Assessing the Impact of Environmental, Geopolitical, and Health Factors on British Economic Stability

Authors: Millor Lei, Leisha B., Chumeng Zhang

Summary

1) **Visualizing the impact of Climate Change on the British economy: How do greenhouse gas emissions affect GDP from 2022 to 2023?**

   We discovered that there is no direct relationship between greenhouse gas emissions and GDP during this period. Despite a decrease in greenhouse gas emissions, GDP also showed a decreasing trend. This suggests that other factors beyond greenhouse gas emissions are influencing GDP in the British economy during this timeframe.

2) **Visualizing the economic impact of the UK leaving the EU: Was Brexit a good decision?**

   Upon analyzing datasets focused solely on employment, unemployment, inflation, and GDP growth, without incorporating external factors or broader datasets, it becomes challenging to definitively assess the economic implications of Brexit.

3) **Visualizing the effect of Covid-19 on the British economy: To what extent did death count affect GDP?**

   For the years 2020 and 2021, an inversely proportional relationship between death count and GDP could be inferred; in other words, as death count increased, GDP decreased. However, for the years 2022 and 2023, our visualizations indicated a proportional relationship.

Motivation

Our primary motivation behind choosing these research questions is the significance of exploring economic dynamics and how they affect decision-making processes at national and international levels. Understanding the potential impacts of climate change policies on global trade, along with the consequences of the UK's relationship with the EU and the effects of global pandemics all contribute towards this. By analyzing and contextualizing the interplay between environmental, geopolitical, and health factors, this research contributes to enhancing our understanding of the dynamic nature of the global economy. The insights gained from this analysis can inform policymakers, businesses, and communities in devising strategies to mitigate risks, capitalize on opportunities, and foster sustainable and resilient economic growth

Data setting

1) **Environmental Environmental Account** (Estimates of greenhouse gas emissions): experimental estimates of total greenhouse gas (GHG) and carbon dioxide (CO_2) emissions, quarter 1 1998 to quarter 2 2022.
2) **Economic output (GDP)**: the percentage change on growth of domestic product from 1955 to 2023 based on quarters.
3) **Employment Rate:** Employment rate in the UK aged 16-64 from 1992 APR to 2023 OCT.
4) **Unemployment Rate:** Unemployment rate aged 16+ from 1992 APR to 2023 OCT.
5) **Inflation:** Annual inflation rate of all times in the UK from 1989 to 2023.
6) **Health and Social Care**: weekly provisional figures of care home resident deaths registered in England and Wales from 2021 to 2023.

Research Question 1

Method

To obtain the data, we first imported the necessary libraries, including Pandas for data manipulation and Matplotlib for plotting. We then loaded the GDP and greenhouse gas emission datasets from CSV files, renaming columns and filtering the data to include only the relevant years. After merging the datasets based on the 'Year' column, we prepared the data for plotting. We converted the 'GDP' column to numerical values for the GDP data to ensure compatibility with the polynomial fit calculation. We then used Matplotlib to create a figure with two subplots, one for GDP and another for greenhouse gas emissions. In the first subplot, we plotted the GDP data as blue dots and added a best-fit line using NumPy's polyfit and poly1d functions, showing the trend in GDP over the years. In the second subplot, we plotted the greenhouse gas emission data as green dots. We adjusted the axis labels, titles, and tick marks for better readability and visual appeal. Overall, the plot provides a visual representation of the GDP and greenhouse gas emissions trends over the specified period, highlighting any notable patterns or correlations.

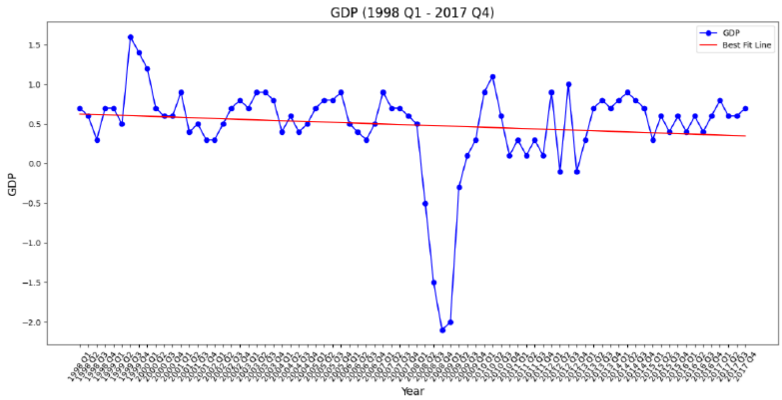
Results

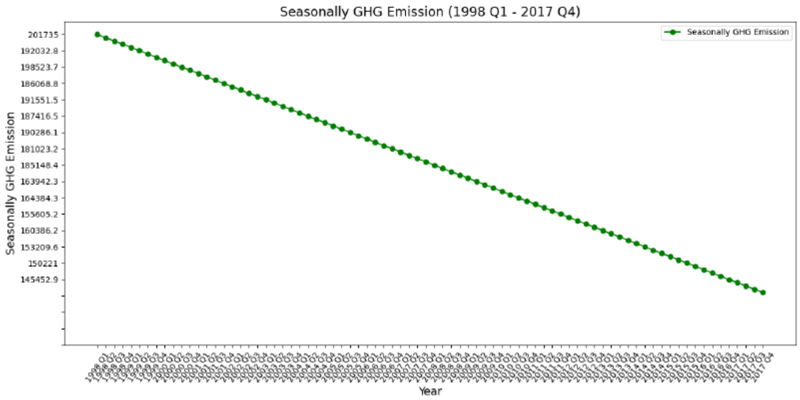
Our analysis revealed no direct relationship between greenhouse gas (GHG) emissions and GDP in the British economy from 2022 to 2023, excluding the pandemic period. Despite a decrease in GHG emissions, GDP also showed a decreasing trend, indicating that other factors are influencing economic performance.

This unexpected result underscores the complexity of the relationship between GHG emissions and GDP. Factors such as changes in consumer behavior, technological advancements, and global economic conditions may be influencing GDP independently of GHG emissions. Additionally, the limitations of our data and model may have impacted the results.

These findings have implications for policymakers and researchers. Policymakers should adopt a comprehensive approach to economic and environmental policies, considering various factors influencing GDP. Researchers should further investigate

These findings have implications for policymakers and researchers. Policymakers should adopt a comprehensive approach to economic and environmental policies, considering various factors influencing GDP. Researchers should further investigate the drivers of GDP changes to understand the dynamics at play better.





Impact and Limitations

Policymakers can benefit by gaining a more nuanced understanding of the complexities involved in addressing climate change while maintaining economic growth. Economists could also benefit by incorporating these findings into their models and theories regarding the impact of GHG emissions on GDP. This could lead to more informed policy decisions that balance environmental concerns with economic considerations. Therefore, environmentalists may find our results useful in advocating for sustainable policies that consider both environmental and economic impacts. However, our analysis has potential drawbacks. Advocates for strict emissions reduction policies may be disappointed, as our findings suggest that the relationship between emissions and GDP is not straightforward. Industries reliant on GHG emissions may also be negatively affected if policies are implemented based solely on emissions reduction without considering broader economic implications.

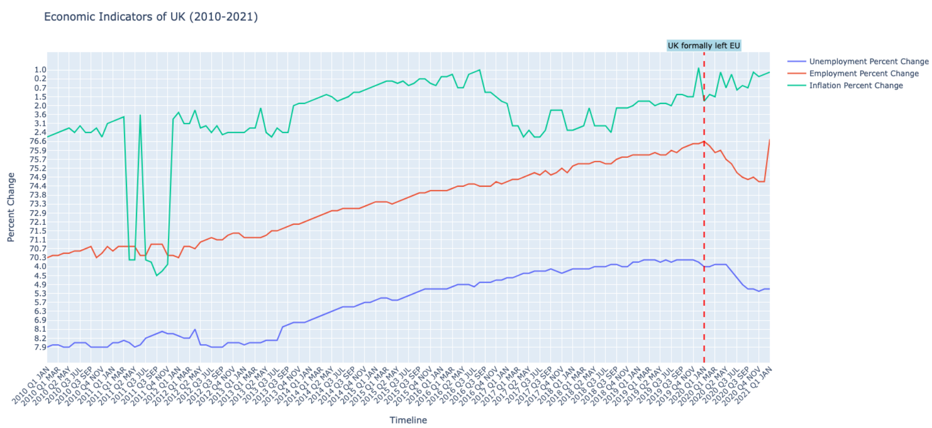
Research Question 2

Method

We began by gathering comprehensive datasets covering key economic indicators before and after Brexit, such as GDP, unemployment rates, employment rates, and inflation rate of the UK. These data sets were sourced from official government databases. The preparation phase involved cleaning the data for inconsistencies, missing values, and ensuring compatibility across different datasets for comparative analysis. Once the datasets were meticulously cleaned, we proceeded with a merge operation to consolidate these datasets, aiming to facilitate a more intuitive visualization process. To effectively visualize the economic impact of Bexit, we employed Plotly, which allowed us to create detailed line graphs that depict the temporal changes in these critical economic indicators, thereby illustrating the economic trajectory of the UK in the wake of Brexit.

Result

The visual data analysis reveals a noticeable decline in both employment and unemployment rates after Brexit, hinting at a nuanced economic landscape that could be symptomatic of stagflation. This condition entails slow or negative economic growth paired with high inflation, signaling supply-side constraints like rising production costs or supply chain disruptions, alongside diminished labor market participation. People might be leaving the workforce due to discouragement or other non-economic reasons, reducing the unemployment rate not because of job growth but due to a shrinking labor force. At the same time, inflationary pressures could arise from increased costs being passed on to consumers, demand-pull factors, or expansive monetary policies. This scenario poses significant challenges for policymakers, as it combines the adverse effects of economic downturn with the complexities of managing inflation. Based on this information, we conclude that Brexit was not a good decision.



Impact and Limitations

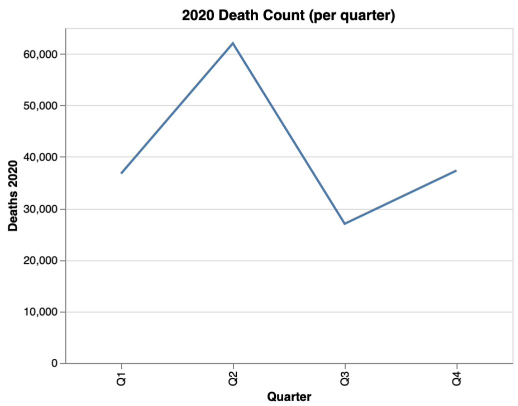
The findings lay the groundwork for future research on the long-term impacts of Brexit on the UK economy, particularly concerning employment, unemployment, and inflation trends. It opens avenues for exploring the effectiveness of different policy responses to such economic conditions. While the study suggests a correlation between Brexit and the observed economic indicators, establishing a direct causality is challenging. Other external factors, such as global economic trends, pandemic impacts, or international trade dynamics, could also influence the UK's economic conditions.

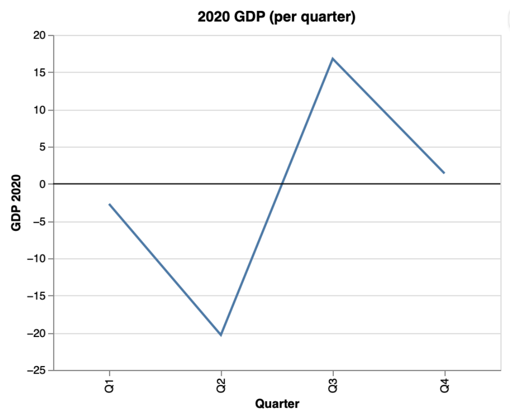
Research Question 3

Methods

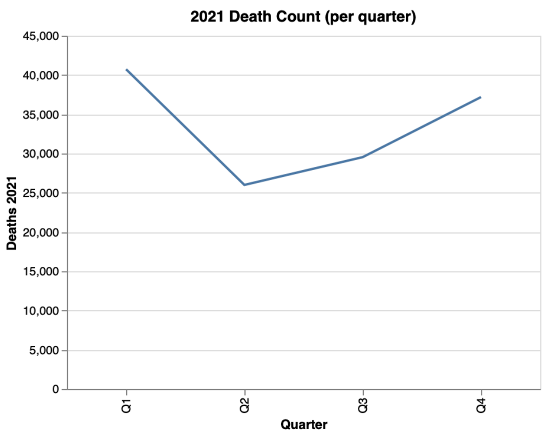
We started by first filtering and cleaning the data, in order to get 2 datasets that had the death count and GDP, respectively, organized by quarter. While using the weekly death count would have been more accurate, GDP data was only available on a quarterly basis. Once the datasets were finalized, we used Altair to plot the death count and GDP per year side-by-side. While our original plan was to plot them both together, the resulting graph was not effectively displaying the relevant information due to the difference in ranges between the two datasets– death count was in the ten thousand, while GDP was mostly in one-digit decimals. Once the plots were finalized, we were able to make inferences and establish correlations.

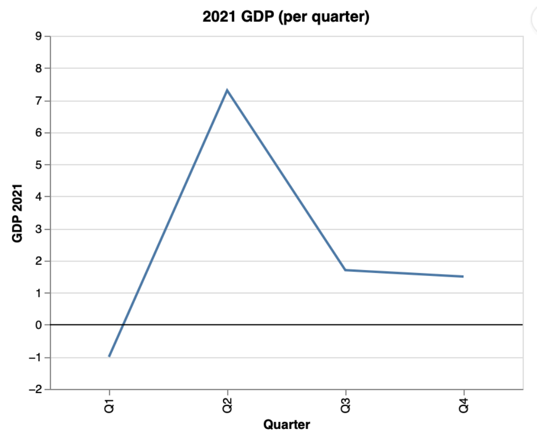
Results



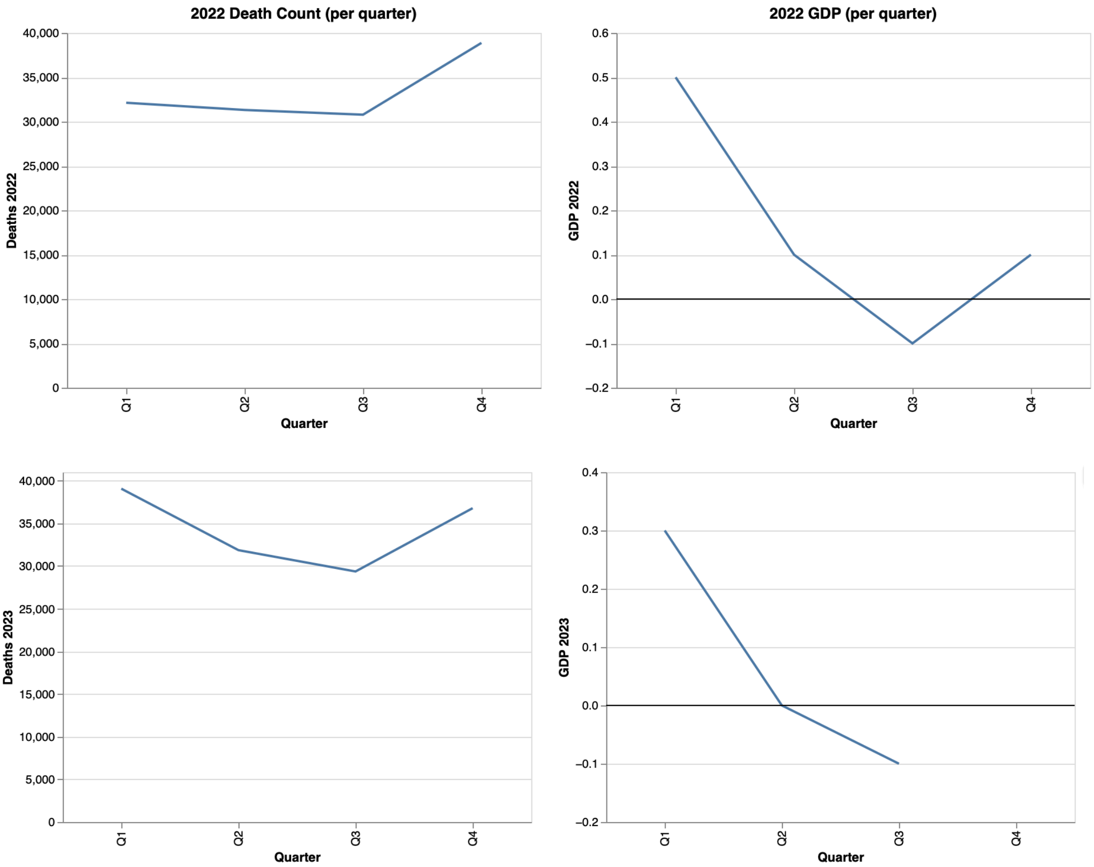


We can see that for the year 2020, the increases in the quarterly death count correspond to a decrease in GDP, and the decreases in the quarterly death count correspond to an increase in GDP. In other words, for the year 2020, GDP and death count look to be inversely proportional.





For 2021, we can again see an inverse correlation between GDP and death count. However, the correlation is not as explicit as it was in 2020.



For the years 2022 and 2023, a correlation can be seen; however, it implies that the relationship is proportional as opposed to earlier years which implied the relationship was inversely proportional.

Impact and Limitations

From this investigation, we can conclude that death count and GDP are related, as a correlation can be deduced from each of the years examined. However, we cannot conclude that there is a cause-and-effect relationship as the nature of the correlation (proportional vs inversely proportional) differs between years. Therefore, while death count and GDP are related, there are numerous other factors that affect both of these variables. Economists can benefit from this analysis, as it might lead to a gateway of investigating what exactly affects GDP and deducing a cause-and-effect relationship.

Certain conspiracy theorists who believe that the pandemic had no effect on the economy will be negatively impacted as this analysis shows there is a relationship.

A major limitation was that there was no data available for the fourth quarter of 2023, as the database had been released before that information was known. As such, the plots and implications for 2023 are not as accurate as they are for other years, due to missing data.

Challenge Goals

We followed through with our multiple datasets challenge goal, as we wanted to examine how various factors affected the British economy. We utilized 6 different datasets in our project through join and merge operations.

However, given certain time constraints, we felt we could not satisfy the machine learning challenge goal. As such, we chose to work with new libraries (Plotly and Altair). This allowed us to generate effective data visualizations that we could use to answer our research questions.

Plan Evaluation

In reflecting on our proposed work plan, we've found that our estimates for the hours spent on the project were quite accurate. We were able to stick to our planned schedule and allocate our time effectively. However, the predictions we made regarding the results based on the data have turned out to be different from what we expected.

This discrepancy could be due to several reasons. One possibility is that our initial assumptions or models were not able to fully capture the complexities of the data, leading to unexpected outcomes. Additionally, external factors such as changes in the economic environment may have influenced the results.

Moving forward, it will be important for us to reevaluate our approach and consider refining our models or data analysis techniques to better align with the actual outcomes. This might involve conducting further analysis or incorporating additional data sources to improve the accuracy of our predictions. Overall, while our work plan estimates for hours spent were accurate, there is still room for improvement in predicting results based on data analysis.

Testing

Testing was performed as we went along. We felt this was best since it let us catch bugs as we went along. We created test sets that were subsets of our full dataset. Because our data didn't produce any calculation outputs, we didn't use the assert equals function at any point. Our results can be trusted since we have a reliable data source and because we ran our program on a test set each time we changed something significant. We checked our data to ensure that the result made sense and adjusted accordingly if there were any problems. For example, we removed the data from the pandemic since it showed up as an extreme outlier because the year hasn't finished yet so that data is incomplete.

For the third research question, we relied on assert statements that ensured our data was in the correct format. This made sure the plots were as accurate as possible.

Collaboration

- We consulted the documentation for Plotly and Altair, as well as online discussion posts relating to bugs (such as in StackOverflow).
- We utilized some generative AI tools to help debugging with the error message given by the JupyterHub as prompt

Final

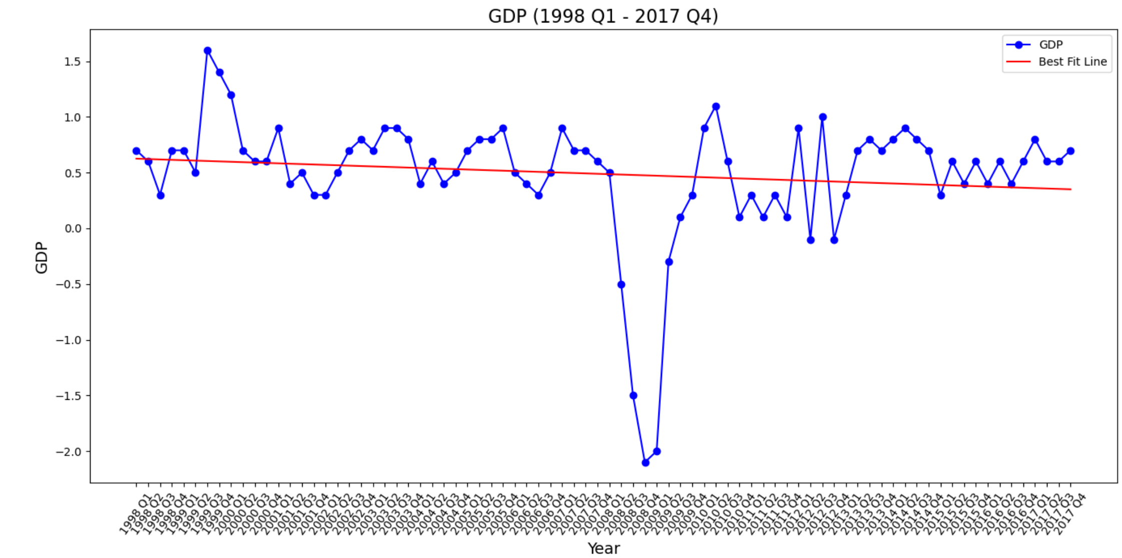
March 11, 2024

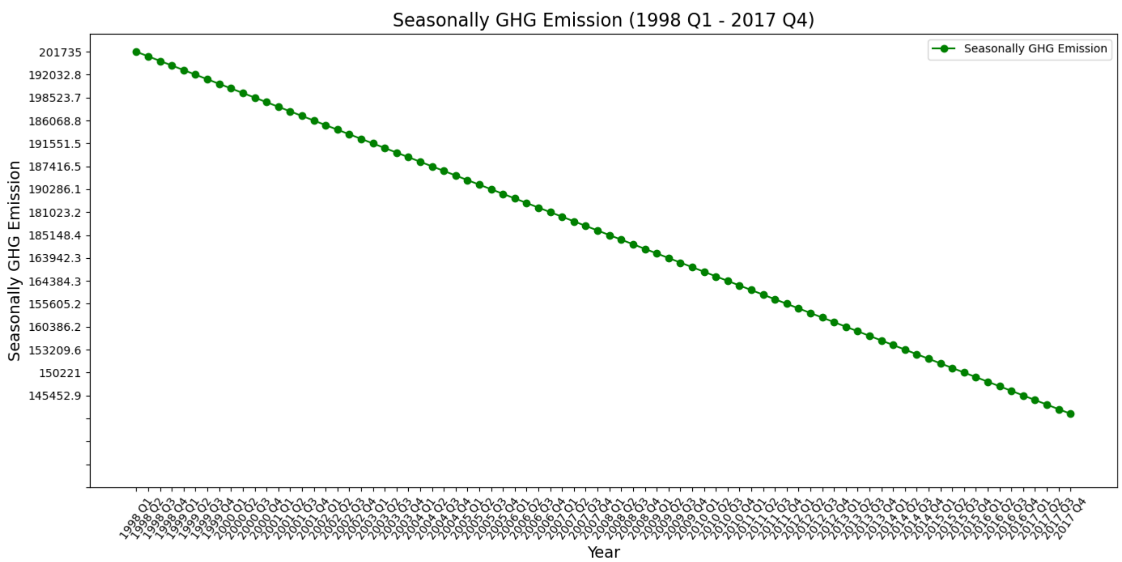
[4]:
```
# All csv files and codes are in Millor Lei's JupyterLab
!pip install plotly
```

```
Requirement already satisfied: plotly in /opt/conda/lib/python3.10/site-packages
(5.19.0)
Requirement already satisfied: tenacity>=6.2.0 in
/opt/conda/lib/python3.10/site-packages (from plotly) (8.2.3)
Requirement already satisfied: packaging in /opt/conda/lib/python3.10/site-
packages (from plotly) (23.2)
```

[71]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

gdp = pd.read_csv('gdp.csv')
gdp = gdp.rename(columns={'Title': 'Year', 'Gross Domestic Product: Quarter on␣
 ↪Quarter growth: CVM SA %': 'GDP'})
gdp = gdp[gdp['Year'].between('1998 Q1', '2017 Q4')]

emission = pd.read_csv('green gas emission.csv')
emission = emission.iloc[:, [0, 2]]
emission = emission.rename(columns={'Experimental estimates of total greenhouse␣
 ↪gas (GHG)1  and carbon dixoide (CO2) emissions, quarter 1 (Jan to Mar) 1998␣
 ↪to quarter 2 (Apr to Jun) 2022': 'Year'})
emission = emission.rename(columns={emission.columns[1]: 'Seasonally GHG␣
 ↪Emission'})
emission = emission[emission['Year'].between('1998 Q1', '2017 Q4')]

data = gdp.merge(emission, on='Year')

# Convert 'GDP' column to numerical values
data['GDP'] = pd.to_numeric(data['GDP'], errors='coerce')

# Set a larger width and height for the plots
plt.figure(figsize=(14, 14))

# Plot GDP data in blue
```

```python
plt.subplot(2, 1, 1)
plt.plot(data['Year'], data['GDP'], label='GDP', marker='o', color='blue')  # Show dot for each data point

# best fit line
z = np.polyfit(range(len(data['GDP'])), data['GDP'], 1)
p = np.poly1d(z)
plt.plot(data['Year'], p(range(len(data['GDP']))), color='red', label='Best Fit Line')

plt.xlabel('Year', fontsize=14)
plt.ylabel('GDP', fontsize=14)
plt.title('GDP (1998 Q1 - 2017 Q4)', fontsize=16)
plt.xticks(rotation=55)
plt.legend()

# Plot Seasonally GHG Emission data in green
plt.subplot(2, 1, 2)
plt.plot(data['Year'], data['Seasonally GHG Emission'], label='Seasonally GHG Emission', marker='o', color='green')  # Show dot for each data point
plt.xlabel('Year', fontsize=14)
plt.ylabel('Seasonally GHG Emission', fontsize=14)
plt.title('Seasonally GHG Emission (1998 Q1 - 2017 Q4)', fontsize=16)
plt.yticks(np.arange(0, 100, step=5))
plt.xticks(rotation=55)
plt.gca().invert_yaxis()  # Invert the y-axis
plt.legend()


plt.tight_layout()
plt.show()
```





[5]:
```
import pandas as pd
inflation = pd.read_csv('inflation.csv')
inflation = inflation.rename(columns={'Title': 'Year', 'CPIH ANNUAL RATE 00: ALL ITEMS 2015=100': 'Inflation Percent Change'})
inflation = inflation.iloc[434:567]
inflation.reset_index(drop=True, inplace=True)

employment = pd.read_csv('employment.csv')
employment = employment.rename(columns={'Title': 'Year', 'LFS Experimental: Employment rate: UK: All: Aged 16-64 (%): SA': 'Employment Percent Change'})
employment = employment.iloc[220:353]
employment.reset_index(drop=True, inplace=True)
```

```
unemployment = pd.read_csv('unemployment.csv')
unemployment = unemployment.rename(columns={'Title': 'Year', 'LFS Experimental:␣
↪ILO Unemployment rate: UK: All: Aged 16+ (%): SA': 'Unemployment Percent␣
↪Change'})
unemployment = unemployment.iloc[220:353]
unemployment.reset_index(drop=True, inplace=True)

df_merged = pd.merge(unemployment, employment, on='Year', how='inner')
df_merged = pd.merge(df_merged, inflation, on='Year', how='inner')

def month_to_quarter(month):
    quarters = {
        'JAN': 'Q1', 'FEB': 'Q1', 'MAR': 'Q1',
        'APR': 'Q2', 'MAY': 'Q2', 'JUN': 'Q2',
        'JUL': 'Q3', 'AUG': 'Q3', 'SEP': 'Q3',
        'OCT': 'Q4', 'NOV': 'Q4', 'DEC': 'Q4'
    }
    return quarters[month]

df_merged['YearOnly'] = df_merged['Year'].apply(lambda x: x[:4])
df_merged['Month'] = df_merged['Year'].apply(lambda x: x[5:])
df_merged['Quarter'] = df_merged['Month'].apply(month_to_quarter)

# Set the multi-index
df_merged.set_index(['YearOnly', 'Quarter', 'Month'], inplace=True)
df_merged.drop(columns=['Year'], inplace=True)

import plotly.io as pio
pio.renderers.default = 'iframe' # or 'colab' or 'iframe' or 'iframe_connected'␣
↪or 'sphinx_gallery'
import plotly.express as px
import plotly.graph_objects as go

if 'YearOnly' not in df_merged.columns:
    # If 'YearOnly' was previously set as index, reset it to column
    df_merged.reset_index(inplace=True)

df_merged['Timeline'] = df_merged['YearOnly'] + ' ' + df_merged['Quarter'] + '␣
↪' + df_merged['Month']

fig = go.Figure()

fig.add_trace(go.Scatter(x=df_merged['Timeline'], y=df_merged['Unemployment␣
↪Percent Change'], mode='lines', name='Unemployment Percent Change'))

fig.add_trace(go.Scatter(x=df_merged['Timeline'], y=df_merged['Employment␣
↪Percent Change'], mode='lines', name='Employment Percent Change'))
```

```
fig.add_trace(go.Scatter(x=df_merged['Timeline'], y=df_merged['Inflation␣
↪Percent Change'], mode='lines', name='Inflation Percent Change'))

# Update plot layout
fig.update_layout(title='Economic Indicators of UK (2010-2021)',␣
↪xaxis_title='Timeline', yaxis_title='Percent Change', xaxis={'type':␣
↪'category'})
fig.show()
```

[12]:
```
import pandas as pd
import altair as alt
death_database = pd.read_csv("deaths20222023.csv")
death_database = death_database.loc[:, ["Week number", "Total deaths\nEngland␣
↪and Wales (2023)", "Total deaths\nEngland and Wales (2022)"]]
death_database = death_database.dropna()
death_database = death_database.rename(columns={"Total deaths\nEngland and␣
↪Wales (2023)": "Deaths 2023", "Total deaths\nEngland and Wales (2022)":␣
↪"Deaths 2022"})
d2_database = pd.read_csv("deaths20202021.csv")
d2_database = d2_database.loc[:, ["Week number", "Care home resident deaths,␣
↪all causes (2021)", "Care home resident deaths, all causes (2020)5"]]
d2_database = d2_database.set_index("Week number")
death_database["Deaths 2021"] = d2_database["Care home resident deaths, all␣
↪causes (2021)"].values
death_database["Deaths 2020"] = d2_database["Care home resident deaths, all␣
↪causes (2020)5"].values
def get_quarter(week_number):
    if 1 <= week_number <= 12:
        return "Q1"
    elif 13 <= week_number <= 25:
        return "Q2"
    elif 26 <= week_number <= 38:
        return "Q3"
    elif 39 <= week_number <= 52:
        return "Q4"
    else:
        return "Error: week number out of range (1-52)"
assert get_quarter(1) == "Q1"
assert get_quarter(15) == "Q2"
assert get_quarter(37) == "Q3"
assert get_quarter(51) == "Q4"
assert get_quarter(0) == "Error: week number out of range (1-52)"
assert get_quarter(100) == "Error: week number out of range (1-52)"
death_database["Quarter"] = death_database["Week number"].apply(get_quarter)
death_database = death_database.drop(labels="Week number", axis="columns")
```

```
# setting index twice because this way, the column "Week number" is␣
↪automatically removed
death_database = death_database.set_index("Quarter")
# this gets rid of the commas in the values
death_database[["Deaths 2023", "Deaths 2022", "Deaths 2021", "Deaths 2020"]] =␣
↪death_database[["Deaths 2023", "Deaths 2022", "Deaths 2021", "Deaths 2020"]].
↪replace(",", "", regex=True)
# this converts all the numbers from string to integer
death_database[["Deaths 2023", "Deaths 2022", "Deaths 2021", "Deaths 2020"]] =␣
↪death_database[["Deaths 2023", "Deaths 2022", "Deaths 2021", "Deaths 2020"]].
↪astype(int)
# aggregates the data for each quarter
deaths = death_database.groupby(level=0).sum()
# need to reset index because altair cannot plot using index values
deaths = deaths.reset_index()
gdp = pd.read_csv("gdp.csv")
gdp = gdp.rename(columns={'Title': 'Year', 'Gross Domestic Product: Quarter on␣
↪Quarter growth: CVM SA %': 'GDP Percent Change'})
gdp = gdp[gdp['Year'].between('2020 Q1', '2023 Q4')]
gdp["Quarter"] = gdp["Year"].str[-2:]
# this code creates a new column "Quarter"; the values were formatted like␣
↪"2020 Q1"
gdp['Year'] = gdp['Year'].str.replace(' Q1', '').str.replace(' Q2', '').str.
↪replace(' Q3', '').str.replace(' Q4', '')
# this removes Q1/Q2/Q3/Q4 from the end of years
gdp = gdp.pivot(index='Quarter', columns='Year', values='GDP Percent Change')
gdp.columns.name = None # gets rid of the 'Year' column
# 2023 Q4 has an NaN value because that information was not known when this␣
↪datasheet was released
gdp = gdp.rename(columns={"2023": "GDP 2023", "2022": "GDP 2022", "2021": "GDP␣
↪2021", "2020": "GDP 2020"})
# this converts all the numbers from string to integer
gdp[["GDP 2020", "GDP 2021", "GDP 2022", "GDP 2023"]] = gdp[["GDP 2020",
                                                  "GDP 2021", "GDP 2022", "GDP␣
↪2023"]].astype(float)
gdp = gdp.reset_index()
# testing deaths dataset
assert deaths.loc[0, "Deaths 2023"] == 39064
assert deaths.loc[1, "Deaths 2022"] == 31316
assert deaths.loc[2, "Deaths 2021"] == 29537
assert deaths.loc[3, "Deaths 2020"] == 37329
# testing gdp dataset
assert gdp.loc[0, "GDP 2023"] == 0.3
assert gdp.loc[1, "GDP 2022"] == 0.1
assert gdp.loc[2, "GDP 2021"] == 1.7
assert gdp.loc[3, "GDP 2020"] == 1.4
```

[8]:
```
# Plotting 2020 data
deaths2020 = alt.Chart(deaths).mark_line().encode(y=alt.Y("Deaths 2020").
 ↪scale(domain=(0,65000)), x='Quarter')
deaths2020 = deaths2020.properties(width=400, height=300, title="2020 Death␣
 ↪Count (per quarter)")
gdp2020 = alt.Chart(gdp).mark_line().encode(y=alt.Y("GDP 2020").
 ↪scale(domain=(-22,18)), x='Quarter')
gdp2020 = gdp2020.properties(width=400, height=300, title="2020 GDP (per␣
 ↪quarter)")
line = alt.Chart().mark_rule().encode(y=alt.datum(0))
chart2020 = deaths2020 | (gdp2020 + line)
chart2020
```

[8]: alt.HConcatChart(…)

[9]:
```
# Plotting 2021 data
deaths2021 = alt.Chart(deaths).mark_line().encode(y=alt.Y("Deaths 2021").
 ↪scale(domain=(0,45000)), x='Quarter')
deaths2021 = deaths2021.properties(width=400, height=300, title="2021 Death␣
 ↪Count (per quarter)")
gdp2021 = alt.Chart(gdp).mark_line().encode(y=alt.Y("GDP 2021").
 ↪scale(domain=(-2,8.5)), x='Quarter')
gdp2021 = gdp2021.properties(width=400, height=300, title="2021 GDP (per␣
 ↪quarter)")
line = alt.Chart().mark_rule().encode(y=alt.datum(0))
chart2021 = deaths2021 | (gdp2021 + line)
chart2021
```

[9]: alt.HConcatChart(…)

[10]:
```
# Plotting 2022 data
deaths2022 = alt.Chart(deaths).mark_line().encode(y=alt.Y("Deaths 2022").
 ↪scale(domain=(0,40000)), x='Quarter')
deaths2022 = deaths2022.properties(width=400, height=300, title="2022 Death␣
 ↪Count (per quarter)")
gdp2022 = alt.Chart(gdp).mark_line().encode(y=alt.Y("GDP 2022").
 ↪scale(domain=(-0.2,0.6)), x='Quarter')
gdp2022 = gdp2022.properties(width=400, height=300, title="2022 GDP (per␣
 ↪quarter)")
line = alt.Chart().mark_rule().encode(y=alt.datum(0))
chart2022 = deaths2022 | (gdp2022 + line)
chart2022
```

[10]: alt.HConcatChart(…)

[11]:
```
# Plotting 2023 data
deaths2023 = alt.Chart(deaths).mark_line().encode(y=alt.Y("Deaths 2023").
 ↪scale(domain=(0,41000)), x='Quarter')
deaths2023 = deaths2023.properties(width=400, height=300, title="2023 Death␣
 ↪Count (per quarter)")
gdp2023 = alt.Chart(gdp).mark_line().encode(y=alt.Y("GDP 2023").
 ↪scale(domain=(-0.2,0.4)), x='Quarter')
gdp2023 = gdp2023.properties(width=400, height=300, title="2023 GDP (per␣
 ↪quarter)")
line = alt.Chart().mark_rule().encode(y=alt.datum(0))
chart2023 = deaths2023 | (gdp2023 + line)
chart2023
```

[11]: alt.HConcatChart(…)

[ ]:
```
```