National Institute for Health and Care Excellence

FINAL

Menopause (update)

[B1] TSU NMA software code

NICE guideline NG23

Supplement 4

November 2024

FINAL

Supplementary material was developed by the NICE Technical Support Unit

Disclaimer

The recommendations in this guideline represent the view of NICE, arrived at after careful consideration of the evidence available. When exercising their judgement, professionals are expected to take this guideline fully into account, alongside the individual needs, preferences and values of their patients or service users. The recommendations in this guideline are not mandatory and the guideline does not override the responsibility of healthcare professionals to make decisions appropriate to the circumstances of the individual patient, in consultation with the patient and/or their carer or guardian.

Local commissioners and/or providers have a responsibility to enable the guideline to be applied when individual health professionals and their patients or service users wish to use it. They should do so in the context of local and national priorities for funding and developing services, and in light of their duties to have due regard to the need to eliminate unlawful discrimination, to advance equality of opportunity and to reduce health inequalities. Nothing in this guideline should be interpreted in a way that would be inconsistent with compliance with those duties.

NICE guidelines cover health and care in England. Decisions on how they apply in other UK countries are made by ministers in the Welsh Government, Scottish Government, and Northern Ireland Executive. All NICE guidance is subject to regular review and may be updated or withdrawn.

Copyright

ISBN: 978-1-4731-6572-4

Contents

Sample WinBUGS code

Genitourinary symptoms (normal likelihood, identity link function, analysis using SMD)

```
# Normal likelihood, identity link: SMD with arm-based means (External SDs for
standardising)

# Random effects model for multi-arm trials

# Fixed class effects

model{                              # *** PROGRAM STARTS

for(i in 1:ns){                     #   LOOP THROUGH STUDIES

  w[i,1] <- 0   # adjustment for multi-arm trials is zero for control arm

  delta[i,1] <- 0                   # treatment effect is zero for control arm

  mu[i] ~ dnorm(0,.0001)            # vague priors for all trial baselines

 }

# (1) CFB DATA

for(i in 1:ns){

  # calculate pooled.sd and adjustment for SMD

  df[i] <- sum(n[i,1:na[i]]) - na[i] # denominator for pooled.var

  #Pooled.var[i] <- sum(nvar[i,1:na[i]])/df[i]

  #Pooled.sd[i] <- sqrt(Pooled.var[i])

  H[i] <- 1 - 3/(4*df[i]-1)         # use Hedges' g

  #H[i] <- 1                        # use Cohen's d (ie no adjustment)

  for (k in 1:na[i]){

    se[i,k] <- sdCFB[i,k]/sqrt(n[i,k])

    var[i,k] <- pow(se[i,k],2)      # calcultate variances

    prec[i,k] <- 1/var[i,k]         # set precisions

    yCFB[i,k] ~ dnorm(phi[i,k], prec[i,k]) # normal likelihood

    #phi[i,k] <- theta[i,k] * (Pooled.sd[i]/H[i]) # theta is stand mean

        phi[i,k] <- theta[i,k] * (scalesd[i]/H[i]) # theta is stand mean

    theta[i,k] <- mu[i] + delta[i,k] # model for linear predictor, delta is SMD

    dev[i,k] <- (yCFB[i,k]-phi[i,k])*(yCFB[i,k]-phi[i,k])*prec[i,k]

    #nvar[i,k] <- (n[i,k]-1) * pow(sdCFB[i,k],2) # for pooled.sd
```

```
   }
  # summed residual deviance contribution for this trial
  resdev[i] <- sum(dev[i,1:na[i]])
 }

# RE MODEL (CFB data)
for(i in 1:ns){                    # LOOP THROUGH STUDIES WITH CFB DATA
  for (k in 2:na[i]){              # LOOP THROUGH ARMS
    # trial-specific RE distributions
    delta[i,k] ~ dnorm(md[i,k], taud[i,k])
    md[i,k] <- d[t[i,k]] - d[t[i,1]] + sw[i,k]
    # precision of RE distributions (with multi-arm trial correction)
    taud[i,k] <- tau *2*(k-1)/k
    #adjustment, multi-arm RCTs
    w[i,k] <- delta[i,k] - d[t[i,k]] + d[t[i,1]]
    # cumulative adjustment for multi-arm trials
    sw[i,k] <-sum(w[i,1:k-1])/(k-1)
   }
 }

totresdev <- sum(resdev[])             # Total Residual Deviance (all data)

# Priors and model assumptions (classes)
d[1] <- 0                  # treatment effect is zero for reference treatment
# treatment effects from Class
for (k in 2:nt) {
  d[k] <- m[D[k]]
}

m[1] <- 0
# prior for mean class effect
```

```
for (k in 2:nc){  m[k] ~ dnorm(0, .0001)  }


sd ~ dunif(0,4)                    # vague prior for between-trial SD

tau <- pow(sd,-2)                    # between-trial precision


# all pairwise differences

for (c in 1:(nt-1)) {  for (k in (c+1):nt)  { diff[c,k] <- d[k] - d[c] }  }


# pairwise SMDs for all possible class comparisons

for (c in 1:(nc-1)){

  for (k in (c+1):nc){ diffClass[c,k] <- (m[k]-m[c]) }

 }


# treatments of interest to rank

for (k in 1:ntR) {

            dR[k] <- d[tRcode[k]]

}

# classes of interest to rank

for (k in 1:ncR) {

            mR[k] <- m[cRcode[k]]

}


for (k in 1:ntR){

#  rk2[k] <- ntR+1-rank(dR[],k)            # lower values are "bad"

  rk[k] <- rank(dR[],k)         # lower values are "good"

  best[k]  <- equals(rk[k],1)  # Smallest is best (i.e. rank 1)

  # prob treat k is h-th best, prob[1,k]=best[k]

  for (h in 1:ntR) { prob[h,k] <- equals(rk[k],h) }

 }

#

for (k in 1:ncR){
  rkClass[k]  <- rank(mR[],k)   # lower values are "good"
  bestClass[k]  <- equals(rkClass[k],1)    # Smallest is best (i.e. rank 1)
  # prob class k is h-th best, prob[1,k]=best[k]
  for (h in 1:ncR) { probClass[h,k] <- equals(rkClass[k],h) }
 }
}                              # *** PROGRAM ENDS
```

Discontinuation due to adverse events (Binomial likelihood, Logit link function)

```
model{
for(i in 1:ns){
  w[i,1] <- 0                                    # adjustment for multi-arm trials is zero for control arm
  delta[i,1] <- 0    # treatment effect is zero for control arm
  mu[i] ~ dnorm(0,.0001) # vague priors for all trial baselines
  for (k in 1:na[i]) {
    r[i,k] ~ dbin(p[i,k],n[i,k])                                            # binomial likelihood
    logit(p[i,k]) <- mu[i] + delta[i,k]                   # model for linear predictor
    rhat[i,k] <- p[i,k] * n[i,k]                                            # expected value of the
numerators
#Deviance contribution
    dev[i,k] <- 2 * (r[i,k] * (log(r[i,k])-log(rhat[i,k]))
        +  (n[i,k]-r[i,k]) * (log(n[i,k]-r[i,k]) - log(n[i,k]-rhat[i,k])))
   }
#Summed residual deviance contribution for this trial
  resdev[i] <- sum(dev[i,1:na[i]])
  for (k in 2:na[i]) {
    delta[i,k] ~ dnorm(md[i,k],taud[i,k])             # trial-specific LOR distributions
    # mean of LOR distributions (with multi-arm trial correction)
    md[i,k] <-  d[t[i,k]] - d[t[i,1]] + sw[i,k]
    # precision of LOR distributions (with multi-arm trial correction)
    taud[i,k] <- tau *2*(k-1)/k
    # adjustment for multi-arm RCTs
    w[i,k] <- (delta[i,k] - d[t[i,k]] + d[t[i,1]])
    # cumulative adjustment for multi-arm trials
    sw[i,k] <- sum(w[i,1:(k-1)])/(k-1)
   }
 }
totresdev <- sum(resdev[])          # Total Residual Deviance
d[1]<-0      # treatment effect is zero for reference treatment

for (k in 2:nt) {
  d[k] <- m[D[k]]
}

m[1] <- 0
# prior for mean class effect
for (k in 2:nc){  m[k] ~ dnorm(0, .0001)  }

sd ~ dunif(0,5)     # vague prior for between-trial SD
tau <- pow(sd,-2)   # between-trial precision = (1/between-trial variance)
# pairwise ORs and LORs for all possible pair-wise comparisons
for (c in 1:(nt-1)){
  for (k in (c+1):nt){
    or[c,k] <- exp(d[k] - d[c])
    lor[c,k] <- (d[k]-d[c])
   }
 }
# treatments of interest to rank
for (k in 1:ntR) {
            dR[k] <- d[tRcode[k]]
}
# classes of interest to rank
```

```
for (k in 1:ncR) {
            mR[k] <- m[cRcode[k]]
}
for (k in 1:ntR){
  rk[k] <- rank(dR[],k)                       # assumes events are “bad”
  best[k] <- equals(rk[k],1)                                       #calculate probability that treat k is best
  # calculates probability that treat k is h-th best
  for (h in 1:ntR){ prob[h,k] <- equals(rk[k],h) }
 }
#
# pairwise differences for classes
for (c in 1:(nc-1)){
  for (k in (c+1):nc){
    lorClass[c,k] <- m[k] - m[c]
    orClass[c,k] <- exp(m[k] - m[c])
   }
 }
# rank classes
for (k in 1:ncR)  {
    rkClass[k]  <- rank(mR[],k)
    bestClass[k]  <- equals(rkClass[k],1)    # Smallest is best (i.e. rank 1)
# prob class k is h-th best, prob[1,k]=best[k]
    for (h in 1:ncR) { probClass[h,k] <- equals(rkClass[k],h) }
    }
}
```