

In [1]:

```
import yfinance as yf
import numpy as np
import pandas as pd
import scipy
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
import sklearn
```

In [19]:

```
data = yf.download('NFLX ^GSPC', start='2017-05-22', end= '2022-05-02')
data
#scarico i dati degli ultimi 5 anni '2017-05-22', end= '2022-05-02' riguardanti le azioni
di Netflix e l'indice S&P500
```

[*****100%*****] 2 of 2 completed

Out[19]:

	Adj Close		Close		High		Low		Open	
	NFLX	^GSPC	NFLX	^GSPC	NFLX	^GSPC	NFLX	^GSPC	NFLX	^GSPC
Date										
2017-05-22	157.160004	2394.020020	157.160004	2394.020020	158.600006	2395.459961	156.429993	2386.919922	157.860001	2387
2017-05-23	157.949997	2398.419922	157.949997	2398.419922	158.309998	2400.850098	156.800003	2393.879883	157.750000	2397
2017-05-24	157.750000	2404.389893	157.750000	2404.389893	158.479996	2405.580078	157.169998	2397.989990	158.350006	2401
2017-05-25	163.050003	2415.070068	163.050003	2415.070068	164.100006	2418.709961	160.550003	2408.010010	161.000000	2409
2017-05-26	162.429993	2415.820068	162.429993	2415.820068	163.050003	2416.679932	161.119995	2412.199951	162.839996	2414
...
2022-04-25	209.910004	4296.120117	209.910004	4296.120117	216.660004	4299.020020	204.509995	4200.819824	213.809998	4255
2022-04-26	198.399994	4175.200195	198.399994	4175.200195	209.149994	4278.140137	198.279999	4175.040039	208.800003	4278
2022-04-27	188.539993	4183.959961	188.539993	4183.959961	200.190002	4240.709961	187.770004	4162.899902	194.009995	4186
2022-04-28	199.520004	4287.500000	199.520004	4287.500000	201.479996	4308.450195	185.600006	4188.629883	191.559998	4222
2022-04-29	190.360001	4131.930176	190.360001	4131.930176	204.710007	4269.680176	190.000000	4124.279785	198.679993	4253

1245 rows x 12 columns

In [20]:

```
adj_close = data['Adj Close']
adj_close
#prendo in considerazione solamente la chiusura aggiustata, ovvero il prezzo aumentato per i dividendi
```

Out[20]:

	NFLX	^GSPC
Date		
2017-05-22	157.160004	2394.020020
2017-05-23	157.949997	2398.419922
2017-05-24	157.750000	2404.389893
2017-05-25	163.050003	2415.070068
2017-05-26	162.429993	2415.820068
...
2022-04-25	209.910004	4296.120117
2022-04-26	198.399994	4175.200195
2022-04-27	188.539993	4183.959961
2022-04-28	199.520004	4287.500000
2022-04-29	190.360001	4131.930176

1245 rows × 2 columns

In [21]:

```
returns = adj_close.pct_change()
returns = returns.dropna(axis=0) #così dropa la prima colonna che ha i valori Nan, perché è il valore di avvio quindi nullo
returns
#calcolo i ritorni sul prezzo aggiustato, calcolato come Ritorno = (Pt+1 + DIV -P)/P nell'arco di tempo stabilito di 5 anni.
```

Out[21]:

	NFLX	^GSPC
Date		
2017-05-23	0.005027	0.001838
2017-05-24	-0.001266	0.002489
2017-05-25	0.033597	0.004442
2017-05-26	-0.003803	0.000311
2017-05-30	0.004864	-0.001205
...
2022-04-25	-0.026030	0.005698
2022-04-26	-0.054833	-0.028146
2022-04-27	-0.049698	0.002098
2022-04-28	0.058237	0.024747
2022-04-29	-0.045910	-0.036285

1244 rows × 2 columns

In [22]:

```
returns.describe()
#mi dà i principali indicatori sui ritorni per colonna.
```

Out[22]:

	NFLX	^GSPC
count	1244.000000	1244.000000
mean	0.000561	0.000519
std	0.027918	0.012654

min	NFLX	GSPC
25%	-0.012127	-0.003537
50%	0.000467	0.000938
75%	0.014374	0.006336
max	0.168543	0.093828

In [23]:

```
y = returns['NFLX'] #variabile dipendente
x1 = returns['^GSPC'] #variabile indipendente

y, x1
```

Out[23]:

```
(Date
2017-05-23    0.005027
2017-05-24   -0.001266
2017-05-25    0.033597
2017-05-26   -0.003803
2017-05-30    0.004864
...
2022-04-25   -0.026030
2022-04-26   -0.054833
2022-04-27   -0.049698
2022-04-28    0.058237
2022-04-29   -0.045910
Name: NFLX, Length: 1244, dtype: float64,
Date
2017-05-23    0.001838
2017-05-24    0.002489
2017-05-25    0.004442
2017-05-26    0.000311
2017-05-30   -0.001205
...
2022-04-25    0.005698
2022-04-26   -0.028146
2022-04-27    0.002098
2022-04-28    0.024747
2022-04-29   -0.036285
Name: ^GSPC, Length: 1244, dtype: float64)
```

In [24]:

```
X = sm.add_constant(x1)
```

```
C:\Users\errea\anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:142: FutureWarning
: In a future version of pandas all arguments of concat except for the argument 'objs' wi
ll be keyword-only
  x = pd.concat(x[:, :order], 1)
```

In [25]:

```
model = sm.OLS(y, X)
model
#uso come modello l'Ordinary least squares
```

Out[25]:

```
<statsmodels.regression.linear_model.OLS at 0x15c231d4eb0>
```

In [26]:

```
results = sm.OLS(y, X).fit()
results.summary()
#La retta di regressione che minimizza la distanza tra i punti dei ritoni dell'indice S&P
500 (ovvero l'indice di mercato
#preso come riferimento) mi dà un coefficiente angolare m di 1.0299.
#Dunque il mio Beta per Netflix sarà 1.0299, che mi indica la rischiosità che aggiunge al
```

```

mio portafoglio di mercato
#che possiedo secondo le assunzioni del modello di Capital Asset Pricing Model. Dunque ch
iederò un rendimento maggiore
#rispetto a quello di mercato di 1.0299, questo perché quando il mercato è sceso di 1 med
iamente l'azione Netflix
#è scesa di 1.0299.

```

Out [26] :

OLS Regression Results

Dep. Variable:	NFLX	R-squared:	0.218
Model:	OLS	Adj. R-squared:	0.217
Method:	Least Squares	F-statistic:	346.1
Date:	Mon, 23 May 2022	Prob (F-statistic):	2.49e-68
Time:	12:01:38	Log-Likelihood:	2839.8
No. Observations:	1244	AIC:	-5676.
Df Residuals:	1242	BIC:	-5665.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	2.582e-05	0.001	0.037	0.971	-0.001	0.001
^GSPC	1.0299	0.055	18.603	0.000	0.921	1.139

Omnibus:	834.573	Durbin-Watson:	1.979
Prob(Omnibus):	0.000	Jarque-Bera (JB):	78344.140
Skew:	-2.268	Prob(JB):	0.00
Kurtosis:	41.612	Cond. No.	79.1

Notes:

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

In [27] :

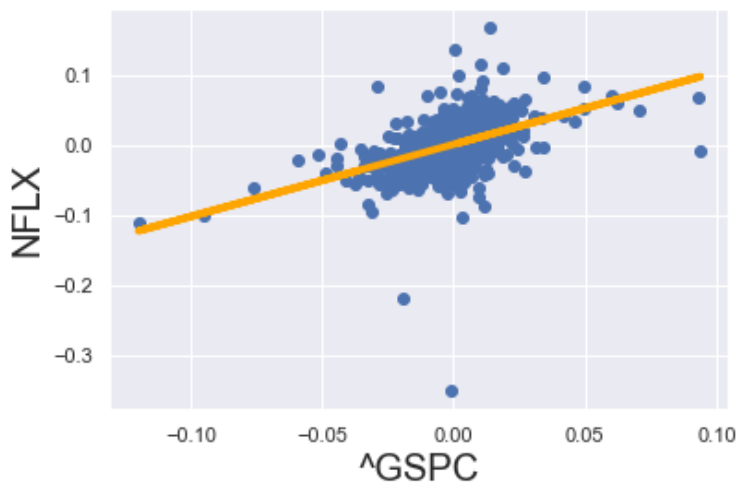
```

plt.scatter(x1,y)
yhat = 1.0299*x1 + 0.0009 #retta di regressione
fig = plt.plot(x1, yhat, lw=4, c='orange', label = 'regression line')
plt.xlabel('^GSPC', fontsize = 20)
plt.ylabel('NFLX', fontsize = 20)

```

Out [27] :

Text(0, 0.5, 'NFLX')



In [1]:

```
#Con un beta > 1, Il rendimento richiesto è più alto in quanto aggiunge rischio al mio portafoglio, cioè quando il mercato varia
#tendenzialmente Netflix oscilla di più.
#Dunque in caso di bisogno, se in quel momento il mercato sta scendendo, significherebbe che il mio portafoglio avrebbe uno
#scossone maggiore, e nel caso mi servisse liquidità per far fronte ad una emergenza potrei avere una liquidità minore
#di quella aspettata.
#Secondo la teoria ottimale del portafoglio:
# Con due investimenti con stesso rendimento medio, l'investitore razionale sceglierebbe sempre quello con std inferiore
#perché l'altro avrebbe una probabilità maggiore
# che il prossimo anno in caso di bisogno di liquidità, il portafoglio sarebbe più volatile e in caso di una caduta del mercato
# il rendimento del portafoglio amplificherebbe in negativo quello del mercato, e dunque in caso di vendita andrei ad ottenere
# una liquidità minore di quella aspettata, anche se il rendimento medio nel lungo termine è lo stesso.
```

In []:

```
#Dunque il coefficiente beta nel modello di CAPM, il beta mi dà lo slope della retta di regressione,
#e quindi mi aiuta a predire come saranno i rendimenti futuri quando l'Sp 500 varia.
```