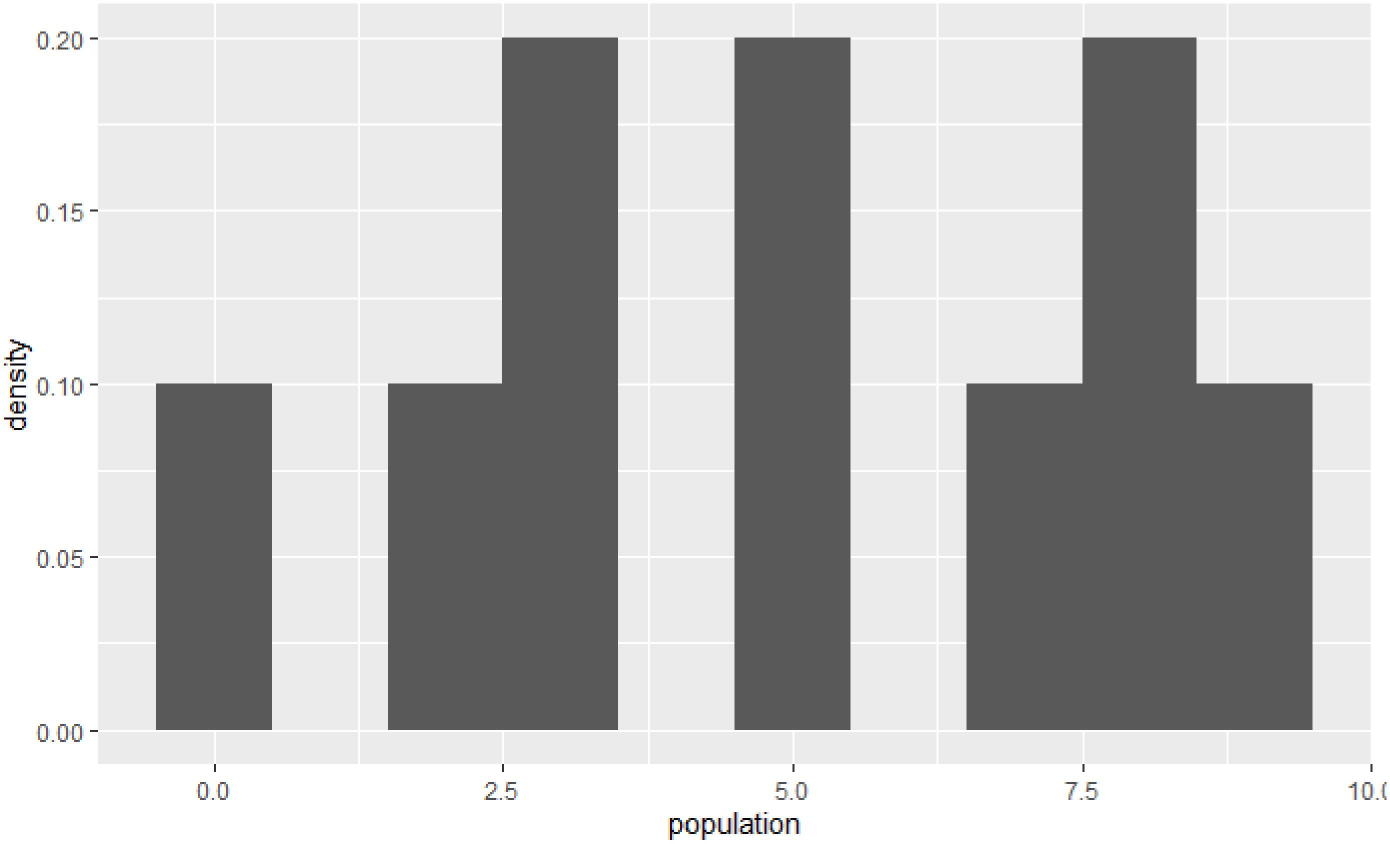
HW3 3F-J

Code ▾

Problem 3F

Hide

```
library(ggplot2)
population <- c(3,0,3,5,8,5,7,9,2,8)
df_population <- data.frame(population)
ggplot(df_population, aes(x=population)) + geom_histogram(aes(y=..density..),binwidth = 1)
```



Hide

```
population_mean <- mean(population)
population_mean
```

```
[1] 5
```

Hide

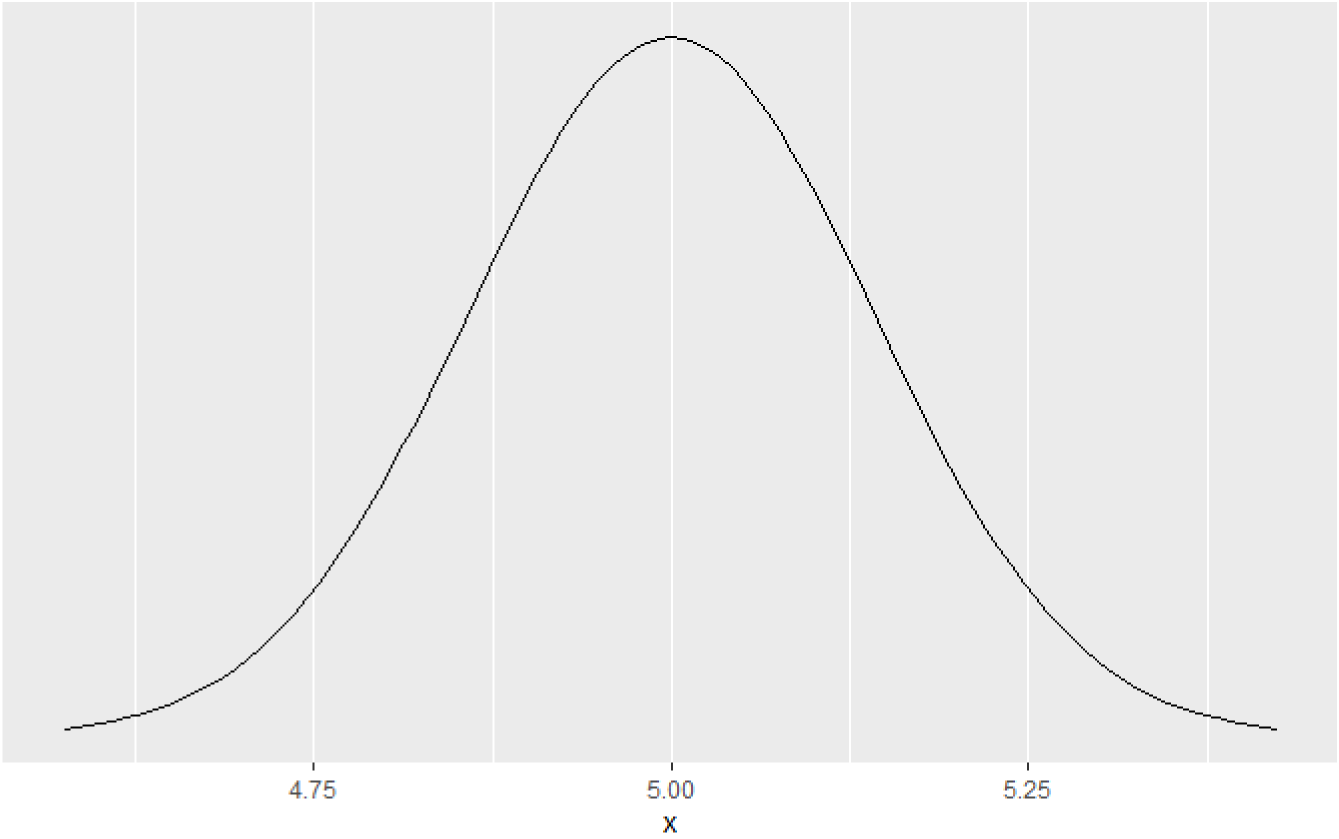
```
population_sd <- sqrt(mean((population-population_mean)^2))
population_sd
```

```
[1] 2.828427
```

Problem 3G

Hide

```
expected <- mean(population)
se <- population_sd/sqrt(400)
ggplot(data=data.frame(x=c(expected-3*se, expected+3*se)),aes(x))+ stat_function(fun= dnorm,n=10
1,args=list(mean= expected,sd= se))+ ylab("")+ scale_y_continuous(breaks=NULL)
```



Problem 3H

Hide

```
num_samples <- 100
sample_size <- 400

find_mean <- function(n){
  temp <- sample(population,size=n,replace=TRUE)
  mean(temp)}

means <- replicate(num_samples, find_mean(sample_size))
head(means)
```

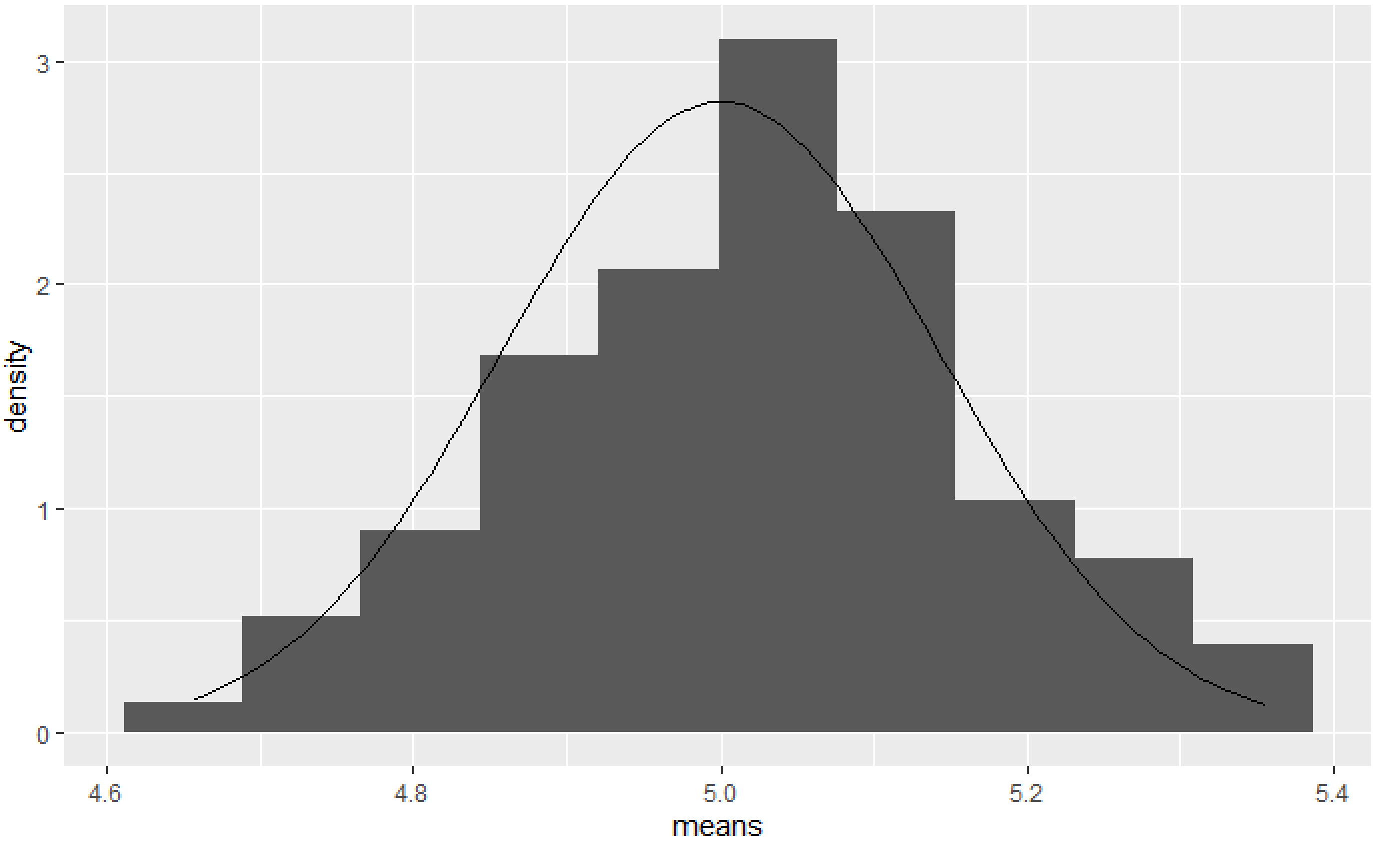
```
[1] 4.9300 5.0125 5.0500 5.0950 5.2925 4.9475
```

Hide

```
## [1] 4.9800 5.2100 4.9225 5.2950 5.0625 4.9675

new_data <- data.frame(means)

ggplot(data = new_data, aes(x = means)) +
  geom_histogram(aes(y = ..density..), bins=10) +
  stat_function(fun = dnorm, n = 101, args=list(mean = expected, sd = se))
```



Problem 3I

Hide

```
num_samples <- 100
sample_size <- 400

CI_range <- function(n){
  temp <- sample(population,size=n,replace=TRUE)
  CI_lower_bound <- mean(temp)-1.96*se
  CI_upper_bound <- mean(temp)+1.96*se
  ifelse(mean(population) < CI_upper_bound&mean(population) > CI_lower_bound, 1, 0)}

vec_good <- replicate(num_samples, CI_range(sample_size))
vec_good
```

```
  [1] 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1
 [70] 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Hide

```
num_good <- sum(vec_good)
num_good
```

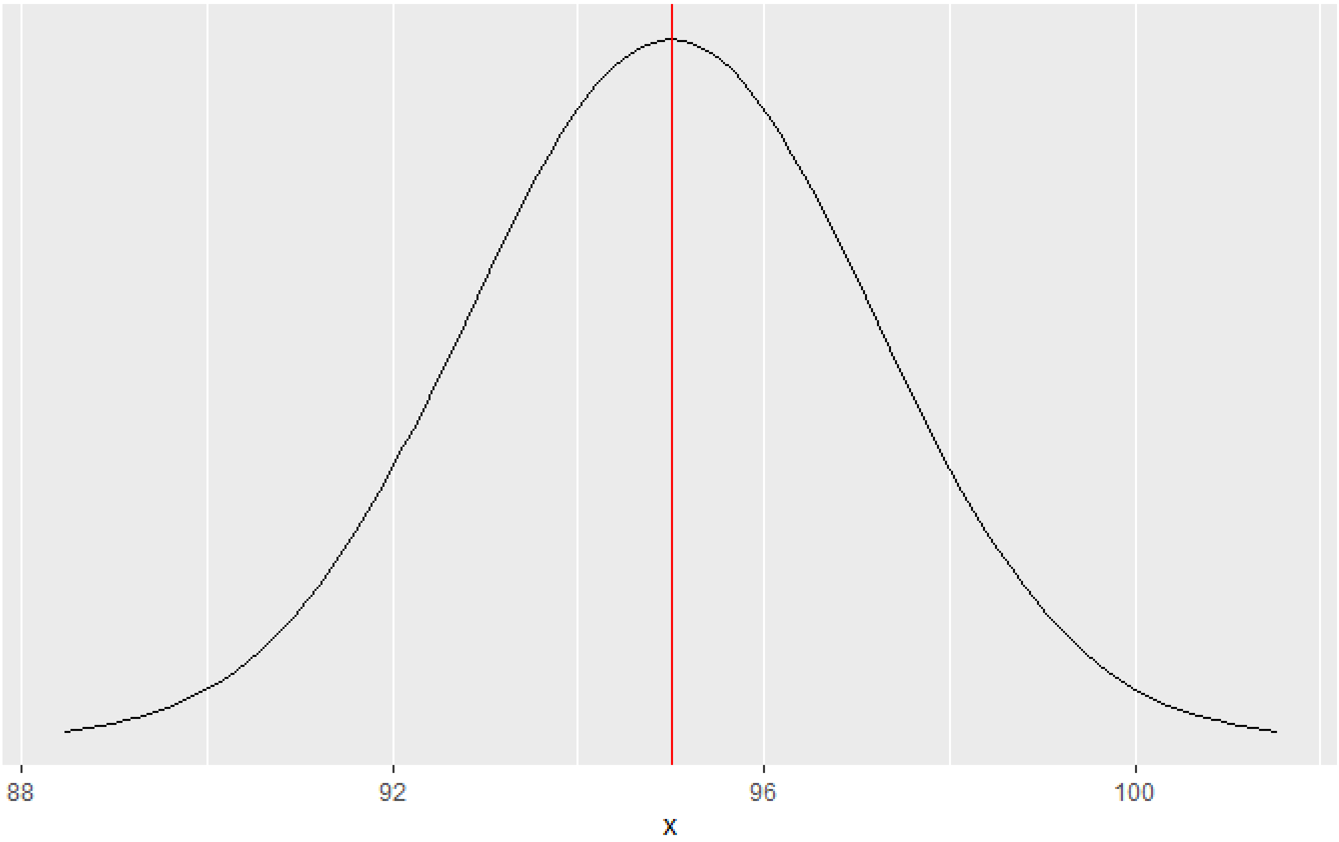
```
[1] 95
```

Problem 3J

Hide

```
expected = 100*(0.95)
se = sqrt(expected*(0.05))

ggplot(data = data.frame(x=c(expected-3*se, expected+3*se)), aes(x)) +
  stat_function(fun = dnorm, n = 101, args=list(mean=expected, sd=se)) +
  ylab("") +
  scale_y_continuous(breaks=NULL) +
  geom_vline(xintercept = sum(vec_good), col="red")
```



Based on the values of E(G) and SE(G), the mark looks exactly where it should be. It aligns directly in the middle of the normal distribution which, if all data is correct, is the most likely place for it to be. Overall, it is expected to be anywhere near the middle, which is what we see in this run of sampling.