

Python Pandas Course for Data Analytics

Quick Study Notebook: Inspecting & Summarizing Data (Video Timeline [19:04 - 23:19])

This study reference guide focuses on how to cleanly inspect, query, and gather summary statistics from a loaded DataFrame. When starting any real-world analysis, these functions serve as the initial diagnostics toolset to evaluate structure and content.

1. The Reference Dataset

To demonstrate these methods practically, we use the following real-world style retail transaction snapshot representing a store's product sales logs:

Index	date	product	quantity	price
0	2023-12-02	tea	4	40
1	2025-05-20	coffee	2	40
2	2024-10-03	tea	3	30
3	2022-11-01	coffee	5	100
4	2026-01-05	tea	10	100

```
# Initializing the dataset in Pandas
import pandas as pd

sales_data = {
    'date': ['2023-12-02', '2025-05-20', '2024-10-03', '2022-11-01', '2026-01-05'],
    'product': ['tea', 'coffee', 'tea', 'coffee', 'tea'],
    'quantity': [4, 2, 3, 5, 10],
    'price': [40, 40, 30, 100, 100]
}
df = pd.DataFrame(sales_data)
```

2. Core Inspection Methods & Functions

df.info() **Definition:** Provides a complete technical summary of the DataFrame: total row count, explicit data types (dtypes), number of non-null parameters, and memory utilization profile.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   date        5 non-null      object
1   product     5 non-null      object
2   quantity    5 non-null      int64
3   price       5 non-null      int64
dtypes: int64(2), object(2)
memory usage: 288.0+ bytes
```

df.shape **Definition:** Returns an structural tuple indicating the dimensional layout containing absolute rows and columns counts: (rows, columns).

```
df.shape
```

```
(5, 4)
```

df.columns **Definition:** Extracts and outputs an index iterable array of all descriptive column header tags explicitly assigned to the structure.

```
df.columns
```

```
Index(['date', 'product', 'quantity', 'price'], dtype='object')
```

df.dtypes **Definition:** Returns an explicit series listing mapping out data types associated with every individual column matrix tracking elements directly.

```
df.dtypes
```

```
date        object
product     object
quantity    int64
price       int64
dtype: object
```

df.head(n) / **df.tail(n)** **Definition:** head(n) previews the top n rows of a dataset, while tail(n) displays the bottom n rows. (Default is 5 entries if blank).

```
df.head(2)
```

	date	product	quantity	price
0	2023-12-02	tea	4	40
1	2025-05-20	coffee	2	40

df.describe() **Definition:** Computes major summary distribution statistics (count, mean, standard deviation, min, percentiles, and max values) for numerical parameters.

```
df.describe()
```

	quantity	price
count	5.000000	5.000000
mean	4.800000	62.000000
std	3.114482	33.466401
min	2.000000	30.000000
25%	3.000000	40.000000
50%	4.000000	40.000000
75%	5.000000	100.000000
max	10.000000	100.000000

df.describe(include=['object']) **Definition:** Tailors summary statistics reporting directly for non-numerical / textual datasets measuring item variance, distinct labels count, top occurrence element, and categorical raw count frequencies.

```
df.describe(include=['object'])
```

	date	product
count	5	5
unique	5	2
top	2023-12-02	tea
freq	1	3

3. Categorical Unique Value Queries

df['column'].unique() **Definition:** Returns an isolated flat structural array showing all explicit distinctive categorical unique string identifiers grouped sequentially inside a single column vector block.

```
df['product'].unique()
```

```
array(['tea', 'coffee'], dtype=object)
```

df['column'].nunique() **Definition:** A descriptive scalar indicator counting and summing the overall integer number value representing unique unique classifications found within the checked series.

```
df['product'].nunique()
```

```
2
```