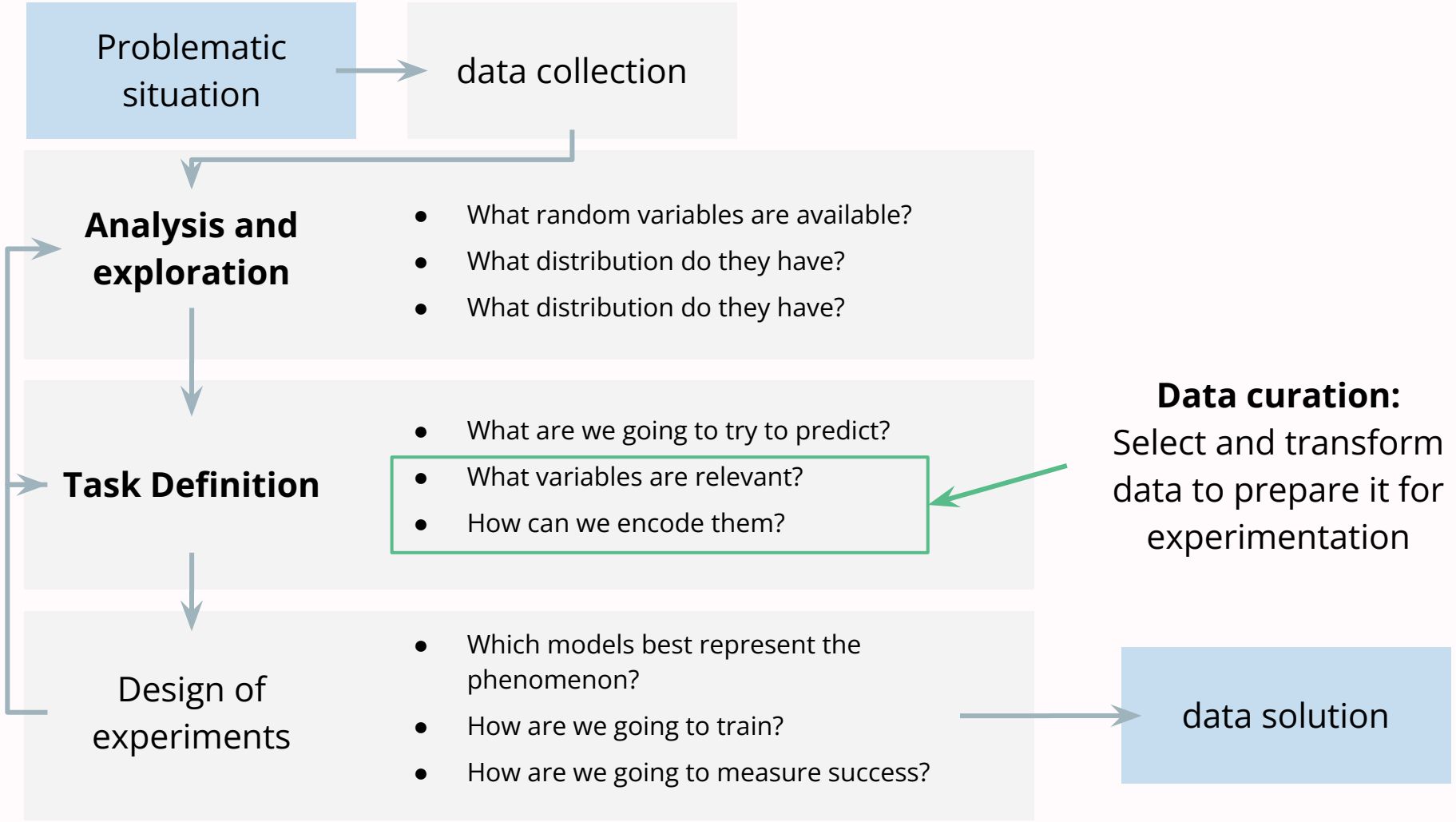
Analyse et manipulation de données

DigitalLab@LaPlataforme_

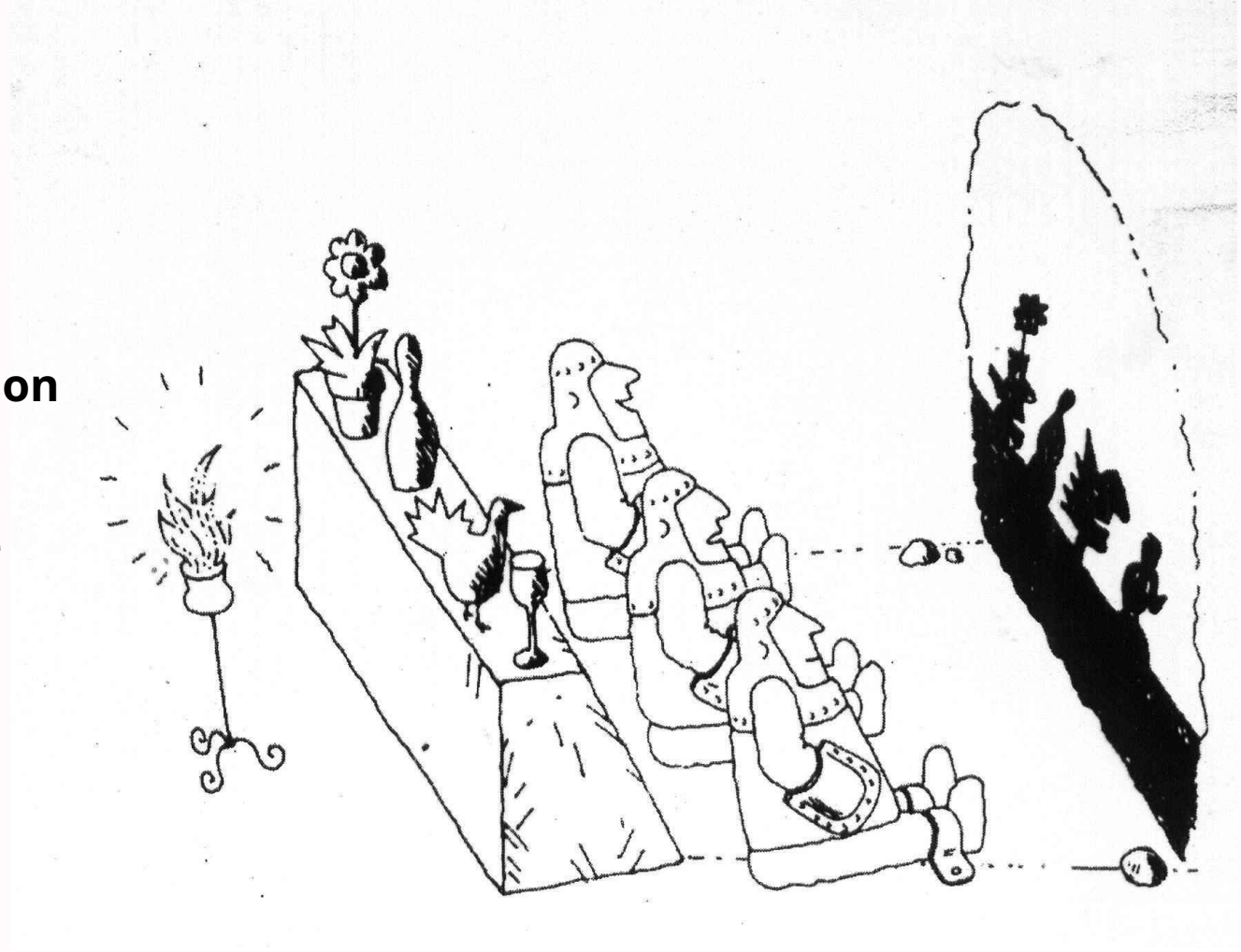
What is it about?


Problematic situation
data collection
Analysis and exploration
What random variables are available?
What distribution do they have?
What distribution do they have?
Task Definition
What are we going to try to predict?
What variables are relevant?
How can we encode them?
Design of experiments
Which models best represent the phenomenon?
How are we going to train?
How are we going to measure success?
data solution
Data curation:
Select and transform data to prepare it for experimentation

We want to bring out the important features for a given task

Data science is like Plato's cave allegory

The data is a **projection** that shows us only certain aspects of the phenomenon we are studying.



Data Curation

Conceptual aspects

- Outlier Treatment
- Bias Detection
- Value Imputation

Practical Aspects

- Reading and Cleaning
- Aggregation and Transformation
- Reproducibility
- Partitioning and Sampling

Data Exploration

- To decide on curation processes, we have to understand our data as a whole. It includes:
  - All the analytics tools we've seen.
  - More complex techniques for data analysis that allow multiple variables to be related.
  - Unstructured data visualization techniques

Problematic situation	Data	Curation decisions
Predict programmers salaries in Argentina in 2020	Voluntary survey with age, gender, years of experience and salary columns	• Delete ages less than 18 and greater than 99 • Eliminate salaries greater than 1 million pesos • Standardize the years of experience so that the mean is 0. • Rescale the ages in a range from 1 to 0, such that 18 years or less corresponds to 0 and 70 years or more corresponds to 1. • Delete the gender column.

Problematic situation	Data	Curation decisions
Predict programmers salaries in Argentina in 2020	Voluntary survey with age, gender, years of experience and salary columns	• Delete ages less than 18 and greater than 99 • Eliminate salaries greater than 1 million pesos • Standardize the years of experience so that the mean is 0. • Rescale the ages in a range from 1 to 0, such that 18 years or less corresponds to 0 and 70 years or more corresponds to 1. • Delete the gender column.
Predict the price of a property	Government database with records of real estate transactions. It has price, date and location.	• Delete day and month of the transaction. • *Scrape* buying/selling sites to extract additional information about each property. • Impute missing values using estimates based on similar examples.

Tradeoff: using domain knowledge vs limiting our modeling too much

The curse of the categories

What **information** does the address of a property give me?

The address of a property for sale is a categorical variable that cannot be used without transforming it. Intuitively, we infer the neighborhood of a property based on its address, and based on that we estimate the value.

- The categories give me information because they group different examples. The fewer examples they group together, the less informative they are.

The curse of the categories

- Delete the variable.
- Combine it with another variable.
  - Ex: We only use the zipcode for neighborhoods that have more than one postal code, or to differentiate homonymous localities.
- Create new categories:
  - Group similar categories.
  - Create an “other” category for categories that don't have many examples.

Demo notebook

01_exploration.ipynb

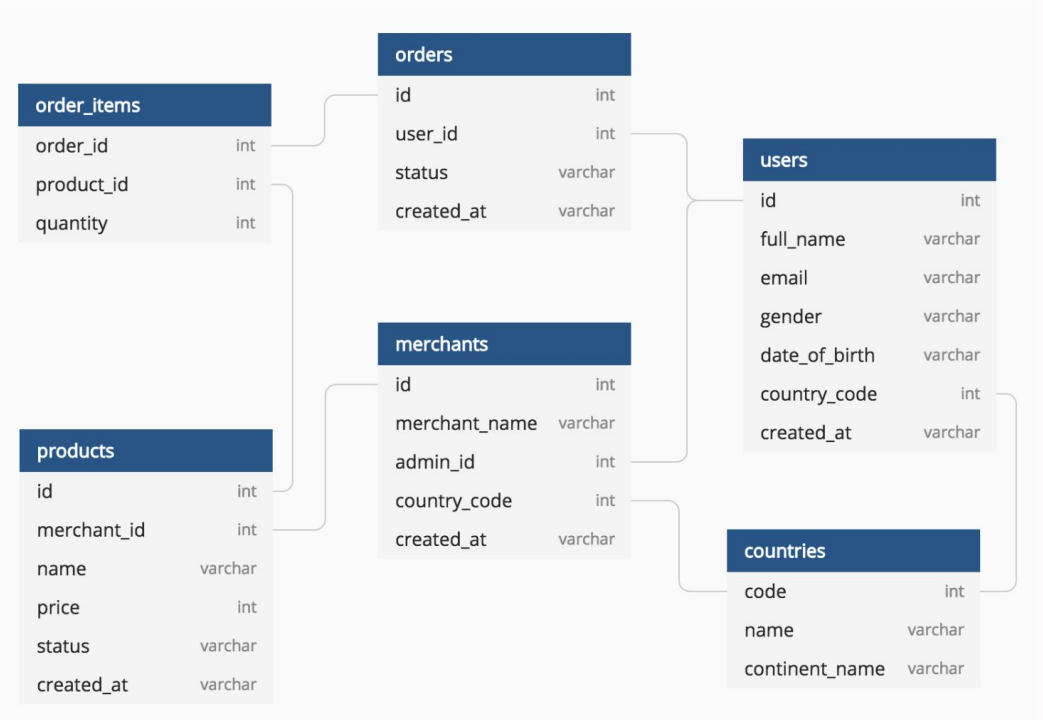
Data types

Data Structure

- We call a set of records “data”.
- Each record has a set of features associated with it, and the features can be related in complex ways.
- Different structures are often stored with particular file formats
    - The structure of the data is not the same as the type of database or files in which it is stored

Structured Data

- All records have the same characteristics with the same type
- Characteristics of some records may be records in another table



- Files in CSV format, parquet, etc.
- Relational databases like MySQL, Postgres

https://www.holistics.io/blog/top-5-free-database-diagram-design-tools/

Semi-structured data

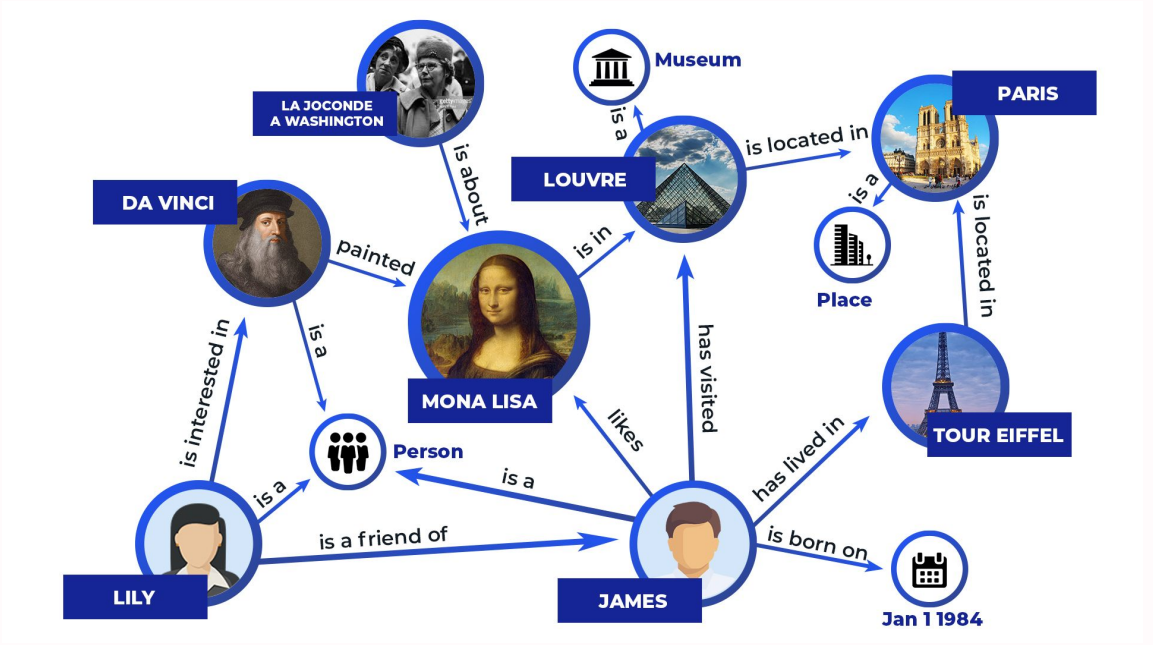
- Each record has a different set of characteristics
- Records can be nested


```
{"orders": [
   {
     "client_id": 1458,
     "items": [
       {"description": "Empanadas", "amount": 12},
       {"description": "Hot sauce", "amount": 1}
     ],
     "total": 950,
     "payment_method": "cash"
   },
   {
     "client_id": 985,
     "items": [
       {"description": "Full sandwich", "amount": 2,
        "observations": "One without egg"}
     ],
     "total": 1400,
     "payment_method": "debit",
     "debit_card": "Mastercard"
   }
 ]}
```


- Files in JSON format
- Non-relational databases like MongoDB

Semi-structured data

- Records can have complex relationships
  - Hierarchies
  - Graph Structure (Twitter)



- Triple RDF
- Graph-oriented databases

https://www.komunicando.es/wp-content/uploads/2017/12/knowledge-graph.jpg

Unstructured Data

- Collections of different types:
  - Text documents
  - Images
  - Audio
- May or may not have associated metadata

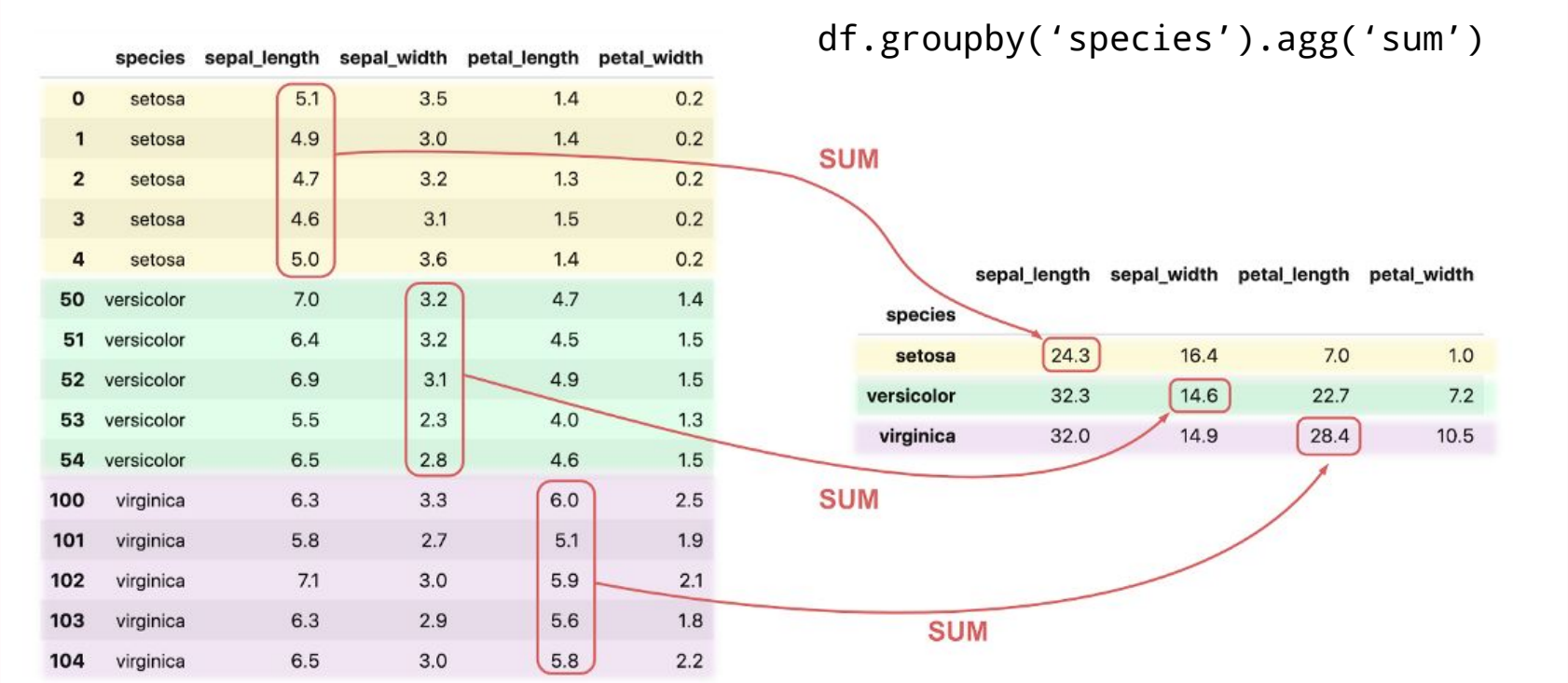
Data Enrichment

Combining different datasets

Grouping and aggregation

1. groupby:
   - Takes a series of columns A, B, C
   - For each combination of column values (a1, b1, c1), group the rows that have those values.
2. agg:
   - Takes a function f
   - For each group of rows, apply the function f to each column.

Grouping and aggregation



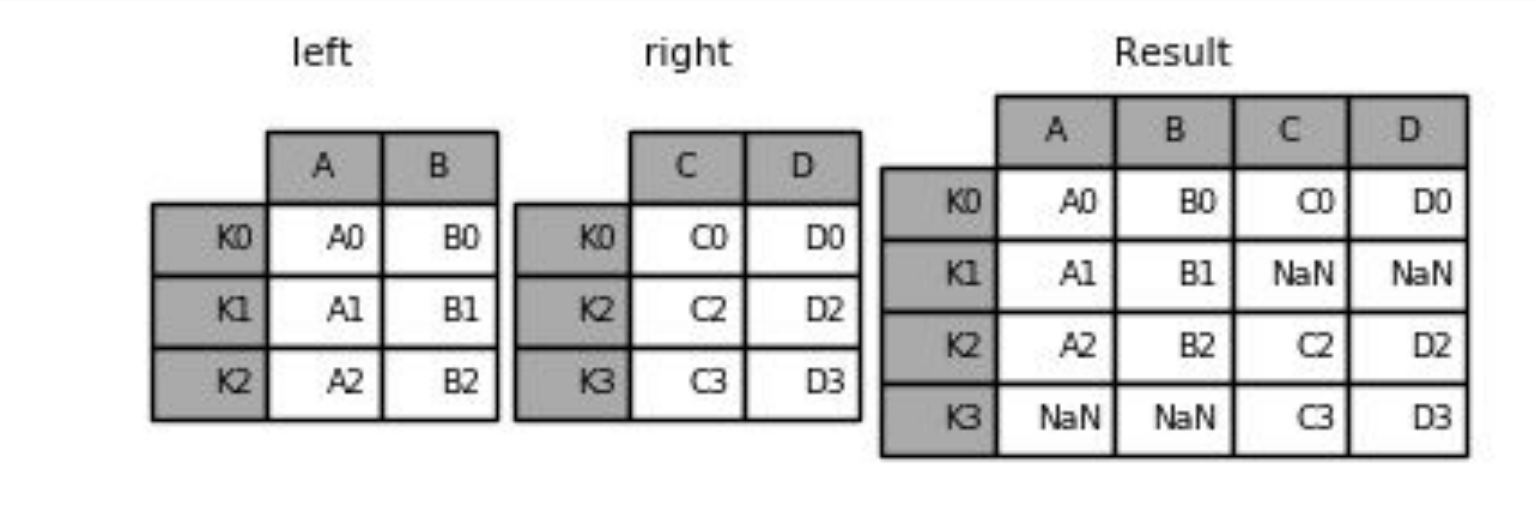
	species	sepal_length	sepal_width	petal_length	petal_width
0	setosa	5.1	3.5	1.4	0.2
1	setosa	4.9	3.0	1.4	0.2
2	setosa	4.7	3.2	1.3	0.2
3	setosa	4.6	3.1	1.5	0.2
4	setosa	5.0	3.6	1.4	0.2
50	versicolor	7.0	3.2	4.7	1.4
51	versicolor	6.4	3.2	4.5	1.5
52	versicolor	6.9	3.1	4.9	1.5
53	versicolor	5.5	2.3	4.0	1.3
54	versicolor	6.5	2.8	4.6	1.5
100	virginica	6.3	3.3	6.0	2.5
101	virginica	5.8	2.7	5.1	1.9
102	virginica	7.1	3.0	5.9	2.1
103	virginica	6.3	2.9	5.6	1.8
104	virginica	6.5	3.0	5.8	2.2

species	sepal_length	sepal_width	petal_length	petal_width
setosa	24.3	16.4	7.0	1.0
versicolor	32.3	14.6	22.7	7.2
virginica	32.0	14.9	28.4	10.5

https://towardsdatascience.com/how-to-use-the-split-apply-combine-strategy-in-pandas-groupby-29e0eb44b62e

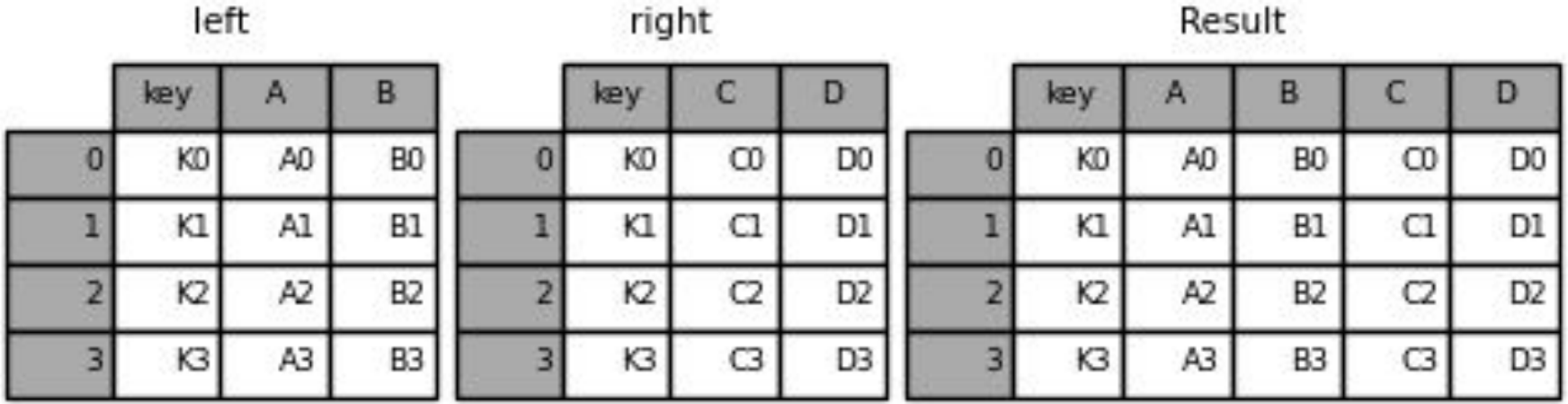
Join and merge

1. `df1.join(df2, how='outer')`
   - Horizontally join the DataFrames and match the rows where the index value is the same



Join and merge

1. `df1.merge(df2, on='key')`
   - Same as join, but instead of comparing indexes, it compares a set of columns.



left

	key	A	B
0	K0	A0	B0
1	K1	A1	B1
2	K2	A2	B2
3	K3	A3	B3

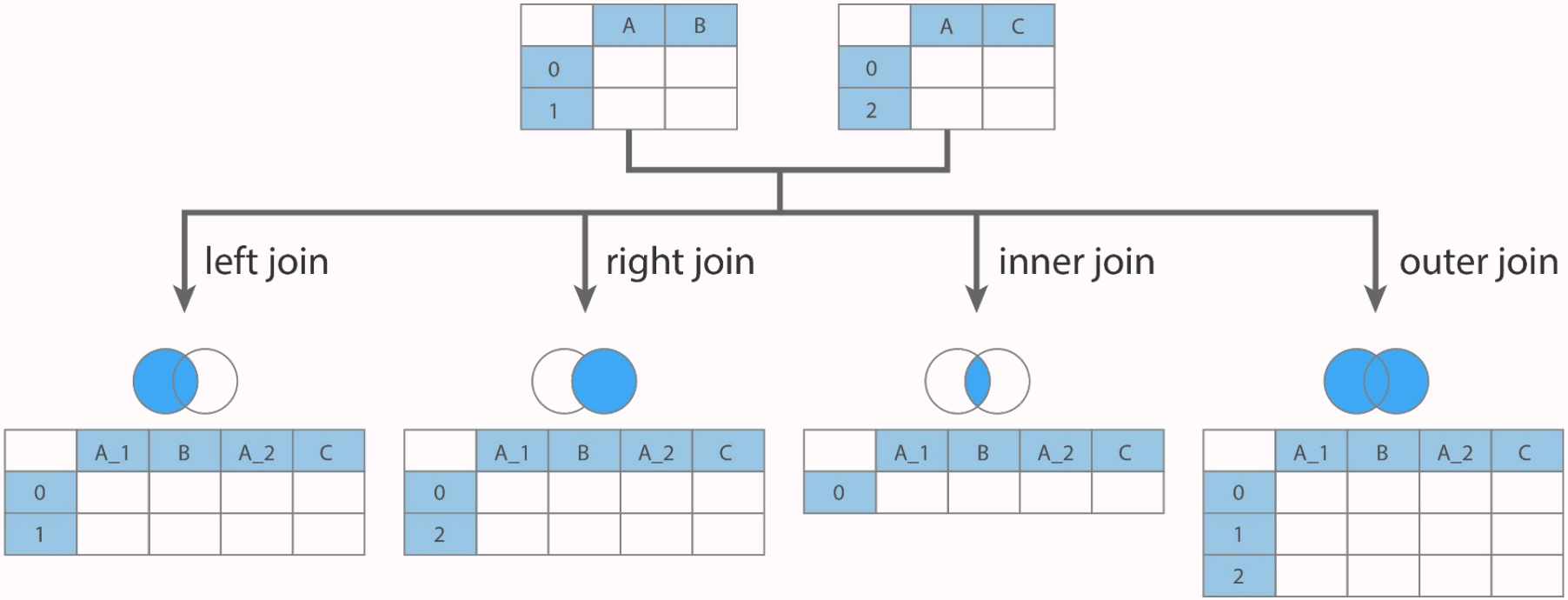
right

	key	C	D
0	K0	C0	D0
1	K1	C1	D1
2	K2	C2	D2
3	K3	C3	D3

Result

	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K2	A2	B2	C2	D2
3	K3	A3	B3	C3	D3

Join and merge



https://towardsdatascience.com/python-pandas-dataframe-join-merge-and-concatenate-84985c29ef78

Unexpected duplicates!

df1

Product	Sales
R22	45
J14	10
R5	58
P17	24

df2

Product	Category
R22	T-shirt
J14	Jean
J14	Trousers
R5	T-shirt
P17	Trousers

```
all_sales = df1.merge(
    df2, on='Product')
```

Product	Category	Sales
R22	T-shirt	45
J14	Jean	10
J14	Trousers	10
R5	T-shirt	58
P17	Trousers	24

```
cat_sales = all_sales\
.groupby(Category).sum()
```

Category	Sales
T-shirt	103
Jean	10
Trousers	34

```
total_sales =
    all_sales.Sales.sum()
```

Demo notebook

02_combining_datasets.ipynb