Lab 4a - Foundations for statistical inference - Sampling distributions

Michael Y.

March 17th, 2019

In this lab, we investigate the ways in which the statistics from a random sample of data can serve as point estimates for population parameters. We're interested in formulating a *sampling distribution* of our estimate in order to learn about the properties of the estimate, such as its distribution.

The data

We consider real estate data from the city of Ames, Iowa. The details of every real estate transaction in Ames is recorded by the City Assessor's office. Our particular focus for this lab will be all residential home sales in Ames between 2006 and 2010. This collection represents our population of interest. In this lab we would like to learn about these home sales by taking smaller samples from the full population. Let's load the data.

```
load("more/ames.RData")
```

We see that there are quite a few variables in the data set, enough to do a very in-depth analysis. For this lab, we'll restrict our attention to just two of the variables: the above ground living area of the house in square feet (`Gr.Liv.Area`) and the sale price (`SalePrice`). To save some effort throughout the lab, create two variables with short names that represent these two variables.

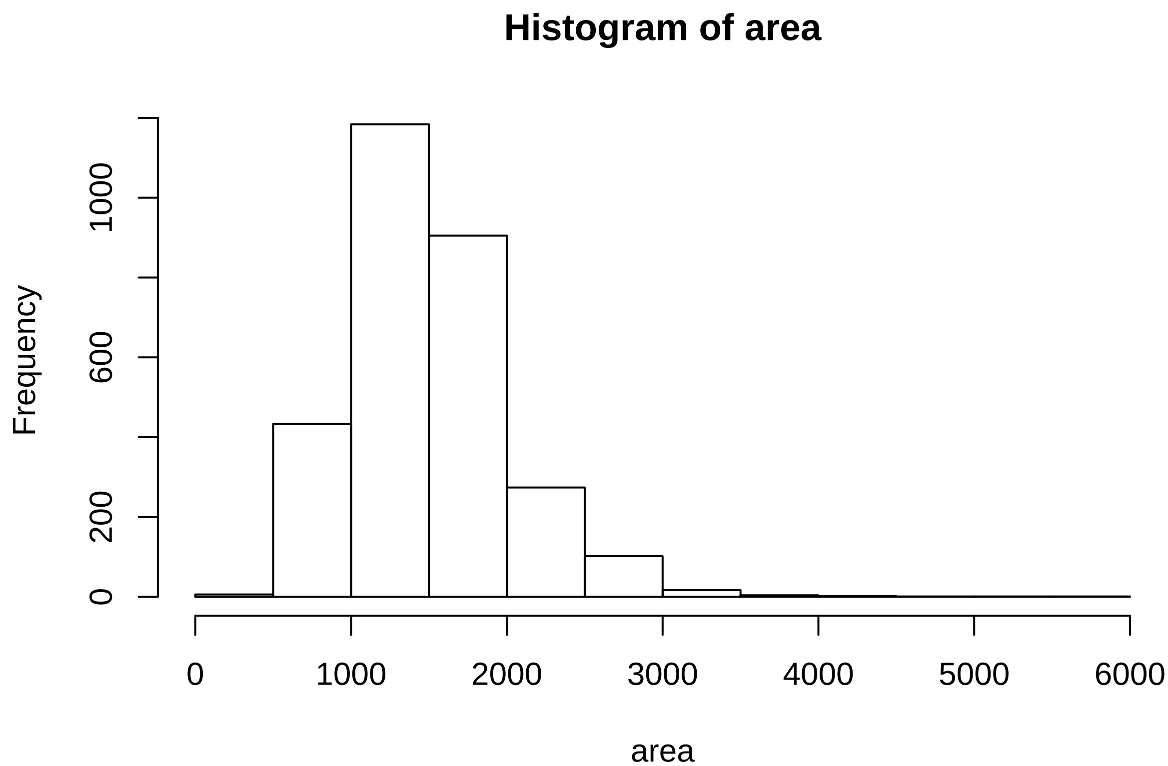
```
area <- ames$Gr.Liv.Area
price <- ames$SalePrice
```

Let's look at the distribution of area in our population of home sales by calculating a few summary statistics and making a histogram.

```
summary(area)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334    1126    1442    1500    1743    5642
```

```
hist(area)
```



1. Describe this population distribution.

The distribution of areas of houses in Ames is unimodal and right-skewed, as the mean (1,500sf) is greater than the median (1,442sf). There are 2,930 observations.

The middle 50% of the houses range between 1,126 square feet and 1,743 square feet, so the IQR is 617 square feet. The smallest house is 334 square feet and the largest is 5,642 square feet.

End of response to Exercise 1 .

.#.

The unknown sampling distribution

In this lab we have access to the entire population, but this is rarely the case in real life. Gathering information on an entire population is often extremely costly or impossible. Because of this, we often take a sample of the population and use that to understand the properties of the population.

If we were interested in estimating the mean living area in Ames based on a sample, we can use the following command to survey the population.

Sample 1, size = 50

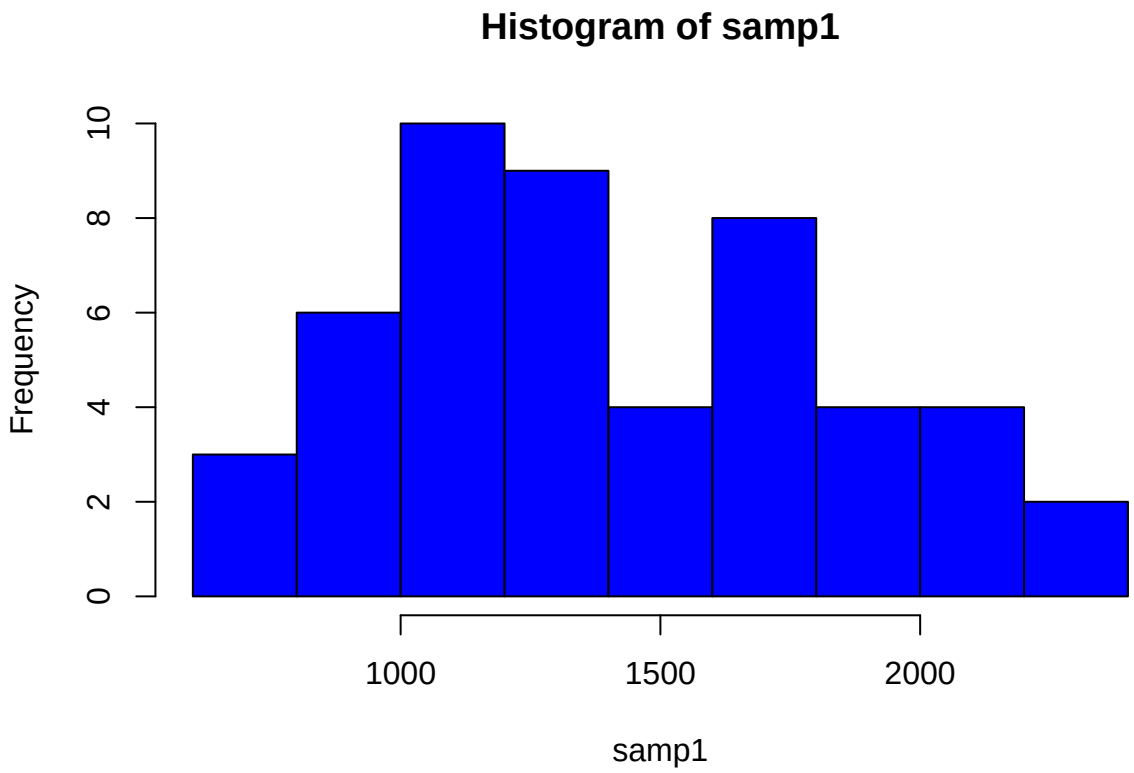
```
samp1 <- sample(area, 50)
```

This command collects a simple random sample of size 50 from the vector `area`, which is assigned to `samp1`. This is like going into the City Assessor's database and pulling up the files on 50 random home sales. Working with these 50 files would be considerably simpler than working with all 2930 home sales.

2. Describe the distribution of this sample. How does it compare to the distribution of the population?

Histogram of sample 1:

```
hist(samp1,breaks=11,col = "blue")
```



Sample 1 statistics:

```
summarys1 <- summary(samp1)
summarys1
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     747    1077    1301    1402    1671    2385
```

```
mins1  <- as.numeric(summarys1["Min."])
means1 <- round(as.numeric(summarys1["Mean"]),2)
meds1  <- as.numeric(summarys1["Median"])
maxs1  <- as.numeric(summarys1["Max."])
```

```
iqrs1  <- as.numeric(IQR(samp1))
cat(paste("Inter-Quartile Range of the sample: ",iqrs1,"\n"))
```

```
## Inter-Quartile Range of the sample:  594
```

```
stdevs1 <- round(as.numeric(sd(samp1)),2)
cat(paste("Standard Deviation of the sample: ",stdevs1,"\n"))
```

```
## Standard Deviation of the sample:  421.36
```

Population statistics:

```
summarypop <- summary(area)
summarypop
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334    1126    1442    1500    1743    5642
```

```
minpop  <- as.numeric(summarypop["Min."])
meanpop <- round(as.numeric(summarypop["Mean"]),2)
medpop  <- as.numeric(summarypop["Median"])
maxpop  <- as.numeric(summarypop["Max."])
iqrpop  <- as.numeric(IQR(area))
cat(paste("Inter-Quartile Range of the population: ",iqrpop,"\n"))
```

```
## Inter-Quartile Range of the population:  616.75
```

```
stdevpop <- round(as.numeric(sd(area)),2)
cat(paste("Standard Deviation of the population: ",stdevpop,"\n"))
```

```
## Standard Deviation of the population:  505.51
```

Z-Scores:

```
s1medZscore = round((meds1 - medpop) / stdevpop, 4)
cat(paste("Z-Score of the sample median vs. the population: ",s1medZscore,"\n"))
```

```
## Z-Score of the sample median vs. the population:  -0.2789
```

```
s1meanZscore = round((means1 - meanpop) / stdevpop, 4)
cat(paste("Z-Score of the sample mean vs. the population: ",s1meanZscore,"\n"))
```

```
## Z-Score of the sample mean vs. the population:  -0.1931
```

Similar to the population, this sample is a unimodal distribution which appears close to Normal.

The sample has an IQR of 594, which is tighter than the population IQR, 616.75.

The sample has a StandardDeviation of 421.36, which is less than the population StandardDeviation, 505.51.

The sample has a median of 1301, which is less than the population median, 1442.

The sample median is -0.28 standard deviations below the population median.

The sample has a mean of 1402.08, which is less than the population mean, 1499.69.

The sample mean is -0.19 standard deviations below the population mean.

The sample mean, 1402.08, is greater than the sample median, 1301, which suggests a right skew on this sample.

NB: The above results will change each time the sampling is rerun (which includes each time the file is re-knit.)

The population mean, 1499.69, is greater than the population median, 1442, which indicates a right skew on the population.

End of response to Exercise 2 .

.#.

If we're interested in estimating the average living area in homes in Ames using the sample, our best single guess is the sample mean.

```
mean(samp1)
```

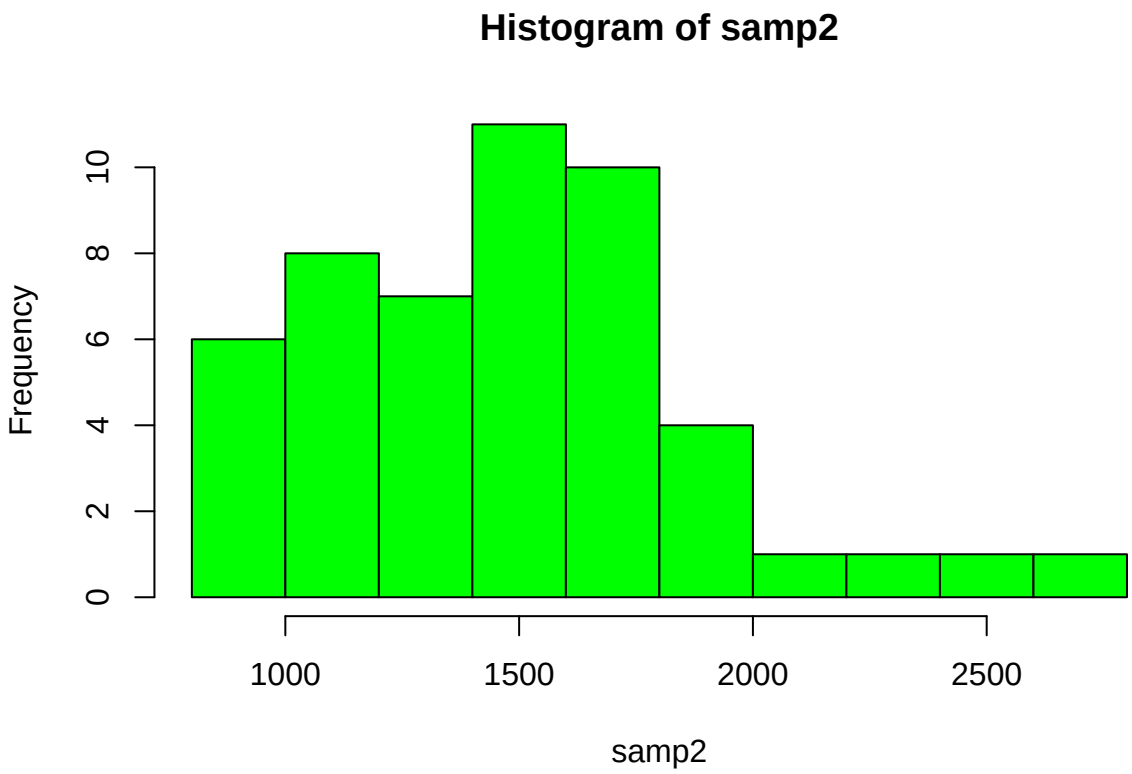
```
## [1] 1402
```

Depending on which 50 homes you selected, your estimate could be a bit above or a bit below the true population mean of 1499.69 square feet. In general, though, the sample mean turns out to be a pretty good estimate of the average living area, and we were able to get it by sampling less than 3% of the population.

3. Take a second sample, also of size 50, and call it `samp2`. How does the mean of `samp2` compare with the mean of `samp1`? Suppose we took two more samples, one of size 100 and one of size 1000. Which would you think would provide a more accurate estimate of the population mean?

Sample 2, size = 50

```
samp2 <- sample(area, 50)
hist(samp2,breaks=11,col="green")
```



sample 2 statistics

```
summarys2 <- summary(samp2)
summarys2
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     816    1184    1468    1486    1718    2772
```

```
mins2  <- as.numeric(summarys2["Min."])
means2 <- round(as.numeric(summarys2["Mean"]),2)
meds2  <- as.numeric(summarys2["Median"])
maxs2  <- as.numeric(summarys2["Max."])
iqrs2  <- as.numeric(IQR(samp2))
cat(paste("Inter-Quartile Range of sample 2: ",iqrs2,"\n"))
```

```
## Inter-Quartile Range of sample 2:  533.5
```

```
stdevs2 <- round(as.numeric(sd(samp2)),2)
cat(paste("Standard Deviation of sample 2: ",stdevs2,"\n"))
```

```
## Standard Deviation of sample 2:  422.12
```

```
s2medZscore = round((meds2 - medpop) / stdevpop, 4)
cat(paste("Z-Score of sample 2 median vs. the population: ",s2medZscore,"\n"))
```

```
## Z-Score of sample 2 median vs. the population:  0.0504
```

```
s2meanZscore = round((means2 - meanpop) / stdevpop, 4)
cat(paste("Z-Score of sample 2 mean vs. the population: ",s2meanZscore,"\n"))
```

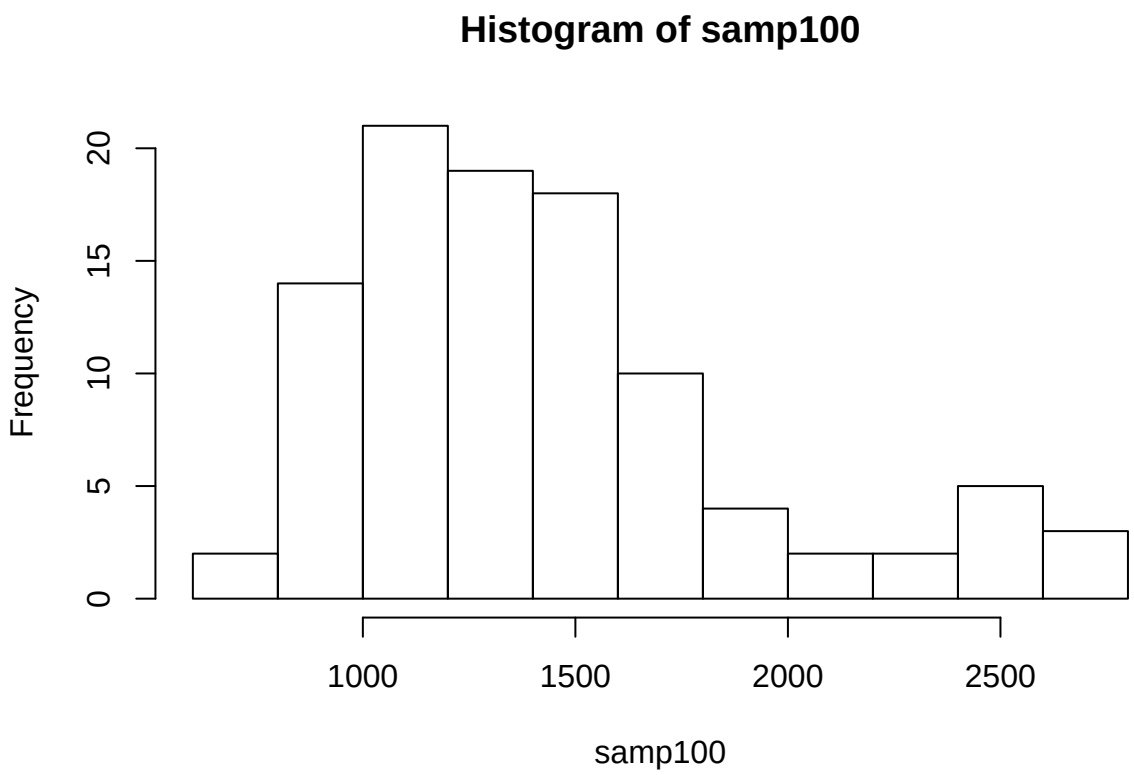
```
## Z-Score of sample 2 mean vs. the population:  -0.0274
```

Sample 2 has a mean of 1485.86, which is greater than the mean of sample 1, 1402.08.

The sample 2 mean is -0.03 standard deviations below the population mean, 1499.69, while the sample 1 mean is -0.19 standard deviations below the population mean.

Sample size = 100

```
samp100 <- sample(area, 100)
hist(samp100)
```



sample 100 statistics

```
summarys100 <- summary(samp100)
summarys100
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     780    1090    1362    1429    1625    2790
```

```r
mins100  <- as.numeric(summarys100["Min."])
means100 <- round(as.numeric(summarys100["Mean"]),2)
meds100  <- as.numeric(summarys100["Median"])
maxs100  <- as.numeric(summarys100["Max."])
iqrs100  <- as.numeric(IQR(samp100))
cat(paste("Inter-Quartile Range of sample 100: ",iqrs100,"\n"))
```

```
## Inter-Quartile Range of sample 100:  535.5
```

```r
stdevs100 <- round(as.numeric(sd(samp100)),2)
cat(paste("Standard Deviation of sample 100: ",stdevs100,"\n"))
```

```
## Standard Deviation of sample 100:  472.3
```

```r
s100medZscore = round((meds100 - medpop) / stdevpop, 4)
cat(paste("Z-Score of sample 2 median vs. the population: ",s100medZscore,"\n"))
```

```
## Z-Score of sample 2 median vs. the population:  -0.1583
```

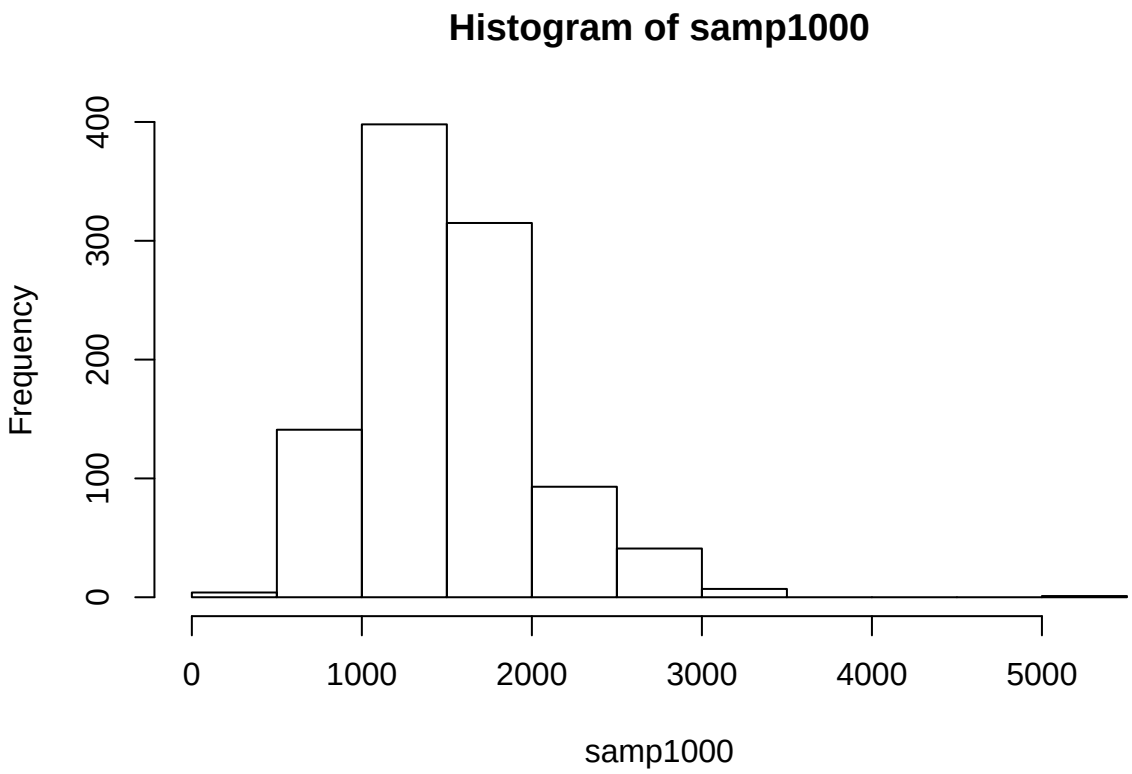
```r
s100meanZscore = round((means100 - meanpop) / stdevpop, 4)
cat(paste("Z-Score of sample 2 mean vs. the population: ",s100meanZscore,"\n"))
```

```
## Z-Score of sample 2 mean vs. the population:  -0.1401
```

The sample 100 mean, 1428.88 is -0.14 standard deviations below the population mean, 1499.69 .

Sample size = 1000

```r
samp1000 <- sample(area, 1000)
hist(samp1000)
```



sample 1000 statistics

```
summarys1000 <- summary(samp1000)
summarys1000
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334    1142    1457    1505    1759    5095
```

```
mins1000  <- as.numeric(summarys1000["Min."])
means1000 <- round(as.numeric(summarys1000["Mean"]),2)
meds1000  <- as.numeric(summarys1000["Median"])
maxs1000  <- as.numeric(summarys1000["Max."])
iqrs1000  <- as.numeric(IQR(samp1000))
cat(paste("Inter-Quartile Range of sample 1000: ",iqrs1000,"\n"))
```

```
## Inter-Quartile Range of sample 1000:  617.75
```

```
stdevs1000 <- round(as.numeric(sd(samp1000)),2)
cat(paste("Standard Deviation of sample 1000: ",stdevs1000,"\n"))
```

```
## Standard Deviation of sample 1000:  502.05
```

```
s1000medZscore = round((meds1000 - medpop) / stdevpop, 4)
cat(paste("Z-Score of sample 2 median vs. the population: ",s1000medZscore,"\n"))
```

```
## Z-Score of sample 2 median vs. the population:  0.0297
```

```
s1000meanZscore = round((means1000 - meanpop) / stdevpop, 4)
cat(paste("Z-Score of sample 2 mean vs. the population: ",s1000meanZscore,"\n"))
```

```
## Z-Score of sample 2 mean vs. the population:  0.0115
```

The sample 1000 mean, 1505.49 is 0.01 standard deviations above the population mean, 1499.69 .

Summary of samples of size 50, 100, 1000

```
SampleSummary <- data.frame(matrix(
  c(50,means1,s1meanZscore,
    50,means2,s2meanZscore,
    100,means100,s100meanZscore,
    1000,means1000,s1000meanZscore),
  nrow = 4,ncol = 3, byrow = T,
  dimnames = list(NULL,c("SampleSize","SampleMean","ZScore"))
  )
  )
SampleSummary %>%
  kable() %>%
  kable_styling()
```

SampleSize	SampleMean	ZScore
50	1402	-0.19
50	1486	-0.03
100	1429	-0.14
1000	1505	0.01

We would expect the larger sample size to have the closest sample mean to the population mean (1499.69) Because of the variation among random samples, this doesn't always happen in the case of individual samples. As the Standard Error of the mean is

$$SE = \frac{\sigma}{\sqrt{N}}$$

where sigma is the standard deviation of the population (i.e., sd(area) = 505.51) and N represents the sample size, quadrupling N causes the standard error of the mean to be cut in half.

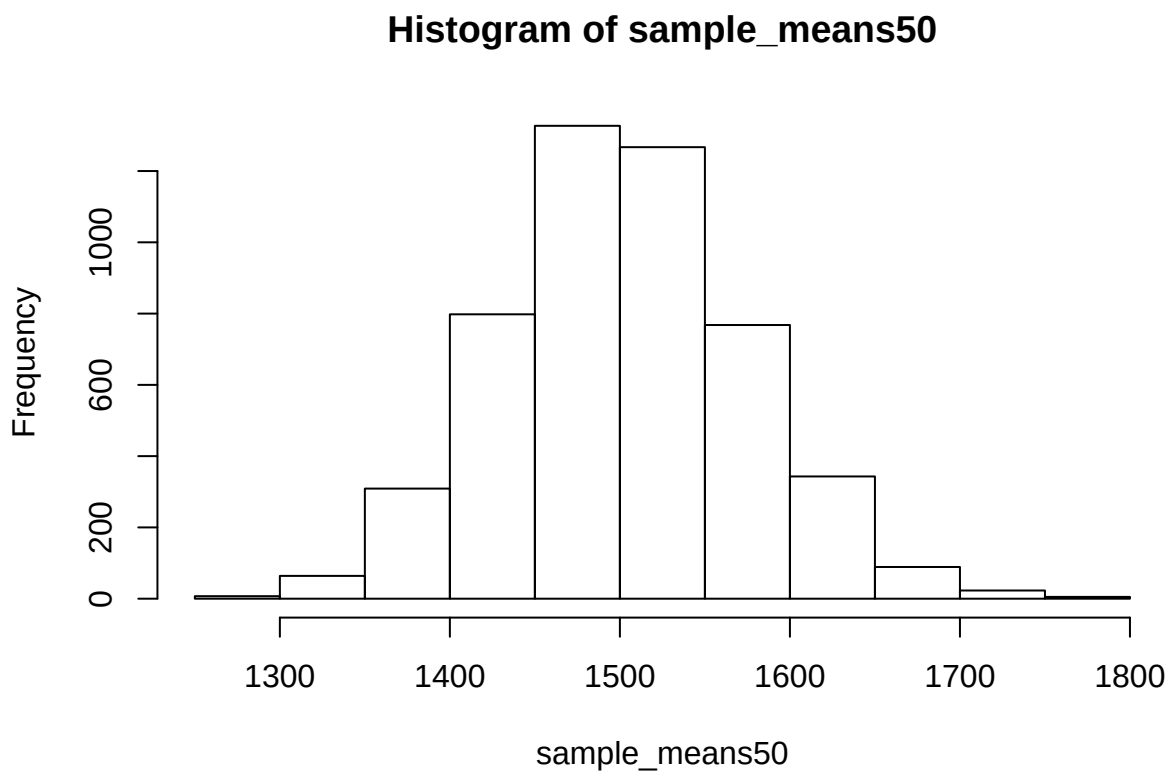
End of response to Exercise 3 .

.#.

Not surprisingly, every time we take another random sample, we get a different sample mean. It's useful to get a sense of just how much variability we should expect when estimating the population mean this way. The distribution of sample means, called the *sampling distribution*, can help us understand this variability. In this lab, because we have access to the population, we can build up the sampling distribution for the sample

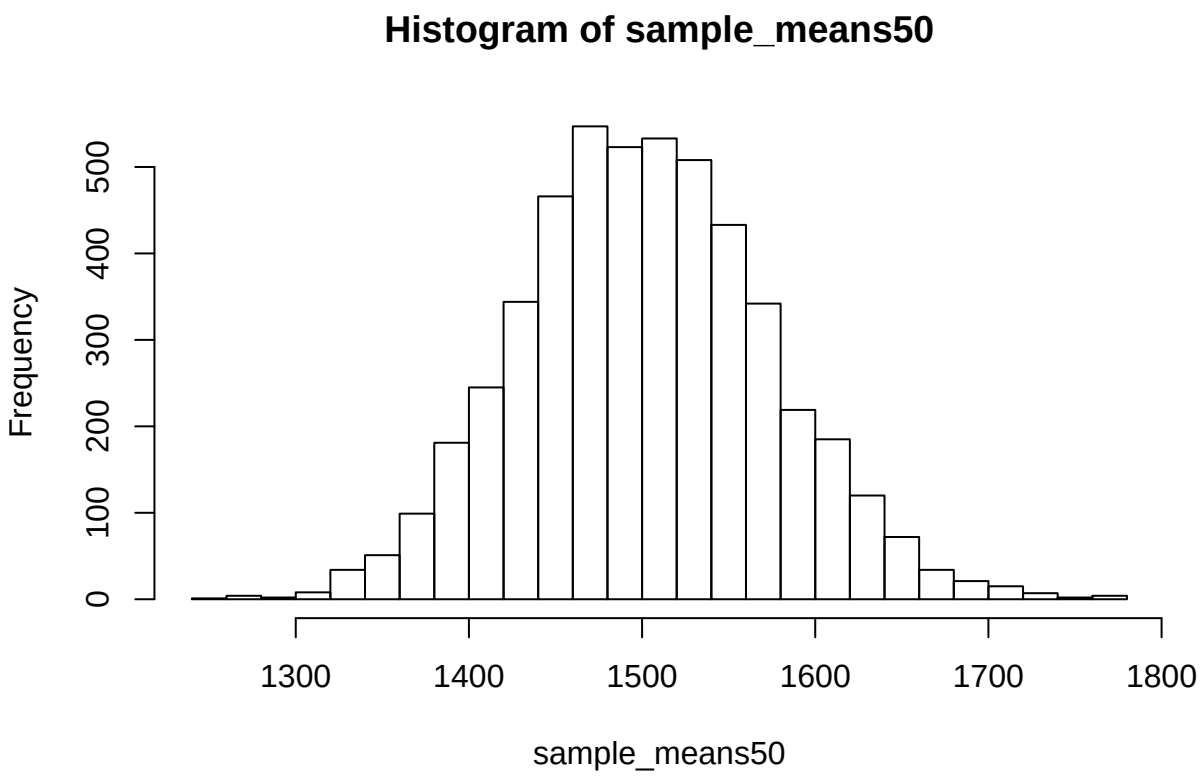
mean by repeating the above steps many times. Here we will generate 5000 samples and compute the sample mean of each.

```
sample_means50 <- rep(NA, 5000)

for(i in 1:5000){
   samp <- sample(area, 50)
   sample_means50[i] <- mean(samp)
   }

hist(sample_means50)
```



If you would like to adjust the bin width of your histogram to show a little more detail, you can do so by changing the `breaks` argument.

```
hist(sample_means50, breaks = 25)
```



Here we use R to take 5000 samples of size 50 from the population, calculate the mean of each sample, and store each result in a vector called `sample_means50`. On the next page, we'll review how this set of code works.

4. How many elements are there in `sample_means50`? Describe the sampling distribution, and be sure to specifically note its center. Would you expect the distribution to change if we instead collected 50,000 sample means?

```
summary_sample_means50 <- summary(sample_means50)
summary_sample_means50
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1251    1453    1500    1502    1549    1771
```

```
stdev_summary_sample_means50 <- sd(sample_means50)
stdev_summary_sample_means50
```

```
## [1] 72
```

```
min_sample_means50  <- as.numeric(summary_sample_means50["Min."])
mean_sample_means50 <- round(as.numeric(summary_sample_means50["Mean"]),2)
med_sample_means50  <- as.numeric(summary_sample_means50["Median"])
max_sample_means50  <- as.numeric(summary_sample_means50["Max."])
iqr_sample_means50  <- round(as.numeric(IQR(sample_means50)),2)
cat(paste("Inter-Quartile Range of Mean of Sampling Distribution (5000 draws, each of size 50): ",iqr_s
```

```
## Inter-Quartile Range of Mean of Sampling Distribution (5000 draws, each of size 50):  95.4
```

```
stdev_sample_means50 <- round(as.numeric(sd(sample_means50)),2)
cat(paste("Standard Deviation of Mean of Sampling Distribution (5000 draws, each of size 50): ", stdev_s
```

```
## Standard Deviation of Mean of Sampling Distribution (5000 draws, each of size 50):  71.95
```

```
theoretical_stdev_sample_means50 <- round(sd(area) / sqrt(50),2)
cat(paste("Theoretical Standard Error of Mean of samples of size 50: ", theoretical_stdev_sample_means50
```

```
## Theoretical Standard Error of Mean of samples of size 50:  71.49
```

There are 5000 elements in `sample_means50` .

The sampling distribution of the sample mean is a symmetric unimodal distribution which appears to be Normal.

The sampling distribution of the sample mean has an IQR of 95.4.

The sampling distribution of the sample mean has a StandardDeviation of 71.95, while the Standard Deviation of the entire population is 505.51 .

The theoretical standard deviation of the sampling distribution of the sample mean, or the Standard Error of the sample mean, is 71.49, which is quite close to that actually observed (71.95) on this set of samples, where each sample is of size 50:

$$SE_{\bar{x}} = \frac{\sigma_{pop}}{\sqrt{N}} = \frac{505.51}{\sqrt{50}} = \frac{505.51}{0.7071} = 71.49$$

The sampling distribution of the sample mean has a median of 1499.92, which is greater than the population median, 1442.

The sampling distribution of the sample mean has a mean of 1501.61, which is greater than the population mean, 1499.69.

The mean of the sampling distribution of the sample mean , 1501.61, is greater than the median of the sampling distribution of the sample mean , 1499.92, which suggests a right skew.

NB: The above results will change each time the sampling is rerun (which includes each time the file is re-knit.)

The population mean, 1499.69, is greater than the population median, 1442, which indicates a right skew on the population.

Would you expect the distribution to change if we instead collected 50,000 sample means?

No, if we increase the number of simulations to 50,000 (while keeping the size of each sample at 50) we do NOT expect a significant change in the sampling distribution of the mean.

This is because:

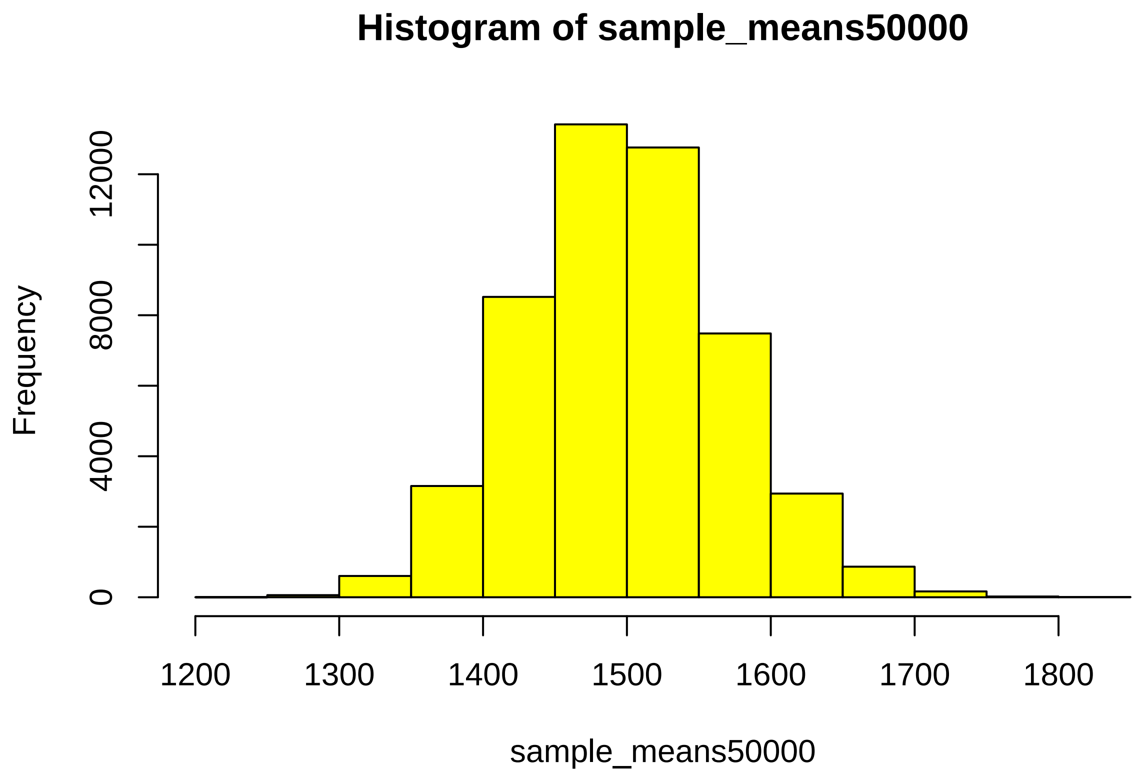
the sample mean is an unbiased estimator,

the sampling distribution is centered at the true average of the the population, and

the spread of the distribution indicates how much variability is induced by sampling only 50 home sales.

Checking:

```
sample_means50000 <- rep(NA, 50000)

for(i in 1:50000){
   samp <- sample(area, 50)
   sample_means50000[i] <- mean(samp)
   }

hist(sample_means50000,col="yellow")
```




```r
summary(sample_means50000)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1244    1451    1497    1499    1545    1842
```

```r
sd(sample_means50000)
```

```
## [1] 71
```

Above we observe that the distribution under 50,000 samples is quite similar to that under 5,000 samples. As long as the number of samples is not extremely small, we will not see any appreciable change. (On the other hand, if SIZE of each sample were changed from the present 50, then we would expect to see a change to the spread of the distribution, though the mean would remain the same.)

End of response to Exercise 4 .

.#.

Interlude: The `for` loop

Let's take a break from the statistics for a moment to let that last block of code sink in. You have just run your first `for` loop, a cornerstone of computer programming. The idea behind the for loop is *iteration*: it allows you to execute code as many times as you want without having to type out every iteration. In the

case above, we wanted to iterate the two lines of code inside the curly braces that take a random sample of size 50 from `area` then save the mean of that sample into the `sample_means50` vector. Without the `for` loop, this would be painful:

```
sample_means50 <- rep(NA, 5000)

samp <- sample(area, 50)
sample_means50[1] <- mean(samp)

samp <- sample(area, 50)
sample_means50[2] <- mean(samp)

samp <- sample(area, 50)
sample_means50[3] <- mean(samp)

samp <- sample(area, 50)
sample_means50[4] <- mean(samp)
```

and so on...

With the for loop, these thousands of lines of code are compressed into a handful of lines. We've added one extra line to the code below, which prints the variable `i` during each iteration of the `for` loop. Run this code.

```
sample_means50 <- rep(NA, 5000)

for(i in 1:5000){
   samp <- sample(area, 50)
   sample_means50[i] <- mean(samp)
   ## printing suppressed for final knitting
   #######(print(i))
   }
```

Let's consider this code line by line to figure out what it does. In the first line we *initialized a vector*. In this case, we created a vector of 5000 zeros called `sample_means50`. This vector will will store values generated within the `for` loop.

The second line calls the `for` loop itself. The syntax can be loosely read as, "for every element `i` from 1 to 5000, run the following lines of code". You can think of `i` as the counter that keeps track of which loop you're on. Therefore, more precisely, the loop will run once when `i = 1`, then once when `i = 2`, and so on up to `i = 5000`.

The body of the `for` loop is the part inside the curly braces, and this set of code is run for each value of `i`. Here, on every loop, we take a random sample of size 50 from `area`, take its mean, and store it as the ith element of `sample_means50`.

In order to display that this is really happening, we asked R to print `i` at each iteration. This line of code is optional and is only used for displaying what's going on while the `for` loop is running.

The `for` loop allows us to not just run the code 5000 times, but to neatly package the results, element by element, into the empty vector that we initialized at the outset.

5. To make sure you understand what you've done in this loop, try running a smaller version. Initialize a vector of 100 zeros called `sample_means_small`. Run a loop that takes a sample of size 50 from `area` and stores the sample mean in `sample_means_small`, but only iterate from 1 to 100. Print the output to your screen (type `sample_means_small` into the console and press enter). How many elements are there in this object called `sample_means_small`? What does each element represent?

```
sample_means_small <- rep(0, 100)
```

```
for(i in 1:100){
   samp <- sample(area, 50)
   sample_means_small[i] <- mean(samp)
   ## printing suppressed for final knitting
   #######(print(i))
}
sample_means_small
```

```
##   [1] 1483 1521 1500 1556 1617 1616 1499 1533 1400 1533 1580 1552 1375 1637
##  [15] 1580 1502 1617 1515 1487 1583 1494 1484 1625 1506 1532 1364 1750 1487
##  [29] 1364 1478 1516 1559 1470 1588 1505 1482 1409 1593 1419 1456 1362 1514
##  [43] 1396 1358 1404 1630 1404 1446 1525 1540 1484 1510 1579 1457 1361 1501
##  [57] 1502 1411 1498 1451 1390 1475 1531 1536 1371 1499 1544 1487 1465 1388
##  [71] 1489 1498 1561 1572 1609 1444 1404 1502 1602 1447 1622 1457 1395 1474
##  [85] 1476 1459 1563 1460 1431 1552 1536 1464 1327 1501 1536 1440 1505 1607
##  [99] 1468 1446
```

How many elements are there in this object called `sample_means_small`?

There are 100 elements in the object `sample_means_small` .

What does each element represent?

Each element represents the average square footage from a sample of 50 houses randomly selected from the population (2930 houses in Ames, Iowa).

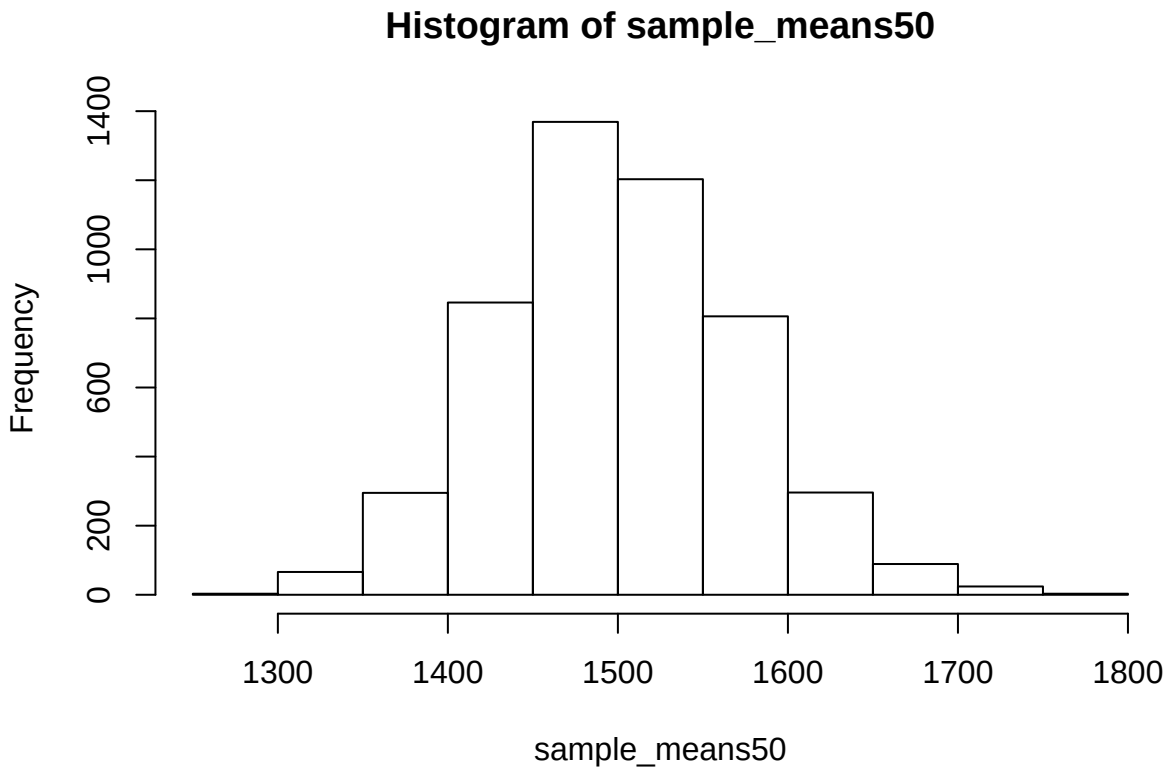
End of response to Exercise 5 .

.#.

Sample size and the sampling distribution

Mechanics aside, let's return to the reason we used a `for` loop: to compute a sampling distribution, specifically, this one.

```
hist(sample_means50)
```



The sampling distribution that we computed tells us much about estimating the average living area in homes in Ames. Because the sample mean is an unbiased estimator, the sampling distribution is centered at the true average living area of the the population, and the spread of the distribution indicates how much variability is induced by sampling only 50 home sales.

To get a sense of the effect that sample size has on our distribution, let's build up two more sampling distributions: one based on a sample size of 10 and another based on a sample size of 100.

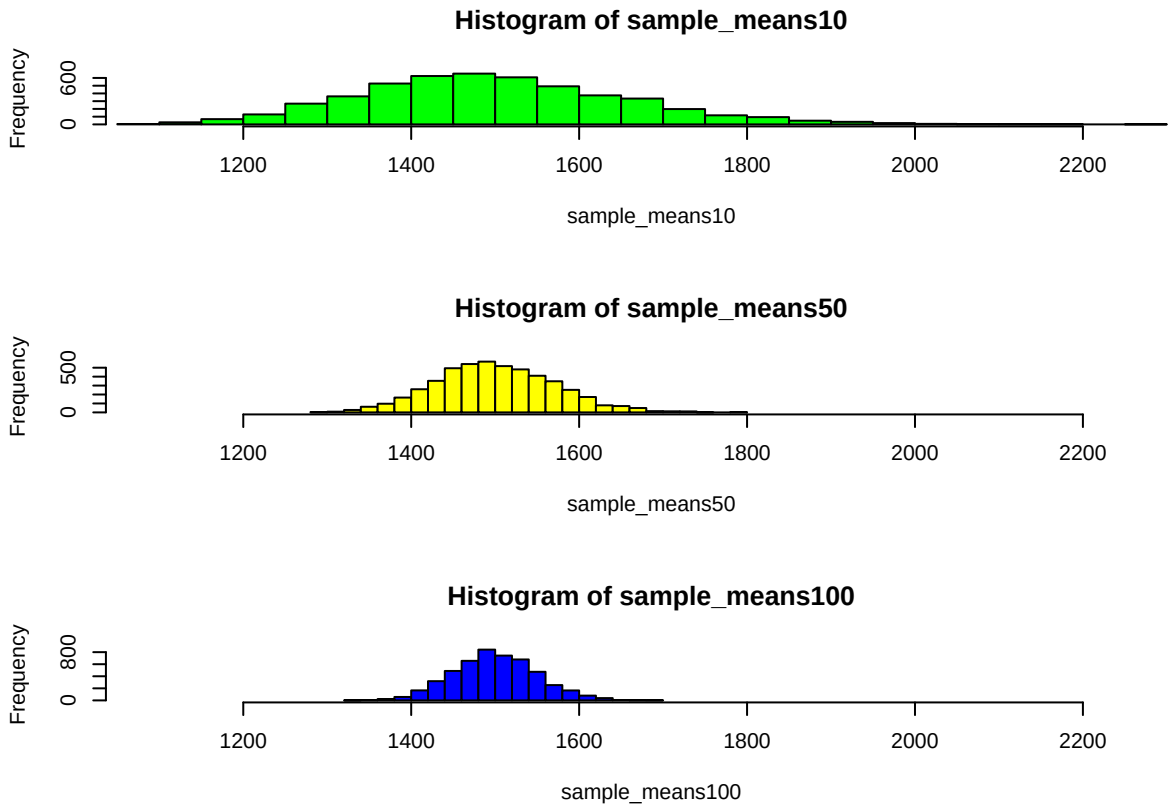
```
sample_means10 <- rep(NA, 5000)
sample_means100 <- rep(NA, 5000)

for(i in 1:5000){
  samp <- sample(area, 10)
  sample_means10[i] <- mean(samp)
  samp <- sample(area, 100)
  sample_means100[i] <- mean(samp)
}
```

Here we're able to use a single `for` loop to build two distributions by adding additional lines inside the curly braces. Don't worry about the fact that `samp` is used for the name of two different objects. In the second command of the `for` loop, the mean of `samp` is saved to the relevant place in the vector `sample_means10`. With the mean saved, we're now free to overwrite the object `samp` with a new sample, this time of size 100. In general, anytime you create an object using a name that is already in use, the old object will get replaced with the new one.

To see the effect that different sample sizes have on the sampling distribution, plot the three distributions on

top of one another.

```
par(mfrow = c(3, 1))

xlimits <- range(sample_means10)

hist(sample_means10, breaks = 20, xlim = xlimits, col="green")
hist(sample_means50, breaks = 20, xlim = xlimits, col="yellow")
hist(sample_means100, breaks = 20, xlim = xlimits, col="blue")
```



The first command specifies that you'd like to divide the plotting area into 3 rows and 1 column of plots (to return to the default setting of plotting one at a time, use `par(mfrow = c(1, 1))`). The `breaks` argument specifies the number of bins used in constructing the histogram. The `xlim` argument specifies the range of the x-axis of the histogram, and by setting it equal to `xlimits` for each histogram, we ensure that all three histograms will be plotted with the same limits on the x-axis.

6. When the sample size is larger, what happens to the center? What about the spread?

```
summary_sample_means10 <- summary(sample_means10)
summary_sample_means10
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1084    1387    1488    1499    1598    2288
```

```
stdev_summary_sample_means10 <- sd(sample_means10)
stdev_summary_sample_means10
```

```
## [1] 160
```

```
min_sample_means10  <- as.numeric(summary_sample_means10["Min."])
mean_sample_means10 <- round(as.numeric(summary_sample_means10["Mean"]),2)
med_sample_means10  <- as.numeric(summary_sample_means10["Median"])
max_sample_means10  <- as.numeric(summary_sample_means10["Max."])
iqr_sample_means10  <- round(as.numeric(IQR(sample_means10)),2)
cat(paste("Inter-Quartile Range of Mean of Sampling Distribution (5000 draws, each of size 10): ",iqr_sa
```

```
## Inter-Quartile Range of Mean of Sampling Distribution (5000 draws, each of size 10):  211.35
```

```
stdev_sample_means10 <- round(as.numeric(sd(sample_means10)),2)
cat(paste("Standard Deviation of Mean of Sampling Distribution (5000 draws, each of size 10): ", stdev_s
```

```
## Standard Deviation of Mean of Sampling Distribution (5000 draws, each of size 10):  160.11
```

```
theoretical_stdev_sample_means10 <- round(sd(area) / sqrt(10),2)
cat(paste("Theoretical Standard Error of Mean of samples of size 10: ", theoretical_stdev_sample_means10
```

```
## Theoretical Standard Error of Mean of samples of size 10:  159.86
```

```
summary_sample_means100 <- summary(sample_means100)
summary_sample_means100
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1333    1466    1498    1500    1533    1694
```

```
stdev_summary_sample_means100 <- sd(sample_means100)
stdev_summary_sample_means100
```

```
## [1] 50
```

```
min_sample_means100  <- as.numeric(summary_sample_means100["Min."])
mean_sample_means100 <- round(as.numeric(summary_sample_means100["Mean"]),2)
med_sample_means100  <- as.numeric(summary_sample_means100["Median"])
max_sample_means100  <- as.numeric(summary_sample_means100["Max."])
iqr_sample_means100  <- round(as.numeric(IQR(sample_means100)),2)
cat(paste("Inter-Quartile Range of Mean of Sampling Distribution (5000 draws, each of size 100): ",iqr_s
```

```
## Inter-Quartile Range of Mean of Sampling Distribution (5000 draws, each of size 100):  67.12
```

```
stdev_sample_means100 <- round(as.numeric(sd(sample_means100)),2)
cat(paste("Standard Deviation of Mean of Sampling Distribution (5000 draws, each of size 100): ", stdev_
```

```
## Standard Deviation of Mean of Sampling Distribution (5000 draws, each of size 100):  49.54
```

```
theoretical_stdev_sample_means100 <- round(sd(area) / sqrt(100),2)
cat(paste("Theoretical Standard Error of Mean of samples of size 100: ", theoretical_stdev_sample_means1
```

```
## Theoretical Standard Error of Mean of samples of size 100:  50.55
```

Summary of sampling distributions of sample size 10, 50, 100

```
SamplingDistributionsSummary <- data.frame(
  matrix(
  c(10,mean_sample_means10,100*(mean_sample_means10 - meanpop)/meanpop,
    stdev_sample_means10,theoretical_stdev_sample_means10,

    50,mean_sample_means50,100*(mean_sample_means50 - meanpop)/meanpop,
    stdev_sample_means50,theoretical_stdev_sample_means50,
```

```
    100,mean_sample_means100,100*(mean_sample_means100 - meanpop)/meanpop,
    stdev_sample_means100,theoretical_stdev_sample_means100),

    nrow = 3,ncol = 5, byrow = T,
  dimnames = list(NULL,c("SampleSize","Mean of 5,000 Samples","Pct.Error","Actual Std Dev of sampling di

  )
SamplingDistributionsSummary %>%
  kable() %>%
  kable_styling()
```

SampleSize	Mean.of.5.000.Samples	Pct.Error	Actual.Std.Dev.of.sampling.dist	Theoretical.Std.Error.of.the.Mean
10	1499	-0.07	160	160
50	1502	0.13	72	71
100	1500	0.01	50	51

As the sample size increases, the center of the sampling distribution becomes a more reliable estimate for the true population mean (here, 1499.69), as shown in the table above.

When the sample size is larger, the center converges closer to the population mean. Specifically, the increase in the sample size causes the standard deviation (a measure of dispersion) to narrow in proportion to the reciprocal of the square root of the sample size, in accordance with the rule

$$SE_{\bar{x}} = \frac{\sigma_{pop}}{\sqrt{N}}$$

End of response to Exercise 6 .

.#.

On your own

So far, we have only focused on estimating the mean living area in homes in Ames. Now you'll try to estimate the mean home price.

(1) Take a random sample of size 50 from price. Using this sample, what is your best point estimate of the population mean?

```
samp1px <- sample(price, 50)
px_samp1mean <- round(mean(samp1px),2)
cat(paste("Sample mean of the price (n=50): ", px_samp1mean, "\n"))
```

```
## Sample mean of the price (n=50):  194014.56
```

A point estimate of the population mean is 194014.56 . As this is based upon a single sample of size 50, depending upon the "luck of the draw", the estimate may or may not be close to the actual population mean.

(2) Since you have access to the population, simulate the sampling distribution for $\bar{x}_{price}$ by taking 5000 samples from the population of size 50 and computing 5000 sample means. Store these means in a vector called `sample_means50`.

Plot the data, then describe the shape of this sampling distribution.

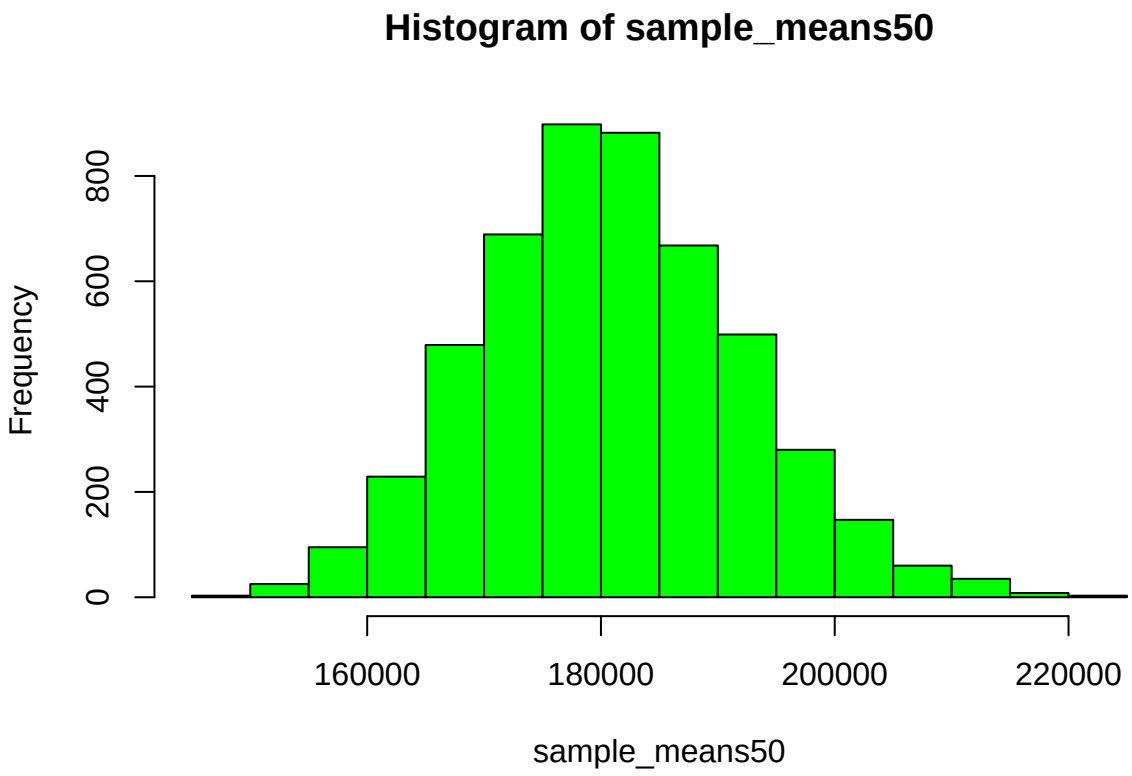
Based on this sampling distribution, what would you guess the mean home price of the population to be?

Finally, calculate and report the population mean.

```
sample_means50 <- rep(NA, 5000)

for(i in 1:5000){
   samp <- sample(price, 50)
   sample_means50[i] <- mean(samp)
   }
#summary(sample_means50)
#sd(sample_means50)
```

Plot the data **:**

```
hist(sample_means50,breaks=25,col="green")
```



Describe the shape of this sampling distribution:

The sampling distribution of the sample mean of Price is a symmetric unimodal distribution which appears to be Normal. The distribution appears to be centered around 180,000.

```
px_summary_sample_means50 <- summary(sample_means50)
px_summary_sample_means50
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  148843  173234  180474  180902  188144  224494
```

```
px_stdev_summary_sample_means50 <- sd(sample_means50)
px_stdev_summary_sample_means50
```

```
## [1] 11162
```

```
px_min_sample_means50  <- as.numeric(px_summary_sample_means50["Min."])
px_mean_sample_means50 <- round(as.numeric(px_summary_sample_means50["Mean"]),2)
cat(paste("\n\n**Sample mean of the distribution** of 5,000 samples (n=50) of Price: ", px_mean_sample_
```

```
##
##
## **Sample mean of the distribution** of 5,000 samples (n=50) of Price:  180902.45
```

```
px_med_sample_means50  <- as.numeric(px_summary_sample_means50["Median"])
px_max_sample_means50  <- as.numeric(px_summary_sample_means50["Max."])
px_iqr_sample_means50  <- round(as.numeric(IQR(sample_means50)),2)
cat(paste("Inter-Quartile Range of Mean Price of Sampling Distribution (5000 draws, each of size 50): "
```

```
## Inter-Quartile Range of Mean Price of Sampling Distribution (5000 draws, each of size 50):  14910.2
```

```
px_stdev_sample_means50 <- round(as.numeric(sd(sample_means50)),2)
cat(paste("Standard Deviation of Mean Price of Sampling Distribution (5000 draws, each of size 50): ", p
```

```
## Standard Deviation of Mean Price of Sampling Distribution (5000 draws, each of size 50):  11162.35
```

```
px_theoretical_stdev_sample_means50 <- round(sd(price) / sqrt(50),2)
cat(paste("Theoretical Standard Error of Mean Price of samples of size 50: ", px_theoretical_stdev_samp
```

```
## Theoretical Standard Error of Mean Price of samples of size 50:  11297.68
```

Based on this sampling distribution, what would you guess the mean home price of the population to be?

The sampling distribution of the sample mean of Price has a mean of 180902.45 . Thus, I would guess the mean home price of the population to be 180902.45 .

Population statistics:

```
px_summarypop <- summary(price)
px_summarypop
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12789  129500  160000  180796  213500  755000
```

```
px_minpop  <- as.numeric(px_summarypop["Min."])

px_meanpop <- round(as.numeric(px_summarypop["Mean"]),2)
cat(paste("\n\nActual mean of the price (pop.): ", px_meanpop, "\n\n"))
```

```
##
##
## Actual mean of the price (pop.):  180796.06
```

```
px_errormean <- round(100*(px_samp1mean - px_meanpop)/px_meanpop,2)
cat(paste("\n\nPercent error of the sample mean (single sample) vs. actual mean (price): ", px_errormea
```

```
##
##
## Percent error of the sample mean (single sample) vs. actual mean (price):  7.31 %.
```

```
px_medpop  <- as.numeric(px_summarypop["Median"])
px_maxpop  <- as.numeric(px_summarypop["Max."])
px_iqrpop  <- as.numeric(IQR(price))
cat(paste("Inter-Quartile Range of the population of Price: ",px_iqrpop,"\n"))
```

```
## Inter-Quartile Range of the population of Price:  84000
```

```
px_stdevpop <- round(as.numeric(sd(price)),2)
cat(paste("Standard Deviation of the population: ",px_stdevpop,"\n"))
```

```
## Standard Deviation of the population:  79886.69
```

The population mean, 180796.06, is greater than the population median, 160000, which indicates a right skew on the population.

The sampling distribution of the sample mean of Price has a StandardDeviation of 11162.35, while the Standard Deviation of the entire population of Price is 79886.69 .

The theoretical standard deviation of the sampling distribution of the sample mean of Price, or the Standard Error of the sample mean (of Price), is 11297.68, which is quite close to that actually observed (11162.35) on this set of samples, where each sample is of size 50:

$$SE_{\bar{px}} = \frac{\sigma_{px_{pop}}}{\sqrt{N}} = \frac{79886.69}{\sqrt{50}} = \frac{79886.69}{7.071} = 11297.68$$

The sampling distribution of the sample mean of Price has a median of 180474.18, which is greater than the population median, 160000.

The sampling distribution of the sample mean has a mean of 180902.45, which is greater than the population mean, 180796.06.

(3) Change your sample size from 50 to 150, then compute the sampling distribution using the same method as above, and store these means in a new vector called `sample_means150`**.**

```
sample_means150 <- rep(NA, 5000)

for(i in 1:5000){
   samp <- sample(price, 150)
   sample_means150[i] <- mean(samp)
   }
summary(sample_means150)
```

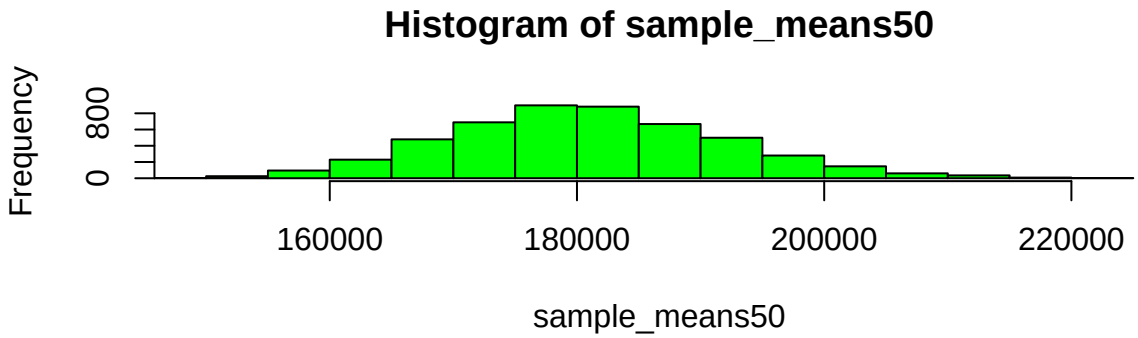
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  159192  176197  180723  180860  185330  207064
```

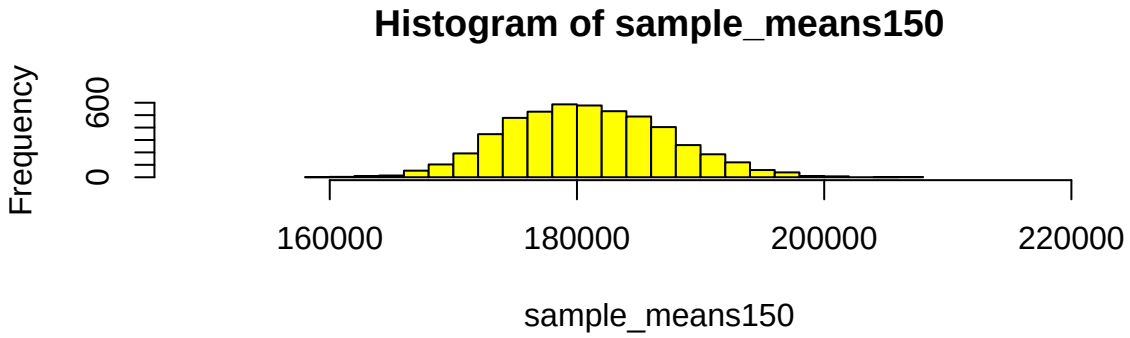
```
sd(sample_means150)
```

```
## [1] 6435
```

Plot the data - compare vs. sample size 50:

```
par(mfrow = c(2, 1))

xlimits <- range(sample_means50)

hist(sample_means50, breaks = 20, xlim = xlimits, col="green")
hist(sample_means150, breaks = 20, xlim = xlimits, col="yellow")
```






```
invisible(par(new))
```

```
px_summary_sample_means150 <- summary(sample_means150)
px_summary_sample_means150
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  159192  176197  180723  180860  185330  207064
```

```
px_stdev_summary_sample_means150 <- sd(sample_means150)
px_stdev_summary_sample_means150
```

```
## [1] 6435
```

```
px_min_sample_means150  <- as.numeric(px_summary_sample_means150["Min."])
px_mean_sample_means150 <- round(as.numeric(px_summary_sample_means150["Mean"]),2)
cat(paste("\n\n**Sample mean of the distribution** of 5,000 samples (n=150) of Price: ", px_mean_sample_
```

```
##
##
## **Sample mean of the distribution** of 5,000 samples (n=150) of Price:  180859.67
```

```
px_med_sample_means150  <- as.numeric(px_summary_sample_means150["Median"])
px_max_sample_means150  <- as.numeric(px_summary_sample_means150["Max."])
px_iqr_sample_means150  <- round(as.numeric(IQR(sample_means150)),2)
cat(paste("Inter-Quartile Range of Mean Price of Sampling Distribution (5,000 draws, each of size 150):
```

```
## Inter-Quartile Range of Mean Price of Sampling Distribution (5,000 draws, each of size 150):  9132.83
```

```r
px_stdev_sample_means150 <- round(as.numeric(sd(sample_means150)),2)
cat(paste("Standard Deviation of Mean Price of Sampling Distribution (5,000 draws, each of size 150): "
```

```
## Standard Deviation of Mean Price of Sampling Distribution (5,000 draws, each of size 150):  6434.81
```

```r
px_theoretical_stdev_sample_means150 <- round(sd(price) / sqrt(150),2)
cat(paste("Theoretical Standard Error of Mean Price of samples of size 150: ", px_theoretical_stdev_sam
```

```
## Theoretical Standard Error of Mean Price of samples of size 150:  6522.72
```

Describe the shape of this sampling distribution, and compare it to the sampling distribution for a sample size of 50.

The sampling distribution of the sample mean of Price with sample size = 150 is a symmetric unimodal distribution which appears to be Normal. Like the distribution with sample size = 50, the distribution also appears to be centered around 180,000. The actual values for the mean of the sample means are nearly the same, with the sample mean of price under samples of size 150 equal to 180859.67 and the sample mean under sample size 50 equal to 180902.45 . However, the distribution is "tighter" under the larger sample size (150) when compared vs. the smaller sample size (50).

Based on this sampling distribution, what would you guess to be the mean sale price of homes in Ames?

Under this distribution, the mean sale price of homes in Ames is estimated to be 180859.67 .

(4) Of the sampling distributions from 2 and 3, which has a smaller spread?

The sampling distribution with sample size 150 has a smaller spread than the sampling distribution with sample size 50. Specifically, the standard error of the mean for the sample size = 50 is 11162.35 while for sample size = 150, the standard error of the mean is 6434.81 .

If we're concerned with making estimates that are more often close to the true value, would we prefer a distribution with a large or small spread?

We would prefer distributions with a smaller spread in order to obtain estimates that are closer to the true value.

Here the population mean is 180796.06.

The estimate based upon samples of size 150, 180859.67, is closer to the actual mean

than the estimate based upon samples of size 50, 180902.45 .

Summary of sampling distributions of sample size 50, 150 :

```
SamplingDistributionsSummaryPrice <- data.frame(
  matrix(
  c(50,px_mean_sample_means50,px_mean_sample_means50-px_meanpop,
    px_stdev_sample_means50,px_theoretical_stdev_sample_means50,

    150,px_mean_sample_means150,px_mean_sample_means150-px_meanpop,
    px_stdev_sample_means150,px_theoretical_stdev_sample_means150),

    nrow = 2,ncol = 5, byrow = T,
  dimnames = list(NULL,c("SampleSize","Mean Price 5000 Samples","Error","Actual Std Dev of Sampling Dis

  )
SamplingDistributionsSummaryPrice %>%
  kable() %>%
  kable_styling()
```

SampleSize	Mean.Price.5000.Samples	Error	Actual.Std.Dev.of.Sampling.Dist	Theoretical.Std.Error.of.the.Mean
50	180902	106	11162	11298
150	180860	64	6435	6523