CAPM

In [1]:
```
import pandas_datareader.data as reader
import pandas as pd
import datetime as dt
import statsmodels.api as sm
```

In [44]:
```
end = dt.date(2019,12,31)
start = dt.date(2010, 3, 30)
funds = ['LMT']
```

In [45]:
```
fundsret =  reader.get_data_yahoo(funds, start, end)['Adj Close'].pct_change()
```

In [46]:
```
fundsret_mtl = fundsret.resample('M').agg(lambda x: (x+1).prod() -1)
```

In [47]:
```
fundsret_mtl
```

Out[47]:

Symbols	**LMT**
Date	
2010-03-31	-0.010346
2010-04-30	0.020068
2010-05-31	-0.051048
2010-06-30	-0.067818
2010-07-31	0.008725
...	...
2019-08-31	0.066684
2019-09-30	0.015490
2019-10-31	-0.034302
2019-11-30	0.044454
2019-12-31	-0.004220

118 rows × 1 columns

```
In [48]: fundsret_mtl = fundsret_mtl[1:]
         fundsret_mtl
```

```
Out[48]:
            Symbols        LMT
               Date
         2010-04-30   0.020068
         2010-05-31  -0.051048
         2010-06-30  -0.067818
         2010-07-31   0.008725
         2010-08-31  -0.066782
                ...        ...
         2019-08-31   0.066684
         2019-09-30   0.015490
         2019-10-31  -0.034302
         2019-11-30   0.044454
         2019-12-31  -0.004220

         117 rows × 1 columns
```

```
In [49]: fundsret_mtl.to_csv("LMT-2.csv")
```

```
In [35]: factors.to_csv("LMT.csv")
```

```
In [33]: factors = reader.DataReader('F-F_Research_Data_Factors','famafrench', start, end)
```

```
In [11]: factors = factors[1:]
```

In [12]:
```python
factors
```

Out[12]:

Date	Mkt-RF	SMB	HML	RF
2010-01	-3.36	0.40	0.43	0.00
2010-02	3.40	1.19	3.22	0.00
2010-03	6.31	1.48	2.21	0.01
2010-04	2.00	4.87	2.89	0.01
2010-05	-7.89	0.09	-2.44	0.01
...	...	...	...	...
2019-08	-2.58	-2.32	-4.95	0.16
2019-09	1.43	-0.97	6.83	0.18
2019-10	2.06	0.28	-1.93	0.15
2019-11	3.87	0.80	-2.02	0.12
2019-12	2.77	0.72	1.79	0.14

120 rows × 4 columns

In [13]:
```python
factors.to_csv("Factors.csv")
```

In [50]:
```python
factors_2 = pd.read_csv("Factors-2.csv")
factors_2
```

Out[50]:

	Date	LMT	MKT-RF	SMB	HML	RF
0	2010-04	0.020068	-1.266800	4.87	2.89	-1.197163
1	2010-05	-0.051048	-5.019409	0.09	-2.44	6.377697
2	2010-06	-0.067818	-1.135353	-1.81	-4.70	-0.165529
3	2010-07	0.008725	5.181172	0.24	-0.30	-2.137014
4	2010-08	-0.066782	-0.973188	-2.95	-1.96	-0.971223
...	...	...	...	...	...	...
112	2019-08	0.066684	8.428932	-2.32	-4.95	-8.845145
113	2019-09	0.015490	-0.481913	-0.97	6.83	-1.642636
114	2019-10	-0.034302	0.767749	0.28	-1.93	-1.671758
115	2019-11	0.044454	3.412230	0.80	-2.02	-0.517136
116	2019-12	-0.004220	3.945300	0.72	1.79	-5.347381

117 rows × 6 columns

In [51]:
```python
df = factors_2
```

In [52]:
```python
df = df.set_index('Date')
df
```

Out[52]:

Date	LMT	MKT-RF	SMB	HML	RF
2010-04	0.020068	-1.266800	4.87	2.89	-1.197163
2010-05	-0.051048	-5.019409	0.09	-2.44	6.377697
2010-06	-0.067818	-1.135353	-1.81	-4.70	-0.165529
2010-07	0.008725	5.181172	0.24	-0.30	-2.137014
2010-08	-0.066782	-0.973188	-2.95	-1.96	-0.971223
...	...	...	...	...	...
2019-08	0.066684	8.428932	-2.32	-4.95	-8.845145
2019-09	0.015490	-0.481913	-0.97	6.83	-1.642636
2019-10	-0.034302	0.767749	0.28	-1.93	-1.671758
2019-11	0.044454	3.412230	0.80	-2.02	-0.517136
2019-12	-0.004220	3.945300	0.72	1.79	-5.347381

117 rows × 5 columns

In [53]:
```python
df[['MKT-RF','SMB','HML','RF']] = df[['MKT-RF','SMB','HML','RF']]/100
```

In [54]:
```python
df[['MKT-RF','SMB','HML','RF']]
```

Out[54]:

Date	MKT-RF	SMB	HML	RF
2010-04	-0.012668	0.0487	0.0289	-0.011972
2010-05	-0.050194	0.0009	-0.0244	0.063777
2010-06	-0.011354	-0.0181	-0.0470	-0.001655
2010-07	0.051812	0.0024	-0.0030	-0.021370
2010-08	-0.009732	-0.0295	-0.0196	-0.009712
...	...	...	...	...
2019-08	0.084289	-0.0232	-0.0495	-0.088451
2019-09	-0.004819	-0.0097	0.0683	-0.016426
2019-10	0.007677	0.0028	-0.0193	-0.016718
2019-11	0.034122	0.0080	-0.0202	-0.005171
2019-12	0.039453	0.0072	0.0179	-0.053474

117 rows × 4 columns

In [55]:

```
df['LMT-RF'] = df.LMT - df.RF
```

In [57]:

```
df
```

Out[57]:

	LMT	MKT-RF	SMB	HML	RF	LMT-RF
Date						
2010-04	0.020068	-0.012668	0.0487	0.0289	-0.011972	0.032039
2010-05	-0.051048	-0.050194	0.0009	-0.0244	0.063777	-0.114825
2010-06	-0.067818	-0.011354	-0.0181	-0.0470	-0.001655	-0.066163
2010-07	0.008725	0.051812	0.0024	-0.0030	-0.021370	0.030095
2010-08	-0.066782	-0.009732	-0.0295	-0.0196	-0.009712	-0.057069
...	...	...	...	...	...	...
2019-08	0.066684	0.084289	-0.0232	-0.0495	-0.088451	0.155135
2019-09	0.015490	-0.004819	-0.0097	0.0683	-0.016426	0.031917
2019-10	-0.034302	0.007677	0.0028	-0.0193	-0.016718	-0.017585
2019-11	0.044454	0.034122	0.0080	-0.0202	-0.005171	0.049626
2019-12	-0.004220	0.039453	0.0072	0.0179	-0.053474	0.049254

117 rows × 6 columns

In [62]:

```
y = df["LMT-RF"]
x =df[['MKT-RF','SMB','HML','RF']]
```

In [63]:

```
x_sm = sm.add_constant(x)
```

```
C:\Users\proma.gupta\Anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:14
2: FutureWarning: In a future version of pandas all arguments of concat except
for the argument 'objs' will be keyword-only
  x = pd.concat(x[::order], 1)
```

In [64]:

```
model = sm.OLS(y, x_sm)
```

In [65]:
```
results = model.fit()
results.summary()
```

Out[65]:
```
OLS Regression Results

       Dep. Variable:          LMT-RF          R-squared:     0.690
              Model:              OLS     Adj. R-squared:     0.679
             Method:    Least Squares        F-statistic:     62.30
               Date: Wed, 13 Apr 2022 Prob (F-statistic):  1.32e-27
               Time:         11:18:49     Log-Likelihood:    187.68
   No. Observations:              117                AIC:    -365.4
       Df Residuals:              112                BIC:    -351.6
           Df Model:                4
    Covariance Type:        nonrobust

              coef  std err        t  P>|t|  [0.025  0.975]
     const  0.0173    0.005    3.700  0.000   0.008   0.027
    MKT-RF  0.0006    0.004    0.143  0.887  -0.008   0.009
       SMB  0.0834    0.208    0.400  0.690  -0.330   0.496
       HML  0.0584    0.207    0.281  0.779  -0.353   0.469
        RF -1.0148    0.065  -15.650  0.000  -1.143  -0.886

         Omnibus:  5.192    Durbin-Watson:  2.210
   Prob(Omnibus):  0.075  Jarque-Bera (JB):  5.321
            Skew: -0.307          Prob(JB): 0.0699
        Kurtosis:  3.845          Cond. No.   54.3


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

In [ ]:
```
##RF
```

In [66]:
```python
df = pd.read_csv("Factors-data.csv")
df
```

Out[66]:
```
        Date  Mkt-RF   SMB    HML    RF    LMT
  0  2010-04    2.00  4.87   2.89  0.01   0.02
  1  2010-05   -7.89  0.09  -2.44  0.01  -0.05
  2  2010-06   -5.57 -1.81  -4.70  0.01  -0.07
  3  2010-07    6.93  0.24  -0.30  0.01   0.01
  4  2010-08   -4.77 -2.95  -1.96  0.01  -0.07
...      ...     ...   ...    ...   ...    ...
112  2019-08   -2.58 -2.32  -4.95  0.16   0.07
113  2019-09    1.43 -0.97   6.83  0.18   0.02
114  2019-10    2.06  0.28  -1.93  0.15  -0.03
115  2019-11    3.87  0.80  -2.02  0.12   0.04
116  2019-12    2.77  0.72   1.79  0.14   0.00

117 rows × 6 columns
```

In [67]:
```python
df = df.set_index('Date')
df
```

Out[67]:
```
         Mkt-RF   SMB    HML    RF    LMT
   Date
2010-04    2.00  4.87   2.89  0.01   0.02
2010-05   -7.89  0.09  -2.44  0.01  -0.05
2010-06   -5.57 -1.81  -4.70  0.01  -0.07
2010-07    6.93  0.24  -0.30  0.01   0.01
2010-08   -4.77 -2.95  -1.96  0.01  -0.07
    ...     ...   ...    ...   ...    ...
2019-08   -2.58 -2.32  -4.95  0.16   0.07
2019-09    1.43 -0.97   6.83  0.18   0.02
2019-10    2.06  0.28  -1.93  0.15  -0.03
2019-11    3.87  0.80  -2.02  0.12   0.04
2019-12    2.77  0.72   1.79  0.14   0.00

117 rows × 5 columns
```

In [69]:
```python
df[['Mkt-RF','SMB','HML','RF']] = df[['Mkt-RF','SMB','HML','RF']]/100
```

In [86]:

```
market_premium = df['Mkt-RF'].mean()
size_premium = df['SMB'].mean()
value_premium = df['HML'].mean()
print(market_premium, size_premium,value_premium)
```

```
0.010651282051282046 -0.00038717948717948716 -0.0025290598290598288
```

In [70]:

```
df['LMT-RF'] = df.LMT - df.RF
```

In [76]:

```
y = df["LMT-RF"]
x =df[['Mkt-RF','SMB','HML']]
```

In [77]:

```
x_sm = sm.add_constant(x)
```

```
C:\Users\proma.gupta\Anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:14
2: FutureWarning: In a future version of pandas all arguments of concat except
for the argument 'objs' will be keyword-only
  x = pd.concat(x[::order], 1)
```

In [78]:

```
model = sm.OLS(y, x_sm)
```

In [79]:

```python
results = model.fit()
results.summary()
```

Out[79]:

OLS Regression Results

Dep. Variable:	LMT-RF	**R-squared:**	0.335
Model:	OLS	**Adj. R-squared:**	0.317
Method:	Least Squares	**F-statistic:**	18.96
Date:	Wed, 13 Apr 2022	**Prob (F-statistic):**	4.96e-10
Time:	11:27:23	**Log-Likelihood:**	211.46
No. Observations:	117	**AIC:**	-414.9
Df Residuals:	113	**BIC:**	-403.9
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	0.0084	0.004	2.130	0.035	0.001	0.016
Mkt-RF	0.8174	0.109	7.515	0.000	0.602	1.033
SMB	-0.4132	0.178	-2.324	0.022	-0.765	-0.061
HML	-0.0551	0.166	-0.333	0.740	-0.383	0.273

Omnibus:	3.944	**Durbin-Watson:**	2.179
Prob(Omnibus):	0.139	**Jarque-Bera (JB):**	3.339
Skew:	-0.375	**Prob(JB):**	0.188
Kurtosis:	3.350	**Cond. No.**	49.3

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

In [80]:

```python
intercept, beta_m, beta_s, beta_v =  results.params
print(beta_m, beta_s, beta_v)
```

```
0.8174423269047699 -0.41321655031987864 -0.055086883541246384
```

In [ ]:

```python
##CAPM
```

In [83]:

```python
risk_free = df['RF'].mean()
risk_free
```

Out[83]:

```
0.00043675213675213674
```

```
In [89]: #market risk premium
         market_premium = df['Mkt-RF'].mean()
         market_premium
```

```
Out[89]: 0.010651282051282046
```

```
In [91]: #Expected return for LMT

         lmt_exp_return =  risk_free + beta_m*market_premium
         lmt_exp_return
```

```
Out[91]: 0.009143560921271143
```