```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

plfs=pd.read_excel("/content/drive/MyDrive/plfs_final.xlsx")

plfs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 62 columns):
 #   Column                                    Non-Null Count  Dtype  
---  ------                                    --------------  -----  
 0   State/UT                                  36 non-null     object 
 1   Rural (Male)                              36 non-null     float64
 2   Rural (Female)                            36 non-null     float64
 3   Rural (Person)                            36 non-null     float64
 4   Urban (Male)                              36 non-null     float64
 5   Urban (Female)                            36 non-null     float64
 6   Urban (Person)                            36 non-null     float64
 7   Rural + Urban (Male)                      36 non-null     object 
 8   Rural + Urban (Female)                    36 non-null     object 
 9   Rural + Urban (Person)                    36 non-null     object 
 10  wpr.Rural (1)                             35 non-null     float64
 11  wpr.Rural (2)                             35 non-null     float64
 12  wpr.Rural (3)                             35 non-null     float64
 13  wpr.Urban (4)                             35 non-null     float64
 14  wpr.Urban (5)                             35 non-null     float64
 15  wpr.Urban (6)                             35 non-null     float64
 16  wpr.Total (7)                             35 non-null     object 
 17  wpr.Total (8)                             35 non-null     object 
 18  wpr.Total (9)                             35 non-null     object 
 19  unemprate.Rural                           36 non-null     float64
 20  unemprate.Urban                           36 non-null     float64
 21  unemprate.Rural + Urban                   36 non-null     float64
 22  unemprate.Rural2                          36 non-null     float64
 23  unemprate.Urban3                          36 non-null     float64
 24  unemprate.Rural + Urban4                  36 non-null     float64
 25  unemprate.Rural5                          36 non-null     object 
 26  unemprate.Urban6                          36 non-null     object 
 27  unemprate.Rural + Urban7                  36 non-null     object 
 28  emprate.Self-Employed (%)                 36 non-null     float64
 29  emprate.Regular Wage/Salary (%)           36 non-null     float64
 30  emprate.Casual Labour (%)                 36 non-null     float64
 31  emprate.Total (%)                         36 non-null     int64  
 32  lfpr_edu.Not Literate                     33 non-null     float64
 33  lfpr_edu.Literate & Upto Primary          33 non-null     float64
 34  lfpr_edu.Middle                           33 non-null     float64
 35  lfpr_edu.Secondary                        33 non-null     float64
```

```
 36  lfpr_edu.Higher Secondary                        33 non-null     float64
 37  lfpr_edu.Diploma/Certificate Course              33 non-null     float64
 38  lfpr_edu.Graduate                                33 non-null     float64
 39  lfpr_edu.Post Graduate & Above                   33 non-null     float64
 40  lfpr_edu.Secondary & Above                       33 non-null     float64
 41  lfpr_edu.All                                     33 non-null     float64
 42  wpr_edu.Not Literate                             36 non-null     float64
 43  wpr_edu.Literate & Up to Primary                 36 non-null     float64
 44  wpr_edu.Middle                                   36 non-null     float64
 45  wpr_edu.Secondary                                36 non-null     float64
 46  wpr_edu.Higher Secondary                         36 non-null     float64
 47  wpr_edu.Diploma/ Certificate Course              36 non-null     float64
 48  wpr_edu.Graduate                                 36 non-null     float64
 49  wpr_edu.Post Graduate & Above                    36 non-null     float64
 50  wpr_edu.Secondary & Above                        36 non-null     float64
 51  wpr_edu.All                                      36 non-null     float64
 52  uemprate_edu.Not Literate                        35 non-null     float64
 53  uemprate_edu.Literate & up to Primary            35 non-null     float64
 54  uemprate_edu.Middle                              35 non-null     float64
 55  uemprate_edu.Secondary                           35 non-null     float64
 56  uemprate_edu.Higher Secondary                    35 non-null     float64
 57  uemprate_edu.Diploma/Certificate Course          35 non-null     float64
 58  uemprate_edu.Graduate                            35 non-null     float64
 59  uemprate_edu.Post Graduate & Above               35 non-null     float64
 60  uemprate_edu.Secondary & Above                   35 non-null     float64
 61  uemprate_edu.All                                 35 non-null     float64
dtypes: float64(51), int64(1), object(10)
memory usage: 17.6+ KB
```

```
plfs.describe()
```

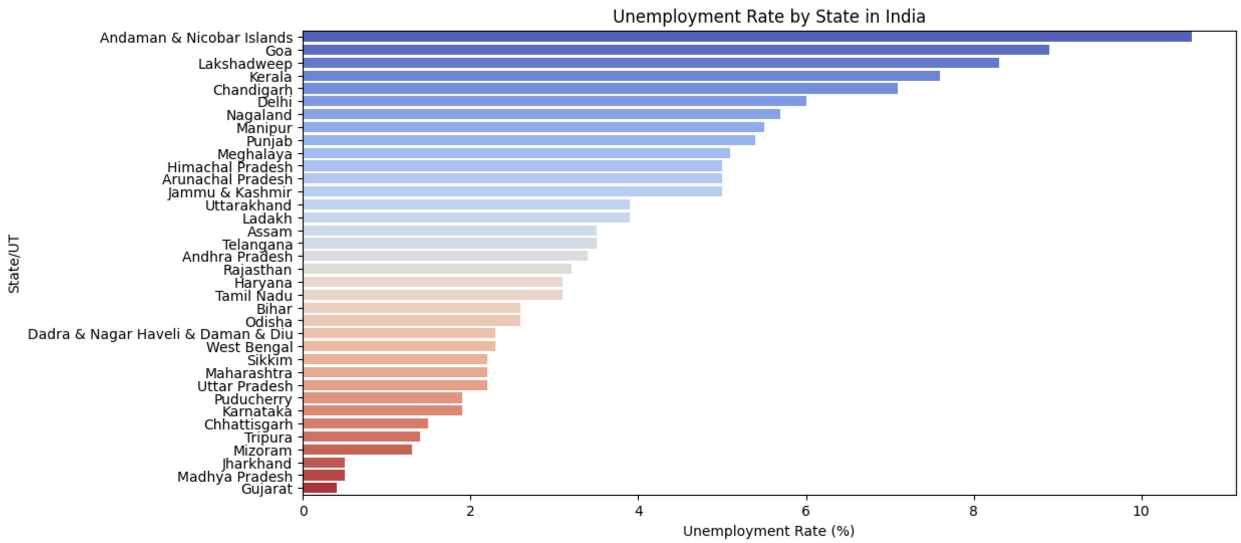
{"type":"dataframe"}

```
plfs_demo=plfs.sort_values(by="unemprate.Rural +
Urban",ascending=False)
```

```
plt.figure(figsize=(12,6))
sns.barplot(x="unemprate.Rural +
Urban",y="State/UT",data=plfs_demo,palette="coolwarm")
plt.xlabel("Unemployment Rate (%)")
plt.ylabel("State/UT")
plt.title("Unemployment Rate by State in India")
plt.show()
```

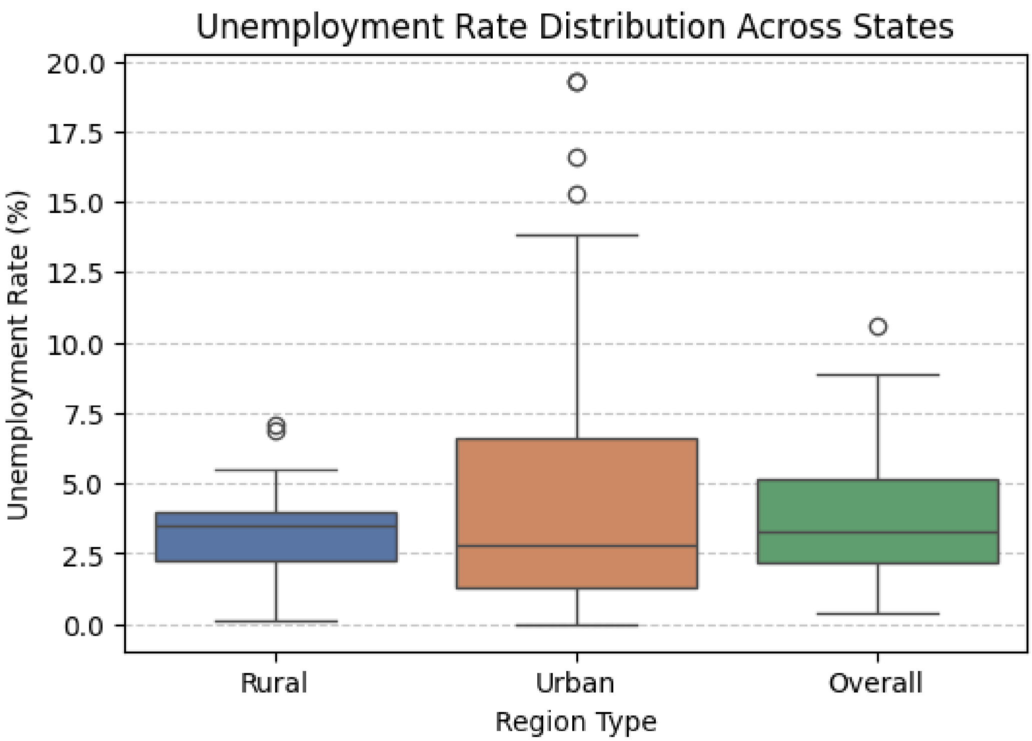
```
<ipython-input-8-83698169e743>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.
```

```
sns.barplot(x="unemprate.Rural + Urban",y="State/UT",data=plfs_demo,palette="coolwarm")
```




```
# Cap extreme outliers at the 99th percentile
upper_limit = np.percentile(plfs[["unemprate.Rural",
"unemprate.Urban", "unemprate.Rural + Urban"]], 99)

# Apply clip to only the relevant columns
plfs_capped = plfs.copy()  # Create a copy to avoid modifying the
original DataFrame
plfs_capped[["unemprate.Rural", "unemprate.Urban", "unemprate.Rural +
Urban"]] = plfs_capped[["unemprate.Rural", "unemprate.Urban",
"unemprate.Rural + Urban"]].clip(upper=upper_limit)
#.clip(upper=upper_limit) replaces any value greater than the 99th
percentile with the threshold.


# Create the boxplot
plt.figure(figsize=(6, 4))  # Adjusted size for better visualization
sns.boxplot(data=plfs_capped[["unemprate.Rural", "unemprate.Urban",
"unemprate.Rural + Urban"]],
            palette=["#4c72b0", "#dd8452", "#55a868"])  # Added color
palette

# Labels and title
plt.xlabel("Region Type")
plt.ylabel("Unemployment Rate (%)")
plt.title("Unemployment Rate Distribution Across States")

# Custom X-ticks
plt.xticks(ticks=[0, 1, 2], labels=["Rural", "Urban", "Overall"])
```

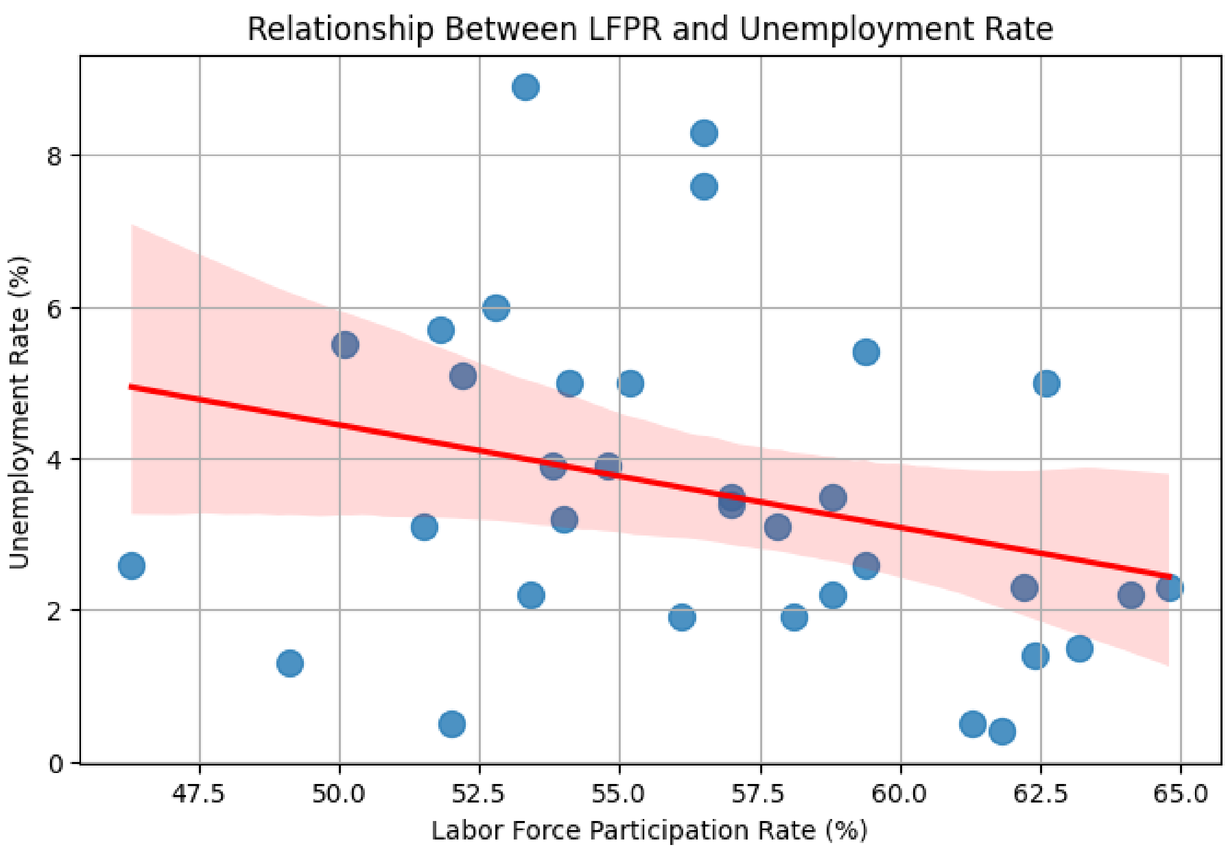
```
# Add grid lines for better readability
plt.grid(axis="y", linestyle="--", alpha=0.7)

plt.show()
```




```
# 1 Higher Unemployment in Urban Areas
# Observation: Urban areas exhibit a higher median unemployment rate
and a greater spread of data compared to rural areas.

# Reasons:
# ✔ Migration Pressure: Rapid urbanization leads to high population
density and increased job competition.
# ✔ Sectoral Dependence: Urban employment relies heavily on
manufacturing, IT, and services, which can be volatile due to economic
downturns.
# ✔ Skill Mismatch: Many urban job seekers do not meet industry
requirements, leading to a higher unemployment rate despite available
vacancies.

# Actionable Policy Measures:
# □ Expansion of skill development programs to align with market
demands.
# □ Encouragement of startups and MSMEs (Micro, Small & Medium
Enterprises) for job creation.
# □ Strengthening of social security schemes to support unemployed
individuals.
```

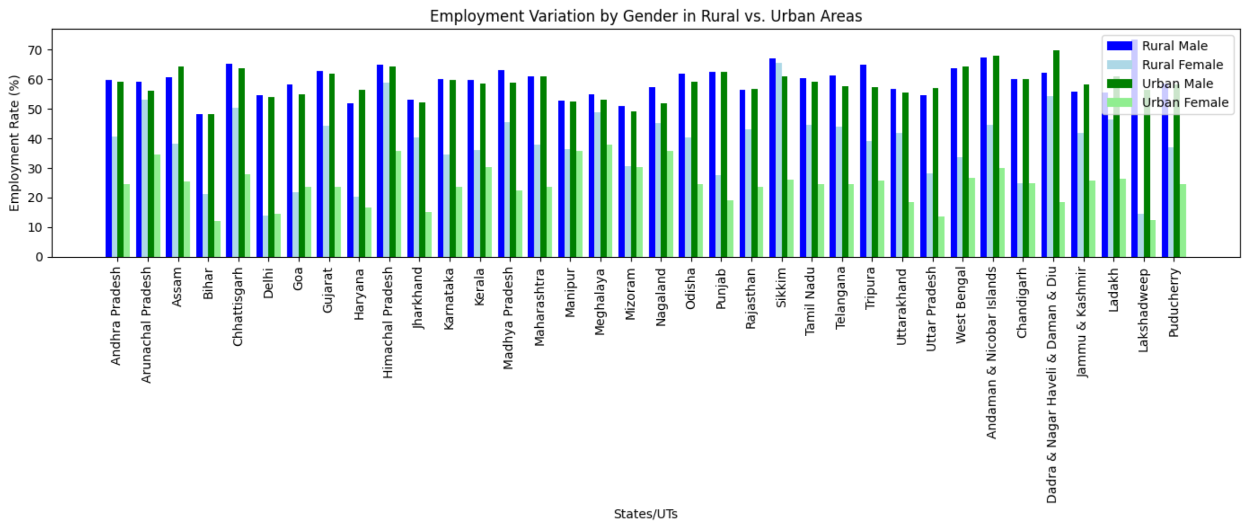
```
2 2 Stability in Rural Unemployment Rates
# Observation: Rural areas display a lower and more stable
unemployment rate with fewer outliers.

# Reasons:
# ✔ Agricultural Employment: A significant portion of the rural
workforce is self-employed in agriculture, reducing visible
unemployment.
# ✔ Government Schemes: Programs such as MGNREGA (Mahatma Gandhi
National Rural Employment Guarantee Act) provide a minimum employment
safety net.
# ✔ Lower Job Competition: Rural job markets tend to be localized and
skill-specific, leading to less variation in employment trends.

# Actionable Policy Measures:
# □ Promotion of rural entrepreneurship through credit facilities and
subsidies.
# □ Enhancement of agri-tech and cooperative farming models for
sustainable employment.
# □ Infrastructure development to attract industries to semi-urban and
rural areas.

3 3 Outliers Indicating Localized Unemployment Crises
# Observation: Some states show extreme outliers, particularly in
urban areas, signaling state-specific unemployment crises.

# Reasons:
# ✔ State-Specific Economic Slowdown: Industries in certain regions
may face decline, automation, or policy shifts affecting employment.
# ✔ Post-Pandemic Job Recovery: Some states may still be recovering
from job losses due to past economic disruptions.
# ✔ Education-Employment Gap: Graduates may struggle with unemployment
due to lack of practical skills.

# Actionable Policy Measures:
# □ Targeted regional employment policies with sector-specific job
drives.
# □ Strengthening of Public-Private Partnerships (PPPs) to boost
industrial employment.
# □ Encouraging investment in high-unemployment regions to create
jobs.


# Convert necessary columns to numeric (if they are stored as object)

df=pd.DataFrame()
df["LFPR"] = pd.to_numeric(plfs["wpr.Total (7)"], errors="coerce")
df["Unemployment Rate"] = pd.to_numeric(plfs["unemprate.Rural +
Urban"], errors="coerce")
```

```python
# Drop rows with NaN values
df = df.dropna(subset=["LFPR", "Unemployment Rate"])

# Scatter plot with regression line
plt.figure(figsize=(8, 5))
sns.regplot(x=df["LFPR"], y=df["Unemployment Rate"], scatter_kws={"s":
100}, line_kws={"color": "red"})

plt.xlabel("Labor Force Participation Rate (%)")
plt.ylabel("Unemployment Rate (%)")
plt.title("Relationship Between LFPR and Unemployment Rate")
plt.grid(True)
plt.show()
```



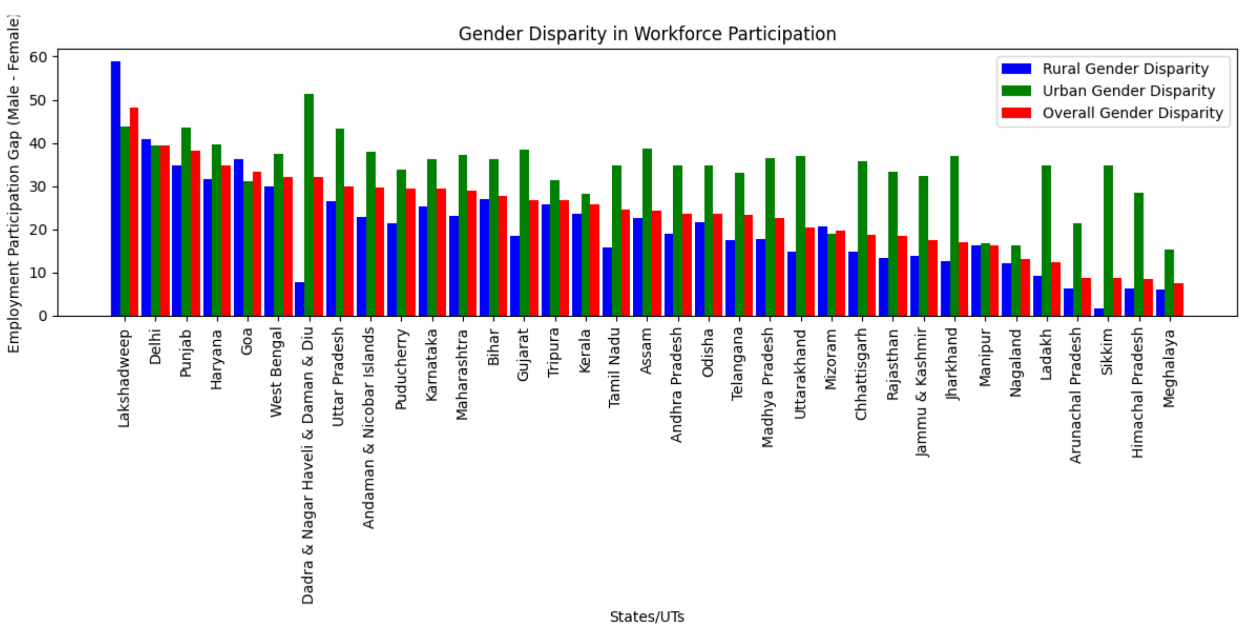

```python
# A higher LFPR (more people in the labor force) is generally
associated with a lower unemployment rate.

# This could indicate that as more people engage in the workforce, job
opportunities also rise, reducing unemployment.

# However, other economic factors such as job availability, economic
policies, and market conditions can also impact this relationship.
```

```python
# Extract relevant columns
states = plfs['State/UT']
rural_male =plfs['Rural (Male)']
rural_female = plfs['Rural (Female)']
urban_male = plfs['Urban (Male)']
urban_female = plfs['Urban (Female)']

# Set bar width and x-axis positions
bar_width = 0.2
x = np.arange(len(states))

# Create the grouped bar chart
fig, ax = plt.subplots(figsize=(14, 6))
ax.bar(x - bar_width*1.5, rural_male, bar_width, label='Rural Male',
color='blue')
ax.bar(x - bar_width/2, rural_female, bar_width, label='Rural Female',
color='lightblue')
ax.bar(x + bar_width/2, urban_male, bar_width, label='Urban Male',
color='green')
ax.bar(x + bar_width*1.5, urban_female, bar_width, label='Urban
Female', color='lightgreen')

# Labels and formatting
ax.set_xlabel('States/UTs')
ax.set_ylabel('Employment Rate (%)')
ax.set_title('Employment Variation by Gender in Rural vs. Urban
Areas')
ax.set_xticks(x)
ax.set_xticklabels(states, rotation=90)
ax.legend()
plt.tight_layout()

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



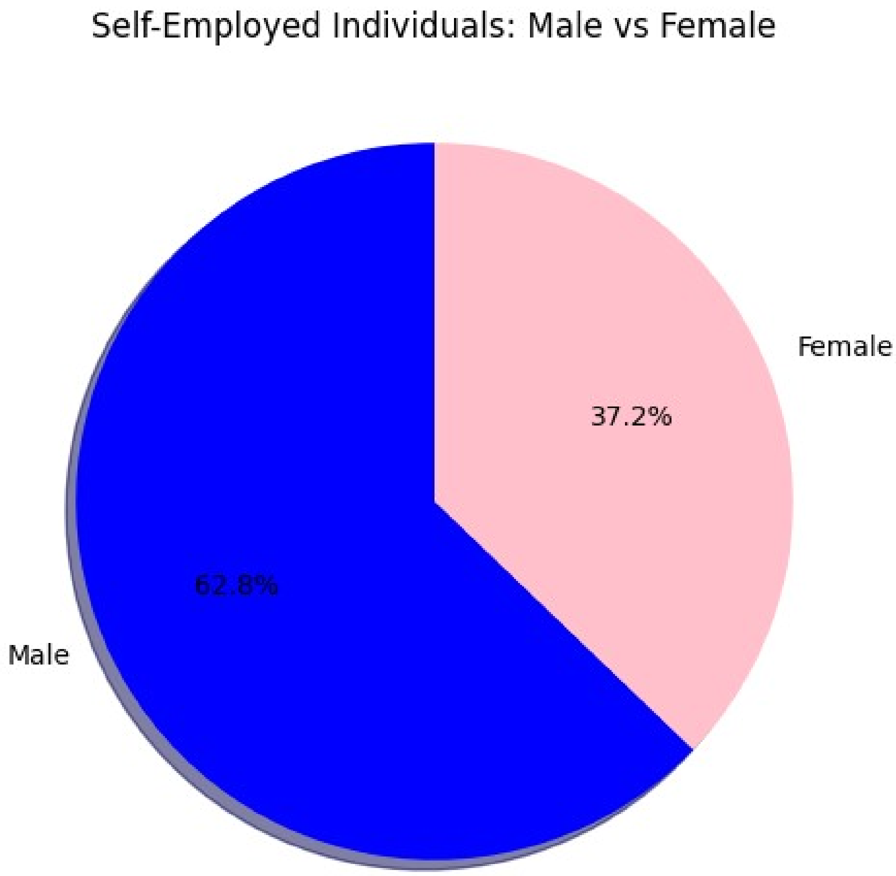

```
# Males dominate employment rates in both rural and urban areas.

# Urban areas provide better employment opportunities for females
compared to rural areas, but gender disparity still exists.

# The employment rate difference between rural and urban males is
smaller than that between rural and urban females.

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Assuming 'plfs' is the DataFrame
df = plfs.copy()  # Work on a copy to avoid modifying original data

# Convert necessary columns to numeric
df["Rural + Urban (Male)"] = pd.to_numeric(df["Rural + Urban (Male)"],
errors='coerce')
df["Rural + Urban (Female)"] = pd.to_numeric(df["Rural + Urban
(Female)"], errors='coerce')

# Calculate gender disparity in employment rates
df["Rural Gender Disparity"] = df["Rural (Male)"] - df["Rural
(Female)"]
df["Urban Gender Disparity"] = df["Urban (Male)"] - df["Urban
(Female)"]
df["Overall Gender Disparity"] = df["Rural + Urban (Male)"] -
df["Rural + Urban (Female)"]

# Drop rows with NaN values in disparity columns
df.dropna(subset=["Overall Gender Disparity", "Rural Gender
Disparity", "Urban Gender Disparity"], inplace=True)

# Sorting by disparity for better visualization
```

```
df_sorted = df.sort_values(by="Overall Gender Disparity",
ascending=False)

# Update x-axis values to match the number of valid rows
x = np.arange(len(df_sorted))  # Corrected x size

# Plotting
width = 0.3  # Bar width
fig, ax = plt.subplots(figsize=(12, 6))

bars1 = ax.bar(x - width, df_sorted["Rural Gender Disparity"], width,
label='Rural Gender Disparity', color='blue')
bars2 = ax.bar(x, df_sorted["Urban Gender Disparity"], width,
label='Urban Gender Disparity', color='green')
bars3 = ax.bar(x + width, df_sorted["Overall Gender Disparity"],
width, label='Overall Gender Disparity', color='red')

# Labels and title
ax.set_xlabel("States/UTs")
ax.set_ylabel("Employment Participation Gap (Male - Female)")
ax.set_title("Gender Disparity in Workforce Participation")
ax.set_xticks(x)
ax.set_xticklabels(df_sorted["State/UT"], rotation=90)
ax.legend()

# Show plot
plt.tight_layout()
plt.show()
```



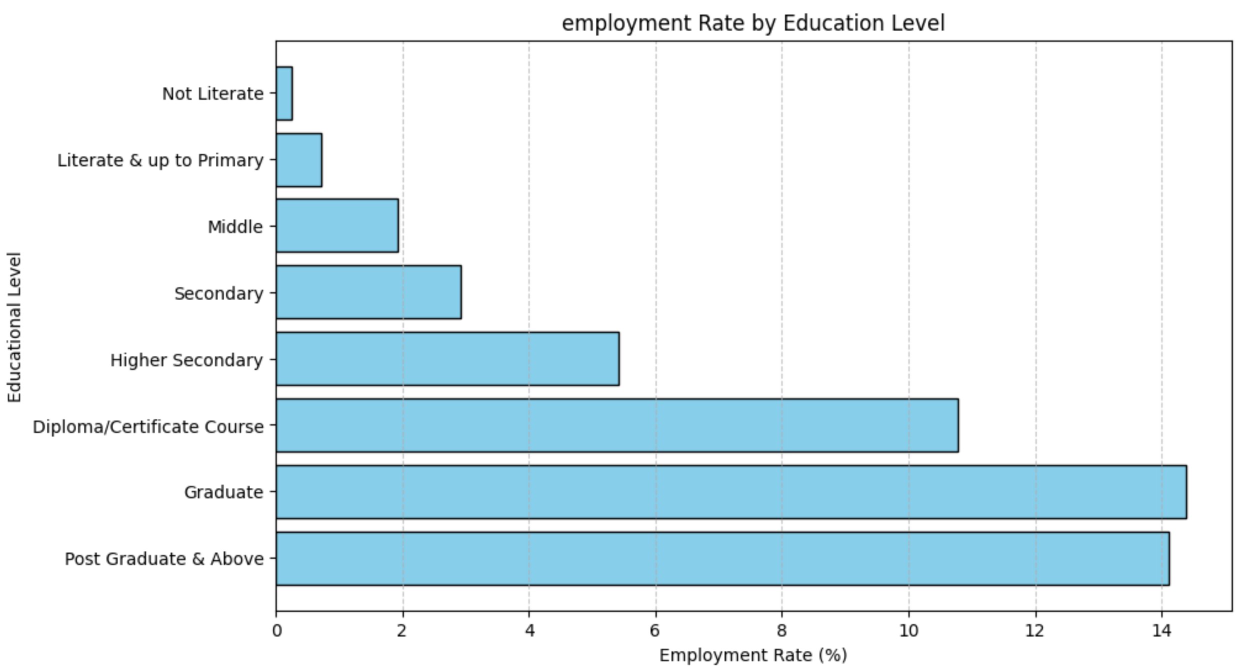
```
# The following observations highlight the disparity:

# Consistently Higher Male Participation: The employment participation
gap (Male - Female) is positive across all regions, showing that male
workforce participation is significantly higher than female
participation.

# State-wise Variation:

# Some states, like Lakshadweep, Delhi, Punjab, and Haryana, exhibit
high disparities, with large differences between male and female
employment participation.

# States like Meghalaya, Himachal Pradesh, and Sikkim have relatively
lower disparities, though a gap still exists.

# Rural vs. Urban Differences:

# The urban gender disparity (green bars) tends to be higher in
several states, indicating that female employment participation is
particularly low in urban areas.

# Rural disparity (blue bars) also remains significant but is
sometimes lower than urban gaps.

# Overall Gender Disparity:

# The overall workforce participation gap (red bars) remains high in
most regions, reinforcing the fact that women's participation in the
workforce is substantially lower than men's.

df=plfs.copy()

# Convert columns to numeric, handling errors
df['Rural + Urban (Male)'] = pd.to_numeric(df['Rural + Urban (Male)'],
errors='coerce')
df['Rural + Urban (Female)'] = pd.to_numeric(df['Rural + Urban
(Female)'], errors='coerce')

# Fill NaN values with 0
df.fillna(0, inplace=True)

# Compute totals
male_self_employed = df['Rural + Urban (Male)'].sum()
female_self_employed = df['Rural + Urban (Female)'].sum()

# Labels and values
labels = ['Male', 'Female']
sizes = [male_self_employed, female_self_employed]
colors = ['blue', 'pink']
```

```python
# Create Pie Chart
plt.figure(figsize=(6,6))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%',
startangle=90, shadow=True)
plt.title("Self-Employed Individuals: Male vs Female")
plt.show()
```



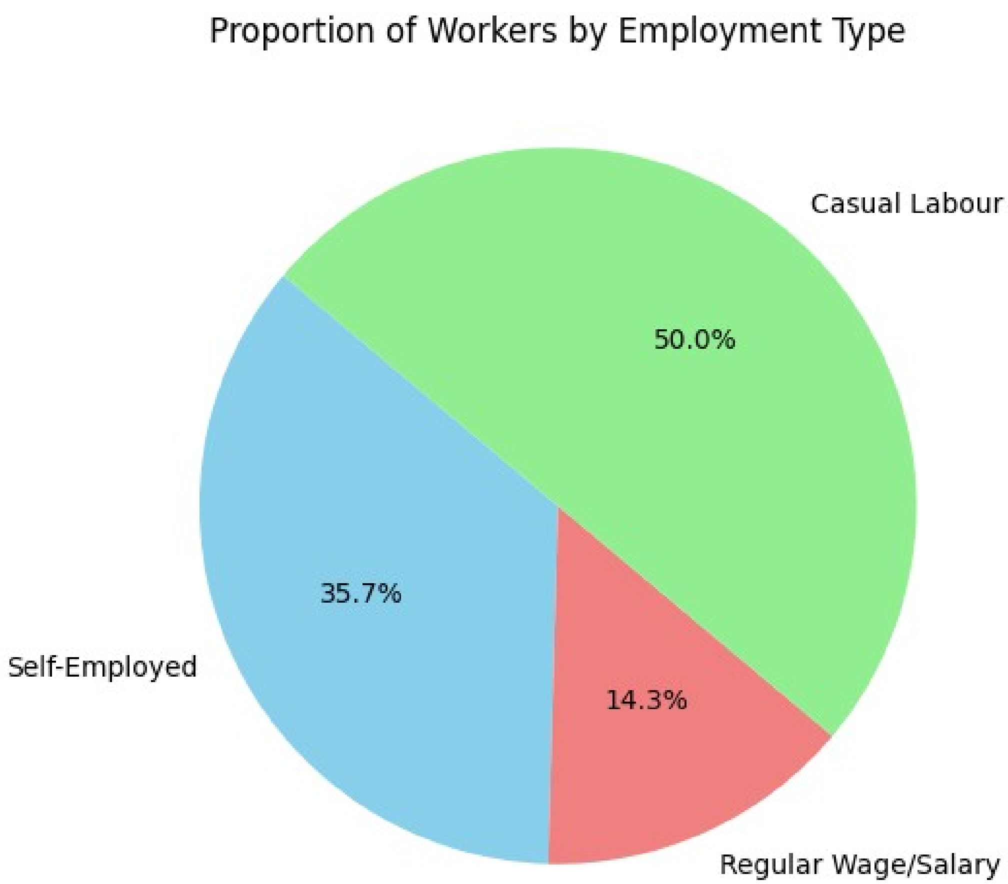

```python
# This means that 62.8% of self-employed individuals are male, while
37.2% are female


# Copying the dataset
df = plfs.copy()

# Selecting relevant columns for unemployment rate by education
education_levels = [
    "uemprate_edu.Not Literate",
    "uemprate_edu.Literate & up to Primary",
    "uemprate_edu.Middle",
    "uemprate_edu.Secondary",
```

```
    "uemprate_edu.Higher Secondary",
    "uemprate_edu.Diploma/Certificate Course",
    "uemprate_edu.Graduate",
    "uemprate_edu.Post Graduate & Above",
]

# Calculate the average unemployment rate for each education level
unemployment_rates = df[education_levels].mean()

# Clean labels by removing 'uemprate_edu.'
clean_labels = [label.replace("uemprate_edu.", "") for label in
unemployment_rates.index]

# Plotting the bar chart
plt.figure(figsize=(10, 6))
plt.barh(clean_labels, unemployment_rates.values, color='skyblue',
edgecolor='black')
plt.xlabel("Employment Rate (%)")
plt.ylabel("Educational Level")
plt.title("employment Rate by Education Level")
plt.gca().invert_yaxis()  # Invert y-axis for better readability
plt.grid(axis="x", linestyle="--", alpha=0.7)
plt.show()
```



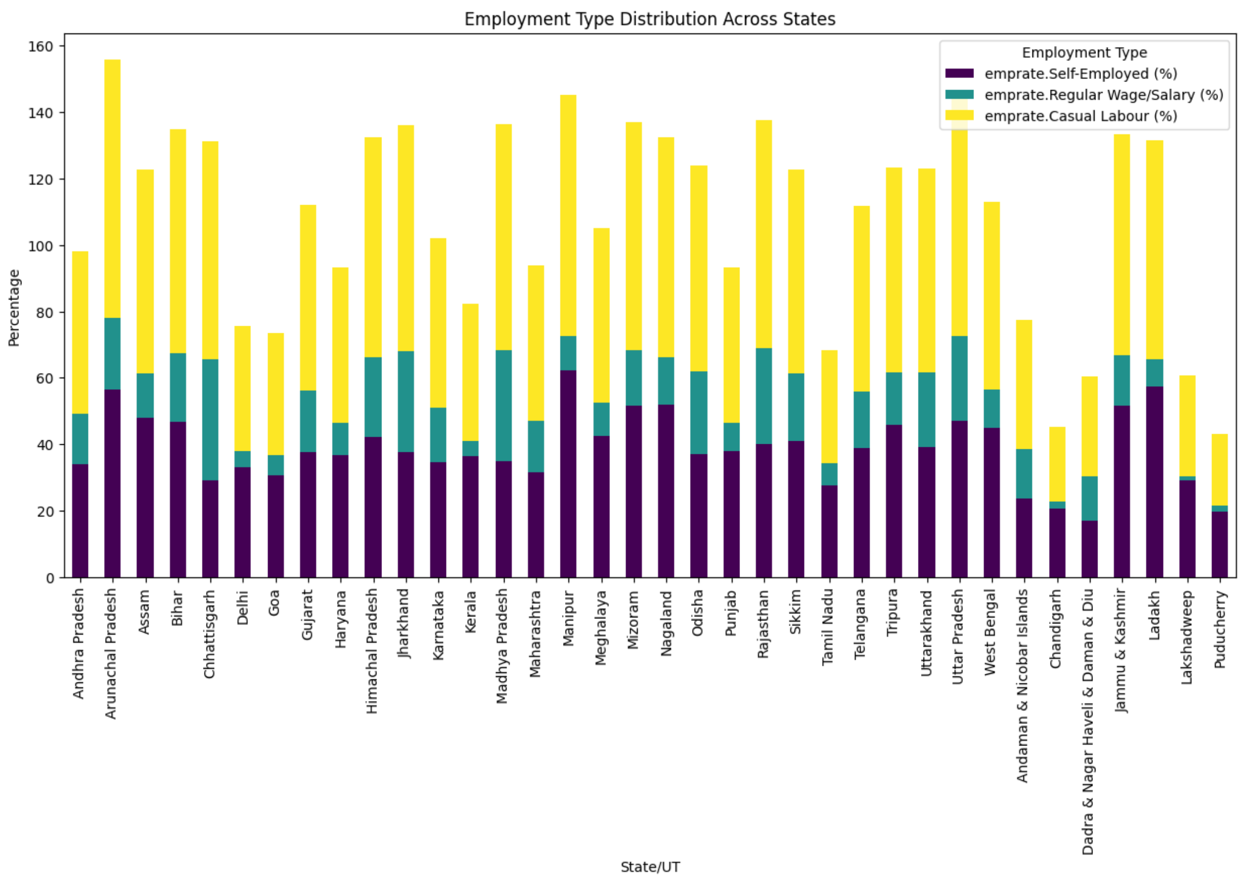

```
import matplotlib.pyplot as plt

# Extract relevant employment proportions
employment_types = [
    "Self-Employed",
```

```
    "Regular Wage/Salary",
    "Casual Labour"
]

proportions = [
    df["emprate.Self-Employed (%)"].mean(),
    df["emprate.Regular Wage/Salary (%)"].mean(),
    df["emprate.Casual Labour (%)"].mean()
]

# Pie Chart for Employment Distribution
plt.figure(figsize=(8, 6))
plt.pie(proportions, labels=employment_types, autopct="%1.1f%%",
colors=["skyblue", "lightcoral", "lightgreen"], startangle=140)
plt.title("Proportion of Workers by Employment Type")
plt.show()
```




```
# Select relevant columns
employment_columns = ['State/UT', 'emprate.Self-Employed (%)',
```

```
'emprate.Regular Wage/Salary (%)', 'emprate.Casual Labour (%)']
df = df[employment_columns]

# Set state as index
df.set_index('State/UT', inplace=True)

# Plot employment type distribution across states
df.plot(kind='bar', stacked=True, figsize=(15, 7), colormap='viridis')
plt.title('Employment Type Distribution Across States')
plt.xlabel('State/UT')
plt.ylabel('Percentage')
plt.legend(title='Employment Type')
plt.xticks(rotation=90)
plt.show()
```

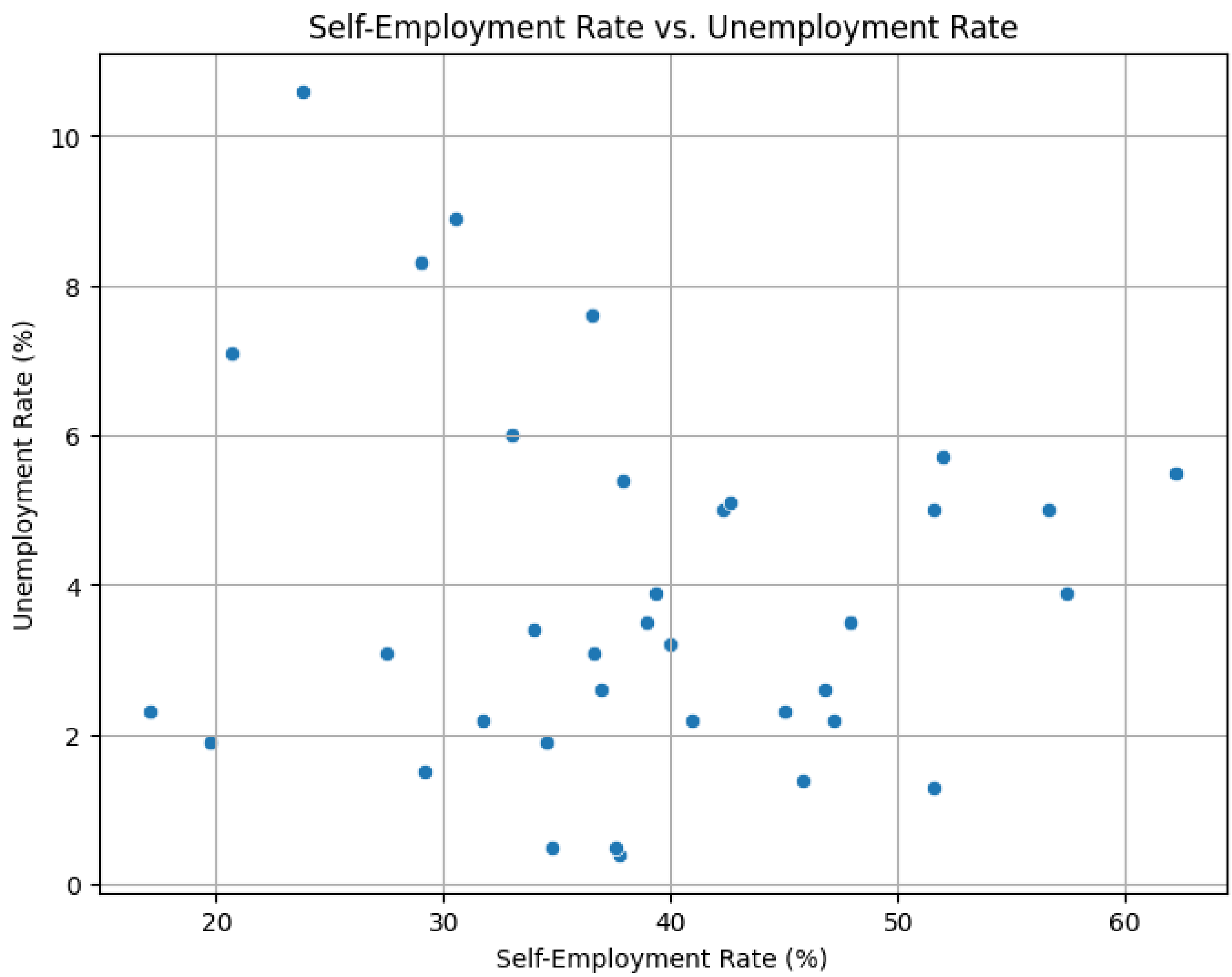



```
# The employment type distribution varies significantly across Indian
states, influenced by economic conditions, industrial presence, and
regional factors. The three primary employment categories—Self-
Employed, Regular Wage/Salary, and Casual Labour—are distributed
differently across states due to various reasons:

# 1. High Self-Employment States (e.g., Himachal Pradesh, Arunachal
```

```
Pradesh, Rajasthan)
# Reason:

# These states have a large agricultural and small-business-based economy.

# Himachal Pradesh and Arunachal Pradesh have challenging terrains, leading to more self-employment in agriculture, handicrafts, and tourism-related businesses.

# Rajasthan has a significant number of small traders, entrepreneurs, and artisans involved in textiles, pottery, and other cottage industries.

# 2. High Regular Wage/Salary Employment States (e.g., Delhi, Goa, Tamil Nadu, Karnataka)
# Reason:

# These states have a strong industrial and service sector presence.

# Delhi and Karnataka (Bangalore) have a booming IT and corporate sector, leading to more formal jobs with regular salaries.

# Goa has a tourism-driven economy, where hotels, casinos, and restaurants provide stable wage employment.

# Tamil Nadu is an industrial hub for manufacturing and automobile industries, ensuring a significant proportion of salaried jobs.

# 3. High Casual Labour States (e.g., Bihar, Odisha, Madhya Pradesh, Uttar Pradesh)
# Reason:

# These states have a large number of migrant laborers and daily wage earners.

# Bihar and Uttar Pradesh have high population density but lower industrialization, leading to more people depending on informal sector jobs.

# Odisha and Madhya Pradesh have a substantial number of workers involved in construction, agriculture, and low-skilled labor-intensive jobs.

# 4. Balanced Employment Distribution (e.g., Maharashtra, Gujarat, Punjab, West Bengal)
# Reason:

# These states have a diverse economic base, including agriculture, industries, and services.
```

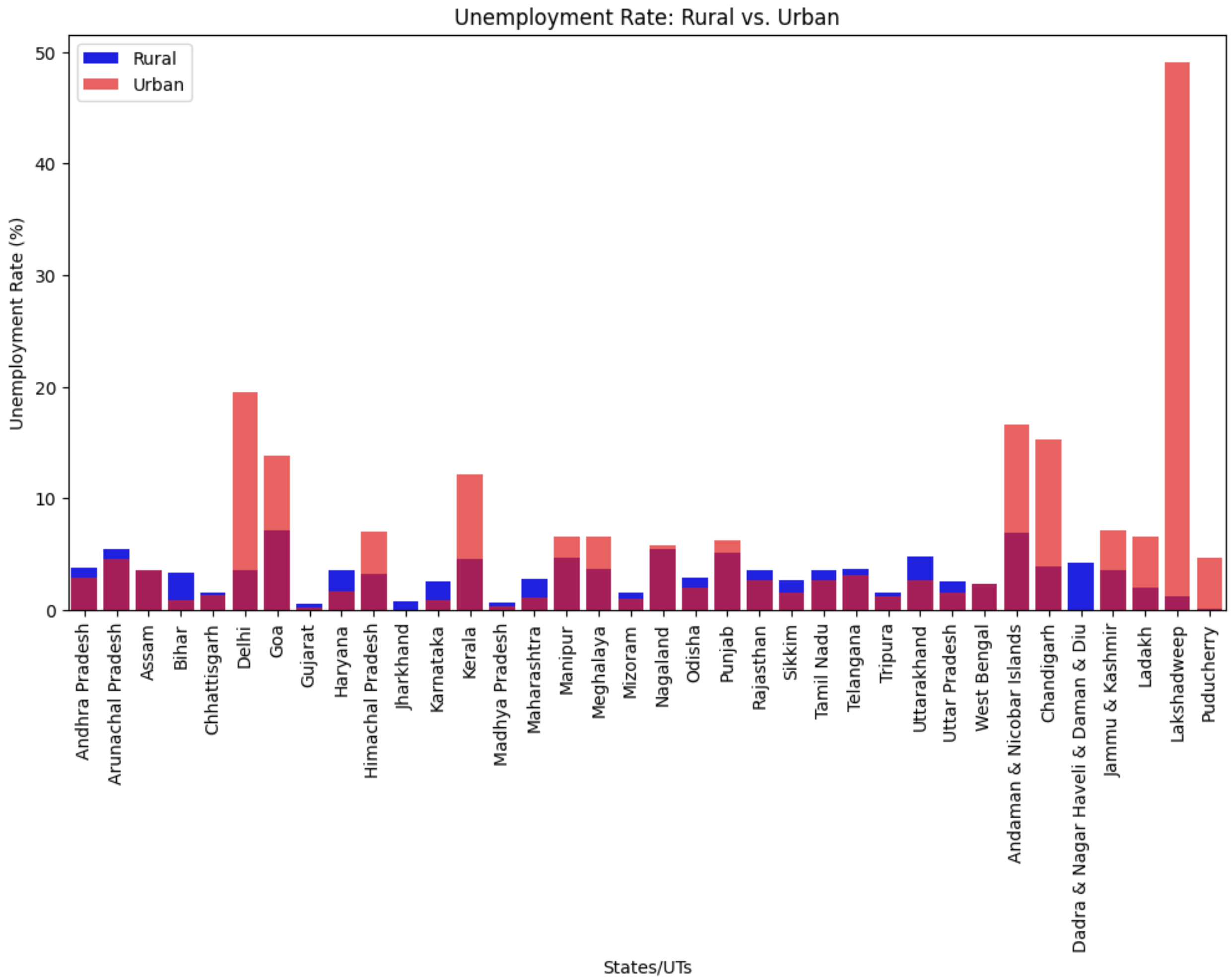
```
# Maharashtra (Mumbai, Pune) has a financial hub, industries, and a
large informal workforce.

# Gujarat has thriving business communities, textile industries, and
agriculture.

# Punjab is agriculturally rich, but urban centers like Ludhiana and
Amritsar also provide wage employment.

# West Bengal has a mix of trade, agriculture, and industrial jobs,
creating a balanced employment structure.

# Conclusion
# Agriculture-dominated states show higher self-employment.

# Industrialized and service-oriented states have a higher proportion
of salaried employment.

# States with underdeveloped economies rely more on casual labor.

# Urbanization, industrial policies, and economic development are key
factors influencing employment types.

df=plfs.copy()

# Selecting relevant columns
self_employment = df['emprate.Self-Employed (%)']
unemployment = df['unemprate.Rural + Urban']

# Handling missing or non-numeric values
df_filtered = df[['emprate.Self-Employed (%)', 'unemprate.Rural +
Urban']].dropna()
df_filtered = df_filtered.apply(pd.to_numeric,
errors='coerce').dropna()

# Calculating correlation
correlation = df_filtered.corr().iloc[0, 1]
print(f'Correlation between Self-Employment Rate and Unemployment
Rate: {correlation:.2f}')

# Scatter plot
plt.figure(figsize=(8, 6))
sns.scatterplot(x=df_filtered['emprate.Self-Employed (%)'],
y=df_filtered['unemprate.Rural + Urban'])
plt.xlabel('Self-Employment Rate (%)')
plt.ylabel('Unemployment Rate (%)')
plt.title('Self-Employment Rate vs. Unemployment Rate')
plt.grid()
plt.show()
```

```
Correlation between Self-Employment Rate and Unemployment Rate: -0.11
```

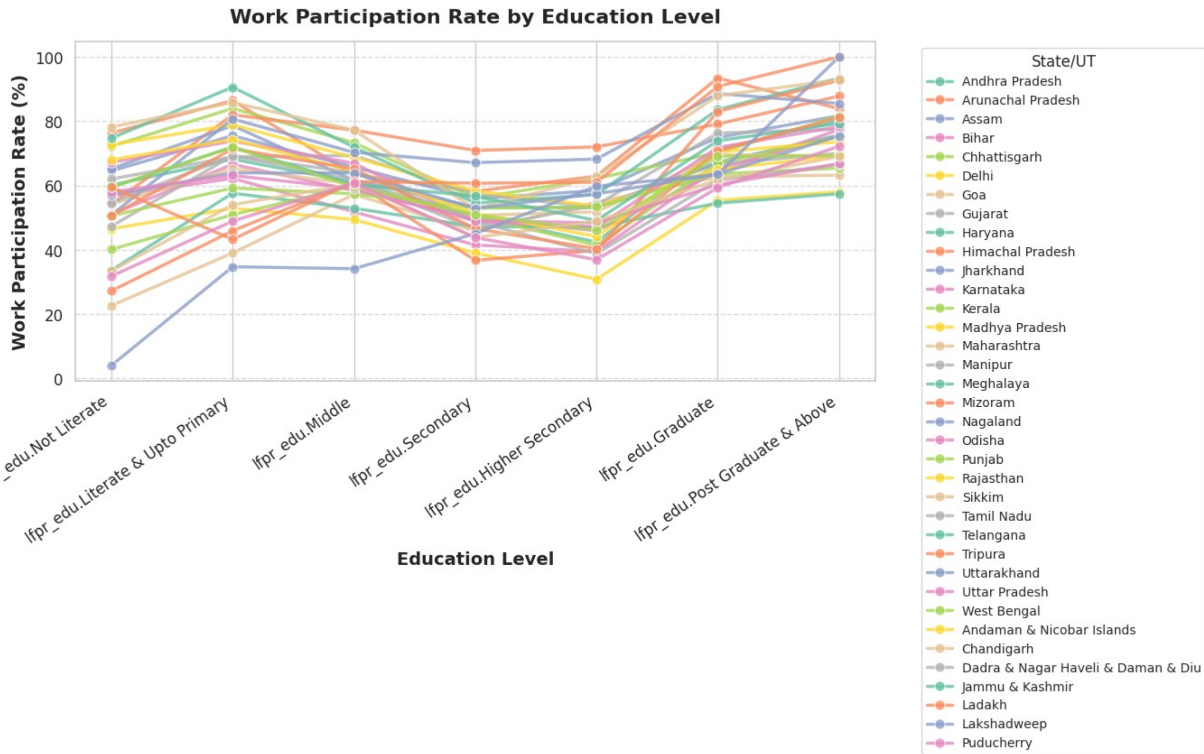



```
# Convert object columns to numeric where necessary
cols_to_convert = ['Rural + Urban (Male)', 'Rural + Urban (Female)',
'Rural + Urban (Person)',
                   'wpr.Total (7)', 'wpr.Total (8)', 'wpr.Total (9)',
                   'unemprate.Rural5', 'unemprate.Urban6',
'unemprate.Rural + Urban7']
for col in cols_to_convert:
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Plot 1: Unemployment Rate (Rural vs. Urban)
plt.figure(figsize=(12, 6))
sns.barplot(x=df["State/UT"], y=df["unemprate.Rural"], color='blue',
label="Rural")
sns.barplot(x=df["State/UT"], y=df["unemprate.Urban"], color='red',
alpha=0.7, label="Urban")
plt.xticks(rotation=90)
plt.ylabel("Unemployment Rate (%)")
plt.xlabel("States/UTs")
plt.title("Unemployment Rate: Rural vs. Urban")
plt.legend()
```

```
plt.show()

education_levels = ["lfpr_edu.Not Literate", "lfpr_edu.Literate & Upto
Primary",
                    "lfpr_edu.Middle", "lfpr_edu.Secondary",
"lfpr_edu.Higher Secondary",
                    "lfpr_edu.Graduate", "lfpr_edu.Post Graduate &
Above"]

# Melt DataFrame for seaborn plotting
df_melted_edu = df.melt(id_vars=["State/UT"],
value_vars=education_levels,
                         var_name="Education Level", value_name="Work
Participation Rate")

# Set plot style
sns.set_theme(style="whitegrid")

plt.figure(figsize=(14, 7))

# Use a visually appealing color palette
palette = sns.color_palette("Set2", len(df["State/UT"].unique()))

# Create the line plot with enhancements
sns.lineplot(x="Education Level", y="Work Participation Rate",
hue="State/UT",
             data=df_melted_edu, marker='o', linewidth=2.5,
markersize=8, alpha=0.8, palette=palette)

# Rotate x-axis labels for better readability
plt.xticks(rotation=35, ha="right", fontsize=12)
plt.yticks(fontsize=12)

# Add grid for better readability
plt.grid(axis='y', linestyle="--", alpha=0.7)

# Labels and title with improved styling
plt.xlabel("Education Level", fontsize=14, fontweight='bold')
plt.ylabel("Work Participation Rate (%)", fontsize=14,
fontweight='bold')
plt.title("Work Participation Rate by Education Level", fontsize=16,
fontweight='bold', pad=15)

# Show only a limited legend for clarity
plt.legend(title="State/UT", bbox_to_anchor=(1.05, 1), loc='upper
left', fontsize=10, frameon=True)

# Display the plot
plt.tight_layout()
plt.show()
```



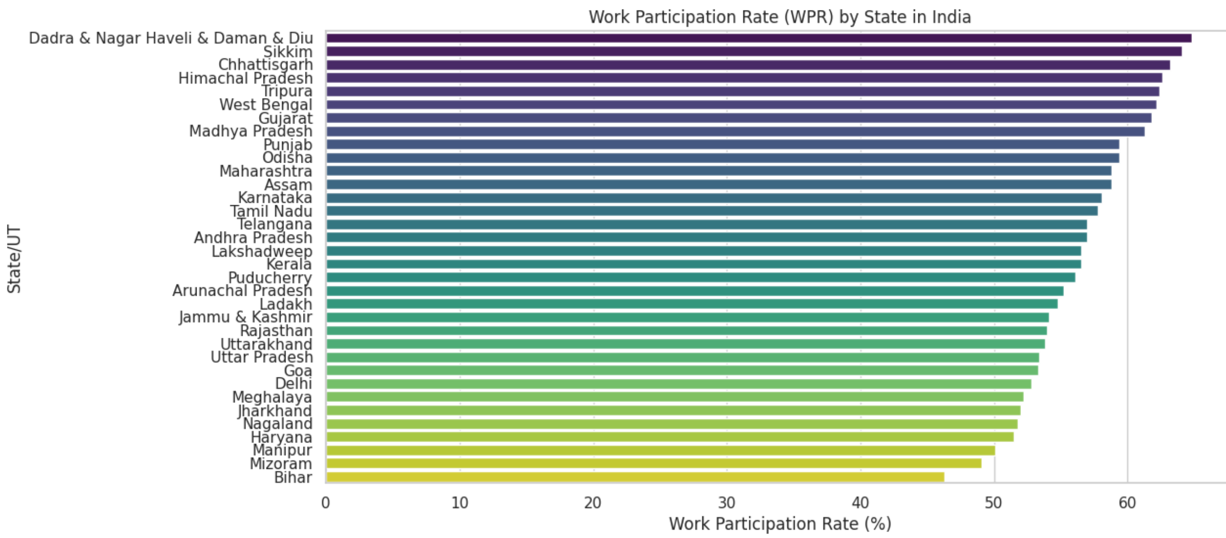
Unemployment Rate: Rural vs. Urban
Rural
Urban
Unemployment Rate (%)
0
10
20
30
40
50
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chhattisgarh
Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jharkhand
Karnataka
Kerala
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttarakhand
Uttar Pradesh
West Bengal
Andaman & Nicobar Islands
Chandigarh
Dadra & Nagar Haveli & Daman & Diu
Jammu & Kashmir
Ladakh
Lakshadweep
Puducherry
States/UTs




```python
# Convert WPR columns to numeric (handling errors for object columns)
wpr_columns = ['wpr.Rural (1)', 'wpr.Rural (2)', 'wpr.Rural (3)',
               'wpr.Urban (4)', 'wpr.Urban (5)', 'wpr.Urban (6)',
               'wpr.Total (7)', 'wpr.Total (8)', 'wpr.Total (9)']

for col in wpr_columns:
    plfs[col] = pd.to_numeric(plfs[col], errors='coerce')  # Convert
to numeric, replacing errors with NaN

# Drop rows with missing WPR data
plfs_clean = plfs.dropna(subset=wpr_columns)

# Compute regional averages
wpr_summary = plfs_clean[['State/UT', 'wpr.Rural (1)', 'wpr.Urban
(4)', 'wpr.Total (7)']].copy()
wpr_summary.columns = ['State/UT', 'Rural WPR', 'Urban WPR', 'Total
WPR']

# Sort by Total WPR
wpr_summary_sorted = wpr_summary.sort_values(by='Total WPR',
ascending=False)

plt.figure(figsize=(12, 6))
sns.barplot(x='Total WPR', y='State/UT', data=wpr_summary_sorted,
palette='viridis')
plt.xlabel('Work Participation Rate (%)')
plt.ylabel('State/UT')
```

```
plt.title('Work Participation Rate (WPR) by State in India')
plt.show()
```

```
<ipython-input-24-9cc00f4700b8>:20: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='Total WPR', y='State/UT', data=wpr_summary_sorted,
palette='viridis')
```




```
df=plfs.copy()

# Selecting only relevant columns
df_filtered = df[['State/UT', 'wpr_edu.All', 'uemprate_edu.All']]

# Convert columns to numeric (handling errors)
df_filtered['wpr_edu.All'] = pd.to_numeric(df_filtered['wpr_edu.All'],
errors='coerce')
df_filtered['uemprate_edu.All'] =
pd.to_numeric(df_filtered['uemprate_edu.All'], errors='coerce')

# Drop missing values
df_filtered = df_filtered.dropna()

# Display summary
df_filtered.head()
```

```
<ipython-input-25-8dea3c751342>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_filtered['wpr_edu.All'] =
pd.to_numeric(df_filtered['wpr_edu.All'], errors='coerce')
<ipython-input-25-8dea3c751342>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df_filtered['uemprate_edu.All'] =
pd.to_numeric(df_filtered['uemprate_edu.All'], errors='coerce')
```

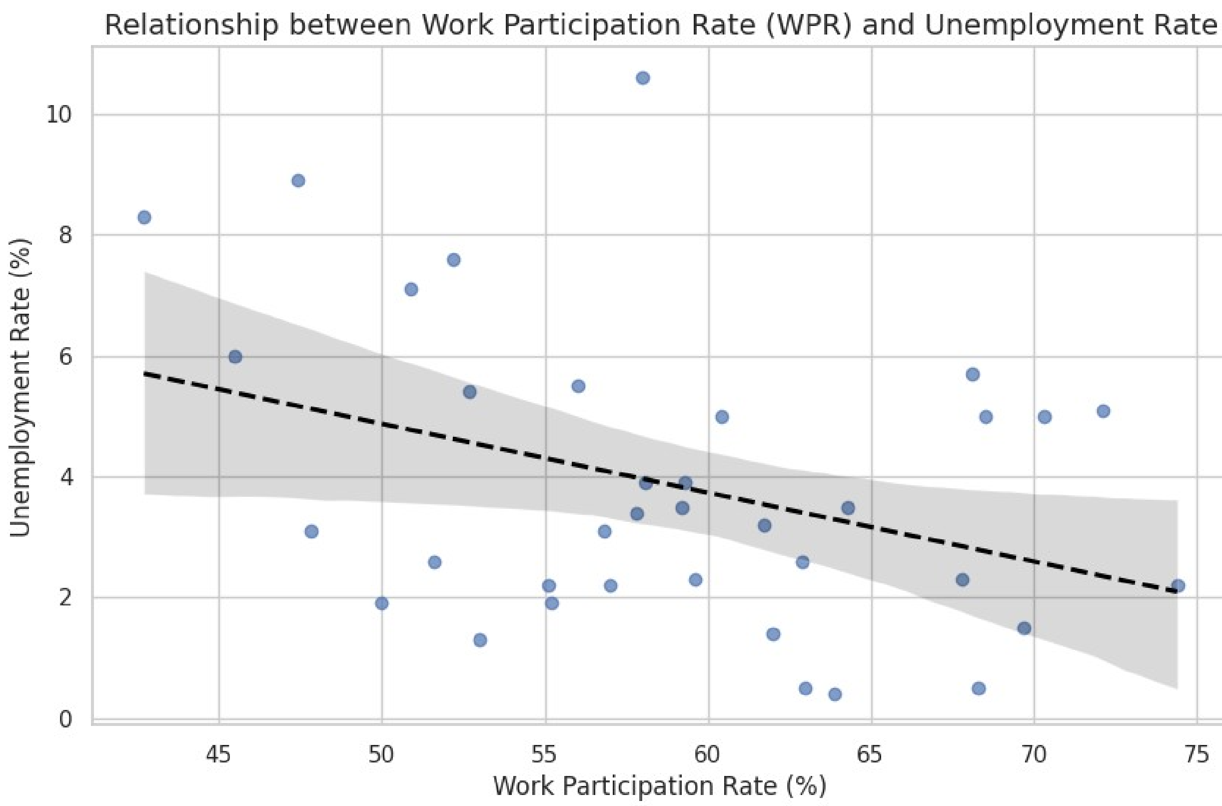
{"summary":"{\n \"name\": \"df_filtered\",\n \"rows\": 35,\n \"fields\": [\n   {\n   \"column\": \"State/UT\",\n \"properties\": {\n   \"dtype\": \"string\",\n \"num_unique_values\": 35,\n   \"samples\": [\n \"Uttarakhand\",\n   \"Madhya Pradesh\",\n \"Telangana\"\n   ],\n   \"semantic_type\": \"\",\n \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"wpr_edu.All\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 7.990743594514674,\n   \"min\": 42.7,\n   \"max\": 74.4,\n   \"num_unique_values\": 35,\n \"samples\": [\n   58.1,\n   68.3,\n   59.2\n],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n }\n   },\n   {\n   \"column\": \"uemprate_edu.All\",\n \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 2.343021105146017,\n   \"min\": 1.0,\n   \"max\": 11.9,\n \"num_unique_values\": 27,\n   \"samples\": [\n   3.4,\n 1.0,\n   5.5\n   ],\n   \"semantic_type\": \"\",\n \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"df_filtered"}

```python
# Calculate correlation
correlation =
df_filtered['wpr_edu.All'].corr(df_filtered['uemprate_edu.All'])

print(f"Correlation between Work Participation Rate (WPR) and
Unemployment Rate: {correlation:.2f}")
```

```
Correlation between Work Participation Rate (WPR) and Unemployment
Rate: -0.33
```

```python
# Selecting relevant columns
wpr_column = "wpr_edu.All"  # Adjust based on the dataset
unemployment_column = "unemprate.Rural + Urban"  # Adjust based on the
dataset
```

```python
# Set plot style
sns.set_style("whitegrid")
plt.figure(figsize=(10, 6))

# Scatter plot with regression line
sns.regplot(x=df[wpr_column], y=df[unemployment_column],
scatter_kws={'alpha':0.7}, line_kws={'color':'black',
'linestyle':'dashed'})

# Titles and labels
plt.title("Relationship between Work Participation Rate (WPR) and
Unemployment Rate", fontsize=14)
plt.xlabel("Work Participation Rate (%)", fontsize=12)
plt.ylabel("Unemployment Rate (%)", fontsize=12)

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




```python
\
# Load data (assuming df is your DataFrame)
df['urban_percentage'] = df['Urban (Person)'] / (df['Urban (Person)']
+ df['Rural (Person)']) * 100

# Define urbanization threshold (median-based categorization)
```

```
median_urban = df['urban_percentage'].median()
df['urbanization_level'] = df['urban_percentage'].apply(lambda x:
'High Urban' if x >= median_urban else 'Low Urban')

# Group by urbanization level
employment_comparison = df.groupby('urbanization_level').agg({
    'wpr.Urban (4)': 'mean',  # Urban WPR
    'wpr.Rural (1)': 'mean',  # Rural WPR
    'unemprate.Urban': 'mean',  # Urban Unemployment Rate
    'unemprate.Rural': 'mean',  # Rural Unemployment Rate
    'emprate.Self-Employed (%)': 'mean',
    'emprate.Regular Wage/Salary (%)': 'mean',
    'emprate.Casual Labour (%)': 'mean'
})

print(employment_comparison)


                    wpr.Urban (4)  wpr.Rural (1)  unemprate.Urban  \
urbanization_level
High Urban              54.611765      54.729412         6.850000
Low Urban               55.750000      59.872222         5.105556

                    unemprate.Rural  emprate.Self-Employed (%)  \
urbanization_level
High Urban                 3.738889                  40.972222
Low Urban                  2.755556                  36.627778

                    emprate.Regular Wage/Salary (%)  emprate.Casual
Labour (%)
urbanization_level

High Urban                                11.927778
52.916667
Low Urban                                 19.200000
55.822222

# Work Participation Rate (WPR) Trends:

# The WPR in urban areas is slightly lower in high urbanization states
(54.61%) than in low urbanization states (55.75%).

# Similarly, rural WPR is lower in high urbanization states (54.73%)
compared to low urbanization states (59.87%).

# This suggests that states with lower urbanization have a higher
workforce participation rate in both rural and urban areas, possibly
due to a greater reliance on labor-intensive sectors like agriculture.

# Unemployment Rate Trends:
```

```python
# The unemployment rate is higher in urban areas of highly urbanized
states (6.85%) than in less urbanized states (5.10%).

# The rural unemployment rate is also higher in highly urbanized
states (3.73%) compared to low urbanization states (2.75%).

# This indicates that employment opportunities might be more
competitive in highly urbanized states, leading to higher
unemployment.

# Employment Type Distribution:

# Self-Employment: More prevalent in high urbanization states (40.97%)
compared to low urbanization states (36.63%). This could be due to
higher entrepreneurial activity and gig economy opportunities in urban
settings.

# Regular Wage/Salary Jobs: Higher in low urbanization states (19.2%)
compared to high urbanization states (11.93%). This suggests that
structured employment is more common in states with lower
urbanization, possibly due to a stronger manufacturing or government
job sector.

# Casual Labor: Slightly higher in low urbanization states (55.82%)
compared to high urbanization states (52.92%). This indicates a
greater dependence on temporary or daily wage jobs in less urbanized
states.

# Overall Insights:
# Highly urbanized states have higher unemployment rates despite
having more self-employment.

# States with lower urbanization have a higher WPR and lower
unemployment, potentially due to stronger agricultural and informal
sector employment.

# Formal salaried jobs are more prevalent in less urbanized states,
while self-employment is more common in urbanized regions.

# Data
data = {
    "Category": ["WPR Urban", "WPR Rural", "Unemployment Urban",
"Unemployment Rural",
                 "Self-Employed", "Regular Wage/Salary", "Casual
Labour"],
    "High Urban": [54.61, 54.73, 6.85, 3.74, 40.97, 11.93, 52.92],
    "Low Urban": [55.75, 59.87, 5.10, 2.75, 36.63, 19.20, 55.82]
}

df = pd.DataFrame(data)
```

```
# Plot
plt.figure(figsize=(12, 6))
df.set_index("Category").plot(kind="bar", colormap="coolwarm",
width=0.8)
plt.title("Employment Trends: High vs. Low Urbanization States")
plt.ylabel("Percentage")
plt.xticks(rotation=45)
plt.legend(title="Urbanization Level")

# Show plot
plt.tight_layout()
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```