**STOCK MARKET DATA ANALYSIS FROM 2013 TO 2021**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('whitegrid')

plt.style.use("fivethirtyeight")

%matplotlib inline

# For reading stock data from yahoo

from pandas\_datareader.data import DataReader

import yfinance as yf

# For time stamps

from datetime import datetime

tech\_list = ['AAPL', 'GOOG', 'MSFT', 'AMZN']

# Set up End and Start times for data grab

tech\_list = ['AAPL', 'GOOG', 'MSFT', 'AMZN']

end = datetime.now()

start = datetime(end.year - 1, end.month, end.day)

for stock in tech\_list:

globals()[stock] = yf.download(stock, start, end)

for company, company\_name in zip(company\_list, tech\_list):

# company["company\_name"] = company\_name

company\_list = [AAPL, GOOG, MSFT, AMZN]

company\_name = [ “MICROSOFT”, “AMAZON” "APPLE", "GOOGLE", ]

for company, com\_name in zip(company\_list, company\_name):

company["company\_name"] = com\_name

df = pd.concat(company\_list, axis=0)

df.tail(10)

Open High Low Close Adj Close Volume company\_name

Date

2013-02-04 2834.750000 2884.949951 2766.659912 2776.909912 2776.909912 11276600 AMAZON

2014-02-04 3112.129883 3224.000000 3012.159912 3152.790039 3152.790039 12640500 AMAZON

2015-02-07 3170.399902 3253.820068 3145.000000 3158.709961 3158.709961 5131200 AMAZON

2016-02-08 3135.010010 3235.850098 3111.010010 3228.270020 3228.270020 3802000 AMAZON

2017-02-09 3257.469971 3276.689941 3205.000000 3223.790039 3223.790039 3439300 AMAZON

2018-02-10 3167.000000 3214.330078 3155.000000 3180.070068 3180.070068 3413400 AMAZON

2019-02-11 3162.689941 3180.000000 3054.679932 3065.870117 3065.870117 3851600 AMAZON

2020-02-14 3035.020020 3168.969971 3033.000000 3103.340088 3103.340088 4022000 AMAZON

2021-02-15 3152.110107 3160.139893 3092.750000 3130.209961 3130.209961 2822000 AMAZON

2022-02-16 3115.810059 3161.645020 3095.000000 3155.254883 3155.254883 1675095 AMAZON

**Summary Stats**

AAPL.describe()

Open High Low Close Adj Close Volume

count 255.000000 255.000000 255.000000 255.000000 255.000000 2.550000e+02

mean 145.436275 146.975765 144.006274 145.553078 145.091826 8.954935e+07

std 17.254608 17.564324 17.052825 17.333592 17.513799 2.849016e+07

min 119.029999 120.400002 116.209999 116.360001 115.672493 3.734861e+07

25% 130.090004 131.074997 129.005005 130.285004 129.630661 6.907255e+07

50% 145.529999 146.949997 144.580002 145.860001 145.452133 8.492240e+07

75% 154.985001 156.574997 153.519997 154.705002 154.282982 1.052436e+08

max 182.630005 182.940002 179.119995 182.009995 181.778397 # General info

AAPL.info()

<class 'pandas.core.frame.DataFrame'>

DatetimeIndex: 255 entries, 2021-02-16 to 2022-02-16

Data columns (total 7 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Open 255 non-null float64

1 High 255 non-null float64

2 Low 255 non-null float64

3 Close 255 non-null float64

4 Adj Close 255 non-null float64

5 Volume 255 non-null int64

6 company\_name 255 non-null object

dtypes: float64(5), int64(1), object(1)

memory usage: 15.9+ KB1.954327e+08plt.figure(figsize=(15, 6))

plt.subplots\_adjust(top=1.25, bottom=1.2)

for i, company in enumerate(company\_list, 1):

plt.subplot(2, 2, i)

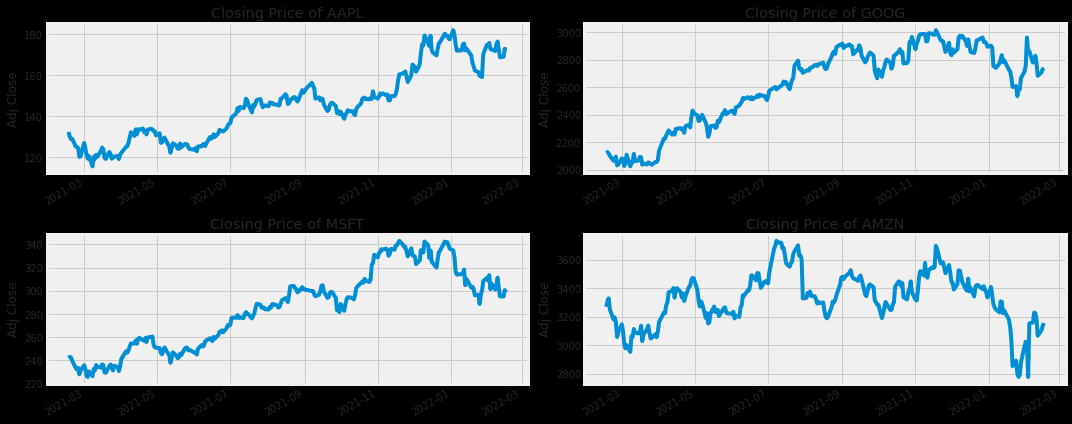
company['Adj Close'].plot()

plt.ylabel('Adj Close')

plt.xlabel(None)

plt.title(f"Closing Price of {tech\_list[i - 1]}")

plt.tight\_layout()



plt.figure(figsize=(15, 7))

plt.subplots\_adjust(top=1.25, bottom=1.2)

for i, company in enumerate(company\_list, 1):

plt.subplot(2, 2, i)

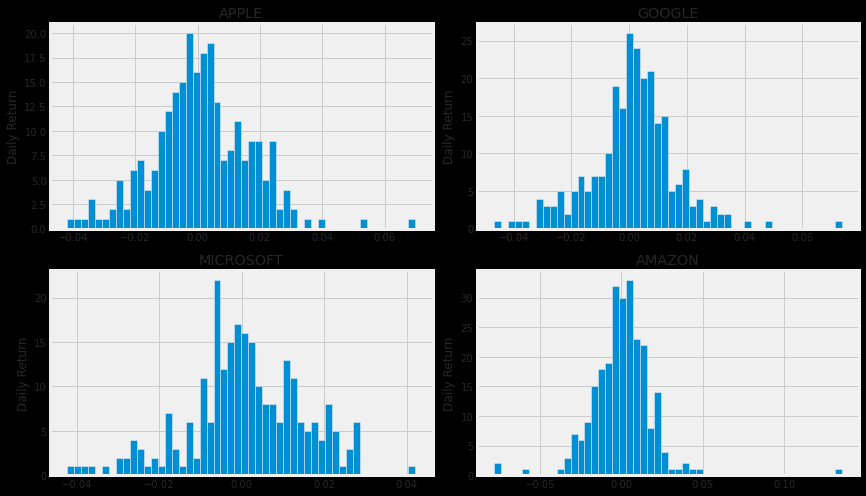
company['Volume'].plot()

plt.ylabel('Volume')

plt.xlabel(None)

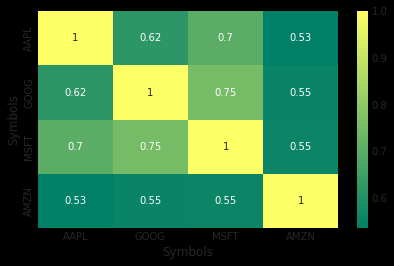
plt.title(f"Sales Volume for {tech\_list[i - 1]}")

plt.tight\_layout()



sns.heatmap(tech\_rets.corr(), annot=True, cmap='summer')

<AxesSubplot:xlabel='Symbols', ylabel='Symbols'>



We start by defining a new DataFrame as a clenaed version of the oriignal tech\_rets DataFrame

rets = tech\_rets.dropna()

area = np.pi \* 20

plt.figure(figsize=(10, 7))

plt.scatter(rets.mean(), rets.std(), s=area)

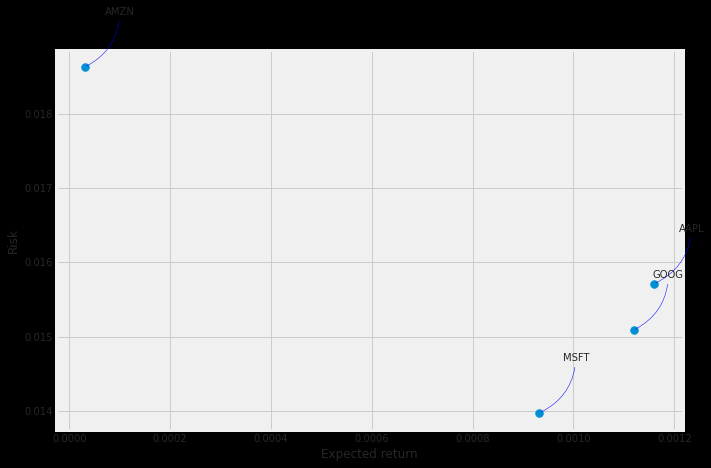
plt.xlabel('Expected return')

plt.ylabel('Risk')

for label, x, y in zip(rets.columns, rets.mean(), rets.std()):

plt.annotate(label, xy=(x, y), xytext=(50, 50), textcoords='offset points', ha='right', va='bottom',

arrowprops=dict(arrowstyle='-', color='blue', connectionstyle='arc3,rad=-0.3'))



**Time Series**

# Import Libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import datetime as dt

import statsmodels.api as sm

from scipy import stats

from pandas.core import datetools

from plotly import tools

import plotly.plotly as py

import plotly.figure\_factory as ff

import plotly.tools as tls

import plotly.graph\_objs as go

from plotly.offline import download\_plotlyjs, init\_notebook\_mode, plot, iplot

init\_notebook\_mode(connected=True)

import warnings

warnings.filterwarnings("ignore")

# plt.style.available

plt.style.use("seaborn-whitegrid")

df = pd.read\_csv('../input/all\_stocks\_5yr.csv')

df.head()

date open high low close volume Name

0 2013-02-08 15.07 15.12 14.63 14.75 8407500 AAL

1 2013-02-11 14.89 15.01 14.26 14.46 8882000 AAL

2 2013-02-12 14.45 14.51 14.10 14.27 8126000 AAL

3 2013-02-13 14.30 14.94 14.25 14.66 10259500 AAL

4 2013-02-14 14.94 14.96 13.16 13.99 31879900 AAL

open high low close volume

count 619029.000000 619032.000000 619032.000000 619040.000000 6.190400e+05

mean 83.023334 83.778311 82.256096 83.043763 4.321823e+06

std 97.378769 98.207519 96.507421 97.389748 8.693610e+06

min 1.620000 1.690000 1.500000 1.590000 0.000000e+00

25% 40.220000 40.620000 39.830000 40.245000 1.070320e+06

50% 62.590000 63.150000 62.020000 62.620000 2.082094e+06

75% 94.370000 95.180000 93.540000 94.410000 4.284509e+06

max 2044.000000 2067.990000 2035.110000 2049.000000 6.182376e+08

# We Replace the column name from name to ticks

df = df.rename(columns={'Name': 'Ticks'})

We analyze some of the stocks.

amzn = df.loc[df['Ticks'] == 'AMZN']

amzn.head()

date open high low close volume Ticks

46387 2013-02-08 261.40 265.25 260.555 261.95 3879078 AMZN

46388 2013-02-11 263.20 263.25 256.600 257.21 3403403 AMZN

46389 2013-02-12 259.19 260.16 257.000 258.70 2938660 AMZN

46390 2013-02-13 261.53 269.96 260.300 269.47 5292996 AMZN

46391 2013-02-14 267.37 270.65 265.400 269.24 3462780 AMZN

amzn.info() # Check whether the date is as object type or date type

<class 'pandas.core.frame.DataFrame'>

Int64Index: 1259 entries, 46387 to 47645

Data columns (total 7 columns):

date 1259 non-null object

open 1259 non-null float64

high 1259 non-null float64

low 1259 non-null float64

close 1259 non-null float64

volume 1259 non-null int64

Ticks 1259 non-null object

dtypes: float64(4), int64(1), object(2)

memory usage: 78.7+ KB

# Change to dateformat

amzn.head()

date open high low close volume Ticks

46387 2013-02-08 261.40 265.25 260.555 261.95 3879078 AMZN

46388 2013-02-11 263.20 263.25 256.600 257.21 3403403 AMZN

46389 2013-02-12 259.19 260.16 257.000 258.70 2938660 AMZN

46390 2013-02-13 261.53 269.96 260.300 269.47 5292996 AMZN

46391 2013-02-14 267.37 270.65 265.400 269.24 3462780 AMZN

Create a copy to avoid the SettingWarning .loc issue

amzn\_df = amzn.copy()

# Change to datetime datatype.

amzn\_df.loc[:, 'date'] = pd.to\_datetime(amzn.loc[:,'date'], format="%Y/%m/%d")

amzn\_df.info()

<class 'pandas.core.frame.DataFrame'>

Int64Index: 1259 entries, 46387 to 47645

Data columns (total 7 columns):

date 1259 non-null datetime64[ns]

open 1259 non-null float64

high 1259 non-null float64

low 1259 non-null float64

close 1259 non-null float64

volume 1259 non-null int64

Ticks 1259 non-null object

dtypes: datetime64[ns](1), float64(4), int64(1), object(1)

memory usage: 78.7+ KB

# Simple plot of Amazon Stock Price

# First Subplot

f, (ax1, ax2) = plt.subplots(1, 2, figsize=(14,5))

ax1.plot(amzn\_df["date"], amzn\_df["close"])

ax1.set\_xlabel("Date", fontsize=12)

ax1.set\_ylabel("Stock Price")

ax1.set\_title("Amazon Close Price History")

# Second Subplot

ax1.plot(amzn\_df["date"], amzn\_df["high"], color="green")

ax1.set\_xlabel("Date", fontsize=12)

ax1.set\_ylabel("Stock Price")

ax1.set\_title("Amazon High Price History")

# Third Subplot

ax1.plot(amzn\_df["date"], amzn\_df["low"], color="red")

ax1.set\_xlabel("Date", fontsize=12)

ax1.set\_ylabel("Stock Price")

ax1.set\_title("Amazon Low Price History")

# Fourth Subplot

ax2.plot(amzn\_df["date"], amzn\_df["volume"], color="orange")

ax2.set\_xlabel("Date", fontsize=12)

ax2.set\_ylabel("Stock Price")

ax2.set\_title("Amazon's Volume History")

plt.show()

