import pandas as pd

from pandas.tseries.offsets import MonthEnd

def fill\_missing\_rates(df, rate\_columns):

# Ensure the DataFrame is sorted by date

df = df.sort\_values(by='date')

for rate\_col in rate\_columns:

# Iterate over the DataFrame rows

for i in range(1, len(df)):

# If the rate is NaN

if pd.isna(df.at[i, rate\_col]):

# Get the current and previous month's end dates

current\_month\_end = df.at[i, 'date'] + MonthEnd(0)

previous\_month\_end = df.at[i, 'date'] - MonthEnd(1)

# Try to get the rate for the current and previous month's end

try:

current\_month\_rate = df[df['date'] == current\_month\_end][rate\_col].values[0]

previous\_month\_rate = df[df['date'] == previous\_month\_end][rate\_col].values[0]

# Calculate the number of days in the current month

days\_in\_month = current\_month\_end.day

# Calculate the rate for the current date

df.at[i, rate\_col] = ((df.at[i, 'date'].day / days\_in\_month) \* current\_month\_rate +

((days\_in\_month - df.at[i, 'date'].day) / days\_in\_month) \* previous\_month\_rate)

except IndexError:

# Handle cases where the rate for the start or end of the period is missing

pass

return df

# Example usage

data = {

'date': pd.date\_range(start='2023-10-01', end='2023-11-30', freq='D'),

'US\_rate': [None] \* 61,

'Japan\_rate': [None] \* 61,

'China\_rate': [None] \* 61

}

# Assuming rates for the end of October and November for demonstration

data['US\_rate'][-1] = 100

data['US\_rate'][-30] = 110

data['Japan\_rate'][-1] = 200

data['Japan\_rate'][-30] = 210

data['China\_rate'][-1] = 300

data['China\_rate'][-30] = 310

df = pd.DataFrame(data)

rate\_columns = ['US\_rate', 'Japan\_rate', 'China\_rate']

df\_filled = fill\_missing\_rates(df, rate\_columns)

print(df\_filled.head(10)) # Display the first 10 rows