In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.simplefilter('ignore')
import yfinance as yf
import seaborn as sns
```

In [2]:

```
start = '2010-12-15'
end = '2020-12-15'
tickers = ['GOOGL','AAPL','KO','IBM']
interval = "1d"
portfolio_stocks =yf.download(tickers,start,end,interval)
```

```
[*********************100%***********************]  4 of 4 completed
```

In [3]:

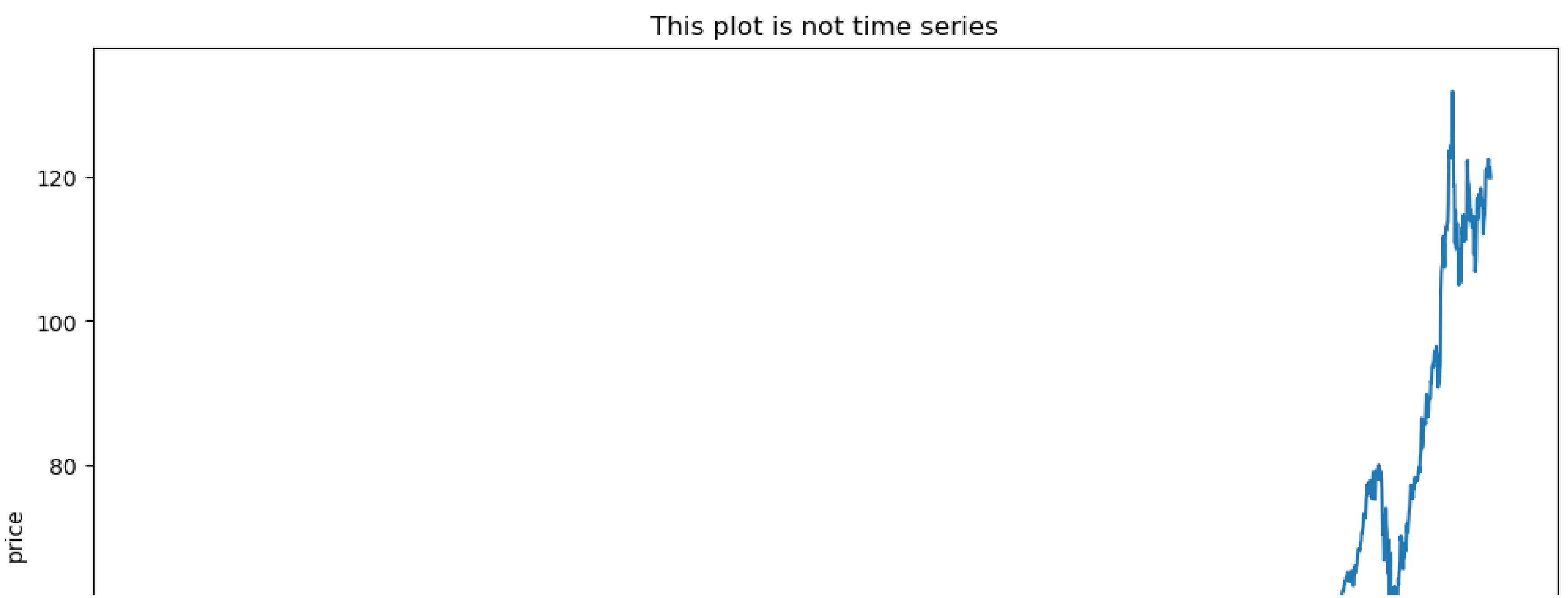
```
portfolio_stocks = portfolio_stocks ['Adj Close']
portfolio_stocks
```

Out[3]:

```
                  AAPL      GOOGL         IBM         KO
      Date
2010-12-15    9.711447  14.772272   87.241577  21.828802
2010-12-16    9.738425  14.807558   87.139114  22.031120
2010-12-17    9.719026  14.784785   87.410385  22.152498
2010-12-20    9.767528  14.891391   87.114998  22.021002
2010-12-21    9.827853  15.091842   87.856468  22.081684
       ...         ...        ...         ...        ...
2020-12-08  122.382462  90.566498  105.037552  48.928303
2020-12-09  119.824226  88.892998  105.939934  49.066311
2020-12-10  121.260780  88.382500  104.410881  48.808697
2020-12-11  120.444107  88.739998  103.834335  49.084717
2020-12-14  119.824226  87.612999  103.216019  49.011108

2517 rows × 4 columns
```

In [4]:

```
plt.figure(num=1,figsize=(12,8))
plt.plot(portfolio_stocks.AAPL.values)
plt.title('This plot is not time series')
plt.xlabel('days')
plt.ylabel('price')
plt.show()
```

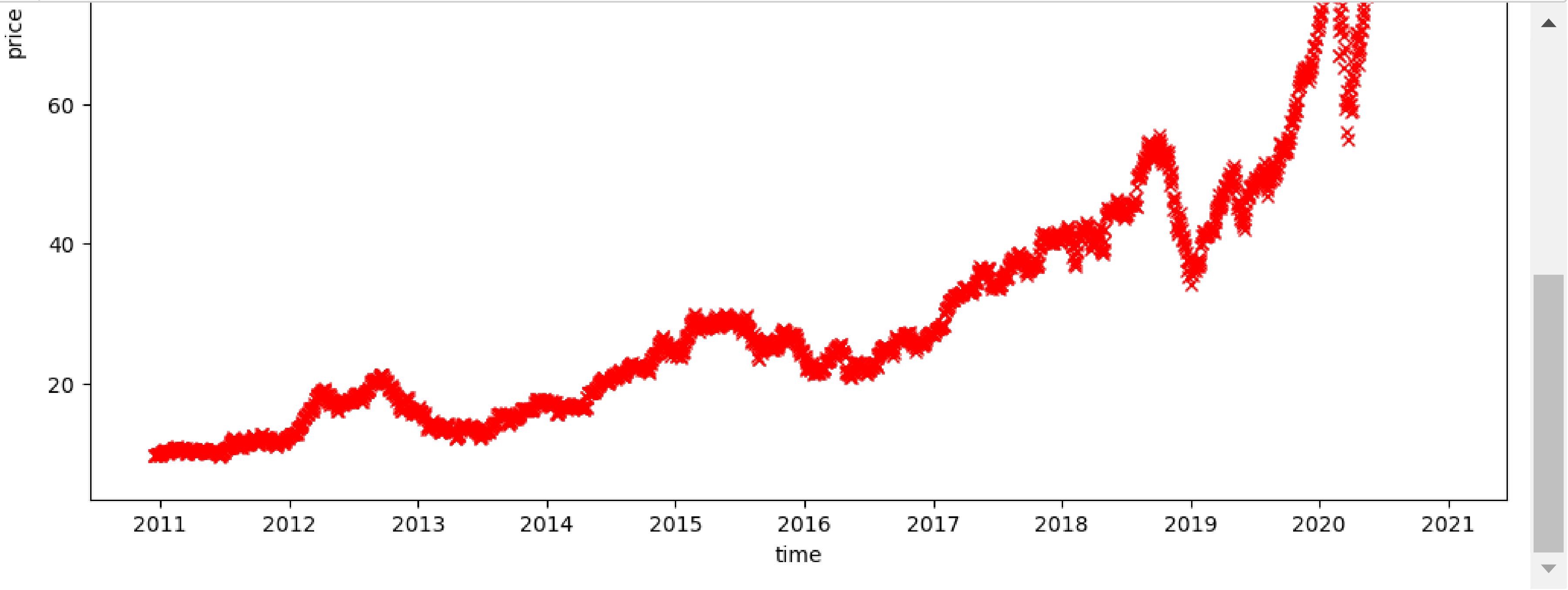


```
# haracter  description
'-' solid line style
'--'    dashed line style
'-.'    dash-dot line style
':' dotted line style
'.' point marker
',' pixel marker
'o' circle marker
'v' triangle_down marker
'^' triangle_up marker
'<' triangle_left marker
'>' triangle_right marker
'1' tri_down marker
'2' tri_up marker
```

```
'3' tri_left marker
'4' tri_right marker
's' square marker
'p' pentagon marker
'*' star marker
'h' hexagon1 marker
'H' hexagon2 marker
'+' plus marker
'x' x marker
'D' diamond marker
'd' thin_diamond marker
'|' vline marker
'_' hline marker
The following color abbreviations are supported:

character   color
‘b’ blue
‘g’ green
‘r’ red
‘c’ cyan
‘m’ magenta
‘y’ yellow
‘k’ black
‘w’ white
```

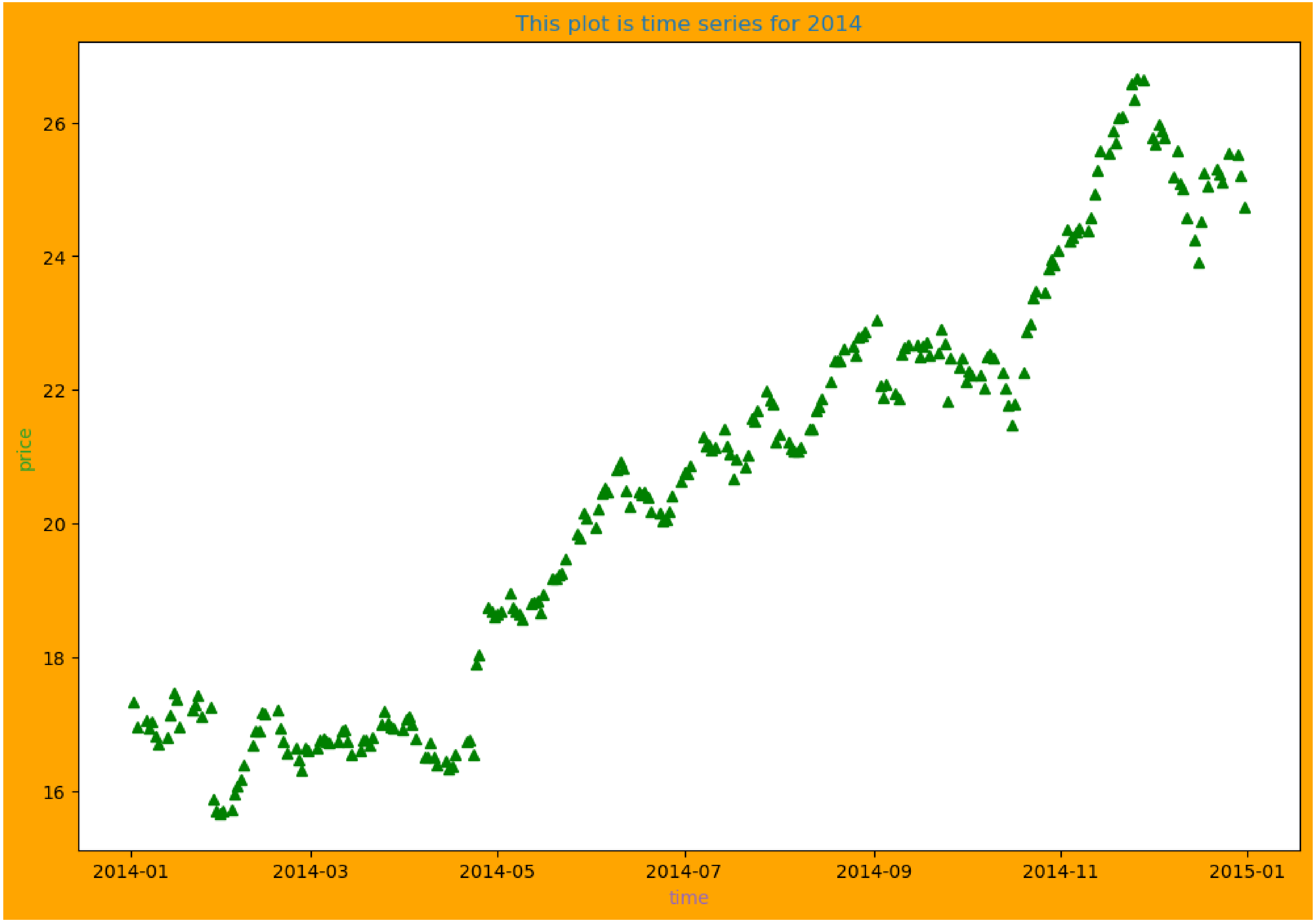
In [6]:

```
plt.figure(num=2,figsize=(12,8))
plt.plot(portfolio_stocks.AAPL,'rx')
plt.title('This plot is time series')
plt.xlabel('time')
plt.ylabel('price')
plt.show()
```



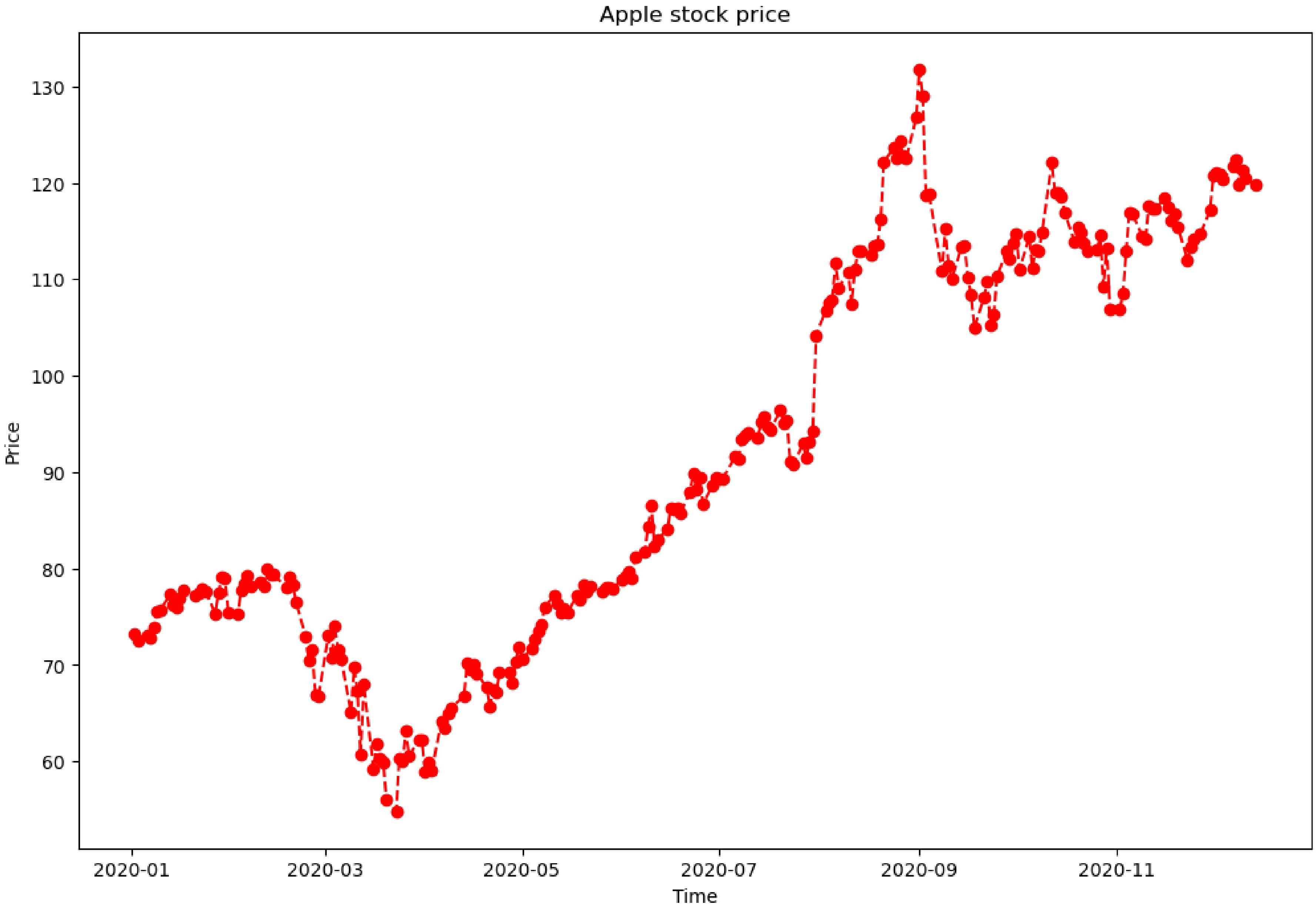
In [7]:

```
plt.figure(num=3,figsize=(12,8),facecolor='orange')
plt.plot(portfolio_stocks.loc['2014','AAPL'],'g^')
plt.title('This plot is time series for 2014',color='C0')
plt.xlabel('time',color='C4')
plt.ylabel('price',color='C2')

plt.show()
```


This plot is time series for 2014
price
26
24
22
20
18
16
time
2014-01
2014-03
2014-05
2014-07
2014-09
2014-11
2015-01

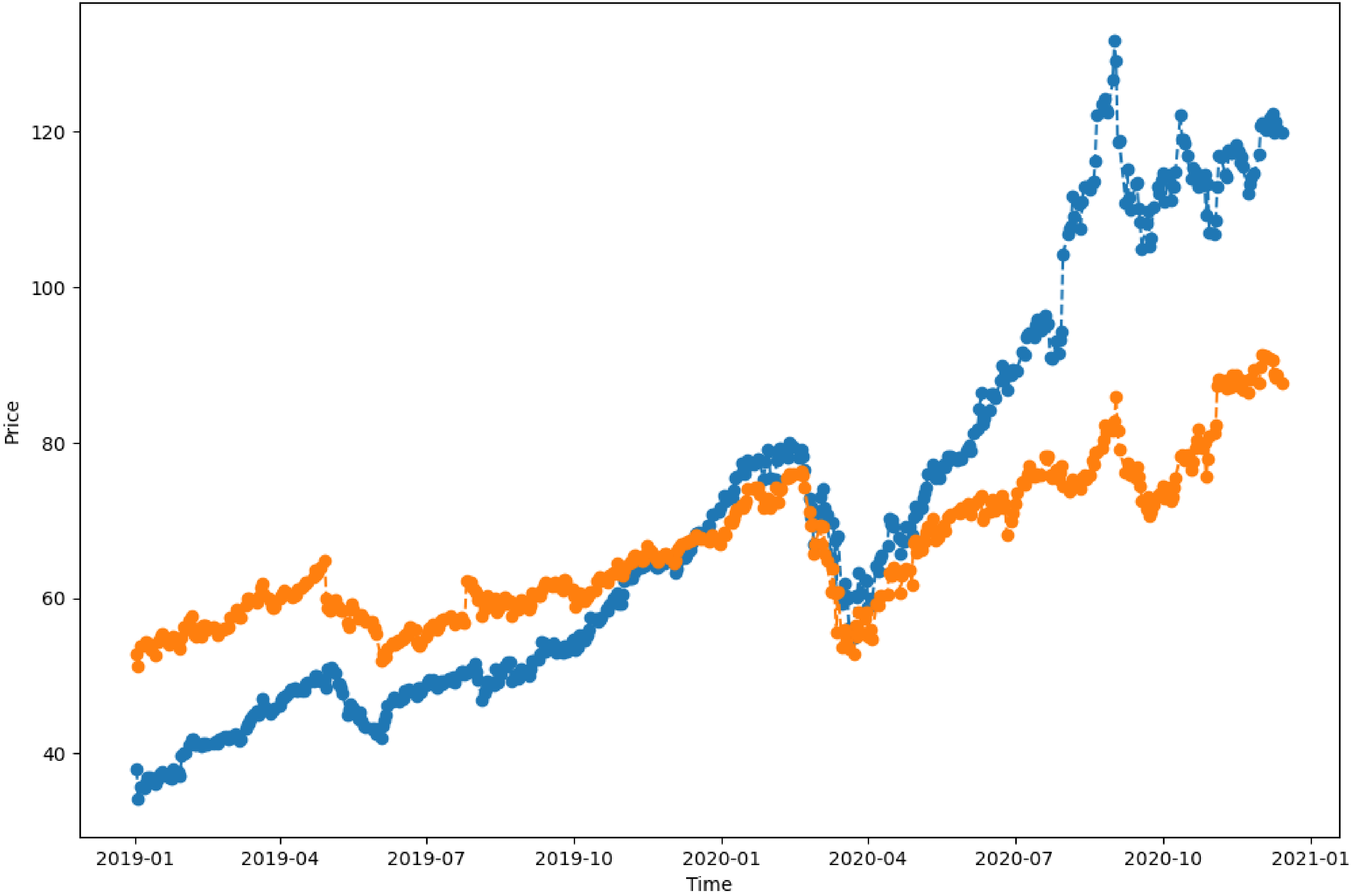
In [8]:

```
fig,ax = plt.subplots(figsize=(12,8))
ax.plot(portfolio_stocks.loc['2020':,'AAPL'],marker="o",linestyle="--",color="r")
ax.set_xlabel("Time")
ax.set_ylabel("Price")
ax.set_title("Apple stock price")
plt.show()
```


Apple stock price
Price
Time
130
120
110
100
90
80
70
60
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11

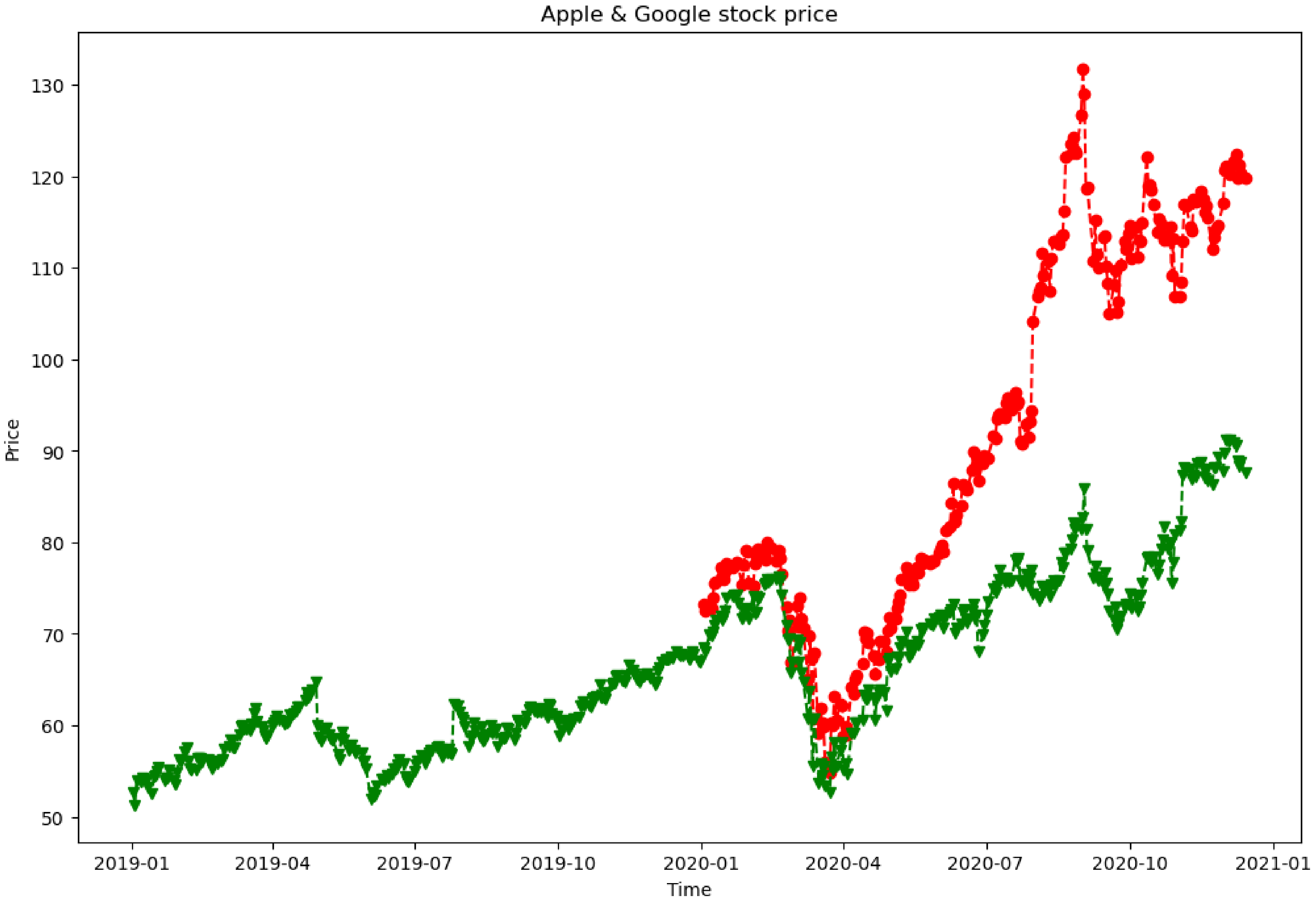
In [9]:

```
fig,ax = plt.subplots(figsize=(12,8))
ax.plot(portfolio_stocks.loc['2019':,['AAPL','GOOGL']],marker="o",linestyle="--")
ax.set_xlabel("Time")
ax.set_ylabel("Price")
plt.show()
```


120
100
80
60
40
Price
2019-01
2019-04
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
Time

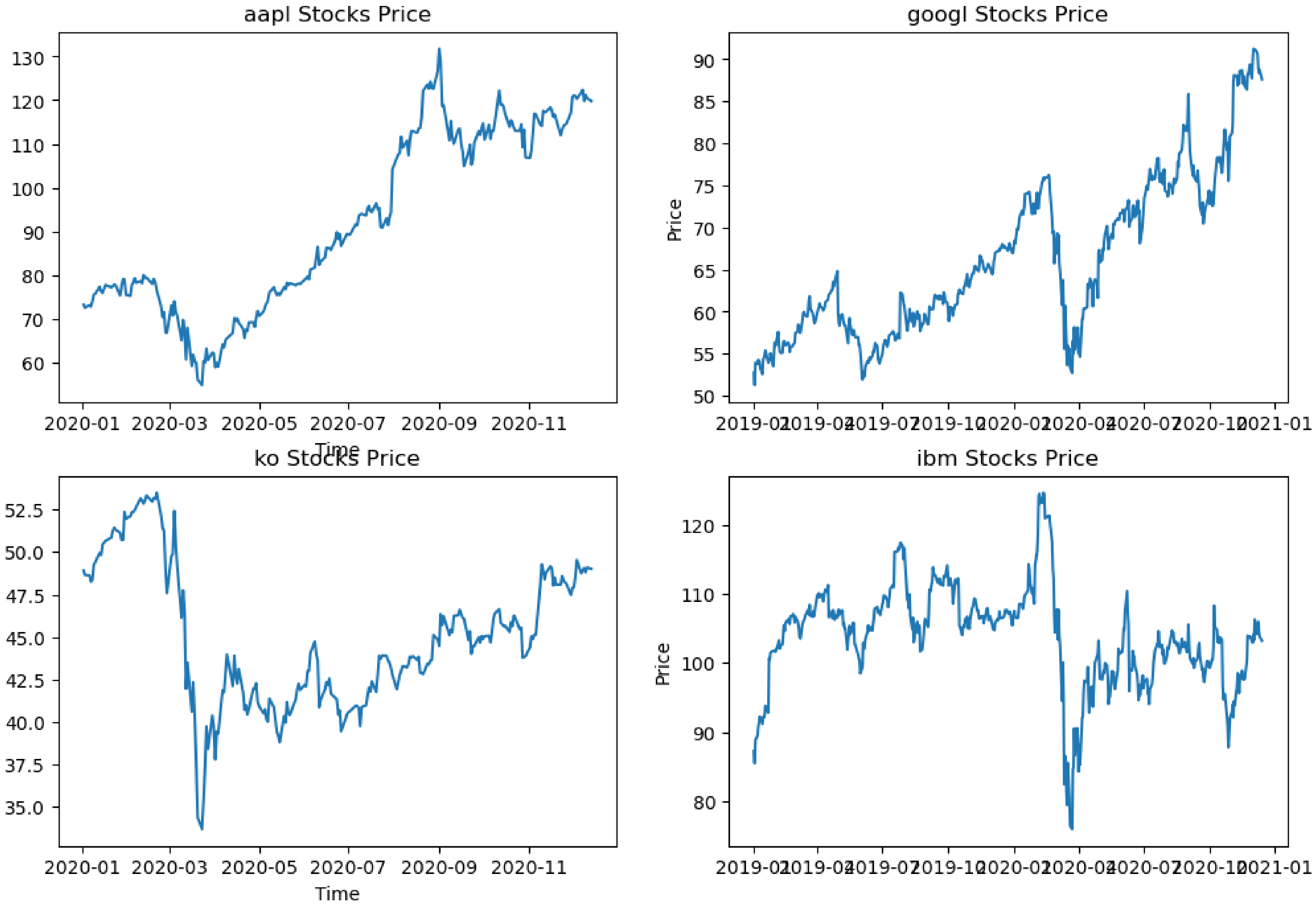
In [10]:

```python
# If put socks data separated, we can customize time and plot color, style
fig,ax = plt.subplots(figsize=(12,8))
ax.plot(portfolio_stocks.loc['2020':,'AAPL'],marker="o",linestyle="--",color="r")
ax.plot(portfolio_stocks.loc['2019':,'GOOGL'],marker="v",linestyle="--",color="g")
ax.set_title("Apple & Google stock price")
ax.set_xlabel("Time")
ax.set_ylabel("Price")
plt.show()
```


Apple & Google stock price
Price
130
120
110
100
90
80
70
60
50
2019-01
2019-04
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
Time

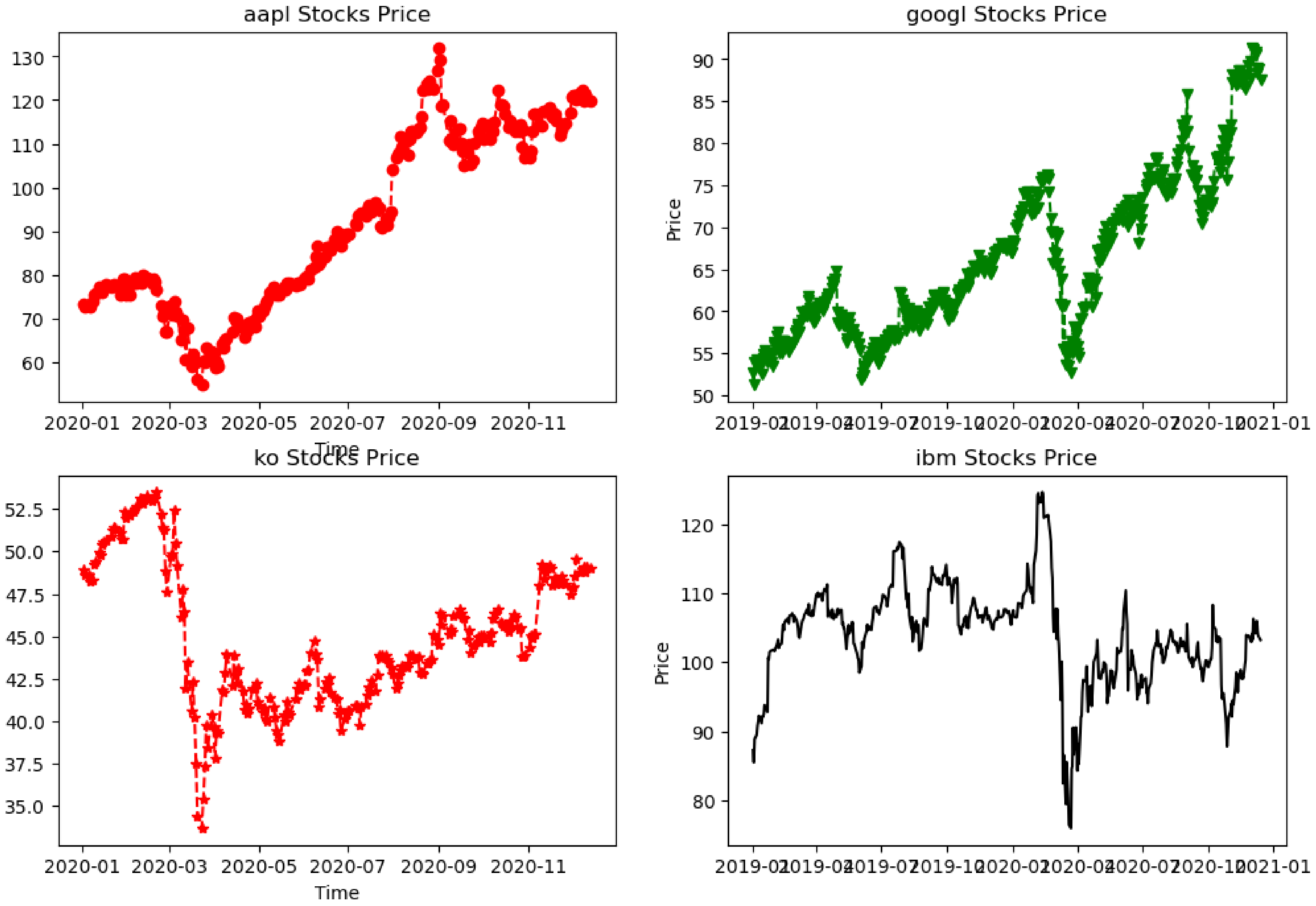
In [11]:

```
fig,ax = plt.subplots(2, 2,figsize=(12,8))
ax[0,0].plot(portfolio_stocks.loc['2020':,'AAPL'])
ax[0,1].plot(portfolio_stocks.loc['2019':,'GOOGL'])
ax[1,0].plot(portfolio_stocks.loc['2020':,'KO'])
ax[1,1].plot(portfolio_stocks.loc['2019':,'IBM'])
ax[0,0].set_title("aapl Stocks Price")
ax[0,1].set_title("googl Stocks Price")
ax[1,0].set_title("ko Stocks Price")
ax[1,1].set_title("ibm Stocks Price")
ax[0,0].set_xlabel("Time")
ax[0,1].set_ylabel("Price")
ax[1,0].set_xlabel("Time")
ax[1,1].set_ylabel("Price")
plt.show()
```


aapl Stocks Price
130
120
110
100
90
80
70
60
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
Time
googl Stocks Price
Price
90
85
80
75
70
65
60
55
50
2019-01
2019-04
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
ko Stocks Price
52.5
50.0
47.5
45.0
42.5
40.0
37.5
35.0
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
Time
ibm Stocks Price
Price
120
110
100
90
80
2019-01
2019-04
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01

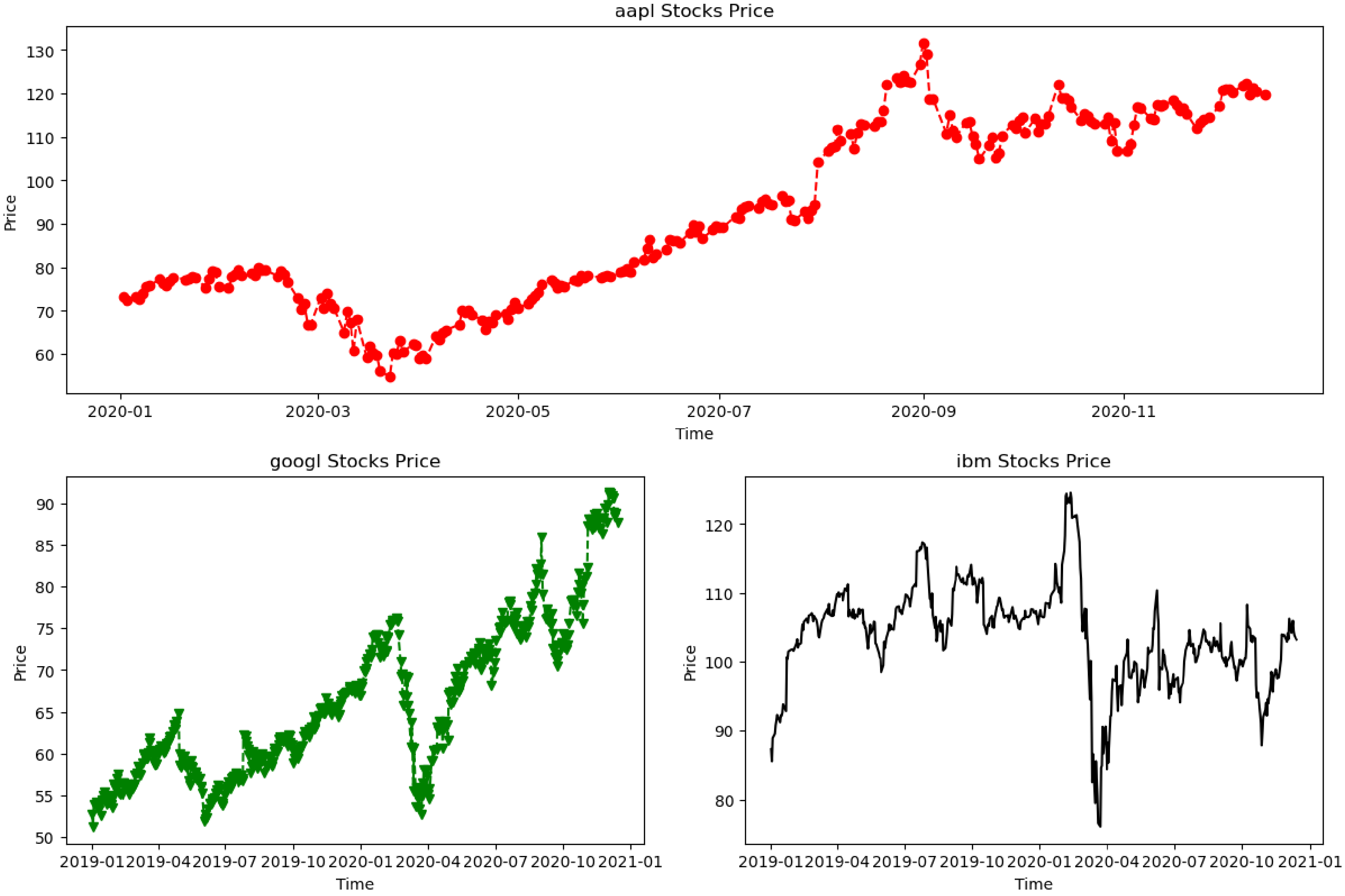
In [12]:

```
fig,ax = plt.subplots(2, 2,figsize=(12,8))

ax[0,0].plot(portfolio_stocks.loc['2020':,'AAPL'],marker="o",linestyle="--",color="r")
ax[0,1].plot(portfolio_stocks.loc['2019':,'GOOGL'],marker="v",linestyle="--",color="g")
ax[1,0].plot(portfolio_stocks.loc['2020':,'KO'],marker="*",linestyle="--",color="r")
ax[1,1].plot(portfolio_stocks.loc['2019':,'IBM'],color="0")
ax[0,0].set_title("aapl Stocks Price")
ax[0,1].set_title("googl Stocks Price")
ax[1,0].set_title("ko Stocks Price")
ax[1,1].set_title("ibm Stocks Price")
ax[0,0].set_xlabel("Time")
ax[0,1].set_ylabel("Price")
ax[1,0].set_xlabel("Time")
ax[1,1].set_ylabel("Price")
plt.show()
```


aapl Stocks Price
130
120
110
100
90
80
70
60
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
Time
googl Stocks Price
Price
90
85
80
75
70
65
60
55
50
2019-01
2019-04
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
ko Stocks Price
52.5
50.0
47.5
45.0
42.5
40.0
37.5
35.0
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
Time
ibm Stocks Price
Price
120
110
100
90
80
2019-01
2019-04
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01

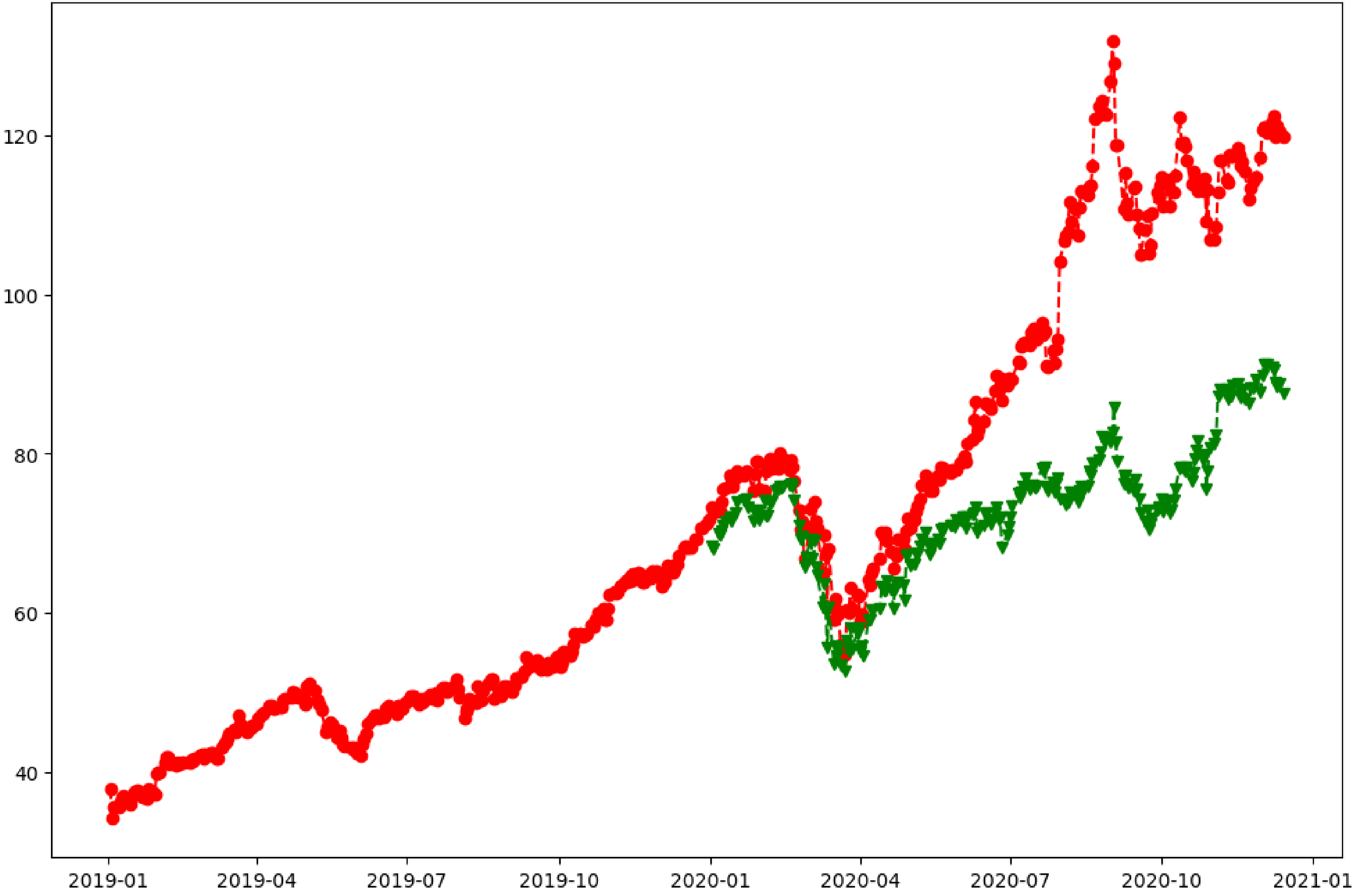
In [13]:

```
# if we want to add grid space to subplots
fig = plt.figure(figsize=(12, 8), constrained_layout=True)
spec = fig.add_gridspec(2, 2, hspace=0.025, wspace=0)

ax0 = fig.add_subplot(spec[0, :])
ax0.plot(portfolio_stocks.loc['2020':,'AAPL'],marker="o",linestyle="--",color="r")
ax0.set_title("aapl Stocks Price")
ax0.set_xlabel("Time")
ax0.set_ylabel("Price")
ax10 = fig.add_subplot(spec[1, 0])
ax10.plot(portfolio_stocks.loc['2019':,'GOOGL'],marker="v",linestyle="--",color="g")
ax10.set_title("googl Stocks Price")
ax10.set_xlabel("Time")
ax10.set_ylabel("Price")

ax11 = fig.add_subplot(spec[1, 1])
ax11.plot(portfolio_stocks.loc['2019':,'IBM'],color="0")
ax11.set_title("ibm Stocks Price")
ax11.set_xlabel("Time")
ax11.set_ylabel("Price")
plt.show()
```


aapl Stocks Price
Price
130
120
110
100
90
80
70
60
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
Time
googl Stocks Price
Price
90
85
80
75
70
65
60
55
50
2019-01
2019-04
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
Time
ibm Stocks Price
Price
120
110
100
90
80
2019-01
2019-04
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
Time

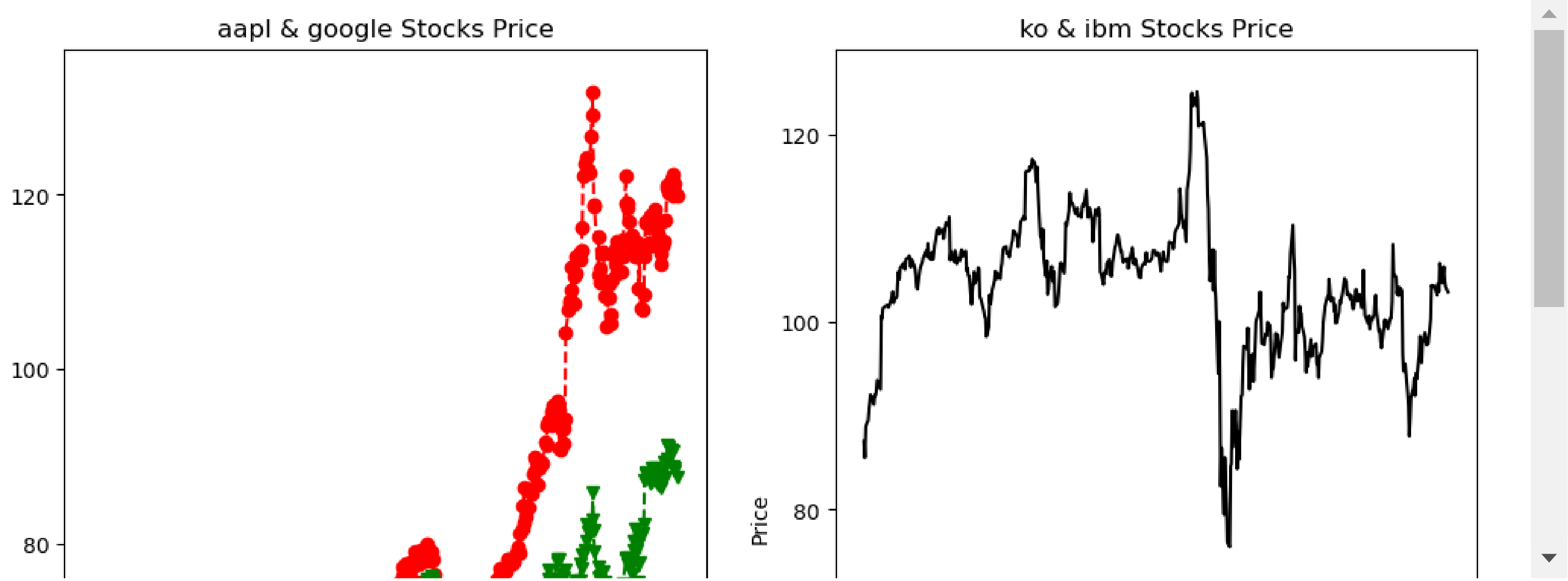
In [14]:

```
fig,ax = plt.subplots(figsize=(12,8))
ax.plot(portfolio_stocks.loc['2019':,'AAPL'],marker="o",linestyle="--",color="r");
ax.plot(portfolio_stocks.loc['2020':,'GOOGL'],marker="v",linestyle="--",color="g")
plt.show()
```



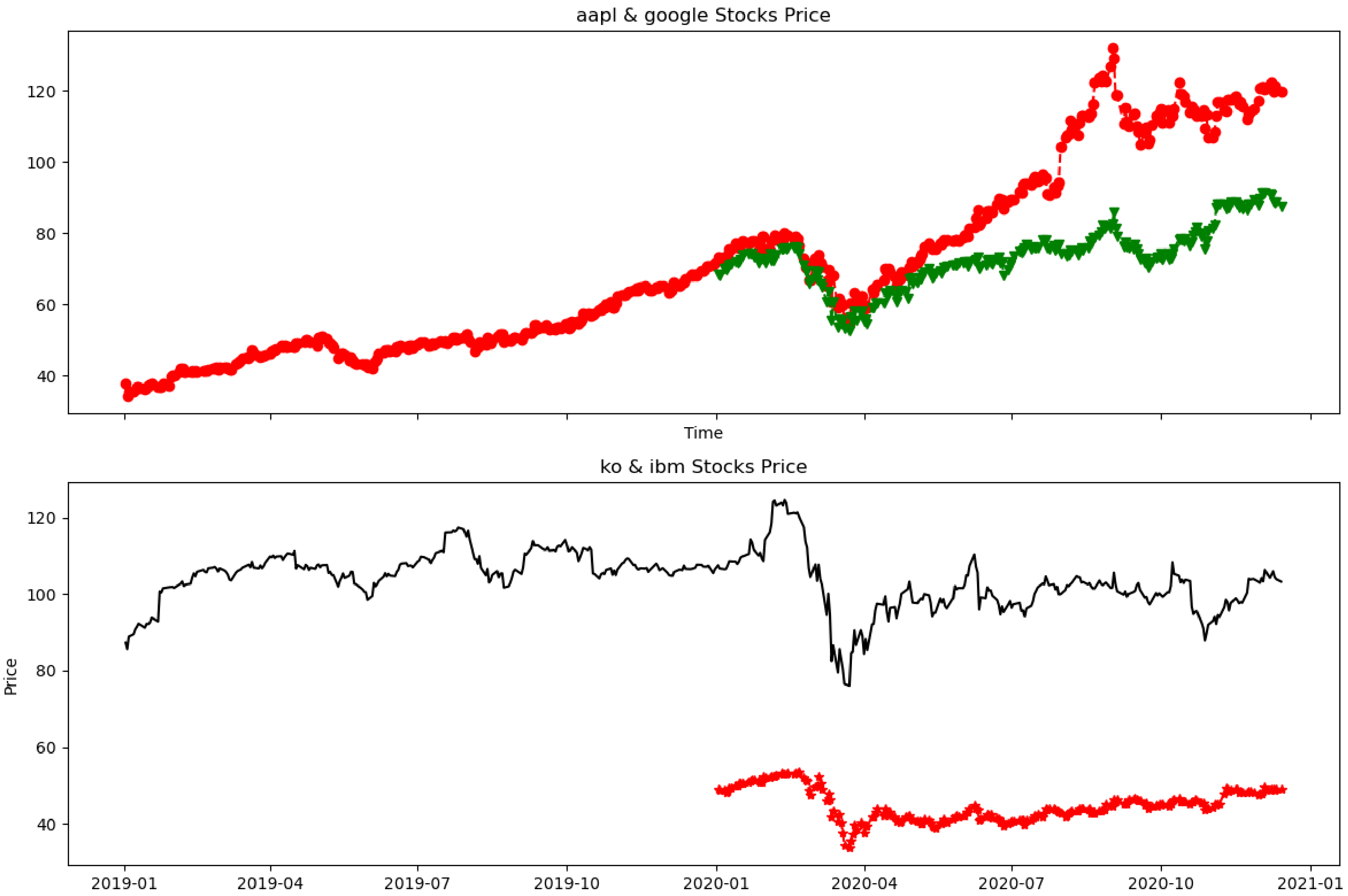
In [15]:

```python
# If we use 2 row or 2 columns, we just index charts 0 and 1
fig,ax = plt.subplots(1, 2,figsize=(12,8))
ax[0].plot(portfolio_stocks.loc['2019':,'AAPL'],marker="o",linestyle="--",color="r")
ax[0].plot(portfolio_stocks.loc['2020':,'GOOGL'],marker="v",linestyle="--",color="g");
ax[1].plot(portfolio_stocks.loc['2020':,'KO'],marker="*",linestyle="--",color="r")
ax[1].plot(portfolio_stocks.loc['2019':,'IBM'],color="0")
ax[0].set_title("aapl & google Stocks Price")
ax[1].set_title("ko & ibm Stocks Price")
ax[0].set_xlabel("Time")
ax[1].set_ylabel("Price")
plt.show()
```



In [16]:

```python
# We can share x axis in 2 rows charts, and y axis in 2 columns charts
fig, ax = plt.subplots(2, 1, sharex=True,figsize=(12,8))
ax[0].plot(portfolio_stocks.loc['2019':,'AAPL'],marker="o",linestyle="--",color="r")
ax[0].plot(portfolio_stocks.loc['2020':,'GOOGL'],marker="v",linestyle="--",color="g");
ax[1].plot(portfolio_stocks.loc['2020':,'KO'],marker="*",linestyle="--",color="r")
ax[1].plot(portfolio_stocks.loc['2019':,'IBM'],color="0")
ax[0].set_title("aapl & google Stocks Price")
ax[1].set_title("ko & ibm Stocks Price")
ax[0].set_xlabel("Time")
ax[1].set_ylabel("Price")
fig.tight_layout() # it places chart more visiable
plt.show()
```



In [17]:

```
#Using twin axes and annotation
```

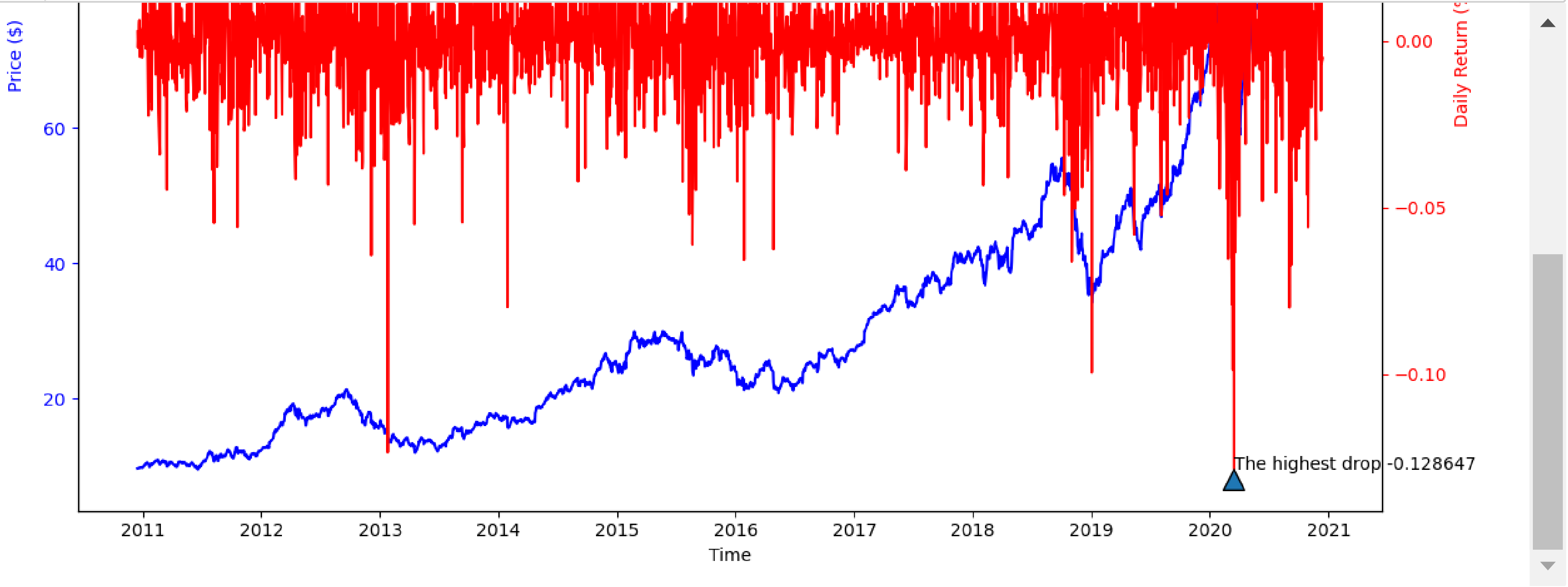
In [18]:

```
ret = portfolio_stocks[['AAPL']].pct_change()
print(ret[ret['AAPL']==ret['AAPL'].min()])
print(ret[ret['AAPL']==ret['AAPL'].max()])
```

```
                AAPL
Date
2020-03-16 -0.128647
                AAPL
Date
2020-03-13  0.119808
```

In [19]:

```
#Using twin axes
fig, ax = plt.subplots(figsize=(12,8))
ax.plot(portfolio_stocks.index, portfolio_stocks['AAPL'],'blue' )
ax.set_xlabel('Time')
ax.set_ylabel('Price ($)',color='b')
ax.tick_params('y', colors='blue')
ax2 = ax.twinx()
ax2.plot(portfolio_stocks.index,portfolio_stocks['AAPL'].pct_change(),'red')
ax2.set_xlabel('Time')
ax2.set_ylabel('Daily Return (%)',color='r')
ax2.tick_params('y', colors='red')
fig.tight_layout()
ax2.annotate("The highest drop -0.128647 ",xy=(pd.Timestamp('2020-03-16'), -0.128647),arrowprops={})
ax2.annotate("The highest increase 0.119808 ",xy=(pd.Timestamp('2020-03-13'), 0.119808),arrowprops={})
plt.show()

```



In [ ]:

```

```

In [ ]:

```

```

In [23]:

```python
df = pd.read_csv("House_Rent_Dataset.csv")
```

In [24]:

```python
df
```

Out[24]:

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	Tenant Preferred	Bathroom	Point of Contact
0	5/18/2022	2.0	10000.0	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Family	2.0	Contact Owner
1	5/13/2022	2.0	20000.0	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Family	1.0	Contact Owner
2	5/16/2022	2.0	17000.0	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bachelors/Family	1.0	Contact Owner
3	7/4/2022	2.0	10000.0	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Family	1.0	Contact Owner
4	5/9/2022	2.0	7500.0	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	Bachelors	1.0	Contact Owner
...	...	...	...	...	...	...	...	...	...	...	...	...
4744	5/18/2022	2.0	15000.0	1000	3 out of 5	Carpet Area	Bandam Kommu	Hyderabad	Semi-Furnished	Bachelors/Family	2.0	Contact Owner
4745	5/15/2022	3.0	29000.0	2000	1 out of 4	Super Area	Manikonda, Hyderabad	Hyderabad	Semi-Furnished	Bachelors/Family	3.0	Contact Owner
4746	7/10/2022	3.0	35000.0	1750	3 out of 5	Carpet Area	Himayath Nagar, NH 7	Hyderabad	Semi-Furnished	Bachelors/Family	3.0	Contact Agent
4747	7/6/2022	3.0	45000.0	1500	23 out of 34	Carpet Area	Gachibowli	Hyderabad	Semi-Furnished	Family	2.0	Contact Agent
4748	5/4/2022	2.0	15000.0	1000	4 out of 5	Carpet Area	Suchitra Circle	Hyderabad	Unfurnished	Bachelors	2.0	Contact Owner

4749 rows × 12 columns

In [25]:

```
df.columns
```

Out[25]:

```
Index(['Posted On', 'BHK', 'Rent', 'Size', 'Floor', 'Area Type',
       'Area Locality', 'City', 'Furnishing Status', 'Tenant Preferred',
       'Bathroom', 'Point of Contact'],
      dtype='object')
```

In [26]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4749 entries, 0 to 4748
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Posted On          4749 non-null   object 
 1   BHK                4748 non-null   float64
 2   Rent               4748 non-null   float64
 3   Size               4749 non-null   int64  
 4   Floor              4747 non-null   object 
 5   Area Type          4747 non-null   object 
 6   Area Locality      4746 non-null   object 
 7   City               4748 non-null   object 
 8   Furnishing Status  4746 non-null   object 
 9   Tenant Preferred   4746 non-null   object 
 10  Bathroom           4746 non-null   float64
 11  Point of Contact   4746 non-null   object 
dtypes: float64(3), int64(1), object(8)
memory usage: 445.3+ KB
```

In [27]:

```
df.describe()
```

Out[27]:

	BHK	Rent	Size	Bathroom
count	4748.000000	4.748000e+03	4749.000000	4746.000000
mean	2.084246	3.504221e+04	967.428722	1.965866
std	0.832546	7.818612e+04	634.523031	0.884532
min	1.000000	1.200000e+03	10.000000	1.000000
25%	2.000000	1.000000e+04	550.000000	1.000000
50%	2.000000	1.600000e+04	850.000000	2.000000
75%	3.000000	3.300000e+04	1200.000000	2.000000
max	6.000000	3.500000e+06	8000.000000	10.000000

In [28]:

```
df.isnull().sum()
```

Out[28]:

```
Posted On            0
BHK                  1
Rent                 1
Size                 0
Floor                2
Area Type            2
Area Locality        3
City                 1
Furnishing Status    3
Tenant Preferred     3
Bathroom             3
Point of Contact     3
dtype: int64
```

In [29]:

```
df.duplicated().sum()
```

Out[29]:

```
0
```

In [30]:

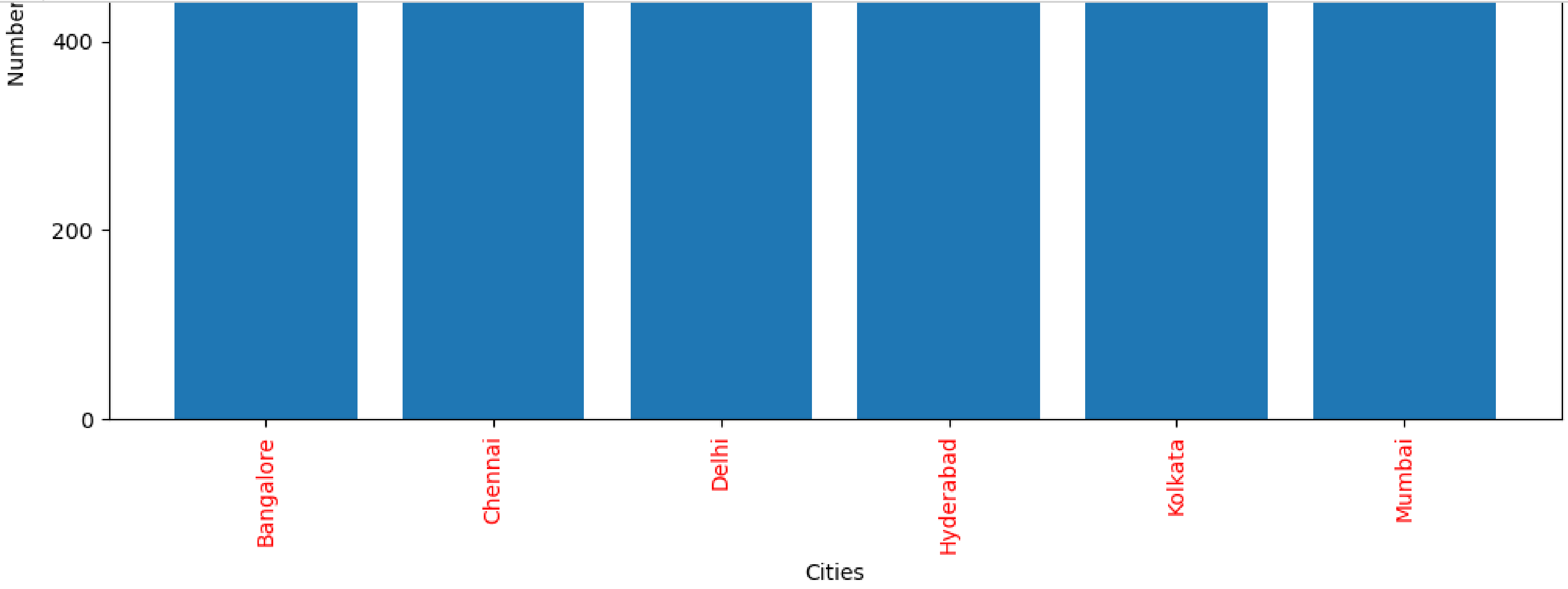
```python
number_of_posted_ad_by_cities = df.groupby('City').count()[['Rent']]
number_of_posted_ad_by_cities.columns = ['number_of_posted_ad']
number_of_posted_ad_by_cities
```

Out[30]:

	number_of_posted_ad
City	
Bangalore	886
Chennai	891
Delhi	605
Hyderabad	868
Kolkata	524
Mumbai	973

In [31]:

```
fig, ax = plt.subplots(figsize=(12,8))
ax.bar(number_of_posted_ad_by_cities.index, number_of_posted_ad_by_cities["number_of_posted_ad"])
ax.set_xlabel('Cities')
ax.set_ylabel("Number of Rental house Ad")
ax.set_xticklabels(number_of_posted_ad_by_cities.index, rotation=90,color='r')
plt.show()
```



In [32]:

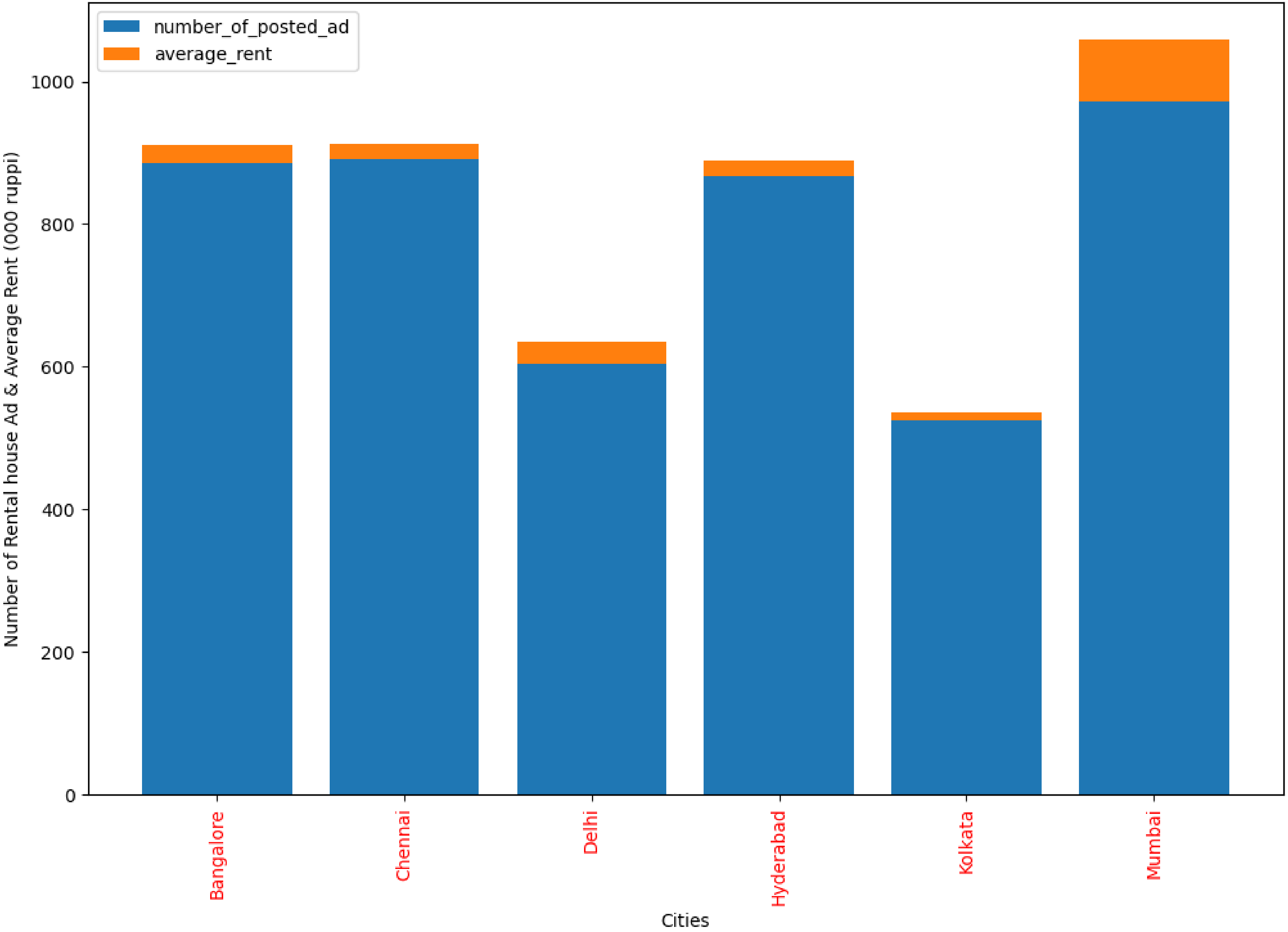
```python
average_rent_by_cities = df.groupby('City').mean()[['Rent']]
average_rent_by_cities.columns =['average_rent']
average_rent_by_cities = average_rent_by_cities/1000
average_rent_by_cities
```

Out[32]:

```
           average_rent
City
Bangalore     24.966366
Chennai       21.614092
Delhi         29.461983
Hyderabad     20.555048
Kolkata       11.645174
Mumbai        85.235058
```

In [33]:

```
fig, ax = plt.subplots(figsize=(12,8))
ax.bar(number_of_posted_ad_by_cities.index, number_of_posted_ad_by_cities["number_of_posted_ad"],label='number_of
ax.bar(average_rent_by_cities.index, average_rent_by_cities["average_rent"],
       bottom=number_of_posted_ad_by_cities["number_of_posted_ad"],label='average_rent')
ax.set_xlabel('Cities')
ax.set_ylabel("Number of Rental house Ad & Average Rent (000 ruppi)")
ax.set_xticklabels(number_of_posted_ad_by_cities.index, rotation=90,color='r')
ax.legend()
plt.show()
```


number_of_posted_ad
average_rent
Number of Rental house Ad & Average Rent (000 ruppi)
0
200
400
600
800
1000
Bangalore
Chennai
Delhi
Hyderabad
Kolkata
Mumbai
Cities

In [34]:

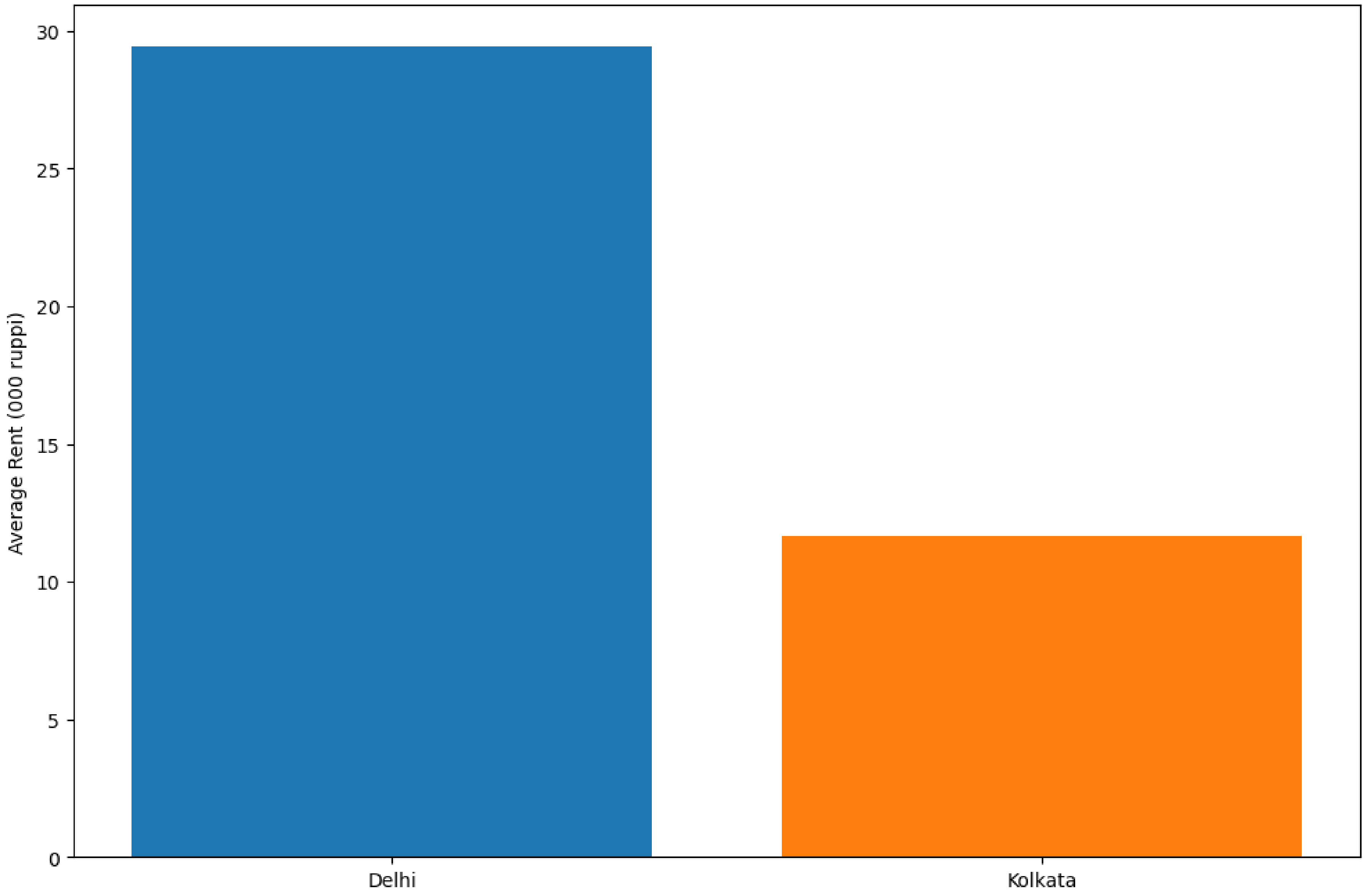
```
average_rent_by_cities.loc['Delhi']
```

Out[34]:

```
average_rent    29.461983
Name: Delhi, dtype: float64
```

In [35]:

```python
fig, ax = plt.subplots(figsize=(12,8))
ax.bar("Delhi", average_rent_by_cities.loc['Delhi'])
ax.bar("Kolkata", average_rent_by_cities.loc['Kolkata'])
ax.set_ylabel("Average Rent (000 ruppi)")
plt.show()
```


30
25
20
15
10
5
0
Average Rent (000 ruppi)
Delhi
Kolkata

In [36]:

```
df[df['City']=='Delhi'][['Size']]
```

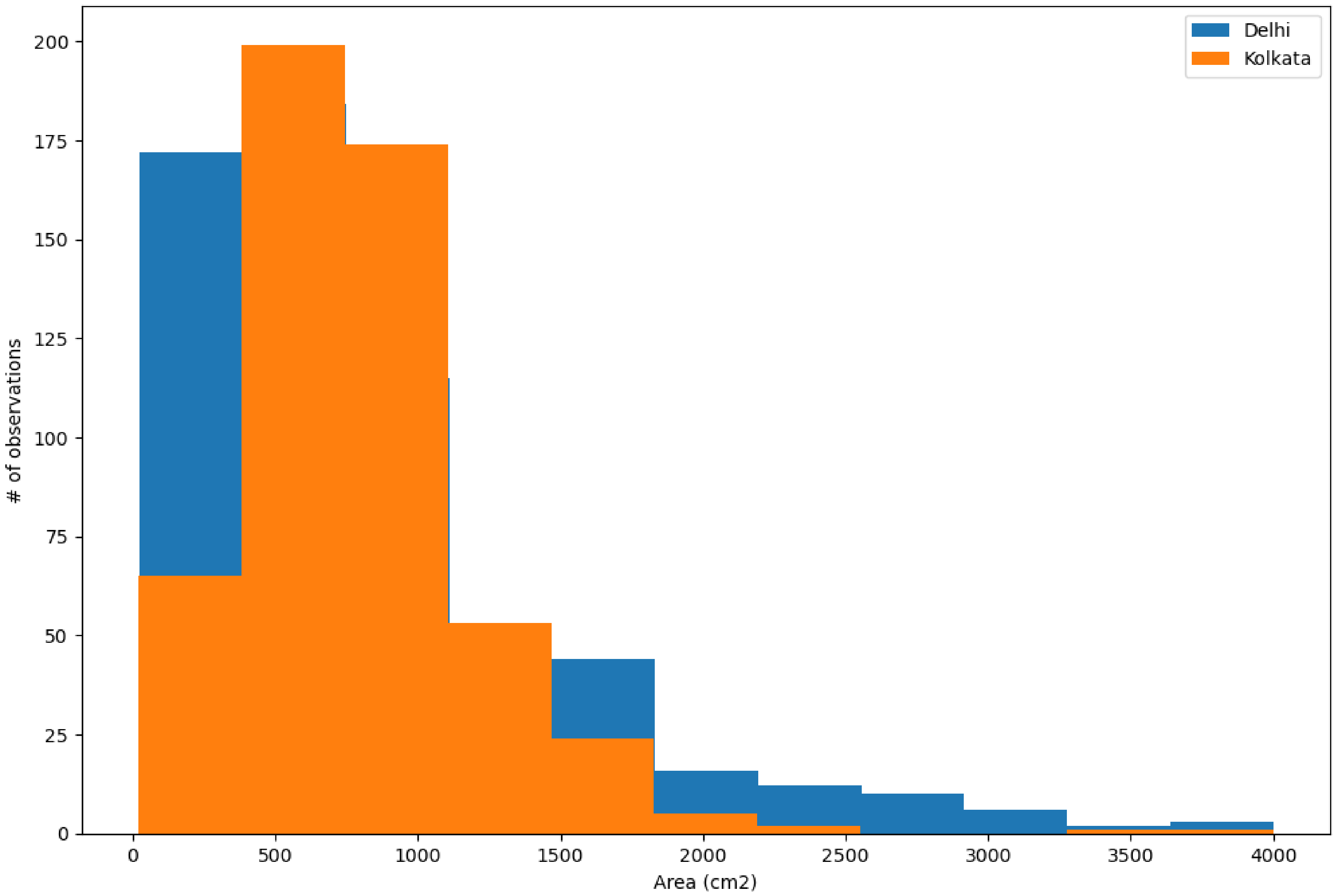
Out[36]:

	Size
2385	800
2386	200
2387	1800
2388	400
2389	600
...	...
2985	1200
2986	250
2987	700
2988	1050
2989	500

605 rows × 1 columns

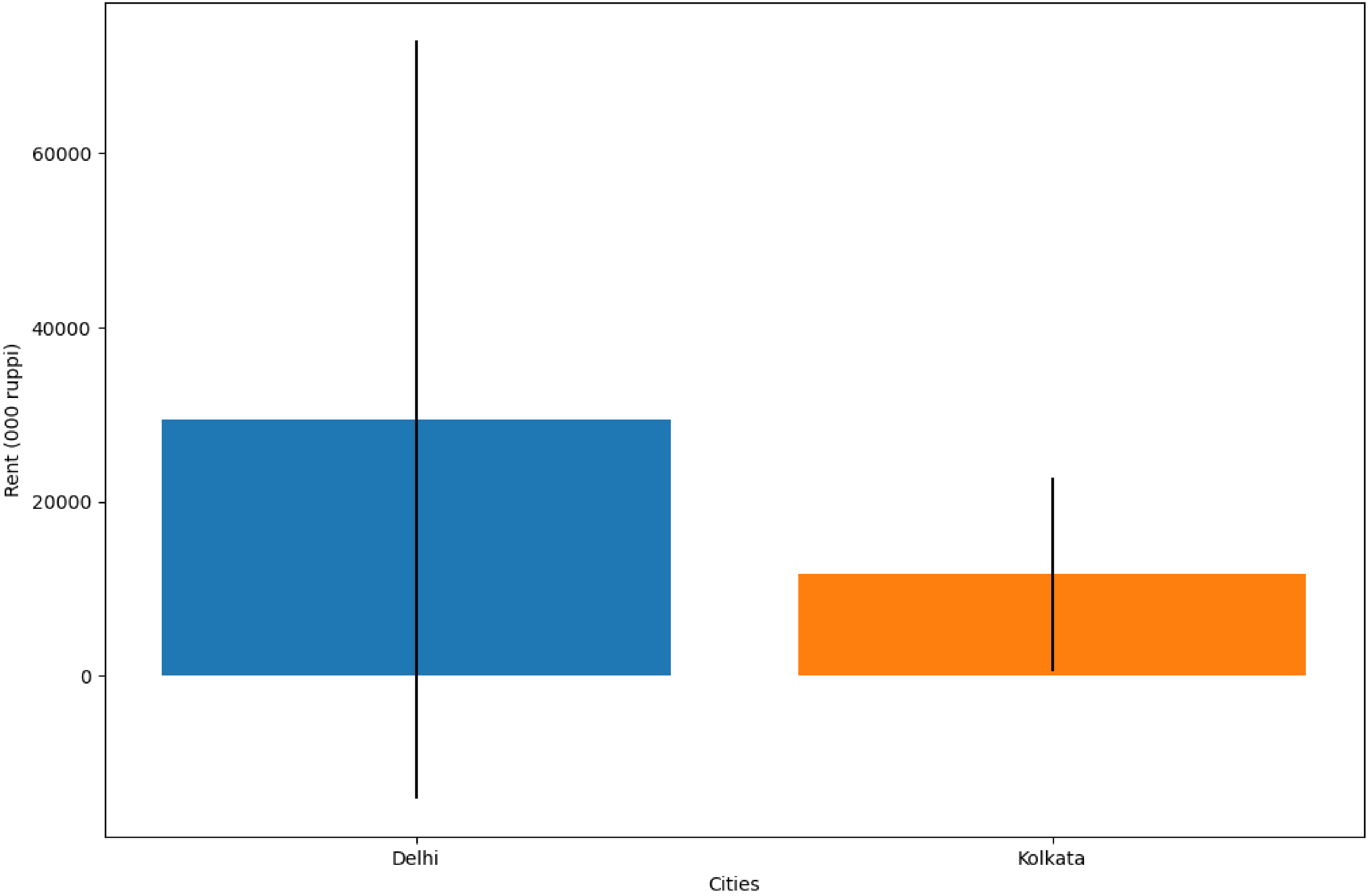
In [37]:

```
fig, ax = plt.subplots(figsize=(12,8))
ax.hist(df[df['City']=='Delhi'][['Size']],label='Delhi',bins=11)
ax.hist(df[df['City']=='Kolkata'][['Size']],label='Kolkata',bins=11)
ax.set_xlabel("Area (cm2)")
ax.set_ylabel("# of observations")
ax.legend()
plt.show()
```


🟦 Delhi
🟧 Kolkata
# of observations
200
175
150
125
100
75
50
25
0
0
500
1000
1500
2000
2500
3000
3500
4000
Area (cm2)

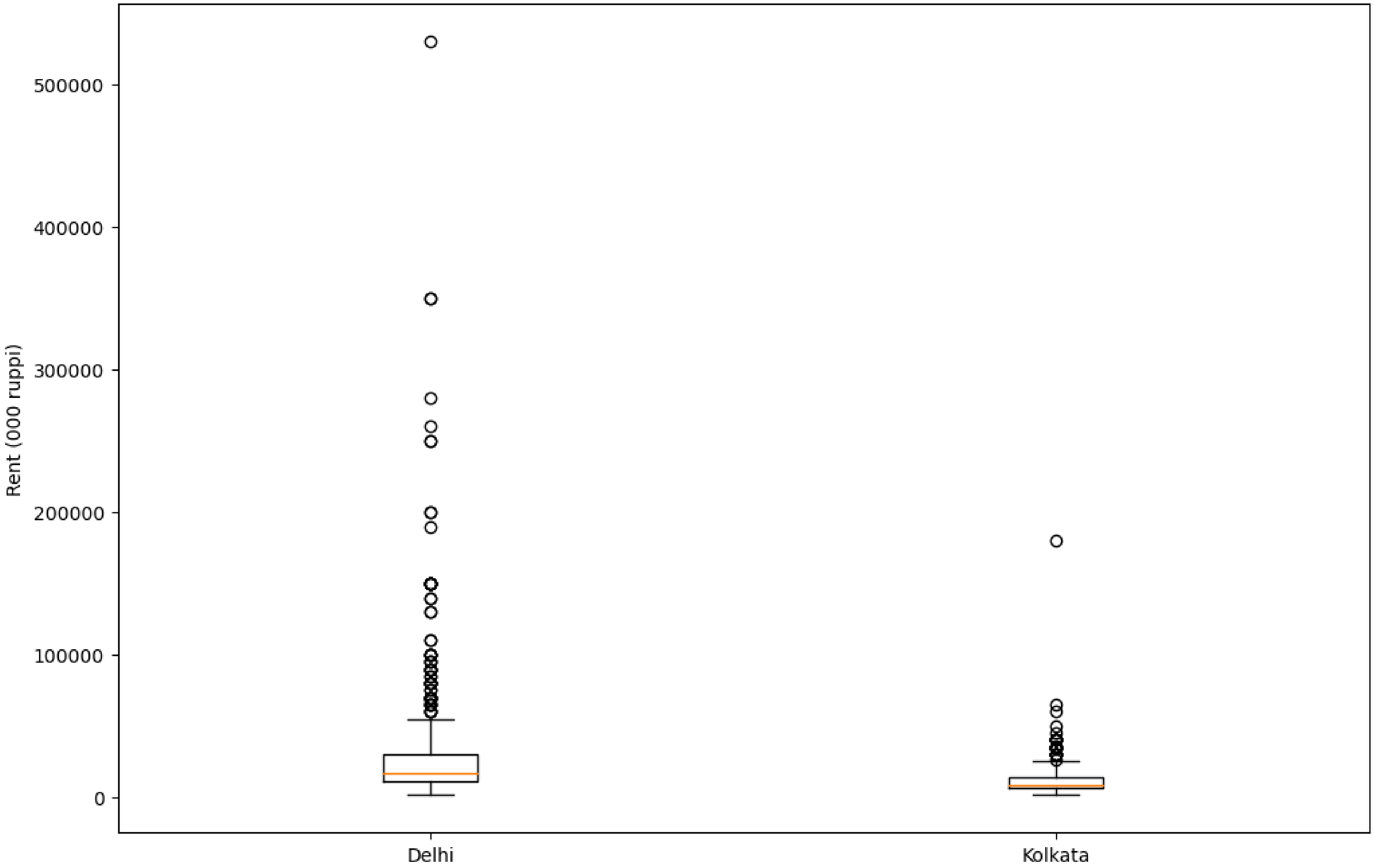
In [38]:

```python
fig, ax = plt.subplots(figsize=(12,8))
ax.bar('Delhi',df[df['City']=='Delhi'][['Rent']].mean(),yerr =df[df['City']=='Delhi'][['Rent']].std() )
ax.bar('Kolkata',df[df['City']=='Kolkata'][['Rent']].mean(),yerr =df[df['City']=='Kolkata'][['Rent']].std())
ax.set_xlabel("Cities")
ax.set_ylabel("Rent (000 ruppi)")

plt.show()
```


60000
40000
20000
0
Rent (000 ruppi)
Delhi
Kolkata
Cities

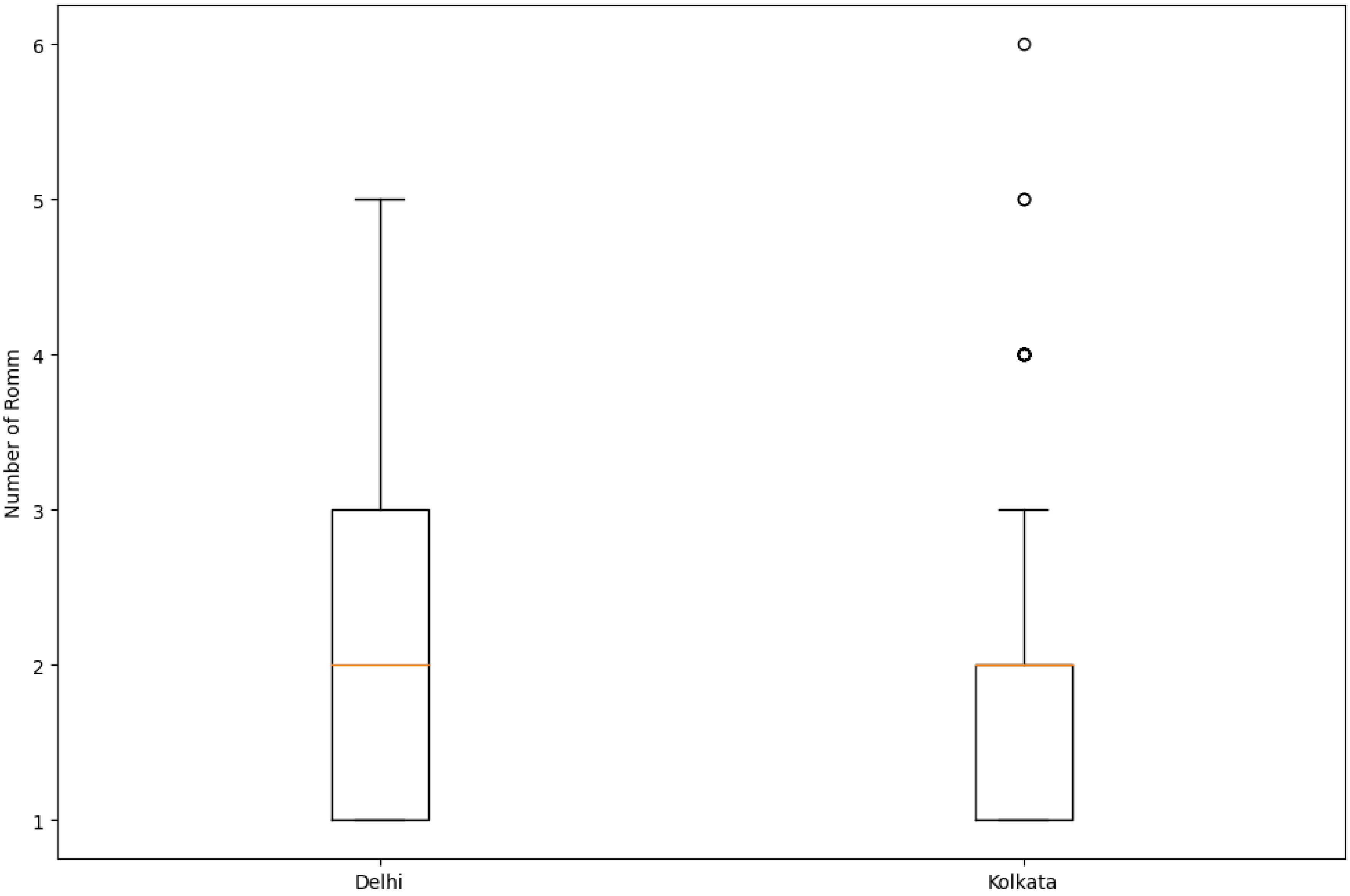
In [39]:

```
fig, ax = plt.subplots(figsize=(12,8))
ax.boxplot([df[df['City']=='Delhi']['Rent'],df[df['City']=='Kolkata']['Rent']])
ax.set_xticklabels(["Delhi","Kolkata"])
ax.set_ylabel("Rent (000 ruppi)")
plt.show()
```



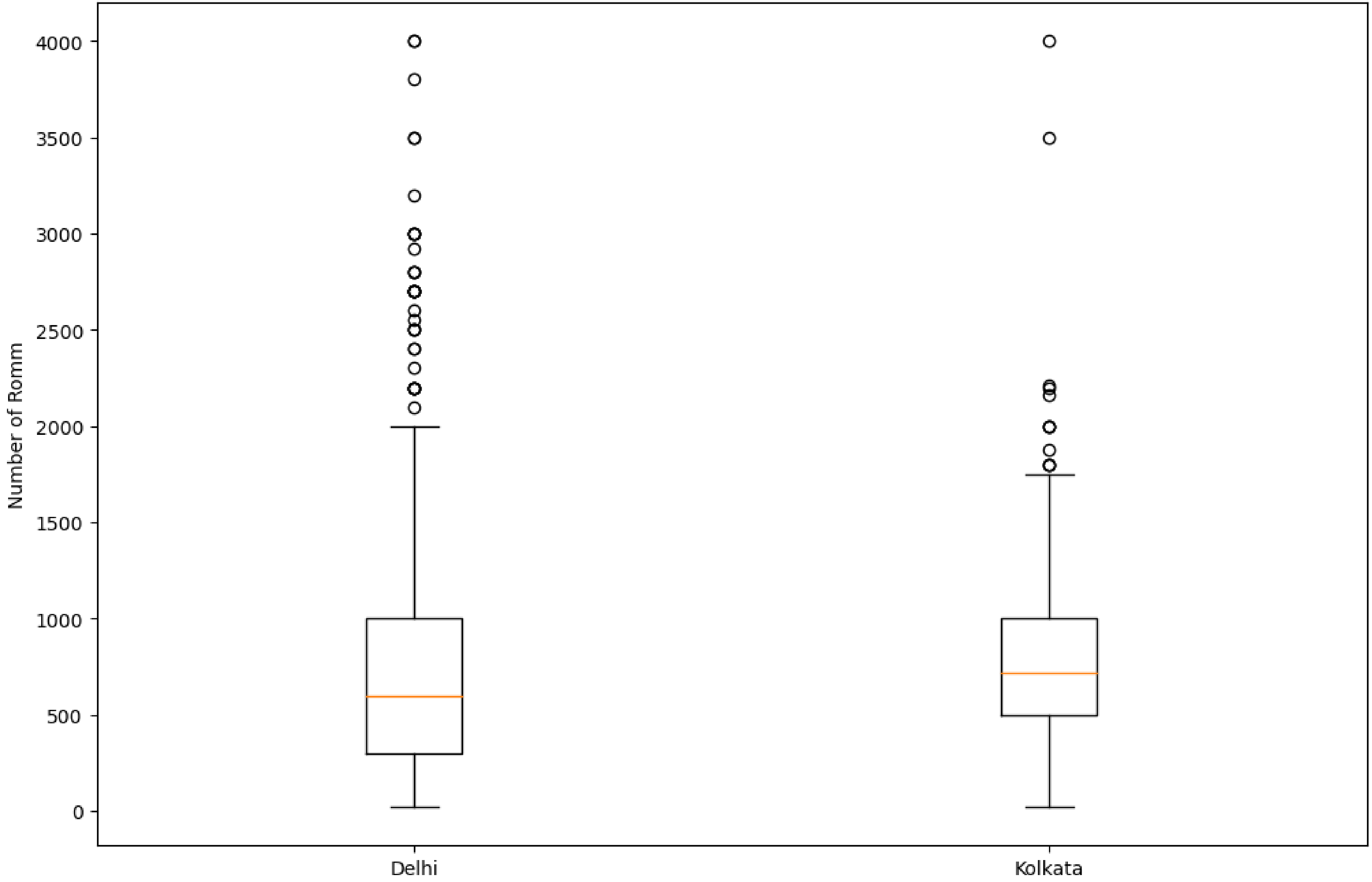
In [40]:

```
fig, ax = plt.subplots(figsize=(12,8))
ax.boxplot([df[df['City']=='Delhi']['BHK'],df[df['City']=='Kolkata']['BHK']])
ax.set_xticklabels(["Delhi","Kolkata"])
ax.set_ylabel("Number of Romm")
plt.show()
```


Number of Romm
6
5
4
3
2
1
Delhi
Kolkata

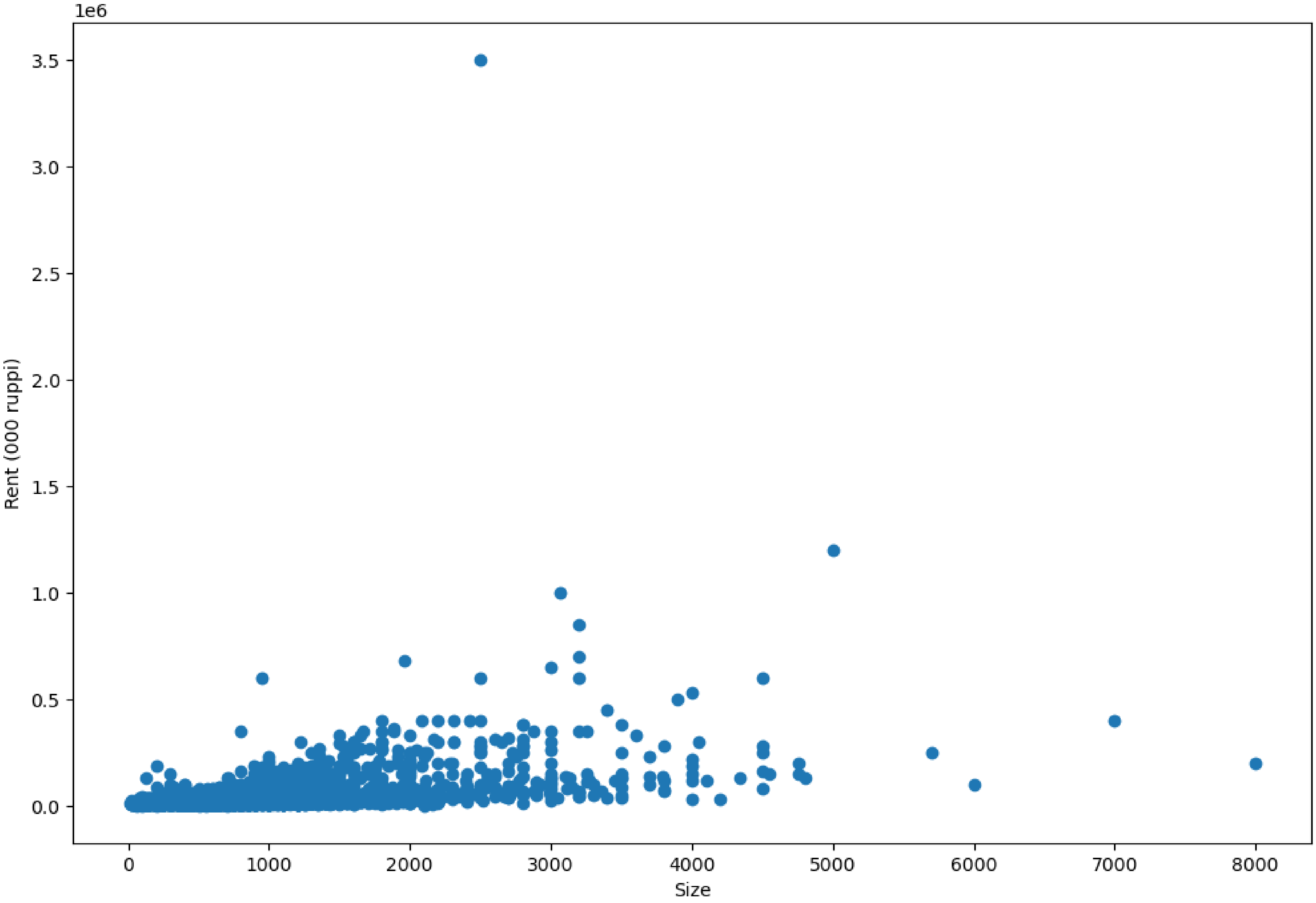
In [41]:

```
fig, ax = plt.subplots(figsize=(12,8))
ax.boxplot([df[df['City']=='Delhi']['Size'],df[df['City']=='Kolkata']['Size']])
ax.set_xticklabels(["Delhi","Kolkata"])
ax.set_ylabel("Number of Romm")
plt.show()
```



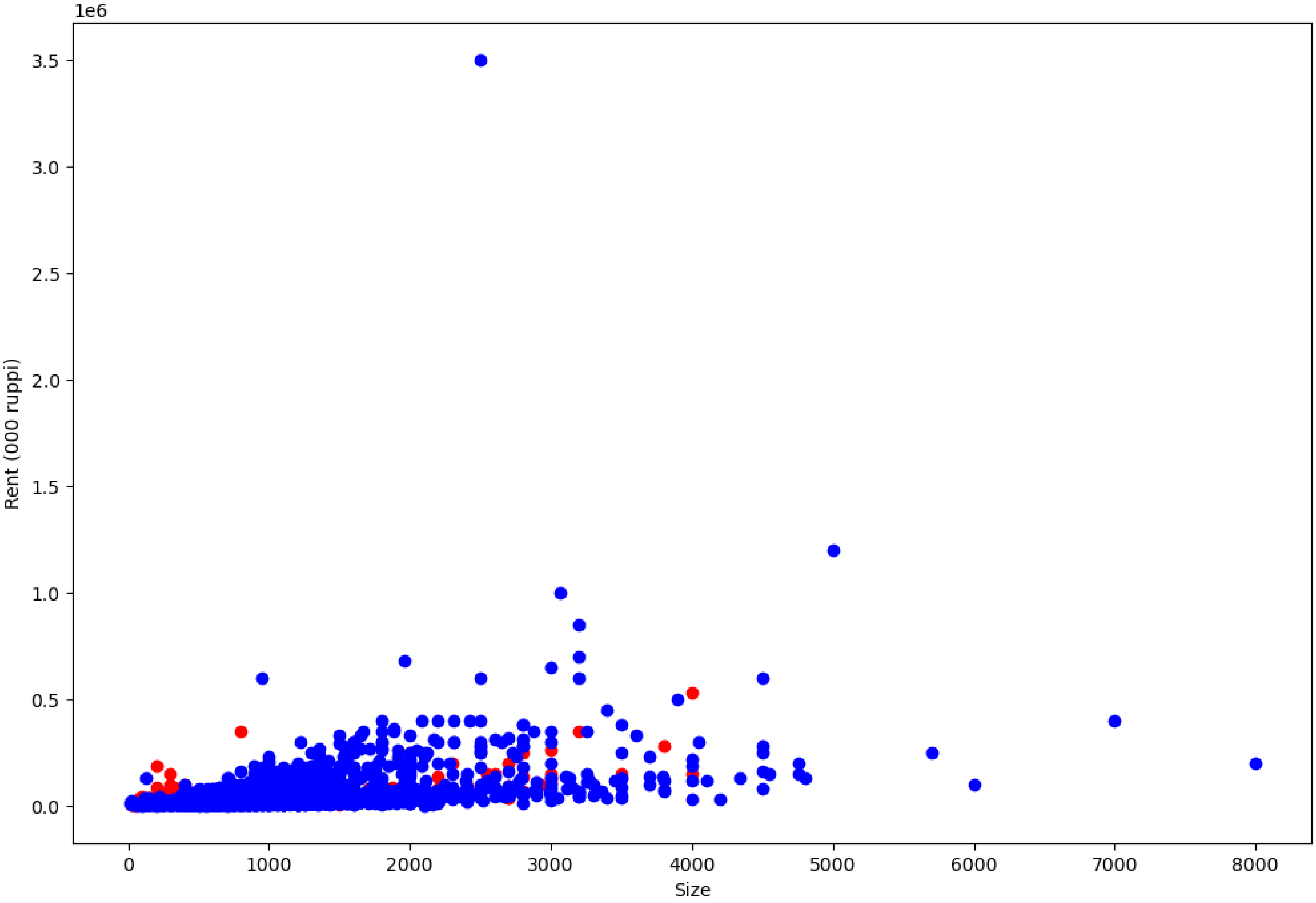
In [42]:

```
fig, ax = plt.subplots(figsize=(12,8))
ax.scatter(df['Size'], df['Rent'])
ax.set_xlabel("Size")
ax.set_ylabel("Rent (000 ruppi)")
plt.show()
```


1e6
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Rent (000 ruppi)
0
1000
2000
3000
4000
5000
6000
7000
8000
Size

In [43]:

```
fig, ax = plt.subplots(figsize=(12,8))
ax.scatter(df[df['City']=='Delhi']['Size'],df[df['City']=='Delhi']['Rent'],label='Delhi',color='r')
ax.scatter(df[df['City']!='Delhi']['Size'], df[df['City']!='Delhi']['Rent'],label='Cities other than Delhi',color
ax.set_xlabel("Size")
ax.set_ylabel("Rent (000 ruppi)")
plt.show()
```


1e6
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Rent (000 ruppi)
0
1000
2000
3000
4000
5000
6000
7000
8000
Size

In [ ]:

```

```