Appendix

In this appendix, we first take the reader through the BUGS program that we used to carry out the *N* / *N* analysis reported in Table 2. We then discuss the key details of the BUGS program that we used to conduct analyses under *t* assumptions at levels 1 and 2, with the degrees of freedom parameters ν_1 and ν_2 fixed at various values (see Tables 2 and 3). We then show how we expanded this program in order to conduct analyses in which ν_1 was treated as an unknown parameter.

Program A

Rectangular data files or S-plus format input files can be read into BUGS. In our analyses, we used S-plus format input files. The BUGS command '**data in "enactive.dat"**' in Program A indicates that our data are contained in a file named **enactive.dat** (see below).

We begin by defining the constants **N** and **I** in a '**const**' statement; **N=35** is the total number of level-1 observations in our sample, and **I=9** is the total number of level-2 units (children). In the '**var**' statement, we define the vectors of values contained in **enactive.dat** (i.e., **CHILD[N], Y[N], AGE[N], MSPAC[I]**). **CHILD** is a vector that contains N=35 elements. This vector of values links each of the level-1 observations in our sample with the appropriate level-2 unit (i.e., child). Thus, elements 1-4 are equal to a value of 1, elements 5-8 are equal to a value of 2, and so on (i.e., **CHILD=c(1, 1, 1, 1, 2, 2, 2, 2, . . . 9, 9, 9)**). The vector **Y** contains the 35 *EG* values for the children in our sample. Thus, elements 1-4 contain the time series of *EG* values for child 1, elements 5-8 contain the series of *EG* values for child 2, etc. The 35-element vector **AGE** contains the age values (specifically, AGE_{ti} - 24) corresponding to the *EG* time series data (see Equation 1 in the body of the chapter). Thus, elements 1-4 contain the values -12, -8, -4 and 0, respectively; elements 5-8 contain the values -12, -8, -4 and 0, and so on. Lastly, **MSPAC** is a vector that contains 9 elements; the first element contains the *MSPAC* value for child 1 centered around the grand mean for *MSPAC*, the second element contains the grand-mean centered *MSPAC* value for child 2, etc.

In the '**var'** statement, we also define scalar and vector quantities corresponding to the various parameters in our model. Thus, **beta00**, **beta10** and **beta11** are scalars corresponding to the fixed effects in our model (see Equation 2). **U1** is a vector consisting of 9 elements; the first element corresponds to the random effect for child 1 (i.e., $U_{1(1)}$), the second element corresponds to the random effect for child 2 ($U_{1(2)}$), etc. **sig2** and **tau11** correspond to the level-1 and level-2 variance components in our model (i.e., σ^2 and τ_{11}).

Note that we have also defined the scalars **sig2inv** and **tau11inv**. These correspond to the level-1 and level-2 precisions $1 / \sigma^2$ and $1 / \tau_{11}$, respectively. We do this because when specifying normality assumptions in BUGS, one must specify precisions rather than variances (see below).

Starting values for the fixed effects and the level-1 and level-2 precsion parameters are contained in the file '**enactive.in**', which is specified in the '**inits in**' statement in our program. Note that one can also specify seed values in the initial values file. We often find it useful to employ estimates for the fixed effects obtained via programs such as HLM as starting values; in addition, one can invert REML estimates of the variance components to obtain starting values for level-1 and level-2 precision parameters in our models (see Endnote 2). As discussed in Endnote 2, it is important to re-run the Gibbs sampler using a range of starting values.

The observations (Y_{ti}) in the current model are assumed independent and normally distributed with expected value μ_{ti} and variance σ^2 (Equation 4), where μ_{ti} is defined as in Equation 5. This portion of our model is specified in lines a-e of our BUGS program. Each of the N=35 level-1 observations in our sample (**Y[n]**) is assumed normally distributed with expected value **mu[n]** and precision **sig2inv**: **Y[n] ~ dnorm(mu[n], sig2inv)**. As can be seen, **mu[n]** is defined in a manner similar to μ_{ti} in Equation 5. Note that the symbol '←' means "is to be replaced by" (Spiegelhalter et al., 1996a, p. 16). Also note that **mu** has been defined in our '**var**' statement.

Nested indexing was used in lines c and d of our program in order to reference the appropriate elements of **MSPAC** and **U1**. For example, when **n** is equal to values of 1, 2, 3 or 4, **CHILD[n]** is equal to a value of 1, and so BUGS will reference the first element of **MSPAC** (i.e., **MSPAC[1]**). Similarly, when **n** is equal to values of 5, 6, 7, or 8, **CHILD[n]** is equal to a value of 2, and BUGS will reference the second element of **MSPAC**.

The prior for the random effects is specified in lines f-h. As can be seen, the random effects (*i*=(1, ….. , 9)) are assumed normally distributed with mean 0 and precision **tau11inv**: **U1[i] ~ dnorm(0, tau11inv)**.

We place diffuse priors on the fixed effects. For example, for **beta00** we specify a normal prior with mean **0** and precision **1.0E-5**. Extremely low precisions correspond to low degrees of prior information. This implies that in drawing inferences concerning the fixed effects, the data will dominate the prior.

Finally, we place mildly informative priors on the precision parameters **sig2inv** and **tau11inv**. These priors are discussed in Endnote 3. We have found priors of this kind to provide good coverage properties for fixed effects in small-sample HM settings.

Program B

We used Program B to conduct analyses under *t* distributional assumptions at levels 1 and 2, with the degrees of freedom parameters ν_1 and ν_2 fixed at particular values. In the current example, both ν_1 and ν_2 are fixed at values of 4.

In the '**var**' statement, we now include the vectors **w[N]**, **winvsig2[N]**, **q[I]**, and **qinvtau[I]**. Note that **w[N]** is a vector whose elements correspond to the level-1 weight parameters in our model (i.e., the ω_{ti}); **winvsig2[N]** is a vector whose elements correspond to the level-1 precision parameters ω_{ti} / σ^2 (see below); **q[I]** is a vector whose elements correspond to the level-2 weight parameters (i.e., the q_i); and **qinvtau[I]** is a vector whose elements correspond to the level-2 precision parameters q_i / τ_{11}.

As noted in the body of our chapter, in the scale mixture of normals representation of the *t*, the assumption that the Y_{ti} are normally distributed with expected value μ_{ti} and variance σ^2, is replaced by the assumption: $N(\mu_{ti}, \sigma^2 / \omega_{ti})$ (Equation 6), where the ω_{ti} — the level-1 weight parameters — are assumed Gamma distributed (i.e., $Gamma(\nu_1 / 2, \nu_1 / 2)$; see Equation 7).

As discussed above, normality assumptions in BUGS are parameterized in terms of precisions rather than variances. Thus in line a, we specify that the level-1 observations are normally distributed with mean **mu[n]** and precision **winvsig2[n]**, where **winvsig2[n]** corresponds to ω_{ti} / σ^2.

In line b, we specify that the ω_{ti} (i.e., **w[n]**, n=(1,….,35)) are gamma distributed. Note that with $\nu_1 = 4$, we have: **w[n] ~ dgamma(2.0, 2.0)**. Note also that in line c we form the precision parameters ω_{ti} / σ^2.

Similar logic applies in employing the scale mixture of normals representation of the *t* at level 2. The random effects are assumed normally distributed with mean 0 and precision q_i / τ_{11} (i.e., **qinvtau[i]**). With $\nu_2 = 4$, the q_i are assumed Gamma distributed as follows: **q[i] ~ dgamma(2.0, 2.0)**. In line d, we form the precision parameters q_i / τ_{11}.

Program C

In treating ν_1 as an unknown, we let $1 / \nu_1$ take on the following values: .01, .02, .03, …. ,.31, .32, .33. As pointed out in the body of our chapter, a value of $(1 / \nu_1) = .01$ corresponds to a value of $\nu_1 = 100$, and a value of $(1 / \nu_1) = .33$ corresponds to a value of $\nu_1 \approx 3$. We define a vector **priorinu** which consists of 33 bins (**NBINS=33**) (see lines a and b). The first bin, as will be seen, corresponds to a value for $(1 / \nu_1)$ of 1 / 100 = .01; the second bin corresponds to a value of 2 / 100 = .02; finally, the last bin corresponds to a value of 33 / 100 = .33. Under Prior I (see p. 19), we assign equal prior probability to each bin; as can be seen in lines d and e, each bin is assigned a prior probability of 1 / 33 (i.e., **1/NBINS**). In line f, the variable **k** is defined as a discrete variable that can take on values 1, 2, …, 33; the probability associated with each value of **k** is given by the probability in the corresponding element of the vector **priorinu**. In lines g-i, we see how the values of **k** translate into values for 1 / ν_1 (i.e., **nu1inv ← k / 100**); ν_1 (i.e., **nu1 ← 1 / nu1inv**); and ν_1 / 2 (i.e., **nu1half ← nu1 / 2.0**). Finally, as can be seen in line c, the prior distribution for the level-1 weight parameters depends upon the current value for ν_1 / 2 (i.e., **nu1half**); as we cycle through the algorithm, we will tend to condition on those values that are most likely given the data and our prior assumptions.

```
model A;  # Program A:  Example 2, N/N Analysis
const
 I=9, # number of children
 N=35; # number of obs in sample
var
 CHILD[N],Y[N],AGE[N],MSPAC[I],
 beta00,beta10,beta11,U1[I],
 sig2inv,sig2,tau11inv,tau11,mu[N];

data  in "enactive.dat";
inits in "enactive.in";

{
 for (n in 1:N) {                                                              #line a
   Y[n] ~ dnorm(mu[n],sig2inv);                                                #line b
    mu[n] <- beta00  +  ( beta10 + ( beta11*MSPAC[CHILD[n]] ) +                #line c
                U1[CHILD[n]]  ) * AGE[n];                                      #line d
   }                                                                           #line e

 # prior for the random effects:
   for (i in 1:I) {                                                            #line f
   U1[i] ~dnorm(0,tau11inv);                                                   #line g
 }                                                                             #line h
 # priors for the fixed effects :
    beta00 ~ dnorm(0,1.0E-5);
    beta10 ~ dnorm(0,1.0E-5);
    beta11 ~ dnorm(0,1.0E-5);

 # prior for level-1 precision parameter:
    sig2inv  ~ dgamma(1.5, 362);                                               #line i
    sig2 <- 1/sig2inv;


 # prior for level-2 precision parameter:
    tau11inv ~ dgamma(1.5, 0.91);                                              #line j
    tau11 <- 1/tau11inv;

 }
```

Structure of ‘enactive.dat’:

```
list(CHILD=c( 1, 1, 1, 1, 2, 2, 2, 2, . . . , 9, 9, 9),
      Y=c(72.2, 33.3, 27.6, 10.1, 93.2,79.0, 34.3, 17.2, . . . , 86.2, 55.0, 20.0),
      AGE=c(-12, -8, -4, 0, -12, -8, -4, 0, . . . , -12, -8, 0),
      MSPAC=c(2.711, 13.011, . . . , -12.889))
```

Structure of ‘enactive.in’:

```
list(beta00=14.10, beta10=-5.32, beta11=-0.057,
      sig2inv=0.0069, tau11inv=2.78, seed=111864)
```

```
model B;  # Program B:  Example 2, t4 / t4 Analysis
const
 I=9, # number of children
 N=35; # number of obs in sample
var
 CHILD[N],Y[N],AGE[N],MSPAC[I],
 beta00,beta10,beta11,U1[I],w[N],winvsig2[N],
 q[I],qinvtau[I],sig2inv,sig2,tau11inv,tau11,mu[N];

data  in "enactive.dat";
inits in "enactivb.in";

{
  for (n in 1:N) {
   Y[n] ~ dnorm(mu[n],winvsig2[n]);                              #line a
   mu[n] <- beta00 + ( beta10  +  (  beta11 * MSPAC[CHILD[n]]  )  +
            U1[CHILD[n]]  )  * AGE[n];

# prior for level-1 weight parameters:
   w[n] ~ dgamma(2.0,2.0);                                       #line b
   winvsig2[n] <- w[n]*sig2inv;                                  #line c
}

 # priors for random effects and level-2 weight parameters:
 for (i in 1:I) {
   U1[i] ~ dnorm(0,qinvtau[i]);
   q[i]  ~ dgamma(2.0,2.0);
   qinvtau[i] <- q[i]*tau11inv;                                  #line d
}

 # priors for fixed effects :
   beta00 ~ dnorm(0,1.0E-5);
   beta10 ~ dnorm(0,1.0E-5);
   beta11 ~ dnorm(0,1.0E-5);

 # prior for level-1 precision parameter:
   sig2inv  ~ dgamma(1.5, 270);                                  #line e
   sig2 <- 1/sig2inv;

 # prior for level-2 precision parameter:
   tau11inv ~ dgamma(1.5, 0.685);                                #line f
   tau11 <- 1/tau11inv;
}
```

```
model C;  # Program C:  Example 2, treating the level-1 df as unknown, and
const        #     fixing the level-2 df at a value of 4
 I=9, # number of children
 N=35, # number of obs in sample
 NBINS=33;                                                              #line a
var
 CHILD[N],Y[N],AGE[N],MSPAC[I],
 beta00,beta10,beta11,U1[I],w[N],winvsig2[N],
 q[I],qinvtau[I], sig2inv,sig2,tau11inv,tau11,mu[N],
  priorinu[NBINS],k,nu1inv,nu1,nu1half;                                 #line b

data  in "enactive.dat";
inits in "enactivc.in";
{
  for (n in 1:N) {
    Y[n] ~ dnorm(mu[n],winvsig2[n]);
    mu[n] <- beta00 + ( beta10  +  (  beta11 * MSPAC[CHILD[n]]  )  +
              U1[CHILD[n]]  )  * AGE[n];

# prior for level-1 weight parameters:
   w[n] ~ dgamma(nu1half, nu1half);                                     #line c
   winvsig2[n] <- w[n]*sig2inv;
}

 # prior for random effects and level-2 weight parameters:
  for (i in 1:I) {
  U1[i] ~ dnorm(0,qinvtau[i]);
  q[i]  ~ dgamma(2.0, 2.0);
  qinvtau[i] <- q[i]*tau11inv;
}
  # priors for fixed effects :
    beta00 ~ dnorm(0,1.0E-5);
    beta10 ~ dnorm(0,1.0E-5);
    beta11 ~ dnorm(0,1.0E-5);

 # prior for level-1 precision parameter:
   sig2inv  ~ dgamma(1.5, 308.0);
   sig2 <- 1/sig2inv;

  # prior for level-2 precision parameter:
    tau11inv ~ dgamma(1.5, 0.685);
    tau11 <- 1/tau11inv;

 for (m in 1:NBINS) {                                                   #line d
  priorinu[m] <- 1/NBINS;                                               #line e
}

 k ~ dcat(priorinu[]);                                                  #line f
 nu1inv <- k / 100;                                                     #line g
 nu1 <- 1/nu1inv;                                                       #line h
 nu1half <- nu1/2.0;                                                    #line i
}
```