FAMA FRENCH 3 Factor Model

In [7]:

```
import pandas as pd
import statsmodels.api as sm
import datetime as dt
import matplotlib.pyplot as plt
import numpy as np
from statsmodels.tsa.stattools import adfuller
from scipy import stats
from math import sqrt
```

In [8]:

```
pip install getFamaFrenchFactors
```

```
Collecting getFamaFrenchFactors
  Downloading getFamaFrenchFactors-0.0.5-py3-none-any.whl (4.6 kB)
Collecting bs4
  Downloading bs4-0.0.1.tar.gz (1.1 kB)
Requirement already satisfied: pandas in c:\users\kanika\anaconda3\lib\site-packages (from getFamaFr
enchFactors) (1.4.2)
Requirement already satisfied: requests in c:\users\kanika\anaconda3\lib\site-packages (from getFama
FrenchFactors) (2.27.1)
Requirement already satisfied: beautifulsoup4 in c:\users\kanika\anaconda3\lib\site-packages (from b
s4->getFamaFrenchFactors) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\kanika\anaconda3\lib\site-packages (from be
autifulsoup4->bs4->getFamaFrenchFactors) (2.3.1)
Requirement already satisfied: pytz>=2020.1 in c:\users\kanika\anaconda3\lib\site-packages (from pan
das->getFamaFrenchFactors) (2022.7.1)
Requirement already satisfied: numpy>=1.18.5 in c:\users\kanika\anaconda3\lib\site-packages (from pa
ndas->getFamaFrenchFactors) (1.21.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\kanika\anaconda3\lib\site-packages
(from pandas->getFamaFrenchFactors) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\kanika\anaconda3\lib\site-packages (from python-
dateutil>=2.8.1->pandas->getFamaFrenchFactors) (1.16.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\kanika\anaconda3\lib\site-packages
(from requests->getFamaFrenchFactors) (1.26.9)
Requirement already satisfied: idna<4,>=2.5 in c:\users\kanika\anaconda3\lib\site-packages (from req
uests->getFamaFrenchFactors) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kanika\anaconda3\lib\site-packages (fr
om requests->getFamaFrenchFactors) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\kanika\anaconda3\lib\site-packa
ges (from requests->getFamaFrenchFactors) (2.0.4)
Building wheels for collected packages: bs4
  Building wheel for bs4 (setup.py): started
  Building wheel for bs4 (setup.py): finished with status 'done'
  Created wheel for bs4: filename=bs4-0.0.1-py3-none-any.whl size=1272 sha256=033db04ad7019544e3b5f2
d53a70ddc2b5b41491f9dd789067a42265547791eb
  Stored in directory: c:\users\kanika\appdata\local\pip\cache\wheels\73\2b\cb\099980278a0c9a3e57ff1
a89875ec07bfa0b6fcbebb9a8cad3
Successfully built bs4
Installing collected packages: bs4, getFamaFrenchFactors
Successfully installed bs4-0.0.1 getFamaFrenchFactors-0.0.5
Note: you may need to restart the kernel to use updated packages.
```

In [9]:

```
import getFamaFrenchFactors as fff
```

In [20]:

```
data = pd.read_csv('FF.csv')
data.set_index('Date',inplace=True)
data.head()
```

Out[20]:

Date	AAPL	AMZN	GOOG	META	NFLX
02-01-2013	16.837122	12.8655	18.013729	28.000000	13.144286
03-01-2013	16.624598	12.9240	18.024191	27.770000	13.798571
04-01-2013	16.161522	12.9575	18.380356	28.760000	13.711429
07-01-2013	16.066454	13.4230	18.300158	29.420000	14.171429
08-01-2013	16.109695	13.3190	18.264042	29.059999	13.880000

In [104]:

```
data.describe()
```

Out[104]:

	AAPL	AMZN	GOOG	META	NFLX
count	2518.000000	2518.000000	2518.000000	2518.000000	2518.000000
mean	60.544588	73.780049	59.315665	155.583022	239.053170
std	49.173579	53.289558	35.169357	83.901712	174.233526
min	12.046192	12.411500	17.506132	22.900000	13.144286
25%	24.639015	21.978375	29.984361	86.730000	89.404287
50%	39.451430	59.735750	51.419500	151.220002	190.705002
75%	89.985527	107.755125	73.789252	194.432502	361.797501
max	180.683868	186.570496	150.709000	382.179993	691.690002

In [21]:

```
# check if there are any null values

data.isnull().sum()
```

Out[21]:

```
AAPL    0
AMZN    0
GOOG    0
META    0
NFLX    0
dtype: int64
```

In [22]:

```
# we calculate daily returns from the data on adjusted closing prices

apple_returns = data['AAPL'].pct_change().dropna()
amazon_returns = data['AMZN'].pct_change().dropna()
google_returns = data['GOOG'].pct_change().dropna()
meta_returns = data['META'].pct_change().dropna()
netflix_returns = data['NFLX'].pct_change().dropna()
```

In [33]:

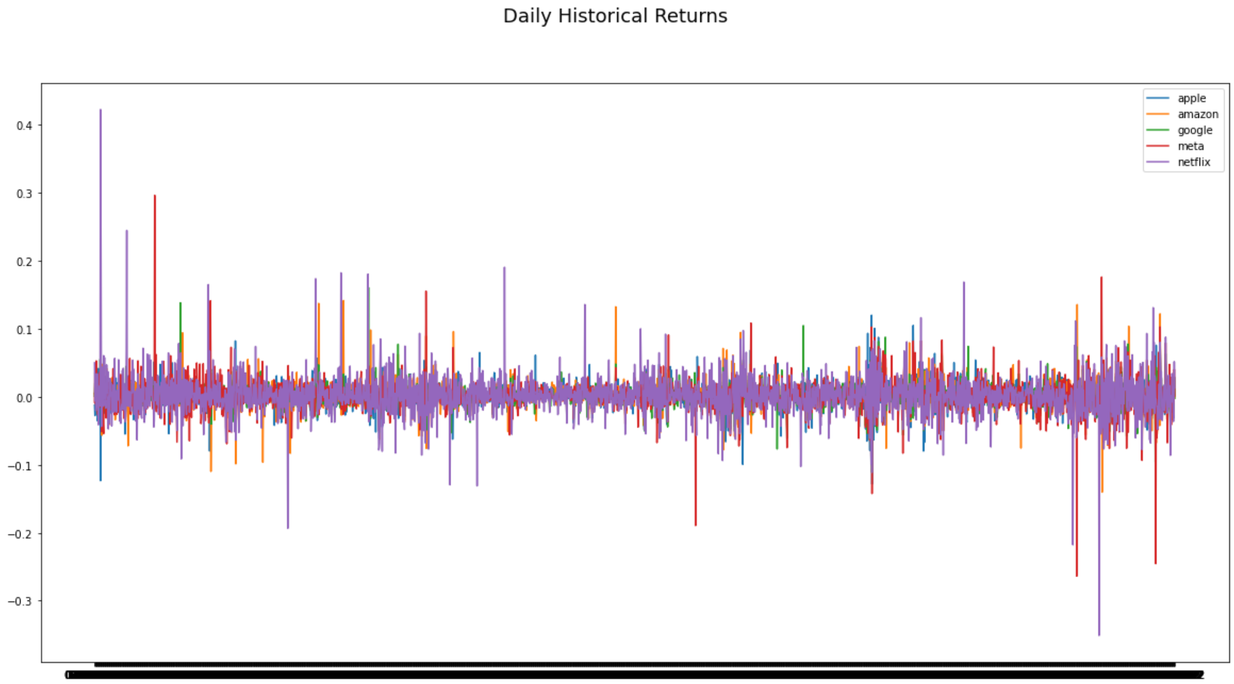
```
# plotting the daily returns of all the 5 stocks

plt.figure()
fig, ax = plt.subplots(figsize=(20, 10))
ax.plot(apple_returns,label='apple')
ax.plot(amazon_returns,label='amazon')
ax.plot(google_returns,label='google')
ax.plot(meta_returns,label='meta')
ax.plot(netflix_returns,label='netflix')
ax.legend()
fig.suptitle('Daily Historical Returns',fontsize=18)
```

Out[33]:

```
Text(0.5, 0.98, 'Daily Historical Returns')
```

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



In [34]:

```
# since this is a time series data, we check for stationarity using the augmented dickey fuller test

adf = adfuller(apple_returns)
print('ADF Statistic: %f' % adf[0])
print('p-value: %f' % adf[1])
```

```
ADF Statistic: -16.320532
p-value: 0.000000
```

In [35]:

```
adf = adfuller(amazon_returns)
print('ADF Statistic: %f' % adf[0])
print('p-value: %f' % adf[1])
```

```
ADF Statistic: -50.811947
p-value: 0.000000
```

In [36]:

```
adf = adfuller(google_returns)
print('ADF Statistic: %f' % adf[0])
print('p-value: %f' % adf[1])
```

```
ADF Statistic: -11.371348
p-value: 0.000000
```

In [37]:

```
adf = adfuller(meta_returns)
print('ADF Statistic: %f' % adf[0])
print('p-value: %f' % adf[1])
```

```
ADF Statistic: -17.336024
p-value: 0.000000
```

In [38]:

```
adf = adfuller(netflix_returns)
print('ADF Statistic: %f' % adf[0])
print('p-value: %f' % adf[1])
```

```
ADF Statistic: -34.033968
p-value: 0.000000
```

In [ ]:

```
# the p-value for all the 5 series is less than 0.05. Therefore, we reject the null hypothesis at 5% level of
# significance and conclude that the data is stationary across time
```

In [65]:

```
# we get the fama-french estimates for the 3 factors - Mkt-RF: Market return - risk free rate,
# SMB: excess return on small cap stocks over large cap stcoks
# HML: excess return on value stock over growth stocks

ff_monthly = fff.famaFrench3Factor(frequency='m')
ff_monthly.rename(columns={"date_ff_factors": 'Date'}, inplace=True)
ff_monthly.set_index('Date', inplace=True)
ff_monthly.index = ff_monthly.index.strftime('%d-%m-%Y')
ff_monthly.head()
```

Out[65]:

	Mkt-RF	SMB	HML	RF
Date				
31-07-1926	0.0296	-0.0256	-0.0243	0.0022
31-08-1926	0.0264	-0.0117	0.0382	0.0025
30-09-1926	0.0036	-0.0140	0.0013	0.0023
31-10-1926	-0.0324	-0.0009	0.0070	0.0032
30-11-1926	0.0253	-0.0010	-0.0051	0.0031

In [66]:

```
# we merge the two dataframes on fama french estimate and returns to get a new dataframe

ff = data.merge(ff_monthly, left_index=True, right_index=True)
ff.head()
```

Out[66]:

	AAPL	AMZN	GOOG	META	NFLX	Mkt-RF	SMB	HML	RF
Date									
31-01-2013	13.968526	13.2750	18.821701	30.980000	23.605715	0.0557	0.0033	0.0096	0.0
28-02-2013	13.615317	13.2135	19.955202	27.250000	26.868570	0.0129	-0.0028	0.0011	0.0
30-04-2013	13.657884	12.6905	20.537271	27.770000	30.867144	0.0155	-0.0236	0.0045	0.0
31-05-2013	13.964083	13.4600	21.699165	24.350000	32.321430	0.0280	0.0173	0.0263	0.0
31-07-2013	14.051023	15.0610	22.110872	36.799999	34.925713	0.0565	0.0186	0.0057	0.0

In [71]:

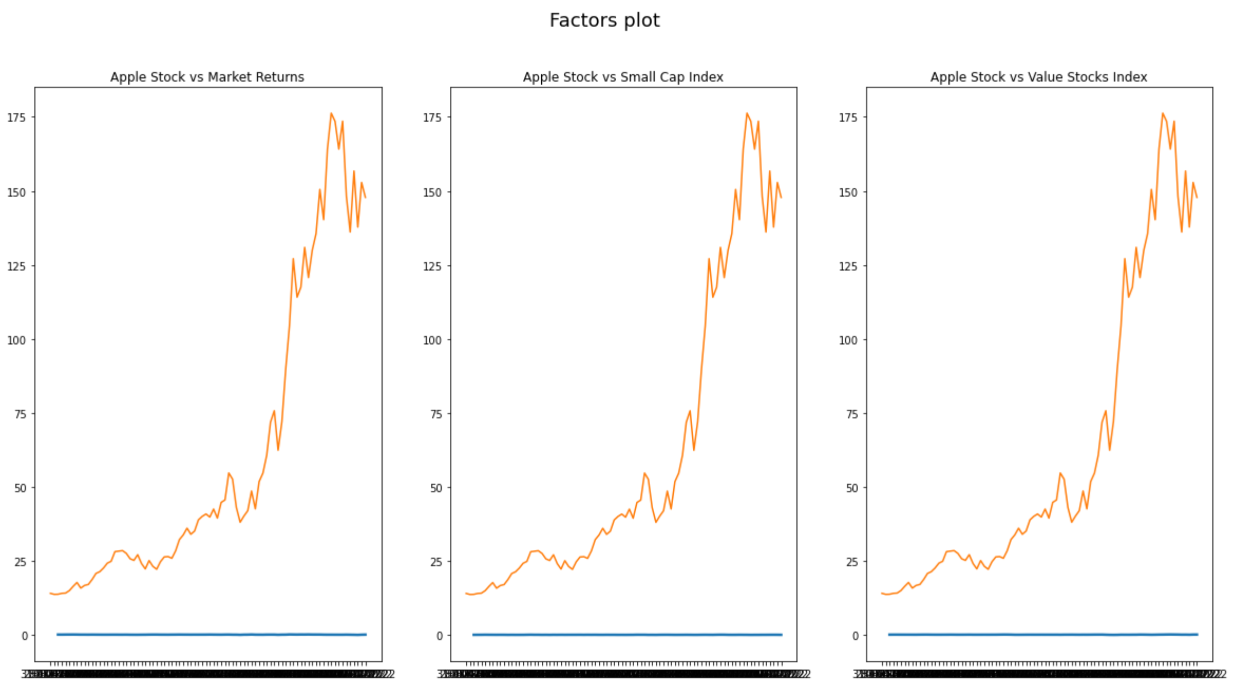
```
# visualize the moving average of the fama-french model for all the five stocks

stock = 'AAPL'
plt.figure()
fig3, axs = plt.subplots(1, 3,figsize=(20, 10))
axs[0].plot(ff['Mkt-RF'].rolling(3).mean(),linewidth=2.5)
axs[0].plot(ff[stock])
axs[0].set_title('Apple Stock vs Market Returns')
axs[1].plot(ff['SMB'].rolling(3).mean(),linewidth=2.5)
axs[1].plot(ff[stock])
axs[1].set_title('Apple Stock vs Small Cap Index')
axs[2].plot(ff['HML'].rolling(3).mean(),linewidth=2.5)
axs[2].plot(ff[stock])
axs[2].set_title('Apple Stock vs Value Stocks Index')
fig3.suptitle('Factors plot',fontsize=18)
```

Out[71]:

```
Text(0.5, 0.98, 'Factors plot')
```

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



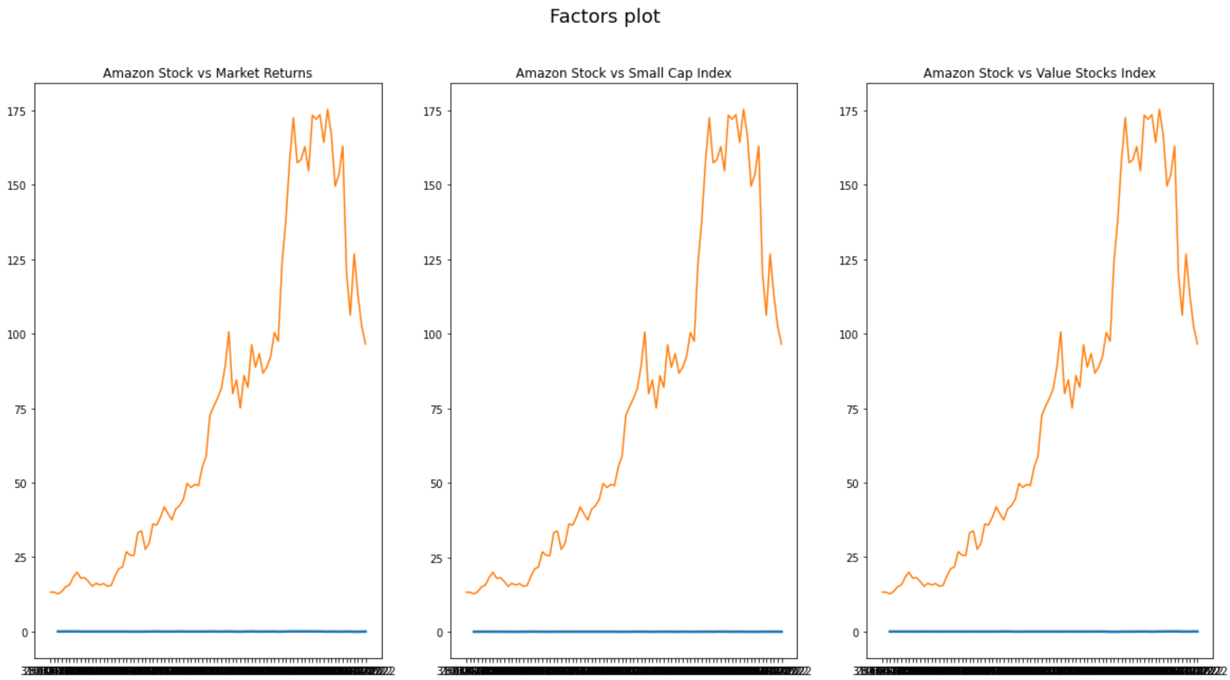
In [72]:

```
stock = 'AMZN'
plt.figure()
fig3, axs = plt.subplots(1, 3,figsize=(20, 10))
axs[0].plot(ff['Mkt-RF'].rolling(3).mean(),linewidth=2.5)
axs[0].plot(ff[stock])
axs[0].set_title('Amazon Stock vs Market Returns')
axs[1].plot(ff['SMB'].rolling(3).mean(),linewidth=2.5)
axs[1].plot(ff[stock])
axs[1].set_title('Amazon Stock vs Small Cap Index')
axs[2].plot(ff['HML'].rolling(3).mean(),linewidth=2.5)
axs[2].plot(ff[stock])
axs[2].set_title('Amazon Stock vs Value Stocks Index')
fig3.suptitle('Factors plot',fontsize=18)
```

Out[72]:

```
Text(0.5, 0.98, 'Factors plot')
```

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



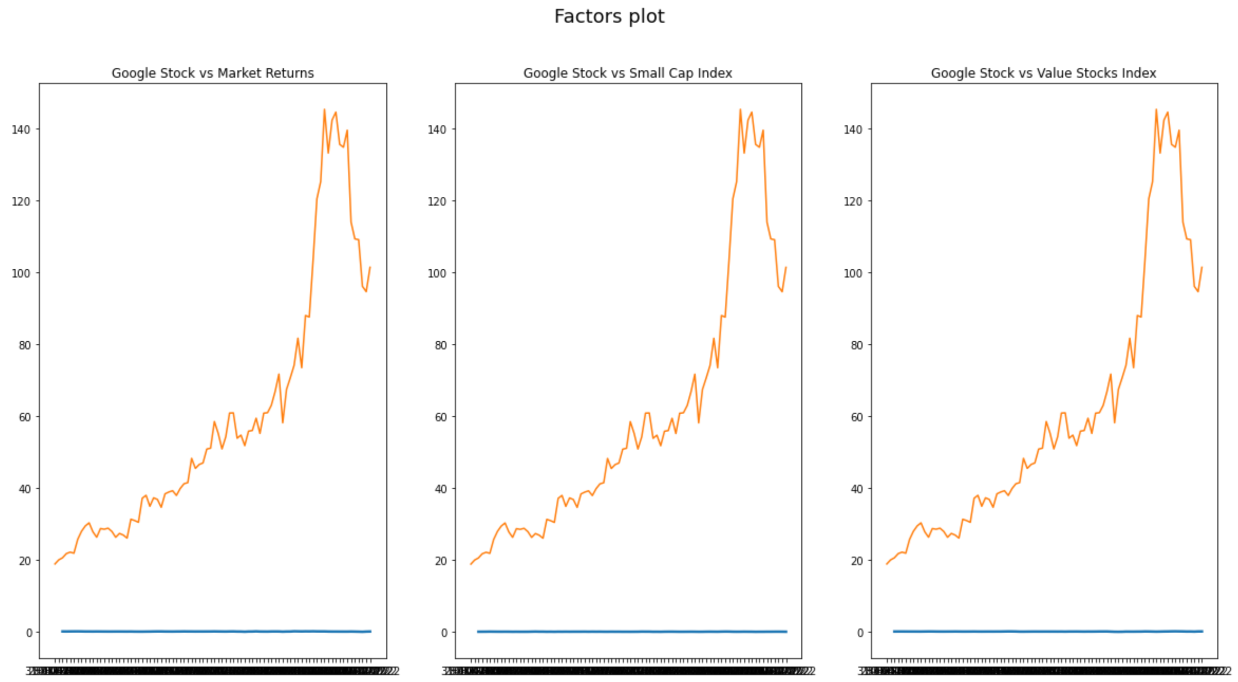
In [73]:

```
stock = 'GOOG'
plt.figure()
fig3, axs = plt.subplots(1, 3,figsize=(20, 10))
axs[0].plot(ff['Mkt-RF'].rolling(3).mean(),linewidth=2.5)
axs[0].plot(ff[stock])
axs[0].set_title('Google Stock vs Market Returns')
axs[1].plot(ff['SMB'].rolling(3).mean(),linewidth=2.5)
axs[1].plot(ff[stock])
axs[1].set_title('Google Stock vs Small Cap Index')
axs[2].plot(ff['HML'].rolling(3).mean(),linewidth=2.5)
axs[2].plot(ff[stock])
axs[2].set_title('Google Stock vs Value Stocks Index')
fig3.suptitle('Factors plot',fontsize=18)
```

Out[73]:

```
Text(0.5, 0.98, 'Factors plot')
```

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



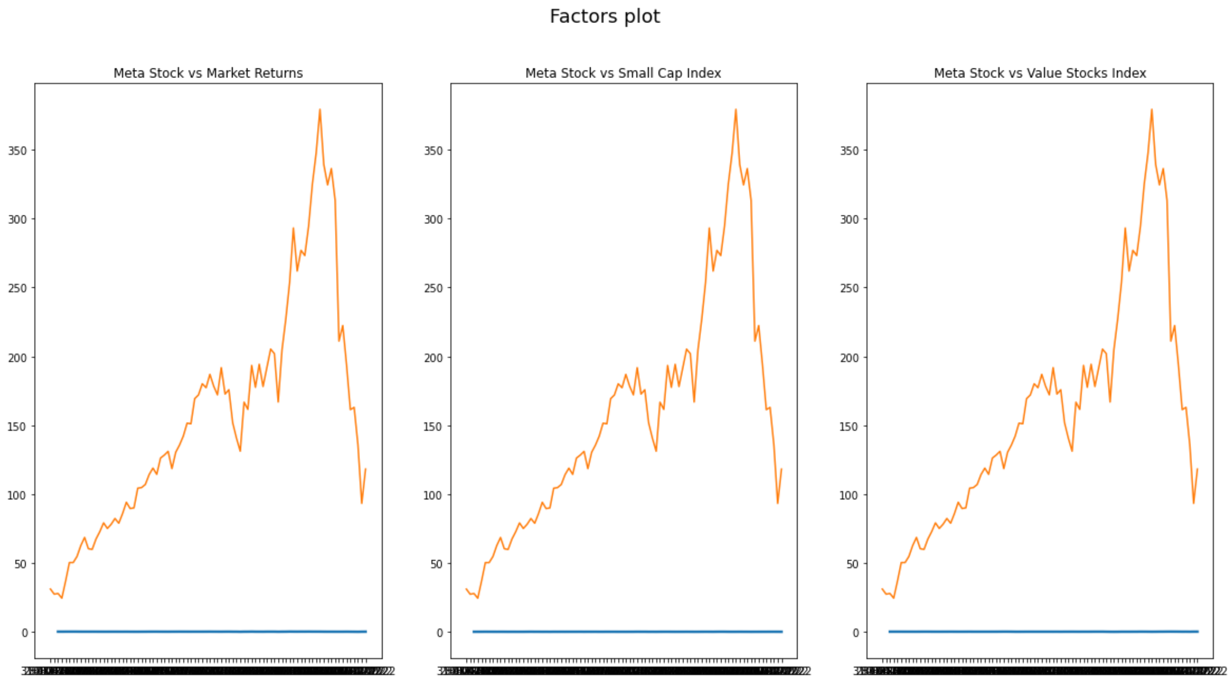
In [74]:

```
stock = 'META'
plt.figure()
fig3, axs = plt.subplots(1, 3,figsize=(20, 10))
axs[0].plot(ff['Mkt-RF'].rolling(3).mean(),linewidth=2.5)
axs[0].plot(ff[stock])
axs[0].set_title('Meta Stock vs Market Returns')
axs[1].plot(ff['SMB'].rolling(3).mean(),linewidth=2.5)
axs[1].plot(ff[stock])
axs[1].set_title('Meta Stock vs Small Cap Index')
axs[2].plot(ff['HML'].rolling(3).mean(),linewidth=2.5)
axs[2].plot(ff[stock])
axs[2].set_title('Meta Stock vs Value Stocks Index')
fig3.suptitle('Factors plot',fontsize=18)
```

Out[74]:

```
Text(0.5, 0.98, 'Factors plot')

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



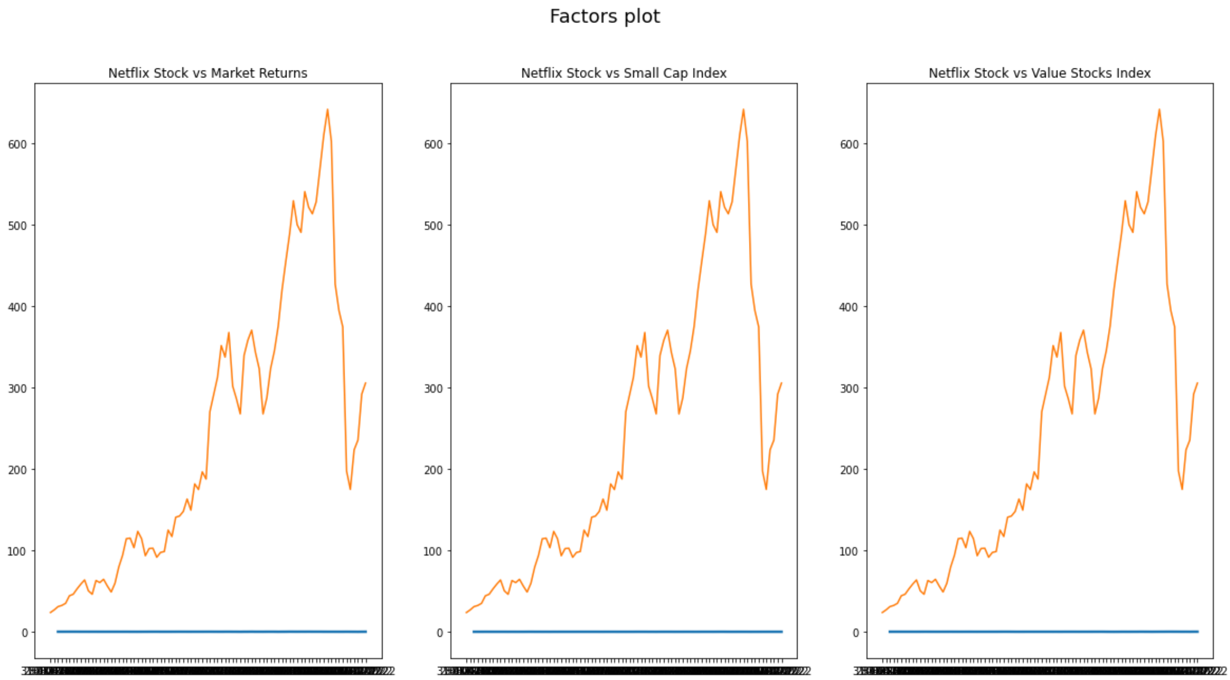
In [75]:

```
stock = 'NFLX'
plt.figure()
fig3, axs = plt.subplots(1, 3,figsize=(20, 10))
axs[0].plot(ff['Mkt-RF'].rolling(3).mean(),linewidth=2.5)
axs[0].plot(ff[stock])
axs[0].set_title('Netflix Stock vs Market Returns')
axs[1].plot(ff['SMB'].rolling(3).mean(),linewidth=2.5)
axs[1].plot(ff[stock])
axs[1].set_title('Netflix Stock vs Small Cap Index')
axs[2].plot(ff['HML'].rolling(3).mean(),linewidth=2.5)
axs[2].plot(ff[stock])
axs[2].set_title('Netflix Stock vs Value Stocks Index')
fig3.suptitle('Factors plot',fontsize=18)
```

Out[75]:

```
Text(0.5, 0.98, 'Factors plot')
```

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



In [88]:

```
# modeling the returns using a simple linear regression model for all the 5 stocks

X = ff[['Mkt-RF', 'SMB', 'HML']]
y = ff['AAPL'] - ff['RF']
X = sm.add_constant(X)
ff_model = sm.OLS(y, X).fit()
print(ff_model.summary())
intercept, b1, b2, b3 = ff_model.params
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.031
Model:                            OLS   Adj. R-squared:                 -0.006
Method:                 Least Squares   F-statistic:                    0.8480
Date:                Sat, 04 Mar 2023   Prob (F-statistic):              0.472
Time:                        19:27:27   Log-Likelihood:                -447.13
No. Observations:                  84   AIC:                             902.3
Df Residuals:                      80   BIC:                             912.0
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         61.3847      5.768     10.643      0.000      49.906      72.863
Mkt-RF       -41.0241    133.236     -0.308      0.759    -306.173     224.124
SMB          -66.2558    229.650     -0.289      0.774    -523.273     390.762
HML          234.1823    154.507      1.516      0.134     -73.297     541.662
==============================================================================
Omnibus:                       12.704   Durbin-Watson:                   0.088
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               13.474
Skew:                           0.932   Prob(JB):                      0.00119
Kurtosis:                       2.385   Cond. No.                         42.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [89]:

```
X = ff[['Mkt-RF', 'SMB', 'HML']]
y = ff['AMZN'] - ff['RF']
X = sm.add_constant(X)
ff_model = sm.OLS(y, X).fit()
print(ff_model.summary())
intercept, b1, b2, b3 = ff_model.params
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.006
Model:                            OLS   Adj. R-squared:                 -0.032
Method:                 Least Squares   F-statistic:                    0.1537
Date:                Sat, 04 Mar 2023   Prob (F-statistic):              0.927
Time:                        19:28:00   Log-Likelihood:                -452.42
No. Observations:                  84   AIC:                             912.8
Df Residuals:                      80   BIC:                             922.6
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         73.0711      6.143     11.895      0.000      60.846      85.296
Mkt-RF        91.2738    141.898      0.643      0.522    -191.113     373.661
SMB          -85.9288    244.580     -0.351      0.726    -572.659     400.801
HML            9.6336    164.553      0.059      0.953    -317.837     337.104
==============================================================================
Omnibus:                       16.917   Durbin-Watson:                   0.030
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                7.675
Skew:                           0.529   Prob(JB):                       0.0215
Kurtosis:                       1.964   Cond. No.                         42.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [90]:

```
X = ff[['Mkt-RF', 'SMB', 'HML']]
y = ff['GOOG'] - ff['RF']
X = sm.add_constant(X)
ff_model = sm.OLS(y, X).fit()
print(ff_model.summary())
intercept, b1, b2, b3 = ff_model.params
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.028
Model:                            OLS   Adj. R-squared:                 -0.009
Method:                 Least Squares   F-statistic:                    0.7569
Date:                Sat, 04 Mar 2023   Prob (F-statistic):              0.522
Time:                        19:28:13   Log-Likelihood:                -416.61
No. Observations:                  84   AIC:                             841.2
Df Residuals:                      80   BIC:                             850.9
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         59.4713      4.011     14.828      0.000      51.490      67.453
Mkt-RF       -28.8491     92.648     -0.311      0.756    -213.224     155.526
SMB          -84.2170    159.691     -0.527      0.599    -402.011     233.577
HML          140.6423    107.439      1.309      0.194     -73.168     354.453
==============================================================================
Omnibus:                       12.017   Durbin-Watson:                   0.098
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               13.559
Skew:                           0.984   Prob(JB):                      0.00114
Kurtosis:                       3.029   Cond. No.                         42.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [91]:

```
X = ff[['Mkt-RF', 'SMB', 'HML']]
y = ff['META'] - ff['RF']
X = sm.add_constant(X)
ff_model = sm.OLS(y, X).fit()
print(ff_model.summary())
intercept, b1, b2, b3 = ff_model.params
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.009
Model:                            OLS   Adj. R-squared:                 -0.028
Method:                 Least Squares   F-statistic:                    0.2526
Date:                Sat, 04 Mar 2023   Prob (F-statistic):              0.859
Time:                        19:28:21   Log-Likelihood:                -490.60
No. Observations:                  84   AIC:                             989.2
Df Residuals:                      80   BIC:                             998.9
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        153.7429      9.678     15.886      0.000     134.484     173.002
Mkt-RF       176.6623    223.551      0.790      0.432    -268.218     621.543
SMB         -236.1256    385.319     -0.613      0.542   -1002.934     530.683
HML          -19.6306    259.241     -0.076      0.940    -535.537     496.275
==============================================================================
Omnibus:                        5.730   Durbin-Watson:                   0.057
Prob(Omnibus):                  0.057   Jarque-Bera (JB):                5.623
Skew:                           0.633   Prob(JB):                       0.0601
Kurtosis:                       2.954   Cond. No.                         42.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [92]:

```
X = ff[['Mkt-RF', 'SMB', 'HML']]
y = ff['NFLX'] - ff['RF']
X = sm.add_constant(X)
ff_model = sm.OLS(y, X).fit()
print(ff_model.summary())
intercept, b1, b2, b3 = ff_model.params
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.015
Model:                            OLS   Adj. R-squared:                 -0.022
Method:                 Least Squares   F-statistic:                    0.4133
Date:                Sat, 04 Mar 2023   Prob (F-statistic):              0.744
Time:                        19:28:27   Log-Likelihood:                -550.53
No. Observations:                  84   AIC:                             1109.
Df Residuals:                      80   BIC:                             1119.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        233.2206     19.752     11.807      0.000     193.912     272.529
Mkt-RF       480.1694    456.275      1.052      0.296    -427.847    1388.186
SMB         -453.3631    786.449     -0.576      0.566   -2018.446    1111.720
HML         -245.8241    529.120     -0.465      0.643   -1298.806     807.158
==============================================================================
Omnibus:                        7.441   Durbin-Watson:                   0.055
Prob(Omnibus):                  0.024   Jarque-Bera (JB):                6.286
Skew:                           0.576   Prob(JB):                       0.0432
Kurtosis:                       2.316   Cond. No.                         42.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [ ]:

```
# it is notable that the p-values for all the three factors across all the stocks is much higher than 0.05.
# this casts a strong doubt on the significance of the three factors in explaining the returns of the FAANG stocks
```

In [94]:

```
# calculate the average of all the factors

rf = fama_french['RF'].mean()
market_premium = ff_monthly['Mkt-RF'].mean()
size_premium = ff_monthly['SMB'].mean()
value_premium = ff_monthly['HML'].mean()
```

In [101]:

```
market_premium
```

Out[101]:

```
0.006729076790336501
```

In [102]:

```
size_premium
```

Out[102]:

```
0.0019342536669542728
```

In [103]:

```
value_premium
```

Out[103]:

```
0.003568852459016399
```