Moderated univariate meta-analysis in MetaSEM

For metaSEM you generally create a different model per moderation (e.g. age, gender, education). You can think of each moderated model as a regression with a predictor. A regression without predictors (just the intercept) just tells you the mean (the baseline model).

To give the basic flow of what we're doing with metaSEM you can check out an analogy with regression:

Let's use mtcars and miles per galon (mpg)

Baseline model

```
mean(mtcars$mpg)
```

```
## [1] 20.09062
```

The baseline model's intercept just gets you the mean of mpg

```
baseline <-lm(mpg ~ 1 , data = mtcars)
coef(baseline)
```

```
## (Intercept)
##    20.09062
```

Moderated model

The moderated model gives you the relationship between another variable and miles per galon. E.g. the number of cylanders a car has.

Here are the average mpg per cylander

```
library(dplyr)
mtcars %>%
  group_by(cyl) %>%
  summarise(avg_mpg = mean(mpg))
```

```
## # A tibble: 3 x 2
##     cyl avg_mpg
##   <dbl>   <dbl>
## 1     4    26.7
## 2     6    19.7
## 3     8    15.1
```

Here's the same information using a regression (analagous to a moderated meta-analsis):

```
moderated_model <-lm(mpg ~ 1 + factor(cyl) , data = mtcars)
summary(moderated_model)
```

```
##
## Call:
## lm(formula = mpg ~ 1 + factor(cyl), data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2636 -1.8357  0.0286  1.3893  7.2364
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   26.6636     0.9718  27.437  < 2e-16 ***
## factor(cyl)6  -6.9208     1.5583  -4.441 0.000119 ***
## factor(cyl)8 -11.5636     1.2986  -8.905 8.57e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.223 on 29 degrees of freedom
## Multiple R-squared:  0.7325, Adjusted R-squared:  0.714
## F-statistic:  39.7 on 2 and 29 DF,  p-value: 4.979e-09
```

The intercept is the average mpg at the reference level (cyl == 4). We can see that when cyl == 6, mpg is 6.92 points lower than the reference level. When it’s 8, it’s lower again. These values when added to the intercept match our averages calculated above.

Check that moderated model improves baseline

We can then compare the moderated model to the baseline to see if it explains more variance.

```
anova(baseline, moderated_model)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ 1
## Model 2: mpg ~ 1 + factor(cyl)
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     31 1126.05
## 2     29  301.26  2    824.78 39.697 4.979e-09 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Given the p-value is tiny, we can see the moderated model improves on the baseline model.

metaSEM

We do a similar process with meta-analytic models. I’ll use a dataset in the metaSEM package you can play with.

In this dataset we are looking at the odds of women vs men getting grants. log-odds are the effect size (negative log-odds is less likely, positive log-odds is more likely). We use log-odds as they can be averaged together, whereas odds cannot.

```r
dat <- metaSEM::Bornmann07
```

Baseline model

```r
library(metaSEM)
baseline <- meta3(y = logOR, v = v, cluster = Cluster, data = dat)
summary(baseline)
```

```
##
## Call:
## meta3(y = logOR, v = v, cluster = Cluster, data = dat)
##
## 95% confidence intervals: z statistic approximation (robust=FALSE)
## Coefficients:
##             Estimate  Std.Error     lbound     ubound z value Pr(>|z|)
## Intercept -0.1007784  0.0401327 -0.1794371 -0.0221198 -2.5111  0.01203 *
## Tau2_2     0.0037965  0.0027210 -0.0015367  0.0091297  1.3952  0.16295
## Tau2_3     0.0141352  0.0091445 -0.0037877  0.0320580  1.5458  0.12216
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Q statistic on the homogeneity of effect sizes: 221.2809
## Degrees of freedom of the Q statistic: 65
## P value of the Q statistic: 0
##
## Heterogeneity indices (based on the estimated Tau2):
##                               Estimate
## I2_2 (Typical v: Q statistic)   0.1568
## I2_3 (Typical v: Q statistic)   0.5839
##
## Number of studies (or clusters): 21
## Number of observed statistics: 66
## Number of estimated parameters: 3
## Degrees of freedom: 63
## -2 log likelihood: 25.80256
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

The intercept (just like in the regression example) tells us the average log-odds for women getting grants. We can convert this to odds by taking its exponential

```r
exp(coef(baseline)[1])
```

```
## Intercept
## 0.9041333
```

For every man that gets a grant, only .9 women get one.

Moderated model

We can introduce predictor variables to our model to see if other variables are related to the odds of getting grants (just like regression). For example, perhaps the discipline of a grant proposal affects the odds of women vs men being successful?

Under the hood

For the lm function, you just needed to tell R the variable and it creates the underlying matrices used in calculations. E.g. for mpg ~ cyl this is (more or less) what r did before doing calculations

```
cyl6 <- as.numeric(mtcars$cyl == 6)
cyl8 <- as.numeric(mtcars$cyl == 8)
pred_matrix = cbind(cyl6, cyl8)

matrix <- cbind(y = mtcars$mpg, intercept = 1, cyl6, cyl8)
matrix[1:5,]
```

```
##         y intercept cyl6 cyl8
## [1,] 21.0         1    1    0
## [2,] 21.0         1    1    0
## [3,] 22.8         1    0    0
## [4,] 21.4         1    1    0
## [5,] 18.7         1    0    1
```

R then uses this matrix to work out the regression slopes for each of the predictors (including for the intercept).

In metaSEM you have to construct the matrix of the predictors yourself. metaSEM then stitches the outcome, clustering variables, and your predictor matrix together. This final matrix is used with the OpenMx package to get model results.

Creating the predictor matrix for metaSEM

These are the disciplines in our dataset

```
unique(dat$Discipline)
```

```
## [1] Physical sciences          Social sciences/humanities
## [3] Life sciences/biology      Multidisciplinary
## 4 Levels: Physical sciences ... Multidisciplinary
```

We leave off the first discipline ("Physical") because if a grant isn't in any of the other disciplines, it must be in the physical sciences. Adding an extra column with redundant information would give us really confusing results. What we're doing by leaving off the first level is the same as what is done in regression using categorical variables.

```
Physical <- as.numeric(dat$Discipline == 'Physical sciences')
Social <- as.numeric(dat$Discipline == 'Social sciences/humanities')
Life <- as.numeric(dat$Discipline == 'Life sciences/biology')
Multi <- as.numeric(dat$Discipline == 'Multidisciplinary')
```

```r
moderator_matrix <- cbind(Social, Life, Multi)
moderator_matrix[1:5,] # I just show the first five rows.
```

```
##      Social Life Multi
## [1,]      0    0     0
## [2,]      0    0     0
## [3,]      0    0     0
## [4,]      0    0     0
## [5,]      1    0     0
```

We then give that predictor matrix to metaSEM to perform the meta-regression.

```r
moderated_model <- meta3(y = logOR, v = v, x = moderator_matrix, cluster = Cluster, data = dat)
summary(moderated_model)
```

```
##
## Call:
## meta3(y = logOR, v = v, cluster = Cluster, x = moderator_matrix,
##     data = dat)
##
## 95% confidence intervals: z statistic approximation (robust=FALSE)
## Coefficients:
##             Estimate  Std.Error     lbound     ubound z value Pr(>|z|)
## Intercept -0.0238785  0.0764858 -0.1737878  0.1260308 -0.3122  0.75489
## Slope_1   -0.2278263  0.1148709 -0.4529692 -0.0026835 -1.9833  0.04733 *
## Slope_2   -0.1170489  0.0824846 -0.2787157  0.0446178 -1.4190  0.15589
## Slope_3    0.0091306  0.0994920 -0.1858701  0.2041313  0.0918  0.92688
## Tau2_2     0.0039094  0.0028395 -0.0016559  0.0094747  1.3768  0.16857
## Tau2_3     0.0071034  0.0064321 -0.0055033  0.0197101  1.1044  0.26944
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Q statistic on the homogeneity of effect sizes: 221.2809
## Degrees of freedom of the Q statistic: 65
## P value of the Q statistic: 0
##
## Explained variances (R2):
##                          Level 2 Level 3
## Tau2 (no predictor)    0.0037965  0.0141
## Tau2 (with predictors) 0.0039094  0.0071
## R2                     0.0000000  0.4975
##
## Number of studies (or clusters): 21
## Number of observed statistics: 66
## Number of estimated parameters: 6
## Degrees of freedom: 60
## -2 log likelihood: 20.07571
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

We can see the log-odds of physical sciences was -.02 (the intercept), Slope 1 to 3 are social sciences through multi-discipline. The slopes show the difference between physical science to a given level.

We can see this model's fit is not significantly better than the baseline model (discipline may not affect how likely women are to get grants).

```r
anova(moderated_model, baseline)
```

```
##                     base            comparison ep minus2LL df        AIC
## 1 Meta analysis with ML                  <NA>  6 20.07571 60  -99.92429
## 2 Meta analysis with ML Meta analysis with ML  3 25.80256 63 -100.19744
##     diffLL diffdf         p
## 1       NA     NA        NA
## 2 5.726842      3 0.1256832
```

If we wanted to just see the values for the slope without a reference level we could have just supplied all four categories and set the intercept to equal zero:

```r
moderator_matrix <- cbind(Physical, Social, Life, Multi)
moderated_model <- meta3(y = logOR, v = v, x = moderator_matrix,
                         intercept.constraints = 0, cluster = Cluster, data = dat)
summary(moderated_model)
```

```
##
## Call:
## meta3(y = logOR, v = v, cluster = Cluster, x = moderator_matrix,
##     data = dat, intercept.constraints = 0)
##
## 95% confidence intervals: z statistic approximation (robust=FALSE)
## Coefficients:
##           Estimate  Std.Error     lbound     ubound z value Pr(>|z|)
## Slope_1 -0.0238785  0.0764858 -0.1737878  0.1260308 -0.3122 0.754892
## Slope_2 -0.2517048  0.1016318 -0.4508995 -0.0525101 -2.4766 0.013263 *
## Slope_3 -0.1409274  0.0438924 -0.2269550 -0.0548999 -3.2107 0.001324 **
## Slope_4 -0.0147478  0.0638994 -0.1399885  0.1104928 -0.2308 0.817472
## Tau2_2   0.0039094  0.0028395 -0.0016559  0.0094747  1.3768 0.168573
## Tau2_3   0.0071034  0.0064321 -0.0055033  0.0197101  1.1044 0.269435
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Q statistic on the homogeneity of effect sizes: 221.2809
## Degrees of freedom of the Q statistic: 65
## P value of the Q statistic: 0
##
## Explained variances (R2):
##                           Level 2 Level 3
## Tau2 (no predictor)    0.0037965  0.0141
## Tau2 (with predictors) 0.0039094  0.0071
## R2                     0.0000000  0.4975
##
## Number of studies (or clusters): 21
## Number of observed statistics: 66
## Number of estimated parameters: 6
## Degrees of freedom: 60
## -2 log likelihood: 20.07571
## OpenMx status1: 0 ("0" or "1": The optimization is considered fine.
## Other values may indicate problems.)
```

Note slope 1 and the previous model's intercept are identical. Slopes 2:4 are changed as they are no longer relative to slope 1. They are relative to the intercept which we constrained to be zero (in other words they just give you the log odds for each of those disciplines).

msemtools

All msemtools is, is a wrapper for metaSEM.

It is not on CRAN, it's currently github-only. Install it like this:

```
remotes::install_github("conig/msemtools")
```

msemtools makes the moderator matrices for you, renames the slopes and formats things to make life easier.

E.g.

```
library(msemtools)
```

```
## A panicked grey bird...
## Red peanuts sliding softly...
## An egg with dark dreams...
##                             [-_-]~
```

```
mod_res <- baseline %>%
  moderate(Discipline)
mod_res
```

```
## Moderation results:
##
## I2(2): 15.7%
## I2(3): 58.4%
## ------------------------------------
##   moderation  k  n R2_2 R2_3 p.value
## ------------------------------------
## 1 Discipline 21 66    0  0.5   0.13
## ------------------------------------
## All models converged.
```

We can see immediately that disciple is not a significant moderator. We can get the slopes like this:

```
mod_res %>%
  summary(hide.insig = FALSE)
```

```
## # A tibble: 6 x 8
##   Moderator k     n     ‘Estimate (95\\~ ‘$SE$‘ ‘$R^2_{(2)}$‘ ‘$R^2_{(3)}$‘
##   <chr>     <chr> <chr> <chr>            <chr>  <chr>         <chr>
## 1 Baseline  21    66    -0.10 [-0.18, -~ 0.04   -             -
## 2 Discipli~ 21    66    -                -      0.00          0.50
## 3 __Physic~ 5     14    -0.02 [-0.17, 0~ 0.08   -             -
## 4 __Life s~ 12    26    -0.14 [-0.23, -~ 0.04   -             -
## 5 __Social~ 5     13    -0.25 [-0.45, -~ 0.10   -             -
## 6 __Multid~ 5     13    -0.01 [-0.14, 0~ 0.06   -             -
## # ... with 1 more variable: ‘$p$‘ <chr>
```

The way it does this is by automatically creating the matrices based on the predictors you give it. It them supplies the predictors to the metaSEM.

E.g.

```
msemtools::character_matrix(dat$Discipline)[1:5,]
```

```
##      Physical sciences Life sciences/biology Social sciences/humanities
## [1,]                 1                     0                          0
## [2,]                 1                     0                          0
## [3,]                 1                     0                          0
## [4,]                 1                     0                          0
## [5,]                 0                     0                          1
##      Multidisciplinary
## [1,]                 0
## [2,]                 0
## [3,]                 0
## [4,]                 0
## [5,]                 0
```

One nice feature of msemtools is it helps you test multiple moderated models all in one step without having to set up all the matrices:

```
res2 <- baseline %>%
  moderate(Discipline, Type, Year, Country)
```

```
res2
```

```
## Moderation results:
##
## I2(2): 15.7%
## I2(3): 58.4%
## ------------------------------------
##   moderation  k  n R2_2 R2_3 p.value
## ------------------------------------
## 1 Discipline 21 66 0.00 0.50    0.13
## 2       Type 21 66 0.07 0.79  0.004*
## 3       Year 21 66 0.00 0.11    0.47
## 4    Country 21 66 0.12 0.66  0.020*
## ------------------------------------
## All models converged.
```

The results should never be different to metaSEM (if they are, you've found a bug - let me know!), your code should just be more efficient.

```
summary(res2, hide.insig = FALSE)
```

```
## # A tibble: 18 x 8
##    Moderator k     n     `Estimate (95\\~ `$SE$` `$R^2_{(2)}$` `$R^2_{(3)}$`
##    <chr>     <chr> <chr> <chr>            <chr>  <chr>         <chr>
##  1 Baseline  21    66    -0.10 [-0.18, -~ 0.04   -             -
##  2 Discipli~ 21    66    -                -      0.00          0.50
##  3 __Physic~ 5     14    -0.02 [-0.17, 0~ 0.08   -             -
##  4 __Life s~ 12    26    -0.14 [-0.23, -~ 0.04   -             -
##  5 __Social~ 5     13    -0.25 [-0.45, -~ 0.10   -             -
##  6 __Multid~ 5     13    -0.01 [-0.14, 0~ 0.06   -             -
##  7 Type      21    66    -                -      0.07          0.79
##  8 __Grant   13    40    -0.01 [-0.08, 0~ 0.04   -             -
##  9 __Fellow~ 11    26    -0.20 [-0.28, -~ 0.04   -             -
## 10 Year      21    66    -                -      0.00          0.11
## 11 __Interc~ -     -     -9.25 [-34.50, ~ 12.89  -             -
## 12 __Year    21    66    0.00 [-0.01, 0.~ 0.01   -             -
## 13 Country   21    66    -                -      0.12          0.66
## 14 __United~ 4     12    0.00 [-0.11, 0.~ 0.06   -             -
## 15 __Canada  1     3     -0.13 [-0.33, 0~ 0.10   -             -
## 16 __Austra~ 5     13    -0.02 [-0.20, 0~ 0.09   -             -
## 17 __United~ 4     10    0.06 [-0.10, 0.~ 0.08   -             -
## 18 __Europe  7     28    -0.22 [-0.32, -~ 0.05   -             -
## # ... with 1 more variable: ‘$p$‘ <chr>
```

Also if you’re willing to play with LaTeX, or the github package ‘papaja’ you can automatically create moderator tables (in PDF or word). Note if you’re not using papaja you’ll need to add this to the YAML header to make this table in PDF documents:

```
header-includes:
  - \usepackage{threeparttable}
  - \usepackage{booktabs}
  - \usepackage{float}
```

Table 1: Moderator analysis results

Moderator	k	n	Estimate (95% CI)	SE	$R^2_{(2)}$	$R^2_{(3)}$	p
Baseline ($I^2_{(2;3)}$: 0.16; 0.58)	21	66	-0.10 [-0.18, -0.02]	0.04	-	-	-
Discipline	21	66	-	-	0.00	0.50	0.13
Type	21	66	-	-	0.07	0.79	.004*
Grant	13	40	-0.01 [-0.08, 0.07]	0.04	-	-	-
Fellowship	11	26	-0.20 [-0.28, -0.12]	0.04	-	-	-
Year	21	66	-	-	0.00	0.11	0.47
Country	21	66	-	-	0.12	0.66	.020*
United States	4	12	0.00 [-0.11, 0.12]	0.06	-	-	-
Canada	1	3	-0.13 [-0.33, 0.07]	0.10	-	-	-
Australia	5	13	-0.02 [-0.20, 0.16]	0.09	-	-	-
United Kingdom	4	10	0.06 [-0.10, 0.21]	0.08	-	-	-
Europe	7	28	-0.22 [-0.32, -0.12]	0.05	-	-	-

Note. p-values are from likelihood ratio tests comparing the baseline model to moderated models

```
res2 %>%
  format_nicely() %>%
  to_apa(caption = "Moderator analysis results",
         note = "p-values are from likelihood ratio tests comparing the baseline model to moderated mode
```