IT 503 Strategic Planning and Management

Etungu Isaac

M21/MIT/7424

It is estimated that more than a billion bicycles are present in the world, with nearly half of them in China. Data set can be grabbed from URL: https://www.worldometers.info/bicycles/ (https://www.worldometers.info/bicycles/)

Dataset representation in:

1. A Column chart
2. A Bar chart

Importing Libraries

In [1]:
```
import requests
import lxml.html as lh
import pandas as pd
data = pd.DataFrame()
```

In [2]:
```
url = 'https://github.com/IsaacEtungu/Data-files/blob/main/convertcsv.csv' #assign the wiki page
#url = 'https://github.com/IsaacEtungu/Data-files/blob/main/convertcsv.csv/'

page = requests.get(url) # creating a handle for contents of the wiki page

doc = lh.fromstring(page.content) # storing content of the wiki page under doc

tr_elements = doc.xpath('//tr') # parsing data stored between tr in the html

[len(T) for T in tr_elements[:12]] # check the length of the first 12 rows
```

Out[2]:
```
[4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
```

In [3]:
```
tr_elements = doc.xpath('//tr') # parse first row as header

col = [] # create empty list
i = 0

for t in tr_elements[0]: # for each row, store each first element (header) and an empty list
    i+=1
    name=t.text_content()
    print("%d:%s" % (i,name))
    col.append((name,[]))
```

```
1:
2:Country
3:Quantity
4:Year
```

In [4]:
```
for j in range(1,len(tr_elements)): # Because header is the first row, data would be store in the subsequent row
    T = tr_elements[j] #T is j'th row

    if len(T)!=4: #if row is not size 3, //tr data is not from the table.
        break

    i = 0 #i is the index of the first column

    for t in T.iterchildren(): #iterate through each element of the row
        data=t.text_content()

        col[i][1].append(data) #append the data to the empty list of the i'th column

        i+=1 #increment i for the next column
```

In [5]:
```
#Checking number of rows in the dataset
[len(C) for (title,C) in col]
```

Out[5]:
```
[24, 24, 24, 24]
```

In [6]:
```
#Reading dataset with Pandas
Dict = {title:column for (title,column) in col}
data = pd.DataFrame(Dict)
```

In [7]:
```
#Viewing dataset
data.head()
```

Out[7]:

	Country	Quantity	Year
0	China	450,000,000	1992
1	USA	100,000,000	1995
2	Japan	72,540,000	1996
3	Germany	62,000,000	1996
4	India	30,800,000	1990

In [8]:
```
data.shape
```

Out[8]:
```
(24, 4)
```

In [9]:
```
def DataFrameCleaner(data):
    for columnname in data: #looping through titles of the table
        temp = []
        for column in data [columnname]: #geting column elements for the each title
            column = str(column)
            column = column.replace(',','') # Removing unwanted data clutter
            column = column.replace('+','') #Removing unwanted '+'sign
            try : #using try except block to convert datatype string to integer while avoiding error
                column = int(column)
            except:
                pass
            temp.append(column)
        data[columnname] = temp
    #df = data.drop(data.tail(1).index) # Deleting the last row
    data = data.replace(r'^\s*$',0,regex=True) # converting empty string to 0
    return data
```

In [10]:
```
data = DataFrameCleaner(data)
data.head()
```

Out[10]:

		Country	Quantity	Year
0	0	China	450000000	1992
1	0	USA	100000000	1995
2	0	Japan	72540000	1996
3	0	Germany	62000000	1996
4	0	India	30800000	1990

In [11]:
```
data.sort_values(['Year'], ascending=False, axis=0, inplace=True)
data.head().transpose()
```

Out[11]:

	10	2	3	19	9
	0	0	0	0	0
Country	Netherlands	Japan	Germany	Switzerland	Brazil
Quantity	16500000	72540000	62000000	3800000	40000000
Year	2000	1996	1996	1996	1996

In [12]:
```
x = data['Year']
y = data['Quantity']
dataplot=(x,y)
dataplot
```

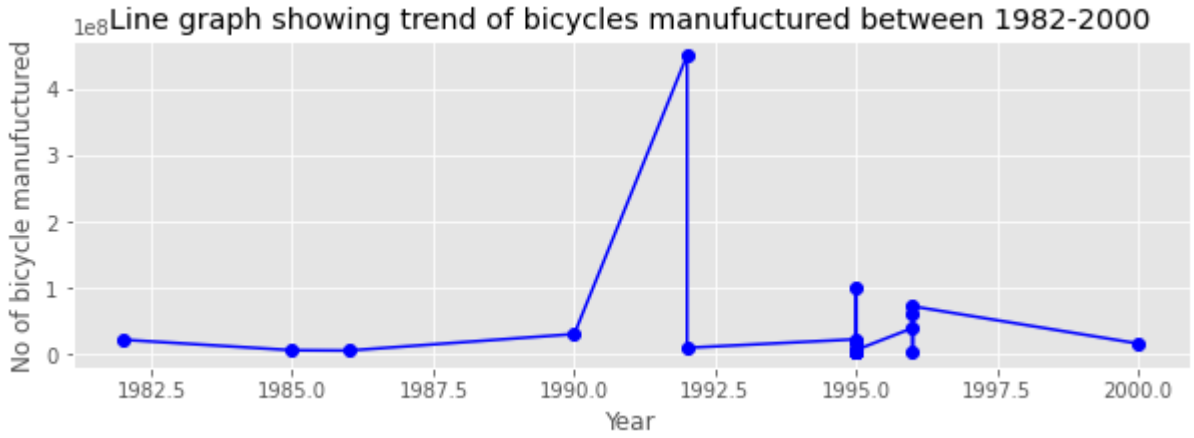
Out[12]:
```
(10    2000
 2     1996
 3     1996
 19    1996
 9     1996
 12    1995
 13    1995
 22    1995
 21    1995
 20    1995
 18    1995
 17    1995
 16    1995
 23    1995
 1     1995
 8     1995
 7     1995
 6     1995
 11    1992
 0     1992
 4     1990
 15    1986
 14    1985
 5     1982
 Name: Year, dtype: int64,
 10     16500000
 2      72540000
 3      62000000
 19      3800000
 9      40000000
 12      6950000
 13      6000000
 22      3250000
 21      3300000
 20      3500000
 18      4500000
 17      5000000
```

```
16      5200000
23      3000000
1     100000000
8      20000000
7      20000000
6      23000000
11     10150000
0     450000000
4      30800000
15      6000000
14      6500000
5      22300000
Name: Quantity, dtype: int64)
```

In [13]:
```
# use the inline backend to generate the plots within the browser
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
mpl.style.use('ggplot') # optional: for ggplot-like style
# check for latest version of Matplotlib
print ('Matplotlib version: ', mpl.__version__) # >= 2.0.0
```

```
Matplotlib version:  3.3.4
```

In [20]:
```
x = data['Year']
y = data['Quantity']
plt.plot(x,y, color='blue', marker='o')
plt.rcParams["figure.figsize"] = (15,3)
plt.title('Line graph showing trend of bicycles manufuctured between 1982-2000')
plt.xlabel('Year')
plt.ylabel('No of bicycle manufuctured')
plt.show()
```



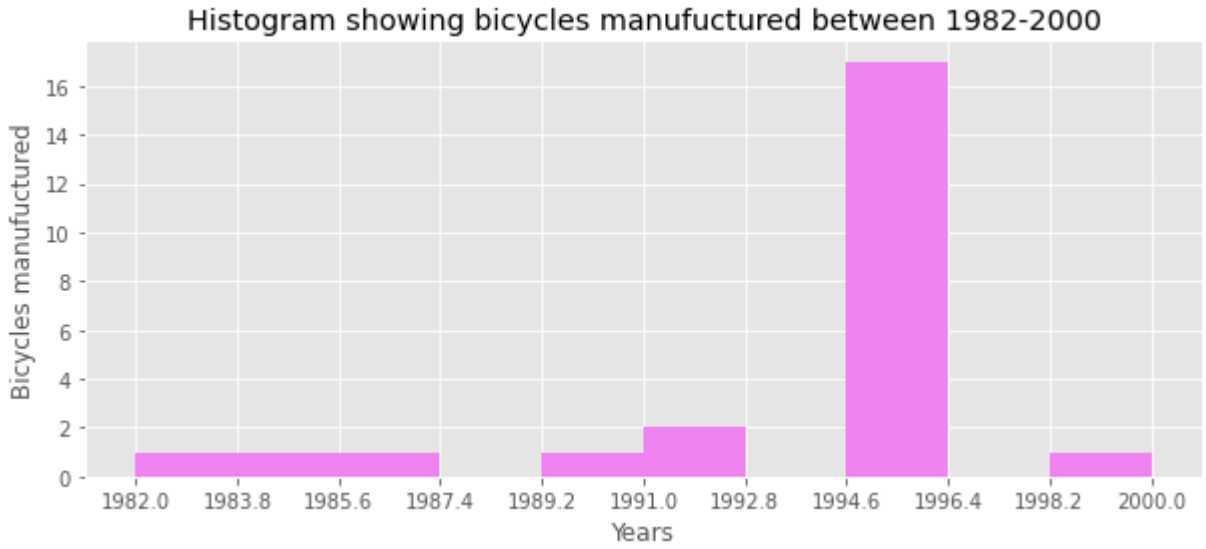
bicycle mamufucturing increased steadily between 1982 to 1990 and had a sharp increase in 1991 then drastically reduced in 1991. It stable production expereinced betwwen 1992 and 1996. It has ever since reduced upto 2000

In [15]:
```
#np.histogram returns 2 values
count, bin_edges = np.histogram(data['Year'])
print(count) # frequency count
print(bin_edges) # bin ranges, default = 10 bins
```

```
[ 1  1  1  0  1  2  0 17  0  1]
[1982.  1983.8 1985.6 1987.4 1989.2 1991.  1992.8 1994.6 1996.4 1998.2
 2000. ]
```

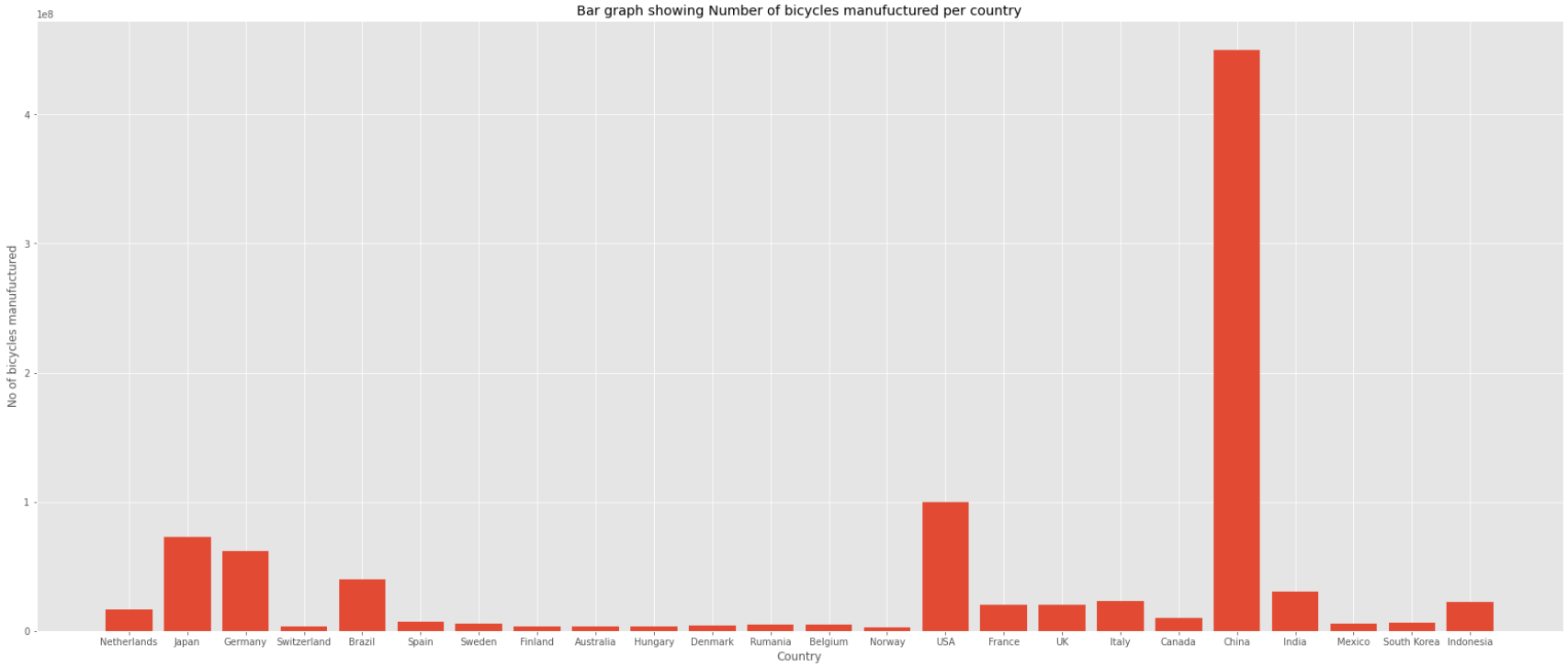
In [16]:
```
count, bin_edges = np.histogram(data['Year'])
data['Year'].plot(kind='hist', figsize=(10, 4), color='violet', xticks=bin_edges)

plt.title('Histogram showing bicycles manufuctured between 1982-2000') # add a title to the histogram
plt.ylabel('Bicycles manufuctured') # add y-label
plt.xlabel('Years') # add x-label
plt.show()
```



bicycle manufucturing was high between 1994 and 1996 with the lowest production in in 1982 up t0 19991 and 1998 to 2000

In [27]:
```
fig = plt.figure()
ax = fig.add_axes([1,0,1.5,3])
a = data['Country']
b = data['Quantity']
ax.bar(a,b)
plt.title('Bar graph showing Number of bicycles manufuctured per country')
plt.xlabel('Country')
plt.ylabel('No of bicycles manufuctured')
plt.show()
```

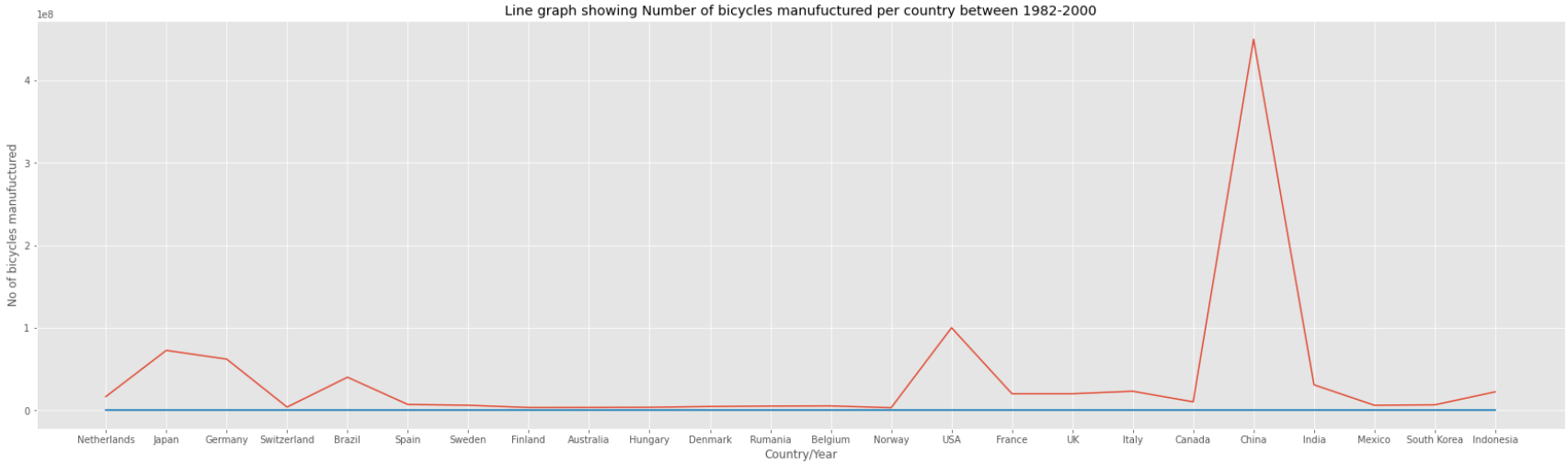


China had the hight production of motorcycle ever

In [18]:
```
data_set = set(data['Country'])
data_set
```

Out[18]:
```
{'Australia',
 'Belgium',
 'Brazil',
 'Canada',
 'China',
 'Denmark',
 'Finland',
 'France',
 'Germany',
 'Hungary',
 'India',
 'Indonesia',
 'Italy',
 'Japan',
 'Mexico',
 'Netherlands',
 'Norway',
 'Rumania',
 'South Korea',
 'Spain',
 'Sweden',
 'Switzerland',
 'UK',
 'USA'}
```

In [28]:
```
fig = plt.figure()
ax = fig.add_axes([0,0,1.5,2])
plt.plot(data['Country'], data['Quantity'], data['Year'])
plt.title('Line graph showing Number of bicycles manufuctured per country between 1982-2000')
plt.xlabel('Country/Year')
plt.ylabel('No of bicycles manufuctured')
plt.show()
```



Ever since 1982, bike production has generall reduced with the highest production ever recorded in China

END