**📘 Pandas Data Analysis Functions — Cheat Sheet**

**🔍 1. Inspecting Data**

| **🧪 Function / Property** | **📝 Description** | **💻 Example Syntax** |
| --- | --- | --- |
| **df.head(n)** | **View first n rows** | **df.head(5)** |
| **df.tail(n)** | **View last n rows** | **df.tail(5)** |
| **df.info()** | **Summary of data types, non-null counts** | **df.info()** |
| **df.describe()** | **Summary stats for numeric columns** | **df.describe()** |
| **df.shape** | **Returns shape as (rows, columns)** | **df.shape** |
| **df.columns** | **Lists column names** | **df.columns** |
| **df.dtypes** | **Shows data types of each column** | **df.dtypes** |

**🧹 2. Cleaning Data**

| **🧪 Function / Property** | **📝 Description** | **💻 Example Syntax** |
| --- | --- | --- |
| **df.dropna()** | **Remove rows with missing values** | **df.dropna()** |
| **df.fillna(value)** | **Fill missing values with specified value** | **df.fillna(0)** |
| **df.fillna(method='ffill')** | **Forward-fill missing values** | **df.fillna(method='ffill')** |
| **df.fillna(method='bfill')** | **Backward-fill missing values** | **df.fillna(method='bfill')** |
| **df.replace(to\_replace, value)** | **Replace values** | **df.replace("?", np.nan)** |
| **df.drop\_duplicates()** | **Remove duplicate rows** | **df.drop\_duplicates()** |
| **df.duplicated()** | **Check for duplicate rows** | **df.duplicated()** |
| **df.rename(columns={})** | **Rename column names** | **df.rename(columns={"A": "Age"})** |
| **df.astype(type)** | **Convert column datatype** | **df["Age"] = df["Age"].astype(int)** |
| **df.str.strip()** | **Remove whitespace from string values** | **df["Name"] = df["Name"].str.strip()** |

**🔄 3. Filtering Rows**

| **🧪 Function / Syntax** | **📝 Description** | **💻 Example Syntax** |
| --- | --- | --- |
| **df[df["col"] > val]** | **Filter rows based on condition** | **df[df["Age"] > 30]** |
| **df.query("condition")** | **SQL-like query for filtering** | **df.query("Salary > 50000")** |
| **df.loc[row, col]** | **Label-based row/column selection** | **df.loc[0:2, "Name"]** |
| **df.iloc[row, col]** | **Index-based row/column selection** | **df.iloc[0:3, 1:3]** |

**📊 4. Aggregation / Grouping**

| **🧪 Function** | **📝 Description** | **💻 Example Syntax** |
| --- | --- | --- |
| **df.groupby("col")** | **Group data by column** | **df.groupby("Department").mean()** |
| **df.mean()** | **Mean of all numeric columns** | **df.mean()** |
| **df.sum()** | **Sum of all numeric columns** | **df.sum()** |
| **df.count()** | **Count of non-null values** | **df.count()** |
| **df["col"].value\_counts()** | **Count unique values in column** | **df["Gender"].value\_counts()** |

**🛠 5. Modifying Data**

| **🧪 Function / Syntax** | **📝 Description** | **💻 Example Syntax** |
| --- | --- | --- |
| **df["new"] = ...** | **Add a new column** | **df["Bonus"] = df["Salary"] \* 0.1** |
| **df.apply(func)** | **Apply function to rows/columns** | **df["Salary"].apply(np.log)** |
| **df.sort\_values(by="col")** | **Sort rows by a column** | **df.sort\_values(by="Age")** |
| **df.sort\_index()** | **Sort by index** | **df.sort\_index()** |

**📈 6. Visualization (using Matplotlib backend)**

| **🧪 Function** | **📝 Description** | **💻 Example Syntax** |
| --- | --- | --- |
| **df.plot()** | **General plotting** | **df.plot(kind="bar")** |
| **df.hist()** | **Histogram of numeric data** | **df.hist()** |
| **df.plot.scatter(x, y)** | **Scatter plot** | **df.plot.scatter(x="Age", y="Salary")** |

**📁 7. Exporting Data**

| **🧪 Function** | **📝 Description** | **💻 Example Syntax** |
| --- | --- | --- |
| **df.to\_csv("file.csv")** | **Export DataFrame to CSV** | **df.to\_csv("data.csv", index=False)** |
| **df.to\_excel("file.xlsx")** | **Export DataFrame to Excel** | **df.to\_excel("data.xlsx")** |
| **df.to\_json("file.json")** | **Export DataFrame to JSON** | **df.to\_json("data.json")** |

**📥 8. Importing Data**

| **🧪 Function** | **📝 Description** | **💻 Example Syntax** |
| --- | --- | --- |
| **pd.read\_csv("file.csv")** | **Read CSV file into DataFrame** | **pd.read\_csv("data.csv")** |
| **pd.read\_excel("file.xlsx")** | **Read Excel file** | **pd.read\_excel("data.xlsx")** |
| **pd.read\_sql("query", con)** | **Read data from SQL query** | **pd.read\_sql("SELECT \* FROM table", con=engine)** |