Presentation 5, Group 2: Classical Meta-Analysis of Binomial Outcomes

Jan Hohenheim

2024-05-14

```
# Visual stuff
library(tidyverse)
library(broom)
library(ggthemes)
library(glue)
library(knitr)

# Meta analysis
library(lme4)
library(metafor)

theme_set(theme_solarized_2())
```

Source for theory: *Chang BH, Hoaglin DC. Meta-Analysis of Odds Ratios: Current Good Practices*

Data

Source: *Pagliaro L, D'Amico G, Sorensen TI, et al. Prevention of first bleeding in cirrhosis. A meta-analysis of randomized trials of nonsurgical treatment*

```
dat.raw <- tibble(
  trial_id = 1:19,
  n_placebo = c(36, 53, 18, 22, 46, 60, 60, 69, 41, 20, 41, 35, 138, 51, 72, 16, 28, 19, 24),
  n_active = c(35, 56, 16, 23, 49, 53, 53, 71, 41, 21, 42, 33, 143, 55, 73, 13, 21, 18, 22),
  deaths_placebo = c(14, 29, 6, 6, 34, 14, 27, 26, 19, 2, 18, 21, 23, 24, 14, 4, 8, 6, 5),
  deaths_active = c(2, 12, 6, 4, 30, 13, 15, 16, 10, 0, 18, 20, 46, 19, 18, 2, 6, 7, 5)
)
dat.raw |>
  kable(caption = "Raw data")
```

Table 1: Raw data

trial_id	n_placebo	n_active	deaths_placebo	deaths_active
1	36	35	14	2
2	53	56	29	12
3	18	16	6	6
4	22	23	6	4
5	46	49	34	30
6	60	53	14	13
7	60	53	27	15
8	69	71	26	16
9	41	41	19	10
10	20	21	2	0
11	41	42	18	18
12	35	33	21	20
13	138	143	23	46
14	51	55	24	19
15	72	73	14	18
16	16	13	4	2
17	28	21	8	6
18	19	18	6	7
19	24	22	5	5

```
offset <- 0.001

dat <- dat.raw |>
  mutate(
    trial_id = as.factor(trial_id),
  ) |>
  mutate(
    p_placebo = (deaths_placebo + offset) / (n_placebo + 2 * offset),
    p_active = (deaths_active + offset) / (n_active + 2 * offset)
  ) |>
  mutate(
    log_odds_control = log(p_placebo / (1 - p_placebo)),
    log_odds_active = log(p_active / (1 - p_active))
  ) |>
  mutate(
    log_or = log_odds_active - log_odds_control,
```

```
    var_log_or =
      1 / (deaths_placebo + offset) +
      1 / (n_placebo - deaths_placebo + offset) +
      1 / (deaths_active + offset) +
      1 / (n_active - deaths_active + offset)
  )
```

```
dat |>
  filter(trial_id != 10) |>
  ggplot(aes(x = log_or, y = trial_id)) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  geom_point(size = 3) +
  geom_errorbarh(aes(
    xmin = log_or - 1.96 * sqrt(var_log_or),
    xmax = log_or + 1.96 * sqrt(var_log_or)), height = 0) +
  labs(
    title = "Forest plot of the included studies",
    x = "Log odds ratio",
    y = "Study ID"
  )
```

Methods

For binomial outcomes, the measured effects are usually transformed to the log-odds ratio.

$$y_i := \log\left(\frac{p_i}{1-p_i}\right)$$

Where p_i is the probability of the event in group i.

The variance of the log-odds ratio is approximated by

$$\mathrm{Var}(y_i) \approx \frac{1}{a} + \frac{1}{b} + \frac{1}{c} + \frac{1}{d}$$

Where a, b, c, d are the number of events and non-events in the two groups:

	Responder	Non-Responder
Active	a	b
Placebo	c	d

Fixed effect

The measurement y is an estimator of the underlying effect μ with an error term ϵ. μ is assumed to be the same for all studies, while ϵ varies between studies.

$$y_i = \mu + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

The estimator $\hat{\mu}$ is the weighted average of the individual studies' estimates:

$$\hat{\mu} = \frac{\sum \kappa_i y_i}{\sum \kappa_i}, \quad \kappa_i := \frac{1}{\sigma_i^2}$$

Where σ_i^2 is the variance inside the study i.

If the number of subjects in each group is large, the variance of $\hat{\mu}$ can be approximated by

$$\mathrm{Var}(\hat{\mu}) \approx \frac{1}{\sum \kappa_i}$$

In R, all of this is done quite simply with the `rma` function from the `metafor` package.

```
rma.fixed <- rma(
  yi = dat$log_or,
  vi = dat$var_log_or,
  method = "FE") # FE stands for "Fixed Effect"

rma.fixed |> summary()
rma.fixed_mean <- rma.fixed$beta[[1]]
rma.fixed_se <- rma.fixed$se[1]
```

```
##
## Fixed-Effects Model (k = 19)
##
##   logLik  deviance       AIC       BIC      AICc
## -29.7597   41.1666   61.5195   62.4639   61.7548
```

```
##
## I^2 (total heterogeneity / total variability):   56.28%
## H^2 (total variability / sampling variability):  2.29
##
## Test for Heterogeneity:
## Q(df = 18) = 41.1666, p-val = 0.0014
##
## Model Results:
##
## estimate      se     zval    pval    ci.lb    ci.ub
##  -0.2528  0.1120  -2.2563  0.0241  -0.4724  -0.0332  *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Random effects

μ can now also vary between studies:

$$y_i = \mu_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Consequently, we now have to consider two sources of variation: The variance inside a study σ_i^2 and the variance between studies τ^2:

$$\mathrm{Var}(y_i) \approx \sigma_i^2 + \tau^2$$

The new estimator $\hat{\mu}'$ looks very similar to the fixed effect estimator, it only uses the new precision

$$\hat{\mu}' = \frac{\sum \kappa_i' y_i}{\sum \kappa_i'}, \quad \kappa_i' := \frac{1}{\sigma_i^2 + \tau^2}$$

Calculating the inter-study variance τ^2 not as straightforward. `rma` lets you choose between different estimators, but the default is already fairly unbiased.

The mixed effects model is more flexible than the fixed effect model, but assumes that y_i and σ_i^2 are independent, which is probably not the case for this kind of analysis. As such, no matter which τ^2 estimator is used, there will always be a bias.

The R code looks very similar to the fixed effect model, with the difference being that we omit the `method` argument. The default already uses a random effects model.

```
rma.random <- rma(
  yi = dat$log_or,
  vi = dat$var_log_or)

rma.random |> summary()
rma.random_mean <- rma.random$beta[[1]]
rma.random_se <- rma.random$se[1]
```

```
##
## Random-Effects Model (k = 19; tau^2 estimator: REML)
##
##   logLik  deviance       AIC       BIC      AICc
## -23.4206   46.8411   50.8411   52.6219   51.6411
##
## tau^2 (estimated amount of total heterogeneity): 0.2844 (SE = 0.1843)
## tau (square root of estimated tau^2 value):      0.5333
## I^2 (total heterogeneity / total variability):   53.80%
## H^2 (total variability / sampling variability):  2.16
##
## Test for Heterogeneity:
## Q(df = 18) = 41.1666, p-val = 0.0014
##
## Model Results:
##
## estimate      se     zval    pval    ci.lb   ci.ub
##  -0.3295  0.1757  -1.8754  0.0607  -0.6739  0.0149  .
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using sample sizes directly

If one has access to the sample size, the model can use the numbers of events and participants directly. In this case, one can do a simple logistic regression. The response variable is the number of successes and events, and the explanatory variables are the trial number and whether the treatment was active or not.

```
dat.glm <- tibble(
  trial_id = rep(dat$trial_id, 2),
  treatment = rep(c("placebo", "active"), each = nrow(dat)),
  deaths = c(dat$deaths_placebo, dat$deaths_active),
  n = c(dat$n_placebo, dat$n_active)
)
dat.glm |>
  kable(caption = "Data for logistic regression")
```

Table 3: Data for logistic regression

trial_id	treatment	deaths	n
1	placebo	14	36
2	placebo	29	53
3	placebo	6	18
4	placebo	6	22
5	placebo	34	46
6	placebo	14	60
7	placebo	27	60
8	placebo	26	69
9	placebo	19	41
10	placebo	2	20
11	placebo	18	41
12	placebo	21	35
13	placebo	23	138
14	placebo	24	51
15	placebo	14	72
16	placebo	4	16
17	placebo	8	28
18	placebo	6	19
19	placebo	5	24
1	active	2	35
2	active	12	56
3	active	6	16
4	active	4	23
5	active	30	49
6	active	13	53
7	active	15	53
8	active	16	71
9	active	10	41
10	active	0	21
11	active	18	42
12	active	20	33
13	active	46	143
14	active	19	55
15	active	18	73
16	active	2	13
17	active	6	21

trial_id	treatment	deaths	n
18	active	7	18
19	active	5	22

```
glm.fixed <- glm(
  cbind(deaths, n - deaths) ~ trial_id + treatment,
  data = dat.glm,
  family = binomial)

glm.fixed |> summary()

glm.fixed_placebo <- tidy(glm.fixed) |>
  filter(term == "treatmentplacebo")
glm.fixed_mean <- -glm.fixed_placebo$estimate
glm.fixed_se <- glm.fixed_placebo$std.error
```

```
##
## Call:
## glm(formula = cbind(deaths, n - deaths) ~ trial_id + treatment,
##     family = binomial, data = dat.glm)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.38568    0.29069  -4.767 1.87e-06 ***
## trial_id2         0.73788    0.34681   2.128  0.03337 *
## trial_id3         0.62480    0.45864   1.362  0.17311
## trial_id4        -0.01288    0.45826  -0.028  0.97758
## trial_id5         1.97540    0.35938   5.497 3.87e-08 ***
## trial_id6         0.06967    0.36030   0.193  0.84667
## trial_id7         0.70586    0.34502   2.046  0.04077 *
## trial_id8         0.39304    0.33932   1.158  0.24672
## trial_id9         0.63640    0.36685   1.735  0.08278 .
## trial_id10       -1.73377    0.77910  -2.225  0.02606 *
## trial_id11        0.97614    0.36096   2.704  0.00684 **
## trial_id12        1.65802    0.37779   4.389 1.14e-05 ***
## trial_id13        0.11725    0.31663   0.370  0.71116
## trial_id14        0.86395    0.34688   2.491  0.01275 *
## trial_id15       -0.02400    0.34815  -0.069  0.94505
## trial_id16       -0.12211    0.54021  -0.226  0.82117
## trial_id17        0.30130    0.42592   0.707  0.47931
## trial_id18        0.62236    0.44734   1.391  0.16415
## trial_id19       -0.05055    0.45737  -0.111  0.91199
## treatmentplacebo  0.28655    0.10878   2.634  0.00843 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 185.74  on 37  degrees of freedom
## Residual deviance:  47.80  on 18  degrees of freedom
## AIC: 230.01
##
## Number of Fisher Scoring iterations: 4
```

Note that the above again assumes that μ is the same for all studies. We can extend this model to include a random effect for the study:

```
numeric_treatment <-
  ifelse(dat.glm$treatment == "active", 1, 0)

glm.random <- glmer(
  cbind(deaths, n - deaths) ~
    trial_id +
    treatment +
    (-1 + numeric_treatment | as.numeric(trial_id)),
  data = dat.glm,
  family = binomial,
  nAGQ = 7)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0039628 (tol = 0.002, component 1)
```

```
glm.random |> summary(correlation=F)

glm.random_placebo <- summary(glm.random)$coefficients["treatmentplacebo",]
glm.random_mean <- -glm.random_placebo["Estimate"][[1]]
glm.random_se <- glm.random_placebo["Std. Error"][[1]]
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 7) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## cbind(deaths, n - deaths) ~ trial_id + treatment + (-1 + numeric_treatment |
##     as.numeric(trial_id))
##    Data: dat.glm
##
##      AIC      BIC   logLik deviance df.resid
##     81.5    115.9    -19.7     39.5       17
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.45860 -0.31914  0.00545  0.36480  1.12671
##
## Random effects:
##  Groups                Name              Variance Std.Dev.
##  as.numeric(trial_id) numeric_treatment 0.1907   0.4367
## Number of obs: 38, groups:  as.numeric(trial_id), 19
##
## Fixed effects:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.20849    0.34626  -3.490 0.000483 ***
## trial_id2         0.78898    0.40026   1.971 0.048705 *
## trial_id3         0.32515    0.52335   0.621 0.534409
## trial_id4        -0.22364    0.51564  -0.434 0.664492
## trial_id5         1.81790    0.42892   4.238 2.25e-05 ***
## trial_id6        -0.25620    0.42815  -0.598 0.549578
## trial_id7         0.56376    0.39921   1.412 0.157894
## trial_id8         0.26222    0.39435   0.665 0.506079
## trial_id9         0.53964    0.42202   1.279 0.201005
```

```
## trial_id10       -1.92586    0.81477  -2.364 0.018095 *
## trial_id11        0.67257    0.43243   1.555 0.119865
## trial_id12        1.35199    0.45032   3.002 0.002680 **
## trial_id13       -0.55506    0.41485  -1.338 0.180903
## trial_id14        0.68129    0.40922   1.665 0.095941 .
## trial_id15       -0.42744    0.42782  -0.999 0.317749
## trial_id16       -0.34949    0.58708  -0.595 0.551640
## trial_id17        0.01773    0.48431   0.037 0.970803
## trial_id18        0.29941    0.51761   0.578 0.562954
## trial_id19       -0.34567    0.52053  -0.664 0.506648
## treatmentplacebo  0.37365    0.15788   2.367 0.017953 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.0039628 (tol = 0.002, component 1)
```

Comparison

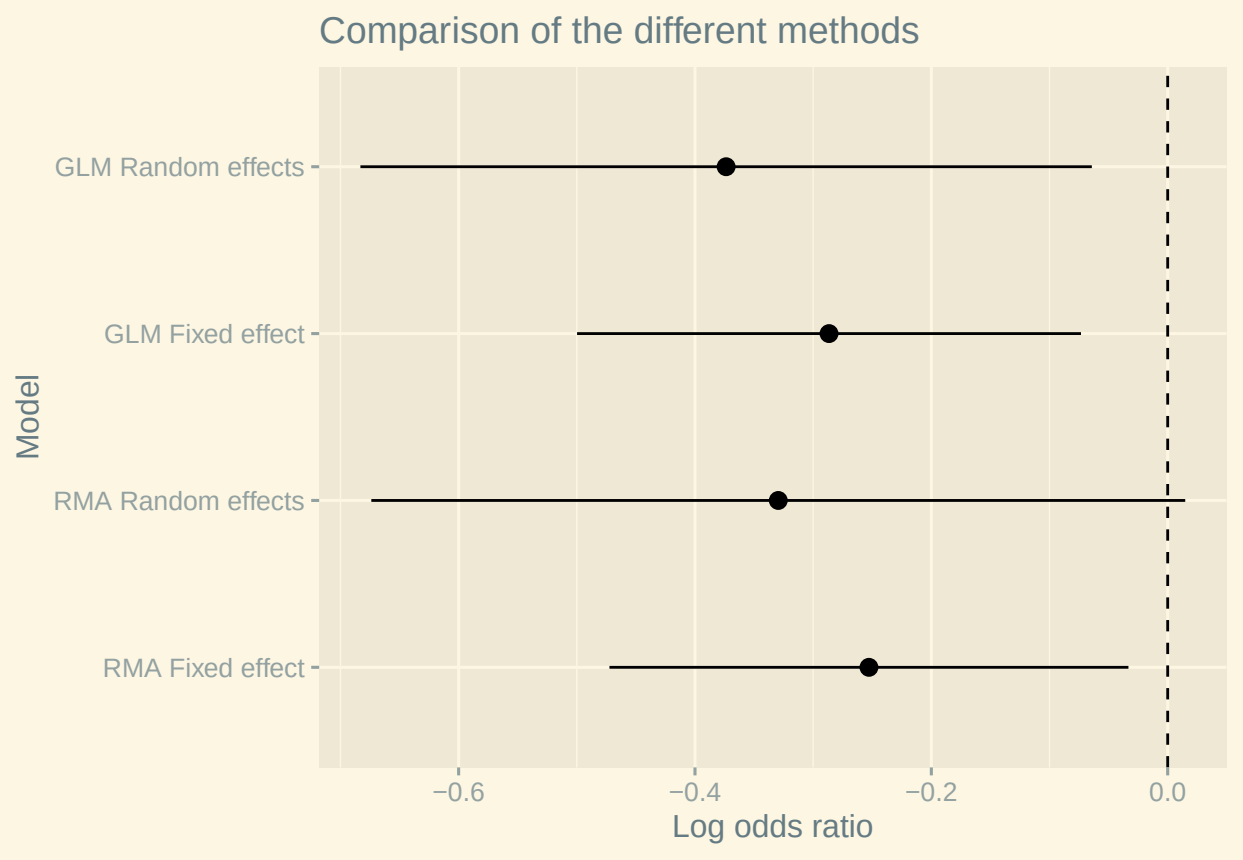
```
comparison <- tibble(
  model = c("RMA Fixed effect", "RMA Random effects", "GLM Fixed effect", "GLM Random effects"),
  mean = c(rma.fixed_mean, rma.random_mean, glm.fixed_mean, glm.random_mean),
  se = c(rma.fixed_se, rma.random_se, glm.fixed_se, glm.random_se)
)

comparison |> kable(caption = "Comparison of the different models")
```

Table 4: Comparison of the different models

model	mean	se
RMA Fixed effect	-0.2527747	0.1120326
RMA Random effects	-0.3295016	0.1756970
GLM Fixed effect	-0.2865532	0.1087809
GLM Random effects	-0.3736463	0.1578841

```
comparison |>
  ggplot(aes(x = model, y = mean, ymin = mean - 1.96 * se, ymax = mean + 1.96 * se)) +
  geom_pointrange() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  scale_x_discrete(limits = c(
    "RMA Fixed effect",
    "RMA Random effects",
    "GLM Fixed effect",
    "GLM Random effects")) +
  labs(
    title = "Comparison of the different methods",
    x = "Model",
    y = "Log odds ratio"
  ) +
  coord_flip()
```


Comparison of the different methods
GLM Random effects
GLM Fixed effect
RMA Random effects
RMA Fixed effect
Model
−0.6
−0.4
−0.2
0.0
Log odds ratio