Código Computacional para Cálculo de Preço de Gasolina, Óleo Diesel e Derivados de Petróleo - Algortimo Modelo AEPET/UNILA

#Universidade Federal Da Integração Latino-americana - UNILA
#Instituto Latino Americano de Tecnologia, Insfraestrutura e Território - ILATIT
#Grupo de Pesquisa em Mobilidade e Matriz Energética - GPMME
#
#Código Computacional para Cálculo de Preço de Óleo Diesel e Outros Derivados de Petróleo - Algortimo Modelo AEPET/UNILA
#
#Autor: Delman Lopes Cardeli Jr. – Acadêmico Engenharia de Energia – ILATIT/UNILA
#email: dlc.junior.2018@aluno.unila.edu.br
#
#Revisão: Prof. Ricardo Hartmann - Engenharia de Energia – ILATIT/UNILA
#email: ricardo.hartmann@unila.edu.br
#

********** Início do Código Computacional **********
#a partir da linha em branco abaixo abaixo, o código pode ser copiado integralmente e colado no #ambiente de programação *Python*
#
#bibliotecas
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import sqlalchemy
import requests
from bs4 import BeautifulSoup
import os
import csv
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import sqlalchemy
import requests
from bs4 import BeautifulSoup
import os
import csv
```

```
from datetime import date, timedelta
import plotly.express as px


# valores do barril brent
caminho = r'C:\\Users\\rafin\\OneDrive\\Área de Trabalho\\UNILA\\20221\\Petroleo\\Brent\\'
arquivo = r'RBRTEd.xls'
p_brent= pd.read_excel(caminho + arquivo,sheet_name='Data 1',header=3, names=['Date','Brent_Price','Week'])
p_brent['Date'] = pd.to_datetime(p_brent['Date'], format='%Y-%m-%d')
plt.title("Histórico do preço do barril Brent")
plt.plot(p_brent['Date'], p_brent['Brent_Price'], color='red')
last_value_oil = p_brent["Brent_Price"].iat[-1]
last_value_date=p_brent["Date"].max()
print("último registro do preço do barril:",(last_value_date),float(last_value_oil))
plt.show()

#valores ppi para refinaria de araucária
caminho1 = r'C:\\Users\\rafin\\OneDrive\\Área de Trabalho\\UNILA\\20221\\Petroleo\\PPI\\'
arquivo1 = r'ppi.xlsx'
ppi_araucaria = pd.read_excel(caminho1 + arquivo1, sheet_name='Diesel R$ semanal',header=1,usecols=("B:D"), names=['Price','Date','Week'])
ppi= ppi_araucaria[['Date','Price']]
ppi_araucaria['Date'] = pd.to_datetime(ppi_araucaria['Date'], format='%Y-%m-%d')

# juntar o valor do brent e do ppi para que forme um data frame
df = pd.merge(ppi_araucaria, p_brent, on='Date', how='outer')


# eliminar dados invalidos do dataframe
df = df.dropna(subset=['Date', 'Price'])

#imprimir as ultimas linhas do dataframe
print(df.tail())

# Cálculo do custo de internação
internacao = 0.0533
ano_atual = 2023
ano_base = 2017
aumento_ano = 0.02
anos_passados = ano_atual - ano_base
i_atual = internacao + (aumento_ano * anos_passados)
print("O custo de internação atual é de:", i_atual)

#Calculo para custo de produção
custo_base = 1.09
```

```
ano_atual1 = 2023
ano_base1 = 2018
aumento_ano1 = 0.02
anos_passados1 = ano_atual1 - ano_base1
custo_atual = custo_base + (aumento_ano1 * anos_passados1)
print("O custo de produção atual é de:", custo_atual)


# fatores de amortecimento
dfa_1 = ppi_araucaria['Price'] - i_atual
dfa_2 = -(1/2) * ((p_brent['Brent_Price'] - 30)/(75 - 30)) * (ppi_araucaria['Price']- custo_atual)
dfa_2m = -(1/2) * ((75 - 30)/(75 - 30)) * (ppi_araucaria['Price']- custo_atual)
dfa_3 = -(1/3) * ((p_brent['Brent_Price']-75)/(120 - 75)) * (ppi_araucaria['Price'] - custo_atual*(1
+ 0.09))


# limitar o intervalo de prltr de 2018 a 2023
mask = (df['Date'] >= pd.to_datetime('2018-01-01')) & (df['Date'] <= pd.to_datetime('2023-03-
01'))
prltr_list = []
prltr_df = pd.DataFrame({'Date': df.loc[mask, 'Date'], 'PRLTR': np.nan})


for index, row in df.loc[mask].iterrows():

    if row['Brent_Price'] <= 30:
        prltr = dfa_1
    elif row['Brent_Price'] <= 75:
        prltr = dfa_1 + dfa_2
    else:
        prltr = dfa_1 + dfa_2m + dfa_3


     # acumular os valores de prltr em uma lista agrupar valores para o mesmo mes e fazer uma
média
    prltr_list.append(prltr)

    # eliminar valores inválidos
    prltr_df['PRLTR'].fillna(method='ffill', inplace=True)

    # preencher o valor de PRLTR na linha correspondente do DataFrame
    prltr_df.loc[prltr_df['Date'] == row['Date'], 'PRLTR'] = prltr_list[-1]


# plotar o gráfico final apenas de linha com os valores de PRLTR para a iteração
plt.plot(prltr_df['Date'], prltr_df['PRLTR'])
plt.title('estimativa preço Diesel 2018 a 2023')
```

```
plt.xlabel('Data')
plt.ylabel('preço por litro')
plt.show()

# mostrar ultimo valor do diesel válido e a data
print("O valor do diesel mais recente: ", prltr_df['Date'].iat[-1], prltr_df['PRLTR'].iat[-1])

# criar uma cópia do dataframe df
df_cop = df.copy()

# renomear a coluna 'PRLTR' do dataframe prltr_df para 'Preço por litro'
prltr_df.rename(columns={'PRLTR': 'Preço por litro'}, inplace=True)

# juntar os dataframes df_copy e prltr_df com base na coluna 'Date'
merged_df = df_cop.merge( prltr_df, on=['Date'])

# Exportar o dataframe merged_df
merged_df.to_excel('dados_e_preco.xlsx', index=False)
```