Probability distributions and inference

Confidence intervals

Jenine Harris
Brown School

Computing and interpreting confidence intervals around means and proportions

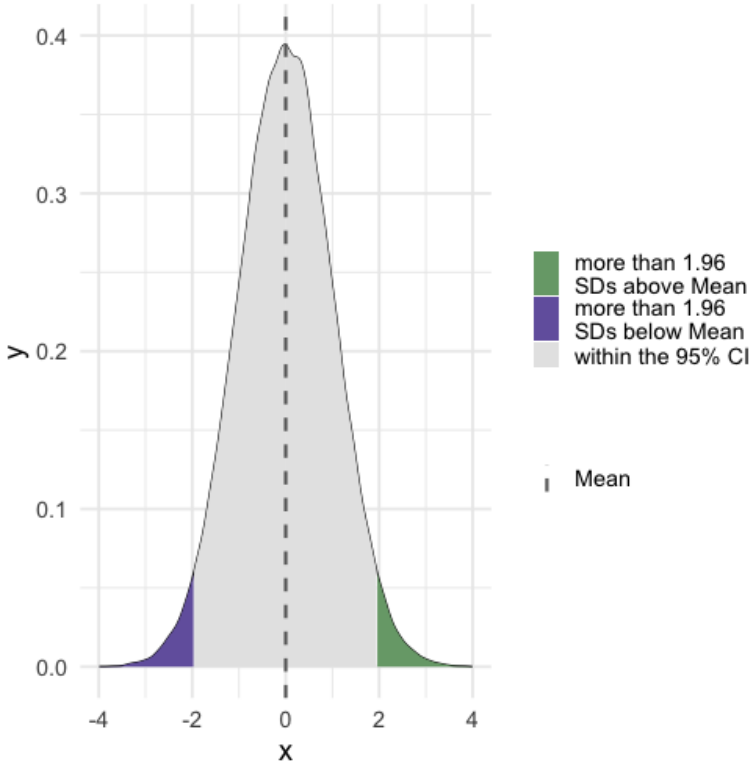
- The range around the sample mean where the population mean *might be* shows the uncertainty of computing a mean from a sample.
- These ranges are reported as **confidence intervals** (or CIs).
- In this context, *confidence* is not about an emotion or feeling, but about how much uncertainty there is in the results.
- Most of the time, social scientists report 95% intervals or **95% confidence intervals** which show the range where the population value would likely be 95 times if the study were conducted 100 times.
- Sometimes smaller or larger intervals are reported, like a *68% confidence interval* (68% CI) or a *99% confidence interval* (99% CI), but usually it's a 95% confidence interval.

95% interval interpretation

- The 95% interval idea comes from the following properties:
  - about 95% of values lie within 2 standard deviations of the mean for a variable that is normally distributed
    - Remember, the number of standard deviations some observation is away from the mean is called a z-score.
  - the standard error of a sample is a good estimate of the standard deviation of the sampling distribution, which is normally distributed
  - the mean of the sampling distribution is a good estimate of the population mean
  - so, most sample means will be within 2 standard errors of the population mean (to be precise, this is actually 1.96 rather than 2).

Observations outside the confidence interval

- With 95% of observations being within 1.96 standard deviations of the mean, this leaves 5% of observations in the tails of the distribution, outside the confidence interval, like this:



Import the distance data

```
# distance to substance abuse facility with medication assisted treatmen
dist.mat <- read.csv("/Users/harrisj/Box/teaching/Teaching/Fall2020/data

# rename variable
library(package = "tidyverse")
dist.mat.cleaned <- dist.mat %>%
  rename('distance' = VALUE)

# review the data
summary(object = dist.mat)
```

```
##     STATEFP         COUNTYFP          YEAR       INDICATOR
##  Min.   : 1.00   Min.   :  1.0   Min.   :2017   Length:3214
##  1st Qu.:19.00   1st Qu.: 35.0   1st Qu.:2017   Class :character
##  Median :30.00   Median : 79.0   Median :2017   Mode  :character
##  Mean   :31.25   Mean   :101.9   Mean   :2017
##  3rd Qu.:46.00   3rd Qu.:133.0   3rd Qu.:2017
##  Max.   :72.00   Max.   :840.0   Max.   :2017
##      VALUE           STATE           STATEABBREVIATION     COUNTY
##  Min.   :  0.00   Length:3214        Length:3214        Length:3214
##  1st Qu.:  9.25   Class :character   Class :character   Class :character
##  Median : 18.17   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 24.04
##  3rd Qu.: 31.00
##  Max.   :414.86
```

Take a sample of 500

```
# set a starting value for sampling
# same seed from prior video
set.seed(seed = 1945)

# sample 500 counties at random
counties.500 <- dist.mat.cleaned %>%
  sample_n(size = 500, replace = TRUE)

# compute the mean death rate in the sample
counties.500 %>%
  summarize(mean.s1 = mean(x = distance, na.rm = TRUE))
```

```
##    mean.s1
## 1 24.40444
```

Use R to computing a 95% confidence interval for a mean

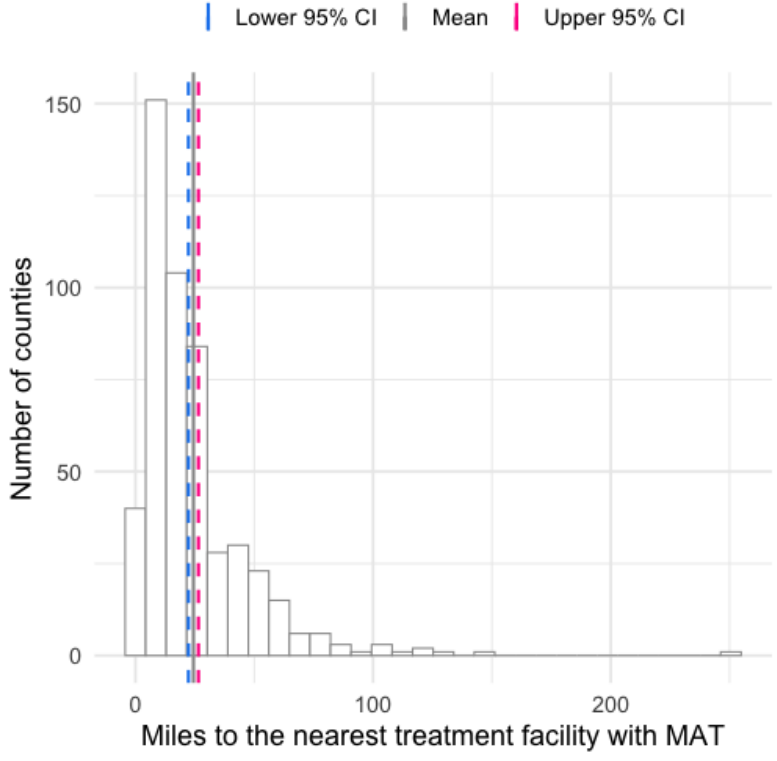
```
# add CI to summary statistics
summ.500.counties <- counties.500 %>%
  summarize(mean.s1 = mean(x = distance),
            sd.s1 = sd(x = distance),
            se.s1 = sd(x = distance)/sqrt(x = length(x = distance)),
            lower.ci.s1 = mean.s1 - 1.96*se.s1,
            upper.ci.s1 = mean.s1 + 1.96*se.s1)
summ.500.counties
```

```
##    mean.s1    sd.s1    se.s1 lower.ci.s1 upper.ci.s1
## 1 24.40444 23.79142 1.063985    22.31903    26.48985
```

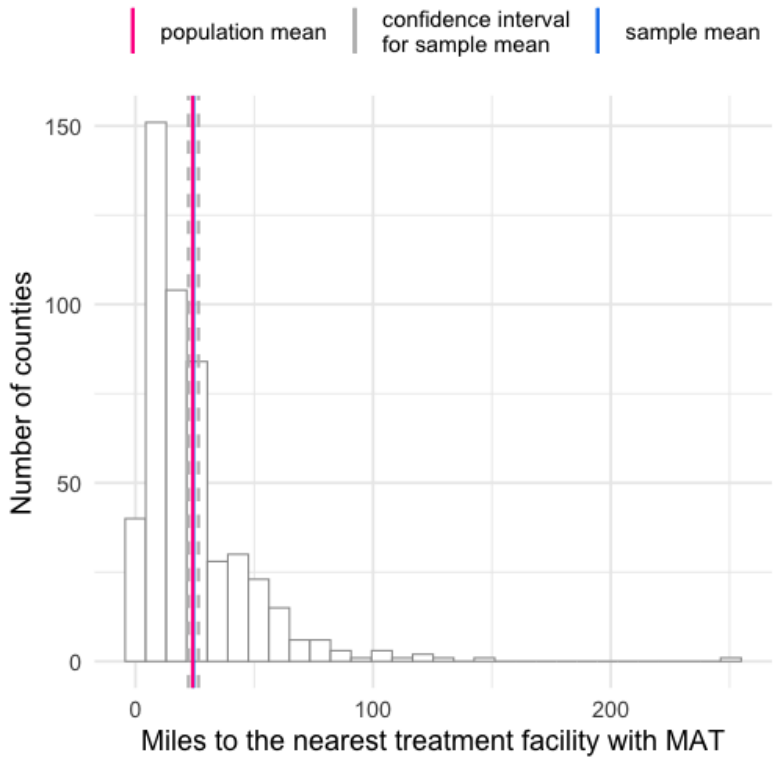
- The 95% confidence interval for the mean distance from the sample of 500 counties was 22.32 - 26.49. Kiara interpreted the results for the team:
- Reporting the mean and sd:
  - The mean distance in miles to the nearest substance abuse treatment facility with MAT in a sample of 500 counties is 24.4; the true or population mean distance in miles to a facility likely lies between 22.32 - 26.49 (m = 24.4; 95% CI = 22.32 - 26.49).

Examine the mean and sd with a histogram

- A histogram of the distance to a treatment facility showing the mean and the 95% confidence interval around the mean:



Examine the sample mean, pop mean, and CI



Examine the other sample

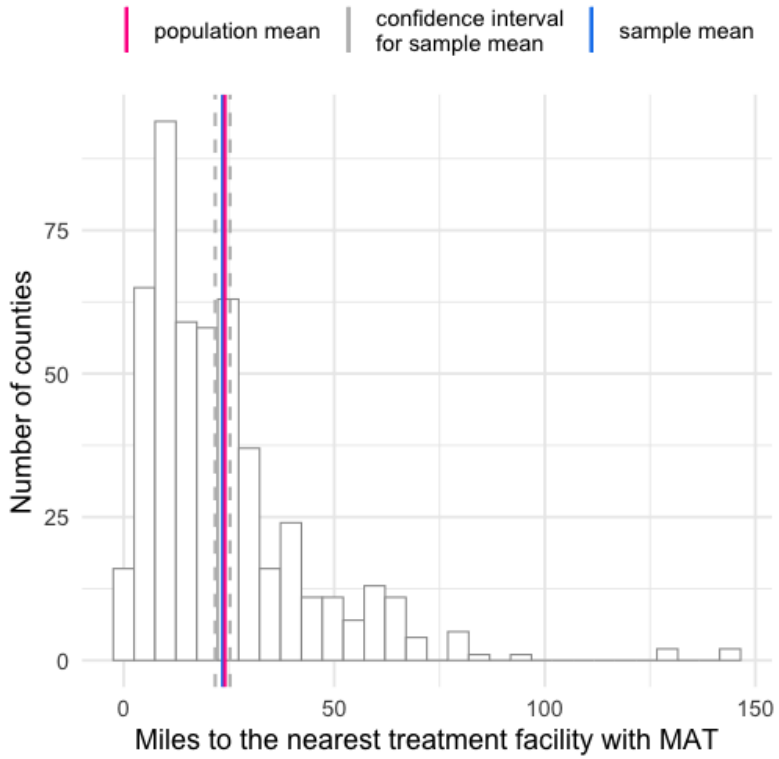
```
# set a different starting value for sampling
set.seed(seed = 48)

# sample 500 counties at random
counties.500.2 <- dist.mat.cleaned %>%
  sample_n(size = 500, replace = TRUE)

# compute the mean death rate in the sample
counties.500.2 %>%
  summarize(mean.s2 = mean(x = distance, na.rm = TRUE))
```

```
##    mean.s2
## 1 23.49652
```

```
# add CI to summary statistics other sample
counties.500.2 %>%
  summarize(mean = mean(x = distance),
            sd = sd(x = distance),
            se = sd(x = distance)/sqrt(x = length(x = distance)),
            lower.ci = mean - 1.96*se,
            upper.ci = mean + 1.96*se)
```

```
##       mean       sd        se lower.ci upper.ci
## 1 23.49652 20.08756 0.8983431 21.73577 25.25727
```

Plot second sample mean with CI



Examine confidence intervals for lots of samples

- For both, the population mean was inside the confidence interval and near the sample mean.
- What about the confidence intervals when they took 20, 100, and 1000 samples.
- Find these values by using `group_by()` and `summarize()`, start with the 20 samples data:

```
# take 20 samples
set.seed(seed = 111)
samples.20 <- bind_rows(replicate(n = 20, dist.mat.cleaned %>%
                                  sample_n(size = 500, replace = TRUE),
                                  simplify = FALSE), .id = "sample_num")

# add CI to summary statistics other sample
samp.20.stats <- samples.20 %>%
  group_by(sample_num) %>%
  summarize(means = mean(x = distance),
            sd = sd(x = distance),
            se = sd(x = distance)/sqrt(x = length(x = distance)),
            lower.ci = means - 1.96*se,
            upper.ci = means + 1.96*se)
samp.20.stats
```

Review the CI for 20 samples

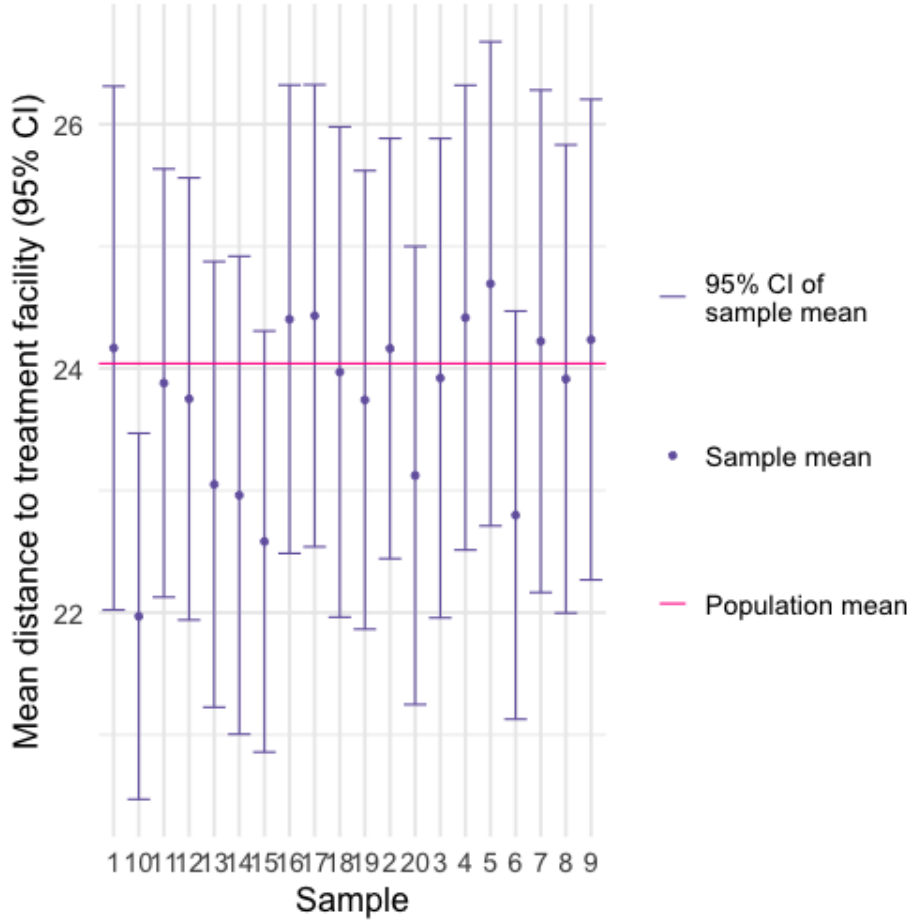
```
## # A tibble: 20 x 6
##    sample_num means    sd    se lower.ci upper.ci
##    <chr>      <dbl> <dbl> <dbl>    <dbl>    <dbl>
##  1 1           24.2  24.5 1.09      22.0     26.3
##  2 10          22.0  17.1 0.765     20.5     23.5
##  3 11          23.9  20.0 0.895     22.1     25.6
##  4 12          23.8  20.7 0.924     21.9     25.6
##  5 13          23.1  20.8 0.931     21.2     24.9
##  6 14          23.0  22.3 0.999     21.0     24.9
##  7 15          22.6  19.7 0.880     20.9     24.3
##  8 16          24.4  21.9 0.979     22.5     26.3
##  9 17          24.4  21.6 0.965     22.5     26.3
## 10 18          24.0  22.9 1.02      22.0     26.0
## 11 19          23.7  21.4 0.958     21.9     25.6
## 12 2           24.2  19.6 0.878     22.4     25.9
## 13 20          23.1  21.4 0.957     21.2     25.0
## 14 3           23.9  22.4 1.00      22.0     25.9
## 15 4           24.4  21.7 0.971     22.5     26.3
## 16 5           24.7  22.6 1.01      22.7     26.7
## 17 6           22.8  19.1 0.853     21.1     24.5
## 18 7           24.2  23.5 1.05      22.2     26.3
## 19 8           23.9  21.9 0.978     22.0     25.8
## 20 9           24.2  22.4 1.00      22.3     26.2
```

- Do they all contain the population mean of 24.04?

Use a graph to examine CI

```
# graph means and CI for 20 samples
samp.20.stats %>%
  ggplot(aes(y = means, x = sample_num)) +
  geom_errorbar(aes(ymin = lower.ci,
                    ymax = upper.ci,
                linetype = "95% CI of\nsample mean"), color = "#7463AC")
  geom_point(stat = "identity", aes(color = "Sample mean")) +
  geom_hline(aes(yintercept = 24.04, alpha = "Population mean"), color =
  labs(y = "Mean distance to treatment facility (95% CI)",
       x = "Sample") +
  scale_color_manual(values = "#7463AC", name="") +
  scale_linetype_manual(values = c(1, 1), name = "") +
  scale_alpha_manual(values = 1, name = "") +
  theme_minimal(base_size = 18)
```

Use a graph to examine CI



- The 95% confidence intervals for 19 of the 20 samples contained the population mean.

Examining the CI for 100 sample means

- Get the 100 sample means

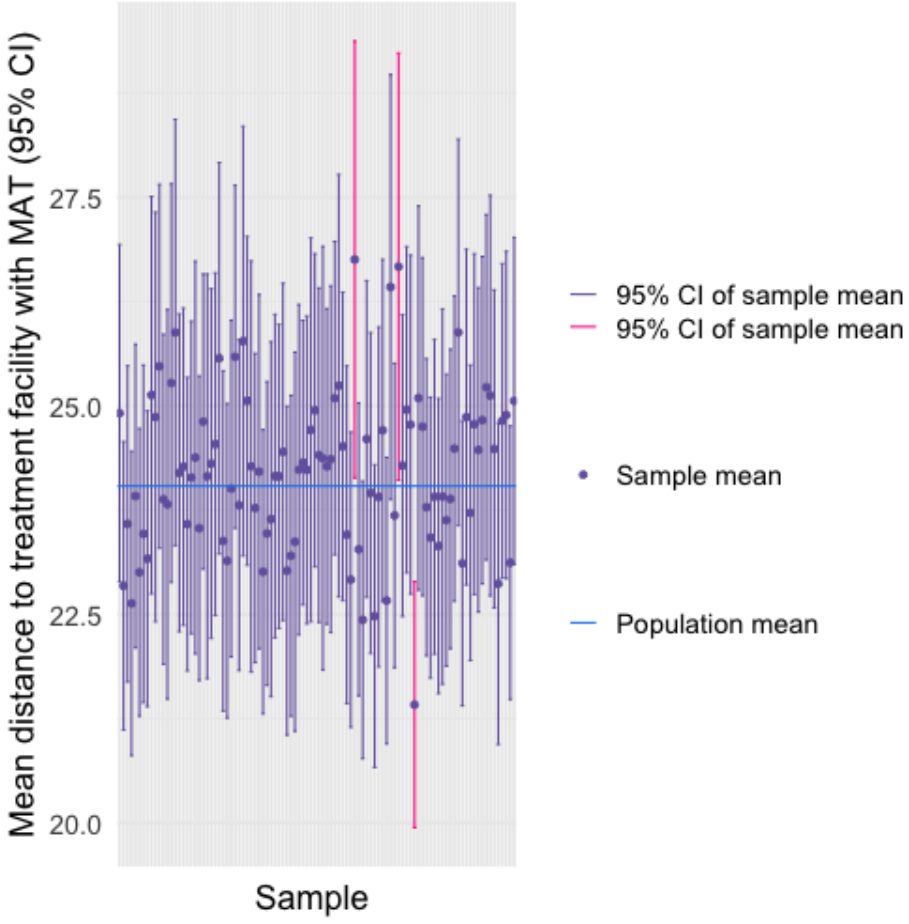
```
# 100 samples
samples.100 <- bind_rows(replicate(n = 100, dist.mat.cleaned %>%
                                   sample_n(size = 500, replace = TRUE),
                                   simplify = FALSE), .id = "sample_num")

# sample statistics
samp.100.stats <- samples.100 %>%
  group_by(sample_num) %>%
  summarize(means = mean(x = distance),
            sd = sd(x = distance),
            se = sd(x = distance)/sqrt(x = length(x = distance)),
            lower.ci = means - 1.96*se,
            upper.ci = means + 1.96*se)
```

Graphing the CI for 100 sample means

```
# graph means and CI for 100 samples
samp.100.stats %>%
  ggplot(aes(y = means, x = sample_num)) +
  geom_errorbar(aes(ymin = lower.ci,
                    ymax = upper.ci,
                linetype = "95% CI of\nsample mean"), color = "#7463AC")
  geom_point(stat = "identity", aes(color = "Sample mean")) +
  geom_hline(aes(yintercept = 24.04, alpha = "Population mean"), color =
  labs(y = "Mean distance to treatment facility (95% CI)",
       x = "Sample") +
  scale_color_manual(values = "#7463AC", name="") +
  scale_linetype_manual(values = c(1, 1), name = "") +
  scale_alpha_manual(values = 1, name = "") +
  theme_minimal(base_size = 18) +
  theme(axis.text.x = element_blank())
```

Graphing the CI for 100 sample means

Highlighting the CI not including the pop mean



Confidence intervals for percentages

- Shockingly, the 95% confidence interval around a proportion is computed in a similar way since the sampling distribution for a binary variable is **normally distributed**.
- For variables that only have two values (e.g., Yes and No, success and failure, 1 and 0), the mean of the variable is the same as the percentage of the group of interest.
- For example, consider a survey of 10 people which asked if they drink coffee or do not drink coffee where drinking coffee is coded as 1 and not drinking coffee is coded as 0, for example:

```
# do you drink coffee?
coffee <- c(1, 0, 1, 1, 0,
            0, 0, 1, 1, 1)

# mean of coffee variable
mean(x = coffee)
```

```
## [1] 0.6
```

- The percentage of people in a sample who have the variable category of interest is the mean of the sample for that variable.
- The mean of a binary variable like this one is typically abbreviated as p for proportion rather than m for mean.

Importing the opioid policy data

```
# open state opioid program data
state.opioid.pgm.2018 <- read.csv(file = "/Users/harrisj/Box/teaching/Te

# recode Yes to 1 and No to 0
# change long name to pdmp
state.opioid.pgm.2018.cleaned <- state.opioid.pgm.2018 %>%
  rename(pdmp = Required.Use.of.Prescription.Drug.Monitoring.Programs) %
  mutate(pdmp = as.numeric(x = as.factor(pdmp)) - 1)

# find the mean of pdmp
state.opioid.pgm.2018.cleaned %>%
  summarize(p = mean(x = pdmp))
```

```
##           p
## 1 0.627451
```

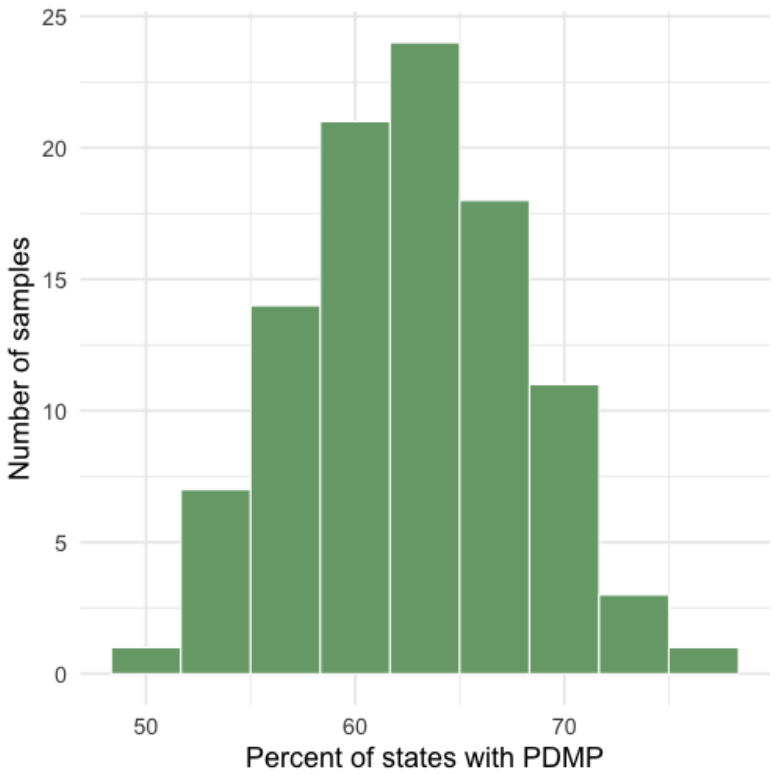
- The mean shows .6275 or 62.75% of the states have a PDMP.

Take samples of states and find sample means

```
# get 100 samples
# each sample has 30 states
# put samples in a data frame with each sample having
# a unique id called sample_num
set.seed(seed = 143)
samples.30.states <- bind_rows(replicate(n = 100, state.opioid.pgm.2018.
                               sample_n(size = 30, replace = FALSE),
                               simplify = FALSE), .id = "sample_num")

# find the mean for each sample
sample.30.means.states <- samples.30.states %>%
  group_by(sample_num) %>%
  summarize(p.pdmp = mean(x = pdmp, na.rm = TRUE))
sample.30.means.states
```

```
## # A tibble: 100 x 2
##    sample_num p.pdmp
##    <chr>       <dbl>
##  1 1           0.567
##  2 10          0.733
##  3 100         0.6
##  4 11          0.533
##  5 12          0.567
##  6 13          0.733
```

Graph the sampling distribution



Standard error for binary variables

- For any given sample, the 95% confidence interval for the mean (which is the percentage in the category of interest) can be computed using the same formula of m + 1.96 · se and m - 1.96 · se.
- The only thing needed now is the standard error.
- For binary variables, the standard error is computed:

$$se_p = \sqrt{\frac{p \cdot (1 - p)}{n}}$$

- Where p is the mean (proportion of 1s) and n is the sample size.

Get sample statistics

```
## # A tibble: 100 x 7
##    sample_num     p     n     se lower.ci upper.ci differs
##    <chr>      <dbl> <int>  <dbl>    <dbl>    <dbl> <lgl>
##  1 1          0.567    30 0.0905    0.389    0.744 FALSE
##  2 10         0.733    30 0.0807    0.575    0.892 FALSE
##  3 100        0.6      30 0.0894    0.425    0.775 FALSE
##  4 11         0.533    30 0.0911    0.355    0.712 FALSE
##  5 12         0.567    30 0.0905    0.389    0.744 FALSE
##  6 13         0.733    30 0.0807    0.575    0.892 FALSE
##  7 14         0.533    30 0.0911    0.355    0.712 FALSE
##  8 15         0.6      30 0.0894    0.425    0.775 FALSE
##  9 16         0.533    30 0.0911    0.355    0.712 FALSE
## 10 17         0.6      30 0.0894    0.425    0.775 FALSE
## # … with 90 more rows
```

Graph means & CI for 100 samples

```
# graph means and CI for 100 samples
samp.100.stats.states %>%
    ggplot(aes(y = p, x = sample_num)) +
  geom_errorbar(aes(ymin = lower.ci,
                    ymax = upper.ci,
                color = differs)) +
  geom_point(stat = "identity", aes(fill = "Sample proportion"), color =
  geom_hline(aes(yintercept = .627451, linetype = "Population proportion
  labs(y = "Proportion of states with PDMP",
       x = "Sample ") +
  scale_fill_manual(values = "#7463AC", name = "") +
  scale_color_manual(values = c("#7463AC", "deeppink"), name = "",
                     labels = c("95% CI of\nsample proportion", "\n95% C
  scale_linetype_manual(values = c(1, 1), name = "") +
  theme_minimal(base_size = 18) +
  theme(axis.text.x = element_blank())
```

- All of the 100 samples had a 95% confidence interval including the population mean of 62.75% of states with PDMP.

Graph means & CI for 100 samples

Wider and narrower confidence intervals

- To compute a wider or narrower confidence interval, replace the 1.96 with the z-score for the interval of interest.
- The three most common intervals have the following z-scores:
  - 90% confidence interval z-score is 1.645
  - 95% confidence interval z-score is 1.96
  - 99% confidence interval z-score is 2.576

CI for small samples

- Confidence intervals for small samples, usually defined as samples with fewer than 30 observations, use a t-statistic instead of a z-statistic in computing confidence intervals for means and in other types of analyses.
- The t-statistic is from the t-distribution and, like the z-score, the t-statistic measures the distance from the mean.
- However, the t-statistic does this using the *standard deviation of the sampling distribution*, also known as the *standard error*, rather then the *standard deviation of the sample*.
- Specifically, the t-statistic is computed using the formula:

$$t = \frac{m}{\frac{s}{\sqrt{n}}}$$

- Where m is the sample mean, s is the sample standard deviation, and n is the sample size. Note that the denominator for m is $\frac{s}{\sqrt{n}}$, which is the standard error.
- The main practical difference between the two is that the t-statistic works better when samples are small; once samples are very large (n > 1000), the two values will be virtually identical.