In [1]:
```python
# Importing required packages:

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
# Invoke data from CSV file:

df_nissan=pd.read_csv('nissan_car.csv', index_col="Date", parse_dates=True)
df_tesla=pd.read_csv('tesla_car.csv', index_col="Date", parse_dates=True)
df_volkswagen=pd.read_csv('volkswagen.csv', index_col="Date", parse_dates=True)
```

In [3]:
```python
# Printing the first 5 rows for nissan:

df_nissan.head()
```

Out[3]:

Date	Open	High	Low	Close	Adj Close	Volume
2016-08-24	961.5	973.799988	955.299988	972.200012	826.390381	14245400
2016-08-25	976.0	993.000000	969.099976	982.500000	835.145630	15255800
2016-08-26	980.0	984.900024	972.000000	974.000000	827.920410	13899600
2016-08-29	1000.0	1005.000000	996.400024	1001.000000	850.870972	15360000
2016-08-30	995.0	1005.500000	988.200012	1003.000000	852.570984	11450800

In [4]:
```python
# Printing the first 5 rows for tesla:

df_tesla.head()
```

Out[4]:

Date	Open	High	Low	Close	Adj Close	Volume
2016-08-24	45.410000	45.430000	44.444000	44.523998	44.523998	12853500
2016-08-25	44.622002	44.759998	44.153999	44.192001	44.192001	8812500
2016-08-26	44.428001	44.571999	43.764000	43.998001	43.998001	11195000
2016-08-29	44.029999	44.080002	43.000000	43.040001	43.040001	16331500
2016-08-30	43.222000	43.222000	42.104000	42.268002	42.268002	15844500

In [5]:
```python
# Printing the first 5 rows for volkswagen:

df_volkswagen.head()
```

Out[5]:

Date	Open	High	Low	Close	Adj Close	Volume
2016-08-24	123.050003	123.900002	121.150002	122.650002	107.715622	721191
2016-08-25	122.050003	122.400002	119.449997	120.400002	105.739594	1002019
2016-08-26	120.699997	124.250000	120.550003	124.199997	109.076889	1262102
2016-08-29	123.199997	124.900002	121.849998	122.400002	107.496063	589692
2016-08-30	123.250000	125.250000	122.800003	124.800003	109.603836	964915

In [6]:
```python
# Create a new dataframe that holds the all 3 cars:

df=pd.DataFrame({'nissan':df_nissan['Close'],'tesla':df_tesla['Close'],'volkswagen':df_volkswagen['Close']})
```

In [7]:
```python
# Show the new dataframe:

df
```

Out[7]:

Date	nissan	tesla	volkswagen
2016-08-24	972.200012	44.523998	122.650002
2016-08-25	982.500000	44.192001	120.400002
2016-08-26	974.000000	43.998001	124.199997
2016-08-29	1001.000000	43.040001	122.400002
2016-08-30	1003.000000	42.268002	124.800003
...	...	...	...
2021-08-18	581.000000	688.989990	198.539993
2021-08-19	565.700012	673.469971	194.839996
2021-08-20	524.700012	680.260010	192.860001
2021-08-23	542.599976	706.299988	194.539993
2021-08-24	561.000000	NaN	NaN

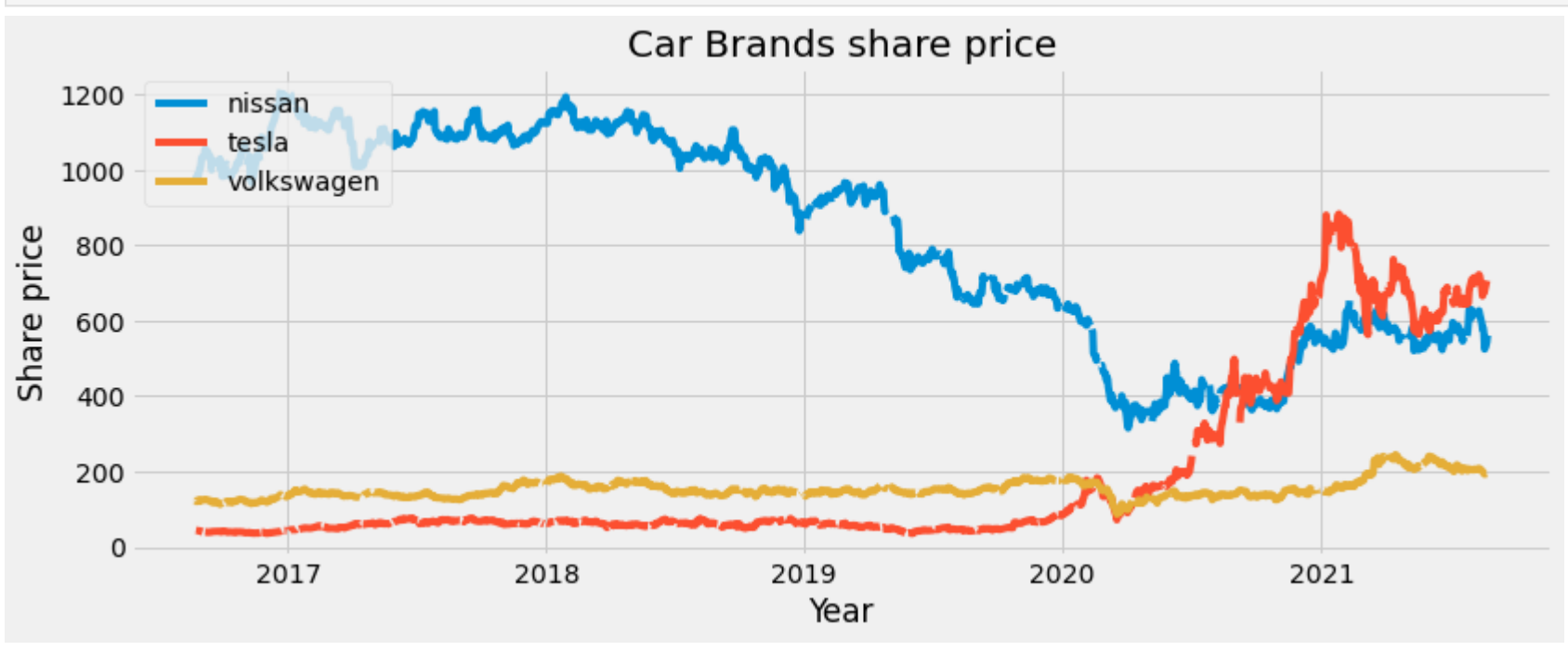
1302 rows × 3 columns

In [8]:
```python
# Get statistics on the data:

df.describe()
```

Out[8]:

	nissan	tesla	volkswagen
count	1241.000000	1258.000000	1266.000000
mean	833.557211	186.397424	153.695276
std	272.248364	230.373307	26.389133
min	315.500000	35.793999	87.199997
25%	579.299988	55.318001	138.000000
50%	935.000000	66.957000	146.730003
75%	1086.000000	172.577003	164.575000
max	1212.000000	883.090027	246.550003

In [9]:
```python
# Visualizing:

plt.style.use('fivethirtyeight')
e_car=df
plt.figure(figsize=(12.2,4.5))
for c in e_car.columns.values:
    plt.plot(e_car[c],label=c)
plt.title('Car Brands share price')
plt.xlabel('Year')
plt.ylabel('Share price')
plt.legend(e_car.columns.values,loc='upper left')
plt.show()
```

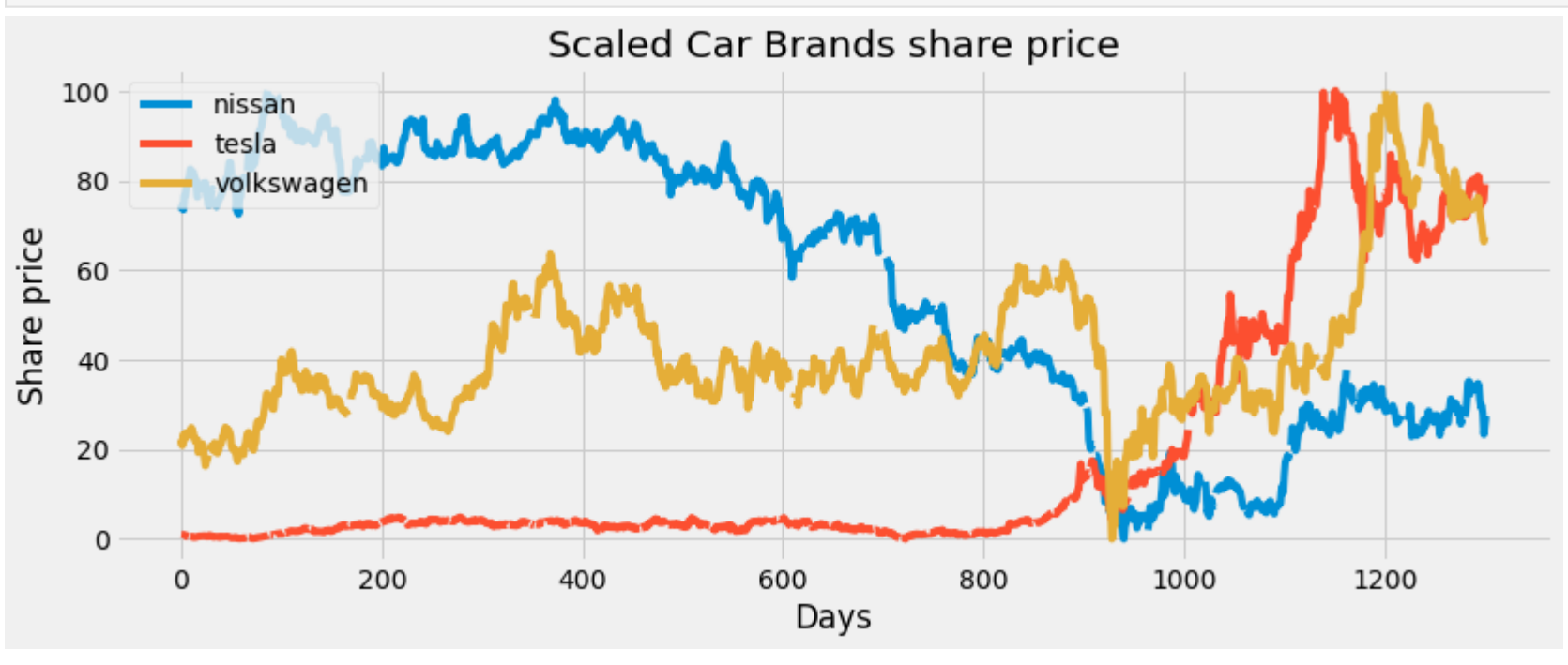


In [10]:
```python
 # Scale the data:
from sklearn import preprocessing
min_max_scaler=preprocessing.MinMaxScaler(feature_range=(0,100))
scaled=min_max_scaler.fit_transform(df)
scaled
```

Out[10]:
```
array([[73.25153508,  1.03033635, 22.24662922],
       [74.40044618,  0.99115324, 20.83464308],
       [73.45231456,  0.96825687, 23.21932765],
       ...,
       [23.33519375, 76.06149323, 66.30687168],
       [25.33184339, 79.13479668, 67.36114964],
       [27.38427217,         nan,         nan]])
```

In [11]:
```python
# Convert the scaled data into a dataframe:

df_scale=pd.DataFrame(scaled,columns=df.columns)
```

In [12]:
```python
# Visualize the scaled data:

plt.style.use('fivethirtyeight')
e_car=df_scale
plt.figure(figsize=(12.2,4.5))
for c in e_car.columns.values:
    plt.plot(e_car[c],label=c)
plt.title('Scaled Car Brands share price')
plt.xlabel('Days')
plt.ylabel('Share price')
plt.legend(e_car.columns.values,loc='upper left')
plt.show()
```



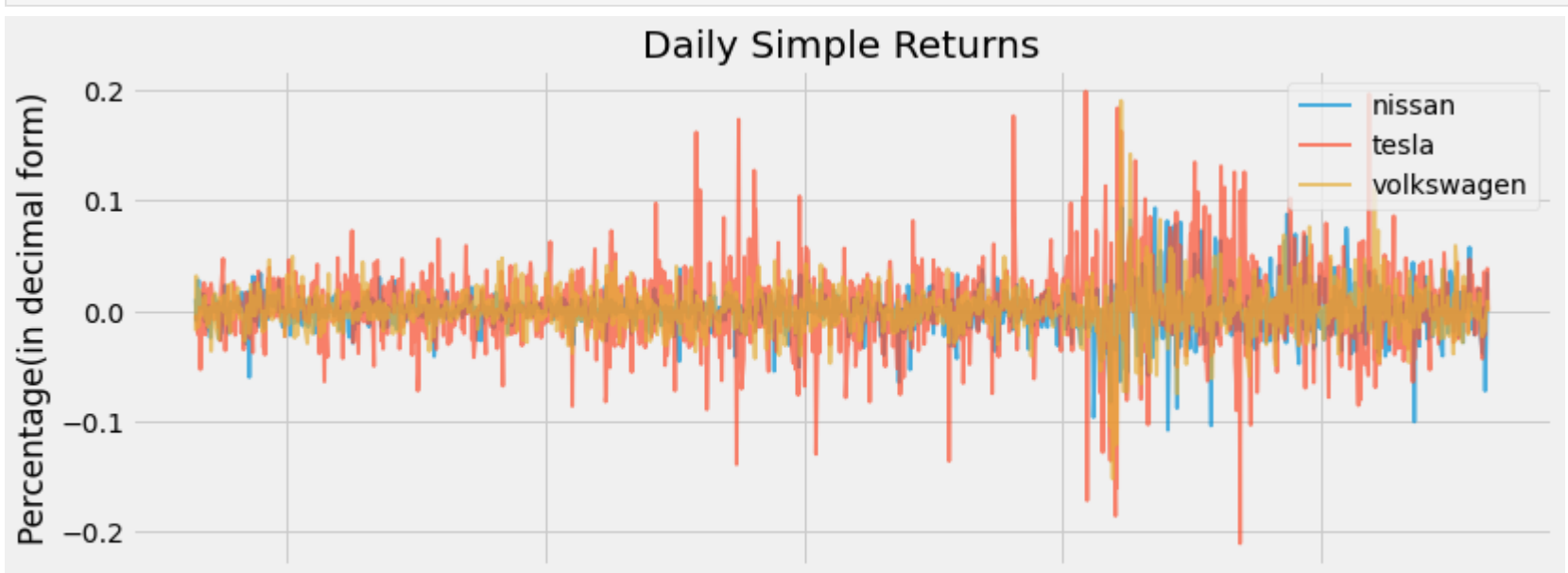
In [13]:
```python
# Get the daily simple return:

DSR=df.pct_change(1)
DSR
```

Out[13]:

Date	nissan	tesla	volkswagen
2016-08-24	NaN	NaN	NaN
2016-08-25	0.010595	-0.007457	-0.018345
2016-08-26	-0.008651	-0.004390	0.031561
2016-08-29	0.027721	-0.021774	-0.014493
2016-08-30	0.001998	-0.017937	0.019608
...	...	...	...
2021-08-18	0.002588	0.034970	-0.008787
2021-08-19	-0.026334	-0.022526	-0.018636
2021-08-20	-0.072477	0.010082	-0.010162
2021-08-23	0.034115	0.038279	0.008711
2021-08-24	0.033911	0.000000	0.000000

1302 rows × 3 columns

In [14]:
```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('Year')
plt.legend(DSR.columns.values,loc='upper right')
plt.show()
```



In [15]:
```python
 # Get the volatility

print('The volatility')
DSR.std()
```

```
The volatility
```

Out[15]:
```
nissan        0.019203
tesla         0.036741
volkswagen    0.020830
dtype: float64
```

In [16]:
```python
# Show the mean / average daily simple return:

DSR.mean()
```

Out[16]:
```
nissan       -0.000238
tesla         0.002798
volkswagen    0.000571
dtype: float64
```

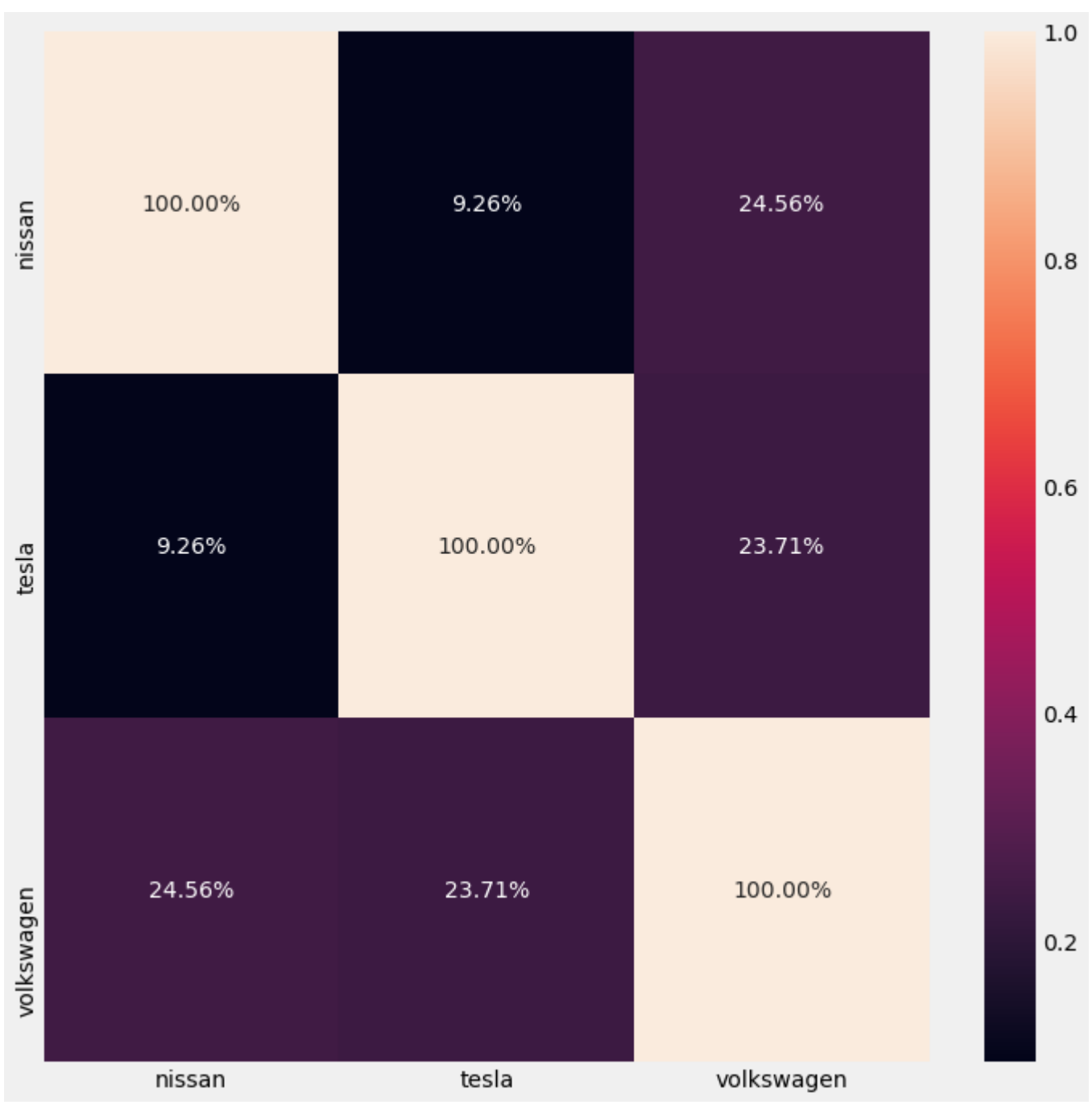
In [17]:
```python
 # Get the correlation:

DSR.corr()
```

Out[17]:

	nissan	tesla	volkswagen
nissan	1.000000	0.092592	0.245615
tesla	0.092592	1.000000	0.237144
volkswagen	0.245615	0.237144	1.000000

In [18]:
```python
 # Visualize the correlation:

plt.subplots(figsize=(11,11))
sns.heatmap(DSR.corr(),annot=True,fmt='.2%')
```

Out[18]:
```
<AxesSubplot:>
```



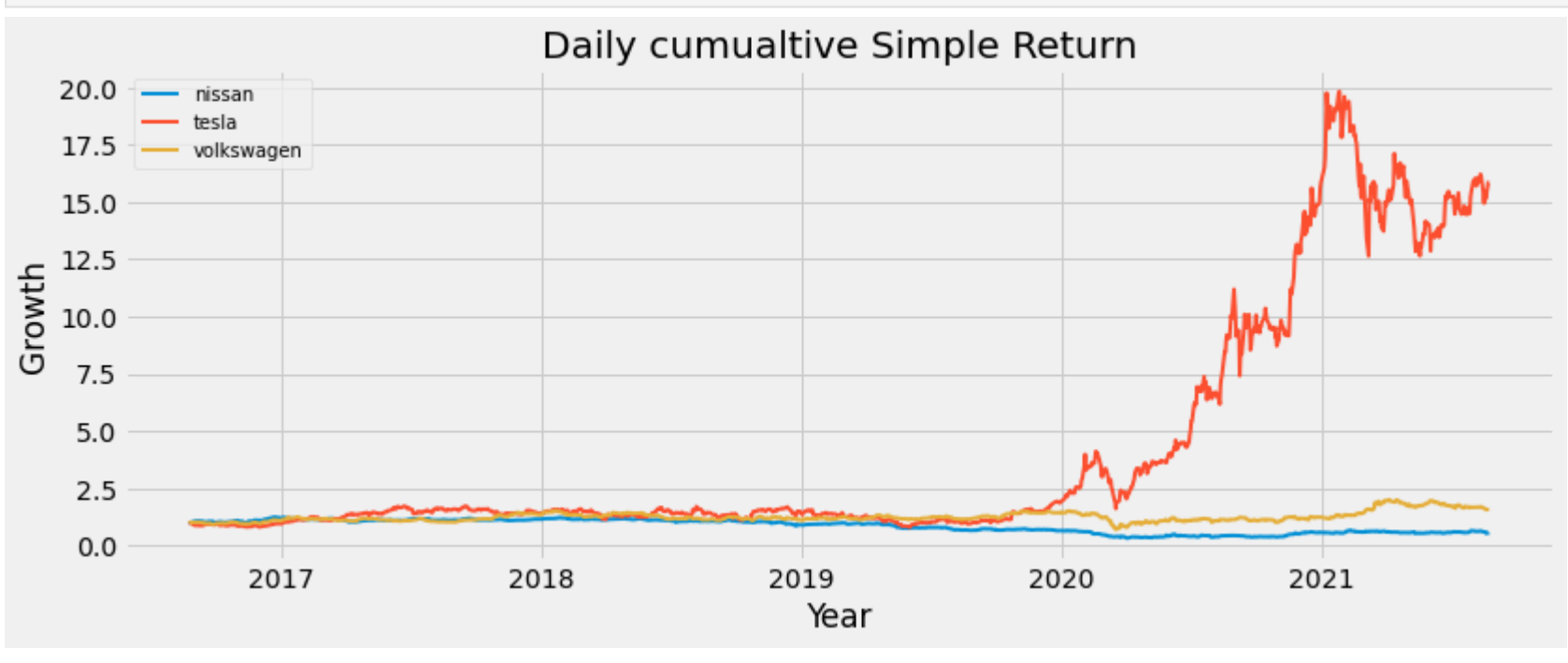
In [19]:
```python
 # Get the daily cumulative simple returns:

DCSR=(DSR+1).cumprod()

#show:

DCSR
```

Out[19]:

Date	nissan	tesla	volkswagen
2016-08-24	NaN	NaN	NaN
2016-08-25	1.010595	0.992543	0.981655
2016-08-26	1.001851	0.988186	1.012638
2016-08-29	1.029624	0.966670	0.997962
2016-08-30	1.031681	0.949331	1.017530
...	...	...	...
2021-08-18	0.597614	15.474576	1.618752
2021-08-19	0.581876	15.125999	1.588585
2021-08-20	0.539704	15.278502	1.572442
2021-08-23	0.558116	15.863355	1.586139
2021-08-24	0.577042	15.863355	1.586139

1302 rows × 3 columns

In [21]:
```python
 # Visualize the daily cumulative simple returns:

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 cumualtive Simple Return')
plt.xlabel('Year')
plt.ylabel('Growth')
plt.legend(DCSR.columns.values,loc='upper left',fontsize=10)
plt.show()
```



In [ ]: