import necessary libraries

In [2]:

```
import datetime as dt
import pandas as pd
import numpy as np
from numpy import arange
import matplotlib.pyplot as plt
from pandas import read_csv
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import RandomizedSearchCV
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

executed in 3.89s, finished 16:44:36 2022-03-24

Import data

In [26]:

```
Daily_goldprices = pd.read_csv('Gold price.csv')
```

executed in 18ms, finished 16:59:21 2022-03-24

In [27]:

```
Daily_goldprices
```

executed in 52ms, finished 16:59:42 2022-03-24

Out[27]:

	Date	Price(per troy ounce)	Price(per 1 gram)	Price(per 10 gram)
0	03-02-2020	1,12,338.7	3611.77	36117.71
1	04-02-2020	1,11,067.5	3570.90	35709.01
2	05-02-2020	1,10,622.1	3556.58	35565.82
3	06-02-2020	1,11,301.1	3578.41	35784.10
4	07-02-2020	1,12,279.4	3609.86	36098.62
...	...	...	...	...
743	NaN	NaN	NaN	NaN
744	NaN	NaN	NaN	NaN
745	NaN	NaN	NaN	NaN
746	NaN	NaN	NaN	NaN
747	NaN	NaN	NaN	NaN

748 rows × 4 columns

Data Understanding

In [4]:

```
Daily_goldprices.isnull().sum().sum()
```

executed in 95ms, finished 16:44:38 2022-03-24

Out[4]:

```
68
```

In [28]:

```
Daily_goldprices.shape
```

executed in 31ms, finished 17:00:21 2022-03-24

Out[28]:

```
(748, 4)
```

In [29]:

```
Daily_goldprices.dtypes
```

executed in 13ms, finished 17:00:54 2022-03-24

Out[29]:

```
Date                      object
Price(per troy ounce)     object
Price(per 1 gram)        float64
Price(per 10 gram)       float64
dtype: object
```

In [30]:

```
Daily_goldprices.info()
```

executed in 223ms, finished 17:01:28 2022-03-24

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 748 entries, 0 to 747
Data columns (total 4 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Date                   731 non-null    object 
 1   Price(per troy ounce)  731 non-null    object 
 2   Price(per 1 gram)      731 non-null    float64
 3   Price(per 10 gram)     731 non-null    float64
dtypes: float64(2), object(2)
memory usage: 23.5+ KB
```

In [6]:

```
Daily_goldprices.describe().T
```

executed in 357ms, finished 16:44:54 2022-03-24

Out[6]:

	count	mean	std	min	25%	50%	75%	max
Price(per 1 gram)	731.0	4280.971135	228.015556	3552.47	4194.895	4305.03	4414.725	4980.2
Price(per 10 gram)	731.0	42809.711491	2280.153543	35524.72	41948.970	43050.30	44147.250	49802.0

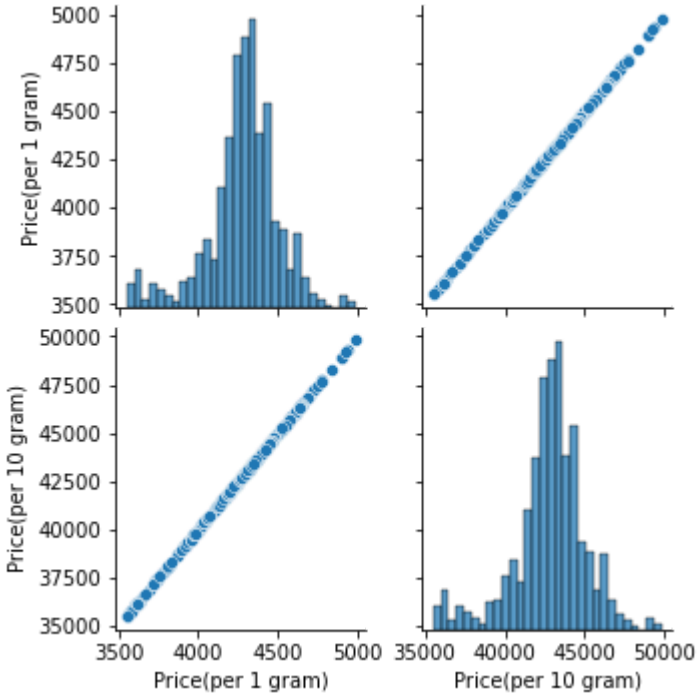
In [7]:

```
sns.pairplot(Daily_goldprices)
```

executed in 8.14s, finished 16:45:50 2022-03-24

Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x276336080d0>
```

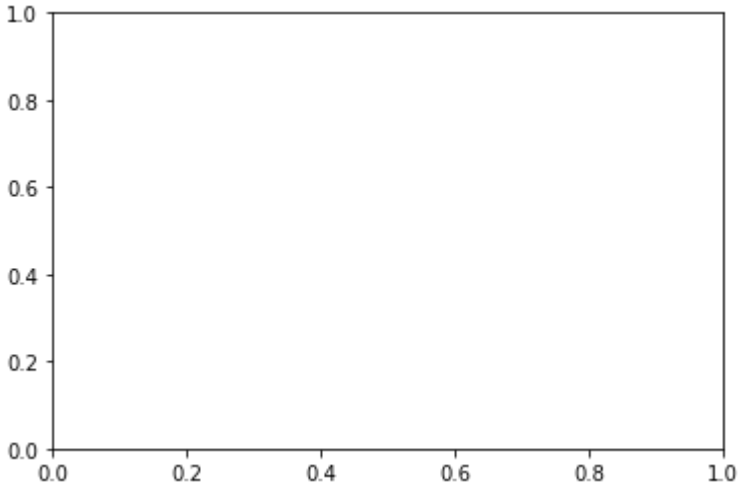


In [8]:

```
plt.boxplot(Daily_goldprices)
```

executed in 1.10s, finished 16:46:09 2022-03-24

```
TypeError: can only concatenate str (not "float") to str
```



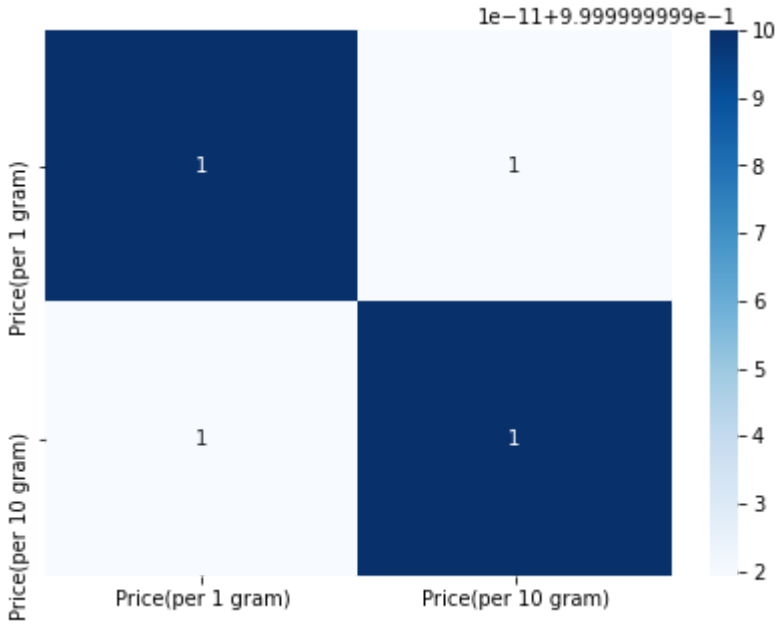
In [10]:

```
plt.figure(figsize=(7,5))
sns.heatmap(Daily_goldprices.corr(),cmap='Blues',annot=True)
```

executed in 829ms, finished 16:47:21 2022-03-24

Out[10]:

<AxesSubplot:>



In [13]:

```
Daily_goldprices.drop(columns=['Price(per 10 gram)'],inplace=True,axis=0)
```

executed in 20ms, finished 16:50:50 2022-03-24

In [17]:

```
amount= Daily_goldprices.interpolate(method = 'linear')
amount.head(7)
```

executed in 31ms, finished 16:52:57 2022-03-24

Out[17]:

	Date	Price(per troy ounce)	Price(per 1 gram)
0	03-02-2020	1,12,338.7	3611.77
1	04-02-2020	1,11,067.5	3570.90
2	05-02-2020	1,10,622.1	3556.58
3	06-02-2020	1,11,301.1	3578.41
4	07-02-2020	1,12,279.4	3609.86
5	08-02-2020	1,12,279.4	3609.86
6	09-02-2020	1,12,279.4	3609.86

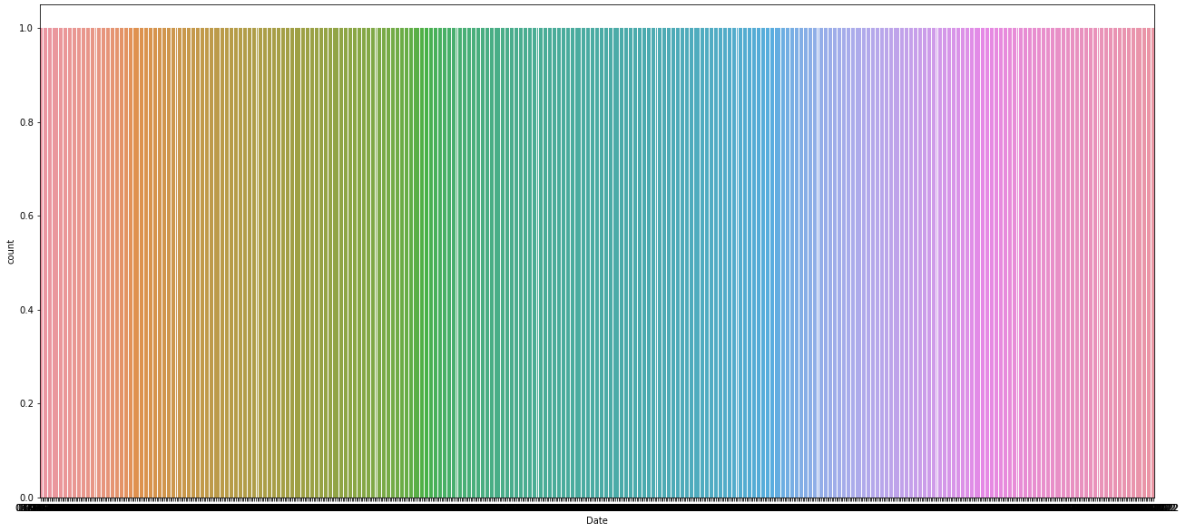
Data visualization

In [31]:

```
print(Daily_goldprices['Date'].value_counts())
plt.figure(figsize=(22,10))
sns.countplot(Daily_goldprices['Date'])
plt.show()
```

executed in 48.8s, finished 17:03:09 2022-03-24

```
10-03-2021    1
22-08-2020    1
09-12-2021    1
13-02-2021    1
17-07-2021    1
             ..
13-12-2021    1
29-03-2021    1
21-08-2020    1
09-04-2021    1
13-03-2021    1
Name: Date, Length: 731, dtype: int64
```

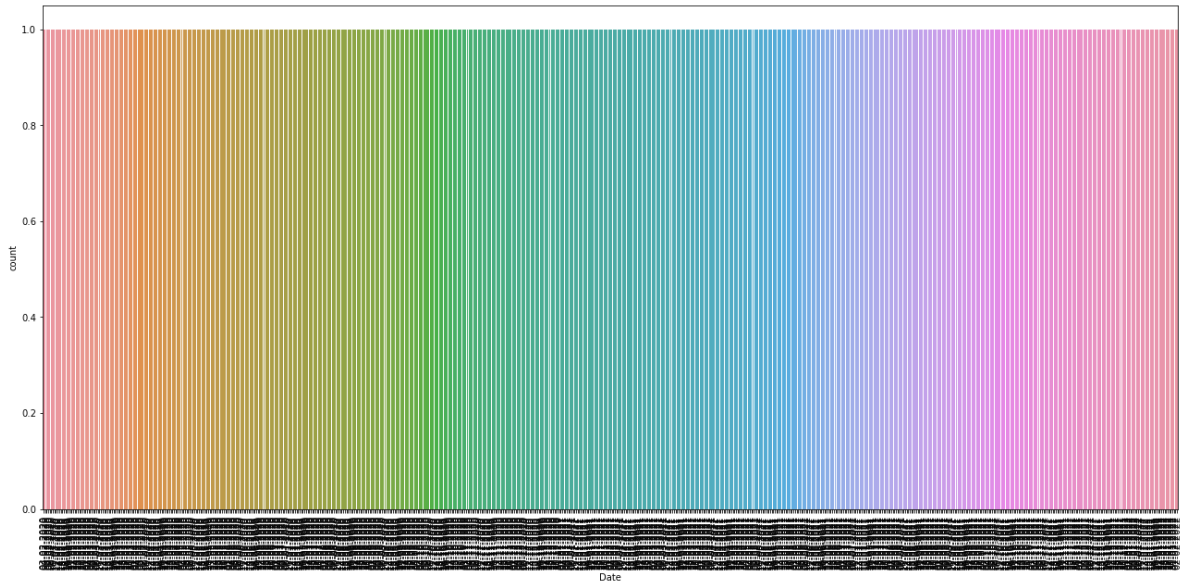


In [32]:

```
print(Daily_goldprices['Date'].value_counts()[:10])
plt.figure(figsize=(22,10))
sns.countplot(Daily_goldprices['Date'])
plt.xticks(rotation=90)
plt.show()
```

executed in 46.4s, finished 17:03:56 2022-03-24

```
10-03-2021    1
22-08-2020    1
09-12-2021    1
13-02-2021    1
17-07-2021    1
22-01-2022    1
20-10-2020    1
05-10-2021    1
25-11-2020    1
12-11-2021    1
Name: Date, dtype: int64
```



In [40]:

```
import plotly.plotly as py
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go
```

executed in 31ms, finished 17:05:49 2022-03-24

```
ModuleNotFoundError: No module named 'plotly'
```

In [41]:

```
data = Daily_goldprices['Price(per 10 gram)']
print(data[:])
data = [go.Prices(
            x=data[:].index,
            y=data[:].values,
            hoverinfo = 'text',
            marker = dict(color = 'rgba(177, 14, 22, 0.5)',
                          line=dict(color='rgb(0,0,0)',width=1.5)),
    )]

layout = dict(
    title = 'Prices',
)
fig = go.Figure(data=data, layout=layout)
iplot(fig)
```

executed in 48ms, finished 17:05:50 2022-03-24

```
0      36117.71
1      35709.01
2      35565.82
3      35784.10
4      36098.62
         ...
743          NaN
744          NaN
745          NaN
746          NaN
747          NaN
Name: Price(per 10 gram), Length: 748, dtype: float64
```

NameError: name 'go' is not defined

In [ ]:

```
Daily_goldprices.head()
```

executed in 47.5s, finished 17:03:56 2022-03-24

In [ ]:

```
Daily_goldprices.tail()
```

executed in 40.9s, finished 17:03:56 2022-03-24

In [ ]: