Assignment 4

```
#------------- Useful functions -------------
mean_pdf   <- function(f, lower, upper){integrate(function(x) x*f(x), lower, upper)$value}
std_pdf <- function(f, lower, upper) {
  mu <- mean_pdf(f, lower, upper)
  sqrt(integrate(function(x) (x - mu)^2 * f(x), lower, upper)$value / integrate(f, lower, upper)$value
}
cumulative <- function(f, lower, X){integrate(f, lower, X,stop.on.error = FALSE)$value}
inverse_cumulative <- function(f, p, lower, upper){uniroot(function(x) cumulative(f, lower, x)-p, c(lo

#inference functions
binom_likelihood <- function(prob, ...) sapply(prob, function(P)  prod(dbinom(prob=P, ...)))
pois_likelihood  <- function(mu, ...)   sapply(mu,    function(MU) prod(dpois (lambda = MU, ...)))
norm_likelihood  <- function(mu, ...)   sapply(mu,    function(MU) prod(dnorm (mean = MU, ...) ))

posterior <- function(parameter, prior, likelihood, lower, upper, ...) {
  unnormalized <- function(x) likelihood(x, ...)*prior(x)
  norm_factor  <- integrate(unnormalized, lower = lower, upper = upper)$value
  unnormalized(parameter)/norm_factor
}
```

Exercise 1: Screening of disease in blood

Scenario

A well established and diffused method for detecting a disease in blood fails to detect the presence of disease in 15% of the patients that actually have the disease.

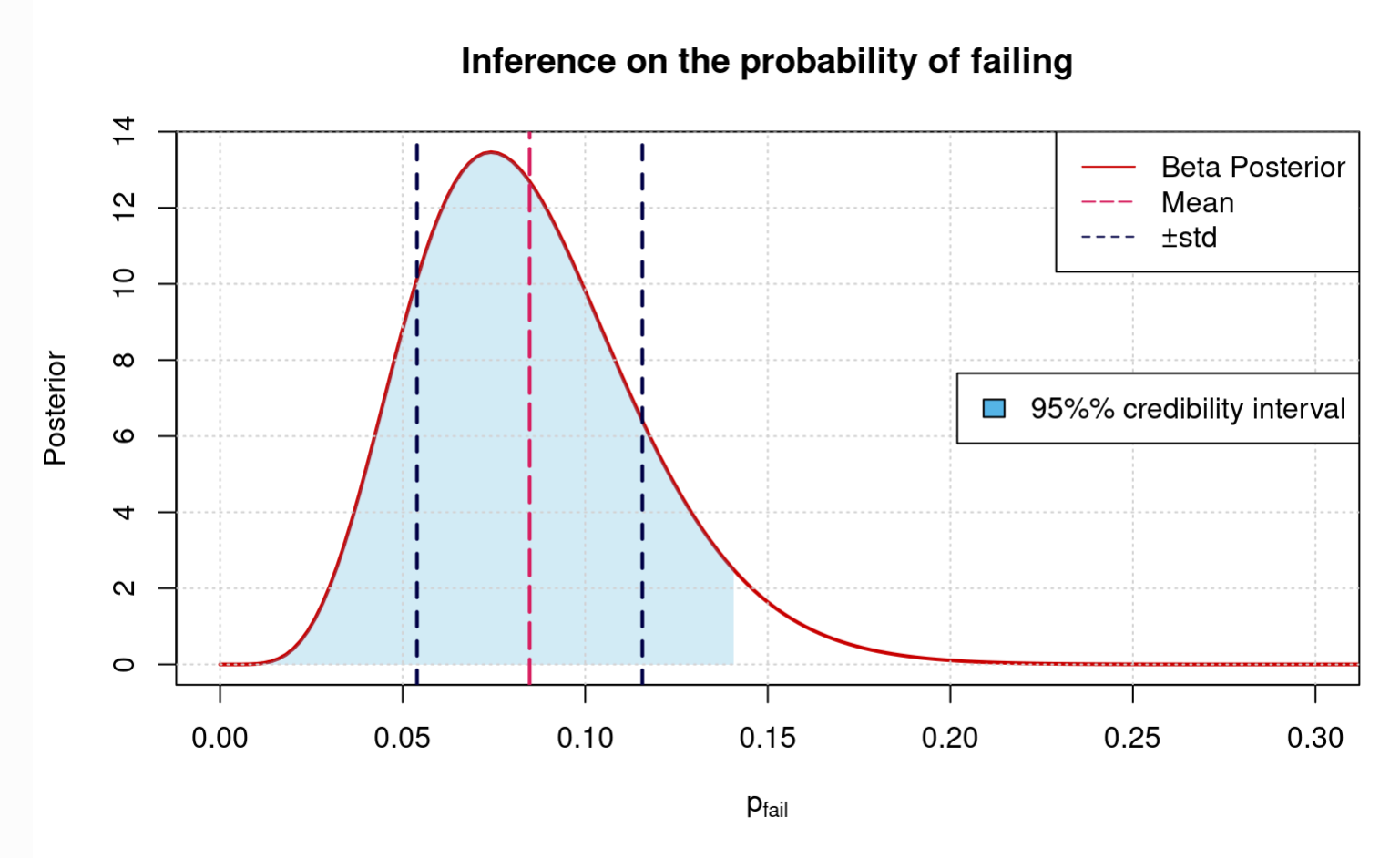
A young UniPD startUp has developed an innovative method of screening. During the qualification phase, a random sample of n = 75 patients known to have the disease is screened using the new method.

1. what is the probability distribution of y, the number of times the new method fails to detect the disease ?
2. on the n = 75 patients sample, the new method fails to detect the disease in y = 6 cases. What is the frequentist estimator of the failure probability of the new method ?
3. setup a bayesian computation of the posterior probability, assuming a beta distribution with mean value 0.15 and standard deviation 0.14. Plot the posterior distribution for y, and mark on the plot the mean value and variance
4. Perform a test of hypothesis assuming that if the probability of failing to the detect the desease in ill patients is greater or equal than 15%, the new test is no better that the traditional method. Test the sample at a 5% level of significance in the Bayesian way.
5. Perform the same hypothesis test in the classical frequentist way.

Answers

1. y follows a binomial distribution $P_n(y)$ with n being the number of patients and p the probability of failing the detection.
2. Since $\mathbb{E}[y] = p * n$, the frequentist estimator is $p = \frac{6}{75} = 0.08$.
3. The parameters for the beta prior are $\alpha \approx 0.83$ and $\beta \approx 4.68$. Since the beta distribution is the conjugate prior for the binomial likelihood, the parameters of the beta posterior are: $\alpha' = \alpha + y = 6.83$ and $\beta' = \beta + n - y = 73.68$.
4. Looking at the plot we can accept at a 5% level of significance that the new method is better than the old one.

```
#------------- SETTING THE INPUT HYPOTESIS -------------

N     <- 75 # number of patients
fails <- 6  # number of failures

#Conditions for the beta prior
mean = 0.15
std  = 0.14

# ----------- INFERENCE CALCULATION ----------------

#calculating the parameters of the prior
alpha <- mean*(mean*(1-mean)/(std^2)-1)
beta <- (1-mean) * (mean*(1-mean)/(std^2) -1)

beta_prior <- function(x) { ifelse(x > 0 & x < 1, dbeta(x, shape1 = alpha, shape2 = beta), 0)}

beta_posterior      <- function(p) posterior(p, beta_prior, binom_likelihood, lower=0, upper=1, size=N
mean_beta_posterior <- mean_pdf(beta_posterior, lower = 0, upper = 1)
std_beta_posterior  <- std_pdf (beta_posterior, lower = 0, upper = 1)
beta_95_interval <- sapply(c(0., 0.95), function(P) inverse_cumulative(beta_posterior, p = P, lower =

#----------------- PLOTTING -----------------------

curve(beta_posterior, from = 0, to = 1, xlab = TeX('$p_{fail}$'), ylab = 'Posterior', col=color_vector

x_plot <- seq(from=beta_95_interval[1], to=beta_95_interval[2], length.out=500)
y_plot <- c(0, beta_posterior(x_plot), 0)
x_plot <- c(beta_95_interval[1], x_plot, beta_95_interval[2])
polygon(x_plot, y_plot, col = adjustcolor(color_vector[5], alpha.f = 0.25),border = NA)

grid()

abline(v=mean_beta_posterior, col = color_vector[7], lwd=2, lty='longdash')
abline(v=mean_beta_posterior - std_beta_posterior, col = color_vector[6], lwd=2, lty='dashed')
abline(v=mean_beta_posterior + std_beta_posterior, col = color_vector[6], lwd=2, lty='dashed')

legend("topright", legend = c("Beta Posterior", "Mean", "±std"), col = c(color_vector[1], color_vector

legend("right", legend="95%% credibility interval", fill=color_vector[5])
```




```
# -------------------- WRITE OUTPUT ---------------------

writeLines(
  sprintf(
'
  - The mean of the posterior is: %.3f
  - The standard deviation is: %.3f
  - The margin of the 95%% credibility interval is: %.3f
', mean_beta_posterior, std_beta_posterior, beta_95_interval[2]
  )
)
```

```
##
##   - The mean of the posterior is: 0.085
##   - The standard deviation is: 0.031
##   - The margin of the 95% credibility interval is: 0.141
```

5. In the frequentist approach, we can use a binomial test. Looking at the results, with this approach it is not possible to reject the hypothesis that the new method is worse than the previous one at a 5% level of significance, since the 95% confidence interval includes p = 15%.

```
binom.test(fails, N, p = 0.15, alternative = "less")
```

```
##
##  Exact binomial test
##
## data:  fails and N
## number of successes = 6, number of trials = 75, p-value = 0.05435
## alternative hypothesis: true probability of success is less than 0.15
## 95 percent confidence interval:
##  0.0000000 0.1517971
## sample estimates:
## probability of success
##                   0.08
```

Exercise 2: Bayesian Inference with Normal Distribution and Step Function Prior

Scenario

A researcher collects 16 observations (n=16) that are supposed to come from a normal distribution with known variance $\sigma^2 = 4$. The observations are:

4.09, 4.68, 1.87, 2.62, 5.58, 8.68, 4.07, 4.78, 4.79, 4.49, 5.85, 5.09, 2.40, 6.27, 6.30, 4.47

The prior distribution for the mean (μ) is a step function defined as:

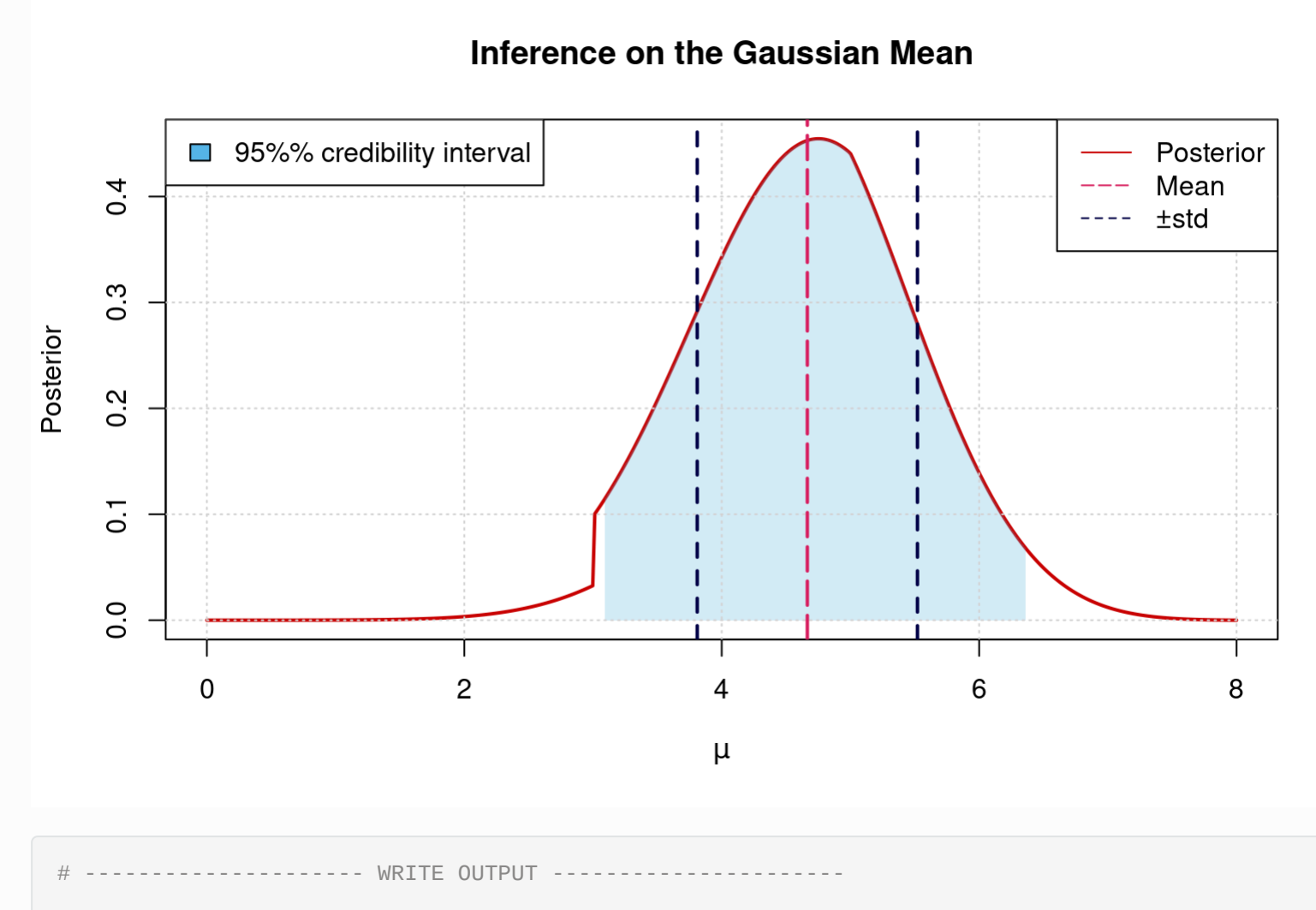
$$g(\mu) = \begin{cases} 1, & 0 < \mu \leq 3 \\ 3, & 3 < \mu \leq 5 \\ 8 - \mu, & 5 < \mu \leq 8 \\ 0, & \mu > 8 \end{cases}$$

Tasks:

1. Find the posterior distribution, posterior mean, and standard deviation.
2. Find the 95% credibility interval for μ.
3. Plot the posterior distribution, indicating the mean value, standard deviation, and 95% credibility interval.
4. Plot the prior, likelihood, and posterior distribution on the same graph.

Answers

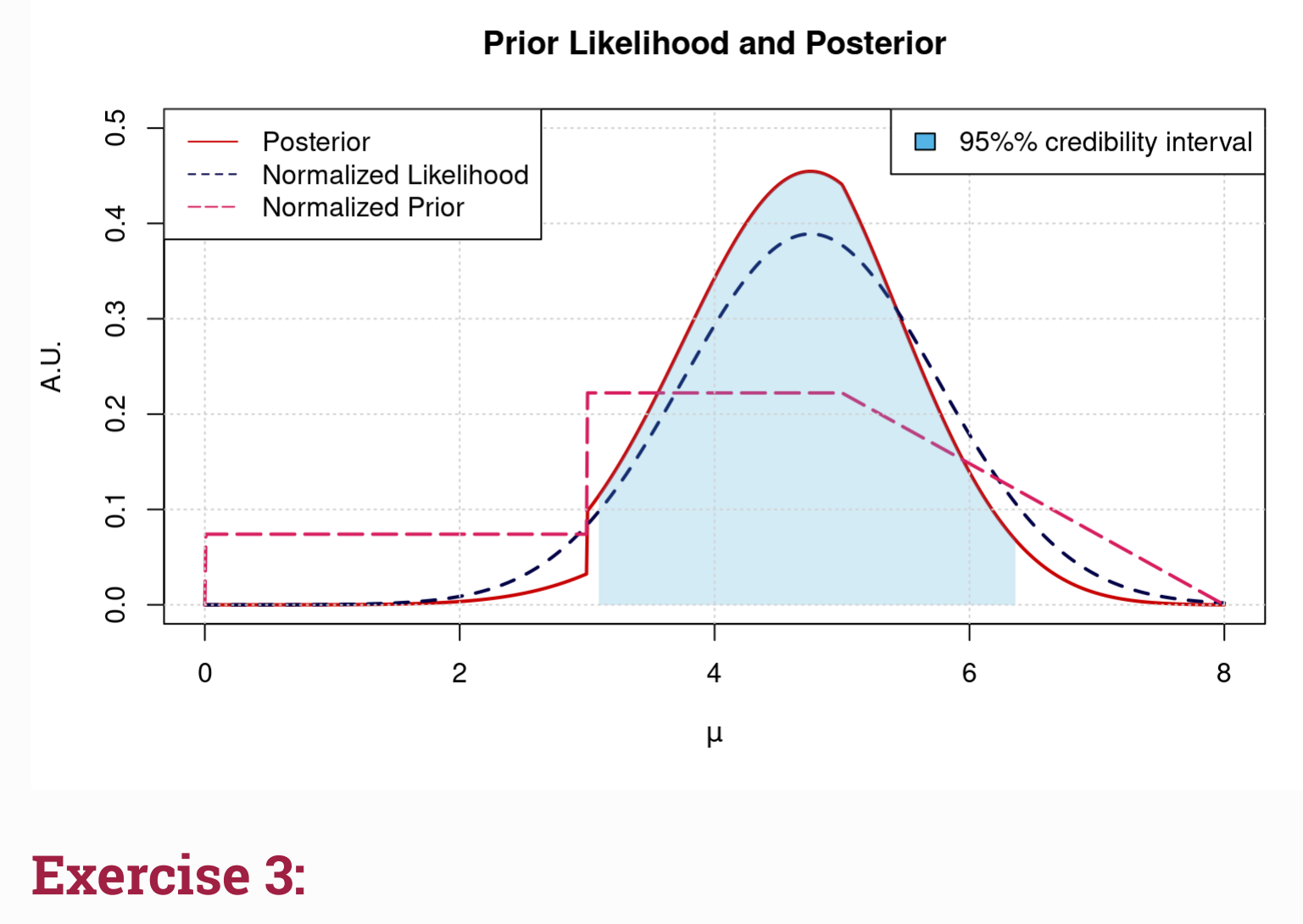
```
n <- 16          # number of observations
sigma_prior <- 4 # assumed to be known
data <- c(4.09, 4.68, 1.87, 2.62, 5.58, 8.68, 4.07, 4.78, 4.79, 4.49, 5.85, 5.09, 2.40, 6.27, 6.30, 4.

prior_values <- function(mu) sapply(mu, function(x) ifelse(x>0 && x <= 3, 1., ifelse(x>3&&x<=5, 3., if
posterior_values <- function(mu) posterior(mu, prior_values, norm_likelihood, lower = 0, upper = 8, x=

normalized_prior <- function(mu) {
  norm_const <- integrate(prior_values, lower = 0, upper = 8)$value
  prior_values(mu)/norm_const
}

normalized_likelihood <- Vectorize(function(mu) {
  norm_const <- integrate(function(X) norm_likelihood(mu=X, x=data, sd=sigma_prior), lower = -Inf, upp
  norm_likelihood(mu, x=data, sd=sigma_prior)/norm_const
})

mean_posterior <- mean_pdf(posterior_values, lower = 0, upper = 8)
std_posterior  <- std_pdf (posterior_values, lower = 0, upper = 8)
cred_95_interval <- sapply(c(0.025, 0.975), function(P) inverse_cumulative(posterior_values, p = P, lo

#------------------ PLOTTING ------------------------

curve(posterior_values, from = 0, to = 8, xlab = expression(mu), ylab = 'Posterior', col=color_vector[

x_plot <- seq(from=cred_95_interval[1], to=cred_95_interval[2], length.out=500)
y_plot <- c(0, posterior_values(x_plot), 0)
x_plot <- c(cred_95_interval[1], x_plot, cred_95_interval[2])
polygon(x_plot, y_plot, col = adjustcolor(color_vector[5], alpha.f = 0.25),border = NA)

grid()

abline(v=mean_posterior, col = color_vector[7], lwd=2, lty='longdash')
abline(v=mean_posterior - std_posterior, col = color_vector[6], lwd=2, lty='dashed')
abline(v=mean_posterior + std_posterior, col = color_vector[6], lwd=2, lty='dashed')

legend("topright", legend = c("Posterior", "Mean", "±std"), col = c(color_vector[1], color_vector[7],

legend("topleft", legend="95%% credibility interval", fill=color_vector[5])
```




```
# -------------------- WRITE OUTPUT ---------------------

writeLines(
  sprintf(
'
  - The mean of the posterior is: %.3f
  - The standard deviation is: %.3f
  - The 95%% credibility interval is: [ %.3f , %.3f ]
', mean_posterior, std_posterior, cred_95_interval[1], cred_95_interval[2]
  )
)
```

```
##
##   - The mean of the posterior is: 4.665
##   - The standard deviation is: 0.855
##   - The 95% credibility interval is: [ 3.093 , 6.358 ]
```

```
#------------------ PLOTTING ------------------------
curve(posterior_values, from = 0, to = 8, xlab = expression(mu), ylab = 'A.U.', col=color_vector[1], m
curve(normalized_likelihood(x), from = 0, to = 8, col=color_vector[6], lwd = 2, lty = 2, n = 1000, add
curve(normalized_prior, from = 0, to = 8, col=color_vector[7], lwd = 2, lty = 5, n = 1000,  add = TRUE

x_plot <- seq(from=cred_95_interval[1], to=cred_95_interval[2], length.out=500)
y_plot <- c(0, posterior_values(x_plot), 0)
x_plot <- c(cred_95_interval[1], x_plot, cred_95_interval[2])
polygon(x_plot, y_plot, col = adjustcolor(color_vector[5], alpha.f = 0.25),border = NA)

grid()

legend("topleft", legend = c("Posterior", "Normalized Likelihood", "Normalized Prior"), col = c(color_

legend("topright", legend="95%% credibility interval", fill=color_vector[5])
```



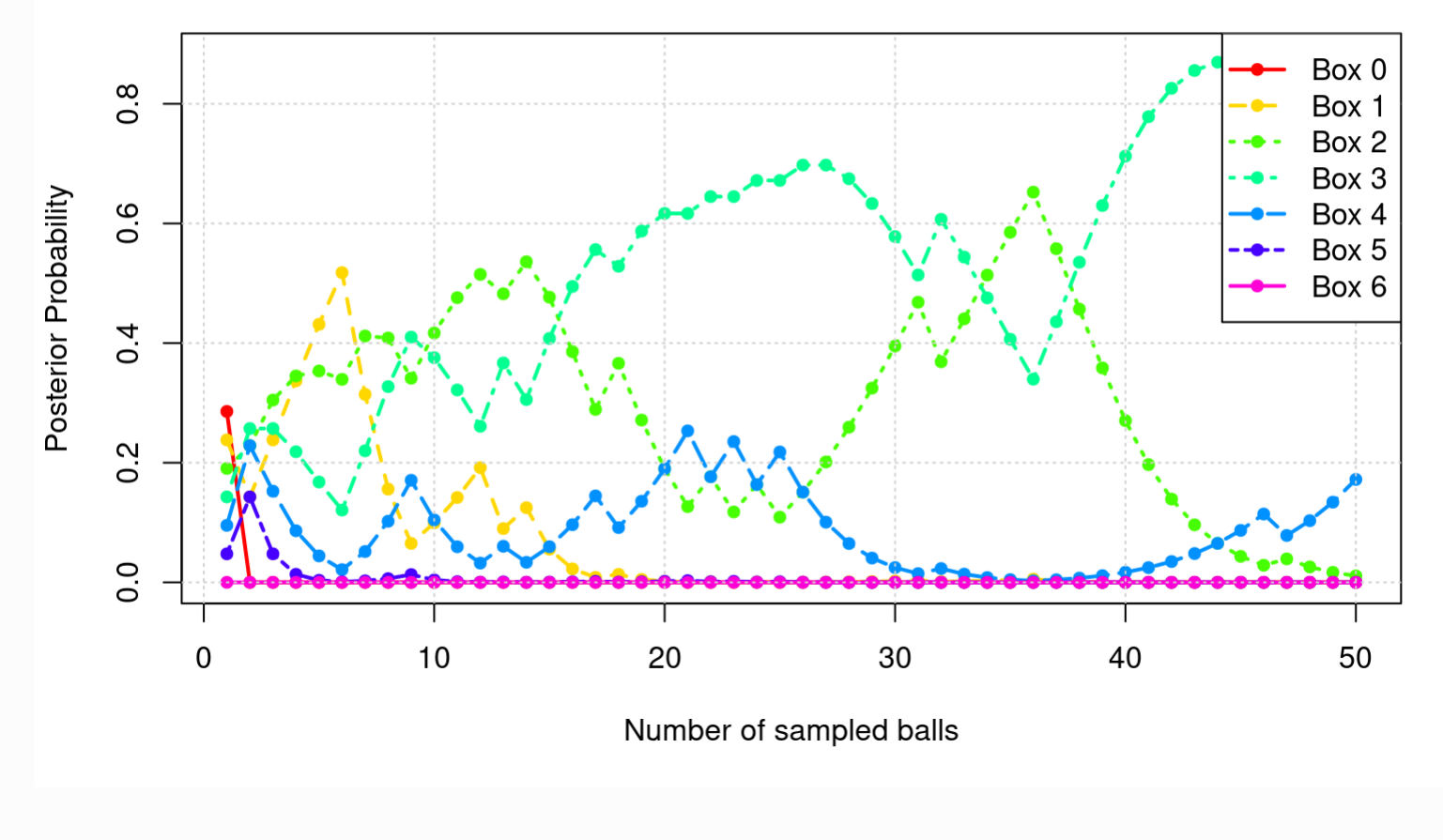
Exercise 3:

Write a program in R that:

1. selects a random box
2. makes random sampling from the box
3. prints on the standard output the probability of selecting each box
4. plots the probability for each box as a function of the number of trial

```
# building the boxes, s.t. the box i contains i white balls (=1)
N_box <- 7
box <- lapply(1:N_box, function(i) sapply(1:N_box-1, function(j) ifelse(j < i, 1, 0)))
P_white <- sapply(0:(N_box-1), function(ibox) ibox/(N_box-1))

N_extracted_box  <- 1 # How many times do we select a new box
N_ball <- 50 # How many times do we select a ball from each box
ball_seq <- 1:N_ball
box_seq  <- 1:N_extracted_box

# Select the boxes
box_sampling <- sample(1:7, N_extracted_box, replace=TRUE)

# Coding the process of selecting some balls
ball_sampling <- lapply(box_sampling, function(i) sample(box[[i]], N_ball, replace=TRUE))

# Calculate the partial sum over the white sampling for each step
N_white <- lapply(box_seq, function(ibox) sapply(ball_seq, function(jball) sum(ball_sampling[[ibox]][1

likelihood_list <- lapply(1:N_box,
                              function (ibox) mapply(
                                function(n_white, n_samples)
                                  dbinom(x=n_white, size = n_samples, prob = P_white[ibox])
                                , N_white[[1]], ball_seq)
                           )

sum_over_boxes <- function(jball) sum(sapply(1:N_box, function(ibox) likelihood_list[[ibox]][jball]))
normalize_over_boxes   <-
  function(jball)
    sapply(1:N_box, function(ibox) likelihood_list[[ibox]][jball]/sum_over_boxes(jball))

posterior_matrix <- sapply(ball_seq, normalize_over_boxes)
```

```
colors <- rainbow(N_box)
line_types <- 1:N_box

plot(posterior_matrix[1, ], type = 'o', pch = 20, col = colors[1],
     ylim = range(posterior_matrix), xlab = "Number of sampled balls", ylab = "Posterior Probability",

void <- sapply(2:N_box, function(ibox) lines(posterior_matrix[ibox, ], type = 'o', pch = 20, col = col

grid()

legend("topright", legend = paste("Box", 0:(N_box-1)), col = colors, lty = line_types, pch = 20, lwd=2
```



Emanuele Coradin
2024-05-20