US CRIMES (1979-1985) 🔔

Dataset Analysed by

- AARYA GUDDADKERI,
- ANUVIND DUBEY,
- ARNAV M,
- ANOUSHKA PANDEY.

In [1]:
```
#Importing and reading the dataset

import pandas as pd
p1=pd.read_csv("us crimes.csv")
p1.head(5)
```

Out[1]:

	year	State	population	violent_crime	homicide	Rape	robbery	aggravated_assault	pro
0	1979	NaN	220099000	1208030	21460	76390	480700	629480	
1	1979	Alaska	406000	1994	54	292	445	1203	
2	1979	Alabama	3769000	15578	496	1037	4127	9918	
3	1979	Arkansas	2180000	7984	198	595	1626	5565	
4	1979	Arizona	2450000	14528	219	1120	4305	8884	

DATA PREPROCESSING

In [2]:
```
# Removing NaN values from the entire DataFrame

import pandas as pd
df = pd.read_csv('us crimes.csv')
df_cleaned = df.dropna(axis=0, how='any')
df_cleaned.to_csv('cleaned_dataset_project.csv', index=False)
```

In [3]:
```
# Reading the cleaned dataset

p1=pd.read_csv("cleaned_dataset_project.csv")
p1
```

Out[3]:

	year	State	population	violent_crime	homicide	Rape	robbery	aggravated_assault
0	1979	Alaska	406000	1994	54	292	445	1203
1	1979	Alabama	3769000	15578	496	1037	4127	9918
2	1979	Arkansas	2180000	7984	198	595	1626	5565
3	1979	Arizona	2450000	14528	219	1120	4305	8884
4	1979	California	22696000	184087	2952	12239	75767	93129
...	...	...	...	...	...	...	...	...
337	1985	Pennsylvania	11853000	39240	550	2886	17429	18375
338	1985	Rhode Island	968000	3355	35	253	1122	1945
339	1985	South Carolina	3347000	21121	304	1385	3143	16289
340	1985	South Dakota	708000	967	13	168	121	665
341	1985	Tennessee	4762000	22592	429	2027	8614	11522

342 rows × 12 columns

In [4]:
```
# Reducing the number of rows to 200

import pandas as pd
data = pd.read_csv('cleaned_dataset_project.csv')
reduced_data = data.sample(n=200, random_state=42)
reduced_data.to_csv('reduced_output_file.csv', index=False)
```

In [5]:
```
#Reading the reduced dataset

p1=pd.read_csv("reduced_output_file.csv")
p1.head(5)
```

Out[5]:

	year	State	population	violent_crime	homicide	Rape	robbery	aggravated_assault	p
0	1983	Pennsylvania	11895000	40782	583	2449	20501	17249	
1	1981	Kansas	2381000	8796	151	733	2611	5301	
2	1981	Idaho	959000	2717	43	198	362	2114	
3	1979	Texas	13385000	67988	2235	6043	25667	34043	
4	1981	North Carolina	5951000	25986	541	1351	4809	19285	

Conclusion :

This code provides a random sample of 200 rows from the original cleaned dataset for further analysis. This sample ensures that the analysis is not biased towards specific subsets of the data, such as certain regions or demographic groups.

In [6]:
```
p1.isnull().sum()
```

Out[6]:
```
year                 0
State                0
population           0
violent_crime        0
homicide             0
Rape                 0
robbery              0
aggravated_assault   0
property_crime       0
burglary             0
larceny              0
Vehicle_theft        0
dtype: int64
```

Conclusion :

By using the above code, we can see that there are no Nan values in the dataset.

MEAN OF THE POPULATION

In [7]:
```
import pandas as pd
df = pd.read_csv('reduced_output_file.csv')

mean_population = df['population'].mean()
print(f"The mean of the population is: {mean_population}")
```

```
The mean of the population is: 4635650.62
```

Conclusion :

The mean of the population is: 4635650.62

MEDIAN OF THE ROBBERIES

In [8]:
```
import pandas as pd
df = pd.read_csv('reduced_output_file.csv')

median_robbery = df['robbery'].median()
print(f"The median of robbery is: {median_robbery}")
```

```
The median of robbery is: 4014.0
```

Conclusion :

The median of robbery is: 4014.0

MODE OF THE STATES

In [9]:
```
import pandas as pd
df = pd.read_csv('reduced_output_file.csv')
mode_state = df['State'].mode()
print(f"The Mode of state is: {mode_state}")
print(f"Mode finds the most occured state with highest crime rate")
```

```
The Mode of state is: 0          Montana
1    New Hampshire
Name: State, dtype: object
Mode finds the most occured state with highest crime rate
```

Conclusion :

Using mode we can find out that the highest crime rate ouccurs in the state of Montana and New Hampshire

---

MAXIMUM NUMBER OF VIOLENT CRIME

In [10]:
```
import pandas as pd
p1=pd.read_csv("reduced_output_file.csv")
p1['violent_crime'].max()
```

Out[10]:
```
208485
```

Conclusion :

Using the max() tag we can find out that 208485 is the highest violent crime which could be murder.

MINIMUM NUMBER OF VIOLENT CRIME

In [11]:
```
import pandas as pd
p1=pd.read_csv("reduced_output_file.csv")
p1['violent_crime'].min()
```

Out[11]:
```
322
```

Conclusion :

Minimum number of violent crime is 322

RANGE OF VIOLENT CRIME

In [12]:
```
import pandas as pd
df = pd.read_csv('reduced_output_file.csv')
range_violent_crime = df['violent_crime'].max() - df['violent_crime'].min(
print(f"The Range of violent_crime is: {range_violent_crime}")
```

```
The Range of violent_crime is: 208163
```

Conclusion :

The Range of violent_crime is: 208163

STANDARD DEVIATION IN PROPERTY CRIME

In [13]:
```
import pandas as pd
df = pd.read_csv('reduced_output_file.csv')
std_property_crime = df['property_crime'].std()
print(f"The Standard deviation in property_crime is: {std_property_crime}"
```

```
The Standard deviation in property_crime is: 284171.3094806382
```

Concusion :

The Standard deviation in property_crime is: 284171.3094806382

SKEWNESS OF HOMICIDE

In [14]:
```
import pandas as pd
df = pd.read_csv('reduced_output_file.csv')
skew_homicide = df['homicide'].skew()
print(f"The skewness of homicide is: {skew_homicide}")
```

```
The skewness of homicide is: 2.5633274990046235
```

Conclusion :

The positive value suggests a right-skewed distribution which means the data has a longer tail on the right side.

KURTOSIS OF BURGARLY

In [15]:
```
import pandas as pd
df = pd.read_csv('reduced_output_file.csv')
kurt_burglary = df['burglary'].kurt()
print(f"The kurtosis of burglary is: {kurt_burglary}")
```

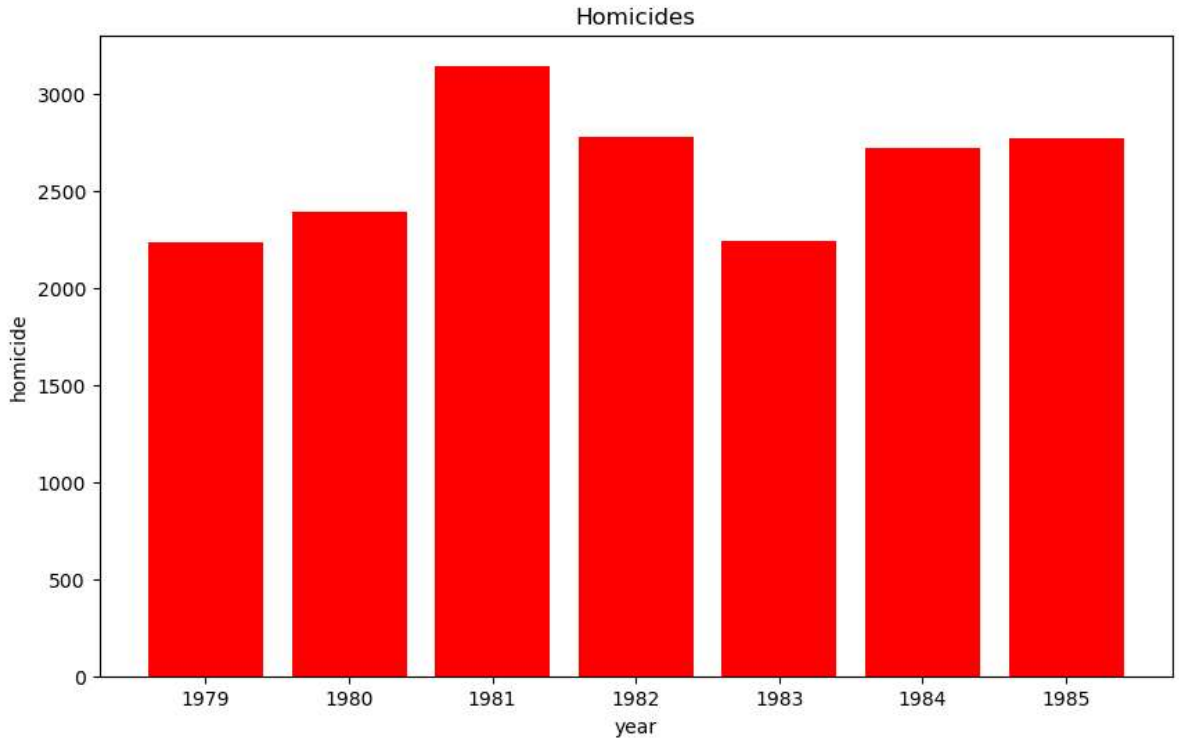
```
The kurtosis of burglary is: 9.143120187810496
```

Conclusion :

The data is leptokurtic which means kurtosis is positive and distribution has heavier tails than a normal distribution.

BAR GRAPH OF HOMICIDES THROUGHOUT THE YEAR

In [16]:
```
1 import matplotlib.pyplot as plt
2 plt.figure(figsize=(10, 6))
3 plt.bar(df['year'], df['homicide'], color='red')
4 plt.xlabel('year')
5 plt.ylabel('homicide')
6 plt.title('Homicides')
7 plt.show()
```



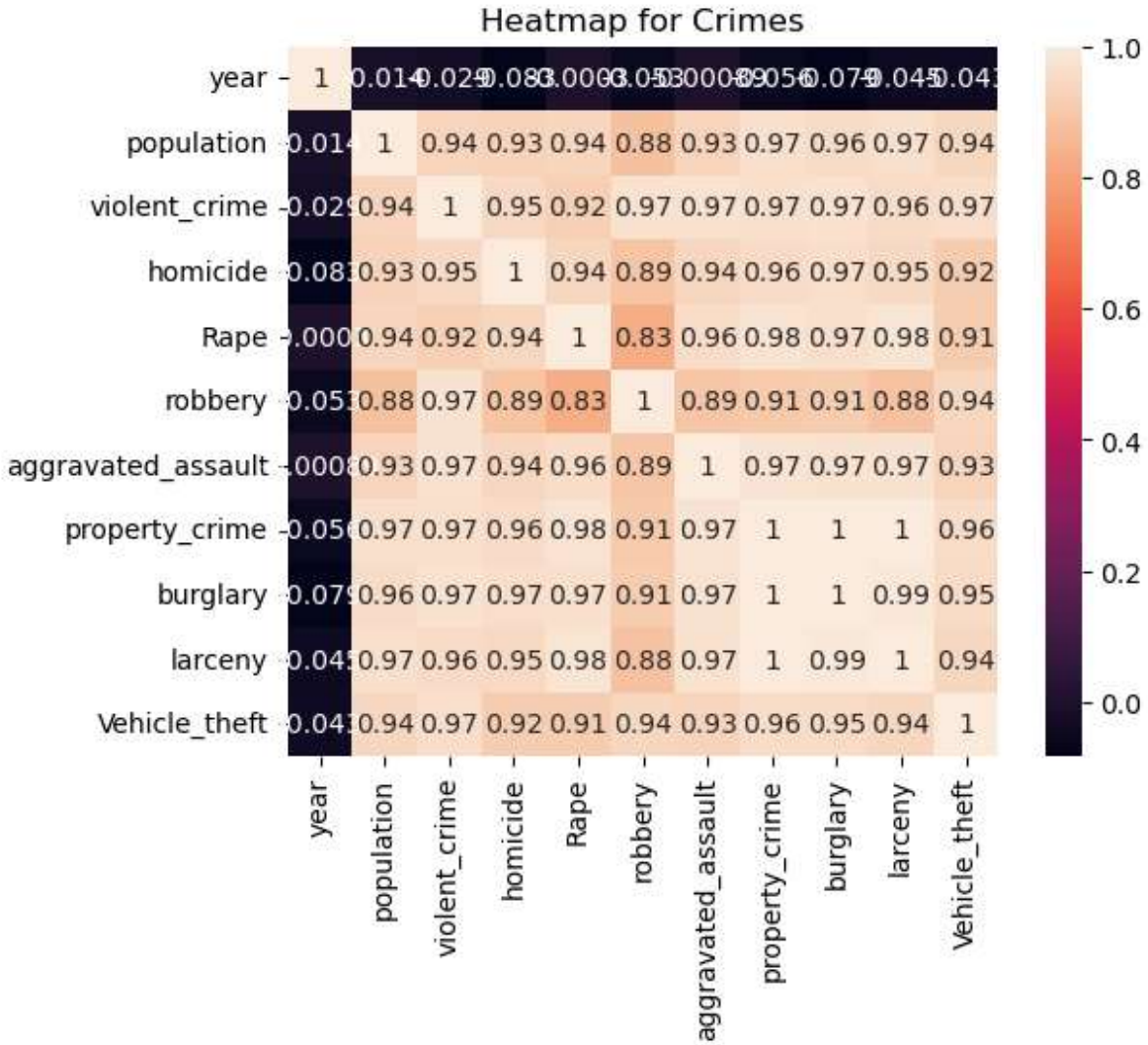
Conclusion :

The graph denotes that homicide was the highest at the year 1981 and was the lowest at the year 1979

HEATMAP

In [17]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv("reduced_output_file.csv")
cor1 = df.corr(numeric_only = True)
plt.imshow(cor1,cmap="YlGn")
plt.title("Heatmap for Crimes")
sns.heatmap(cor1, annot=True , cbar_kws={"shrink": 1})
```

Out[17]: <Axes: title={'center': 'Heatmap for Crimes'}>

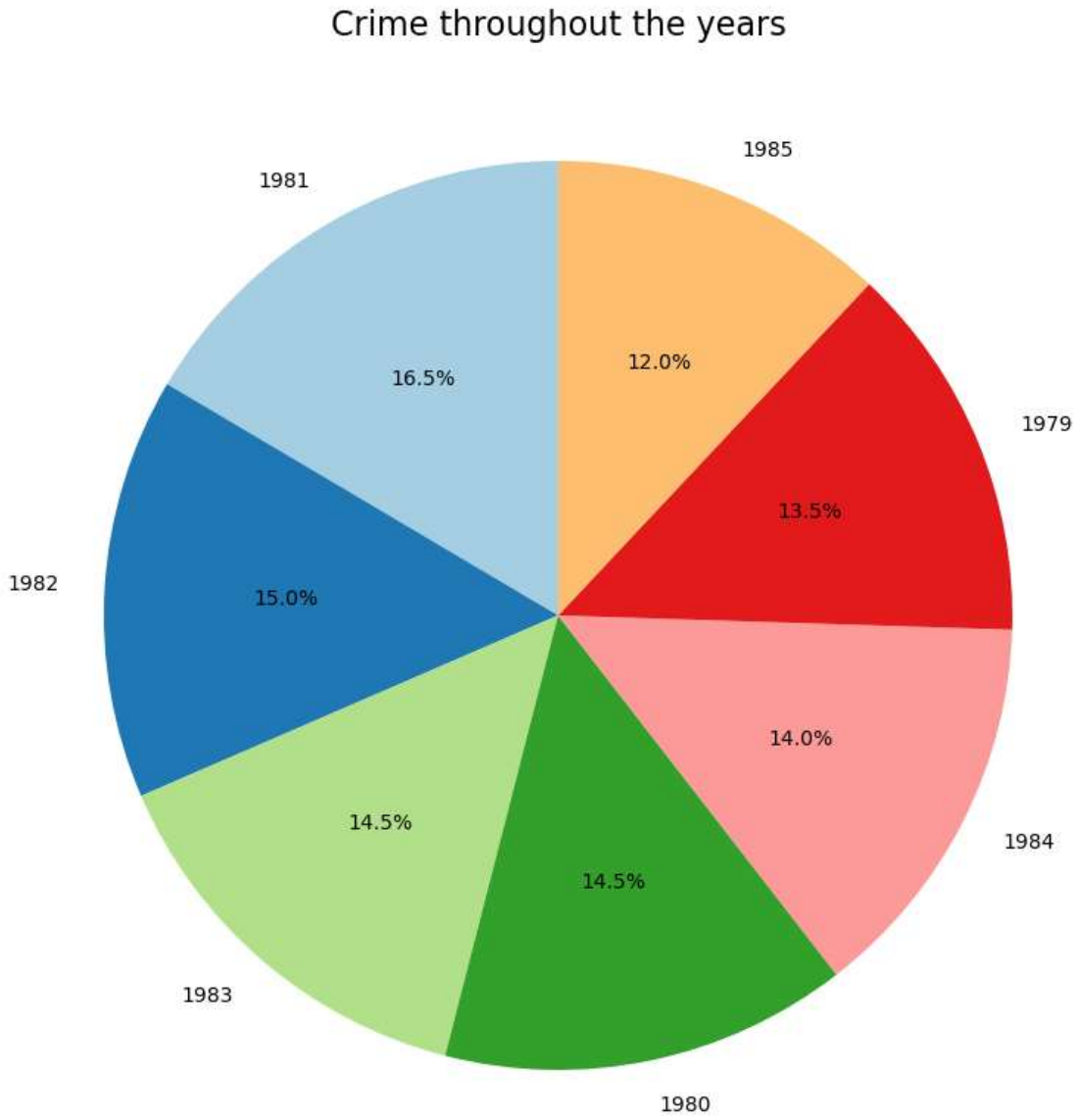


CONCLUSION

PIE CHART

In [20]:

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('reduced_output_file.csv')
pollutant_avg_counts = df['year'].value_counts()

plt.labels={''}
plt.figure(figsize=(12, 10))
plt.pie(pollutant_avg_counts, labels=pollutant_avg_counts.index, autopct=
plt.title('Crime throughout the years', fontsize=16)
plt.show()


```



BAR PLOT

conclusion: It draws a pie chart to show case the customer index and we can see an increment and decrement throughtout the year

In [21]:

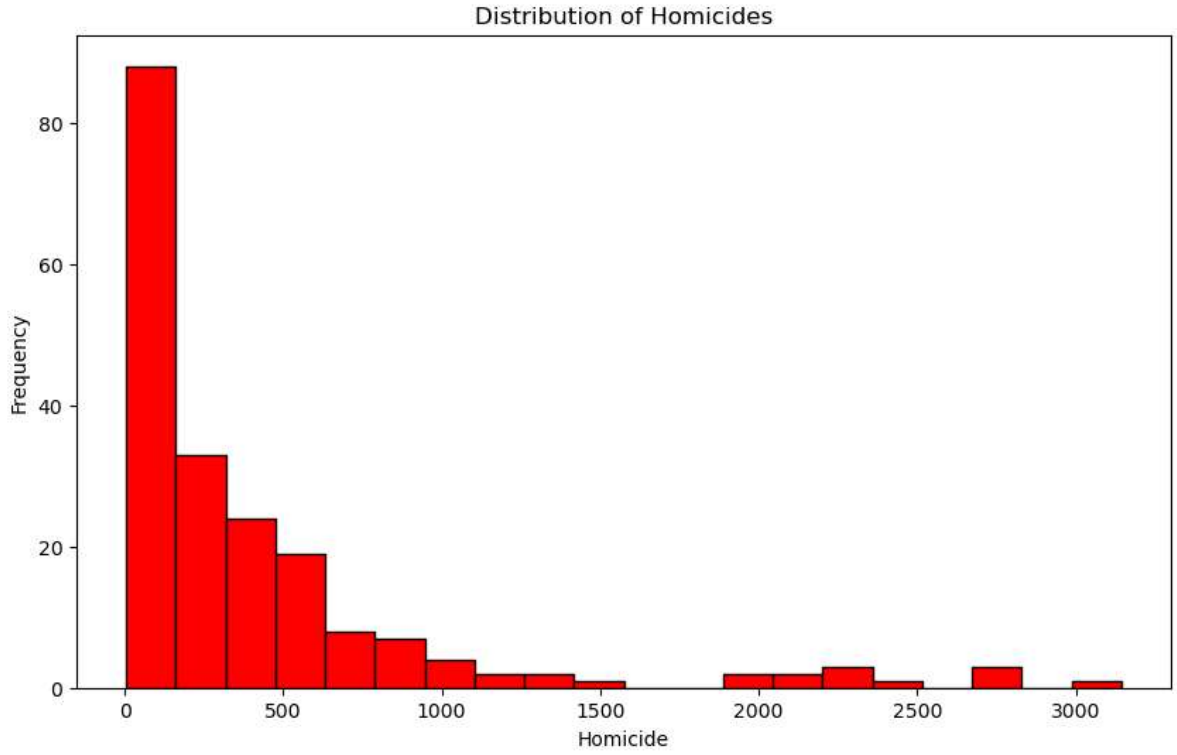
```
import pandas as pd
import plotly.express as px
import numpy as np

# Read the CSV file
df = pd.read_csv('reduced_output_file.csv')

# Calculate percentiles for the 'year' column
percentiles = np.percentile(df['year'], [25, 50, 75])

# Create a boxplot
fig = px.box(df, x='year', points='all', labels={'year': 'homicide'})
fig.update_traces(marker=dict(color='grey'))

# Add annotations for percentiles
for percentile, label in zip(percentiles, ['25th Percentile', '50th Percer
    fig.add_annotation(
        x=percentile, y=0.5,
        text=f'{label}: {percentile}',
        showarrow=True,
        arrowhead=4,
        arrowcolor='black',
        ax=0, ay=-40
    )

# Update layout
fig.update_layout(title_text='homicide percentile', xaxis_title='Year', sh

# Show the plot
fig.show()

```

conclusion:

This code calculates the 25th, 50th (median), and 75th percentiles for the 'year' column in your DataFrame and prints the results. Additionally, it creates a horizontal boxplot to visualize the distribution and percentiles. Adjust the code as needed for your specific requirements

HISTOGRAM

In [22]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.hist(df['homicide'], bins=20, color='red', edgecolor='black')
plt.xlabel('Homicide')
plt.ylabel('Frequency')
plt.title('Distribution of Homicides')
plt.show()

```



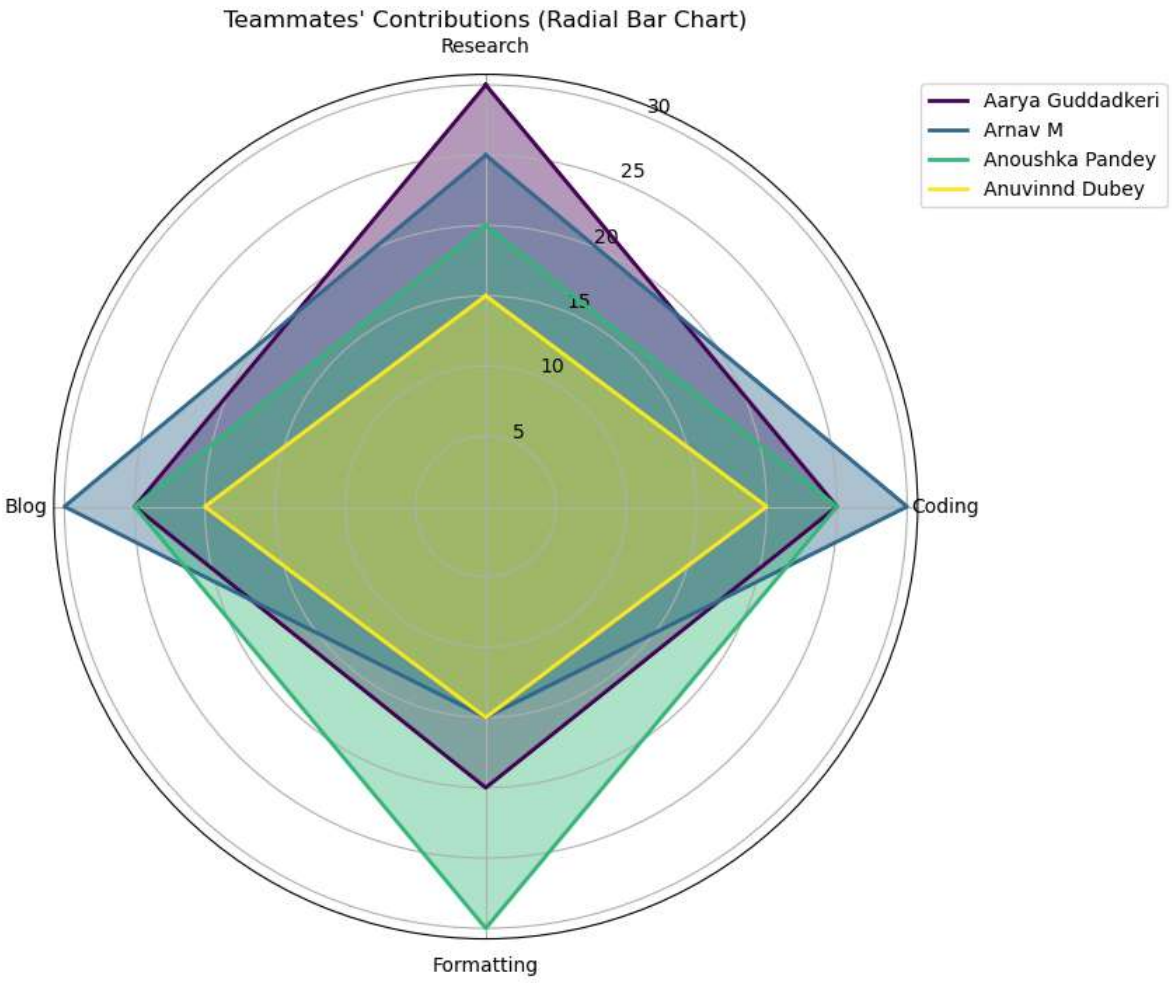
Conclusion:

the frequency distribution of homicides across various states in the United States demonstrates a general downward trend. From the early 1990s to the mid-2000s, the frequency of homicides ranged from approximately 20 per 100,000 in the mid-1990s to a low of around 4.7 per 100,000 in 2004.

CONTRIBUTION:

In [23]:

```
import matplotlib.pyplot as plt
import numpy as np
teammates = ['Aarya Guddadkeri', 'Arnav M', 'Anoushka Pandey', 'Anuvinnd D
categories = ['Research', 'Coding', 'Formatting', 'Blog']
contributions = {
    'Aarya Guddadkeri': [30, 25, 20, 25],
    'Arnav M': [25, 30, 15, 30],
    'Anoushka Pandey': [20, 25, 30, 25],
    'Anuvinnd Dubey': [15, 20, 15, 20]
}

colors = plt.cm.viridis(np.linspace(0, 1, len(categories)))

angles = np.linspace(0, 2 * np.pi, len(categories), endpoint=False).tolist
angles += angles[:1]

fig, ax = plt.subplots(figsize=(8, 8), subplot_kw=dict(polar=True))
ax.set_theta_offset(np.pi / 2)
ax.set_theta_direction(-1)

for i, teammate in enumerate(teammates):
    values = contributions[teammate]
    values += values[:1]
    ax.plot(angles, values, linewidth=2, linestyle='solid', label=teammate
    ax.fill(angles, values, color=colors[i], alpha=0.4)
ax.set_xticks(angles[:-1])
ax.set_xticklabels(categories)
plt.title('Teammates\' Contributions (Radial Bar Chart)')
plt.legend(loc='upper right', bbox_to_anchor=(1.3, 1))

plt.show()
```


Teammates' Contributions (Radial Bar Chart)
Research
Coding
Formatting
Blog
30
25
20
15
10
5
Aarya Guddadkeri
Arnav M
Anoushka Pandey
Anuvinnd Dubey

In [ ]:
```
1
```