Meta-analysis of test accuracy studies in R

A summary of user-written programs and step-by-step guide to using glmer

Christopher Partlett and Yemisi Takwoingi

Version 1.0
August 2016

Please cite this version as: Partlett C, Takwoingi Y. Meta-analysis of test accuracy studies in R: a summary of user-written programs and step-by-step guide to using glmer. Version 1.0. August 2016. Available from: http://methods.cochrane.org/sdt/.

Contents

1 Introduction

In this tutorial, we summarise and illustrate some of the user-written packages for conducting meta-analysis of diagnostic test accuracy (DTA) studies in R. Furthermore, we provide instructions for carrying out the bivariate binomial method by fitting a generalized linear mixed model (GLMM) using the `glmer` function in the R package **lme4**. Table 1 summarises the functionality of the available packages. In particular, it shows the models (bivariate model, HSROC model or both) the packages can fit, and whether or not they allow for meta-regression through the inclusion of covariates in the model.

Table 1: Summary of user written packages for DTA meta-analysis in R

Package (reference)[a]	Model	Meta-regression	Compatible with RevMan	Other software requirements
mada[1]	Bivariate normal[b]	Yes	Yes	None
HSROC[2]	Bivariate and HSROC	No	No	None
bamdit[3]	Bivariate Bayesian	No	No	JAGS
CopulaREMADA[4]	Trivariate[c]	No	No	None
mvmeta[5]	Bivariate normal	Yes	Yes	None
lme4 (glmer function in lme4 for fitting GLMMs in R)[6]	Bivariate binomial	Yes	Yes	None

JAGS = Just Another Gibbs Sampler.

[a]The packages implement the bivariate model with a normal or binomial within-study likelihood, or the HSROC model using a binomial likelihood.

[b]If there are no covariates in the model, **mada** also generates parameters for the HSROC curve by exploiting the relationship between bivariate and HSROC models.[7]

[c]The trivariate model jointly synthesises sensitivity, specificity and prevalence of the target condition.

In section 2 we introduce the example data used in this tutorial and describe how to load the data into R. In section 3 and section 4 we discuss the functionality and plotting capabilities of the **HSROC** and **bamdit** packages, respectively. In section 5 we briefly outline key features of other user written packages listed in Table 1. Section 6 provides a detailed tutorial for carrying out the bivariate binomial method by directly fitting a GLMM using the function `glmer`.

2 Data

2.1 Introduction

The example dataset used in this tutorial, schuetz.csv, is based on a published diagnostic test accuracy review by Schuetz et al.[8] Schuetz and colleagues evaluated the diagnostic performance of multislice computed tomography (CT) and magnetic resonance imaging (MRI) for the diagnosis of coronary artery disease (CAD). Prospective studies that evaluated either CT or MRI (or both); used conventional coronary angiography (CAG) as the reference standard; and used the same threshold for clinically significant coronary artery stenosis (a diameter reduction of 50% or greater) were included in the review. A total of 103 studies provided a 2x2 table for one or both tests and were included in the meta-analysis: 84 studies evaluated only CT, 14 evaluated only MRI, and 5 studies evaluated both CT and MRI.

2.2 Set working directory and read data into R

Set your working directory to the appropriate drive where you saved the file schuetz.csv. This can be done via the File menu (see Figure 1). Select `File -> Change dir…` and then browse to find the correct folder. Alternatively, one can use the command

```
setwd("C:/'insert address here'")
```

It is important to ensure that folders are separated by a / and not \. Once the directory has been set, to read the comma delimited (Excel.csv) file containing the data use

```
(X = read.csv("schuetz.csv"))
```

This assigns the data in ("schuetz.csv") to the object X (referred to as a data frame) in R's memory, and simultaneously displays the data in the R console.

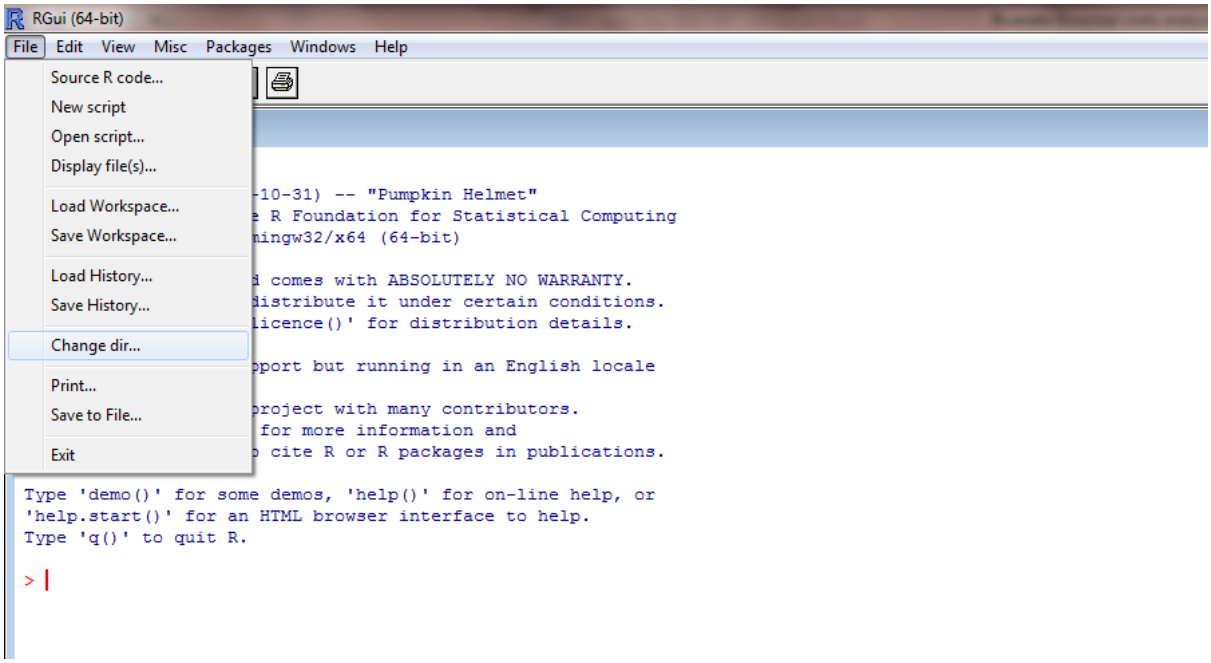

Figure 1: Setting the working directory using the File menu

3 HSROC package

First, if needed, install the **HSROC** package (you need to select an appropriate CRAN mirror, for example, this will typically be the geographically closest)

```
install.packages("HSROC")
```

Next, load the **HSROC** package to the current R workspace

```
library(HSROC)
```

HSROC is very particular about the form of the dataframe containing the test accuracy data; it cannot handle additional columns (e.g. those indicating the study ID).

```
X.CT = X[X$Test=="CT",]
X.MRI = X[X$Test=="MRI",]
X.CT = X.CT[,3:6]
X.MRI = X.MRI[,3:6]
```

Let's fit the HSROC model for the CT data. As we will see, it is convenient to include the data in a folder specific to the test. Thus, first create a new folder named 'CT' and set the working directory to this folder.

```
setwd("C:/'insert address here'/HSROC/CT")

hsroc.CT = HSROC(X.CT , iter.num=10000 )

summ.CT = HSROCSummary(data = X.CT, burn_in=5000, Thin=2, print_plot=TRUE )

summ.CT[[1]]
```

The files generated by the Gibbs sampler process are all stored in the folder 'CT'. Running the functions again for MRI without creating a new folder will overwrite this information. Thus, again create a new folder named 'MRI' and reassign the working directory.

```
setwd("C:/'insert address here'/HSROC/MRI")

hsroc.MRI = HSROC(X.MRI, iter.num=10000 )

summ.MRI =HSROCSummary(data = X.MRI, burn_in=5000, Thin=2, print_plot=TRUE)

summ.MRI[[1]]
```

The output of `summ.MRI` and `summ.CT` is summarised in Table 2. The function `HSROCSummary` fits the HSROC model and returns the model parameters as well as density and trace plots for the HSROC parameters. The SROC curves are presented in Figure 1 below.

It is important to note that HSROC fits the HSROC model using the probit link function and not the logit link function used in the HSROC model developed by Rutter and Gatsonis.[9] As a result, the model parameters returned by HSROC are not compatible with Review Manager (RevMan) and will lead to erroneous SROC curves.

Table 2: Posterior median and upper and lower 95% highest posterior density credible interval for the model parameters

MRI				CT			
Parameter	**Median**	**HPD.low**	**HPD.high**	**Parameter**	**Median**	**HPD.low**	**HPD.high**
THETA	-0.509	-0.779	-0.199	THETA	-0.375	-0.562	-0.195
LAMBDA	1.852	1.465	2.234	LAMBDA	2.934	2.797	3.000
beta	0.404	-0.107	0.750	beta	-0.023	-0.280	0.224
sigma.alpha	0.462	0.092	0.867	sigma.alpha	0.828	0.657	1.027
sigma.theta	0.333	0.183	0.519	sigma.theta	0.307	0.229	0.392
S Overall	0.881	0.832	0.922	S Overall	0.968	0.959	0.977
C Overall	0.692	0.588	0.789	C Overakk	0.858	0.834	0.880
S1_new	0.918	0.717	0.950	S1_new	0.964	0.804	1.000
C1_new	0.662	0.348	0.000	C1_new	0.855	0.583	0.000

HPD = Highest posterior density

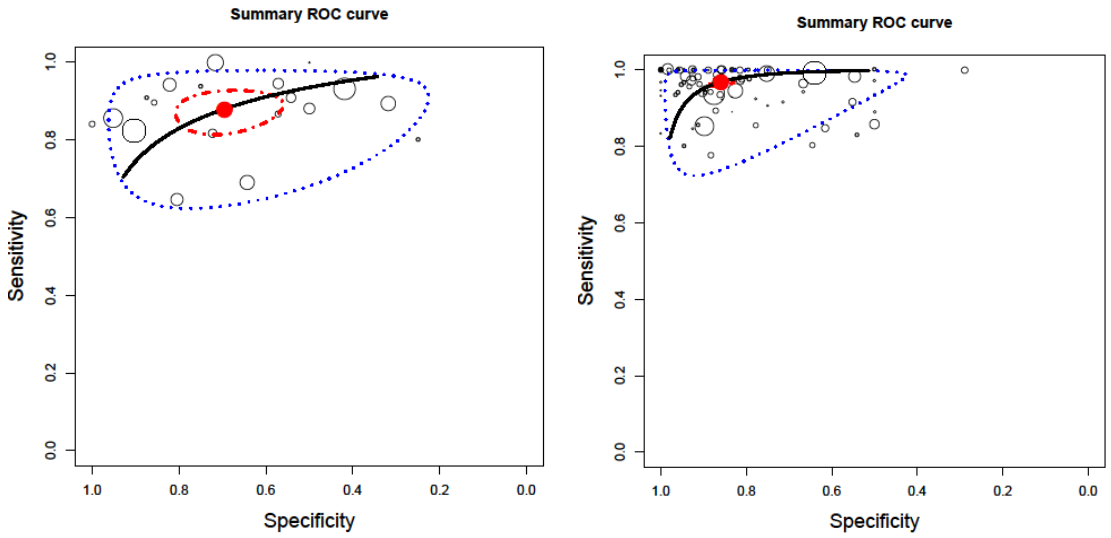

Figure 2: Summary ROC curves for MRI (left) and CT (right) plotted using the HSROC package

4 Bamdit package

First, if needed, install the **bamdit** package

```
install.packages("bamdit")
```

Next, load the **bamdit** package to the current R workspace

```
library(bamdit)
```

Similar to the **HSROC** package, **bamdit** requires the data in a very specific format—a data frame with 4 columns containing the number of true positives, number of patients with disease, number of false positives, and number of patients without disease.

```
Z = X[c("tp","fp")]

Z$n1 = X$tp+X$fn
Z$n0 = X$fp+X$tn

Z = Z[c("tp", "n1" , "fp", "n0")]
```

Again, we generate a separate data frame for each test.

```
Z.CT  = Z[X$Test=="CT",]
Z.MRI = Z[X$Test=="MRI",]
```

The function `metadiag` fits the bivariate random effects model using JAGS (Just Another Gibbs Sampler) to perform MCMC (Markov Chain Monte Carlo) sampling.

```
ma.MRI = metadiag(Z.MRI, re="normal", link="logit" )
ma.CT = metadiag(Z.CT, re="normal", link="logit" )
```

The number of true positives and false positives are modelled with two conditional binomial distributions. The random effects (`re`) can be modelled using bivariate normal (`"normal"`) or scale mixtures (`"sm"`). The function also supports logit, complementary log-log, and probit link functions.

Bayesian SROC curves can be plotted using `bsroc`.

```
par(mfrow=c(2,1))

bsroc(ma.MRI, data = Z.MRI )
bsroc(ma.CT, data = Z.CT )
```

The SROC curves for this data are presented in Figure 2. Using `plotcont`, it is possible to plot the observed data in ROC space, along with the posterior predictive contours.

```
plotcont(ma.MRI, data = Z.MRI , parametric.smooth = TRUE )
plotcont(ma.CT, data = Z.CT , parametric.smooth = TRUE )
```

The function `plotdata` plots the true positive rates and false positive rates of each study included in the meta-analysis. These can be separated according to test type.

```
testtype = as.numeric(X$Test)
plotdata(Z, group = testtype )
```

Further, the predictive posterior surfaces of two fitted models can be compared using

```
plotcompare( ma.CT, ma.MRI, Z , level = 0.95, group = X$Test,
             group.colors = c("blue", "red") , m1.name ="CT" , m2.name ="MRI" )
```

This is demonstrated in Figure 3, which compares the predictive contours of the two models. Finally, posterior densities for sensitivity and specificity can be plotted using `plotsesp`.

```
plotsesp(ma.CT)
plotsesp(ma.MRI)
```

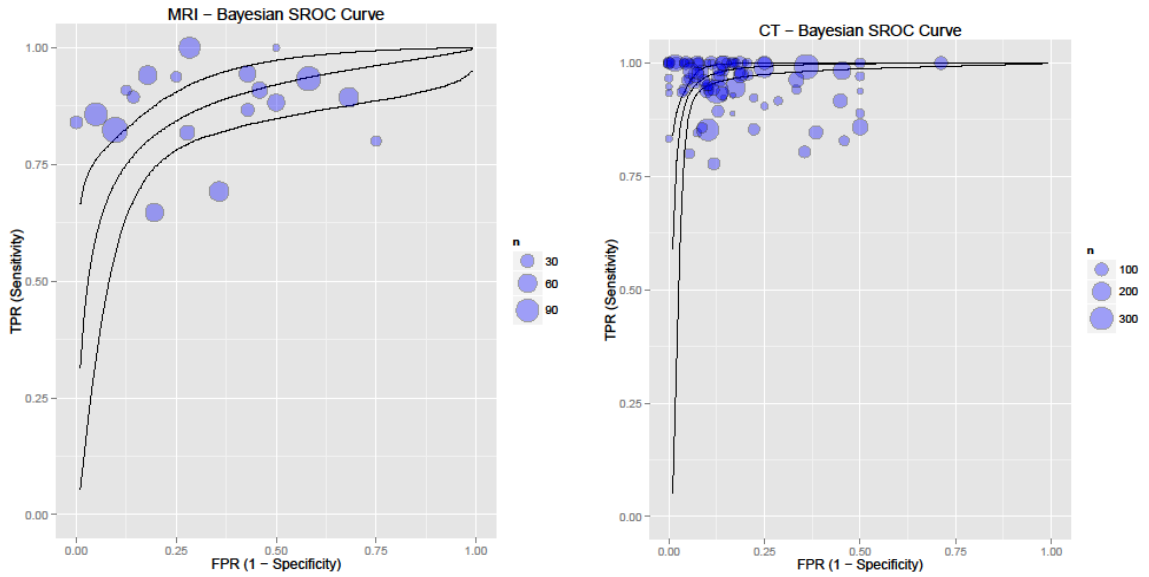

Figure 3: Summary ROC curves (with confidence bounds) plotted using bamdit

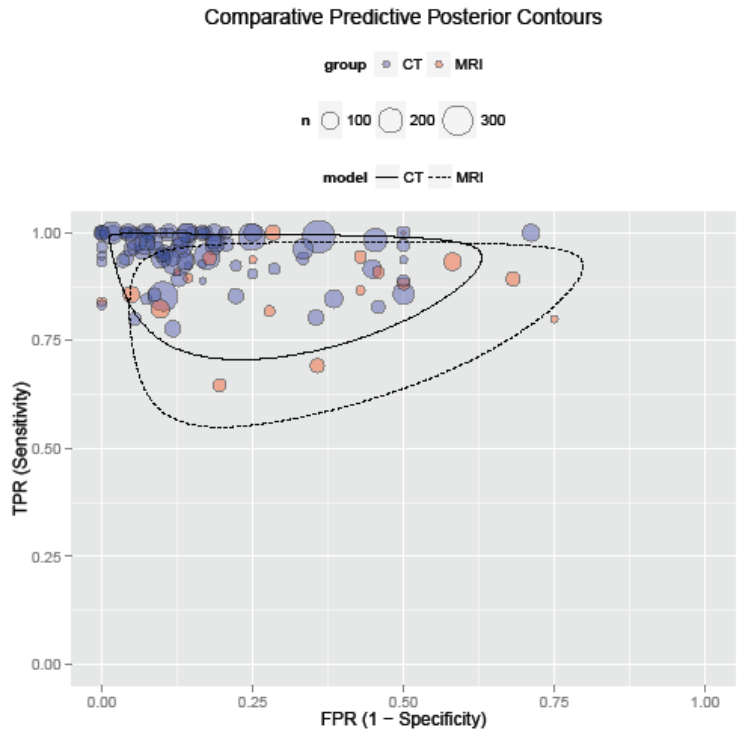

Figure 4: Comparison of predictive posterior contours using bamdit

5 Other user written packages

The **mada** package implements the bivariate normal approach of Reitsma et al.[10] The package can also fit bivariate meta-regression models. However, **mada** **cannot** fit the 'exact' bivariate binomial model of Chu and Cole.[11] If there are no covariates in the model, **mada** also generates parameters for the HSROC curve by exploiting the relationship between bivariate and HSROC models.[7]

The **mvmeta** package implements multivariate meta-analysis without the requirement of the 2x2 data. By simply specifying the logit-sensitivity and specificity (and the corresponding covariance matrix) a bivariate random effects model can be fitted assuming normal within-study variability.

The **CopulaREMADA** package fits copula mixed effects models for meta-analysis of test accuracy studies. In particular, it can be used to fit trivariate models which incorporate the prevalence of the target condition into the model.[4]

6 Direct approach using glmer

6.1 Introduction

Here we describe the approach for fitting the bivariate model using the `glmer` function in the **lme4** package. An **.R** file, *"Bivariate binomial meta-analysis of test accuracy studies in R.R",* accompanies this tutorial. You can either run the commands from the file or you can create your own as you step through the tutorial.

6.2 Meta-analysis using glmer

First, install the **lme4** package if required.

```
install.packages("lme4")
```

Load the package **lme4**

```
library(lme4)
```

In order to specify the GLMM, first, we need to set up the data. In particular, we add 5 new columns to the frame X.

- n1 is the number with the disease
- n0 is the number without the disease
- true1 is the number of true positives
- true0 is the number of true negatives
- study is the unique identifier for each study

```
X$n1 <- X$tp+X$fn
X$n0 <- X$fp+X$tn
X$true1 <- X$tp
X$true0 <- X$tn

X$study <- 1:108
```

Next, reshape the data from wide to long format

```
Y = reshape(X, direction = "long", varying = list( c("n1" , "n0") ,
c( "true1","true0" ) ) , timevar = "sens" , times = c(1,0) ,
v.names = c("n","true") )
```

```
Y = Y[order(Y$id),]
```

The first command assigns the long format data to the object Y. An indicator variable `sens` is defined to identify the sensitivity results. The data is then sorted by study id to cluster the two records per study together. Next add an additional indicator variable to identify specificity results.

```
Y$spec<- 1-Y$sens
```

Generate a separate data frame for each test type

```
Y.CT  =  Y[Y$Test=="CT",]
Y.MRI =  Y[Y$Test=="MRI",]
```

It is now possible to perform a meta-analysis for CT.

```
(MA_Y.CT = glmer( formula = cbind(  true , n - true ) ~ 0 + sens + spec +
(0+sens + spec|study), data = Y.CT , family = binomial  ,  nAGQ = 1 ,
verbose = 2  ))
```

More detail can be obtained by using the summary command

```
(ma_Y.CT = summary(MA_Y.CT))
```

Now, let's examine the model specification and output in more detail.

- The variable `true` specifies the response while `sens` and `spec` are dummy variables. The fixed effect for logit sensitivity and logit specificity are the coefficients of `sens` and `spec` and the constant term is supressed by adding `(0 + ...)` to the model formula.

- Adding `(0 + sens + spec | study)` to the model includes study-level random effects.

- nAGQ defines the number of points per axis for evaluating the adaptive Gauss-Hermite approximation to the log-likelihood. 1 corresponds to the Laplace approximation. A value of zero uses a faster but less exact form of parameter estimation for GLMMs by optimizing the random effects and the fixed effect coefficients in the penalized iteratively reweighted least squares step.

- `family = binomial` specifies the data are in binomial form.

- `verbose` controls the level of reporting on the optimisation process. `verbose = 0` corresponds to no reporting, while `verbose = 1` reduces the amount of output.

- Specified in the form above, the between study covariance matrix is unstructured. In order to assume no correlation between sensitivity and specificity across studies, one must specify the random effects separately as follows:

```
glmer( formula = cbind(  true , n - true ) ~ 0 + sens + spec + (0+sens|study) +
(0+spec|study), data = Y.CT , family = binomial  ,  nAGQ = 1 , verbose = 2  )
```

The output generated by using the summary function is given in Figure 5 below. All of the important parameters for extraction to RevMan are highlighted.

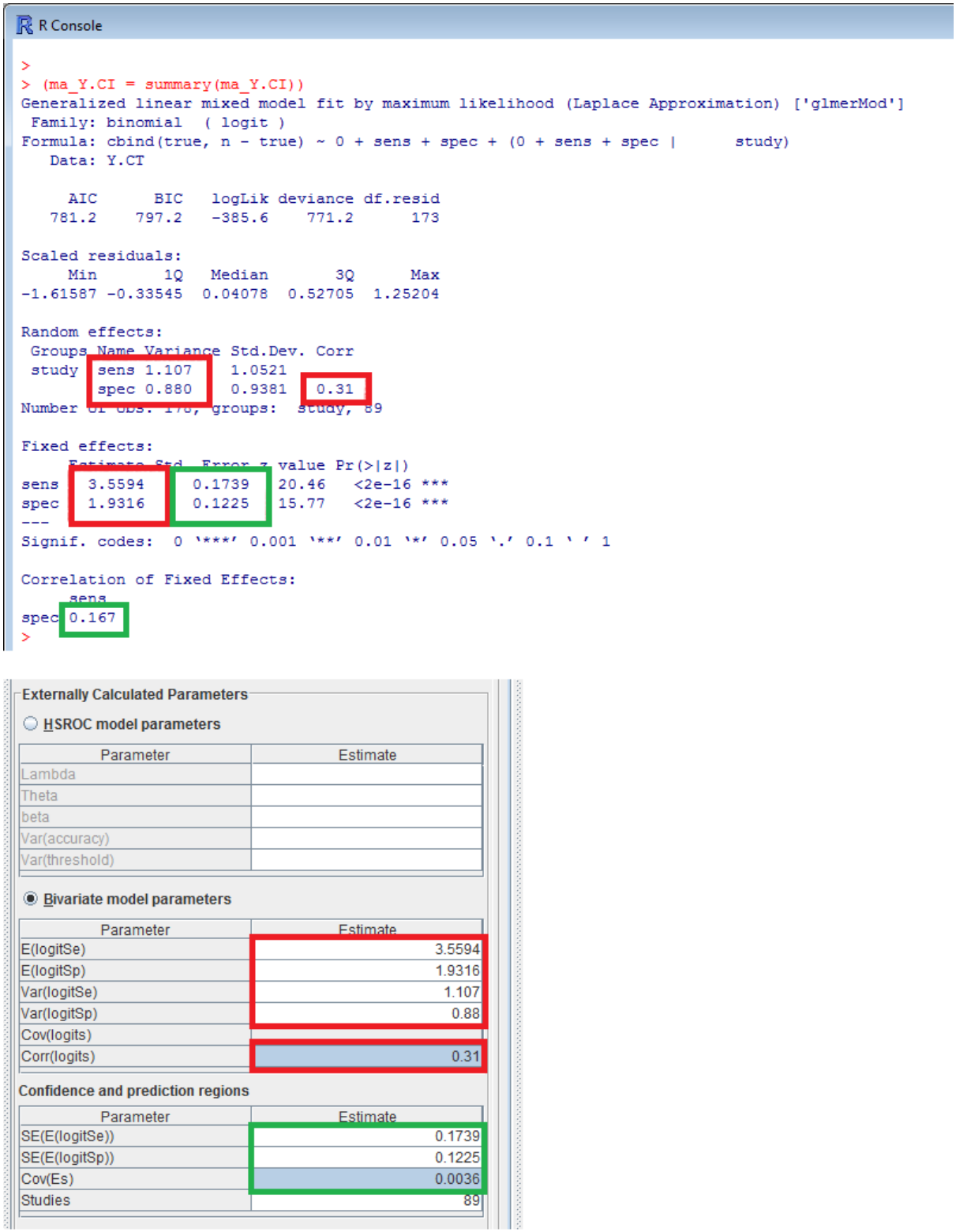

Figure 5: Screenshot of glmer summary output highlighting the values required for input into RevMan

Now, obtaining the sensitivity and specifity on the raw scale (along with their respective confidence intervals) isn't totally straightforward, but it is possible. If you are content to do it manually and also do not require computation of additional summary measures such as diagnostic odds ratios (DORs) and likelihood ratios from the model parameters, then the rest of this section can be skipped.

For the full list of outputs from glmer use `labels`.

```
labels( ma_Y.CT )
```

Therefore, to extract the model coefficients use

```
ma_Y.CT$coeff

(lsens.CT = ma_Y.CT$coeff[1,1])
(lspec.CT = ma_Y.CT$coeff[2,1])

(se.lsens.CT = ma_Y.CT$coeff[1,2])
(se.lspec.CT = ma_Y.CT$coeff[2,2])
```

Then we can manually create 95% confidence intervals for logit sensitivity and logit specificity.

```
(Sens.CT = c(lsens.CT, lsens.CT-qnorm(0.975)*se.lsens.CT,
lsens.CT+qnorm(0.975)*se.lsens.CT ))
(Spec.CT = c(lspec.CT, lspec.CT-qnorm(0.975)*se.lspec.CT,
lspec.CT+qnorm(0.975)*se.lspec.CT ))
```

Alternatively, generate the following dataframe.

```
(logitCT = data.frame( estimate = c(lsens.CT , lspec.CT) ,
       lci =  c(lsens.CT-qnorm(0.975)*se.lsens.CT , lspec.CT-
qnorm(0.975)*se.lspec.CT) ,
       uci = c(lsens.CT+qnorm(0.975)*se.lsens.CT ,
lspec.CT+qnorm(0.975)*se.lspec.CT) ,
             row.names = c("lSens", "lSpec") ) )
```

The sensitivity and specificty estimates can be transformed back to the raw scale using the built in function `plogis`.

```
plogis( Sens.CT )
plogis( Spec.CT )
```

Similarly, we can calculate the diagnostic odds ratio (DOR) and likelihood ratios.

```
(DOR = exp(lsens.CT+lspec.CT ))
(LRp = plogis(lsens.CT)/(1-plogis(lspec.CT)))
(LRm = ((1-plogis(lsens.CT))/plogis(lspec.CT)))
```

Confidence intervals can be obtained using the delta method. This can be implemented in R using the function `deltamethod` in the package **msm**.

```
install.packages("msm")

library(msm)

se.DOR = deltamethod (~ exp(x1+x2) , mean = c(lsens.CT,lspec.CT) , cov =
ma_Y.CT$vcov )

se.LRp = deltamethod (~ (exp(x1)/(1+exp(x1)))/(1-(exp(x2)/(1+exp(x2)))) ,
mean = c(lsens.CT,lspec.CT) , cov = ma_Y.CT$vcov )

se.LRm = deltamethod (~ (1-(exp(x1)/(1+exp(x1))))/(exp(x2)/(1+exp(x2))) ,
mean = c(lsens.CT,lspec.CT) , cov = ma_Y.CT$vcov )
```

Then it is possible to construct a neat dataframe summarising the confidence intervals.

```
data.frame( estimate = c(DOR , LRp , LRm) ,
       lci = c(DOR-qnorm(0.975)*se.DOR , LRp-qnorm(0.975)*se.LRp , LRm-
qnorm(0.975)*se.LRm) ,
       uci = c(DOR+qnorm(0.975)*se.DOR , LRp+qnorm(0.975)*se.LRp ,
LRm+qnorm(0.975)*se.LRm),
             row.names = c("DOR", "LR+" , "LR-" )  )
```

6.3 Meta-regression using glmer

6.3.1 Separate meta-analysis for each test

The bivariate model is flexible and can be extended to investigate sources of heterogeneity or to compare the accuracy of two or more tests by adding a covariate to the model. This is relatively straightforward when building a regression model with `glmer`. Moreover, likelihood ratio tests can be used to compare models with or without a covariate term.

While the assumption of equal variances for the random effects of the logit sensitivities and the logit specificities of different subgroups may be reasonable when investigating heterogeneity in the accuracy of a single test, this is not necessarily true when comparing the accuracy of multiple tests. Macaskill and colleagues provide further guidance in Chapter 10 of the Cochrane Handbook for Systematic Reviews of Diagnostic Test Accuracy.[12]

Since it is possible for variances to differ between tests, we begin by meta-analysing each test separately to determine the variances of the random effects for each test. This will elucidate whether the assumption of equal variances for the tests is likely to be reasonable in a model comparing the tests.

6.3.2 Separate meta-analysis for each test

Meta-analysis of CT

```
glmer( formula = cbind(  true , n - true ) ~ 0 + sens + spec + (0+sens +
spec|study),
data = Y.CT , family = binomial  )
```

Meta-analysis of MRI

```
glmer( formula = cbind(  true , n - true ) ~ 0 + sens + spec + (0+sens +
spec|study),
data = Y.MRI , family = binomial  )
```

Examining the variances of the two tests it is apparent that there is a substantial difference in the between study variance of logit sensitivity of each test. Similarly, there is a large difference in the correlation between logit sensitivity and logit specificity for the two tests.

6.3.3 Compare test accuracy

It will be convenient for the interpretation of the model to introduce some new variables at this point.

```
Y$CT  <- 2 - as.numeric(Y$Test)
Y$MRI <- 1 - Y$CT

Y$seCT  <- (Y$CT)*(Y$sens)
Y$seMRI <- (Y$MRI)*(Y$sens)
```

```
Y$spCT  <- (Y$CT)*(Y$spec)
Y$spMRI <- (Y$MRI)*(Y$spec)
```

CT and MRI are dummy variables which identify the test type. Similarly, seCT, seMRI, spCT and spMRI are dummy variables that denote the respective interactions between sens/spec and test type.

Fit the model without the covariate for test type

```
(A = glmer( formula = cbind(  true , n - true ) ~ 0 + sens + spec + (0+sens
+ spec|study), data = Y , family = binomial  ) )
```

Add covariate terms to the model for both logit sensitivity and logit specificity. This model assumes equal variances for both tests.

```
(B = glmer( formula = cbind(  true , n - true ) ~ 0 + seCT + seMRI + spCT +
spMRI + (0+sens + spec|study), data = Y , family = binomial  ))
```

The models can be formally compared using a likelihood ratio test. The R package **lmtest** contains a function lrtest for this purpose.

```
install.packages("lmtest")

library(lmtest)

lrtest(A,B)
```

There is statistical evidence (chi-square = 27.9, 2 df, P<0.0001) that the expected sensitivity and/or specificity differs between CT and MRI. However, further analysis is required to determine if the difference is in sensitivity, specificity, or both.

Is there a statistically significant difference in sensitivity between CT and MRI?

Fit the model assuming sensitivity is the same for CT and MRI but allow specificity to vary with test type.

```
(C = glmer( formula = cbind(  true , n - true ) ~ 0 + sens + spCT + spMRI +
      (0+sens + spec|study), data = Y , family = binomial  ))

lrtest(B,C)
```

Here, there is statistical evidence (chi-square = 19.7, 1 df, P<0.0001) that the expected sensitivity differs between CT and MRI.

Is there a statistically significant difference in specificity between CT and MRI?

Fit the model assuming specificity is the same for CT and MRI but allow sensitivity to vary with test type.

```
(D = glmer( formula = cbind(  true , n - true ) ~ 0 + seCT + seMRI + spec +
      (0+sens + spec|study), data = Y , family = binomial  ))

lrtest(B,D)
```

There is also statistical evidence (chi-square = 13.4, 1 df, P = 0.0002) that the expected specificity differs between CT and MRI.

As previously mentioned, the assumption of equal variances may not be appropriate. There were differences between the variances of the random effects especially for the logit sensitivities as observed from the meta-analysis of each test. Since there are many studies for each test, it should be possible to fit a model with separate variances for the logits of each test.

```
(E = glmer( formula = cbind(  true , n - true ) ~ 0 + seCT + seMRI + spCT +
spMRI  +(0 +seMRI + spMRI |study) +(0 +seCT + spCT |study), data = Y , family
= binomial  ))

lrtest(B,E)
```

Here, there is statistical evidence (chi-square = 8.54, 3 df, p=0.036) that the assumption of equal variances may not be reasonable.

Finally, let's perform a likelihood ratio test comparing the simplest model (A) with no covariate for test type and the most complex model (E) that includes a covariate for test type and allows for separate variances for each test.

```
lrtest(A,E)
```

There is statistical evidence (chi-square = 36.44, 5 df, P<0.0001) that the expected sensitivity and/or specificity differs between CT and MRI.

The between study covariance between logit sensitivity and specificity for each test can be obtained as shown below.

```
(summary(E))$vcov
```

From the above, the covariance between the estimated mean logit sensitivity and mean logit specificity of CT is 0.00355 and that of MRI is –0.00829.

The parameter estimates for CT and MRI from model E can be entered into the corresponding multiple tests analysis in RevMan to produce a SROC plot with summary operating points for CT and MRI. Figure 6 shows how to extract the model estimates and input them into RevMan for this purpose, while Figure 7 shows the corresponding SROC plot.

Table 3 presents the summary estimates from model E, model B (assuming equal variance), and from modelling sensitivity and specificity separately. From model E, the summary estimate obtained for sensitivity of CT was 97.2% (95% CI: 96.2% to 98.0%) compared to 87.7% (83.9% to 90.8%) for MRI, while CT had a summary specificity of 87.3% (84.4% to 89.8%) compared to 69.9% (59.1% to 78.8%) for MRI.

```
> summary(E)
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: cbind(true, n - true) ~ 0 + seCT + seMRI + spCT + spMRI + (0 +
    seMRI + spMRI | study) + (0 + seCT + spCT | study)
   Data: Y

     AIC      BIC   logLik deviance df.resid
   963.3    997.1   -471.7    943.3      206

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.09767 -0.33454  0.04078  0.53007  1.43652

Random effects:
 Groups  Name  Variance Std.Dev. Corr
 study   seMRI 0.1126   0.3356
         spMRI 0.7121   0.8439   -0.52
 study.1 seCT  1.1069   1.0521
         spCT  0.8800   0.9381   0.31
Number of obs: 216, groups:  study, 108

Fixed effects:
      Estimate Std. Error z value Pr(>|z|)
seCT    3.5594     0.1739  20.465  < 2e-16 ***
seMRI   1.9671     0.1624  12.110  < 2e-16 ***
spCT    1.9316     0.1225  15.773  < 2e-16 ***
spMRI   0.8410     0.2406   3.495 0.000474 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
      seCT   seMRI  spCT
seMRI  0.000
spCT   0.167  0.000
spMRI  0.000 -0.212  0.000
>
```

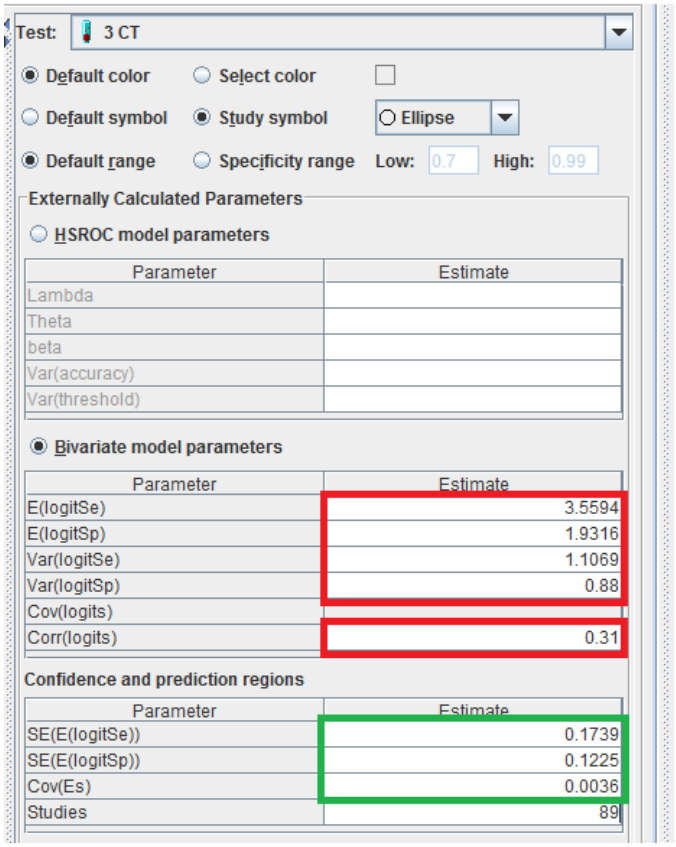

Figure 6: Screenshot of glmer summary output highlighting the values required for input into RevMan

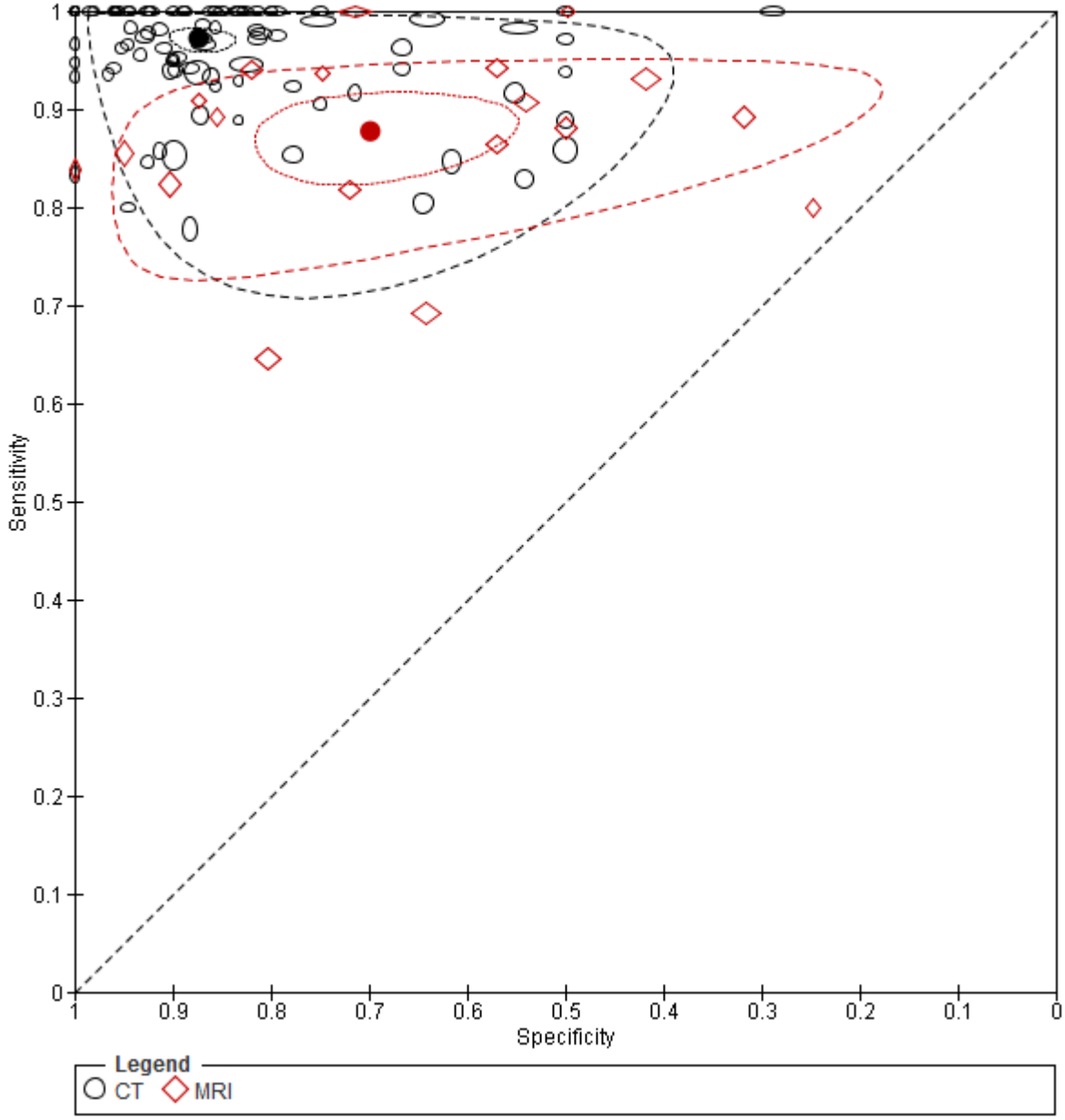

Figure 7: SROC generated using RevMan (based on parameters obtained from model E)

Table 3: Summary sensitivity and specificity (and 95% confidence intervals) of CT and MRI from different models

Method	CT		MRI	
	Sensitivity (95% CI)	Specificity (95% CI)	Sensitivity (95% CI)	Specificity (95% CI)
Separate meta-analysis for each test	97.2 (96.2, 98.0)	87.3 (84.4, 89.8)	87.7 (83.9, 90.8)	69.9 (59.2, 78.8)
Test comparison with equal variances (model B)	97.0 (95.9, 97.8)	87.3 (84.4, 89.7)	89.0 (85.5, 93.2)	70.3 (59.6, 79.7)
Test comparison allowing for different variances (model E)	97.2 (96.1, 98.0)	87.3 (84.4, 89.8)	87.7 (83.9, 90.8)	69.9 (59.1, 78.8)

Summary sensitivities and specificities are presented as percentages.

7 List of references

1. Doebler P. mada: Meta-Analysis of Diagnostic Accuracy. 2015; https://cran.r-project.org/web/packages/mada/index.html. Accessed 5th April 2016.
2. Schiller I. HSROC: Meta-Analysis of Diagnostic Test Accuracy when Reference Test is Imperfect. 2015; https://cran.r-project.org/web/packages/HSROC/index.html. Accessed 5th April 2016.
3. Verde P. bamdit: Bayesian Meta-Analysis of Diagnostic Test Data. 2015; https://cran.r-project.org/web/packages/bamdit/index.html. Accessed 5th April 2016.
4. Nikoloulopoulos A. CopulaREMADA: Copula Mixed Effect Models for Bivariate and Trivariate Meta-Analysis of Diagnostic Test Accuracy Studies. 2015; https://cran.r-project.org/web/packages/CopulaREMADA/index.html. Accessed 5th April 2016.
5. Gasparrini A. mvmeta: Multivariate and Univariate Meta-Analysis and Meta-Regression. 2015; https://cran.r-project.org/web/packages/mvmeta/index.html. Accessed 20th May, 2016.
6. Bates D, Maechler M, Bolker B, Walker S, Christensen RHB, Singmann H, Dai B, Grothendieck G, Green P. lme4: Linear Mixed-Effects Models using 'Eigen' and S4. 2016; https://cran.r-project.org/web/packages/lme4/index.html. Accessed 5th April 2016.
7. Harbord RM, Deeks JJ, Egger M, Whiting P, Sterne JA. A unification of models for meta-analysis of diagnostic accuracy studies. *Biostatistics.* 2007;8(2):239-251.
8. Schuetz GM, Zacharopoulou NM, Schlattmann P, Dewey M. Meta-analysis: noninvasive coronary angiography using computed tomography versus magnetic resonance imaging. *Annals of Internal Medicine.* 2010;152(3):167-177.
9. Rutter CM, Gatsonis CA. A hierarchical regression approach to meta-analysis of diagnostic test accuracy evaluations. *Statistics in Medicine.* 2001;20(19):2865-2884.
10. Reitsma JB, Glas AS, Rutjes AW, Scholten RJ, Bossuyt PM, Zwinderman AH. Bivariate analysis of sensitivity and specificity produces informative summary measures in diagnostic reviews. *Journal of Clinical Epidemiology.* 2005;58(10):982-990.
11. Chu H, Cole SR. Bivariate meta-analysis for sensitivity and specificity with sparse data: a generalized linear mixed model approach (letter to the Editor). *Journal of Clinical Epidemiology.* 2006;59:1331-1331.
12. Macaskill P, Gatsonis C, Deeks J, Harbord R, Takwoingi Y. Chapter 10: Analysing and presenting results. In: Deeks JJ, Bossuyt PM, Gatsonis C, eds. Cochrane Handbook for Systematic Reviews of Diagnostic Test Accuracy Version 1.0. The Cochrane Collaboration; 2010. Accessed at http://dta.cochrane.org on 5th April 2016.