HW2_code

July 1, 2020

HW 2

<p style="list-style-type: none; font-size: 30px; line-height: 65px"><u>Team Members</u>:
<li style="font-size: 25px; line-height: 40px">Brandt Lawson
<li style="font-size: 25px; line-height: 40px">Stephen Kelly
<li style="font-size: 25px; line-height: 40px">Yishan Shi
</p>

#

Part 2: LTCM Risk

##

1. Summary Stats
   a) For both the gross and net series of LTCM excess returns, report the mean and volatility. Since this is monthly data, scale the mean by 12, and scale the volatility by $\sqrt{12}$

[1]:
```
import pandas as pd
import numpy as np
import math
pd.options.display.float_format = '{:,.4f}'.format

# ./ steps into this file's folder and lets you access the data file by just␣
↪its name
path_to_data_file = './hedge_data.xls'
LTCM_excess_returns = pd.read_excel(path_to_data_file)

def mean_vol(data):

    mean = data.mean()*12
    vol = data.std()*math.sqrt(12)
    df = pd.DataFrame(data={'Mean': mean, 'Vol': vol});
    return df

display(mean_vol(LTCM_excess_returns))
```

```
        Mean    Vol
gross 0.2572 0.1380
net   0.1689 0.1132
```

b) Report the annualized Sharpe ratio, (the SR based on the annualized mean and volatility.)

[2]:
```
def sharpe_calc(mean_vol):
    mean_vol['Sharpe'] = mean_vol['Mean']/mean_vol['Vol']
    return mean_vol
sharpe_calc(mean_vol(LTCM_excess_returns))
```

[2]:
```
        Mean    Vol  Sharpe
gross 0.2572 0.1380  1.8636
net   0.1689 0.1132  1.4924
```

c) Comment on whether the mean, volatility, and Sharpe ratio seem especially high or low relative to other assets we have seen.

- LTCM's Sharpe here is almost 1.5, which would be considered very good and higher than most other assets. While LTCM's Vol of .11 is higher than many asset classes, namely bonds and HY bonds, LTCM's higher Sharpe tells us that its returns are more than making up for the extra risk it is taking.
- We saw on HW #1 that domestice equity had a Sharpe of 1.2 from 2009-2019, which was also when domestice equity saw a very long recovery and bull market.

2. Using the series of net LTCM excess returns, denoted $\tilde{r}t$LTCM, estimate the following regression:

$\tilde{r}t\text{LTCM} = \alpha + \beta\text{m}\ \tilde{r}t\text{m} + \epsilon t$

a) Report α and βm. Report the $R2$ stat.

[3]:
```
from scipy import stats

market_return = pd.read_excel('./hedge_data.xls', sheet_name='MktExcessRets', ␣
 ↪index_col='date')

def alpha_beta_R2(market_data, security_return):

    x = market_data.values
    x = x.reshape(security_return.values.shape[0],)
    alpha_beta_R2.x = x

    y = security_return.values
    alpha_beta_R2.y = y

    alpha_beta_R2.beta_m,\
    alpha_beta_R2.alpha,\
    alpha_beta_R2.rsquared,\
    pvalue,\
    stderr\
```

```
    = stats.linregress(x, y)
    alpha_beta_R2.answer = pd.DataFrame([{'Alpha' : alpha_beta_R2.alpha, 'Beta'␣
↪: alpha_beta_R2.beta_m, 'RSquared' : alpha_beta_R2.rsquared}])

alpha_beta_R2(market_return.loc['1994-04-30':'1998-06-30',:],␣
↪LTCM_excess_returns.loc[:,'net'])
alpha_beta_R2.answer
```

[3]:

	Alpha	Beta	RSquared
0	0.0134	0.0420	0.0406

b) From this regression, does LTCM appear to have much exposure to the equity-market factor, $\tilde{r}tm$?

No because the beta_m value tells us how much exposure the LTCM has to the market equity factor. So, since beta is relatively small, and a full one unit increase in the market equity is unlikely it appears that the LCTM does not have much exposure to the market equity factor.

0.1 3. Regression-based metrics.

a) Calculate the Treynor ratio.

[4]:
```
def TR(net_return_data):

    expected_r = net_return_data.mean()
    treynor = expected_r / alpha_beta_R2.beta_m
    TR.answer = pd.DataFrame([{'Treynor Ratio' : treynor}])

TR(LTCM_excess_returns.loc[:,'net'])
TR.answer
```

[4]:

	Treynor Ratio
0	0.3349

b) Calculate the Information ratio.

[5]:
```
def IR():

    predictions = alpha_beta_R2.beta_m*alpha_beta_R2.x + alpha_beta_R2.alpha
    residuals = predictions-alpha_beta_R2.y
    info_rat = alpha_beta_R2.alpha/(residuals).std()
    IR.answer = pd.DataFrame([{'Information Ratio' : info_rat}])

IR()
IR.answer
```

[5]:

	Information Ratio
0	0.4155

0.2 4. Tail risk.

a) Calculate the 5th worst return of the sample.

[6]:
```
#Using the net LTCM returns

LTCM_excess_returns_sorted = LTCM_excess_returns.
 ↪sort_values(by=['net'],ascending=True, ignore_index=True)
LTCM_excess_returns_sorted.index = LTCM_excess_returns_sorted.index + 1
LTCM_excess_returns_sorted['date'] = LTCM_excess_returns_sorted['date'].
 ↪astype(str)
df = LTCM_excess_returns_sorted.loc[5::-1,['date', 'net']].style.
 ↪set_properties(subset=pd.IndexSlice[pd.IndexSlice[5], pd.IndexSlice[:]],␣
 ↪**{'background-color': '#ff6666', 'color': 'black'})
df
```

[6]:
```
<pandas.io.formats.style.Styler at 0x1a1c724e50>
```

b) Calculate the mean of the worst 4 returns from the sample.

[7]:
```
LTCM_excess_returns_sorted.loc[:4,['date','net']].mean()
```

[7]:
```
net   -0.0566
dtype: float64
```

c) Report the skewness of the return distribution. Compare to a normal distribution with skewness of 0.

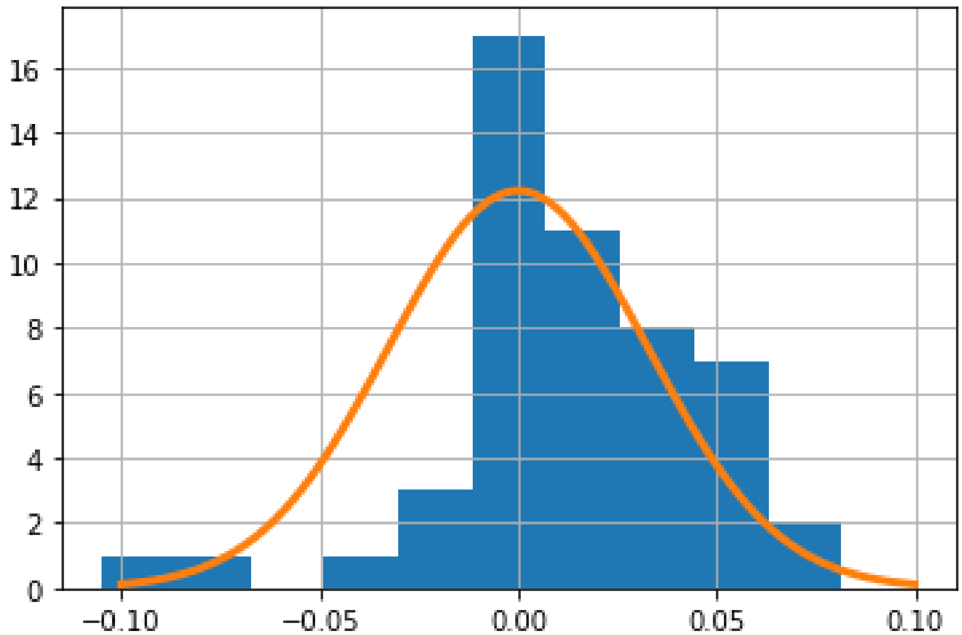
- The skewness of the returns is negative meaning the returns distrobutions are skewed left

[8]:
```
from scipy.stats import kurtosis
from scipy.stats import skew
from scipy.stats import norm
import matplotlib.pyplot as plt

print("The Skewness of the return distribution is " +␣
 ↪str(skew(LTCM_excess_returns.iloc[:,2:])[0]))

LTCM_excess_returns.loc[:,'net'].hist()
x = np.linspace(-.1,.1,1000)
y = norm.pdf(x, loc=0, scale=LTCM_excess_returns.loc[:,'net'].std())
plt.plot(x,y, linewidth=3);
```

```
The Skewness of the return distribution is -0.8318899364747497
```



d) Report the kurtosis of the return distribution. Compare to a normal distribution with kurtosis of three.

- The Kurt of the returns is larger than a normal distrobution. This means the returns have skinnier tails i.e. return distributions are more concentrated close to the mean and don't trail off in the same way a normal distrobution does.

[9]:
```
print("The kurtosis of the return distribution is " +␣
↪str(kurtosis(LTCM_excess_returns.iloc[:,2:])[0]))
```

```
The kurtosis of the return distribution is 2.5874860101941097
```

#

Part 3: Other Hedge Fund Indexes

Analyze the Total Index fund (of the second tab in the data file), by calculating the same statistics you estimated for LTCM. So if you wrote the code above well, it can mostly be re-used for this.

[10]:
```
hedge_excess_rets = pd.read_excel('./hedge_data.xls',␣
↪sheet_name='HedgeFund_ExcessRets',  index_col='date')

hedge_df = sharpe_calc(mean_vol(hedge_excess_rets))
hedge_df['Skew'] = skew(hedge_excess_rets)
hedge_df['Kurt'] = kurtosis(hedge_excess_rets)
hedge_df['5th Percentile'] = hedge_excess_rets.quantile(0.05)
display(hedge_df)
```

```
                                              Mean      Vol   Sharpe       Skew       Kurt  \
```

Total Index	0.0604	0.0732	0.8243	-0.2558	2.6879
Convertible Arbitrage	0.0458	0.0686	0.6677	-2.6403	16.0553
Dedicated Short Bias	-0.0656	0.1682	-0.3898	0.6864	1.4063
Emerging Markets	0.0514	0.1457	0.3531	-0.8801	5.4616
Equity Market Neutral	0.0247	0.1019	0.2428	-11.9950	163.8178
Event Driven	0.0606	0.0622	0.9749	-2.2978	11.3991
Event Driven Distressed	0.0707	0.0649	1.0900	-2.2738	11.7608
Event Driven Multi-Strategy	0.0559	0.0675	0.8276	-1.7772	7.8675
Event Driven Risk Arbitrage	0.0337	0.0407	0.8259	-1.1021	5.1140
Fixed Income Arbitrage	0.0271	0.0568	0.4779	-4.4191	30.8270
Global Macro	0.0922	0.0938	0.9831	-0.0720	4.3355
Long/Short Equity	0.0661	0.0977	0.6771	-0.0774	3.3262
Managed Futures	0.0288	0.1172	0.2455	0.0370	-0.0592
Multi-Strategy	0.0484	0.0534	0.9078	-1.6513	5.7122

	5th Percentile
Total Index	-0.0266
Convertible Arbitrage	-0.0220
Dedicated Short Bias	-0.0757
Emerging Markets	-0.0707
Equity Market Neutral	-0.0130
Event Driven	-0.0264
Event Driven Distressed	-0.0242
Event Driven Multi-Strategy	-0.0270
Event Driven Risk Arbitrage	-0.0153
Fixed Income Arbitrage	-0.0137
Global Macro	-0.0285
Long/Short Equity	-0.0407
Managed Futures	-0.0507
Multi-Strategy	-0.0227

##

2. For each series, run a regression of the series on the market-equity factor. Report the following for each regression:

-

0.2.1 Beta

-

0.2.2 Alpha

-

0.2.3 R-squared

-

0.2.4 Treynor Ratio

-

0.2.5 Information Ratio

[11]:
```
hedge_excess_rets = pd.read_excel('./hedge_data.xls',␣
↪sheet_name='HedgeFund_ExcessRets',  index_col='date')
regression_df = pd.DataFrame()

for (columnData) in hedge_excess_rets.iteritems():

    alpha_beta_R2(market_return, columnData[1])
    TR(columnData[1])
    IR()

    regression_df[columnData[0]] = alpha_beta_R2.answer.loc[0,:].append(TR.
↪answer.loc[0,:]).append(IR.answer.loc[0,:])

regression_df.T
```

[11]:

	Alpha	Beta	RSquared	Treynor Ratio	\
Total Index	0.0035	0.2839	0.6169	0.0177	
Convertible Arbitrage	0.0029	0.1667	0.3869	0.0229	
Dedicated Short Bias	-0.0008	-0.8665	-0.8198	0.0063	
Emerging Markets	0.0014	0.5320	0.5809	0.0081	
Equity Market Neutral	0.0010	0.1894	0.2959	0.0109	
Event Driven	0.0036	0.2606	0.6666	0.0194	
Event Driven Distressed	0.0045	0.2618	0.6417	0.0225	
Event Driven Multi-Strategy	0.0032	0.2633	0.6203	0.0177	
Event Driven Risk Arbitrage	0.0020	0.1447	0.5650	0.0194	
Fixed Income Arbitrage	0.0016	0.1203	0.3371	0.0188	
Global Macro	0.0069	0.1407	0.2386	0.0546	
Long/Short Equity	0.0030	0.4561	0.7431	0.0121	
Managed Futures	0.0028	-0.0767	-0.1042	-0.0312	
Multi-Strategy	0.0033	0.1365	0.4069	0.0296	

	Information Ratio
Total Index	0.2101
Convertible Arbitrage	0.1598
Dedicated Short Bias	-0.0273
Emerging Markets	0.0409
Equity Market Neutral	0.0368
Event Driven	0.2724
Event Driven Distressed	0.3120
Event Driven Multi-Strategy	0.2115
Event Driven Risk Arbitrage	0.2084
Fixed Income Arbitrage	0.1044

```
Global Macro                                  0.2638
Long/Short Equity                             0.1611
Managed Futures                               0.0838
Multi-Strategy                                0.2347
```

##

3. Optional: Re-run this for every hedge-fund index, not just the "Total Index".
   Using our results from 3:

a) The highest Sharpe ratio hedge fund type was Event Driven Distressed, with a Sharpe of 1.09

b) The highest Treynor ratio hedge fund type was Global Macro, with a Treynor ratio of .0546

c) The equity market neutral hedge fund had the most desirable tail-risk profile with a kurt of 163, meaning that the distrobution of returns were very concentrated around the mean return of that asset.

1 Sources

https://www.w3schools.com/html/html_styles.asp

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/sup

https://csrgxtu.github.io/2015/03/20/Writing-Mathematic-Fomulars-in-Markdown/

https://www.w3schools.com/python/python_ml_linear_regression.asp

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.io.formats.style.Styler.set_properties.html
Lecture slides and notes from class