

# Press\_freedom

May 25, 2024

```
[1]: #Importing the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
[2]: #Importing the data
press = pd.read_excel('/content/drive/MyDrive/Computers/Python/Datasets for_
↳data analysis/Media Freedom/Press Freedom Index.xlsx')
```

## 1 Data Cleaning

```
[3]: #Getting the head of the data
press.head()
```

```
[3]:
```

	Year	ISO	Rank	Score	Country	Zone
0	2023	NOR	1	95.18	Norway	UE Balkans
1	2023	IRL	2	89.91	Ireland	UE Balkans
2	2023	DNK	3	89.48	Denmark	UE Balkans
3	2023	SWE	4	88.15	Sweden	UE Balkans
4	2023	FIN	5	87.94	Finland	UE Balkans

```
[4]: #Getting the shape of the data
press.shape
```

```
[4]: (1979, 6)
```

```
[5]: #Getting the info about the dataset
press.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1979 entries, 0 to 1978
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Year        1979 non-null   int64
1   ISO         1979 non-null   object
2   Rank        1979 non-null   int64
```

```

3   Score    1979 non-null   float64
4   Country  1979 non-null   object
5   Zone     1977 non-null   object
dtypes: float64(1), int64(2), object(3)
memory usage: 92.9+ KB

```

```

[6]: #Checking if there any nulls in the dataset
press.isnull().sum()
#There are 2 nulls values in the zone column. But Zone is an unwanted column.
    ↳ that we will drop.

```

```

[6]: Year      0
     ISO       0
     Rank      0
     Score     0
     Country   0
     Zone      2
     dtype: int64

```

```

[7]: #Checking if there are any duplicate values
press.duplicated().sum()
#There are no duplicate values as well

```

```

[7]: 0

```

```

[8]: #Deleting unwanted columns
press = press.drop(['Zone'],axis = 1)

#checking if it got deleted
press.head()

```

```

[8]:   Year  ISO  Rank  Score  Country
0  2023  NOR    1   95.18   Norway
1  2023  IRL    2   89.91  Ireland
2  2023  DNK    3   89.48  Denmark
3  2023  SWE    4   88.15   Sweden
4  2023  FIN    5   87.94  Finland

```

## 2 Data Analysis

```

[9]: #Top 10 countries as of 2023
top_10_countries_2023 = press[press['Year'] ==2023].sort_values(by = "Rank").
    ↳ head(10)
print(top_10_countries_2023[['Country', 'Rank', 'Score']])

```

```

      Country  Rank  Score
0      Norway    1   95.18

```

1	Ireland	2	89.91
2	Denmark	3	89.48
3	Sweden	4	88.15
4	Finland	5	87.94
5	Netherlands	6	87.00
6	Lithuania	7	86.79
7	Estonia	8	85.31
8	Portugal	9	84.60
9	East Timor	10	84.49

```
[10]: #Countries with least Press Freedom
bottom_10_countries_2023 = press[press['Year'] == 2023].sort_values(by =
    ↪'Rank',ascending = False).head(10)
print(bottom_10_countries_2023[['Country', 'Rank', 'Score']])
```

	Country	Rank	Score
179	North Korea	180	21.72
178	China	179	22.97
177	Vietnam	178	24.58
176	Iran	177	24.81
175	Turkmenistan	176	25.82
174	Syria	175	27.22
173	Eritrea	174	27.86
172	Myanmar	173	28.26
171	Cuba	172	29.00
170	Bahrain	171	30.59

```
[11]: #Top 10 countires for 2022
top_10_countries_2022 = press[press['Year'] == 2022].sort_values(by = "Rank").
    ↪head(10)
print(top_10_countries_2022[['Country', 'Rank', 'Score']])
```

	Country	Rank	Score
180	Norway	1	92.65
181	Denmark	2	90.27
182	Sweden	3	88.84
183	Estonia	4	88.83
184	Finland	5	88.42
185	Ireland	6	88.30
186	Portugal	7	87.07
187	Costa Rica	8	85.92
188	Lithuania	9	84.14
189	Liechtenstein	10	84.03

```
[12]: #Bottom 10 for 2022
bottom_10_countries_2022 = press[press['Year'] == 2022].sort_values(by =
    ↪'Rank',ascending = False).head(10)
```

```
print(bottom_10_countries_2022[['Country', 'Rank', 'Score']])
```

	Country	Rank	Score
359	Democratic People's Republic of Korea	180	13.92
358	Eritrea	179	19.62
357	Islamic Republic of Iran	178	23.22
356	Turkmenistan	177	25.01
355	Myanmar	176	25.03
354	China	175	25.17
353	Vietnam	174	26.11
352	Cuba	173	27.32
351	Iraq	172	28.59
350	Syrian Arab Republic	171	28.94

```
[13]: #Top 10 for 2021
top_10_countries_2021 = press[press['Year'] == 2021].sort_values(by = "Rank").\
↳head(10)
print(top_10_countries_2021[['Country', 'Rank', 'Score']])
```

	Country	Rank	Score
360	Norway	1	93.28
361	Finland	2	93.01
362	Sweden	3	92.76
363	Denmark	4	91.43
364	Costa Rica	5	91.24
365	Netherlands	6	90.33
366	Jamaica	7	90.04
367	New Zealand	8	89.96
368	Portugal	9	89.89
369	Switzerland	10	89.45

```
[14]: #Bottom 10 for 2021
bottom_10_countries_2021 = press[press['Year'] == 2021].sort_values(by =\
↳'Rank',ascending = False).head(10)
print(bottom_10_countries_2021[['Country', 'Rank', 'Score']])
```

	Country	Rank	Score
539	Eritrea	180	18.55
538	Democratic People's Republic of Korea	179	18.72
537	Turkmenistan	178	19.97
536	China	177	21.28
535	Djibouti	176	21.38
534	Vietnam	175	21.54
533	Islamic Republic of Iran	174	27.30
532	Syrian Arab Republic	173	29.37
531	Lao People's Democratic Republic	172	29.44
530	Cuba	171	36.06

```
[15]: #Plotting the ranks for India, China, Pakistan, Bangladesh and SriLanka
countries = ['India', 'Pakistan', 'China', 'Sri Lanka', 'Bangladesh']
years = [2021,2022,2023]
scores = {}

for year in years:
    scores[year] = {}
    for country in countries:
        country_data = press[(press['Country'] == country) & (press['Year'] ==
↪year)]
        scores[year][country] = country_data['Score'].values[0]

#Created an empty dictionary and filled the key and values using for loop.

# Create a color map
colors = plt.cm.get_cmap('viridis', len(countries))

fig, ax = plt.subplots(figsize=(15,8))

# Total width for all bars at one x location
total_width = 0.8
# Width of a single bar
individual_width = total_width / len(countries)
# This list will hold the offset for each country
offsets = [-total_width / 2 + individual_width * i for i in
↪range(len(countries))]

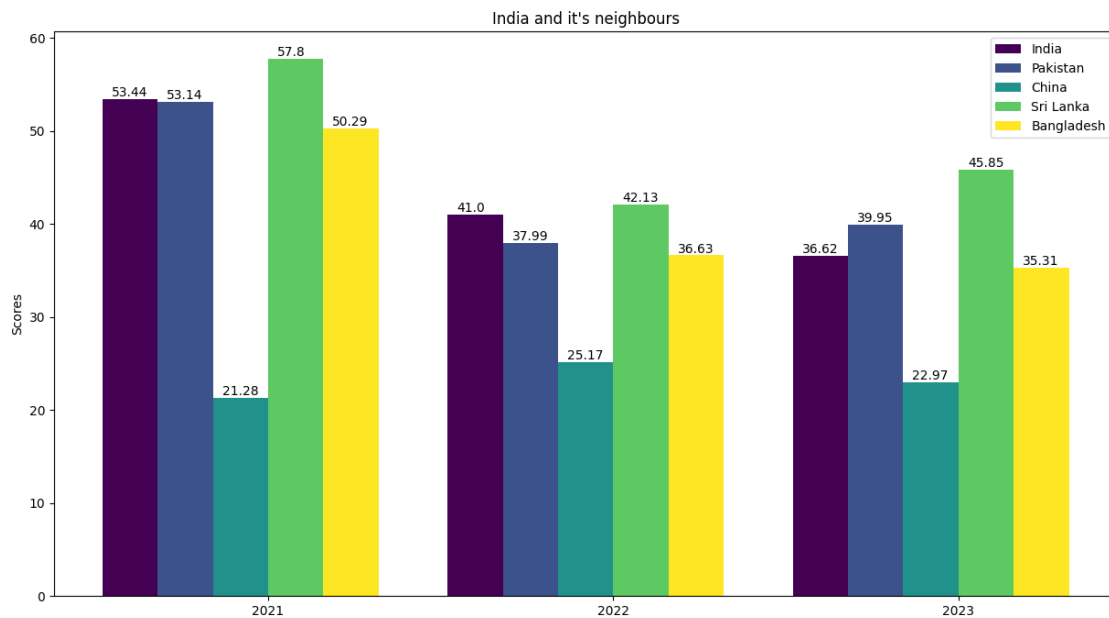
for i, country in enumerate(countries):
    country_scores = [scores[year][country] for year in years]
    # Apply offset to each country's bars
    bars = ax.bar(np.arange(len(years)) + offsets[i], country_scores,
        width=individual_width, color=colors(i/float(len(countries))),
↪label=country)

    # Add scores on top of the bars
    for bar in bars:
        yval = bar.get_height()
        ax.text(bar.get_x() + bar.get_width()/6.0, yval, round(yval, 2),
↪va='bottom')

# Adding labels
ax.set_ylabel('Scores')
ax.set_title("India and it's neighbours")
ax.set_xticks(np.arange(len(years)))
ax.set_xticklabels(years)
ax.legend()
```

```
# Display the figure
plt.show()
```

```
<ipython-input-15-a69e5958da80>:15: MatplotlibDeprecationWarning: The get_cmap
function was deprecated in Matplotlib 3.7 and will be removed two minor releases
later. Use ``matplotlib.colormaps[name]`` or
``matplotlib.colormaps.get_cmap(obj)`` instead.
    colors = plt.cm.get_cmap('viridis', len(countries))
```



```
[16]: #Tracking the rank of India over years
india_rank = press[press['Country'] == 'India'].sort_values(by = 'Year')['Rank']
india_years = press['Year'].unique()
print(india_years)
print(india_rank)
```

```
[2023 2022 2021 2020 2019 2018 2017 2016 2015 2014 2013]
```

```
1939    140
1759    140
1575    136
1392    133
1215    136
1037    138
859     140
681     142
501     142
329     150
160     161
```

Name: Rank, dtype: int64

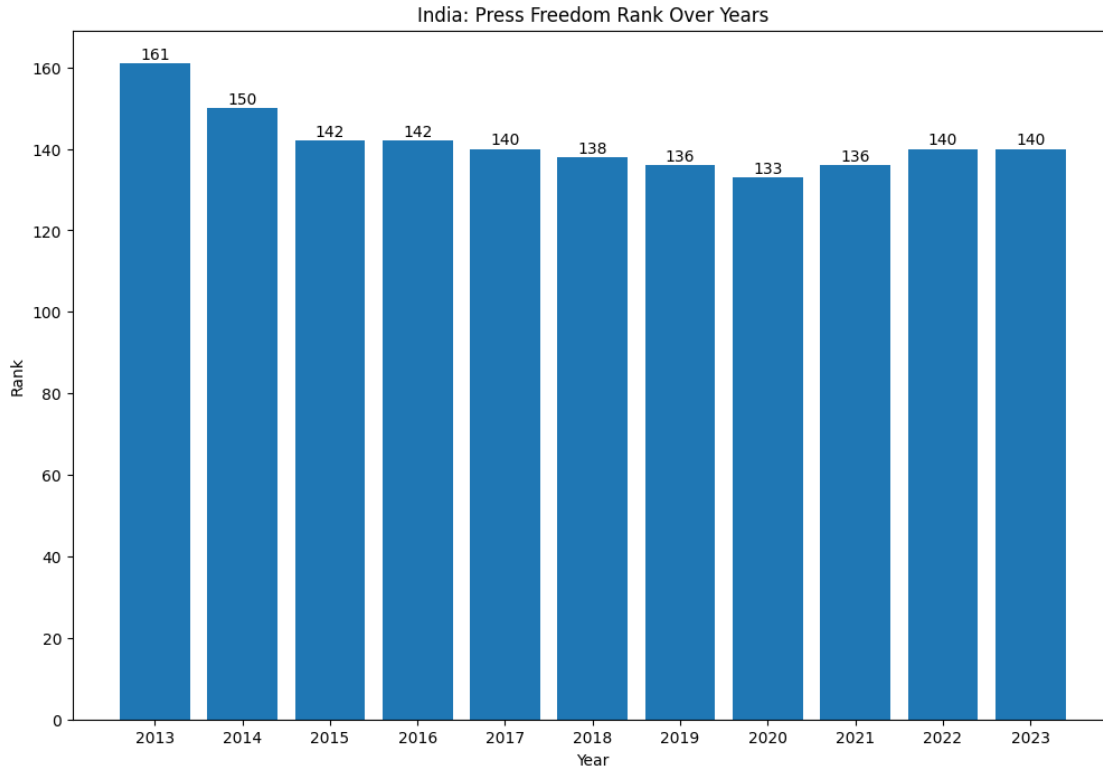
```
[18]: india_rank = press[press['Country'] == 'India'].sort_values(by = 'Year')['Rank']
india_years = press['Year'].unique()

# Create the bar chart
plt.figure(figsize=(12, 8))
bars = plt.bar(india_years, india_rank)

# Add data labels on top of each bar
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, yval + 0.2, yval, ha='center',
    ↪va='bottom')

# Add title and labels
plt.title('India: Press Freedom Rank Over Years')
plt.xlabel('Year')
plt.ylabel('Rank')
plt.xticks(india_years)

# Show the chart
plt.show()
```



```
[38]: #Comparing the values of India and China
comp_country = ['India','China']
comp_years = [2019,2020,2021,2022,2023]
score = {}

for year in comp_years:
    score[year]= {}
    for country in comp_country:
        data_country = press[(press['Country'] == country) & (press['Year'] ==
↪year)]
        score[year][country] = data_country['Score'].values[0]
#Plotting a graph
plt.figure(figsize=(12,8))
years = list(score.keys())
india_scores = [score[year]['India'] for year in years]
china_scores = [score[year]['China'] for year in years]

plt.bar(years,india_scores,label = 'India',width = 0.4, align = 'center')
plt.bar([year + 0.4 for year in years], china_scores, label='China', width=0.4)

#labels
plt.xticks(years)
plt.title("India vs China (higher score = better rank)")
plt.xlabel('Year')
plt.ylabel('Score')
plt.legend()
plt.show()
```



