

Cryptocurrency prices analysis

Member 1 : Somesh Ratre

Roll No. : 210260054

Member 3 : Godwin Arunim Beck

Roll No. : 210260016

Member 2 : Utpal Rathva

Roll No. : 210260043

Member 4 : Rohan Meena

Roll No. : 21d070058

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import statsmodels.api as sm
from scipy.stats import ttest_ind
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

crypto = pd.read_csv('Cryptocurrency Prices by Date.csv')
crypto.head()
```

```
print(f'The dataset has a shape of {crypto.shape}', '\n')
print(f'The dataset has {crypto.isnull().sum().sum()} missing values' , '\n')
print(f'The dataset has {crypto.duplicated().sum()} duplicated rows' , '\n')
```

Output:

The dataset has a shape of (657316, 3)

The dataset has 0 missing values

The dataset has 0 duplicated rows

```
crypto.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 657316 entries, 0 to 657315
```

```
Data columns (total 3 columns):
```

```
#   Column   Non-Null Count  Dtype
---
```

```
0   Currency 657316 non-null object
```

```
1   Date     657316 non-null int64
```

```
2   Price    657316 non-null float64
```

```
dtypes: float64(1), int64(1), object(1)
```

```
memory usage: 15.0+ MB
```

```
pd.set_option('display.max_rows', None)
```

```
#Number of cryptocurrencies
```

```
crypto['Currency'].value_counts().sort_values(ascending=False)
```

```
plt.style.use('ggplot')
```

```
crypto['Currency'].value_counts().sort_values(ascending=False).head(10).plot(kind='bar',
color='blue', figsize=(10,5),title='Top 10 Cryptocurrencies by Volume Traded')
```

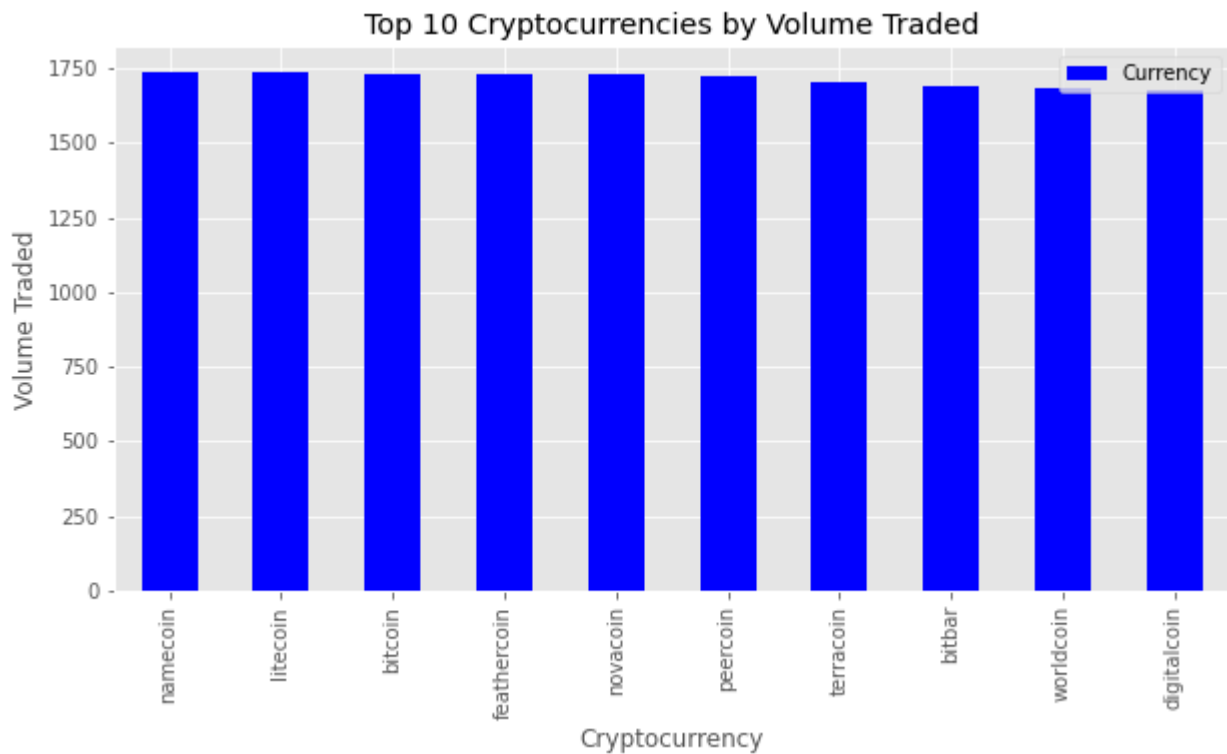
```
plt.legend()
```

```
plt.xlabel('Cryptocurrency')
```

```
plt.ylabel('Volume Traded')
```

```
plt.show();
```

Output:



#Date column is written in this format 1502892561000 which is the number of milliseconds since 1 January 1970

#Convert the date column to datetime format

```
crypto['Date'] = pd.to_datetime(crypto['Date'], unit='ms')
```

#Obtain the year, month, day and time from the date column

```
crypto['month'] = crypto['Date'].dt.month
```

```
crypto['year'] = crypto['Date'].dt.year
```

```
crypto['day'] = crypto['Date'].dt.day
```

```
crypto['time'] = crypto['Date'].dt.time
```

```
crypto['Date'] = crypto['Date'].apply(lambda x: str(x).split(' ')[0])
```

```
crypto.head()
```

```
crypto = crypto[['Currency', 'Date', 'month', 'year', 'day', 'time', 'Price']]
```

```
crypto.head()
```

```
crypto['year'].value_counts().sort_values(ascending=False).plot(kind='bar',color='blue',figsize=(10,5),
title='No. of Cryptocurrencies Traded Each year')
```

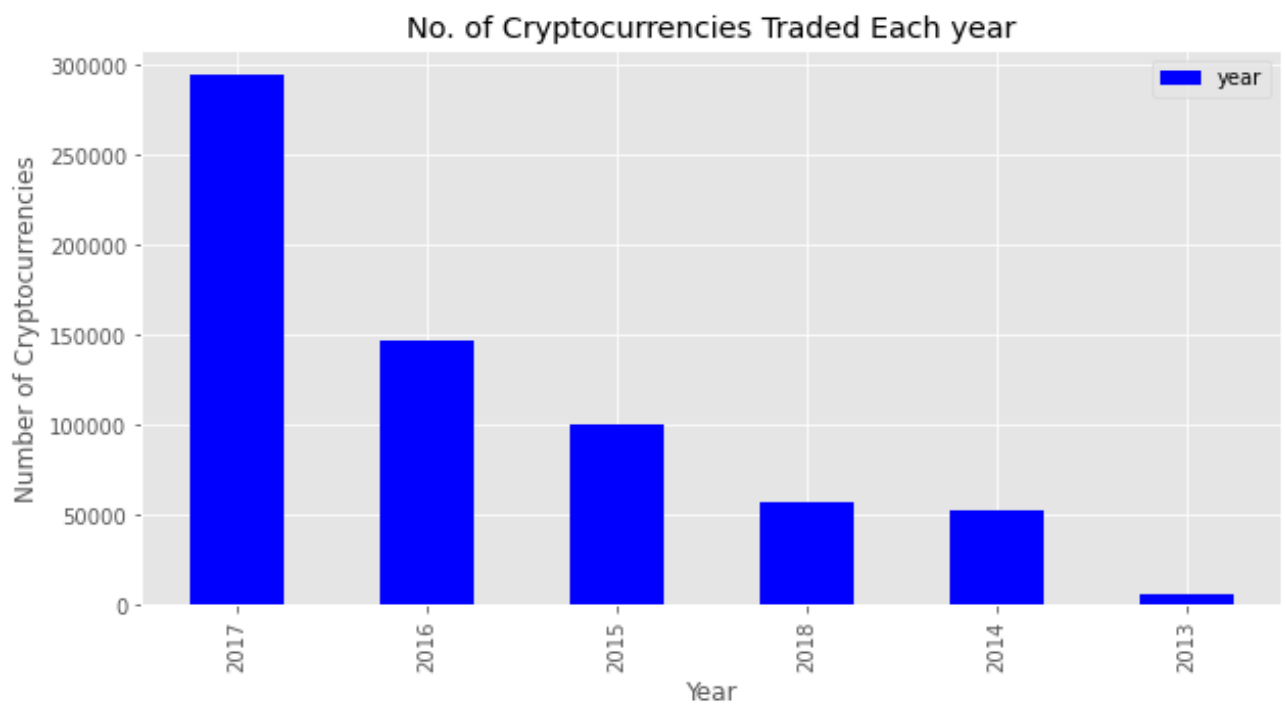
```
plt.legend()
```

```
plt.xlabel('Year')
```

```
plt.ylabel('Number of Cryptocurrencies')
```

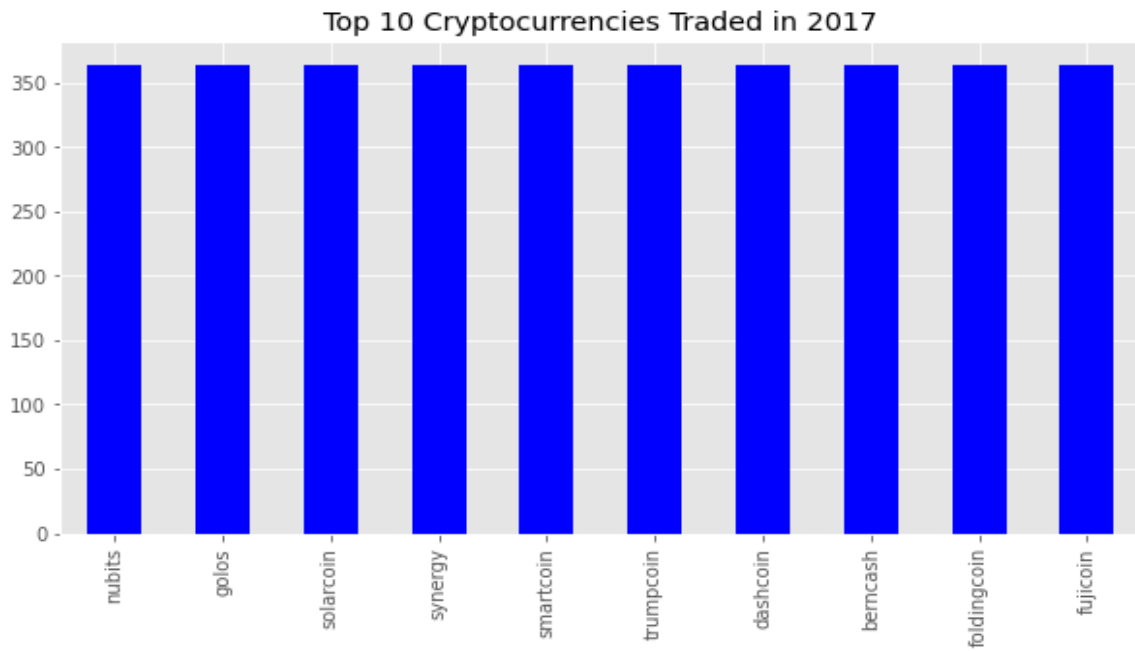
```
plt.show();
```

Output :



#2017 cryptocurrency trade

```
crypto[crypto['year'] ==
2017]['Currency'].value_counts().sort_values(ascending=False).head(10).plot(kind='bar',color='blue',f
igsize=(10,5),title='Top 10 Cryptocurrencies Traded in 2017')
```



```

crypto_17 = crypto.query('year == 2017')

#Plot the number of cryptocurrncies traded each month in 2017

crypto_17['month'].value_counts().sort_values(ascending=False).plot(kind='bar',color='blue',
figsize=(10,5),title='No. of Cryptocurrencies Traded Each Month in 2017')

plt.legend()

#Replace xticks with month names in reverse order

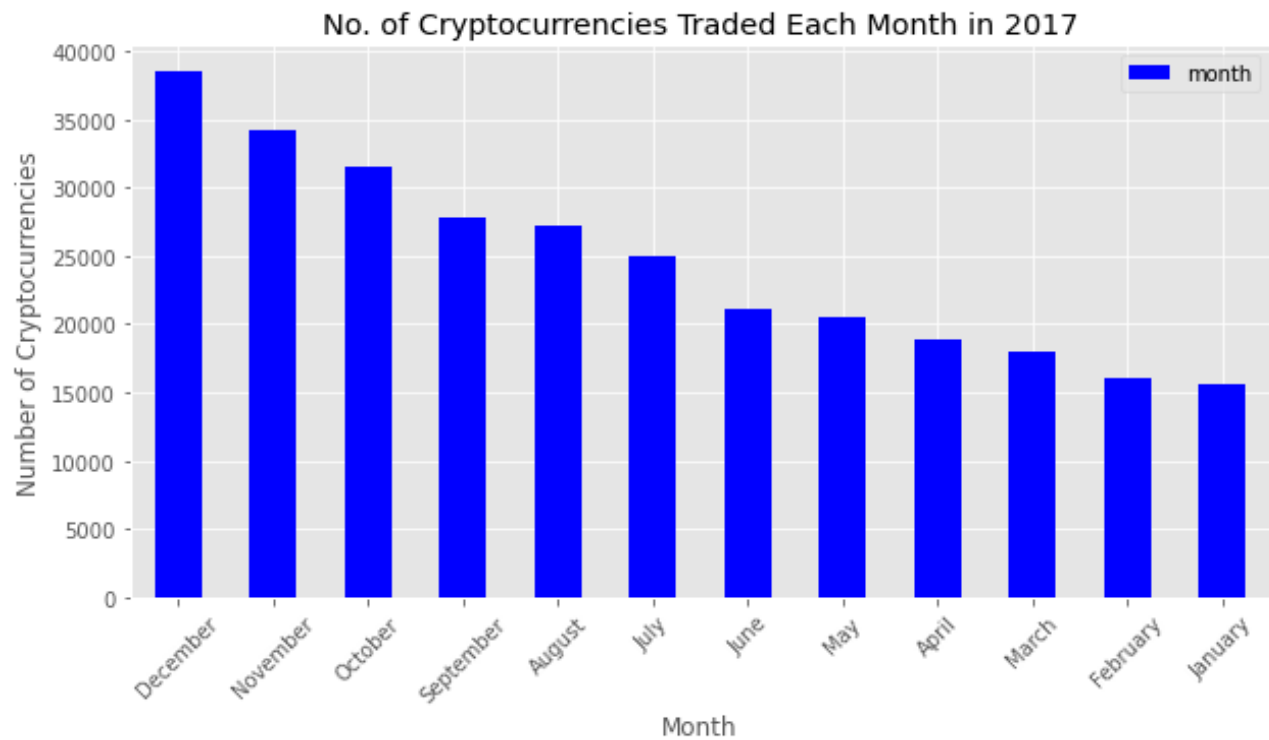
plt.xticks(np.arange(12), ('December',
'November','October','September','August','July','June','May','April','March','February','January'),rota
tion=45)

plt.xlabel('Month')

plt.ylabel('Number of Cryptocurrencies')

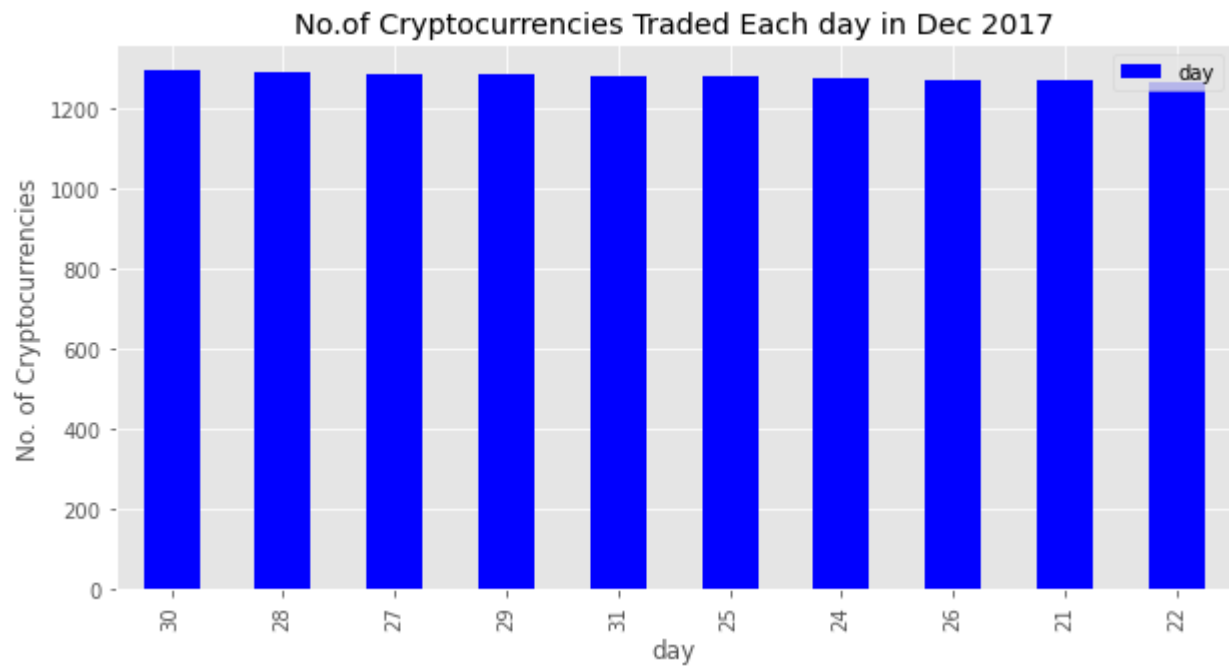
plt.show();

```



#Which Day in Dec 2017 had the most cryptocurrencies traded?

```
crypto_17[crypto_17['month'] ==  
12]['day'].value_counts().sort_values(ascending=False).head(10).plot(kind='bar',color='blue',figsize=(  
10,5),title='No.of Cryptocurrencies Traded Each day in Dec 2017')  
  
plt.legend()  
  
plt.xlabel('day')  
  
plt.ylabel('No. of Cryptocurrencies')  
  
plt.show();
```



Price of Solarcoin in 2017

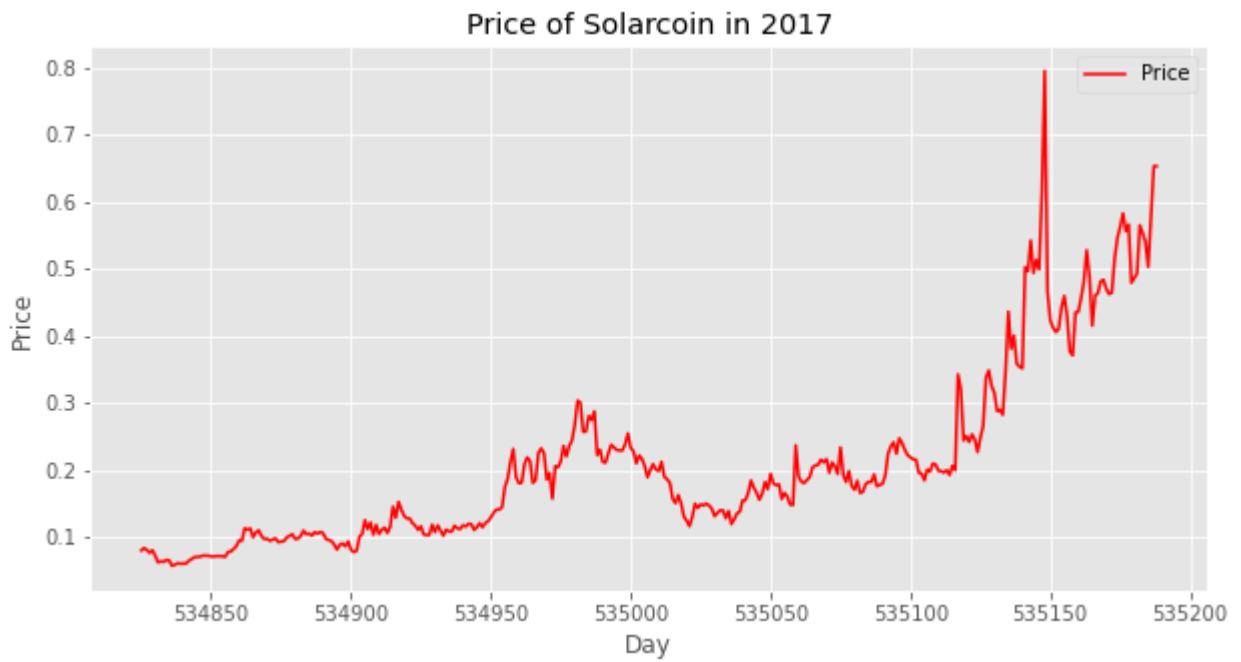
```
crypto_17[crypto_17['Currency'] == 'solarcoin']['Price'].plot(kind='line', color='r', figsize=(10, 5),  
title='Price of Solarcoin in 2017')
```

```
plt.legend()
```

```
plt.xlabel('Day')
```

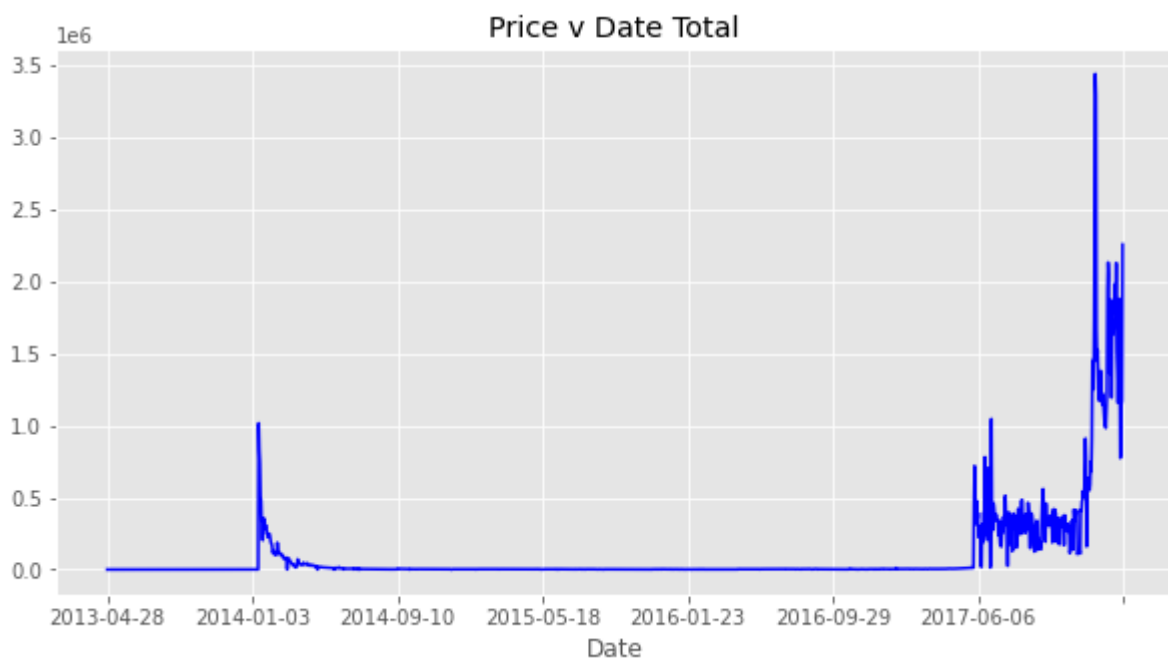
```
plt.ylabel('Price')
```

```
plt.show()
```



Price v date total

```
crypto.groupby('Date')['Price'].sum().plot(kind='line', color='b', figsize=(10, 5), title='Price v Date Total')
```



A combo of a bar and line plot to show the number of cryptocurrencies traded each month in 2017 and the total price of cryptocurrencies traded each month in 2017

```
fig, ax1 = plt.subplots(figsize=(16, 10))
```

```
ax2 = ax1.twinx()
```



```

ax1.bar(crypto_17['month'].value_counts().sort_values(ascending=False).index,
crypto_17['month'].value_counts().sort_values(ascending=False).values, color='b')

ax2.plot(crypto_17.groupby('month')['Price'].sum(), color='r')

ax1.set_xlabel('Month')

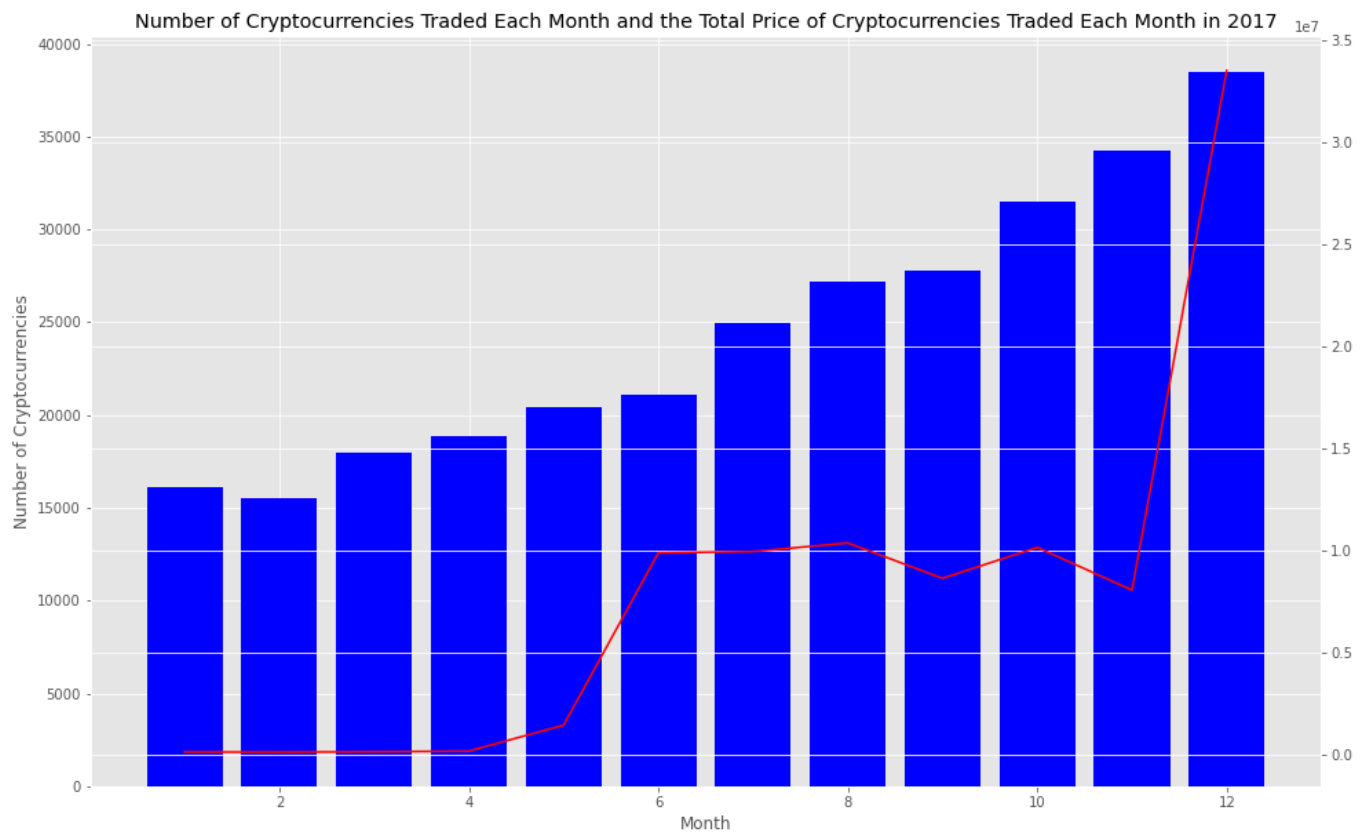
ax1.set_ylabel('Number of Cryptocurrencies')

ax1.set_xlabel('Month')

#ax2.set_ylabel('Total Price')

plt.title('Number of Cryptocurrencies Traded Each Month and the Total Price of Cryptocurrencies
Traded Each Month in 2017');

```



```

# Plot the correlation heatmap

sns.heatmap(crypto.corr(), annot=True, cmap='terrain', linewidths=0.2)

```



#Select two specific cryptocurrencies for hypothesis testing

```
bitcoin_prices = crypto[crypto['Currency'] == 'bitcoin']['Price']
```

```
ethereumcoin_prices = crypto[crypto['Currency'] == 'ethereum']['Price']
```

#Perform a t-test to compare the mean prices of Bitcoin and Ethereum

```
t_stat , p_value = ttest_ind(bitcoin_prices, ethereumcoin_prices)
```

```
print('Hypothesis Testing: Bitcoin vs Ethereumcoin')
```

```
print(f'T-Statistic: {t_stat}')
```

```
print(f'P-Value: {p_value}')
```

```
plt.figure(figsize=(15,10))
```

```
sns.histplot(bitcoin_prices, color= 'blue', label='bitcoin')
```

```
sns.histplot(ethereumcoin_prices, color= 'green', label='ethercoin')
```

```
plt.xlabel('Price')
```

```
plt.ylabel('Price Dist. : Bitcoin vs Ethereumcoin')
```

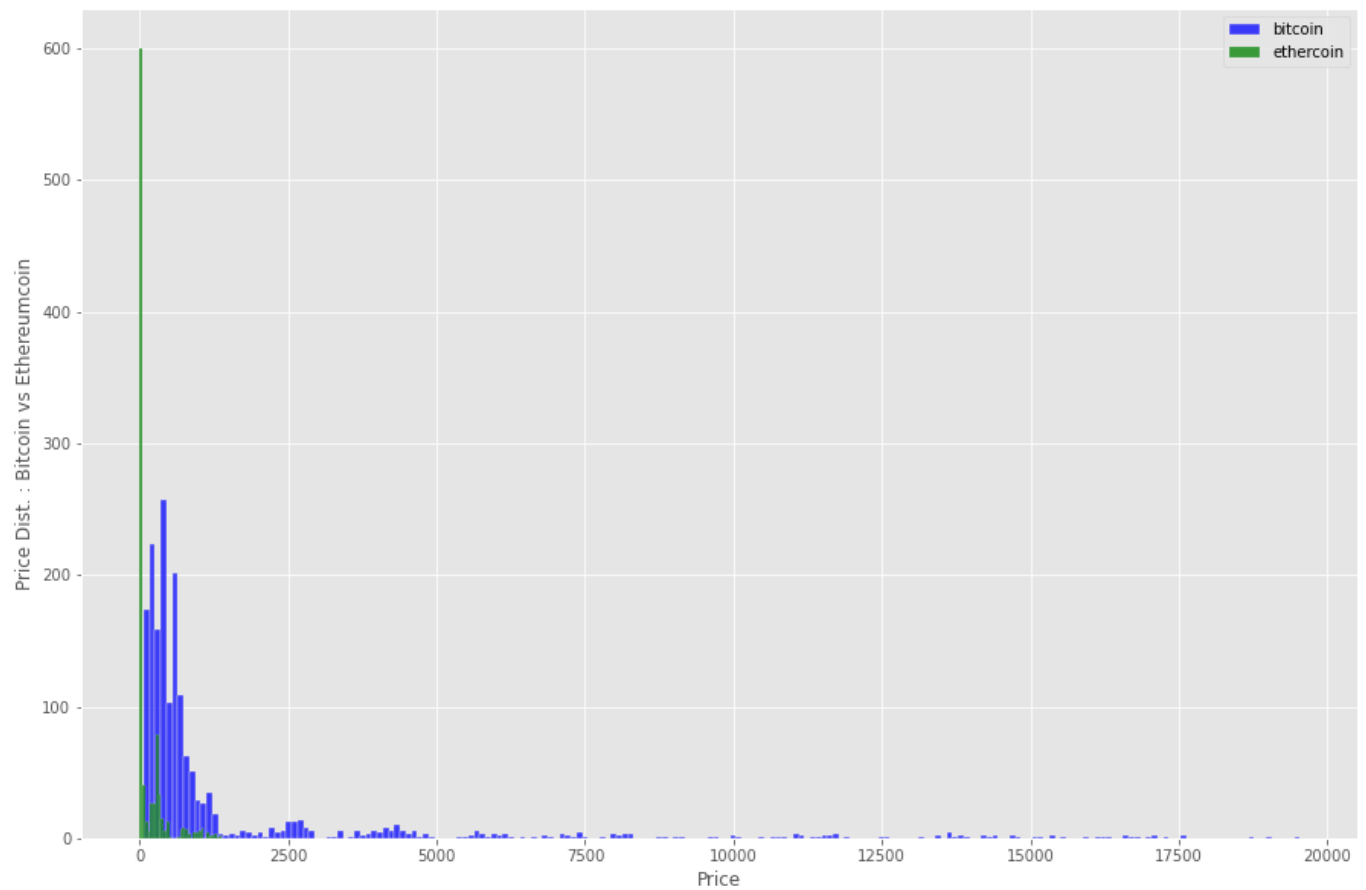
```
plt.legend()
```

```
plt.show()
```

Hypothesis Testing: Bitcoin vs Ethereumcoin

T-Statistic: 13.591758389941736

P-Value: 1.0080870726092734e-40



```
bitcoin = crypto[crypto['Currency'] == 'bitcoin'].copy()
```

```
from numpy.random import choice
```

```
from numpy import *
```

```
from matplotlib.pyplot import *
```

```
import scipy.stats as stats
```

```
bitcoin.describe().Price
```

Output:

```
count    1733.000000
```

```
mean      1436.394988
```

```
std       2883.248598
```

```
min        67.809000
```

```
25%       274.517000
```

```
50%       480.770000
```

```
75%       853.010000
```

```
max    19535.700000
```

```
Name: Price, dtype: float64
```

```
bitprice = bitcoin['Price'].to_numpy()
```

```
bitprice
```

```
Output: array([ 135.3 , 141.96, 135.3 , ..., 8179.93, 8003.77, 9037.51])
```

```
n_sample = 175
```

```
sample= choice(bitprice, n_sample, replace= False)
```

```
sample
```

```
def collect_sample(data, n_sample, n_choice):
```

```
    avgs = zeros(n_choice)
```

```
    for i in range(n_choice):
```

```
        sample = choice(data, n_sample, replace=False)
```

```
        avgs[i] = mean(sample)
```

```
    return avgs
```

```
avgs = collect_sample(bitprice, n_sample,200)
```

```
avgs
```

```
hist(avgs, bins=20,color='b')
```

