In [1]:

```
import pandas as pd
import numpy as np
from gurobipy import *
import math
import os
import matplotlib.pyplot as plt
```

In [3]:

```
table = pd.read_csv("C:/Users/NGDRS-1/Downloads/Cryptocurrency.csv", encoding = "ISO-8859-1
table['Date'] = table.Date.apply(lambda x: pd.to_datetime(x).strftime('%d/%m/%Y'))
#table = table.iloc[::-1]
table = table.set_index('Date')
l = list(a for a in range(50))
data = table.iloc[:,l]
table = data.loc['21/11/2018':'21/11/2020']
```

In [49]:

```
ret.head()
```

Out[49]:

Date	Tether INR (USDT-INR)	Bitcoin INR (BTC-INR)	Ethereum INR (ETH-INR)	XRP INR (XRP-INR)	Litecoin INR (LTC-INR)	EOS INR (EOS-INR)	BitcoinCash INR (BCH-INR)	Chain (LI I
22/11/2018	0.006090	-0.055768	-0.077452	-0.052640	-0.075745	0.010482	-0.021449	-0.041
23/11/2018	-0.017989	-0.006683	-0.029229	-0.046052	-0.003477	-0.095445	-0.130136	-0.137
24/11/2018	-0.003804	-0.108546	-0.080807	-0.080076	-0.092771	0.028224	0.022580	-0.087
25/11/2018	0.001763	0.034331	0.027074	-0.001091	0.058371	-0.047298	-0.011338	0.096
26/11/2018	-0.001346	-0.055246	-0.067396	-0.048658	-0.045604	-0.047238	-0.015323	0.014

5 rows × 51 columns

In [93]:

```
a=table.pct_change().agg(["mean","std"]).T
```

In [87]:

```
a.columns=["Return","Risk"]
```

In [88]:

```
a
```

Out[88]:

	Return	Risk
Tether INR (USDT-INR)	0.000093	0.006400
Bitcoin INR (BTC-INR)	0.002700	0.037500
Ethereum INR (ETH-INR)	0.003070	0.046867
XRP INR (XRP-INR)	0.000692	0.039764
Litecoin INR (LTC-INR)	0.002477	0.049170
EOS INR (EOS-INR)	0.001220	0.051649
BitcoinCash INR (BCH-INR)	0.002161	0.058518
Chainlink INR (LINK-INR)	0.007516	0.068990
TRON INR (TRX-INR)	0.002412	0.051558
Cardano INR (ADA-INR)	0.002832	0.052029
EthereumClassic INR (ETC-INR)	0.001528	0.049100
NEO INR (NEO-INR)	0.002465	0.052775
Monero INR (XMR-INR)	0.002008	0.045884
Zcash INR (ZEC-INR)	0.001198	0.051197
Stellar INR (XLM-INR)	0.000305	0.046140
OmiseGO INR (OMG-INR)	0.003292	0.068367
Dash INR (DASH-INR)	0.001222	0.052854
Qtum INR (QTUM-INR)	0.001769	0.056198
BinanceCoin INR (BNB-INR)	0.003446	0.046605
BasicAttentionToken INR (BAT-INR)	0.001762	0.052268
VeChain INR (VET-INR)	0.003128	0.059382
Dogecoin INR (DOGE-INR)	0.001365	0.043327
Waves INR (WAVES-INR)	0.004315	0.062484
MCO INR (MCO-INR)	0.002677	0.066166
Bancor INR (BNT-INR)	0.002681	0.065048
0x INR (ZRX-INR)	0.001494	0.055752
Status INR (SNT-INR)	0.002199	0.059245
IOTA INR (MIOTA-INR)	0.001329	0.051377
KyberNetwork INR (KNC-INR)	0.004401	0.065727
NEM INR (XEM-INR)	0.002117	0.053080
Aragon INR (ANT-INR)	0.005854	0.070997
Civic INR (CVC-INR)	0.003145	0.082051
BitShares INR (BTS-INR)	0.000699	0.062248
TenX INR (PAY-INR)	-0.000098	0.067281
ICON INR (ICX-INR)	0.002360	0.060674

	Return	Risk
CloLoopring INR (LRC-INR)se	0.004207	0.067609
CloseStorj INR (STORJ-INR)	0.004169	0.087522
CHyperCash INR (HC-INR)lose	0.002432	0.069503
DigiByte INR (DGB-INR)	0.003060	0.067179
CloNano INR (NANO-INR)se	0.001338	0.054219
MonaCoin INR (MONA-INR)	0.003055	0.078711
BitcoinGold INR (BTG-INR)ose	0.000707	0.062492
CAugur INR (REP-INR)lose	0.002455	0.061960
Horizen INR (ZEN-INR)	0.001593	0.055168
Aeternity INR (AE-INR)	-0.000871	0.051672
Decred INR (DCR-INR)	0.001113	0.048351
district0x INR (DNT-INR	0.005374	0.096577
Golem INR (GNT-INR)	0.002008	0.061434
AdEx INR (ADX-INR)	0.005166	0.088444
Zcoin INR (XZC-INR)	0.000799	0.057521

In [89]:

```
a.Return=a.Return*250
a.Risk=a.Risk*np.sqrt(250)
```

In [91]:

```
a.Return.sum()
```

Out[91]:

```
29.609380684429368
```

In [92]:

```
a.Risk.sum()
```

Out[92]:

```
46.12237127411243
```

In [ ]:

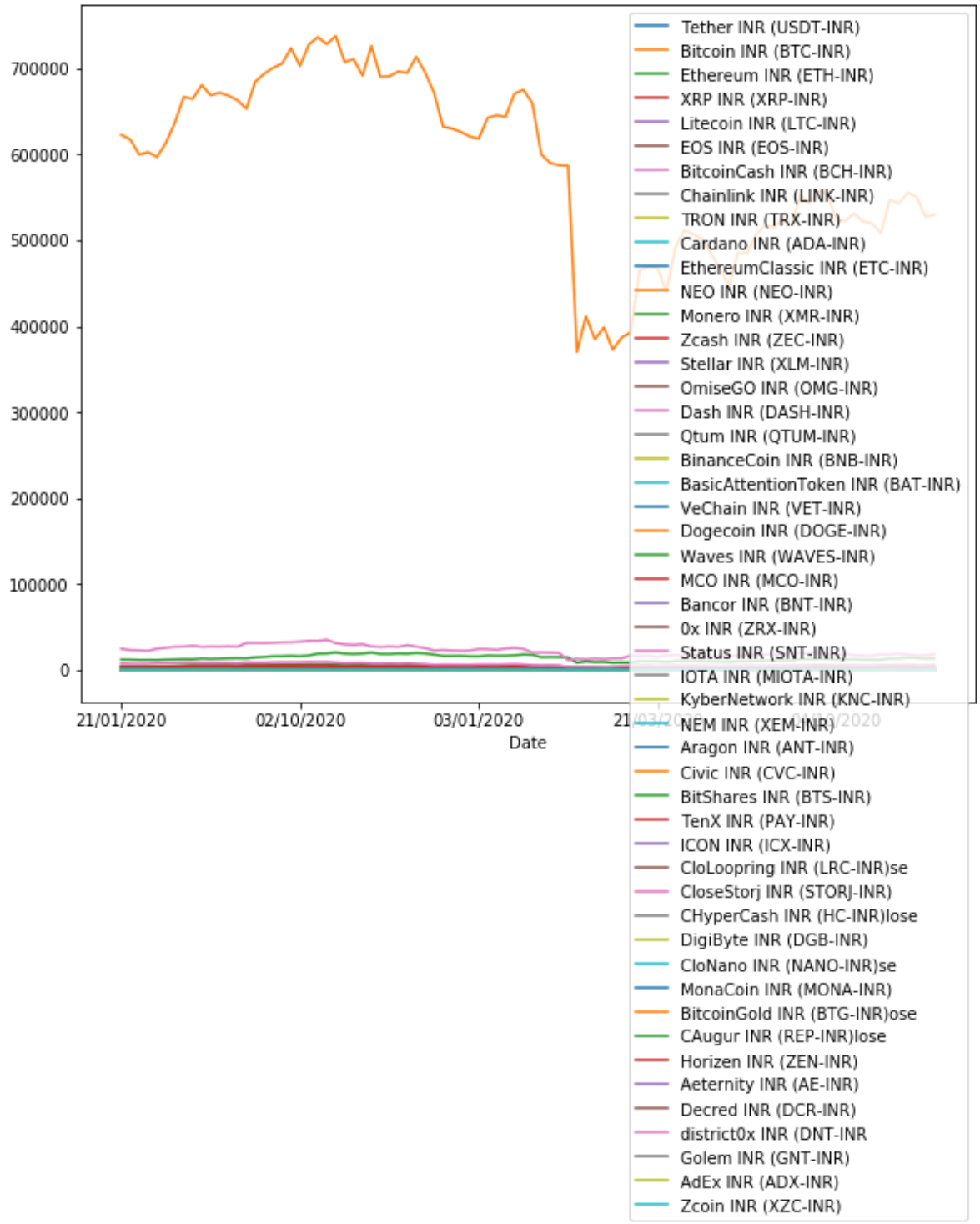
```
StockReturns['Portfolio_EW'] = table.iloc[:, 0:num].mul(port_eq_wt, axis=1).sum(axis=1)
```

In [56]:

```
tb1 = data.loc['21/01/2020':'21/04/2020']

tb1.plot(figsize=(10,8))
```

Out[56]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2aa6a34cac8>
```



In [4]:

```
returns_daily = table.pct_change()
for column in returns_daily:
    returns_daily[column] = returns_daily[column].mask(returns_daily[column]<-0.475, np.nan
avg = returns_daily.mean() * 250/4
cov_daily = returns_daily.cov()
cov = cov_daily * 250/4
#std_daily = returns_daily.std()
#std = std_daily * math.sqrt(125/2)
```

In [5]:

```
cov
```

Out[5]:

	Tether INR (USDT-INR)	Bitcoin INR (BTC-INR)	Ethereum INR (ETH-INR)	XRP INR (XRP-INR)	Litecoin INR (LTC-INR)	EOS INR (EOS-INR)	BitcoinCa INR (BC IN
Tether INR (USDT-INR)	0.002560	0.000221	-0.001099	-0.000795	-0.001064	0.002236	0.0024
Bitcoin INR (BTC-INR)	0.000221	0.087889	0.092119	0.069857	0.091607	-0.014602	-0.0081
Ethereum INR (ETH-INR)	-0.001099	0.092119	0.137285	0.097580	0.120859	-0.018141	-0.0140
XRP INR (XRP-INR)	-0.000795	0.069857	0.097580	0.098824	0.094819	-0.012312	-0.0071
Litecoin INR (LTC-INR)	-0.001064	0.091607	0.120859	0.094819	0.151106	-0.012133	-0.0069
EOS INR (EOS-INR)	0.002236	-0.014602	-0.018141	-0.012312	-0.012133	0.166724	0.1511
BitcoinCash INR (BCH-INR)	0.002437	-0.008134	-0.014019	-0.007128	-0.006939	0.151127	0.2140
Chainlink INR (LINK-INR)	0.003487	-0.013584	-0.019557	-0.014781	-0.023507	0.110914	0.1165
TRON INR (TRX-INR)	0.002331	-0.011799	-0.014777	-0.007960	-0.008939	0.115140	0.1197
Cardano INR (ADA-INR)	0.002939	-0.011453	-0.015970	-0.011234	-0.008846	0.132928	0.1406
EthereumClassic INR (ETC-INR)	0.001848	-0.014029	-0.018186	-0.009586	-0.010425	0.117801	0.1283
NEO INR (NEO-INR)	0.001690	-0.005491	-0.009565	-0.006632	-0.004904	0.128854	0.1314
Monero INR (XMR-INR)	0.001616	-0.013728	-0.017463	-0.010249	-0.011086	0.111705	0.1210
Zcash INR (ZEC-INR)	0.001221	-0.006161	-0.006563	-0.002155	-0.003109	0.113302	0.1264
Stellar INR (XLM-INR)	0.002577	-0.010112	-0.008435	0.000553	-0.002678	0.108360	0.1126
OmiseGO INR (OMG-INR)	0.002658	-0.016333	-0.020441	-0.013500	-0.015812	0.135114	0.1416
Dash INR (DASH-INR)	0.001671	-0.002781	-0.002599	-0.000167	0.002810	0.117836	0.1328
Qtum INR (QTUM-INR)	0.002957	-0.016106	-0.019531	-0.011206	-0.014945	0.137331	0.1382
BinanceCoin INR (BNB-INR)	0.001899	-0.012047	-0.014064	-0.008720	-0.007211	0.100918	0.1035
BasicAttentionToken INR (BAT-INR)	0.001628	-0.013310	-0.019522	-0.010701	-0.013412	0.103515	0.1047
VeChain INR (VET-INR)	0.002540	-0.011767	-0.015594	-0.009174	-0.010756	0.122525	0.1250
Dogecoin INR (DOGE-INR)	0.001004	-0.009376	-0.011694	-0.006795	-0.004295	0.080455	0.0831
Waves INR (WAVES-INR)	0.002454	-0.015386	-0.016814	-0.007373	-0.009818	0.087167	0.0926

	Tether INR (USDT-INR)	Bitcoin INR (BTC-INR)	Ethereum INR (ETH-INR)	XRP INR (XRP-INR)	Litecoin INR (LTC-INR)	EOS INR (EOS-INR)	BitcoinCa INR (BC IN
MCO INR (MCO-INR)	0.001440	-0.012413	-0.009098	-0.001615	-0.003190	0.117804	0.1172
Bancor INR (BNT-INR)	0.002101	-0.012159	-0.018071	-0.010464	-0.010266	0.118426	0.1227
0x INR (ZRX-INR)	0.001597	-0.013104	-0.018336	-0.010319	-0.015487	0.107393	0.1088
Status INR (SNT-INR)	0.001745	-0.013138	-0.015131	-0.005841	-0.009749	0.120105	0.1217
IOTA INR (MIOTA-INR)	0.000939	-0.009586	-0.011332	-0.004856	-0.006690	0.119401	0.1257
KyberNetwork INR (KNC-INR)	0.004088	-0.019952	-0.021672	-0.017129	-0.017699	0.109664	0.1163
NEM INR (XEM-INR)	0.000540	-0.012855	-0.013228	-0.003606	-0.008362	0.101119	0.1145
Aragon INR (ANT-INR)	0.002159	-0.015929	-0.018817	-0.009401	-0.016724	0.088054	0.0998
Civic INR (CVC-INR)	0.001494	-0.003777	-0.011080	-0.010172	-0.001846	0.109123	0.1104
BitShares INR (BTS-INR)	0.002186	-0.011607	-0.011595	-0.005743	-0.002386	0.116999	0.1208
TenX INR (PAY-INR)	0.001705	-0.011527	-0.010504	-0.001879	-0.004796	0.102391	0.1007
ICON INR (ICX-INR)	0.002100	-0.015330	-0.020334	-0.014484	-0.013225	0.115230	0.1184
CloLoopring INR (LRC-INR)se	0.002142	-0.012577	-0.014198	-0.008702	-0.007289	0.106880	0.1117
CloseStorj INR (STORJ-INR)	0.002183	-0.008673	-0.008246	-0.005549	-0.001557	0.107885	0.1084
CHyperCash INR (HC-INR)lose	0.002375	-0.028176	-0.034334	-0.021911	-0.026143	0.120995	0.1157
DigiByte INR (DGB-INR)	0.002546	-0.021888	-0.023633	-0.008911	-0.015182	0.124128	0.1219
CloNano INR (NANO-INR)se	0.001344	-0.018071	-0.019078	-0.010284	-0.017610	0.119062	0.1259
MonaCoin INR (MONA-INR)	0.001989	-0.010607	-0.011446	-0.007665	-0.003692	0.101731	0.1070
BitcoinGold INR (BTG-INR)ose	0.001931	-0.017061	-0.022109	-0.014307	-0.017037	0.131884	0.1513
CAugur INR (REP-INR)lose	0.001053	-0.009750	-0.013979	-0.007716	-0.007250	0.100093	0.1071
Horizen INR (ZEN-INR)	0.002843	-0.008558	-0.008942	-0.005692	-0.009949	0.095211	0.0971
Aeternity INR (AE-INR)	0.001769	-0.015403	-0.015778	-0.009530	-0.009009	0.103535	0.1039
Decred INR (DCR-INR)	0.001952	-0.008217	-0.011512	-0.004781	-0.005863	0.102313	0.1090
district0x INR (DNT-INR	0.002083	-0.013174	-0.022331	-0.018218	-0.017872	0.095175	0.1011
Golem INR (GNT-INR)	0.001156	-0.007753	-0.002633	-0.001843	0.001739	0.099981	0.1040
AdEx INR (ADX-INR)	0.001321	-0.010896	-0.009909	-0.006778	-0.001697	0.090597	0.0864
Zcoin INR (XZC-INR)	0.001742	-0.009993	-0.008832	-0.003856	-0.001660	0.113475	0.1215

50 rows × 50 columns

In [2]:

```
def ann_risk_return(table):
    summary = table.agg(["mean","std"]).T
    summary.columns = ["Returns","Risk"]
    summary.Returns =summary.Returns*250
    summary.Risk = summary.Risk*np.sqrt(250)
    return summary
```

In [6]:

```
ret=table.pct_change().dropna()
```

In [7]:

```
summary=ann_risk_return(table)
```

In [8]:

```
summary
```

Out[8]:

	Returns	Risk
Tether INR (USDT-INR)	1.808760e+04	3.531898e+01
Bitcoin INR (BTC-INR)	1.502994e+08	3.454705e+06
Ethereum INR (ETH-INR)	3.935196e+06	1.086140e+05
XRP INR (XRP-INR)	4.944417e+03	6.954660e+01
Litecoin INR (LTC-INR)	1.062760e+06	2.531721e+04
EOS INR (EOS-INR)	6.170370e+04	1.302602e+03
BitcoinCash INR (BCH-INR)	4.634583e+06	9.113084e+04
Chainlink INR (LINK-INR)	7.000335e+04	4.732502e+03
TRON INR (TRX-INR)	3.747614e+02	6.855268e+00
Cardano INR (ADA-INR)	1.169902e+03	3.517932e+01
EthereumClassic INR (ETC-INR)	1.070361e+05	1.878267e+03
NEO INR (NEO-INR)	2.010654e+05	4.247870e+03
Monero INR (XMR-INR)	1.276589e+06	2.464821e+04
Zcash INR (ZEC-INR)	1.015894e+06	1.957349e+04
Stellar INR (XLM-INR)	1.492471e+03	2.837448e+01
OmiseGO INR (OMG-INR)	2.908489e+04	1.154152e+03
Dash INR (DASH-INR)	1.579692e+06	3.008786e+04
Qtum INR (QTUM-INR)	4.084217e+04	7.535713e+02
BinanceCoin INR (BNB-INR)	3.478510e+05	8.602465e+03
BasicAttentionToken INR (BAT-INR)	4.018375e+03	7.492909e+01
VeChain INR (VET-INR)	1.291016e+02	4.820424e+00
Dogecoin INR (DOGE-INR)	4.640386e+01	5.291298e-01
Waves INR (WAVES-INR)	3.280851e+04	1.116315e+03
MCO INR (MCO-INR)	7.439299e+04	1.447921e+03
Bancor INR (BNT-INR)	1.143927e+04	5.324210e+02
0x INR (ZRX-INR)	5.368556e+03	1.127252e+02
Status INR (SNT-INR)	3.730898e+02	7.871938e+00
IOTA INR (MIOTA-INR)	4.905634e+03	7.579435e+01
KyberNetwork INR (KNC-INR)	9.239940e+03	5.456082e+02
NEM INR (XEM-INR)	1.099512e+03	3.096509e+01
Aragon INR (ANT-INR)	2.326349e+04	1.638700e+03
Civic INR (CVC-INR)	7.887204e+02	2.343866e+01
BitShares INR (BTS-INR)	6.067979e+02	1.646262e+01
TenX INR (PAY-INR)	2.327615e+03	1.135214e+02

	Returns	Risk
ICON INR (ICX-INR)	5.244927e+03	1.219007e+02
CloLoopring INR (LRC-INR)se	1.266350e+03	6.518131e+01
CloseStorj INR (STORJ-INR)	3.628791e+03	1.256959e+02
CHyperCash INR (HC-INR)lose	2.713848e+04	8.162368e+02
DigiByte INR (DGB-INR)	2.376770e+02	8.211902e+00
CloNano INR (NANO-INR)se	1.731991e+04	3.153817e+02
MonaCoin INR (MONA-INR)	2.304288e+04	6.183390e+02
BitcoinGold INR (BTG-INR)ose	2.156706e+05	6.174611e+03
CAugur INR (REP-INR)lose	2.454327e+05	4.937920e+03
Horizen INR (ZEN-INR)	1.236089e+05	2.547230e+03
Aeternity INR (AE-INR)	4.652470e+03	1.689715e+02
Decred INR (DCR-INR)	3.386082e+05	5.595454e+03
district0x INR (DNT-INR	1.904993e+02	9.275223e+00
Golem INR (GNT-INR)	1.174392e+03	2.662620e+01
AdEx INR (ADX-INR)	2.172340e+03	5.856794e+01
Zcoin INR (XZC-INR)	1.008935e+05	2.187667e+03

In [17]:

```
noa=50
nop=100000
```

In [18]:

```
np.random.seed(123)
matrix=np.random.random(noa*nop).reshape(nop,noa)
matrix
```

Out[18]:

```
array([[0.69646919, 0.28613933, 0.22685145, ..., 0.98555979, 0.51948512,
        0.61289453],
       [0.12062867, 0.8263408 , 0.60306013, ..., 0.39887629, 0.2408559 ,
        0.34345601],
       [0.51312815, 0.66662455, 0.10590849, ..., 0.04857903, 0.7086974 ,
        0.83924335],
       ...,
       [0.771363  , 0.66399452, 0.70980034, ..., 0.42080155, 0.18014488,
        0.02020186],
       [0.00555788, 0.05765405, 0.66167542, ..., 0.42020726, 0.05788854,
        0.58869437],
       [0.60038571, 0.05553236, 0.03331703, ..., 0.7533014 , 0.86757063,
        0.89091337]])
```

In [19]:

```
matrix.sum(axis=1,keepdims=True)
```

Out[19]:

```
array([[25.09289335],
       [25.04999465],
       [26.01579807],
       ...,
       [23.99961788],
       [23.50694082],
       [24.48699498]])
```

In [20]:

```
w=matrix/matrix.sum(axis=1,keepdims=True)
w.sum(axis=1,keepdims=True)
```

Out[20]:

```
array([[1.],
       [1.],
       [1.],
       ...,
       [1.],
       [1.],
       [1.]])
```

In [21]:

```
port_ret=ret.dot(w.T)
port_ret
```

Out[21]:

Date	0	1	2	3	4	5	6	7
22/11/2018	-0.033147	-0.035146	-0.033249	-0.033532	-0.031938	-0.034349	-0.032588	-0.03594
23/11/2018	-0.095726	-0.097346	-0.098610	-0.095275	-0.091410	-0.097405	-0.101046	-0.09223
24/11/2018	0.006797	0.004054	0.004369	0.003945	-0.000442	0.002626	0.009631	0.00146
25/11/2018	-0.037199	-0.040631	-0.044557	-0.044691	-0.041315	-0.038317	-0.043169	-0.04841
26/11/2018	0.015776	0.018201	0.023219	0.020915	0.016660	0.017891	0.023692	0.01935
...	...	...	...	...	...	...	...	...
17/11/2020	-0.013440	-0.005096	-0.015854	-0.003619	-0.004411	-0.006965	-0.010712	-0.01203
18/11/2020	-0.004803	-0.008487	-0.005921	0.000392	-0.005452	-0.008667	-0.004975	-0.00712
19/11/2020	0.057077	0.057509	0.049263	0.066310	0.061040	0.044003	0.048140	0.05486
20/11/2020	0.117998	0.122962	0.125830	0.125428	0.122961	0.120141	0.109942	0.12761
21/11/2020	-0.031837	-0.031326	-0.036606	-0.033889	-0.025319	-0.028580	-0.035684	-0.03818

731 rows × 100000 columns

In [22]:

```
port_summary=ann_risk_return(port_ret)
port_summary
```

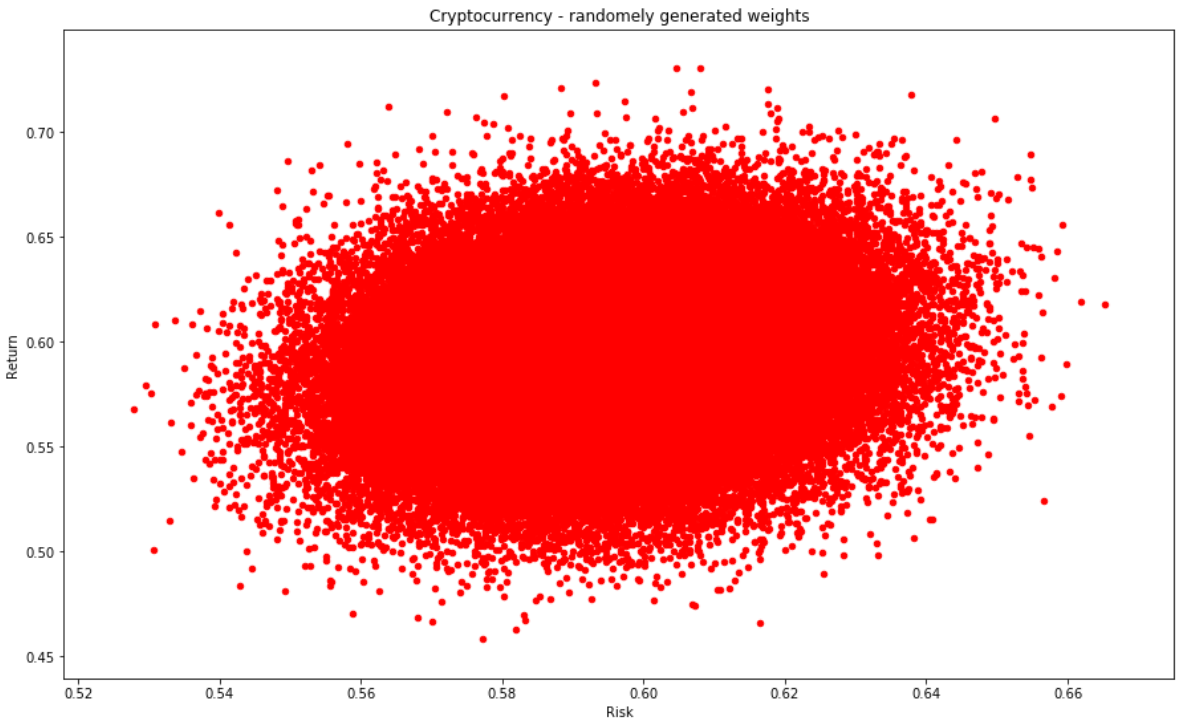
Out[22]:

	Returns	Risk
0	0.593940	0.594631
1	0.554334	0.577944
2	0.586833	0.623476
3	0.649346	0.603632
4	0.592277	0.578928
...	...	...
99995	0.625062	0.606713
99996	0.612650	0.580884
99997	0.532824	0.572634
99998	0.572540	0.630216
99999	0.615094	0.633344

100000 rows × 2 columns

In [28]:

```
plt.figure(figsize=(15,9))
plt.scatter(port_summary.loc[:,"Risk"], port_summary.loc[:,"Returns"], s=20, color="red")
plt.xlabel("Risk")
plt.ylabel("Return")
plt.title("Cryptocurrency - randomely generated weights")
plt.show()
```



Optimized Portfolio

In [29]:

```
model = Model('min_risk')
```

```
Using license file C:\Users\NGDRS-1\gurobi.lic
Academic license - for non-commercial use only - expires 2021-01-24
```

In [30]:

```
tickers = table.columns
variables = pd.Series(model.addVars(tickers),index=tickers)
```

In [31]:

```
port_risk = cov.dot(variables).dot(variables)
```

In [32]:

```
model.setObjective(port_risk,GRB.MINIMIZE)
```

In [33]:

```
model.addConstr(variables.sum() == 1,'weights')
model.update()
```

In [34]:

```
model.setParam('OutputFlag',0)
model.update()
```

In [35]:

```
model.optimize()
```

In [36]:

```
n = 0
weights = {}
for v in variables:
    weights.update({tickers[n]:v.x})
    n = n + 1
weights = pd.DataFrame([weights])
weights = weights.transpose()
weights.columns = ['Weights']

print('\nMin Risk, Optimal Weights Per Stock')
print(weights['Weights'])
```

```
Min Risk, Optimal Weights Per Stock
Tether INR (USDT-INR)                9.473253e-01
Bitcoin INR (BTC-INR)                1.695089e-08
Ethereum INR (ETH-INR)               9.791203e-03
XRP INR (XRP-INR)                    2.134613e-02
Litecoin INR (LTC-INR)               2.196463e-03
EOS INR (EOS-INR)                    2.837521e-09
BitcoinCash INR (BCH-INR)            1.991587e-09
Chainlink INR (LINK-INR)             1.534451e-09
TRON INR (TRX-INR)                   2.797294e-09
Cardano INR (ADA-INR)                1.655755e-09
EthereumClassic INR (ETC-INR)        4.841418e-09
NEO INR (NEO-INR)                    3.643384e-09
Monero INR (XMR-INR)                 1.327322e-08
Zcash INR (ZEC-INR)                  1.000854e-08
Stellar INR (XLM-INR)                1.945455e-09
OmiseGO INR (OMG-INR)                1.936626e-09
Dash INR (DASH-INR)                  3.748214e-09
Qtum INR (QTUM-INR)                  1.624386e-09
BinanceCoin INR (BNB-INR)            6.372570e-09
BasicAttentionToken INR (BAT-INR)    3.815447e-08
VeChain INR (VET-INR)                2.048120e-09
Dogecoin INR (DOGE-INR)              8.795022e-03
Waves INR (WAVES-INR)                3.844703e-09
MCO INR (MCO-INR)                    6.905247e-09
Bancor INR (BNT-INR)                 3.291302e-09
0x INR (ZRX-INR)                     8.258925e-09
Status INR (SNT-INR)                 3.592596e-09
IOTA INR (MIOTA-INR)                 1.418254e-03
KyberNetwork INR (KNC-INR)           1.250701e-09
NEM INR (XEM-INR)                    6.912305e-03
Aragon INR (ANT-INR)                 5.185938e-09
Civic INR (CVC-INR)                  1.757155e-08
BitShares INR (BTS-INR)              2.642486e-09
TenX INR (PAY-INR)                   4.942745e-09
ICON INR (ICX-INR)                   3.788558e-09
CloLoopring INR (LRC-INR)se          3.139335e-09
CloseStorj INR (STORJ-INR)           2.599769e-09
CHyperCash INR (HC-INR)lose          4.515287e-09
DigiByte INR (DGB-INR)               2.365910e-09
CloNano INR (NANO-INR)se             8.023998e-08
MonaCoin INR (MONA-INR)              4.591906e-09
BitcoinGold INR (BTG-INR)ose         5.467119e-09
CAugur INR (REP-INR)lose             1.943116e-03
Horizen INR (ZEN-INR)                2.245311e-09
Aeternity INR (AE-INR)               7.995945e-09
```

```
Decred INR (DCR-INR)                4.349105e-09
district0x INR (DNT-INR             1.026095e-08
Golem INR (GNT-INR)                 1.109661e-08
AdEx INR (ADX-INR)                  2.718539e-04
Zcoin INR (XZC-INR)                 4.412508e-09
Name: Weights, dtype: float64
```

In [37]:

```
print('\nMinimized Portfolio Variance : '+str(port_risk.getValue()))
```

```
Minimized Portfolio Variance : 0.0024115776381447764
```

In [38]:

```
min_vol = math.sqrt(port_risk.getValue())
print('Volatility : '+str(min_vol))
```

```
Volatility : 0.04910781646688006
```

In [39]:

```
port_return = avg.dot(variables)
Rmin = port_return.getValue()
print('Expected Return (Rmin) : '+str(Rmin))
```

```
Expected Return (Rmin) : 0.010835483146367978
```

In [40]:

```
Rmax = avg.max()
```

In [41]:

```
target = model.addConstr(port_return == Rmin, 'target')
```

In [42]:

```
eff = {}
iterations = 50
diff = (Rmax-Rmin)/(iterations-1)
Rrange = np.arange(Rmin,Rmax+diff,diff)
for r in Rrange:
    target.rhs = r
    model.optimize()
    temp = math.sqrt(port_risk.getValue())
    eff.update({temp:r})
```

In [43]:

```
frontier = pd.DataFrame([eff]).transpose()
frontier.columns = ['Returns']
frontier['Risk'] = frontier.index
frontier = frontier.reset_index(drop=True)
```

In [44]:

```
print('\nEfficient Frontier')
print(frontier)
```

```
Efficient Frontier
     Returns      Risk
0   0.010835  0.049108
1   0.020201  0.049801
2   0.029566  0.051588
3   0.038931  0.054310
4   0.048297  0.057877
5   0.057662  0.062180
6   0.067027  0.067077
7   0.076393  0.072448
8   0.085758  0.078196
9   0.095123  0.084241
10  0.104489  0.090522
11  0.113854  0.096994
12  0.123219  0.103621
13  0.132585  0.110375
14  0.141950  0.117235
15  0.151315  0.124182
16  0.160681  0.131204
17  0.170046  0.138288
18  0.179411  0.145426
19  0.188776  0.152611
20  0.198142  0.159835
21  0.207507  0.167094
22  0.216872  0.174384
23  0.226238  0.181700
24  0.235603  0.189040
25  0.244968  0.196401
26  0.254334  0.203780
27  0.263699  0.211177
28  0.273064  0.218588
29  0.282430  0.226013
30  0.291795  0.233451
31  0.301160  0.240924
32  0.310526  0.248960
33  0.319891  0.257768
34  0.329256  0.267302
35  0.338621  0.277531
36  0.347987  0.288485
37  0.357352  0.300136
38  0.366717  0.312440
39  0.376083  0.325708
40  0.385448  0.339930
41  0.394813  0.354993
42  0.404179  0.370794
43  0.413544  0.387243
44  0.422909  0.404261
45  0.432275  0.421778
46  0.441640  0.439746
47  0.451005  0.461081
48  0.460371  0.495043
49  0.469736  0.545417
```

In [45]:

```
frontier['Sharpe'] = frontier['Returns']/frontier['Risk']
idx = frontier['Sharpe'].max()
sharpeMax = frontier.loc[frontier['Sharpe'] == idx]
sharpeMax = sharpeMax.reset_index(drop=True)
```

In [46]:

```
target.rhs = sharpeMax['Returns'][0]
model.optimize()
n = 0
sharpe_weights = {}
for v in variables:
    sharpe_weights.update({tickers[n]:v.x})
    n = n + 1
sharpe_weights = pd.DataFrame([sharpe_weights])
sharpe_weights = sharpe_weights.transpose()
sharpe_weights.columns = ['Weights']
```

In [47]:

```
print('\nMaximum Sharpe Ratio')
print(sharpeMax)
print(sharpe_weights)
```

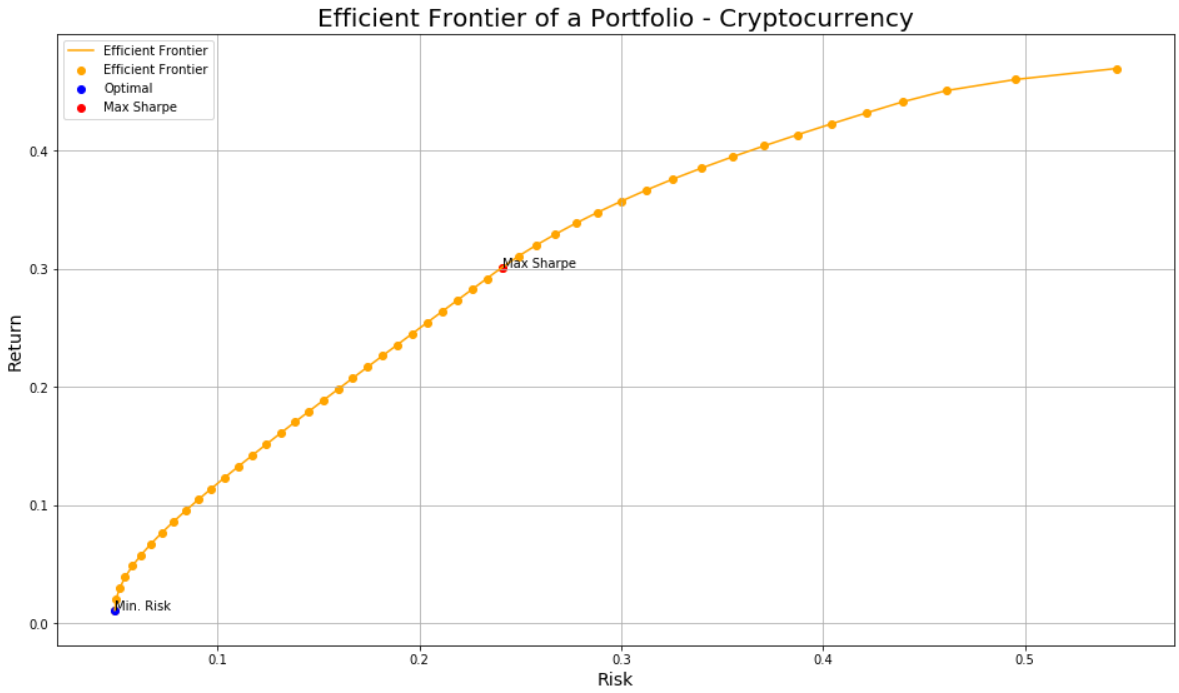
```
Maximum Sharpe Ratio
   Returns      Risk    Sharpe
0  0.30116  0.240924  1.250022
                                      Weights
Tether INR (USDT-INR)              2.959577e-08
Bitcoin INR (BTC-INR)              3.300820e-01
Ethereum INR (ETH-INR)             1.185570e-01
XRP INR (XRP-INR)                  3.691626e-11
Litecoin INR (LTC-INR)             1.162540e-10
EOS INR (EOS-INR)                  2.383807e-11
BitcoinCash INR (BCH-INR)          2.723621e-11
Chainlink INR (LINK-INR)           2.093759e-01
TRON INR (TRX-INR)                 7.248697e-11
Cardano INR (ADA-INR)              4.316684e-11
EthereumClassic INR (ETC-INR)      4.077590e-11
NEO INR (NEO-INR)                  3.051713e-11
Monero INR (XMR-INR)               4.998770e-11
Zcash INR (ZEC-INR)                2.001286e-11
Stellar INR (XLM-INR)              1.744048e-11
OmiseGO INR (OMG-INR)              5.033940e-11
Dash INR (DASH-INR)                2.019668e-11
Qtum INR (QTUM-INR)                2.168347e-11
BinanceCoin INR (BNB-INR)          2.032976e-08
BasicAttentionToken INR (BAT-INR)  3.627730e-11
VeChain INR (VET-INR)              3.995972e-11
Dogecoin INR (DOGE-INR)            9.667605e-11
Waves INR (WAVES-INR)              3.864578e-02
MCO INR (MCO-INR)                  1.004097e-10
Bancor INR (BNT-INR)               3.069768e-11
0x INR (ZRX-INR)                   2.459102e-11
Status INR (SNT-INR)               3.238882e-11
IOTA INR (MIOTA-INR)               2.065924e-11
KyberNetwork INR (KNC-INR)         2.558506e-02
NEM INR (XEM-INR)                  5.497104e-11
Aragon INR (ANT-INR)               1.758635e-01
Civic INR (CVC-INR)                2.601091e-11
BitShares INR (BTS-INR)            1.895863e-11
TenX INR (PAY-INR)                 1.502352e-11
ICON INR (ICX-INR)                 3.124182e-11
CloLoopring INR (LRC-INR)se        1.024755e-09
CloseStorj INR (STORJ-INR)         1.129860e-10
CHyperCash INR (HC-INR)lose        1.962072e-08
DigiByte INR (DGB-INR)             1.177742e-10
CloNano INR (NANO-INR)se           2.305884e-11
MonaCoin INR (MONA-INR)            6.135450e-03
BitcoinGold INR (BTG-INR)ose       2.353759e-11
CAugur INR (REP-INR)lose           1.205355e-10
Horizen INR (ZEN-INR)              3.715403e-11
Aeternity INR (AE-INR)             1.869567e-11
Decred INR (DCR-INR)               2.640217e-11
district0x INR (DNT-INR            4.148178e-02
Golem INR (GNT-INR)                3.752087e-11
AdEx INR (ADX-INR)                 5.427338e-02
Zcoin INR (XZC-INR)                1.804867e-11
```

In [48]:

```
fig, ax = plt.subplots(nrows=1,ncols=1)
fig.set_size_inches(16,9)
ax.set_title('Efficient Frontier of a Portfolio - Cryptocurrency',fontsize=20)
ax.set_xlabel('Risk',fontsize=14)
ax.set_ylabel('Return',fontsize=14)

ax.scatter(x=frontier['Risk'],y=frontier['Returns'],color='orange',label='Efficient Frontie
ax.plot()#x=frontier['Risk'],y=frontier['Returns'],color='orange')
temp = pd.DataFrame([eff]).transpose()
temp.columns = ['Efficient Frontier']
temp.plot(color='orange',label='Efficient Frontier',ax=ax)

i = 0

ax.scatter(x=min_vol,y=Rmin,color='blue',label='Optimal')
ax.annotate('Min. Risk',(min_vol,Rmin))

ax.scatter(x=sharpeMax['Risk'],y=sharpeMax['Returns'],color='red',label='Max Sharpe')
ax.annotate('Max Sharpe',(sharpeMax['Risk'],sharpeMax['Returns']))

ax.grid()
ax.legend(loc='upper left')
```

Out[48]:

```
<matplotlib.legend.Legend at 0x2aa7415abc8>
```



In [27]:

```
# Calculate mean returns for each stock
avg_rets = returns_daily.mean()

# Calculate mean returns for portfolio overall,
# using dot product to
# normalize individual means against investment weights
 # https://en.wikipedia.org/wiki/Dot_product#:~:targetText=In%20mathematics%2C%20the%20dot%
port_mean = avg_rets.dot(sharpe_weights)

# Calculate portfolio standard deviation
port_stdev = np.sqrt(sharpe_weights.T.dot(cov).dot(sharpe_weights))
```

In [29]:

```
initial_investment = 10000
# Calculate mean of investment
mean_investment = (1+port_mean) * initial_investment

#x+a*x=mean inv

# Calculate standard deviation of investmnet
stdev_investment = initial_investment * port_stdev
```

In [30]:

```
# Select our confidence interval (I'll choose 95% here)
conf_level1 = 0.05

# Using SciPy ppf method to generate values for the
# inverse cumulative distribution function to a normal distribution
# Plugging in the mean, standard deviation of our portfolio
# as calculated above
# https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.norm.html
from scipy.stats import norm
cutoff1 = norm.ppf(conf_level1, mean_investment, stdev_investment)
cutoff1
```

Out[30]:

```
array([[6085.33881067]])
```

In [31]:

```
#Finally, we can calculate the VaR at our confidence interval
var_1d1 = initial_investment - cutoff1
var_1d1
#output
```

Out[31]:

```
array([[3914.66118933]])
```