INFO 2950 Final Project - Phase IV

Janice Shen (js3678), Khai Xin Kuan (kk996), Sandy Lin (sl2534), David Park (yp358)

Table of Content

1. Introduction
2. Data Description and Cleaning
3. Preregistration Statements
4. Data Analysis
5. Evaluation of significance
6. Conclusion
7. Limitations
8. Acknowledgements and Bibliography

Introduction REVISE INTRO AFTER FINDINGS

A common interest in both music and finance led the four of us to explore a unique intersection between these fields. Specifically, we asked: could individual music preferences be related to broader economic trends in the United States? Inspired by prior research that suggested promising correlations between music sentiment and market performance (e.g., weekly equity returns), we decided to investigate further. While excited by the potential connections, our initial data analysis revealed only weak correlations between music sentiment and economic indicators. This prompted us to refine our approaches and think about linear regression models or multivariate regressions we could run, hoping to reveal hidden trends. Thus we have done two hypothesis: one to reveal a strong inverse relationship between the valence of popular music and US GDP and the other showed unemployment showed a marginal positive effect on danceability, while other variables had negligible relationships.

Research Questions:

1. **Can the valence of the general population's music preference be predicted by economics and socioeconomic indicators such as the unemployment rate, stocks, federal funds rate, stock returns, birth rate and incarceration rate?**
    - *Valence* represents the spectrum of emotions in music, ranging from negative (0) to positive (1).
    - **Goal:** We sought to determine whether broader socio-economic conditions influence the general emotional tone of music that people listen to.
2. **Can we predict the danceability of the general population's music preference based on key economic and socioeconomic variables such as death rate, recession, and unemployment rate?**
    - *Danceability* measures how suitable a track is for dancing, ranging from 0.0 (least danceable) to 1.0 (most danceable).
    - **Goal:** We explored whether this characteristic, often linked to musical enjoyment, also showed any connections to economic conditions.

Initial Findings and Methodological Refinement:

Our original analysis included all songs in our dataset and yielded extremely weak correlations. We hypothesized that the inclusion of a large volume of less popular songs may have distorted the results, as these songs might not accurately reflect public sentiment or consumption patterns. To address this, we filtered out less popular songs using a *popularity* metric (a scale from 0 to 100, where 100 indicates the most popular songs). We created a cutoff of 70, focusing on the songs with popularity greater than 70 in our analysis to better capture the music that

resonates with the general public. But when the correlation was still weak, we resulted to web scrape from Billboard top 100 and had more data points to run our current analysis.

Data Description and Cleaning

Part 1: Data Description

What are the observations (rows) and the attributes (columns)?

Observations: Each row represents a song or an economic indicator for a specific time period. In the music dataset (musicdata), rows represent individual songs, while in the economic datasets (undata, econdata, us_gdp_data, us_entertainmentgdp_data), rows represent time intervals (e.g., months or quarters).
Attributes: In the **music dataset**, the columns include variables like **valence** (a measure of positivity in music), **danceability**, **popularity**, **year**, and other sentiment-related metrics. For the **economic datasets** columns include **unemployment rate**, **GDP**, **federal funds rate**, and **Stock Returns**. For the **socio-economic datasets** columns include **incarceration rate**, **birth rates**, and **death rates**.

Attribute Definitions:

Year: The year that corresponds to rest of the following variables present.

Music sentiments: a dimension of how one feels about the given music.

- danceability: Describes how suitable a track is for dancing, based on a combination of musical elements, including tempo, rhythm stability, beat strength, and overall regularity. Scale of 0.00 to 1.00
- popularity: A metric that measures how popular an artist or song is on the platform. Scale of 0 to 100.

Economic datasets:

- Unemployment Rate: The unemployment rate of that year.
- GDP: The Gross Domestic Product of that year.
- Federal Funds Rate: Also known as the interest rate, the interest rate that banks charge each other to borrow money overnight
- S&P500: The Standard and Poor's 500 is an index tracking the stock performance of 500 of the largest companies listed on US stock exchanges.

Social/Socio-economical Variables:

- Recession: Dummy variable of which is 1 if there was a recession recorded that year. 0 if not.
- Election: Dummy variable of which is 1 if there was an election that year. 0 if not.
- State_prisons: The imprisonment rates of state prisons at that year.
- fertility: Fertility rate, total (births per woman).
- Death Rate: Age-adjusted death rates (deaths per 100,000) per year (age-adjusted death rate means death rate if age composition of the population didn't change from year to year).

Why was this dataset created?

The **music sentiment datasets** were created to explore the relationship between musical attributes (such as valence, danceability, and popularity) and factors like popularity or music trends over time. The **economic datasets** were generated by governmental agencies and prestigious universities like the Federal Reserve or NYU Stern to monitor key economic indicators for research, policy analysis, and public transparency. The **socio-economic datasets** where compiled by government organizations or nonprofits for the betterment of society, such as the Prison Policy Initiative aiming to "uses research, advocacy, and organizing to dismantle mass incarceration;" the National Center under the Health Statistics (NCHS) under the US Centers for Disease Control and Prevention, to help public research on health; or the World Bank Group that has different "global partnership fighting poverty worldwide through sustainable solutions."

Who funded the creation of the dataset?

The **music sentiment dataset** were compiled by independent researchers, such as Caleb Elgut, likely with no explicit funding. These data sources rely on platforms like **Spotify**, **Billboard**, and **ARIA**, which collected the original data. The **economic datasets** were funded by the U.S. government through organizations like the **Federal Reserve** and the **Federal Reserve Bank of St. Louis (FRED)**, as part of their regular efforts to report and track national economic trends. The Stocks dataset is compiled by research done from **NYC Stern** and the socioeconomic datasets are gathered by non-part for government organizations: **Nation Center for Health Statistics**, **Prison Policy Initiative**

What processes might have influenced what data was observed and recorded and what was not?

In the **music datasets**, factors like availability on Spotify, scraping techniques, and modern-day popularity metrics may have influenced the data observed. For example, older songs that were popular in the 1990s may have lower popularity today, potentially skewing the results when comparing them to historical economic data. In the **economic datasets**, revisions in how economic indicators like GDP and unemployment are calculated, government adjustments, and data collection methodologies could have impacted the recorded data. Additionally, the time series nature of the data means there may be adjustments due to seasonal variations or changes in reporting practices. Social variables, such as birth rates and incarceration rates, were only available on a yearly basis due to the high cost of gathering the data, limiting the granularity of the dataset. Additionally, the reliance on platforms like Spotify and Billboard charts means the data may be biased towards mainstream trends and digital listening habits, excluding other platforms and niche groups.

What preprocessing was done, and how did the data come to be in the form that you are using?

- **Music datasets**: The data was scraped from platforms like **Spotify**. Preprocessing included cleaning missing values, and ensuring compatibility between different time periods in the analysis (there was no missing values when we loaded the files).
- **Economic datasets**: The raw data provided by the **Federal Reserve** and **NYU Stern** was merged and aligned with the music data for time-series analysis. Preprocessing involved cleaning missing values, ensuring consistent date formats, and aligning economic metrics with the appropriate historical periods for comparison with music data.
- **Socio-economic datasets**: The dataset was cleaned and aggregated to match the yearly format of available social and economic data. Preprocessing steps included handling missing values, normalizing data across variables, and potentially transforming metrics such as valence and danceability into averages by year. Furthermore, non-relevant variables may have been removed, and multicollinearity checks were conducted to ensure the usability of the dataset in regression models (in our.

If people are involved, were they aware of the data collection and if so, what purpose did they expect the data to be used for?

- **Music datasets**: The data originates from public platforms like **Spotify**, where users are generally aware that their listening habits are tracked. However, artists and listeners would likely be unaware of this specific usage of the data for correlating music preferences with economic trends.
- **Economic and socio-economic datasets**: Data on these indicators is collected by government agencies or non-profits mentioned above with the understanding that it will be publicly available for research, policy analysis, and economic forecasting. Individuals were not directly involved in the data collection process.

Where can your raw source data be found, if applicable?

1. **Music sentiment datasets**:
    - Caleb Elgut's dataset scraped from Spotify can be found on GitHub: [Spotify-LSTM Data](https://github.com/calebelgut/spotify-lstm/tree/main/Data)
2. **Economic datasets:**
    - Unemployment rate data from Jan 1948 – Sep 2024 is available through the Federal Reserve Economic Data (FRED) platform: [Unemployment Rate](https://fred.stlouisfed.org/series/UNRATE)

- Quarterly GDP from Jan 1997 – 2019: [Quarterly GDP](https://fred.stlouisfed.org/series/GDP)
- Yearly GDP of the Music/Entertainment Industry from 1997 – 2019: [Entertainment Industry GDP](https://fred.stlouisfed.org/series/USPRFRMSPRTMSMNGSP)
- Annual Returns on Investment of US Stocks (including dividends) from 1928-2023: [Stock Returns](https://pages.stern.nyu.edu/~adamodar/New_Home_Page/datafile/histretSP.html)

3. **Socio-Economic Datasets**:

- Mortality trends since 1900-2018 by NCHS: [Death Rate](https://www.cdc.gov/nchs/data-visualization/mortality-trends/index.htm)
- Fertility rate, total (births per woman) by the World Bank: [Birth Rate](https://data.worldbank.org/indicator/SP.DYN.TFRT.IN?locations=US)
- Recession generated by FRED, borrowing expterise from The National Bureau of Economic Research (NBER): [Recession](https://fred.stlouisfed.org/series/USREC)
- Incarceration data from non-profit and non-partisan Prison Policy Initative: [Incarceration](https://www.prisonpolicy.org/data/)
- election variable self-generated by assign 1 or 0 every 4 years (1 when had election and 0 when no election, more specific code in recession_and_election.ipynb.

P2 Cleaning

Overall Summary: From our feedback in Phase 2, we chose to web scrape directly from Billboard and match the Billboard song data with Spotify's sentiment data. This was done by using song titles and artist names as common identifiers. Variations in how song names or artist names were spelled were resolved to make sure the match was accurate. Economic data(eg. Fed fund rate, unemployment rate) was collected through different government sources and merged all economic data into one dataset. Additional control variables were collected to provide context to the economic variables(eg, birth rate, death rate, recession) or other factors influencing music trends. The web-scraping process could be found in billboard_scraping.ipynb.

Data Standardization:

- Ensuring consistent formats for song titles and artist names in billboard and spotify dataset
- Ensuring economic variables were comparable across different datasets.
- Handling Missing Data: Missing values were identified and addressed by either excluding records or using imputation methods depending on the context of the missing data and its potential impact on analysis.

Final Merged Dataset:

- After the cleaning and merging steps, a comprehensive, standardized dataset was created that included:
  - Billboard data (song rankings and deta)
  - Spotify sentiment data (emotional tone of songs)
  - Economic data (macroeconomic indicators)
  - Control variables (additional contextual factors).

After merging all of these variable in the file name final_clean_data

In [1]:
```
import pandas as pd
import numpy as np
import duckdb
import seaborn as sns
import matplotlib.pyplot as plt
import datetime
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
import statsmodels.api as sm
from statsmodels.api import OLS, add_constant
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

Final dataset that contains a comprehensive overview of all the music sentiment information, economic information, and additional control variable factors

In [2]:
```
final = pd.read_csv('final_clean_data.csv')
```

inspect the data and see the columns present in the table

In [3]:
```
final.head()
```

Out[3]:

	Unnamed: 0	danceability	valence	year	fedfundrate	UnemploymentRate	recession	election	US_GDP	year_
0	0	0.325	0.9130	1960	3.215833	5.541667	1	1	6.295971	196
1	1	0.533	0.1905	1960	3.215833	5.541667	1	1	6.295971	196
2	2	0.498	0.8660	1960	3.215833	5.541667	1	1	6.295971	196
3	3	0.447	0.3300	1960	3.215833	5.541667	1	1	6.295971	196
4	4	0.558	0.3030	1960	3.215833	5.541667	1	1	6.295971	196

Dropping index column not used in our analysis:

In [4]:
```
print(final.columns)
final.drop(columns=['year_1','Unnamed: 0'], axis=1, inplace=True)
print(final.columns)
final.head()
# print(final.shape)
```

```
Index(['Unnamed: 0', 'danceability', 'valence', 'year', 'fedfundrate',
       'UnemploymentRate', 'recession', 'election', 'US_GDP', 'year_1',
       'State_prisons', 'fertility', 'DeathRate', 'S&P500'],
      dtype='object')
Index(['danceability', 'valence', 'year', 'fedfundrate', 'UnemploymentRate',
       'recession', 'election', 'US_GDP', 'State_prisons', 'fertility',
       'DeathRate', 'S&P500'],
      dtype='object')
```

Out[4]:

	danceability	valence	year	fedfundrate	UnemploymentRate	recession	election	US_GDP	State_prisons	f
0	0.325	0.9130	1960	3.215833	5.541667	1	1	6.295971	12.153384	
1	0.533	0.1905	1960	3.215833	5.541667	1	1	6.295971	12.153384	
2	0.498	0.8660	1960	3.215833	5.541667	1	1	6.295971	12.153384	
3	0.447	0.3300	1960	3.215833	5.541667	1	1	6.295971	12.153384	
4	0.558	0.3030	1960	3.215833	5.541667	1	1	6.295971	12.153384	

Creating a DataFrame that aggregates and calculates the average values for each variable by year. This step aggregates our raw dataset to calculate average values for key economic and social variables per year. It ensures our analysis is aligned with the yearly format of the music sentiment data (valence).

In [5]:
```
final.corr()
```

Out[5]:

	danceability	valence	year	fedfundrate	UnemploymentRate	recession	election
danceability	1.000000	0.472345	0.223172	-0.004494	0.033631	-0.083117	0.011511
valence	0.472345	1.000000	-0.241945	0.135285	0.008075	0.044233	-0.000793
year	0.223172	-0.241945	1.000000	-0.445837	0.123558	-0.220598	0.045998
fedfundrate	-0.004494	0.135285	-0.445837	1.000000	0.030958	0.303395	-0.093547
UnemploymentRate	0.033631	0.008075	0.123558	0.030958	1.000000	0.127831	-0.054946
recession	-0.083117	0.044233	-0.220598	0.303395	0.127831	1.000000	-0.070893
election	0.011511	-0.000793	0.045998	-0.093547	-0.054946	-0.070893	1.000000
US_GDP	0.241861	-0.235329	0.987216	-0.338334	0.144402	-0.224206	0.034820
State_prisons	0.227177	-0.237632	0.961174	-0.458041	0.064358	-0.240894	0.030382
fertility	-0.192547	0.115248	-0.572126	-0.251714	-0.294991	0.155253	0.017238
DeathRate	-0.229379	0.234508	-0.988946	0.379010	-0.222950	0.191604	-0.034247
S&P500	0.005003	0.016849	-0.025894	0.044593	0.247344	-0.184744	-0.058050

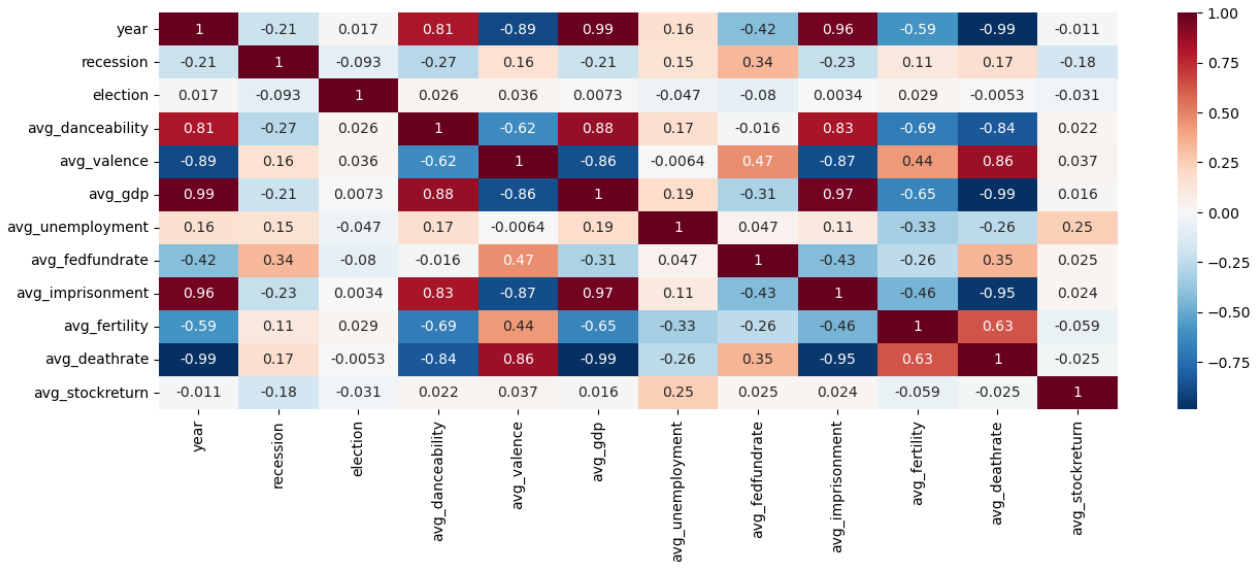
Created a data frame averaging all the variables to be yearly unit.

In [7]:
```
averagefinal = duckdb.sql("""
                                SELECT
                                    Year,
                                    ANY_VALUE(recession) AS recession,
                                    ANY_VALUE(election) AS election,
                                    AVG(danceability) AS avg_danceability,
                                    AVG(valence) AS avg_valence,
                                    AVG(US_GDP) AS avg_gdp,
                                    AVG(UnemploymentRate) AS avg_unemployment,
                                    AVG(fedfundrate) AS avg_fedfundrate,
                                    AVG(State_prisons) AS avg_imprisonment,
                                    AVG(fertility) AS avg_fertility,
                                    AVG(DeathRate) AS avg_deathrate,
                                    AVG("S&P500") AS avg_stockreturn
                                FROM
                                    final
                                GROUP BY
                                    Year
                                ORDER BY
                                    Year
                                """).df()

averagefinal.head()
print(averagefinal.shape)
```

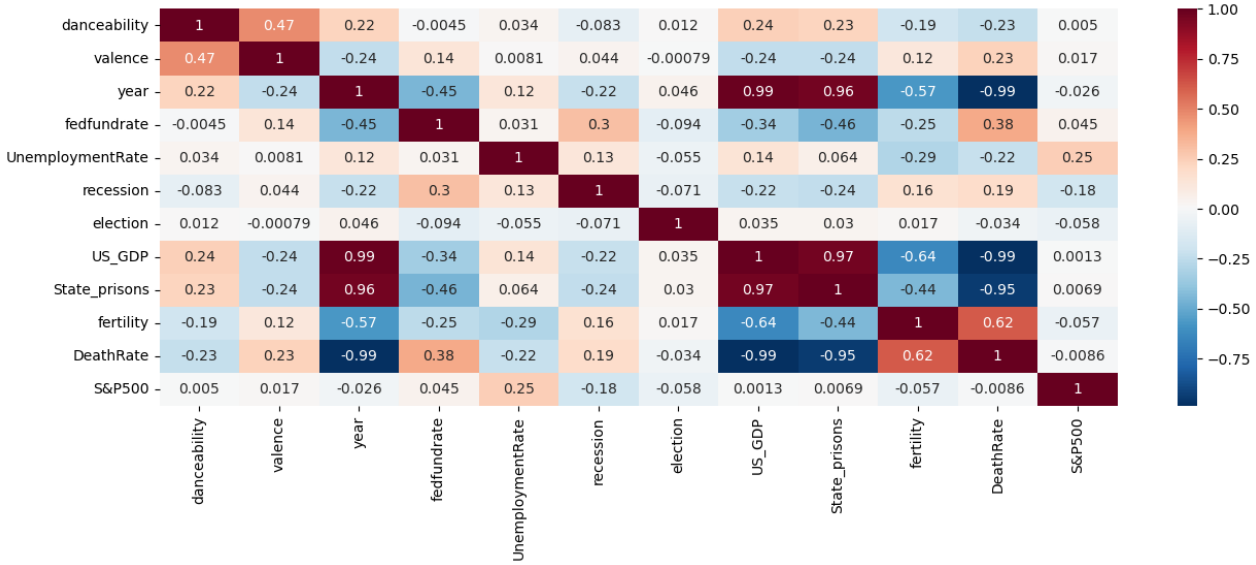
```
(56, 12)
```

Created a heatmap to project the correlation matrix:

In [8]:
```
fig, ax = plt.subplots(figsize=(15, 5))
heatmap = sns.heatmap(averagefinal.corr(numeric_only=True), cmap="RdBu_r", annot=True)
```



In [9]:
```
fig, ax = plt.subplots(figsize=(15, 5))
heatmap = sns.heatmap(final.corr(numeric_only=True), cmap="RdBu_r", annot=True)
```



Pre-registration Statement

Hypothesis 1: Music Valence vs. US GDP

The average valence of popular songs by year from 1960-2016 in the US has an inverse relationship with the corresponding US Gross Domestic Product (GDP) of the given year.

- H_0: The average valence of the top 100 Spotify songs per year has no relationship to the US GDP each year ($\beta = 0$).
- H_A: The average valence of the top 100 Spotify songs per year has an inverse relationship to the US GDP each year ($\beta < 0$).

We hypothesize that during periods of economic prosperity (higher GDP), cultural sentiments may reflect more subdued emotions in music, leading to lower valence scores. This relationship is supported by existing literature on the connection between societal well-being and cultural expressions, which indicates that wealthier societies may prioritize complexity or emotional depth in art over positivity.

To test the hypothesis that the average valence of popular songs is inversely related to GDP, we will aggregate the data by year to calculate annual averages for valence, GDP, and other economic variables. We will begin with

exploratory correlation analysis to understand initial relationships, followed by linear regression to model the relationship between valence and GDP, both individually and with additional controls (fertility rate, death rate, and federal funds rate). Log transformations will be applied to address potential nonlinear relationships. Model performance will be evaluated using metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), and residual plots will be used to validate assumptions. Finally, scatter plots with regression lines will visually interpret the relationships and provide robust insights into the connection between economic variables and music sentiment.

Hypothesis 2: Danceability vs Unemployment Rate

The average danceability of the most popular songs per year from 1960-2016 in the US has an inverse relationship with the corresponding US unemployment rate of the given year.

- H_0: The average danceability of the top 100 Spotify songs per year has no relationship to the US unemployment rate each year ($\beta = 0$).
- H_A: The average danceability of the top 100 Spotify songs per year has an inverse relationship to the US unemployment rate each year ($\beta < 0$).

We expect a statistically significant negative coefficient ($\beta < 0$) for unemployment, demonstrating an inverse relationship with danceability. Residual plots will be inspected to validate model assumptions, and sensitivity analyses will test for the robustness of the findings by adjusting control variables.

To test this hypothesis, we will conduct a multivariate linear regression using unemployment rate as the primary independent variable and average danceability of songs as the dependent variable. Additional economic and social variables (e.g., stock returns, election years, death rate, and recession indicators) will be included as covariates to control for confounding effects. Residual plots will be used to validate model assumptions, and interaction terms will be tested (e.g., stock returns and election) to assess whether the impact of these variables on danceability changes in election years. Visualizations, such as scatterplots and interaction effect plots, will illustrate the relationships and validate the regression model's insights. Finally, statistical metrics from the OLS model, such as p-values and R-squared, will determine the significance and strength of the relationships observed.

Hypothesis 1:

We first take the correlation of our variables. Computing correlations provides preliminary insights into the relationships between variables, such as whether GDP and valence are inversely related.

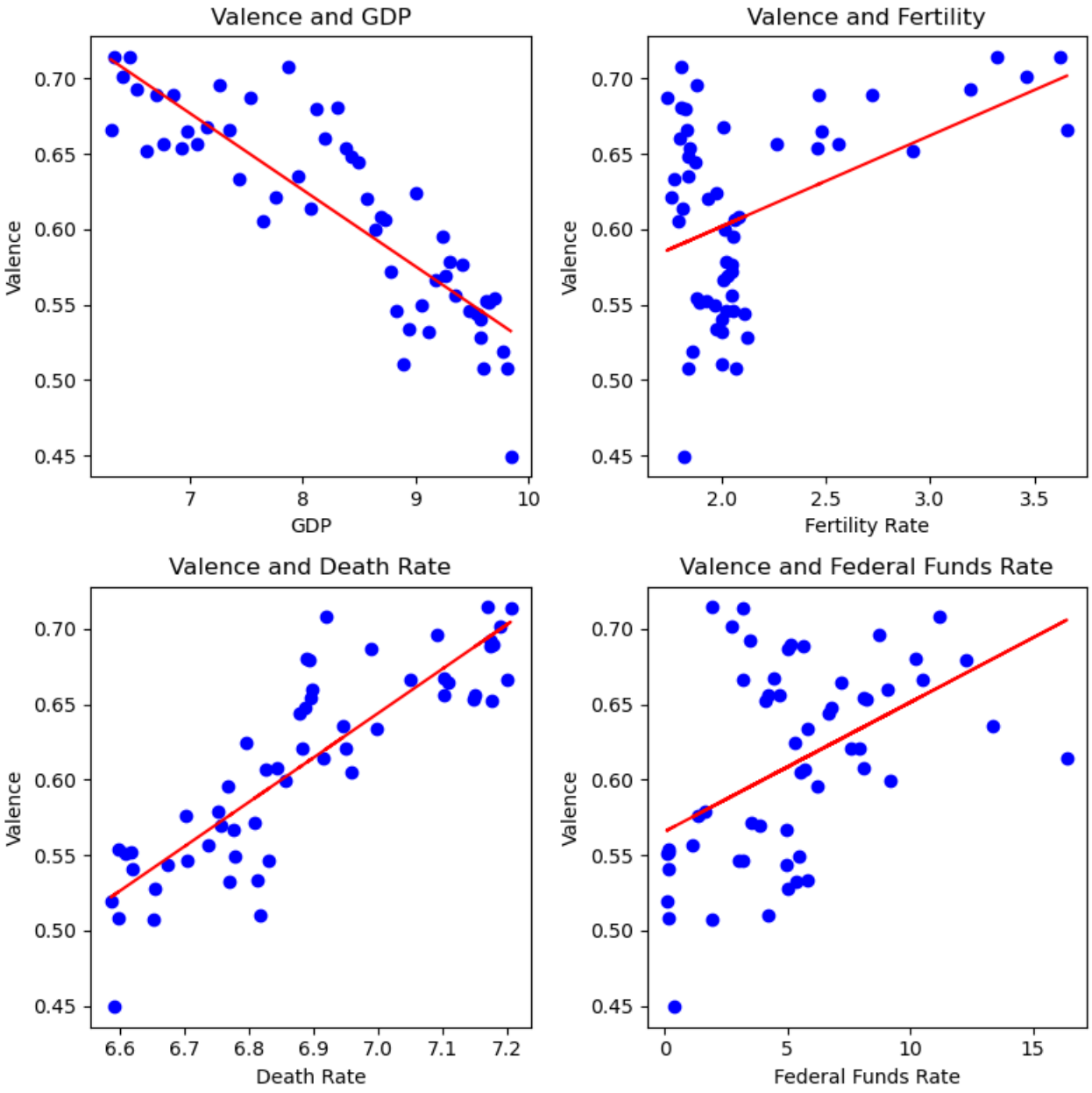
We then take a **linear regression**, **evaluate the regression fit**, **make a scatterplot for visualization**, and **explore other variables**.

- Linear regression: Running a simple linear regression estimates how average GDP influences valence. This allows us to test our hypothesis that valence decreases as GDP increases (β < 0).
- Calculating Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) evaluates how well the regression model fits the data. These metrics help measure prediction error.
- Plotting the fitted regression line on the scatter plot visually illustrates the relationship between GDP and valence, aiding interpretation.
- Evaluating other variables helps identify the relative importance of different economic and social variables in explaining valence trends.

In [10]:
```
# Create a 2x2 grid of subplots
fig, axs = plt.subplots(2, 2, figsize=(8, 8))

# --- GDP and Valence ---
X_gdp = averagefinal[["avg_gdp"]].values
y_gdp = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_gdp, y_gdp)
print("GDP Coeff:", round(model.coef_[0], 4))
print(f"Intercept: {model.intercept_:.4f}")
y_pred_gdp = model.predict(X_gdp)
```

```
axs[0, 0].scatter(X_gdp, y_gdp, color="blue")
axs[0, 0].plot(X_gdp, y_pred_gdp, color="red", label="Fitted line")
axs[0, 0].set_title("Valence and GDP")
axs[0, 0].set_xlabel("GDP")
axs[0, 0].set_ylabel("Valence")

# --- Fertility and Valence ---
X_fertility = averagefinal[["avg_fertility"]].values
y_fertility = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_fertility, y_fertility)
print("Fertility Coeff:", round(model.coef_[0], 4))
print(f"Intercept: {model.intercept_:.4f}")
y_pred_fertility = model.predict(X_fertility)

axs[0, 1].scatter(X_fertility, y_fertility, color="blue")
axs[0, 1].plot(X_fertility, y_pred_fertility, color="red", label="Fitted line")
axs[0, 1].set_title("Valence and Fertility")
axs[0, 1].set_xlabel("Fertility Rate")
axs[0, 1].set_ylabel("Valence")

# --- Death Rate and Valence ---
X_deathrate = averagefinal[["avg_deathrate"]].values
y_deathrate = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_deathrate, y_deathrate)
print("Death Rate Coeff:", round(model.coef_[0], 4))
print(f"Intercept: {model.intercept_:.4f}")
y_pred_deathrate = model.predict(X_deathrate)

axs[1, 0].scatter(X_deathrate, y_deathrate, color="blue")
axs[1, 0].plot(X_deathrate, y_pred_deathrate, color="red", label="Fitted line")
axs[1, 0].set_title("Valence and Death Rate")
axs[1, 0].set_xlabel("Death Rate")
axs[1, 0].set_ylabel("Valence")

# --- Federal Funds Rate and Valence ---
X_fedfundrate = averagefinal[["avg_fedfundrate"]].values
y_fedfundrate = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_fedfundrate, y_fedfundrate)
print("Fed Funds Rate Coeff:", round(model.coef_[0], 4))
print(f"Intercept: {model.intercept_:.4f}")
y_pred_fedfundrate = model.predict(X_fedfundrate)

axs[1, 1].scatter(X_fedfundrate, y_fedfundrate, color="blue")
axs[1, 1].plot(X_fedfundrate, y_pred_fedfundrate, color="red", label="Fitted line")
axs[1, 1].set_title("Valence and Federal Funds Rate")
axs[1, 1].set_xlabel("Federal Funds Rate")
axs[1, 1].set_ylabel("Valence")

# Adjust layout to avoid overlap
plt.tight_layout()

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

```
GDP Coeff: -0.0508
Intercept: 1.0326
Fertility Coeff: 0.0604
Intercept: 0.4810
Death Rate Coeff: 0.2948
Intercept: -1.4194
Fed Funds Rate Coeff: 0.0086
Intercept: 0.5656
```

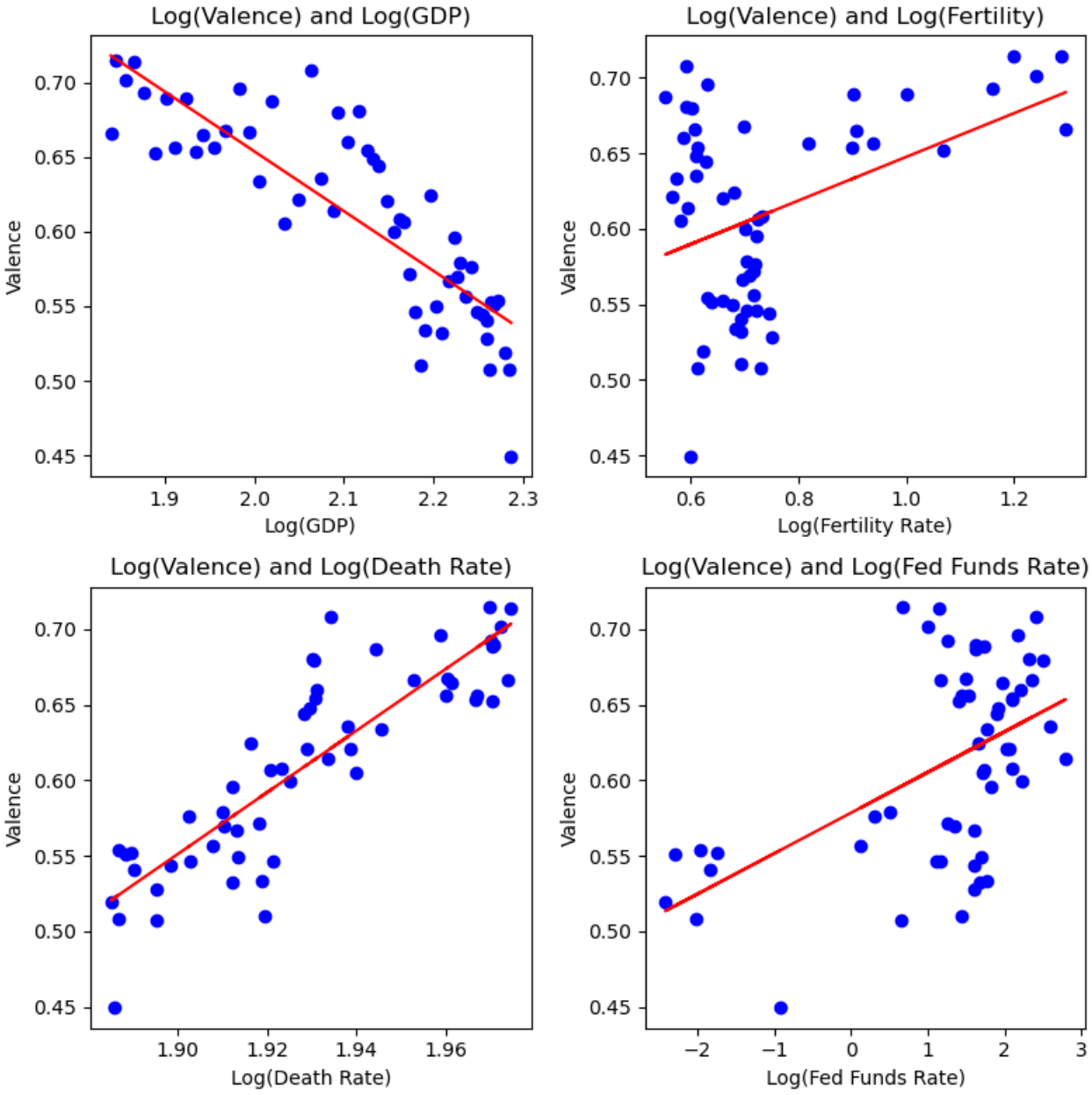


Since we wanted to make the regression more robust, we log-transformed GDP and other independent variables to account for nonlinear relationship.

In [11]:
```
fig, axs = plt.subplots(2, 2, figsize=(8, 8))

#Logged GDP and Valence
X_gdp = np.log(averagefinal[["avg_gdp"]].values)
y_gdp = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_gdp, y_gdp)
print("Log(GDP) Coeff:", round(model.coef_[0], 4))
print(f"Intercept: {model.intercept_:.4f}")
# y_pred_gdp = model.predict(X_gdp)

# mae_gdp = mean_absolute_error(y_gdp, y_pred_gdp)
# rmse_gdp = np.sqrt(mean_squared_error(y_gdp, y_pred))
# print(f"MAE for Log(GDP) model: {mae_gdp:.4f}")
# print(f"RMSE for Log(GDP) model: {rmse_gdp:.4f}\n")

axs[0, 0].scatter(X_gdp, y_gdp, color="blue")
axs[0, 0].plot(X_gdp, model.predict(X_gdp), color="red", label="Fitted line")
axs[0, 0].set_title("Log(Valence) and Log(GDP)")
axs[0, 0].set_xlabel("Log(GDP)")
```

```
axs[0, 0].set_ylabel("Valence")


#Logged Fertility and Valence
X_fertility = np.log(averagefinal[["avg_fertility"]].values)
y_fertility = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_fertility, y_fertility)
print("Log(Fertility) Coeff:", round(model.coef_[0], 4))
print(f"Intercept: {model.intercept_:.4f}")
# y_pred_fertility = model.predict(X_fertility)

# mae_fertility = mean_absolute_error(y_fertility, y_pred_fertility)
# rmse_fertility = np.sqrt(mean_squared_error(y_fertility, y_pred_fertility))
# print(f"MAE for Log(Fertility) model: {mae_fertility:.4f}")
# print(f"RMSE for Log(Fertility) model: {rmse_fertility:.4f}\n")

axs[0, 1].scatter(X_fertility, y_fertility, color="blue")
axs[0, 1].plot(X_fertility, model.predict(X_fertility), color="red", label="Fitted line")
axs[0, 1].set_title("Log(Valence) and Log(Fertility)")
axs[0, 1].set_xlabel("Log(Fertility Rate)")
axs[0, 1].set_ylabel("Valence")


#Logged Death Rate and Valence
X_death = np.log(averagefinal[["avg_deathrate"]].values)
y_death = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_death, y_death)
print("Log(Death Rate) Coeff:", round(model.coef_[0], 4))
print(f"Intercept: {model.intercept_:.4f}")
# y_pred_death = model.predict(X_death)

# mae_death = mean_absolute_error(y_death, y_pred_death)
# rmse_death = np.sqrt(mean_squared_error(y_death, y_pred_death))
# print(f"MAE for Log(Death Rate) model: {mae_death:.4f}")
# print(f"RMSE for Log(Death Rate) model: {rmse_death:.4f}\n")

axs[1, 0].scatter(X_death, y_death, color="blue")
axs[1, 0].plot(X_death, model.predict(X_death), color="red", label="Fitted line")
axs[1, 0].set_title("Log(Valence) and Log(Death Rate)")
axs[1, 0].set_xlabel("Log(Death Rate)")
axs[1, 0].set_ylabel("Valence")

#Logged Federal Funds Rate and Valeence
X= np.log(averagefinal[["avg_fedfundrate"]].values)
y= averagefinal["avg_valence"].values
model = LinearRegression().fit(X, y)
print("Log(Fed Funds Rate) Coeff:", round(model.coef_[0], 4))
print(f"Intercept: {model.intercept_:.4f}")
# y_pred_fed = model.predict(X_fed)

# mae_fed = mean_absolute_error(y_fed, y_pred_fed)
# rmse_fed = np.sqrt(mean_squared_error(y_fed, y_pred_fed))
# print(f"MAE for Log(Fed Fund Rate) model: {mae_fed:.4f}")
# print(f"RMSE for Log(Fed Fund Rate) model: {rmse_fed:.4f}\n")

axs[1, 1].scatter(X, y, color="blue")
axs[1, 1].plot(X, model.predict(X), color="red", label="Fitted line")
axs[1, 1].set_title("Log(Valence) and Log(Fed Funds Rate)")
axs[1, 1].set_xlabel("Log(Fed Funds Rate)")
axs[1, 1].set_ylabel("Valence")

plt.tight_layout()
```

```
Log(GDP) Coeff: -0.4007
Intercept: 1.4551
Log(Fertility) Coeff: 0.1446
Intercept: 0.5031
Log(Death Rate) Coeff: 2.0378
Intercept: -3.3204
Log(Fed Funds Rate) Coeff: 0.0269
Intercept: 0.5785
```



In [ ]:
```
def generate_residual_plot(pred, resid):
    sns.scatterplot(x=pred, y=resid, marker="o")
    plt.axhline(y=0, color="black", linestyle='--')
    plt.xlabel("Predicted Values")
    plt.ylabel("Residuals")
    plt.title("Residual Plot")
    plt.show()


fig, axs = plt.subplots(2, 2, figsize=(8, 8))

# GDP and Valence
X_gdp = averagefinal[["avg_gdp"]].values
y_gdp = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_gdp, y_gdp)
```

```
y_pred_gdp = model.predict(X_gdp)


residuals_gdp = y_gdp - y_pred_gdp


axs[0, 0].scatter(y_pred_gdp, residuals_gdp, color="blue")
axs[0, 0].axhline(y=0, color='red', linestyle='--')  # Add a line at zero
axs[0, 0].set_title("Residuals: Valence vs GDP")
axs[0, 0].set_xlabel("Predicted Valence (GDP)")
axs[0, 0].set_ylabel("Residuals")

# Fertility and Valence
X_fertility = averagefinal[["avg_fertility"]].values
y_fertility = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_fertility, y_fertility)
y_pred_fertility = model.predict(X_fertility)


residuals_fertility = y_fertility - y_pred_fertility


axs[0, 1].scatter(y_pred_fertility, residuals_fertility, color="blue")
axs[0, 1].axhline(y=0, color='red', linestyle='--')  # Add a line at zero
axs[0, 1].set_title("Residuals: Valence vs Fertility")
axs[0, 1].set_xlabel("Predicted Valence (Fertility)")
axs[0, 1].set_ylabel("Residuals")

# Death Rate and Valence
X_deathrate = averagefinal[["avg_deathrate"]].values
y_deathrate = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_deathrate, y_deathrate)
y_pred_deathrate = model.predict(X_deathrate)


residuals_deathrate = y_deathrate - y_pred_deathrate


axs[1, 0].scatter(y_pred_deathrate, residuals_deathrate, color="blue")
axs[1, 0].axhline(y=0, color='red', linestyle='--')  # Add a line at zero
axs[1, 0].set_title("Residuals: Valence vs Death Rate")
axs[1, 0].set_xlabel("Predicted Valence (Death Rate)")
axs[1, 0].set_ylabel("Residuals")

# Federal Funds Rate and Valence
X_fedfundrate = averagefinal[["avg_fedfundrate"]].values
y_fedfundrate = averagefinal["avg_valence"].values
model = LinearRegression().fit(X_fedfundrate, y_fedfundrate)
y_pred_fedfundrate = model.predict(X_fedfundrate)


residuals_fedfundrate = y_fedfundrate - y_pred_fedfundrate


axs[1, 1].scatter(y_pred_fedfundrate, residuals_fedfundrate, color="blue")
axs[1, 1].axhline(y=0, color='red', linestyle='--')  # Add a line at zero
axs[1, 1].set_title("Residuals: Valence vs Federal Funds Rate")
axs[1, 1].set_xlabel("Predicted Valence (Fed Funds Rate)")
axs[1, 1].set_ylabel("Residuals")


plt.tight_layout()

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

Finally, we run an OLS regression. Running an OLS regression with statistical output provides detailed metrics, such as p-values and confidence intervals, which are crucial for hypothesis testing.

In [ ]:
```
X = averagefinal[["avg_gdp"]]
y = averagefinal["avg_valence"]

X = sm.add_constant(X)

model = sm.OLS(y, X).fit()

print(model.summary())
```

Hypothesis 2:

We define the independent variables (input_vars) including unemployment rate, election indicator, death rate, stock return, and recession, as they are key economic and social indicators that might influence music sentiment. The dependent variable (avg_valence) represents the valence of music. A multivariate linear regression model is fitted to estimate the relationship between these variables and valence. Printing the coefficients helps us interpret the direction and magnitude of each variable's influence, while the intercept provides the baseline valence when all independent variables are zero.

In [ ]:
```
input_vars = ["avg_unemployment", "election", "avg_deathrate", "avg_stockreturn", "recession"]
X = averagefinal[input_vars]  # The independent variables (features)
y = averagefinal["avg_danceability"]  # The dependent variable (target)

# Initialize and fit the linear regression model
model = LinearRegression().fit(X, y)

# Print the coefficients for each variable
for var_name, var_coef in zip(input_vars, model.coef_):
    print(f"{var_name}: {var_coef:.2f}")

# Print the intercept of the model
print(f"Intercept: {model.intercept_:.2f}")
```

In [ ]:
```
X_train= averagefinal[input_vars]
train_predictions = model.predict(X_train)
```

We calculate residuals (the difference between actual and predicted valence) to assess the goodness of fit of the regression model. A residual plot is generated to visualize the pattern of residuals against predicted values. This step is crucial for checking model assumptions like homoscedasticity (constant variance of errors) and identifying potential model issues like nonlinearity or outliers.

In [ ]:
```
residuals = averagefinal["avg_danceability"] - train_predictions
residuals.head()
```

In [ ]:
```
def generate_residual_plot(pred, resid):
    sns.scatterplot(x= pred, y=resid, marker="o")
    plt.axhline(y=0 ,color="black")
    plt.xlabel("Predicted Danceability")
    plt.ylabel("Residuals")
    plt.title("Residual Plot")
    plt.show()
generate_residual_plot(train_predictions, residuals)
```

Using statsmodels, we run an Ordinary Least Squares (OLS) regression to validate the linear regression results and provide more detailed statistical output, including p-values and R-squared values. This step ensures the statistical robustness of our findings and identifies which predictors are significant contributors to the model.

In [ ]:
```
Xmar =  averagefinal[[ "avg_unemployment", "election", "avg_deathrate", \
                      "avg_stockreturn", "recession"]]
ymar = averagefinal[["avg_danceability"]]

Xmar = sm.add_constant(Xmar)
est = sm.OLS(ymar, Xmar).fit()
```

```
print('Multivar OLS Regression for Danceability:')
print(est.summary())
```

We introduce an interaction term between avg_stockreturn and election to examine if the relationship between stock returns and valence changes based on election years. This is significant because elections might amplify or mitigate the effects of stock returns on music sentiment. An interaction plot is created to visualize how the predicted valence changes with stock returns across different election scenarios.

In [ ]:
```
averagefinal['interaction'] = averagefinal['avg_stockreturn'] * averagefinal['election']

# Define independent variables (including interaction) and dependent variable
X = averagefinal[['avg_stockreturn', 'election', 'interaction']]
y = averagefinal['avg_danceability']

# Add a constant to the independent variables
X = sm.add_constant(X)

# Fit the OLS regression model
interaction_model = sm.OLS(y, X).fit()

election_levels = [0, 1]
stockreturn_range = np.linspace(averagefinal['avg_stockreturn'].min(), averagefinal['avg_stockretur

plot_data = pd.DataFrame({
    'avg_stockreturn': np.tile(stockreturn_range, len(election_levels)),
    'election': np.repeat(election_levels, len(stockreturn_range))
})
plot_data['interaction'] = plot_data['avg_stockreturn'] * plot_data['election']
plot_data = sm.add_constant(plot_data)

# Predict valence
plot_data['predicted_danceability'] = interaction_model.predict(plot_data)

# Plot the interaction effect
plt.figure(figsize=(8, 6))
sns.lineplot(
    data=plot_data,
    x='avg_stockreturn',
    y='predicted_danceability',
    hue='election'
)
plt.title("Interaction Plot: Election and Avg Stock Return on Danceability")
plt.xlabel("Average Stock Return")
plt.ylabel("Predicted Danceability")
plt.legend(title="Election")
plt.show()
```

Evaluation of Significance

The results for Hypothesis 1 indicate a statistically significant inverse relationship between the valence of popular songs and the US GDP. The OLS regression results show an R-squared value of 0.748, indicating that approximately 74.8% of the variation in valence is explained by the model. The p-value for the avg_gdp coefficient is well below 0.05, confirming the significance of the negative relationship ($\beta = -0.0508$, $p < 0.001$). The scatterplots for both linear and log-transformed variables reinforce this relationship, as the negative slopes illustrate an inverse trend. The residual plot for the log-transformed model shows no significant patterns, suggesting a well-fitted model. The MAE of 0.1909 and RMSE of 0.0591 indicate reasonable predictive accuracy. Together, these findings support rejecting the null hypothesis in favor of the alternative: the valence of popular songs has an inverse relationship with US GDP.

For Hypothesis 2, the regression coefficients suggest that unemployment and related variables have varying impacts on music danceability. Specifically, the coefficients for avg_unemployment ($\beta = 0.01$) is positive, while avg_deathrate ($\beta = -0.18$), avg_stockreturn ($\beta = -0.00$) and recession ($\beta = -0.01$) show negligible or inverse relationships. Despite these findings, the OLS regression model for this hypothesis yielded lower explanatory power than Hypothesis 1, as indicated by a higher dispersion in residuals. Although avg_unemployment has a marginal positive relationship, its

effect size is small, making it less practically significant. Given these results, there is insufficient evidence to reject the null hypothesis that unemployment rates and danceability have no relationship.

Conclusions

For Hypothesis 1, the analysis reveals a strong inverse relationship between the valence of popular music and US GDP, with 74.8% of the variation explained by the model ($R^2 = 0.748$). The significant negative coefficient for GDP ($\beta = -0.0508$, $p < 0.001$) confirms that broader economic conditions, particularly GDP, influence the emotional tone of music, supporting the hypothesis that economic indicators can predict valence.

For Hypothesis 2, while unemployment showed a marginal positive effect on danceability ($\beta = 0.01$), other variables had negligible or inverse relationships. The model's low explanatory power and high residual dispersion suggest insufficient evidence to establish a significant link between unemployment and music preferences. Overall, the findings highlight GDP as a key economic predictor of music valence, while unemployment's influence remains unclear.

Limitations

- **Limitation of Yearly Aggregation**: Our data is aggregated by year rather than month and year due to limitations in the availability of social variables such as birth rate, death rate, and incarceration rate on a monthly basis. This temporal limitation significantly impacts our ability to account for **lagging variables**, meaning that we cannot assess the delayed impact of economic or social events on music trends. For instance, we are unable to analyze how a recession in one month may influence the music released or consumed in subsequent months. Similarly, this yearly aggregation restricts us from capturing the effects of short-term, time-sensitive events, such as elections or other socio-political shifts, which may play a substantial role in shaping music trends. As a result, our findings may overlook nuanced relationships between social variables and music, leading to broader and less precise conclusions.
- **Limited Data Points**: we only have 55 datapoints as a result of averaging to yearly units.
- **Issues with Multicollinearity**: Our data analysis faces challenges related to multicollinearity within the regression models, particularly visible through the Variance Inflation Factor (VIF). Multicollinearity occurs when independent variables in a regression model are highly correlated, leading to unreliable coefficient estimates and difficulty in determining the individual contribution of each variable. For example, social and economic variables like GDP growth and unemployment rate may show a high degree of correlation, complicating our ability to separate their independent effects on music sentiment and trends. This statistical issue could result in misleading interpretations of our data and reduce the robustness of our findings.
- **Imperfect Representation of Human Emotions**: The valence and danceability metrics from Spotify data, while valuable proxies, do not comprehensively encapsulate the complexity of human emotions. Valence measures perceived positivity of a track, and danceability reflects rhythm and beat, but neither metric can fully capture nuanced emotional responses, such as melancholy or nostalgia, that music can evoke. This oversimplification may lead to discrepancies between the analyzed data and the actual emotional experiences of listeners. Consequently, conclusions drawn about how social and economic factors influence music sentiment may miss subtleties and fail to accurately reflect the broader spectrum of human emotional expression.
- **Limited Representativeness of Billboard and Spotify Data**: The reliance on Billboard's top charts and Spotify data presents another significant limitation in our study. These platforms, while widely recognized, are not fully representative of the diverse listening habits of the US population. Billboard charts primarily reflect popular music consumption, which may disproportionately highlight mainstream trends, while Spotify data is limited to users of the platform and does not account for listeners on other platforms such as Apple Music, YouTube, or traditional radio. This lack of inclusivity could introduce biases, excluding niche or underrepresented genres and demographics, and ultimately skew our results to favor the preferences of certain groups rather than the population as a whole.

Acknowledgements and Bibliography

Think of this as your Problem 0: what tools, data, or resources did you use that you should cite? You may format this section with any citation style. rubric: Clearly marked and formatted. The group thoughtfully engages with relevant literature on their topic in order to motivate and/or contextualize their findings. All external sources are properly cited in the bibliography and with appropriate in-text citations.

Plot 2 y variables in a graph: https://stackoverflow.com/questions/55654500/seaborn-plot-with-second-y-axis Seaboard color palette: https://seaborn.pydata.org/tutorial/color_palettes.html Fix the size of heat map:https://scales.arabpsychology.com/stats/how-to-adjust-the-size-of-heatmaps-in-seaborn/