Stock_Alpha_Beta

September 29, 2021

1 Stock Alpha & Beta

Alpha is a measurement of performance. A positive alpha of 1.0 means the fund or stock has outperformed its benchmark index by 1 percent. A negative alpha of 1.0 would indicate an underperformance of 1 percent.

Beta is a measurement of volatile. A beta of less than 1 means that the security will be less volatile than the market.

[1]:

```
# Importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")

# fix_yahoo_finance is used to fetch data
import fix_yahoo_finance as yf
yf.pdr_override()
```

[2]:

```
# input
ticker = "AMD"
spx = "^GSPC"
start = '2014-01-01'
end = '2019-01-01'

# Read data
stock = yf.download(ticker,start,end)
market = yf.download(spx, start, end)
```

```
[*********************100%***********************]  1 of 1 downloaded
[*********************100%***********************]  1 of 1 downloaded
```

[3]:

```
# View columns
stock.head()
```

[3]:
```
            Open  High   Low  Close  Adj Close    Volume
Date
2014-01-02  3.85  3.98  3.84   3.95       3.95  20548400
2014-01-03  3.98  4.00  3.88   4.00       4.00  22887200
2014-01-06  4.01  4.18  3.99   4.13       4.13  42398300
2014-01-07  4.19  4.25  4.11   4.18       4.18  42932100
2014-01-08  4.23  4.26  4.14   4.18       4.18  30678700
```

[4]:
```
# View columns
market.head()
```

[4]:
```
                   Open         High          Low        Close    Adj Close  \
Date
2014-01-02  1845.859985  1845.859985  1827.739990  1831.979980  1831.979980
2014-01-03  1833.209961  1838.239990  1829.130005  1831.369995  1831.369995
2014-01-06  1832.310059  1837.160034  1823.729980  1826.770020  1826.770020
2014-01-07  1828.709961  1840.099976  1828.709961  1837.880005  1837.880005
2014-01-08  1837.900024  1840.020020  1831.400024  1837.489990  1837.489990

                Volume
Date
2014-01-02 -1214367296
2014-01-03 -1520697296
2014-01-06 -1000117296
2014-01-07  -783217296
2014-01-08  -642827296
```

[5]:
```
prices = stock['Adj Close']
values = market['Adj Close']
```

[6]:
```
#ret = prices.pct_change(1)[1:]
#ret = np.log(prices/prices.shift(1))
ret = (np.log(prices) - np.log(prices.shift(1))).dropna()
```

[7]:
```
ret.head()
```

[7]:
```
Date
2014-01-03    0.012579
2014-01-06    0.031983
2014-01-07    0.012034
2014-01-08    0.000000
2014-01-09   -0.021766
Name: Adj Close, dtype: float64
```

[8]:
```
mrk = values.pct_change(1).dropna()
```

[9]:
```
mrk.head()
```

```
[9]: Date
     2014-01-03   -0.000333
     2014-01-06   -0.002512
     2014-01-07    0.006082
     2014-01-08   -0.000212
     2014-01-09    0.000348
     Name: Adj Close, dtype: float64
```

```
[10]: from scipy import stats

      beta, alpha, r_value, p_value, std_err = stats.linregress(ret, mrk)
```

```
[11]: print("Beta:                         %9.6f" % beta)
      print("Alpha:                         %9.6f" % alpha)
      print("R-Squared:                 %9.6f" % r_value)
      print("p-value:                 %9.6f" % p_value)
      print("Standard Error:         %9.6f" % std_err)
```

```
Beta:                    0.076480
Alpha:                   0.000190
R-Squared:               0.352070
p-value:                 0.000000
Standard Error:          0.005739
```