DATA VISUALIZATION
(PMC-434)

PROJECT-1
MSC Mathematics and Computing, IV Sem

SUBMITTED BY: MUDITA SHARMA (302303008)
SUBMITTED TO: DR. KAVITA

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, PATIALA

INDEX

Objective

Have you ever wondered how much money political parties spend on social media ads to influence voters during elections?

If you are highly active on Facebook or Instagram you may have noticed election related ads from various political parties, like BJP and INC. Political parties in India invest heavily in election campaigns, through digital advertisements. I analyzed data from Meta Ads to explore the spending patterns of political parties on Facebook and Instagram during the Indian elections of 2024, broken down by state. In this presentation, we will examine the analysis of election ad spending using Python and its impact on voting patterns.

Dataset Overview

The dataset I have collected contains three files:

1. The Advertisers Dataset provide details party pages, amount of money they spend on election ads and the volume of ads they run.
2. The Locations Dataset shows how much money was spent on ads in different locations.
3. The Results Dataset provides actual voting data showing how many people voted in each area and the percentage of voter turnout.

1.Advertisers Dataset

- Page ID: A unique identifier for the advertiser's page.
- Page name: The name of the advertiser's page.
- Disclaimer: Information about the advertiser, typically who paid for the ads.
- Amount spent (INR): The total amount of money spent on ads in Indian Rupees.
- Number of ads in Library: The number of ads associated with the advertiser.

```
advertisers.shape
```

```
(20832, 5)
```

```
advertisers.size
```

```
104160
```

```
advertisers.dtypes
```

	0
Page ID	int64
Page name	object
Disclaimer	object
Amount spent (INR)	object
Number of ads in Library	int64

```
advertisers.head()
```

	Page ID	Page name	Disclaimer	Amount spent (INR)	Number of ads in Library
0	121439954563203	Bharatiya Janata Party (BJP)	Bharatiya Janata Party (BJP)	193854342	43455
1	351616078284404	Indian National Congress	Indian National Congress	108787100	846
2	132715103269897	Ama Chinha Sankha Chinha	Ama Chinha Sankha Chinha	73361399	1799
3	192856493908290	Ama Chinha Sankha Chinha	Ama Chinha Sankha Chinha	32294327	680
4	109470364774303	Ellorum Nammudan	Populus Empowerment Network Private Limited	22399499	879

2. Locations Dataset

- **Location name**: The name of the location.
- **Amount spent (INR)**: The total amount of money spent on ads in that location in Indian Rupees.

```
locations.shape
```

```
(36, 2)
```

```
locations.size
```

```
72
```

```
locations.dtypes
```

	0
Location name	object
Amount spent (INR)	int64

```
locations.head()
```

	Location name	Amount spent (INR)
0	Andaman and Nicobar Islands	377858
1	Andhra Pradesh	100819732
2	Arunachal Pradesh	1385654
3	Assam	17478091
4	Bihar	53619242

4. Results Dataset

- _id: A unique identifier for the entry.
- Sl No: Serial number.
- State: The name of the state.
- PC_Name: The name of the parliamentary constituency.
- Total Electors: The total number of registered voters.
- Polled (%): The percentage of votes polled.
- Total Votes: The total number of votes cast.
- Phase: The phase of the election.

```
results.shape
```

```
(550, 8)
```

```
results.size
```

```
4400
```

```
results.dtypes
```

	0
_id	int64
Sl No	float64
State	object
PC_Name	object
Total Electors	int64
Polled (%)	float64
Total Votes	int64
Phase	float64

```
results.head()
```

	_id	Sl No	State	PC_Name	Total Electors	Polled (%)	Total Votes	Phase
0	1	1.0	Andaman and Nicobar Islands	Andaman and Nicobar Islands	315148	64.10	202018	1.0
1	2	2.0	Arunachal Pradesh	Arunachal East	375310	83.31	312658	1.0
2	3	3.0	Arunachal Pradesh	Arunachal West	517384	73.60	380783	1.0
3	4	4.0	Assam	Dibrugarh	1659588	76.75	1273744	1.0
4	5	5.0	Assam	Jorhat	1727121	79.89	1379749	1.0

The results dataset contains a "state" column, while the location dataset contains a "location name" column. We will merge these two datasets using a left join in such a way that all rows from the results dataset are retained. If no match is found in the "location name" column, a NaN value will be added in such cases.

```
merged_data.head()
```

	_id	Sl No	State	PC_Name	Total Electors	Polled (%)	Total Votes	Phase	Location name	Amount spent (INR)
0	1	1.0	andaman and nicobar islands	Andaman and Nicobar Islands	315148	64.10	202018	1.0	andaman and nicobar islands	377858.0
1	2	2.0	arunachal pradesh	Arunachal East	375310	83.31	312658	1.0	arunachal pradesh	1385654.0
2	3	3.0	arunachal pradesh	Arunachal West	517384	73.60	380783	1.0	arunachal pradesh	1385654.0
3	4	4.0	assam	Dibrugarh	1659588	76.75	1273744	1.0	assam	17478091.0
4	5	5.0	assam	Jorhat	1727121	79.89	1379749	1.0	assam	17478091.0

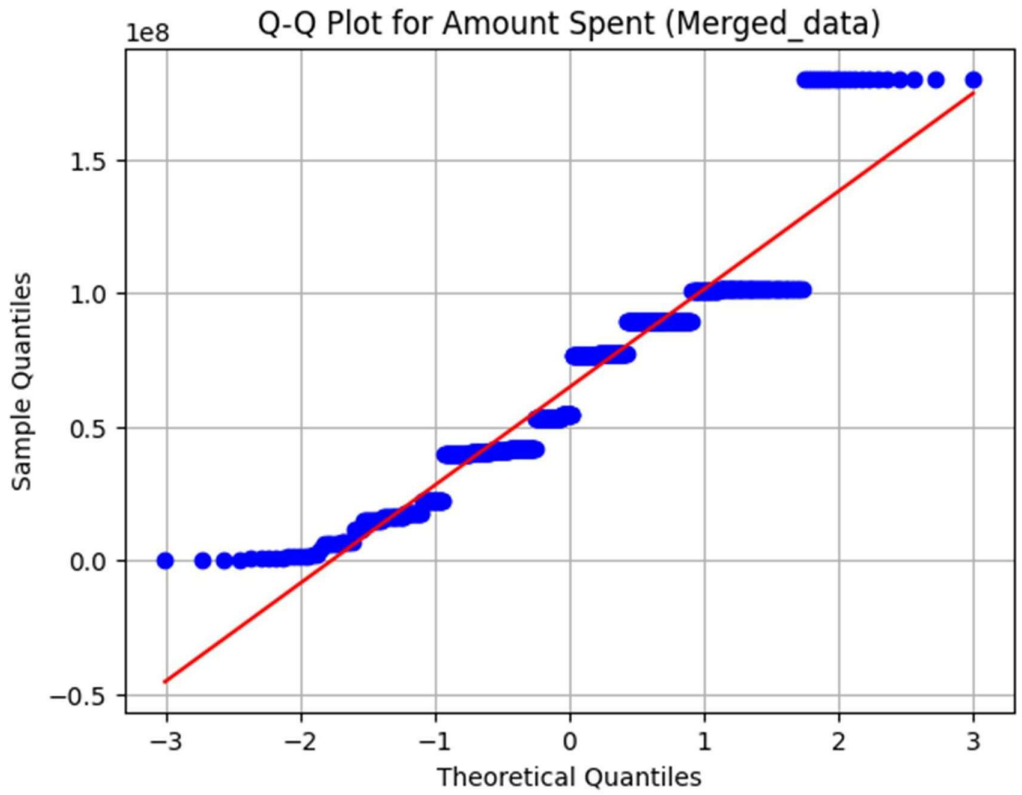
```
nan_rows = merged_data[merged_data['Location name'].isna() & merged_data['State'].notna()]
print(nan_rows[['State', 'Location name']])
```

```
                                  State Location name
208  dadra and nagar haveli and\ndaman and diu           NaN
209  dadra and nagar haveli and\ndaman and diu           NaN
392                                ladakh           NaN
456                          nct of delhi           NaN
457                          nct of delhi           NaN
458                          nct of delhi           NaN
459                          nct of delhi           NaN
460                          nct of delhi           NaN
461                          nct of delhi           NaN
462                          nct of delhi           NaN
514                                punjab           NaN
515                                punjab           NaN
516                                punjab           NaN
517                                punjab           NaN
```

Graphic Display of Basic Statistical Descriptions of data

1. Q-Q Plot of Amount Spent in merged_data Dataset

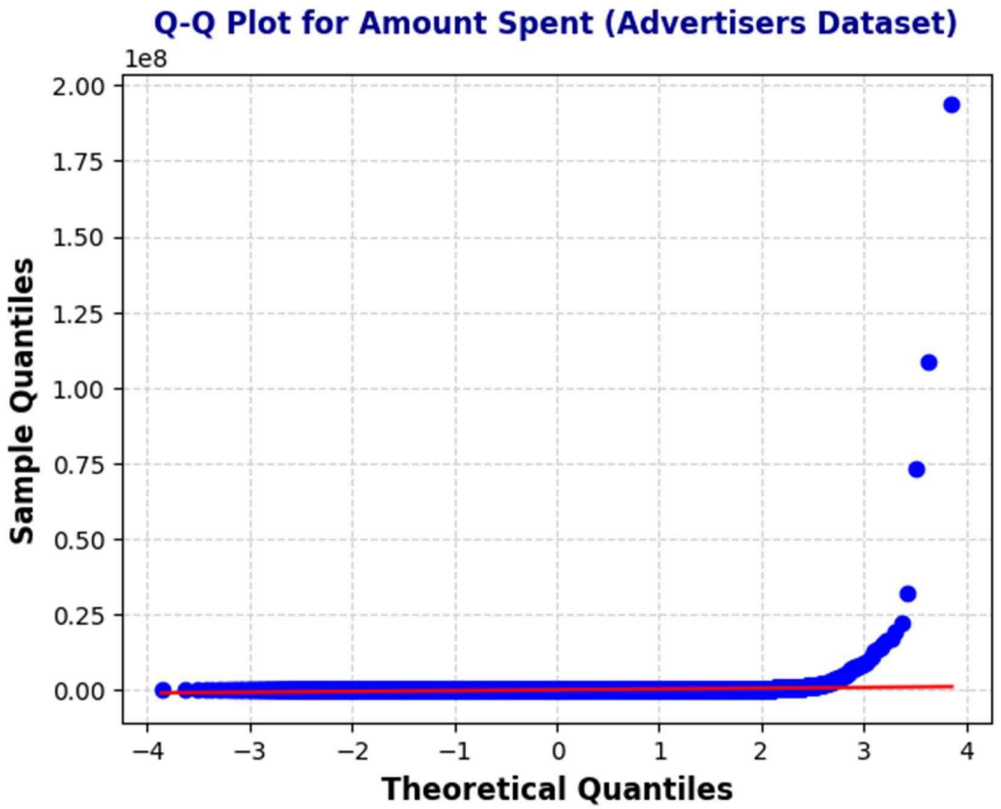
```
merged_data.dropna(subset=['Amount spent (INR)'],inplace=True)
stats.probplot(merged_data['Amount spent (INR)'],dist="norm",plot=plt)
plt.title('Q-Q Plot for Amount Spent (Merged_data)')
plt.xlabel('Theoretical Quantiles')
plt.ylabel('Sample Quantiles')
plt.grid(True)
plt.show()
```



- Compares the Amount Spent from the Merged_data dataset to a theoretical normal distribution. x-axis represents the theoretical quantiles of a normal distribution. y-axis represents the quantiles from the dataset
- The red diagonal line represents the expected distribution under normality.The blue dots represent the actual data distribution.
- The points are deviating from the red line at the upper end (right side), indicating that the data is not normally distributed and is a right-skewed distribution.
- The points in the middle range follow the red line indicating that the central values are normally distributed.
- So,there are few individuals have spent significantly more than the majority leading to a non-normal distribution.

2. Q-Q Plot of Amount Spent in advertisers Dataset

```
advertisers['Amount spent (INR)'] = pd.to_numeric(advertisers['Amount spent (INR)']
                                    ,errors='coerce')
advertisers.dropna(subset=['Amount spent (INR)'], inplace=True)
stats.probplot(advertisers['Amount spent (INR)'], dist="norm", plot=plt)
plt.title('Q-Q Plot for Amount Spent (Advertisers Dataset)', fontweight='bold',
          color='darkblue')
plt.xlabel('Theoretical Quantiles', fontsize=12, fontweight='bold')
plt.ylabel('Sample Quantiles', fontsize=12, fontweight='bold')
plt.grid(color='lightgrey', linestyle='--')
plt.show()
```

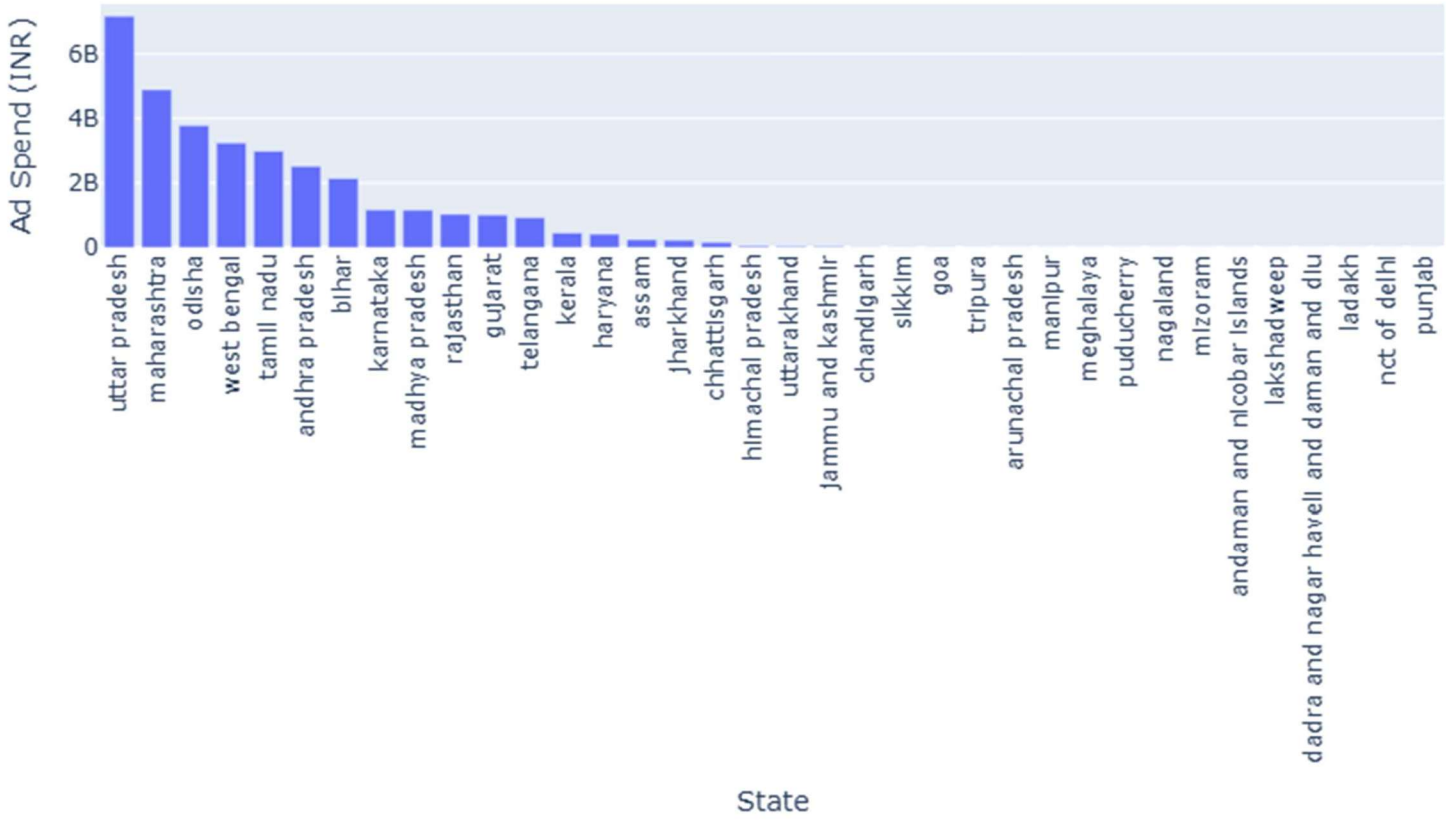


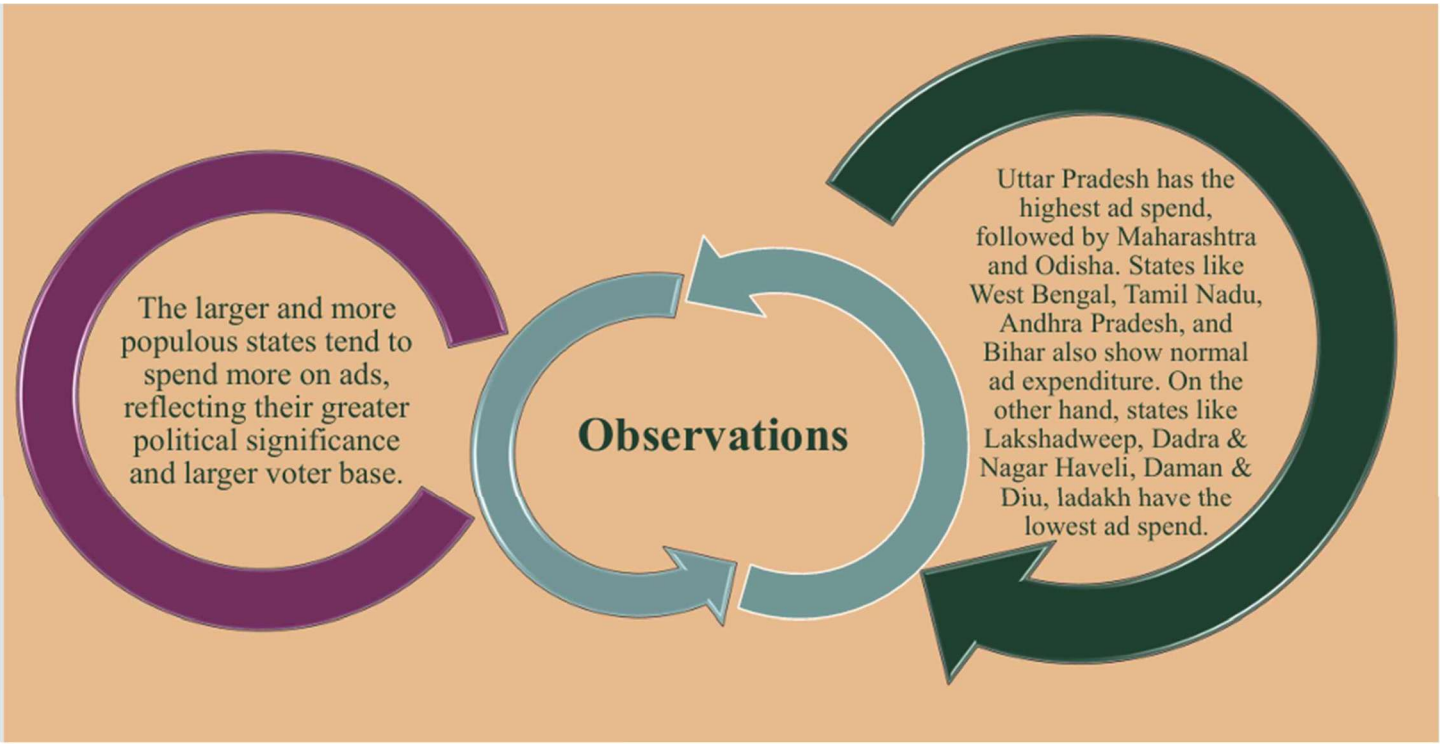
- Compares the Amount Spent from the Advertisers dataset to a theoretical normal distribution. x-axis represents the theoretical quantiles of a normal distribution. y-axis represents the quantiles from the dataset
- The red diagonal line represents the expected distribution under normality.The blue dots represent the actual data distribution.
- The majority of points lie almost flat along x-axis which means most data points are close to zero.
- .A few points on the right (upper) deviate from the red line indicating right-skewed distribution, i.e. distribution isn't normal.

Data Visualization

1. Total ad spend by state

```
import plotly.express as px
state_ad_spend = merged_data.groupby('State')['Amount spent (INR)'].sum().reset_index()
fig = px.bar(state_ad_spend, x='State', y='Amount spent (INR)',labels={'State': 'State', 'Amount spent (INR)': 'Ad Spend (INR)'},
      title='Total Ad Spend by State')
fig.update_layout(xaxis={'categoryorder': 'total descending'},xaxis_tickangle=-90)
fig.show()
```





2.Voter turnout by state

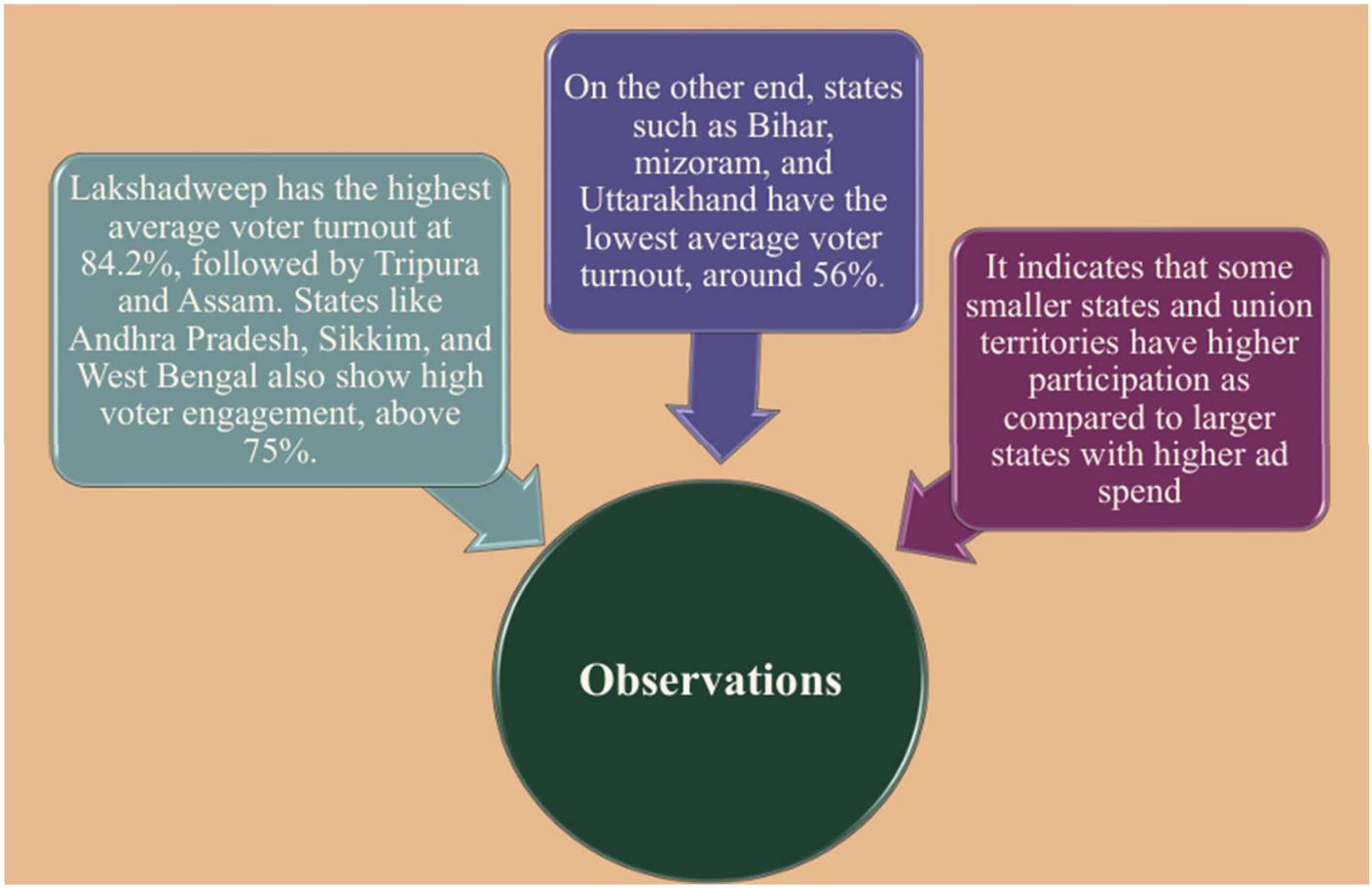
```python
from matplotlib.patches import Circle, PathPatch
from matplotlib.path import Path
import math

def get_face_parameters(turnout):    # Func for face parameters based on turnout
    if turnout >= 80:
        return 0.15  # Big smile (U-shaped, downward)
    elif turnout >= 60:
        return 0.08  # Moderate smile (downward)
    elif turnout >= 40:
        return 0.0   # Neutral (straight line)
    else:
        return -0.10  # Sad (frown, upward)

def draw_chernoff_face(ax, turnout, x=0, y=0):    # Func to draw Chernoff faces
    smile_control = get_face_parameters(turnout)

    face = Circle((x, y), 0.4, fill=True, color='yellow', linewidth=1.5)
    ax.add_patch(face)

    eye_size = 0.05
    left_eye = Circle((x-0.20, y+0.10), eye_size, color='blue')
    right_eye = Circle((x+0.20, y+0.10), eye_size, color='blue')
    ax.add_patch(left_eye)
    ax.add_patch(right_eye)
```

```
    verts = [(x-0.2, y-0.10),                  # Left point
        (x, y-0.10 - smile_control),  # Control point for downward smile
        (x+0.2, y-0.10)]                # Right point
    codes = [Path.MOVETO, Path.CURVE3, Path.CURVE3]
    path = Path(verts, codes)
    mouth = PathPatch(path, fill=False, color='red', linewidth=1.5)
    ax.add_patch(mouth)

df = merged_data.groupby('State')['Polled (%)'].mean().reset_index()
df = df.sort_values('Polled (%)', ascending=False)

n_states = len(df)
n_cols = min(5, n_states)  # Max 5 columns
n_rows = math.ceil(n_states / n_cols)

fig, axs = plt.subplots(n_rows, n_cols, figsize=(n_cols * 2, n_rows * 2))
fig.suptitle('Voter Turnout by State (Chernoff Faces)\nSorted by Turnout Percentage', fontsize=14)

# legend showing the turnout ranges
legend_text = (
    'Turnout Ranges:\n'
    '≥ 80%: Very Happy\n'
    '≥ 60%: Happy\n'
    '≥ 40%: Neutral\n'
    '< 40%: Sad'
)
plt.figtext(0.02, 0.02, legend_text, fontsize=8, ha='left')
```

```
if n_rows == 1:
    axs = [axs]
if n_cols == 1:
    axs = [[ax] for ax in axs]

# Draw faces for each state
for idx, (state, turnout) in enumerate(zip(df['State'], df['Polled (%)'])):
    row = idx // n_cols
    col = idx % n_cols
    ax = axs[row][col]
    draw_chernoff_face(ax, turnout)
    ax.set_title(f'{state}\n{turnout:.1f}%', fontsize=10, pad=10)
    ax.set_xlim(-0.8, 0.8)
    ax.set_ylim(-0.8, 0.8)
    ax.axis('equal')
    ax.axis('off')

# Hide empty subplots
for idx in range(n_states, n_rows * n_cols):
    row = idx // n_cols
    col = idx % n_cols
    axs[row][col].set_visible(False)

plt.tight_layout()
# Adjust layout to make room for legend
plt.subplots_adjust(bottom=0.15)
plt.show()
```

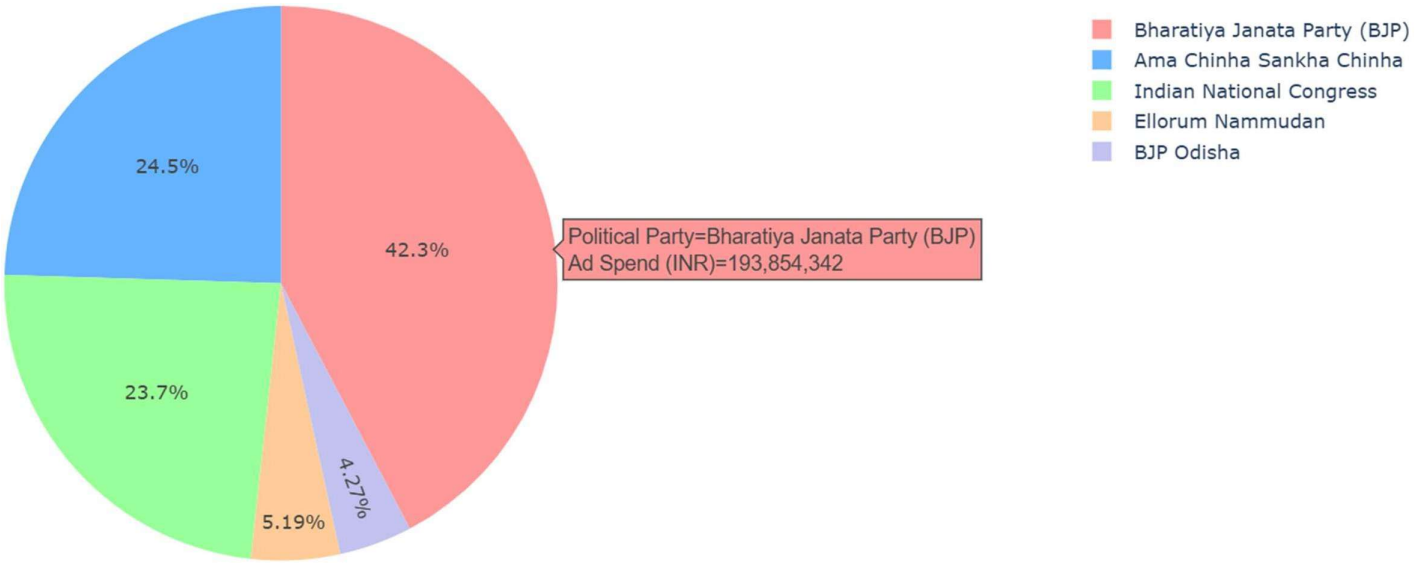
Voter Turnout by State (Chernoff Faces)
Sorted by Turnout Percentage

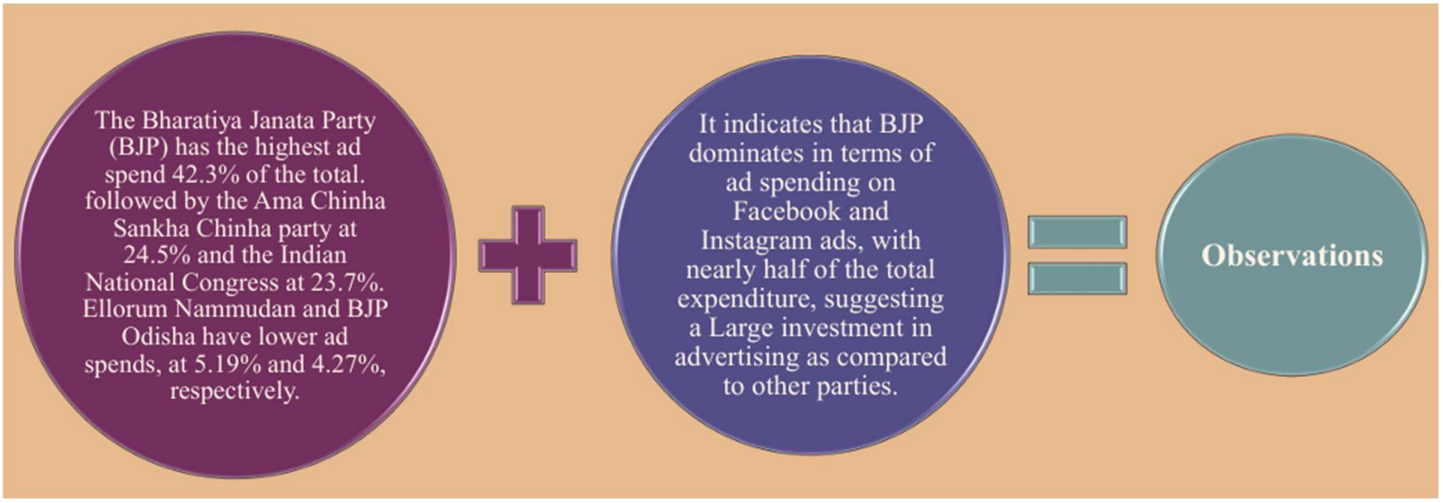
Turnout Ranges:
≥ 80%: Very Happy
≥ 60%: Happy
≥ 40%: Neutral
< 40%: Sad



3. Top 5 parties by ad spend

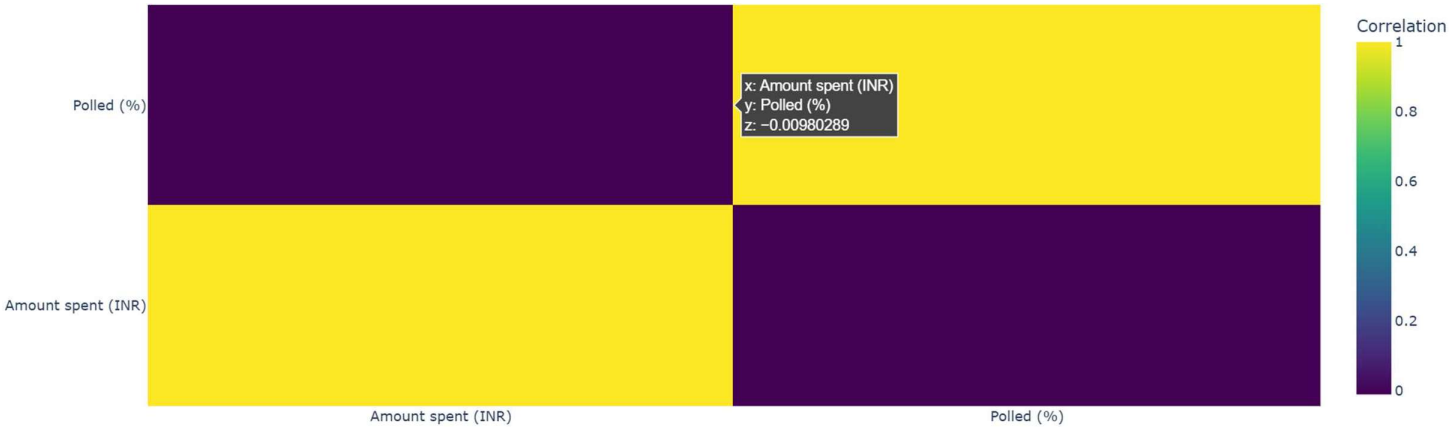
```
party_ad_spend = advertisers.groupby('Page name')['Amount spent (INR)'].sum().sort_values(ascending=False)
top_5_parties = party_ad_spend.head(5).reset_index()
colors = ['#ff9999', '#66b3ff', '#99ff99', '#ffcc99', '#c2c2f0']
fig = px.pie(top_5_parties, values='Amount spent (INR)', names='Page name',title='Top 5 Parties by Ad Spend', color_discrete_sequence=colors,
             labels={'Page name': 'Political Party', 'Amount spent (INR)': 'Ad Spend (INR)'})
fig.show()
```

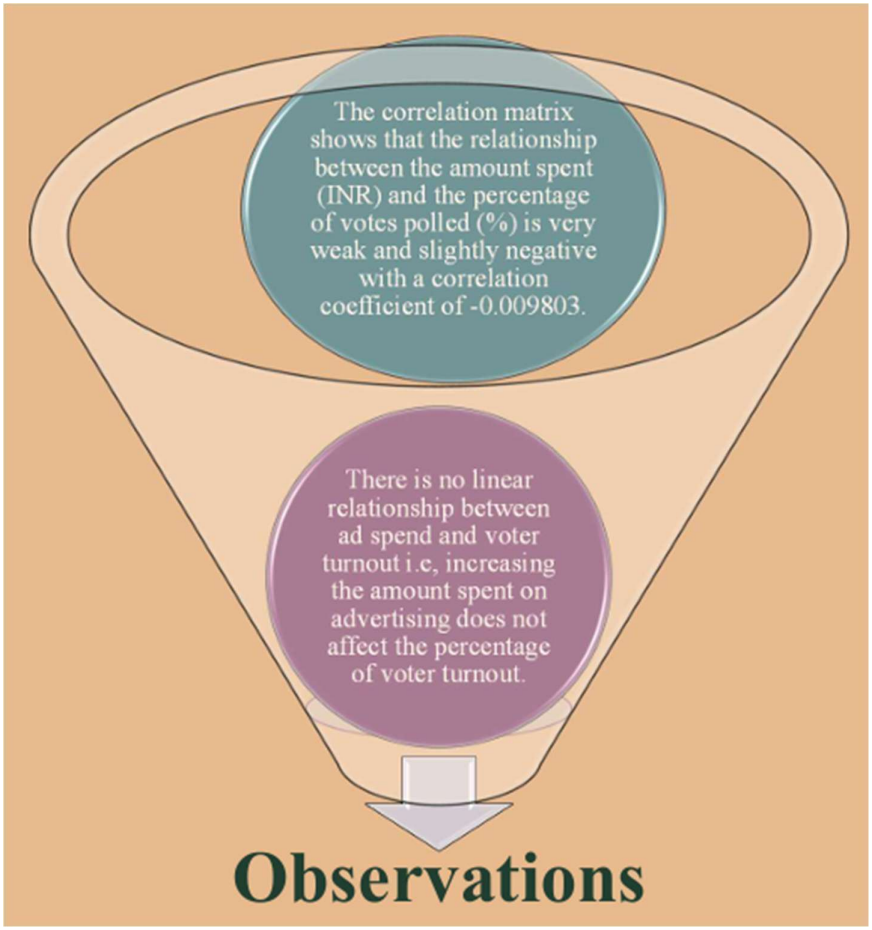




4. Correlation between ad spend and voter turnout

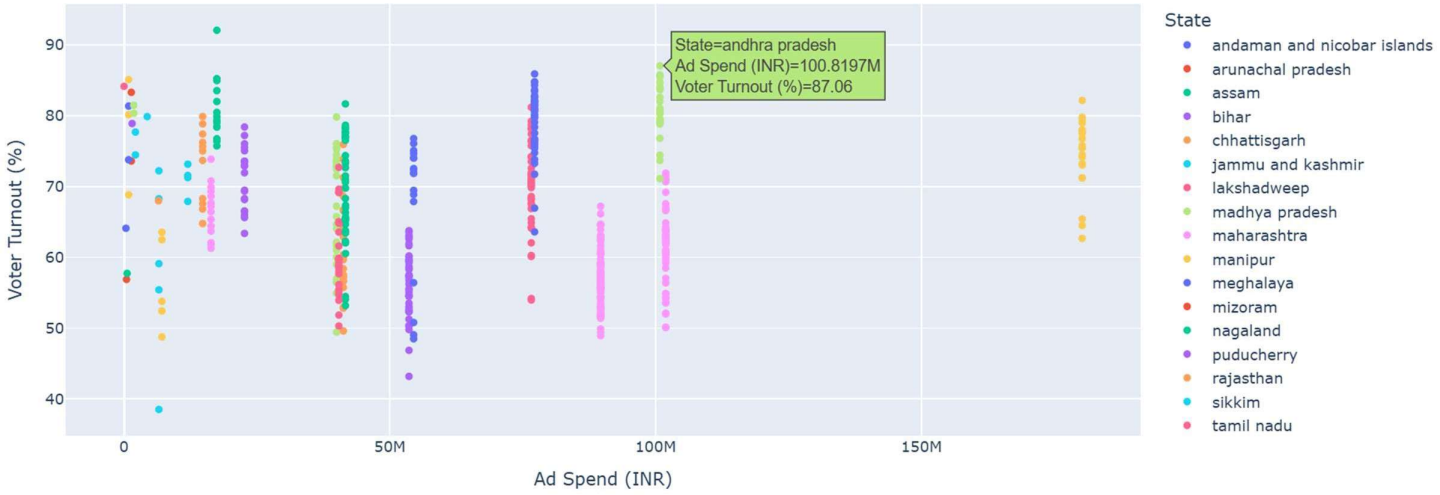
```
import plotly.graph_objects as go
correlation = merged_data[['Amount spent (INR)', 'Polled (%)']].corr()
fig = go.Figure(data=go.Heatmap(z=correlation.values, x=correlation.columns,y=correlation.columns,
      colorscale='Viridis',colorbar=dict(title='Correlation'),showscale=True))
fig.update_layout(title='Correlation Heatmap: Ad Spend vs Voter Turnout')
fig.show()
```





5. Relationship between ad spend and voter turnout by parliamentary constituency

```
merged_constituency_data = results.merge(locations,left_on='State',right_on='Location name',how='left')
fig = px.scatter(merged_constituency_data, x='Amount spent (INR)', y='Polled (%)',color='State',
                 labels={'Amount spent (INR)': 'Ad Spend (INR)', 'Polled (%)': 'Voter Turnout (%)'},
                 title='Ad Spend and Voter Turnout by Parliamentary Constituency')
fig.show()
```

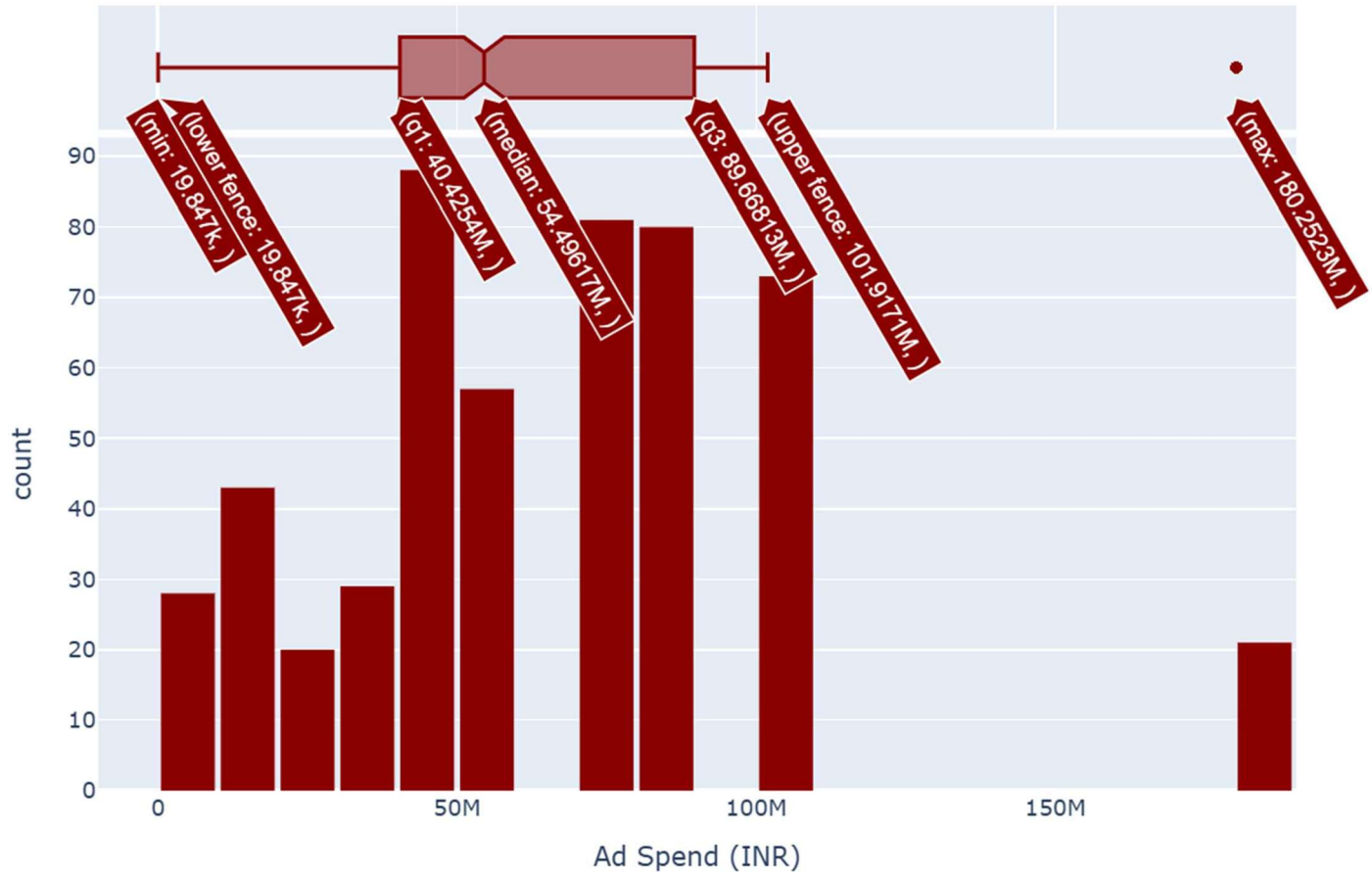


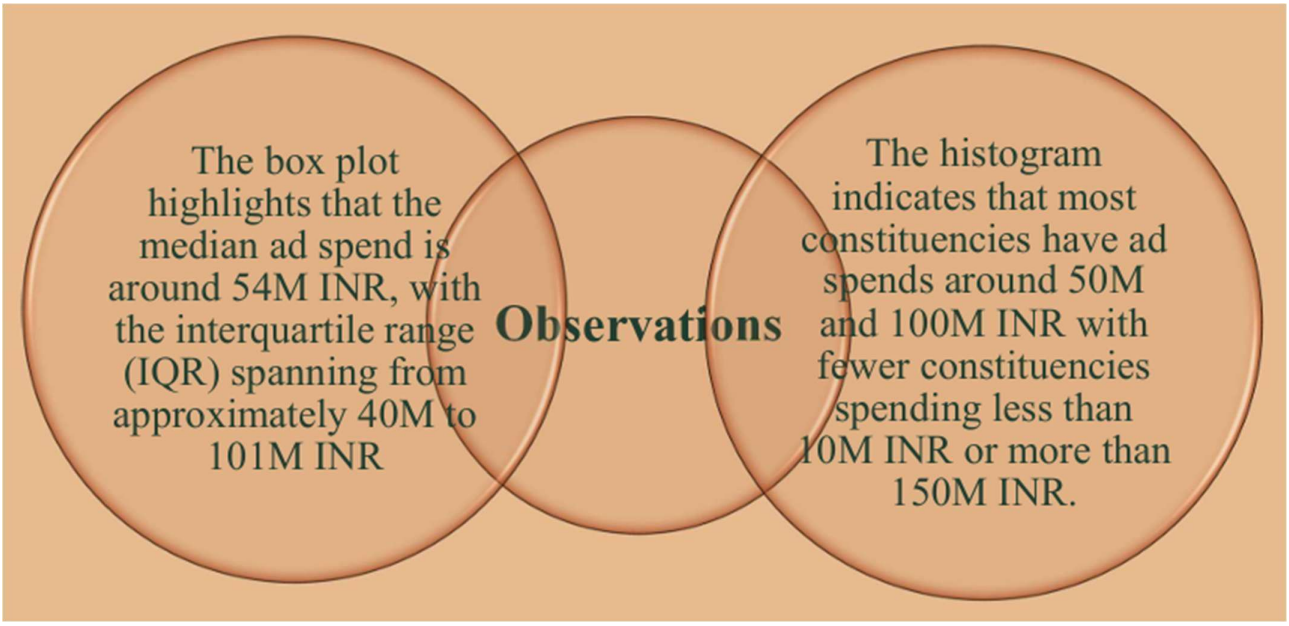
Observations

- It shows that higher ad spending does not correlate with higher voter turnout.
- Voter turnout seems to grouped between 60% and 80% across most constituencies regardless of the ad spend amount which ranges from 0 to 150 million INR.
- Other factors besides ad spend also play a significant role in influencing voter turnout.

6.Distribution of ad spending

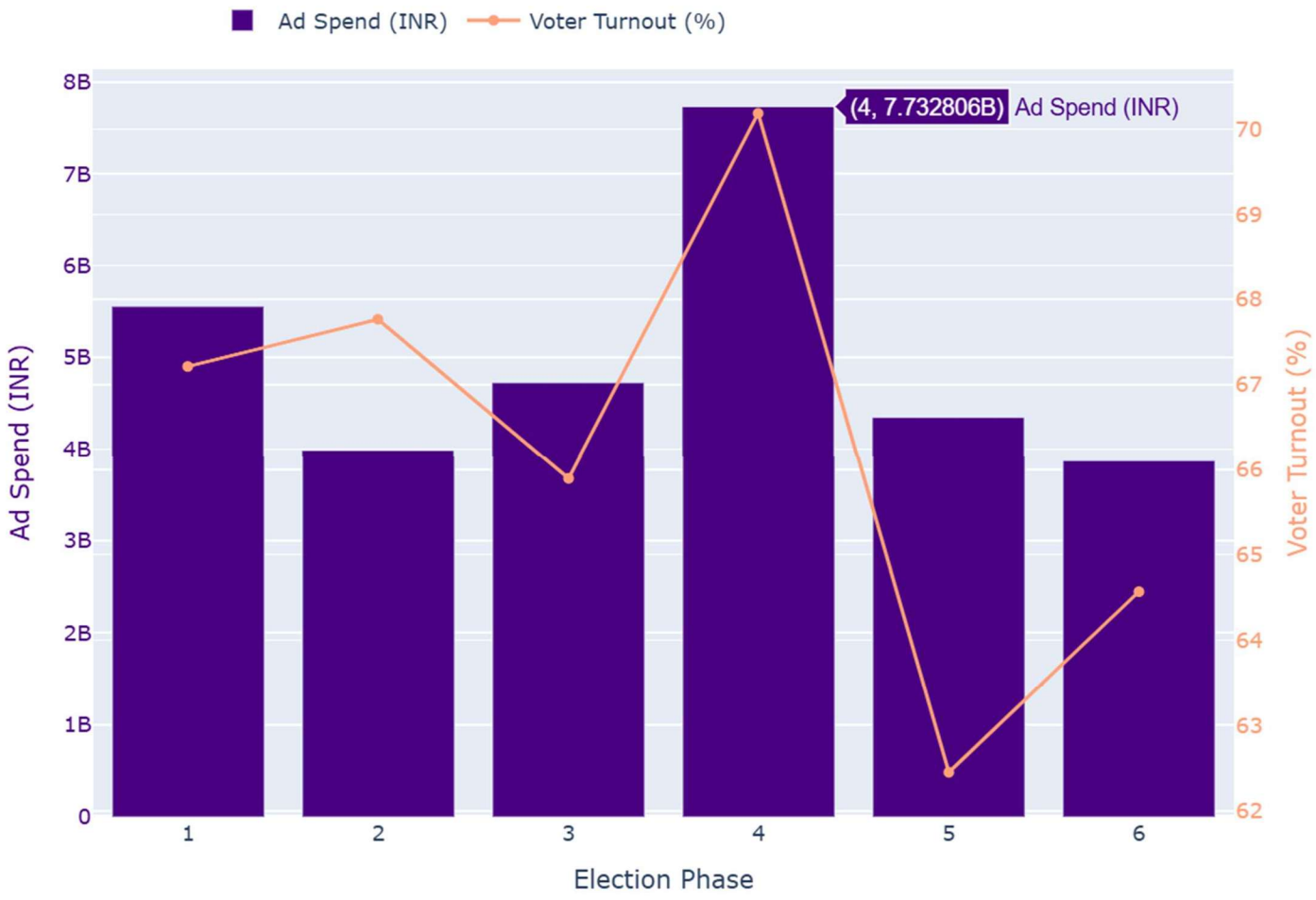
```
fig = px.histogram(merged_data, x='Amount spent (INR)', nbins=30, marginal='box',labels={'Amount spent (INR)': 'Ad Spend (INR)'},
                   title='Distribution of Ad Spend',color_discrete_sequence=['darkred'])
fig.update_traces(marker=dict(line=dict(color='black')))
fig.update_layout(bargap=0.1, width=800, height=600)
fig.show()
```

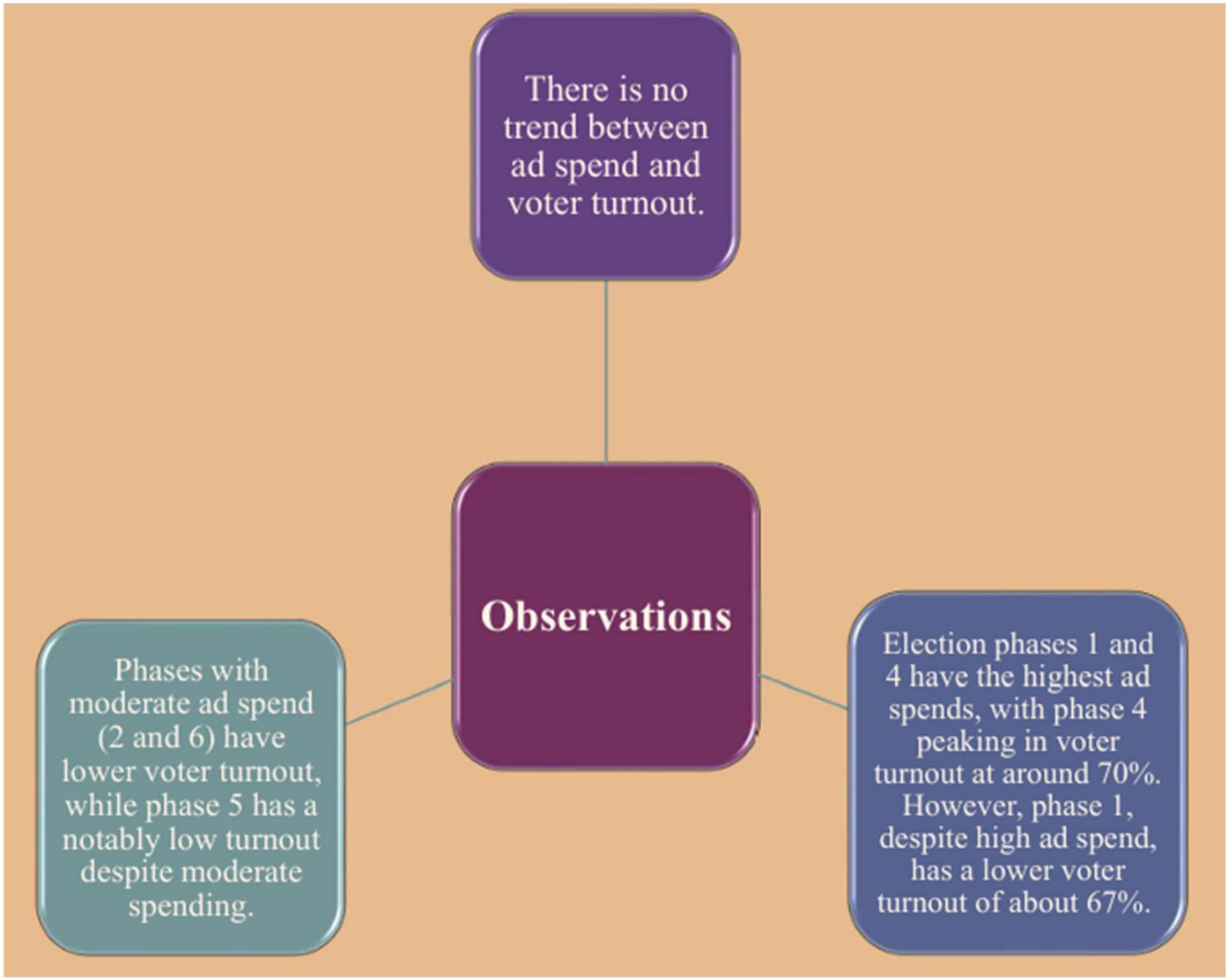




7. Ad spending and voter turnout by election phase

```
import plotly.graph_objects as go
phase_analysis = merged_data.groupby('Phase').agg({'Amount spent (INR)': 'sum','Polled (%)': 'mean'}).reset_index()
fig = go.Figure()
fig.add_trace(go.Bar(x=phase_analysis['Phase'],y=phase_analysis['Amount spent (INR)'],name='Ad Spend (INR)',marker_color='#4B0082',yaxis='y1'))
fig.add_trace(go.Scatter(x=phase_analysis['Phase'],y=phase_analysis['Polled (%)'],name='Voter Turnout (%)',marker_color='lightsalmon',yaxis='y2'))
fig.update_layout(title='Ad Spend and Voter Turnout by Election Phase',xaxis=dict(title='Election Phase'),yaxis=dict(
title='Ad Spend (INR)',titlefont=dict(color='#4B0082'),tickfont=dict(color='#4B0082')),
yaxis2=dict(title='Voter Turnout (%)',titlefont=dict(color='lightsalmon'),tickfont=dict(color='lightsalmon'),overlaying='y',side='right'),
legend=dict(x=0.1, y=1.1, orientation='h'),width=800,height=600)

fig.show()
```




There is no trend between ad spend and voter turnout.
Observations
Phases with moderate ad spend (2 and 6) have lower voter turnout, while phase 5 has a notably low turnout despite moderate spending.
Election phases 1 and 4 have the highest ad spends, with phase 4 peaking in voter turnout at around 70%. However, phase 1, despite high ad spend, has a lower voter turnout of about 67%.