HW2.0_Mary_Futey

Mary Futey

3/19/2020

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggpubr)
```

```
## Loading required package: ggplot2
## Loading required package: magrittr
```

```r
library(magrittr)
library(ggplot2)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
## The following object is masked from 'package:magrittr':
##
##     extract
```

1. Measures of center

1.0 Use given vector and write mode, median, mean functions/one-liners

1.1 Compare results for own and built-ins for median and mean

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

mode_self <- function(x) {
  uni_x <- unique(x)
  uni_x[which.max(tabulate(match(x, uni_x)))]
}

mode_self(x)
```

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

```r
median_self <- function(x) {
  x <- sort(x)
  if (length(x) / 2 != 0) {
    return(x[ceiling(length(x)/2)])
  } else {
    return((x[length(x)/2] + x[length(x)/2+1]) / 2)
  }
}

median_self(x)
```

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

```r
median(x)
```

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

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

mean_self(x)
```

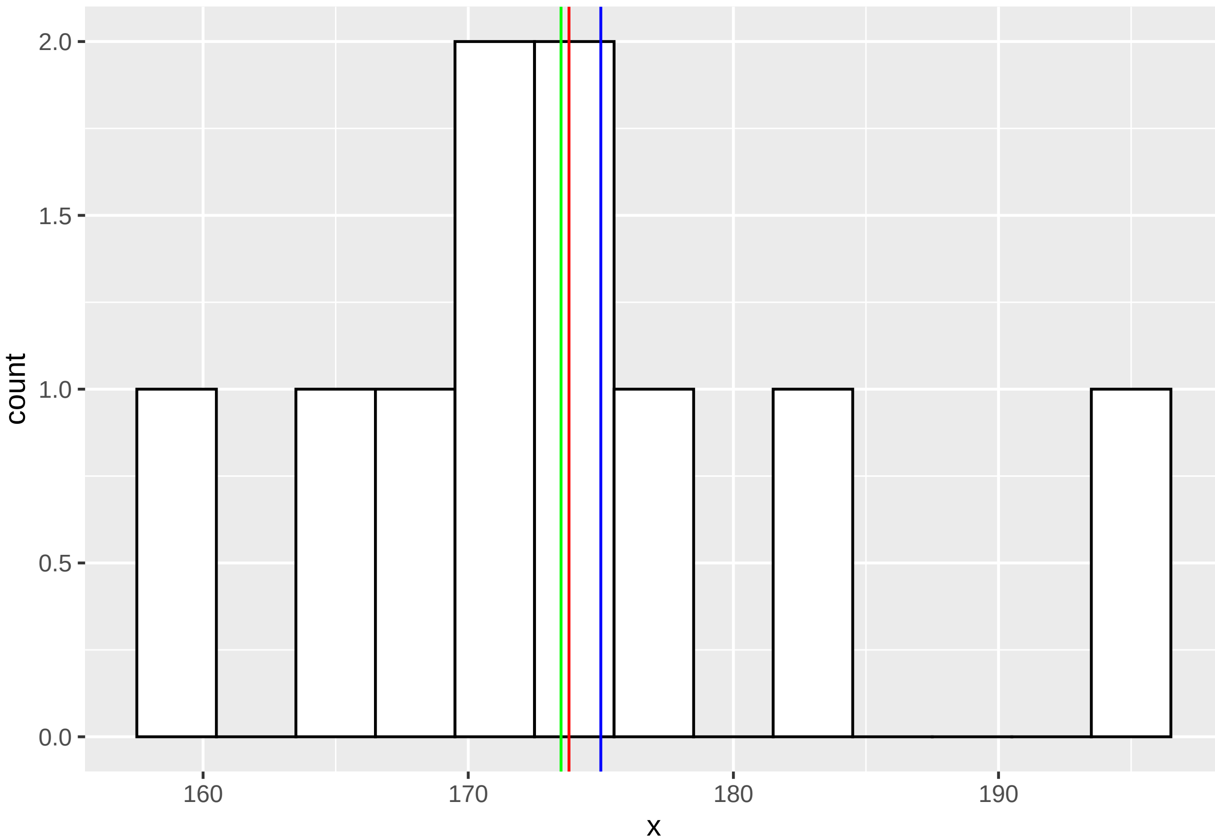
```
## [1] 173.8
```

```r
mean(x)
```

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

1.2 Visualize a histogram with 3 vertical lines for measures of center

```r
ggplot(as.data.frame(x),
       aes(x = x)) +
  geom_histogram(binwidth = 3, colour="black", fill="white") +
  geom_vline(xintercept = mean(x),
            col = "red") +
  geom_vline(xintercept = median(x),
             color = "green") +
  geom_vline(xintercept = mode_self(x),
             color = "blue")
```



1.3 Spoil your sample with an outlier - repeat steps 1.1 and 1.2

```
x_s <- c(175, 176, 180, 165, 167, 172, 175, 146, 158, 17)

mode_self(x_s)
```

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

```
mode(x_s)
```

```
## [1] "numeric"
```

```
median_self(x_s)
```

```
## [1] 167
```

```
median(x_s)
```

```
## [1] 169.5
```

```
mean_self(x_s)
```

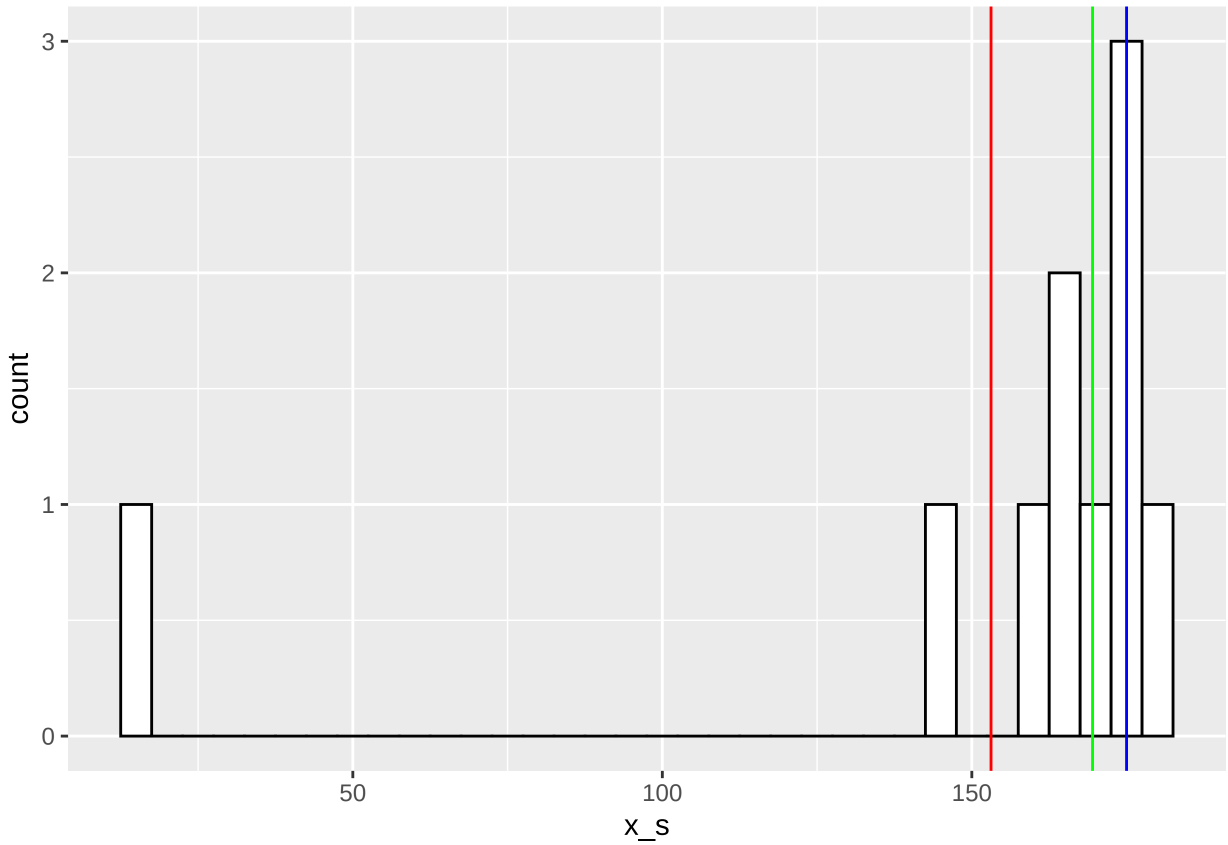
```
## [1] 153.1
```

```
mean(x_s)
```

```
## [1] 153.1
```

```
ggplot(as.data.frame(x_s),
       aes(x = x_s)) +
  geom_histogram(binwidth = 5, colour="black", fill="white") +
```

```
  geom_vline(xintercept = mean(x_s),
             col = "red") +
  geom_vline(xintercept = median(x_s),
              color = "green") +
  geom_vline(xintercept = mode_self(x_s),
              color = "blue")
```



2. Measures of spread

2.0 Functions for variance and sd, calculate result, compare with the built-ins

```
var_self <- function(x) {
    n <- length(x)
  m <- mean(x)
  return(sum((x-m)^2)/n)
}

var_self(x)
```

```
## [1] 94.76
```

```
var(x)
```

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

```
sd_self <- function(x) {
  return(var_self(x)^(.5))
}
```

```r
sd_self(x)
```

```
## [1] 9.734475
```

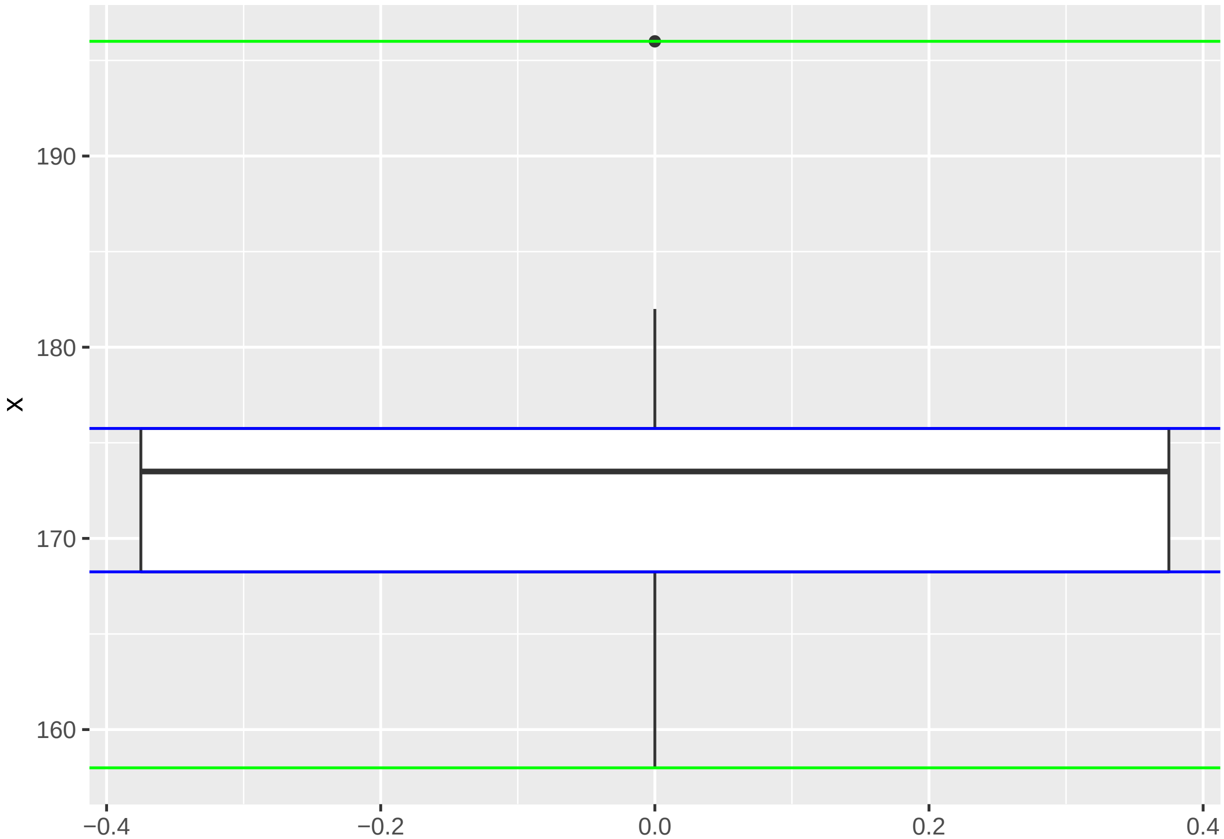
```r
sd(x)
```

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

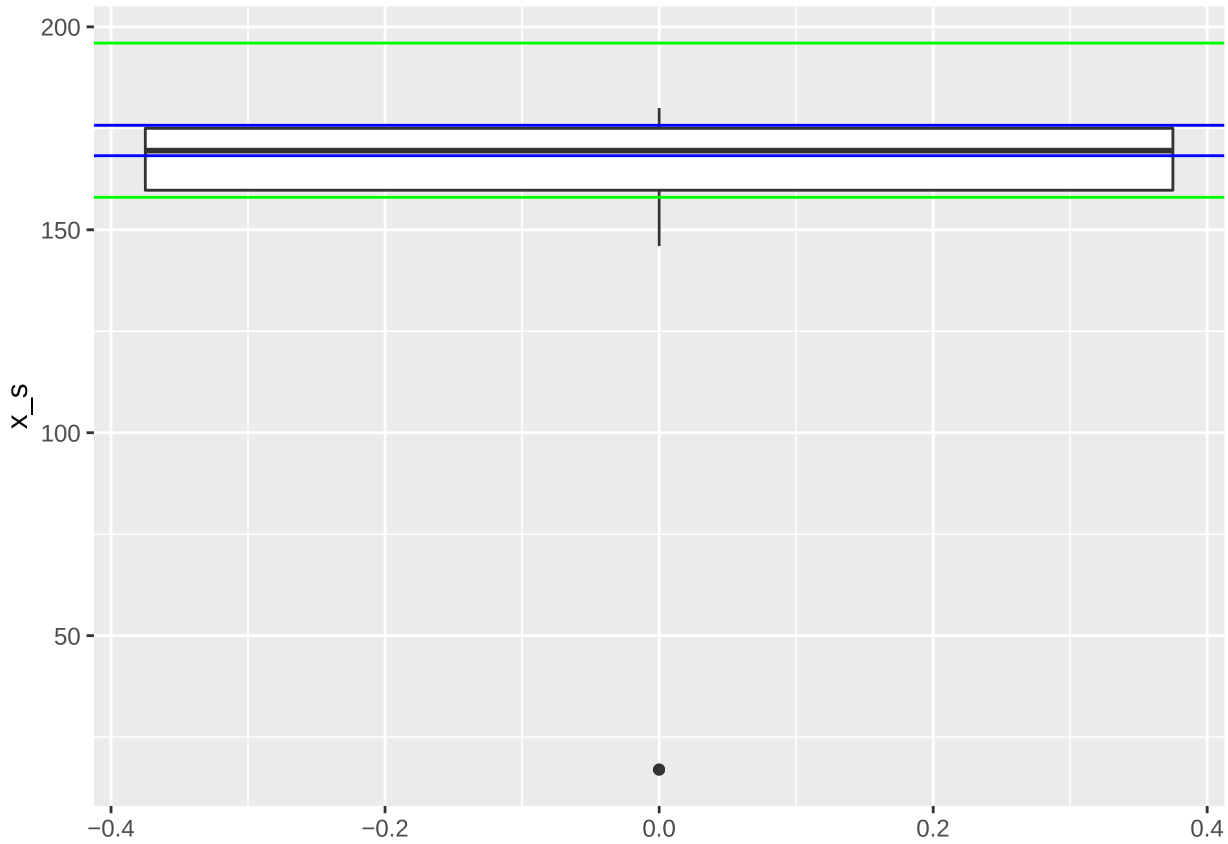
2.1 Visualize with the box plot and add horizontal lines for range, IQR, 1-sd borders (use built-ins)

2.2 Spoil your sample with the outlier, repeat step 2.1

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




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



3. 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?)

```
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
```

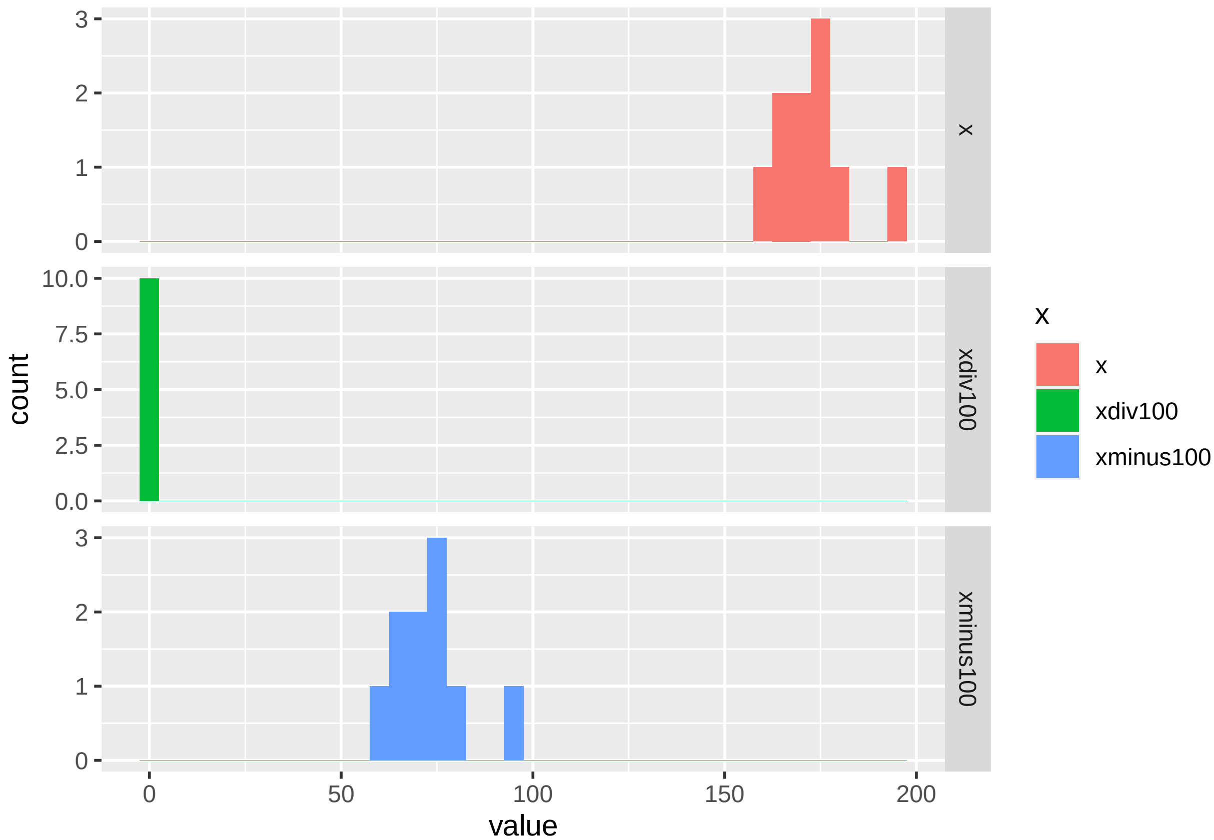
```
## [1] FALSE
```

```
x1 <- x - 100
x2 <- x / 100
```

```
table <- matrix(c(mean(x),mean(x1),mean(x2),
                  var(x), var(x1),var(x2),
                  sd(x),sd(x1),sd(x2)),
                ncol=3,byrow=FALSE)
colnames(table) <- c("Mean","Var","SD")
rownames(table) <- c("x","x - 100","x / 100")
table
```

```
##            Mean          Var         SD
## x       173.800 105.28888889 10.2610374
## x - 100  73.800 105.28888889 10.2610374
## x / 100   1.738   0.01052889  0.1026104
```

```
data <- data.frame("x" = x , "xminus100" = x1, "xdiv100" = x2)
data <- data %>% gather(x, value, x:xdiv100)
data %>% ggplot(aes(value, fill=x))+
  geom_histogram(binwidth = 5)+
  facet_grid(x ~., scales = 'free')
```



#

4. Normal distribution

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

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

```r
pnorm(156, mean = 175, sd = 10)
```

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

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

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

```r
pnorm(172, mean = 175, sd = 10) - pnorm(168, mean = 175, sd = 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
pnorm(1) - pnorm(-1)
```

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

```r
pnorm(2) - pnorm(-2)
```

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

```r
pnorm(3) - pnorm(-3)
```

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

Standardization

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

4.3 Standardize, find the same

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

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

```r
sd(sample)
```

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

```r
sample_std <- scale(sample)
mean(sample_std)
```

```
## [1] -2.744457e-16
```

```r
sd(sample_std)
```

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

5. Central Limit Theorem

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.

```r
set.seed(42)
pop <- rnorm(100000)

ten <- replicate(10, sample(pop, 30))
ten_mean <- colMeans(ten)

fifty <- replicate(50, sample(pop, 30))
fifty_mean <- colMeans(fifty)

hund <- replicate(100, sample(pop, 30))
hund_mean <- colMeans(hund)

fivehun <- replicate(500, sample(pop, 30))
fivehun_mean <- colMeans(fivehun)

std_error <- function(x) sqrt(var(x)/length(x))
table <- matrix(c(mean(ten_mean), sd(ten_mean), std_error(ten_mean),
                  mean(fifty_mean), sd(fifty_mean), std_error(fifty_mean),
                  mean(hund_mean), sd(hund_mean), std_error(hund_mean),
                  mean(fivehun_mean), sd(fivehun_mean), std_error(fivehun_mean)),
                  ncol = 3, byrow = TRUE)
colnames(table) <- c("mean","sd","st error")
rownames(table) <- c("10","50","100", "500")

table_final <- as.table(table)
table_final
```

```
##              mean          sd     st error
## 10    0.070195979  0.203574821  0.064376011
## 50   -0.028704060  0.153338710  0.021685368
## 100   0.014681682  0.176592748  0.017659275
## 500   0.003548260  0.180869332  0.008088722
```

##5.0 ### 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

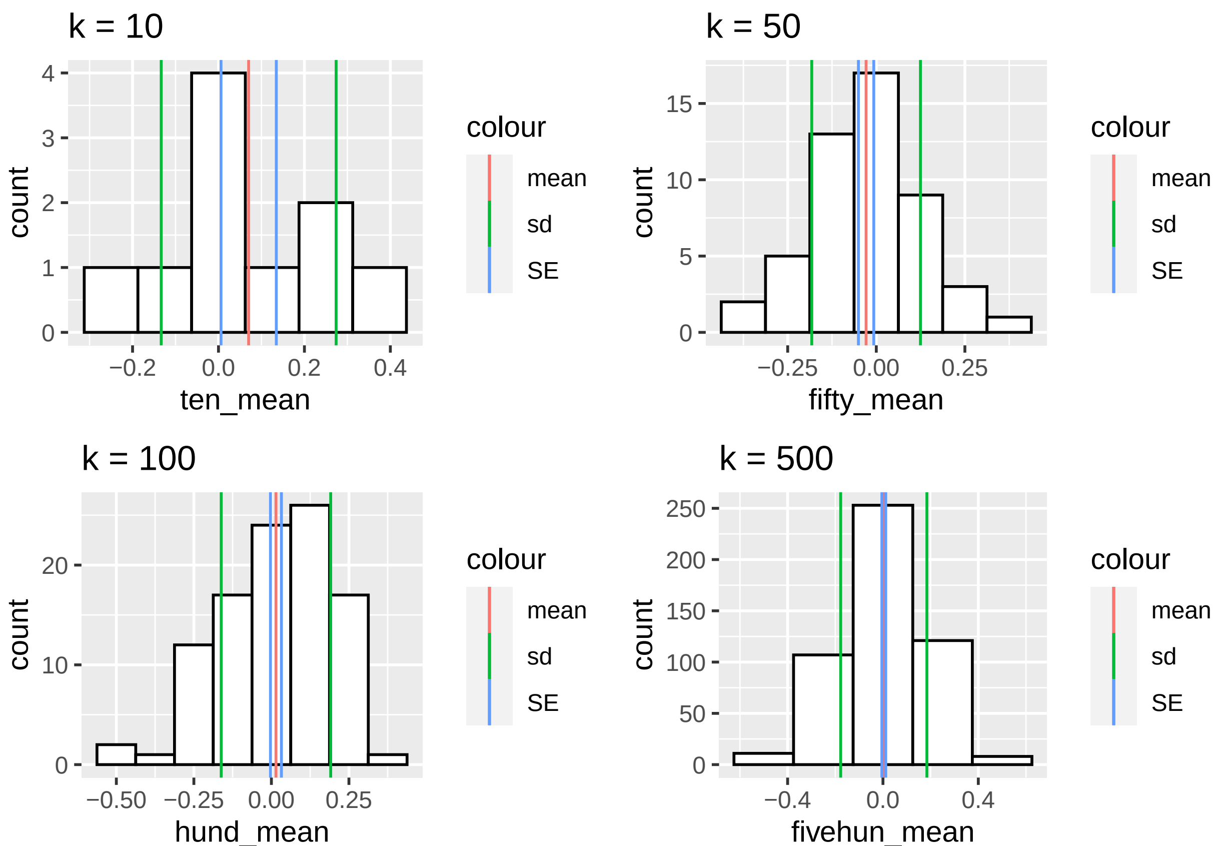
```r
ten_plot <- ggplot() +
  aes(ten_mean) +
  geom_histogram(binwidth=0.125, color="black", fill="white") +
  geom_vline(aes(xintercept=mean(ten_mean), color="mean")) +
  geom_vline(aes(xintercept=c(mean(ten_mean) + sd(ten_mean),
                        mean(ten_mean) - sd(ten_mean)), color="sd")) +
  geom_vline(aes(xintercept=c(mean(ten_mean) + std_error(ten_mean),
                        mean(ten_mean) - std_error(ten_mean)), color="SE")) +
  ggtitle(label = 'k = 10')

fifty_plot <- ggplot() +
  aes(fifty_mean) +
  geom_histogram(binwidth=0.125, color="black", fill="white") +
  geom_vline(aes(xintercept=mean(fifty_mean), color="mean")) +
  geom_vline(aes(xintercept=c(mean(fifty_mean) + sd(fifty_mean),
                        mean(fifty_mean) - sd(fifty_mean)), color="sd")) +
  geom_vline(aes(xintercept=c(mean(fifty_mean) + std_error(fifty_mean),
                        mean(fifty_mean) - std_error(fifty_mean)), color="SE")) +
  ggtitle(label = "k = 50")
```

```r
hund_plot <- ggplot() +
  aes(hund_mean) +
  geom_histogram(binwidth=0.125, color="black", fill="white") +
  geom_vline(aes(xintercept=mean(hund_mean), color="mean")) +
  geom_vline(aes(xintercept=c(mean(hund_mean) + sd(hund_mean),
                              mean(hund_mean) - sd(hund_mean)), color="sd")) +
  geom_vline(aes(xintercept=c(mean(hund_mean) + std_error(hund_mean),
                              mean(hund_mean) - std_error(hund_mean)), color="SE")) +
  ggtitle(label = 'k = 100')

fivehun_plot <- ggplot() +
  aes(fivehun_mean) +
  geom_histogram(binwidth=0.25, color="black", fill="white") +
  geom_vline(aes(xintercept=mean(fivehun_mean), color="mean")) +
  geom_vline(aes(xintercept=c(mean(fivehun_mean) + sd(fivehun_mean),
                              mean(fivehun_mean) - sd(fivehun_mean)), color="sd")) +
  geom_vline(aes(xintercept=c(mean(fivehun_mean) + std_error(fivehun_mean),
                              mean(fivehun_mean) - std_error(fivehun_mean)), color="SE")) +
  ggtitle(label = 'k = 500')

ggarrange(ten_plot, fifty_plot, hund_plot, fivehun_plot, ncol = 2, nrow = 2)
```



Comparison: More observations results in the mean being closer to the population mean, smaller sd and a smaller standard of error