**1.What is the purpose of using Python for data analysis?**

Python is commonly used for data analysis due to

1. Its readability, Scalability , large python developer community

2 Manily its powerful libraries such as Pandas, NumPy, and Matplotlib. These libraries provide tools for data manipulation, numerical computing, and data visualization. As libraries like Matplotlib allow data analysts to convert numbers into pie charts, graphics, histograms, etc. This makes it easier for data analysts to make their driven data visually appealing and comprehensible.

**2.How can you create a DataFrame in pandas?**

DataFrame is a data structure like a table with rows and columns. The DataFrame**()** function is used to create a DataFrame in Pandas.

**import** pandas as pd

df **=** pd.DataFrame()

print(df)

**3.Explain the difference between numpy arrays and lists in Python.**

* Lists can store any data type, in numpy arrays all the elements must have the same type.
* Lists lack specialized NumPy functions for numerical operations.
* Lists may not store elements in contiguous memory locations, causing memory fragmentation and inefficiency.
* We can append elements to a list, but you can't change the size of a ´numpy.ndarray´ without making a full copy.
* Lists in Python store additional information about each element, such as its type and reference count. This overhead can be significant when dealing with a large number of elements.

**.What are some common data visualization techniques used in matplotlib?**

Data Visualization is the process of presenting data in the form of graphs or charts. It helps in finding trends and correlations in our data by representing it pictorially. data visualization techniques used in matplotlib are:

* Line chart- is one of the basic plots and can be created using the plot() function. It is used to represent a relationship between two data X and Y on a different axis.
* Bar chart- is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. It can be created using the bar() method.
* Scatterplots**-** are used to observe relationships between variables. It can be created using the scatter() method.
* Pie chart- is a circular chart used to display only one series of data. The area of slices of the pie represents the percentage of the parts of the data. It can be created using the pie() method.

**5.How can you handle missing data in a pandas DataFrame?**

 Missing Data can also refer to as NA(Not Available) values in pandas. Pandas treat None and NaN as essentially interchangeable for indicating missing or null values. There are several useful functions for

1. Detecting-

* [isnull()](https://www.geeksforgeeks.org/python-pandas-isnull-and-notnull/)
* [notnull()](https://www.geeksforgeeks.org/python-pandas-isnull-and-notnull/)

1. Removing-

* [dropna()](https://www.geeksforgeeks.org/python-pandas-dataframe-dropna/)

1. Replacing null values-

* [replace()](https://www.geeksforgeeks.org/python-pandas-dataframe-replace/)
* [fillna()](https://www.geeksforgeeks.org/python-pandas-dataframe-fillna-to-replace-null-values-in-dataframe/)
* [interpolate()](https://www.geeksforgeeks.org/python-pandas-dataframe-interpolate/)

**6.What is the purpose of using the groupby function in pandas?**

Groupby function groups the data/rows based on the distinct values in the given column or columns. It used for grouping the data according to the categories and applying a function to the categories. it aggregate data, efficiently.

**7.Explain the concept of broadcasting in numpy.**

Broadcasting refers to the ability of NumPy to treat arrays with different dimensions during arithmetic operations. In broadcasting, the smaller array is broadcast to the larger array to make their shapes compatible with each other.

**8.How can you concatenate multiple DataFrames in pandas?**

Using the `concat` function. The `concat` function takes a sequence of DataFrames as input and concatenates them along a specified axis. We can concat two or more data frames either along rows  (axis=0) or along columns (axis=1).

Other functions are:

* Using pd.merge() specify the type of merge (inner, outer, left, or right), determining how the DataFrames are combined.
* Using dataframe.reindex**()**for vertical concatenation of DataFrames to  ensure proper stacking.

**9.What are the advantages of using seaborn over matplotlib for data visualization?**

Advantages of using seaborn over matplotlib for data visualization:

* Attractive default styles: Seaborn comes with several built-in themes and color palettes that make it easy to create aesthetically pleasing visualizations without extensive customization.
* Statistical visualization: Seaborn simplifies the process of creating complex statistical plots, such as scatter plots with regression lines, box plots, and violin plots.
* Seaborn's functions often require fewer lines of code compared to matplotlib, making it quicker to generate complex visualizations.
* Seaborn provides functions to visualize the distribution of a dataset, including histograms, KDE plots, and rug plots.
* Integration with pandas: Seaborn is designed to work seamlessly with pandas data frames, allowing for easy integration with data manipulation and analysis workflows.
* Automatic handling of missing data: Seaborn automatically handles missing data, allowing users to focus on the visualization itself rather than data preprocessing.

**10.Write a code snippet to create a scatter plot using matplotlib.**

import matplotlib.pyplot as plt  
import numpy as np  
  
x = np.array([89,43,36,36,95,10,66,34,38,20])  
y = np.array([26,34,90,33,38,20,56,2,47,15])  
  
plt.scatter(x, y)  
plt.show()