Fondamenti di Analisi dei Dati

from data analysis to predictive techniques

Prof. Antonino Furnari (antonino.furnari@unict.it)
Corso di Studi in Informatica
Dip. di Matematica e Informatica
Università di Catania

Università di Catania

Association Between Variables

Understanding relationships and interactions between multiple variables in exploratory data analysis

From Univariate to Multivariate Analysis

01

Univariate Analysis

Examining variables individually, one at a time

Example: Looking at a class's average test scores alone.

02

Multivariate Analysis

Studying interactions between multiple variables simultaneously

Example: Analyzing how test scores change based on the number of study hours.

03

Association

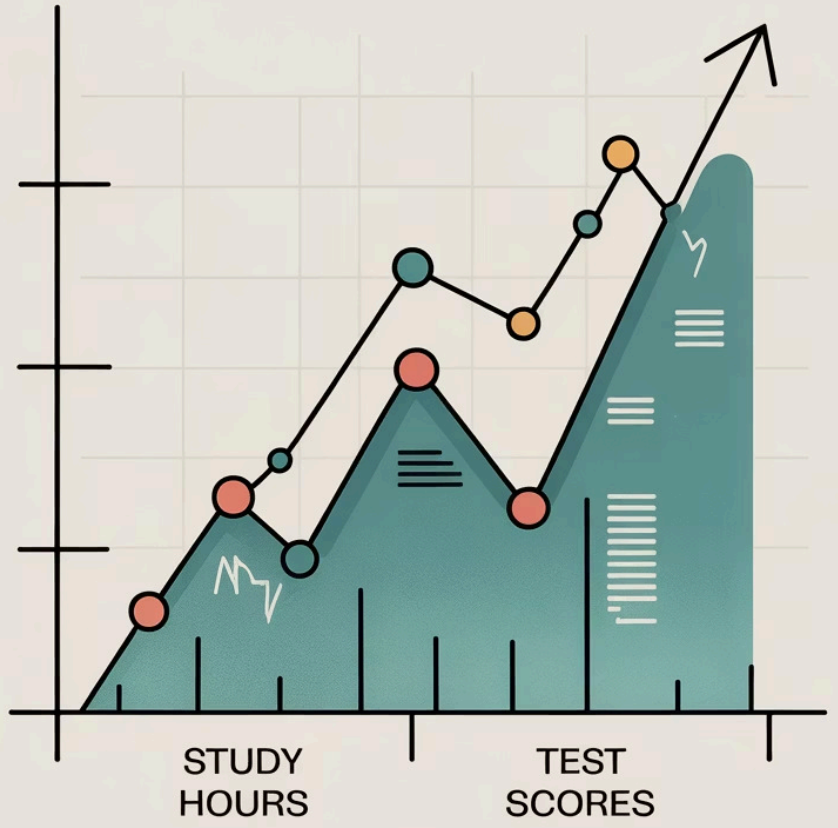
Observing one variable reveals information about another

Example: Discovering that knowing a student's study hours helps predict their test performance.

What Is Association?

> Two variables are associated if observing the value of one variable tells us something about the value of the other variable

This concept is closely related to **dependence in probability**. If two variables are independent, they typically show no statistical association.



Examples of Associated Variables

Height & Weight

Taller individuals tend to weigh more than shorter individuals

Education & Income

Higher education levels are often linked to higher earnings

Age & Blood Pressure

Older individuals tend to have elevated blood pressure

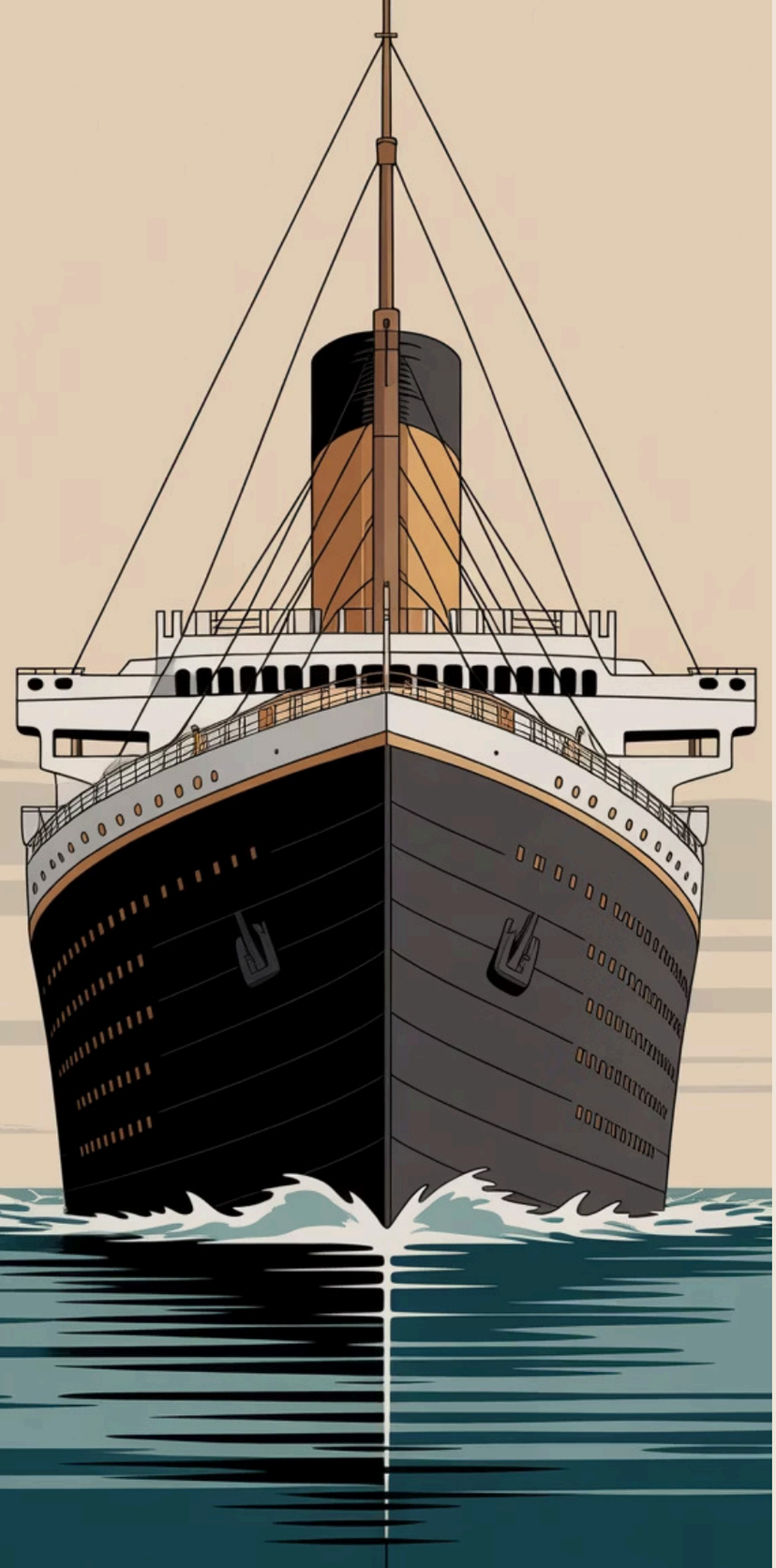
Study Time & Performance

More hours studied typically correlate with better exam scores

Correlation ≠ Causation

Important: Two variables may be statistically associated without one directly causing the other. Confounding factors or reverse causality may be at play.

For instance, **ice cream sales** and **drowning incidents** are positively correlated, but ice cream doesn't cause drowning. **Both increase during summer months due to hot weather** (the confounding variable).



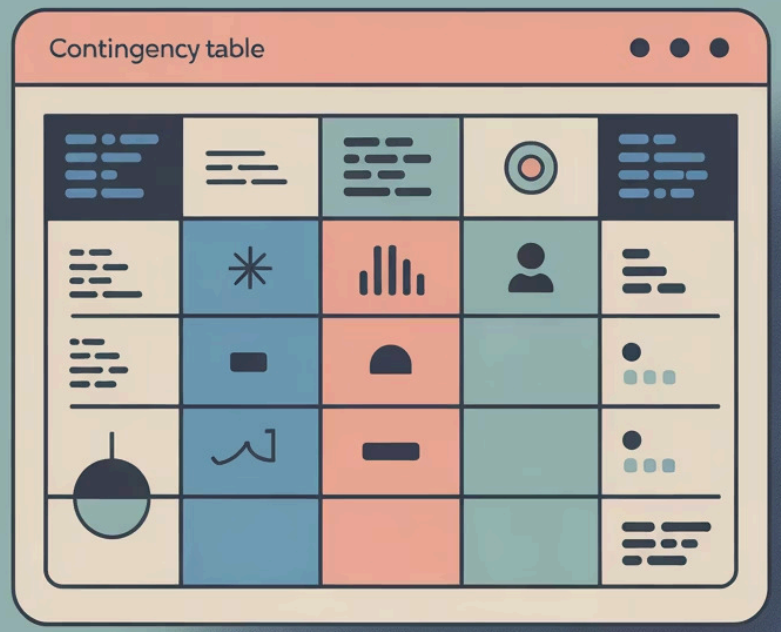
The Titanic Dataset

We'll use the Titanic dataset to explore associations between variables such as passenger class, survival rate, age, and fare.

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Key Questions

- Does passenger class influence survival probability?
- Does age affect survival chances?
- Does fare paid correlate with survival?



Discrete Variables

Measures of Association

We will start by looking at measures of association for discrete variables.

Conditional Frequency Distributions

Examining how survival rates vary across different passenger classes using conditional frequencies:

f(Survived | Pclass=1)

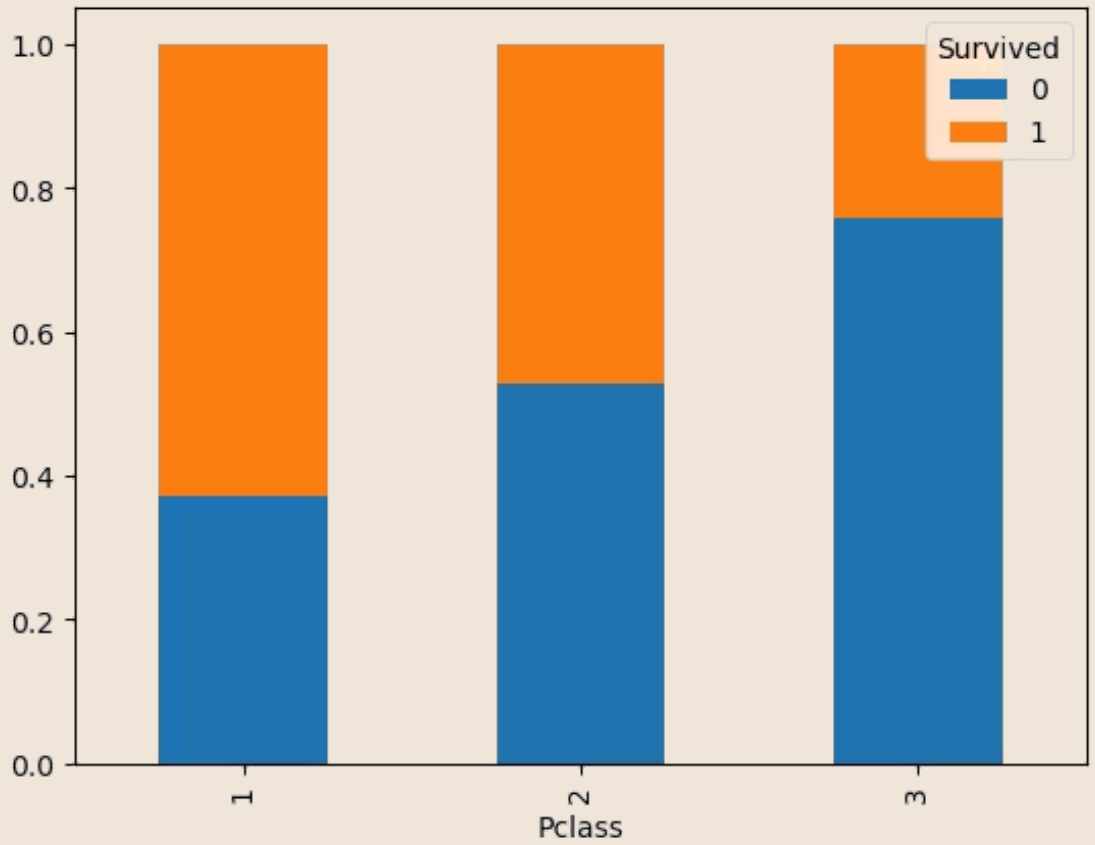
First class passengers

f(Survived | Pclass=2)

Second class passengers

f(Survived | Pclass=3)

Third class passengers



The following table shows the conditional probabilities of survival (1) and death (0) given each passenger class:

1	0.370370	0.629630
2	0.527174	0.472826
3	0.757637	0.242363

This table demonstrates how **survival rates vary significantly across passenger classes**, with first class having the highest survival rate (62.96%) and third class having the lowest (24.24%).

Contingency Table and Correlation

	Y=y_1	**Y=y_2**	**...**	**Y=y_l**	**Total**
X=x_1	n_{11}	n_{12}	...	n_{1l}	n_{1+}
X=x_2	n_{21}	n_{22}	...	n_{2l}	n_{2+}
...	...	...	...	...	...
X=x_k	n_{k1}	n_{k2}	...	n_{kl}	n_{k+}
Total	n_{+1}	n_{+2}	...	n_{+l}	n

Let's now imagine to drop all inner terms. These are the ones characterising the correlation (how a variable influence the other one):

	Y=y_1	**Y=y_2**	**...**	**Y=y_l**	**Total**
X=x_1			...		n_{1+}
X=x_2			...		n_{2+}
...	...	...	...	...	...
X=x_k			...		n_{k+}
Total	n_{+1}	n_{+2}	...	n_{+l}	n

How would we reconstruct the internal numbers if the variables where **independent?**

Independence & Expected Frequencies

Two variables are **independent** if observing one provides no information about the other.

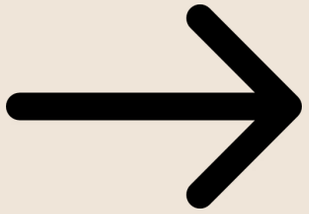
Under independence, joint probability follows:

$$P(X = x_i, Y = y_j) = P(X = x_i) \cdot P(Y = y_j)$$

Expected frequencies can be reconstructed from marginal totals:

$$\tilde{n}_{ij} = \frac{n_{i+} \cdot n_{+j}}{n}$$

Survived	0	1	All
Pclass			
1	80	136	216
2	97	87	184
3	372	119	491
All	549	342	891



Survived	0	1	All
Pclass			
1	133.090909	82.909091	216.0
2	113.373737	70.626263	184.0
3	302.535354	188.464646	491.0
All	549.000000	342.000000	891.0

Pearson's χ^2 Statistic

Measures discrepancies between observed and expected frequencies:

$$\chi^2 = \sum_{i=1}^{k} \sum_{j=1}^{l} \frac{(n_{ij} - \tilde{n}_{ij})^2}{\tilde{n}_{ij}}$$

This measure is reliable when each entry of the contingency table larger than or equal to 5.

Scaling by expected frequencies ensures small discrepancies in small samples weigh appropriately. If observed = expected, then $\chi^2 = 0$.

```
from scipy.stats import chi2_contingency
contingency = pd.crosstab(titanic['Pclass'], titanic['Survived'])
chi2_contingency(contingency)
```

```
Chi2ContingencyResult(statistic=np.float64(102.88898875696056), pvalue=np.float64(4.549251711298793e-23)
       [113.37373737,  70.62626263],
       [302.53535354, 188.46464646]]))
```

Cramér's V Statistic

Normalised version of Pearson's χ^2 that adjusts for sample size:

$$V = \sqrt{\frac{\chi^2}{n \cdot (min(k, l) - 1)}}$$

```
from scipy.stats.contingency import association

print(f"{association(pd.crosstab(titanic['Pclass'], titanic['Survived'])):0.2f}")
```

```
0.34
```

No Association

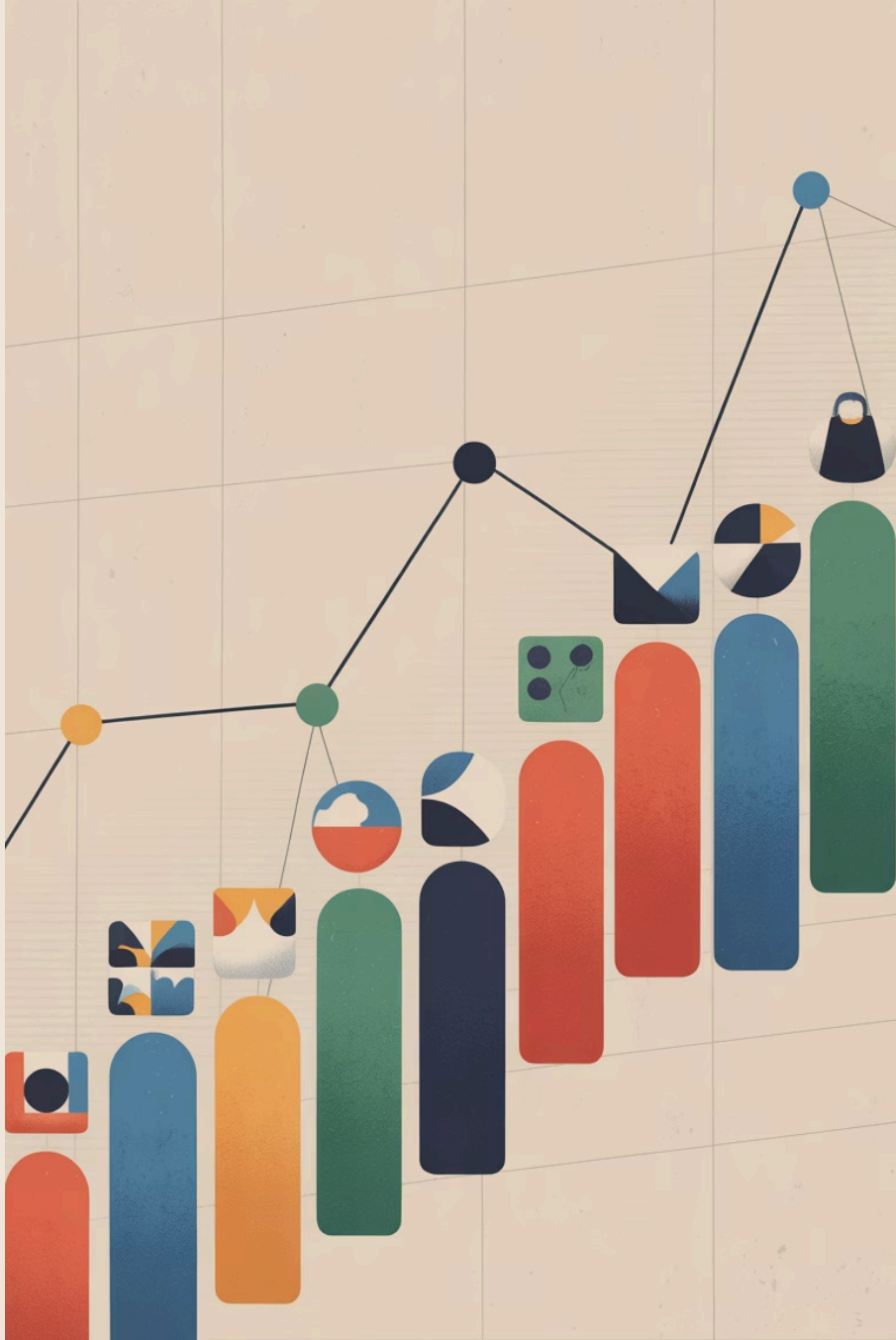
Weak Association

Titanic Pclass vs Survived

Perfect Association

Continuous Variables

Visualisation & Correlation



Diabetes Dataset

```
from statsmodels.datasets import get_rdataset
data = get_rdataset('Diabetes','heplots').data
data.head()
```

	relwt	glufast	glutest	instest	sspg	group
0	0.81	80	356	124	55	Normal
1	0.95	97	289	117	76	Normal
2	0.94	105	319	143	105	Normal
3	1.04	90	356	199	108	Normal
4	1.00	90	323	240	143	Normal

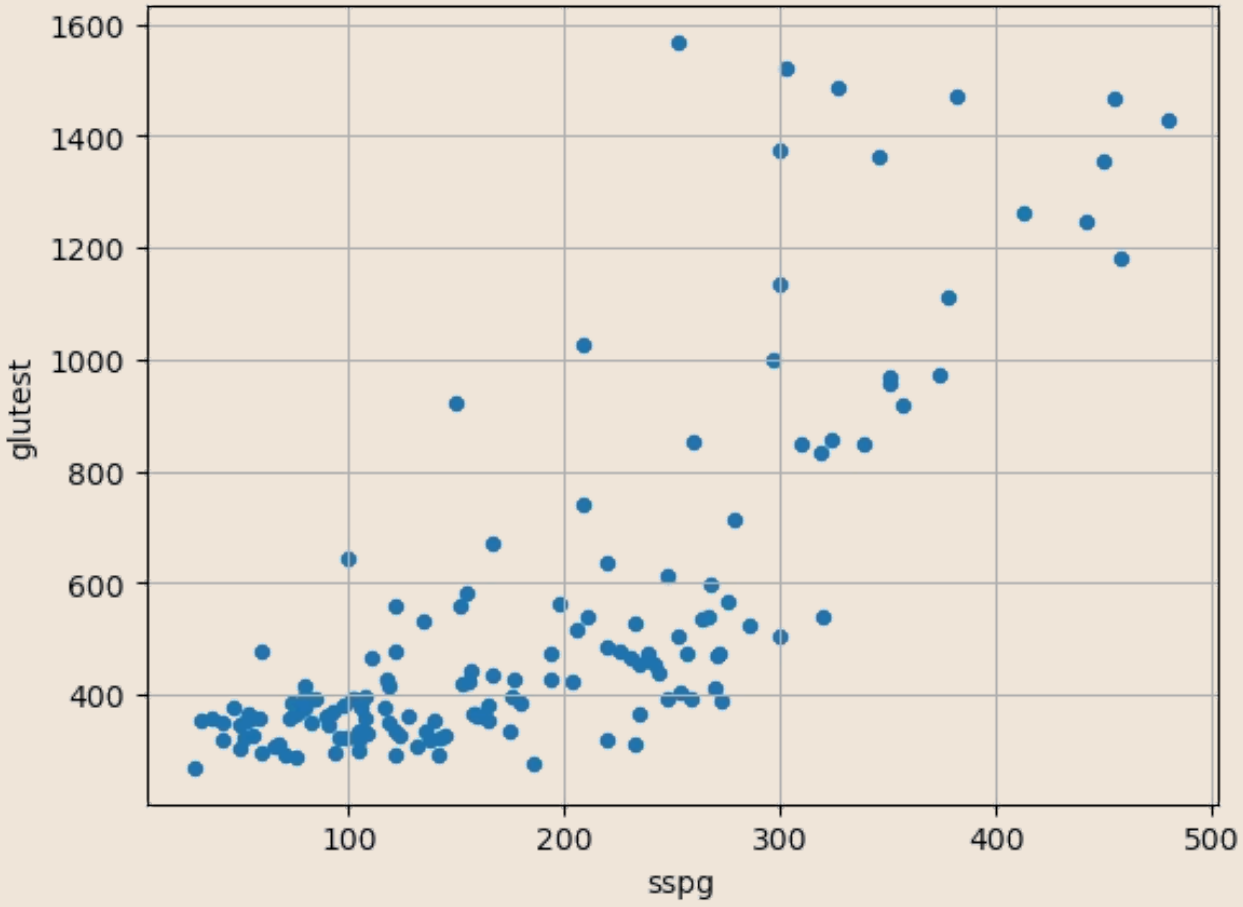
The 6 variables have the following meanings:

- `relwt` relative weight, expressed as the ratio of actual weight to expected weight, given the person's height, a numeric vector
- `glufast` fasting plasma glucose level, a numeric vector
- `glutest` test plasma glucose level, a measure of glucose intolerance, a numeric vector
- `instest` plasma insulin during test, a measure of insulin response to oral glucose, a numeric vector
- `sspg` steady state plasma glucose, a measure of insulin resistance, a numeric vector
- `group` diagnostic group, a factor with levels `Normal` `Chemical_Diabetic` `Overt_Diabetic`

Scatterplots & Scatter Matrix

Visualising bivariate relationships through scatterplots reveals patterns:

- Positive correlation (upward trend)
- Negative correlation (downward trend)
- Linear vs non-linear relationships
- No apparent correlation



Scatter Matrix

```
from matplotlib import pyplot as plt
import seaborn as sns
sns.pairplot(data)
plt.show()
```

Scatter Matrix with Hue

```
from matplotlib import pyplot as plt
import seaborn as sns
sns.pairplot(data, hue='group')
plt.show()
```

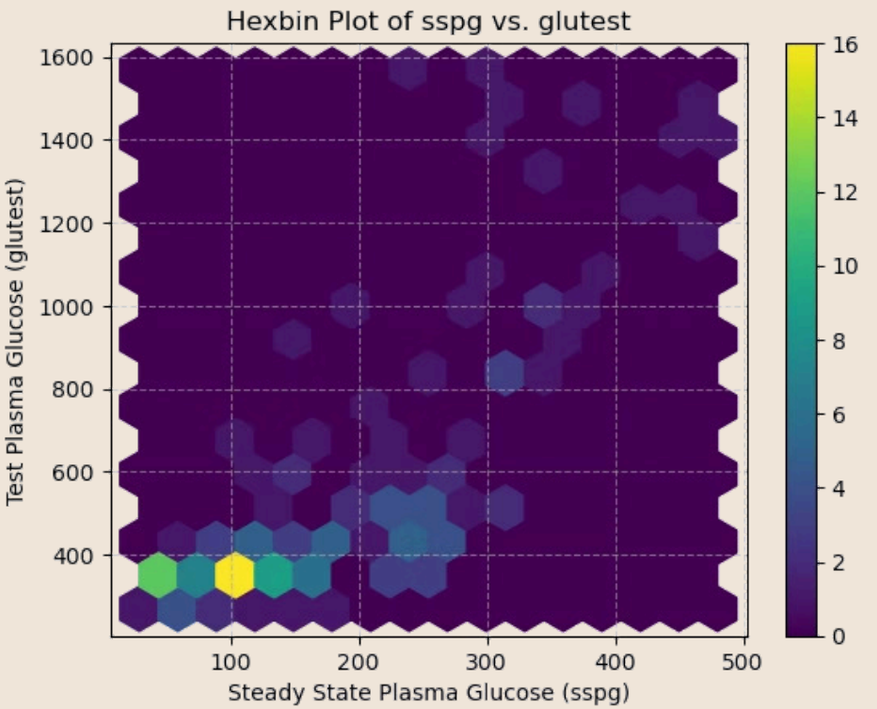
Hexbin Plots: A 2D Histogram

When scatterplots have a large number of data points, **overplotting** can occur, obscuring the true density of the data. Points may overlap, making it difficult to discern areas of high concentration.

> A **hexbin plot** addresses this by dividing the 2D space into hexagonal bins, coloring each based on the number of data points it contains. Think of it as a two-dimensional histogram for visualizing density.

We can generate an hexbin plot with the `hexbin` function:

```
from matplotlib import pyplot as plt
# let's use the hexbin function
# cmap is the color map
# gridsize regulates the number of hexagons
data.plot.hexbin(x='sspg', y='glutest', gridsize=15, cmap='viridis')
plt.xlabel('Steady State Plasma Glucose (sspg)')
plt.ylabel('Test Plasma Glucose (glutest)')
plt.title('Hexbin Plot of sspg vs. glutest')
plt.grid(linestyle='--', alpha=0.6)
plt.show()
```



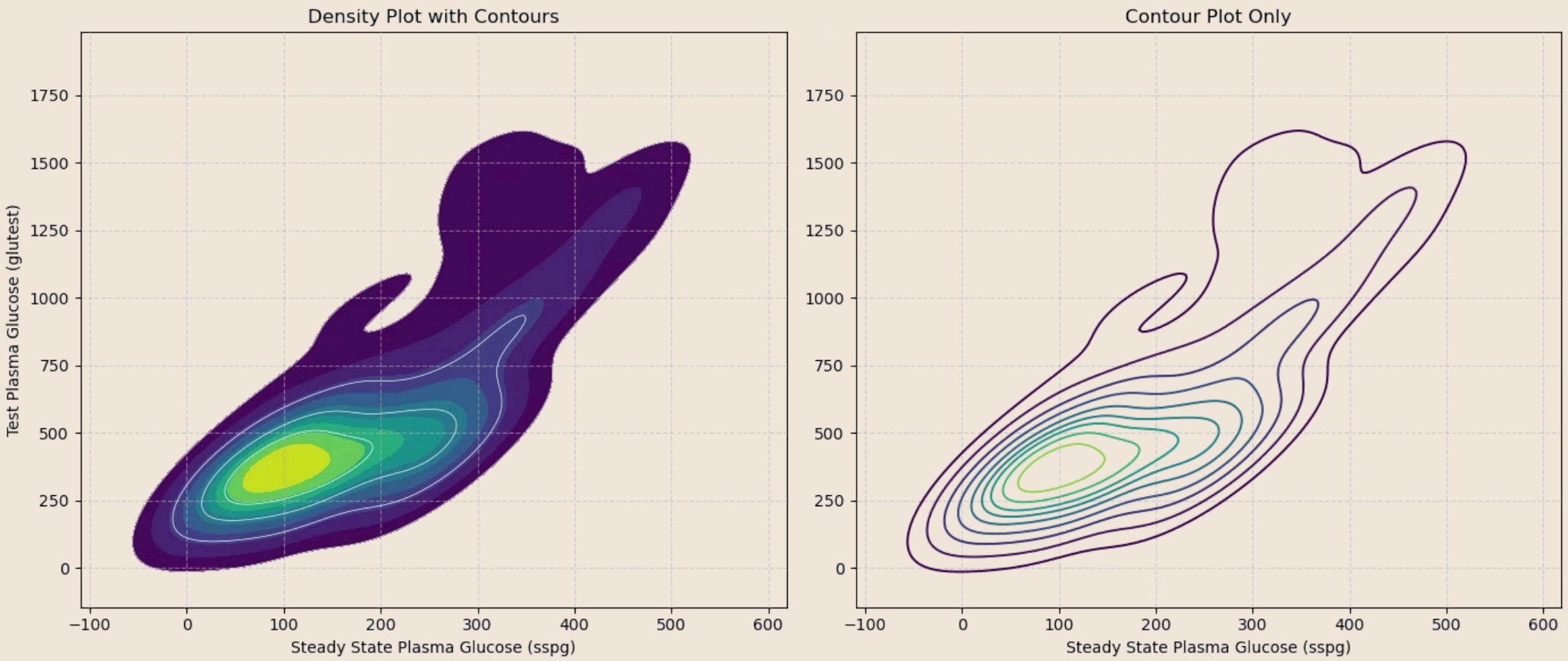
Density and Contour Plots

A Smoother View of Distribution

We can extend Kernel Density Estimation (KDE) to two variables, creating a smoother alternative to binned plots like hexbins. A **2D density plot** visualises data point probability density using a continuous colour gradient. Overlaying **contour lines** connects points of equal density, much like a topographical map.

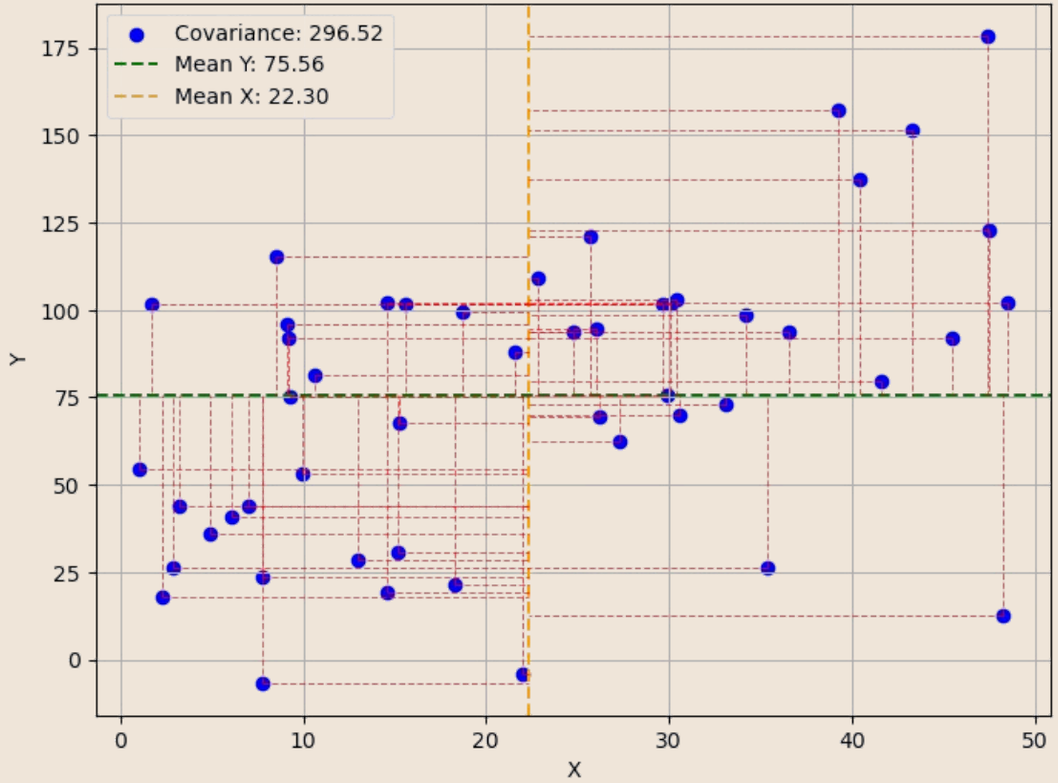
This method excels at identifying the shape and peaks of a bivariate distribution without being constrained by a grid.

```
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# density (fill=True)
sns.kdeplot(data=data, x='sspg', y='glutest', fill=True, cmap='viridis', ax=axes[0])
# contour plot (no fill)
sns.kdeplot(data=data, x='sspg', y='glutest', levels=5, color='white', linewidths=0.5, ax=axes[0])

# only contour (no fill)
sns.kdeplot(data=data, x='sspg', y='glutest', cmap='viridis', ax=axes[1])
```



The left plot's colour-filled areas intuitively map data density, highlighting a single, dense cluster in the lower-left corner. The contour-only plot on the right is particularly useful for comparing multiple distributions on the same axes without visual clutter.

Covariance



Measures how two variables vary together around their means:

$$Cov(X,Y) = \frac{1}{n}\sum_{i}^{n}(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})$$

Note that we can obtain an unbiased estimation of the covariance similar to the variance as follows:

$$Cov(X,Y) = \frac{1}{n-1}\sum_{i}^{n}(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})$$

Positive Terms - Quadrants 1,3

Both values above or both below their means (concordant). These contribute positively to the covariance.

Negative Terms - Quadrants 2-4

One above, one below their means (discordant). These contribute negatively to the covariance.

Near-Zero Terms

Values close to their respective means. These do not contribute much to the covariance.

Covariance Matrix

When we have multiple variables $X = (X_1, \ldots, X_k)$, we can compute a covariance matrix Σ such that

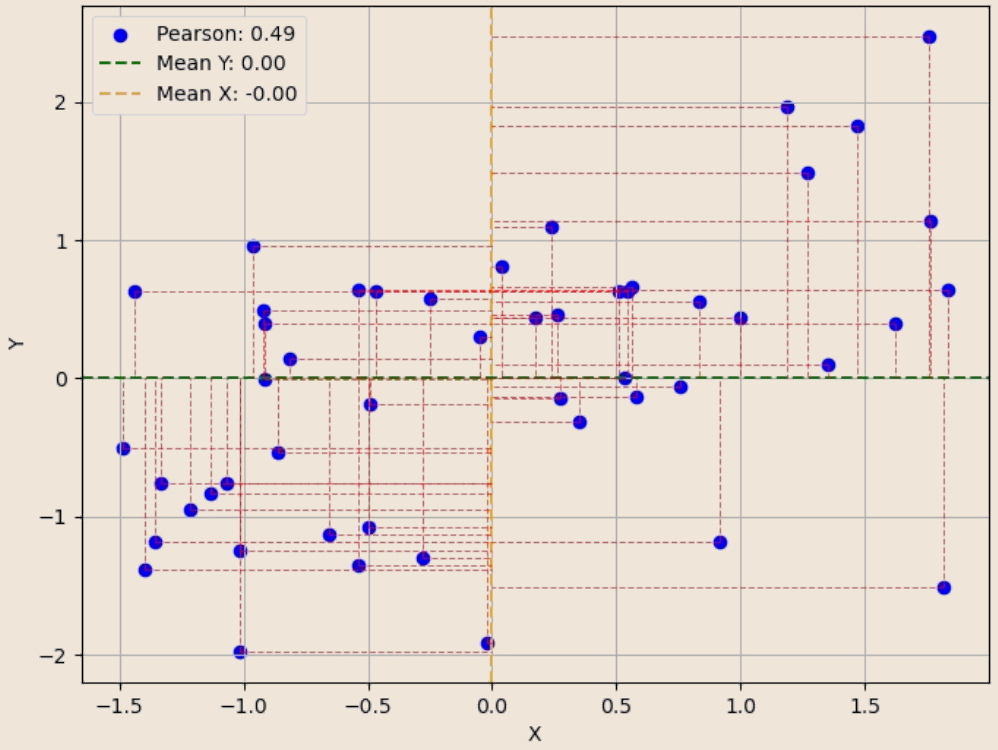
$$\Sigma_{ij} = Cov(X_i, X_j)$$

The covariance matrix can be computed in Python as follows:

```
data.drop('group',axis=1).cov() #drop the group column to compute covariance only on numerical columns
```

	relwt	glufast	glutest	instest	sspg
relwt	0.016702	-0.072815	0.982426	3.473373	5.266255
glufast	-0.072815	4087.097031	19546.064080	-3063.463649	4849.905651
glutest	0.982426	19546.064080	100457.849808	-12918.162739	25908.490182
instest	3.473373	-3063.463649	-12918.162739	14625.312548	101.482519
sspg	5.266255	4849.905651	25908.490182	101.482519	11242.331897

Pearson Correlation Coefficient



Normalised measure of linear association:

$$\rho(X,Y) = \frac{Cov(X,Y)}{s_X \cdot s_Y}$$

Equivalently, covariance of z-scored variables:

$$\rho(X,Y) = Cov(z(X), z(Y))$$

By normalizing, we remove the influence of individual variable scales or units of measurement.

Furthermore, the score is now in the $[-1, 1]$ range, where 1 denotes perfect positive correlation and -1 denotes perfect negative correlation. A score of 0 indicates decorrelation.

Interpreting Pearson's ρ

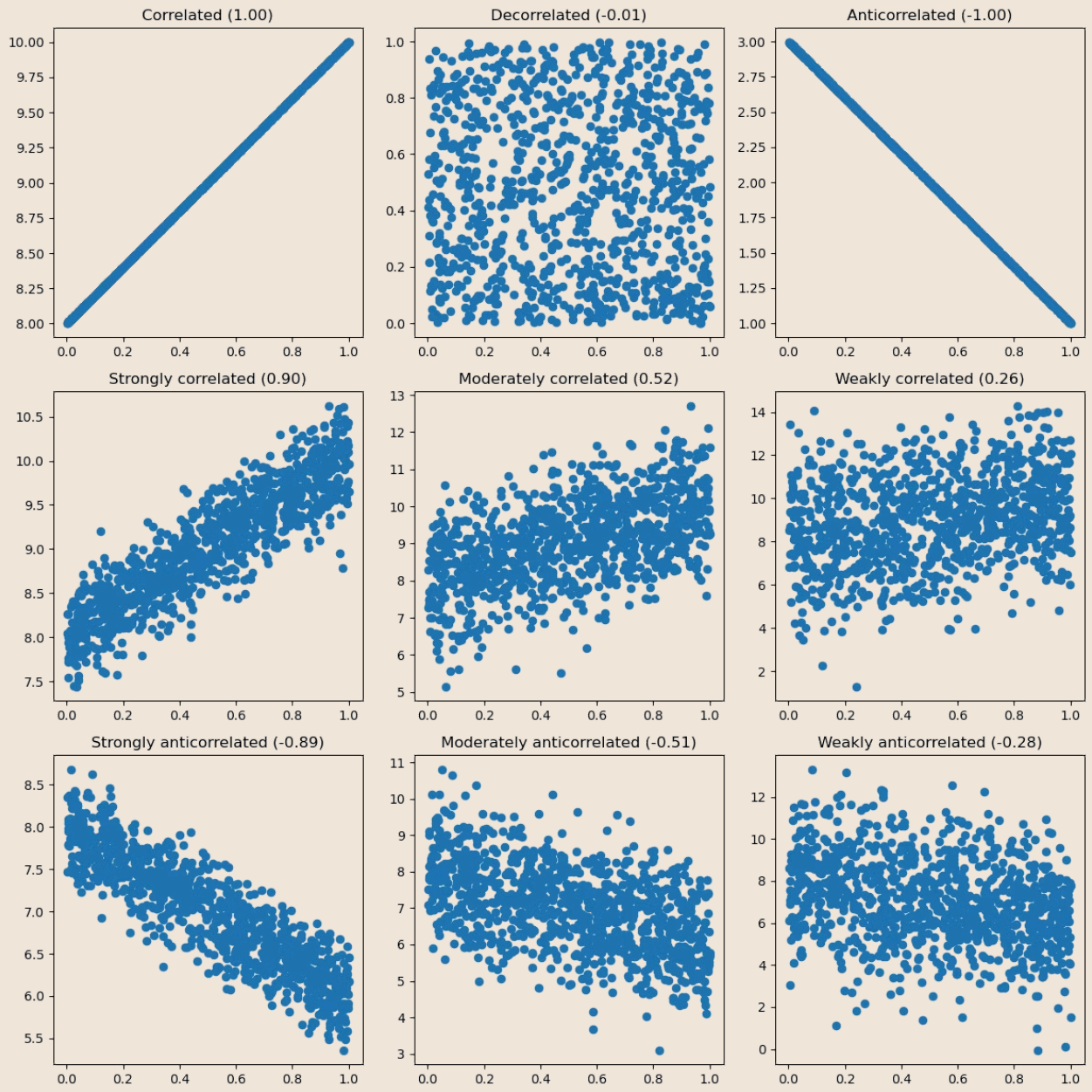
Sign (Direction)

- **Positive:** Variables increase together
- **Negative:** Variables move inversely

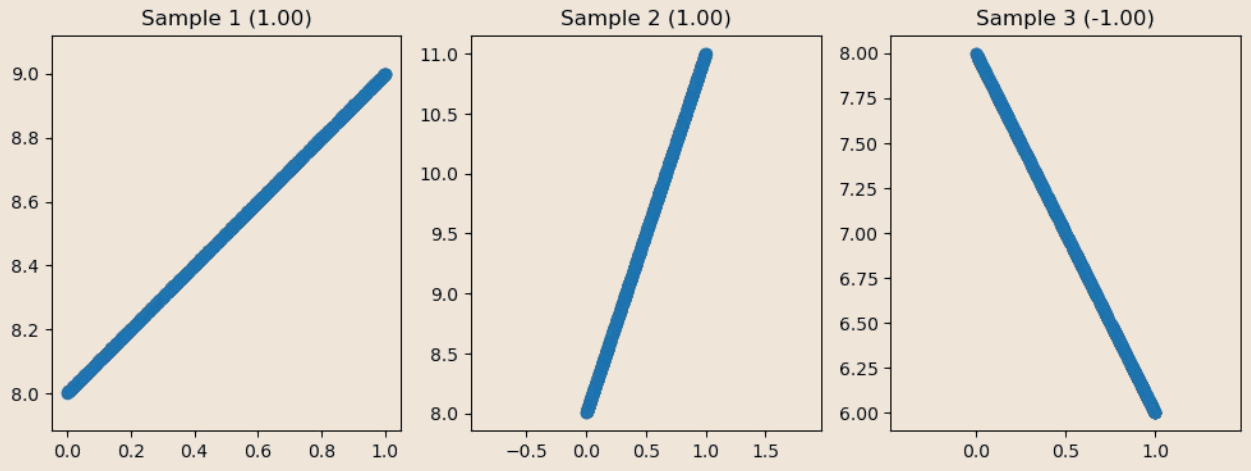
Magnitude (Strength)

- **0.0 – 0.3:** Weak
- **0.3 – 0.7:** Moderate
- **0.7 – 1.0:** Strong

Limitation: Pearson's coefficient only captures **linear** relationships. Non-linear correlations may exist even with low ρ values.



Note that there is no relationship between Pearson’s correlation coefficient and the slope of the line.



When to Use Pearson Correlation

Continuous Variables Only

Pearson's ρ measures linear relationships between two **continuous variables** (e.g., age, income, test scores).

NOT for Nominal/Ordinal

Do not use Pearson's ρ with categorical variables (e.g., colors, satisfaction levels) as it assumes interval data.

Exception: Point-Biserial

When one variable is continuous and the other is dichotomous (binary, 0/1), a Point-Biserial correlation is used, which is mathematically equivalent to Pearson's ρ.

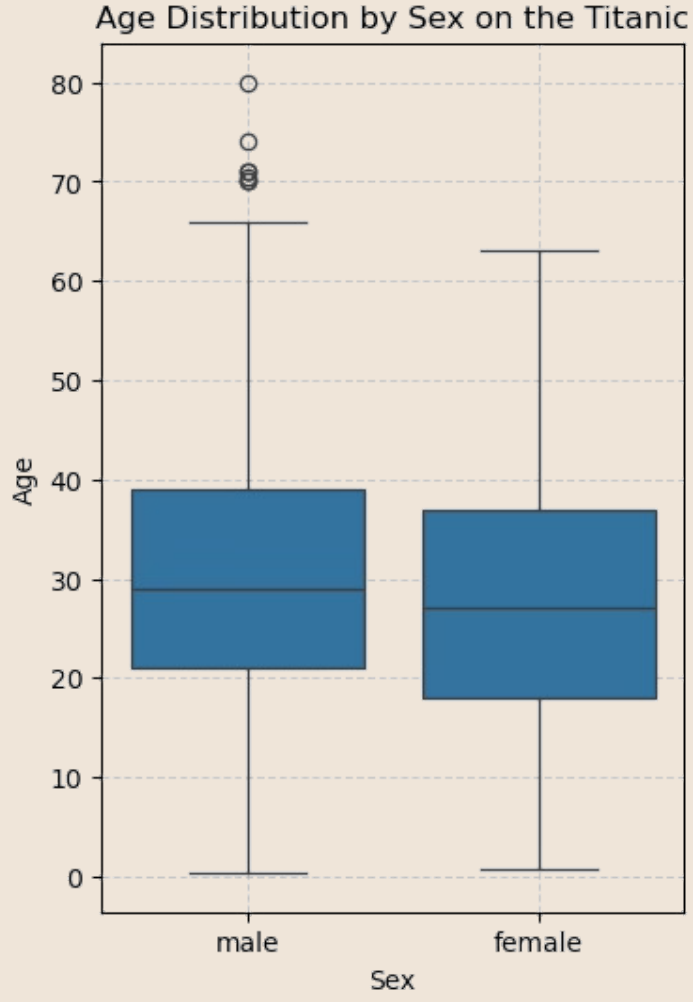
Example: Point-Biserial Correlation in Python

The example below demonstrates how to compute the point-biserial correlation coefficient between `Sex` and `Age` using the Titanic dataset:

```
titanic_cleaned = titanic.dropna(subset=['Age'])

# Extract the two variables of interest
age = titanic_cleaned['Age']
survived = titanic_cleaned['Sex'].replace({'male':'1','female':'0'}).astype(int)

pearson_result = pearsonr(age, survived)
print(f"Pearson Correlation (on the same data): {pearson_result[0]:.4f}")
```

```
Point-Biserial Correlation: 0.0933
Pearson Correlation (on the same data): 0.0933
```

> Males are slightly older than women



Spearman's Rank Correlation Coefficient

Based on ranks rather than raw values:

$$R = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}$$

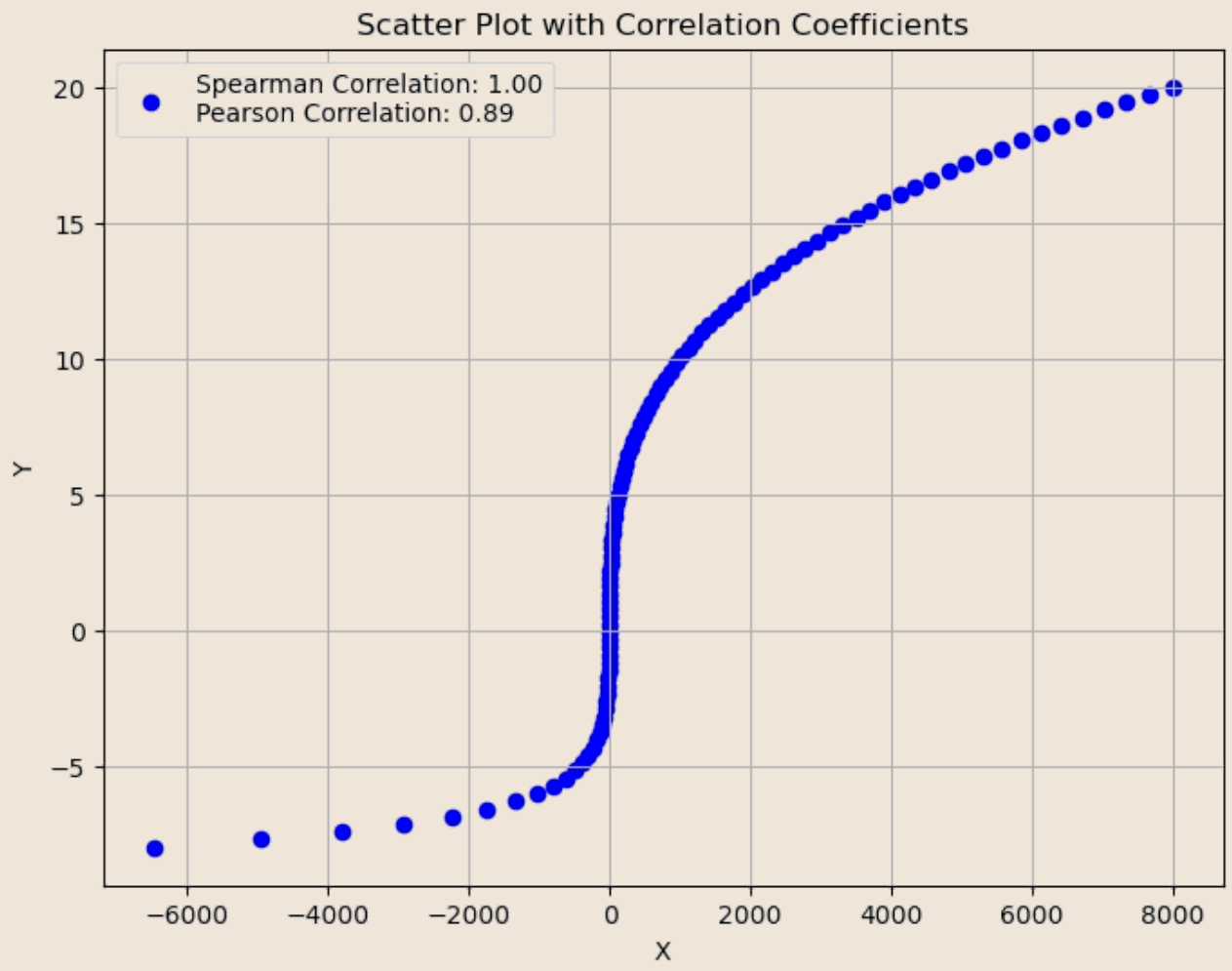
where d_i is the difference in ranks.

Captures non-linear monotonic relationships

Spearman's rank correlation coefficient captures non-linear but monotonic relationships, which are typical of individual judgments.

	Participant	Chocolate	Vanilla	Strawberry	MintChip
0	Alice	5	4	3	2
1	Bob	3	5	4	2
2	Charlie	4	3	5	2
3	David	2	3	2	5

Spearman Correlation (Chocolate vs. Vanilla): 0.21
Spearman Correlation (Chocolate vs. Strawberry): 0.40
Spearman Correlation (Chocolate vs. MintChip): -0.77



Kendall's Rank Correlation Coefficient

Kendall's τ measures the strength of association between two variables based on the number of concordant and discordant pairs.

$$\tau = \frac{\text{concordant pairs} - \text{discordant pairs}}{\text{total pairs}}$$

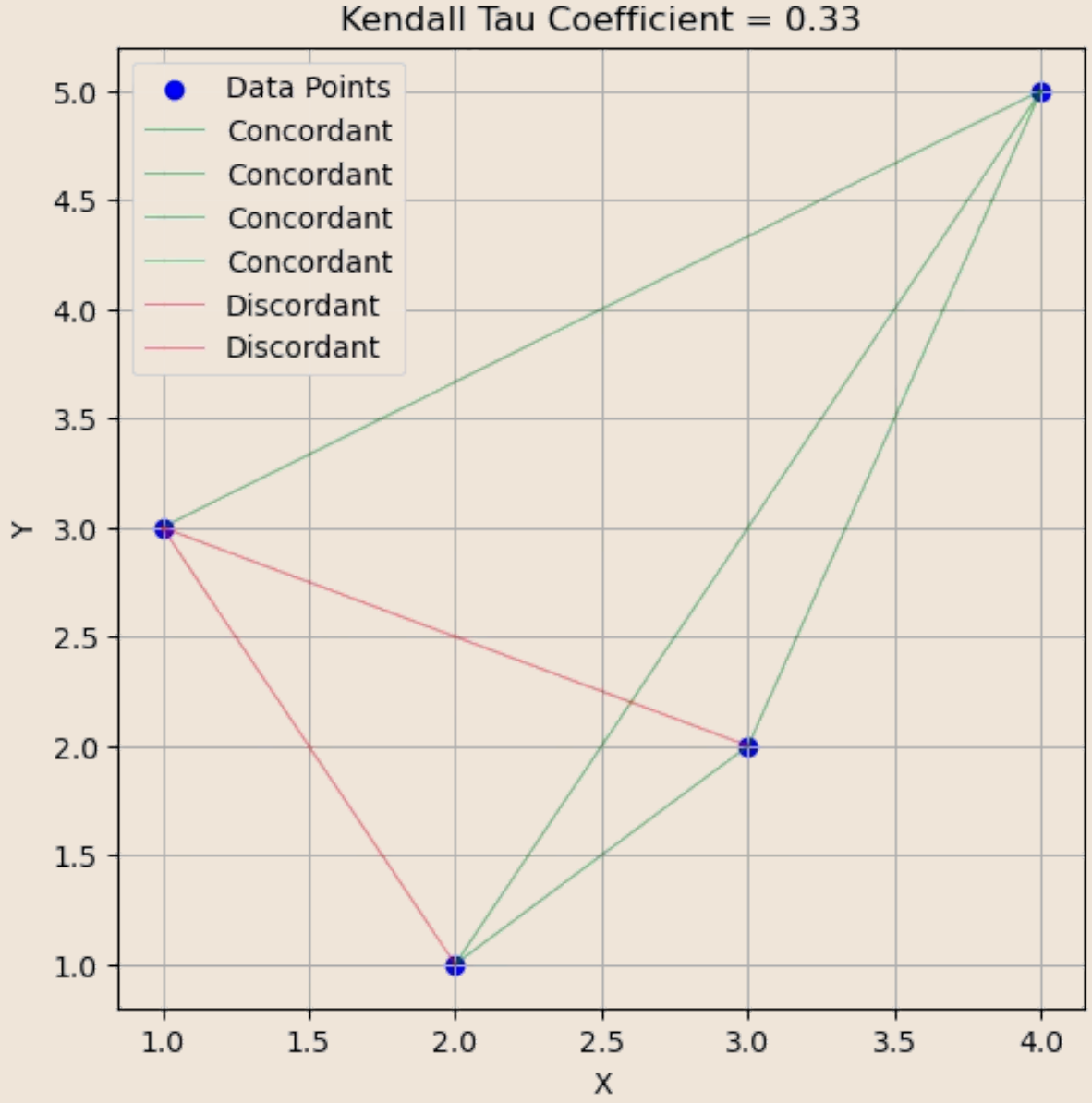
It is generally more robust to smaller sample sizes and ties in data than Spearman's coefficient.

	Participant	Chocolate	Vanilla	Strawberry	MintChip
0	Alice	5	4	3	2
1	Bob	3	5	4	2
2	Charlie	4	3	5	2
3	David	2	3	2	5

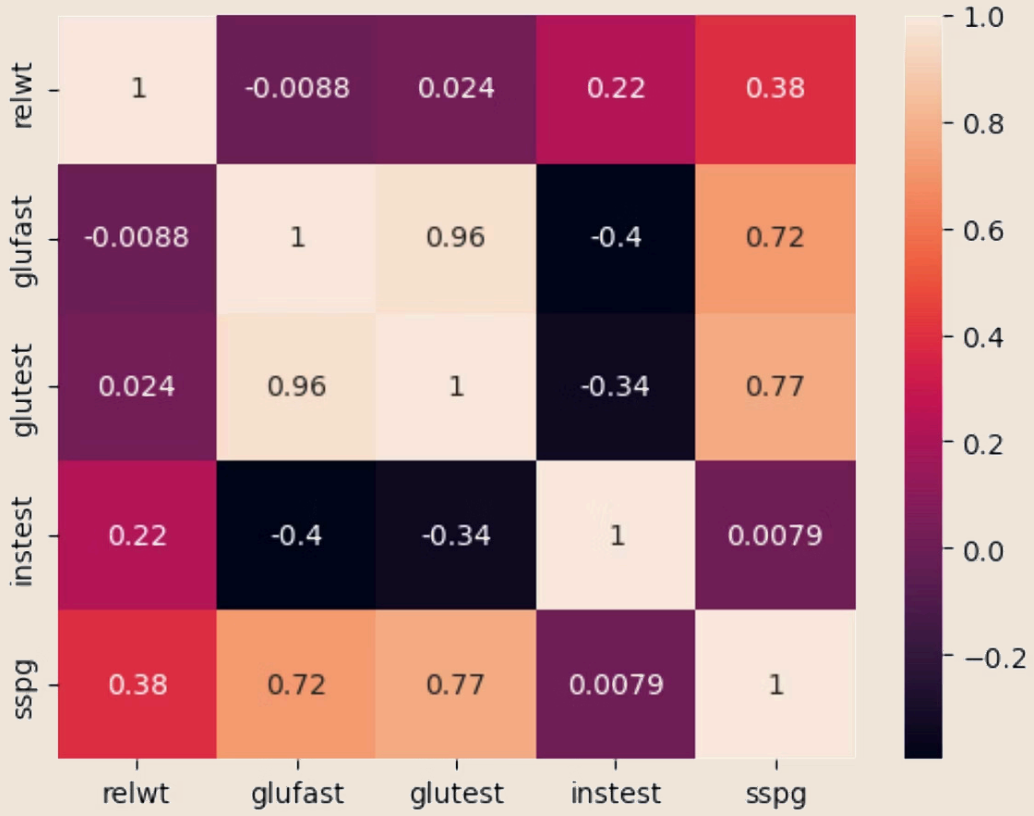
Kendall Correlation (Chocolate vs. Vanilla): 0.18
Kendall Correlation (Chocolate vs. Strawberry): 0.33
Kendall Correlation (Chocolate vs. MintChip): -0.71

The graph shows a sample with six points. Each line shows a possible pair of points. The green lines show the concordant pairs, while the red ones show the discordant pairs. Kendall's coefficient for this example is given by:

$$\tau = \frac{4-2}{6} = \frac{1}{3}$$



```
sns.heatmap(data.drop('group',axis=1).corr(), annot=True)
plt.show()
```



Correlation Matrix & Heatmap

Pairwise correlations can be easily computed in Python.

Compute with `.corr()` method:

- Default: Pearson
- method='spearman'
- method='kendall'

Conclusions and Next Steps

We Have Explored:

- The concept of association between two variables;
- Association between discrete variables;
- Association between continuous variables;
- Scatterplots and scattermatrices;

In the next lectures, we will look at data distributions.

References

- Chapter 3 of: Heumann, Christian, and Michael Schomaker Shalabh. Introduction to statistics and data analysis. Springer International Publishing Switzerland, 2016.
- https://en.wikipedia.org/wiki/Odds_ratio

Relative Risk (Optional)

Common in epidemiology for measuring correlation between exposure and outcome.

	Exposed	Non Exposed
Diseased	20	6
Healthy	380	594

Example: Disease & Exposure

Risk if exposed:

$$P(Diseased|Exposed) = \frac{\#(Diseased, Exposed)}{\#Exposed} = \frac{20}{380+20} = 0.05$$

Risk if not exposed:

$$P(Diseased|Non\,Exposed) = \frac{\#(Diseased, Non\,Exposed)}{\#Non\,Exposed} = \frac{6}{594} = 0.01$$

Relative Risk

$$RR = \frac{P(Diseased|Exposed)}{P(Diseased|Non\,Exposed)} = \frac{0.05}{0.01} = 5$$

- **RR = 1:** No association
- **RR > 1:** Positive association
- **RR < 1:** Negative association

Odds Ratio (Optional)

Useful when full population data is unavailable (limited sampling).

	Exposed	Non Exposed
Diseased	20	6
Healthy	10	16

Relative Risk

$$RR = \frac{P(Diseased|Exposed)}{P(Diseased|Non\ Exposed)} = \frac{\frac{\#\ (Diseased, Exposed)}{\#\ Exposed}}{\frac{\#\ (Diseased, Non\ Exposed)}{\#\ Non\ Exposed}} = \frac{20/(20+10)}{6/(6+16)} = 2.45$$

Odds Ratio

$$OR = \frac{\frac{P(Diseased|Exposed)}{P(Healthy|Exposed)}}{\frac{P(Diseased|Non\ Exposed)}{P(Healthy|Non\ Exposed)}} = \frac{2}{0.375} \approx 5.3$$

$$odd = \frac{P(E)}{1 - P(E)}$$

Under the **rare-disease assumption**, odds ratio approximates relative risk. Absolute counts cancel out in the formula.