R SCRIPT

```
# various utility functions for parameter recovery routines


# density function for Wald (unshifted)
dwald = function(x,gamma,alpha){
  return((alpha/(sqrt(2*pi*x^3)))*exp(-(alpha-gamma*x)^2/(2*x)))
}

# function to generate random shifted Wald data
# adapted from pp. 79-80, Dagpunar, J. (1988). Principles of Random
Variate Generation. Clarendon Press, Oxford.
# code modified from Heathcote (2004)
rwald = function(n, gamma, alpha, theta) {
  y2 = rchisq(n,1)
  y2onm = y2/gamma
  u = runif(n)
  r1 = (2*alpha + y2onm - sqrt(y2onm*(4*alpha+y2onm)))/(2*gamma)
  r2 = (alpha/gamma)^2/r1
  ifelse(u < alpha/(alpha+gamma*r1), theta+r1, theta+r2)
}

# basic function that returns Root Mean Squared Error
rmse = function(error){
  sqrt(mean(error^2))
}

# negative log likelihood for shifted Wald
nll.wald = function(par,dat){
  return(-sum(log(dwald(dat-par[3], gamma=par[1], alpha=par[2]))))
}

# Start point estimate for SW, based on first two moments
# assumes s = p*min(x), where x is a data vector
# from Heathcote (2004)
waldinit = function(x, p = 0.9) {
  theta = p*min(x)
  x = x - theta
  gamma = sqrt(mean(x)/var(x))
  alpha = gamma*mean(x)
  return(c(gamma,alpha,theta))
}
#Running simulations for Fit Functions

# define simulation parameters

nSub = 20
nTrials = 20

# parent distribution means and sds (from F, Vick, Bowman, 2018)
G = 3.91 # gamma
G.sd = 0.70
A = 0.92 # alpha
A.sd = 0.17
H = 0.32 # theta
H.sd = 0.05

# each simulation run starts HERE
# build simulated RT matrix
# rows = subjects, columns = trials
rts = matrix(0, nrow=nSub, ncol=nTrials)

# build matrix to store "target" parameters
# rows = subjects, columns = gamma, alpha, theta
targets = matrix(0, nrow=nSub, ncol=3)

# randomly draw target shifted Wald parameters from parent distribution
# (one unique parameter value for each subject)
for (i in 1:nSub){
  targets[i,1] = rnorm(1, mean=G, sd=G.sd) # draw random gamma
  targets[i,2] = rnorm(1, mean=A, sd=A.sd) # draw random alpha
  targets[i,3] = rnorm(1, mean=H, sd=H.sd) # draw random theta
}

# from these target SW values, generate the distribution of observed RTs
for each subject

for (i in 1:nSub){
  rts[i,] = rwald(nTrials,
                  gamma = targets[i,1],
                  alpha = targets[i,2],
                  theta = targets[i,3])
}

# now let’s use CMLE to fit shifted Wald distribution to each row of
observed RTs
# first, we’ll store these estimates in a matrix
predictions = matrix(0, nrow=nSub, ncol=3)

# now do the fits (one per subject)
for (i in 1:nSub){
  fit = optim(waldinit(rts[i,]), nll.wald, dat=rts[i,])
  predictions[i,] = fit$par
}
```