

The Python CheetSheet:

Panda's, Numpy and Matplotlib CheatSheet:

A

- **agg() [Pandas]:** A method used with `groupby()` to apply multiple aggregation functions (like `sum`, `mean`, `count`) at once.
- **apply() [Pandas]:** A method to apply a function along an axis of a DataFrame, typically to each column or row.
- **arange() [NumPy]:** Creates an array with evenly spaced values within a given interval, with a specified step size.
- **array() [NumPy]:** The fundamental function to create a NumPy array from a Python list or tuple.
- **Axes [Matplotlib]:** The individual plot object within a figure that contains the data, x-axis, y-axis, title, etc. You interact with this object when using the object-oriented approach.
- **astype() [Pandas/NumPy]:** A method to cast an object (like a DataFrame column or NumPy array) to a specified data type (`dtype`).

B

- **bar() [Matplotlib]:** Creates a bar chart, ideal for comparing numerical values across different categories.
- **Boolean Indexing [NumPy/Pandas]:** The powerful technique of using a boolean array (True/False values) to filter data and select elements that meet a certain condition.

C

- **concat() [Pandas]**: A function to stack multiple DataFrames or Series together either vertically (along rows) or horizontally (along columns).
- **corr() [Pandas]**: A method to compute the pairwise correlation of columns in a DataFrame.
- **count() [Pandas]**: A method to count the number of non-null entries in a Series or each column of a DataFrame.
- **cumsum() [Pandas/NumPy]**: Calculates the cumulative sum of elements along a given axis.

D

- **DataFrame [Pandas]**: The primary Pandas data structure; a 2-dimensional, size-mutable, and potentially heterogeneous tabular data structure with labeled axes (rows and columns).
- **describe() [Pandas]**: A method that generates descriptive statistics (count, mean, std, min, max, quartiles) for the numerical columns of a DataFrame.
- **dot() [NumPy]**: A function for matrix multiplication or calculating the dot product of two vectors.
- **dropna() [Pandas]**: A method to remove missing values (NaN) from a Series or DataFrame.
- **drop() [Pandas]**: A method to remove specified rows or columns from a DataFrame.
- **dtype [NumPy/Pandas]**: An object that describes the data type of the elements in an array or Series (e.g., int64, float64, object).

F

- **figure() [Matplotlib]**: Creates a new figure, which is the top-level container for all plot elements. You can specify its size with the figsize argument.
- **fillna() [Pandas]**: A method to fill missing (NaN) values using a specified value, mean, or method like forward-fill (ffill).

G

- **grid()** [Matplotlib]: A function to configure and display grid lines on a plot.
- **groupby()** [Pandas]: A powerful method used to split a DataFrame into groups based on some criteria, apply a function to each group independently, and then combine the results. This is the core of the "split-apply-combine" pattern.

H

- **head()** [Pandas]: A method to view the first `n` rows of a DataFrame (default is 5).
- **hist()** [Matplotlib]: Creates a histogram, which is used to visualize the distribution of a single numerical variable.

I

- **iloc[]** [Pandas]: An attribute for integer-location based indexing, used to select data from a DataFrame by its integer position (e.g., `df.iloc[0, 0]` for the top-left cell).
- **index** [Pandas]: The labeled axis for the rows of a DataFrame. It can be numbers, dates, or strings.
- **info()** [Pandas]: A method that prints a concise summary of a DataFrame, including the index dtype, column dtypes, non-null values, and memory usage.
- **isin()** [Pandas]: A method used for filtering data by checking whether each element in a Series or DataFrame is contained in a given sequence of values.

J

- **join()** [Pandas]: A convenient method for combining columns of two potentially differently-indexed DataFrames into a single DataFrame.

L

- **legend() [Matplotlib]**: Places a legend on the plot to identify different plotted lines or markers.
- **linspace() [NumPy]**: Creates an array with a specified number of evenly spaced values over a specified interval.
- **loc[] [Pandas]**: An attribute for label-based indexing, used to select data from a DataFrame by its row and column labels (e.g., df.loc['2025-06-28', 'Close']).

M

- **map() [Pandas]**: A Series method used for substituting each value in a Series with another value, based on a function or a dictionary.
- **merge() [Pandas]**: A function for combining two DataFrames based on the values of common columns, similar to a SQL join.

N

- **NaN (Not a Number) [NumPy/Pandas]**: The standard representation for missing data.
- **ndarray [NumPy]**: The N-dimensional array object that is the core of the NumPy library.

P

- **pct_change() [Pandas]**: A method to calculate the percentage change between the current and a prior element. Essential for calculating financial returns.
- **pivot_table() [Pandas]**: A function that creates a spreadsheet-style pivot table as a DataFrame, allowing you to easily summarize and aggregate data.
- **plot() [Matplotlib/Pandas]**: The fundamental function for creating a line plot. Pandas DataFrames and Series have a

.plot() method that acts as a convenient wrapper around Matplotlib's plt.plot().

R

- **random [NumPy]**: A submodule for generating random numbers and arrays from various probability distributions (rand, randn, randint).
- **read_csv() [Pandas]**: The essential function for reading a comma-separated values (CSV) file into a DataFrame.
- **rename() [Pandas]**: A method to alter axis labels (i.e., column names or index labels).
- **resample() [Pandas]**: A method for time series frequency conversion and resampling (e.g., converting daily data to monthly data).
- **reshape() [NumPy]**: A function that gives a new shape to an array without changing its data.

S

- **savefig() [Matplotlib]**: A function to save the current figure to a file (e.g., .png, .pdf).
- **scatter() [Matplotlib]**: Creates a scatter plot, used to visualize the relationship between two numerical variables.
- **Series [Pandas]**: The one-dimensional labeled array that is the second core data structure of Pandas. A DataFrame is a collection of Series.
- **set_index() [Pandas]**: A method to set a DataFrame column as the index.
- **shape [NumPy/Pandas]**: An attribute that returns a tuple representing the dimensions (rows, columns) of an array or DataFrame.
- **show() [Matplotlib]**: A function that displays all open figures.

- **sort_values() [Pandas]**: A method to sort a DataFrame by the values in one or more columns.
- **subplots() [Matplotlib]**: A function that creates a figure and a grid of subplots in a single call, providing a powerful way to manage complex figures (the object-oriented approach).
- **sum() [NumPy/Pandas]**: A function or method to calculate the sum of elements.

T

- **tail() [Pandas]**: A method to view the last `n` rows of a DataFrame (default is 5).
- **title() [Matplotlib]**: Sets a title for the current axes (the individual plot).
- **to_csv() [Pandas]**: A method to write a DataFrame to a comma-separated values (CSV) file.

U

- **unique() [Pandas]**: A Series method that returns an array of the unique values in that Series.

V

- **value_counts() [Pandas]**: A Series method that returns a new Series containing the counts of unique values.
- **Vectorization [NumPy/Pandas]**: The practice of applying operations to entire arrays at once instead of iterating through elements one by one. This is the key to performance.

X

- **xlabel() [Matplotlib]**: Sets the label for the x-axis.
- **xlim() [Matplotlib]**: Sets the limits for the x-axis.

- **xticks() [Matplotlib]**: Sets the tick locations and labels for the x-axis.

Y

- **ylabel() [Matplotlib]**: Sets the label for the y-axis.
- **ylim() [Matplotlib]**: Sets the limits for the y-axis.
- **yticks() [Matplotlib]**: Sets the tick locations and labels for the y-axis.