Import the Necessary Libraries:

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly.graph_objects as go
import plotly.express as px

print("All packages have been installed successfully!")
```

```
All packages have been installed successfully!
```

Import the Dataset:

In [2]:

```
df = pd.read_csv('BD_economic_indicators.csv')
df.head()
```

Out[2]:

	Year	GDP	GDP per capita	GDP growth	Inflation rate	Unemployed rate	Government debt	Total Investment
0	1980	41.2	500	3.10%	15.40%	NaN	NaN	14.44%
1	1981	47.4	560	5.60%	14.50%	NaN	NaN	17.16%
2	1982	52.0	597	3.20%	12.90%	NaN	NaN	17.36%
3	1983	56.5	633	4.60%	9.50%	NaN	NaN	16.56%
4	1984	61.0	664	4.20%	10.40%	NaN	NaN	16.48%

Data Cleaning and Processing:

Clean the data by removing percentage symbols, converting columns to numeric types, and filling missing values

In [3]:

```
df['GDP growth'] = df['GDP growth'].dropna().map(lambda x: x.rstrip('%'))
df['Inflation rate'] = df['Inflation rate'].dropna().map(lambda x: x.rstrip
('%'))
df['Total Investment'] = df['Total Investment'].dropna().map(lambda x: x.rstri
p('%'))
df['Unemployed rate'] = df['Unemployed rate'].dropna().map(lambda x: x.rstrip
('%'))
df['Government debt'] = df['Government debt'].dropna().map(lambda x: x.rstrip
('%'))
```

In [4]:

```
df['GDP per capita'] = df['GDP per capita'].replace(',', '', regex=True)
df['GDP per capita'] = pd.to_numeric(df['GDP per capita'], errors='coerce')
```

In [5]:

```
# Ensure all columns are treated as strings before removing percentage symbols
df['GDP growth'] = df['GDP growth'].astype(str).str.rstrip('%').astype(float)
df['Inflation rate'] = df['Inflation rate'].astype(str).str.rstrip('%').astype
(float)
df['Total Investment'] = df['Total Investment'].astype(str).str.rstrip('%').as
type(float)
df['Unemployed rate'] = df['Unemployed rate'].astype(str).str.rstrip('%').asty
pe(float)
df['Government debt'] = df['Government debt'].astype(str).str.rstrip('%').asty
pe(float)

# Remove commas from GDP per capita and convert to numeric
df['GDP per capita'] = df['GDP per capita'].replace(',', '', regex=True).astyp
e(float)

# Fill missing values with the mean of the respective columns using a dictiona
ry
df.fillna({
    'Unemployed rate': df['Unemployed rate'].mean(),
    'Government debt': df['Government debt'].mean()
}, inplace=True)
```

```
In [6]: df.head(200)
```

Out[6]:

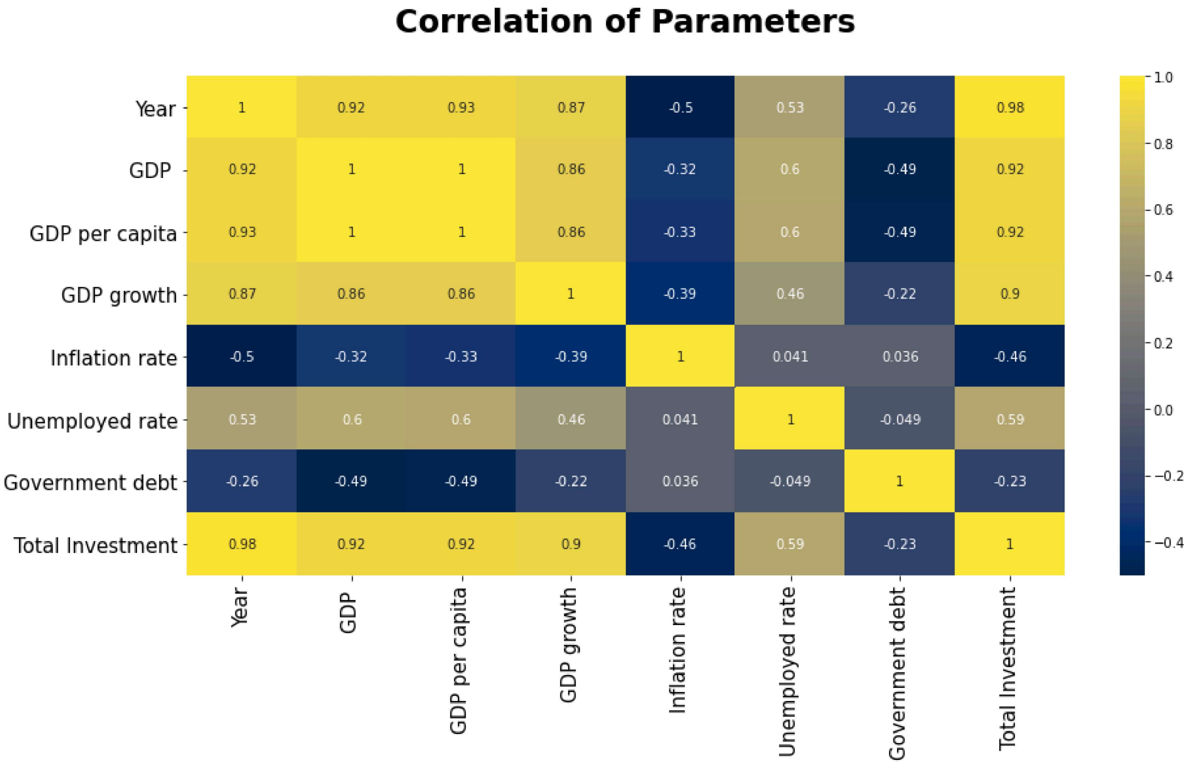
	Year	GDP	GDP per capita	GDP growth	Inflation rate	Unemployed rate	Government debt	Total Investment
0	1980	41.2	500.0	3.1	15.4	3.608276	37.694118	14.44
1	1981	47.4	560.0	5.6	14.5	3.608276	37.694118	17.16
2	1982	52.0	597.0	3.2	12.9	3.608276	37.694118	17.36
3	1983	56.5	633.0	4.6	9.5	3.608276	37.694118	16.56
4	1984	61.0	664.0	4.2	10.4	3.608276	37.694118	16.48
5	1985	65.3	693.0	3.7	10.5	3.608276	37.694118	15.83
6	1986	69.3	715.0	4.0	10.2	3.608276	37.694118	16.18
7	1987	73.1	735.0	2.9	10.8	3.608276	37.694118	15.47
8	1988	77.5	759.0	2.4	9.7	3.608276	37.694118	15.74
9	1989	84.0	801.0	4.3	8.7	3.608276	37.694118	16.12
10	1990	91.1	848.0	4.6	10.5	3.608276	37.694118	16.46
11	1991	98.1	892.0	4.2	8.3	2.200000	37.694118	16.90
12	1992	105.1	935.0	4.8	3.6	2.250000	37.694118	17.31
13	1993	112.3	977.0	4.3	3.0	2.370000	37.694118	17.95
14	1994	119.9	1021.0	4.5	6.2	2.440000	37.694118	18.40
15	1995	128.2	1069.0	4.8	10.1	2.480000	37.694118	19.12
16	1996	137.1	1120.0	5.0	2.5	2.510000	37.694118	20.73
17	1997	146.8	1175.0	5.3	5.0	2.690000	37.694118	21.82
18	1998	155.9	1223.0	5.0	8.6	2.830000	37.694118	22.12
19	1999	166.9	1284.0	5.4	6.2	3.100000	37.694118	22.72
20	2000	180.2	1361.0	5.6	2.5	3.270000	37.694118	23.81
21	2001	193.2	1434.0	4.8	1.9	3.550000	37.694118	24.17
22	2002	205.7	1501.0	4.8	3.7	3.960000	37.694118	24.34
23	2003	221.9	1594.0	5.8	5.4	4.320000	44.300000	24.68
24	2004	241.9	1713.0	6.1	6.1	4.300000	43.500000	24.99
25	2005	265.5	1855.0	6.3	7.0	4.250000	42.300000	25.83
26	2006	291.4	2018.0	6.9	6.8	3.590000	42.300000	26.14
27	2007	319.7	2183.0	6.5	9.1	3.770000	41.900000	26.18
28	2008	344.0	2325.0	5.5	8.9	4.070000	40.600000	26.20
29	2009	365.0	2441.0	5.3	4.9	5.000000	39.500000	26.21
30	2010	391.7	2592.0	6.0	9.4	3.370000	35.500000	26.25
31	2011	425.8	2785.0	6.5	11.5	3.710000	36.600000	27.42
32	2012	460.8	2979.0	6.3	6.2	4.040000	36.200000	28.26
33	2013	496.5	3171.0	6.0	7.5	4.430000	35.800000	28.39
34	2014	537.3	3396.0	6.3	7.0	4.410000	35.300000	28.58

	Year	GDP	GDP per capita	GDP growth	Inflation rate	Unemployed rate	Government debt	Total Investment
35	2015	581.6	3638.0	6.8	6.2	4.420000	33.600000	28.89
36	2016	629.9	3900.0	7.2	5.7	4.350000	33.300000	29.65
37	2017	690.5	4231.0	7.6	5.6	4.370000	32.600000	30.51
38	2018	753.4	4630.0	7.9	5.6	4.300000	34.000000	31.23
39	2019	817.6	5028.0	8.1	5.5	4.290000	33.500000	31.60

Correlation Heatmap

The correlation heatmap shows strong relationships between certain economic indicators, such as GDP growth and total investment, as well as between inflation and unemployment. These correlations can be useful for policymakers to understand which areas to focus on to achieve balanced economic growth.

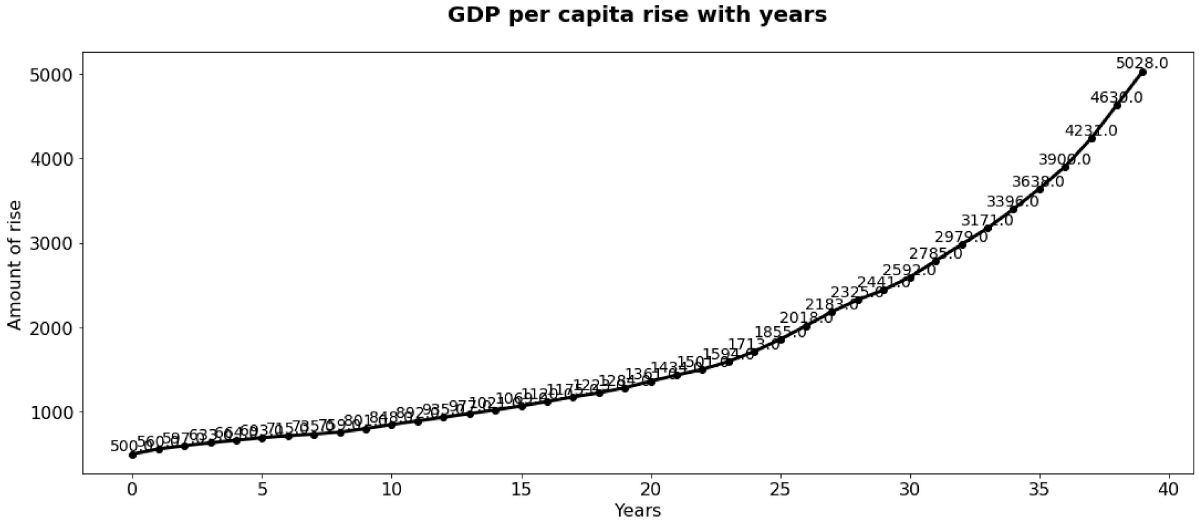
In [7]:
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(15, 7))
sns.heatmap(df.corr(), annot=True, cmap='cividis')  # Use 'cividis' color pale
tte
plt.xticks(fontsize=15, color='black')
plt.yticks(fontsize=15, color='black')
plt.title("Correlation of Parameters\n", fontsize=24, fontweight='bold', color
='black')
plt.show()
```



Line Plot of GDP per Capita Over the Years

This section visualizes the rise in GDP per capita over the years.

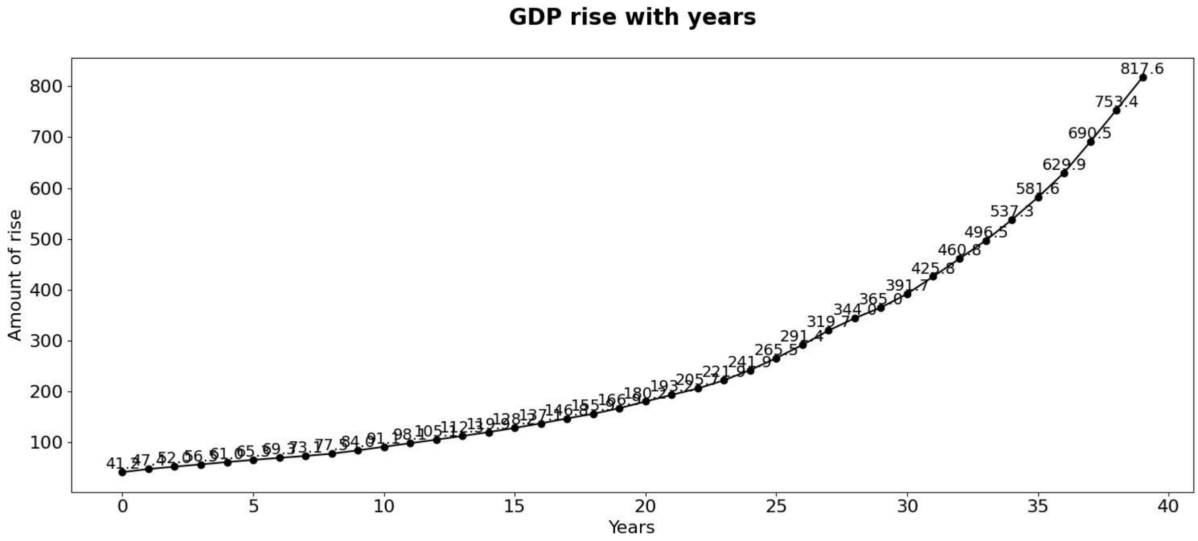
In [20]:
```
# If you want to create df1 as a copy of df (optional)
df1 = df.copy()

import matplotlib.pyplot as plt
import seaborn as sns

# Set the color palette
sns.set_palette('cividis')

# Create the line plot using df or df1 depending on what you intend
plt.figure(figsize=(18,7))
plt.plot(df.index, df['GDP per capita'], color='black', marker='o', linestyle
='-', linewidth=3)

# Customize ticks and labels
plt.xticks(fontsize=16, color='black')
plt.xlabel('Years', fontsize=16, color='black')
plt.ylabel('Amount of rise', fontsize=16, color='black')
plt.yticks(fontsize=16, color='black')
plt.title('GDP per capita rise with years\n', fontsize=20, fontweight='bold',
color='black')

# Add data labels on each point
for i, value in enumerate(df['GDP per capita']):
    plt.text(df.index[i], value, round(value, 2), ha='center', va='bottom', fo
ntsize=14, color='black')

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



Line Plot of GDP Over the Years

This section visualizes the rise in GDP over the years.

```
In [68]: import matplotlib.pyplot as plt
         import seaborn as sns

         # Set the color palette to 'Cividis'
         sns.set_palette('cividis')

         # Create the line plot
         plt.figure(figsize=(18,7))
         plt.plot(df1.index, df1['GDP '], color='black', marker='o', linestyle='-')

         # Customize ticks and labels
         plt.xticks(fontsize=16, color='black')
         plt.xlabel('Years', fontsize=16, color='black')
         plt.ylabel('Amount of rise', fontsize=16, color='black')
         plt.yticks(fontsize=16, color='black')
         plt.title('GDP rise with years\n', fontsize=20, fontweight='bold', color='blac
         k')

         # Add data labels on each point
         for i, value in enumerate(df1['GDP ']):
             plt.text(df1.index[i], value, round(value, 2), ha='center', va='bottom', f
         ontsize=14, color='black')

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



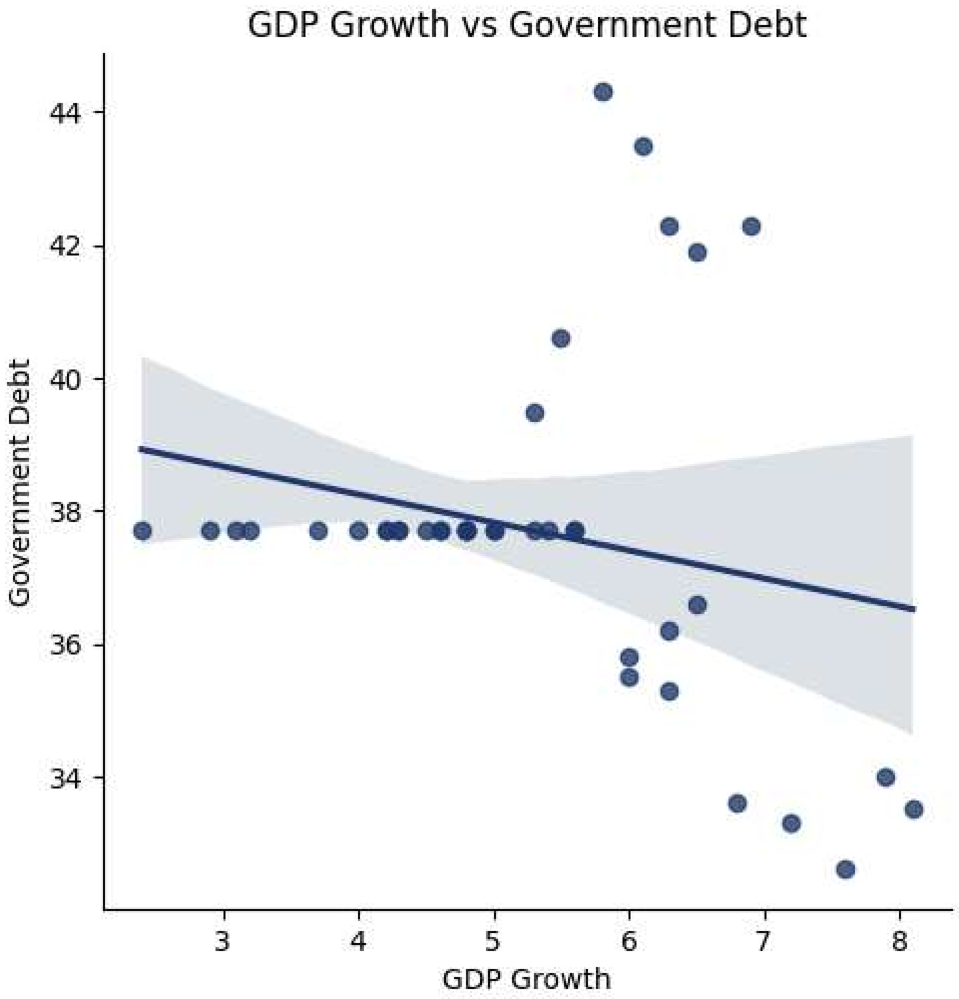
Linear Regression Plot: GDP Growth vs. Government Debt

This section creates a scatter plot with a regression line showing the relationship between GDP growth and government debt.

In [69]:

```
import seaborn as sns

plt.figure(figsize=(18,7))
sns.lmplot(x='GDP growth', y='Government debt', data=df)
plt.title('GDP Growth vs Government Debt')
plt.xlabel('GDP Growth')
plt.ylabel('Government Debt')
plt.show()
```

```
<Figure size 1800x700 with 0 Axes>
```

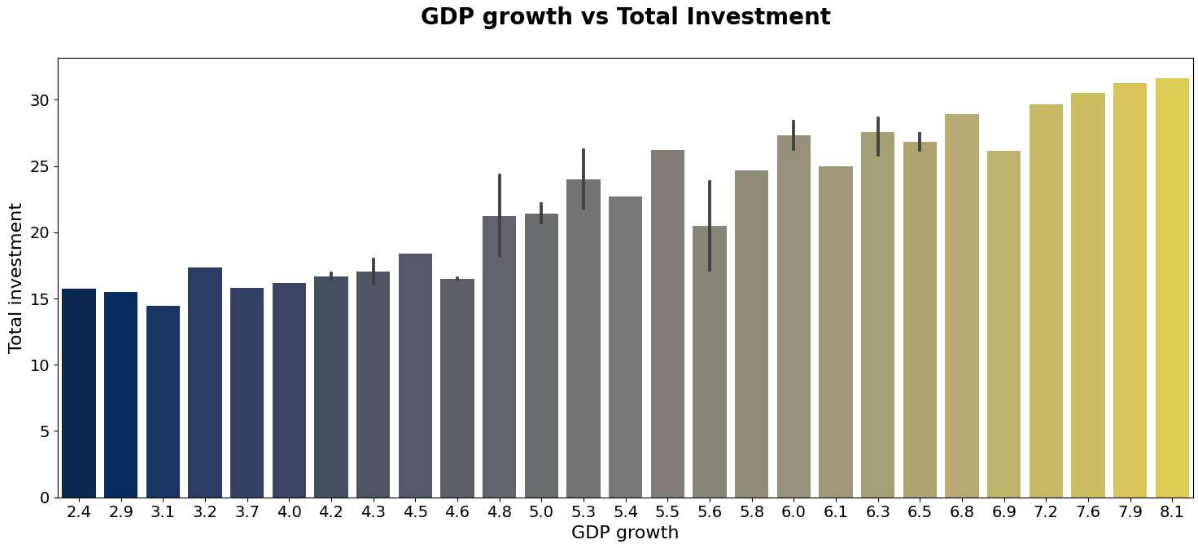


Bar Plot: GDP Growth vs. Total Investment

This section creates a bar plot to visualize the relationship between GDP growth and total investment.

In [70]:

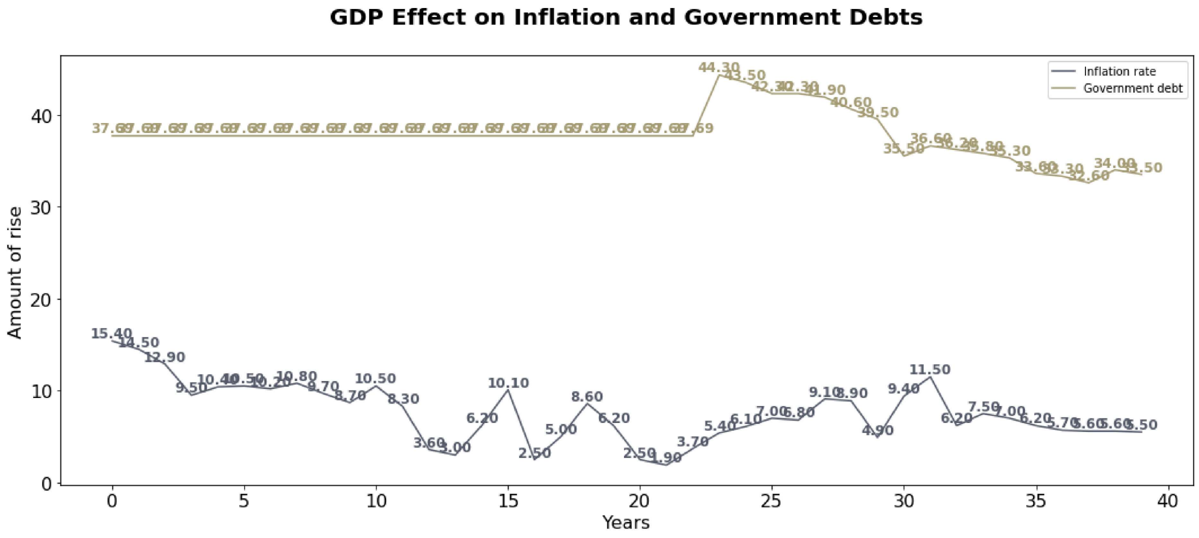
```python
# Create the bar plot using Seaborn with the 'Cividis' color palette
plt.figure(figsize=(18,7))
sns.barplot(x='GDP growth', y='Total Investment', data=df, palette='cividis')

# Customize ticks and labels
plt.xticks(fontsize=14, color='black')
plt.xlabel('GDP growth', fontsize=16, color='black')
plt.ylabel('Total investment', fontsize=16, color='black')
plt.yticks(fontsize=14, color='black')
plt.title('GDP growth vs Total Investment\n', fontsize=20, fontweight='bold',
color='black')


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



Dual Line Plot: GDP Effect on Inflation and Government Debt

This section visualizes the relationship between GDP, inflation rate, and government debt.

```
In [14]: import matplotlib.pyplot as plt
         import seaborn as sns

         # Set the 'Cividis' color palette
         cividis_colors = sns.color_palette("cividis", n_colors=2)

         # Plotting
         plt.figure(figsize=(18,7))
         df['Inflation rate'].plot(color=cividis_colors[0], label='Inflation rate')
         df['Government debt'].plot(color=cividis_colors[1], label='Government debt')

         # Customize ticks and labels
         plt.xticks(fontsize=16, color='black')
         plt.xlabel('Years', fontsize=16, color='black')
         plt.ylabel('Amount of rise', fontsize=16, color='black')
         plt.yticks(fontsize=16, color='black')

         # Add title and legend
         plt.title('GDP Effect on Inflation and Government Debts\n', fontsize=20, fontw
         eight='bold', color='black')
         plt.legend()

         # Add data labels on top of bars
         for i in range(len(df)):
             plt.text(df.index[i], df['Inflation rate'].iloc[i], f'{df["Inflation rat
         e"].iloc[i]:.2f}',
                       ha='center', va='bottom', fontsize=12, color=cividis_colors[0], f
         ontweight='bold')
             plt.text(df.index[i], df['Government debt'].iloc[i], f'{df["Government deb
         t"].iloc[i]:.2f}',
                       ha='center', va='bottom', fontsize=12, color=cividis_colors[1], f
         ontweight='bold')

         # Show plot
         plt.show()
```



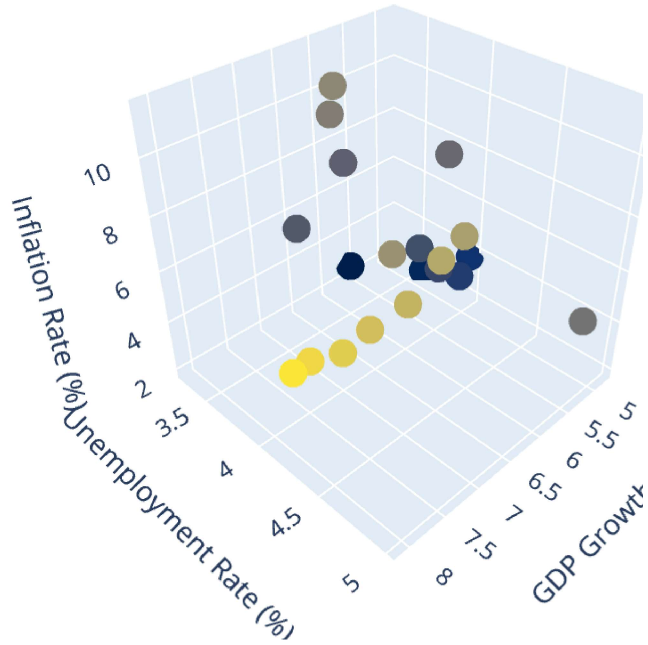
3D Scatter Plot: GDP Growth, Unemployment Rate & Inflation Rate

This section creates a 3D scatter plot to visualize the relationship between GDP growth, unemployment rate, and inflation rate.

In [13]:

```
import plotly.express as px

# Create a 3D scatter plot with 'Cividis' color palette and labels
fig = px.scatter_3d(df.tail(20),
                    x='GDP growth',
                    y='Unemployed rate',
                    z='Inflation rate',
                    color='Year',
                    color_continuous_scale='Cividis',  # Apply Cividis color p
alette
                    labels={
                        'GDP growth': 'GDP Growth (%)',  # Label for x-axis
                        'Unemployed rate': 'Unemployment Rate (%)',  # Label f
or y-axis
                        'Inflation rate': 'Inflation Rate (%)',  # Label for z
-axis
                        'Year': 'Year'  # Label for color legend
                    })

# Update layout with a title
fig.update_layout(title='3D Scatter Plot of GDP Growth, Unemployment Rate & In
flation Rate')

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

3D Scatter Plot of GDP Growth, Unemployment Rate & Inflation Rate



Conclusion

The data analysis provided valuable insights into the economic conditions of Bangladesh over the years. The trends in GDP growth, inflation, government debt, and other key indicators reveal the challenges and opportunities the country has faced. These insights can be crucial for policymakers, economists, and analysts in understanding the factors driving economic growth and in making informed decisions for future economic planning.The analysis and visualizations provide a clear picture of how various economic factors like GDP growth, inflation rate, total investment, unemployment rate, and government debt have evolved over time.