Drawing Inferences from Data:

Topics include:
statistical tests, pivot tables, and Fisher's Exact test

Lecture 6

Statistical Hypothesis Tests

Test whether data support a give hypothesis or model.

For example:

"A £1 coin has equal probability of coming up heads or tails."

heads

tails

"A £1 coin has equal probability of coming up heads or tails."

heads

tails

By testing can we prove this statement to be true?

"A £1 coin has equal probability of coming up heads or tails."

heads

tails

By testing can we prove this statement to be true?

What if we do 10 trials and get 5 heads and 5 tails?

So, it looks like the statement is true.

"A £1 coin has equal probability of coming up heads or tails."

heads

tails

By testing can we prove this statement to be true?

What if we do 10 trials and get 5 heads and 5 tails?

So, it looks like the statement is true.

But, then we do 1,000,000 more trials and it turns out that the true probabilities are 55% heads and 45% tails.

In statistics, it is difficult to verify that a hypothesis is true.

"A £1 coin has equal probability of coming up heads or tails."

heads

tails

On the other hand, if the statement is false, can we show that?

If we were to filp this coin 1,000 times and get 763 heads and 237 tails, it's looking like the probabilities are not equal.

Some natural variation is expected, due to randomness of the individual filps. However, this is insufficient to explain the large difference between the heads and tails.

In statistics (and science), it would often be easier if hypotheses could be proven true.

However, we are often forced to work in the ***reverse*** direction, by ***falsifying*** hypothesis.

> *"In so far as a scientific statement speaks about reality, it must be falsifiable"*
>
> *– Karl Popper*

If a hypothesis is falsified with increasingly sensitive statistical tests, it *may* be true. (Or at least approximately true.)

Null Hypothesis Test

In statistics, we often want to determine whether something has an effect on data.

For example, in medicine, we may want to learn whether patients who get a particular treatment have a better outcome than patient who don't.

However, it may be difficult to directly test this, so we have to approach this problem from another direction.

Null Hypothesis Test

The "null hypothesis" is the hypothesis that there the treatment has not effect. (This is often denoted H_0.)

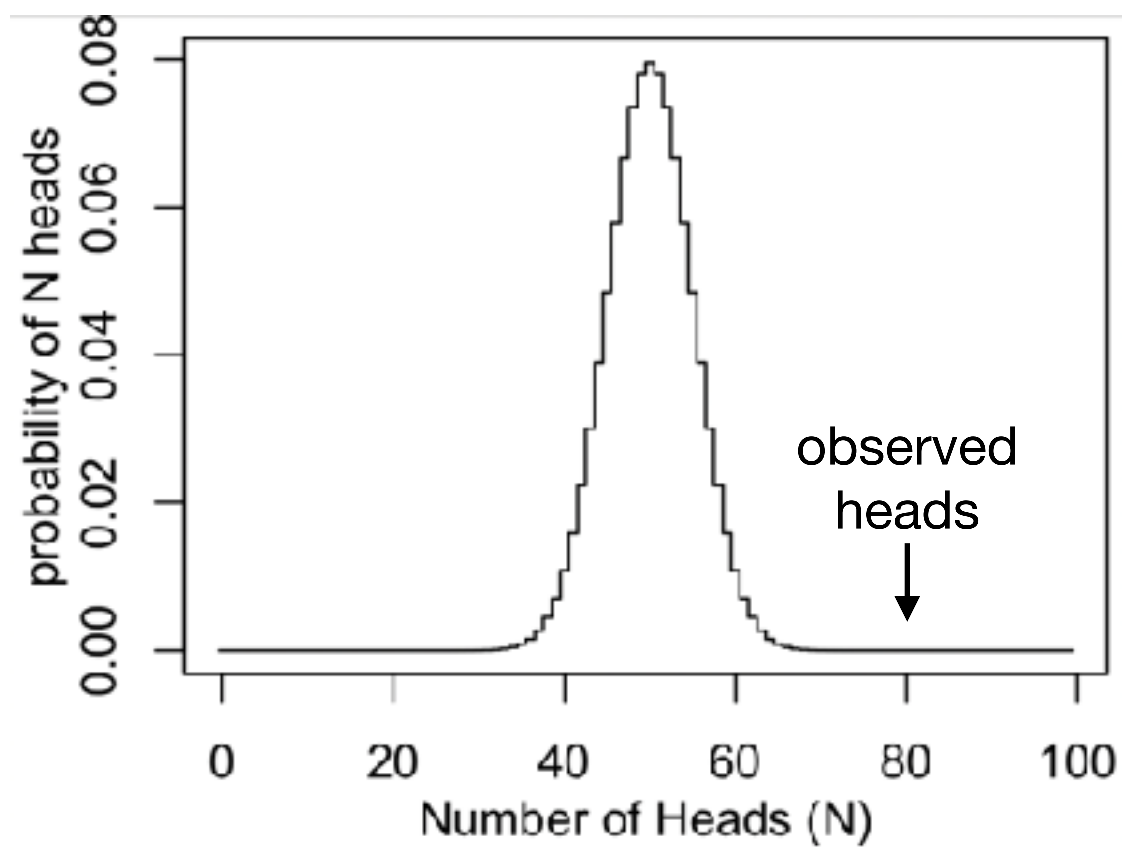
The "alternative hypothesis" is that the treatment has an effect on the data. (Often denoted H_1.)

We can more make a prediction ***assuming*** the null hypothesis. If there is a large mismatch between the prediction and the observed value (i.e., the probability of getting the observed value by change is very small), we ***reject*** the null hypothesis, implying the alternative hypothesis is correct.

The coin example

Our null hypothesis is that a coin is has a 50% probability of heads and a 50% probability of tails.

Let's filp this coin 100 times. The probability of getting N heads is given by the following binomial distribution, where 50 is the peak of the distribution.



Suppose we observe 80 heads. Given the distribution above, the probability of observing 80 *or more* is $p=5\times10^{-10}$. This probability is called the ***p-value***.

This will effectively never happen by random chance, so, if we observed 80 heads, we can reject thi s

Fischer's Exact Test for Contingency Tables

Student preparation	Test outcome: Passed	Failed	Row Total
Studied	45	10	55
Did not study	9	10	19
Column Total	54	20	74

Fischer's Exact Test for Contingency Tables

What is our null hypothesis?

There is no association between the two categorical variables. This implies that the probability of an observation falling into a particular category of one variable is independent of its classification in the other variable.

What is our alternate hypothesis?

There is an association between the two categorical variables.

We test this by assuming that the null hypothesis is true. Then, given this assumption, calculate the probability of obtaining the observed result.

If the probability of the total observed result or any more extremem result is very low, we can reject the null hypothesis. If the probability is not low, we say that our results are "consistent with the null hypothesis" and that no effect is detected.

Fischer's Exact Test for Contingency Tables

Student preparation	Test outcome		
	Passed	**Failed**	**Row Total**
Studied	a	b	a+b
Did not study	c	d	c+d
Column Total	a+c	b+d	n=a+b+c+d

$$p = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}} = \frac{\binom{a+b}{b}\binom{c+d}{d}}{\binom{n}{b+d}} = \frac{(a+b)!\;(c+d)!\;(a+c)!\;(b+d)!}{a!\;\;b!\;\;c!\;\;d!\;\;n!}$$

$$p_{\text{value}} = \sum\nolimits_{\text{all tables as or more extreme}} p(\text{table})$$

		Test outcome		
		Passed	**Failed**	**Row Total**
Student preparation	**Studied**	a	b	a+b
	Did not study	c	d	c+d
	Column Total	a+c	b+d	n=a+b+c+d

P values

Let's say we decide to use a probability threshold of 0.05 to reject the null hypothesis.

Which of the following would demonstrate a statistically significant effect?

$p=0.06$

$p=0.99$

$p=0.5$

$p=0.001$

$p=0.0001$

P values

Let's say we decide to use a probability threshold of 0.05 to reject the null hypothesis.

Which of the following would demonstrate a statistically significant effect?

p=0.06

p=0.99

p=0.5

p=0.001

p=0.0001

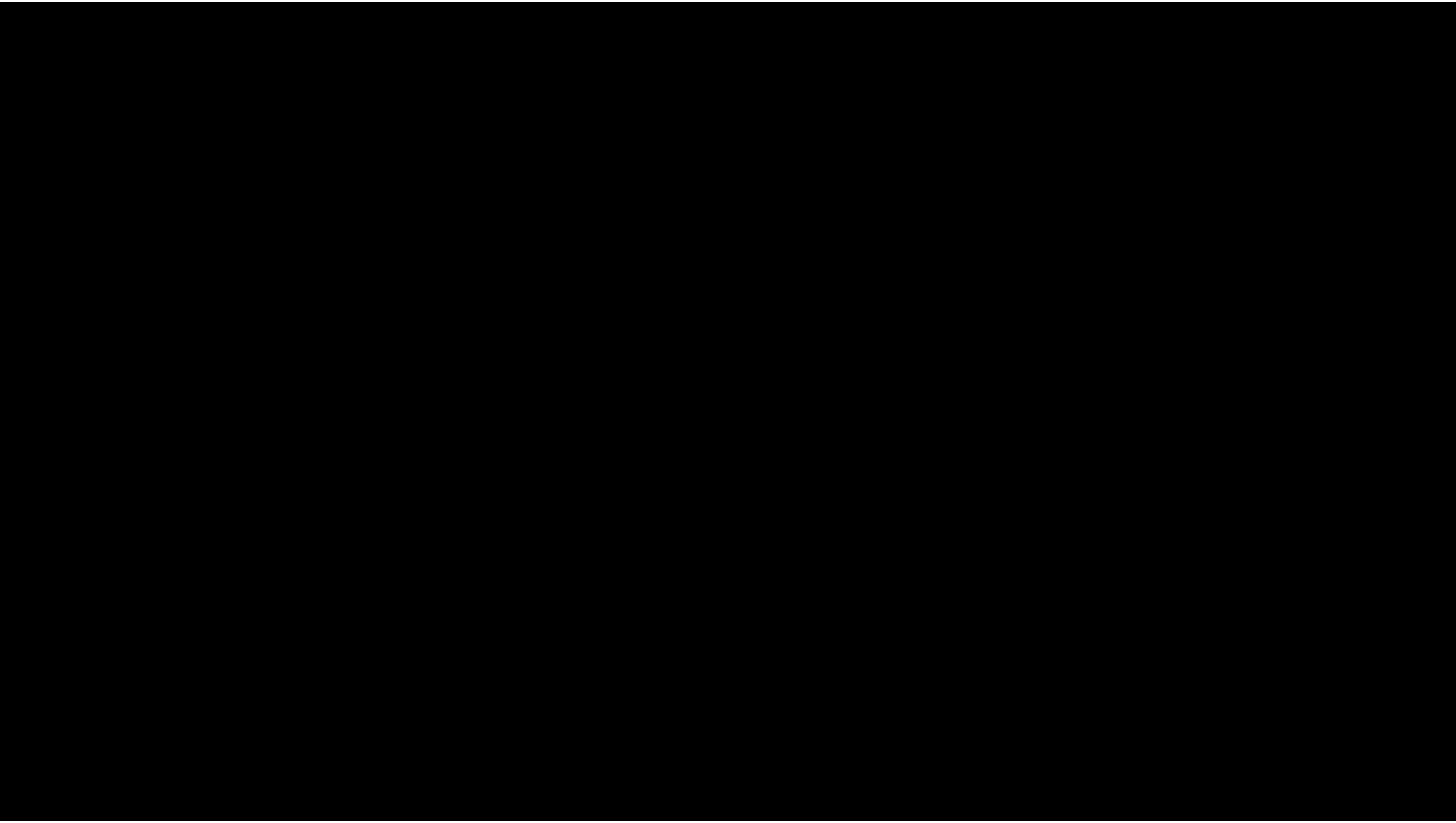
Fischer's Exact Test for Contingency Tables

		Test outcome		
		Passed	Failed	Row Total
Student preparation	Studied	45	10	55
	Did not study	9	10	19
	Column Total	54	20	74

```
In [1]: import pandas as pd
   ...: from scipy.stats import fisher_exact

In [2]: data = {
   ...:     'Passed': [45, 9],
   ...:     'Failed': [10, 10]
   ...: }

In [3]: df = pd.DataFrame(data, index=['Studied', 'Did_Not_Study'])

In [4]: print(df)
               Passed  Failed
Studied            45      10
Did_Not_Study       9      10

In [5]: odds_ratio, p_value = fisher_exact(df.values)
   ...:
   ...: print(f"Odds Ratio: {odds_ratio:.4f}, P-value: {p_value:.4f}")

Odds Ratio: 5.0000, P-value: 0.0064
```

And now for something completely different

Pivot Tables



Pandas Pivot Tables

Get mean value for *col3*,
indexed by *col1* and *col2*.

```
df.groupby(['col1', 'col2'])['col3'].aggregate('mean').unstack()
```

```
df.pivot_table('col3', index='col1', columns='col2')
```


Willy Stöwer
1912

The Titanic Dataset

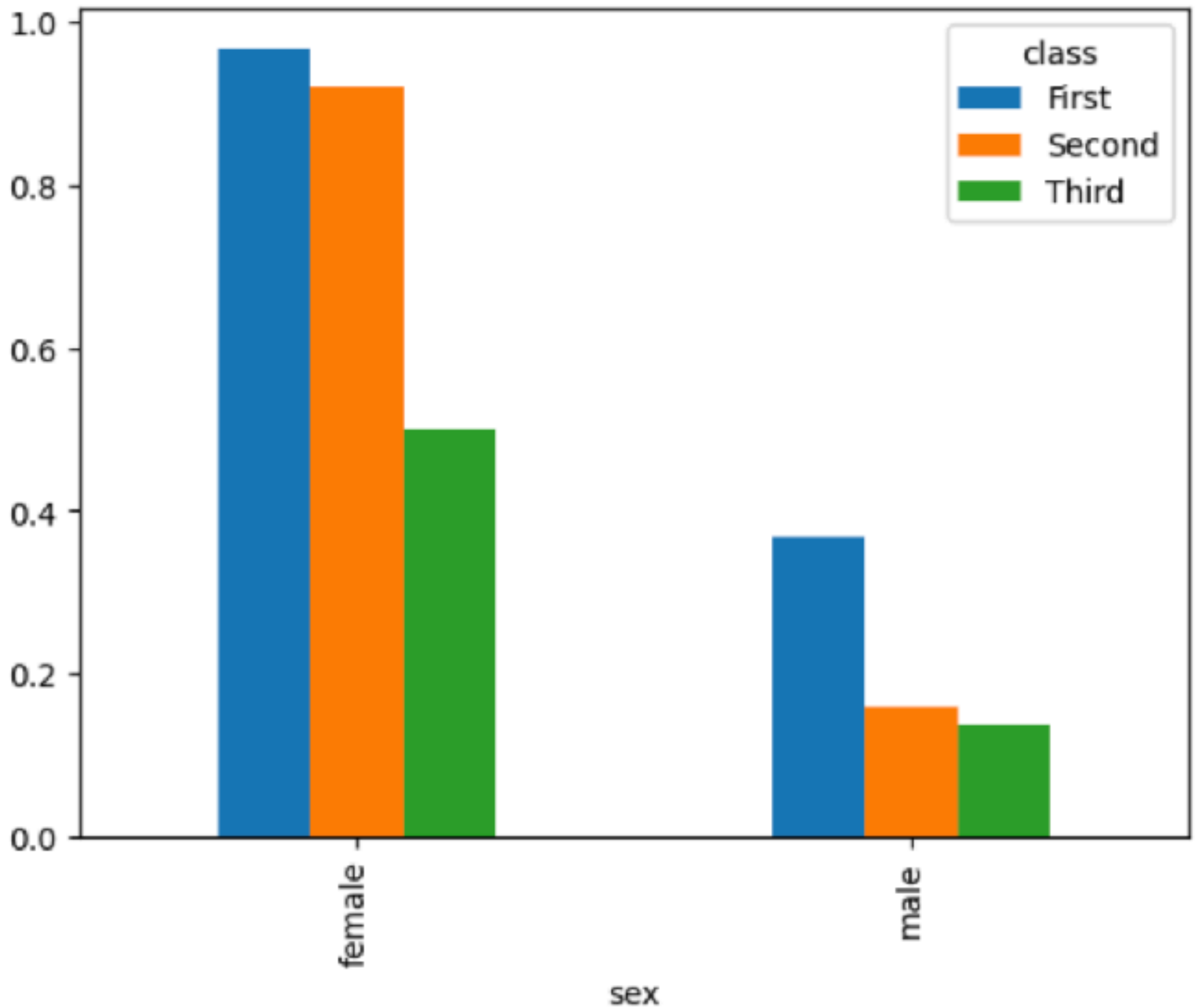
```
In [1]: import numpy as np
   ...: import pandas as pd
   ...: import seaborn as sns
   ...: titanic = sns.load_dataset('titanic')

In [2]: titanic
Out[2]:
     survived     sex   age     fare embarked   class    who  adult_male  embark_town alive  alone
0           0    male  22.0   7.2500        S   Third    man        True  Southampton    no  False
1           1  female  38.0  71.2833        C   First  woman       False    Cherbourg   yes  False
2           1  female  26.0   7.9250        S   Third  woman       False  Southampton   yes   True
3           1  female  35.0  53.1000        S   First  woman       False  Southampton   yes  False
4           0    male  35.0   8.0500        S   Third    man        True  Southampton    no   True
..        ...     ...   ...      ...      ...     ...    ...         ...          ...   ...    ...
886         0    male  27.0  13.0000        S  Second    man        True  Southampton    no   True
887         1  female  19.0  30.0000        S   First  woman       False  Southampton   yes   True
888         0  female   NaN  23.4500        S   Third  woman       False  Southampton    no  False
889         1    male  26.0  30.0000        C   First    man        True    Cherbourg   yes   True
890         0    male  32.0   7.7500        Q   Third    man        True   Queenstown    no   True

[891 rows x 11 columns]
```

I removed some columns for better display.

Comparing with GroupBy

```
In [8]: titanic.groupby(['sex', 'class'])['survived'].aggregate('mean').unstack()
   ...:
Out[8]:
class      First    Second     Third
sex
female  0.968085  0.921053  0.500000
male    0.368852  0.157407  0.135447

In [9]: titanic.pivot_table('survived', index='sex', columns='class')
   ...:
Out[9]:
class      First    Second     Third
sex
female  0.968085  0.921053  0.500000
male    0.368852  0.157407  0.135447
```

```
In [21]: titanic.pivot_table('survived', index=['sex'], columns=['class'],
aggfunc='mean').plot.bar()
Out[21]: <AxesSubplot:xlabel='sex'>

In [22]: plt.show()
```



Multi-level Pivot Tables

```
In [23]: age = pd.cut(titanic['age'], [0, 18, 80])  # slices "age" column
    ...: age
Out[23]:
0       (18.0, 80.0]
1       (18.0, 80.0]
2       (18.0, 80.0]
3       (18.0, 80.0]
4       (18.0, 80.0]
            ...
886     (18.0, 80.0]
887     (18.0, 80.0]
888              NaN
889     (18.0, 80.0]
890     (18.0, 80.0]
Name: age, Length: 891, dtype: category
Categories (2, interval[int64, right]): [(0, 18] < (18, 80]]

In [24]: titanic.pivot_table('survived', ['sex', age], 'class')
Out[24]:
class               First    Second     Third
sex    age
female (0, 18]   0.909091  1.000000  0.511628
       (18, 80]  0.972973  0.900000  0.423729
male   (0, 18]   0.800000  0.600000  0.215686
       (18, 80]  0.375000  0.071429  0.133663
```

Multi-level Pivot Tables

```
In [25]: titanic.pivot_table('alone', 'sex', ['embark_town','survived'])
Out[25]:
embark_town Cherbourg           Queenstown           Southampton
survived            0         1          0         1           0         1
sex
female       0.222222  0.359375   0.666667  0.777778    0.301587  0.378571
male         0.696970  0.482759   0.736842  0.666667    0.750000  0.623377
```

Let's make a contingency table

Test whether 'sex' has an effect on 'survival'.

```
In [33]: test_tab = titanic.pivot_table(values=  ?  , index= ? ,
columns=  ?    , aggfunc=  ?  )
```

Let's make a contingency table

Test whether 'sex' has an effect on 'survival'.

```
In [33]: test_tab = titanic.pivot_table(values=   ?   , index='sex',
columns=   ?     , aggfunc=   ?   )
```

Let's make a contingency table

Test whether 'sex' has an effect on 'survival'.

```
In [33]: test_tab = titanic.pivot_table(values=   ?   , index='sex',
columns='survived', aggfunc=   ?   )
```

Let's make a contingency table

Test whether 'sex' has an effect on 'survival'.

```
In [33]: test_tab = titanic.pivot_table(values=   ?   , index='sex',
columns='survived', aggfunc='count')
```

Let's make a contingency table

Test whether 'sex' has an effect on 'survival'.

```
In [33]: test_tab = titanic.pivot_table(values='class', index='sex',
columns='survived', aggfunc='count')
```

```
In [34]: print(test_tab)
survived    0    1
sex
female     81  233
male      468  109
```

Let's make a contingency table

Test whether 'sex' has an effect on 'survival'.

```
In [33]: test_tab = titanic.pivot_table(values='class', index='sex',
columns='survived', aggfunc='count')
```

```
In [34]: print(test_tab)
survived    0    1
sex
female     81  233
male      468  109
```

```
In [35]: odds_ratio, p_value = fisher_exact(test_tab.values)
In [36]: print(p_value)
6.463921564583144e-60
```

Let's make a contingency table

Test whether 'sex' has an effect on 'survival'.

```
In [33]: test_tab = titanic.pivot_table(values='class', index='sex',
columns='survived', aggfunc='count')
```

```
In [34]: print(test_tab)
survived    0    1
sex
female     81  233
male      468  109
```

Is this statistically significant?

```
In [35]: odds_ratio, p_value = fisher_exact(test_tab.values)
In [36]: print(p_value)
6.463921564583144e-60
```