Lab 4 - Foundations for statistical inference - Sampling distributions

Your name

Date of lab session

Lab report

Load data

```
ames <- read_csv("https://dyurovsky.github.io/85309/data/lab4/ames.csv")
```

```
## Rows: 2930 Columns: 82
```

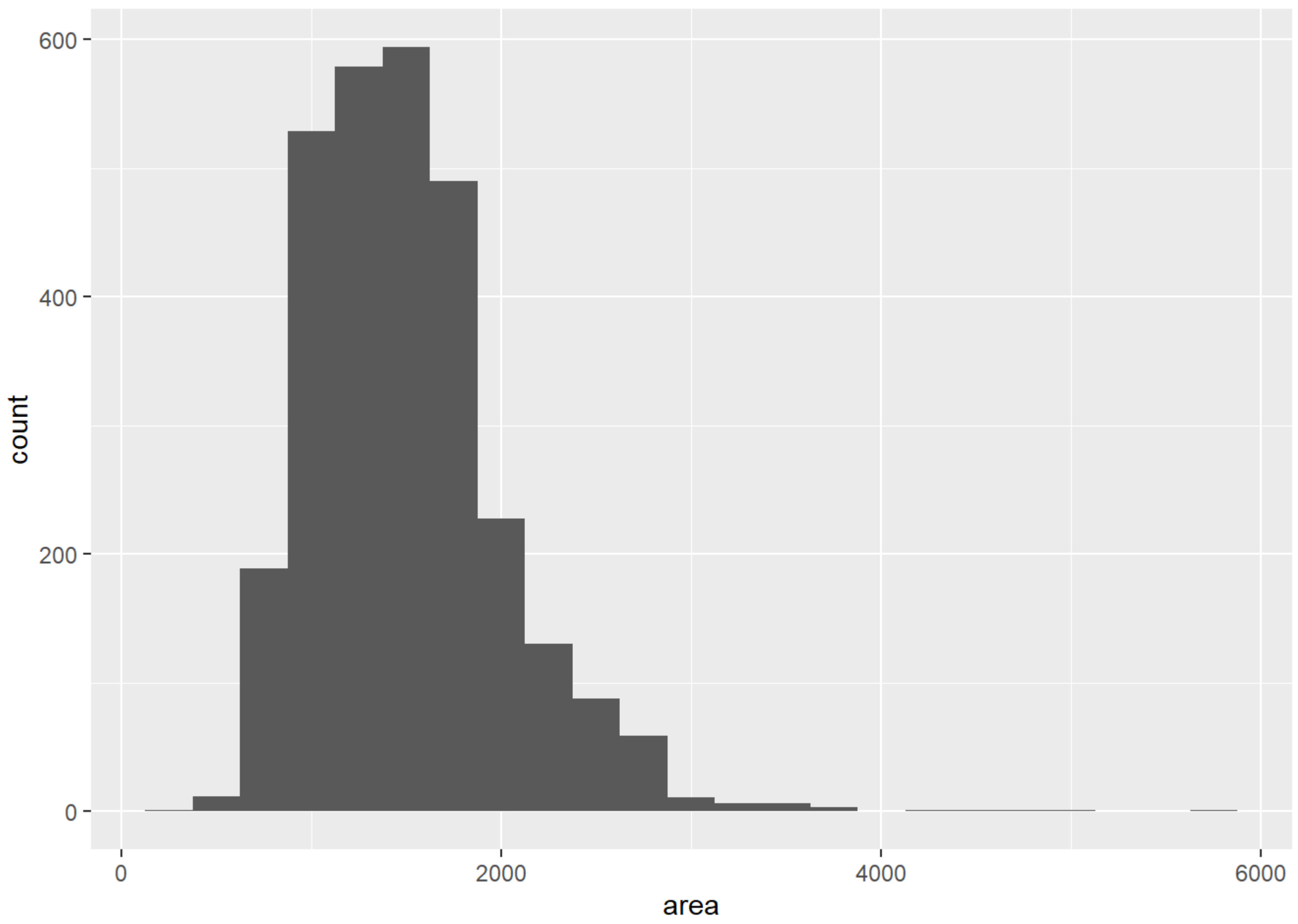
```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (43): MS.Zoning, Street, Alley, Lot.Shape, Land.Contour, Utilities, Lot....
## dbl (39): Order, PID, area, price, MS.SubClass, Lot.Frontage, Lot.Area, Over...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
set.seed(85309)
```

Exercise 1:

```
# enter your code for Exercise 1 here
ggplot(ames, aes(x = area)) +
  geom_histogram(binwidth = 250)
```



```
ames %>%
  summarise(mu = mean(area),
            pop_med = median(area),
            sigma = sd(area),
            pop_iqr = IQR(area))
```

```
## # A tibble: 1 x 4
##      mu pop_med sigma pop_iqr
##   <dbl>   <dbl> <dbl>   <dbl>
## 1 1500.    1442  506.    617.
```

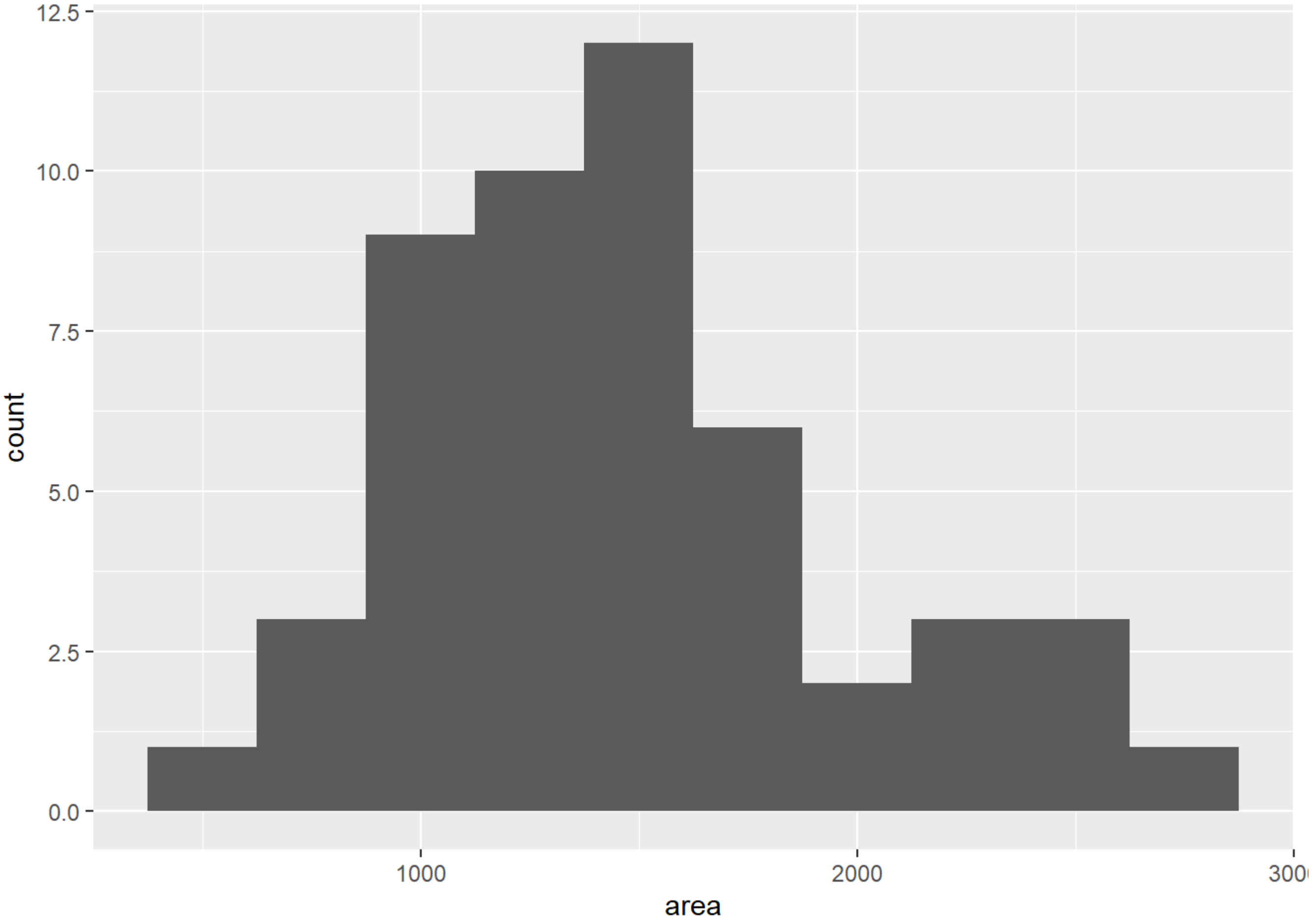
distribution of areas is unimodal and slightly right skewed with a few outliers in far right. because of skew and outliers, I’d like to use median and IQR rather than mean and sd. But I keep median and SD to compare it with the results of other questions below.

Exercise 2:

```
# enter your code for Exercise 2 here
set.seed(85309)

samp1 <- ames %>%
  sample_n(50)

ggplot(samp1, aes(x = area)) +
  geom_histogram(binwidth = 250)
```



```
sample1_tibble <- samp1 %>%
  summarise(mean1 = mean(area),
            med1 = median(area),
            SD1 = sd(area),
            iqr1 = IQR(area))
sample1_tibble
```

```
## # A tibble: 1 x 4
##   mean1  med1   SD1  iqr1
##   <dbl> <dbl> <dbl> <dbl>
## 1 1470. 1402.  497.  598.
```

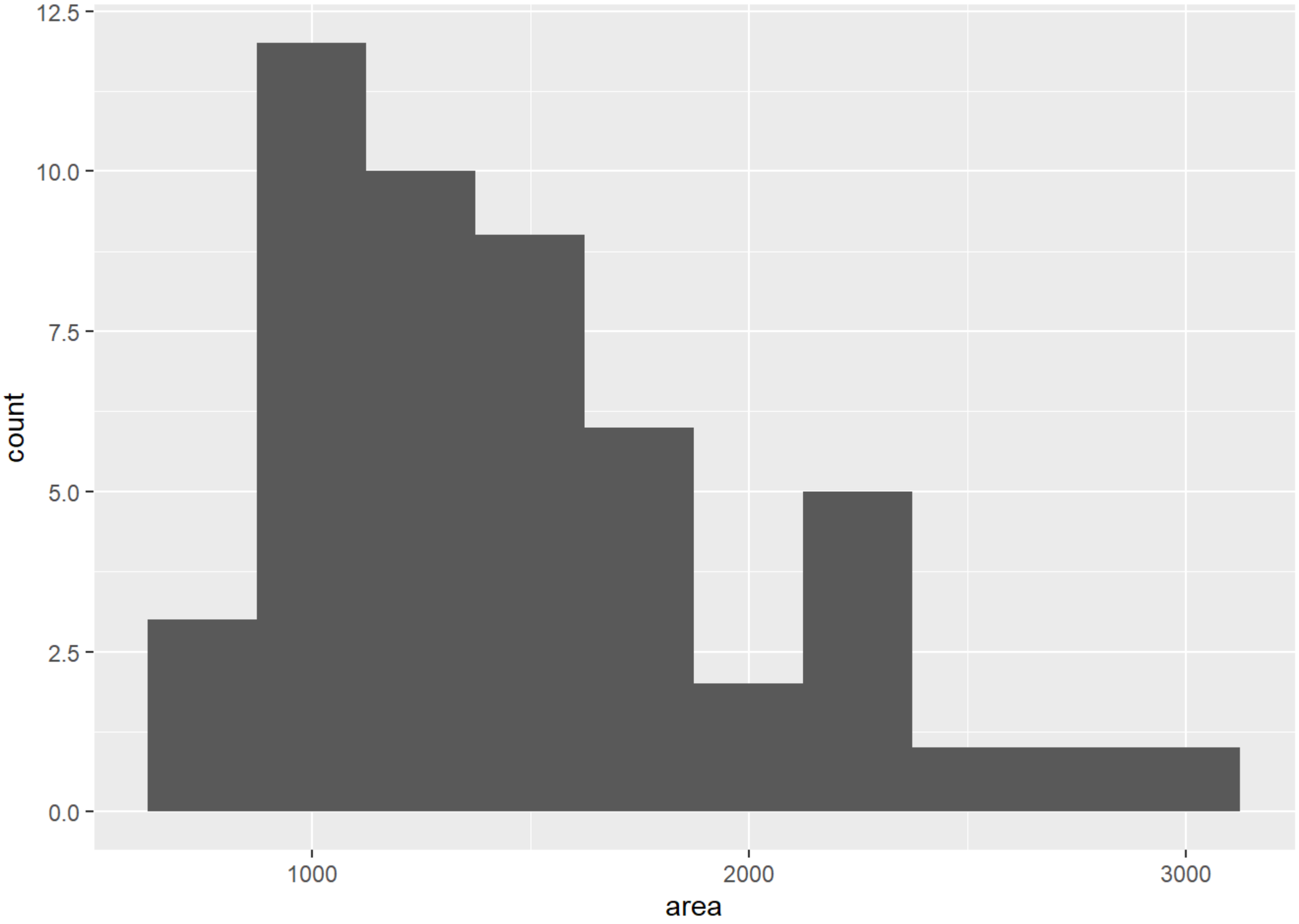
sample1_tibble is very similar to the population’s distribution of areas. It is unimodal and slightly right skewed. The mean is 1469.58 , the median is 1401.5 , the SD is 497.2234783 , the IQR is 597.75.

Exercise 3:

```
# enter your code for Exercise 3 here
set.seed(856609)

samp2 <- ames %>%
  sample_n(50)

ggplot(samp2, aes(x = area)) +
  geom_histogram(binwidth = 250)
```

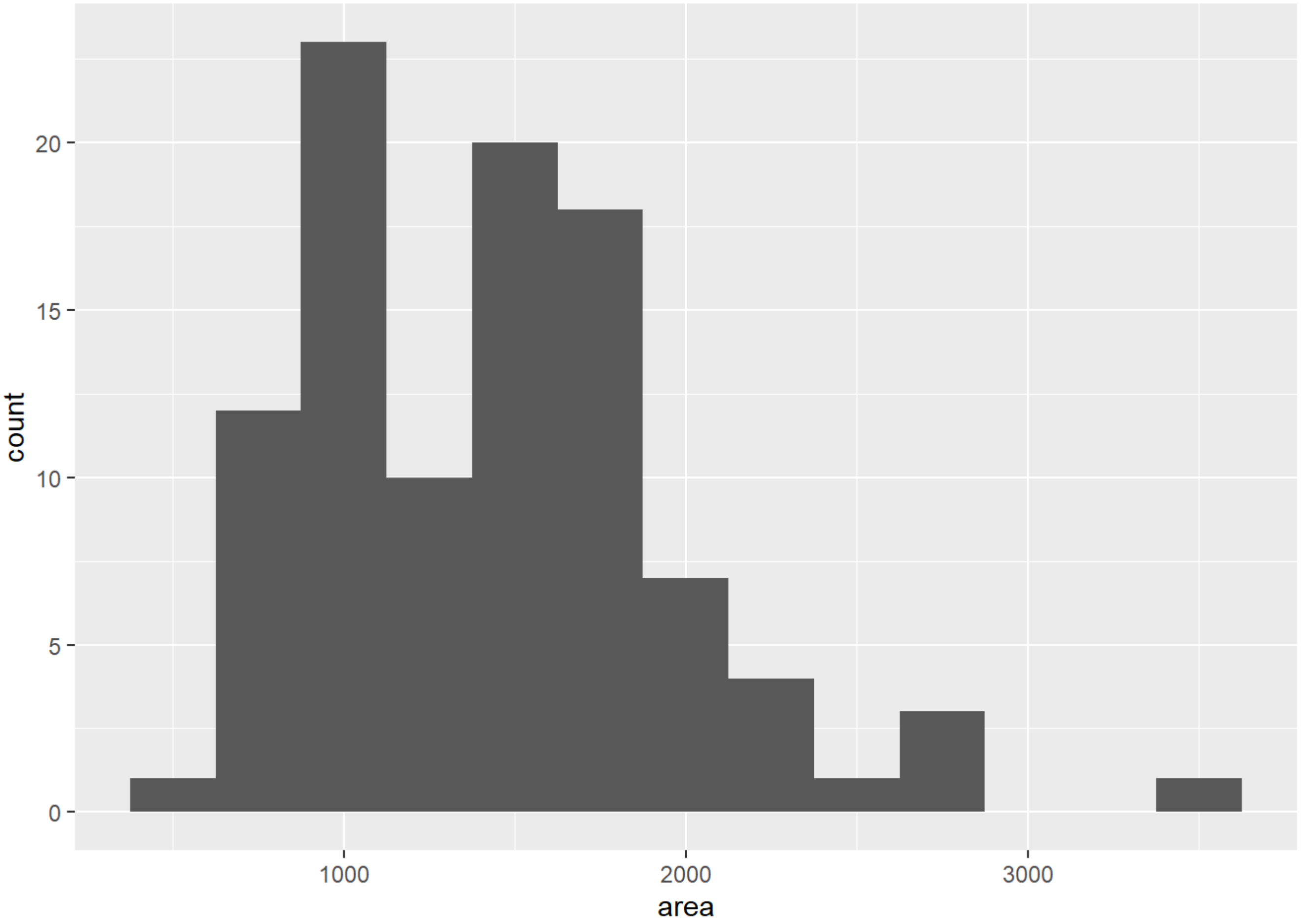


```
sample2_tibble <- samp2 %>%
  summarise(mean2 = mean(area),
            med2 = median(area),
            SD2 = sd(area),
            iqr2 = IQR(area))
sample2_tibble
```

```
## # A tibble: 1 x 4
##   mean2  med2   SD2  iqr2
##   <dbl> <dbl> <dbl> <dbl>
## 1 1494.  1394  523.  617.
```

```
samp3 <- ames %>%
  sample_n(100)

ggplot(samp3, aes(x = area)) +
  geom_histogram(binwidth = 250)
```

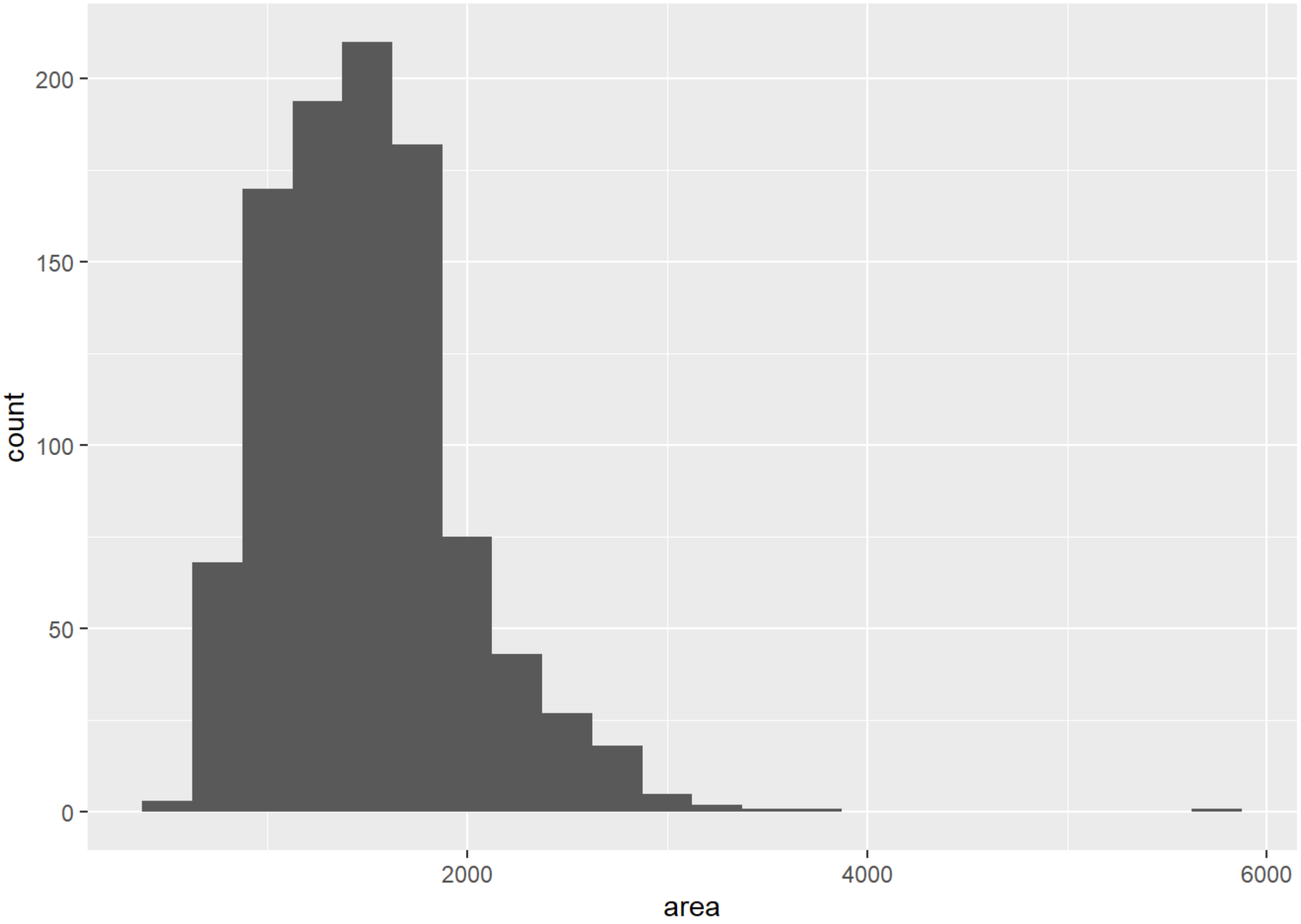


```
sample3_tibble <- samp3 %>%
  summarise(mean3 = mean(area),
            med3 = median(area),
            SD3 = sd(area),
            iqr3 = IQR(area))
sample3_tibble
```

```
## # A tibble: 1 x 4
##   mean3  med3   SD3  iqr3
##   <dbl> <dbl> <dbl> <dbl>
## 1 1443.  1450  525.   712
```

```
samp4 <- ames %>%
  sample_n(1000)

ggplot(samp4, aes(x = area)) +
  geom_histogram(binwidth = 250)
```



```
sample4_tibble <- samp4 %>%
  summarise(mean4 = mean(area),
            med4 = median(area),
            SD4 = sd(area),
            iqr4 = IQR(area))
sample4_tibble
```

```
## # A tibble: 1 x 4
##   mean4  med4   SD4  iqr4
##   <dbl> <dbl> <dbl> <dbl>
## 1 1497.  1453  493.  592.
```

means are quite similar but not exactly the same. I’d rather the size 1000 to estimate, because bigger the sample the more accurate the sample results

Exercise 4:

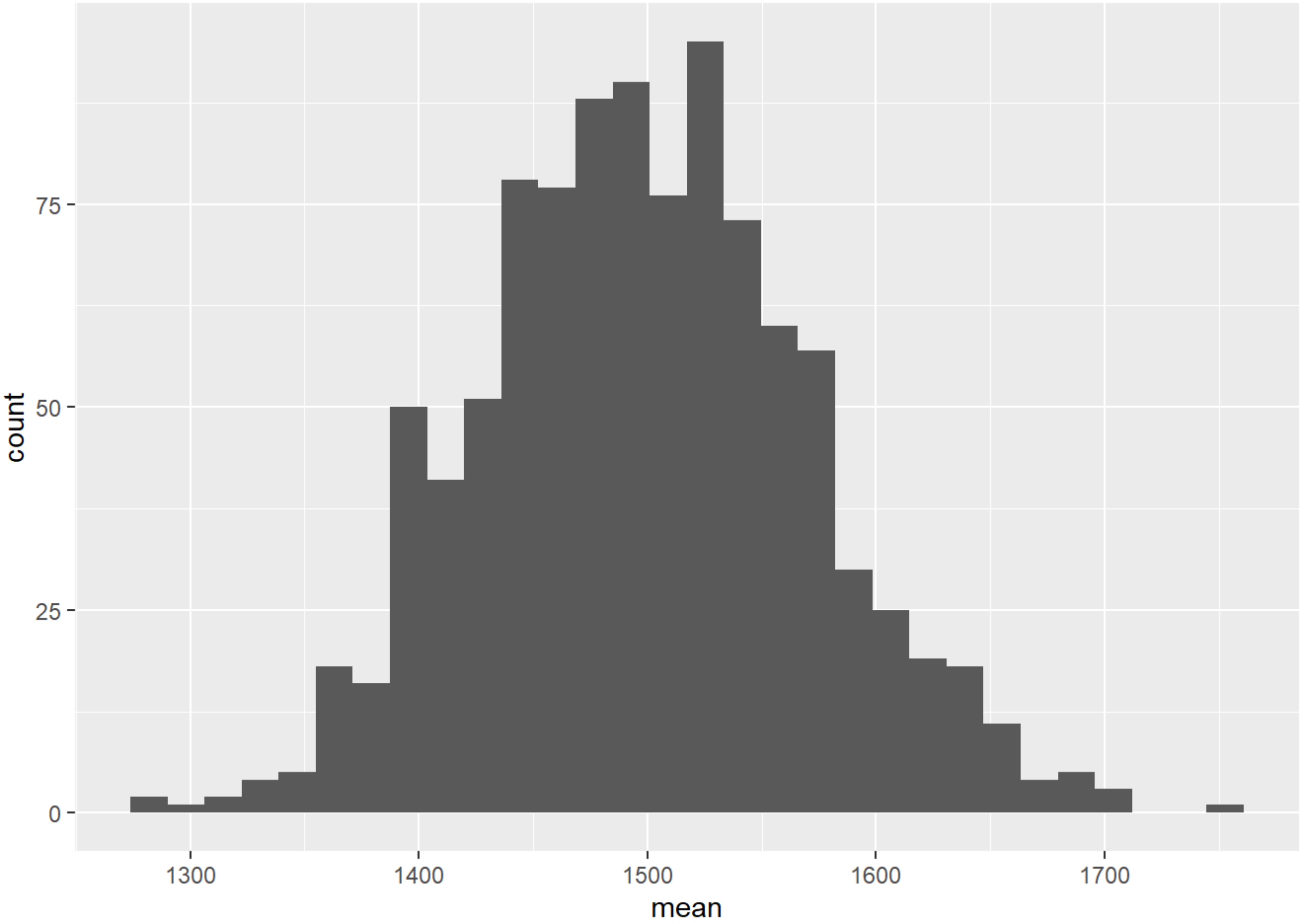
```
# enter your code for Exercise 4 here
set.seed(856609)

sample_50 <- function() {
  ames %>%
    sample_n(50, replace = TRUE) %>%
    summarise(x_bar = mean(area)) %>%
    pull()
}

sample_means50 <- tibble(sample = 1:1000,
                         mean = replicate(1000, sample_50()))

ggplot(sample_means50, aes(x = mean)) +
  geom_histogram()
```

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



```
sample_means_mean <- sample_means50 %>%
  summarise(mean = mean(mean))

sample_means_mean
```

```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1 1499.
```

```
nrow(sample_means50)
```

```
## [1] 1000
```

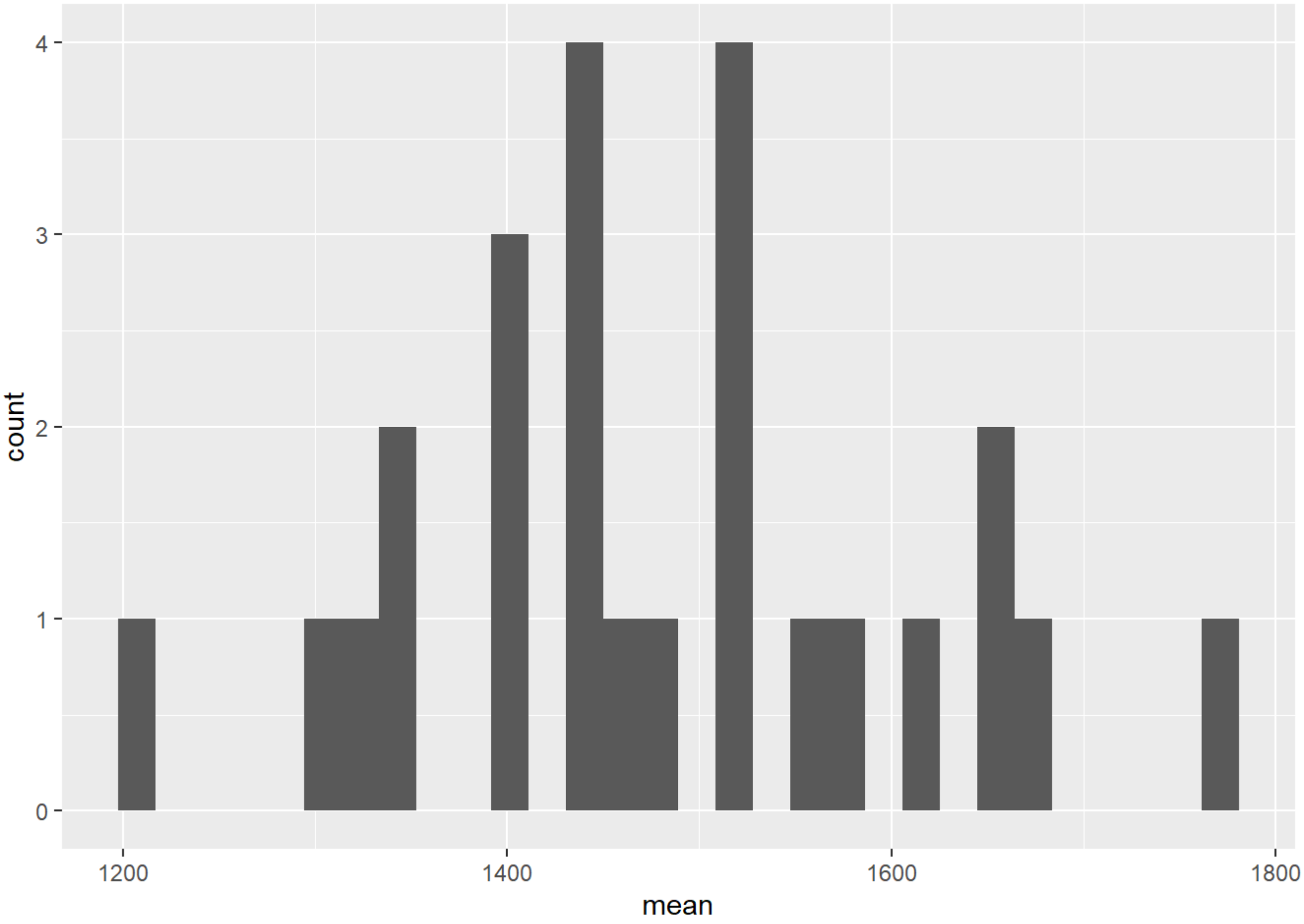
There should be 1000 elements. this is normal distribution and it is centered around population mean. It has 1498.68838 which is very close to the population mean.

Exercise 5:

```
# enter your code for Exercise 5 here
set.seed(856609)

sample_means_small <- tibble(sample = 1:25,
                         mean = replicate(25, ames %>%
                                            sample_n(10, replace = TRUE) %>%
                                            summarise(x_bar = mean(area)) %>%
                                            pull()))
ggplot(sample_means_small, aes(x = mean)) +
  geom_histogram()
```

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



```
sample_means_mean <- sample_means_small %>%
  summarise(mean = mean(mean))

sample_means_small
```

```
## # A tibble: 25 x 2
##    sample  mean
##     <int> <dbl>
##  1      1 1439.
##  2      2 1432.
##  3      3 1772.
##  4      4 1614.
##  5      5 1331.
##  6      6 1403.
##  7      7 1572.
##  8      8 1444.
##  9      9 1347.
## 10     10 1678.
## # ... with 15 more rows
```

```
sample_means_mean
```

```
## # A tibble: 1 x 1
##    mean
##   <dbl>
## 1 1481.
```

```
nrow(sample_means_small)
```

```
## [1] 25
```

I have 25 observations each of which represents the mean of the sample of size 10.

Exercise 6:

```
# enter your code for Exercise 6 here
set.seed(856609)

sample_varying <- function(n) {
  ames %>%
    sample_n(n, replace = TRUE) %>%
    summarise(x_bar = mean(area)) %>%
    pull()
}

sample_means20 <- tibble(sample = 1:1000,
                         x_bar = replicate(1000, sample_varying(20)))

descriptors_20 <- sample_means20 %>%
  summarise(mean= mean(x_bar),
            sd = sd(x_bar))

descriptors_20
```

```
## # A tibble: 1 x 2
##    mean    sd
##   <dbl> <dbl>
## 1 1500.  113.
```

```
sample_means100 <- tibble(sample = 1:1000,
                         x_bar = replicate(1000, sample_varying(100)))

descriptors_100 <- sample_means100 %>%
  summarise(mean= mean(x_bar),
            sd = sd(x_bar))

descriptors_100
```

```
## # A tibble: 1 x 2
##    mean    sd
##   <dbl> <dbl>
## 1 1500.  50.7
```

```
sample_means1000 <- tibble(sample = 1:1000,
                         x_bar = replicate(1000, sample_varying(1000)))

descriptors_1000 <- sample_means1000 %>%
  summarise(mean= mean(x_bar),
            sd = sd(x_bar))

descriptors_1000
```

```
## # A tibble: 1 x 2
##    mean    sd
##   <dbl> <dbl>
## 1 1500.  16.0
```

```
sample_means20
```

```
## # A tibble: 1,000 x 2
##    sample x_bar
##     <int> <dbl>
##  1      1 1436.
##  2      2 1693.
##  3      3 1367.
##  4      4 1508.
##  5      5 1512.
##  6      6 1364.
##  7      7 1481.
##  8      8 1425
##  9      9 1551.
## 10     10 1440.
## # ... with 990 more rows
```

```
#playing with the number of simulations

sample_means20_simulations <- tibble(sample = 1:10,
                         x_bar = replicate(10, sample_varying(20)))

descriptors_20_simulations <- sample_means20_simulations %>%
  summarise(mean= mean(x_bar),
            sd = sd(x_bar))

descriptors_20_simulations
```

```
## # A tibble: 1 x 2
##    mean    sd
##   <dbl> <dbl>
## 1 1542.  70.9
```

```
sample_means100_simulations <- tibble(sample = 1:10,
                         x_bar = replicate(10, sample_varying(100)))

descriptors_100_simulations <- sample_means100_simulations %>%
  summarise(mean= mean(x_bar),
            sd = sd(x_bar))

descriptors_100_simulations
```

```
## # A tibble: 1 x 2
##    mean    sd
##   <dbl> <dbl>
## 1 1480.  47.7
```

```
sample_means1000_simulations <- tibble(sample = 1:10,
                         x_bar = replicate(10, sample_varying(1000)))

descriptors_1000_simulations <- sample_means1000_simulations %>%
  summarise(mean= mean(x_bar),
            sd = sd(x_bar))

descriptors_1000_simulations
```

```
## # A tibble: 1 x 2
##    mean    sd
##   <dbl> <dbl>
## 1 1500.  17.2
```

Each observation of any of the samples here represents the mean of the sample of the size that it has. By increasing the sample size, the mean is about the same and SD is changing and it is getting smaller and smaller if the size of sample is getting bigger. As the sample size increase, the normal distribution is getting more smoother in its bell-shape. in terms of numbers of simulations, as we have more simulations, our mean is getting more accurate. ### More Practice:

Exercise 7:

```
# enter your code for Exercise 7 here
set.seed(856609)
sample_varying2 <- function(n) {
  ames %>%
    sample_n(n, replace = TRUE) %>%
    summarise(x_bar = mean(price)) %>%
    pull()
}


sample_varying2(15)
```

```
## [1] 166653.3
```

The sample’s mean should be close to the population’s mean. So population’s mean should be around 166653.3.

Exercise 8:

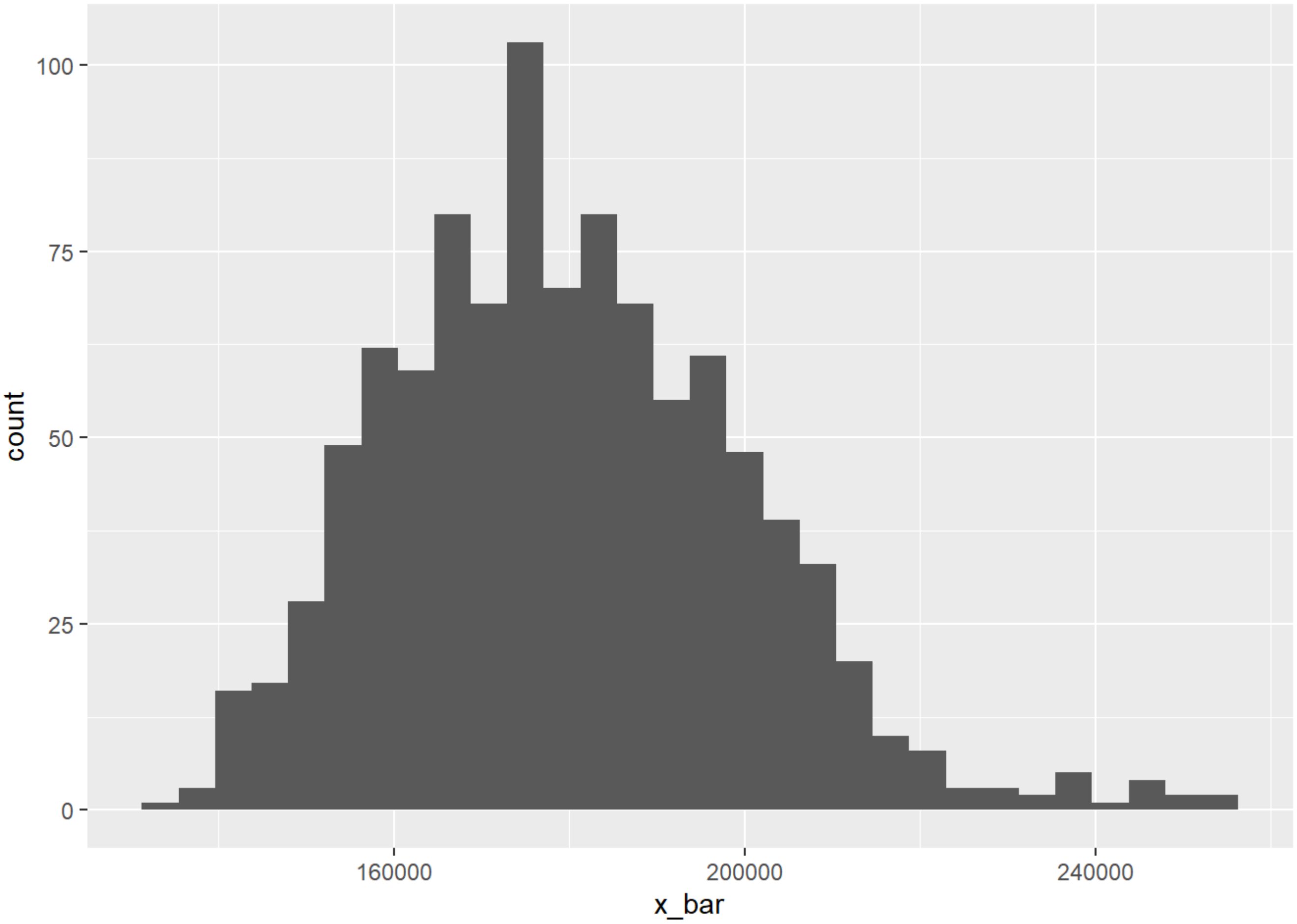
```
# enter your code for Exercise 8 here
set.seed(85649)
sample_varying2 <- function(n) {
  ames %>%
    sample_n(n, replace = TRUE) %>%
    summarise(x_bar = mean(price)) %>%
    pull()
}

sample_means15 <- tibble(sample = 1:1000,
                         x_bar = replicate(1000, sample_varying2(15)))




ggplot(sample_means15, aes(x = x_bar)) +
  geom_histogram()
```

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



```
sample_means_mean <- sample_means15 %>%
  summarise(mean = mean(x_bar),
            sd = sd(x_bar))

sample_means_mean
```

```
## # A tibble: 1 x 2
##      mean     sd
##     <dbl>  <dbl>
## 1 179790. 20005.
```

```
ames %>%
  summarise(mu = mean(price))
```

```
## # A tibble: 1 x 1
##        mu
##     <dbl>
## 1 180796.
```

the sampling distribution is a bel-shaped normal distribution with the mean of 179790.2, and SD of 2004.92. I am guessing that the mean home price of the population be close to the mean of this sampling distribution which after calculating the population mean, It came to be 180796.1 which is close to the 179790.2.

Exercise 9:

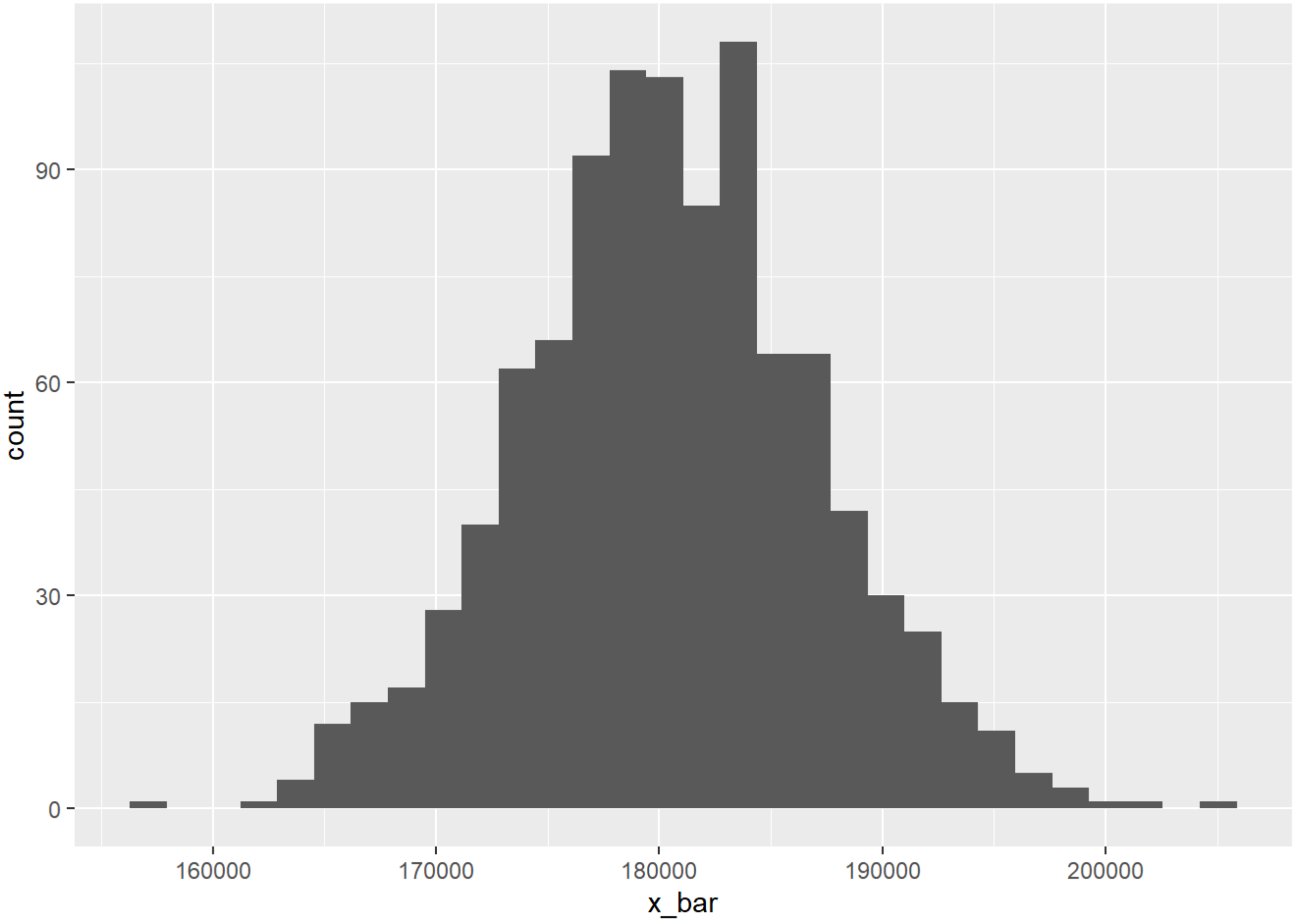
```
# enter your code for Exercise 9 here
set.seed(85629)
sample_varying2 <- function(n) {
  ames %>%
    sample_n(n, replace = TRUE) %>%
    summarise(x_bar = mean(price)) %>%
    pull()
}

sample_means150 <- tibble(sample = 1:1000,
                          x_bar = replicate(1000, sample_varying2(150)))




ggplot(sample_means150, aes(x = x_bar)) +
  geom_histogram()
```

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



```
sample_means_mean <- sample_means150 %>%
  summarise(mean = mean(x_bar),
            sd = sd(x_bar))

sample_means_mean
```

```
## # A tibble: 1 x 2
##      mean    sd
##     <dbl> <dbl>
## 1 180466. 6640.
```

```
ames %>%
  summarise(mu = mean(price))
```

```
## # A tibble: 1 x 1
##        mu
##     <dbl>
## 1 180796.
```

the sampling distribution is again a bel-shaped normal distribution but it is very smoother and more concentrated towards the mean, and the bell-shape is more visible compared to the sample_means15. So the Standard deviation is much smaller when the sample size is 150.It is 6639. Its mean is 180465.5 which is about similar to the mean of the sample_means15 however, the SD of 150 is way smaller than sample size of 15. compared to the sample of size 15,I am guessing that the mean home price of the population be even closer to the mean of this sampling distribution of size 150 which after calculating the population mean, It came to be 180796.1 which is very close to the 180465.5.

Exercise 10:

```
# enter your code for Exercise 10 here
```

sampling distribution of exercise 9 with the sample size of 150 has the smaller spread or smaller variability or smaller SD. And if we are concerned with accurate estimation, we should select a sampling distribution of smaller spread, which happen with the samples with higher sizes. Because as the sample size increases, the estimations are more accurate.