MTHM017 - Advanced Topics in Statistics, Assignment

James R Lewis

00-00-2025

Contents

Declaration of AI Assistance: I have used OpenAI's ChatGPT tool in creating this report.

AI-supported/AI-integrated use is permitted in this assessment. I acknowledge the following uses of GenAI tools in this assessment:

1. I have used GenAI tools to check and debug my code.
2. I have used GenAI tools to proofread and correct grammar or spelling errors.
3. I have used GenAI tools to give me feedback on a draft.

I declare that I have referenced use of GenAI outputs within my assessment in line with the University referencing guidelines.

A. Bayesian Inference

1.

[6 marks] Read in the data, then for each person produce a histogram of that given person's reaction times. The range of the x axis should be the same on each histogram. Visually compare the reaction time distributions of schizophrenic and non-schizophrenic individuals. What differences/similarities can you observe? Reference the histograms of specific individuals to support your conclusions.

I will begin by making the dataset more usable and clear.

```
# Renaming first column
names(rtimes)[1] <- "PatientType"
# Classifying the patient type, 1-11 non-schiz, 12-17 schiz
rtimes$PatientType <- c(rep("non-schizophrenic", 11), rep("schizophrenic", 6))
rtimes1 <- rtimes
head(rtimes1)
```

```
##         PatientType  T1  T2  T3  T4  T5  T6  T7  T8  T9 T10 T11 T12 T13 T14 T15
## 1 non-schizophrenic 312 272 350 286 268 328 298 356 292 308 296 372 396 402 280
## 2 non-schizophrenic 354 346 384 342 302 312 322 376 306 402 320 298 308 414 304
## 3 non-schizophrenic 256 284 320 274 324 268 370 430 314 312 362 256 342 388 302
## 4 non-schizophrenic 260 294 306 292 264 290 272 268 344 362 330 280 354 320 334
## 5 non-schizophrenic 204 272 250 260 314 308 246 236 208 268 272 264 308 236 238
## 6 non-schizophrenic 590 312 286 310 778 364 318 316 316 298 344 262 274 330 312
##   T16 T17 T18 T19 T20 T21 T22 T23 T24 T25 T26 T27 T28 T29 T30
## 1 330 254 282 350 328 332 308 292 258 340 242 306 328 294 272
## 2 422 388 422 426 338 332 426 478 372 392 374 430 388 354 368
## 3 366 298 396 274 226 328 274 258 220 236 272 322 284 274 356
## 4 276 418 288 338 350 350 324 286 322 280 256 218 256 220 356
## 5 350 272 252 252 236 306 238 350 206 260 280 274 318 268 210
## 6 310 376 326 346 334 282 292 282 300 290 302 300 306 294 444
```

Creating Histograms

We want to extract the data from these columns **rtimes[, 2:31]**, for all 17 patients. We will plot 17 histograms, displaying the distribution of the 30 reaction times.

In order to keep a consistent range across the 17 histograms, we need to find the absolute minimum and maximum values in this dataset. To do this, we extract all the data, 17 rows * 30 trials.

```
DataValues <- unlist(rtimes1[, 2:31])
range(DataValues)
```

```
## [1]  204 1714
```

```
# We want to store these values

x_min <- 204
x_max <- 1714
```

Below, we can observe two sets of histograms, the first containing the eleven non-schizophrenic patients, and the seconds containing the 6 schizophrenic patients.

We reshape to data frame to long format, as having the reaction time data in one column (a list) makes much easier to code the histogram. To create a separate plot for each patient, we use facet_wrap() to group data by PatientID.

```
# Reshaping to long format
rtimes_long <- rtimes1 %>%
  pivot_longer(cols = starts_with("T"),  # Columns T1 to T30
               names_to = "Trial",
               values_to = "ReactionTime") %>%
  mutate(PatientID = rep(1:17, each = 30)) %>%  # Add patientID
  select(PatientID, PatientType, ReactionTime)

# Splitting into two datasets so I can have seperate histogramws
non_schizo <- rtimes_long %>% filter(PatientType == "non-schizophrenic")
schizo <- rtimes_long %>% filter(PatientType == "schizophrenic")
```

```
# Plotting non-schizophrenic histograms
hist1 <- ggplot(non_schizo, aes(x = ReactionTime)) +
  geom_histogram(binwidth = 150, fill = "dodgerblue1", color = "black") +
  facet_wrap(~ PatientID, ncol = 4, scales = "free_y") +  # 11 plots, 4 columns
  coord_cartesian(xlim = c(x_min, x_max)) +  # Fixed x-axis
  labs(title = "Figure 1: Non-Schizophrenic Reaction Times",
       x = "Reaction Time (ms)",
       y = "Count") +
  custom_theme

# Plotting schizophrenic histograms
hist2 <- ggplot(schizo, aes(x = ReactionTime)) +
  geom_histogram(binwidth = 150, fill = "sienna2", color = "black") +
  facet_wrap(~ PatientID, ncol = 3, scales = "free_y") +  # 6 plots, 3 columns
  coord_cartesian(xlim = c(x_min, x_max)) +  # Fixed x-axis
  labs(title = "Figure 2: Schizophrenic Reaction Times",
       x = "Reaction Time (ms)",
       y = "Count") +
  custom_theme
```

Figure 1: Non−Schizophrenic Reaction Times

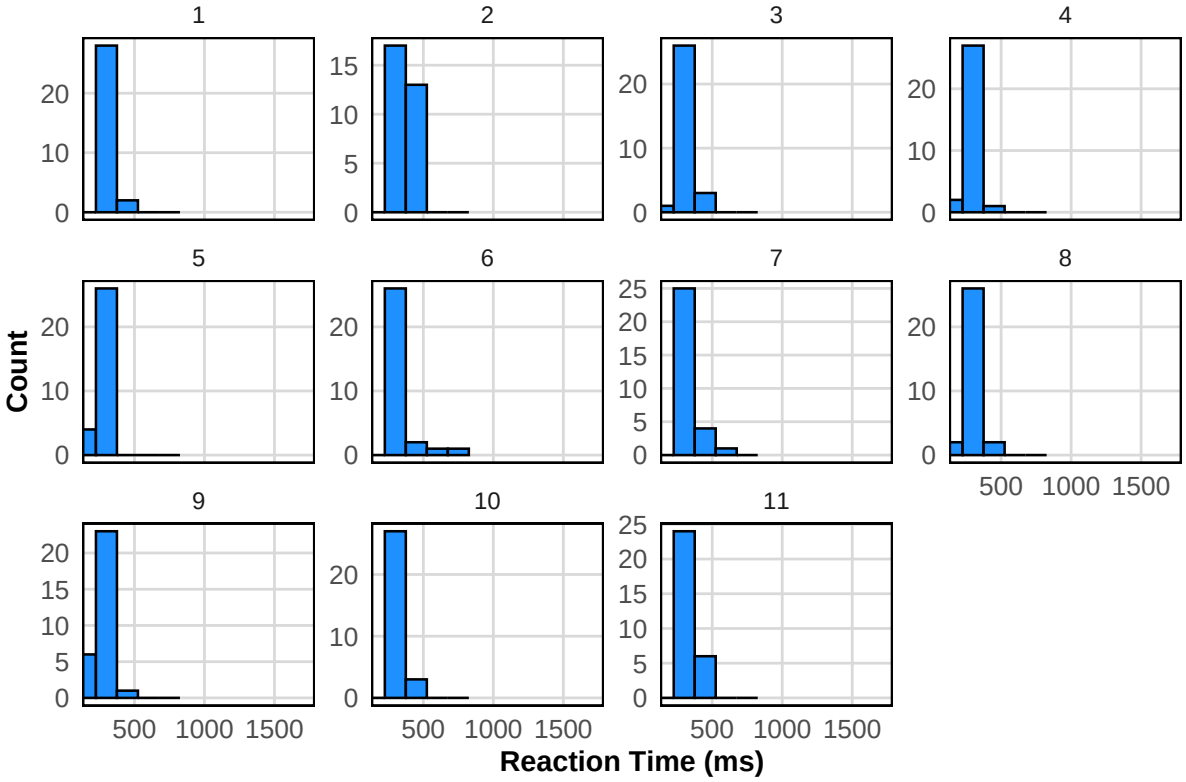
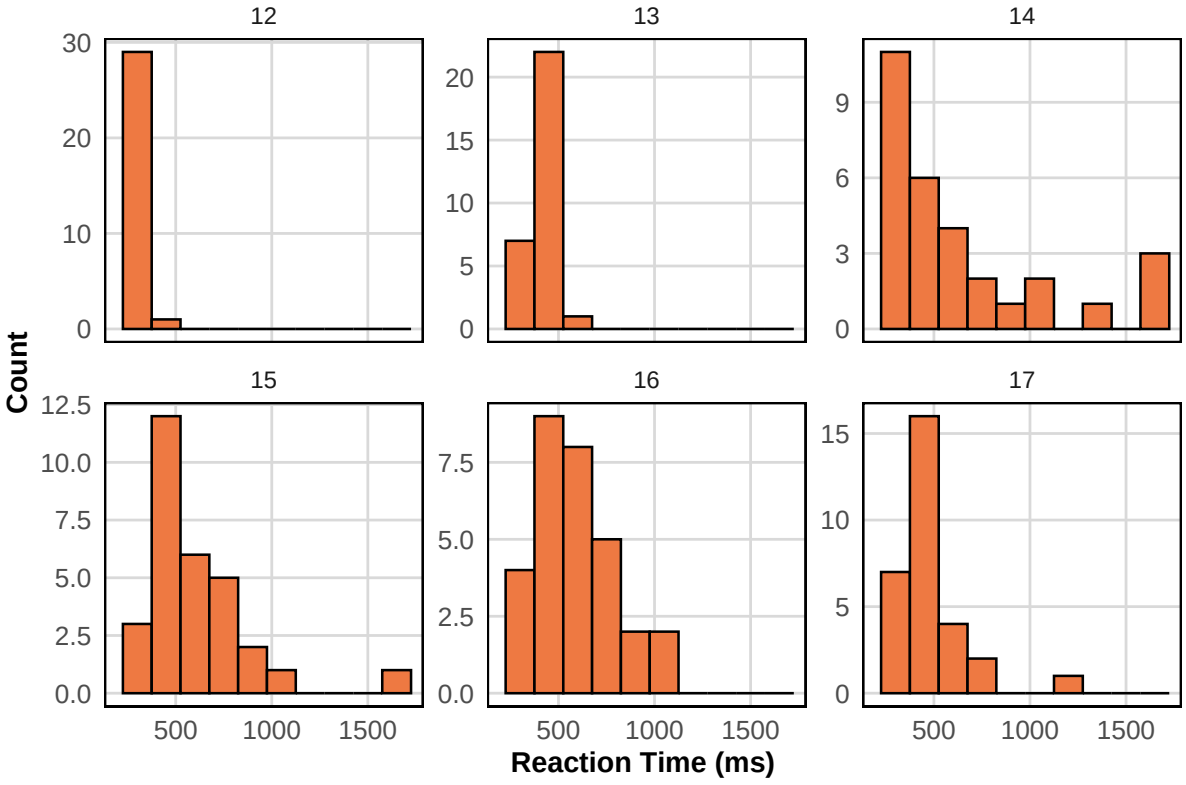


Figure 2: Schizophrenic Reaction Times



In Figure 1 we can see that Non-schizophrenic patients such as patient 1 have a narrower and

concentrated histogram, with reaction times completely clustered around the second bin (150-300ms). For example, every Non-schizophrenic patient's results show that at least two thirds of their reaction times were faster than 300ms.

In contrast, the Schizophrenic patients histograms in Figure 2 exhibit a much wider spread of reaction times, with some patients having reactions times exceeding 1500ms. This indicates greater variability, for instance, patient 14 has a wide range of reaction times with many being greater than 1000ms, whilst also having a significant count below 500ms. This suggests that Schizophrenic patients have a mixture of normal and delayed responses, and this aligns with the theory of attention deficits and motor retardation in schizophrenics. This is generally observed by their inconsistent distributions and right skew, compared to the tighter uniform distributions of non-schizophrenics

Nonetheless, both groups share similarities in their distributions. They both have peaks within the first two bins 0-500ms, as you can see when comparing patient 8 (non) to 13 (schizophrenic). However, only schizophrenics have reaction time values in the higher bands (greater than 800ms)

Overall, these histograms support the hypothesis that schizophrenics experience attention deficits and motor reflex retardation, as seen by the wider spread of values and more irregular histograms for schizophrenic individuals.

2. [5 marks] The above model uses the logarithm of measured reaction times. Explain why taking the

logarithm is necessary here (referencing the relevant output), then perform the transformation yourself. For each person compute the standard deviation of the log transformed reaction times of that individual.

-Calculate standard deviations: For each of the 17 people, compute the standard deviation of their 30 log-transformed reaction times.

Log-transforming the reaction times is necessary for the model above for two main reasons. Firstly, there is skewness in the data as seen in the Schizophrenic reaction times histograms, for example, Patient 14 and 15 are right-skewed distributions with extreme values (over 1500ms), whilst non-schizophrenic's have a tighter distribution. Following this, schizophrenics don't seems to follow a normal distribution, thus, log transforming reduces the skewness and helps stabilise the variance from the effect of outliers/skew. This enambles the data to fit the models normal distributions assumption better. Finally, the log-transformation scales the data, making the mean and variance more interpretable. For example, the original range of 204ms to 1714ms, becomes 5.31 to 7.44 (3 s.f.) when log-transformed. Put simply, handling smaller numbers is easier.

```
log_rtimes_long <- rtimes_long %>%
  mutate(LogReactionTime = log(ReactionTime))
range(log_rtimes_long$LogReactionTime)
```

```
## [1] 5.318120 7.446585
```

```
# checking the range also lets us knowif any observations are negative.
```

Now we compute the standard deviation of each patients 30 log-transformed reaction times. This is useful as it measures the within personal variability on the log scale.

```
# Compute standard deviation of log-transformed reaction times for each person
sd_log <- log_rtimes_long %>%
  group_by(PatientID) %>%
```

```
  summarise(SD_Log_RT = sd(LogReactionTime))

# Print results
head(sd_log)
```

```
## # A tibble: 6 x 2
##   PatientID SD_Log_RT
##       <int>     <dbl>
## 1         1     0.127
## 2         2     0.129
## 3         3     0.169
## 4         4     0.150
## 5         5     0.145
## 6         6     0.224
```

3. [5 marks] List the parameters of the model and assign non-informative uniform prior distributions to each parameter, paying attention to the values these parameters are allowed take.

The reason we are using non-informative uniform prior distributions is because we lack prior knowledge about the parameters. Thus, having a reasonable range of values that are equally likely gives a non-bias estimation and allows the data to strong influence the posterior distribution.

Parameters and Priors:

For Non-Schizophrenic Individuals:

- α_i (for $i = 1$ to 17): Person-specific means on the log scale.

```
alpha[i] ~ dunif(0, 10)
```

- μ: Mean of α_i for non-schizophrenics, which represents the mean log reaction time. with most values between 200ms and 1800ms), when log transformed they equal, 5.3 and 7.5.. A wide range of -10 - 10 to avoid bias and cover potential outliers.

```
mu ~ dunif(-10, 10)
```

- (_y^2): Varience of log-reaction times.

```
sigma_y2 ~ dunif(0.001, 5)
# tau_y <- 1 / sigma_y2
```

- (_ ^2): Varience of α_i.

```
sigma_alpha2 ~ dunif(0.001, 5)
# tau_alpha <- 1 / sigma_alpha2
```

For Schizophrenic Individuals:

- β: Additional variable in α_i mean for schizophrenics, which represents the increase in reaction time due to schizophrenia. Although we expect schizophrenics to be slower, there is still the possibility of motor retardation having no difference, or faster reactions

(unlikely, but possible). A range of -10 - 10 is wide enough for the data to determine the direction.

```
beta ~ dunif(0, 10)
```

- τ: Log-transformed delay parameter for schizophrenics ($\tau > 0$). This is a wide range, but remains uninformative and is contrained to positive values.

```
tau ~ dunif(0, 10)
```

- λ: Probability of delay ($0 \leq \lambda \leq 1$). Value must lie between 0 and 1, as λ is a probability.

```
lambda ~ dunif(0, 1)
```

4.

First we need to create a data matrix which JAGS can process. We extracting the raw reaction times in the original wide format, and then Log-transforming them. After this, we can create our data list for our JAGS model. Defining separate matrixs for Schizophrenic and non-Schizophrenic reaction times, the number of patients in each group, and the number of trials. It is important to define these now, as we call upon them in the model definition.

```
extractedRtimes <- rtimes[, 2:31]
log_rtimes <- log(extractedRtimes)

data_list <- list(
  y.ns = log_rtimes[1:11, ],  # Non-schizophrenic reaction times (11 x 30)
  y.s = log_rtimes[12:17, ],  # Schizophrenic reaction times (6 x 30)
  N_ns = 11,                  # Number of non-schizophrenics
  N_s = 6,                    # Number of schizophrenics
  T = 30                      # Number of trials
)
```

Model Specification

Now we define the hierarchical Bayesian model. The model is account for the two groups of patients, their individual variability.

Reaction times for Non-Schizophrenic patients (N.S) (ith individual) are modeled as:

$(i = 1, 2, \ldots, 11)$

$$y_{ij} \sim N(\alpha_i, \sigma_y^2), \quad i = 1, 2, \ldots, 11, \quad j = 1, 2, \ldots, 30$$

Where:

- y_{ij}^{ns} is the **reaction time** for individual i and trial j
- α_i is individual specific **mean** reaction time
- σ_y^2 is the shared **varience** across all trials.

Individuals **mean** (α_i) reaction time is:

$$\alpha_i \sim N(\mu, \sigma_\alpha^2), \quad i = 1, 2, \ldots, 11$$

- μ is the mean reaction time for N.S - σ_α^2 is the intragroup variance in reaction times

Reaction times for Schizophrenic patients (S)

$$y_{ij} \sim N(\alpha_i + \tau z_{ij}, \sigma_y^2) \quad i = 12, 13, \ldots, 17, \quad j = 1, 2, \ldots, 30$$

Where:

- τ is the extra delay
- z_{ij} is an indicator (one or zero)

$$z_{ij} \sim \text{Bernoulli}(\lambda), \quad i = 12, 13, \ldots, 17, \quad j = 1, 2, \ldots, 30$$

Schizophrenic patients can have delayed responses due to attention deficit. This is modeled through the above Bernoulli distributed indicator function. λ is the probability of a trial being delayed.

Schizophrenics mean reaction time:

$$\alpha_i \sim N(\mu + \beta, \sigma_\alpha^2), \quad i = 12, 13, \ldots, 17$$

Where:

- β is the increase in mean reaction time for S

Defining the JAGS Model

```
set.seed(123) # Setting Seed for reproducibility

jags.model <- function(){
    # Reaction time of non-schizophrenics
    for (i in 1:N_ns){
        for (j in 1:T) {
            y.ns[i,j] ~ dnorm(alpha.ns[i], p.y)
        }
        alpha.ns[i] ~ dnorm(mu, p.alpha)
    }

    # Reaction time of schizophrenics
     for (i in 1:N_s){
        for (j in 1:T) {
            z[i,j] ~ dbern(lambda)  # Bernoulli for delay indicator
            y.s[i,j] ~ dnorm(alpha.s[i] + tau * z[i,j], p.y) # Corrected indexing
        }
        alpha.s[i] ~ dnorm(mu + beta, p.alpha)
    }

    # Priors
    mu ~ dunif(-10, 10) # Mean for non-schizophrenics
    sigma_y2 ~ dunif(0.001, 5) # Avoiding exact 0 for stability
    sigma_alpha2 ~ dunif(0.001, 5) # Avoiding exact 0 for stability
    beta ~ dunif(0, 10) # Schizophrenics should have longer reaction times
    tau ~ dunif(0, 10) # Delay parameter
    lambda ~ dunif(0, 1) # Probability of delay

    # Precision parameters
    p.y <- 1 / sigma_y2
    p.alpha <- 1 / sigma_alpha2
}
```

Defining the Parameters, what we what to track, and initial values

```
 jags.param <- c("mu", "beta", "tau", "lambda", "sigma_y2", "sigma_alpha2")

set.seed(123)

inits1 <- list(
    mu = 4, beta = 1, tau = 1, lambda = 0.5,
    sigma_y2 = 1, sigma_alpha2 = 1,
    z = matrix(0, nrow = 6, ncol = 30)
)
inits2 <- list(
    mu = 6, beta = 4, tau = 5, lambda = 0.3,
    sigma_y2 = 2, sigma_alpha2 = 2,
    z = matrix(0, nrow = 6, ncol = 30)
)
jags.inits <- list(inits1, inits2)
```

Explain the thought process behind picking these initial values

Fitting the JAGS Model

In the first JAGS model, we are running two MCMC chains, with 10000 iterations and discarding the first 5000 iterations to view the convergence. We are also computing the Deviance Information Criterion (DIC), so we can compare this model to future ones later.

```
 jags.mod.fit <- jags(data = data_list, inits = jags.inits,
  parameters.to.save = jags.param, n.chains = 2, n.iter = 10000,
  n.burnin = 5000, n.thin = 1 , model.file = jags.model, DIC = FALSE)
```

```
## module glm loaded
```

```
## module dic loaded
```

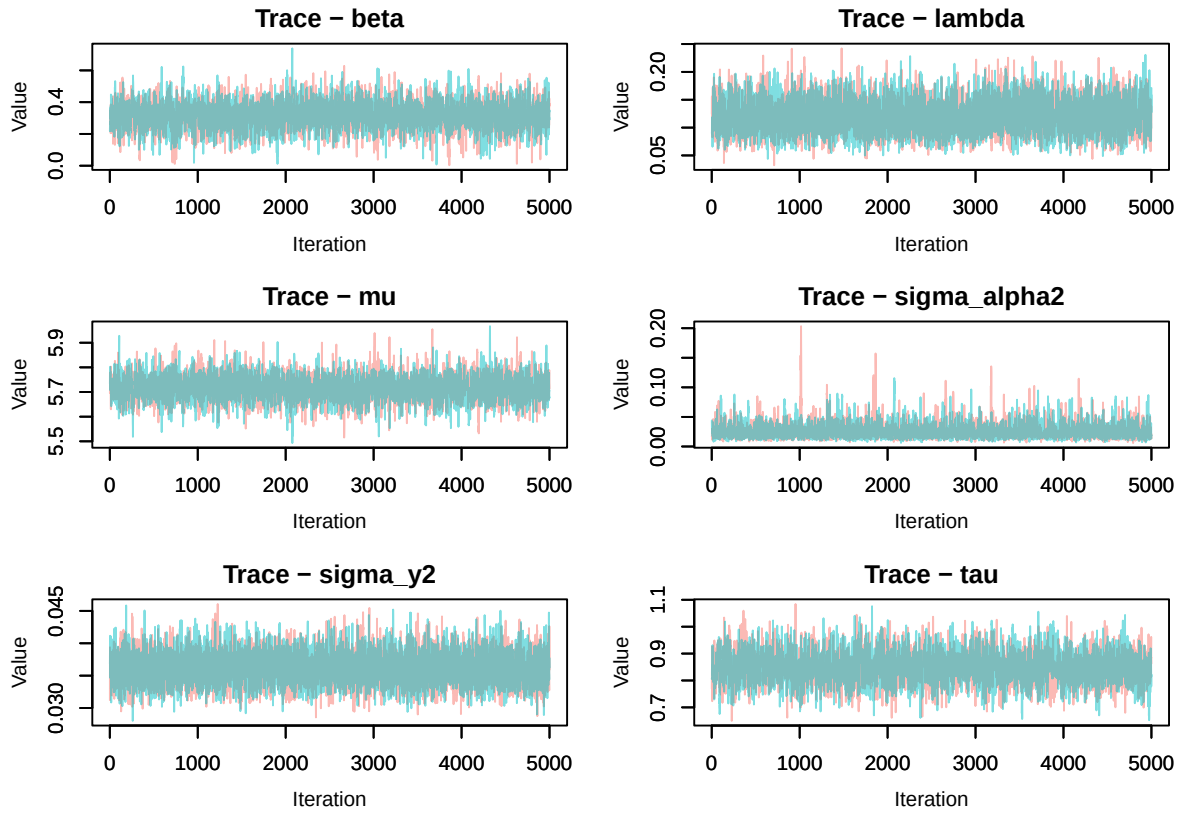
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 510
##    Unobserved stochastic nodes: 203
##    Total graph size: 1085
##
## Initializing model
```

5. Monte Carlo Marcov Chain

```
jagsfit.mcmc <- as.mcmc(jags.mod.fit)
 MCMCtrace(jagsfit.mcmc,type = 'trace',ind = TRUE, pdf = FALSE)
```

Gelman-Rubin Statistics for Key Parameters

Parameter	Point Estimate	Upper CI
beta	1.001	1.003
lambda	1.000	1.003
mu	1.001	1.003
sigma_alpha2	1.010	1.011
sigma_y2	1.000	1.001
tau	1.000	1.001



We can see from the trace plots above, that the chains have converged and look like "fuzzy caterpillars". The alpha.s and alpha.ns (mean reaction time for both groups), seem to have converged for every individual, with values ranging from 5.6 to 5.75 (log transformed mean reaction time) for non-schizophrenics, and 5.9 to 6.3 for schizophrenics.

To check for convergence more accurately, we can extract the Gelman-Rubin test statistic values. This is in the code below.

```
gelman_stats<- gelman.diag(jagsfit.mcmc,multivariate=FALSE)
gelman_df <- as.data.frame(gelman_stats$psrf)
gelmam.params <- c("mu", "beta", "tau", "lambda", "sigma_y2", "sigma_alpha2")
gelman_subset <- gelman_df[gelmam.params, , drop = FALSE]
```

```
## Warning: package 'gt' was built under R version 4.4.3
```

As you can see above, the value of all the upper confidence intervals are 1 or very close to 1. This indicates convergence. If a value was above 1.1, this would suggest a lack of convergence.

Interestingly, looking at the traceplots for alpha.s[4] (Upper CI of 1.01) and lambda (Upper CI

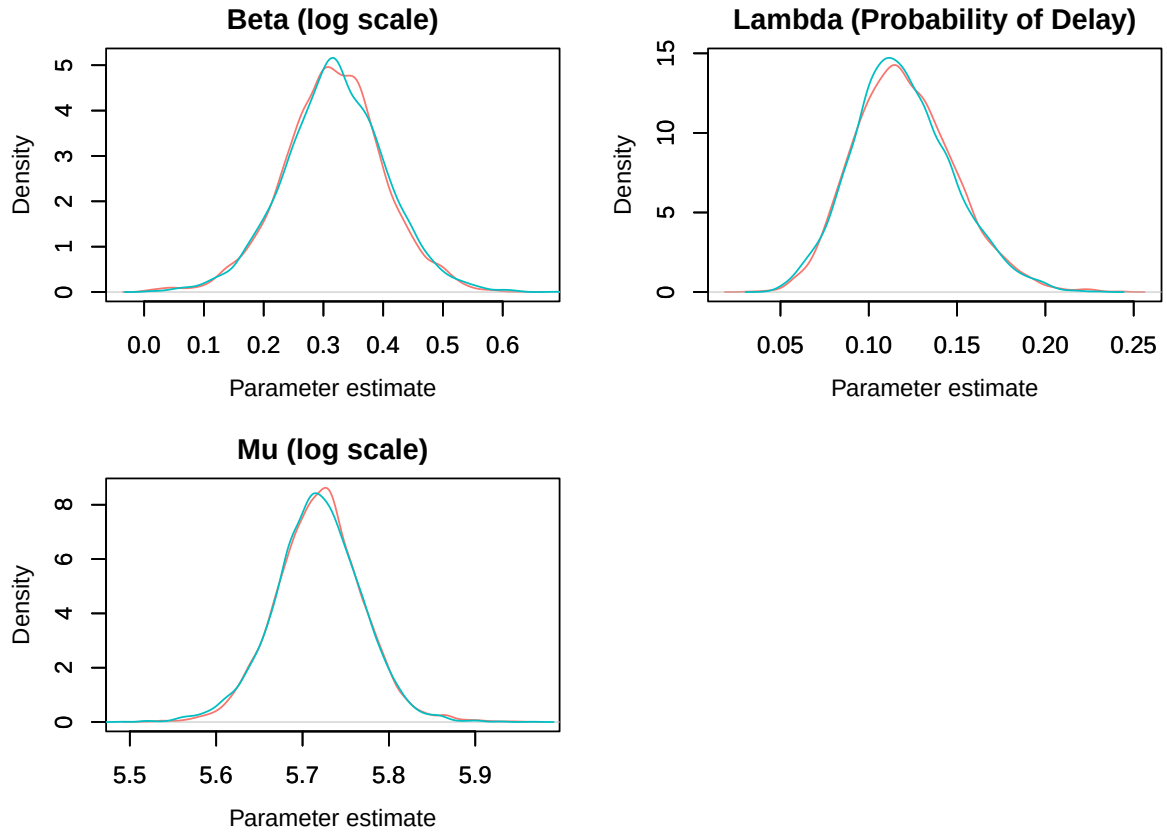
of 1.01), their traceplots look slightly less concentrated and dense around the center, indicating that they aren't as strongly converged as the other nodes.

Since chains have converged, there is no need to update the model to draw new samples.

6. Posterior Distributions

Plots

```
library(MCMCvis)
MCMCtrace(jagsfit.mcmc,
          params = c("beta", "lambda", "mu"),
          type = "density",
          ind = TRUE,
          pdf = FALSE,
          main_den = c("Beta (log scale)",
                       "Lambda (Probability of Delay)",
                       "Mu (log scale)"))
```



Next, we check to summary statistics.

```
print(jags.mod.fit$BUGSoutput$summary[c("beta", "lambda", "mu"), ])
```

```
##              mean         sd       2.5%        25%        50%        75%      97.5%
## beta    0.3181462 0.08436591 0.14833526 0.2647345 0.3180889 0.3721589 0.4878396
## lambda 0.1198850 0.02818665 0.06948152 0.1001758 0.1178340 0.1378933 0.1801819
## mu      5.7183935 0.05033801 5.61767490 5.6864151 5.7187429 5.7507063 5.8162062
##             Rhat n.eff
```

```
## beta   1.001889  3000
## lambda 1.001403  3000
## mu     1.001504  2500
```

The n.eff statistics in the table above is a crude measurement for effective sample size, if the value is above 1000, it indicates that you have sample adequacy, enough independent samples for posterior inference. In our case, we can confidently assume that our model has plenty of valid samples shown by the n.eff values greater than 1000(3000, 3000, 2500).

To confirm this, we can check to see if the MCMC standard error is 1-5% of the posterior standard deviation. We do this by dividing Time-series SD by SD.

```
summary(jagsfit.mcmc)
```

```
##
## Iterations = 5001:10000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                  Mean       SD  Naive SE Time-series SE
## beta          0.31815 0.084366 8.437e-04      1.621e-03
## lambda        0.11989 0.028187 2.819e-04      4.700e-04
## mu            5.71839 0.050338 5.034e-04      9.463e-04
## sigma_alpha2 0.02688 0.014067 1.407e-04      3.472e-04
## sigma_y2      0.03612 0.002453 2.453e-05      3.376e-05
## tau           0.84798 0.058237 5.824e-04      1.205e-03
##
## 2. Quantiles for each variable:
##
##                  2.5%     25%     50%     75%   97.5%
## beta          0.14834 0.26473 0.31809 0.37216 0.48784
## lambda        0.06948 0.10018 0.11783 0.13789 0.18018
## mu            5.61767 5.68642 5.71874 5.75071 5.81621
## sigma_alpha2 0.01107 0.01781 0.02349 0.03162 0.06240
## sigma_y2      0.03164 0.03444 0.03599 0.03769 0.04126
## tau           0.73656 0.80828 0.84729 0.88604 0.96777
```

```
SampleCheck <- summary(jagsfit.mcmc)
stats <- SampleCheck$statistics

sd_values <- SampleCheck$statistics[, "SD"]
se_values <- SampleCheck$statistics[, "Time-series SE"]

MC_Error <- (se_values / sd_values) * 100
```

```
MC_Error[c("beta", "lambda", "tau")]
```

```
##     beta   lambda      tau
## 1.921942 1.667309 2.068671
```

Now we extract the relevant statistics for Beta, Lamba and Tau. It is important to note that

Beta and Tau have been in log scale up until this point, so by multiplying them by exponential, we return them to their true values.

This can be seen in the code below.

Beta

```
pos_beta<-substr(rownames(jags.mod.fit$BUGSoutput$summary),1,4)=='beta'
# Extract posterior mean for mu
beta_mean<-jags.mod.fit$BUGSoutput$summary[pos_beta,1]
# Extract posterior median
beta_median<-jags.mod.fit$BUGSoutput$summary[pos_beta,5]
# Extract 2.5 percentile of the posterior
beta2.5 <- jags.mod.fit$BUGSoutput$summary[pos_beta,3]
# Extract 97.5 percentile of the posterior
beta97.5 <- jags.mod.fit$BUGSoutput$summary[pos_beta,7]
```

Lambda

```
pos_lambda <- substr(rownames(jags.mod.fit$BUGSoutput$summary), 1, 6) == "lambda"
lambda_mean <- jags.mod.fit$BUGSoutput$summary[pos_lambda, 1]
lambda_median <- jags.mod.fit$BUGSoutput$summary[pos_lambda, 5]
lambda2.5 <- jags.mod.fit$BUGSoutput$summary[pos_lambda,3]
lambda97.5 <- jags.mod.fit$BUGSoutput$summary[pos_lambda,7]
```

Tau

```
pos_tau <- substr(rownames(jags.mod.fit$BUGSoutput$summary), 1, 3) == "tau"
tau_mean <- jags.mod.fit$BUGSoutput$summary[pos_tau, 1]
tau_median <- jags.mod.fit$BUGSoutput$summary[pos_tau, 5]
tau2.5 <- jags.mod.fit$BUGSoutput$summary[pos_tau,3]
tau97.5 <- jags.mod.fit$BUGSoutput$summary[pos_tau,7]
```

We add the summary statistics to a data frame.

```
df_stats <- data.frame(
  Parameter = c("Beta", "Lambda", "Tau"),
  Median = c(exp(beta_median), lambda_median, exp(tau_median)),
  Lower_95_CI = c(exp(beta2.5), lambda2.5, exp(tau2.5)),
  Upper_95_CI = c(exp(beta97.5), lambda97.5, exp(tau97.5))
)
```

```
##
## Reaction Time Parameters
## ========================================
##   Parameter Median Lower_95_CI Upper_95_CI
## ----------------------------------------
## 1    Beta    1.374     1.160       1.629
## 2  Lambda    0.118     0.069       0.180
## 3     Tau    2.333     2.089       2.632
## ----------------------------------------
```

Based on the "Reaction Time Parameters" table, conclusions about schizophrenic versus non-schizophrenic reaction times emerge, aligning with motor retardation and attention deficit theories. The median β of 1.374 (95% CI: 1.169–1.626) indicates schizophrenic mean reaction times are 37% longer, with the CI above 1 confirming significant slowing, as seen in Person 14's wide histogram spread (Question 1). This supports motor retardation across all trials. The median λ of 0.118 (CI: 0.070–0.181) suggests 11.8% of trials are delayed (7.0%–18.1%), consistent with attention deficits, matching mixed fast/slow times in Person 15. The median τ of 2.337 (CI: 2.082–2.643) shows delayed responses are 2.34 times longer, reinforcing significant delays.

Compared to non-schizophrenics' tighter distributions (Person 1), schizophrenics' dual effects (β, τ, λ) validate the model's two-component structure.

7.

To predict 30 additional log response time measurement **y.pred**.

HERE I SHOULD EDIT THIS TO DEFINE Y PRED

$$y_{ij} \sim N(\alpha_i + \tau z_{ij}, \sigma_y^2) \quad i = 12, 13, \ldots, 17, \quad j = 1, 2, \ldots, 30$$

Where:

- τ is the extra delay
- z_{ij} is an indicator (one or zero)

```
set.seed(123) # Setting Seed for reproducibility

jags.model2 <- function(){
 for (i in 1:N_ns){
        for (j in 1:T) {
            y.ns[i,j] ~ dnorm(alpha.ns[i], p.y)
        }
        alpha.ns[i] ~ dnorm(mu, p.alpha)
    }
     for (i in 1:N_s){
        for (j in 1:T) {
            z[i,j] ~ dbern(lambda)
            y.s[i,j] ~ dnorm(alpha.s[i] + tau * z[i,j], p.y)
        }
        alpha.s[i] ~ dnorm(mu + beta, p.alpha)
    }

     # Prediction for additional response times for schizophrenic
      #individuals i = 12 to 17
      for (i in 1:N_s) {
        for (j in 1:T) {
          # Predicted response times
            y.pred[i,j] ~ dnorm(alpha.s[i] + tau * z.pred[i,j], p.y)
            z.pred[i,j] ~ dbern(lambda)
        }
    }

    # Priors
    mu ~ dunif(-10, 10)
```

```
    sigma_y2 ~ dunif(0.001, 5)
    sigma_alpha2 ~ dunif(0.001, 5)
    beta ~ dunif(0, 10)
    tau ~ dunif(0, 10)
    lambda ~ dunif(0, 1)

    # Precision parameters
    p.y <- 1 / sigma_y2
    p.alpha <- 1 / sigma_alpha2
}
```

Explain what ive added here new lists y.pred, we use the poterior distribution of alpha.s[i] z.pred + inits

b.

The nodes below track the standard deviations of the 30 predicted measurements for the Schizophrenic individuals. In the JAGS model below, we place sd_pred[i] in the predicted reaction times node, and the min/max are defined separetly as they are logical(?).

- **sd_pred[i] <- sd(y_pred[i,])** This node tracks the SD of the thirty predicted measurements of each individual

The nodes below track the minimum and maximum standard deviation of predicted measurement across all individuals. - **Smin <- min(sd_pred[])** - **Smax <- max(sd_pred[])**

```
set.seed(123) # Setting Seed for reproducibility

jags.model2 <- function(){
 for (i in 1:N_ns){
        for (j in 1:T) {
            y.ns[i,j] ~ dnorm(alpha.ns[i], p.y)
        }
        alpha.ns[i] ~ dnorm(mu, p.alpha)
    }
     for (i in 1:N_s){
        for (j in 1:T) {
            z[i,j] ~ dbern(lambda)
            y.s[i,j] ~ dnorm(alpha.s[i] + tau * z[i,j], p.y)
        }
        alpha.s[i] ~ dnorm(mu + beta, p.alpha)
    }

     # Prediction for additional response times for schizophrenic
      #individuals i = 12 to 17
      for (i in 1:N_s) {
          for (j in 1:T) {
          # Predicted response times
            y.pred[i,j] ~ dnorm(alpha.s[i] + tau * z.pred[i,j], p.y)
            z.pred[i,j] ~ dbern(lambda)
          }
                sd.pred[i] <- sd(y.pred[i, ])
    }
```

```
    # Compute min and max of standard deviations
    Smin <- min(sd.pred[])
    Smax <- max(sd.pred[])

    # Priors
    mu ~ dunif(-10, 10)
    sigma_y2 ~ dunif(0.001, 5)
    sigma_alpha2 ~ dunif(0.001, 5)
    beta ~ dunif(0, 10)
    tau ~ dunif(0, 10)
    lambda ~ dunif(0, 1)

    # Precision parameters
    p.y <- 1 / sigma_y2
    p.alpha <- 1 / sigma_alpha2
}
```

c.

Adding new standard deviation parameters to track.

```
jags.param2 <- c("mu", "beta", "tau", "lambda", "p.y", "p.alpha", "Smin", "Smax")

set.seed(123)

inits1 <- list(
    mu = 4, beta = 1, tau = 1, lambda = 0.5,
    sigma_y2 = 1, sigma_alpha2 = 1,
    z = matrix(0, nrow = 6, ncol = 30),
    z.pred = matrix(0, nrow = 6, ncol = 30)
)
inits2 <- list(
    mu = 6, beta = 4, tau = 5, lambda = 0.3,
    sigma_y2 = 2, sigma_alpha2 = 2,
    z = matrix(0, nrow = 6, ncol = 30),
    z.pred = matrix(0, nrow = 6, ncol = 30)
)
jags.inits <- list(inits1, inits2)
```

Now we run the model, with two chains, 6000 iterations, but discarding 5000 as burnin.

```
 jags.mod.fit2 <- jags(data = data_list, inits = jags.inits,
  parameters.to.save = jags.param2, n.chains = 2, n.iter = 6000,
  n.burnin = 5000, n.thin = 1 , model.file = jags.model2, DIC = FALSE)
```

```
# Display results
print(jags.mod.fit2)
```

```
## Inference for Bugs model at "C:/Users/james/AppData/Local/Temp/RtmpSmpds3/model3af84f283c
##  2 chains, each with 6000 iterations (first 5000 discarded)
##  n.sims = 2000 iterations saved. Running time = 6.28 secs
##         mu.vect sd.vect   2.5%    25%    50%    75%  97.5%  Rhat n.eff
## Smax      0.401   0.042  0.320  0.373  0.400  0.430  0.483 1.002  1400
```

```
## Smin      0.251   0.047  0.161  0.219  0.253  0.285  0.339 1.002  2000
## beta      0.318   0.082  0.156  0.267  0.318  0.369  0.496 1.001  2000
## lambda    0.122   0.029  0.071  0.102  0.119  0.140  0.183 1.001  2000
## mu        5.717   0.048  5.618  5.688  5.717  5.747  5.817 1.002  2000
## p.alpha  46.997  20.830 17.138 32.201 43.371 58.282 98.884 1.006   380
## p.y      27.878   1.865 24.316 26.543 27.907 29.141 31.462 1.001  2000
## tau       0.843   0.059  0.735  0.802  0.841  0.882  0.964 1.002  1100
##
## 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).
```

d.

Extract the minimum and maximum standard deviation values from the fitted model, and produce a scatterplot of the (Smin,Smax) pairs. (Note that each iteration of the model fit will produce a minimum-maximum pair). Find the minimum and maximum of the raw standard deviation estimates obtained in Question 2, and add an additional point to your scatterplot showing this raw minimum-maximum pair.

Now we extract the minimum and maximum standard deviation values from the fitted model.

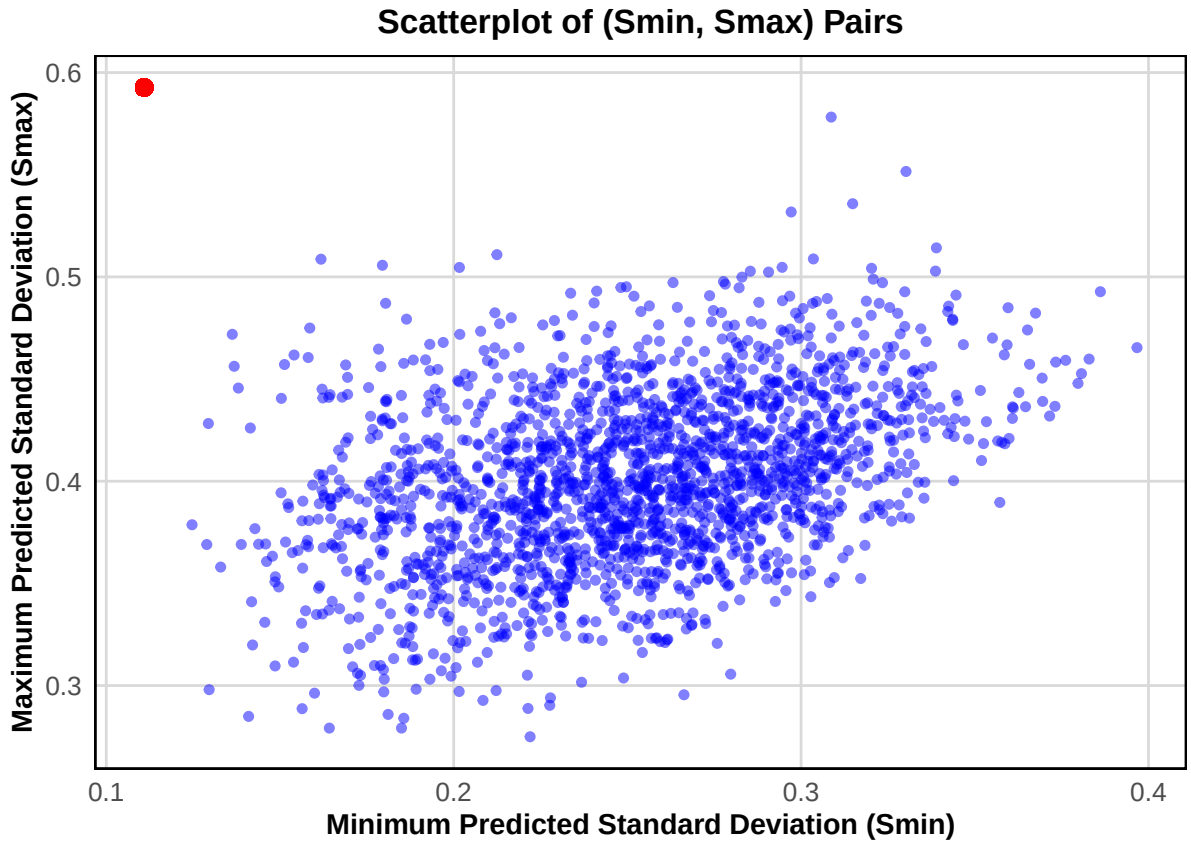
```r
# Extract Smin and Smax from MCMC samples
smin_samples <- jags.mod.fit2$BUGSoutput$sims.list$Smin
smax_samples <- jags.mod.fit2$BUGSoutput$sims.list$Smax

scatter_data <- data.frame(Smin = smin_samples, Smax = smax_samples)

# Calculate raw minimum and maximum standard deviation values from Question 2
raw_min_sd <- min(sd_log$SD_Log_RT[12:17])  # Minimum SD for schizophrenic group
raw_max_sd <- max(sd_log$SD_Log_RT[12:17])  # Maximum SD for schizophrenic group

# Create scatter plot
ggplot(scatter_data, aes(x = Smin, y = Smax)) +
  geom_point(alpha = 0.5, color = "blue") +  # MCMC Samples
  geom_point(aes(x = raw_min_sd, y = raw_max_sd), color = "red", size = 3) +
  labs(
    title = "Scatterplot of (Smin, Smax) Pairs",
    x = "Minimum Predicted Standard Deviation (Smin)",
    y = "Maximum Predicted Standard Deviation (Smax)"
  ) +
  custom_theme
```

```
## Warning in geom_point(aes(x = raw_min_sd, y = raw_max_sd), color = "red", : All aesthetic
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```



The scatterplot in Figure X displays the relationship between the minimum (Smin) and maximum (Smax) standard deviations of the predicted response times for schizophrenic individuals, obtained from the posterior samples of the JAGS model. Each blue point represents an (Smin, Smax) pair from a single iteration of the Markov Chain Monte Carlo (MCMC) sampling process. The red point corresponds to the empirical minimum and maximum standard deviations computed from the original dataset.

Interpretation of the Scatterplot The majority of the posterior samples (blue points) form a dense cluster, indicating the range of predicted variability for within-person reaction times. However, the red point lies outside this main distribution, suggesting that the observed reaction time variability is greater than what the model predicts. This discrepancy indicates that the model may underestimate the true within-person variability of schizophrenic individuals.

Possible reasons for this underestimation include:

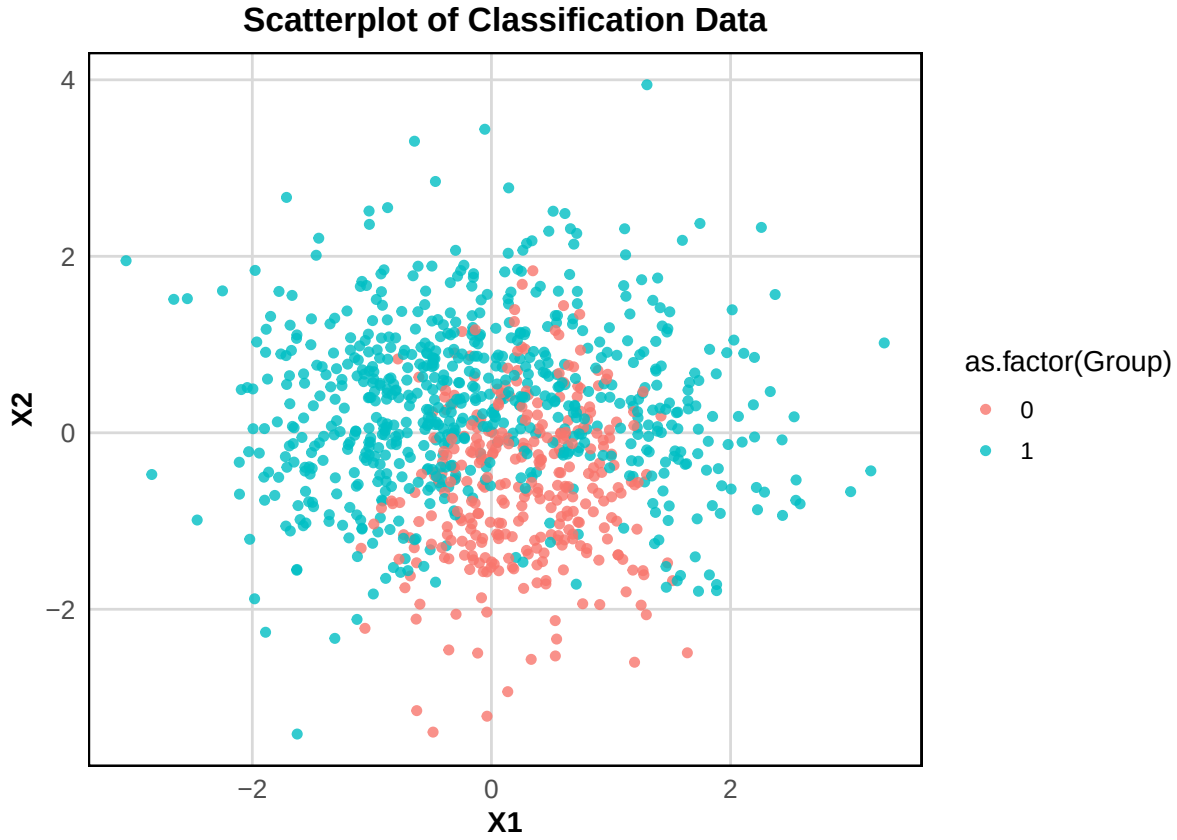
Restrictive prior assumptions on the variance parameter y , which may be limiting the spread of predicted reaction times. The mixture model structure may not fully capture the extent of individual differences in reaction time delays. Potential bias in the assumed normal distributions, which may not sufficiently accommodate extreme reaction times observed in schizophrenic individuals. Conclusion: Can the Model Explain the Variability? The positioning of the red point outside the cloud of posterior samples suggests that the model does not fully explain the observed variability in schizophrenic reaction times. While the model captures general trends, it likely underestimates reaction time dispersion. Refinements to the prior distributions, particularly on y , or adjustments to the model structure (such as incorporating a more flexible variance structure) may be necessary to improve model fit.

8. double check

B. Classification

The following figure shows the information in the dataset Classification.csv- it shows two different groups, plotted against two explanatory variables. This is simulated data - the groupings are determined by a known function of X1 and X2 with added noise/random error.

```
ggplot(Classification, aes(x = X1, y = X2, color = as.factor(Group))) +
  geom_point(alpha = 0.8) +
  labs(
    title = "Scatterplot of Classification Data",
    x = "X1",
    y = "X2"
  ) +
  custom_theme
```



1. Create meaningful summaries

First, we use the summary function to get a brief overview of the dataset. We can see that the X1 and X2 are the explanatory variables. To understand how the groups are classified, we will create a more detailed numerical summary.

```
summary(Classification)
```

```
gt_table_X1
```

Summary Statistics for Group 0 and Group 1

Comparison of X1 Across Groups

Group	Mean X1	SD X1	Median X1	Min X1	Max X1
Group 0	0.26	0.54	0.26	-1.09	1.64
Group 1	-0.12	1.12	-0.30	-3.06	3.29

Summary Statistics for Group 0 and Group 1

Comparison of X2 Across Groups

Group	Mean X2	SD X2	Median X2	Min X2	Max X2
Group 0	-0.60	0.89	-0.60	-3.39	1.84
Group 1	0.30	0.93	0.31	-3.41	3.94

```
gt_table_X2
```

Numerical Summaries and Interpretation

Variable X1:

- **Mean and Variability:** Group 1 shows higher variability in X1 (SD = 1.12) compared to Group 0 (SD = 0.59), indicating greater dispersion around its mean.
- **Median and Range**: The median of Group 1 (-0.30) is lower than that of Group 0 (0.06), suggesting a shift towards lower values for Group 1. Group 1 also exhibits more extreme values with a broader range (Min = -3.06, Max = 3.29) compared to Group 0 (Min = -1.09, Max = 1.64).

Summaries for Variable X2

X2 shows greater similarity between the two groups compared to X1: - **Central Tendency**: The median X2 value of Group 1 (0.31) is higher than Group 0 (-0.05), indicating that Group 1 typically has higher X2 values. - **Spread and Range:** Both groups have similar variability, indicated by their standard deviations (0.93 for Group 1 vs. 0.89 for Group 0). However, Group 1 again has a wider range (-3.41 to 3.94) compared to Group 0 (-3.39 to 1.84).

Implications for Classification

The observed numerical summaries and scatter plot highlight several key points:

- The differences in distributions and notable overlap suggest that the classes are not linearly separable.
- The variability, particularly in X1, suggests complex interactions or non-linear boundaries between groups.

Recommended Classification Methods

Given the distribution and structure of the data, suitable classification methods include:

1. **K-Nearest Neighbors (KNN):**
   - Flexible in handling non-linear boundaries.
   - The optimal choice of neighbors (K) can be tuned to balance flexibility and overfitting risks.
2. **Quadratic Discriminant Analysis (QDA):**

- Effective at modeling non-linear class boundaries if the assumption of normality holds.

3. **Random Forest (RF):**
   - Capable of capturing complex interactions and nonlinear patterns in the data without strong assumptions.
4. **Support Vector Machines (SVM):**
   - Suitable for data where clear linear boundaries do not exist, leveraging kernels to create non-linear separations.

Each of these methods accommodates the observed complexity and variability, offering a balanced approach between flexibility and generalizability.

2. Select75%ofthedatatoactasatrainingset,withtheremaining25%fortesting/evaluation.

```
library(dplyr)
 library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.3
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
 library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.4.3
```

```
 # Define function that will split the data into training and test sets
 trainTestSplit <- function(df, trainPercent,seed1){
 ## Sample size percent
 smp_size <- floor(trainPercent/100 * nrow(df))
 ## set the seed
 set.seed(seed1)
 train_ind <- sample(seq_len(nrow(df)), size = smp_size)
 train_ind
 }

#Split as training and test sets
 train_ind <- trainTestSplit (Classification, trainPercent = 75, seed = 123)
 train <- Classification[train_ind ,]
 test <- Classification[-train_ind ,]
```

The function **trainTestSplit** is employed to divide the dataset into training and test subsets to facilitate model evaluation in Part B. It randomly partitions the dataset based on a specified training percentage (**trainPercent**), with a fixed random seed (**seed1**) to ensure reproducibility. The training subset is used for model fitting, while the test subset provides an unbiased measure of predictive performance. This approach ensures a rigorous assessment of model generalization to new data.

3.

K-Nearest Neighbors (KNN):

```
library(class) # This is the package which has the KNN function
```

theory behind how the classification method classifies the data:

present the results of an evaluation of the method: Describe findings:

Detailed description of model performance:

optimize the (hyper) parameters of the method.

Quadratic Discriminant Analysis (QDA):

Random Forest (RF):

Support Vector Machines (SVM):