

```
In [2]: # LEGO Set Explorer Interactive Tool
# Comprehensive Python Script with Detailed Comments and Advanced Visualizations

# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import ipywidgets as widgets
from IPython.display import display, clear_output

# Step 1: Load and Prepare the Data
lego_df = pd.read_csv('lego_sets.csv')

# Remove unwanted columns
lego_df.drop(['minifigs', 'bricksetURL', 'thumbnailURL'], axis=1, inplace=True)

# Filter out records with missing values in critical columns
lego_df = lego_df.dropna(subset=['US_retailPrice', 'agerange_min', 'pieces', 'imageUrl'])

# Convert relevant columns to correct data types
lego_df['US_retailPrice'] = pd.to_numeric(lego_df['US_retailPrice'], errors='coerce')
lego_df['agerange_min'] = pd.to_numeric(lego_df['agerange_min'], errors='coerce')
lego_df['pieces'] = pd.to_numeric(lego_df['pieces'], errors='coerce')

# Create Age Range Column
lego_df['Age Range'] = pd.cut(lego_df['agerange_min'], bins=[0, 4, 9, 17, 100], labels=['1 to 4', '5 to 9', '10 to 17', 'Over 18'])

# Create Price Range Column
lego_df['Price Range'] = pd.cut(lego_df['US_retailPrice'], bins=[0, 25, 50, 100, 500, float('inf')], labels=['$', '$$', '$$$', '$$$$', '$$$$$'])

# Add Measures
total_sets = lego_df['set_id'].nunique()
total_groups = lego_df['themeGroup'].nunique()
avg_age = lego_df['agerange_min'].mean()
avg_price = lego_df['US_retailPrice'].mean()
avg_pieces = lego_df['pieces'].mean()

# Step 2: Design the Report Layout & Visuals
# Display basic statistics
print(f'Total Sets: {total_sets}')
print(f'Total Theme Groups: {total_groups}')
print(f'Average Age: {avg_age:.2f}')
print(f'Average Price: ${avg_price:.2f}')
print(f'Average Pieces: {avg_pieces:.2f}')

# Create a price vs pieces scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=lego_df, x='pieces', y='US_retailPrice', hue='Age Range', alpha=0.6)
plt.title('Price vs Pieces by Age Range')
```



```

plt.xlabel('Number of Pieces')
plt.ylabel('Price ($)')
plt.grid(True)
plt.show()

# Step 3: Add Interactive Components
# Interactive filter for Max Price
max_price_slider = widgets.IntSlider(value=850, min=0, max=850, step=5, description='Max Price:')

# Dropdowns for filtering
theme_group_dropdown = widgets.Dropdown(options=lego_df['themeGroup'].unique(), description='Theme Group:')
age_range_dropdown = widgets.Dropdown(options=lego_df['Age Range'].unique(), description='Age Range:')

# Function to filter and display the data
def filter_data(max_price, theme_group, age_range):
    filtered_df = lego_df[(lego_df['US_retailPrice'] <= max_price)]
    if theme_group:
        filtered_df = filtered_df[filtered_df['themeGroup'] == theme_group]
    if age_range:
        filtered_df = filtered_df[filtered_df['Age Range'] == age_range]

    fig = px.scatter(filtered_df, x='pieces', y='US_retailPrice', color='theme',
                    title='Filtered LEGO Sets', hover_data=['name', 'year'])
    fig.show()

# Display interactive widgets
ui = widgets.VBox([max_price_slider, theme_group_dropdown, age_range_dropdown])
out = widgets.interactive_output(filter_data, {
    'max_price': max_price_slider,
    'theme_group': theme_group_dropdown,
    'age_range': age_range_dropdown
})

display(ui, out)

# Final Step: Answer the Project Question
disney_sets_5_9 = lego_df[(lego_df['theme'] == 'Disney') & (lego_df['Age Range'] == '5 to 9')]
avg_disney_pieces = disney_sets_5_9['pieces'].mean()
print(f"Average number of pieces for Disney-themed LEGO sets targeting 5 to 9 year-olds: {avg_disney_pieces:.0f}")

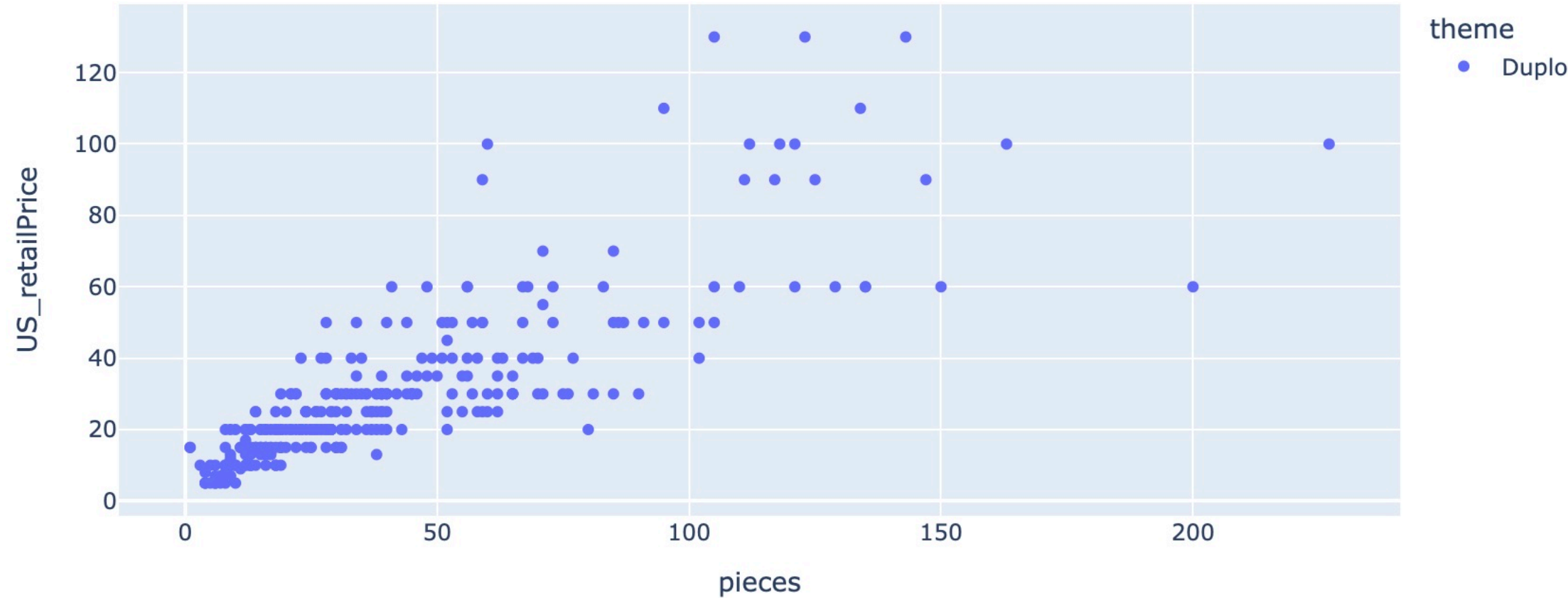
```

Total Sets: 4385
 Total Theme Groups: 15
 Average Age: 6.97
 Average Price: \$44.74
 Average Pieces: 410.98



Average number of pieces for Disney-themed LEGO sets targeting 5 to 9 year-olds: 216

Filtered LEGO Sets



Max Price: 850

Theme Gro... ▼

Age Range: ▼

Filtered LEGO Sets

