Confidence-Intervals-Levels

Schwinn(Xuan) Chen

April 30, 2017

If we 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 we have access to only a sample of the population, as is often the case, the task becomes more complicated. What is our best guess for the typical size if we only know the sizes of several dozen houses? This sort of situation requires that we use our sample to make inference on what our population looks like.

Load tools and packages.

```
library(statsr)
library(dplyr)
library(ggplot2)
```

The data

We consider real estate data from the city of Ames, Iowa. Dataset ‘ames’ contains every real estate transaction in Ames, recorded by the City Assessor’s office. Our particular focus for this report 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.

```
data(ames)
```

Process

We will randomly collect samples form the population to learn how confidence intervals vary from one sampling statistic to another.

Here is the outline:

- Randomly obtain random 100 observations from the population.
- Repeat it 50 times. We will alculate the mean of each sampling.
- Then compare confidence intervals (CI) at three the confidence levels (CL) (90%, 95% and 99%) and visualize the proportion of the population mean in the intervals.

First, let’s learn and visualize the population data.

```
# choose "area" (House Size) variable from poluplation 'ames', remove NA data, and assigned a new dataframe named ames_area, calculate mu=mean, sd, and median.

ames_area<-ames%>%
filter(!is.na(area))%>%
select(area)

mu<-mean(ames_area$area)
median<-median(ames_area$area)
sd<-sd(ames_area$area)
paste('Population mean is',mu)
```

```
## [1] "Population mean is 1499.69044368601"
```

```
paste('Population median is',median)
```

```
## [1] "Population median is 1442"
```

```
paste('Population standard deviation is', sd)
```

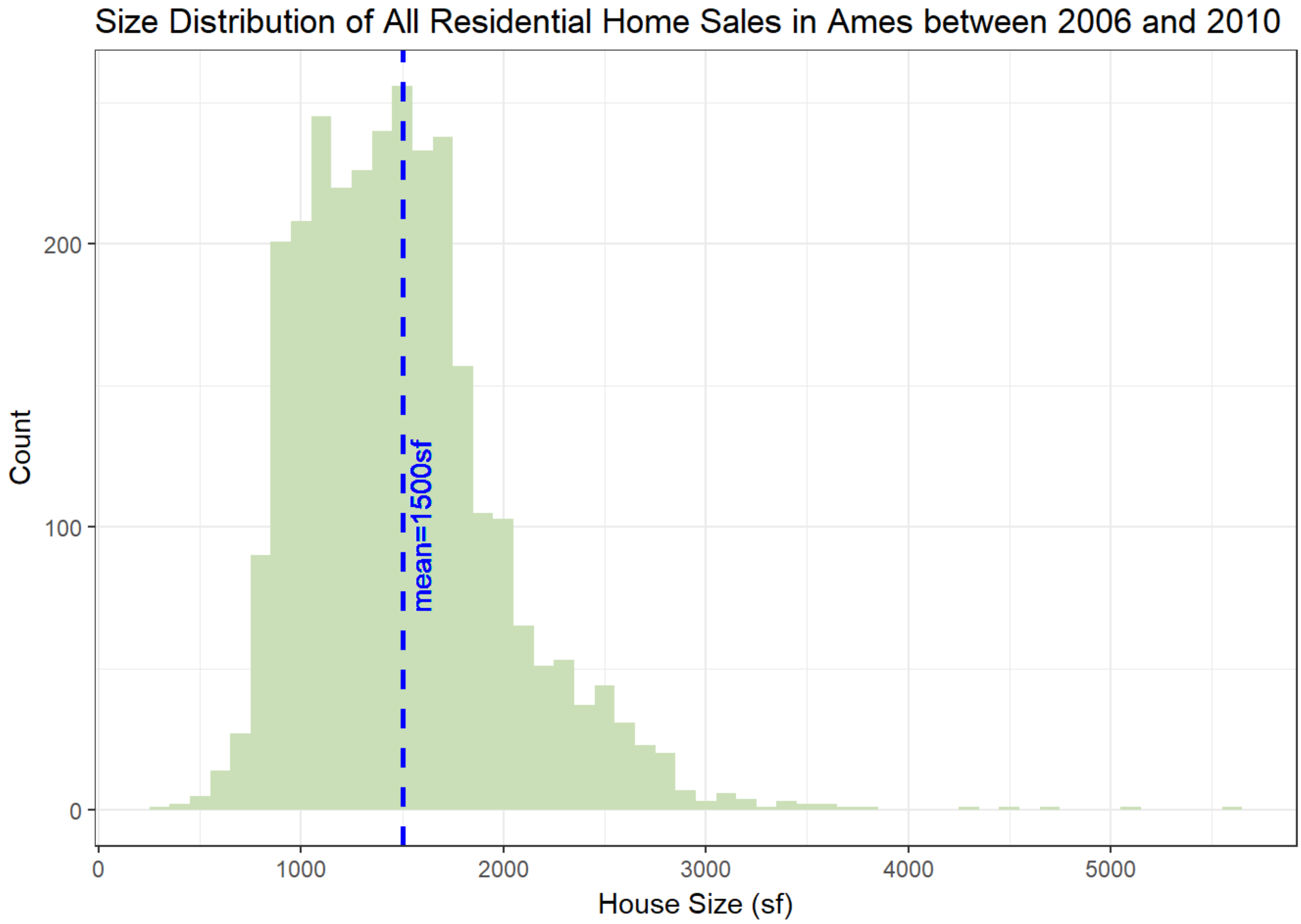
```
## [1] "Population standard deviation is 505.508887472041"
```

The above calculation shows that the mean, median and standard division of the size of all residential home sales in Ames between 2006 and 2010 are 1499.69sf, 1442sf and 505.5sf.

The histogram shows the distribution of the size of all residential home sales in Ames between 2006 and 2010. It is right skewed with a mean of 1500sf, which is marked by the vertical blue line.

```
library(ggplot2)
ggplot(ames_area, aes(x=area))+geom_histogram(binwidth=100, fill='#cadfb7')+theme_bw()+ylab('Count')+xlab('House Size (sf)')+ggtitle('Size Distribution of All Residential Home Sales in Ames between 2006 and 2010')+ geom_vline(aes(xintercept=mu),color="blue", linetype="dashed", size=1)+ geom_text(aes(x=1499.69, label="mean=1500sf", y=100), colour="blue", angle=90, vjust = 1.3, text=element_text(size=11))
```

```
## Warning: Ignoring unknown parameters: text
```



Now we can compare CIs of three the confidence levels (90%, 95% and 99%) and visualize whether the population mean falls in the intervals. We will need to select 100 random examples from the population.

Bounds of the CI can be written as: lower bound: $\bar{X}$ - Z*SE upper bound:* $\bar{X}$ + ZSE Z is the critical value of each CL. The Z score is calculated using the following codes:

```
Z_90 =-qnorm(0.05)
paste("Z score at 90% CL is",Z_90)
```

```
## [1] "Z score at 90% CL is 1.64485362695147"
```

```
Z_95=-qnorm(0.025)
paste("Z score at 95% CL is",Z_95)
```

```
## [1] "Z score at 95% CL is 1.95996398454005"
```

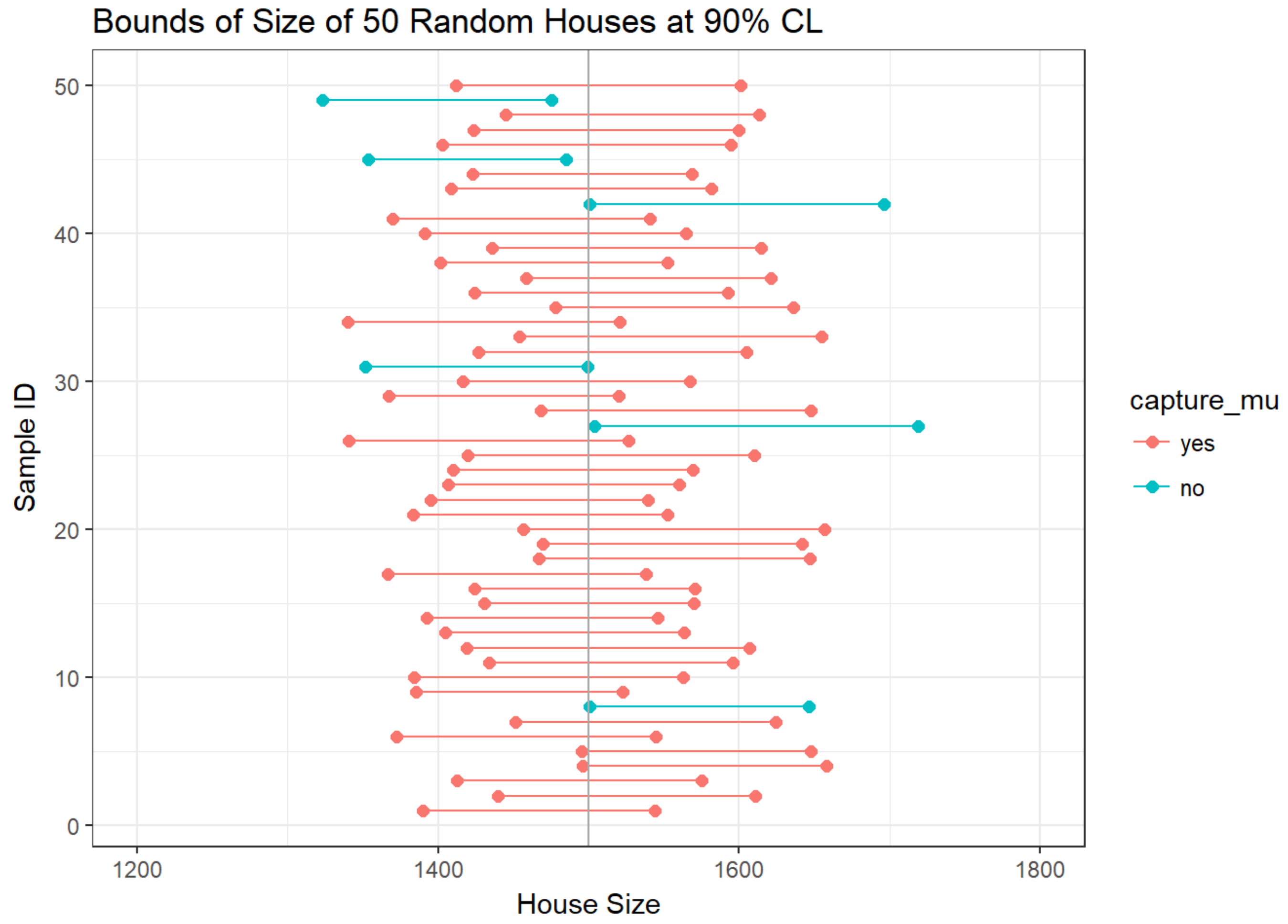
```
Z_99=-qnorm(0.005)
paste("Z score at 99% CL is",Z_99)
```

```
## [1] "Z score at 99% CL is 2.5758293035489"
```

Next, we will select 100 samples from the population and repeat the process 50 times. Therefore,each sampling contains 100 random samples from the population. We will graph the sampling distribution at 90%, 95% and 99% CL. We want to learn the bounds of CI and whether it captures the population mean.

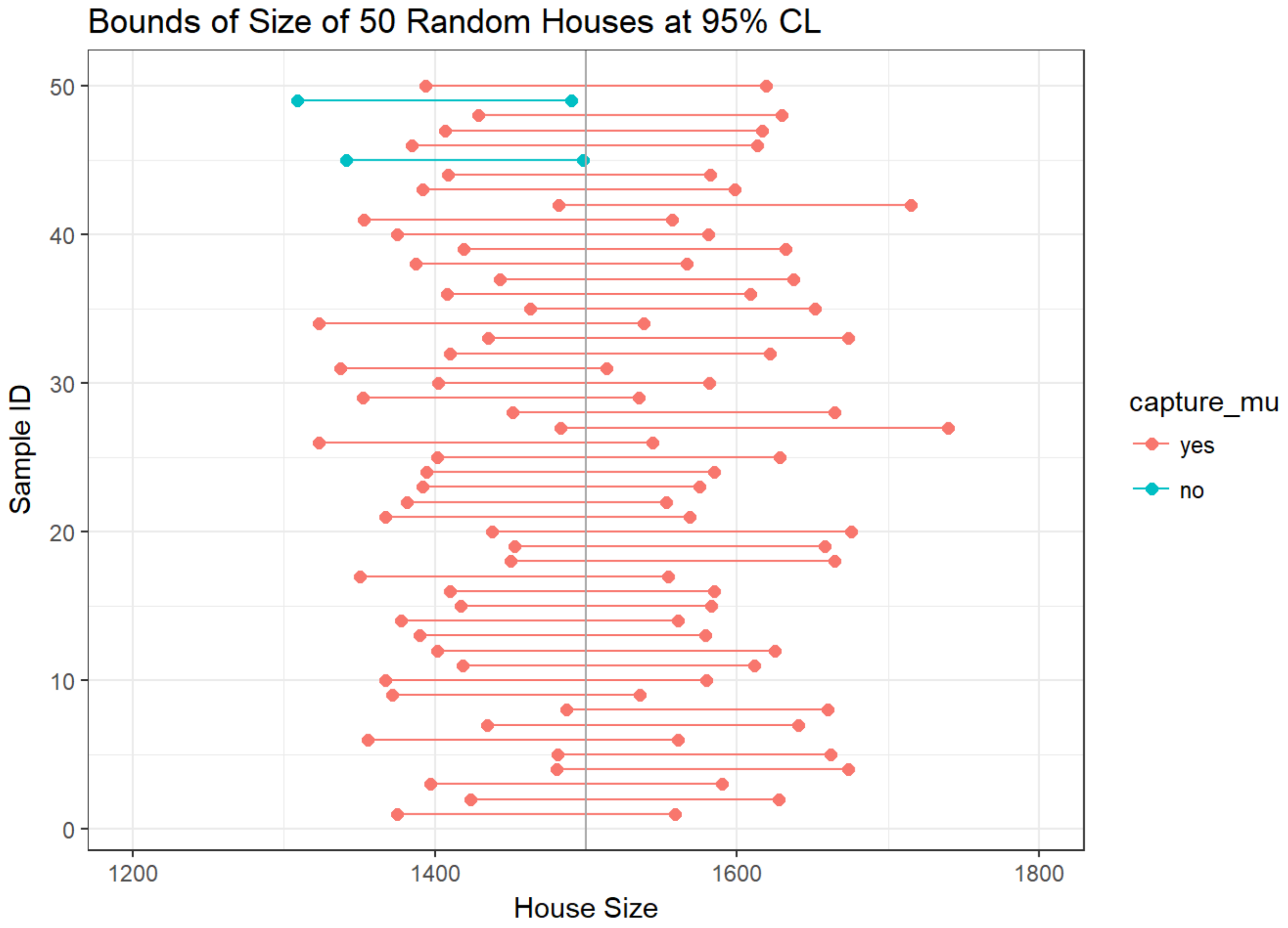
Sampling Statistics at 90% Confidence Level

```
n <- 100

set.seed(99-95-90)    ## use set.seed function to randomly sample the same data points
ci_90 <- ames_area %>%
 rep_sample_n(size = n, reps = 50, replace = TRUE) %>%
        summarise(lower = mean(area) - Z_90 * (sd(area) / sqrt(n)),
                  upper = mean(area) + Z_90 * (sd(area) / sqrt(n)))

#Add a column indicating the population mean falls in the confidence interval or not

ci_90<- ci_90 %>%
  mutate(capture_mu = ifelse(lower < mu & upper > mu, "yes", "no"))

# Rearrange ci_90 for plotting purposes

ci_90_arrange <- data.frame(ci_id = c(1:50, 1:50),
                      ci_bounds = c(ci_90$lower, ci_90$upper),
                      capture_mu = c(ci_90$capture_mu, ci_90$capture_mu))
ci_90_arrange$capture_mu=factor(ci_90_arrange$capture_mu, c("yes","no"))

#Plot 90% confidence intervals of 50 random sample of 100. Mark the population mean

ggplot(data = ci_90_arrange, aes(x = ci_bounds, y = ci_id,
                           group = ci_id, color = capture_mu)) + theme_bw()+labs(x
='House Size', y='Sample ID', title='Bounds of Size of 50 Random Houses at 90% CL'
)+scale_fill_discrete(name="Capture Population Mean")+
  geom_point(size = 2) +  # add points at the ends, size = 2
  geom_line() +           # connect with lines
  geom_vline(xintercept = mu, color = "darkgray")+ # draw vertical line
scale_x_continuous(limits=c(1200,1800))
```



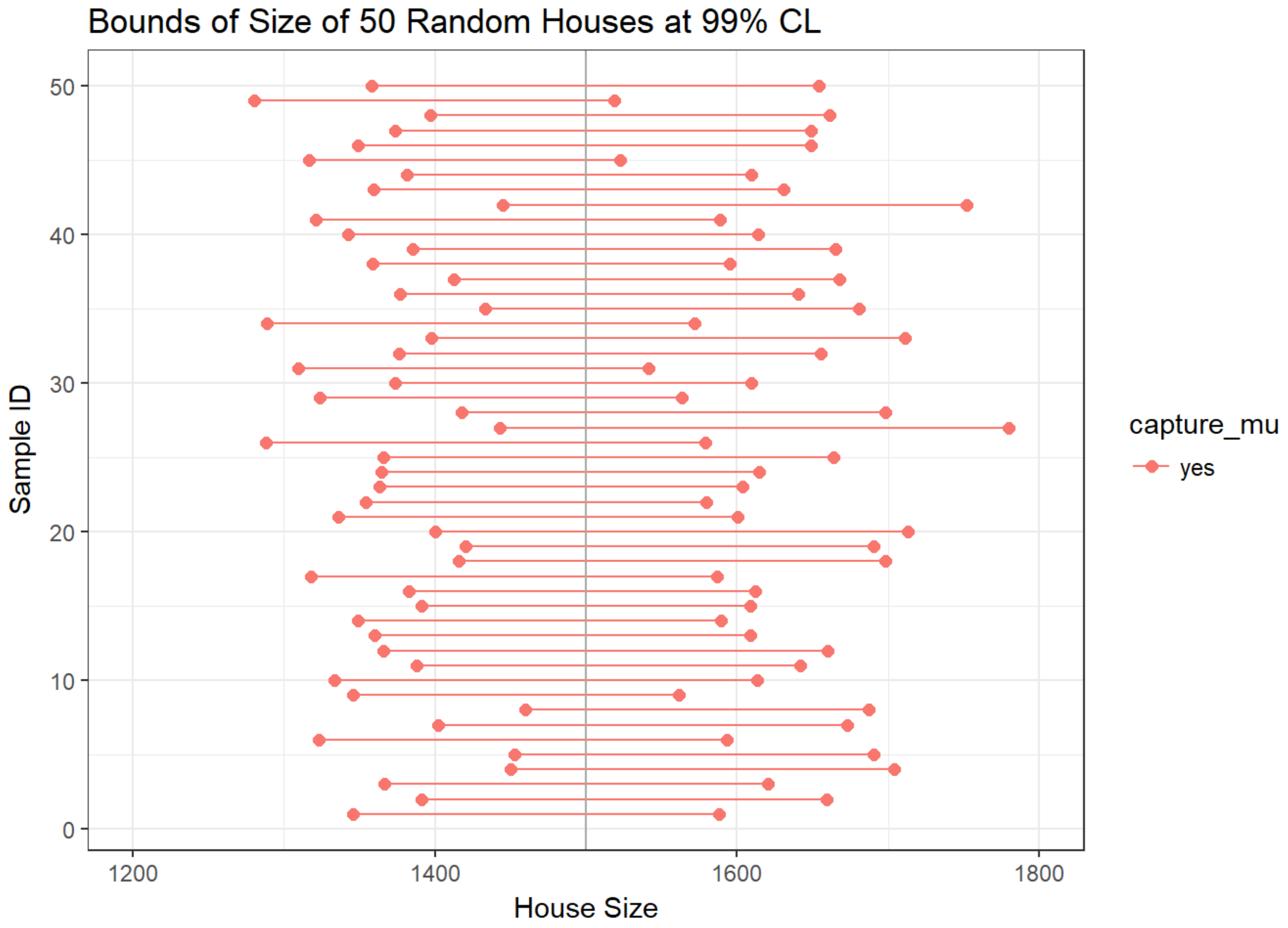
Sampling Statistics at 95% Confidence Level

```r
set.seed(99-95-90)   ## use set.seed function to randomly sample the same data points
ci_95 <- ames_area %>%
 rep_sample_n(size = n, reps = 50, replace = TRUE) %>%
        summarise(lower = mean(area) - Z_95 * (sd(area) / sqrt(n)),
                  upper = mean(area) + Z_95 * (sd(area) / sqrt(n)))

#Add a column indicating the population mean falls in the confidence interval or not

ci_95<- ci_95 %>%
  mutate(capture_mu = ifelse(lower < mu & upper > mu, "yes", "no"))

# Rearrange ci_90 for plotting purposes

ci_95_arrange <- data.frame(ci_id = c(1:50, 1:50),
                      ci_bounds = c(ci_95$lower, ci_95$upper),
                      capture_mu = c(ci_95$capture_mu, ci_95$capture_mu))
ci_95_arrange$capture_mu=factor(ci_95_arrange$capture_mu, c("yes","no"))

#Plot 90% confidence intervals of 50 random sample of 100. Mark the population mean

ggplot(data = ci_95_arrange, aes(x = ci_bounds, y = ci_id,
                           group = ci_id, color=capture_mu)) +theme_bw()+labs(x='House Size', y='Sample ID', title='Bounds of Size of 50 Random Houses at 95% CL' )+scale_fill_discrete(name="Capture Population Mean")+
  geom_point(size = 2) +  # add points at the ends, size = 2
  geom_line() +           # connect with lines
  geom_vline(xintercept =mu, color = "darkgray")+ # draw vertical line
scale_x_continuous(limits=c(1200,1800))
```



Sampling Statistics at 99% Confidence Level

```
set.seed(99-95-90)  ## use set.seed function to randomly sample the same data point
s
ci_99 <- ames_area %>%
 rep_sample_n(size = n, reps = 50, replace = TRUE) %>%
        summarise(lower = mean(area) - Z_99 * (sd(area) / sqrt(n)),
                  upper = mean(area) + Z_99 * (sd(area) / sqrt(n)))

#Add a column indicating the population mean falls in the confidence interval or no
t

ci_99<- ci_99 %>%
  mutate(capture_mu = ifelse(lower < mu & upper > mu, "yes", "no"))

# Rearrange ci_90 for plotting purposes

ci_99_arrange <- data.frame(ci_id = c(1:50, 1:50),
                      ci_bounds = c(ci_99$lower, ci_99$upper),
                      capture_mu = c(ci_99$capture_mu, ci_99$capture_mu))
ci_99_arrange$capture_mu=factor(ci_99_arrange$capture_mu, c("yes","no"))

#Plot 90% confidence intervals of 50 random sample of 100. Mark the population mean

ggplot(data = ci_99_arrange, aes(x = ci_bounds, y = ci_id,
                           group = ci_id, color=capture_mu)) +theme_bw()+labs(x='H
ouse Size', y='Sample ID', title='Bounds of Size of 50 Random Houses at 99% CL' )+s
cale_fill_discrete(name="Capture Population Mean")+
  geom_point(size = 2) +  # add points at the ends, size = 2
  geom_line() +           # connect with lines
  geom_vline(xintercept = mu, color = "darkgray")+ # draw vertical line
  scale_x_continuous(limits=c(1200,1800))
```



Summary

As we can see from the three graphs: At 90% CL, 6 out of 50 samplings does not capture the population mean. It has the smallest bounds of CI.
At 95% CL, 2 out of 50 samplings does not capture the population mean. It has the 2nd largest bounds of CI. At 99% CL, all 50 samplings capture the population mean. It has the largest bounds of CI.