



MARVEL SUPERHEROES

Presentation by 梁廣廷



INTRODUCTION



OUTLINE

- 🔍 SHOW THE DATASET NAME AND SOURCE
- 🔍 COLUMN NAMES DESCRIPTIONS AND RECORDS
- 🔍 YOUR QUESTIONS AND CORRESPONDING ANALYSIS PLOTS
- 🔍 CONCLUSIONS

DATASET NAME AND SOURCE ADDRESS



Marvel Superheroes

<https://www.kaggle.com/datasets/danielr/marvel-superheroes>

LIBRARY

streamlit

pandas

matplotlib

seaborn

plotly

pre-commit

ruff

STREAMLIT

A faster way to build and share data apps

Streamlit turns data scripts into shareable web apps in minutes.

All in pure Python. No front-end experience required.

[Try Streamlit now](#)

[Deploy on Community Cloud \(it's free!\)](#)



Presentation by 梁廣廷 | 資料視覺化 | 2023 FALL | Tamkang University

QUESTIONS

🔍 FIRST APPEARANCE

🔍 IDENTITY DISTRIBUTION

🔍 GENDER VS. ALIGNMENT

🔍 IDENTITY VS. ALIGNMENT

🔍 ABILITIES & POWERS

🔍 TOTAL POWER OF THE TEAM
BY CHARACTERS

🔍 COMPARISON OF CHARACTERS

TABLEAU PUBLIC LINK

[link](#)

DEMO

LINK

X Share ☆ ○ :

app

- first appearance
- demographic
- abilities powers
- total power

Select a data visualization to view.

Marvel Universe: Civil War

Marvel Universe Analysis

Link', 'Demograph: [Link](#)', and 'Abilities & Powers: [Link](#)'. In the top right corner of the main area, there are icons for 'Share', a star, a circle, and more options."/>

First Appearances: [Link](#)

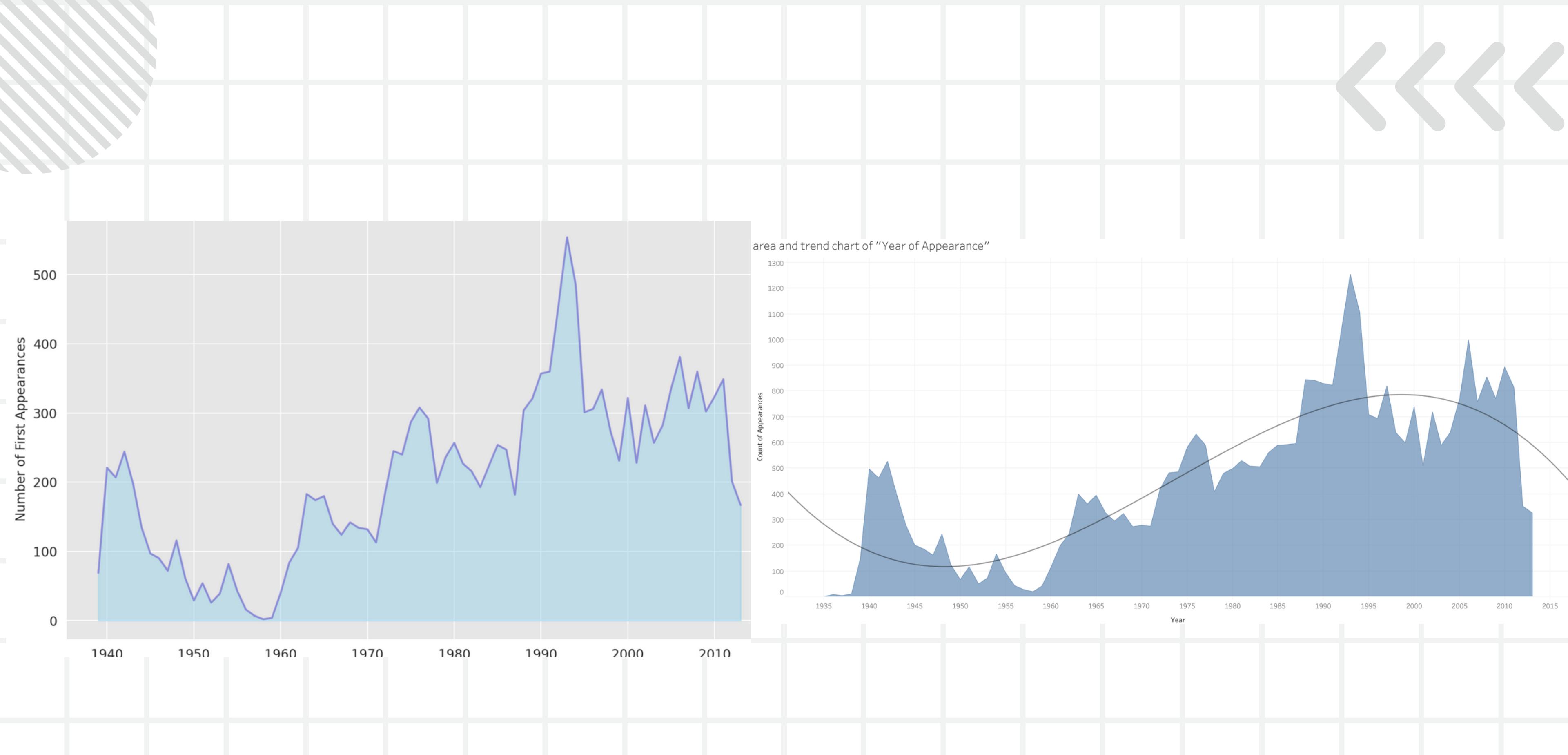
Demograph: [Link](#)

Abilities & Powers: [Link](#)

MARVEL UNIVERSE ANALYSIS



Presentation by 梁廣廷 | 資料視覺化 | 2023 FALL | Tamkang University





```
# Import libraries
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt

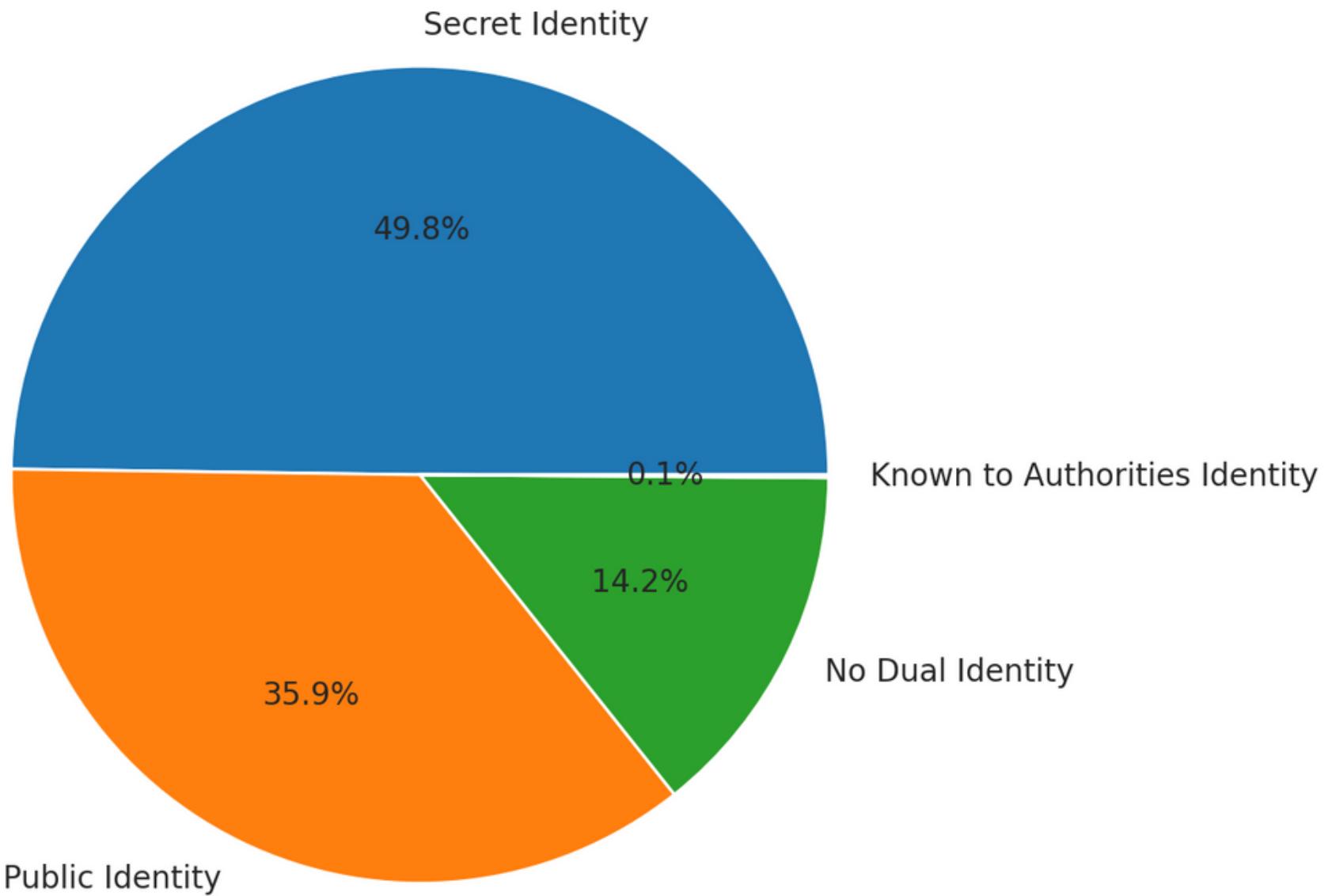
# Read the data
wiki = pd.read_csv("data/marvel-wikia-data.csv")

# Group by 'Year' and count the number of appearances
year_counts = wiki['Year'].value_counts().sort_index()

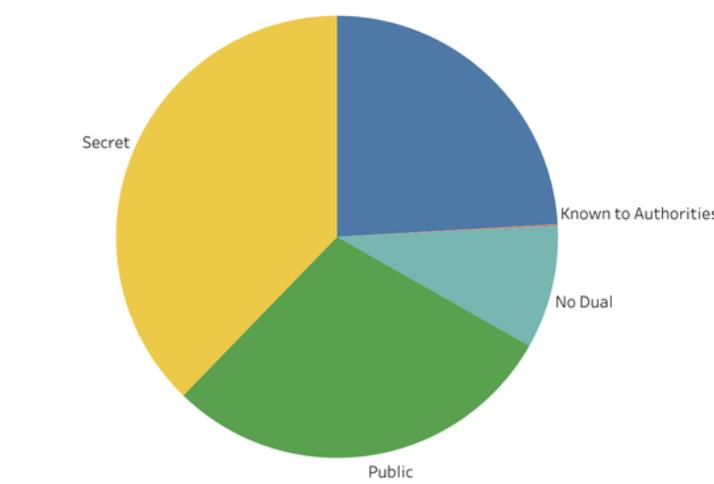
# Create an area chart
plt.figure(figsize=(10,6))
plt.fill_between(year_counts.index, year_counts, color="skyblue",
alpha=0.4)
plt.title('Number of First Appearances by Year')
plt.xlabel('Year')
plt.ylabel('Number of First Appearances')

# Display the plot in Streamlit
st.pyplot(plt)
```

Identity Distribution



pie chart of Identity



Identity
Null
Identity Unknown
Known to Authorities
No Dual
Public
Secret

```
● ● ●

# Import libraries
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Read the data
wiki = pd.read_csv('data/marvel-wikia-data.csv')

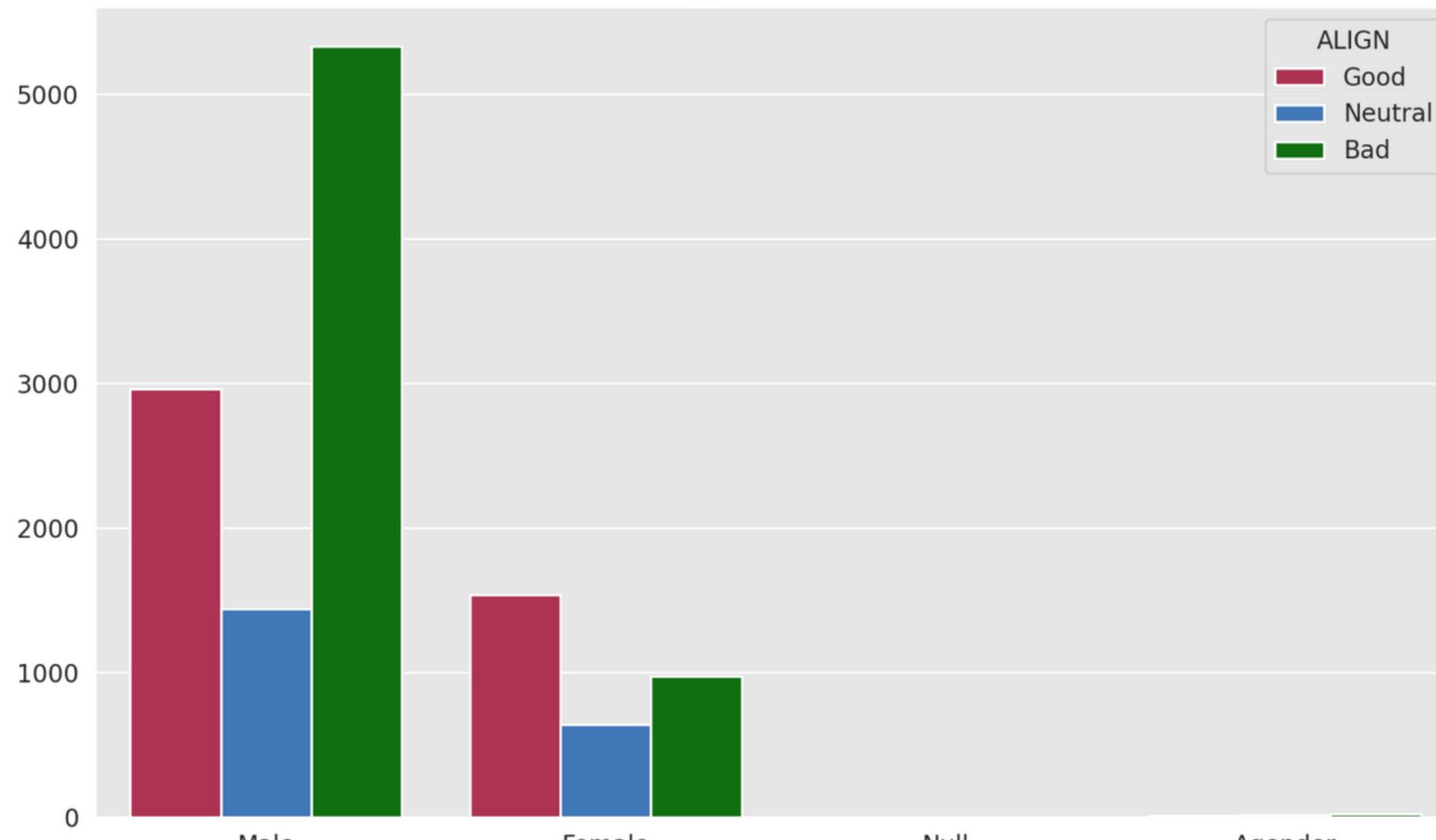
# Count the values
identity_counts = wiki['ID'].value_counts()

# Create a pie chart
plt.figure(figsize=(10, 6))
plt.pie(identity_counts, labels=identity_counts.index,
        autopct='%1.1f%%')
plt.title('Identity Distribution')
plt.show()

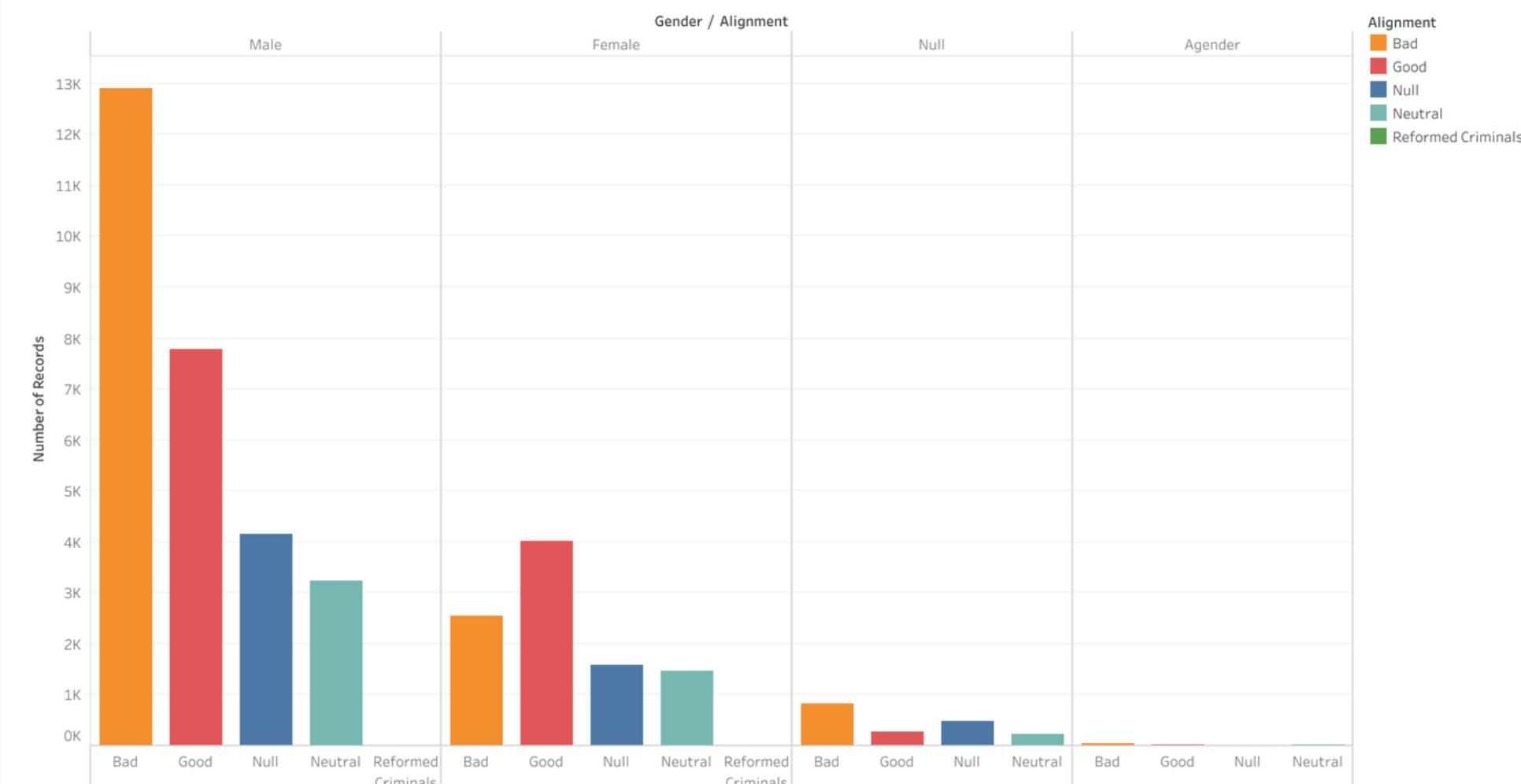
# Display the plot in Streamlit
st.markdown('### Identity Distribution')
st.pyplot(plt)

# Display the plot in Streamlit
st.pyplot(plt)
```

Alignment / Sex



bar chart of "Gender vs. Alignment"



```
# Import libraries
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Read the data
wiki = pd.read_csv("data/marvel-wikia-data.csv")

# Filter the data
p1 = wiki[wiki["ALIGN"] != ""]
p1 = p1.copy()

# Replace 'Identity' in 'ID' column
p1["ID"] = p1["ID"].str.replace("Identity", "")

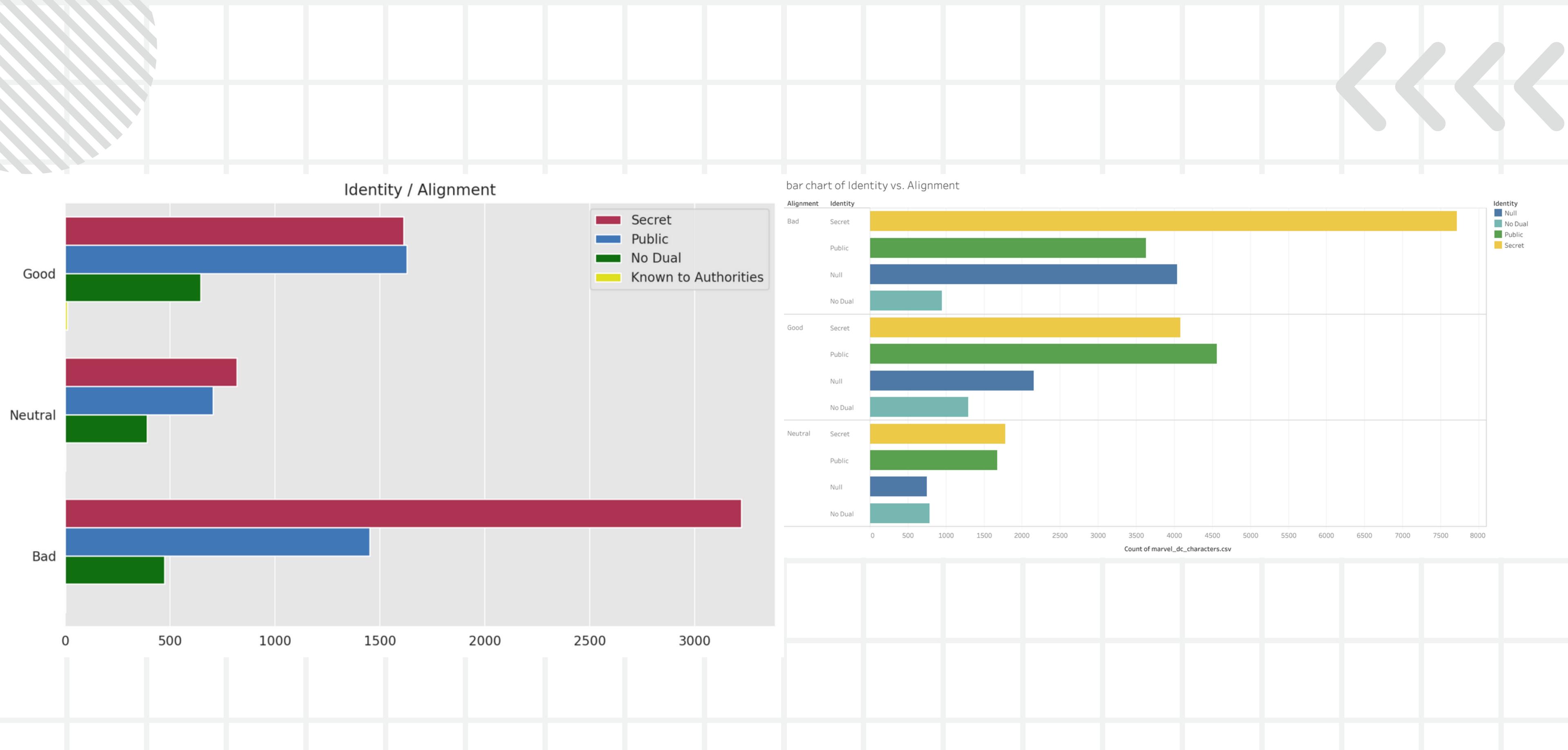
# Replace empty strings in 'ID' and 'SEX' columns with 'Null'
p1["ID"] = p1["ID"].replace("", "Null")
p1["SEX"] = p1["SEX"].replace("", "Null")

# Remove 'Characters' from 'SEX' and 'ALIGN' columns
p1["SEX"] = p1["SEX"].str.replace("Characters", "")
p1["ALIGN"] = p1["ALIGN"].str.replace("Characters", "")

# Replace 'Genderfluid' with 'Agender' in 'SEX' column
p1["SEX"] = p1["SEX"].replace("Genderfluid", "Agender")

# Convert 'SEX' column to categorical and specify the order of categories
p1["SEX"] = pd.Categorical(p1["SEX"], categories=["Male", "Female", "Null", "Agender"],
                           ordered=True)
# Create a bar plot
plt.figure(figsize=(10, 6))
sns.countplot(x="SEX", hue="ALIGN", data=p1, palette=["#C21E48", "#2C75CB", "green"])

# Show the plot
st.pyplot(plt)
```





```
# Import libraries
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Prepare the data
# pl_filtered = pl[pl['ID'] != 'Known to Authorities']
pl_filtered = pl.copy()

# Create a bar plot
plt.figure(figsize=(10, 6))
sns.countplot(y="ALIGN", hue="ID", data=pl_filtered, palette=["#C21E48", "#2C75CB", "green",
"yellow"])
# Set the background color to black
sns.set_style("darkgrid", {"axes.facecolor": ".9"})

# Move the legend to the top
plt.legend(loc="upper right")

# Set the labels and title
plt.xlabel(None)
plt.ylabel(None)
plt.title("Identity / Alignment")

# Show the plot
st.markdown("## Identity / Alignment Plot")
st.pyplot(plt)
```

Abilities & Powers



```
# Import libraries
import streamlit as st
import pandas as pd
import plotly.express as px

# Read the data
stats = pd.read_csv("data/charcters_stats.csv")

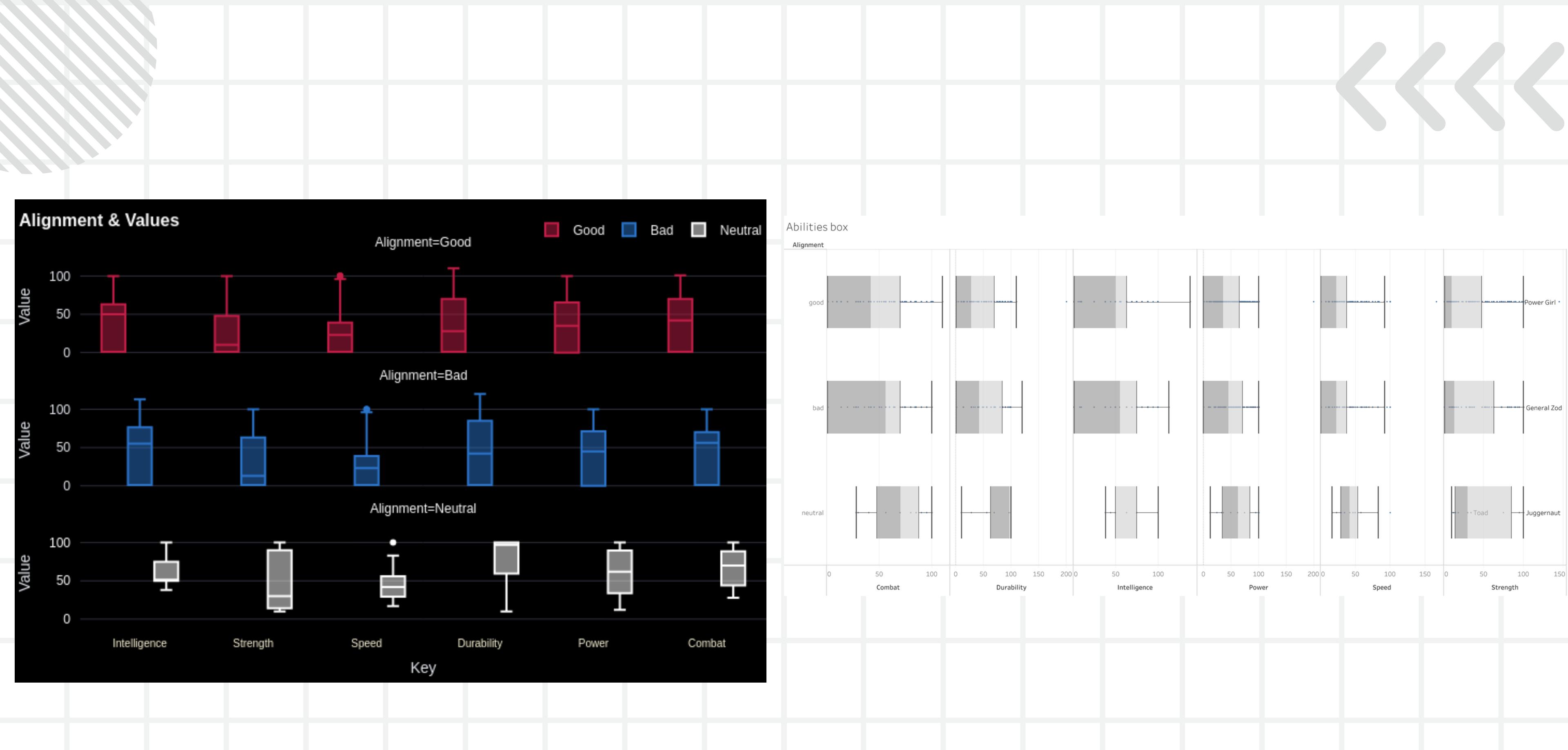
# Reshape the data
melted_stats = pd.melt(
    stats,
    id_vars=["Alignment", "Name"],
    var_name="Key",
    value_name="Value",
)
melted_stats = melted_stats[melted_stats["Key"] != "Total"]

# Modify the Alignment column
melted_stats["Alignment"] = melted_stats["Alignment"].str.title()

# Group and arrange the data
grouped_stats = melted_stats.groupby("Key").apply(lambda x: x.sort_values("Value",
ascending=False))
# Plotting
fig = px.strip(
    grouped_stats,
    x="Key",
    y="Value",
    color="Key",
    facet_col="Alignment",
    facet_col_wrap=1,
    title="Abilities & Powers",
)

# Update the layout
fig.update_layout(
    legend_title="Abilities & Powers",
    xaxis={"tickfont": {"color": "white"}},
    yaxis={"tickfont": {}},
    title={"font": {"color": "seagreen"}},
    font={"color": "white"},
)
fig.update_xaxes(title_text="Abilities")
fig.update_yaxes(title_text="Power")

# Display the plot
# fig.show()
st.plotly_chart(fig)
```



```
# Import libraries
import streamlit as st
import pandas as pd
import plotly.express as px # type: ignore

# Read the data
stats = pd.read_csv('data/charcters_stats.csv')

# Select and filter the data
filtered_stats = stats.drop(columns=['Name', 'Total']).query('Alignment != ""')

# Reshape the data
melted_stats = pd.melt(filtered_stats, id_vars=['Alignment'], var_name='Key',
value_name='Value')
# Drop the rows where Alignment is NaN
melted_stats = melted_stats.dropna(subset=['Alignment'])

# Modify the Alignment column
melted_stats['Alignment'] = melted_stats['Alignment'].str.title()

# Plotting
fig = px.box(
    melted_stats,
    x='Key',
    y='Value',
    color='Alignment',
    facet_col='Alignment',
    facet_col_wrap=1,
    color_discrete_sequence=['#C21E48', '#2C75CB', 'white'],
    title='Alignment & Values',
)

# Update the layout
fig.update_layout(
    plot_bgcolor='black',
    paper_bgcolor='black',
    legend_bgcolor='black',
    legend_title=None,
    xaxis=dict(tickfont=dict(color='cornsilk', size=10)),
    yaxis=dict(tickfont=dict(color='white')),
    legend=dict(orientation='h', yanchor='bottom', y=1.02, xanchor='right', x=1),
    font=dict(color='white'),
    margin=dict(t=50, b=50),
    boxmode='group',
    boxgap=0.3,
    boxgroupgap=0.1,
    showlegend=True,
)

# Set the y-axis limits
fig.update_yaxes(range=[-10, 130])

# Display the plot
st.plotly_chart(fig)
```

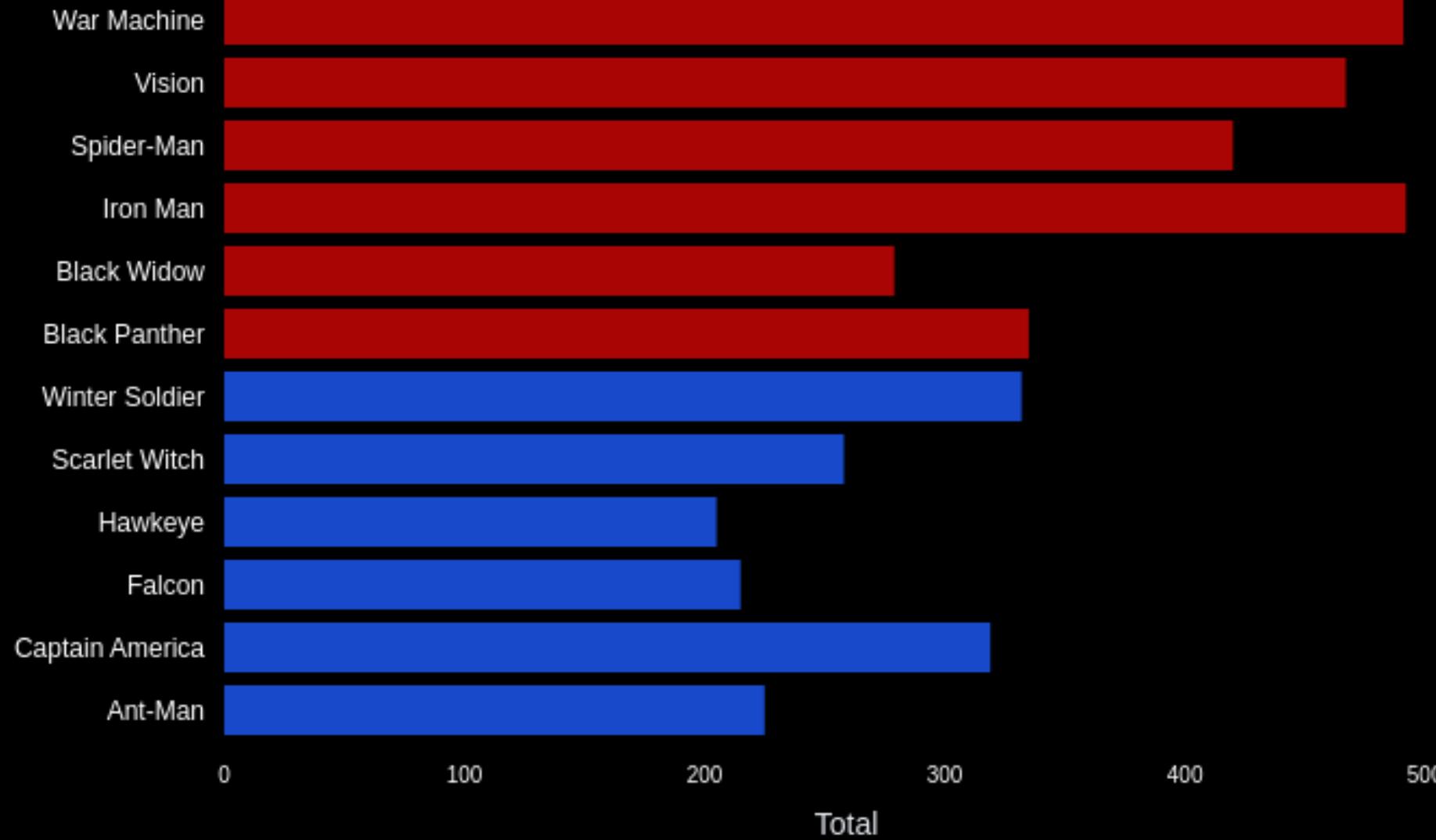
CIVIL WAR ANALYSIS



Team Cap vs Team Iron Man

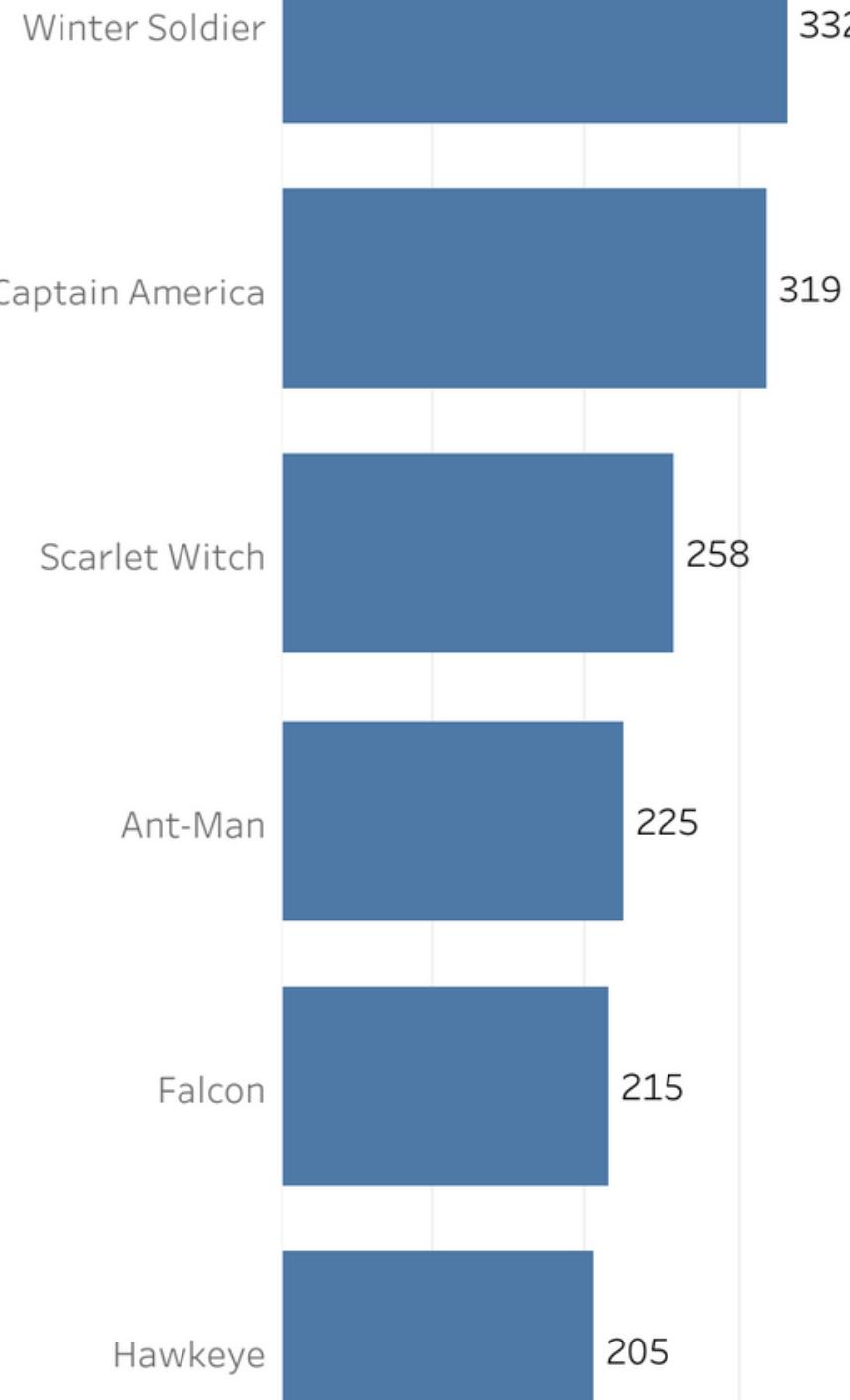
Total Power of the Team by Characters

Name



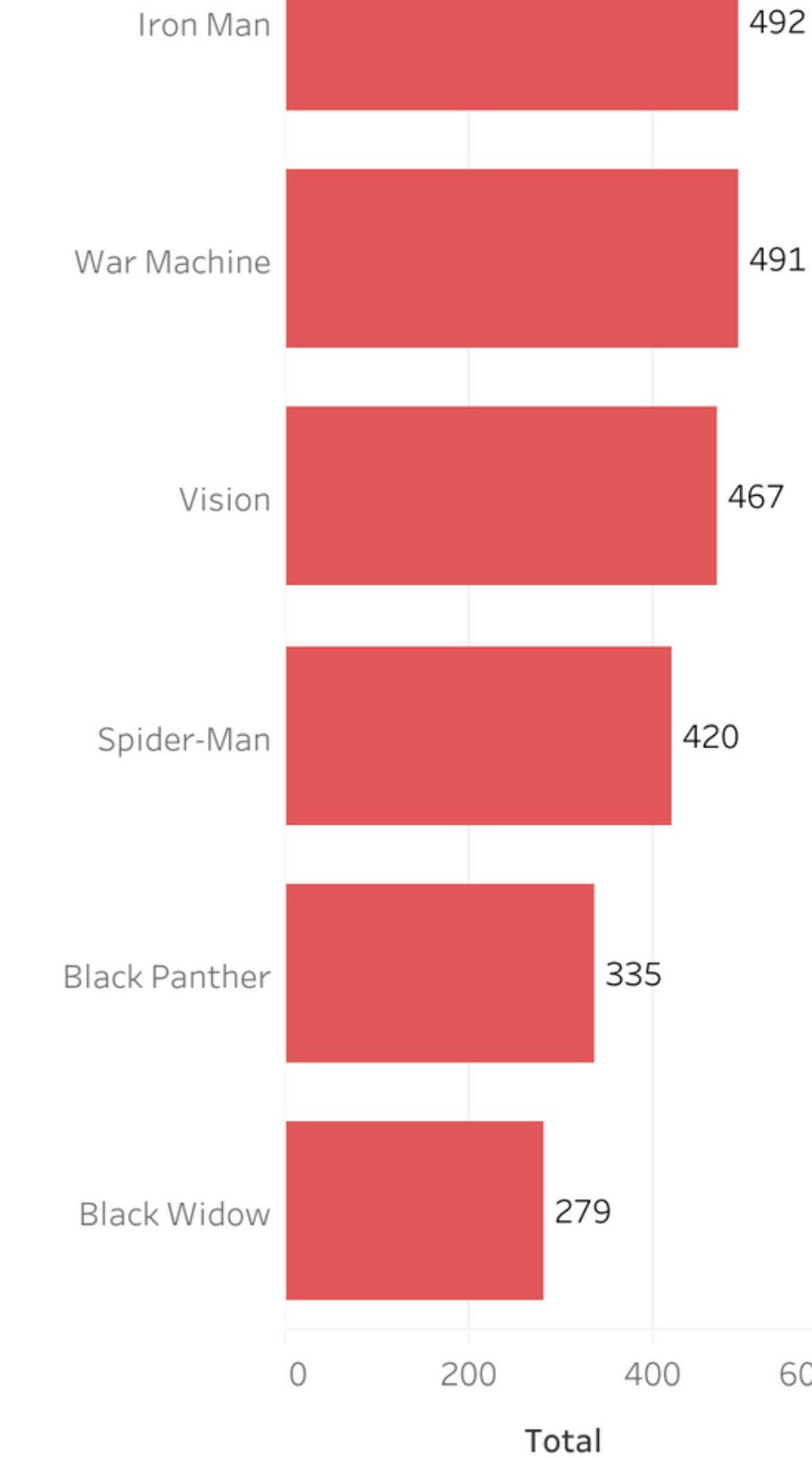
Team Cap

Name



Team Iron Man

Name



```
● ● ●

import pandas as pd
import plotly.express as px
import streamlit as st

# Read the data
stats = pd.read_csv('data/charcters_stats.csv')

# Mutate the Team column
stats['Team'] = stats['Name'].map({
    "Black Panther": "Team Iron Man",
    "Iron Man": "Team Iron Man",
    "Black Widow": "Team Iron Man",
    "Vision": "Team Iron Man",
    "Spider-Man": "Team Iron Man",
    "War Machine": "Team Iron Man",
    "Winter Soldier": "Team Cap",
    "Captain America": "Team Cap",
    "Scarlet Witch": "Team Cap",
    "Hawkeye": "Team Cap",
    "Ant-Man": "Team Cap",
    "Falcon": "Team Cap"
})

# Filter out rows with missing Team values
filtered_stats = stats.dropna(subset=['Team'])

# Reshape the data
melted_stats = pd.melt(filtered_stats, id_vars=['Name', 'Team'], var_name='Key',
value_name='Value')
# Filter for Key == "Total"
filtered_stats = melted_stats[melted_stats['Key'] == 'Total']

# Plotting
fig = px.bar(
    filtered_stats,
    x='Value',
    y='Name',
    color='Team',
    orientation='h',
    color_discrete_sequence=["#1849CA", "#AA0505"],
    labels={'Value': 'Total', 'Name': 'Name'},
    title='Total Power of the Team by Characters'
)

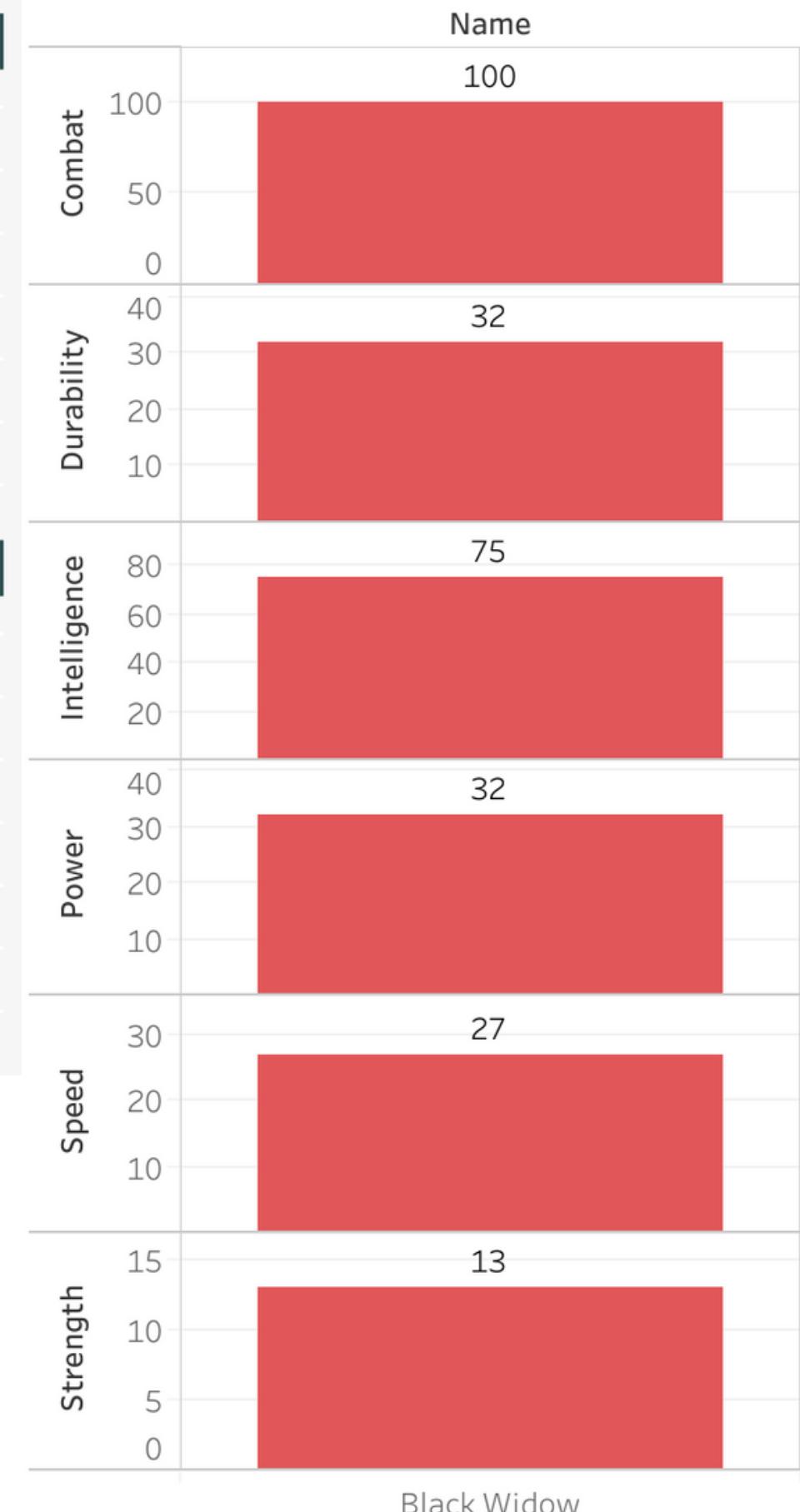
# Update the layout
fig.update_layout(
    plot_bgcolor='black',
    paper_bgcolor='black',
    legend_bgcolor='black',
    legend_title=None,
    xaxis=dict(tickfont=dict(color='white', size=10)),
    yaxis=dict(tickfont=dict(color='white')),
    font=dict(color='white'),
    margin=dict(t=50, b=50),
    showlegend=False
)

# Display the plot
st.plotly_chart(fig)
```

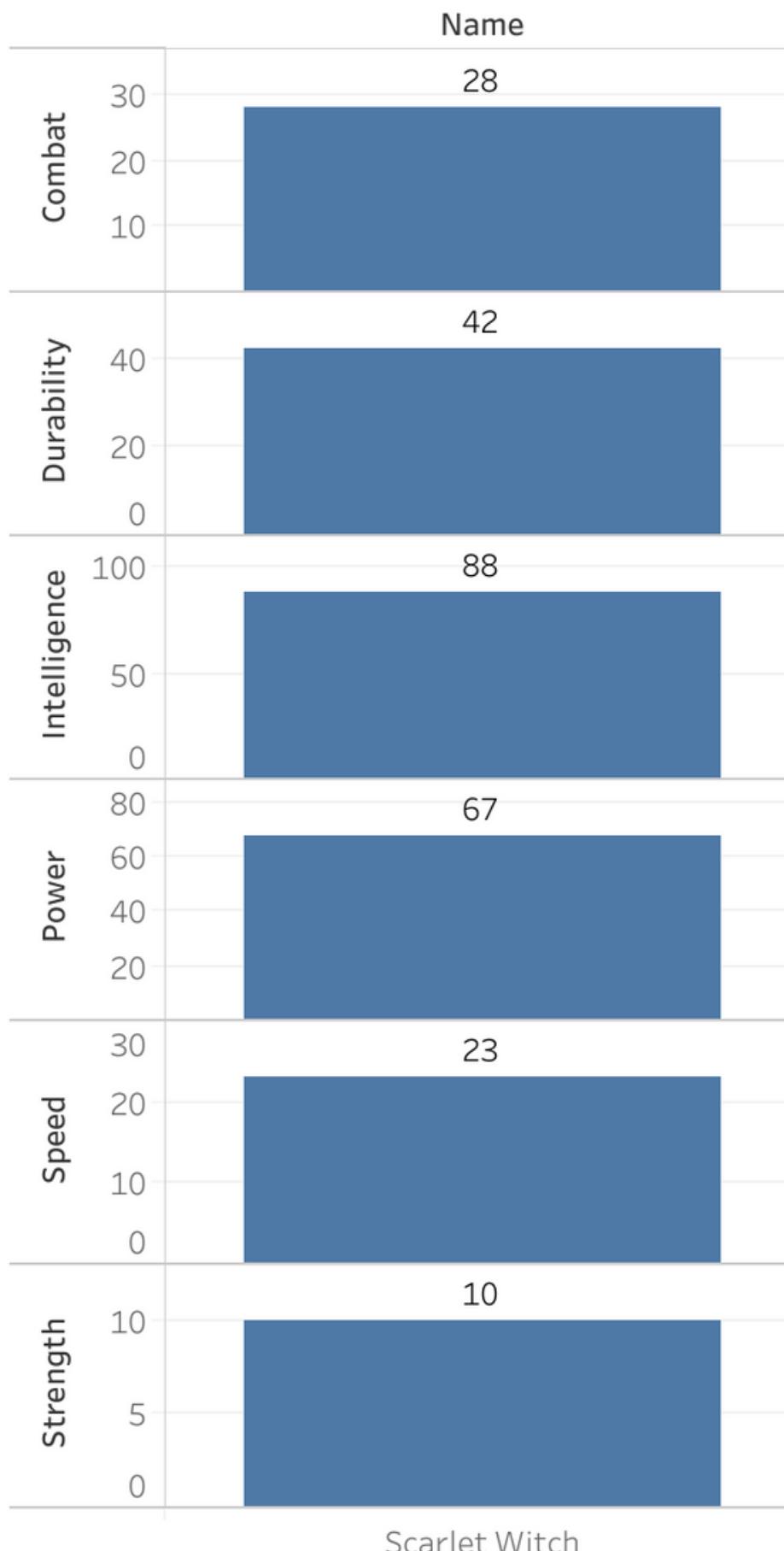
Team Iron Man vs Team Cap

Black Widow vs Scarlet Witch		Iron Man vs Captain America		Spider-Man vs Ant-Man	
Total -	258	279	492	319	420
Strength -	10	13	85	19	55
Speed -	23	27	58	35	60
Power -	67	32	100	46	58
Intelligence -	88	75	100	63	88
Durability -	42	32	85	56	74
Combat -	28	100	64	100	85
Vision vs Hawkeye		War Machine vs Falcon		Winter Soldier vs Black Panther	
Total -	467	205	491	215	332
Strength -	72	12	80	13	32
Speed -	54	23	63	50	35
Power -	76	26	100	22	60
Intelligence -	100	50	63	38	56
Durability -	95	14	100	28	65
Combat -	70	80	85	64	84

Black Widow



Scarlet Witch



```
● ● ●

import pandas as pd
import plotly.express as px
import streamlit as st
st.title('comparison')

# Read the data
stats = pd.read_csv('data/charcters_stats.csv')

# Mutate the Team column
stats['Team'] = stats['Name'].map(
    {
        'Black Panther': 'Team Iron Man',
        'Iron Man': 'Team Iron Man',
        'Black Widow': 'Team Iron Man',
        'Vision': 'Team Iron Man',
        'Spider-Man': 'Team Iron Man',
        'War Machine': 'Team Iron Man',
        'Winter Soldier': 'Team Cap',
        'Captain America': 'Team Cap',
        'Scarlet Witch': 'Team Cap',
        'Hawkeye': 'Team Cap',
        'Ant-Man': 'Team Cap',
        'Falcon': 'Team Cap',
    }
)

# Filter out rows with missing Team values
filtered_stats = stats.dropna(subset=['Team'])

# Create a new DataFrame for the VS matchups
vs_matchups = pd.DataFrame(
    [
        {
            'Name': [
                'Black Panther',
                'Iron Man',
                'Black Widow',
                'Vision',
                'Spider-Man',
                'War Machine',
                'Winter Soldier',
                'Captain America',
                'Scarlet Witch',
                'Hawkeye',
                'Ant-Man',
                'Falcon',
            ],
            'VS': [
                'Winter Soldier vs Black Panther',
                'Iron Man vs Captain America',
                'Black Widow vs Scarlet Witch',
                'Vision vs Hawkeye',
                'Spider-Man vs Ant-Man',
                'War Machine vs Falcon',
                'Winter Soldier vs Black Panther',
                'Iron Man vs Captain America',
                'Black Widow vs Scarlet Witch',
                'Vision vs Hawkeye',
                'Spider-Man vs Ant-Man',
                'War Machine vs Falcon',
            ],
        },
    ]
)

# Merge the VS matchups DataFrame with the filtered_stats DataFrame
merged_stats = pd.merge(filtered_stats, vs_matchups, on='Name')

# Reshape the data
melted_stats = pd.melt(
    merged_stats, id_vars=['Name', 'Team', 'VS'], var_name='Key', value_name='Value'
)

# Filter for Black Panther and Winter Soldier
filtered_melted_stats = melted_stats[melted_stats['Name'].isin(['Black Panther', 'Winter Soldier'])]
# Plotting
fig = px.bar(
    melted_stats,
    x="VS",
    y="Value",
    color="Name",
    facet_row="Key",
    labels={"Value": "Value", "VS": "VS"},
    color_discrete_sequence=[
        "#9D152C",
        "#080808",
        "#7A1F1E",
        "#1849CA",
        "#a71930",
        "#6f195f",
        "#AA0505",
        "#FF3030",
        "#DF1F20",
        "#08991f",
        "#666666",
        "#6a9a9a",
    ],
)
# Update the layout
fig.update_layout(
    plot_bgcolor="#F6F6F7",
    paper_bgcolor="#F6F6F7",
    title={
        'text': 'Team Iron Man vs Team Cap',
        'x': 0.5,
        'y': 0.95,
        'xanchor': 'center',
        'yanchor': 'top',
    },
    font=dict(color="#0E305D", size=20),
    showlegend=False,
)
# Display the plot
st.plotly_chart(fig)
```

REFERENCES

- 🔍 **STREAMLIT FOR DATA SCIENCE:
CREATE INTERACTIVE DATA APPS IN PYTHON**
- 🔍 **DOCS.STREAMLIT.IO/**
- 🔍 **HARNESSING GITHUB COPILOT WITH PYTHON**

THANK YOU

Presentation by 梁廣廷

