Recession Analysis using Python

"In economics, a 'recession' denotes a significant downturn in economic activity lasting for two consecutive quarters." During a recession, there is a notable decline in the overall economic output, characterized by reduced consumer spending, business investment, and employment levels. This decline in economic activity often leads to a chain reaction of negative effects, including decreased income, lower corporate profits, and financial instability.

Components of Recession Analysis: Recession analysis involves examining key economic indicators to understand the underlying factors contributing to economic downturns:

1. **Growth in GDP:** Gross Domestic Product (GDP) serves as a comprehensive measure of a country's economic performance. It represents the total monetary value of all goods and services produced within the nation's borders over a specific period. Analyzing GDP growth provides insights into the overall health and trajectory of the economy.

2. **Unemployment Rate:** The unemployment rate measures the percentage of the labor force that is actively seeking employment but remains unemployed. High unemployment rates are often indicative of reduced consumer confidence, decreased consumer spending, and overall economic distress.

3. **Consumer Spending Rate:** Consumer spending is a critical component of economic activity, representing the expenditure on goods and services by households. Changes in consumer spending patterns can signal shifts in consumer confidence, economic sentiment, and overall economic conditions.

Measuring Recession: "While several indicators contribute to recession analysis, the most common approach involves analyzing the 'monthly GDP growth data'." A recession is typically identified by a sustained period of negative GDP growth, often lasting for two consecutive quarters or more. However, other factors such as rising unemployment rates, declining consumer spending, and decreased business investment may also indicate the onset of a recession.

Python Implementation for Recession Analysis:

1. Data Retrieval and Visualization: We begin the recession analysis process by importing necessary Python libraries such as Pandas, Plotly, and Matplotlib. We then load the dataset containing 'monthly GDP growth data' for the United Kingdom, which serves as the basis for our analysis. Visualization techniques such as line plots, heatmaps, and bar charts are employed to visualize the trends in GDP growth, unemployment rates, and consumer spending over time.

2. Conversion to Quarterly Data: To analyze recession trends effectively, we convert the 'monthly GDP growth data' into 'quarterly data'. This aggregation allows us to identify patterns and trends at a broader temporal scale, facilitating a more comprehensive analysis of recessionary periods.

3. Recession Identification and Visualization: We identify recession periods based on negative GDP growth over consecutive quarters. This involves defining criteria for recession onset and analyzing the quarterly GDP growth data to detect periods of economic contraction. Visualizations such as line plots and heatmaps are used to visualize the recessionary periods alongside overall GDP growth trends.

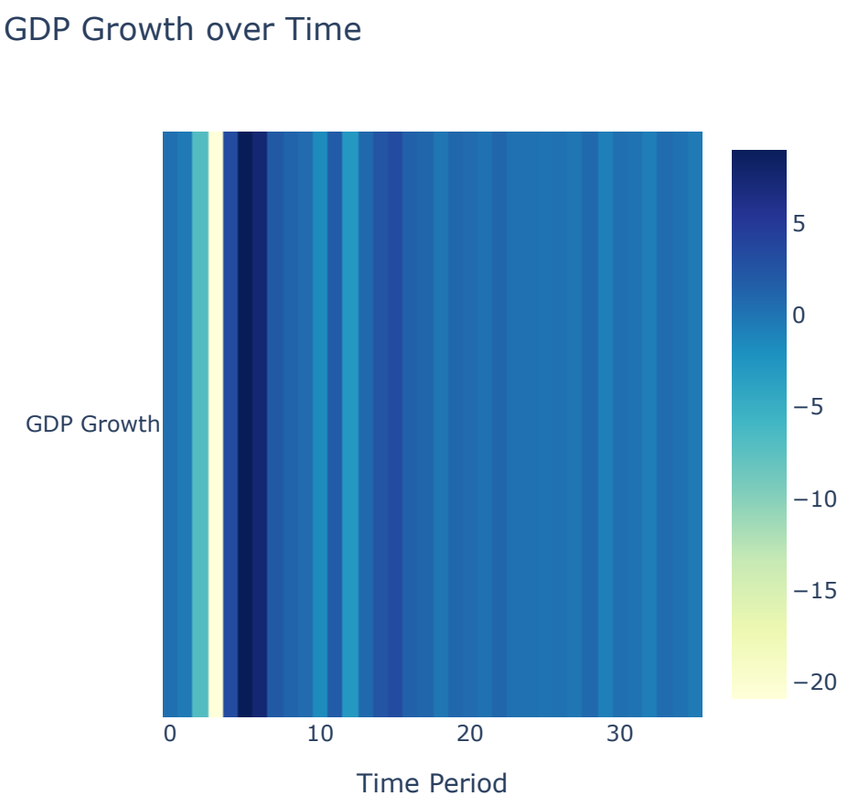
4. Analysis of Recession Severity: We assess the severity of the recession by analyzing additional metrics such as the duration and depth of economic contraction. By examining the duration of recessionary periods and the magnitude of GDP decline, we can gain insights into the severity of the economic downturn and its potential impact on various sectors of the economy.

"In conclusion, recession analysis using Python provides valuable insights into economic trends and dynamics, enabling policymakers, economists, and businesses to make informed decisions and formulate effective strategies for navigating economic downturns." By leveraging data science techniques and visualization tools, we can gain a deeper understanding of recessionary periods and their underlying causes, facilitating proactive measures to mitigate the adverse effects of economic downturns and promote long-term economic stability and growth.

In [1]:
```
import pandas as pd
import plotly.graph_objs as go
import plotly.express as px
import plotly.io as pio
pio.templates.default = "plotly_white"

data = pd.read_csv('C:/Users/anike/OneDrive/Desktop/Projects/Machine Learning/Recessic
print(data.head())
```

```
  Time Period  GDP Growth
0    /01/2020         0.3
1    /02/2020        -0.5
2    /03/2020        -7.0
3    /04/2020       -20.9
4    /05/2020         3.2
```

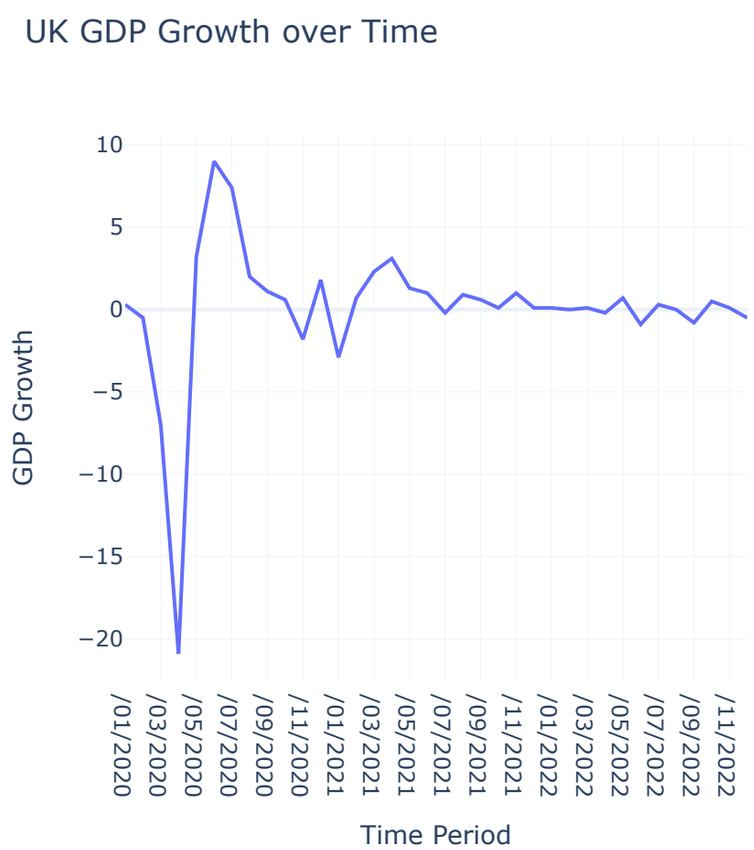
In [2]:
```
fig = go.Figure(data=go.Heatmap(
                   z=[data['GDP Growth']],
                   x=data.index,
                   y=['GDP Growth'],
                   colorscale='YlGnBu'))  # Change colorscale to 'YlGnBu'

fig.update_layout(title='GDP Growth over Time',
                  xaxis_title='Time Period',
                  yaxis_title='',
                  width=500,  # Set the width of the figure
                  height=500)  # Set the height of the figure

fig.show()
```



UK GDP Growth over Time

The line plot above depicts the monthly GDP growth rate of the United Kingdom over time. It provides a visual representation of the fluctuations in economic growth, allowing us to identify periods of expansion and contraction in the economy. Positive values indicate periods of economic growth, while negative values signify economic contraction or recessionary periods. Analyzing the trend in GDP growth enables us to understand the overall trajectory of the economy and identify potential recessionary periods. This visualization serves as a valuable tool for policymakers, economists, and analysts to monitor economic performance and formulate strategies to promote sustainable economic growth and stability.

In [3]:
```
# Visualizing GDP Growth Data

# Creating a line plot for GDP growth over time
fig = go.Figure(data=go.Scatter(x=data['Time Period'], y=data['GDP Growth'], mode='lir

# Updating layout
fig.update_layout(title='UK GDP Growth over Time',
                  xaxis_title='Time Period',
                  yaxis_title='GDP Growth',
                  width=500,  # Set the width of the figure
                  height=500)

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



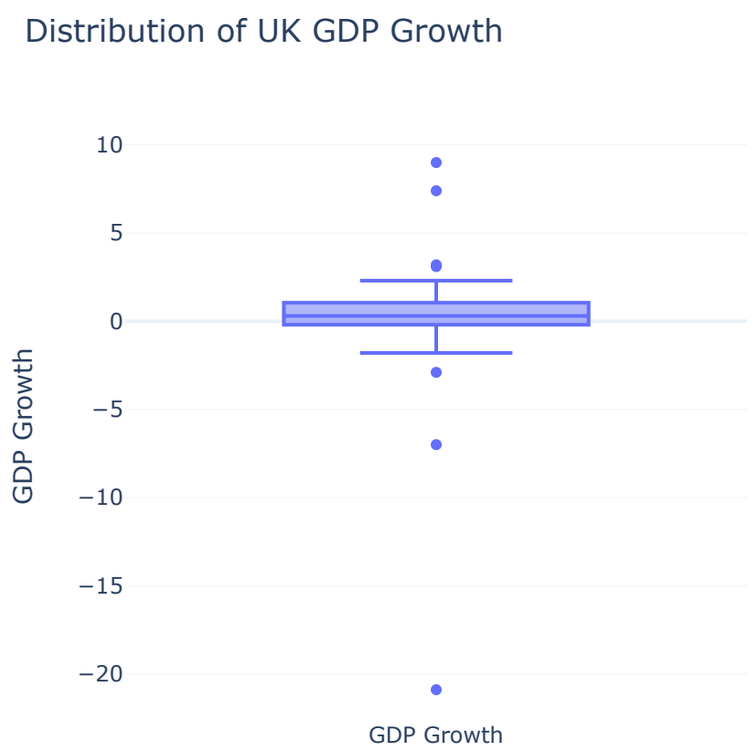
Distribution of UK GDP Growth:

The box plot provides a graphical representation of the distribution of GDP growth rates observed throughout the time period under consideration. It illustrates key statistics such as the median, quartiles, and potential outliers in the data. The central box represents the interquartile range (IQR), with the median marked by the line inside the box. The whiskers extend to the minimum and maximum values within 1.5 times the IQR from the first and third quartiles, respectively. Outliers, if present, are depicted as individual points beyond the whiskers.

Analyzing the distribution of GDP growth rates helps us understand the variability and range of economic performance over time. It enables us to identify periods of high growth, stability, or volatility in the economy, providing valuable insights for economic analysis and decision-making. Policymakers and analysts can use this visualization to assess the overall health of the

economy and formulate appropriate strategies to promote sustainable economic growth and stability.

In [4]:
```
# Creating a box plot for GDP growth distribution
fig_boxplot = go.Figure(data=go.Box(y=data['GDP Growth'], name='GDP Growth'))

# Updating layout for box plot
fig_boxplot.update_layout(
    title='Distribution of UK GDP Growth',
    yaxis_title='GDP Growth',
    height=500,
    width=500)

# Displaying the box plot
fig_boxplot.show()
```



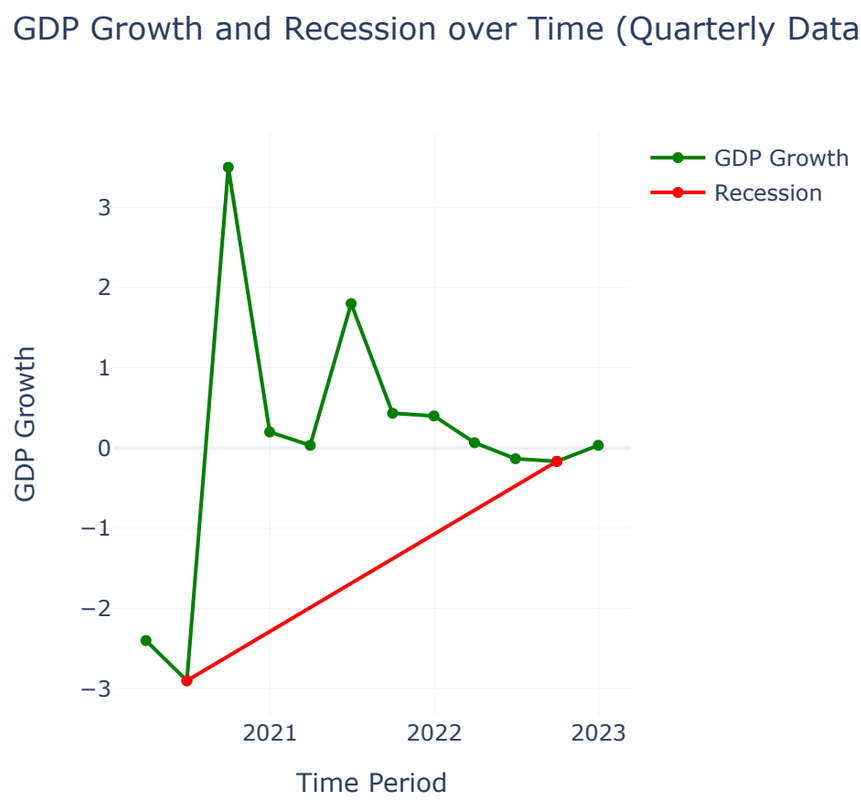
Converting the monthly data into quarterly data will allow us to analyze the recession more effectively, as it aligns with the conventional definition of a recession as two consecutive quarters of economic decline. Let's proceed with this conversion

In [5]:
```
# Convert monthly data to quarterly data using resample method
data['Time Period'] = pd.to_datetime(data['Time Period'], format='/%m/%Y')
data.set_index('Time Period', inplace=True)
quarterly_data = data.resample('Q').mean()
print(quarterly_data.head())
```

```
            GDP Growth
Time Period
2020-03-31    -2.400000
2020-06-30    -2.900000
2020-09-30     3.500000
2020-12-31     0.200000
2021-03-31     0.033333
```

In [6]:
```
 #Calculate recession based on quarterly GDP growth
quarterly_data['Recession'] = ((quarterly_data['GDP Growth'] < 0) & (quarterly_data['G

# Fill missing values with False (since the first quarter cannot be in a recession)
quarterly_data['Recession'].fillna(False, inplace=True)

# Plot the GDP growth and recession data
fig = go.Figure()
fig.add_trace(go.Scatter(x=quarterly_data.index,
                         y=quarterly_data['GDP Growth'],
```

```
                         name='GDP Growth',
                         line=dict(color='green', width=2)))
fig.add_trace(go.Scatter(x=quarterly_data[quarterly_data['Recession']].index,
                         y=quarterly_data[quarterly_data['Recession']]['GDP Growth'],
                         name='Recession', line=dict(color='red', width=2)))

fig.update_layout(title='GDP Growth and Recession over Time (Quarterly Data)',
                  xaxis_title='Time Period',
                  yaxis_title='GDP Growth',
                 height=500,
    width=500)

fig.show()
```



The red line depicted on the graph signifies periods characterized by negative GDP growth, which are typically regarded as recessionary phases. Conversely, the green line represents the overarching trend of GDP growth across time.

Moving forward, let's delve into an analysis of recession severity. Recession severity pertains to the degree to which the economy contracts during a recessionary period. A severe recession entails a more profound and protracted decline in economic activity, leading to adverse impacts on various economic metrics such as employment rates, income levels, and other pertinent indicators.

Here's a methodology for evaluating recession severity:

In [7]:
```
quarterly_data['Recession Start'] = quarterly_data['Recession'].ne(quarterly_data['Rec
recession_periods = quarterly_data.groupby('Recession Start')
recession_duration = recession_periods.size()
recession_severity = recession_periods['GDP Growth'].sum()

fig = go.Figure()
fig.add_trace(go.Bar(x=recession_duration.index, y=recession_duration,
                     name='Recession Duration'))
fig.add_trace(go.Bar(x=recession_severity.index, y=recession_severity,
                     name='Recession Severity'))

fig.update_layout(title='Duration and Severity of Recession',
                  xaxis_title='Recession Periods',
                  yaxis_title='Duration/Severity',
                 height=500,
    width=500)
```

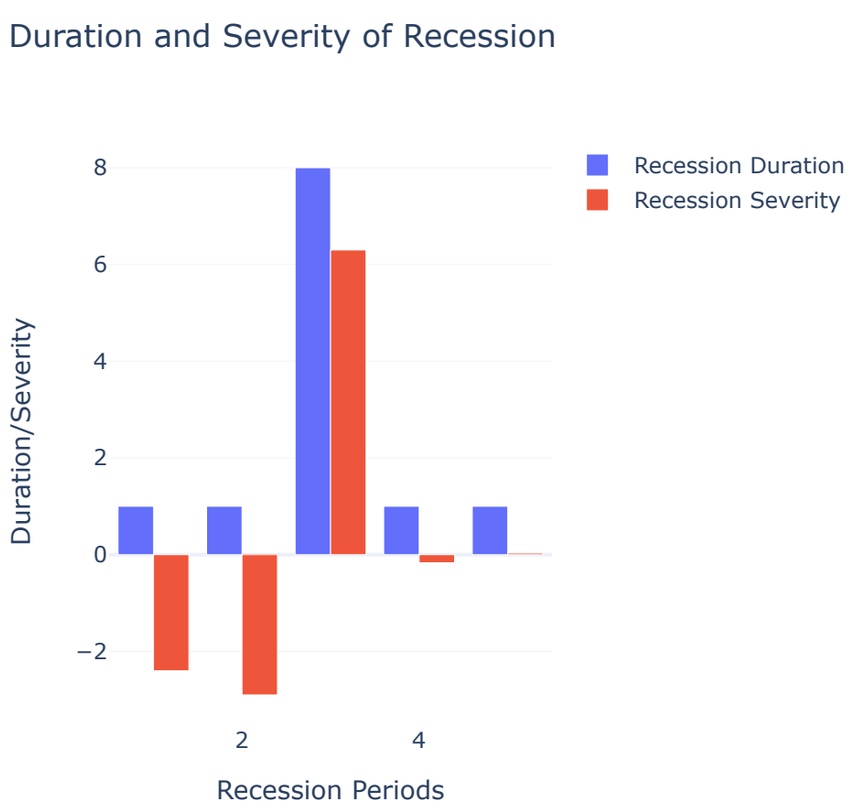
```
fig.show()
```

1. **Calculation of Recession Start:**
   - A new column called 'Recession Start' is added to the `quarterly_data` DataFrame. This column assigns a unique identifier to each recession period, starting from 1 and incrementing for each new recession.
   - The `ne()` function compares consecutive elements of the 'Recession' column to identify the start of each recession period. The result is then cumulative summed using `cumsum()`.
2. **Grouping and Calculation:**
   - Recession periods are grouped based on the 'Recession Start' column.
   - The `size()` function calculates the duration of each recession period by counting the number of quarters within each group, representing the duration of recession.
3. **Calculation of Recession Severity:**
   - The severity of each recession period is calculated by summing the GDP growth rates over the quarters within each recession period.
4. **Visualization:**
   - A bar chart is created using Plotly to visualize both the duration and severity of recession periods.
   - The x-axis represents the recession periods, while the y-axis denotes either the duration or severity of the recession.
   - Two sets of bars are plotted: one for recession duration and another for recession severity.

This visualization provides insights into the temporal distribution and intensity of recession periods, facilitating a deeper understanding of economic downturns and their impacts.

In [8]:
```
# Calculate the start of each recession period
quarterly_data['Recession Start'] = quarterly_data['Recession'].ne(quarterly_data['Rec

# Group recession periods and calculate their duration
recession_periods = quarterly_data.groupby('Recession Start')
```

```
recession_duration = recession_periods.size()

# Calculate the severity of each recession period
recession_severity = recession_periods['GDP Growth'].sum()

# Create a bar chart to visualize recession duration and severity
fig = go.Figure()
fig.add_trace(go.Bar(x=recession_duration.index, y=recession_duration,
                     name='Recession Duration'))
fig.add_trace(go.Bar(x=recession_severity.index, y=recession_severity,
                     name='Recession Severity'))

fig.update_layout(title='Duration and Severity of Recession',
                  xaxis_title='Recession Periods',
                  yaxis_title='Duration/Severity',
                 height=500,
    width=500)

fig.show()
```



Summary:

This Python-based analysis delves deep into the realm of recession trends, utilizing GDP growth data to unveil crucial insights into economic dynamics. Beginning with meticulous data preparation, we employ Python libraries to import and structure the dataset, laying the foundation for insightful analysis. Through visualization techniques such as heatmaps, we gain a comprehensive understanding of GDP growth trends over time, providing a bird's eye view of economic fluctuations.

Transitioning to quarterly data allows for a granular examination of recessionary patterns, enabling the identification and characterization of recession periods with precision. By scrutinizing quarterly GDP growth rates, we delineate recessionary epochs, shedding light on the cyclical nature of economic downturns. Moreover, quantitative analysis of recession severity, through metrics like duration and severity, offers a nuanced understanding of the economic impact of recessionary periods.

In conclusion, this Python-based analysis equips stakeholders with actionable insights into recession trends, empowering informed decision-making and strategic planning. By leveraging

Python's capabilities for data manipulation and visualization, we navigate the complex terrain of economic cycles with precision and foresight, fostering resilience in the face of economic uncertainty.