Overview

This document analyzes the DBLP (http://dblp.uni-trier.de/) computer science bibliography, distributed by AMiner (https://aminer.org/citation). I specifically analyze the DBLP-Citation-network V10 (https://static.aminer.org/lab-datasets/citation/dblp.v10.zip) dataset, containing 3,079,007 papers and 25,166,994 citations. Pre-processing of the data was performed in preprocessing.py (https://github.com/iwsmith/general_exam/blob/master/preprocessing.py.ipynb).

We are investigating the following research question:

> As noted in (4) search engines and recommender systems could be affecting what is being searched in the scholarly literature. Are readers, collectively, diverging in their reading interests or are they converging? What evidences supports these alternative interpretations? Using open, bibliographic data sets, investigate this question. The code and responses should be written in a Jupypter notebook with text, graphs and metrics interspersed. Be sure to justify your metrics and analyses and to include your interpretations of your data.

In [35]:
```
%matplotlib inline
import pandas as pd
import numpy as np
import seaborn as sns
figsize=(14,8)
```

In [48]:
```
df=pd.read_pickle("data.pickle")
```

In [3]:
```
df.head()
```

Out[3]:

	year	references	authors	title	out_cite	author_cnt	ti
id							
00127ee2-cb05-48ce-bc49-9de556b93346	2013	[51c7e02e-f5ed-431a-8cf5-f761f266d4be, 69b625b...	[Makoto Satoh, Ryo Muramatsu, Mizue Kayama, Ka...	Preliminary Design of a Network Protocol Learn...	2	8	3
001c58d3-26ad-46b3-ab3a-c1e557d16821	2011	[10482dd3-4642-4193-842f-85f3b70fcf65, 3133714...	[Gareth Beale, Graeme Earl]	A methodology for the physically accurate visu...	13	2	3
001c8744-73c4-4b04-9364-22d31a10dbf1	2009	[2d84c0f2-e656-4ce7-b018-90eda1c132fe, a083a1b...	[Altaf Hossain, Faisal Zaman, Mohammed Nasser,...	Comparison of GARCH, Neural Network and Suppor...	2	4	3
00338203-9eb3-40c5-9f31-cbac73a519ec	2011	[8c78e4b0-632b-4293-b491-85b1976675e6, 9cdc54f...	[Jea-Bum Park, Byungmok Kim, Jian Shen, Sun-Yo...	Development of Remote Monitoring and Control D...	2	5	3
0040b022-1472-4f70-a753-74832df65266	1998	None	[Giovanna Guerrini, Isabella Merlo]	Reasonig about Set-Oriented Methods in Object ...	0	2	3

```
In [50]: df.describe()
```

Out[50]:

	year	out_cite	author_cnt	title_len	flow	
count	3.079007e+06	3.079007e+06	3.079007e+06	3.079007e+06	2.725533e+06	1.9859
mean	2.007767e+03	8.173737e+00	3.077669e+00	7.316575e+01	3.669007e-07	1.2672
std	7.816538e+00	9.707329e+00	1.780460e+00	2.559721e+01	2.312681e-06	5.6168
min	1.936000e+03	0.000000e+00	0.000000e+00	4.000000e+00	0.000000e+00	1.0000
25%	2.004000e+03	1.000000e+00	2.000000e+00	5.500000e+01	0.000000e+00	2.0000
50%	2.010000e+03	6.000000e+00	3.000000e+00	7.100000e+01	5.301480e-08	4.0000
75%	2.013000e+03	1.200000e+01	4.000000e+00	8.800000e+01	2.095020e-07	1.0000
max	2.018000e+03	1.532000e+03	3.510000e+02	4.760000e+02	7.349310e-04	1.6229

The dataframe consists of 6 columns. Most are self explanitory, except for `flow`, which referes to the Eigenfactor score of a given paper. These scores were generated by running: `./infomap dblp.pajek output -t --inner-parallelization -i pjk`

Of note is that the collection spans from 1936 to 2018, with a maximum of 1,523 out citations ("Image analysis and computer vision : 1998"), and 16,229 in citations ("Distinctive Image Features from Scale-Invariant Keypoints").

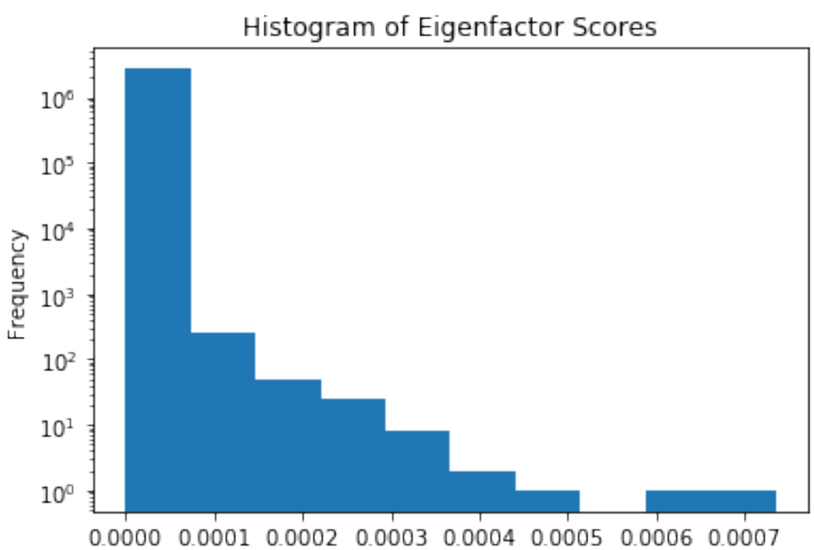
In [6]:
```
df.sort_values("flow", ascending=False)[["year","title","flow"]].head(
)
```

Out[6]:

	year	title	flow
id			
6a6b9aa6-683f-4c7c-b06e-9c3018d10fd3	1989	Genetic Algorithms in Search, Optimization and...	0.000735
b944f77f-113b-4a02-ae5e-d4a124b8fd5b	2004	Distinctive Image Features from Scale-Invarian...	0.000599
c1b6b493-01ef-420f-be44-7bacfe34e846	2011	LIBSVM: A library for support vector machines	0.000474
a662a4e7-415e-417e-8a8f-fe085d7e487f	1974	The Design and Analysis of Computer Algorithms	0.000416
d3e00e7e-1c64-4d7a-b2b2-1ad98ba4c706	1988	Probabilistic Reasoning in Intelligent Systems...	0.000372

In [57]:
```
df.flow.plot(kind="hist", logy=True, title="Histogram of Eigenfactor S
cores")
```

Out[57]: `<matplotlib.axes._subplots.AxesSubplot at 0x262c5f710>`



Looking at the top papers by flow we find reasonable titles. Notably these all seem to be real papers, and not mislabled journals or other erros. The same holds true for in-citations.

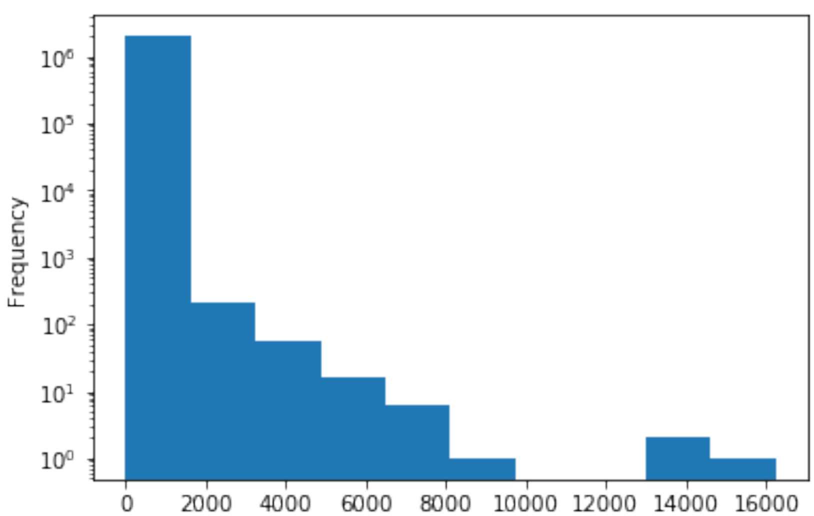
In [7]:
```
df.sort_values("in_cite", ascending=False)[["year","title","in_cite"]]
.head()
```

Out[7]:

	year	title	in_cite
id			
b944f77f-113b-4a02-ae5e-d4a124b8fd5b	2004	Distinctive Image Features from Scale-Invarian...	16229.0
c1b6b493-01ef-420f-be44-7bacfe34e846	2011	LIBSVM: A library for support vector machines	13475.0
6a6b9aa6-683f-4c7c-b06e-9c3018d10fd3	1989	Genetic Algorithms in Search, Optimization and...	13267.0
dd83785a-dd19-41e3-9b25-ebabbd48d336	2005	Histograms of oriented gradients for human det...	8477.0
f6bd8b64-684d-429a-aab5-8ff3a2c23cd6	2001	Random Forests	7968.0

In [12]:
```
df.in_cite.plot(kind="hist", logy=True, title="Histogram of in-citatio
ns")
```

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x1deb5e550>



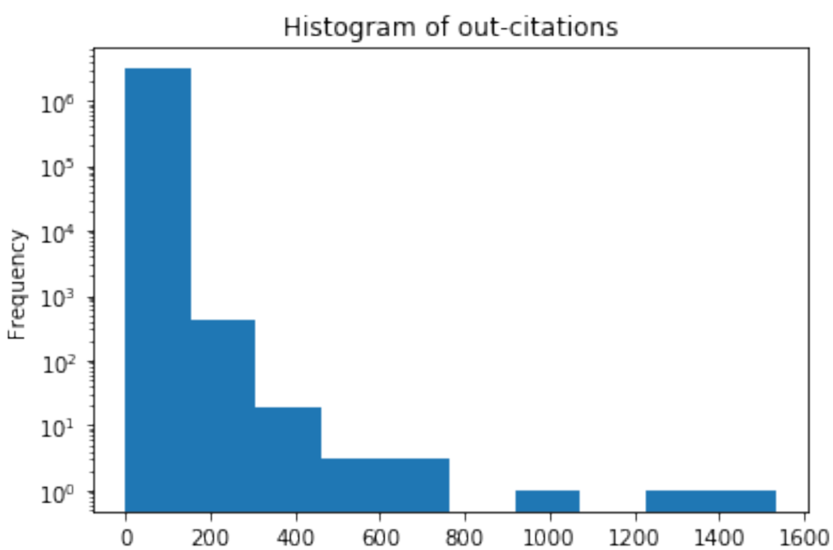
In [8]:
```
df.sort_values("out_cite", ascending=False)[["year","title","out_cite"
]].head()
```

Out[8]:

	year	title	out_cite
id			
da170252-5470-4ed6-947f-04003976f579	1999	Image analysis and computer vision : 1998	1532
ac2acfac-4597-4ebd-b1c5-08c888d73271	1997	Image analysis and computer vision: 1996	1362
d3be0271-593b-44e9-a2c4-cacb26dc1833	2000	Image analysis and computer vision: 1999	1001
c6090c87-4730-4ae6-86cd-7663adb23b2b	2002	Comprehensive frequency-dependent substrate no...	759
d0c11ccc-e743-4a45-bbbc-ba5030b78e33	1991	Image analysis and computer vision: 1990	734

In [58]:
```
df.out_cite.plot(kind="hist", logy=True, title="Histogram of out-citat
ions")
```

Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x3270f09b0>



Here we find an error, `Image analysis and computer vision: 1998` is likely the title of a journal, not an individual paper. This sort of error is common in bibliometric datasets. A classic example is the Google Scholar profile for the well known et al (https://scholar.google.nl/citations?user=qGuYgMsAAAAJ&hl=en), who has nearly 115k citations this year alone.

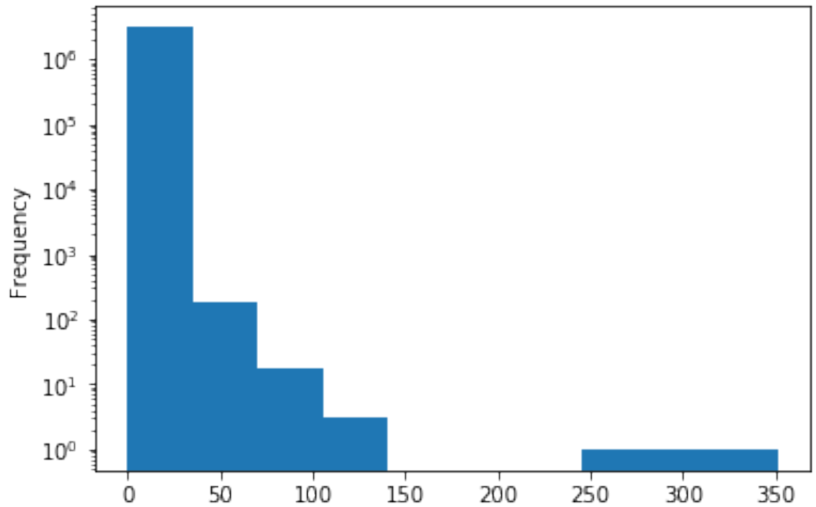
In [15]:
```
df.sort_values("author_cnt", ascending=False)[["year","title","author_
cnt"]].head()
```

Out[15]:

	year	title	author_cnt
id			
5de90b6c-92cb-40f5-898a-5eecd10c3d14	2017	Construction and Analysis of Weighted Brain Ne...	351
23d6baa9-65fb-4cd1-a470-63b4227f2955	2015	The IceProd framework : Distributed data proce...	286
2c12b8a7-4258-4513-b40a-475427d492c5	2014	A promoter-level mammalian expression atlas	261
66fe9d26-426c-4c98-ab7e-fe71dd4d48ed	2006	Length Sensing and Control in the Virgo Gravit...	119
106b4db9-2ae6-4744-9533-f7809345f1ca	2002	An Overview of the BlueGene/L Supercomputer	115

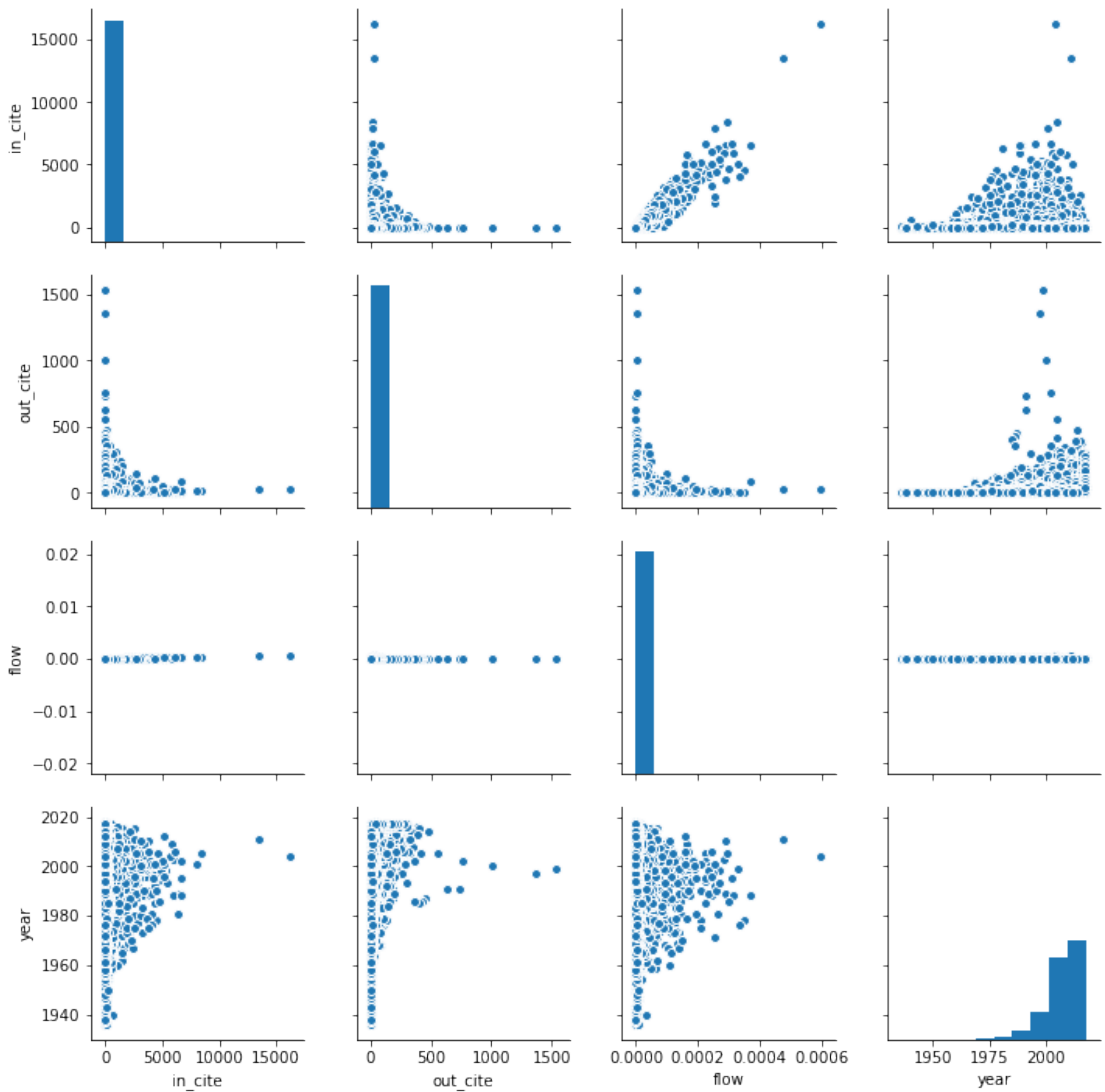
In [16]:
```
df.author_cnt.plot(kind="hist", logy=True, title="Histogram of author counts")
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x23a388f28>



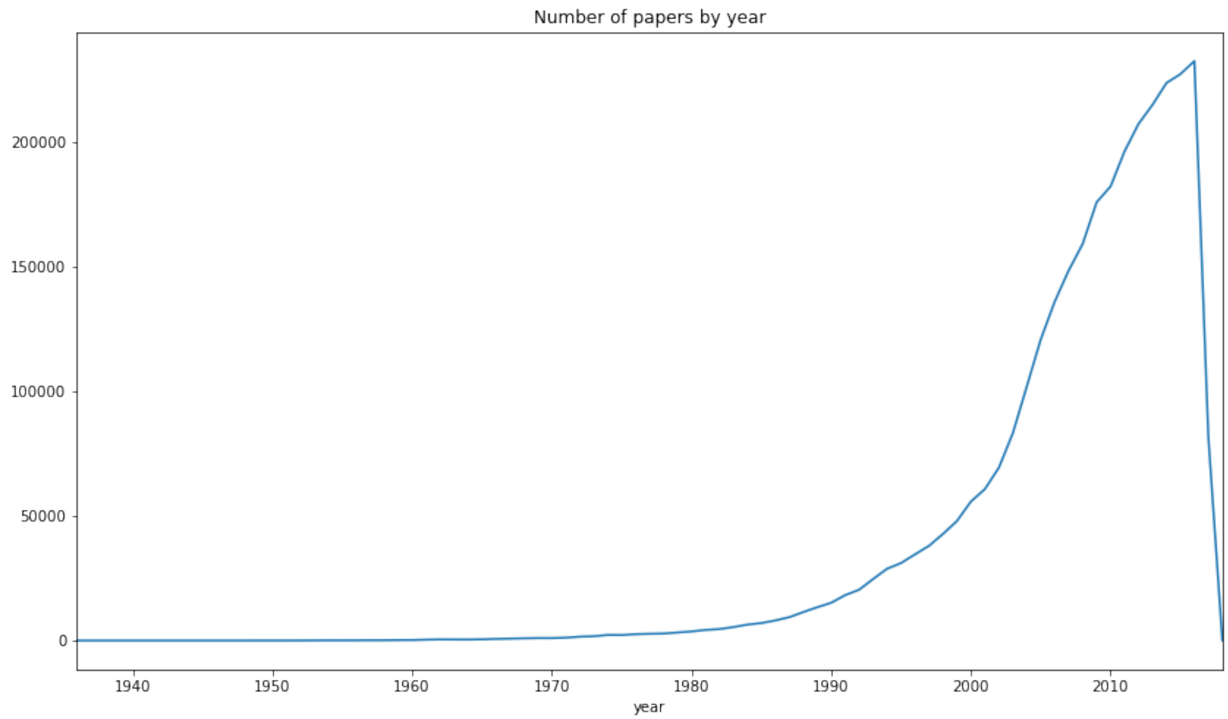
In [61]:
```
sns.pairplot(df.dropna(), vars=["in_cite","out_cite","flow", "year"])
```

Out[61]: <seaborn.axisgrid.PairGrid at 0x220a4f550>


in_cite
out_cite
flow
year
16000
10000
5000
0
1500
1000
500
0
0.02
0.01
0.00
−0.01
−0.02
2020
2000
1980
1960
1940
0
5000
10000
15000
in_cite
0
500
1000
1500
out_cite
0.0000
0.0002
0.0004
0.0006
flow
1950
1975
2000
year

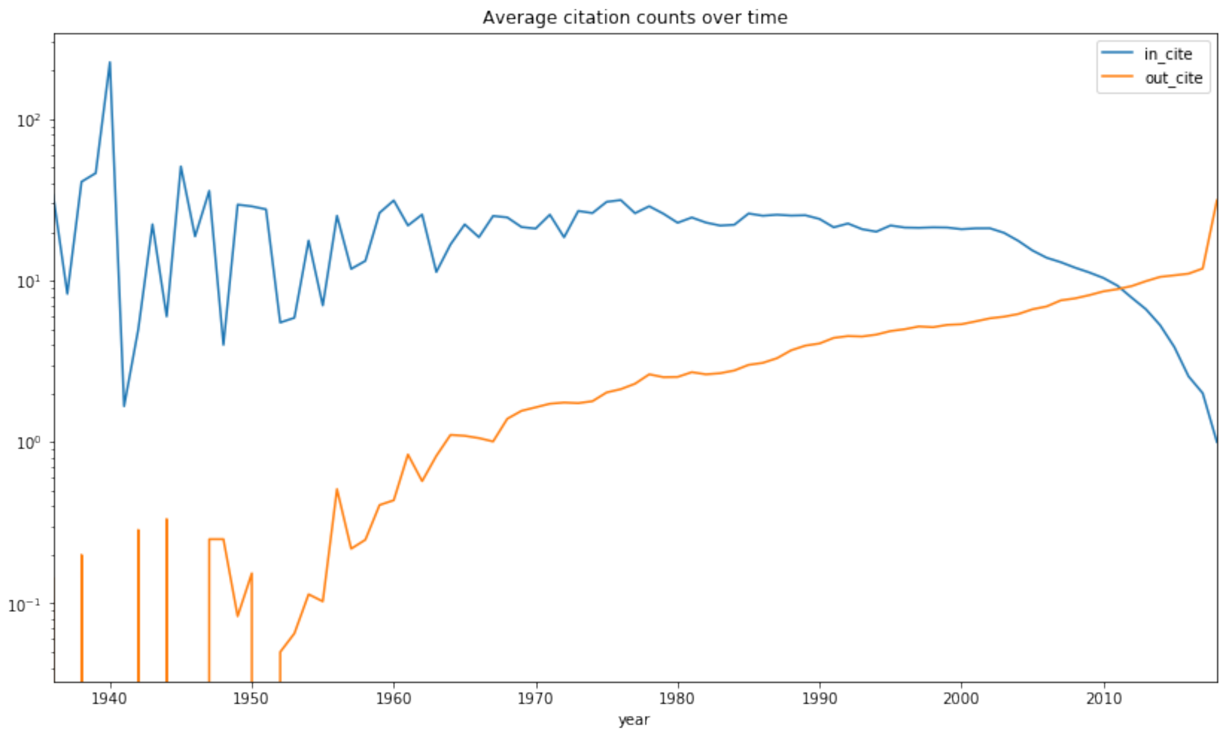
In [40]:
```
df.groupby("year").size().plot(title="Number of papers by year", figsi
ze=figsize)
```

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x1e601a908>



In [76]:
```
df[["in_cite", "out_cite", "year"]].groupby("year").mean().plot(logy=True, figsize=figsize, title="Average citation counts over time")
```

Out[76]: <matplotlib.axes._subplots.AxesSubplot at 0x265fcea90>

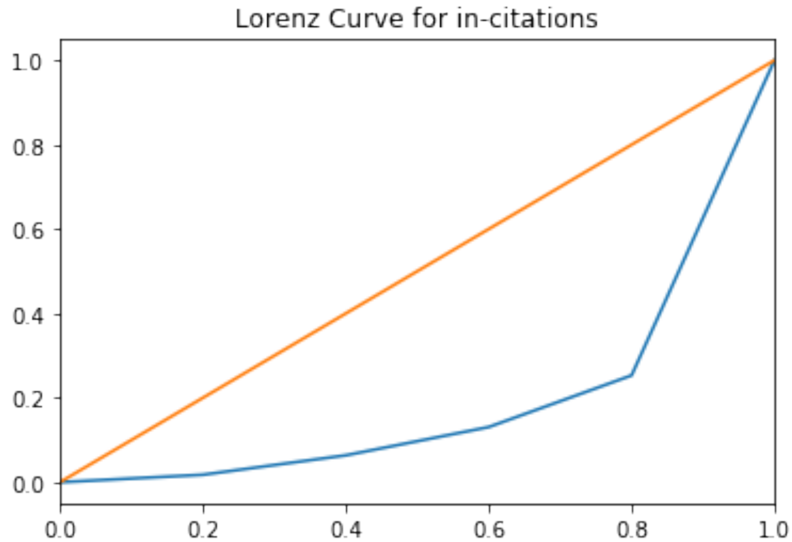


Since we are investigating questions around citation patterns we should understand what they look like in this dataset. First, the top plot shows a sharp increase in the number of papers after 2000. In the second plot we see a steady increase in the average number of out citations over time, while the average in bound citations remain steady until apporximately 2000, where they begin falling off. This could be due to papers not yet being fixed in the citation graph, though 10 years is a longer estimate than is normally provided, especially for copmuter science, which this collection covers.

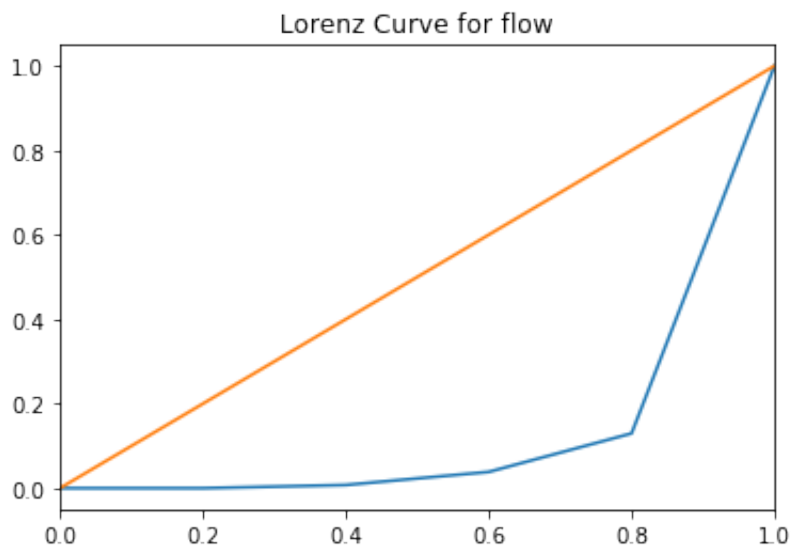
In [68]:
```
def lorenz_curve(ds, quantiles=[.2,.4,.6,.8,1]):
    breaks = ds.quantile(quantiles)
    t = ds.sum()
    cs = [ds[ds <= b].sum()/t for b in breaks]
    cs.insert(0, 0)
    quantiles.insert(0,0)
    return pd.Series(cs, index=quantiles)

def plot_lorenz(ls, tl=None):
    ax = ls.plot(title=tl)
    ax.plot([0,1],[0,1])
```

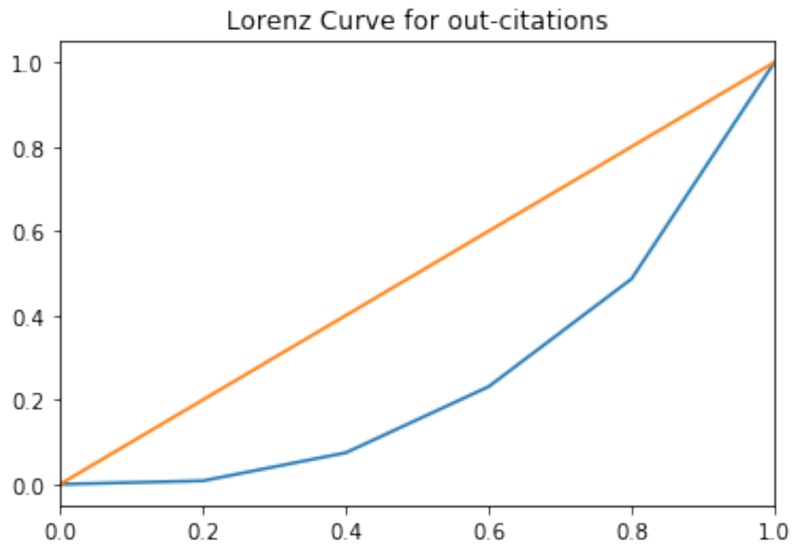
In [71]:
```
plot_lorenz(lorenz_curve(df.in_cite), "Lorenz Curve for in-citations")
```



In [72]:
```
plot_lorenz(lorenz_curve(df.flow), "Lorenz Curve for flow")
```



In [73]:
```
plot_lorenz(lorenz_curve(df.out_cite), "Lorenz Curve for out-citations
")
```



In the above plots we compare the Lorenz Curves for in-citations, Eigenfactor score (flow), and out-citations. A Lorenz Curve is a visualization of the distribution of some value, with the x-axis being the cumulative percent of the populace (papers in this case), and the y-axis the cumulative percent of the value being examined. The Yellow line in each represents perfect equality (e.g. each paper has the exact same number of citations). It is interesting to see that Eigenfactor scores have worse inequality than in-citations.

A common method for calculating inequality is via the Gini coefficient. We calculate the Gini coefficient for a non-decreasing $y_i <= y_{i+1}$ as:

$$G = \frac{1}{n}(n + 1 - 2\frac{\sum_{i=1}^{n}(n + 1 - i)y_i}{\sum_{i=1}^{n} y_i})$$

In [21]:
```
def gini(vs):
    if isinstance(vs, pd.Series):
        vs = vs.values
    vs = vs[~np.isnan(vs)]
    s = 0
    n = len(vs)
    vs.sort()
    svs = sum(vs)
    if svs == 0:
        return 0
    for i, y in enumerate(vs, start=1):
        s+=(n+1-i)*y
    return 1/n*(n+1-2*s/svs)
```

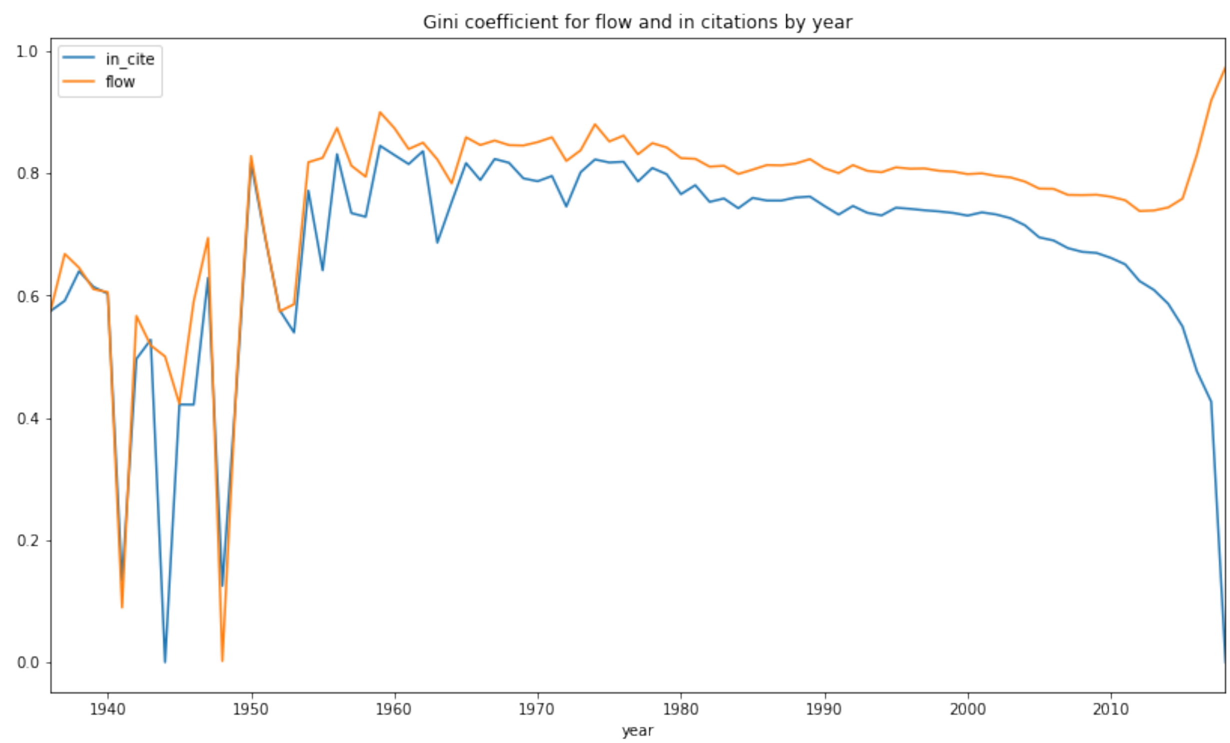
In [75]:
```
gini(df.in_cite)
```

Out[75]: 0.7162279553832788

Above we calculate the Gini coefficient over the entire dataset for in-citations. We find an answer of 71.6. If we were looking at global income inequality this would be higher than any country in the world. The World Bank estimates South Africa's coeficient at 63.4. For reference, the US has a coeficient of 41. This means the distribution of in-citations in the DBLP corpus is more unequal than the most unequal country.

In [45]:
```
gi = df[["year","in_cite"]].groupby("year").agg(gini)
gf = df[["year","flow"]].groupby("year").agg(gini)
gi.join(gf).plot(title="Gini coefficient for flow and in-citations by year", figsize=figsize)
```

Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x239986898>



Here we calcuate the Gini coefficient for in-citations and Eigenfactor score (flow) for each year. There appears to be a slight decrease from the 1970's on, with a drastic drop after 2000, corresponding to the drop in the average in-citations we observed earlier. There are a few possible interpretations here: 1) paper citations becoming more diverse or 2) the highly cited/classic papers haven't yet been identified. Given the corresponding drop in the average number of in-citations (1) seems to have more evidence supporting it.

In [49]:
```
df[["in_cite", "out_cite", "year", "author_cnt", "title_len"]].corr()
```

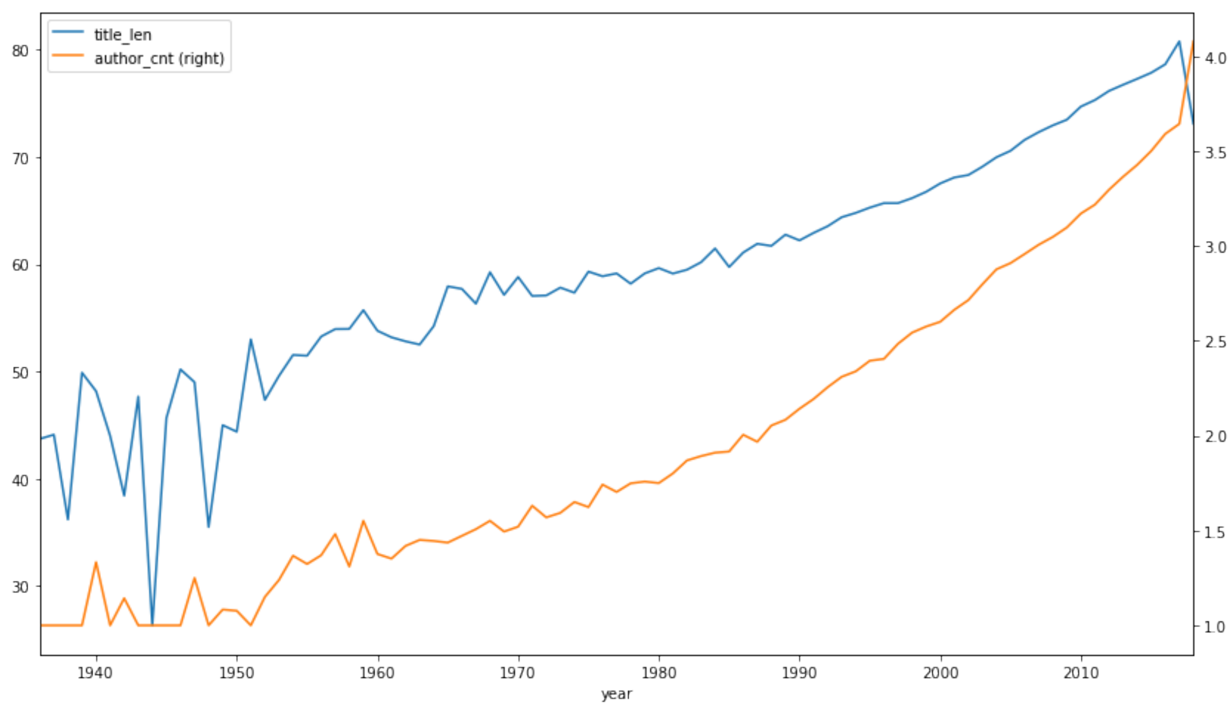
Out[49]:

	in_cite	out_cite	year	author_cnt	title_len
in_cite	1.000000	0.082356	-0.102966	-0.006078	-0.053284
out_cite	0.082356	1.000000	0.221947	0.069297	0.000437
year	-0.102966	0.221947	1.000000	0.230172	0.185475
author_cnt	-0.006078	0.069297	0.230172	1.000000	0.129574
title_len	-0.053284	0.000437	0.185475	0.129574	1.000000

If there is a change in citation counts over time we might find evidence of this in correlations. We find that in-citations are weakly negatively correlated with time, which makes sense: older papers have more time to gather citations. We also see out-citatoins positively correlated with time, a finding that matches earlier analysis.

In [55]:
```
df[["title_len", "author_cnt", "year"]].groupby("year").mean().plot(figsize=figsize, secondary_y=["author_cnt"])
```

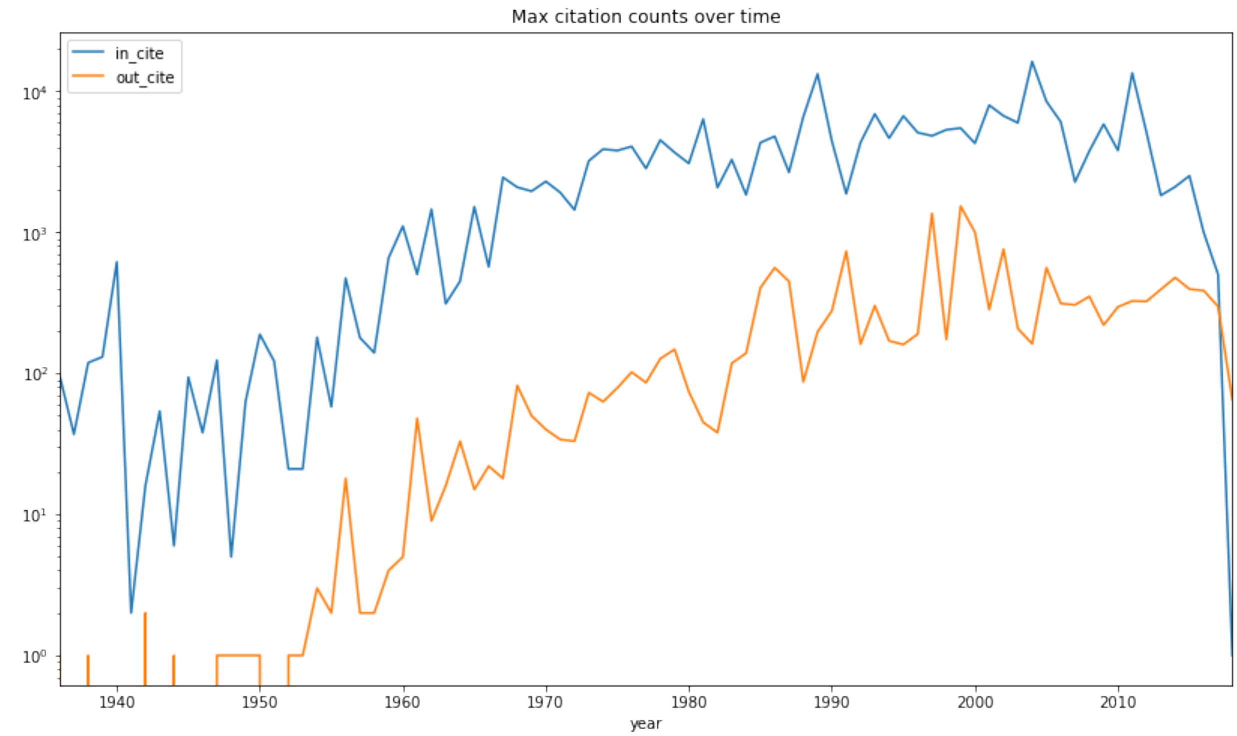
Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x1e54e0b70>



Finally, for a bit of fun I decided to see if the number of authors or title lengths are changing over time. It turns out both are increasing, nearly doubling over the lifetime of the entire dataset. I can understand the increase in authors, there are more scientists than there ever have been. I am less clear on why title lengths have gotten so much longer though, perhaps some kind of academic SEO in action?

In [78]:
```
df[["in_cite", "out_cite", "year"]].groupby("year").max().plot(logy=True, figsize=figsize, title="Max citation counts over time")
```

Out[78]: <matplotlib.axes._subplots.AxesSubplot at 0x3169cc710>



Conclusions

I analyzed the DBLP computer science bibliography looking for effects of recommenders in the citation patterns of the papers, specifically the Matthew effect (the rich get richer, while the poor get poorer). Using a Gini coefficient for each year I did not increasing inequality in in-citations over time. Actually there appears to be a small decrease over time. This could correspond to an increasing number of papers being written, which are diluting the effects of superstars. This would be the case if there were some sort of upper limit on in-citations. The above graph investigates this, showing that the maximum number of in-citations over time is increasing, though perhaps not as quickly as the number of additional papers being written.

Future investigations should try and derive a distance metric for papers. This would allow us to measure how diverse citations lists are for each year.