Deviations From Normality

In [29]:
```
# Import needed modules
%load_ext autoreload
%autoreload 2

import pandas as pd
import edhex_risk_kit as erk
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [4]:
```
hfi = erk.get_hifi_returns()
```

In [5]:
```
hfi.head()
```

Out[5]:

	Convertible Arbitrage	CTA Global	Distressed Securities	Emerging Markets	Equity Market Neutral	Event Driven	Fixed Income Arbitrage	Global Macro	Long/Short Equity	Merger Arbitrage
date										
1997-01	0.0119	0.0393	0.0178	0.0791	0.0189	0.0213	0.0191	0.0573	0.0281	0.0150
1997-02	0.0123	0.0298	0.0122	0.0525	0.0101	0.0084	0.0122	0.0175	-0.0006	0.0034
1997-03	0.0078	-0.0021	-0.0012	-0.0120	0.0016	-0.0023	0.0109	-0.0119	-0.0084	0.0060
1997-04	0.0086	-0.0170	0.0030	0.0119	0.0119	-0.0005	0.0130	0.0172	0.0084	-0.0001
1997-05	0.0156	-0.0015	0.0233	0.0315	0.0189	0.0346	0.0118	0.0108	0.0394	0.0197

In [6]:
```
pd.concat([hfi.mean(), hfi.median(), hfi.mean()>hfi.median()], axis = 'columns')
```

Out[6]:

	0	1	2
Convertible Arbitrage	0.005508	0.0065	False
CTA Global	0.004074	0.0014	True
Distressed Securities	0.006946	0.0089	False
Emerging Markets	0.006253	0.0096	False
Equity Market Neutral	0.004498	0.0051	False
Event Driven	0.006344	0.0084	False
Fixed Income Arbitrage	0.004365	0.0055	False
Global Macro	0.005403	0.0038	True
Long/Short Equity	0.006331	0.0079	False
Merger Arbitrage	0.005356	0.0060	False
Relative Value	0.005792	0.0067	False
Short Selling	-0.001701	-0.0053	True
Funds Of Funds	0.004262	0.0052	False

$$S(R) = \frac{E[(R - E(R))^3]}{\sigma_R^3}$$

```
In [8]: erk.skewness(hfi).sort_values()
```

```
Out[8]: Fixed Income Arbitrage   -3.940320
        Convertible Arbitrage    -2.639592
        Equity Market Neutral    -2.124435
        Relative Value           -1.815470
        Event Driven             -1.409154
        Merger Arbitrage         -1.320083
        Distressed Securities    -1.300842
        Emerging Markets         -1.167067
        Long/Short Equity        -0.390227
        Funds Of Funds           -0.361783
        CTA Global                0.173699
        Short Selling             0.767975
        Global Macro              0.982922
        dtype: float64
```

```
In [11]: import scipy.stats
         scipy.stats.skew(hfi)
```

```
Out[11]: array([-2.63959223,  0.17369864, -1.30084204, -1.16706749, -2.12443538,
                -1.40915356, -3.94032029,  0.98292188, -0.39022677, -1.32008333,
                -1.81546975,  0.76797484, -0.36178308])
```

```
In [13]: #see if they match up
         erk.skewness(hfi)
```

```
Out[13]: Convertible Arbitrage    -2.639592
         CTA Global                0.173699
         Distressed Securities    -1.300842
         Emerging Markets         -1.167067
         Equity Market Neutral    -2.124435
         Event Driven             -1.409154
         Fixed Income Arbitrage   -3.940320
         Global Macro              0.982922
         Long/Short Equity        -0.390227
         Merger Arbitrage         -1.320083
         Relative Value           -1.815470
         Short Selling             0.767975
         Funds Of Funds           -0.361783
         dtype: float64
```

module skewness returns same as scipy

```
In [15]: import numpy as np
         normal_rets = np.random.normal(0, .15, size=(263,1))
```

```
In [18]: erk.skewness(normal_rets)
```

```
Out[18]: 0.03483464934841344
```

Kurtosis

$$K(R) = \frac{E[(R - E(R))^4]}{\sigma_R^4}$$

In [21]:
```
erk.kurtosis(normal_rets)
```

Out[21]:
```
3.47533504118101
```

In [22]:
```
erk.kurtosis(hfi)
```

Out[22]:
```
Convertible Arbitrage     23.280834
CTA Global                 2.952960
Distressed Securities      7.889983
Emerging Markets           9.250788
Equity Market Neutral     17.218555
Event Driven               8.035828
Fixed Income Arbitrage    29.842199
Global Macro               5.741679
Long/Short Equity          4.523893
Merger Arbitrage           8.738950
Relative Value            12.121208
Short Selling              6.117772
Funds Of Funds             7.070153
dtype: float64
```

In [23]:
```
scipy.stats.kurtosis(normal_rets)
```

Out[23]:
```
array([0.47533504])
```

In [24]:
```
scipy.stats.jarque_bera(normal_rets)
```

Out[24]:
```
(2.5291527875869226, 0.28235887978449625)
```

In [26]:
```
'''
If you run it this way it will look at the data set as one whole
and will not give you a distributed value for each fund
'''
scipy.stats.jarque_bera(hfi)
```

Out[26]:
```
(25656.585999171326, 0.0)
```

In [30]:
```
erk.is_normal(normal_rets)
```

Out[30]:
```
True
```

```
In [31]: # To apply is_normal to each column
         hfi.aggregate(erk.is_normal)
```

```
Out[31]: Convertible Arbitrage     False
         CTA Global                 True
         Distressed Securities     False
         Emerging Markets          False
         Equity Market Neutral     False
         Event Driven              False
         Fixed Income Arbitrage    False
         Global Macro              False
         Long/Short Equity         False
         Merger Arbitrage          False
         Relative Value            False
         Short Selling             False
         Funds Of Funds            False
         dtype: bool
```

```
In [32]: ffme = erk.get_ffme_returns()
         erk.skewness(ffme)
```

```
Out[32]: SmallCap    4.410739
         LargeCap    0.233445
         dtype: float64
```

```
In [33]: erk.kurtosis(ffme)
```

```
Out[33]: SmallCap    46.845008
         LargeCap    10.694654
         dtype: float64
```

```
In [34]: ffme.aggregate(erk.is_normal)
```

```
Out[34]: SmallCap    False
         LargeCap    False
         dtype: bool
```

```
In [ ]:
```