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

In [2]:
```
import numpy as np
```

In [5]:
```
dataset=pd.read_csv("C:/Users/infosoft/Downloads/Amazon/Amazon.csv")
```

In [6]:
```
dataset
```

Out[6]:

	Date	Open	High	Low	Close	Adj Close	Volume	Da
0	5/15/1997	2.437500	2.500000	1.927083	1.958333	1.958333	72156000	
1	5/16/1997	1.968750	1.979167	1.708333	1.729167	1.729167	14700000	
2	5/19/1997	1.760417	1.770833	1.625000	1.708333	1.708333	6106800	
3	5/20/1997	1.729167	1.750000	1.635417	1.635417	1.635417	5467200	2
4	5/21/1997	1.635417	1.645833	1.375000	1.427083	1.427083	18853200	2
...	...	...	...	...	...	...	...	
6150	10/21/2021	3414.250000	3440.280029	3403.000000	3435.010010	3435.010010	1881400	2
6151	10/22/2021	3421.000000	3429.840088	3331.300049	3335.550049	3335.550049	3133800	2
6152	10/25/2021	3335.000000	3347.800049	3297.699951	3320.370117	3320.370117	2226000	2
6153	10/26/2021	3349.510010	3416.120117	3343.979980	3376.070068	3376.070068	2693700	2
6154	10/27/2021	3388.000000	3412.000000	3371.453369	3396.189941	3396.189941	1080291	2

6155 rows × 10 columns

In [46]:
```
sns.set_style("darkgrid")
```

In [14]:
```
#Coverting the date column to a format that pandas recognizes as date
dataset["Date"] = pd.to_datetime(dataset["Date"])
```

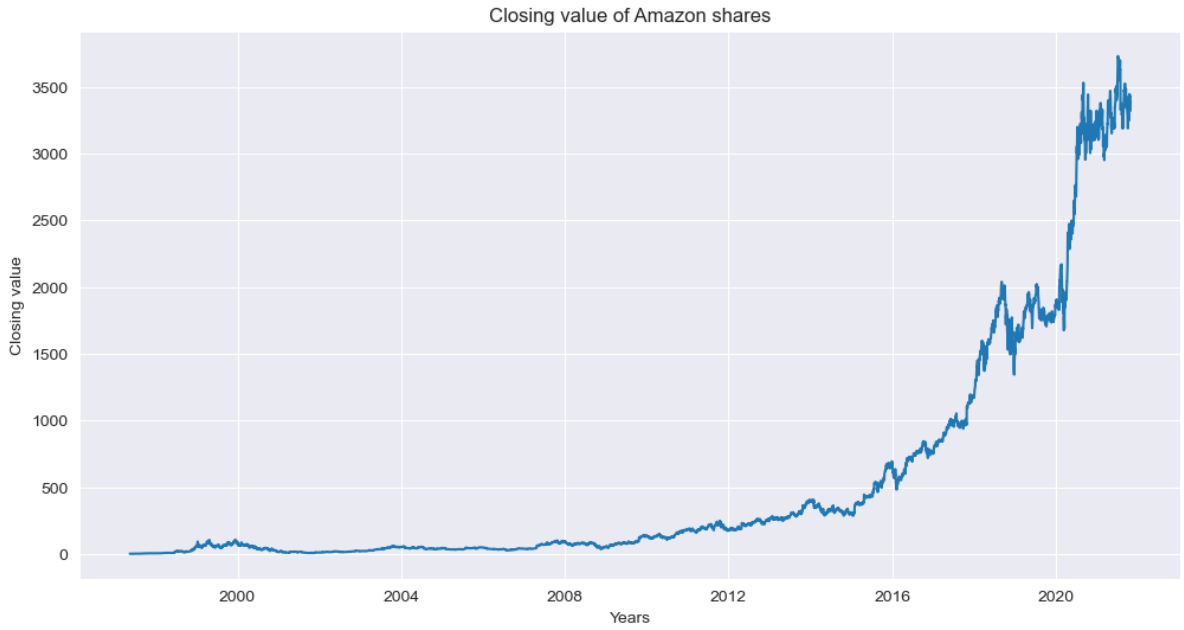
In [15]:
```
dataset
```

Out[15]:

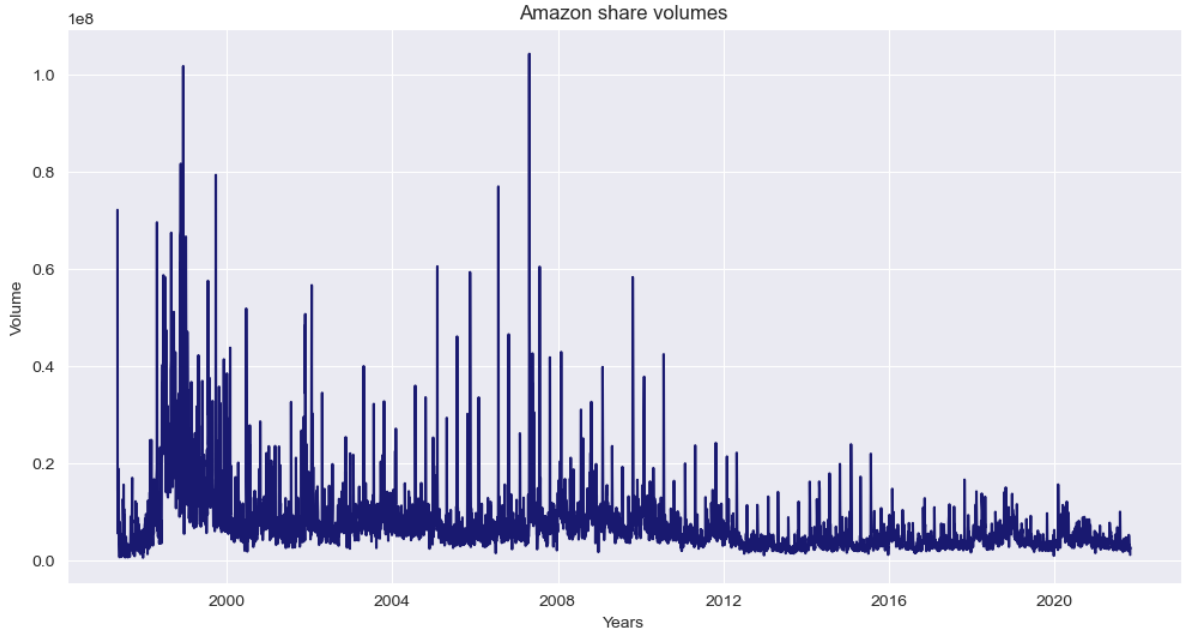
	Date	Open	High	Low	Close	Adj Close	Volume	Day	M
0	1997-05-15	2.437500	2.500000	1.927083	1.958333	1.958333	72156000	15	
1	1997-05-16	1.968750	1.979167	1.708333	1.729167	1.729167	14700000	16	
2	1997-05-19	1.760417	1.770833	1.625000	1.708333	1.708333	6106800	19	
3	1997-05-20	1.729167	1.750000	1.635417	1.635417	1.635417	5467200	20	
4	1997-05-21	1.635417	1.645833	1.375000	1.427083	1.427083	18853200	21	
...	...	...	...	...	...	...	...	...	
6150	2021-10-21	3414.250000	3440.280029	3403.000000	3435.010010	3435.010010	1881400	21	
6151	2021-10-22	3421.000000	3429.840088	3331.300049	3335.550049	3335.550049	3133800	22	
6152	2021-10-25	3335.000000	3347.800049	3297.699951	3320.370117	3320.370117	2226000	25	
6153	2021-10-26	3349.510010	3416.120117	3343.979980	3376.070068	3376.070068	2693700	26	
6154	2021-10-27	3388.000000	3412.000000	3371.453369	3396.189941	3396.189941	1080291	27	

6155 rows × 10 columns

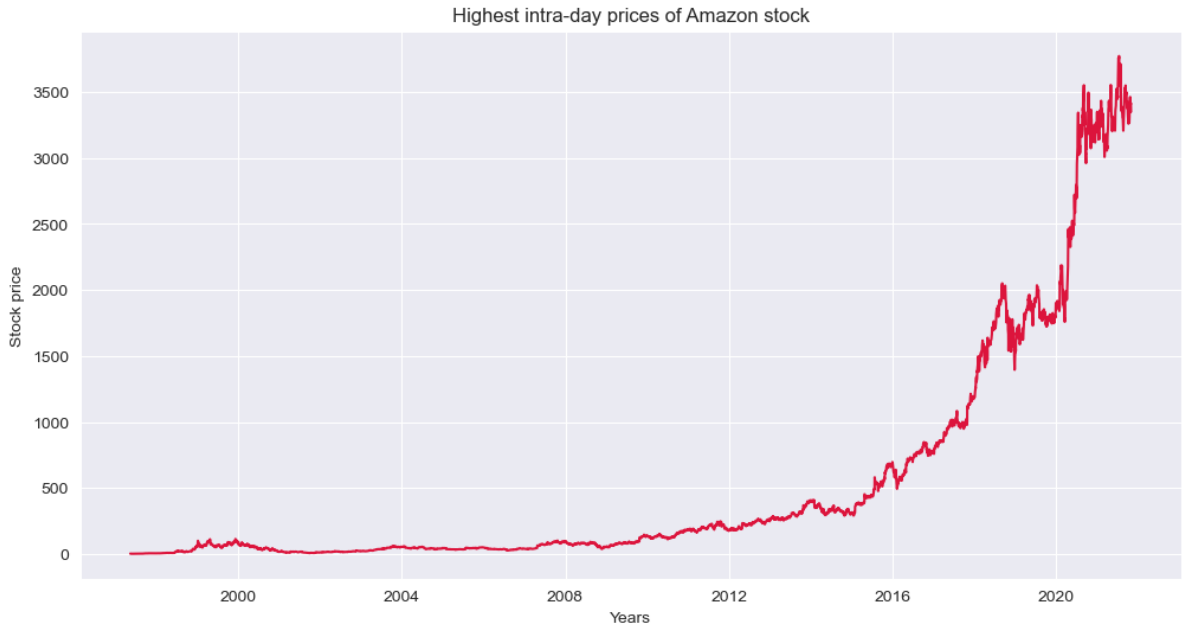
In [130]:
```
#Plotting time series graph of whole dataset of closing values
plt.figure(figsize=(12, 6))
plt.title("Closing value of Amazon shares")
plt.xlabel("Years")
plt.ylabel("Closing value")
plt.plot(dataset.Date, dataset.Close);
```



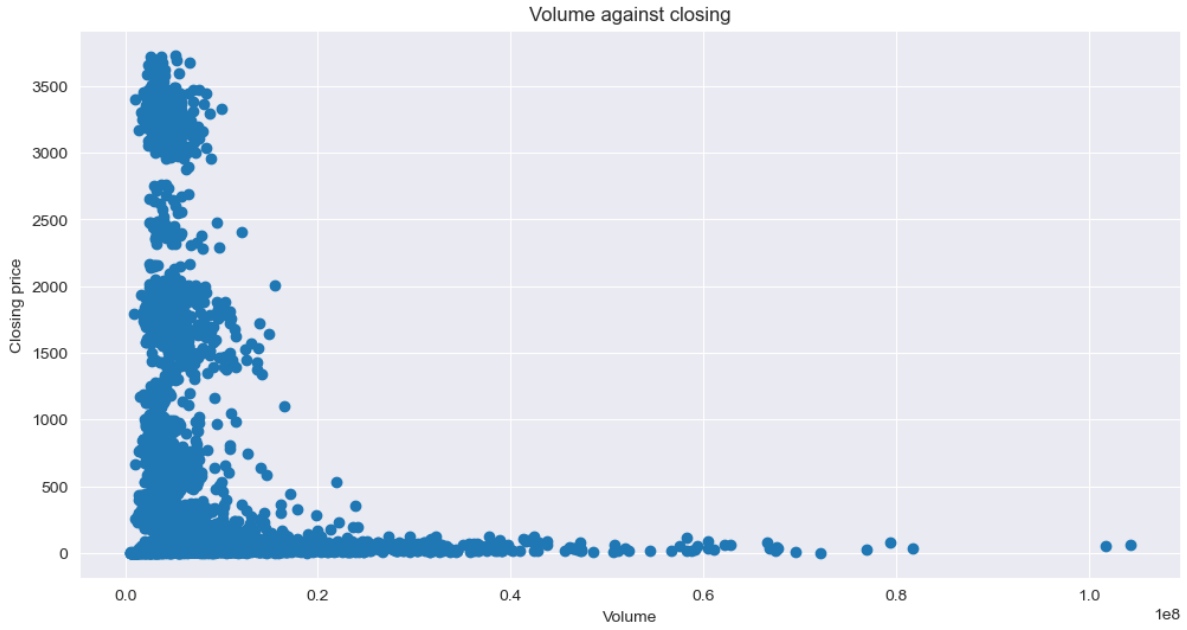
In [131]:
```
#Plotting time series graph of volumes
plt.figure(figsize=(12, 6))
plt.title("Amazon share volumes")
plt.xlabel("Years")
plt.ylabel("Volume")
plt.plot(dataset.Date, dataset.Volume, 'midnightblue');
```



In [132]:
```
#Plotting graph of intra-day high values of Amazon stock
plt.figure(figsize=(12, 6))
plt.title("Highest intra-day prices of Amazon stock")
plt.xlabel("Years")
plt.ylabel("Stock price")
plt.plot(dataset.Date, dataset.High, 'crimson');
```



In [100]:
```
#plotting volume against closing value
plt.figure(figsize=(12, 6))
plt.title("Volume against closing")
plt.xlabel("Volume")
plt.ylabel("Closing price")
plt.scatter(dataset.Volume, dataset.Close);
```



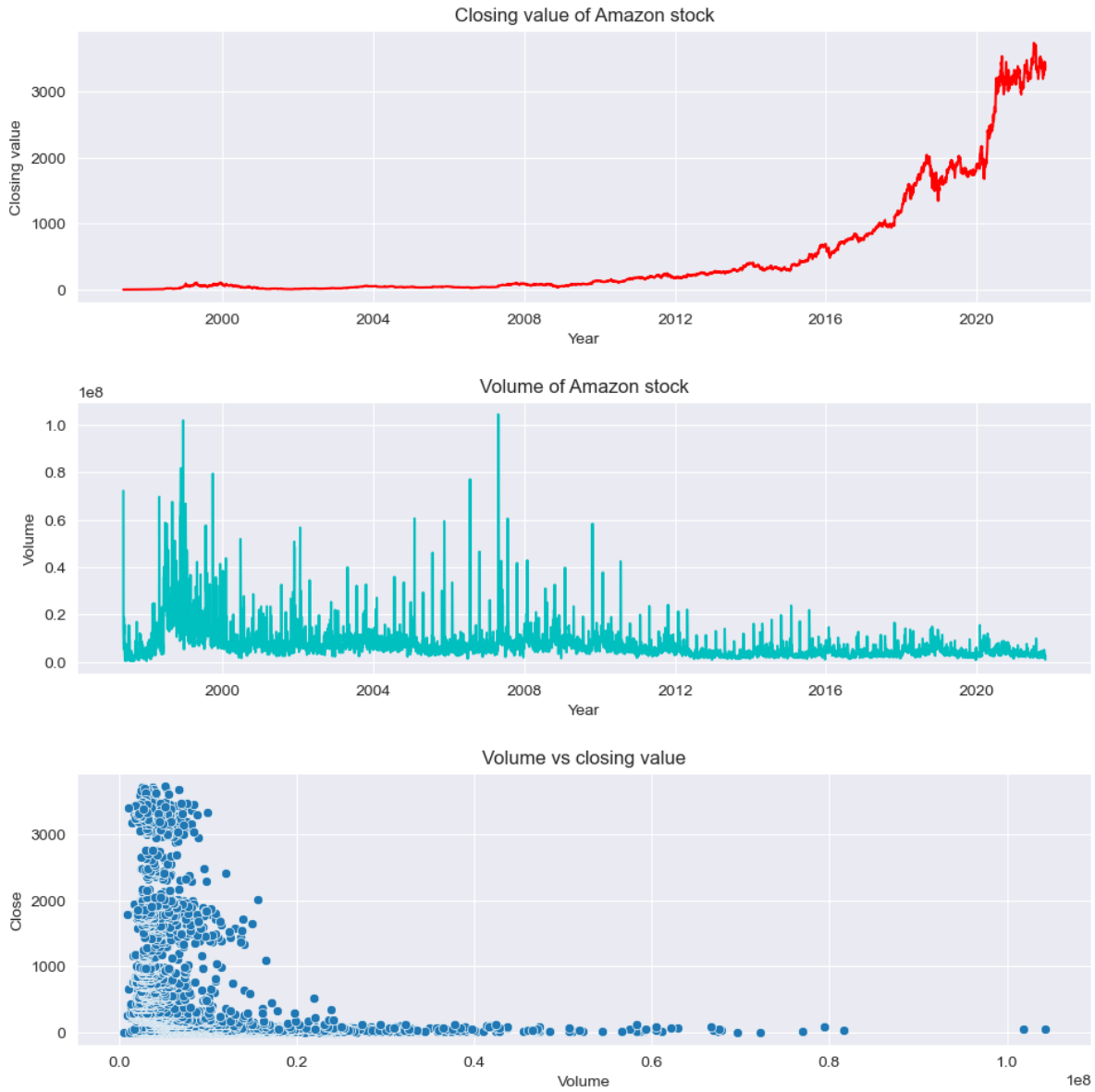
In [51]:
```
#Plotting time series of volume and closing value alongside volume-closing valu
fig, axes = plt.subplots(3, 1, figsize=(10,10))

axes[0].plot(dataset.Date, dataset.Close, 'r')
axes[0].set_xlabel('Year')
axes[0].set_ylabel('Closing value')
axes[0].set_title('Closing value of Amazon stock')

axes[1].plot(dataset.Date, dataset.Volume, 'c')
axes[1].set_xlabel('Year')
axes[1].set_ylabel('Volume')
axes[1].set_title('Volume of Amazon stock')

axes[2].set_title("Volume vs closing value")
sns.scatterplot(x=dataset.Volume, y=dataset.Close)

plt.tight_layout(pad=2);
```

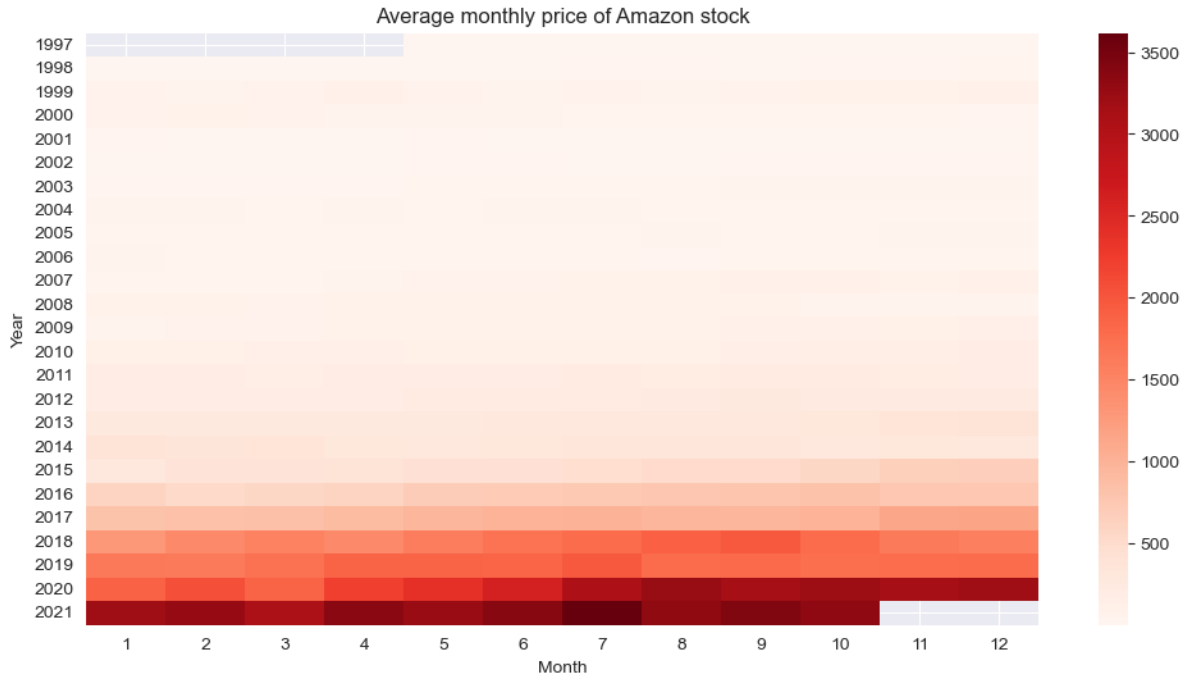


In [85]:
```
#Finding average closing value of Amazon's stock price in each month
avgtable=np.round(pd.pivot_table(dataset,values="Close", index="Year", columns
avgtable
```

Out[85]:

Month	1	2	3	4	5	6	7	8	9	10
Year										
1997	NaN	NaN	NaN	NaN	1.59	1.54	2.21	2.23	3.49	4.27
1998	4.80	5.30	6.73	7.63	7.47	11.55	19.79	19.95	15.01	17.67
1999	67.08	54.63	67.27	91.03	66.83	55.80	59.87	53.02	65.14	79.61
2000	68.05	72.46	66.70	56.38	53.69	45.47	36.78	36.11	41.72	30.79
2001	17.81	13.50	10.91	13.32	15.81	14.16	14.78	10.42	7.44	7.65
2002	12.08	13.07	15.32	14.55	18.11	17.39	14.78	14.55	16.29	18.38
2003	21.51	21.47	25.12	26.25	32.17	35.26	38.85	42.19	47.44	56.06
2004	53.95	45.22	42.36	46.53	43.26	50.86	46.00	37.93	40.55	38.27
2005	42.36	36.20	34.49	33.70	34.66	35.09	37.61	44.22	43.04	43.64
2006	45.22	38.82	36.38	36.32	34.13	35.27	33.26	27.80	31.54	33.93
2007	37.56	39.50	38.66	46.63	64.56	70.36	74.28	76.82	88.46	91.66
2008	81.92	72.28	68.58	77.08	77.34	79.82	72.43	82.30	75.94	56.41
2009	52.32	63.43	68.52	78.61	77.69	83.54	83.61	84.08	86.74	101.24
2010	127.42	118.02	130.90	141.07	127.61	121.67	117.52	126.37	147.66	160.74
2011	182.35	181.12	168.80	184.58	198.17	191.67	215.20	199.46	223.24	226.84
2012	185.03	183.24	189.53	195.29	220.88	219.27	224.06	238.99	255.74	244.29
2013	268.38	263.70	265.76	263.07	262.73	274.10	298.98	291.15	304.73	325.97
2014	394.86	354.34	362.63	321.64	302.95	324.57	339.92	327.33	330.31	308.41
2015	302.75	375.75	375.32	394.21	426.93	432.60	478.71	518.46	520.96	566.74
2016	601.06	530.62	572.37	613.59	697.47	716.39	741.47	764.84	788.97	824.44
2017	807.51	835.75	854.24	903.39	961.72	990.44	1008.48	971.44	968.99	1000.72
2018	1309.01	1442.36	1540.37	1468.22	1594.90	1698.82	1784.65	1897.85	1966.08	1782.06
2019	1640.03	1626.94	1722.49	1866.20	1869.38	1852.62	1965.88	1793.60	1799.12	1752.33
2020	1884.24	2066.18	1872.31	2228.71	2394.18	2613.55	3053.85	3249.25	3160.75	3230.60
2021	3200.05	3263.87	3068.27	3352.17	3246.26	3367.73	3616.01	3312.92	3427.18	3326.48

In [101]:
```
#Making a heatmap of average closing value of Amazon's stock price in each mont
plt.figure(figsize=(12, 6))
plt.title("Average monthly price of Amazon stock")
sns.heatmap(avgtable, cmap="Reds");
```

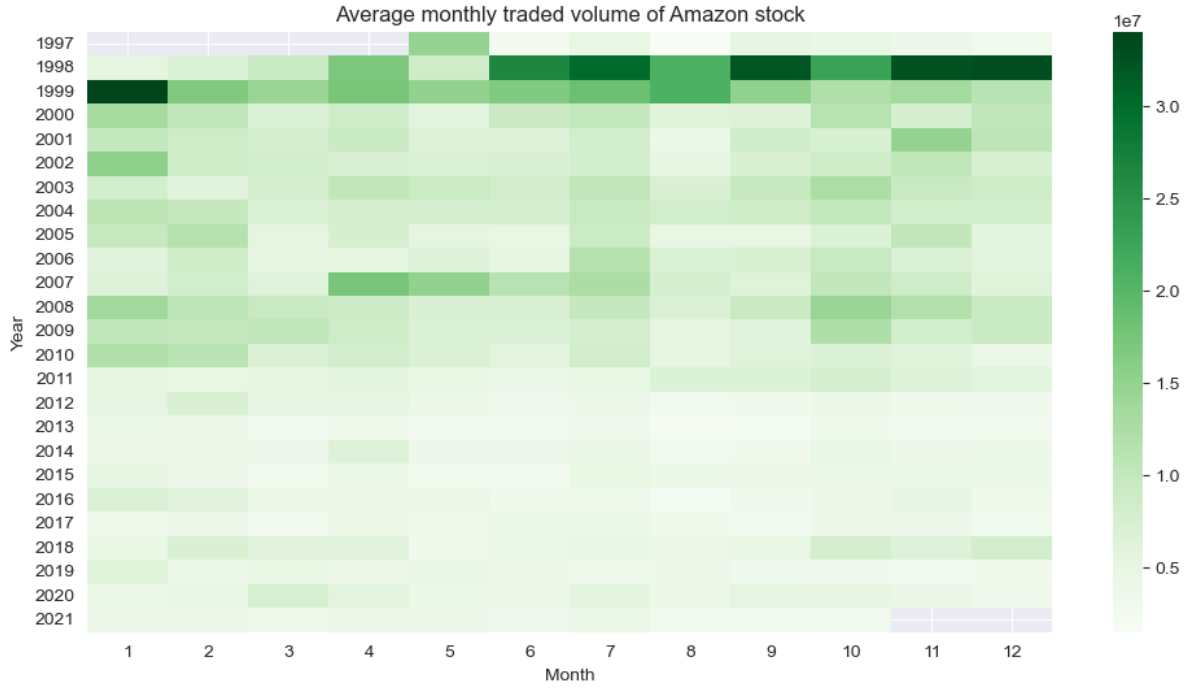


In [95]:
```
#Finding average traded volume of Amazon's stock price in each month
avgvoltable=np.round(pd.pivot_table(dataset,values="Volume", index="Year", col
avgvoltable
```

Out[95]:

Month	1	2	3	4	5	6	7	
Year								
1997	NaN	NaN	NaN	NaN	14939673.0	2531600.0	4937018.0	15
1998	5604000.0	6874800.0	9400855.0	16898514.0	8743500.0	26539936.0	30179400.0	209
1999	34018358.0	16744526.0	14391739.0	17299305.0	15188390.0	16579427.0	18507162.0	208
2000	13110450.0	10372885.0	6808026.0	8526974.0	5940391.0	9233677.0	9926230.0	62
2001	10023181.0	8809163.0	7619814.0	9489020.0	6576668.0	6509329.0	8194224.0	43
2002	15368533.0	8630995.0	8240890.0	7225391.0	7038205.0	7130890.0	8026918.0	52
2003	8057010.0	6102205.0	7813981.0	10459252.0	9087324.0	8165276.0	10419977.0	71
2004	10896735.0	9776195.0	6729152.0	7744600.0	7509825.0	7678629.0	9639857.0	81
2005	9746760.0	11508163.0	5434845.0	7737681.0	5568943.0	4828800.0	9130055.0	49
2006	6170515.0	8712879.0	5312091.0	5618874.0	6677091.0	5232609.0	11433170.0	69
2007	6521765.0	8314495.0	6461505.0	17314350.0	15011018.0	11370890.0	12610848.0	76
2008	13730257.0	10560890.0	9634280.0	8993155.0	7265390.0	7281224.0	9809695.0	70
2009	10203615.0	9830537.0	10348741.0	8811048.0	6880170.0	6912973.0	7975518.0	53
2010	12150579.0	11060305.0	6871417.0	8239290.0	7045495.0	6062659.0	8348867.0	51
2011	5680565.0	5040863.0	5173004.0	5837470.0	5060690.0	4343805.0	4640425.0	67
2012	5526845.0	7329020.0	5135232.0	5526770.0	4217014.0	3130743.0	3682867.0	26
2013	4010781.0	3613774.0	2925285.0	3496191.0	2684114.0	2928790.0	3069859.0	19
2014	3857710.0	4368895.0	3583500.0	6584700.0	3741867.0	3646671.0	4519364.0	29
2015	5152855.0	3728747.0	2522855.0	4019276.0	2599150.0	2499432.0	4639545.0	39
2016	6852679.0	6207240.0	4273159.0	3736390.0	4314976.0	3388223.0	3431775.0	21
2017	3530700.0	3776226.0	2639596.0	3870511.0	3463727.0	4369791.0	3940620.0	33
2018	4589105.0	7251789.0	6209529.0	6186648.0	3255250.0	4092443.0	4643862.0	41
2019	6381033.0	4259837.0	4801533.0	3868533.0	4464291.0	3737325.0	3324945.0	36
2020	4033252.0	4868953.0	7445868.0	5933800.0	4129220.0	3991741.0	5795545.0	39
2021	3764679.0	3795105.0	3399478.0	3659029.0	3759195.0	3045955.0	3986705.0	28

In [102]:
```
#Making a heatmap of average traded volume of Amazon's stock price in each mont
plt.figure(figsize=(12, 6))
plt.title("Average monthly traded volume of Amazon stock")
sns.heatmap(avgvoltable, cmap="Greens");
```

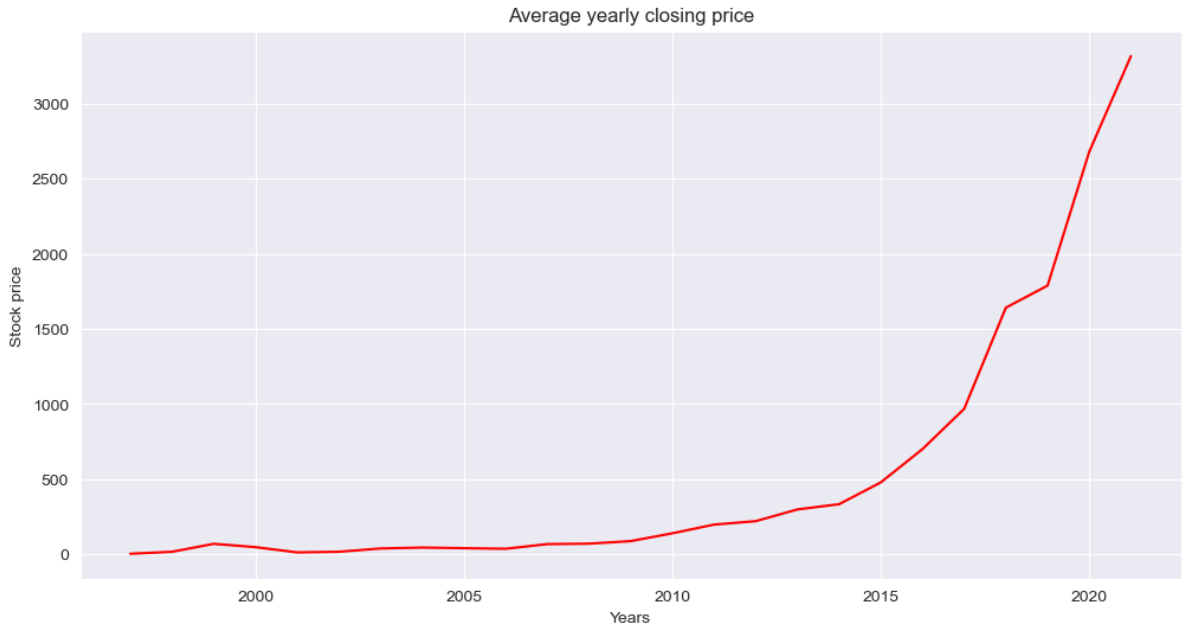


In [117]:
```
#Finding average yearly closing price of Amazon's stock
Avgyearlyclosing=np.round(pd.pivot_table(dataset,values="Close", index="Year",
Avgyearlyclosing
```

Out[117]:

Year	Close
1997	3.0
1998	16.0
1999	69.0
2000	47.0
2001	12.0
2002	16.0
2003	38.0
2004	44.0
2005	40.0
2006	36.0
2007	67.0
2008	70.0
2009	87.0
2010	139.0
2011	197.0
2012	220.0
2013	298.0
2014	333.0
2015	478.0
2016	700.0
2017	968.0
2018	1642.0
2019	1789.0
2020	2681.0
2021	3318.0

In [123]:
```
#Plotting average yearly closing price of Amazon's stock
plt.figure(figsize=(12, 6))
plt.title("Average yearly closing price")
plt.xlabel("Years")
plt.ylabel("Stock price")
plt.plot(Avgyearlyclosing.index,Avgyearlyclosing.Close, 'r');
```

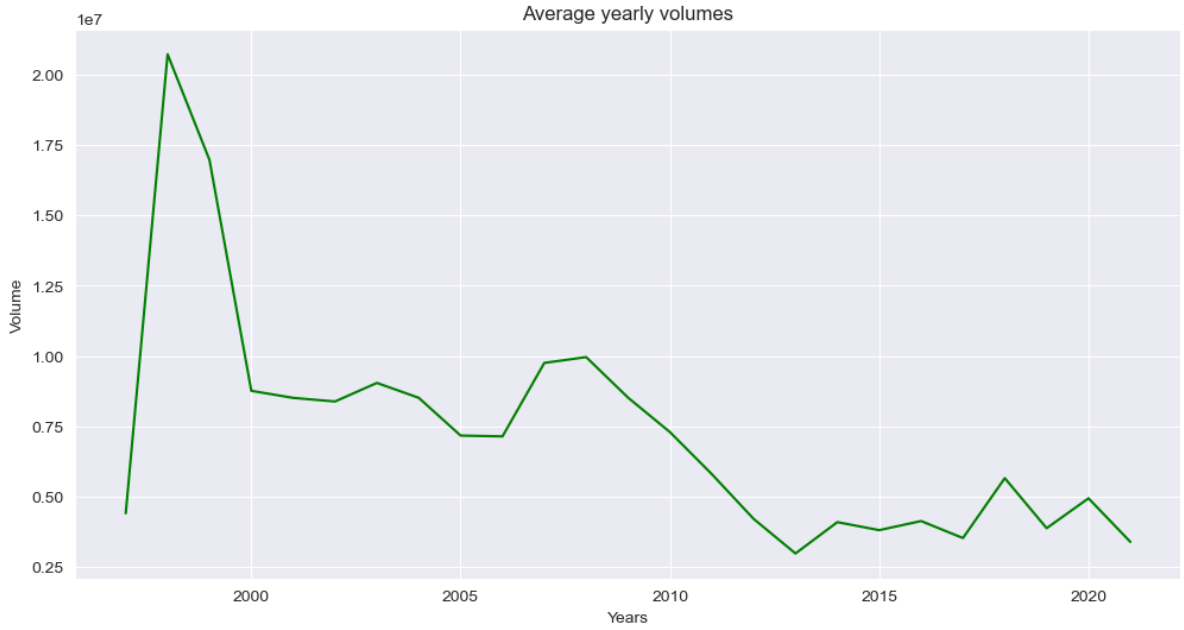


In [121]:
```
#Finding average yearly volume of the stock
Avgyearlyvol=np.round(pd.pivot_table(dataset,values="Volume", index="Year",agg
Avgyearlyvol
```

Out[121]:

Year	Volume
1997	4399065.0
1998	20738014.0
1999	16978239.0
2000	8757967.0
2001	8506342.0
2002	8378608.0
2003	9039111.0
2004	8508856.0
2005	7164625.0
2006	7135865.0
2007	9751969.0
2008	9957347.0
2009	8515255.0
2010	7289629.0
2011	5793566.0
2012	4199616.0
2013	2967880.0
2014	4083598.0
2015	3798024.0
2016	4122049.0
2017	3516755.0
2018	5648994.0
2019	3867659.0
2020	4930991.0
2021	3380463.0

In [124]:
```
#Plotting average yearly volume of the stock
plt.figure(figsize=(12, 6))
plt.title("Average yearly volumes")
plt.xlabel("Years")
plt.ylabel("Volume")
plt.plot(Avgyearlyvol.index,Avgyearlyvol.Volume, 'g');
```



In [ ]: