Measuring Risks and Returns,volatility and Returns for some Finnish Companies in the OXM H Index.Finally,a Portfolio Allocation for these companies in relation to Returns and Volatility.

In [1]:

```
import numpy as np
import pandas as pd
import pandas_datareader.data as web
import datetime
import matplotlib.pyplot as plt
%matplotlib inline
```

```
C:\Users\chumj\Anaconda3\Ben\lib\site-packages\pandas_datareader\compat\__init__.py:7:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the public API at
pandas.testing instead.
  from pandas.util.testing import assert_frame_equal
```

In [27]:

```
start=datetime.datetime (2013,1,1)
end=datetime.datetime (2020,9,2)
```

In [28]:

```
Cargotec=web.DataReader('CGCBV.HE','yahoo',start,end)
Kone=web.DataReader('KNEBV.HE','yahoo',start,end)
Konecrane=web.DataReader('KCR.HE','yahoo',start,end)
Metso=web.DataReader('METSO.HE','yahoo',start,end)
Wartsila=web.DataReader('WRT1V.HE','yahoo',start,end)
FORTUM=web.DataReader('FORTUM.HE','yahoo',start,end)
```

In [29]:

```
stock_comp=(Cargotec['Close'],Kone['Close'],Konecrane['Close'],Metso['Close'],Wartsila['Close'])
stocks=pd.concat(stock_comp,axis=1)
stocks.columns=['Cargotec','Kone','Konecrane','Metso','Wartsila']
```

In [30]:

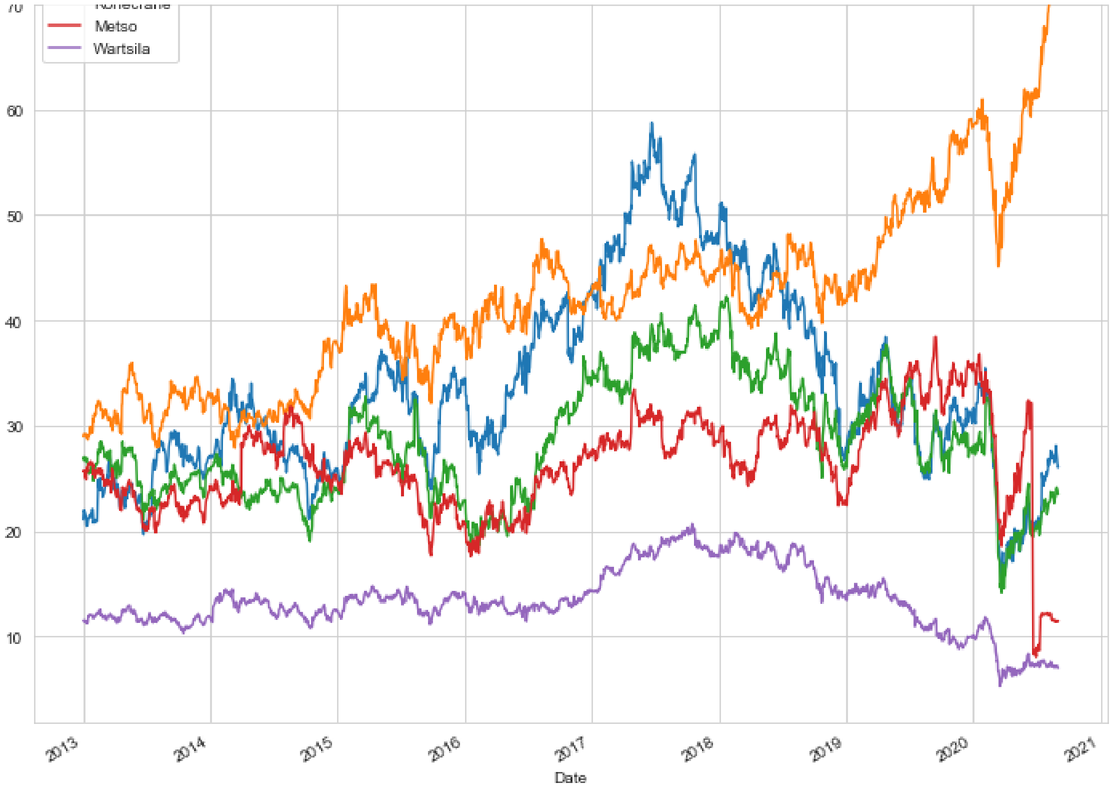
```
stocks.head()
```

Out[30]:

	Cargotec	Kone	Konecrane	Metso	Wartsila
Date					
2013-01-02	21.100000	29.000000	26.780001	25.655199	11.4667
2013-01-03	21.580000	29.075001	26.900000	25.762699	11.4733
2013-01-04	21.980000	29.200001	27.030001	25.770399	11.5133
2013-01-07	21.670000	29.200001	26.820000	25.555401	11.4733
2013-01-08	21.459999	29.200001	26.629999	25.140699	11.3767

In [31]:

```
stocks.plot(figsize=(12,10));
```



In [32]:

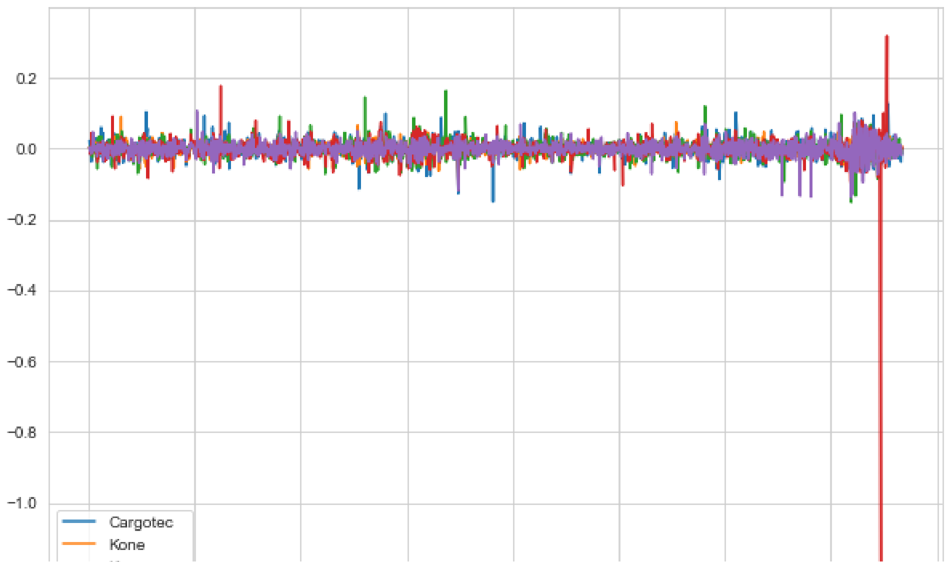
```
log_ret=np.log(1+stocks.pct_change())
log_ret.head()
```

Out[32]:

	Cargotec	Kone	Konecrane	Metso	Wartsila
Date					
2013-01-02	NaN	NaN	NaN	NaN	NaN
2013-01-03	0.022494	0.002583	0.004471	0.004181	0.000575
2013-01-04	0.018366	0.004290	0.004821	0.000299	0.003480
2013-01-07	-0.014204	0.000000	-0.007800	-0.008378	-0.003480
2013-01-08	-0.009738	0.000000	-0.007109	-0.016361	-0.008455

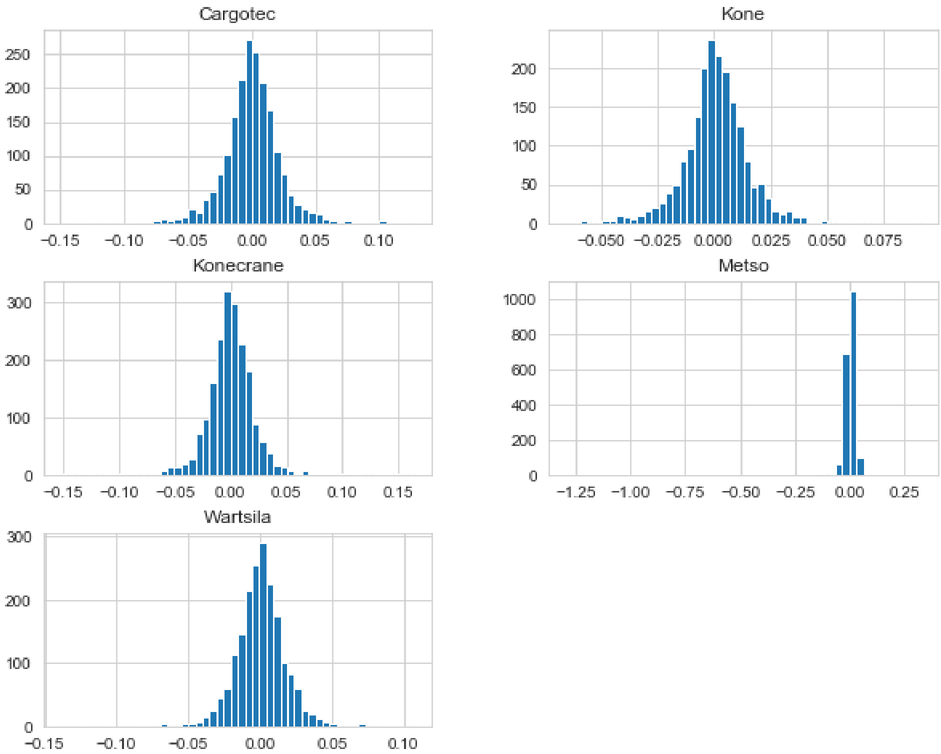
In [33]:

```
log_ret.plot(figsize=(10,8));
```



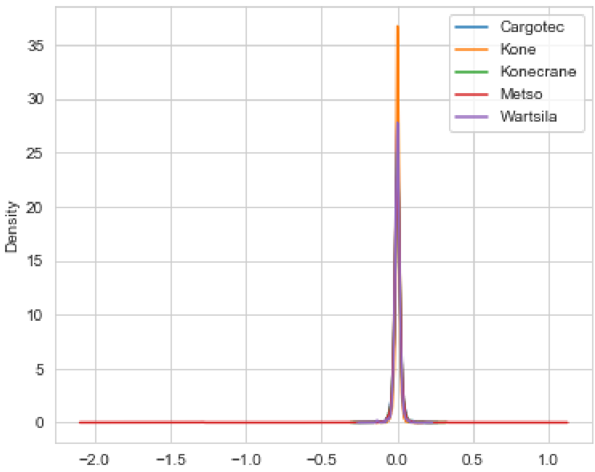
In [34]:

```
log_ret.hist(bins=50,figsize=(10,8));
```



In [35]:

```
log_ret.plot(kind='kde',figsize=(6,5));
```



In [36]:

```
log_ret.mean()
```

Out[36]:

```
Cargotec     0.000111
Kone         0.000480
Konecrane   -0.000068
Metso       -0.000418
```

```
Wartsila     -0.000258
dtype: float64
```

In [37]:

```python
log_ret.std()
```

Out[37]:

```
Cargotec     0.022190
Kone         0.014277
Konecrane    0.021134
Metso        0.036473
Wartsila     0.019183
dtype: float64
```

In [38]:

```python
log_ret.corr()
```

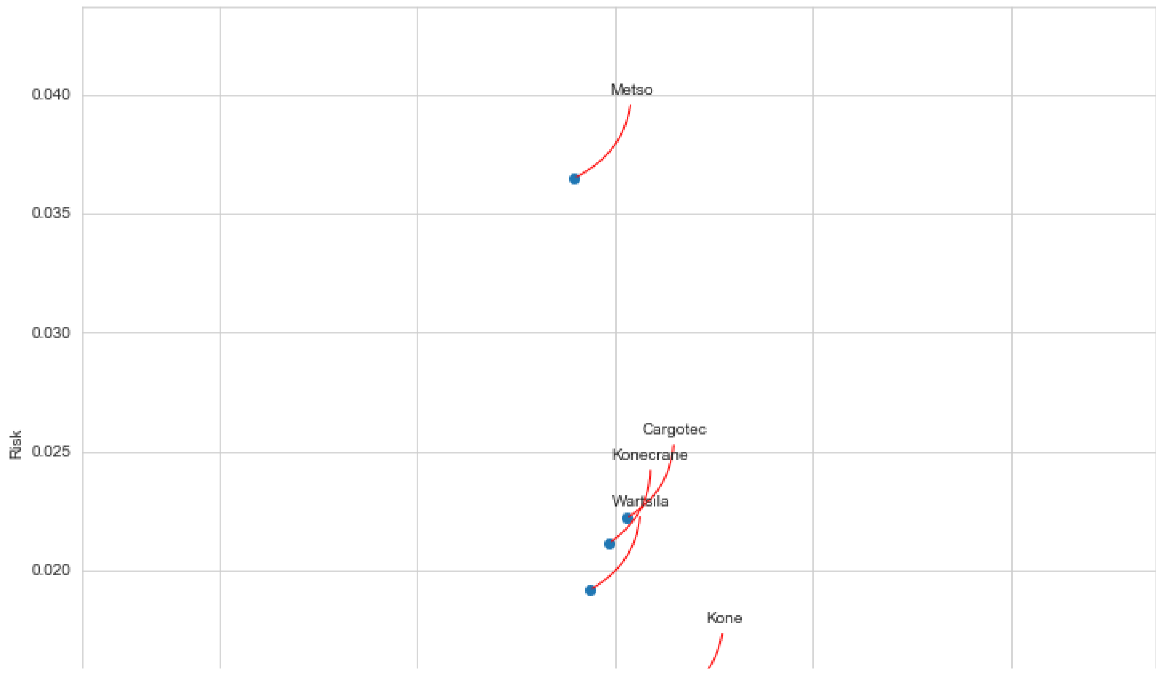
Out[38]:

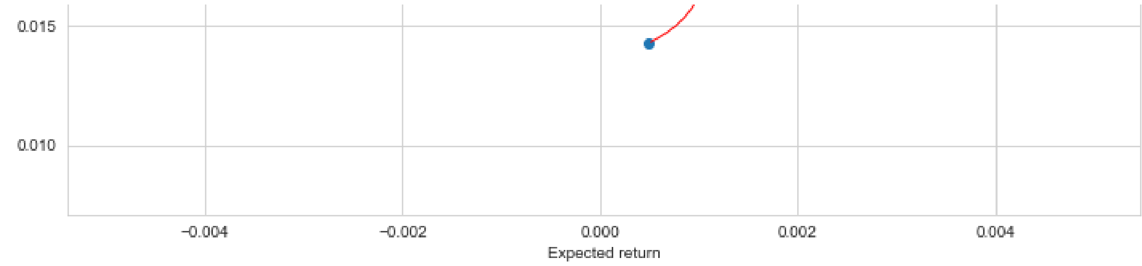
	Cargotec	Kone	Konecrane	Metso	Wartsila
Cargotec	1.000000	0.396663	0.560257	0.323892	0.523833
Kone	0.396663	1.000000	0.412066	0.284650	0.439006
Konecrane	0.560257	0.412066	1.000000	0.323901	0.532132
Metso	0.323892	0.284650	0.323901	1.000000	0.342962
Wartsila	0.523833	0.439006	0.532132	0.342962	1.000000

How much value do we put at risk by investing in a particular stock

In [39]:

```python
import seaborn as sns
sns.set_style('whitegrid')
plt.figure(figsize=(12, 10))
plt.scatter(log_ret.mean(), log_ret.std())
plt.xlabel('Expected return')
plt.ylabel('Risk')
for label, x, y in zip(log_ret.columns, log_ret.mean(),log_ret.std()):
    plt.annotate(label, xy=(x, y), xytext=(50, 50), textcoords='offset points', ha='right', va='bottom',
                 arrowprops=dict(arrowstyle='-', color='red', connectionstyle='arc3,rad=-0.3'))
```





In [40]:

```
import seaborn as sns
```

In [41]:

```
sns.heatmap(log_ret.corr(),annot=True)
```

Out[41]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f81ba38388>
```

In [42]:

```
log_ret.cov()*252
```

Out[42]:

	Cargotec	Kone	Konecrane	Metso	Wartsila
Cargotec	0.124081	0.031668	0.066209	0.066059	0.056191
Kone	0.031668	0.051368	0.031332	0.037354	0.030300
Konecrane	0.066209	0.031332	0.112551	0.062916	0.054364
Metso	0.066059	0.037354	0.062916	0.335239	0.060471
Wartsila	0.056191	0.030300	0.054364	0.060471	0.092735

In [43]:

```
np.random.seed(101)
num_ports=7000
all_weights=np.zeros((num_ports,len(stocks.columns)))
ret_arr=np.zeros(num_ports)
vol_arr=np.zeros(num_ports)
sharpe_arr=np.zeros(num_ports)
for ind in range(num_ports):
    #weights
    weights=np.array(np.random.random(5))
    weights=weights/np.sum(weights)
    #save weights
    all_weights[ind,:]=weights

    #expected returns
```

```
    ret_arr[ind]=np.sum((log_ret.mean() *weights) *252)

    #expected volatility

    vol_arr[ind]=np.sqrt(np.dot(weights.T,np.dot(log_ret.cov()*252,weights)))

    #Sharpe Ratio

    sharpe_arr[ind]=ret_arr[ind]/vol_arr[ind]
```

In [44]:

```
sharpe_arr.max()
```

Out[44]:

```
0.39045287768432957
```

In [45]:

```
sharpe_arr.argmax()
```

Out[45]:

```
2505
```

In [46]:

```
all_weights[2505,:]
```

Out[46]:

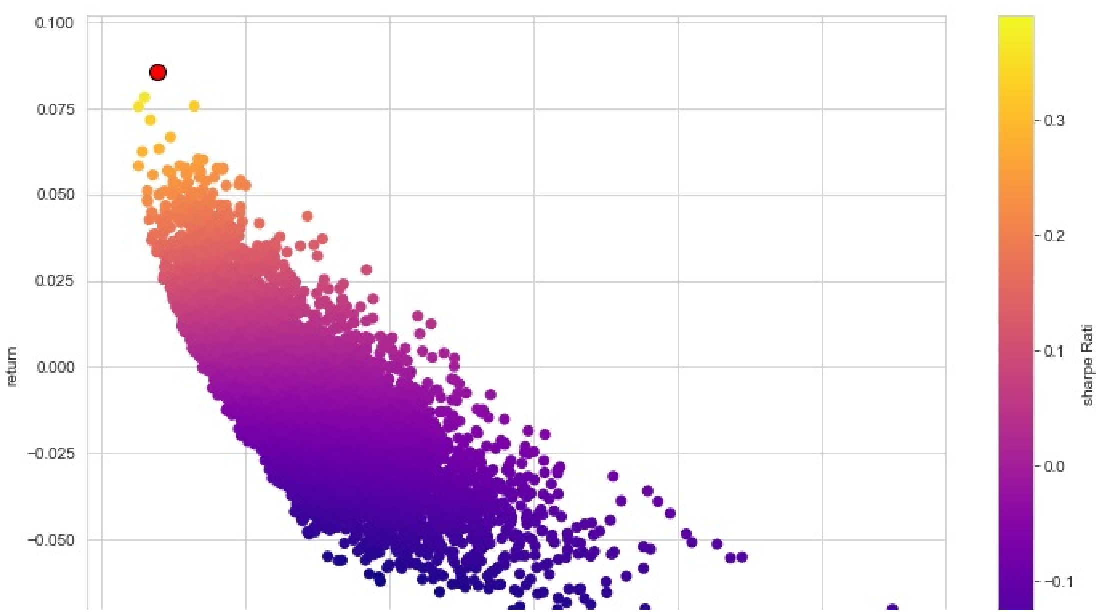
```
array([0.09376358, 0.77622538, 0.02125425, 0.08300423, 0.02575255])
```

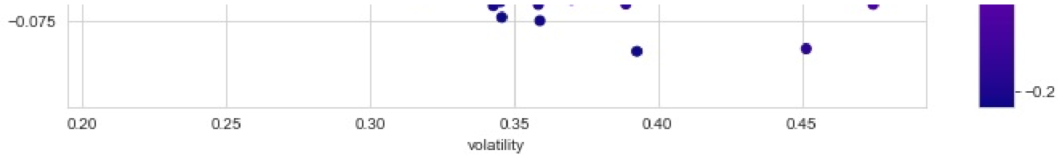
In [47]:

```
max_sr_ret=ret_arr[2505]
max_sr_vol=vol_arr[2505]
```

In [48]:

```
plt.figure(figsize=(12,8))
plt.scatter(vol_arr,ret_arr,c=sharpe_arr,cmap='plasma')
plt.colorbar(label='sharpe Rati')
plt.xlabel('volatility')
plt.ylabel('return')
plt.scatter(max_sr_vol,max_sr_ret,c='red',s=100,edgecolors='black');
```





In [51]:

```
def get_ret_vol_sr(weights):
    weights=np.array(weights)
    ret=np.sum((log_ret.mean()*weights)*252)
    vol=np.sqrt(np.dot(weights.T,np.dot(log_ret.cov()*252,weights)))
    sr=ret/vol
```

In [ ]:

```

```