STA130 W6

Monday, October 21, 2019 2:11 PM

P-value and error:

- **Rejecting** (i.e. a small p-value) does not guarantee that is false, nor that the results were not just due to chance. It **DOES mean that the result looked unusual** under the assumption that is true.
- **Not rejecting** (i.e. a large p-value) does n o t guarantee that is true, nor that the results are just due to chance. It **DOES mean that the result did not look unusual** under the assumption that is true.
- Sometimes p-values will be **small** even if H_0 is really **true**
- Sometimes p-values will be **large** even if H_0 is really **false**

If the p-value is small(close to 0), it means that the test stat **we observed** would be unusual, if H_0 were true
This provides evidence against H_0

Over-dependence on p-value:
School teach p-value -> journal editor and scientist use them -> school teach p-value

We estimate a characteristic of a population (a parameter) from incomplete observed data. What is a range of plausible values for what it could actually be?

Population: complete collection of individuals we are interested in (we don't have data for everyone)
Sample: subset of a population (for which we have data)
Goal: Use sample data to make **inference** about the population (i.e. learn something about the population)

So far: Null hypothesis gave us a model for the population
The value of the parameter in the population that we are interested in testing
Estimate the parameter instead of test a specific value:
Sample need to be **representative** of the population(randomly selected) **Also required for hypothesis testing**
Strategy: random selection (statistical theory tells us that on average, a random sample will be representative of the population)

2 main branches of statistical inference

Testing	Estimation
Hypothesis test: evaluate evidence against a particular value for parameter	**Confidence interval:** estimate of a parameter(gives range of plausible values of parameter) **Reasonable / probable**

Both based on
statistics: estimates of parameters from sample
Sampling distributions(or estimates of distribution) of statistics

*if you have all relevant data for each individual in the **population**(generally we don't), you can calculate **the true value of the parameter**
*Every random sample give **different** value of the statistic, but we hope that the values are **similar**(and close to the true parameter value)

Goal: Estimate the mean arrival delay in minutes (actual - scheduled time) of flights from NY to SF in 2013.
Population: all flights from New York to San Francisco (airport code SFO) in 2013

```
library(tidyverse)
library(nycflights13) Package with flights data
SF <- flights %>% filter(dest=="SFO" & !is.na(arr_delay)) With no missing value for arrival delay
```
Note: arr_delay is missing for 158 of the flights (so we exclude these)

```
SF %>% Population that we will be working with
summarise(
  mean_delay = mean(arr_delay),
  median_delay = median(arr_delay),
  max_delay = max(arr_delay))
```

```
## # A tibble: 1 x 3
##   mean_delay median_delay max_delay
##       <dbl>       <dbl>     <dbl>
## 1      2.67          -8      1007
```
Parameter, not statistic(because we are working with population)

```
population_mean <- SF %>%
  summarize(population_mean_delay = mean(arr_delay))
population_mean <-  as.numeric(population_mean) *save the value of mean_delay for later use
```

Suppose we only had data for **a** *sample of 25 flights* from New York to San Francisco.
The function **sample_n** from **dplyr** can be used to draw samples of any size (default is to sample without replacement)

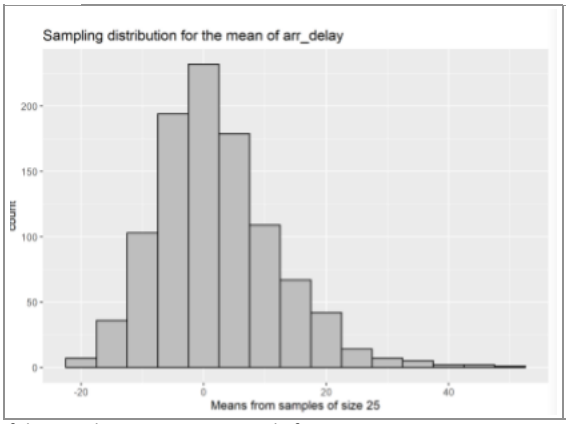
```
sample25 <- SF %>% sample_n(size=25, replace = FALSE)
```

*summary values for this sample will be **statistics**
***for other sample, we expect to see not the same statistics but similar value

Recall, the **sampling distribution** of the mean of arr_delay is the distribution of all the values that mean_delay could be for random samples of size n=25

```
set.seed(98)
sample_means <- rep(NA, 1000)
for(i in 1:1000){
 sample25 <- SF %>% sample_n(size=25) *Default: without replacement, guarantee that we get 25 different flights
       Extract certain number of rows but keep all columns
 sample_means[i] <- as.numeric(sample25 %>%
                               summarize(mean(arr_delay))) Where we store the means
}
sample_means <- data_frame(mean_delay = sample_means)

ggplot(sample_means, aes(x=mean_delay)) +
 geom_histogram(binwidth=5, color="black", fill="gray") +
 labs(x="Means from samples of size 25",
 title="Sampling distribution for the mean of arr_delay")
```



- Slightly right skewed
- Unimodal
- Centered around 0
- Spread? Variance/IQR?
  values from about -20 minutes to 45 minutes

If the sample size was 100 instead of 25:

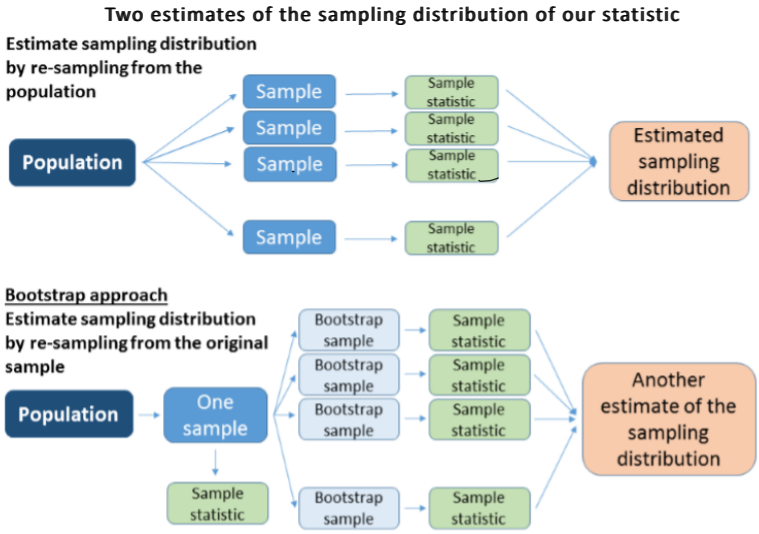
- Since we are getting samples from the same population, the **mean** of each sampling distribution will be **closer** to the population mean
- **Standard deviation** will be **smaller** because each estimate of mean delay based on a sample of size 100 should be closer(in general) to the true mean arrival delay
- The sampling distribution would be **more symmetrical**

If we only have 1 sample:
We do not have the full population

- Use our one sample(which we assume is representative of the population instead)

Resampling from the sample:(bootstrap samples):
Key assumption: our sample must be **representative** of the full population

- Drawn many random samples of size from **our original sample instead of the population** and each time calculated an estimate of the parameter



The bootstrap method
Start with a sample of size **n** from the population (we assume it is representative)
1. Draw many bootstrap samples of size **n** (i.e. **with replacement**) from the original sample
2. For each bootstrap sample, calculate the statistic

The distribution of the values of the statistic for all bootstrap samples is **the bootstrap sampling distribution** (this is another estimate of the sampling distribution of the statistic)
*does not create new data, but a tool to explore the **variability of estimates** from our original sample

Only observed one sample of size **200:**

```
observed_data <- SF %>%
  sample_n(size=200)
obs_mean <- observed_data %>%
 summarize(mean(arr_delay))
as.numeric(obs_mean)
## [1] 3.06
```

we need a lot of replications when bootstrapping
As with any simulations, the results are different every time
You can experiment with how many replications are needed to a relatively stable estimate (at least 1,000, but ideally 10,000+)
For this sample

```
boot_means <- rep(NA, 5000)
for(i in 1:5000){
  boot_samp <- observed_data %>% Original sample
       sample_n(size=200, replace=TRUE)
  boot_means[i] <-
    as.numeric(boot_samp %>%
            summarize(mean_delay =
mean(arr_delay)))
}
boot_means <- data_frame(mean_delay =
boot_means)
```

Observed_data: original sample of size n=200

Boot_simp: one bootstrap sample, of size n=200

Boot_means: Vector of results

Boot_means[i]: the mean arrival delay for one bootstrap sample
Boot_means[1]: the mean arrival delay for the first bootstrap sample

*We are treating the original sample as "population"

```
ggplot(boot_means, aes(x=mean_delay)) +
 geom_histogram(binwidth=1, fill="tan2", color="black") +
  labs(x="Means from bootstrap samples",
      title="Bootstrap sampling distribution for the mean arrival delay")
```

*The value of the mean of the bootstrap sampling distribution of mean arrival delay will be close to 3.06
*we still don't know the **true population mean**

Goal: Using only data from our sample, we want to make inferences about a range of values that would be plausible (i.e. believable) for the true population parameter, instead of just estimating one value

- look at the range of values that **the middle of the bootstrap distribution covers** (excluding values in the tails)

Percentiles (quantiles): an extension of quartiles
For a number between 0 and 100, the p^{th} percentile is the smallest value that is larger or equal to p% of all the values
Median(Q_2): 50th percentile First quartile
(Q_1): 25th percentile Third quartile
(Q_3): 75th percentile

Use the **quantile()** function in R to calculate these

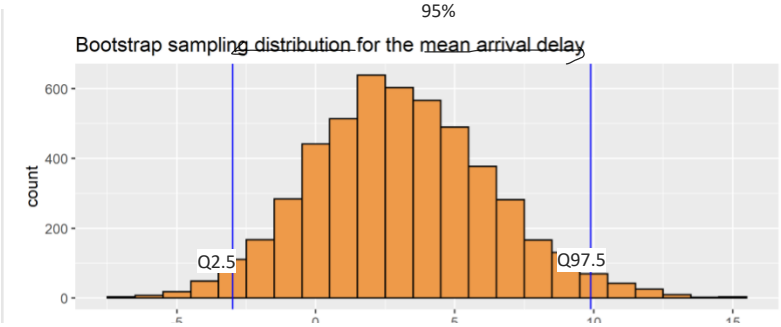
```
                                  Quantiles you want to calculate
quantile(boot_means$mean_delay, c(0.25, 0.5, 0.75)) Vector of values
##     25%     50%     75%
## 0.83875 2.91000 5.18000
quantile(boot_means$mean_delay, c(0.025, 0.4, 0.57)) *other quantile
##      2.5%       40%       57%
## -2.995125  2.153000  3.520000
2.5% of points are smaller than this value
```

2.5th and 97.5th percentiles:

```
quantile(boot_means$mean_delay, c(0.025,
0.975))
##     2.5%    97.5%
## -2.995125  9.900250
```

True population mean:

```
as.numeric(population_mean)
## [1] 2.672892
```



Means from bootstrap samples

The interval that is the middle 95% of our bootstrap distribution(95% confidence interval) is **(−3.0,9.90)**

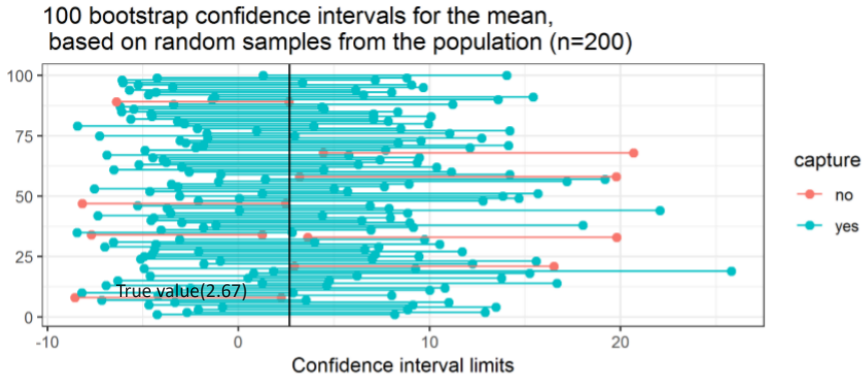
- **include our true population mean**
- **it does not always include**

How often does this procedure give an interval that captures the population mean?
We have this for the flights data
Let's use simulation!!
Suppose:

- We **know** the true population mean
- We can **get many samples** from the population

Then to see how often an interval calculated in this way would include the true population mean, we:
1. Randomly draw another sample of size 200 from the population
2. Find the bootstrap sampling distribution of the mean from many (ex: 5000) replications of bootstrap samples of this new data.
3. Find the interval that is the middle 95% of the bootstrap distribution.
4. Repeat steps 1-3 100 times.
5. Check how many of these intervals include the true population mean
   The bootstrap approach in practice(only do it once)

Statistical theory says that these intervals should capture the population mean 95% of the time. These are called 95% **confidence intervals** for the mean.



92 out o f 100 CIs contain the true parameter value
In the long run, we would see that **95%** of the CIs would contain the true value

Confidence Intervals:
A 95% **confidence interval** for a population parameter is calculated from sample data in such a way that the interval will include the parameter for 95% of possible samples. Here, 95% is the **confidence level**

Each of the CIs on the previous slide gives a range of plausible values (i.e. believable) for what the true parameter might be, based on the incomplete and imperfect information we have in each sample from the population.

- A "good" interval captures the population mean
- We used the middle 95% of the bootstrap sampling distribution to get each CI, so 95% of them should be "good"

We are **95% confident** that the mean arrival delay for all flights from NY to SF in 2013 is between -3 (i.e. 3 minutes early) and 9.9 minutes (i.e. 9.9 minutes late).
It is how we are confident with the method

In this case, we know that nu=2.67 minutes, so the interval we computed does contain .
However, in practice we don't know, and this is why we are calculating a confidence interval the first place.
In general we don’t know a specific interval contains the true value of the parameter, but we can say that we are confident that it does because our method guarantees that most intervals constructed in this way contain the true value

0. start with one sample of size n from the population
1. Take a bootstrap sample of the data by sampling with replacement, the same number of observations as the original data.(n)
2. For the bootstrap sample, calculate the statistic that estimates the parameter you are interested in.
   We could do this for any statistic
3. Repeat steps 1 and 2 many times to get a distribution of bootstrap statistics.
4. A 95% confidence interval for the parameter is the middle 95% of values of the bootstrap statistics.

P-value	**CI**
Hypothesis testing	provide **variation** for estimate

Same: sampling distribution, statistical significance

Increase confidence level, **decrease** type 1 error
Keep the same value: set seebhd & variation of the data

There is a 95% **chance** that between 56% and 73% of all kissing couples in the population tilt their head to the right when they kiss.
Wrong! Chance is different from confident level, it is either 0 or 100 - either in the interval or not in the interval

If we considered many random samples of 124 couples, and we calculated 95% confidence intervals for each sample, 95% of these confidence intervals will include the true proportion of kissing couples in the population who tilt their heads to the right when they kiss.
Yes.

```
quantile(boot_means$mean_delay,
         c(0.005, 0.01, 0.025, 0.05, 0.1,
          0.9, 0.95, 0.975, 0.99, 0.995))
##     0.5%       1%     2.5%       5%      10%      90%      95%
## -4.725050 -3.910100 -2.995125 -2.085000 -0.955500  7.245000  8.715000
##    97.5%      99%    99.5%
##  9.900250 11.205500 12.030150
```

*80% confidence interval: (-0.96, 7.25)
*95% confidence interval: (-3, 9.9)
*99% confidence interval: (-4.73, 12.03)
Wider Confidence interval, higher confidence level - less accurate, may get very different values from each simulation

If the sample is biased **(i.e. not representative of the population),** the bootstrap CI will also be biased
larger samples reflect the population better - the bootstrap may work poorly when you start with a small sample

Can the bootstrap give us **better** estimate of the population parameter
No! The bootstrap **doesn't create new data**, it only re-uses our sample data

Purpose of bootstrap: to get a confidence interval (CI)
Purpose of confidence intervals: to assess the variability of our estimate (get a range of plausible values)

The confidence interval method we've used is the **percentile bootstrap method** .

- There are other bootstrap methods that are more robust, that is they are better at capturing the parameter the correct percentage of the time.
- The percentile bootstrap method works best for **large** samples and when the bootstrap distribution is approximately **symmetric** and **continuous**.
- You may see other versions of the bootstrap method in future statistics courses