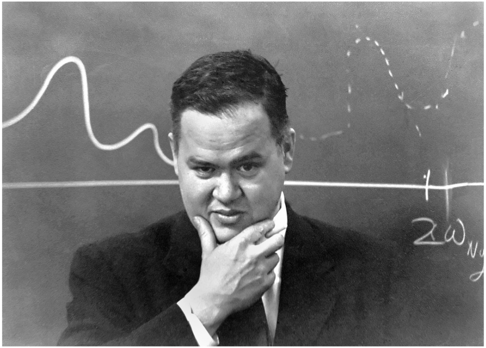
Exploratory Data Analysis, Part 1: Tabular Methods

DS 6001: Practice and Applications of Data Science

What is Exploratory Data Analysis?

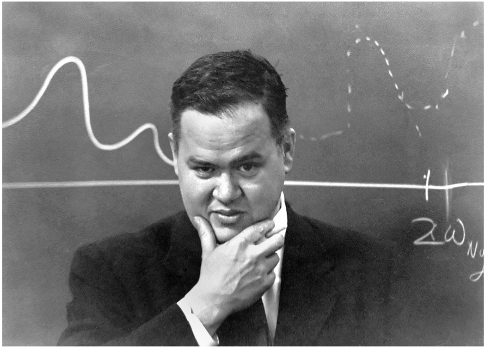
According to statistician John Tukey, one of the originators of data science, exploratory data analysis (EDA) is



“an attitude”, and “a flexibility”.

What is Exploratory Data Analysis?

According to statistician John Tukey, one of the originators of data science, exploratory data analysis (EDA) is



“an attitude”, and “a flexibility”.

The goal of EDA is not to answer a specific research question, but rather to “dig in” to the data to get a better sense of the important properties of the data.

What is Exploratory Data Analysis?

EDA includes fast and simple approaches to

- collect preliminary findings,
- assess the assumptions that underlie other methods,
- and identify problems and complications such as outliers.

What is Exploratory Data Analysis?

EDA includes fast and simple approaches to

- collect preliminary findings,
- assess the assumptions that underlie other methods,
- and identify problems and complications such as outliers.

There's no fixed set of methods that comprise EDA. In Tukey's words:

No catalog of techniques can convey a willingness to look for what can be seen, whether or not anticipated. Yet this is at the heart of exploratory data analysis. . . . [T]he picture-examining eye is the best finder we have of the wholly unanticipated.

What is Exploratory Data Analysis?

EDA includes fast and simple approaches to

- collect preliminary findings,
- assess the assumptions that underlie other methods,
- and identify problems and complications such as outliers.

There's no fixed set of methods that comprise EDA. In Tukey's words:

No catalog of techniques can convey a willingness to look for what can be seen, whether or not anticipated. Yet this is at the heart of exploratory data analysis. . . . [T]he picture-examining eye is the best finder we have of the wholly unanticipated.

EDA uses tables and graphs as a means for researchers to simply **see what's there**. After all the work to get and clean the data, EDA can be very joyful part of a data project.

Descriptive Statistics

There are a few broad categories of descriptive statistics.

Descriptive Statistics

There are a few broad categories of descriptive statistics.

Measures of location or central tendency describe the values that **typical datapoints** take on: the mean, a weighted mean, and the median.

Descriptive Statistics

There are a few broad categories of descriptive statistics.

Measures of location or central tendency describe the values that **typical datapoints** take on: the mean, a weighted mean, and the median.

Measures of variability or dispersion report the **typical distance** that each datapoint's value is away from the middle or mean of the distribution: the variance, standard deviation, the interquartile range, min and max, and percentiles.

Descriptive Statistics

There are a few broad categories of descriptive statistics.

Measures of location or central tendency describe the values that **typical datapoints** take on: the mean, a weighted mean, and the median.

Measures of variability or dispersion report the **typical distance** that each datapoint's value is away from the middle or mean of the distribution: the variance, standard deviation, the interquartile range, min and max, and percentiles.

Measures of frequency report the count of **how many times each distinct value** of categorical features appears in the data, or how many values of a continuous feature exist within pre-specified bins: raw counts and percentages.

Descriptive Statistics

The simple mean automatically ignores missing values, which assumes that missing values are equal to the mean:

```
anes.ftbiden.mean()
42.15189466923571
```

Descriptive Statistics

The simple mean automatically ignores missing values, which assumes that missing values are equal to the mean:

```
anes.ftbiden.mean()
42.15189466923571
```

A trimmed mean sorts the values of a column and removes the top and bottom percentage from the column. It is one way to deal with **outliers**. To remove the top and bottom 10% of values, type:

```
stats.trim_mean(anes.ftbiden, .1)
41.58981444926964
```

Descriptive Statistics

The simple mean automatically ignores missing values, which assumes that missing values are equal to the mean:

```
anes.ftbiden.mean()
42.1518946692357
```

A trimmed mean sorts the values of a column and removes the top and bottom percentage from the column. It is one way to deal with **outliers**. To remove the top and bottom 10% of values, type:

```
stats.trim_mean(anes.ftbiden, .1)
41.58981444926964
```

Another way to account for outliers is to calculate the median. Half of the values exist at or above the median and half exist at or below the median:

```
anes.ftbiden.median()
42.0
```

Descriptive Statistics

Surveys often draw samples that are very different demographically from the population. It is usually the case that some races, genders, educational levels, and socioeconomic statuses are over-represented relative to others.

Descriptive Statistics

Surveys often draw samples that are very different demographically from the population. It is usually the case that some races, genders, educational levels, and socioeconomic statuses are over-represented relative to others.

One way to address these sampling biases is to calculate sampling weights that can be used to place greater or lesser emphasis on individual values when calculating statistics like means.

Descriptive Statistics

Surveys often draw samples that are very different demographically from the population. It is usually the case that some races, genders, educational levels, and socioeconomic statuses are over-represented relative to others.

One way to address these sampling biases is to calculate sampling weights that can be used to place greater or lesser emphasis on individual values when calculating statistics like means.

Say we draw a sample that contains **60% men and 40% women** from a population with 50% men and 50% women: we reweight each row with a man's responses as $.5/.6 = .833$ the rows for women by $.5/.4 = 1.25$.

Descriptive Statistics

To calculate a weighted mean, use `np.average()` with the `weights` parameter:

```
anes_temp = anes.loc[~anes.ftbiden.isna()]
np.average(anes_temp['ftbiden'], weights=anes_temp.weight)
43.31193635270897
```

Descriptive Statistics

To calculate a weighted mean, use `np.average()` with the `weights` parameter:

```
anes_temp = anes.loc[~anes.ftbiden.isna()]
np.average(anes_temp['ftbiden'], weights=anes_temp.weight)
43.31193635270897
```

To calculate a weighted median, use the `weighted.median()` function from the `wquantiles` package:

```
weighted.median(anes.ftbiden, anes.weight)
47.0
```

Descriptive Statistics

Measures of variability report on how far from the mean the "typical" value in a column happens to be. The most common measures of variability are the variance and the standard deviation,

```
[anes.ftbiden.var(), anes.ftbiden.std()]
[1118.0106501193195, 33.43666253071934]
```

Descriptive Statistics

Measures of variability report on how far from the mean the "typical" value in a column happens to be. The most common measures of variability are the variance and the standard deviation,

```
[anes.ftbiden.var(), anes.ftbiden.std()]
[1118.0106501193195, 33.43666253071934]
```

and the minimum and maximum:

```
[anes.ftbiden.min(), anes.ftbiden.max()]
[0.0, 100.0]
```

Descriptive Statistics

A percentile is the value in the column for which the specified percent of values are below that value. A quantile is the same as a percentile, using proportions instead of percents:

```
anes.ftbiden.quantile([0, .1, .25, .33, .5, .67, .75, .9, 1])
```

```
0.00      0.0
0.10      1.0
0.25      7.0
0.33     16.0
0.50     42.0
0.67     60.0
0.75     70.0
0.90     90.0
1.00    100.0
Name: ftbiden, dtype: float64
```

Descriptive Statistics

Percentiles can show us situations in which a column has a high degree of variability: the smaller the low percentiles are and the bigger the high percentiles are, the most variance exists in the column.

Descriptive Statistics

Percentiles can show us situations in which a column has a high degree of variability: the smaller the low percentiles are and the bigger the high percentiles are, the most variance exists in the column.

A simple way to understand the distance between high and low percentiles is to calculate the interquartile range (IQR), which is simply the difference between the 75th and 25th percentiles:

```
[anes.ftbiden.quantile(0.75),
 anes.ftbiden.quantile(0.25),
 anes.ftbiden.quantile(0.75) - anes.ftbiden.quantile(0.25)]
```

```
[70.0, 7.0, 63.0]
```

Descriptive Statistics

Categorical features can be either ordered or unordered.

Descriptive Statistics

Categorical features can be either ordered or unordered.

If the categories are ordered, then we can convert the column to numeric, and calculate the mean, median, etc.:

```
anes['ui_num'] = anes.universal_income.map({'Oppose a great deal':1,
                                            'Oppose a moderate amount':2,
                                            'Oppose a little':3,
                                            'Neither favor nor oppose':4,
                                            'Favor a little':5,
                                            'Favor a moderate amount':6,
                                            'Favor a great deal':7})
[anes.ui_num.mean(),
 anes.ui_num.median(),
 anes.ui_num.std(),
 anes.ui_num.quantile(.75)-anes.ui_num.quantile(.25)]
```

```
[3.51911532385466, 4.0, 2.134250040957768, 4.0]
```

Descriptive Statistics

Whether or not the categories are ordered, we can generate a frequency table:

```
anes.universal_income.value_counts()
```

```
Oppose a great deal         1007
Neither favor nor oppose     704
Favor a great deal           377
Favor a little               349
Favor a moderate amount      321
Oppose a moderate amount     216
Oppose a little              191
Name: universal_income, dtype: int64
```

Descriptive Statistics

Whether or not the categories are ordered, we can generate a frequency table:

```
anes.universal_income.value_counts()
```

```
Oppose a great deal          1007
Neither favor nor oppose      704
Favor a great deal            377
Favor a little                349
Favor a moderate amount       321
Oppose a moderate amount      216
Oppose a little               191
Name: universal_income, dtype: int64
```

A better version is available from the `sidetable` package:

```
anes.stb.freq(['universal_income'])
```

	universal_income	Count	Percent	Cumulative Count	Cumulative Percent
0	Oppose a great deal	1007	0.318167	1007	0.318167
1	Neither favor nor oppose	704	0.222433	1711	0.540600
2	Favor a great deal	377	0.119115	2088	0.659716
3	Favor a little	349	0.110269	2437	0.769984
4	Favor a moderate amount	321	0.101422	2758	0.871406
5	Oppose a moderate amount	216	0.068246	2974	0.939652
6	Oppose a little	191	0.060348	3165	1.000000

Descriptive Statistics

A continuous-valued feature can be turned categorical by placing the values into equal or unequal-sized bins:

```
binnedbiden = pd.cut(anes['ftbiden'], 10)
binnedbiden.value_counts()
```

```
(-0.1, 10.0]     909
(90.0, 100.0]    308
(40.0, 50.0]     293
(50.0, 60.0]     292
(60.0, 70.0]     253
(80.0, 90.0]     252
(10.0, 20.0]     234
(70.0, 80.0]     205
(30.0, 40.0]     192
(20.0, 30.0]     176
Name: ftbiden, dtype: int64
```

```
binnedbiden = pd.cut(anes['ftbiden'], [-.1, 0, 10, 30, 50, 70, 99, 100])
binnedbiden.value_counts()
```

```
(70.0, 99.0]     670
(0.0, 10.0]      644
(50.0, 70.0]     545
(30.0, 50.0]     485
(10.0, 30.0]     410
(-0.1, 0.0]      265
(99.0, 100.0]     95
Name: ftbiden, dtype: int64
```

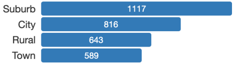
Descriptive Statistics

The `pandas_profiling` package provides an EDA dashboard.
`minimal=True` is faster, but `minimal=False` provides more info:

```
profile = ProfileReport(anes,
                        title='Pandas Profiling Report',
                        html={'style':{'full_width':True}},
                       minimal=True)
profile.to_notebook_iframe()
```

liveurban
Categorical

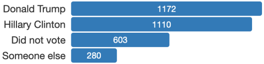
Distinct count	4
Unique (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	24.7 KiB



Toggle details

vote16
Categorical

Distinct count	4
Unique (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	24.7 KiB



Toggle details

Correlations

Relationships between two continuous features can be quantified with a correlation: a number that can be positive, negative, or zero:

Correlations

Relationships between two continuous features can be quantified with a correlation: a number that can be positive, negative, or zero:

- Positive numbers mean that the two features tend to increase together or decrease together.
- Negative numbers mean increases in one feature tend to occur with decreases in the other.
- Zero indicates that there is no discernible relationship.

Correlations

Relationships between two continuous features can be quantified with a correlation: a number that can be positive, negative, or zero:

- Positive numbers mean that the two features tend to increase together or decrease together.
- Negative numbers mean increases in one feature tend to occur with decreases in the other.
- Zero indicates that there is no discernible relationship.

The most common correlation is Pearson's correlation coefficient, which can take on any real value between -1 and 1.

Correlations

The `.corr()` method produces a symmetric matrix with 1s on the diagonal (indicating that each feature is perfectly correlated with itself), and the correlation between the two features in the off-diagonal elements:

```
anes.loc[:,'fttrump':'ftimmig'].corr()
```

	fttrump	ftobama	ftbiden	ftwarren	ftsanders	ftbuttigieg
fttrump	1.000000	-0.754178	-0.646357	-0.699664	-0.678443	-0.588964
ftobama	-0.754178	1.000000	0.805100	0.783064	0.720092	0.714641
ftbiden	-0.646357	0.805100	1.000000	0.733601	0.664075	0.728557
ftwarren	-0.699664	0.783064	0.733601	1.000000	0.798636	0.706680
ftsanders	-0.678443	0.720092	0.664075	0.798636	1.000000	0.612547
ftbuttigieg	-0.588964	0.714641	0.728557	0.706680	0.612547	1.000000

Conditional Means and Other Statistics

The best way to describe the relationship between a categorical feature and a continuous one is with a table with one row for every category and a column for each statistic we calculate within these categories.

Conditional Means and Other Statistics

The best way to describe the relationship between a categorical feature and a continuous one is with a table with one row for every category and a column for each statistic we calculate within these categories.

We can also **define functions** for custom statistics and use these within .agg():

```
def q25(x): return x.quantile(0.25)
def q75(x): return x.quantile(0.75)
anes.groupby('partyID').agg({'ftbiden':['mean', 'median', q25, q75]}).round(2)
```

	ftbiden			
	mean	**median**	**q25**	**q75**
partyID				
Democrat	66.38	70.0	50.0	88.0
Independent	35.26	33.0	6.0	53.0
Republican	18.82	8.0	1.0	31.0

Cross-Tabulations

A cross-tab describes the relationship between two categorical features.

Cross-Tabulations

A cross-tab describes the relationship between two categorical features.

The categories of one feature comprise the rows, and the categories of the other feature comprise the columns, and the cells contain a statistic (often the frequency):

```
pd.crosstab(anes.universal_income, anes.ideology)
```

ideology	Conservative	Liberal	Moderate
universal_income			
Favor a great deal	55	200	102
Favor a little	72	129	114
Favor a moderate amount	46	154	102
Neither favor nor oppose	135	181	241
Oppose a great deal	717	55	219
Oppose a little	46	55	76
Oppose a moderate amount	82	46	73

Cross-Tabulations

To change the order the categories in the table, convert the column to the category data type, and use the `.cat.reorder_categories()` method:

```
anes['universal_income'] = anes['universal_income'].cat.reorder_categories(['Oppose a great deal',
                                                                            'Oppose a moderate amount',
                                                                            'Oppose a little',
                                                                            'Neither favor nor oppose',
                                                                            'Favor a little',
                                                                            'Favor a moderate amount',
                                                                            'Favor a great deal'])
anes['ideology'] = anes['ideology'].cat.reorder_categories(['Liberal','Moderate','Conservative'])
pd.crosstab(anes.universal_income, anes.ideology)
```

ideology	**Liberal**	**Moderate**	**Conservative**
universal_income			
Oppose a great deal	55	219	717
Oppose a moderate amount	46	73	82
Oppose a little	55	76	46
Neither favor nor oppose	181	241	135
Favor a little	129	114	72
Favor a moderate amount	154	102	46
Favor a great deal	200	102	55

Cross-Tabulations

Raw counts are not always the most informative statistic to place within the cells. We can convert these counts to percents.

Cross-Tabulations

Raw counts are not always the most informative statistic to place within the cells. We can convert these counts to percents.

Row percents calculate the quotient of the count to the row total. Use `normalize='index'`:

```
(pd.crosstab(anes.universal_income, anes.ideology, normalize='index')*100).round(2)
```

ideology universal_income	Liberal	Moderate	Conservative
Oppose a great deal	5.55	22.10	72.35
Oppose a moderate amount	22.89	36.32	40.80
Oppose a little	31.07	42.94	25.99
Neither favor nor oppose	32.50	43.27	24.24
Favor a little	40.95	36.19	22.86
Favor a moderate amount	50.99	33.77	15.23
Favor a great deal	56.02	28.57	15.41

Cross-Tabulations

Column percents calculate the quotient of the count to the column total. Use `normalize='columns'`:

```
(pd.crosstab(anes.universal_income, anes.ideology, normalize='columns')*100).round(2)
```

ideology	Liberal	Moderate	Conservative
universal_income			
Oppose a great deal	6.71	23.62	62.19
Oppose a moderate amount	5.61	7.87	7.11
Oppose a little	6.71	8.20	3.99
Neither favor nor oppose	22.07	26.00	11.71
Favor a little	15.73	12.30	6.24
Favor a moderate amount	18.78	11.00	3.99
Favor a great deal	24.39	11.00	4.77

Cross-Tabulations

Cell percents calculate the quotient of the count to the overall total. Use `normalize=True`:

```
(pd.crosstab(anes.universal_income, anes.ideology, normalize=True)*100).round(2)
```

ideology universal_income	Liberal	Moderate	Conservative
Oppose a great deal	1.90	7.55	24.72
Oppose a moderate amount	1.59	2.52	2.83
Oppose a little	1.90	2.62	1.59
Neither favor nor oppose	6.24	8.31	4.66
Favor a little	4.45	3.93	2.48
Favor a moderate amount	5.31	3.52	1.59
Favor a great deal	6.90	3.52	1.90

Cross-Tabulations

We can populate the cells with statistics other than counts and percents. These cells are calculated from a third column, which we specify with the `values` parameter. We use the `aggfunction` to specify the function to apply within each cell:

```
pd.crosstab(anes.liveurban, anes.ideology,
            values=anes.partisanship, aggfunc='mean').round(2)
```

ideology liveurban	Liberal	Moderate	Conservative
City	56.66	21.66	-45.38
Rural	46.16	6.45	-64.23
Suburb	55.42	21.39	-63.63
Town	52.12	19.71	-63.97

Hypothesis Tests

Hypothesis tests are ways to measure certainty about what we see regarding relationships and comparisons in the data.

Hypothesis Tests

Hypothesis tests are ways to measure certainty about what we see regarding relationships and comparisons in the data.

Here's how hypothesis tests work:

Hypothesis Tests

Hypothesis tests are ways to measure certainty about what we see regarding relationships and comparisons in the data.

Here's how hypothesis tests work:

Suppose that I grab a soapbox and place it in the middle of Times Square in New York City, and I hop on to it, lift a megaphone to my mouth, and yell:

Hypothesis Tests

Hypothesis tests are ways to measure certainty about what we see regarding relationships and comparisons in the data.

Here's how hypothesis tests work:

Suppose that I grab a soapbox and place it in the middle of Times Square in New York City, and I hop on to it, lift a megaphone to my mouth, and yell:

CATS ARE JUST SMALL DOGS

Hypothesis Tests

Hypothesis tests are ways to measure certainty about what we see regarding relationships and comparisons in the data.

Here's how hypothesis tests work:

Suppose that I grab a soapbox and place it in the middle of Times Square in New York City, and I hop on to it, lift a megaphone to my mouth, and yell:

CATS ARE JUST SMALL DOGS

Suppose for a moment that **this insane thing to say is actually true**. Then think about how cats and dogs behave in the real world. How compatible is the real world behavior of cats and dogs with the assertion that cats are small dogs?

Hypothesis Tests

CATS ARE JUST SMALL DOGS

Hypothesis Tests

CATS ARE JUST SMALL DOGS

1. Cats like to sit on windowsills, while dogs seldom have the patience to sit on a windowsill for very long. In that way, if cats are small dogs, then **cats are very unusual dogs**.

Hypothesis Tests

CATS ARE JUST SMALL DOGS

1. Cats like to sit on windowsills, while dogs seldom have the patience to sit on a windowsill for very long. In that way, if cats are small dogs, then **cats are very unusual dogs**.

2. Dogs like to fetch and they respond when we call them by name. I've never seen a cat that would fetch or react at all to its name. In that way, again, if cats are small dogs, then **cats are very peculiar dogs**.

Hypothesis Tests

CATS ARE JUST SMALL DOGS

1. Cats like to sit on windowsills, while dogs seldom have the patience to sit on a windowsill for very long. In that way, if cats are small dogs, then **cats are very unusual dogs**.

2. Dogs like to fetch and they respond when we call them by name. I've never seen a cat that would fetch or react at all to its name. In that way, again, if cats are small dogs, then **cats are very peculiar dogs**.

We are left with one of two conclusions. Either cats are randomly the strangest collection of dogs in the world, or the initial assumption that cats are small dogs was wrong.

Hypothesis Tests

Hypothesis testing follows the exact same logic.

Hypothesis Tests

Hypothesis testing follows the exact same logic.

First we make an assumption about the data. Then we look at the data to see how compatible the data are with that assumption.

Hypothesis Tests

Hypothesis testing follows the exact same logic.

First we make an assumption about the data. Then we look at the data to see how compatible the data are with that assumption.

The initial assumption is called a **null hypothesis**.

Hypothesis Tests

Hypothesis testing follows the exact same logic.

First we make an assumption about the data. Then we look at the data to see how compatible the data are with that assumption.

The initial assumption is called a **null hypothesis**.

Based on what we see in the data, we will conclude either that

- the null hypothesis is wrong,
- or that we don't have enough evidence to conclude that the null hypothesis is wrong.

We don't ever conclude that the null hypothesis is true.

p-Values

A *p*-value is the probability that a test statistic could be as extreme as it is in the sample **under the assumption that the null hypothesis (no relationship, equal means, etc.) is true**.

p-Values

A *p*-value is the probability that a test statistic could be as extreme as it is in the sample **under the assumption that the null hypothesis (no relationship, equal means, etc.) is true**.

If the probability is really low, then <u>one of two things must be true</u>

p-Values

A p-value is the probability that a test statistic could be as extreme as it is in the sample **under the assumption that the null hypothesis (no relationship, equal means, etc.) is true**.

If the probability is really low, then one of two things must be true

1. the sample was **really, really extraordinary and unlikely**,

p-Values

A p-value is the probability that a test statistic could be as extreme as it is in the sample **under the assumption that the null hypothesis (no relationship, equal means, etc.) is true**.

If the probability is really low, then one of two things must be true

1. the sample was **really, really extraordinary and unlikely**,
2. or **the null hypothesis of no relationship or equal means is wrong**.

p-Values

A p-value is the probability that a test statistic could be as extreme as it is in the sample **under the assumption that the null hypothesis (no relationship, equal means, etc.) is true**.

If the probability is really low, then one of two things must be true

1. the sample was **really, really extraordinary and unlikely**,
2. or **the null hypothesis of no relationship or equal means is wrong**.

For very small values of p, we reject the possibility of the first option and go with the second, which we understand to mean that there is sufficient evidence of an effect or of different means.

Interpretation of p-values

Some common standards for rejecting the null hypothesis:

Interpretation of p-values

Some common standards for rejecting the null hypothesis:

- $p < .05$ – the most common standard in many fields

Interpretation of p-values

Some common standards for rejecting the null hypothesis:

- $p < .05$ – the most common standard in many fields
- $p < .01$ – a more conservative standard for concluding that x has an effect on y

Interpretation of p-values

Some common standards for rejecting the null hypothesis:

- $p < .05$ – the most common standard in many fields
- $p < .01$ – a more conservative standard for concluding that x has an effect on y
- $p < .1$ – a less conservative standard

Interpretation of p-values

Some common standards for rejecting the null hypothesis:

- $p < .05$ – the most common standard in many fields
- $p < .01$ – a more conservative standard for concluding that x has an effect on y
- $p < .1$ – a less conservative standard

If the standard is met, we say that a test statistic is "statistically significantly different from 0" although many researchers just say **"significant."**

Interpretation of p-values

Some common standards for rejecting the null hypothesis:

- $p < .05$ – the most common standard in many fields
- $p < .01$ – a more conservative standard for concluding that x has an effect on y
- $p < .1$ – a less conservative standard

If the standard is met, we say that a test statistic is "statistically significantly different from 0" although many researchers just say **"significant."**

Important: choose a standard before running any tests and stick with it. **Bending the standard to favor a conclusion is academically dishonest**.

MISinterpretation of p-values

Mistake 1: "type 1 error" — concluding that a hypothesis is false even though it is actually true.

MISinterpretation of p-values

Mistake 1: "type 1 error" — concluding that a hypothesis is false even though it is actually true.

$p = .05$ means there's only a **1/20 chance** that your t could have been as big as it is, assuming that the true test statistic is 0.

MISinterpretation of p-values

Mistake 1: "type 1 error" — concluding that a hypothesis is false even though it is actually true.

$p = .05$ means there's only a **1/20 chance** that your t could have been as big as it is, assuming that the true test statistic is 0.

But a 1/20 chance means that if you do 20 tests, you WOULD expect one on average to be unusual!

MISinterpretation of p-values

Mistake 1: "type 1 error" — concluding that a hypothesis is false even though it is actually true.

$p = .05$ means there's only a **1/20 chance** that your t could have been as big as it is, assuming that the true test statistic is 0.

But a 1/20 chance means that if you do 20 tests, you WOULD expect one on average to be unusual!

Some researchers do *test after test after test after test*. That will eventually lead you to claim that a **null relationship is significant**.

MISinterpretation of p-values

Mistake 2: "type 2 error" — concluding that a hypothesis is true even though it is actually false.

MISinterpretation of p-values

Mistake 2: "type 2 error" — concluding that a hypothesis is true even though it is actually false.

$p = .35$ means we cannot reject the null of independence between x and y. But that **does NOT mean that x and y actually are independent**!

MISinterpretation of p-values

Mistake 2: "type 2 error" — concluding that a hypothesis is true even though it is actually false.

$p = .35$ means we cannot reject the null of independence between x and y. But that **does NOT mean that x and y actually are independent**!

A null finding is not "no effect" but rather a lack of enough evidence to meet an arbitrary standard of $p < .05$. Don't write "has no effect" in your papers.

MISinterpretation of p-values

Mistake 3: interpreting the size of the p-values

MISinterpretation of p-values

Mistake 3: interpreting the size of the p-values

The size of a p-value says NOTHING about the size of the effect of x on y. **That's the test statistic!**

MISinterpretation of p-values

Mistake 3: interpreting the size of the p-values

The size of a p-value says NOTHING about the size of the effect of x on y. **That's the test statistic!**

It is possible for strong effects to have high p-values and it is possible for small effects to have low p-values.

MISinterpretation of p-values

Mistake 3: interpreting the size of the p-values

The size of a p-value says NOTHING about the size of the effect of x on y. **That's the test statistic!**

It is possible for strong effects to have high p-values and it is possible for small effects to have low p-values.

Don't fall into the traps of saying that one test is

MISinterpretation of p-values

Mistake 3: interpreting the size of the p-values

The size of a p-value says NOTHING about the size of the effect of x on y. **That's the test statistic!**

It is possible for strong effects to have high p-values and it is possible for small effects to have low p-values.

Don't fall into the traps of saying that one test is

- "more significant" than another,

MISinterpretation of p-values

Mistake 3: interpreting the size of the p-values

The size of a p-value says NOTHING about the size of the effect of x on y. **That's the test statistic!**

It is possible for strong effects to have high p-values and it is possible for small effects to have low p-values.

Don't fall into the traps of saying that one test is

- "more significant" than another,
- "strongly" or "marginally" significant,

MISinterpretation of p-values

Mistake 3: interpreting the size of the p-values

The size of a p-value says NOTHING about the size of the effect of x on y. **That's the test statistic!**

It is possible for strong effects to have high p-values and it is possible for small effects to have low p-values.

Don't fall into the traps of saying that one test is

- "more significant" than another,
- "strongly" or "marginally" significant,
- "increasingly significant" when we make changes.

MISinterpretation of p-values

Mistake 3: interpreting the size of the p-values

The size of a p-value says NOTHING about the size of the effect of x on y. **That's the test statistic!**

It is possible for strong effects to have high p-values and it is possible for small effects to have low p-values.

Don't fall into the traps of saying that one test is

- "more significant" than another,
- "strongly" or "marginally" significant,
- "increasingly significant" when we make changes.

A test result is **significant** or **not**. Don't interpret the size of p.

MISinterpretation of p-values

Mistake 4: placing too much emphasis on p-values

MISinterpretation of p-values

Mistake 4: placing too much emphasis on p-values

p-values depend on the size of the sample. The bigger the sample size, in general, the lower the p-values.

If we have a lot of data, then all of our test results will be significant! In that case, p-values don't tell us very much at all.

MISinterpretation of p-values

Mistake 4: placing too much emphasis on p-values

p-values depend on the size of the sample. The bigger the sample size, in general, the lower the p-values.

If we have a lot of data, then all of our test results will be significant! In that case, p-values don't tell us very much at all.

Also, p-values don't demonstrate that one feature causes another. There are a whole lot of other factors that we need to take into consideration to make a statement about causality.

Comparison of Means Tests

To compare the mean of a column to a pre-specified value, use a **one-sample t-test**:

Comparison of Means Tests

To compare the mean of a column to a pre-specified value, use a **one-sample** t**-test**:

```
mytest = stats.ttest_1samp(anes['ftbiden'].dropna(), 40)
mytest
```

```
Ttest_1sampResult(statistic=3.5913467876717964, pvalue=0.0003340593853690189)
```

Comparison of Means Tests

To compare the mean of a column to a pre-specified value, use a **one-sample *t*-test**:

```
mytest = stats.ttest_1samp(anes['ftbiden'].dropna(), 40)
mytest
```

```
Ttest_1sampResult(statistic=3.5913467876717964, pvalue=0.00033405938536901890)
```

The *p*-value is the probability that the sample could have a mean at least 2.152 units away from 40 if we assume that the true mean in the population is 40.

Comparison of Means Tests

To compare the mean of a column to a pre-specified value, use a **one-sample *t*-test**:

```
mytest = stats.ttest_1samp(anes['ftbiden'].dropna(), 40)
mytest
```

```
Ttest_1sampResult(statistic=3.5913467876717964, pvalue=0.0003340593853690189)
```

The *p*-value is the probability that the sample could have a mean at least 2.152 units away from 40 if we assume that the true mean in the population is 40.

Here the *p*-value is .0003, which is quite a bit smaller than the .05 standard we use to reject the null hypothesis. So we say that the mean thermometer rating for Joe Biden is statistically significantly different from 40.

Comparison of Means Tests

To compare the mean of a column across two groups, use an **independent-samples t-test**:

Comparison of Means Tests

To compare the mean of a column across two groups, use an **independent-samples *t*-test**:

```
ftbiden_men = anes.query("sex=='Male'").ftbiden.dropna()
ftbiden_women = anes.query("sex=='Female'").ftbiden.dropna()
stats.ttest_ind(ftbiden_men, ftbiden_women, equal_var=False)
```

```
Ttest_indResult(statistic=-4.684509884485571, pvalue=2.927022297618164e-06)
```

Comparison of Means Tests

To compare the mean of a column across two groups, use an **independent-samples *t*-test**:

```
ftbiden_men = anes.query("sex=='Male'").ftbiden.dropna()
ftbiden_women = anes.query("sex=='Female'").ftbiden.dropna()
stats.ttest_ind(ftbiden_men, ftbiden_women, equal_var=False)
```

```
Ttest_indResult(statistic=-4.684509884485571, pvalue=2.927022297618164e-06)
```

The *p*-value is about .0000002, which is the probability that under the assumption that men and women approve of Biden equally, on average, that we could draw a sample with a difference between these two means of 4.68 or higher.

Comparison of Means Tests

To compare the mean of a column across two groups, use an **independent-samples *t*-test**:

```
ftbiden_men = anes.query("sex=='Male'").ftbiden.dropna()
ftbiden_women = anes.query("sex=='Female'").ftbiden.dropna()
stats.ttest_ind(ftbiden_men, ftbiden_women, equal_var=False)
```

```
Ttest_indResult(statistic=-4.684509884485571, pvalue=2.927022297618164e-06)
```

The *p*-value is about .0000002, which is the probability that under the assumption that men and women approve of Biden equally, on average, that we could draw a sample with a difference between these two means of 4.68 or higher.

We reject the null hypothesis and conclude that there is a statisitically significant difference between men and women in terms of how highly they rate Joe Biden.

Comparison of Means Tests

To compare the means of two columns, use a **paired t-test**:

Comparison of Means Tests

To compare the means of two columns, use a **paired** ***t*****-test**:

```
anes_ttest = anes[['fttrump', 'ftbiden']].dropna()
stats.ttest_rel(anes_ttest['fttrump'], anes_ttest['ftbiden'])
```

```
Ttest_relResult(statistic=1.6327284676310017, pvalue=0.10262803725374475)
```

Comparison of Means Tests

To compare the means of two columns, use a **paired** *t*-**test**:

```
anes_ttest = anes[['fttrump', 'ftbiden']].dropna()
stats.ttest_rel(anes_ttest['fttrump'], anes_ttest['ftbiden'])
```

```
Ttest_relResult(statistic=1.6327284676310017, pvalue=0.10262803725374475)
```

The p-value, about 0.1, is the probability that a sample could have produced a difference in means of 1.72 or greater in either direction if the truth is that the columns have the same mean in the population.

Comparison of Means Tests

To compare the means of two columns, use a **paired *t*-test**:

```
anes_ttest = anes[['fttrump', 'ftbiden']].dropna()
stats.ttest_rel(anes_ttest['fttrump'], anes_ttest['ftbiden'])
```

```
Ttest_relResult(statistic=1.6327284676310017, pvalue=0.10262803725374475)
```

The *p*-value, about 0.1, is the probability that a sample could have produced a difference in means of 1.72 or greater in either direction if the truth is that the columns have the same mean in the population.

Because this *p*-value is greater than .05, we fail to reject the null hypothesis that the two candidates have the same average thermometer rating, which is is NOT the same thing as concluding the null hypothesis is true.

Tests of Multiple Comparisons

To compare the mean of a column across more than two groups, use an **analysis of variance (ANOVA) test with an f-test statistic**.

Tests of Multiple Comparisons

To compare the mean of a column across more than two groups, use an **analysis of variance (ANOVA) test with an f-test statistic**.

The null hypothesis in this case is that all of the groups have the same mean. If even one group has a different-enough mean, the null will be rejected:

Tests of Multiple Comparisons

To compare the mean of a column across more than two groups, use an **analysis of variance (ANOVA) test with an f-test statistic**.

The null hypothesis in this case is that all of the groups have the same mean. If even one group has a different-enough mean, the null will be rejected:

```
stats.f_oneway(anes.query("partyID=='Democrat'").age.dropna(),
               anes.query("partyID=='Independent'").age.dropna(),
               anes.query("partyID=='Republican'").age.dropna())
```

```
F_onewayResult(statistic=52.588970634465824, pvalue=3.517577203359592e-23)
```

Tests of Multiple Comparisons

To compare the mean of a column across more than two groups, use an **analysis of variance (ANOVA) test with an f-test statistic**.

The null hypothesis in this case is that all of the groups have the same mean. If even one group has a different-enough mean, the null will be rejected:

```
stats.f_oneway(anes.query("partyID=='Democrat'").age.dropna(),
               anes.query("partyID=='Independent'").age.dropna(),
               anes.query("partyID=='Republican'").age.dropna())
```

```
F_onewayResult(statistic=52.588970634465824, pvalue=3.517577203359592e-23)
```

The p-value is very small, and much smaller than .05, so we reject the null hypothesis that the three groups have the same average age.

Tests of Association

To test the relationship between two categorical features, we can test whether the row percents in a cross-tab are equal on each row (or whether the column percents are equal on each column). For example:

```
(pd.crosstab(anes.universal_income, anes.ideology, normalize='index')*100).round(2)
```

ideology	Liberal	Moderate	Conservative
universal_income			
Oppose a great deal	5.55	22.10	72.35
Oppose a moderate amount	22.89	36.32	40.80
Oppose a little	31.07	42.94	25.99
Neither favor nor oppose	32.50	43.27	24.24
Favor a little	40.95	36.19	22.86
Favor a moderate amount	50.99	33.77	15.23
Favor a great deal	56.02	28.57	15.41

Tests of Association

To test the relationship between two categorical features, we can test whether the row percents in a cross-tab are equal on each row (or whether the column percents are equal on each column). For example:

```
(pd.crosstab(anes.universal_income, anes.ideology, normalize='index')*100).round(2)
```

ideology	Liberal	Moderate	Conservative
universal_income			
Oppose a great deal	5.55	22.10	72.35
Oppose a moderate amount	22.89	36.32	40.80
Oppose a little	31.07	42.94	25.99
Neither favor nor oppose	32.50	43.27	24.24
Favor a little	40.95	36.19	22.86
Favor a moderate amount	50.99	33.77	15.23
Favor a great deal	56.02	28.57	15.41

The null hypothesis is that the row percents will be the same on every row.

Tests of Association

To test this null hypothesis, run a χ^2 (chi-square) test of association:

```
crosstab = pd.crosstab(anes.universal_income, anes.ideology)
stats.chi2_contingency(crosstab.values)
```

```
(849.5464372904162,
 3.8910750579483107e-174,
 12,
 array([[280.2137931 , 316.77827586, 394.00793103],
        [ 56.83448276,  64.25068966,  79.91482759],
        [ 50.04827586,  56.57896552,  70.37275862],
        [157.49655172, 178.04793103, 221.45551724],
        [ 89.06896552, 100.69137931, 125.23965517],
        [ 85.39310345,  96.53586207, 120.07103448],
        [100.94482759, 114.11689655, 141.93827586]]))
```

Tests of Association

To test this null hypothesis, run a χ^2 (chi-square) test of association:

```
crosstab = pd.crosstab(anes.universal_income, anes.ideology)
stats.chi2_contingency(crosstab.values)
```

```
(849.5464372904162,
 3.8910750579483107e-174,
 12,
 array([[280.2137931 , 316.77827586, 394.00793103],
        [ 56.83448276,  64.25068966,  79.91482759],
        [ 50.04827586,  56.57896552,  70.37275862],
        [157.49655172, 178.04793103, 221.45551724],
        [ 89.06896552, 100.69137931, 125.23965517],
        [ 85.39310345,  96.53586207, 120.07103448],
        [100.94482759, 114.11689655, 141.93827586]]))
```

The *p*-value represents the probability that a cross-tab with row-by-row (or column-by-column) differences as extreme as the ones we see if we assume that these two features are independent. We reject this null hypothesis.

Correlations with p-Values

We might want to use a hypothesis test to confirm that a correlation is not equal to 0, which would let us conclude that two features are correlated to some nonzero extent.

Correlations with p-Values

We might want to use a hypothesis test to confirm that a correlation is not equal to 0, which would let us conclude that two features are correlated to some nonzero extent.

We have to use a different function from the `scipy.stats` module to calculate this correlation:

```
anes_corr = anes[['fttrump', 'ftbiden']].dropna()
stats.pearsonr(anes_corr['fttrump'], anes_corr['ftbiden'])
```

```
(-0.6463572448004329, 0.0)
```

Correlations with p-Values

We might want to use a hypothesis test to confirm that a correlation is not equal to 0, which would let us conclude that two features are correlated to some nonzero extent.

We have to use a different function from the `scipy.stats` module to calculate this correlation:

```
anes_corr = anes[['fttrump', 'ftbiden']].dropna()
stats.pearsonr(anes_corr['fttrump'], anes_corr['ftbiden'])
```

```
(-0.6463572448004329, 0.0)
```

The p-value is the probability that a random sample could produce a correlation as extreme as .65 in either direction assuming that the correlation is 0 in the population.

Correlations with p-Values

We might want to use a hypothesis test to confirm that a correlation is not equal to 0, which would let us conclude that two features are correlated to some nonzero extent.

We have to use a different function from the `scipy.stats` module to calculate this correlation:

```
anes_corr = anes[['fttrump', 'ftbiden']].dropna()
stats.pearsonr(anes_corr['fttrump'], anes_corr['ftbiden'])
```

```
(-0.6463572448004329, 0.0)
```

The p-value is the probability that a random sample could produce a correlation as extreme as .65 in either direction assuming that the correlation is 0 in the population.

Because the p-value is so small, we reject the null hypothesis that these two features are uncorrelated.