

# The metaplot Gallery

Paul Murrell  
University of Auckland

David J. Scott  
University of Auckland

---

## Abstract

This gallery exhibits the capabilities of the **metaplot** package and also serves as a test routine for the package.

*Keywords:* meta-analysis, forest plots, **metaplot**, **rmeta**, **meta**, **metafor**, R .

---

## 1. Introduction

This document gives examples of the use of the package **metaplot** to produce forest plots using output from meta-analysis packages available on CRAN.

It also serves as a regression test for the package. Because the output from **metaplot** is graphical, the usual test procedures for evaluating numerical output are not appropriate.

The **metaplot** package was first developed to draw forest plots using meta-analysis results from the **meta** package, so in many cases the examples use data from that package.

## 2. The drawMeta() function

The basic function to draw forest plots is the **drawMeta()** function, which is generic. We start with a simple example of the use of this function, using output from the function **metabin()** from the package **meta**.

Analysis of binary outcome data by the package **meta** produces an object of class `c('metabin', 'meta')`.

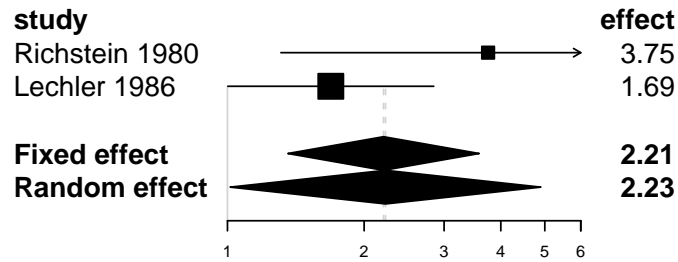
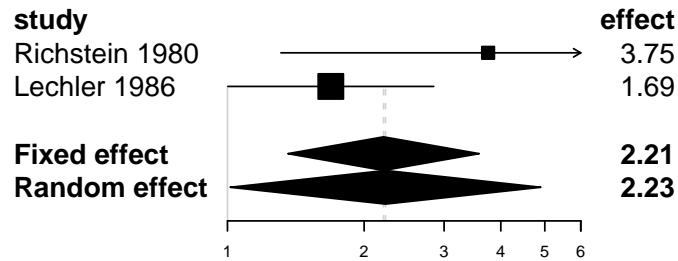
We first initialize, then perform a meta-analysis using the **meta** package. The generic functions **meta2DF()** and **metaDF2Matrix** convert the output from **metabin()** to a data frame, and from a data frame to a matrix respectively. The actual functions carrying out these transformations are **meta2DF.metabin()** and **metaDF2Matrix.metabinDF()**, which respectively produce objects of class `'metabinDF'` and `'metabinM'`.

```
library(meta)
```

```
Warning: package 'meta' was built under R version 3.1.2
```

```
Loading 'meta' package (version 3.8-0).
```

```
library(metaplot)
```

Figure 1: Basic forest plot from `drawMeta()`Figure 2: Basic forest plot from `drawMeta()`

Loading required package: *grid*

```
binary <- metabin(c(12, 16), c(16, 20), c(3, 9), c(15, 19),
                 studlab = c("Richstein 1980", "Lechler 1986"))
df <- meta2DF(binary)
matrix <- metaDF2Matrix(df, stat = NULL)
```

Then the forest plot shown in Figure 1 is produced using the basic `drawMeta()` function:

```
drawMeta(matrix)
```

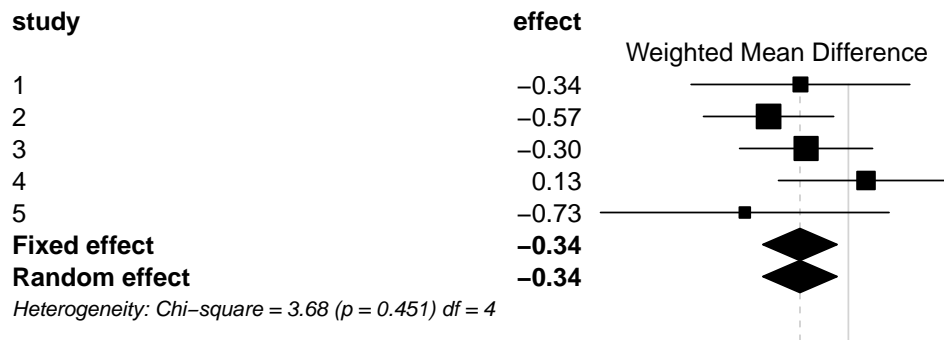
Loading required package: *rmeta*

Since `drawMeta()` is generic, the function which produces the plot is actually `drawMeta.metabinM`.

It is easy to change the location of the columns relative to the plot using an argument to `drawMeta()` with the result shown in Figure 2.

```
drawMeta(matrix, plotCol = 2)
```

For continuous outcome data, the `metacont()` function from the 'meta' package produces an object of class `c('metacont', 'meta')`. This object is converted to a data frame then a

Figure 3: Forest plot from `drawMeta()` with continuous outcome data

matrix after which it can be plotted by a call to the generic `drawMeta`. Here is an example of this procedure when the data comprises a single group only. First the data is retrieved and transformed.

```
data(Fleiss93cont)
meta1 <- metacont(n.e, mean.e, sd.e, n.c, mean.c,
                  sd.c, data = Fleiss93cont, sm = "SMD")
Data <- meta2DF(meta1)

matrix <- metaDF2Matrix(Data,
  order = c("study", "effect"),
  roundCols = c("effect" = 2),
  hgap = 2)
```

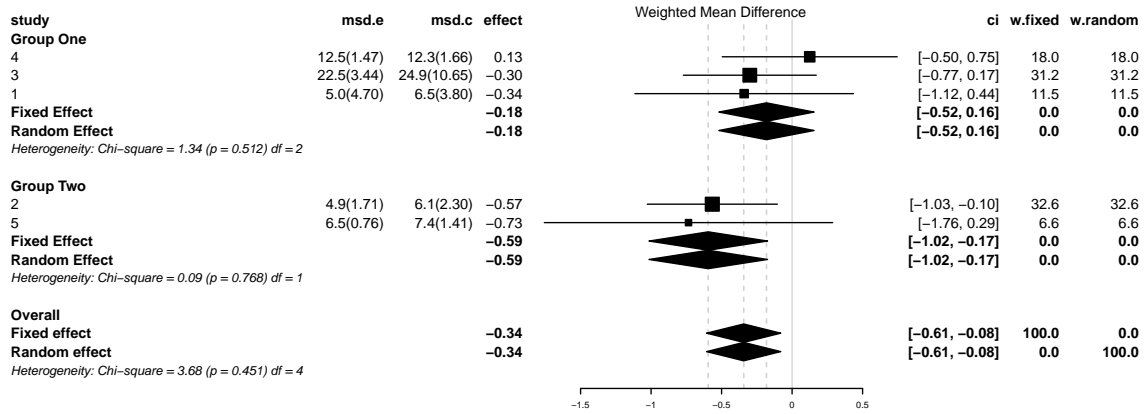
Then the plot can be drawn.

```
drawMeta(matrix,
  plotCol = 3,
  plotHead = "Weighted Mean Difference")
```

It is also possible to produce a forest plot of data where there are groups

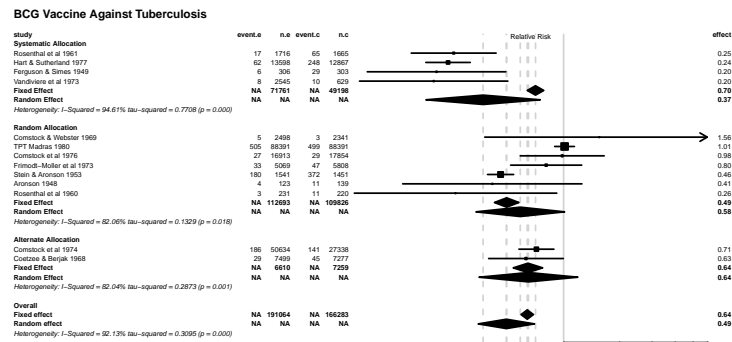
```
meta <- metacont(n.e, mean.e, sd.e, n.c, mean.c, sd.c,
  data = Fleiss93cont, sm = "SMD",
  byvar = c(1,2,1,1,2), bylab = "group")
Data <- meta2DF(meta, title = "Mental Health Treatment",
  rowOrder = "effect", decreasing = TRUE)
matrix <- metaDF2Matrix(Data,
  groupLab = c("Group One", "Group Two", "Overall"),
  order = c("study", "msd.e", "msd.c", "effect",
    "ci", "w.fixed", "w.random"),
  roundCols = c("effect" = 2, "w.fixed" = 1,
    "w.random" = 1),
  hgap=c(2,11,18), vgap = 5)
```

## Mental Health Treatment

Figure 4: Forest plot from `drawMeta()` with grouped continuous outcome data

```
drawMeta(matrix,
  plotCol = 6,
  plotHead = "Weighted Mean Difference")
```



Figure 5: Forest plot from `drawMeta()` with continuous outcome data

```

"Random Allocation",
"Alternate Allocation",
"Overall"),
order = c("study", "event.e", "n.e",
          "event.c", "n.c", "effect"),
roundCols = c("effect" = 2))

```

Then the plot can be drawn.

```

drawMeta(matrix,
          plotCol = 6,
          plotHead = "Relative Risk")

```

### Affiliation:

David J Scott

Department of Statistics

The University of Auckland

PB 92019

Auckland 1142

New Zealand

E-mail: [d.scott@auckland.ac.nz](mailto:d.scott@auckland.ac.nz)

URL: <http://statistics.auckland.ac.nz/~dscott>