

## Numpy

1. Create a NumPy array from a Python list `[10, 20, 30, 40]`.
2. Create a  $3 \times 3$  matrix filled with zeros.
3. Create a  $4 \times 4$  identity matrix.
4. Create an array from 1 to 20.
5. Create an array with values from 0 to 1 with step 0.1.
6. Create a 1D array of 12 elements and reshape it into:
  - $3 \times 4$
  - $2 \times 6$
7. Convert a 2D array into 1D.
8. Check the shape and dimension of an array.
9. Find total number of elements in an array.
10. Create a 3D array of shape (2,3,4).
11. From array `[10, 20, 30, 40, 50]`, extract:
  - First element
  - Last element
  - Middle three elements
12. Reverse an array.
13. Extract second row from a 2D array.
14. Extract last column from a 2D matrix.
15. Replace all even numbers with 0.
16. Add two arrays element-wise.
17. Multiply two arrays.
18. Find mean, median, and standard deviation.
19. Find maximum and minimum value.
20. Sort an array.
21. Find all elements greater than 50.
22. Replace negative values with 0.
23. Count number of even numbers.
24. Find indices where value  $> 100$ .
25. Create array of random integers between 10 and 100.
26. Matrix multiplication of two  $3 \times 3$  matrices.
27. Find transpose of matrix.
28. Find determinant.
29. Find inverse of matrix.
30. Solve linear equation using NumPy.
31. Add scalar 10 to entire matrix.
32. Subtract row vector from matrix.
33. Multiply column vector with matrix.
34. Explain why broadcasting fails in some cases.
35. Fix shape mismatch error.

36. Extract diagonal elements.
37. Replace diagonal elements with 0.
38. Get elements at positions [0,2,4].
39. Extract elements greater than mean.
40. Create checkerboard matrix.
41. Convert loop-based sum into vectorized NumPy solution.
42. Compare execution time of Python list vs NumPy.
43. Create large 10000×10000 array and optimize memory usage.
44. Explain difference between `copy()` and `view()`.
45. Demonstrate memory sharing.
46. Create 3D array and slice:
  - First matrix
  - Last row of second matrix
47. Sum across:
  - Rows
  - Columns
  - Depth
48. Flatten 3D array.
49. Find max value index in 3D.
50. Swap axes in 3D array.
51. Normalize array using Min-Max scaling.
52. Standardize array (Z-score normalization).
53. Remove duplicate rows.
54. Find unique elements and their counts.
55. Create sliding window of size 3.
56. Why NumPy is faster than Python list?
57. What is vectorization?
58. What is broadcasting?
59. Difference between:
  - `np.array` vs `np.asarray`
  - `reshape` vs `resize`
60. What happens if you modify a sliced array?
61. Implement softmax using NumPy.
62. Implement gradient descent using NumPy only.
63. Compute covariance matrix manually.
64. Perform PCA using NumPy only.
65. Create one-hot encoding manually.
66. Implement convolution operation.
67. Simulate image grayscale conversion.

68. Compute cosine similarity.
69. Remove outliers using IQR method.
70. Generate random numbers from normal distribution and verify distribution.
71. Difference between:
  - `np.dot()` and `@`
  - `np.matmul()` and `np.multiply()`
72. What is `axis` parameter?
73. What is `keepdims=True`?
74. How to handle NaN values?
75. Difference between `ravel()` and `flatten()`?

## **Pandas**

1. Create a Pandas Series from a list `[10, 20, 30, 40]`.
2. Create a DataFrame from a dictionary:
  3. `{"Name": ["A", "B"], "Age": [22, 25]}`
4. Check datatype of each column.
5. Display first 5 rows.
6. Display last 3 rows.
6. Find shape of DataFrame.
7. Find column names.
8. Get summary statistics.
9. Count null values.
10. Check memory usage.
11. Select single column.
12. Select multiple columns.
13. Filter rows where `Age > 25`.
14. Filter rows with multiple conditions.
15. Select rows using `.loc` and `.iloc`.
16. Add new column "Salary".
17. Delete a column.
18. Rename columns.
19. Change datatype of column.
20. Replace values in a column.
21. Drop rows with missing values.
22. Fill missing values with mean.
23. Forward fill missing data.
24. Replace specific values.
25. Detect duplicate rows.
26. Group by single column and calculate mean.
27. Group by multiple columns.
28. Count number of records per group.

29. Apply custom function using `.apply()`.
30. Aggregate multiple functions.
31. Sort by single column.
32. Sort by multiple columns.
33. Rank values.
34. Get top 5 highest salaries.
35. Get nth largest value.
36. Merge two DataFrames on common column.
37. Left join.
38. Right join.
39. Inner join.
40. Outer join.
41. Create pivot table.
42. Create cross-tab.
43. Multi-index DataFrame.
44. Reset index.
45. Set custom index.
46. Convert string to datetime.
47. Extract year, month.
48. Filter data between two dates.
49. Resample time series data.
50. Rolling average calculation.
51. Find employee with highest salary per department.
52. Find second highest salary.
53. Remove outliers using IQR.
54. Normalize salary column.
55. Create new column based on condition.
56. Why Pandas is faster than Python loops?
57. Difference between:
  - `.loc` and `.iloc`
  - `.apply()` and `.map()`
58. Vectorization in Pandas.
59. When to use `iterrows()`?
60. Memory optimization techniques.
61. Implement rolling window manually.
62. Create custom aggregation function.
63. Convert wide format to long format.
64. Convert long to wide.
65. Detect change in values between rows.
66. Calculate moving standard deviation.
67. Calculate correlation matrix.

68. Detect seasonality.
69. Chunk large CSV file processing.
70. Optimize large dataset (10 million rows).

71. Difference between:

- Series vs DataFrame
- Copy vs View

72. What is inplace?
73. Why chained indexing is dangerous?
74. What is MultiIndex?
75. When to use pivot vs groupby?

### **matplotlib**

1. Plot a simple line graph using:
  2. `x = [1, 2, 3, 4]`
  3. `y = [10, 20, 25, 30]`
  4. Add title and axis labels.
  5. Change line color and style.
  6. Add markers to line.
  7. Add grid to plot.
6. Plot two lines on same graph.
7. Add legend.
8. Customize legend location.
9. Change figure size.
10. Save plot as image file.
11. Create vertical bar chart.
12. Create horizontal bar chart.
13. Add values on top of bars.
14. Change bar width.
15. Create grouped bar chart.
16. Create histogram of random numbers.
17. Change number of bins.
18. Add edge color.
19. Plot histogram for two datasets.
20. Normalize histogram.
21. Create scatter plot.
22. Change marker size and color.
23. Add transparency (alpha).
24. Add trend line.
25. Color points based on third variable.

26. Create 2×2 subplot grid.
27. Share axes.
28. Different plots in each subplot.
29. Adjust spacing.
30. Use `plt.tight_layout()`.
31. Change background color.
32. Customize ticks.
33. Rotate labels.
34. Add annotations.
35. Highlight specific point.
36. Create pie chart.
37. Add percentage labels.
38. Explode slice.
39. Change colors.
40. Create donut chart.
41. Dual axis plot.
42. Logarithmic scale.
43. Customize spine.
44. Remove top & right borders.
45. Custom colormap.
46. Plot moving average.
47. Compare multiple distributions.
48. Visualize correlation matrix.
49. Plot time-series data.
50. Highlight max and min points.
51. Difference between:
  - `plt.plot()` and `ax.plot()`
52. What is object-oriented approach?
53. What is backend in Matplotlib?
54. When to use seaborn instead?
55. How to improve plot readability?
56. Interactive plot using Matplotlib.
57. Animation plot.
58. Heatmap using only Matplotlib.
59. 3D plot.
60. Save plot in high resolution for publication.
61. What is `figure()`?
62. What is `axes()`?
63. What is difference between `plt.show()` and `plt.savefig()`?
64. What are rcParams?
65. Why do we use alpha?

## Seaborn

1. Create a simple line plot using Seaborn.
2. Create a bar plot from categorical data.
3. Create a count plot.
4. Load built-in dataset using:
  5. `import seaborn as sns`
  6. `sns.load_dataset("tips")`
7. Display first 5 rows of dataset.
6. Plot histogram using `histplot()`.
7. Plot KDE (Kernel Density Estimate).
8. Combine histogram + KDE.
9. Plot distribution of `total_bill` in `tips` dataset.
10. Change number of bins.
11. Create boxplot.
12. Create violin plot.
13. Compare boxplot vs violin plot.
14. Create strip plot.
15. Combine boxplot + stripplot.
16. Create scatter plot using `scatterplot()`.
17. Add hue based on category.
18. Change size based on third variable.
19. Create regression plot using `regplot()`.
20. Plot multiple regression plots using `lmplot()`.
21. Use `hue` parameter.
22. Use `col` parameter in `relplot()`.
23. Create facet grid.
24. Compare distributions by category.
25. Multi-category visualization.
26. Create correlation matrix.
27. Plot heatmap.
28. Add annotations.
29. Change colormap.
30. Mask upper triangle.
31. Pairplot.
32. Customize pairplot.
33. Jointplot.
34. Compare multiple variables.
35. Highlight correlation.
36. Plot time series data.
37. Rolling mean visualization.
38. Compare categories over time.
39. Add confidence intervals.

40. Resample time data before plotting.
41. Change theme (`darkgrid`, `whitegrid`, etc.)
42. Modify color palette.
43. Create custom palette.
44. Adjust context (`talk`, `paper`)
45. Combine multiple plots in single figure.
46. Detect outliers visually.
47. Visualize class imbalance.
48. Compare model performance visually.
49. Visualize feature importance.
50. Build dashboard-style visualization.
51. Difference between:
  - `matplotlib` and `seaborn`
52. What is `hue`?
53. What is `palette`?
54. Difference between `regplot` and `lmplot`?
55. What is `FacetGrid`?
56. Why Seaborn is used in Data Science?
57. When to use heatmap?
58. How to remove legend?
59. How to rotate x-axis labels?
60. How to increase figure size?
61. Visualize salary distribution.
62. Compare sales across regions.
63. Show correlation between features.
64. Detect skewness visually.
65. Compare two distributions side by side.