Description:

This dataset contains district-wise data on the number of women teachers working in middle schools in Punjab over multiple years. It includes various districts as well as the overall total for Punjab. The data has been cleaned by replacing missing values with 0 and ensuring all numerical values are properly formatted. Several visualization techniques have been applied to analyze trends, distributions, and variations in the dataset.

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df = pd.read_csv('District-wise number of women teachers working in middle
df
```

Out[3]:

	Year/District	2017	2016	2015	2014	2013	2012	2011	2010	2009	...	
0	Gurdaspur	2076	2155	2084	2148	1571	2175	4722	4926.0	4069.0	...	3
1	Pathankot	782	765	761	775	612	745	2507	NaN	NaN	...	
2	Amritsar	2187	2263	2316	2213	1481	2260	3952	3151.0	3532.0	...	5
3	Tarn Taran	1056	1090	946	1016	917	1099	680	1557.0	1474.0	...	
4	Kapurthala	1063	1148	1049	1097	643	1123	1801	2928.0	1407.0	...	1
5	Jalandhar	2010	2110	2119	2069	1301	2155	2807	4210.0	2419.0	...	2
6	SBS Nagar	673	724	693	720	524	811	841	926.0	904.0	...	
7	Hoshiarpur	1984	2108	2048	2104	1488	2160	2227	5900.0	2576.0	...	2
8	Rupnagar	804	843	790	835	599	1827	1733	1064.0	1011.0	...	1
9	SAS Nagar	1288	1268	1115	1116	413	1409	1538	1660.0	1201.0	...	
10	Ludhiana	2568	2737	2683	2604	1967	2633	4948	5382.0	4613.0	...	3
11	Ferozepur	965	938	788	869	933	771	1844	3188.0	2660.0	...	2
12	Fazilka	802	783	775	768	1321	911	786	NaN	NaN	...	
13	Faridkot	656	668	662	684	550	676	1802	1290.0	1060.0	...	
14	Sri Muktsar Sahib	831	827	738	771	785	1080	1900	1472.0	1411.0	...	
15	Moga	983	1109	908	920	730	1000	2003	1772.0	1601.0	...	1
16	Bathinda	1171	1103	1071	1090	891	1105	3163	2338.0	2076.0	...	1
17	Mansa	660	625	600	622	745	692	1393	1007.0	903.0	...	
18	Sangrur	1387	1321	1326	1372	1403	862	2146	2346.0	1643.0	...	1
19	Barnala	476	450	452	474	425	517	1016	817.0	529.0	...	
20	Patiala	1875	1788	1784	1820	1421	775	3441	3099.0	2129.0	...	2
21	Fatehgarh Sahib	760	762	747	765	548	782	1307	1126.0	775.0	...	
22	Punjab	27057	27585	26455	26852	21268	27568	48557	50159.0	37993.0	...	33

23 rows × 30 columns

In [4]:
```
df.replace({"Year/District": {"Punjab": "Total"}}, inplace=True)
df
```

Out[4]:

	Year/District	2017	2016	2015	2014	2013	2012	2011	2010	2009	...	
0	Gurdaspur	2076	2155	2084	2148	1571	2175	4722	4926.0	4069.0	...	3
1	Pathankot	782	765	761	775	612	745	2507	NaN	NaN	...	
2	Amritsar	2187	2263	2316	2213	1481	2260	3952	3151.0	3532.0	...	5
3	Tarn Taran	1056	1090	946	1016	917	1099	680	1557.0	1474.0	...	
4	Kapurthala	1063	1148	1049	1097	643	1123	1801	2928.0	1407.0	...	1
5	Jalandhar	2010	2110	2119	2069	1301	2155	2807	4210.0	2419.0	...	2
6	SBS Nagar	673	724	693	720	524	811	841	926.0	904.0	...	
7	Hoshiarpur	1984	2108	2048	2104	1488	2160	2227	5900.0	2576.0	...	2
8	Rupnagar	804	843	790	835	599	1827	1733	1064.0	1011.0	...	1
9	SAS Nagar	1288	1268	1115	1116	413	1409	1538	1660.0	1201.0	...	
10	Ludhiana	2568	2737	2683	2604	1967	2633	4948	5382.0	4613.0	...	3
11	Ferozepur	965	938	788	869	933	771	1844	3188.0	2660.0	...	2
12	Fazilka	802	783	775	768	1321	911	786	NaN	NaN	...	
13	Faridkot	656	668	662	684	550	676	1802	1290.0	1060.0	...	
14	Sri Muktsar Sahib	831	827	738	771	785	1080	1900	1472.0	1411.0	...	
15	Moga	983	1109	908	920	730	1000	2003	1772.0	1601.0	...	1
16	Bathinda	1171	1103	1071	1090	891	1105	3163	2338.0	2076.0	...	1
17	Mansa	660	625	600	622	745	692	1393	1007.0	903.0	...	
18	Sangrur	1387	1321	1326	1372	1403	862	2146	2346.0	1643.0	...	1
19	Barnala	476	450	452	474	425	517	1016	817.0	529.0	...	
20	Patiala	1875	1788	1784	1820	1421	775	3441	3099.0	2129.0	...	2
21	Fatehgarh Sahib	760	762	747	765	548	782	1307	1126.0	775.0	...	
22	Total	27057	27585	26455	26852	21268	27568	48557	50159.0	37993.0	...	33

23 rows × 30 columns

In [5]:
```
df.fillna(0, inplace=True)
```

In [6]:
```
for col in df.columns[1:]:
    df[col] = pd.to_numeric(df[col], errors='coerce').round(0).astype(int)
```

In [7]:
```
df
```

Out[7]:

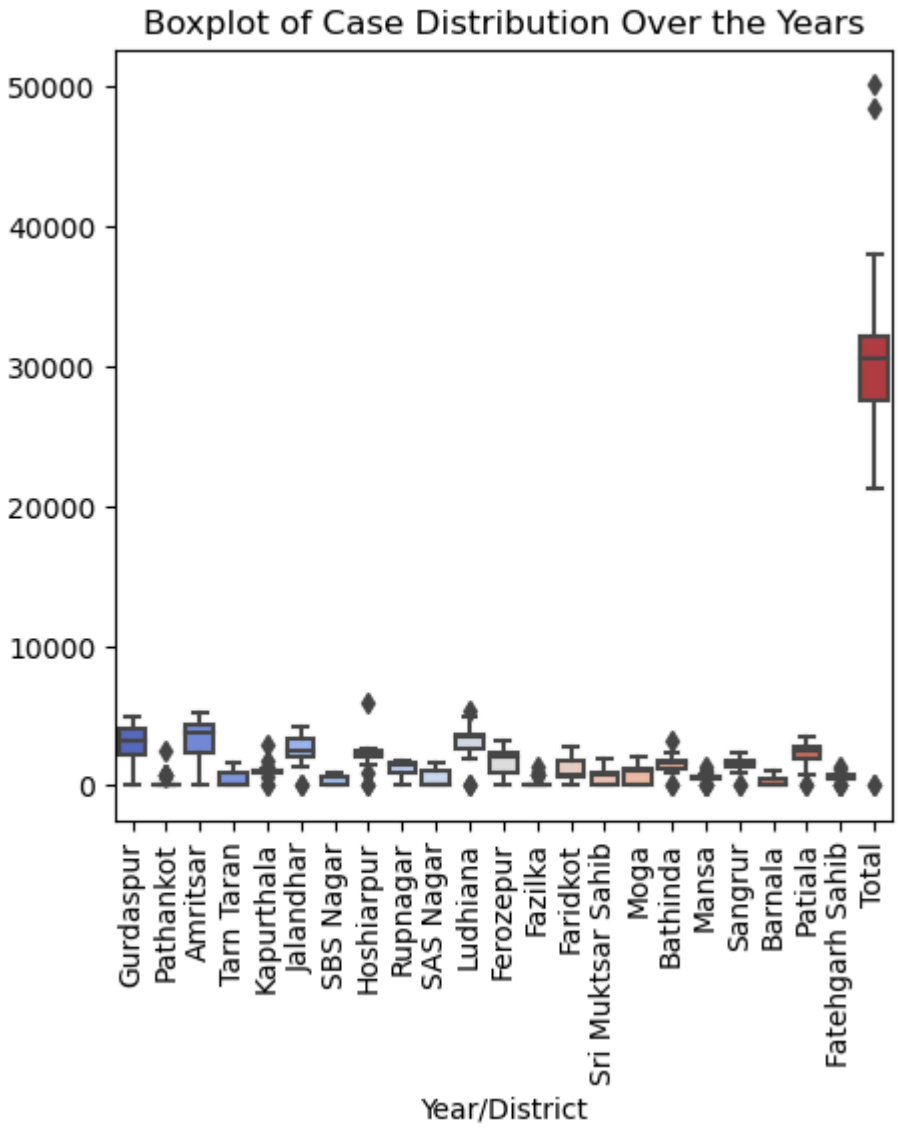
	Year/District	2017	2016	2015	2014	2013	2012	2011	2010	2009	...	1998
0	Gurdaspur	2076	2155	2084	2148	1571	2175	4722	4926	4069	...	3408
1	Pathankot	782	765	761	775	612	745	2507	0	0	...	0
2	Amritsar	2187	2263	2316	2213	1481	2260	3952	3151	3532	...	5190
3	Tarn Taran	1056	1090	946	1016	917	1099	680	1557	1474	...	0
4	Kapurthala	1063	1148	1049	1097	643	1123	1801	2928	1407	...	1070
5	Jalandhar	2010	2110	2119	2069	1301	2155	2807	4210	2419	...	2726
6	SBS Nagar	673	724	693	720	524	811	841	926	904	...	565
7	Hoshiarpur	1984	2108	2048	2104	1488	2160	2227	5900	2576	...	2480
8	Rupnagar	804	843	790	835	599	1827	1733	1064	1011	...	1680
9	SAS Nagar	1288	1268	1115	1116	413	1409	1538	1660	1201	...	0
10	Ludhiana	2568	2737	2683	2604	1967	2633	4948	5382	4613	...	3941
11	Ferozepur	965	938	788	869	933	771	1844	3188	2660	...	2419
12	Fazilka	802	783	775	768	1321	911	786	0	0	...	0
13	Faridkot	656	668	662	684	550	676	1802	1290	1060	...	824
14	Sri Muktsar Sahib	831	827	738	771	785	1080	1900	1472	1411	...	895
15	Moga	983	1109	908	920	730	1000	2003	1772	1601	...	1143
16	Bathinda	1171	1103	1071	1090	891	1105	3163	2338	2076	...	1728
17	Mansa	660	625	600	622	745	692	1393	1007	903	...	564
18	Sangrur	1387	1321	1326	1372	1403	862	2146	2346	1643	...	1818
19	Barnala	476	450	452	474	425	517	1016	817	529	...	0
20	Patiala	1875	1788	1784	1820	1421	775	3441	3099	2129	...	2804
21	Fatehgarh Sahib	760	762	747	765	548	782	1307	1126	775	...	516
22	Total	27057	27585	26455	26852	21268	27568	48557	50159	37993	...	33771

23 rows × 30 columns

In [8]:
```
df.info()
```

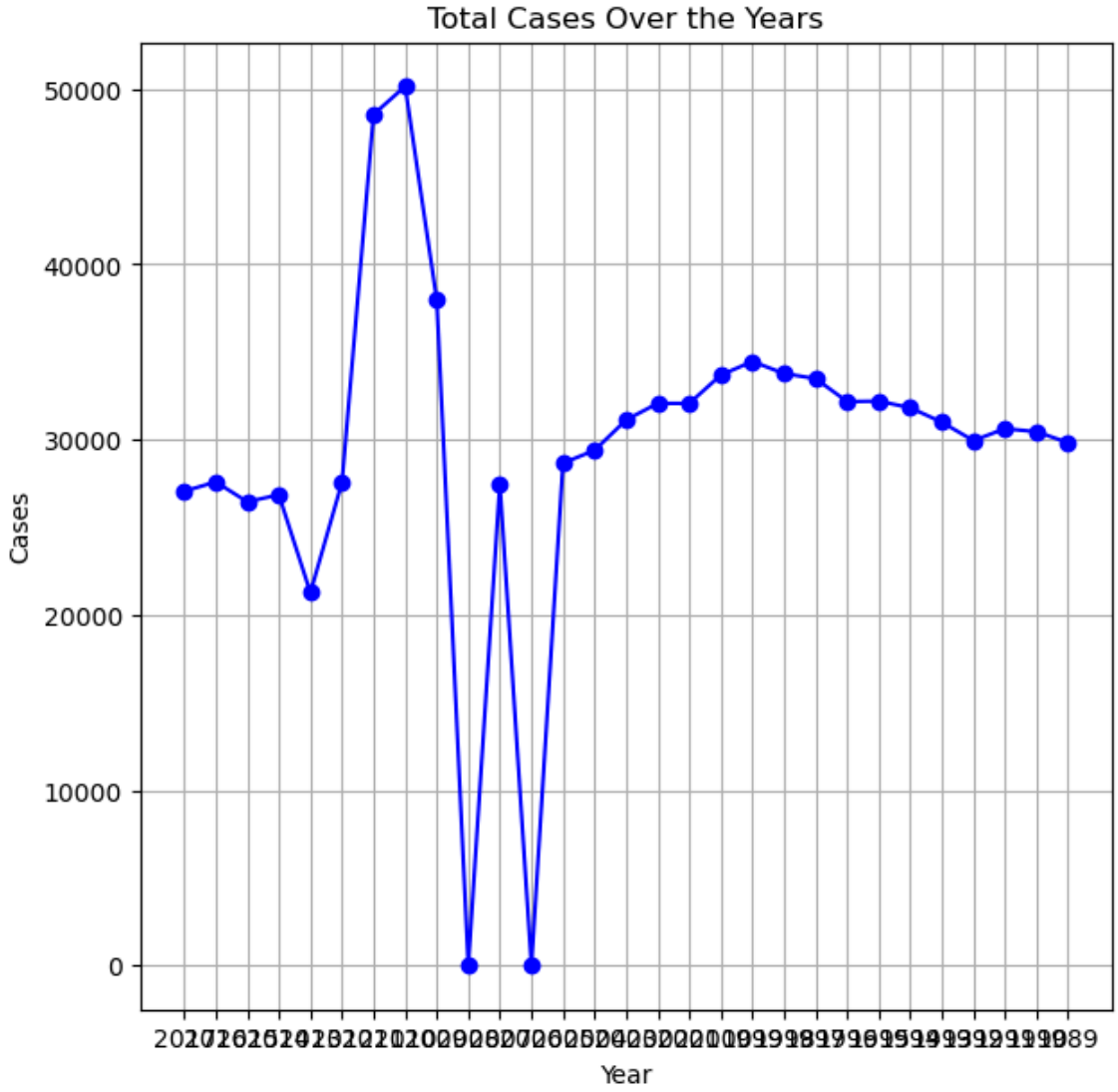
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23 entries, 0 to 22
Data columns (total 30 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Year/District  23 non-null     object
 1   2017           23 non-null     int32 
 2   2016           23 non-null     int32 
 3   2015           23 non-null     int32 
 4   2014           23 non-null     int32 
 5   2013           23 non-null     int32 
 6   2012           23 non-null     int32 
 7   2011           23 non-null     int32 
 8   2010           23 non-null     int32 
 9   2009           23 non-null     int32 
 10  2008           23 non-null     int32 
 11  2007           23 non-null     int32 
 12  2006           23 non-null     int32 
 13  2005           23 non-null     int32 
 14  2004           23 non-null     int32 
 15  2003           23 non-null     int32 
 16  2002           23 non-null     int32 
 17  2001           23 non-null     int32 
 18  2000           23 non-null     int32 
 19  1999           23 non-null     int32 
 20  1998           23 non-null     int32 
 21  1997           23 non-null     int32 
 22  1996           23 non-null     int32 
 23  1995           23 non-null     int32 
 24  1994           23 non-null     int32 
 25  1993           23 non-null     int32 
 26  1992           23 non-null     int32 
 27  1991           23 non-null     int32 
 28  1990           23 non-null     int32 
 29  1989           23 non-null     int32 
dtypes: int32(29), object(1)
memory usage: 2.9+ KB
```

In [9]:
```
plt.figure(figsize=(5, 5))
sns.boxplot(data=df.set_index("Year/District").T, palette="coolwarm")
plt.xticks(rotation=90)
plt.title("Boxplot of Case Distribution Over the Years")
plt.show()
```



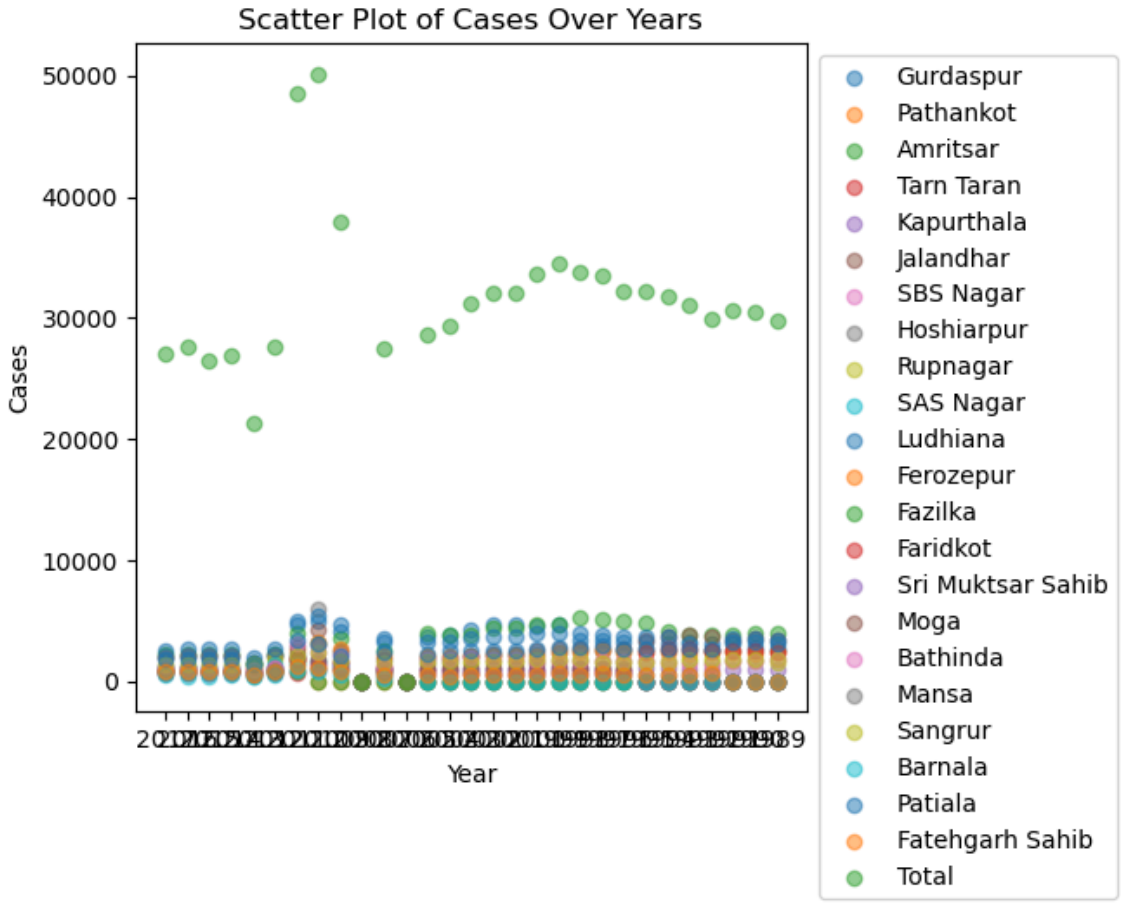
This creates a boxplot to visualize the distribution of cases over the years

In [10]:
```
plt.figure(figsize=(7, 7))
plt.plot(df.columns[1:], df[df['Year/District'] == 'Total'].values.flatten(
plt.title("Total Cases Over the Years")
plt.xlabel("Year")
plt.ylabel("Cases")
plt.grid(True)
plt.show()
```



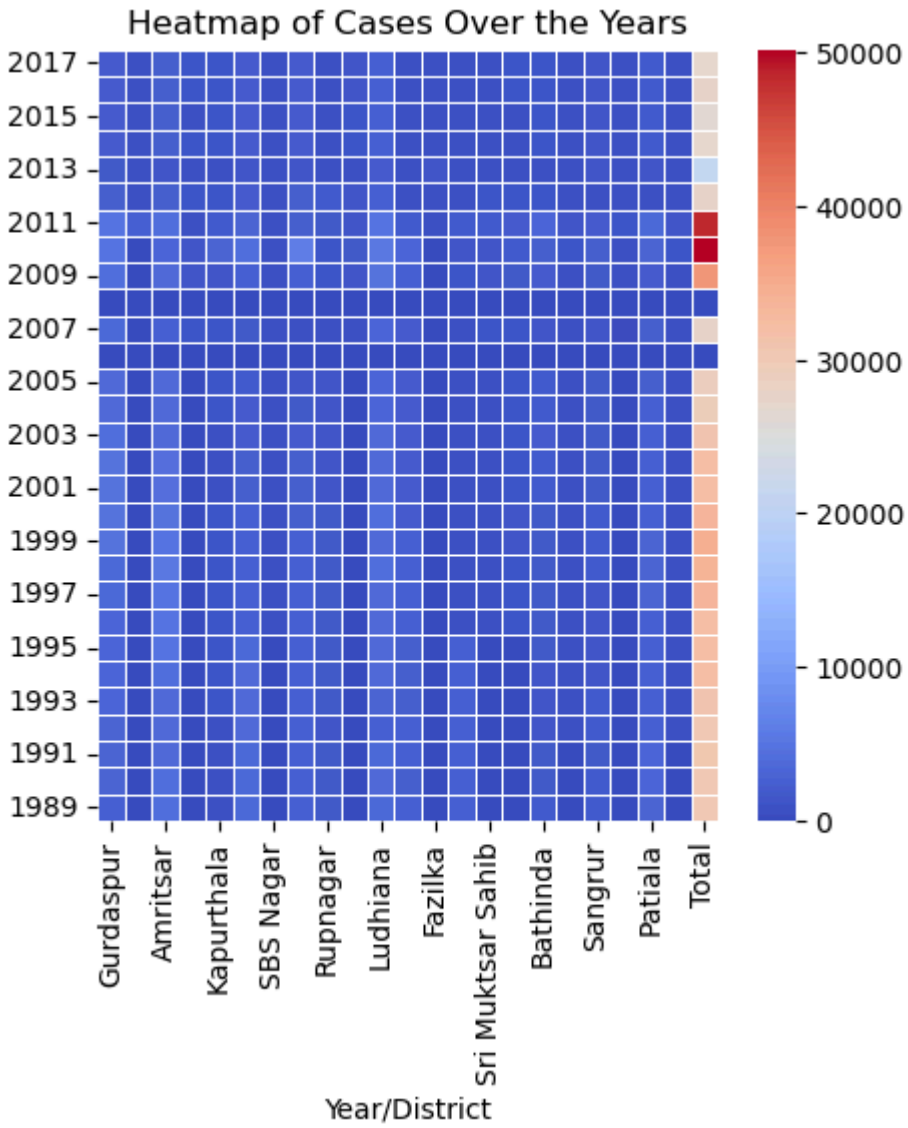
This plots the total cases over the years using a line graph with markers, showing trends in case counts over time.

In [11]:
```
plt.figure(figsize=(5, 5))
for district in df["Year/District"]:
    plt.scatter(df.columns[1:], df[df["Year/District"] == district].values.
plt.title("Scatter Plot of Cases Over Years")
plt.xlabel("Year")
plt.ylabel("Cases")
plt.legend(loc='upper left', bbox_to_anchor=(1,1))
plt.show()
```



It generates a scatter plot showing the distribution of cases over the years for each district, with different markers representing different districts.

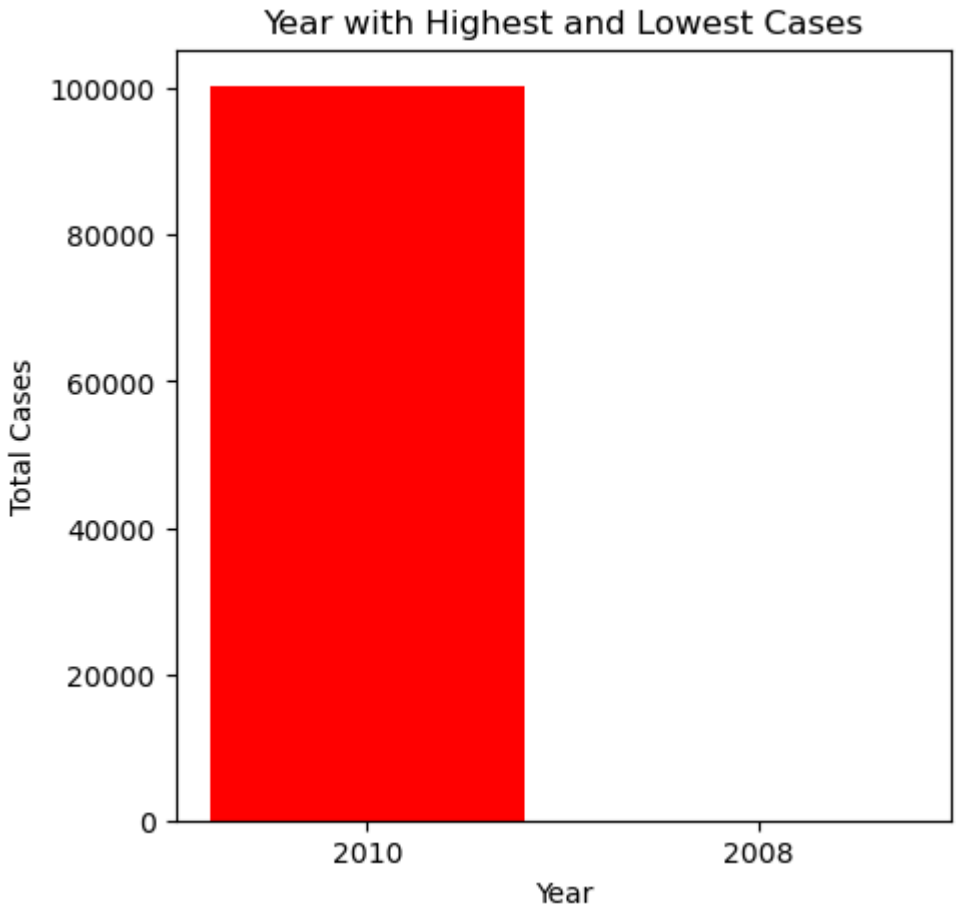
In [12]:
```
plt.figure(figsize=(5, 5))
sns.heatmap(df.set_index("Year/District").T, cmap='coolwarm', annot=False,
plt.title("Heatmap of Cases Over the Years")
plt.show()
```



It creates a heatmap to visualize the distribution and intensity of cases over the years across different districts using a color gradient.

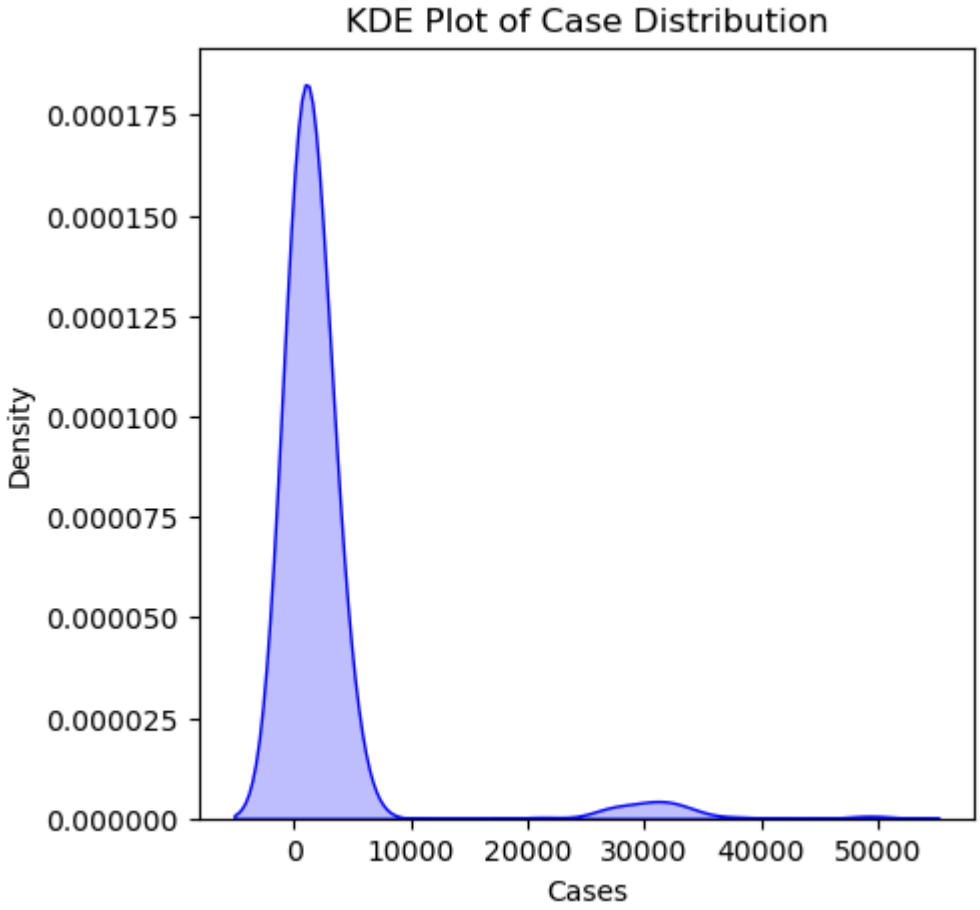
In [13]:
```
highest_year = df.set_index("Year/District").sum().idxmax()
lowest_year = df.set_index("Year/District").sum().idxmin()

plt.figure(figsize=(5, 5))
plt.bar([highest_year, lowest_year], [df.set_index("Year/District")[highest
plt.title("Year with Highest and Lowest Cases")
plt.xlabel("Year")
plt.ylabel("Total Cases")
plt.show()
```



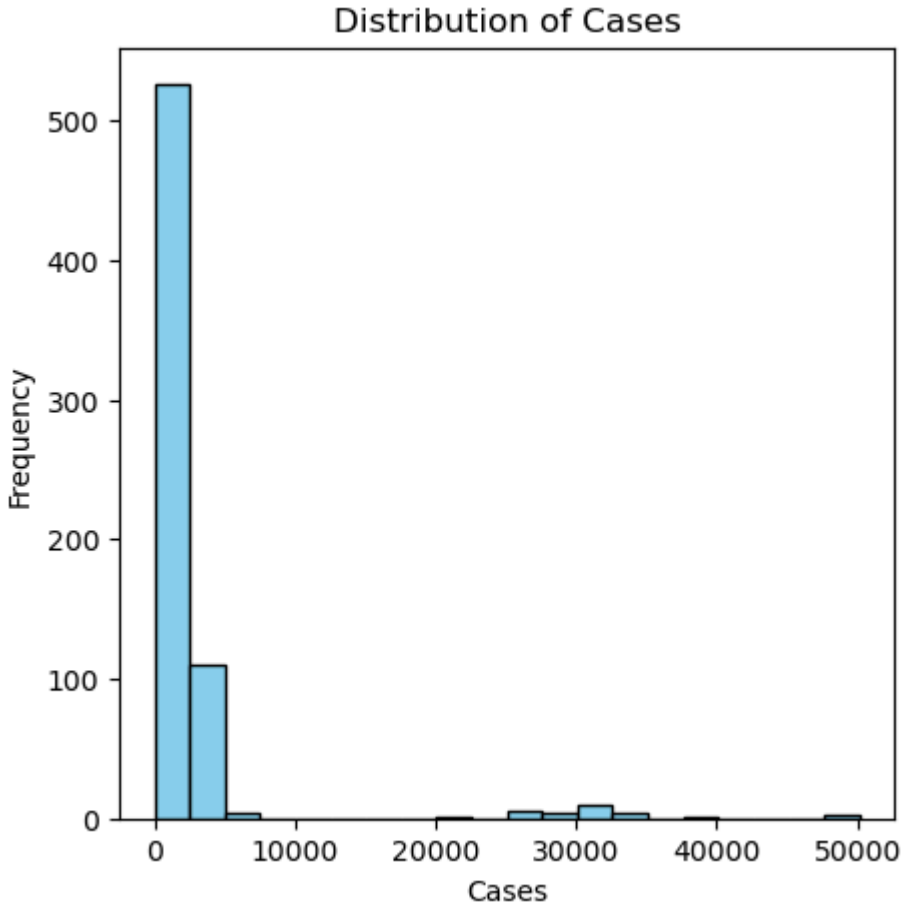
It identifies the years with the highest and lowest total cases and visualizes them using a bar chart with red and green bars.

In [14]:
```
plt.figure(figsize=(5, 5))
sns.kdeplot(df.set_index("Year/District").values.flatten(), fill=True, colo
plt.title("KDE Plot of Case Distribution")
plt.xlabel("Cases")
plt.show()
```



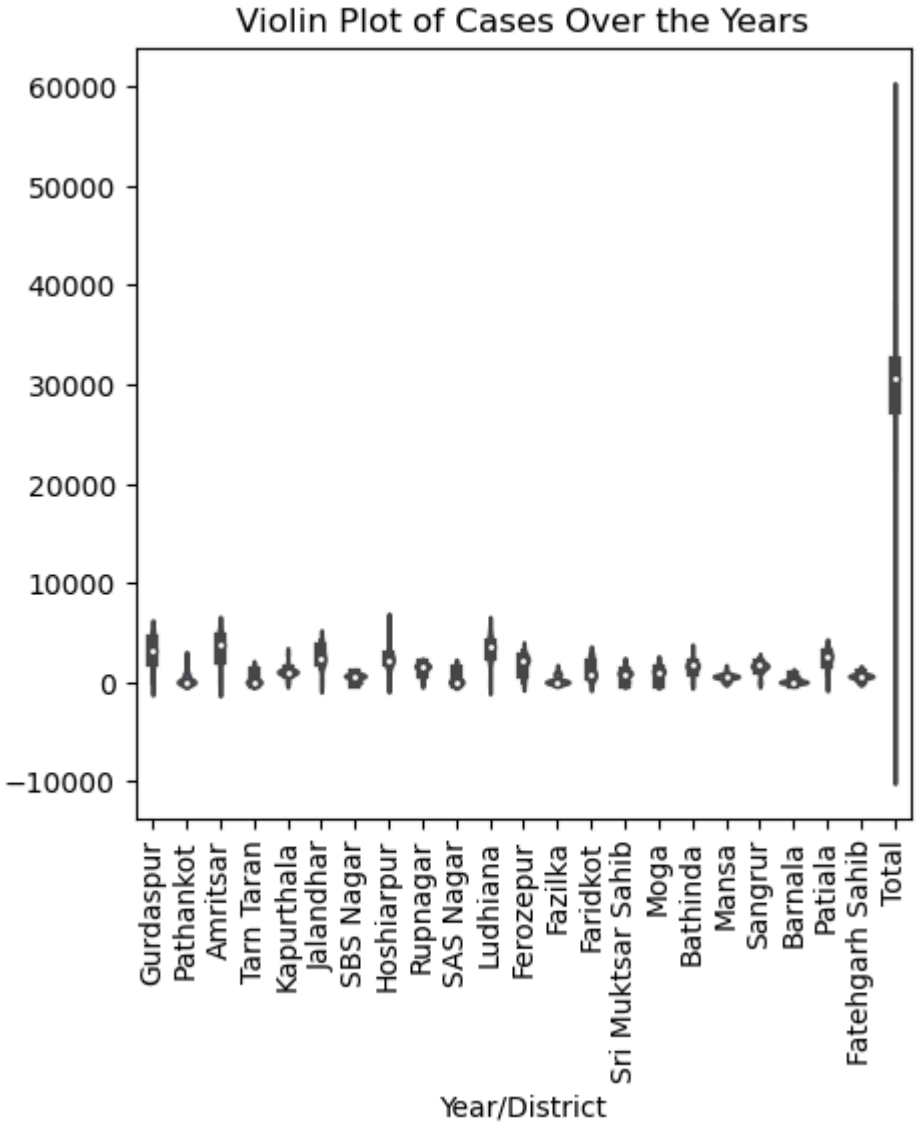
It generates a Kernel Density Estimate (KDE) plot to visualize the distribution of case values across all years and districts, highlighting density variations smoothly.

In [15]:
```
plt.figure(figsize=(5, 5))
plt.hist(df.set_index("Year/District").values.flatten(), bins=20, color='sk
plt.title("Distribution of Cases")
plt.xlabel("Cases")
plt.ylabel("Frequency")
plt.show()
```

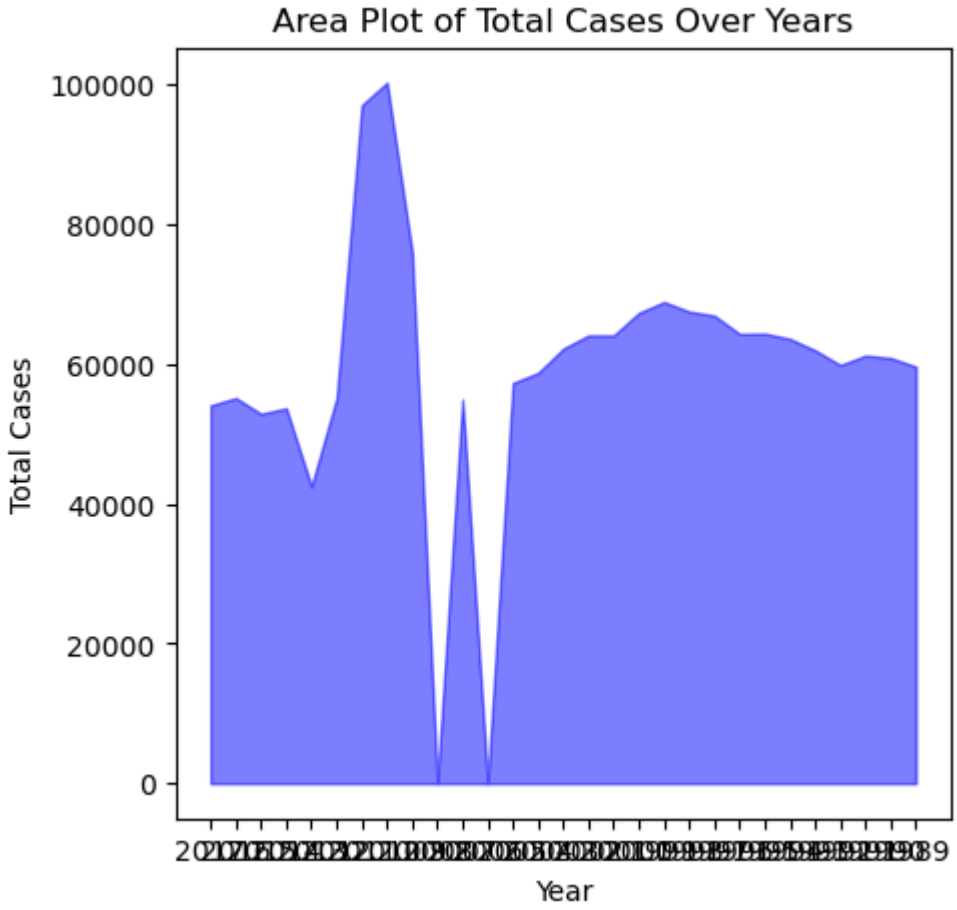


This creates a histogram to visualize the distribution of case counts, showing how frequently different case values occur.

In [16]:
```
plt.figure(figsize=(5, 5))
sns.violinplot(data=df.set_index("Year/District").T, palette="coolwarm")
plt.xticks(rotation=90)
plt.title("Violin Plot of Cases Over the Years")
plt.show()
```



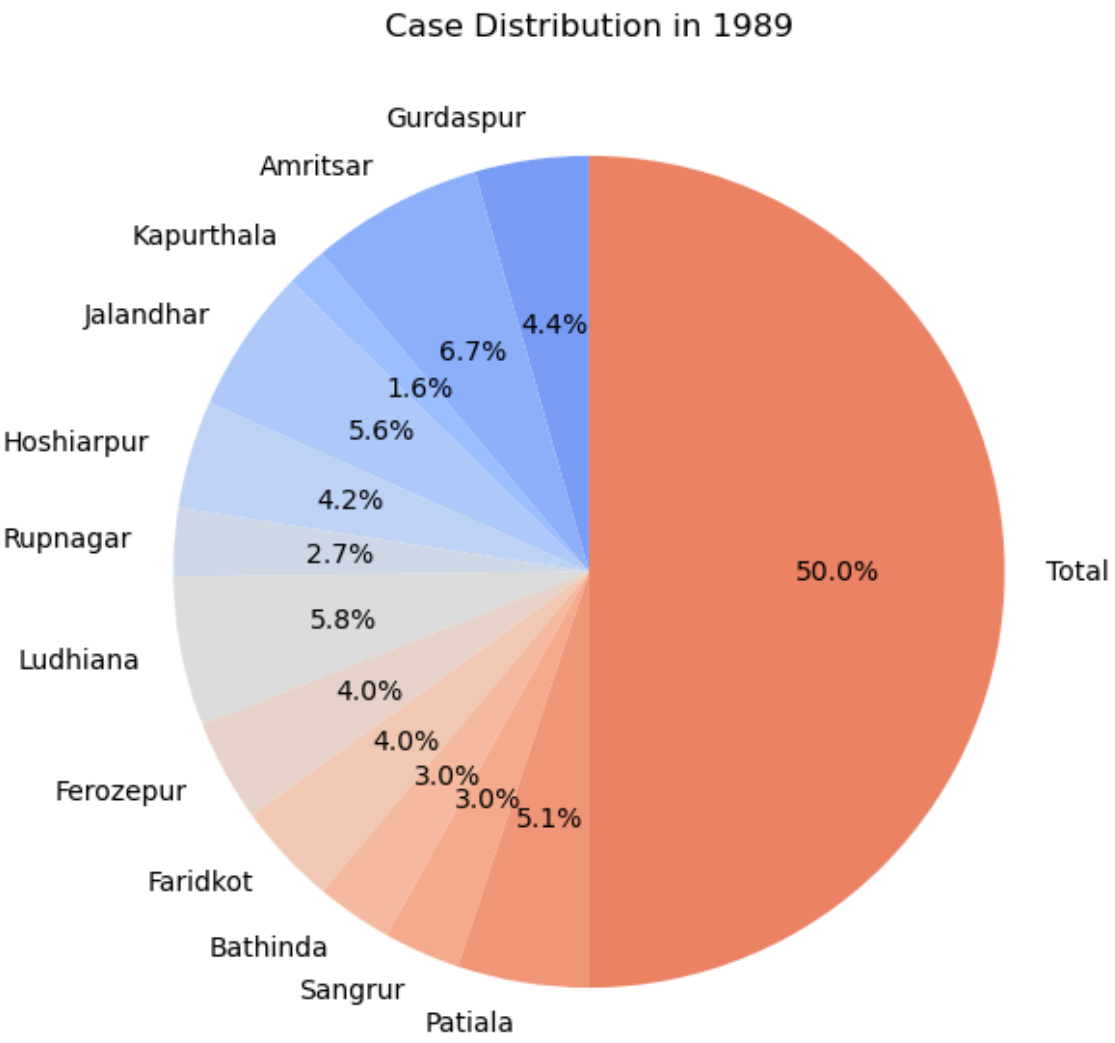
In [17]:
```
plt.figure(figsize=(5, 5))
plt.fill_between(df.set_index("Year/District").columns, df.set_index("Year/
plt.title("Area Plot of Total Cases Over Years")
plt.xlabel("Year")
plt.ylabel("Total Cases")
plt.show()
```



It creates an area plot showing the total cases over the years, highlighting trends with a shaded region.

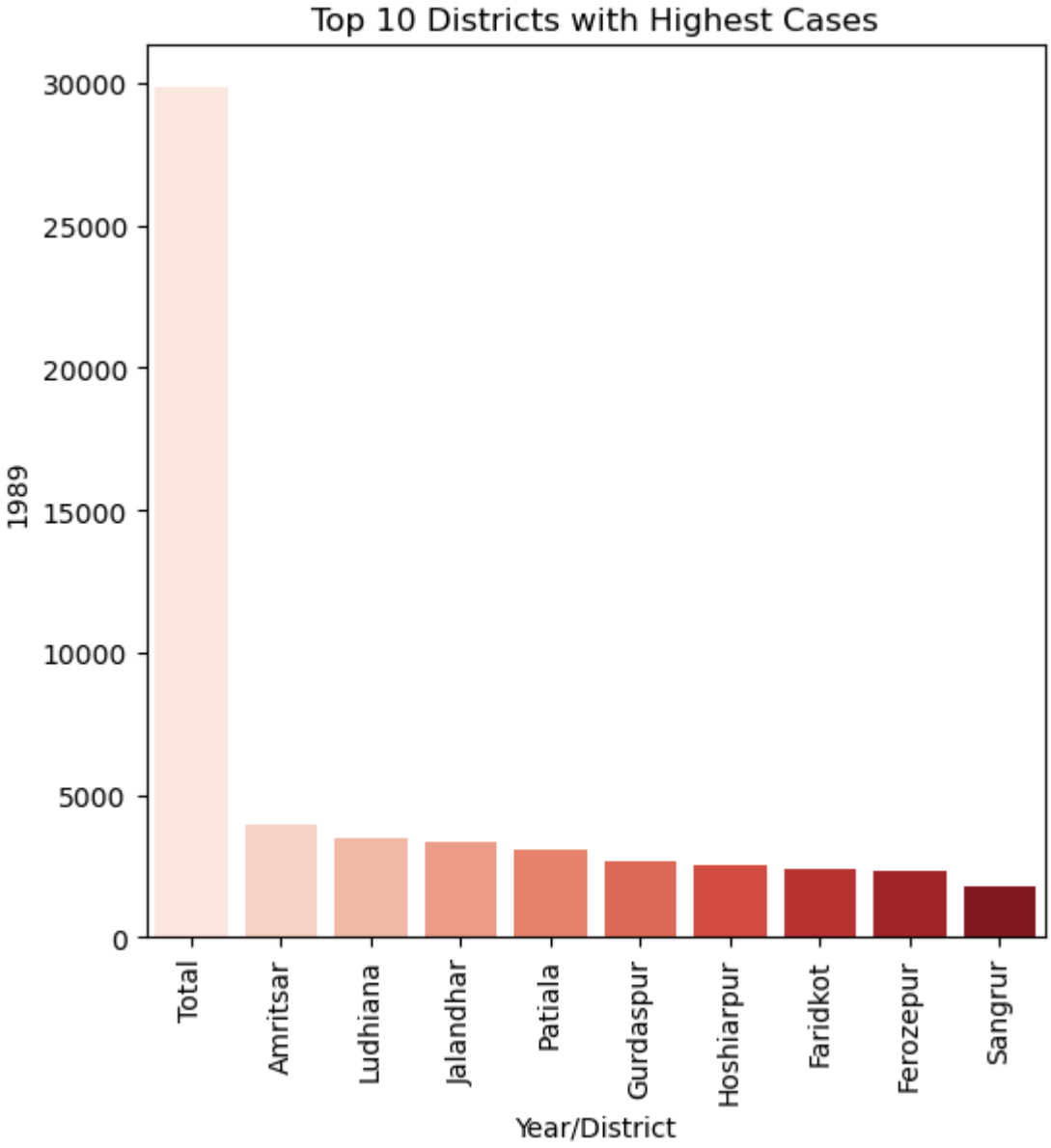
In [18]:
```
plt.figure(figsize=(7, 7))
year = df.columns[-1]
pie_data = df.set_index("Year/District")[year]
pie_data = pie_data[pie_data > 0]
colors = plt.get_cmap("coolwarm")(np.linspace(0.2, 0.8, len(pie_data)))

plt.pie(pie_data, labels=pie_data.index, autopct='%1.1f%%', startangle=90,
plt.title(f"Case Distribution in {year}")
plt.show()
```



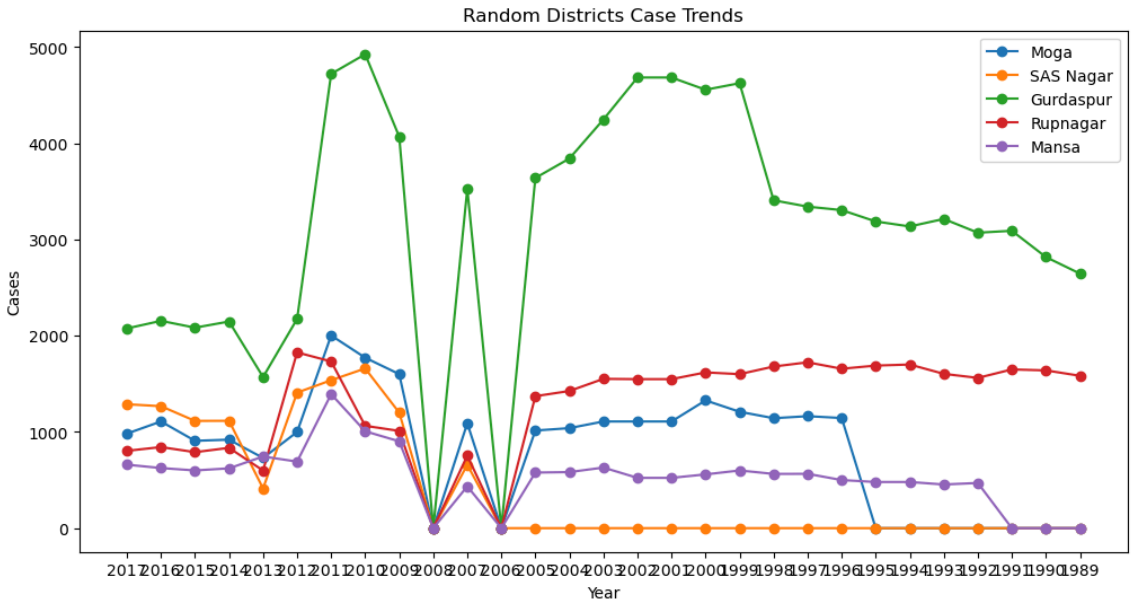
It generates a pie chart showing the distribution of cases across districts for the most recent year in the dataset, using a "coolwarm" color map.

In [19]:
```
plt.figure(figsize=(6, 6))
df_sorted = df.sort_values(by=df.columns[-1], ascending=False)
sns.barplot(x=df_sorted["Year/District"].head(10), y=df_sorted[df.columns[-
plt.xticks(rotation=90)
plt.title("Top 10 Districts with Highest Cases")
plt.show()
```



It creates a bar plot of the top 10 districts with the highest cases in the most recent year, sorting them in descending order.

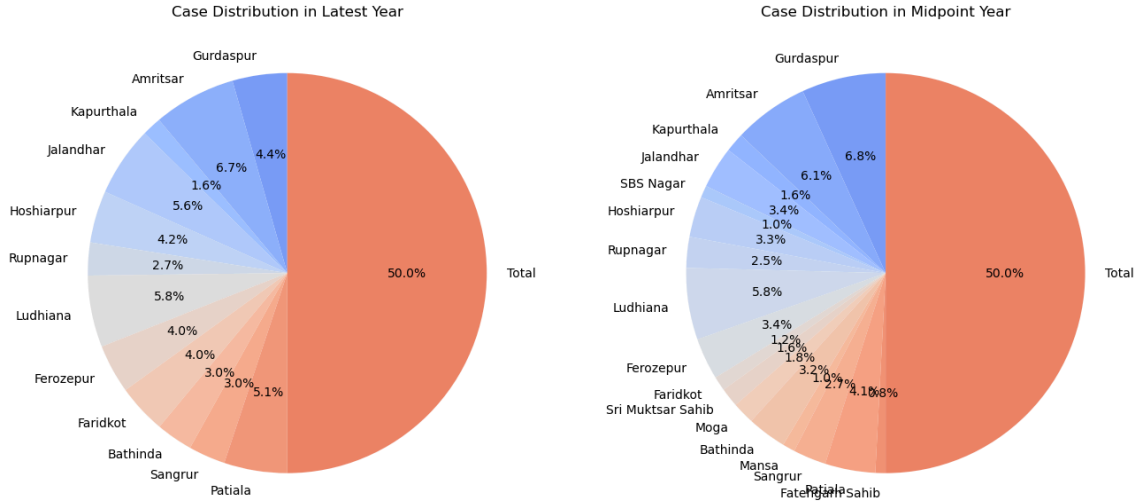
In [20]:
```
plt.figure(figsize=(12, 6))
districts = df["Year/District"].sample(5, random_state=42)  # Random 5 dist
for district in districts:
    plt.plot(df.columns[1:], df[df["Year/District"] == district].values.fla
plt.title("Random Districts Case Trends")
plt.xlabel("Year")
plt.ylabel("Cases")
plt.legend()
plt.show()
```



It randomly selects 5 districts and plots their case trends over the years using a line plot.

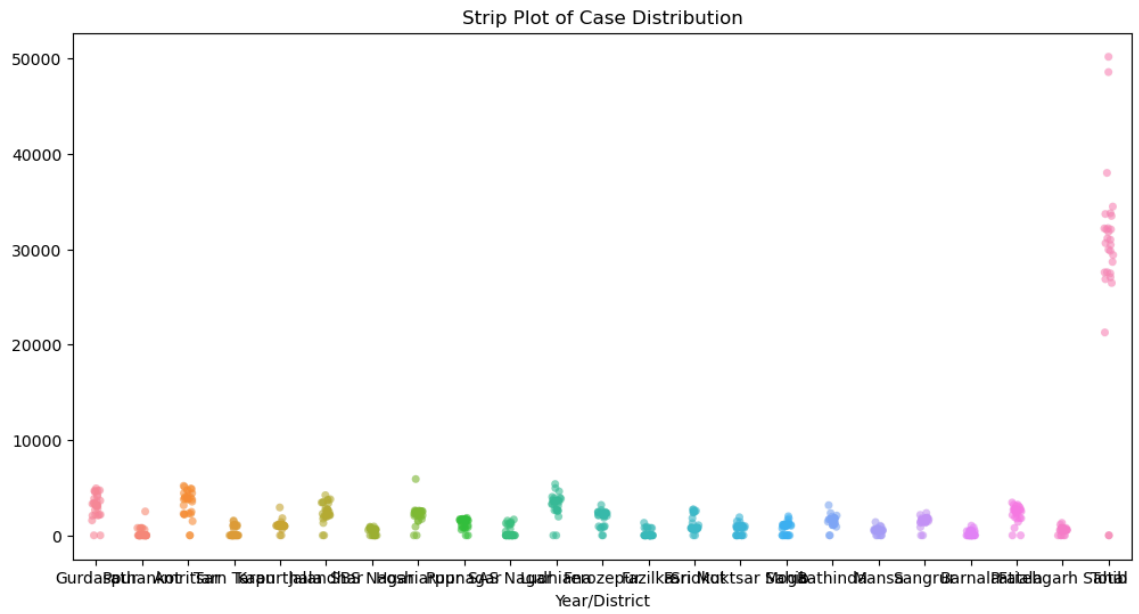
In [21]:
```
years = [df.columns[-1], df.columns[len(df.columns)//2]]
titles = ["Latest Year", "Midpoint Year"]

plt.figure(figsize=(16, 8))
for i, (year, title) in enumerate(zip(years, titles), start=1):
    plt.subplot(1, 2, i)
    pie_data = df.set_index("Year/District")[year]
    pie_data = pie_data[pie_data > 0]
    colors = plt.get_cmap("coolwarm")(np.linspace(0.2, 0.8, len(pie_data)))
    plt.pie(pie_data, labels=pie_data.index, autopct='%1.1f%%', startangle=
    plt.title(f"Case Distribution in {title}")
plt.show()
```



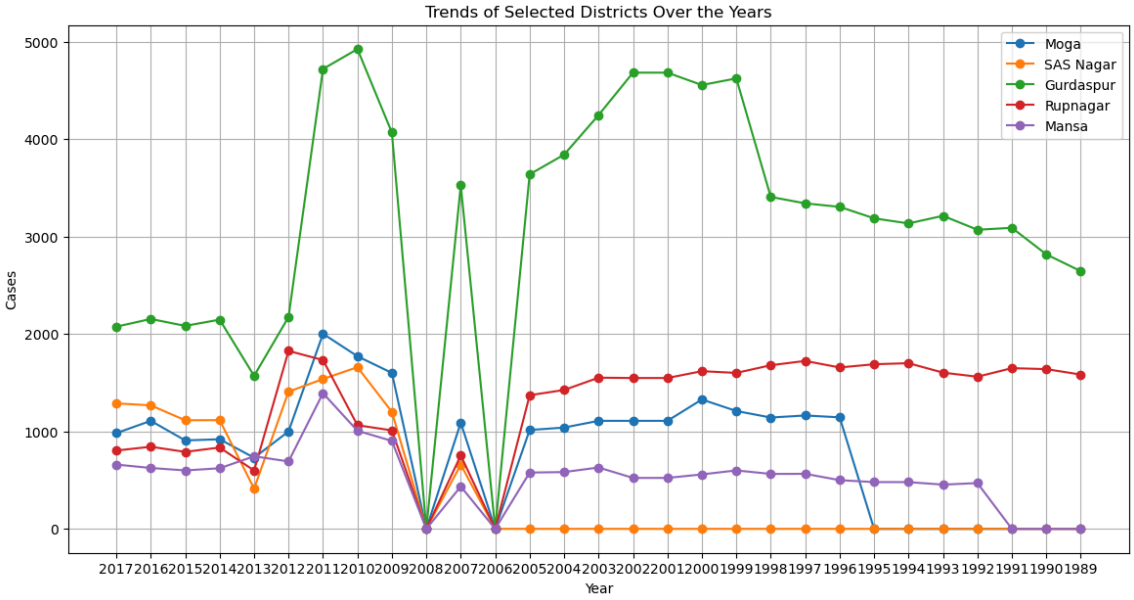
It generates side-by-side pie charts showing the distribution of cases for the latest and midpoint years.

In [22]:
```
plt.figure(figsize=(12, 6))
sns.stripplot(data=df.set_index("Year/District").T, jitter=True, alpha=0.6)
plt.title("Strip Plot of Case Distribution")
plt.show()
```



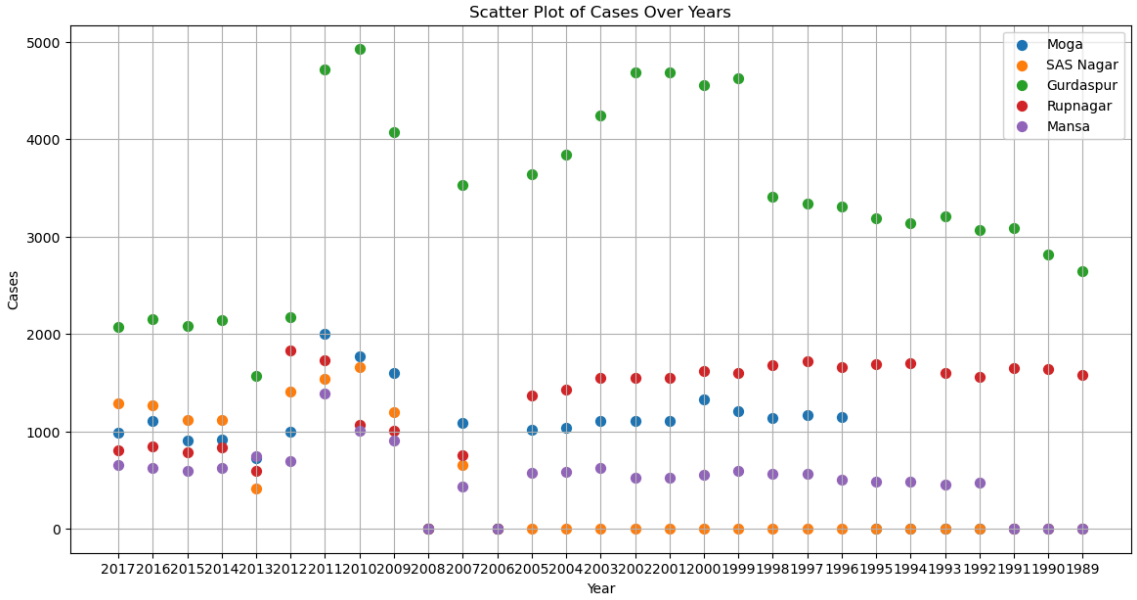
It creates a strip plot to visualize the distribution of case counts over the years, adding jitter for better visibility of overlapping points.

In [23]:
```
plt.figure(figsize=(14, 7))
selected_districts = df["Year/District"].sample(5, random_state=42)  # Rand
for district in selected_districts:
    plt.plot(df.columns[1:], df[df["Year/District"] == district].values.fla
plt.title("Trends of Selected Districts Over the Years")
plt.xlabel("Year")
plt.ylabel("Cases")
plt.legend()
plt.grid(True)
plt.show()
```



It randomly selects 5 districts and plots their case trends over the years using a line plot.

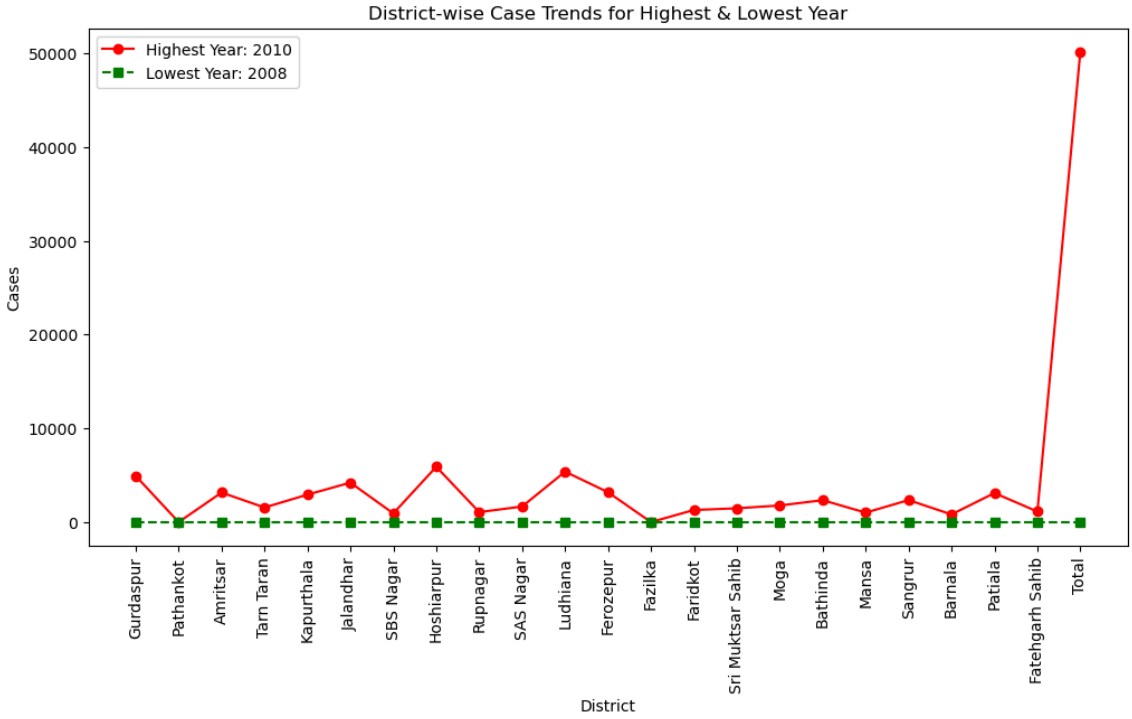
In [24]:
```
plt.figure(figsize=(14, 7))
selected_districts = df["Year/District"].sample(5, random_state=42)
for district in selected_districts:
    plt.scatter(df.columns[1:], df[df["Year/District"] == district].values.
plt.title("Scatter Plot of Cases Over Years")
plt.xlabel("Year")
plt.ylabel("Cases")
plt.legend()
plt.grid(True)
plt.show()
```



It generates a scatter plot of cases over the years for five randomly selected districts, highlighting trends in case distribution.

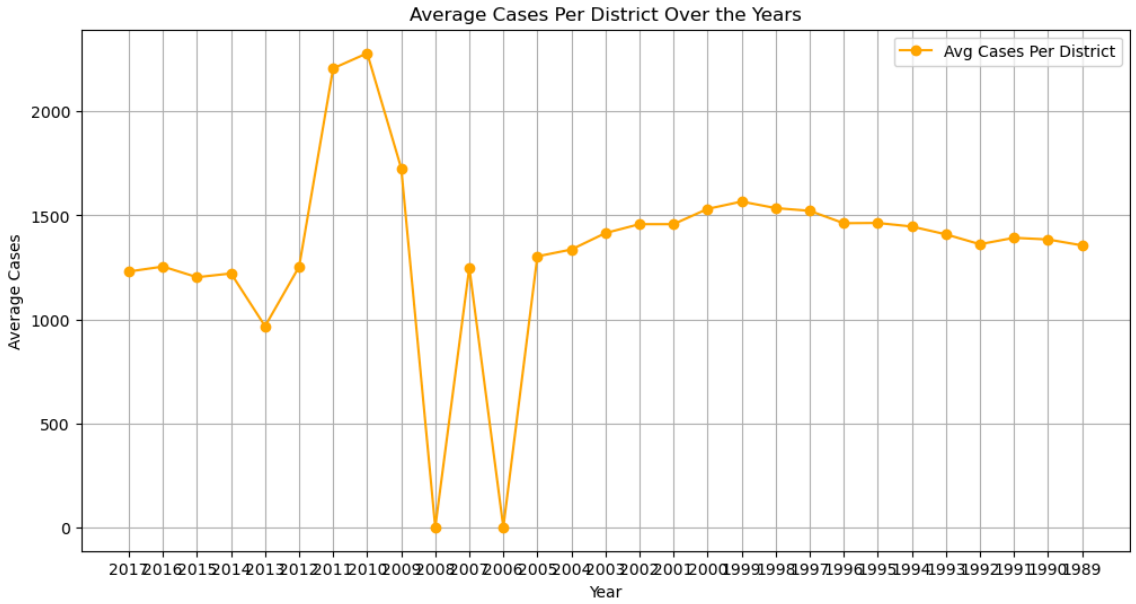
In [25]:
```
highest_year = df.set_index("Year/District").sum().idxmax()
lowest_year = df.set_index("Year/District").sum().idxmin()

plt.figure(figsize=(12, 6))
plt.plot(df["Year/District"], df[highest_year], marker='o', linestyle='-',
plt.plot(df["Year/District"], df[lowest_year], marker='s', linestyle='--',
plt.xticks(rotation=90)
plt.title("District-wise Case Trends for Highest & Lowest Year")
plt.xlabel("District")
plt.ylabel("Cases")
plt.legend()
plt.show()
```



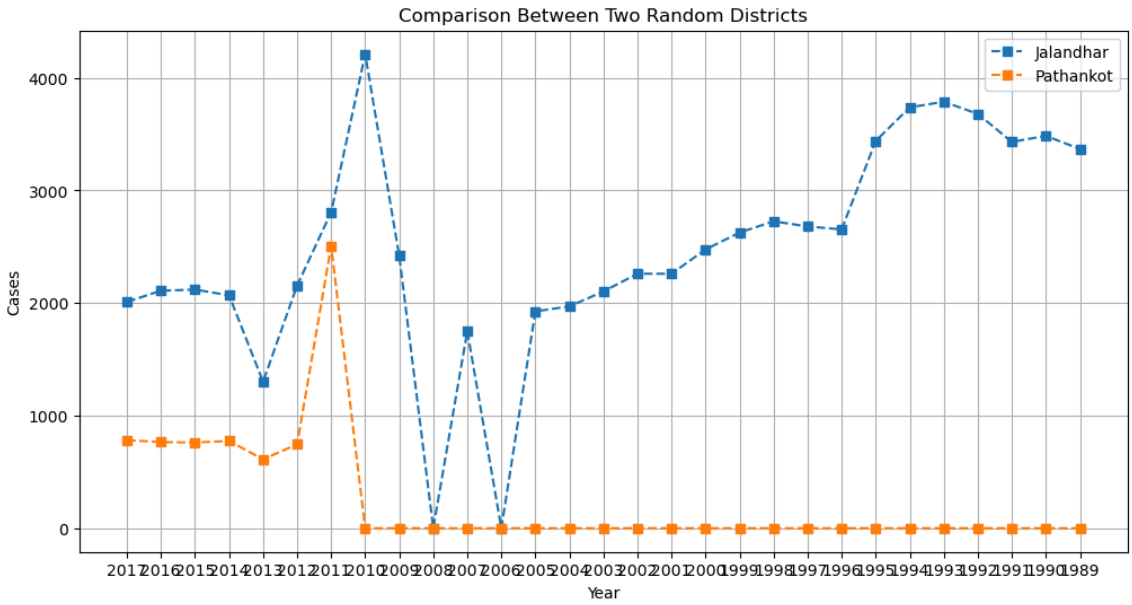
In [26]:
```
plt.figure(figsize=(12, 6))
average_cases = df.set_index("Year/District").drop(index="Total").mean()

plt.plot(average_cases.index, average_cases.values, marker='o', linestyle='
plt.title("Average Cases Per District Over the Years")
plt.xlabel("Year")
plt.ylabel("Average Cases")
plt.legend()
plt.grid(True)
plt.show()
```



It plots the average number of cases per district over the years, excluding the "Total" row, using a line chart with markers for better visualization.

In [27]:
```
plt.figure(figsize=(12, 6))
districts = df["Year/District"].sample(2, random_state=10)
for district in districts:
    plt.plot(df.columns[1:], df[df["Year/District"] == district].values.fla
plt.title("Comparison Between Two Random Districts")
plt.xlabel("Year")
plt.ylabel("Cases")
plt.legend()
plt.grid(True)
plt.show()
```



This randomly selects two districts and plots their case trends over the years using a dashed line with square markers for comparison.

Observations:

The dataset provides insights into the number of women teachers working in middle schools across different districts of Punjab over multiple years. Data cleaning was performed by replacing missing values with 0 and converting all numerical columns into integers for uniformity. Analyzing the trends over time, it was observed that the total number of women teachers fluctuated, with certain years experiencing a peak while others showed a decline. Some districts consistently had a higher number of teachers, whereas others exhibited fluctuations rather than a steady increase or decrease.Visualizations provided deeper insights into these trends. The boxplot highlighted variations in teacher numbers across years,while the line plot showed an overall trend of growth or decline in total teachers. Scatter plots depicted yearly distributionsacross different districts, and heatmaps revealed density variations in teacher numbers over time. The bar chart identified years with the highest and lowest total teachers, while the histogram and KDE plot provided an overview of the distribution,indicating common and rare occurrences. The violin plot and strip plot illustrated the variability and spread of data, whereas pie charts displayed district-wise contributions for the latest and midpoint years.

In [ ]: