Statistical Power and ANOVA - Introduction

Introduction

In this section you'll continue to deepen your knowledge of hypothesis testing and t-tests by examining the concept of power; an idea closely related to type II errors. With that, you'll see how the rate of type I errors, power, sample size, and effect size are intrinsically related to one another. You will then move on to ANOVA - Analysis of Variance, which allows you to test for the influence of multiple factors all at once.

Statistical power

Statistical power is equal to $1 - \beta$ where β is the rate of type II errors. As you will see, power is related to α, sample size, and effect size. Typically a researcher will select an acceptable alpha value and then examine required sample sizes to achieve the desired power such as 0.8 (or higher).

Welch's t-test

After an initial exploration of statistical power, you'll take a look at Welch's t-test. This is an adaptation of the unpaired student's t-test you've seen previously which allows for different sample sizes or different variances between the two groups.

Multiple comparisons

From there, you'll look at some of the issues that arise when trying to perform multiple comparisons - from the risks of spurious correlations to the importance of corrections such as the Bonferroni correction to deal with the cumulative risks of type I errors inherent in multiple comparisons.

ANOVA

Finally, you'll take a look at the more generalized procedure for conducting multiple comparisons: Analysis of Variance or ANOVA. You'll see that ANOVA of only two groups is statistically equivalent to a two sided t-test. That said, ANOVA fully supports comparing multiple factors simultaneously.

Summary

Without a good understanding of experimental design, it's easy to end up drawing false conclusions. In this section, you'll cover a range of tools and techniques to deepen your understanding of hypothesis testing and ensure that you design experiments rigorously and interpret them thoughtfully.

Statistical Power

Introduction

You've started to investigate hypothesis testing, p-values and their use for accepting or rejecting the null hypothesis. With this, the power of a statistical test measures an experiment's ability to detect a difference, when one exists. In the case of testing whether a coin is fair, the power of our statistical test would be the probability of rejecting the null hypothesis "this coin is fair" when the coin was unfair. As you might assume, the power of this statistical test would thus depend on several factors including our p-value threshold for rejecting the null hypothesis, the size of our sample and the 'level of unfairness' of the coin in question.

Objectives

You will be able to:

- Define power in relation to p-value and the null hypothesis
- Describe the impact of sample size and effect size on power
- Perform power calculation using SciPy and Python
- Demonstrate the combined effect of sample size and effect size on statistical power using simulations

The power of a statistical test

The power of a statistical test is defined as the probability of rejecting the null hypothesis, given that it is indeed false. As with any probability, the power of a statistical test, therefore, ranges from 0 to 1, with 1 being a perfect test that guarantees rejecting the null hypothesis when it is indeed false.

Intrinsically, this is related to β, the probability of type II errors. When designing a statistical test, a researcher will typically determine an acceptable α, such as .05, the probability of type I errors. (Recall that type I errors are when the null-hypothesis is rejected when actually true.) From this given α value, an optimal threshold for rejecting the null-hypothesis can be determined. That is, for a given α value, you can calculate a threshold that maximizes the power of the test. For any given α, $power = 1 - \beta$.

> Note: Ideally, α and β would both be minimized, but this is often costly, impractical or impossible depending on the scenario and required sample sizes.

Effect size

The effect size is the magnitude of the difference you are testing between the two groups. Thus far, you've mainly been investigating the mean of a sample. For example, after flipping a coin n number of times, you've investigated using a t-test to determine whether the coin is a fair coin (p(heads)=0.5). To do this, you compared the mean of the sample to that of another sample, if comparing coins, or to a know theoretical distribution. Similarly, you might compare the mean income of a sample population to that of a census tract to determine if the populations are statistically different. In such cases, Cohen's D is typically the metric used as the effect size.

Cohen's D is defined as: $d = \frac{m_1 - m_2}{s}$, where m_1 and m_2 are the respective sample means and s is the overall standard deviation of the samples.

> When looking at the difference of means of two populations, Cohen's D is equal to the difference of the sample means divided by the pooled standard deviation of the samples. The pooled standard deviation of the samples is the average spread of all data points in the two samples around their group mean.

Power analysis

Since α, power, sample size, and effect size are all related quantities, you can take a look at some plots of the power of some t-tests, given varying sample sizes. This will allow you to develop a deeper understanding of how these quantities are related and what constitutes a convincing statistical test. There are three things to go into the calculation of power for a test. They are:

- alpha value
- effect size
- sample size

A fantastic visual representation of these values' effect on one another can be found on Kristoffer Magnusson's website.

Let's look at how power might change in the context of varying effect size. To start, imagine the scenario of trying to detect whether or not a coin is fair. In this scenario, the null-hypothesis would be $H_0(heads) = 0.5$ because our assumption is that we are dealing with a fair coin. From here, the power will depend on both the sample size and the effect size (that is the threshold for the null hypothesis to be rejected). For example, if the alternative hypothesis has a large margin from the null-hypothesis such as $H_a(heads) = 0.8$ or $H_a(heads) = 0.9$ (large effect size), then there is a higher chance of rejecting the null-hypothesis (power is increased). If there is a smaller margin between the null hypothesis and an alternate hypothesis, an unfair coin where $P(heads) = .6$ for example (small effect size), there is a lower chance of rejecting the null hypothesis (power is reduced).

To start, you might choose an alpha value that you are willing to accept such as $\alpha = 0.05$. From there, you can observe the power of various statistical tests against various sample and effect sizes.

For example, if we wish to state the alternative hypothesis $H_a = .55$, then the effect size (using Cohen's D) would be:

$$d = \frac{m_1 - m_2}{s}$$

$$d = \frac{.55 - .5}{s}$$

Furthermore, since we are dealing with a binomial variable, the standard deviation of the sample should follow the formula $\sqrt{n \cdot p(1-p)}$.
So some potential effect size values for various scenarios might look like this:

In [1]:

```python
import numpy as np
import pandas as pd
```

In [2]:

```python
m1 = .55
m2 = .5
p = m2
rows = []
for n in [10, 20, 50, 500]:
    std = np.sqrt(n*p*(1-p))
    d = (m1-m2)/std
    rows.append({'Effect_Size': d, 'STD': std, 'Num_observations': n})
print('Hypothetical effect sizes for p(heads)=.55 vs p(heads)=.5')
pd.DataFrame(rows)
```

```
Hypothetical effect sizes for p(heads)=.55 vs p(heads)=.5
```

Out[2]:

	Effect_Size	STD	Num_observations
0	0.031623	1.581139	10
1	0.022361	2.236068	20
2	0.014142	3.535534	50
3	0.004472	11.180340	500

As a general rule of thumb, all of these effect sizes are quite small. here's the same idea expanded to other alternative hypotheses:

In [3]:

```python
m2 = .5
rows = {}
for n in [10, 20, 50, 500]:
    temp_dict = {}
    for m1 in [.51, .55, .6, .65, .7, .75, .8, .85, .9]:
        p = m1
        std = np.sqrt(n*p*(1-p))
        d = (m1-m2)/std
        temp_dict[m1] = d
    rows[n] = temp_dict
print('Hypothetical effect sizes for various alternative hypotheses')
df = pd.DataFrame.from_dict(rows, orient='index')
# df.index = [10,20,50, 500]
# df.index.name = 'Sample_Size'
# df.columns.name = 'Alternative Hypothesis'
df
```

```
Hypothetical effect sizes for various alternative hypotheses
```

Out[3]:

	0.51	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
10	0.006326	0.031782	0.064550	0.099449	0.138013	0.182574	0.237171	0.309965	0.421637
20	0.004473	0.022473	0.045644	0.070321	0.097590	0.129099	0.167705	0.219179	0.298142
50	0.002829	0.014213	0.028868	0.044475	0.061721	0.081650	0.106066	0.138621	0.188562
500	0.000895	0.004495	0.009129	0.014064	0.019518	0.025820	0.033541	0.043836	0.059628

While a bit long winded, you can see that realistic effect sizes for this scenario could be anywhere from 0.05 (or lower) up to approximately .4.

Now that you have some parameter estimates for α and the effect size, you can map subsequent relationships for the power and sample size. Again, this is because any three of these quantities (alpha, effect size, sample size and power) will determine the fourth.

As you've also seen, a common statistical test for comparing sample means is the t-test. Statsmodels has some convenient build in functions for calculating the power of a t-test and plotting power curves. Take a look:

In [4]:

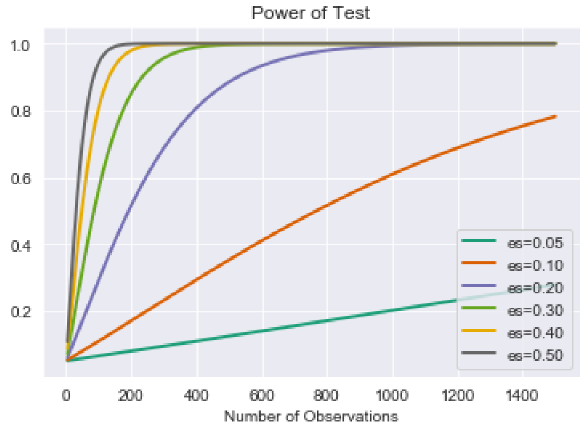
```python
from statsmodels.stats.power import TTestIndPower, TTestPower
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style('darkgrid') # Nice background styling on plots
```

In [5]:

```python
power_analysis = TTestIndPower()
```

In [6]:

```python
power_analysis.plot_power(dep_var='nobs',
                          nobs = np.array(range(5,1500)),
                          effect_size=np.array([.05, .1, .2,.3,.4,.5]),
                          alpha=0.05)
plt.show()
```



As this should demonstrate, detecting small perturbances can be quite difficult!

Similarly, just because a t-test has an incredibly small p-value doesn't necessarily imply a strong statistical test. As is mentioned in the article *Using Effect Size - or Why the P Value Is Not Enough*, referenced below, using incredibly large sample sizes such as 22,000 can make even the most trivial effect size statistically significant. Realizing these reciprocal relationships and considering all 4 parameters: alpha, effect size, sample size, and power are all important when interpreting the results (such as the p-value) of a statistical test.

In addition to plotting a full curve, you can also calculate specific values. Simply don't specify one of the four parameters.

In [7]:

```python
# Calculate power
power_analysis.power(effect_size=.2, nobs1=80, alpha=.05)
power_analysis.solve_power(effect_size=.2, nobs1=80, alpha=.05)
```

Out[7]:

```
0.24175778678474308
```

In [8]:

```
# Calculate sample size required
power_analysis.solve_power(effect_size=.2, alpha=.05, power=.8)
```

Out[8]:

```
393.4056989990322
```

In [9]:

```
# Calculate minimum effect size to satisfy desired alpha and power as well as respect sample size limitations
power_analysis.solve_power(nobs1=25, alpha=.05, power=.8)
```

Out[9]:

```
0.8087077886680407
```

In [10]:

```
# Calculate alpha (less traditional)
power_analysis.solve_power(nobs1=25, effect_size=.3, power=.8)
```

Out[10]:

```
0.661363427343157
```

You can also simulate your own data to verify results:

In [11]:

```
import scipy.stats as stats
def run_ttest_sim(p1, p2, std, nobs, alpha=0.05, n_sim=10**5):
    """p1 and p2 are the underlying means probabilities for 2 normal variables
    Samples will be generated using these parameters."""
    # Calculate Normalized Effect Size
    effect_size = np.abs(p1-p2)/std

    # Run a Simulation
    # Initialize array to store results
    p = (np.empty(n_sim))
    p.fill(np.nan)

    #  Run a for loop for range of values in n_sim
    for s in range(n_sim):
        control = np.random.normal(loc= p1, scale=std, size=nobs)
        experimental = np.random.normal(loc= p2, scale=std, size=nobs)
        t_test = stats.ttest_ind(control, experimental)
        p[s] = t_test[1]

    num_null_rejects = np.sum(p < alpha)
    power = num_null_rejects/n_sim
    # Store results
    stat_dict = {'alpha':alpha,
                 'nobs':nobs,
                 'effect_size':effect_size,
                 'power': power}
    return stat_dict

run_ttest_sim(.5, .7, 1, 50)
```

Out[11]:

```
{'alpha': 0.05,
 'nobs': 50,
 'effect_size': 0.19999999999999996,
 'power': 0.16877}
```

And going back to the full stats model implementation for verification:

In [12]:

```
power_analysis.solve_power(nobs1=50, effect_size=0.19999999999999996, alpha=0.05)
```

Out[12]:

```
0.16767548634547413
```

In [13]:

```
power_analysis.solve_power(nobs1=50, effect_size=0.19999999999999996, power=0.16719)
```

Out[13]:

```
0.049779515826212685
```

In [14]:

```
power_analysis.solve_power(nobs1=50, power=0.16719, alpha=0.05)
```

Out[14]:

```
0.19959710069445363
```

In [15]:

```
power_analysis.solve_power(power=0.16719, effect_size=0.19999999999999996, alpha=0.05)
```

Out[15]:

```
49.803133138534754
```

Additional Resources

- Statsmodels documentation
- Using Effect Size—or Why the P Value Is Not Enough
- Understanding Statistical Power and Significance Testing - an interactive visualization

Summary

In this lesson, you learned about the idea of "statistical power" and how sample size, alpha, and effect size impact the power of an experiment. Remember, the power of a statistical test is the probability of rejecting the null hypothesis when it is indeed false.

Welch's T-Test

Introduction

Thus far, you've seen the traditional Student's t-test for hypothesis testing between two sample means. Recall that z-tests are also appropriate for statistics, such as the mean, which can be assumed to be normally distributed. However, when sample sizes are low (n_observations < 30), the t-test is more appropriate, as the t-distribution has heavier tails. Even with this modification, remember that there are still several assumptions to the model. Most notably, traditional t-tests assume that sample sizes and sample variances between the two groups are equal. When these assumptions are not met, Welch's t-test is generally a more reliable test.

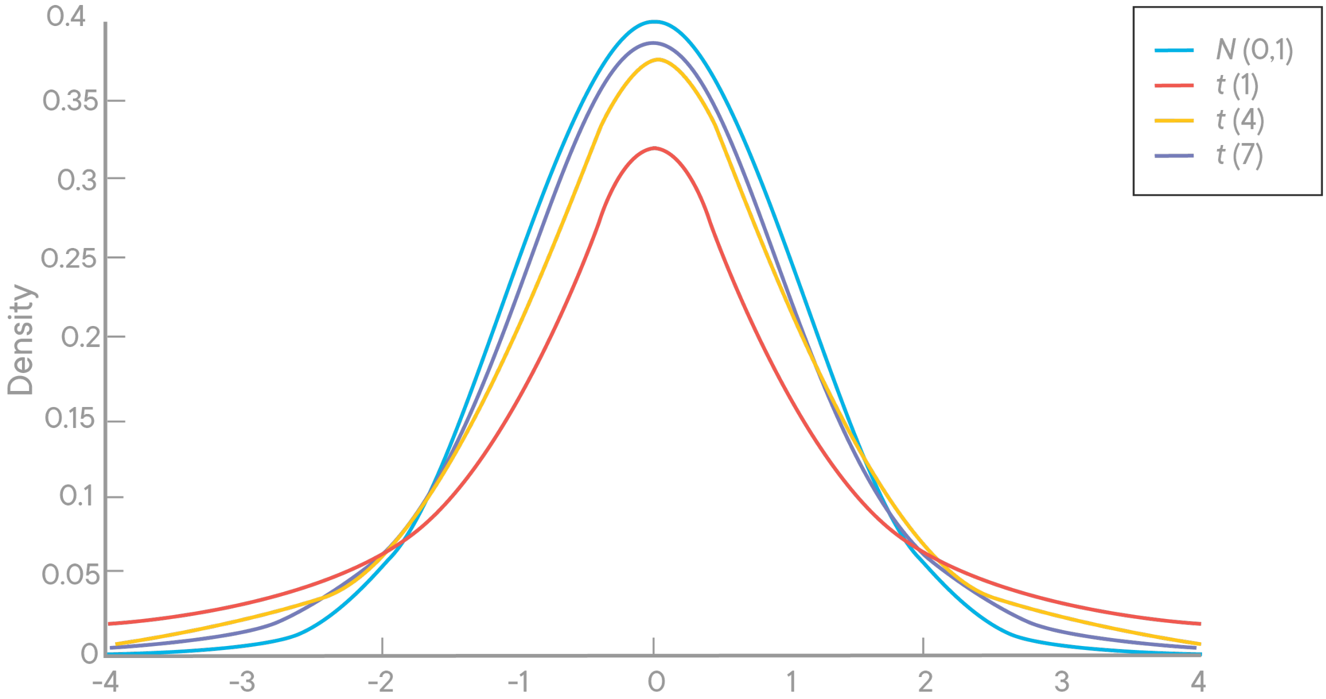
Objectives

You will be able to:

- List the conditions needed to require a Welch's t-test
- Calculate the degrees of freedom for a Welch's t-test
- Calculate p-values using Welch's t-test

T-test review

Recall that t-tests are a useful method for determining whether the mean of two small samples indicate different underlying population parameters. The reasoning behind this begins with the use of z-tests to calculate the likelihood of sampling a particular value from a normal distribution. Furthermore, by the central limit theorem, the mean of a sample is a normally distributed variable centered around the actual underlying population mean. That said, t-tests are more appropriate for small samples (n_observations < 30), due to disproportionate tails. Finally, recall that the t-distribution actually converges to a normal distribution as the degrees of freedom continues to increase.



> A normal distribution vs. t-distributions with varying degrees of freedom. Note how the t-distribution approaches the normal distribution as the degrees of freedom increases. Recall that when performing a two-sample t-test, assuming that sample variances are equal, the degrees of freedom equals the total number of observations in the samples minus two.

Welch's t-test

Just as Student's t-test is a useful adaptation of the normal distribution which can lead to better likelihood estimates under certain conditions, the Welch's t-test is a further adaptation that accounts for additional perturbations in the underlying assumptions of the model. Specifically, the Student's t-test assumes that the samples are of equal size and equal variance. When these assumptions are not met, then Welch's t-test provides a more accurate p-value.

Here is how you calculate it:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{se_1^2 + se_2^2}}$$ where

- $\bar{X}_i$ - mean of sample i
- s_i^2 - variance of sample i
- N_i - sample size of sample i

The modification is related to the **degrees of freedom** in the t-test, which tends to increase the test power for samples with unequal variance. When two groups have equal sample sizes and variances, Welch’s t-test tends to give the same result as the Student’s t-test. However, when sample sizes and variances are unequal, Student’s t-test is quite unreliable, whereas Welch’s tends perform better.

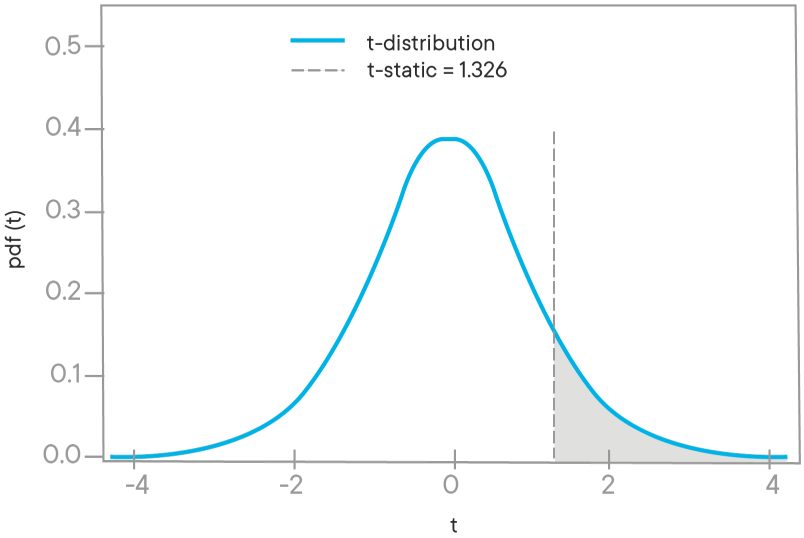
Calculate the degrees of freedom

Once the t-score has been calculated for the experiment using the above formula, you then must calculate the degrees of freedom for the t-distribution. Under the two-sample Student's t-test, this is simply the total number of observations in the samples size minus two, but given that the sample sizes may vary using the Welch's t-test, the calculation is a bit more complex:

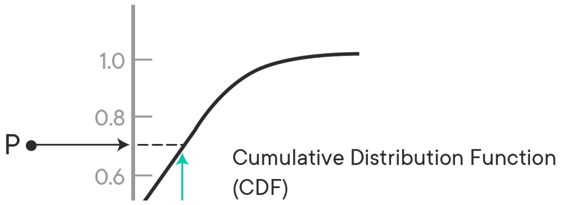
$$\nu \approx \frac{\left(\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}\right)^2}{\frac{s_1^4}{N_1^2 \nu_1} + \frac{s_2^4}{N_2^2 \nu_2}}$$

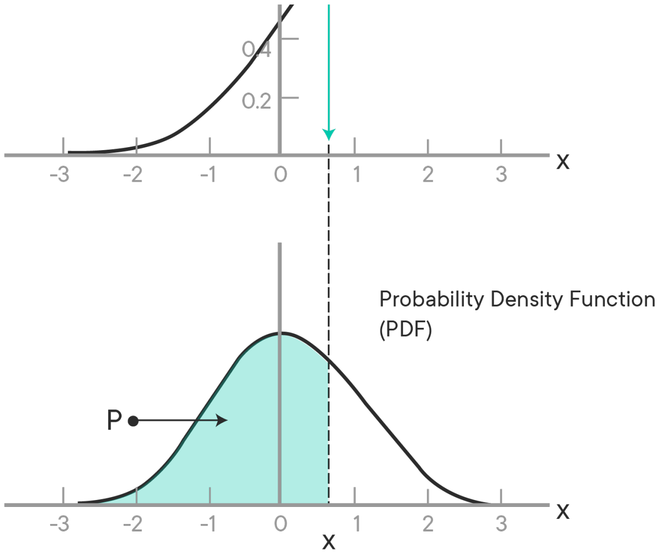
Calculate p-values

Finally, as with the Student's t-test (or a z-test for that matter), you convert the calculated score into a p-value in order to confirm or reject the null-hypothesis of your statistical experiment. For example, you might be using a one-sided t-test to determine whether a new drug had a positive effect on patient outcomes. The p-value for the experiment is equivalent to the area under the t-distribution with the degrees of freedom, as calculated above, and the corresponding t-score.



The easiest method for determining said p-values is to use the `.cdf()` method from `scipy.stats` to find the complement and subtracting this from 1.





Relations Between Two Different Typical Representations of a Population

Here's the relevant code snippet:

```
import scipy.stats as stats


p = 1 - stats.t.cdf(t, df)
```

Summary

This lesson briefly introduced you to another statistical test for comparing the means of two samples: Welch's t-test. Remember that when your samples are not of equal size or do not have equal variances, it is a more appropriate statistical test than the Student's t-test!

Effect Size, P-Values and Power - Lab

Introduction

In this lab, you'll run simulations to continue to investigate the relationship between effect size, p-values, power, and sample size!

Objectives

You will be able to:

- Run a simulation that creates a visualization to demonstrate the interaction between power, sample size, and effect size

Philosophical review

Remember that the underlying question behind all hypothesis tests is:

> "What is the probability I would see this effect due to random fluctuations if there was actually no effect?"

This is exactly what a p-value represents: the chance that the observed data would satisfy the null hypothesis. As such, if the p-value is sufficiently low, you can declare the results statistically significant and reject the null hypothesis. Recall that this threshold is defined as α, and is also the rate of type I errors. In this lab, you'll investigate the robustness of p-values and their relation with effect-size and sample-size.

Import starter functions

To start, import the functions stored in the `flatiron_stats.py` file. It contains the stats functions that you previously coded in the last lab: `welch_t(a,b)`, `welch_df(a, b)`, and `p_value(a, b, two_sided=False)`. You'll then use these functions below to further investigate the relationship between p-values and sample size.

In [1]:

```
# Your code here; import the contents from flatiron_stats.py
# You may also wish to open up flatiron_stats.py in a text editor to preview its contents
```

Generate random samples

Before you start running simulations, it will be useful to have a helper function that will generate random samples. Write such a function below which generates 2 random samples from 2 normal distributions. The function should take 6 input parameters:

- m1 - The underlying population mean for sample 1
- s1 - The underlying population standard deviation for sample 1
- n1 - The sample size for sample 1
- m2 - The underlying population mean for sample 2
- s2 - The underlying population standard deviation for sample 2
- n2 - The sample size for sample 2

In [2]:

```
import numpy as np
def generate_samples(m1, s1, n1, m2, s2, n2):
    # Your code here; have the function create two random samples using the input parameters
    return sample1, sample2
```

Run a simulation

For your first simulation, you're going to investigate how the p-value of an experiment relates to sample size when both samples are from identical underlying distributions. To do this, use your `generate_samples()` function along with the `p_value_welch_ttest()` function defined in the `flatiron_stats` file. Use sample sizes from 5 to 750. For each sample size, simulate 100 experiments. For each of these experiments, generate 2 samples of the given sample size. Each sample should have a standard deviation of 1. The first sample should have a mean of 5 and the second should have a mean of 5 plus the effect size, you hope to detect. Calculate the corresponding p-values for a Welch's t-test for each of these sample pairs. Finally, use the p-values to calculate the power of the test. Remember that for all of the simulations where the effect size does not equal zero, the null hypothesis is not true. Store the overall power from the 100 simulations along with the corresponding sample size and effect size. Use varying effect sizes such as [0, 0.01, 0.1, 0.2, 0.5, 1, 2]. You'll then plot power vs sample size for various effect sizes.

In [3]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style('darkgrid')
```

In [4]:

```
# Your code here

# Pseudo code outline
# for effect size:
#     for sample_size:
#         perform 100 simulations
#         calculate power
#         store effect_size, sample_size, power for simulations
```

Now that you've simulated the data, go ahead and graph it! Label the x-axis sample size, the y-axis power, and be sure to include a legend for the various effect sizes.

In [5]:

```
# Your code here
```

As you can see, it's also typically incredibly difficult (if not impossible) to accurately detect effect sizes below 0.1!

Summary

This lesson summarizes and further builds upon the ideas that we saw in the previous labs. We learned how p-value can be described as a function of effect size and for a given effect size, the p-value may get lower if we increase the sample size considerably. We also saw how p-value alone can not be used in order to identify some results as truly significant, as this can be achieved when there is not a significant effect size.

The Multiple Comparisons Problem

Introduction

In this lesson, you'll learn about the problems that can arise from doing multiple comparisons in a single experiment.

Objectives

You will be able to:

- Explain why multiple comparisons increases the likelihood of misleading results
- Explain the concept of spurious correlation
- Use corrections to deal with multiple comparison problems

What is the multiple comparisons problem?

Obtaining an incredibly low p-value does not guarantee that the null-hypothesis is incorrect. For example, a p-value of 0.001 states that there is still a 1 in 1000 chance that the null hypothesis is true. Yet, as you've seen, p-values alone can be misleading. For example, if you perform repeated experiments, at some point you're apt to stumble upon a small p-value, whether or not the null hypothesis is valid.

To restate this, imagine we take 100 scientific studies with a p-value of 0.03. Are all of these conclusions valid? Sadly, probably not. Remember, for any experiment with a p-value of 0.03, there is still a 3% chance that the null-hypothesis is actually true. So collectively, the probability that **all** of these null hypotheses are false is actually quite small. You can be fairly confident in each study, but there is also apt to be a false-conclusion drawn somewhere. (In fact, the p-value itself implies that, on average, 3 of these 100 conclusions will be false.)

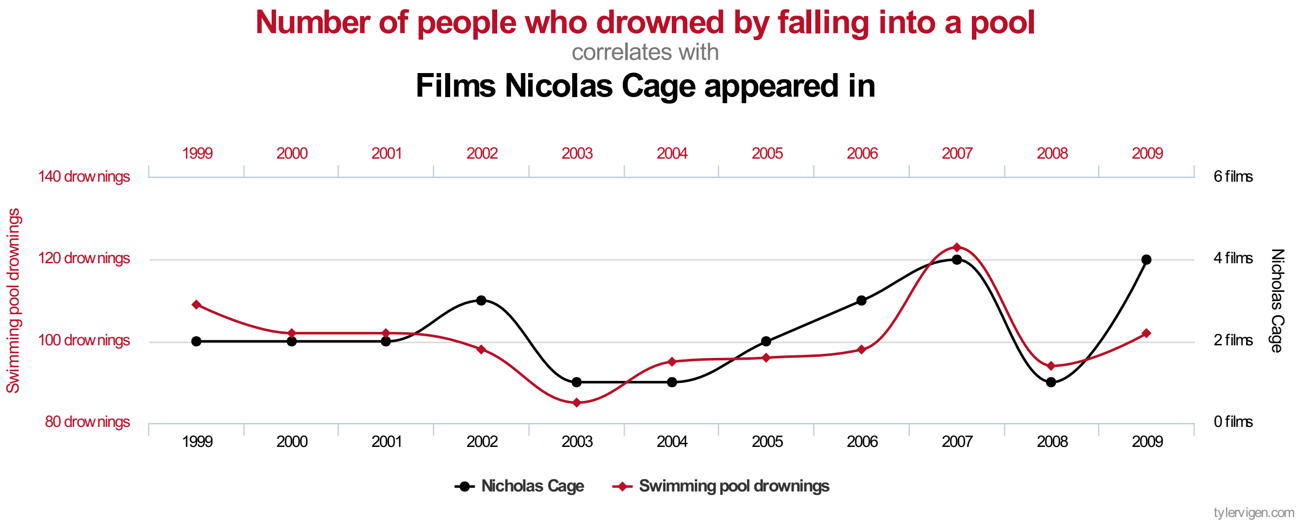
In [1]:

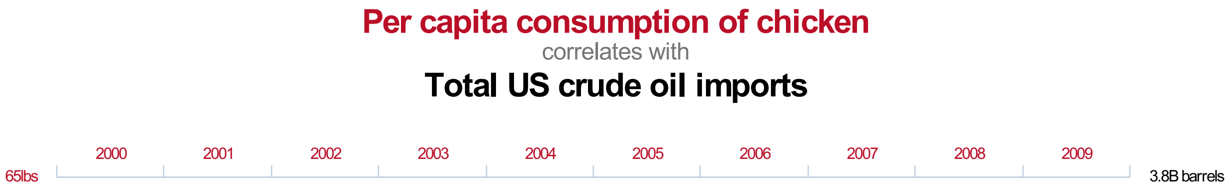
```
0.97**100 # Probability all 100 experiments with p=0.03 are all true
```

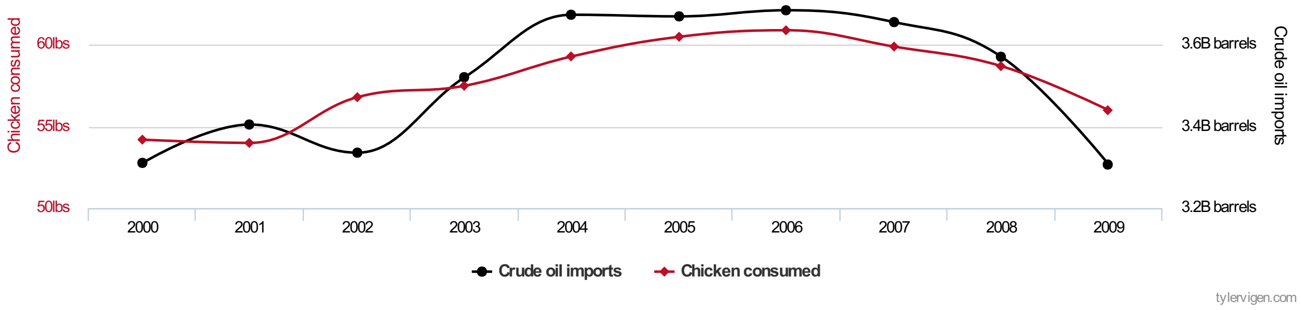
Out[1]:

```
0.04755250792540563
```

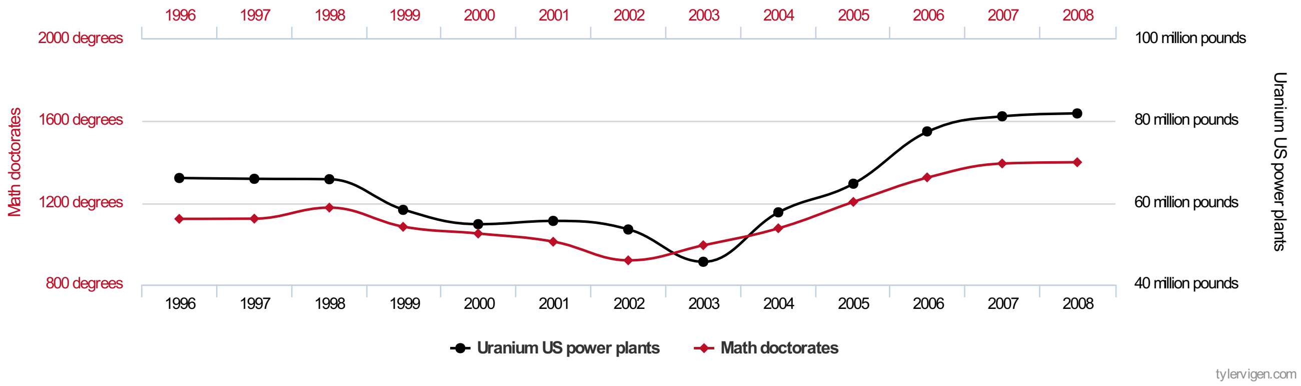
Similarly, if you are testing multiple metrics simultaneously in an experiment, the chances that one of these will satisfy your alpha threshold increases. A fun similar phenomenon is spurious correlation. If we start comparing a multitude of quantities, we are bound to find some quantities that are highly correlated, whether or not an actual relationship exists. Tyler Vigen set out to find such relationships; here are a few entertaining ones (of many):



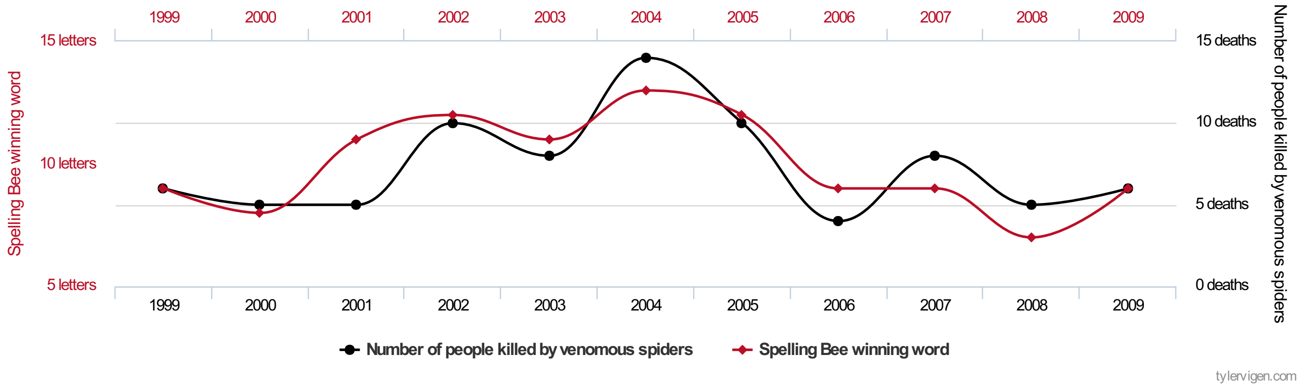




Math doctorates awarded
correlates with
Uranium stored at US nuclear power plants



Letters in Winning Word of Scripps National Spelling Bee
correlates with
Number of people killed by venomous spiders



As we can see, although these graphs show that each pair of quantities is strongly correlated, it seems unreasonable to expect that any of them have any causal relationships. Regardless of what the statistics tell us, there is no relationship through which the length of spelling bee word affects the number of people killed by venomous spiders.

How do multiple comparisons increase the chances of finding spurious correlations?

Spurious correlation is a ***Type 1 Error***, meaning that it's a type of ***False Positive***. We think we've found something important when really there isn't any. With each comparison we make in an experiment, we try to set a really low p-value to limit our exposure to type 1 errors. When we only reject the null hypothesis when p < 0.05, for example, we are effectively saying "I'm only going to accept these results as true if there is less than a 5% chance that I didn't actually find anything important, and my data only looks like this due to randomness". However, when we make ***multiple comparisons*** by checking for many things at once, each of the small risks of a Type 1 error becomes cumulative!

Here's another easy to way to phrase this -- a p-value threshold of less than 0.05 means that we will only make a Type 1 error 1 in every 20 times. This means that statistically, if we have 20 findings where the p-value is less than 0.05 at the same time, 1 of them is almost guaranteed to be a Type 1 error (False Positive) -- but we have no idea of which one!

The Bonferroni correction

Back to the problem of multiple comparisons. Due to the cumulative risk of drawing false conclusions when statistically testing

Back to the problem of multiple comparisons. Due to the cumulative risk of drawing false conclusions when statistically testing multiple quantities simultaneously, statisticians have devised methods to minimize the chance of type 1 errors. One of these is the ***Bonferroni correction***. With the Bonferroni correction, you divide α by the number of comparisons you are making to set a new, adjusted threshold rejecting the null hypothesis.

For example, if you desire $\alpha = 0.05$, but are making 10 comparisons simultaneously, the Bonferroni Correction would advise you set our adjusted p-value threshold to $\frac{0.05}{10} = 0.005$! The stricter p-value threshold helps control for Type 1 errors. This doesn't mean that you are immune to them -- it just helps reduce the cumulative chance that one occurs. That said, the effective power of these tests is therefore reduced (and in turn type 2 errors are more likely).

Additional Resources

- Tyle Vigen - Spurious correlations
- Nick Cage movies vs. drownings, and more strange (but spurious) correlations

Summary

In this lesson, you learned about the problems that can arise from doing multiple comparisons in a single experiment, as well as some entertaining spurious correlations that exist with real-world data.

Goodhart's Law and Metric Tracking

Introduction

In this lesson, you'll learn about ***Goodhart's law*** and why you should be cautious and thoughtful when making policy recommendations based on data.

Objectives

You will be able to:

- Define Goodhart's law and its relationship to hypothesis testing
- Identify real-world examples of Goodhart's law in action

What is Goodhart's law?

Goodhart's law is an observation made by the British economist Charles Goodhart in 1975. Charles Goodhart famously said:

> "Any observed statistical regularity will tend to collapse once pressure is placed upon it for control purposes." -- Charles Goodhart

In plain English, this translates to:

> "Any measure which becomes a target ceases to be an effective measure!"

So what does that mean?

Goodhart's law succinctly explains a cardinal sin that many data scientists, project managers, and CEOs make all the time without realizing it -- they make policy or set goals based on statistical metrics without considering the unintended consequences and effects these policies might have!

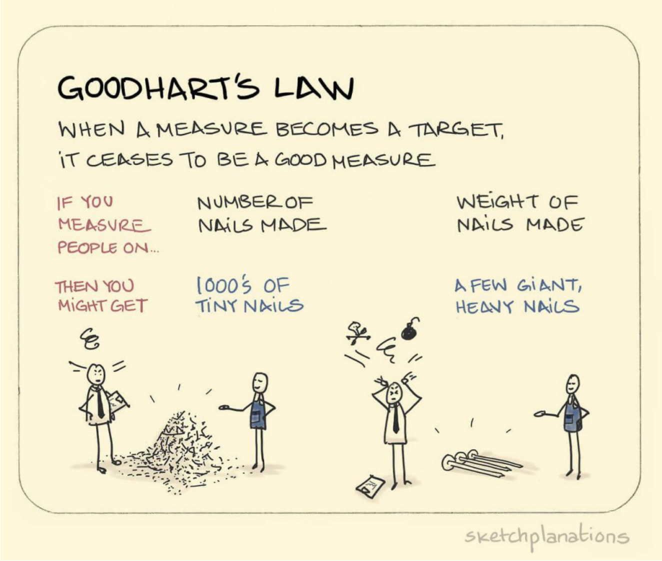


Image from: Sketchplantations

Example 1: Cobra skins

The Cobra effect refers to an anecdote that demonstrates an example of Goodhart's law in effect during the time of British rule of colonial India. As the story goes, a high-ranking officer in the British military was concerned about the number of highly venomous cobras that could be found in Delhi. He had the bright idea of offering a bounty for every cobra skin brought to him! Initially, this seemed to work -- people hunted cobras, sold the skins to the British government for their bounty, and the cobra population dipped slightly in the city. However, this soon backfired spectacularly, when citizens started breeding cobras! As a result, the cobra population stopped declining and even repopulated a bit. After a while, the officer caught onto the breeding, as he realized they were paying out many bounties but the cobra problem in the city was still prevalent as ever. After realizing this, he canceled the bounty. Ironically, this meant that all the cobra breeders now had no reason to keep the cobras they were breeding, so they dumped them in the street -- causing the city to have even more cobras than before the bounty program had been implemented in the first place!

(Fun fact: The French military made the same mistake when Hanoi, Vietnam was under their colonial rule with a rat bounty, and there is solid evidence to prove that this actually happened!)

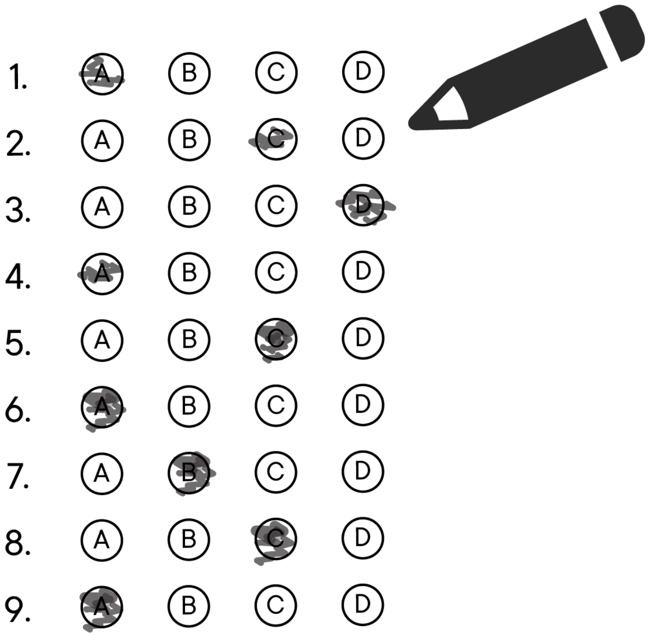
The problem with proxy metrics

The first mistake by this British commander was using a ***proxy metric*** in the form of "cobra skins collected". He mistakenly assumed that there was an inverse relationship between the number of skins turned in for a bounty and the number of wild cobras in the city of Delhi! Although this may have been the case at first, as hunting cobras was pretty much the only way to obtain skins to turn in for the bounty, he failed to realize that there were other possible sources for cobra skins that he hadn't accounted for. He wanted to reduce one metric, *Cobra population*, but he wasn't actually tracking that metric -- he was tracking a proxy for that metric which he assumed he could use to gauge what was happening to his target metric. The system he implemented had no way of determining if the cobra skins turned in for bounties were skins from cobras on the streets of Delhi -- with no way to tell, he had no way of knowing as his proxy metric became less and less relevant.

Policies can change things you didn't plan for

This leads to his other mistake -- he failed to account for how his policies might change things. Policies do not happen in a vacuum. They have a tendency to change things in unexpected ways, if not crafted thoughtfully and carefully! At first glance, introducing a monetary incentive for cobra skins seems like a good way to reduce the cobra population. However, he failed to account for the way this new incentive might change people's behaviors. By making cobra skins highly valuable, he inadvertently caused people to realize that breeding cobras was much safer, easier, and more lucrative than hunting them. Although his policy may have caused the change he wanted, in the beginning, he had no way of knowing what other sorts of behaviors this new policy might create or encourage.

Example 2: Standardized testing in US schools



A more depressing real-world example of Goodhart's law in action is the prevalence of standardized testing in the American public school system. These tests were originally designed as a way to measure both individual student performance and overall teacher and school effectiveness. However, school funding is tied directly to test scores. This incentivizes schools to "teach to the test", spending a disproportionate amount of class time each year focusing on test preparation. By having incentives for schools to focus heavily on preparing students for these tests, the system has created ripple effects including reorientating student's focus on preparing for tests rather than other learning goals that might be more characteristic of real-world applications such as project orientated tasks. In this case, policymakers started out with a harmless, positive intention -- measure student and school performance -- but failing to account for Goodhart's law and offering strong incentives in relation to these metrics has degraded the usefulness of these test scores by altering behaviors.

Why does this matter for Data Scientists?

Goodhart's law is something that matters much to Data Scientists because it is our findings and experiments that often drive the policies and decisions made by a company. Data Science is complex, and often, project managers, CEOs, and other decision makers don't want to know about experimental methodologies or confidence intervals -- they just want to know what the best decision they can make is, based on what the data says! It's quite common for decision makers to not realize that setting a target for one metric can negatively affect other metrics in ways that aren't immediately obvious. For instance, pushing employees at a call center to reduce call times might reduce customer satisfaction; it seems reasonable to imagine employees hustling to get off the phone based on this shorter call time "target" handed down from management.

As a data scientist, it is important to communicate your results clearly to stakeholders -- but it is also important to be the voice of reason at times. This is why communication with stakeholders is important throughout the process of any data science project. The sooner you know how they plan on using your results, the more you can help them avoid ugly unforeseen problems that come from Goodhart's law -- always remember that massive amounts of data are no substitute for *critical thinking*! At the very least, you should get a bit nervous when you see targets being set for certain metrics. Note that this doesn't necessarily mean "don't set targets" -- instead, seek to encourage decision makers to think critically about any unintended consequences these targets could have, and track changes in metrics early and often when new policies or targets are put in place to ensure that unintended consequences are caught early!

Summary

In this lesson, you learned about Goodhart's law and why you should be cautious and thoughtful when making policy recommendations based on data.

The Kolmogorov-Smirnov Test

Introduction

During data analysis, you have to satisfy a number of assumptions for the underlying dataset. One of the most common assumptions that you will come across is the "Normality Assumption", i.e., the underlying data roughly follows a normal distribution.

If the data is not found to be normally distributed (i.e. data with kurtosis and skew while doing linear regression), you may first answer a question like: “Given my data … if there is a deviation from normality, will there be a material impact on my results?”

In this lesson, we'll look at a popular statistical test for satisfying the normality assumption, the Kolmogorov-Smirnov test, or simply, the K-S test.

Objectives

You will be able to:

- Explain the role of the normality assumption in statistical tests
- Calculate a one-and two-sample Kolmogorov-Smirnov test
- Interpret the results of a one- and two-sample Kolmogorov-Smirnov test

Normality assumption

Formal normality tests always reject the huge sample sizes we work with today. When n (our sample size) gets large, even the smallest deviation from perfect normality will lead to a significant result. And as every dataset has some degree of random noise, no single dataset will be a **perfectly** normally distributed sample.

> **In applied statistics, the question is not whether the data/residuals are perfectly normal, but normal enough for the assumptions to hold**.

This question is answered through visualization techniques like qqplots, boxplots, or more advanced statistical tests including:

- The Shapiro-Wilk test;
- The Anderson-Darling test, and;
- The Kolmogorov-Smirnov test

In this lesson, we'll focus on the Kolmogorov-Smirnov test (K-S test) which will give you a strong foundation to help you understand and implement other tests when needed.

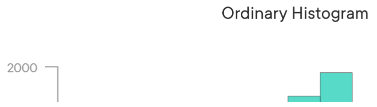
Kolmogorov-Smirnov Test

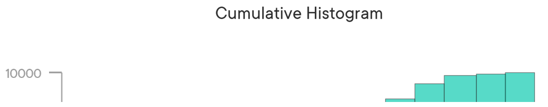
A K-S test provides a way of comparing distributions, whether two sample distributions or a sample distribution with a theoretical distribution - comparable to what we've already seen when we learned about one sample or two-sample t-tests. The distributions are compared in their cumulative form as **Empirical Cumulative Distribution Functions**. The test statistic in K-S test used to compare distributions is simply the maximum vertical distance between the two functions. Essentially, we are testing the sample data against another sample, to compare their distributions for similarities.

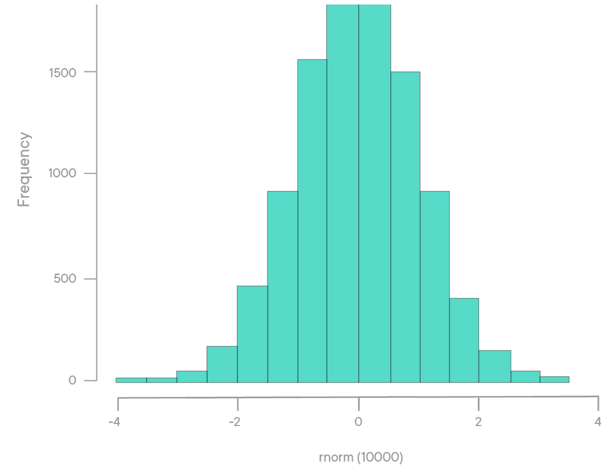
The Empirical Cumulative Distribution Function (ECDF)

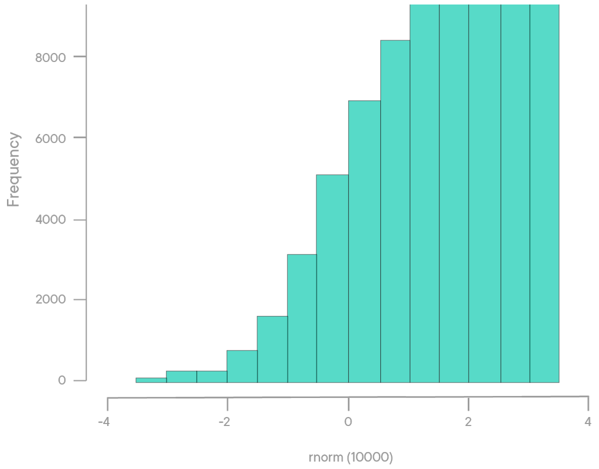
> An empirical cumulative distribution function (CDF) is a non-parametric estimator of the underlying CDF of a random variable. It assigns a probability to each data point, orders the data from smallest to largest in value, and calculates the sum of the assigned probabilities up to and including each data point.

The most intuitive way to think about the empirical distribution function is that it relates to the cumulative distribution function (CDF) in a similar way to how a histogram relates to a probability density function. Let's look at the following figures to get this idea:

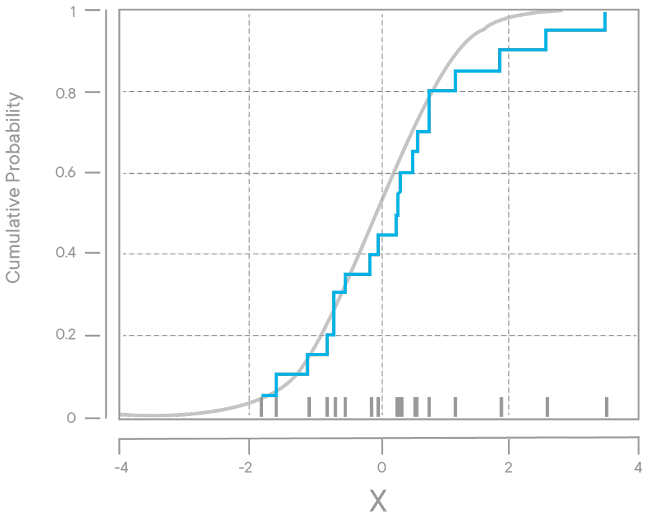

Ordinary Histogram


Cumulative Histogram





The left figure shows a regular histogram with samples looking like a normal distribution. The right figure shows the same samples except each bin in the histogram contains the cumulative count of samples up to that bin, which approximates the shape of the CDF for this random variable. Now the right figure doesn't exactly represent an empirical distribution function because the Y-axis is not normalized to 1 and the samples are binned instead of just plotted cumulatively. Nonetheless, the idea remains the same. An example of an empirical CDF is given below:



This image sums up the intuition for empirical distribution function. The blue line is our empirical CDF whereas the grey one is our theoretical CDF (i.e. plotted using parameters and fitting a probability function).

If X is a random variable with CDF $F(x) = P(X \le x)$, and $x1, \ldots, xn$ are i.i.d. random variables sampled from X . Then, the empirical distribution function, $F(x)$, is a CDF:

$$\hat{F}(x) = \frac{\#\text{ of elements in sample} \le x}{n} = \frac{1}{n}\sum_{i=1}^{n} I(x_i \le x)$$

One-Sample K-S test

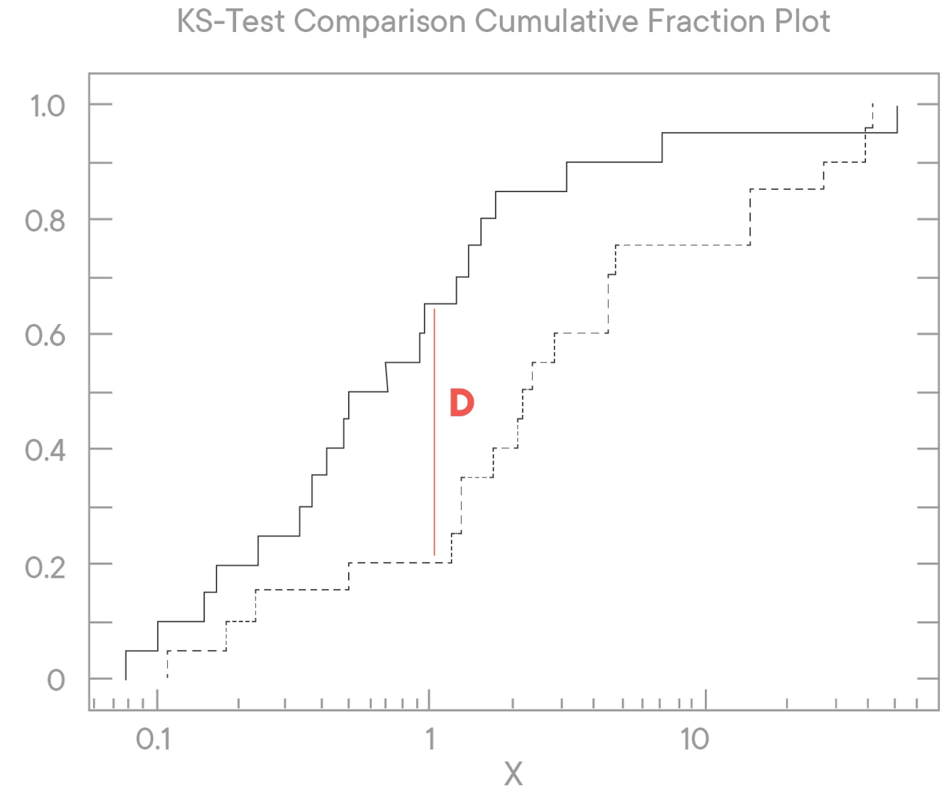
This is also known as the **Kolmogorov-Smirnov Goodness of Fit test**. It calculates the similarity between an observed (empirical) distribution and a completely specified theoretical continuous distribution. It is sensitive to all attributes of a distribution including mean, variance, and shape.

The key assumption of the one-sample test is that the theoretical distribution is fully defined continuous distribution, in terms of its parameters. This obviously means that its most common use case is that of testing normality. The test statistic, d, is simply the largest deviation between the observed cumulative function and the expected theoretical cumulative frequency distribution, i.e.

$$d = max(abs[F_0(X) - F_r(X)])$$

where

- **d** is the maximum deviation Kolmogorov statistic
- **F$_0$(X)** = (No.of observations ≤ X)/(Total no.of observations) i.e. the non parametric empirical distribution
- **F$_r$(X)** = The theoretical frequency distribution of X - parametric (e.g. based on mean value)



Null Hypothesis: There is no difference between the distribution of our sample and a normal distribution.

Acceptance Criteria: If the calculated value is less than the critical value, accept the null hypothesis.

Rejection Criteria: If the calculated value is greater than the critical value, reject the null hypothesis.

Example

Problem Statement:

In a study done from various modules of a data science course with 60 students, equal number of students are samples from each module. These students are interviewed and their intention to join the advanced machine learning module was noted. Following shows how many students showed a positive intention

- Python (5)
- Data Visualizations (9)
- SQL (11)
- Statistics (16)
- NLP (19)

It was expected that 12 students from each module would join advanced ML.

Let's use K-S test to find if there is any difference among student classes with regard to their intention of joining the advanced machine learning module.

First, we need to set up our null hypothesis.

> H_0: There is no difference among students of different modules with respect to their intention of joining advanced ML.

Streams	No. of students interested in joining		FO(X)	Fr(X)	|FO(X)-FT(X)|
	Observed(O)	Theoretical(T)			
Python	5	12	5/60	12/60	7/60
Viz.	9	12	14/60	24/60	10/60
SQL	11	12	25/60	36/60	11/60
Stats	16	12	41/60	48/60	7/60
NLP	19	12	60/40	60/60	60/60

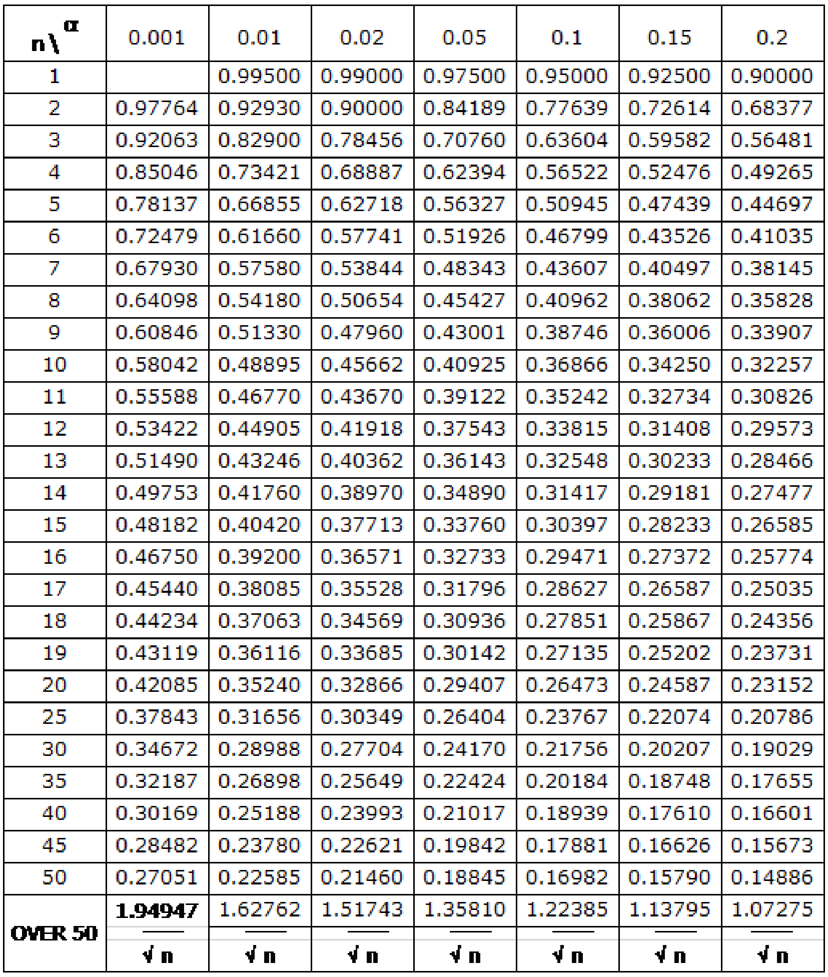
```
    Total       n=60
```

According to the formula above,

$$d = max(abs[F_0(X) - F_r(X)])$$

$$d = 11/60 = 0.183$$

Here's the Smirnov d-statistic for reference:



n\α	0.001	0.01	0.02	0.05	0.1	0.15	0.2
1		0.99500	0.99000	0.97500	0.95000	0.92500	0.90000
2	0.97764	0.92930	0.90000	0.84189	0.77639	0.72614	0.68377
3	0.92063	0.82900	0.78456	0.70760	0.63604	0.59582	0.56481
4	0.85046	0.73421	0.68887	0.62394	0.56522	0.52476	0.49265
5	0.78137	0.66855	0.62718	0.56327	0.50945	0.47439	0.44697
6	0.72479	0.61660	0.57741	0.51926	0.46799	0.43526	0.41035
7	0.67930	0.57580	0.53844	0.48343	0.43607	0.40497	0.38145
8	0.64098	0.54180	0.50654	0.45427	0.40962	0.38062	0.35828
9	0.60846	0.51330	0.47960	0.43001	0.38746	0.36006	0.33907
10	0.58042	0.48895	0.45662	0.40925	0.36866	0.34250	0.32257
11	0.55588	0.46770	0.43670	0.39122	0.35242	0.32734	0.30826
12	0.53422	0.44905	0.41918	0.37543	0.33815	0.31408	0.29573
13	0.51490	0.43246	0.40362	0.36143	0.32548	0.30233	0.28466
14	0.49753	0.41760	0.38970	0.34890	0.31417	0.29181	0.27477
15	0.48182	0.40420	0.37713	0.33760	0.30397	0.28233	0.26585
16	0.46750	0.39200	0.36571	0.32733	0.29471	0.27372	0.25774
17	0.45440	0.38085	0.35528	0.31796	0.28627	0.26587	0.25035
18	0.44234	0.37063	0.34569	0.30936	0.27851	0.25867	0.24356
19	0.43119	0.36116	0.33685	0.30142	0.27135	0.25202	0.23731
20	0.42085	0.35240	0.32866	0.29407	0.26473	0.24587	0.23152
25	0.37843	0.31656	0.30349	0.26404	0.23767	0.22074	0.20786
30	0.34672	0.28988	0.27704	0.24170	0.21756	0.20207	0.19029
35	0.32187	0.26898	0.25649	0.22424	0.20184	0.18748	0.17655
40	0.30169	0.25188	0.23993	0.21017	0.18939	0.17610	0.16601
45	0.28482	0.23780	0.22621	0.19842	0.17881	0.16626	0.15673
50	0.27051	0.22585	0.21460	0.18845	0.16982	0.15790	0.14886
OVER 50	$\frac{1.94947}{\sqrt{n}}$	$\frac{1.62762}{\sqrt{n}}$	$\frac{1.51743}{\sqrt{n}}$	$\frac{1.35810}{\sqrt{n}}$	$\frac{1.22385}{\sqrt{n}}$	$\frac{1.13795}{\sqrt{n}}$	$\frac{1.07275}{\sqrt{n}}$

The table value of d at 5% significance level is given by

$$d(0.05) = \frac{1.36}{\sqrt{n}} = \frac{1.36}{\sqrt{60}} = 0.175$$

Since the calculated d value (0.183) is greater than the critical value (0.175), hence we reject the null hypothesis and conclude that there is a difference among students of different modules in their intention of joining the advanced ML course.

Two-Sample K-S Test

The two-sample K-S test checks if two **independent** samples have been drawn from the same population, or, equivalently, from two identical populations (X = Y).

As with the one-sample test, it is moderately sensitive to all parameters of the distribution. The one-tailed version of this test has a specific purpose i.e .to test whether values of one population are larger than values of another population. Similar to one-sample test, cumulative distributions are compared, but here two sample distributions are compared instead of a sample distribution and a theoretical distribution as we saw above. For the two-tailed version of the test, the test statistic (d) is the largest absolute deviation between the two observed cumulative step functions, irrespective of the direction of the difference.

> The null hypothesis states for this test that there is no difference between the two distributions. The d-statistic is calculated in the same manner as we saw above.

$$d = max[abs[F_{n1}(X) - F_{n2}(X)]]$$

- n1 = Observations from first sample.
- n2 = Observations from second sample.

When the cumulative distribution shows large maximum deviation d, it is a reflection of the difference between the two sample distributions.

The critical value of d for samples where n1=n2 and is ≤ 40, the K-S table for two sample case is used. When n1 and/or n2 > 40 then the K-S table for large samples of two-sample test should be used. The null hypothesis is accepted if the calculated value is less than the table value and vice-versa.

Thus, the use of any of these nonparametric tests helps a researcher to test the significance of his results when the characteristics of the target population are unknown or no assumptions had been made about them.

Example

Given two samples, test if their distributions are the same.

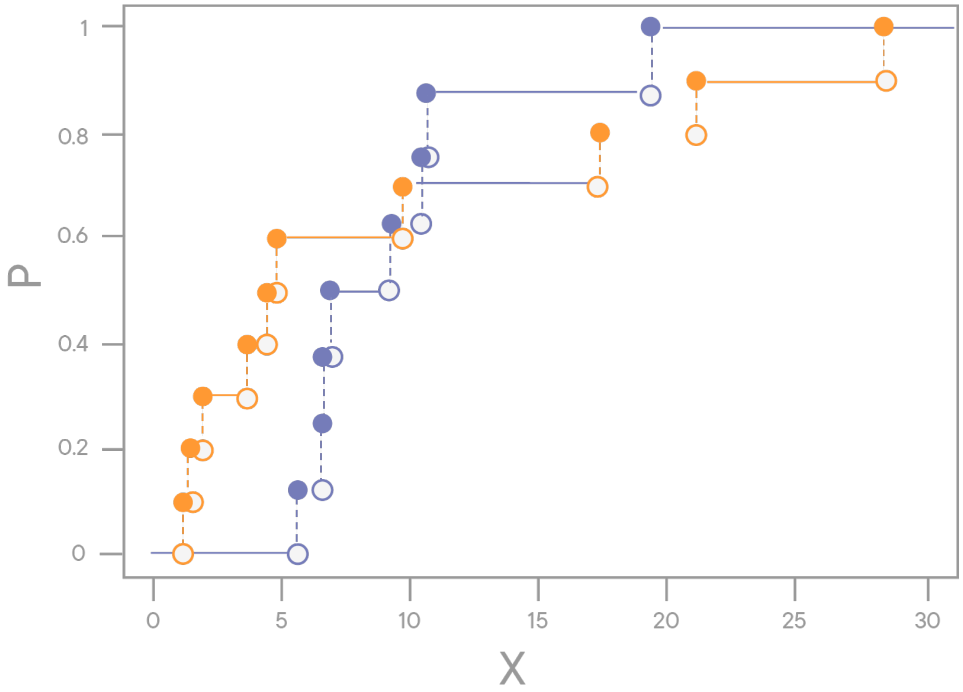
Compute the observed cumulative distribution functions of the two samples and compute their maximum difference.

```
X : 1.2, 1.4, 1.9, 3.7, 4.4, 4.8, 9.7, 17.3, 21.1, 28.4
Y : 5.6, 6.5, 6.6, 6.9, 9.2, 10.4, 10.6, 19.3
```

We sort the combined sample, in order to compute the empirical cdfs:

the combined sample, in order to compute the empirical cdf’s:

```
   1.2 1.4 1.9 3.7 4.4 4.8 5.6 6.5 6.6 6.9 9.2 9.7 10.4 10.6 17.3 19.3 21.1 28.4
Fx 0.1 0.2 0.3 0.4 0.5 0.6 0.6 0.6 0.6 0.6 0.6 0.7 0.7  0.7  0.8  0.8  0.9  1.0
Fy 0.0 0.0 0.0 0.0 0.0 0.0 0.1 0.2 0.4 0.5 0.6 0.6 0.8  0.9  0.9  1.0  1.0  1.0
```



The Kolmogorov-Smirnov statistic is again the maximum absolute difference of the two observed distribution functions. From the above image, and also by feeding above values in the given formula, we get **d = 0.6**.

For two samples, the 95% critical value can be approximated by the formula:

$$K(0.05) = 1.36\sqrt{1/ + 1/} = 0.645$$

$$d(0.05) = 1.36\sqrt{1/n_1 + 1/n_2} = 0.645$$

Since 0.6 < 0.645, we retain the null hypothesis in this case.

Kolmogorov-Smirnov tests have the advantages that:

- the distribution of statistic does not depend on cumulative distribution function being tested and
- the test is exact

They have the disadvantage that they are more sensitive to deviations near the center of the distribution than at the tails.

Summary

In this lesson, we looked at K-S test and how this test can be used to test for normality assumptions. We also looked at a one-sample K-S test and a two-sample K-S test with simple examples. Next, we'll see how to implement these tests in Python.

ANOVA

Introduction

ANOVA (Analysis of Variance) is a method for generalizing statistical tests to multiple groups. As you'll see, ANOVA analyses the overall variance of a dataset by partitioning the total sum of squared deviations (from the mean) into the sum of squares for each of these groups and sum of squares for error. By comparing the statistical test for multiple groups, it can serve as a useful alternative to the t-tests you've encountered thus far when you wish to test multiple factors simultaneously.

Objectives

You will be able to:

- Explain the methodology behind ANOVA tests
- Use ANOVA for testing multiple pairwise comparisons

Explanation of ANOVA

To perform ANOVA, you begin with sample observations from multiple groups. Since ANOVA is looking to explain the total variance as combinations of variance from the various groups, you typically design a multiple groups experiment to test various independent factors that we hypothesize may influence the overall result. For example, imagine an email campaign designed to optimize donation contributions. In order to get the most money in donations, one might send out two different emails, both copies being identical except for the subject line. This would form a sensible hypothesis test, but if you wanted to test multiple changes simultaneously, swapping out subject line, time sent, thank you gift offers, or other details in the email campaign, then ANOVA would be a more appropriate methodology. In this scenario, you would change one or more of these various features and record the various donations. Once you have sample observations from various combinations of these features, you can then use ANOVA to analyze and compare the effectiveness of the individual features themselves.

The general idea is to break the sum of squared deviations into multiple parts: the sum of squared deviations of the mean of each of the test groups to the observations within the group itself, and the sum of squared deviations of the mean of these test groups to the mean of all observations.

This is easier to understand through the context of an example. For the email case described above, ANOVA would compare the variance of donations within each of the groups to the overall variance of all donations (or lack thereof) as a whole. If the variance of a single group's donations versus that of the overall sample is substantial, there is reason to reject the null hypothesis for that feature. This forms the foundation of the f-test which is at the heart of ANOVA.

Recall that you would not perform multiple t-tests with such a scenario because of the multiple comparisons problem. Type I errors will be confounded when conducting multiple t-tests. While the alpha threshold for any one test might be 0.05, it would not be surprising to reject the null hypothesis in at least one of these cases, just by pure chance, if you conduct 5 or 10 t-tests.

ANOVA in Python

In [1]:

```
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

Loading the data

As usual, we start by loading in a dataset of our sample observations. This particular table is of salaries in IT and has 4 columns:

- S - the individuals salary
- X - years of experience
- E - education level (1-Bachelors, 2-Masters, 3-PHD)
- M - management (0-no management, 1-yes management)

In [2]:

```
df = pd.read_csv('IT_salaries.csv')
```

```
df.head()
```

Out[2]:

	S	X	E	M
0	13876	1	1	1
1	11608	1	3	0
2	18701	1	3	1
3	11283	1	2	0
4	11767	1	3	0

Generating the ANOVA table

In order to generate the ANOVA table, you first fit a linear model and then generate the table from this object. Our formula will be written as:

`Control_Column ~ C(factor_col1) + factor_col2 + C(factor_col3) + ... + X`

We indicate categorical variables by wrapping them with `C()`.

In [3]:

```
formula = 'S ~ C(E) + C(M) + X'
lm = ols(formula, df).fit()
table = sm.stats.anova_lm(lm, typ=2)
print(table)
```

```
                sum_sq    df           F        PR(>F)
C(E)      9.152624e+07   2.0   43.351589  7.672450e-11
C(M)      5.075724e+08   1.0  480.825394  2.901444e-24
X         3.380979e+08   1.0  320.281524  5.546313e-21
Residual  4.328072e+07  41.0         NaN           NaN
```

Interpreting the table

For now, simply focus on the outermost columns. On the left, you can see our various groups, and on the right, the probability that the factor is indeed influential. Values less than 0.05 (or whatever we set α to) indicate rejection of the null hypothesis. In this case, notice that all three factors appear influential, with management being the potentially most significant, followed by years experience, and finally, educational degree.

Summary

In this lesson, you examined the ANOVA technique to generalize testing methods to multiple groups and factors.

Statistical Power and ANOVA - Recap

Introduction

You've covered quite a bit in this section and should be gearing up to start conducting your own hypothesis testing! Before moving on to that exciting realm, take a minute to review some of the key takeaways.

Key Takeaways

Remember that the section began where the last left off, examining the relationship between α, power, effect size, and sample size. As you saw, these 4 quantities form a deterministic relationship; know any 3, and you can caulculate the fourth. While a lower alpha value will lead to fewer type I errors, and a higher power will lead to fewer type II errors, in practice these are often set to common default standards due to exploding sample sizes required to detect various effect sizes. Some common thresholds used are:

- Setting alpha equal to 0.05 (or 0.01)
- Requiring power values of 0.8 or greater

After a thorough investigation of this relationship, you then also saw an alternative t-test, Welch's t-test which can be used for comparing samples of different sizes or different variances. While the formula was a bit complicated, the most important piece to remember is that when the assumptions that sample size and sample variance are equal for the two samples is violated, use Welch's t-test rather than the Student's t-test.

Aside from ensuring that the assumptions of a t-test are met, it's also important to know how type I errors are compounded if you perform multiple tests. This is known as the multiple comparison problem and you saw that type I errors compound under multiple tests. So while the probability of a type I error is equal to α for any one test, the collective probability that there is at least 1 type I error continues to increase as you perform more tests, further detracting from the confidence that you have uncovered a meaningful relationship. In order to account for this, you can use stricter criteria when defining α such as the Bonferroni correction. Alternatively, ANOVA is equivalent to a 2-sided t-test when comparing two groups, but also generalizes appropriately to multiple group comparisons.

Summary

Remember that simply observing a low p-value is not meaningful in and of itself. There are a number of factors to take into consideration when interpreting the results of a statistical test, from alpha, power, sample size, effect size, and the formulation of the problem itself. Good hypothesis testing requires careful thought and design.