🔬 Chapter 1: Exploratory Data Analysis (EDA)

This chapter focuses on the **first and foundational step of any data science project: exploring the data**.

1. The Role of EDA

- **Classic Statistics vs. EDA:**
  - **Classic Statistics** generally focuses on **inference** (drawing conclusions about a large population based on a small sample).
  - **Exploratory Data Analysis (EDA)**, as pioneered by **John Tukey**, focuses on generating hypotheses, uncovering patterns, and understanding the data structure *before* formal modeling.
- **Core Tools of EDA:**
  - **Simple Plots:** Boxplots, scatterplots, histograms.
  - **Summary Statistics:** Mean, median, quantiles, mode (estimates of location and variability).
- **Goal:** To help **"paint a picture"** of the dataset.

📊 1.1 Elements of Structured Data

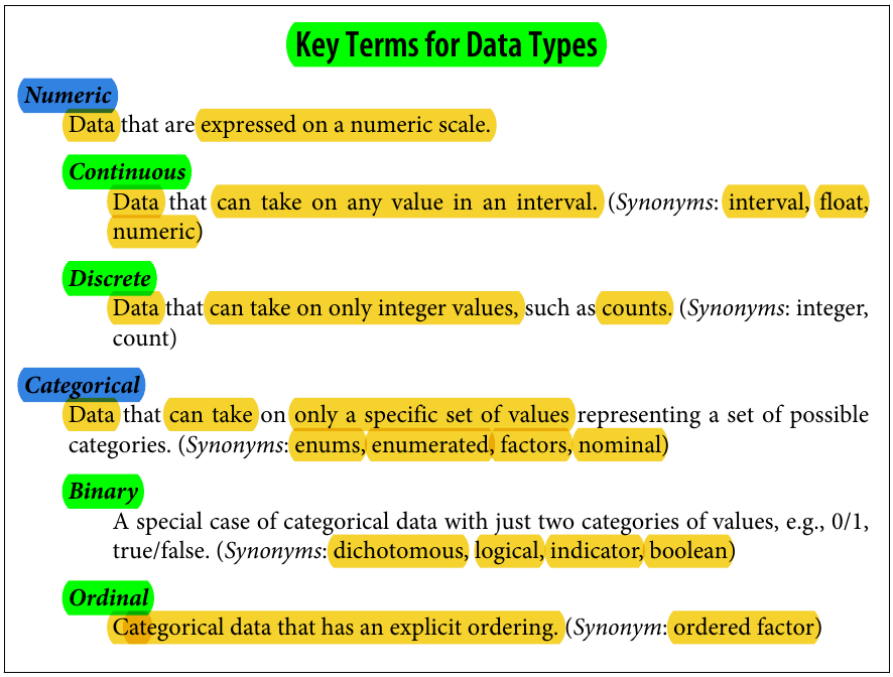
The major challenge in data science is to transform raw, unstructured data into **actionable information**. To apply statistical concepts and modeling, raw data must be processed into a **structured form**.

From Raw to Structured

Source of Data	Structure
Raw Data: Sensor measurements, events, text, images, videos.	**Unstructured**
Processed Data: A table/spreadsheet (rows and columns).	**Structured**

Basic Types of Structured Data

Data types are crucial because they determine the **type of visualization, statistical analysis, and predictive model** that can be applied.



Nominal Data (No Order): The categories are merely labels; there is **no intrinsic order or ranking** among them.

- *Example:* Type of TV Screen (Plasma, LCD, LED), Marital Status.

Ordinal Data (Has Order): The categories have a **meaningful sequence or rank**.

- *Example:* Customer Rating (Poor, Fair, Good, Excellent), Education Level (High School, Bachelor's, Master's).

NB: **Software Classification:** Ordinal data is often represented as a `factor` in R and Python, which **preserves the order** for use in charts, tables, and models.

🗃️ 1.2 & 1.3 Rectangular Data and Indexes

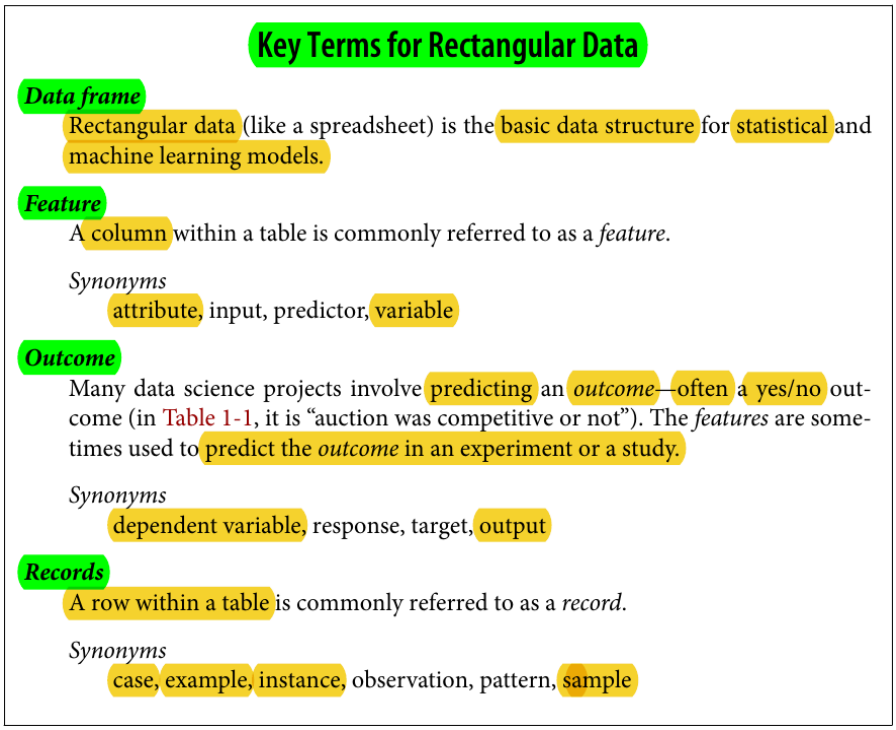
Rectangular Data

- **Definition:** The **spreadsheet or table** format, also known as a **2D matrix**.
- **Structure:**
  - **Rows:** Represent a **case** or observation (e.g., a customer, a transaction, a patient).
  - **Columns:** Represent **variables** or features (e.g., age, purchase price, diagnosis).

- **Data Science Format:**
  - In R and Python, this structure is commonly called a **Data Frame** (or `data.frame` in R, `DataFrame` in Python/Pandas).
- **Relational Databases:** Data extracted from relational databases must often be transformed and combined into a **single, rectangular table** for effective data analysis and modeling.

Indexes (Accessing Rows)

- **Purpose:** Indexes are used to **efficiently facilitate reaching different rows** of data.
- **Python (Pandas):**
  - The `DataFrame` object automatically includes an **integer index** referring to the order of the rows.
  - Supports **user-specified** or **multilevel (hierarchical) indexation** to improve data access and processing efficiency.
- **R:**
  - The native `data.frame` does not support user-specified or multilevel indexes.
  - Newer packages like **`data.table`** and **`dplyr`** provide support for multilevel and user-specified indexing.



Key Terms for Rectangular Data

Data frame

Rectangular data (like a spreadsheet) is the basic data structure for statistical and machine learning models.

Feature

A column within a table is commonly referred to as a *feature*.

Synonyms

attribute, input, predictor, variable

Outcome

Many data science projects involve predicting an *outcome*—often a yes/no outcome (in Table 1-1, it is "auction was competitive or not"). The *features* are sometimes used to predict the *outcome* in an experiment or a study.

Synonyms

dependent variable, response, target, output

Records

A row within a table is commonly referred to as a *record*.

Synonyms

case, example, instance, observation, pattern, sample

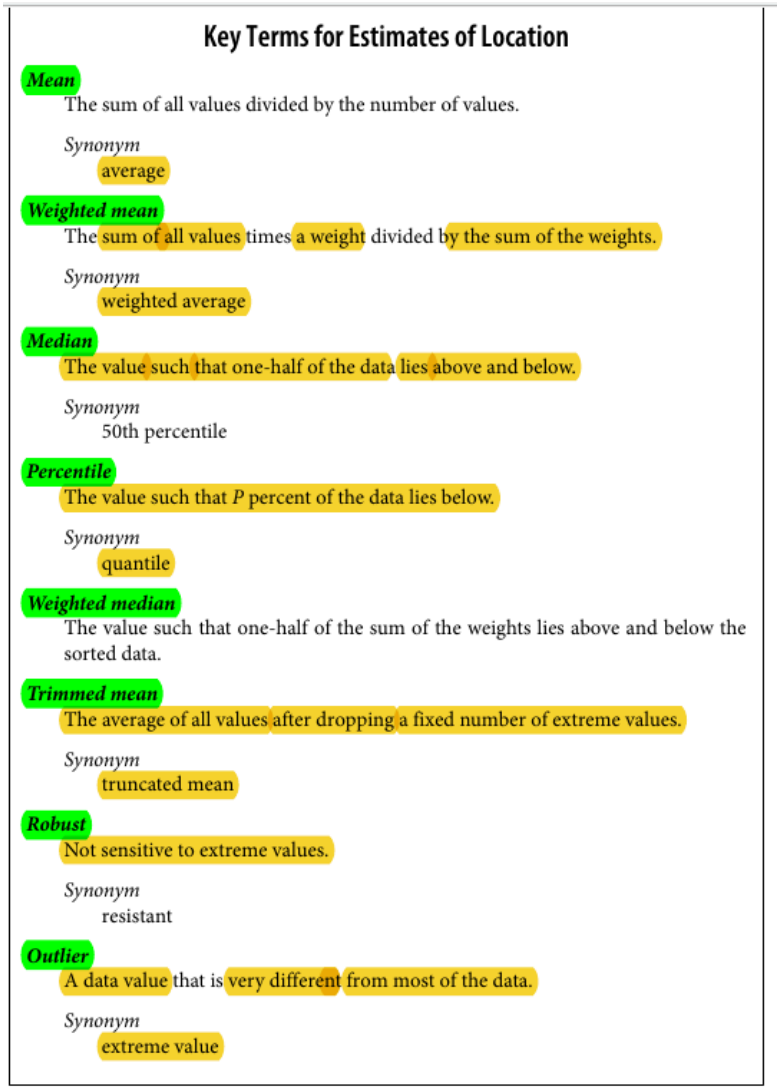
🌐 1.4 Non-Rectangular Data

While rectangular data is the most common form for basic EDA, data comes in other forms:

- **Time Series Data**
- **Spatial Data Structures** (e.g., geographical coordinates)
- **Graph/Network Data** (e.g., social connections)

📍 1.5 Estimates of Location (Central Tendency)

When a dataset contains a large number of distinct values, a basic step in Exploratory Data Analysis (EDA) is to estimate where the majority of the data is located. This is known as finding the **typical value** or **central tendency**.



Key Terms for Estimates of Location

Mean
The sum of all values divided by the number of values.

Synonym
average

Weighted mean
The sum of all values times a weight divided by the sum of the weights.

Synonym
weighted average

Median
The value such that one-half of the data lies above and below.

Synonym
50th percentile

Percentile
The value such that *P* percent of the data lies below.

Synonym
quantile

Weighted median
The value such that one-half of the sum of the weights lies above and below the sorted data.

Trimmed mean
The average of all values after dropping a fixed number of extreme values.

Synonym
truncated mean

Robust
Not sensitive to extreme values.

Synonym
resistant

Outlier
A data value that is very different from most of the data.

Synonym
extreme value

💡 Statistician Estimates vs. Data Scientist Metrics

While the terms often refer to the same calculated values (like the mean or median), the difference lies in the mindset:

- **Statistician Mindset (Estimates):** Focus is on how the value **estimates** a theoretical **true state of affairs** or population parameter.
- **Data Scientist/Business Analyst Mindset (Metrics):** Focus is on **measuring** the actual data on hand.

1.5.1 The Mean

The mean is the most common estimate of location. It is calculated by summing all data points and dividing by the number of data points (n).

$$\text{Mean} = \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

1.5.2 A Trimmed Mean

The mean is **not always the best measure** for central value, especially when the data contains **extreme values** (outliers). A trimmed mean offers a more robust alternative.

- **Definition:** A type of average that ignores a specified percentage (P) of the extreme values (both the smallest and largest) before computing the mean.
- **Procedure:** The values must be **sorted first**. The percentage P dictates how many numbers are removed from each end of the sorted data.
- **Benefit:** Provides an estimate that is **less sensitive to outliers** than the standard mean.

$$\text{Trimmed mean} = \bar{x} = \frac{\sum_{i=p+1}^{n-p} x_{(i)}}{n-2p}$$

1.5.3 Weighted Mean

A weighted mean is used when not all data points (xi) have the same **importance or weight** (wi).

$$\text{Trimmed mean} = \bar{x} = \frac{\sum_{i=p+1}^{n-p} x_{(i)}}{n-2p}$$

- **Calculation:** Each data point is multiplied by its weight, and the sum of these products is divided by the sum of the weights.

- **Use Cases for Weighted Mean:**
    1. **Data are More Valuable:** Some observations inherently carry more importance.
    2. **Unequal Representation (Sampling Bias):** The data collected does not equally represent the different groups present in the overall population.

🎯 Example of Use Case 2 (Sampling Bias)

In your example, the collected sample is biased, over-representing the older groups and under-representing the younger group, compared to the true population proportions.

Age Group	**Sample Size (Data Collected)**	**Average Time Spent (xi)**
18–25	200 users	4 min
26–40	600 users	5 min
40+	50 users	7 min
Total	**850 users**	

Age Group	True % of Total Users
18–25	**50%**
26–40	**30%**
40+	**20%**

The true **Population Distribution** should be used to determine the weights (wi):

The Problem: The 18–25 age group makes up only **24%** (200/850) of the *sample*, but represents **50%** of the *true population*.

- **Solution:** A weighted mean uses the **true population percentage** as the weight for each group's average time spent to correct this sampling distortion.

This section completes your notes on **Estimates of Location** by focusing on the **Median** and how to calculate different measures of central tendency in common data science tools.

I have organized and clarified your notes, providing the definition, use case, and a clear explanation of the **Weighted Median** example.

📍 1.5.4 The Median and Robust Estimates

The Median

- **Definition:** The value that **separates the upper and lower halves** of a sorted dataset.
    - If the dataset size is **odd**, the median is the single middle value.
    - If the dataset size is **even**, the median is the average of the two middle adjacent values.
- **Robustness to Outliers:** The median is **robust to outliers**. This is one of its major advantages over the mean.
    - *Example:* In a study of regional incomes, the presence of an extremely wealthy individual (like Bill Gates) will **drastically inflate the Mean**, but will have only a minimal effect on the **Median**, giving a more accurate measure of the "typical" income.

> **Note:** Being an outlier does not make a data value automatically invalid. However, outliers are often the result of **data errors** (e.g., mixing units like kilometers vs. meters) or faulty sensor readings.

Weighted Median

The weighted median is used when some data points (or groups) are **more relevant or important** than others.

- **Calculation Goal:** Find the value x_k such that the sum of the weights for values less than or equal to x_k is at least half the total weight, and the sum of the weights for values greater than or equal to x_k is also at least half the total weight.

Income (xi)	Weight (wi)	Cumulative Weight
40k	100 people	100
45k	100 people	**200**
55k	100 people	300
1,000k (1M)	1 person	301

Total Weight: 301 people
Half the Total Weight: 301 / 2 = 150.5
The Weighted Median is 45k: The cumulative weight reaches and exceeds 150.5 at the second income level (45k).
Since 100 + 100 = 200, and 150.5 is included within this cumulative weight, **45k** is the weighted median.

Anomaly Detection

In contrast to typical data analysis, where outliers are sometimes informative and sometimes a nuisance, in *anomaly detection* the points of interest are the outliers, and the greater mass of data serves primarily to define the "normal" against which anomalies are measured.

Trimmed Mean as a Middle Ground

- The **trimmed mean** is a good solution against outliers and is often considered a compromise **between the Mean and the Median**.
- It is generally **less effective when the dataset is very small** because trimming even a small percentage can remove a significant portion of the data, potentially leading to instability in the estimate.

Practical Calculation: Population Rate and Murder Rate Example

When computing the average of a metric like the **Murder Rate**, it is essential to use a **weighted estimate** (weighted mean or weighted median) where the **Population** serves as the **weight**. This ensures that states with larger populations contribute proportionally more to the average rate.

#	State	Population	Murder Rate (per 100k)	Abbreviation
1	**Alabama**	4,779,736	5.7	**AL**
2	**Alaska**	710,231	5.6	**AK**
3	**Arizona**	6,392,017	4.7	**AZ**
4	**Arkansas**	2,915,918	5.6	**AR**
5	**California**	37,253,956	4.4	**CA**
6	**Colorado**	5,029,196	2.8	**CO**
7	**Connecticut**	3,574,097	2.4	**CT**
8	**Delaware**	897,934	5.8	**DE**

Compute the mean, trimmed mean and median witghted meand and medianusing R and python (look script state.py state.R)

In trimmed-mean, we did trim = 0.1 it means trip 10% from each end,

Next, if we want to compute the average murder rate, we need to use either the **weighted mean** or the **weighted median**. Base R doesn't include a function for the weighted median, so we need to install the **matrixStats** package. In Python, the weighted mean is available in **NumPy**, and for the weighted median we can use the **wquantiles** package.

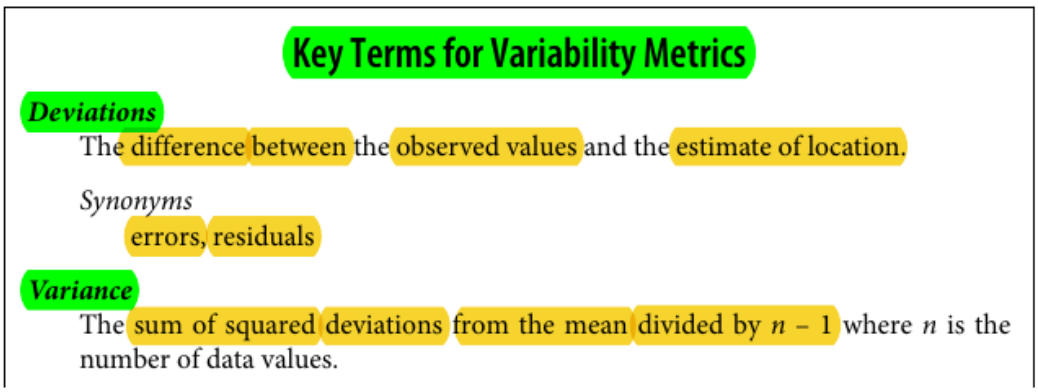
Metric	R Calculation Function/Method	Python Calculation Function/Method
Mean	mean(...)	… .mean() (Pandas Series method)
Trimmed Mean	mean(..., trim = 0.1)	scipy.stats.trim_mean(..., 0.1)
Median	median(...)	… .median() (Pandas Series method)
Weighted Mean	weighted.mean(...)	numpy.average(..., weights=...)
Weighted Median	weightedMedian(...) (requires matrixStats library)	wquantiles.median() (requires wquantiles library)

📐 1.6 Estimates of Variability (Dispersion)

While **Estimates of Location** (like the mean or median) describe the **typical value** (the 1D center) of a feature, **Estimates of Variability** (also called **Dispersion**) describe how **spread out** the data is (the 2D measure).

- **Definition:** Variability measures whether data values are **tightly clustered** or widely **spread out** by calculating the distance of the values from a central location.
- **Central Role in Statistics:** Variability is at the heart of statistical analysis. Statisticians and data scientists must:
    - **Measure** it precisely.
    - **Reduce** it where possible (e.g., through better sampling).
    - **Distinguish** real variability from noise or error.
    - **Identify** various sources of variability.
    - **Make decisions** and draw conclusions in the presence of it.

.



📐 1.6.1 Standard Deviation, Variance, and MAD

Deviations and the Need for Squares/Absolute Values : a deviation is the difference between an individual value and the mean of the dataset.

The Problem:
The sum of all deviations from the mean is always zero.
For example, in the set {5, 4, 2} with a mean of 3, the deviations are {+2, +1, -1}, and 2 + 1 + (-1) = 2.
(Correction: that specific example does not sum to zero, but for a set like {5, 4, 3, 2, 1}, the deviations are {+2, +1, 0, -1, -2}, which do sum to 0.)

The Fix:
To prevent positive and negative deviations from canceling each other out, we must remove the signs. This is done in two main ways, leading to two common measures of variability:

1. **Squaring each deviation** (used for variance and standard deviation).
2. **Taking the absolute value of each deviation** (used for the mean absolute deviation).

This is a comprehensive set of notes detailing the core **Estimates of Variability** used in Biostatistics and Exploratory Data Analysis.

I have organized and clarified your text, providing clear definitions, the mathematical logic, and proper interpretation for each estimate.

1.6.1.1 Mean Absolute Deviation (MAD)

The Mean Absolute Deviation (MAD) is calculated by averaging the absolute deviations from the mean.

$$\text{Mean absolute deviation} = \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n}$$

Example (5, 7, 9, mean = 7):
The absolute deviations are:
$|5 - 7| = 2$
$|7 - 7| = 0$
$|9 - 7| = 2$

Calculation:
MAD = (2 + 0 + 2) / 3 = 4 / 3 ≈ 1.33
(Correction: $|7 - 7|$ is 0, not 1, in your notes.)

Interpretation:
A MAD of approximately 1.33 means that, on average, each data point is 1.33 units away from the mean.

1.6.1.2 Variance and Standard Deviation

Variance and **Standard Deviation** (SD) are the most common measures of variability, despite not being robust to outliers.

- **Variance:** The average of the **squared** deviations from the mean.
- **Standard Deviation (SD):** The square root of the Variance.

$$\text{Variance} = s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}$$
$$\text{Standard deviation} = s = \sqrt{\text{Variance}}$$

Why Standard Deviation instead of Variance?
Standard deviation is preferred because it is expressed in the same units as the original data.
Variance, on the other hand, is in squared units, which makes it less intuitive to interpret.

Degrees of Freedom (n − 1)
When calculating the **sample variance** as an estimate of the **population variance**, we divide by **n − 1** instead of n.

This happens because computing the sample mean uses one piece of information from the data. As a result, only **n − 1 values are free to vary**.
Dividing by n − 1 corrects for this loss of freedom and gives an **unbiased estimate** of the population variance.

1.6.1.3 Robust Estimate: Median Absolute Deviation (MAD)

Since Variance and Standard Deviation are heavily influenced by outliers (because deviations are squared), a robust measure is often preferred.

Median Absolute Deviation (from the Median):
This is the median of the absolute differences between each data point and the median (m) of the entire dataset.

Robust MAD = Median($|x_1 - m|$, $|x_2 - m|$, …, $|x_{\square} - m|$)

Scaling for Normality:
The unscaled MAD is naturally smaller than the Standard Deviation.
To make it comparable to the SD for data that follow a normal distribution, it is multiplied by a constant: **1.4826**.

Interpretation:
For a normal distribution, the interval
[m − MAD, m + MAD]
is expected to contain approximately **50% of the data points**.

We can also calculate strimmed standard deviation, analogous to trimmed mean.

📈 1.6.1.4 **Estimates Based on Percentiles**

Percentile-based estimates offer a different approach to quantifying **dispersion** by looking at the spread of the **sorted data**.

1/ **Range and Percentiles**

Range: The simplest measure: **Maximum Value − Minimum Value**.
Pro: Easy to calculate.
Con: Extremely sensitive to **outliers** and generally not useful as a sole measure of **dispersion**. **Max** and **Min** are best used to identify potential **outliers**.

Percentile: The value below which a given percentage of **observations** falls (e.g., the 80th **percentile** is the value that is greater than 80% of the **data**).

Quantile: A related term; the 0.8 **quantile** is the 80th **percentile**.

2/ The **Interquartile Range (IQR)**

The **IQR** is a highly robust measure of **dispersion**.

Definition: The difference between the 75th **percentile** (**Third Quartile, Q3**) and the 25th **percentile** (**First Quartile, Q1**).

IQR = Q3 − Q1

Interpretation: The **IQR** describes the spread of the middle 50% of the **data**.

Example (IQR = 6): "The middle 50% of the **scores** are spread out over 6 points."

Example (Population IQR = 2,958,479.25): "Half of the **states** in the sample have a **population** spread out over a range of 2.96 million."

Measure of Variability	R Function	Python (Pandas/Statsmodels)
Standard Deviation (SD)	sd(data)	data.std() (Pandas Series Method)
Interquartile Range (IQR)	IQR(data)	data.quantile(0.75) - data.quantile(0.25) (Pandas Method)
Median Absolute Deviation (MAD)	mad(data)	robust.scale.mad(data) (from statsmodels.robust.scale)

1.7 Exploring the Data distribution

Key Terms for Exploring the Distribution

Boxplot
A plot introduced by Tukey as a quick way to visualize the distribution of data.

Synonym
box and whiskers plot

Frequency table
A tally of the count of numeric data values that fall into a set of intervals (bins).

Histogram
A plot of the frequency table with the bins on the x-axis and the count (or proportion) on the y-axis. While visually similar, bar charts should not be confused with histograms. See "Exploring Binary and Categorical Data" on page 27 for a discussion of the difference.

Density plot
A smoothed version of the histogram, often based on a *kernel density estimate*.

📈 1.7.1 **Percentiles, Quartiles, and Boxplots**

Percentiles and **quartiles** provide a robust way to understand **data distribution** and **variability** by dividing the **sorted data** into equal segments.

Percentiles and Quantiles
Percentile: A value below which a specified percentage of **data** falls (e.g., the 95th **percentile**).

Quantile: A general term for a value that divides the **data** into equal probability segments.

Quartiles divide the **data** into four segments (25th, 50th, 75th **percentiles**).

Deciles divide the **data** into ten segments (10th, 20th, 30th **percentiles**, etc.).

💻 **Calculation Example (Murder Rate)**

Using R or Python, you can calculate specific quantiles for the 'Murder Rate' feature:

Quantile (p)	Percentile	Interpretation	Murder Rate Value (per 100,000)
0.25	**25th (Q1)**	25% of states have a rate below this value.	4
0.5	**50th (Median/Q2)**	The middle value; 50% of states are above/below.	**5.15**
0.75	**75th (Q3)**	75% of states have a rate below this value.	5.625

The median is 5.15 murder per 100.000 person

This is a great set of notes detailing **Percentiles, Quartiles, and the Boxplot**, which are crucial tools for visualizing location and variability in Biostatistics and EDA.

I will organize your text into a clear, structured summary, defining the key concepts and explaining the interpretation of the boxplot elements.

📈 1.7.1 Percentiles, Quartiles, and Boxplots

Percentiles and quartiles provide a robust way to understand data distribution and variability by dividing the sorted data into equal segments.

Percentiles and Quantiles

- **Percentile:** A value below which a specified percentage of data falls (e.g., the 95th percentile).
- **Quantile:** A general term for a value that divides the data into equal probability segments.
    - **Quartiles** divide the data into four segments (25th, 50th, 75th percentiles).
    - **Deciles** divide the data into ten segments (10th, 20th, 30th percentiles, etc.).

💻 Calculation Example (Murder Rate)

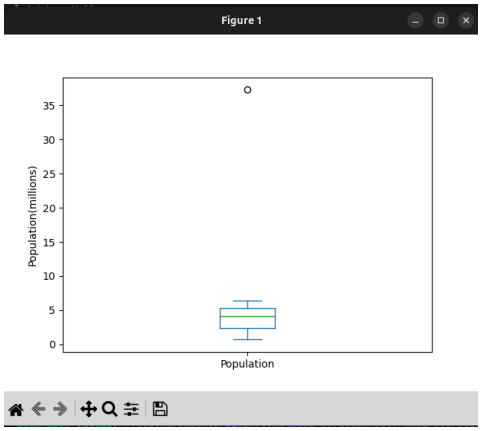
Using R or Python, you can calculate specific quantiles for the 'Murder Rate' feature:

R : quantile(state[['Murder.Rate']], p=c(.05, .25, .5, .75, .95))
Python : state['Murder.Rate'].quantile([0.05, 0.25, 0.5, 0.75, 0.95])

Quantile (Percentile)	5% (0.05)	25% (Q1)	50% (Median/Q2)	75% (Q3)	95% (0.95)
Murder Rate Value (per 100,000)	2.54	4	5.15	5.625	5.765

The Boxplot (Box and Whisker Plot)

The **boxplot** is a standard EDA **visualization** that compactly displays the **five-number summary** of a dataset: **minimum, Q1, median, Q3, and maximum (or outliers)**.



Boxplot Components and Interpretation

- **Horizontal Line inside the Box (Median / Q2) :** The 50th **Percentile**.
- **Bottom of the Box (Q1) :** The 25th **Percentile (First Quartile)**.
- **Top of the Box (Q3) :** The 75th **Percentile (Third Quartile)**.
- **The Box Width (IQR):** The **Interquartile Range (Q3 − Q1)**.
  Interpretation: Measures the **variability** or **spread** of the most concentrated portion of the data (the **middle 50%**).

- **Whiskers (Dashed Lines):** extend from the **box** to the furthest **data points** that are not classified as **outliers** (typically up to 1.5 × IQR away from the box).
    - **Interpretation:** They indicate the overall **range** of the bulk of the **data (along with the box in normal distribution)**.

- **Individual Points (Dots/Circles):** any **data points** that fall outside the range covered by the **whiskers** (beyond 1.5 × IQR).
    - **Interpretation:** Plotted as single points, these represent **outliers**—values that are statistically unusual compared to the rest of the **distribution**.

📏 **Boxplot Rule for Whiskers**

- Standard statistical software (like R's **boxplot** and Python's **matplotlib**) typically extends the **whiskers** to a maximum distance of 1.5 times the **IQR** away from the edges of the **box (Q1 and Q3)**.

- Any **data points** falling beyond this 1.5 × IQR limit are considered **outliers** and are plotted individually

🎯 **Interpretation Example (Population Boxplot)**

- **Median (Horizontal Line):** 5 million.

- **The Box:** Half of the **states' populations** are situated between approximately 2 million (Q1) and 7 million (Q3).

- **Outliers:** There are one or more higher **population states** plotted as single points outside the **whiskers**, indicating they are unusually large compared to the rest of the **sample**.

📈 1.7.2 Frequency Tables and Histograms

A **Frequency Table** and its graphical representation, the **Histogram**, are fundamental tools for visualizing the entire distribution of a numerical variable.

📊 **Constructing a Frequency Table**

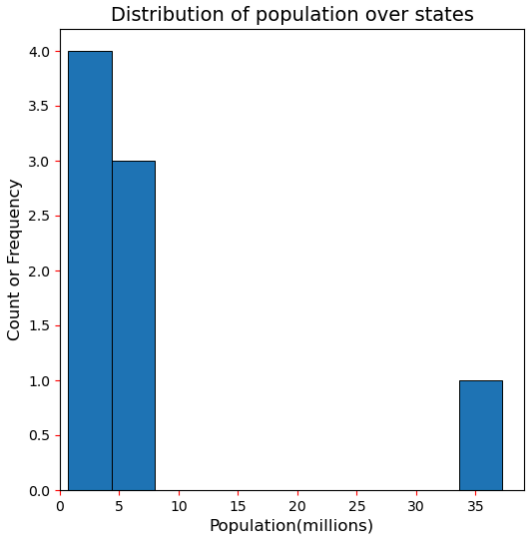
A frequency table divides the data range into equal-sized segments called **bins** and counts how many values fall within each bin.

Task	R Function	Python (Pandas/NumPy) Equivalent
Read Data	read.csv()	pd.read_csv()
Calculate Breaks	seq(..., length=11)	np.linspace(..., num=11)
Bin Data	cut(..., breaks=breaks, right=TRUE, include.lowest=TRUE)	pd.cut(..., bins=breaks, right=True, include_lowest=True)
Generate Frequencies	table()	df.value_counts()
Sort by Bin Range	*(Implicit/Manual)*	.sort_index()

Drawing the Histogram

The **Histogram** plots the bins (x-axis) against the frequency (y-axis). It provides a visual shape of the distribution, revealing patterns like skewness and multimodality.

Task	R Function	Python (Pandas/Matplotlib)
Generate Histogram	hist(...)	data['col'].plot.hist(...)
Set X-Axis Label	xlab="..." (Argument within hist())	ax.set_xlabel(...)
Set Y-Axis Label	ylab="..." (Argument within hist())	ax.set_ylabel(...)
Set Title	main="..." (Argument within hist())	ax.set_title(...)
Set Axis Limits	xlim or ylim (Argument within hist())	ax.set_xlim(...)
Set Ticks/Colors	col="...", axes=TRUE, etc.	ax.tick_params(...)
Display Plot	*(Often automatic)*	plt.show()



📈 The Four Moments of a Distribution

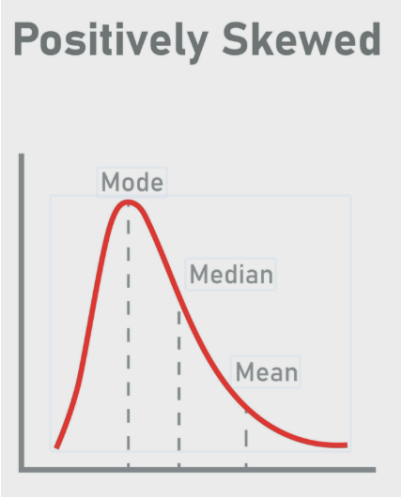
The "moments" are quantitative metrics used to numerically describe a probability distribution or a dataset, complementing the visual information provided by a histogram.

Moment Number	Statistical Name	Concept Described	Key Metrics Used
First	**Location**	Where the data is centered.	Mean, Median, Mode
Second	**Variability**	How spread out the data is.	Standard Deviation, Variance, IQR
Third	**Skewness**	The asymmetry of the data's shape.	Skewness Coefficient
Fourth	**Kurtosis**	The weight of extreme values (tails).	Kurtosis Coefficient

1. Skewness (The Third Moment)

Skewness refers to the asymmetry of the distribution's shape. It indicates the direction in which the "tail" of the data is pulling.

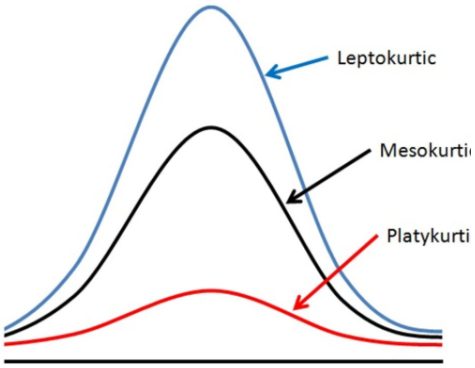
- **Skewed to Larger Values (Positive Skew / Right Skew):**
  - The long tail extends to the **right** (larger values).
  - The **Mean > Median**.
  - *Example:* Income or population data, where a few high-value outliers pull the mean up.
- **Skewed to Smaller Values (Negative Skew / Left Skew):**
  - The long tail extends to the **left** (smaller values).
  - The **Mean < Median**.



2. Kurtosis (The Fourth Moment)

Kurtosis indicates the distribution's propensity to produce **extreme values (outliers)** relative to a normal distribution.

- **High Kurtosis (Leptokurtic):**
    - Suggests a distribution with **heavy tails** and often a **sharp central peak**.
    - This implies that extreme outliers are **more frequent** than in a normal distribution.
- **Low Kurtosis (Platykurtic):**
    - Suggests a distribution with **lighter tails** and a **broader central peak**.



📈 1.8 **Density Plots and Estimates**

A **Density Plot** (or **Density Estimate**) is a method of visualizing the **distribution** of **data values** as a continuous, smooth line.
It can be thought of as a **smoothed histogram**, offering a cleaner representation of the **distribution's shape** without being affected by the choice of **bin size**.

Kernel Density Estimation (KDE)

The smooth curve of a **density plot** is computed using **Kernel Density Estimation (KDE)**.

- **Density Estimation:** The process of estimating the continuous, underlying shape of the **data's distribution**.
- **Kernel:** A mathematical function that is applied to the **dataset**. It measures the distance from a specific point **x** to all other **data points**.

Calculation Steps (Under the Hood):

1. The **kernel** measures the distance from a **target point x** (on the x-axis) to each **data point** in the set.
2. It assigns **weights** based on these distances (**closer points get higher weights**).

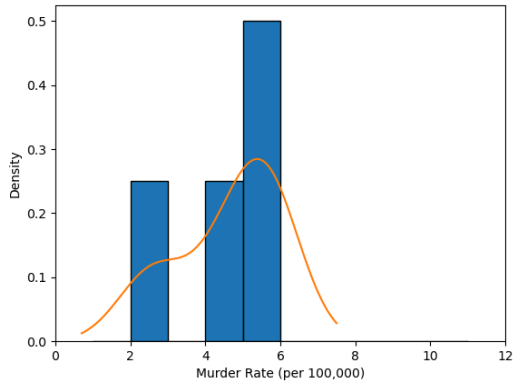
3. It adds all the **weighted contributions** to get the estimated **density value f(x)** for that specific point.
4. This process is repeated for many **x-values** to generate the smooth **curve**.

`freq = FALSE`

This is important:

- `freq = TRUE` → histogram shows **counts** (frequency).
- `freq = FALSE` → histogram shows **density** (area under bars = 1).

Language/Library	Function/Method / Setting	Purpose	Explanation / Context
R (Base)	density()	Calculates the Kernel Density Estimate (KDE) data.	Provides the data structure for the smooth curve.
R (Base)	lines()	Plots (draws) the calculated curve onto the existing plot.	Superimposes the density curve onto the histogram.
R (Base)	freq = FALSE	**Histogram Setting**	Sets the Y-axis to density scale (instead of counts). This is required so the total area of the histogram is 1 and the scale matches the density curve for correct overlaying.
Python (Pandas/Matplotlib)	.plot.density() (Series method)	Calculates & Plots the Kernel Density Estimate (KDE) curve.	Performs both calculation and plotting in one step.
Python (Pandas/Matplotlib)	density=True (in .plot.hist())	**Histogram Setting**	Sets the Y-axis to density scale (equivalent to freq=FALSE in R) to allow proper overlaying of the density curve.



🎯 Interpretation: Area Under the Curve

The most important statistical interpretation of a density plot is that the **Total Area Under the Curve is exactly equal to 1**.

Area Represents Proportion/Probability

The area under the curve between any two points (a and b) on the x-axis directly corresponds to the **proportion** of the entire dataset that lies between those two points.

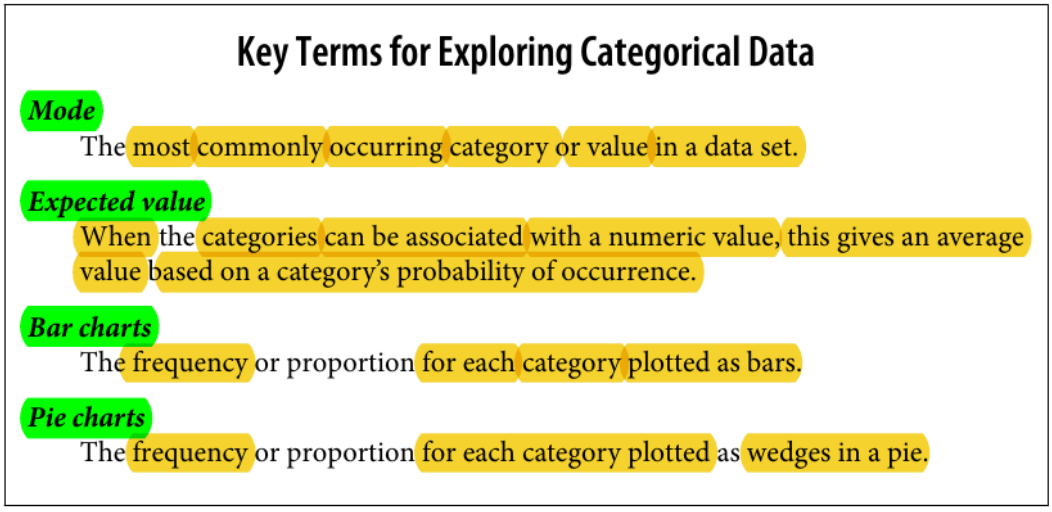
- **Proportion as Area:** Since the total area under the curve is 1 (or 100%), any sub-area is a fraction of 1, which represents a proportion.
- **Probability:** The area under the curve between a and b gives you the **probability** that a randomly chosen data point (X) will fall within the range $a < X < b$.

Example (Murder Rate)

If the area under the curve between a Murder Rate of 4.0 and 8.0 is calculated to be **0.45**:

- **Proportion: 45%** of the states in the dataset have a murder rate between 4.0 and 8.0.
- **Probability:** The probability that a randomly chosen state has a murder rate between 4.0 and 8.0 is **0.45**.

🟦 1.9 Exploring Binary and Categorical Data



Categorical Data represents qualitative variables (labels or groups). Unlike numerical data, categorical data is generally easier to interpret directly because the values already represent distinct entities.

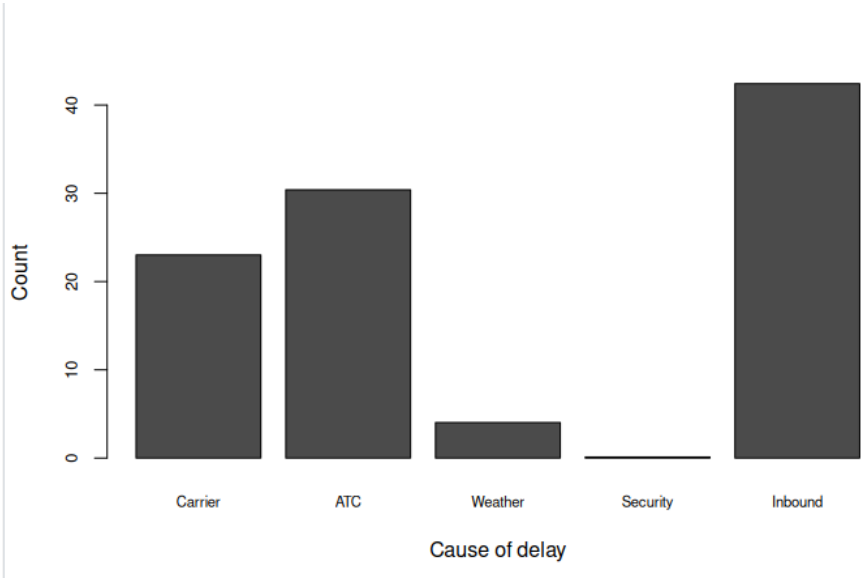
1.9.1 Bar Charts

A **Bar Chart** is the standard visualization tool for categorical data. It displays the frequency (or proportion) of each category using rectangular bars whose height is proportional to the count.

📊 Bar Charts vs. Histograms

It is crucial to distinguish between a bar chart and a histogram, even though they look similar:

Language/Library	Purpose	Function / Method	Context
R (Base)	**Main Plotting Function (draws the bar chart).**	barplot()	Takes the data matrix and plots bar heights proportional to the values.
R (Base)	Prepares data by converting the data frame into a numeric structure suitable for barplot().	as.matrix()	Converts the delay data frame into a matrix format.
Python (Pandas/Matplotlib)	**Plotting Method (draws the bar chart).**	.plot.bar()	A method chained from the pandas object (Series or DataFrame) that generates a bar plot using Matplotlib.
Python (Pandas)	Prepares data by swapping rows and columns, which is often necessary when plotting category data from a single-row DataFrame.	.transpose()	Flips the axes of the DataFrame before calling the plotting method.
Python (Matplotlib)	**Display Function (renders the plot on the screen).**	plt.show()	Opens the Matplotlib figure containing the bar chart.



Bar charts vs histogram

- Non linked bars vs linked bars
- In x axis : different categorical variables vs 1 variables at different range
- With gaps / no gaps(only absence of occurrence)

💡 **Converting Numerical to Categorical Data**

- When you create a **histogram**, the process of dividing continuous numerical data into fixed **bins** essentially converts the numerical data into ordered **categorical data**.
- This conversion is a common and beneficial practice in data analysis as it helps **reduce complexity and size**, which aids in discovering relationships between features, especially early in the analysis process.

1.9.2 Mode

The **Mode** is the simplest estimate of location for categorical data.

- **Definition:** The value or values that appear most frequently in the dataset.
- **Usage:** The mode is primarily used for **categorical data**. It is generally not useful for numeric data, particularly continuous numeric data, where a repeated value is rare.
- *Example:* If analyzing religious preference in a region, the category with the highest count is the mode.

💰 1.9.3 Expected Value (EV)

The **Expected Value (EV)** is a summary statistic that represents the **average outcome** you can anticipate from an event if it were repeated many times.

- **Calculation:** The Expected Value is a form of **weighted mean**, where the **weights are the probabilities** of each outcome occurring.

$$EV = \sum(\text{Value}_i \times \text{Probability}_i)$$

🎯 Example: Marketing Webinar Revenue

A marketer offers two subscription levels ($300/$50) after a webinar, with specific sign-up probabilities:

EV = 0 . 05 * 300 + 0 . 15 * 50 + 0 . 80 * 0 = 22 . 5 dollars

The calculation uses a **weighted mean**, where:

1. **The "Values"** are the potential financial outcomes (the subscription revenues and the non-sign-up revenue).
2. **The "Weights"** are the probabilities of each outcome occurring (the sign-up percentages). probability weights, often based on subjective judgment.

the expected value of a webinar attendee is thus $22.50 per month,

Interpretation and Use

The result, **$22.50 per month**, doesn't mean every single attendee will pay exactly that amount (in fact, no one pays $22.50).

Instead, it means that if the company hosts **many webinars** with these same probabilities, the average revenue earned from each person who attends, when spread across all attendees (sign-ups and non-sign-ups), will be **$22.50 per month**.

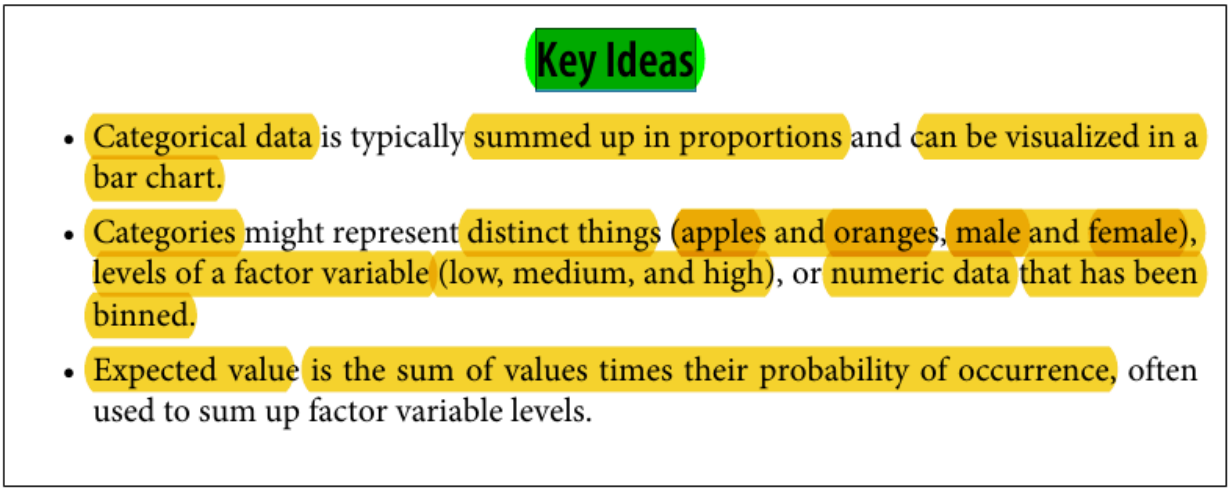
Examples : the expected value of five years of profits from a new acquisition, or the expected cost savings from new patient management software at a clinic.

⚖️ 1.9.4 Operational Definition of Probability

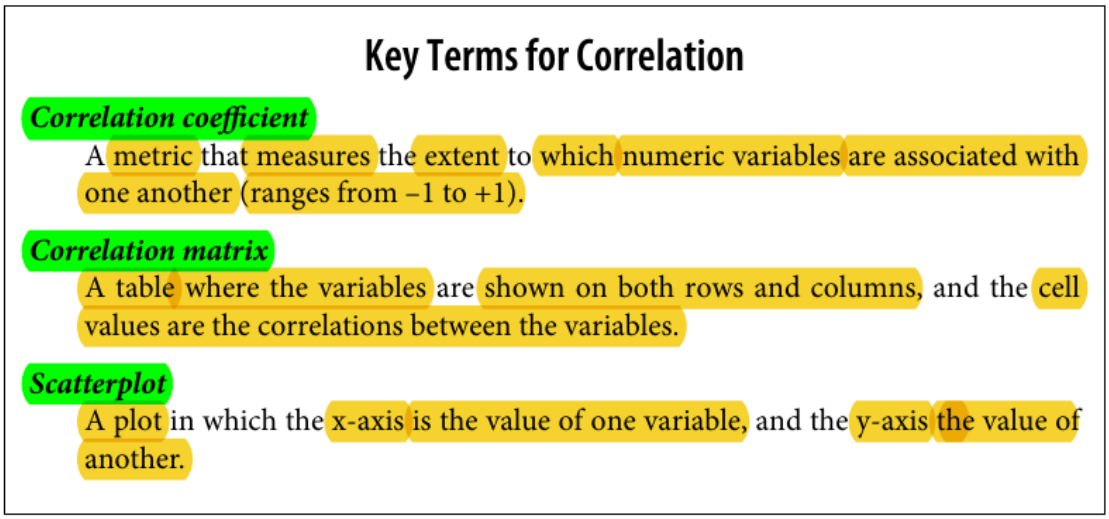
For the purpose of data science and statistical modeling, probability has an operational definition:

> "The probability that an event will happen is the **proportion of times it will occur** if the situation could be repeated over and over, **countless times**."

This frequentist view of probability, while often an imaginary construction (especially in biostatistics where trials cannot always be repeated), provides an adequate operational understanding for data analysis and modeling.



🔗 2.0.0 **Correlation and Measures of Association**



1/Predictors and Variables

In the context of building a **prediction model**, **variables** take on specific roles:

- **Predictor (Feature):** A variable X used in a model to predict or influence the **Response Variable Y**.
- **Response (Target/Dependent Variable):** The variable Y that the model aims to predict.

Exploratory Data Analysis (EDA) includes examining the **correlation** among **Predictors** (to avoid redundancy) and between **Predictors** and the **Target Variable** (to assess relevance).

- **Positively Correlated:** When X goes high, Y goes high (or X goes low, Y goes low).
- **Negatively Correlated:** When X goes high, Y goes low.

2/The Correlation Coefficient (Pearson's r)

While the simple **vector sum of products** (as shown in your example, $1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 32$) is a metric for **association**, it depends heavily on the **scale** and **magnitude** of the numbers.

A more useful and **standardized metric** is the **Correlation Coefficient r**, which gives an estimate of the **strength** and **direction** of the **linear correlation** between two **variables X and Y**.

Formula for the sample Pearson's correlation coefficient:

r = Σ (xi − x̄)(yi − ȳ) / √[Σ(xi − x̄)² * Σ(yi − ȳ)²]

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- **Degrees of Freedom:** Note that the formula implicitly uses **n − 1 (degrees of freedom)** because it standardizes the terms using **sample standard deviations**, which divide by n − 1.

Interpretation of r

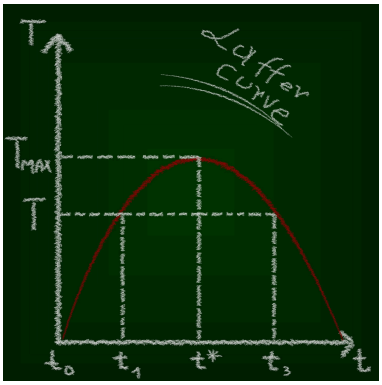
The **correlation coefficient** always falls between **−1 and +1**:

- **+1:** Perfect positive linear correlation.
- **−1:** Perfect negative linear correlation.
- **0:** No linear correlation.

📉 Tax Rates and Revenue: A Non-Linear Association

The relationship between tax rates and the total amount of tax revenue collected by the government is not a simple straight line (i.e., it is **non-linear**).

1. The Relationship Curve



Shutterstock

Imagine the relationship plotted on a graph:

- **X-axis:** Tax Rate (from 0% to 100%)
- **Y-axis:** Tax Revenue Collected

The curve typically looks like an **inverted U-shape** (a parabolic or bell-shaped curve).

2. The Two Phases of the Curve

Phase A: Low to Moderate Tax Rates

- **Action:** As tax rates increase from **0%** (where revenue is obviously zero) up to a moderate level (the peak of the curve, T^*), people are motivated to work, invest, and report income.
- **Result: Tax Revenue Increases.** The government collects more revenue with higher rates in this range. The correlation here is **positive** (as one goes up, so does the other).

Phase B: High Tax Rates

- Once tax rates pass the peak and become very high, approaching **100%**, two main things happen:
    - **Reduced Incentive:** The incentive to work, produce, and invest declines sharply, as a large portion of the reward is taken away.

    - **Tax Avoidance/Evasion:** People and businesses increase efforts to legally avoid taxes (e.g., through loopholes) or illegally evade them, and some economic activity may move to the underground economy.
- **Result: Tax Revenue Declines.** Despite the rate being higher, the *base* of taxable income shrinks dramatically. The correlation here is **negative** (as the rate goes up, the revenue goes down).

3. Why Correlation Coefficient Fails

The **correlation coefficient r** measures the **strength** and **direction** of a **linear relationship**.

- A value of **r = +1** means a **perfect positive straight line**.
- A value of **r = -1** means a **perfect negative straight line**.

Because the **tax rate/revenue relationship** is a **curve** (positive then negative), the overall **correlation coefficient** might be calculated as close to **zero (0)**, which would **misleadingly suggest no relationship at all**.

- **Misleading Conclusion:** A correlation of **r ≈ 0** would suggest that **changing tax rates has no effect on revenue**.
- **Reality:** Changing the **tax rate** has a very **significant effect**, but the **direction** of that effect depends entirely on **where you are on the curve**. The **non-linear relationship** is strong, but a **linear metric** like the **correlation coefficient** fails to capture it.

🔗 2.1 Correlation Matrix

A **Correlation Matrix** is a square table that displays the **correlation coefficients (r)** between every possible pair of variables in a dataset. It is a fundamental tool in Exploratory Data Analysis (EDA) for quickly assessing relationships across many features simultaneously.

Structure and Interpretation

	T	CTL	FTR	VZ	LVLT
T	1	0.475	0.328	0.678	0.279
CTL	0.475	1	0.42	0.417	0.287
FTR	0.328	0.42	1	0.287	0.26
VZ	0.678	0.417	0.287	1	0.242
LVLT	0.279	0.287	0.26	0.242	1

The Diagonal of 1s: The main diagonal of the matrix always contains the value **1**. This represents the correlation of a variable with itself (e.g., $r(\text{T, T}) = 1$), which is a perfect positive correlation.

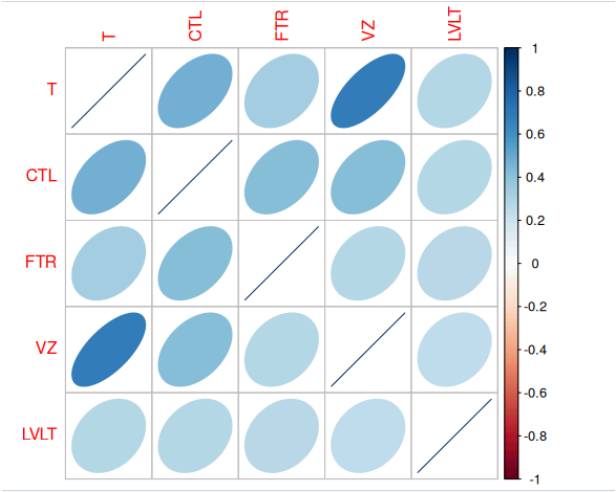
- **Symmetry:** The matrix is symmetrical, meaning the correlation between A and B is the same as the correlation between B and A (e.g., $r(\text{T, CTL}) = r(\text{CTL, T}) = 0.475$). You only need to look at the lower or upper triangle.
- **Identifying Strong Relationships:** You look for values closest to $+1$ or -1.
    - **Observation:** In the example, **VZ and T** have the highest correlation coefficient of **0.678**, indicating a strong positive relationship between their daily stock returns.

Visualizing the Matrix

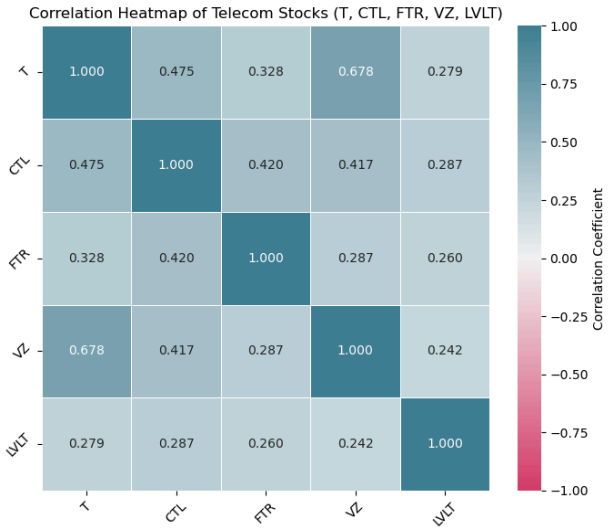
The table of correlation is commonly **plotted** (often as a heatmap) to visually display the relationships between variables. In a heatmap:

- Strong positive correlations (near +1) are typically shown in **dark colors** (e.g., dark blue/red).
- Strong negative correlations (near -1) are shown in **contrasting dark colors**.
- Near-zero correlations are shown in **light or neutral colors**.

R(correlation plot using ellipses)



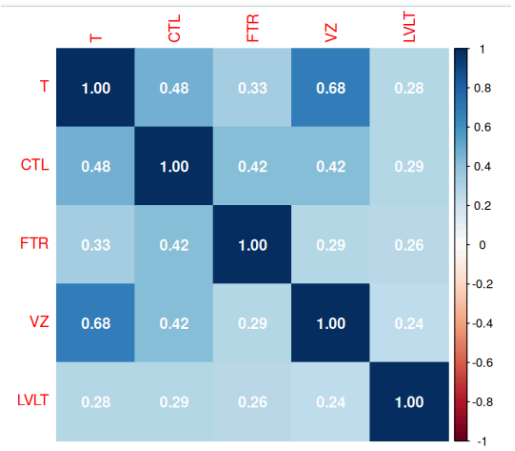
python (heatmap)



R Function/Argument	Python Function/Method	Purpose in Visualization
read.csv("file", row.names = 1)	pd.read_csv("file", index_col=0)	Data Loading: Reads the CSV file, correctly assigning the first column as row/index labels.
as.matrix()	(Implicit/Not needed)	Data Type: Converts the data frame to a matrix, which corrplot prefers. Python's sns.heatmap works directly with pandas DataFrames.
library(corrplot)	import seaborn as sns	Package Import: Loads the necessary visualization library.
(telecom_corr_matrix) (Argument)	sns.heatmap(telecom_corr_data, ...)	Core Plotting: The main function that generates the visualization. R uses ellipses/shapes; Python uses a color scale (heatmap).

(Implicit via plot device)	plt.figure(figsize=(7, 6))	Figure Sizing: Sets the size of the output image/figure.
(Internal color scheme)	sns.diverging_palette(...)	Color Map: Defines the color scheme to be used for the correlation visualization.
(None—often manual)	annot=True (Argument)	Annotation: Displays the numeric values on the plot (Python handles this easily in the call).
(Custom settings needed)	plt.title(...)	Labeling: Sets the title of the plot.
(Custom settings needed)	plt.xticks()/plt.yticks()	Axis Customization: Manually adjusts the rotation of axis labels.
(None—manual)	plt.show()	Rendering: Displays the final generated plot.

If you want a classic look for heatmap with values using R, just change :
corrplot(telecom_corr_matrix, method='color', addCoef.col="white")



Like the mean and standard deviation, the correlation coefficient is sensitive to outliers in the data.
in R and python there is packages offers robust alternative ti the classical correlation coefficient:

R : function covRob

Python : scikit-learn module *sklearn.covariance* implement a variety of approaches

In my R code and the Seaborn heatmap, the **correlation coefficient is usually the Pearson correlation coefficient** by default.

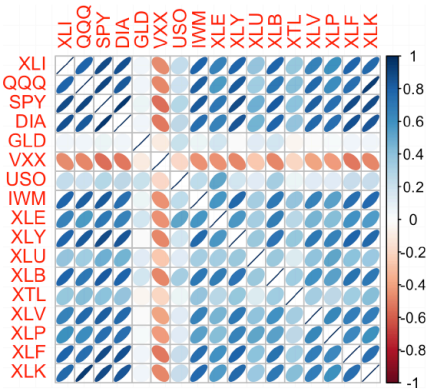
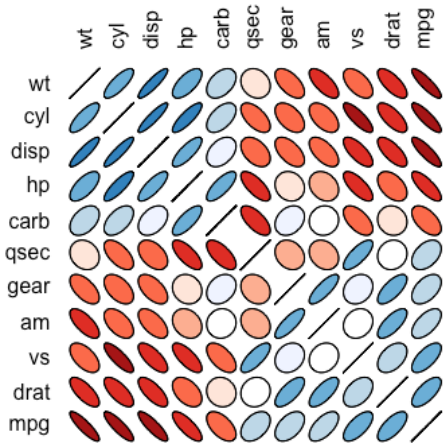


Figure 1-6. Correlation between ETF returns

- **Similar Assets Move Together:** Stocks or funds in the same sector (like two tech funds) or same market (like S&P 500 and Dow) usually have **high positive correlation** (they trend the same way).

- **Defensive Assets Move Differently:** Assets like gold or oil tend to have **weak or negative correlation** with the stock market, meaning they often move opposite or independently.

How to Read the Correlation Ellipse:



- **Positive Relationship:** The ellipse points **up and to the right.**
- **Negative Relationship:** The ellipse points **up and to the left.**
- **Strong Relationship:** The ellipse is **thin and dark** (points are tightly clustered).
- **Weak Relationship:** The ellipse is **wide and light** (points are dispersed).

Other Correlation Estimates:

Correlation coefficients based on the **rank of the data**, such as **Spearman's rho** or **Kendall's tau**, are **robust to outliers**. Rank-based estimates are mostly used for **smaller data sets** and **specific hypothesis tests**.

However, **data scientists** can generally stick to **Pearson's correlation coefficient** and its **robust alternatives** for **exploratory analysis**.

2.2 scatterplot

In 2.telecom.csv file, Each value in that table represents the **daily percentage change in the closing price** of the corresponding stock. These values are often referred to as "Daily Returns" in finance.

T telecom company:

Date	Open	High	Low	Close ⓘ	Adj Close ⓘ	Volume
Dec 2, 2025	25.83	25.87	25.45	25.50	25.50	21,285,030
Dec 1, 2025	25.95	25.97	25.55	25.79	25.79	34,681,900
Nov 28, 2025	25.82	26.02	25.76	26.02	26.02	16,345,800
Nov 26, 2025	25.83	26.07	25.80	25.82	25.82	27,713,000
Nov 25, 2025	25.76	26.21	25.75	25.86	25.86	44,336,200
Nov 24, 2025	25.99	26.02	25.43	25.62	25.62	66,605,900
Nov 21, 2025	25.52	26.18	25.50	25.93	25.93	59,017,100

Daily Return

The daily return (R) is calculated as the change in price from one day to the next, divided by the previous day's price:

$$R_{\text{day } t} = \frac{\text{Price}_t - \text{Price}_{t-1}}{\text{Price}_{t-1}}$$

- **Positive Value** (e.g., 0.0051): Indicates the stock price went **up** for that day.
- **Negative Value** (e.g., -0.0021): Indicates the stock price went **down** for that day.

R vs python scatterplot:

The scatter plot

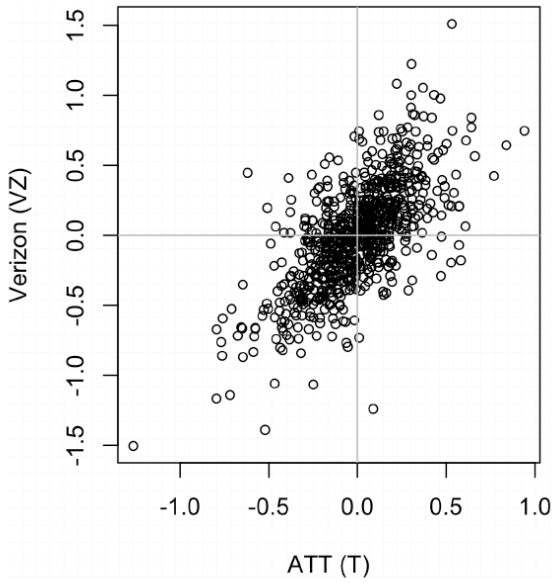


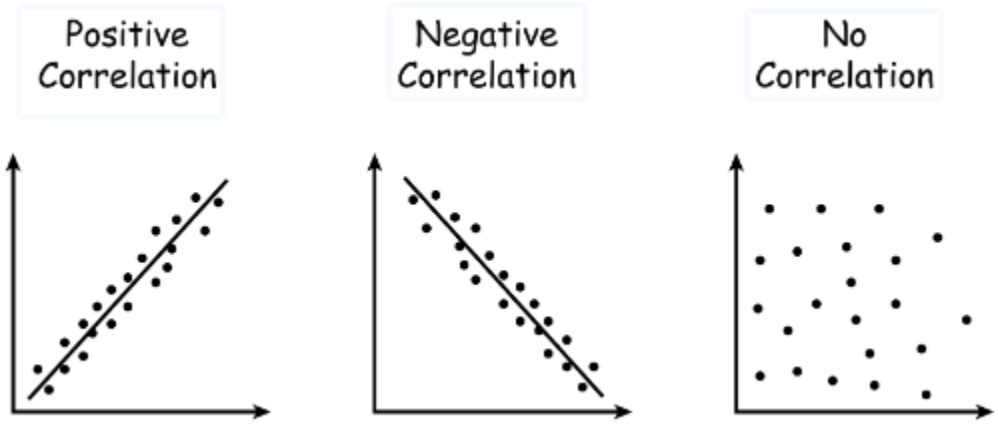
Figure 1-7. Scatterplot of correlation between returns for ATT and Verizon

A scatter plot is one of the most powerful tools in data analysis because it immediately tells a story about the relationship between two variables.

In the case of your stock return scatter plot (AT&T vs. Verizon)

1. Correlation (The Shape and Direction)

Correlation measures how closely the two variables move together. You read this by looking at the **overall shape** and **direction** of the cloud of points.



Visual Feature	Meaning for T and VZ Returns	Financial Implication
Direction (Slope)	If the points trend up and to the right (positive slope), the correlation is positive. If they trend down and to the left (negative slope), the correlation is negative.	Positive Correlation: When T's price goes up (positive return), VZ's price also tends to go up. They move in the same direction.
Tightness (Clustering)	If the points are tightly clustered along a line, the correlation is strong (close to +1 or -1). If they are widely dispersed (like in the file I created for you with a lower correlation), the correlation is weak (close to 0).	Strong Correlation: The stocks are highly influenced by the same market factors (e.g., sector news, interest rate changes). They offer little diversification benefit against each other.
Your Plot:	Your initial plot showed a strong positive correlation (tightly clustered, upward trend), and the second plot showed a weak positive correlation (more dispersed, slight upward trend).	

2. Magnitude (The Axes)

The axes represent the scale of the daily movements:

Axis Feature	**Meaning**	**Interpretation**
X-axis (T)	Shows the daily return magnitude for AT&T.	How much T moved on a given day. Points far from the central 0 line on the X-axis represent days with large T price changes.
Y-axis (VZ)	Shows the daily return magnitude for Verizon.	How much VZ moved on a given day. Points far from the central 0 line on the Y-axis represent days with large VZ price changes.
Center (0, 0)	The center of the plot where the axes cross.	Represents days where both stocks had virtually no price change.

3. Outliers (The Exceptions)

Outliers are points that lie far away from the main cluster or the regression line. They represent exceptional trading days.

- **Financial Outlier:** A point far from the regression line suggests a day where one stock moved significantly, but the other did *not* move as expected based on their historical relationship.
    - *Example:* If a point is far to the right (large positive T return) but near the center vertically (near zero VZ return), it means AT&T had a major positive news event that day, but Verizon did not, defying their usual strong correlation.

Summary of Information Read from the Plot:

The scatter plot allows you to quickly assess:

1. **If a relationship exists** (Is the cloud random, or does it have a shape?).
2. **The nature of the relationship** (Are they positive or negatively correlated?).
3. **The strength of the relationship** (How tightly are the points clustered?).
4. **The deviation** (Are there any days where one stock behaved completely differently than expected?).

"Books response + gemini" :

The returns have a positive relationship: while they cluster around zero, on most days, the stocks go up or go down in tandem (upper-right and lower-left quadrants).

There are fewer days where one stock goes down significantly while the other stock goes up, or vice versa (lower-right and upper-left quadrants).

While the plot Figure 1-7 displays only 754 data points, it's already obvious how difficult it is to identify details in the middle of the plot. We will see later how adding transparency to the points, or using hexagonal binning and density plots, can help to find additional structure in the data.

Interpreting the Scatter Plot by Quadrant

That quote perfectly describes how to read a scatter plot of correlated stock returns!

It formalizes the concepts we discussed (direction and magnitude) by referring to the four quadrants of the plot, which is the standard way to interpret financial correlation visually.

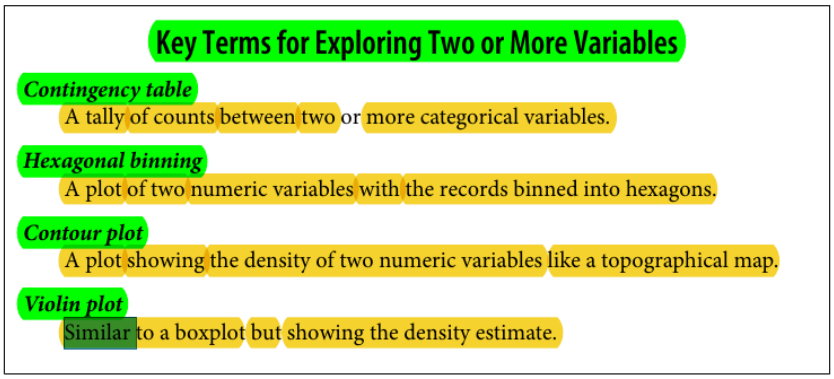
Imagine the plot is divided into four sections by the X-axis (T return = 0) and the Y-axis (VZ return = 0).

Quadrant	T Return (X-Axis)	VZ Return (Y-Axis)	Financial Meaning	Correlation Signal
Upper-Right (I)	Positive (Up)	Positive (Up)	Both stocks gained value on the same day (a "win-win" day).	Strong Positive Relationship
Lower-Left (III)	Negative (Down)	Negative (Down)	Both stocks lost value on the same day (a "lose-lose" day).	Strong Positive Relationship
Upper-Left (II)	Negative (Down)	Positive (Up)	T lost value, but VZ gained value.	Indicates Diversification / Weak or Negative Correlation
Lower-Right (IV)	Positive (Up)	Negative (Down)	T gained value, but VZ lost value.	Indicates Diversification / Weak or Negative Correlation

The Book's Key Takeaways Explained

1. **"The returns have a positive relationship... on most days, the stocks go up or go down in tandem (upper-right and lower-left quadrants)."**
    - This means the vast majority of your data points will be found in Quadrants I and III. This confirms the positive correlation: they move together.
2. **"There are fewer days where one stock goes down significantly while the other stock goes up, or vice versa (lower-right and upper-left quadrants)."**
    - This means very few points fall into Quadrants II and IV. These are the days where the stocks moved against each other. If the correlation were zero, the points would be equally distributed across all four quadrants. If the correlation were negative, Quadrants II and IV would be the most populated.
3. **"it's already obvious how difficult it is to identify details in the middle of the plot."**
    - The middle of the plot (near the origin $0, 0$) represents days when the returns are very small (e.g., -0.001 to $+0.001$). Since these points are so close together, they often overlap and look like a single blob, making it hard to see the true density and any underlying structure there.
4. **How to see more detail:**
    - The book suggests advanced techniques like **transparency** (making points see-through so overlapping areas are darker) or **hexagonal binning** (which groups points into small areas and colors those areas based on density). These methods are essential for seeing structure when you have thousands of overlapping points.

3.0.0 Exploring Multiple Variables (Multivariate Analysis)



- Univariate : one variable : Mean, Variance ….
- Bivariate : two variables : Correlation Analysis, Scatter Plot
- Multivariate : two or more variables : Additional estimates and plots (e.g., Hexagonal Binning)

⚠️ The Problem with Scatter Plots on Large Datasets

While a **Scatter Plot** is great for bivariate analysis on small datasets (like the example with 750 points), it becomes ineffective with large datasets.

- **Issue:** For data with **10,000 or millions** of records, a scatter plot will be **too dense** (often called "overplotting") and appear as a single, dark, visual cloud, making it impossible to display the data visually or see patterns.

✅ Solution: Hexagonal Binning Plot (Numeric vs. Numeric)

Hexagonal binning is a method used to visualize the relationship between two numeric variables when the dataset is too large for a standard scatter plot.

Data Context:

- **File:** The technique is demonstrated using the 1.kc_tax.csv file.
- **Variables:** This file contains variables like **"tax-assessed values"**, **"SqFtTotLiving"** (Square Feet Total Living), and **"Zipcode"**.
- **Data Preparation:** To focus the analysis, very expensive and very small or large residences are excluded using subsetting functions (e.g., subset in R or loc() in Python).

How Hexagonal Binning Works:

1. **Grouping:** The plot area (the 2D space defined by the two numeric variables) is divided into a grid of small, regular **hexagons** (the "bins").
2. **Counting:** The technique counts how many individual data records (e.g., house records) fall into each hexagonal bin.
3. **Coloring:** A color intensity is assigned to each hexagon based on the count.
    - **Darker color** implies **More records** (more houses) in that bin.

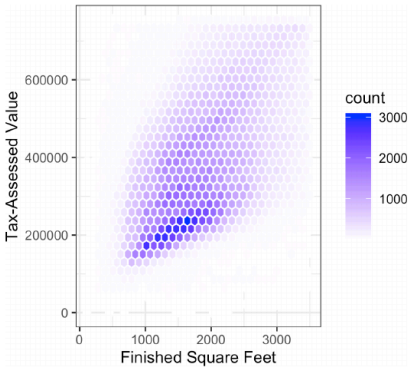


Figure 1-8. Hexagonal binning for tax-assessed value versus finished square feet

- **Histogram Bins:** Defined by a range on a **single axis (1D)**.
- **Hexagonal Bins:** Defined by a single **hexagonal shape (a 2D area)**.

Result Plot (Figure 1-8):

- Figure 1-8 displays the relationship between the **square feet** and **tax-assessed value** for homes.
- It replaces the "dark cloud" of an overplotted scatter plot with a binned visualization.
- The plot shows the records grouped into hexagonal bins, and the color intensity indicates the density of the data at that specific combination of square footage and tax value.

1. "The positive relationship... is clear."

- **Interpretation:** This confirms the primary, overall trend. As you move right along the X-axis (more Finished Square Feet), you also move up the Y-axis (higher Tax-Assessed Value). Bigger houses are generally worth more.
- **Visual Cue:** The darkest cluster of hexagons slopes upward and to the right.

2. "The Main (Darkest) Band at the Bottom"

- **Interpretation:** This dark, dense band represents the **majority of the homes** and establishes the standard, expected market rate for houses of that size.
- **Example:** A 2,000 sq ft house in the main band might be assessed at $500,000. This is the **median or average** quality and location for that size.

3. "Hint of Additional Bands Above the Main Band"

This is the key insight. The presence of *other, fainter bands of hexagons* lying above the main band means that for the same X-value (Square Footage), there are groups of homes with a significantly higher Y-value (Tax-Assessed Value).

- **Same X-Value, Higher Y-Value:** You can draw a vertical line up from a specific square footage (e.g., 2,000 sq ft). If the dark main band is at $500,000, but a fainter band is clustered at $750,000, you have found a market anomaly.

- **What This Represents:** These "additional bands" signify homes that are exactly the same size as the average home but command a **premium value**. This premium is due to non-square-footage factors that influence the tax assessment:
    - **Location:** (e.g., Waterfront, highly desirable school district).
    - **Quality/Features:** (e.g., Recent high-end renovations, luxury materials, views).
    - **Year Built:** (e.g., A historic or brand-new custom build).

✅ **Contour Plot is (Figure 1-9)**

The contour plot (often called a 2D density plot) is a visual technique borrowed from **topography** (map-making).

Contour Plot in Data: In a contour plot, the lines connect regions of equal data **density** (equal number of records/homes).

The "Peak": A closed, central contour line signifies the **highest density of points**—the most common combination of finished square footage and tax-assessed value. This is the "peak" or the highest concentration of houses.

2. Reading the Density from the Contours

- **Inner Contours (Small Loops):** Represent very high density (the most common types of homes).
- **Outer Contours (Large Loops):** Represent lower density, showing the general spread of the data.
- **The Analogy:** If you imagine the data as a mountain, the lines show you the paths you can walk that stay at the same elevation (density).

3. Confirming the "Secondary Peak" (The Key Insight)

The most important part of the quote is: **"This plot shows a similar story as Figure 1-8: there is a secondary peak 'north' of the main peak."**

- **Figure 1-8 (Hexbin):** Showed this feature as a "hint of additional bands above the main band."
- **Figure 1-9 (Contour):** Confirms this feature by showing a **second, smaller cluster of closed contour lines** located *above* the primary, darker cluster.

Since the Y-axis is the Tax-Assessed Value, the second peak being "north" (higher on the Y-axis) means:

- **Same X (Square Footage):** Both peaks are roughly aligned vertically.
- **Higher Y (Value):** The secondary peak represents a group of homes that are similarly sized to the majority but consistently have a much higher tax-assessed value.

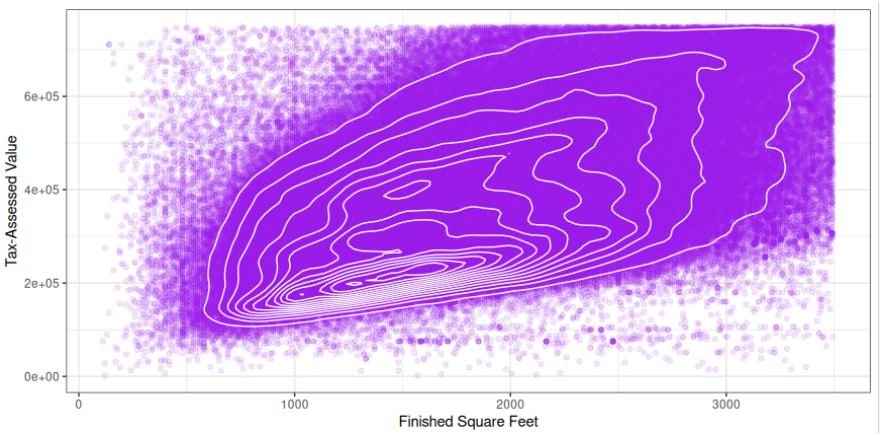
in a contour plot, you focus on the lines (or bands) themselves.

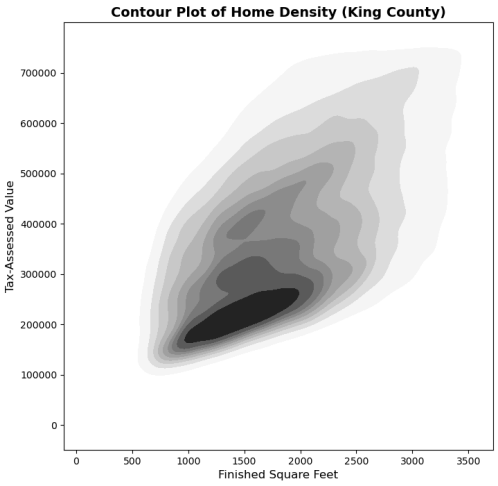
The Role of the White Lines (Contours)

In a contour plot, the "white lines" are the **contour lines** themselves. You don't necessarily focus on the white space *between* them, but on the lines, which act as boundaries.

Here is what the white lines signify:

1. **Equal Density Boundary:** Each white line connects all the points on the 2D plot that have the **same data density** (the same count or concentration of homes in that area).
2. **Topography Analogy:** Think of the plot as a mountain rising out of a plain.
    - **The White Lines (Contours):** Are lines of equal elevation.
    - **The Space Between Lines:** Represents a slope. A narrow gap between lines means a steep density gradient (density changes fast). A wide gap means a shallow gradient (density changes slowly).
3. **Increasing Density:** As you move **inward** toward the center of a closed loop of contours, the density is **increasing** until you hit the central "peak" (the highest concentration of homes).





3.1 Two Categorical Variables

A way to summarize **two categorical variables**, is a **contingency table**.
This is **contingency table** between the **grade of personal loan** and **outcome of the loan**.
This table shows **count** and **row percentages**.
We can see that **High-grade loans** have a **very low late/charge-off percentage** as compared with **lower-grade loans**.

Table 1-8. Contingency table of loan grade and status

Grade	Charged off	Current	Fully paid	Late	Total
A	1562	50051	20408	469	72490
	0.022	0.690	0.282	0.006	0.161
B	5302	93852	31160	2056	132370
	0.040	0.709	0.235	0.016	0.294
C	6023	88928	23147	2777	120875
	0.050	0.736	0.191	0.023	0.268
D	5007	53281	13681	2308	74277
	0.067	0.717	0.184	0.031	0.165
E	2842	24639	5949	1374	34804
	0.082	0.708	0.171	0.039	0.077
F	1526	8444	2328	606	12904
	0.118	0.654	0.180	0.047	0.029
G	409	1990	643	199	3241
	0.126	0.614	0.198	0.061	0.007
Total	22671	321185	97316	9789	450961

Task	R (descr::CrossTable)	
Primary Goal	Generate a table with counts and statistical tests (chi^2).	Generate a summary table with counts or any other aggregate function.
Function	CrossTable(row_var, col_var, ...)	df.pivot_table(index, columns, aggfunc, ...)
Aggregation (Counting)	Automatic by default.	aggfunc=lambda x: len(x) or aggfunc='count'
Row Percentages	prop.r = TRUE (Displays the row proportion within the cell).	Requires manual normalization: df[cols].div(df['All'], axis=0)
Column Percentages	prop.c = TRUE (Displays the column proportion within the cell).	Requires manual normalization: df[cols].div(df['All'], axis=1)
Displaying Totals	total = TRUE (default).	margins=True (Adds the 'All' column/row total).
Example (Loans)	CrossTable(lc_loans$grade, lc_loans$status, prop.r = TRUE)	Step 1: Create the count table. Step 2: Apply division.

3.2 Categorical and Numerical Data

Using a **boxplot** is a simple way to visualize **numerical data** grouped according to a **categorical variable** (e.g., how the **percentage of flight delay** varies across **different airlines**).

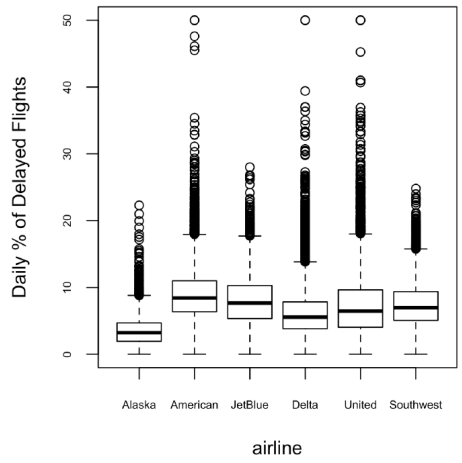


Figure 1-10. Boxplot of percent of airline delays by carrier

Violin Plot

A **violin plot** is an enhancement to a **boxplot**, plotting the **density estimates** with **density on the y-axis**.

The advantage of a **violin plot** is that it can show **nuances in the distribution** that aren't perceptible in a **boxplot**.

However, a **boxplot** is clearer at **showing outliers** in the data.

For Shape/Density: It shows where most of your data points are clustered. The "fatter" the violin, the more elements exist at that specific value

Feature	R (ggplot2)	
Main Function	ggplot(...) + geom_violin()	sns.violinplot(...)
Data Input	data=airline_stats (in ggplot)	data=airline_stats (in sns.violinplot)
Defining Axes (Aesthetics)	aes(x=airline, y=pct_carrier_delay)	x='airline', y='pct_carrier_delay'
Adding Internal Statistics	(Requires adding geom_boxplot() or similar)	inner='quartile' (or 'box', 'point')
Defining Y-Axis Limit	ylim(0, 50) (as a separate layer)	ax.set_ylim(0, 50) (method of the Axes object)
Full Plot Syntax	r ggplot(data=df, aes(x=category, y=value)) + geom_violin() + ylim(0, max_y)	```python
plt.figure()		
ax = sns.violinplot(data=df, x='category', y='value')		
ax.set_ylim(0, max_y)		

3.3 Visualizing Multiple Variables

So till now we have seen **2D analysis**, **scatterplots**, **hexagonal binning**, and **boxplots**.
These plots can be used in **multiple variable analysis** through the notion of **conditioning**.

As we see in **Figure 1-8**, the **relationship between tax assessed value and sqf** is shown with different **bands**.
In the next figure, we will account for the effect of the **location of the home** by plotting **ZIP code data**.

Tax assessed value is much higher in **98105** and **98126** than in others. This disparity gives rise to the **clusters observed** in **Figure 1-8**.

Condition here is ZIP code.

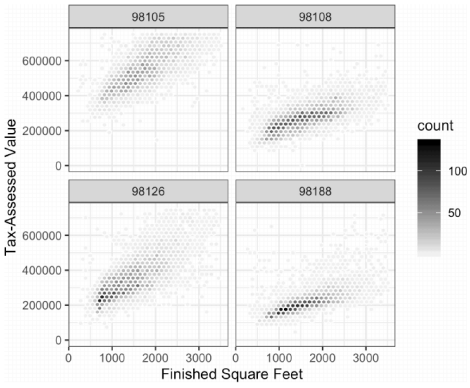


Figure 1-12. Tax-assessed value versus finished square feet by zip code

Concept of Conditioning Variables in Graphic Plots

Conditioning variables are also integral to **business intelligence platforms** such as **Tableau** and **Spotfire**. With the advent of **vast computing power**, modern **visualization platforms** have moved well beyond the humble beginnings of **exploratory data analysis**. However, **key concepts and tools** developed a half-century ago (e.g., simple **boxplots**) still form a **foundation** for these systems.

While it is in principle possible to create **faceted graphs** (show **multiple graphs in one chart**) using **Matplotlib**, the code can get **complicated**. Fortunately, **Seaborn** has a relatively **straightforward way** of creating these graphs.

Key Ideas

- Hexagonal binning and contour plots are useful tools that permit graphical examination of two numeric variables at a time, without being overwhelmed by huge amounts of data.
- Contingency tables are the standard tool for looking at the counts of two categorical variables.
- Boxplots and violin plots allow you to plot a numeric variable against a categorical variable.

Summary

Exploratory data analysis (EDA), pioneered by John Tukey, set a foundation for the field of data science. The key idea of EDA is that the first and most important step in any project based on data is to *look at the data*. By summarizing and visualizing the data, you can gain valuable intuition and understanding of the project.

This chapter has reviewed concepts ranging from simple metrics, such as estimates of location and variability, to rich visual displays that explore the relationships between multiple variables, as in Figure 1-12. The diverse set of tools and techniques being developed by the open source community, combined with the expressiveness of the *R* and *Python* languages, has created a plethora of ways to explore and analyze data. Exploratory analysis should be a cornerstone of any data science project.