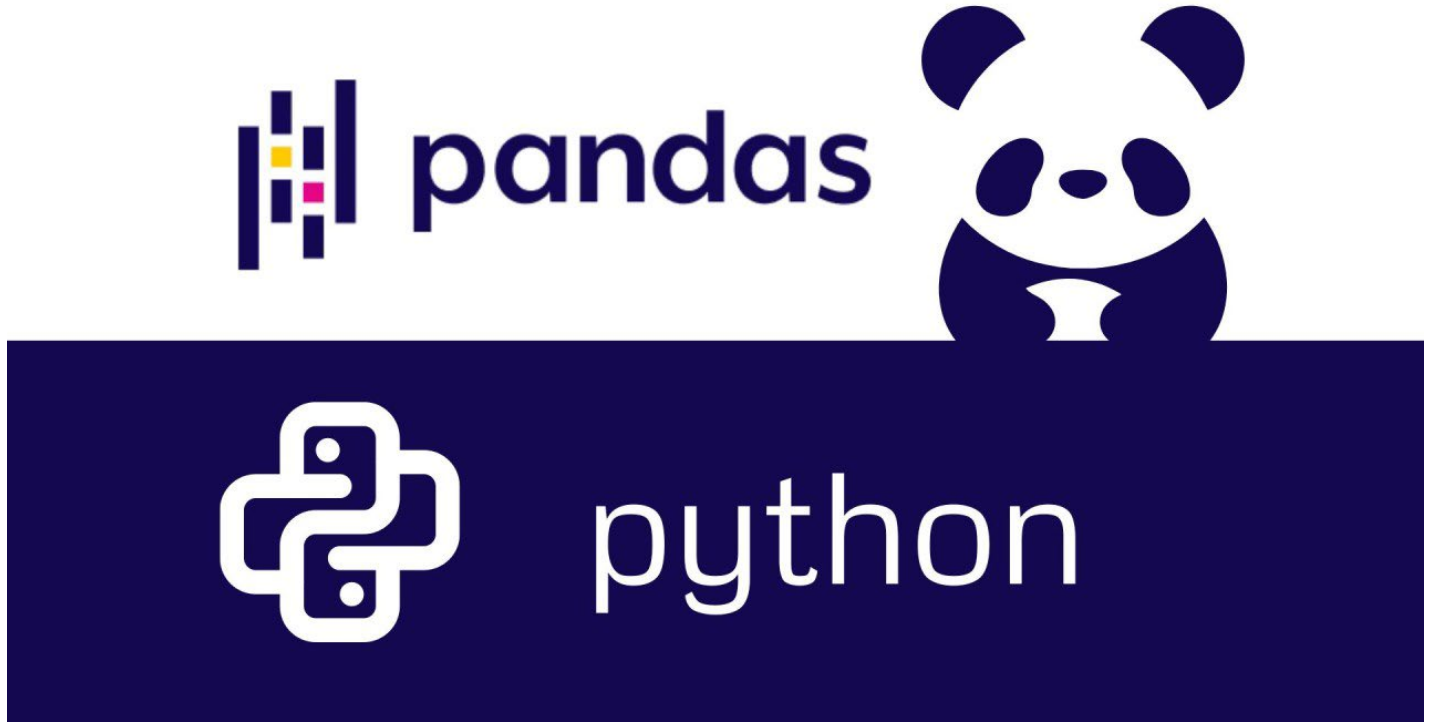
DATA SCIENCE

Introduction to Pandas



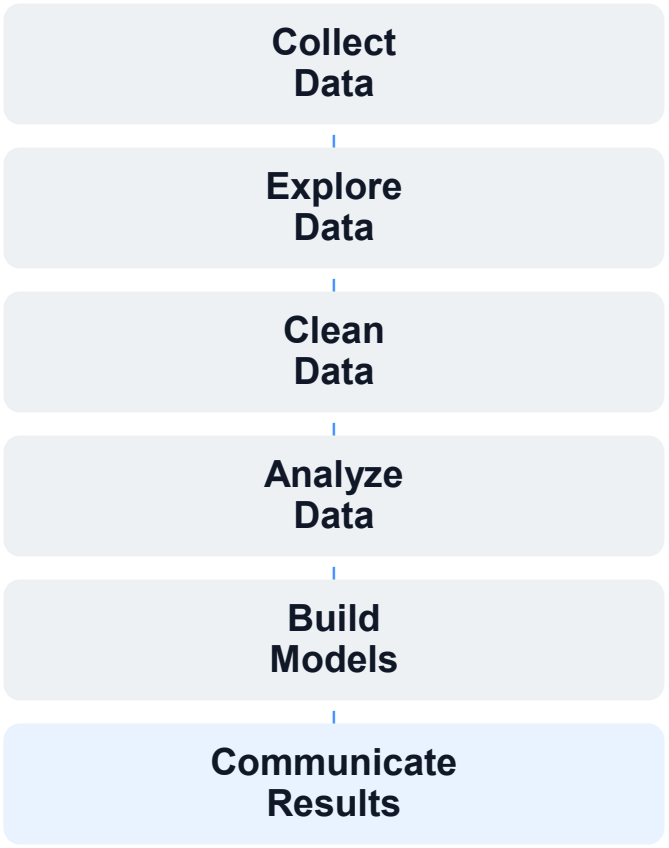
Data Cleaning and Manipulation

UNC GREENSBORO

MOTIVATION

What Does a Data Scientist Actually Do?

A typical workflow starts long before a model is trained.



Before building models, we need clean and organized data.

Data Cleaning is a core data-science task.

MOTIVATION

The Reality of Real-World Data

Look closely: a small table can contain several different problems.

Name	Age	Score	Date
Alice	20	95	2026-01-01
Bob	twenty-one	88	Jan 2, 2026
Charlie		92	01/03/26
David	21	105	2026/01/04

What problems do you see?

MOTIVATION

Why Do We Clean Data?

Cleaning makes information trustworthy enough to use.

- Remove incorrect values
- Fill missing information
- Standardize formats
- Detect unusual observations

Student	Exam Score
Alice	85
Bob	90
Cindy	88
David	92
Eric	900

Employee	Salary
Alice	$60,000
Bob	$65,000
Cindy	$70,000
David	$500,000

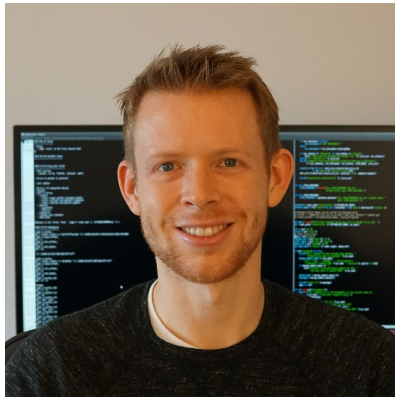
Employee	Salary
Alice	$60,000
Bob	$65,000
Cindy	$70,000
David	$5,000,000,000

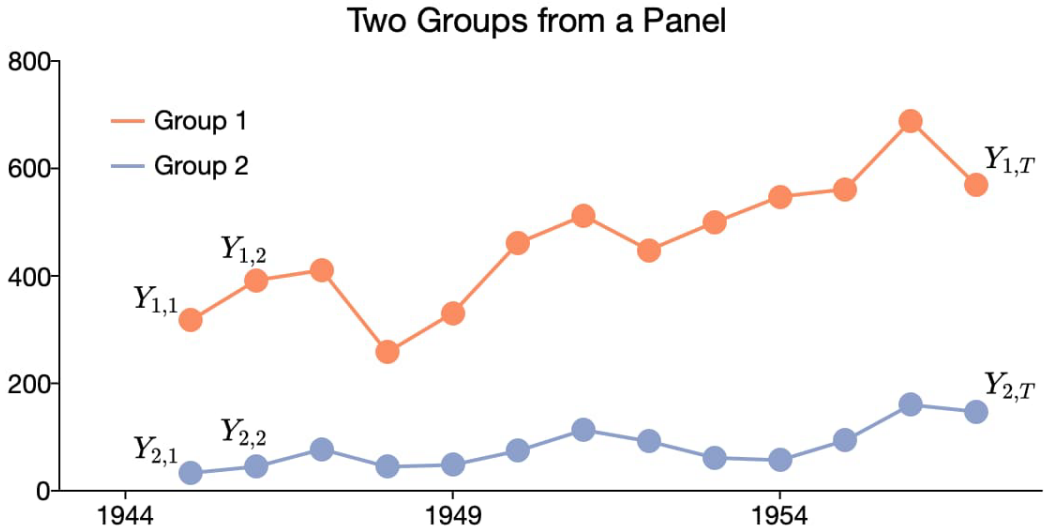
MOTIVATION

Pandas: A Tool for Data Scientists

A Python library designed for data manipulation and analysis.

- Created by Wes McKinney in 2008, now maintained by many others.
- Powerful and productive Python data analysis and Management Library
- Panel Data System
  - The name is derived from the term "panel data", an econometrics term for datasets that include both time-series and cross-sectional data
- Its an open-source product.





TABULAR DATA

Most Data Looks Like Tables

Rows describe observations; columns describe variables.

Student	Major	GPA	Year
Amy	CS	3.8	Junior
Bob	Math	3.5	Senior
Cindy	Biology	3.9	Junior

Term	Meaning
Row	One data record
Column	One attribute
Cell	One value

Rows → observations / records

Columns → variables / features

TABULAR DATA

Columns Have Different Data Types

Computers need to know what type of information each column contains.

Column	Type	Example
Name	Text	Amy
Age	Integer	20
GPA	Float	3.8
Birthday	Date	2026-01-01
Major	Category	CS

Correct types make sorting, calculations, and comparisons possible.

TABULAR DATA

Sources of Tabular Data

Pandas can read tables from many common places.

- Excel spreadsheets
- CSV files
- Databases
- Sensors
- Surveys

DIRTY DATA

Data Is Not Always Ready for Analysis

Dirty data contains errors, missing information, or inconsistent representations.

1
Outliers

2
Missing values

3
Inconsistent formats

4
Duplicate records

> “ Before asking questions of the data, ask whether the data is believable.

DIRTY DATA

Outliers

An outlier is a value that is unusually different from others.

```
Student exam
scores

85
90
92
88
95
900
```

WITHOUT THE OUTLIER

```
(85 + 90 + 92 + 88 + 95) / 5
```

= 90

WITH THE OUTLIER

```
(85 + 90 + 92 + 88 + 95 + 900)
/ 6
```

= 225

The average no longer represents a typical score.

Missing Values

Missing information can appear in many different forms.

Name	Age	GPA
Amy	20	3.8
Bob		3.5
Cindy	21	

COMMON REPRESENTATIONS

empty **NA** **N/A** **null** **?**

These may all mean the same thing: we do not have a value.

DIRTY DATA

How Should We Handle Missing Data?

There is no single correct choice—it depends on the situation.

1
Remove rows

```
Delete incomplete records
```

2
Fill values

```
Age = average age
```

3
Keep missing information

```
Missing may be meaningful
```

DIRTY DATA

Inconsistent Formats

The same idea can be represented in different ways.

DATE COLUMN

```
2026-01-01
01/02/2026
Jan 3, 2026
```

SPELLING DIFFERENCES

```
Computer Science
Computer science
CS
```

Other Data Quality Problems

Small inconsistencies can make grouping and analysis unreliable.

DUPLICATE DATA

```
Alice Smith
Alice Smith
```

WRONG DATA TYPES

```
Age:
20
"twenty-one"
```

WHY PANDAS?

Python Has Built-in Data Structures

Can we use Python lists and dictionaries?

Yes.

But they are not designed for data analysis.

Lists
&
Dictionaries

Using a Python List

Lists can store data—but column operations become awkward.

```
students = [
    ["Amy", 20, 3.8],
    ["Bob", 21, 3.5],
    ["Cindy", 22, 3.9]
]
```

HOW DO WE GET THE GPA COLUMN?

```
students[0][2]
students[1][2]
students[2][2]
```

Hard to manage.

Using a Dictionary

A dictionary gives columns names, but still lacks analysis tools.

```
students = {
  "name": ["Amy","Bob","Cindy"],
  "age": [20,21,22],
  "gpa": [3.8,3.5,3.9]
}
```

BETTER

```
students["gpa"]
```

- Need to manually manage structure
- No built-in data operations

WHY PANDAS?

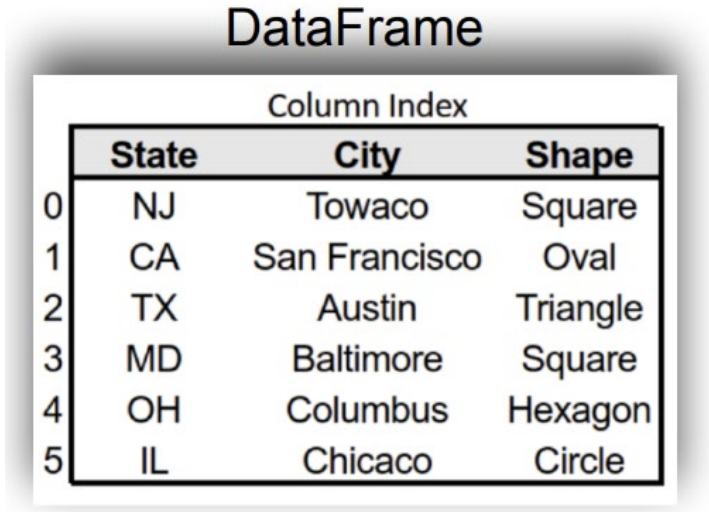
Pandas Solution

The same data becomes a labeled, analyzable table.

```
import pandas as pd

df = pd.DataFrame({
  "name": ["Amy","Bob","Cindy"],
  "age": [20,21,22],
  "gpa": [3.8,3.5,3.9]
})
```

	name	age	gpa
0	Amy	20	3.8
1	Bob	21	3.5
2	Cindy	22	3.9



Example: Filtering Data

Find students with GPA greater than 3.7.

```
df[df["gpa"] > 3.7]
```

→

Name	GPA
Amy	3.8
Cindy	3.9

> Filter rows by asking a true-or-false question about each row.

Example: Selecting Data

Selecting one column is a direct operation.

PANDAS

```
df["gpa"]
```

A DataFrame column is a Series.

RESULT

```
0    3.8
1    3.5
2    3.9
```

Series

	Value
0	NJ
1	CA
2	TX
3	MD
4	OH
5	IL

What Is a Series?

A Series is a one-dimensional labeled array.

```
gpa = pd.Series([3.8, 3.5, 3.9])
```

```
0    3.8
1    3.5
2    3.9
```

> **Think: one column of data plus labels.**

Series Example

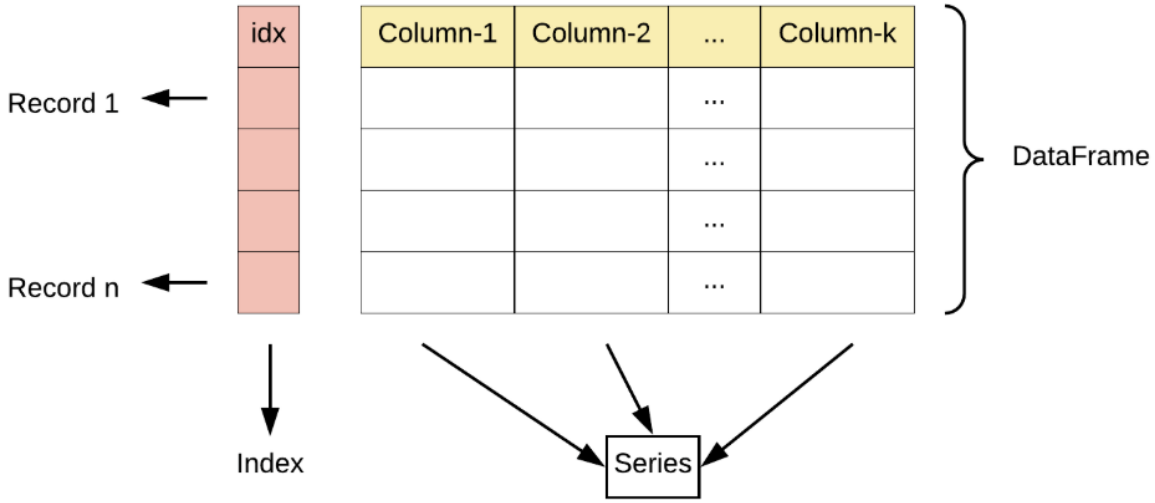
Indexes do not have to be numbers.

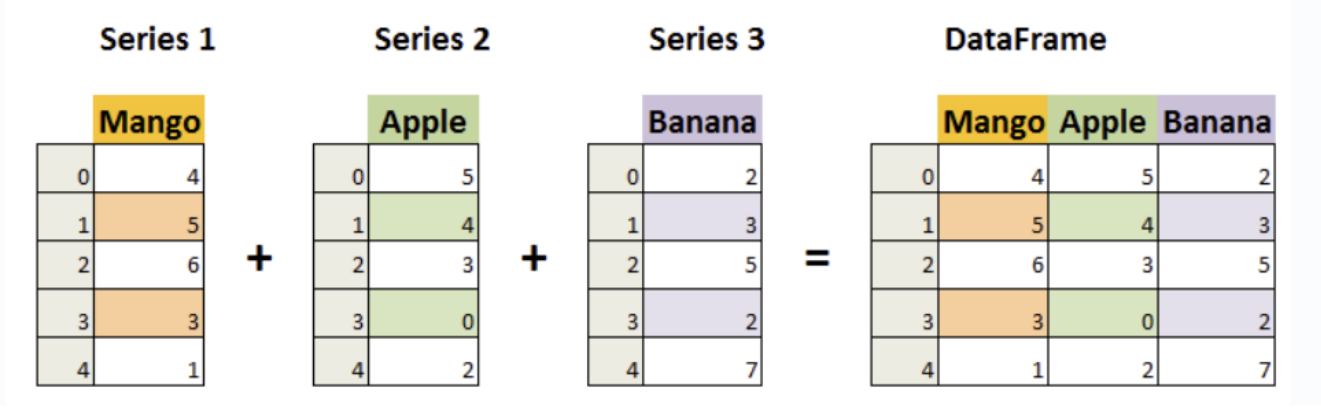
```
scores = pd.Series(
  [90, 85, 95],
  index=["Amy", "Bob", "Cindy"]
)
```

```
Amy      90
Bob      85
Cindy    95
```

DataFrame Columns Are Series

Each column inside a DataFrame is a Series.





PRACTICE / DEMO

Explore Data

Start by previewing the first few rows.

```
df.head()
```

Shows the first five rows by default.

	Name	Age	GPA
0	Amy	20	3.8
1	Bob	21	3.5
2	Cindy	22	3.9
3	David	20	3.6
4	Eva	21	3.7

PRACTICE / DEMO

Check Dataset Information

Use df.info() to inspect structure and completeness.

```
df.info()
```

Column names

Data types

Missing values

> **Before analysis, check what you have and what is missing.**

Test Your Understanding…

Create a pandas DataFrame called df1 with the following student scores:

Name	Math	English
Alice	85	78
Bob	90	88
Cindy	78	82

- Compute the mean score of the Math column.
- Compute the mean score of the English column.
- Print both mean values.
- A new student joins the class, Concatenate df1 and df2 by rows to form a new DataFrame df.

Name	Math	English
David	92	95