import pandas as pd

# Ensure datetime format

daily\_reviews\_copy['newtime'] = pd.to\_datetime(daily\_reviews\_copy['newtime'])

# Drop the year column if it exists

if 'year' in daily\_reviews\_copy.columns:

    daily\_reviews\_copy = daily\_reviews\_copy.drop(columns=['year'])

# Filter for the period 2013–2021

mask = (daily\_reviews\_copy['newtime'].dt.year >= 2013) & (daily\_reviews\_copy['newtime'].dt.year <= 2021)

df\_filtered = daily\_reviews\_copy.loc[mask].copy()

# Set index as date

df\_filtered = df\_filtered.set\_index('newtime')

# Reindex to have continuous daily data (NaN for missing dates)

all\_days = pd.date\_range(start="2013-01-01", end="2021-12-31", freq='D')

df\_full = df\_filtered.reindex(all\_days)

# Fill missing review\_count with rolling mean (7-day window)

df\_full['review\_count'] = df\_full['review\_count'].fillna(

    df\_full['review\_count'].rolling(7, min\_periods=1).mean()

)

# Handle remaining NaNs (edges or long empty gaps)

df\_full['review\_count'] = df\_full['review\_count'].fillna(method='ffill')

df\_full['review\_count'] = df\_full['review\_count'].fillna(method='bfill')

# Reset index back to column

df\_full = df\_full.reset\_index().rename(columns={'index': 'date'})

# Check if any NaNs remain

print("Remaining NaNs per column:")

print(df\_full.isna().sum())

print(df\_full.head(10))

print(df\_full.tail(10))