

```

model{
# T = probability of true eGFR belonging to each category
#
# All categories
for (i in 1:ny){ # loop through studies with all categories
  y[i,1:4] ~ dmulti(T[,], N[i])
  # calculate residual deviance
  for (m in 1:4){ # loop through all reported thresholds
    yhat[i,m] <- T[m] * N[i] # predicted number events
    y1[i,m] <- max(y[i,m], 0.1) # correction for zero cell
    yhat1[i,m] <- max(yhat[i,m], 0.1) # correction for p=0
    # Deviance contribution when non-zero cell (allows p=0)
    ydvl[i,m] <- 2*y1[i,m]*(log(y1[i,m])-log(yhat1[i,m]))
    # Calculate deviance contribution, when zero cell=zero
    ydv[i,m] <- ydvl[i,m]*(1-equals(y[i,m],0))
  }
  yresdev[i] <- sum(ydv[i,]) # summed residual deviance contribution for this
study
}
totresdevT <- sum(yresdev[]) # Total Residual Deviance
T[1:4] ~ ddirch(omega[]) # prior distribution for T (WinBUGS compatible)
for (j in 1:4){ # loop through all categories
  omega[j] <- 1 # Dirichlet parameter (non-inf)
}
# type A data: 0-30; 30-60; >60
for (i in (ny+1):(ny+nyA)){ # loop through studies with type A data
  y[i,1:3] ~ dmulti(TA[,], N[i])
  # calculate residual deviance
  for (m in 1:3){ # loop through all reported thresholds
    yhat[i,m] <- TA[m] * N[i] # predicted number events
    y1[i,m] <- max(y[i,m], 0.1) # correction for zero cell
    yhat1[i,m] <- max(yhat[i,m], 0.1) # correction for p=0
    # Deviance contribution when non-zero cell (allows p=0)
    ydvl[i,m] <- 2*y1[i,m]*(log(y1[i,m])-log(yhat1[i,m]))
    # Calculate deviance contribution, when zero cell=zero
    ydv[i,m] <- ydvl[i,m]*(1-equals(y[i,m],0))
  }
  yresdev[i] <- sum(ydv[i,1:3]) # summed residual deviance contribution for this
study
}
# link probabilities
TA[1] <- T[1] # type A: true < 30
TA[2] <- T[2] + T[3] # type A: 30 < true < 60
TA[3] <- T[4] # type A: true > 60
#
# type C data: 0-30; 30-45; >45
for (i in (ny+nyA+1):(ny+nyA+nyC)){ # loop through studies with type C data
  y[i,1:3] ~ dmulti(TC[,], N[i])
  # calculate residual deviance
  for (m in 1:3){ # loop through all reported thresholds
    yhat[i,m] <- TC[m] * N[i] # predicted number events
    y1[i,m] <- max(y[i,m], 0.1) # correction for zero cell
    yhat1[i,m] <- max(yhat[i,m], 0.1) # correction for p=0
    # Deviance contribution when non-zero cell (allows p=0)
    ydvl[i,m] <- 2*y1[i,m]*(log(y1[i,m])-log(yhat1[i,m]))
    # Calculate deviance contribution, when zero cell=zero
    ydv[i,m] <- ydvl[i,m]*(1-equals(y[i,m],0))
  }
  yresdev[i] <- sum(ydv[i,1:3]) # summed residual deviance contribution for this
study
}
# link probabilities
TC[1] <- T[1] # type C: true < 30
TC[2] <- T[2] # type C: 30 < true < 45
TC[3] <- T[3] + T[4] # type C: true > 45
#
# type E data: 0-60; >60

```

```

for (i in (ny+nyA+nyC+1):(ny+nyA+nyC+nyE)){ # loop through studies with type E
data
  y[i,1] ~ dbin(TE[1], N[i])
  # Deviance contribution
  yhat[i,1] <- TE[1] * N[i] # expected value of the numerators
  y1[i,1] <- max(y[i,1], 0.1) # correction for zero cell
  yhat1[i,1] <- max(yhat[i,1], 0.1) # correction for p=0
  # Deviance contribution when non-zero cell (allows p=0)
  ydev1[i,1] <- 2 * (y1[i,1]*(log(y1[i,1])-log(yhat1[i,1])))
    + (N[i]-y1[i,1])*(log(N[i]-y1[i,1])
    - log(N[i]-yhat1[i,1])))
  # Deviance contribution when zero cell (allows p=0)
  ydev0[i,1] <- 2*N[i]*log(N[i]/(N[i]-yhat[i,1]))
  # Calculate deviance contribution
  yresdev[i] <- ydev1[i,1]*(1-equals(y[i,1],0)) + ydev0[i,1]*equals(y[i,1],0)
}
# link probabilities
TE[1] <- T[1] + T[2] + T[3] # type E: true < 60
TE[2] <- T[4] # type E: true > 60
#
# type F data: 0-30; >30
for (i in (ny+nyA+nyC+nyE+1):(ny+nyA+nyC+nyE+nyF)){ # loop through studies with
type F data
  y[i,1] ~ dbin(TF[1], N[i])
  # Deviance contribution
  yhat[i,1] <- TF[1] * N[i] # expected value of the numerators
  y1[i,1] <- max(y[i,1], 0.1) # correction for zero cell
  yhat1[i,1] <- max(yhat[i,1], 0.1) # correction for p=0
  # Deviance contribution when non-zero cell (allows p=0)
  ydev1[i,1] <- 2 * (y1[i,1]*(log(y1[i,1])-log(yhat1[i,1])))
    + (N[i]-y1[i,1])*(log(N[i]-y1[i,1])
    - log(N[i]-yhat1[i,1])))
  # Deviance contribution when zero cell (allows p=0)
  ydev0[i,1] <- 2*N[i]*log(N[i]/(N[i]-yhat[i,1]))
  # Calculate deviance contribution
  yresdev[i] <- ydev1[i,1]*(1-equals(y[i,1],0)) + ydev0[i,1]*equals(y[i,1],0)
}
# link probabilities
TF[1] <- T[1] # type F: true < 30
TF[2] <- T[2] + T[3] + T[4] # type F: true > 30
#
# p[j,m]: probability of being in true category j and POC category m
#
# All categories
for (i in 1:ns){ # loop through studies with all categories
  for (j in 1:4){ # loop through all categories
    r[i,j,1:4] ~ dmulti(p[j,], n[i,j])
    # calculate residual deviance
    for (m in 1:4){ # loop through all reported thresholds
      # predicted number events
      rhat[i,j,m] <- p[j,m] * n[i,j]
      r1[i,j,m] <- max(r[i,j,m], 0.1) # correction for zero cell
      rhat1[i,j,m] <- max(rhat[i,j,m], 0.1) # correction for p=0
      # Deviance contribution when non-zero cell (allows p=0)
      dvl[i,j,m] <- 2*r1[i,j,m]*(log(r1[i,j,m])-log(rhat1[i,j,m]))
      # Calculate deviance contribution, when zero cell=zero
      dv[i,j,m] <- dvl[i,j,m]*(1-equals(r[i,j,m],0))
    }
    dv[i,j,m] <- 2*r[i,j,m]*(log(r[i,j,m])-log(rhat[i,j,m]))
  }
  dev[i,j] <- sum(dv[i,j,])
}
# summed residual deviance contribution for this trial
resdev[i] <- sum(dev[i,1:4])
}
totresdev <- sum(resdev[]) # Total Residual Deviance
for (j in 1:4){ # loop through all categories
  p[j,1:4] ~ ddirch(alpha[]) # prior distribution for p (WinBUGS compatible)
  alpha[j] <- 1 # Dirichlet parameter (non-inf)
}

```

```

}
#
# type E data: 0-60; >60
for (i in (ns+1):(ns+nsE)){ # loop through studies with type E data
  for (j in 1:2){ # loop through all categories
    r[i,j,1] ~ dbin(pE[j,1], n[i,j])
    # Deviance contribution
    rhat[i,j,1] <- pE[j,1] * n[i,j] # expected value of the numerators
    r1[i,j,1] <- max(r[i,j,1], 0.1) # correction for zero cell
    rhat1[i,j,1] <- max(rhat[i,j,1], 0.1) # correction for p=0
    # Deviance contribution when non-zero cell (allows p=0)
    dev1[i,j,1] <- 2 * (r1[i,j,1]*(log(r1[i,j,1])-log(rhat1[i,j,1]))
      + (n[i,j]-r1[i,j,1])*(log(n[i,j]-r1[i,j,1])
        - log(n[i,j]-rhat1[i,j,1])))
    # Deviance contribution when zero cell (allows p=0)
    dev0[i,j,1] <- 2*n[i,j]*log(n[i,j]/(n[i,j]-rhat[i,j,1]))
    # Calculate deviance contribution
    dev[i,j] <- dev1[i,j,1]*(1-equals(r[i,j,1],0)) + dev0[i,j,1]*equals(r[i,j,1],0)
  }
  # summed residual deviance contribution for this study
  resdev[i] <- sum(dev[i,1:2])
}
for (j in 1:2){ # loop through all categories
  pE[j,2] <- 1-pE[j,1]
}
# link probabilities
# type E: true < 60
sumE <- T[1]+T[2]+T[3]
pE[1,1] <- (p[1,1]+p[1,2]+p[1,3])*T[1]/sumE + (p[2,1]+p[2,2]+p[2,3])*T[2]/sumE
+ (p[3,1]+p[3,2]+p[3,3])*T[3]/sumE # POC <60
# type E: true > 60
pE[2,1] <- p[4,1]+p[4,2]+p[4,3] # POC <60
#
# type A2 data: 0-30; 30-60; >60
for (i in (ns+nsE+1):(ns+nsE+nsA2)){ # loop through studies with type A2 data
  for (j in 1:3){ # loop through all categories
    r[i,j,1] ~ dbin(pA2[j,1], n[i,j])
    # Deviance contribution
    rhat[i,j,1] <- pA2[j,1] * n[i,j] # expected value of the numerators
    r1[i,j,1] <- max(r[i,j,1], 0.1) # correction for zero cell
    rhat1[i,j,1] <- max(rhat[i,j,1], 0.1) # correction for p=0
    # Deviance contribution when non-zero cell (allows p=0)
    dev1[i,j,1] <- 2 * (r1[i,j,1]*(log(r1[i,j,1])-log(rhat1[i,j,1]))
      + (n[i,j]-r1[i,j,1])*(log(n[i,j]-r1[i,j,1])
        - log(n[i,j]-rhat1[i,j,1])))
    # Deviance contribution when zero cell (allows p=0)
    dev0[i,j,1] <- 2*n[i,j]*log(n[i,j]/(n[i,j]-rhat[i,j,1]))
    # Calculate deviance contribution
    dev[i,j] <- dev1[i,j,1]*(1-equals(r[i,j,1],0)) + dev0[i,j,1]*equals(r[i,j,1],0)
  }
  # summed residual deviance contribution for this study
  resdev[i] <- sum(dev[i,1:3])
}
for (j in 1:3){ # loop through all categories
  pA2[j,2] <- 1-pA2[j,1]
}
# link probabilities
# type A2: true < 30
pA2[1,1] <- p[1,1] # POC <30
# type A2: true 30-60
pA2[2,1] <- pA[2,1] # POC >30
# probability for type A data
sumA <- T[2]+T[3]
pA[2,1] <- p[2,1]*T[2]/sumA + p[3,1]*T[3]/sumA # POC <30
# type A2: true > 60
pA2[3,1] <- p[4,1] + p[4,2] + p[4,3] # POC >60
}

```