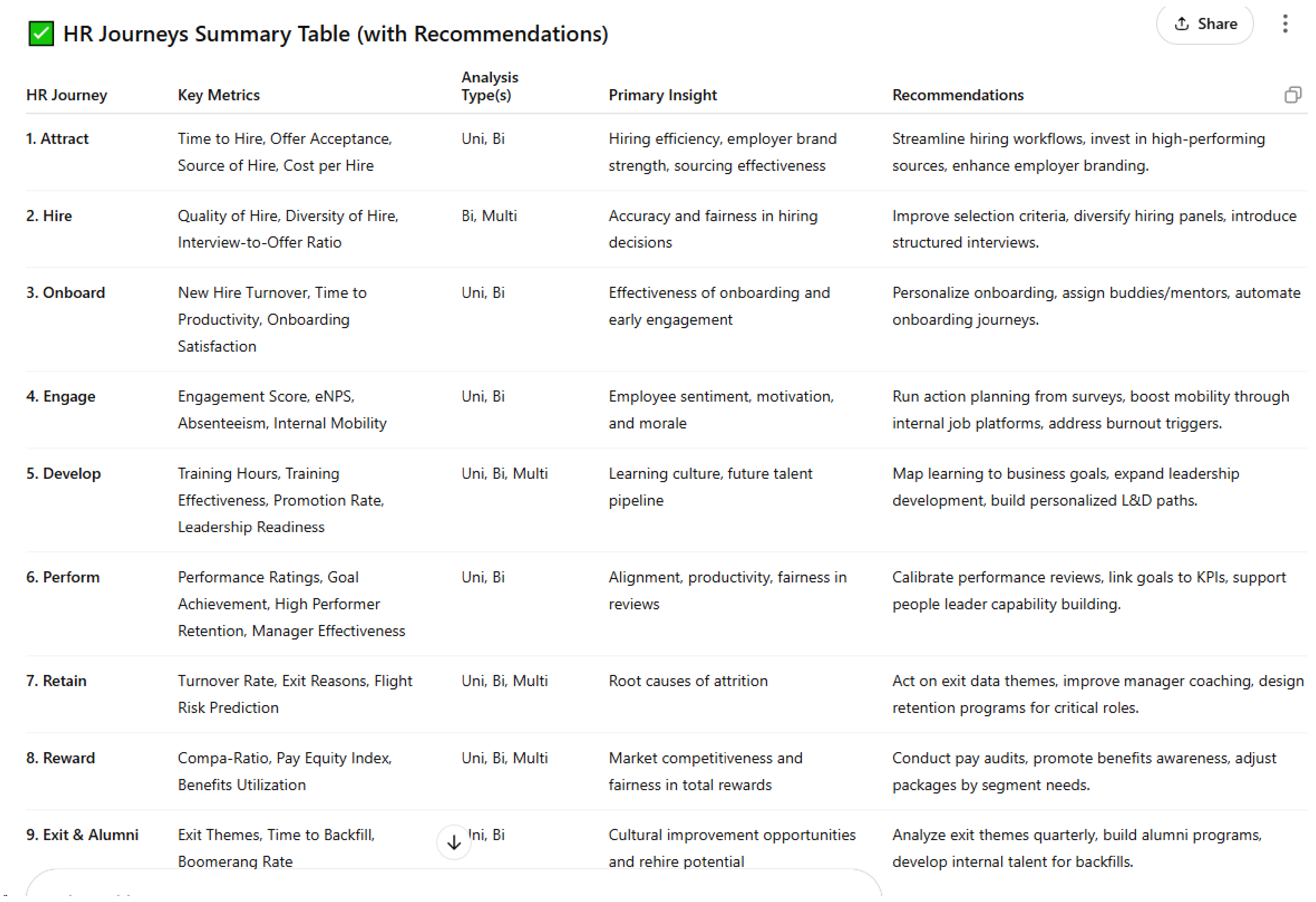
Class 1
HR Analytics

Journeys



✅ HR Journeys Summary Table (with Recommendations)

HR Journey	Key Metrics	Analysis Type(s)	Primary Insight	Recommendations
1. Attract	Time to Hire, Offer Acceptance, Source of Hire, Cost per Hire	Uni, Bi	Hiring efficiency, employer brand strength, sourcing effectiveness	Streamline hiring workflows, invest in high-performing sources, enhance employer branding.
2. Hire	Quality of Hire, Diversity of Hire, Interview-to-Offer Ratio	Bi, Multi	Accuracy and fairness in hiring decisions	Improve selection criteria, diversify hiring panels, introduce structured interviews.
3. Onboard	New Hire Turnover, Time to Productivity, Onboarding Satisfaction	Uni, Bi	Effectiveness of onboarding and early engagement	Personalize onboarding, assign buddies/mentors, automate onboarding journeys.
4. Engage	Engagement Score, eNPS, Absenteeism, Internal Mobility	Uni, Bi	Employee sentiment, motivation, and morale	Run action planning from surveys, boost mobility through internal job platforms, address burnout triggers.
5. Develop	Training Hours, Training Effectiveness, Promotion Rate, Leadership Readiness	Uni, Bi, Multi	Learning culture, future talent pipeline	Map learning to business goals, expand leadership development, build personalized L&D paths.
6. Perform	Performance Ratings, Goal Achievement, High Performer Retention, Manager Effectiveness	Uni, Bi	Alignment, productivity, fairness in reviews	Calibrate performance reviews, link goals to KPIs, support people leader capability building.
7. Retain	Turnover Rate, Exit Reasons, Flight Risk Prediction	Uni, Bi, Multi	Root causes of attrition	Act on exit data themes, improve manager coaching, design retention programs for critical roles.
8. Reward	Compa-Ratio, Pay Equity Index, Benefits Utilization	Uni, Bi, Multi	Market competitiveness and fairness in total rewards	Conduct pay audits, promote benefits awareness, adjust packages by segment needs.
9. Exit & Alumni	Exit Themes, Time to Backfill, Boomerang Rate	Uni, Bi	Cultural improvement opportunities and rehire potential	Analyze exit themes quarterly, build alumni programs, develop internal talent for backfills.

Compare between different Times

Quarter to Quarter

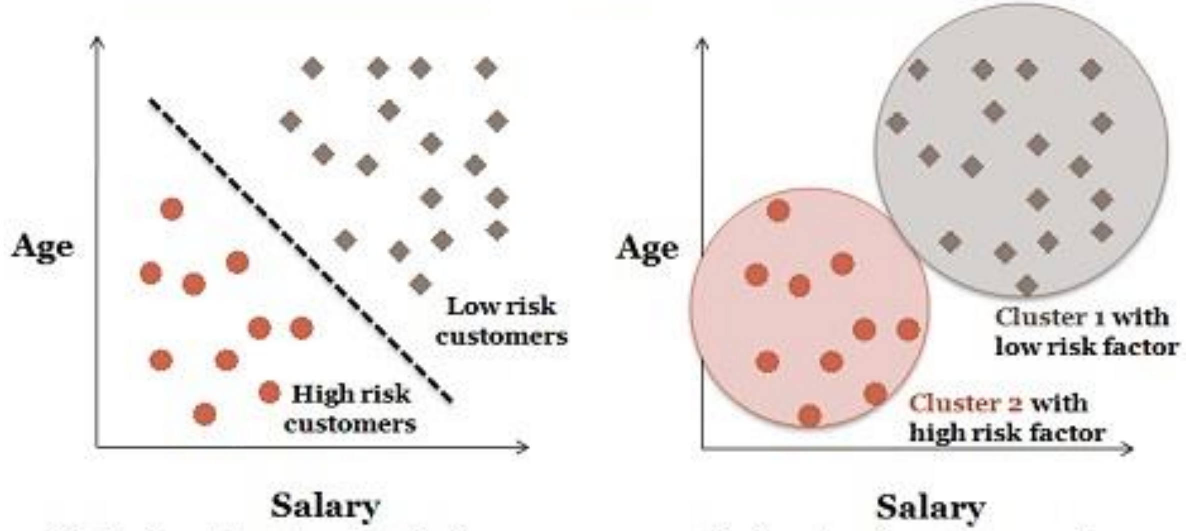
Compere between different Companies or Market

Company score Vs Market

Data Driven Decision Making

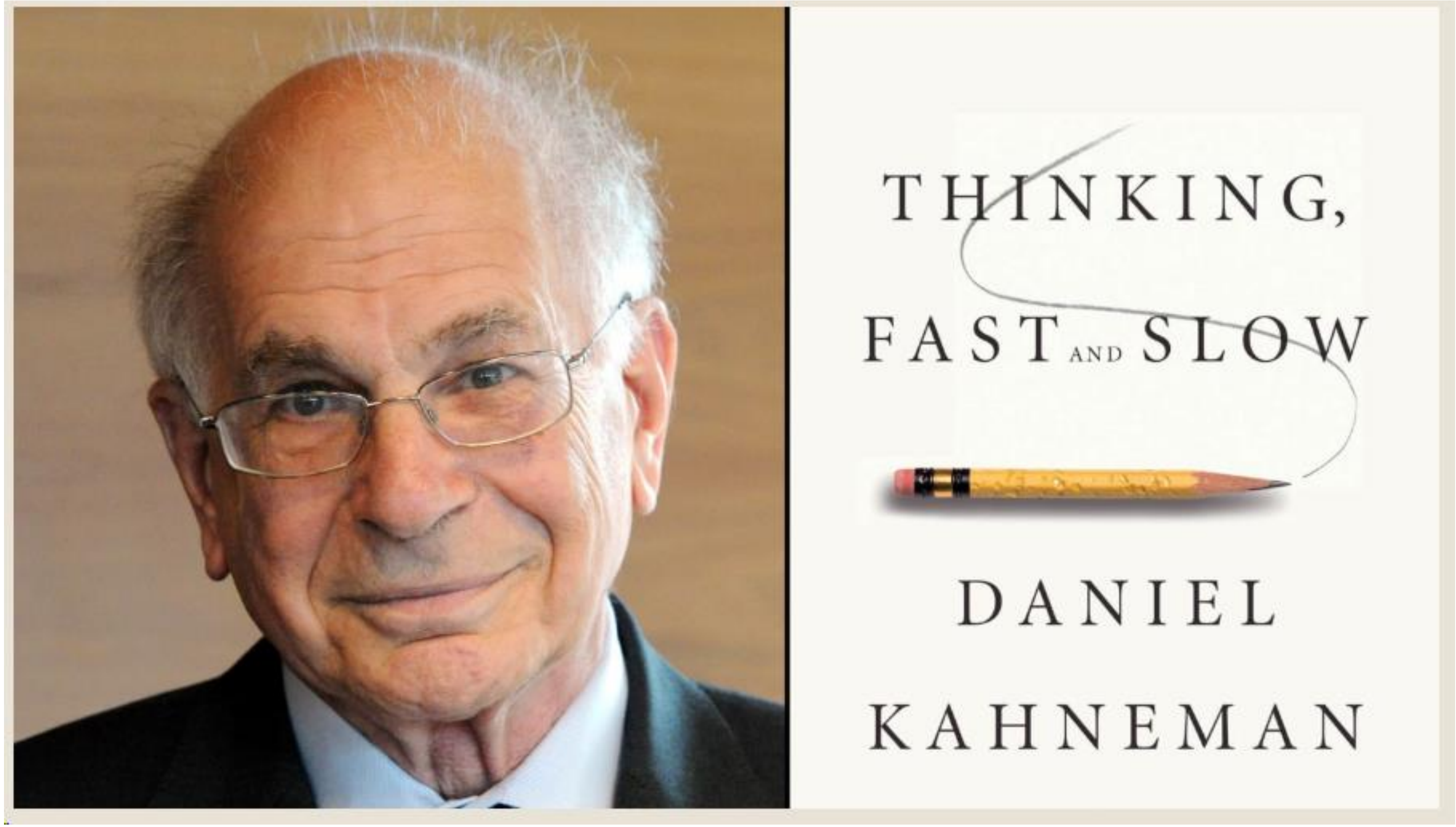
The Study of Dots

Class No : 1



Risk classification for the loan payees on the basis of customer salary

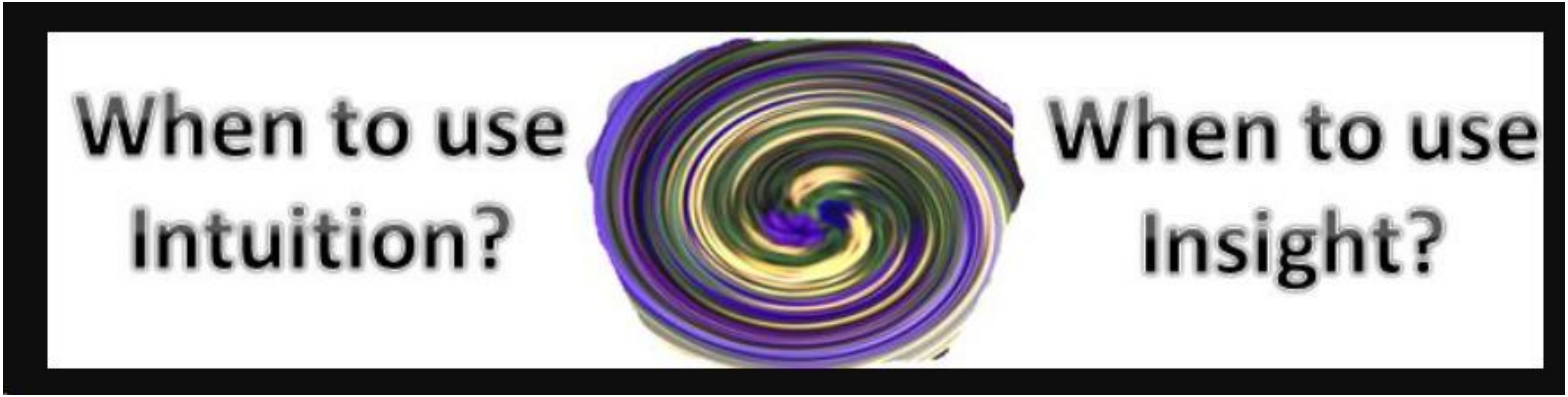
The 2 Approaches



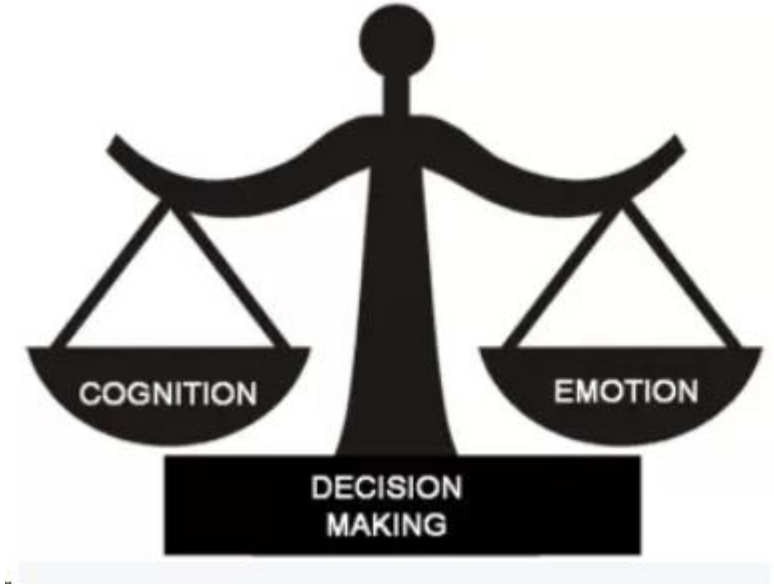
The 2 Approaches

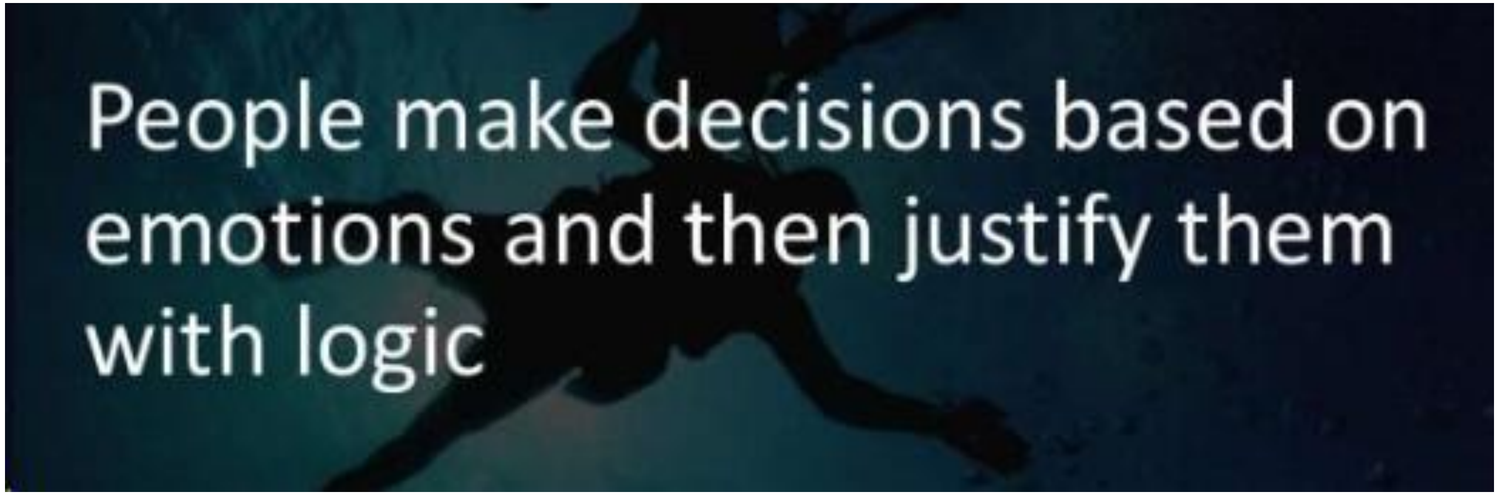
System 1

System 2



Tendencies





People make decisions based on emotions and then justify them with logic

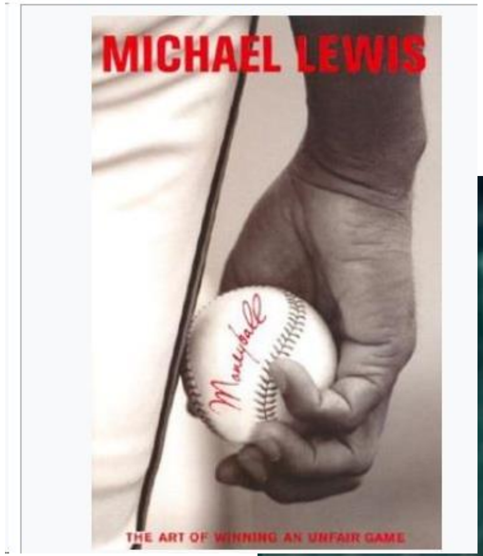
Heuristics = Mental Shortcuts

- Recency Effect
- Similarity Effect
- Anchoring Effect
- Confirmation Bias
- Loss Aversion
- Good Enough
- Known Devil is better Effect
- Rare and Scarcity Effect
- Superior Brand Effect
- Feel Good and Bad Effect

Inspiration the Movie

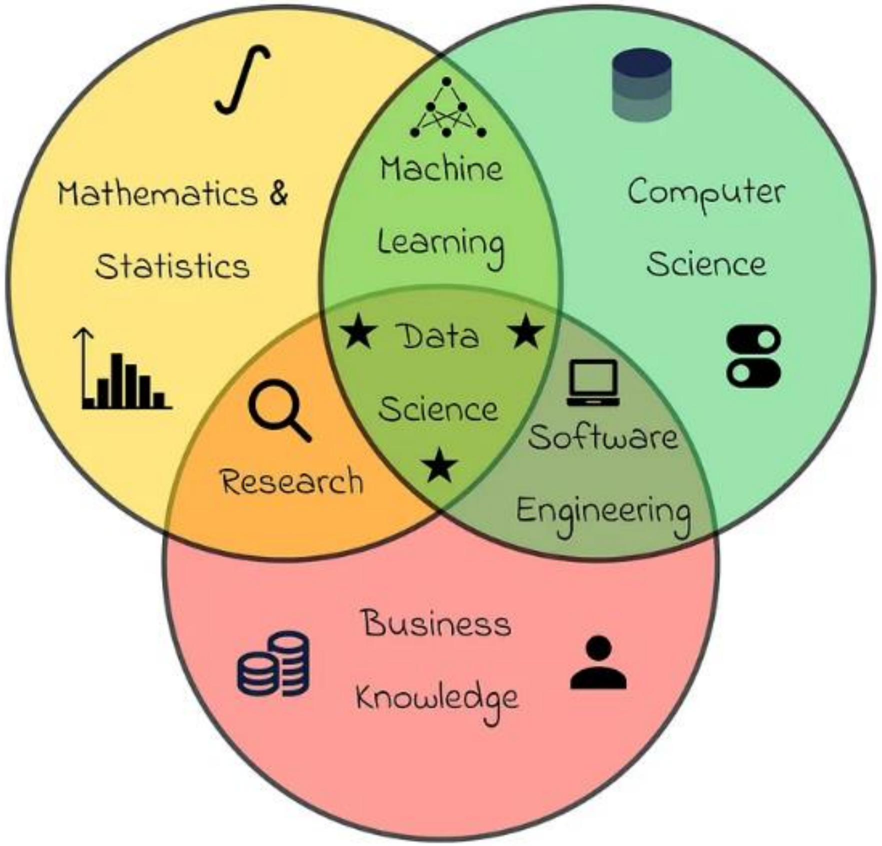
Moneyball

The Art of Winning an Unfair Game



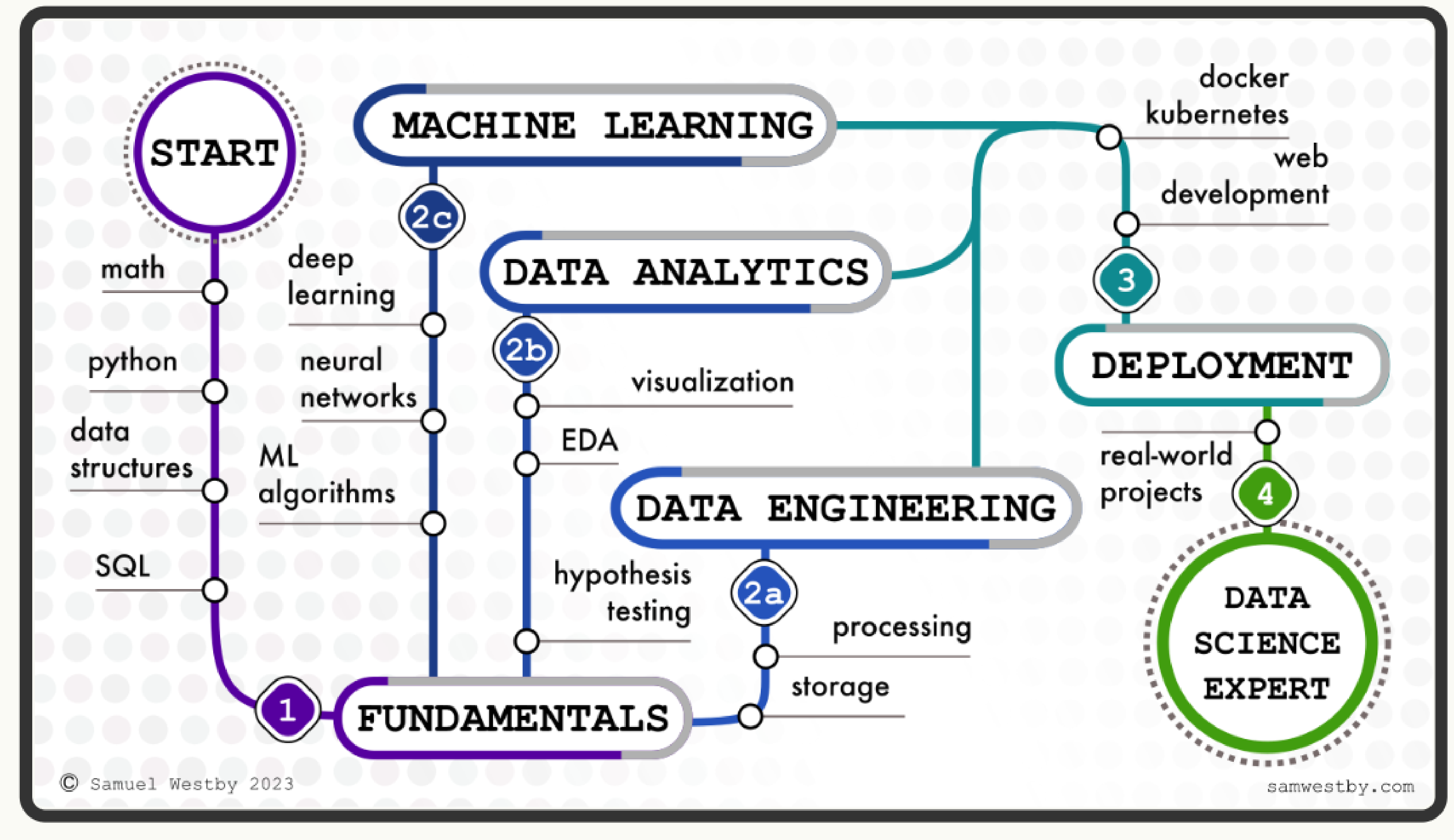
Bill James, who coined the term "sabermetrics"

Scope



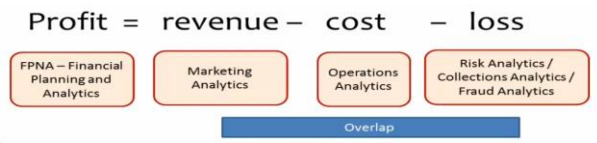
Venn diagram presenting the key Data Science components

Full stack HR Analyst

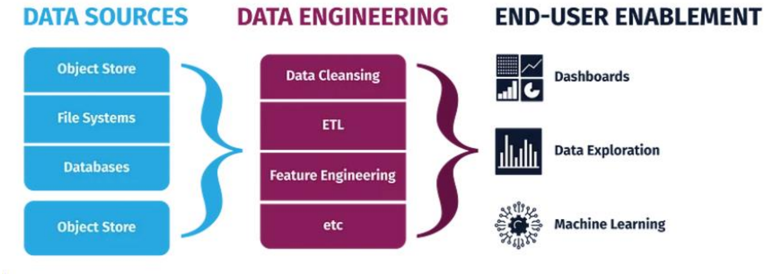


Data Analytics Framework

Data Strategy (Business Value)


Profit = revenue – cost – loss
FPNA – Financial Planning and Analytics
Marketing Analytics
Operations Analytics
Risk Analytics / Collections Analytics / Fraud Analytics
Overlap

Data Engineering /Management (Infrastructure)

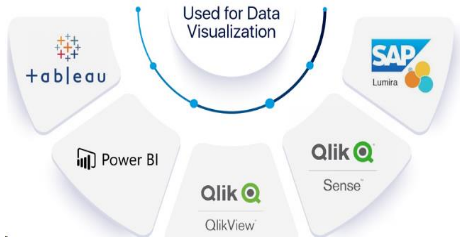


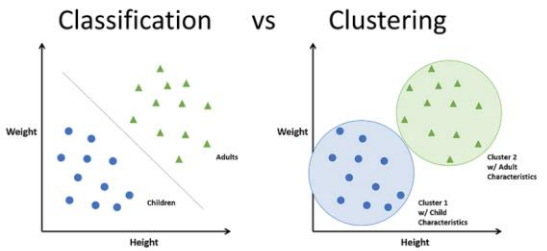
Data Visualization (Descriptive)

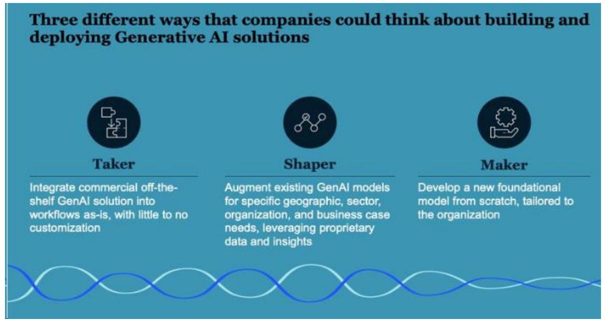
Data Modeling (Discriminative and Generative)

Discriminative

Generative







Three different ways that companies could think about building and deploying Generative AI solutions

Taker

Integrate commercial off-the-shelf GenAI solution into workflows as-is, with little to no customization

Shaper

Augment existing GenAI models for specific geographic, sector, organization, and business case needs, leveraging proprietary data and insights

Maker

Develop a new foundational model from scratch, tailored to the organization

FOUR CATEGORIES OF ANALYTICS

Data analytics techniques are commonly described as part of four distinct categories: **descriptive**, **diagnostic**, **predictive** and **prescriptive**

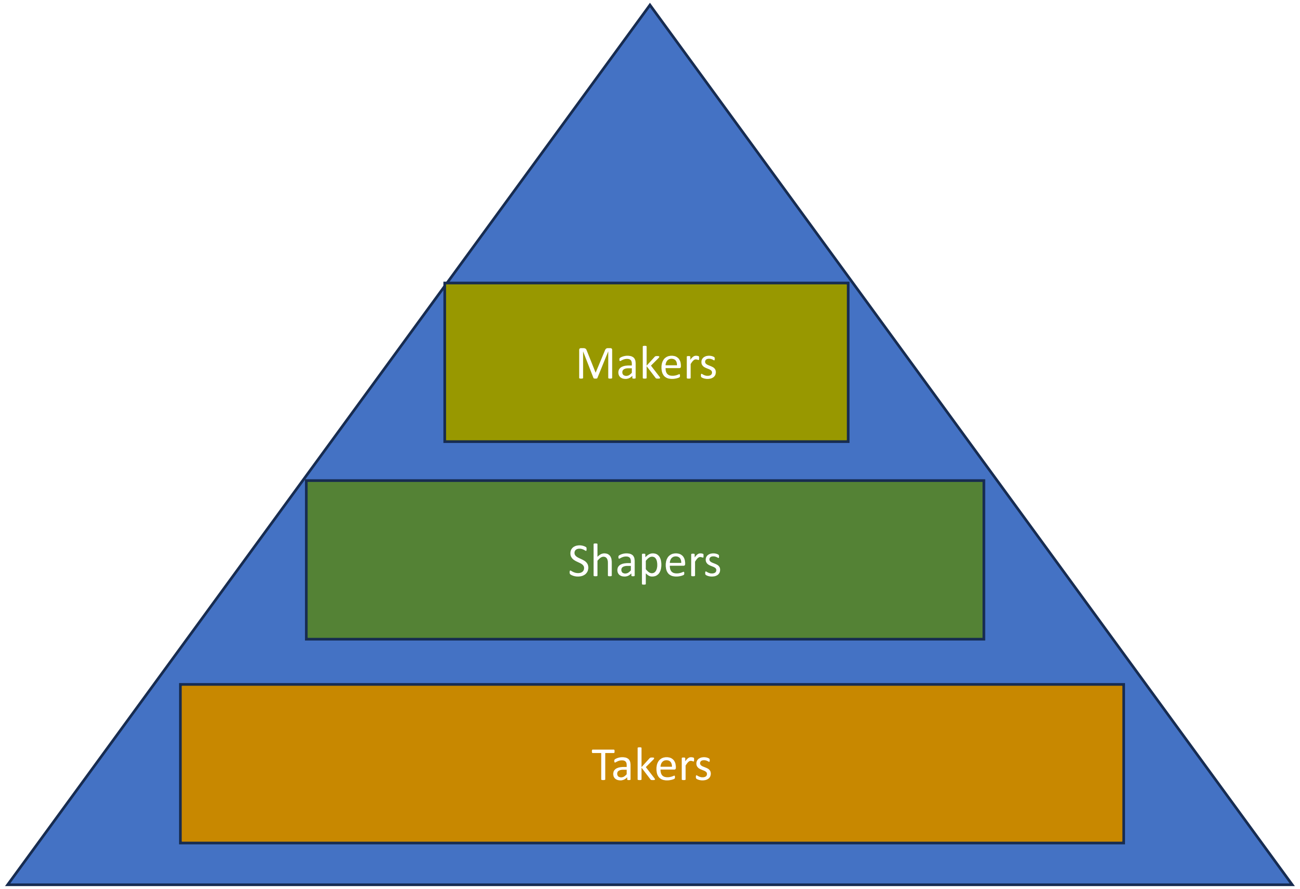
DESCRIPTIVE
What happened?

DIAGNOSTIC
Why did it happen?

PREDICTIVE
What will happen next?

PRESCRIPTIVE
What's our course of action?

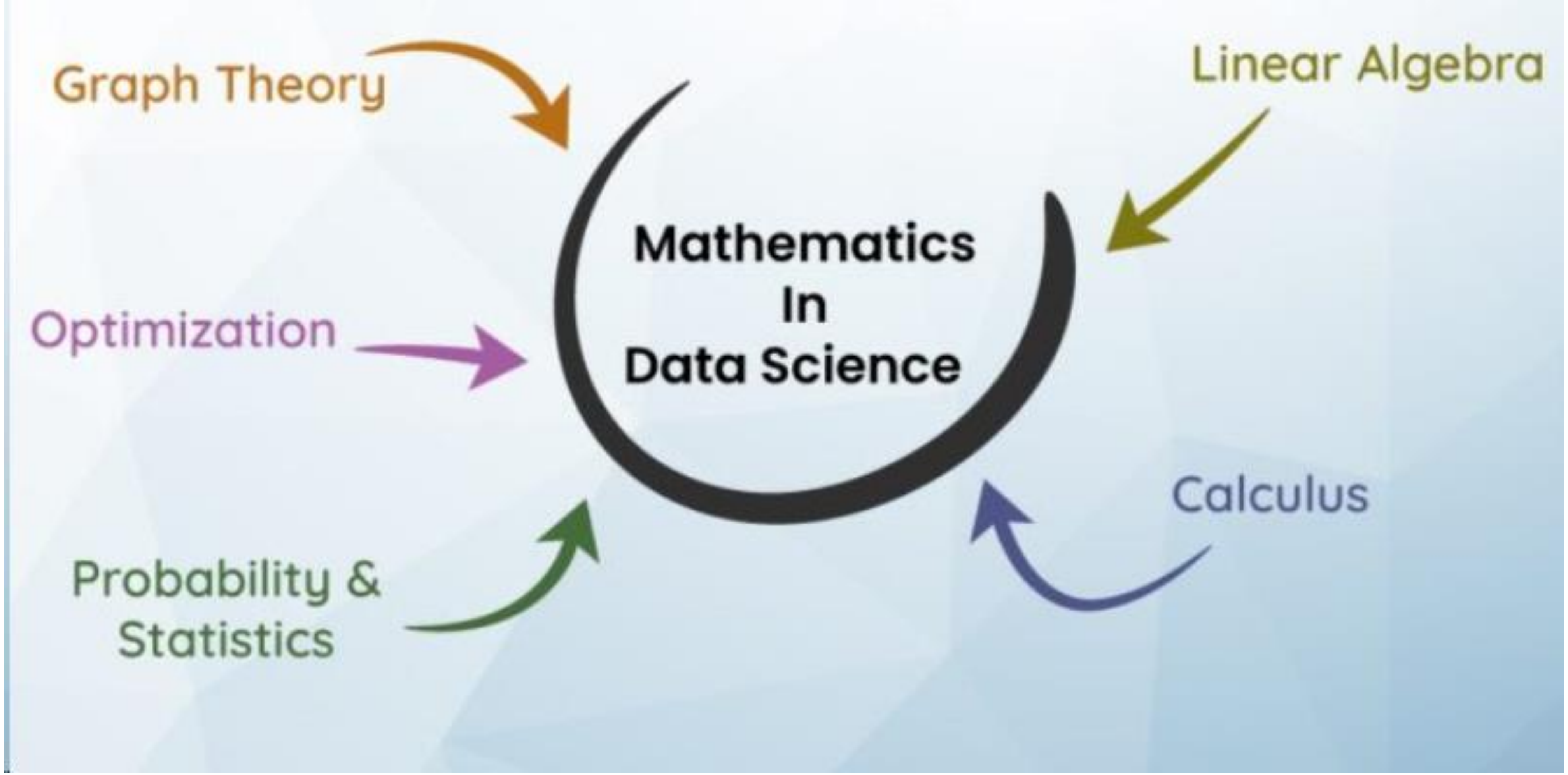
Adoption Levels



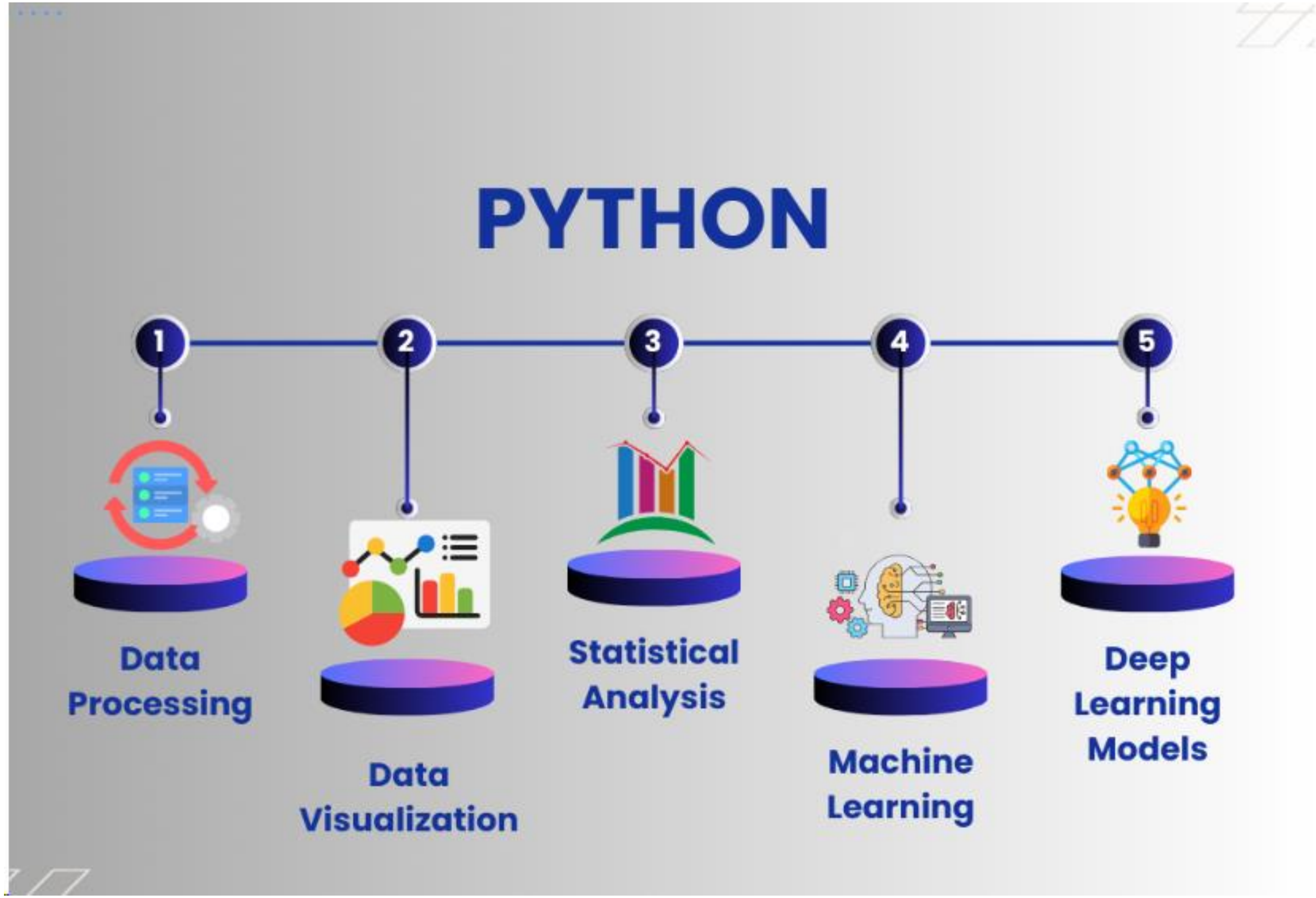
Stats Required

- Descriptive Statistics
- Inferential Statistics
- Probability Destributions
- Sampling Methods
- Regression Analysis
- Hypothesis Testing
- Data Visualization
- Anova
- Time Series Analysis
- Bayesian Statsitics

Maths Required



Programming Required



Required

- Google Account
- Collab Access
- Sam’s Github Access
- Datasets
- ChatGPT
- Ananconda Distribution
- Vscode
- Terminal (Windows)

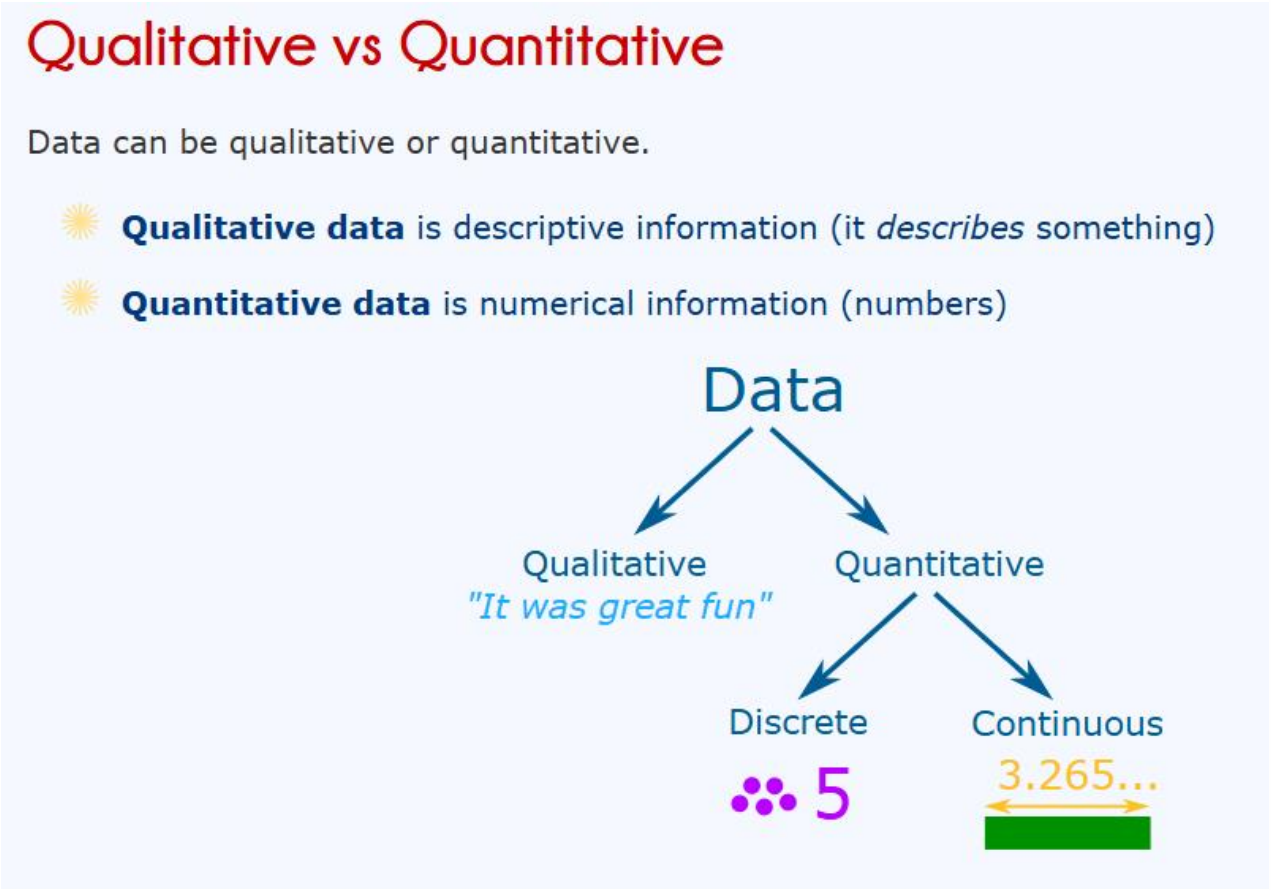
Dataset – for Descriptive Analytics

p1	p2	p3	p4
Columns			
Features			
Properties			
Dimensions			

Dataset – for Descriptive Analytics

Employee No	Name	Tier	Designation	Gender	Salary
Columns					
Features					
Properties					
Dimensions					

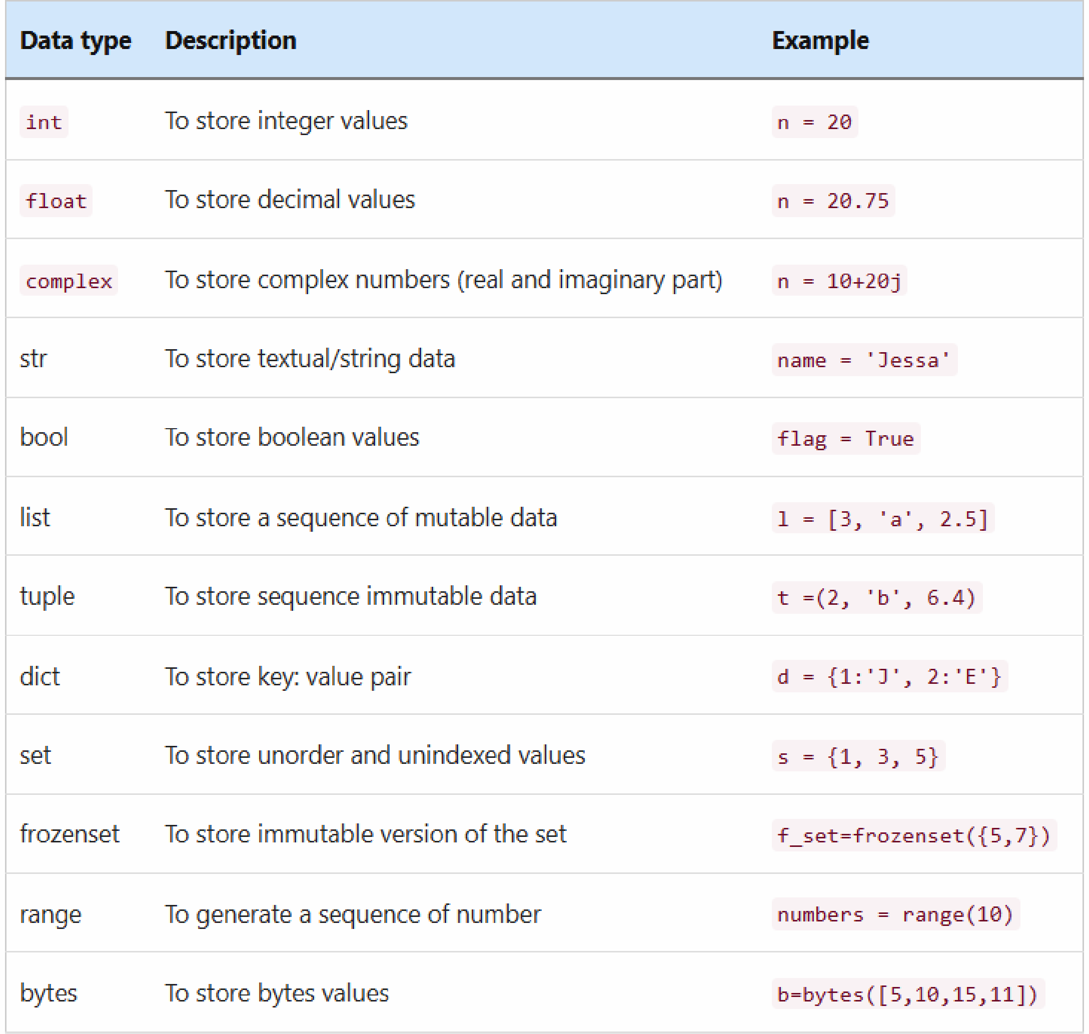
Data Types



Qualitative vs Quantitative

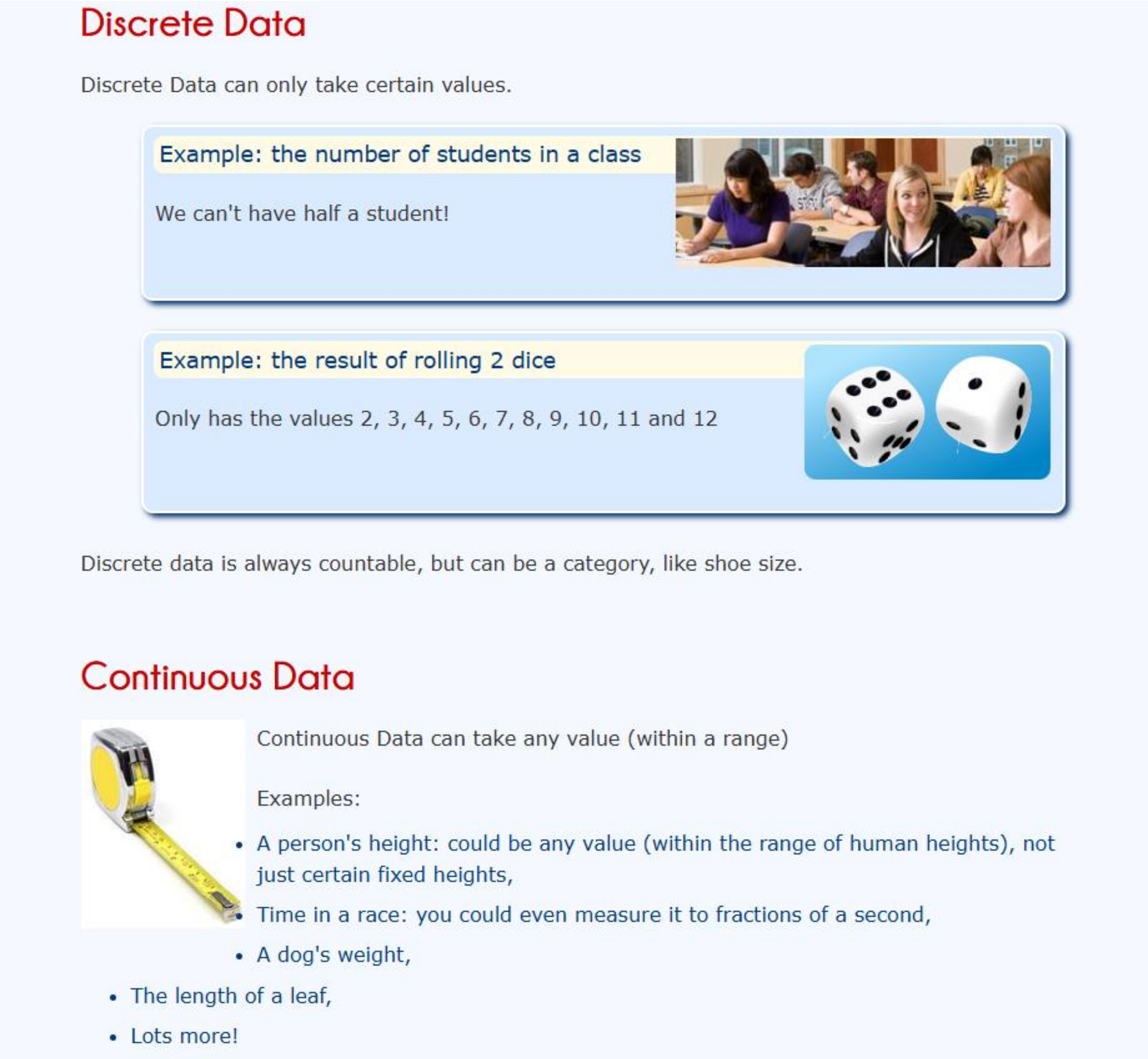
Data can be qualitative or quantitative.

- **Qualitative data** is descriptive information (it *describes* something)
- **Quantitative data** is numerical information (numbers)



Data type	Description	Example
`int`	To store integer values	`n = 20`
`float`	To store decimal values	`n = 20.75`
`complex`	To store complex numbers (real and imaginary part)	`n = 10+20j`
str	To store textual/string data	`name = 'Jessa'`
bool	To store boolean values	`flag = True`
list	To store a sequence of mutable data	`l = [3, 'a', 2.5]`
tuple	To store sequence immutable data	`t =(2, 'b', 6.4)`
dict	To store key: value pair	`d = {1:'J', 2:'E'}`
set	To store unorder and unindexed values	`s = {1, 3, 5}`
frozenset	To store immutable version of the set	`f_set=frozenset({5,7})`
range	To generate a sequence of number	`numbers = range(10)`
bytes	To store bytes values	`b=bytes([5,10,15,11])`

Python Data Types



Discrete Data

Discrete Data can only take certain values.

Example: the number of students in a class

We can't have half a student!

Example: the result of rolling 2 dice

Only has the values 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12

Discrete data is always countable, but can be a category, like shoe size.

Continuous Data

Continuous Data can take any value (within a range)

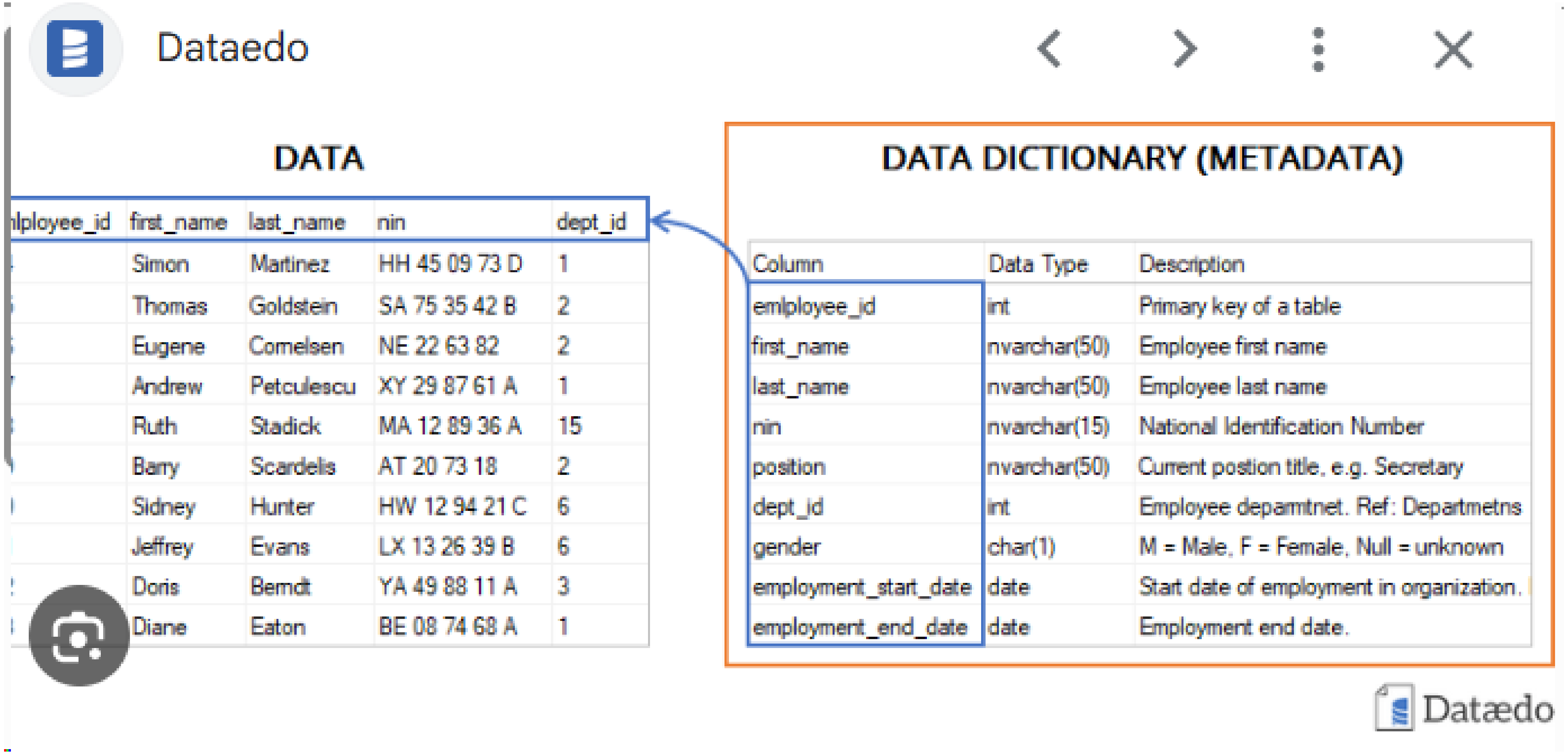
Examples:

- A person's height: could be any value (within the range of human heights), not just certain fixed heights,
- Time in a race: you could even measure it to fractions of a second,
- A dog's weight,
- The length of a leaf,
- Lots more!

Data Types : Categorical & Numerical

p1	p2	p3	p4
Manager	48	25,000	Male
Executive	25	45,000	Female
Consultant	33	64,000	Male

Governance : Data Dictionary



Dataedo

DATA

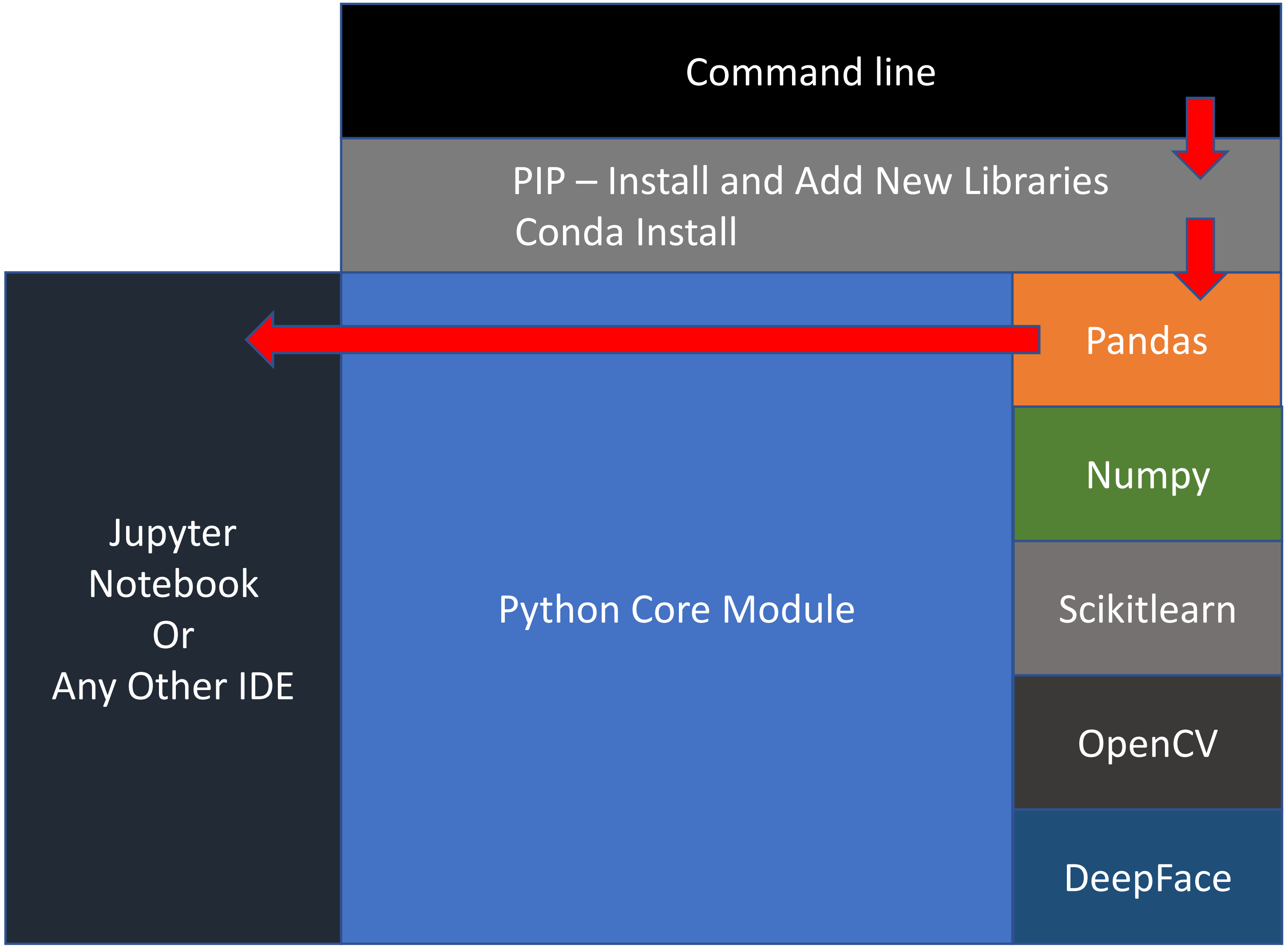
nlployee_id	first_name	last_name	nin	dept_id
	Simon	Martinez	HH 45 09 73 D	1
	Thomas	Goldstein	SA 75 35 42 B	2
	Eugene	Comelsen	NE 22 63 82	2
	Andrew	Petculescu	XY 29 87 61 A	1
	Ruth	Stadick	MA 12 89 36 A	15
	Barry	Scardelis	AT 20 73 18	2
	Sidney	Hunter	HW 12 94 21 C	6
	Jeffrey	Evans	LX 13 26 39 B	6
	Doris	Berndt	YA 49 88 11 A	3
	Diane	Eaton	BE 08 74 68 A	1

DATA DICTIONARY (METADATA)

Column	Data Type	Description
emlployee_id	int	Primary key of a table
first_name	nvarchar(50)	Employee first name
last_name	nvarchar(50)	Employee last name
nin	nvarchar(15)	National Identification Number
position	nvarchar(50)	Current postion title, e.g. Secretary
dept_id	int	Employee depamtnet. Ref: Departmetns
gender	char(1)	M = Male, F = Female, Null = unknown
employment_start_date	date	Start date of employment in organization.
employment_end_date	date	Employment end date.

Datædo

Python Data Science Suite



Command line

PIP – Install and Add New Libraries
Conda Install

Jupyter
Notebook
Or
Any Other IDE

Python Core Module

Pandas

Numpy

Scikitlearn

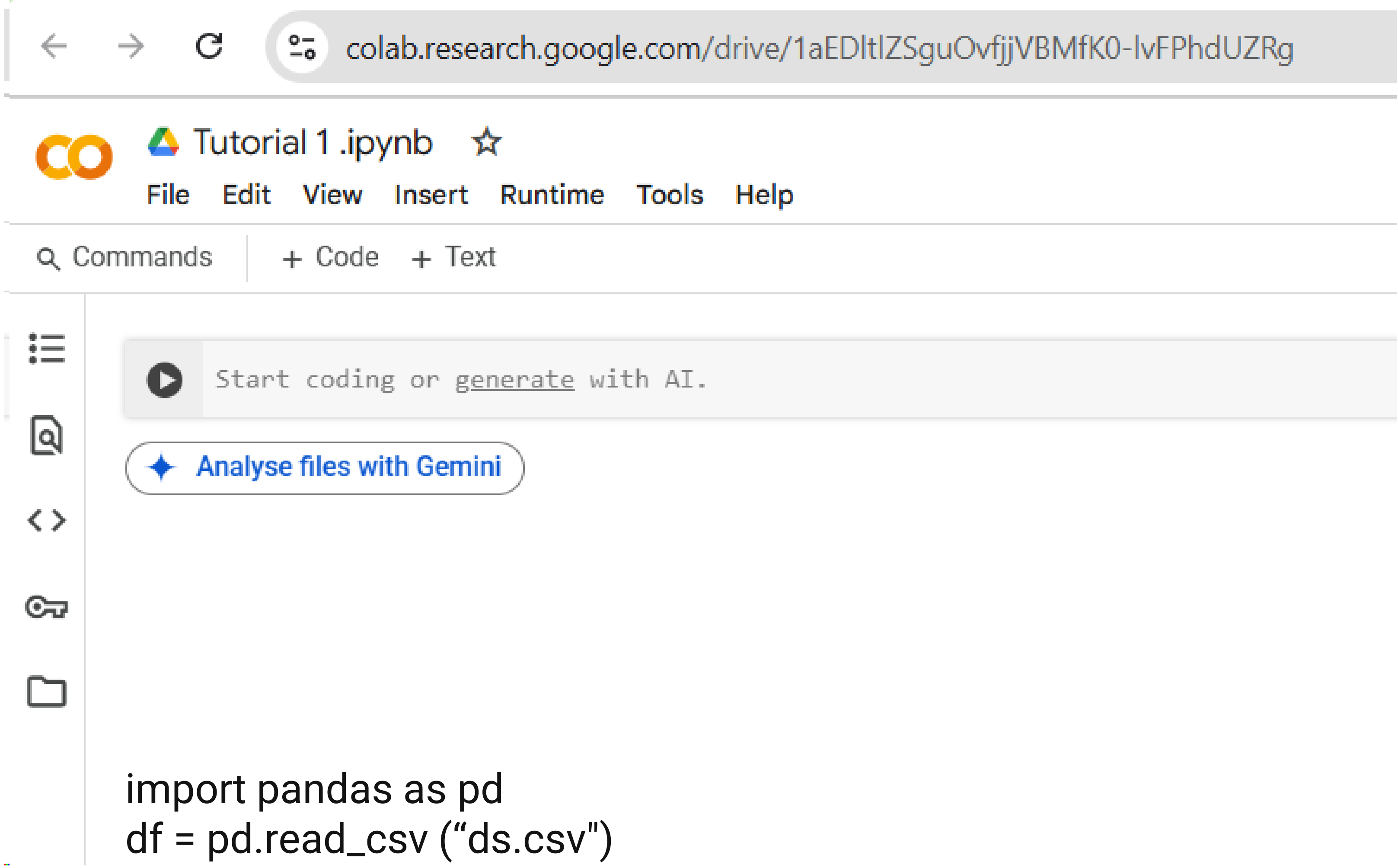
OpenCV

DeepFace

Prompt Engineering

- You **Can’t** Do Prompting if you **Don’t** Know the **Basics** and the **Subject**

Convert to a Data Frame / Table


colab.research.google.com/drive/1aEDltlZSguOvfjjVBMfK0-lvFPhdUZRg
Tutorial 1 .ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Commands
+ Code
+ Text
Start coding or generate with AI.
Analyse files with Gemini

```
import pandas as pd
df = pd.read_csv (“ds.csv")
df.head()
```

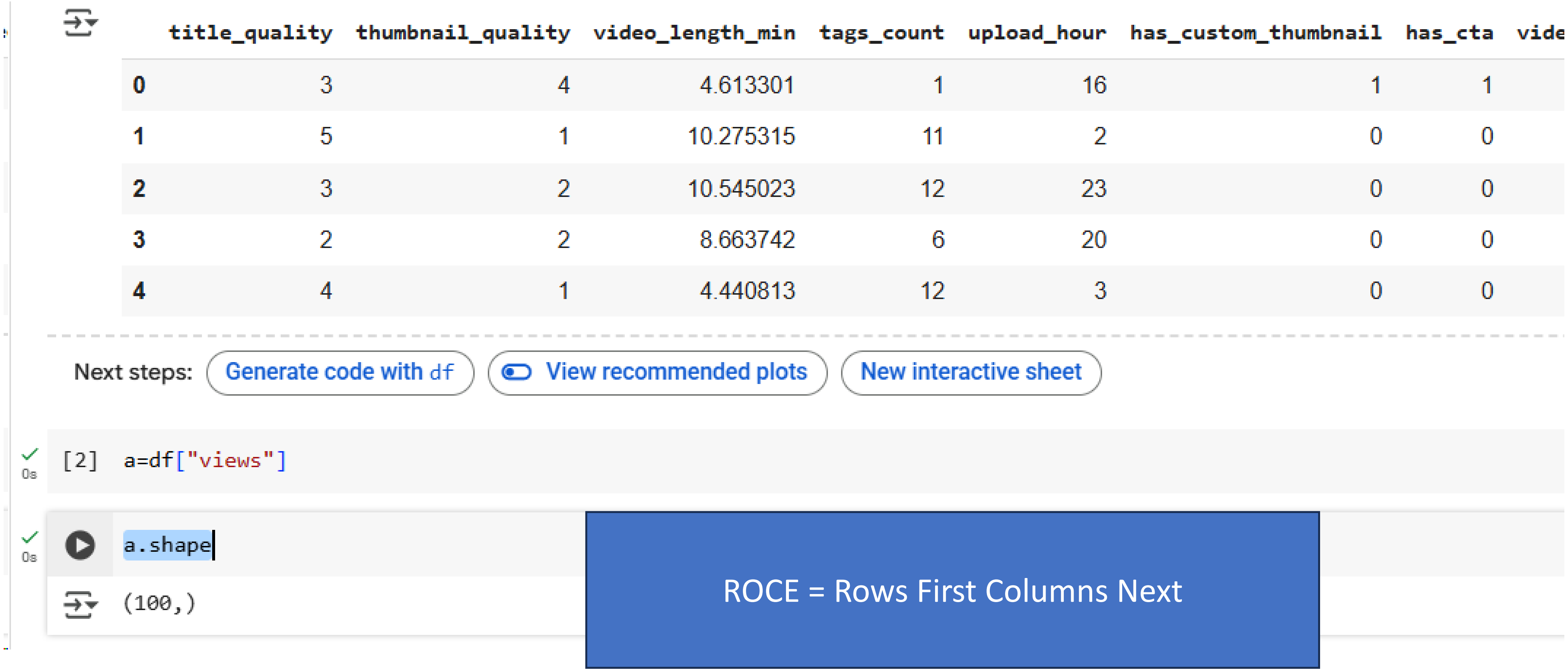
Assign a Matrix to a Variable

p1	p2	p3	p4

```
Y=df['Sales’]

X=df[['TV','Newspaper','Radio']]
```

Check Shape

```
a.shape
```



	title_quality	thumbnail_quality	video_length_min	tags_count	upload_hour	has_custom_thumbnail	has_cta	vide
0	3	4	4.613301	1	16	1	1	
1	5	1	10.275315	11	2	0	0	
2	3	2	10.545023	12	23	0	0	
3	2	2	8.663742	6	20	0	0	
4	4	1	4.440813	12	3	0	0	

```
[2] a=df["views"]
```

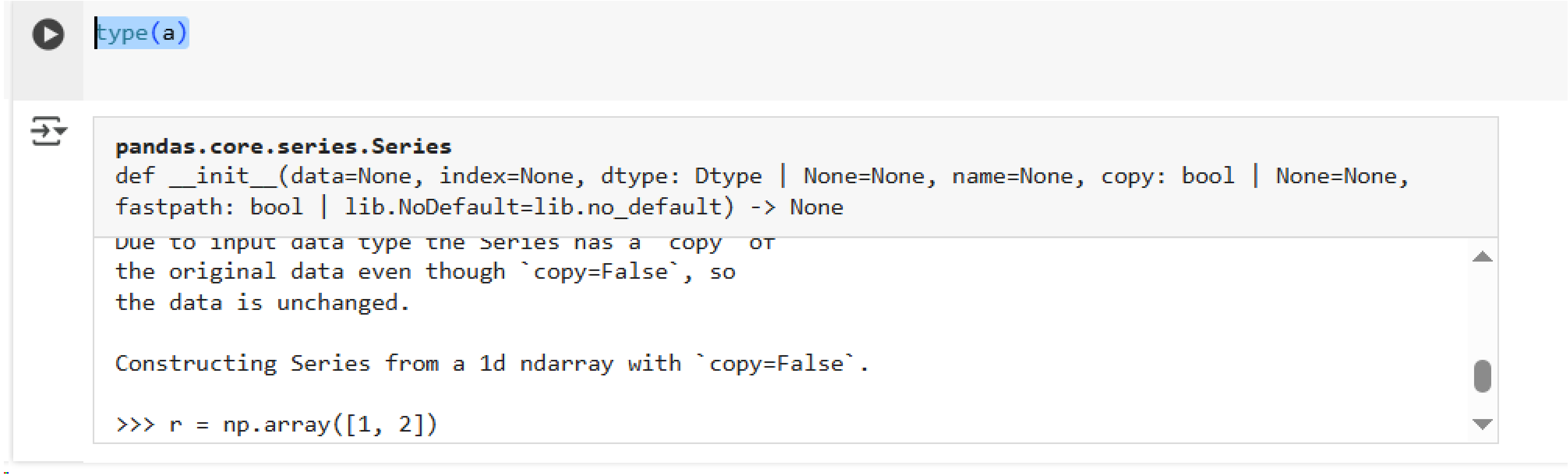
```
a.shape
```

```
(100,)
```

ROCE = Rows First Columns Next

Check Variable Type

```
type(a)
```



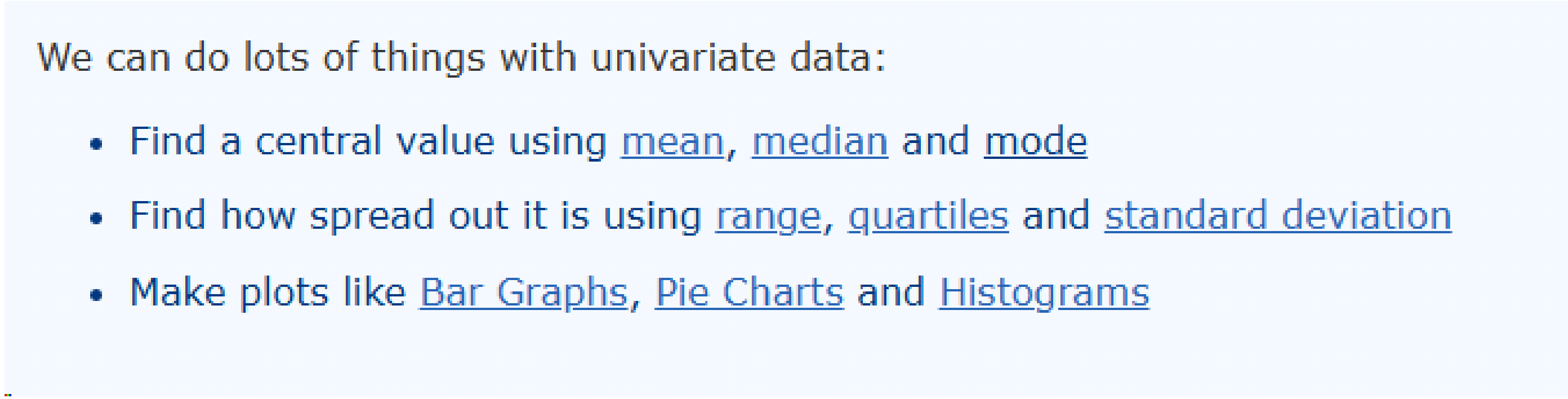
```
type(a)
```

```
pandas.core.series.Series
def __init__(data=None, index=None, dtype: Dtype | None=None, name=None, copy: bool | None=None,
fastpath: bool | lib.NoDefault=lib.no_default) -> None
Due to input data type the Series has a `copy` of
the original data even though `copy=False`, so
the data is unchanged.

Constructing Series from a 1d ndarray with `copy=False`.

>>> r = np.array([1, 2])
```

Analytics

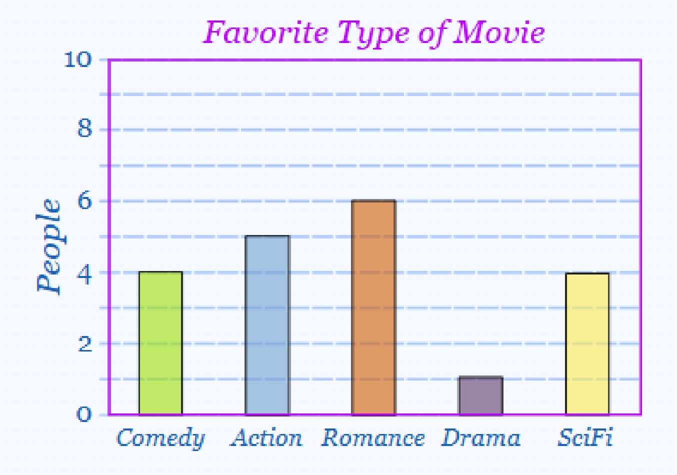
- Univariate Analysis – Only 1 Variable
- Bi-Variate – 2 Variables and Relationship
- Muti-Variate – More than 2 Variables and their Relationships

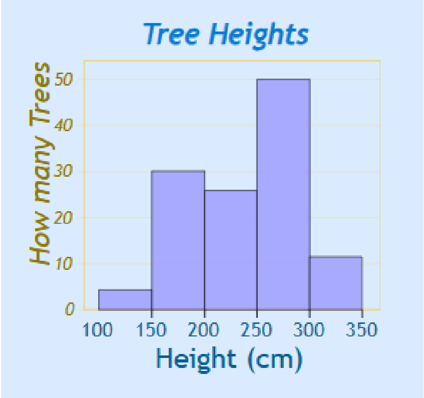
Uni-Variate Data Analysis (One Column)

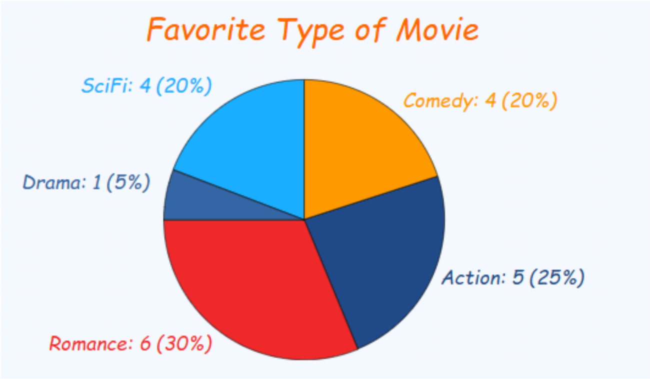


We can do lots of things with univariate data:

- Find a central value using mean, median and mode
- Find how spread out it is using range, quartiles and standard deviation
- Make plots like Bar Graphs, Pie Charts and Histograms







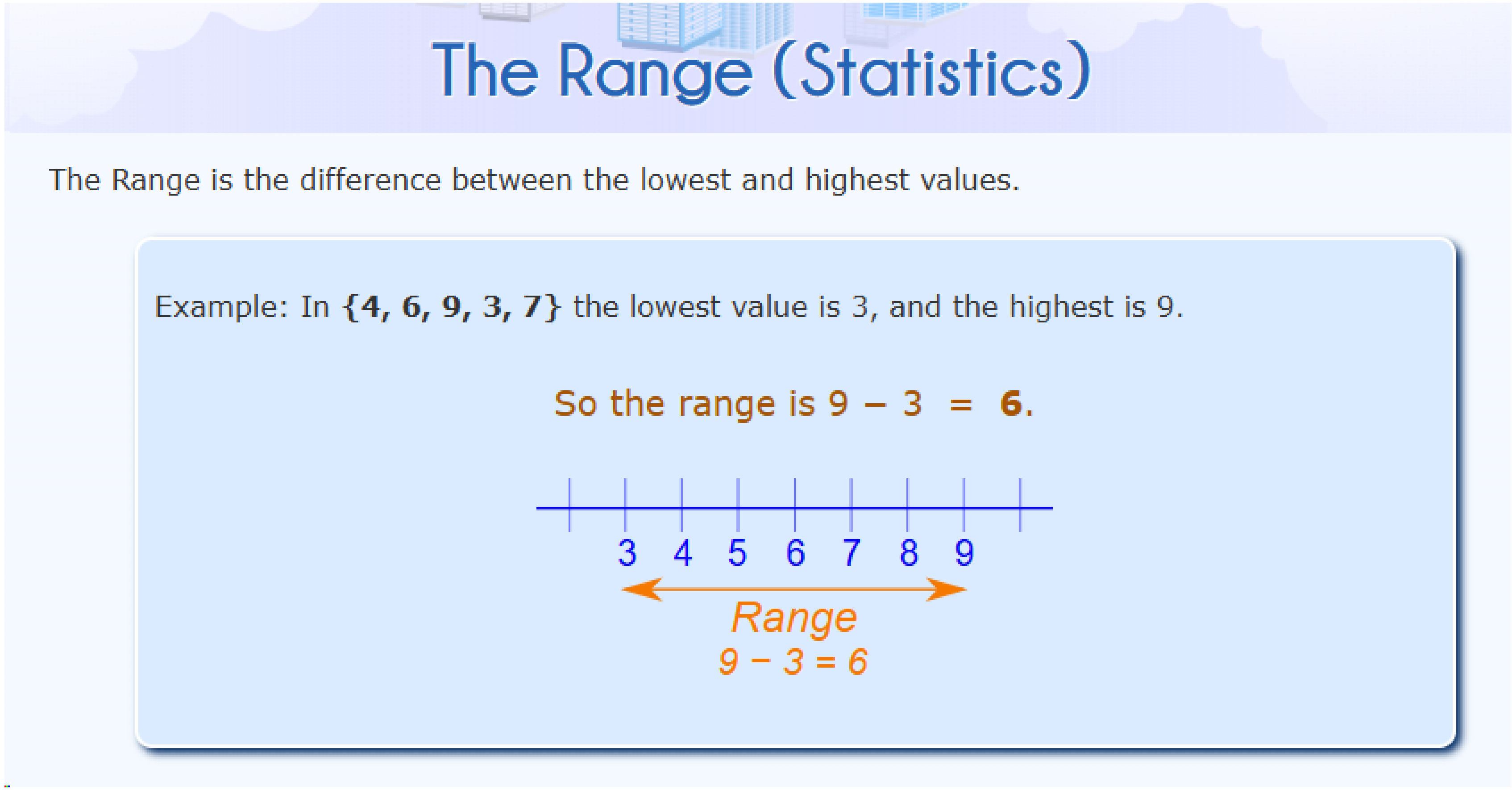
Import Statistics Library
&
Basic Central Tendency Measures

```python
import statistics

# Calculating mean, median, and mode
mean_value = statistics.mean(a)
median_value = statistics.median(a)
mode_value = statistics.mode(a)
std_dev = statistics.stdev(a)


# Displaying results
print("Mean:", mean_value)
print("Median:", median_value)
print("Mode:", mode_value)
print("Standard Deviation:", std_dev)
```

Measures of Dispersion

Uni-Variate Analytics

Range



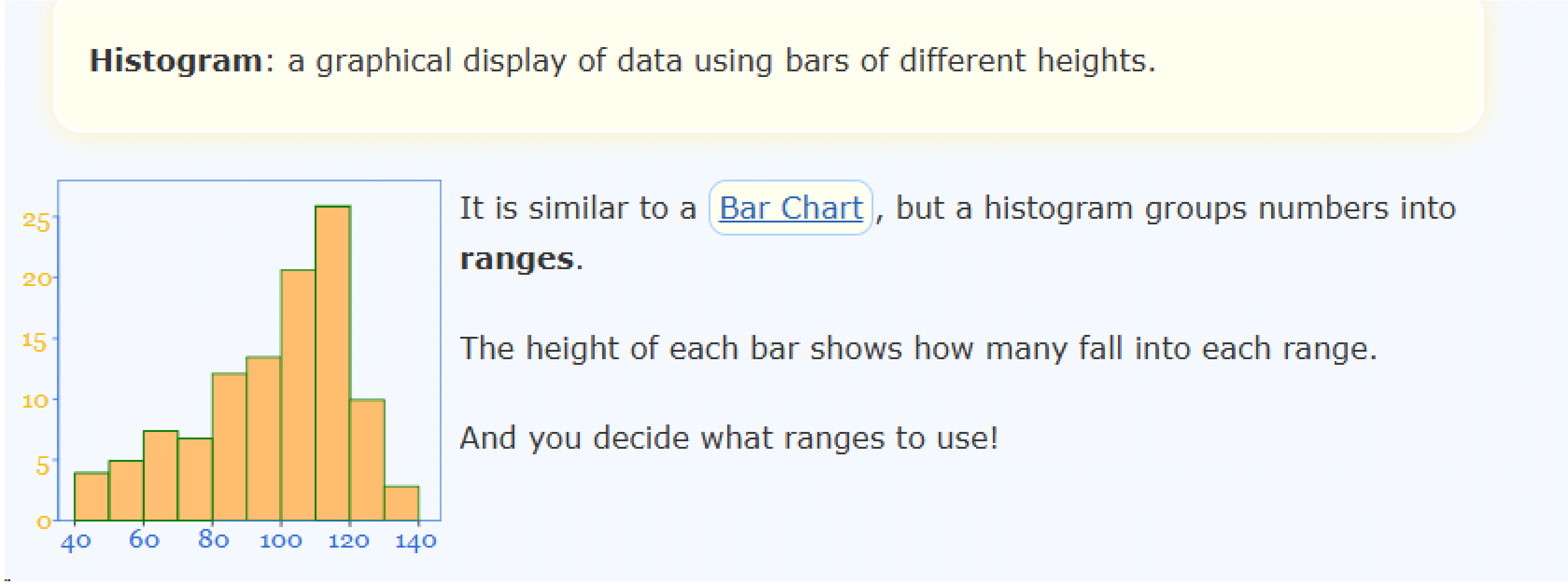
The Range (Statistics)

The Range is the difference between the lowest and highest values.

Example: In **{4, 6, 9, 3, 7}** the lowest value is 3, and the highest is 9.

So the range is 9 − 3 = **6**.

Histograms



Histogram: a graphical display of data using bars of different heights.

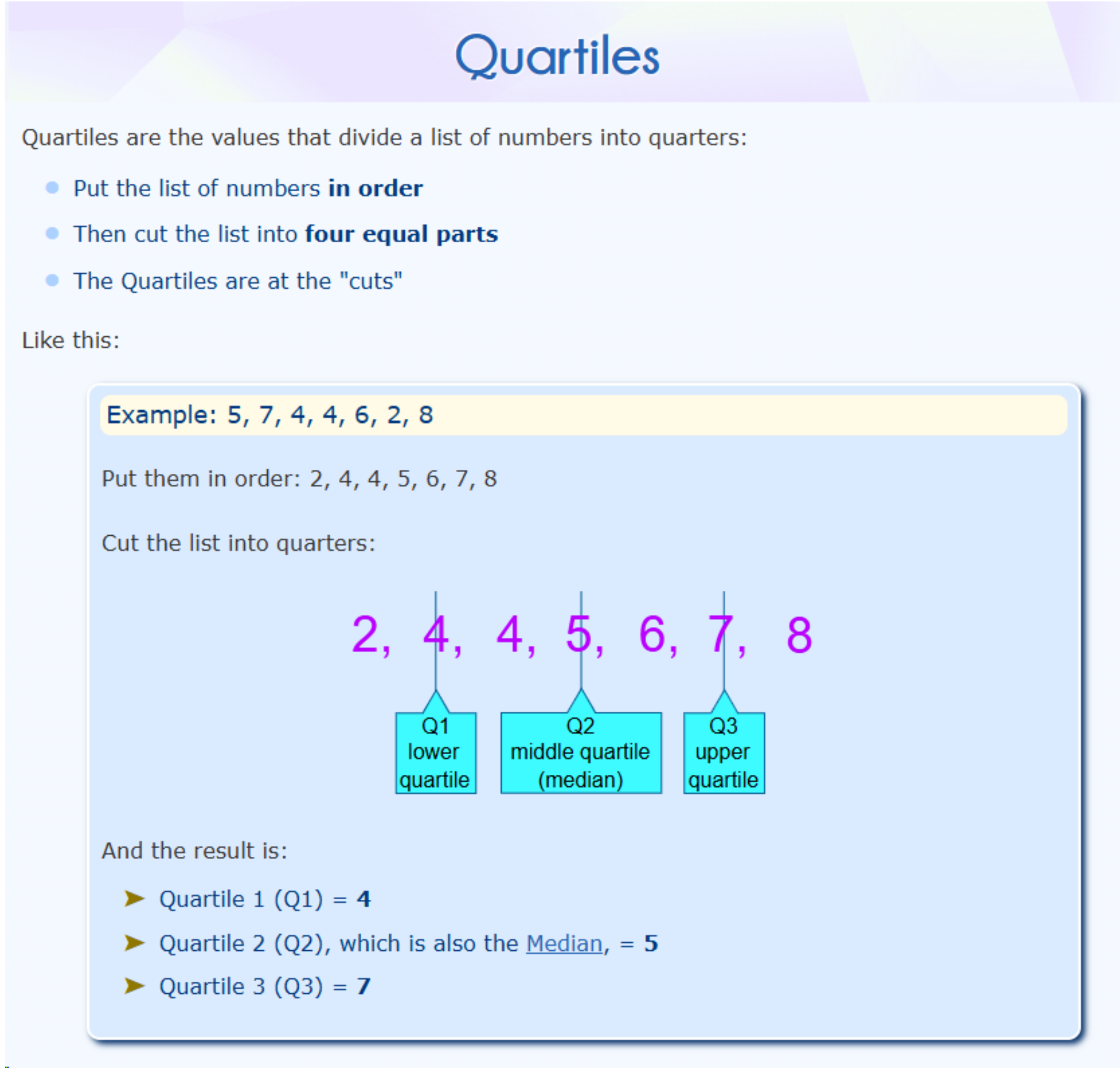
It is similar to a Bar Chart, but a histogram groups numbers into **ranges**.

The height of each bar shows how many fall into each range.

And you decide what ranges to use!

Ranges or Bins on X Axis and Values on Y Axis

Quartiles



Quartiles

Quartiles are the values that divide a list of numbers into quarters:

- Put the list of numbers **in order**
- Then cut the list into **four equal parts**
- The Quartiles are at the "cuts"

Like this:

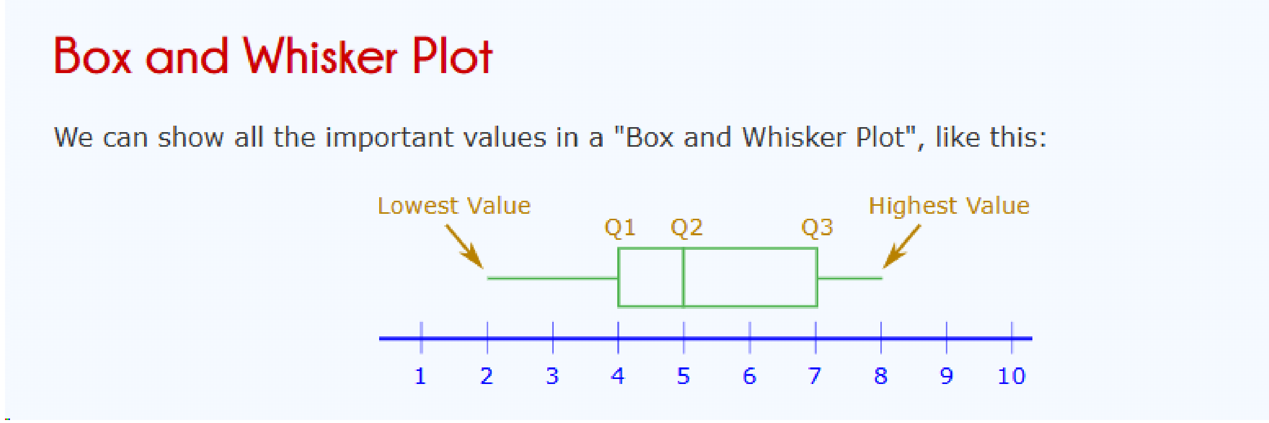
Example: 5, 7, 4, 4, 6, 2, 8

Put them in order: 2, 4, 4, 5, 6, 7, 8

Cut the list into quarters:

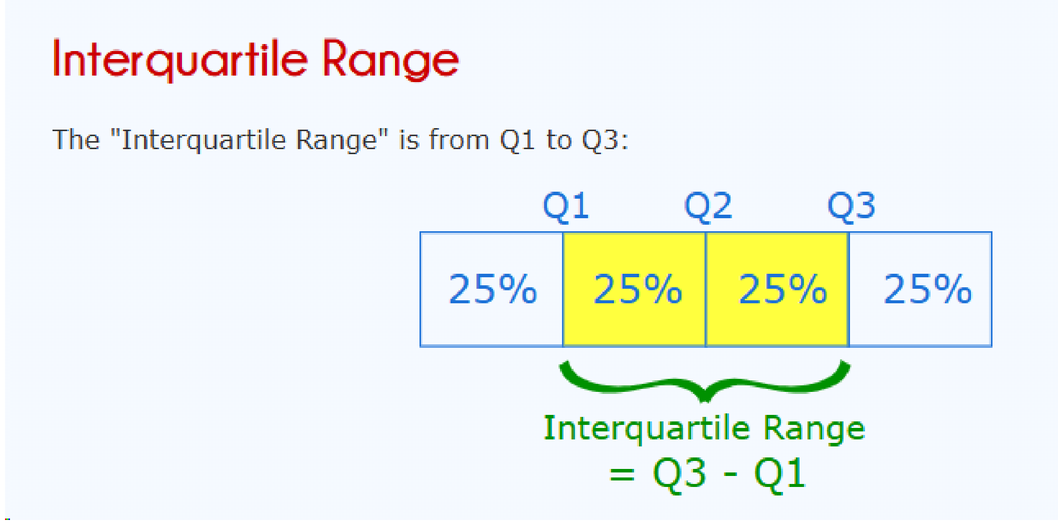
And the result is:

- Quartile 1 (Q1) = **4**
- Quartile 2 (Q2), which is also the Median, = **5**
- Quartile 3 (Q3) = **7**



Box and Whisker Plot

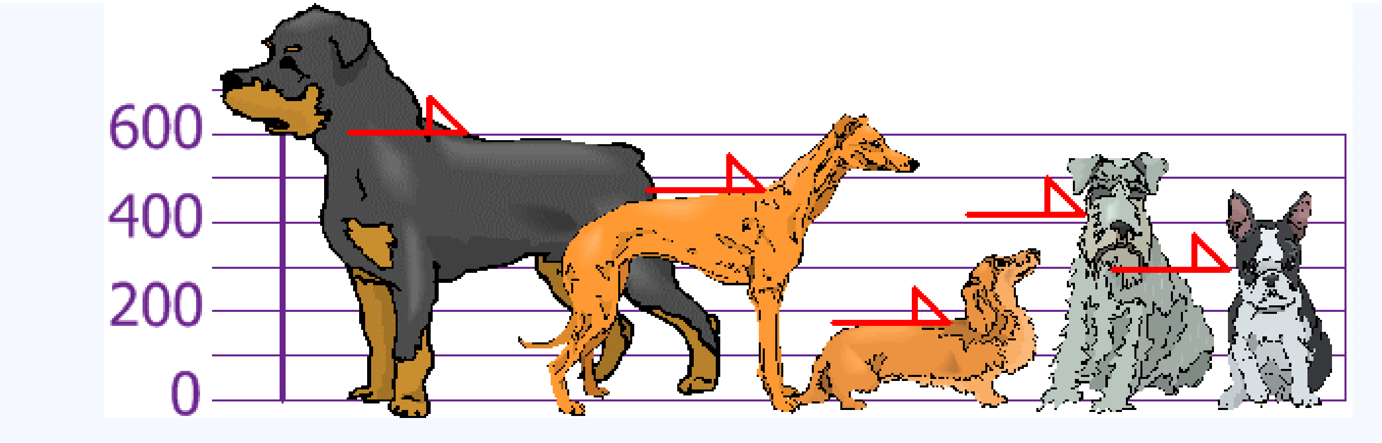
We can show all the important values in a "Box and Whisker Plot", like this:

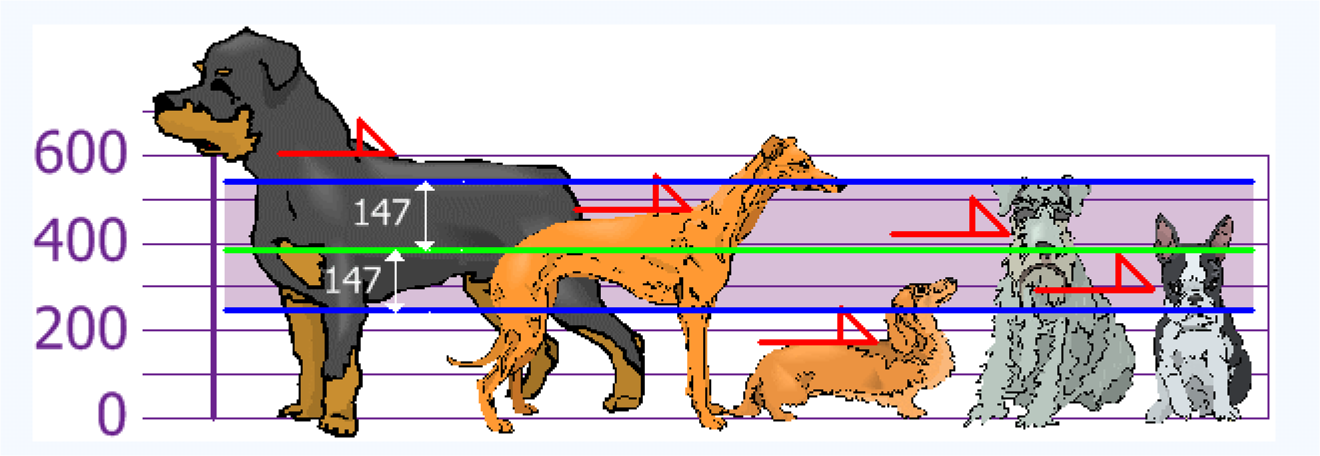


Interquartile Range

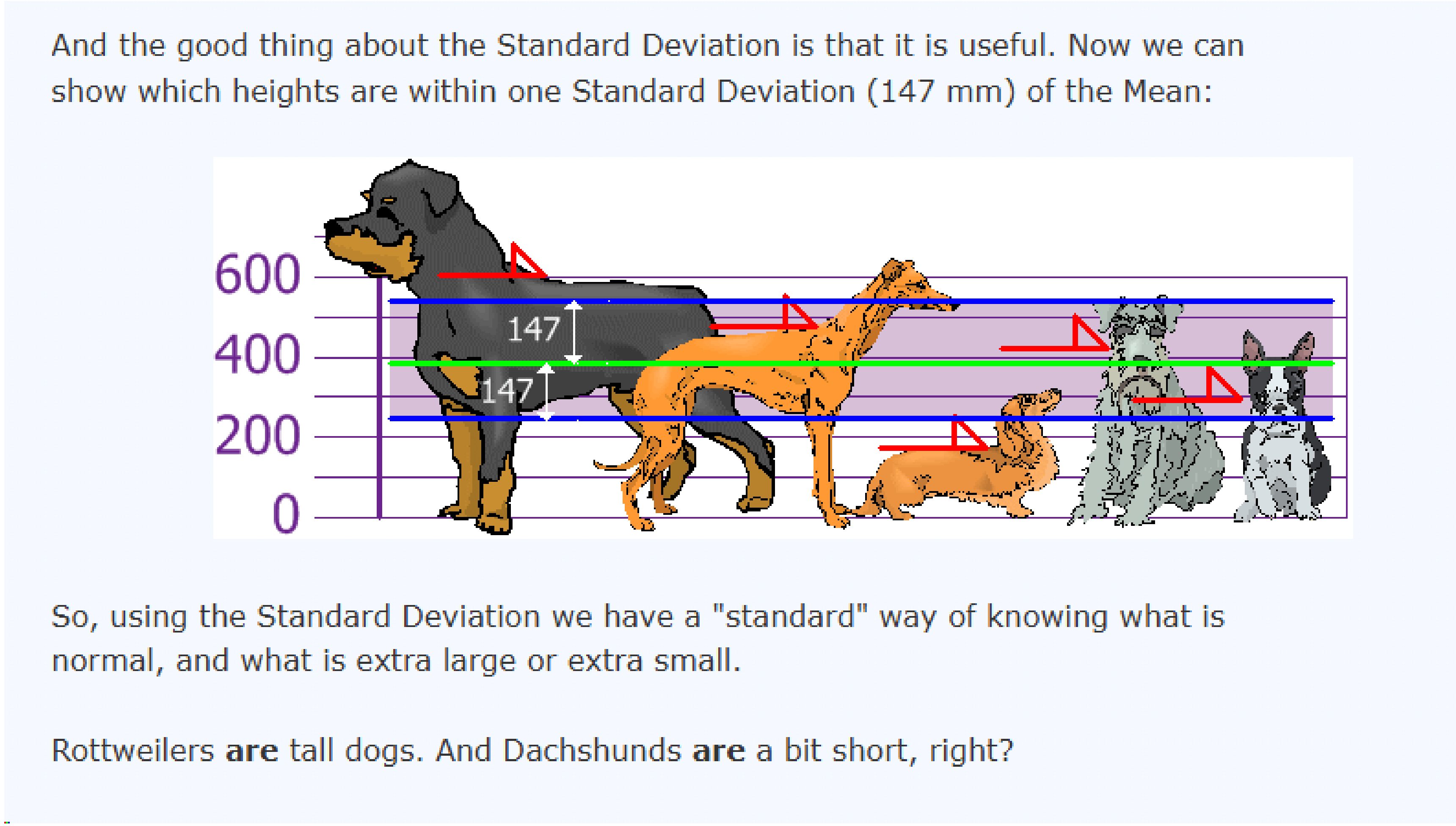
The "Interquartile Range" is from Q1 to Q3:

Measures of Variation





Standard Deviation



And the good thing about the Standard Deviation is that it is useful. Now we can show which heights are within one Standard Deviation (147 mm) of the Mean:

So, using the Standard Deviation we have a "standard" way of knowing what is normal, and what is extra large or extra small.

Rottweilers **are** tall dogs. And Dachshunds **are** a bit short, right?

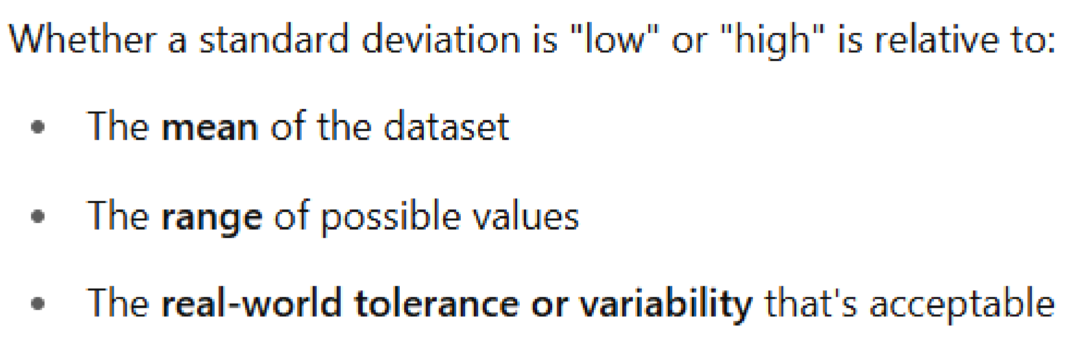
Code

```
import statistics
```

Assign data column to variable “a”

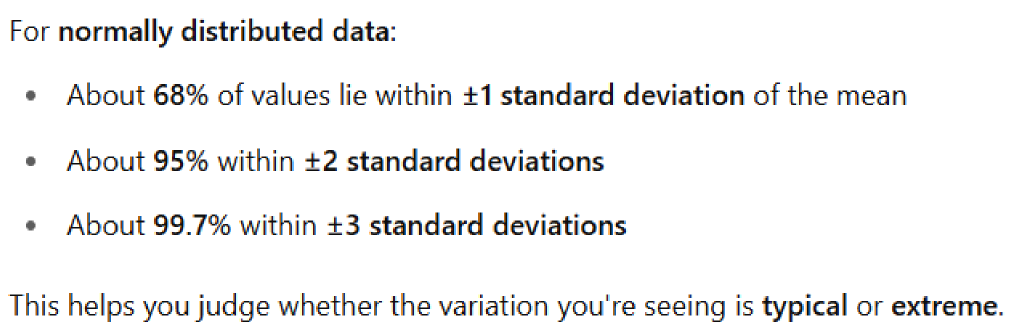
```
std_dev = statistics.stdev(a)

print("Standard Deviation:", std_dev)
```

Standard Deviation



Whether a standard deviation is "low" or "high" is relative to:

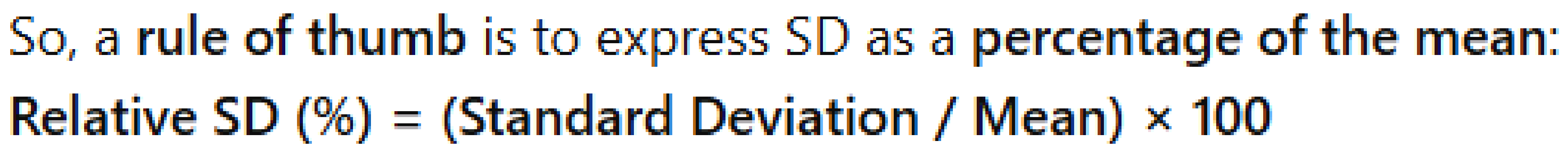
- The **mean** of the dataset
- The **range** of possible values
- The **real-world tolerance or variability** that's acceptable



For **normally distributed data**:

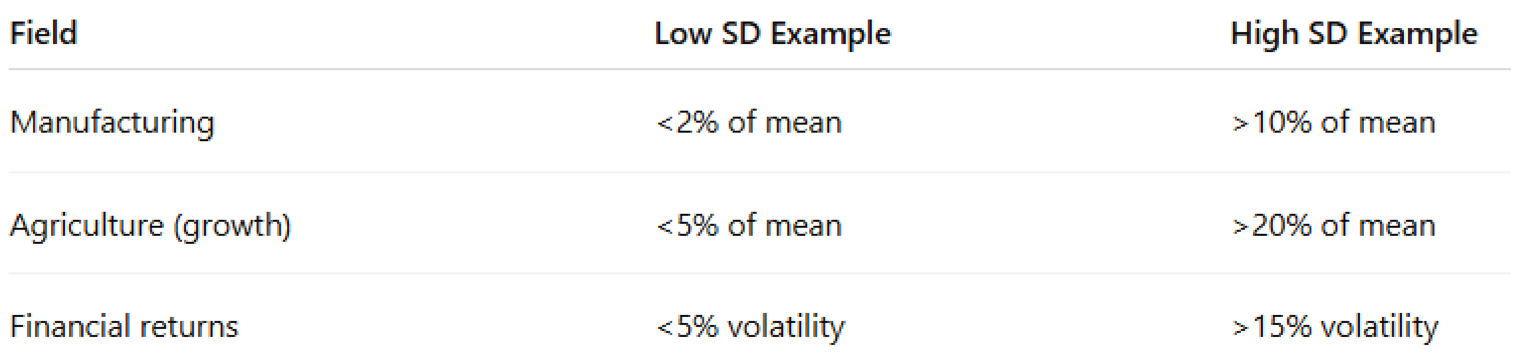
- About **68%** of values lie within **±1 standard deviation** of the mean
- About **95%** within **±2 standard deviations**
- About **99.7%** within **±3 standard deviations**

This helps you judge whether the variation you're seeing is **typical** or **extreme**.



So, a **rule of thumb** is to express SD as a **percentage of the mean:**

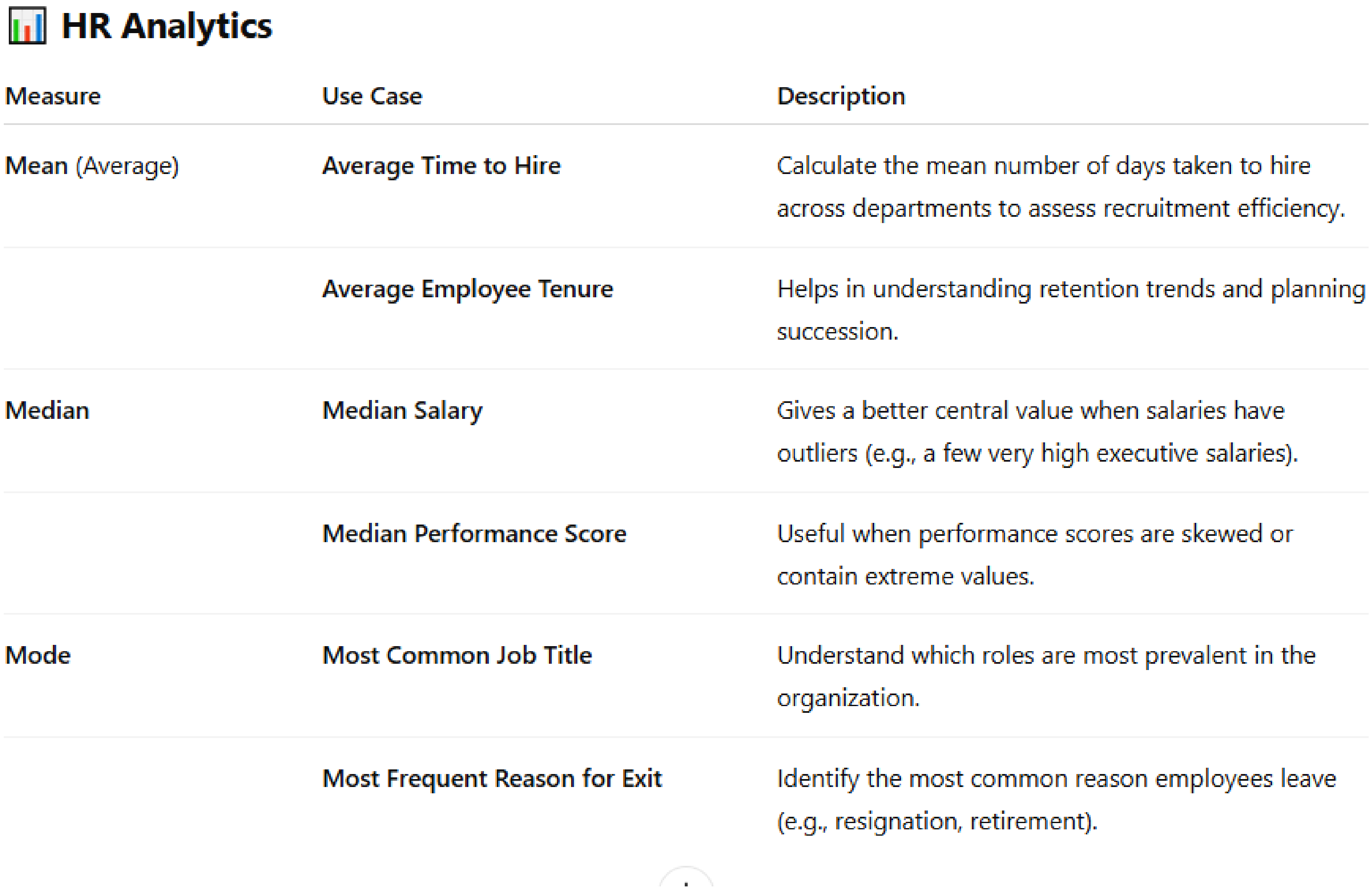
Relative SD (%) = (Standard Deviation / Mean) × 100



Field	Low SD Example	High SD Example
Manufacturing	<2% of mean	>10% of mean
Agriculture (growth)	<5% of mean	>20% of mean
Financial returns	<5% volatility	>15% volatility

So What ?

HR Analytics

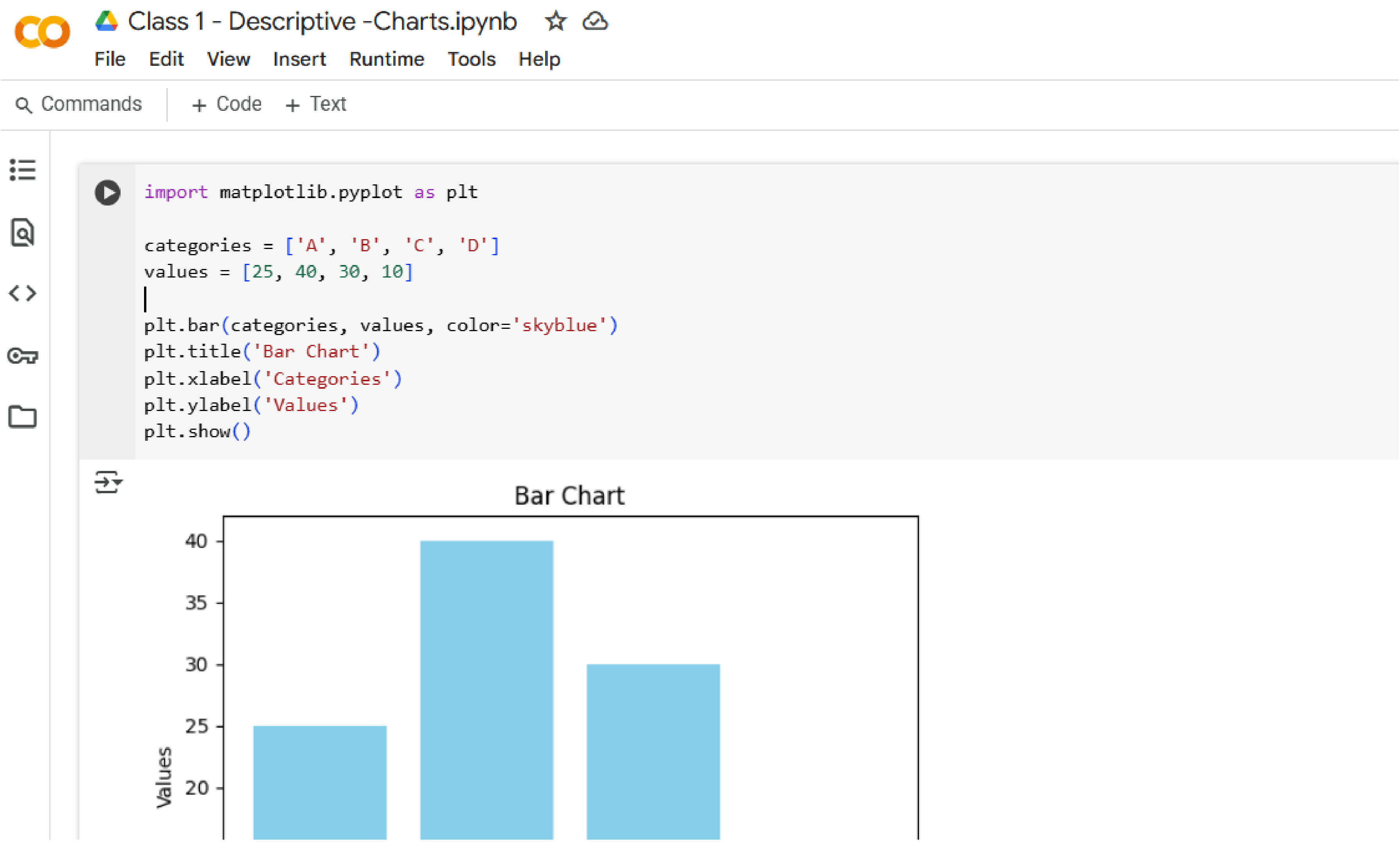


📊 HR Analytics

Measure	Use Case	Description
Mean (Average)	**Average Time to Hire**	Calculate the mean number of days taken to hire across departments to assess recruitment efficiency.
	Average Employee Tenure	Helps in understanding retention trends and planning succession.
Median	**Median Salary**	Gives a better central value when salaries have outliers (e.g., a few very high executive salaries).
	Median Performance Score	Useful when performance scores are skewed or contain extreme values.
Mode	**Most Common Job Title**	Understand which roles are most prevalent in the organization.
	Most Frequent Reason for Exit	Identify the most common reason employees leave (e.g., resignation, retirement).

Charts & Dashboards

Code – Check the Jupyter Notebook



```python
import matplotlib.pyplot as plt

categories = ['A', 'B', 'C', 'D']
values = [25, 40, 30, 10]

plt.bar(categories, values, color='skyblue')
plt.title('Bar Chart')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.show()
```