```
library('ggplot2')
library('ggpubr')
```

```
## Loading required package: magrittr
```

```
our_sample <- c(175, 176, 182, 165, 167, 172, 175,196, 158, 172)
```

Measures of center

Mode, Median, Mean functions

```
mean(our_sample)
```

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

```
my_mean <- function(x){
  print(sum(x)/length(x))
}
my_mean(our_sample)
```

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

mode with trimming

```
mean(our_sample, trim = 0.1)
```

```
## [1] 173
```

```
median(our_sample)
```

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

```
my_median <- function(x){
if (length(x) %% 2 == 0) {
  print((sort(x)[length(x)/2] + sort(x)[(length(x)/2) + 1])/2)
}
  else {
    print((sort(x)[(length(x)/2) + 1]))
    }
}
my_median(our_sample)
```

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

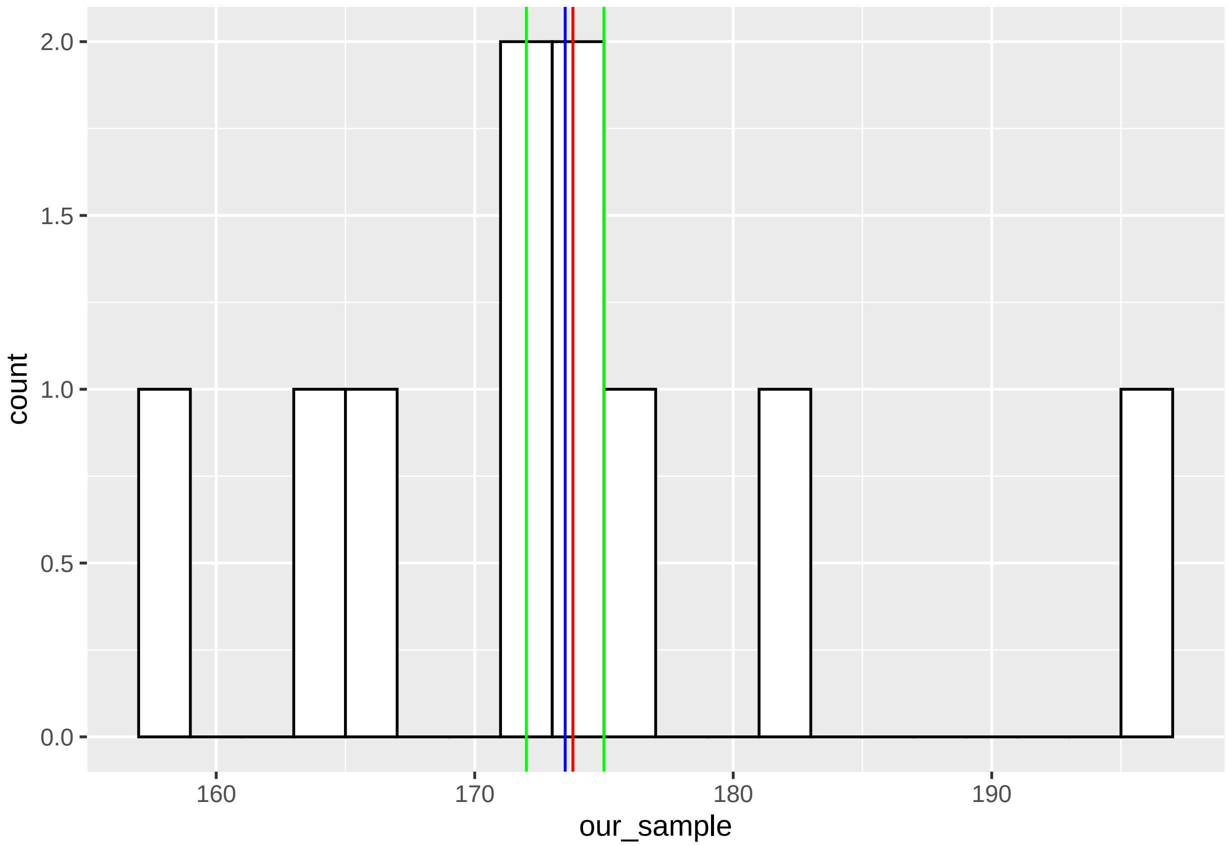
```
my_mode <- function(x) {
t <- table(x)
print(as.numeric(names(t[t == max(t)])))
}
my_mode(our_sample)
```

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

Histogram

```
ggplot() +
  aes(our_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=mean(our_sample), color="red") +
  geom_vline(xintercept=median(our_sample), color="blue") +
  geom_vline(xintercept=my_mode(our_sample), color="green")
```

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



Sample with the outlier

```
our_spoil_sample <- c(175, 176, 182, 165, 167, 172, 175,196, 158, 172, 235)
```

```
mean(our_spoil_sample)
```

```
## [1] 179.3636
```

```
my_mean(our_spoil_sample)
```

```
## [1] 179.3636
```

```
median(our_spoil_sample)
```

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

```
my_median(our_spoil_sample)
```

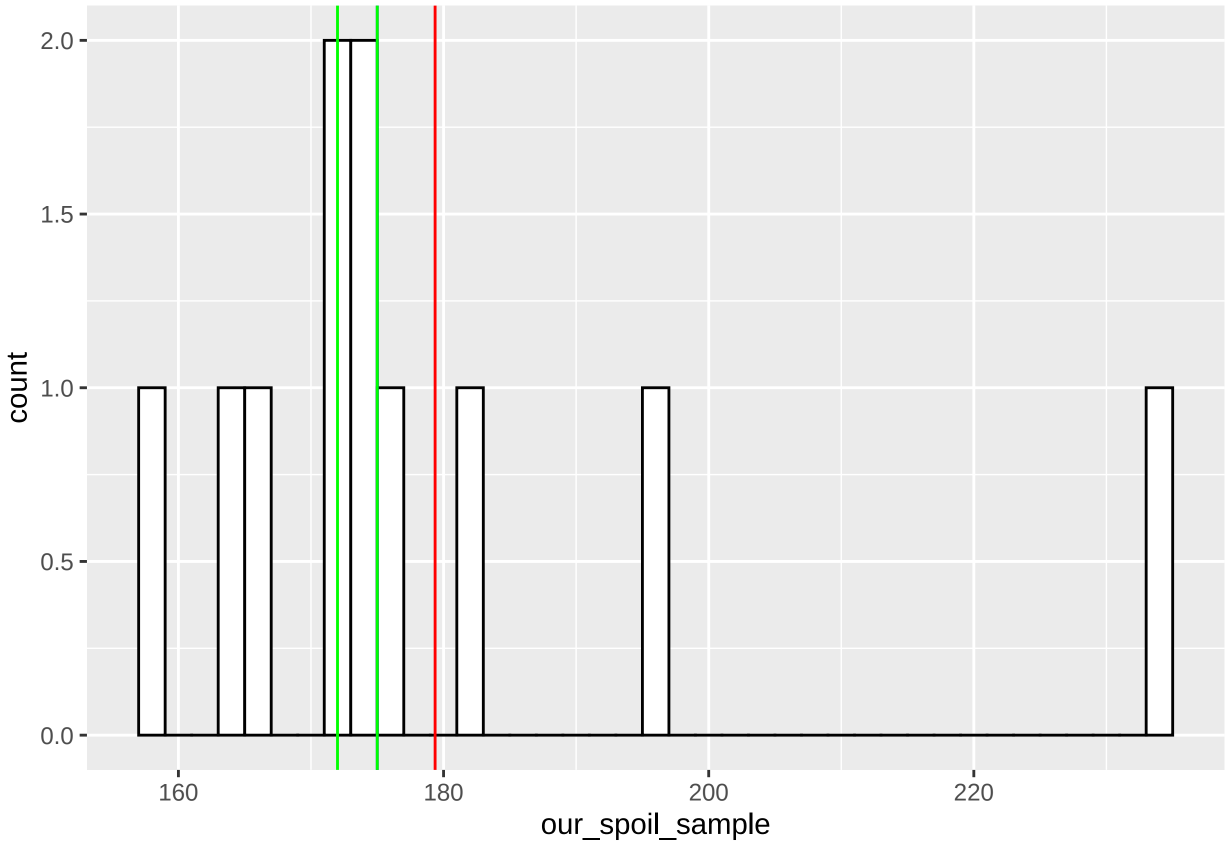
```
## [1] 175
```

```
my_mode(our_spoil_sample)
```

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

```
ggplot() +
  aes(our_spoil_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=mean(our_spoil_sample), color="red") +
  geom_vline(xintercept=median(our_spoil_sample), color="blue") +
  geom_vline(xintercept=my_mode(our_spoil_sample), color="green")
```

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



as median is the same as one of the mode value - there are only three lines

Measures of spread

Variance and Sd functions

R uses variance for unbiased estimators, I made the same

```
var(our_sample)
```

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

```
my_var <- function(x){
 n <- sapply(x, function(a)(a - mean(x))^2)
 print(sum(n)/(length(x)-1))
}
my_var(our_sample)
```

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

```
sd(our_sample)
```

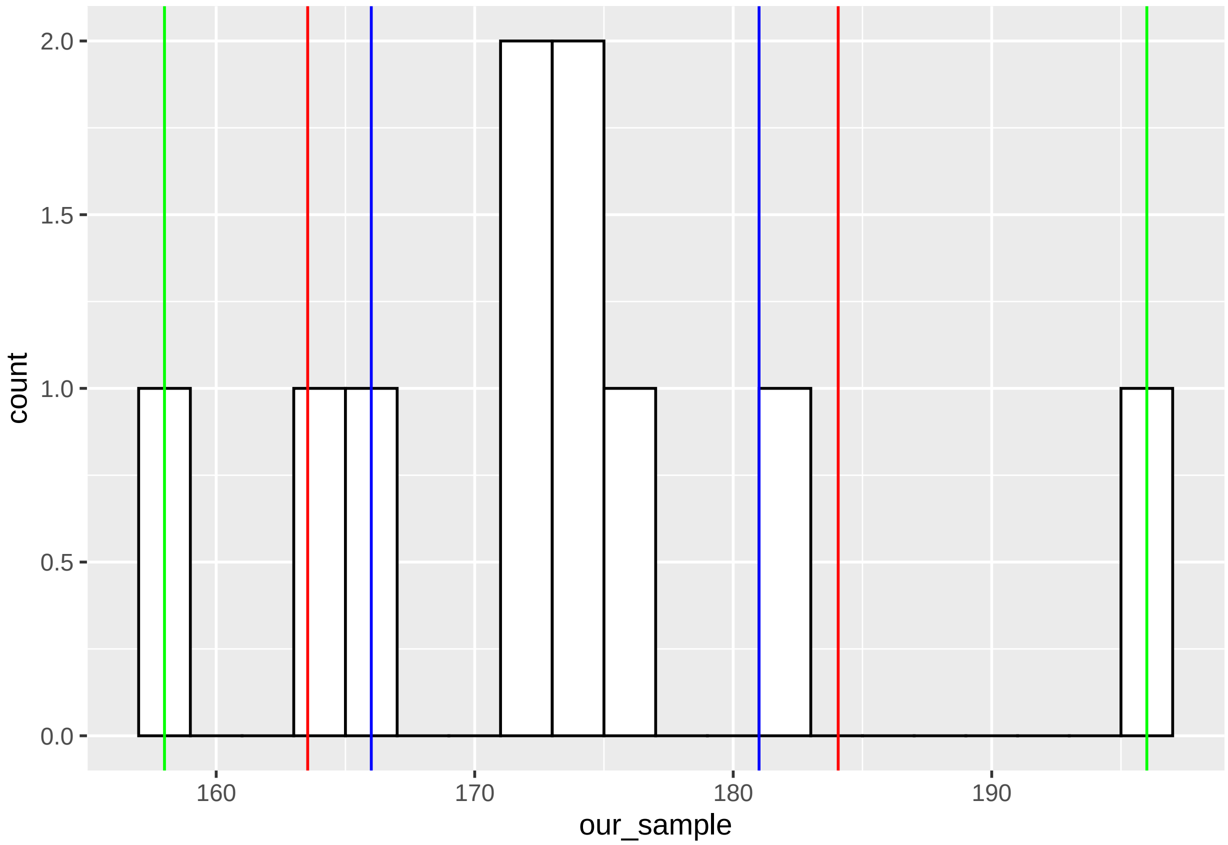
```
## [1] 10.26104
```

```r
my_sd <- function(x){
 n <- sapply(x, function(a)(a - mean(x))^2)
 print((sum(n)/(length(x)-1))^0.5)
}
my_sd(our_sample)
```

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

Histogram

```r
ggplot() +
  aes(our_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=c(median(our_sample) +
                            IQR(our_sample),median(our_sample) - IQR(our_sample)), color="blue") +
  geom_vline(xintercept=c(mean(our_sample) +
                            sd(our_sample),mean(our_sample) - sd(our_sample)), color="red") +
  geom_vline(xintercept=range(our_sample), color="green")
```



Sample with the outlier

```
var(our_spoil_sample)
```

```
## [1] 435.2545
```

```
my_var(our_spoil_sample)
```

```
## [1] 435.2545
```

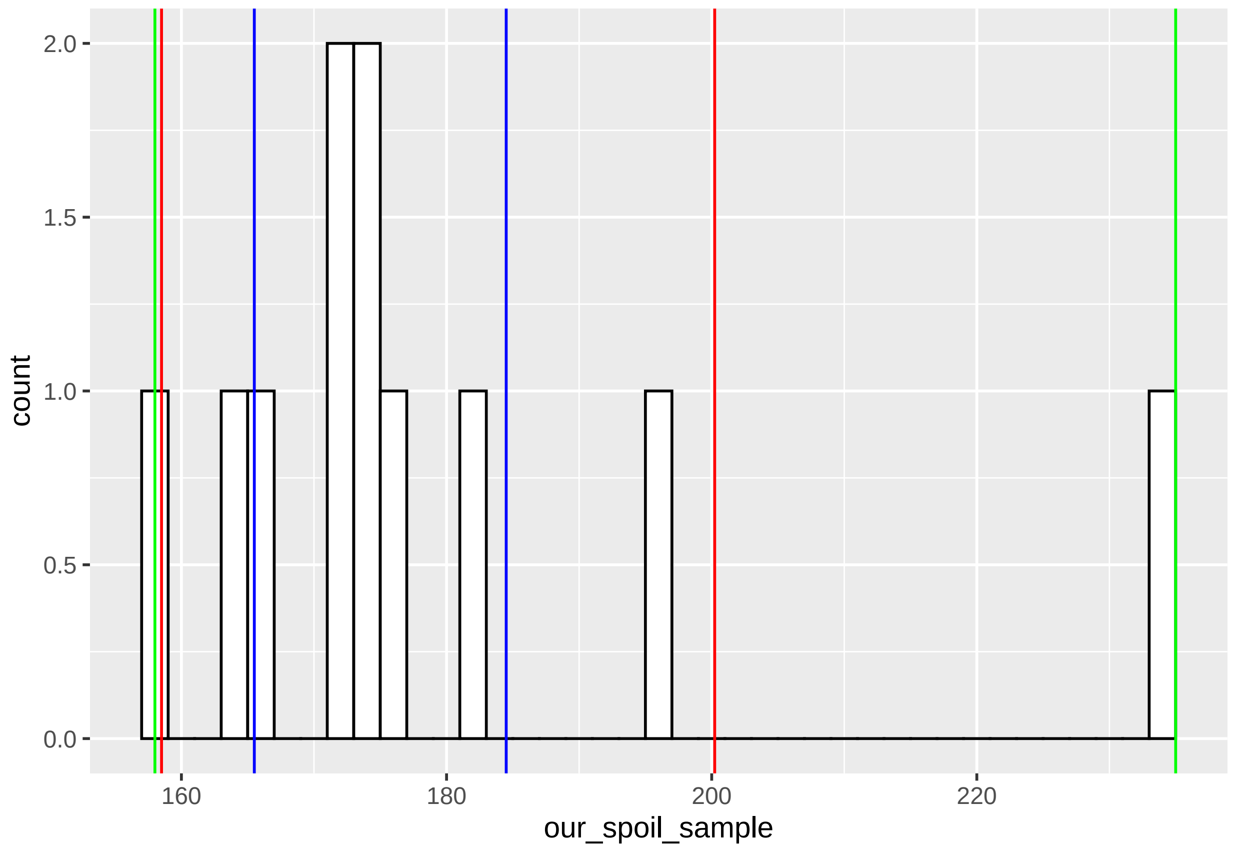
```
sd(our_spoil_sample)
```

```
## [1] 20.86275
```

```
my_sd(our_spoil_sample)
```

```
## [1] 20.86275
```

```
ggplot() +
  aes(our_spoil_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=c(median(our_spoil_sample) +
                              IQR(our_spoil_sample),median(our_spoil_sample) -
                              IQR(our_spoil_sample)), color="blue") +
  geom_vline(xintercept=c(mean(our_spoil_sample) +
                              sd(our_spoil_sample),mean(our_spoil_sample) -
                              sd(our_spoil_sample)), color="red") +
  geom_vline(xintercept=range(our_spoil_sample), color="green")
```



Check the properties for mean and sd for your sample

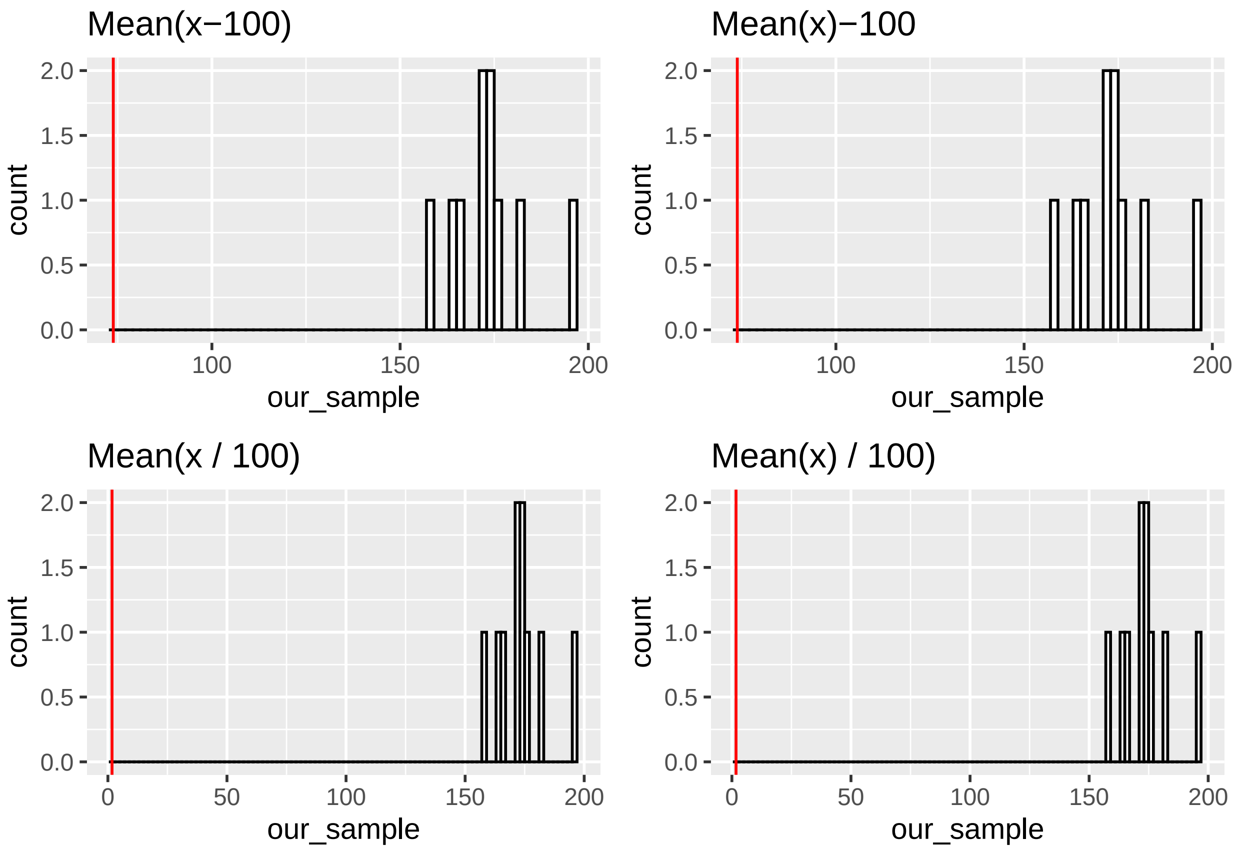
```
sum_table
```

```
##                  X        X-100        X/100
## mean 173.80000000  73.80000000   1.73800000
## var  105.28888889 105.28888889   0.01052889
## sd    10.26103742  10.26103742   0.10261037
```

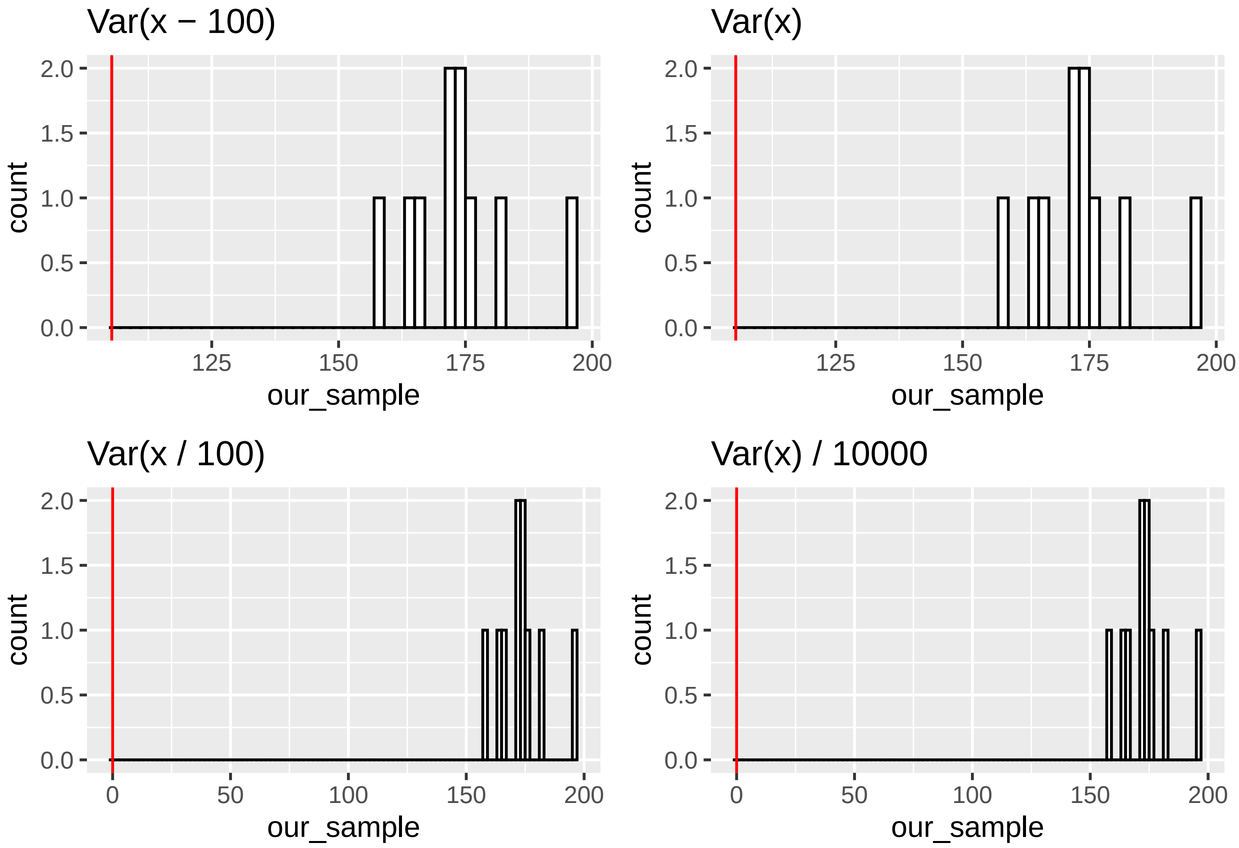
```
abs(sum(our_sample - mean(our_sample)) - 0) < 0.000000001
```

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

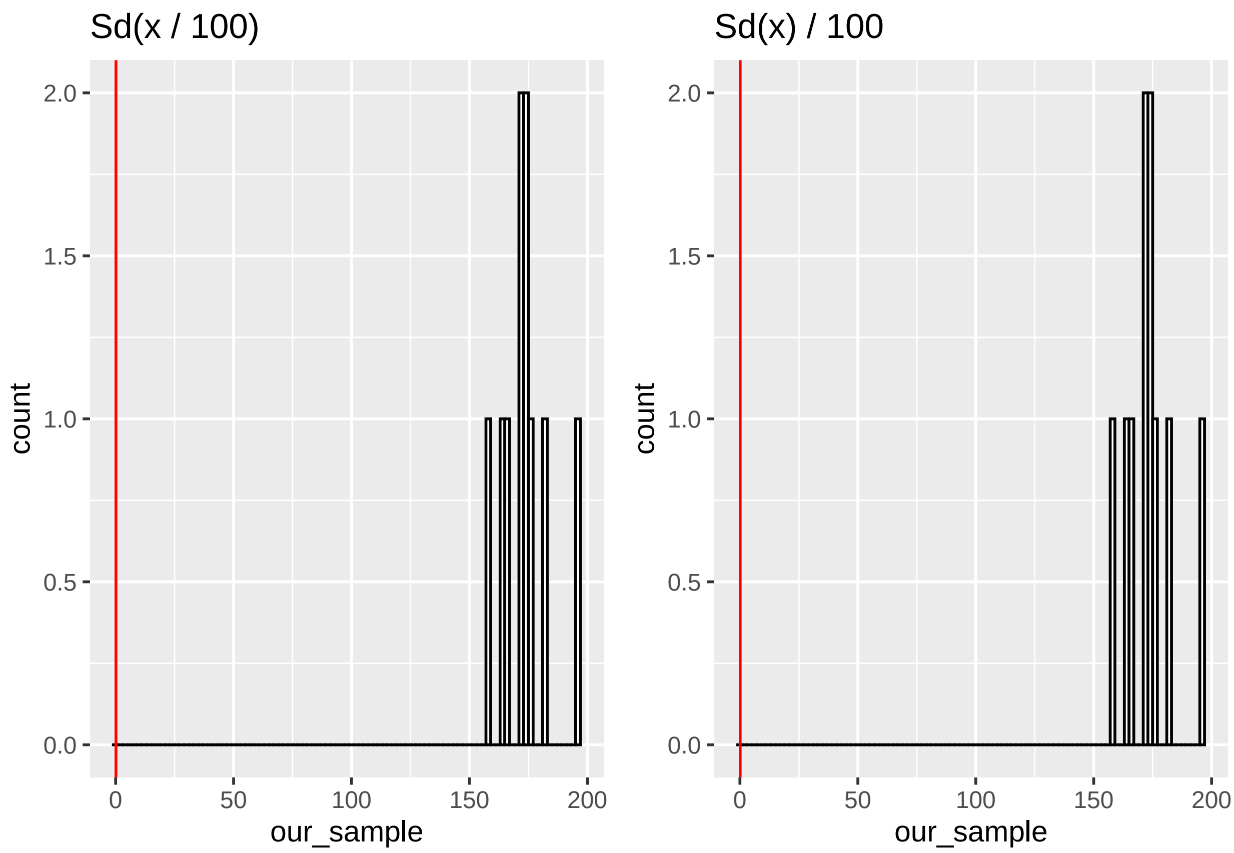
```
a <- ggplot() +
  aes(our_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=mean(our_sample-100), color="red") +
  ggtitle(label = 'Mean(x-100)')

b <- ggplot() +
  aes(our_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=(mean(our_sample) -100), color="red") +
  ggtitle(label = 'Mean(x)-100')

c <- ggplot() +
  aes(our_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=(mean(our_sample/100)), color="red") +
  ggtitle(label = 'Mean(x / 100)')
d <- ggplot() +
  aes(our_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=(mean(our_sample)/100), color="red") +
  ggtitle(label = 'Mean(x) / 100)')

ggarrange(a, b, c, d, ncol = 2, nrow = 2)
```



```
e <- ggplot() +
  aes(our_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=(var(our_sample - 100)), color="red") +
  ggtitle(label = 'Var(x - 100)')
f <- ggplot() +
  aes(our_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=(var(our_sample)), color="red") +
  ggtitle(label = 'Var(x)')
g <- ggplot() +
  aes(our_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=(var(our_sample/100)), color="red") +
  ggtitle(label = 'Var(x / 100)')
k <- ggplot() +
  aes(our_sample) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(xintercept=(var(our_sample/10000)), color="red") +
  ggtitle(label = 'Var(x) / 10000')
ggarrange(e, f, g, k, ncol = 2, nrow = 2)
```



```
ggarrange(l,m, ncol = 2, nrow = 1)
```



Normal distribution

For the population N(175, 10) find the probability to be less than 156cm

```
pnorm(156, 175, 10)
```

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

For the population N(175, 10) find the probability to be more than 198 cm

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

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

For the population N(175, 10) find the probability to be between 168 and 172 cm

```
pnorm(172, 175, 10) - pnorm(168, 175, 10)
```

```
## [1] 0.1401249
```

Standard normal distribution

```
pnorm(1) - pnorm(-1) # 68% of the data is within 1 standard deviation
```

```
## [1] 0.6826895
```

```
pnorm(2) - pnorm(-2) # 95% of the data is within 2 standard deviations
```

```
## [1] 0.9544997
```

```
pnorm(3) - pnorm(-3) # 99.7% of the data is within 3 standard deviations
```

```
## [1] 0.9973002
```

Generate sample using rnorm() from N(175, 10), find mean and sd

```
set.seed(42)
norm_sample <- rnorm(1000,175,10)
mean(norm_sample)
```

```
## [1] 174.7418
```

```
sd(norm_sample)
```

```
## [1] 10.02521
```

Standardize

```
stand_norm_sample <- scale(norm_sample)
mean(norm_sample)
```

```
## [1] 174.7418
```

```
sd(norm_sample)
```

```
## [1] 10.02521
```

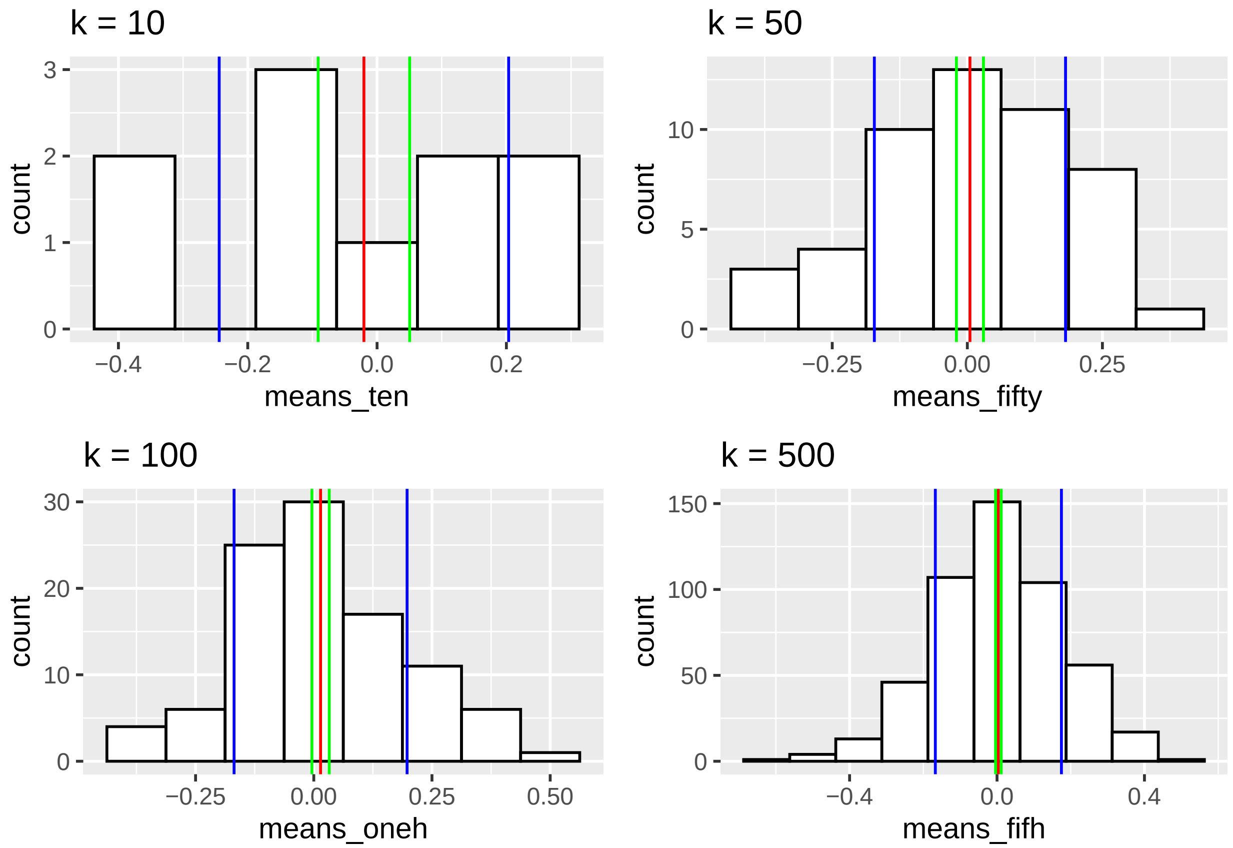
Central Limit Theorem

```
set.seed(42)
large_population <- rnorm(1000000)
my_samples_ten <- replicate(10, sample(large_population, 30)) #k  = 10
means_ten <- colMeans(my_samples_ten)
my_samples_fifty <- replicate(50, sample(large_population, 30)) #k = 50
means_fifty <- colMeans(my_samples_fifty)
my_samples_oneh <- replicate(100, sample(large_population, 30)) #k = 100
means_oneh <- colMeans(my_samples_oneh)
my_samples_fifh <- replicate(500, sample(large_population, 30)) #k = 500
means_fifh <- colMeans(my_samples_fifh)

se <- function(x) sqrt(var(x)/length(x))
means_table <- matrix(c(mean(means_ten), sd(means_ten), se(means_ten),
                      mean(means_fifty), sd(means_fifty), se(means_fifty),
                      mean(means_oneh), sd(means_oneh), se(means_oneh),
                      mean(means_fifh), sd(means_fifh), se(means_fifh)), ncol = 3, byrow = TRUE)

colnames(means_table) <- c("mean","sd","SE")
rownames(means_table) <- c("10","50","100", "500")
means_table <- as.table(means_table)
```

```
means_table
```

```
##             mean           sd           SE
## 10  -0.020343779  0.223801085  0.070772117
## 50   0.004837480  0.176904602  0.025018089
## 100  0.014391655  0.182947489  0.018294749
## 500  0.003676981  0.171083842  0.007651102
```

```
n <- ggplot() +
  aes(means_ten) +
  geom_histogram(binwidth=0.125, colour="black", fill="white") +
  geom_vline(xintercept=mean(means_ten), color="red") +
  geom_vline(xintercept=c(mean(means_ten) + sd(means_ten),
                          mean(means_ten) - sd(means_ten)), color="blue") +
  geom_vline(xintercept=c(mean(means_ten) + se(means_ten),
                          mean(means_ten) - se(means_ten)), color="green") +
  ggtitle(label = 'k = 10')

o <- ggplot() +
  aes(means_fifty) +
  geom_histogram(binwidth=0.125, colour="black", fill="white") +
  geom_vline(xintercept=mean(means_fifty), color="red") +
  geom_vline(xintercept=c(mean(means_fifty) + sd(means_fifty),
                          mean(means_fifty) - sd(means_fifty)), color="blue") +
  geom_vline(xintercept=c(mean(means_fifty) + se(means_fifty),
                          mean(means_fifty) - se(means_fifty)), color="green") +
  ggtitle(label = 'k = 50')

p <- ggplot() +
  aes(means_oneh) +
  geom_histogram(binwidth=0.125, colour="black", fill="white") +
  geom_vline(xintercept=mean(means_oneh), color="red") +
  geom_vline(xintercept=c(mean(means_oneh) + sd(means_oneh),
                          mean(means_oneh) - sd(means_oneh)), color="blue") +
  geom_vline(xintercept=c(mean(means_oneh) + se(means_oneh),
                          mean(means_oneh) - se(means_oneh)), color="green") +
  ggtitle(label = 'k = 100')

k <- ggplot() +
  aes(means_fifh) +
  geom_histogram(binwidth=0.125, colour="black", fill="white") +
  geom_vline(xintercept=mean(means_fifh), color="red") +
  geom_vline(xintercept=c(mean(means_fifh) + sd(means_fifh),
                          mean(means_fifh) - sd(means_fifh)), color="blue") +
  geom_vline(xintercept=c(mean(means_fifh) + se(means_fifh),
                          mean(means_fifh) - se(means_fifh)), color="green") +
  ggtitle(label = 'k = 500')

ggarrange(n, o, p, k, ncol = 2, nrow = 2)
```


k = 10
count
means_ten
k = 50
count
means_fifty
k = 100
count
means_oneh
k = 500
count
means_fifh