Assignment #5

Elliot Smith

9/23/2018

Problem 1

Part i

Prior Distribution: $p(\theta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}$
Likelihood: $p(y|\theta) = \theta e^{-\theta y}$

Posterior Distribution:

$$\begin{aligned} p(\theta|y) &\propto p(y|\theta)p(\theta) \\ &\propto \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta} \times \theta^n e^{-\sum_{i=1}^n \theta y_i} \\ &\propto \theta^{(\alpha+n)-1}e^{-(\beta+\sum_{i=1}^n y_i)\theta} \\ &\sim Gamma(\alpha+n, \beta+\sum_{i=1}^n y_i) \end{aligned}$$

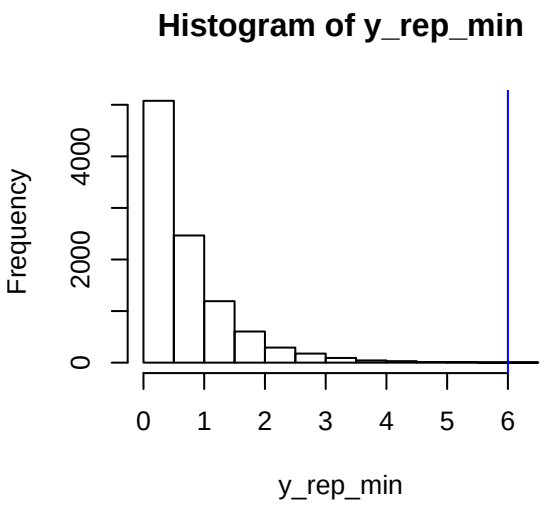
Please refer to the Code Appendix section for the sampling technique used.

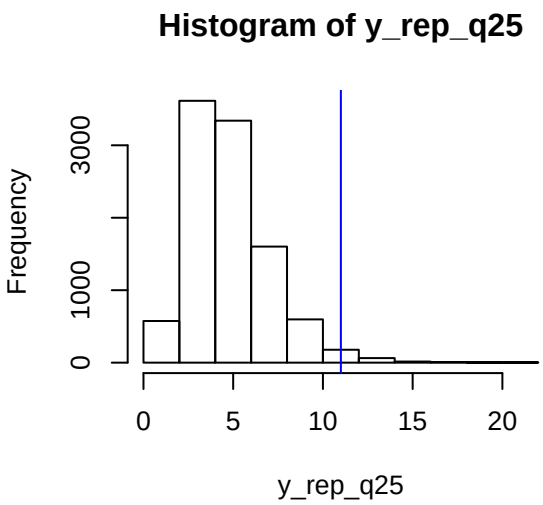
Part ii

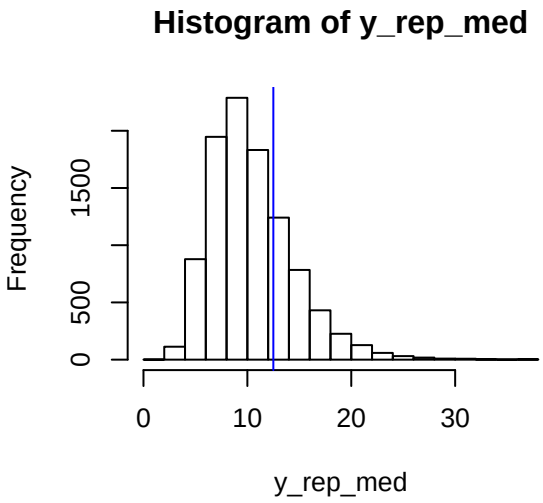
Please refer to the Code Appendix section for the technique used to construct the replicate data set.

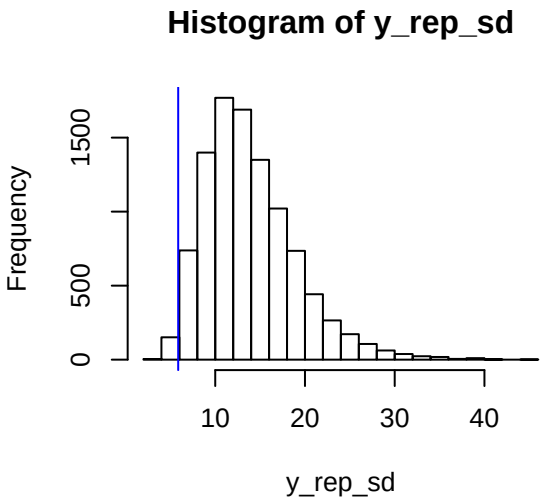
Part iii

Please refer to the Code Appendix section for how the test statistics were computed.









Part iv

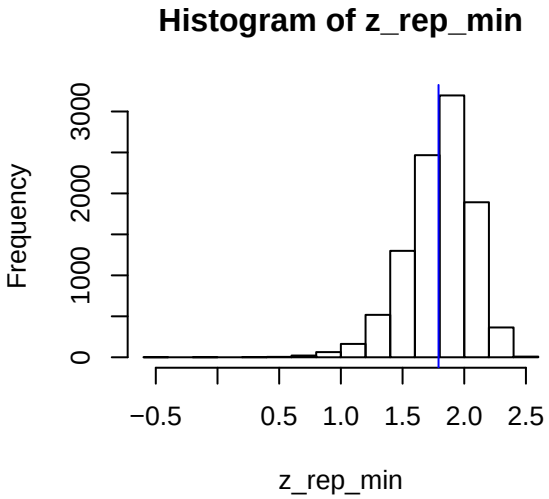
```
## Minimum p-value:  0.0004
## 25% Quantile p-value:  0.0148
## Median p-value:  0.2581
## Standard Deviation p-value:  0.9863
## Probability 10 Minute Wait:  0.5078
```

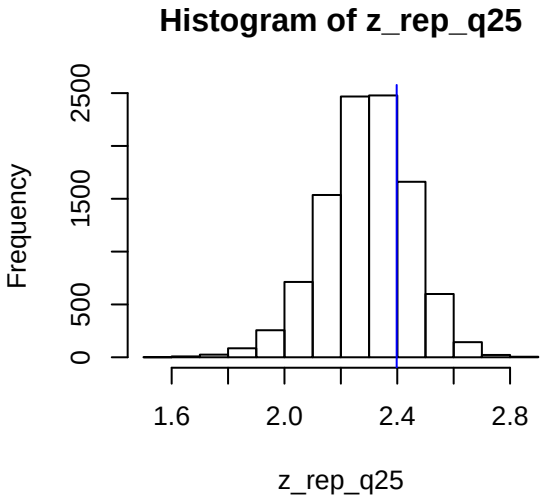
Our results imply that our model not be correct since many of our p-values are quite different from our observed value as our p-values tend to hover around either extreme of 0 or 1; this tells us that it is very unlikely that the observed value of each statistic lies in our posterior predictive distributions. This suggests that the quantity we computed in class may be incorrect since the model that we computed for our result may not be very accurate.

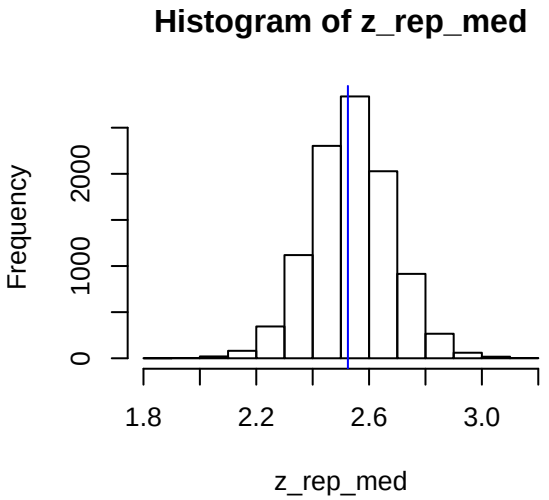
Problem 2

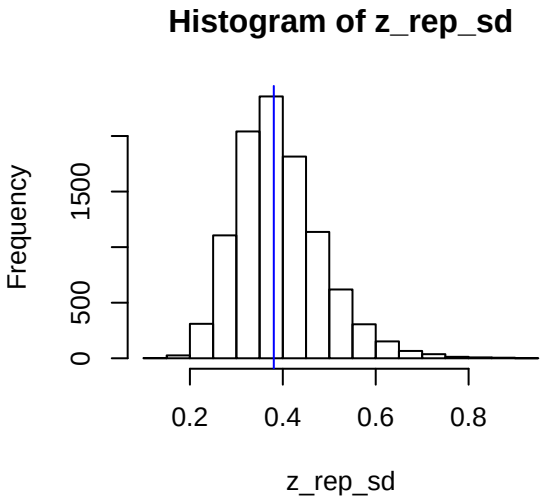
Part i

Please refer to the Code Appendix section for the sampling technique, the technique used to construct the replicate data set and how test statistics were computed.










```
## Minimum p-value:  0.5596
## 25% Quantile p-value:  0.248
## Median p-value:  0.5434
## Standard Deviation p-value:  0.5064
```

Part ii

```
## Probability 10 Minute Wait:  0.2759
```

Yes this estimate seems to be much more reasonable as our result implies that we would certainly expect to see our result less often than originally predicted in the previous problem; our data and results, given that

our Bayesian p-values tend to better represent that statistics in question, reinforce this result.

Problem 3

Part i

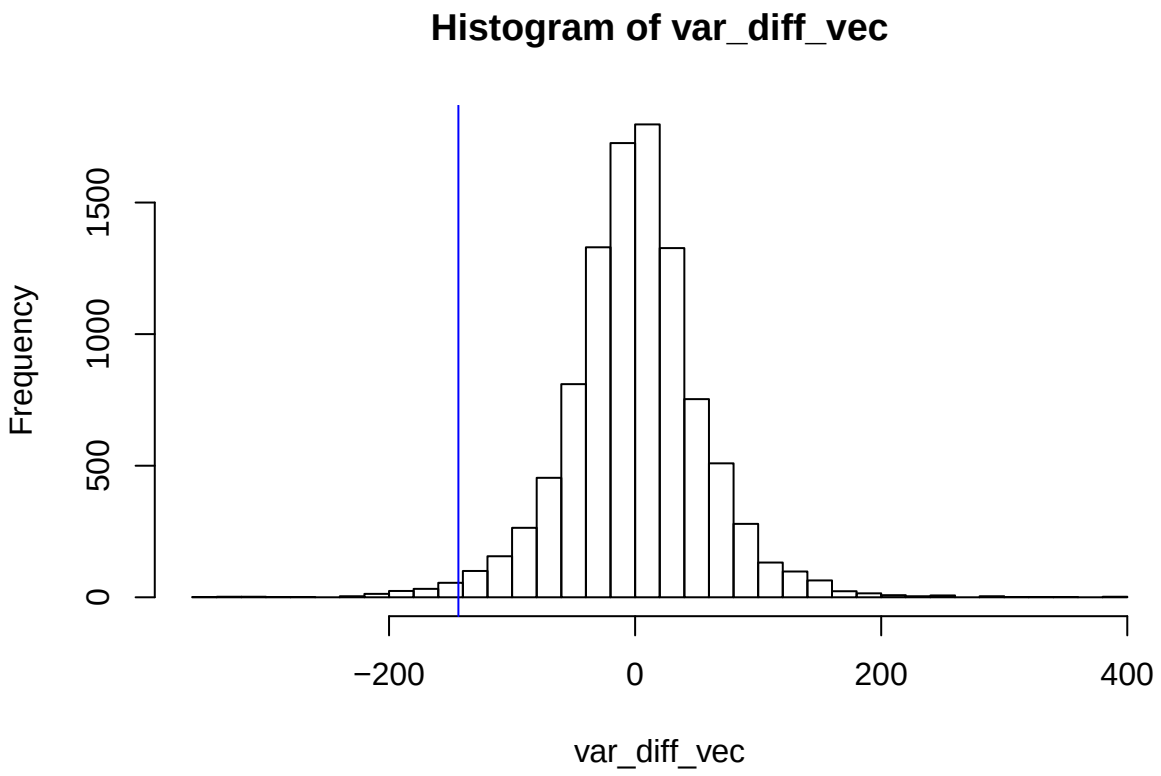
Please refer to the Code Appendix section for the sampling technique used.

Part ii

Please refer to the Code Appendix section for the technique used to construct the replicate data set.

Part iii

For my test statistic, I looked at the difference in variance of the first 10 observations compared to the second 10 observations. I used this statistic because there is a clear difference in variance between the first 10 observation (58.2666) and the second 10 (6.0111); this led me to believe this would be a good avenue to explore.



Part iv

```
## Probability 10 Minute Wait:  0.9877
```

As we can see by the extreme value of our p-value, there is strong evidence of model misspecification. A possible correction would be to explore whether or not the two sets of observations should be modeled together or whether they should be treated and modeled separately. If we choose to model the first 10 observations

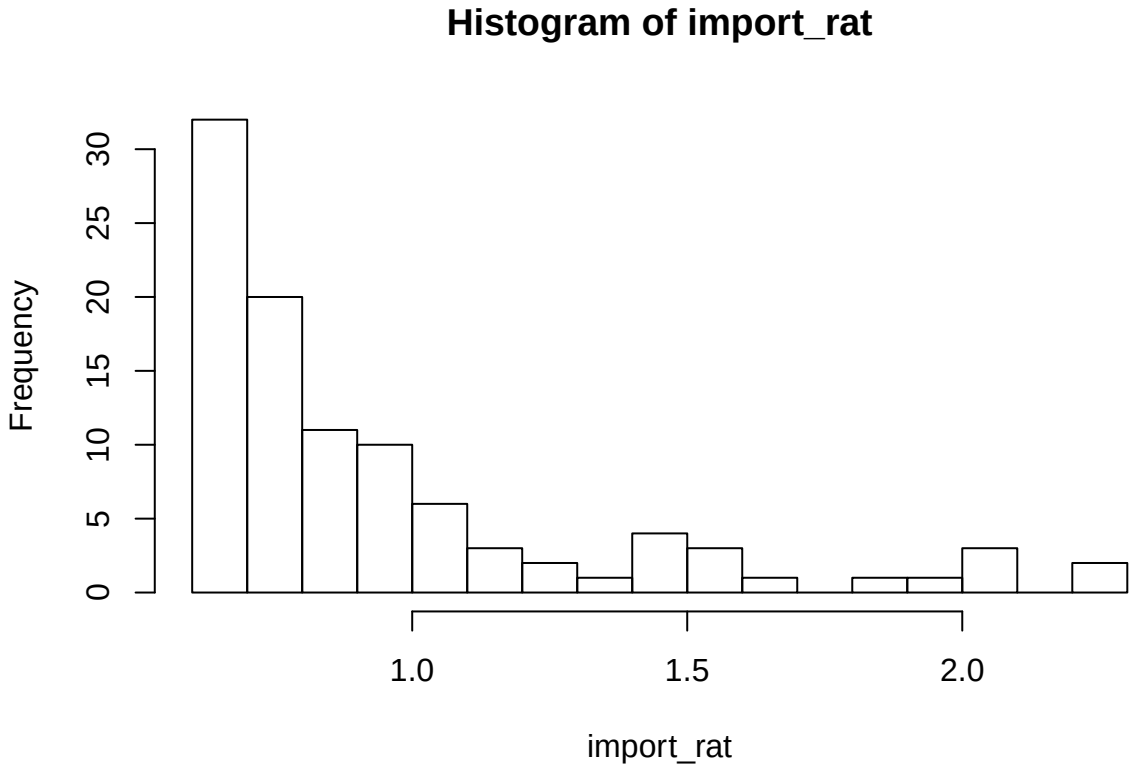
separately from the second 10 observations (thus treating them as samples drawn from separate populations) and create two models, we could be certain that each of the models more closely represents the data it is modeling; this would improve our ability to derive knowledge from our data.

Problem 4

Part i - BDA 10.6

Part a

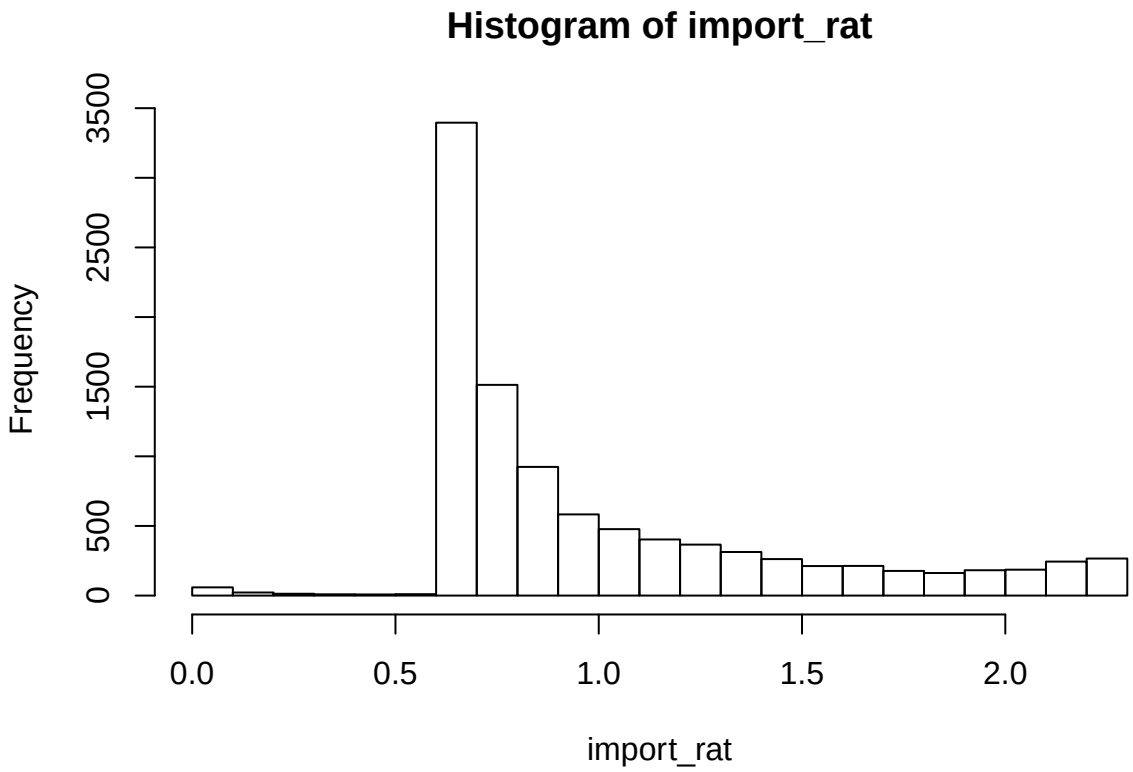
Please refer to the Code Appendix section for the sampling technique used.



Part b

```
## Importance Sampled Expected Value:  0.171
## True Expected Value:  0
## Importance Sampled Variance:  0.8228
## True Variance:  1
```

Part c



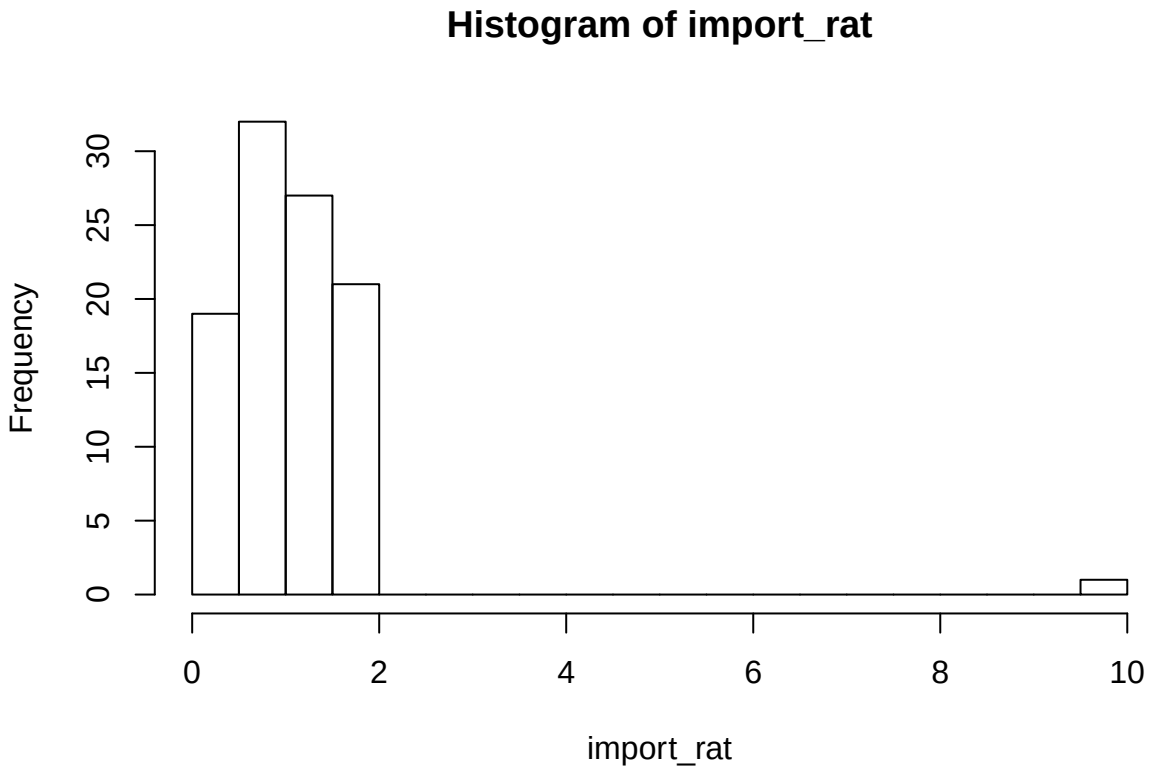

```
## Importance Sampled Expected Value:  -0.0219
## True Expected Value:  0
## Importance Sampled Variance:  1.0089
## True Variance:  1
```

Part d

```
## Effective Sample Size:  8197.834
```

Part ii - BDA 10.7

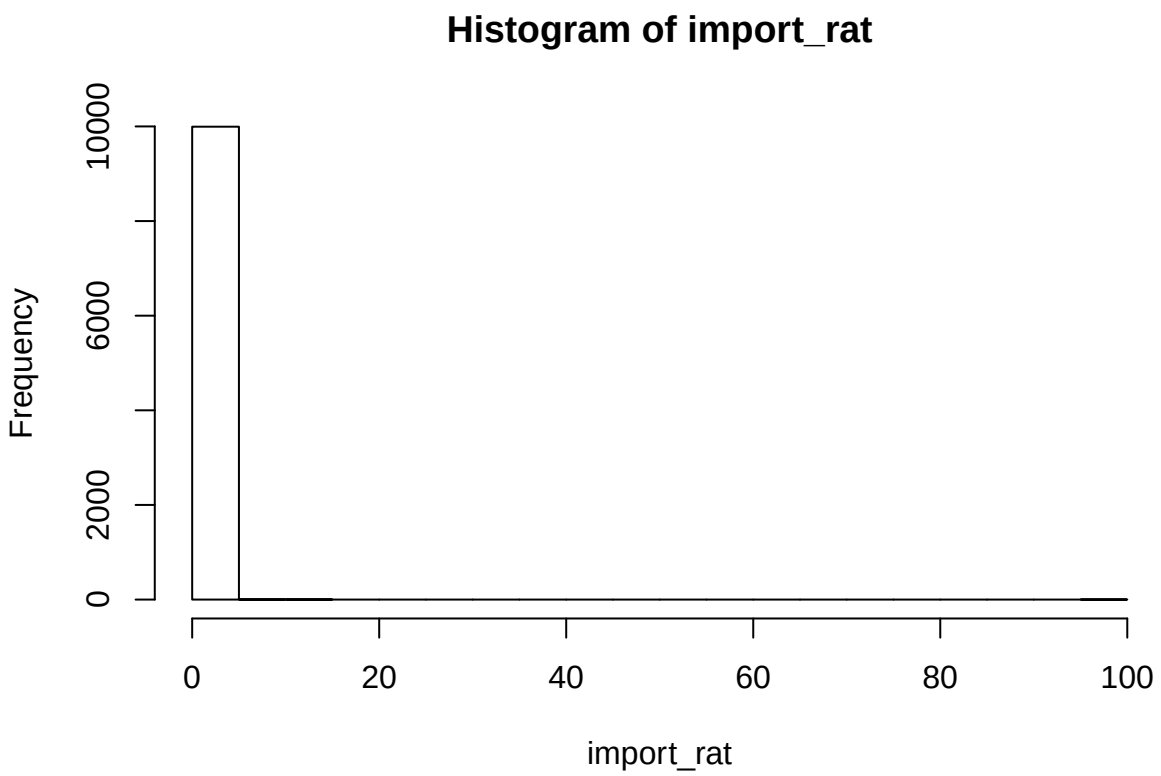
Part a



Part b

```
## Importance Sampled Expected Value:  -0.4582
## True Expected Value:  0
## Importance Sampled Variance:  1.6712
## True Variance:  1
```

Part c




```
## Importance Sampled Expected Value:  -0.0427
```

```
## True Expected Value:  0
```

```
## Importance Sampled Variance:  0.893
```

```
## True Variance:  1
```

Our estimates of $var(\theta|y)$ are sytematically too low due to the fact that the importance weights are not well-behaved because for some values of theta our importance ratios are too large with low probability (we can see this very clearly in our histogram of importance weights). The result of this is that fewer values will be sampled from this region, causing our variance to be skewed.

Part d

```
## Effective Sample Size:  4705.651
```

Code Appendix

```
## No scientific notation

options(scipen = 999)

## Load in packages

library(fGarch)
```

```
########## Problem 1 ##########



## Input the known parameters and data

y <- c(15,9,15,8,8,20,25,31,13,12,11,15,11,13,13,11,11,12,6,14)
n <- length(y)
S <- 10000
alpha <- 5
beta <- alpha * 15



##### Part i #####



## Sample from the Gamma

samp_1i <- rgamma(n = S, shape = n + alpha, rate = beta + sum(y))



##### Part ii #####



## Construct the replicate data sets

y_rep <- matrix(0, nrow = S, ncol = n)

for(s in 1:S) {

    y_rep[s,] <- rexp(n = n, rate = samp_1i[s])

}



##### Part iii #####



## Compute the required test statistics for each replicate data set

y_rep_min <- apply(y_rep, 1, min)
y_rep_q25 <- apply(y_rep, 1, quantile, probs = 0.25)
y_rep_med <- apply(y_rep, 1, median)
y_rep_sd <- apply(y_rep, 1, sd)

par(mfrow = c(2, 2))
```

```r
hist(x = y_rep_min)
abline(v = min(y), col = 'blue')

hist(x = y_rep_q25)
abline(v = quantile(y, 0.25), col = 'blue')

hist(x = y_rep_med)
abline(v = median(y), col = 'blue')

hist(x = y_rep_sd)
abline(v = sd(y), col = 'blue')



##### Part iv #####



## Compute the posterior predictive p-values for each statistic

pval_min <- mean(y_rep_min > min(y))
pval_q25 <- mean(y_rep_q25 > quantile(y, 0.25))
pval_med <- mean(y_rep_med > median(y))
pval_sd <- mean(y_rep_sd > sd(y))

## Probability of waiting at most 10 minutes

wait_10 <- mean(y_rep < 10)

## Output the results

cat("Minimum p-value: ", round(pval_min, 4))
cat("25% Quantile p-value: ", round(pval_q25, 4))
cat("Median p-value: ", round(pval_med, 4))
cat("Standard Deviation p-value: ", round(pval_sd, 4))
cat("Probability 10 Minute Wait: ", round(wait_10, 4))



########## Problem 2 ##########



## Input the known parameters and data

z <- log(y)
n <- length(z)
S <- 10000
alpha <- 0.01
beta <- 0.01

## Draw the sample from the posterior
```

```r
s_squared <- sd(z)^2
post_tau <- rgamma(n = S,
                   shape = alpha + (n - 1)/2,
                   rate = beta + (n - 1) * (s_squared / 2))

post_sigma <- 1/sqrt(post_tau)

Q_mu <- n * post_tau
ell_mu <- n * post_tau * mean(z)
post_mu <- rnorm(n = S, mean = (Q_mu^-1) * ell_mu, sd = sqrt(Q_mu^-1))



##### Part i #####



## Construct the replicate data set

z_rep <- matrix(0, nrow = S, ncol = n)

for(s in 1:S) {

    z_rep[s,] <- rnorm(n = n, mean = post_mu[s], sd = post_sigma[s])

}

## Compute the required test statistics for each replicate data set

z_rep_min <- apply(z_rep, 1, min)
z_rep_q25 <- apply(z_rep, 1, quantile, probs = 0.25)
z_rep_med <- apply(z_rep, 1, median)
z_rep_sd <- apply(z_rep, 1, sd)

par(mfrow = c(2, 2))

hist(x = z_rep_min)
abline(v = min(z), col = 'blue')

hist(x = z_rep_q25)
abline(v = quantile(z, 0.25), col = 'blue')

hist(x = z_rep_med)
abline(v = median(z), col = 'blue')

hist(x = z_rep_sd)
abline(v = sd(z), col = 'blue')

## Compute the posterior predictive p-values for each statistic

pval_min <- mean(z_rep_min > min(z))
pval_q25 <- mean(z_rep_q25 > quantile(z, 0.25))
pval_med <- mean(z_rep_med > median(z))
```

```r
pval_sd <- mean(z_rep_sd > sd(z))

## Output the results

cat("Minimum p-value: ", round(pval_min, 4))
cat("25% Quantile p-value: ", round(pval_q25, 4))
cat("Median p-value: ", round(pval_med, 4))
cat("Standard Deviation p-value: ", round(pval_sd, 4))



##### Part ii #####



## Probability of waiting at most 10 minutes

wait_10 <- mean(z_rep < log(10))

## Output the results

cat("Probability 10 Minute Wait: ", round(wait_10, 4))



########## Problem 3 ##########



## Load in the data and known parameters

y <- c(1.960,
       1.823,
       0.968,
       2.753,
       0.779,
       3.003,
       2.463,
       2.197,
       2.806,
       3.137,
       -1.813,
       -4.242,
       -19.780,
       8.597,
       -3.134,
       7.486,
       12.621,
       -11.511,
       -12.746,
       17.244)
n <- length(y)
S <- 10000
```

```r
alpha <- 0.01
beta <- 0.01



##### Part i #####



## Draw the sample from the posterior

s_squared <- sd(y)^2
post_tau <- rgamma(n = S,
                   shape = alpha + (n - 1)/2,
                   rate = beta + (n - 1) * (s_squared / 2))

post_sigma <- 1/sqrt(post_tau)

Q_mu <- n * post_tau
ell_mu <- n * post_tau * mean(y)
post_mu <- rnorm(n = S, mean = (Q_mu^-1) * ell_mu, sd = sqrt(Q_mu^-1))



##### Part ii #####



## Construct the replicate data set

y_rep <- matrix(0, nrow = S, ncol = n)

for(s in 1:S) {

    y_rep[s,] <- rnorm(n = n, mean = post_mu[s], sd = post_sigma[s])

}



##### Part iii #####



## Compute the variance difference of the first 10 minus second 10 observation

y_1_var <- var(y[1:10])
y_2_var <- var(y[11:20])

var_diff <- y_1_var - y_2_var

## Compute the same variance for the replicate data
```

```r
var_diff_vec <- numeric()

for (i in 1:S) {

    y_1_temp <- var(y_rep[i, 1:10])
    y_2_temp <- var(y_rep[i, 11:20])
    var_diff_vec[i] <- y_1_temp - y_2_temp

}

## Plot the histogram of the results

par(mfrow = c(1, 1))

hist(var_diff_vec, breaks = 50)
abline(v = var_diff, col = 'blue')



##### Part iv #####



## Posterior predictive p-value

post_pred <- mean(var_diff < var_diff_vec)

## Output the results

cat("Probability 10 Minute Wait: ", round(post_pred, 4))



########## Problem 4 ##########



##### Exercise 10.6 #####



### Part a ####



## Input the known parameters

S <- 100
mu <- 0
var <- 1
deg_free <- 3

## Sample theta from g
```

```r
theta <- rstd(n = S, mean = mu, sd = sqrt(var), nu = deg_free)

## Calculate and plot the importance ratios

import_rat <- dnorm(x = theta, mean = mu, sd = sqrt(var)) /
    dstd(x = theta, mean = mu, sd = sqrt(var), nu = deg_free)

hist(import_rat, breaks = 20)



### Part b ###



## Compute the importance sampled EX and VarX

ex_imp <- ((1/S) * sum(theta * import_rat)) / ((1/S) * sum(import_rat))
var_imp <- ((1/S) * sum(theta^2 * import_rat)) / ((1/S) * sum(import_rat)) - ex_imp^2

## Compute the true EX and VarX

ex_true <- 0 # mean(dnorm(x = theta, mean = mu, sd = sqrt(var)))
var_true <- 1 # var(dnorm(x = theta, mean = mu, sd = sqrt(var)))

## Output the results

cat("Importance Sampled Expected Value: ", round(ex_imp, 4))
cat("True Expected Value: ", round(ex_true, 4))
cat("Importance Sampled Variance: ", round(var_imp, 4))
cat("True Variance: ", round(var_true, 4))



### Part c ###



## Input the known parameters

S <- 10000
mu <- 0
var <- 1
deg_free <- 3

## Sample theta from g

theta <- rstd(n = S, mean = mu, sd = sqrt(var), nu = deg_free)

## Calculate and plot the importance ratios

import_rat <- dnorm(x = theta, mean = mu, sd = sqrt(var)) /
    dstd(x = theta, mean = mu, sd = sqrt(var), nu = deg_free)
```

```r
hist(import_rat, breaks = 20)

## Compute the importance sampled EX and VarX

ex_imp <- ((1/S) * sum(theta * import_rat)) / ((1/S) * sum(import_rat))
var_imp <- ((1/S) * sum(theta^2 * import_rat)) / ((1/S) * sum(import_rat)) - ex_imp^2

## Compute the true EX and VarX

ex_true <- 0 # mean(dnorm(x = theta, mean = mu, sd = sqrt(var)))
var_true <- 1 # var(dnorm(x = theta, mean = mu, sd = sqrt(var)))

## Output the results

cat("Importance Sampled Expected Value: ", round(ex_imp, 4))
cat("True Expected Value: ", round(ex_true, 4))
cat("Importance Sampled Variance: ", round(var_imp, 4))
cat("True Variance: ", round(var_true, 4))



### Part d ###



## Compute the effective sample size

S_eff <- 1 / (sum((import_rat / sum(sample(import_rat)))^2))

## Output the results

cat("Effective Sample Size: ", round(S_eff, 4))



##### Exercise 10.7 #####



### Part a ####



## Input the known parameters

S <- 100
mu <- 0
var <- 1
deg_free <- 3

## Sample theta from g

theta <- rnorm(n = S, mean = mu, sd = sqrt(var))
```

```r
## Calculate and plot the importance ratios

import_rat <- dstd(x = theta, mean = mu, sd = sqrt(var), nu = deg_free) /
    dnorm(x = theta, mean = mu, sd = sqrt(var))

hist(import_rat, breaks = 20)



### Part b ###



## Compute the importance sampled EX and VarX

ex_imp <- ((1/S) * sum(theta * import_rat)) / ((1/S) * sum(import_rat))
var_imp <- ((1/S) * sum(theta^2 * import_rat)) / ((1/S) * sum(import_rat)) - ex_imp^2

## Compute the true EX and VarX

ex_true <- 0 # mean(dstd(x = theta, mean = mu, sd = sqrt(var), nu = deg_free))
var_true <- 1 # var(dstd(x = theta, mean = mu, sd = sqrt(var), nu = deg_free))

## Output the results

cat("Importance Sampled Expected Value: ", round(ex_imp, 4))
cat("True Expected Value: ", round(ex_true, 4))
cat("Importance Sampled Variance: ", round(var_imp, 4))
cat("True Variance: ", round(var_true, 4))



### Part c ###



## Input the known parameters

S <- 10000
mu <- 0
var <- 1
deg_free <- 3

## Sample theta from g

theta <- rnorm(n = S, mean = mu, sd = sqrt(var))

## Calculate and plot the importance ratios

import_rat <- dstd(x = theta, mean = mu, sd = sqrt(var), nu = deg_free) /
    dnorm(x = theta, mean = mu, sd = sqrt(var))

hist(import_rat, breaks = 20)
```

```r
## Compute the importance sampled EX and VarX

ex_imp <- ((1/S) * sum(theta * import_rat)) / ((1/S) * sum(import_rat))
var_imp <- ((1/S) * sum(theta^2 * import_rat)) / ((1/S) * sum(import_rat)) - ex_imp^2

## Compute the true EX and VarX

ex_true <- 0 # mean(dstd(x = theta, mean = mu, sd = sqrt(var), nu = deg_free))
var_true <- 1 # var(dstd(x = theta, mean = mu, sd = sqrt(var), nu = deg_free))

## Output the results

cat("Importance Sampled Expected Value: ", round(ex_imp, 4))
cat("True Expected Value: ", round(ex_true, 4))
cat("Importance Sampled Variance: ", round(var_imp, 4))
cat("True Variance: ", round(var_true, 4))



### Part d ###



## Compute the effective sample size

S_eff <- 1 / (sum((import_rat / sum(sample(import_rat)))^2))

## Output the results

cat("Effective Sample Size: ", round(S_eff, 4))
```