Name	Sahil Shah
UID No.	2021300115
Course	Advanced Data Visualization

Experiment 2

Aim	Create advanced charts using Tableau/Power BI/R/Python/Plotly or Chart or D3.js to be performed on the dataset - Socio economic data 1. Advanced - Word chart, Box and whisker plot, Violin plot, Regression plot (linear and nonlinear), 3D chart, Jitter, Line, Area, Waterfall, Donut, Treemap, Funnel 2. Write observations from each chart

1. Importing Libraries and Dataset

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import plotly.graph_objects as go
```

2. Data Preprocessing

In [2]:
```
data = pd.read_csv('../Datasets/india-districts-census-2011.csv')
print(data.shape)
data.head()
```

```
(640, 118)
```

Out[2]:

	District code	State Name	District name	Population	Male	Female	Literate	Male_Literate
0	1	JAMMU AND KASHMIR	Kupwara	870354	474190	396164	439654	282823
1	2	JAMMU AND KASHMIR	Badgam	753745	398041	355704	335649	207741
2	3	JAMMU AND KASHMIR	Leh(Ladakh)	133487	78971	54516	93770	62834
3	4	JAMMU AND KASHMIR	Kargil	140802	77785	63017	86236	56301
4	5	JAMMU AND KASHMIR	Punch	476835	251899	224936	261724	163333

5 rows × 118 columns

In [3]:
```
# keep only first 50 columns and only those rows where state is maharashtra
data = data[data['State Name'] == 'MAHARASHTRA']
data = data.iloc[:, :50]
print(data.shape)
data.head()
```

```
(35, 50)
```

Out[3]:

	District code	State Name	District name	Population	Male	Female	Literate	Male
496	497	MAHARASHTRA	Nandurbar	1648295	833170	815125	906509	
497	498	MAHARASHTRA	Dhule	2050862	1054031	996831	1293916	
498	499	MAHARASHTRA	Jalgaon	4229917	2197365	2032552	2891882	
499	500	MAHARASHTRA	Buldana	2586258	1337560	1248698	1879874	
500	501	MAHARASHTRA	Akola	1813906	932334	881572	1411281	

5 rows × 50 columns

In [4]:
```
# save
data.to_csv('../Datasets/maharashtra-census-2011.csv', index=False)
```

3. Advanced Charts & Plots

In [5]:
```
# load
df = pd.read_csv('../Datasets/maharashtra-census-2011.csv')
df.head()
```

Out[5]:

	District code	State Name	District name	Population	Male	Female	Literate	Male_Li
0	497	MAHARASHTRA	Nandurbar	1648295	833170	815125	906509	5
1	498	MAHARASHTRA	Dhule	2050862	1054031	996831	1293916	7
2	499	MAHARASHTRA	Jalgaon	4229917	2197365	2032552	2891882	16
3	500	MAHARASHTRA	Buldana	2586258	1337560	1248698	1879874	10
4	501	MAHARASHTRA	Akola	1813906	932334	881572	1411281	7

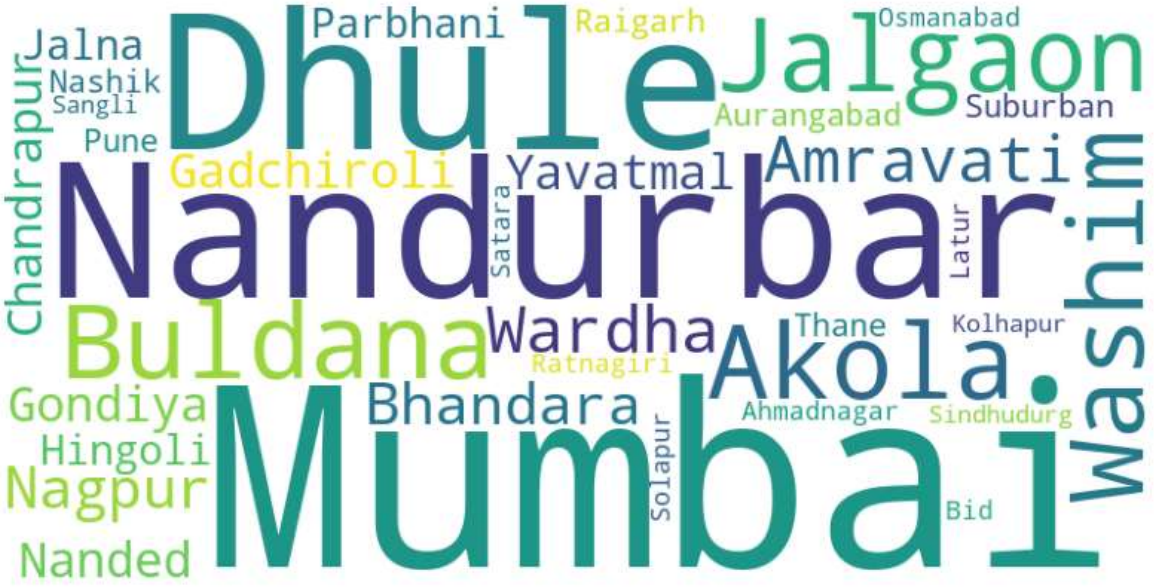
5 rows × 50 columns

In [6]:
```
# print all columns
print(df.columns)
```

```
Index(['District code', 'State Name', 'District name', 'Population', 'Male',
       'Female', 'Literate', 'Male_Literate', 'Female_Literate', 'SC',
       'Male_SC', 'Female_SC', 'ST', 'Male_ST', 'Female_ST', 'Workers',
       'Male_Workers', 'Female_Workers', 'Main_Workers', 'Marginal_Workers',
       'Non_Workers', 'Cultivator_Workers', 'Agricultural_Workers',
       'Household_Workers', 'Other_Workers', 'Hindus', 'Muslims', 'Christians',
       'Sikhs', 'Buddhists', 'Jains', 'Others_Religions',
       'Religion_Not_Stated', 'LPG_or_PNG_Households',
       'Housholds_with_Electric_Lighting', 'Households_with_Internet',
       'Households_with_Computer', 'Rural_Households', 'Urban_Households',
       'Households', 'Below_Primary_Education', 'Primary_Education',
       'Middle_Education', 'Secondary_Education', 'Higher_Education',
       'Graduate_Education', 'Other_Education', 'Literate_Education',
       'Illiterate_Education', 'Total_Education'],
      dtype='object')
```

3.1 Word Chart

In [7]:
```
# create a word cloud chart from data to see which state has the most number of
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(
plt.figure(figsize=(10, 8))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



Observations:

- The word cloud chart shows the frequency of words in the Districts column of the dataset.
- The size of the word represents the frequency of the word in the column. "Nagar" is the most frequent word in the column.
- Directional words like north, south, east and west are also very common in district names.

3.2 Box and Whisker Plot

In [8]:
```
# Create the box plot for population
fig = px.box(df, y="Population", title="Box and Whisker Plot: Population Distrib

max_population = df.loc[df["Population"].idxmax()]
min_population = df.loc[df["Population"].idxmin()]

# Add annotations for max and min districts
fig.add_trace(
    go.Scatter(
        x=[1, 1],  # Position annotations over the boxplot
        y=[max_population["Population"], min_population["Population"]],
        text=[
            f"Max: {max_population['District name']} ({max_population['Populatio
            f"Min: {min_population['District name']} ({min_population['Populatio
        ],
        mode="markers+text",
        textposition="top right",
        marker=dict(color="red", size=10),
    )
)

fig.update_layout(yaxis_title="Population", xaxis_title="Districts", showlegend=
fig.show()
```

Observations:

- The box plot shows the population distribution of the districts.

- Most districts have a population range between 1.5 to 4 million. The median population is around 2.5 million.
- There are a few outliers with a population of more than 4 million like Thane, Pune, Mumbai, and Nashik.

3.3 Violin Plot

In [9]:
```
# Prepare the data for the violin plot
literacy_data = df[["Male_Literate", "Female_Literate"]].melt(
    var_name="Gender", value_name="Literate"
)

# Create the violin plot
fig = px.violin(
    literacy_data,
    y="Literate",
    x="Gender",
    box=True,
    title="Violin Plot: Male and Female Literacy Rates Across Districts in Mahar
    color="Gender",
    color_discrete_map={"Male_Literate": "green", "Female_Literate": "pink"},
)

# Display the plot
fig.update_layout(yaxis_title="Number of Literate Individuals", xaxis_title="Gen
fig.show()
```

Observations:

- The Violin Plot reveals the distribution and density of male and female literacy rates across districts in Maharashtra:
- Male literacy rates generally have a wider distribution, with higher maximum values, indicating that some districts have significantly more literate males than females. The spread is more pronounced, reflecting both higher peaks and lower troughs.
- The density of female literacy rates is higher towards the lower end, showing that in some districts, female literacy is lower compared to male literacy. The distribution is narrower, suggesting fewer districts with exceptionally high female literacy rates.
- Comparison: While there is overlap between male and female literacy distributions, the plot highlights a gender disparity, with male literacy rates tending to be higher in many districts.

3.4 Regression Plot

In [10]:
```
# Prepare the data for the regression plot
df1 = df[["Population", "Literate"]]
df1 = df1.dropna()

# Create the regression plot
fig = px.scatter(
    df1,
    x="Population",
    y="Literate",
    trendline="ols",
```

```
    trendline_color_override="red",
    title="Regression Plot: Population vs Literacy Rate Across Districts in Maha
)

# Display the plot
fig.update_layout(scene_zaxis_type="log")
fig.show()
```

Observation:

- The linear regression plot shows a positive correlation between population and literacy rates, indicating that districts with larger populations tend to have higher literacy rates.
- The spread of the data points around the regression line suggests that while there is a relationship, population alone does not fully explain literacy variations across districts. Other factors may also be influencing literacy rates.

3.5 3D Chart

In [11]:
```
# 3D Chart - Population, Literate, and Workers
# Create a 3D scatter plot
fig = px.scatter_3d(
    df,
    x="Population",
    y="Literate",
    z="Workers",
    color="District name",
    title="3D Scatter Plot: Population, Literacy Rate, and Workforce Participati
    labels={
        "Population": "Total Population",
        "Literate": "Number of Literate Individuals",
        "Workers": "Number of Workers",
    },
)

# Update the layout for better visualization
fig.update_layout(
    scene=dict(
        xaxis_title="Population",
        yaxis_title="Literate Individuals",
        zaxis_title="Workers",
    ),
    margin=dict(l=0, r=0, b=0, t=40),
)

# Show the plot
fig.show()
```

Observation:

- The 3D scatter plot reveals several key insights:
- Districts with higher populations tend to have higher literacy rates, suggesting that larger districts may have better access to educational resources.
- There is also a positive correlation between the number of workers and both population size and literacy rates. Districts with more literate individuals generally

have higher workforce participation, indicating that education plays a crucial role in employment.

- Some districts deviate from the general trend, indicating that other factors, such as economic development, access to jobs, or regional policies, may also play a significant role in shaping these outcomes.

3.6 Jitter Plot

In [12]:
```
# Prepare the data
education_columns = [
    "Below_Primary_Education",
    "Primary_Education",
    "Middle_Education",
    "Secondary_Education",
    "Higher_Education",
    "Graduate_Education",
    "Other_Education",
]

df2 = df[["District name", "Population"] + education_columns]

education_data = df2.melt(
    id_vars=["District name", "Population"],
    value_vars=education_columns,
    var_name="Education_Level",
    value_name="Count",
)

# Add jitter to the population values
jitter_strength = 0.1  # Adjust this value for more or less jitter
education_data["Jittered_Population"] = education_data[
    "Population"
] + np.random.uniform(-jitter_strength, jitter_strength, size=len(education_data

# Create the jitter plot
fig = go.Figure()

for education_level in education_columns:
    subset = education_data[education_data["Education_Level"] == education_level
    fig.add_trace(
        go.Scatter(
            x=subset["Jittered_Population"],
            y=subset["Count"],
            mode="markers",
            name=education_level,
            text=subset["District name"],
            marker=dict(size=8),
        )
    )

# Update the layout
fig.update_layout(
    title="Jitter Plot: Education Levels vs. Population Across Districts in Maha
    xaxis_title="Population",
    yaxis_title="Number of Individuals",
    legend_title="Education Level",
)
```

```
# Show the plot
fig.show()
```

Observation:

- The Jitter Plot provides several key insights into the distribution of education levels across districts in Maharashtra:
- The plot reveals that primary and secondary education levels are prevalent across most districts, irrespective of population size. This indicates that foundational education is widespread, even in less populated districts.
- Higher education levels, such as graduate education, are more concentrated in districts with larger populations. This suggests that access to advanced education is more limited in smaller or less populated districts, which may lack the necessary infrastructure or resources.

3.7 Line Plot

In [13]:
```
# Prepare the data for literacy rates
literacy_data = df[["District name", "Literate"]]

# Sort by literacy rate for better trend visualization
literacy_data_sorted = literacy_data.sort_values(by="Literate")

# Create the line plot
fig = go.Figure()

fig.add_trace(
    go.Scatter(
        x=literacy_data_sorted["District name"],
        y=literacy_data_sorted["Literate"],
        mode="lines+markers",
        name="Literate Individuals",
        line=dict(color="blue"),
        marker=dict(size=8, color="blue"),
    )
)

# Update the layout
fig.update_layout(
    title="Line Plot: Literacy Rates Across Districts in Maharashtra",
    xaxis_title="Districts",
    yaxis_title="Number of Literate Individuals",
    xaxis_tickangle=-45,  # Rotate x-axis labels for better readability
    showlegend=True,
)

# Show the plot
fig.show()
```

Observation:

- The Line Plot provides insights into:

- The plot shows how literacy rates vary across different districts in Maharashtra. It displays the number of literate individuals in each district, connected by lines to reveal trends.
- By examining the line plot, we can see which districts have higher or lower literacy rates. For example, districts with steeper lines indicate significant changes in literacy rates, while flatter sections show more consistency.

3.8 Area Plot

In [14]:
```
# Prepare the data by summing up the different types of workers
workers_data = df[
    [
        "District name",
        "Cultivator_Workers",
        "Agricultural_Workers",
        "Household_Workers",
        "Other_Workers",
    ]
]

# Create the area plot
fig = go.Figure()

# Add traces for each type of worker
for worker_type in [
    "Cultivator_Workers",
    "Agricultural_Workers",
    "Household_Workers",
    "Other_Workers",
]:
    fig.add_trace(
        go.Scatter(
            x=workers_data["District name"],
            y=workers_data[worker_type],
            mode="lines",
            stackgroup="one",
            name=worker_type,
        )
    )

# Update the layout
fig.update_layout(
    title="Area Plot: Distribution of Different Types of Workers Across District
    xaxis_title="Districts",
    yaxis_title="Number of Workers",
    legend_title="Worker Type",
    xaxis_tickangle=-45,  # Rotate x-axis labels for better readability
)

# Show the plot
fig.show()
```

Observation:

- The Area Plot provides several insights into the distribution of different types of workers across Maharashtra's districts:

- The plot shows the number of individuals in various worker categories, such as cultivator, agricultural, household, and other workers. Each type of worker is represented by a different color, with the area under the curve indicating the proportion of that worker type across districts.
- The stacked areas allow for an easy comparison of the relative sizes of each worker category. For example, if one category consistently occupies a larger area, it suggests that this type of work is more prevalent in those districts.
- By examining the plot, we can identify trends or disparities in workforce distribution. For instance, if a particular worker type dominates in certain districts, it may indicate a specialized economic activity or local industry. Conversely, if the areas are more balanced, it may reflect a diverse workforce.

3.9 Waterfall Chart

In [15]:
```
# Calculate the total literacy from different education levels
education_totals = {
    "Below_Primary_Education": df["Below_Primary_Education"].sum(),
    "Primary_Education": df["Primary_Education"].sum(),
    "Middle_Education": df["Middle_Education"].sum(),
    "Secondary_Education": df["Secondary_Education"].sum(),
    "Higher_Education": df["Higher_Education"].sum(),
    "Graduate_Education": df["Graduate_Education"].sum(),
    "Other_Education": df["Other_Education"].sum(),
}

# Prepare the data for the waterfall chart
education_data = list(education_totals.items())
education_data.insert(0, ("Start", 0))  # Starting point
education_data.append(("Total Literacy", df["Literate"].sum()))  # Ending point

# Create the waterfall chart
fig = go.Figure()

# Add the waterfall chart bars
fig.add_trace(
    go.Waterfall(
        measure=["relative"] * (len(education_data) - 2) + ["total"],
        x=[item[0] for item in education_data],
        y=[item[1] for item in education_data],
        text=[f"{item[1]:,}" for item in education_data],
        textposition="outside",
        connector=dict(line=dict(color="rgba(63, 63, 63, 0.8)", width=2)),
        name="Literacy Contribution",
    )
)

# Update the layout
fig.update_layout(
    title="Waterfall Chart: Contribution of Different Education Levels to Total
    xaxis_title="Education Level",
    yaxis_title="Number of Literate Individuals",
    xaxis_tickangle=-45,  # Rotate x-axis labels for better readability
    showlegend=False,
)
```

```
# Show the plot
fig.show()
```

Observation:

- Each bar in the chart represents how different education levels contribute to the total literacy rate. It starts with zero and cumulatively adds the number of literate individuals from each education category.
- Primary and secondary education levels contribute the most to the total literacy rate, highlighting their importance in the overall educational landscape.
- Higher and graduate education levels add less to the total literacy rate, which may reflect the relative scarcity of individuals with these levels of education compared to foundational education.
- The final bar shows the total number of literate individuals, providing a clear picture of how different educational achievements accumulate to contribute to overall literacy.

3.10 Donut Chart

In [16]:
```
# Prepare the data for religious groups
religious_groups = {
    "Hindus": df["Hindus"].sum(),
    "Muslims": df["Muslims"].sum(),
    "Christians": df["Christians"].sum(),
    "Sikhs": df["Sikhs"].sum(),
    "Buddhists": df["Buddhists"].sum(),
    "Jains": df["Jains"].sum(),
    "Others": df["Others_Religions"].sum(),
    "Religion Not Stated": df["Religion_Not_Stated"].sum(),
}

# Prepare data for the donut chart
religion_names = list(religious_groups.keys())
religion_values = list(religious_groups.values())

# Create the donut chart
fig = go.Figure()

fig.add_trace(
    go.Pie(
        labels=religion_names,
        values=religion_values,
        hole=0.4,  # Hole size for the donut effect
        textinfo="label+percent",
        insidetextorientation="radial",
        marker=dict(colors=px.colors.qualitative.Plotly),
    )
)

# Update the layout
fig.update_layout(
    title="Religious Groups in Maharashtra", showlegend=True
)
```

```
# Show the plot
fig.show()
```

Observation:

- Each segment of the donut chart represents a different religious group, with the size of the segment showing the proportion of the group within the total population of Maharashtra's districts.
- The chart highlights the predominant religious groups, such as Hindus and Muslims, which have larger segments, indicating they make up significant portions of the population.
- Smaller segments represent minority religions, showing their relative size compared to the major groups.

3.11 Treemap

In [17]:
```
# Prepare the data for the treemap
# Flatten the dataset to include total population as a value and district names
treemap_data = df[["District name", "Population", "Male", "Female"]]

# Create the treemap
fig = px.treemap(
    treemap_data,
    path=["District name"],
    values="Population",
    color="Population",
    color_continuous_scale="Viridis",
    title="Treemap: Population Distribution Across Districts in Maharashtra",
    labels={"Population": "Total Population"},
    hover_data={
        "District name": True,
        "Population": True,
        "Male": True,
        "Female": True,
    },
)

# Update the layout
fig.update_layout(margin=dict(t=40, l=0, r=0, b=0))

# Show the plot
fig.show()
```

Observation:

- Each block represents a district, with the size of the block proportional to the district's population. Larger blocks indicate districts with larger populations, while smaller blocks represent districts with smaller populations.
- The color gradient helps to visually differentiate between districts with varying population sizes. Darker or lighter shades represent different population magnitudes.
- Although the current data is flat (with only districts as categories), the treemap can be extended to include more hierarchical levels if needed (e.g., further breakdown by

male and female populations).

3.12 Funnel Chart

In [18]:
```
# Prepare the data for the funnel plot
# Define the categories and their values
worker_categories = [
    "Total Workforce",
    "Main Workers",
    "Marginal Workers",
    "Non-Workers",
]
worker_values = [
    df["Workers"].sum(),
    df["Main_Workers"].sum(),
    df["Marginal_Workers"].sum(),
    df["Non_Workers"].sum(),
]

# Create the funnel plot
fig = go.Figure()

# Add the funnel plot bars
fig.add_trace(
    go.Funnel(
        y=worker_categories,
        x=worker_values,
        text=worker_values,
        textinfo="value+percent initial",
        marker=dict(color="orange"),
    )
)

# Update the layout
fig.update_layout(
    title="Funnel Plot: Contribution of Different Worker Types to Total Workforc
    xaxis_title="Number of Workers",
    yaxis_title="Worker Type",
    showlegend=False,
)

# Show the plot
fig.show()
```

Observation:

- The plot shows how various worker types contribute to the total workforce, beginning with the broad category of total workforce and narrowing down to specific types of workers like main workers, marginal workers, and non-workers.
- The width of each section of the funnel indicates the number of workers in each category. Wider sections represent larger numbers, while narrower sections show smaller counts.
- By examining the funnel plot, we can see how different categories of workers are distributed relative to the total workforce. This helps in understanding which types of workers are more prevalent and which are less represented.

Conclusion

In this experiment, we learned how to create advanced charts and plotly in python. We explored various visualization techniques, such as word clouds, box plots, violin plots, regression plots, 3D charts, jitter plots, line plots, area plots, waterfall charts, donut charts, treemaps, and funnel charts. Each visualization provided unique insights into the socio-economic data of Maharashtra's districts, revealing patterns, trends, and relationships that can inform decision-making and policy development.