Lab 6 - Foundations for statistical inference - Confidence intervals

SANAZ SAADATIFAR

3/2/2022

Lab report

Load data

```
ames <- read_csv("https://dyurovsky.github.io/85309/data/lab6/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)
```

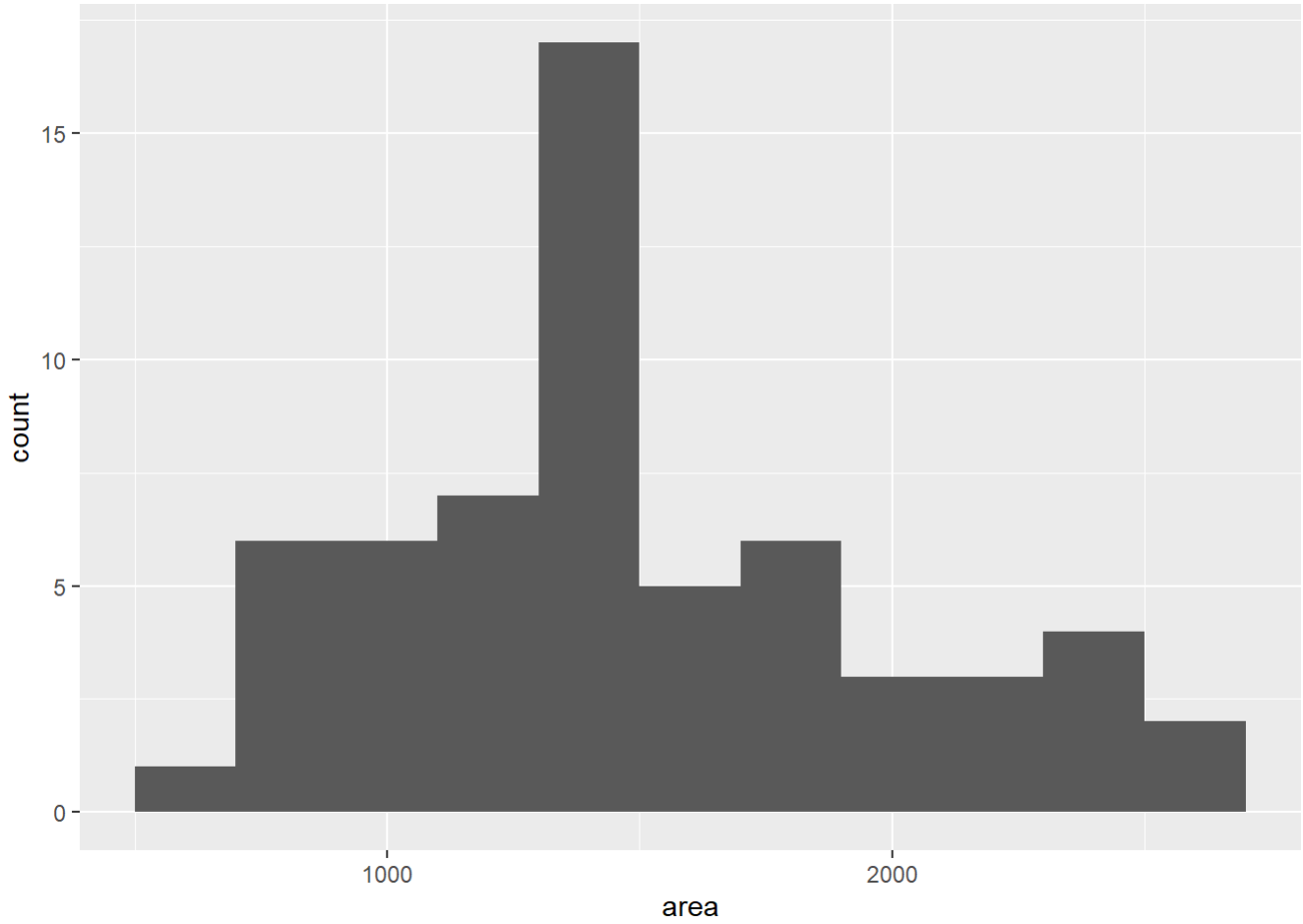
Exercises:

Exercise 1:

```
# enter your code for Exercise 1 here
set.seed(85309)
n <- 60

samp <- ames %>%
  sample_n(n)

samp %>%
  summarise(mean = mean(area),
            sd = sd(area))
```

```
## # A tibble: 1 x 2
##    mean    sd
##   <dbl> <dbl>
## 1 1490.  482.
```

```
ggplot(samp, aes(x = area)) +
  geom_histogram(binwidth = 200)
```



because the distribution of areas is not too skewed, we use the mean to describe the “typical” area. the mean area of the houses is 1490.483.

Exercise 2:

we should not expect another classmate’s mean area to be the same because they have the different sample of 60 houses. but we should expect the similar numbers because the central limit theorem says that it should be distributed around the population mean.

Exercise 3:

```
# enter your code for Exercise 3 here
set.seed(85309)
z_star_95 <- qnorm(0.975)
z_star_95
```

```
## [1] 1.959964
```

```
samp %>%
  summarise(lower = mean(area) - z_star_95 * (sd(area) / sqrt(n)),
            upper = mean(area) + z_star_95 * (sd(area) / sqrt(n)))
```

```
## # A tibble: 1 x 2
##   lower upper
##   <dbl> <dbl>
## 1 1368. 1613.
```

1. independent samples. that's true because we know how the samples are taken.
2. sample population distribution approximately normal or sample size big enough given the skew in the population. here the population distribution is roughly normal with no too skew, and sample size is 60 which is big enough so that is ok.

Exercise 4:

The 95% of the time we take a sample from the population (randomly, of this size), the population mean will fall in the range given by the 95% confidence interval.

Exercise 5:

```
# enter your code for Exercise 5 here
mu <- ames %>%
  summarise(mu = mean(area)) %>%
  pull()
```

my 95% confidence interval captures the true mean value. everybody captures roughly true mean.

Exercise 6:

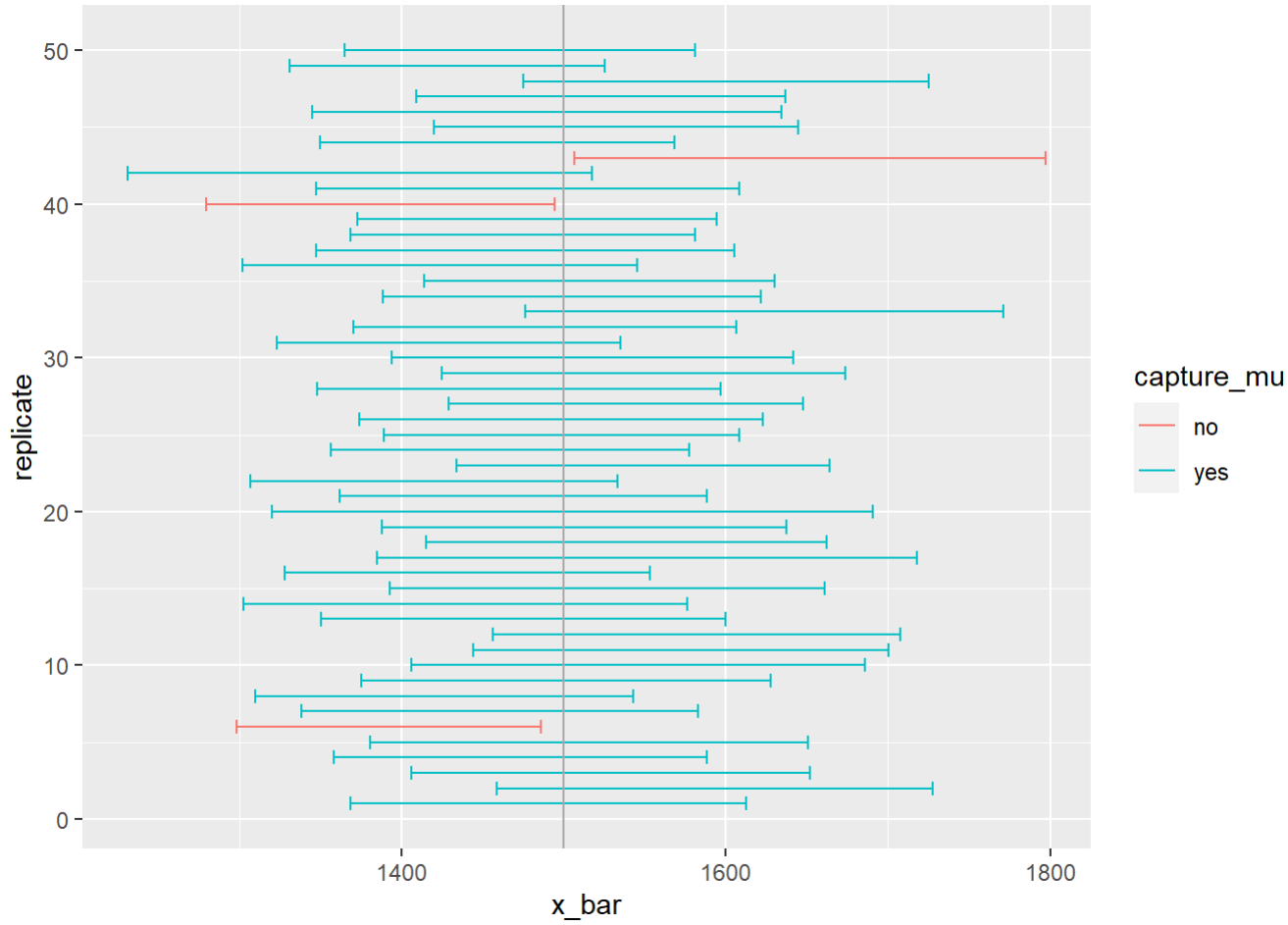
we should expect the 95% of people to capture the true mean because we had a 95% confidence interval.

Exercise 7:

```
# enter your code for Exercise 7 here
set.seed(85309)

one_sample <- function() {
  ames %>%
    sample_n(n) %>%
    summarise(x_bar = mean(area),
              se = sd(area) / sqrt(n),
              me = z_star_95 * se,
              lower = x_bar - me,
              upper = x_bar + me)
}

ci <- replicate(50, one_sample(), simplify = FALSE) %>%
  bind_rows() %>%
  mutate(replicate = 1:n()) # number each replication



ci %>%
  slice(1:5)
```

```
## # A tibble: 5 x 6
##   x_bar    se    me lower upper replicate
##   <dbl> <dbl> <dbl> <dbl> <dbl>     <int>
## 1 1490.  62.3  122. 1368. 1613.         1
## 2 1593.  68.6  135. 1459. 1728.         2
## 3 1529.  62.7  123. 1406. 1652.         3
## 4 1473.  58.7  115. 1358. 1588.         4
## 5 1516.  68.9  135. 1381. 1651.         5
```

```
ci_captured <- ci %>%
  mutate(capture_mu = if_else(lower < mu & upper > mu, "yes", "no"))


ggplot(ci_captured, aes(x = replicate, y = x_bar, color = capture_mu)) +
  geom_errorbar(aes(ymin = lower, ymax = upper)) +
  geom_hline(aes(yintercept = mu), color = "darkgray") + # draw vertical line
  coord_flip()
```




```
ci_captured %>%
  summarise(p = mean(capture_mu == 'yes'))
```

```
## # A tibble: 1 x 1
##       p
##   <dbl>
## 1  0.94
```

we should expect 95% of the times of sample of 50 to capture the mean. so this is around 2.5 which is consistent with the answer we found.

More Practice

Exercise 8:

```
# enter your code for Exercise 8 here
z_star_99.7 <- qnorm(0.9985)
z_star_99.7
```

```
## [1] 2.967738
```

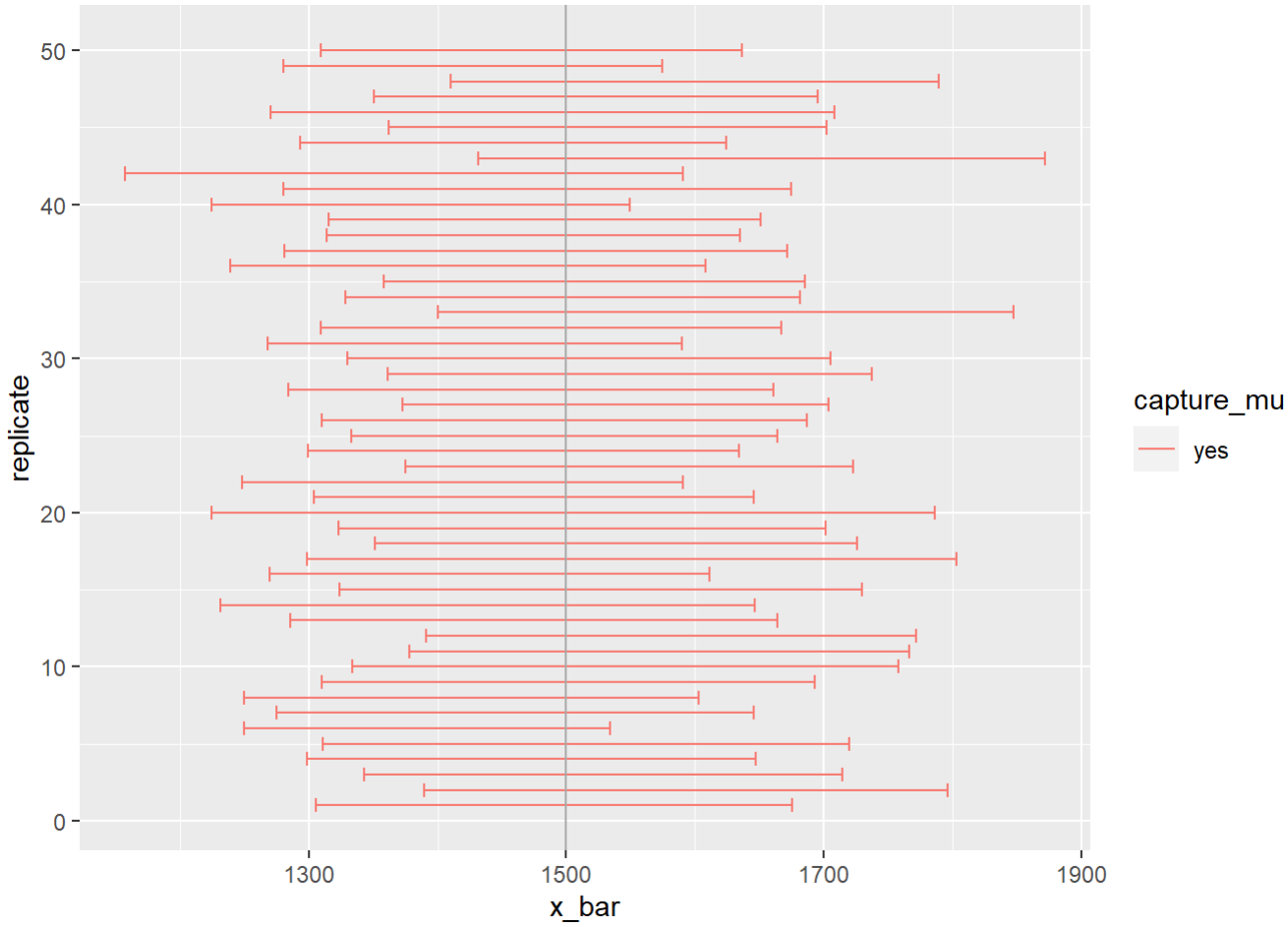
the critical value is 2.967738 for the confidence level of 99.7%.

Exercise 9:

```
# enter your code for Exercise 9 here
set.seed(85309)

one_sample2 <- function() {
  ames %>%
    sample_n(n) %>%
    summarise(x_bar = mean(area),
              se = sd(area) / sqrt(n),
              me = z_star_99.7 * se,
              lower = x_bar - me,
              upper = x_bar + me)
}

ci2 <- replicate(50, one_sample2(), simplify = FALSE) %>%
  bind_rows() %>%
  mutate(replicate = 1:n()) # number each replication



ci2 %>%
  slice(1:5)
```

```
## # A tibble: 5 x 6
##   x_bar    se    me lower upper replicate
##   <dbl> <dbl> <dbl> <dbl> <dbl>     <int>
## 1 1490.  62.3  185. 1306. 1675.         1
## 2 1593.  68.6  204. 1389. 1797.         2
## 3 1529.  62.7  186. 1343. 1715.         3
## 4 1473.  58.7  174. 1299. 1647.         4
## 5 1516.  68.9  205. 1311. 1720.         5
```

```
ci_captured2 <- ci2 %>%
  mutate(capture_mu = if_else(lower < mu & upper > mu, "yes", "no"))


ggplot(ci_captured2, aes(x = replicate, y = x_bar, color = capture_mu)) +
  geom_errorbar(aes(ymin = lower, ymax = upper)) +
  geom_hline(aes(yintercept = mu), color = "darkgray") + # draw vertical line
  coord_flip()
```




```
ci_captured2 %>%
  summarise(p = mean(capture_mu == 'yes'))
```

```
## # A tibble: 1 x 1
##       p
##   <dbl>
## 1     1
```

the proportion of intervals that include the true population mean is 1 (100%) which is very close to the confidence interval. because we expect that with the confidence interval of 99.7%, the 99.7% of the time of sample of 50 to capture the mean. so worst case we would just 1 sample that does not capture the mean of population. however in this sample list, all samples could capture the mean.