Natural Gas Price Estimator Script

This document contains the Python code to estimate natural gas prices based on historical data. The script performs data analysis, interpolation, and extrapolation, and includes example test cases.

# Python Code

import pandas as pd  
import numpy as np  
from scipy.interpolate import interp1d  
import matplotlib.pyplot as plt  
from datetime import datetime  
  
# Load the dataset  
file\_path = 'Nat\_Gas.csv' # Change this to your file path if needed  
gas\_data = pd.read\_csv(file\_path)  
  
# Convert 'Dates' column to datetime format  
gas\_data['Dates'] = pd.to\_datetime(gas\_data['Dates'])  
  
# Plotting the historical price data  
plt.figure(figsize=(12, 6))  
plt.plot(gas\_data['Dates'], gas\_data['Prices'], marker='o', linestyle='-')  
plt.title('Natural Gas Prices Over Time')  
plt.xlabel('Date')  
plt.ylabel('Price ($)')  
plt.grid(True)  
plt.show()  
  
# Prepare data for interpolation  
dates\_num = gas\_data['Dates'].map(pd.Timestamp.toordinal)  
prices = gas\_data['Prices']  
  
# Create interpolation function  
interp\_func = interp1d(dates\_num, prices, kind='linear', fill\_value="extrapolate")  
  
# Function to estimate price for any given date  
def estimate\_gas\_price(input\_date\_str):  
 input\_date = pd.to\_datetime(input\_date\_str)  
 max\_date = gas\_data['Dates'].max() + pd.DateOffset(years=1)  
 if input\_date > max\_date:  
 return "Date is beyond the extrapolation limit of one year."  
 price\_estimate = float(interp\_func(input\_date.toordinal()))  
 return round(price\_estimate, 2)  
  
# Example test cases  
test\_dates = ['2021-06-30', '2024-12-31', '2025-10-01']  
print("Natural Gas Price Estimates:")  
for date in test\_dates:  
 print(f"{date}: {estimate\_gas\_price(date)}")  
  
# Note for documentation  
note = """  
Note: This role demands a strong foundation in data analysis and machine learning. Python is a critical tool,   
extensively used at JPMorgan Chase—especially in quantitative research—to perform advanced computations,   
analyze vast datasets, and build robust predictive models for informed decision-making.  
"""  
print(note)

# Note

This role demands a strong foundation in data analysis and machine learning. Python is a critical tool, extensively used at JPMorgan Chase—especially in quantitative research—to perform advanced computations, analyze vast datasets, and build robust predictive models for informed decision-making.