

# Untitled

January 12, 2020

## 1 Multifactor CAPM Analysis

```
[124]: #Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

```
[125]: #Importing the dataset
df = pd.read_csv("Multifactor CAPM.csv", parse_dates=True)
```

```
[126]: df.head()
```

```
[126]:
```

	Date	SMB %	HML %	WML %	VIX %	Rm %	Rf %	\
0	01-Jan-15	0.634136	1.568746	0.241641	-0.63	0.349101	0.031936	
1	02-Jan-15	0.229000	0.281989	0.504802	-8.19	0.905457	0.031936	
2	05-Jan-15	0.700766	-0.323167	0.609444	2.59	-0.103746	0.031936	
3	06-Jan-15	0.377347	-1.698323	-0.268163	23.09	-2.778106	0.031936	
4	07-Jan-15	0.306515	-0.611127	1.035972	4.13	-0.033870	0.031936	

	Rm-Rf %	MSCI World (Rgm %)	Rgm-Rf %	Info Edge Returns %
0	0.317165	0.00	-0.031936	0.506121
1	0.873521	-0.29	-0.321936	0.052699
2	-0.135682	-1.96	-1.991936	-1.234857
3	-2.810042	-1.01	-1.041936	-1.114008
4	-0.065806	0.47	0.438064	1.000719

### 1.0.1 Single Factor CAPM

```
[127]: x1 = df['Rm-Rf %']
y1 = df['Info Edge Returns %']
```

```
[128]: x1 = sm.add_constant(x1)
```

```
[129]: model1 = sm.OLS(endog = y1, exog =x1).fit()
model1.summary()
```

```
[129]: <class 'statsmodels.iolib.summary.Summary'>
"""
```

```

                                OLS Regression Results
=====
Dep. Variable:      Info Edge Returns %      R-squared:                0.054
Model:                                OLS      Adj. R-squared:        0.053
Method:                    Least Squares      F-statistic:              70.03
Date:                Sun, 12 Jan 2020      Prob (F-statistic):      1.57e-16
Time:                16:56:01      Log-Likelihood:         -2711.7
No. Observations:      1234      AIC:                    5427.
Df Residuals:          1232      BIC:                    5438.
Df Model:                1
Covariance Type:      nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	0.1070	0.062	1.724	0.085	-0.015	0.229
Rm-Rf %	0.6091	0.073	8.368	0.000	0.466	0.752

```

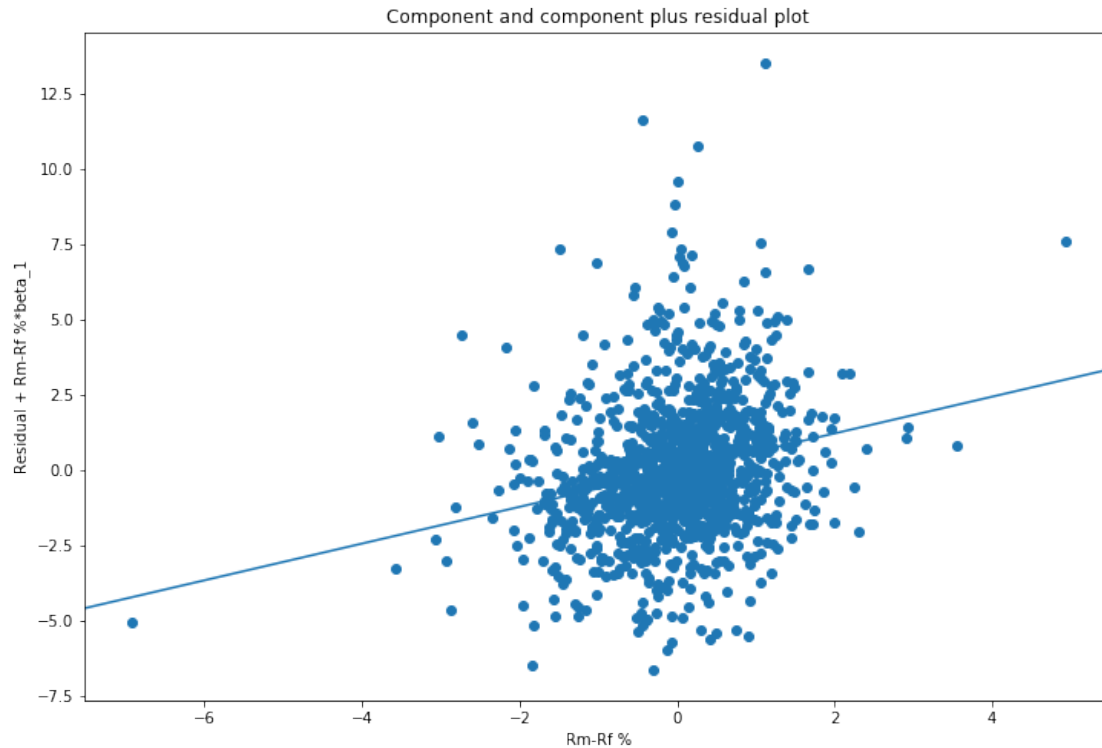
=====
Omnibus:                219.647      Durbin-Watson:            1.993
Prob(Omnibus):           0.000      Jarque-Bera (JB):         659.992
Skew:                    0.892      Prob(JB):                 4.84e-144
Kurtosis:                6.107      Cond. No.                 1.17
=====

```

```

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

```
[130]: fig1, ax1 = plt.subplots(figsize=(12, 8))
fig1 = sm.graphics.plot_ccpr(model1, "Rm-Rf %", ax=ax1)
```



## 1.0.2 Multifactor CAPM

```
[131]: x2 = df[['Rm-Rf %', 'Rgm-Rf %', 'SMB %', 'HML %', 'WML %', 'VIX %']]
       y2 = df['Info Edge Returns %']
```

```
[132]: x2 = sm.add_constant(x2)
```

```
[133]: model2 = sm.OLS(endog = y2, exog = x2).fit()
       model2.summary()
```

```
[133]: <class 'statsmodels.iolib.summary.Summary'>
       """
```

```

                                OLS Regression Results
=====
Dep. Variable:          Info Edge Returns %      R-squared:                0.059
Model:                            OLS          Adj. R-squared:        0.054
Method:                 Least Squares          F-statistic:             12.76
Date:                  Sun, 12 Jan 2020         Prob (F-statistic):       5.13e-14
Time:                  16:56:05                Log-Likelihood:          -2708.5
No. Observations:      1234                    AIC:                   5431.
Df Residuals:          1227                    BIC:                   5467.
Df Model:               6
```

Covariance Type: nonrobust

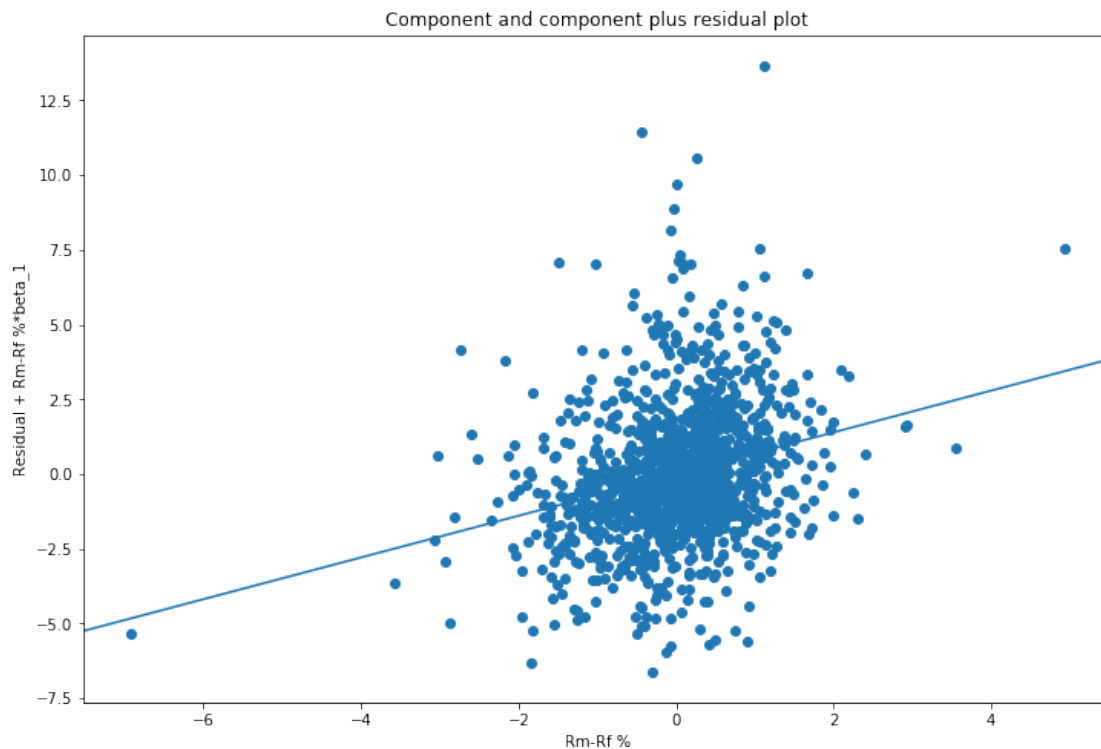
	coef	std err	t	P> t	[0.025	0.975]
const	0.1081	0.063	1.725	0.085	-0.015	0.231
Rm-Rf %	0.6977	0.105	6.641	0.000	0.492	0.904
Rgm-Rf %	-0.0819	0.093	-0.878	0.380	-0.265	0.101
SMB %	0.0804	0.097	0.826	0.409	-0.111	0.271
HML %	-0.1540	0.076	-2.027	0.043	-0.303	-0.005
WML %	-0.0838	0.084	-0.994	0.321	-0.249	0.082
VIX %	-0.0074	0.016	-0.480	0.632	-0.038	0.023
Omnibus:	221.835		Durbin-Watson:	1.990		
Prob(Omnibus):	0.000		Jarque-Bera (JB):	672.584		
Skew:	0.898		Prob(JB):	8.92e-147		
Kurtosis:	6.140		Cond. No.	9.70		

Warnings:

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

""

```
[134]: fig2, ax2 = plt.subplots(figsize=(12, 8))
fig2 = sm.graphics.plot_ccpr(model2, "Rm-Rf %", ax=ax2)
```



[ ]: