Statistics in R

Lisa

3/10/2020

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

1. Measures of center

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

```r
mean_fun <- function(x){
  return(sum(x)/length(x))
}

med_fun <- function(x){
  return(ifelse(length(x)%%2==0, (sort(x)[length(x)/2]+sort(x)[(length(x)/2)+1])/2,
              sort(x)[length(x)/2]))
}

mode_fun = function(x){
    return(as.numeric(names(table(x)[table(x)==max(table(x))])))
}
```

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

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

# mean
mean_fun(x)
```

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

```r
mean(x)
```

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

```r
# mean with trimming
mean(x, trim = 0.1)
```

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

```r
mean(sort(x)[-c(1,10)])
```

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

```r
# median
median(x)
```

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

```r
med_fun(x)
```

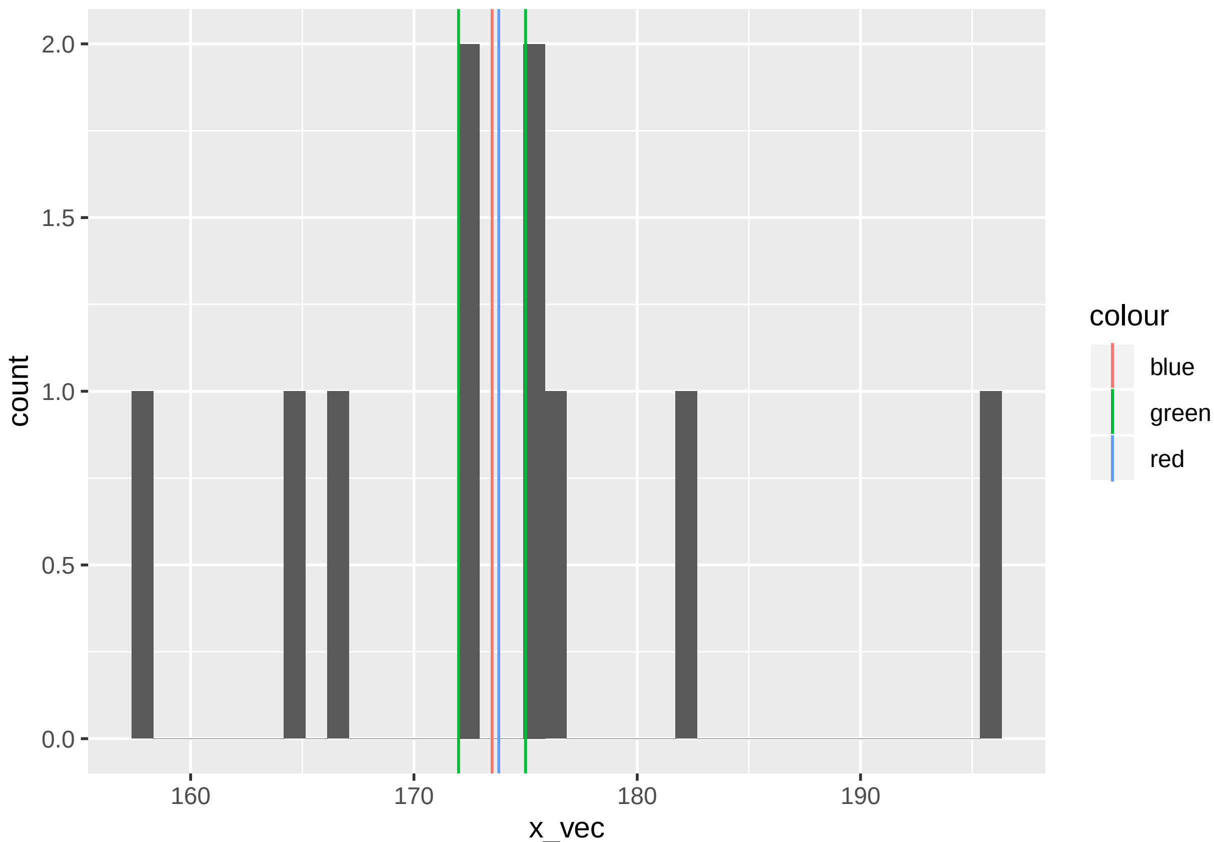
```
## [1] 173.5
```

```r
# mode
mode_fun(x)
```

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

1.2 visualize histogram with 3 vertical lines for measures of center

```r
library(ggplot2)

x_vec <- x

ggplot()+
  geom_histogram(aes(x_vec), bins=40)+
  geom_vline(aes(xintercept=mean(x_vec), col='red'))+
  geom_vline(aes(xintercept=median(x_vec), col='blue'))+
  geom_vline(aes(xintercept=mode_fun(x_vec), col='green'))
```



1.3 spoil your sample with the outlier - repeat steps 1.1 and 1.2

```r
x_vec_bias <- c(x_vec,210)

# mean
mean_fun(x_vec_bias)
```

```
## [1] 177.0909
```

```r
mean(x_vec_bias)
```

```
## [1] 177.0909
```

```r
# mean with trimming
mean(x_vec_bias, trim = 0.1)
```

```
## [1] 175.5556
```

```r
mean(sort(x_vec_bias)[-c(1,10)])
```

```
## [1] 177.1111
```

```r
# median
median(x_vec_bias)
```

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

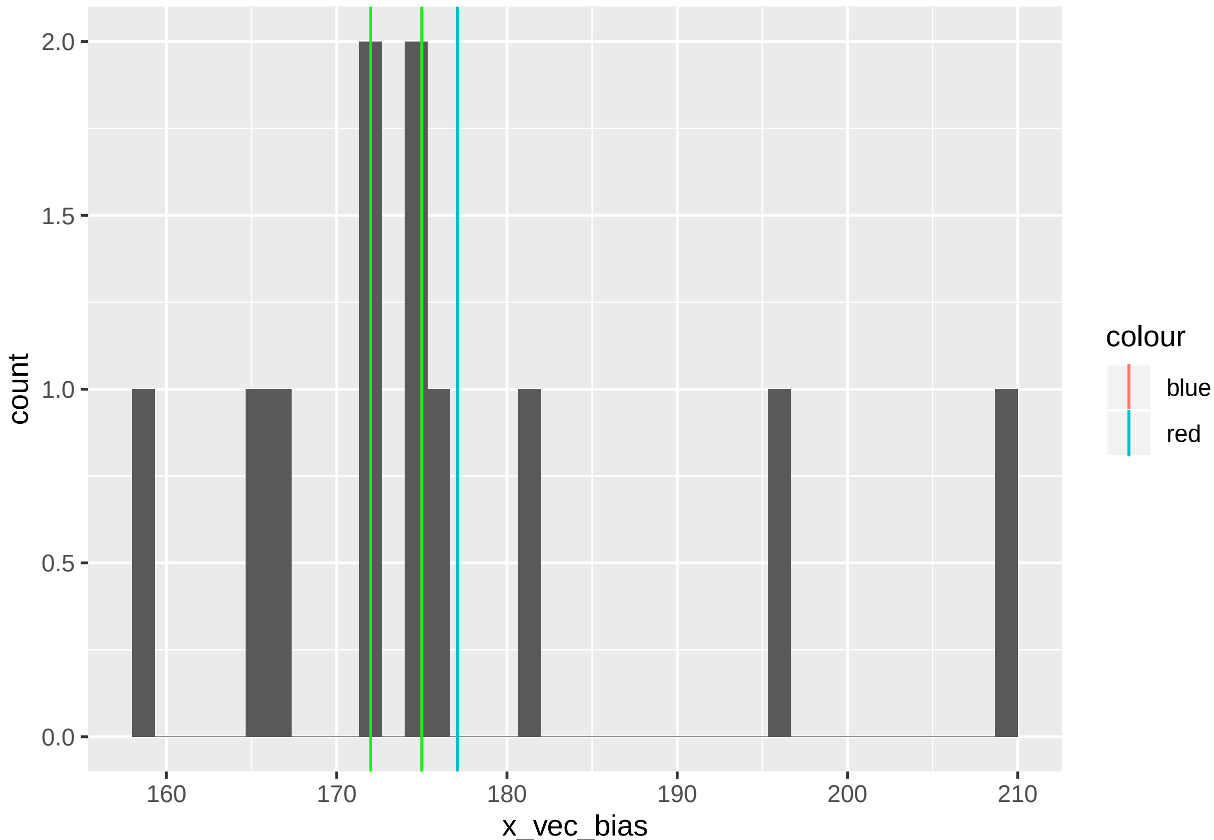
```r
med_fun(x_vec_bias)
```

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

```r
# mode
mode_fun(x_vec_bias)
```

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

```r
# plot
ggplot()+
  geom_histogram(aes(x_vec_bias), bins=40)+
  geom_vline(aes(xintercept=mean(x_vec_bias), col='red'))+
  geom_vline(aes(xintercept=median(x_vec_bias), col='blue'))+
  geom_vline(aes(xintercept=mode_fun(x_vec_bias)), col='green')
```



2. Measures of spread

2.0 write the functions/one-liners for variance and sd, calculate result, compare with the built-ins

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

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


variance(x)
```

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

```
var(x)
```

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

```
std(x)
```

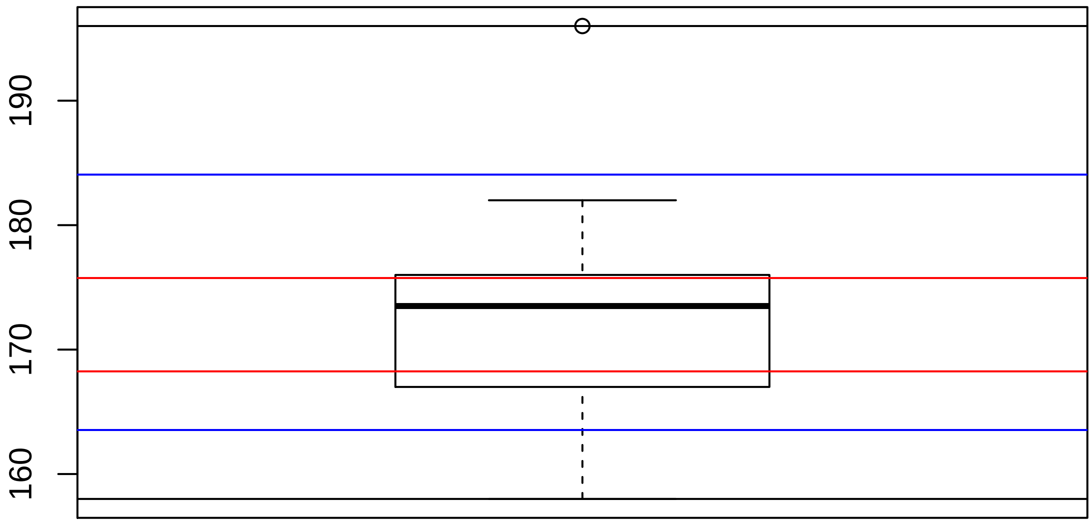
```
## [1] 10.26104
```

```
sd(x)
```

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

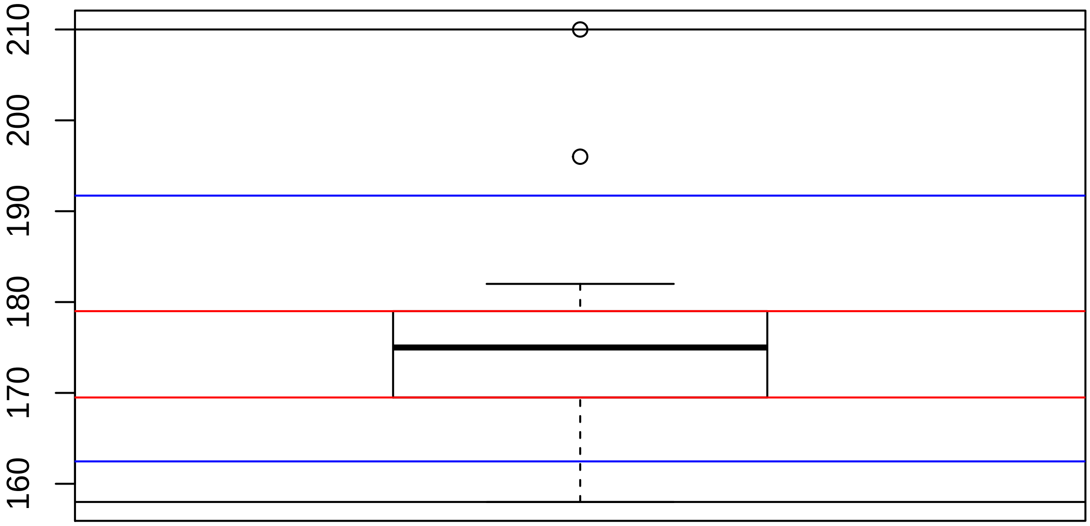
2.1 visualize with the box plot and add horizontal lines for range, IQR, 1-sd borders (use built-ins)

```r
boxplot(x)
# range
abline(h=range(x)[1])
abline(h=range(x)[2])
# IQR - red line
abline(h=quantile(x, c(0.25, 0.75)), col="red")
# 1-sd borders - blue line
abline(h=(mean(x)-sd(x)), col='blue')
abline(h = (mean(x)+sd(x)), col='blue')
```



2.2 spoil your sample with the outlier, repeat step 2.1

```r
boxplot(x_vec_bias)
# range
abline(h=range(x_vec_bias)[1])
abline(h=range(x_vec_bias)[2])
# IQR - red line
abline(h=quantile(x_vec_bias, c(0.25, 0.75)), col="red")
# 1-sd borders - blue line
abline(h=(mean(x_vec_bias)-sd(x_vec_bias)), col='blue')
abline(h = (mean(x_vec_bias)+sd(x_vec_bias)), col='blue')
```



3. Properties

3.0 check the properties for mean and sd for your sample

```
mean(x-100)
```

```
## [1] 73.8
```

```
mean(x) - 100
```

```
## [1] 73.8
```

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

```
## [1] 1.738
```

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

```
## [1] 1.738
```

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

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

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

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

```
var(x / 100)
```

```
## [1] 0.01052889
```

```
var(x) / 10000
```

```
## [1] 0.01052889
```

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

```
## [1] 0.1026104
```

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

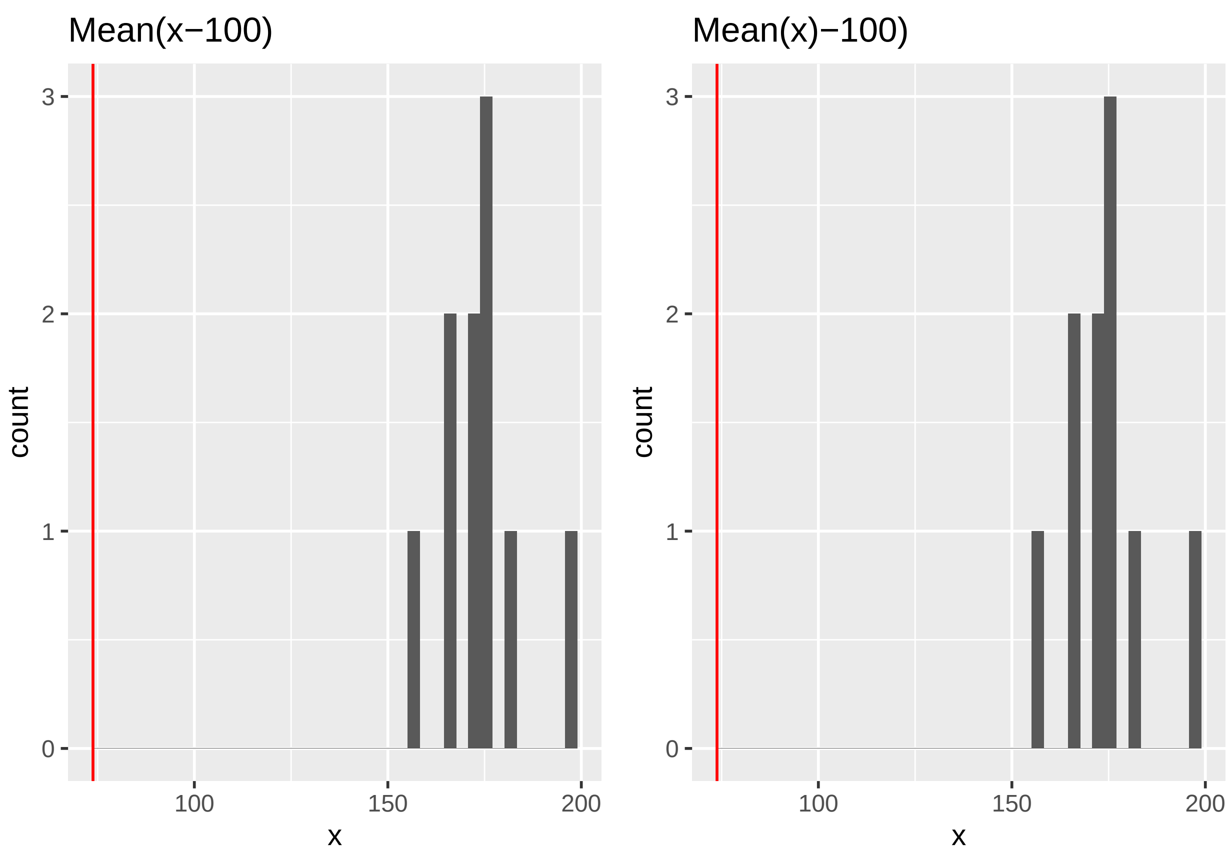
```
## [1] 0.1026104
```

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

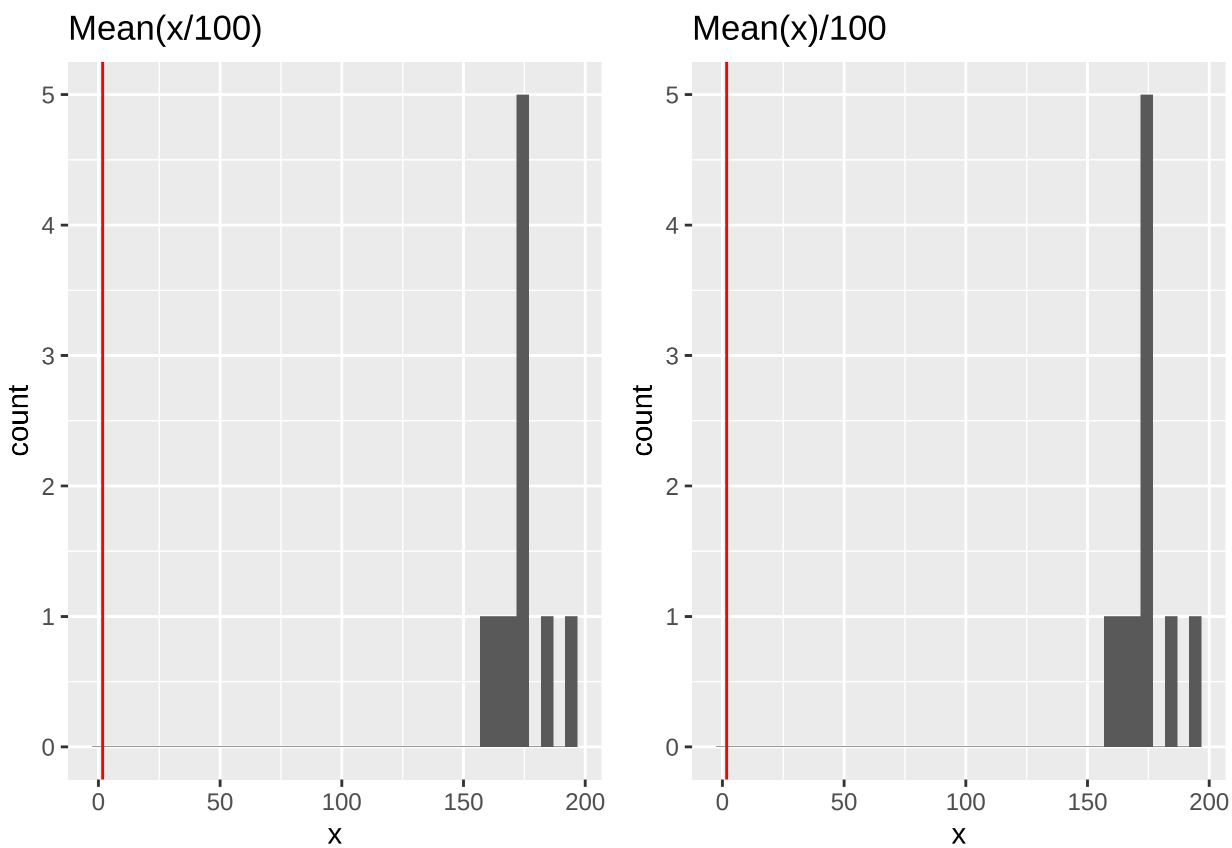
```r
names_ <- c('mean_extr', 'mean_div', 'var_extr', 'var_div', 'sd_div')
rules1 <-c((mean(x-100)), (mean(x/100)), (var(x-100)), (var(x/100)), (sd(x/100)))
rules2 <- c((mean(x) - 100), (mean(x) / 100),
            (var(x)), (var(x) / 10000), (sd(x) / 100))
rul <- data.frame(names_, rules1, rules2)
rul
```

```
##      names_       rules1       rules2
## 1 mean_extr  73.80000000  73.80000000
## 2  mean_div   1.73800000   1.73800000
## 3  var_extr 105.28888889 105.28888889
## 4   var_div   0.01052889   0.01052889
## 5    sd_div   0.10261037   0.10261037
```

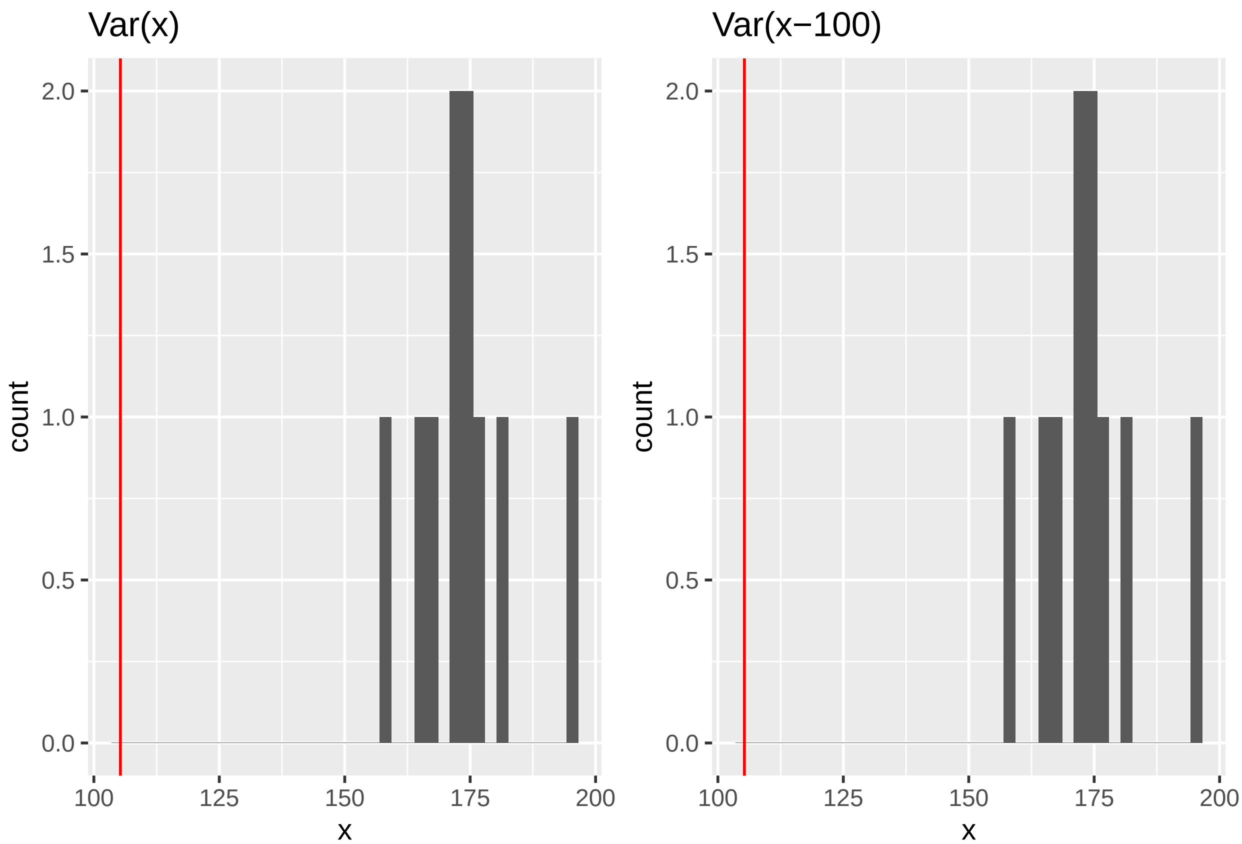
```r
# mean_extr
a <- ggplot()  +
  geom_histogram(aes(x), bins=40) +
  geom_vline(xintercept=mean(x-100), color="red") +
  ggtitle(label = 'Mean(x-100)')

b <- ggplot()  +
  geom_histogram(aes(x), bins=40) +
  geom_vline(xintercept=mean(x)-100, color="red") +
  ggtitle(label = 'Mean(x)-100)')

ggarrange(a, b, ncol = 2, nrow = 1)
```



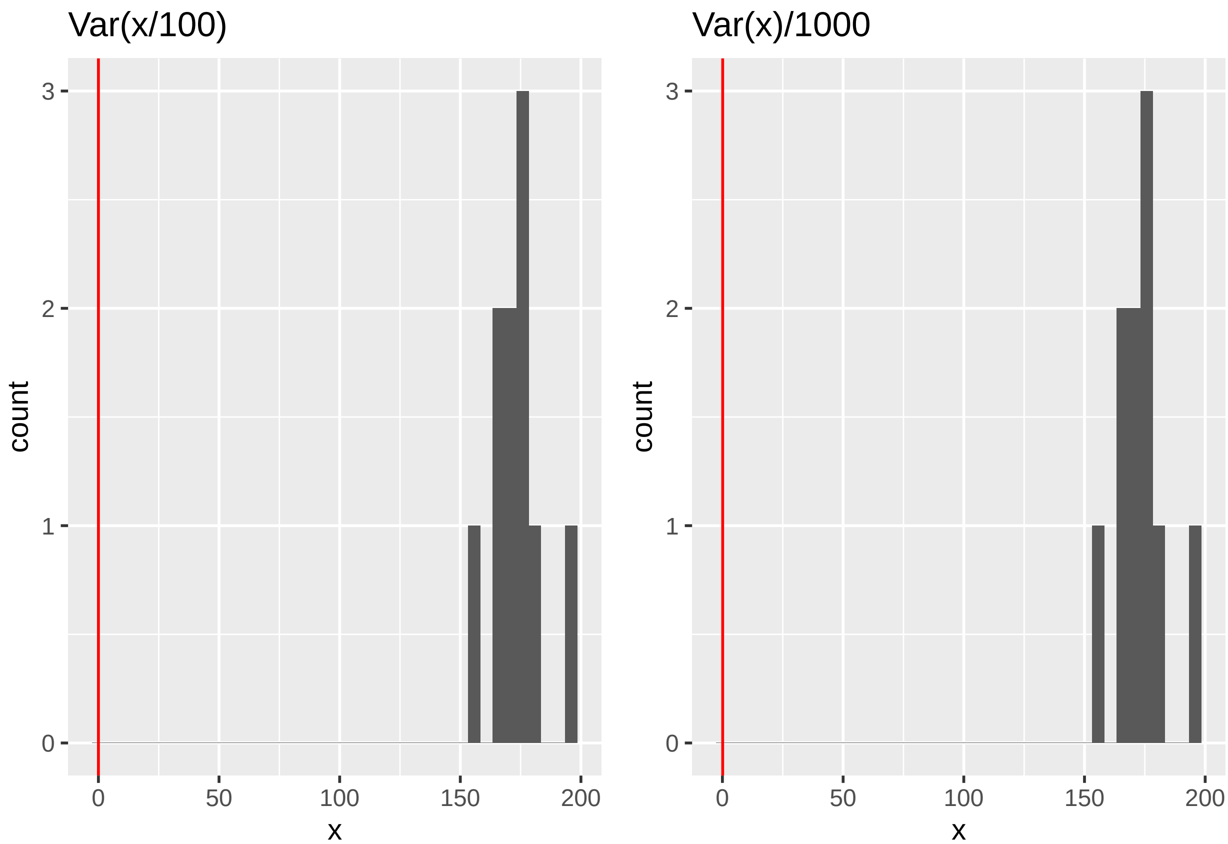
```r
# mean_div
c <- ggplot()  +
  geom_histogram(aes(x), bins=40) +
  geom_vline(xintercept=mean(x/100), color="red") +
  ggtitle(label = 'Mean(x/100)')

d <- ggplot()  +
  geom_histogram(aes(x), bins=40) +
  geom_vline(xintercept=mean(x)/100, color="red") +
  ggtitle(label = 'Mean(x)/100')

ggarrange(c, d, ncol = 2, nrow = 1)
```



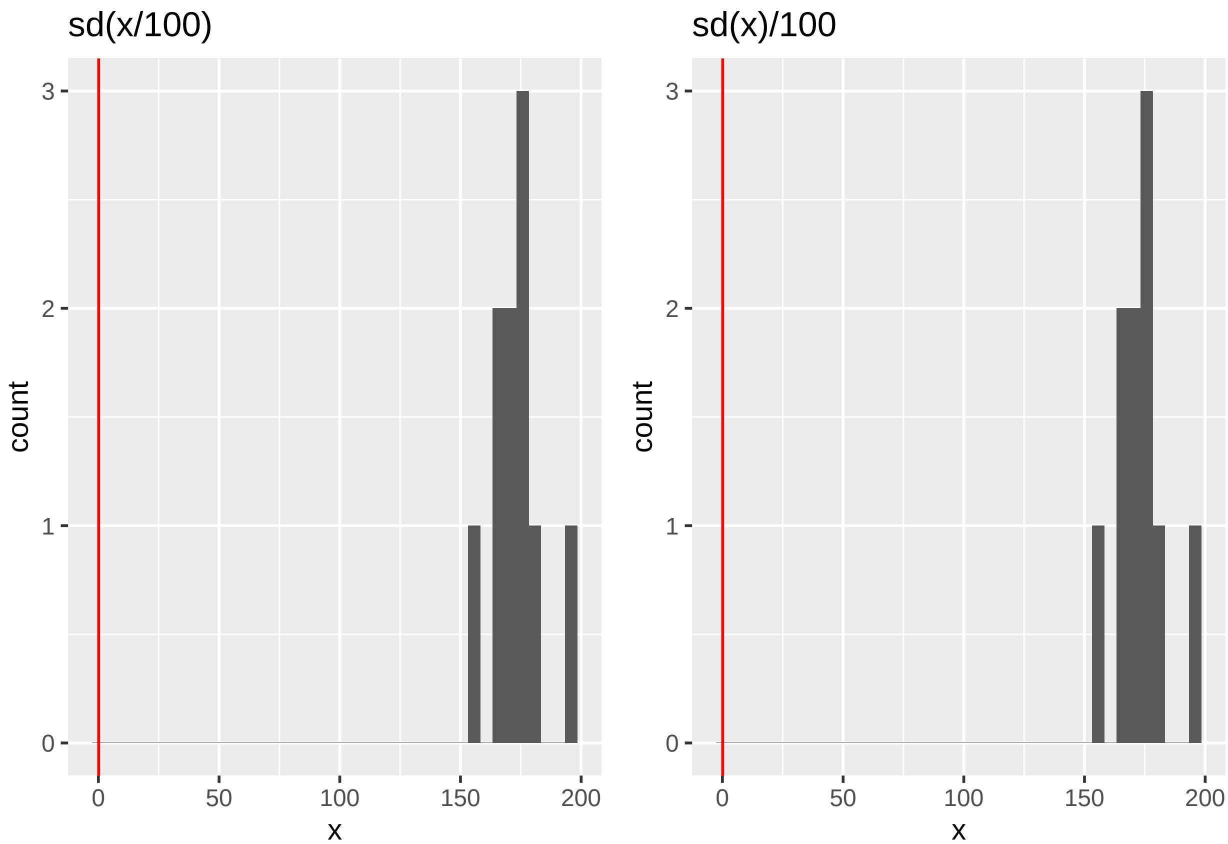

```r
# var_extr
e <- ggplot()  +
  geom_histogram(aes(x), bins=40) +
  geom_vline(xintercept=var(x), color="red") +
  ggtitle(label = 'Var(x)')

f <- ggplot()  +
  geom_histogram(aes(x), bins=40) +
  geom_vline(xintercept=var(x-100), color="red") +
  ggtitle(label = 'Var(x-100)')

ggarrange(e, f, ncol = 2, nrow = 1)
```




```r
# var_div
g <- ggplot()  +
  geom_histogram(aes(x), bins=40) +
  geom_vline(xintercept=var(x/100), color="red") +
  ggtitle(label = 'Var(x/100)')

h <- ggplot()  +
  geom_histogram(aes(x), bins=40) +
  geom_vline(xintercept=var(x)/1000, color="red") +
  ggtitle(label = 'Var(x)/1000')

ggarrange(g, h, ncol = 2, nrow = 1)
```




```r
# sd_div
m <- ggplot()  +
  geom_histogram(aes(x), bins=40) +
  geom_vline(xintercept=sd(x/100), color="red") +
  ggtitle(label = 'sd(x/100)')

n <- ggplot()  +
  geom_histogram(aes(x), bins=40) +
  geom_vline(xintercept=sd(x)/100, color="red") +
  ggtitle(label = 'sd(x)/100')

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



4. Normal distribution

4.0 for the population N(175, 10) find the probability to be:

```r
# less than 156cm,
pnorm(156, 175, 10)
```

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

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

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

```r
# between 168 and 172 cm
pnorm(172, 175, 10)-pnorm(168, 175, 10)
```

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

Standard normal distribution

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

```r
# 1-sd : ~68% results
pnorm(1)-pnorm(-1)
```

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

```
# 2-sd : ~95% results
pnorm(2)-pnorm(-2)
```

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

```
# 3-sd : ~99.7% results
pnorm(3)-pnorm(-3)
```

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

Standardization

set.seed() rnorm()

4.2 generate sample using rnorm() from N(175, 10), find mean ans sd;

```
sample <- rnorm(100, 175, 10)
mean(sample)
```

```
## [1] 172.9314
```

```
sd(sample)
```

```
## [1] 9.910642
```

4.3 standardize, find the same

```
sample_st <- (sample-mean(sample))/sd(sample)
# mean ~0
mean(sample_st)
```

```
## [1] -1.406322e-15
```

```
# sd ~1
sd(sample_st)
```

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

5. Central Limit Theorem

set.seed() rnorm() sample()

5.0 Generate large population (n ~ 100 000 - 1 000 000) distributed as N(0, 1) Sample from population k observations for 30 times - you will have set of 30 samples. For each sample calculate mean. For the set calculate means of means, sd of means, SE. Create table with k, mean of means, sd of means, SE. Visualize distribution of means with histogram and lines for mean of means and SE.

```
set.seed(42)
pop <- rnorm(1e6, 0, 1)
# 5.1 k = 10
s_10 <- replicate(30,sample(pop, 10))
mean(s_10)
```

```
## [1] 0.02522645
```

```
means_10 <- colMeans(s_10)

# 5.2 k = 50
```

```
s_50 <- replicate(30,sample(pop, 50))
mean(s_50)
```

```
## [1] 0.00549262
```

```
means_50 <- colMeans(s_50)
# 5.3 k = 100
s_100 <- replicate(30,sample(pop, 100))
mean(s_100)
```

```
## [1] 0.01995338
```

```
means_100 <- colMeans(s_100)
# 5.4 k = 500
s_500 <- replicate(30,sample(pop, 500))
mean(s_500)
```

```
## [1] 0.01044726
```

```
means_500 <- colMeans(s_500)

se <- function(x) sqrt(var(x)/length(x))

# table
means_table <- data.frame(c('mean', 'sd', 'se'),
                          c(mean(means_10), sd(means_10), se(means_10)),
                      c(mean(means_50), sd(means_50), se(means_50)),
                      c(mean(means_100), sd(means_100), se(means_100)),
                     c(mean(means_500), sd(means_500), se(means_500)))
names(means_table) <- c('names','10', '50', '100', '500')

means_table
```

```
##   names         10         50        100         500
## 1  mean 0.02522645 0.00549262 0.01995338 0.010447262
## 2    sd 0.30371696 0.17057208 0.11070980 0.050676353
## 3    se 0.05545088 0.03114206 0.02021275 0.009252194
```

```
# plots
plot_10 <- ggplot() +
  geom_histogram(aes(means_10), bins = 40) +
  geom_vline(xintercept=mean(means_10), color="red") +
  geom_vline(xintercept=c(mean(means_10) + se(means_10),
                          mean(means_10) - se(means_10)), color="blue") +
  ggtitle(label='10')

plot_50 <- ggplot() +
  geom_histogram(aes(means_50), bins = 40) +
  geom_vline(xintercept=mean(means_50), color="red") +
  geom_vline(xintercept=c(mean(means_50) + se(means_50),
                          mean(means_50) - se(means_50)), color="blue") +
  ggtitle(label='50')

plot_100 <- ggplot() +
  geom_histogram(aes(means_100), bins= 40) +
  geom_vline(xintercept=mean(means_100), color="red") +
```

```
  geom_vline(xintercept=c(mean(means_100) + se(means_100),
                          mean(means_100) - se(means_100)), color="blue") +
  ggtitle(label='100')

plot_500 <- ggplot() +
  geom_histogram(aes(means_500), bins = 40) +
  geom_vline(xintercept=mean(means_500), color="red") +
  geom_vline(xintercept=c(mean(means_500) + se(means_500),
                          mean(means_500) - se(means_500)), color="blue") +
  ggtitle(label='50')

# facet
ggarrange(plot_10, plot_50, plot_100, plot_500, ncol = 2, nrow = 2)
```