

Comprehensive Guide: NumPy, Pandas, Matplotlib

NumPy Functions (Basic to Advanced)

1. `np.array()` - Creates an array from a list or tuple.

Example: `np.array([1, 2, 3])`

2. `np.arange()` - Creates an array with a range of values.

Example: `np.arange(0, 10, 2)`

3. `np.linspace()` - Generates an array of evenly spaced values.

Example: `np.linspace(0, 1, 5)`

4. `np.zeros()`, `np.ones()` - Creates arrays of zeros or ones.

Example: `np.zeros((3, 3))`, `np.ones((2, 2))`

5. `np.random.rand()` - Generates random values in a specified shape.

Example: `np.random.rand(3, 3)`

6. `np.mean()`, `np.median()`, `np.std()` - Mean, median, and standard deviation.

Example: `np.mean(arr)`, `np.median(arr)`, `np.std(arr)`

7. `np.dot()` - Matrix multiplication or dot product of two arrays.

Example: `np.dot(arr1, arr2)`

8. `np.linalg.inv()` - Computes the inverse of a matrix.

Example: `np.linalg.inv(matrix)`

Advanced Functions:

1. `np.einsum()` - Efficiently computes sum of products along specific axes.

Example: `np.einsum('i,i->', arr1, arr2)`

2. `np.fft.fft()` - Fast Fourier Transform.

Example: `np.fft.fft(signal)`

3. `np.apply_along_axis()` - Applies a function along a specified axis.

Example: `np.apply_along_axis(func, 0, arr)`

4. `np.vectorize()` - Vectorizes a function to work element-wise on arrays.

Example: `np.vectorize(func)(arr)`

5. `np.broadcast()` - Allows broadcasting arrays with different shapes.

Example: `np.broadcast(arr1, arr2)`

Pandas Functions (Basic to Advanced)

1. `pd.Series()` - Creates a series from a list, dictionary, or scalar.

Example: `pd.Series([1, 2, 3])`

2. `pd.DataFrame()` - Creates a DataFrame from a dictionary or array.

Example: `pd.DataFrame({'A': [1, 2], 'B': [3, 4]})`

3. `df.head()`, `df.tail()` - Display first/last n rows of DataFrame.

Example: `df.head(5)`, `df.tail(5)`

4. `df.info()`, `df.describe()` - Shows DataFrame summary and statistics.

Example: `df.info()`, `df.describe()`

5. `df.isnull()`, `df.fillna()` - Checking and filling missing values.

Example: `df.isnull()`, `df.fillna(0)`

6. `df.groupby()` - Group data by columns for aggregation.

Example: `df.groupby('column_name').sum()`

7. `df.merge()` - Merges DataFrames based on a common column.

Example: `pd.merge(df1, df2, on='id')`

8. `df.pivot_table()` - Creates a pivot table from data.

Example: `df.pivot_table(values='sales', index='category', aggfunc='sum')`

Advanced Functions:

1. `pd.concat()` - Concatenates DataFrames along rows or columns.

Example: `pd.concat([df1, df2])`

2. `pd.melt()` - Unpivots DataFrames from wide to long format.

Example: `pd.melt(df, id_vars=['id'], value_vars=['col1', 'col2'])`

3. `pd.crosstab()` - Computes cross-tabulations of data.

Example: `pd.crosstab(df['column1'], df['column2'])`

4. `df.apply()` - Applies a function along an axis (rows/columns).

Example: `df.apply(func, axis=1)`

5. `df.query()` - Queries the DataFrame using a string expression.

Example: `df.query('age > 30')`

6. `df.sort_values()` - Sorts values by specified column.

Example: `df.sort_values(by='sales')`

Matplotlib Functions (Basic to Advanced)

1. `plt.plot()` - Basic plot function for 2D data.

Example: `plt.plot(x, y)`

2. `plt.scatter()` - Creates a scatter plot.

Example: `plt.scatter(x, y)`

3. `plt.bar()` - Creates a bar chart.

Example: `plt.bar(x, height)`

4. `plt.hist()` - Creates a histogram.

Example: `plt.hist(data, bins=10)`

5. `plt.xlabel()`, `plt.ylabel()` - Labels x and y axes.

Example: `plt.xlabel('X-axis Label'), plt.ylabel('Y-axis Label')`

6. `plt.title()` - Adds a title to the plot.

Example: `plt.title('Plot Title')`

7. `plt.legend()` - Adds a legend to the plot.

Example: `plt.legend(['Label1', 'Label2'])`

8. `plt.show()` - Displays the plot.

Example: `plt.show()`

Advanced Functions:

1. `plt.subplot()` - Creates a grid of subplots.

Example: `plt.subplot(2, 2, 1)`

2. `plt.contour()` - Creates contour plots.

Example: `plt.contour(X, Y, Z)`

3. `plt.pie()` - Creates a pie chart.

Example: `plt.pie(sizes, labels=labels)`

4. `plt.fill()` - Fills an area under a plot.

Example: `plt.fill(x, y, color='blue')`

5. `plt.scatter()` - Creates a scatter plot with customization options.

Example: `plt.scatter(x, y, color='red', size=50)`

6. `plt.tight_layout()` - Adjusts layout to fit elements properly.

Example: `plt.tight_layout()`

7. `plt.savefig()` - Saves the plot to a file.

Example: `plt.savefig('plot.png')`