In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('C:\\Users\\Anirudh\\Downloads\\TCS1.csv')
df
```

Out[2]:

| | Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|---|
| 0 | 2004-08-27 | 122.800003 | 122.800003 | 119.820000 | 120.332497 | 88.080272 | 30646000.0 |
| 1 | 2004-08-30 | 121.237503 | 123.750000 | 120.625000 | 123.340001 | 90.293548 | 24485208.0 |
| 2 | 2004-08-31 | 123.312500 | 123.750000 | 122.000000 | 123.512497 | 90.416122 | 21194656.0 |
| 3 | 2004-09-01 | 123.750000 | 124.375000 | 122.949997 | 123.497503 | 90.397820 | 19935644.0 |
| 4 | 2004-09-02 | 123.737503 | 125.574997 | 123.250000 | 124.297497 | 90.924898 | 21356352.0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 4489 | 2022-10-18 | 3150.000000 | 3195.350098 | 3128.550049 | 3144.699951 | 3144.699951 | 1793722.0 |
| 4490 | 2022-10-19 | 3159.000000 | 3159.000000 | 3112.000000 | 3121.850098 | 3121.850098 | 1194289.0 |
| 4491 | 2022-10-20 | 3165.000000 | 3160.000000 | 3105.000000 | 3157.300049 | 3157.300049 | 1587601.0 |
| 4492 | 2022-10-21 | 3157.800049 | 3160.399902 | 3127.000000 | 3137.399902 | 3137.399902 | 1021913.0 |
| 4493 | 2022-10-24 | 3170.100098 | 3178.000000 | 3155.000000 | 3161.699951 | 3161.699951 | 2609449.0 |

4494 rows × 7 columns

In [3]:
```
df.isnull().sum()
```

Out[3]:
```
Date         0
Open         8
High         8
Low          8
Close        8
Adj Close    8
Volume       8
dtype: int64
```

In [4]:
```
df['Open'] = df['Open'].fillna(df['Open'].mean())
df['High'] = df['High'].fillna(df['High'].mean())
df['Low'] = df['Low'].fillna(df['Low'].mean())
df['Close'] = df['Close'].fillna(df['Close'].mean())
df['Adj Close'] = df['Adj Close'].fillna(df['Adj Close'].mean())
df['Volume'] = df['Volume'].fillna(df['Volume'].mean())
```

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
Date         0
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

In [6]:
```
df.duplicated().sum()
```

Out[6]:
```
0
```

In [7]:
```
df.corr()
```

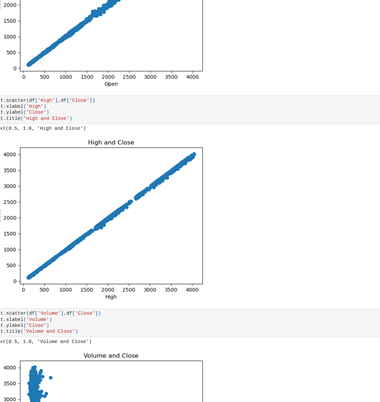
```
C:\Users\Anirudh\AppData\Local\Temp\ipykernel_18104\1134722465.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  df.corr()
```

Out[7]:

| | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|
| **Open** | 1.000000 | 0.999893 | 0.999691 | 0.999779 | 0.998755 | -0.235007 |
| **High** | 0.999893 | 1.000000 | 0.999877 | 0.999913 | 0.998893 | -0.232812 |
| **Low** | 0.999691 | 0.999877 | 1.000000 | 0.999905 | 0.998864 | -0.236762 |
| **Close** | 0.999779 | 0.999913 | 0.999905 | 1.000000 | 0.998958 | -0.234861 |
| **Adj Close** | 0.998755 | 0.998893 | 0.998864 | 0.998958 | 1.000000 | -0.221112 |
| **Volume** | -0.235007 | -0.232812 | -0.236762 | -0.234861 | -0.221112 | 1.000000 |

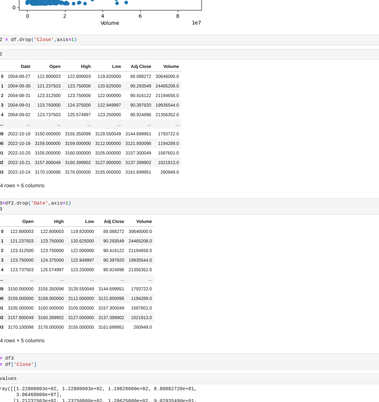
In [8]:
```
plt.figure(figsize=(10, 8))  # Optional: Set the figure size
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

# Optional: Add a title to the plot
plt.title("Correlation Heatmap")

# Display the plot
plt.show()
```

```
C:\Users\Anirudh\AppData\Local\Temp\ipykernel_18104\3968757405.py:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
```



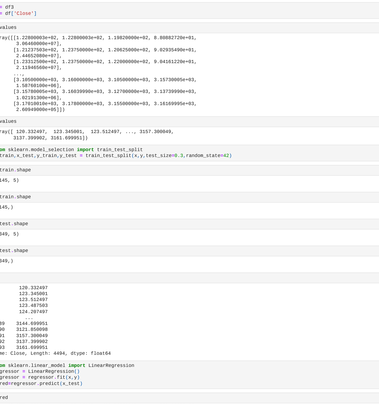

In [9]:
```
plt.scatter(df['Open'],df['Close'])
plt.xlabel('Open')
plt.ylabel('Close')
plt.title('open and Close')
```

Out[9]:
```
Text(0.5, 1.0, 'open and Close')
```



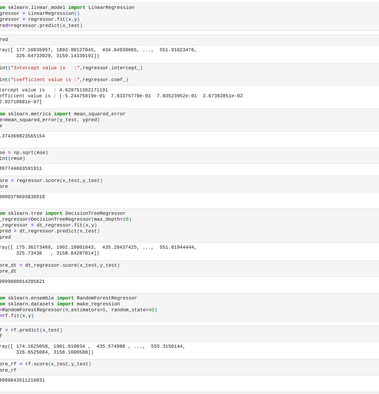

In [10]:
```
plt.scatter(df['High'],df['Close'])
plt.xlabel('High')
plt.ylabel('Close')
plt.title('High and Close')
```

Out[10]:
```
Text(0.5, 1.0, 'High and Close')
```




In [11]:
```
plt.scatter(df['Volume'],df['Close'])
plt.xlabel('Volume')
plt.ylabel('Close')
plt.title('Volume and Close')
```

Out[11]:
```
Text(0.5, 1.0, 'Volume and Close')
```




In [12]:
```
df2 = df.drop('Close',axis=1)
```

In [13]:
```
df2
```

Out[13]:

| | Date | Open | High | Low | Adj Close | Volume |
|---|---|---|---|---|---|---|
| 0 | 2004-08-27 | 122.800003 | 122.800003 | 119.820000 | 88.080272 | 30646000.0 |
| 1 | 2004-08-30 | 121.237503 | 123.750000 | 120.625000 | 90.293548 | 24485208.0 |
| 2 | 2004-08-31 | 123.312500 | 123.750000 | 122.000000 | 90.416122 | 21194656.0 |
| 3 | 2004-09-01 | 123.750000 | 124.375000 | 122.949997 | 90.397820 | 19935644.0 |
| 4 | 2004-09-02 | 123.737503 | 125.574997 | 123.250000 | 90.924898 | 21356352.0 |
| ... | ... | ... | ... | ... | ... | ... |
| 4489 | 2022-10-18 | 3150.000000 | 3195.350098 | 3128.550049 | 3144.699951 | 1793722.0 |
| 4490 | 2022-10-19 | 3159.000000 | 3159.000000 | 3112.000000 | 3121.850098 | 1194289.0 |
| 4491 | 2022-10-20 | 3165.000000 | 3160.000000 | 3105.000000 | 3157.300049 | 1587601.0 |
| 4492 | 2022-10-21 | 3157.800049 | 3160.399902 | 3127.000000 | 3137.399902 | 1021913.0 |
| 4493 | 2022-10-24 | 3170.100098 | 3178.000000 | 3155.000000 | 3161.699951 | 2609449.0 |

4494 rows × 6 columns

In [14]:
```
df3=df2.drop('Date',axis=1)
df3
```

Out[14]:

| | Open | High | Low | Adj Close | Volume |
|---|---|---|---|---|---|
| 0 | 122.800003 | 122.800003 | 119.820000 | 88.080272 | 30646000.0 |
| 1 | 121.237503 | 123.750000 | 120.625000 | 90.293548 | 24485208.0 |
| 2 | 123.312500 | 123.750000 | 122.000000 | 90.416122 | 21194656.0 |
| 3 | 123.750000 | 124.375000 | 122.949997 | 90.397820 | 19935644.0 |
| 4 | 123.737503 | 125.574997 | 123.250000 | 90.924898 | 21356352.0 |
| ... | ... | ... | ... | ... | ... |
| 4489 | 3150.000000 | 3195.350098 | 3128.550049 | 3144.699951 | 1793722.0 |
| 4490 | 3159.000000 | 3159.000000 | 3112.000000 | 3121.850098 | 1194289.0 |
| 4491 | 3165.000000 | 3160.000000 | 3105.000000 | 3157.300049 | 1587601.0 |
| 4492 | 3157.800049 | 3160.399902 | 3127.000000 | 3137.399902 | 1021913.0 |
| 4493 | 3170.100098 | 3178.000000 | 3155.000000 | 3161.699951 | 2609449.0 |

4494 rows × 5 columns

In [15]:
```
x = df3
y = df['Close']
```

In [16]:
```
x.values
```

Out[16]:
```
array([[1.22800003e+02, 1.22800003e+02, 1.19820000e+02, 8.80882720e+01,
        3.06460000e+07],
       [1.21237503e+02, 1.23750000e+02, 1.20625000e+02, 9.02935490e+01,
        2.44852080e+07],
       [1.23312500e+02, 1.23750000e+02, 1.22000000e+02, 9.04161220e+01,
        2.11946560e+07],
       ...,
       [3.16500000e+03, 3.16000000e+03, 3.10500000e+03, 3.15730005e+03,
        1.58760100e+06],
       [3.15700005e+03, 3.16039990e+03, 3.12700000e+03, 3.13739990e+03,
        1.02191300e+06],
       [3.17010010e+03, 3.17800000e+03, 3.15500000e+03, 3.16169955e+03,
        2.60944900e+06]])
```

In [17]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=42)
```

In [18]:
```
x_train.shape
```

Out[18]:
```
(3145, 5)
```

In [19]:
```
y_train.shape
```

Out[19]:
```
(3145,)
```

In [20]:
```
x_test.shape
```

Out[20]:
```
(1349, 5)
```

In [21]:
```
y_test.shape
```

Out[21]:
```
(1349,)
```

In [29]:
```
y
```

Out[29]:
```
0       120.332497
1       123.340001
2       123.512497
3       123.497503
4       124.297497
           ...
4489    3144.699951
4490    3121.850098
4491    3157.300049
4492    3137.399902
4493    3161.699951
Name: Close, Length: 4494, dtype: float64
```

In [23]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor = regressor.fit(x,y)
ypred=regressor.predict(x_test)
```

In [22]:
```
ypred
```

Out[22]:
```
array([ 177.10839957,  1893.98127045,   434.94930065, ...,  551.91023476,
        326.64733929,  3159.14339191])
```

In [23]:
```
print("Intercept value is   ",regressor.intercept_)
```

```
print("coefficient value is ",regressor.coef_)
```

```
Intercept value is   -4.629751362171191
coefficient value is  [-5.24475019e-01  7.83375779e-01  7.83523952e-01  3.67382851e-02
 -2.92726561e-07]
```

In [24]:
```
from sklearn.metrics import mean_squared_error
mse=mean_squared_error(y_test, ypred)
mse
```

Out[24]:
```
62.37436962356154
```

In [25]:
```
rmse = np.sqrt(mse)
print(rmse)
```

```
7.897744603591911
```

In [26]:
```
score = regressor.score(x_test,y_test)
score
```

Out[26]:
```
0.9999379693838918
```

In [27]:
```
from sklearn.tree import DecisionTreeRegressor
dt_regressor=DecisionTreeRegressor(max_depth=10)
dt_regressor = dt_regressor.fit(x,y)
y_pred = dt_regressor.predict(x_test)
y_pred
```

Out[27]:
```
array([ 175.36273469,  1902.10001043,   435.28437425, ...,  551.81944444,
        325.73438   ,  3158.84287814])
```

In [28]:
```
score_dt = dt_regressor.score(x_test,y_test)
score_dt
```

Out[28]:
```
0.9999880914205621
```

In [29]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.datasets import make_regression
rf=RandomForestRegressor(n_estimators=5, random_state=42)
rf=rf.fit(x,y)
```

In [30]:
```
yrf = rf.predict(x_test)
yrf
```

Out[30]:
```
array([ 174.1625058,  1901.010034 ,  435.574988 , ...,  555.3150144,
        326.6525064,  3158.1600588])
```

In [31]:
```
score_rf = rf.score(x_test,y_test)
score_rf
```

Out[31]:
```
0.9999843511216031
```

In [ ]:
```

```

In [ ]:
```

```