National Institute of Technology, Karnataka

Analysis of Co-Authorship network using Scopus databases

Presented by-

- Gavali Deshabhakt Naganath
- Mohammad Ahsan
- Mayur Anil Shimpi

Course Instructor: Dr. Pushparaj Shetty
Subject: Big Data Analytics
Specialization: CDS
Date: 14/04/2021

AGENDA

- Downloading Database
- Data Preprocessing
- Generating files necessary for analysis
- Finding answers

DOWNLOADING DATABASE

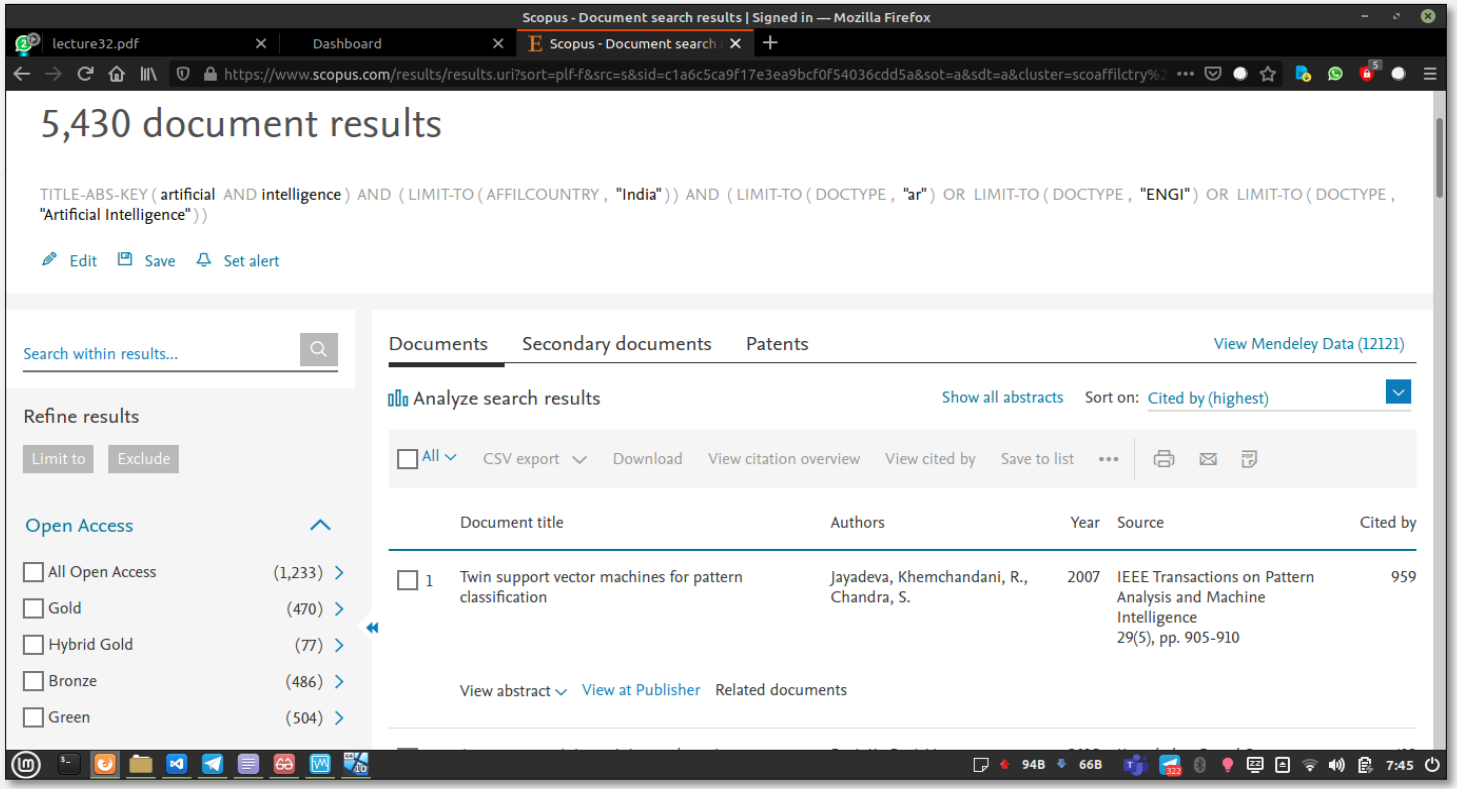
- Scopus
- Keyword – Artificial intelligence
- Document type – Article
- Subject area – Engineering

Query:

TITLE-ABS-KEY(Artificial Intelligence) AND (LIMIT-TO (DOCTYPE,"ar") OR LIMIT-TO (DOCTYPE,"ENGI") OR LIMIT-TO (DOCTYPE,"Artificial Intelligence"))

DOWNLOADING INDIA'S DATABASE

TITLE-ABS-KEY(Artificial Intelligence) AND (LIMIT-TO (AFFILCOUNTRY,"India")) AND (LIMIT-TO (DOCTYPE,"ar") OR LIMIT-TO (DOCTYPE,"ENGI") OR LIMIT-TO (DOCTYPE,"Artificial Intelligence"))



SETTING PATH OF DATA DIRECTORY

```
[ ]  1  dir_lst = []
     2  for data_file in os.listdir(base_dir):
     3      dir_lst.append(os.path.join(base_dir,data_file))
```

```
[ ]  1  dir_lst
```

```
['/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-Australia.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-Canada.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-China.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-France.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-Germany.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-India.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-Iran.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-Italy.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-Japan.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-Netherlands.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-SouthKorea.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-Spain.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-Taiwan.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-UK.csv',
 '/mnt/General_Stuff/Study Stuff/Documents/CDS/Sem-II/BigDataAnalytics/BigData-Programs/Mini-Project/Data/Artificial-Intelligence-US.csv']
```

IMPORTING AND PREPROCESSING COUNTRY-WISE DATABASE

```
country_lst = ['Australia','Canada','China','France','Germany','India','Iran','Italy','Japan','Netherlands','South Korea','Spain',
'Taiwan','United Kingdom', 'United States']
df_lst = []

for data_file, country in zip(dir_lst, country_lst):
        print(country)
        df_tmp = pd.read_csv(data_file)

        df_tmp = df_tmp.drop(['Author(s) ID' ,'Source title','Volume', 'Issue', 'Art. No.', 'Page start', 'Page end', 'Page count',
'DOI', 'Link', 'Document Type', 'Publication Stage','Open Access', 'Source', 'EID'],axis='columns')

        df_tmp = df_tmp.fillna(0)
        df_tmp['Country'] = country
        df_lst.append(df_tmp)
```

```
Australia
Canada
China
France
Germany
India
Iran
Italy
Japan
Netherlands
South Korea
Spain
Taiwan
United Kingdom
United States
```

FORMING MAIN DATAFRAME

```
df = pd.concat(df_lst)
```

```
df.shape
```

```
(67694, 5)
```

```
df.head()
```

	Authors	Title	Year	Cited by	Country
0	Soares, J.V.B., Leandro, J.J.G., Cesar Jr., R....	Retinal vessel segmentation using the 2-D Gabo...	2006	1083.0	Australia
1	Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuch...	The graph neural network model	2009	1031.0	Australia
2	Karantonis, D.M., Narayanan, M.R., Mathie, M.,...	Implementation of a real-time human movement c...	2006	908.0	Australia
3	Mirjalili, S.	Dragonfly algorithm: a new meta-heuristic opti...	2016	865.0	Australia
4	Naseem, I., Togneri, R., Bennamoun, M.	Linear regression for face recognition	2010	768.0	Australia

APPLYING SAME DATA PIPELINE TO DATABASE WITH SPONSORSHIP COLUMN

```
df_addon = pd.concat(df_lst)
```

```
df_addon.head()
```

	Authors	Title	Year	Cited by	Funding Details	Country
0	Tao F., Qi Q., Liu A., Kusiak A.	Data-driven smart manufacturing	2018	375.0	National Natural Science Foundation of China\n...	Australia
1	Zhang K., Gao X., Tao D., Li X.	Single image super-resolution with non-local m...	2012	362.0	National Natural Science Foundation of China\n...	Australia
2	Kristan M., Matas J., Leonardis A., Vojir T., ...	A Novel Performance Evaluation Methodology for...	2016	264.0	Seventh Framework Programme	Australia
3	Ding C., Choi J., Tao D., Davis L.S.	Multi-Directional Multi-Level Dual-Cross Patte...	2016	243.0	National Science Foundation\n\nAustralian Rese...	Australia
4	Celebi M.E., Kingravi H.A., Iyatomi H., Asland...	Border detection in dermoscopy images using st...	2008	241.0	National Cancer Institute	Australia

UPDATING FUNDING DETAILS OF MAIN DATAFRAME

```
df['Funding_Details'] = 0

titles_old = list(df.Title)
titles_new = list(df_addon.Title)

count = 1
for i in range(len(titles_new)):
    for j in range(len(titles_old)):
        if titles_new[i]==titles_old[j]:
            df.iloc[j,-1] = df_addon.iloc[i,-2]
    print(f'{count}\t{titles_new[i]}')
    count += 1
```

SAVING DATABASE TO CSV FILE

```
df.head()
```

	Unnamed: 0	Authors	Title	Year	Cited by	Country	Funding Details
0	0	Soares, J.V.B., Leandro, J.J.G., Cesar Jr., R....	Retinal vessel segmentation using the 2-D Gabo...	2006	1083.0	Australia	0
1	1	Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuch...	The graph neural network model	2009	1031.0	Australia	0
2	2	Karantonis, D.M., Narayanan, M.R., Mathie, M.,...	Implementation of a real-time human movement c...	2006	908.0	Australia	0
3	3	Mirjalili, S.	Dragonfly algorithm: a new meta-heuristic opti...	2016	865.0	Australia	0
4	4	Naseem, I., Togneri, R., Bennamoun, M.	Linear regression for face recognition	2010	768.0	Australia	0

```
df.to_csv('Complete_database.csv')
```

Authors	Title	Year	Cited by	Country	Funding_Details
Soares, J.V.B., Leandro, J.J.G., Cesar Jr	Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification	2006	1083	Australia	0
Scarselli, F., Gori, M., Tsoi, A.C., Hager	The graph neural network model	2009	1031	Australia	0
Karantonis, D.M., Narayanan, M.R., Mi	Implementation of a real-time human movement classifier using a triaxial accelerometer for	2006	908	Australia	0
Mirjalili, S.	Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objectiv	2016	865	Australia	0
Naseem, I., Togneri, R., Bennamoun, I	Linear regression for face recognition	2010	768	Australia	0
Geng, X., Zhou, Z.-H., Smith-Miles, K.	Automatic age estimation based on facial aging patterns	2007	675	Australia	0
Chen, Y., Zhao, X., Jia, X.	Spectral-Spatial Classification of Hyperspectral Data Based on Deep Belief Network	2015	599	Australia	0
Phung, S.L., Bouzerdoum, A., Chai, D.	Skin segmentation using color pixel classification: Analysis and comparison	2005	594	Australia	0
Li, X., Yao, X.	Cooperatively coevolving particle swarms for large scale optimization	2012	508	Australia	0
Dissanayake, S.D., Armstrong, J.	Comparison of ACO-OFDM, DCO-OFDM and ADO-OFDM in IM/DD systems	2013	427	Australia	0
Ong, Y.-S., Lim, M.-H., Zhu, N., Wong,	Classification of adaptive memetic algorithms: A comparative study	2006	425	Australia	0
Mian, A.S., Bennamoun, M., Owens, R	An efficient multimodal 2D-3D hybrid approach to automatic face recognition	2007	403	Australia	0
Tao, F., Qi, Q., Liu, A., Kusiak, A.	Data-driven smart manufacturing	2018	373	Australia	National Natural Science
Mian, A.S., Bennamoun, M., Owens, R	Three-dimensional model-based object recognition and segmentation in cluttered scenes	2006	364	Australia	0
Zhang, K., Gao, X., Tao, D., Li, X.	Single image super-resolution with non-local means and steering kernel regression	2012	362	Australia	National Natural Science
Hong, C., Yu, J., Wan, J., Tao, D., Wang	Multimodal Deep Autoencoder for Human Pose Recovery	2015	354	Australia	0

GENERATING FILES NECESSARY FOR FURTHER ANALYSIS

```
df_without_countries = df.drop('Country',axis='columns')
```

```
df_without_countries = df_without_countries.drop_duplicates(subset=['Title'],keep='first')
```

```
df_without_countries.shape
```

```
(59215, 5)
```

```
authors_lst = list(df_without_countries.loc[:,'Authors'].values)
```

```
print(authors_lst[0])
```

```
Soares, J.V.B., Leandro, J.J.G., Cesar Jr., R.M., Jelinek, H.F., Cree, M.J.
```

```
len(authors_lst)
```

```
59215
```

SEPARATING AUTHOR NAMES

```
set_authors = ['Jayadeva', 'Khemchandani, R.', 'Chandra, S.']
for i in range(len(authors_lst)):
    authors_sub_lst = authors_lst[i].split(',')
    authors_sub_lst_mod = []
    if authors_sub_lst == 'Jayadeva, Khemchandani, R., Chandra, S.'.split(','):
        continue
    for j in range(0,len(authors_sub_lst)-1,2):
        authors_sub_lst_mod.append(authors_sub_lst[j].strip()+','+authors_sub_lst[j+1])

    for author in authors_sub_lst_mod:
        if(author not in set_authors):
            set_authors.append(author)
            print(f'{i}\t{author}')
```

```
len(set_authors)
```

```
120158
```

STORING AUTHOR NAMES TO TEXT FILE

```
with open('Authors_list.txt','w') as filehandle:
        filehandle.writelines("%s\n" % author for author in set_authors)
filehandle.close()
```

READING AUTHOR NAMES FROM TEXT FILE

```
set_authors = []
with open('Authors_list.txt','r') as filehandle:
    filecontents = filehandle.readlines()

    for line in filecontents:
        # remove linebreak which is the last character of the string
        author = line[:-1]

        # add item to the list
        set_authors.append(author)
```

```
len(set_authors)
```

```
120158
```

CREATING AUTHOR: DATABASE DICTIONARY

```
# '''
dct_author_database = {}
count=0
for author in set_authors:
    print(f'{count}\t{author}')

    df_auth = pd.DataFrame(columns = ['Authors', 'Title', 'Year', 'Cited_by','Funding Details'] )

    filt= df_without_countries['Authors'].str.contains(author, na=False)
    df_auth= df_without_countries.loc[filt,'Authors':].reset_index(drop=True)

    dct_author_database[author] = df_auth

    count += 1
# '''
```

```
print(set_authors[10])
dct_author_database[set_authors[10]]
```

```
Tsoi, A.C.
```

	Authors	Title	Year	Cited_by	Funding Details
0	Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuch...	The graph neural network model	2009	1031.0	0
1	Shilton, A., Palaniswami, M., Ralph, D., Tsoi,...	Incremental training of support vector machines	2005	159.0	0
2	Scarselli, F., Tsoi, A.C., Hagenbuchner, M., N...	Solving graph data issues using a layered arch...	2013	10.0	0
3	Scarselli, F., Tsoi, A.C., Hagenbuchner, M.	The Vapnik–Chervonenkis dimension of graph and...	2018	4.0	0
4	Pucci, A., Gori, M., Hagenbuchner, M., Scarsel...	Investigations into the application of Graph N...	2006	3.0	0

ALTERNATIVE CODE

```
dct_author_database = {}

count = 0
for author in set_authors:
    print(f'{count}\t{author}')

    df_auth = pd.DataFrame(columns = ['Authors', 'Title', 'Year', 'Cited_by'] )

    for authors in authors_lst:

        if author in authors:
            df_tmp = df_without_countries[df_without_countries.Authors==authors]
            df_auth = df_auth.append(df_tmp,ignore_index=True)

    dct_author_database[author] = df_auth

    count += 1
```

STORING DICTIONARY TO TEXT FILE

```
with open('Author_database_dictonary.txt','wb') as file:
    file.write(pickle.dumps(dct_author_database))
file.close()
```

READING DICTIONARY FROM TEXT FILE

```
dct_author_database_from_file = {}
with open('Author_database_dictonary.txt','rb') as file:
    dct_author_database_from_file = pickle.load(file)
file.close()
```

```
print(type(dct_author_database_from_file))
```

```
<class 'dict'>
```

SEPARATING INDIAN AUTHORS NAMES

```
authors_from_ind_database = list(df[df.Country=='India']['Authors'].unique())

set_of_authors_from_indian_database = ['Jayadeva', 'Khemchandani, R.', 'Chandra, S.']

count = 0
for authors in authors_from_ind_database:

    authors_sub_lst = authors.split(',')

    if authors_sub_lst == 'Jayadeva, Khemchandani, R., Chandra, S.'.split(','):
        continue

    authors_sub_lst_mod = []
    for i in range(0,len(authors_sub_lst)-1,2):
            authors_sub_lst_mod.append(authors_sub_lst[i].strip()+','+authors_sub_lst[i+1])

    for author in authors_sub_lst_mod:
        if(author not in set_of_authors_from_indian_database):
            set_of_authors_from_indian_database.append(author)
    print(f'{count}\t{authors}')
    count += 1
```

SEPARATING FOREIGN AUTHORS NAMES

```
set_of_foreign_authors = []

count = 0
for author in set_authors:
    if author not in set_of_indian_authors_from_file:
        set_of_foreign_authors.append(author)
        print(f'{count}\t{author}')
        count += 1
```

GENERATING AUTHOR :PUBLICATION COUNT WITH INDIAN AUTHORS DICTIONARY

```
dct_foreign_author_coauth_count = {}
for foreign_author in set_of_foreign_authors_from_file:
    dct_foreign_author_coauth_count[foreign_author] = 0

count = 0
for foreign_author in set_of_foreign_authors_from_file:
    row_foreign_auth_list = dct_author_database_from_file[foreign_author].iloc[:,0].values
    for indian_author in set_of_indian_authors_from_file:
        for authors in row_foreign_auth_list:
            if indian_author in authors:
                dct_foreign_author_coauth_count[foreign_author] += 1
    print(f'{count}     {foreign_author}        {dct_foreign_author_coauth_count[foreign_author]}')
    count += 1
```

1. HIGHEST CITED AUTHOR AND HIS H-INDEX (FROM THE WORLD)

```
author_with_highest_citations = ""
max_citations = 0
for author in set_authors:
    cites = dct_author_database[author]['Cited_by'].sum()
    if max_citations<cites:
        author_with_highest_citations = author
        max_citations = cites
```

```
df_of_highest_cited_author = dct_author_database[author_with_highest_citations]
rows,cols = df_of_highest_cited_author.shape
avg_citations_of_author_with_highest_citations = max_citations/rows
h_index = min(rows,avg_citations_of_author_with_highest_citations)
```

```
print(f'Max cited author    = {author_with_highest_citations}')
print(f'Total cited by      = {max_citations}')
print(f'His h-index         = {h_index}')
```

```
Max cited author    = Hassabis, D.
Total cited by      = 17466.0
His h-index         = 13
```

2. HIGHEST PUBLICATION AUTHOR

```
author_with_highest_publication = ""
max_publication_count = 0
for author in set_authors:
    rows, columns = dct_author_database[author].shape
    if rows>max_publication_count:
        max_publication_count=rows
        author_with_highest_publication = author
        # print(f'{author}                    \t{rows}')
```

```
print(f'Author\t\t= {author_with_highest_publication}\nPublications\t= {max_publication_count}')
```

```
Author          = Wang, Y.
Publications    = 439
```

3. HIGHEST CITED AUTHORS AVERAGE CITATIONS, AND THE COUNTRY NAME

```
   rows, cols = dct_author_database[author_with_highest_citations].shape

   avg_citations_of_author_with_highest_citations = max_citations/rows
   print(f'Highest Cited Author \t= {author_with_highest_citations}')
   print(f'His Average Citations \t= {avg_citations_of_author_with_highest_citations}')
   print(f'His Total Publications \t= {rows}')
```

```
Highest Cited Author    = Hassabis, D.
His Average Citations   = 1343.5384615384614
His Total Publications  = 13
```

A google search with above author's name tells us that he's from 'United Kingdom' and doing research on 'Artificial Intelligence'

4. TOTAL NUMBER OF PUBLICATIONS OF THE HIGHEST CITED AUTHOR

```
print(f'Highest Cited Author \t= {author_with_highest_citations}')
print(f'Total Publications \t= {rows}')
dct_author_database[author_with_highest_citations]
```

```
Highest Cited Author    = Hassabis, D.
Total Publications      = 13
```

5. TOTAL PUBLICATION IN YEAR

```
year_lst = sorted(list(df['Year'].unique()))
country_lst = list(df['Country'].unique())
```

```
df_without_duplicates = df.drop_duplicates(subset=['Authors','Title'],keep='first')
```

```
dct_df_per_year_publications = {}
for year in year_lst:
    dct_df_per_year_publications[year], cols = df_without_duplicates[df_without_duplicates.Year==year].shape
```

```
dct_df_per_year_publications = sort_dict(dct_df_per_year_publications,'Values','reverse')
dct_df_per_year_publications
```

*Output in next slide

{2020: 6222,
2019: 4688,
2018: 4122,
2016: 3341,
2014: 3159,
2015: 3117,
2017: 3049,
2021: 2688,
2013: 2277,
2004: 2191,
2012: 2010,
2008: 1933,
2009: 1565,
2011: 1559,
2010: 1542,

2006: 1464,
2007: 1401,
2003: 1357,
2005: 1346,
2000: 793,
1989: 734,
1999: 730,
2001: 726,
1997: 718,
1996: 713,
1994: 703,
2002: 702,
1998: 697,
1995: 601,
1988: 531,

1991: 394,
1992: 357,
1987: 351,
1986: 165,
1985: 117,
1984: 70,
1983: 25,
1982: 23,
1977: 17,
1980: 13,
1973: 12,
1978: 12,
1979: 12,
1993: 485,
1990: 457,

1981: 11,
1974: 8,
1975: 8,
1976: 8,
1971: 7,
1972: 6,
1970: 3,
1962: 1,
1963: 1,
1964: 1,
1965: 1,
1968: 1,
1969: 1}

6. TOTAL CITATION PER YEAR

```
dct_citations_per_year = {}
for year in year_lst:
    dct_citations_per_year[year] = df_without_duplicates[df_without_duplicates.Year==year]['Cited by'].sum()

dct_citations_per_year = sort_dict(dct_citations_per_year,'Values','reverse')
```

{2008: 86548.0,
2016: 85426.0,
2015: 81343.0,
2005: 78325.0,
2004: 75544.0,
2007: 73896.0,
2006: 73652.0,
2014: 69065.0,
2009: 68433.0,
2018: 66871.0,
2013: 63245.0,
2017: 61390.0,

2010: 61118.0,
2012: 56435.0,
2011: 55280.0,
2019: 42281.0,
2000: 36085.0,
2003: 32783.0,
1999: 28128.0,
1998: 26193.0,
2002: 25418.0,
2001: 25176.0,
1997: 21638.0,
2020: 21386.0,

1995: 18774.0,
1994: 18694.0,
1996: 16943.0,
1989: 12072.0,
1992: 10540.0,
1990: 9800.0,
1993: 8807.0,
1991: 8453.0,
1988: 8385.0,
1987: 5439.0,
1986: 3652.0,
1980: 2616.0,

1985: 1930.0,
1977: 1644.0,
2021: 1568.0,
1979: 997.0,
1984: 812.0,
1971: 446.0,
1973: 256.0,
1978: 252.0,
1976: 215.0,
1983: 208.0,
1982: 202.0,
1975: 186.0,

1970: 83.0,
1981: 76.0,
1972: 49.0,
1974: 33.0,
1963: 17.0,
1962: 16.0,
1969: 4.0,
1964: 2.0,
1965: 2.0,
1968: 0.0}

7. AUTHOR(COUNTRY) HAVING HIGHEST CO-AUTHORSHIP WITH INDIAN AUTHORS

```
mx_pubs_with_indian_authors = 0
foreign_auth_corr_to_mx_pubs_with_indian_authors = ""
for author in set_of_foreign_authors:
    pubs = dct_foreign_author_coauth_count[author]
    if mx_pubs_with_indian_authors < pubs:
        mx_pubs_with_indian_authors = pubs
        foreign_auth_corr_to_mx_pubs_with_indian_authors = author
```

```
print(f'Foreign author with Highest Co-authorship with Indian Authors =  {foreign_auth_corr_to_mx_pubs_with_indian_authors}')
print(mx_pubs_with_indian_authors)
```

```
Foreign author with Highest Co-authorship with Indian Authors =  Nicolaides, A.
127
```

A google search with above authors name tells us that he is from "Vascular Screening and Diagnostic Centre, University of Nicosia, Nicosia, Cyprus"

8. HIGHEST CITED AUTHOR FROM INDIA AND THE UNIVERSITY

```
max_cites_of_indian_author = 0
highest_cited_indian_author = ""

for author in set_of_indian_authors:
    cites = dct_author_database[author]['Cited_by'].sum()
    if max_cites_of_indian_author < cites:
        max_cites_of_indian_author = cites
        highest_cited_indian_author = author
```

```
print(f'Highest Cited Author from India = {highest_cited_indian_author}')
print(f'His Total Citations = {max_cites_of_indian_author}')
```

```
Highest Cited Author from India = Raghava, G.P.
His Total Citations = 3132.0
```

Raghava G.P. is a Professor & Head at Department of Computational Biology, Indraprastha Institute of Information Technology (IIIT-Delhi), India

9. COMPARATIVE YEAR WISE ARTICLE PUBLICATION ANALYSIS OF INDIA, CHINA AND USA

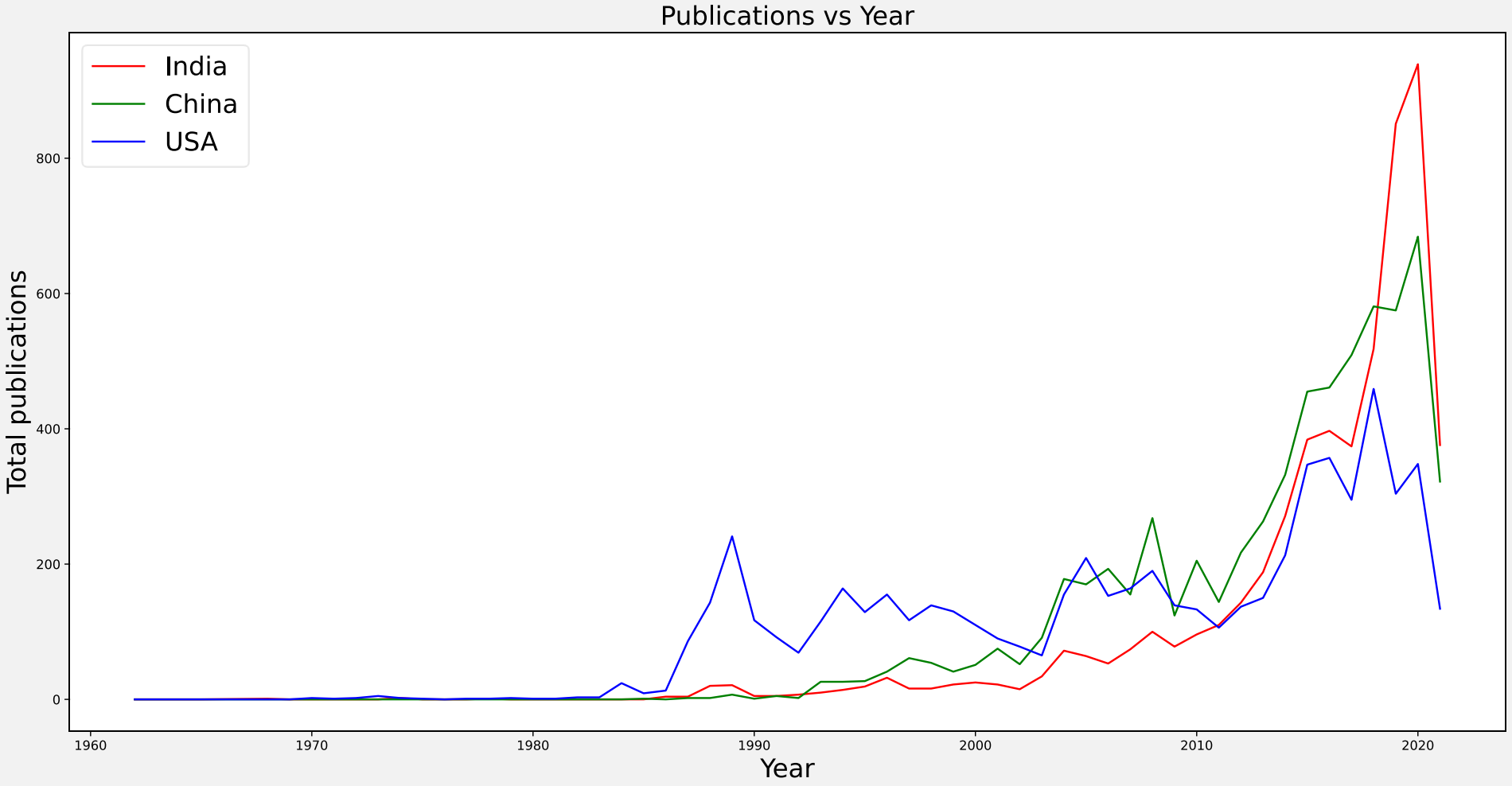
```
df_india = df[df.Country=='India'].copy().reset_index(drop=True)
df_china = df[df.Country=='China'].copy().reset_index(drop=True)
df_usa = df[df.Country=='United States'].copy().reset_index(drop=True)
```

```
dct_india_year_publications = {}
dct_china_year_publications = {}
dct_usa_year_publications = {}
for year in year_lst:
    dct_india_year_publications[year], cols1 = df_india[df_india.Year==year].shape
    dct_china_year_publications[year], cols2 = df_china[df_china.Year==year].shape
    dct_usa_year_publications[year], cols3 = df_usa[df_usa.Year==year].shape
```

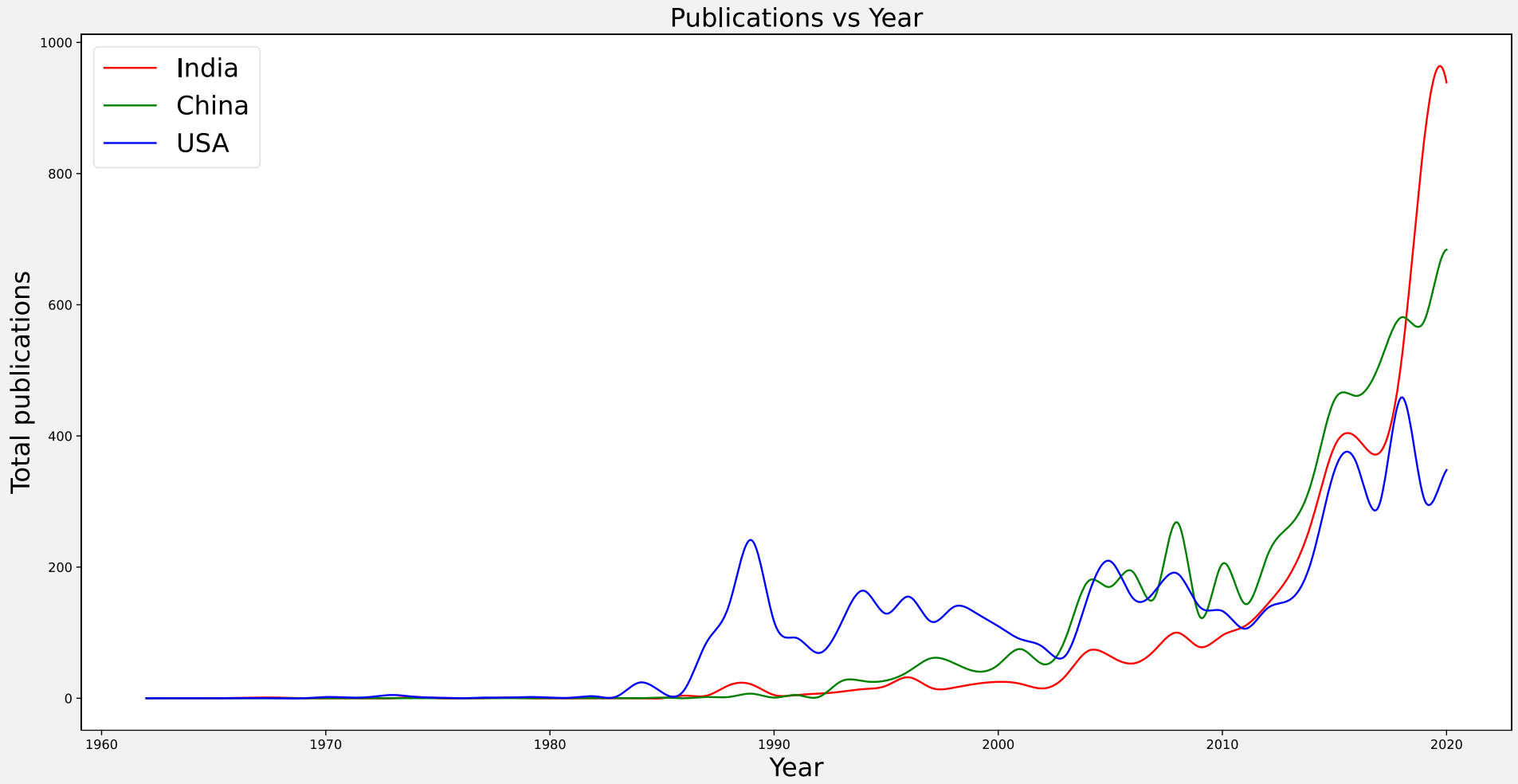
```
x_data = [list(dct_india_year_publications.keys()),
          list(dct_china_year_publications.keys()),
          list(dct_usa_year_publications.keys())]

y_data = [list(dct_india_year_publications.values()),
          list(dct_china_year_publications.values()),
          list(dct_usa_year_publications.values())]
```

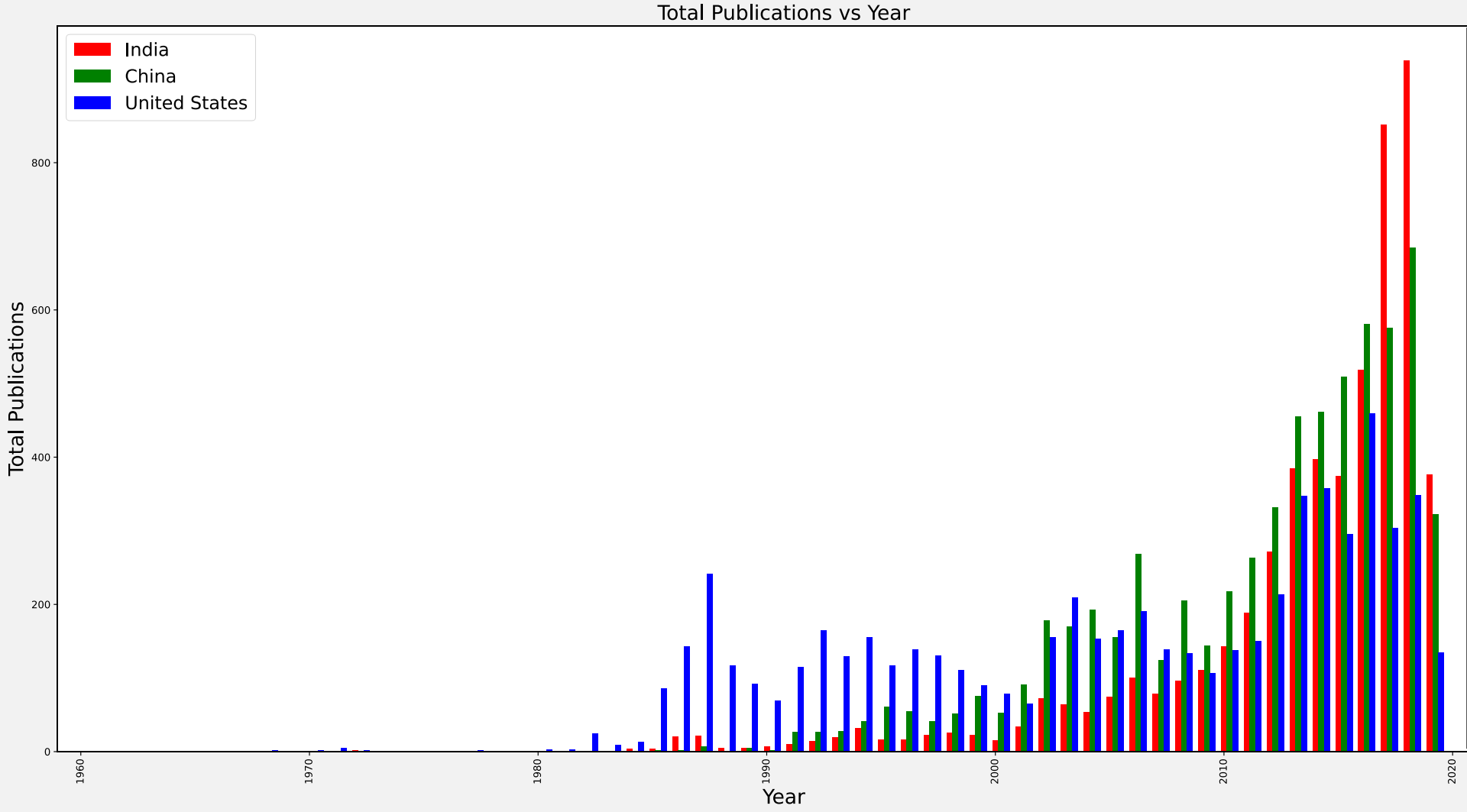
ROUGH LINE PLOT



SMOOTH LINE PLOT



BAR PLOT


Total Publications vs Year
India
China
United States
Total Publications
0
200
400
600
800
1960
1970
1980
1990
2000
2010
2020
Year

10. TOTAL NUMBER OF GRANTS GIVEN TO THE FIELD

```
grants, cols = df_without_duplicates[df_without_duplicates['Funding_Details']!='0'].shape
```

```
print(f'Grants given to field = {grants}')
```

```
Grants given to field = 4771
```

11. COUNTRY WISE TOTAL NUMBER OF PUBLICATION

```
dct_country_publications = {}
for country in country_lst:
    rows, columns = df[df.Country==country].shape
    dct_country_publications[country] = rows
```

```
dct_country_publications = sort_dict(dct_country_publications,'Value','reverse')
```

*Output in next slide

```
dct_country_publications
```

```
{'United Kingdom': 8994,
 'China': 6401,
 'United States': 6104,
 'India': 5383,
 'Germany': 5186,
 'Spain': 4759,
 'Canada': 4486,
 'Japan': 4324,
 'Italy': 4214,
 'France': 4133,
 'Australia': 3361,
 'South Korea': 3026,
 'Iran': 2720,
 'Taiwan': 2430,
 'Netherlands': 2173}
```

REFERENCES

- Scopus
- Pandas Documentation
- Numpy Documentation
- Matplotlib Documentation
- GeeksForGeeks
- StackOverFlow

THANK YOU