# R Script

FIT FUNCTIONS 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]))))
}

# Hierach. negative log likelihood for shifted Wald
nll.wald.h = function(par,dat){
  n_subjects <- dim(dat)[1]
  n_trials <- dim(dat)[2]
 par <- exp(par)

gamma_group_mean  <- par[1]
gamma_group_sd  <- par[2]
alpha_group_mean  <- par[3]
```

```
alpha_group_sd <- par[4]
theta_group_mean <- par[5]
theta_group_sd <- par[6]
gamma_ind <- par[6 + 1:(n_subjects-1)]
alpha_ind <- par[6 + (n_subjects-1) + 1:(n_subjects-1)]
theta_ind <- par[6 + 2*(n_subjects-1) + 1:(n_subjects-1)]

gamma_ind <- c(gamma_ind, n_subjects*gamma_group_mean - sum(gamma_ind))
alpha_ind <- c(alpha_ind, n_subjects*alpha_group_mean - sum(alpha_ind))
theta_ind <- c(theta_ind, n_subjects*theta_group_mean - sum(theta_ind))

theta_mat<- matrix(rep(theta_ind, each=n_trials), n_subjects, n_trials, byrow=TRUE)
if(any(dat-theta_mat <0)) {return(Inf)}
nll <-  0
  for (s in 1:n_subjects){
    nll <- nll - sum(log(dwald(dat[s,]-theta_ind[s], gamma=gamma_ind[s],
alpha=alpha_ind[s])))
}

nll <- nll -sum(dnorm(gamma_ind, mean=gamma_group_mean, sd=gamma_group_sd), log=TRUE)
nll <- nll -sum(dnorm(alpha_ind, mean=alpha_group_mean, sd=alpha_group_sd), log=TRUE)
nll <- nll -sum(dnorm(theta_ind, mean=theta_group_mean, sd=theta_group_sd), log=TRUE)

  return(nll)
}
# 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))
}

waldinit.h = function(x, p = 0.9) {
  #input:
  #x: matrix of rt w rows indicating subject and columns indicating trial
 n_subjects  <- dim(x)[1]
 n_trials  <- dim(x)[2]
   theta_ind  <- p*apply(x,1, min) #vector of shift parameters
  theta_group_mean <- mean(theta_ind)   #group mean shift parameter
  theta_group_sd <- sd(theta_ind)   #SD of shift across group
  theta_mat<- matrix(rep(theta_ind, each=n_trials), n_subjects, n_trials, byrow=TRUE)
#diff shift per person > create matrix
  x = x - theta_mat


  gamma_ind = sqrt(apply(x, 1, mean) / apply(x,1, var)) #gamma vector
  gamma_group_mean <- mean(gamma_ind)
  gamma_group_sd <- sd(gamma_ind)
  alpha_ind <- gamma_ind *apply(x, 1, mean) #alpha vector
  alpha_group_mean <- mean(alpha_ind) #group mean alpha
  alpha_group_sd  <- sd(alpha_ind)

  #to make sure you pull the right parameter, we have to be careful; pull group first
  par <- c(gamma_group_mean, gamma_group_sd,
           alpha_group_mean, alpha_group_sd,
           theta_group_mean, theta_group_sd,
           gamma_ind[-n_subjects], alpha_ind[-n_subjects], theta_ind[-n_subjects])
  return(log(par))
}
```

CMLE R SCRIPT

```
#source("fitFunctions.R")
##############################
# 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
}

# now lets compare the predictions (fits) to the targets

# mean of parameter estimates
apply(predictions, 2, mean)
```

```
# RMSD
apply(predictions-targets, 2, rmse)

# mean bias
apply(predictions-targets, 2, mean)

# correlation
diag(cor(predictions,targets))
```

HMLE R SCRIPT

```
#source("fitFunctions.R")
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

#matrices
rmse_mat = matrix(NA, 200, 3+(3*nSub))
mb_mat = matrix(NA, 200, 3+(3*nSub))
ind_means_mat = matrix(NA, 200, (3*nSub))

for (x in 1:200){

  # 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)
  # mean of parameter estimates
  #apply(predictions, 2, mean)

  # RMSD
  #apply(predictions-targets, 2, rmse)

  # mean bias
  #apply(predictions-targets, 2, mean)

  # correlation
  #diag(cor(predictions,targets))


  #HIERARCHICAL MODEL FIT

  fit.h = optim(waldinit.h(rts), nll.wald.h, dat=rts)

  # now lets compare the predictions (fits) to the targets
  #will need to do
  par <- exp(fit.h$par)
```

```
gamma_group_mean  <- par[1]
gamma_group_sd  <- par[2]
alpha_group_mean  <- par[3]
alpha_group_sd <- par[4]
theta_group_mean <- par[5]
theta_group_sd <- par[6]
gamma_ind <- par[6 + 1:(nSub-1)]
alpha_ind <- par[6 + (nSub-1) + 1:(nSub-1)]
theta_ind <- par[6 + 2*(nSub-1) + 1:(nSub-1)]

gamma_ind <- c(gamma_ind, nSub*gamma_group_mean - sum(gamma_ind))
alpha_ind <- c(alpha_ind, nSub*alpha_group_mean - sum(alpha_ind))
theta_ind <- c(theta_ind, nSub*theta_group_mean - sum(theta_ind))

##RMSE
#g level error
gg_rmse = sqrt((G- gamma_group_mean)^2)
ga_rmse = sqrt((A- alpha_group_mean)^2)
gh_rmse = sqrt((H- theta_group_mean)^2)

#i level error
ig_rmse = sqrt((targets[,1]-gamma_ind)^2)
ia_rmse = sqrt((targets[,2]-alpha_ind)^2)
ih_rmse = sqrt((targets[,3]-theta_ind)^2)

rmse_mat[x,] = c(gg_rmse,ga_rmse, gh_rmse, ig_rmse, ia_rmse, ih_rmse)

##MEAN BIAS
#g level mb
gg_mb = -1*(G- gamma_group_mean)
ga_mb = -1*(A- alpha_group_mean)
gh_mb = -1*(H- theta_group_mean)

#i level mb
ig_mb = -1*(targets[,1]-gamma_ind)
ia_mb = -1*(targets[,2]-alpha_ind)
ih_mb = -1*(targets[,3]-theta_ind)

mb_mat[x,] = c(gg_mb,ga_mb, gh_mb, ig_mb, ia_mb, ih_mb)

#MEANS (individual level)
ind_means_mat[x,] = c(gamma_ind, alpha_ind, theta_ind)
```

```
}

write.csv(rmse_mat, file= "hrmse.csv")
write.csv(mb_mat, file= "hmb.csv")
write.csv(ind_means_mat, file = "MeansMat.csv")
```