HW0

Dun A

```
library(ggplot2)
library(magrittr)
library(ggpubr)
x <- c(175, 176, 182, 165, 167, 172, 175, 196, 158, 172)
```

Measures of center

1.0

#create own sample or use given vector and write mode, median, mean functions/one-liners

```
###   Mean
my_mean <- function(x){
  mean_result <-  sum(x)/length(x)
  print(mean_result)
}

## Median
my_median <- function(x){
  if ((length(x)) %% 2==0){
    x1 <- sort(x)
    central <- length(x)/2
    result <-(x1[central]+x1[central+1])/2
  } else {
    x1 <- sort(x)
    result <- x1[round(length(x)/2)]
  }
  return(result)
}


#check on vector with length
a <- c(175, 176, 182, 165, 167, 196, 158)
median(a)
```

```
## [1] 175
```

```
my_median(a)
```

```
## [1] 175
```

```
###   Mode
my_mode <- function(x) {
  x2 <- unique(x)
  x3 <- tabulate(match(x, x2))
  x2[x3 == max(x3)]
}


###### 1.1  #######
#calculate mode, median and mean for the sample. Compare results for own and  built-ins for median and 

my_mean(x)
```

```
## [1] 173.8
```

```
mean(x)
```

```
## [1] 173.8
```

```
print(my_median(x))
```

```
## [1] 173.5
```

```
median(x)
```

```
## [1] 173.5
```

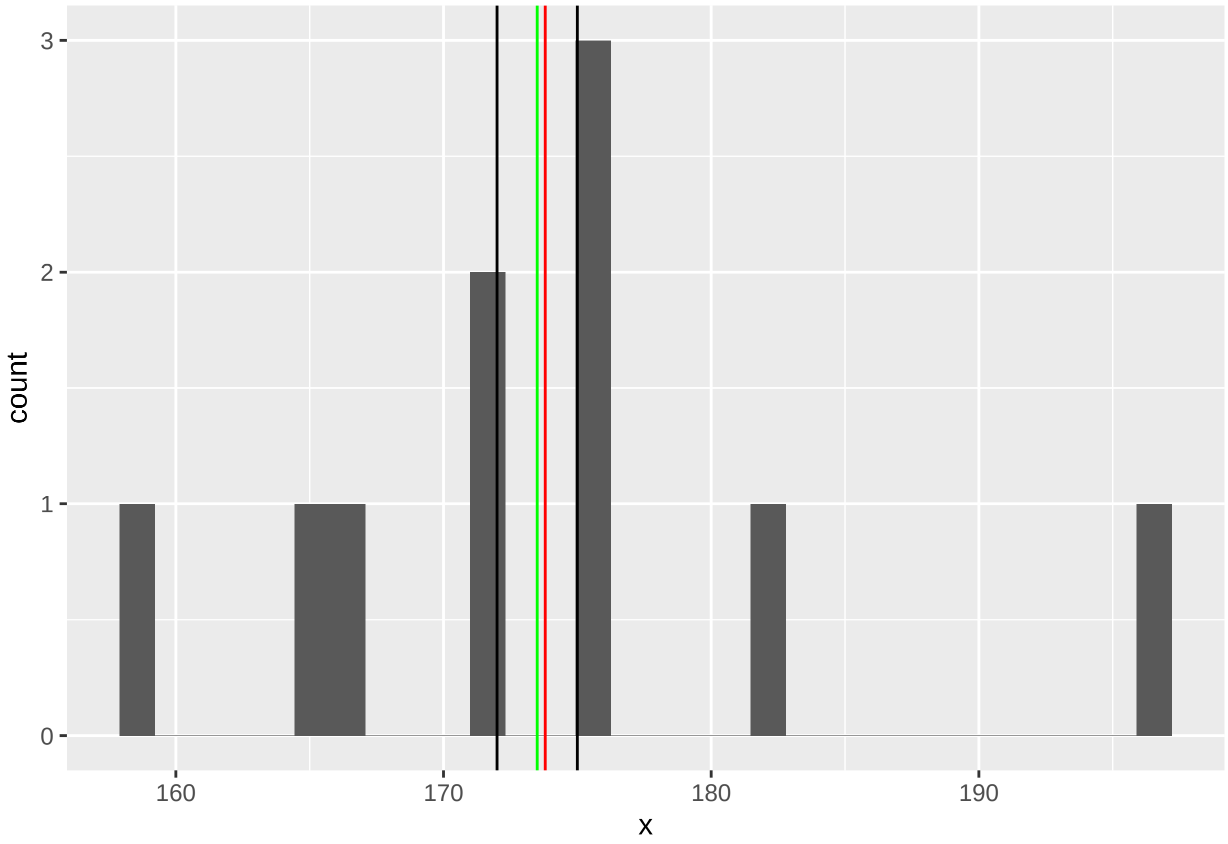
```
my_mode(x)
```

```
## [1] 175 172
```

```
####### 1.2 ########
#visualize histogram with 3 vertical lines for measures of center

ggplot(as.data.frame(x), aes(x)) +
  geom_histogram() +
  geom_vline(xintercept = my_mean(x), color = "red") +
  geom_vline(xintercept = my_median(x), color = "green") +
  geom_vline(xintercept = my_mode(x), color = "black")
```

```
## [1] 173.8
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
####### 1.3 ########
#spoil your sample with the outlier - repeat steps 1.1 and 1.2
x <- c(175, 176, 182, 165, 167, 172, 175, 196, 158, 172, 50, 250)
my_mean(x)
```

```
## [1] 169.8333
```

```
mean(x)
```

```
## [1] 169.8333
```

```
mean(x,trim=0.3)
```

```
## [1] 172.8333
```

```
print(my_median(x))
```

```
## [1] 173.5
```

```
median(x)
```

```
## [1] 173.5
```

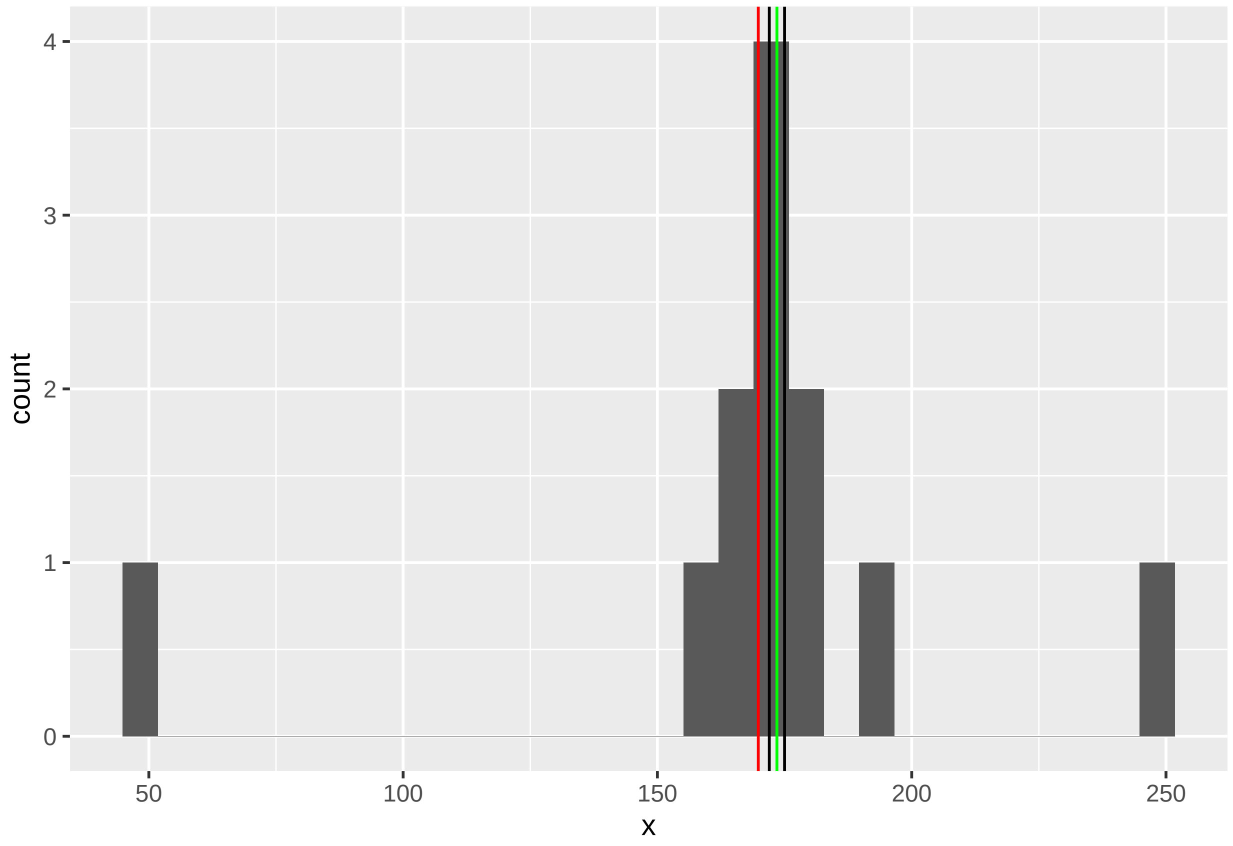
```
my_mode(x)
```

```
## [1] 175 172
```

```
ggplot(as.data.frame(x), aes(x)) +
  geom_histogram() +
  geom_vline(xintercept = my_mean(x), color = "red") +
  geom_vline(xintercept = my_median(x), color = "green") +
  geom_vline(xintercept = my_mode(x), color = "black")
```

```
## [1] 169.8333
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
########            Measurea of spread
```

```
#### 2.0 ######

# 2.0 write the functions/one-liners for variance and sd, calculate result, compare with the built-ins
x <- c(175, 176, 182, 165, 167, 172, 175, 196, 158, 172)

variance <- function(x){
  return(sum((x - mean(x))^2)/(length(x)-1))
```

```r
}

std <- function(x){
  return(sqrt(sum((x - mean(x))^2)/(length(x)-1)))
}

variance(x)
```

```
## [1] 105.2889
```

```r
var(x)
```

```
## [1] 105.2889
```

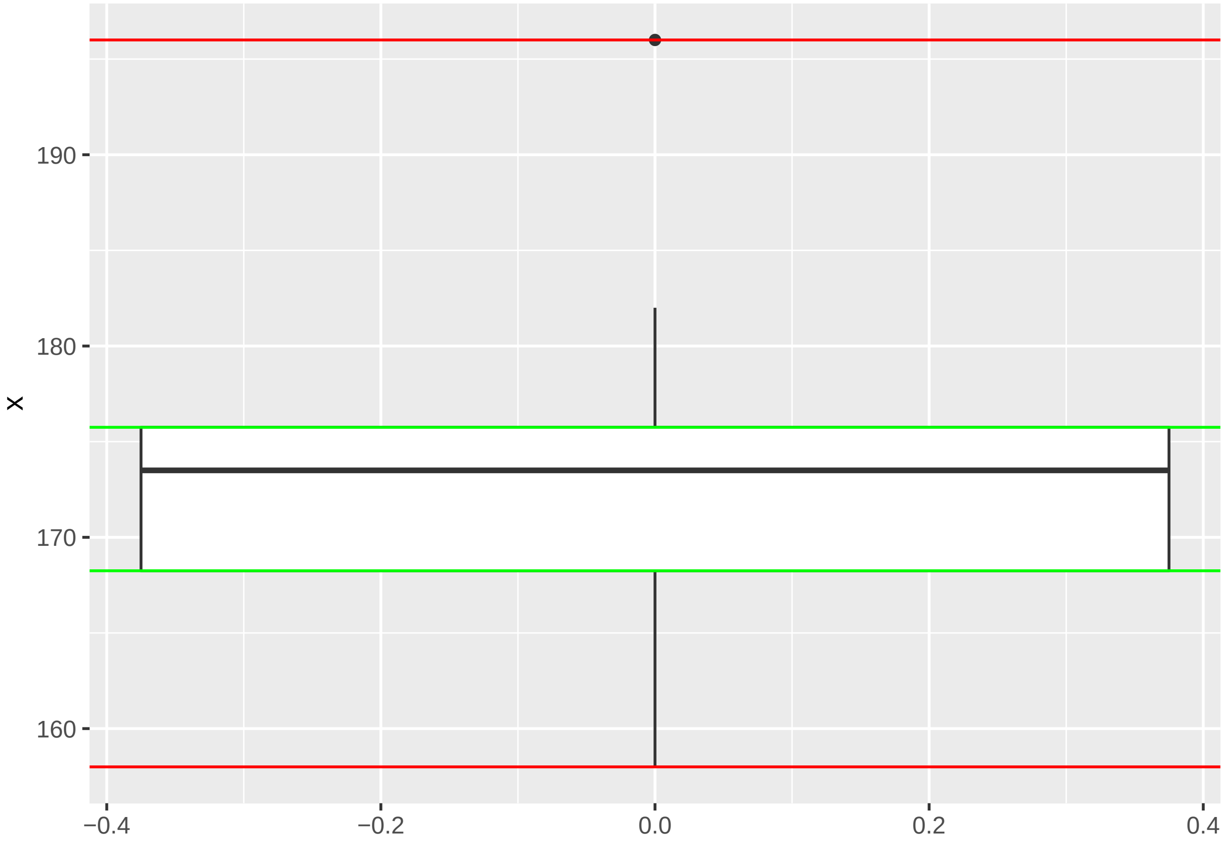
```r
std(x)
```

```
## [1] 10.26104
```

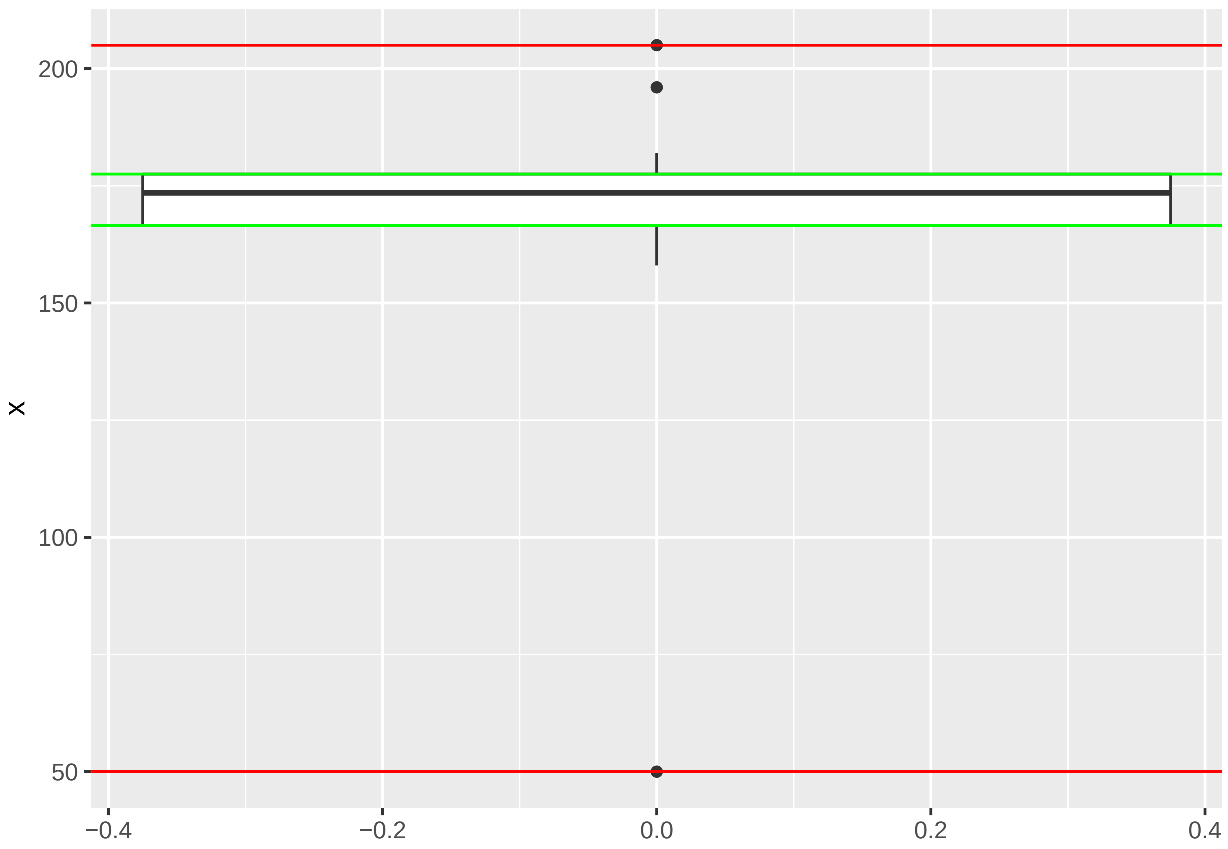
```r
sd(x)
```

```
## [1] 10.26104
```

```r
########  2.1 #####
#visualize with the box plot and add horizontal lines for range, IQR, 1-sd borders (use built-ins)

ggplot(as.data.frame(x), aes(y = x)) +
  geom_boxplot() +
  geom_hline(yintercept = min(x), color = 'red') +
  geom_hline(yintercept = max(x), color = 'red') +
  geom_hline(yintercept = quantile(x, 3/4), color = 'green') +
  geom_hline(yintercept = quantile(x, 1/4), color = 'green')
```



```
####### 2.2######## spoil your sample with the outlier, repeat step 2.1
x <- c(175, 176, 182, 165, 167, 172, 175, 196, 158, 172, 50, 205)
ggplot(as.data.frame(x), aes(y = x)) +
  geom_boxplot() +
  geom_hline(yintercept = min(x), color = 'red') +
  geom_hline(yintercept = max(x), color = 'red') +
  geom_hline(yintercept = quantile(x, 3/4), color = 'green') +
  geom_hline(yintercept = quantile(x, 1/4), color = 'green')
```



```
##### 3 ####### properties
x <- c(175, 176, 182, 165, 167, 172, 175, 196, 158, 172)

# 3.0 check the properties for mean and sd for your sample
mean(x-100) == mean(x) - 100
```

```
## [1] FALSE
```

```
mean(x / 100) == mean(x) / 100
```

```
## [1] FALSE
```

```
abs(sum(x - mean(x)) - 0) < 0.000000001
```

```
## [1] TRUE
```

```
var(x - 100) == var(x)
```

```
## [1] TRUE
```

```
var(x / 100) == var(x) / 10000
```

```
## [1] FALSE
```

```
sd(x / 100) == sd(x) / 100
```

```
## [1] FALSE
```

```
# 3.1 visualize result tabularly and graphically (maybe with facetting free scales?)
#tabularly

properties <- as.table(matrix(c(mean(x), mean(x - 100), mean(x/ 100),
                              var(x), var(x - 100),var(x / 100),
                              sd(x), sd(x - 100), sd(x / 100)), ncol = 3, byrow = TRUE))
colnames(properties) <- c("x","x-100","x/100")
rownames(properties) <- c("Mean","Variance","SD")
properties
```

```
##                     x        x-100        x/100
## Mean     173.80000000  73.80000000   1.73800000
## Variance 105.28888889 105.28888889   0.01052889
## SD        10.26103742  10.26103742   0.10261037
```

```
#graphically
library(dplyr)
library(ggplot2)



##### 4 ####### normal distribution


# 4.0 for the population N(175, 10) find the probability to be:
# less than 156cm,
pnorm(156,175,10)
```

```
## [1] 0.02871656
```

```
# more than 198,
pnorm(198,175,10,lower.tail = FALSE)
```

```
## [1] 0.01072411
```

```
# between 168 and 172 cm
pnorm(168,175,10,lower.tail = FALSE)-pnorm(172,175,10,lower.tail = TRUE)
```

```
## [1] 0.3759478
```

```
## Standard normal distribution
# 4.1 check the properties of 1-2-3-sd's for standard normal distribution using pnorm()

## Standardization
# 4.2 generate sample using rnorm() from N(175, 10), find mean ans sd;
set.seed(1)
y = rnorm(10000,175,10)
mean(y)
```

```
## [1] 174.9346
```

```
sd(y)
```

```
## [1] 10.12356
```

```
# 4.3 standardize, find the same
y1 = (y-mean(y))/sd(y)
mean(y1)
```

```
## [1] 1.142693e-15
```

```
sd(y1)
```

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

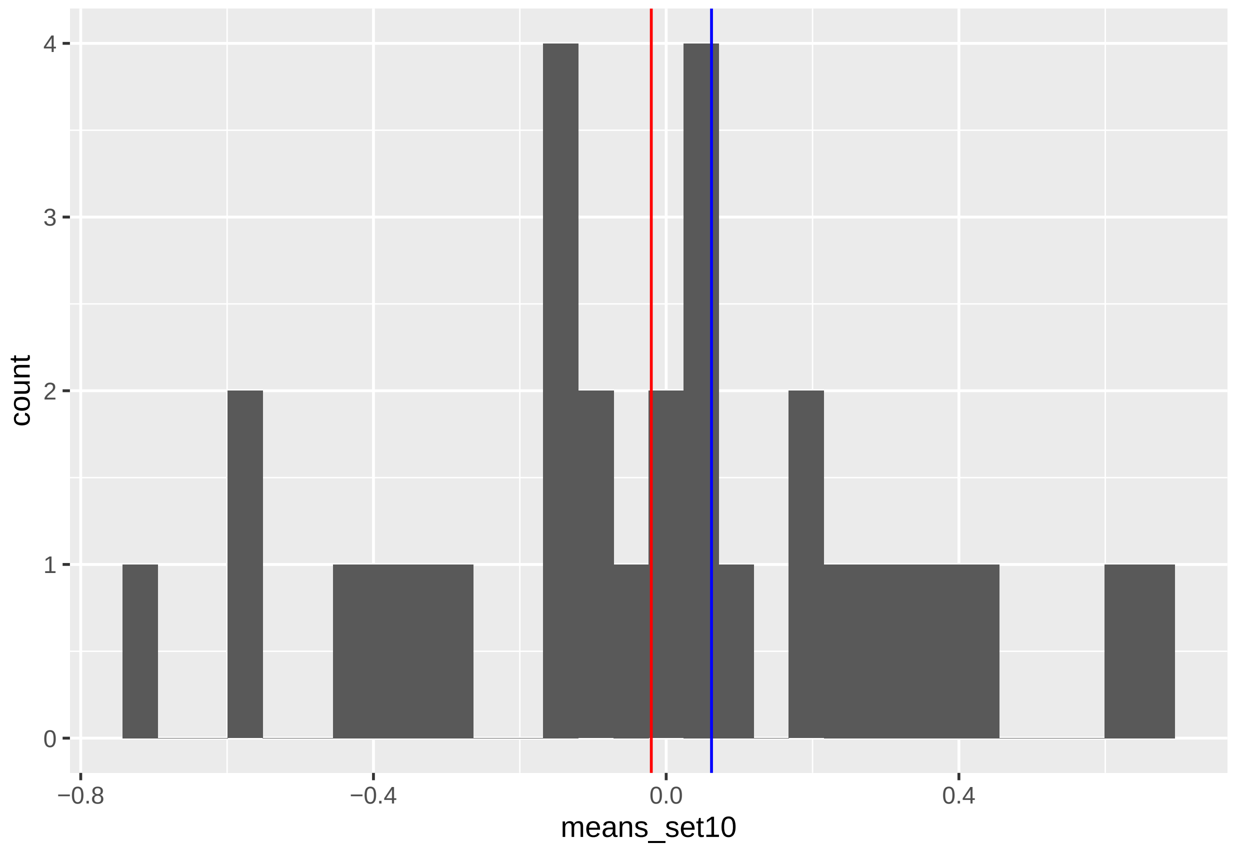
```
##### 5 #######
##Central limit theorem
set.seed(42)
b <-  rnorm(1e6,0,1)

# 5.0 Generate large population (n ~ 100 000 - 1 000 000) distributed as N(0, 1)
set.seed(42)
pop <-  rnorm(1e6,0,1)
mean(pop)
```

```
## [1] 0.0005737398
```

```
# Sample from population k observations for 30 times - you will have set of 30 samples.

set10 <- replicate(30, sample(pop,10))
set50 <-  replicate(30, sample(pop,50))
set100 <-  replicate(30, sample(pop,100))
set500 <-  replicate(30, sample(pop,500))

# For each sample calculate mean. For the set calculate means of means, sd of means, SE.
means_each_sample <- function(set){
  mean_i <- c()
  for (i in 1:ncol(set)) {
    mean_i[i] <- mean(set[,i])
  }
  return(mean_i)
}

SE <- function(means){
  return(sd(means)/sqrt(length(means)))
}


means_set10 <- means_each_sample(set10)
mean_of_means10 <- mean(means_set10)
```

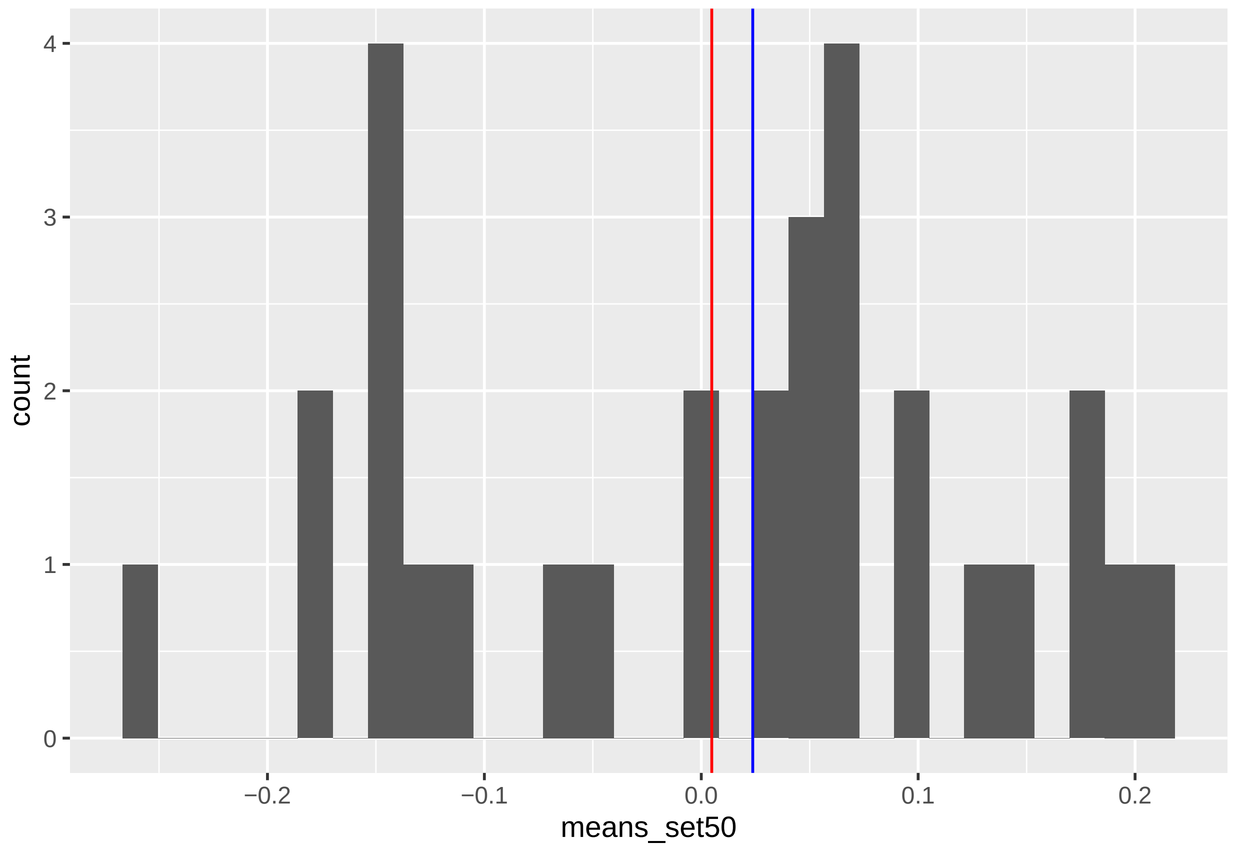
```r
sd10 <- sd(means_set10)
SE10 <- SE(means_set10)

means_set50 <- means_of_means(set50)
mean_of_means50 <- mean(means_set50)
sd50 <- sd(means_set50)
SE50 <- SE(means_set50)

means_set100 <- means_of_means(set100)
mean_of_means100 <- mean(means_set100)
sd100 <- sd(means_set100)
SE100 <- SE(means_set100)

means_set500 <- means_of_means(set500)
mean_of_means500 <- mean(means_set500)
sd500 <- sd(means_set500)
SE500 <- SE(means_set500)

# Create table with k, mean of means, sd of means, SE.
k <- c(10,50,100,500)
means_of_mean <- c(mean_of_means10,mean_of_means50,mean_of_means100,mean_of_means500)
sd_means <- c(sd10,sd50,sd100,sd500)
SE_means <- c(SE10,SE50,SE100, SE500)
table_clt <- data.frame(k, means_of_mean, sd_means,SE_means )
# Visualize distribution of means with histogram and lines for mean of means and SE
# 5.1 k = 10
ggplot(as.data.frame(means_set10), aes(means_set10)) +
  geom_histogram() +
  geom_vline(xintercept = mean(means_set10), color = "red") +
  geom_vline(xintercept = SE(means_set10), color = "blue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



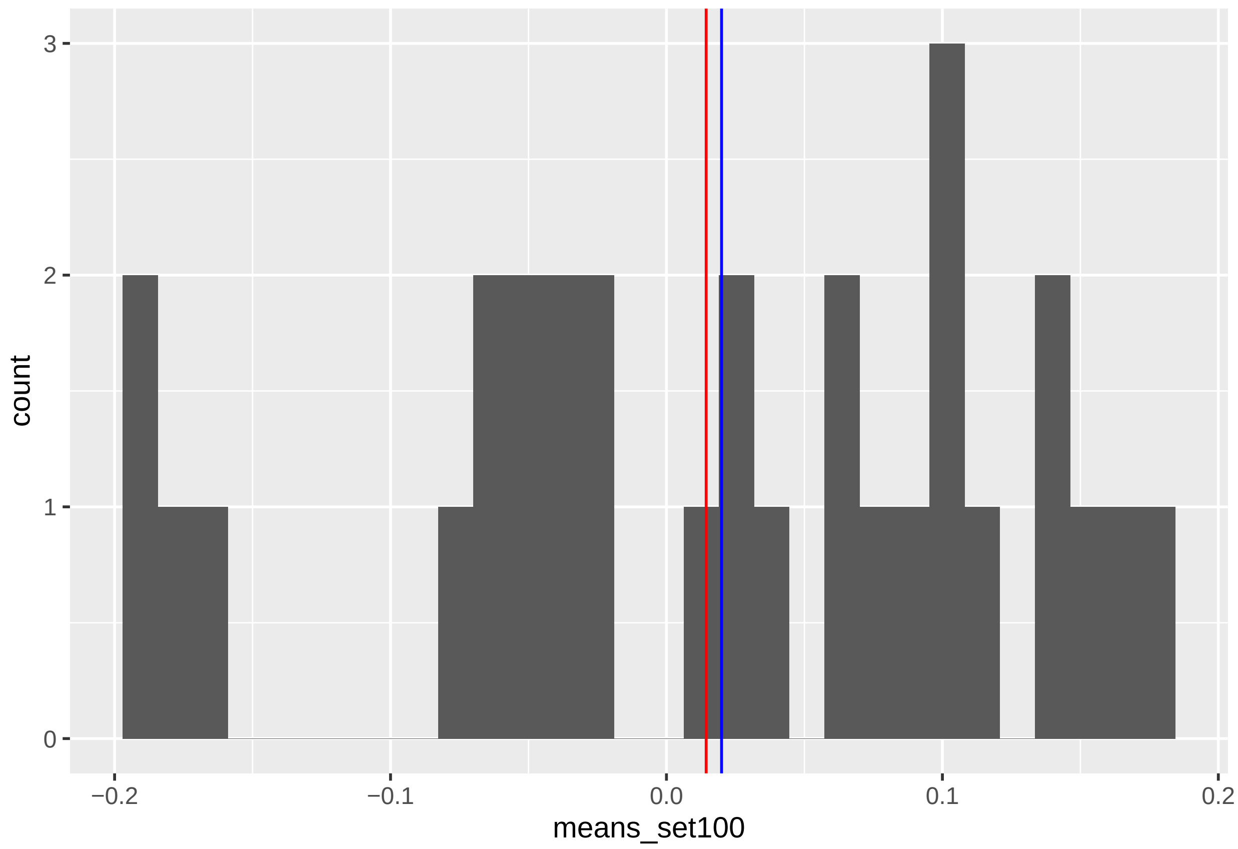
```
# 5.2 k = 50
ggplot(as.data.frame(means_set50), aes(means_set50)) +
  geom_histogram() +
  geom_vline(xintercept = mean(means_set50), color = "red") +
  geom_vline(xintercept = SE(means_set50), color = "blue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



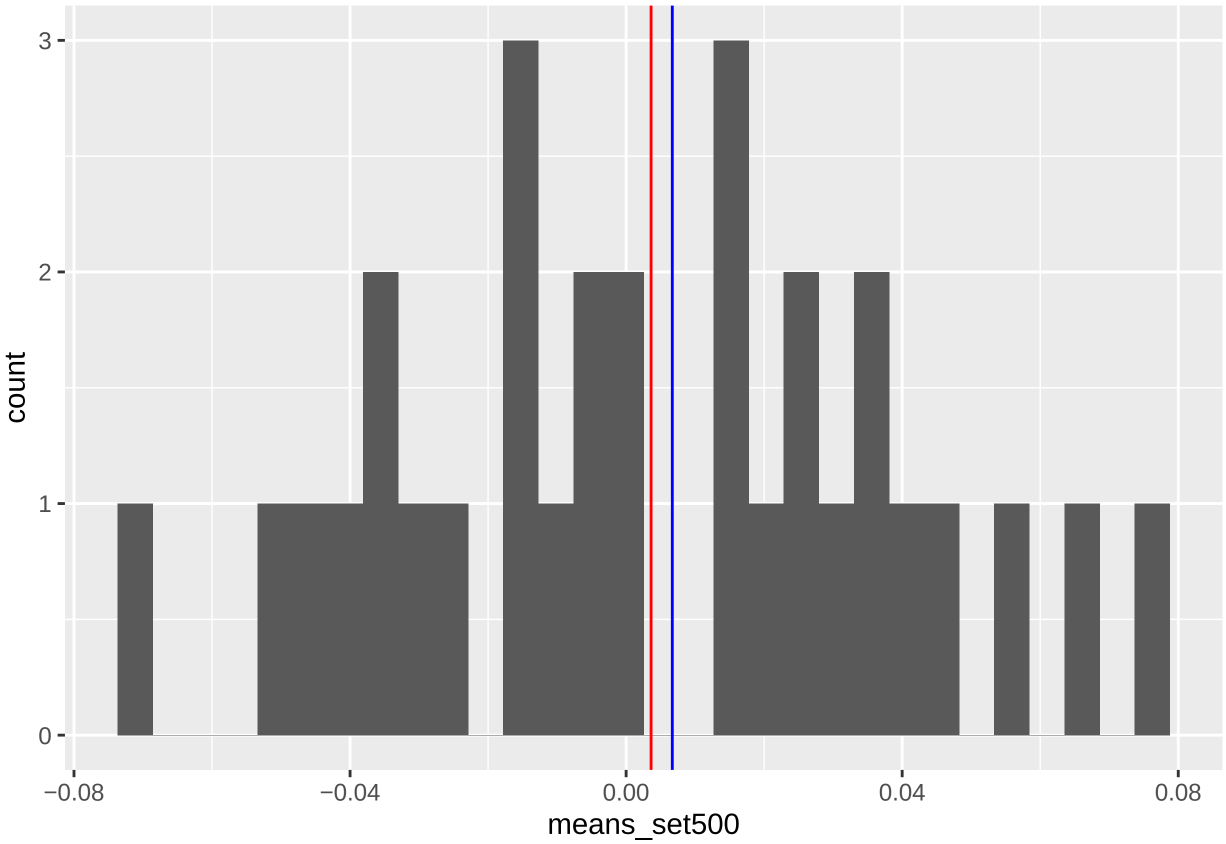
```
# 5.3 k = 100
ggplot(as.data.frame(means_set100), aes(means_set100)) +
  geom_histogram() +
  geom_vline(xintercept = mean(means_set100), color = "red") +
  geom_vline(xintercept = SE(means_set100), color = "blue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# 5.4 k = 500
ggplot(as.data.frame(means_set500), aes(means_set500)) +
  geom_histogram() +
  geom_vline(xintercept = mean(means_set500), color = "red") +
  geom_vline(xintercept = SE(means_set500), color = "blue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



# 5.5 Compare results #the larger the number of observations the lesser SE and sd #And, moreover, the actual mean is coming closer tot the population mean rmarkdown::render('file.rmd', output_format = 'html_document')