MARVEL CHARACTER REPORT

September 2024

Presented To

MARVEL

Presented By

SIVASURIYAN M

Documentation for Marvel Character Dataset Exploratory Data Analysis (EDA)

1. Project Overview:

The purpose of this project is to perform an Exploratory Data Analysis (EDA) on a dataset of Marvel comic characters. The dataset contains detailed information about the characters, including their names, alignment (Good, Bad, Neutral), number of appearances, first appearance date, gender, and more.

This analysis aims to gain insights into:

- Character popularity (based on appearances).
- Gender representation over time.
- The relationship between character alignment and survival status.
- Character debuts across different decades.

We use Python libraries like Pandas, Seaborn, Matplotlib to visualize, and explore the data.

2. Dataset Information:

The dataset includes the following columns:

- page_id: A unique ID for each character entry.
- name: Name of the character.
- urlslug: A unique identifier for the character's web URL.
- ID: Identity type (Public, Secret, No Dual Identity).
- ALIGN: Character alignment (Good, Bad, Neutral).
- EYE: Eye color of the character.
- HAIR: Hair color of the character.
- SEX: Character's gender.
- GSM: Gender or sexual minority status.

- ALIVE: Whether the character is alive or deceased.
- APPEARANCES: Number of comic book appearances.
- FIRST APPEARANCE: Date of first appearance.
- Year: The year of first appearance.

3. Key Steps in the Analysis:

3.1 Data Preprocessing:

- **Missing Data Handling**: Missing values in important columns like ALIGN, SEX, and APPEARANCES were filled with placeholders like 'Unknown' or imputed based on the data context.
- **Type Conversion**: Columns like Year and APPEARANCES were converted to numeric data types, handling invalid or missing entries using pd.to_numeric().

Example:

Python

```
# Convert 'Year' to numeric and handle missing values
data['Year'] = pd.to_numeric(data['Year'], errors='coerce')
data = data.dropna(subset=['Year'])  # Remove rows with missing Year values
data['Year'] = data['Year'].astype(int)
```

3.2 Filtering and Visualization:

We used Streamlit for creating an interactive dashboard, allowing users to filter data based on character alignment, gender, and explore various visualizations.

- **Sidebar Filters**:   - Character alignment filter: Users can filter by Good, Bad, Neutral, or Unknown alignment.   - Gender filter: Users can filter based on gender (Male, Female, Unknown).

Example:

Python

```
# Sidebar filters
alignment_filter = st.sidebar.multiselect('Filter by Alignment', options=data['ALIGN'].unique(), default=data['ALIGN'].unique())
gender_filter = st.sidebar.multiselect('Filter by Gender', options=data['SEX'].unique(), default=data['SEX'].unique())

# Filtered data based on user selection
filtered_data = data[(data['ALIGN'].isin(alignment_filter)) & (data['SEX'].isin(gender_filter))]
```

4. Visualizations and Insights:

4.1 Alignment vs. Appearances Boxplot:

This visualization shows how character alignment (Good, Bad, Neutral) correlates with the number of comic book appearances. A boxplot was used to visualize this relationship.

Python

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='ALIGN', y='APPEARANCES', data=filtered_data)
plt.title('Alignment vs. Number of Appearances')
plt.xticks(rotation=45)
```

4.2 Gender Representation Over Decades:

A stacked bar chart shows the gender distribution of characters across different decades. This helps to track the representation of male and female characters over time.

Python

```
# Gender distribution over decades
gender_decade = pd.crosstab(filtered_data['Year'] // 10 * 10, filtered_data['SEX'])
```

```
# Plot the stacked bar chart for gender distribution over decades

plt.figure(figsize=(12, 6))

gender_decade.plot(kind='bar', stacked=True, color=['lightblue', 'pink', 'purple'])

plt.title('Gender Representation Over Decades')
```

4.3 Top 10 Most Popular Characters by Appearances:

This bar chart visualizes the top 10 characters with the highest number of appearances in Marvel comics.

Python

```
# Top 10 most popular characters by appearances

top_characters = filtered_data[['name',
'APPEARANCES']].sort_values(by='APPEARANCES', ascending=False).head(10)


plt.figure(figsize=(12, 8))

sns.barplot(x='APPEARANCES', y='name', data=top_characters, palette='viridis')

plt.title('Top 10 Most Popular Marvel Characters by Appearances')
```

4.4 Alignment vs. Survival Status:

A stacked bar chart comparing the alignment (Good, Bad, Neutral) of characters to their survival status (Alive or Deceased).

Python

```
# Alignment vs. Survival Status

alignment_alive = pd.crosstab(filtered_data['ALIGN'], filtered_data['ALIVE'])


plt.figure(figsize=(10, 6))

alignment_alive.plot(kind='bar', stacked=True, color=['blue', 'gray'])
```

```
plt.title('Alignment vs. Survival Status')

st.pyplot(plt)
```

4.5 Character Debuts Over Time:

A line chart showing the number of new Marvel character debuts over time.

Python

```
plt.figure(figsize=(12, 6))

character_debuts = filtered_data['Year'].value_counts().sort_index()

plt.plot(character_debuts.index, character_debuts.values, marker='o')

plt.title('Number of Marvel Character Debuts Over Time')

plt.xlabel('Year')

plt.ylabel('Number of Characters')

plt.grid(True)

st.pyplot(plt)
```

5. Tools and Libraries Used:

- Pandas: For data manipulation and preprocessing.
- Seaborn and Matplotlib: For data visualization.
- Streamlit: For creating an interactive dashboard to visualize and filter data dynamically.

6. Conclusion:

This analysis and interactive dashboard provide insights into the Marvel Universe, such as:

- **Character Popularity**: The most popular characters by appearances.
- **Gender Representation**: How gender representation has changed over the decades.
- **Character Survival**: How alignment correlates with survival.

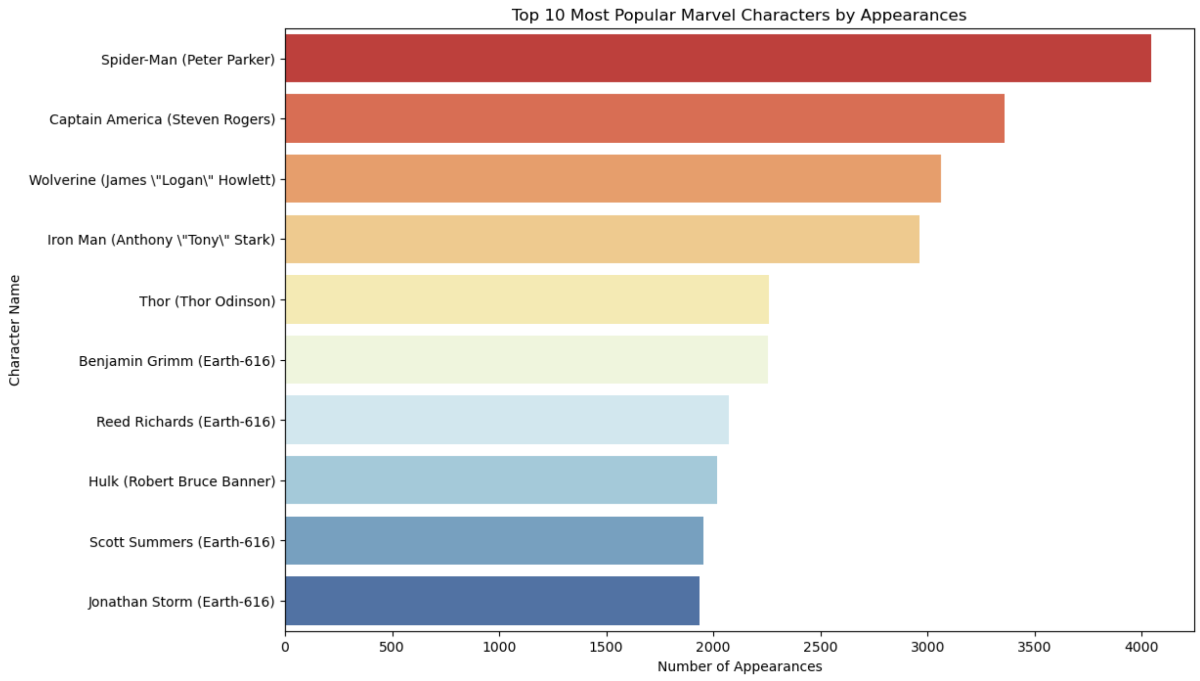
- **Debut Trends**: The introduction of characters across time.

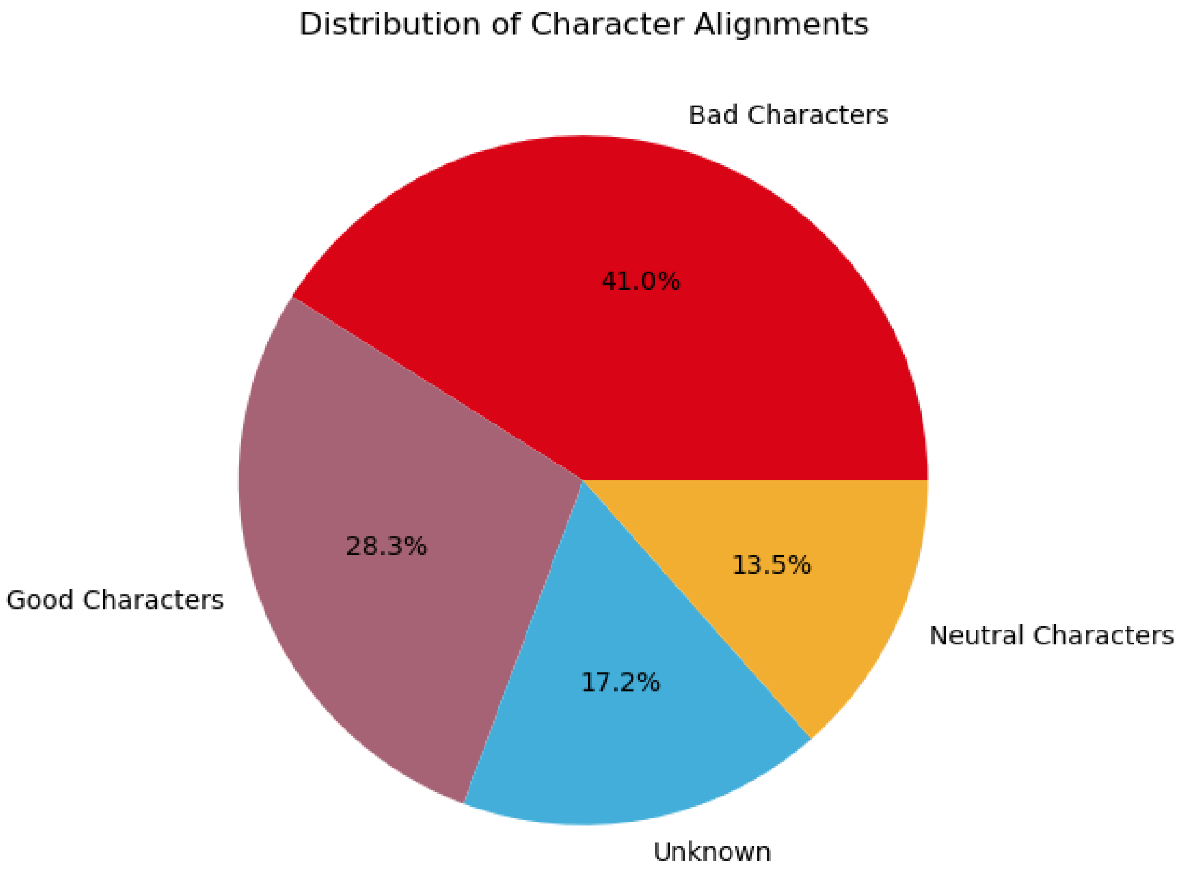
With the Streamlit dashboard, users can dynamically explore various aspects of the data, providing flexibility in analyzing the dataset based on specific filters and criteria.

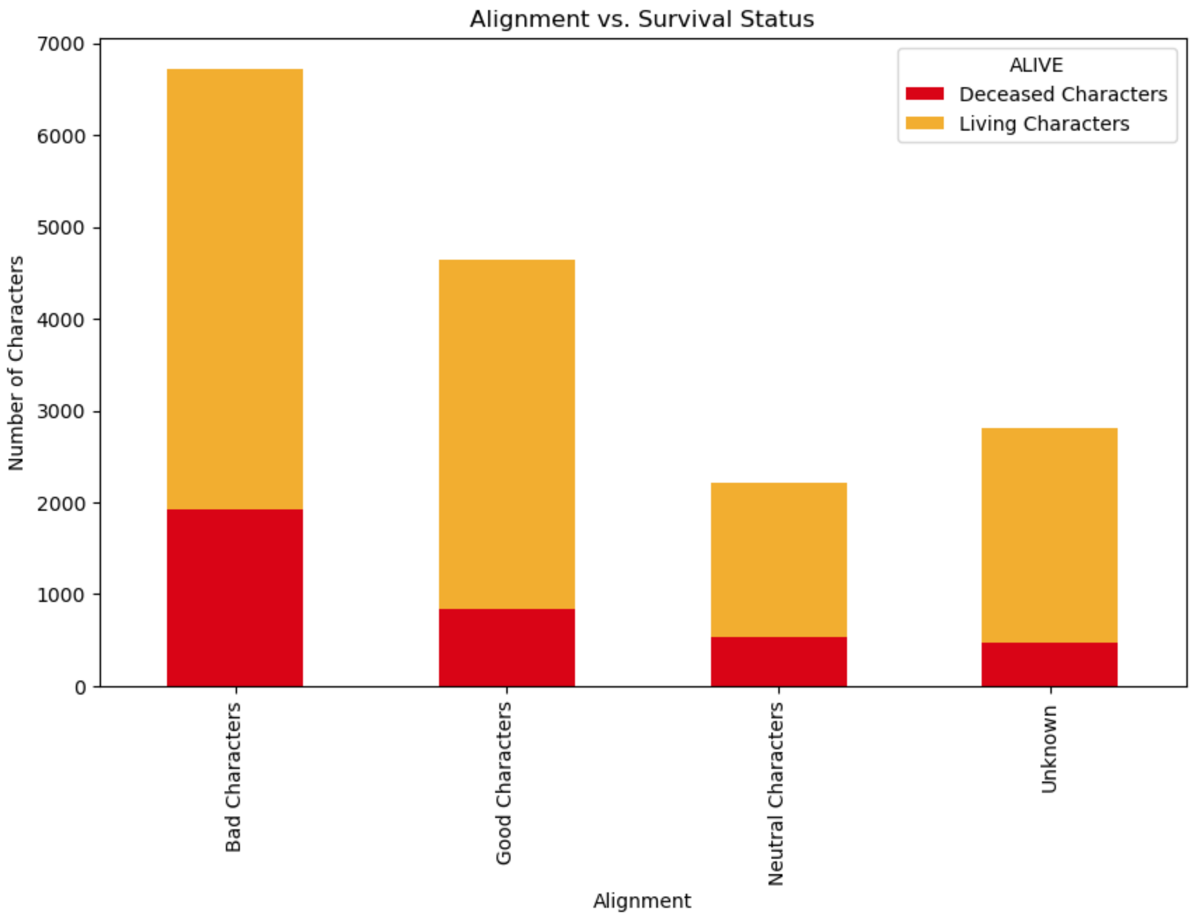
7. Future Work and Improvements:

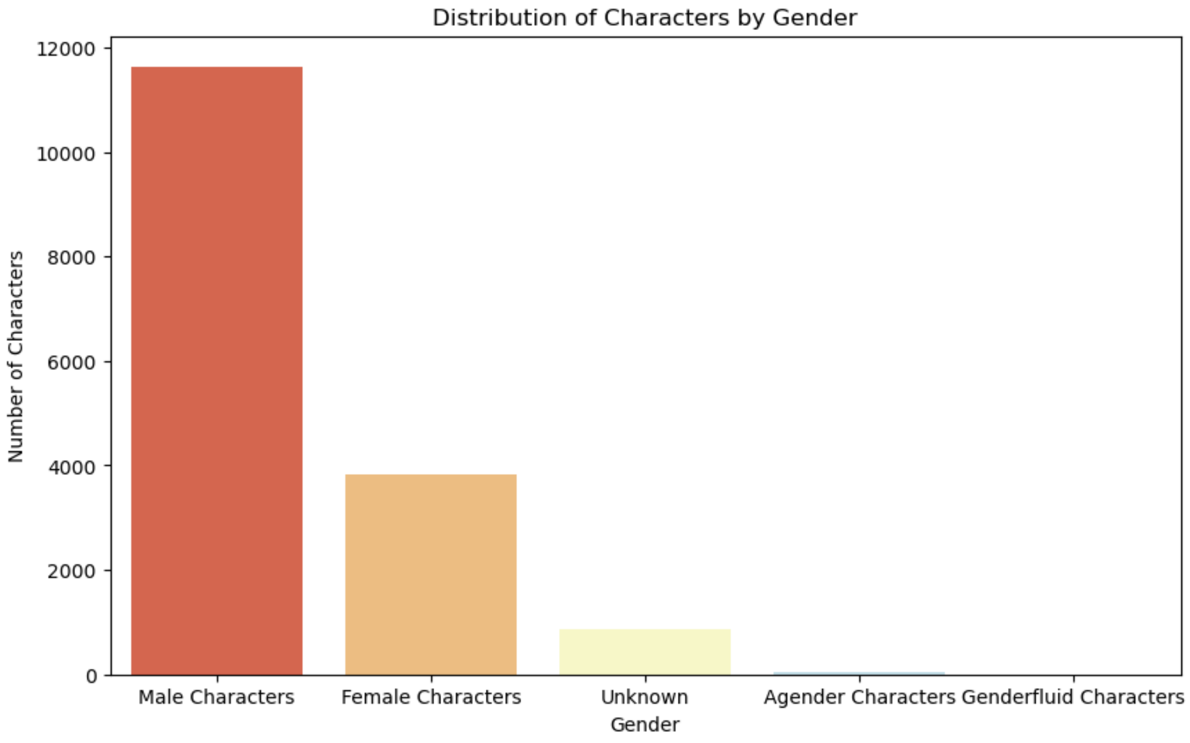
- **Add Predictive Modeling**: Use machine learning models to predict character survival or popularity based on traits.
- **Expand Filters**: Add more filtering options such as filtering by year, eye color, hair color, etc.
- **More Visualizations**: Include advanced visualizations like violin plots, correlation heatmaps, and network graphs to show character relationships.

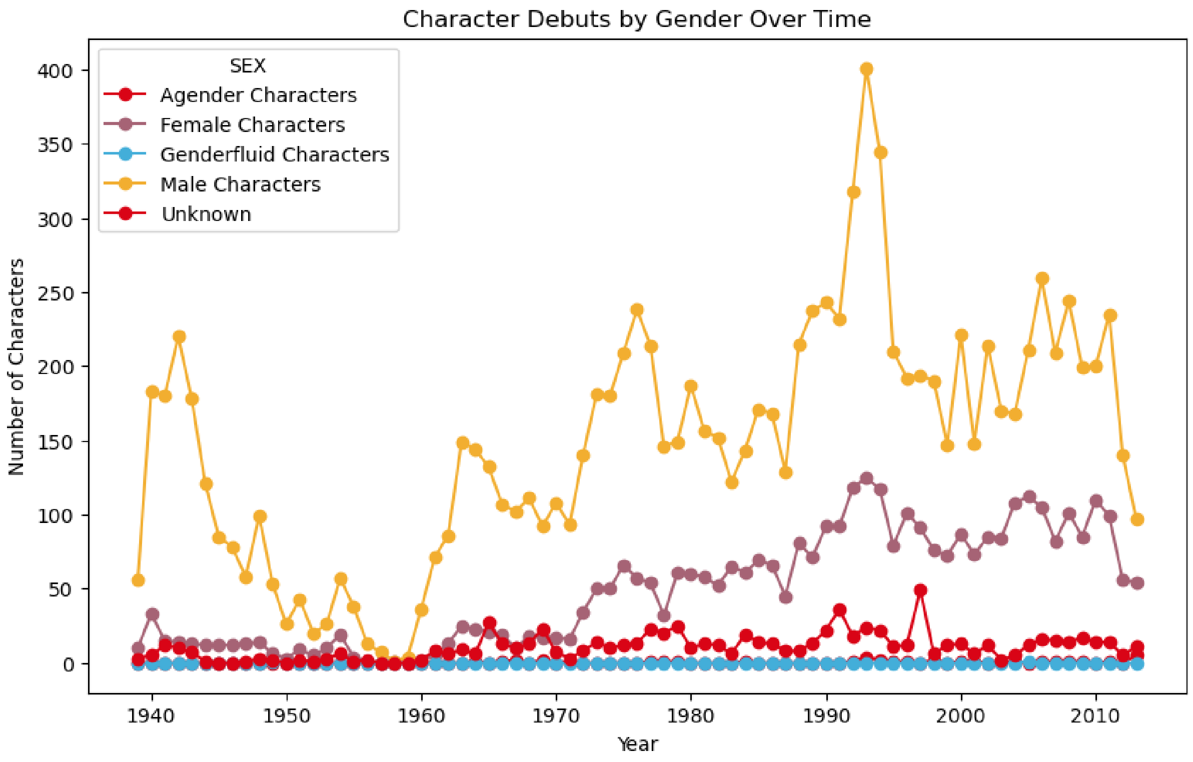
Visual Representation:

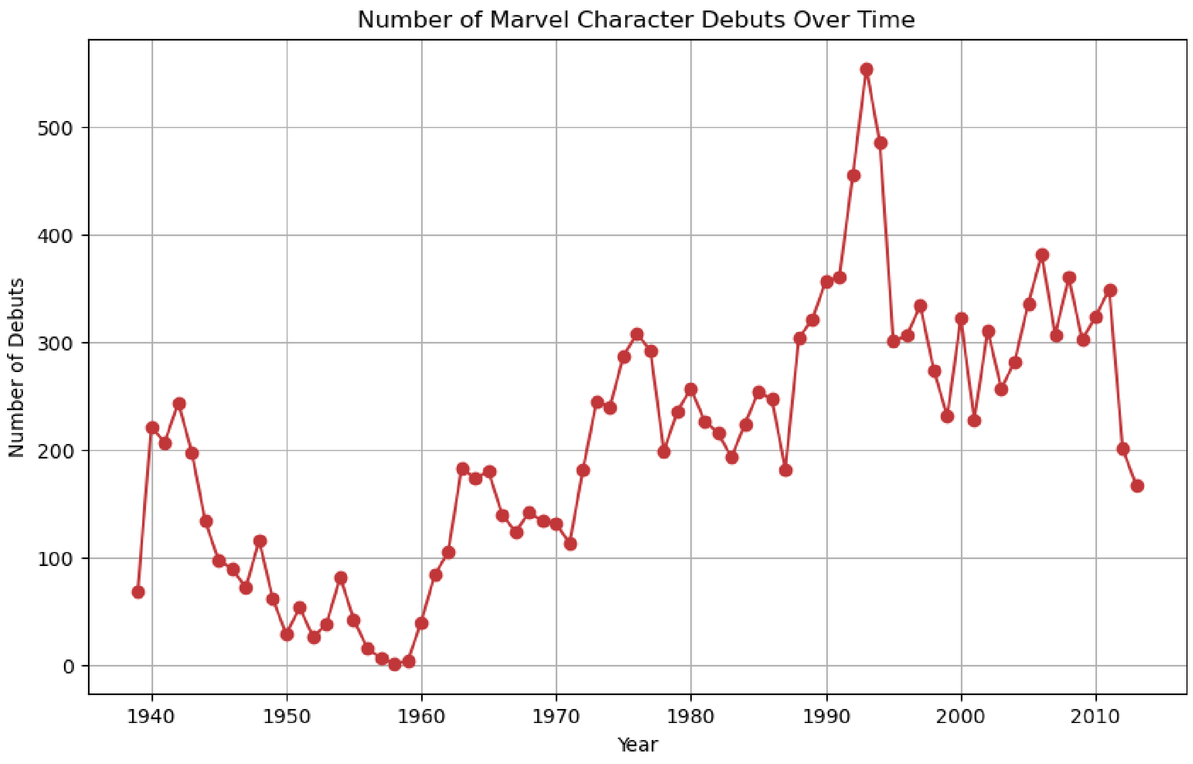


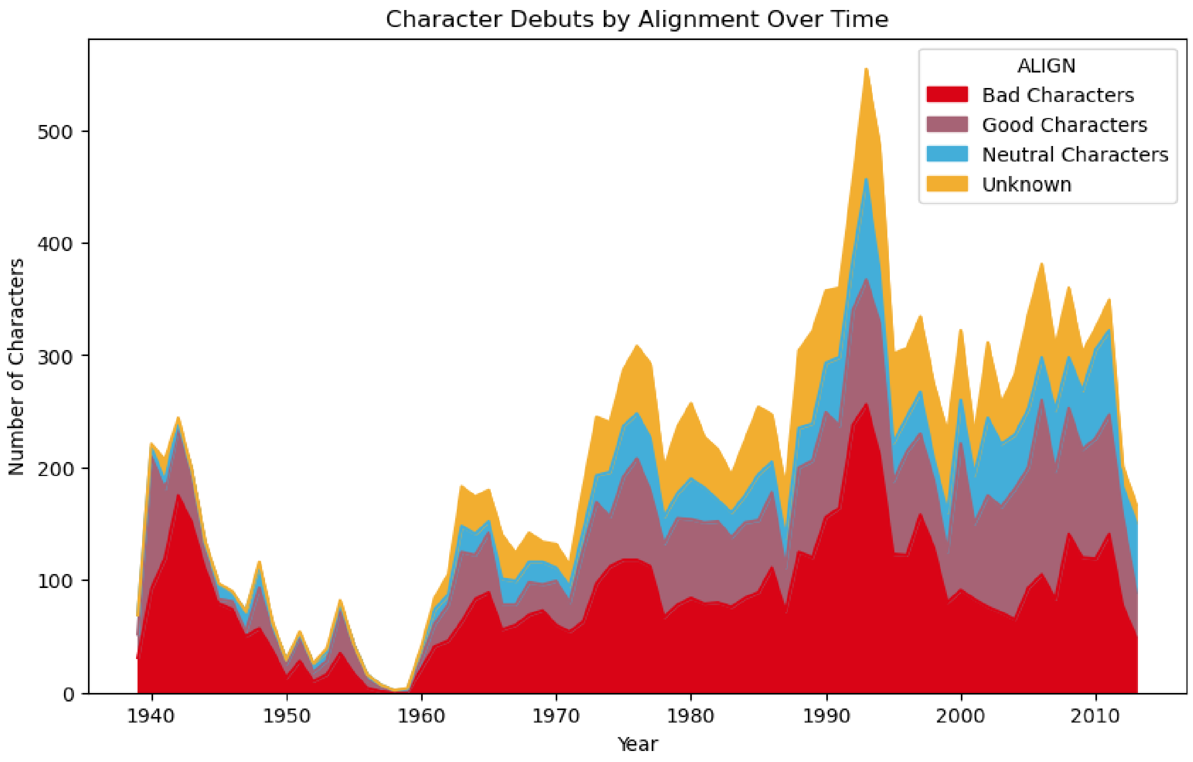


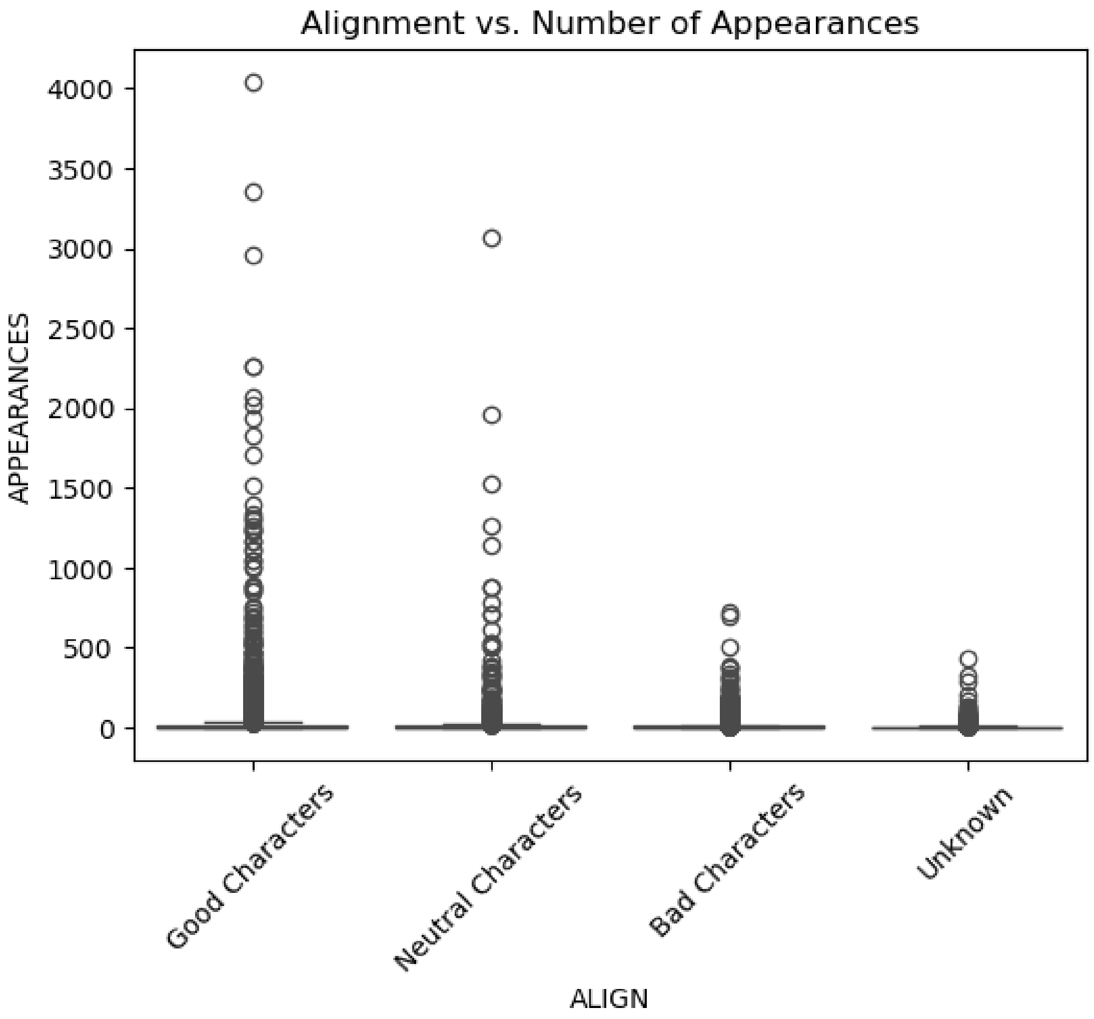

Alignment vs. Survival Status
ALIVE
Deceased Characters
Living Characters
Number of Characters
0
1000
2000
3000
4000
5000
6000
7000
Bad Characters
Good Characters
Neutral Characters
Unknown
Alignment

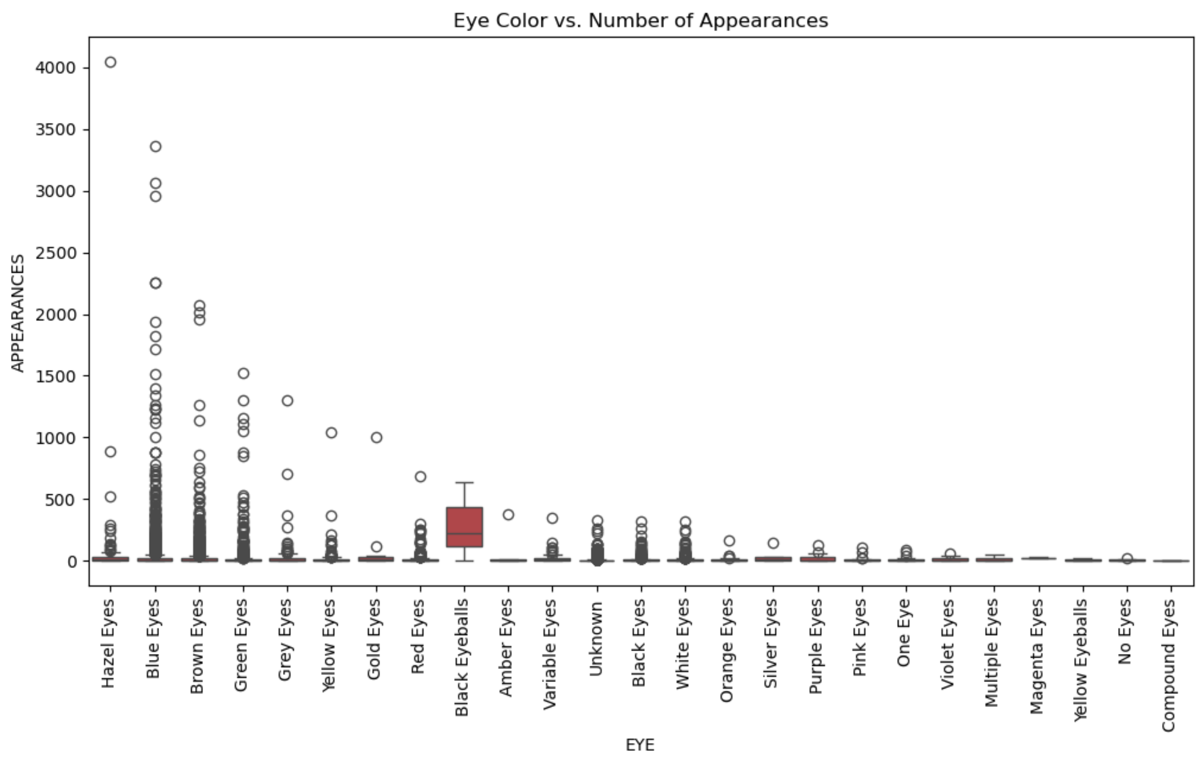

Distribution of Characters by Gender
Number of Characters
0
2000
4000
6000
8000
10000
12000
Male Characters
Female Characters
Unknown
Agender Characters
Genderfluid Characters
Gender

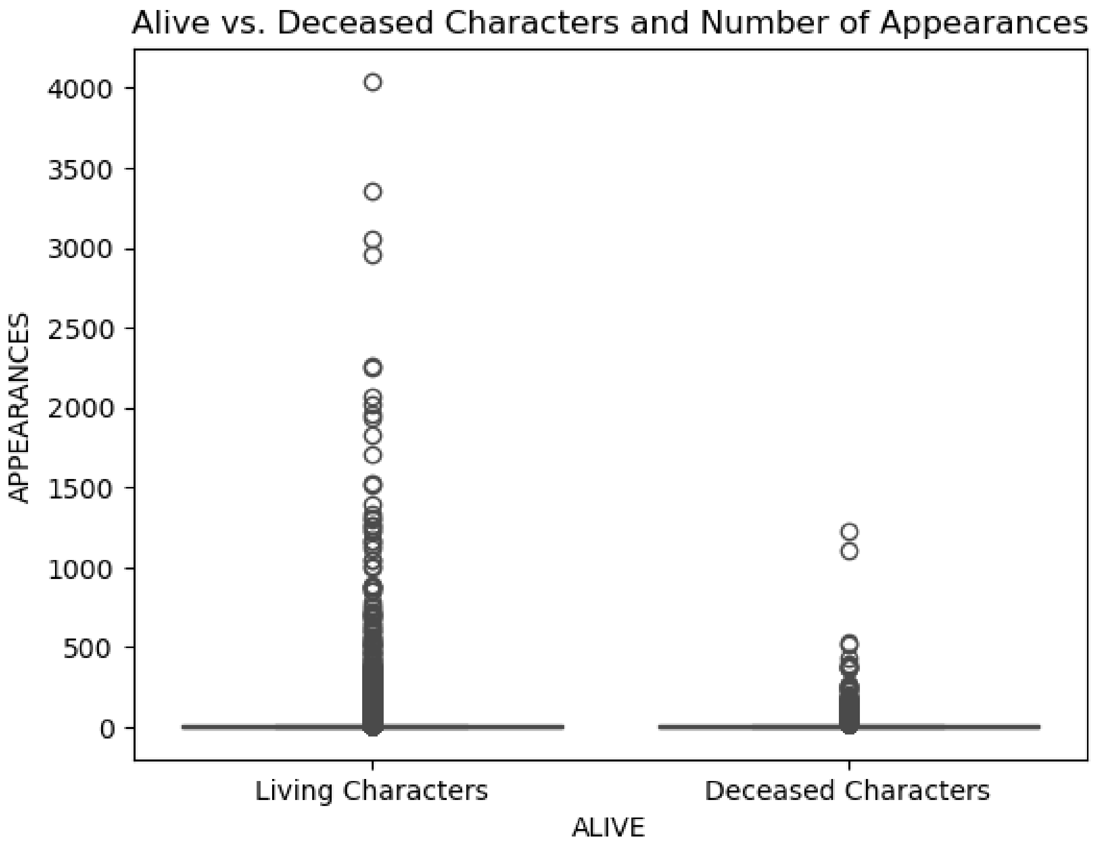

Character Debuts by Gender Over Time
SEX
Agender Characters
Female Characters
Genderfluid Characters
Male Characters
Unknown
Number of Characters
Year
0
50
100
150
200
250
300
350
400
1940
1950
1960
1970
1980
1990
2000
2010

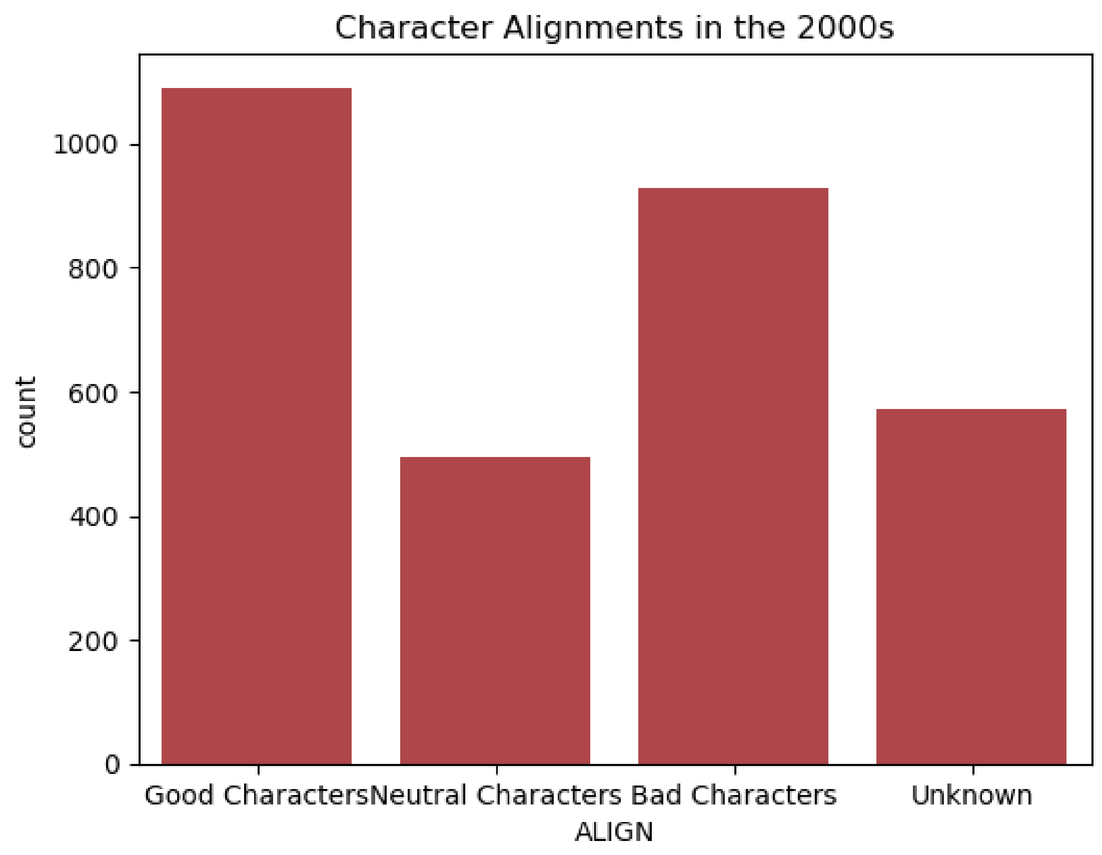

Number of Marvel Character Debuts Over Time
Number of Debuts
Year
0
100
200
300
400
500
1940
1950
1960
1970
1980
1990
2000
2010

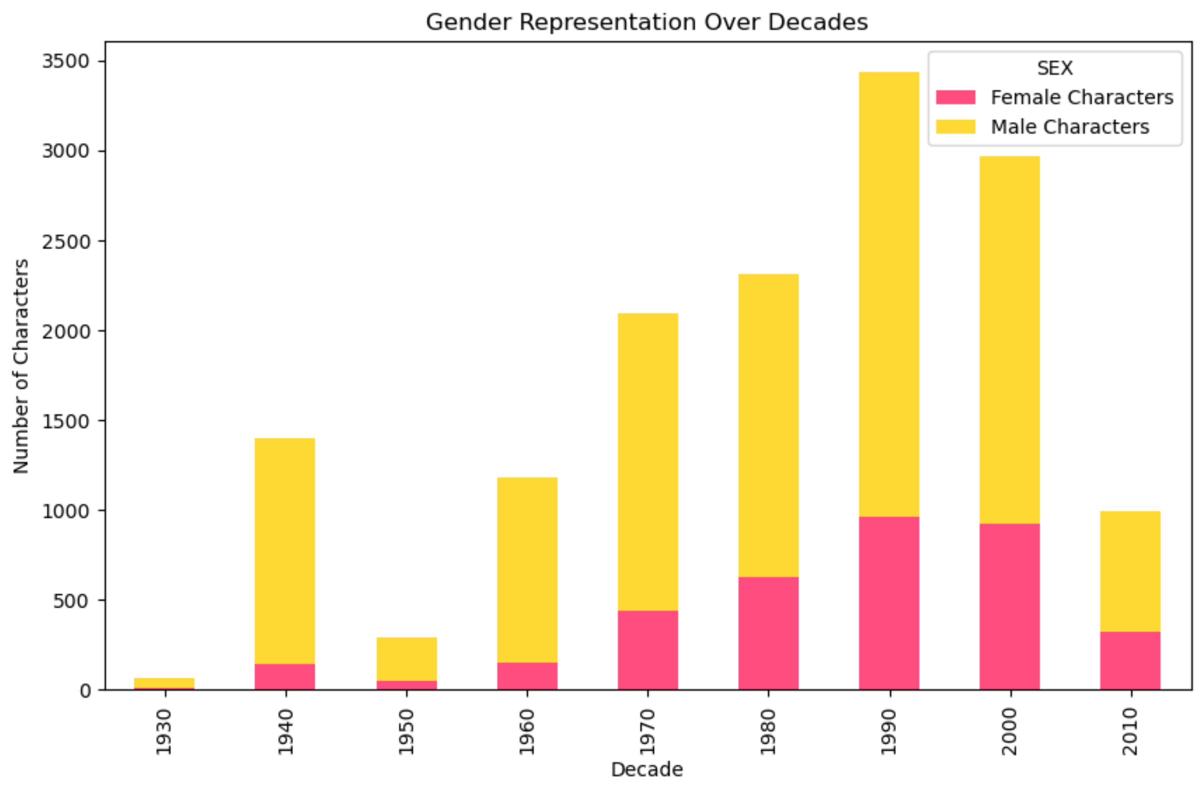

Character Debuts by Alignment Over Time
ALIGN
Bad Characters
Good Characters
Neutral Characters
Unknown
Number of Characters
0
100
200
300
400
500
1940
1950
1960
1970
1980
1990
2000
2010
Year


Alignment vs. Number of Appearances
APPEARANCES
0
500
1000
1500
2000
2500
3000
3500
4000
Good Characters
Neutral Characters
Bad Characters
Unknown
ALIGN


Eye Color vs. Number of Appearances
APPEARANCES
4000
3500
3000
2500
2000
1500
1000
500
0
Hazel Eyes
Blue Eyes
Brown Eyes
Green Eyes
Grey Eyes
Yellow Eyes
Gold Eyes
Red Eyes
Black Eyeballs
Amber Eyes
Variable Eyes
Unknown
Black Eyes
White Eyes
Orange Eyes
Silver Eyes
Purple Eyes
Pink Eyes
One Eye
Violet Eyes
Multiple Eyes
Magenta Eyes
Yellow Eyeballs
No Eyes
Compound Eyes
EYE


Alive vs. Deceased Characters and Number of Appearances
APPEARANCES
4000
3500
3000
2500
2000
1500
1000
500
0
Living Characters
Deceased Characters
ALIVE


Character Alignments in the 2000s
count
1000
800
600
400
200
0
Good Characters
Neutral Characters
Bad Characters
Unknown
ALIGN


Gender Representation Over Decades
Number of Characters
3500
3000
2500
2000
1500
1000
500
0
SEX
Female Characters
Male Characters
1930
1940
1950
1960
1970
1980
1990
2000
2010
Decade