thegraph_data_access

March 14, 2021

1 TheGraph data access

courtesty of @markusbkoch submitted by @mzargham

```python
[1]: import pandas as pd
     import json
     import requests
     import matplotlib.pyplot as plt
     url = 'https://api.thegraph.com/subgraphs/name/balancer-labs/balancer'
     query = '''
     query {{
         pools(first: 1000, skip:{}) {{
             id
             liquidity
         }}
     }}'''
     n = 0
     pools = []
     while True:
         print(f'request {n+1}')
         r = requests.post(url, json = {'query':query.format(n*1000)})
         p = json.loads(r.content)['data']['pools']
         print(f'results {len(p)}')
         pools.extend(p)
         print(f'total {len(pools)}')
         n += 1
         if len(p) < 1000:
             break
     subgraph_tvl = pd.DataFrame(pools)
```

```
request 1
results 1000
total 1000
request 2
results 1000
total 2000
request 3
results 954
```

```
total 2954
```

[2]: `subgraph_tvl`

[2]:
```
                                              id  \
0       0x003a70265a3662342010823bea15dc84c6f7ed54
1       0x004e74ff81239c8f2ec0e2815defb970f3754d86
2       0x0077732357ac0f29e26ea629b79ab3b266ddb796
3       0x0092b2d25d76d84d27b999fe93d5e1c70511cd2b
4       0x0099447ef539718bba3c4d4d4b4491d307eedc53
...                                            ...
2949    0xffe8c31fb0ab62c99fc6e8c724d0f1949dbaa44f
2950    0xfff293e1f6c174867f23351c1510833c8087fecb
2951    0xfff29c8bce4fbe8702e9fa16e0e6c551f364f420
2952    0xfff2a5f81d14729408201341df42af29f3b30458
2953    0xfff82910d352abe04d00d542f0ded0bfc8516f78


                       liquidity
0        1439162.2561307821099600776800548975
1          680.92848691143123644704148766372
2        0.86531404204648888144268185911831250
3          8.41728837682050716701128507145078
4                                           0
...                                        ...
2949      2331.2040118092227555466454707190590
2950                                         0
2951                                         0
2952      3941064.328046486158164903983221526
2953                                         0

[2954 rows x 2 columns]
```

[3]: `subgraph_tvl.liquidity= subgraph_tvl.liquidity.apply(float)`

[4]: `subgraph_tvl.sort_values('liquidity', inplace=True)`

[5]: `subgraph_tvl.liquidity`

[5]:
```
1476    0.000000e+00
1717    0.000000e+00
1715    0.000000e+00
1714    0.000000e+00
1712    0.000000e+00
            ...
1071    6.912330e+07
1613    8.530380e+07
1051    2.238022e+08
```

```
362     3.400291e+08
2279    3.486736e+08
Name: liquidity, Length: 2954, dtype: float64
```

[6]: ```python
plt_df=subgraph_tvl[subgraph_tvl.liquidity>1].copy().sort_values('liquidity',
 ↪ascending=False)
```

[7]: ```python
subgraph_tvl.describe()
```

[7]:
```
          liquidity
count  2.954000e+03
mean   6.784773e+05
std    1.034519e+07
min    0.000000e+00
25%    0.000000e+00
50%    0.000000e+00
75%    6.201755e+02
max    3.486736e+08
```

[8]: ```python
plt_df.tail()
```

[8]:
|      | id                                         | liquidity |
|------|--------------------------------------------|-----------|
| 336  | 0x1d261ec7ab834fedb01602c5b7ffc6fc68362bbf | 1.577654  |
| 980  | 0x53f160490d7e48ba2c31be4790f3d87a2f4dc662 | 1.371422  |
| 1838 | 0x9e4a4b53e19410ae519be74f92659e5b0ef9489b | 1.330313  |
| 2342 | 0xcb8ec8236aff8e112517f4e9a9ffb413a237e6b7 | 1.153105  |
| 1288 | 0x6d42692518c8b09c883e7c1e69c97518107f2185 | 1.030083  |

[9]: ```python
plt_df.plot(x='id', y='liquidity', logy=True)
plt.xticks(rotation=90)
```
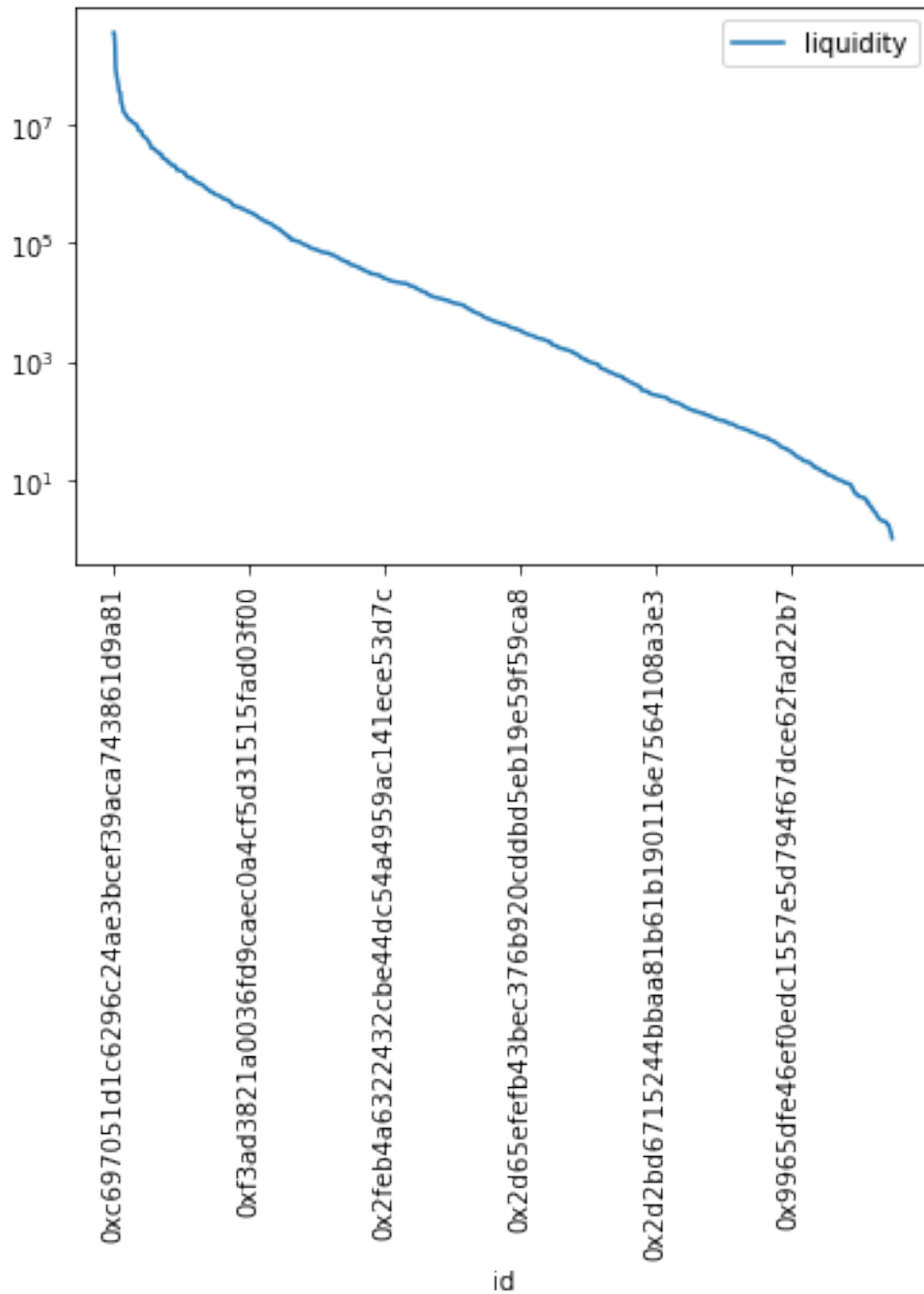
[9]: ```
(array([-200.,   0., 200., 400., 600., 800., 1000., 1200., 1400.]),
 [Text(-200.0, 0, '0xa8e4faa056c7f1c41f5838a896daa091dea93901'),
  Text(0.0, 0, '0xc697051d1c6296c24ae3bcef39aca743861d9a81'),
  Text(200.0, 0, '0xf3ad3821a0036fd9caec0a4cf5d31515fad03f00'),
  Text(400.0, 0, '0x2feb4a6322432cbe44dc54a4959ac141ece53d7c'),
  Text(600.0, 0, '0x2d65efefb43bec376b920cddbd5eb19e59f59ca8'),
  Text(800.0, 0, '0x2d2bd6715244bbaa81b61b190116e7564108a3e3'),
  Text(1000.0, 0, '0x9965dfe46ef0edc1557e5d794f67dce62fad22b7'),
  Text(1200.0, 0, ''),
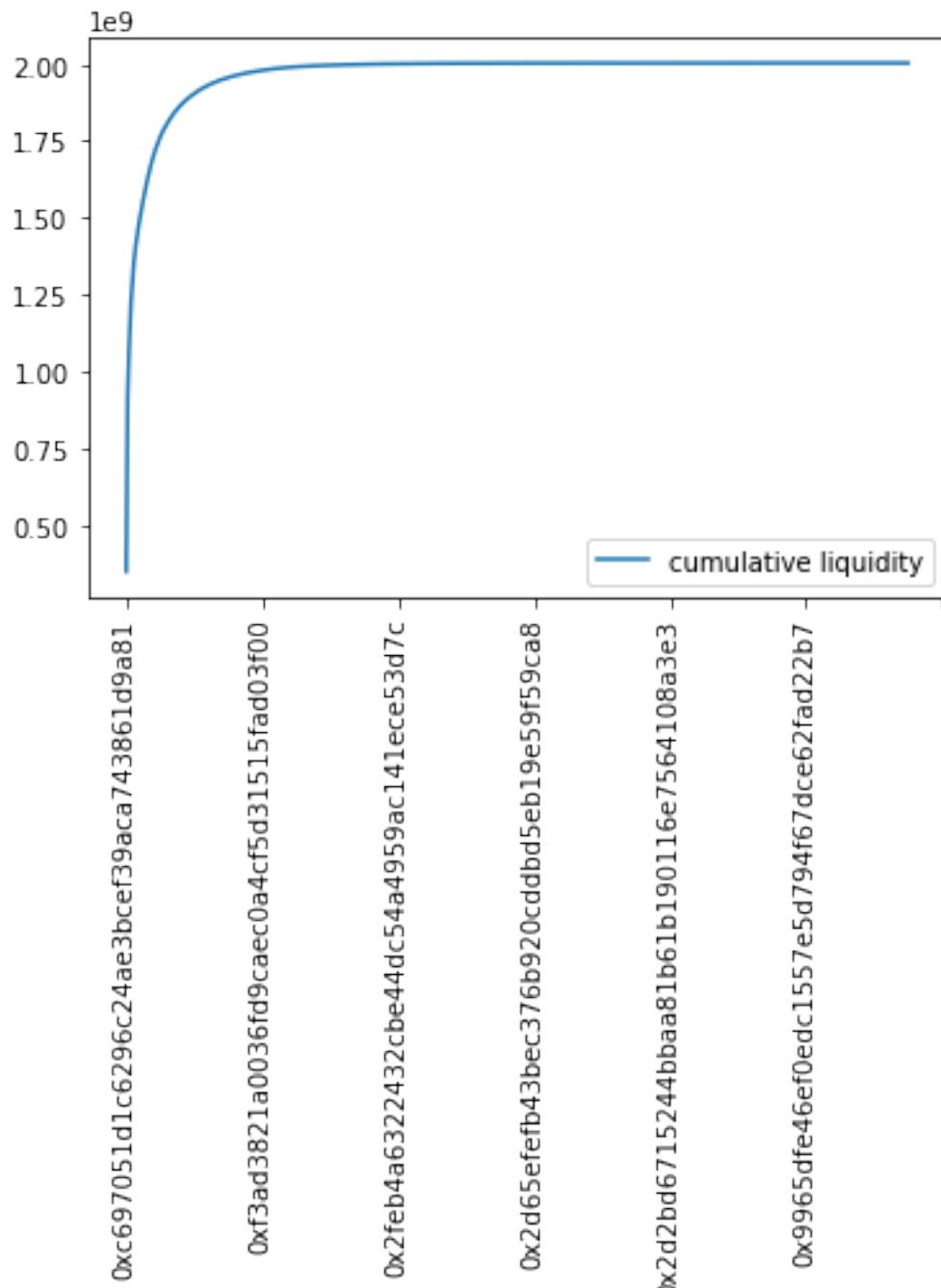 Text(1400.0, 0, '')])
```

```
[10]: plt_df['cumulative liquidity'] = plt_df.liquidity.cumsum()
```

```
[11]: plt_df.plot(x='id', y='cumulative liquidity', logy=False)
      plt.xticks(rotation=90)
```

```
[11]: (array([-200.,    0.,  200.,  400.,  600.,  800., 1000., 1200., 1400.]),
       [Text(-200.0, 0, '0xa8e4faa056c7f1c41f5838a896daa091dea93901'),
        Text(0.0, 0, '0xc697051d1c6296c24ae3bcef39aca743861d9a81'),
        Text(200.0, 0, '0xf3ad3821a0036fd9caec0a4cf5d31515fad03f00'),
        Text(400.0, 0, '0x2feb4a6322432cbe44dc54a4959ac141ece53d7c'),
        Text(600.0, 0, '0x2d65efefb43bec376b920cddbd5eb19e59f59ca8'),
        Text(800.0, 0, '0x2d2bd6715244bbaa81b61b190116e7564108a3e3'),
        Text(1000.0, 0, '0x9965dfe46ef0edc1557e5d794f67dce62fad22b7'),
        Text(1200.0, 0, ''),
        Text(1400.0, 0, '')])
```

The x-axis labels, from left to right:

0xc697051d1c6296c24ae3bcef39aca743861d9a81

0xf3ad3821a0036fd9caec0a4cf5d31515fad03f00

0x2feb4a6322432cbe44dc54a4959ac141ece53d7c

0x2d65efefb43bec376b920cddbd5eb19e59f59ca8

0x2d2bd6715244bbaa81b61b190116e7564108a3e3

0x9965dfe46ef0edc1557e5d794f67dce62fad22b7

id

[ ]:

[ ]: