1. **What is Pandas?**
   * Pandas is an open-source data manipulation and analysis library for Python, providing data structures and functions needed to work with structured data seamlessly.
2. **What is Python pandas used for?**
   * Pandas is used for data manipulation, analysis, and cleaning. It provides tools for reading and writing data between in-memory data structures and different file formats, reshaping and pivoting datasets, and performing data alignments, aggregations, and transformations.
3. **What is a Series in Pandas?**
   * A Series in Pandas is a one-dimensional labeled array capable of holding any data type (integers, strings, floats, etc.). It is similar to a column in a DataFrame.
4. **Mention the different types of data structures in pandas?**
   * The main data structures in pandas are:
     + **Series**: One-dimensional labeled array.
     + **DataFrame**: Two-dimensional labeled data structure.
     + **Panel**: Three-dimensional labeled array (deprecated in recent versions).
5. **Explain Reindexing in pandas?**
   * Reindexing is the process of conforming a DataFrame to a new index, filling or removing missing values as necessary. It allows you to change/add/delete the index on a specified axis.
6. **What are the key features of pandas library?**
   * Key features include:
     + Data alignment and handling of missing data.
     + Flexibility in handling different types of data structures.
     + Tools for reading/writing data from various file formats.
     + Integrated time-series functionality.
     + Label-based slicing, indexing, and subsetting of large datasets.
     + Powerful group by functionality for data aggregation and transformation.
7. **What is pandas used for?**
   * Pandas is used for:
     + Data cleaning and preparation.
     + Data analysis and exploration.
     + Data modeling and transformation.
     + Reading and writing data in different formats (CSV, Excel, SQL databases, etc.).
8. **How can we create a copy of series in Pandas?**
   * You can create a copy of a Series using the **copy()** method:

python

Copy code

series\_copy = original\_series.copy()

1. **What is Time Series in pandas?**
   * Time Series in pandas refers to a sequence of data points indexed in time order. It provides tools for handling date and time data, performing resampling, and generating rolling statistics.
2. **Explain Categorical Data in Pandas?**
   * Categorical data in pandas refers to variables that can take on a limited, and usually fixed, number of possible values (categories). Pandas provides a specific data type **Categorical** for such data, which can improve performance and memory usage.
3. **How will you create a series from dict in Python?**
   * You can create a Series from a dictionary using the **pd.Series()** constructor:

python

Copy code

data = {'a': 1, 'b': 2, 'c': 3} series = pd.Series(data)

1. **What are operations on Series in pandas?**
   * Operations on Series include:
     + Element-wise operations: addition, subtraction, multiplication, etc.
     + Statistical operations: mean, median, mode, etc.
     + Indexing and slicing.
     + Handling missing data.
2. **What is a DataFrame in pandas?**
   * A DataFrame is a two-dimensional, size-mutable, and potentially heterogeneous tabular data structure with labeled axes (rows and columns). It is similar to a spreadsheet or SQL table.
3. **What are the different ways in which a DataFrame can be created in Pandas?**
   * DataFrames can be created from:
     + Dictionaries of lists/arrays.
     + Lists of dictionaries.
     + 2D numpy arrays.
     + Series.
     + Reading data from files (CSV, Excel, SQL databases).
4. **How will you create an empty DataFrame in pandas?**
   * You can create an empty DataFrame using:

python

Copy code

empty\_df = pd.DataFrame()

1. **How will you add a column to a pandas DataFrame?**
   * You can add a column to a DataFrame by assigning values to a new column name:

python

Copy code

df['new\_column'] = [value1, value2, value3]

1. **How will you retrieve a single column from pandas DataFrame?**
   * You can retrieve a single column by selecting it with its column name:

python

Copy code

column = df['column\_name']

1. **range() vs xrange() functions in Python?**
   * In Python 2, **range()** returns a list, whereas **xrange()** returns an xrange object, which generates numbers on demand and is more memory efficient. In Python 3, **range()** behaves like **xrange()** in Python 2, and **xrange()** is not available.
2. **What is the name of pandas library tools used to create a scatter plot matrix?**
   * The **scatter\_matrix** function from **pandas.plotting** is used to create a scatter plot matrix.
3. **What is pylab?**
   * **pylab** is a module in Matplotlib that combines pyplot and numpy into a single namespace. It's designed to mimic MATLAB's functionality but is generally not recommended for use due to namespace pollution.
4. **Define the different ways a DataFrame can be created in pandas?**
   * DataFrames can be created from:
     + Dictionaries of lists/arrays.
     + Lists of dictionaries.
     + 2D numpy arrays.
     + Series.
     + Reading data from files (CSV, Excel, SQL databases).
5. **Explain Categorical data in Pandas?**
   * Categorical data in pandas refers to data that can take on a limited number of categories. It is stored in the **Categorical** data type, which can help with memory usage and performance for large datasets with repeating values.
6. **How will you create a series from dict in Pandas?**
   * You can create a Series from a dictionary using the **pd.Series()** constructor:

python

Copy code

data = {'a': 1, 'b': 2, 'c': 3} series = pd.Series(data)

1. **How can we create a copy of the series in Pandas?**
   * You can create a copy of a Series using the **copy()** method:

python

Copy code

series\_copy = original\_series.copy()

1. **How will you create an empty DataFrame in Pandas?**
   * You can create an empty DataFrame using:

python

Copy code

empty\_df = pd.DataFrame()

1. **How will you add a column to a pandas DataFrame?**
   * You can add a column to a DataFrame by assigning values to a new column name:

python

Copy code

df['new\_column'] = [value1, value2, value3]

1. **How to add an Index, row, or column to a Pandas DataFrame?**
   * **Add Index**: Use **set\_index** or assign a new index directly.

python

Copy code

df.index = ['index1', 'index2']

* + **Add Row**: Use **loc** or **append**.

python

Copy code

df.loc[new\_index] = [value1, value2] df = df.append(new\_row)

* + **Add Column**: Assign values to a new column name.

python

Copy code

df['new\_column'] = [value1, value2]

1. **How to delete indices, rows, or columns from a Pandas DataFrame?**
   * **Delete Indices**: Use **drop**.

python

Copy code

df = df.drop(['index1', 'index2'])

* + **Delete Rows**: Use **drop**.

python

Copy code

df = df.drop([0, 1])

* + **Delete Columns**: Use **drop** with **axis=1**.

python

Copy code

df = df.drop(['column1', 'column2'], axis=1)

1. **How to rename the index or columns of a Pandas DataFrame?**
   * Use the **rename** method:

python

Copy code

df = df.rename(index={'old\_index': 'new\_index'}, columns={'old\_column': 'new\_column'})

1. **How to iterate over a Pandas DataFrame?**
   * You can iterate using **iterrows()** for rows or **itertuples()** for named tuples:

python

Copy code

for index, row in df.iterrows(): print(row['column1'], row['column2'])

1. **How to get the items of series A not present in series B?**
   * Use the **~** operator and **isin** method:

python

Copy code

result = A[~A.isin(B)]

1. **How to get the items not common to both series A and series B?**
   * Use **symmetric\_difference** with sets:

python

Copy code

result = pd.Series(list(set(A).symmetric\_difference(set(B))))

1. **How to get the minimum, 25th percentile, median, 75th, and max of a numeric series?**
   * Use the **quantile** method:

python

Copy code

min\_val = series.min() q25 = series.quantile(0.25) median = series.median() q75 = series.quantile(0.75) max\_val = series.max()

1. **How to get frequency counts of unique items of a series?**
   * Use the **value\_counts** method:

python

Copy code

counts = series.value\_counts()

1. **How to convert a numpy array to a dataframe of given shape?**
   * Use **pd.DataFrame** with **reshape**:

python

Copy code

df = pd.DataFrame(array.reshape(shape))

1. **How can we convert a Series to DataFrame?**
   * Use **to\_frame** method:

python

Copy code

df = series.to\_frame(name='column\_name')

1. **How can we sort the DataFrame?**
   * Use **sort\_values** for sorting by values or **sort\_index** for sorting by index:

python

Copy code

df = df.sort\_values(by='column\_name') df = df.sort\_index()

1. **How to convert String to date?**
   * Use **pd.to\_datetime**:

python

Copy code

df['date\_column'] = pd.to\_datetime(df['date\_string'])

1. **What is Data Aggregation?**
   * Data aggregation involves summarizing data by applying aggregate functions like sum, mean, count, etc., often grouped by certain fields.
2. **What is Pandas Index?**
   * The index in pandas is the label or integer-based identifier for rows (or columns), allowing for easy data selection and alignment.
3. **Define ReIndexing?**
   * Reindexing changes the row (or column) labels and can also change the order of the data.
4. **Define Multiple Indexing?**
   * Multiple indexing, or MultiIndex, allows hierarchical indexing of rows or columns, providing multi-level or tree-like structures for more complex data manipulation.
5. **How to set the index?**
   * Use **set\_index**:

python

Copy code

df = df.set\_index('column\_name')

1. **How to reset the index?**
   * Use **reset\_index**:

python

Copy code

df = df.reset\_index()

1. **Describe Data Operations in Pandas?**
   * Data operations include filtering, grouping, merging, joining, aggregation, reshaping, and manipulating datasets for analysis.
2. **Define GroupBy in Pandas?**
   * GroupBy involves splitting the data into groups based on some criteria, applying a function to each group, and then combining the results.
3. **How will you append new rows to a pandas DataFrame?**
   * Use **append** or **loc**:

python

Copy code

df = df.append(new\_row, ignore\_index=True) df.loc[new\_index] = [value1, value2]

1. **How will you delete rows from a pandas DataFrame?**
   * Use **drop**:

python

Copy code

df = df.drop(index)

1. **How will you get the number of rows and columns of a DataFrame in pandas?**
   * Use **shape** attribute:

python

Copy code

rows, cols = df.shape

1. **What is Pandas ml?**
   * Pandas-ml is a library integrating pandas with machine learning libraries such as scikit-learn, providing an easy interface for machine learning operations.
2. **What is Pandas Charm?**
   * There is no widely recognized library or tool called "Pandas Charm." It might be a misunderstanding or confusion with another tool.
3. **How will you add a scalar column with the same value for all rows to a pandas DataFrame?**
   * Assign a scalar value to a new column:

python

Copy code

df['new\_column'] = scalar\_value

1. **How can we select a column in pandas DataFrame?**
   * Select a column using its name:

python

Copy code

column = df['column\_name']

1. **How can we retrieve a row in pandas DataFrame?**
   * Retrieve a row using **loc** or **iloc**:

python

Copy code

row = df.loc[row\_index] row = df.iloc[row\_number]

1. **How will you convert a DataFrame to an array in pandas?**
   * Use **values** attribute or **to\_numpy** method:

python

Copy code

array = df.values array = df.to\_numpy()

1. **How can you check if a DataFrame is empty in pandas?**
   * Use **empty** attribute:

python

Copy code

is\_empty = df.empty

1. **How can you get the sum of values of a column in pandas DataFrame?**
   * Use **sum** method:

python

Copy code

total\_sum = df['column\_name'].sum()

1. **How will you get the average of values of a column in pandas DataFrame?**
   * Use **mean** method:

python

Copy code

average = df['column\_name'].mean()

1. **How will you apply a function to every data element in a DataFrame?**
   * Use **applymap**:

python

Copy code

df = df.applymap(function)

1. **How will you get the top 2 rows from a DataFrame in pandas?**
   * Use **head** method:

python

Copy code

top\_rows = df.head(2)

1. **List major features of the Python pandas?**
   * Major features include:
     + Data alignment and handling of missing data.
     + Flexible reshaping and pivoting of datasets.
     + Label-based slicing, indexing, and subsetting.
     + Grouping data for aggregation and transformations.
     + High performance for large datasets.
2. **Enlist different types of Data Structures available in Pandas?**
   * The main data structures are:
     + **Series**: One-dimensional labeled array.
     + **DataFrame**: Two-dimensional labeled array.
     + **Panel**: Three-dimensional labeled array (deprecated).
3. **How to write a Pandas DataFrame to a file?**
   * Use functions like **to\_csv**, **to\_excel**, **to\_sql**, etc.:

python

Copy code

df.to\_csv('file.csv') df.to\_excel('file.xlsx')

1. **When, why, and how you should reshape your Pandas DataFrame?**
   * Reshaping is needed when you want to transform the structure of your DataFrame to suit analysis requirements. Methods include **pivot**, **melt**, **stack**, **unstack**.
2. **Does Pandas recognize dates when importing data?**
   * Pandas can recognize dates when importing data by specifying the **parse\_dates** parameter:

python

Copy code

df = pd.read\_csv('file.csv', parse\_dates=['date\_column'])

1. **How to format the data in your Pandas DataFrame?**
   * Use methods like **astype** for type conversion, **apply** for element-wise operations, and formatting functions for strings and dates.
2. **How to add an Index, Row, or Column to a Pandas DataFrame?**
   * Add index: Use **set\_index**.
   * Add row: Use **loc** or **append**.
   * Add column: Assign values to a new column name.
3. **How to select an Index or Column From a Pandas DataFrame?**
   * Select index: Use **loc** or **iloc**.
   * Select column: Use the column name.
4. **How will you get the average of values of a column in pandas DataFrame?**
   * Use **mean** method:

python

Copy code

average = df['column\_name'].mean()

1. **How to apply function to every row in a Pandas DataFrame?**
   * Use **apply** with **axis=1**:

python

Copy code

df = df.apply(function, axis=1)

1. **What is the use of GroupBy objects in Pandas?**
   * GroupBy objects are used for splitting data into groups based on some criteria, applying a function to each group, and then combining the results.
2. **What is Pandas NumPy?**
   * Pandas and NumPy are complementary libraries. Pandas is built on top of NumPy, providing data structures like Series and DataFrame that are more suitable for data manipulation and analysis.
3. **What is Vectorization in Python pandas?**
   * Vectorization refers to performing operations on entire arrays or DataFrames without explicit loops, resulting in faster execution.
4. **List some alternatives of Python Pandas?**
   * Alternatives include:
     + **Dask**: For parallel computing with pandas-like interface.
     + **Vaex**: For out-of-core DataFrames.
     + **Koalas**: For pandas-like API on Apache Spark.
     + **Modin**: For distributed DataFrame computation.
5. **How to convert a DataFrame to an array in Pandas?**
   * Use **values** attribute or **to\_numpy** method:

python

Copy code

array = df.values array = df.to\_numpy()

1. **List some statistical functions in Python Pandas?**
   * Common statistical functions include:
     + **mean()**
     + **median()**
     + **mode()**
     + **std()**
     + **var()**
     + **min()**
     + **max()**
     + **quantile()**
2. **Explain Series in pandas. How to create a copy of Series in pandas?**
   * A Series is a one-dimensional labeled array. To create a copy, use the **copy** method:

python

Copy code

series\_copy = original\_series.copy()

1. **What are the different ways a DataFrame can be created in pandas?**
   * DataFrames can be created from:
     + Dictionaries of lists/arrays.
     + Lists of dictionaries.
     + 2D numpy arrays.
     + Series.
     + Reading data from files (CSV, Excel, SQL databases).