

```
In [1]: import numpy as np
import pandas as pd

In [8]: df_btc = pd.read_csv('D:/G-PYTHON/Python 42/Data science/Data Science Projects/Python For Finance Cryptocurren
df_eth = pd.read_csv('D:/G-PYTHON/Python 42/Data science/Data Science Projects/Python For Finance Cryptocurren
df_doge = pd.read_csv('D:/G-PYTHON/Python 42/Data science/Data Science Projects/Python For Finance Cryptocurren

In [9]: df_btc.head()

Out[9]:
```

	SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap
0	1	Bitcoin	BTC	2013-04-29 23:59:59	147.488007	134.000000	134.444000	144.539993	0.0	1.603769e+09
1	2	Bitcoin	BTC	2013-04-30 23:59:59	146.929993	134.050003	144.000000	139.000000	0.0	1.542813e+09
2	3	Bitcoin	BTC	2013-05-01 23:59:59	139.889999	107.720001	139.000000	116.989998	0.0	1.298955e+09
3	4	Bitcoin	BTC	2013-05-02 23:59:59	125.599998	92.281898	116.379997	105.209999	0.0	1.168517e+09
4	5	Bitcoin	BTC	2013-05-03 23:59:59	108.127998	79.099998	106.250000	97.750000	0.0	1.085995e+09

```
In [10]: df_eth.head()

Out[10]:
```

	SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap
0	1	Ethereum	ETH	2015-08-08 23:59:59	2.798810	0.714725	2.793760	0.753325	674188.0	4.548689e+07
1	2	Ethereum	ETH	2015-08-09 23:59:59	0.879810	0.629191	0.706136	0.701897	532170.0	4.239957e+07
2	3	Ethereum	ETH	2015-08-10 23:59:59	0.729854	0.636546	0.713989	0.708448	405283.0	4.281836e+07
3	4	Ethereum	ETH	2015-08-11 23:59:59	1.131410	0.663235	0.708087	1.067860	1463100.0	6.456929e+07
4	5	Ethereum	ETH	2015-08-12 23:59:59	1.289940	0.883608	1.058750	1.217440	2150620.0	7.364501e+07

```
In [11]: df_doge.head()

Out[11]:
```

	SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap
0	1	Dogecoin	DOGE	2013-12-16 23:59:59	0.000866	0.000150	0.000299	0.000205	0.0	1.509085e+06
1	2	Dogecoin	DOGE	2013-12-17 23:59:59	0.000289	0.000116	0.000207	0.000269	0.0	2.169688e+06
2	3	Dogecoin	DOGE	2013-12-18 23:59:59	0.000362	0.000205	0.000267	0.000362	0.0	3.188943e+06
3	4	Dogecoin	DOGE	2013-12-19 23:59:59	0.001520	0.000328	0.000395	0.001162	0.0	1.115034e+07
4	5	Dogecoin	DOGE	2013-12-20 23:59:59	0.001143	0.000662	0.001143	0.000704	0.0	7.284337e+06

```
In [14]: df = pd.DataFrame({'BTC': df_btc['Close'],'ETH': df_eth['Close'],'DOGE': df_doge['Close']})
df.head()

Out[14]:
```

	BTC	ETH	DOGE
0	144.539993	0.753325	0.000205
1	139.000000	0.701897	0.000269
2	116.989998	0.708448	0.000362
3	105.209999	1.067860	0.001162
4	97.750000	1.217440	0.000704

```
In [15]: df

Out[15]:
```

	BTC	ETH	DOGE
0	144.539993	0.753325	0.000205
1	139.000000	0.701897	0.000269
2	116.989998	0.708448	0.000362
3	105.209999	1.067860	0.001162
4	97.750000	1.217440	0.000704
...
2986	33897.048590	NaN	NaN
2987	34668.548402	NaN	NaN
2988	35287.779766	NaN	NaN
2989	33746.002456	NaN	NaN
2990	34235.193451	NaN	NaN

2991 rows x 3 columns

```
In [16]: df.tail()

Out[16]:
```

	BTC	ETH	DOGE
2986	33897.048590	NaN	NaN
2987	34668.548402	NaN	NaN
2988	35287.779766	NaN	NaN
2989	33746.002456	NaN	NaN
2990	34235.193451	NaN	NaN

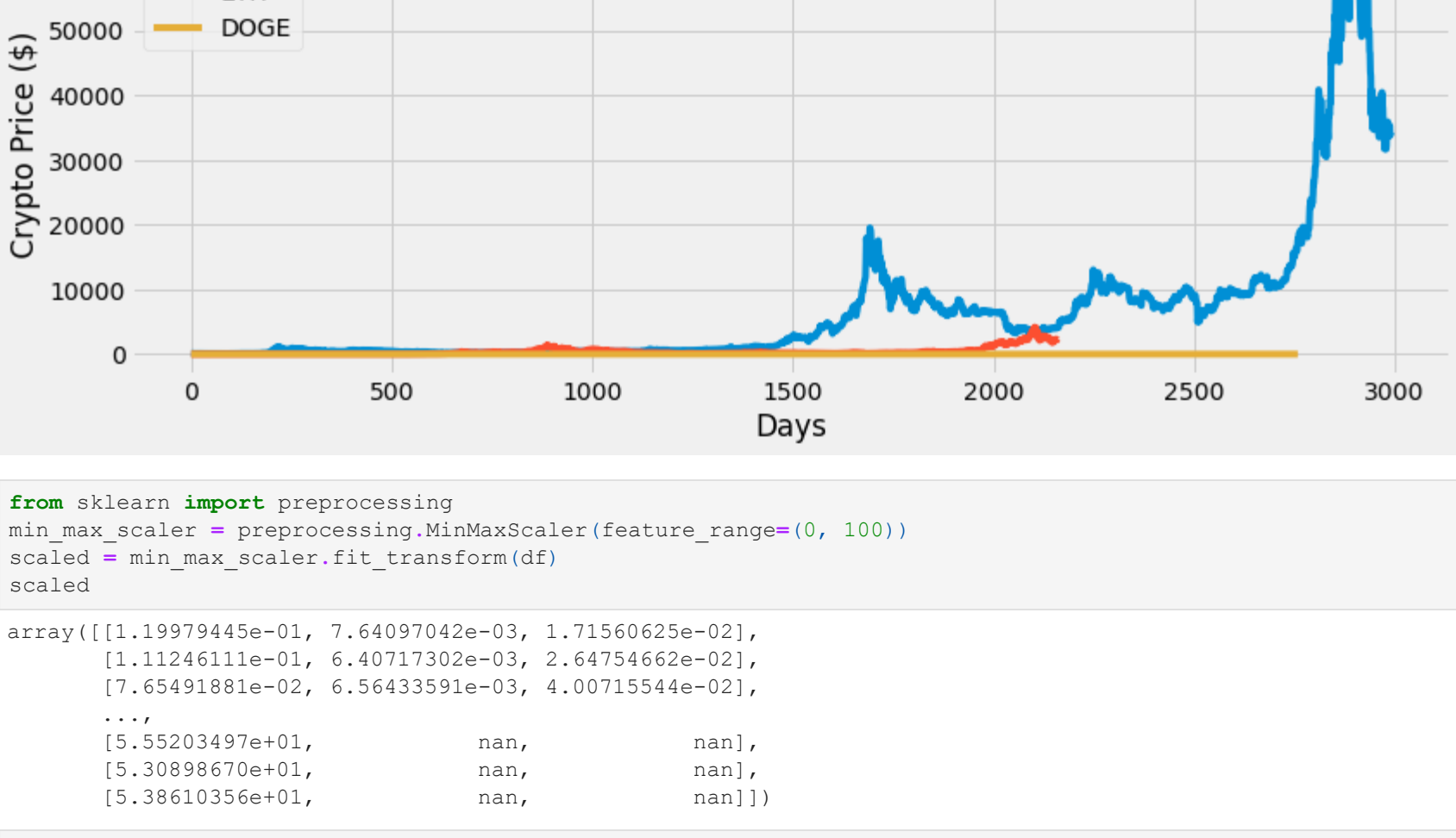
```
In [17]: df.describe()

Out[17]:
```

	BTC	ETH	DOGE
count	2991.000000	2160.000000	2760.000000
mean	6711.290443	383.910691	0.013763
std	11298.141921	601.078766	0.062559
min	68.431000	0.434829	0.000087
25%	430.569489	13.819200	0.000226
50%	2286.409912	198.643691	0.001844
75%	8576.238715	386.435272	0.002899
max	63503.457930	4168.701049	0.684777

```
In [18]: import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
my_crypto = df
plt.figure(figsize=(12,2,4.5))
for c in my_crypto.columns.values:
    plt.plot(my_crypto[c], label=c)
plt.title('Cryptocurrency Graph')
plt.xlabel('Days')
plt.ylabel('Crypto Price ($)')
plt.legend(my_crypto.columns.values, loc='upper left')
plt.show()

Cryptocurrency Graph
```



```
In [20]: from sklearn import preprocessing
min_max_scaler = preprocessing.MinMaxScaler(feature_range=(0, 100))
scaled = min_max_scaler.fit_transform(df)
scaled

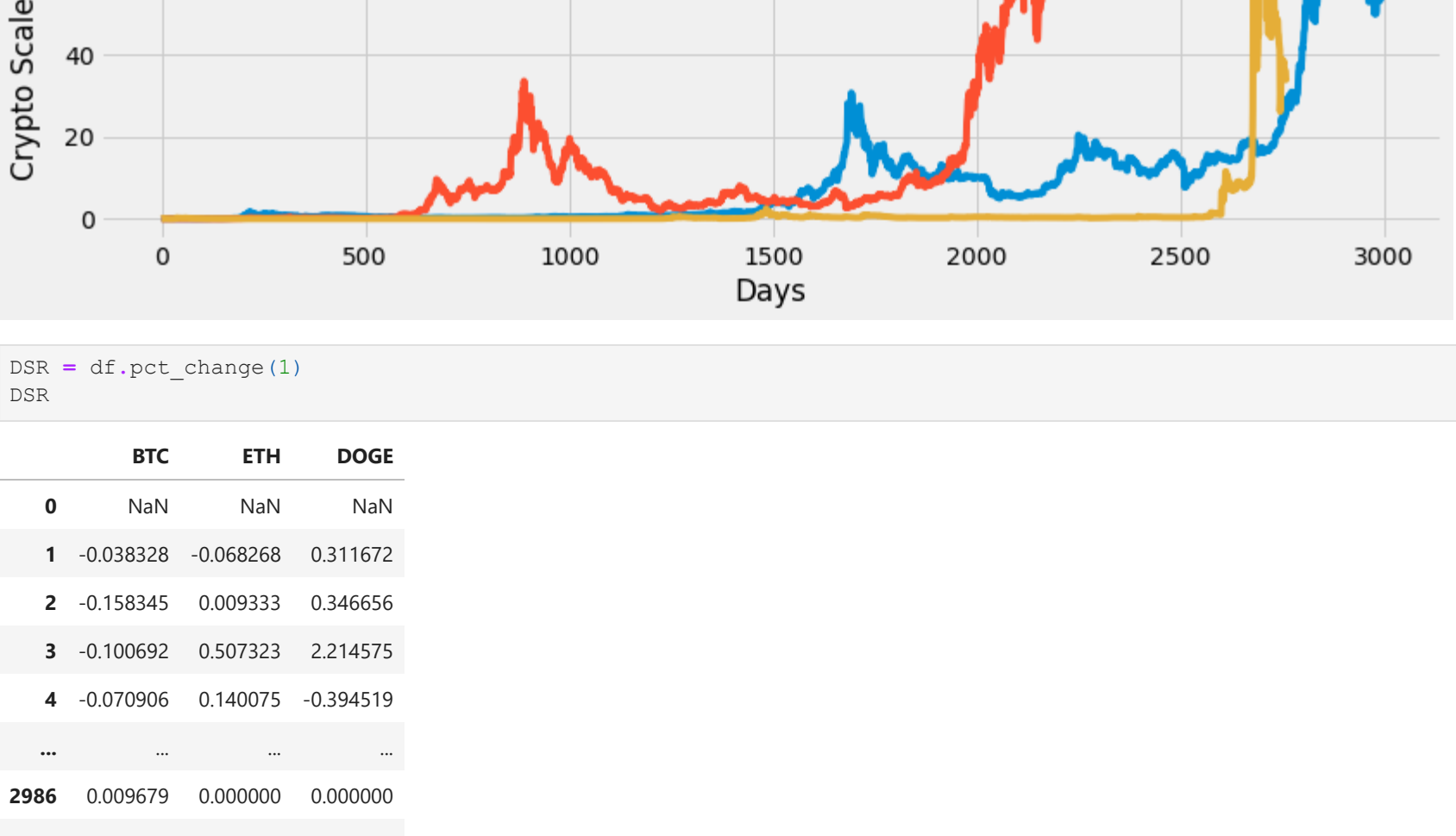
Out[20]:
```

```
array([[1.19979445e-01, 7.64097042e-03, 1.71560625e-02],
       [1.11246111e-01, 6.40717302e-03, 2.64754662e-02],
       [7.65491881e-02, 6.56433591e-03, 4.00715544e-02],
       ...,
       [5.55203497e+01, nan, nan],
       [5.30898670e+01, nan, nan],
       [5.38610356e+01, nan, nan]])

In [21]: df_scale = pd.DataFrame(scaled, columns = df.columns)

In [22]: my_crypto = df_scale
plt.figure(figsize=(12,4,4.5))
for c in my_crypto.columns.values:
    plt.plot(my_crypto[c], label=c)
plt.title('Cryptocurrency Scaled Graph')
plt.xlabel('Days')
plt.ylabel('Crypto Scaled Price ($)')
plt.legend(my_crypto.columns.values, loc='upper left')
plt.show()

Cryptocurrency Scaled Graph
```



```
In [23]: DSR = df.pct_change(1)
DSR

Out[23]:
```

	BTC	ETH	DOGE
0	NaN	NaN	NaN
1	-0.038328	-0.068268	0.311672
2	-0.158345	0.009333	0.346656
3	-0.100692	0.507323	2.214575
4	-0.070906	0.140075	-0.394519
...
2986	0.009679	0.000000	0.000000
2987	0.022760	0.000000	0.000000
2988	0.017861	0.000000	0.000000
2989	-0.043692	0.000000	0.000000
2990	0.014496	0.000000	0.000000

2991 rows x 3 columns

```
In [24]: DSR.head()

Out[24]:
```

	BTC	ETH	DOGE
0	NaN	NaN	NaN
1	-0.038328	-0.068268	0.311672
2	-0.158345	0.009333	0.346656
3	-0.100692	0.507323	2.214575
4	-0.070906	0.140075	-0.394519

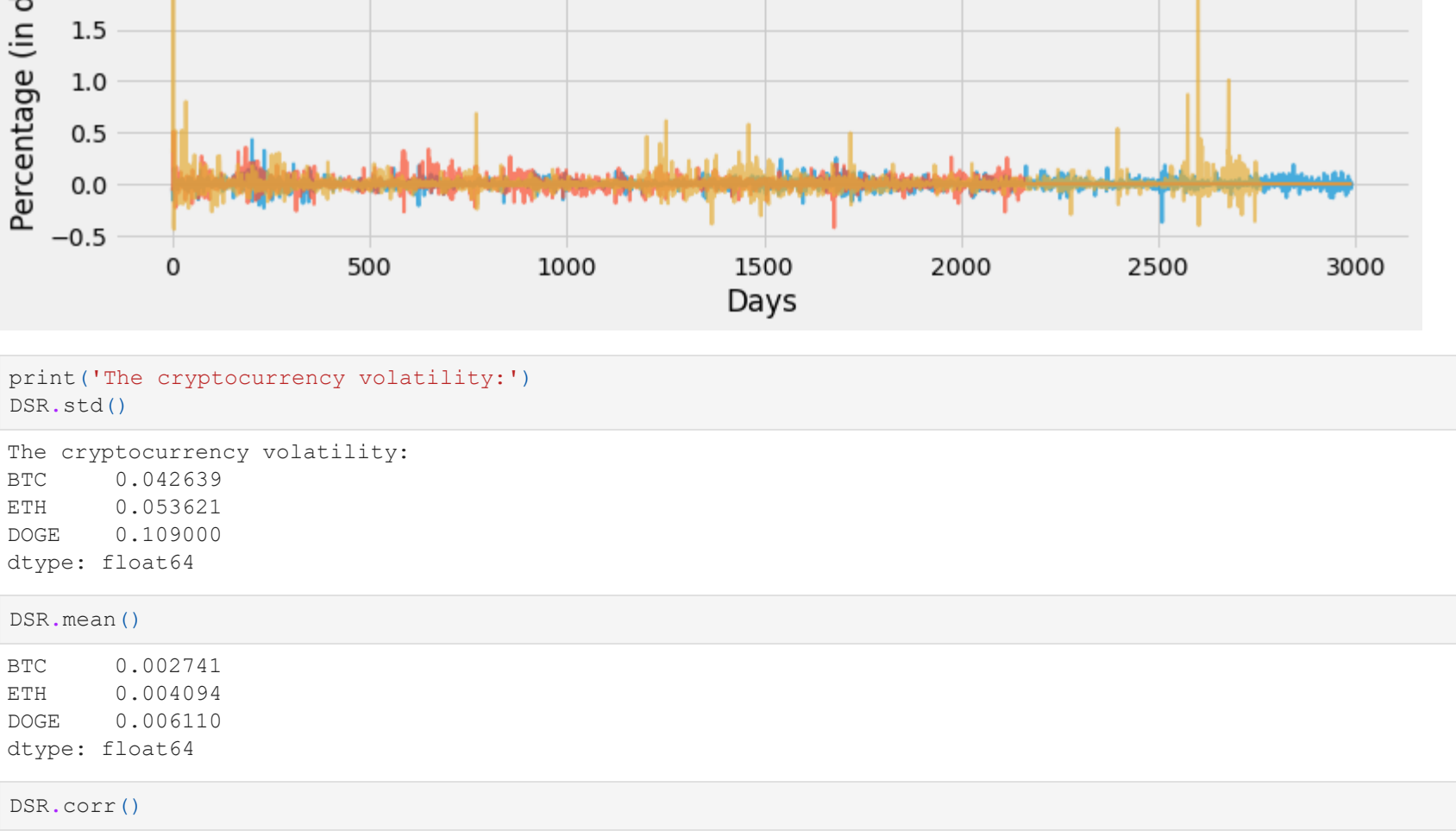
```
In [25]: DSR.tail()

Out[25]:
```

	BTC	ETH	DOGE
2986	0.009679	0.0	0.0
2987	0.022760	0.0	0.0
2988	0.017861	0.0	0.0
2989	-0.043692	0.0	0.0
2990	0.014496	0.0	0.0

```
In [26]: plt.figure(figsize=(12,4,4.5))
for c in DSR.columns.values:
    plt.plot(DSR.index, DSR[c], label=c, lw=2, alpha=.7)
plt.title('Daily Simple Returns')
plt.ylabel('Percentage (in decimal form)')
plt.xlabel('Days')
plt.legend(DSR.columns.values, loc='upper right')
plt.show()

Daily Simple Returns
```



```
In [27]: print('The cryptocurrency volatility:')
DSR.std()

The cryptocurrency volatility:
BTC      0.042639
ETH      0.053621
DOGE     0.109000
dtype: float64

In [28]: DSR.mean()

Out[28]:
```

```
BTC      0.002741
ETH      0.004094
DOGE     0.006110
dtype: float64


In [29]: DSR.corr()

Out[29]:
```

	BTC	ETH	DOGE
BTC	1.000000	-0.005488	-0.030597
ETH	-0.005488	1.000000	0.024557
DOGE	-0.030597	0.024557	1.000000

```
In [30]: import seaborn as sns
plt.subplots(figsize=(11,11))
sns.heatmap(DSR.corr(), annot=True, fmt='.2%')

Out[30]: <AxesSubplot:~>
```



```
In [31]: DCSR = (DSR+1).cumprod()
DCSR

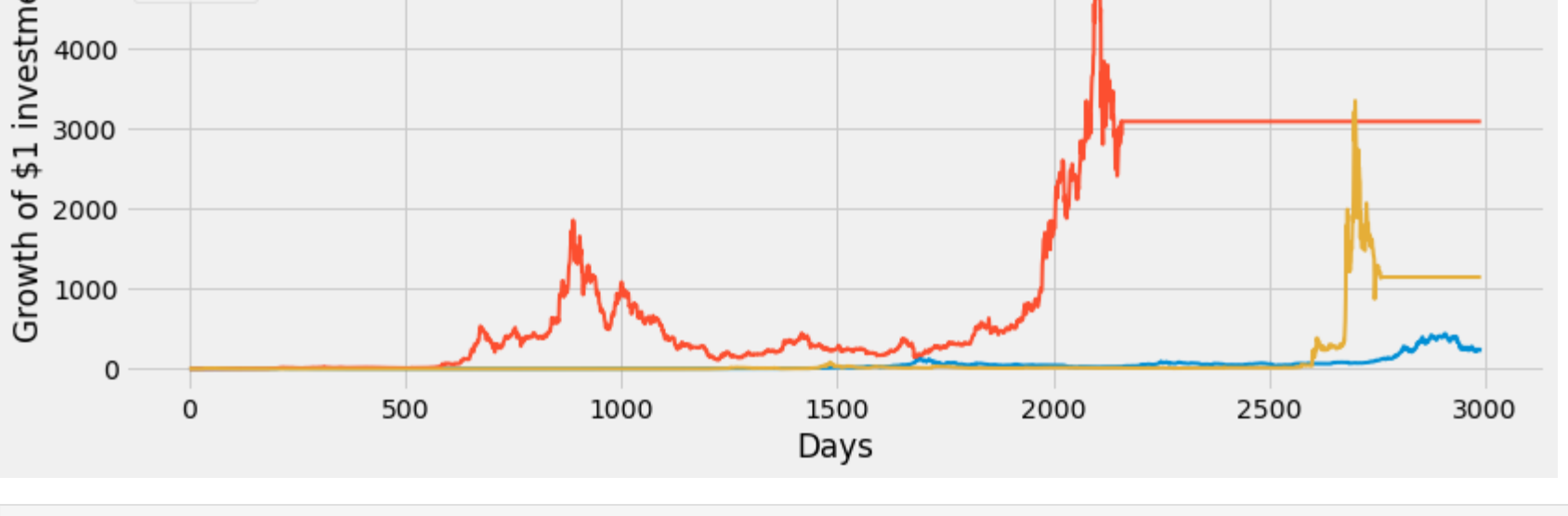
Out[31]:
```

	BTC	ETH	DOGE
0	NaN	NaN	NaN
1	0.961672	0.931732	1.311672
2	0.809395	0.940428	1.766372
3	0.727895	1.417529	5.678134
4	0.676283	1.616089	3.438004
...
2986	234.516744	3085.891871	1145.023290
2987	239.854366	3085.891871	1145.023290
2988	244.138518	3085.891871	1145.023290
2989	233.471731	3085.891871	1145.023290
2990	236.856199	3085.891871	1145.023290

2991 rows x 3 columns

```
In [32]: plt.figure(figsize=(12,2,4.5))
for c in DCSR.columns.values:
    plt.plot(DCSR.index, DCSR[c], lw=2, label=c)
plt.title('Daily Cumulative Simple Return')
plt.xlabel('Days')
plt.ylabel('Growth of $1 investment')
plt.legend(DCSR.columns.values, loc='upper left', fontsize=10)
plt.show()

Daily Cumulative Simple Return
```



```
In [ ]: new_predictions = pd.DataFrame([svm_prediction1,y1_test, svm_prediction2, y2_test, svm_prediction3[:30],y3_test
new_predictions = new_predictions.transpose()
new_predictions.head()
```