National Institute for Health and Care Excellence

Menopause (update)

[B1] TSU NMA software code

NICE guideline number tbc

Supplement 4

November 2023

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 All NICE guidance is subject to regular review and may be updated or withdrawn.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:

Contents

1   Sample WinBUGS code

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

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

6   # Random effects model for multi-arm trials

7   # Fixed class effects

```
8    model{                   # *** PROGRAM STARTS

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

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

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

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

13    }

14   # (1) CFB DATA

15   for(i in 1:ns){

16    # calculate pooled.sd and adjustment for SMD

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

32      }
```

```
1      # summed residual deviance contribution for this trial

2     resdev[i] <- sum(dev[i,1:na[i]])

3    }

4

5    # RE MODEL (CFB data)

6    for(i in 1:ns){              # LOOP THROUGH STUDIES WITH CFB DATA

7     for (k in 2:na[i]){          # LOOP THROUGH ARMS

8       # trial-specific RE distributions

9       delta[i,k] ~ dnorm(md[i,k], taud[i,k])

10      md[i,k] <- d[t[i,k]] - d[t[i,1]] + sw[i,k]

11      # precision of RE distributions (with multi-arm trial correction)

12      taud[i,k] <- tau *2*(k-1)/k

13      #adjustment, multi-arm RCTs

14      w[i,k] <- delta[i,k] - d[t[i,k]] + d[t[i,1]]

15      # cumulative adjustment for multi-arm trials

16      sw[i,k] <-sum(w[i,1:k-1])/(k-1)

17     }

18    }

19

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

21

22    # Priors and model assumptions (classes)

23    d[1] <- 0              # treatment effect is zero for reference treatment

24    # treatment effects from Class

25    for (k in 2:nt) {

26      d[k] <- m[D[k]]

27    }

28

29    m[1] <- 0

30    # prior for mean class effect

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

32
```

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

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

3

4    # all pairwise differences

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

6

7    # pairwise SMDs for all possible class comparisons

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

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

10    }

11

12   # treatments of interest to rank

13   for (k in 1:ntR) {

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

15   }

16   # classes of interest to rank

17   for (k in 1:ncR) {

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

19   }

20

21   for (k in 1:ntR){

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

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

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

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

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

27    }

28   #

29   for (k in 1:ncR){

30     rkClass[k]  <- rank(mR[],k)   # lower values are "good"

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

32     # prob class k is h-th best, prob[1,k]=best[k]
```

1   for (h in 1:ncR) { probClass[h,k] <- equals(rkClass[k],h) }

2   }

3   }   # *** PROGRAM ENDS

4

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

7   model{

8   for(i in 1:ns){

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

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

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

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

13   r[i,k] ~ dbin(p[i,k],n[i,k])   # binomial likelihood

14   logit(p[i,k]) <- mu[i] + delta[i,k]   # model for linear predictor

15   rhat[i,k] <- p[i,k] * n[i,k]   # expected value of the
16   numerators

17   #Deviance contribution

18   dev[i,k] <- 2 * (r[i,k] * (log(r[i,k])-log(rhat[i,k]))

19   + (n[i,k]-r[i,k]) * (log(n[i,k]-r[i,k]) - log(n[i,k]-rhat[i,k])))

20   }

21   #Summed residual deviance contribution for this trial

22   resdev[i] <- sum(dev[i,1:na[i]])

23   for (k in 2:na[i]) {

24   delta[i,k] ~ dnorm(md[i,k],taud[i,k])   # trial-specific LOR distributions

25   # mean of LOR distributions (with multi-arm trial correction)

26   md[i,k] <- d[t[i,k]] - d[t[i,1]] + sw[i,k]

27   # precision of LOR distributions (with multi-arm trial correction)

28   taud[i,k] <- tau *2*(k-1)/k

29   # adjustment for multi-arm RCTs

30   w[i,k] <- (delta[i,k] - d[t[i,k]] + d[t[i,1]])

31   # cumulative adjustment for multi-arm trials

32   sw[i,k] <- sum(w[i,1:(k-1)])/(k-1)

```
1        }

2      }

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

4      d[1]<-0      # treatment effect is zero for reference treatment

5

6      for (k in 2:nt) {

7        d[k] <- m[D[k]]

8      }

9

10     m[1] <- 0

11     # prior for mean class effect

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

13

14     sd ~ dunif(0,5)     # vague prior for between-trial SD

15     tau <- pow(sd,-2)   # between-trial precision = (1/between-trial variance)

16     # pairwise ORs and LORs for all possible pair-wise comparisons

17     for (c in 1:(nt-1)){

18       for (k in (c+1):nt{

19         or[c,k] <- exp(d[k] - d[c])

20         lor[c,k] <- (d[k]-d[c])

21       }

22     }

23     # treatments of interest to rank

24     for (k in 1:ntR) {

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

26     }

27     # classes of interest to rank

28     for (k in 1:ncR) {

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

30     }

31     for (k in 1:ntR){

32       rk[k] <- rank(dR[],k)              # assumes events are "bad"
```

```
1      best[k] <- equals(rk[k],1)                              #calculate probability that treat k is best
2      # calculates probability that treat k is h-th best
3      for (h in 1:ntR){ prob[h,k] <- equals(rk[k],h) }
4     }
5     #
6     # pairwise differences for classes
7     for (c in 1:(nc-1)){
8       for (k in (c+1):nc){
9         lorClass[c,k] <- m[k] - m[c]
10        orClass[c,k] <- exp(m[k] - m[c])
11      }
12     }
13    # rank classes
14    for (k in 1:ncR)  {
15       rkClass[k]  <- rank(mR[],k)
16       bestClass[k]  <- equals(rkClass[k],1)    # Smallest is best (i.e. rank 1)
17    # prob class k is h-th best, prob[1,k]=best[k]
18       for (h in 1:ncR) { probClass[h,k] <- equals(rkClass[k],h) }
19       }
20    }
21
22
```