```
from google.colab import files

# Dosya yükle
uploaded = files.upload()
```

Choose Files 6202001.xlsx

- **6202001.xlsx**(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 749278 bytes, last modified: 2/18/2025 - 100% done

```
import pandas as pd

# Define the file path
file_path = "6202001.xlsx"

# Load the Excel file
xls = pd.ExcelFile(file_path)

# Display the sheet names
xls.sheet_names
```

```
['Index', 'Data1', 'Enquiries']
```

```
# Read the first few rows from each sheet
index_df = pd.read_excel(xls, sheet_name="Index")
data1_df = pd.read_excel(xls, sheet_name="Data1")
enquiries_df = pd.read_excel(xls, sheet_name="Enquiries")

# Display the first five rows of each sheet
index_df.head(), data1_df.head(), enquiries_df.head()
```

```
 [5 rows x 115 columns],
    Unnamed: 0                                    Unnamed: 1
 0         NaN                           Time Series Workbook
 1         NaN                                            NaN
 2         NaN                                            NaN
 3         NaN                   6202.0 Labour Force, Australia
 4         NaN  Table 1. Labour force status by Sex, Australia...)
```

Index Sheet, Contains description and titles Data1 Shee,Includes metrics like employed persons, unemployment rate, and participation rate. Data is presented in different categories: "Trend", "Seasonally Adjusted", "Original". Enquiries Sheet,Probably contains data sources and explanations. May not be directly useful for analysis.

```
# Display the first 10 rows of the Data1 sheet
data1_df.head(10)
```

	Unnamed: 0	Employed total ; Persons ;	Employed total ; Persons ;.1	Employed total ; Persons ;.2	Employed total ; > Males ;	Employed total ; > Males ;.1	Employed total ; > Males ;.2	Employed total ; > Females ;	Employed total ; > Females ;.1	Employed total ; > Females ;.2	...	P
0	Unit	000	000	000	000	000	000	000	000	000	...	
1	Series Type	Trend	Seasonally Adjusted	Original	Trend	Seasonally Adjusted	Original	Trend	Seasonally Adjusted	Original	...	
2	Data Type	STOCK	STOCK	STOCK	STOCK	STOCK	STOCK	STOCK	STOCK	STOCK	...	
3	Frequency	Month	Month	Month	Month	Month	Month	Month	Month	Month	...	
4	Collection Month	1	1	1	1	1	1	1	1	1	...	
5	Series Start	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	...	
6	Series End	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	...	
7	No. Obs	563	563	563	563	563	563	563	563	563	...	
8	Series ID	A84423127L	A84423043C	A84423085A	A84423113X	A84423029J	A84423071L	A84423141J	A84423057T	A84423099R	...	
9	1978-02-01 00:00:00	6008.013099	5997.549872	5985.659716	3889.313983	3881.109135	3887.387982	2118.699117	2116.440737	2098.271734	...	

10 rows × 115 columns

It has 115 columns and contains various labor force statistics. The "Series Start" column shows data starting from 1978. The "Series End" column extends to 2024. The dataset includes three data types: Trend, Seasonally Adjusted, and Original.

```
# Display all column names in the dataset
data1_df.columns.tolist()
```

```
 '> Unemployment rate looked for only part-time work ;  Persons ;.1',
 '> Unemployment rate looked for only part-time work ;  Persons ;.2',
 '> Unemployment rate looked for only part-time work ;  > Males ;',
 '> Unemployment rate looked for only part-time work ;  > Males ;.1',
 '> Unemployment rate looked for only part-time work ;  > Males ;.2',
 '> Unemployment rate looked for only part-time work ;  > Females ;',
 '> Unemployment rate looked for only part-time work ;  > Females ;.1',
 '> Unemployment rate looked for only part-time work ;  > Females ;.2',
 'Labour force total ;  Persons ;',
 'Labour force total ;  Persons ;.1',
 'Labour force total ;  Persons ;.2',
 'Labour force total ;  > Males ;',
 'Labour force total ;  > Males ;.1',
 'Labour force total ;  > Males ;.2',
 'Labour force total ;  > Females ;',
 'Labour force total ;  > Females ;.1',
 'Labour force total ;  > Females ;.2',
 'Participation rate ;  Persons ;',
 'Participation rate ;  Persons ;.1',
 'Participation rate ;  Persons ;.2',
 'Participation rate ;  > Males ;',
 'Participation rate ;  > Males ;.1',
 'Participation rate ;  > Males ;.2',
 'Participation rate ;  > Females ;',
 'Participation rate ;  > Females ;.1',
 'Participation rate ;  > Females ;.2',
 'Not in the labour force (NILF) ;  Persons ;',
 'Not in the labour force (NILF) ;  > Males ;',
 'Not in the labour force (NILF) ;  > Females ;',
 'Civilian population aged 15 years and over ;  Persons ;',
 'Civilian population aged 15 years and over ;  > Males ;',
 'Civilian population aged 15 years and over ;  > Females ;']
```

```
# Select only the columns related to unemployment rate
unemployment_columns = [col for col in data1_df.columns if "Unemployment rate" in col]

# Create a new DataFrame with selected columns
unemployment_df = data1_df[unemployment_columns]

# Display the first 10 rows of the unemployment data
unemployment_df.head(10)
```

	Unemployment rate ; Persons ;	Unemployment rate ; Persons ;.1	Unemployment rate ; Persons ;.2	Unemployment rate ; > Males ;	Unemployment rate ; > Males ;.1	Unemployment rate ; > Males ;.2	Unemployment rate ; > Females ;	Unemployment rate ; > Females ;.1	Unemployment rate ; > Females ;.2	Une ra ti
0	Percent	Percent	Percent	Percent	Percent	Percent	Percent	Percent	Percent	
1	Trend	Seasonally Adjusted	Original	Trend	Seasonally Adjusted	Original	Trend	Seasonally Adjusted	Original	
2	PERCENT	PERCENT	PERCENT	PERCENT	PERCENT	PERCENT	PERCENT	PERCENT	PERCENT	
3	Month	Month	Month	Month	Month	Month	Month	Month	Month	
4	1	1	1	1	1	1	1	1	1	
5	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00	1
6	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00	2
7	563	563	563	563	563	563	563	563	563	
8	A84423134K	A84423050A	A84423092X	A84423120W	A84423036F	A84423078C	A84423148X	A84423064R	A84423106A	A8
9	6.303158	6.643856	7.46415	5.356839	5.62557	6.188083	7.991956	8.455191	9.738792	

10 rows × 27 columns

Double-click (or enter) to edit

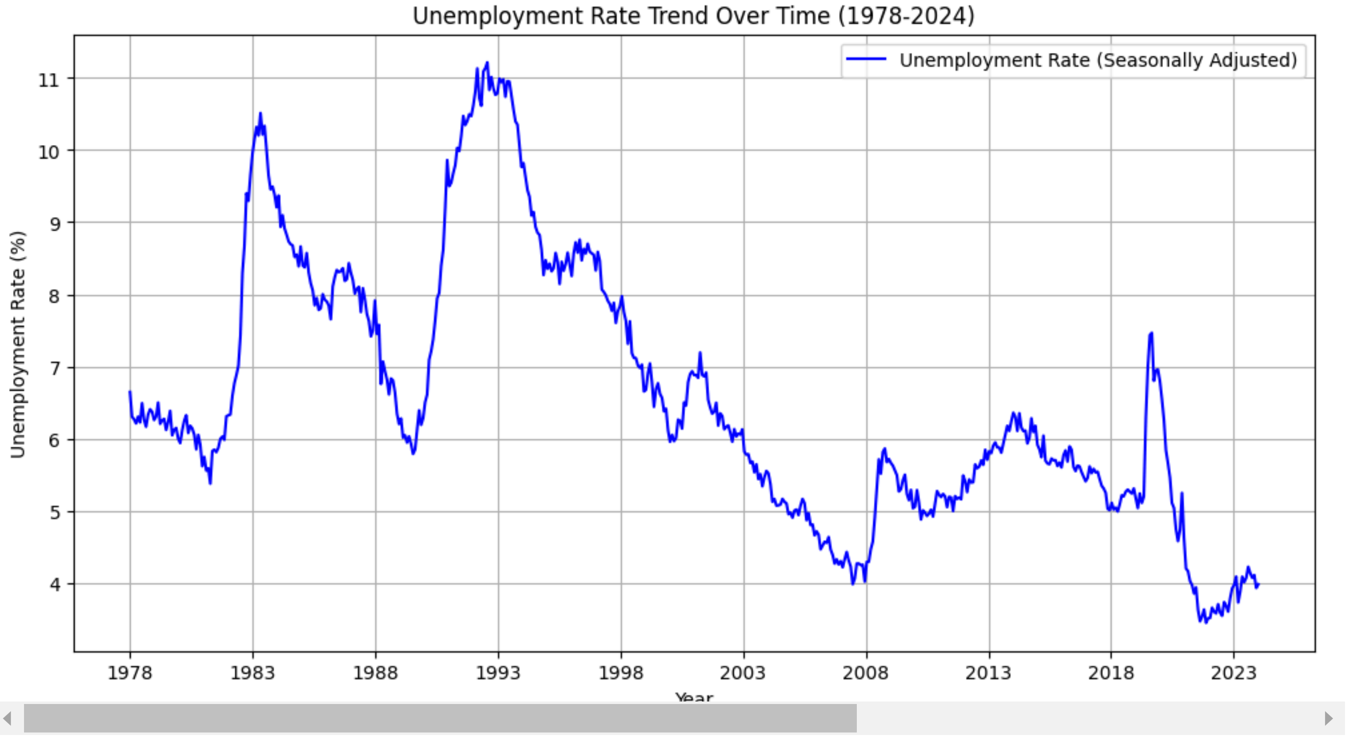
```
# Display all column names in the dataset to find the correct unemployment rate column
print(data1_df.columns.tolist())
```

```
['Unnamed: 0', 'Employed total ;  Persons ;', 'Employed total ;  Persons ;.1', 'Employed total ;  Persons ;.2', 'Employed total ;  >
```

```
# Display the first few rows of Unemployment Rate columns
unemployment_cols = ["Unemployment rate ;  Persons ;", "Unemployment rate ;  Persons ;.1", "Unemployment rate ;  Persons ;.2"]
unemployment_cols = ["Unemployment rate ;  Persons ;", "Unemployment rate ;  Persons ;.1", "Unemployment rate ;  Persons ;.2"]
data1_df[unemployment_cols].head(10)
```

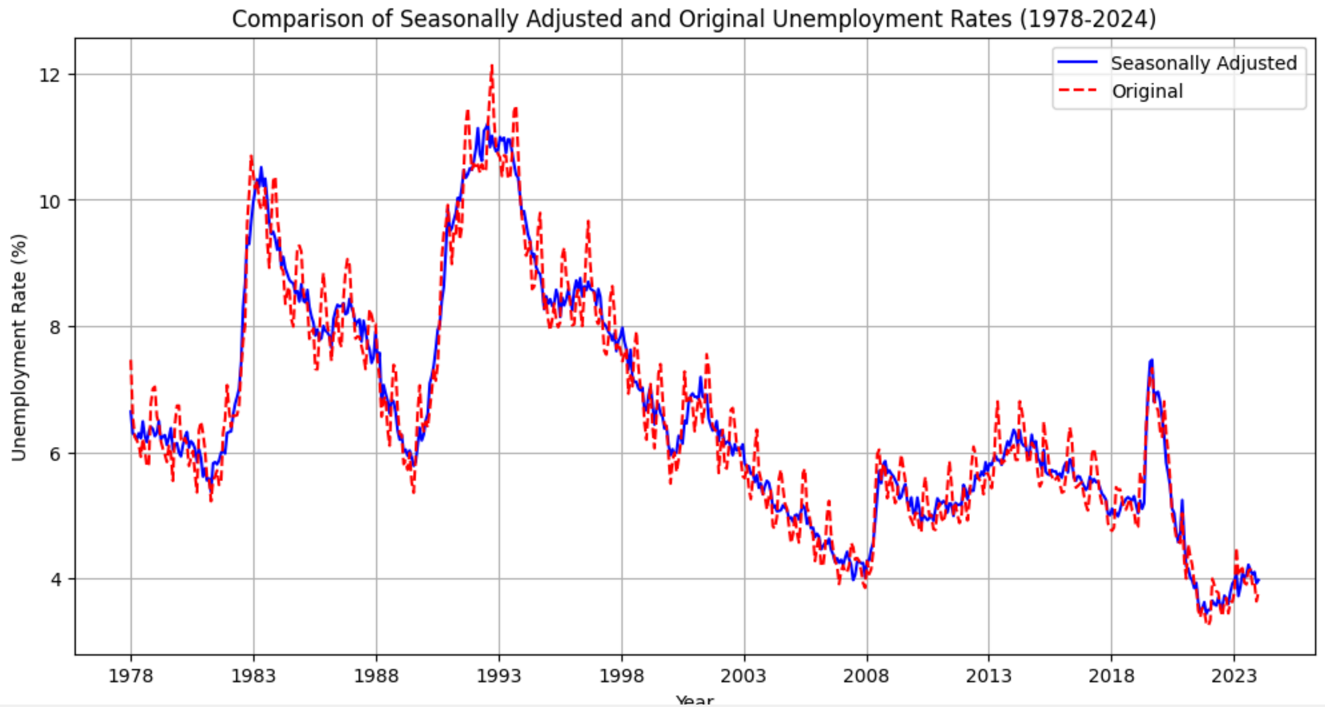
	Unemployment rate ; Persons ;	Unemployment rate ; Persons ;.1	Unemployment rate ; Persons ;.2
0	Percent	Percent	Percent
1	Trend	Seasonally Adjusted	Original
2	PERCENT	PERCENT	PERCENT
3	Month	Month	Month
4	1	1	1
5	1978-02-01 00:00:00	1978-02-01 00:00:00	1978-02-01 00:00:00
6	2024-12-01 00:00:00	2024-12-01 00:00:00	2024-12-01 00:00:00
7	563	563	563
8	A84423134K	A84423050A	A84423092X
9	6.303158	6.643856	7.46415

```
import matplotlib.pyplot as plt
import numpy as np

# Select the correct unemployment rate column
selected_column = "Unemployment rate ;  Persons ;.1"

# Extract time series data and remove non-numeric rows
unemployment_rate = data1_df[selected_column].iloc[9:].astype(float).reset_index(drop=True)

# Generate year labels (assuming monthly data from 1978 to 2024)
start_year = 1978
end_year = 2024
months = len(unemployment_rate)
years = np.linspace(start_year, end_year, months)

# Plot the unemployment rate over time with years on the x-axis
plt.figure(figsize=(12,6))
plt.plot(years, unemployment_rate, label="Unemployment Rate (Seasonally Adjusted)", color="blue")

# Add labels and title
plt.xlabel("Year")
plt.ylabel("Unemployment Rate (%)")
plt.title("Unemployment Rate Trend Over Time (1978-2024)")
plt.xticks(np.arange(start_year, end_year+1, step=5))  # Show every 5 years
plt.legend()
plt.grid(True)

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



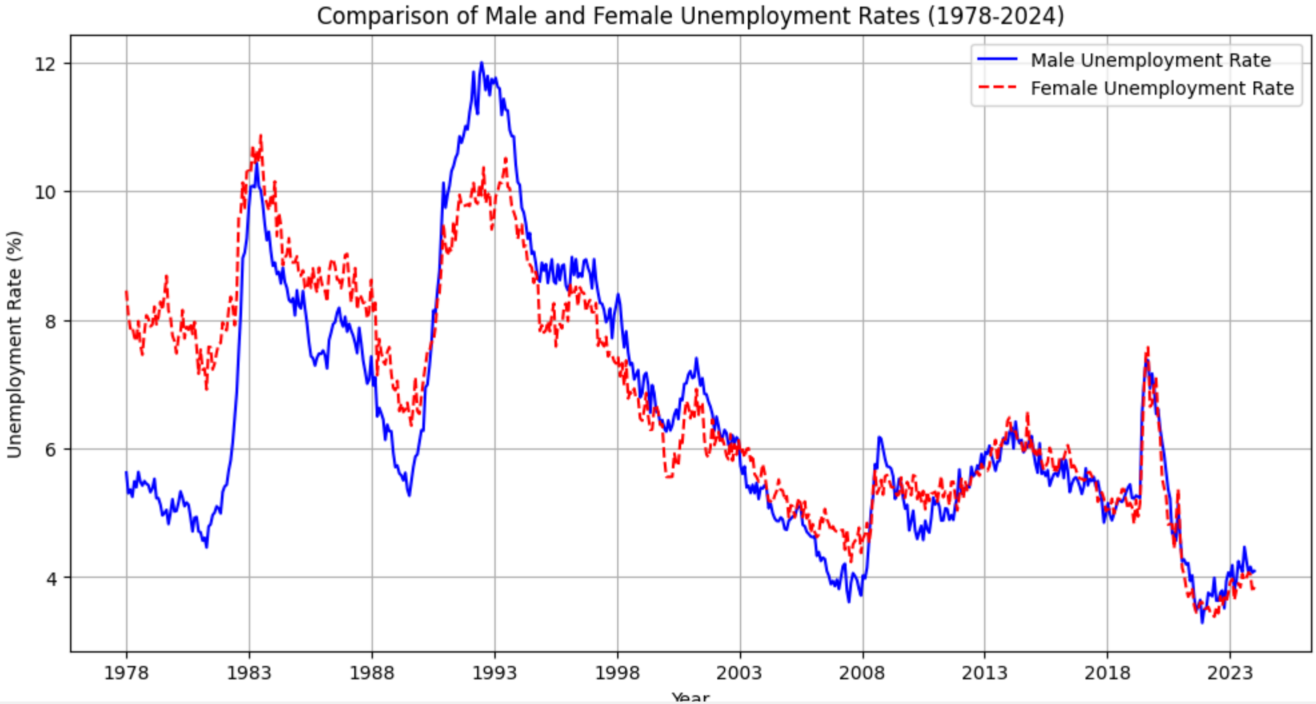
There is a significant unemployment spike in the early 1980s, possibly due to a global recession. Unemployment rose again in the early 1990s, likely due to economic downturns in Australia. A temporary spike in unemployment is observed after the 2008 Global Financial Crisis, but it recovered. The COVID-19 pandemic caused a sharp increase in unemployment, followed by a quick decline. As we approach 2024, the unemployment rate has dropped to around 4-5%.

```
# Define the correct column names
seasonally_adjusted_col = "Unemployment rate ;  Persons ;.1"  # Seasonally Adjusted
original_col = "Unemployment rate ;  Persons ;.2"  # Original

# Extract and clean the data
seasonally_adjusted = data1_df[seasonally_adjusted_col].iloc[9:].astype(float).reset_index(drop=True)
original = data1_df[original_col].iloc[9:].astype(float).reset_index(drop=True)

# Generate year labels
import numpy as np
start_year = 1978
end_year = 2024
months = len(seasonally_adjusted)
years = np.linspace(start_year, end_year, months)

# Plot both Seasonally Adjusted and Original Unemployment Rates
import matplotlib.pyplot as plt

plt.figure(figsize=(12,6))
plt.plot(years, seasonally_adjusted, label="Seasonally Adjusted", color="blue")
plt.plot(years, original, label="Original", color="red", linestyle="dashed")

# Add labels and title
plt.xlabel("Year")
plt.ylabel("Unemployment Rate (%)")
plt.title("Comparison of Seasonally Adjusted and Original Unemployment Rates (1978-2024)")
plt.xticks(np.arange(start_year, end_year+1, step=5))
plt.legend()
plt.grid(True)

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



The original data shows more fluctuations. Seasonally adjusted data provides a clearer view of long-term trends. In 1983, 1992, 2008, and 2020, the red dashed line (original) has significant spikes. This indicates that seasonal effects were more pronounced in these years. 📌 What events can reflect seasonal effects? Seasonal hiring in agriculture and tourism sectors Temporary employment spikes during Christmas and holiday seasons Students entering the job market during summer vacations

```
import matplotlib.pyplot as plt
import numpy as np

# Define column names for male and female unemployment rates (Seasonally Adjusted)
male_unemployment_col = "Unemployment rate ;  > Males ;.1"
female_unemployment_col = "Unemployment rate ;  > Females ;.1"

# Extract and clean the data
male_unemployment = data1_df[male_unemployment_col].iloc[9:].astype(float).reset_index(drop=True)
female_unemployment = data1_df[female_unemployment_col].iloc[9:].astype(float).reset_index(drop=True)

# Generate year labels
start_year = 1978
end_year = 2024
months = len(male_unemployment)
years = np.linspace(start_year, end_year, months)

# Plot Male and Female Unemployment Rates Over Time
plt.figure(figsize=(12,6))
plt.plot(years, male_unemployment, label="Male Unemployment Rate", color="blue")
plt.plot(years, female_unemployment, label="Female Unemployment Rate", color="red", linestyle="dashed")

# Add labels and title
plt.xlabel("Year")
plt.ylabel("Unemployment Rate (%)")
plt.title("Comparison of Male and Female Unemployment Rates (1978-2024)")
plt.xticks(np.arange(start_year, end_year+1, step=5))
plt.legend()
plt.grid(True)

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

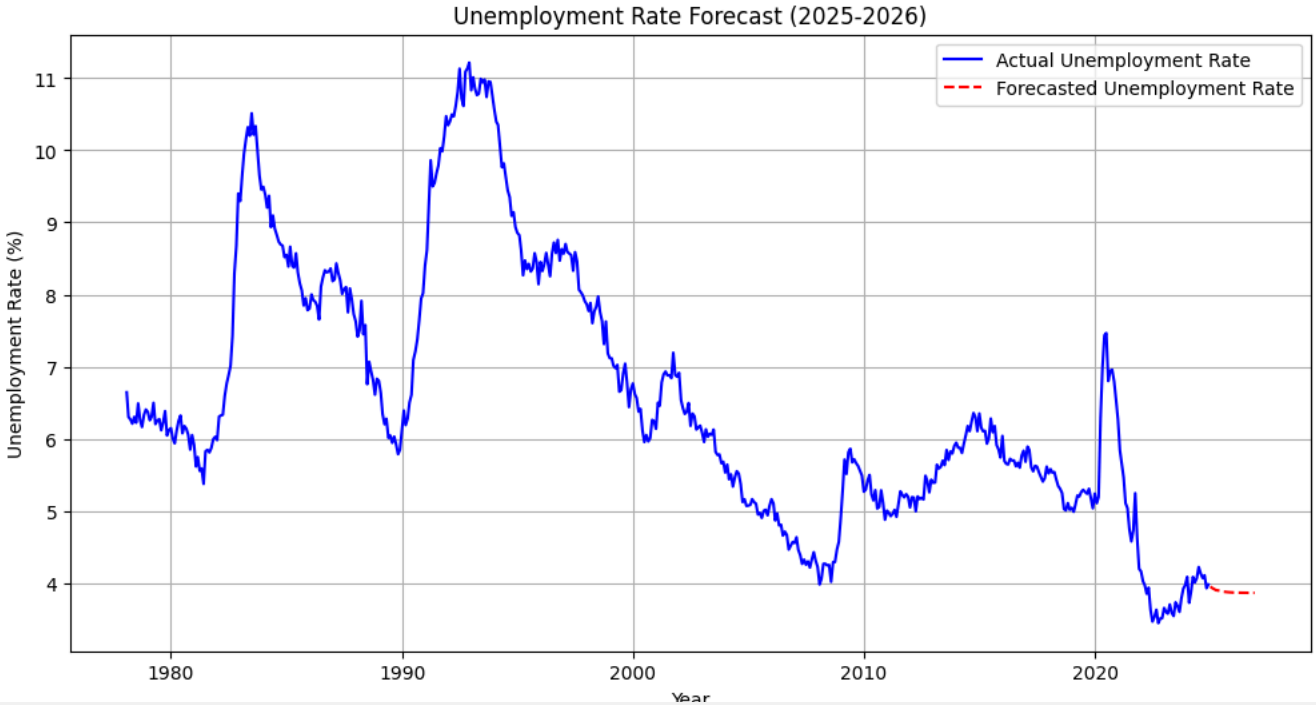


Double-click (or enter) to edit

In the early 1980s, female unemployment was around 8%, while male unemployment was around 5-6%. Between 1993 and 2010, unemployment rates for men and women became closer, and overall unemployment declined. After 2018, the female unemployment rate fluctuated more than the male rate, especially during the COVID-19 pandemic (2020).

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA

# Define the correct unemployment rate column (Seasonally Adjusted)
unemployment_col = "Unemployment rate ;  Persons ;.1"

# Extract and clean the data
unemployment_rate = data1_df[unemployment_col].iloc[9:].astype(float).reset_index(drop=True)

# Generate year labels
start_year = 1978
end_year = 2024
months = len(unemployment_rate)
years = np.linspace(start_year, end_year, months)

# Convert to Pandas Series with datetime index
date_range = pd.date_range(start=f"{start_year}-01", periods=months, freq="M")
unemployment_series = pd.Series(unemployment_rate.values, index=date_range)

# Fit ARIMA model
model = ARIMA(unemployment_series, order=(5,1,0))  # (p,d,q) values
model_fit = model.fit()

# Forecast the next 24 months (2 years)
forecast_steps = 24
forecast = model_fit.forecast(steps=forecast_steps)

# Create future date range for the forecast
future_dates = pd.date_range(start=unemployment_series.index[-1], periods=forecast_steps+1, freq="M")[1:]

# Plot original data and forecast
plt.figure(figsize=(12,6))
plt.plot(unemployment_series.index, unemployment_series, label="Actual Unemployment Rate", color="blue")
plt.plot(future_dates, forecast, label="Forecasted Unemployment Rate", color="red", linestyle="dashed")

# Add labels and title
plt.xlabel("Year")
plt.ylabel("Unemployment Rate (%)")
plt.title("Unemployment Rate Forecast (2025-2026)")
plt.legend()
plt.grid(True)
plt.grid(True)

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

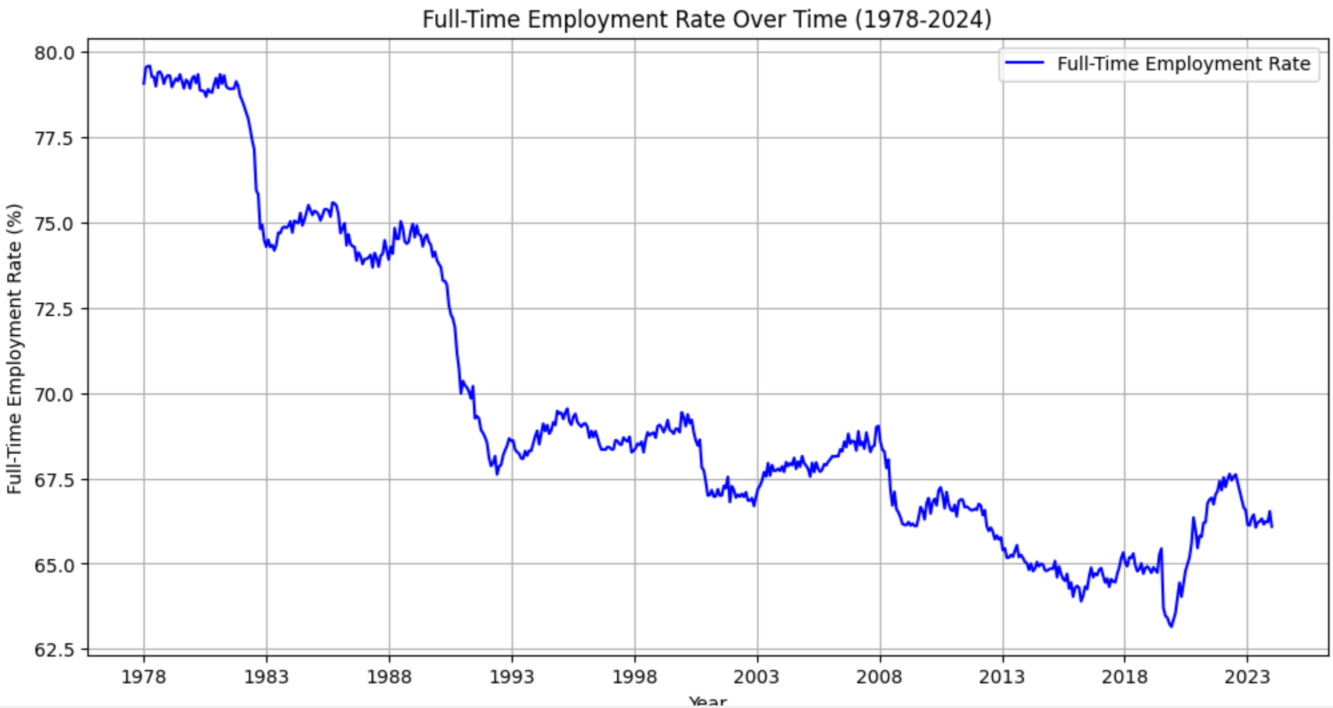
```
<ipython-input-51-6e0e7a254a92>:20: FutureWarning: 'M' is deprecated and will be removed in a future version, please use 'ME' instea
  date_range = pd.date_range(start=f"{start_year}-01", periods=months, freq="M")
<ipython-input-51-6e0e7a254a92>:32: FutureWarning: 'M' is deprecated and will be removed in a future version, please use 'ME' instea
  future_dates = pd.date_range(start=unemployment_series.index[-1], periods=forecast_steps+1, freq="M")[1:]
```



```
# Display all column names
data1_df.columns.tolist()
```

```
 'Labour force total ;  > Females ;.2',
 'Participation rate ;  Persons ;',
 'Participation rate ;  Persons ;.1',
 'Participation rate ;  Persons ;.2',
 'Participation rate ;  > Males ;',
 'Participation rate ;  > Males ;.1',
 'Participation rate ;  > Males ;.2',
 'Participation rate ;  > Females ;',
 'Participation rate ;  > Females ;.1',
 'Participation rate ;  > Females ;.2',
 'Not in the labour force (NILF) ;  Persons ;',
 'Not in the labour force (NILF) ;  > Males ;',
 'Not in the labour force (NILF) ;  > Females ;',
 'Civilian population aged 15 years and over ;  Persons ;',
 'Civilian population aged 15 years and over ;  > Males ;',
 'Civilian population aged 15 years and over ;  > Females ;']
```

```
import matplotlib.pyplot as plt
import numpy as np

# Define column names for full-time employment and total labour force
full_time_col = "> Employed full-time ;  Persons ;.1"
labour_force_col = "Labour force total ;  Persons ;.1"

# Extract and clean the data
full_time_employed = data1_df[full_time_col].iloc[9:].astype(float).reset_index(drop=True)
labour_force_total = data1_df[labour_force_col].iloc[9:].astype(float).reset_index(drop=True)

# Calculate full-time employment rate
full_time_employment_rate = (full_time_employed / labour_force_total) * 100

# Generate year labels
start_year = 1978
end_year = 2024
months = len(full_time_employment_rate)
years = np.linspace(start_year, end_year, months)

# Plot Full-Time Employment Rate Over Time
plt.figure(figsize=(12,6))
plt.plot(years, full_time_employment_rate, label="Full-Time Employment Rate", color="blue")

# Add labels and title
plt.xlabel("Year")
plt.ylabel("Full-Time Employment Rate (%)")
plt.title("Full-Time Employment Rate Over Time (1978-2024)")
plt.xticks(np.arange(start_year, end_year+1, step=5))
plt.legend()
plt.grid(True)

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



1978-1982:The full-time employment rate was around 79%, which was quite high. 1982-1993:There was a sharp decline in 1982, bringing the rate down to 75%. Between 1988-1993, it dropped further to around 70%. 1993-2008:The rate fluctuated between 67%-70%, showing a gradual downward trend. 2008-2018:The 2008 Global Financial Crisis accelerated the decline. The rate dropped to around 65%. 2018-2024:During the 2020 COVID-19 pandemic, it reached its lowest point (~63%). Since 2021, it has been recovering, and now it is around 67%.

```
# Filter unemployment rate data
unemployment_rate = data1_df[data1_df["Data Item Description"].str.contains("Unemployment rate", na=False)]

# Display the first few rows to confirm selection
print(unemployment_rate.head(10))
```

```
0                               Data Item Description  NaN  NaN  \
63                     Unemployment rate ;  Persons ;  NaN  NaN
```