# NMA Example Document

Elaona Lemoto

6/3/2023

The purpose of this rmarkdown file is to introduce the data preparation and model code needed to run Lu and Ades [1], Contrast-Based, and Arm-Based NMA (I am using the Hong et al [2] definition of Contrast-based and Arm-Based NMA). In addition, this document will show how to store model results, run forest plots of the results, and utilize the sucra plots of results. The process will assume the following:

1. These aggregated outcome data follow binomial distribution.
2. The dataframe is in long format which means each row is a unique arm within a study/group. If your project-specific data are not in this form, you can easily transform it using R-package tidyverse and function pivot_longer().

#Data

First we begin with the data preparation. I have code for generating outcome data for 30 studies comparing 3 treatments in long format. Of the 30 studies, 10 studies compare treatment A vs. B, 10 studies compare treatment A vs. C, and 10 studies compare B vs. C.

```
#Num. studies
I <- 30
#size of studies
size <- 200

#true probability of event for trt b and trt c
true_p<-c(0.4, 0.38)

# (Prob. trt A)
mu_i <-runif(1, 0.5-.01, 0.5+.01)
Delta <- cbind(mu_i,
                 true_p[1],
                 true_p[2])
#Using rbinom to generate treatment outcome values
Ybar <- cbind(rbinom(I,size, Delta[, 1] ),
               rbinom(I,size, Delta[, 2] ),
               rbinom(I,size, Delta[, 3] ))
  MCAR.ind <- order(runif(I))
  ind1 <- MCAR.ind[1:10]    # A vs B
  ind2 <- MCAR.ind[11:20]  # A vs B
  ind3 <- MCAR.ind[21:I]  # B vs C
  Y1.g1 <- Ybar[ind1,]
  Y1.g2 <- Ybar[ind2,]
  Y1.g3 <- Ybar[ind3,]

  temp11 <- cbind(1:10, Y1.g1[,1], size, 1, 1)#study[], y[], n[], trt[], baseline[]
```

```r
temp21 <- cbind(1:10, Y1.g1[,2], size, 2, 1)

temp12<- cbind(11:20, Y1.g2[,1], size, 1, 1)
temp31<- cbind(11:20, Y1.g2[,3], size, 3, 1)

temp22<- cbind(21:30, Y1.g3[,2], size, 2, 2)
temp32<- cbind(21:30, Y1.g3[,3], size, 3, 2)


#Generated NMA data
nma_data <- data.frame(rbind(temp11,temp21,temp12,temp31, temp22, temp32))
nma_data <- nma_data[order(nma_data[,1],nma_data[,5]),]
colnames(nma_data)<-c("study", "y", "n", "trt", "base")

#Data used for models
N<-nrow(nma_data);
NS<-I;
NT<-3;
s<-nma_data[,1];
y<-nma_data[,2];
n<-nma_data[,3];
t<-nma_data[,4];
```

# Model Specification

Now I will discuss the different models. Before introducing the code, I will discuss model specification and their assumptions. After, I will introduce the code for running the models with MCMC procedure. For simplicity, we are including the models as functions and running them within this markdown file. However, you can also run the models externally from a text file. After specifing the R models as functions, we use the Rjags package to execute the MCMC procedure. The NMA data generated above must be entered as a list as well as the initial values. The Lu and Ades model requires the data input to be in matrix form.

##Lu and Ades Model

The Lu and Ades model, a type of Contrast-Based Model, requires study-specific reference treatments ($b_i$) to be defined for each study i. The model is written as,

$$\theta^a_{ik} = \alpha^a_{ib_i} + \delta^c_{ib_ik} \text{ for k} \in R_i$$

From the LA model above, $\alpha_{ib_i}$ describes the study intercept, or in our case, the log odds in arm $b_i$ of study i and it is our fixed effect. The study-specific treatment contrast is $\delta^c_{ib_ik}$ and compares treatment k to treatment $b_i$ for treatment $k \in R_i$. Additionally, $\delta^c_{ib_ik} = 0$ if $k = b_i$. The
$\delta^c_{ib_ik}$ is modeled as,

$$\delta^c_{ib_ik} \sim N(\mu^c_k - \mu^c_{b_i}, \sigma^{c2})$$

$\sigma^{c2}$ describes the contrast heterogeneity variance (also the c is a superscript, it's not a product of powers). While the model specification above assumes homogenous variance accross contrasts, you can allow the contrast heterogeneity vary between treatment contrasts. The code below allows for contrast-specific variance.

###Running Lu and Ades Model with Random Effects (LARE)

```
#Lu and Ades Model with Random Effects Model Function

LARE<-function(){
for (i in 1:Nstudy){
  for (j in 1:Narm[i]){
      y[i,j] ~ dbinom(mean[i,j],n[i,j])
      logit(mean[i,j]) <- mu[i] + lor[i,j]
    }
  }
  for (i in 1:Nstudy){
    w[i,1] <- 0
    lor[i,1] <- 0
    for (j in 2:Narm[i]){
      lor[i,j] ~ dnorm(md[i,j], inv[i,j])
      md[i,j] <- d[drug[i,j]] - d[drug[i,1]] + sw[i,j]
      #Incorporating information from overall treatment effect back
      #into study specific treatment effect
      w[i,j] <- lor[i,j] - d[drug[i,j]] + d[drug[i,1]]
      sw[i,j] <- sum(w[i,1:(j-1)])/(j-1)
      #contrast-specific variance
      inv[i,j] <- inv.d*2*(j-1)/j
    }
  }
  for (j in 1:Nstudy) {
      mu[j] ~ dnorm(0, 0.01) }
  d[1] <- 0
  for (k in 2:Ndrug) {
     d[k] ~ dnorm(0, 0.01)
    }
  tau ~ dunif(0.01, 10)
  inv.d <- 1/pow(tau, 2)
  # ranking
  mp <- mean(mu[])
  for (k in 1:Ndrug) { G[k] <- exp(mp + d[k])/(1+exp(mp + d[k])) }
  T.rank <- rank(G)
  for (k in 1:Ndrug) {
    rk[k] <- T.rank[k]
    best1[k] <- equals(rk[k],1)
    best2[k] <- equals(rk[k],2)
    best3[k] <- equals(rk[k],3)
    best12[k] <- best1[k] + best2[k]
  }
}
### Running the LARE Model ###
##Additional data preparation for this model
drug_list<-unique(nma_data$trt)
Narm <- as.numeric(table(nma_data$study))
n.obs <- matrix(NA,nrow=NS, ncol=max(Narm))
n.eve <- matrix(NA,nrow=NS, ncol=max(Narm))
dr <- matrix(NA,nrow=NS, ncol=max(Narm))
study<-unique(nma_data$study)
for (i in 1:NS){
  n.obs[i,1:Narm[i]] <- nma_data$n[nma_data$study==study[i]]
```

```
  n.eve[i,1:Narm[i]] <- nma_data$y[nma_data$study==study[i]]
  dr[i,1:Narm[i]] <- match(nma_data$trt[nma_data$study==study[i]],drug_list)
}
##putting data into list form
data_LA <- list('Narm'=Narm, 'Nstudy'=NS,'Ndrug'=NT, 'drug'=dr,'y'=n.eve,'n'=n.obs)
init_LA <- list(list(mu=rep(0,max(NS)), d=c(NA,rep(0,max(t)-1))),
             list(mu=rep(0,max(NS)), d=c(NA,rep(0,max(t)-1))))
para_LA <- c('d','tau','best1', 'best2', 'best3')
fit_LA <- jags(data=data_LA, inits=init_LA, para_LA,
                     n.iter=20000, n.burnin = 5000, n.chains = 2, n.thin = 1,
                     DIC=TRUE, model.file=LARE)
```

```
## module glm loaded

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 60
##    Unobserved stochastic nodes: 63
##    Total graph size: 505
##
## Initializing model
```

```
#output data
fit_LA$BUGSoutput$summary[,c(1, 3, 7)]
```

```
##                   mean         2.5%       97.5%
## best1[1]   0.00000000   0.00000000   0.0000000
## best1[2]   0.01653333   0.00000000   0.0000000
## best1[3]   0.98346667   1.00000000   1.0000000
## best2[1]   0.00000000   0.00000000   0.0000000
## best2[2]   0.98346667   1.00000000   1.0000000
## best2[3]   0.01653333   0.00000000   0.0000000
## best3[1]   1.00000000   1.00000000   1.0000000
## best3[2]   0.00000000   0.00000000   0.0000000
## best3[3]   0.00000000   0.00000000   0.0000000
## d[1]       0.00000000   0.00000000   0.0000000
## d[2]      -0.34112050  -0.46277678  -0.2219380
## d[3]      -0.47374655  -0.59634126  -0.3557808
## deviance 402.96806379 384.75331323 422.7029373
## tau        0.10299863   0.01556469   0.2216161
```

```
#saving treatment effect output
LA_trt_results<-data.frame(fit_LA$BUGSoutput$summary[,c(1, 3, 7)])
LA_trt_results <- tibble::rownames_to_column(LA_trt_results, "drug_list")
LA_trt_results<-LA_trt_results%>%
  filter(drug_list %in% c("d[1]", "d[2]", "d[3]"))

#saving rank probability of treatment to be the best treatment (number 1)
LA_rank_prob<-data.frame(fit_LA$BUGSoutput$summary[,c(1, 3, 7)])
LA_rank_prob <- tibble::rownames_to_column(LA_rank_prob, "best")
```

```
LA_rank_prob<-data.frame(cbind(LA_rank_prob[LA_rank_prob$best%in%c("best1[1]", "best1[2]", "best1[3]"),1
                               LA_rank_prob[LA_rank_prob$best%in%c("best2[1]", "best2[2]", "best2[3]"),1
                               LA_rank_prob[LA_rank_prob$best%in%c("best3[1]", "best3[2]", "best3[3]"),1
rownames(LA_rank_prob)<-c("trt_1", "trt_2", "trt_3")
```

Besides the model and the data input, initial values are required for study specifc means as well as treatment effects. Lastly, the we can use the BUGSoutput function to obtain the estimated posterior means of treatment effects as well as other estimates defined by 'para_LA'.

##Contrast-Based Model

The CB model under the missing data framework takes into consideration all arms, not just the observed arms like in the Lu and Ades model. The model specification is as follows,

$$\theta_{ik}^{a} = \alpha_{i1}^{a} + \delta_{i1k}^{c}$$
$$\delta_{i}^{c} = (\delta_{i12}^{c}, ..., \delta_{i1k}^{c}) \sim MVN(\mu^{c}, \Sigma^{c})$$

Instead of having study specific baselines treatments($b_i$) as the LA model, the CB model assumes the baseline treatment is treatment 1 (having ordered treatments). Additionally, $\delta_{i11}^{c} = 0$ and $\mu^{c} = (\mu_{12}^{c}, ..., \mu_{1k})$ is the vector of overall mean effects for treatments 2,...,k compared to reference treatment 1. Finally $\Sigma^{c}$, the heterogeniety variance matrix, requires some care. In the coded model below, we assume

$$\Sigma^{c} = \sigma^{a2} P_{k}(\rho^{a})$$

Where $P_k(\rho^a)$ is the k x k matrix with all diagonal elements equal to $\sigma^{a2}$ and off-diagonal elements equal to $\rho^{a}.\sigma^{a2}$ describes the treatment specific variabilty and $\rho^a$ estimates the relationships among the different treatments. There are several different ways to characterize the covariance matrix. White et al [3] discusses in greater detail how the covariance matrix changes across the different NMA models.

###Running Contrast-Based Model

```
CBWish<-function(){
  for (i in 1:Narm){
    y[i] ~ dbinom(mean[i],n[i])
    logit(mean[i]) <- mu[study[i]] + delta[study[i],drug[i]]*(1-equals(drug[i],1))
  }
  for (j in 1:Nstudy){
    delta[j,1:Ndrug] ~ dmnorm(d[1:Ndrug], invR[1:Ndrug, 1:Ndrug])
  }
  invR[1:Ndrug, 1:Ndrug] ~ dwish(Omega[1:Ndrug,1:Ndrug], Ndrug)
  R[1:Ndrug, 1:Ndrug] <- inverse(invR[ , ])
  for (k in 1:Ndrug){
    tau[k] <- sqrt(R[k,k])
  }
  for (j in 1:Ndrug){
    for (k in (j+1):Ndrug){
      rho[j,k] <- R[j,k]/(tau[j]*tau[k])
    }
  }
  for (j in 1:Nstudy) { mu[j] ~ dnorm(0, 0.01) }
  d[1] <- 0
  for (k in 2:Ndrug) { d[k] ~ dnorm(0, 0.01) }
  # ranking
```

```
  mp <- mean(mu[])
  for (k in 1:Ndrug) { T[k] <- exp(mp + d[k])/(1+exp(mp + d[k])) }
  T.rank <- rank(T)
  for (k in 1:Ndrug) {
    rk[k] <- T.rank[k]
    best1[k] <- equals(rk[k],1)
    best2[k] <- equals(rk[k],2)
    best3[k]<-equals(rk[k],3)
    best12[k] <- best1[k] + best2[k]
  }
}
### Running Contrast Based  Model  ########
  data_CB <- list('Narm'=N, 'Nstudy'=NS,
                  'Ndrug'=NT, 'study'= s, 'drug'=t,
                  'y'=y, 'n'=n ,'Omega'=diag(rep(0.2,times=3)))
  inits_CB<- list(list(mu=rep(0,30), d=c(NA,rep(1,2))),
                       list(mu=rep(0,30), d=c(NA,rep(0,2))))
  para_CB<-c( "d", "tau", "best1", 'best2', 'best3')
  fit_CB<-jags(data=data_CB, inits=inits_CB, para_CB,
                  n.iter=20000, n.burnin = 1500, n.chains = 2, n.thin = 1,
                  DIC=TRUE, model.file=CBWish)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 60
##    Unobserved stochastic nodes: 63
##    Total graph size: 614
##
## Initializing model
```

```
  #output data
  fit_CB$BUGSoutput$summary[,c(1, 3, 7)]
```

```
##                    mean        2.5%       97.5%
## best1[1]  7.297297e-04   0.0000000   0.0000000
## best1[2]  5.840541e-02   0.0000000   1.0000000
## best1[3]  9.408649e-01   0.0000000   1.0000000
## best2[1]  8.108108e-05   0.0000000   0.0000000
## best2[2]  9.409730e-01   0.0000000   1.0000000
## best2[3]  5.894595e-02   0.0000000   1.0000000
## best3[1]  9.991892e-01   1.0000000   1.0000000
## best3[2]  6.216216e-04   0.0000000   0.0000000
## best3[3]  1.891892e-04   0.0000000   0.0000000
## d[1]      0.000000e+00   0.0000000   0.0000000
## d[2]     -3.433745e-01  -0.5017170  -0.1867059
## d[3]     -4.762566e-01  -0.6313891  -0.3220968
## deviance  3.978597e+02 380.1354833 418.5589762
## tau[1]    4.220837e-01   0.1637526   1.1566094
## tau[2]    2.230985e-01   0.1364741   0.3529156
## tau[3]    2.152907e-01   0.1319694   0.3420177
```

```
  #saving treatment effect output
CB_trt_results<-data.frame(fit_CB$BUGSoutput$summary[,c(1, 3, 7)])
CB_trt_results <- tibble::rownames_to_column(CB_trt_results, "drug_list")
CB_trt_results<-CB_trt_results%>%
  filter(drug_list %in% c("d[1]", "d[2]", "d[3]"))
#saving rank probability of treatment to be the best treatment (number 1)
CB_rank_prob<-data.frame(fit_CB$BUGSoutput$summary[,c(1, 3, 7)])
CB_rank_prob <- tibble::rownames_to_column(CB_rank_prob, "best")

CB_rank_prob<-data.frame(cbind(CB_rank_prob[CB_rank_prob$best%in%c("best1[1]", "best1[2]", "best1[3]"),1
                               CB_rank_prob[CB_rank_prob$best%in%c("best2[1]", "best2[2]", "best2[3]"),1
                               CB_rank_prob[CB_rank_prob$best%in%c("best3[1]", "best3[2]", "best3[3]"),1
rownames(CB_rank_prob)<-c("trt_1", "trt_2", "trt_3")
```

Running the contrast-based model requires the input data to be in list format. Additionally, the matrix Omega used for the inverse wishart prior also requires to be in matrix form. For this model, I chose a 3 x 3 matrix (dimensions determined by number of treatments) with 0.2 on the diagonal and 0 in the off-diagonal, but other choices can be considered especially as number of treatments increase.

##Arm-Based Model

The arm-based model is similar to the contrast-based model, however, it handles the treatments symetrically.

$$\theta_{ik}^{a} = \mu_{k}^{c} + \nu_{ik}^{a}$$
$$\nu_{i}^{a} = (\nu_{i1}^{a}, ...., \nu_{ik}^{a}) \sim MVN(0, \Sigma^{a})$$

Above, $\mu_k^c$ describes the fixed mean effect of treatment k relative to treatment 1 and $\nu_{ik}^a$ is the study-level treatment random effect. The treatment effect, log odds ratio in this case, can be estimated as the the difference between $(\mu_k - \mu_1)$. $\Sigma^a$ of the random effects can also be customized differently. For the model code below, we assume

$$\Sigma^{a} = \sigma^{a2} P_k(\rho^a)$$

Therefore, $\Sigma^a$ can be described as having treatment-specific variability $\sigma^{a2}$ along the diagonal and the correlation among treatments, $\rho^a$, on the off-diagonal. This characterization of $\Sigma^a$ for the study-level treatment random effect can be altered as mentioned previously.

###Running Arm-Based Model

```
ABWish<-function(){
  for (i in 1:Narm){
    y[i] ~ dbinom(mean[i],n[i])
    logit(mean[i]) <- mu[drug[i]] + v[study[i],drug[i]]
  }
  for (j in 1:Nstudy) {
  v[j,1:Ndrug] ~ dmnorm(zero.AB[1:Ndrug], invR[1:Ndrug, 1:Ndrug]) }
  invR[1:Ndrug, 1:Ndrug] ~ dwish(Omega[1:Ndrug,1:Ndrug], Ndrug)
  R[1:Ndrug, 1:Ndrug] <- inverse(invR[ , ])
  for (k in 1:Ndrug){
    tau[k] <- sqrt(R[k,k])
  }
  for (j in 1:Ndrug){
    for (k in (j+1):Ndrug){
      rho[j,k] <- R[j,k]/(tau[j]*tau[k])
```

```
    }
  }
  for (k in 1:Ndrug) { mu[k] ~ dnorm(0, 0.001) }
  for (k in 1:Ndrug) { lor[k] <- mu[k] - mu[1] }
  # ranking
  for (k in 1:Ndrug) { G[k] <- exp(mu[k])/(1+exp(mu[k])) }
  T.rank <- rank(G)
  for (k in 1:Ndrug) {
    rk[k] <- T.rank[k]
    best1[k] <- equals(rk[k],1)
    best2[k] <- equals(rk[k],2)
    best3[k]<-equals(rk[k],3)
    best12[k] <- best1[k] + best2[k]
  }
}
### Running Arm Based  Model  ########
  data_AB <- list('Narm'=N, 'Nstudy'=NS,
                  'Ndrug'=NT, 'study'= s, 'drug'=t,
                  'y'=y, 'n'=n ,'Omega'=diag(rep(0.2,times=3)),
                  'zero.AB' = (rep(0, times=3)))
  inits_AB<- list(list(mu=rep(0,3)),
                       list(mu=rep(0,3)))
  para_AB<-c( "lor", "tau", "best1", "best2", "best3")
  fit_AB<-jags(data=data_AB, inits=inits_AB, para_AB,
                 n.iter=20000, n.burnin = 1500, n.chains = 2, n.thin = 1,
                 DIC=TRUE, model.file=ABWish)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 60
##    Unobserved stochastic nodes: 34
##    Total graph size: 521
##
## Initializing model
```

```
  #output data
  fit_AB$BUGSoutput$summary[,c(1, 3, 7)]
```

```
##                   mean        2.5%       97.5%
## best1[1]  0.000000e+00   0.0000000   0.0000000
## best1[2]  3.075676e-02   0.0000000   1.0000000
## best1[3]  9.692432e-01   0.0000000   1.0000000
## best2[1]  2.702703e-05   0.0000000   0.0000000
## best2[2]  9.692432e-01   0.0000000   1.0000000
## best2[3]  3.072973e-02   0.0000000   1.0000000
## best3[1]  9.999730e-01   1.0000000   1.0000000
## best3[2]  0.000000e+00   0.0000000   0.0000000
## best3[3]  2.702703e-05   0.0000000   0.0000000
## deviance  3.917837e+02 376.3771126 409.4981192
## lor[1]    0.000000e+00   0.0000000   0.0000000
## lor[2]   -3.787299e-01  -0.5206055  -0.2364783
```

```
## lor[3]   -5.162747e-01  -0.6578804  -0.3754587
## tau[1]    1.645694e-01   0.1122748   0.2405145
## tau[2]    1.840971e-01   0.1230899   0.2709722
## tau[3]    1.772406e-01   0.1189031   0.2619346
```

```
  #saving treatment effect output
AB_trt_results<-data.frame(fit_AB$BUGSoutput$summary[,c(1, 3, 7)])
AB_trt_results <- tibble::rownames_to_column(AB_trt_results, "drug_list")
AB_trt_results<-AB_trt_results%>%
  filter(drug_list %in% c("lor[1]", "lor[2]", "lor[3]"))
#saving rank probability of treatment to be the best treatment (number 1)
AB_rank_prob<-data.frame(fit_AB$BUGSoutput$summary[,c(1, 3, 7)])
AB_rank_prob <- tibble::rownames_to_column(AB_rank_prob, "best")

AB_rank_prob<-data.frame(cbind(AB_rank_prob[AB_rank_prob$best%in%c("best1[1]", "best1[2]", "best1[3]"),1
                               AB_rank_prob[AB_rank_prob$best%in%c("best2[1]", "best2[2]", "best2[3]"),1
                               AB_rank_prob[AB_rank_prob$best%in%c("best3[1]", "best3[2]", "best3[3]"),1
rownames(AB_rank_prob)<-c("trt_1", "trt_2", "trt_3")
```

The arm-based model requires the data to be in list format, very similar to the list format required in the contrast-based code. The difference here is that input data are needed for zero mean in the study-level random effects. Additionally, the parameters of interest have changed. For the arm-based model, we want the posterior esimtates for the log odds ratios.

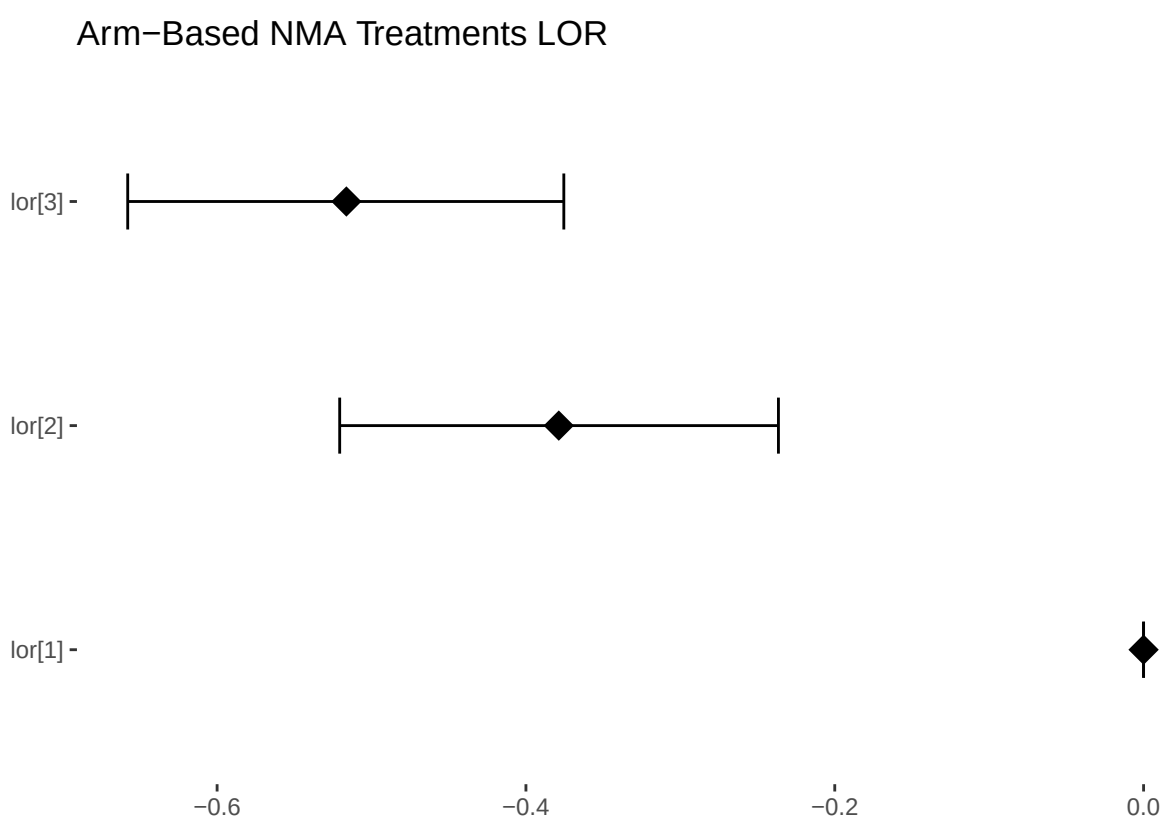
#Model Result Plots

## Forest plot

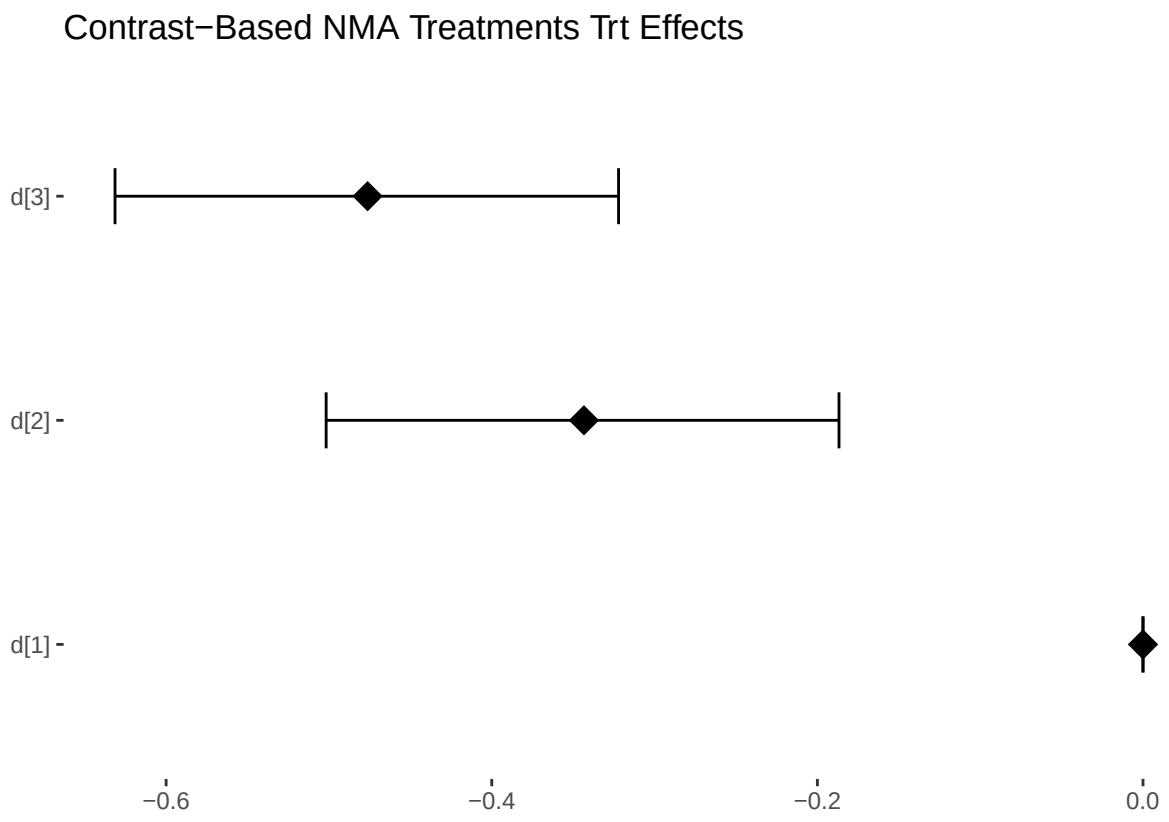
Below is code to run the lor or treament effects derived from the model output code above. There are defintely better ways to run a forest plot and you can look up additional functions from ggplot to improve the forest plot below, but the basics are there for you to at least see how it works.

```
#### Cleaning the data before running the forest plot ####
ABresults<-AB_trt_results%>%
  mutate(LL = as.numeric(X2.5.),
         UL = as.numeric(X97.5.),
         mean = as.numeric(mean))%>%
  filter(!(drug_list==1))
CBresults<-CB_trt_results%>%
  mutate(LL = as.numeric(X2.5.),
         UL = as.numeric(X97.5.),
         mean = as.numeric(mean))%>%
  filter(!(drug_list==1))
LAresults<-LA_trt_results%>%
  mutate(LL = as.numeric(X2.5.),
         UL = as.numeric(X97.5.),
         mean = as.numeric(mean))%>%
  filter(!(drug_list==1))

##Runnning forest plots##
ggplot(ABresults, aes(y = drug_list, x =mean )) +
  geom_point(shape = 18, size = 5) +
  geom_errorbarh(aes(xmin = LL, xmax = UL), height = 0.25)+
```

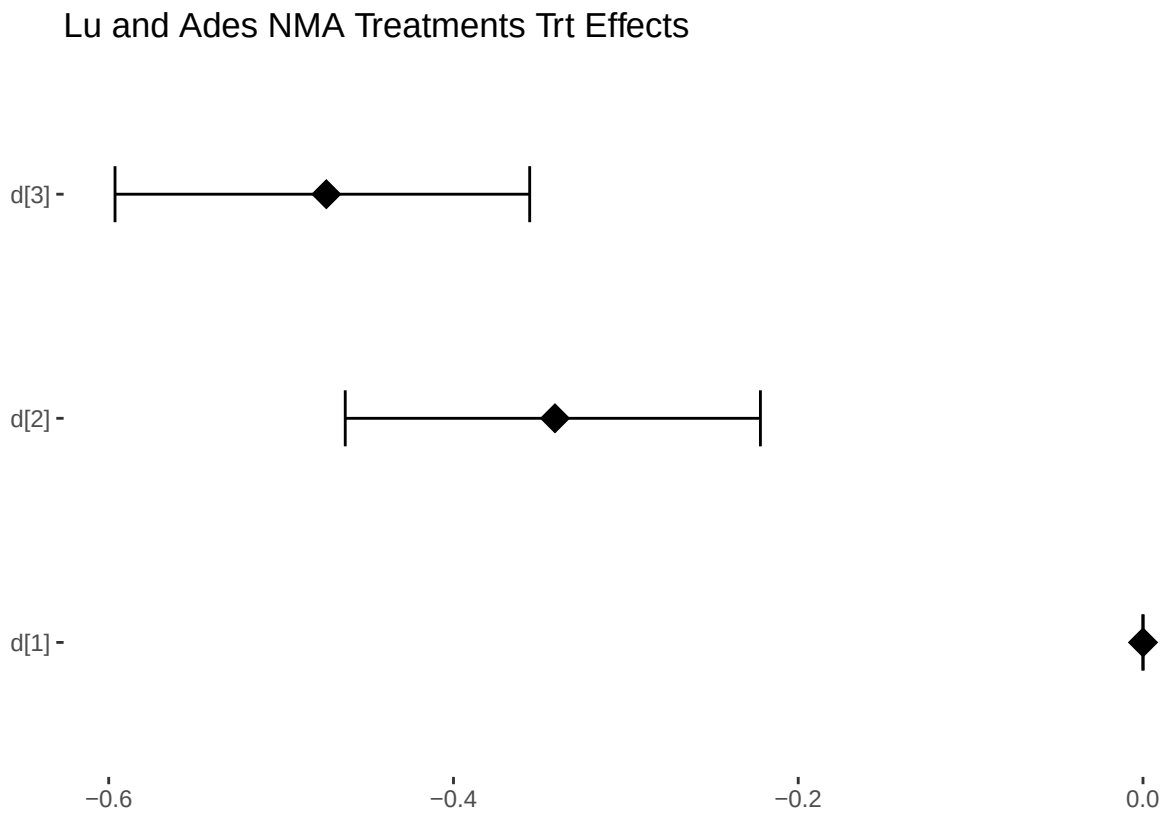
```
  theme(panel.border = element_blank(),
        panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  labs(y = "", x="", title = "Arm-Based NMA Treatments LOR")
```




```
ggplot(CBresults, aes(y = drug_list, x =mean )) +
  geom_point(shape = 18, size = 5) +
  geom_errorbarh(aes(xmin = LL, xmax = UL), height = 0.25)+
  theme(panel.border = element_blank(),
        panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  labs(y = "", x="", title = "Contrast-Based NMA Treatments Trt Effects")
```




```
ggplot(LAresults, aes(y = drug_list, x =mean )) +
  geom_point(shape = 18, size = 5) +
  geom_errorbarh(aes(xmin = LL, xmax = UL), height = 0.25)+
  theme(panel.border = element_blank(),
        panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  labs(y = "", x="", title = "Lu and Ades NMA Treatments Trt Effects")
```




##Sucra Plots

From the model output above, I also captured the rank probability of each treament being first best, second best, and third best. The result is a matrix where each column is the rank probability. If you are familiar with the gemtc package, the matrix derived is the same as using the rank probability function. Now the SUCRA score is a metric to evaluate which treatment in a network is likely to be the most effective in the context of NMA. The SUCRA score is calculated using the formula described in Salanti, Ades and Ioannidis (2011):

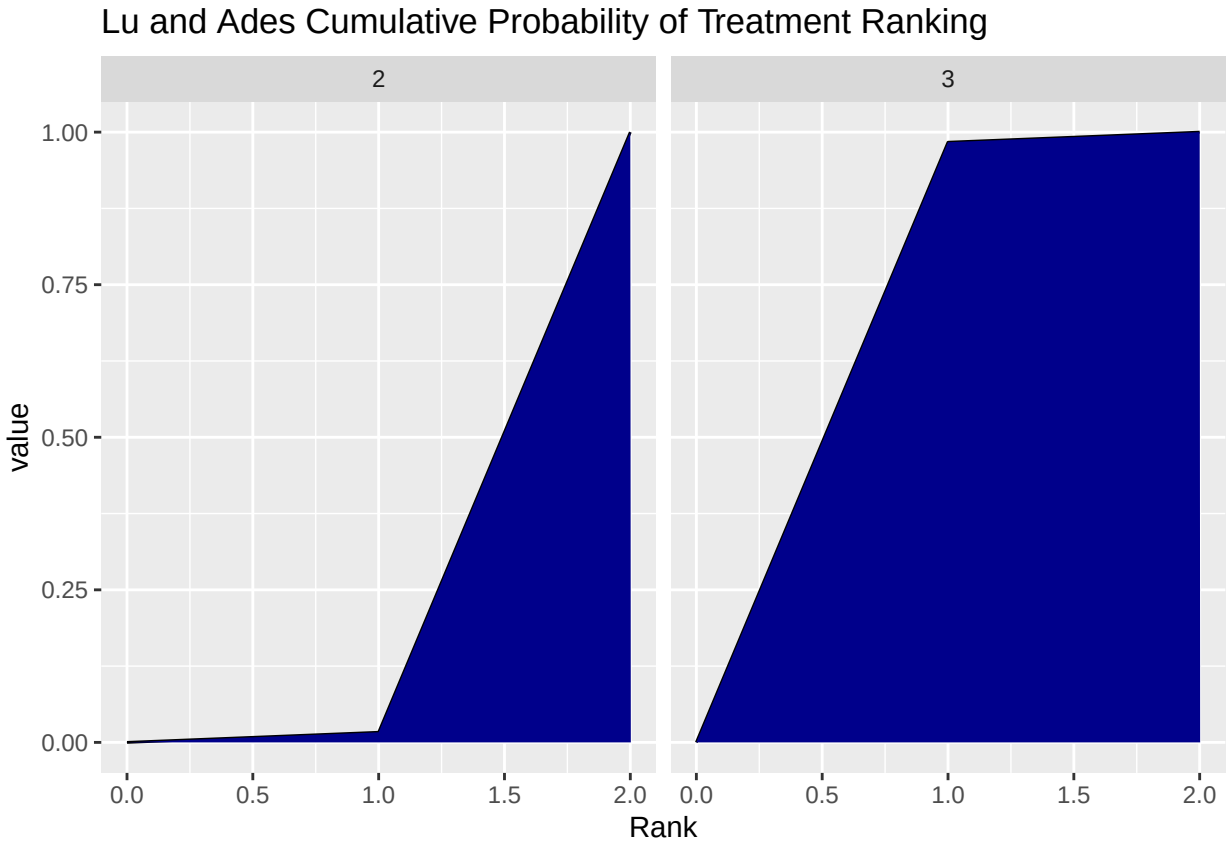
$$SUCRA_k = \frac{\sum_{b=1}^{a-1} cum_{kb}}{a-1}$$

Where k is some treatment within the network, a are all the competing treatments, b are the b = 1,...a-1 best treatments.

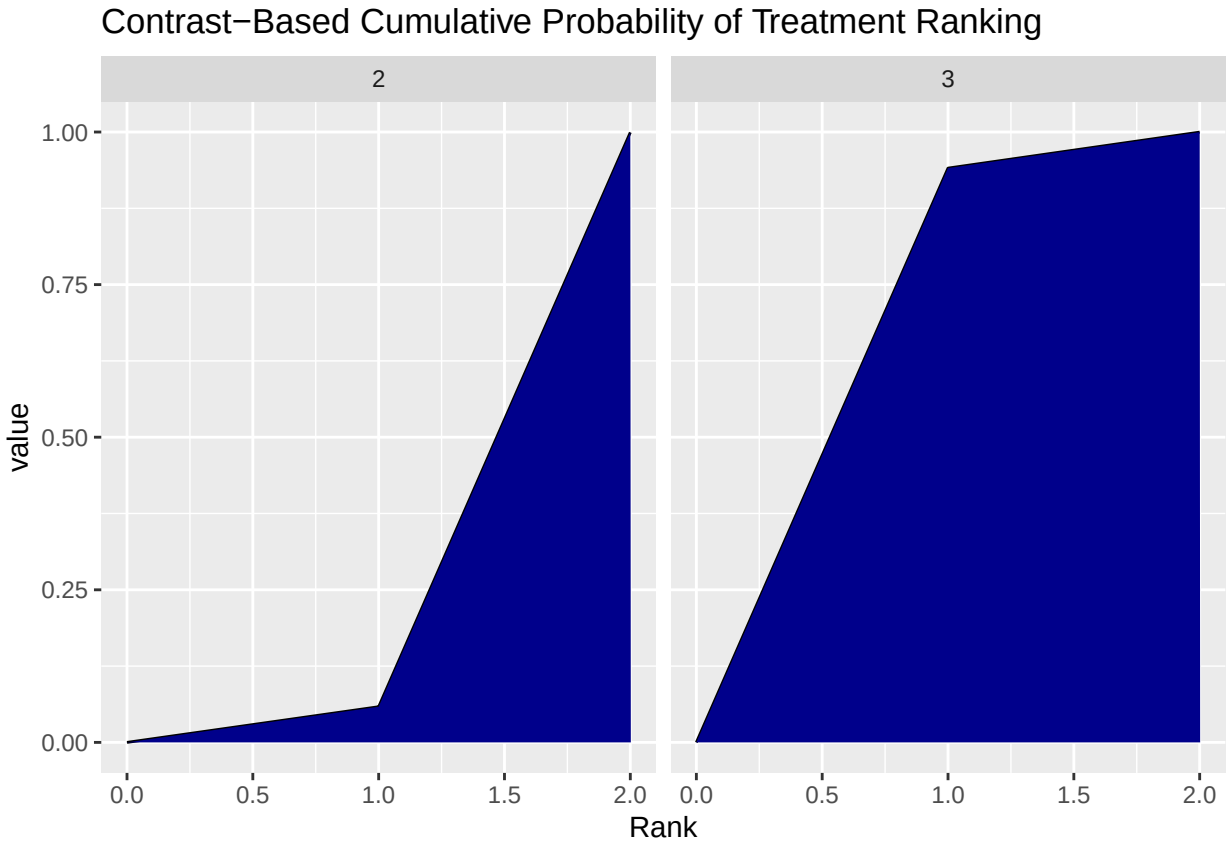
First we can calculate SUCRA using the sucra function. Additionally, you can plot the sucra values using a simple bar plot. Another plot of interest related to SUCRA plots the cumulative probabilities for each treatment. The code for the cumulative probability plots are below. For simplicity, I removed the third best treatment (placebo).

```
  ### SUCRA
LA_sucra<-sucra(LA_rank_prob)
CB_sucra<-sucra(CB_rank_prob)
AB_sucra<-sucra(AB_rank_prob)

treatment<-c(1,2,3)
```

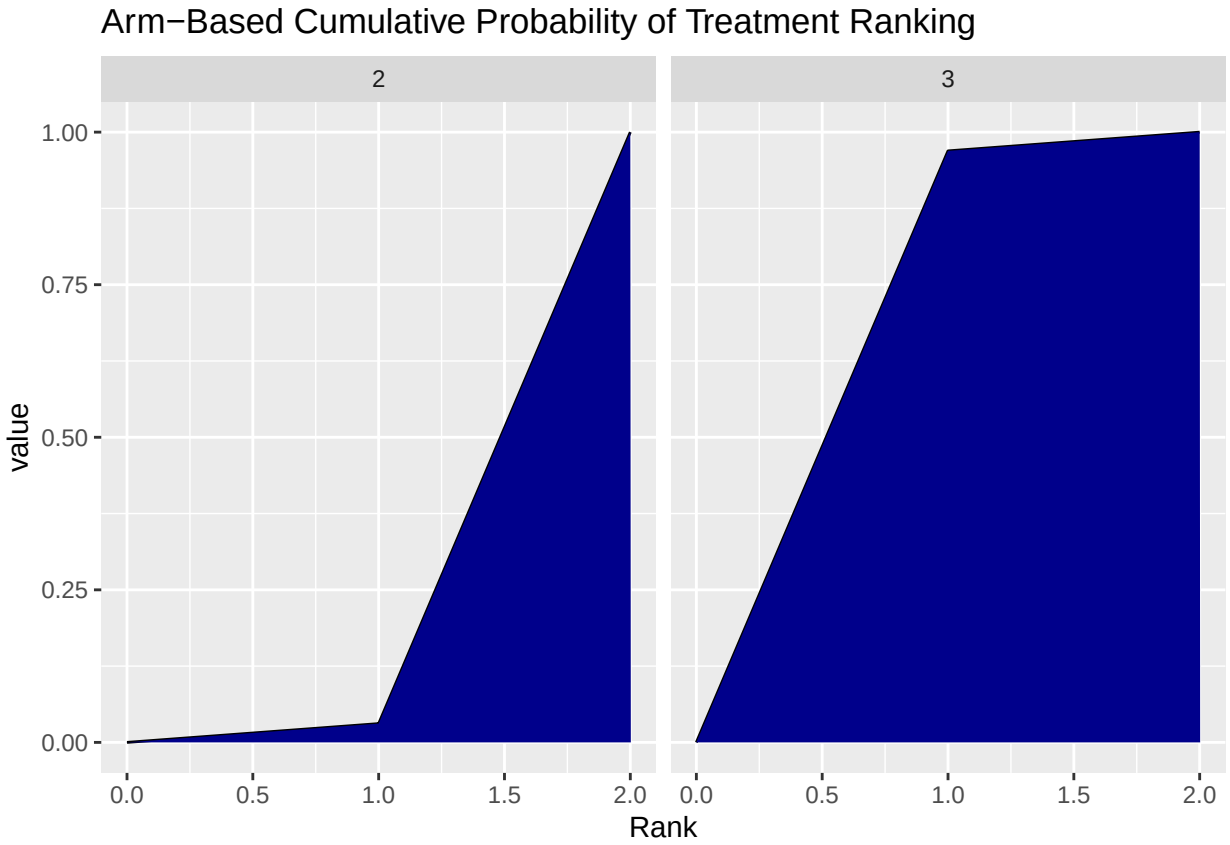
```r
## Manually creating cumulative probability plots
  LAcum.prob<-cbind(treatment, LA_rank_prob)
  LAcum.prob <- LAcum.prob%>%
    mutate('0' = 0,
           '1' = X1,
           '2' = X1 + X2,
           '3'=  X1 + X2 + X3)%>%
    pivot_longer(!treatment,names_to = "prob", values_to ="value")%>%
    filter(!(treatment==1), prob %in% c("0","1", "2"))%>%
    mutate(treatment=as.factor(treatment),
           prob = as.numeric(prob))
  CBcum.prob<-cbind(treatment, CB_rank_prob)
  CBcum.prob <- CBcum.prob%>%
    mutate('0' = 0,
           '1' = X1,
           '2' = X1 + X2,
           '3'=  X1 + X2 + X3)%>%
    pivot_longer(!treatment,names_to = "prob", values_to ="value")%>%
    filter(!(treatment==1), prob %in% c("0","1", "2"))%>%
    mutate(treatment=as.factor(treatment),
           prob = as.numeric(prob))

  ABcum.prob<-cbind(treatment, AB_rank_prob)
  ABcum.prob <- ABcum.prob%>%
    mutate('0' = 0,
           '1' = X1,
           '2' = X1 + X2,
           '3'=  X1 + X2 + X3)%>%
    pivot_longer(!treatment,names_to = "prob", values_to ="value")%>%
    filter(!(treatment==1), prob %in% c("0","1", "2"))%>%
    mutate(treatment=as.factor(treatment),
           prob = as.numeric(prob))
  ##Plotting Cumulative Probabilty Ranking

ggplot(LAcum.prob,aes(prob,value))+
      geom_line()+
      geom_area( fill = "darkblue")+
    facet_wrap(~treatment)+
    labs(title="Lu and Ades Cumulative Probability of Treatment Ranking",
         x="Rank")
```




```
ggplot(CBcum.prob,aes(prob,value))+
      geom_line()+
      geom_area( fill = "darkblue")+
    facet_wrap(~treatment)+
    labs(title="Contrast-Based Cumulative Probability of Treatment Ranking",
         x="Rank")
```




```
ggplot(ABcum.prob,aes(prob,value))+
      geom_line()+
      geom_area( fill = "darkblue")+
    facet_wrap(~treatment)+
    labs(title="Arm-Based Cumulative Probability of Treatment Ranking",
         x="Rank")
```




#References: 1. Lu G, Ades AE. Assessing evidence inconsistency in mixed treatment comparisons. J Am Stat Assoc. 2006;101:447-459. 2. Hong H, Chu H, Zhang J, Carlin BP. A Bayesian missing data framework for generalizedmultiple outcome mixed treatment comparisons. Res Synth Methods. 2016;7(1):6-22. 3. Karahalios A, McKenzie JE, White IR. Contrast-Based and Arm-Based Models for Network Meta-Analysis. Methods Mol Biol. 2022;2345:203-221. doi: 10.1007/978-1-0716-1566-9_13. PMID: 34550593. 4.Georgia Salanti, A.E. Ades, John P.A. Ioannidis,Graphical methods and numerical summaries for presenting results from multiple-treatment meta-analysis: an overview and tutorial,Journal of Clinical Epidemiology,Volume 64, Issue 2,2011,Pages 163-171,ISSN 0895-4356,