Homework 8

Zachary Lazerick

March 28 2023

1) Craps - In the game of craps, two dice are rolled. If the first roll is a 7 or an 11, the player wins. If the first roll is a 2, 3, or 12, the player loses. If any other outcome is observed on the first roll, the player wins if that outcome is rolled again before a 7 is rolled; otherwise he loses.

```
Craps <- function(iterations, first_roll) {
  output <- c(); rolls <- c()
  while (length(output) < iterations) {
    sum <- 0; num_rolls <- 0
    while (sum != 7) {
      d_1 <- sample(1:6, size = 1); d_2 <- sample(1:6, size = 1)
      sum <- d_1 + d_2; num_rolls <- num_rolls + 1
      if (sum == first_roll) {
        output <- append(output, 1)
        rolls <- append(rolls, num_rolls)
        break
      }
      if (sum == 7) {
        output <- append(output, 0)
        rolls <- append(rolls, num_rolls)
        break
      }
    }
  }
  mylist <- list("Output" = output, "Rolls" = rolls)
  return(mylist)
}

set.seed(855)
results1 <- Craps(1000, first_roll = 8)
```

a) Suppose that the first roll is an 8. What is the probability that the player wins the game?

```
mean(results1$Output)
```

```
## [1] 0.458
```

b) Given that a player wins the game, what is the probability that they won on their first roll?

```
mean(results1$Output == 1 & results1$Rolls == 1)/mean(results1$Output)
```

```
## [1] 0.2729258
```

c) What is the expected number of times that you will roll the dice before the game ends?

```
mean(results1$Rolls)
```

```
## [1] 3.252
```

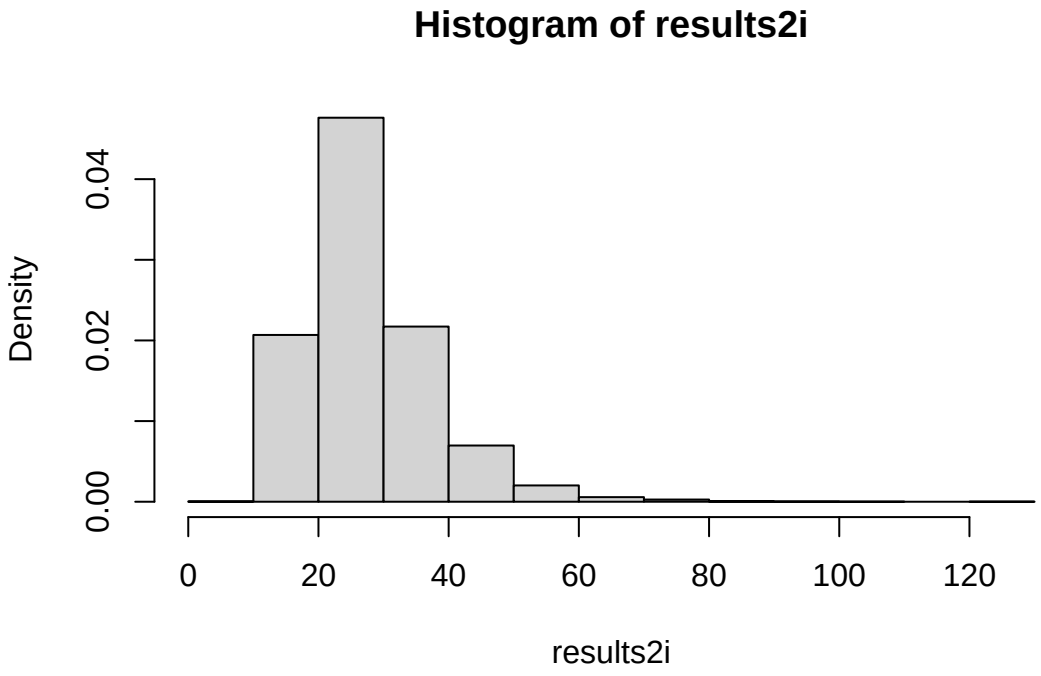
2) Sampling from an Inverse Gamma Distribution – Suppose that $n = 21$ and $s^2 = 25$. The goal of this question is to become comfortable sampling from the Inverse-Gamma distribution. In class, we stated that:

$$\sigma^2|Y \sim Inv - Gamma(\frac{n-1}{2}, \frac{1}{2}(n-1)s^2)$$

Sample 10000 values from this distribution in three ways:

i) Simulate a gamma random variable using the function rgamma() and then take its inverse. Hint: Be careful how you code this, R has two different parameterization of the gamma distribution.

```
num.samp <- 10000; n <- 21; s2 <- 25
set.seed(1249)
gamma2i <- rgamma(10000, (n-1)/2, (1/2)*(n-1)*s2)
results2i <- 1/gamma2i

hist(results2i, freq = F)
```




```
mean(results2i)
```

```
## [1] 27.56189
```

```
quantile(results2i, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 14.50756 51.62877
```

ii) Use the Inverse CDF method [i.e. draw a uniform random number and then convert to a gamma] together with the built-in gamma CDF function qgamma() – then take the inverse.

```
set.seed(1250)
unif2ii <- runif(num.samp)
gamma2ii <- qgamma(unif2ii, (n-1)/2, (1/2)*(n-1)*s2)
results2ii <- 1/gamma2ii

hist(results2ii, freq = F)
```


```
mean(results2ii)
```

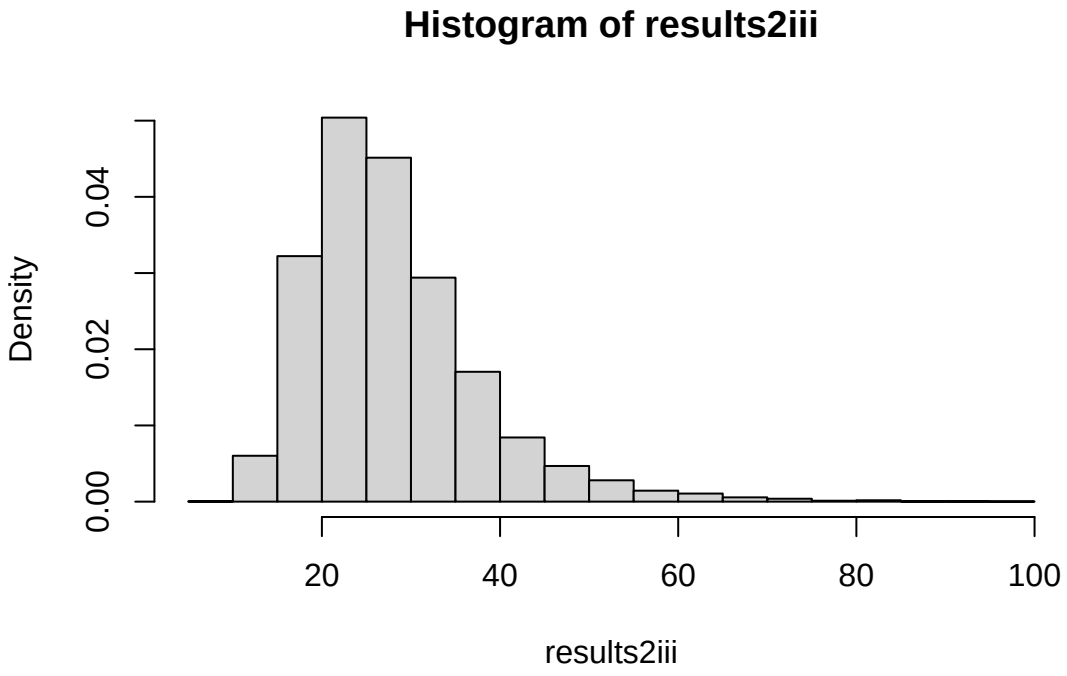
```
## [1] 27.89698
```

```
quantile(results2ii, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 14.74737 52.60898
```

iii) Sampling from the Scaled-Inverse χ^2 distribution1. [i.e. Draw a χ^2 RV, take its inverse, and then multiply by the scale factor. Wikipedia has a decent explanation of this procedure].

```
set.seed(1251)
chisq2iii <- rchisq(num.samp, n-1)
invchisq2iii <- 1/chisq2iii
results2iii <- invchisq2iii*(n-1)*s2

hist(results2iii, freq = F)
```




```
mean(results2iii)
```

```
## [1] 27.92188
```

```
quantile(results2iii, c(0.025, 0.975))
```

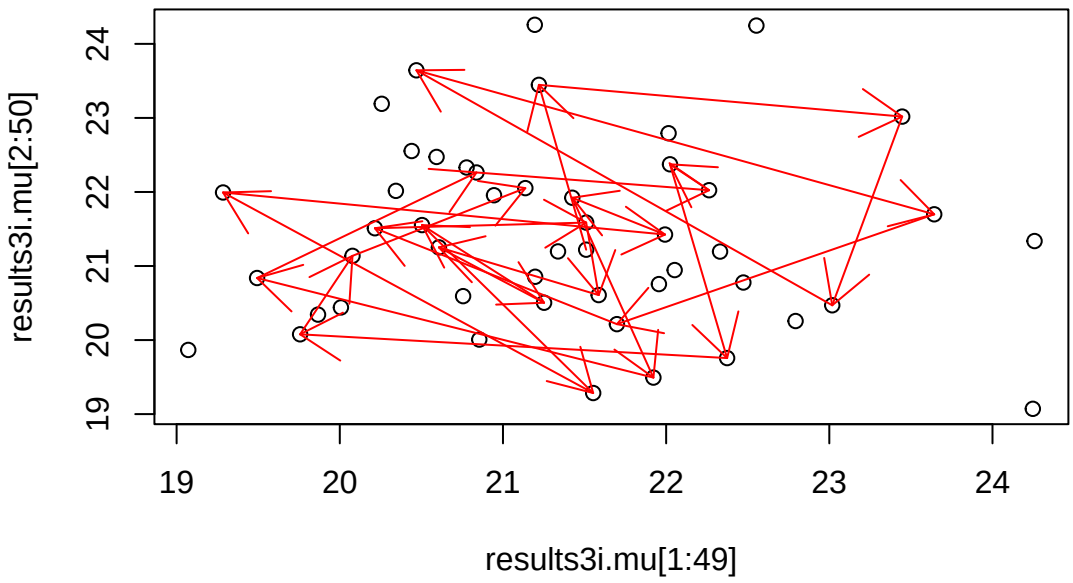
```
##     2.5%    97.5%
## 14.65284 52.54149
```

3) The Normal Distribution with Unknown Mean and Unknown Variance: Noninformative Priors – The data set on Canvas contains 45 realizations of a normal distribution. Your goal is to determine the mean and variance of the normal distribution that produced this data set. Ultimately, we are going to want to sample from the marginal posterior distributions for both μ and σ^2. Obtain 25,000 samples from $P(\mu|Y)$ and $P(\sigma^2|Y)$ using:

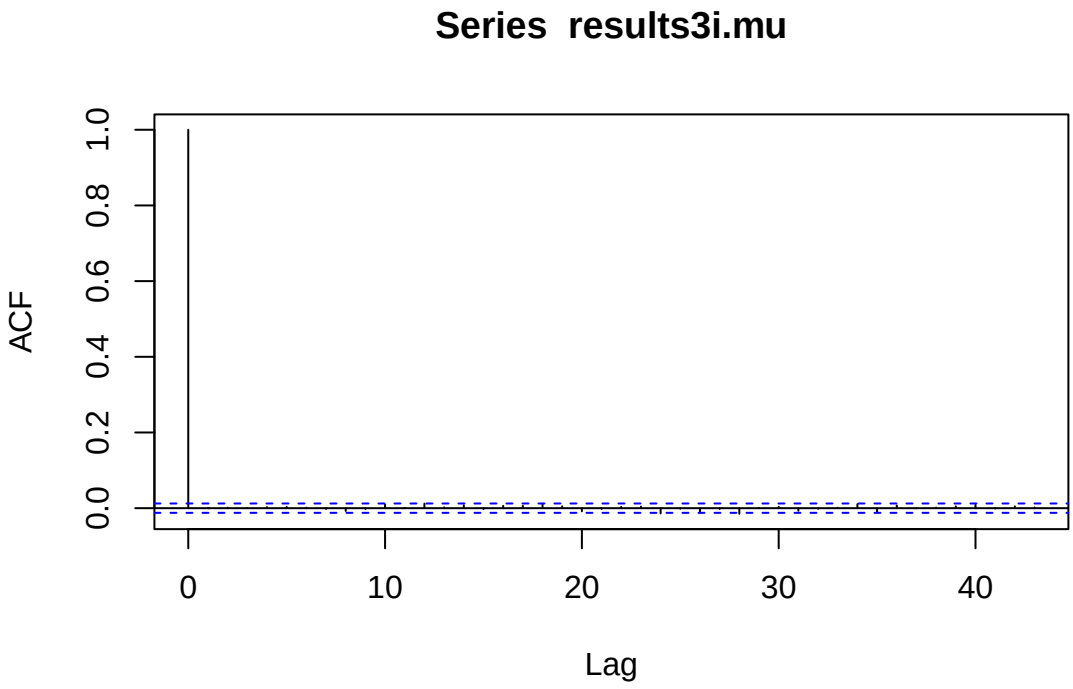
```
target_df <- read.delim("HW8_data.txt", header = F)
attach(target_df)
```

i) The Gibbs sampler [make sure you include an appropriate burn-in and lag, and write down what you decide to use for each, as well as what you used for a starting value]

```
num.samp <- 25000; n = 45

results3i.mu <- rep(0, num.samp); results3i.sigma <- rep(0, num.samp)

## Initialize Sigma
next_sigma <- var(V1); y_bar <- mean(V1); s2 <- var(V1)

set.seed(1821)
for (i in 1:num.samp) {
  next_mu <- rnorm(1, y_bar, sqrt(next_sigma/n))
  gamma <- rgamma(1, n/2, (1/2)*((n-1)*s2 + n*(y_bar - next_mu)^2))
  next_sigma <- 1/gamma

  results3i.mu[i] <- next_mu; results3i.sigma[i] <- next_sigma
}

plot(results3i.mu[1:49], results3i.mu[2:50])
arrows(x0 = results3i.mu[1:25], y0 = results3i.mu[2:26],
       x1 = results3i.mu[2:26], y1 = results3i.mu[3:27], col='red')
```

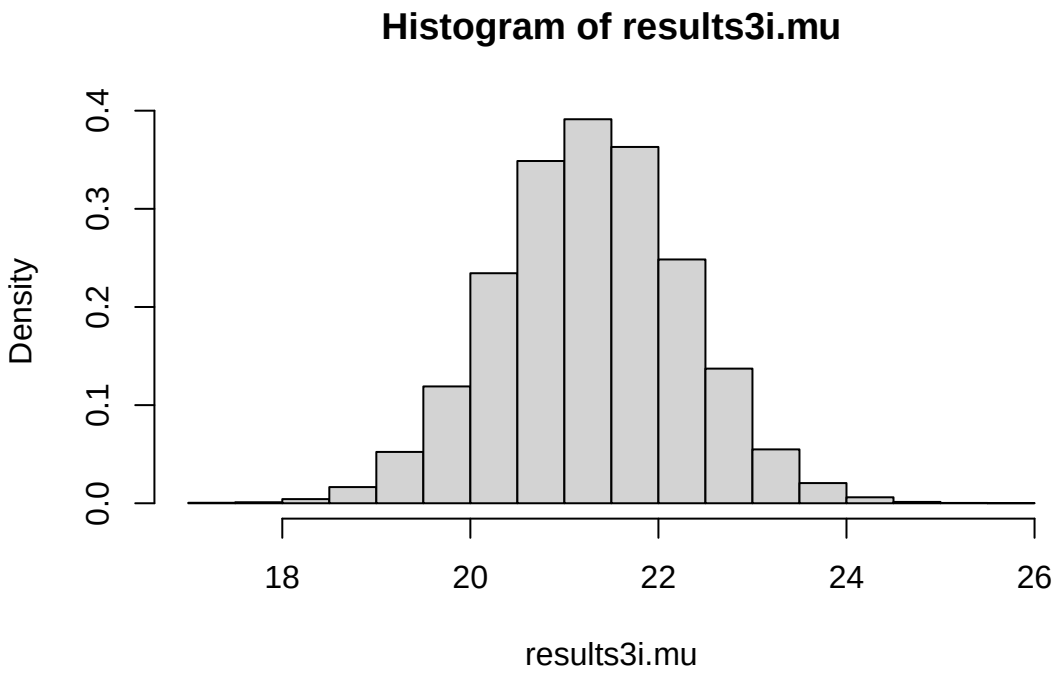



```
acf(results3i.mu)
```



After examining the plots, no burn-in or lag is necessary. The summary statistics are:

```r
## Mu Statistics
hist(results3i.mu, freq = F)
```



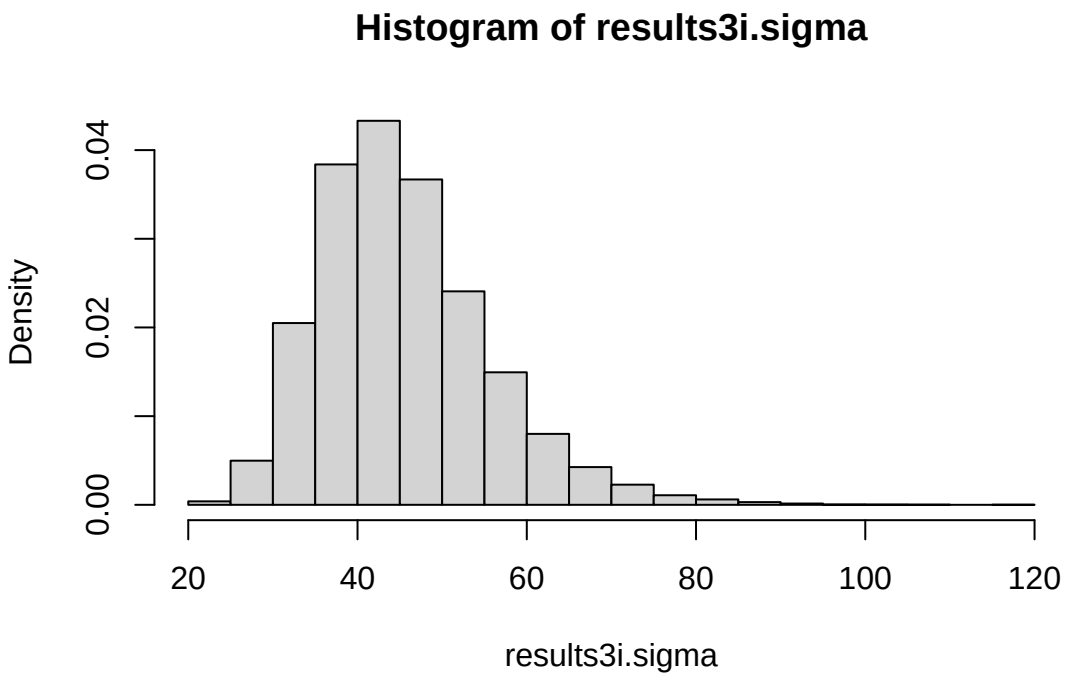

```r
quantile(results3i.mu, c(0.025, 0.975)) ## 95% CI
```

```
##     2.5%    97.5%
## 19.32856 23.24876
```

```r
quantile(results3i.mu, c(0.005, 0.995)) ## 99% CI
```

```
##     0.5%    99.5%
## 18.66830 23.91259
```

```r
## Sigma Statistics
hist(results3i.sigma, freq = F)
```




```
quantile(results3i.sigma, c(0.025, 0.975)) ## 95% CI
```

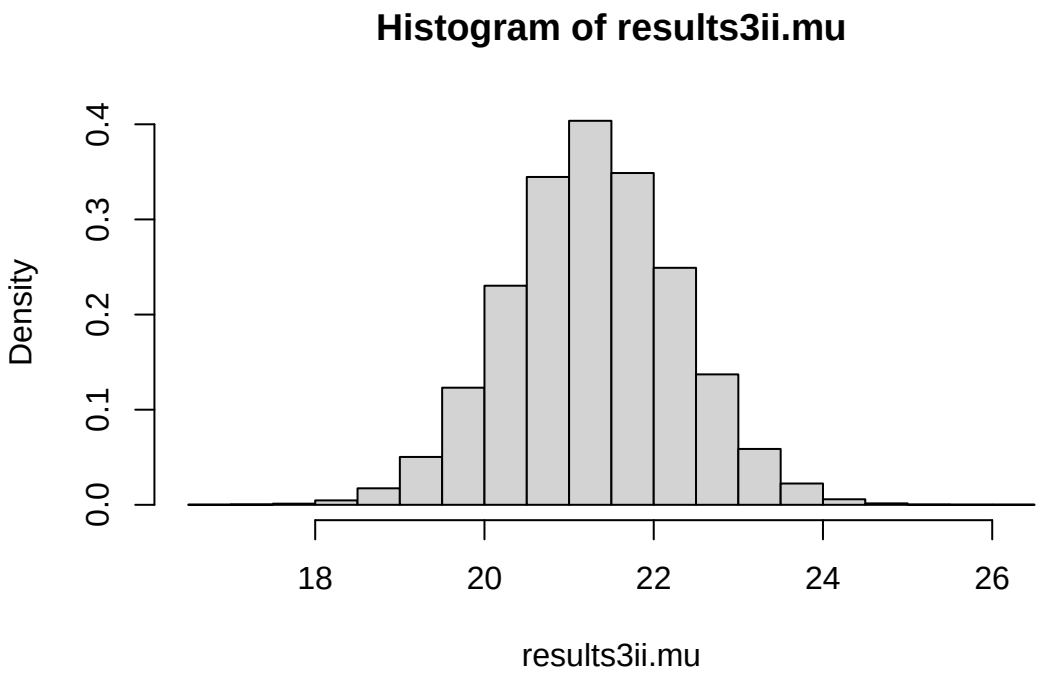
```
##     2.5%    97.5%
## 29.79541 69.16149
```

```
quantile(results3i.sigma, c(0.005, 0.995)) ## 99% CI
```

```
##     0.5%    99.5%
## 26.59360 80.83452
```

ii) The full conditional for μ and marginal posterior for σ^2 sampler

```
results3ii.mu <- rep(0, num.samp); results3ii.sigma <- rep(0, num.samp)

next_sigma = sd(V1); y_bar <- mean(V1); s2 <- var(V1) ## Initialize Sigma
set.seed(1835)
for (i in 1:num.samp) {
  gamma <- rgamma(1, (n-1)/2, (1/2)*(n-1)*s2)
  next_sigma <- 1/gamma
  next_mu <- rnorm(1, y_bar, sqrt(next_sigma/n))
  results3ii.mu[i] <- next_mu; results3ii.sigma[i] <- next_sigma
}

## Mu Statistics
hist(results3ii.mu, freq = F)
```



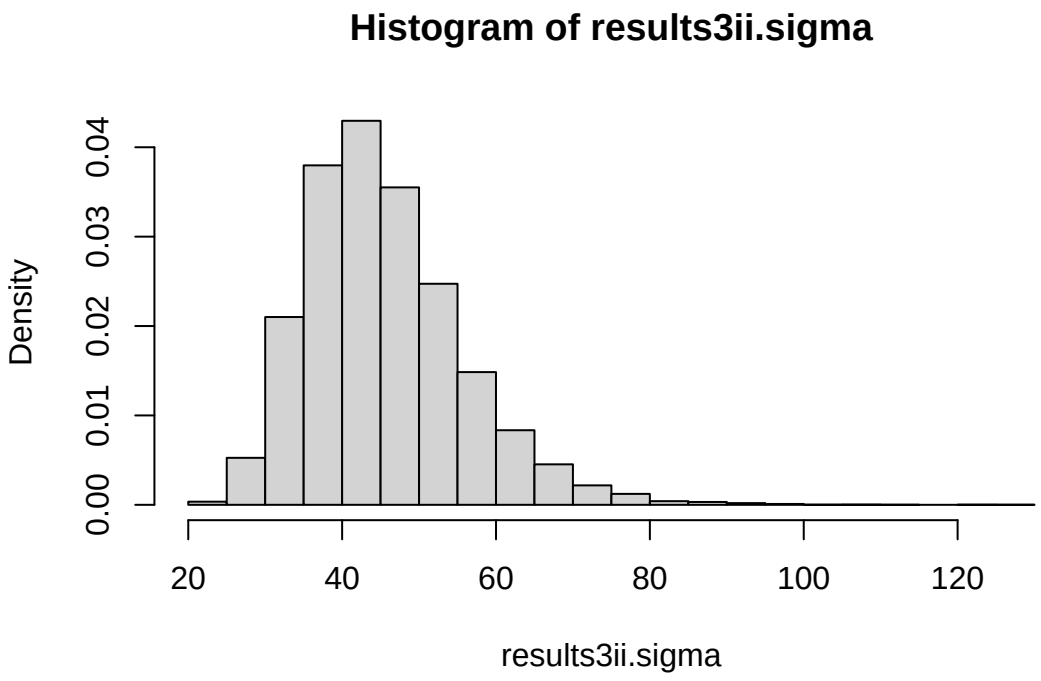

```
quantile(results3ii.mu, c(0.025, 0.975)) ## 95% CI
```

```
##     2.5%    97.5%
## 19.31430 23.25558
```

```
quantile(results3ii.mu, c(0.005, 0.995)) ## 99% CI
```

```
##     0.5%    99.5%
## 18.65586 23.88595
```

```
## Sigma Statistics
hist(results3ii.sigma, freq = F)
```




```
quantile(results3ii.sigma, c(0.025, 0.975)) ## 95% CI
```

```
##     2.5%    97.5%
## 29.69187 69.05148
```

```
quantile(results3ii.sigma, c(0.005, 0.995)) ## 99% CI
```

```
##     0.5%    99.5%
## 26.45838 80.76200
```

4) Sleeping Habits. Suppose we are interested in learning about the sleeping habits of students at a particular college. We collect $y_1, \ldots, y_20$, the sleeping times (in hours) for 20 randomly selected students in a statistics course. Here are the observations:

9.0	8.5	7.0	8.5	6.0	12.5	6.0	9.0	8.5	7.5
8.0	6.0	9.0	8.0	7.0	10.0	9.0	7.5	5.0	6.5

a) Assuming that the observations represent a random sample from a normal population with mean μ and variance σ^2 and the usual non-informative prior is placed on (μ, σ^2), simulate a sample of 1000 draws from the joint posterior distribution.

```
Y <- c(9.0, 8.5, 7.0, 8.5, 6.0, 12.5, 6.0, 9.0, 8.5, 7.5, 8.0, 6.0, 9.0, 8.0,
       7.0, 10.0, 9.0, 7.5, 5.0, 6.5)
y_bar <- mean(Y); s2 <- var(Y); n <- length(Y)

set.seed(721)
results4a.mu <- rep(0, 1000); results4a.sigma <- rep(0, 1000)
for (i in 1:length(results4a.mu)) {
  gamma <- rgamma(1, (n-1)/2, (1/2)*(n-1)*s2)
  sigma <- 1/gamma
  mu <- rnorm(1, y_bar, sqrt(sigma/n))
  results4a.mu[i] <- mu; results4a.sigma[i] <- sigma
}
```

b) Use the simulated sample to find 90% interval estimates (i.e. Bayesian credible intervals) for the mean μ and the standard deviation σ.

```
quantile(results4a.mu, c(0.05, 0.95)) ## 90% CI for Mu
```

```
##       5%      95%
## 7.274658 8.531150
```

```
quantile(results4a.sigma, c(0.05, 0.95)) ## 90% CI for Sigma
```

```
##       5%      95%
## 1.804430 5.603236
```

c) Suppose one is interested in estimating the upper quartile p_{75} of the normal population. Using the fact that $p_{75} = \mu + 0.674\sigma$, find the posterior mean and posterior standard deviation of p_{75}.

```
p_75 <- results4a.mu + 0.674 * results4a.sigma
mean(p_75); sd(p_75)
```

```
## [1] 10.13623
```

```
## [1] 0.9130532
```

5) The Behrens-Fisher Problem. Suppose that we observe two independent normal samples, the first distributed according to an $N(\mu_1, \sigma_1)$ distribution, and the second according to an $N(\mu_2, \sigma_2)$ distribution. Denote the first sample by $x_1, \ldots, x_m$ and the second sample by $y_1, \ldots, y_n$. Suppose also that the parameters $(\mu_1, \sigma_1^2, \mu_2\sigma_2^2)$ are assigned the vague prior $g(\mu_1, \sigma_1^2, \mu_2, \sigma_2^2) \propto (\sigma_1^2\sigma_2^2)$. Because the samples are independent, the posterior distribution of the vectors (μ_1, σ_1^2) and $(\mu_2\sigma_2^2)$ are independent as well. Here is the actual problem: The following data give the mandible lengths in millimeters for 10 male and ten female golden jackals in the collection of the British Museum.

Males	120	107	110	116	114	111	113	117	114	112
Females	110	111	107	108	110	105	107	106	111	111

Using the Gibbs sampler, find the posterior density of the difference in mean mandible length between the sexes. Is there sufficient evidence to conclude that the males have a larger average? Steps:

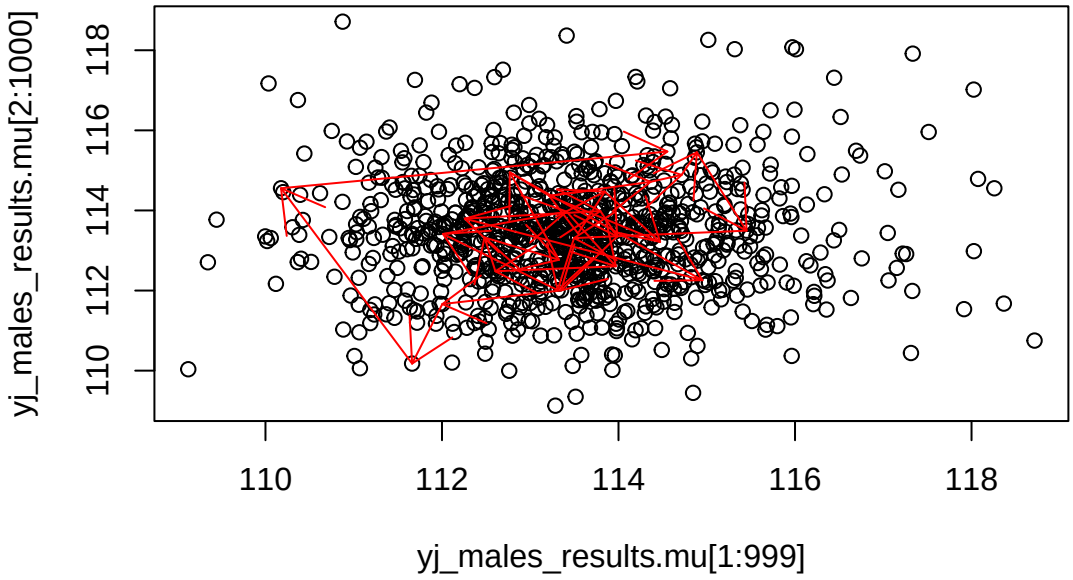
a) Use the Gibbs sampler to simulate values (independently) from both male and female golden jackets

```
yj_males <- c(120, 107, 110, 116, 114, 111, 113, 117, 114, 112)
yj_females <- c(110, 111, 107, 108, 110, 105, 107, 106, 111, 111)

m.y_bar <- mean(yj_males); m.s2 <- var(yj_males); m.n <- length(yj_males)
f.y_bar <- mean(yj_females); f.s2 <- var(yj_females); f.n <- length(yj_females)

## Simulate Males
set.seed(734)
next_sigma <- m.s2
yj_males_results.mu <- rep(0, 1000); yj_males_results.sigma <- rep(0, 1000)
for (i in 1:length(yj_males_results.mu)) {
  next_mu <- rnorm(1, m.y_bar, sqrt(next_sigma/m.n))
  gamma <- rgamma(1, (m.n/2), (1/2)*((m.n - 1)*m.s2 + m.n*(m.y_bar - next_mu)^2))
  next_sigma <- 1/gamma

  yj_males_results.mu[i] <- next_mu; yj_males_results.sigma[i] <- next_sigma
}

## Simulate Females
set.seed(742)
next_sigma <- f.s2
yj_females_results.mu <- rep(0, 1000); yj_females_results.sigma <- rep(0, 1000)
for (i in 1:length(yj_females_results.mu)) {
  next_mu <- rnorm(1, f.y_bar, sqrt(next_sigma/f.n))
  gamma <- rgamma(1, (f.n/2), (1/2)*((f.n - 1)*f.s2 + f.n*(f.y_bar - next_mu)^2))
  next_sigma <- 1/gamma

  yj_females_results.mu[i] <- next_mu; yj_females_results.sigma[i] <- next_sigma
}
```

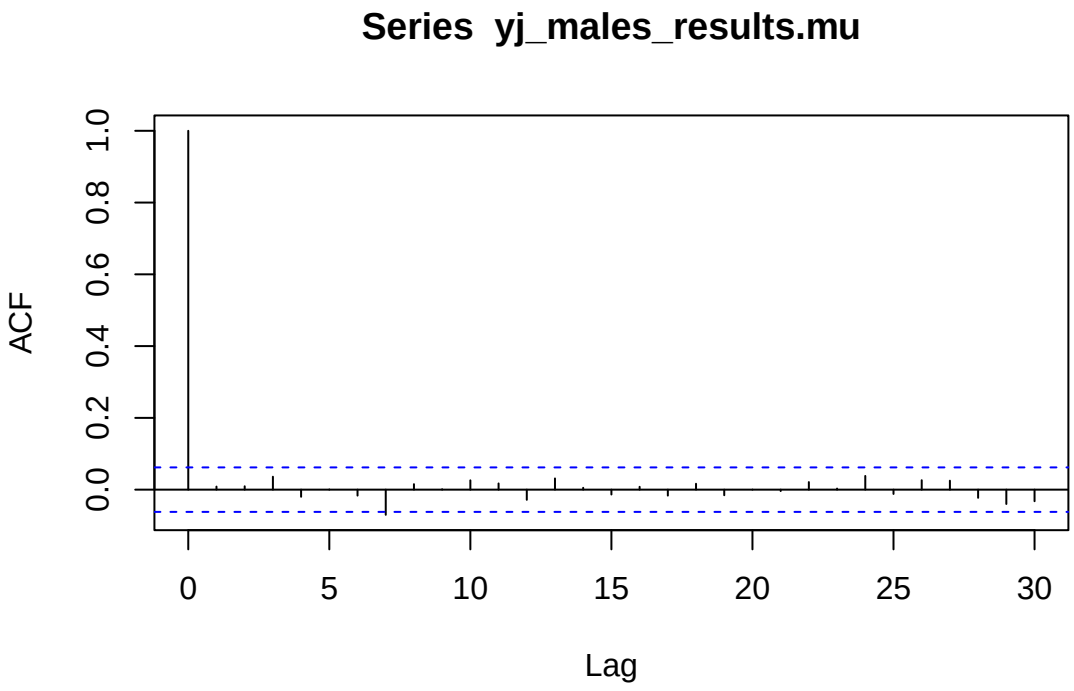
b) Focusing on the mean, and identify an appropriate burn-in and lag for each set of sampled values. Use the appropriate graphs to justify your choice for each data set. In the end, you'll need the same number of samples for both the male and female golden jackets, so use the larger of the two values for both the burn-in and lag and apply that to both sets of sampled values.

After plotting and checking the first 25 generated points for both data sets, there appears to be no need for a burn-in period. Both distributions were initialized and stayed rather consistent throughout the process. Similar results showed when testing for lag. Both acf plots show that the correlation is significant at lag 0 and insignificant everywhere else, thus no lag is necessary. For males, correlation was significant at lag 7, but insignificant from lag 1-6, thus this is likely a false positive. Same for the females at lag 4.

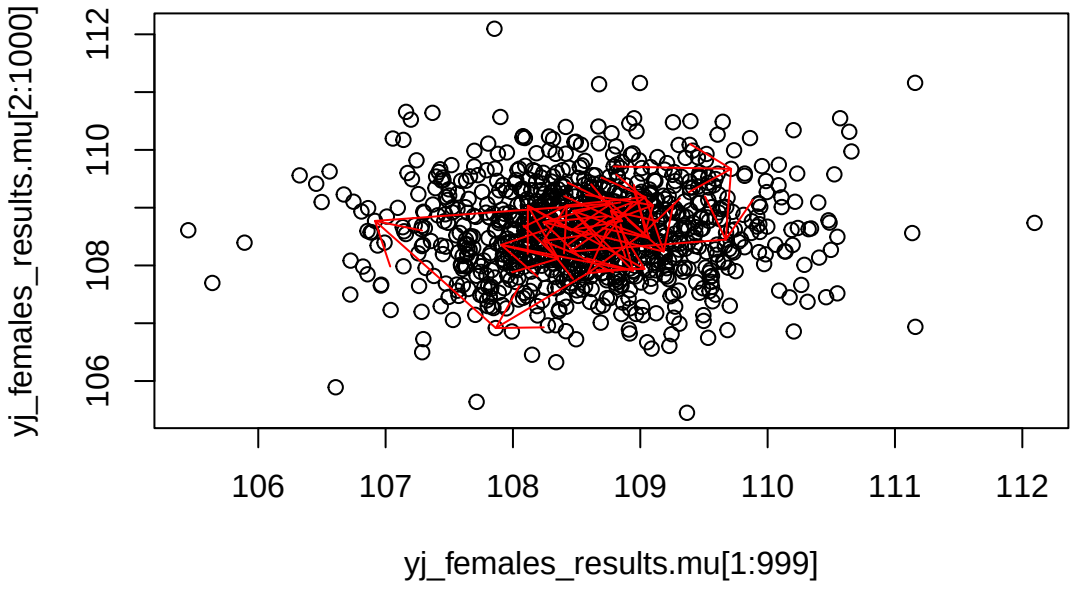
```
## Males
plot(yj_males_results.mu[1:999], yj_males_results.mu[2:1000])
arrows(x0 = yj_males_results.mu[1:25], y0 = yj_males_results.mu[2:26],
       x1 = yj_males_results.mu[2:26], y1 = yj_males_results.mu[3:27], col='red')
```



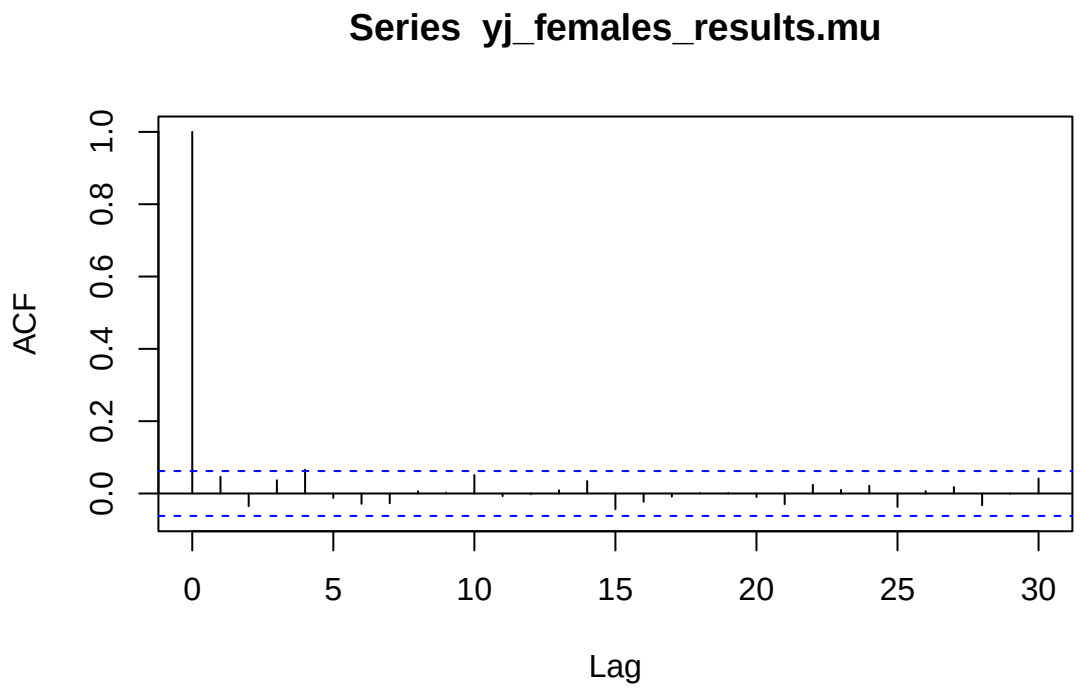

```
acf(yj_males_results.mu)
```




```
## Females
plot(yj_females_results.mu[1:999], yj_females_results.mu[2:1000])
arrows(x0 = yj_females_results.mu[1:25], y0 = yj_females_results.mu[2:26],
       x1 = yj_females_results.mu[2:26], y1 = yj_females_results.mu[3:27], col='red')
```

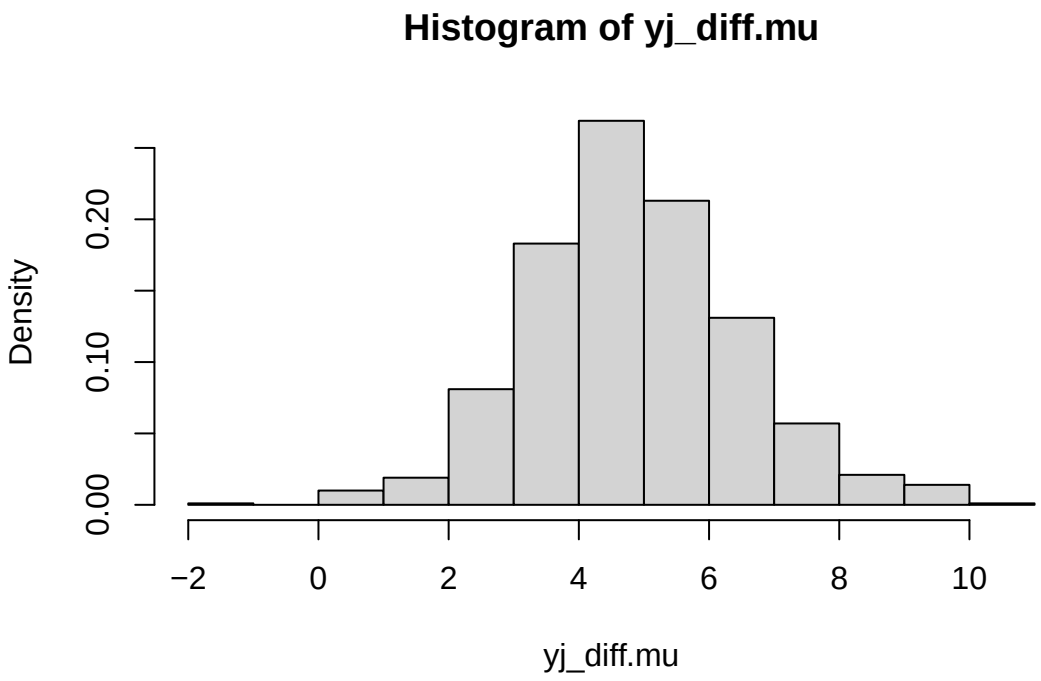


```
acf(yj_females_results.mu)
```



c) Calculate the difference between the two sets of posterior means. Make a histogram of these values, and then determine both the mean and a 95% Bayesian credible interval for the difference.

```
yj_diff.mu <- yj_males_results.mu - yj_females_results.mu

hist(yj_diff.mu, freq = F)
```




```
mean(yj_diff.mu)
```

```
## [1] 4.855904
```

```
quantile(yj_diff.mu, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 1.893779 8.461649
```

d) Calculate the probability that males have a larger average than females.

```
mean(yj_males_results.mu > yj_females_results.mu)
```

```
## [1] 0.999
```