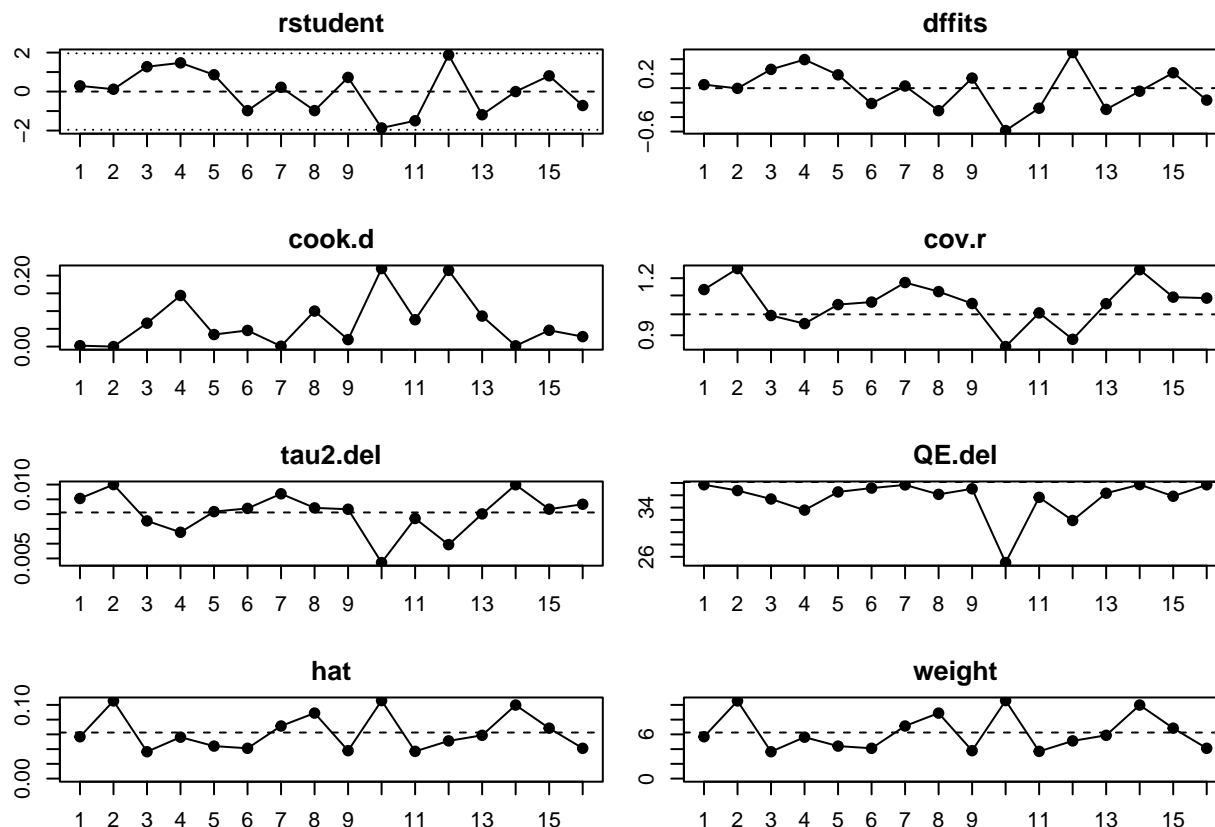


Figura 1. Diagnóstico de influencia. Estudios particularmente influyentes serían representados con un punto rojo. En este caso, no hay ningún estudio que se considere demasiado influyente, por lo éste análisis sugiere que podemos estar tranquilos con nuestro meta-análisis.

3.3. 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`).

```
par(mar=c(4,0,0,0))
forest(res)
```

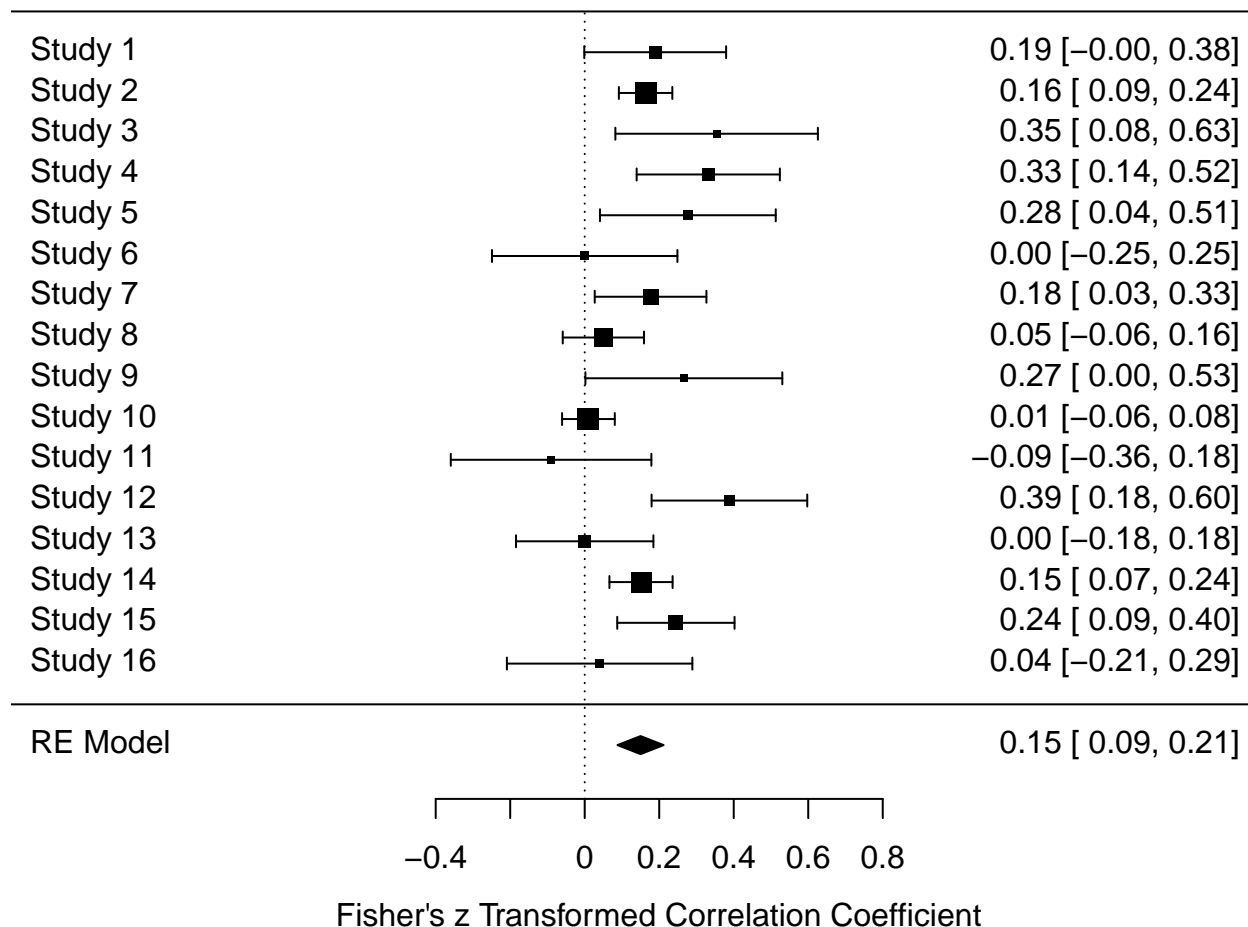


Figura 2. Forest plot básico de `metafor`. Para cada estudio meta-analizado, tenemos el efecto (correlación, en este caso en valores z de Fisher), así como sus intervalos de confianza entre paréntesis cuadrados. Esta misma información está representada gráficamente, con los cuadrados representando el efecto de cada estudio así como sus intervalos de confianza, 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 2, 3 y 4 (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 algunos 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-analizados (en 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`) así como una columna con los pesos dados a cada estudio, y detalles del modelo final³:

³Estas opciones están explicadas [aquí](#).


```
# forest plot con anotaciones adicionales
par(mar=c(4,0,0,0))
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")), "%"))))
# 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)
```

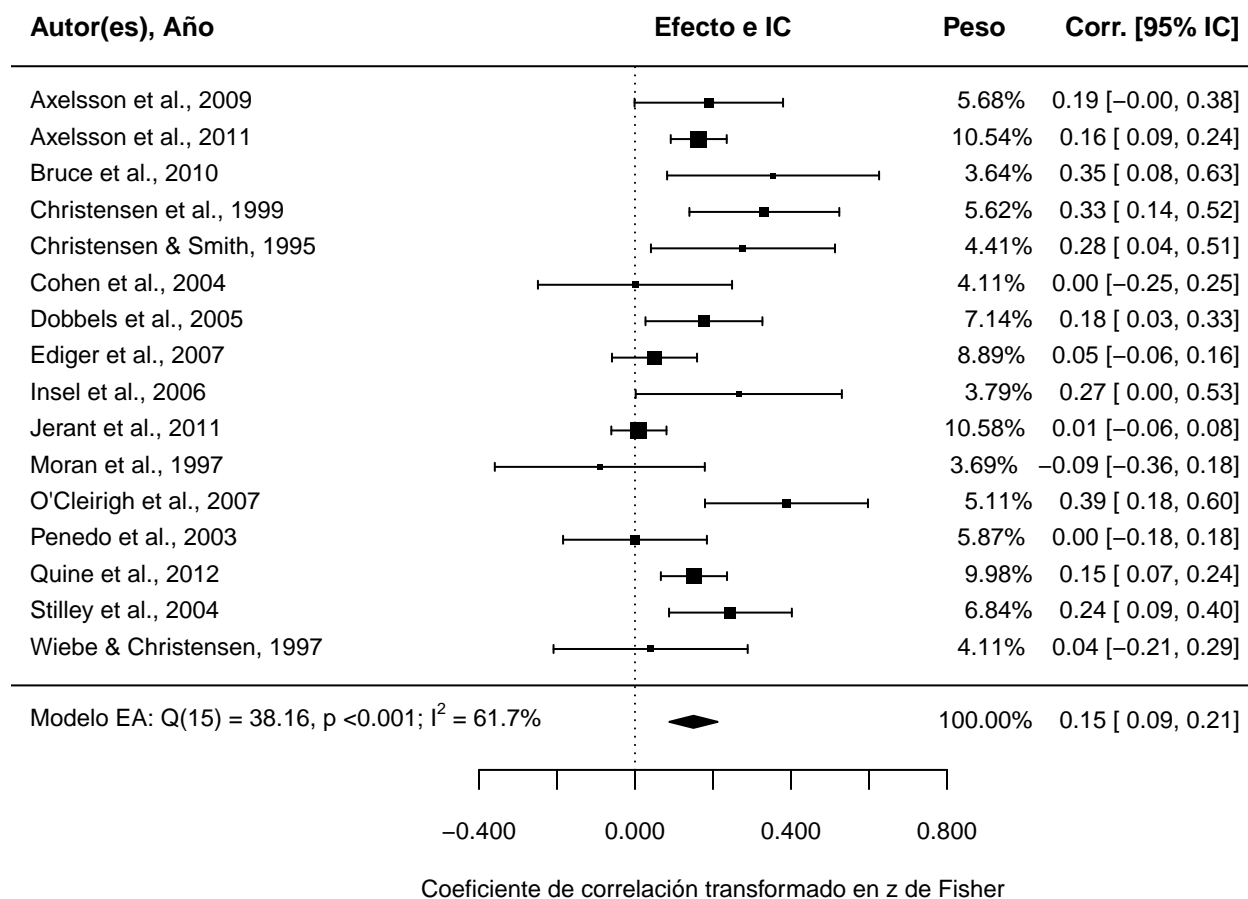


Figura 3. Forest plot anotado, creado con [metafor](#). En esta versión agregué algunos encabezados en español, así como estadísticos generales del modelo de meta-análisis. Modelo EA se refiere al modelo meta-analizado, de efectos aleatorios.

O, para una incluso más sofisticada, 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)

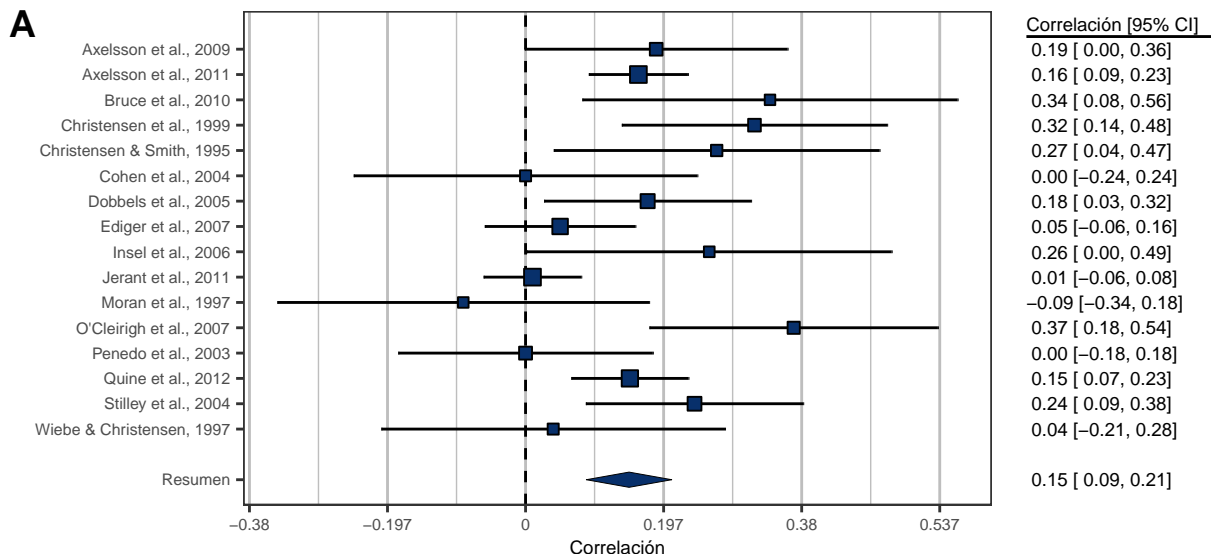
# B. Variante "thick"
viz_forest(res,
  study_labels = paste(dat$authors, dat$year, sep = ", "),
  xlab = "Correlación",
  variant = "thick",
  col = "Greens",
  annotate_CI = TRUE,
  summary_label = "Resumen",
  text_size = 2.6,
  x_trans_function = tanh)

# 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* usando diferentes variantes y escalas de colores, y transformando de vuelta los coeficientes de z de Fisher a r de Pearson.

Por supuesto, es cuestión de gusto cuál usar.



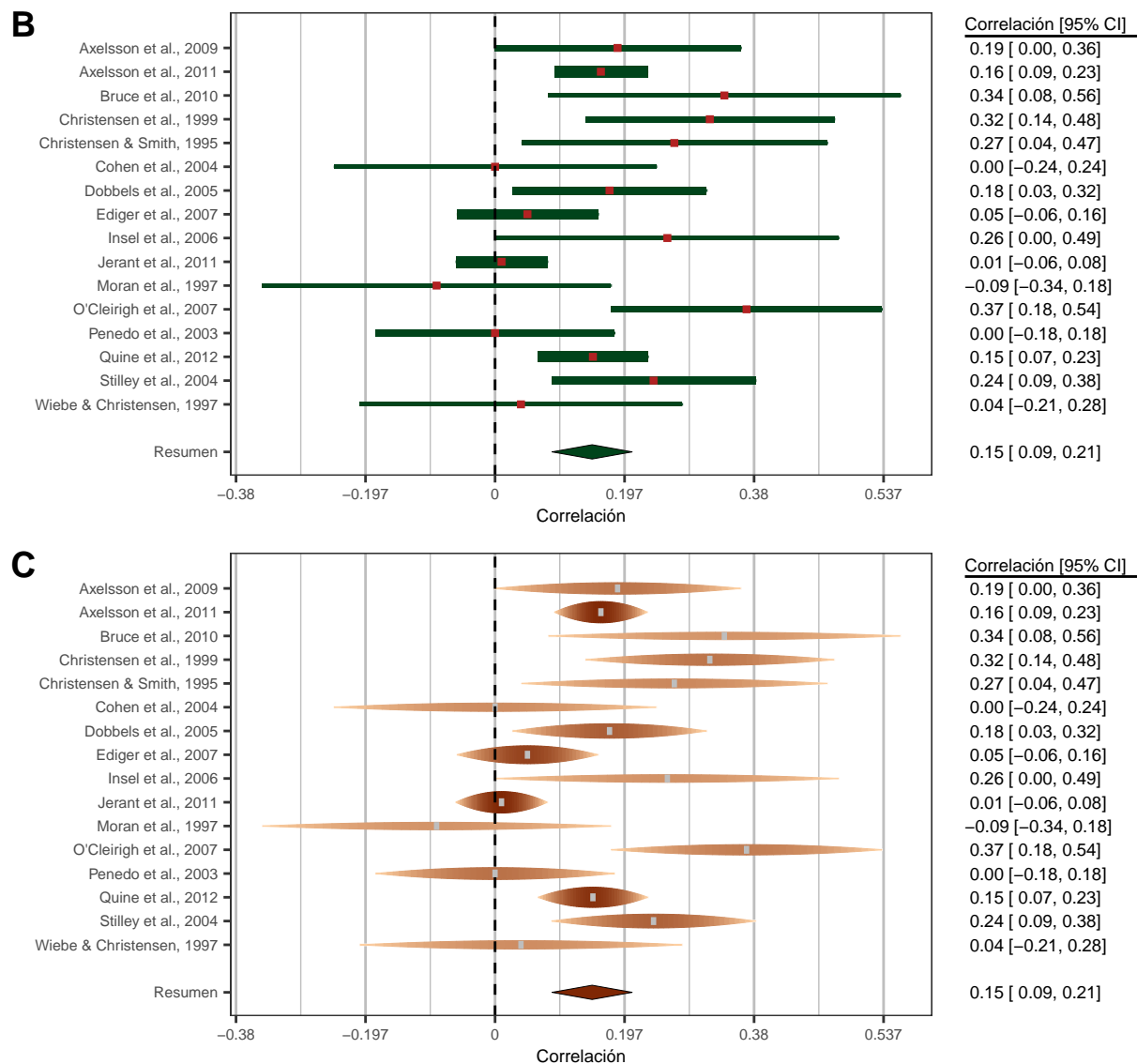


Figura 4. Variantes de forest plots creados con *metaviz*. **A.** Variante clásica (opción por defecto). **B.** Variante “thick” y escala de colores “Greens”. **C.** Variante “rain” y escala de colores “Oranges”.

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

En este punto, es en donde más errores se cometen. 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).

El principal error que la mayoría de los investigadores (meta-analistas) cometen, es que simplemente basándose en éstos métodos, concluyen que un meta-análisis tiene (o no) riesgo de sufrir de un sesgo de publicación. 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.

3.4.1. *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`).

```
funnel(res)
```

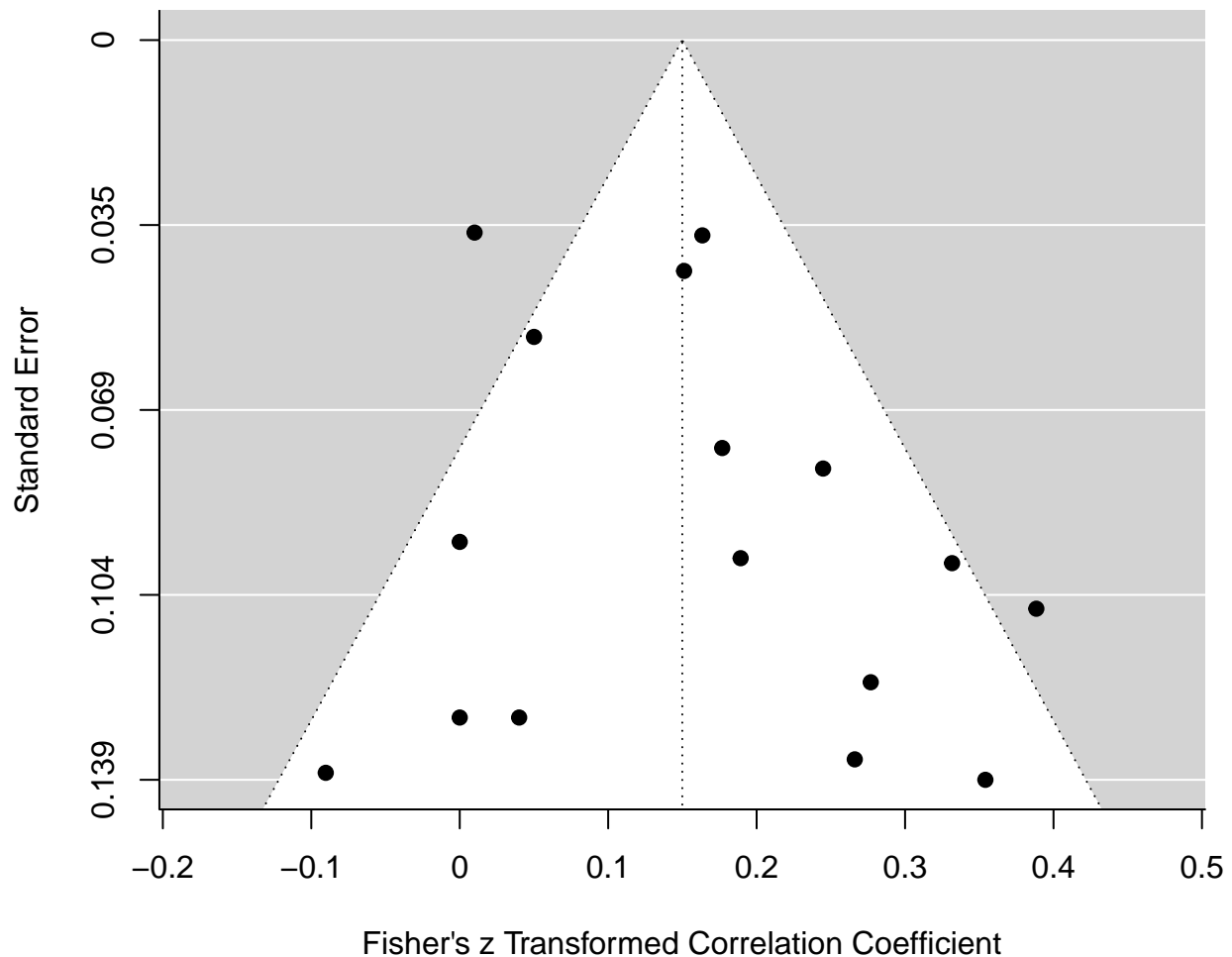


Figura 5. Funnel plot básico de [metafor](#). Para cada estudio meta-analizado, tenemos el efecto (correlación, en este caso en valores z de Fisher) en el eje X, así como su error estándar en el eje Y. La línea punteada vertical representa el efecto meta-analizado que hemos encontrado, así que podemos ver los estudios que encontraron un efecto mayor (derecha de la línea punteada) o menor (izquierda) de éste. A primera vista no parece haber mucha asimetría, pero es importante tener en cuenta que es un análisis muy subjetivo.

De nuevo, 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 3.4.2, a continuación), transformando los tamaños de efecto de regreso a r de Pearson (`x_trans_function = tanh`), y con los títulos de los ejes en español.

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

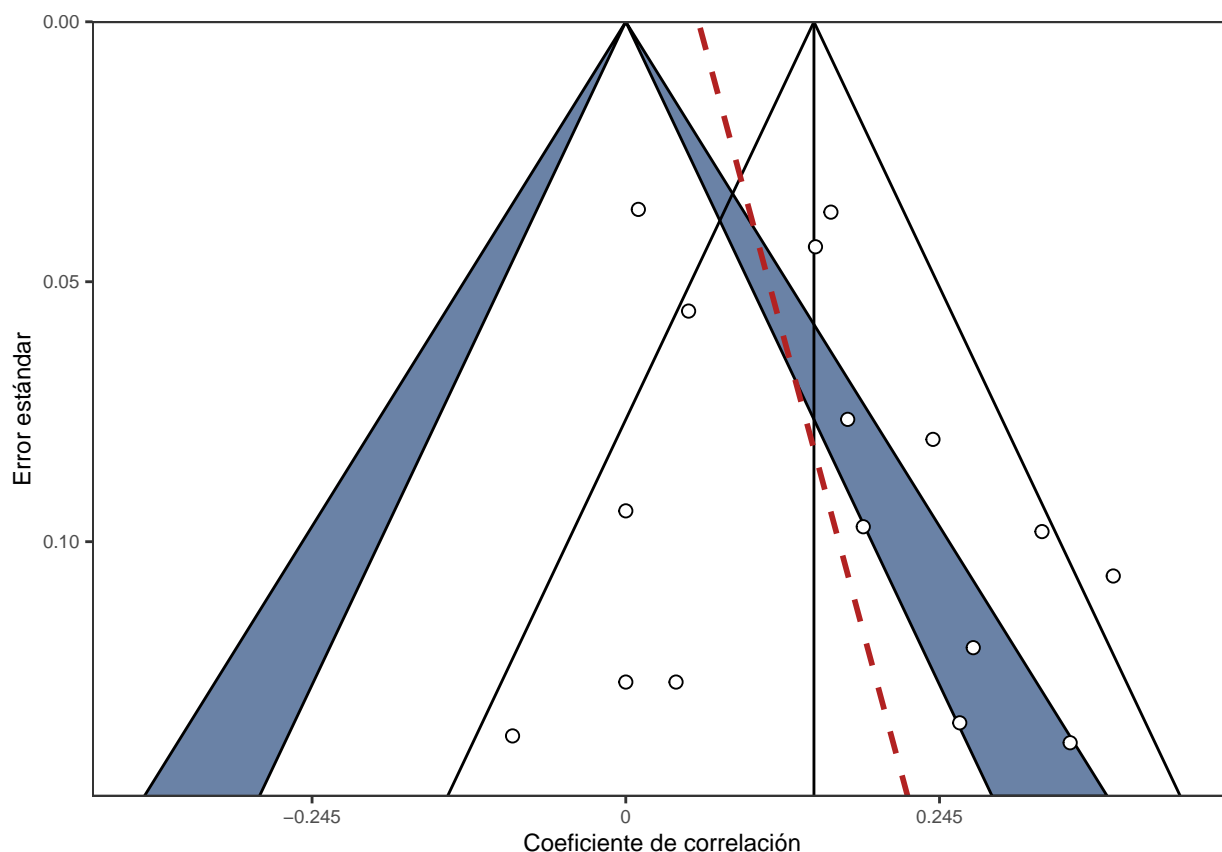


Figura 6. Funnel plot creado con `metaviz`. En azul, se representa el área donde estudios, según su error (y su tamaño de muestra), tendrían un efecto significativo al 5 % (i.e. $p > 0.05$), y fuera de ésta, donde tendrían un efecto significativo al 1 % (i.e. $p > 0.01$). La línea negra vertical representa 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. A continuación, muestro dos versiones de esta función. En ambos casos, agregué el efecto *real* encontrado con el 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,
  ylab = "Error estándar",
  xlab = "Coeficiente de correlación")

# B. Escala de poder continua
viz_sunset(res,
  contours = TRUE,
  x_trans_function = tanh,
  power_contours = "continuous",
  ylab = "Error estándar",
  xlab = "Coeficiente de correlación")
```

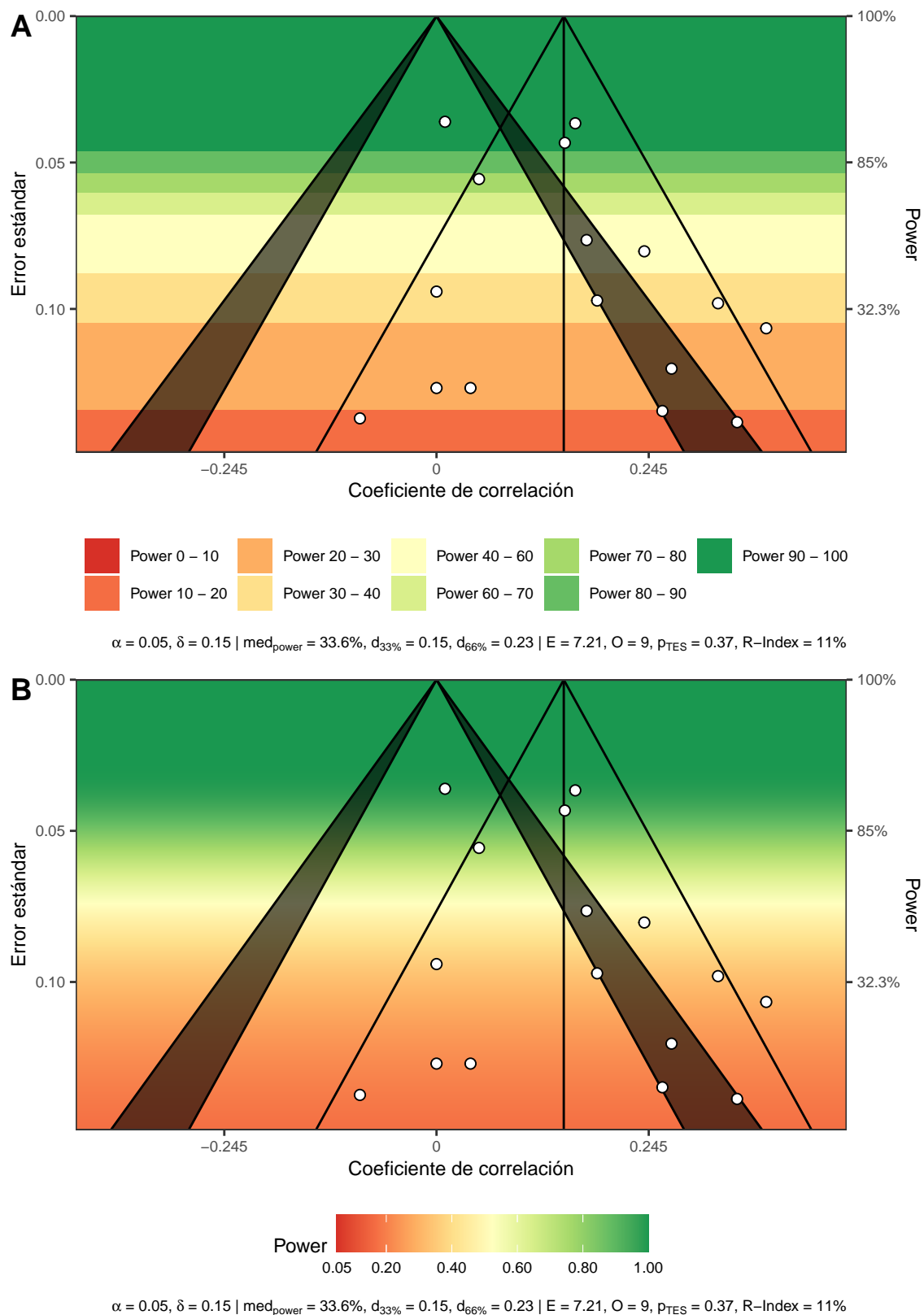


Figura 7. Dos versiones de funnel plot creados con [metaviz](#), usando la función `viz-sunset`, que estima el poder de cada estudio para detectar un efecto de interés. **A.** Poder representado por bandas discretas de color. **B.** Poder representado de manera continua en una escala de color. En ambos casos, y tal como en la Fig. 6, 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.

3.4.2. Regresión de Egger

Para hacer una prueba formal de sesgo de estudios pequeños, 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 5

```
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 5, explicaré cómo evaluar si hay sesgo de publicación).

4. Meta-análisis de correlación con moderador

4.1. Ejemplo 1: Moderación de la edad promedio de los participantes

Primero, y como ejemplo, vamos a ver si la edad (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 estudios⁴, lo que podría moderar (afectar) la asociación entre diligencia (*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 6

```
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:
```

⁴De 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.

```

      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) 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 columna `pval` para el efecto de `meanage` (0.232).

4.1.1. 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 mejorado (tal como el ejemplo de la figura 3), pero es importante tener en cuenta que esta opción no creará un resumen del meta-análisis.

```

# forest plot con anotaciones adicionales
par(mar=c(4,0,0,0))
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)
text(x = 1, y = 18, labels = "Peso", pos = 2)
text(x = 1.6, y = 18, labels = "Corr. [95% IC]", pos = 2)

```

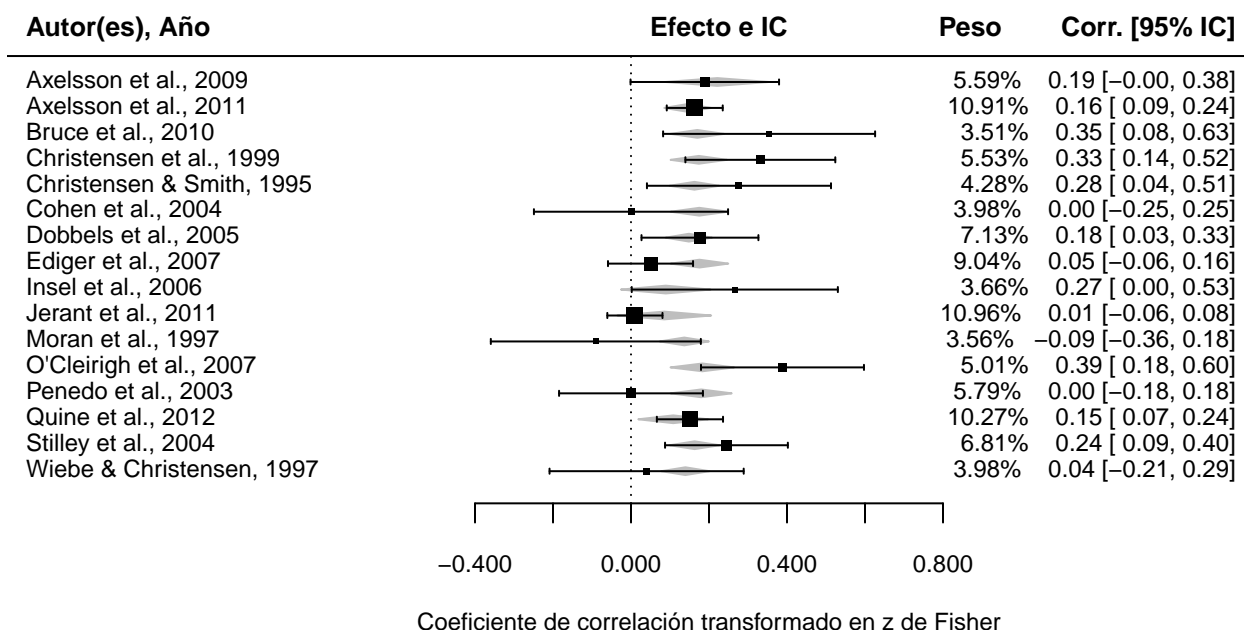


Figura 8. Forest plot básico de `metafor`, para un meta-análisis incluyendo la edad promedio de los participantes como moderador. En la ilustración gráfica, además de los efectos originales, se puede ver el efecto de cada estudio estimado cuando se incluye el moderador como polígonos (diamantes) de color gris. Sin embargo, ya no obtenemos una fila al final representando el efecto promediado del meta-análisis, ya que no tenemos un solo efecto.

Es importante tener en cuenta que la función `viz_forest` del paquete `metaviz` tendrá problemas para crear un *forest plot* de un meta-análisis con moderadores).

De manera similar, podemos obtener un *funnel plot* de nuestro meta-análisis con moderador, pero éste nos mostrará, en vez de los coeficientes de correlación (transformados a z de Fisher), los valores residuales de cada estudio (es decir, qué tanto se alejan del resultado de nuestro meta-análisis):

```
funnel(res.modage)
```

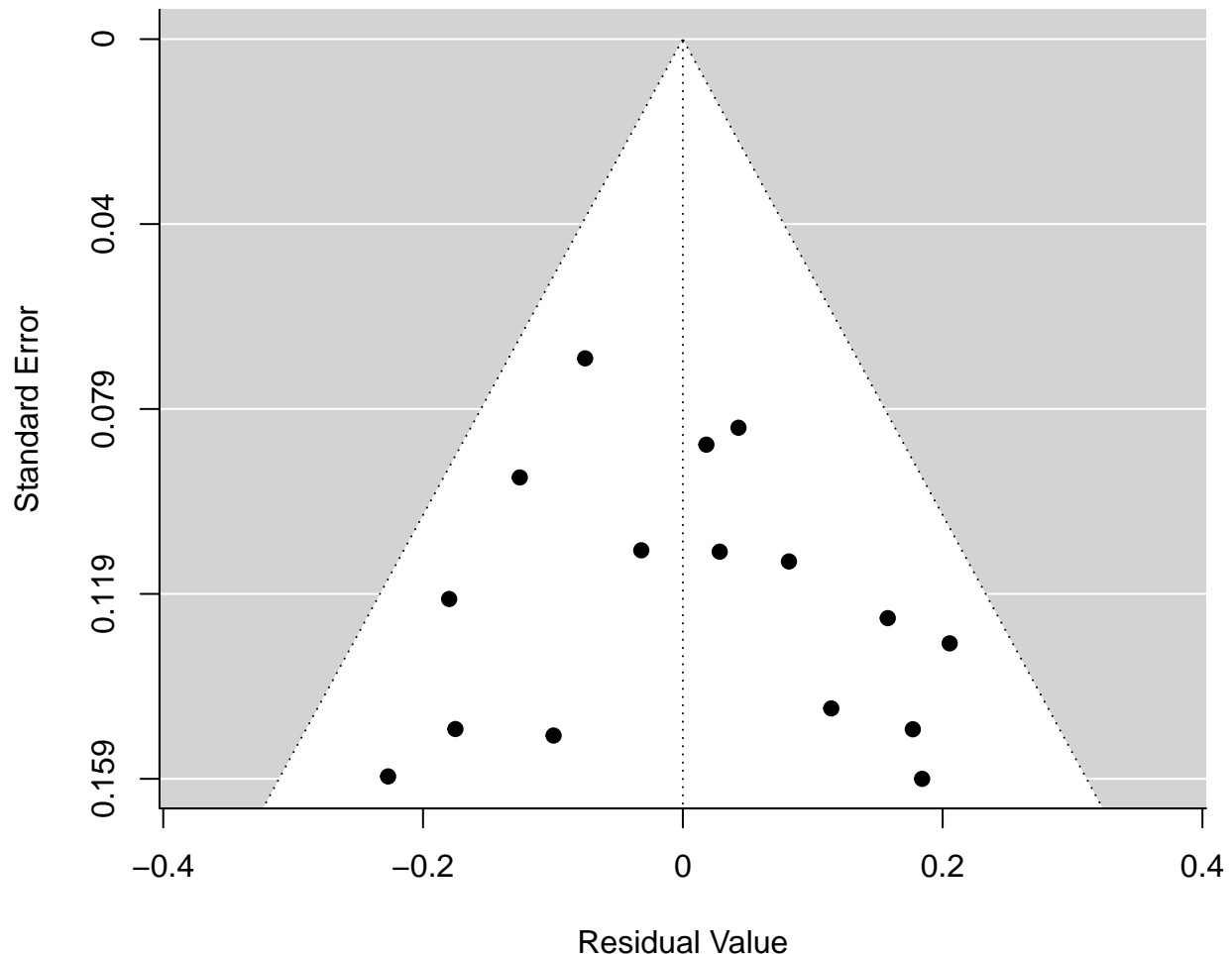


Figura 9. Funnel plot básico de `metafor`, para un meta-análisis incluyendo la edad promedio de los participantes como moderador. La línea punteada vertical representa el efecto meta-analizado que hemos encontrado, así que podemos ver los estudios que encontraron un efecto mayor (derecha de la línea punteada) o menor (izquierda) de éste.

4.2. Ejemplo 2: Moderación de la calidad de los estudios meta-analizados

La base de datos con tiene una medida de calidad metodológica de los estudios (variable `quality`). Dicha calidad, también podría moderar la asociación entre diligencia (*conscientiousness*) y adherencia a la medicación prescrita. Siguiendo los mismos pasos, puedo hacer éste análisis, pero voy a asignar este meta-análisis a un objeto llamado `res.modq` para diferenciarlo de los demás.

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

Consola de R: Output 7

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

tau^2 (estimated amount of residual heterogeneity):    0.0078 (SE = 0.0057)
tau (square root of estimated tau^2 value):           0.0884
I^2 (residual heterogeneity / unaccounted variability): 57.79%
H^2 (unaccounted variability / sampling variability):  2.37
R^2 (amount of heterogeneity accounted for):           3.73%

Test for Residual Heterogeneity:
QE(df = 14) = 30.4205, p-val = 0.0067

Test of Moderators (coefficient 2):
QM(df = 1) = 0.6393, p-val = 0.4240

Model Results:

      estimate      se      zval      pval      ci.lb      ci.ub
intrcpt      0.2082  0.0796   2.6149  0.0089   0.0521  0.3643  **
quality     -0.0312  0.0391  -0.7995  0.4240  -0.1078  0.0453

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

De nuevo, encontramos que éste moderador (**quality**), al igual que la edad promedio (**meanage**), no tiene un efecto significativo, como se puede ver en la columna **pval** para el efecto de **quality** (0.424).

Por supuesto, *forest plots* y *funnel plots* pueden ser creados, tal y como describí en la sección 4.1.1.

4.3. Ejemplo 3: Moderación de las controles usados en cada estudio meta-analizado

Como último ejemplo, voy a mirar si el hecho de que los estudios tengan variables que fueron controladas, modera la asociación entre diligencia (*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.mes**. Son embargo, dado que la variable que contiene esta información (**controls**) es un factor, pero no está definido como tal, debo hacerlo en la usando la función **factor** al ingresar el argumento **mods** (i.e. **mods = ~factor(controls)**).

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

Consola de R: Output 8

```
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      0.0167  0.0296   0.5635  0.5731  -0.0413  0.0746
factor(controls)none 0.1621  0.0361   4.4858 <.0001   0.0913  0.2329  ***

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

En éste caso, a diferencia de los ejemplos de moderación anteriores, la variable moderadora (`controls`) sí tiene un efecto significativo, como se puede ver en la columna `pval` para el efecto de `factor(controls)none` (<0.001), y en los asteriscos que aparecen al final de esa fila (`***`).

Por supuesto, *forest plots* y *funnel plots* pueden ser creados, tal y como describí en la sección 4.1.1.

5. Sesgo de publicación (*publication bias*)

Para determinar el sesgo de publicación, se puede usar la función `weightfunct` del paquete `{weightr}`, que nos permite “estimar tanto el modelo de función de peso para el sesgo de publicación que se publicó originalmente en Vevea y Hedges (1995) como la versión modificada presentada en Vevea y 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 al del estudio original (0.1486), dado que usa un método ligeramente diferente.

Nos da valores de heterogeneidad τ^2 , τ y Q .

Pero lo más importante, es que nos da los resultados 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 (ver SESGO DE PUBLICACIÓN); estudios con valores p mayores a 0.05 (o 0.10), tienen menos probabilidad de ser 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, a *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. 5, 6 y 7) y la regresión de Egger (sección 3.4.2), sugerían lo contrario.

6. Poder estadístico del meta-análisis

En esta sección explicaré cómo hacer un análisis de poder de un meta-análisis; la idea de esto 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 más mayor. Si nouviésemos 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), que permite calcular y visualizar el poder estadístico de un meta-análisis para detectar un rango de posibles efectos *reales*.

6.1. Instalación de metameta

El paquete `metameta` se debe instalar desde GitHub⁵ 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")
```

6.2. Análisis de poder

Una vez instalado, podemos cargar el paquete.

```
library(metameta)
```

Como datos, necesitamos no solamente los tamaños de datos 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*.

Debemos asumir el efecto real, pero esto nunca podemos saberlo. En este caso, asumimos un efecto 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.

⁵GitHub 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 no es ni mucho menos específico para paquetes de R.

```

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_dat <- power$dat
power_dat

```

Consola de R: Output 10

	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 power_es08						
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				

6.3. 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 0.4, lo que está muy por encima de nuestras estimaciones de un efecto real (0.15 en nuestro meta-análisis)

original, 0.09 en nuestro meta-análisis con pesos ajustados)

```
power_list <- list(power$power_median_dat)
power.plot <- firepower(power_list)
```

```
power.plot
```

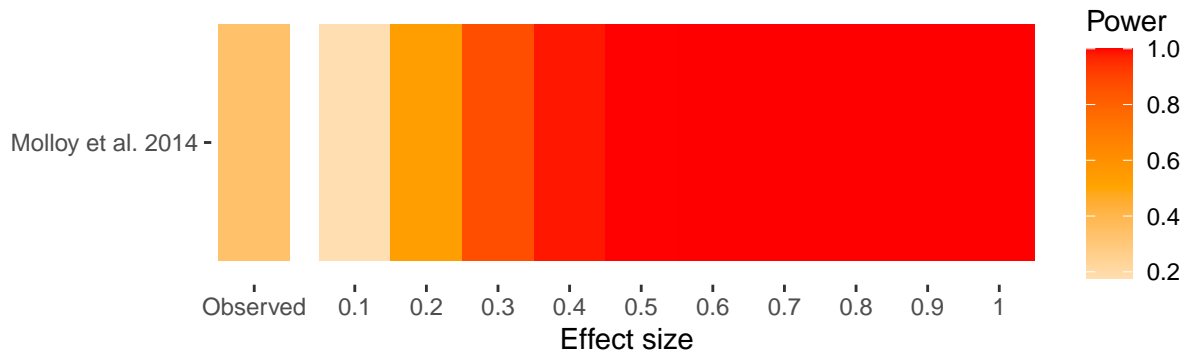


Figura 10. Fireplot básico de [metameta](#), para un análisis de poder de nuestro meta-análisis. *Observed* hace referencia al tamaño de efecto observado en nuestro meta-análisis original; en este caso, 0.15.

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”⁶. 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))
```

⁶Para 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`.

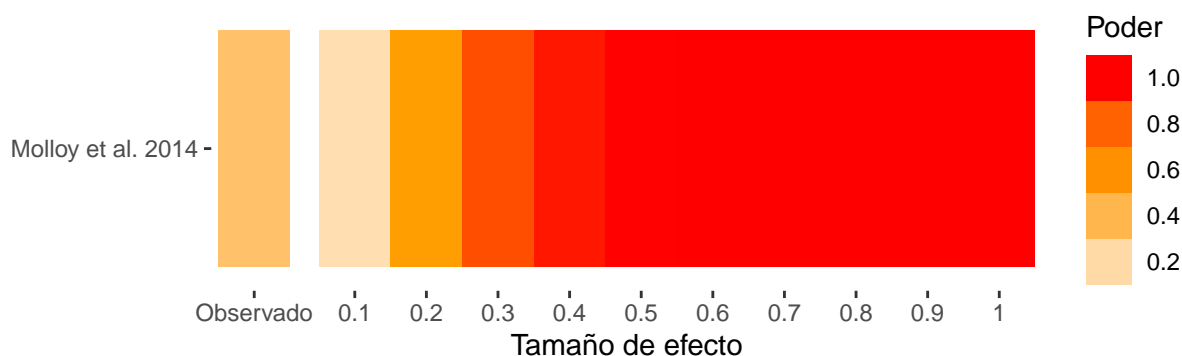


Figura 11. Fireplot básico de `metameta`, para un análisis de poder de nuestro meta-análisis, con el texto traducido a español y con la leyenda en una escala discreta para facilitar su lectura. *Observado* hace referencia al tamaño de efecto observado en nuestro meta-análisis original (en este caso, 0.15).

APÉNDICE 1: 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), o *p-curve* (curva de valores *p*), 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*.

Referencias

- 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>
- 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>
- 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>
- 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/>
- 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*. [Archivo de Vídeo]. YouTube. https://youtu.be/ntBbkOn9D_o
- Molloy, 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*. [Archivo de Vídeo]. YouTube. <https://youtu.be/1H4VZMTEZSc>
- Quintana, D. S. (2022). *Metameta: A suite of tools to re-evaluate published meta-analyses*. <https://github.com/dsquintana/metameta>
- 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
- 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>