

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sn
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
amazon_data = pd.read_csv('C:/Users/amits/amazon_stock_price.csv')
```

In [3]:

```
amazon_data
```

Out[3]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	15-05-1997	2.437500	2.500000	1.927083	1.958333	1.958333	72156000
1	16-05-1997	1.968750	1.979167	1.708333	1.729167	1.729167	14700000
2	19-05-1997	1.760417	1.770833	1.625000	1.708333	1.708333	6106800
3	20-05-1997	1.729167	1.750000	1.635417	1.635417	1.635417	5467200
4	21-05-1997	1.635417	1.645833	1.375000	1.427083	1.427083	18853200
...	...	...	...	...	...	...	...
6120	09-09-2021	3526.020020	3549.989990	3480.370117	3484.159912	3484.159912	2719200
6121	10-09-2021	3501.830078	3508.449951	3462.909912	3469.149902	3469.149902	2393300
6122	13-09-2021	3482.800049	3497.959961	3438.000000	3457.169922	3457.169922	2569000
6123	14-09-2021	3475.550049	3486.810059	3437.709961	3450.000000	3450.000000	1936900
6124	15-09-2021	3442.520020	3485.149902	3402.010010	3475.790039	3475.790039	2921019

6125 rows × 7 columns

In [4]:

```
amazon_data.head()
```

Out[4]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	15-05-1997	2.437500	2.500000	1.927083	1.958333	1.958333	72156000
1	16-05-1997	1.968750	1.979167	1.708333	1.729167	1.729167	14700000
2	19-05-1997	1.760417	1.770833	1.625000	1.708333	1.708333	6106800
3	20-05-1997	1.729167	1.750000	1.635417	1.635417	1.635417	5467200
4	21-05-1997	1.635417	1.645833	1.375000	1.427083	1.427083	18853200

In [5]:

```
amazon_data.tail()
```

Out[5]:

	Date	Open	High	Low	Close	Adj Close	Volume
6120	09-09-2021	3526.020020	3549.989990	3480.370117	3484.159912	3484.159912	2719200
6121	10-09-2021	3501.830078	3508.449951	3462.909912	3469.149902	3469.149902	2393300
6122	13-09-2021	3482.800049	3497.959961	3438.000000	3457.169922	3457.169922	2569000

	Date	Open	High	Low	Close	Adj Close	Volume
6123	14-09-2021	3475.550049	3486.810059	3437.709961	3450.000000	3450.000000	1936900
6124	15-09-2021	3442.520020	3485.149902	3402.010010	3475.790039	3475.790039	2921019

In [6]: `amazon_data.shape`

Out[6]: (6125, 7)

In [7]: `amazon_data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6125 entries, 0 to 6124
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Date        6125 non-null    object 
 1   Open         6125 non-null    float64
 2   High         6125 non-null    float64
 3   Low          6125 non-null    float64
 4   Close        6125 non-null    float64
 5   Adj Close    6125 non-null    float64
 6   Volume       6125 non-null    int64  
dtypes: float64(5), int64(1), object(1)
memory usage: 335.1+ KB
```

In [8]: `amazon_data.describe()`

	Open	High	Low	Close	Adj Close	Volume
<b>count</b>	6125.000000	6125.000000	6125.000000	6125.000000	6125.000000	6.125000e+03
<b>mean</b>	506.711946	512.263599	500.544531	506.589874	506.589874	7.351046e+06
<b>std</b>	836.045119	844.598987	826.233207	835.548822	835.548822	7.159739e+06
<b>min</b>	1.406250	1.447917	1.312500	1.395833	1.395833	4.872000e+05
<b>25%</b>	38.619999	39.439999	38.049999	38.720001	38.720001	3.600100e+06
<b>50%</b>	91.800003	93.400002	89.750000	91.300003	91.300003	5.483600e+06
<b>75%</b>	512.989990	519.409973	502.570007	511.890015	511.890015	8.310500e+06
<b>max</b>	3744.000000	3773.080078	3696.790039	3731.409912	3731.409912	1.043292e+08

In [9]: `amazon_data.columns`

Out[9]: Index(['Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')

In [10]: `amazon_data.isnull().sum()`

Date	0
Open	0
High	0
Low	0
Close	0
Adj Close	0

```
Volume ..... 0
dtype: int64
```

In [11]:

```
amazon_data = amazon_data.drop(columns=['Adj Close'])
```

In [12]:

```
amazon_data.head()
```

Out[12]:

	Date	Open	High	Low	Close	Volume
0	15-05-1997	2.437500	2.500000	1.927083	1.958333	72156000
1	16-05-1997	1.968750	1.979167	1.708333	1.729167	14700000
2	19-05-1997	1.760417	1.770833	1.625000	1.708333	6106800
3	20-05-1997	1.729167	1.750000	1.635417	1.635417	5467200
4	21-05-1997	1.635417	1.645833	1.375000	1.427083	18853200

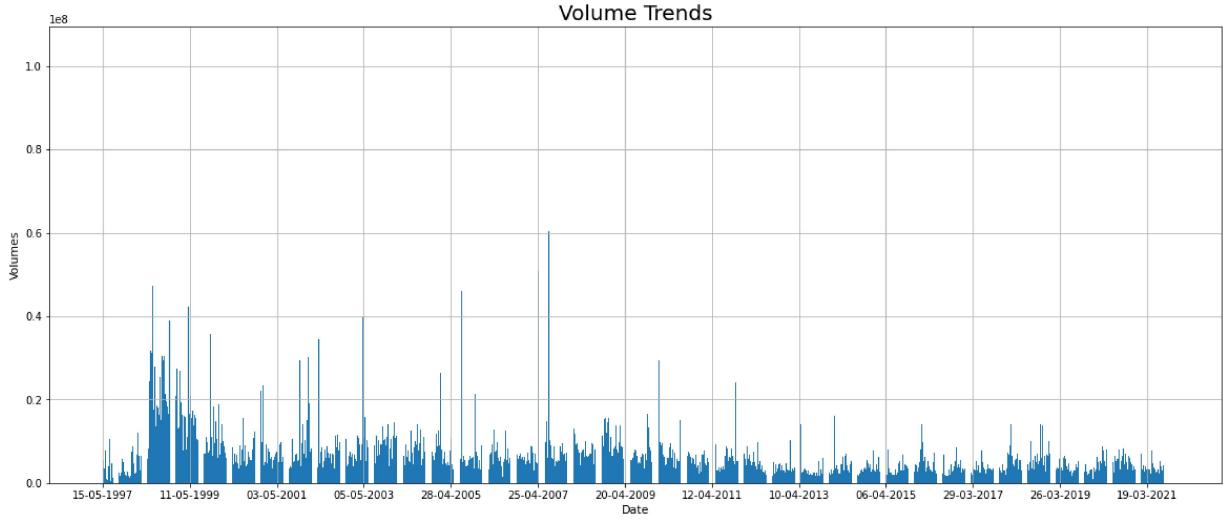
In [13]:

```
fig, ax = plt.subplots(figsize=(20,8))
ax.plot(amazon_data['Date'], amazon_data['Close'], color='Red')
ax.xaxis.set_major_locator(plt.MaxNLocator(15))
ax.set_xlabel('Date', fontsize='11')
ax.set_ylabel('Price in USD', fontsize='11')
plt.title('Amazon Stock Prices')
plt.grid()
plt.show()
```



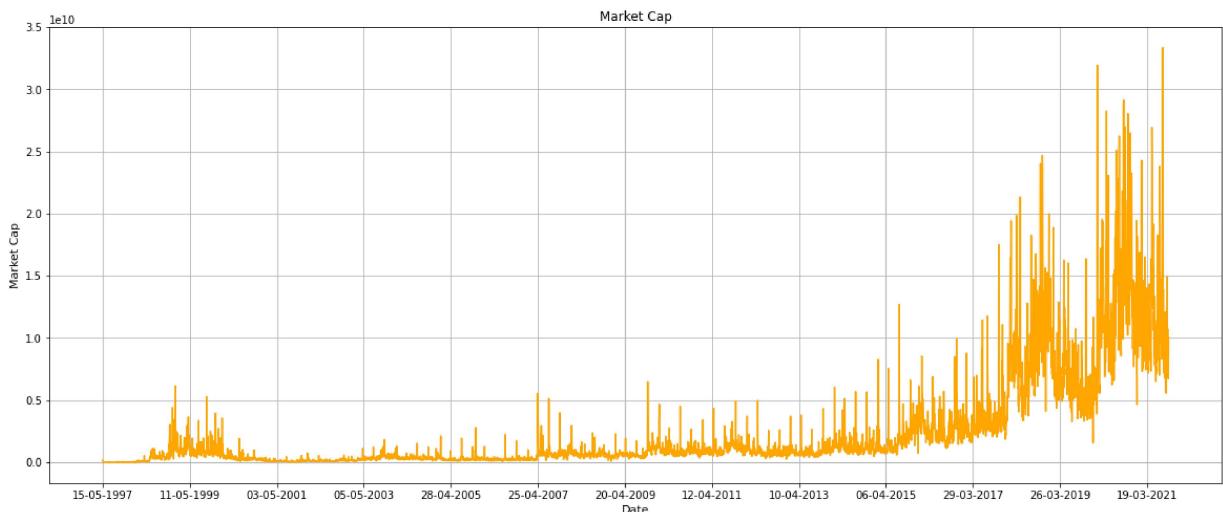
In [14]:

```
fig, ax = plt.subplots(figsize=(20,8))
ax.bar(amazon_data['Date'], amazon_data['Volume'])
ax.xaxis.set_major_locator(plt.MaxNLocator(15))
ax.set_xlabel('Date', fontsize='11')
ax.set_ylabel('Volumes', fontsize='11')
plt.title('Volume Trends', fontsize='20')
plt.grid()
plt.show()
```



```
In [15]: amazon_data['Market Cap'] = amazon_data['Open']*amazon_data['Volume']
```

```
In [16]: fig, ax = plt.subplots(figsize=(20,8))
ax.plot(amazon_data['Date'], amazon_data['Market Cap'], color='orange')
ax.xaxis.set_major_locator(plt.MaxNLocator(15))
ax.set_xlabel('Date', fontsize='11')
ax.set_ylabel('Market Cap', fontsize='11')
plt.title('Market Cap')
plt.grid()
plt.show()
```



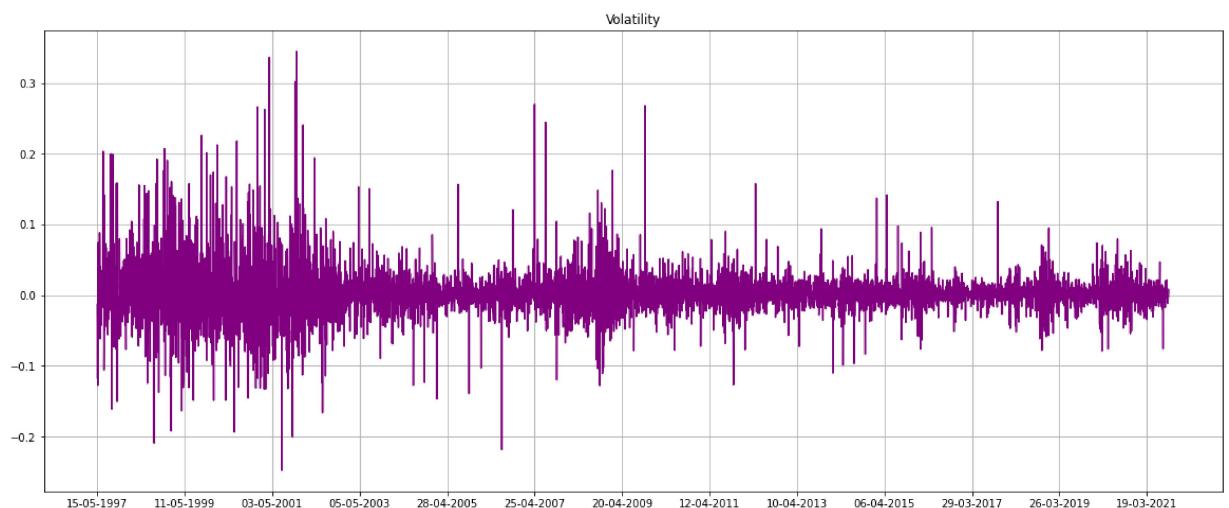
```
In [17]: amazon_data.iloc[amazon_data['Market Cap'].argmax()]
```

```
Out[17]: Date ..... 30-07-2021
          Open ..... 3347.949951
          High ..... 3368.139893
          Low ..... 3306.97998
          Close ..... 3327.590088
          Volume ..... 9957100
          Market Cap ... 33335872457.1021
          Name: 6092, dtype: object
```

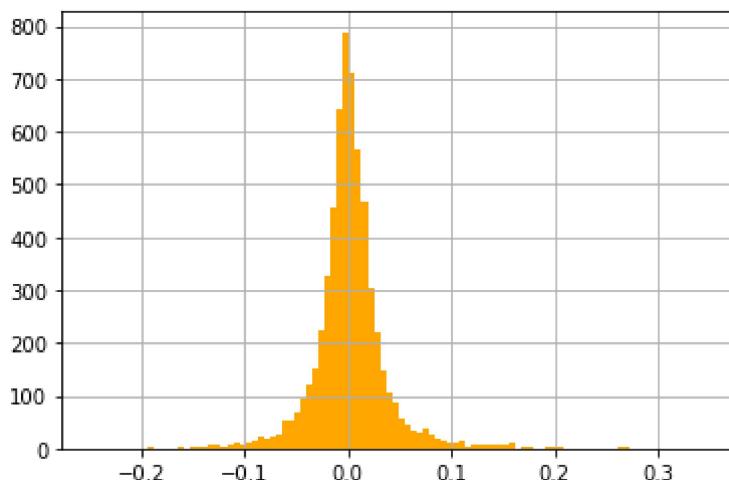
```
In [18]: amazon_data['vol'] = (amazon_data['Close']/amazon_data['Close'].shift(1)) - 1
```

```
In [19]: fig, ax = plt.subplots(figsize=(20,8))
```

```
ax.plot(amazon_data[ 'Date' ], amazon_data[ 'vol' ], color='purple')
ax.xaxis.set_major_locator(plt.MaxNLocator(15))
plt.title('Volatility')
plt.grid()
plt.show()
```



In [20]: `amazon_data[ 'vol' ].hist(bins=100, color='orange');`



In [21]: `amazon_data[ 'Cumulative Return' ] = (1 + amazon_data[ 'vol' ]).cumprod()`

In [22]: `fig, ax = plt.subplots(figsize=(20,8))
ax.plot(amazon_data[ 'Date' ], amazon_data[ 'Cumulative Return' ], color='green')
ax.xaxis.set_major_locator(plt.MaxNLocator(15))
ax.set_xlabel('Date', fontsize='11')
ax.set_ylabel('Cumulative Return', fontsize='11')
plt.title('Cumulative Return')
plt.grid()
plt.show()`

## Amazon Stock Price



In [ ]:

In [ ]:

In [ ]: