

Bayes Analysis Assignment 2025

by

Myeni L (MYNLUN004), Mthetho K (MTHKWE006)



September 6, 2025


Plagiarism declaration

Our group, **Lungani Myeni, Kwetsi Mthetho** Hereby declare that the work presented in this assignment for the course Experimental Design is entirely our own. We confirm that:

1. We recognise that plagiarism is a serious form of academic dishonesty.
2. Our work has not been previously submitted in whole or in part, for any other course or assessment.
3. We have acknowledged and referenced all the sources of information used in this assignment.
4. We have abided by all ethical guidelines for academic integrity as outlined by the university.

A handwritten signature in black ink, consisting of a large, stylized capital 'L' followed by a horizontal line extending to the right.

Myeni L.

A handwritten signature in black ink, featuring a dense, circular scribble of lines with a small mark to the right.

Mthetho K.

Contents

Question 1	3
Animal Data	3
Determining the Posterior	3
Posterior Plots	5
Question 2	6
The Algorithm	6
Sampling from Candidate Distribution	8
Posterior Draws Summaries and Plots	10
Question 3	11
Expectation	11
Variance	11
Appendix	12
Source Code	12

List of Figures

1	Observed Movement Path	3
2	Posterior Distribution Versions	5
3	Natural Log Maximisation	6
4	Gamma Function	7
5	Typical Candidate Samples	9
6	Contrasting Posterior Draws Density Against Theoretical	10

List of Tables

1	Summary of Posterior Samples for Draws	10
---	--	----

Question 1

Animal Data

The data of this assignment is based on **STA3043_Assignment1_2025.RData** and specifically, the data of Lungani Myeni (MYNLUN004).

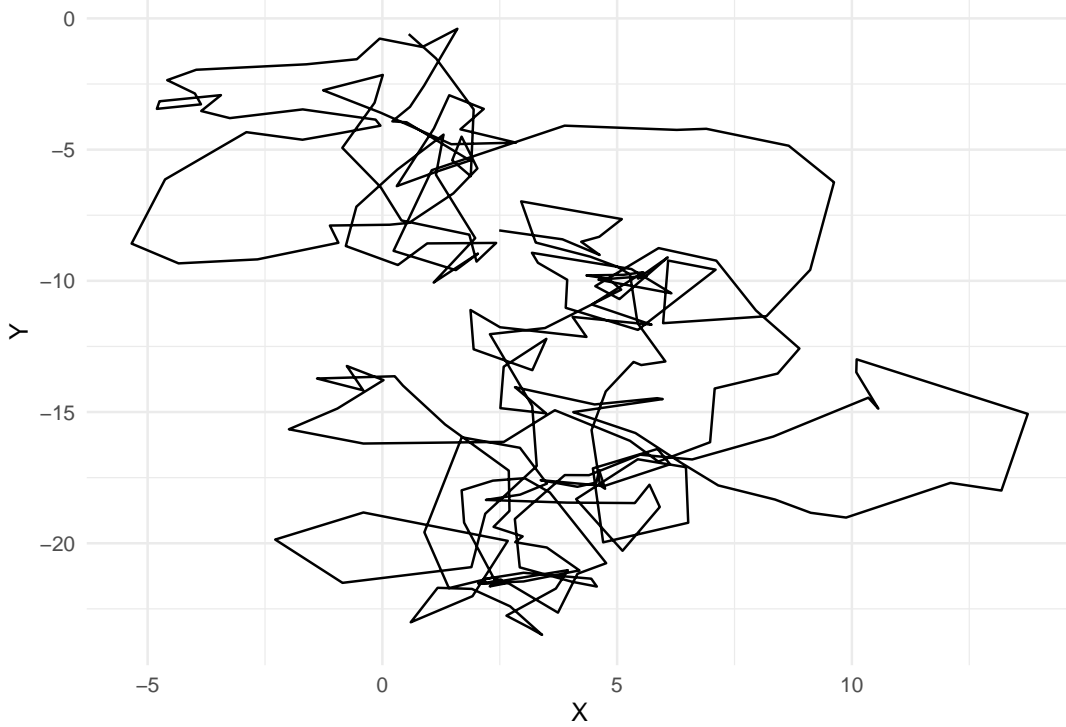


Figure 1: Observed Movement Path

Determining the Posterior

We would like to find the posterior distribution of $\theta|V$ where θ is the unknown angle of rotation.

$$\pi(\theta|V) \propto \pi(\theta)L(\theta|V)$$

where $\pi(\theta)$ is the prior distribution of θ and $L(\theta|V)$ is the likelihood function of θ given V .

Assuming a uniform prior distribution for θ over the interval $[-\pi, \pi]$, we have that:

$$\pi(\theta|V) \propto L(\theta|V)$$

Then, we need to find the likelihood function $L(\theta|V)$. Note that:

$$v_t = aMv_{t-1} + e_t$$

where $a = 0.5$ and $e_t \sim \text{MVN}(0, I_2)$ iid. and M is a matrix given by:

$$M = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

Now,

$$\begin{aligned}
 L(\theta|V) &= \prod_{t=1}^n \exp \left\{ -\frac{1}{2} (v_i - aMv_{i-1})^\top (v_i - aMv_{i-1}) \right\} \\
 &= \exp \left\{ -\frac{1}{2} \sum_{t=1}^n (v_i - aMv_{i-1})^\top (v_i - aMv_{i-1}) \right\} \\
 &= \exp \left\{ -\frac{1}{2} \sum_{t=1}^n A \right\}
 \end{aligned}$$

We evaluate the quadratic form inside the summation separately:

$$\begin{aligned}
 A &= (v_i - aMv_{i-1})^\top (v_i - aMv_{i-1}) \\
 &= v_i^\top v_i - 2av_i^\top Mv_{i-1} + a^2 v_{i-1}^\top M^\top M v_{i-1} \\
 &= v_i^\top v_i - 2av_i^\top Mv_{i-1} + a^2 v_{i-1}^\top I_2 v_{i-1} \\
 &= v_i^\top v_i - 2av_i^\top Mv_{i-1} + a^2 v_{i-1}^\top v_{i-1}
 \end{aligned}$$

Notice that $v_i = \begin{pmatrix} x_i - x_{i-1} \\ y_i - y_{i-1} \end{pmatrix}$ which can be expressed as $v_i = \begin{pmatrix} \Delta x_i \\ \Delta y_i \end{pmatrix}$.

Using that we can express $v_i^\top v_i$ as:

$$\begin{aligned}
 v_i^\top v_i &= (\Delta x_i, \Delta y_i) \begin{pmatrix} \Delta x_i \\ \Delta y_i \end{pmatrix} \\
 &= \Delta x_i^2 + \Delta y_i^2
 \end{aligned}$$

Moreover, we can express $v_i^\top Mv_{i-1}$ as:

$$\begin{aligned}
 v_i^\top Mv_{i-1} &= (\Delta x_i, \Delta y_i) \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \Delta x_{i-1} \\ \Delta y_{i-1} \end{pmatrix} \\
 &= (\Delta x_i, \Delta y_i) \begin{pmatrix} \Delta x_{i-1} \cos \theta - \Delta y_{i-1} \sin \theta \\ \Delta x_{i-1} \sin \theta + \Delta y_{i-1} \cos \theta \end{pmatrix} \\
 &= \Delta x_i \Delta x_{i-1} \cos \theta - \Delta x_i \Delta y_{i-1} \sin \theta + \Delta y_i \Delta x_{i-1} \sin \theta + \Delta y_i \Delta y_{i-1} \cos \theta \\
 &= (\Delta x_i \Delta x_{i-1} + \Delta y_i \Delta y_{i-1}) \cos \theta + (\Delta y_i \Delta x_{i-1} - \Delta x_i \Delta y_{i-1}) \sin \theta
 \end{aligned}$$

Finally, we have that:

$$\begin{aligned}
 A &\propto -2a [(\Delta x_i \Delta x_{i-1} + \Delta y_i \Delta y_{i-1}) \cos \theta + (\Delta y_i \Delta x_{i-1} - \Delta x_i \Delta y_{i-1}) \sin \theta] \\
 &= -2a [s_1 \cos \theta + s_2 \sin \theta]
 \end{aligned}$$

Hence, we can express the likelihood function as:

$$\begin{aligned}
 L(\theta|V) &\propto \exp \left\{ -\frac{1}{2} \sum_{t=1}^n -2a [s_1 \cos \theta + s_2 \sin \theta] \right\} \\
 &= \exp \left\{ \sum_{t=1}^n [s_1 \cos \theta + s_2 \sin \theta] \right\}
 \end{aligned}$$

where $s_1 = a(\Delta x_i \Delta x_{i-1} + \Delta y_i \Delta y_{i-1})$ and $s_2 = a(\Delta y_i \Delta x_{i-1} - \Delta x_i \Delta y_{i-1})$.

Finally, we have that the posterior distribution of $\theta|V$ is:

$$\pi(\theta|V) \propto \exp \left\{ \frac{s_1 \cos \theta + s_2 \sin \theta}{2} \right\}$$

Posterior Plots

Below we have plots of the posterior distributions based on our data.

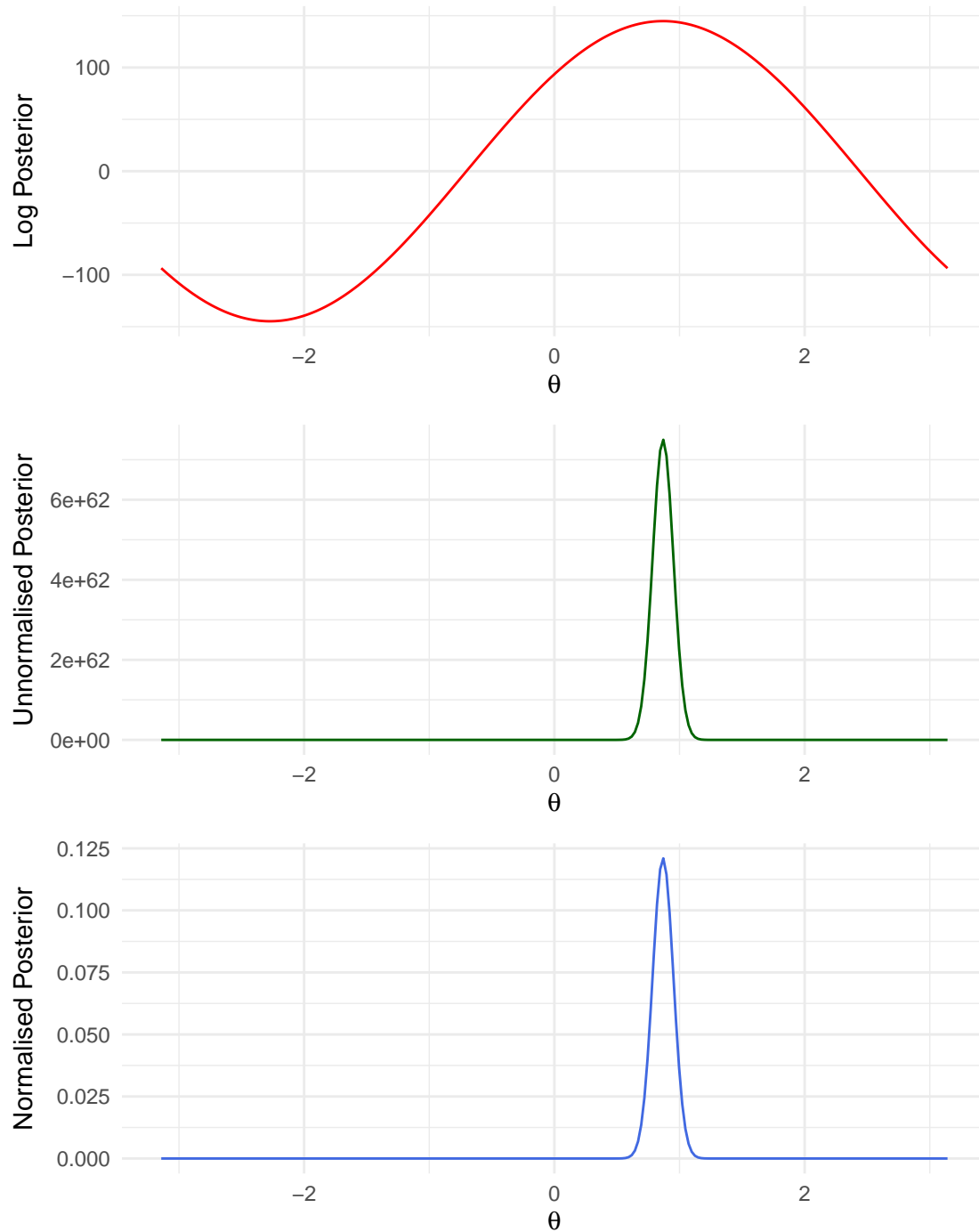


Figure 2: Posterior Distribution Versions

Question 2

The Algorithm

It was found that the posterior distribution of $\theta|V$ had the following form:

$$\pi(\theta|V) \propto \exp \left\{ \frac{s_1 \cos \theta + s_2 \sin \theta}{2} \right\}$$

for $\theta \in [-\pi, \pi]$ where s_1, s_2 are constants that depend on the data V .

Note: for ease of notation,
Let $\frac{s_1 \cos \theta + s_2 \sin \theta}{2} = f(\theta)$

We would like to now find posterior samples of $\theta|V$ using the *Acceptance-Rejection* algorithm. We are given a suitable candidate distribution $h(\theta)$ such that:

$$h(\theta) = \exp \{3 \cos \theta\}$$

for $\theta \in [-\pi, \pi]$ and $h(\theta) = 0$ otherwise.

The algorithm requires an expression for C such that:

$$\gamma(\theta) = \frac{f(\theta)}{C h(\theta)}$$

where $C = \max_{\Theta} \left\{ \frac{f(\theta)}{h(\theta)} \right\}$, $\Theta = [-\pi, \pi]$

using the explanatory advantage we have with R, we can easily find that C by using `optimise` function in R to find that:

$$C \approx 1.0993194 \times 10^{62}$$

Visually, we can see the location of the natural log C as follows:

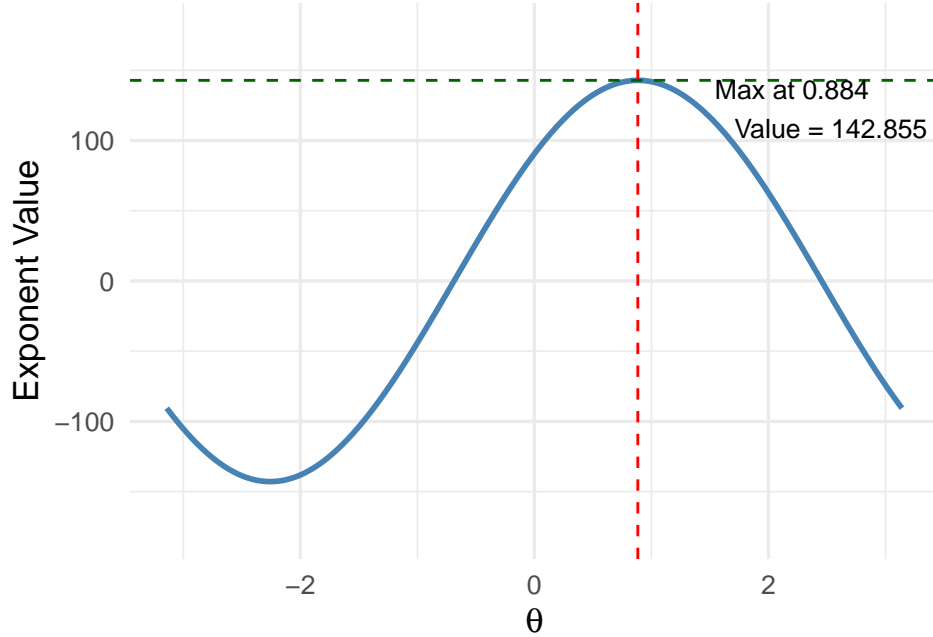


Figure 3: Natural Log Maximisation

We are now able to additionally plot the function $\gamma(\theta)$ as follows:

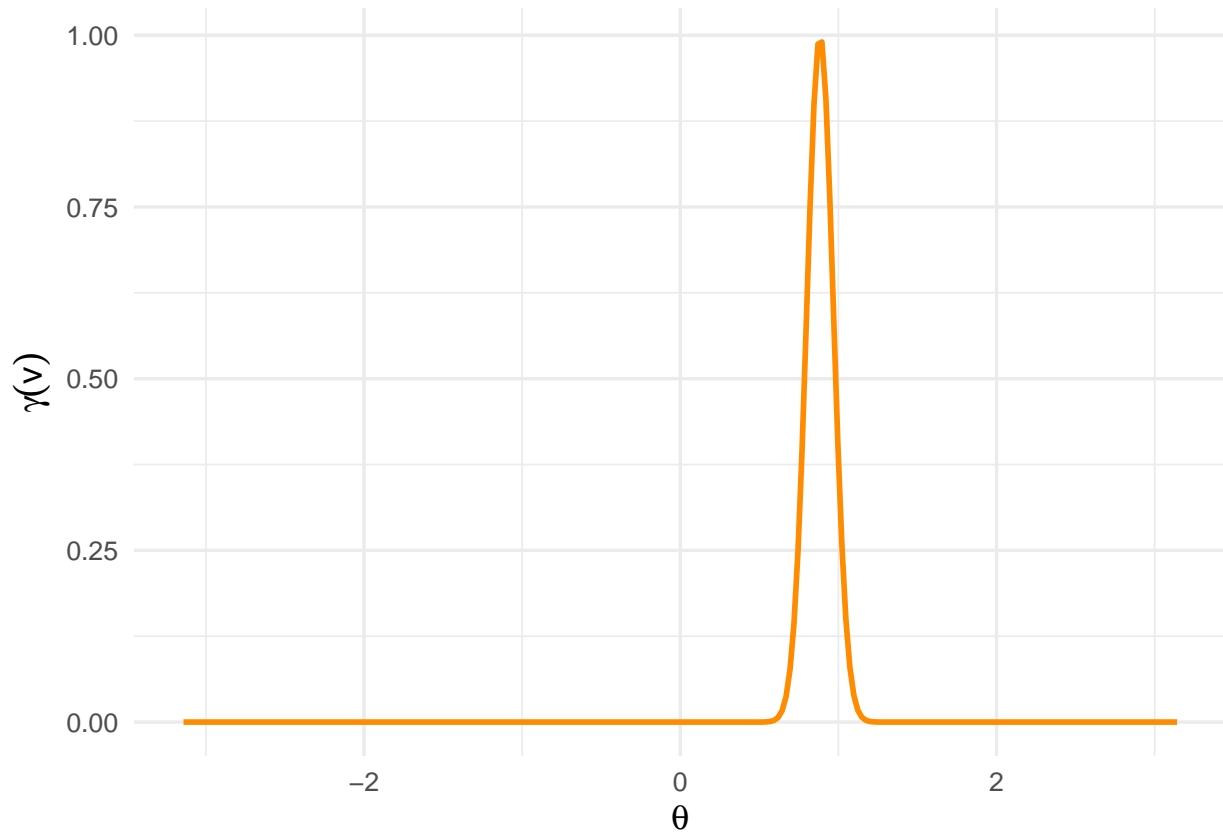


Figure 4: Gamma Function

Sampling from Candidate Distribution

The general procedure we take in finding draws from the posterior distribution of $\theta|V$ using the acceptance-rejection algorithm goes something:

1. Sample θ^* 's from $h(\theta)$
2. Sample U 's from $Unif(0, 1)$
3. If $U \leq \gamma(\theta^*)$, accept θ^* as a draw from the posterior distribution of $\theta|V$. Otherwise, reject and return to step 1.
4. Repeat until you have the desired number of posterior draws, in our case 5000.

A road block was found in step 1, as we needed to find a way to sample from $h(\theta)$. We decided to get draws of θ^* 's using an additional acceptance-rejection algorithm, using a uniform distribution as the candidate distribution!

So indeed call $g(\theta)$ the candidate distribution, where is naturally:

$$g(\theta) = \frac{1}{2\pi}$$

for $\theta \in [-\pi, \pi]$ and $g(\theta) = 0$ otherwise.

In this case, we need to find M as our interim constant such that:

$$M = \max_{\Theta} \left\{ \frac{h(\theta)}{g(\theta)} \right\}$$

which we determine to be $M = 2\pi e^3$ by inspection as cosine has a maximum value of 1.

Consequently, our γ in this case is:

$$\gamma(\theta) = \frac{h(\theta)}{Mg(\theta)} = \frac{\exp\{3 \cos \theta\}}{2\pi e^3 \times \frac{1}{2\pi}}$$

hence, we can obtain draws from $h(\theta)$ using the algorithm described above and get draws that will generally look like the below:

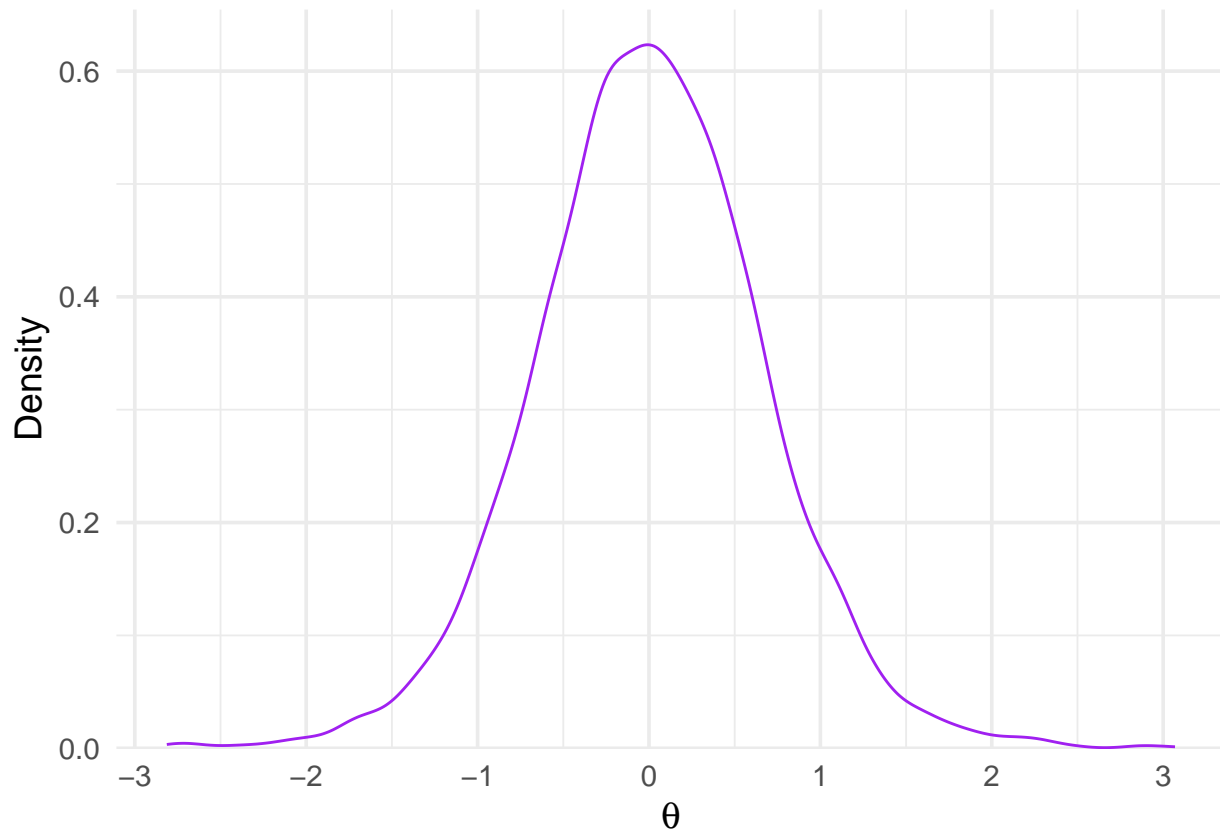


Figure 5: Typical Candidate Samples

Now then, we have a way to generate draws from $h(\theta)$, we can now implement the original acceptance-rejection algorithm to get draws from the posterior distribution of $\theta|V$. So practically, the implementation into R would go something like this with the help of a constructed function called `ar`:

```
ar = function (m){
  draw = numeric(0)
  while (length(draw) < m){
    k = m - length(draw)
    # proposed thetas from candidate h(theta)
    prop2 = theta_samps (k)
    # random uniform generation from 0,1
    unif2 = runif (k)
    # accepted proposals condition, using log acceptance
    logaccept = expo_gamma (prop2)
    accept2 = log (unif2) <= logaccept
    # updating draw
    draw = c(draw, prop2[accept2])
  }
  # returning final posterior draws
  draw
}
```

Of note, the function, `ar`, used log acceptance for general efficiency because repeated iterations took quite a while because of the a) the nested acceptance-rejection algorithm and b) the fact that the large C influences the acceptance rate.

Posterior Draws Summaries and Plots

The current situation is that we have all the variables known or with a means to get them. We can implement the acceptance-rejection algorithm to finally sample from $\theta|V$.

After implementing the algorithm, we get the following summaries:

Table 1: Summary of Posterior Samples for Draws

Summary	Values
Minimum	0.5706
1st Quartile	0.8099
Median	0.8689
3rd Quartile	0.8669
Maximum	0.9243

Moreover, we have the following graphical representation of the posterior draws distribution for $\theta|V$. We additionally plot the trace plot of the sampled density and the theoretical density for comparison.

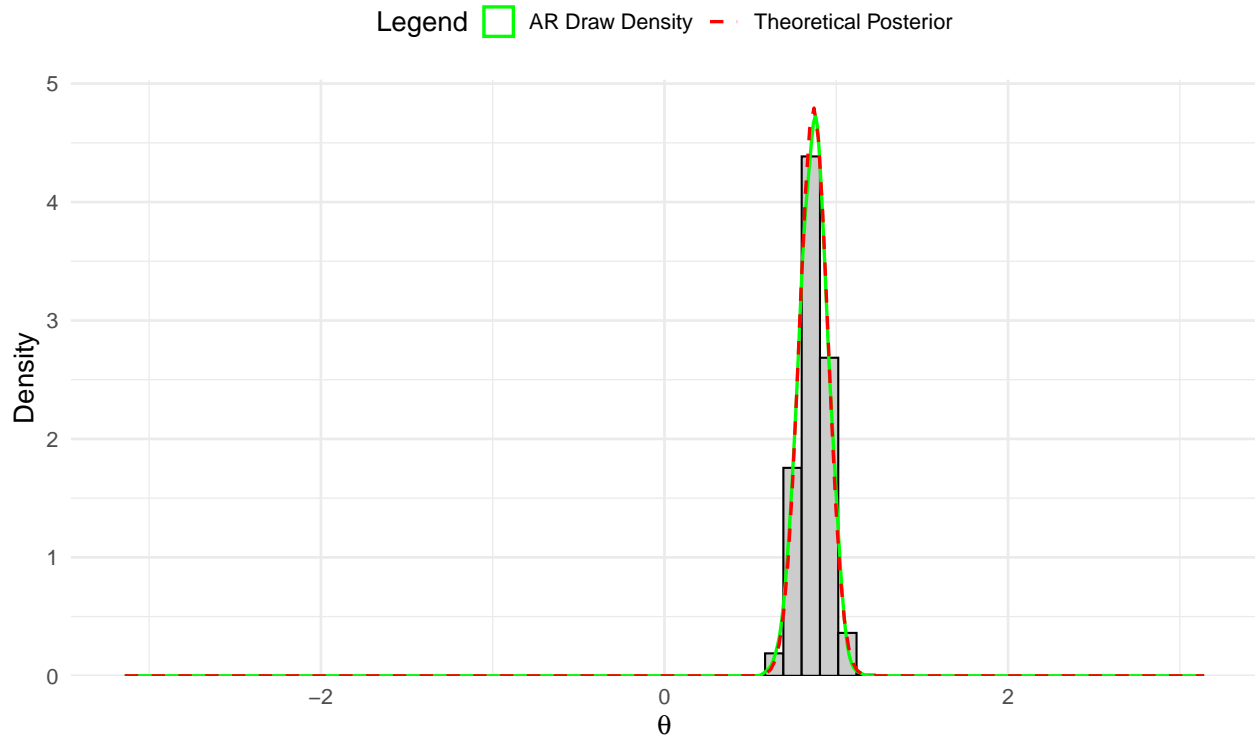


Figure 6: Contrasting Posterior Draws Density Against Theoretical

Question 3

Expectation

We want to find $\mathbb{E}(\mu_{251}|V)$ and $\text{Var}(\mu_{251}|V)$, using our posterior distribution draws of theta.

Note:

$$\mu_{n+1}|V$$

has unknown distribution but we know that the posterior distribution is such that

$$\mu_{n+1}|V, \theta \sim \text{MVN}(\mu_n + aM(\mu_n - \mu_{n-1}), I_2)$$

hence, by the law of total expectation, we have that:

$$\begin{aligned}\mathbb{E}(\mu_{251}|V) &= \mathbb{E}(\mathbb{E}(\mu_{251}|V, \theta)) \\ &= \mathbb{E}(\mu_{250} + aM(\mu_{250} - \mu_{249}))\end{aligned}$$

Now to find the expectation above, we make use of *Monte Carlo Integration*, i.e. we use the posterior draws to estimate the expectation as follows:

$$\mathbb{E}(\mu_{251}|V) \approx \frac{1}{N} \sum_{i=1}^N (\mu_{250} + aM_{(i)}(\mu_{250} - \mu_{249}))$$

where N is the number of posterior draws of θ . In our case, $N = 5000$.

This is easily implemented in R to find that the 251th true location estimate is:

$$\mathbb{E}(\mu_{251}|V) \approx \begin{pmatrix} 2.1823 \\ -7.4577 \end{pmatrix}$$

Variance

Now to find the variance, we make use of the law of total variance:

$$\begin{aligned}\text{Var}(\mu_{251}|V) &= \mathbb{E}(\text{Var}(\mu_{251}|V, \theta)) + \text{Var}(\mathbb{E}(\mu_{251}|V, \theta)) \\ &= \mathbb{E}(0) + \text{Var}(\mu_{250} + aM(\mu_{250} - \mu_{249}))\end{aligned}$$

where we used the fact that $\text{Var}(\mu_{251}|V, \theta) = 0$ since $\mu_{251}|V, \theta$ is MVN with identity covariance matrix.

So the job of finding the variance reduces to finding the variance of $\mu_{250} + aM(\mu_{250} - \mu_{249})$ which we can also estimate using Monte Carlo integration as follows:

Note: for ease of notation,
Let $X(\theta) = \mu_{250} + aM(\mu_{250} - \mu_{249})$ and
Let $\bar{X} = \frac{1}{M} \sum_{i=1}^M X(\theta_{(i)})$

$$\begin{aligned}\text{Var}(\mu_{251}|V) &= \text{Var}(X(\theta)) \\ &\approx \frac{1}{M-1} \sum_{i=1}^M (X(\theta_{(i)}) - \bar{X})(X(\theta_{(i)}) - \bar{X})^\top\end{aligned}$$

or equivalently:

$$\begin{aligned}\text{Var}(\mu_{251}|V) &= \mathbb{E}((X(\theta))(X(\theta))^\top) - \mathbb{E}(X(\theta))^2 \\ &\approx \frac{1}{M} \sum_{i=1}^M ((X(\theta_{(i)}))(X(\theta_{(i)}))^\top) - ((\bar{X})(\bar{X})^\top)\end{aligned}$$

which can also be easily implemented in R to find that the variance of the 251th true location estimate is:

$$\text{Var}(\mu_{251}|V) \approx \begin{pmatrix} 0.0027 & 0.0013 \\ 0.0013 & 7 \times 10^{-4} \end{pmatrix}$$

Appendix

Source Code

```
knitr::opts_chunk$set(echo = TRUE)
# Install ggplot2 if you don't have it installed already
if (!requireNamespace("ggplot2", quietly = TRUE)) {
  install.packages("ggplot2")
}
# Load ggplot2
library(ggplot2)
knitr::opts_chunk$set(echo=TRUE)
if (!requireNamespace("dplyr", quietly = TRUE)) {
  install.packages("dplyr")
}
library(dplyr)
knitr::opts_chunk$set(echo=TRUE)
if (!requireNamespace("patchwork", quietly = TRUE)) {
  install.packages("patchwork")
}
library(patchwork) # for nice plot combining
knitr::opts_chunk$set(echo=TRUE)
if (!requireNamespace("broom", quietly = TRUE)) {
  install.packages("broom")
}
library(broom)
knitr::opts_chunk$set(echo=TRUE)
if (!requireNamespace("formatR", quietly = TRUE)) {
  install.packages("formatR")
}
library(formatR)
knitr::opts_chunk$set(echo = TRUE)
if (!requireNamespace("knitr", quietly = TRUE)) {
  install.packages("knitr")
}
library(knitr)
if (!requireNamespace("kableExtra", quietly = TRUE)) {
  install.packages("kableExtra")
}
library(kableExtra)
knitr::opts_chunk$set(echo=TRUE)
if (!requireNamespace("car", quietly = TRUE)) {
  install.packages("car")
}
library(car)
knitr::opts_chunk$set(echo=TRUE)
if (!requireNamespace("latex2exp", quietly = TRUE)) {
  install.packages("latex2exp")
}
library(latex2exp)
knitr::opts_chunk$set(echo=TRUE)
if (!requireNamespace("pracma", quietly = TRUE)) {
  install.packages("pracma")
}
```

```

library(pracma)

options(show.signif.stars = FALSE)
rm (list=ls())

# loading the assignment data
load ("STA3043_Assignment1_2025.RData")

# retrieval of student observations
kwe_obs = Class.List$MTHKWE006
lun_obs = Class.List$MYNLUN004
x = lun_obs

x_obs = x[,2]
y_obs = x[,3]
df = data.frame(x = x_obs, y = y_obs)

ggplot(df, aes(x = x, y = y)) +
  geom_path(color = "black") +
  theme_minimal() +
  labs(x = TeX("X"), y = TeX("Y"))
# computing s1 and s2
# getting the differences of points and isolating a vector for both change in
# x and change in y
diff_x = diff (x)
dx = diff_x[,2]
dy = diff_x[,3]
# need lagged difference of points
dx_LAG = c(0, head(dx, -1))
dy_LAG = c(0, head(dy, -1))
# so that
s1 = sum (dx*dx_LAG + dy*dy_LAG)
s2 = sum (dx*dy_LAG - dy*dx_LAG)

# exponent of function for posterior
expo_postv = function(theta){
  # combining the s1,s2 with trig functions
  x1 = s1*cos(theta) + s2*sin(theta)
  0.5*x1
}

# vectorise function above to CORRECTLY handle the theta vector
vec_expo_postv = Vectorize (FUN = expo_postv, vectorize.args = "theta")

# theta's range
theta_range = seq (from = -pi, to = pi, length = 250)

# data prep
log_post_vals <- vec_expo_postv(theta_range)
unnorm_post_vals <- exp(log_post_vals)

norm_post_vals <- unnorm_post_vals / sum(unnorm_post_vals)

```

```

# put into one dataframe for plotting
df <- data.frame(
  theta = theta_range,
  log_post = log_post_vals,
  unnorm_post = unnorm_post_vals,
  norm_post = norm_post_vals
)

# make three ggplots
p1 <- ggplot(df, aes(x = theta, y = log_post)) +
  geom_line(color = "red") +
  labs(x = TeX("$\\theta$"), y = TeX("Log Posterior")) +
  theme_minimal()

p2 <- ggplot(df, aes(x = theta, y = unnorm_post)) +
  geom_line(color = "darkgreen") +
  labs(x = TeX("$\\theta$"), y = TeX("Unnormalised Posterior")) +
  theme_minimal()

p3 <- ggplot(df, aes(x = theta, y = norm_post)) +
  geom_line(color = "royalblue") +
  labs(x = TeX("$\\theta$"), y = TeX("Normalised Posterior")) +
  theme_minimal()

# combine with patchwork (1 row, 3 cols)
#(p1 | p2 | p3)
# OR (3 rows, 1 col)
p1 / p2 / p3 # + plot_layout(heights = c(10, 10, 10)) # 1.5× the default height

# want value of C
# function (ln component, exponent) to maximise
expo = function (theta){
  # the previous work:
  x1 = s1*cos(theta) + s2*sin(theta)
  # candidate sampling distr
  x2 = cos(theta)
  0.5*x1 -3*x2
}

# vectorise the function to CORRECTLY deal with vector val of theta
vec_expo = Vectorize (FUN = expo, vectorize.args = "theta")

# tool to find maximum of exponent component
opt = optimise(vec_expo, interval = c(-pi,pi), maximum = TRUE)

# hence value of C the normalising constant
C = exp(opt$objective)
C_tex = round(C, 8)

# Data for plotting
theta_range <- seq(-pi, pi, length.out = 250)
df_max <- data.frame(
  theta = theta_range,

```

```

    val = vec_expo(theta_range)
  )

  # Extract maximum location & value
  theta_max <- opt$maximum
  val_max <- opt$objective

  # Plot
  ggplot(df_max, aes(x = theta, y = val)) +
    geom_line(color = "steelblue", size = 1) +
    geom_vline(xintercept = theta_max, linetype = "dashed", color = "red") +
    geom_hline(yintercept = val_max, linetype = "dashed", color = "darkgreen") +
    annotate("text", x = theta_max, y = val_max,
             label = paste0("Max at ", round(theta_max,3),
                            "\nValue = ", round(val_max,3)),
             hjust = -0.5, vjust = 1, size = 3.5) +
    labs(
      x = TeX("$\\theta$"),
      y = "Exponent Value"
    ) +
    ylim (-180,180) +
    theme_minimal(base_size = 13)
gamma = function(theta, normconstant=C){
  # the previous work:
  x1 = s1*cos(theta) + s2*sin(theta)
  x2 = cos(theta)
  C = normconstant

  (exp(0.5*x1 -3*x2))/(C)
}

# vectorise the function to CORRECTLY deal with vector val of theta
vec_gamma = Vectorize (gamma, "theta")

gamma_df <- data.frame(
  theta = theta_range,
  gamma_val = vec_gamma(theta_range)
)

ggplot(gamma_df, aes(x = theta, y = gamma_val)) +
  geom_line(color = "darkorange", size = 1) +
  labs(
    x = TeX("$\\theta$"),
    y = TeX("$\\gamma(v)$")
  ) +
  theme_minimal(base_size = 13)

# since h(theta) isnt a known distribution we sample by AR too where
# our candidate is a uniform
theta_samps = function (n){
  out = numeric (0)
  while (length(out) < n){
    # want to sample the difference between length of samples and desired n

```



```

    m = n - length(out)
    # getting the proposed values using the candidate of uniform -pi,pi
    prop1 = runif (m, -pi, pi)
    # getting the random uniform 0,1 values
    unif1 = runif (m, 0, 1)
    # accept if unif1 < gamma(prop1), note this is indeed gamma because
    #  $C = 2\pi e^{-3}$ 
    accept1 = unif1 <= exp(3*(cos(prop1)-1))
    # updating out
    out = c(out, prop1[accept1])
  }

  #return the samples from h(theta)
  out[1:n]
}

# function named ar to spit out draws BUT first
# need log of gamma to allow for log-scale acceptance since C is LARGE
ln_C = log(C)
expo_gamma = function(theta, normconstant=ln_C){
  # the previous work:
  x1 = s1*cos(theta) + s2*sin(theta)
  x2 = cos(theta)
  A = normconstant

  0.5*x1 -3*x2 - A
}

# getting theta draws
candi_draws = theta_samps(5000)
# convert to df
df_candi_draws = data.frame(candi_draws = candi_draws)

# ggplot
ggplot() +
  # smoothed kernel density of draws
  geom_density(data = df_candi_draws, aes(x = candi_draws),
    color = "purple", size = 0.5) +
  labs(
    x = TeX("$\\theta$"),
    y = "Density"
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    legend.position = "none"
  )
}

ar = function (m){
  draw = numeric(0)
  while (length(draw) < m){
    k = m - length(draw)
    # proposed thetas from candidate h(theta)
    prop2 = theta_samps (k)

```

```

# random uniform generation from 0,1
unif2 = runif (k)
# accepted proposals condition, using log acceptance
logaccept = expo_gamma (prop2)
accept2 = log (unif2) <= logaccept
# updating draw
draw = c(draw, prop2[accept2])
}
# returning final posterior draws
draw
}
# picking our number of samples to get back from ar()
n1 = 5000
samps = ar (n1)

# to get summary of posterior samples
post_summary = summary(samps)
# convert to data frame for nicer formatting
post_summary_df = data.frame(
  Summary = c("Minimum", "1st Quartile", "Median", "3rd Quartile", "Maximum"),
  Values = as.numeric(post_summary[c(1,2,3,4,5)])
)

# display as a kable
kable(post_summary_df,
      caption = "Summary of Posterior Samples for Draws",
      digits = 4,
      format = "pipe")

# theoretical
# recompute theoretical posterior density properly normalized
df$norm_post <- unnorm_post_vals / trapz(df$theta, unnorm_post_vals)

# convert AR draws into a data frame
df_samps <- data.frame(samps = samps)

# ggplot
ggplot() +
  # histogram of AR draws
  geom_histogram(data = df_samps, aes(x = samps, y = ..density..),
                bins = 60, fill = "grey80", color = "black") +
  # smoothed kernel density of draws
  geom_density(data = df_samps, aes(x = samps, color = "AR Draw Density"),
              size = 0.8) +
  # theoretical posterior overlay
  geom_line(data = df, aes(x = theta, y = norm_post, color = "Theoretical Posterior"),
           linetype = "dashed",
           size = 0.8) +
  labs(
    x = TeX("$\\theta$"),
    y = "Density",
    color = "Legend", # legend title for lines
    fill = "Legend"   # legend title for histogram
  ) +

```

```

scale_fill_manual(values = c("Histogram" = "grey80")) +
scale_color_manual(values = c(
  "AR Draw Density" = "green",
  "Theoretical Posterior" = "red"
)) +
scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +
theme_minimal(base_size = 14) +
theme(
  plot.title = element_text(hjust = 0.5, face = "bold"),
  legend.position = "top"
)

# We do this by using Monte Carlo
# 1st find f(theta) which will be the conditional of estimate mu given V and
# theta

# fuction to take in theta sample and data to set monte carllo for est of mu_251
f_theta = function (theta_samp, dataobs=x){
  # getting v_250 and mu_250
  mu_249 = dataobs[250,2:3]
  mu_250 = dataobs[251,2:3]

  # defining matrix M
  M = function (theta_samp){
    rbind ( cbind (cos(theta_samp), sin(theta_samp)),
            cbind (-sin(theta_samp), cos(theta_samp)) )
  }

  mu_250 + 0.5 * M(theta_samp) %*% (mu_250-mu_249)
}

# get expectation and variance by for loops
est_Exp = 0
sum1 = 0
for (i in 1:n1) {
  sum1 = sum1 + f_theta(samps[i])
  est_Exp = (1/n1) * (sum1)
}

mu_1 = round (est_Exp[1,], 4)
mu_2 = round (est_Exp[2,], 4)

est_Var = 0
sum2 = 0
for (i in 1:n1) {
  sum2 = sum2 + (f_theta(samps[i])) %*% t(f_theta(samps[i]))
  est_Var = (1/n1) * (sum2) - (est_Exp) %*% t(est_Exp)
}

cv_11 = round (est_Var[1,1], 4)
cv_12 = round (est_Var[1,2], 4)
cv_21 = round (est_Var[2,1], 4)
cv_22 = round (est_Var[2,2], 4)

```