



UvA-DARE (Digital Academic Repository)

Bayesian explorations in mathematical psychology

Matzke, D.

Publication date

2014

Document Version

Final published version

[Link to publication](#)

Citation for published version (APA):

Matzke, D. (2014). *Bayesian explorations in mathematical psychology*. [Thesis, fully internal, Universiteit van Amsterdam].

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

Part V

Appendices

Appendix to Chapter 2: “Psychological Interpretation of the Ex–Gaussian and Shifted Wald Parameters: A Diffusion Model Analysis”

A.1 The Distribution of the Diffusion Model Parameter Values

Figure A.1 presents histograms of the best-fitting diffusion model parameter values and the corresponding z/a and s_z/a ratios found in 23 applications of the diffusion model. The histograms are based on the the parameter values reported for each experimental condition of the 23 articles. The exact parameter values, including references, are available as supplemental materials at <http://dora.erbe-matzke.com/publications.html>.

A.2 Results for the Diffusion Model Trial-to-Trial Variability Parameters

This appendix shows how the ex–Gaussian and shifted Wald parameters change as a function of the manipulation of the trial-to-trial variability in drift rate η , the trial-to-trial variability in starting point s_z , and the trial-to-trial variability in nondecision time s_t parameters of the diffusion model. Table A.1 gives a summary of the associations between the ex–Gaussian and shifted Wald parameter and the diffusion model variability parameters. Figure A.2 and Figure A.3 then show the detailed changes in the ex–Gaussian and shifted Wald parameters as a function of changes in the diffusion model variability parameters.

Ex–Gaussian Parameters

With respect to trial-to-trial variability in drift rate η , Figure A.2a shows that both μ and σ decrease as η increases. In contrast, τ increases for low values of η and decreases for high values of η . However, the changes in the three ex–Gaussian parameters are all extremely small. Turning to trial-to-trial variability in starting point s_z , Figure A.2b shows that both σ and τ increase as s_z increases, but in contrast, μ decreases with increasing s_z . However, the changes in the three

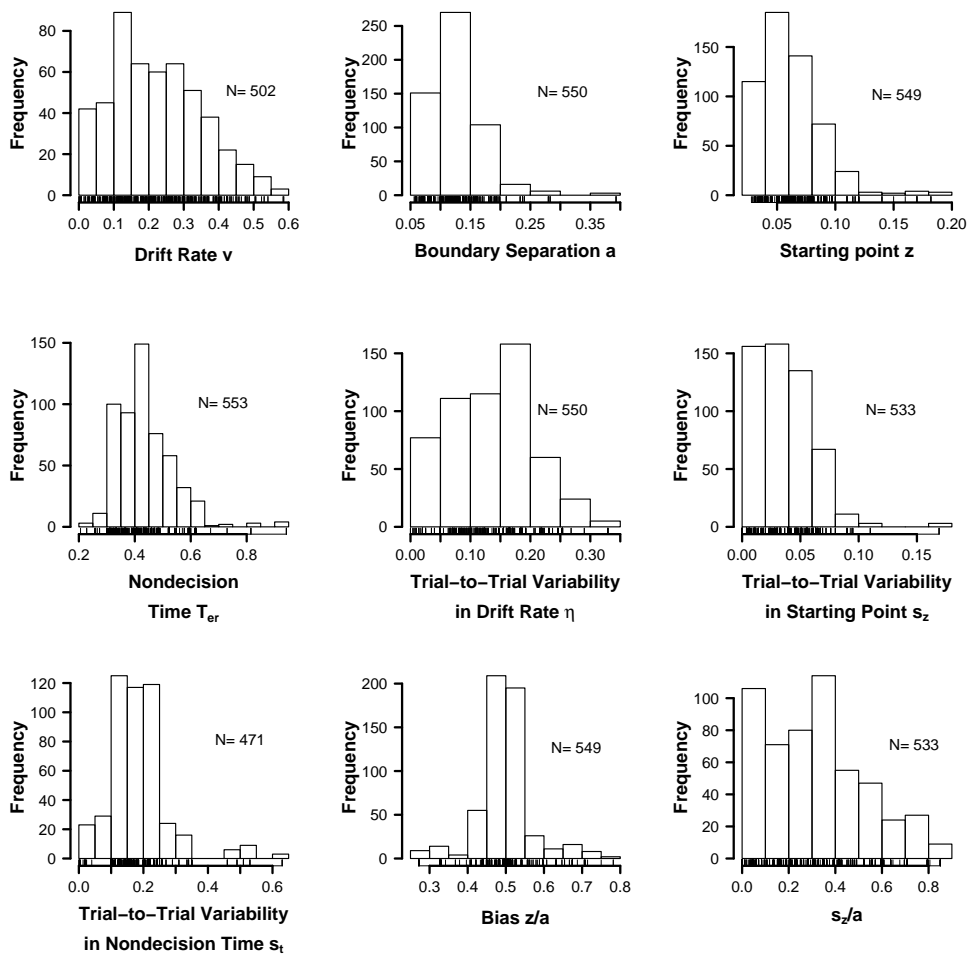


Figure A.1 Histograms of the diffusion model parameter values.

ex-Gaussian parameters are all negligible. With respect to trial-to-trial variability in nondecision time s_t , Figure A.2c shows that both σ and τ parameters as s_t increases, whereas μ decreases with increasing s_t . Note that σ is the only parameter that is substantially influenced by s_t . In fact, σ changes substantially more as function of s_t than as a function of any other diffusion model parameter.

To summarize, these results further support the conclusion that the two most important parameters of the ex-Gaussian distribution, μ and τ , do not correspond uniquely to parameters of the diffusion model. Neither of these ex-Gaussian parameters is influenced substantially by any of the variability parameters of the diffusion model. In contrast, σ seems to be uniquely associated with s_t , the parameter for trial-to-trial variability in nondecision time.

Shifted Wald Parameters

With respect to trial-to-trial variability in drift rate η , Figure A.3a shows that both α and γ decrease as η increases, but in contrast, θ increases with increasing η . However, the changes in the three shifted Wald parameters are all extremely small. Turning to trial-to-trial variability in

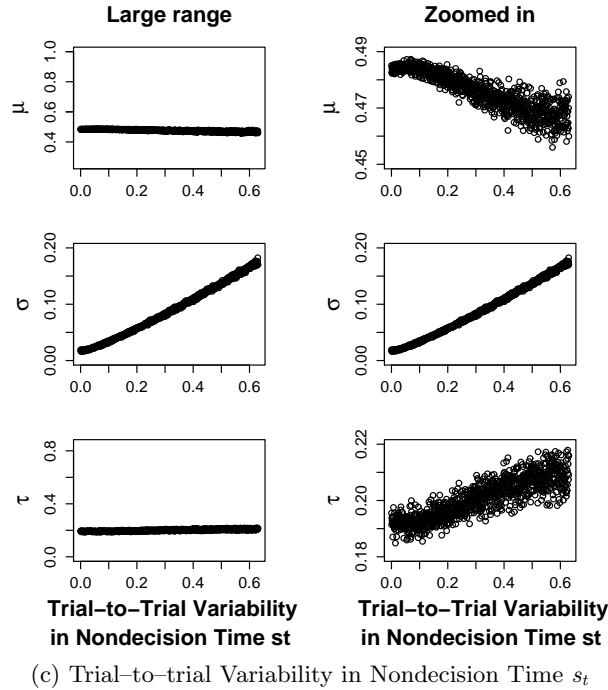
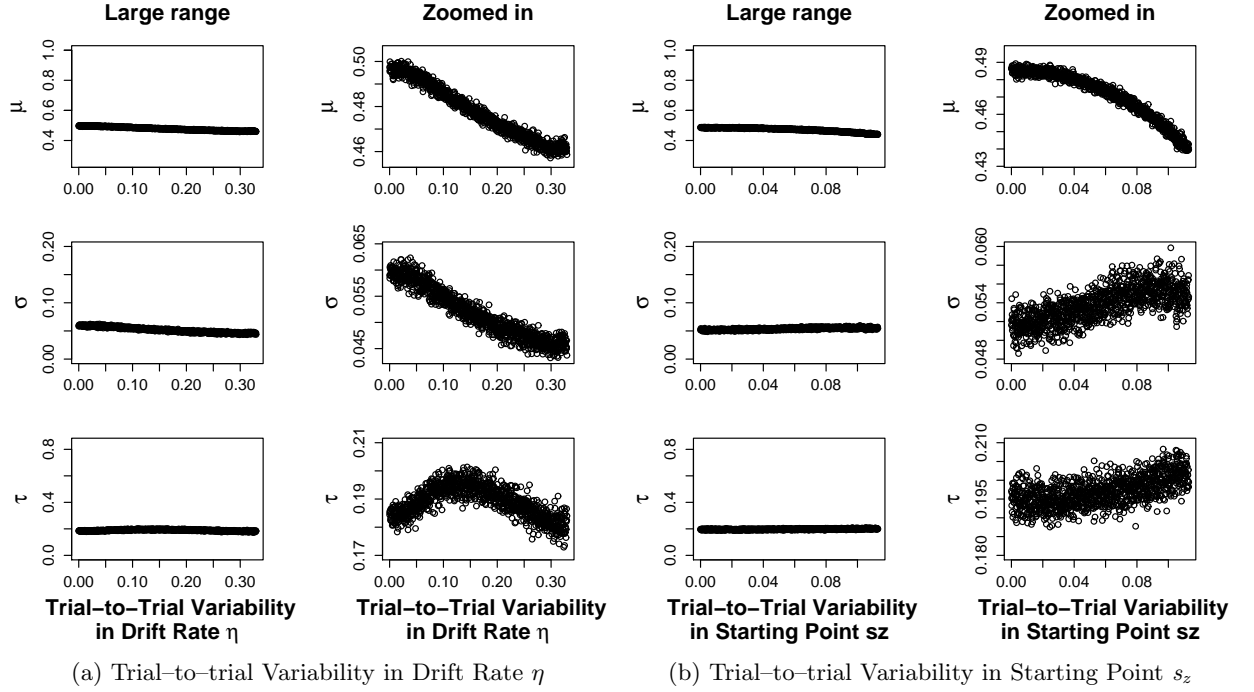


Figure A.2 Changes in the ex-Gaussian parameters μ , σ , and τ as a function of systematic changes in the diffusion model parameters trial-to-trial variability in drift rate η (panel a), trial-to-trial variability in starting point s_z (panel b), and trial-to-trial variability in nondecision time s_t (panel c). The left-hand figures in each panel plot the results on scales ranging from the minimum to the maximum values of the ex-Gaussian parameters found across all simulations. The right-hand figures in each panel plot the same results on scales ranging from the minimum to the maximum values of the ex-Gaussian parameters found for the manipulation of the given diffusion model parameter.

A. APPENDIX TO CHAPTER 2: “PSYCHOLOGICAL INTERPRETATION OF THE EX-GAUSSIAN AND SHIFTED WALD PARAMETERS: A DIFFUSION MODEL ANALYSIS”

Table A.1 The Associations Between the Parameters of the Ex-Gaussian and Shifted Wald Distributions and the Variability Parameters of the Diffusion Model.

		Diffusion model parameters		
		η	s_z	s_t
Ex-Gaussian	μ	-	-	-
	σ	-	+	++
	τ	+/-	+	+
Shifted Wald	α	-	+	++/- -
	θ	+	-	- -
	γ	-	×	+/-

Note. ++, substantial positive association; +, weak positive association; - -, substantial negative association; -, weak negative association; ×, no association; η , variability in drift rate; s_z , variability in starting point; s_t , variability in nondecision time.

starting point s_z , Figure A.3b shows that γ is unaffected by changes in s_z , whereas α increases and θ decreases with increasing s_z . However, the changes in both α and θ are extremely small. With respect to trial-to-trial variability in nondecision time s_t , Figure A.3c shows that both α and γ increase for low and intermediate values of s_t and then decrease for high values. In contrast, θ decreases for low and intermediate values of s_t and equals 0 for high values. Although α changes more than either θ or γ , the change in θ is also substantial. Note that α changes just as much as a function of s_t than as a function of boundary separation a .

To summarize, these results further support the conclusion that the shifted Wald parameters do not correspond uniquely to parameters of the diffusion model. The γ parameter is not influenced substantially by any of the variability parameters of the diffusion model. In contrast, both α and θ are substantially influenced by s_t , the trial-to-trial variability in nondecision time. In addition to the influence of the key diffusion model parameters, changes in α and θ can therefore also reflect the influence of s_t .

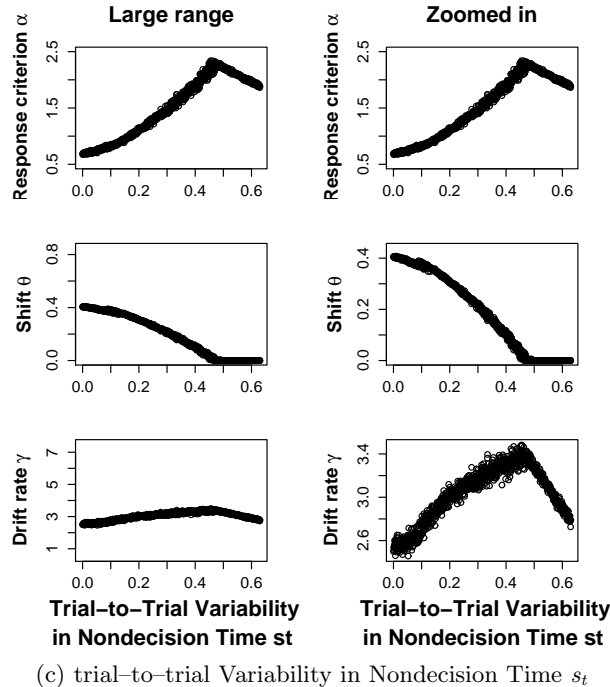
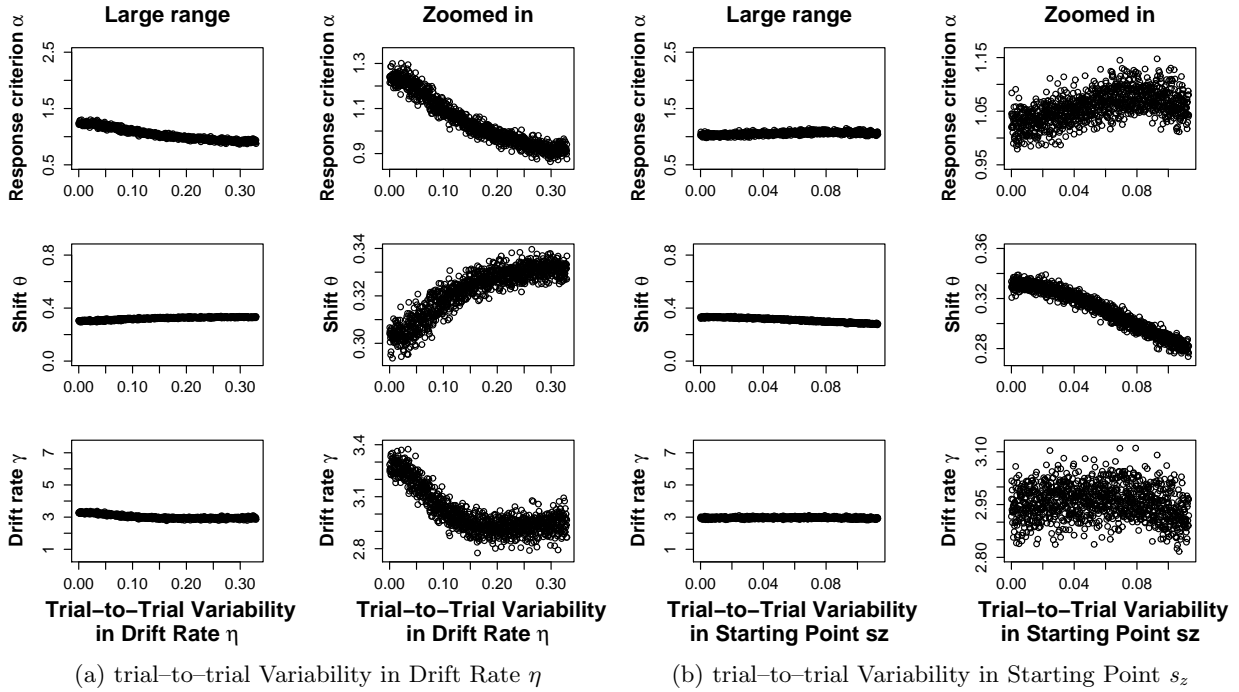


Figure A.3 Changes in the shifted Wald parameters α , θ , and γ as a function of systematic changes in the diffusion model parameters trial-to-trial variability in drift rate η (panel a), trial-to-trial variability in starting point s_z (panel b), and trial-to-trial variability in nondecision time s_t (panel c). The left-hand figures in each panel plot the results on scales ranging from the minimum to the maximum values of the shifted Wald parameters found across all simulations. The right-hand figures in each panel plot the same results on scales ranging from the minimum to the maximum values of the shifted Wald parameters found for the manipulation of the given diffusion model parameter.

Appendix to Chapter 3: “Bayesian Parametric Estimation of Stop-Signal Reaction Time Distributions”

B.1 WinBUGS Script

Individual Bayesian Parametric Approach (BPA) Model

The WinBUGS script for the individual BPA is as follows:

```
model
{
  # Priors for parameters
  mu_go ~ dunif(1,1000)
  sigma_go ~ dunif(1,300)
  tau_go ~ dunif(1,300)

  mu_stop ~ dunif(1,600)
  sigma_stop ~ dunif(1,250)
  tau_stop ~ dunif(1,250)

  # Go RTs come from an ex-Gaussian distribution
  for (g in 1:n.gort){
    go_rt[g] ~ ExGaussian(mu_go, sigma_go, tau_go)
  }

  # Signal respond trials; signal-respond RTs (srirt) at each SSD come from
  # a censored ex-Gaussian distribution
  # (see first part of Equation 14.)
  for (d in 1:end_SR){
    for (r in 1:n.srirt[d]){
      srirt[d,r] ~ CensoredExGaussian_SR(mu_go, sigma_go, tau_go,
                                         mu_stop, sigma_stop, tau_stop, ssd_SR[d])
    }
  }

  # Signal inhibit trials; Successful inhibitions come from a censored ex-Gaussian
  # distribution (see second part of Equation 14.)
  for (h in 1:end_I){
```

B. APPENDIX TO CHAPTER 3: “BAYESIAN PARAMETRIC ESTIMATION OF STOP-SIGNAL REACTION TIME DISTRIBUTIONS”

```
for (i in 1:n.inhibitions[h]){
  zeros[h,i] <- 0
  zeros[h,i] ~ dpois(phi[h,i])
  phi[h,i] <- - intg[h]
}
# Compute integral in Equation 14 using Simpson's rule of numerical integration
# The first and the second arguments define the limits of integration,
# and the third argument defines the number of subintervals used for
# computing the integral.
intg[h] <- CensoredExGaussian_I(1, 6000, 2000, mu_go, sigma_go, tau_go,
                               mu_stop, sigma_stop, tau_stop, ssd_I[h])
}
```

The `ExGaussian` and `CensoredExGaussian_SR` distributions and the `CensoredExGaussian_I` function are implemented with the WinBUGS Development Interface (WBDev; Lunn, 2003). For a WBDev tutorial for social scientists, see Wetzels, Lee, and Wagenmakers (2010). The WinBUGS and WBDev scripts are available in the supplemental materials at <http://dora.erbe-matzke.com/publications.html>. For computational reasons, the indefinite integral in Equation 3.14 is replaced by a definite integral (i.e., `CensoredExGaussian_I`) with limits of integration well beyond the range of stop-signal reaction times that may be encountered in the stop-signal paradigm.

Hierarchical Bayesian Parametric Approach (BPA) Model

The WinBUGS script for the hierarchical BPA is as follows:

```
model
{
  # Priors for the group-level parameters
  # The I(0,) construct denotes distributional censoring, with a lower bound of 0,
  # and an upper bound of infinity
  mu_mu_go ~ dnorm(500,0.0001)I(0,)
  lambda_mu_go <- 1/pow(sigma_mu_go,2)
  sigma_mu_go ~ dunif(0,300)

  mu_sigma_go ~ dnorm(100,0.001)I(0,)
  lambda_sigma_go <- 1/pow(sigma_sigma_go,2)
  sigma_sigma_go ~ dunif(0,200)

  mu_tau_go ~ dnorm(80,0.001)I(0,)
  lambda_tau_go <- 1/pow(sigma_tau_go,2)
  sigma_tau_go ~ dunif(0,200)

  mu_mu_stop ~ dnorm(200,0.0001)I(0,)
  lambda_mu_stop <- 1/pow(sigma_mu_stop,2)
  sigma_mu_stop ~ dunif(0,200)

  mu_sigma_stop ~ dnorm(40,0.001)I(0,)
  lambda_sigma_stop <- 1/pow(sigma_sigma_stop,2)
  sigma_sigma_stop ~ dunif(0,100)

  mu_tau_stop ~ dnorm(30,0.001)I(0,)
  lambda_tau_stop <- 1/pow(sigma_tau_stop ,2)
  sigma_tau_stop ~ dunif(0,100)

  # C has to be large enough so that all phi[s,k,n] are positive
  # C <- 10000
}
```

```

# Participant loop
for (j in 1:n.subjects){

  # Go RTs come from an ex-Gaussian distribution
  for (g in 1:n.gort){
    go_rt[g,j] ~ ExGaussian(mu_go[j], sigma_go[j], tau_go[j])
  }

  # Signal respond trials; signal-respond RTs (srst) at each SSD come from
  # a censored ex-Gaussian distribution
  # (see first part of Equation 14.)
  for (d in 1:end_SR[j]){
    for (r in 1:n.srst[d,j]){
      srst[d,r,j] ~ CensoredExGaussian_SR(mu_go[j], sigma_go[j], tau_go[j],
      mu_stop[j], sigma_stop[j], tau_stop[j], ssd_SR[d,j])
    }
  }

  # Signal inhibit trials; Successful inhibitions come from a censored ex-Gaussian
  # distribution (see second part of Equation 14.)
  # The following code implements the zeros trick (see WinBUGS manual)
  # Because phi[s,k,n] is a Poisson mean, it should always be positive.
  # As a result, we may need to add constant C to ensure that all phi[s,k,n]
  # are positive
  for (h in 1:end_I[j]){
    for (i in 1:n.inhibitions[h,j]){
      zeros[h,i,j] <- 0
      zeros[h,i,j] ~ dpois(phi[h,i,j])
      phi[h,i,j] <- - intg[h,j] #+C
    }

    # Compute integral in Equation 14 using Simpson's rule of numerical
    # integration. The first and the second arguments define the limits
    # of integration, and the third argument defines the number of
    # subintervals used for computing the integral.
    intg[h,j] <- CensoredExGaussian_I(1, 3000, 1000, mu_go[j], sigma_go[j],
    tau_go[j], mu_stop[j], sigma_stop[j],
    tau_stop[j], ssd_I[h,j])
  }

  # Individual parameters come from truncated normal distributions
  # The third argument specifies the truncation point
  mu_go[j] ~ TruncatedNormal(mu_mu_go, lambda_mu_go,0)
  sigma_go[j] ~ TruncatedNormal(mu_sigma_go,lambda_sigma_go,1)
  tau_go[j] ~ TruncatedNormal(mu_tau_go,lambda_tau_go,1)

  mu_stop[j] ~ TruncatedNormal(mu_mu_stop, lambda_mu_stop,0)
  sigma_stop[j] ~ TruncatedNormal(mu_sigma_stop,lambda_sigma_stop,1)
  tau_stop[j] ~ TruncatedNormal(mu_tau_stop,lambda_tau_stop,1)
}
}

```

The TruncatedNormal distribution is implemented with WBDev and is available in the supplemental materials.

Appendix to Chapter 4: “Release the BEESTS: Bayesian Estimation of Ex-Gaussian Stop-Signal Reaction Time Distributions”

C.1 Prior Distribution of the Model Parameters

This appendix presents the prior distributions of the model parameters in the BEESTS implementation of the Bayesian parametric approach (BPA). The name of each parameter as shown in the BEESTS output is in brackets.

Individual BPA

The priors for the go and stop parameters are uniform distributions, spanning a plausible but wide range of values. BEESTS relies on slightly more diffuse priors than the WinBUGS implementation of the BPA (see Matzke et al., 2013):

$$\begin{aligned}
 \mu_{go} \text{ (mu_go)} &\sim \text{Uniform}(0.001, 1000) \\
 \sigma_{go} \text{ (sigma_go)} &\sim \text{Uniform}(1, 500) \\
 \tau_{go} \text{ (tau_go)} &\sim \text{Uniform}(1, 500) \\
 \mu_{stop} \text{ (mu_stop)} &\sim \text{Uniform}(0.001, 600) \\
 \sigma_{stop} \text{ (sigma_stop)} &\sim \text{Uniform}(1, 350) \\
 \tau_{stop} \text{ (tau_stop)} &\sim \text{Uniform}(1, 350).
 \end{aligned}
 \tag{C.1}$$

Hierarchical BPA

Individual Parameters

The hierarchical BPA assumes that the μ_{go} , σ_{go} , τ_{go} , μ_{stop} , σ_{stop} , and τ_{stop} parameters of each participant $j = 1, \dots, J$ come from truncated normal group-level distributions. The group-level distributions are themselves characterized by a group mean (μ) and a group standard deviation (σ) parameter. The WinBUGS implementation relies on normal group-level distributions that are

truncated only at the lower end, whereas BEESTS uses normal distributions that are truncated at the lower *and* the upper ends:

$$\begin{aligned}
 \mu_{go_j} \text{ (mu_go.subj)} &\sim \text{Normal}(\mu_{\mu_{go}}, \sigma_{\mu_{go}})[0.001, 1000] \\
 \sigma_{go_j} \text{ (sigma_go.subj)} &\sim \text{Normal}(\mu_{\sigma_{go}}, \sigma_{\sigma_{go}})[1, 500] \\
 \tau_{go_j} \text{ (tau_go.subj)} &\sim \text{Normal}(\mu_{\tau_{go}}, \sigma_{\tau_{go}})[1, 500] \\
 \mu_{stop_j} \text{ (mu_stop.subj)} &\sim \text{Normal}(\mu_{\mu_{stop}}, \sigma_{\mu_{stop}})[0.001, 600] \\
 \sigma_{stop_j} \text{ (sigma_stop.subj)} &\sim \text{Normal}(\mu_{\sigma_{stop}}, \sigma_{\sigma_{stop}})[1, 350] \\
 \tau_{stop_j} \text{ (tau_stop.subj)} &\sim \text{Normal}(\mu_{\tau_{stop}}, \sigma_{\tau_{stop}})[1, 350].
 \end{aligned} \tag{C.2}$$

Group-Level Parameters

The priors for the group mean and group standard deviations are uniform distributions. Note that the WinBUGS implementation uses censored normal priors for the group-level means and relies on slightly less diffuse priors for the group-level standard deviations than BEESTS:

$$\begin{aligned}
 \mu_{\mu_{go}} \text{ (mu_go)} &\sim \text{Uniform}(0.001, 1000) \\
 \sigma_{\mu_{go}} \text{ (mu_go_var)} &\sim \text{Uniform}(0.01, 300) \\
 \mu_{\sigma_{go}} \text{ (sigma_go)} &\sim \text{Uniform}(1, 500) \\
 \sigma_{\sigma_{go}} \text{ (sigma_go_var)} &\sim \text{Uniform}(0.01, 200) \\
 \mu_{\tau_{go}} \text{ (tau_go)} &\sim \text{Uniform}(1, 500) \\
 \sigma_{\tau_{go}} \text{ (tau_go_var)} &\sim \text{Uniform}(0.01, 200) \\
 \mu_{\mu_{stop}} \text{ (mu_stop)} &\sim \text{Uniform}(0.001, 600) \\
 \sigma_{\mu_{stop}} \text{ (mu_stop_var)} &\sim \text{Uniform}(0.01, 300) \\
 \mu_{\sigma_{stop}} \text{ (sigma_stop)} &\sim \text{Uniform}(1, 350) \\
 \sigma_{\sigma_{stop}} \text{ (sigma_stop_var)} &\sim \text{Uniform}(0.01, 200) \\
 \mu_{\tau_{stop}} \text{ (tau_stop)} &\sim \text{Uniform}(1, 350) \\
 \sigma_{\tau_{stop}} \text{ (tau_stop_var)} &\sim \text{Uniform}(0.01, 200).
 \end{aligned} \tag{C.3}$$

Appendix to Chapter 7: “The Issue of Power in the Identification of ‘ g ’ with Lower-Order Factors”

D.1 R Code for Power Calculations

This appendix presents the R code that can be used to calculate power for various sample sizes. The code takes as inputs the goodness-of-fit statistic of M_A (TA), the chosen Type I error probability (α), df_{diff} (df), the sample size (N) used to obtain the non-centrality parameter λ , and the minimum (minN) and maximum (maxN) sample sizes of interest. The output provided by the code consists of the power coefficients corresponding to sample sizes ranging from the minimum and the maximum sample size of interest.

```
#Goodness-of-fit statistic of M_A (i.e., non-centrality parameter lambda)
TA = 7.23
#Type I error probability
alpha = 0.05*2
#Degrees of freedom (i.e., df.diff)
df = 1
#Sample size used to calculate the non-centrality parameter lambda
N = 200
#Critical value
C = qchisq(alpha, df=df, ncp=0, lower.tail=F)
#Minimum sample size of interest
minN = 100
#Maximum sample size of interest
maxN = 2000
power = matrix(0,maxN-minN+1,2)

#Nnew is new sample size of interest
for (Nnew in minN:maxN){
  #lambda.new is the value of the non-centrality parameter
  #corresponding to the new sample size
  lambda.new = (TA/N)*Nnew
  #calculate power
  power[Nnew-minN+1,1] = pchisq(C, df=df, ncp=lambda.new, lower.tail=F)
  power[Nnew-minN+1,2] = Nnew
}
```

D. APPENDIX TO CHAPTER 7: "THE ISSUE OF POWER IN THE IDENTIFICATION OF 'g' WITH LOWER-ORDER FACTORS"

```
}  
#power plot  
plot(power[,2],power[,1],type='l',xlab="Sample size", ylab="Power")  
#power for original N  
print(power[power[,2]==N,1])
```

Appendix to Chapter 8: “Accounting for Measurement Error and the Attenuation of Correlation: A Bayesian Approach”

E.1 WinBUGS Script

The WinBUGS script that implements the Bayesian correction for the attenuation of the correlation is as follows (see Chapter 8 and Behseta et al., 2009 for details):

```
# Bayesian correction for the attenuation of correlation as a results of uncertainty in measurement
model {
  # Data
  for (i in 1:N){
    # eta[i,1] = theta[i]; eta[i,2] = beta[i];
    # mu[1] = mu_theta; mu[2] = mu_beta;
    # ISigma_cov = inverse of Sigma_cov matrix
    eta[i,1:2] ~ dnorm(mu[],ISigma_cov[,])

    # observed[i,1] = theta_hat[i]; observed[i,2] = beta_hat[i];
    # Isigma_epsilon[i,1] = Inverse of sigma_epsilon^2_theta[i];
    # Isigma_epsilon[i,2] = Inverse of sigma_epsilon^2_beta[i]
    for (j in 1:2){
      observed[i,j] ~ dnorm(eta[i,j],Isigma_epsilon[i,j])
    }
  }

  # Priors
  mu[1] ~ dnorm(0,.001)
  mu[2] ~ dnorm(0,.001)

  # sigma[1] = sigma_theta; sigma[2] = sigma_beta
  sigma[1] ~ dunif(0,mysigma_1)
  sigma[2] ~ dunif(0,mysigma_2)

  rho ~ dunif(-1,1)

  # Reparameterization
  Sigma_cov[1,1] <- pow(sigma[1],2)
  Sigma_cov[1,2] <- rho*sigma[1]*sigma[2]
  Sigma_cov[2,1] <- rho*sigma[1]*sigma[2]
```

E. APPENDIX TO CHAPTER 8: “ACCOUNTING FOR MEASUREMENT ERROR AND THE ATTENUATION OF CORRELATION: A BAYESIAN APPROACH”

```
Sigma_cov[2,2] <- pow(sigma[2],2)
ISigma_cov[1:2,1:2] <- inverse(Sigma_cov[1:2,1:2])
}
```

The R script that calls the WinBUGS script using the R2WinBUGS (Sturtz, Ligges, & Gelman, 2005) package is available in the supplemental materials at <http://dora.erbe-matzke.com/publications.html>. The R script allows users to adjust the range of the uniform prior distribution of `sigma[1]` and `sigma[2]` by specifying the value of `my_sigma1` and `my_sigma2`.

Appendix to Chapter 9: “A Default Bayesian Hypothesis Test for Mediation”

F.1 JAGS Code

JAGS Code for Correlation

```
##### Cauchy-prior on alpha #####
model
{
  for (i in 1:n)
  {
    mu[i] <- intercept + alpha*x[i]
    y[i] ~ dnorm(mu[i],phi)
  }

  # uninformative prior on intercept,
  # Jeffreys' prior on precision phi
  intercept ~ dnorm(0,.0001)
  phi ~ dgamma(.0001,.0001)

  # inverse-gamma prior on g:
  g <- 1/invg
  a.gamma <- 1/2
  b.gamma <- n/2
  invg ~ dgamma(a.gamma,b.gamma)

  # g-prior on beta:
  vari <- (g/phi) * invSigma
  prec <- 1/vari
  alpha ~ dnorm(0, prec)
}

# Explanation-----
# Prior on g:
# We know that  $g \sim \text{inverse\_gamma}(1/2, n/2)$ , with 1/2 the shape
# parameter and n/2 the scale parameter.
# It follows that  $1/g \sim \text{gamma}(1/2, 2/n)$ .
# However, BUGS/JAGS uses the *rate parameterization* 1/theta instead of the
```

```
# scale parametrization theta. Hence we obtain, in de BUGS/JAGS rate notation:
# 1/g ~ dgamma(1/2, n/2)
#-----
```

JAGS Code for Partial Correlation

```
##### Cauchy-prior on beta and tau' #####

# theta contains beta and tau'

model
{
  for (i in 1:n)
  {
    mu[i] <- intercept + theta[1]*x[i,1] + theta[2]*x[i,2]
    y[i] ~ dnorm(mu[i],phi)
  }

# uninformative prior on intercept,
# Jeffreys' prior on precision phi
intercept ~ dnorm(0,.0001)
phi ~ dgamma(.0001,.0001)

# inverse-gamma prior on g:
g <- 1/invg
a.gamma <- 1/2
b.gamma <- n/2
invg ~ dgamma(a.gamma,b.gamma)

# calculation of the inverse matrix of V
inverse.V <- inverse(V)
# calculation of the elements of prior precision matrix
for(i in 1:2)
{
  for (j in 1:2)
  {
    prior.T[i,j] <- inverse.V[i,j] * phi/g
  }
}
# multivariate prior for the theta vector
theta[1:2] ~ dnorm( mu.theta, prior.T )
for(i in 1:2) { mu.theta[i] <- 0 }

}
```

F.2 Testing the Correctness of Our JAGS Implementation

To assess the correctness of our JAGS implementation, we compared the analytical results for the two-sided Bayes factor against the Savage-Dickey density ratio results based on the MCMC samples from JAGS. The distribution that fit the posterior samples best¹ is the non-standardized

¹We compared the fit of four distributions: a non-standardized t-distribution, a normal distribution, a non-parametric distribution estimated with the spline interpolation function `splinefun` in R, and a non-parametric

t-distribution with the following density:

$$p(x|\nu, \mu, \sigma) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{(\pi\nu\sigma)}} \left(1 + \frac{1}{\nu} \left(\frac{x - \mu}{\sigma}\right)^2\right)^{-\frac{\nu+1}{2}}, \quad (\text{F.1})$$

with ν degrees of freedom, location parameter μ , and scale parameter σ . With the samples of the parameter of interest, we can estimate ν , μ , and σ and thus the exact shape of the distribution and the exact height of the distribution at the point of interest.

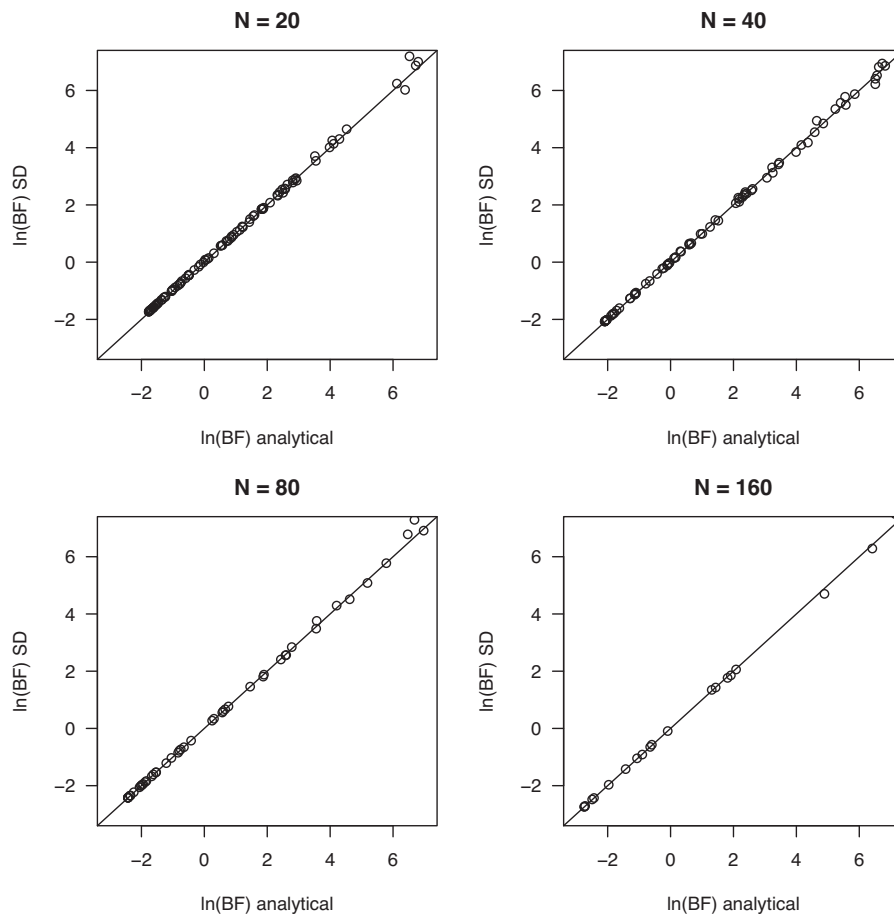


Figure F.1 Natural logarithm of the Bayes factors for correlation obtained with analytical calculations (x axis) or obtained with the SD method based on a non-standardized t-distribution (y axis) for different sample sizes (N). The graphs show fewer points as the samples grow larger, because in these situations there are more extreme Bayes factors that fall outside the axis limits. We restricted the graphs, since it is most important that the lower Bayes factors lie on the diagonal: it is not important whether a Bayes factor is 2000 or 3000, since it is overwhelming evidence in any case.

We checked the fit of this distribution and the performance of the SD method in a small simulation study. We considered the following sample sizes: $N = 20, 40, 80,$ or 160 . We simu-

distributions estimated with the R function `logspline` that also uses splines to estimate the log density. All four distributions fitted reasonably well: the Bayes factors of the analytical test and the SD method are similar with all different posterior distributions. All four distributions are therefore included in the R package `BayesMed` and can be used when applying the SD method.

lated correlational data by drawing N values for the X from a standard normal distribution, and conditional on X we simulated values for Y according to the following equation:

$$Y_i = \beta_0 + \tau X_i + \epsilon, \quad (\text{F.2})$$

where the subscript i denotes subject i and τ represents the relation between X and Y . For each of the four sample sizes, we generated 100 datasets, each in which τ was drawn from a standard uniform distribution.

Next, we tested the correlation in each dataset with both the analytical Bayesian correlation test and the SD method with the non-standardized t-distribution and compared the results. The results are shown in Figure F.1. The figure shows that the proposed SD method performs well: the Bayes factors of the analytical test and the SD method are similar for all sample sizes and correlations.