

Lesson 09: GroupBy Operations

Data Science with Python – BSc Course

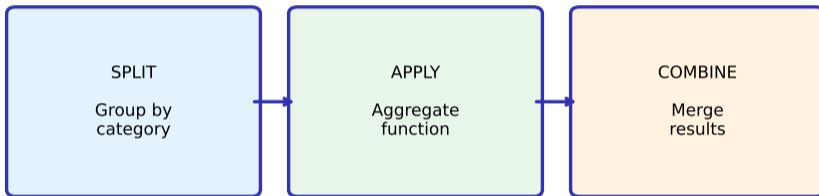
45 Minutes

After this lesson, you will be able to:

- Split-apply-combine paradigm
- `groupby()` basics
- Aggregation functions
- `transform()` vs `agg()`
- Multi-column grouping

Finance application: Stock data processing and analysis

Split-Apply-Combine Paradigm



GroupBy Workflow

```
df.groupby("Sector")["Return"].mean()
```

| _____ | | _____ | | _____ |

Group

Select

Aggregate

Aggregation Functions

mean()

Average value

sum()

Total sum

count()

Number of values

std()

Standard deviation

min()/max()

Extremes

first()/last()

First/last value

agg() vs transform()

agg()

Returns ONE value

per group

Result: smaller

transform()

Returns SAME shape

as input

Result: same size

Multi-Column GroupBy

```
df.groupby(["Sector", "Year"])["Return"].mean()
```

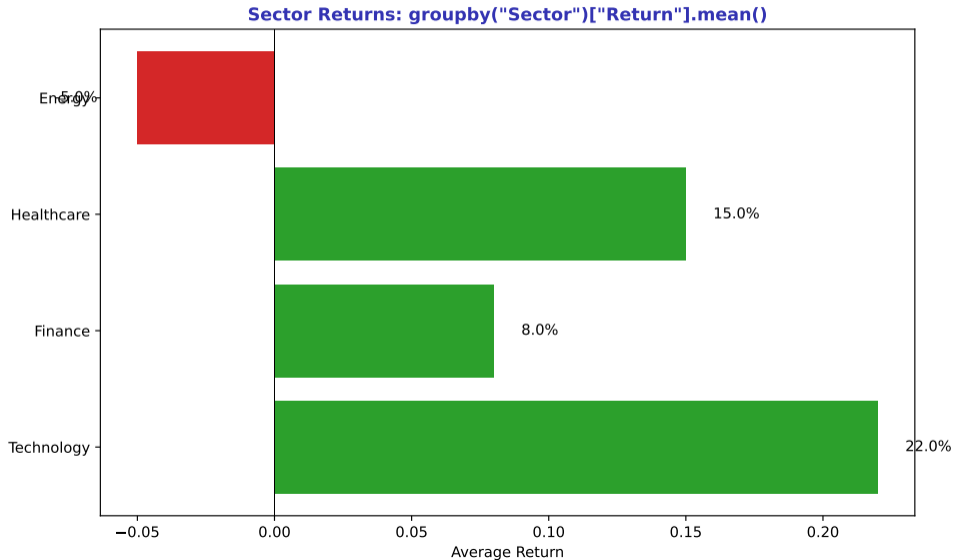
Creates hierarchical grouping:

Technology, 2023 -> 0.15

Technology, 2024 -> 0.22

Finance, 2023 -> 0.08

06 Sector Analysis



Common GroupBy Patterns

```
df.groupby("X")["Y"].agg(["mean", "std"])
```

```
df.groupby("X").agg({"A": "sum", "B": "mean"})
```

```
df.groupby("X")["Y"].transform("mean")
```

```
df.groupby("X").apply(custom_function)
```

GroupBy in Finance

Sector returns

```
groupby("Sector")["Return"].mean()
```

Monthly aggregation

```
groupby(df.index.month).sum()
```

Portfolio weights

```
groupby("Asset")["Value"].transform(lambda x: x/x.sum())
```

Risk by category

```
groupby("Rating")["Volatility"].mean()
```

Key Takeaways:

- Split-apply-combine paradigm
- `groupby()` basics
- Aggregation functions
- `transform()` vs `agg()`
- Multi-column grouping

Practice: Apply these concepts to the stock price dataset.