Computer Practicals

Evidence Synthesis Methods Course

ISPM September 2019

Georgia Salanti, Konstantina Chalkou, Virginia Chioccia, Tasnim Hamza, Mike Seo

ISPM University of Bern

Table of Contents

Basic Meta-Analysis

Exercise 1

R packages

We will use the package readxl to import excel data and the meta package to perform meta-analysis

```
library(meta)
library(readxl)
```

Dataset

Dataset diuretic.xls contains the data from the review of diuretics in pregnancy. Open the dataset in R, either by using *Import Dataset* tab or type:

```
# First set the directory to where the data is saved
setwd("C:/Users/kc19o338/Desktop/Practicals/data/”)

DiureticData <- read_excel("diuretic.xls")

DiureticData <- as.data.frame(DiureticData)
```

Use the str(), or summary() command to obtain details of the dataset.

```
str(DiureticData)
summary(DiureticData)
```

These data are from the meta analysis published by Collins et al. (British Medical Journal, 290, 17-23). They examine the prevention of pre-eclampsia with diuretics.

trialid: name of first author of trial

pet: number of treated patients with pre-eclampsia

pec: number of control patients with pre-eclampsia

We will start by calculating log odds ratio and its standard error for each study, for use in meta-analyses. (We will see in question 2 that these calculations can be done automatically). First, derive the number of healthy mothers (those who did not experience pre-eclampsia) in the treatment and control arm of each trial

```
DiureticData$hc<-DiureticData$nc-DiureticData$pec
DiureticData$ht<-DiureticData$nt-DiureticData$pet
DiureticData
```

Now derive the odds ratio, then the log odds ratio and its corresponding standard error for each study.

```
DiureticData$or=(DiureticData$pet/DiureticData$ht)/(DiureticData$pec/Diuretic
Data$hc)
DiureticData$logor=log(DiureticData$or)
DiureticData$selogor=sqrt(1/DiureticData$pet+1/DiureticData$ht+1/DiureticData
$pec+1/DiureticData$hc)
DiureticData
```

In the above list, which trials have the smallest standard errors and which have the biggest standard errors?

Meta-analysis

We will now use the **metagen** command to perform a fixed and then a random-effects meta-analysis and the **forest** command to obtain the forest plots, using inverse-variance weighting. The basic output is produced by typing:

```
metagen(DiureticData$logor, DiureticData$selogor, comb.random = FALSE)
```

To display the output on the odds ratio scale, type:

```
fixed.effect<-metagen(DiureticData$logor, DiureticData$selogor, sm= "OR", com
b.random = FALSE)
print(fixed.effect)
forest(fixed.effect)
```

To derive a random-effects analysis:

```
random.effects<-metagen(DiureticData$logor,DiureticData$selogor, sm="OR", com
b.fixed = FALSE)
print(random.effects)
forest(random.effects)
```

Check that these results agree with those in the handout. What happens to the squares in the forest plot when you conduct a random-effects analysis?

If you don't specify the comb.fixed or comb.random parameters, default is to display both the fixed and random-effects estimates.

I

```
both.models<-metagen(DiureticData$logor,DiureticData$selogor, sm="OR")
print(both.models)
forest(both.models)
```

There are a large number of options to derive a forest plot, which you can examine by typing:

```
?forest
```

Experiment with the use graphics options of the **forest** command. The following options are likely to be useful:

xlab - label the horizontal axis (Odds Ratio)

prediction=TRUE - add prediction interval for random-effects model

layout = "JAMA" - a specific layout of the forest plot

col.diamond = "green" - The colour of diamonds representing the results for fixed effect and random effects models.

col.diamond.lines = "red" - The colour of the outer line of diamonds representing the results for fixed effect and random effects models.

col.square = "yellow" - The colour for squares reflecting study's weight in the meta-analysis

col.square.lines = "blue" - The colour for the outer lines of squares reflecting study's weight in the meta-analysis

Exercise 2

The **metabin** command will use the raw (2X2) data from each study to calculate a range of treatment effect estimates and then derive meta-analyses using either Mantel-Haenszel or inverse-variance weights, for odds ratios, risk ratios and risk differences, also the Peto method for odds ratios, and generalized linear mixed model are available for pooling. The default is to use the Mantel-Haenszel method. The user supplies the names of variables containing the number of: i) events in treatment group, ii) observations in treatment group, iii) events in control group and iv) observations in control group. This means that you do not need to start by deriving the treatment effect and its standard error in each study. For example (random effects):

```
?metabin
random.effects.or = metabin(pet, nt, pec, nc, comb.fixed = FALSE,
                    data = DiureticData, sm = "OR")
summary(random.effects.or)
forest(random.effects.or)
```

As well as the graph options used for question 1, you can use the options available for the forest plot.

Use this command to derive the same analysis as in question 1, by using the comb.random=FALSE for fixed effect model:

```
fixed.effect.or = metabin(pet, nt, pec, nc, comb.random=FALSE,
                    data = DiureticData, sm = "OR")
summary(fixed.effect.or)
forest(fixed.effect.or)
```

Use this command to derive the same analysis as in question 1 for both fixed effect model and random effects model:

```
both.models.or = metabin(pet, nt, pec, nc,
                    data = DiureticData, sm = "OR")
summary(both.models.or)
forest(both.models.or)
```

Compare the results using Mantel-Haenszel and inverse variance weights, for odds ratios and risk ratios, and using the Peto method for odds ratios.

You can choose different method to be used for pooling the studies by "method" parameter and different summary measure by "sm" parameter. See helpfile for metabin for more information.

The **metabin** command stores the estimated log of treatment effect (i.e. log of odds ratio) and the standard error of the log effect with following variable names

```
random.effects.or$TE.random

random.effects.or$seTE.random
```

These estimates should match the result we get from metagen (note in metagen inverse variance method was used)

```
random.effects$TE.random
random.effects$seTE.random
```

Some R meta-analysis commands require the user to provide the log odds ratio and its standard error: the metabin command is useful for deriving these variables.

Exercise 3

Datasets streptok.xls and magnes.xls give the effects (from all known trials) of streptokinase and magnesium respectively on the prevention of mortality after acute myocardial infarction.

For the streptokinase data, perform meta-analyses on all studies: a) performed up to and including 1977

b) excluding the ISIS-2 trial, trial's number = 22 - (hint: ?metabin, use exclude=…)

c) including all trials

The basic meta-analysis is conducted by deriving:

pooledOR = metabin(cases1, pop1, cases0, pop0, data = StreptokData, sm = "OR")

To restrict to trials performed up to and including 1977:

pooledOR = metabin(cases1, pop1, cases0, pop0, subset= year<=1977, data = StreptokData, sm = "OR")

and so on.

Is there any evidence of heterogeneity between studies? Examine how the evidence for an effect of streptokinase accumulated using the **metacum** (cumulative meta-analysis) command. For example:

```
CumulativeMA<-metacum(pooledOR)
CumulativeMA
forest(CumulativeMA)
```

Routine treatment with streptokinase after myocardial infarction was not recommended in the majority of textbooks and reviews until the late 1980s.

Exercise 4

(Optional - solutions not provided)

For the magnesium data, perform meta-analyses on all studies: a) performed up to and including 1991

b) excluding the ISIS-4 trial

c) including all trials

Subgroup analysis, meta-regression and bias

R packages

We will use the package **readxl** to import excel data and the package **meta** to run meta-analyses.

```
library(readxl)
library(meta)
```

Results (treatment estimates and confidence intervals) should be rounded to two digits (default is four digits).

```
settings.meta(digits = 2)
```

Exercise 2

Dataset "bcgtrial"

Dataset bcgtrial.xls contains data from the meta-analysis of the efficacy of BCG vaccine against tuberculosis performed by Colditz et al. (JAMA 1994; 271: 698-702). Open the dataset bcgtrial.xls by using the *Import Dataset* tab or type:

```
BCG = read_excel("bcgtrial.xls")
BCG <- as.data.frame(BCG)
```

Check the data structure:

```
str(BCG)
```

Use the log risk ratio and its standard error to calculate risk ratios and 95% confidence intervals for each study:

```
BCG$rr <-       (BCG$cases1/BCG$pop1) / (BCG$cases0/BCG$pop0)

BCG$logrr <-    log(BCG$rr)

BCG$selogrr <-  sqrt(1/BCG$cases1 - 1/BCG$pop1 + 1/BCG$cases0 - 1/BCG$pop0)

BCG$loglci <-   BCG$logrr-1.96*BCG$selogrr

BCG$loguci <-   BCG$logrr+1.96*BCG$selogrr

BCG$lci <-      exp(BCG$loglci)

BCG$uci <-      exp(BCG$loguci)
```

Print the results for each study:

```
BCG[,c("trial", "trialnam", "startyr", "latitude", "rr", "lci", "uci")]
```

Do you think that there is evidence of heterogeneity? Explain your answer.

Meta-analysis of the effect of BCG on tubercolosis

Perform a meta-analysis of the effect of BCG on tuberculosis.

```
ma1 = metabin(cases1, pop1, cases0, pop0, data = BCG,
                    studlab = trialnam, sm = "RR")
summary(ma1)
```

Compare the fixed-effects and random-effects estimates.

Fixed effects OR (95% CI):

Random effects OR (95% CI):

Compare the weights used in the two types of meta-analysis (run `print(ma1)` and check *%W(fixed)* and *%W(random)* in the output). Comment on the difference between the fixed- and random-effects estimates.

Create forest plots

To obtain a forest plot we will use the function `forest`. You can use the `leftcols` and `rightcols` arguments to add columns displaying study characteristics to the graph.

```
forest(ma1, leftcols = c("trialnam","authors"), leftlabs=c("Trial name", "Authors"),
       just.addcols = "left", rightcols = c("effect", "ci", "w.fixed", "w.random", "startyr", "alloc"),
       rightlabs = c("Year", "Allocation"), digits.addcols.right = 0)
```

Subgroup analysis

Trials 1, 8, 10 and 11 were performed in the USA. Use the `byvar` argument of the `metabin` function to compare results in the USA from results elsewhere (you first need to create a variable to distinguish them).

```
BCG$USA <- ifelse(BCG$trial %in% c(1,8,10,11), "USA", "elsewhere")

table(BCG$trialnam, BCG$USA)

ma2 = metabin(cases1, pop1, cases0, pop0, data = BCG,
                  studlab = trialnam, sm = "RR", byvar = USA)

forest(ma2, bylab = "")
```

In the light of the various analyses you have performed, what vaccination policy would you recommend to the US Centers for Disease Control?

Sort the data by latitude and produce a forest plot of the effect in each study.

```
forest(ma1, leftcols = c("trialnam","authors", "latitude"), just.addcols = "left", sortvar = latitude)
```

What is the impression?

Do the same for year of start of study. What is the impression?

```
forest(ma1, leftcols = c("trialnam","authors", "startyr"), just.addcols = "left", sortvar = startyr)
```

Use the byvar argument of the metabin function to compare the results of meta-analyses of the trials that used random allocation (alloc=0) and those that used non-random allocation methods (alloc=1). Does the method of allocation explain much of the heterogeneity?

```
table(BCG$trialnam, BCG$alloc)

ma3 = metabin(cases1, pop1, cases0, pop0, data = BCG,
                    studlab = trialnam, sm = "RR", byvar = alloc)
summary(ma3)

forest(ma3, bylab = "")
```

Meta-regression

Meta-regression can be used to assess whether there is formal statistical evidence that the effect varies with a covariate or between subgroups of studies. The metareg function requires an object of class meta as argument. Conduct a meta-regression for R object *ma3* and compare the results with the subgroup analysis. Look at the results of *Test of Moderators* in the meta-regression and the *Test for subgroup differences* in the random effects subgroup meta-analysis.

As R object *ma3* was generated using argument byvar, we do not have to specify a covariate in the meta-regression command.

```
mr1 = metareg(ma3)
mr1
```

We get the same result explicitly stating the covariate 'alloc' using *ma1*.

```
mr1 = metareg(ma1, alloc)
```

In the output, the coefficient ('estimate') for alloc is the estimate of difference in the log risk ratio between the trials with Random Allocation (alloc=1) and those with non-random allocation methods (alloc=0) (the coefficient labeled 'intrcpt' is the estimate of the log risk ratio for the alloc=0 trials and is of less interest here). The output also gives the standard error of each coefficient, a test of the null hypothesis that it is zero, and a confidence interval. Is there statistical evidence that the results vary according to the method of allocation?

Note, coefficients can be extracted using R function coef and as results for the meta-regression are reported on the log-scale we can use exp for back-transformation.

Coefficients of the meta-regression:

```
coef(mr1)
```

Odds ratio for the non-random allocation subgroup:

```
logOR.nra = coef(mr1)[1]

round(exp(logOR.nra), 2)
```

Odds ratio for the random allocation subgroup:

```
logOR.ra = sum(coef(mr1))

round(exp(logOR.ra), 2)
```

Use the `metareg` function to examine the association of the log risk ratio with latitude and with year the trial started. With continuous covariates, such as these, the output can be interpreted as the difference in the log risk ratio associated with a one-unit increase in the covariate.

The `bubble` function can be used with a continuous covariate to plot the fitted regression line together with circles representing the estimates from each study. The size of each circle indicates the precision of each estimate (the inverse of its within-study variance). For example, to look at the association with latitude:

```
mr2 = metareg(ma1, latitude)
mr2

bubble(mr2, col.line = "blue")
```

And for the association with the year the trial started:

```
mr3 = metareg(ma1, startyr)
mr3
```

Comment the extent to which different study characteristics explain the between-trial heterogeneity.

Exercise 2

Dataset “magnes”

Dataset magnes.xls gives the effects (from all known trials) of magnesium on the prevention of mortality after acute myocardial infarction.

Open the dataset:

```
Magnesium = read_excel("magnes.xls")
```

Perform meta-analyses

a) on studies performed up to and including 1991

Summarize the study-specific odds ratios for mortality from studies using metabin. We will create an object of class meta called "OR1". The argument 'subset' specifies we only include studies performed up to 1991.

```
OR1 = metabin( deaths1, pop1, deaths0, pop0, data = Magnesium,
                    studlab = trialnam, sm = "OR", subset = year<=1991)
summary(OR1)
```

Fixed effects OR (95% CI):

Random effects OR (95% CI):

b) excluding the ISIS-4 trial

```
OR2 <- update(OR1, subset = trialnam!="ISIS-4")
summary(OR2)
```

Fixed effects OR (95% CI):

Random effects OR (95% CI):

c) including all trials

```
ORall = metabin(deaths1, pop1, deaths0, pop0, data = Magnesium,
                      studlab = trialnam, sm = "OR")
summary(ORall)
```

Fixed effects OR (95% CI):

Random effects OR (95% CI):

Funnel plots

The function `funnel` produces funnel plots. Examine the funnel plot of the magnesium data, with and without inclusion of the ISIS-4 trial. For example:

```
funnel(ORall)

funnel(OR2)
```

Function `metabias` produces a test for funnel plot asymmetry. The user needs to specify which test shoulds be conducted and reported in the `method.bias` argument. For example, for the Harbord's score-based test:

```
metabias(ORall, method.bias = "score")
```

Using this command, use Harbord's test to examine whether there is evidence of funnel plot asymmetry in the magnesium dataset, both including and excluding the ISIS-4 mega-trial. Comment.

```
metabias(OR2, method.bias = "score")
```

Dose-response meta-analysis

Exercise 1

Background

Second-generation antidepressants are the first-line options for the pharmacological treatment for depression. The therapeutic dose is at least 20 mg of fluoxetine or equivalent on a daily basis. However, there is still an ongoing debate about whether and how much antidepressants should be titrated up in order to maximize their effect as the side effects increase with increased dosage. The aim here is to find an optimal dose range where both the efficacy and acceptability are optimised.

R packages

We will use the packages **meta**, **readxl**, **dosresmeta** and **rms** to run and plot dose-response analyses.

If you haven't already installed **dosresmeta** and **rms**, please do so by typing

```
install.packages("dosresmeta")
install.packages("rms")
```

Next, we make the R packages available.

```
library(dosresmeta)
library(rms)
library(meta)
library(readxl)
```

Dataset

We will import the Excel dataset AntidepressantsDOSE.xls.

```
AntidepressantsDOSE = read_excel("AntidepressantsDOSE.xls", na = "NA")
AntidepressantsDOSE = as.data.frame(AntidepressantsDOSE)
```

This has RCTs that compare the outcomes with antidepressants at various dosages. How many studies are available?

```
length(unique(AntidepressantsDOSE$Study_No))
```

How many different doses as observed? Which is the most frequently studies dose?

```
table(AntidepressantsDOSE$dose)
length(table(AntidepressantsDOSE$dose))
```

Take some time to browse the column logRR in the dataset. How are these numbers calculated?

Linear association

We will use the dosresmeta function to investigate the association between the dose and the response to antidepressants. Use the help to understand the arguments in the function and its output. Although the shape is expected to be non-linear, we will start with a linear model.

```
doseresRR = dosresmeta(logRR ~ dose, Study_No, data = AntidepressantsDOSE,
                       cases = Responders, n = No_randomised,
                       type = type, se = selogRR, method = "reml")
summary(doseresRR)
```

Now, let us plot the results

```
with(predict(doseresRR, exp = TRUE, order = TRUE), {
     plot(dose, pred, log = "y", type = "l",
          xlim = c(0, 160), ylim = c(0.95, 2.5),
          xlab = "Dose", ylab = "RR response", main = c("Linear"))
     lines(dose, ci.lb, lty = 2)
     lines(dose, ci.ub, lty = 2)
     rug(dose, quiet = TRUE)
     })
with(AntidepressantsDOSE, points(dose[logRR != 0], exp(logRR[logRR != 0])))
with(AntidepressantsDOSE, rug(dose, quiet = TRUE))
```

What do the points represent in the graph?

Splines

The anticipated dose-response shape is expected to increase up a certain dose and then flaten out. We can use the rcs function for such a shape and will need to define some knots (at least 3).

```
knots = c(10, 20, 50)
doseresRR = dosresmeta(logRR ~ rcs(dose, knots), Study_No,
                       data = AntidepressantsDOSE,
                       cases = Responders, n = No_randomised,
```

```
                        type = type, se = selogRR, proc = "1stage")
summary(doseresRR)
```

Now, let us plot the results.

```
newdata = data.frame(dose = seq(0, 160))
xref = min(AntidepressantsDOSE$dose)
with(predict(doseresRR, newdata, xref, exp = TRUE), {
     plot(get("rcs(dose, knots)dose"), pred, log = "y", type = "l",
          xlim = c(0, 160), ylim = c(0.95, 2.5),
          xlab = "Dose", ylab = "RR", main = c("Splines", text))
     matlines(get("rcs(dose, knots)dose"), cbind(ci.ub, ci.lb),
              col = 1, lty = "dashed")
     })
with(AntidepressantsDOSE, points(dose[logRR != 0], exp(logRR[logRR != 0])))
with(AntidepressantsDOSE, rug(dose, quiet = TRUE))
```

Now, change the knots; put more knots at different places and plot the results to see if the shape changes a lot. Note the AIC and select the set of knots that provides the best AIC.