Visualizing FAANG Stocks Project

New Section

Analyze and visualize FAANG stocks, as of the beginning of 2020:

- Facebook (NASDAQ:FB)
- Apple (NASDAQ:AAPL)
- Amazon (NASDAQ:AMZN)
- Netflix (NASDAQ: NFLX)
- Google (NASDAQ:GOOGL)

Using Pandas, Pandas-Datarader, and Matplotlib, take a visual look into the similarities and differences between these stocks during the six month period from January through June 2020. Perform the following analysis:

1. Visualize the stock prices using matplotlib
2. Calculate and visualize the daily simple rate of return
3. Calculate and visualize the mean rates of return
4. Calculate and visualize the variances of the returns
5. Calculate and visualize the standard deviations of the returns
6. Write a short thesis based on the correlations between the tech stocks

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

In [26]:
```
# Import pandas data reader module
import pandas_datareader as web
```

In [3]:
```
# Import visualization package
import matplotlib.pyplot as plt
%matplotlib inline
```

In [4]:
```
# Load adjusted closings for the FAANG Stocks
# Define stocks
# Create dates
# Retrieve data
symbols = ['FB', 'AAPL', 'AMZN', 'NFLX', 'GOOGL']
start_date = "2020-01-01"
end_date = "2020-06-01"
stock_data = web.get_data_yahoo(symbols, start_date, end_date)
```

In [5]:
```
# View data
stock_data
```

Out[5]:

Attributes	Adj Close					Close				
Symbols	FB	AAPL	AMZN	NFLX	GOOGL	FB	AAPL	AMZN	NFLX	G
Date										
2020-01-02	209.779999	298.829956	1898.010010	329.809998	1368.680054	209.779999	300.350006	1898.010010	329.809998	1
2020-01-03	208.669998	295.924713	1874.969971	325.899994	1361.520020	208.669998	297.429993	1874.969971	325.899994	1
2020-01-06	212.600006	298.282715	1902.880005	335.829987	1397.810059	212.600006	299.799988	1902.880005	335.829987	1
2020-01-07	213.059998	296.879883	1906.859985	330.750000	1395.109985	213.059998	298.390015	1906.859985	330.750000	1
2020-01-08	215.220001	301.655548	1891.969971	339.260010	1405.040039	215.220001	303.190002	1891.969971	339.260010	1
...	...	...	...	...	...	...	...	...	...	...
2020-05-26	232.199997	316.730011	2421.860107	414.769989	1421.369995	232.199997	316.730011	2421.860107	414.769989	1
2020-05-27	229.139999	318.109985	2410.389893	419.890015	1420.280029	229.139999	318.109985	2410.389893	419.890015	1
2020-05-28	225.460007	318.250000	2401.100098	413.440002	1418.239990	225.460007	318.250000	2401.100098	413.440002	1
2020-05-29	225.089996	317.940002	2442.370117	419.730011	1433.520020	225.089996	317.940002	2442.370117	419.730011	1
2020-06-01	231.910004	321.850006	2471.040039	425.920013	1434.869995	231.910004	321.850006	2471.040039	425.920013	1

104 rows × 30 columns

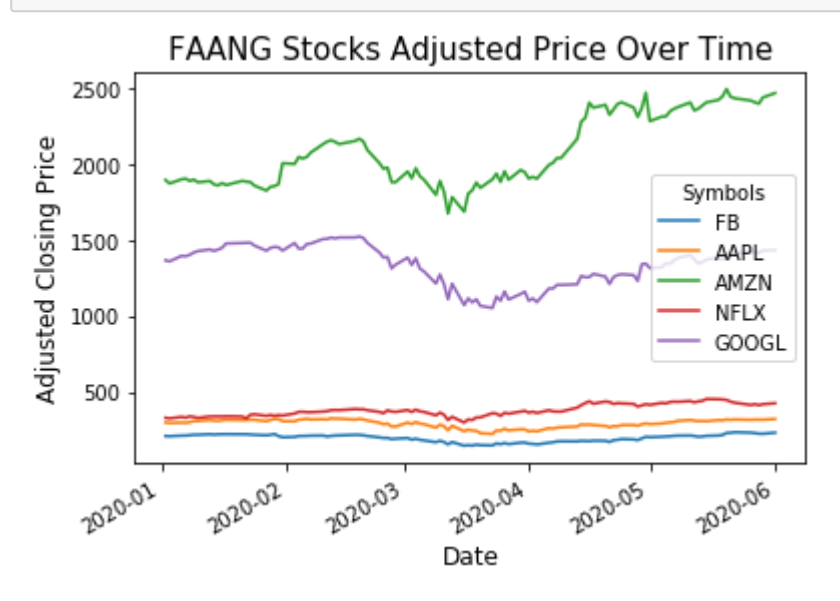
In [6]:
```
# View adj. close data
stock_data['Adj Close']
```

Out[6]:

Symbols	FB	AAPL	AMZN	NFLX	GOOGL
Date					
2020-01-02	209.779999	298.829956	1898.010010	329.809998	1368.680054
2020-01-03	208.669998	295.924713	1874.969971	325.899994	1361.520020
2020-01-06	212.600006	298.282715	1902.880005	335.829987	1397.810059
2020-01-07	213.059998	296.879883	1906.859985	330.750000	1395.109985
2020-01-08	215.220001	301.655548	1891.969971	339.260010	1405.040039
...	...	...	...	...	...
2020-05-26	232.199997	316.730011	2421.860107	414.769989	1421.369995
2020-05-27	229.139999	318.109985	2410.389893	419.890015	1420.280029
2020-05-28	225.460007	318.250000	2401.100098	413.440002	1418.239990
2020-05-29	225.089996	317.940002	2442.370117	419.730011	1433.520020
2020-06-01	231.910004	321.850006	2471.040039	425.920013	1434.869995

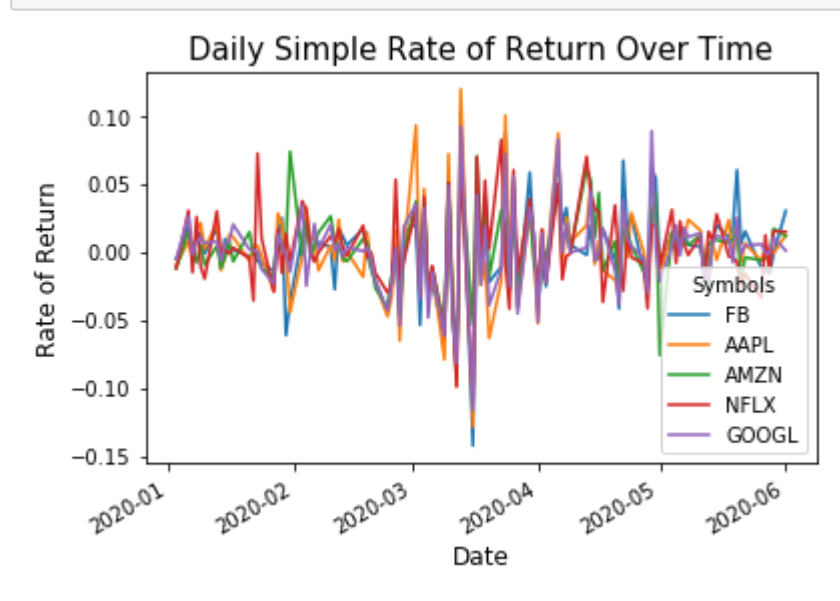
104 rows × 5 columns

In [7]:
```
# Plot adj. closing prices over time
stock_data_closing_prices = stock_data['Adj Close']
stock_data_closing_prices.plot()
plt.xlabel('Date', fontsize=12)
plt.ylabel('Adjusted Closing Price', fontsize=12)
plt.title('FAANG Stocks Adjusted Price Over Time', fontsize=15)
plt.figure(figsize=(15,15))
plt.show()
plt.savefig('adjpot.png')
```



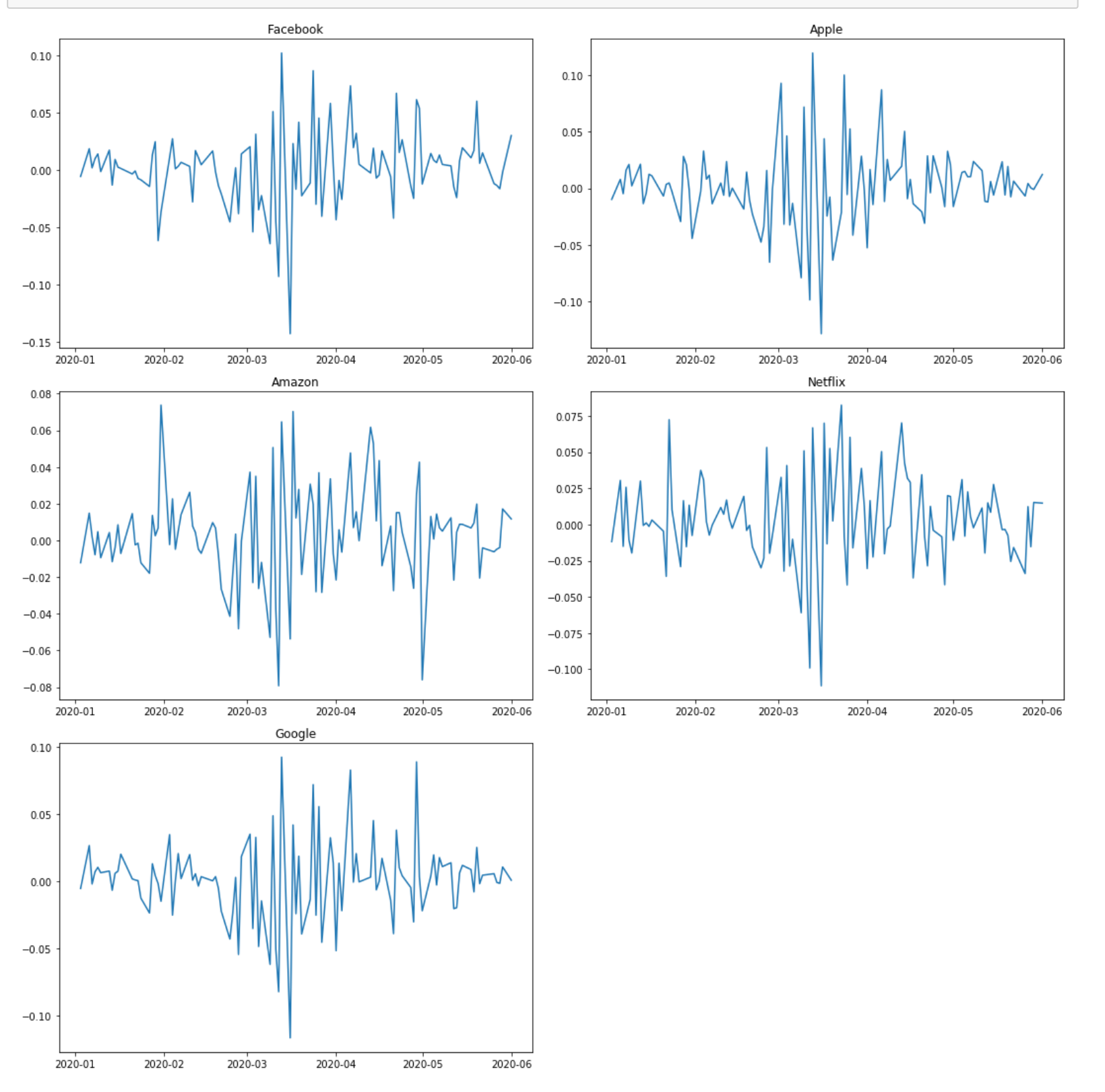

```
<Figure size 1080x1080 with 0 Axes>
<Figure size 432x288 with 0 Axes>
```

In [8]:
```
# Calculate and plot the daily simple rate of return over time
stock_data_daily_return = stock_data_closing_prices.pct_change()
stock_data_daily_return.plot()
plt.xlabel('Date', fontsize=12)
plt.ylabel('Rate of Return', fontsize=12)
plt.title('Daily Simple Rate of Return Over Time', fontsize=15)
plt.figure(figsize=(15,15))
plt.show()
plt.savefig('dailysimplerot.png')
```




```
<Figure size 1080x1080 with 0 Axes>
<Figure size 432x288 with 0 Axes>
```

In [9]:
```
# Create subplots of daily simple rate of return for each stock
fig = plt.figure(figsize=(15,15))
ax1 = fig.add_subplot(321)
ax2 = fig.add_subplot(322)
ax3 = fig.add_subplot(323)
ax4 = fig.add_subplot(324)
ax5 = fig.add_subplot(325)
ax1.plot(stock_data['Adj Close']['FB'].pct_change())
ax1.set_title('Facebook')
ax2.plot(stock_data['Adj Close']['AAPL'].pct_change())
ax2.set_title('Apple')
ax3.plot(stock_data['Adj Close']['AMZN'].pct_change())
ax3.set_title('Amazon')
ax4.plot(stock_data['Adj Close']['NFLX'].pct_change())
ax4.set_title('Netflix')
ax5.plot(stock_data['Adj Close']['GOOGL'].pct_change())
ax5.set_title('Google')
plt.tight_layout()
plt.show()
plt.savefig('subplots.png')
```




```
<Figure size 432x288 with 0 Axes>
```

In [10]:
```
# Daily mean rate of return
daily_mean = stock_data_daily_return.mean()
daily_mean
```

Out[10]:
```
Symbols
FB       0.001563
AAPL     0.001349
AMZN     0.002924
NFLX     0.003002
GOOGL    0.000948
dtype: float64
```

In [11]:
```
# Daily mean index for the x axis
daily_mean.keys()
```

Out[11]:
```
Index(['FB', 'AAPL', 'AMZN', 'NFLX', 'GOOGL'], dtype='object', name='Symbols')
```

In [12]:
```
# Grab each daily mean value for the y axis
height = []
for key in daily_mean.keys():
  height.append(daily_mean[key])
height
```

Out[12]:
```
[0.0015631549917857611,
 0.0013485405581890463,
 0.002924018669142578,
 0.0030015786210504996,
 0.0009478476568743521]
```

In [13]:
```
# Arrange keys on x axis based on length
x_pos = np.arange(len(daily_mean.keys()))
x_pos
```

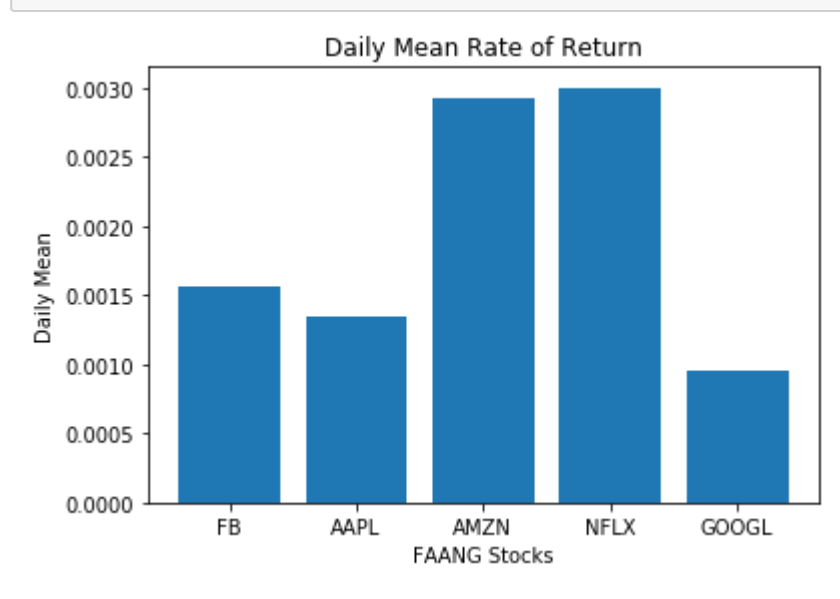
Out[13]:
```
array([0, 1, 2, 3, 4])
```

In [14]:
```
# Plot bars
plt.bar(x_pos, height)

# Create names on the x-axis
plt.xticks(x_pos, daily_mean.keys())

# Label chart
plt.xlabel('FAANG Stocks')
plt.ylabel('Daily Mean')
plt.title('Daily Mean Rate of Return')

# Show graphic
plt.figure(figsize=(15,15))
plt.show()
plt.savefig('dailymeanrot.png')
```




```
<Figure size 1080x1080 with 0 Axes>
<Figure size 432x288 with 0 Axes>
```

Netlix has the highest mean simple rate of return over the period of data collected. Thus Netflix would have been a good choice for investment over this period of time. Google, on the other hand, has the lowest mean simple rate of return over the period.

In [15]:
```
# Calculate variance
daily_var = stock_data_daily_return.var()
daily_var
```

Out[15]:
```
Symbols
FB       0.001176
AAPL     0.001265
AMZN     0.000726
NFLX     0.001037
GOOGL    0.000984
dtype: float64
```

In [16]:
```
# Var index for the x axis
daily_var.keys()
```

Out[16]:
```
Index(['FB', 'AAPL', 'AMZN', 'NFLX', 'GOOGL'], dtype='object', name='Symbols')
```

In [17]:
```
# Grab each variance value for the y axis
height = []
for key in daily_var.keys():
  height.append(daily_var[key])
height
```

Out[17]:
```
[0.001175645980770691,
 0.0012652357300907296,
 0.0007255268587608915,
 0.0010368280019109847,
 0.0009835590252993505]
```

In [18]:
```
# Arrange keys on x axis based on length
x_pos = np.arange(len(daily_var.keys()))
x_pos
```

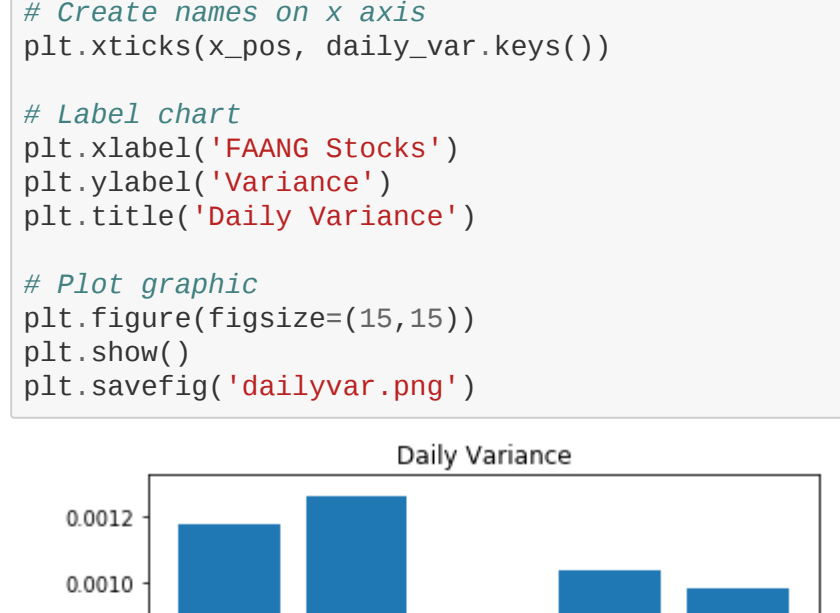
Out[18]:
```
array([0, 1, 2, 3, 4])
```

In [19]:
```
# Plot bars
plt.bar(x_pos, height)

# Create names on x axis
plt.xticks(x_pos, daily_var.keys())

# Label chart
plt.xlabel('FAANG Stocks')
plt.ylabel('Variance')
plt.title('Daily Variance')

# Plot graphic
plt.figure(figsize=(15,15))
plt.show()
plt.savefig('dailyvar.png')
```




```
<Figure size 1080x1080 with 0 Axes>
<Figure size 432x288 with 0 Axes>
```

Apple shows the highest variance of all the stocks, indicating it can be a riskier investment. Amazon shows the lowest variance, indicating that the returns are more predictable. This goes along with the typical understanding of higher return, high risks stocks, and lower return, low risk stocks.

In [20]:
```
# Calculate std. deviation
daily_std = stock_data_daily_return.std()
daily_std
```

Out[20]:
```
Symbols
FB       0.034288
AAPL     0.035570
AMZN     0.026936
NFLX     0.032200
GOOGL    0.031362
dtype: float64
```

In [21]:
```
# Std. deviation index for the x axis
daily_std.keys()
```

Out[21]:
```
Index(['FB', 'AAPL', 'AMZN', 'NFLX', 'GOOGL'], dtype='object', name='Symbols')
```

In [22]:
```
# Grab each std. deviation value for the y axis
height = []
for key in daily_std.keys():
  height.append(daily_std[key])
height
```

Out[22]:
```
[0.03428769430525609,
 0.03557015223597911,
 0.026935605780470048,
 0.03219981369373097,
 0.03136174461504574]
```

In [23]:
```
# Arrange keys on x axis based on length
x_pos = np.arange(len(daily_std.keys()))
x_pos
```

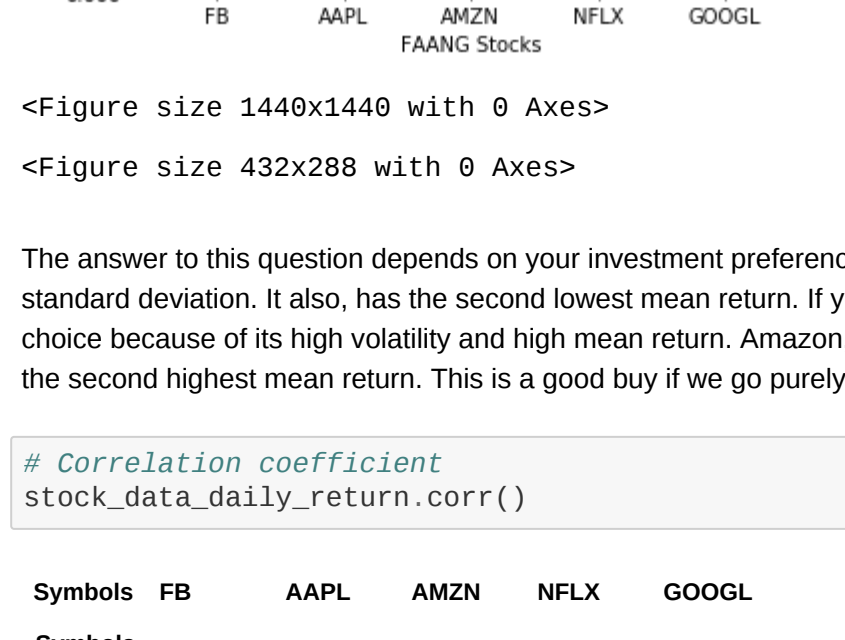
Out[23]:
```
array([0, 1, 2, 3, 4])
```

In [27]:
```
# Plot bars
plt.bar(x_pos, height)

# Create names on x axis
plt.xticks(x_pos, daily_std.keys())

# Label chart
plt.xlabel('FAANG Stocks')
plt.ylabel('Std. Deviation')
plt.title('Daily Std. Deviation')

# Show graphic
plt.figure(figsize=(20,20))
plt.show()
plt.savefig('dailystdev.png')
```




```
<Figure size 1440x1440 with 0 Axes>
<Figure size 432x288 with 0 Axes>
```

The answer to this question depends on your investment preferences. Apple is the most volatile stock, as it has the largest standard deviation. It also, has the second lowest mean return. If you are a more risky investor, Netflix could be your stock of choice because of its high volatility and high mean return. Amazon, on the other hand, is the least volatile stock, and also has the second highest mean return. This is a good buy if we go purely off of its low volatility and high mean return value.

In [25]:
```
# Correlation coefficient
stock_data_daily_return.corr()
```

Out[25]:

Symbols	FB	AAPL	AMZN	NFLX	GOOGL
Symbols					
FB	1.000000	0.850796	0.666986	0.591586	0.883529
AAPL	0.850796	1.000000	0.705728	0.637058	0.890887
AMZN	0.666986	0.705728	1.000000	0.740197	0.723769
NFLX	0.591586	0.637058	0.740197	1.000000	0.636848
GOOGL	0.883529	0.890887	0.723769	0.636848	1.000000

None of the stocks are negatively correlated. Apple and Google are highly correlated, while Facebook and Netflix exhibit the lowest correlation.