# pa-lab-6

Pradeep Paladugula

3/23/2020

## Sampling from Ames, Iowa

#If you have access to data on an entire population, say the size of # every house in Ames, Iowa, it's straight forward to answer questions # like, "How big is the typical house in Ames?" and "How much variation is # there in sizes of houses?". If you have access to only a sample of the #population, as is often the case, the task becomes more complicated.

## Q: What is your best guess for the typical size if you only know the sizes

## of several dozen houses? This sort of situation requires that you use

#your sample to make inference on what your population looks like.

## Q: How much variation is # there in sizes of houses?"

## Q What is your confidence in your estimate? WHat is the uncertainty of the estimate??

## What is you confidence that it good sample estimate of the true population mean!!!

#The data

#In the previous lab, ``Sampling Distributions'', we looked at the population # data of houses from Ames, Iowa. Let's start by loading that data set.

```
download.file("http://www.openintro.org/stat/data/ames.RData", destfile =
"ames.RData")
load("ames.RData")
names(ames)
```

```
##  [1] "Order"           "PID"             "MS.SubClass"     "MS.Zoning"
##  [5] "Lot.Frontage"    "Lot.Area"        "Street"          "Alley"
##  [9] "Lot.Shape"       "Land.Contour"    "Utilities"       "Lot.Config"
## [13] "Land.Slope"      "Neighborhood"    "Condition.1"     "Condition.2"
## [17] "Bldg.Type"       "House.Style"     "Overall.Qual"    "Overall.Cond"
## [21] "Year.Built"      "Year.Remod.Add"  "Roof.Style"      "Roof.Matl"
## [25] "Exterior.1st"    "Exterior.2nd"    "Mas.Vnr.Type"    "Mas.Vnr.Area"
## [29] "Exter.Qual"      "Exter.Cond"      "Foundation"      "Bsmt.Qual"
## [33] "Bsmt.Cond"       "Bsmt.Exposure"   "BsmtFin.Type.1"  "BsmtFin.SF.1"
## [37] "BsmtFin.Type.2"  "BsmtFin.SF.2"    "Bsmt.Unf.SF"     "Total.Bsmt.SF"
## [41] "Heating"         "Heating.QC"      "Central.Air"     "Electrical"
## [45] "X1st.Flr.SF"     "X2nd.Flr.SF"     "Low.Qual.Fin.SF" "Gr.Liv.Area"
## [49] "Bsmt.Full.Bath"  "Bsmt.Half.Bath"  "Full.Bath"       "Half.Bath"
## [53] "Bedroom.AbvGr"   "Kitchen.AbvGr"   "Kitchen.Qual"    "TotRms.AbvGrd"
## [57] "Functional"      "Fireplaces"      "Fireplace.Qu"    "Garage.Type"
## [61] "Garage.Yr.Blt"   "Garage.Finish"   "Garage.Cars"     "Garage.Area"
## [65] "Garage.Qual"     "Garage.Cond"     "Paved.Drive"     "Wood.Deck.SF"
## [69] "Open.Porch.SF"   "Enclosed.Porch"  "X3Ssn.Porch"     "Screen.Porch"
## [73] "Pool.Area"       "Pool.QC"         "Fence"           "Misc.Feature"
## [77] "Misc.Val"        "Mo.Sold"         "Yr.Sold"         "Sale.Type"
## [81] "Sale.Condition"  "SalePrice"
```

#In this lab we'll start with a simple random sample of size 60 from the #population. Specifically, this is a simple random sample of size 60. #Note that the data set has information on many housing variables, but for #the first portion of the lab we'll focus on the size of the house, #represented by the variable Gr.Liv.Area .

```
population <- ames$Gr.Liv.Area
samp <- sample(population, 60)
mean(samp)
```

```
## [1] 1504.217
```

## 1.Describe the distribution of your sample.

## What would you say is the "typical" size within your sample?

## Also state precisely what you interpreted "typical" to mean.

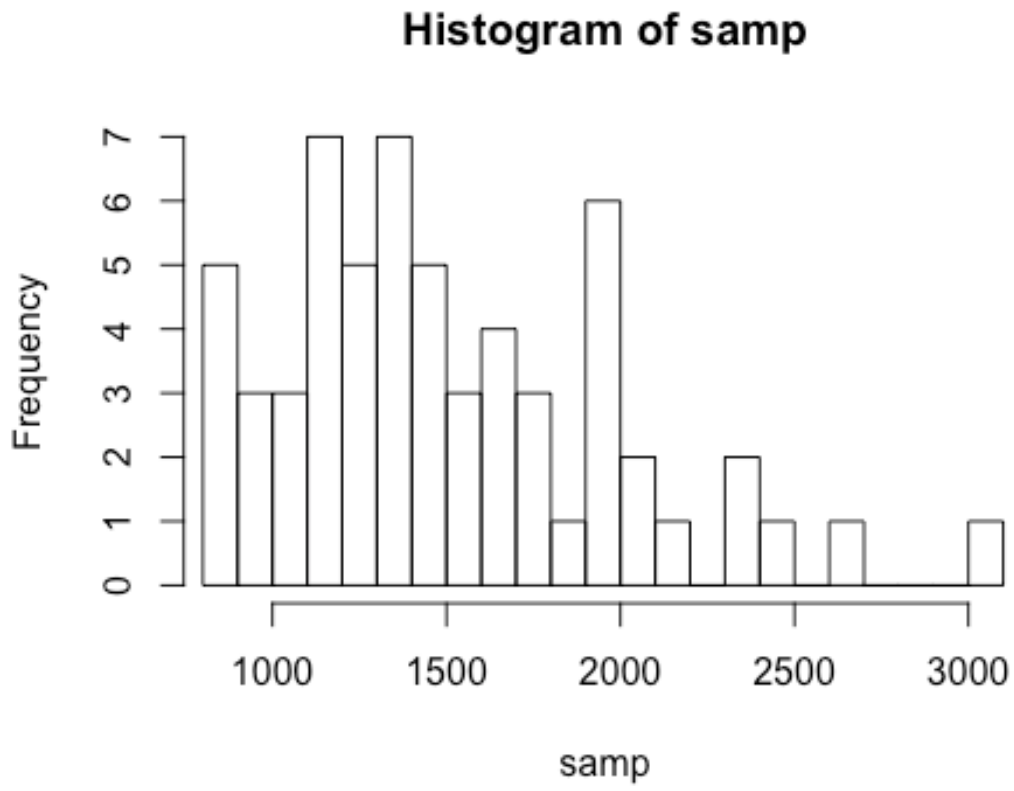
```
summary(population)
```

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

```
summary(samp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     835    1154    1402    1504    1808    3078
```

```r
xlimits <- range(samp) # limiting the values of the x axis
hist(samp, breaks = 20, xlim = xlimits)
```



## 2.Would you expect another student's distribution to be identical to yours?

## Would you expect it to be similar? Why or why not?

## Confidence intervals

## One of the most common ways to describe the typical or central value of a

## distribution is to use the mean. In this case we can calculate the mean of

## the sample using,

```
sample_mean <- mean(samp)
```

**Return for a moment to the question that first motivated this lab:**

**based on this sample, what can we infer about the population?**

**Based only on this single sample, the best estimate of the average living**

**area of houses sold in Ames would be the sample mean, usually denoted**

**as X-bar (here we're calling it sample_mean ). That serves as a**

**good point estimate but it would be useful to also communicate how**

**uncertain we are of that estimate.**

**This can be captured by using a CONFIDENCE INTERVAL.**

**We can calculate a 95% confidence interval for a sample mean by adding and**

**subtracting 1.96 standard errors to the point estimate**

**(See Section 4.2.3 if you are unfamiliar with this formula).**

**Also look at Pg 175**

```
se <- sd(samp) / sqrt(60)
lower <- sample_mean - 1.96 * se
upper <- sample_mean + 1.96 * se
c(lower, upper)
```

```
## [1] 1382.857 1625.576
```

**This is an important inference that we've just made:**

**even though we don't know what the full population looks like,**

**we're 95% confident that the true average size of houses in Ames**

**lies between the values lower and upper.**

**There are a few conditions that must be met for this interval to be valid.**

**3.For the confidence interval to be valid,**

**a) the sample mean must be normally distributed and**

**b) have standard error std/sqrt{n}.**

**********
*

## What conditions must be met for this to be true?

#` obs must be independent -> if sampling is random & <10% of population # Sample Size & Skewness: The population distribution is normal possible # If not, the sample size should be large enough (>30) to CLT to apply and assume normality.
#**********

## Confidence levels

## 4.What does “95% confidence” mean? If you’re not sure, see Section 4.2.2.

## In this case we have the luxury of knowing the true population mean since

## we have data on the entire population.

## This value can be calculated using the following command:

```
  mean(population)
```

```
## [1] 1499.69
```

## 5.Does your confidence interval capture the true average size of houses in Ames?

# If you are working on this lab in a classroom, does your neighbor’s # interval capture this value?

**6.Each student in your class should have gotten a slightly different confidence interval.**

**What proportion of those intervals would you expect to capture the true population mean? Why? If you are working in this lab in a classroom, collect data on the intervals created by other students in the class and calculate the proportion of intervals that capture the true population mean.**

**Using R, we're going to recreate many samples to learn more about how sample means and confidence intervals vary from one sample to another. Loops come in handy here (If you are unfamiliar with loops, review the Sampling Distribution Lab).**

#Here is the rough outline: # .Obtain a random sample. # .Calculate and store the sample's mean and standard deviation. # .Repeat steps (1) and (2) 50 times. # .Use these stored statistics to calculate many confidence intervals.

**But before we do all of this, we need to first create empty vectors where we can save the means and standard deviations that will be calculated from each sample. And while we're at it, let's also store the desired sample size as n .**

```
samp_mean <- rep(NA, 50)
samp_sd <- rep(NA, 50)
n <- 60
```

## Now we're ready for the loop where we calculate the means and standard deviations of 50 random samples.

```
for(i in 1:50){
  samp <- sample(population, n) # obtain a sample of size n = 60 from the
population
  samp_mean[i] <- mean(samp)    # save sample mean in ith element of
samp_mean
  samp_sd[i] <- sd(samp)        # save sample sd in ith element of samp_sd
}
```

#Lastly, we construct the confidence intervals.

```
lower_vector <- samp_mean - 1.96 * samp_sd / sqrt(n)
lower_vector

##  [1] 1300.517 1371.496 1422.423 1413.568 1326.518 1466.941 1447.809
1369.797
##  [9] 1406.312 1344.497 1401.710 1382.800 1403.054 1375.741 1361.121
1391.275
## [17] 1385.604 1286.331 1376.222 1418.727 1373.216 1330.526 1294.173
1398.639
## [25] 1266.927 1430.543 1431.273 1479.502 1356.074 1284.856 1271.060
1410.393
## [33] 1380.278 1377.541 1363.001 1355.166 1293.256 1363.194 1352.057
1334.012
## [41] 1321.419 1363.684 1389.146 1353.642 1325.293 1418.365 1374.638
1442.076
## [49] 1333.683 1290.212

upper_vector <- samp_mean + 1.96 * samp_sd / sqrt(n)
upper_vector

##  [1] 1565.149 1611.538 1649.710 1720.766 1556.815 1722.792 1748.624
1592.703
##  [9] 1675.755 1578.670 1632.156 1623.067 1659.846 1618.959 1628.146
1647.025
## [17] 1624.663 1505.369 1596.778 1665.006 1626.717 1562.041 1578.527
1658.794
## [25] 1554.273 1711.657 1700.527 1745.298 1599.760 1518.111 1512.740
1656.674
## [33] 1678.489 1666.925 1616.365 1598.401 1486.910 1624.706 1692.109
1605.988
## [41] 1550.581 1607.816 1648.054 1613.491 1545.874 1684.568 1566.829
1708.857
## [49] 1536.850 1506.388
```

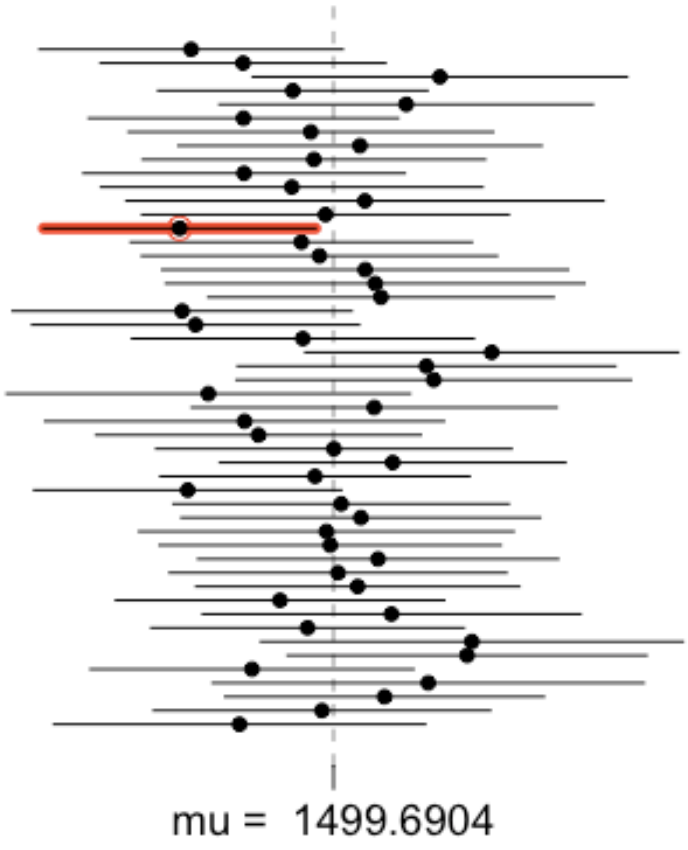
## Lower bounds of these 50 confidence intervals are stored in lower_vector,

## and the upper bounds are in upper_vector . Let's view the first interval.

```
c(lower_vector[15], upper_vector[15])
```

```
## [1] 1361.121 1628.146
```

```
par(mfrow = c(1, 1))
plot_ci(lower_vector, upper_vector, mean(population))
```




#On your own

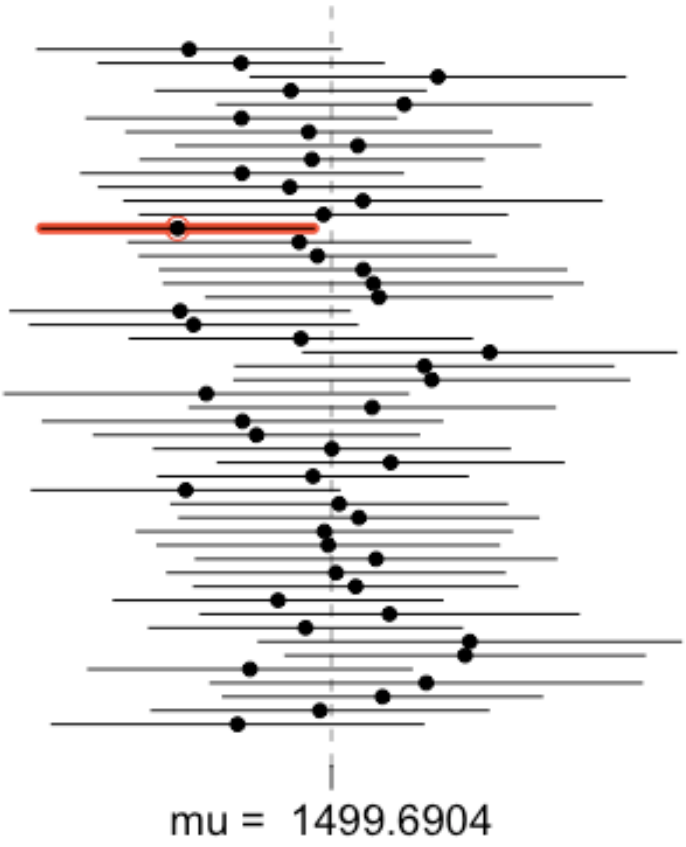
**.Using the following function (which was downloaded with the data set),**

**plot all intervals. What proportion of your confidence intervals include**

**the true population mean? Is this proportion exactly equal to the**

**confidence level? If not, explain why.**

```
par(mfrow = c(1, 1))
plot_ci(lower_vector, upper_vector, mean(population))
```



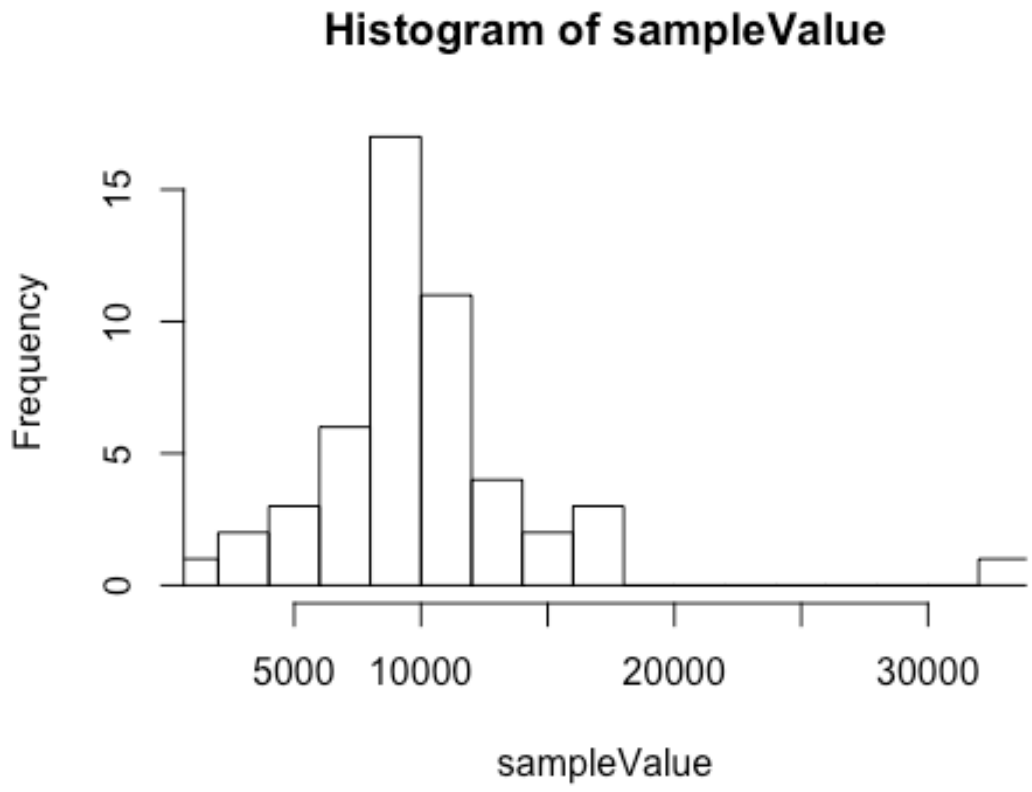
## .Pick a confidence level of your choosing, provided it is not 95%.

## What is the appropriate critical value?

```
lotArea <- ames$Lot.Area
sampleValue <- sample(lotArea, 50)
xlimits <- range(sampleValue)
critical_level <- 95
alpha <- 1-(critical_level/100)
critical_value <- 1-(alpha/2)
critical_value

## [1] 0.975

hist(sampleValue, breaks = 20, xlim = xlimits)
```

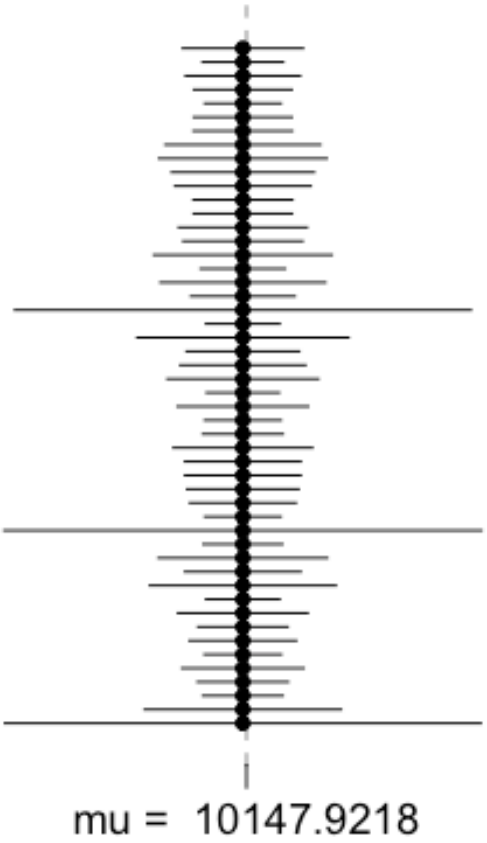



# .Calculate 50 confidence intervals at the confidence level you chose # in the previous question. You do not need to obtain new samples, # simply calculate new intervals based on the sample means and # standard deviations you have already collected. Using the plot_ci function, plot all intervals and calculate the proportion of intervals that include the true population mean. How does this percentage compare to the confidence level selected for the intervals?

```r
sampleValueMean <- mean(sampleValue)
sampleMean <- rep(NA, 50)
sampleSd <- rep(NA, 50)
n <- 60
for (i in 1:50) {
  sampVal <- sample(lotArea, n)
  sampleMean[i] <- mean(sampVal)
  sampleSd[i] <- sd(sampVal)
}
lowerVector <- sampleValueMean - 1.96 * sampleSd/sqrt(n)
upperVector <- sampleValueMean + 1.96 * sampleSd/sqrt(n)
c(lowerVector[1], upperVector[1])
```

```
## [1]  4923.313 15216.007
```

```r
par(mfrow = c(1, 1))
plot_ci(lowerVector, upperVector, mean(lotArea))
```




#This is a product of OpenIntro that is released under a Creative Commons Attribution-ShareAlike 3.0 Unported. This lab was written for OpenIntro by Andrew Bray and Mine ?etinkaya-Rundel.