fuel_cleaning

May 14, 2021

[ ]:
```
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
import datetime
from datetime import datetime as dt
```

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[ ]:
```
%cd '/content/drive/MyDrive/CSE544_PROJECT'
%ls -l
```

```
/content/drive/.shortcut-targets-by-
id/1YQyVsZWGB7sACOZzGllQA0QwFc_E5Nb1/CSE544_PROJECT
total 1007
-rw------- 1 root root  13113 May  8 15:12  14.csv
-rw------- 1 root root   9808 May 10 08:37 '2a EWMA.ipynb'
-rw------- 1 root root 218894 May 11 02:37  2c.ipynb
-rw------- 1 root root  10403 May  8 19:17  August.csv.xlsx
-rw------- 1 root root   4319 May  8 21:23  August_Final.csv
-rw------- 1 root root  27056 May 10 05:29  clean.csv
-rw------- 1 root root  19894 May 10 03:40  CSE544_PROJECT.ipynb
-rw------- 1 root root  10672 May 10 22:46  fuel_clean.csv
-rw------- 1 root root  49328 May 11 03:19  fuel_cleaning.ipynb
-rw------- 1 root root   5099 May 10 22:09  fuel_unclean.csv
-rw------- 1 root root  14155 May  8 21:06  OCT_NOV_DEC.xlsx
-rw------- 1 root root  29958 May 10 20:39  post-cleaning.ipynb
-rw------- 1 root root  20198 May  8 15:35  sample.csv
-rw------- 1 root root  19754 May 10 08:34  SNEH_clean.csv
-rw------- 1 root root 153968 May 10 08:44  Sneh_trial.ipynb
-rw------- 1 root root   3807 May  8 22:40  temp2.csv
-rw------- 1 root root   3819 May  8 23:22  temp3.csv
-rw------- 1 root root   3814 May 10 00:18  temp.csv
-rw------- 1 root root  10849 May 10 23:30  USA_clean.csv
-rw------- 1 root root 166320 May 11 03:18  USA_cleaning.ipynb
```

```
-rw------- 1 root root 131790 May 10 22:49  US_confirmed.csv
-rw------- 1 root root  99103 May 10 23:04  US_deaths.csv
```

[ ]:
```
df = pd.read_csv('fuel_unclean.csv')
df.head()
```

[ ]:

	Date	Jet Fuel Spot Price
0	1/22/2020	1.711
1	1/23/2020	1.702
2	1/24/2020	1.665
3	1/27/2020	1.598
4	1/28/2020	1.641

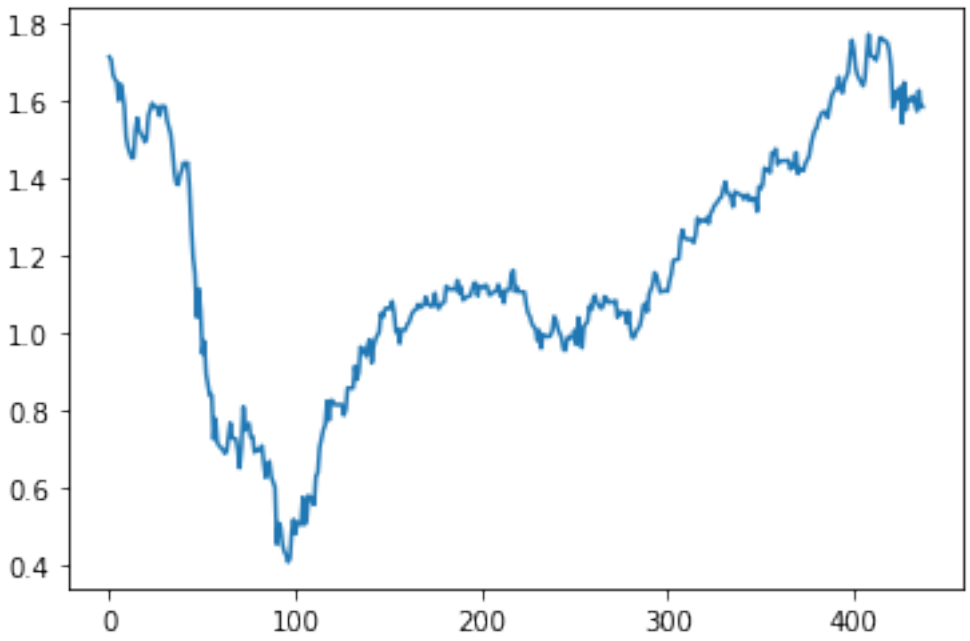
[ ]:
```
# filling in the missing values for the price with the median from a 10 day␣
 ↪window around the missing day

date = np.array(df['Date'])
price = np.array(df['Jet Fuel Spot Price'])
price_list = price.tolist()

# plt.plot(price)

start_date = datetime.datetime(2020, 1, 22)
end_date = datetime.datetime(2021, 4, 3)
delta = datetime.timedelta(days=1)

date_formatted = []
for x in date:
  date_format = dt.strptime(x, "%m/%d/%Y")
  date_formatted.append(date_format)



  # np.insert(date_formatted, dt.strptime(date, "%M/%d/%Y"))

while start_date <= end_date:
    if start_date not in date_formatted:
      index = np.searchsorted(date_formatted, start_date, side='right')
      # np.insert(date, index, start_date)
      date_formatted.insert(index, start_date)
      median = np.median(price_list[index-2:index+2])
      price_list.insert(index, median)
    start_date += delta

price_list.pop()
date_formatted.pop()
print(len(date_formatted))
```

```
print(len(price_list))
```

```
438
438
```

[ ]:
```
plt.plot(price_list)
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x7f1932522050>]
```

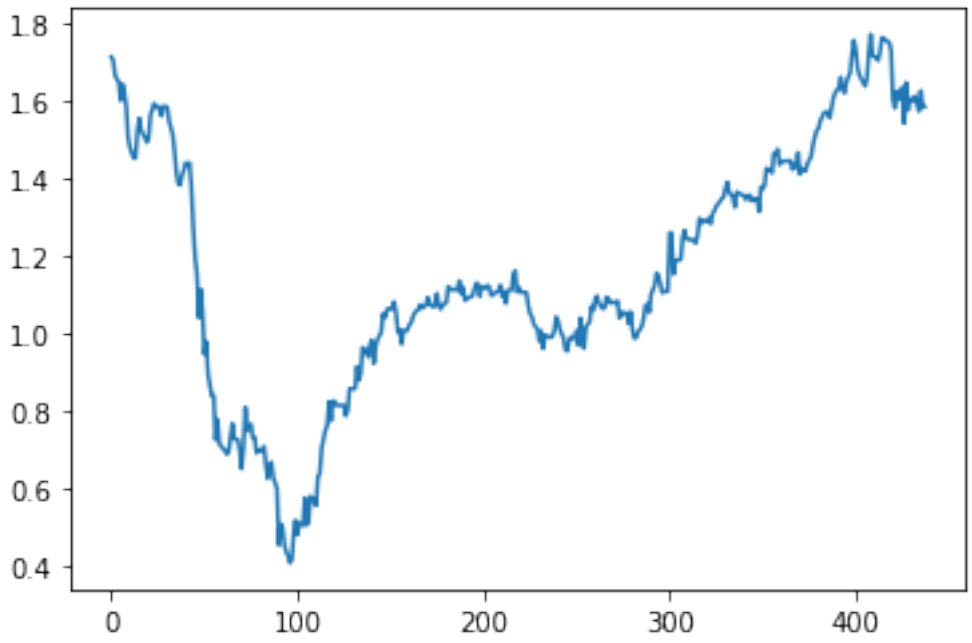


[ ]:
```
# apply tukeys rule for removing outliers

import statistics

def tukey(price_list):
  month_price_list = []
  # lst1 = price_list[i:i+30] for i in range(0,len(df)-30+1,30)]
  for i in range(0,len(price_list)-30+1,30):
    month_price_list.append(price_list[i:i+30])
  month_price_list.append(price_list[420:])
  price_list_tukey = []
  for month in month_price_list:
    median = statistics.median(month)
    month_sorted = np.sort(month)
    q25 = month_sorted[math.ceil((25/100)*len(month))-1]
    q75 = month_sorted[math.ceil((75/100)*len(month))-1]
    iqr = q75 - q25
```

```
    cut_off = iqr * 1.5
    lower, upper = q25 - cut_off, q75 + cut_off
    numchanges = 0
    for i, x in enumerate(month):
        if x < lower or x > upper:
            month[i] = median
            numchanges += 1
    print("outliers = ", numchanges)
    price_list_tukey.extend(month)
    # plt.plot(price_list_tukey)
  return price_list_tukey


price_list = tukey(price_list)
plt.plot(price_list)
```

```
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  0
outliers =  2
outliers =  0
outliers =  0
outliers =  0
outliers =  1
```

[ ]: [<matplotlib.lines.Line2D at 0x7f19326ac610>]



[ ]:
```
dict = {'Date': date_formatted, 'Price': price_list}
df1 = pd.DataFrame(dict)
```

[ ]:
```
df1.to_csv('fuel_clean.csv')
```

[ ]:
```
df = pd.read_csv('fuel_clean.csv')
df.head()
```

[ ]:

	Unnamed: 0	Date	Price
0	0	2020-01-22	1.711
1	1	2020-01-23	1.702
2	2	2020-01-24	1.665
3	3	2020-01-25	1.653
4	4	2020-01-26	1.647

[ ]:
```
plt.plot(df['Date'], df['Price'])
plt.xticks(np.arange(df['Date'][0]), max(x)+1, 1.0))
```

```
  File "<ipython-input-10-707191cb525f>", line 2
    plt.xticks(np.arange(df['Date'][0]), max(x)+1, 1.0))
                                                       ^
SyntaxError: invalid syntax
```

[ ]:
```
print(type(df['Date'][0]))
```

```
<class 'str'>
```

[ ]: