# Import the necessary library

import warnings

warnings.filterwarnings('ignore')

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

sns.set\_style('darkgrid')

import os

# List files in the input directory to verify the file path

df = None # Initialize df to None

for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

file\_path = os.path.join(dirname, filename)

print(f"Found file: {file\_path}") # Print the found file path

# Assuming your target file is found in this loop, you can load it here

if 'steam\_game\_data.csv' in filename:

# Load the dataset

df = pd.read\_csv(file\_path)

print(f"Loaded dataset from: {file\_path}")

# You might want to break the loop once the file is found and loaded

# break

# If the file was not found and loaded inside the loop, df will not be defined.

# You can add a check here or ensure the loop logic correctly identifies your file.

# For example, if the file is guaranteed to be at a specific location:

# df = pd.read\_csv('/kaggle/input/steam-top-20-weekly-game/steam\_game\_data.csv') # Use this if you are absolutely sure of the path

# Add a check to see if df was loaded

if df is not None:

# Check the shape of the dataset

print("Dataset shape:", df.shape)

# Display data types of each column

print("\nData types:\n", df.dtypes)

# Check for missing values

print("\nMissing values:\n", df.isnull().sum())

# Handle missing values for release\_date and user\_review\_percentage with placeholders

df['release\_date'] = df['release\_date'].fillna("Not Released")

df['user\_review\_percentage'] = df['user\_review\_percentage'].fillna("N/A")

# For all\_time\_peak, convert to numeric if needed and fill missing with 0

df['all\_time\_peak'] = pd.to\_numeric(df['all\_time\_peak'], errors='coerce').fillna(0)

# Verify that missing values have been handled

print("Missing values after handling:")

print(df.isnull().sum())

# Move plotting code inside this block

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

sns.barplot(data=df, x='title', y='all\_time\_peak')

plt.xlabel("Games")

plt.ylabel("All Time Peak")

plt.title("Weekly Top 20 Games All Time Peak")

plt.xticks(rotation=45, ha='right')

plt.show() # Added plt.show() to display the plot

# Remove non-numeric characters, replace empty strings with NaN, then convert to float

# This code block was moved inside the 'if df is not None:' block

# to ensure df is a DataFrame when these operations are performed.

df['review\_pct\_numeric'] = pd.to\_numeric(

df['user\_review\_percentage']

.str.replace(r'[^0-9.]', '', regex=True)

.replace('', np.nan),

errors='coerce'

)

# Optionally, fill NaN values with 0 (or any other default value)

df['review\_pct\_numeric'] = df['review\_pct\_numeric'].fillna(0)

# Verify the conversion

print(df[['title', 'user\_review\_percentage', 'review\_pct\_numeric']].head())

else:

print("Error: 'steam\_game\_data.csv' not found in the input directory.")

# You might want to add further handling here, like sys.exit() or raising an exception

# Create a scatter plot

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

#plt.scatter(df['review\_pct\_numeric'], df['all\_time\_peak'], s=100, alpha=0.7, color='limegreen')

# Annotate each point with the game title

#for i, row in df.iterrows():

#plt.text(row['review\_pct\_numeric'] + 0.5, row['all\_time\_peak'], row['title'], fontsize=9)

#plt.xlabel('User Review Percentage')

#plt.ylabel('All Time Peak')

#plt.title('All Time Peak vs. User Review Percentage (Game Titles Annotated)')

plt.grid(True)

plt.tight\_layout()

plt.show()