PD LAB

ASSIGNMENT - 4

Name: Raunak Thanawala

Registration Number: 231070051

Branch: Computer Engineering

Batch: 3

**Aim:-**

Plot different graphs using matplotlib.

Select any dataset from data.world website and create corresponding dashboard using streamlit

**Theory:-**

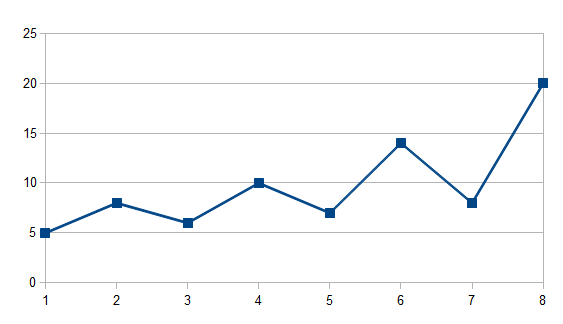
Matplotlib is a low level graph plotting library in python that serves as a visualization utility.

It was created by John D. Hunter.

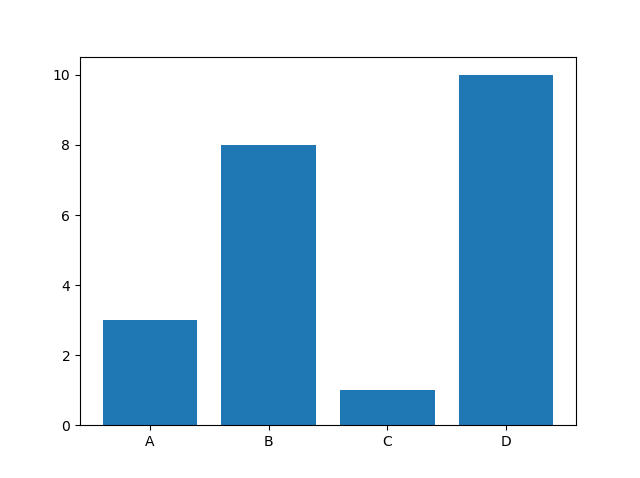
It is open source and we can use it freely.

Basic Types of Graphs in MatPlotLib:

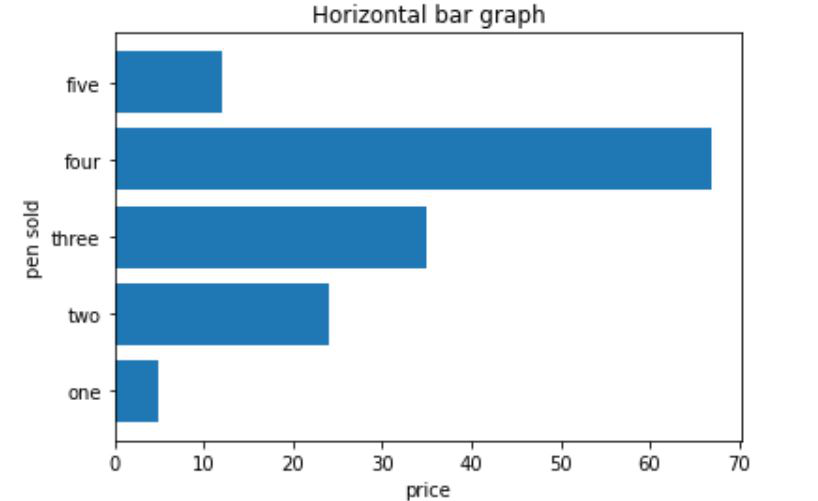
1.Line Graph



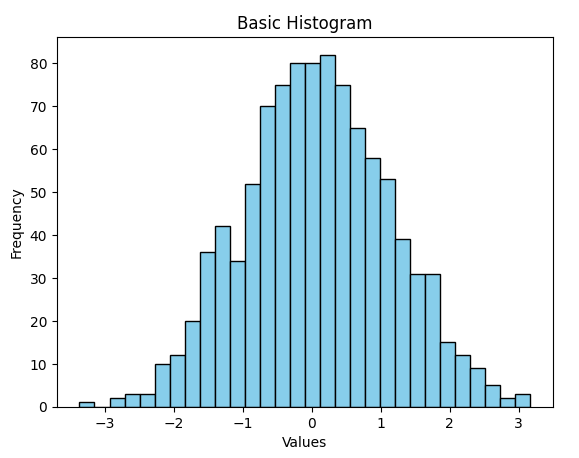
2.Bar Chart



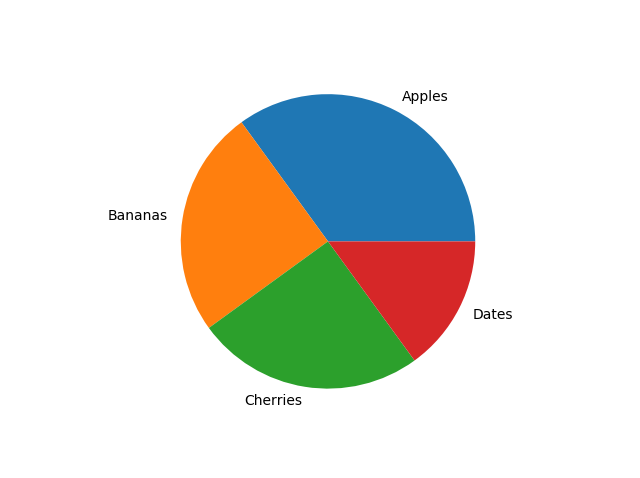
3.Horizontal Bar Chart:



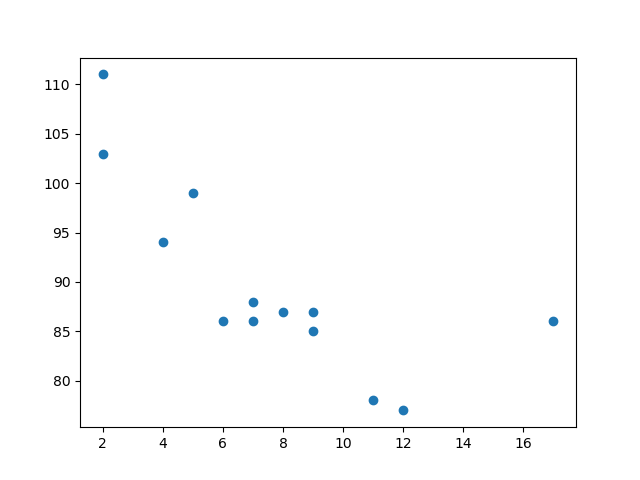
4. Histogram:



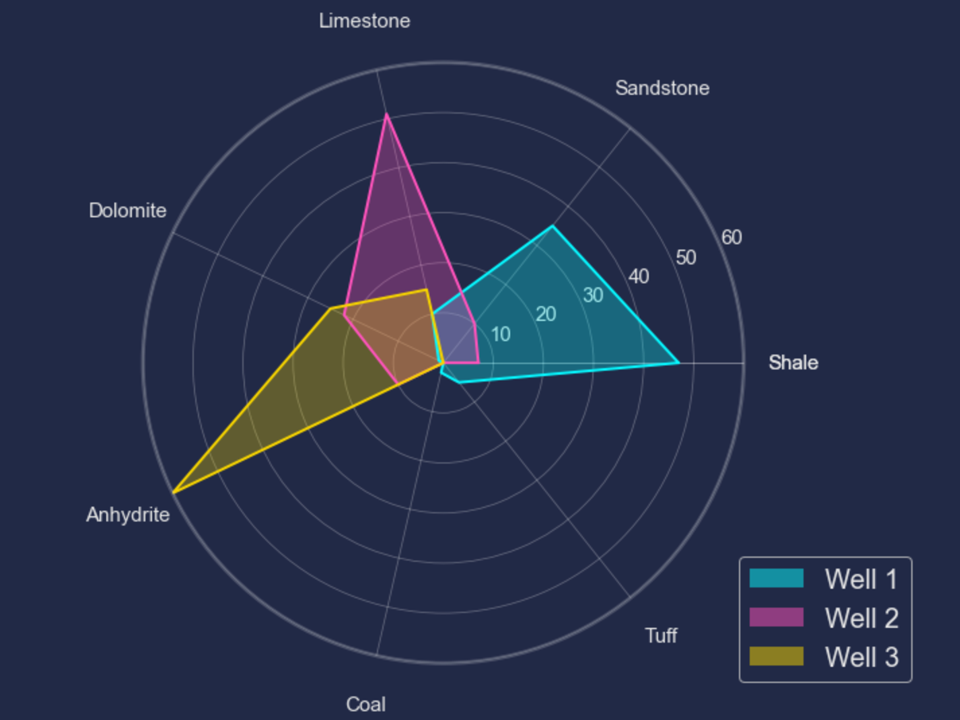
5. Pie Chart:



6.Scatter Plot:



7. Radar Chart:



**StreamLit:**

Streamlit is an open-source Python library that simplifies the creation of interactive web applications for data science and machine learning.

It enables developers to turn Python scripts into shareable apps with minimal coding. Streamlit provides a range of built-in widgets, such as sliders, buttons, and text inputs, for user interaction.

It seamlessly integrates with popular data visualization libraries like Matplotlib, allowing for dynamic and real-time data presentation.

With Streamlit, we can build apps that automatically update in response to code changes or user input, making it a powerful tool for data exploration and sharing insights.

**Code and Output:**

import streamlit as st

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

file\_path = 'vgsales.csv'

vgsales\_data = pd.read\_csv(file\_path)

st.sidebar.header('Filters')

selected\_genre = st.sidebar.multiselect(

'Select Genre/s',

options=vgsales\_data['Genre'].unique(),

default=[],

key='genre'

)

selected\_publisher = st.sidebar.multiselect(

'Select Publisher/s',

options=vgsales\_data['Publisher'].unique(),

default=[],

key='publisher'

)

selected\_platform = st.sidebar.multiselect(

'Select Platform/s',

options=vgsales\_data['Platform'].unique(),

default=[],

key='platform'

)

filtered\_data = vgsales\_data

if selected\_genre:

filtered\_data = filtered\_data[filtered\_data['Genre'].isin(selected\_genre)]

if selected\_publisher:

filtered\_data = filtered\_data[filtered\_data['Publisher'].isin(selected\_publisher)]

if selected\_platform:

filtered\_data = filtered\_data[filtered\_data['Platform'].isin(selected\_platform)]

st.header('Video Game Sales Dashboard')

st.subheader('Top 10 Games in Sales')

top\_games = filtered\_data.sort\_values('Global\_Sales', ascending=False).head(10)

fig, ax = plt.subplots()

bars = ax.barh(top\_games['Name'], top\_games['Global\_Sales'], color='skyblue', edgecolor='white')

ax.set\_xlabel('Global Sales (in millions)')

ax.set\_title('Top 10 Games in Sales')

ax.invert\_yaxis()

for bar in bars:

bar.set\_edgecolor('white')

bar.set\_linewidth(1.5)

ax.tick\_params(axis='both', colors='white')

ax.title.set\_color('white')

st.pyplot(fig, transparent=True)

st.subheader('Sales by Genre')

genre\_sales = filtered\_data.groupby('Genre')['Global\_Sales'].sum()

fig, ax = plt.subplots()

ax.pie(genre\_sales, labels=genre\_sales.index, autopct='%1.1f%%', startangle=140)

ax.axis('equal')

fig.patch.set\_alpha(0)

ax.patch.set\_alpha(0)

ax.set\_facecolor('none')

fig.patch.set\_facecolor('none')

for text in ax.texts:

text.set\_color('white')

st.pyplot(fig, transparent=True)

st.subheader('Sales Trends Over Time')

sales\_over\_time = filtered\_data.groupby('Year')['Global\_Sales'].sum()

fig, ax = plt.subplots()

ax.plot(sales\_over\_time.index, sales\_over\_time, marker='o', linestyle='-', color='skyblue', linewidth=1.5)

ax.set\_xlabel('Year', color='white')

ax.set\_ylabel('Global Sales (in millions)', color='white')

ax.set\_title('Sales Over the Years', color='white')

ax.tick\_params(axis='both', colors='white')

fig.patch.set\_alpha(0)

ax.patch.set\_alpha(0)

ax.set\_facecolor('none')

fig.patch.set\_facecolor('none')

st.pyplot(fig, transparent=True)

st.sidebar.header('Game Analysis')

selected\_game = st.sidebar.selectbox(

'Select a Game',

options=filtered\_data['Name'].unique(),

key='game'

)

game\_data = filtered\_data[filtered\_data['Name'] == selected\_game].iloc[0]

if not game\_data.empty:

st.subheader(f'Sales Distribution for {selected\_game}')

labels = ['NA\_Sales', 'EU\_Sales', 'JP\_Sales', 'Other\_Sales']

values = [game\_data[label] for label in labels]

num\_vars = len(labels)

angles = np.linspace(0, 2 \* np.pi, num\_vars, endpoint=False).tolist()

values += values[:1]

angles += angles[:1]

fig, ax = plt.subplots(figsize=(6, 6), subplot\_kw=dict(polar=True))

ax.fill(angles, values, color='skyblue', alpha=0.25)

ax.plot(angles, values, color='skyblue', linewidth=1.5)

ax.set\_yticklabels([])

ax.set\_xticks(angles[:-1])

ax.set\_xticklabels(labels, color='white')

ax.set\_title(f'Sales by Region of {selected\_game}', color='white')

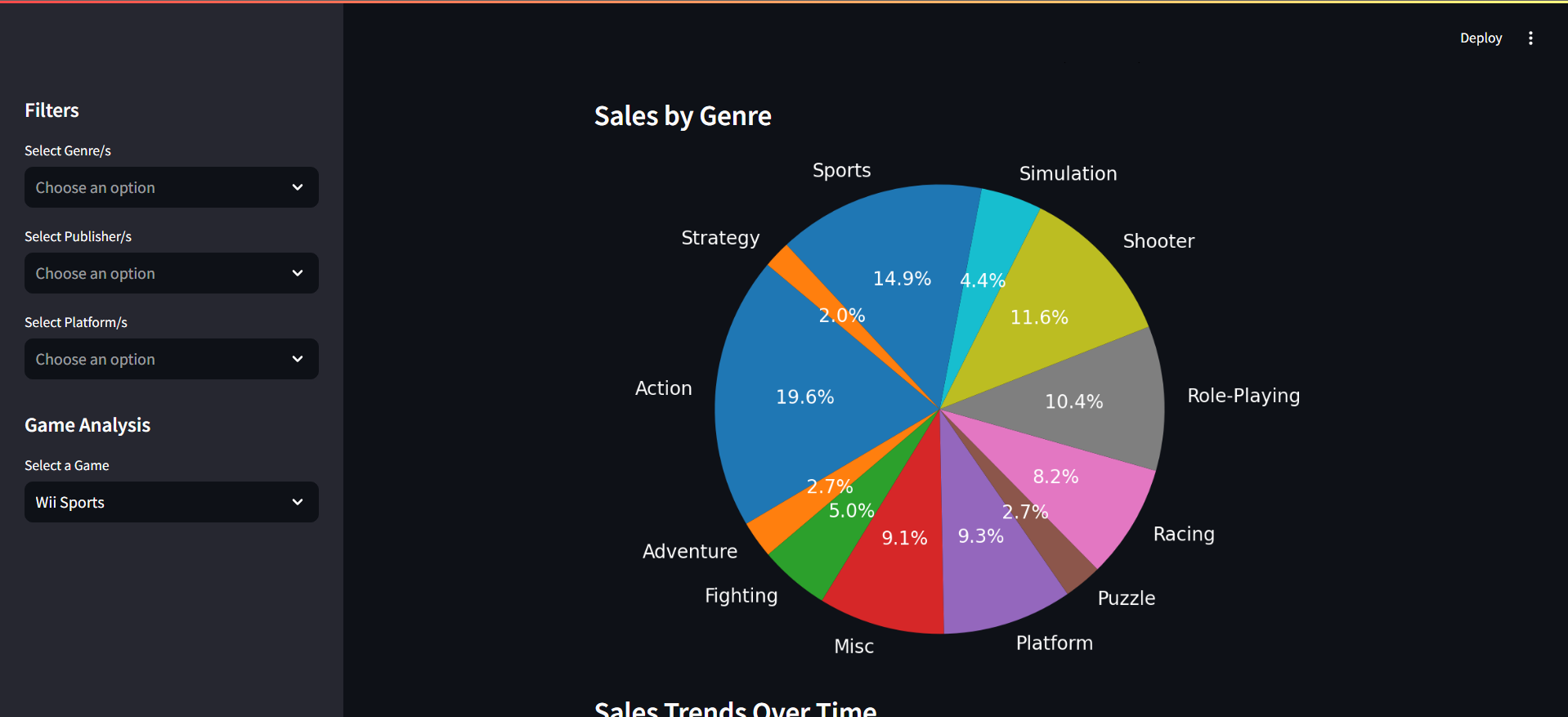
st.pyplot(fig, transparent=True)

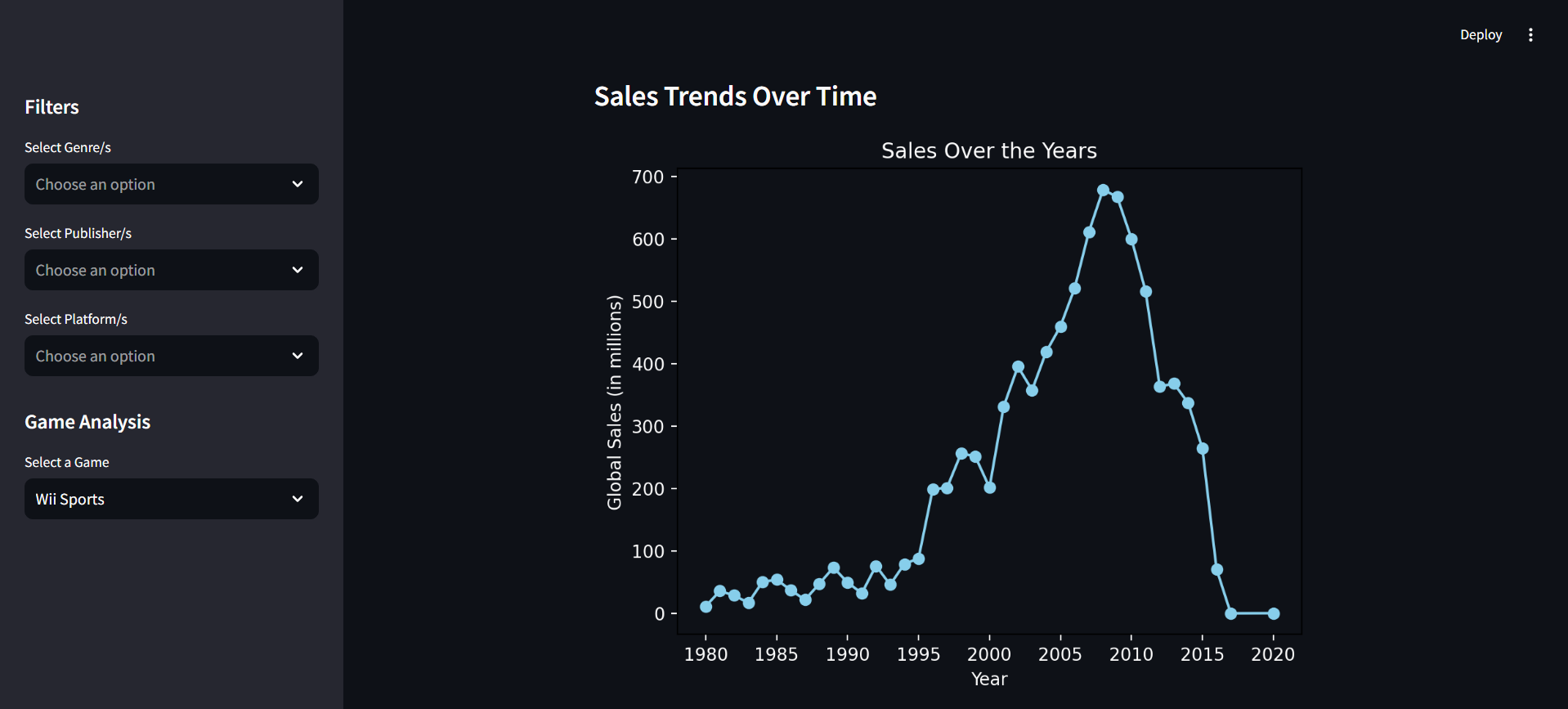
st.subheader('Raw Data: ')

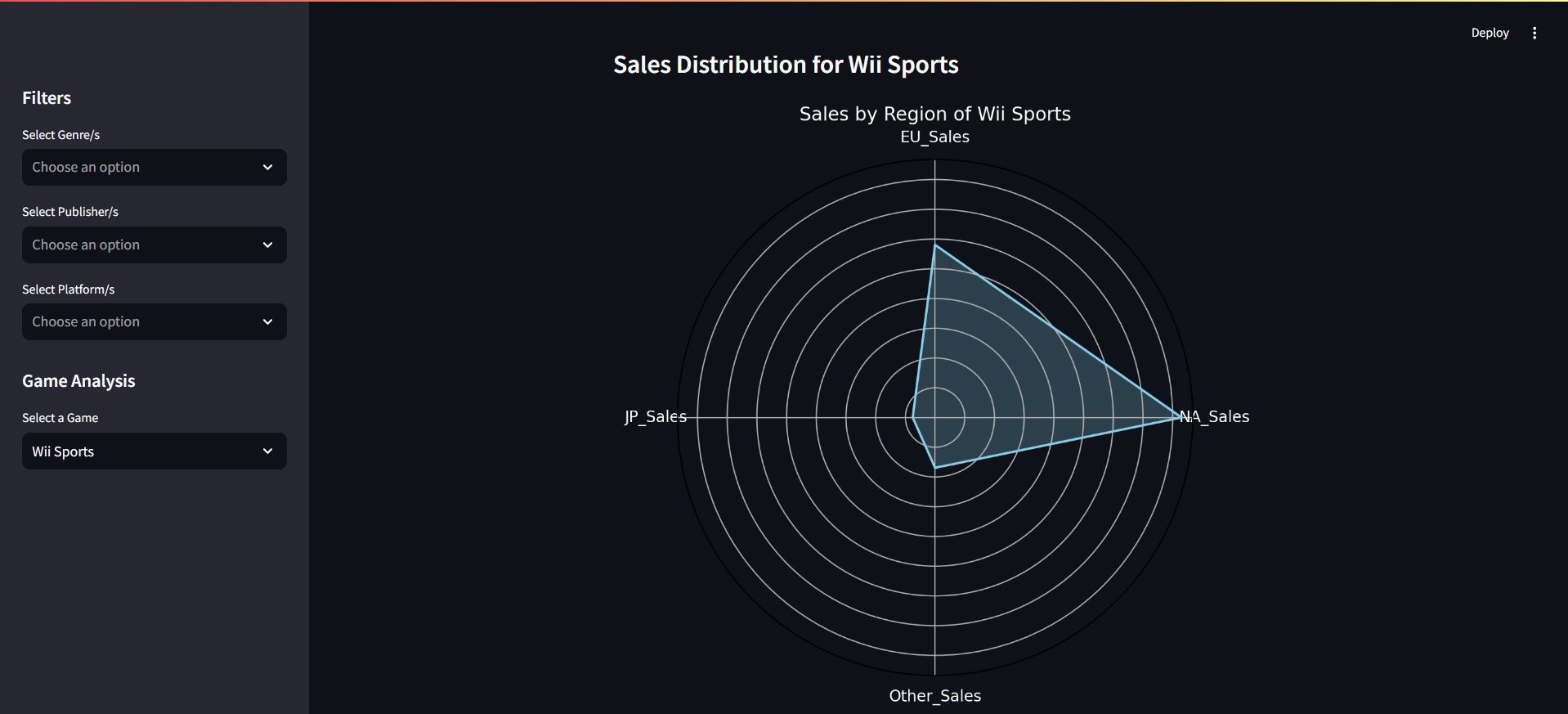
st.dataframe(filtered\_data)

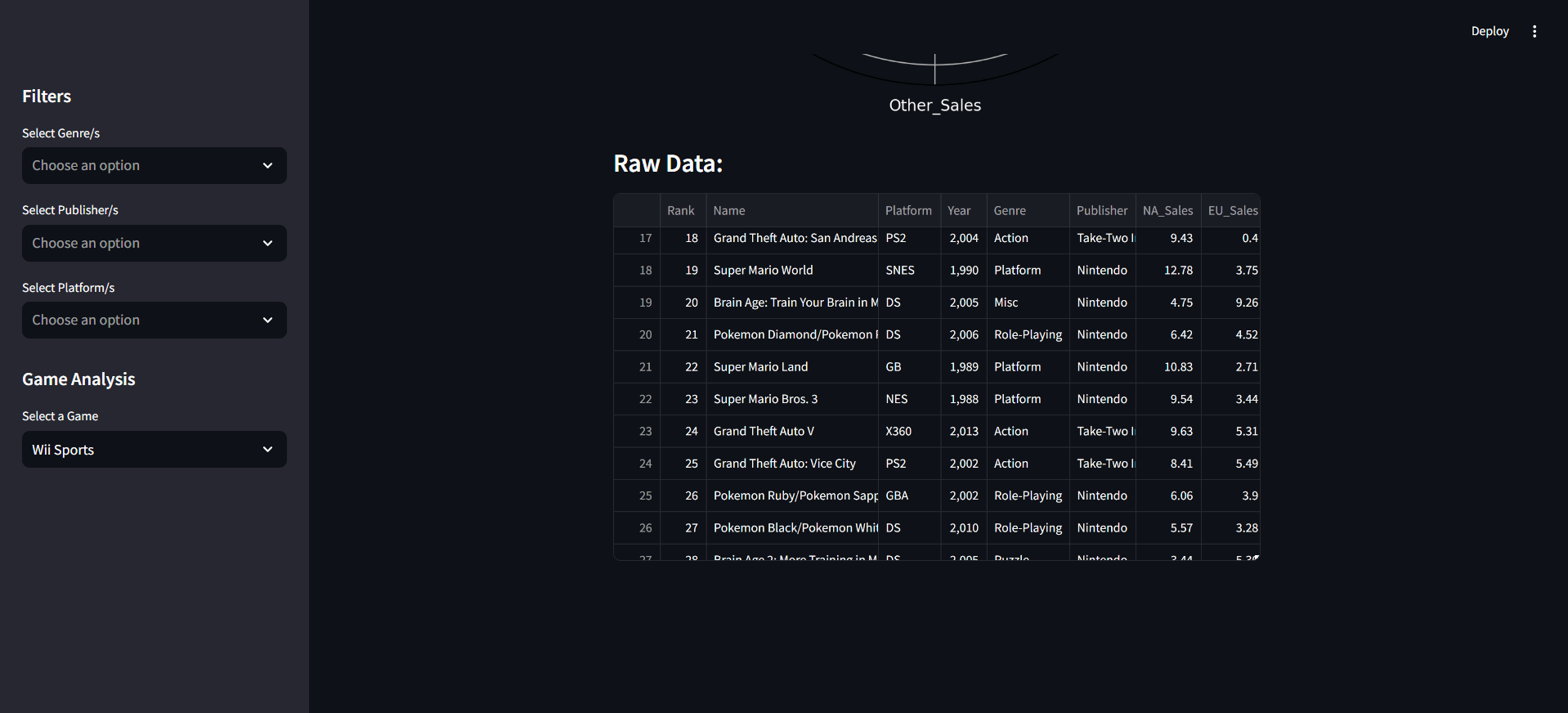


Output when we open it directly

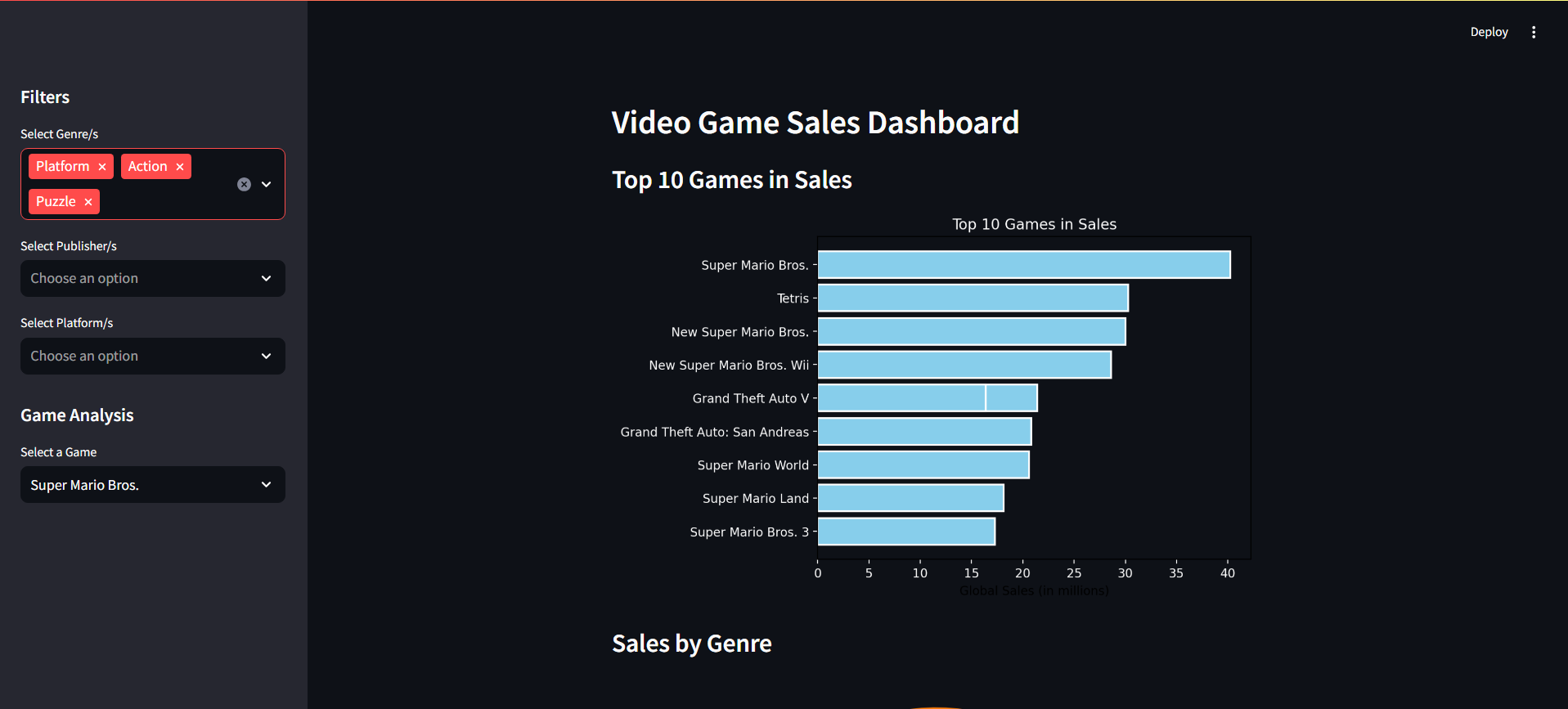


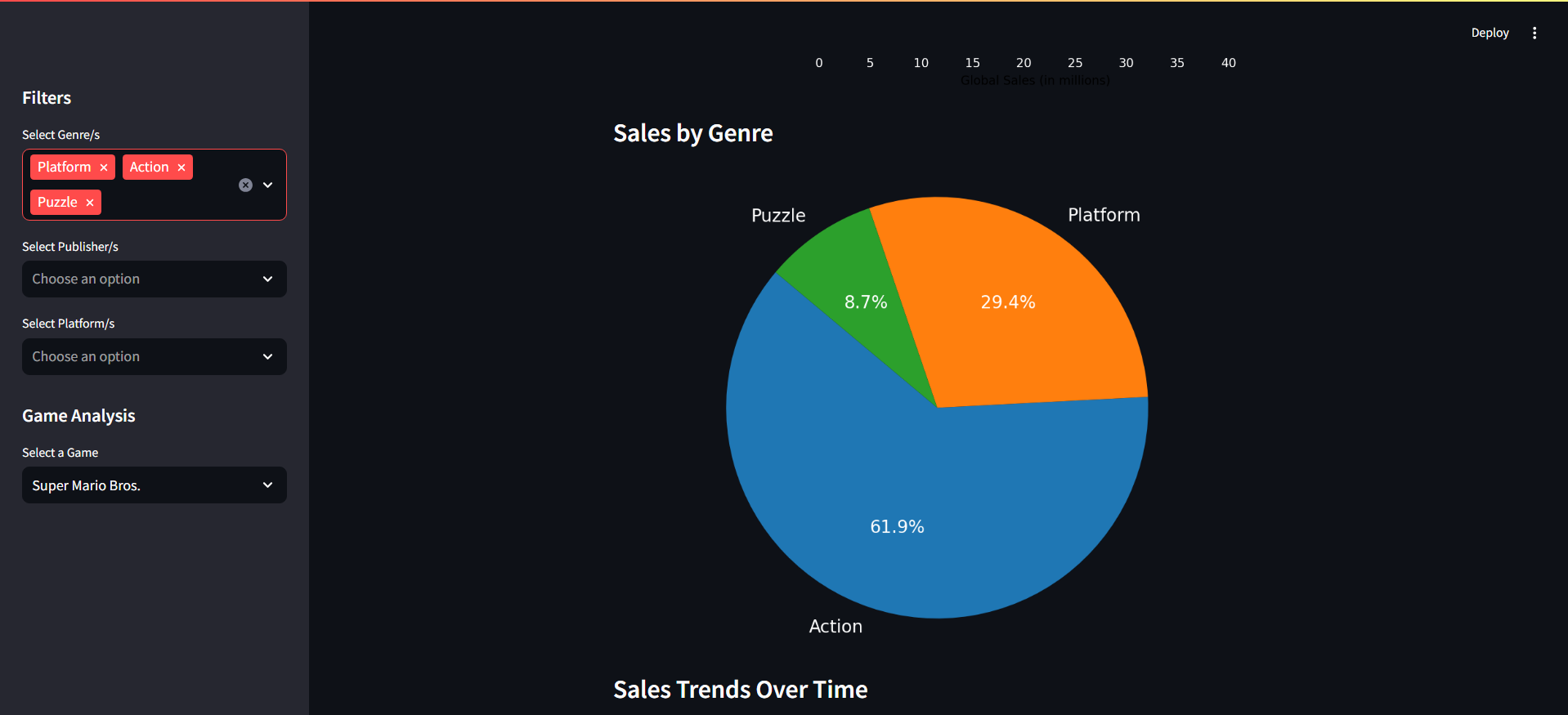


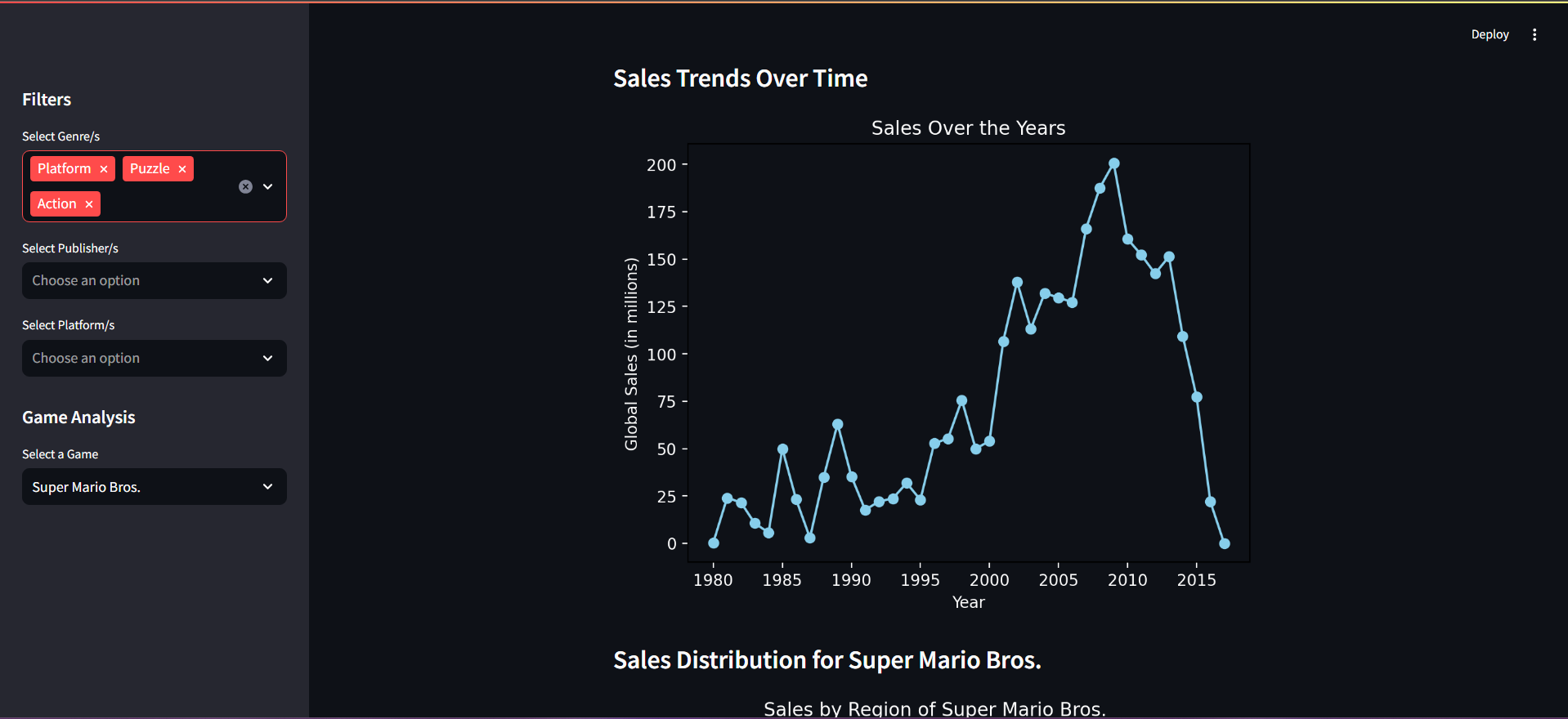


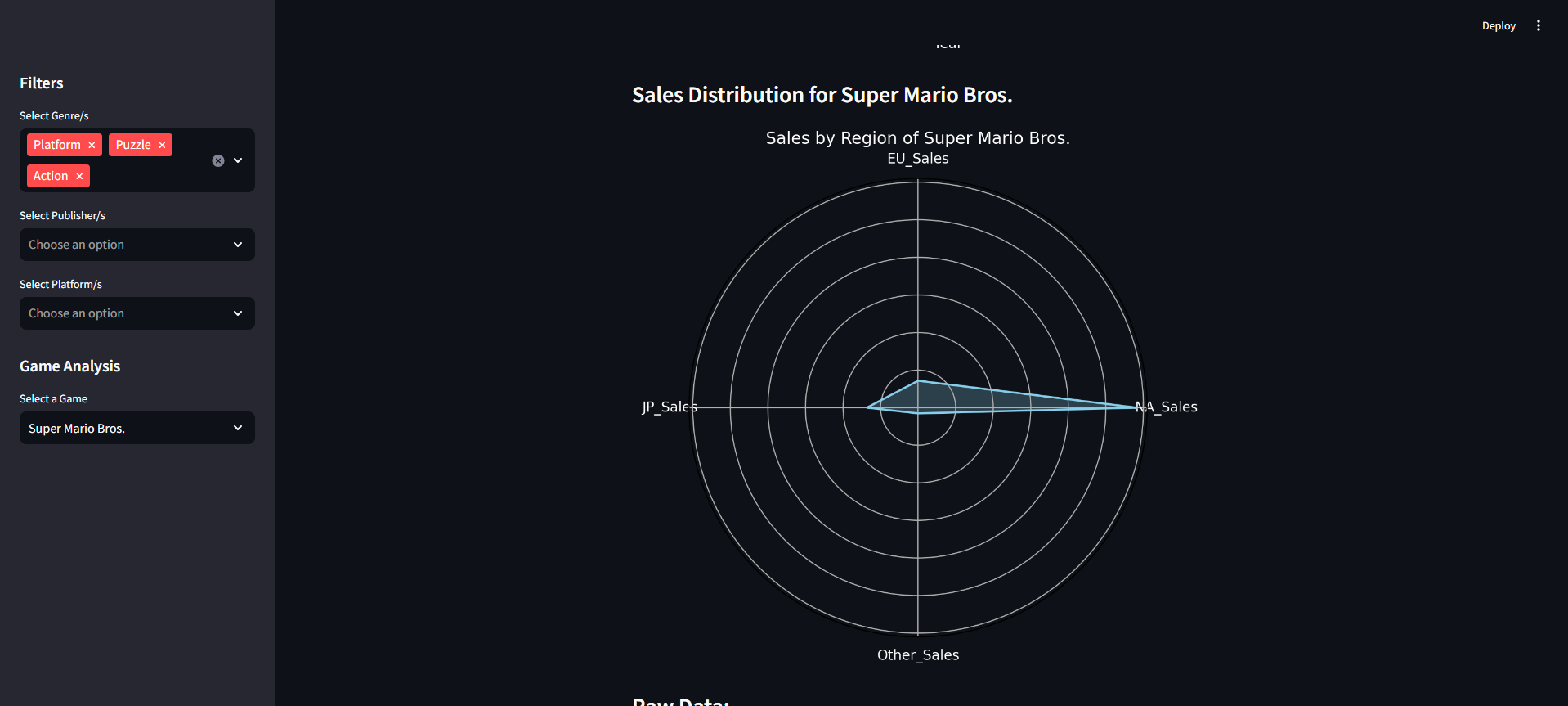


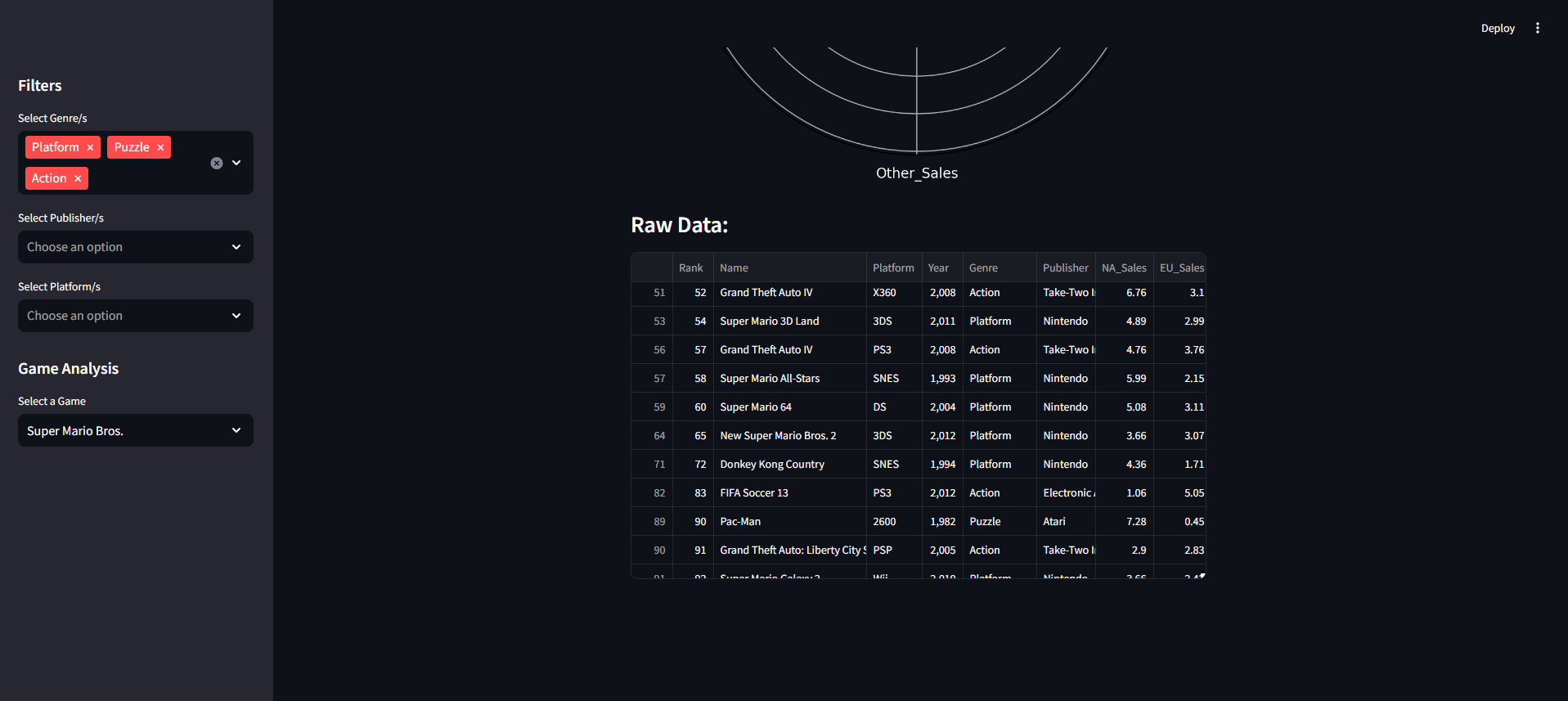
When we select a couple of genres to filter



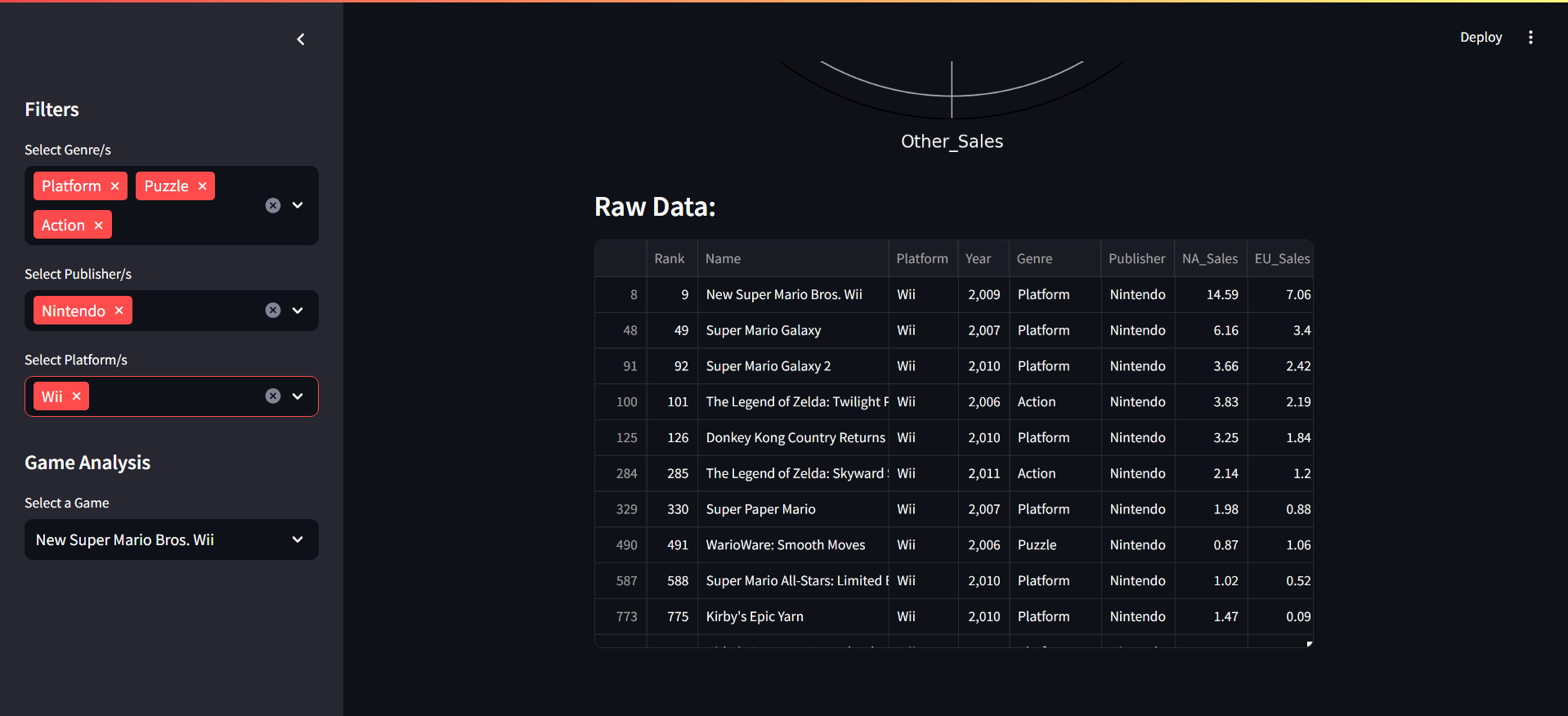
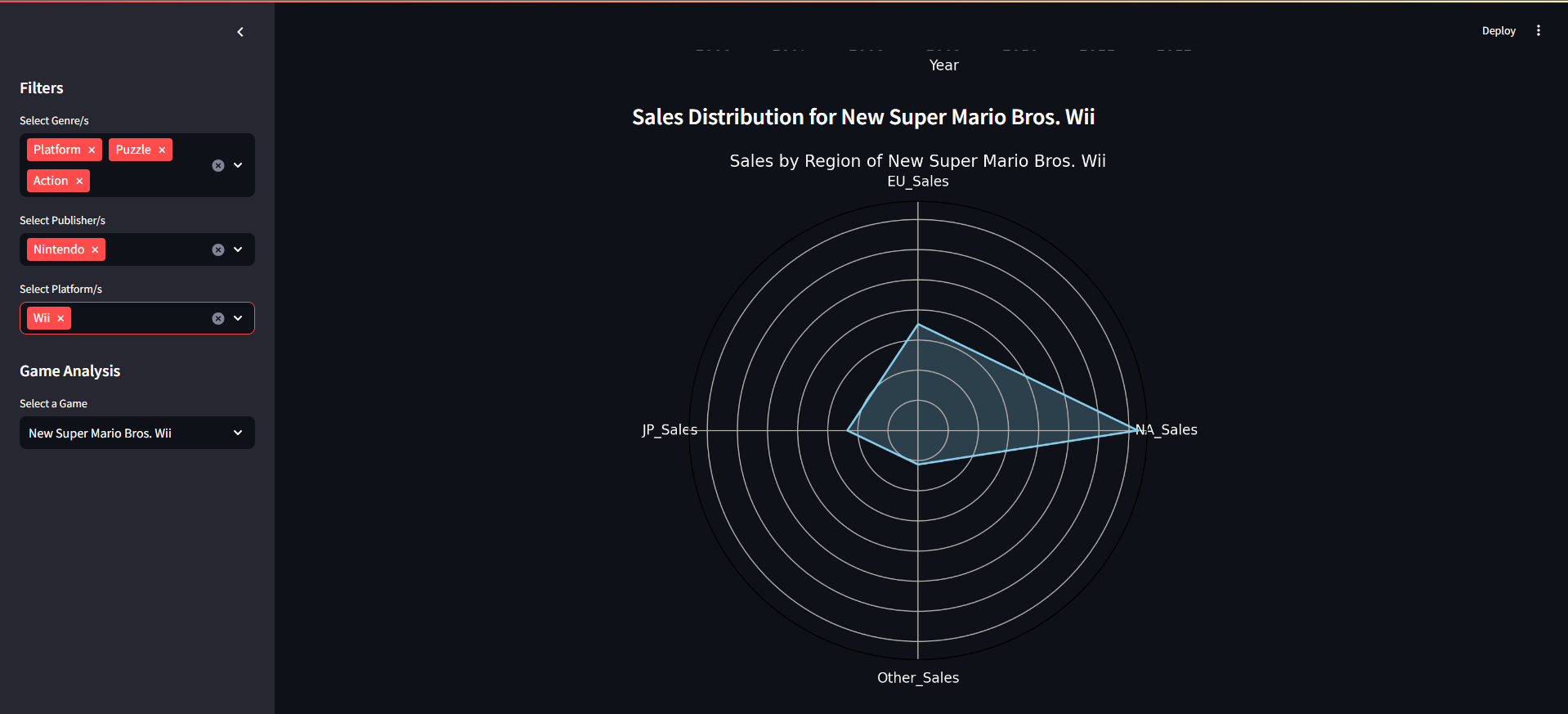
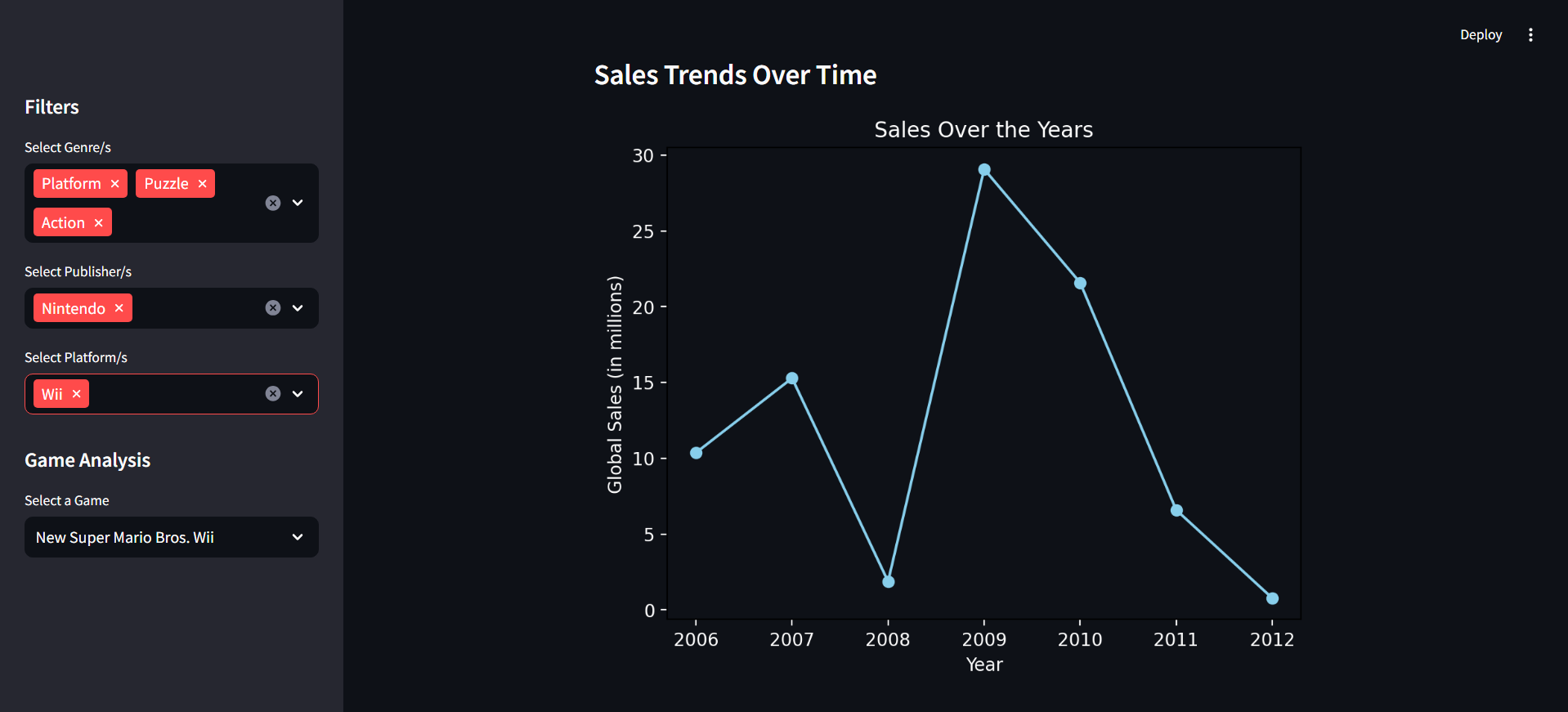
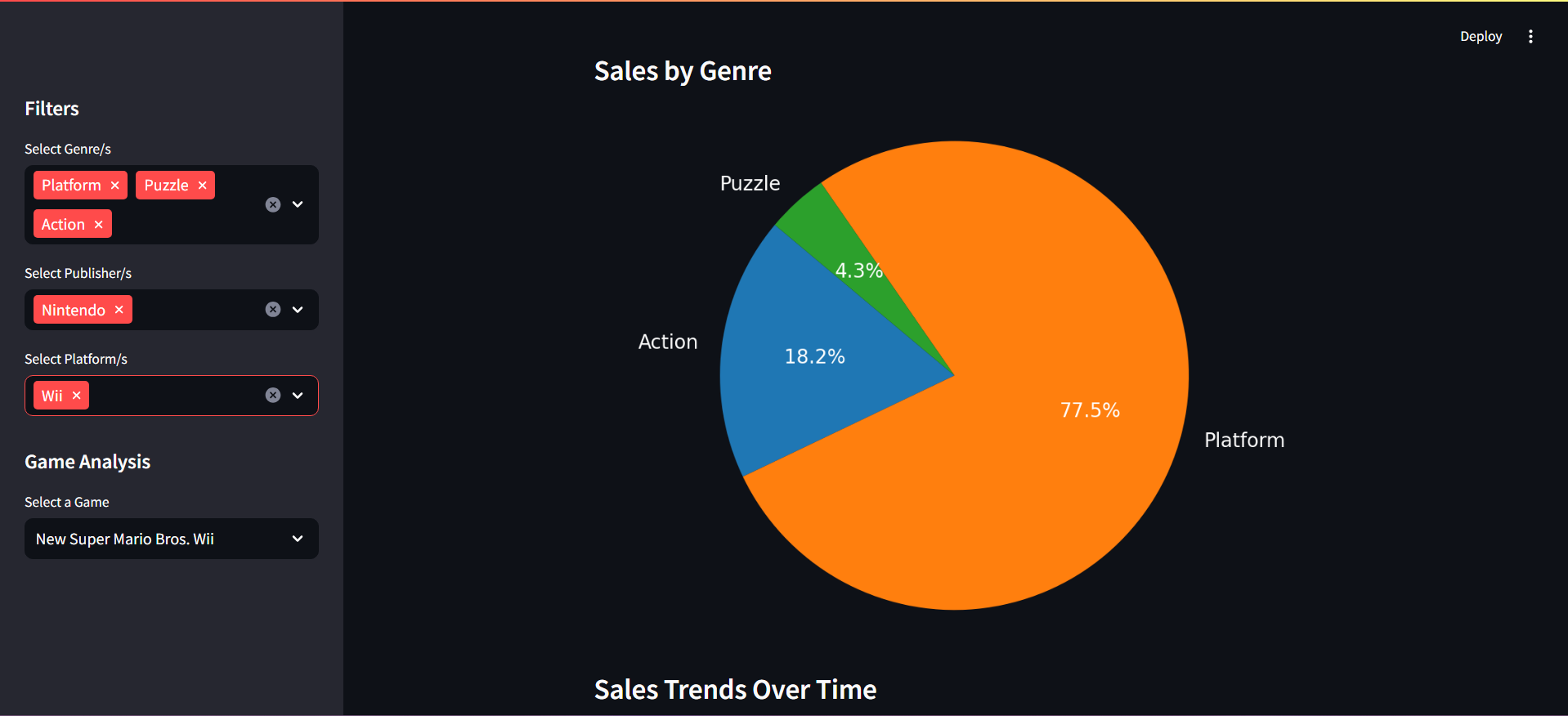




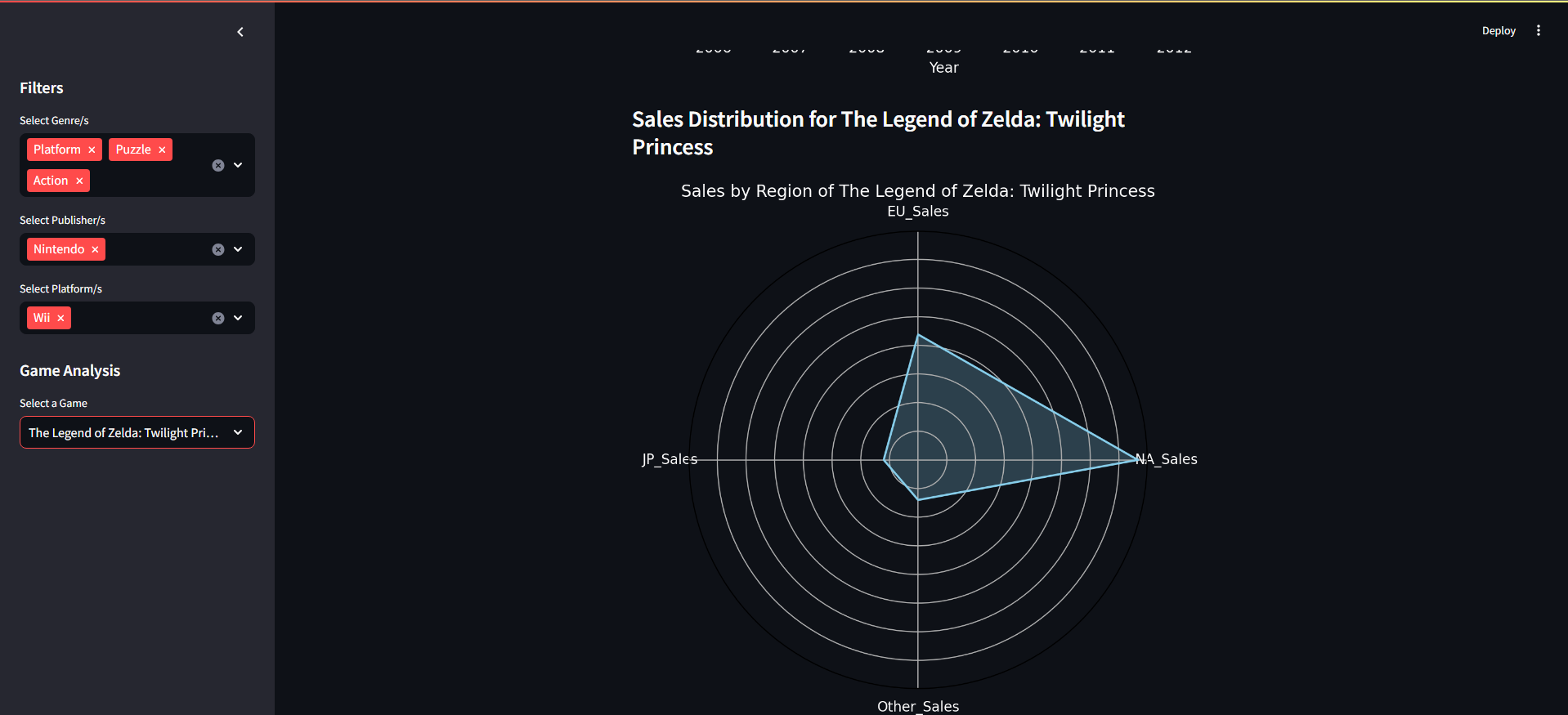




When we also select a Publisher and a Platform



When we select a specific game the radar chart shows its sales per region



**Conclusion:**

Thus we have written a program to make a dashboard with the help of matplotlib to plot some graphs to make it look more appealing and also with streamlit to make the dashboard more seamless.

The dashboard we made was for video game sales for which the dataset was found on kaggle in the format of a csv file.