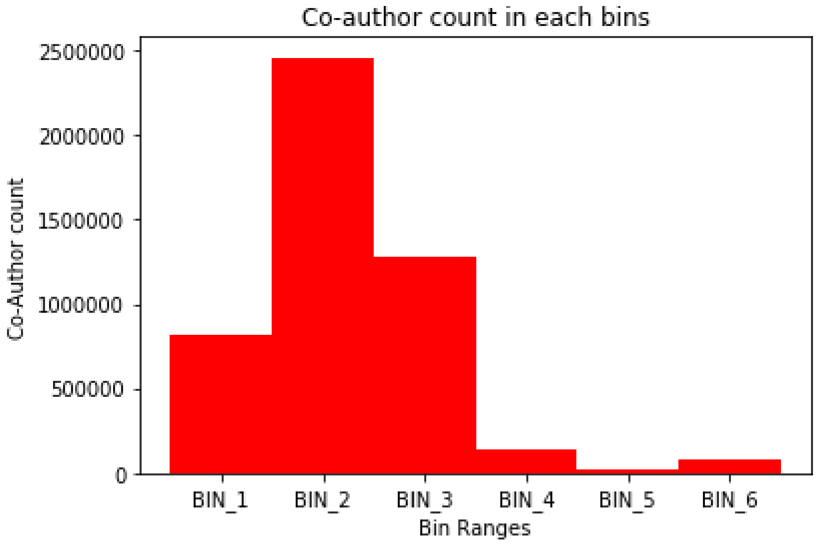
In [1]:

```
source_file_path = "C:/Users/eunic/Downloads/part-r-00000 (24)"   #Replace with the downloaded Mapreduce output file location
destination_folder_path = "C:/Users/eunic/Downloads"  #Replace with the location where you want the ouptut csv files to be stored

with open(source_file_path,'r', encoding="utf8") as f:
    FINDSTRING = "All_venues,"
    targets = [line.split(',', 1)[-1] for line in f if FINDSTRING in line]
    with open(destination_folder_path+'/allvenues.csv','wb') as file:
        #Write Header to the csv file
        file.write("BINS,Articles,Inproceedings,Proceedings,Book,Incollection,Phdthesis,Mastersthesis,Total\n".encode())
        for line in targets:
            file.write(line.encode())

with open(source_file_path,'r', encoding="utf8") as f:
    FINDSTRING = "AuthorScore,"
    targets = [line.split(',', 1)[-1] for line in f if FINDSTRING in line]
    with open(destination_folder_path+'/authorscore.csv','wb') as file:
        for line in targets:
            file.write(line.encode())

with open(source_file_path,'r', encoding="utf8") as f:
    FINDSTRING = "Journal_Inproceedings_Year,"
    targets = [line.split(',', 1)[-1] for line in f if FINDSTRING in line]
    with open(destination_folder_path+'/journal_inproc_year.csv','wb') as file:
        #Write Header to the csv file
        file.write("BIN,Journal,Inproceedings,YearRange(<1990),YearRange(1991-2000),YearRange(2001-2019)\n".encode())
        for line in targets:
            file.write(line.encode())

with open(source_file_path,'r', encoding="utf8") as f:
    FINDSTRING = "Co-AuthorCount,"
    targets = [line.split(',', 1)[-1] for line in f if FINDSTRING in line]
    with open(destination_folder_path+'/CoAuthor.csv','wb') as file:
        #Write Header to the csv file
        file.write("BIN,Co-Author Count\n".encode())
        for line in targets:
            file.write(line.encode())

with open(source_file_path,'r', encoding="utf8") as f:
    FINDSTRING = "MMA,"
    targets = [line.split(',', 1)[-1] for line in f if FINDSTRING in line]
    with open(destination_folder_path+'/MMA.csv','wb') as file:
        #Write Header to the csv file
        file.write("Author Name,Max,Median,Average\n".encode())
        for line in targets:
            file.write(line.encode())
```

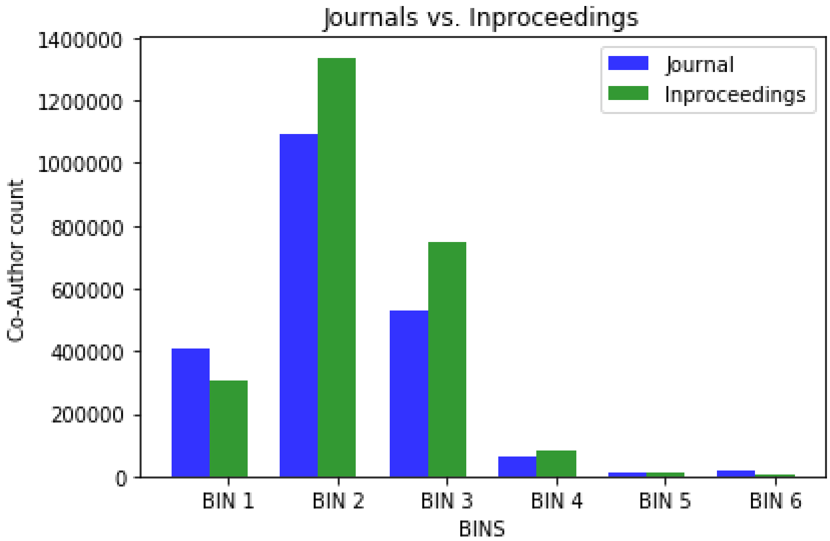
In [7]:

```
import matplotlib.pyplot as plt
import pandas as pd

#Histogram for Co-author count in each bins
d = pd.read_csv(destination_folder_path+'/CoAuthor.csv')
bins = d['BIN']
author_count = d['Co-Author Count']
plt.bar(bins,author_count,  width=1, color='r')
plt.xlabel('Bin Ranges')
plt.ylabel('Co-Author count')
plt.title('Co-author count in each bins')
plt.show()
```



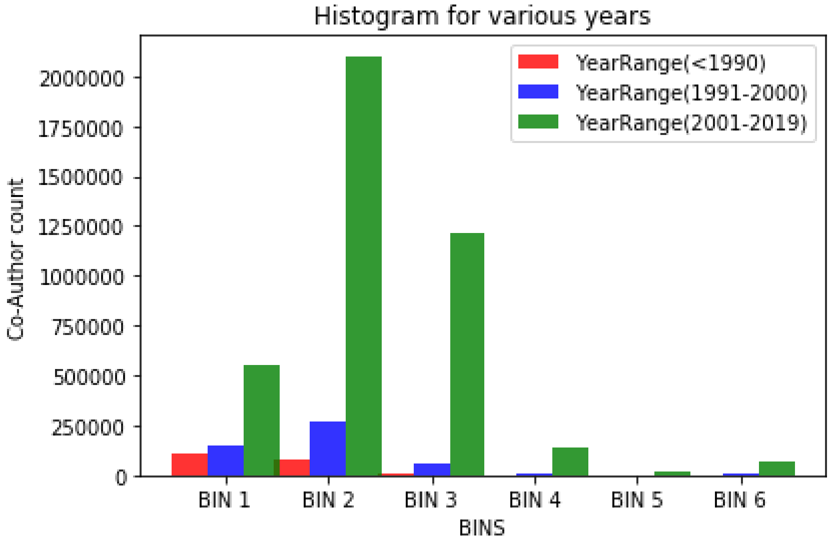
In [3]:

```python
import numpy as np
import matplotlib.pyplot as plt

#Histogram for journal and inprocedings in each bin
d = pd.read_csv(destination_folder_path+'/journal_inproc_year.csv')

# data to plot
n_groups = 6
journal_count = d['Journal']
inproc_count = d['Inproceedings']

# create plot
fig, ax = plt.subplots()
index = np.arange(n_groups)
bar_width = 0.35
opacity = 0.8

rects1 = plt.bar(index, journal_count, bar_width, alpha=opacity, color='b',label='Journal')
rects2 = plt.bar(index + bar_width, inproc_count, bar_width, alpha=opacity, color='g', label='Inproceedings')

plt.xlabel('BINS')
plt.ylabel('Co-Author count')
plt.title('Journals vs. Inproceedings')
plt.xticks(index + bar_width, ('BIN 1', 'BIN 2', 'BIN 3', 'BIN 4', 'BIN 5', 'BIN 6'))
plt.legend()

plt.tight_layout()
plt.show()
```



In [4]:

```
import numpy as np
import matplotlib.pyplot as plt

#Histogram for year ranges
d = pd.read_csv(destination_folder_path+'/journal_inproc_year.csv')

# data to plot
n_groups = 6
yearRange1 = d['YearRange(<1990)']
yearRange2 = d['YearRange(1991-2000)']
yearRange3 = d['YearRange(2001-2019)']

# create plot
fig, ax = plt.subplots()
index = np.arange(n_groups)
bar_width = 0.35
opacity = 0.8

rects3 = plt.bar(index , yearRange1, bar_width, alpha=opacity, color='r', label='YearRange(<1990)')
rects4 = plt.bar(index + bar_width, yearRange2, bar_width, alpha=opacity, color='b', label='YearRange(1991-2000)')
rects5 = plt.bar(index + 2*bar_width, yearRange3, bar_width, alpha=opacity, color='g', label='YearRange(2001-2019)')

plt.xlabel('BINS')
plt.ylabel('Co-Author count')
plt.title('Histogram for various years')
plt.xticks(index + bar_width, ('BIN 1', 'BIN 2', 'BIN 3', 'BIN 4', 'BIN 5', 'BIN 6'))
plt.legend()

plt.tight_layout()
plt.show()
```



In [5]:

```
import sys, csv ,operator
data = csv.reader(open(destination_folder_path+'/authorscore.csv', encoding="utf8"),delimiter=',')
sortedlist = sorted(data, key=operator.itemgetter(1))    # 0 specifies according to first column we want to sort
print("Top 100 authors with least co-authors")
for x in range(0,100):
    print(sortedlist[x])
```

```
Top 100 authors with least co-authors
['Ai Kaiho', '0.00379']
['Akiko Saka', '0.00379']
['Alan J. Knox', '0.00379']
['Albert S. B. Edge', '0.00379']
['Alessandro Bonetti', '0.00379']
['Alka Saxena', '0.00379']
['Anthony G. Beckhouse', '0.00379']
['Antje Blumenthal', '0.00379']
['Antti Sajantila', '0.00379']
['Atsutaka Kubosaki', '0.00379']
['Beatrice Bodega', '0.00379']
['Berit Lilje', '0.00379']
['Carlo V. Cannistraci', '0.00379']
['Chieko Kai', '0.00379']
['Christian Schmidl', '0.00379']
['Dipti Vijayan', '0.00379']
['Dmitry A. Ovchinnikov', '0.00379']
['Emiliano Dalla', '0.00379']
['Emily J. Wood', '0.00379']
['Eri Saijyo', '0.00379']
['Ernst Wolvetang', '0.00379']
['Fumi Hori', '0.00379']
['Fumio Nakahara', '0.00379']
['Gundula G. Schulze-Tanzil', '0.00379']
['Helena Persson', '0.00379']
['Hideki Enomoto', '0.00379']
['Hideki Tatsukawa', '0.00379']
['Hiroko Ohmiya', '0.00379']
['Hiromi Nishiyori', '0.00379']
['Hiroo Toyoda', '0.00379']
['Hozumi Motohashi', '0.00379']
['James Briggs', '0.00379']
['James G. D. Prendergast', '0.00379']
['Judith S. Kempfle', '0.00379']
['Jun-ichi Furusawa', '0.00379']
['Kaoru Kaida', '0.00379']
['Kazuhiro Kajiyama', '0.00379']
['Kelly J. Hitchens', '0.00379']
['Kenichi Nakazato', '0.00379']
['Linda M. van den Berg', '0.00379']
['Louise N. Winteringham', '0.00379']
['Lynsey Fairbairn', '0.00379']
['Magda Babina', '0.00379']
['Marc van de Wetering', '0.00379']
['Marco Roncador', '0.00379']
['Margaret Patrikakis', '0.00379']
['Mary C. Farach-Carson', '0.00379']
['Masahide Hamaguchi', '0.00379']
['Matthias Edinger', '0.00379']
['Matthias Harbers', '0.00379']
['Meenhard Herlyn', '0.00379']
['Mette Jørgensen', '0.00379']
['Michael Detmar', '0.00379']
['Michela Fagiolini', '0.00379']
['Michihira Tagami', '0.00379']
['Miki Kojima', '0.00379']
['Misako Yoneda', '0.00379']
['Mitsuhiro Endoh', '0.00379']
['Mitsuhiro Ohshima', '0.00379']
['Mitsuko Hara', '0.00379']
```

```
['Mitsuru Morimoto', '0.00379']
['Mitsuyoshi Murata', '0.00379']
['Mizuho Sakai', '0.00379']
['Morten B. Rye', '0.00379']
['Mutsumi Kanamori-Katayama', '0.00379']
['Naganari Ohkura', '0.00379']
['Naoko Suzuki', '0.00379']
['Niklas Mejhert', '0.00379']
['Noriko Ninomiya', '0.00379']
['Oliver M. Hofmann', '0.00379']
['Peter Arner', '0.00379']
['Peter G. Zhang', '0.00379']
['RIKEN CLST (DGT)', '0.00379']
['RIKEN PMII', '0.00379']
['Ri-ichiroh Manabe', '0.00379']
['Robert Passier', '0.00379']
['Rolf K. Swoboda', '0.00379']
['S. Peter Klinken', '0.00379']
['Sarah Krampitz', '0.00379']
['Sayaka Nagao-Sato', '0.00379']
['Shigehiro Yoshida', '0.00379']
['Shigeo Koyasu', '0.00379']
['Shimon Sakaguchi', '0.00379']
['Shohei Noma', '0.00379']
['Silvano Piazza', '0.00379']
['Silvia Zucchelli', '0.00379']
['Soichi Kojima', '0.00379']
['Sugata Roy', '0.00379']
['Susan E. Zabierowski', '0.00379']
['Suzana Savvi', '0.00379']
['Sven Guhl', '0.00379']
['Swati Pradhan-Bhatt', '0.00379']
['Tadasuke Nozaki', '0.00379']
['Taeko Dohi', '0.00379']
['Teunis B. Geijtenbeek', '0.00379']
['Thomas J. Ha', '0.00379']
['Tomokatsu Ikawa', '0.00379']
['Tony J. Kenna', '0.00379']
['Toshio Kitamura', '0.00379']
['Tsugumi Kawashima', '0.00379']
```

In [6]:

```
data = csv.reader(open(destination_folder_path+'/authorscore.csv', encoding="utf8"),delimiter=',')
sortedlistrev = sorted(data, key=operator.itemgetter(1), reverse=True)
print("Top 100 authors with the most co-authors")
for x in range(0,100):
    print(sortedlistrev[x])
```

```
Top 100 authors with the most co-authors
['Edward J. Delp', '99.950424']
['Kun Wang', '99.922455']
['Madhu Sudan', '99.83746']
['Sadaaki Miyamoto', '99.816696']
['Jiandong Li 0001', '99.80573']
['Toshihide Ibaraki', '99.801094']
['Yukio Ohsawa', '99.77469']
['Xiao Li', '99.76136']
['Jianfeng Ma', '99.72973']
['Jun Ma', '99.72175']
['Noam Nisan', '99.71767']
['Belur V. Dasarathy', '99.66667']
['Deepak Kapur', '99.62743']
['David N. Blank-Edelman', '99.625']
['Nan Li', '99.493416']
['Robin Milner', '99.48332']
['Mark Guzdial', '99.447205']
['Vijay K. Bhargava', '99.42911']
['Mohamed G. Gouda', '99.40975']
['Alessandro Astolfi', '99.36719']
['Markus H. Gross', '99.344826']
['Harald Haas', '99.3171']
['Ronald Fagin', '99.2445']
['Stefan Edelkamp', '99.2242']
['Seong-Whan Lee', '99.21649']
['John D. McGregor', '99.16364']
['Costas S. Iliopoulos', '99.113335']
['Raghu Ramakrishnan', '99.09887']
['Sudip Misra', '99.04636']
['Shigeo Hirose', '98.99845']
['Yang Zhao', '98.981064']
['Ioannis K. Argyros', '98.954216']
['Beng Chin Ooi', '98.94933']
['Derick Wood', '98.89578']
['David A. Bader', '98.85732']
['Kai-Kit Wong', '98.80602']
['Qiang Ji', '98.79476']
['Dorit S. Hochbaum', '98.77496']
['Patrick Solé', '98.74876']
['Kiyoharu Aizawa', '98.74702']
['Riccardo Poli', '98.73688']
['Jia Li', '98.73585']
["Pierre L'Ecuyer", '98.709526']
['Dirk T. M. Slock', '98.70749']
['Hamid Sarbazi-Azad', '98.64461']
['Jianping Wu', '98.63219']
['Sanjoy K. Baruah', '98.54054']
['Guang Gong', '98.53742']
['Chen Chen', '98.52862']
['Frank S. de Boer', '98.526054']
['Gerrit Bleumer', '98.5']
['John McLeod', '98.49705']
['Paul D. Seymour', '98.396935']
['Mohammed Atiquzzaman', '98.39276']
['Alexander G. Hauptmann', '98.3787']
['Philip A. Bernstein', '98.367775']
['Darwin G. Caldwell', '98.36631']
['Wenbo Wang 0007', '98.33633']
['Wiebe van der Hoek', '98.25816']
['Petros Maragos', '98.25222']
```

```
['Oscar C. Au', '98.22831']
['Hamid Jafarkhani', '98.221954']
['Ahmad-Reza Sadeghi', '98.13556']
['Wei Guo', '98.1297']
['Harold N. Gabow', '98.076385']
['Victor Vianu', '98.073616']
['Yue Zhao', '98.07087']
['Shaogang Gong', '98.0652']
['Gareth J. F. Jones', '98.042114']
['George J. Pappas', '98.04153']
['Johan Håstad', '98.03574']
['Rick S. Blum', '98.03151']
['Zhi-Quan Luo', '97.99513']
['Sherali Zeadally', '97.97731']
['Ling Guan', '97.88565']
['Geoffrey E. Hinton', '97.87771']
['Weisi Lin', '97.857056']
['Meng Wang', '97.85048']
['Ning Zhang', '97.843414']
['Lang Tong', '97.81757']
['Masaaki Harada', '97.71661']
['Susanne Albers', '97.70841']
['Rosalind W. Picard', '97.68526']
['Emanuele Viola', '97.67084']
['Randy H. Katz', '97.666145']
['Ricardo Baeza-Yates', '97.63946']
['Bing Li', '97.57449']
['Ronald V. Book', '97.53341']
['Michael J. Carey 0001', '97.524155']
['Pedro M. Domingos', '97.511406']
['Michel Grabisch', '97.481705']
['Frank K. Hwang', '97.46107']
['Bart De Bruyn', '97.41666']
['Philippe Balbiani', '97.38209']
['Jürgen Ziegler 0001', '97.32273']
['Godfried T. Toussaint', '97.13786']
['Kokichi Sugihara', '97.133736']
['Peng Xu', '97.11642']
['Xinghuo Yu', '97.048096']
['Ling Li', '97.02137']
```