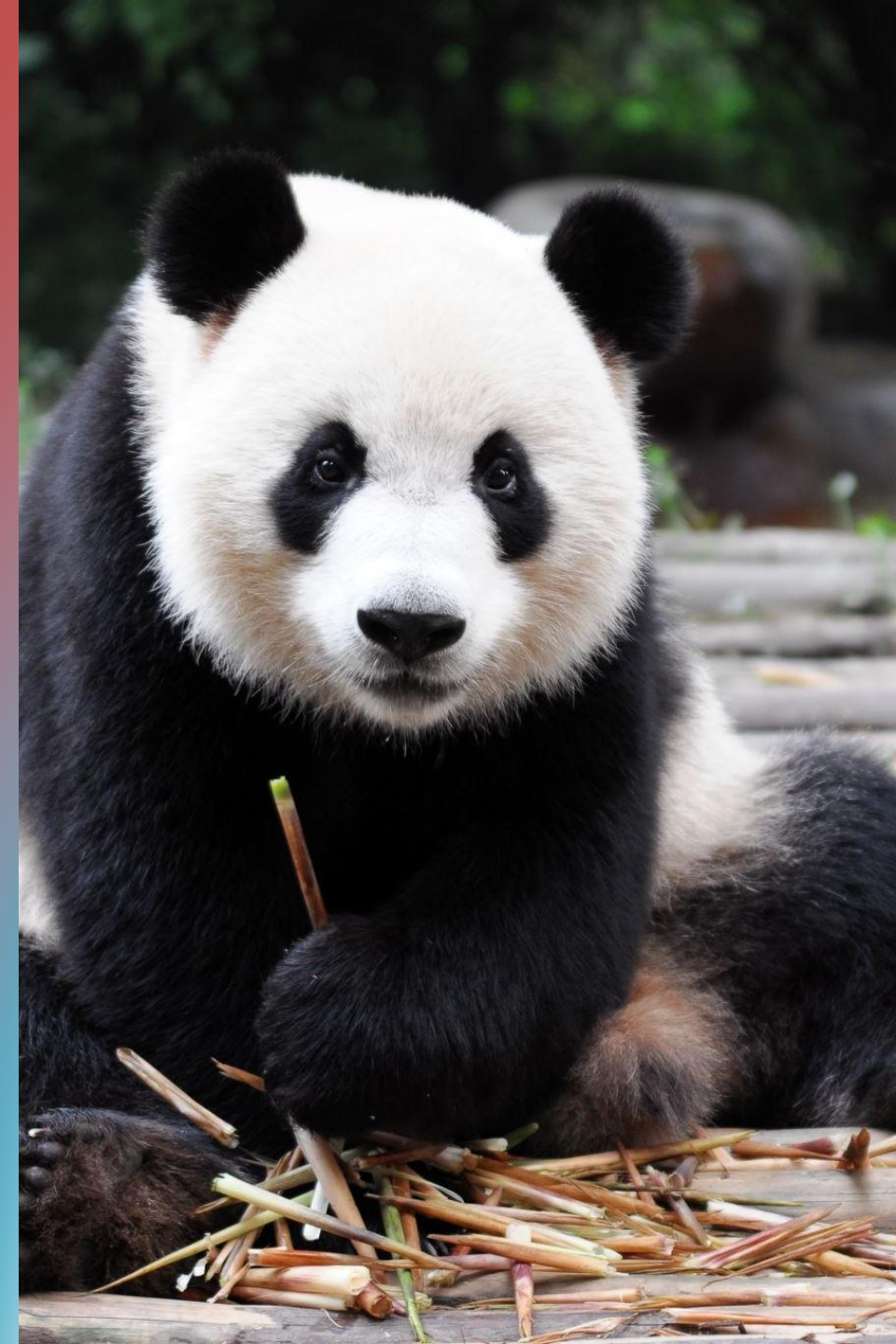




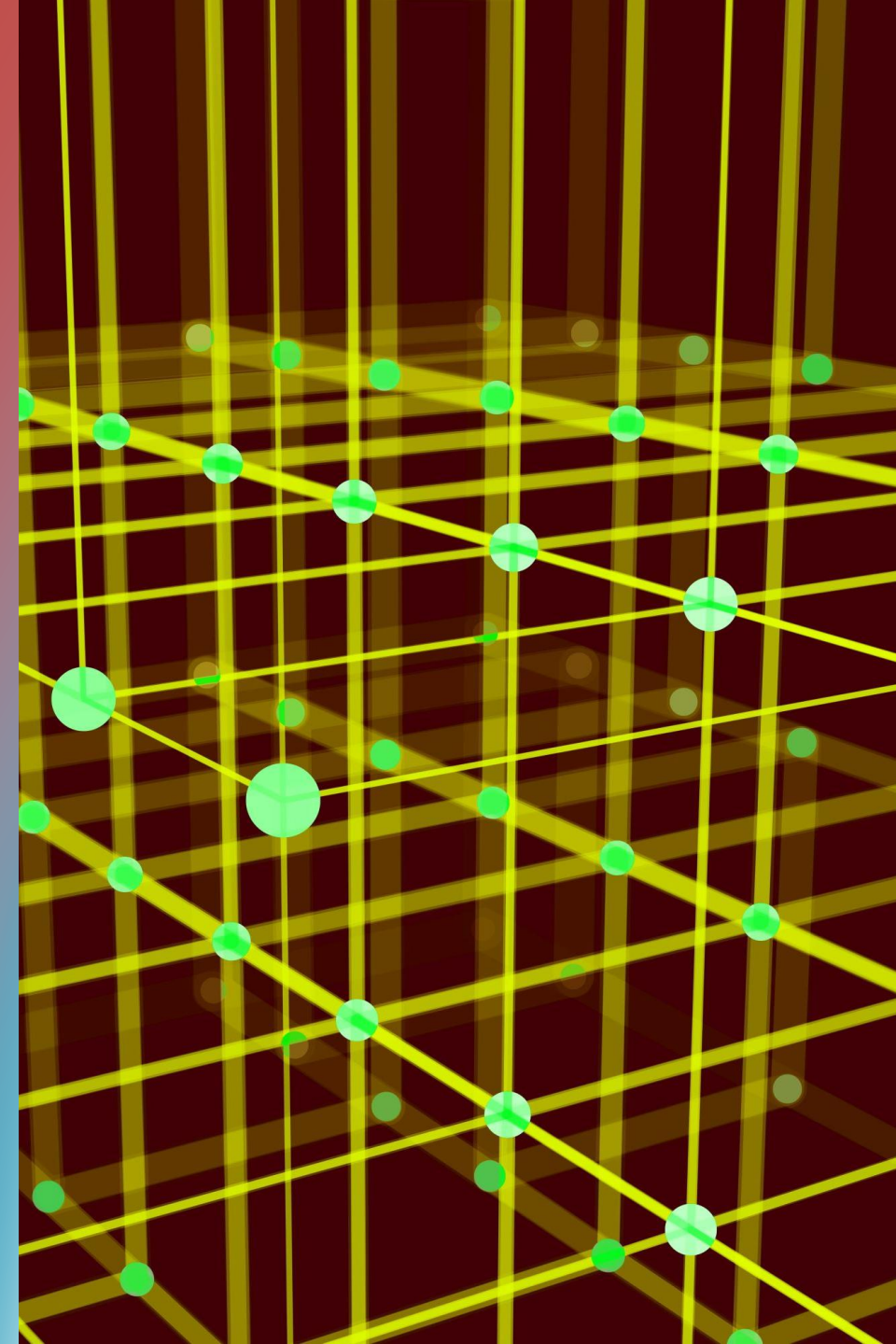
Pandas Commands Overview

A quick guide to commonly
used pandas commands



What is Pandas?

- Pandas is a powerful Python library for data analysis and manipulation.
- Key Features:
 - - Handle structured data like tables and time series
 - - Easy data cleaning, manipulation, and visualization
 - - Supports CSV, Excel, SQL, and more formats
 - - Ideal for data preprocessing before machine learning



Series vs. DataFrame in Pandas

- Series:
 - - A one-dimensional array-like object
 - - Can hold any data type: integers, floats, strings, etc.
 - - Example: A single column in an Excel sheet
- DataFrame:
 - - A two-dimensional, tabular data structure
 - - Consists of rows and columns
 - - Example: An entire Excel sheet with multiple columns

Why Use Pandas?

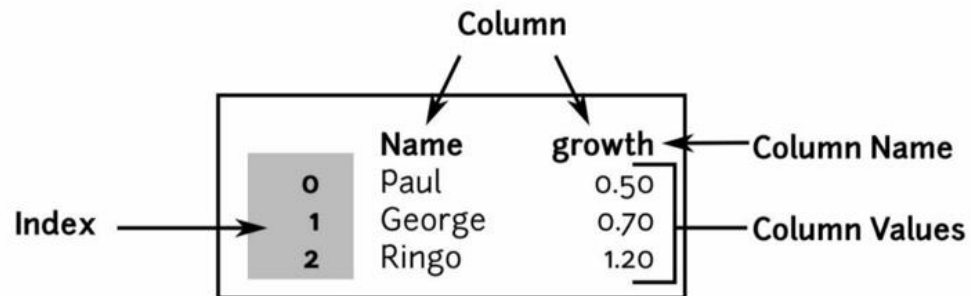
- 1. Easy-to-use data structures for analysis.
- 2. Efficient handling of missing data.
- 3. Powerful tools for group operations, merging, and reshaping data.
- 4. Seamlessly integrates with other libraries like NumPy and Matplotlib.

- To Install Pandas and xlrd libraries , use following commands in GC

⇒ !pip install pandas

⇒ !pip install xlrd

Data Frame



Different dimensions of pandas data structures

DATA STRUCTURE	DIMENSIONALITY	SPREADSHEET ANALOG
Series	1D	Column
DataFrame	2D	Single Sheet
Panel	3D	Multiple Sheets

Data Structures

Series

	age
0	15
1	16
2	16
3	15

	teacher
0	Ashby
1	Ashby
2	Jones
3	Jones

	name
0	Adam
1	Bob
2	Dave
3	Fred

Data Frame

	age	name	teacher
0	15	Adam	Ashby
1	16	Bob	Ashby
2	16	Dave	Jones
3	15	Fred	Jones

Panel

	age	name	teacher
0	15	Adam	Ashby
1	16	Bob	Ashby
2	16	Dave	Jones
3	15	Fred	Jones



pd.read_csv()

- Description:
- Reads a CSV file into a DataFrame.
- Example:
- `df = pd.read_csv('file.csv')`

pd.DataFrame()

- Description:
- Creates a new DataFrame from data.
- Example:
- `df = pd.DataFrame({'col1': [1, 2], 'col2': [3, 4]})`



df.head()

- Description:
- Returns the first few rows of a DataFrame.
- Example:
- `df.head()`

A red pushpin is stuck into a map, which is visible on the left side of the slide. The map shows some streets and landmarks, though they are out of focus. The pushpin is sharp and its shadow is cast on the map.

df.info()

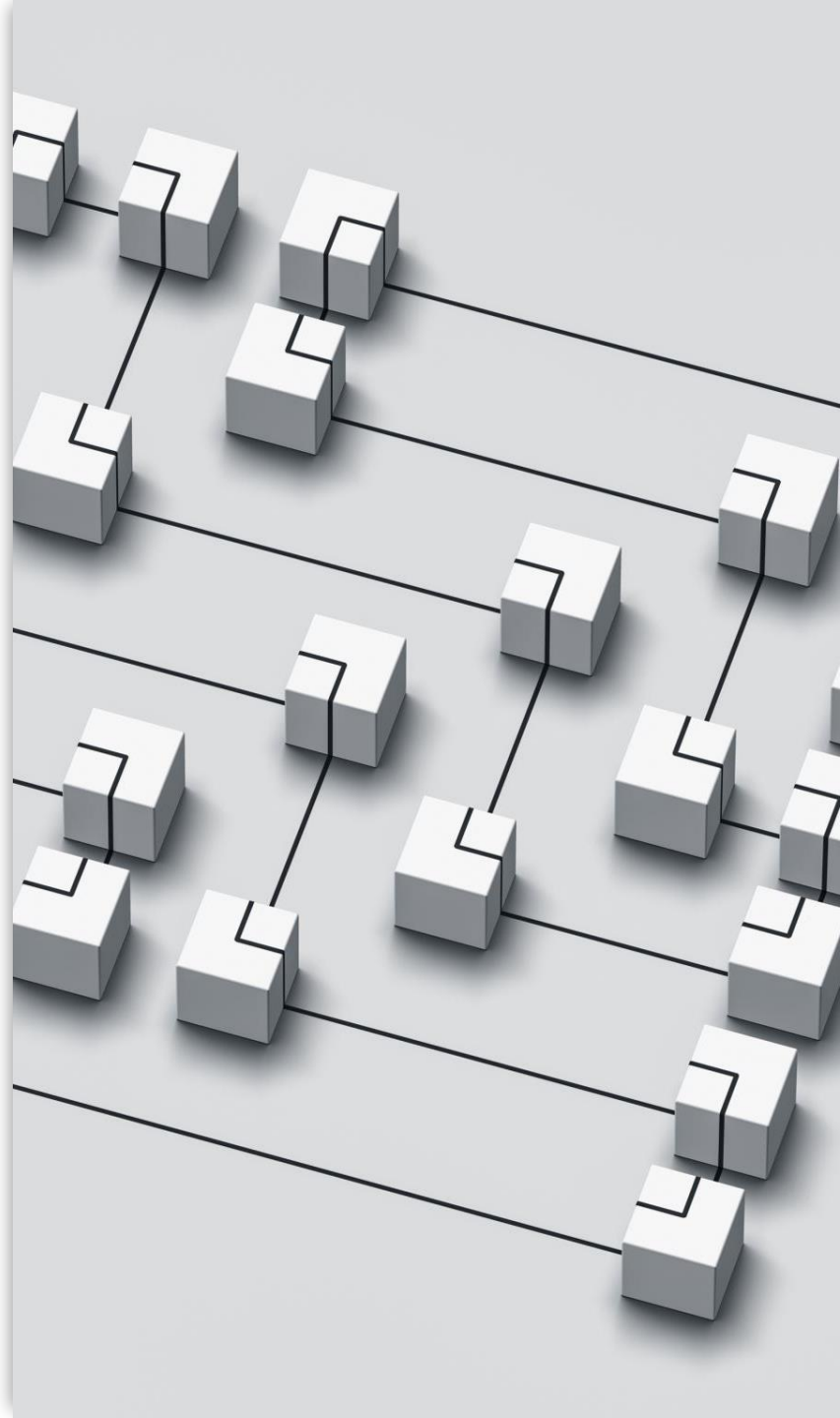
- Description:
- Displays a summary of the DataFrame.
- Example:
- `df.info()`

df.describe()

- Description:
- Generates descriptive statistics of DataFrame columns.
- Example:
- `df.describe()`

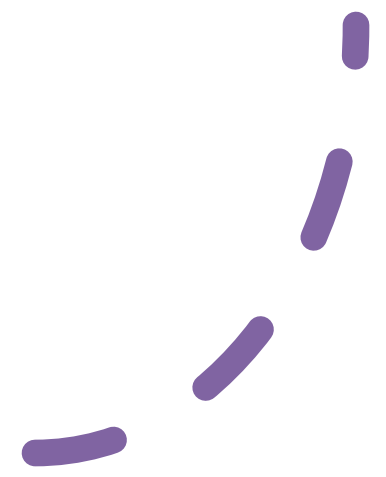
df.loc[]

- Description:
- Accesses a group of rows and columns by labels or a boolean array.
- Example:
- `df.loc[0:2, 'col1']`





df.iloc[]

- Description:
 - Accesses rows and columns by integer-based index.
 - Example:
 - `df.iloc[0:2, 0:2]`
- 

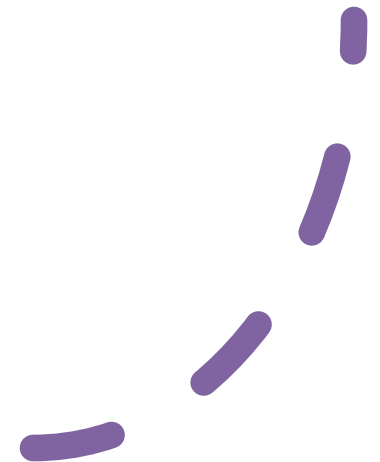


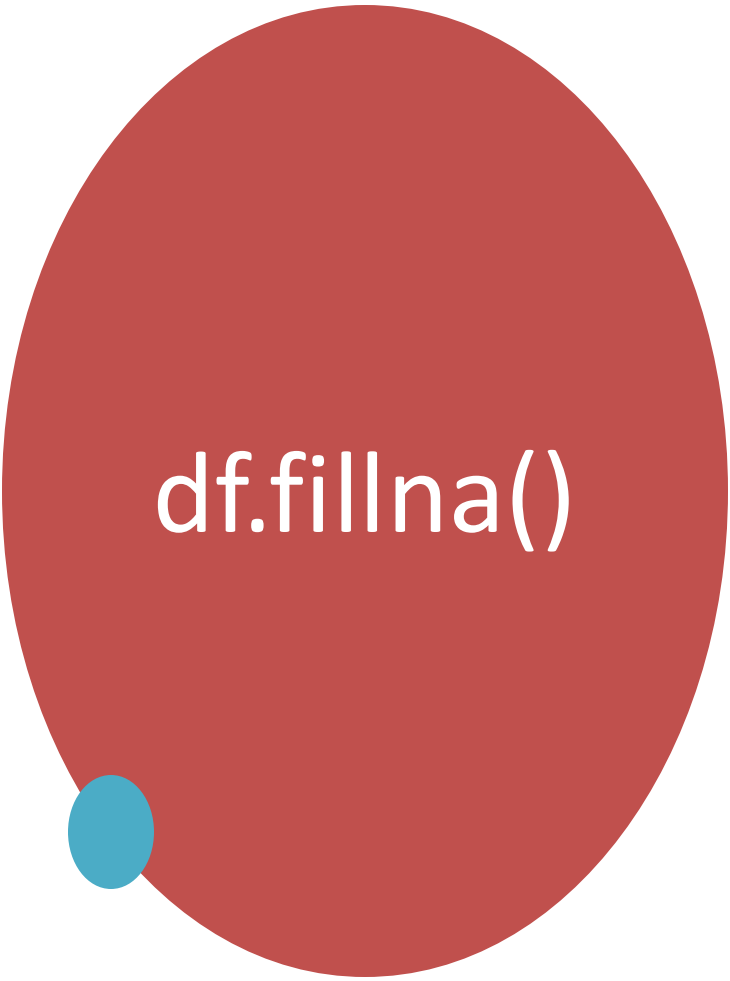
df.drop()

- Description:
 - Drops specified labels from rows or columns.
 - Example:
 - `df.drop('col1', axis=1)`
- 

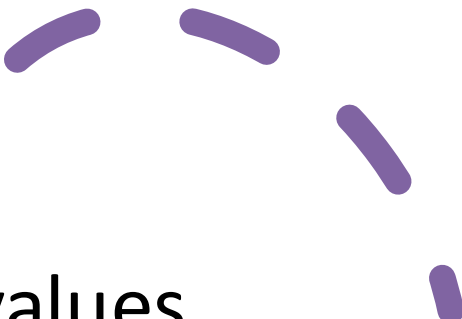
df.isnull()

- Description:
- Detects missing values.
- Example:
- df.isnull()





`df.fillna()`

- 
- Description:
 - Fills missing values with specified values.
 - Example:
 - `df.fillna(0)`

df.groupby()

- Description:
- Groups data by a column or columns.
- Example:
- `df.groupby('col1').mean()`

pd.merge()

- Description:
- Merges DataFrame objects by columns or indexes.
- Example:
- `pd.merge(df1, df2, on='key')`

df.sort_values()

- Description:
- Sorts the DataFrame by specified column(s).
- Example:
- `df.sort_values('col1')`

df.apply()

- Description:
- Applies a function along an axis of the DataFrame.
- Example:
- `df['col1'].apply(lambda x: x**2)`

pd.concat()

- Description:
- Concatenates pandas objects along a particular axis.
- Example:
- `pd.concat([df1, df2])`

pd.read_csv()

- Description:
- Reads a CSV file into a DataFrame.
- Example:
- `df = pd.read_csv('data.csv')`

pd.read_excel()

- Description:
- Reads an Excel file into a DataFrame.
- Example:
- `df = pd.read_excel('data.xlsx')`

pd.to_datetime()

- Description:
- Converts a column or series to datetime format.
- Example:
- `df['date'] =
pd.to_datetime(df['date'])`

pd.cut()

- Description:
- Bins values into discrete intervals.
- Example:
- `pd.cut(df['ages'], bins=[0, 18, 35, 60], labels=['Child', 'Adult', 'Senior'])`

pd.get_dummies()

- Description:
- Converts categorical data into dummy/indicator variables.
- Example:
- `pd.get_dummies(df['category'])`

pd.pivot_table()

- Description:
- Creates a pivot table from data.
- Example:
- `pd.pivot_table(df, values='sales', index='month', columns='product')`

df.tail()

- Description:
- Returns the last few rows of a DataFrame.
- Example:
- `df.tail()`

df.columns

- Description:
- Returns the column labels of the DataFrame.
- Example:
- `df.columns`

df.rename()

- Description:
- Renames columns or rows in a DataFrame.
- Example:
- `df.rename(columns={'old_name': 'new_name'})`



df.duplicated()

- Description:
- Finds duplicate rows in the DataFrame.
- Example:
- `df.duplicated()`



`df.drop_duplicates()`

- Description:
- Drops duplicate rows from the DataFrame.
- Example:
- `df.drop_duplicates()`



df.sample()

- Description:
 - Returns a random sample of rows from the DataFrame.
 - Example:
 - `df.sample(n=5)`
-



df.nunique()

- Description:
 - Counts the number of unique values in each column.
 - Example:
 - `df.nunique()`
-



df.memory_usage()

- Description:
 - Displays memory usage of DataFrame columns.
 - Example:
 - `df.memory_usage(deep=True)`
-

A decorative element on the left side of the slide consisting of four vertical bars of increasing height from left to right.

df.sort_index()

- Description:
 - Sorts the DataFrame by its index.
 - Example:
 - `df.sort_index()`
-



df.pivot()

- Description:
 - Reshapes data based on column values.
 - Example:
 - `df.pivot(index='date', columns='product', values='sales')`
-

df.melt()

- Description:
- Transforms DataFrame from wide to long format.
- Example:
- `pd.melt(df, id_vars=['id'], value_vars=['col1', 'col2'])`

df.transpose()

- Description:
- Transposes the DataFrame (rows become columns and vice versa).
- Example:
- `df.transpose()`

df.corr()

- Description:
- Calculates correlation between columns in the DataFrame.
- Example:
- `df.corr()`



df.replace()

- Description:
- Replaces specified values with others in the DataFrame.
- Example:
- `df.replace({'old_value': 'new_value'})`

df.astype()

- Description:
- Converts columns to a specified data type.
- Example:
- `df['col1'] = df['col1'].astype(float)`

df.query()

Description:

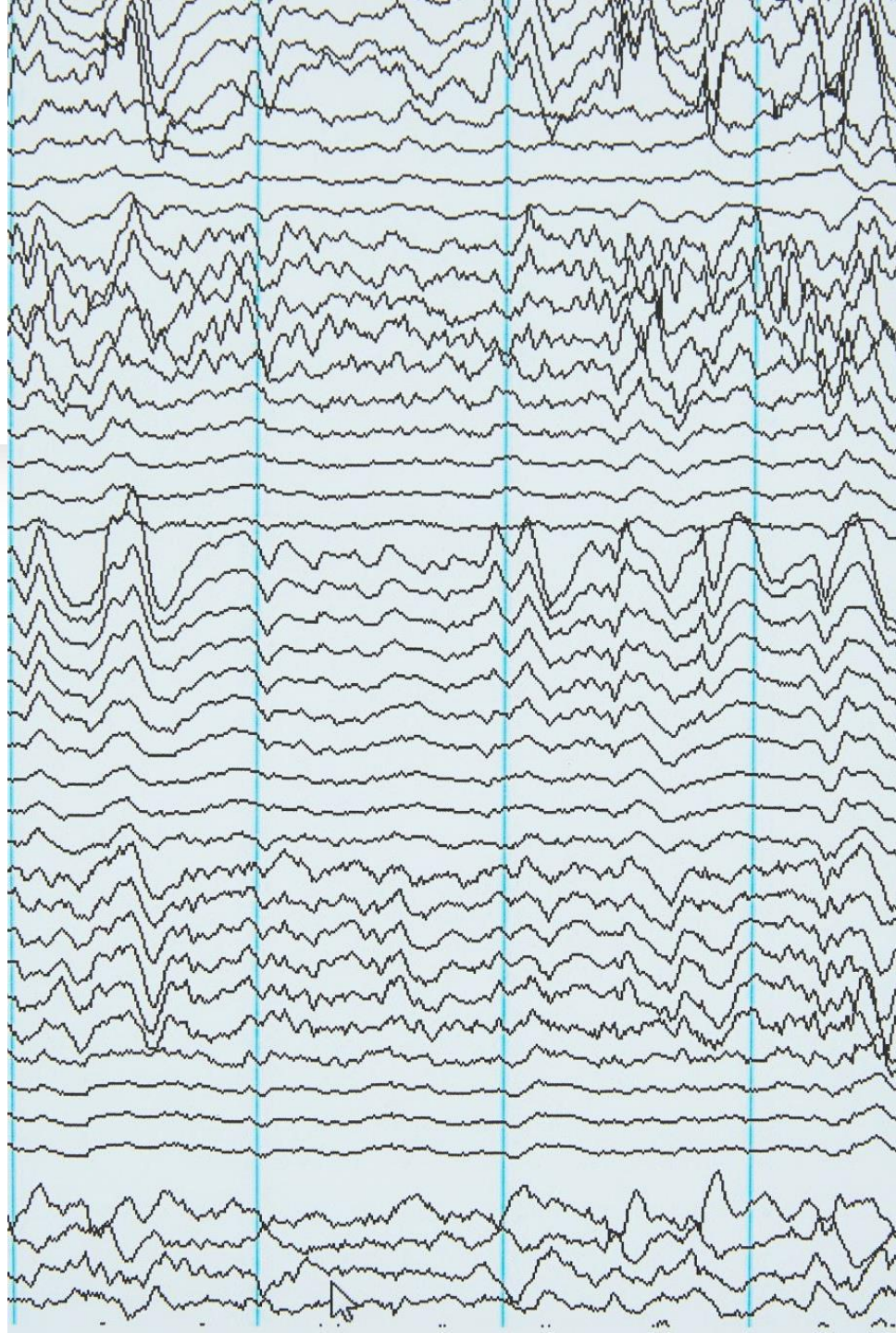
Queries the DataFrame using a string expression.

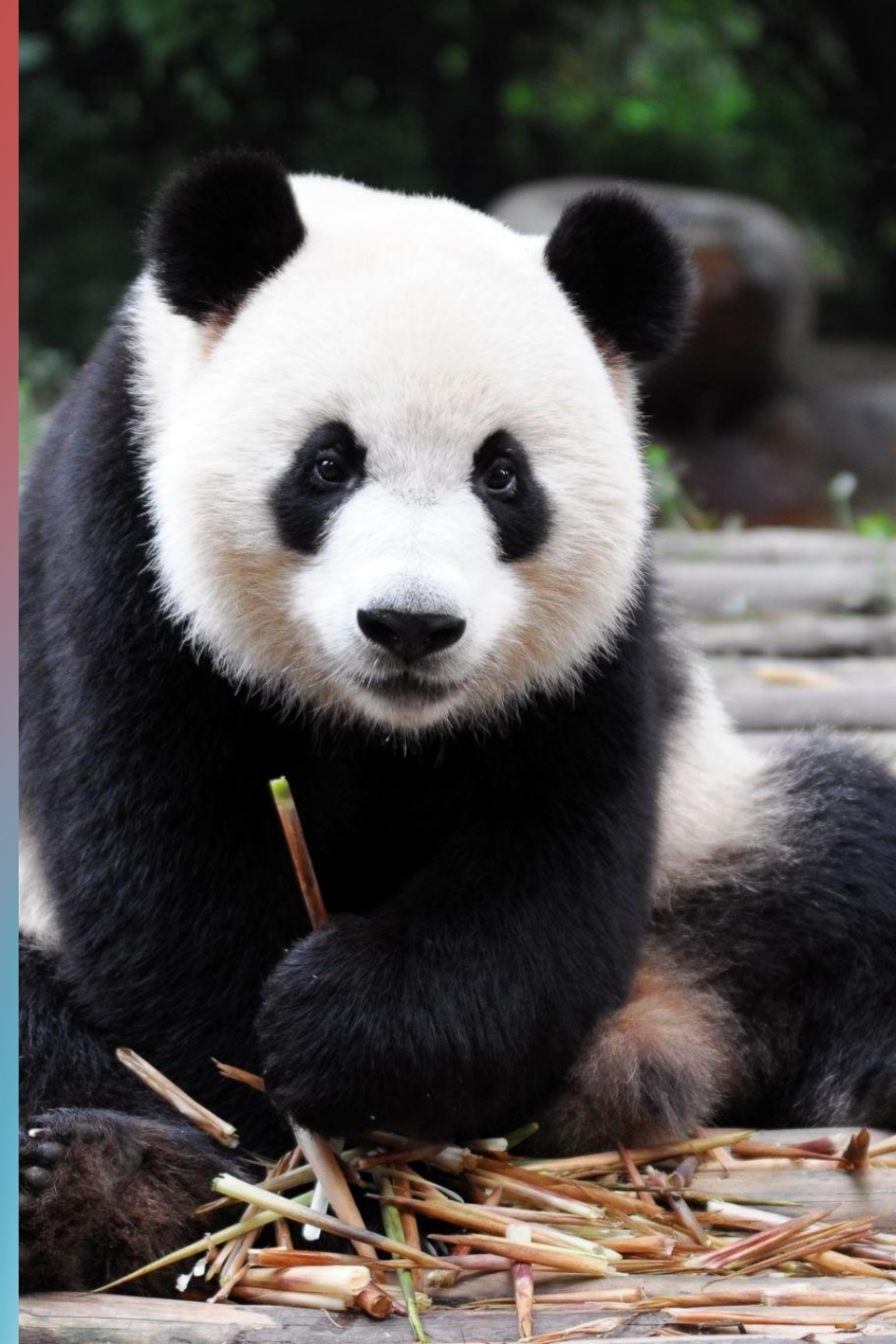
Example:

```
df.query('col1 > 10')
```

df.plot()

- Description:
- Generates plots using matplotlib.
- Example:
- `df['col1'].plot(kind='line')`





pd.Series()

- Description:
- Creates a pandas Series object.
- Example:
- `s = pd.Series([1, 2, 3], index=['a', 'b', 'c'])`

pd.isna()

- Description:
- Checks for missing values in a DataFrame or Series.
- Example:
- `pd.isna(df['col1'])`





pd.notna()

- Description:
- Checks for non-missing values in a DataFrame or Series.
- Example:
- `pd.notna(df['col1'])`

df.value_counts()

- Description:
- Counts occurrences of unique values in a Series.
- Example:
- `df['col1'].value_counts()`