Nitya CloudTech
Dream.Achieve.Succeed

Data Analyst Interview Scenarios

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How would you create a new column based on the year of a datetime column? If it's this year, assign 0; if it's last year, assign 1.

Candidate:

Candidate: You can use apply() to extract the year and compare it to the current year.

```
import datetime
current_year = datetime.datetime.now().year
df['YearFlag'] = df['Date'].apply(lambda x: 0 if x.year == current_year else 1)
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How would you handle the situation where you need to convert a string column representing dates in yyyy-mm-dd format into actual datetime objects?

Candidate:

You can use pd.to_datetime() to convert the string column into datetime objects.

```
df['Date'] = pd.to_datetime(
df['Date'], format='%Y-%m-%d')
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How can you create a new column in a DataFrame based on a condition from another column? For example, if the value in column 'A' is greater than 10, assign 'High' to a new column, otherwise assign 'Low'.

Candidate:

You can use the apply function or np.where for this.

```
df['NewColumn'] =
df['A'].apply(lambda x: 'High' if x > 10 else 'Low')
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How can you merge two DataFrames on multiple columns with different names, while keeping all the data from both DataFrames?

Candidate:

We can use the merge() function with the left_on and right_on parameters to specify different column names for merging.

```
df1.merge(df2, left_on=['col1', 'col2'],
right_on=['colA', 'colB'], how='outer')
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: What would you do if you want to drop rows that have NaN values in a specific subset of columns, but not in the entire DataFrame?

Candidate:

We can use the dropna() method with the subset parameter to specify the columns.

df.dropna(subset=['column1', 'column2'], inplace=True)

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How can you create a pivot table to find the average of 'Sales' grouped by 'Region' and 'Month', and also include a column that counts the number of transactions?

Candidate:

You can use the pivot_table() function with multiple aggregation functions

```
pivot = df.pivot_table(values=['Sales'],
index=['Region', 'Month'],
aggfunc={'Sales': 'mean', 'TransactionID':'count'})
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How would you apply a function that uses multiple columns in a DataFrame and returns a new column, say, summing two columns, 'A' and 'B'?

Candidate:

You can use apply() with axis=1 to apply a function across rows.

```
df['SumAB'] = df.apply(lambda row: row['A'] + row['B'], axis=1)
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How would you filter rows in a DataFrame where a specific column contains values from a list of options?

Candidate:

You can use the isin() method to filter rows

```
df_filtered = df[df['Category'].isin(['A', 'B', 'C'])]
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How can you group the DataFrame by one column and apply a custom aggregation function to another column?

Candidate:

You can use the groupby() method and then apply the custom aggregation function using agg()

df.groupby('Category')['Amount'].agg(lambda x: x.max() - x.min())

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: If you want to calculate the rolling mean of a column with a window size of 7, how would you do it?

Candidate:

You can use the rolling() function followed by mean().

```
df['RollingMean'] =
df['Sales'].rolling(window=7).mean()
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How would you handle duplicates in a DataFrame and keep only the first occurrence of each duplicate row?

Candidate:

You can use the drop_duplicates() method and specify keep='first' to keep the first occurrence.

```
df.drop_duplicates(keep='first', inplace=True)
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How can you handle a situation where you have a multi-index DataFrame and you need to reset it back to a flat DataFrame?

Candidate:

You can use reset_index() to flatten a multi-index DataFrame.

```
df_reset = df.reset_index()
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How can you get the top N rows for each group in a DataFrame?

Candidate:

You can use groupby() with head()

df.groupby('Category').head(3)

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How do you perform a cross join between two DataFrames?

Candidate:

You can use merge() with an artificial key (like key=1) to perform a cross join.

```
df1['key'] = 1
df2['key'] = 1
df_cross = pd.merge(df1, df2, on='key').drop('key', axis=1)
```

Nitya CloudTech

Dream.Achieve.Succeed

Interviewer: How would you count the number of occurrences of each value in a column?

Candidate:

You can use the value_counts() method.

```
df['Category'].value_counts()
```

Nitya CloudTech
Dream.Achieve.Succeed

Interviewer: How can you filter rows where a string column contains a specific substring?

Candidate:

Candidate: You can use the str.contains() method for this.

```
df_filtered = df[df['ProductName'].str.contains('Laptop')]
```

Nitya CloudTech

Dream.Achieve.Succeed

Interviewer: How would you handle missing values in a column and replace them with the mean of that column?

Candidate:

Candidate: You can use fillna() to replace missing values

```
df['Column'] =
df['Column'].fillna(df['Column'].mean())
```

Nitya CloudTech

Dream.Achieve.Succeed

FOR CAREER GUIDANCE, CHECK OUT OUR PAGE

www.nityacloudtech.com

Follow Us on Linkedin:

Aditya Chandak

Nitya CloudTech

Dream.Achieve.Succeed