

THE_UR_Global_Peers_Research_Performance_Profile_update

January 10, 2020

1 Install packages

```
In [3]: import tensorflow as tf
```

```
In [10]: from sklearn.cluster import KMeans
```

```
In [12]: import pandas as pd
```

```
In [13]: import numpy as np
```

2 Plot distribution of USA universities CitationCounts

```
In [1112]: cd "C:\Users\jchen148\THE Rankings\Report to Jane"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane
```

```
In [1113]: citation = pd.read_csv('Updated_THE_Ranked_Universities_CitationCounts_2014_2018.csv')
citation.head()
```

```
Out[1113]:
```

	Country	CountryCode	Uid	UniversityName	\
0	United States	USA	508076	Harvard University	
1	United States	USA	508219	Stanford University	
2	Canada	CAN	501048	University of Toronto	
3	United States	USA	508094	Johns Hopkins University	
4	United Kingdom	GBR	315091	University of Oxford	

	metric	Citation2014	Citation2015	Citation2016	Citation2017	\
0	CitationCount	837994.0	686576.0	529054.0	363995.0	
1	CitationCount	404346.0	363567.0	282005.0	194136.0	
2	CitationCount	360177.0	344355.0	254462.0	172125.0	
3	CitationCount	324631.0	289131.0	218379.0	147459.0	
4	CitationCount	355751.0	313129.0	238271.0	149907.0	

```

        Citation2018
0      172830.0
1      95069.0
2      86219.0
3      77682.0
4      75747.0

```

```
In [1114]: totalcitation=citation['Citation2014']+citation['Citation2015']+citation['Citation2016']
```

```
In [1115]: citation['Total']=totalcitation
```

```
In [1116]: citation.head()
```

```
        citation.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1002 entries, 0 to 1001
Data columns (total 11 columns):
Country          1002 non-null object
CountryCode      1002 non-null object
Uid              1002 non-null int64
UniversityName   1002 non-null object
metric           1002 non-null object
Citation2014     998 non-null float64
Citation2015     995 non-null float64
Citation2016     998 non-null float64
Citation2017     1000 non-null float64
Citation2018     1001 non-null float64
Total            993 non-null float64
dtypes: float64(6), int64(1), object(4)
memory usage: 86.2+ KB

```

```
In [1119]: changedtype=lambda x: int(x)
```

```
In [31]: #citation.fillna(0)
```

```

for i in range(0,len(citation)):
    if citation.loc[i]['Citation2014'] is np.nan:
        print("yes")

```

```
In [1117]: citation['Citation2014'].isnull()
```

```
        citation=citation.fillna(0)
```

3 change all citationcount to int64

```
In [1120]: citation['Citation2018']=citation['Citation2018'].apply(changedtype)
```

```

In [1121]: citation['Citation2017']=citation['Citation2017'].apply(changedtype)
In [1122]: citation['Citation2016']=citation['Citation2016'].apply(changedtype)
In [1123]: citation['Citation2015']=citation['Citation2015'].apply(changedtype)
In [1124]: citation['Citation2014']=citation['Citation2014'].apply(changedtype)
In [1125]: citation.info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1002 entries, 0 to 1001
Data columns (total 11 columns):
Country          1002 non-null object
CountryCode      1002 non-null object
Uid              1002 non-null int64
UniversityName   1002 non-null object
metric           1002 non-null object
Citation2014     1002 non-null int64
Citation2015     1002 non-null int64
Citation2016     1002 non-null int64
Citation2017     1002 non-null int64
Citation2018     1002 non-null int64
Total            1002 non-null float64
dtypes: float64(1), int64(6), object(4)
memory usage: 86.2+ KB

```

```

In [1126]: citation.head()

```

```

Out[1126]:
   Country CountryCode  Uid  UniversityName \
0  United States      USA  508076  Harvard University
1  United States      USA  508219  Stanford University
2    Canada          CAN  501048  University of Toronto
3  United States      USA  508094  Johns Hopkins University
4  United Kingdom      GBR  315091  University of Oxford

   metric  Citation2014  Citation2015  Citation2016  Citation2017 \
0  CitationCount      837994      686576      529054      363995
1  CitationCount      404346      363567      282005      194136
2  CitationCount      360177      344355      254462      172125
3  CitationCount      324631      289131      218379      147459
4  CitationCount      355751      313129      238271      149907

   Citation2018  Total
0      172830  2590449.0
1       95069  1339123.0
2       86219  1217338.0
3       77682  1057282.0
4       75747  1132805.0

```

```
In [1127]: new=citation.sort_values(['CountryCode','Total'], ascending=False)
new.head()
```

```
Out[1127]:
```

	Country	CountryCode	Uid	UniversityName	\
334	South Africa	ZAF	115007	University of Pretoria	
307	South Africa	ZAF	115005	University of Johannesburg	
369	South Africa	ZAF	115001	North West University	
575	South Africa	ZAF	115010	University of the Western Cape	
586	South Africa	ZAF	115003	Tshwane University of Technology	

	metric	Citation2014	Citation2015	Citation2016	Citation2017	\
334	CitationCount	20169	20294	18564	12800	
307	CitationCount	13732	17059	16450	12228	
369	CitationCount	9833	8025	14378	13220	
575	CitationCount	9538	8008	7081	5315	
586	CitationCount	3215	2665	2844	3173	

	Citation2018	Total
334	5918	77745.0
307	6622	66091.0
369	5225	50681.0
575	2365	32307.0
586	2271	14168.0

4 Filtered the universities in USA

```
In [1128]: USdata=new[new['CountryCode']=='USA']
```

```
In [1129]: USdata.head()
```

```
Out[1129]:
```

	Country	CountryCode	Uid	UniversityName	\
0	United States	USA	508076	Harvard University	
1	United States	USA	508219	Stanford University	
3	United States	USA	508094	Johns Hopkins University	
6	United States	USA	508358	University of Washington	
11	United States	USA	508111	Massachusetts Institute of Technology	

	metric	Citation2014	Citation2015	Citation2016	Citation2017	\
0	CitationCount	837994	686576	529054	363995	
1	CitationCount	404346	363567	282005	194136	
3	CitationCount	324631	289131	218379	147459	
6	CitationCount	314702	269985	218378	150820	
11	CitationCount	285399	241655	186864	121031	

	Citation2018	Total
0	172830	2590449.0
1	95069	1339123.0
3	77682	1057282.0

```

6          70792  1024677.0
11         57319   892268.0

```

5 Use seaborn

```

In [1130]: import numpy as np
           import pandas as pd
           import seaborn as sns
           import matplotlib.pyplot as plt
           from scipy import stats

```

```

In [1131]: sns.set(color_codes=True)

```

```

In [1132]: USpartial=USdata.loc[:,['UniversityName','Total']]

```

```

In [1133]: USpartial.head()

```

```

USpartial2=USpartial.reset_index()

```

```

USpartial2=USpartial2.iloc[:,1:]

```

```

USpartial2.head()

```

```

Out[1133]:
           UniversityName      Total
0          Harvard University  2590449.0
1        Stanford University  1339123.0
2    Johns Hopkins University  1057282.0
3   University of Washington  1024677.0
4  Massachusetts Institute of Technology   892268.0

```

```

In [1134]: target=USpartial2[USpartial2['UniversityName']=='University of Rochester']

```

```

target.head()

```

```

Out[1134]:
           UniversityName      Total
28  University of Rochester  254555.0

```

6 Change datatype to int64

```

In [1135]: target.loc[:,['Total']]=target['Total'].astype(int)

```

```

In [1136]: target.head()

```

```

Out[1136]:
           UniversityName      Total
28  University of Rochester  254555.0

```

```

In [1137]: USpartial2.head()

```

```

USpartial2.set_index('UniversityName')

```

```

USpartial2.loc[:,['Total']]=USpartial2['Total'].astype(int)

```

```
In [1138]: USpartial2=USpartial2.set_index('UniversityName')
```

```
In [1139]: USpartial2.head()
```

```
Out[1139]:
```

	Total
UniversityName	
Harvard University	2590449.0
Stanford University	1339123.0
Johns Hopkins University	1057282.0
University of Washington	1024677.0
Massachusetts Institute of Technology	892268.0

```
In [1140]: target.head()
```

```
target.set_index('UniversityName')
```

```
Out[1140]:
```

	Total
UniversityName	
University of Rochester	254555.0

```
In [1141]: target=target.set_index('UniversityName')
```

```
In [1142]: target.head()
```

```
Out[1142]:
```

	Total
UniversityName	
University of Rochester	254555.0

```
In [1143]: len(USpartial2)
```

```
Out[1143]: 153
```

7 THE has 163 USA Universities ranking in top 300.

8 Below is the distribution plot of the total CitationCount

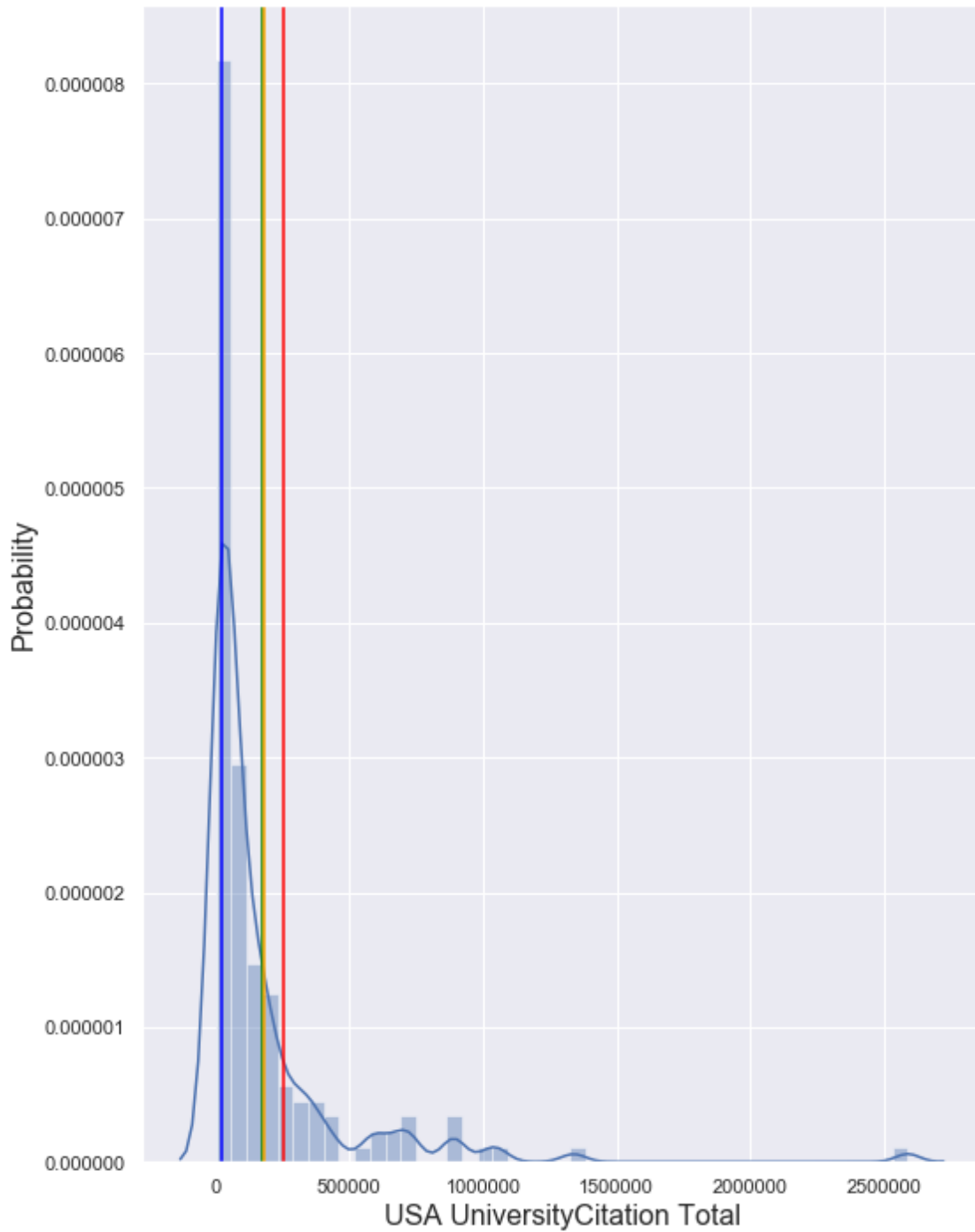
9 from 2014 to 2018.

10 And we can see where UofR lies.

```
In [1144]: import pandas as pd
fig, ax = plt.subplots(figsize=(8,10))
x = pd.Series(USpartial2['Total'], name="CitationCount Total")
ax = sns.distplot(x)

ax.set_xlabel("USA UniversityCitation Total",fontsize=16)
ax.set_ylabel("Probability",fontsize=16)
plt.axvline(254555, color='red') # this is where U of R
```

```
plt.axvline(np.mean(USpartial2['Total']), color='green') # this is the mean, 175882
plt.axvline(np.percentile(USpartial2['Total'], 25.0), color='blue') # Q1
plt.axvline(np.percentile(USpartial2['Total'], 75.0), color='orange') # Q3
#plt.legend()
plt.tight_layout()
```



- 11 We can see it is a highly right-skewed distribution,
- 12 and the mean, which is the green line, and Q3,
- 13 which is the orange line are very close.
- 14 UofR has the CitationCounts much above Q3.

```
In [147]: import matplotlib
          from matplotlib import mlab
          import matplotlib.pyplot as plt
          import numpy as np
          import matplotlib.dates as mdates
          import matplotlib.pyplot as plt
```

```
In [152]: np.percentile(USpartial2['Total'], np.array([25.0,75.0]))
```

```
Out[152]: array([ 24316., 180027.])
```

```
In [122]: target
```

```
Out[122]:
```

	Total
UniversityName	
University of Rochester	254555

```
In [136]: np.round(np.mean(USpartial2['Total']), 2)
```

```
Out[136]: 175882.56
```

- 15 The following are data cleaning process,
- 16 and how to use Python Requests to retrieve
- 17 data from Scopus and SciVal REST APIs

```
In [ ]: school_list = open(r"C:\Users\jchen148\THE Rankings\Report to Jane\THE_School_List_OK.txt")
        school_name=school_list.read()
```

```
In [ ]: import pandas as pd
        t = school_name

        data=[]

        for i in t.split("\n"):
            if i[:1].isdigit():
                data.append(" ".join(i.split(" ")[1:20]))
                print(" ".join(i.split(" ")[1:20]))
```



```

data_want = pd.DataFrame(data, columns=['Scool Name'])

data_want.to_csv("all_university_name.csv", index=False) # all the university name

In [ ]: # cleaned all the ranks and leadning and trailing whitespace

t = school_name

uni_name = []

for i in t.split("\n"):
    if i[:1].isdigit():
        uni_name.append(" ".join(i.split(" ")[-5:]))
        print(" ".join(i.split(" ")[-5:]))
        uni_name.append(" ".join(i.split(" ")[-5:]))

In [ ]: # remove trailing whitespace

import re
import string

cleaned=[]

for line in uni_name:
    line=str(line)
    # print(line.strip(' \t\n\r'))
    # print(line.rstrip(string.digits))
    # print(re.sub('^d+[\W_]+', '', line))
    want_data = re.sub('^d+[\W_]+', '', line)
    print(want_data.strip())
    cleaned.append(want_data.strip())

In [ ]: # remove existing numbers

import string
import re

want_3=[]

for name in cleaned:
    print(name)
    print(re.sub('^d+[\W_]+', '', name))
    want_3.append(re.sub('^d+[\W_]+', '', name))

In [11]: want_3.append('University of Rochester')

```

```

In [15]: DF={}

          DF=pd.DataFrame({'UniName':want_3})

In [17]: DF=DF.drop_duplicates()

In [19]: DF=DF.reset_index()

In [21]: DF=DF.iloc[:,1]

In [25]: DF=pd.DataFrame(DF)

In [26]: DF.to_csv("UniNameList_OK.csv", index=False)

```

18 Use APIs

```

In [ ]: for line in want_3:
        url= "https://api.elsevier.com/metrics/institution/search?query=name("+line+)"
        #     print(url)

```

19 Combine all the Uids to retrieve data from APIs

```

In [11]: import requests
         import json

In [14]: # add "Emory University" country code and university id

          UniversityName=[]
          Universityid=[]
          Country=[]
          CountryCode=[]

          url='https://api.elsevier.com/analytics/scival/institution/search?query=name(Emory%20U)

          resp = requests.get(url, headers={'Accept': 'application/json',
                                             'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})

          parsed=json.dumps(resp.json(),
                             sort_keys=True,
                             indent=4, separators=(',', ': '))

          #     with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
          #         json.dump(resp.json(), jsonfile)
          #     print(parsed)
          #     data.update(a_dict)
          result=json.loads(parsed)
          UniversityName.append(result['results'][0]['name'])
          Universityid.append(result['results'][0]['id'])
          Country.append(result['results'][0]['country'])
          CountryCode.append(result['results'][0]['countryCode'])

```

```
ELmory=pd.DataFrame({'University Name':UniversityName, 'University id':Universityid,
                     'Country Code': CountryCode})
```

```
In [34]: filename='THE_CountryCode_Result_1202_{}'
```

```
for i in range(1,14):
    print(filename.format(i))
```

```
THE_CountryCode_Result_1202_1
THE_CountryCode_Result_1202_2
THE_CountryCode_Result_1202_3
THE_CountryCode_Result_1202_4
THE_CountryCode_Result_1202_5
THE_CountryCode_Result_1202_6
THE_CountryCode_Result_1202_7
THE_CountryCode_Result_1202_8
THE_CountryCode_Result_1202_9
THE_CountryCode_Result_1202_10
THE_CountryCode_Result_1202_11
THE_CountryCode_Result_1202_12
THE_CountryCode_Result_1202_13
```

```
In [47]: cd "C:\Users\jchen148\THE Rankings\Report to Jane"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane
```

```
In [48]: filename='THE_CountryCode_Result_1202_{}.csv'
```

```
chucks=[]

for i in range(1,14):
    # print(filename.format(i))
    chucks.append(pd.read_csv(filename.format(i)))

data=pd.concat(chucks, ignore_index=True)

data.head()
```

```
Out[48]:
```

	Unnamed: 0	University Name \
0	0	University of Oxford
1	1	Jet Propulsion Laboratory, California Institut...
2	2	California Institute of Technology
3	3	University of Cambridge
4	4	Stanford University

	University id	Country	Country Code
--	---------------	---------	--------------

0	315091	United Kingdom	GBR
1	508092	United States	USA
2	508021	United States	USA
3	315068	United Kingdom	GBR
4	508219	United States	USA

In [49]: `len(data)`

Out[49]: 1272

In [50]: `del data['Unnamed: 0']`

In [26]: `ELmory`

Out[26]:

	University Name	University id	Country	Country Code
0	Emory University	508059	United States	USA

In [51]: `data=pd.concat([data, ELmory]).drop_duplicates()`

In [52]: `data.head()`

Out[52]:

	University Name	University id	\
0	University of Oxford	315091	
1	Jet Propulsion Laboratory, California Institut...	508092	
2	California Institute of Technology	508021	
3	University of Cambridge	315068	
4	Stanford University	508219	

	Country	Country Code
0	United Kingdom	GBR
1	United States	USA
2	United States	USA
3	United Kingdom	GBR
4	United States	USA

In [31]: `cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Input Data"`

C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Input Data

In [32]: `data.to_csv("THE_Universities_SciVal_Uids_1008.csv", index=False)`

20 Use SciVal institution metrics API

In [42]: `# https://api.elsevier.com/analytics/scival/institution/metrics`

In [53]: `data.tail()`

```
Out [53]:
```

	University Name	University id	\
1268	Imam Abdulrahman Bin Faisal University	703099	
1269	Istanbul Medipol University	705124	
1270	Ivane Javakhishvili Tbilisi State University	204001	
1271	University of Rochester	508335	
0	Emory University	508059	

	Country	Country Code
1268	Saudi Arabia	SAU
1269	Turkey	TUR
1270	Georgia	GEO
1271	United States	USA
0	United States	USA

```
In [54]: data.reset_index(inplace=True)
```

```
In [57]: data=data.iloc[:,1:]
```

```
In [58]: data.tail()
```

```
Out [58]:
```

	University Name	University id	\
1003	Imam Abdulrahman Bin Faisal University	703099	
1004	Istanbul Medipol University	705124	
1005	Ivane Javakhishvili Tbilisi State University	204001	
1006	University of Rochester	508335	
1007	Emory University	508059	

	Country	Country Code
1003	Saudi Arabia	SAU
1004	Turkey	TUR
1005	Georgia	GEO
1006	United States	USA
1007	United States	USA

```
In [64]: for line in data['University id'][:2]:
          print(line)
```

```
315091
508092
```

21 ScholarlyOutput

```
In [59]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\ScholarlyOutput"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\ScholarlyOutput
```

```

In [60]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(2)

inst_country=[]
inst_cc=[]
inst_id=[]
inst_link=[]
inst_name=[]
metricType=[]
value2014=[]
value2015=[]
value2016=[]
value2017=[]
value2018=[]

for line in data['University id'][1000:]:
    url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Ci
    # print(url.format(line))
    resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                    'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
    parsed=json.dumps(resp.json(),
                       sort_keys=True,
                       indent=4, separators=(',', ': '))
    # with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
    #     json.dump(resp.json(), jsonfile)
    # print(parsed)
    # data.update(a_dict)
    result=json.loads(parsed)
    if result['results'] is not None:
        if len(result['results'])>=1:
            if 'institution' in result['results'][0]:
                # if 'country' in result['results'][0]['institution']:
                inst_country.append(result['results'][0]['institution']['country'])
                # if 'countryCode' in result['results'][0]['institution']:
                inst_cc.append(result['results'][0]['institution']['countryCode'])
                # if 'id' in result['results'][0]['institution']:
                inst_id.append(result['results'][0]['institution']['id'])
                # if 'link' in result['results'][0]['institution']:
                inst_link.append(result['results'][0]['institution']['link'])
                # if 'name' in result['results'][0]['institution']:
                inst_name.append(result['results'][0]['institution']['name'])
            if 'metrics' in result['results'][0]:
                # if len(result['results'][0]['metrics'])>=1:

```

```

        if 'metricType' in result['results'][0]['metrics'][0]:
            metricType.append(result['results'][0]['metrics'][0]['metricType'])
        if 'valueByYear' in result['results'][0]['metrics'][0]:
            if '2014' in result['results'][0]['metrics'][0]['valueByYear']:
                value2014.append(result['results'][0]['metrics'][0]['valueByYear'])
            if '2015' in result['results'][0]['metrics'][0]['valueByYear']:
                value2015.append(result['results'][0]['metrics'][0]['valueByYear'])
            if '2016' in result['results'][0]['metrics'][0]['valueByYear']:
                value2016.append(result['results'][0]['metrics'][0]['valueByYear'])
            if '2017' in result['results'][0]['metrics'][0]['valueByYear']:
                value2017.append(result['results'][0]['metrics'][0]['valueByYear'])
            if '2018' in result['results'][0]['metrics'][0]['valueByYear']:
                value2018.append(result['results'][0]['metrics'][0]['valueByYear'])

s1=pd.Series(inst_country, name='country')
s2=pd.Series(inst_cc, name='countryCode')
s3=pd.Series(inst_id, name='institution_id')
s4=pd.Series(inst_link, name='link')
s5=pd.Series(inst_name, name='institution_name')
s6=pd.Series(metricType, name='metricType')
s7=pd.Series(value2014, name='2014')
s8=pd.Series(value2015, name='2015')
s9=pd.Series(value2016, name='2016')
s10=pd.Series(value2017, name='2017')
s11=pd.Series(value2018, name='2018')

DF=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11], axis=1)
DF.to_csv("THE_UNI_CitationCount_ALL_12.csv", index=False)

```

22 CitationCount, CitedPublications, FWCI, and Publicationin-TopJournal Percentile

In [122]: # FWCI

In [61]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\FNCI"

C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\FNCI

```

In [63]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(2)

inst_country=[]

```

```

inst_cc=[]
inst_id=[]
inst_link=[]
inst_name=[]
metricType=[]
value2014=[]
value2015=[]
value2016=[]
value2017=[]
value2018=[]
percentage2014=[]
percentage2015=[]
percentage2016=[]
percentage2017=[]
percentage2018=[]

for line in data['University id'][1000:]:
    url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Fi
    # print(url.format(line))
    resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                    'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
    parsed=json.dumps(resp.json(),
                       sort_keys=True,
                       indent=4, separators=(',', ': '))
    # with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
    #     json.dump(resp.json(), jsonfile)
    # print(parsed)
    # data.update(a_dict)
    result=json.loads(parsed)
    if result['results'] is not None:
        if len(result['results'])>=1:
            if 'institution' in result['results'][0]:
                # if 'country' in result['results'][0]['institution']:
                #     inst_country.append(result['results'][0]['institution']['country'])
                # if 'countryCode' in result['results'][0]['institution']:
                #     inst_cc.append(result['results'][0]['institution']['countryCode'])
                # if 'id' in result['results'][0]['institution']:
                #     inst_id.append(result['results'][0]['institution']['id'])
                # if 'link' in result['results'][0]['institution']:
                #     inst_link.append(result['results'][0]['institution']['link'])
                # if 'name' in result['results'][0]['institution']:
                #     inst_name.append(result['results'][0]['institution']['name'])
            if 'metrics' in result['results'][0]:
                # if len(result['results'][0]['metrics'])>=1:
                #     if 'metricType' in result['results'][0]['metrics'][0]:
                #         metricType.append(result['results'][0]['metrics'][0]['metricType'])
                #     if 'valueByYear' in result['results'][0]['metrics'][0]:

```



```

        if '2014' in result['results'][0]['metrics'][0]['valueByYear']:
            value2014.append(result['results'][0]['metrics'][0]['valueByYear'])
        if '2015' in result['results'][0]['metrics'][0]['valueByYear']:
            value2015.append(result['results'][0]['metrics'][0]['valueByYear'])
        if '2016' in result['results'][0]['metrics'][0]['valueByYear']:
            value2016.append(result['results'][0]['metrics'][0]['valueByYear'])
        if '2017' in result['results'][0]['metrics'][0]['valueByYear']:
            value2017.append(result['results'][0]['metrics'][0]['valueByYear'])
        if '2018' in result['results'][0]['metrics'][0]['valueByYear']:
            value2018.append(result['results'][0]['metrics'][0]['valueByYear'])
    if 'percentageByYear' in result['results'][0]['metrics'][0]:
        if '2014' in result['results'][0]['metrics'][0]['percentageByYear']:
            percentage2014.append(result['results'][0]['metrics'][0]['percentageByYear'])
        if '2015' in result['results'][0]['metrics'][0]['percentageByYear']:
            percentage2015.append(result['results'][0]['metrics'][0]['percentageByYear'])
        if '2016' in result['results'][0]['metrics'][0]['percentageByYear']:
            percentage2016.append(result['results'][0]['metrics'][0]['percentageByYear'])
        if '2017' in result['results'][0]['metrics'][0]['percentageByYear']:
            percentage2017.append(result['results'][0]['metrics'][0]['percentageByYear'])
        if '2018' in result['results'][0]['metrics'][0]['percentageByYear']:
            percentage2018.append(result['results'][0]['metrics'][0]['percentageByYear'])
    else:
        percentage2014.append('')
        percentage2015.append('')
        percentage2016.append('')
        percentage2017.append('')
        percentage2018.append('')

```

```

s1=pd.Series(inst_country, name='country')
s2=pd.Series(inst_cc, name='countryCode')
s3=pd.Series(inst_id, name='institution_id')
s4=pd.Series(inst_link, name='link')
s5=pd.Series(inst_name, name='institution_name')
s6=pd.Series(metricType, name='metricType')
s7=pd.Series(value2014, name='2014')
s8=pd.Series(value2015, name='2015')
s9=pd.Series(value2016, name='2016')
s10=pd.Series(value2017, name='2017')
s11=pd.Series(value2018, name='2018')
#s12=pd.Series(percentage2014, name='percent2014')
#s13=pd.Series(percentage2015, name='percent2015')
#s14=pd.Series(percentage2016, name='percent2016')
#s15=pd.Series(percentage2017, name='percent2017')
#s16=pd.Series(percentage2018, name='percent2018')

```

```

DF=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11], axis=1)

```

```
DF.to_csv("THE_UNI_FWCI_12.csv", index=False)
```

```
In [134]: # CitationCount
```

```
In [133]: import requests
```

```
import json
```

```
import pandas as pd
```

```
import numpy as np
```

```
from time import sleep
```

```
sleep(2)
```

```
inst_country=[]
```

```
inst_cc=[]
```

```
inst_id=[]
```

```
inst_link=[]
```

```
inst_name=[]
```

```
metricType=[]
```

```
value2014=[]
```

```
value2015=[]
```

```
value2016=[]
```

```
value2017=[]
```

```
value2018=[]
```

```
percentage2014=[]
```

```
percentage2015=[]
```

```
percentage2016=[]
```

```
percentage2017=[]
```

```
percentage2018=[]
```

```
for line in data['University id'][1000:]:
```

```
url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=F
```

```
# print(url.format(line))
```

```
resp = requests.get(url.format(line), headers={'Accept': 'application/json',  
        'X-ELS-APIKey': "a464321ef5063d696ada17f8c159a44c"})
```

```
parsed=json.dumps(resp.json(),
```

```
        sort_keys=True,
```

```
        indent=4, separators=(',', ' '))
```

```
# with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
```

```
# json.dump(resp.json(), jsonfile)
```

```
# print(parsed)
```

```
# data.update(a_dict)
```

```
result=json.loads(parsed)
```

```
if result['results'] is not None:
```

```
    if len(result['results'])>=1:
```

```
        if 'institution' in result['results'][0]:
```

```
#             if 'country' in result['results'][0]['institution']:
```

```
                inst_country.append(result['results'][0]['institution']['country'])
```

```
#             if 'countryCode' in result['results'][0]['institution']:
```

```

        inst_cc.append(result['results'][0]['institution']['countryCode'])
#     if 'id' in result['results'][0]['institution']:
        inst_id.append(result['results'][0]['institution']['id'])
#     if 'link' in result['results'][0]['institution']:
        inst_link.append(result['results'][0]['institution']['link'])
#     if 'name' in result['results'][0]['institution']:
        inst_name.append(result['results'][0]['institution']['name'])
if 'metrics' in result['results'][0]:
#     if len(result['results'][0]['metrics'])>=1:
        if 'metricType' in result['results'][0]['metrics'][0]:
            metricType.append(result['results'][0]['metrics'][0]['metricType'])
        if 'valueByYear' in result['results'][0]['metrics'][0]:
            if '2014' in result['results'][0]['metrics'][0]['valueByYear']:
                value2014.append(result['results'][0]['metrics'][0]['valueByYear'])
            if '2015' in result['results'][0]['metrics'][0]['valueByYear']:
                value2015.append(result['results'][0]['metrics'][0]['valueByYear'])
            if '2016' in result['results'][0]['metrics'][0]['valueByYear']:
                value2016.append(result['results'][0]['metrics'][0]['valueByYear'])
            if '2017' in result['results'][0]['metrics'][0]['valueByYear']:
                value2017.append(result['results'][0]['metrics'][0]['valueByYear'])
            if '2018' in result['results'][0]['metrics'][0]['valueByYear']:
                value2018.append(result['results'][0]['metrics'][0]['valueByYear'])
        if 'percentageByYear' in result['results'][0]['metrics'][0]:
            if '2014' in result['results'][0]['metrics'][0]['percentageByYear']:
                percentage2014.append(result['results'][0]['metrics'][0]['percentageByYear'])
            if '2015' in result['results'][0]['metrics'][0]['percentageByYear']:
                percentage2015.append(result['results'][0]['metrics'][0]['percentageByYear'])
            if '2016' in result['results'][0]['metrics'][0]['percentageByYear']:
                percentage2016.append(result['results'][0]['metrics'][0]['percentageByYear'])
            if '2017' in result['results'][0]['metrics'][0]['percentageByYear']:
                percentage2017.append(result['results'][0]['metrics'][0]['percentageByYear'])
            if '2018' in result['results'][0]['metrics'][0]['percentageByYear']:
                percentage2018.append(result['results'][0]['metrics'][0]['percentageByYear'])
        else:
            percentage2014.append('')
            percentage2015.append('')
            percentage2016.append('')
            percentage2017.append('')
            percentage2018.append('')

s1=pd.Series(inst_country, name='country')
s2=pd.Series(inst_cc, name='countryCode')
s3=pd.Series(inst_id, name='institution_id')
s4=pd.Series(inst_link, name='link')
s5=pd.Series(inst_name, name='institution_name')
s6=pd.Series(metricType, name='metricType')
s7=pd.Series(value2014, name='2014')

```

```

s8=pd.Series(value2015, name='2015')
s9=pd.Series(value2016, name='2016')
s10=pd.Series(value2017, name='2017')
s11=pd.Series(value2018, name='2018')
#s12=pd.Series(percentage2014, name='percent2014')
#s13=pd.Series(percentage2015, name='percent2015')
#s14=pd.Series(percentage2016, name='percent2016')
#s15=pd.Series(percentage2017, name='percent2017')
#s16=pd.Series(percentage2018, name='percent2018')

DF=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11], axis=1)
DF.to_csv("THE_UNI_FWCI_11.csv", index=False)

```

```
In [146]: # CitedPublications
```

```
In [64]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\PercPublsCited"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\PercPublsCited
```

```
In [65]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(2)

inst_country=[]
inst_cc=[]
inst_id=[]
inst_link=[]
inst_name=[]
metricType=[]
value2014=[]
value2015=[]
value2016=[]
value2017=[]
value2018=[]
percentage2014=[]
percentage2015=[]
percentage2016=[]
percentage2017=[]
percentage2018=[]

for line in data['University id'][1000:]:
    url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Ci
    # print(url.format(line))

```

```

resp = requests.get(url.format(line), headers={'Accept': 'application/json',
                                              'X-ELS-APIKey': 'd3794058e2b24417b5dfd0ef8990e2dc'})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ': '))
# with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
#     json.dump(resp.json(), jsonfile)
# print(parsed)
# data.update(a_dict)
result=json.loads(parsed)
if 'results' in result:
    if len(result['results'])>=1:
        if 'institution' in result['results'][0]:
#             if 'country' in result['results'][0]['institution']:
inst_country.append(result['results'][0]['institution']['country'])
#             if 'countryCode' in result['results'][0]['institution']:
inst_cc.append(result['results'][0]['institution']['countryCode'])
#             if 'id' in result['results'][0]['institution']:
inst_id.append(result['results'][0]['institution']['id'])
#             if 'link' in result['results'][0]['institution']:
inst_link.append(result['results'][0]['institution']['link'])
#             if 'name' in result['results'][0]['institution']:
inst_name.append(result['results'][0]['institution']['name'])
if 'metrics' in result['results'][0]:
#             if len(result['results'][0]['metrics'])>=1:
if 'metricType' in result['results'][0]['metrics'][0]:
    metricType.append(result['results'][0]['metrics'][0]['metricType'])
if 'valueByYear' in result['results'][0]['metrics'][0]:
    if '2014' in result['results'][0]['metrics'][0]['valueByYear']:
        value2014.append(result['results'][0]['metrics'][0]['valueByYear'])
    if '2015' in result['results'][0]['metrics'][0]['valueByYear']:
        value2015.append(result['results'][0]['metrics'][0]['valueByYear'])
    if '2016' in result['results'][0]['metrics'][0]['valueByYear']:
        value2016.append(result['results'][0]['metrics'][0]['valueByYear'])
    if '2017' in result['results'][0]['metrics'][0]['valueByYear']:
        value2017.append(result['results'][0]['metrics'][0]['valueByYear'])
    if '2018' in result['results'][0]['metrics'][0]['valueByYear']:
        value2018.append(result['results'][0]['metrics'][0]['valueByYear'])
if 'percentageByYear' in result['results'][0]['metrics'][0]:
    if '2014' in result['results'][0]['metrics'][0]['percentageByYear']:
        percentage2014.append(result['results'][0]['metrics'][0]['percentageByYear'])
    if '2015' in result['results'][0]['metrics'][0]['percentageByYear']:
        percentage2015.append(result['results'][0]['metrics'][0]['percentageByYear'])
    if '2016' in result['results'][0]['metrics'][0]['percentageByYear']:
        percentage2016.append(result['results'][0]['metrics'][0]['percentageByYear'])
    if '2017' in result['results'][0]['metrics'][0]['percentageByYear']:
        percentage2017.append(result['results'][0]['metrics'][0]['percentageByYear'])
    if '2018' in result['results'][0]['metrics'][0]['percentageByYear']:

```

```

        percentage2018.append(result['results'][0]['metrics'][0]['per
else:
    percentage2014.append('')
    percentage2015.append('')
    percentage2016.append('')
    percentage2017.append('')
    percentage2018.append('')

s1=pd.Series(inst_country, name='country')
s2=pd.Series(inst_cc, name='countryCode')
s3=pd.Series(inst_id, name='institution_id')
s4=pd.Series(inst_link, name='link')
s5=pd.Series(inst_name, name='institution_name')
s6=pd.Series(metricType, name='metricType')
s7=pd.Series(value2014, name='2014')
s8=pd.Series(value2015, name='2015')
s9=pd.Series(value2016, name='2016')
s10=pd.Series(value2017, name='2017')
s11=pd.Series(value2018, name='2018')
s12=pd.Series(percentage2014, name='percent2014')
s13=pd.Series(percentage2015, name='percent2015')
s14=pd.Series(percentage2016, name='percent2016')
s15=pd.Series(percentage2017, name='percent2017')
s16=pd.Series(percentage2018, name='percent2018')

DF=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,s16], axis=1)
DF.to_csv("THE_UNI_CitedPublications_12.csv", index=False)

```

In [160]: *#PublicationsInTopJournalPercentiles*

```

In [186]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(2)

inst_country=[]
inst_cc=[]
inst_id=[]
inst_link=[]
inst_name=[]
metricType=[]
threshold=[]
t1_value2014=[]
t1_value2015=[]

```

```

t1_value2016=[]
t1_value2017=[]
t1_value2018=[]
t1_percentage2014=[]
t1_percentage2015=[]
t1_percentage2016=[]
t1_percentage2017=[]
t1_percentage2018=[]
t5_value2014=[]
t5_value2015=[]
t5_value2016=[]
t5_value2017=[]
t5_value2018=[]
t5_percentage2014=[]
t5_percentage2015=[]
t5_percentage2016=[]
t5_percentage2017=[]
t5_percentage2018=[]
t10_value2014=[]
t10_value2015=[]
t10_value2016=[]
t10_value2017=[]
t10_value2018=[]
t10_percentage2014=[]
t10_percentage2015=[]
t10_percentage2016=[]
t10_percentage2017=[]
t10_percentage2018=[]
t25_value2014=[]
t25_value2015=[]
t25_value2016=[]
t25_value2017=[]
t25_value2018=[]
t25_percentage2014=[]
t25_percentage2015=[]
t25_percentage2016=[]
t25_percentage2017=[]
t25_percentage2018=[]

```

```

for line in data['University id'][:2]:
    url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=P
    # print(url.format(line))
    resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                    'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
    parsed=json.dumps(resp.json(),
                       sort_keys=True,

```

```

        indent=4, separators=(',', ' : '))
#     with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
#         json.dump(resp.json(), jsonfile)
#     print(parsed)
#     data.update(a_dict)
result=json.loads(parsed)
if 'results' in result:
    if len(result['results'])>=1:
        if 'institution' in result['results'][0]:
#             if 'country' in result['results'][0]['institution']:
#                 inst_country.append(result['results'][0]['institution']['country'])
#             if 'countryCode' in result['results'][0]['institution']:
#                 inst_cc.append(result['results'][0]['institution']['countryCode'])
#             if 'id' in result['results'][0]['institution']:
#                 inst_id.append(result['results'][0]['institution']['id'])
#             if 'link' in result['results'][0]['institution']:
#                 inst_link.append(result['results'][0]['institution']['link'])
#             if 'name' in result['results'][0]['institution']:
#                 inst_name.append(result['results'][0]['institution']['name'])
        if 'metrics' in result['results'][0]:
#             if len(result['results'][0]['metrics'])>=1:
#                 if 'metricType' in result['results'][0]['metrics'][0]:
#                     metricType.append(result['results'][0]['metrics'][0]['metricType'])
#                 if 'values' in result['results'][0]['metrics'][0]:
#                     if 'threshold' in result['results'][0]['metrics'][0]['values']:
#                         threshold.append(result['results'][0]['metrics'][0]['values'])
#                     if 'valueByYear' in result['results'][0]['metrics'][0]['values']:
#                         if '2014' in result['results'][0]['metrics'][0]['values']:
#                             t1_value2014.append(result['results'][0]['metrics'][0]['values'])
#                         if '2015' in result['results'][0]['metrics'][0]['values']:
#                             t1_value2015.append(result['results'][0]['metrics'][0]['values'])
#                         if '2016' in result['results'][0]['metrics'][0]['values']:
#                             t1_value2016.append(result['results'][0]['metrics'][0]['values'])
#                         if '2017' in result['results'][0]['metrics'][0]['values']:
#                             t1_value2017.append(result['results'][0]['metrics'][0]['values'])
#                         if '2018' in result['results'][0]['metrics'][0]['values']:
#                             t1_value2018.append(result['results'][0]['metrics'][0]['values'])
#                     if 'percentageByYear' in result['results'][0]['metrics'][0]['values']:
#                         if '2014' in result['results'][0]['metrics'][0]['values']:
#                             t1_percentage2014.append(result['results'][0]['metrics'][0]['values'])
#                         if '2015' in result['results'][0]['metrics'][0]['values']:
#                             t1_percentage2015.append(result['results'][0]['metrics'][0]['values'])
#                         if '2016' in result['results'][0]['metrics'][0]['values']:
#                             t1_percentage2016.append(result['results'][0]['metrics'][0]['values'])
#                         if '2017' in result['results'][0]['metrics'][0]['values']:
#                             t1_percentage2017.append(result['results'][0]['metrics'][0]['values'])
#                         if '2018' in result['results'][0]['metrics'][0]['values']:
#                             t1_percentage2018.append(result['results'][0]['metrics'][0]['values'])

```



```

#                                     else:
#                                     t1_value2014.append('')
#                                     t1_value2015.append('')
#                                     t1_value2016.append('')
#                                     t1_value2017.append('')
#                                     t1_value2018.append('')
#                                     t1_percentage2014.append('')
#                                     t1_percentage2015.append('')
#                                     t1_percentage2016.append('')
#                                     t1_percentage2017.append('')
#                                     t1_percentage2018.append('')

```

```

s1=pd.Series(inst_country, name='country')
s2=pd.Series(inst_cc, name='countryCode')
s3=pd.Series(inst_id, name='institution_id')
s4=pd.Series(inst_link, name='link')
s5=pd.Series(inst_name, name='institution_name')
s6=pd.Series(metricType, name='metricType')
s7=pd.Series(threshold, name='threshold')
s8=pd.Series(t1_value2014, name='2014')
s9=pd.Series(t1_value2015, name='2015')
s10=pd.Series(t1_value2016, name='2016')
s11=pd.Series(t1_value2017, name='2017')
s12=pd.Series(t1_value2018, name='2018')
s13=pd.Series(t1_percentage2014, name='percent2014')
s14=pd.Series(t1_percentage2015, name='percent2015')
s15=pd.Series(t1_percentage2016, name='percent2016')
s16=pd.Series(t1_percentage2017, name='percent2017')
s17=pd.Series(t1_percentage2018, name='percent2018')

```

```

DF=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,s16, s17], axis=1)
DF.to_csv("THE_UNI_PublicationsInTopJournalPercentiles_TEST_1.csv", index=False)

```

```

In [206]: metricType=[]
          threshold=[]
          value2014=[]
          value2015=[]
          value2016=[]
          value2017=[]
          value2018=[]
          percent2014=[]
          percent2015=[]
          percent2016=[]
          percent2017=[]
          percent2018=[]

```

```

for line in data['University id'][:2]:
    url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=P
    # print(url.format(line))
    resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                    'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
    parsed=json.dumps(resp.json(),
                       sort_keys=True,
                       indent=4, separators=(',', ': '))
    # with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
    #     json.dump(resp.json(), jsonfile)
    # print(parsed)
    # data.update(a_dict)
    result=json.loads(parsed)
    print(result['results'][0]['metrics'][0]['values'][3]['percentageByYear'])

```

```
{'2014': 67.55675, '2015': 73.333336, '2016': 67.42509, '2017': 66.53675, '2018': 64.18532}
```

```

In [214]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(2)

```

```

inst_country=[]
inst_cc=[]
inst_id=[]
inst_link=[]
inst_name=[]
metricType=[]
threshold=[]
t1_value2014=[]
t1_value2015=[]
t1_value2016=[]
t1_value2017=[]
t1_value2018=[]
t1_percentage2014=[]
t1_percentage2015=[]
t1_percentage2016=[]
t1_percentage2017=[]
t1_percentage2018=[]
t5_value2014=[]
t5_value2015=[]
t5_value2016=[]
t5_value2017=[]
t5_value2018=[]
t5_percentage2014=[]

```

```

t5_percentage2015=[]
t5_percentage2016=[]
t5_percentage2017=[]
t5_percentage2018=[]
t10_value2014=[]
t10_value2015=[]
t10_value2016=[]
t10_value2017=[]
t10_value2018=[]
t10_percentage2014=[]
t10_percentage2015=[]
t10_percentage2016=[]
t10_percentage2017=[]
t10_percentage2018=[]
t25_value2014=[]
t25_value2015=[]
t25_value2016=[]
t25_value2017=[]
t25_value2018=[]
t25_percentage2014=[]
t25_percentage2015=[]
t25_percentage2016=[]
t25_percentage2017=[]
t25_percentage2018=[]

```

```

for line in data['University id'][50:75]:
    url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=P
    # print(url.format(line))
    resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                    'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ': '))
    # with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
    #     json.dump(resp.json(), jsonfile)
    # print(parsed)
    # data.update(a_dict)
    result=json.loads(parsed)
    if 'results' in result:
        if len(result['results'])>=1:
            if 'institution' in result['results'][0]:
#                 if 'country' in result['results'][0]['institution']:
#                     inst_country.append(result['results'][0]['institution']['country'])
#                 if 'countryCode' in result['results'][0]['institution']:
#                     inst_cc.append(result['results'][0]['institution']['countryCode'])
#                 if 'id' in result['results'][0]['institution']:

```

```

        inst_id.append(result['results'][0]['institution']['id'])
#     if 'link' in result['results'][0]['institution']:
        inst_link.append(result['results'][0]['institution']['link'])
#     if 'name' in result['results'][0]['institution']:
        inst_name.append(result['results'][0]['institution']['name'])
if 'metrics' in result['results'][0]:
#     if len(result['results'][0]['metrics'])>=1:
        if 'metricType' in result['results'][0]['metrics'][0]:
            metricType.append(result['results'][0]['metrics'][0]['metricType'])
        if 'values' in result['results'][0]['metrics'][0]:
#             print(result['results'][0]['metrics'][0]['values'][1]['threshold'])
            for i in range(0, len(result['results'][0]['metrics'][0]['values'])):
                threshold.append(result['results'][0]['metrics'][0]['values'][i]['threshold'])
            if 'valueByYear' in result['results'][0]['metrics'][0]['values']:
#                 if i ==0:
                    if '2014' in result['results'][0]['metrics'][0]['values']:
                        t1_value2014.append(result['results'][0]['metrics'][0]['values']['2014'])
                    if '2015' in result['results'][0]['metrics'][0]['values']:
                        t1_value2015.append(result['results'][0]['metrics'][0]['values']['2015'])
                    if '2016' in result['results'][0]['metrics'][0]['values']:
                        t1_value2016.append(result['results'][0]['metrics'][0]['values']['2016'])
                    if '2017' in result['results'][0]['metrics'][0]['values']:
                        t1_value2017.append(result['results'][0]['metrics'][0]['values']['2017'])
                    if '2018' in result['results'][0]['metrics'][0]['values']:
                        t1_value2018.append(result['results'][0]['metrics'][0]['values']['2018'])
#                 if i ==1:
                    if '2014' in result['results'][0]['metrics'][0]['values']:
                        t5_value2014.append(result['results'][0]['metrics'][0]['values']['2014'])
                    if '2015' in result['results'][0]['metrics'][0]['values']:
                        t5_value2015.append(result['results'][0]['metrics'][0]['values']['2015'])
                    if '2016' in result['results'][0]['metrics'][0]['values']:
                        t5_value2016.append(result['results'][0]['metrics'][0]['values']['2016'])
                    if '2017' in result['results'][0]['metrics'][0]['values']:
                        t5_value2017.append(result['results'][0]['metrics'][0]['values']['2017'])
                    if '2018' in result['results'][0]['metrics'][0]['values']:
                        t5_value2018.append(result['results'][0]['metrics'][0]['values']['2018'])
#                 if i ==2:
                    if '2014' in result['results'][0]['metrics'][0]['values']:
                        t10_value2014.append(result['results'][0]['metrics'][0]['values']['2014'])
                    if '2015' in result['results'][0]['metrics'][0]['values']:
                        t10_value2015.append(result['results'][0]['metrics'][0]['values']['2015'])
                    if '2016' in result['results'][0]['metrics'][0]['values']:
                        t10_value2016.append(result['results'][0]['metrics'][0]['values']['2016'])
                    if '2017' in result['results'][0]['metrics'][0]['values']:
                        t10_value2017.append(result['results'][0]['metrics'][0]['values']['2017'])
                    if '2018' in result['results'][0]['metrics'][0]['values']:
                        t10_value2018.append(result['results'][0]['metrics'][0]['values']['2018'])

```



```

s8=pd.Series(t1_value2014, name='t1_2014')
s9=pd.Series(t1_value2015, name='t1_2015')
s10=pd.Series(t1_value2016, name='t1_2016')
s11=pd.Series(t1_value2017, name='t1_2017')
s12=pd.Series(t1_value2018, name='t1_2018')
s13=pd.Series(t1_percentage2014, name='t1_percent2014')
s14=pd.Series(t1_percentage2015, name='t1_percent2015')
s15=pd.Series(t1_percentage2016, name='t1_percent2016')
s16=pd.Series(t1_percentage2017, name='t1_percent2017')
s17=pd.Series(t1_percentage2018, name='t1_percent2018')
s18=pd.Series(t5_value2014, name='t5_2014')
s19=pd.Series(t5_value2015, name='t5_2015')
s20=pd.Series(t5_value2016, name='t5_2016')
s21=pd.Series(t5_value2017, name='t5_2017')
s22=pd.Series(t5_value2018, name='t5_2018')
s23=pd.Series(t5_percentage2014, name='t5_percent2014')
s24=pd.Series(t5_percentage2015, name='t5_percent2015')
s25=pd.Series(t5_percentage2016, name='t5_percent2016')
s26=pd.Series(t5_percentage2017, name='t5_percent2017')
s27=pd.Series(t5_percentage2018, name='t5_percent2018')
s28=pd.Series(t10_value2014, name='t10_2014')
s29=pd.Series(t10_value2015, name='t10_2015')
s30=pd.Series(t10_value2016, name='t10_2016')
s31=pd.Series(t10_value2017, name='t10_2017')
s32=pd.Series(t10_value2018, name='t10_2018')
s33=pd.Series(t10_percentage2014, name='t10_percent2014')
s34=pd.Series(t10_percentage2015, name='t10_percent2015')
s35=pd.Series(t10_percentage2016, name='t10_percent2016')
s36=pd.Series(t10_percentage2017, name='t10_percent2017')
s37=pd.Series(t10_percentage2018, name='t10_percent2018')
s38=pd.Series(t25_value2014, name='t25_2014')
s39=pd.Series(t25_value2015, name='t25_2015')
s40=pd.Series(t25_value2016, name='t25_2016')
s41=pd.Series(t25_value2017, name='t25_2017')
s42=pd.Series(t25_value2018, name='t25_2018')
s43=pd.Series(t25_percentage2014, name='t25_percent2014')
s44=pd.Series(t25_percentage2015, name='t25_percent2015')
s45=pd.Series(t25_percentage2016, name='t25_percent2016')
s46=pd.Series(t25_percentage2017, name='t25_percent2017')
s47=pd.Series(t25_percentage2018, name='t25_percent2018')

DF=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,s16, s17,s18,s19,s20,
              s28,s29,s30,s31,s32,s33,s34,s35,s36,s37,s38,s39,s40, s41,s42,s43,s44,s45,s46,s47])

DF.to_csv("THE_UNI_PublicationsInTopJournalPercentiles_ALL_3.csv", index=False) # 0

```

```
#print(threshold)
```

```
In [66]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\PubTopJournalP
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\PubTopJournalPercentile
```

```
In [67]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(2)
```

```
inst_country=[]
inst_cc=[]
inst_id=[]
inst_link=[]
inst_name=[]
metricType=[]
#threshold=[]
t1_value2014=[]
t1_value2015=[]
t1_value2016=[]
t1_value2017=[]
t1_value2018=[]
t1_percentage2014=[]
t1_percentage2015=[]
t1_percentage2016=[]
t1_percentage2017=[]
t1_percentage2018=[]
t5_value2014=[]
t5_value2015=[]
t5_value2016=[]
t5_value2017=[]
t5_value2018=[]
t5_percentage2014=[]
t5_percentage2015=[]
t5_percentage2016=[]
t5_percentage2017=[]
t5_percentage2018=[]
t10_value2014=[]
t10_value2015=[]
t10_value2016=[]
t10_value2017=[]
t10_value2018=[]
```



```

t10_percentage2014=[]
t10_percentage2015=[]
t10_percentage2016=[]
t10_percentage2017=[]
t10_percentage2018=[]
t25_value2014=[]
t25_value2015=[]
t25_value2016=[]
t25_value2017=[]
t25_value2018=[]
t25_percentage2014=[]
t25_percentage2015=[]
t25_percentage2016=[]
t25_percentage2017=[]
t25_percentage2018=[]

for line in data['University id'][1000:]:
    url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Pub
    # print(url.format(line))
    resp = requests.get(url.format(line), headers={'Accept':'application/json',
        'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
    parsed=json.dumps(resp.json(),
        sort_keys=True,
        indent=4, separators=(',', ': '))
    # with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
    #     json.dump(resp.json(), jsonfile)
    # print(parsed)
    # data.update(a_dict)
    result=json.loads(parsed)
    if 'results' in result:
        if len(result['results'])>=1:
            if 'institution' in result['results'][0]:
                # if 'country' in result['results'][0]['institution']:
                #     inst_country.append(result['results'][0]['institution']['country'])
                # if 'countryCode' in result['results'][0]['institution']:
                #     inst_cc.append(result['results'][0]['institution']['countryCode'])
                # if 'id' in result['results'][0]['institution']:
                #     inst_id.append(result['results'][0]['institution']['id'])
                # if 'link' in result['results'][0]['institution']:
                #     inst_link.append(result['results'][0]['institution']['link'])
                # if 'name' in result['results'][0]['institution']:
                #     inst_name.append(result['results'][0]['institution']['name'])
            if 'metrics' in result['results'][0]:
                # if len(result['results'][0]['metrics'])>=1:
                #     if 'metricType' in result['results'][0]['metrics'][0]:
                #         metricType.append(result['results'][0]['metrics'][0]['metricType'])

```

```

if 'values' in result['results'][0]['metrics'][0]:
    print(result['results'][0]['metrics'][0]['values'][1]['threshold'])
    for i in range(0, len(result['results'][0]['metrics'][0]['values']
        threshold.append(result['results'][0]['metrics'][0]['values']
if 'valueByYear' in result['results'][0]['metrics'][0]['values']:
    if i ==0:
        if '2014' in result['results'][0]['metrics'][0]['values'][0]:
            t1_value2014.append(result['results'][0]['metrics'][0]['v
        if '2015' in result['results'][0]['metrics'][0]['values'][0]:
            t1_value2015.append(result['results'][0]['metrics'][0]['v
        if '2016' in result['results'][0]['metrics'][0]['values'][0]:
            t1_value2016.append(result['results'][0]['metrics'][0]['v
        if '2017' in result['results'][0]['metrics'][0]['values'][0]:
            t1_value2017.append(result['results'][0]['metrics'][0]['v
        if '2018' in result['results'][0]['metrics'][0]['values'][0]:
            t1_value2018.append(result['results'][0]['metrics'][0]['v
    if i ==1:
if 'valueByYear' in result['results'][0]['metrics'][0]['values']:
    if '2014' in result['results'][0]['metrics'][0]['values'][1]:
        t5_value2014.append(result['results'][0]['metrics'][0]['v
    if '2015' in result['results'][0]['metrics'][0]['values'][1]:
        t5_value2015.append(result['results'][0]['metrics'][0]['v
    if '2016' in result['results'][0]['metrics'][0]['values'][1]:
        t5_value2016.append(result['results'][0]['metrics'][0]['v
    if '2017' in result['results'][0]['metrics'][0]['values'][1]:
        t5_value2017.append(result['results'][0]['metrics'][0]['v
    if '2018' in result['results'][0]['metrics'][0]['values'][1]:
        t5_value2018.append(result['results'][0]['metrics'][0]['v
    if i ==2:
if 'valueByYear' in result['results'][0]['metrics'][0]['values']:
    if '2014' in result['results'][0]['metrics'][0]['values'][2]:
        t10_value2014.append(result['results'][0]['metrics'][0]['v
    if '2015' in result['results'][0]['metrics'][0]['values'][2]:
        t10_value2015.append(result['results'][0]['metrics'][0]['v
    if '2016' in result['results'][0]['metrics'][0]['values'][2]:
        t10_value2016.append(result['results'][0]['metrics'][0]['v
    if '2017' in result['results'][0]['metrics'][0]['values'][2]:
        t10_value2017.append(result['results'][0]['metrics'][0]['v
    if '2018' in result['results'][0]['metrics'][0]['values'][2]:
        t10_value2018.append(result['results'][0]['metrics'][0]['v
    if i ==3:
if 'valueByYear' in result['results'][0]['metrics'][0]['values']:
    if '2014' in result['results'][0]['metrics'][0]['values'][3]:
        t25_value2014.append(result['results'][0]['metrics'][0]['v
    if '2015' in result['results'][0]['metrics'][0]['values'][3]:
        t25_value2015.append(result['results'][0]['metrics'][0]['v

```

```

        if '2016' in result['results'][0]['metrics'][0]['values'][3]:
            t25_value2016.append(result['results'][0]['metrics'][0]['values'][3])
        if '2017' in result['results'][0]['metrics'][0]['values'][3]:
            t25_value2017.append(result['results'][0]['metrics'][0]['values'][3])
        if '2018' in result['results'][0]['metrics'][0]['values'][3]:
            t25_value2018.append(result['results'][0]['metrics'][0]['values'][3])

    if 'percentageByYear' in result['results'][0]['metrics'][0]['values']:
        #
        if i == 0:
            if '2014' in result['results'][0]['metrics'][0]['values'][0]:
                t1_percentage2014.append(result['results'][0]['metrics'][0]['values'][0])
            if '2015' in result['results'][0]['metrics'][0]['values'][0]:
                t1_percentage2015.append(result['results'][0]['metrics'][0]['values'][0])
            if '2016' in result['results'][0]['metrics'][0]['values'][0]:
                t1_percentage2016.append(result['results'][0]['metrics'][0]['values'][0])
            if '2017' in result['results'][0]['metrics'][0]['values'][0]:
                t1_percentage2017.append(result['results'][0]['metrics'][0]['values'][0])
            if '2018' in result['results'][0]['metrics'][0]['values'][0]:
                t1_percentage2018.append(result['results'][0]['metrics'][0]['values'][0])

        #
        if i == 1:
            if 'percentageByYear' in result['results'][0]['metrics'][0]['values']:
                if '2014' in result['results'][0]['metrics'][0]['values'][1]:
                    t5_percentage2014.append(result['results'][0]['metrics'][0]['values'][1])
                if '2015' in result['results'][0]['metrics'][0]['values'][1]:
                    t5_percentage2015.append(result['results'][0]['metrics'][0]['values'][1])
                if '2016' in result['results'][0]['metrics'][0]['values'][1]:
                    t5_percentage2016.append(result['results'][0]['metrics'][0]['values'][1])
                if '2017' in result['results'][0]['metrics'][0]['values'][1]:
                    t5_percentage2017.append(result['results'][0]['metrics'][0]['values'][1])
                if '2018' in result['results'][0]['metrics'][0]['values'][1]:
                    t5_percentage2018.append(result['results'][0]['metrics'][0]['values'][1])

        #
        if i == 2:
            if 'percentageByYear' in result['results'][0]['metrics'][0]['values']:
                if '2014' in result['results'][0]['metrics'][0]['values'][2]:
                    t10_percentage2014.append(result['results'][0]['metrics'][0]['values'][2])
                if '2015' in result['results'][0]['metrics'][0]['values'][2]:
                    t10_percentage2015.append(result['results'][0]['metrics'][0]['values'][2])
                if '2016' in result['results'][0]['metrics'][0]['values'][2]:
                    t10_percentage2016.append(result['results'][0]['metrics'][0]['values'][2])
                if '2017' in result['results'][0]['metrics'][0]['values'][2]:
                    t10_percentage2017.append(result['results'][0]['metrics'][0]['values'][2])
                if '2018' in result['results'][0]['metrics'][0]['values'][2]:
                    t10_percentage2018.append(result['results'][0]['metrics'][0]['values'][2])

        #
        if i == 3:
            if 'percentageByYear' in result['results'][0]['metrics'][0]['values']:

```



```

s11=pd.Series(t1_value2017, name='t1_2017')
s12=pd.Series(t1_value2018, name='t1_2018')
s13=pd.Series(t1_percentage2014, name='t1_percent2014')
s14=pd.Series(t1_percentage2015, name='t1_percent2015')
s15=pd.Series(t1_percentage2016, name='t1_percent2016')
s16=pd.Series(t1_percentage2017, name='t1_percent2017')
s17=pd.Series(t1_percentage2018, name='t1_percent2018')
s18=pd.Series(t5_value2014, name='t5_2014')
s19=pd.Series(t5_value2015, name='t5_2015')
s20=pd.Series(t5_value2016, name='t5_2016')
s21=pd.Series(t5_value2017, name='t5_2017')
s22=pd.Series(t5_value2018, name='t5_2018')
s23=pd.Series(t5_percentage2014, name='t5_percent2014')
s24=pd.Series(t5_percentage2015, name='t5_percent2015')
s25=pd.Series(t5_percentage2016, name='t5_percent2016')
s26=pd.Series(t5_percentage2017, name='t5_percent2017')
s27=pd.Series(t5_percentage2018, name='t5_percent2018')
s28=pd.Series(t10_value2014, name='t10_2014')
s29=pd.Series(t10_value2015, name='t10_2015')
s30=pd.Series(t10_value2016, name='t10_2016')
s31=pd.Series(t10_value2017, name='t10_2017')
s32=pd.Series(t10_value2018, name='t10_2018')
s33=pd.Series(t10_percentage2014, name='t10_percent2014')
s34=pd.Series(t10_percentage2015, name='t10_percent2015')
s35=pd.Series(t10_percentage2016, name='t10_percent2016')
s36=pd.Series(t10_percentage2017, name='t10_percent2017')
s37=pd.Series(t10_percentage2018, name='t10_percent2018')
s38=pd.Series(t25_value2014, name='t25_2014')
s39=pd.Series(t25_value2015, name='t25_2015')
s40=pd.Series(t25_value2016, name='t25_2016')
s41=pd.Series(t25_value2017, name='t25_2017')
s42=pd.Series(t25_value2018, name='t25_2018')
s43=pd.Series(t25_percentage2014, name='t25_percent2014')
s44=pd.Series(t25_percentage2015, name='t25_percent2015')
s45=pd.Series(t25_percentage2016, name='t25_percent2016')
s46=pd.Series(t25_percentage2017, name='t25_percent2017')
s47=pd.Series(t25_percentage2018, name='t25_percent2018')

DF=pd.concat([s1,s2,s3,s4,s5,s6,s8,s9,s10,s11,s12,s13,s14,s15,s16, s17,s18,s19,s20,s2
s28,s29,s30,s31,s32,s33,s34,s35,s36,s37,s38,s39,s40, s41,s42,s43,s44,s45

DF.to_csv("THE_UNI_PubPercentile_All_18.csv", index=False) # OK

#print(threshold)

```

23 Combine all the subfiles and subset the USA universities

24 CitationCount

```
In [68]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\CitationCount"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\CitationCount
```

```
In [69]: filename='THE_UNI_CitationCount_ALL_{}.csv'
```

```
In [72]: chucks=[]
```

```
for i in range(1, 13):
    chucks.append(pd.read_csv(filename.format(i)))
```

```
cc_data=pd.concat(chucks, ignore_index=True)
```

```
cc_data.head()
```

```
Out[72]:
```

	country	countryCode	institution_id	\
0	United Kingdom	GBR	315091	
1	United States	USA	508092	
2	United States	USA	508021	
3	United Kingdom	GBR	315068	
4	United States	USA	508219	

	link	\
0	{ '@href': 'https://api.elsevier.com/analytics/...	
1	{ '@href': 'https://api.elsevier.com/analytics/...	
2	{ '@href': 'https://api.elsevier.com/analytics/...	
3	{ '@href': 'https://api.elsevier.com/analytics/...	
4	{ '@href': 'https://api.elsevier.com/analytics/...	

	institution_name	metricType	2014	\
0	University of Oxford	CitationCount	362631.0	
1	Jet Propulsion Laboratory, California Institut...	CitationCount	40303.0	
2	California Institute of Technology	CitationCount	131650.0	
3	University of Cambridge	CitationCount	264596.0	
4	Stanford University	CitationCount	411975.0	

	2015	2016	2017	2018
0	320264.0	245401.0	157032.0	82250.0
1	31088.0	33761.0	19335.0	9915.0
2	99103.0	92609.0	60850.0	31739.0
3	241231.0	203383.0	133380.0	70147.0
4	371956.0	290615.0	203160.0	103230.0

```
In [73]: cc_data.tail()
```

```

Out [73]:
country countryCode institution_id \
1263 Saudi Arabia SAU 703099
1264 Turkey TUR 705124
1265 Georgia GEO 204001
1266 United States USA 508335
1267 United States USA 508059

link \
1263 {'@href': 'https://api.elsevier.com/analytics/...
1264 {'@href': 'https://api.elsevier.com/analytics/...
1265 {'@href': 'https://api.elsevier.com/analytics/...
1266 {'@href': 'https://api.elsevier.com/analytics/...
1267 {'@href': 'https://api.elsevier.com/analytics/...

institution_name metricType 2014 \
1263 Imam Abdulrahman Bin Faisal University CitationCount 2412.0
1264 Istanbul Medipol University CitationCount 2063.0
1265 Ivane Javakhishvili Tbilisi State University CitationCount 8208.0
1266 University of Rochester CitationCount 84957.0
1267 Emory University CitationCount 156631.0

2015 2016 2017 2018
1263 2418.0 2644.0 2404.0 3054.0
1264 2211.0 1480.0 1227.0 629.0
1265 8605.0 6183.0 4373.0 4073.0
1266 73383.0 47472.0 35627.0 20159.0
1267 136820.0 108550.0 70004.0 36505.0

```

```

In [74]: cc_data.to_csv('THE_ALLUNI_CC.csv', index=True)

```

25 FWCI

```

In [75]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\FNCI"

```

```

C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\FNCI

```

```

In [76]: filename='THE_UNI_FWCI_{}.csv'

```

```

In [77]: chucks=[]

```

```

for i in range(1, 13):
    chucks.append(pd.read_csv(filename.format(i)))

fwci_data=pd.concat(chucks, ignore_index=True)

fwci_data.head()

```

```

Out [77]:      country countryCode  institution_id  \

0  United Kingdom      GBR          315091
1   United States      USA          508092
2   United States      USA          508021
3  United Kingdom      GBR          315068
4   United States      USA          508219

                                     link  \

0  {'@href': 'https://api.elsevier.com/analytics/...
1  {'@href': 'https://api.elsevier.com/analytics/...
2  {'@href': 'https://api.elsevier.com/analytics/...
3  {'@href': 'https://api.elsevier.com/analytics/...
4  {'@href': 'https://api.elsevier.com/analytics/...

                                     institution_name  \

0                                     University of Oxford
1  Jet Propulsion Laboratory, California Institut...
2                                     California Institute of Technology
3                                     University of Cambridge
4                                     Stanford University

                                     metricType      2014      2015      2016      2017  \

0  FieldWeightedCitationImpact  2.232452  2.178834  2.202485  1.966025
1  FieldWeightedCitationImpact  1.611136  1.462793  1.656759  1.470790
2  FieldWeightedCitationImpact  1.890797  1.740487  1.921985  1.847315
3  FieldWeightedCitationImpact  1.904510  1.990053  2.050378  1.946377
4  FieldWeightedCitationImpact  2.445251  2.476393  2.568147  2.269981

                                     2018

0  1.804821
1  1.346227
2  1.605074
3  1.763683
4  2.244260

```

```

In [78]: fwci_data.tail()

```

```

Out [78]:      country countryCode  institution_id  \

1263  Saudi Arabia      SAU          703099
1264      Turkey      TUR          705124
1265      Georgia      GEO          204001
1266  United States      USA          508335
1267  United States      USA          508059

                                     link  \

1263  {'@href': 'https://api.elsevier.com/analytics/...
1264  {'@href': 'https://api.elsevier.com/analytics/...
1265  {'@href': 'https://api.elsevier.com/analytics/...

```



```
1266 {'@href': 'https://api.elsevier.com/analytics/...
1267 {'@href': 'https://api.elsevier.com/analytics/...
```

```

                                institution_name \
1263      Imam Abdulrahman Bin Faisal University
1264      Istanbul Medipol University
1265  Ivane Javakhishvili Tbilisi State University
1266      University of Rochester
1267      Emory University
```

```

                                metricType      2014      2015      2016      2017 \
1263  FieldWeightedCitationImpact  0.698082  0.649441  0.772378  0.787842
1264  FieldWeightedCitationImpact  0.661692  0.577872  0.552237  0.635915
1265  FieldWeightedCitationImpact  1.556471  1.550585  1.177011  1.246151
1266  FieldWeightedCitationImpact  1.827581  2.117681  1.646356  1.700714
1267  FieldWeightedCitationImpact  1.999690  2.186228  2.209265  1.948478
```

```

                                2018
1263  0.880861
1264  0.666751
1265  1.725649
1266  1.717541
1267  1.967104
```

```
In [79]: fwc_data.to_csv("THE_ALLUNI_FWCI.csv", index=False)
```

26 PercPublsCited

```
In [107]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PercPublsCited"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PercPublsCited
```

```
In [81]: filename='THE_UNI_CitedPublications_{}.csv'
```

```
In [82]: chucks=[]
```

```

for i in range(1, 13):
    chucks.append(pd.read_csv(filename.format(i)))

cp_data=pd.concat(chucks, ignore_index=True)

cp_data.head()
```

```
Out [82]:
   country countryCode institution_id \
0  United Kingdom      GBR          315091
1  United States      USA          508092
2  United States      USA          508021
```

3	United Kingdom	GBR	315068
4	United States	USA	508219

	link \
0	{'@href': 'https://api.elsevier.com/analytics/...
1	{'@href': 'https://api.elsevier.com/analytics/...
2	{'@href': 'https://api.elsevier.com/analytics/...
3	{'@href': 'https://api.elsevier.com/analytics/...
4	{'@href': 'https://api.elsevier.com/analytics/...

	institution_name	metricType \
0	University of Oxford	CitedPublications
1	Jet Propulsion Laboratory, California Institut...	CitedPublications
2	California Institute of Technology	CitedPublications
3	University of Cambridge	CitedPublications
4	Stanford University	CitedPublications

	2014	2015	2016	2017	2018	percent2014	percent2015 \
0	10893.0	11679.0	11798.0	11474.0	10570.0	86.555420	85.53537
1	1514.0	1451.0	1722.0	1588.0	1406.0	79.142710	80.61111
2	3879.0	3770.0	3914.0	3779.0	3487.0	85.856575	86.72648
3	9116.0	9238.0	9558.0	9125.0	8418.0	88.060280	86.17537
4	11156.0	11846.0	11699.0	11642.0	10731.0	87.149445	86.74575

	percent2016	percent2017	percent2018
0	84.115210	78.56213	70.69761
1	78.954605	77.38792	63.90909
2	85.965300	82.74578	71.39640
3	84.330330	79.49991	71.42372
4	85.619150	81.79583	72.85627

In [83]: cp_data.tail()

Out[83]:

	country	countryCode	institution_id \
1263	Saudi Arabia	SAU	703099
1264	Turkey	TUR	705124
1265	Georgia	GEO	204001
1266	United States	USA	508335
1267	United States	USA	508059

	link \
1263	{'@href': 'https://api.elsevier.com/analytics/...
1264	{'@href': 'https://api.elsevier.com/analytics/...
1265	{'@href': 'https://api.elsevier.com/analytics/...
1266	{'@href': 'https://api.elsevier.com/analytics/...
1267	{'@href': 'https://api.elsevier.com/analytics/...

institution_name	metricType	2014 \
------------------	------------	--------

1263	Imam Abdulrahman Bin Faisal University	CitedPublications	296.0
1264	Istanbul Medipol University	CitedPublications	203.0
1265	Ivane Javakhishvili Tbilisi State University	CitedPublications	295.0
1266	University of Rochester	CitedPublications	3011.0
1267	Emory University	CitedPublications	5206.0

	2015	2016	2017	2018	percent2014	percent2015	percent2016	\
1263	305.0	364.0	384.0	560.0	79.78437	75.495056	79.302826	
1264	293.0	295.0	262.0	207.0	78.07692	72.524750	70.743410	
1265	347.0	383.0	342.0	366.0	65.84821	68.441810	66.034485	
1266	2939.0	2919.0	2795.0	2649.0	83.59245	83.022600	83.044090	
1267	5230.0	5187.0	5126.0	4687.0	88.08799	87.810610	84.977066	

	percent2017	percent2018
1263	66.435990	59.447987
1264	62.679430	47.260273
1265	65.391970	60.098522
1266	76.912490	68.948460
1267	80.992256	72.252200

```
In [84]: cp_data.to_csv("THEUNI_CITEDPUBLS.csv",index=False)
```

27 PubTopJournalPercentile

```
In [85]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\PubTopJournalP
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\PubTopJournalPercentile
```

```
In [86]: filename='THE_UNI_PubPercentile_All_{}.csv'
```

```
In [87]: chucks=[]
```

```
for i in range(1, 19):
    chucks.append(pd.read_csv(filename.format(i)))

pp_data=pd.concat(chucks, ignore_index=True)

pp_data.head()
```

```
Out [87]:
```

	country	countryCode	institution_id	\
0	United Kingdom	GBR	315091	
1	United States	USA	508092	
2	United Kingdom	GBR	315091	
3	United States	USA	508092	
4	United States	USA	508021	

link \

```

0 {'@href': 'https://api.elsevier.com/analytics/...
1 {'@href': 'https://api.elsevier.com/analytics/...
2 {'@href': 'https://api.elsevier.com/analytics/...
3 {'@href': 'https://api.elsevier.com/analytics/...
4 {'@href': 'https://api.elsevier.com/analytics/...

                                institution_name \
0                                University of Oxford
1 Jet Propulsion Laboratory, California Institut...
2                                University of Oxford
3 Jet Propulsion Laboratory, California Institut...
4                                California Institute of Technology

                                metricType  t1_2014  t1_2015  t1_2016  t1_2017 \
0 PublicationsInTopJournalPercentiles    686.0    846.0    887.0    771.0
1 PublicationsInTopJournalPercentiles     74.0     82.0     83.0     71.0
2 PublicationsInTopJournalPercentiles    686.0    846.0    887.0    771.0
3 PublicationsInTopJournalPercentiles     74.0     82.0     83.0     71.0
4 PublicationsInTopJournalPercentiles    305.0    245.0    259.0    253.0

    ...  t25_2014  t25_2015  t25_2016  t25_2017  t25_2018  t25_percent2014 \
0 ...    8098.0    8796.0    9089.0    9372.0    10096.0         73.77915
1 ...    1012.0    1056.0    1215.0    1195.0    1233.0         67.55675
2 ...    8098.0    8796.0    9089.0    9372.0    10096.0         73.77915
3 ...    1012.0    1056.0    1215.0    1195.0    1233.0         67.55675
4 ...    2948.0    2945.0    3094.0    3108.0    3251.0         76.41265

    t25_percent2015  t25_percent2016  t25_percent2017  t25_percent2018
0         75.321110         73.55345         74.210150         74.923935
1         73.333336         67.42509         66.536750         64.185320
2         75.321110         73.55345         74.210150         74.923935
3         73.333336         67.42509         66.536750         64.185320
4         79.102875         77.93451         76.835594         74.752820

```

[5 rows x 46 columns]

In [88]: pp_data.tail()

```

Out[88]:
    country countryCode  institution_id \
1265 Saudi Arabia      SAU          703099
1266 Turkey            TUR          705124
1267 Georgia           GEO          204001
1268 United States     USA          508335
1269 United States     USA          508059

                                link \
1265 {'@href': 'https://api.elsevier.com/analytics/...
1266 {'@href': 'https://api.elsevier.com/analytics/...

```

```

1267 {'@href': 'https://api.elsevier.com/analytics/...
1268 {'@href': 'https://api.elsevier.com/analytics/...
1269 {'@href': 'https://api.elsevier.com/analytics/...

```

```

                                institution_name \
1265      Imam Abdulrahman Bin Faisal University
1266      Istanbul Medipol University
1267  Ivane Javakhishvili Tbilisi State University
1268      University of Rochester
1269      Emory University

```

```

                                metricType  t1_2014  t1_2015  t1_2016  t1_2017 \
1265  PublicationsInTopJournalPercentiles      1.0      2.0      5.0      6.0
1266  PublicationsInTopJournalPercentiles      3.0      2.0      2.0      5.0
1267  PublicationsInTopJournalPercentiles      1.0      1.0      2.0      2.0
1268  PublicationsInTopJournalPercentiles     162.0     164.0     143.0     138.0
1269  PublicationsInTopJournalPercentiles     277.0     348.0     318.0     348.0

```

```

...  t25_2014  t25_2015  t25_2016  t25_2017  t25_2018  t25_percent2014 \
1265 ...      87.0      103.0      140.0      198.0      353.0      26.605505
1266 ...      67.0      106.0      77.0      99.0      90.0      27.800830
1267 ...     205.0     241.0     278.0     266.0     335.0     50.368546
1268 ...    2163.0    2125.0    2131.0    2130.0    2309.0     70.364340
1269 ...    3967.0    3923.0    3854.0    3975.0    4143.0     73.708660

```

```

                                t25_percent2015  t25_percent2016  t25_percent2017  t25_percent2018
1265      29.428572      34.567900      38.521400      40.762123
1266      29.041096      20.810812      27.049181      25.210085
1267      53.200882      52.751423      55.416668      58.566433
1268      69.331154      67.436710      66.645805      66.522610
1269      72.246780      68.907560      67.407160      68.186310

```

[5 rows x 46 columns]

```
In [89]: pp_data.to_csv("THE_ALLUNI_PP.csv", index=False)
```

28 ScholarlyOutput

```
In [106]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data" # needs to u
```

```
[WinError 2] The system cannot find the file specified: 'C:\\Users\\jchen148\\THE Rankings\\Re
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PubTopJournalPercentile
```

```
In [94]: data.tail()
```

```

Out[94]:
                                University Name  University id \
1003      Imam Abdulrahman Bin Faisal University      703099

```

1004	Istanbul Medipol University	705124
1005	Ivane Javakhishvili Tbilisi State University	204001
1006	University of Rochester	508335
1007	Emory University	508059

	Country	Country Code
1003	Saudi Arabia	SAU
1004	Turkey	TUR
1005	Georgia	GEO
1006	United States	USA
1007	United States	USA

```
In [100]: url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=ScholarlyOutput'
```

```
resp = requests.get(url.format(line), headers={'Accept': 'application/json',
                                                'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ': '))
# with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
#     json.dump(resp.json(), jsonfile)
# print(parsed)
# data.update(a_dict)
result=json.loads(parsed)
result['results']
```

```
Out[100]: [{'institution': {'country': 'United States',
                           'countryCode': 'USA',
                           'id': 508059,
                           'link': {'@href': 'https://api.elsevier.com/analytics/scival/institution/508059?api-key=d3794058e2b24417b5dfd0ef8990e2dc',
                                    '@ref': 'self',
                                    '@type': 'application/json'},
                           'name': 'Emory University',
                           'uri': 'Institution/508059'},
            'metrics': [{'metricType': 'ScholarlyOutput',
                        'valueByYear': {'2014': 5910,
                                         '2015': 5956,
                                         '2016': 6104,
                                         '2017': 6329,
                                         '2018': 6487}}]}]
```

```
In [108]: country=[]
countryCode=[]
institution_id=[]
link=[]
institution_name=[]
metricType=[]
value2014=[]
```

```

value2015=[]
value2016=[]
value2017=[]
value2018=[]

url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=ScholarlyOutput'

resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                              'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ': '))
# with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
#     json.dump(resp.json(), jsonfile)
# print(parsed)
# data.update(a_dict)
result=json.loads(parsed)
result['results']
country.append(result['results'][0]['institution']['country'])
countryCode.append(result['results'][0]['institution']['countryCode'])
institution_id.append(result['results'][0]['institution']['id'])
link.append(result['results'][0]['institution']['link'])
institution_name.append(result['results'][0]['institution']['name'])
metricType.append(result['results'][0]['metrics'][0]['metricType'])
value2014.append(result['results'][0]['metrics'][0]['valueByYear']['2014'])
value2015.append(result['results'][0]['metrics'][0]['valueByYear']['2015'])
value2016.append(result['results'][0]['metrics'][0]['valueByYear']['2016'])
value2017.append(result['results'][0]['metrics'][0]['valueByYear']['2017'])
value2018.append(result['results'][0]['metrics'][0]['valueByYear']['2018'])

DF=pd.DataFrame({'country':country, 'countryCode': countryCode, 'institution_id': institution_id,
                 'institution_name':institution_name, 'metricType':metricType,
                 '2014': value2014, '2015': value2015, '2016': value2016, '2017':value2017, '2018':value2018})

DF.to_csv("THE_UNI_SCHOLAROUTPUT_ALL_15.csv", index=False)

In [109]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\ScholarlyOutput"

C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\ScholarlyOutput

In [110]: filename='THE_UNI_SCHOLAROUTPUT_ALL_{}.csv'

In [111]: chucks=[]

for i in range(1, 16):
    chucks.append(pd.read_csv(filename.format(i)))

so_data=pd.concat(chucks, ignore_index=True)

```

```
so_data.head()
```

```
Out[111]:
```

	country	countryCode	institution_id	\
0	United Kingdom	GBR	315091	
1	United States	USA	508092	
2	United States	USA	508021	
3	United Kingdom	GBR	315068	
4	United States	USA	508219	

	link	\
0	{'@href': 'https://api.elsevier.com/analytics/...	
1	{'@href': 'https://api.elsevier.com/analytics/...	
2	{'@href': 'https://api.elsevier.com/analytics/...	
3	{'@href': 'https://api.elsevier.com/analytics/...	
4	{'@href': 'https://api.elsevier.com/analytics/...	

	institution_name	metricType	2014	\
0	University of Oxford	ScholarlyOutput	12585	
1	Jet Propulsion Laboratory, California Institut...	ScholarlyOutput	1913	
2	California Institute of Technology	ScholarlyOutput	4518	
3	University of Cambridge	ScholarlyOutput	10352	
4	Stanford University	ScholarlyOutput	12801	

	2015	2016	2017	2018
0	13654	14026	14605	14951
1	1800	2181	2052	2200
2	4347	4553	4567	4884
3	10720	11334	11478	11786
4	13656	13664	14233	14729

```
In [112]: so_data.to_csv("THE_ALLUNI_SO.csv", index=False)
```

29 USA University Publication Output

30 Total

```
In [261]: so_data.head()
```

```
Out[261]:
```

	country	countryCode	institution_id	\
0	United Kingdom	GBR	315091	
1	United States	USA	508092	
2	United States	USA	508021	
3	United Kingdom	GBR	315068	
4	United States	USA	508219	

	link	\
0	{'@href': 'https://api.elsevier.com/analytics/...	


```

1 {'@href': 'https://api.elsevier.com/analytics/...
2 {'@href': 'https://api.elsevier.com/analytics/...
3 {'@href': 'https://api.elsevier.com/analytics/...
4 {'@href': 'https://api.elsevier.com/analytics/...

```

	institution_name	metricType	2014	\
0	University of Oxford	ScholarlyOutput	12585	
1	Jet Propulsion Laboratory, California Institut...	ScholarlyOutput	1913	
2	California Institute of Technology	ScholarlyOutput	4518	
3	University of Cambridge	ScholarlyOutput	10352	
4	Stanford University	ScholarlyOutput	12801	

	2015	2016	2017	2018
0	13654	14026	14605	14951
1	1800	2181	2052	2200
2	4347	4553	4567	4884
3	10720	11334	11478	11786
4	13656	13664	14233	14729

```

In [113]: so_data[so_data.countryCode=='USA'].head()
          so_data_USA=so_data[so_data.countryCode=='USA']

```

```

In [114]: import seaborn as sns
          import matplotlib.pyplot as plt
          %matplotlib inline

```

```

In [115]: so_data_USA=so_data_USA.iloc[:,-7:]

```

```

In [116]: so_data_USA.head()

```

```

Out[116]:
          institution_name      metricType  2014  \
1  Jet Propulsion Laboratory, California Institut...  ScholarlyOutput  1913
2          California Institute of Technology  ScholarlyOutput  4518
4          Stanford University  ScholarlyOutput  12801
5  Massachusetts Institute of Technology  ScholarlyOutput  9645
6          Princeton University  ScholarlyOutput  4335

```

	2015	2016	2017	2018
1	1800	2181	2052	2200
2	4347	4553	4567	4884
4	13656	13664	14233	14729
5	9957	10023	10191	10458
6	4629	4544	4635	4891

```

In [117]: del so_data_USA['metricType']

```

```

In [118]: so_data_USA.head()

```

```

Out[118]:
          institution_name  2014  2015  2016  \
1  Jet Propulsion Laboratory, California Institut...  1913  1800  2181

```

2	California Institute of Technology	4518	4347	4553
4	Stanford University	12801	13656	13664
5	Massachusetts Institute of Technology	9645	9957	10023
6	Princeton University	4335	4629	4544

	2017	2018
1	2052	2200
2	4567	4884
4	14233	14729
5	10191	10458
6	4635	4891

```
In [119]: so_data_USA=so_data_USA.set_index('institution_name')
```

```
In [120]: so_data_USA.agg('sum')
```

```
Out[120]: 2014    412305
          2015    424108
          2016    433662
          2017    450239
          2018    467010
          dtype: int64
```

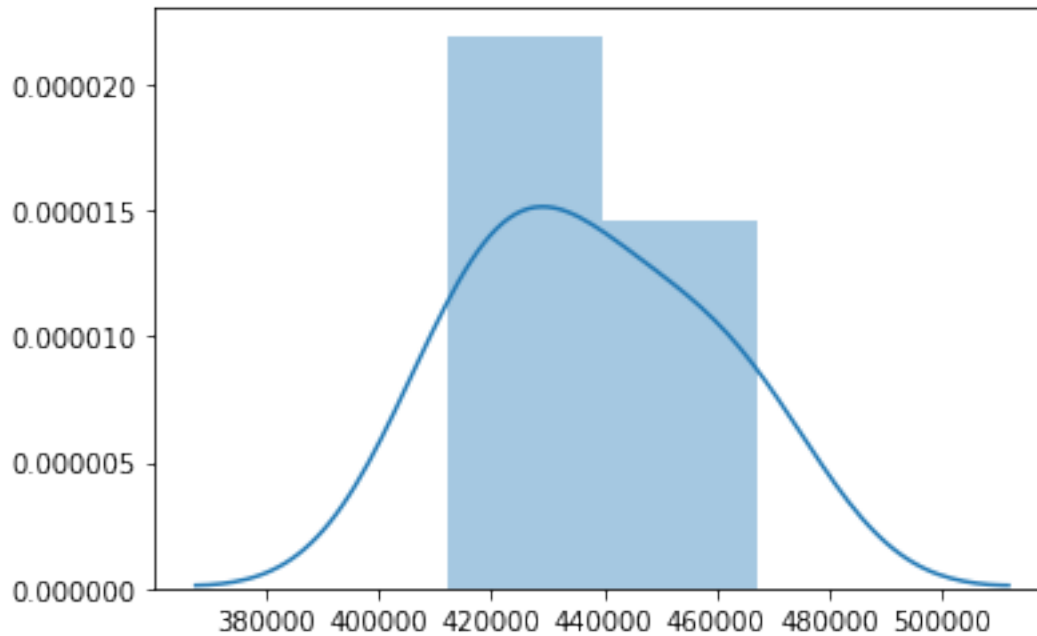
31 THE 163 USA Universities ranked before top 300,

32 The total ScholarlyOutput presents a Bell-shaped

33 Distribution.

```
In [121]: sns.distplot(so_data_USA.agg('sum'))
```

```
Out[121]: <matplotlib.axes._subplots.AxesSubplot at 0x171c8234a90>
```



```
In [122]: len(so_data_USA) # 164 USA universities
```

```
Out[122]: 164
```

```
In [124]: so_data_USA.agg('sum')
```

```
Out[124]: 2014    412305
          2015    424108
          2016    433662
          2017    450239
          2018    467010
          dtype: int64
```

```
In [125]: so_data_USA=so_data_USA.reset_index()
```

```
In [412]: so_data_USA.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 163 entries, 0 to 162
Data columns (total 7 columns):
institution_name    163 non-null object
2014                163 non-null int64
2015                163 non-null int64
2016                163 non-null int64
2017                163 non-null int64
2018                163 non-null int64
Total              0 non-null float64
```

```
dtypes: float64(1), int64(5), object(1)
memory usage: 9.0+ KB
```

```
In [391]: sep_sum=lambda x: x.agg('sum')
```

```
In [126]: so_data_USA['Total']=so_data_USA.sum(axis=1)
```

```
In [127]: so_data_USA['Total']=so_data_USA.Total.astype(int)
so_data_USA.head()
```

```
Out[127]:
```

	institution_name	2014	2015	2016	\
0	Jet Propulsion Laboratory, California Institut...	1913	1800	2181	
1	California Institute of Technology	4518	4347	4553	
2	Stanford University	12801	13656	13664	
3	Massachusetts Institute of Technology	9645	9957	10023	
4	Princeton University	4335	4629	4544	

	2017	2018	Total
0	2052	2200	10146
1	4567	4884	22869
2	14233	14729	69083
3	10191	10458	50274
4	4635	4891	23034

```
In [128]: URpp=so_data_USA[so_data_USA.institution_name=='University of Rochester']
```

```
In [129]: URpp=URpp.reset_index()
```

```
In [130]: URpp['Total']=URpp.sum(axis=1)
```

```
In [131]: URpp
```

```
Out[131]:
```

	index	institution_name	2014	2015	2016	2017	2018	Total
0	162	University of Rochester	3602	3540	3515	3633	3842	36426

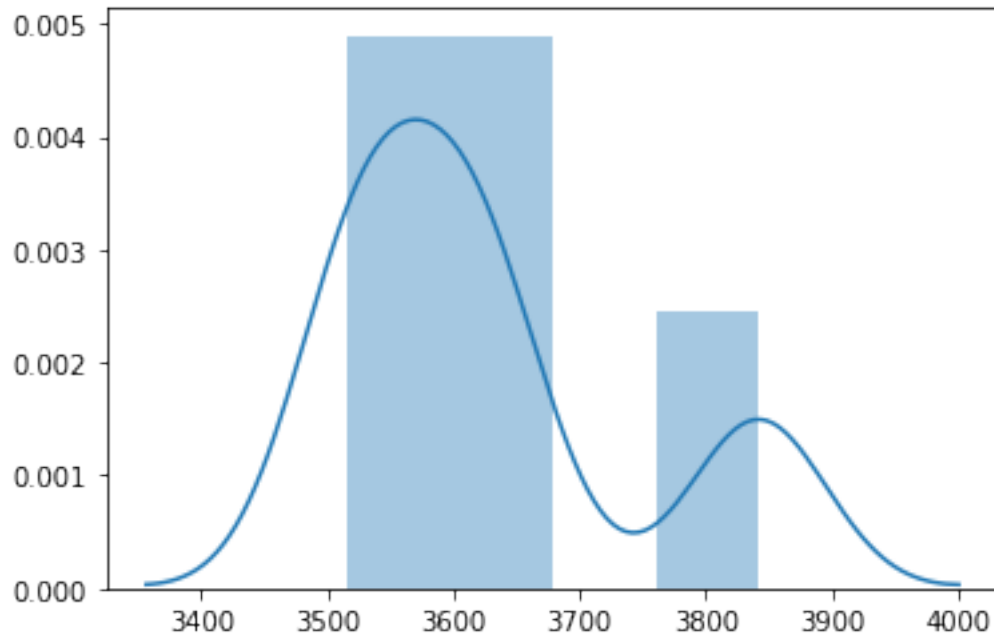
34 UofR's ScholarlyOutput from 2014 to 2018.

35 It seems more like a Bi-modal distribution.

```
In [132]: # UR Publs Distribution
inputdata=URpp[['2014', '2015', '2016', '2017', '2018']]
sns.distplot(inputdata)

# seems a bi-modal distribution but the overall trend is downward
```

```
Out[132]: <matplotlib.axes._subplots.AxesSubplot at 0x171c86b2518>
```



36 Top 1% and top 10% highly cited publications

In [135]: `pp_data.tail()`

```
Out[135]:
```

	country	countryCode	institution_id	\	link	\	institution_name	\
1265	Saudi Arabia	SAU	703099		{ '@href': 'https://api.elsevier.com/analytics/...		Imam Abdulrahman Bin Faisal University	
1266	Turkey	TUR	705124		{ '@href': 'https://api.elsevier.com/analytics/...		Istanbul Medipol University	
1267	Georgia	GEO	204001		{ '@href': 'https://api.elsevier.com/analytics/...		Ivane Javakhishvili Tbilisi State University	
1268	United States	USA	508335		{ '@href': 'https://api.elsevier.com/analytics/...		University of Rochester	
1269	United States	USA	508059		{ '@href': 'https://api.elsevier.com/analytics/...		Emory University	

		metricType	t1_2014	t1_2015	t1_2016	t1_2017	\
1265	PublicationsInTopJournalPercentiles		1.0	2.0	5.0	6.0	
1266	PublicationsInTopJournalPercentiles		3.0	2.0	2.0	5.0	
1267	PublicationsInTopJournalPercentiles		1.0	1.0	2.0	2.0	
1268	PublicationsInTopJournalPercentiles		162.0	164.0	143.0	138.0	
1269	PublicationsInTopJournalPercentiles		277.0	348.0	318.0	348.0	

	...	t25_2014	t25_2015	t25_2016	t25_2017	t25_2018	t25_percent2014	\
1265	...	87.0	103.0	140.0	198.0	353.0	26.605505	
1266	...	67.0	106.0	77.0	99.0	90.0	27.800830	
1267	...	205.0	241.0	278.0	266.0	335.0	50.368546	
1268	...	2163.0	2125.0	2131.0	2130.0	2309.0	70.364340	
1269	...	3967.0	3923.0	3854.0	3975.0	4143.0	73.708660	

		t25_percent2015	t25_percent2016	t25_percent2017	t25_percent2018
1265		29.428572	34.567900	38.521400	40.762123
1266		29.041096	20.810812	27.049181	25.210085
1267		53.200882	52.751423	55.416668	58.566433
1268		69.331154	67.436710	66.645805	66.522610
1269		72.246780	68.907560	67.407160	68.186310

[5 rows x 46 columns]

In [436]: pp_data.head()

Out[436]:

	country	countryCode	institution_id	\
0	United Kingdom	GBR	315091	
1	United States	USA	508092	
2	United Kingdom	GBR	315091	
3	United States	USA	508092	
4	United States	USA	508021	

	link	\
0	{ '@href': 'https://api.elsevier.com/analytics/...' }	
1	{ '@href': 'https://api.elsevier.com/analytics/...' }	
2	{ '@href': 'https://api.elsevier.com/analytics/...' }	
3	{ '@href': 'https://api.elsevier.com/analytics/...' }	
4	{ '@href': 'https://api.elsevier.com/analytics/...' }	

	institution_name	\
0	University of Oxford	
1	Jet Propulsion Laboratory, California Institut...	
2	University of Oxford	
3	Jet Propulsion Laboratory, California Institut...	
4	California Institute of Technology	

	metricType	t1_2014	t1_2015	t1_2016	t1_2017	\
0	PublicationsInTopJournalPercentiles	686.0	846.0	887.0	771.0	

1	PublicationsInTopJournalPercentiles	74.0	82.0	83.0	71.0
2	PublicationsInTopJournalPercentiles	686.0	846.0	887.0	771.0
3	PublicationsInTopJournalPercentiles	74.0	82.0	83.0	71.0
4	PublicationsInTopJournalPercentiles	305.0	245.0	259.0	253.0

	...	t25_2014	t25_2015	t25_2016	t25_2017	t25_2018	t25_percent2014	\
0	...	8098.0	8796.0	9089.0	9372.0	10096.0	73.77915	
1	...	1012.0	1056.0	1215.0	1195.0	1233.0	67.55675	
2	...	8098.0	8796.0	9089.0	9372.0	10096.0	73.77915	
3	...	1012.0	1056.0	1215.0	1195.0	1233.0	67.55675	
4	...	2948.0	2945.0	3094.0	3108.0	3251.0	76.41265	

	t25_percent2015	t25_percent2016	t25_percent2017	t25_percent2018
0	75.321110	73.55345	74.210150	74.923935
1	73.333336	67.42509	66.536750	64.185320
2	75.321110	73.55345	74.210150	74.923935
3	73.333336	67.42509	66.536750	64.185320
4	79.102875	77.93451	76.835594	74.752820

[5 rows x 46 columns]

In [136]: USA_pp=pp_data[pp_data.countryCode=='USA']

In [137]: len(USA_pp)

Out[137]: 166

In [138]: # we want t1 and t10 values

USA_pp.head()

Out[138]:

	country	countryCode	institution_id	\
1	United States	USA	508092	
3	United States	USA	508092	
4	United States	USA	508021	
6	United States	USA	508219	
7	United States	USA	508111	

	link	\
1	{'@href': 'https://api.elsevier.com/analytics/...	
3	{'@href': 'https://api.elsevier.com/analytics/...	
4	{'@href': 'https://api.elsevier.com/analytics/...	
6	{'@href': 'https://api.elsevier.com/analytics/...	
7	{'@href': 'https://api.elsevier.com/analytics/...	

	institution_name	\
1	Jet Propulsion Laboratory, California Institut...	
3	Jet Propulsion Laboratory, California Institut...	
4	California Institute of Technology	

```

6                                Stanford University
7                                Massachusetts Institute of Technology

```

		metricType	t1_2014	t1_2015	t1_2016	t1_2017	\
1		PublicationsInTopJournalPercentiles	74.0	82.0	83.0	71.0	
3		PublicationsInTopJournalPercentiles	74.0	82.0	83.0	71.0	
4		PublicationsInTopJournalPercentiles	305.0	245.0	259.0	253.0	
6		PublicationsInTopJournalPercentiles	954.0	1016.0	1073.0	1027.0	
7		PublicationsInTopJournalPercentiles	820.0	818.0	918.0	822.0	
	...	t25_2014	t25_2015	t25_2016	t25_2017	t25_2018	t25_percent2014 \
1	...	1012.0	1056.0	1215.0	1195.0	1233.0	67.55675
3	...	1012.0	1056.0	1215.0	1195.0	1233.0	67.55675
4	...	2948.0	2945.0	3094.0	3108.0	3251.0	76.41265
6	...	8211.0	8963.0	8819.0	9276.0	9716.0	75.16478
7	...	5930.0	6190.0	6450.0	6475.0	6957.0	75.70535
		t25_percent2015	t25_percent2016	t25_percent2017	t25_percent2018		
1		73.333336	67.425090	66.536750	64.18532		
3		73.333336	67.425090	66.536750	64.18532		
4		79.102875	77.934510	76.835594	74.75282		
6		76.541420	74.535164	75.279980	74.75571		
7		75.839260	76.813150	77.120056	78.15975		

```
[5 rows x 46 columns]
```

```
In [442]: USA_pp.columns
```

```

Out[442]: Index(['country', 'countryCode', 'institution_id', 'link', 'institution_name',
                'metricType', 't1_2014', 't1_2015', 't1_2016', 't1_2017', 't1_2018',
                't1_percent2014', 't1_percent2015', 't1_percent2016', 't1_percent2017',
                't1_percent2018', 't5_2014', 't5_2015', 't5_2016', 't5_2017', 't5_2018',
                't5_percent2014', 't5_percent2015', 't5_percent2016', 't5_percent2017',
                't5_percent2018', 't10_2014', 't10_2015', 't10_2016', 't10_2017',
                't10_2018', 't10_percent2014', 't10_percent2015', 't10_percent2016',
                't10_percent2017', 't10_percent2018', 't25_2014', 't25_2015',
                't25_2016', 't25_2017', 't25_2018', 't25_percent2014',
                't25_percent2015', 't25_percent2016', 't25_percent2017',
                't25_percent2018'],
                dtype='object')

```

```
In [139]: USA_pp=USA_pp.loc[:,['institution_name','t1_2014','t1_2015','t1_2016','t1_2017','t1_2018']]
```

```
In [140]: USA_pp=USA_pp.drop_duplicates()
```

```
In [141]: USA_pp=USA_pp.reset_index()
```

```
In [142]: USA_pp=USA_pp.iloc[:,1:]
```

```
In [143]: USA_pp.head()
```



```
Out[143]:
```

	institution_name	t1_2014	t1_2015	\
0	Jet Propulsion Laboratory, California Institut...	74.0	82.0	
1	California Institute of Technology	305.0	245.0	
2	Stanford University	954.0	1016.0	
3	Massachusetts Institute of Technology	820.0	818.0	
4	Princeton University	271.0	269.0	

	t1_2016	t1_2017	t1_2018	t10_2014	t10_2015	t10_2016	t10_2017	t10_2018
0	83.0	71.0	123.0	455.0	483.0	694.0	687.0	679.0
1	259.0	253.0	293.0	1709.0	1579.0	2051.0	1906.0	1967.0
2	1073.0	1027.0	1025.0	5558.0	6068.0	5974.0	6233.0	6307.0
3	918.0	822.0	869.0	4401.0	4418.0	4689.0	4613.0	4781.0
4	227.0	241.0	253.0	1810.0	1740.0	1916.0	1871.0	1990.0

```
In [144]: USA_pp.tail()
```

```
Out[144]:
```

	institution_name	t1_2014	t1_2015	t1_2016	\
149	University of Texas at El Paso	15.0	11.0	13.0	
150	Susquehanna University	3.0	2.0	0.0	
151	California State University Long Beach	14.0	4.0	14.0	
152	University of Rochester	162.0	164.0	143.0	
153	Emory University	277.0	348.0	318.0	

	t1_2017	t1_2018	t10_2014	t10_2015	t10_2016	t10_2017	t10_2018
149	20.0	30.0	164.0	238.0	254.0	257.0	246.0
150	1.0	1.0	14.0	10.0	9.0	14.0	10.0
151	13.0	12.0	91.0	93.0	110.0	113.0	119.0
152	138.0	133.0	1404.0	1308.0	1310.0	1309.0	1318.0
153	348.0	274.0	2411.0	2494.0	2348.0	2464.0	2390.0

```
In [145]: USA_pp['2014_Total']=USA_pp.loc[:,['t1_2014','t10_2014']].sum(axis=1)
```

```
In [146]: USA_pp.head()
```

```
Out[146]:
```

	institution_name	t1_2014	t1_2015	\
0	Jet Propulsion Laboratory, California Institut...	74.0	82.0	
1	California Institute of Technology	305.0	245.0	
2	Stanford University	954.0	1016.0	
3	Massachusetts Institute of Technology	820.0	818.0	
4	Princeton University	271.0	269.0	

	t1_2016	t1_2017	t1_2018	t10_2014	t10_2015	t10_2016	t10_2017	\
0	83.0	71.0	123.0	455.0	483.0	694.0	687.0	
1	259.0	253.0	293.0	1709.0	1579.0	2051.0	1906.0	
2	1073.0	1027.0	1025.0	5558.0	6068.0	5974.0	6233.0	
3	918.0	822.0	869.0	4401.0	4418.0	4689.0	4613.0	
4	227.0	241.0	253.0	1810.0	1740.0	1916.0	1871.0	

	t10_2018	2014_Total
--	----------	------------

0	679.0	529.0
1	1967.0	2014.0
2	6307.0	6512.0
3	4781.0	5221.0
4	1990.0	2081.0

```
In [147]: USA_pp['2015_Total']=USA_pp.loc[:,['t1_2015','t10_2015']].sum(axis=1)
USA_pp['2016_Total']=USA_pp.loc[:,['t1_2016','t10_2016']].sum(axis=1)
USA_pp['2017_Total']=USA_pp.loc[:,['t1_2017','t10_2017']].sum(axis=1)
USA_pp['2018_Total']=USA_pp.loc[:,['t1_2018','t10_2018']].sum(axis=1)
```

```
In [148]: USA_pp.head()
```

```
Out[148]:
```

	institution_name	t1_2014	t1_2015	\
0	Jet Propulsion Laboratory, California Institut...	74.0	82.0	
1	California Institute of Technology	305.0	245.0	
2	Stanford University	954.0	1016.0	
3	Massachusetts Institute of Technology	820.0	818.0	
4	Princeton University	271.0	269.0	

	t1_2016	t1_2017	t1_2018	t10_2014	t10_2015	t10_2016	t10_2017	\
0	83.0	71.0	123.0	455.0	483.0	694.0	687.0	
1	259.0	253.0	293.0	1709.0	1579.0	2051.0	1906.0	
2	1073.0	1027.0	1025.0	5558.0	6068.0	5974.0	6233.0	
3	918.0	822.0	869.0	4401.0	4418.0	4689.0	4613.0	
4	227.0	241.0	253.0	1810.0	1740.0	1916.0	1871.0	

	t10_2018	2014_Total	2015_Total	2016_Total	2017_Total	2018_Total
0	679.0	529.0	565.0	777.0	758.0	802.0
1	1967.0	2014.0	1824.0	2310.0	2159.0	2260.0
2	6307.0	6512.0	7084.0	7047.0	7260.0	7332.0
3	4781.0	5221.0	5236.0	5607.0	5435.0	5650.0
4	1990.0	2081.0	2009.0	2143.0	2112.0	2243.0

```
In [149]: UR_percentile=USA_pp[USA_pp.institution_name=='University of Rochester']
```

```
In [150]: UR_percentile=UR_percentile.set_index('institution_name')
```

```
In [151]: UR_percentile
```

```
Out[151]:
```

	t1_2014	t1_2015	t1_2016	t1_2017	t1_2018	\
institution_name						
University of Rochester	162.0	164.0	143.0	138.0	133.0	

	t10_2014	t10_2015	t10_2016	t10_2017	t10_2018	\
institution_name						
University of Rochester	1404.0	1308.0	1310.0	1309.0	1318.0	

	2014_Total	2015_Total	2016_Total	2017_Total	\

```

institution_name
University of Rochester      1566.0      1472.0      1453.0      1447.0

                                2018_Total
institution_name
University of Rochester      1451.0

In [152]: basedata=UR_percentile[['2014_Total', '2015_Total', '2016_Total', '2017_Total', '2018_To

In [153]: basedata

Out[153]:
                                2014_Total  2015_Total  2016_Total  2017_Total  \
institution_name
University of Rochester      1566.0      1472.0      1453.0      1447.0

                                2018_Total
institution_name
University of Rochester      1451.0

In [154]: smalldata=UR_percentile.iloc[:, :10]

In [155]: smalldata1=smalldata.loc[:, [['t1_2014', 't1_2015', 't1_2016', 't1_2017', 't1_2018']]

In [156]: smalldata1

Out[156]:
                                t1_2014  t1_2015  t1_2016  t1_2017  t1_2018
institution_name
University of Rochester      162.0      164.0      143.0      138.0      133.0

In [157]: smalldata2=smalldata.loc[:, [['t10_2014', 't10_2015', 't10_2016', 't10_2017', 't10_2018']]

In [158]: smalldata2

Out[158]:
                                t10_2014  t10_2015  t10_2016  t10_2017  t10_2018
institution_name
University of Rochester      1404.0      1308.0      1310.0      1309.0      1318.0

In [478]: # UR's ScholarlyOutput

so_data_USA.head()

Out[478]:
                                institution_name  2014  2015  2016  \
0  Jet Propulsion Laboratory, California Institut...  1913  1800  2181
1                California Institute of Technology  4518  4347  4553
2                Stanford University  12801  13656  13664
3      Massachusetts Institute of Technology  9645  9957  10023
4                Princeton University  4335  4629  4544

                                2017  2018  Total
0   2052  2200  10146

```

```

1    4567    4884    22869
2    14233   14729   69083
3    10191   10458   50274
4     4635    4891   23034

```

```
In [479]: UR_so=so_data_USA[so_data_USA.institution_name=='University of Rochester']
```

```
In [486]: UR_so
          del UR_so['Total']
```

```
In [513]: UR_so
```

```
Out[513]:
```

	2014	2015	2016	2017	2018
institution_name					
University of Rochester	3602	3540	3515	3633	3842

```
In [514]: combinedata=pd.DataFrame({'2014':[int(162.0),int(1404.0),3602], '2015':[int(164.0),int(1308.0),3540],
                                     '2016':[int(143.0), int(1310.0),3515],
                                     '2017':[int(138.0),int(1309.0),3633],
                                     '2018':[int(133.0), int(1318.0),3842]})
```

```
In [515]: combinedata
```

```
Out[515]:
```

	2014	2015	2016	2017	2018
0	162	164	143	138	133
1	1404	1308	1310	1309	1318
2	3602	3540	3515	3633	3842

```
In [516]: data_1=combinedata.iloc[2,:]
          data_2=combinedata.iloc[1,:]
          data_3=combinedata.iloc[0,:]
```

```
In [532]: data_1
```

```
Out[532]:
```

2014	3602
2015	3540
2016	3515
2017	3633
2018	3842

Name: 2, dtype: int64

```
In [908]: A=pd.DataFrame(data=[data_1[:5]], columns=['2014','2015','2016','2017','2018'])
```

```
In [909]: A
```

```
Out[909]:
```

	2014	2015	2016	2017	2018
2	3602	3540	3515	3633	3842

```
In [911]: A.reset_index(inplace=True, drop=True)
```

```
In [912]: B=pd.DataFrame(data=[data_2[:5]], columns=['2014', '2015', '2016', '2017', '2018'])
```

```
In [914]: B.reset_index(inplace=True, drop=True)
```

```
In [915]: B
```

```
Out[915]:
```

	2014	2015	2016	2017	2018
0	1404	1308	1310	1309	1318

```
In [918]: C=pd.DataFrame(data=[data_3[:5]], columns=['2014', '2015', '2016', '2017', '2018'])
```

```
In [919]: C.reset_index(inplace=True, drop=True)
```

```
In [920]: C
```

```
Out[920]:
```

	2014	2015	2016	2017	2018
0	162	164	143	138	133

```
In [166]: def show_values_on_bars(axes, h_v="v", space=0.8):
    def _show_on_single_plot(ax):
        if h_v == "v":
            for p in ax.patches:
                _x = p.get_x() + p.get_width() / 2
                _y = p.get_y() + p.get_height()
                value = int(p.get_height())
                ax.text(_x, _y, value, ha="center", color='red')
        elif h_v == "h":
            for p in ax.patches:
                _x = p.get_x() + p.get_width() + float(space)
                _y = p.get_y() + p.get_height()
                value = int(p.get_width())
                ax.text(_x, _y, value, ha="left", color='black')

    if isinstance(axes, np.ndarray):
        for idx, ax in np.ndenumerate(axes):
            _show_on_single_plot(ax)
    else:
        _show_on_single_plot(axes)
```

```
In [1024]: import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="whitegrid")
sns.set_style("ticks", {"xtick.major.size": 10, "ytick.major.size": 8})

# Initialize the matplotlib figure
f, ax = plt.subplots(figsize=(6, 15))

# Load the example car crash dataset
#crashes = sns.load_dataset("car_crashes").sort_values("total", ascending=False)
```

```

# Plot the total crashes
sns.set_color_codes("pastel")
g=sns.barplot(data=A,
              label="Total", color="b")

# Plot the crashes where alcohol was involved
sns.set_color_codes("dark")
g=sns.barplot(data=B,
              label="Top 10%", color="b")

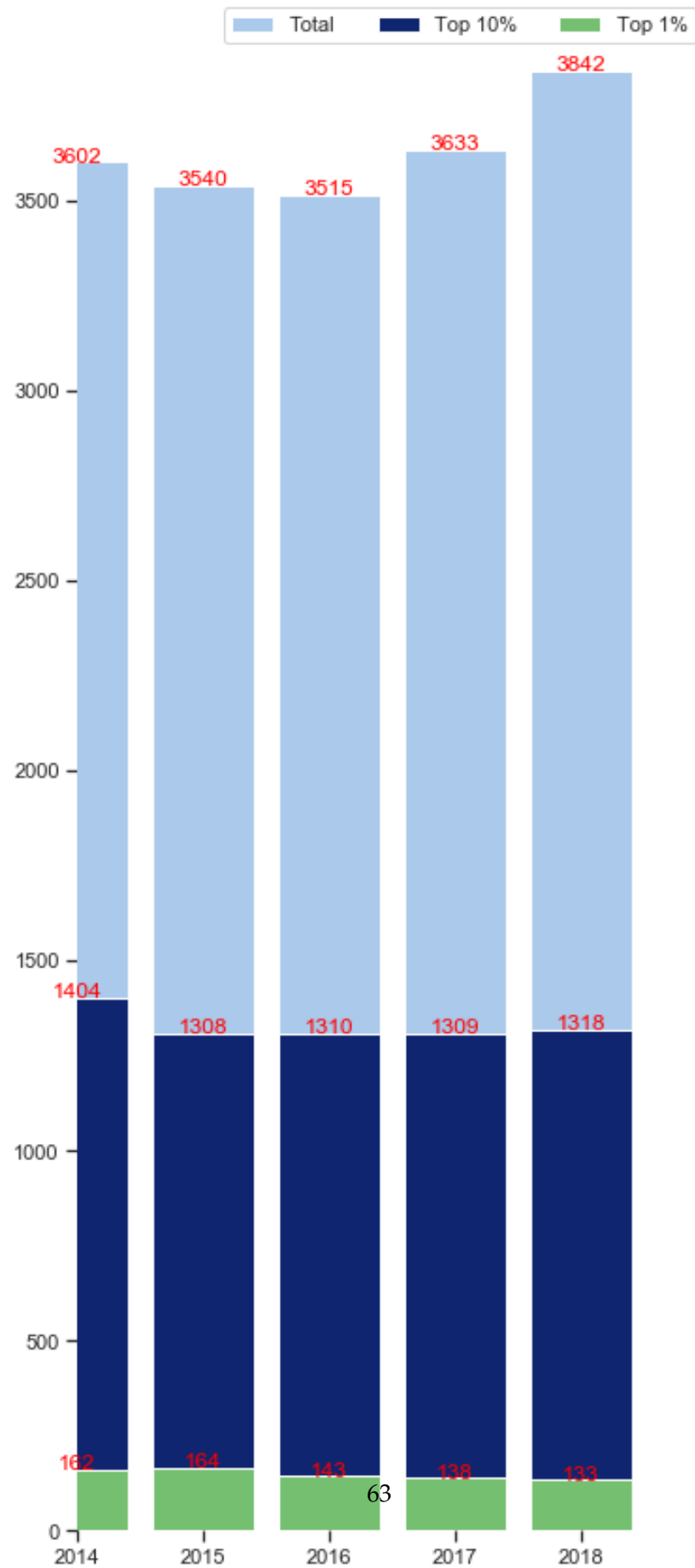
# Plot the crashes where alcohol was involved
sns.set_color_codes("muted")
g=sns.barplot(data=C,
              label="Top 1%", color="g")

show_values_on_bars(g, "v", 0.8)

# Add a legend and informative axis label
plt.yticks(np.arange(0, 4000, step=500))
plt.xticks(np.arange(5), ('2014', '2015', '2016', '2017', '2018'))
ax.legend(ncol=3, loc="upper right", frameon=True)
ax.set(xlim=(0,5), ylabel="",
      title="U of R publication output: total, top 1 % and top 10 % highly cited p
sns.despine(left=True, bottom=True)

```

U of R publication output: total, top 1 % and top 10 % highly cited publs



37 From 2014-2018 ,our top 1% cited publs
 38 and top10% cited pulbs slightly dropped.
 39 However, since our 2018 total publs increased a lot,
 40 it would definitely influence our overall research performance.

41 Trends in FWCI values of total U of R publication output

```
In [539]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\FNCI"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\FNCI
```

```
In [540]: FWCI_all=pd.read_csv('THE_ALLUNI_FWCI.csv')
```

```
In [541]: FWCI_all.head()
```

```
Out[541]:
```

	country	countryCode	institution_id	\
0	United Kingdom	GBR	315091	
1	United States	USA	508092	
2	United States	USA	508021	
3	United Kingdom	GBR	315068	
4	United States	USA	508219	

	link	\
0	{ '@href': 'https://api.elsevier.com/analytics/...	
1	{ '@href': 'https://api.elsevier.com/analytics/...	
2	{ '@href': 'https://api.elsevier.com/analytics/...	
3	{ '@href': 'https://api.elsevier.com/analytics/...	
4	{ '@href': 'https://api.elsevier.com/analytics/...	

	institution_name	\
0	University of Oxford	
1	Jet Propulsion Laboratory, California Institut...	
2	California Institute of Technology	
3	University of Cambridge	
4	Stanford University	

	metricType	2014	2015	2016	2017	\
0	FieldWeightedCitationImpact	2.232452	2.178834	2.202485	1.966025	
1	FieldWeightedCitationImpact	1.611136	1.462793	1.656759	1.470790	
2	FieldWeightedCitationImpact	1.890797	1.740487	1.921985	1.847315	


```

3 FieldWeightedCitationImpact  1.904510  1.990053  2.050378  1.946377
4 FieldWeightedCitationImpact  2.445251  2.476393  2.568147  2.269981

```

```

          2018
0  1.804821
1  1.346227
2  1.605074
3  1.763683
4  2.244260

```

```
In [1064]: UR_FWCI=FWCI_all[FWCI_all.institution_name=='University of Rochester']
```

```
In [1065]: UR_FWCI=UR_FWCI.iloc[:, -7:]
```

```
In [1066]: del UR_FWCI['metricType']
```

```
In [1067]: UR_FWCI
```

```

Out[1067]:
          institution_name      2014      2015      2016      2017 \
1259  University of Rochester  1.827581  2.117681  1.646356  1.700714

          2018
1259  1.717541

```

```
In [1043]: UR_FWCI.reset_index(inplace=True, drop=True)
```

```
In [1068]: UR_FWCI.set_index('institution_name', inplace=True, drop=True)
```

```
In [1083]: UR_FWCI.reset_index(inplace=True)
```

```
In [1084]: UR_FWCI
```

```

Out[1084]:
          institution_name      2014      2015      2016      2017      2018
0  University of Rochester  1.827581  2.117681  1.646356  1.700714  1.717541

```

42 UofR FWCI

```

In [69]: def show_values_on_bars_1(axs, h_v="v", space=0.4):
          def _show_on_single_plot_1(ax):
              if h_v == "v":
                  for p in ax.patches:
                      _x = p.get_x() + p.get_width() / 2
                      _y = p.get_y() + p.get_height()
                      value = round(p.get_height(),2)
                      ax.text(_x, _y, value, ha="center", color='red')
              elif h_v == "h":
                  for p in ax.patches:
                      _x = p.get_x() + p.get_width() + float(space)
                      _y = p.get_y() + p.get_height()

```

```

        value = round(p.get_width(),2)
        ax.text(_x, _y, value, ha="left")

    if isinstance(axes, np.ndarray):
        for idx, ax in np.ndenumerate(axes):
            _show_on_single_plot_1(ax)
    else:
        _show_on_single_plot_1(axes)

```

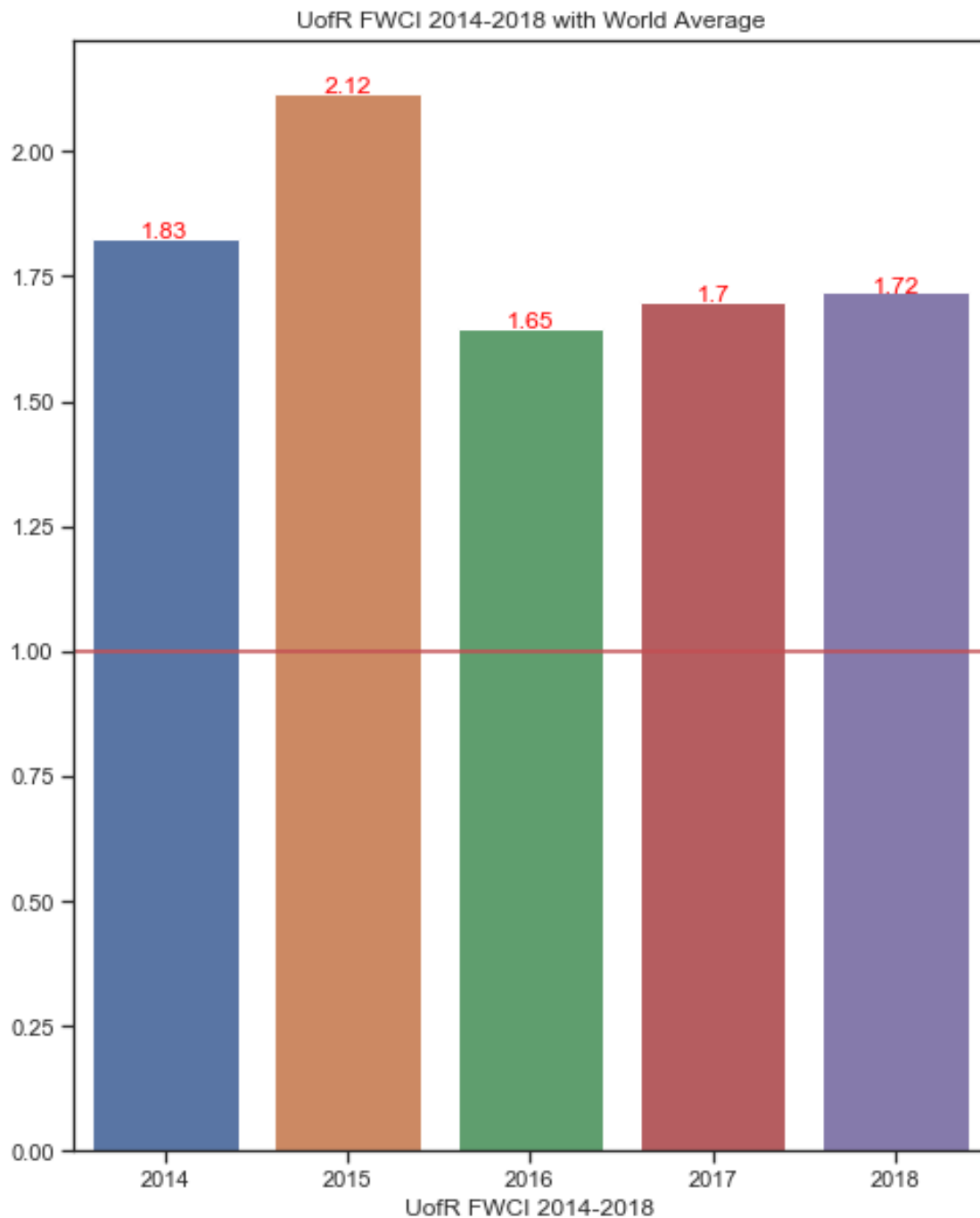
43 UofR FWCI have always been above global average which is 1.00

```

In [1094]: import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="whitegrid")
sns.set_style("ticks", {"xtick.major.size": 10, "ytick.major.size": 8})

# Initialize the matplotlib figure
f, ax = plt.subplots(figsize=(8, 10))
g= sns.barplot(data=UR_FWCI)
plt.axhline(1.00, ls='-', color='r')
plt.title('UofR FWCI 2014-2018 with World Average')
plt.xlabel("UofR FWCI 2014-2018")
#plt.ylabel("Filed-weighted Cited Index")
show_values_on_bars_1(g, 'v', 0.3)

```



44 Comparator analysis: top 10 % highly cited publications for USA universities

In [574]: `USA_pp.head()`

```

Out [574]:
           institution_name  t1_2014  t1_2015  \
0  Jet Propulsion Laboratory, California Institut...    74.0    82.0
1                California Institute of Technology   305.0   245.0
2                Stanford University    954.0  1016.0
3      Massachusetts Institute of Technology    820.0   818.0
4                Princeton University    271.0   269.0

           t1_2016  t1_2017  t1_2018  t10_2014  t10_2015  t10_2016  t10_2017  \
0         83.0     71.0    123.0    455.0    483.0    694.0    687.0
1        259.0    253.0    293.0   1709.0   1579.0   2051.0   1906.0
2       1073.0   1027.0   1025.0   5558.0   6068.0   5974.0   6233.0
3        918.0    822.0    869.0   4401.0   4418.0   4689.0   4613.0
4        227.0    241.0    253.0   1810.0   1740.0   1916.0   1871.0

           t10_2018  2014_Total  2015_Total  2016_Total  2017_Total  2018_Total
0         679.0        529.0        565.0        777.0        758.0        802.0
1        1967.0       2014.0       1824.0       2310.0       2159.0       2260.0
2        6307.0       6512.0       7084.0       7047.0       7260.0       7332.0
3        4781.0       5221.0       5236.0       5607.0       5435.0       5650.0
4        1990.0       2081.0       2009.0       2143.0       2112.0       2243.0

```

```

In [159]: UR_peer=['Boston University','Carnegie Mellon University','Case Western Reserve Univ
           'Northwestern University','Vanderbilt University','Washington University','J
           'Stanford University','Tulane University','University of Chicago','University

```

```

In [3]: import pandas as pd

```

```

In [6]: UR_peer_df=pd.DataFrame({'UR_Peer':UR_peer})

```

```

In [8]: UR_peer_df=UR_peer_df.iloc[:14,: ]

```

```

In [9]: UR_peer_df['UR_Peer']

```

```

Out [9]: 0          Boston University
1      Carnegie Mellon University
2      Case Western Reserve University
3          Duke University
4          Emory University
5      Northwestern University
6          Vanderbilt University
7      Washington University
8      Johns Hopkins University
9          New York University
10         Stanford University
11         Tulane University
12      University of Chicago
13      University of Pennsylvania
Name: UR_Peer, dtype: object

```

45 Get UofR's Global set's Publication in Top Journal Percentile

```
In [160]: chuck=[]
          for name in UR_peer_df['UR_Peer']:
              chuck.append(USA_pp[USA_pp.institution_name==name])
```

```
In [161]: DF=pd.concat(chuck, ignore_index=True)
```

```
In [162]: DF.head()
```

```
Out[162]:
```

	institution_name	t1_2014	t1_2015	t1_2016	t1_2017	\
0	Boston University	309.0	320.0	365.0	351.0	
1	Case Western Reserve University	174.0	172.0	206.0	198.0	
2	Duke University	541.0	508.0	563.0	543.0	
3	Emory University	277.0	348.0	318.0	348.0	
4	Northwestern University	511.0	526.0	621.0	527.0	

	t1_2018	t10_2014	t10_2015	t10_2016	t10_2017	t10_2018	2014_Total	\
0	369.0	2219.0	2296.0	2421.0	2580.0	2564.0	2528.0	
1	163.0	1516.0	1551.0	1572.0	1594.0	1513.0	1690.0	
2	478.0	3823.0	3910.0	3747.0	3828.0	3875.0	4364.0	
3	274.0	2411.0	2494.0	2348.0	2464.0	2390.0	2688.0	
4	603.0	3313.0	3412.0	3524.0	3682.0	3636.0	3824.0	

	2015_Total	2016_Total	2017_Total	2018_Total
0	2616.0	2786.0	2931.0	2933.0
1	1723.0	1778.0	1792.0	1676.0
2	4418.0	4310.0	4371.0	4353.0
3	2842.0	2666.0	2812.0	2664.0
4	3938.0	4145.0	4209.0	4239.0

```
In [178]: UR_percentile=USA_pp[USA_pp.institution_name=='University of Rochester']
```

```
In [179]: UR_percentile=UR_percentile.reset_index()
```

```
In [172]: Global_top10=DF.loc[:][['institution_name','t10_2014','t10_2015','t10_2016','t10_2017','t10_2018']]
```

```
In [180]: UR_pcer_top10=UR_percentile.loc[:][['institution_name','t10_2014','t10_2015','t10_2016','t10_2017','t10_2018']]
```

```
In [174]: Global_top10.head()
```

```
Out[174]:
```

	institution_name	t10_2014	t10_2015	t10_2016	t10_2017	\
0	Boston University	2219.0	2296.0	2421.0	2580.0	
1	Case Western Reserve University	1516.0	1551.0	1572.0	1594.0	
2	Duke University	3823.0	3910.0	3747.0	3828.0	
3	Emory University	2411.0	2494.0	2348.0	2464.0	
4	Northwestern University	3313.0	3412.0	3524.0	3682.0	

	t10_2018
0	2564.0
1	1513.0
2	3875.0
3	2390.0
4	3636.0

```

0    2564.0
1    1513.0
2    3875.0
3    2390.0
4    3636.0

```

```
In [175]: Global_top10['Top10_Total']=Global_top10.sum(axis=1)
```

```
In [176]: Global_top10.head()
```

```

Out[176]:
      institution_name  t10_2014  t10_2015  t10_2016  t10_2017  \
0      Boston University    2219.0    2296.0    2421.0    2580.0
1  Case Western Reserve University    1516.0    1551.0    1572.0    1594.0
2      Duke University     3823.0    3910.0    3747.0    3828.0
3      Emory University     2411.0    2494.0    2348.0    2464.0
4  Northwestern University     3313.0    3412.0    3524.0    3682.0

      t10_2018  Top10_Total
0      2564.0      12080.0
1      1513.0       7746.0
2      3875.0     19183.0
3      2390.0     12107.0
4      3636.0     17567.0

```

```
In [169]: len(Global_top10)
```

```
Out[169]: 12
```

```
In [181]: UR_pcer_top10
```

```

Out[181]:
      institution_name  t10_2014  t10_2015  t10_2016  t10_2017  t10_2018
0  University of Rochester    1404.0    1308.0    1310.0    1309.0    1318.0

```

```
In [182]: UR_pcer_top10['Top10_Total']=UR_pcer_top10.sum(axis=1)
```

```
In [183]: UR_pcer_top10
```

```

Out[183]:
      institution_name  t10_2014  t10_2015  t10_2016  t10_2017  t10_2018  \
0  University of Rochester    1404.0    1308.0    1310.0    1309.0    1318.0

      Top10_Total
0          6649.0

```

```
In [184]: Gall=pd.concat([Global_top10, UR_pcer_top10])
```

```
In [185]: len(Gall)
```

```
Out[185]: 13
```

```
In [647]: import re
```

```

In [186]: abb=[]
          for i in Gall.institution_name:
              abb.append(i.split("\t")[0].strip(" "))
          abb # not work

Out[186]: ['Boston University',
           'Case Western Reserve University',
           'Duke University',
           'Emory University',
           'Northwestern University',
           'Vanderbilt University',
           'Johns Hopkins University',
           'New York University',
           'Stanford University',
           'Tulane University',
           'University of Chicago',
           'University of Pennsylvania',
           'University of Rochester']

In [187]: Gall['UniAbbr']=['Boston','CWRU','Duke','Emory','Northwestern','Vanderbilt','JohnsHop
In [188]: Gall=Gall.sort_values(by='Top10_Total', ascending=False)

```

46 Comparator analysis: top 10% highly cited publications UR and GlobalPeers

```

In [192]: sns.set(style="whitegrid")

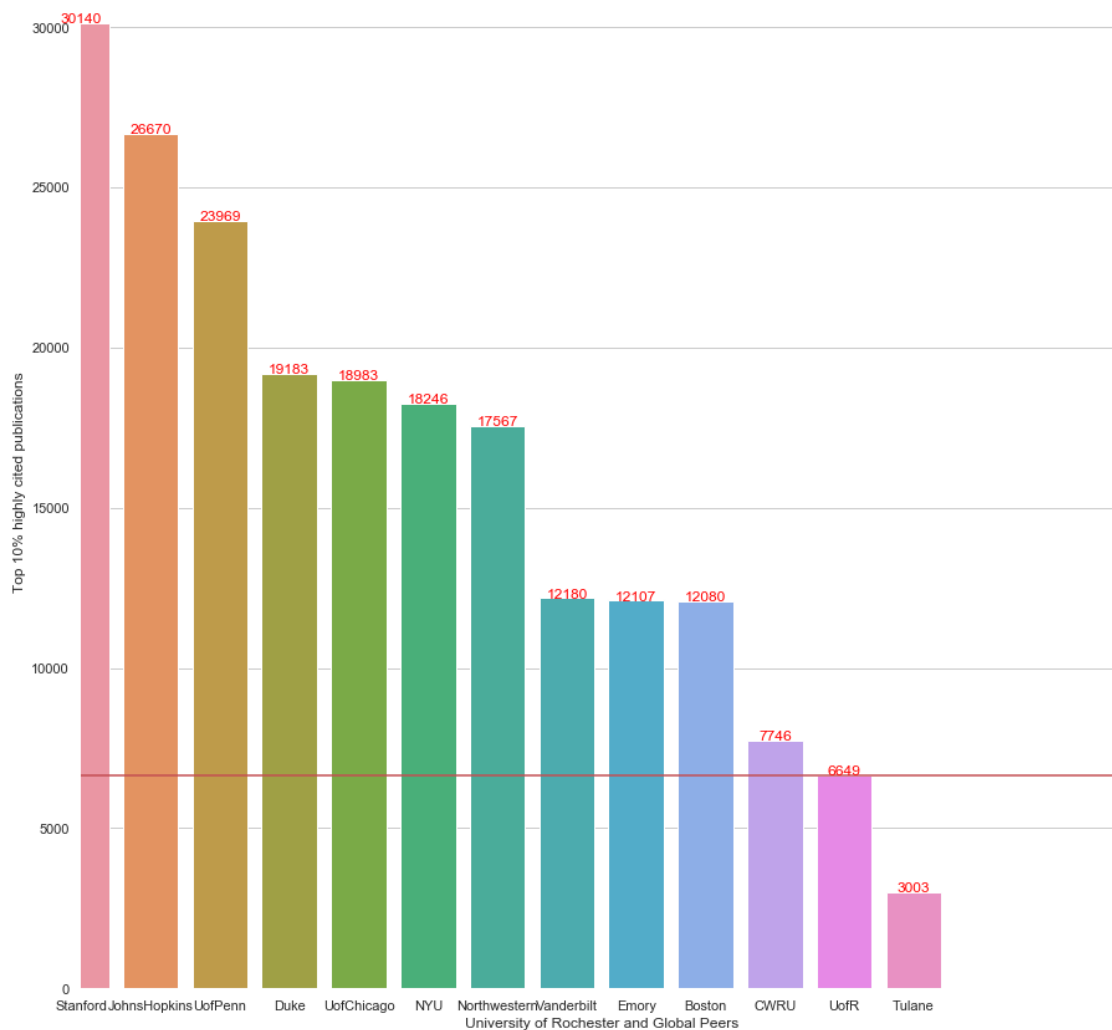
          # Initialize the matplotlib figure
          f, ax = plt.subplots(figsize=(15, 15))
          g=sns.barplot(x=Gall.UniAbbr, y=Gall.Top10_Total, data=Gall)
          plt.axhline(6649, ls='--', color='r') # can add a red base line for UofR value

          #ax.text(Gall.UniAbbr, Gall.Top10_Total,color='black', ha="center")

          # Add a legend and informative axis label
          #ax.legend(ncol=12, loc="upper right", frameon=True)
          ax.set(xlim=(0, 15),
                  xlabel="University of Rochester and Global Peers", ylabel="Top 10% highly cit
          sns.despine(left=True, bottom=True)

          show_values_on_bars(g, 'v', 0.5)

```



47 Among our other 12 USA peers,

48 our top 10% highly-cited pulbs is relatively fewer

49 Comparator analysis: Field-weighted Citation Impact

```
In [193]: fwci_data.head()
```

```
Out[193]:
```

	country	countryCode	institution_id \
0	United Kingdom	GBR	315091
1	United States	USA	508092
2	United States	USA	508021
3	United Kingdom	GBR	315068

4 United States USA 508219

```
link \
0 {'@href': 'https://api.elsevier.com/analytics/...
1 {'@href': 'https://api.elsevier.com/analytics/...
2 {'@href': 'https://api.elsevier.com/analytics/...
3 {'@href': 'https://api.elsevier.com/analytics/...
4 {'@href': 'https://api.elsevier.com/analytics/...
```

```
institution_name \
0 University of Oxford
1 Jet Propulsion Laboratory, California Institut...
2 California Institute of Technology
3 University of Cambridge
4 Stanford University
```

	metricType	2014	2015	2016	2017 \
0	FieldWeightedCitationImpact	2.232452	2.178834	2.202485	1.966025
1	FieldWeightedCitationImpact	1.611136	1.462793	1.656759	1.470790
2	FieldWeightedCitationImpact	1.890797	1.740487	1.921985	1.847315
3	FieldWeightedCitationImpact	1.904510	1.990053	2.050378	1.946377
4	FieldWeightedCitationImpact	2.445251	2.476393	2.568147	2.269981

2018

0	1.804821
1	1.346227
2	1.605074
3	1.763683
4	2.244260

In [194]: fwci_data.tail()

Out[194]:

	country	countryCode	institution_id \
1263	Saudi Arabia	SAU	703099
1264	Turkey	TUR	705124
1265	Georgia	GEO	204001
1266	United States	USA	508335
1267	United States	USA	508059

```
link \
1263 {'@href': 'https://api.elsevier.com/analytics/...
1264 {'@href': 'https://api.elsevier.com/analytics/...
1265 {'@href': 'https://api.elsevier.com/analytics/...
1266 {'@href': 'https://api.elsevier.com/analytics/...
1267 {'@href': 'https://api.elsevier.com/analytics/...
```

```
institution_name \
1263 Imam Abdulrahman Bin Faisal University
```

```

1264 Istanbul Medipol University
1265 Ivane Javakhishvili Tbilisi State University
1266 University of Rochester
1267 Emory University

```

	metricType	2014	2015	2016	2017 \
1263	FieldWeightedCitationImpact	0.698082	0.649441	0.772378	0.787842
1264	FieldWeightedCitationImpact	0.661692	0.577872	0.552237	0.635915
1265	FieldWeightedCitationImpact	1.556471	1.550585	1.177011	1.246151
1266	FieldWeightedCitationImpact	1.827581	2.117681	1.646356	1.700714
1267	FieldWeightedCitationImpact	1.999690	2.186228	2.209265	1.948478

	2018
1263	0.880861
1264	0.666751
1265	1.725649
1266	1.717541
1267	1.967104

```
In [195]: US_fwci=fwci_data[fwci_data.countryCode=='USA']
```

```
In [196]: US_fwci.head()
```

```

Out[196]:
   country countryCode  institution_id \
1  United States      USA           508092
2  United States      USA           508021
4  United States      USA           508219
5  United States      USA           508111
6  United States      USA           508191

                                     link \
1  {'@href': 'https://api.elsevier.com/analytics/...
2  {'@href': 'https://api.elsevier.com/analytics/...
4  {'@href': 'https://api.elsevier.com/analytics/...
5  {'@href': 'https://api.elsevier.com/analytics/...
6  {'@href': 'https://api.elsevier.com/analytics/...

                                     institution_name \
1  Jet Propulsion Laboratory, California Institut...
2                                     California Institute of Technology
4                                     Stanford University
5                                     Massachusetts Institute of Technology
6                                     Princeton University

   metricType  2014  2015  2016  2017 \
1  FieldWeightedCitationImpact  1.611136  1.462793  1.656759  1.470790
2  FieldWeightedCitationImpact  1.890797  1.740487  1.921985  1.847315
4  FieldWeightedCitationImpact  2.445251  2.476393  2.568147  2.269981

```

```

5 FieldWeightedCitationImpact 2.271606 2.301666 2.355942 2.132760
6 FieldWeightedCitationImpact 2.111493 2.144071 2.101741 1.906495

```

```

2018
1 1.346227
2 1.605074
4 2.244260
5 1.971292
6 1.919808

```

```
In [200]: UR_peer_df
```

```

Out[200]:
          UR_Peer
0      Boston University
1  Carnegie Mellon University
2  Case Western Reserve University
3      Duke University
4      Emory University
5  Northwestern University
6  Vanderbilt University
7  Washington University
8  Johns Hopkins University
9    New York University
10   Stanford University
11   Tulane University
12   University of Chicago
13  University of Pennsylvania

```

```
In [212]: UR=pd.DataFrame({'UR_Peer':['University of Rochester']})
```

```
In [214]: UR_peer_df=pd.concat([UR_peer_df, UR])
```

```
In [215]: UR_peer_df.reset_index(inplace=True)
```

```
In [198]: len(Gall.institution_name) # Global peers and UofR
```

```
Out[198]: 13
```

```
In [216]: chuck=[]
```

```

for name in UR_peer_df.UR_Peer:
    if US_fwci[US_fwci.institution_name==name] is not None:
        chuck.append(US_fwci[US_fwci.institution_name==name])

```

```
In [217]: UR_Peer_FWCI=pd.concat(chuck, ignore_index=True)
```

```
In [24]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\FNCI"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\FNCI
```

```
In [218]: UR_Peer_FWCI.to_csv('UR_Global_Peer_FWCI_Comparison.csv', index=False)
```

```
In [29]: ALL_FWCI=pd.read_csv('THE_ALLUNI_FWCI.csv')
```

```
In [30]: ALL_FWCI=ALL_FWCI.drop_duplicates()
```

```
In [31]: ALL_FWCI.head()
```

```
Out[31]:
```

	country	countryCode	institution_id	\	link	\	institution_name	\	metricType	2014	2015	2016	2017	\	2018
0	United Kingdom	GBR	315091		{ '@href': 'https://api.elsevier.com/analytics/...		University of Oxford		FieldWeightedCitationImpact	2.232452	2.178834	2.202485	1.966025		1.804821
1	United States	USA	508092		{ '@href': 'https://api.elsevier.com/analytics/...		Jet Propulsion Laboratory, California Institut...		FieldWeightedCitationImpact	1.611136	1.462793	1.656759	1.470790		1.346227
2	United States	USA	508021		{ '@href': 'https://api.elsevier.com/analytics/...		California Institute of Technology		FieldWeightedCitationImpact	1.890797	1.740487	1.921985	1.847315		1.605074
3	United Kingdom	GBR	315068		{ '@href': 'https://api.elsevier.com/analytics/...		University of Cambridge		FieldWeightedCitationImpact	1.904510	1.990053	2.050378	1.946377		1.763683
4	United States	USA	508219		{ '@href': 'https://api.elsevier.com/analytics/...		Stanford University		FieldWeightedCitationImpact	2.445251	2.476393	2.568147	2.269981		2.244260

```
In [219]: UR_Peer_FWCI=UR_Peer_FWCI.iloc[:, -7:]
```

```
In [222]: UR_Peer_FWCI=UR_Peer_FWCI.drop_duplicates()
```

```
In [223]: abb=[]
for name in Gall.UniAbbr:
    abb.append(name)
abb
```

```
Out [223]: ['Stanford',
            'JohnsHopkins',
            'UofPenn',
            'Duke',
            'UofChicago',
            'NYU',
            'Northwestern',
            'Vanderbilt',
            'Emory',
            'Boston',
            'CWRU',
            'UofR',
            'Tulane']
```

```
In [764]: UR_Peer_FWCI=UR_Peer_FWCI.drop_duplicates()
```

```
In [224]: UR_Peer_FWCI.reset_index(inplace=True, drop=True)
```

```
In [225]: UR_Peer_FWCI
```

```
Out [225]:
```

	institution_name	metricType	2014	\
0	Boston University	FieldWeightedCitationImpact	2.102325	
1	Case Western Reserve University	FieldWeightedCitationImpact	1.785904	
2	Duke University	FieldWeightedCitationImpact	2.060966	
3	Emory University	FieldWeightedCitationImpact	1.999690	
4	Northwestern University	FieldWeightedCitationImpact	1.814437	
5	Vanderbilt University	FieldWeightedCitationImpact	1.935092	
6	Johns Hopkins University	FieldWeightedCitationImpact	2.039671	
7	New York University	FieldWeightedCitationImpact	2.036910	
8	Stanford University	FieldWeightedCitationImpact	2.445251	
9	Tulane University	FieldWeightedCitationImpact	1.272190	
10	University of Chicago	FieldWeightedCitationImpact	1.899739	
11	University of Pennsylvania	FieldWeightedCitationImpact	2.049064	
12	University of Rochester	FieldWeightedCitationImpact	1.827581	

	2015	2016	2017	2018
0	2.017788	2.082099	1.699757	1.916557
1	1.952037	1.955858	1.773011	1.861512
2	2.188656	1.971610	1.903780	1.848280
3	2.186228	2.209265	1.948478	1.967104
4	2.036129	2.104611	2.028618	1.968102
5	2.053712	1.799374	1.751226	1.698510
6	2.063183	2.086808	1.941970	1.903996
7	2.071903	1.961854	1.787898	1.819667
8	2.476393	2.568147	2.269981	2.244260
9	1.784176	1.650411	1.381838	1.569006
10	1.866923	2.041074	1.860755	1.800015
11	2.078196	2.042546	1.896038	1.865539
12	2.117681	1.646356	1.700714	1.717541

```
In [230]: UR_Peer_FWCI['UniAbbr']=['Boston', 'CWRU', 'Duke', 'Emory', 'Northwestern', 'Vanderbilt',
                                     'UofC', 'UofPenn', 'UofR']
```

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html>

```
In [231]: UR_Peer_FWCI.head()
```

```
Out [231]:
```

	institution_name	metricType	2014	\
0	Boston University	FieldWeightedCitationImpact	2.102325	
1	Case Western Reserve University	FieldWeightedCitationImpact	1.785904	
2	Duke University	FieldWeightedCitationImpact	2.060966	
3	Emory University	FieldWeightedCitationImpact	1.999690	
4	Northwestern University	FieldWeightedCitationImpact	1.814437	

	2015	2016	2017	2018	UniAbbr
0	2.017788	2.082099	1.699757	1.916557	Boston
1	1.952037	1.955858	1.773011	1.861512	CWRU
2	2.188656	1.971610	1.903780	1.848280	Duke
3	2.186228	2.209265	1.948478	1.967104	Emory
4	2.036129	2.104611	2.028618	1.968102	Northwestern

```
In [234]: UR_Peer_FWCI['AVERAGE_FWCI']=round(UR_Peer_FWCI[['2014', '2015', '2016', '2017', '2018']])
```

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html>
 """Entry point for launching an IPython kernel.

```
In [235]: UR_Peer_FWCI=UR_Peer_FWCI.sort_values(by='AVERAGE_FWCI', ascending=False)
```

```
In [236]: UR_Peer_FWCI.head()
```

```
Out [236]:
```

	institution_name	metricType	2014	2015	\
8	Stanford University	FieldWeightedCitationImpact	2.445251	2.476393	
3	Emory University	FieldWeightedCitationImpact	1.999690	2.186228	
6	Johns Hopkins University	FieldWeightedCitationImpact	2.039671	2.063183	
2	Duke University	FieldWeightedCitationImpact	2.060966	2.188656	
4	Northwestern University	FieldWeightedCitationImpact	1.814437	2.036129	

	2016	2017	2018	UniAbbr	AVERAGE_FWCI
--	------	------	------	---------	--------------

8	2.568147	2.269981	2.244260	Stanford	2.4008
3	2.209265	1.948478	1.967104	Emory	2.0622
6	2.086808	1.941970	1.903996	JHopkins	2.0071
2	1.971610	1.903780	1.848280	Duke	1.9947
4	2.104611	2.028618	1.968102	Northwestern	1.9904

```
In [237]: UR_Peer_FWCI[UR_Peer_FWCI.UniAbbr=='UofR']
```

```
Out[237]:
```

	institution_name	metricType	2014	2015	\
12	University of Rochester	FieldWeightedCitationImpact	1.827581	2.117681	

	2016	2017	2018	UniAbbr	AVERAGE_FWCI
12	1.646356	1.700714	1.717541	UofR	1.802

50 Comparatory analysis: Field-weighted Citation Impact

51 Our average FWCI 2014-2018 is 1.8,

52 but most of our USA peers have higher FWCI,

**53 this may explain why our overall score did not reflect our good
FWCI**

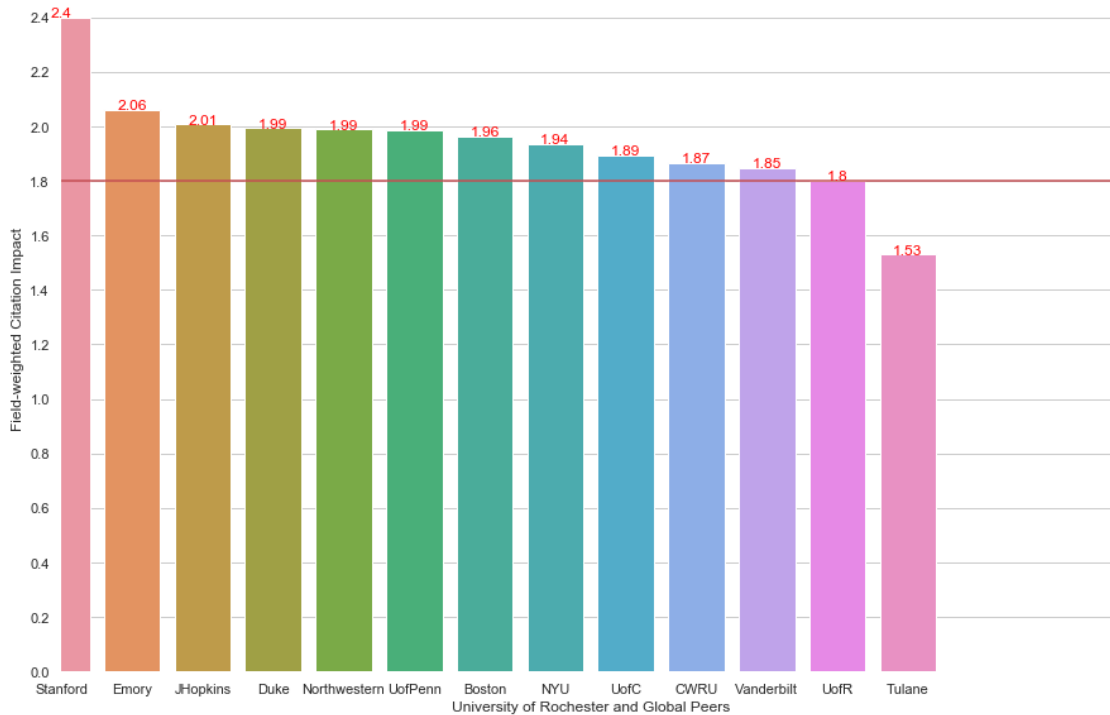
```
In [238]: sns.set(style="whitegrid")
```

```
# Initialize the matplotlib figure
f, ax = plt.subplots(figsize=(15, 10))
g=sns.barplot(x=UR_Peer_FWCI.UniAbbr, y=UR_Peer_FWCI.AVERAGE_FWCI, data=UR_Peer_FWCI)
plt.axhline(1.802, ls='--', color='r')

show_values_on_bars_1(g, 'v', 0.5)

#ax.text(Gall.UniAbbr, Gall.Top10_Total,color='black', ha="center")

# Add a legend and informative axis label
#ax.legend(ncol=12, loc="upper right", frameon=True)
plt.yticks(np.arange(0, 2.5, step=0.2))
ax.set(xlim=(0, 15),
       xlabel="University of Rochester and Global Peers", ylabel="Field-weighted Citat
sns.despine(left=True, bottom=True)
```



54 Comparatory analysis:

55 top publication output and highly cited publications

```
In [19]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\PubTopJournalP
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\PubTopJournalPercentile
```

```
In [2]: import pandas as pd
import numpy as np
```

```
In [20]: A_PP=pd.read_csv('THE_ALLUNI_PP.csv')
```

```
In [21]: USA_PP=A_PP[A_PP.countryCode=='USA']
```

```
In [22]: USA_PP=USA_PP.drop_duplicates()
```

```
In [32]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\research_perfor
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\research_performance_Profile
```

```
In [33]: UR_peer=pd.read_csv('UR_GloPeers_Research_Performance_Profile.csv')
```



```
In [34]: chuck=[]
        for name in UR_peer.institution_name:
            chuck.append(USA_PP[USA_PP.institution_name==name])
```

```
In [35]: UR_Peers_PP=pd.concat(chuck, ignore_index=True)
```

```
In [36]: UR_Peers_PP
```

```
Out[36]:
```

	country	countryCode	institution_id	\
0	United States	USA	508219	
1	United States	USA	508094	
2	United States	USA	508331	
3	United States	USA	508166	
4	United States	USA	508053	
5	United States	USA	508270	
6	United States	USA	508175	
7	United States	USA	508363	
8	United States	USA	508059	
9	United States	USA	508013	
10	United States	USA	508032	
11	United States	USA	508335	
12	United States	USA	508239	

	link	\
0	{ '@href': 'https://api.elsevier.com/analytics/...	
1	{ '@href': 'https://api.elsevier.com/analytics/...	
2	{ '@href': 'https://api.elsevier.com/analytics/...	
3	{ '@href': 'https://api.elsevier.com/analytics/...	
4	{ '@href': 'https://api.elsevier.com/analytics/...	
5	{ '@href': 'https://api.elsevier.com/analytics/...	
6	{ '@href': 'https://api.elsevier.com/analytics/...	
7	{ '@href': 'https://api.elsevier.com/analytics/...	
8	{ '@href': 'https://api.elsevier.com/analytics/...	
9	{ '@href': 'https://api.elsevier.com/analytics/...	
10	{ '@href': 'https://api.elsevier.com/analytics/...	
11	{ '@href': 'https://api.elsevier.com/analytics/...	
12	{ '@href': 'https://api.elsevier.com/analytics/...	

	institution_name	metricType	\
0	Stanford University	PublicationsInTopJournalPercentiles	
1	Johns Hopkins University	PublicationsInTopJournalPercentiles	
2	University of Pennsylvania	PublicationsInTopJournalPercentiles	
3	New York University	PublicationsInTopJournalPercentiles	
4	Duke University	PublicationsInTopJournalPercentiles	
5	University of Chicago	PublicationsInTopJournalPercentiles	
6	Northwestern University	PublicationsInTopJournalPercentiles	
7	Vanderbilt University	PublicationsInTopJournalPercentiles	
8	Emory University	PublicationsInTopJournalPercentiles	

9	Boston University	PublicationsInTopJournalPercentiles
10	Case Western Reserve University	PublicationsInTopJournalPercentiles
11	University of Rochester	PublicationsInTopJournalPercentiles
12	Tulane University	PublicationsInTopJournalPercentiles

	t1_2014	t1_2015	t1_2016	t1_2017	...	t25_2014	t25_2015	t25_2016	\
0	954.0	1016.0	1073.0	1027.0	...	8211.0	8963.0	8819.0	
1	615.0	610.0	754.0	796.0	...	8211.0	8685.0	8866.0	
2	693.0	719.0	777.0	701.0	...	7190.0	7286.0	7455.0	
3	475.0	492.0	509.0	495.0	...	5484.0	5704.0	5969.0	
4	541.0	508.0	563.0	543.0	...	5910.0	5894.0	5863.0	
5	582.0	535.0	569.0	515.0	...	5553.0	5666.0	5705.0	
6	511.0	526.0	621.0	527.0	...	5070.0	5210.0	5301.0	
7	295.0	339.0	320.0	316.0	...	3876.0	3917.0	4054.0	
8	277.0	348.0	318.0	348.0	...	3967.0	3923.0	3854.0	
9	309.0	320.0	365.0	351.0	...	3576.0	3668.0	3775.0	
10	174.0	172.0	206.0	198.0	...	2619.0	2569.0	2643.0	
11	162.0	164.0	143.0	138.0	...	2163.0	2125.0	2131.0	
12	62.0	78.0	78.0	70.0	...	988.0	1049.0	1070.0	

	t25_2017	t25_2018	t25_percent2014	t25_percent2015	t25_percent2016	\
0	9276.0	9716.0	75.164780	76.541420	74.535164	
1	9088.0	9362.0	73.608246	73.921190	72.672130	
2	7541.0	8363.0	73.150880	72.678310	71.641365	
3	6262.0	6624.0	69.373820	68.582420	67.097570	
4	5972.0	6376.0	75.885980	74.054530	73.077410	
5	5613.0	5999.0	75.142080	74.425330	74.100530	
6	5606.0	5733.0	74.177030	74.280014	73.594340	
7	4218.0	4354.0	73.772360	73.310875	71.612785	
8	3975.0	4143.0	73.708660	72.246780	68.907560	
9	3967.0	4141.0	74.006620	74.311190	74.077705	
10	2683.0	2670.0	70.214480	67.392440	65.631990	
11	2130.0	2309.0	70.364340	69.331154	67.436710	
12	1088.0	1125.0	64.786890	65.358260	66.171930	

	t25_percent2017	t25_percent2018
0	75.279980	74.755710
1	71.921490	71.226420
2	70.635070	70.389700
3	67.449370	65.851470
4	71.908485	72.347670
5	73.190765	74.043450
6	73.840890	73.368310
7	72.486680	70.877420
8	67.407160	68.186310
9	73.503800	72.853620
10	65.137170	62.441532
11	66.645805	66.522610

12 66.099640 66.137566

[13 rows x 46 columns]

```
In [38]: UR_Peers_top1=UR_Peers_PP.loc[:,[['institution_name','t1_2014','t1_2015','t1_2016','t1_2017','t1_2018','t1_2019','t1_2020']]]
```

```
In [39]: UR_Peers_top10=UR_Peers_PP.loc[:,[['institution_name','t10_2014','t10_2015','t10_2016','t10_2017','t10_2018','t10_2019','t10_2020']]]
```

```
In [40]: UR_Peers_top1['top1_all']=UR_Peers_top1.sum(axis=1)
```

```
In [41]: UR_Peers_top10['top10_all']=UR_Peers_top10.sum(axis=1)
```

```
In [42]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\ScholarlyOutput"
```

C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\ScholarlyOutput

```
In [43]: ALL_so=pd.read_csv('THE_ALLUNI_SO.csv')
```

```
In [45]: USA_so=ALL_so[ALL_so.countryCode=='USA']
```

```
In [49]: chuck=[]
        for name in UR_Peers_PP.institution_name:
            chuck.append(USA_so[USA_so.institution_name==name])
        UR_Peers_SO=pd.concat(chuck, ignore_index=True)
```

```
In [52]: UR_Peers_SO=UR_Peers_SO.drop_duplicates()
```

```
In [53]: UR_Peers_SO['Total_PUBLS']=UR_Peers_SO.sum(axis=1)
```

```
In [60]: O=UR_Peers_SO.loc[:,[['institution_name','Total_PUBLS']]]
```

```
In [61]: P=UR_Peers_top1.loc[:,[['institution_name','top1_all']]]
```

```
In [62]: Q=UR_Peers_top10.loc[:,[['institution_name','top10_all']]]
```

```
In [63]: part1=O.join(P.set_index('institution_name'), on='institution_name')
```

```
In [64]: part2=part1.join(Q.set_index('institution_name'), on='institution_name')
```

```
In [65]: part2
```

```
Out[65]:
```

	institution_name	Total_PUBLS	top1_all	top10_all
0	Stanford University	577302	5095.0	30140.0
1	Johns Hopkins University	574103	3467.0	26670.0
2	University of Pennsylvania	566904	3589.0	23969.0
3	New York University	558900	2469.0	18246.0
5	Duke University	553164	2633.0	19183.0
6	University of Chicago	552365	2750.0	18983.0
7	Northwestern University	548490	2788.0	17567.0
9	Vanderbilt University	539419	1605.0	12180.0
10	Emory University	538845	1565.0	12107.0
11	Boston University	536886	1714.0	12080.0
12	Case Western Reserve University	530266	913.0	7746.0
13	University of Rochester	526467	740.0	6649.0
14	Tulane University	517260	370.0	3003.0

```
In [66]: part2['remaining_90%']=part2.Total_PUBLS-part2.top1_all-part2.top10_all
```

```
In [67]: part2
```

```
Out [67]:
```

	institution_name	Total_PUBLS	top1_all	top10_all	\
0	Stanford University	577302	5095.0	30140.0	
1	Johns Hopkins University	574103	3467.0	26670.0	
2	University of Pennsylvania	566904	3589.0	23969.0	
3	New York University	558900	2469.0	18246.0	
5	Duke University	553164	2633.0	19183.0	
6	University of Chicago	552365	2750.0	18983.0	
7	Northwestern University	548490	2788.0	17567.0	
9	Vanderbilt University	539419	1605.0	12180.0	
10	Emory University	538845	1565.0	12107.0	
11	Boston University	536886	1714.0	12080.0	
12	Case Western Reserve University	530266	913.0	7746.0	
13	University of Rochester	526467	740.0	6649.0	
14	Tulane University	517260	370.0	3003.0	

	remaining_90%
0	542067.0
1	543966.0
2	539346.0
3	538185.0
5	531348.0
6	530632.0
7	528135.0
9	525634.0
10	525173.0
11	523092.0
12	521607.0
13	519078.0
14	513887.0

```
In [70]: part2['Abbr']=['Stanford','JohnsHopkins','UofPenn','NYU','Duke','UofChicago','Northwes
```

```
In [71]: part2
```

```
Out [71]:
```

	institution_name	Total_PUBLS	top1_all	top10_all	\
0	Stanford University	577302	5095.0	30140.0	
1	Johns Hopkins University	574103	3467.0	26670.0	
2	University of Pennsylvania	566904	3589.0	23969.0	
3	New York University	558900	2469.0	18246.0	
5	Duke University	553164	2633.0	19183.0	
6	University of Chicago	552365	2750.0	18983.0	
7	Northwestern University	548490	2788.0	17567.0	
9	Vanderbilt University	539419	1605.0	12180.0	
10	Emory University	538845	1565.0	12107.0	
11	Boston University	536886	1714.0	12080.0	

12	Case Western Reserve University	530266	913.0	7746.0
13	University of Rochester	526467	740.0	6649.0
14	Tulane University	517260	370.0	3003.0

	remaining_90%	Abbr
0	542067.0	Stanford
1	543966.0	JohnsHopkins
2	539346.0	UofPenn
3	538185.0	NYU
5	531348.0	Duke
6	530632.0	UofChicago
7	528135.0	Northwestern
9	525634.0	Vanderbilt
10	525173.0	Emory
11	523092.0	Boston
12	521607.0	CWRU
13	519078.0	UofR
14	513887.0	Tulane

```
In [84]: A=part2[['Abbr','remaining_90%']]
```

```
In [95]: A.reset_index(inplace=True)
```

```
In [86]: B=part2[['Abbr','top10_all']]
```

```
In [99]: B
```

```
Out[99]:
```

	Abbr	top10_all
0	Stanford	30140.0
1	JohnsHopkins	26670.0
2	UofPenn	23969.0
3	NYU	18246.0
5	Duke	19183.0
6	UofChicago	18983.0
7	Northwestern	17567.0
9	Vanderbilt	12180.0
10	Emory	12107.0
11	Boston	12080.0
12	CWRU	7746.0
13	UofR	6649.0
14	Tulane	3003.0

```
In [87]: C=part2[['Abbr','top1_all']]
```

```
In [101]: len(C.Abbbr)
```

```
Out[101]: 13
```

```
In [172]: def show_values_on_bars(axes, h_v="v", space=0.8):
           def _show_on_single_plot(ax):
```

```

if h_v == "v":
    for p in ax.patches:
        _x = p.get_x() + p.get_width() / 2
        _y = p.get_y() + p.get_height()
        value = int(p.get_height())
        ax.text(_x, _y, value, ha="center", color='red')
elif h_v == "h":
    for p in ax.patches:
        _x = p.get_x() + p.get_width() + float(space)
        _y = p.get_y() + p.get_height()
        value = int(p.get_width())
        ax.text(_x, _y, value, ha="left", color='black')

if isinstance(axes, np.ndarray):
    for idx, ax in np.ndenumerate(axes):
        _show_on_single_plot(ax)
else:
    _show_on_single_plot(axes)

```

56 Compare to our USA peers, our top 1% and top 10% highly cited

57 publications is relatively low

```

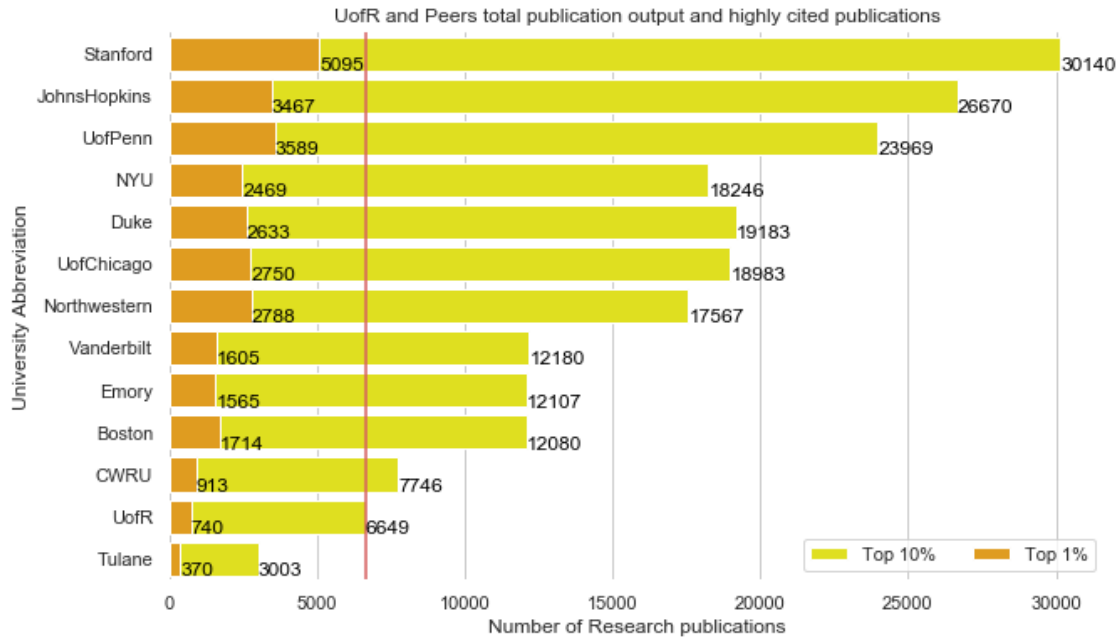
In [174]: # Plot the crashes where alcohol was involved
          # Plot the crashes where alcohol was involved
          sns.set(style="whitegrid")
          f, ax = plt.subplots(figsize=(10, 6))
          sns.set_color_codes()
          g=sns.barplot(x='top10_all',y='Abbr',data=B,
                        label="Top 10%", color="yellow")
          sns.set_color_codes("muted")
          g=sns.barplot(x='top1_all', y='Abbr', data=C,
                        label="Top 1%", color="orange")

          plt.axvline(6649, ls='-', color='r')

          ax.legend(ncol=2, loc="lower right", frameon=True)
          #plt.xticks(np.arange(0,3000, step=100))
          ax.set(xlabel='Number of Research publications', ylabel="University Abbreviation",
                  title="UofR and Peers total publication output and highly cited publications")
          sns.despine(left=True, bottom=True)

          show_values_on_bars(g, "h", 2)

```



58 Use all THE University ids to get Topic Cluster ids

```
In [175]: url='https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/{}?topicId={}'
```

```
In [176]: import requests
import json
import pandas as pd
import numpy as np
```

```
In [177]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PercPublsCited"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PercPublsCited
```

```
In [178]: ALL_Uids=pd.read_csv("THEUNI_CITEDPUBLS.csv")
```

```
In [182]: ALL_Uids=ALL_Uids.iloc[:, :3]
```

```
In [184]: USA_Uids=ALL_Uids[ALL_Uids.countryCode=='USA']
```

```
In [186]: USA_Uids.reset_index(inplace=True, drop=True)
```

```
In [189]: USA_Uids.head() # these are the USA university ids we'll use in API
len(USA_Uids) # 165 universities
```

```
Out[189]: 165
```

```

In [191]: inst_ids=USA_Uids

In [ ]: # get topic ids

In [ ]: # THE

In [216]: url='https://api.elsevier.com/analytics/scival/subjectArea/classificationType/THE?'

In [197]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data"

C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data

In [198]: import time
          from time import sleep

In [214]: time.sleep(1)

          subjectAreas_name=[]
          subjectAreas_id=[]
          subjectAreas_uri=[]
          classificationType=[]
          classificationName=[]

          resp = requests.get(url, headers={'Accept': 'application/json',
                                             'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
          parsed=json.dumps(resp.json(),
                             sort_keys=True,
                             indent=4, separators=(',', ': '))

          # print(parsed)
          result=json.loads(parsed)
          result["subjectAreas"][0]['children'][0]

Out[214]: {'classificationType': 'ASJC',
          'id': 1000,
          'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/1000?class',
                   '@ref': 'self',
                   '@type': 'application/json'},
          'name': 'Multidisciplinary',
          'uri': 'Class/ASJC/Code/1000'}

In [204]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\THE Code"

C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\THE Code

In [235]: subjectAreas_name=[]
          subjectAreas_id=[]
          subjectAreas_uri=[]

```



```

classificationType=[]
classificationName=[]

resp = requests.get(url, headers={'Accept': 'application/json',
                                   'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ' : '))
# print(parsed)
result=json.loads(parsed)
#if 'children' in result['subjectAreas'][0]:
#    if len(result['subjectAreas']) >=1:
#        if "name" in result["subjectAreas"][0]['children'][0]:
#            subjectAreas_name.append(result["subjectAreas"][0]['children'][0]["name"])
#subjectAreas_name
#result
result

```

```

Out[235]: {'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/classificationName/32',
                    '@ref': 'self',
                    '@type': 'application/json'},
            'subjectAreas': [{'classificationName': 'Times Higher Education',
                              'classificationType': 'THE',
                              'id': 32,
                              'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/32?classificationName=Times%20Higher%20Education',
                                        '@ref': 'self',
                                        '@type': 'application/json'},
                              'name': 'Life Sciences',
                              'uri': 'Class/THE/Code/32'},
                             {'classificationName': 'Times Higher Education',
                              'classificationType': 'THE',
                              'id': 33,
                              'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/33?classificationName=Times%20Higher%20Education',
                                        '@ref': 'self',
                                        '@type': 'application/json'},
                              'name': 'Arts and Humanities',
                              'uri': 'Class/THE/Code/33'},
                             {'classificationName': 'Times Higher Education',
                              'classificationType': 'THE',
                              'id': 34,
                              'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/34?classificationName=Times%20Higher%20Education',
                                        '@ref': 'self',
                                        '@type': 'application/json'},
                              'name': 'Clinical, pre-clinical and health',
                              'uri': 'Class/THE/Code/34'},
                             {'classificationName': 'Times Higher Education',
                              'classificationType': 'THE',
                              'id': 35,

```

```

'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/35?classs
  '@ref': 'self',
  '@type': 'application/json'},
'name': 'Business and Economics',
'uri': 'Class/THE/Code/35'},
{'classificationName': 'Times Higher Education',
'classificationType': 'THE',
'id': 36,
'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/36?classs
  '@ref': 'self',
  '@type': 'application/json'},
'name': 'Engineering and Technology',
'uri': 'Class/THE/Code/36'},
{'classificationName': 'Times Higher Education',
'classificationType': 'THE',
'id': 37,
'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/37?classs
  '@ref': 'self',
  '@type': 'application/json'},
'name': 'Physical Sciences',
'uri': 'Class/THE/Code/37'},
{'classificationName': 'Times Higher Education',
'classificationType': 'THE',
'id': 38,
'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/38?classs
  '@ref': 'self',
  '@type': 'application/json'},
'name': 'Computer Science',
'uri': 'Class/THE/Code/38'},
{'classificationName': 'Times Higher Education',
'classificationType': 'THE',
'id': 39,
'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/39?classs
  '@ref': 'self',
  '@type': 'application/json'},
'name': 'Psychology',
'uri': 'Class/THE/Code/39'},
{'classificationName': 'Times Higher Education',
'classificationType': 'THE',
'id': 40,
'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/40?classs
  '@ref': 'self',
  '@type': 'application/json'},
'name': 'Social Sciences',
'uri': 'Class/THE/Code/40'},
{'classificationName': 'Times Higher Education',
'classificationType': 'THE',
'id': 41,

```

```

        'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/41?classs
        '@ref': 'self',
        '@type': 'application/json'},
        'name': 'Education',
        'uri': 'Class/THE/Code/41'},
        {'classificationName': 'Times Higher Education',
        'classificationType': 'THE',
        'id': 42,
        'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/42?classs
        '@ref': 'self',
        '@type': 'application/json'},
        'name': 'Law',
        'uri': 'Class/THE/Code/42'}}}]

```

```

In [244]: subjectAreas_name=[]
          subjectAreas_id=[]
          subjectAreas_uri=[]
          classificationType=[]
          classificationName=[]

```

```

resp = requests.get(url, headers={'Accept': 'application/json',
                                  'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
parsed=json.dumps(resp.json(),
                   sort_keys=True,
                   indent=4, separators=(',', ' : '))
# print(parsed)
result=json.loads(parsed)
DF=pd.DataFrame.from_dict(result['subjectAreas'])
DF.to_csv("THE_Classification_Code.csv", index=False)

```

```

In [192]: # ASJC

```

```

In [259]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\ASJC Code"

```

```

C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\ASJC Code

```

```

In [247]: url='https://api.elsevier.com/analytics/scival/subjectArea/classificationType/ASJC?'

```

```

In [268]: classificationType=[]
          classification_id=[]
          link=[]
          name=[]
          uri=[]

```

```

resp = requests.get(url, headers={'Accept': 'application/json',
                                  'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
parsed=json.dumps(resp.json(),
                   sort_keys=True,

```

```

        indent=4, separators=(',', ' ': ))
#     print(parsed)
result=json.loads(parsed)
with open("ASJC.json", "w") as json_file:
    json.dump(resp.json(), json_file)

In [257]: resp = requests.get(url, headers={'Accept': 'application/json',
                                             'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ' ': ))
#     print(parsed)
    result=json.loads(parsed)
    result['subjectAreas'][0]['children'][0]

Out[257]: {'classificationType': 'ASJC',
           'id': 1000,
           'link': {'@href': 'https://api.elsevier.com/analytics/scival/subjectArea/1000?class=
           '@ref': 'self',
           '@type': 'application/json'},
           'name': 'Multidisciplinary',
           'uri': 'Class/ASJC/Code/1000'}

In [261]: # ASJC code result from earlier data

In [265]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\ASJC Code"
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\ASJC Code

In [271]: with open("ASJC.json") as output:
    data=json.load(output)

In [277]: name=[]
    Acode=[]
    uri=[]
    classificationType=[]

    for i in range(0,len(data['subjectAreas'])):
        name.append(data['subjectAreas'][i]['children'][0]['name'])
        Acode.append(data['subjectAreas'][i]['children'][0]['id'])
        uri.append(data['subjectAreas'][i]['children'][0]['uri'])
        classificationType.append(data['subjectAreas'][i]['children'][0]['classificationType'])
    DF=pd.DataFrame({'name':name,
                     'ASJC_Code': Acode,
                     'uri':uri,
                     'classificationType':classificationType})
    DF.to_csv("NEW_ASJC.csv", index=False)

In [339]: url='https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?top

```

```
In [281]: for item in inst_ids.institution_id[:2]: # test
          print(item)
```

508092

508021

```
In [349]: pwd
```

```
Out[349]: 'C:\\Users\\jchen148\\THE Rankings\\Report to Jane\\OK Files\\Output Data\\ASJC Code'
```

```
In [351]: with open("ASJC.json") as output:
          topic_id=json.load(output)
```

```
In [354]: ASJC_Code=pd.read_csv("NEW_ASJC.csv")
```

```
In [355]: ASJC_Code.head()
```

```
Out[355]:
```

	name	ASJC_Code	\
0	Multidisciplinary	1000	
1	General Agricultural and Biological Sciences	1100	
2	General Arts and Humanities	1200	
3	General Biochemistry,Genetics and Molecular Bi...	1300	
4	General Business,Management and Accounting	1400	

```
uri classificationType
0 Class/ASJC/Code/1000 ASJC
1 Class/ASJC/Code/1100 ASJC
2 Class/ASJC/Code/1200 ASJC
3 Class/ASJC/Code/1300 ASJC
4 Class/ASJC/Code/1400 ASJC
```

```
In [368]: topic_id=ASJC_Code['ASJC_Code']
          topic_id[:2]
```

```
Out[368]: 0    1000
          1    1100
          Name: ASJC_Code, dtype: int64
```

```
In [285]: cd "C:\\Users\\jchen148\\THE Rankings\\Report to Jane\\OK Files\\Output Data\\ASJC Code"
```

```
C:\\Users\\jchen148\\THE Rankings\\Report to Jane\\OK Files\\Output Data\\ASJC Code
```

```
In [328]: url='https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/{}?&metri'
```

```
In [369]: for item in inst_ids.institution_id[20:22]:
          print(item)
```

508358

508166

```
In [357]: url
```

```
Out[357]: 'https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/{}?topicId={}'
```

```
In [373]: inst_ids
```

```
Out[373]:
```

	country	countryCode	institution_id
0	United States	USA	508092
1	United States	USA	508021
2	United States	USA	508219
3	United States	USA	508111
4	United States	USA	508191
5	United States	USA	508076
6	United States	USA	508383
7	United States	USA	508270
8	United States	USA	508331
9	United States	USA	714942
10	United States	USA	706736
11	United States	USA	508529
12	United States	USA	714855
13	United States	USA	714951
14	United States	USA	508094
15	United States	USA	508726
16	United States	USA	508046
17	United States	USA	508047
18	United States	USA	508053
19	United States	USA	508175
20	United States	USA	508358
21	United States	USA	508166
22	United States	USA	508359
23	United States	USA	508018
24	United States	USA	508013
25	United States	USA	508305
26	United States	USA	702176
27	United States	USA	508122
28	United States	USA	508256
29	United States	USA	508363
..
135	United States	USA	508095
136	United States	USA	508096
137	United States	USA	508101
138	United States	USA	508126
139	United States	USA	508316
140	United States	USA	508322
141	United States	USA	508182
142	United States	USA	508189
143	United States	USA	508339
144	United States	USA	508220

145	United States	USA	508344
146	United States	USA	508806
147	United States	USA	508233
148	United States	USA	508353
149	United States	USA	508376
150	United States	USA	508373
151	United States	USA	508380
152	United States	USA	508062
153	United States	USA	508063
154	United States	USA	508108
155	United States	USA	508110
156	United States	USA	508125
157	United States	USA	508172
158	United States	USA	508195
159	United States	USA	508347
160	United States	USA	704980
161	United States	USA	508026
162	United States	USA	508335
163	United States	USA	508335
164	United States	USA	508059

[165 rows x 3 columns]

```
In [374]: UR_ins_id='508335'
```

```
In [378]: url='https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?top'
```

```
In [372]: for item in inst_ids.institution_id[:2]:
           for tid in topic_id[:2]:
               print(url.format(item, tid))
```

```
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508092?topicId=1000&metr
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508092?topicId=1100&metr
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508021?topicId=1000&metr
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508021?topicId=1100&metr
```

```
In [380]: for tid in topic_id[:10]:
           print(tid)
```

```
1000
1100
1200
1300
1400
1500
1600
1700
1800
```

1900

```
In [382]: for tid in topic_id[:10]:  
          print(url.format(tid))
```

```
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?topicId=1000&metr  
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?topicId=1100&metr  
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?topicId=1200&metr  
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?topicId=1300&metr  
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?topicId=1400&metr  
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?topicId=1500&metr  
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?topicId=1600&metr  
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?topicId=1700&metr  
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?topicId=1800&metr  
https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/508335?topicId=1900&metr
```

```
In [385]: pwd
```

```
Out[385]: 'C:\\Users\\jchen148\\THE Rankings\\Report to Jane\\OK Files\\Output Data\\ASJC Code
```

```
In [389]: for i in topic_id[:20]:  
          print(i)
```

```
1000  
1100  
1200  
1300  
1400  
1500  
1600  
1700  
1800  
1900  
2000  
2100  
2200  
2300  
2400  
2500  
2600  
2700  
2800  
2900
```

```
In [427]: source=[]  
          endyear=[]
```



```

startyear=[]
name=[]

for tid in topic_id[:2]:
    resp = requests.get(url.format(tid), headers={'Accept': 'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ' : '))
    # print(parsed)
    result=json.loads(parsed)
result

```

```

Out[427]: {'dataSource': {'lastUpdated': '2020-01-01',
                        'metricEndYear': 2018,
                        'metricStartYear': 2014,
                        'sourceName': 'Scopus'},
          'link': {'@href': 'https://api.elsevier.com/analytics/scival/topic/metrics/institut',
                  '@ref': 'self',
                  '@type': 'application/json'},
          'results': [{'metrics': [{'metricType': 'ScholarlyOutput', 'value': 1}],
                    'topic': {'id': 1100,
                              'link': {'@href': 'https://api.elsevier.com/analytics/scival/topic/1100',
                                      '@ref': 'self',
                                      '@type': 'application/json'},
                              'name': 'Time; Time Perception; Temporal bisection',
                              'overallScholarlyOutput': 868,
                              'prominencePercentile': 94.98377,
                              'scholarlyOutput': 1,
                              'uri': 'Topic/1100'}}]}

```

```

In [428]: source=[]
endyear=[]
startyear=[]
name=[]

for tid in topic_id[:2]:
    resp = requests.get(url.format(tid), headers={'Accept': 'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ' : '))
    # print(parsed)
    result=json.loads(parsed)
    #result
    source.append(result['dataSource'])
    endyear.append(result['dataSource']['metricEndYear'])
    startyear.append(result['dataSource']['metricStartYear'])

```

```

        name.append(result['dataSource']['sourceName'])

Data_title=pd.DataFrame({'dataSource': source,
                        'metricEndYear':endyear,
                        'metricStartYear':startyear,
                        'sourceName':name})

Data_title.to_csv("ASJC_Data_Title.csv", index=False)

In [396]: td=[]
metricType=[]
metricvalue=[]
link=[]
name=[]
ACode=[]
uri=[]
prominencePercentile=[]
scholarlyOutput=[]
overallScholarlyOutput=[]

for tid in topic_id[20:]:
    resp = requests.get(url.format(tid), headers={'Accept':'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ': '))
    # print(parsed)
    result=json.loads(parsed)
    with open("THE_UNI_ASJC_after20.json", "w") as json_file:
        json.dump(resp.json(), json_file)
    #result
    td.append(tid)
    if result['results'] is None:
        metricType.append('')
        metricvalue.append('')
        link.append('')
        name.append('')
        ACode.append('')
        uri.append('')
        prominencePercentile.append('')
        scholarlyOutput.append('')
        overallScholarlyOutput.append('')
    else:
        if len(result['results']) >=1:
            if "metrics" in result['results'][0]:
                metricType.append(result['results'][0]["metrics"][0]["metricType"])
                metricvalue.append(result['results'][0]["metrics"][0]["value"])

```

```

        if "topic" in result['results'][0]:
            link.append(result['results'][0]["topic"]["link"])
            name.append(result['results'][0]["topic"]["name"])
            Acode.append(result['results'][0]["topic"]["id"])
            uri.append(result['results'][0]["topic"]["uri"])
            prominencePercentile.append(result['results'][0]["topic"]["prominencePercentile"])
            scholarlyOutput.append(result['results'][0]["topic"]["scholarlyOutput"])
            overallScholarlyOutput.append(result['results'][0]["topic"]["overallScholarlyOutput"])

s1=pd.Series(td, name='Topic_ID')
s2=pd.Series(metricType, name='metricType')
s3=pd.Series(metricvalue, name='metricvalue')
s4=pd.Series(link, name='link')
s5=pd.Series(name, name='name')
s6=pd.Series(Acode, name='ACode')
s7=pd.Series(uri, name='uri')
s8=pd.Series(prominencePercentile, name='prominencePercentile')
s9=pd.Series(scholarlyOutput, name='scholarlyOutput')
s10=pd.Series(overallScholarlyOutput, name='overallScholarlyOutput')

ASJC_20=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10], axis=1)

ASJC_20.to_csv("ASJC_after20.csv", index=False)

#DF=pd.DataFrame({'Topic_ID':td,
#                  'metricType':metricType,
#                  'metricvalue':metricvalue,
#                  'link':link,
#                  'name':name,
#                  'ACode':ACode,
#                  'uri':uri,
#                  'prominencePercentile':prominencePercentile,
#                  'scholarlyOutput':scholarlyOutput,
#                  'overallScholarlyOutput':overallScholarlyOutput
#                  })

#DF.to_csv("ASJC_TID20.csv", index=False)

#with open("THE_UNI_ASJC_after20.json", "w") as json_file:
#    json.dump(resp.json(), json_file)

```

In [398]: UR_ASJC_1=pd.read_csv("ASJC_20.csv")

In [399]: UR_ASJC_1.head()

```

Out[399]:   Topic_ID      metricType  metricvalue  \
0       1000  ScholarlyOutput           1.0
1       1100  ScholarlyOutput           1.0

```

2	1200	ScholarlyOutput	2.0
3	1300	ScholarlyOutput	3.0
4	1400	ScholarlyOutput	1.0

	link \
0	{'@href': 'https://api.elsevier.com/analytics/...
1	{'@href': 'https://api.elsevier.com/analytics/...
2	{'@href': 'https://api.elsevier.com/analytics/...
3	{'@href': 'https://api.elsevier.com/analytics/...
4	{'@href': 'https://api.elsevier.com/analytics/...

	name	ACode	uri \
0	Time; Time Perception; Temporal bisection	1100.0	Topic/1100
1	Intubation; Laryngoscopes; Video laryngoscopy	1200.0	Topic/1200
2	Speech enhancement; Speech; Binary mask	1300.0	Topic/1300
3	Acromegaly; Patients; Acromegalic patients	1600.0	Topic/1600
4	Glaucoma, Angle-Closure; Anterior Chamber; Gla...	2100.0	Topic/2100

	prominencePercentile	scholarlyOutput	overallScholarlyOutput
0	94.983770	1.0	868.0
1	94.222880	1.0	1361.0
2	95.479546	2.0	1334.0
3	93.386840	3.0	690.0
4	89.228570	1.0	819.0

In [400]: UR_ASJC_2=pd.read_csv("ASJC_after20.csv")

In [401]: UR_ASJC_2.head()

Out[401]:

	Topic_ID	metricType	metricvalue \
0	3000.0	ScholarlyOutput	1.0
1	3100.0	ScholarlyOutput	2.0
2	3200.0	ScholarlyOutput	2.0
3	3300.0	NaN	NaN
4	3400.0	NaN	NaN

	link \
0	{'@href': 'https://api.elsevier.com/analytics/...
1	{'@href': 'https://api.elsevier.com/analytics/...
2	{'@href': 'https://api.elsevier.com/analytics/...
3	NaN
4	NaN

	name	ACode	uri \
0	Staphylococcus aureus; Enterotoxins; Staphyloc...	1100	Topic/3100
1	Activities of Daily Living; Health; Healthy life	1200	Topic/3300
2	Stars; Distance; RRab stars	1300	Topic/3500
3	NaN	1600	NaN

4		NaN	2100	NaN
---	--	-----	------	-----

	prominencePercentile	scholarlyOutput	overallScholarlyOutput
0	90.867250	1.0	582.0
1	94.985855	2.0	725.0
2	95.439890	2.0	520.0
3	NaN	NaN	NaN
4	NaN	NaN	NaN

```
In [403]: del UR_ASJC_1['Topic_ID']
```

```
In [405]: del UR_ASJC_1['ACode']
```

```
In [406]: del UR_ASJC_2['Topic_ID']
```

```
In [407]: del UR_ASJC_2['ACode']
```

```
In [409]: UR_ASJC_1=UR_ASJC_1.dropna()
```

```
In [410]: UR_ASJC_2=UR_ASJC_2.dropna()
```

```
In [412]: UR_ASJC=pd.concat([UR_ASJC_1, UR_ASJC_2])
```

```
In [413]: UR_ASJC.head()
```

```
Out[413]:
```

	metricType	metricvalue	\
0	ScholarlyOutput	1.0	
1	ScholarlyOutput	1.0	
2	ScholarlyOutput	2.0	
3	ScholarlyOutput	3.0	
4	ScholarlyOutput	1.0	

	link	\
0	{'@href': 'https://api.elsevier.com/analytics/...	
1	{'@href': 'https://api.elsevier.com/analytics/...	
2	{'@href': 'https://api.elsevier.com/analytics/...	
3	{'@href': 'https://api.elsevier.com/analytics/...	
4	{'@href': 'https://api.elsevier.com/analytics/...	

	name	uri	\
0	Time; Time Perception; Temporal bisection	Topic/1100	
1	Intubation; Laryngoscopes; Video laryngoscopy	Topic/1200	
2	Speech enhancement; Speech; Binary mask	Topic/1300	
3	Acromegaly; Patients; Acromegalic patients	Topic/1600	
4	Glaucoma, Angle-Closure; Anterior Chamber; Gla...	Topic/2100	

	prominencePercentile	scholarlyOutput	overallScholarlyOutput
0	94.983770	1.0	868.0
1	94.222880	1.0	1361.0
2	95.479546	2.0	1334.0
3	93.386840	3.0	690.0
4	89.228570	1.0	819.0

```
In [419]: chuck=[]
         for line in UR_ASJC.uri:
             chuck.append(str(line).split('/')[1])
         UR_ASJC['ASJC_Code']=chuck
```

```
In [420]: UR_ASJC.head()
```

```
Out[420]:
```

	metricType	metricvalue	\
0	ScholarlyOutput	1.0	
1	ScholarlyOutput	1.0	
2	ScholarlyOutput	2.0	
3	ScholarlyOutput	3.0	
4	ScholarlyOutput	1.0	

	link	\
0	{'@href': 'https://api.elsevier.com/analytics/...	
1	{'@href': 'https://api.elsevier.com/analytics/...	
2	{'@href': 'https://api.elsevier.com/analytics/...	
3	{'@href': 'https://api.elsevier.com/analytics/...	
4	{'@href': 'https://api.elsevier.com/analytics/...	

	name	uri	\
0	Time; Time Perception; Temporal bisection	Topic/1100	
1	Intubation; Laryngoscopes; Video laryngoscopy	Topic/1200	
2	Speech enhancement; Speech; Binary mask	Topic/1300	
3	Acromegaly; Patients; Acromegalic patients	Topic/1600	
4	Glaucoma, Angle-Closure; Anterior Chamber; Gla...	Topic/2100	

	prominencePercentile	scholarlyOutput	overallScholarlyOutput	ASJC_Code
0	94.983770	1.0	868.0	1100
1	94.222880	1.0	1361.0	1200
2	95.479546	2.0	1334.0	1300
3	93.386840	3.0	690.0	1600
4	89.228570	1.0	819.0	2100

```
In [421]: UR_ASJC=UR_ASJC.loc[:,['ASJC_Code','metricType','name','link','uri','prominencePerce
```

```
In [422]: UR_ASJC
```

```
Out[422]:
```

	ASJC_Code	metricType	\
0	1100	ScholarlyOutput	
1	1200	ScholarlyOutput	
2	1300	ScholarlyOutput	
3	1600	ScholarlyOutput	
4	2100	ScholarlyOutput	
5	2400	ScholarlyOutput	
6	2700	ScholarlyOutput	
0	3100	ScholarlyOutput	
1	3300	ScholarlyOutput	

```
2      3500  ScholarlyOutput
```

	name \	link	uri \
0	Time; Time Perception; Temporal bisection		
1	Intubation; Laryngoscopes; Video laryngoscopy		
2	Speech enhancement; Speech; Binary mask		
3	Acromegaly; Patients; Acromegalic patients		
4	Glaucoma, Angle-Closure; Anterior Chamber; Gla...		
5	Pharmaceutical Preparations; Electrocardiograph...		
6	Maximum likelihood estimation; Maximum likelih...		
0	Staphylococcus aureus; Enterotoxins; Staphyloc...		
1	Activities of Daily Living; Health; Healthy life		
2	Stars; Distance; RRab stars		
0	{ '@href': 'https://api.elsevier.com/analytics/...	Topic/1100	
1	{ '@href': 'https://api.elsevier.com/analytics/...	Topic/1200	
2	{ '@href': 'https://api.elsevier.com/analytics/...	Topic/1300	
3	{ '@href': 'https://api.elsevier.com/analytics/...	Topic/1600	
4	{ '@href': 'https://api.elsevier.com/analytics/...	Topic/2100	
5	{ '@href': 'https://api.elsevier.com/analytics/...	Topic/2400	
6	{ '@href': 'https://api.elsevier.com/analytics/...	Topic/2700	
0	{ '@href': 'https://api.elsevier.com/analytics/...	Topic/3100	
1	{ '@href': 'https://api.elsevier.com/analytics/...	Topic/3300	
2	{ '@href': 'https://api.elsevier.com/analytics/...	Topic/3500	
	prominencePercentile	scholarlyOutput	overallScholarlyOutput
0	94.983770	1.0	868.0
1	94.222880	1.0	1361.0
2	95.479546	2.0	1334.0
3	93.386840	3.0	690.0
4	89.228570	1.0	819.0
5	93.968210	9.0	682.0
6	89.025040	1.0	878.0
0	90.867250	1.0	582.0
1	94.985855	2.0	725.0
2	95.439890	2.0	520.0

```
In [423]: UR_ASJC.to_csv("UR_ASJC_0110.csv", index=False)
```

```
In [1024]: import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="whitegrid")
sns.set_style("ticks", {"xtick.major.size": 10, "ytick.major.size": 8})

# Initialize the matplotlib figure
f, ax = plt.subplots(figsize=(6, 15))
```

```

# Load the example car crash dataset
#crashes = sns.load_dataset("car_crashes").sort_values("total", ascending=False)

# Plot the total crashes
sns.set_color_codes("pastel")
g=sns.barplot(data=A,
              label="Total", color="b")

# Plot the crashes where alcohol was involved
sns.set_color_codes("dark")
g=sns.barplot(data=B,
              label="Top 10%", color="b")

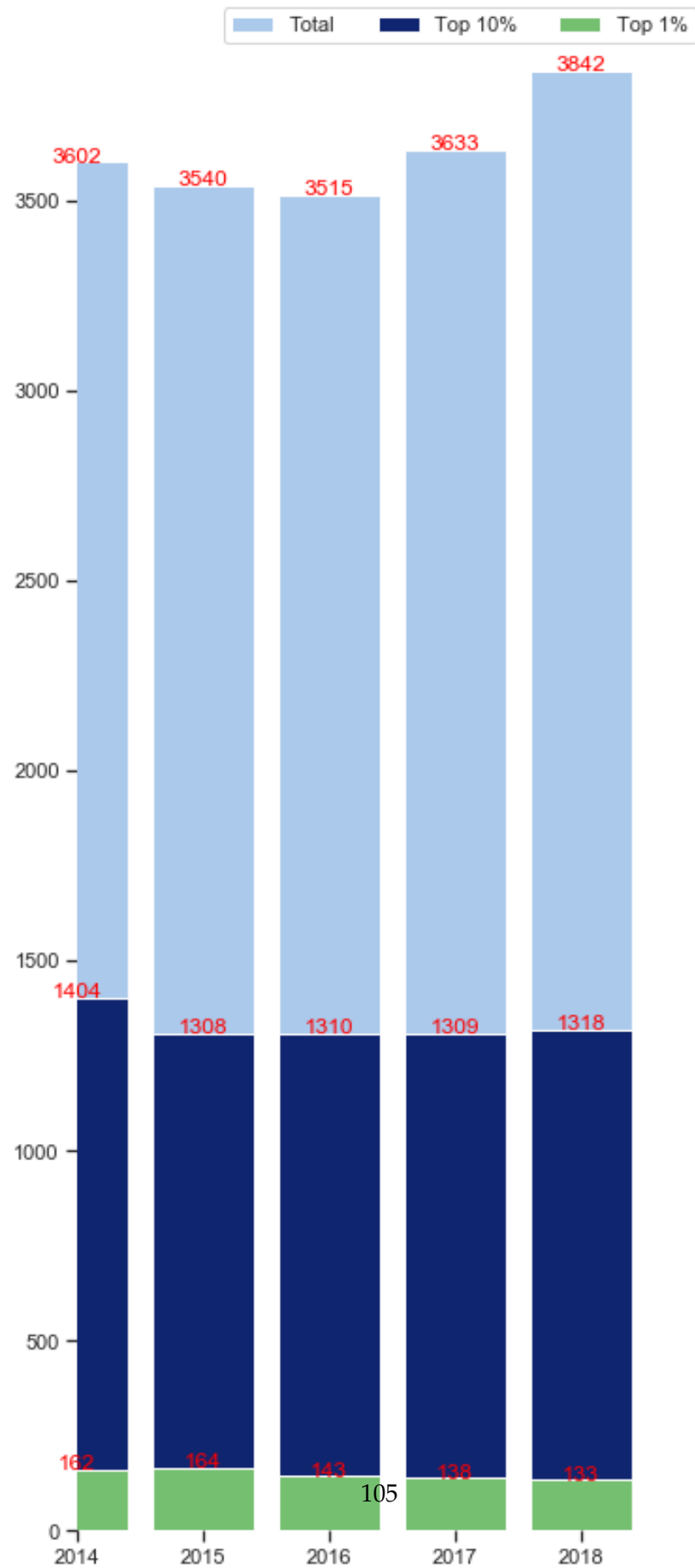
# Plot the crashes where alcohol was involved
sns.set_color_codes("muted")
g=sns.barplot(data=C,
              label="Top 1%", color="g")

show_values_on_bars(g, "v", 0.8)

# Add a legend and informative axis label
plt.yticks(np.arange(0, 4000, step=500))
plt.xticks(np.arange(5), ('2014', '2015', '2016', '2017', '2018'))
ax.legend(ncol=3, loc="upper right", frameon=True)
ax.set(xlim=(0,5), ylabel="",
       title="U of R publication output: total, top 1 % and top 10 % highly cited p
sns.despine(left=True, bottom=True)

```


U of R publication output: total, top 1 % and top 10 % highly cited publs



```

In [329]: for item in inst_ids.institution_id[20:22]:
#         for tid in topic_id[20:100]:
            resp = requests.get(url.format(item), headers={'Accept': 'application/json',
                                                            'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
            parsed=json.dumps(resp.json(),
                               sort_keys=True,
                               indent=4, separators=(',', ' : '))
#         print(parsed)
            result=json.loads(parsed)
result

Out[329]: {'dataSource': {'lastUpdated': '2020-01-01',
                          'metricEndYear': 2018,
                          'metricStartYear': 2014,
                          'sourceName': 'Scopus'},
           'link': {'@href': 'https://api.elsevier.com/analytics/scival/topic/metrics/institut',
                    '@ref': 'self',
                    '@type': 'application/json'},
           'results': [{'metrics': [{'metricType': 'ScholarlyOutput', 'value': 8}],
                        'topic': {'id': 0,
                                  'link': {'@href': 'https://api.elsevier.com/analytics/scival/topic/0',
                                             '@ref': 'self',
                                             '@type': 'application/json'},
                                  'name': 'Solar cells; Fullerenes; Organic photovoltaics',
                                  'overallScholarlyOutput': 9500,
                                  'prominencePercentile': 99.98956,
                                  'scholarlyOutput': 8,
                                  'uri': 'Topic/0'}}]}

In [298]: time.sleep(2)
for item in inst_ids.institution_id[20:100]:
    for tid in topic_id[20:100]:
        resp = requests.get(url.format(item, tid), headers={'Accept': 'application/json',
                                                            'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
        parsed=json.dumps(resp.json(),
                           sort_keys=True,
                           indent=4, separators=(',', ' : '))
#         print(parsed)
        result=json.loads(parsed)
    with open("THE_UNI_Versus_ASJC_3.json", 'w') as json_file:
        json.dump(resp.json(), json_file)

```

TypeError

Traceback (most recent call last)

```

C:\ProgramData\Anaconda3\lib\site-packages\urllib3\connectionpool.py in _make_request(
376         try: # Python 2.7, use buffering of HTTP responses
--> 377             httplib_response = conn.getresponse(buffering=True)
378         except TypeError: # Python 3

```

TypeError: getresponse() got an unexpected keyword argument 'buffering'

During handling of the above exception, another exception occurred:

KeyboardInterrupt

Traceback (most recent call last)

```

<ipython-input-298-31d8c4a13cf3> in <module>
      3     for tid in topic_id[20:100]:
      4         resp = requests.get(url.format(item, tid), headers={'Accept':'application/
----> 5                                 'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
      6         parsed=json.dumps(resp.json(),
      7                             sort_keys=True,

```

```

C:\ProgramData\Anaconda3\lib\site-packages\requests\api.py in get(url, params, **kwargs)
    73
    74     kwargs.setdefault('allow_redirects', True)
--> 75     return request('get', url, params=params, **kwargs)
    76
    77

```

```

C:\ProgramData\Anaconda3\lib\site-packages\requests\api.py in request(method, url, **kwargs)
    58     # cases, and look like a memory leak in others.
    59     with sessions.Session() as session:
--> 60         return session.request(method=method, url=url, **kwargs)
    61
    62

```

```

C:\ProgramData\Anaconda3\lib\site-packages\requests\sessions.py in request(self, method, url, **kwargs)
    531     }
    532     send_kwargs.update(settings)
--> 533     resp = self.send(prepare_request(method, url, **kwargs), **send_kwargs)
    534
    535     return resp

```

```

C:\ProgramData\Anaconda3\lib\site-packages\requests\sessions.py in send(self, request, **kwargs)

```

```

644
645         # Send the request
--> 646         r = adapter.send(request, **kwargs)
647
648         # Total elapsed time of the request (approximately)

C:\ProgramData\Anaconda3\lib\site-packages\requests\adapters.py in send(self, request,
447             decode_content=False,
448             retries=self.max_retries,
--> 449             timeout=timeout
450         )
451

C:\ProgramData\Anaconda3\lib\site-packages\urllib3\connectionpool.py in urlopen(self, r,
598             timeout=timeout_obj,
599             body=body, headers=headers,
--> 600             chunked=chunked)
601
602         # If we're going to release the connection in ``finally``, then

C:\ProgramData\Anaconda3\lib\site-packages\urllib3\connectionpool.py in _make_request(self,
378         except TypeError: # Python 3
379             try:
--> 380                 httplib_response = conn.getresponse()
381             except Exception as e:
382                 # Remove the TypeError from the exception chain in Python 3;

C:\ProgramData\Anaconda3\lib\http\client.py in getresponse(self)
1319         try:
1320             try:
-> 1321                 response.begin()
1322             except ConnectionError:
1323                 self.close()

C:\ProgramData\Anaconda3\lib\http\client.py in begin(self)
294         # read until we get a non-100 response
295         while True:
--> 296             version, status, reason = self._read_status()
297             if status != CONTINUE:
298                 break

C:\ProgramData\Anaconda3\lib\http\client.py in _read_status(self)

```

```

255
256     def _read_status(self):
--> 257         line = str(self.fp.readline(_MAXLINE + 1), "iso-8859-1")
258         if len(line) > _MAXLINE:
259             raise LineTooLong("status line")

```

```

C:\ProgramData\Anaconda3\lib\socket.py in readinto(self, b)
587         while True:
588             try:
--> 589                 return self._sock.recv_into(b)
590             except timeout:
591                 self._timeout_occurred = True

```

```

C:\ProgramData\Anaconda3\lib\site-packages\urllib3\contrib\pyopenssl.py in recv_into(s
295     def recv_into(self, *args, **kwargs):
296         try:
--> 297             return self.connection.recv_into(*args, **kwargs)
298         except OpenSSL.SSL.SysCallError as e:
299             if self.suppress_ragged_eofs and e.args == (-1, 'Unexpected EOF'):

```

```

C:\ProgramData\Anaconda3\lib\site-packages\OpenSSL\SSL.py in recv_into(self, buffer, n
1819         result = _lib.SSL_peek(self._ssl, buf, nbytes)
1820     else:
-> 1821         result = _lib.SSL_read(self._ssl, buf, nbytes)
1822         self._raise_ssl_error(self._ssl, result)
1823

```

KeyboardInterrupt:

```

In [303]: with open("THE_UNI_Versus_ASJC.json") as output:
          data=json.load(output)

```

```

In [330]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\ASJC Code"

```

```

C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\ASJC Code

```

```

In [341]: url

```

```

Out[341]: 'https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/{}?topicId={}'

```

```

In [347]:

```

```

Out[347]: []

```

```

In [346]: for item in inst_ids.institution_id[:5]:
           for tid in topic_id:

In [336]: inst_id=[]
           topic_id=[]
           metrics=[]
           metrics_value=[]
           link=[]
           name=[]
           Acode=[]
           uri=[]
           prominencePercentile=[]
           scholarlyOutput=[]
           overallScholarlyOutput=[]

           for item in inst_ids.institution_id[:5]:
               for tid in topic_id[:5]:
                   resp = requests.get(url.format(item, tid), headers={'Accept': 'application/json',
                                                                           'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
                   parsed=json.dumps(resp.json(),
                                       sort_keys=True,
                                       indent=4, separators=(',', ': '))
           #     print(parsed)
                   data=json.loads(parsed)
           data

Out[336]: {'link': {'@ref': 'self',
                    '@href': 'https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/50',
                    '@type': 'application/json'},
           'dataSource': {'sourceName': 'Scopus',
                          'lastUpdated': '2020-01-01',
                          'metricStartYear': 2014,
                          'metricEndYear': 2018},
           'results': [{'metrics': [{'metricType': 'ScholarlyOutput', 'value': 2}],
                        'topic': {'link': {'@ref': 'self',
                                             '@href': 'https://api.elsevier.com/analytics/scival/topic/1100',
                                             '@type': 'application/json'},
                                   'name': 'Time; Time Perception; Temporal bisection',
                                   'id': 1100,
                                   'uri': 'Topic/1100',
                                   'prominencePercentile': 94.98377,
                                   'scholarlyOutput': 2,
                                   'overallScholarlyOutput': 868}}]}

In [332]: inst_id=[]
           topic_id=[]
           metrics=[]
           metrics_value=[]

```

```

link=[]
name=[]
Acode=[]
uri=[]
prominencePercentile=[]
scholarlyOutput=[]
overallScholarlyOutput=[]

for item in inst_ids.institution_id[:5]:
    for tid in topic_id[:5]:
        resp = requests.get(url.format(item, tid), headers={'Accept':'application/json',
                                                             'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
        parsed=json.dumps(resp.json(),
                           sort_keys=True,
                           indent=4, separators=(',', ' : '))
#     print(parsed)
        data=json.loads(parsed)
        inst_id.append(item)
        topic_id.append(tid)
        link.append(data['link'])
        if len(data['results']) >=1:
            if 'metrics' in data['results'][0]:
                if len(data['results'][0]['metrics']) >=1:
                    metrics.append(data['results'][0]['metrics'][0]['metricType'])
                    metrics_value.append(data['results'][0]['metrics'][0]['value'])
            if 'topic' in data['results'][0]:
                if len(data['results'][0]['topic']) >=1:
                    name.append(data['results'][0]['topic']['name'])
                    Acode.append(data['results'][0]['topic']['id'])
                    uri.append(data['results'][0]['topic']['uri'])
                    prominencePercentile.append(data['results'][0]['topic']['prominencePercentile'])
                    scholarlyOutput.append(data['results'][0]['topic']['scholarlyOutput'])
                    overallScholarlyOutput.append(data['results'][0]['topic']['overallScholarlyOutput'])

s1=pd.Series(inst_id, name="inst_id")
s2=pd.Series(topic_id, name="topic_id")
s3=pd.Series(metrics, name="metrics")
s4=pd.Series(metrics_value, name="metrics_value")
s5=pd.Series(link, name="link")
s6=pd.Series(name, name="name")
s7=pd.Series(Acode, name="Acode")
s8=pd.Series(uri, name="uri")
s9=pd.Series(prominencePercentile, name="prominencePercentile")
s10=pd.Series(scholarlyOutput, name="scholarlyOutput")
s11=pd.Series(overallScholarlyOutput, name="overallScholarlyOutput")

DF=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11], axis=1)
DF.to_csv("TEST.csv", index=False)

```

```
#with open("THE_UNI_Versus_ASJC_Test.json", 'w') as json_file:
#    json.dump(resp.json(), json_file)
```

```
In [325]: inst_id=[]
         topic_id=[]
         metrics=[]
         metrics_value=[]
         link=[]
         name=[]
         Acode=[]
         uri=[]
         prominencePercentile=[]
         scholarlyOutput=[]
         overallScholarlyOutput=[]

         data
```

```
Out[325]: {'link': {'@ref': 'self',
 '@href': 'https://api.elsevier.com/analytics/scival/topic/metrics/institutionId/50',
 '@type': 'application/json'},
 'dataSource': {'sourceName': 'Scopus',
 'lastUpdated': '2020-01-01',
 'metricStartYear': 2014,
 'metricEndYear': 2018},
 'results': [{'metrics': [{'metricType': 'ScholarlyOutput', 'value': 2}],
 'topic': {'link': {'@ref': 'self',
 '@href': 'https://api.elsevier.com/analytics/scival/topic/1100',
 '@type': 'application/json'},
 'name': 'Time; Time Perception; Temporal bisection',
 'id': 1100,
 'uri': 'Topic/1100',
 'prominencePercentile': 94.98377,
 'scholarlyOutput': 2,
 'overallScholarlyOutput': 868}}]}
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

59 Comparatory analysis: research performance profile

```
In [239]: UR_Peer_FWCI.institution_name
```



```
Out [239]: 8          Stanford University
           3          Emory University
           6      Johns Hopkins University
           2          Duke University
           4      Northwestern University
          11      University of Pennsylvania
           0          Boston University
           7          New York University
          10          University of Chicago
           1      Case Western Reserve University
           5          Vanderbilt University
          12          University of Rochester
           9          Tulane University
           Name: institution_name, dtype: object
```

```
In [240]: so_data_USA.head()
```

```
Out [240]:
```

	institution_name	2014	2015	2016	\
0	Jet Propulsion Laboratory, California Institut...	1913	1800	2181	
1	California Institute of Technology	4518	4347	4553	
2	Stanford University	12801	13656	13664	
3	Massachusetts Institute of Technology	9645	9957	10023	
4	Princeton University	4335	4629	4544	

	2017	2018	Total
0	2052	2200	10146
1	4567	4884	22869
2	14233	14729	69083
3	10191	10458	50274
4	4635	4891	23034

```
In [800]: chuck=[]
```

```
for name in UR_Peer_FWCI.institution_name:
    chuck.append(so_data_USA[so_data_USA.institution_name==name])
```

```
In [241]: Ttl_publs_output=pd.concat(chuck, ignore_index=True)
```

```
In [243]: Ttl_publs_output.head()
```

```
Out [243]:
```

	country	countryCode	institution_id	\
0	United States	USA	508013	
1	United States	USA	508032	
2	United States	USA	508053	
3	United States	USA	508059	
4	United States	USA	508175	

	link	\
0	{'@href': 'https://api.elsevier.com/analytics/...	

```

1 {'@href': 'https://api.elsevier.com/analytics/...
2 {'@href': 'https://api.elsevier.com/analytics/...
3 {'@href': 'https://api.elsevier.com/analytics/...
4 {'@href': 'https://api.elsevier.com/analytics/...

```

	institution_name	metricType	2014 \
0	Boston University	FieldWeightedCitationImpact	2.102325
1	Case Western Reserve University	FieldWeightedCitationImpact	1.785904
2	Duke University	FieldWeightedCitationImpact	2.060966
3	Emory University	FieldWeightedCitationImpact	1.999690
4	Northwestern University	FieldWeightedCitationImpact	1.814437

	2015	2016	2017	2018
0	2.017788	2.082099	1.699757	1.916557
1	1.952037	1.955858	1.773011	1.861512
2	2.188656	1.971610	1.903780	1.848280
3	2.186228	2.209265	1.948478	1.967104
4	2.036129	2.104611	2.028618	1.968102

```
In [244]: Ttl_publs_output['Total']=Ttl_publs_output.sum(axis=1)
```

```
In [245]: A=Ttl_publs_output[['institution_name','Total']]
```

```
In [246]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PercPublsCited"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PercPublsCited
```

```
In [247]: ALL_PP=pd.read_csv("THEUNI_CITEDPUBLS.csv")
```

```
In [248]: ALL_PP.head()
```

```
Out[248]:
```

	country	countryCode	institution_id \
0	United Kingdom	GBR	315091
1	United States	USA	508092
2	United States	USA	508021
3	United Kingdom	GBR	315068
4	United States	USA	508219

	link \
0	{'@href': 'https://api.elsevier.com/analytics/...
1	{'@href': 'https://api.elsevier.com/analytics/...
2	{'@href': 'https://api.elsevier.com/analytics/...
3	{'@href': 'https://api.elsevier.com/analytics/...
4	{'@href': 'https://api.elsevier.com/analytics/...

	institution_name	metricType \
0	University of Oxford	CitedPublications
1	Jet Propulsion Laboratory, California Institut...	CitedPublications

```

2          California Institute of Technology  CitedPublications
3          University of Cambridge  CitedPublications
4          Stanford University  CitedPublications

```

	2014	2015	2016	2017	2018	percent2014	percent2015	\
0	10893.0	11679.0	11798.0	11474.0	10570.0	86.555420	85.53537	
1	1514.0	1451.0	1722.0	1588.0	1406.0	79.142710	80.61111	
2	3879.0	3770.0	3914.0	3779.0	3487.0	85.856575	86.72648	
3	9116.0	9238.0	9558.0	9125.0	8418.0	88.060280	86.17537	
4	11156.0	11846.0	11699.0	11642.0	10731.0	87.149445	86.74575	

	percent2016	percent2017	percent2018
0	84.115210	78.56213	70.69761
1	78.954605	77.38792	63.90909
2	85.965300	82.74578	71.39640
3	84.330330	79.49991	71.42372
4	85.619150	81.79583	72.85627

In [249]: ALL_PP.tail()

```

Out[249]:
country countryCode institution_id \
1263 Saudi Arabia SAU 703099
1264 Turkey TUR 705124
1265 Georgia GEO 204001
1266 United States USA 508335
1267 United States USA 508059

```

```

link \
1263 {'@href': 'https://api.elsevier.com/analytics/...
1264 {'@href': 'https://api.elsevier.com/analytics/...
1265 {'@href': 'https://api.elsevier.com/analytics/...
1266 {'@href': 'https://api.elsevier.com/analytics/...
1267 {'@href': 'https://api.elsevier.com/analytics/...

```

	institution_name	metricType	2014	\
1263	Imam Abdulrahman Bin Faisal University	CitedPublications	296.0	
1264	Istanbul Medipol University	CitedPublications	203.0	
1265	Ivane Javakhishvili Tbilisi State University	CitedPublications	295.0	
1266	University of Rochester	CitedPublications	3011.0	
1267	Emory University	CitedPublications	5206.0	

	2015	2016	2017	2018	percent2014	percent2015	percent2016	\
1263	305.0	364.0	384.0	560.0	79.78437	75.495056	79.302826	
1264	293.0	295.0	262.0	207.0	78.07692	72.524750	70.743410	
1265	347.0	383.0	342.0	366.0	65.84821	68.441810	66.034485	
1266	2939.0	2919.0	2795.0	2649.0	83.59245	83.022600	83.044090	
1267	5230.0	5187.0	5126.0	4687.0	88.08799	87.810610	84.977066	

	percent2017	percent2018
1263	66.435990	59.447987
1264	62.679430	47.260273
1265	65.391970	60.098522
1266	76.912490	68.948460
1267	80.992256	72.252200

```
In [250]: US_PP=ALL_PP[ALL_PP.countryCode=='USA']
```

```
In [251]: chuck=[]
```

```
for name in UR_Peer_FWCI.institution_name:
    chuck.append(US_PP[US_PP.institution_name==name])
```

```
In [252]: UR_Peer_PP=pd.concat(chuck, ignore_index=True)
```

```
In [253]: UR_Peer_PP=UR_Peer_PP[['institution_name','percent2014','percent2015','percent2016'],
```

```
In [254]: UR_Peer_PP=UR_Peer_PP.drop_duplicates()
```

```
In [255]: UR_Peer_PP.shape[0]
```

```
Out[255]: 13
```

```
In [256]: UR_Peer_PP.loc[:,['UniAbbr']]=abb
```

```
In [257]: UR_Peer_PP.loc[:,['Mean_%PubCited']]=UR_Peer_PP.iloc[:,1:5].mean(axis=1)
```

```
In [258]: UR_Peer_PP
```

```
Out[258]:
```

	institution_name	percent2014	percent2015	percent2016	\
0	Stanford University	87.149445	86.745750	85.619150	
1	Emory University	88.087990	87.810610	84.977066	
2	Johns Hopkins University	89.002870	87.677800	86.510290	
3	Duke University	88.075410	86.854250	85.007920	
4	Northwestern University	86.777405	86.313380	85.312300	
6	University of Pennsylvania	87.413540	85.737160	83.930275	
7	Boston University	86.645850	87.164610	84.995610	
8	New York University	85.079050	83.887920	81.683710	
10	University of Chicago	84.766730	82.955670	83.135560	
11	Case Western Reserve University	84.712010	83.535610	83.017590	
12	Vanderbilt University	88.025280	87.215770	85.106384	
13	University of Rochester	83.592450	83.022600	83.044090	
15	Tulane University	82.566730	83.222595	83.673470	

	percent2017	percent2018
0	81.795830	72.856270
1	80.992256	72.252200
2	82.894350	73.282555
3	81.594154	71.465890

4	82.407074	71.779500
6	80.257805	69.561550
7	80.863884	71.050520
8	77.177086	66.417710
10	78.535040	71.705900
11	77.285620	67.933495
12	78.452440	68.441520
13	76.912490	68.948460
15	77.565506	65.554360

```
In [259]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PercPublsCited"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PercPublsCited
```

```
In [261]: UR_Peer_PP['Mean_%PubCited']=UR_Peer_PP.mean(axis=1)
```

```
In [262]: UR_Peer_PP=UR_Peer_PP.sort_values(by='Mean_%PubCited', ascending=False)
```

```
In [263]: UR_Peer_PP.reset_index(inplace=True, drop=True)
```

```
In [264]: C=UR_Peer_PP[['institution_name','Mean_%PubCited']]
```

```
In [265]: UR_Peer_PP.to_csv("UofR_Global_Peers_Cited_Publs.csv", index=False)
```

```
In [836]: # Top 1 % cited
```

```
In [266]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PubTopJournal"
```

```
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\PubTopJournalPercentile
```

```
In [267]: Top1All=pd.read_csv("THE_ALLUNI_PP.csv")
```

```
In [268]: Top1All.columns
```

```
Out[268]: Index(['country', 'countryCode', 'institution_id', 'link', 'institution_name',
               'metricType', 't1_2014', 't1_2015', 't1_2016', 't1_2017', 't1_2018',
               't1_percent2014', 't1_percent2015', 't1_percent2016', 't1_percent2017',
               't1_percent2018', 't5_2014', 't5_2015', 't5_2016', 't5_2017', 't5_2018',
               't5_percent2014', 't5_percent2015', 't5_percent2016', 't5_percent2017',
               't5_percent2018', 't10_2014', 't10_2015', 't10_2016', 't10_2017',
               't10_2018', 't10_percent2014', 't10_percent2015', 't10_percent2016',
               't10_percent2017', 't10_percent2018', 't25_2014', 't25_2015',
               't25_2016', 't25_2017', 't25_2018', 't25_percent2014',
               't25_percent2015', 't25_percent2016', 't25_percent2017',
               't25_percent2018'],
              dtype='object')
```

```
In [269]: Top1All.tail()
```

```

Out[269]:
country countryCode institution_id \
1265 Saudi Arabia SAU 703099
1266 Turkey TUR 705124
1267 Georgia GEO 204001
1268 United States USA 508335
1269 United States USA 508059

link \
1265 {'@href': 'https://api.elsevier.com/analytics/...
1266 {'@href': 'https://api.elsevier.com/analytics/...
1267 {'@href': 'https://api.elsevier.com/analytics/...
1268 {'@href': 'https://api.elsevier.com/analytics/...
1269 {'@href': 'https://api.elsevier.com/analytics/...

institution_name \
1265 Imam Abdulrahman Bin Faisal University
1266 Istanbul Medipol University
1267 Ivane Javakhishvili Tbilisi State University
1268 University of Rochester
1269 Emory University

metricType t1_2014 t1_2015 t1_2016 t1_2017 \
1265 PublicationsInTopJournalPercentiles 1.0 2.0 5.0 6.0
1266 PublicationsInTopJournalPercentiles 3.0 2.0 2.0 5.0
1267 PublicationsInTopJournalPercentiles 1.0 1.0 2.0 2.0
1268 PublicationsInTopJournalPercentiles 162.0 164.0 143.0 138.0
1269 PublicationsInTopJournalPercentiles 277.0 348.0 318.0 348.0

... t25_2014 t25_2015 t25_2016 t25_2017 t25_2018 t25_percent2014 \
1265 ... 87.0 103.0 140.0 198.0 353.0 26.605505
1266 ... 67.0 106.0 77.0 99.0 90.0 27.800830
1267 ... 205.0 241.0 278.0 266.0 335.0 50.368546
1268 ... 2163.0 2125.0 2131.0 2130.0 2309.0 70.364340
1269 ... 3967.0 3923.0 3854.0 3975.0 4143.0 73.708660

t25_percent2015 t25_percent2016 t25_percent2017 t25_percent2018
1265 29.428572 34.567900 38.521400 40.762123
1266 29.041096 20.810812 27.049181 25.210085
1267 53.200882 52.751423 55.416668 58.566433
1268 69.331154 67.436710 66.645805 66.522610
1269 72.246780 68.907560 67.407160 68.186310

```

[5 rows x 46 columns]

```
In [270]: Top1=Top1All[['institution_name','t1_percent2014','t1_percent2015','t1_percent2016',
```

```
In [271]: Top1=Top1.drop_duplicates()
```

```
In [272]: Top1['Total_Top1']=Top1[['institution_name','t1_percent2014','t1_percent2015','t1_per
```

```
In [273]: Top1.head()
```

```
Out[273]:
```

	institution_name	t1_percent2014	\
0	University of Oxford	6.250000	
1	Jet Propulsion Laboratory, California Institut...	4.939920	
4	California Institute of Technology	7.905651	
5	University of Cambridge	7.298050	
6	Stanford University	8.733065	

	t1_percent2015	t1_percent2016	t1_percent2017	t1_percent2018	Total_Top1
0	7.244391	7.178117	6.104996	6.456400	6.646781
1	5.694445	4.605993	3.953229	6.402915	5.119300
4	6.580714	6.523930	6.254635	6.737181	6.800422
5	7.159152	7.903886	7.334815	6.474752	7.234131
6	8.676345	9.068627	8.334686	7.886435	8.539832

```
In [274]: chuck=[]
```

```
for name in UR_Peer_FWCI.institution_name:
    chuck.append(Top1[Top1.institution_name==name])
```

```
In [275]: UR_PEER_Top1=pd.concat(chuck, ignore_index=True)
```

```
In [276]: UR_PEER_Top1=UR_PEER_Top1.sort_values(by='Total_Top1', ascending=False)
```

```
In [277]: UR_PEER_Top1.reset_index(inplace=True, drop=True)
```

```
In [278]: D=UR_PEER_Top1[['institution_name','Total_Top1']] # top1%
```

```
In [279]: # top 10%
```

```
Top10=Top1All[['institution_name','t10_percent2014','t10_percent2015','t10_percent2016','t10_percent2017','t10_percent2018','Total_Top10']]
```

```
In [280]: Top10=Top10.drop_duplicates()
```

```
In [281]: Top10['Total_Top10']=Top10[['institution_name','t10_percent2014','t10_percent2015','t10_percent2016','t10_percent2017','t10_percent2018']]
```

```
In [282]: Top10.head()
```

```
Out[282]:
```

	institution_name	t10_percent2014	\
0	University of Oxford	48.004738	
1	Jet Propulsion Laboratory, California Institut...	30.373833	
4	California Institute of Technology	44.297565	
5	University of Cambridge	48.022285	
6	Stanford University	50.878800	

	t10_percent2015	t10_percent2016	t10_percent2017	t10_percent2018	\
0	49.169380	47.268753	47.715576	46.048240	
1	33.541668	38.512764	38.251670	35.346176	
4	42.412033	51.662468	47.119900	45.228786	

5	49.646930	50.010223	49.757526	47.258140
6	51.818962	50.490196	50.584324	48.526580

	Total_Top10
0	47.641337
1	35.205222
4	46.144150
5	48.939021
6	50.459772

In [283]: Top10.tail()

```
Out[283]:
```

	institution_name	t10_percent2014	\
1258	Imam Abdulrahman Bin Faisal University	8.562691	
1259	Istanbul Medipol University	9.128631	
1260	Ivane Javakhishvili Tbilisi State University	33.906630	
1261	University of Rochester	45.673390	
1269	Emory University	44.797474	

	t10_percent2015	t10_percent2016	t10_percent2017	t10_percent2018	\
1258	9.142857	16.543210	14.007781	17.205542	
1259	8.767123	8.108108	12.841530	12.885155	
1260	27.814570	35.294117	46.041668	47.727272	
1261	42.675365	41.455696	40.957447	37.971764	
1269	45.930020	41.981050	41.783960	39.335090	

	Total_Top10
1258	13.092416
1259	10.346109
1260	38.156851
1261	41.746732
1269	42.765519

In [284]: chuck=[]

```
for name in UR_Peer_FWCI.institution_name:
    chuck.append(Top10[Top10.institution_name==name])
```

In [285]: UR_PEER_Top10=pd.concat(chuck, ignore_index=True)

In [286]: UR_PEER_Top10=UR_PEER_Top10.sort_values(by='Total_Top10', ascending=False)

In [287]: UR_PEER_Top10.reset_index(inplace=True, drop=True)

In [288]: E=UR_PEER_Top10[['institution_name','Total_Top10']]

In [301]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\ScholarlyOutput"

C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\OUtput Data\ScholarlyOutput


```
In [303]: # filter publication data for UR global peer
```

```
In [305]: so_data_USA.head()
```

```
Out[305]:
```

	institution_name	2014	2015	2016	\
0	Jet Propulsion Laboratory, California Institut...	1913	1800	2181	
1	California Institute of Technology	4518	4347	4553	
2	Stanford University	12801	13656	13664	
3	Massachusetts Institute of Technology	9645	9957	10023	
4	Princeton University	4335	4629	4544	

	2017	2018	Total
0	2052	2200	10146
1	4567	4884	22869
2	14233	14729	69083
3	10191	10458	50274
4	4635	4891	23034

```
In [306]: chuck=[]
```

```
for name in UR_Peer_FWCI.institution_name:
    chuck.append(so_data_USA[so_data_USA.institution_name==name])
```

```
In [307]: A=pd.concat(chuck, ignore_index=True)
```

```
In [308]: A=A.drop_duplicates()
```

```
In [310]: A=A.sort_values(by='Total', ascending=False)
```

```
In [311]: A.head()
```

```
Out[311]:
```

	institution_name	2014	2015	2016	2017	2018	Total
0	Stanford University	12801	13656	13664	14233	14729	69083
2	Johns Hopkins University	12185	12936	13173	13668	14047	66009
6	University of Pennsylvania	10988	11239	11531	11792	13023	58573
8	New York University	9108	9707	10073	10599	11247	50734
3	Duke University	8646	8885	8838	9171	9571	45111

```
In [319]: A=A.iloc[:,[0,-1]]
```

```
In [320]: A.reset_index(inplace=True, drop=True)
```

```
In [321]: A.head()
```

```
Out[321]:
```

	institution_name	Total
0	Stanford University	69083
1	Johns Hopkins University	66009
2	University of Pennsylvania	58573
3	New York University	50734
4	Duke University	45111

```
In [313]: len(A)
```

```
Out[313]: 13
```

```
In [312]: C.head()
```

```
Out[312]:
```

	institution_name	Mean_%PubCited
0	Johns Hopkins University	83.873573
1	Stanford University	82.833289
2	Emory University	82.824024
3	Duke University	82.599525
4	Northwestern University	82.517932

```
In [314]: len(C)
```

```
Out[314]: 13
```

```
In [315]: D.head()
```

```
Out[315]:
```

	institution_name	Total_Top1
0	Stanford University	8.539832
1	Northwestern University	7.651071
2	University of Chicago	7.156997
3	University of Pennsylvania	6.827791
4	Boston University	6.607170

```
In [317]: len(E)
```

```
Out[317]: 13
```

```
In [322]: part1=A.join(C.set_index('institution_name'), on='institution_name')
```

```
In [323]: part2=part1.join(D.set_index('institution_name'), on='institution_name')
```

```
In [324]: part3=part2.join(E.set_index('institution_name'), on='institution_name')
```

```
In [325]: part3
```

```
Out[325]:
```

	institution_name	Total	Mean_%PubCited	Total_Top1	\
0	Stanford University	69083	82.833289	8.539832	
1	Johns Hopkins University	66009	83.873573	5.689941	
2	University of Pennsylvania	58573	81.380066	6.827791	
3	New York University	50734	78.849095	5.585733	
4	Duke University	45111	82.599525	6.461731	
5	University of Chicago	44095	80.219780	7.156997	
6	Northwestern University	40315	82.517932	7.651071	
7	Vanderbilt University	31056	81.448279	5.699216	
8	Emory University	30786	82.824024	5.530431	
9	Boston University	28873	82.144095	6.607170	
10	Case Western Reserve University	22234	79.296865	4.582277	

11	University of Rochester	18132	79.104018	4.659141
12	Tulane University	9021	78.516532	4.564512

	Total_Top10
0	50.459772
1	43.892943
2	45.489412
3	41.123320
4	46.996039
5	49.374273
6	48.214202
7	43.224036
8	42.765519
9	46.571934
10	38.889898
11	41.746732
12	37.115668

```
In [326]: B=UR_Peer_FWCI[['institution_name','AVERAGE_FWCI']]
```

```
In [327]: B.head()
```

```
Out[327]:
```

	institution_name	AVERAGE_FWCI
8	Stanford University	2.4008
3	Emory University	2.0622
6	Johns Hopkins University	2.0071
2	Duke University	1.9947
4	Northwestern University	1.9904

```
In [328]: part4=part3.join(B.set_index('institution_name'), on='institution_name')
```

```
In [329]: part4.sort_values(by='Total', ascending=False)
```

```
Out[329]:
```

	institution_name	Total	Mean_%PubCited	Total_Top1	\
0	Stanford University	69083	82.833289	8.539832	
1	Johns Hopkins University	66009	83.873573	5.689941	
2	University of Pennsylvania	58573	81.380066	6.827791	
3	New York University	50734	78.849095	5.585733	
4	Duke University	45111	82.599525	6.461731	
5	University of Chicago	44095	80.219780	7.156997	
6	Northwestern University	40315	82.517932	7.651071	
7	Vanderbilt University	31056	81.448279	5.699216	
8	Emory University	30786	82.824024	5.530431	
9	Boston University	28873	82.144095	6.607170	
10	Case Western Reserve University	22234	79.296865	4.582277	
11	University of Rochester	18132	79.104018	4.659141	
12	Tulane University	9021	78.516532	4.564512	

	Total_Top10	AVERAGE_FWCI
--	-------------	--------------

0	50.459772	2.4008
1	43.892943	2.0071
2	45.489412	1.9863
3	41.123320	1.9356
4	46.996039	1.9947
5	49.374273	1.8937
6	48.214202	1.9904
7	43.224036	1.8476
8	42.765519	2.0622
9	46.571934	1.9637
10	38.889898	1.8657
11	41.746732	1.8020
12	37.115668	1.5315

```
In [330]: cd "C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\research_perfi
C:\Users\jchen148\THE Rankings\Report to Jane\OK Files\Output Data\research_performance_Profile
```

```
In [331]: part4.to_csv('UR_GloPeers_Research_Performance_Profile.csv', index=False)
```

60 THE USA 163 Universities Distribution Plot

61 From the distribution plot below,

62 we can see we are above 75% of the other USA Universities

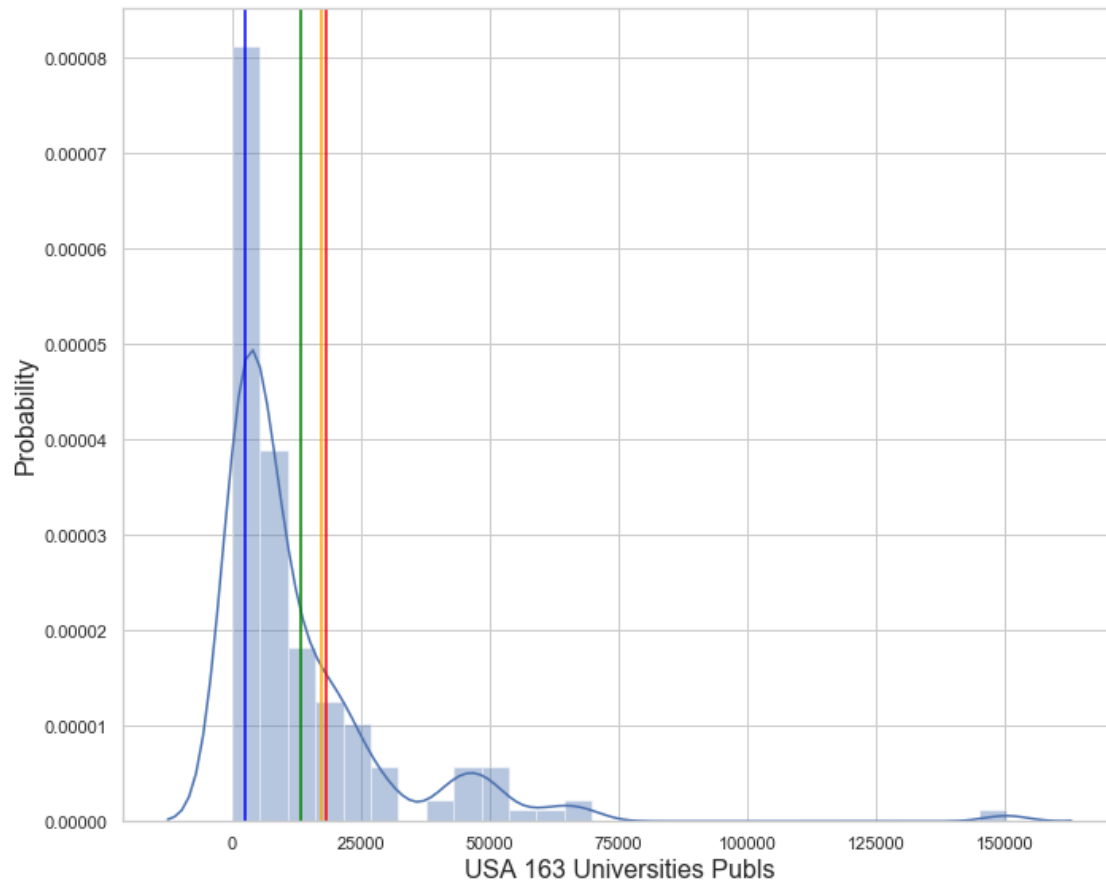
63 in publications from 2014 to 2018.

64 However, we can see Q3 is very close to the mean,

65 which is the green line. This is a right-skewed distribution.

```
In [1110]: import pandas as pd
fig, ax = plt.subplots(figsize=(10,8))
x = pd.Series(so_data_USA['Total'], name="USA Universities Publs") # 163 universities
ax = sns.distplot(x)

ax.set_xlabel("USA 163 Universities Publs",fontSize=16)
ax.set_ylabel("Probability",fontSize=16)
plt.axvline(18132, color='red') # this is where U of R
plt.axvline(np.mean(so_data_USA['Total']), color='green') # this is the mean, 17588
plt.axvline(np.percentile(so_data_USA['Total'], 25.0), color='blue') # Q1
plt.axvline(np.percentile(so_data_USA['Total'], 75.0), color='orange') # Q3
#plt.legend()
plt.tight_layout()
```



```
In [354]: so_data_USA=so_data_USA.reset_index()
```

```
In [355]: URpp=so_data_USA[so_data_USA.institution_name=='University of Rochester']
          URpp
```

```
Out[355]:
```

	institution_name	2014	2015	2016	2017	2018
162	University of Rochester	3602	3540	3515	3633	3842

```
In [356]: URpp=URpp.set_index('institution_name')
```

```
In [368]: URpp.agg('sum')
```

```
Out[368]:
```

2014	3602
2015	3540
2016	3515
2017	3633
2018	3842
dtype: int64	

```
In [340]: inputdata=pd.DataFrame(data.iloc[:, :6], columns=['2014', '2015', '2016', '2017', '2018'])
```

```
In [341]: inputdata.head()
```

```
Out[341]:
```

	2014	2015	2016	2017	2018
institution_name					
University of Rochester	3602	3540	3515	3633	3842

```
In [342]: inputdata.reset_index(drop=True, inplace=True)
```

```
In [214]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(2)
```

```
inst_country=[]
inst_cc=[]
inst_id=[]
inst_link=[]
inst_name=[]
metricType=[]
threshold=[]
t1_value2014=[]
t1_value2015=[]
t1_value2016=[]
t1_value2017=[]
t1_value2018=[]
t1_percentage2014=[]
t1_percentage2015=[]
t1_percentage2016=[]
t1_percentage2017=[]
t1_percentage2018=[]
t5_value2014=[]
t5_value2015=[]
t5_value2016=[]
t5_value2017=[]
t5_value2018=[]
t5_percentage2014=[]
t5_percentage2015=[]
t5_percentage2016=[]
t5_percentage2017=[]
t5_percentage2018=[]
t10_value2014=[]
t10_value2015=[]
t10_value2016=[]
t10_value2017=[]
t10_value2018=[]
t10_percentage2014=[]
```

```

t10_percentage2015=[]
t10_percentage2016=[]
t10_percentage2017=[]
t10_percentage2018=[]
t25_value2014=[]
t25_value2015=[]
t25_value2016=[]
t25_value2017=[]
t25_value2018=[]
t25_percentage2014=[]
t25_percentage2015=[]
t25_percentage2016=[]
t25_percentage2017=[]
t25_percentage2018=[]

```

```

for line in data['University id'][50:75]:
    url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=P
    # print(url.format(line))
    resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                    'X-ELS-APIKey': "d3794058e2b24417b5dfd0ef8990e2dc"})
    parsed=json.dumps(resp.json(),
                       sort_keys=True,
                       indent=4, separators=(',', ': '))
    # with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
    #     json.dump(resp.json(), jsonfile)
    # print(parsed)
    # data.update(a_dict)
    result=json.loads(parsed)
    if 'results' in result:
        if len(result['results'])>=1:
            if 'institution' in result['results'][0]:
                # if 'country' in result['results'][0]['institution']:
                #     inst_country.append(result['results'][0]['institution']['country'])
                # if 'countryCode' in result['results'][0]['institution']:
                #     inst_cc.append(result['results'][0]['institution']['countryCode'])
                # if 'id' in result['results'][0]['institution']:
                #     inst_id.append(result['results'][0]['institution']['id'])
                # if 'link' in result['results'][0]['institution']:
                #     inst_link.append(result['results'][0]['institution']['link'])
                # if 'name' in result['results'][0]['institution']:
                #     inst_name.append(result['results'][0]['institution']['name'])
            if 'metrics' in result['results'][0]:
                # if len(result['results'][0]['metrics'])>=1:
                #     if 'metricType' in result['results'][0]['metrics'][0]:
                #         metricType.append(result['results'][0]['metrics'][0]['metricType'])
                #     if 'values' in result['results'][0]['metrics'][0]:

```

```

# print(result['results'][0]['metrics'][0]['values'][1]['threshold'])
for i in range(0, len(result['results'][0]['metrics'][0]['values'])):
    threshold.append(result['results'][0]['metrics'][0]['values'][i])
    if 'valueByYear' in result['results'][0]['metrics'][0]['values']:
#         if i ==0:
#             if '2014' in result['results'][0]['metrics'][0]['values']:
#                 t1_value2014.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2015' in result['results'][0]['metrics'][0]['values']:
#                 t1_value2015.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2016' in result['results'][0]['metrics'][0]['values']:
#                 t1_value2016.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2017' in result['results'][0]['metrics'][0]['values']:
#                 t1_value2017.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2018' in result['results'][0]['metrics'][0]['values']:
#                 t1_value2018.append(result['results'][0]['metrics'][0]['values'][i])
#         if i ==1:
#             if '2014' in result['results'][0]['metrics'][0]['values']:
#                 t5_value2014.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2015' in result['results'][0]['metrics'][0]['values']:
#                 t5_value2015.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2016' in result['results'][0]['metrics'][0]['values']:
#                 t5_value2016.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2017' in result['results'][0]['metrics'][0]['values']:
#                 t5_value2017.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2018' in result['results'][0]['metrics'][0]['values']:
#                 t5_value2018.append(result['results'][0]['metrics'][0]['values'][i])
#         if i ==2:
#             if '2014' in result['results'][0]['metrics'][0]['values']:
#                 t10_value2014.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2015' in result['results'][0]['metrics'][0]['values']:
#                 t10_value2015.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2016' in result['results'][0]['metrics'][0]['values']:
#                 t10_value2016.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2017' in result['results'][0]['metrics'][0]['values']:
#                 t10_value2017.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2018' in result['results'][0]['metrics'][0]['values']:
#                 t10_value2018.append(result['results'][0]['metrics'][0]['values'][i])
#         if i ==3:
#             if '2014' in result['results'][0]['metrics'][0]['values']:
#                 t25_value2014.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2015' in result['results'][0]['metrics'][0]['values']:
#                 t25_value2015.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2016' in result['results'][0]['metrics'][0]['values']:
#                 t25_value2016.append(result['results'][0]['metrics'][0]['values'][i])
#             if '2017' in result['results'][0]['metrics'][0]['values']:
#                 t25_value2017.append(result['results'][0]['metrics'][0]['values'][i])

```



```

        if '2018' in result['results'][0]['metrics'][0]['values']:
            t25_value2018.append(result['results'][0]['metrics']

if 'percentageByYear' in result['results'][0]['metrics'][0][
#
    if i ==0:
        if '2014' in result['results'][0]['metrics'][0]['values']:
            t1_percentage2014.append(result['results'][0]['metrics']
        if '2015' in result['results'][0]['metrics'][0]['values']:
            t1_percentage2015.append(result['results'][0]['metrics']
        if '2016' in result['results'][0]['metrics'][0]['values']:
            t1_percentage2016.append(result['results'][0]['metrics']
        if '2017' in result['results'][0]['metrics'][0]['values']:
            t1_percentage2017.append(result['results'][0]['metrics']
        if '2018' in result['results'][0]['metrics'][0]['values']:
            t1_percentage2018.append(result['results'][0]['metrics']

#
    if i ==1:
        if '2014' in result['results'][0]['metrics'][0]['values']:
            t5_percentage2014.append(result['results'][0]['metrics']
        if '2015' in result['results'][0]['metrics'][0]['values']:
            t5_percentage2015.append(result['results'][0]['metrics']
        if '2016' in result['results'][0]['metrics'][0]['values']:
            t5_percentage2016.append(result['results'][0]['metrics']
        if '2017' in result['results'][0]['metrics'][0]['values']:
            t5_percentage2017.append(result['results'][0]['metrics']
        if '2018' in result['results'][0]['metrics'][0]['values']:
            t5_percentage2018.append(result['results'][0]['metrics']

#
    if i ==2:
        if '2014' in result['results'][0]['metrics'][0]['values']:
            t10_percentage2014.append(result['results'][0]['metrics']
        if '2015' in result['results'][0]['metrics'][0]['values']:
            t10_percentage2015.append(result['results'][0]['metrics']
        if '2016' in result['results'][0]['metrics'][0]['values']:
            t10_percentage2016.append(result['results'][0]['metrics']
        if '2017' in result['results'][0]['metrics'][0]['values']:
            t10_percentage2017.append(result['results'][0]['metrics']
        if '2018' in result['results'][0]['metrics'][0]['values']:
            t10_percentage2018.append(result['results'][0]['metrics']

#
    if i ==3:
        if '2014' in result['results'][0]['metrics'][0]['values']:
            t25_percentage2014.append(result['results'][0]['metrics']
        if '2015' in result['results'][0]['metrics'][0]['values']:
            t25_percentage2015.append(result['results'][0]['metrics']
        if '2016' in result['results'][0]['metrics'][0]['values']:
            t25_percentage2016.append(result['results'][0]['metrics']
        if '2017' in result['results'][0]['metrics'][0]['values']:

```

```

        t25_percentage2017.append(result['results'][0]['metrics'][0]['values'][0])
        if '2018' in result['results'][0]['metrics'][0]['values']:
            t25_percentage2018.append(result['results'][0]['metrics'][0]['values'][0])
        else:
            t1_value2014.append('')
            t1_value2015.append('')
            t1_value2016.append('')
            t1_value2017.append('')
            t1_value2018.append('')
            t1_percentage2014.append('')
            t1_percentage2015.append('')
            t1_percentage2016.append('')
            t1_percentage2017.append('')
            t1_percentage2018.append('')

        else:
            t1_value2014.append('')
            t1_value2015.append('')
            t1_value2016.append('')
            t1_value2017.append('')
            t1_value2018.append('')
            t1_percentage2014.append('')
            t1_percentage2015.append('')
            t1_percentage2016.append('')
            t1_percentage2017.append('')
            t1_percentage2018.append('')

        if 'threshold' in result['results'][0]['metrics'][0]['values']:
            threshold.append(result['results'][0]['metrics'][0]['values'][0])

s1=pd.Series(inst_country, name='country')
s2=pd.Series(inst_cc, name='countryCode')
s3=pd.Series(inst_id, name='institution_id')
s4=pd.Series(inst_link, name='link')
s5=pd.Series(inst_name, name='institution_name')
s6=pd.Series(metricType, name='metricType')
s7=pd.Series(threshold, name='threshold')
s8=pd.Series(t1_value2014, name='t1_2014')
s9=pd.Series(t1_value2015, name='t1_2015')
s10=pd.Series(t1_value2016, name='t1_2016')
s11=pd.Series(t1_value2017, name='t1_2017')
s12=pd.Series(t1_value2018, name='t1_2018')
s13=pd.Series(t1_percentage2014, name='t1_percent2014')
s14=pd.Series(t1_percentage2015, name='t1_percent2015')
s15=pd.Series(t1_percentage2016, name='t1_percent2016')
s16=pd.Series(t1_percentage2017, name='t1_percent2017')
s17=pd.Series(t1_percentage2018, name='t1_percent2018')

```

```

s18=pd.Series(t5_value2014, name='t5_2014')
s19=pd.Series(t5_value2015, name='t5_2015')
s20=pd.Series(t5_value2016, name='t5_2016')
s21=pd.Series(t5_value2017, name='t5_2017')
s22=pd.Series(t5_value2018, name='t5_2018')
s23=pd.Series(t5_percentage2014, name='t5_percent2014')
s24=pd.Series(t5_percentage2015, name='t5_percent2015')
s25=pd.Series(t5_percentage2016, name='t5_percent2016')
s26=pd.Series(t5_percentage2017, name='t5_percent2017')
s27=pd.Series(t5_percentage2018, name='t5_percent2018')
s28=pd.Series(t10_value2014, name='t10_2014')
s29=pd.Series(t10_value2015, name='t10_2015')
s30=pd.Series(t10_value2016, name='t10_2016')
s31=pd.Series(t10_value2017, name='t10_2017')
s32=pd.Series(t10_value2018, name='t10_2018')
s33=pd.Series(t10_percentage2014, name='t10_percent2014')
s34=pd.Series(t10_percentage2015, name='t10_percent2015')
s35=pd.Series(t10_percentage2016, name='t10_percent2016')
s36=pd.Series(t10_percentage2017, name='t10_percent2017')
s37=pd.Series(t10_percentage2018, name='t10_percent2018')
s38=pd.Series(t25_value2014, name='t25_2014')
s39=pd.Series(t25_value2015, name='t25_2015')
s40=pd.Series(t25_value2016, name='t25_2016')
s41=pd.Series(t25_value2017, name='t25_2017')
s42=pd.Series(t25_value2018, name='t25_2018')
s43=pd.Series(t25_percentage2014, name='t25_percent2014')
s44=pd.Series(t25_percentage2015, name='t25_percent2015')
s45=pd.Series(t25_percentage2016, name='t25_percent2016')
s46=pd.Series(t25_percentage2017, name='t25_percent2017')
s47=pd.Series(t25_percentage2018, name='t25_percent2018')

DF=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,s16, s17,s18,s19,s20,
              s28,s29,s30,s31,s32,s33,s34,s35,s36,s37,s38,s39,s40, s41,s42,s43,s44,s45,
              s46,s47])

DF.to_csv("THE_UNI_PublicationsInTopJournalPercentiles_ALL_3.csv", index=False) # 0

#print(threshold)

In [103]: for line in data['University id'][:2]:
            url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=S
            # print(url.format(line))
            resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                            'X-ELS-APIKey': "a464321ef5063d696ada17f8c159a44c"})
            parsed=json.dumps(resp.json()),

```

```

        sort_keys=True,
        indent=4, separators=(',', ' ': ))
#     with open("THE_UNI_ID_METRIC_ALL.json", 'w') as jsonfile:
#         json.dump(resp.json(), jsonfile)
#     print(parsed)
#     data.update(a_dict)
    result=json.loads(parsed)
    print(result['results'])

```

```
[{'institution': {'country': 'United States', 'countryCode': 'USA', 'id': 508092, 'link': {'@h
```

```
In [66]: with open("THE_UNI_ID_METRIC_TEST.json") as outputfile:
        out=json.load(outputfile)
```

```
In [67]: out
```

```
Out[67]: {'link': {'@ref': 'self',
  '@href': 'https://api.elsevier.com/analytics/scival/institution/metrics?journalImpa
  '@type': 'application/json'},
  'dataSource': {'sourceName': 'Scopus', 'lastUpdated': '2020-01-01'},
  'results': [{'metrics': [{'metricType': 'ScholarlyOutput',
    'valueByYear': {'2014': 1913,
      '2015': 1800,
      '2016': 2181,
      '2017': 2052,
      '2018': 2200}}],
    'institution': {'link': {'@ref': 'self',
      '@href': 'https://api.elsevier.com/analytics/scival/institution/508092?apiKey=7a
      '@type': 'application/json'},
      'name': 'Jet Propulsion Laboratory, California Institute of Technology',
      'id': 508092,
      'uri': 'Institution/508092',
      'country': 'United States',
      'countryCode': 'USA'}}}]}
```

```
In [86]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(2)

university_name=[]
university_id=[]
country=[]
countryCode=[]
df=pd.DataFrame()
```

```

for line in want_3[:4]:
#     query = "name(school)"
url= "https://api.elsevier.com/metrics/institution/search?query=name({})&start=0&
resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ': '))
#     print(parsed)
#     data.update(a_dict)
result=json.loads(parsed)
#     data=parsed[1]
#     print(result)
data=result['results']
print(data)
#     if (data[0]['country'] is not None):

```

```

[{'country': 'United Kingdom', 'countryCode': 'GBR', 'id': 315091, 'link': {'@href': 'https://api.elsevier.com/metrics/institution/search?query=name({})&start=0&
[{'country': 'United States', 'countryCode': 'USA', 'id': 508092, 'link': {'@href': 'https://api.elsevier.com/metrics/institution/search?query=name({})&start=0&
[{'country': 'United Kingdom', 'countryCode': 'GBR', 'id': 315068, 'link': {'@href': 'https://api.elsevier.com/metrics/institution/search?query=name({})&start=0&
[{'country': 'United States', 'countryCode': 'USA', 'id': 508219, 'link': {'@href': 'https://api.elsevier.com/metrics/institution/search?query=name({})&start=0&

```

In [3]: cd "C:\Users\jchen148\THE Rankings\Report to Jane"

C:\Users\jchen148\THE Rankings\Report to Jane

```

In [97]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(0.1)

university_name=[]
university_id=[]
country=[]
countryCode=[]
df=pd.DataFrame()

for line in want_3[:10]:
#     query = "name(school)"
url= "https://api.elsevier.com/metrics/institution/search?query=name({})&start=0&
resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ': '))

```

```

result=json.loads(parsed)
data=result['results']
for i in data:
    if i is not None:
#     if data[0] is not None:
        countries=i['country']
        unames=i['name']
        uids=i['id']
        codes=i['countryCode']
        if (countries is not None):
            country.append(countries)
        else:
            country.append("")
        if (unames is not None):
            university_name.append(unames)
        else:
            university_name.append("")
        if (uids is not None):
            university_id.append(uids)
        else:
            university_id.append("")
        if (codes is not None):
            countryCode.append(codes)
        else:
            countryCode.append("")
df=pd.DataFrame({'University Name':university_name, 'University id':university_id, 'Country Code':countryCode})
df.to_csv("THE_CountryCode_Result_1202.csv")

```

```

In [98]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(0.1)

university_name=[]
university_id=[]
country=[]
countryCode=[]
df=pd.DataFrame()

for line in want_3[10:20]:
#     query = "name(school)"
    url= "https://api.elsevier.com/metrics/institution/search?query=name({})&start=0&size=10"
    resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                    'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})

```

```

        parsed=json.dumps(resp.json(),
                           sort_keys=True,
                           indent=4, separators=(',', ': '))
    result=json.loads(parsed)
    data=result['results']
    for i in data:
        if i is not None:
#         if data[0] is not None:
            countries=i['country']
            unames=i['name']
            uids=i['id']
            codes=i['countryCode']
            if (countries is not None):
                country.append(countries)
            else:
                country.append("")
            if (unames is not None):
                university_name.append(unames)
            else:
                university_name.append("")
            if (uids is not None):
                university_id.append(uids)
            else:
                university_id.append("")
            if (codes is not None):
                countryCode.append(codes)
            else:
                countryCode.append("")
    df=pd.DataFrame({'University Name':university_name, 'University id':university_id, 'Country Code':countryCode})
    df.to_csv("THE_CountryCode_Result_1202_2.csv")

```

```

In [100]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(0.1)

university_name=[]
university_id=[]
country=[]
countryCode=[]
df=pd.DataFrame()

for line in want_3[20:30]:
#     query = "name(school)"
    url= "https://api.elsevier.com/metrics/institution/search?query=name({})&start=0"

```

```

resp = requests.get(url.format(line), headers={'Accept': 'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ': '))
result=json.loads(parsed)
data=result['results']
for i in data:
    if i is not None:
#     if data[0] is not None:
        countries=i['country']
        unames=i['name']
        uids=i['id']
        codes=i['countryCode']
        if (countries is not None):
            country.append(countries)
        else:
            country.append("")
        if (unames is not None):
            university_name.append(unames)
        else:
            university_name.append("")
        if (uids is not None):
            university_id.append(uids)
        else:
            university_id.append("")
        if (codes is not None):
            countryCode.append(codes)
        else:
            countryCode.append("")
df=pd.DataFrame({'University Name':university_name, 'University id':university_id, 'Country Code':countryCode})
df.to_csv("THE_CountryCode_Result_1202_3.csv")

```

```

In [ ]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(0.1)

university_name=[]
university_id=[]
country=[]
countryCode=[]
df=pd.DataFrame()

for line in want_3[20:30]:

```



```

#     query = "name(school)"
url= "https://api.elsevier.com/metrics/institution/search?query=name({})&start=0&c
resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ': '))
result=json.loads(parsed)
data=result['results']
for i in data:
    if i is not None:
#     if data[0] is not None:
        countries=i['country']
        unames=i['name']
        uids=i['id']
        codes=i['countryCode']
        if (countries is not None):
            country.append(countries)
        else:
            country.append("")
        if (unames is not None):
            university_name.append(unames)
        else:
            university_name.append("")
        if (uids is not None):
            university_id.append(uids)
        else:
            university_id.append("")
        if (codes is not None):
            countryCode.append(codes)
        else:
            countryCode.append("")
df=pd.DataFrame({'University Name':university_name, 'University id':university_id, 'Country Code':countryCode})
df.to_csv("THE_CountryCode_Result_1202_3.csv")

```

```

In [169]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(3)

university_name=[]
university_id=[]
country=[]
countryCode=[]
df=pd.DataFrame()

```

```

for line in want_3[75:]:
    line=re.sub('[^A-Za-z0-9]+',' ', line)
    # query = "name(school)"
    url= "https://api.elsevier.com/metrics/institution/search?query=name({})&start=0"
    resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                    'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
    # try:
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ': '))
    result=json.loads(parsed)
    data=result['results']
    # except ValueError:
    #     pass
    #     result=json.loads(parsed)
    #     data=result['results']
    for i in data:
        if i is None:
            pass
        else:
            # try:
            # if data[0] is not None:
            countries=i['country']
            unames=i['name']
            uids=i['id']
            codes=i['countryCode']
            if (countries is not None):
                country.append(countries)
            else:
                country.append("")
            if (unames is not None):
                university_name.append(unames)
            else:
                university_name.append("")
            if (uids is not None):
                university_id.append(uids)
            else:
                university_id.append("")
            if (codes is not None):
                countryCode.append(codes)
            else:
                countryCode.append("")
            # except (RuntimeError, TypeError, NameError, JSONDecodeError):
            #     pass
            df=pd.DataFrame({'University Name':university_name, 'University id':university_id, 'Country Code':countryCode})
            df.to_csv("THE_CountryCode_Result_1202_12.csv")
    # except ValueError:
    #     continue

```

```

In [153]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(3)

university_name=[]
university_id=[]
country=[]
countryCode=[]
df=pd.DataFrame()

for line in want_3[47:50]:
    # query = "name(school)"
    url= "https://api.elsevier.com/metrics/institution/search?query=name({})&start=0"
    resp = requests.get(url.format(line), headers={'Accept': 'application/json',
                                                    'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
    try:
        parsed=json.dumps(resp.json(),
                           sort_keys=True,
                           indent=4, separators=(',', ': '))
        result=json.loads(parsed)
        data=result['results']
        for i in data:
            if i is None:
                pass
            else:
                try:
# if data[0] is not None:
                    countries=i['country']
                    unames=i['name']
                    uids=i['id']
                    codes=i['countryCode']
                    if (countries is not None):
                        country.append(countries)
                    else:
                        country.append("")
                    if (unames is not None):
                        university_name.append(unames)
                    else:
                        university_name.append("")
                    if (uids is not None):
                        university_id.append(uids)
                    else:
                        university_id.append("")
                    if (codes is not None):
                        countryCode.append(codes)

```

```

        else:
            countryCode.append("")
    except (RuntimeError, TypeError, NameError, JSONDecodeError):
        pass
    df=pd.DataFrame({'University Name':university_name, 'Univers:
df.to_csv("THE_CountryCode_Result_1202_8.csv")

except ValueError:
    continue

```

```

In [126]: for line in want_3[38:40]:
           print(re.sub('[^A-Za-z0-9]+',' ', line))

```

North Carolina at Chapel Hill
Wageningen University Research

```

In [ ]: for line in want_3[38:40]:
        line=re.sub('[^A-Za-z0-9]+',' ', line)
        # query = "name(school)"
        url= "https://api.elsevier.com/metrics/institution/search?query=name("{}")&start:
        # resp = requests.get(url.format(line), headers={'Accept':'application/json',
        #                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})

```

```

In [137]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(3)

university_name=[]
university_id=[]
country=[]
countryCode=[]
df=pd.DataFrame()

for line in want_3[40:50]:
    # line=re.sub('[^A-Za-z0-9]+',' ', line)
    # query = "name(school)"
    url= "https://api.elsevier.com/metrics/institution/search?query=name("{}")&start:
    resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                    'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ': '))
    result=json.loads(parsed)
    data=result['results']
    for i in data:

```

```

        if i is not None:
#     if data[0] is not None:
            countries=i['country']
            unames=i['name']
            uids=i['id']
            codes=i['countryCode']
            if (countries is not None):
                country.append(countries)
            else:
                country.append("")
            if (unames is not None):
                university_name.append(unames)
            else:
                university_name.append("")
            if (uids is not None):
                university_id.append(uids)
            else:
                university_id.append("")
            if (codes is not None):
                countryCode.append(codes)
            else:
                countryCode.append("")
            df=pd.DataFrame({'University Name':university_name, 'University id':university_id, 'Country Code':countryCode})
            df.to_csv("THE_CountryCode_Result_1202_6.csv")

```

```

In [117]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(3)

university_name=[]
university_id=[]
country=[]
countryCode=[]
df=pd.DataFrame()

#for line in want_3[40:50]:
#    line=re.sub('[^A-Za-z0-9]+',' ', line)
#    query = "name(school)"
#name="University of Rochester"
url= "https://api.elsevier.com/metrics/institution/search?query=name(University%20of%20Rochester)"
resp = requests.get(url, headers={'Accept':'application/json',
                                   'X-ELS-APIKey': "a464321ef5063d696ada17f8c159a44c"})
parsed=json.dumps(resp.json()),

```

```

        sort_keys=True,
        indent=4, separators=(',', ' : '))
result=json.loads(parsed)
data=result['results']
#print(data)
for i in data:
    if i is not None:
#        if data[0] is not None:
            countries=i['country']
            unames=i['name']
            uids=i['id']
            codes=i['countryCode']
            if (countries is not None):
                country.append(countries)
            else:
                country.append("")
            if (unames is not None):
                university_name.append(unames)
            else:
                university_name.append("")
            if (uids is not None):
                university_id.append(uids)
            else:
                university_id.append("")
            if (codes is not None):
                countryCode.append(codes)
            else:
                countryCode.append("")
df=pd.DataFrame({'University Name':university_name, 'University id':university_id, 'Country Code':countryCode})
df.to_csv("THE_CountryCode_Result_1202_13.csv")

```

```

In [110]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(3)

university_name=[]
university_id=[]
country=[]
countryCode=[]
df=pd.DataFrame()

#for line in want_3[40:50]:
#    line=re.sub('[^A-Za-z0-9]+' , ' ', line)
#    query = "name(school)"
name="University of Rochester"

```

```

url= "https://api.elsevier.com/metrics/institution/search?query=name('{}')&start=0
resp = requests.get(url.format(name), headers={'Accept': 'application/json',
                                              'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ' : '))
result=json.loads(parsed)
data=result['results']
for i in data:
    if i is not None:
#     if data[0] is not None:
        countries=i['country']
        unames=i['name']
        uids=i['id']
        codes=i['countryCode']
        if (countries is not None):
            country.append(countries)
        else:
            country.append("")
        if (unames is not None):
            university_name.append(unames)
        else:
            university_name.append("")
        if (uids is not None):
            university_id.append(uids)
        else:
            university_id.append("")
        if (codes is not None):
            countryCode.append(codes)
        else:
            countryCode.append("")
df=pd.DataFrame({'University Name':university_name, 'University id':university_id, 'CountryCode':countryCode})
df.to_csv("THE_CountryCode_Result_1202_13.csv")

```

66 concatenate all files

```
In [22]: link =r"C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_{}.
```

```

for i in range(0, 12):
    i+=1
    print(link.format(i))

```

```

C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_1.csv
C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_2.csv
C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_3.csv

```

```

C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_4.csv
C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_5.csv
C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_6.csv
C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_7.csv
C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_8.csv
C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_9.csv
C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_10.csv
C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_11.csv
C:\Users\jchen148\THE Rankings\Report to Jane\THE_CountryCode_Result_1202_12.csv

```

```
In [5]: import pandas as pd
```

```
In [4]: chucks=[]
```

```

filename='THE_CountryCode_Result_1202_{}.csv'
for i in range(0, 13):
    i+=1
    print(filename.format(i))
    #     chucks.append(filename.format(i))

```

```

THE_CountryCode_Result_1202_1.csv
THE_CountryCode_Result_1202_2.csv
THE_CountryCode_Result_1202_3.csv
THE_CountryCode_Result_1202_4.csv
THE_CountryCode_Result_1202_5.csv
THE_CountryCode_Result_1202_6.csv
THE_CountryCode_Result_1202_7.csv
THE_CountryCode_Result_1202_8.csv
THE_CountryCode_Result_1202_9.csv
THE_CountryCode_Result_1202_10.csv
THE_CountryCode_Result_1202_11.csv
THE_CountryCode_Result_1202_12.csv
THE_CountryCode_Result_1202_13.csv

```

```
In [5]: import pandas as pd
```

```

filename='THE_CountryCode_Result_1202_{}.csv'

chucks=[]
for i in range(0, 13):
    i+=1
    chucks.append(pd.read_csv(filename.format(i)))

namedata=pd.concat(chucks, ignore_index=True)

namedata.head()

```



```
Out [5]:
```

	Unnamed: 0	University Name \
0	0	University of Oxford
1	1	Jet Propulsion Laboratory, California Institut...
2	2	California Institute of Technology
3	3	University of Cambridge
4	4	Stanford University

	University id	Country	Country Code
0	315091	United Kingdom	GBR
1	508092	United States	USA
2	508021	United States	USA
3	315068	United Kingdom	GBR
4	508219	United States	USA

```
In [6]: namedata.reset_index()
```

```
namedata2=namedata[:]
```

```
namedata2.head()
```

```
namedata2=namedata.iloc[:,1:] # delete the first column
```

```
namedata2.head()
```

```
Out [6]:
```

	University Name	University id \
0	University of Oxford	315091
1	Jet Propulsion Laboratory, California Institut...	508092
2	California Institute of Technology	508021
3	University of Cambridge	315068
4	Stanford University	508219

	Country	Country Code
0	United Kingdom	GBR
1	United States	USA
2	United States	USA
3	United Kingdom	GBR
4	United States	USA

```
In [137]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(3)

university_name=[]
university_id=[]
country=[]
```

```

countryCode=[]
df=pd.DataFrame()

for line in want_3[40:50]:
#     line=re.sub('[^A-Za-z0-9]+',' ', line)
#     query = "name(school)"
url= "https://api.elsevier.com/metrics/institution/search?query=name("{}")&sta
resp = requests.get(url.format(line), headers={'Accept':'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
parsed=json.dumps(resp.json(),
                   sort_keys=True,
                   indent=4, separators=(',', ': '))
result=json.loads(parsed)
data=result['results']
for i in data:
    if i is not None:
#     if data[0] is not None:
        countries=i['country']
        unames=i['name']
        uids=i['id']
        codes=i['countryCode']
        if (countries is not None):
            country.append(countries)
        else:
            country.append("")
        if (unames is not None):
            university_name.append(unames)
        else:
            university_name.append("")
        if (uids is not None):
            university_id.append(uids)
        else:
            university_id.append("")
        if (codes is not None):
            countryCode.append(codes)
        else:
            countryCode.append("")
df=pd.DataFrame({'University Name':university_name, 'University id':univ
df.to_csv("THE_CountryCode_Result_1202_6.csv")

```

```

In [7]: Uidlist=namedata2['University id']
        Uidlist.head()

```

```

Out[7]: 0    315091
        1    508092
        2    508021

```

```
3    315068
4    508219
Name: University id, dtype: int64
```

```
In [8]: import requests
import requests_oauthlib
import pandas as pd
import numpy as np
```

```
In [ ]: import time
time.sleep(2)
```

```
url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Citation'
```

```
resp = requests.get(url.format(line), headers={'Accept': 'application/json',
                                                'X-ELS-APIKey': "a464321ef5063d696ada17f8c159a44c"})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ': '))
result=json.loads(parsed)
```

```
In [9]: len(Uidlist)
```

```
Out[9]: 1272
```

```
In [125]: import requests
import requests_oauthlib
import json
import pandas as pd
import numpy as np
```

```
import time
time.sleep(2)
```

```
country=[]
countryCode=[]
Uid=[]
uname=[]
uri=[]
metric=[]
CitationCount2014=[]
CitationCount2015=[]
CitationCount2016=[]
CitationCount2017=[]
CitationCount2018=[]
```

```
url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Citation'
```

```

for uid in Uidlist[1270:]:
#     print(url.format(uid))
    resp = requests.get(url.format(uid), headers={'Accept': 'application/json',
                                                  'X-ELS-APIKey': "a464321ef5063d696ada17f8c159a44c"})
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ' : '))
    result=json.loads(parsed)
#     print(result)
    if 'results' not in result:
        pass
    else:
        if list(result['results']) is None:
            pass
        else:
#             if list(result['results'])[0] is None:
#                 pass
#             else:
#                 data=result['results']
                if len(list(result['results']))<1:
                    pass
                else:
                    if 'institution' not in list(result['results'])[0]:
                        pass
                    else:
                        if 'country' in result['results'][0]['institution']:
                            country.append(result['results'][0]['institution']['country'])
                        else:
                            country.append("")
                        if 'countryCode' in result['results'][0]['institution']:
                            countryCode.append(result['results'][0]['institution']['countryCode'])
                        else:
                            countryCode.append("")
                        if 'id' in result['results'][0]['institution']:
                            Uid.append(result['results'][0]['institution']['id'])
                        else:
                            Uid.append("")
                        if 'name' in result['results'][0]['institution']:
                            uname.append(result['results'][0]['institution']['name'])
                        else:
                            uname.append("")
                        if 'uri' in result['results'][0]:
                            uri.append(result['results'][0]['institution']['uri'])
                        else:
                            uri.append("")
                    if 'metrics' not in result['results'][0]:
                        pass

```

```

else:
    if 'metricType' not in result['results'][0]['metrics'][0]:
        pass
    else:
        metric.append(result['results'][0]['metrics'][0]['metricType'])
        if 'valueByYear' in result['results'][0]['metrics'][0]:
            if '2014' in result['results'][0]['metrics'][0]['valueByYear']:
                CitationCount2014.append(result['results'][0]['metrics'][0]['valueByYear']['2014'])
            else:
                CitationCount2014.append("")
            if '2015' in result['results'][0]['metrics'][0]['valueByYear']:
                CitationCount2015.append(result['results'][0]['metrics'][0]['valueByYear']['2015'])
            else:
                CitationCount2015.append("")
            if '2016' in result['results'][0]['metrics'][0]['valueByYear']:
                CitationCount2016.append(result['results'][0]['metrics'][0]['valueByYear']['2016'])
            else:
                CitationCount2016.append("")
            if '2017' in result['results'][0]['metrics'][0]['valueByYear']:
                CitationCount2017.append(result['results'][0]['metrics'][0]['valueByYear']['2017'])
            else:
                CitationCount2017.append("")
            if '2018' in result['results'][0]['metrics'][0]['valueByYear']:
                CitationCount2018.append(result['results'][0]['metrics'][0]['valueByYear']['2018'])
            else:
                CitationCount2018.append("")
        else:
            CitationCount2014.append("")
            CitationCount2015.append("")
            CitationCount2016.append("")
            CitationCount2017.append("")
            CitationCount2018.append("")

#         else:
#             metric.append("")

s1=pd.Series(country, name='Country')
s2=pd.Series(countryCode, name='CountryCode')
s3=pd.Series(Uid, name='Uid')
s4=pd.Series(uname, name='UniversityName')
s5=pd.Series(uri, name='uri')
s6=pd.Series(metric, name='metric')
s7=pd.Series(CitationCount2014, name='Citation2014')
s8=pd.Series(CitationCount2015, name='Citation2015')
s9=pd.Series(CitationCount2016, name='Citation2016')
s10=pd.Series(CitationCount2017, name='Citation2017')
s11=pd.Series(CitationCount2018, name='Citation2018')

Times_df=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11], axis=1)

```

```

Times_df.to_csv("Times_11.csv",index=False)
# df=pd.DataFrame(pd.DataFrame(result['results'][0]['metrics']))
# df.to_csv("1213_THE.csv", index=False)

```

67 Save data dictionary

```

In [38]: import requests
import requests_oauthlib
import json
import pandas as pd
import numpy as np

import time
time.sleep(2)

url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Citati

for uid in Uidlist[:5]:
#     print(url.format(uid))
    resp = requests.get(url.format(uid), headers={'Accept':'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})

#     parsed=json.dumps(resp.json(),
#                         sort_keys=True,
#                         indent=4, separators=(',', ': '))
#     print(parsed)
#     #result=json.loads(parsed)
with open("Uni_Metric_Data_Dictionary_Test4.json", 'w') as jsonfile:
    json.dump(resp.json(),
              sort_keys=True,
              indent=4, separators=(',', ': '), fp=jsonfile)

In [49]: import requests
import requests_oauthlib
import json
import pandas as pd
import numpy as np

import time
time.sleep(2)

url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Citati

for uid in Uidlist[:2]:
#     print(url.format(uid))
    resp = requests.get(url.format(uid), headers={'Accept':'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
    parsed=json.dumps(resp.json(),

```

```

        sort_keys=True,
        indent=4, separators=(',', ' '))
#     print(parsed)
#     parsed = json.loads(resp.text)
#     print(parsed)
    result=json.loads(parsed)
    print(result)
#     with open("Uni_Metric_Data_Dictionary_2.txt", 'a') as text_file:
#         print(parsed, file=text_file)
#         json.dump(resp.json(),
#                   sort_keys=True,
#                   indent=4, separators=(',', ' '), fp=jsonfile)

```

```

{'dataSource': {'lastUpdated': '2019-12-11', 'sourceName': 'Scopus'}, 'link': {'@href': 'https
{'dataSource': {'lastUpdated': '2019-12-11', 'sourceName': 'Scopus'}, 'link': {'@href': 'https

```

```

In [24]: import requests
import requests_oauthlib
import json
import pandas as pd
import numpy as np

import time
time.sleep(2)

url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Citati

for uid in Uidlist[25:]:
#     print(url.format(uid))
    resp = requests.get(url.format(uid), headers={'Accept': 'application/json',
                                                'X-ELS-APIKey': "dcfb521197bf15867d12c3c86c46c69b"})
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ' '))
#     print(parsed)
#     result=json.loads(parsed)
with open("Uni_Metric_Data_Dic_2.json", 'w') as jsonfile:
    json.dump(parsed, jsonfile)

```

```

In [125]: import requests
import requests_oauthlib
import json
import pandas as pd
import numpy as np

import time
time.sleep(2)

```

```

country=[]
countryCode=[]
Uid=[]
uname=[]
uri=[]
metric=[]
CitationCount2014=[]
CitationCount2015=[]
CitationCount2016=[]
CitationCount2017=[]
CitationCount2018=[]

url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Citat

for uid in Uidlist[1270:]:
#     print(url.format(uid))
    resp = requests.get(url.format(uid), headers={'Accept':'application/json',
                                                'X-ELS-APIKey': "a464321ef5063d696ada17f8c159a44c"})
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ': '))
    result=json.loads(parsed)
#     print(result)
    if 'results' not in result:
        pass
    else:
        if list(result['results']) is None:
            pass
        else:
#             if list(result['results'])[0] is None:
#                 pass
#             else:
#                 data=result['results']
#                 if len(list(result['results']))<1:
#                     pass
#                 else:
#                     if 'institution' not in list(result['results'])[0]:
#                         pass
#                     else:
#                         if 'country' in result['results'][0]['institution']:
#                             country.append(result['results'][0]['institution']['country'])
#                         else:
#                             country.append("")
#                         if 'countryCode' in result['results'][0]['institution']:
#                             countryCode.append(result['results'][0]['institution']['countryCode'])
#                         else:
#                             countryCode.append("")

```



```

        if 'id' in result['results'][0]['institution']:
            Uid.append(result['results'][0]['institution']['id'])
        else:
            Uid.append("")
        if 'name' in result['results'][0]['institution']:
            uname.append(result['results'][0]['institution']['name'])
        else:
            uname.append("")
        if 'uri' in result['results'][0]:
            uri.append(result['results'][0]['institution']['uri'])
        else:
            uri.append("")
    if 'metrics' not in result['results'][0]:
        pass
    else:
        if 'metricType' not in result['results'][0]['metrics'][0]:
            pass
        else:
            metric.append(result['results'][0]['metrics'][0]['metricType'])
            if 'valueByYear' in result['results'][0]['metrics'][0]:
                if '2014' in result['results'][0]['metrics'][0]['valueByYear']:
                    CitationCount2014.append(result['results'][0]['metrics'][0]['valueByYear']['2014'])
                else:
                    CitationCount2014.append("")
                if '2015' in result['results'][0]['metrics'][0]['valueByYear']:
                    CitationCount2015.append(result['results'][0]['metrics'][0]['valueByYear']['2015'])
                else:
                    CitationCount2015.append("")
                if '2016' in result['results'][0]['metrics'][0]['valueByYear']:
                    CitationCount2016.append(result['results'][0]['metrics'][0]['valueByYear']['2016'])
                else:
                    CitationCount2016.append("")
                if '2017' in result['results'][0]['metrics'][0]['valueByYear']:
                    CitationCount2017.append(result['results'][0]['metrics'][0]['valueByYear']['2017'])
                else:
                    CitationCount2017.append("")
                if '2018' in result['results'][0]['metrics'][0]['valueByYear']:
                    CitationCount2018.append(result['results'][0]['metrics'][0]['valueByYear']['2018'])
                else:
                    CitationCount2018.append("")
            else:
                CitationCount2014.append("")
                CitationCount2015.append("")
                CitationCount2016.append("")
                CitationCount2017.append("")
                CitationCount2018.append("")
        #
        #
    else:
        metric.append("")

```

```

s1=pd.Series(country, name='Country')
s2=pd.Series(countryCode, name='CountryCode')
s3=pd.Series(Uid, name='Uid')
s4=pd.Series(uname, name='UniversityName')
s5=pd.Series(uri, name='uri')
s6=pd.Series(metric, name='metric')
s7=pd.Series(CitationCount2014, name='Citation2014')
s8=pd.Series(CitationCount2015, name='Citation2015')
s9=pd.Series(CitationCount2016, name='Citation2016')
s10=pd.Series(CitationCount2017, name='Citation2017')
s11=pd.Series(CitationCount2018, name='Citation2018')

Times_df=pd.concat([s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11], axis=1)
Times_df.to_csv("Times_11.csv",index=False)
# df=pd.DataFrame(pd.DataFrame(result['results'][0]['metrics']))
# df.to_csv("1213_THE.csv", index=False)

```

68 Combine all subfiles

```
In [126]: filename='Times_{}.csv'
```

```

for i in range(1,12):
    print(filename.format(i))

```

```

Times_1.csv
Times_2.csv
Times_3.csv
Times_4.csv
Times_5.csv
Times_6.csv
Times_7.csv
Times_8.csv
Times_9.csv
Times_10.csv
Times_11.csv

```

```

In [127]: chuck=[]
for i in range(1,12):
    chuck.append(pd.read_csv(filename.format(i)))

total=pd.concat(chuck, ignore_index=True)

total.head()

```

```

Out[127]:
   Country CountryCode  Uid \
0  United Kingdom      GBR  315091

```

1	United States	USA	508092
2	United States	USA	508021
3	United Kingdom	GBR	315068
4	United States	USA	508219

	UniversityName	uri	metric	\
0	University of Oxford	NaN	CitationCount	
1	Jet Propulsion Laboratory, California Institut...	NaN	CitationCount	
2	California Institute of Technology	NaN	CitationCount	
3	University of Cambridge	NaN	CitationCount	
4	Stanford University	NaN	CitationCount	

	Citation2014	Citation2015	Citation2016	Citation2017	Citation2018
0	355751.0	313129.0	238271.0	149907.0	75747.0
1	39797.0	30488.0	32933.0	18670.0	9264.0
2	129593.0	97259.0	90365.0	58679.0	29486.0
3	260407.0	236169.0	197754.0	127682.0	64830.0
4	404346.0	363567.0	282005.0	194136.0	95069.0

```
In [128]: del total['uri']
```

```
In [129]: total.head()
```

```
total.to_csv("THE_Ranked_University_CitationCount_2014_2018.csv", index=False)
```

```
In [130]: total.head()
```

```
Out[130]:
```

	Country	CountryCode	Uid	\
0	United Kingdom	GBR	315091	
1	United States	USA	508092	
2	United States	USA	508021	
3	United Kingdom	GBR	315068	
4	United States	USA	508219	

	UniversityName	metric	\
0	University of Oxford	CitationCount	
1	Jet Propulsion Laboratory, California Institut...	CitationCount	
2	California Institute of Technology	CitationCount	
3	University of Cambridge	CitationCount	
4	Stanford University	CitationCount	

	Citation2014	Citation2015	Citation2016	Citation2017	Citation2018
0	355751.0	313129.0	238271.0	149907.0	75747.0
1	39797.0	30488.0	32933.0	18670.0	9264.0
2	129593.0	97259.0	90365.0	58679.0	29486.0
3	260407.0	236169.0	197754.0	127682.0	64830.0
4	404346.0	363567.0	282005.0	194136.0	95069.0

```
In [132]: ranked=total.sort_values(by='Citation2018', ascending=False)
```



```
In [11]: import json
```

```
In [22]: url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Citati
```

```
    #for uid in df_id['uid']:
    for item in Uidlist[100:]:
        #    query = "name(school)"
        #    url= "https://api.elsevier.com/metrics/institution/search?name({})&start=0&count=100"

        resp = requests.get(url.format(item), headers={'Accept':'application/json',
                                                       'X-ELS-APIKey': "ba88a424c653ea37282b6a4cdf423a1d"})
        parsed=json.dumps(resp.json(),
                           sort_keys=True,
                           indent=4, separators=(',', ': '))
        #    print(parsed)
        #    result=json.loads(parsed)
        with open("Data_Dic_1218_6.txt", "a") as text_file:
            print(parsed, file=text_file)
        #    result=parsed[2]
```

```
In [180]: import requests
           import json
           import pandas as pd
           import numpy as np
           from time import sleep
           sleep(2)
```

```
country=[]
countryCode=[]
universityid=[]
uniname=[]
metricType=[]
percentage2014=[]
percentage2015=[]
percentage2016=[]
percentage2017=[]
percentage2018=[]
value2014=[]
value2015=[]
value2016=[]
value2017=[]
value2018=[]
ScholarlyOutput2014=[]
ScholarlyOutput2015=[]
ScholarlyOutput2016=[]
ScholarlyOutput2017=[]
ScholarlyOutput2018=[]
CitationCount2014=[]
```

```

CitationCount2015=[]
CitationCount2016=[]
CitationCount2017=[]
CitationCount2018=[]
CitedPublicationsValue2014=[]
CitedPublicationsValue2015=[]
CitedPublicationsValue2016=[]
CitedPublicationsValue2017=[]
CitedPublicationsValue2018=[]
CitedPublicationspercentage2014=[]
CitedPublicationspercentage2015=[]
CitedPublicationspercentage2016=[]
CitedPublicationspercentage2017=[]
CitedPublicationspercentage2018=[]
impactType=[]
CiteScorepercentage2014=[]
CiteScorepercentage2015=[]
CiteScorepercentage2016=[]
CiteScorepercentage2017=[]
CiteScorepercentage2018=[]
CiteScorevalue2014=[]
CiteScorevalue2015=[]
CiteScorevalue2016=[]
CiteScorevalue2017=[]
CiteScorevalue2018=[]
PublicationsInTopJournalPercentilespercentage2014=[]
PublicationsInTopJournalPercentilespercentage2015=[]
PublicationsInTopJournalPercentilespercentage2016=[]
PublicationsInTopJournalPercentilespercentage2017=[]
PublicationsInTopJournalPercentilespercentage2018=[]
PublicationsInTopJournalPercentilesvalue2014=[]
PublicationsInTopJournalPercentilesvalue2015=[]
PublicationsInTopJournalPercentilesvalue2016=[]
PublicationsInTopJournalPercentilesvalue2017=[]
PublicationsInTopJournalPercentilesvalue2018=[]

PublicationsInTopJournalPercentByYear2014=[]
PublicationsInTopJournalPercentByYear2015=[]
PublicationsInTopJournalPercentByYear2016=[]
PublicationsInTopJournalPercentByYear2017=[]
PublicationsInTopJournalPercentByYear2018=[]

```

```

url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Citat

```

```

#for uid in df_id['uid']:
for item in inputdata:
#     query = "name(school)"
#     url= "https://api.elsevier.com/metrics/institution/search?name({})&start=0&coun

```

```

resp = requests.get(url.format(item), headers={'Accept': 'application/json',
                                                'X-ELS-APIKey': "a464321ef5063d696ada17f8c159a44c"})
parsed=json.dumps(resp.json(),
                  sort_keys=True,
                  indent=4, separators=(',', ' ': ))
#   print(parsed)
result=json.loads(parsed)
#   result=parsed[2]
data=result['results']
#   print(data[0])
for i in data:
    if i is None:
        pass
    else:
        if i['institution'] is None:
            pass
        else:
            if i['institution']['country'] is not None:
                country.append(i['institution']['country'])
            else:
                country.append("")
            if i['institution']['countryCode'] is not None:
                countryCode.append(i['institution']['countryCode'])
            else:
                countryCode.append("")
            if i['institution']['id'] is not None:
                universityid.append(i['institution']['id'])
            else:
                universityid.append("")
            if i['institution']['name'] is not None:
                uniname.append(i['institution']['name'])
            else:
                uniname.append("")
        if i['metrics'] is None:
            pass
        else:
            if i['metrics'][0] is None:
                pass
            else:
                if i['metrics'][0]['metricType'] is not None:
                    metricType.append(i['metrics'][0]['metricType'])
                else:
                    metricType.append("")
                if i['metrics'][0]['valueByYear'] is None:
                    pass
                else:
                    if i['metrics'][0]['valueByYear']['2014'] is not None:

```

```

        CitationCount2014.append(i['metrics'][0]['valueByYear'])
    else:
        CitationCount2014.append("")
    if i['metrics'][0]['valueByYear']['2015'] is not None:
        CitationCount2015.append(i['metrics'][0]['valueByYear'])
    else:
        CitationCount2015.append("")
    if i['metrics'][0]['valueByYear']['2016'] is not None:
        CitationCount2016.append(i['metrics'][0]['valueByYear'])
    else:
        CitationCount2016.append("")
    if i['metrics'][0]['valueByYear']['2017'] is not None:
        CitationCount2017.append(i['metrics'][0]['valueByYear'])
    else:
        CitationCount2017.append("")
    if i['metrics'][0]['valueByYear']['2018'] is not None:
        CitationCount2018.append(i['metrics'][0]['valueByYear'])
    else:
        CitationCount2018.append("")
if i['metrics'][1] is None:
    pass
else:
    if i['metrics'][1]['metricType'] is not None:
        metricType.append(i['metrics'][1]['metricType'])
    else:
        metricType.append("")
    if i['metrics'][1]['percentageByYear'] is None:
        pass
    else:
        if i['metrics'][1]['percentageByYear']['2014'] is not None:
            CitedPublicationspercentage2014.append(i['metrics'][1]['percentageByYear'])
        else:
            CitedPublicationspercentage2014.append("")
        if i['metrics'][1]['percentageByYear']['2015'] is not None:
            CitedPublicationspercentage2015.append(i['metrics'][1]['percentageByYear'])
        else:
            CitedPublicationspercentage2015.append("")
        if i['metrics'][1]['percentageByYear']['2016'] is not None:
            CitedPublicationspercentage2016.append(i['metrics'][1]['percentageByYear'])
        else:
            CitedPublicationspercentage2016.append("")
        if i['metrics'][1]['percentageByYear']['2017'] is not None:
            CitedPublicationspercentage2017.append(i['metrics'][1]['percentageByYear'])
        else:
            CitedPublicationspercentage2017.append("")
        if i['metrics'][1]['percentageByYear']['2018'] is not None:
            CitedPublicationspercentage2018.append(i['metrics'][1]['percentageByYear'])
        else:
            CitedPublicationspercentage2018.append("")

```



```

        CitedPublicationspercentage2018.append("")
    if i['metrics'][1]['valueByYear'] is None:
        pass
    else:
        if i['metrics'][1]['valueByYear']['2014'] is not None:
            CitedPublicationsValue2014.append(i['metrics'][1]['valueByYear']['2014'])
        else:
            CitedPublicationsValue2014.append("")
        if i['metrics'][1]['valueByYear']['2015'] is not None:
            CitedPublicationsValue2015.append(i['metrics'][1]['valueByYear']['2015'])
        else:
            CitedPublicationsValue2015.append("")
        if i['metrics'][1]['valueByYear']['2016'] is not None:
            CitedPublicationsValue2016.append(i['metrics'][1]['valueByYear']['2016'])
        else:
            CitedPublicationsValue2016.append("")
        if i['metrics'][1]['valueByYear']['2017'] is not None:
            CitedPublicationsValue2017.append(i['metrics'][1]['valueByYear']['2017'])
        else:
            CitedPublicationsValue2017.append("")
        if i['metrics'][1]['valueByYear']['2018'] is not None:
            CitedPublicationsValue2018.append(i['metrics'][1]['valueByYear']['2018'])
        else:
            CitedPublicationsValue2018.append("")
    if i['metrics'][2] is None:
        pass
    else:
        if i['metrics'][2]['metricType'] is not None:
            metricType.append(i['metrics'][2]['metricType'])
            ScholarlyOutput2014.append(i['metrics'][2]['valueByYear']['2014'])
        else:
            metricType.append("")
        if i['metrics'][2]['valueByYear'] is None:
            pass
        else:
            if i['metrics'][2]['valueByYear']['2014'] is not None:
                ScholarlyOutput2014.append(i['metrics'][2]['valueByYear']['2014'])
            else:
                ScholarlyOutput2014.append("")
            if i['metrics'][2]['valueByYear']['2015'] is not None:
                ScholarlyOutput2015.append(i['metrics'][2]['valueByYear']['2015'])
            else:
                ScholarlyOutput2015.append("")
            if i['metrics'][2]['valueByYear']['2016'] is not None:
                ScholarlyOutput2016.append(i['metrics'][2]['valueByYear']['2016'])
            else:
                ScholarlyOutput2016.append("")
            if i['metrics'][2]['valueByYear']['2017'] is not None:
                ScholarlyOutput2017.append(i['metrics'][2]['valueByYear']['2017'])
            else:
                ScholarlyOutput2017.append("")

```

```

        ScholarlyOutput2017.append(i['metrics'][2]['valueByYear'])
    else:
        ScholarlyOutput2017.append("")
    if i['metrics'][2]['valueByYear']['2018'] is not None:
        ScholarlyOutput2018.append(i['metrics'][2]['valueByYear'])
    else:
        ScholarlyOutput2018.append("")
if i['metrics'][3] is None:
    pass
else:
    if i['metrics'][3]['impactType'] is not None:
        impactType.append(i['metrics'][3]['impactType'])
    else:
        impactType.append("")
    if i['metrics'][3]['metricType'] is not None:
        metricType.append(i['metrics'][3]['metricType'])
    else:
        metricType.append("")
    if i['metrics'][3]['values'] is None:
        pass
    else:
        if i['metrics'][3]['values'][0]['percentageByYear'] is None:
            pass
        else:
            if i['metrics'][3]['values'][0]['percentageByYear']['2014'] is not None:
                CiteScorepercentage2014.append(i['metrics'][3]['valueByYear']['2014'])
            else:
                CiteScorepercentage2014.append("")
            if i['metrics'][3]['values'][0]['percentageByYear']['2015'] is not None:
                CiteScorepercentage2015.append(i['metrics'][3]['valueByYear']['2015'])
            else:
                CiteScorepercentage2015.append("")
            if i['metrics'][3]['values'][0]['percentageByYear']['2016'] is not None:
                CiteScorepercentage2016.append(i['metrics'][3]['valueByYear']['2016'])
            else:
                CiteScorepercentage2016.append("")
            if i['metrics'][3]['values'][0]['percentageByYear']['2017'] is not None:
                CiteScorepercentage2017.append(i['metrics'][3]['valueByYear']['2017'])
            else:
                CiteScorepercentage2017.append("")
            if i['metrics'][3]['values'][0]['percentageByYear']['2018'] is not None:
                CiteScorepercentage2018.append(i['metrics'][3]['valueByYear']['2018'])
            else:
                CiteScorepercentage2018.append("")
        if i['metrics'][3]['values'][0]['percentageByYear'] is None:
            pass
        else:
            if i['metrics'][3]['values'][0]['valueByYear']['2014'] is not None:

```

```

        CiteScorevalue2014.append(i['metrics'][3]['values'][0])
    else:
        CiteScorevalue2014.append("")
    if i['metrics'][3]['values'][0]['valueByYear']['2015'] != 0:
        CiteScorevalue2015.append(i['metrics'][3]['values'][0])
    else:
        CiteScorevalue2015.append("")
    if i['metrics'][3]['values'][0]['valueByYear']['2016'] != 0:
        CiteScorevalue2016.append(i['metrics'][3]['values'][0])
    else:
        CiteScorevalue2016.append("")
    if i['metrics'][3]['values'][0]['valueByYear']['2017'] != 0:
        CiteScorevalue2017.append(i['metrics'][3]['values'][0])
    else:
        CiteScorevalue2017.append("")
    if i['metrics'][3]['values'][0]['valueByYear']['2018'] != 0:
        CiteScorevalue2018.append(i['metrics'][3]['values'][0])
    else:
        CiteScorevalue2018.append("")

testfile= pd.DataFrame({'country': country, 'countryCode': countryCode, 'university': university,
                        'uniname':uniname, 'CitationCount2014':CitationCount2014,
                        'CitationCount2015':CitationCount2015, 'CitationCount2016':CitationCount2016,
                        'CitationCount2017':CitationCount2017, 'CitationCount2018':CitationCount2018,
                        'CitedPublicationspercentage2014':CitedPublicationspercentage2014,
                        'CitedPublicationspercentage2015':CitedPublicationspercentage2015,
                        'CitedPublicationspercentage2016':CitedPublicationspercentage2016,
                        'CitedPublicationspercentage2017':CitedPublicationspercentage2017,
                        'CitedPublicationspercentage2018':CitedPublicationspercentage2018,
                        'CitedPublicationsValue2014':CitedPublicationsValue2014,
                        'CitedPublicationsValue2015':CitedPublicationsValue2015,
                        'CitedPublicationsValue2016':CitedPublicationsValue2016,
                        'CitedPublicationsValue2017':CitedPublicationsValue2017,
                        'CitedPublicationsValue2018':CitedPublicationsValue2018,
                        'ScholarlyOutput2014':ScholarlyOutput2014, 'ScholarlyOutput2015':ScholarlyOutput2015,
                        'ScholarlyOutput2016': ScholarlyOutput2016, 'ScholarlyOutput2017':ScholarlyOutput2017,
                        'ScholarlyOutput2018':ScholarlyOutput2018,
                        'CiteScorepercentage2014':CiteScorepercentage2014,
                        'CiteScorepercentage2015':CiteScorepercentage2015,
                        'CiteScorepercentage2016':CiteScorepercentage2016,
                        'CiteScorepercentage2017':CiteScorepercentage2017,
                        'CiteScorepercentage2018':CiteScorepercentage2018,
                        'CiteScorevalue2014':CiteScorevalue2014,
                        'CiteScorevalue2015':CiteScorevalue2015,
                        'CiteScorevalue2016':CiteScorevalue2016,
                        'CiteScorevalue2017':CiteScorevalue2017,
                        'CiteScorevalue2018':CiteScorevalue2018})

```

```

testfile.to_csv("testfile_01.csv", index=False)

# data_dict = data[0]['institution']
# data_dict_2 = data[0]['institution']
# df_file_2=pd.DataFrame(data_dict_2)
# df_file_2.to_csv("File_3.csv", index=False)
# data_df=pd.DataFrame(data=data_dict.value())
# data_df.to_csv("File.csv", index=False)
# print(data[0]['institution']['name'])
# print(data[0]) # get 'MetricsType'
# inst=data[0]['institution']
# metrics=data[0]['metrics']
# df_test = pd.DataFrame({'institution':inst, 'metrics':metrics})
# df_test.to_csv("Test_Inst.csv", index=False)
# df=pd.DataFrame(data[0]['metrics'][0])
# df.to_csv("Test_MetricsType.csv", index=False)
# metrics=result[1]['metrics']

# print(data)
# print(data)
# df=pd.DataFrame(parsed)
# df.to_csv("Test_DataFrame.csv", index=False)

```

```

In [141]: import requests
import json
import pandas as pd
import numpy as np
from time import sleep
sleep(2)

```

```

country=[]
countryCode=[]
universityid=[]
uniname=[]
metricType=[]
percentage2014=[]
percentage2015=[]
percentage2016=[]
percentage2017=[]
percentage2018=[]
value2014=[]
value2015=[]
value2016=[]
value2017=[]
value2018=[]
ScholarlyOutput2014=[]
ScholarlyOutput2015=[]
ScholarlyOutput2016=[]

```

```

ScholarlyOutput2017=[]
ScholarlyOutput2018=[]
CitationCount2014=[]
CitationCount2015=[]
CitationCount2016=[]
CitationCount2017=[]
CitationCount2018=[]
CitedPublicationsValue2014=[]
CitedPublicationsValue2015=[]
CitedPublicationsValue2016=[]
CitedPublicationsValue2017=[]
CitedPublicationsValue2018=[]
CitedPublicationspercentage2014=[]
CitedPublicationspercentage2015=[]
CitedPublicationspercentage2016=[]
CitedPublicationspercentage2017=[]
CitedPublicationspercentage2018=[]
impactType=[]
CiteScorepercentage2014=[]
CiteScorepercentage2015=[]
CiteScorepercentage2016=[]
CiteScorepercentage2017=[]
CiteScorepercentage2018=[]
CiteScorevalue2014=[]
CiteScorevalue2015=[]
CiteScorevalue2016=[]
CiteScorevalue2017=[]
CiteScorevalue2018=[]
PublicationsInTopJournalPercentilespercentage2014=[]
PublicationsInTopJournalPercentilespercentage2015=[]
PublicationsInTopJournalPercentilespercentage2016=[]
PublicationsInTopJournalPercentilespercentage2017=[]
PublicationsInTopJournalPercentilespercentage2018=[]
PublicationsInTopJournalPercentilesvalue2014=[]
PublicationsInTopJournalPercentilesvalue2015=[]
PublicationsInTopJournalPercentilesvalue2016=[]
PublicationsInTopJournalPercentilesvalue2017=[]
PublicationsInTopJournalPercentilesvalue2018=[]

PublicationsInTopJournalPercentByYear2014=[]
PublicationsInTopJournalPercentByYear2015=[]
PublicationsInTopJournalPercentByYear2016=[]
PublicationsInTopJournalPercentByYear2017=[]
PublicationsInTopJournalPercentByYear2018=[]

url='https://api.elsevier.com/analytics/scival/institution/metrics?metricTypes=Citat.

#for uid in df_id['uid']:

```

```

for item in Uidlist[100:]:
#     query = "name(school)"
#     url= "https://api.elsevier.com/metrics/institution/search?name({})&start=0&count=100"

    resp = requests.get(url.format(item), headers={'Accept': 'application/json',
                                                    'X-ELS-APIKey': "a464321ef5063d696ada17f8c159a44c"})
    parsed=json.dumps(resp.json(),
                      sort_keys=True,
                      indent=4, separators=(',', ': '))
#     print(parsed)
    result=json.loads(parsed)
#     result=parsed[2]
    data=result['results']
#     print(data[0])
    for i in data:
        if i is None:
            pass
        else:
            if i['institution'] is None:
                pass
            else:
                if i['institution']['country'] is not None:
                    country.append(i['institution']['country'])
                else:
                    country.append("")
                if i['institution']['countryCode'] is not None:
                    countryCode.append(i['institution']['countryCode'])
                else:
                    countryCode.append("")
                if i['institution']['id'] is not None:
                    universityid.append(i['institution']['id'])
                else:
                    universityid.append("")
                if i['institution']['name'] is not None:
                    uniname.append(i['institution']['name'])
                else:
                    uniname.append("")
            if i['metrics'] is None:
                pass
            else:
                if i['metrics'][0] is None:
                    pass
                else:
                    if i['metrics'][0]['metricType'] is not None:
                        metricType.append(i['metrics'][0]['metricType'])
                    else:
                        metricType.append("")
                    if i['metrics'][0]['valueByYear'] is None:

```

```

        pass
    else:
        if i['metrics'][0]['valueByYear']['2014'] is not None:
            CitationCount2014.append(i['metrics'][0]['valueByYear'])
        else:
            CitationCount2014.append("")
        if i['metrics'][0]['valueByYear']['2015'] is not None:
            CitationCount2015.append(i['metrics'][0]['valueByYear'])
        else:
            CitationCount2015.append("")
        if i['metrics'][0]['valueByYear']['2016'] is not None:
            CitationCount2016.append(i['metrics'][0]['valueByYear'])
        else:
            CitationCount2016.append("")
        if i['metrics'][0]['valueByYear']['2017'] is not None:
            CitationCount2017.append(i['metrics'][0]['valueByYear'])
        else:
            CitationCount2017.append("")
        if i['metrics'][0]['valueByYear']['2018'] is not None:
            CitationCount2018.append(i['metrics'][0]['valueByYear'])
        else:
            CitationCount2018.append("")
    if i['metrics'][1] is None:
        pass
    else:
        if i['metrics'][1]['metricType'] is not None:
            metricType.append(i['metrics'][1]['metricType'])
        else:
            metricType.append("")
        if i['metrics'][1]['percentageByYear'] is None:
            pass
        else:
            if i['metrics'][1]['percentageByYear']['2014'] is not None:
                CitedPublicationspercentage2014.append(i['metrics'][1]['percentageByYear'])
            else:
                CitedPublicationspercentage2014.append("")
            if i['metrics'][1]['percentageByYear']['2015'] is not None:
                CitedPublicationspercentage2015.append(i['metrics'][1]['percentageByYear'])
            else:
                CitedPublicationspercentage2015.append("")
            if i['metrics'][1]['percentageByYear']['2016'] is not None:
                CitedPublicationspercentage2016.append(i['metrics'][1]['percentageByYear'])
            else:
                CitedPublicationspercentage2016.append("")
            if i['metrics'][1]['percentageByYear']['2017'] is not None:
                CitedPublicationspercentage2017.append(i['metrics'][1]['percentageByYear'])
            else:
                CitedPublicationspercentage2017.append("")

```

```

        if i['metrics'][1]['percentageByYear']['2018'] is not None:
            CitedPublicationspercentage2018.append(i['metrics'][1]['percentageByYear']['2018'])
        else:
            CitedPublicationspercentage2018.append("")
    if i['metrics'][1]['valueByYear'] is None:
        pass
    else:
        if i['metrics'][1]['valueByYear']['2014'] is not None:
            CitedPublicationsValue2014.append(i['metrics'][1]['valueByYear']['2014'])
        else:
            CitedPublicationsValue2014.append("")
        if i['metrics'][1]['valueByYear']['2015'] is not None:
            CitedPublicationsValue2015.append(i['metrics'][1]['valueByYear']['2015'])
        else:
            CitedPublicationsValue2015.append("")
        if i['metrics'][1]['valueByYear']['2016'] is not None:
            CitedPublicationsValue2016.append(i['metrics'][1]['valueByYear']['2016'])
        else:
            CitedPublicationsValue2016.append("")
        if i['metrics'][1]['valueByYear']['2017'] is not None:
            CitedPublicationsValue2017.append(i['metrics'][1]['valueByYear']['2017'])
        else:
            CitedPublicationsValue2017.append("")
        if i['metrics'][1]['valueByYear']['2018'] is not None:
            CitedPublicationsValue2018.append(i['metrics'][1]['valueByYear']['2018'])
        else:
            CitedPublicationsValue2018.append("")
    if i['metrics'][2] is None:
        pass
    else:
        if i['metrics'][2]['metricType'] is not None:
            metricType.append(i['metrics'][2]['metricType'])
            ScholarlyOutput2014.append(i['metrics'][2]['valueByYear']['2014'])
        else:
            metricType.append("")
        if i['metrics'][2]['valueByYear'] is None:
            pass
        else:
            if i['metrics'][2]['valueByYear']['2014'] is not None:
                ScholarlyOutput2014.append(i['metrics'][2]['valueByYear']['2014'])
            else:
                ScholarlyOutput2014.append("")
            if i['metrics'][2]['valueByYear']['2015'] is not None:
                ScholarlyOutput2015.append(i['metrics'][2]['valueByYear']['2015'])
            else:
                ScholarlyOutput2015.append("")
            if i['metrics'][2]['valueByYear']['2016'] is not None:
                ScholarlyOutput2016.append(i['metrics'][2]['valueByYear']['2016'])
            else:
                ScholarlyOutput2016.append("")

```



```

else:
    ScholarlyOutput2016.append("")
if i['metrics'][2]['valueByYear']['2017'] is not None:
    ScholarlyOutput2017.append(i['metrics'][2]['valueByYear'])
else:
    ScholarlyOutput2017.append("")
if i['metrics'][2]['valueByYear']['2018'] is not None:
    ScholarlyOutput2018.append(i['metrics'][2]['valueByYear'])
else:
    ScholarlyOutput2018.append("")
if i['metrics'][3] is None:
    pass
else:
    if i['metrics'][3]['impactType'] is not None:
        impactType.append(i['metrics'][3]['impactType'])
    else:
        impactType.append("")
    if i['metrics'][3]['metricType'] is not None:
        metricType.append(i['metrics'][3]['metricType'])
    else:
        metricType.append("")
    if i['metrics'][3]['values'] is None:
        pass
    else:
        if i['metrics'][3]['values'][0]['percentageByYear'] is None:
            pass
        else:
            if i['metrics'][3]['values'][0]['percentageByYear']['2014'] is not None:
                CiteScorepercentage2014.append(i['metrics'][3]['values'][0]['percentageByYear']['2014'])
            else:
                CiteScorepercentage2014.append("")
            if i['metrics'][3]['values'][0]['percentageByYear']['2015'] is not None:
                CiteScorepercentage2015.append(i['metrics'][3]['values'][0]['percentageByYear']['2015'])
            else:
                CiteScorepercentage2015.append("")
            if i['metrics'][3]['values'][0]['percentageByYear']['2016'] is not None:
                CiteScorepercentage2016.append(i['metrics'][3]['values'][0]['percentageByYear']['2016'])
            else:
                CiteScorepercentage2016.append("")
            if i['metrics'][3]['values'][0]['percentageByYear']['2017'] is not None:
                CiteScorepercentage2017.append(i['metrics'][3]['values'][0]['percentageByYear']['2017'])
            else:
                CiteScorepercentage2017.append("")
            if i['metrics'][3]['values'][0]['percentageByYear']['2018'] is not None:
                CiteScorepercentage2018.append(i['metrics'][3]['values'][0]['percentageByYear']['2018'])
            else:
                CiteScorepercentage2018.append("")
        if i['metrics'][3]['values'][0]['percentageByYear'] is None:

```

```

        pass
    else:
        if i['metrics'][3]['values'][0]['valueByYear']['2014'] != i['CiteScorevalue2014']:
            CiteScorevalue2014.append(i['metrics'][3]['values'][0]['valueByYear']['2014'])
        else:
            CiteScorevalue2014.append("")
        if i['metrics'][3]['values'][0]['valueByYear']['2015'] != i['CiteScorevalue2015']:
            CiteScorevalue2015.append(i['metrics'][3]['values'][0]['valueByYear']['2015'])
        else:
            CiteScorevalue2015.append("")
        if i['metrics'][3]['values'][0]['valueByYear']['2016'] != i['CiteScorevalue2016']:
            CiteScorevalue2016.append(i['metrics'][3]['values'][0]['valueByYear']['2016'])
        else:
            CiteScorevalue2016.append("")
        if i['metrics'][3]['values'][0]['valueByYear']['2017'] != i['CiteScorevalue2017']:
            CiteScorevalue2017.append(i['metrics'][3]['values'][0]['valueByYear']['2017'])
        else:
            CiteScorevalue2017.append("")
        if i['metrics'][3]['values'][0]['valueByYear']['2018'] != i['CiteScorevalue2018']:
            CiteScorevalue2018.append(i['metrics'][3]['values'][0]['valueByYear']['2018'])
        else:
            CiteScorevalue2018.append("")

testfile= pd.DataFrame({'country': country, 'countryCode': countryCode, 'university': university,
    'uname':uname, 'CitationCount2014':CitationCount2014,
    'CitationCount2015':CitationCount2015, 'CitationCount2016':CitationCount2016,
    'CitationCount2017':CitationCount2017, 'CitationCount2018':CitationCount2018,
    'CitedPublicationspercentage2014':CitedPublicationspercentage2014,
    'CitedPublicationspercentage2015':CitedPublicationspercentage2015,
    'CitedPublicationspercentage2016':CitedPublicationspercentage2016,
    'CitedPublicationspercentage2017':CitedPublicationspercentage2017,
    'CitedPublicationspercentage2018':CitedPublicationspercentage2018,
    'CitedPublicationsValue2014':CitedPublicationsValue2014,
    'CitedPublicationsValue2015':CitedPublicationsValue2015,
    'CitedPublicationsValue2016':CitedPublicationsValue2016,
    'CitedPublicationsValue2017':CitedPublicationsValue2017,
    'CitedPublicationsValue2018':CitedPublicationsValue2018,
    'ScholarlyOutput2014':ScholarlyOutput2014, 'ScholarlyOutput2015':ScholarlyOutput2015,
    'ScholarlyOutput2016': ScholarlyOutput2016, 'ScholarlyOutput2017':ScholarlyOutput2017,
    'ScholarlyOutput2018':ScholarlyOutput2018,
    'CiteScorepercentage2014':CiteScorepercentage2014,
    'CiteScorepercentage2015':CiteScorepercentage2015,
    'CiteScorepercentage2016':CiteScorepercentage2016,
    'CiteScorepercentage2017':CiteScorepercentage2017,
    'CiteScorepercentage2018':CiteScorepercentage2018,
    'CiteScorevalue2014':CiteScorevalue2014,
    'CiteScorevalue2015':CiteScorevalue2015,
    'CiteScorevalue2016':CiteScorevalue2016,
    'CiteScorevalue2017':CiteScorevalue2017,
    'CiteScorevalue2018':CiteScorevalue2018})

```

```

        'CiteScorevalue2017':CiteScorevalue2017,
        'CiteScorevalue2018':CiteScorevalue2018})

testfile.to_csv("1213_THE_4.csv", index=False)

# data_dict = data[0]['institution']
# data_dict_2 = data[0]['institution']
# df_file_2=pd.DataFrame(data_dict_2)
# df_file_2.to_csv("File_3.csv", index=False)
# data_df=pd.DataFrame(data=data_dict.value())
# data_df.to_csv("File.csv", index=False)
# print(data[0]['institution']['name'])
# print(data[0]) # get 'MetricsType'
# inst=data[0]['institution']
# metrics=data[0]['metrics']
# df_test = pd.DataFrame({'institution':inst, 'metrics':metrics})
# df_test.to_csv("Test_Inst.csv", index=False)
# df=pd.DataFrame(data[0]['metrics'][0])
# df.to_csv("Test_MetricsType.csv", index=False)
# metrics=result[1]['metrics']

# print(data)
# print(data)
# df=pd.DataFrame(parsed)
# df.to_csv("Test_DataFrame.csv", index=False)

```

In [185]: chuck=[]

```

filename='1213_THE_{}.csv'

for i in range(1,5):
    chuck.append(pd.read_csv(filename.format(i)))

total_df2=pd.concat(chuck, ignore_index=True)

total_df2.head()

total_df2.to_csv("Updated_Uni_Metrics.csv", index=False)

```

In [170]: chuck=[]

```

filename='1213_THE_{}.csv'

for i in range(1,5):
    chuck.append(pd.read_csv(filename.format(i)))

```

```
total_df=pd.concat(chuck, axis=1)
```

```
total_df.head()
```

```
changedtype=lambda x: int(x[:])
```

```
In [ ]: total_df.universityid.fillna(0)
```

```
In [182]: total_df.head()
```

```
Out[182]:
```

	country	countryCode	universityid	\	uniname	CitationCount2014	\	CitationCount2015	CitationCount2016	CitationCount2017	CitationCount2018	\	CitedPublicationspercentage2014	...	CiteScorepercentage2014	\	CiteScorepercentage2015	CiteScorepercentage2016	CiteScorepercentage2017	\	CiteScorepercentage2018	CiteScorevalue2014	CiteScorevalue2015	\
0	United Kingdom	GBR	315091.0		University of Oxford	355751.0		313129.0	238271.0	149907.0	75747.0		86.504585	...	3.975581		2.919932	3.363425	3.181687		3.027766	267.0	190.0	
1	United States	USA	508092.0		Jet Propulsion Laboratory, California Institut...	39797.0		30488.0	32933.0	18670.0	9264.0		78.933620	...	5.387561		4.998725	5.137615	4.487027					
2	United States	USA	508021.0		California Institute of Technology	129593.0		97259.0	90365.0	58679.0	29486.0		85.711120	...	7.383966		7.643185	6.673326	6.509327					
3	United Kingdom	GBR	315068.0		University of Cambridge	260407.0		236169.0	197754.0	127682.0	64830.0		87.955536	...	3.898226		3.769248	4.672479	4.187925					
4	United States	USA	508219.0		Stanford University	404346.0		363567.0	282005.0	194136.0	95069.0		87.081640	...	6.278291		5.832855	5.982447	5.228088					

1	5.246510	408.0	392.0
2	6.352154	350.0	383.0
3	4.584467	167.0	164.0
4	5.670861	319.0	305.0

	CiteScorevalue2016	CiteScorevalue2017	CiteScorevalue2018
0	231.0	221.0	217.0
1	420.0	377.0	481.0
2	334.0	342.0	342.0
3	209.0	197.0	219.0
4	334.0	306.0	347.0

[5 rows x 136 columns]

```
In [183]: total_df.to_csv("Updated_THE_Uni_Metrics.csv", index=False)
```

```
In [26]: from sklearn.cluster import KMeans
```

```
In [11]: import requests
```

```
url = "https://api.elsevier.com/metrics/institution/scopus_id/60027165?apiKey=dcfb521
#url = "https://api.elsevier.com/content/abstract/scopus_id/60027165?apiKey=2bbd32fdf
response = requests.get(url)
print(response.headers)
```

```
{'allow': 'GET', 'Content-Encoding': 'gzip', 'Content-Type': 'text/html; charset=utf-8', 'Date':
```

```
In [36]: ## read in all the spreadsheets
```

```
import pandas as pd
```

```
First_5 = pd.read_csv(r"C:\Users\jchen148\THE Rankings\THE_Uni_First5.csv", delimiter=
print(type(First_5))
```

```
Start_6 = pd.read_csv(r"C:\Users\jchen148\THE Rankings\THE_Uni_6.csv", delimiter=",")
```

```
Start_11 = pd.read_csv(r"C:\Users\jchen148\THE Rankings\THE_Uni_11.csv", delimiter=",")
```

```
Start_311 = pd.read_csv(r"C:\Users\jchen148\THE Rankings\THE_Uni_311.csv", delimiter=
```

```
combined_df = pd.concat([First_5, Start_6, Start_11, Start_311])
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
In [25]: cd "C:\Users\jchen148\THE Rankings\Json files"
```

C:\Users\jchen148\THE Rankings\Json files

```
In [27]: import json
```

```
    with open("Test_THE_Country", 'w') as fd:
        fd.write(json.dumps(data_loaded, sort_keys=True, indent=4, separators=(',', ': ')))
```

```
In [29]: with open("Test_THE_Country", 'r') as fd:
        University_data=json.load(fd)
```

```
In [13]: # University SciVal institution id
        print(u_id)
```

```
['203212', '203166', '706710', '203440', '209064', '207019', '331008', '207104', '217090', '21
```

```
In [14]: # countryCode
```

```
        print(country)
```

```
['BWA', 'IND', 'KEN', 'ARE', 'ARE', 'CHN', 'SDN', 'IND', 'CHN', 'IND', 'ARE', 'TWN']
```