# The Impact of COVID-19 on Employment in India

## Introduction:

The COVID-19 pandemic had a profound impact on the global economy, affecting employment opportunities and labor markets across various sectors. This report focuses on the employment situation in India during the pandemic, analyzing key factors such as unemployment rates, labor participation rates, and employment trends across different regions and time periods.

The dataset used for this analysis contains monthly records of the unemployment rate, employment numbers, and labor force participation rates for Indian states, covering the period before and after the COVID-19 outbreak. By comparing these indicators over time and across regions, this report aims to uncover the broader implications of the pandemic on the Indian labor market.

The report provides a detailed analysis of how the pandemic influenced employment patterns, identifying trends, regional disparities, and seasonal effects on the unemployment rate. Insights gained from this study can help inform policymakers and stakeholders in understanding the long-term impact of the pandemic on employment and guide future strategies to support economic recovery.

## Data Overview:

**Region: Name of the state or geographic area in India.**

**Date: The date when the data was collected (e.g., month or day).**

**Frequency: Data collection frequency (e.g., Monthly "M").**

**Estimated Unemployment Rate (%): Percentage of people unemployed in each region.**

**Estimated Employed: Estimated number of employed individuals in the region.**

**Estimated Labour Participation Rate (%): Percentage of eligible people participating in the labor force.**

**Area: Specifies if the data pertains to a rural or urban area.**

**Region.1: Likely a duplicate or secondary version of the "Region" column.**

**Day: Day extracted from the date.**

**Month: Month extracted from the date.**

**Day Name: Name of the day (e.g., Monday, Tuesday).**

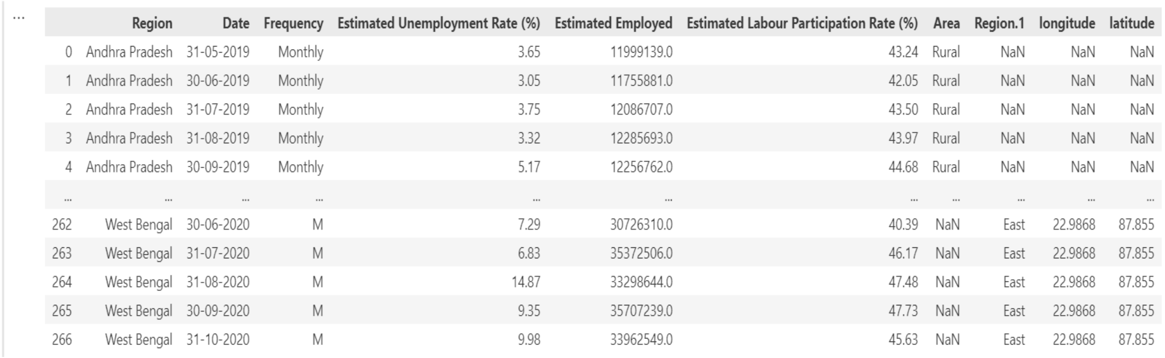
**Month Name: Name of the month (e.g., January, February).**

**Year: The year when the data was recorded.**

**Before_After_COVID: Indicates whether the data is before or after the COVID-19 pandemic.**

**Season: The season when the data was collected (e.g., Summer, Winter).**

# Background about the dataset :



|  | Region | Date | Frequency | Estimated Unemployment Rate (%) | Estimated Employed | Estimated Labour Participation Rate (%) | Area | Region.1 | longitude | latitude |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Andhra Pradesh | 31-05-2019 | Monthly | 3.65 | 11999139.0 | 43.24 | Rural | NaN | NaN | NaN |
| 1 | Andhra Pradesh | 30-06-2019 | Monthly | 3.05 | 11755881.0 | 42.05 | Rural | NaN | NaN | NaN |
| 2 | Andhra Pradesh | 31-07-2019 | Monthly | 3.75 | 12086707.0 | 43.50 | Rural | NaN | NaN | NaN |
| 3 | Andhra Pradesh | 31-08-2019 | Monthly | 3.32 | 12285693.0 | 43.97 | Rural | NaN | NaN | NaN |
| 4 | Andhra Pradesh | 30-09-2019 | Monthly | 5.17 | 12256762.0 | 44.68 | Rural | NaN | NaN | NaN |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 262 | West Bengal | 30-06-2020 | M | 7.29 | 30726310.0 | 40.39 | NaN | East | 22.9868 | 87.855 |
| 263 | West Bengal | 31-07-2020 | M | 6.83 | 35372506.0 | 46.17 | NaN | East | 22.9868 | 87.855 |
| 264 | West Bengal | 31-08-2020 | M | 14.87 | 33298644.0 | 47.48 | NaN | East | 22.9868 | 87.855 |
| 265 | West Bengal | 30-09-2020 | M | 9.35 | 35707239.0 | 47.73 | NaN | East | 22.9868 | 87.855 |
| 266 | West Bengal | 31-10-2020 | M | 9.98 | 33962549.0 | 45.63 | NaN | East | 22.9868 | 87.855 |

# Cleaning and feature engineering:

```
df.columns = df.columns.str.strip()
```

**is used to clean the column names of a DataFrame by removing any leading or trailing whitespace**

```
df["Area"] = df["Area"].fillna("not specified")
```

This code replaces any missing values in the "Area" column of the DataFrame with the string "not specified," ensuring that the dataset remains complete and ready for analysis.

```
df.dropna(inplace=True)
```

This code removes any rows with missing values from the DataFrame

```
df['Date'] = pd.to_datetime(df['Date'], format='%d-%m-%Y', errors='coerce')
```

**This code converts the "Date" column of the DataFrame to a datetime format (day-month-year)**

```
df['Day']=df['Date'].dt.day
```

**This code extracts the day of the month from the "Date" column and stores it in a new column called "Day"**

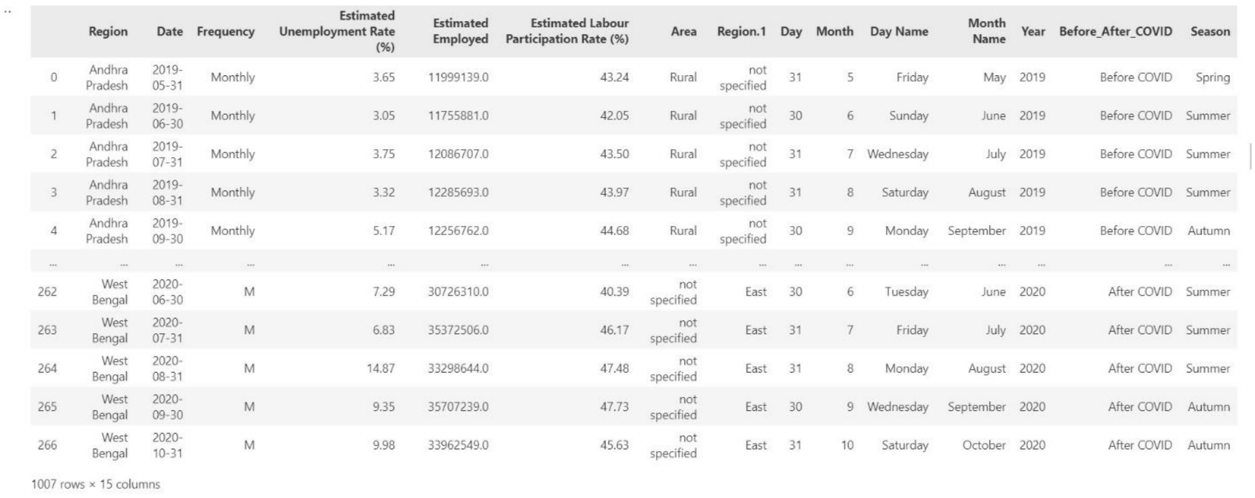
```
pandemic_start_date = pd.to_datetime('2020-03-11')

df['Before_After_COVID'] = df['Date'].apply(lambda x: 'Before COVID' if x < pandemic_start_date else 'After COVID')
```

**This code categorizes dates in the DataFrame into 'Before COVID' and 'After COVID' based on the pandemic's start date (March 11, 2020), creating a new column called "Before_After_COVID" for further analysis of trends and impacts related to the pandemic.**

```
def map_months(x):
    if x in [12, 1, 2]:
        return 'Winter'
    elif x in [3, 4, 5]:
        return 'Spring'
    elif x in [6, 7, 8]:
        return 'Summer'
    elif x in [9, 10, 11]:
        return 'Autumn'

df['Season'] = df['Month'].apply(map_months)
```

**This code categorizes the months in the dataset into their corresponding seasons (Winter, Spring, Summer, Autumn) and creates a new "Season" column, enabling seasonal trend analysis in the data.**

## The Dataset After cleaning as follows:



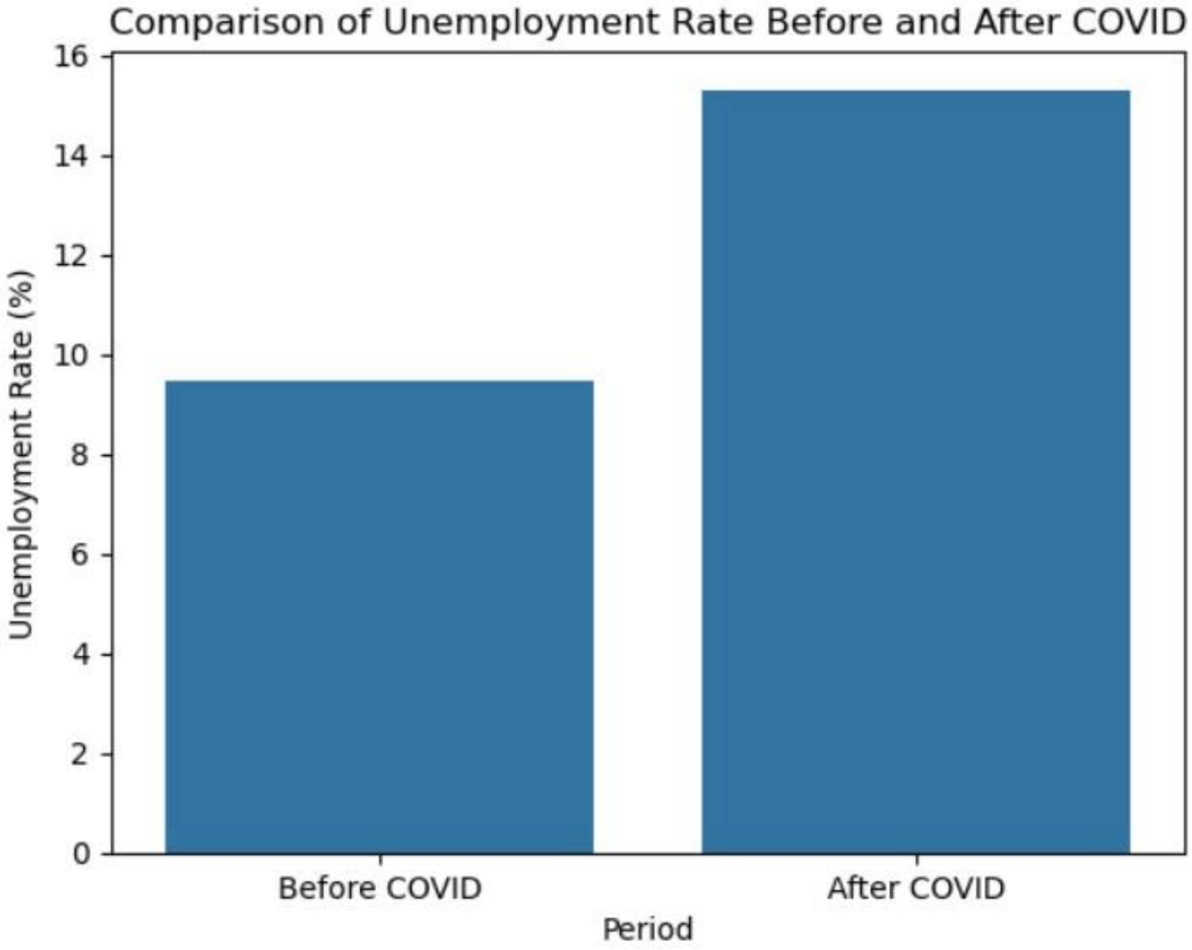
|  | Region | Date | Frequency | Estimated Unemployment Rate (%) | Estimated Employed | Estimated Labour Participation Rate (%) | Area | Region.1 | Day | Month | Day Name | Month Name | Year | Before_After_COVID | Season |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Andhra Pradesh | 2019-05-31 | Monthly | 3.65 | 11999139.0 | 43.24 | Rural | not specified | 31 | 5 | Friday | May | 2019 | Before COVID | Spring |
| 1 | Andhra Pradesh | 2019-06-30 | Monthly | 3.05 | 11755881.0 | 42.05 | Rural | not specified | 30 | 6 | Sunday | June | 2019 | Before COVID | Summer |
| 2 | Andhra Pradesh | 2019-07-31 | Monthly | 3.75 | 12086707.0 | 43.50 | Rural | not specified | 31 | 7 | Wednesday | July | 2019 | Before COVID | Summer |
| 3 | Andhra Pradesh | 2019-08-31 | Monthly | 3.32 | 12285693.0 | 43.97 | Rural | not specified | 31 | 8 | Saturday | August | 2019 | Before COVID | Summer |
| 4 | Andhra Pradesh | 2019-09-30 | Monthly | 5.17 | 12256762.0 | 44.68 | Rural | not specified | 30 | 9 | Monday | September | 2019 | Before COVID | Autumn |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 262 | West Bengal | 2020-06-30 | M | 7.29 | 30726310.0 | 40.39 | not specified | East | 30 | 6 | Tuesday | June | 2020 | After COVID | Summer |
| 263 | West Bengal | 2020-07-31 | M | 6.83 | 35372506.0 | 46.17 | not specified | East | 31 | 7 | Friday | July | 2020 | After COVID | Summer |
| 264 | West Bengal | 2020-08-31 | M | 14.87 | 33298644.0 | 47.48 | not specified | East | 31 | 8 | Monday | August | 2020 | After COVID | Summer |
| 265 | West Bengal | 2020-09-30 | M | 9.35 | 35707239.0 | 47.73 | not specified | East | 30 | 9 | Wednesday | September | 2020 | After COVID | Autumn |
| 266 | West Bengal | 2020-10-31 | M | 9.98 | 33962549.0 | 45.63 | not specified | East | 31 | 10 | Saturday | October | 2020 | After COVID | Autumn |

1007 rows × 15 columns

## Data Visualization using python:

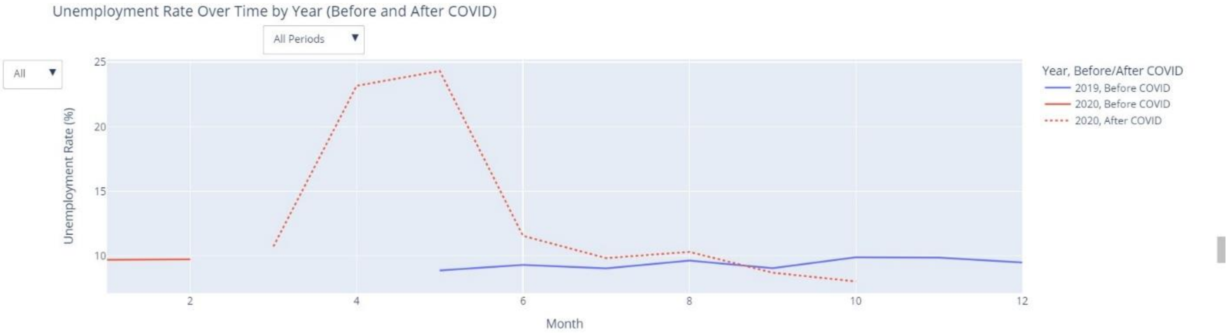
```
print(df['Before_After_COVID'].unique())

before_covid = df[df['Before_After_COVID'] == 'Before COVID']

after_covid = df[df['Before_After_COVID'] == 'After COVID']

before_avg = before_covid['Estimated Unemployment Rate (%)'].mean()

after_avg = after_covid['Estimated Unemployment Rate (%)'].mean()

print(f"Average unemployment rate before Covid:{before_avg:.2f}%")

print(f"Average unemployment rate after Covid:: {after_avg:.2f}%")

data = {

  'Period': ['Before COVID', 'After COVID'],

  'Unemployment Rate (%)': [before_avg, after_avg]

}

df_comparison = pd.DataFrame(data)

sns.barplot(x='Period', y='Unemployment Rate (%)', data=df_comparison)

plt.title('Comparison of Unemployment Rate Before and After COVID')

plt.ylabel('Unemployment Rate (%)')

plt.show()
```

**This analysis compares the average unemployment rates before and after the onset of COVID-19. The data is filtered into two periods—'Before COVID' and 'After COVID'. The average unemployment rate for each period is calculated and visualized in a bar chart, which highlights the changes in unemployment trends due to the pandemic.**



**Average unemployment rate before Covid:9.48%**

**Average unemployment rate after Covid:: 15.31%**



The chart will display lines for each year, showing the unemployment rate across the months.

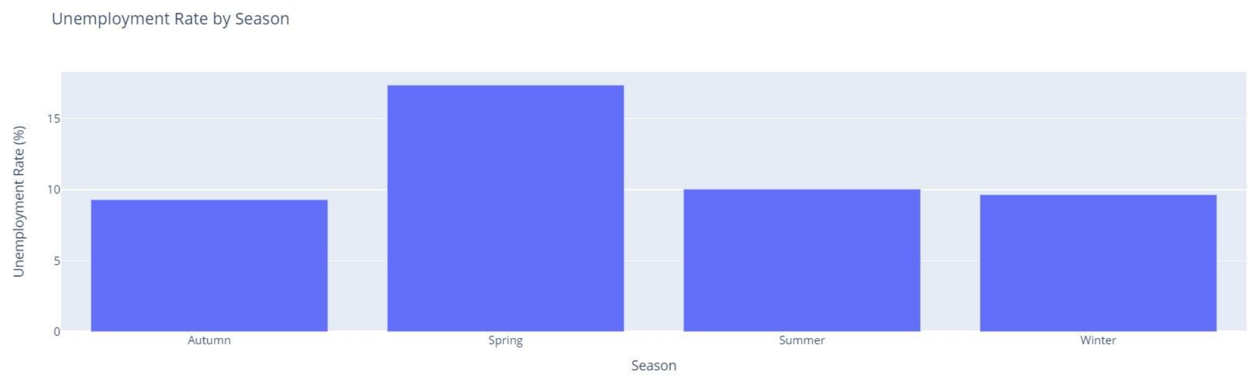
The lines are visually distinguished based on whether the data is from "Before COVID" (dashed lines) or "After COVID" (solid lines).

Two dropdown menus:

One allows to filter by year.

The other allows to switch between viewing all periods, before COVID, or after COVID.

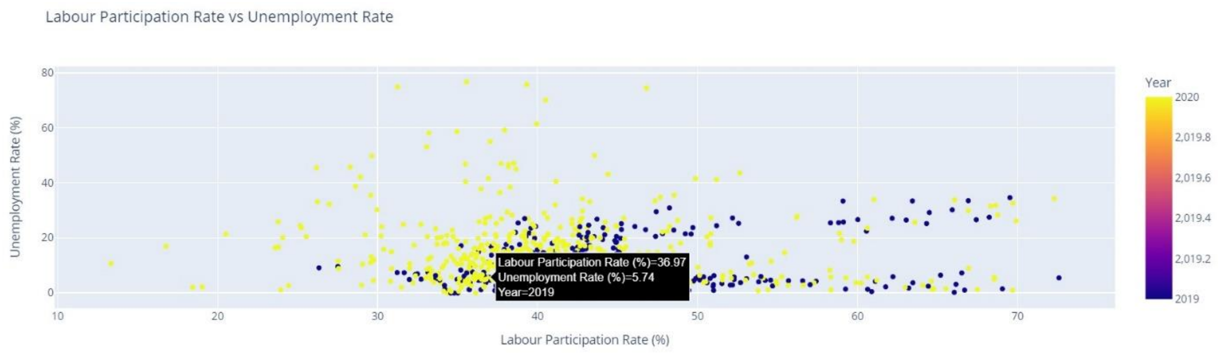
```
seasonal_unemployment = df.groupby('Season')['Estimated Unemployment Rate (%)'].mean().reset_index()

fig = px.bar(

   seasonal_unemployment,

   x='Season',

   y='Estimated Unemployment Rate (%)',

   title='Unemployment Rate by Season',

   labels={'Estimated Unemployment Rate (%)': 'Unemployment Rate (%)'}

)

fig.show()
```



**The bar chart will display four bars, each representing the average unemployment rate for a specific season (Winter, Spring, Summer, and Autumn).**

**The height of each bar corresponds to the unemployment rate, helping to compare how unemployment varies seasonally.**

```
fig = px.scatter(
    df,
    x='Estimated Labour Participation Rate (%)',
    y='Estimated Unemployment Rate (%)',
    color='Year',
    title='Labour Participation Rate vs Unemployment Rate',
    labels={
        'Estimated Labour Participation Rate (%)': 'Labour Participation Rate (%)',
        'Estimated Unemployment Rate (%)': 'Unemployment Rate (%)'
    }
)fig.show()
```

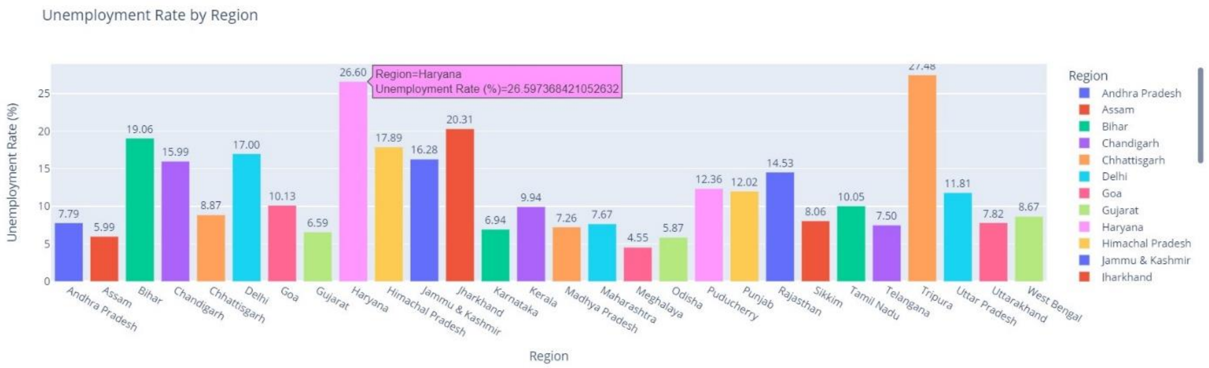


The scatter plot will display data points where each one represents a combination of labor participation rate and unemployment rate for a particular year.

The color-coding by year helps highlight trends across time.

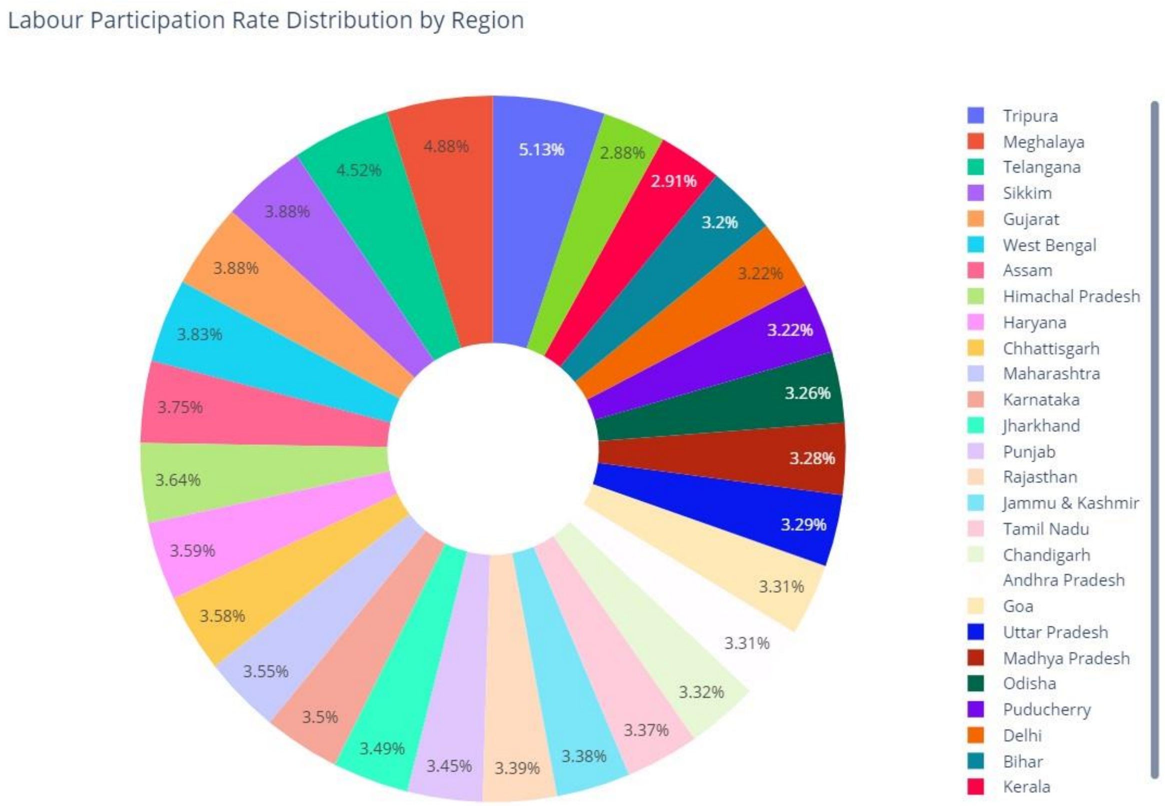
This visualization allows you to explore the relationship between the two rates—whether higher participation correlates with lower or higher unemployment—and see if there are differences in this relationship across different years.

```
unemployment_by_region = df.groupby('Region')['Estimated Unemployment Rate (%)'].mean().reset_index()

fig = px.bar(

  unemployment_by_region,

  x='Region',

  y='Estimated Unemployment Rate (%)',

  title='Unemployment Rate by Region',

  labels={'Estimated Unemployment Rate (%)': 'Unemployment Rate (%)'},

  text='Estimated Unemployment Rate (%)',

  color='Region'

)

fig.update_traces(texttemplate='%{text:.2f}', textposition='outside')

fig.update_layout(uniformtext_minsize=8, uniformtext_mode='hide')    fig.show()
```



The chart offers a clear comparison of unemployment rates across regions.

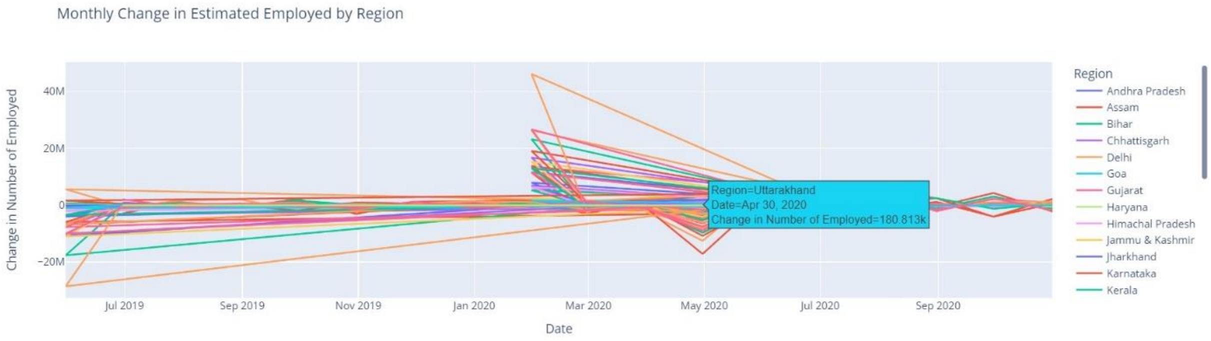
```
participation_by_region = df.groupby('Region')['Estimated Labour Participation Rate (%)'].mean().reset_index()

fig = px.pie(

   participation_by_region,

   values='Estimated Labour Participation Rate (%)',

   names='Region',

   title='Labour Participation Rate Distribution by Region',

   hole=0.3,

   width=900,

   height=700

)

fig.show()
```



This chart provides a clear view of how labor participation rates vary by region, offering a quick comparison of participation contributions across different areas.

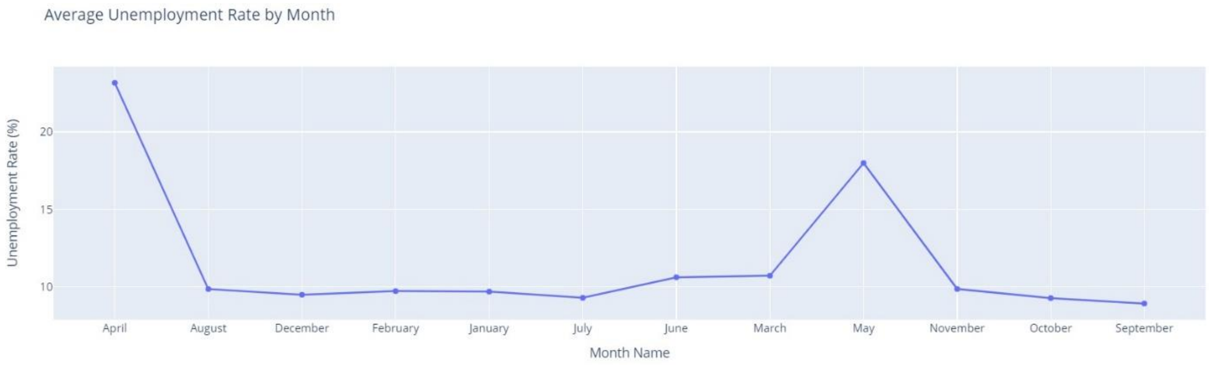
```
df['Monthly Change'] = df.groupby('Region')['Estimated Employed'].diff()

fig = px.line(

  df,

  x='Date',

  y='Monthly Change',

  color='Region',

  title='Monthly Change in Estimated Employed by Region',

  labels={'Monthly Change': 'Change in Number of Employed'},

)
```

```
fig.show()
```



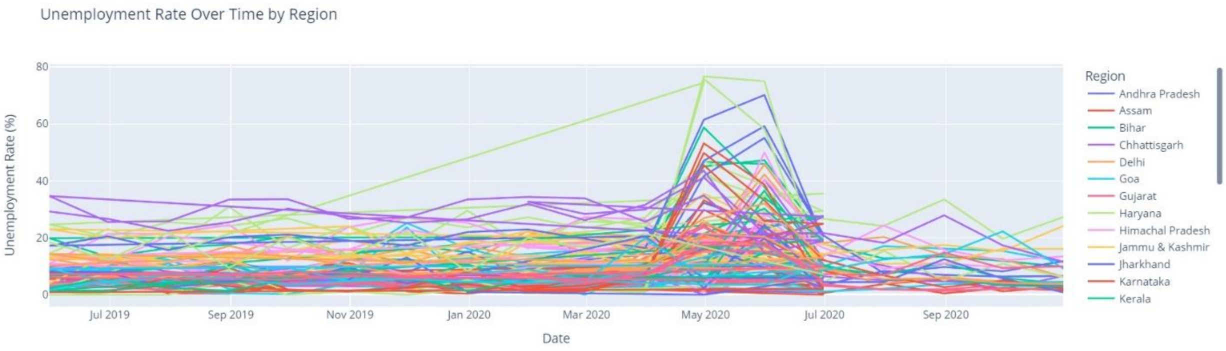
This line chart helps in understanding employment dynamics across different regions and how employment numbers fluctuate over time.

```
unemployment_by_month = df.groupby('Month Name')['Estimated Unemployment Rate (%)'].mean().reset_index()
order_of_months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
unemployment_by_month['Month Name'] = pd.Categorical(unemployment_by_month['Month Name'], categories=order_of_months, ordered=True)
fig = px.line(
    unemployment_by_month,
    x='Month Name',
    y='Estimated Unemployment Rate (%)',
    title='Average Unemployment Rate by Month',
    labels={'Estimated Unemployment Rate (%)': 'Unemployment Rate (%)'}
)

fig.update_traces(mode='markers+lines')
fig.show()
```



This chart is useful for identifying monthly patterns or fluctuations in the unemployment rate throughout the year.

```
fig = px.line(

   df,

   x='Date',

   y='Estimated Unemployment Rate (%)',

   color='Region',

   title='Unemployment Rate Over Time by Region',

   labels={'Estimated Unemployment Rate (%)': 'Unemployment Rate (%)'}

)


fig.show()
```



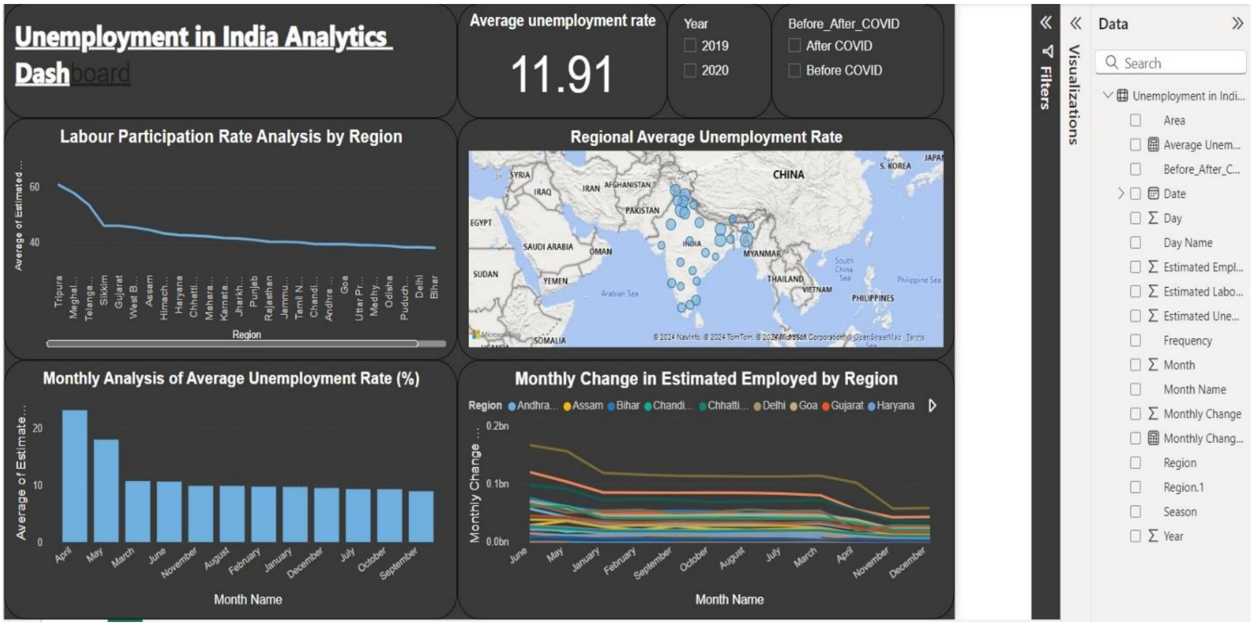
This chart is helpful in understanding how the unemployment rate has changed over time for different regions and to compare trends between regions.

## Data Visualization using Power BI:

### Use DAX Function

```
Average Unemployment Rate by Region =
AVERAGE('Unemployment in India Full'[Estimated Unemployment Rate (%)])
```

```
Monthly Change in Estimated Employed =
  SUM('Unemployment in India Full'[Estimated Employed]) -
  CALCULATE(
    SUM('Unemployment in India Full'[Estimated Employed]),
    PREVIOUSMONTH('Unemployment in India Full'[Date])
  )
```



**Unemployment in India Analytics Dashboard**

Average unemployment rate

11.91

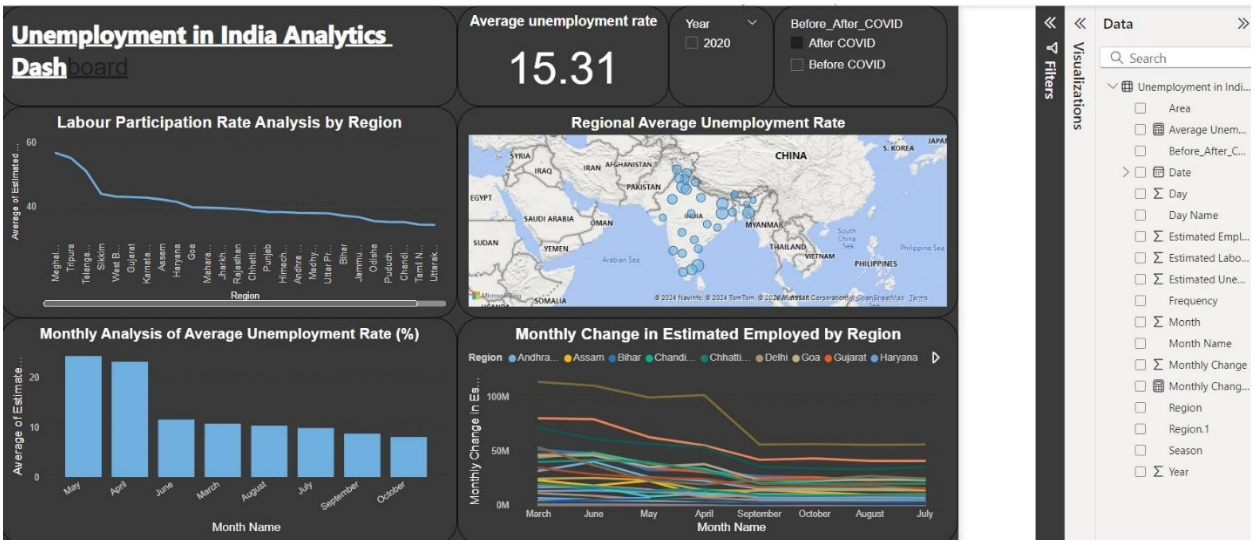
Year
□ 2019
□ 2020

Before_After_COVID
□ After COVID
□ Before COVID

Data

Search

- Unemployment in Indi...
  - □ Area
  - □ Average Unem...
  - □ Before_After_C...
  - □ Date
  - □ Σ Day
  - □ Day Name
  - □ Σ Estimated Empl...
  - □ Σ Estimated Labo...
  - □ Σ Estimated Une...
  - □ Frequency
  - □ Σ Month
  - □ Month Name
  - □ Σ Monthly Change
  - □ Monthly Chang...
  - □ Region
  - □ Region.1
  - □ Season
  - □ Σ Year



**Unemployment in India Analytics Dashboard**

Average unemployment rate

15.31

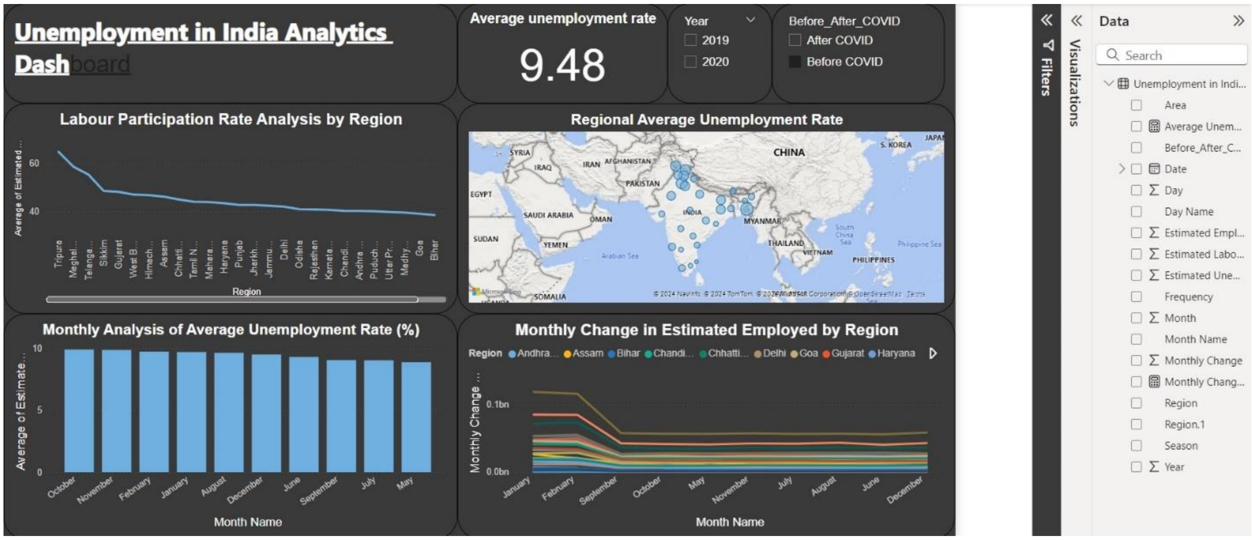
Year
□ 2020

Before_After_COVID
■ After COVID
□ Before COVID

Data

Search

- Unemployment in Indi...
  - □ Area
  - □ Average Unem...
  - □ Before_After_C...
  - □ Date
  - □ Σ Day
  - □ Day Name
  - □ Σ Estimated Empl...
  - □ Σ Estimated Labo...
  - □ Σ Estimated Une...
  - □ Frequency
  - □ Σ Month
  - □ Month Name
  - □ Σ Monthly Change
  - □ Monthly Chang...
  - □ Region
  - □ Region.1
  - □ Season
  - □ Σ Year



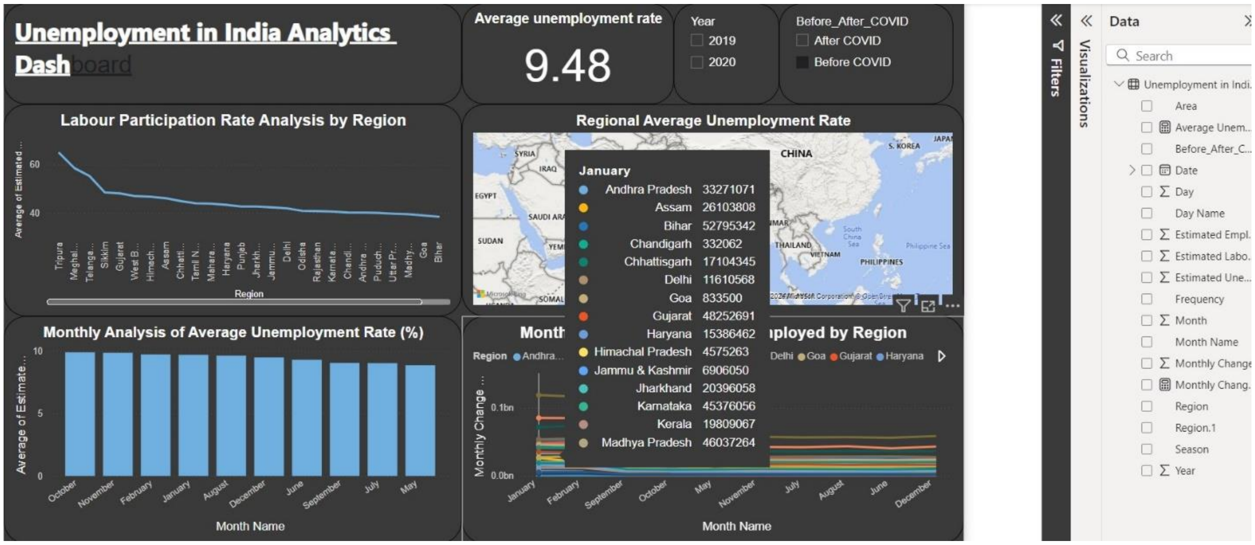
**Unemployment in India Analytics Dashboard**

Average unemployment rate

9.48

Year
☐ 2019
☐ 2020

Before_After_COVID
☐ After COVID
■ Before COVID



**Unemployment in India Analytics Dashboard**

Average unemployment rate

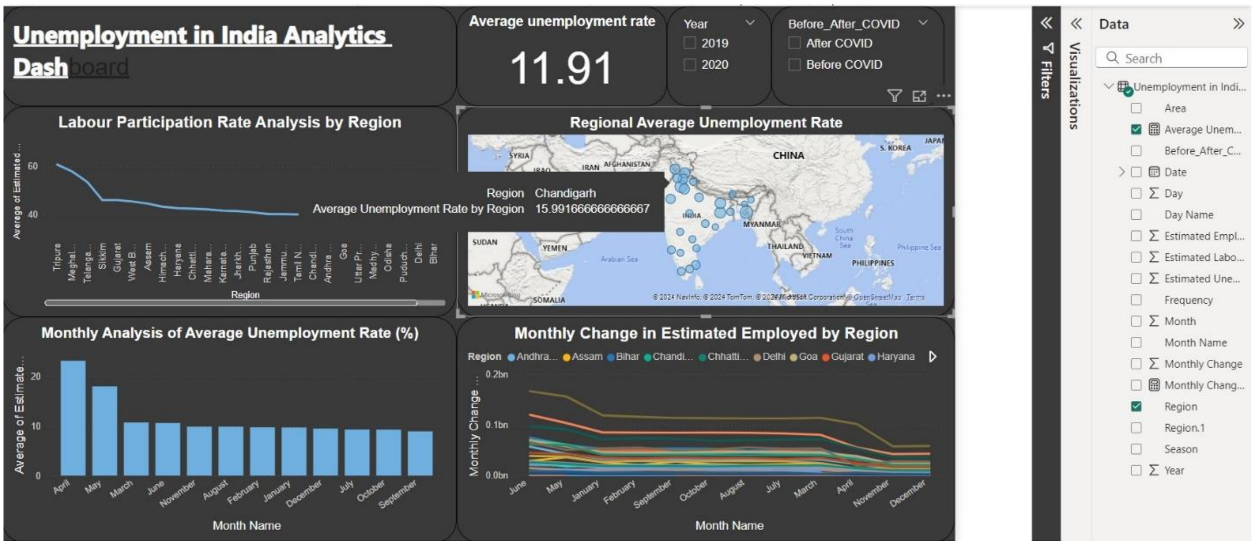
9.48

Year
☐ 2019
☐ 2020

Before_After_COVID
☐ After COVID
■ Before COVID

January

| | |
|---|---|
| Andhra Pradesh | 33271071 |
| Assam | 26103808 |
| Bihar | 52795342 |
| Chandigarh | 332062 |
| Chhattisgarh | 17104345 |
| Delhi | 11610568 |
| Goa | 833500 |
| Gujarat | 48252691 |
| Haryana | 15386462 |
| Himachal Pradesh | 4575263 |
| Jammu & Kashmir | 6906050 |
| Jharkhand | 20396058 |
| Karnataka | 45376056 |
| Kerala | 19809067 |
| Madhya Pradesh | 46037264 |



**Unemployment in India Analytics Dashboard**

Average unemployment rate

11.91

Year
☐ 2019
☐ 2020

Before_After_COVID
☐ After COVID
☐ Before COVID

Region Chandigarh
Average Unemployment Rate by Region 15.991666666666667

## Recommendations:

**Post-COVID-19 recovery: Given that unemployment was higher during COVID-19, regions that have not yet fully recovered should receive additional support to boost their local economies and labor markets.**

**Seasonal Employment Interventions: The analysis revealed that unemployment rates fluctuate significantly across different seasons, with spring showing the highest rates. To address these seasonal challenges, regions experiencing elevated unemployment during this period could benefit from targeted temporary employment programs or initiatives. These programs could aim to bridge gaps in seasonal employment, providing support to those affected and enhancing overall workforce stability. Implementing such measures may not only alleviate unemployment during peak periods but also promote a more resilient labor market year-round.**

**Ongoing Monitoring: Regular tracking of unemployment and labor participation rates by month and region can provide further insights, allowing for proactive adjustments in policy and interventions to address emerging trends.**