jupyter zbMATH Links API demo

This notebooks demonstrates some of the capabilites of the zbMATH links API.

In [1]:

```
import pandas as pd
pd.options.plotting.backend = "plotly"
```

Load statics for msc distribtion.

In [2]:

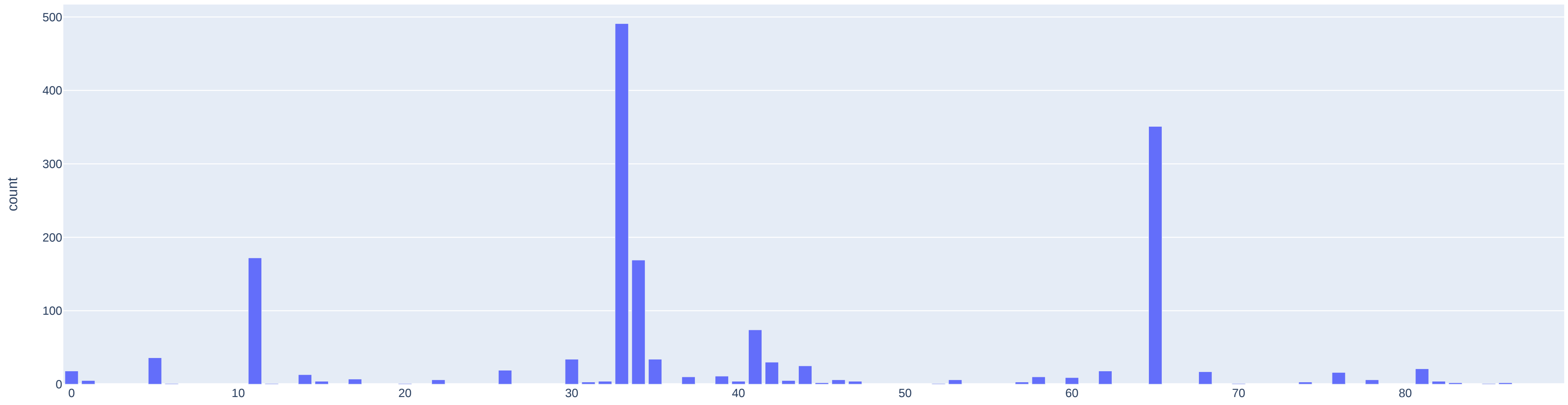
```
msc = pd.read_json('https://zblink.formulasearchengine.com/links_api/statistics/msc/')
msc = msc.rename(columns={msc.columns[0]:'msc',msc.columns[1]:'count'})
msc = msc.set_index('msc')
msc.head()
```

Out[2]:

	count
msc	
33	491
65	351
11	172
34	169
41	74

In [3]:

```
fig=msc.plot(y='count',kind='bar')
fig.show()
```



In the same way one can derive the year distribution

In [5]:

```
y = pd.read_json('https://zblink.formulasearchengine.com/links_api/statistics/years/')
y = y.rename(columns={y.columns[0]:'year',y.columns[1]:'count'})
y = y.set_index('year')
y
```

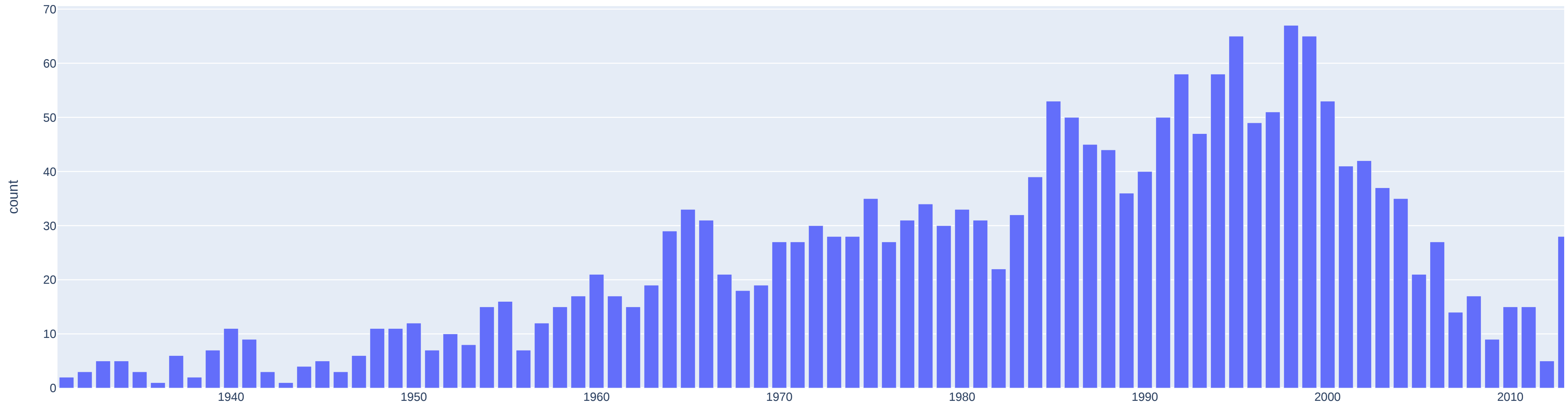
Out[5]:

	count
year	
1998	67
1999	65
1995	65
1992	58
1994	58
...	...
1938	2
1931	2
1943	1
1936	1
2017	1

87 rows × 1 columns

In [6]:

```
f=y.plot(kind='bar',y='count')
f.show()
```



Instead of the preprocessed statistics one can also access the uderling data in the scholix format. In the example below we filter for links that point to a MSC class 14.

In [51]:

```
import requests, json
alg_geo_json = requests.get('https://zblink.formulasearchengine.com/links_api/link/?msc%20classification%20code=14')
alg_geo_data = json.loads(alg_geo_json.text)
df = pd.json_normalize(alg_geo_data)
df
```

Out[51]:

	LinkPublicationDate	RelationshipType	Source.Identifier.ID	Source.Identifier.IDScheme	Source.Identifier.IDURL	Source.Type	Source.Title	Source.Publisher.Name	Source.Publisher.Identifier.ID	Source.Publisher.Identifier.IDScheme	...	Target.Type.Name	Target.Typ

	LinkPublicationDate	RelationshipType	Source.Identifier.ID	Source.Identifier.IDScheme	Source.Identifier.IDURL	Source.Type	Source.Title	Source.Publisher.Name	Source.Publisher.Identifier.ID	Source.Publisher.Identifier.IDScheme	...	Target.Type.Name	Target.Typ
0	1900-01-01T00:00:00	none	21.7#ii.info	DLMF scheme	https://dlmf.nist.gov/21.7#ii.info	{'Name': 'DLMF bibliographic entry'}	§21.7(ii) Fay's Trisecant Identity ‣ §21.7 Rie...	DLMF	DLMF	name of partner	...	book	14K25 14 14-02 33E 58J
1	1900-01-01T00:00:00	none	32.2#iv.p1	DLMF scheme	https://dlmf.nist.gov/32.2#iv.p1	{'Name': 'DLMF bibliographic entry'}	§32.2(iv) Elliptic Form ‣ §32.2 Differential E...	DLMF	DLMF	name of partner	...	book_article	14H52 34M 14N
2	1900-01-01T00:00:00	none	32.7#viii.p1	DLMF scheme	https://dlmf.nist.gov/32.7#viii.p1	{'Name': 'DLMF bibliographic entry'}	§32.7(viii) Affine Weyl Groups ‣ §32.7 Bäcklun...	DLMF	DLMF	name of partner	...	serial_article	37
3	1900-01-01T00:00:00	none	about/bio/AIBobenko#p2	DLMF scheme	https://dlmf.nist.gov/about/bio/AIBobenko#p2	{'Name': 'DLMF bibliographic entry'}	Profile Alexander I. Bobenko ‣ About the Project	DLMF	DLMF	name of partner	...	book	35-02 35C 37J35 37K 14H
4	1900-01-01T00:00:00	none	20.7#v.info	DLMF scheme	https://dlmf.nist.gov/20.7#v.info	{'Name': 'DLMF bibliographic entry'}	§20.7(v) Watson's Identities ‣ §20.7 Identitie...	DLMF	DLMF	name of partner	...	book	11-02 11C 33E05 1 11R
...	...	...	...	...	...	...	...	...	...	...	...	...	
111	1900-01-01T00:00:00	none	21.6#ii.info	DLMF scheme	https://dlmf.nist.gov/21.6#ii.info	{'Name': 'DLMF bibliographic entry'}	§21.6(ii) Addition Formulas ‣ §21.6 Products ‣...	DLMF	DLMF	name of partner	...	serial_article	58J99 35C 35B99 14H 53C
112	1900-01-01T00:00:00	none	21.9#info	DLMF scheme	https://dlmf.nist.gov/21.9#info	{'Name': 'DLMF bibliographic entry'}	§21.9 Integrable Equations ‣ Applications ‣ Ch...	DLMF	DLMF	name of partner	...	serial_article	58J99 35C 35B99 14H 53C
113	1900-01-01T00:00:00	none	20.12#i.p1	DLMF scheme	https://dlmf.nist.gov/20.12#i.p1	{'Name': 'DLMF bibliographic entry'}	§20.12(i) Number Theory ‣ §20.12 Mathematical ...	DLMF	DLMF	name of partner	...	book	11G05 14 1
114	1900-01-01T00:00:00	none	21.7#i.p4	DLMF scheme	https://dlmf.nist.gov/21.7#i.p4	{'Name': 'DLMF bibliographic entry'}	§21.7(i) Connection of Riemann Theta Functions...	DLMF	DLMF	name of partner	...	serial_article	58J99 35C 35B99 14H 53C
115	1900-01-01T00:00:00	none	21.9#p2	DLMF scheme	https://dlmf.nist.gov/21.9#p2	{'Name': 'DLMF bibliographic entry'}	§21.9 Integrable Equations ‣ Applications ‣ Ch...	DLMF	DLMF	name of partner	...	serial_article	35Q53 35B 37K10 14H

From the returned articles we select only those where the primary MSC is "14".

In [46]:

```
alg_geo_prim=df[['Source.Identifier.ID','Target.Identifier.ID','Target.Type.Subtype']][df['Target.Type.Subtype'].str.startswith('14')]
alg_geo_prim
```

Out[46]:

	Source.Identifier.ID	Target.Identifier.ID	Target.Type.Subtype
0	21.3#i.info	0509.14049	14K25 11F11 33E05 14-02 14K10 14H10 14K30
14	21.7#i.p4	0549.14014	14K25 14K30 58J60 14-02 33E05 14H40 58J15 35Q99
16	21.8#p1	0251.14016	14-02 14K25 32N05
17	21#info	0251.14016	14-02 14K25 32N05
18	21#info	0509.14049	14K25 11F11 33E05 14-02 14K10 14H10 14K30
20	21.10#l1.i1.p1	0964.14047	14Q15 30F10 14H55 32G20 14-04
24	21.7#iii.info	0549.14014	14K25 14K30 58J60 14-02 33E05 14H40 58J15 35Q99
28	21.3#ii.info	0509.14049	14K25 11F11 33E05 14-02 14K10 14H10 14K30
30	21.5#ii.info	0251.14016	14-02 14K25 32N05
33	21.5#i.info	0509.14049	14K25 11F11 33E05 14-02 14K10 14H10 14K30
37	21#info	0549.14014	14K25 14K30 58J60 14-02 33E05 14H40 58J15 35Q99
38	21.6#i.info	0509.14049	14K25 11F11 33E05 14-02 14K10 14H10 14K30
47	21.7#ii.info	0549.14014	14K25 14K30 58J60 14-02 33E05 14H40 58J15 35Q99
52	21.6#ii.p1	0347.14023	14K25
60	22.18#iv.p1	0615.14018	14Hxx 14-02 11-02 11R58 11F03 14H05 11G15 14K2...
61	19#info	1105.14001	14-02 33-02 14K25 33E05 11E25 14-01 33-01 14K2...
73	21.10#l1.i3.p1	1054.14079	14Q05 14H70 30-04 30F30 35B10
74	32.2#iv.p1	0948.14025	14H52 34M55 34M15 14N35 14K20
75	21.8#p1	0743.14033	14K20 14H05 33E05 14-01 32N05
77	21.7#i.p4	0588.14019	14Hxx 14-02 14-03 14H20 30F10 14-01 58C15 51N1...
79	22#info	1105.14001	14-02 33-02 14K25 33E05 11E25 14-01 33-01 14K2...
83	20#info	1105.14001	14-02 33-02 14K25 33E05 11E25 14-01 33-01 14K2...
103	20.9#ii.p2	0848.14012	14H05 14H42 14-02 14K25
107	23.20#ii.p6	0872.14041	14Q05 14H52 11Y16 68W30 14G35 14-04
109	23#info	1105.14001	14-02 33-02 14K25 33E05 11E25 14-01 33-01 14K2...

We group by target article to account for articles that are linked multiple times.

In [47]:

```
alg_geo_prim.groupby('Target.Identifier.ID').count()
```

Out[47]:

	Source.Identifier.ID	Target.Type.Subtype
Target.Identifier.ID		
0251.14016	3	3
0347.14023	1	1
0509.14049	5	5
0549.14014	4	4
0588.14019	1	1
0615.14018	1	1
0743.14033	1	1
0848.14012	1	1
0872.14041	1	1
0948.14025	1	1
0964.14047	1	1
1054.14079	1	1
1105.14001	4	4

In the end, we get the same number as in the statistics output.

In [48]:

```
_.count()
```

Out[48]:

```
Source.Identifier.ID    13
Target.Type.Subtype     13
dtype: int64
```