In [30]:
```python
# Importing the libraries
import numpy as np
import pandas as pd
```

In [31]:
```python
# Load the data
df_btc=pd.read_csv('coin_Bitcoin.csv')
df_eth=pd.read_csv('coin_Ethereum.csv')
df_ltc=pd.read_csv('coin_Litecoin.csv')
```

In [32]:
```python
# print the first 5 rows of data for BTC
df_btc.head()
```

Out[32]:

	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 [33]:
```python
# print the first 5 rows of data for ETH
df_eth.head()
```

Out[33]:

	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 [34]:
```python
# print the first 5 rows of data for LTC
df_ltc.head()
```

Out[34]:

	SNo	Name	Symbol	Date	High	Low	Open	Close	Volume	Marketcap
0	1	Litecoin	LTC	2013-04-29 23:59:59	4.57360	4.22564	4.36676	4.38390	0.0	7.538896e+07
1	2	Litecoin	LTC	2013-04-30 23:59:59	4.57238	4.16896	4.40352	4.29649	0.0	7.402092e+07
2	3	Litecoin	LTC	2013-05-01 23:59:59	4.35686	3.52029	4.28954	3.80101	0.0	6.560460e+07
3	4	Litecoin	LTC	2013-05-02 23:59:59	4.03930	3.00717	3.78002	3.37198	0.0	5.828798e+07
4	5	Litecoin	LTC	2013-05-03 23:59:59	3.45361	2.39594	3.39044	3.04491	0.0	5.269485e+07

In [35]:
```python
# Create a new dataframe that holds the closing price of all 3 crypto currencies
df=pd.DataFrame({'BTC':df_btc['Close'],
                 'ETH':df_eth['Close'],
                 'LTC':df_ltc['Close']})
```

In [36]:
```python
# Show the new dataframe
df
```

Out[36]:

	BTC	ETH	LTC
0	144.539993	0.753325	4.383900
1	139.000000	0.701897	4.296490
2	116.989998	0.708448	3.801010
3	105.209999	1.067860	3.371980
4	97.750000	1.217440	3.044910
...	...	...	...
2986	33897.048590	NaN	136.943696
2987	34668.548402	NaN	140.279688
2988	35287.779766	NaN	144.905849
2989	33746.002456	NaN	138.073246
2990	34235.193451	NaN	138.985636

2991 rows × 3 columns

In [37]:
```python
# Get statistics on the data
df.describe()
```

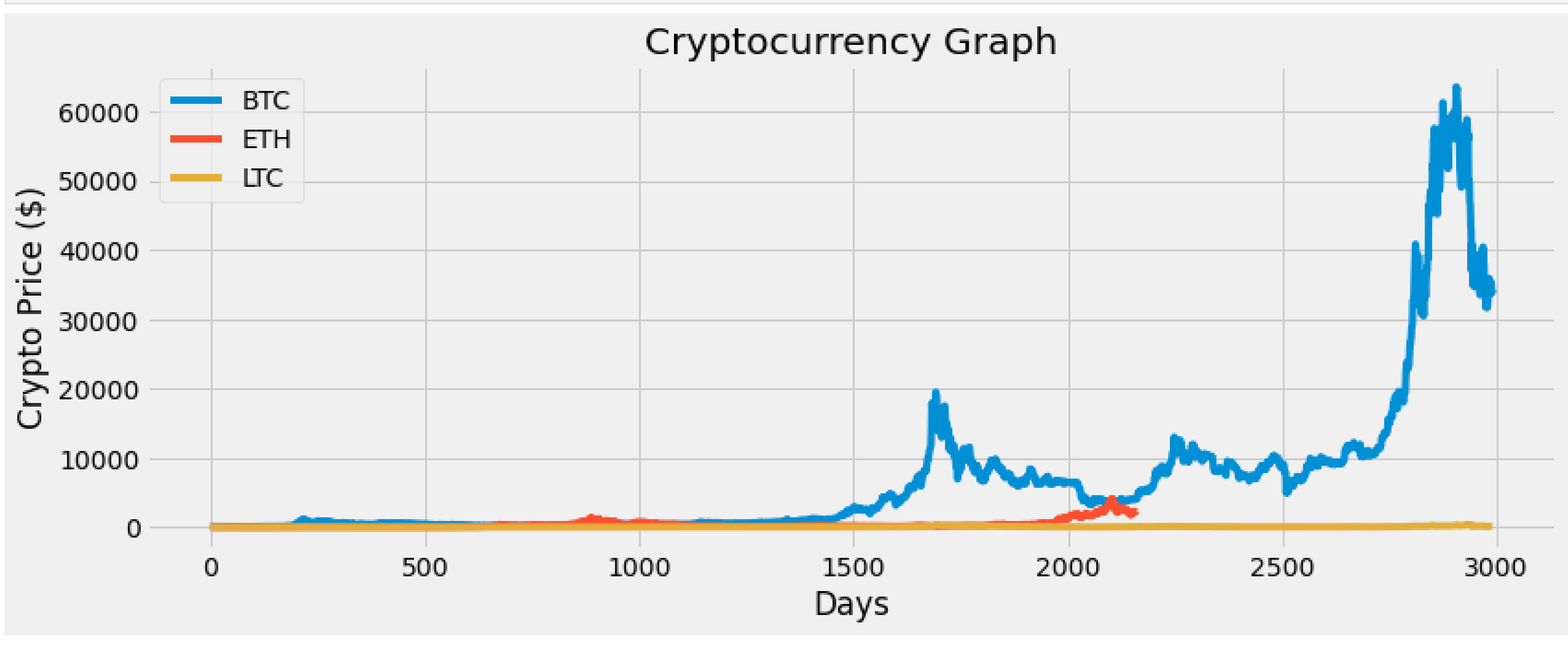
Out[37]:

	BTC	ETH	LTC
count	2991.000000	2160.000000	2991.000000
mean	6711.290443	383.910691	49.279008
std	11298.141921	601.078766	63.240464
min	68.431000	0.434829	1.157010
25%	430.569489	13.819200	3.794135
50%	2286.409912	198.643691	29.900200
75%	8576.238715	386.435272	62.025043
max	63503.457930	4168.701049	386.450779

In [38]:
```python
# Visualize the cryptocurrency closing prices
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()
```



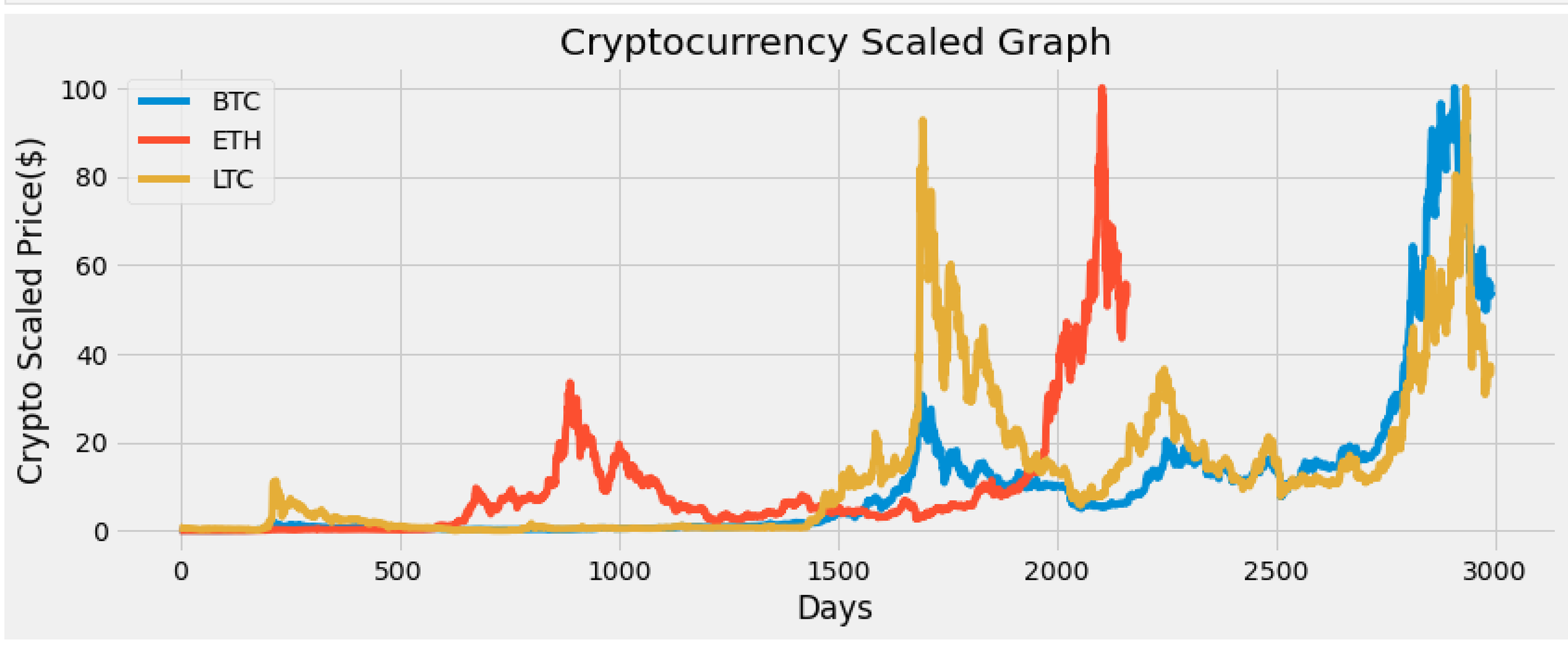
In [39]:
```python
# Scale the data
from sklearn import preprocessing
min_max_scaler=preprocessing.MinMaxScaler(feature_range=(0,100))
scaled=min_max_scaler.fit_transform(df)
scaled
```

Out[39]:
```
array([[1.19979445e-01, 7.64097042e-03, 8.37514246e-01],
       [1.11246111e-01, 6.40717302e-03, 8.14827668e-01],
       [7.65491881e-02, 6.56433591e-03, 6.86229611e-01],
       ...,
       [5.55203497e+01,            nan, 3.73088928e+01],
       [5.30898670e+01,            nan, 3.55355437e+01],
       [5.38610356e+01,            nan, 3.57723475e+01]])
```

In [40]:
```python
# Convert the scaled data into a dataframe
df_scale=pd.DataFrame(scaled,columns=df.columns)
```

In [41]:
```python
# Visualize the scaled data
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()
```



In [42]:
```python
# Get the daily simple return
DSR=df.pct_change(1)
DSR
```

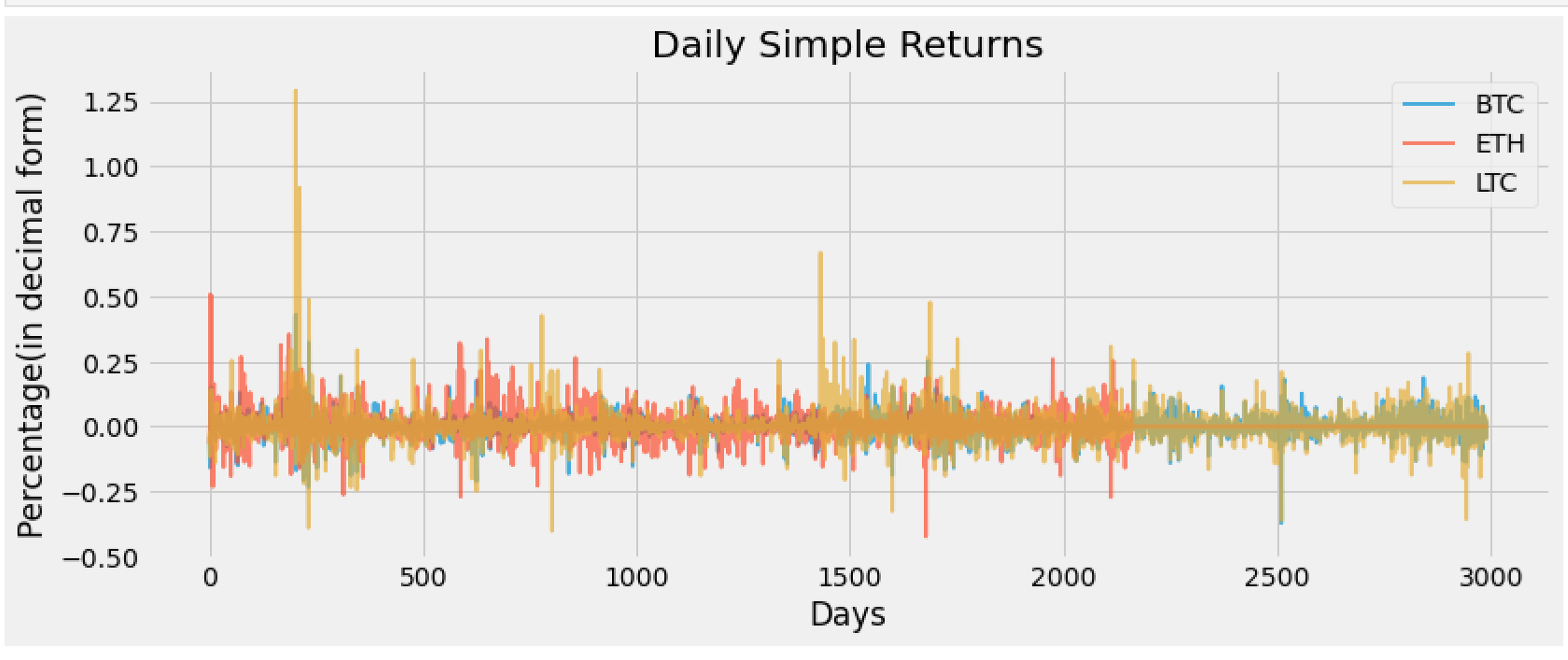
Out[42]:

	BTC	ETH	LTC
0	NaN	NaN	NaN
1	-0.038328	-0.068268	-0.019939
2	-0.158345	0.009333	-0.115322
3	-0.100692	0.507323	-0.112873
4	-0.070906	0.140075	-0.096996
...	...	...	...
2986	0.009679	0.000000	-0.004607
2987	0.022760	0.000000	0.024360
2988	0.017861	0.000000	0.032978
2989	-0.043692	0.000000	-0.047152
2990	0.014496	0.000000	0.006608

2991 rows × 3 columns

In [43]:
```python
# Visualize the daily simple returns
plt.figure(figsize=(12,4.5))
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()
```



In [44]:
```python
# Get the volatility
print('The cryptocurrency volatility')
DSR.std()
```

```
The cryptocurrency volatility
```

Out[44]:
```
BTC    0.042639
ETH    0.053621
LTC    0.068532
dtype: float64
```

In [45]:
```python
# Show the mean / average daily simple return
DSR.mean()
```

Out[45]:
```
BTC    0.002741
ETH    0.004094
LTC    0.003262
dtype: float64
```

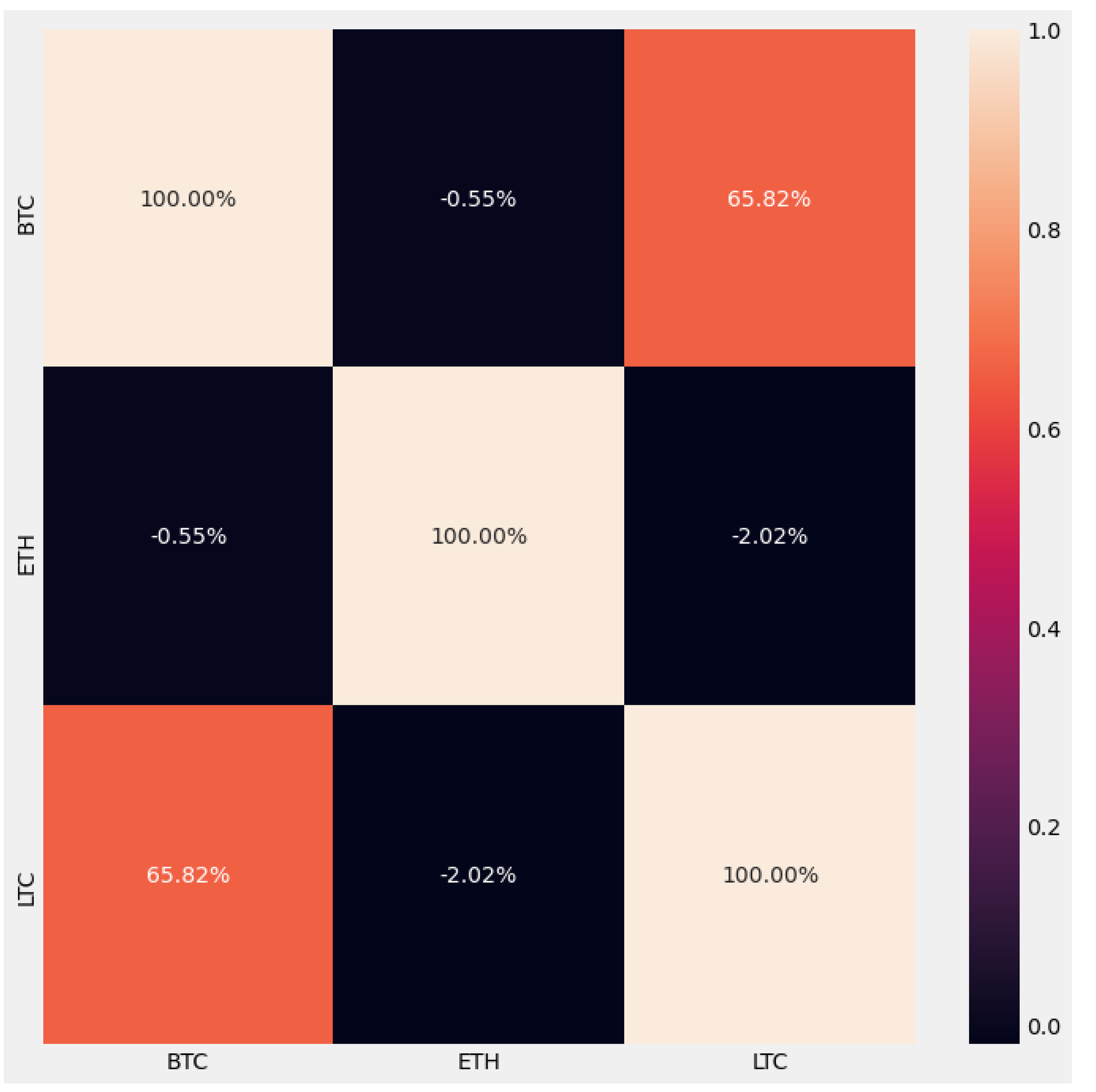
In [46]:
```python
# Get the correlation
DSR.corr()
```

Out[46]:

	BTC	ETH	LTC
BTC	1.000000	-0.005488	0.658199
ETH	-0.005488	1.000000	-0.020173
LTC	0.658199	-0.020173	1.000000

In [47]:
```python
# Visualize the correlation
import seaborn as sns
plt.subplots(figsize=(11,11))
sns.heatmap(DSR.corr(),annot=True,fmt='.2%')
```

Out[47]: <AxesSubplot:>



In [48]:
```python
# Get the daily cumulative simple returns
DSCR=(DSR+1).cumprod()

#show
DSCR
```

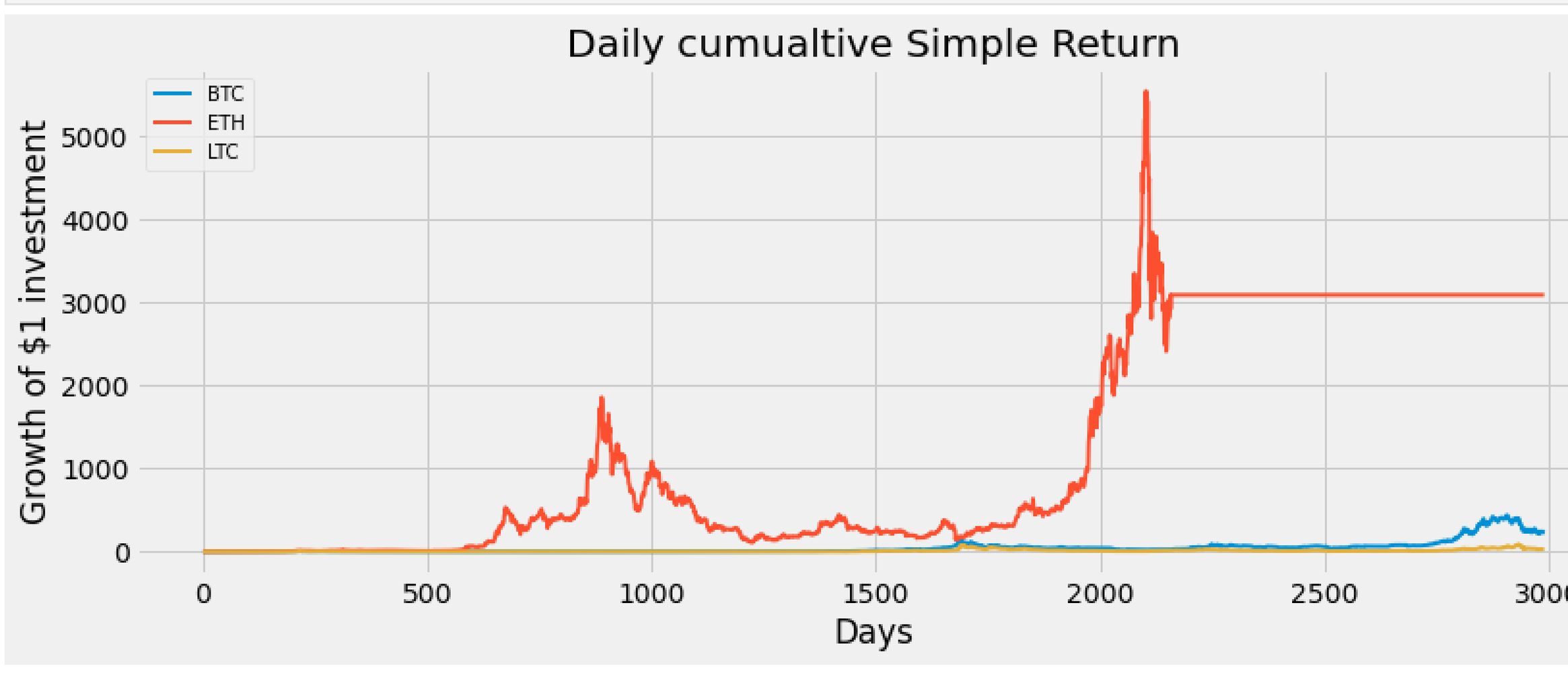
Out[48]:

	BTC	ETH	LTC
0	NaN	NaN	NaN
1	0.961672	0.931732	0.980061
2	0.809395	0.940428	0.867038
3	0.727895	1.417529	0.769174
4	0.676283	1.616089	0.694566
...	...	...	...
2986	234.516744	3085.891871	31.237868
2987	239.854366	3085.891871	31.998833
2988	244.138518	3085.891871	33.054094
2989	233.471731	3085.891871	31.495527
2990	236.856199	3085.891871	31.703650

2991 rows × 3 columns

In [49]:
```python
# Visualize the daily cumulative simple returns
plt.figure(figsize=(12.2,4.5))
for c in DSCR.columns.values:
    plt.plot(DSCR.index,DSCR[c],lw=2,label=c)

plt.title('Daily cumualtive Simple Return')
plt.xlabel('Days')
plt.ylabel('Growth of $1 investment')
plt.legend(DSCR.columns.values,loc='upper left',fontsize=10)
plt.show()
```



In [ ]: