Assignment 10

In [1]:
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from tiingo import TiingoClient
import numpy as np
from datetime import date
import warnings
warnings.filterwarnings('ignore')
from dateutil.relativedelta import relativedelta
config = {}

config['session'] = True

config['api_key'] = "110ee73e29ec4269f49eb85cfb4b976ab8e73361"

client = TiingoClient(config)
```

In [21]:
```python
def download_financial_data(ticker):
    fin_data = client.get_ticker_price(ticker,
                                       fmt='csv',
                                       startDate = date.today() - relativedelta(years=2),
                                       endDate = date.today(),
                                       frequency = 'daily')
    file_name = f"{ticker}.csv"
    with open(file_name,'w') as outfile:
        outfile.write(fin_data)
    print(f'{ticker}.csv created')
    return pd.read_csv(f"{ticker}.csv")
```

In [22]:
```python
cpi_df = download_financial_data('CPI')
cpi_df.head()
```

```
CPI.csv created
```

Out[22]:

	date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	adjVolume	divCash	splitFactor
0	2020-06-11	27.07	27.09	27.07	27.09	1236	26.458009	26.477557	26.458009	26.477557	1236	0.0	1.0
1	2020-06-12	27.16	27.16	27.12	27.14	736	26.545974	26.545974	26.506879	26.526426	736	0.0	1.0
2	2020-06-15	27.17	27.19	27.17	27.19	313	26.555748	26.575296	26.555748	26.575296	313	0.0	1.0
3	2020-06-16	27.21	27.21	27.21	27.21	67	26.594844	26.594844	26.594844	26.594844	67	0.0	1.0
4	2020-06-17	27.19	27.20	27.18	27.20	2509	26.575296	26.585070	26.565522	26.585070	2509	0.0	1.0

Using Numpy

In [23]:
```python
close = cpi_df['adjClose'].to_numpy()
close
```

Out[23]:
```
array([26.45800902, 26.54597433, 26.55574825, 26.59484394, 26.57529609,
       26.55574825, 26.54597433, 26.54597433, 26.52642648, 26.50687863,
       26.48733079, 26.46778294, 26.51303621, 26.5362004 , 26.62416571,
       26.57529609, 26.57558931, 26.5362004 , 26.56552217, 26.5362004 ,
       26.53131344, 26.51665256, 26.52642648, 26.54597433, 26.52642648,
       26.56552217, 26.5362004 , 26.54597433, 26.58536323, 26.57744636,
       26.56034199, 26.58634063, 26.62416571, 26.69610178, 26.68388438,
       26.67166697, 26.65368295, 26.69698143, 26.67753133, 26.68261377,
       26.71692024, 26.7218072 , 26.66834384, 26.69747013, 26.6627727 ,
       26.66296818, 26.68906456, 26.67577202, 26.59933994, 26.63872885,
       26.64361581, 26.66316366, 26.66648679, 26.64371355, 26.66814836,
       26.69737239, 26.68984647, 26.69747013, 26.72170946, 26.68632786,
       26.6632614 , 26.61839909, 26.67381724, 26.60764778, 26.60452012,
       26.68769621, 26.70890562, 26.73656582, 26.69072612, 26.61908327,
       26.56073295, 26.59494168, 26.5312157 , 26.54568111, 26.59015246,
       26.64293164, 26.6278798 , 26.63296224, 26.67674941, 26.70704857,
       26.71692024, 26.70343222, 26.72034111, 26.75601593, 26.75591819,
       26.78514222, 26.74614426, 26.72190494, 26.72649868, 26.73167886,
       26.66316366, 26.67802002, 26.67059184, 26.52017117, 26.68251603,
       26.65251008, 26.62631597, 26.54841781, 26.57959662, 26.52828353,
       26.59064115, 26.64772086, 26.7013797 , 26.71701797, 26.69287639,
       26.74145278, 26.78152587, 26.7953071 , 26.77702986, 26.83577114,
       26.85913081, 26.87349848, 26.84368801, 26.85404837, 26.85346194,
       26.85346194, 26.85952177, 26.8633336 , 26.87340074, 26.85219133,
       26.8633336 , 26.84896593, 26.87369396, 26.88786615, 26.88258823,
       26.89285085, 26.86343134, 26.81534364, 26.85873986, 26.86079238,
       26.89226441, 26.9091733 , 26.91718791, 26.90252703, 26.88307692,
       26.8905051 , 26.89676041, 26.90252703, 26.92197714, 26.91718791,
       26.92143957, 26.9438454 , 26.92627607, 26.93130998, 26.91127305,
       26.94137779, 26.92114346, 26.88245146, 26.87504865, 26.85629488,
       26.84119315, 26.86705362, 26.93634389, 26.89094001, 26.94601689,
       26.86428991, 26.96072379, 26.955986  , 26.84770762, 26.90406765,
       26.7118908 , 26.87346939, 26.73627071, 26.85728192, 26.89686225,
       26.92647348, 26.86399379, 26.98056331, 26.97730608, 26.99092724,
       26.99023631, 26.94305576, 26.93861408, 26.94206872, 26.92134087,
       26.94621429, 26.93150739, 26.9341724 , 26.85293894, 26.84336464,
       26.86438861, 26.85738062, 26.8206627 , 26.77002751, 26.79805947,
       26.80822599, 26.80812728, 26.82322901, 26.83783721, 26.81513527,
       26.81789899, 26.82470957, 26.82697976, 26.79292686, 26.791545  ,
       26.81266767, 26.81523398, 26.80812728, 26.7939139 , 26.8086208 ,
       26.79302556, 26.77604846, 26.7747653 , 26.79332167, 26.84020611,
       26.85402468, 26.85895989, 26.88866981, 26.8932102 , 26.89824411,
       26.92617737, 26.91393806, 26.96654734, 26.94473373, 26.95568989,
       26.94690522, 26.96585641, 26.94799097, 26.97720738, 26.97562811,
       26.96398103, 26.96664604, 26.97108772, 26.94660911, 26.9609212 ,
       26.93950242, 26.96595511, 26.9802672 , 27.01017454, 26.97809571,
       26.95677563, 26.89390113, 26.91808363, 26.94621429, 26.9511495 ,
       26.9512482 , 26.92153827, 26.96595511, 26.9609212 , 26.99053243,
       26.99556634, 26.99102595, 26.99388837, 27.00060024, 27.00040284,
       27.004351  , 26.96121731, 27.02665812, 27.03001406, 27.02063717,
       27.04363522, 27.06031621, 27.04995228, 27.02024236, 27.02547367,
       26.9560847 , 27.00040284, 26.9560847 , 26.97089031, 26.98076072,
       27.01037195, 27.014024  , 27.01066806, 27.03978576, 27.0745296 ,
       27.03998317, 27.02024236, 27.08440001, 27.0794648 , 27.0987121 ,
       27.09417171, 27.1301987 , 27.09910692, 27.06456049, 27.09969914,
       27.12111793, 27.08834817, 27.05686157, 27.08163629, 27.06613975,
       27.11884773, 27.13523261, 27.12881685, 27.14322764, 27.15220971,
       27.16770625, 27.15329546, 27.17323368, 27.19800841, 27.16375809,
       27.15842807, 27.09427042, 27.11401123, 27.10907603, 27.12891555,
       27.13385075, 27.1640542 , 27.16583088, 27.15724362, 27.04491838,
       27.03149462, 26.94621429, 27.17816889, 27.17816889, 27.15349287,
       27.15349287, 27.1979097 , 27.21577514, 27.20787881, 27.20195657,
       27.22159868, 27.22416499, 27.17895852, 27.188138  , 27.20778011,
       27.08321556, 27.16336327, 27.09772506, 27.1446095 , 27.19800841,
       27.15625658, 27.12931037, 27.15842807, 27.16859459, 27.15418379,
       27.15753973, 27.0745296 , 27.06179677, 27.08114277, 27.10917473,
       27.14381987, 27.12092052, 27.14855766, 27.11065529, 27.16069826,
       27.10295638, 27.09920562, 27.13039611, 27.15003822, 27.18734837,
       26.98569593, 27.15349287, 27.1519136 , 27.12615184, 27.20778011,
       27.24755786, 27.25752697, 27.26186995, 27.26700256, 27.2921721 ,
       27.28230169, 27.27726778, 27.30154899, 27.27697167, 27.3163546 ,
       27.36807554, 27.38061096, 27.39551527, 27.3260276 , 27.3018451 ,
       27.33116021, 27.29671249, 27.26759478, 27.28674338, 27.30648419,
       27.29927879, 27.24232654, 27.27687297, 27.27183906, 27.20472028,
       27.22900149, 27.18310409, 27.13691058, 27.1882367 , 27.16760755,
       27.22159868, 27.2196246 , 27.25950105, 27.22258572, 27.3210924 ,
       27.22752093, 27.23245613, 27.24726174, 27.22544814, 27.19623173,
       27.17915593, 27.19386284, 27.24657081, 27.29661378, 27.32000665,
       27.33422004, 27.32200047, 27.36946099, 27.29514912, 27.26710877,
       27.32200047, 27.20538037, 27.21162256, 27.21300972, 27.19576936,
       27.26473079, 27.28246656, 27.23778035, 27.2675051 , 27.1830868 ,
       27.19249964, 27.16346847, 27.13840059, 27.11333272, 27.10124466,
       27.07082633, 27.05953092, 27.08975108, 27.09787585, 27.08915659,
       27.17337672, 27.03961534, 27.07954559, 27.08469788, 27.02475296,
       27.15851434, 27.08975108, 27.11888134, 27.07924834, 27.1077841 ,
       27.11036025, 27.12740244, 27.11392722, 27.06934009, 27.05943184,
       27.07478963, 27.1044153 , 27.11888134, 27.27741335, 27.68216538,
       27.39621327, 27.52511961, 27.84218362, 27.59447736, 27.34687018,
       27.2476886 , 27.28761885, 27.03971442, 27.01494379, 27.06468321,
       27.4719123 , 27.57485902, 27.56703151, 27.6055746 , 27.74329928,
       27.78075247, 27.79264237, 27.62717459, 27.69355986, 27.74349745,
       27.53354162, 27.4157028 , 27.57127051, 27.35695486, 27.1978086 ,
       27.25496606, 27.28538378, 27.12504466, 27.20924009, 27.35606022,
       27.28647722, 27.2906522 , 27.30635808, 27.38588151, 27.28687484,
       26.98408938, 26.94850264, 26.86897921, 26.76957492, 27.10088942,
       26.72772572, 26.63108487, 26.67206574, 27.0234443 , 26.64035791,
       26.61094342, 26.16882862, 26.08965876, 26.18059442, 26.15466978,
       26.4240866 , 26.45330167, 26.52698747, 26.23593355, 26.18887036,
       26.16733297, 26.31241123, 26.34142689, 26.44871501, 26.6697724 ,
       26.90179792, 26.74664893, 26.705     , 26.926     , 26.8977    ,
       26.8598    , 27.        , 26.925     , 26.7381    , 26.4453    ])
```

In [24]:
```python
cpi_df['adjClose'].dtype
```

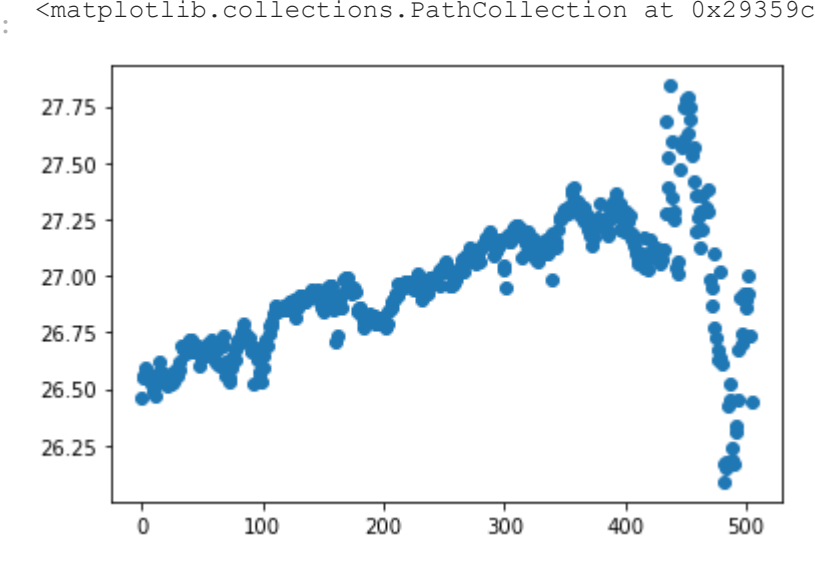
Out[24]:
```
dtype('float64')
```

In [25]:
```python
year = list(range(len(close)))
A = np.vstack([year, np.ones(len(year))]).T
A
```

Out[25]:
```
array([[  0.,   1.],
       [  1.,   1.],
       [  2.,   1.],
       ...,
       [502.,   1.],
       [503.,   1.],
       [504.,   1.]])
```

In [26]:
```python
plt.scatter(x=year, y=close)
```

Out[26]:
```
<matplotlib.collections.PathCollection at 0x29359cd1d90>
```



In [31]:
```python
m, b = np.linalg.lstsq(A, close, rcond=None)[0]
print(f'OLS is y = {m:.10f} x + {b:.2f}')
```

```
OLS is y = 0.0011365998 x + 26.67
```

Using Sklearn

In [35]:
```python
from sklearn.linear_model import LinearRegression
yhat = close
year = np.array(year)
x_years = year.reshape((-1, 1))
```

In [36]:
```python
model = LinearRegression()
model.fit(x_years, yhat)
```

Out[36]:
```
LinearRegression()
```

In [37]:
```python
mm = model.coef_
bb = model.intercept_
mm[0]
bb
print(f'OLS is y = {mm[0]:.10f} x + {bb:.2f}')
```

```
OLS is y = 0.0011365998 x + 26.67
```

Other Stocks

In [42]:
```python
import datetime as dt
import pandas_datareader as pdr
tickers = ['GE','IBM','WFC','^GSPC']
start = dt.datetime(2021, 1, 1)
end = dt.datetime(2022, 1, 1)

df1 = pdr.get_data_yahoo(tickers, start, end, interval="d")
df1.head()
```

Out[42]:

Attributes	Adj Close				Close				High		...	
Symbols	**GE**	**IBM**	**WFC**	**^GSPC**	**GE**	**IBM**	**WFC**	**^GSPC**	**GE**	**IBM**	**...**	**WFC**
Date												
2021-01-04	83.421616	110.134727	29.011139	3700.649902	83.760002	118.489487	29.700001	3700.649902	87.199997	120.382408	...	29.400000
2021-01-05	85.811913	112.089684	29.821888	3726.860107	86.160004	120.592735	30.530001	3726.860107	87.040001	121.108986	...	29.820000
2021-01-06	90.512833	114.888817	31.931789	3748.139893	90.879997	123.604210	32.689999	3748.139893	92.959999	126.080307	...	31.790001
2021-01-07	89.795746	114.622231	32.654629	3803.790039	90.160004	123.317398	33.430000	3803.790039	92.559998	124.722755	...	33.320000
2021-01-08	90.353493	114.213470	32.420189	3824.679932	90.720001	122.877632	33.189999	3824.679932	91.519997	123.632889	...	32.669998

5 rows × 24 columns

In [44]:
```python
price_df = df1['Adj Close']
ret_df = price_df.pct_change()
ret_df.dropna(inplace=True)
```

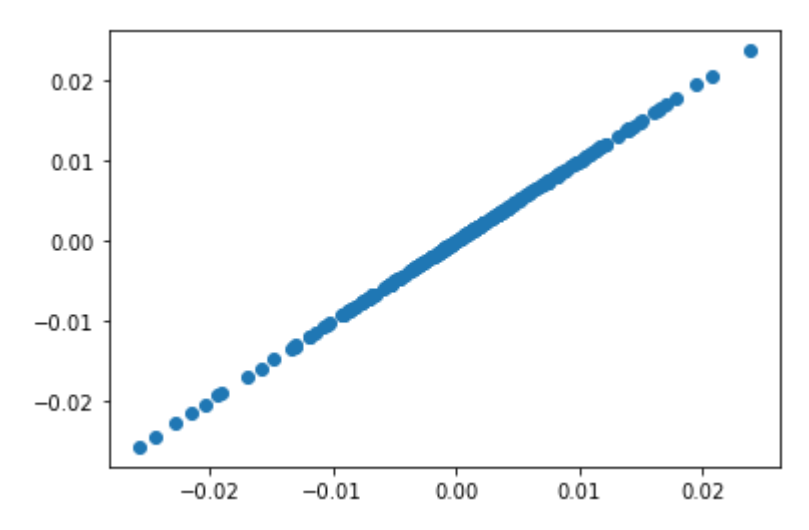
In [45]:
```python
ret_df = ret_df[ret_df['^GSPC'] != 0.0]
```

In [58]:
```python
ret_df.head()
ret_df.shape
```

Out[58]:
```
(251, 4)
```

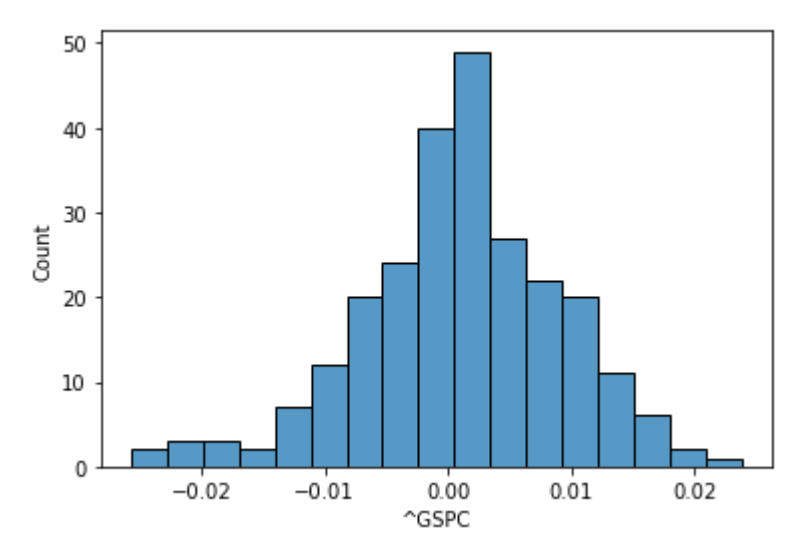
In [47]:
```python
plt.scatter(x=ret_df['^GSPC'], y=ret_df['^GSPC'])
```

Out[47]:
```
<matplotlib.collections.PathCollection at 0x2935bfc0f70>
```



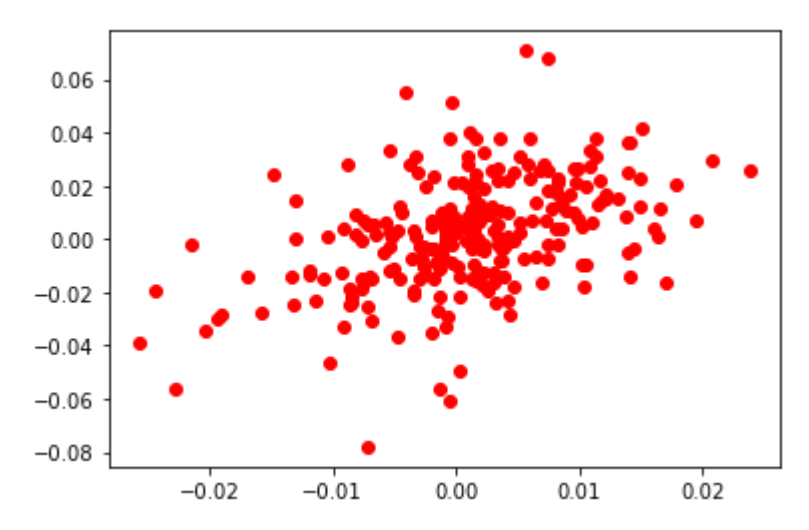
In [49]:
```python
sns.histplot(ret_df['^GSPC'])
```

Out[49]:
```
<AxesSubplot:xlabel='^GSPC', ylabel='Count'>
```



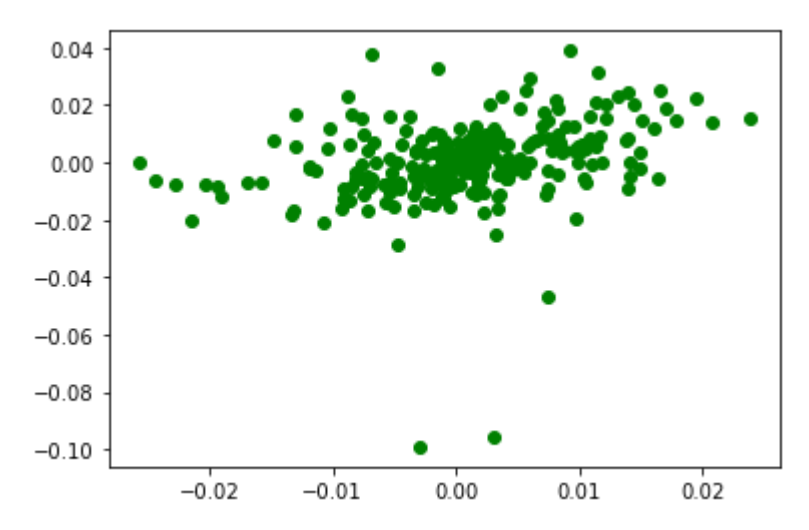
In [50]:
```python
plt.scatter(x=ret_df['^GSPC'], y=ret_df['WFC'], c='r')
```

Out[50]:
```
<matplotlib.collections.PathCollection at 0x2935c2f3fa0>
```



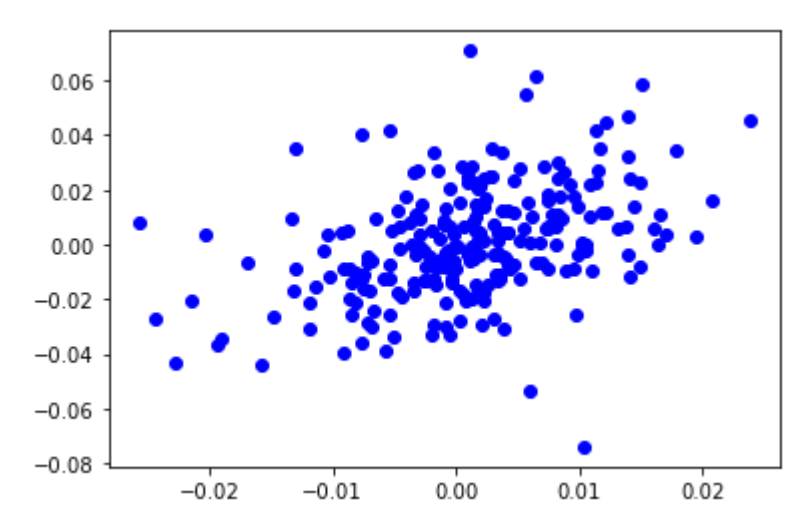
In [51]:
```python
plt.scatter(x=ret_df['^GSPC'], y=ret_df['IBM'], c='g')
```

Out[51]:
```
<matplotlib.collections.PathCollection at 0x2935c076a90>
```



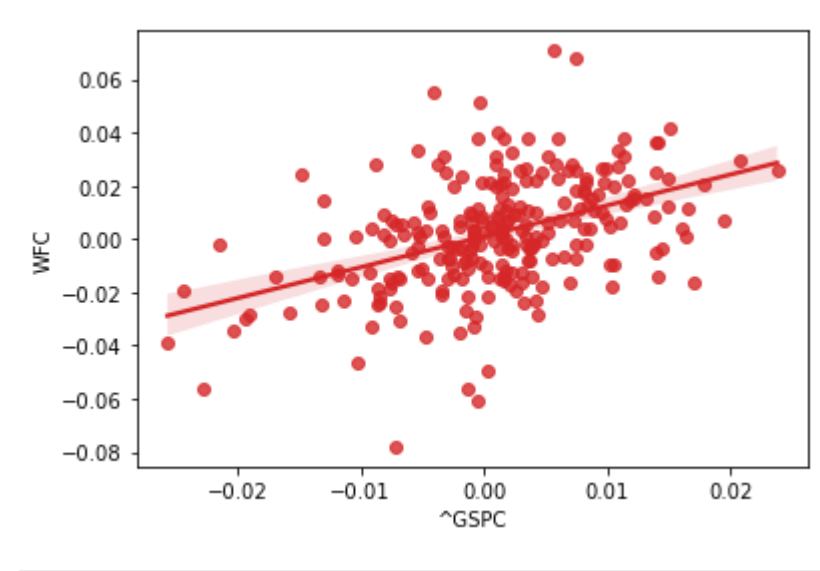
In [52]:
```python
plt.scatter(x=ret_df['^GSPC'], y=ret_df['GE'], c='b')
```

Out[52]:
```
<matplotlib.collections.PathCollection at 0x2935c028550>
```



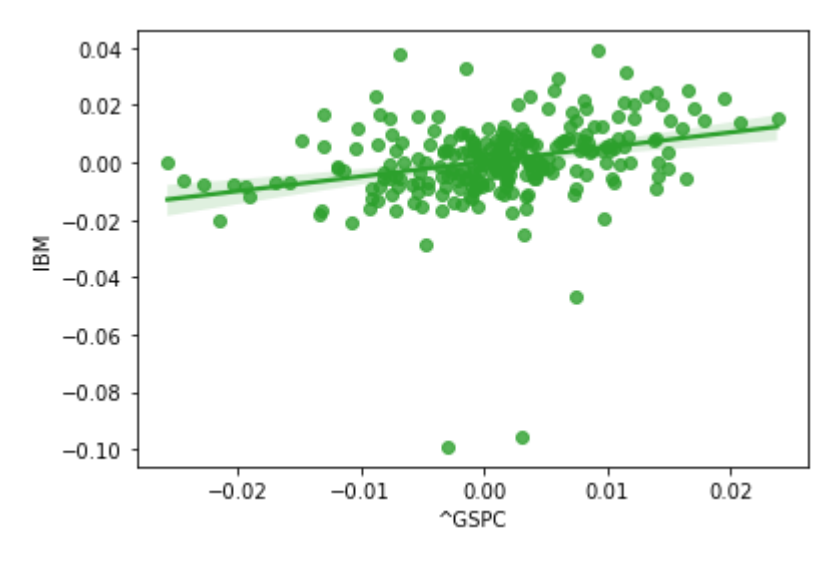
In [53]:
```python
sns.regplot(data=ret_df, x='^GSPC', y='WFC', color='tab:red', fit_reg=True)
```

Out[53]:
```
<AxesSubplot:xlabel='^GSPC', ylabel='WFC'>
```



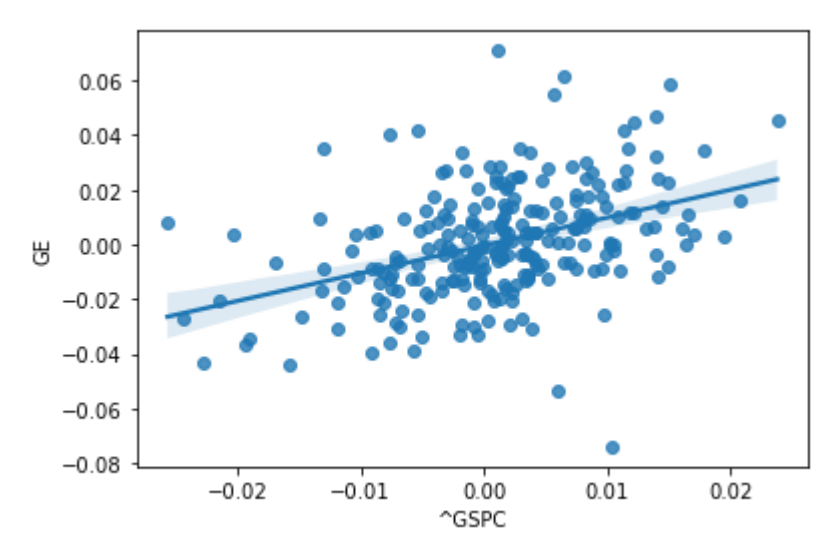
In [54]:
```python
sns.regplot(data=ret_df, x='^GSPC', y='IBM', color='tab:green', fit_reg=True)
```

Out[54]:
```
<AxesSubplot:xlabel='^GSPC', ylabel='IBM'>
```



In [55]:
```python
sns.regplot(data=ret_df, x='^GSPC', y='GE', color='tab:blue', fit_reg=True)
```

Out[55]:
```
<AxesSubplot:xlabel='^GSPC', ylabel='GE'>
```



In [62]:
```python
ticker = ['WFC','IBM','GE']

for i in ticker:
    regr = LinearRegression()
    X = ret_df['^GSPC'].values.reshape(-1,1)
    y = ret_df[i].values.reshape(-1,1)
    regr.fit(X, y)
    bbb = regr.intercept_[0]
    mmm = regr.coef_[0][0]
    print(f'OLS alpha and beta for {i} is y = {mmm:.2f} x + {bbb:.10f}')
```

```
OLS alpha and beta for WFC is y = 1.16 x + 0.0009818659
OLS alpha and beta for IBM is y = 0.51 x + 0.0002477049
OLS alpha and beta for GE is y = 1.02 x + -0.0003595743
```