K10plus

May 3, 2018

0.1 Quick comparison of K10plus, finc and ai

Let's quickly compare ISSN lists between K10plus, finc and ai.

- https://verbundwiki.gbv.de/display/VZG/K10plus-Zentral

  The K10Matches task uses the CSV linked on the above site:

- https://verbundwiki.gbv.de/download/attachments/23920642/Zeitschriften_issn.csv?version=1&modifi

  Required input: The output of K10Matches task.

```
In [1]: from __future__ import division
        from siskin.workflows.adhoc import K10Matches
        import pandas as pd

        import matplotlib
        matplotlib.use('agg') # Only necessary, if other backends are registered, e.g. itermpl

        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
In [2]: finc_solr_url = "xxx"
        ai_solr_url = "xxx"
```

```
In [3]: task = K10Matches(finc=finc_solr_url, ai=ai_solr_url)
```

```
In [4]: if not task.complete():
            raise RuntimeError("Run K10Matches task first, via luigi.")
```

```
In [5]: df = pd.read_csv(task.output().path, sep="\t", header=None, names=["issn", "k10", "ai"
```

```
In [6]: df.head()
```

```
Out[6]:        issn     k10      ai  finc
        0  19326203  373876  385645   682
        1  00293970  338744   35556     5
        2  03029743  248162  401444    53
        3  00219258  243186  104432     2
        4  10959203  239131  441892    19
```

```
In [7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 334515 entries, 0 to 334514
Data columns (total 4 columns):
issn    334515 non-null object
k10     334515 non-null int64
ai      334515 non-null int64
finc    334515 non-null int64
dtypes: int64(3), object(1)
memory usage: 10.2+ MB
```

There are about 334515 ISSN in the CSV file provided by K10. Most entries look like an ISSN.

```
In [8]: len(df[df.issn.str.match("^[\d]{7,7}[xX\d]$")])
```

```
Out[8]: 334305
```

```
In [9]: df.k10.describe()
```

```
Out[9]: count    334515.000000
        mean        352.033568
        std        2698.749649
        min           1.000000
        25%           1.000000
        50%           1.000000
        75%           6.000000
        max      373876.000000
        Name: k10, dtype: float64
```

```
In [10]: df.ai.describe()
```

```
Out[10]: count    334515.000000
         mean        422.225087
         std        4033.179211
         min           0.000000
         25%           0.000000
         50%           0.000000
         75%           3.000000
         max      486101.000000
         Name: ai, dtype: float64
```

On average a K10 ISSN has 352 entries associated with each ISSN, AI 422. Both distributions are right-skewed, as the median is much smaller then the mean.

```
In [11]: df[df.ai == df.ai.max()]
```

```
Out[11]:            issn  k10      ai  finc
         93179  09317597    4  486101     1
```

```
In [12]: df[df.k10 == df.k10.max()]
```

```
Out[12]:        issn     k10      ai  finc
         0  19326203  373876  385645   682
```

Better coverage on an ISSN in K10plus than in ai in about 73% of the ISSN.

```
In [13]: df[df.k10 > df.ai].shape
```

```
Out[13]: (244604, 4)
```

```
In [14]: len(df[df.k10 > df.ai]) / len(df) * 100
```

```
Out[14]: 73.12198257178302
```

How many ISSN are completely missing in ai?

```
In [15]: len(df[df.ai == 0]) / len(df) * 100
```

```
Out[15]: 63.18909465943231
```

63% of the ISSN in the CSV file are completely absent in AI.

0.1.1 Random ISSN in K10 but not in AI

Use a random result an google it.

One reason is probably the absence of PubMed in AI.

```
In [16]: randrow = df[df.ai == 0].sample(n=1)
         issn = randrow.issn.values[0]
         issn = "%s-%s" % (issn[:4], issn[4:])
```

```
In [17]: issn
```

```
Out[17]: '0303-6960'
```

```
In [18]: !googler --np -n 3 "$issn"
```

```
1 Indian Journal of Nematology - 0303-6960 - ABE-IPS
https://www.abe.pl/en/journal/0303-6960/
Indian Journal of Nematology - Technology, engineering, agriculture -
0303-6960.

2 Indian Journal of Nematology - Indian Journals
http://www.indianjournals.com/ijor.aspx?target=ijor:ijn&type=home
Publisher: The Nematological Society of India. Print ISSN: 0303-6960. Online
ISSN: 0974-4444. Number of issues per year: 2. Print frequency: Half-Yearly.
Month(s) of publication: June and December. Description: The Indian journal of
Nematology is published half - yearly by the Nematological society of India.
The journală...
```

```
3 ISSN: 0303-6960 - ISSN Database
http://www.issn.cc/0303-6960
About Indian journal of nematology. Abbreviation: Indian J. Nematol. ISSN:
0303-6960 (Print) 0303-6960 (Linking); Publisher: New Delhi : Nematological
Society of India; Language: Englishă...
```

Are there journals, where we have more entries in finc than K10plus? It seems, in about 5% of the cases.

```
In [19]: len(df[df.finc > df.k10]) / len(df) * 100
```

```
Out[19]: 5.229959792535461
```

Where does finc and ai combined has more coverage?

```
In [20]: len(df[(df.finc > df.k10) | (df.ai > df.k10)]) / len(df) * 100
```

```
Out[20]: 23.69818991674514
```

In about 23% of the cases.

When K10plus has better coverage, how much better is it? For example, if there are both entries in k10 and ai, what percentage do we have in ai, on average?

```
In [21]: better = df[(df.k10 > df.ai) & (df.ai > 0)] # Better k10
         (better.ai / better.k10).describe()
```

```
Out[21]: count    33227.000000
         mean         0.415194
         std          0.310612
         min          0.000007
         25%          0.125000
         50%          0.395051
         75%          0.666667
         max          0.999946
         dtype: float64
```

The ai contains on average 41% of records in k10, in the 33227 cases, where K10 has more and ai has some entries.

The other way around.

```
In [22]: better = df[(df.ai > df.k10) & (df.k10 > 0)] # Better ai
         (better.k10 / better.ai).describe()
```

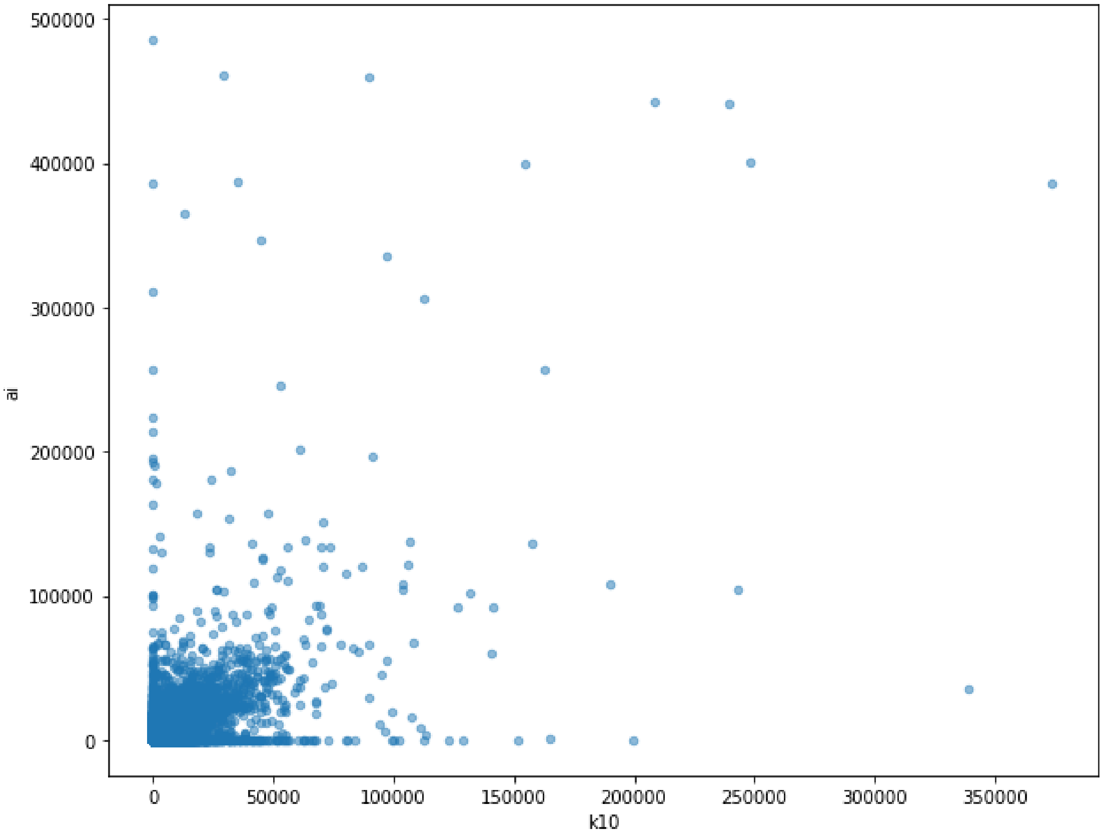
```
Out[22]: count    67772.000000
         mean         0.262434
         std          0.307783
         min          0.000003
         25%          0.007937
```

```
50%          0.071429
75%          0.500000
max          0.999901
dtype: float64
```

When ai contains more records than k10, then k10 contains around 26% of the records.

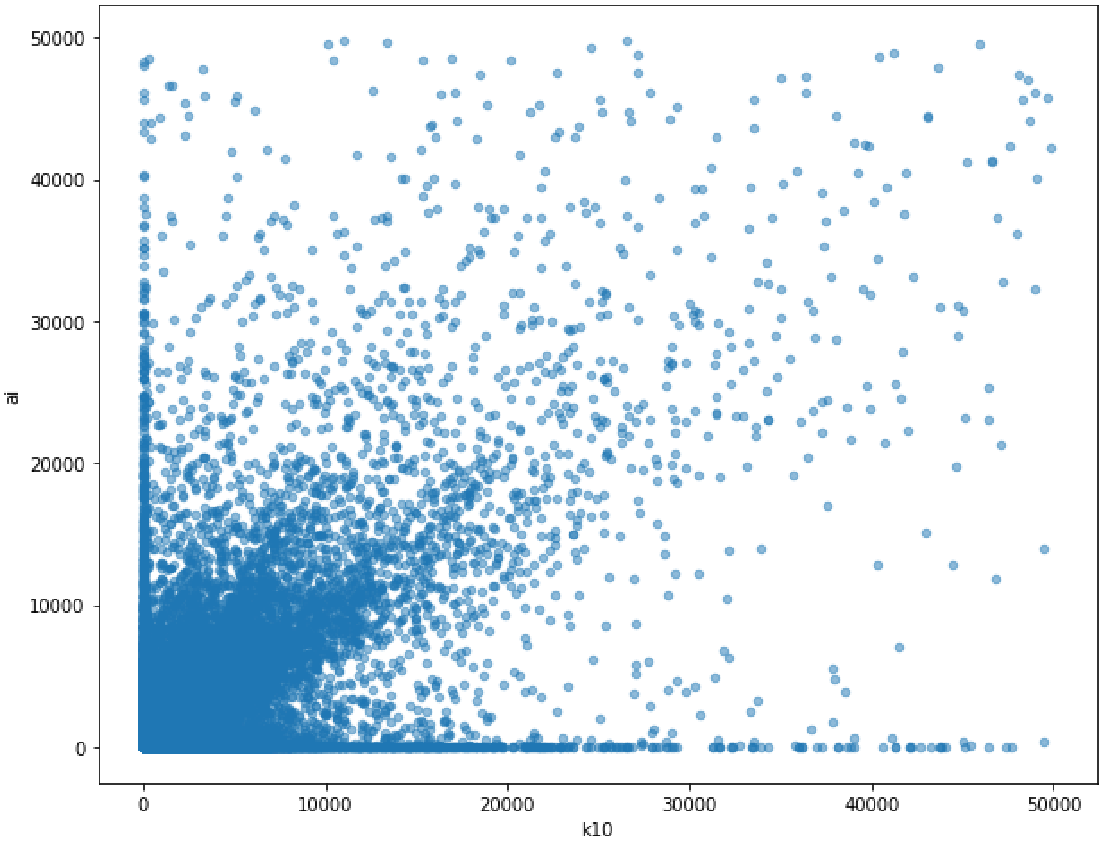
```
In [23]: df.plot(kind="scatter", x="k10", y="ai", alpha=0.5, figsize=(10, 8))
```

```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x10ee5d630>
```



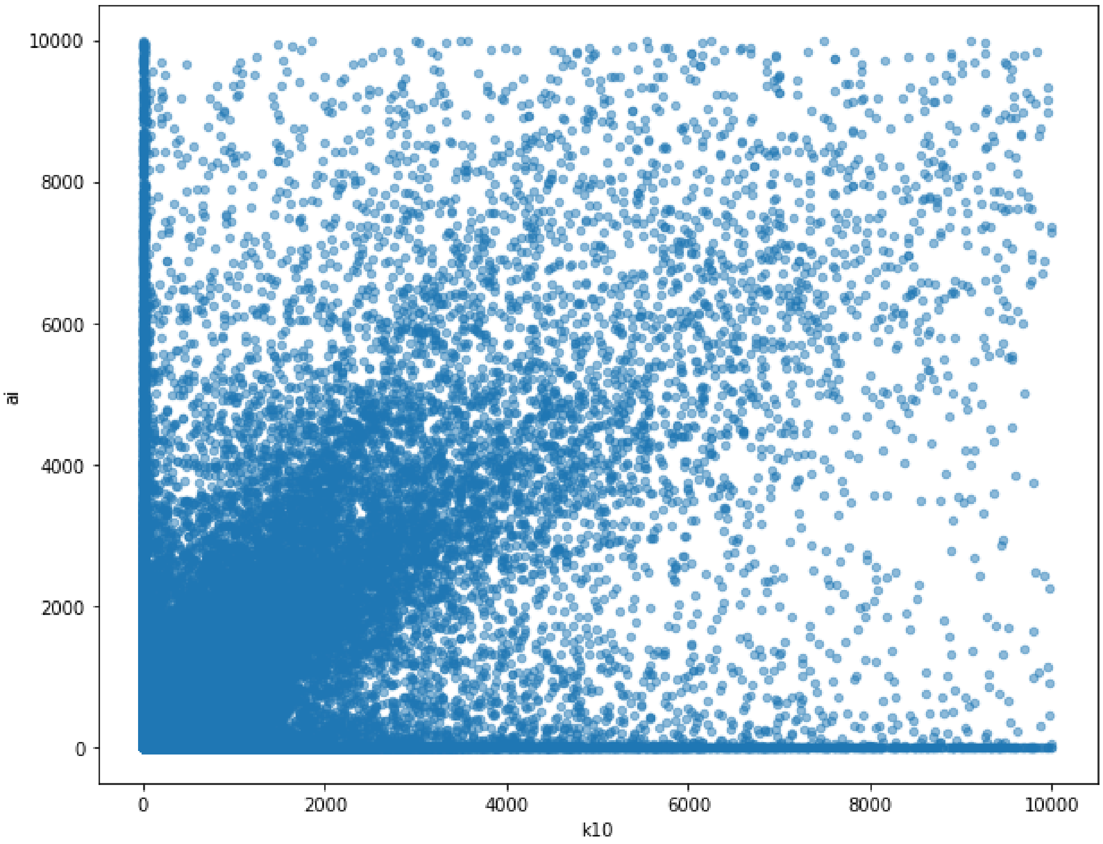
```
In [24]: df[(df.k10 < 50000) & (df.ai < 50000)].plot(kind="scatter", x="k10", y="ai", alpha=0.5
```

```
Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x113b75278>
```



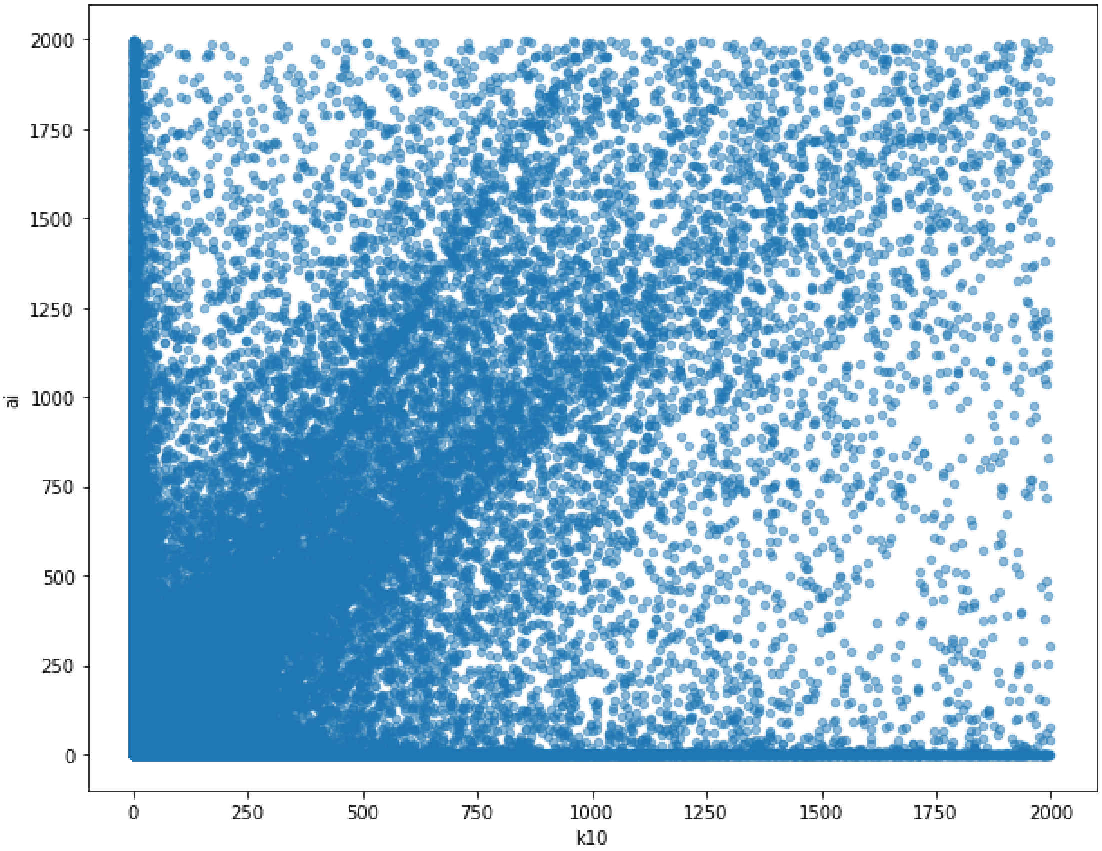
```
In [25]: df[(df.k10 < 10000) & (df.ai < 10000)].plot(kind="scatter", x="k10", y="ai", alpha=0.5

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x113e87048>
```



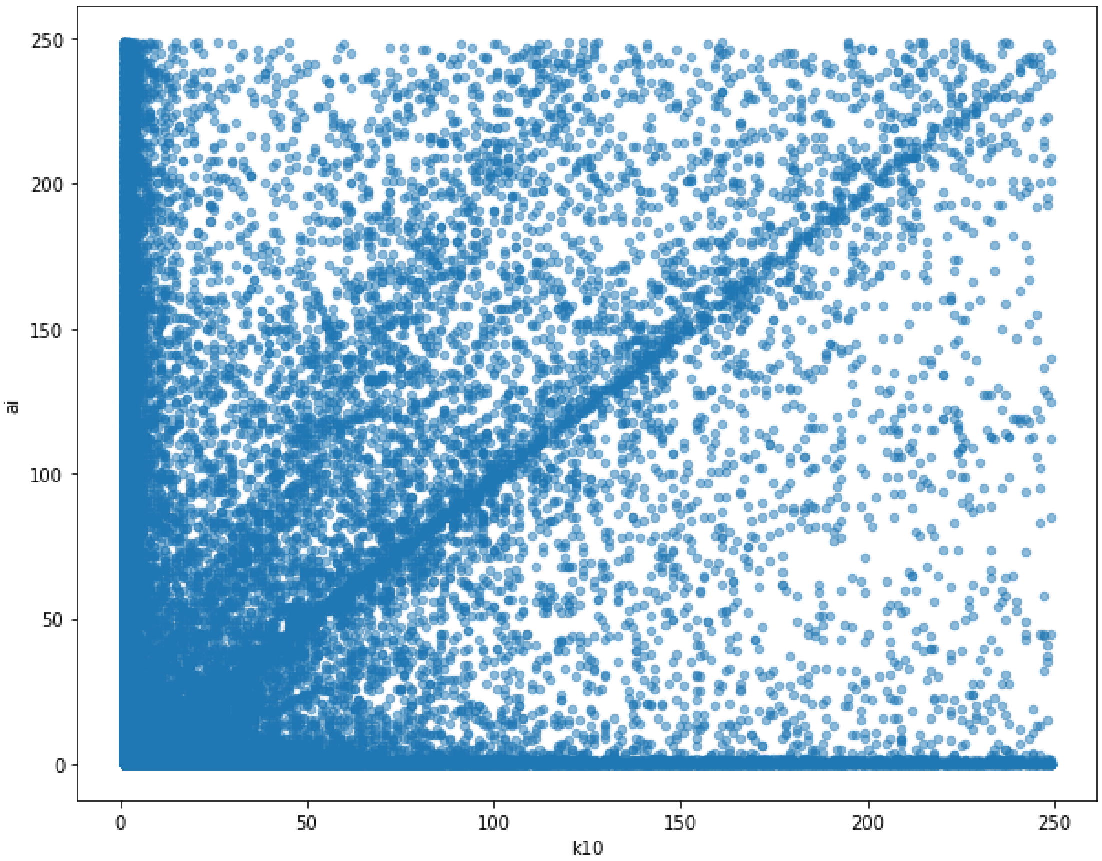
```
In [26]: df[(df.k10 < 2000) & (df.ai < 2000)].plot(kind="scatter", x="k10", y="ai", alpha=0.5,

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x113ec9128>
```



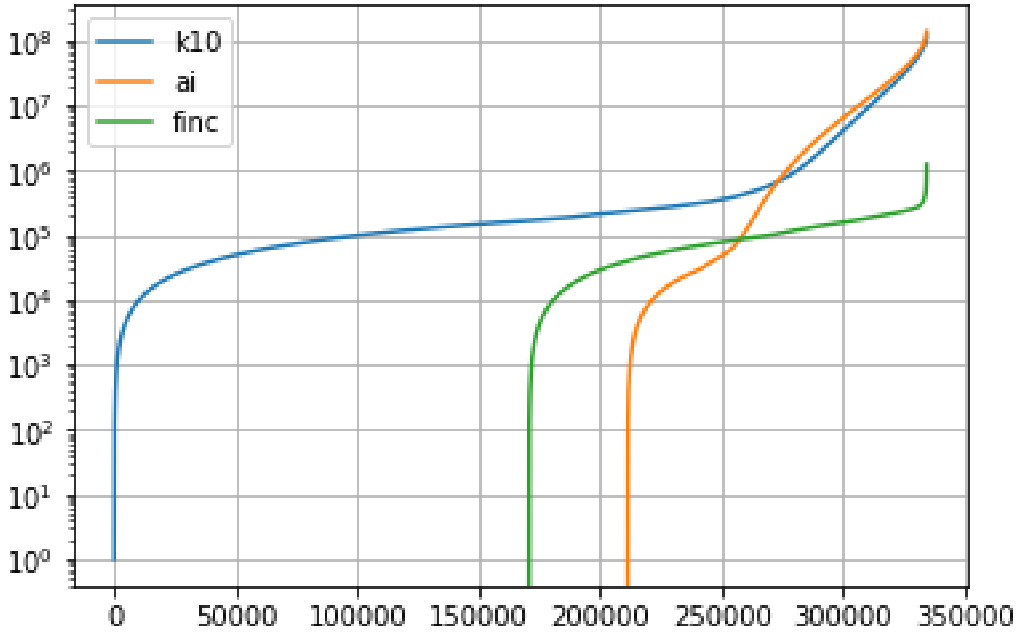
```
In [27]: df[(df.k10 < 250) & (df.ai < 250)].plot(kind="scatter", x="k10", y="ai", alpha=0.5, fi

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x114010f60>
```



```
In [28]: df.k10.sort_values().cumsum().reset_index(drop=True).plot(label="k10", logy=True)
         df.ai.sort_values().cumsum().reset_index(drop=True).plot(label="ai", logy=True)
         df.finc.sort_values().cumsum().reset_index(drop=True).plot(label="finc", logy=True)
         plt.grid(True)
         plt.legend()

Out[28]: <matplotlib.legend.Legend at 0x10edab978>
```


k10
ai
finc
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
0
50000
100000
150000
200000
250000
300000
350000