**Pandas Operations**

We will cover data processing, normalization, manipulation and analysis, along with techniques for grouping and aggregating data. These concepts will help you efficiently clean, transform and analyze datasets. By the end of this section, you’ll learn Pandas operations to handle real-world data effectively.

* [Data Processing with Pandas](https://www.geeksforgeeks.org/data-processing-with-pandas/?ref=ml_lbp#missing-data-handing).
* [Data Normalization in Pandas](https://www.geeksforgeeks.org/data-normalization-with-pandas/?ref=ml_lbp)
* [Data Manipulation in Pandas](https://www.geeksforgeeks.org/data-manipulattion-in-python-using-pandas/?ref=ml_lbp)
* [Data Analysis using Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe-groupby/)
* [Grouping and Aggregating with Pandas](https://www.geeksforgeeks.org/grouping-and-aggregating-with-pandas/)
* [Different Types of Joins in Pandas](https://www.geeksforgeeks.org/different-types-of-joins-in-pandas/)

**Advanced Pandas Operations**

In this section, we will explore advanced Pandas functionalities for deeper data analysis and visualization. We will cover techniques for finding correlations, working with time series data and using Pandas’ built-in plotting functions for effective data visualization. By the end of this section, you’ll have a strong grasp of advanced Pandas operations and how to apply them to real-world datasets.

* [Finding Correlation between Data](https://www.geeksforgeeks.org/python-pandas-dataframe-corr/)
* [Data Visualization with Pandas](https://www.geeksforgeeks.org/pandas-built-in-data-visualization-ml/)
* [Pandas Plotting Functions for Data Visualization](https://www.geeksforgeeks.org/pandas-plotting-functions-for-quick-data-visualization/)
* [Basic of Time Series Manipulation Using Pandas](https://www.geeksforgeeks.org/pandas-time-series-manipulation/)
* [Time Series Analysis & Visualization in Python](https://www.geeksforgeeks.org/time-series-data-visualization-in-python/)

Practicing **Pandas** is essential for any data work in Python — and mastering its methods makes preprocessing, transformation, and analysis seamless.

Here’s a **structured, complete list** of the most important **Pandas classes and methods**, grouped by category, so you can use it as a **learning + practice guide**.

**✅ 1. Core Classes**

| **Class** | **Purpose** |
| --- | --- |
| pd.Series | 1D labeled array |
| pd.DataFrame | 2D labeled data structure (table) |
| pd.Index | Immutable array for row/col labels |

**✅ 2. Creating Data**

python

CopyEdit

import pandas as pd

* pd.Series(data)
* pd.DataFrame(data, columns, index)
* pd.read\_csv(), pd.read\_excel(), pd.read\_json()
* pd.DataFrame.from\_dict()
* pd.DataFrame.from\_records()
* pd.DataFrame(np\_array, columns=[...])

**✅ 3. Inspecting Data**

python

CopyEdit

df.head(), df.tail(), df.sample()

df.shape, df.columns, df.index, df.dtypes

df.info(), df.describe()

df.memory\_usage()

df.isnull().sum()

**✅ 4. Selecting & Filtering**

df['col'], df[['col1', 'col2']]

df.iloc[0], df.iloc[:, 0], df.iloc[1:5, 0:2]

df.loc[1], df.loc[:, 'col'], df.loc[2:5, ['col1', 'col2']]

df.query("col > 5")

df[df['col'] > 10]

**✅ 5. Modifying Data**

df['new'] = df['col1'] + df['col2']

df['col'] = df['col'].astype('int')

df.rename(columns={'old': 'new'})

df.drop(columns=['col1']), df.drop(index=3)

df.insert(loc=1, column='new', value=val)

df.replace({'val1': 'val2'}), df.map(), df.apply()

**✅ 6. Sorting & Ranking**

df.sort\_values('col')

df.sort\_index()

df.rank()

**✅ 7. Handling Missing Data**

df.isnull(), df.notnull()

df.dropna(), df.dropna(axis=1)

df.fillna(0), df.fillna(method='ffill'), df.ffill(), df.bfill()

df.interpolate()

**✅ 8. Grouping & Aggregation**

df.groupby('col').mean()

df.groupby(['col1', 'col2']).agg(['sum', 'mean'])

df.pivot\_table(values='val', index='A', columns='B', aggfunc='sum')

**✅ 9. Merging, Joining, and Concatenation**

pd.concat([df1, df2], axis=0)

pd.merge(df1, df2, on='key', how='inner')

df1.join(df2, how='left')

**✅ 10. Window & Time-Series Operations**

df.rolling(window=3).mean()

df.expanding().mean()

df.ewm(span=3).mean()

For time series:

pd.to\_datetime(df['date'])

df.set\_index('date').resample('M').mean()

**✅ 11. Text and String Operations**

df['col'].str.lower(), str.upper(), str.strip()

df['col'].str.contains('word'), str.replace(), str.extract()

df['col'].str.split('-')

**✅ 12. Exporting Data**

df.to\_csv('file.csv', index=False)

df.to\_excel('file.xlsx')

df.to\_json()

df.to\_sql() # (with SQLAlchemy)

**✅ 13. Miscellaneous Handy Tools**

df.value\_counts()

df.duplicated(), df.drop\_duplicates()

df.nunique(), df.unique()

df.corr(), df.cov()

pd.cut(), pd.qcut()

pd.get\_dummies()

**📌 Suggested Practice Plan**

| **Level** | **Focus Areas** |
| --- | --- |
| **Beginner** | Create, load, inspect, select, modify |
| **Intermediate** | Filtering, aggregation, merging, missing values |
| **Advanced** | Grouping, window ops, pivoting, time series |

**🧪 Want Practice Tasks?**

Just ask:

* "Give me beginner exercises for Pandas"
* "Let’s do groupby and aggregation practice"
* "Help me explore time series in Pandas"