Circular Drift Difussion Model on JAGS: Full example

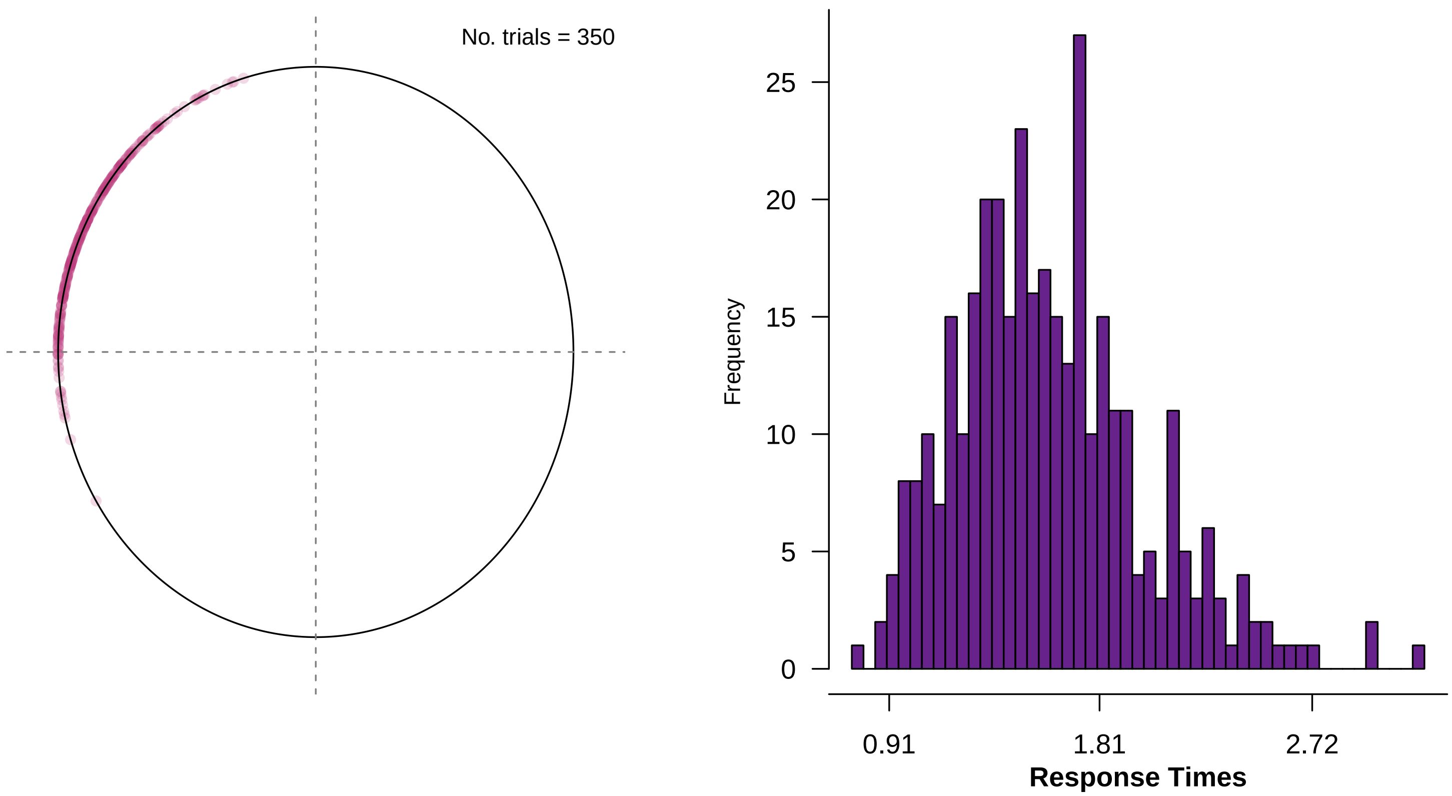
07 September, 2022

1. Generate/load simulated data

```r
# Establish no. of trials
trials <- 350
# Call Rscript to generate simulated data / load it  if already existing
source("./getData.R")
head(data)
```

```
##   Choice     RT
## 1 2.4316 1.0805
## 2 2.6597 1.9685
## 3 2.5052 1.8530
## 4 3.1506 1.4555
## 5 2.0544 1.8920
## 6 2.8280 1.5275
```

```r
# Plot data
cddm.plotData(data)
```




```r
# Print parameter values used to generate this data
par
```

```
## true.theta0  true.drift  true.bound   true.ter0
##       2.685       2.870       3.990       0.230
```

2. Write JAGS model

```
modelFile <- "cddm.bug"
write('
            model{
                  # Likelihood
                    for (i in 1:N) {
                         X[1:2,i] ~ dcddm(drift, bound, ter0, theta0)
                    }

                  # Priors
                    drift  ~ dnorm(0, 0.1)T(0,)
                    theta0 ~ dnorm(0, 0.1)T(-3.14, 3.14)
                    bound  ~ dgamma(3, 2)
                    ter0   ~ dexp(1)T(, tmin)
                  }',
      modelFile)
```

where:

- `drift` is the magnitude of the drift vector composed by the individual drift rates related to the average motion observed across the x and y axes, according to the CDDM.
- `bound` is the threshold (i.e. the radius of the circle)
- `ter0` is the non-decision time (a.k.a. "time for encoding and response")
- `theta0` is the direction of the drift vector, in radians.

Prepare Settings to be passed to JAGS

```
n.chains = 4
n.iter = 2500
n.burnin = 500
n.thin = 1
perParticipant = FALSE
perTask = FALSE

sampling.Settings <- list(n.chains,n.iter,n.burnin,n.thin,perParticipant,perTask)
names(sampling.Settings) <- c("n.chains","n.iter","n.burnin","n.thin","perParticipant","perTask")
```

Run JAGS model

```
# Load Rscript with main function to run jags CDDM model
source("../Functions/runCDDMjags.R")
```

```
samplesFile <- "samples.RData"

if(file.exists(samplesFile)){
   load(file=samplesFile)
   samples
}else{
   myJAGSsampling.CDDM(sampling.Settings,modelFile,samplesFile,data)
 }
```

```
## Inference for Bugs model at "cddm.bug", fit using jags,
##  4 chains, each with 2500 iterations (first 500 discarded)
##  n.sims = 8000 iterations saved
##            mu.vect sd.vect    2.5%      25%      50%      75%    97.5%    Rhat
## bound        3.610   0.803   2.133    3.241    3.976    4.150    4.444   8.221
## drift        2.592   0.388   1.844    2.451    2.773    2.846    2.975   7.847
## ter0         0.310   0.169   0.108    0.193    0.242    0.426    0.626   2.615
```

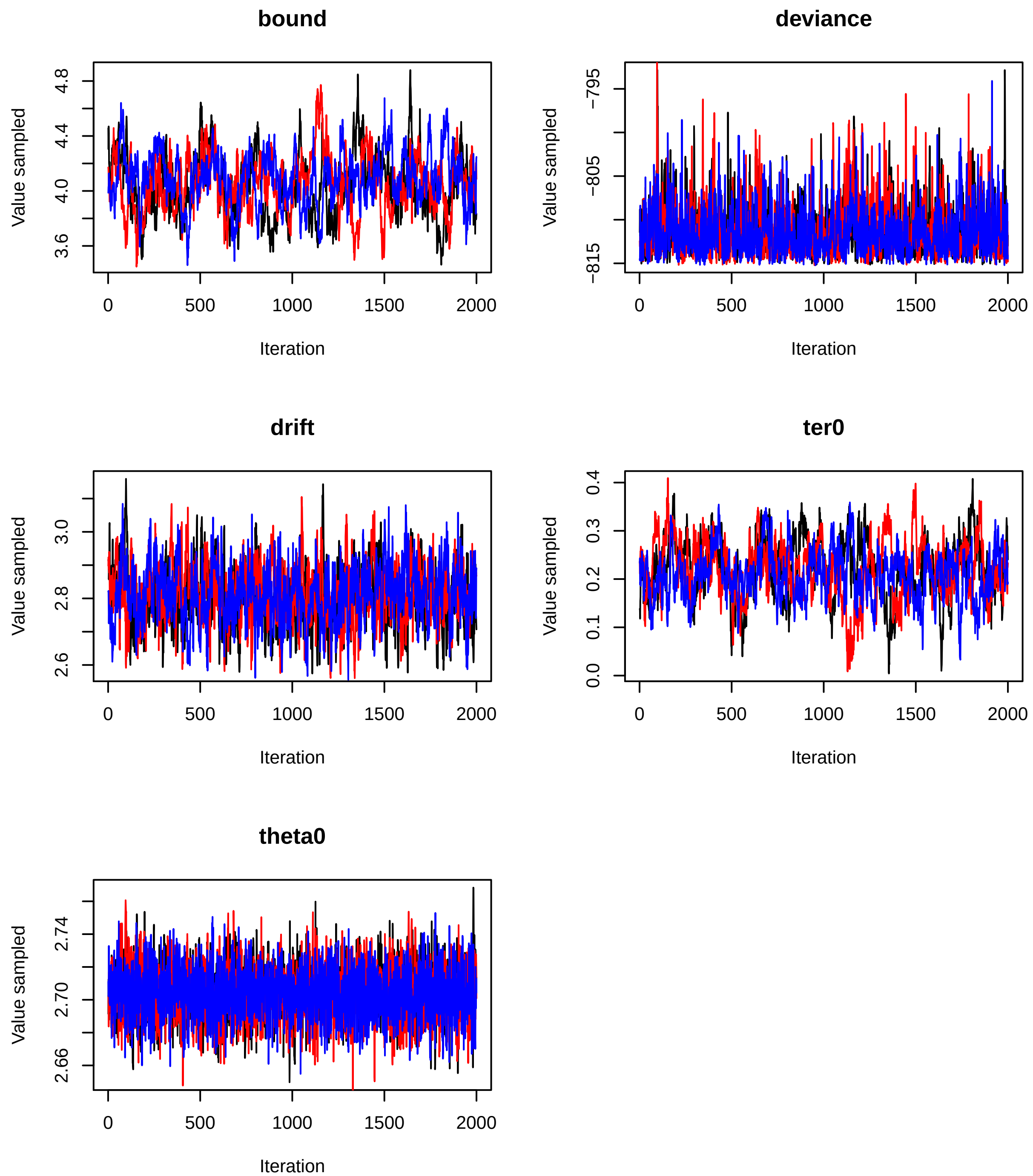
```
## theta0      1.244   2.530   -3.140    1.201    2.698    2.711    2.734 287.737
## deviance -704.660 184.664 -814.736 -812.956 -810.715 -689.986 -380.591  95.095
##          n.eff
## bound        4
## drift        4
## ter0         5
## theta0       4
## deviance     4
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 4.7 and DIC = -699.9
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```r
source("../Functions/processJAGSsamples.R")
drift  <- myJAGSsampling.extractSamples("drift",samples)
bound  <- myJAGSsampling.extractSamples("bound",samples)
ter0   <- myJAGSsampling.extractSamples("ter0",samples)
theta0 <- myJAGSsampling.extractSamples("theta0",samples)
```

```r
# Get descriptive statistics for posterior samples
############################################################
map.theta0 <- JAGSoutput.maxDensity(theta0)
map.drift <- JAGSoutput.maxDensity(drift)
map.bound <- JAGSoutput.maxDensity(bound)
map.ter0 <- JAGSoutput.maxDensity(ter0)
MAPS <- c(map.theta0,map.drift,map.bound,map.ter0)
names(MAPS) <- c("map.theta0","map.drift","map.bound","map.ter0")

mean.theta0 <- mean(theta0)
mean.drift <- mean(drift)
mean.bound <- mean(bound)
mean.ter0 <- mean(ter0)
means <- c(mean.theta0,mean.drift,mean.bound,mean.ter0)
names(means) <- c("mean.theta0","mean.drift","mean.bound","mean.ter0")
```

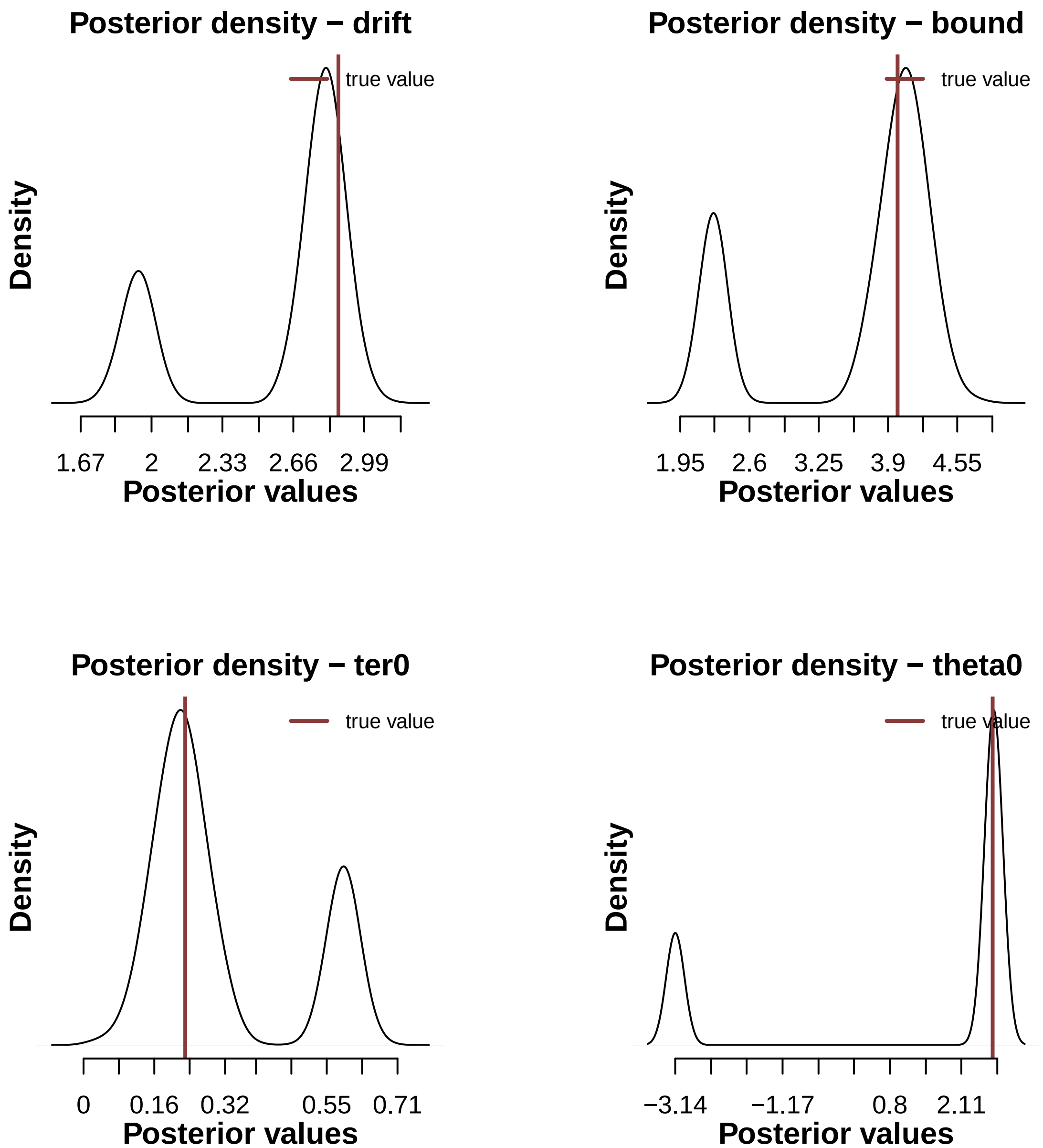
```r
source("../Functions/plotJAGSsamples.R")
par(mfrow = c(3,2))
plot.ShowAllChains(samples)
```




```r
source("../Functions/processJAGSsamples.R")
myJAGSsampling.Rhat.max(samples)
```

```
##    bound deviance    drift     ter0   theta0
## 1.539606 1.532334 1.529922 1.545678 1.529171
```

```r
par(mfrow = c(2,2))
plot.PosteriorDensity(drift,par["true.drift"])
plot.PosteriorDensity(bound,par["true.bound"])
plot.PosteriorDensity(ter0,par["true.ter0"])
plot.PosteriorDensity(theta0,par["true.theta0"])
```



Check against EZ-estimates

```
source("../Functions/ezcdm.R")
EZ <- ezcdm.fit(data$Choice,data$RT)


## First, compare EZ estimates against true parameter values
EZ
```

```
## EZ.theta0  EZ.drift  EZ.bound   EZ.ter0
## 2.7043808 2.8427654 4.0235061 0.2442079
```

```
par
```

```
## true.theta0  true.drift  true.bound   true.ter0
##       2.685       2.870       3.990       0.230
```

```
# Then, compare against point descriptors for posterior samples
##########################################################################
MAPS
```

```
## map.theta0  map.drift  map.bound   map.ter0
##      2.704      2.810      4.068      0.219
```

```
means
```

```
## mean.theta0  mean.drift  mean.bound   mean.ter0
##   1.2435970   2.5920722   3.6100331   0.3096048
```