Data Analyst

Python

📘 Python Data Analysis Interview Notes (with Real-World Scenarios)

---

1. What is the role of Python in Data Analysis?

✅ Answer:

Python provides powerful libraries like pandas, numpy, matplotlib, seaborn, and scikit-learn that help in data wrangling, visualization, statistical analysis, and machine learning.

🟢 Real-World Scenario:

An e-commerce company uses Python to analyze customer purchase history and recommend products using pandas and scikit-learn.

---

2. What is the difference between a list and a NumPy array?

✅ Answer:

- Lists are Python's built-in data structures; they can store elements of different data types.
- NumPy arrays are used for numerical computation and store data of the same type, offering better performance.

🟢 Scenario:

When analyzing 1 million sales records, NumPy is preferred over lists because of its speed and lower memory usage.

---

3. Explain the difference between loc[] and iloc[] in pandas.

✅ Answer:

- loc[]: Label-based indexing.
- iloc[]: Integer-based indexing.

🟢 Scenario:

In a retail dataset, use df.loc[df['Region'] == 'East'] to filter all eastern region sales. Use df.iloc[0:5] to get the first 5 rows.

---

4. How do you handle missing data in a dataset?

✅ Answer:

- dropna() to remove missing values
- fillna() to replace them
- Interpolation methods (e.g., forward fill, mean imputation)

🟢 Scenario:
In a healthcare dataset, if patient temperature readings are missing, you might use df['Temp'].fillna(df['Temp'].mean()).

---

5. How do you merge or join datasets in pandas?

✅ Answer:
Use pd.merge() or pd.concat():
python
CopyEdit
pd.merge(df1, df2, on='CustomerID', how='inner')

🟢 Scenario:
In an online store, merge customer data with order data on CustomerID to analyze customer-wise purchases.

---

6. What is GroupBy in pandas and how is it used?

✅ Answer:
groupby() is used to split data into groups based on some criteria, apply a function, and combine the results.

🟢 Scenario:
Group sales data by region and calculate total revenue:
python
CopyEdit
df.groupby('Region')['Sales'].sum()

---

7. Explain the difference between apply(), map(), and applymap().

✅ Answer:

- map() works on Series.
- apply() works on both Series and DataFrames (for row/column-wise operations).
- applymap() works element-wise on DataFrames.

🟢 Scenario:
To convert all prices to USD in a DataFrame, use applymap() with a conversion function.

---

8. How do you detect outliers in a dataset?

✅ Answer:

- Using IQR (Interquartile Range)
- Z-score method
- Visualization: Boxplots

🟢 Scenario:
In a loan application dataset, use boxplots to detect unusually high-income entries that may be data entry errors.

---

9. How do you perform time series analysis in Python?

✅ Answer:
Use pandas datetime functionality:
python
CopyEdit
df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)
df.resample('M').sum()

🟢 Scenario:
Analyze monthly electricity consumption trends from daily meter readings using .resample().

---

10. What is the use of pivot_table() in pandas?

✅ Answer:
Creates a spreadsheet-style pivot table as a DataFrame:
python
CopyEdit
df.pivot_table(values='Sales', index='Region', columns='Product', aggfunc='sum')

🟢 Scenario:
A company wants to analyze sales by region and product category using a pivot table.

---

11. What visualization libraries do you use in Python?

✅ Answer:

- matplotlib: Basic plotting
- seaborn: Statistical visualization
- plotly: Interactive charts

🟢 Scenario:
In a financial report, use seaborn for correlation heatmaps and plotly for interactive stock price charts.

---

12. What is the purpose of using lambda functions in data analysis?

✅ Answer:
Anonymous functions used for quick calculations or transformations.

🟢 Scenario:
Add a 10% discount column using:
python
CopyEdit
df['Discounted_Price'] = df['Price'].apply(lambda x: x * 0.9)

---

13. What is the difference between a shallow copy and a deep copy in Python?

✅ Answer:

- Shallow copy: Copies the object but not nested objects.
- Deep copy: Copies the object and all nested objects.

🟢 Scenario:
While modifying a copy of a DataFrame during feature engineering, use .copy(deep=True) to avoid altering the original data.

---

14. How do you export a pandas DataFrame to Excel or CSV?

✅ Answer:

```
python
CopyEdit
df.to_csv('output.csv', index=False)
df.to_excel('output.xlsx', index=False)
```

🟢 Scenario:

After customer churn analysis, export results to Excel for a stakeholder presentation.

---

15. How do you optimize large datasets in Python?

✅ Answer:

- Use dtypes optimization (category, float32)
- Process in chunks (chunksize in pandas)
- Use libraries like Dask or Vaex for big data

🟢 Scenario:

Handling 10 million financial transactions, convert string columns to category and process in batches to reduce memory usage.

✅ Final Tips:

- Practice with datasets from Kaggle, GitHub, or your own domain.
- Showcase end-to-end projects: data cleaning → analysis → visualization → insights.
- Be ready to explain your logic and optimize code under constraints.

(Dipankar pal)
(dippal351@gmail.com)
(www.linkedin.com/in/dipankar-data-analyst)