How Do We Perform Estimation?

1. Define the population of interest.
2. Select the right sampling method according to the specific characteristics of our population of interest.
3. Select our sample size (Power Analysis).
4. Collect the sampled data.
5. Measure and calculate the sample statistic.
6. Infer the population value based on this sample statistic while accounting for sampling uncertainty.

Population Proportion P_E

```
listings |>
  group_by(room_type) |>
  summarise(n = n()) |>
  mutate(freq = round(n / sum(n), 3))
```

```
library(infer)
# Sampling WITHOUT replacement is the default
sample_1 <- rep_sample_n(listings, size = 40)
#This gives us 100 random samples of size n = 40 from the
    population
samples_100 <- rep_sample_n(listings, size = 40, reps =
    100)
sampling_dist <- samples_100 |>
  group_by(replicate) |>
  summarise(
    n_E = sum(room_type == "Entire home/apt"),
    p_hat_E = sum(room_type == "Entire home/apt") / 40
  )
#Mean of sampling dist
mean_sampling_dist <- sampling_dist |>
    pull(p_hat_E)|>
    mean() |>
    round(3)
#Histogram plot of sampling_dist_plot
sampling_dist_plot <- sampling_dist |>
  ggplot(aes(x = p_hat_E)) +
  geom_histogram(binwidth = 0.025) +
  xlab("Sample Proportion Based on n = 40") +
  ggtitle("Sampling Dist of the Sample Proportions") +
  theme(text = element_text(size = 16.5)) +
  scale_x_continuous(breaks = seq(0.4, 1, 0.05)) +
  geom_vline(xintercept = 0.756, color = "red") # Mean
      proportion estimate as a vertical red line
  summary(df$column) #Get summary of column
```

Why Sampling Distributions?

Sampling is usually costly in terms of human, monetary, and time resources.
There might be some ethical implications in a given inferential/causal study.
Since our sample statistic is also a random variable, it will have some variability associated when estimating a population parameter.
The sample distribution and the sampling distribution of the sample estimate are not the same.
They also do not relate to the population in the same manner. Mean of Population = Multiple samples mean

```
  multiple_samples_n50 <- rep_sample_n(listings, size = 50,
      reps = 1000)
  multiple_samples_mean_price_n50 <- multiple_samples_n50
      |>
  summarise(sample_mean = mean(price))
```

The distribution of one_sample looks somewhat like the distribution of the population.
Sample mean is not exactly equal to the population mean.
The shape of the sampling distribution from multiple_samples_n50 is different from the shape of the population distribution and the sample distribution.

Sampling Distributions and their Relationship to Sample Size n

As we increase the sample size n,
- our sampling distribution gets narrower.
- the standard error gets smaller.
- each sample is more likely to have an estimate closer to the true population parameter we are trying to estimate (compared to samples with a smaller number of observations).

Quantifying the variability/uncertainty around our point estimate vary from sample to sample.Two ways to quantify

1. Via computation: We can use bootstrapping. This approach is pretty flexible since it can be applied to different statistics such as the sample mean, median, a given quantile, etc.
2. Via a theoretical shortcut: Central Limit Theorem (CLT).

Bootstrapping

A random sample taken with replacement from the original sample of the same size n.
Calculate the bootstrap point estimate from that bootstrap sample.
Do this multiple reps and calculate the point estimates for all the reps (empirical confidence interval)

```
  six_bootstrap_samples <- one_sample |>
  rep_sample_n(size = 50, replace = TRUE, reps = 6)
```

There is a bell shape in both sampling distributions. Moreover, we could state that the spread is graphically similar, except for some outliers on the right-hand side for the regular sampling distribution. This makes the distribution slightly right-skewed.
The means of these two distributions are different: the mean of the bootstrap sample means is almost exactly that of the original one_sample mean, whereas the mean of the sampling distribution of sample means is almost exactly that of the population parameter.

Confidence Intervals

No, confidence intervals are not done only on bootstrapped samples. If you take a larger sample size n, your confidence interval at a specified level will narrower. If we report a point estimate, we probably will not hit the exact value of the population parameter. If we report a range of plausible values, we have a good shot at capturing the parameter. 95% of the time, we would expect our population parameter's value to lie within the confidence interval. One way to calculate a range of plausible values for the population parameter 95 percentile.
Two samples of the same size are drawn from the same population, a 95% confidence interval from sample A will always be wider than a 90% confidence interval from sample B? True, but not always - it depends on the variability in the observations of the samples.

1. Our endpoints are at the 2.5th and 97.5th percentiles.
2. For the bootstrap distribution below, the values of 21 and 29.3

```
  bootstrap_dist <- df |>
    specify(response = col) |>
    generate(reps = 10000, type = "bootstrap") |>
    calculate(stat = "median")
  ci <- bootstrap_dist |>
    get_confidence_interval(level = 0.90, type = "percentile
        ")
```

Questions we can answer with hypothesis testing

hypothesis tests are necessary for many important Data Science-related inquiries (mostly inferential or causal, and sometimes even predictive!)

1. Null hypothesis (H_o): The status quo. It is usually a claim that there really is "no effect" or "no difference".
2. Alternative hypothesis (H_A): Our hypothesis of interest. It is the claim for which we seek significant statistical evidence.

Hypotheses in A/B Testing

Let us revisit the A/B testing question: will changing the design of the website lead to a change in customer engagement (measured by the CTRs)?

1. Null hypothesis The population CTRs for the two versions of the website are equal
2. Alternative hypothesis The population CTRs for the two versions of the website are NOT equal.

The Hypothesis Testing Framework

Our testing procedure is developed under a null framework, i.e., the one corresponding to H_O
Provided a strong enough statistical evidence via our random sample drawn from the population of interest, we can reject H_O in favour of H_A

How to assess the sample's statistical evidence?

Create a model of what we would expect under the null hypothesis, H_O
Define a test statistic that corresponds to our model of H_O
Since we are working under a null framework, we assume that ANY observed difference between both treatments will only happen due to chance.
Therefore, any permuted relabeling WOULD BE as similar as our real observed δ. This method is called permutation (sampling WITHOUT replacement).

Six Steps of Hypothesis Testing/ A-B Testing

1. Define your null and alternative hypotheses.
2. Compute the **observed** test statistic $\hat{\delta}$ coming from your original sample.
3. Use the null model to generate **r random permuted** samples from the original sample and calculate their corresponding r test statistics.
4. Generate the null distribution using these r test statistics.
5. Check where your **observed** test statistic $\hat{\delta}$ falls on this distribution.
6. If $\hat{\delta}$ is near the extremes past some threshold defined with a significance level α (i.e., p-value is less than α), we reject the null hypothesis. Otherwise, we fail to reject the null hypothesis.

```
ci_mean <- function(sample, var, level = 0.95, type = "
    percentile") {
  sample |>
  rep_sample_n(nrow(sample), replace = TRUE, reps = 15000)
      |>
  summarise(stat = mean({{ var }})) |>
  get_confidence_interval(level = level, type = type)
}
ownership_means <- ownership_data |>
  group_by(ownership) |>
  nest() |>
  mutate(
  ci = map(data, ~ ci_mean(.x, percent_score)),
  mean = map_dbl(data, ~ mean(.x$percent_score))
  ) |>
  unnest(ci)
ownership_means <- ownership_data |>
  group_by(ownership) |>
  summarise(mean = mean(percent_score, na.rm = TRUE))
public_mean <- ownership_means |> filter(ownership == "
    Public") |> pull(mean)
private_mean <- ownership_means |> filter(ownership == "
    Private") |> pull(mean)
delta_star <- public_mean - private_mean
null_distribution_ownership <- ownership_data |>
  specify(formula = percent_score ~ ownership) |>
  hypothesize(null = "independence") |>
  generate(1000, type = "permute") |>
  calculate(stat = 'diff in means', order = c('Public','
      Private'))
#Since alpha is 0.05
threshold <- quantile(null_distribution_ownership$stat, c
    (0.025, 0.975))
#p-value
ownership_pvalue <- null_distribution_ownership |>
  get_p_value(obs_stat = delta_star, direction = "two-sided
      ")
```