HW0

Maria Firuleva

24/3/2020

```
library(dplyr)
library(ggplot2)
library(ggpubr)
library(magrittr)
```

Measures of center

- 1.0 create own sample or use given vector and write mode, median, mean functions/one-liners
- 1.1 calculate mode, median and mean for the sample. Compare results for own and built-ins for median and mean
- 1.2 visualize histogram with 3 vertical lines for measures of center
- 1.3 spoil your sample with the outlier - repeat steps 1.1 and 1.2

```
get_mode <- function(x) {
  which(table(x) == max(table(x))) %>% names() %>% as.numeric()
}

get_median <- function(x) {
  ifelse(length(x) %% 2, sort(x)[length(x) / 2], sum(sort(x)[round(length(x) / 2, 0)], sort(x)[round(le
}

get_mean <- function(x, trim=0) {
  mean(sort(x)[(floor(length(x) * trim) + 1): (length(x) - floor(length(x) * trim))])
}

x <- c(175, 176, 182, 165, 167, 172, 175, 196, 158, 172)
x_with_outliers <- c(x, 100, 110, 210)
vectors <- list(x, x_with_outliers)
names(vectors) <- c("without_outliers", "with_outliers")
lapply(vectors, function(vec) median(vec))
```

```
## $without_outliers
## [1] 173.5
##
## $with_outliers
## [1] 172
```

```
lapply(vectors, function(vec) get_median(vec))
```

```
## $without_outliers
## [1] 173.5
##
## $with_outliers
## [1] 172
```

```
lapply(vectors, function(vec) get_mode(vec))
```

```
## $without_outliers
## [1] 172 175
##
```

```
## $with_outliers
## [1] 172 175
```

```
lapply(vectors, function(vec) mean(vec))
```

```
## $without_outliers
## [1] 173.8
##
## $with_outliers
## [1] 166
```

```
lapply(vectors, function(vec) get_mean(vec))
```

```
## $without_outliers
## [1] 173.8
##
## $with_outliers
## [1] 166
```

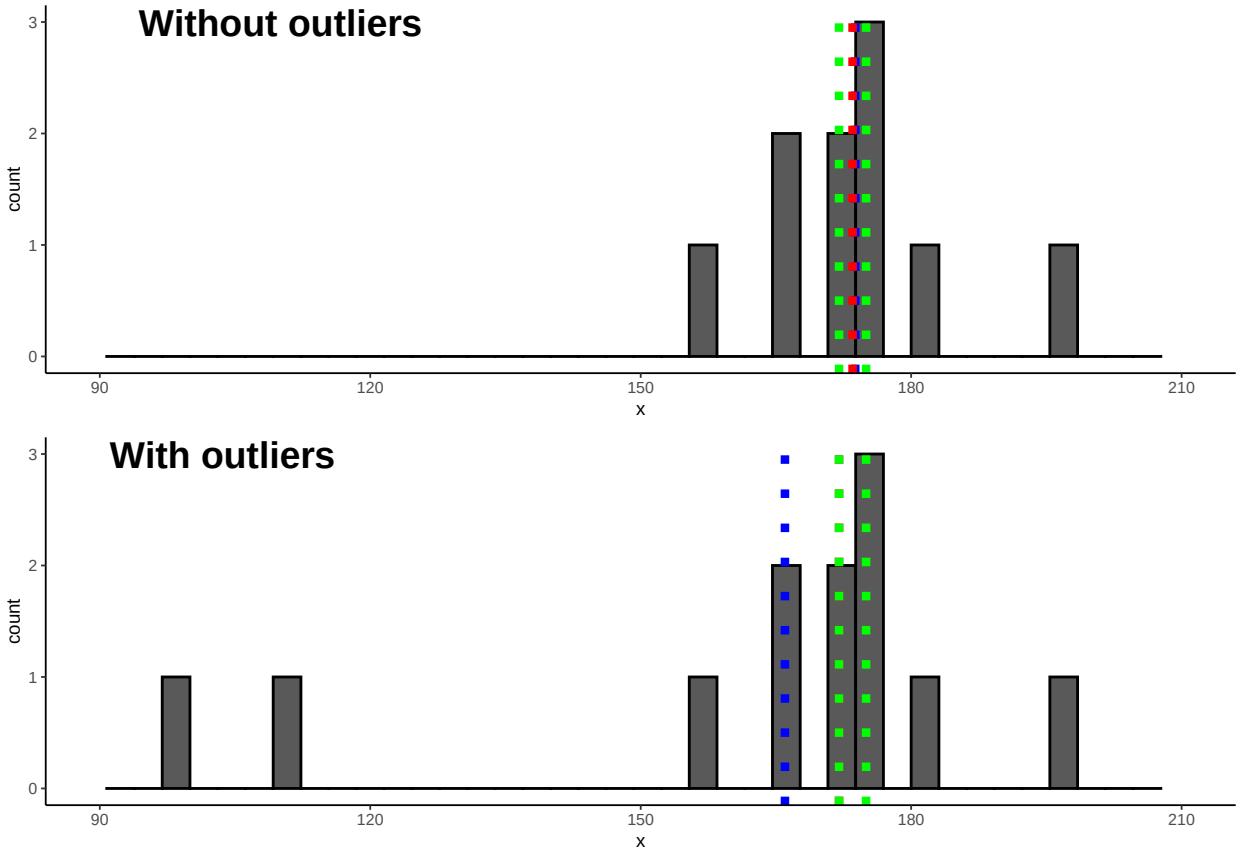
```
lapply(vectors, function(vec) mean(vec, trim = 0.1))
```

```
## $without_outliers
## [1] 173
##
## $with_outliers
## [1] 168
```

```
lapply(vectors, function(vec) get_mean(vec, trim = 0.1))
```

```
## $without_outliers
## [1] 173
##
## $with_outliers
## [1] 168
```

```
hist_centers <- function(x) {
  ggplot()+
    geom_histogram(aes(x = x), col='black', bins = 40)+
    theme_classic(base_size = 7)+
    scale_x_continuous(limits = c(90, 210))+
    geom_vline(xintercept = mean(x), linetype="dotted",
               color = "blue", size=1.5)+
    geom_vline(xintercept = median(x), linetype="dotted",
               color = "red", size=1.5)+
    geom_vline(xintercept = get_mode(x), linetype="dotted",
               color = "green", size=1.5)
}


p1 <- hist_centers(vectors$without_outliers)
p2 <- hist_centers(vectors$with_outliers)
ggarrange(p1, p2,
                    labels = c("Without outliers", "With outliers"),
                    ncol = 1, nrow = 2)
```



Measures of spread

- 2.0 write the functions/one-liners for variance and sd, calculate result, compare with the built-ins
- 2.1 visualize with the box plot and add horizontal lines for range, IQR, 1-sd borders (use built-ins)
- 2.2 spoil your sample with the outlier, repeat step 2.1

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

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

lapply(vectors, function(vec) var(vec))
```

```
## $without_outliers
## [1] 105.2889
##
## $with_outliers
## [1] 915.3333
```

```
lapply(vectors, function(vec) get_var(vec))
```

```
## $without_outliers
## [1] 105.2889
##
## $with_outliers
```

```
## [1] 915.3333
```

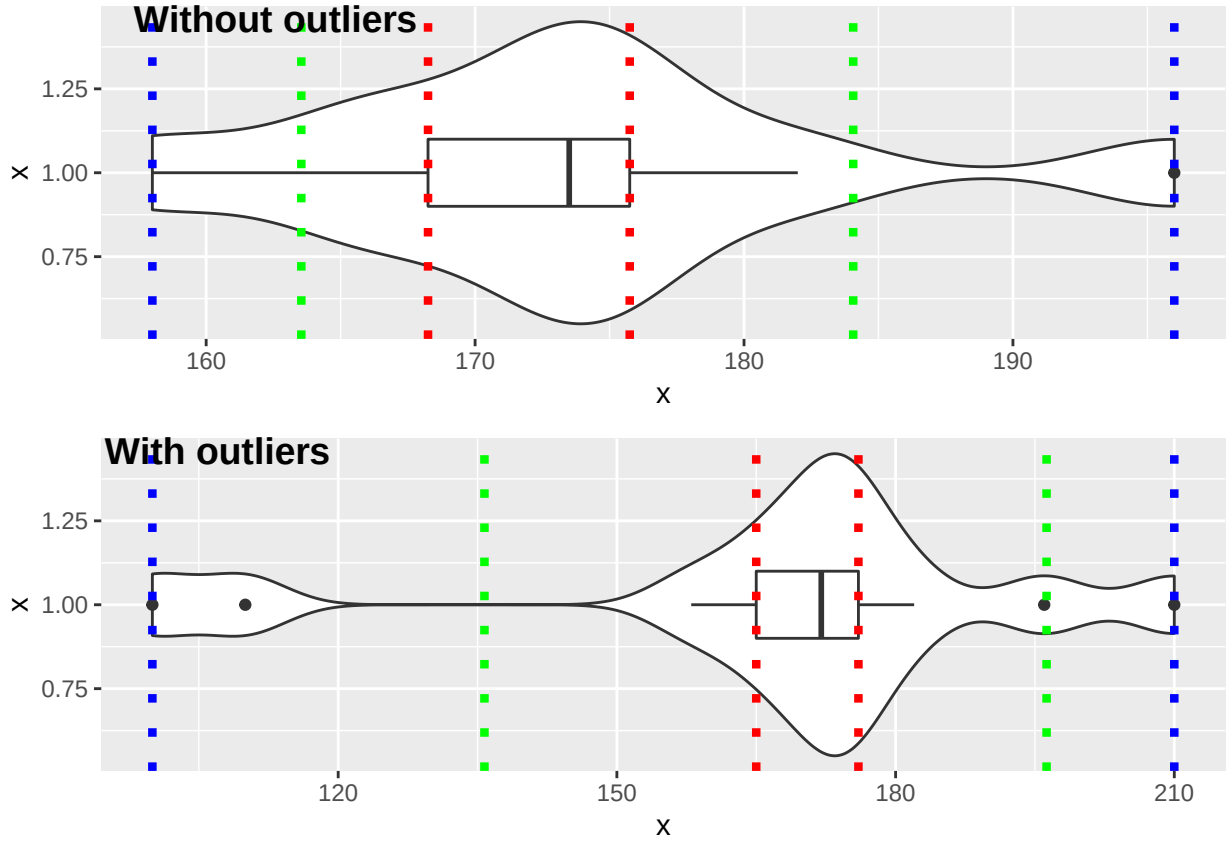
```
lapply(vectors, function(vec) sd(vec))
```

```
## $without_outliers
## [1] 10.26104
##
## $with_outliers
## [1] 30.25448
```

```
lapply(vectors, function(vec) get_sd(vec))
```

```
## $without_outliers
## [1] 10.26104
##
## $with_outliers
## [1] 30.25448
```

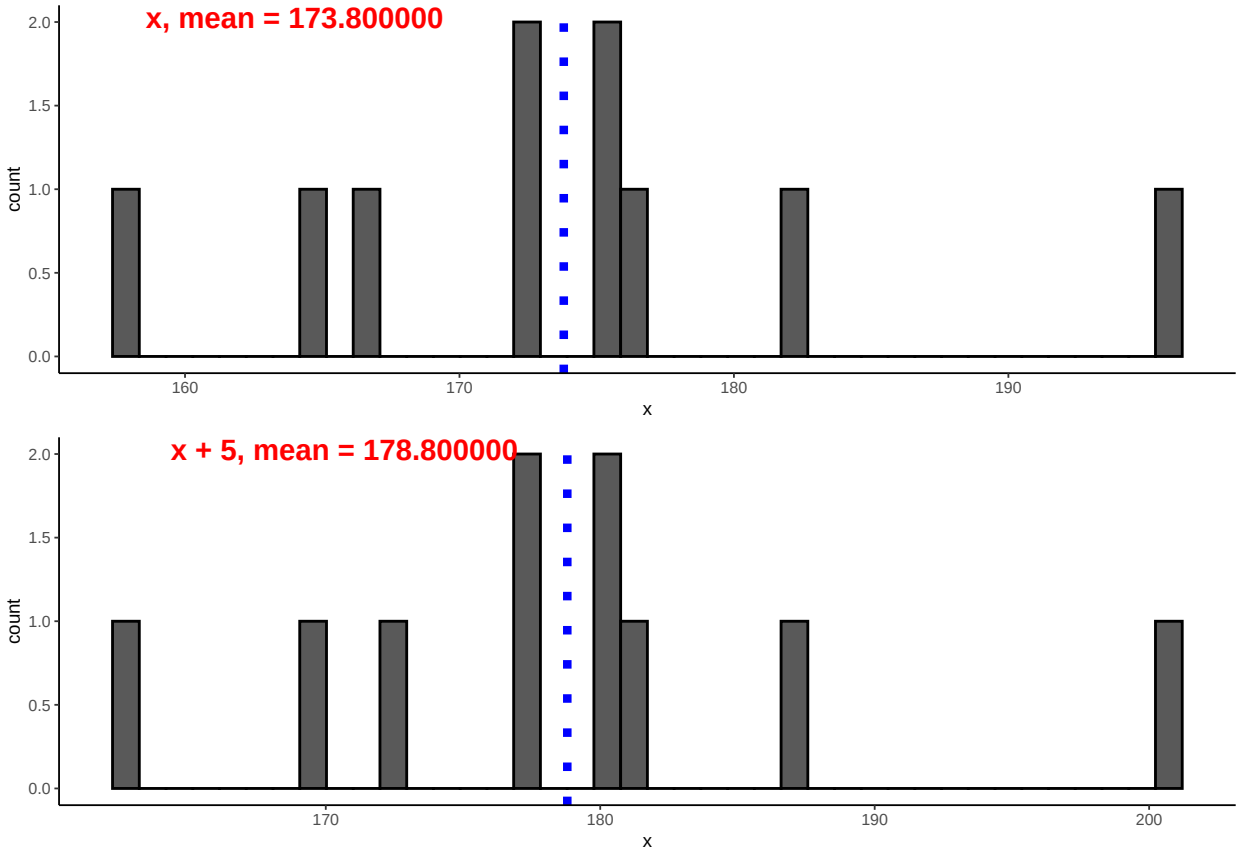
```
spread_plot <- function(x) {
  ggplot() + geom_violin(aes(y = x, x = 1))+geom_boxplot(aes(y = x, x = 1), width = 0.2)+
    geom_hline(yintercept = range(x), linetype="dotted",
               color = "blue", size=1.5)+
    geom_hline(yintercept = quantile(x)[c(2,4)], linetype="dotted",
               color = "red", size=1.5)+
    geom_hline(yintercept = c(mean(x) - sd(x), mean(x) + sd(x)), linetype="dotted",
               color = "green", size=1.5)+
    coord_flip()
}


p1 <- spread_plot(vectors$without_outliers)
p2 <- spread_plot(vectors$with_outliers)
ggarrange(p1, p2,
                    labels = c("Without outliers", "With outliers"),
                    ncol = 1, nrow = 2)
```



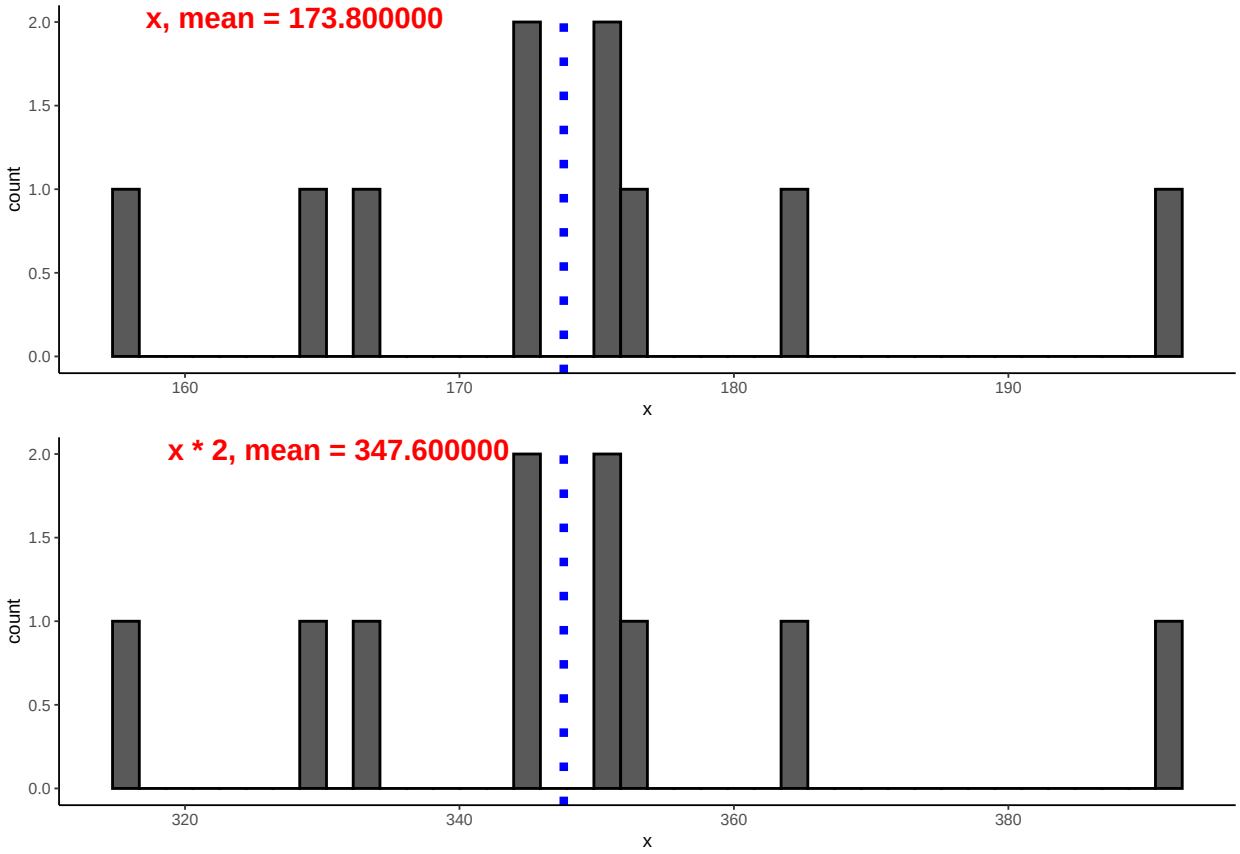
Properties

- 3.0 check the properties for mean and sd for your sample
- 3.1 visualize result tabularly and graphically (maybe with facetting free scales?)

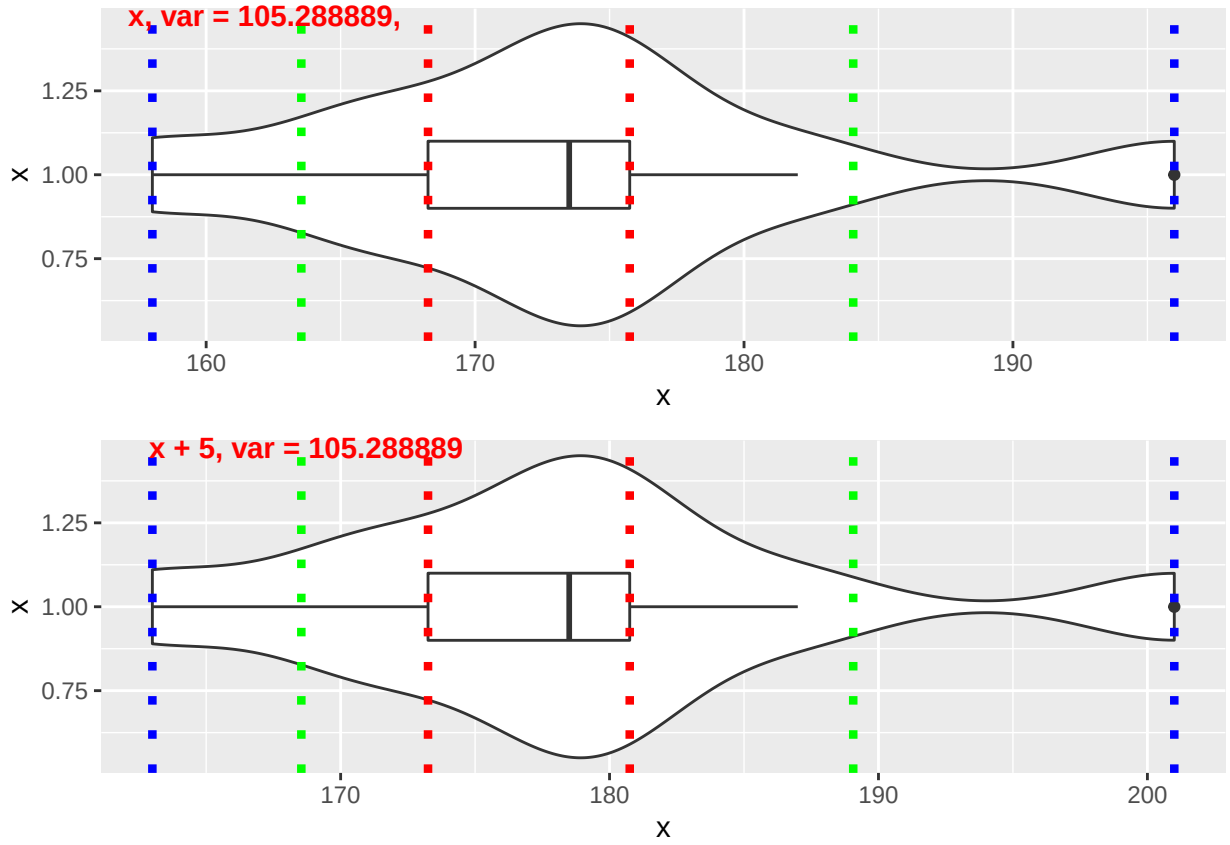
```
hist_mean_sum <- function(x) {
  ggplot() + geom_histogram(aes(x = x), col='black', bins = 40)+
    theme_classic(base_size = 7)+
    geom_vline(xintercept = mean(x), linetype="dotted",
               color = "blue", size=1.5)
}

p1 <- hist_mean_sum(vectors$without_outliers)
p2 <- hist_mean_sum(vectors$without_outliers + 5)
ggarrange(p1, p2,
                    labels = c(sprintf("x, mean = %f", mean(x)), sprintf("x + 5, mean = %f", mean(x + 5
                    ncol = 1, nrow = 2, font.label = list(size = 11, color = "red"))
```



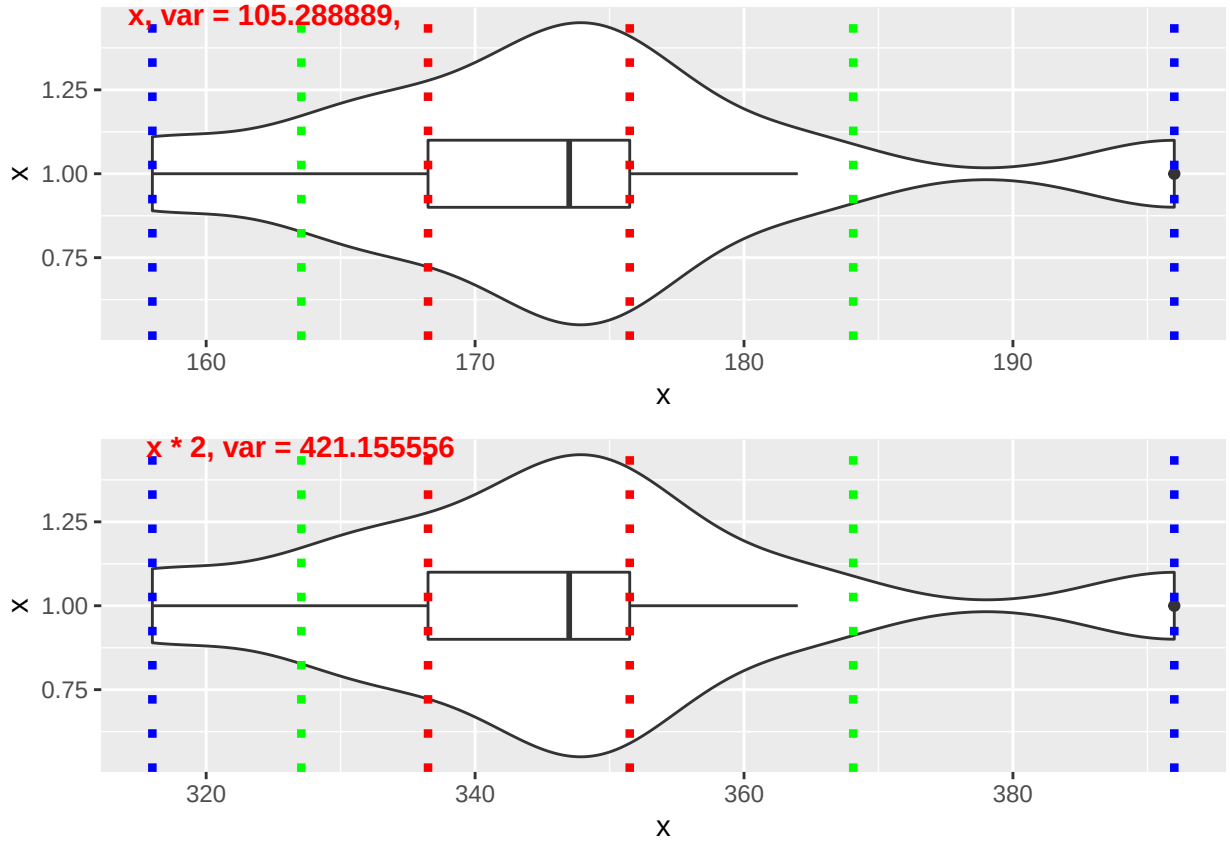

```
p1 <- hist_mean_sum(vectors$without_outliers)
p2 <- hist_mean_sum(vectors$without_outliers * 2)
ggarrange(p1, p2,
                    labels = c(sprintf("x, mean = %f", mean(x)), sprintf("x * 2, mean = %f", mean(x * 2
                    ncol = 1, nrow = 2)
```




```
p1 <- spread_plot(vectors$without_outliers)
p2 <- spread_plot(vectors$without_outliers + 5)
ggarrange(p1, p2,
                    labels = c(sprintf("x, var = %f,", var(x)), sprintf("x + 5, var = %f", var(x + 5))),
                    ncol = 1, nrow = 2)
```



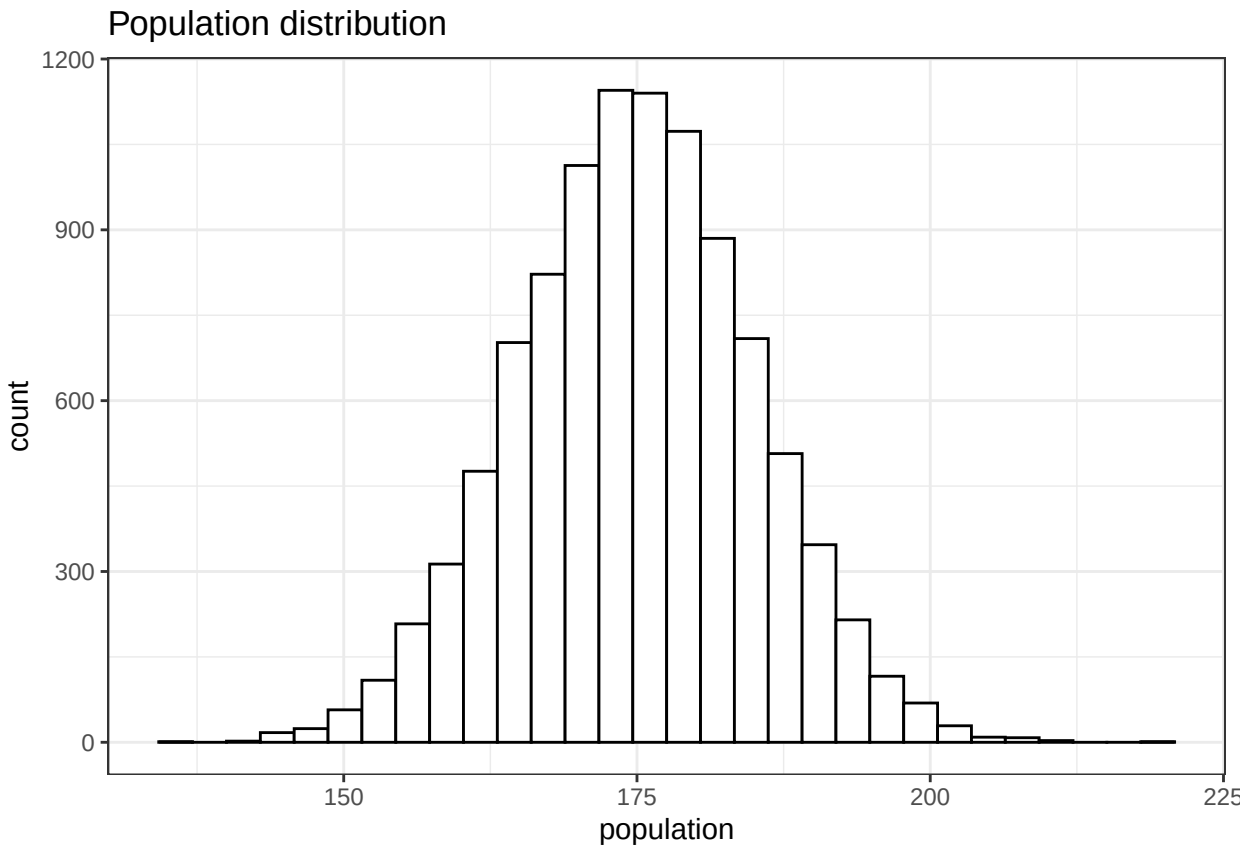

```
p1 <- spread_plot(vectors$without_outliers)
p2 <- spread_plot(vectors$without_outliers * 2)
ggarrange(p1, p2,
                    labels = c(sprintf("x, var = %f,", var(x)), sprintf("x * 2, var = %f", var(x * 2))),
                    ncol = 1, nrow = 2)
```



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

- less than 156cm,
- more than 198,
- between 168 and 172 cm

```
set.seed(42)
population <- rnorm(n = 1e4, mean = 175, sd = 10)
ggplot()+geom_histogram(aes(x = population), fill='white', col='black')+theme_bw()+
  ggtitle("Population distribution")
```




```
sprintf("P(X < 156) = %f", pnorm(156, mean = 175, sd = 10))
```

```
## [1] "P(X < 156) = 0.028717"
```

```
sprintf("P(X > 198) = %f", pnorm(198, mean = 175, sd = 10, lower.tail = F))
```

```
## [1] "P(X > 198) = 0.010724"
```

```
sprintf("P(168 > X > 172) = %f", pnorm(172, mean = 175, sd = 10) - pnorm(168, mean = 175, sd = 10))
```

```
## [1] "P(168 > X > 172) = 0.140125"
```

Standard normal distribution

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

```
set.seed(42)
pnorm(1, mean = 0, sd = 1) - pnorm(-1, mean = 0, sd = 1)
```

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

```
pnorm(2, mean = 0, sd = 1) - pnorm(-2, mean = 0, sd = 1)
```

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

```
pnorm(3, mean = 0, sd = 1) - pnorm(-3, mean = 0, sd = 1)
```

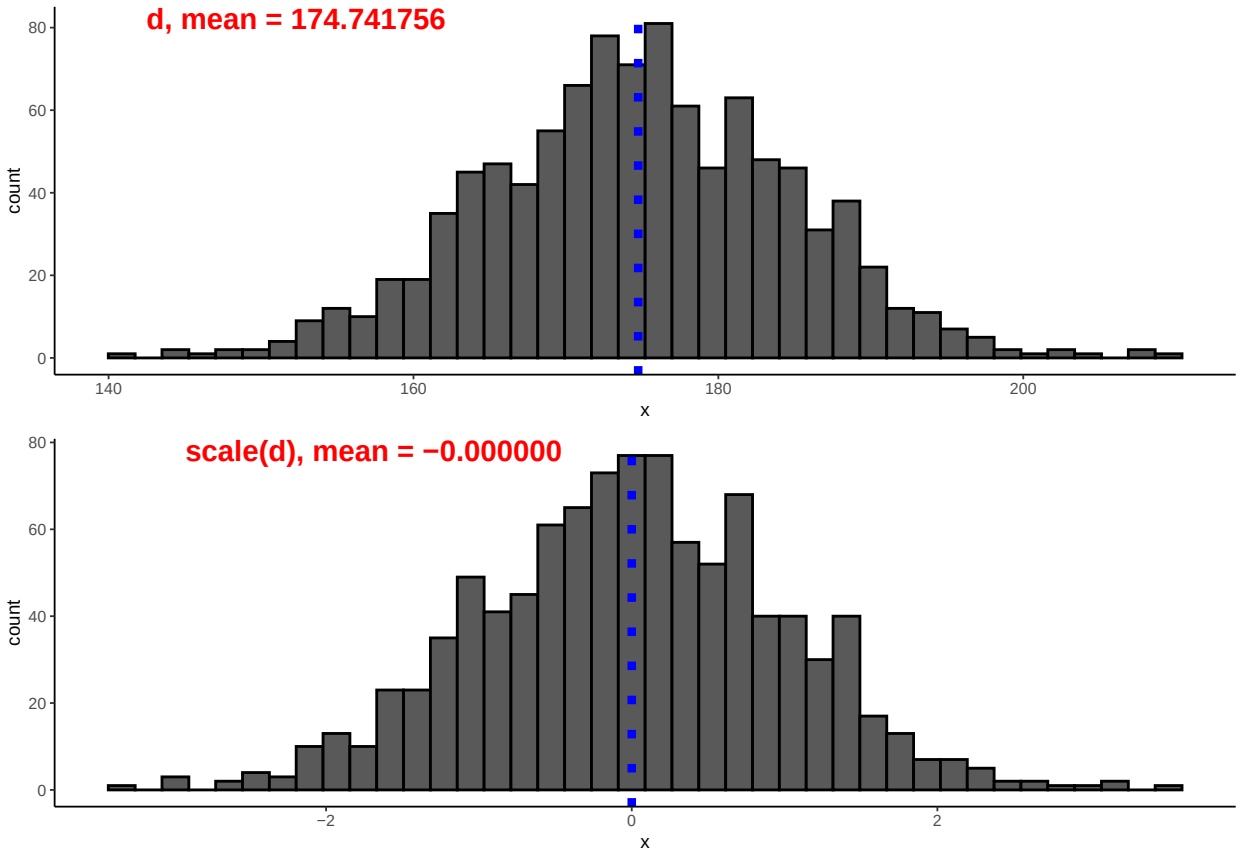
```
## [1] 0.9973002
```

Standardization

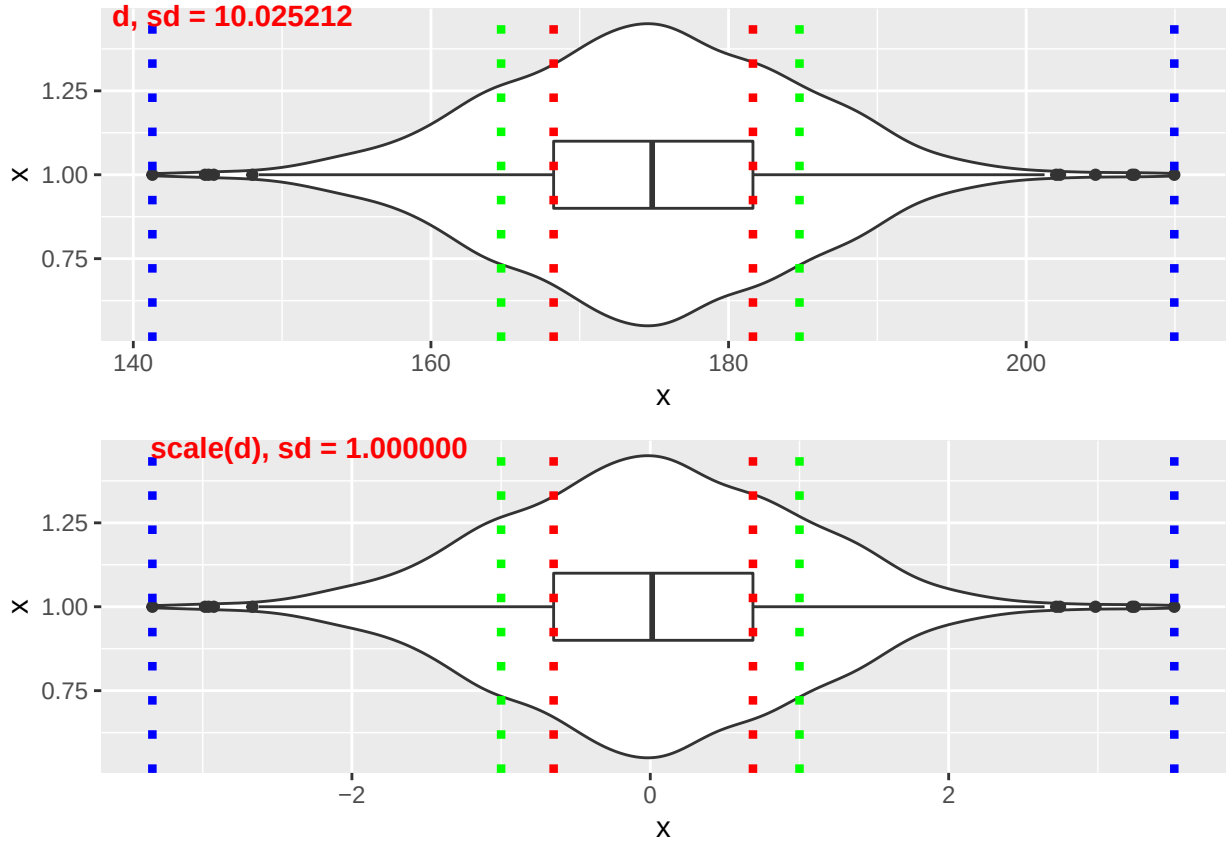
- 4.2 generate sample using rnorm() from N(175, 10), find mean ans sd;

- 4.3 standardize, find the same

```
set.seed(42)
d <- rnorm(1e3, mean = 175, sd = 10)
p1 <- hist_mean_sum(d)
p2 <- hist_mean_sum(scale(d))
ggarrange(p1, p2,
                    labels = c(sprintf("d, mean = %f", mean(d)), sprintf("scale(d), mean = %f", mean(sc
                    ncol = 1, nrow = 2, font.label = list(size = 11, color = "red"))
```



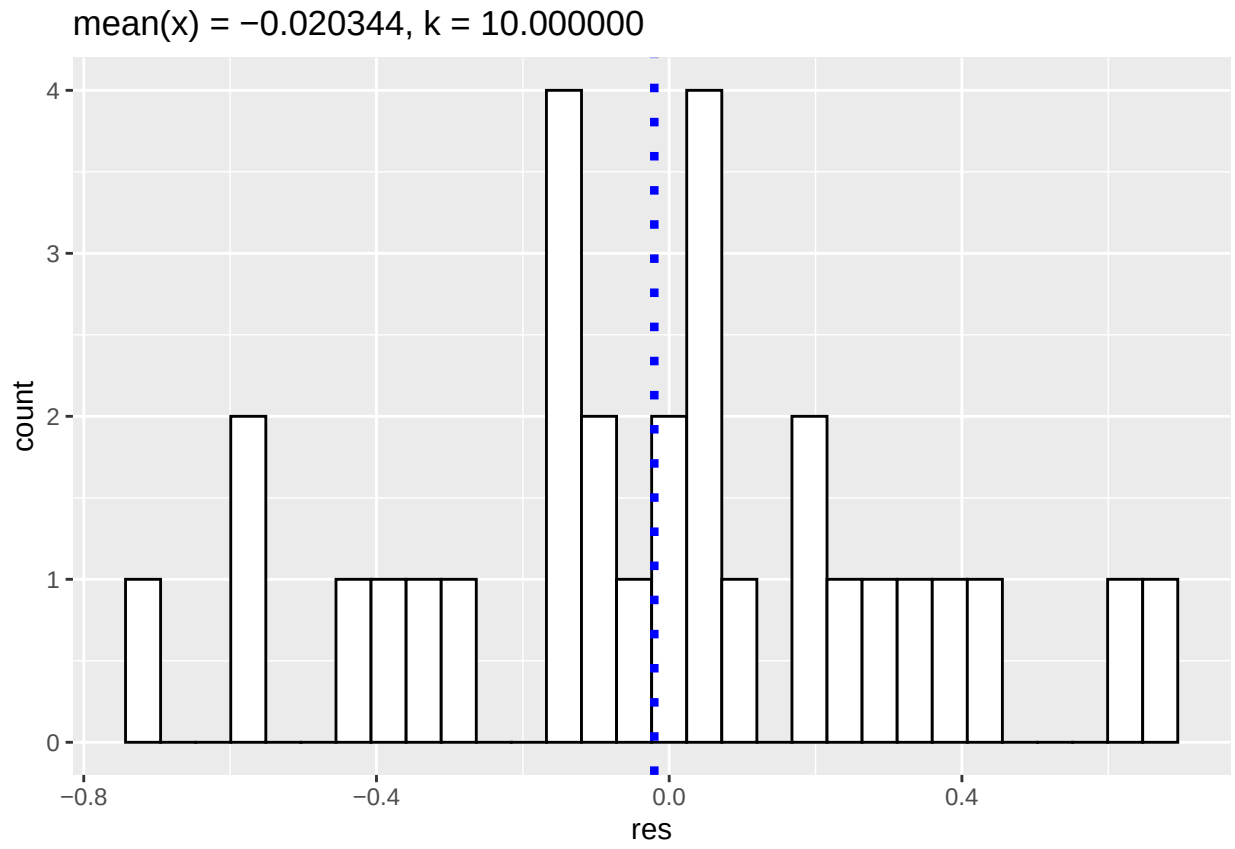

```
p1 <- spread_plot(d)
p2 <- spread_plot(scale(d))
ggarrange(p1, p2,
                    labels = c(sprintf("d, sd = %f", sd(d)), sprintf("scale(d), sd = %f", sd(scale(d)))
                    ncol = 1, nrow = 2, font.label = list(size = 11, color = "red"))
```



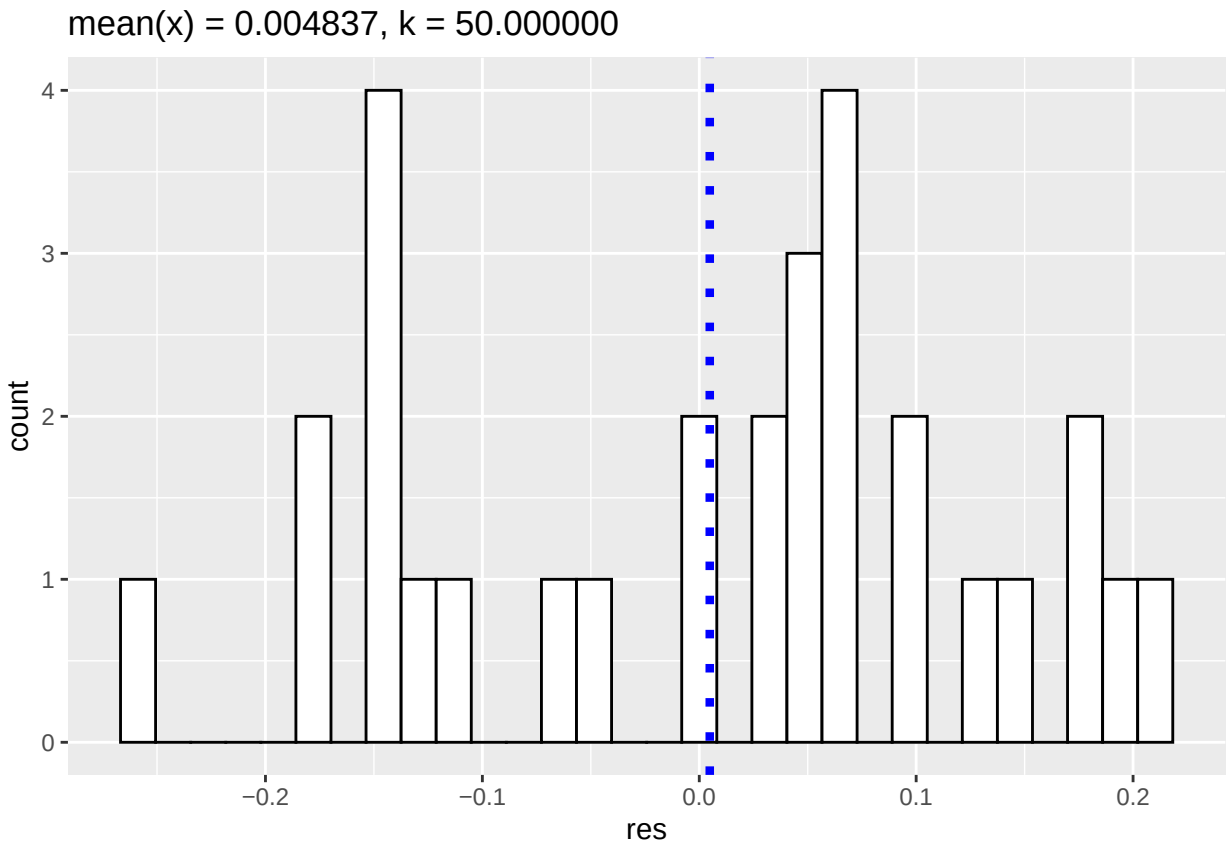
Central Limit Theorem

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. * 5.1 k = 10 * 5.2 k = 50 * 5.3 k = 100 * 5.4 k = 500 * 5.5 Compare results

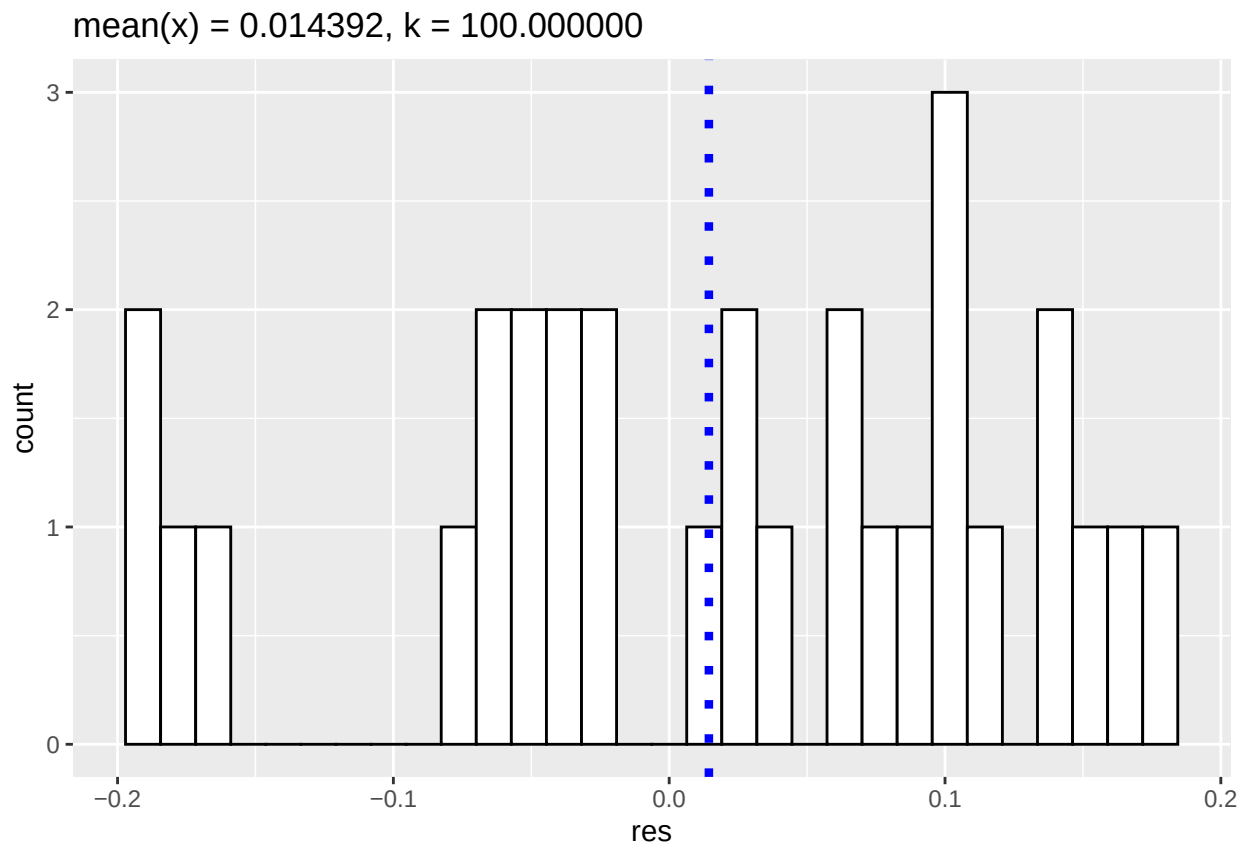
```
set.seed(42)

get_properties <- function(d, k) {
  res <- sapply(replicate(30, sample(d, k), simplify=FALSE), mean)
  p <- ggplot()+
    geom_histogram(aes(x = res), fill='white', col='black')+
    geom_vline(xintercept = mean(res), linetype="dotted",
               color = "blue", size=1.5)+
    ggtitle(sprintf("mean(x) = %f, k = %f", mean(res), k))
  list(p, c(k, mean(res), sd(res), sd(d)/sqrt(k)))
}



d <- rnorm(1e6, mean = 0, sd = 1)
df <- sapply(c(10, 50, 100, 500), function(x) get_properties(d, x))


df[1][[1]]
```



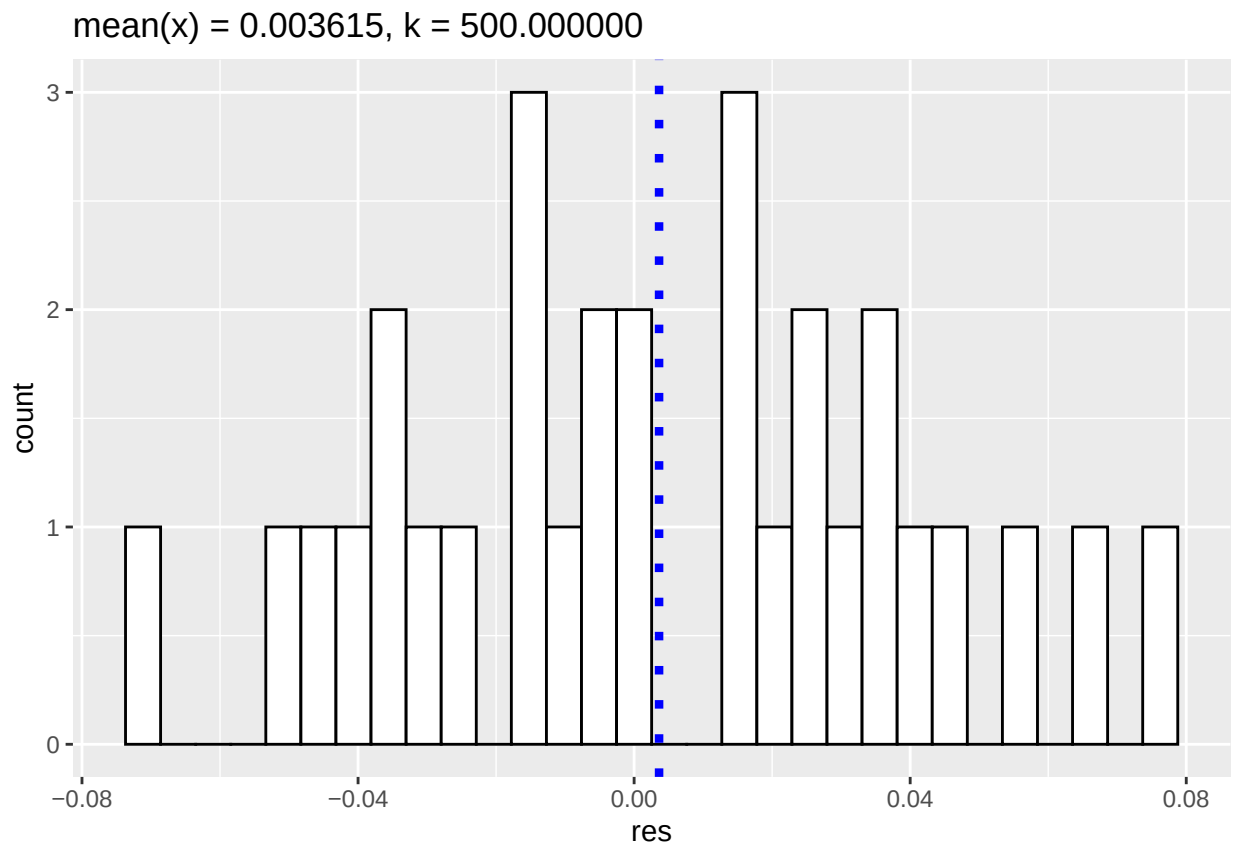

```
df[3][[1]]
```




```
df[5][[1]]
```




```
df[7][[1]]
```




```
df <- c(df[2], df[4], df[6], df[8])%>%
  as.data.frame() %>%
  t() %>%
  set_colnames(c("k", "mean(means)", "sd(means)", "se")) %>%
  set_rownames(1:4)

knitr::kable(df)
```

k	mean(means)	sd(means)	se
10	-0.0203438	0.3392652	0.3165548
50	0.0048375	0.1299833	0.1415676
100	0.0143917	0.1093768	0.1001034
500	0.0036154	0.0366488	0.0447676

The higher the number of observation, the closest mean(means) to the population mean, the smaller the sd(means) and the smaller the standard error.