In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
# Load the dataset
df = pd.read_csv('boat_data.csv')
```

In [3]:

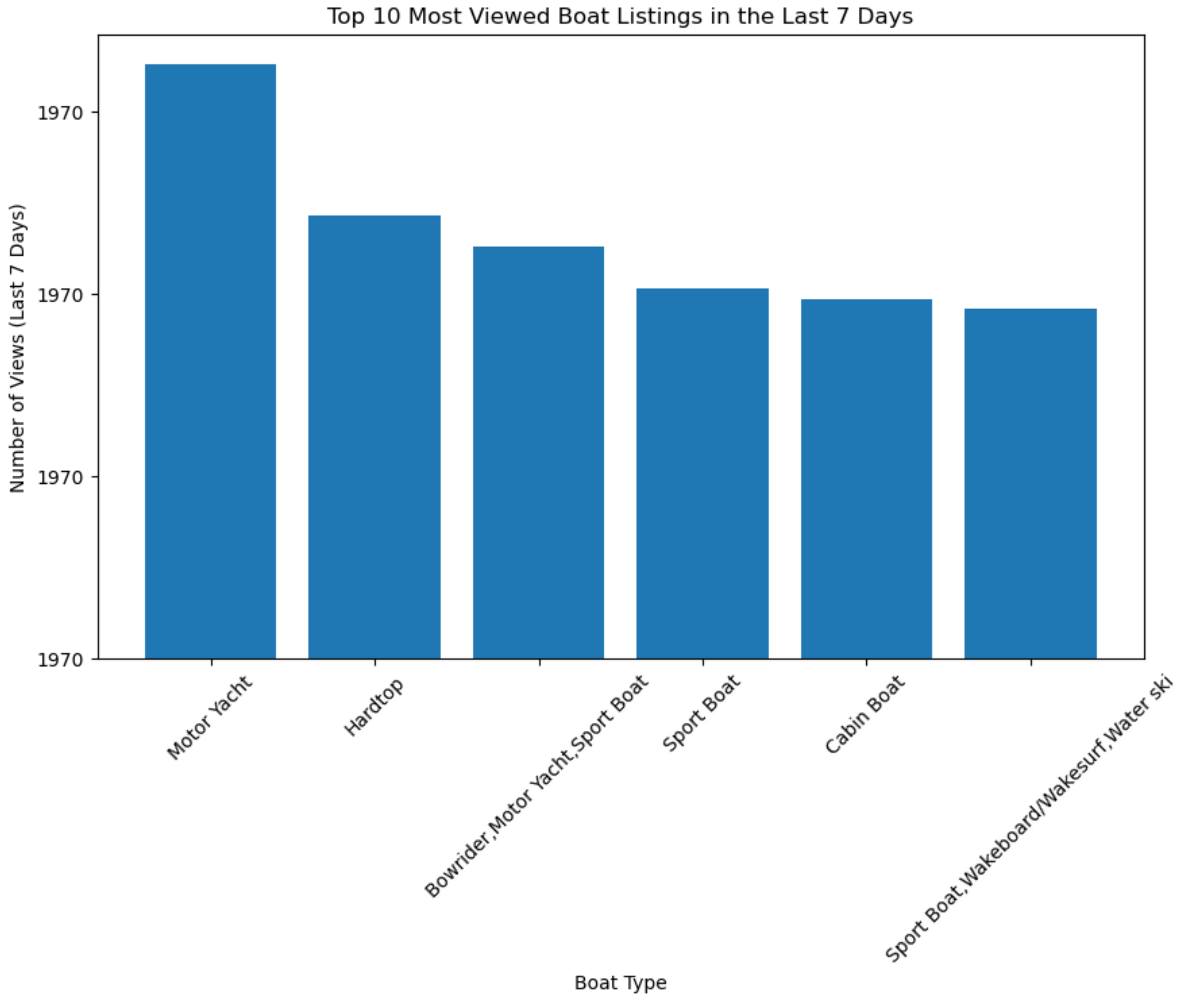
```
# Display the first few rows of the dataset to understand its structure
print(df.head())
```

```
       Price            Boat Type          Manufacturer                 Type  \
0   CHF 3337          Motor Yacht  Rigiflex power boats  new boat from stock   
1   EUR 3490  Center console boat     Terhi power boats  new boat from stock   
2   CHF 3770           Sport Boat    Marine power boats  new boat from stock   
3  DKK 25900           Sport Boat    Pioner power boats  new boat from stock   
4   EUR 3399         Fishing Boat    Linder power boats  new boat from stock   

   Year Built  Length  Width       Material  \
0        2017    4.00   1.90            NaN   
1        2020    4.00   1.50  Thermoplastic   
2           0    3.69   1.42      Aluminium   
3        2020    3.00   1.00            NaN   
4        2019    3.55   1.46      Aluminium   

                                     Location  Number of views last 7 days  
0      Switzerland Â» Lake Geneva Â» VÃ©senaz                          226  
1                    Germany Â» BÃ¶nningstedt                           75  
2  Switzerland Â» Lake of Zurich Â» StÃ¤fa ZH                          124  
3                        Denmark Â» Svendborg                           64  
4               Germany Â» Bayern Â» MÃ¼nchen                           58  
```

In [7]:

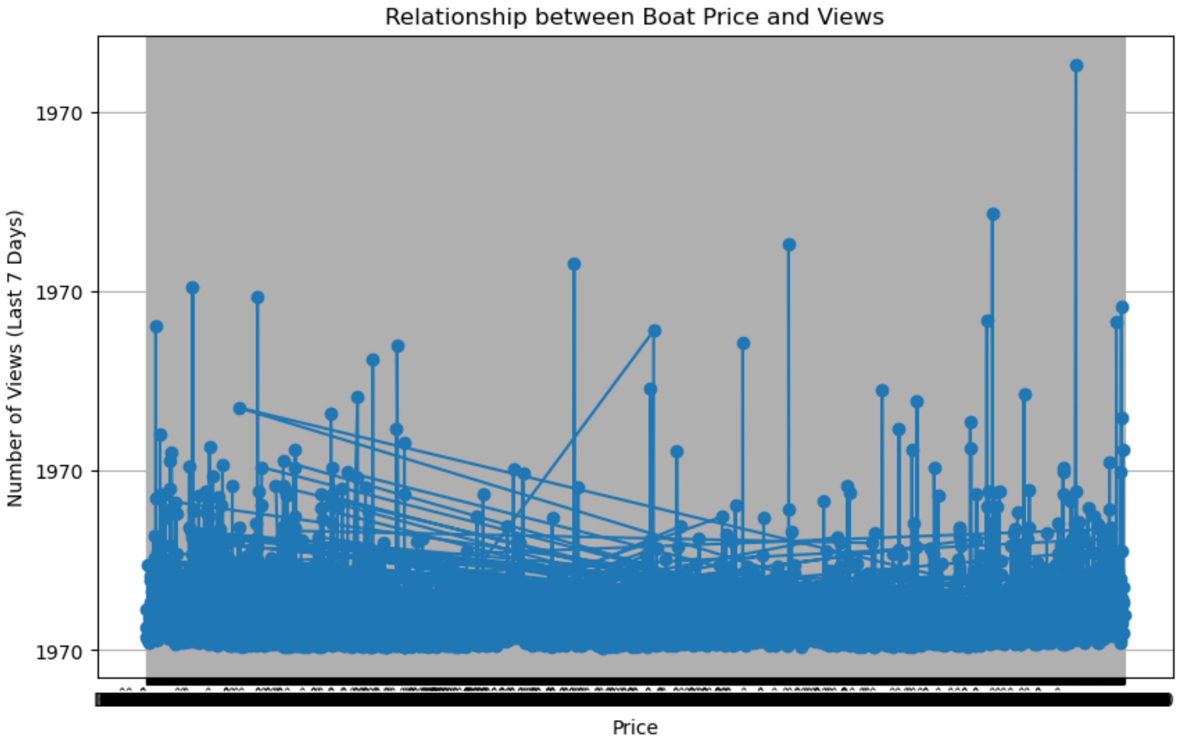
```
# Filter data for the last 7 days based on 'Number of views in the last 7 days'
recent_data = df.nlargest(10, 'Number of views last 7 days')

# Display the top 10 most viewed boat listings
print(recent_data)

# Create a bar plot to visualize the number of views for the top 10 listings
plt.figure(figsize=(10, 6))
plt.bar(recent_data['Boat Type'], recent_data['Number of views last 7 days'])
plt.xlabel('Boat Type')
plt.ylabel('Number of Views (Last 7 Days)')
plt.title('Top 10 Most Viewed Boat Listings in the Last 7 Days')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()
```

```
           Price                                Boat Type  \
9580   CHF 14900                              Motor Yacht
8723   CHF 35000                                  Hardtop
6211  CHF 125900          Bowrider,Motor Yacht,Sport Boat
3700  EUR 949000                                  Hardtop
308    CHF 19900                               Sport Boat
894    CHF 48500                               Cabin Boat
9878    CHF 4999  Sport Boat,Wakeboard/Wakesurf,Water ski
8684   CHF 36000                               Sport Boat
9843    CHF 6500                               Sport Boat
35      CHF 3800                               Cabin Boat

                   Manufacturer                Type  Year Built  Length  \
9580       Bayliner power boats  Used boat,Unleaded        1992    7.70
8723       Princess power boats    Used boat,Diesel        1979   11.12
6211          Windy power boats    Used boat,Diesel        2002   12.35
3700       Pershing power boats    Used boat,Diesel        2009   20.30
308         Sea Ray power boats  Used boat,Unleaded        1993    6.14
894           Viper power boats  Used boat,Unleaded        2014    6.65
9878  Tullio Abbate power boats  Used boat,Unleaded        1980    6.00
8684  Correct Craft power boats  Used boat,Unleaded        2005    6.40
9843           Baha power boats  Used boat,Unleaded        1995    6.80
35            Draco power boats  Used boat,Unleaded        1980    6.20

      Width Material                                         Location  \
9580   2.46  Plastic                Switzerland Â» Le Landeron (NE)
8723   3.88      GRP     Switzerland Â» Neuenburgersee Â» Hauterive
6211   3.48      GRP  Switzerland Â» Lago Maggiore Â» 6600 Locarno
3700   5.20      GRP                    Neustadt in Holstein (Ostsee)
308    2.34  Plastic           Switzerland Â» Murtensee Â» Avenches
894    2.30      GRP                             Switzerland Â» Horn
9878   2.10      GRP          Switzerland Â» Lake of Zurich Â» Rafz
8684   2.31      GRP         Switzerland Â» Lago Maggiore Â» Ticino
9843   2.38      GRP                                             Thun
35     2.45      GRP          Switzerland Â» Walensee Â» Walenstadt

       Number of views last 7 days
9580 1970-01-01 00:00:00.000003263
8723 1970-01-01 00:00:00.000002432
6211 1970-01-01 00:00:00.000002261
3700 1970-01-01 00:00:00.000002154
308  1970-01-01 00:00:00.000002026
894  1970-01-01 00:00:00.000001970
9878 1970-01-01 00:00:00.000001917
8684 1970-01-01 00:00:00.000001834
9843 1970-01-01 00:00:00.000001831
35   1970-01-01 00:00:00.000001804
```



In [11]:

```
# Create a line plot to visualize the relationship between price and views
plt.figure(figsize=(10, 6))
plt.plot(df['Price'], df['Number of views last 7 days'], marker='o', linestyle='-'
plt.xlabel('Price')
plt.ylabel('Number of Views (Last 7 Days)')
plt.title('Relationship between Boat Price and Views')
plt.grid(True)
plt.show()
```



In [12]:
```
# Select the top most viewed boats in the last 7 days
top_viewed_boats = df.nlargest(10, 'Number of views last 7 days')
common_boat_types = top_viewed_boats['Boat Type'].value_counts()
print("Common Boat Types among the Top Viewed Boats:")
print(common_boat_types)
```

```
Common Boat Types among the Top Viewed Boats:
Sport Boat                                  3
Hardtop                                     2
Cabin Boat                                  2
Motor Yacht                                 1
Bowrider,Motor Yacht,Sport Boat             1
Sport Boat,Wakeboard/Wakesurf,Water ski     1
Name: Boat Type, dtype: int64
```

In [21]:
```
# Filter data for used boats (built before 2020)
used_boats = df[df['Year Built'] < 2020]

# Check for missing values in the 'length' column
missing_values = used_boats['Length'].isna().sum()

if missing_values > 0:
    # Remove rows with missing 'length' values
    used_boats = used_boats.dropna(subset=['Length'])

# Calculate the average length of used boats in feet
average_used_boat_length = used_boats['Length'].mean()

# Print the result
print(f"Average length of Used boats: {average_used_boat_length:.2f} meters")
```

```
Average length of Used boats: 12.01 meters
```

In [23]:
```
# Convert the "Number of views last 7 days" column to numeric (in case it's not alr
df['Number of views last 7 days'] = pd.to_numeric(df['Number of views last 7 days']

# Calculate the top 5 most popular boat types based on the number of views
popular_boat_types = df.groupby('Boat Type')['Number of views last 7 days'].sum().n
print("Top 5 Most Popular Boat Types:")
print(popular_boat_types)
```

```
Top 5 Most Popular Boat Types:
Boat Type
Motor Yacht    351659
Sport Boat     251418
Cabin Boat     131700
Flybridge      127534
Trawler         75456
Name: Number of views last 7 days, dtype: int64
```

In [25]:

```
# Analyze price distribution
# Clean the "Price" column by removing non-numeric characters and converting to num
df['Price'] = df['Price'].str.replace('[^\d.]', '', regex=True)  # Remove non-numer
df['Price'] = pd.to_numeric(df['Price'], errors='coerce')  # Convert to numeric, No

# Analyze price distribution
median_price = df['Price'].median()
std_deviation_price = df['Price'].std()
print(f"Median Price: {median_price:.2f}")
print(f"Standard Deviation of Price: {std_deviation_price:.2f}")
```

```
Median Price: 95000.00
Standard Deviation of Price: 1007482.24
```

In [26]:

```
# Identify the location with the highest number of boat listings
most_common_location = df['Location'].mode().values[0]

# Calculate the average price of boats in that location
average_price_in_location = df[df['Location'] == most_common_location]['Price'].mea

print(f"Location with the Most Listings: {most_common_location}")
print(f"Average Price in {most_common_location}: {average_price_in_location:.2f}")
```

```
Location with the Most Listings: Netherlands Â» In verkoophaven
Average Price in Netherlands Â» In verkoophaven: 138652.74
```

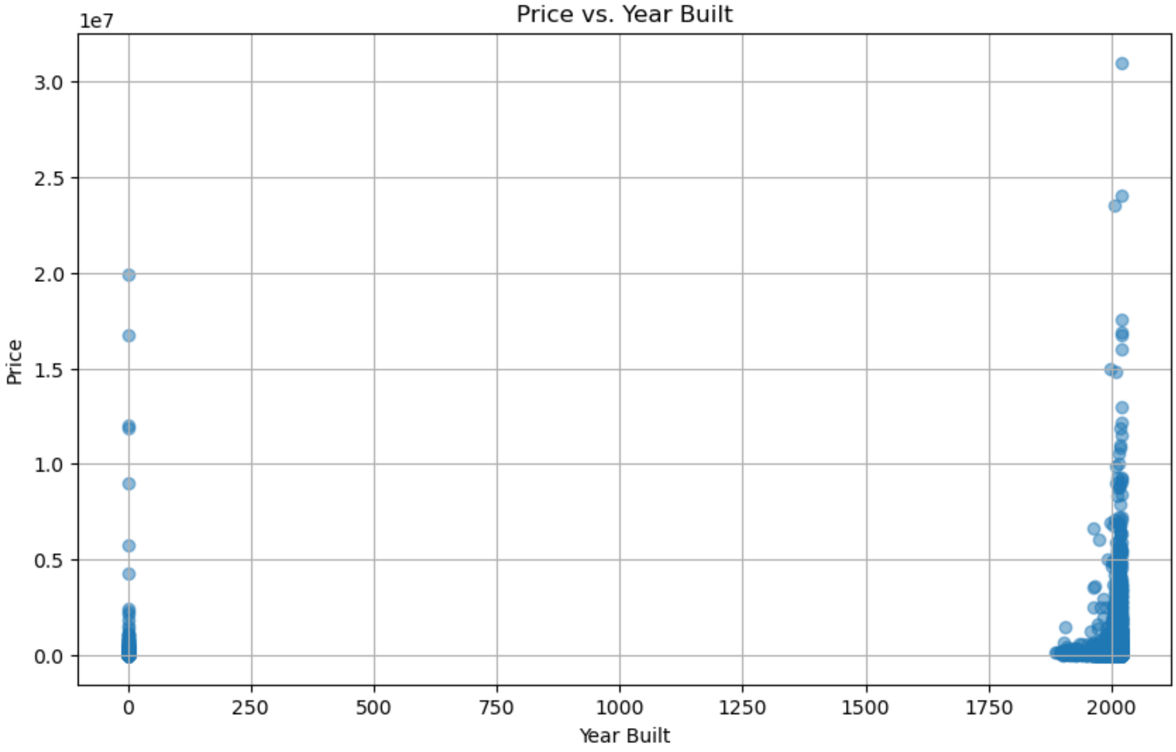
In [27]:

```
# Calculate the percentage of boats made of each material
material_percentage = df['Material'].value_counts(normalize=True) * 100
print("Percentage of Boats by Material:")
print(material_percentage)
```

```
Percentage of Boats by Material:
GRP                    67.379285
PVC                    13.797764
Steel                  11.537044
Wood                    2.887333
Aluminium               2.813613
Plastic                 0.946062
Carbon Fiber            0.368596
Thermoplastic           0.184298
Hypalon                 0.061433
Reinforced concrete     0.012287
Rubber                  0.012287
Name: Material, dtype: float64
```

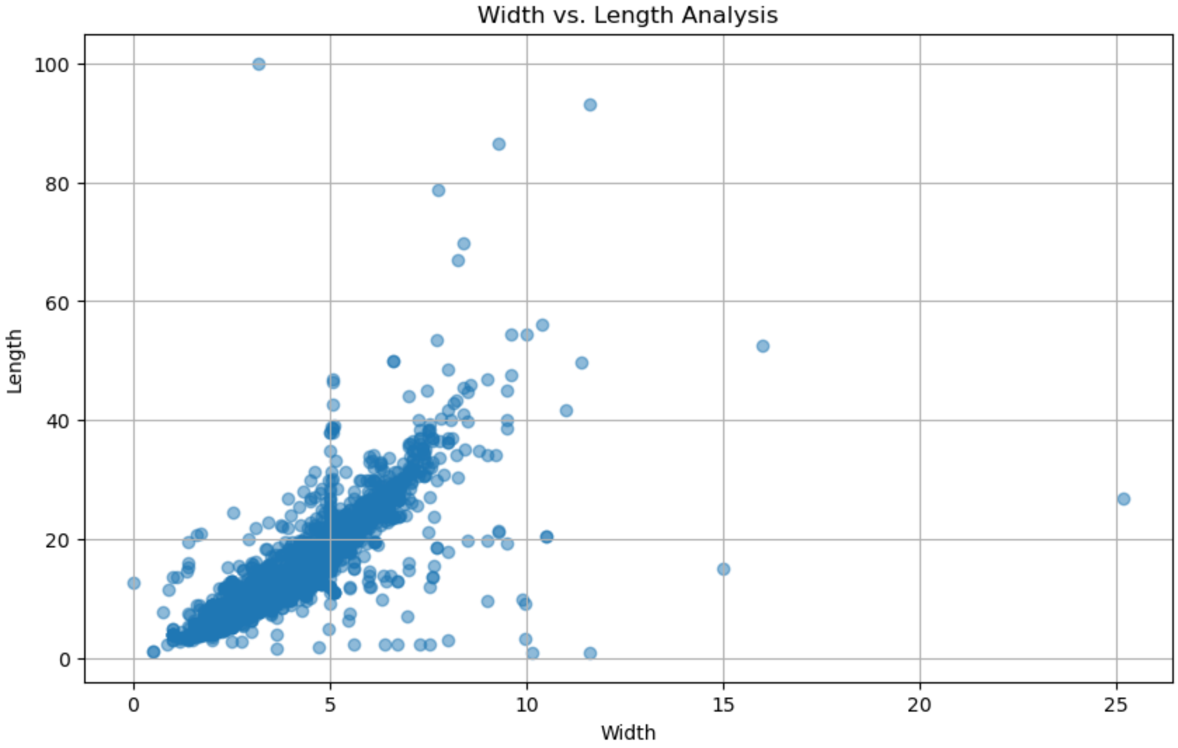
In [28]:

```
# Create a scatter plot to analyze the relationship between boat prices and the yea
plt.figure(figsize=(10, 6))
plt.scatter(df['Year Built'], df['Price'], alpha=0.5)
plt.xlabel('Year Built')
plt.ylabel('Price')
plt.title('Price vs. Year Built')
plt.grid(True)
plt.show()
```



In [29]:

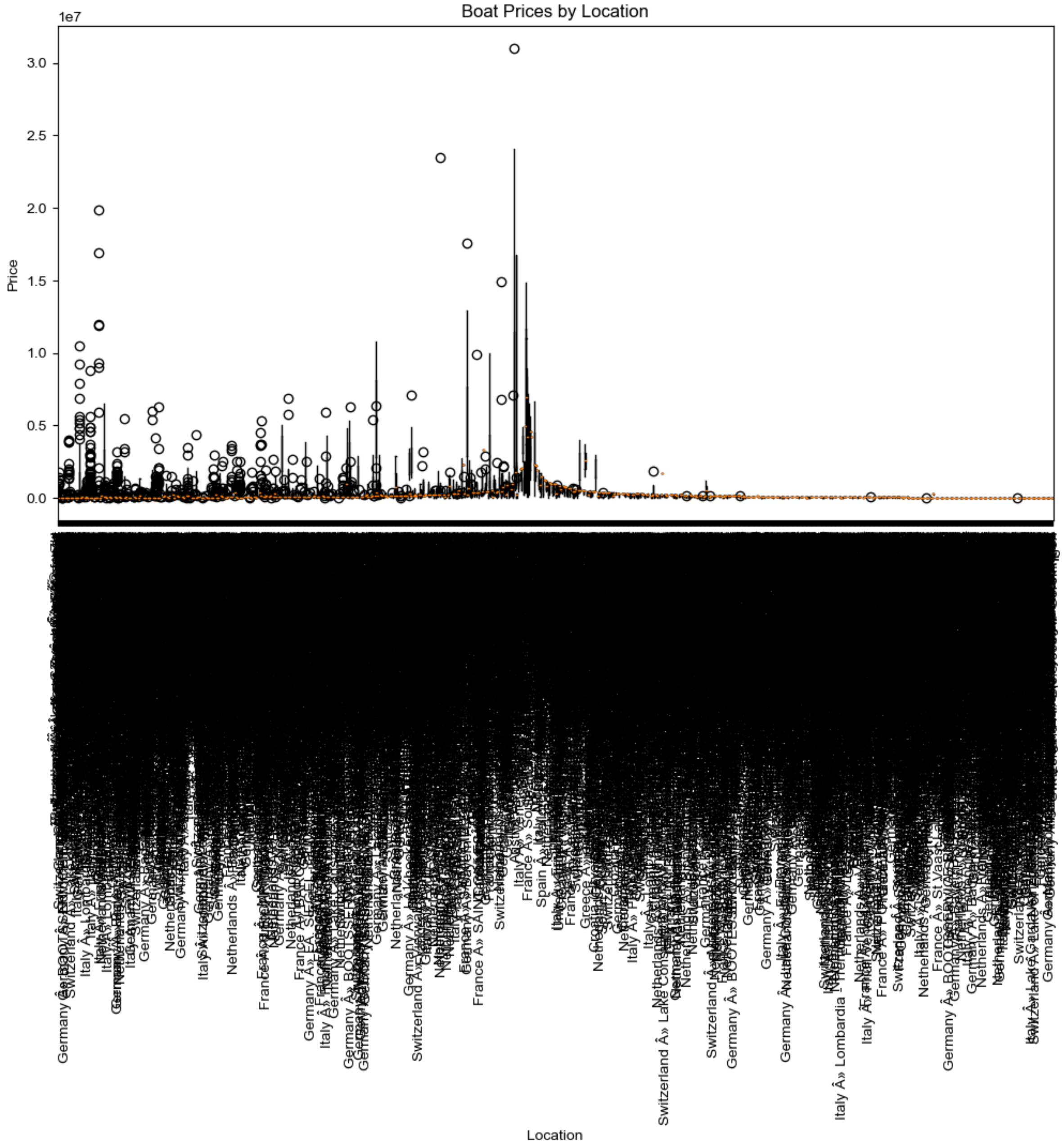
```python
# Create a scatter plot to analyze the relationship between boat width and length
plt.figure(figsize=(10, 6))
plt.scatter(df['Width'], df['Length'], alpha=0.5)
plt.xlabel('Width')
plt.ylabel('Length')
plt.title('Width vs. Length Analysis')
plt.grid(True)
plt.show()
```



In [31]:

```python
# Specify a font that includes the necessary glyphs (replace 'Arial' with your pref
plt.rcParams['font.family'] = 'Arial'
# Create a box plot to visualize boat prices by location
plt.figure(figsize=(12, 6))
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
plt.title('Boat Prices by Location')
plt.xlabel('Location')
plt.ylabel('Price')
plt.boxplot([df[df['Location'] == location]['Price'] for location in df['Location']
plt.show()
```

```
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 156 (\x9c) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 150 (\x96) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 159 (\x9f) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 9 (       ) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 133 (\x85) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 137 (\x89) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 132 (\x84) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 134 (\x86) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 128 (\x80) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 153 (\x99) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 152 (\x98) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 142 (\x8e) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



In [32]:

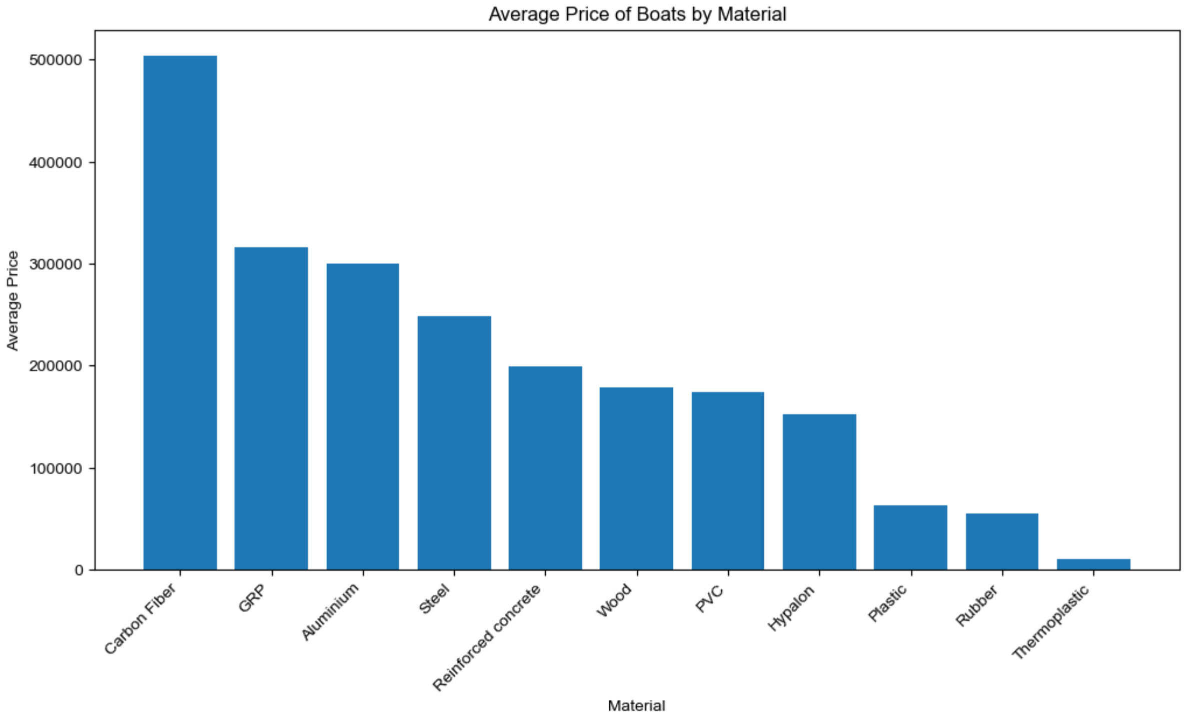
```
# Manufacturer Analysis: Identify the top manufacturers with the highest number of
top_manufacturers = df['Manufacturer'].value_counts().head(10)
print("Top Manufacturers with Most Listings:")
print(top_manufacturers)
```

```
Top Manufacturers with Most Listings:
BÃ©nÃ©teau power boats                            631
Jeanneau power boats                              537
Sunseeker power boats                             383
Princess power boats                              241
Sea Ray power boats                               239
Cranchi power boats                               219
Azimut power boats                                215
Bavaria power boats                               185
Fairline power boats                              172
Quicksilver (Brunswick Marine) power boats        167
Name: Manufacturer, dtype: int64
```

In [33]:

```
# Group the data by material and calculate the average price for each material
material_prices = df.groupby('Material')['Price'].mean().reset_index()

# Sort the data by average price in descending order
material_prices = material_prices.sort_values(by='Price', ascending=False)
```

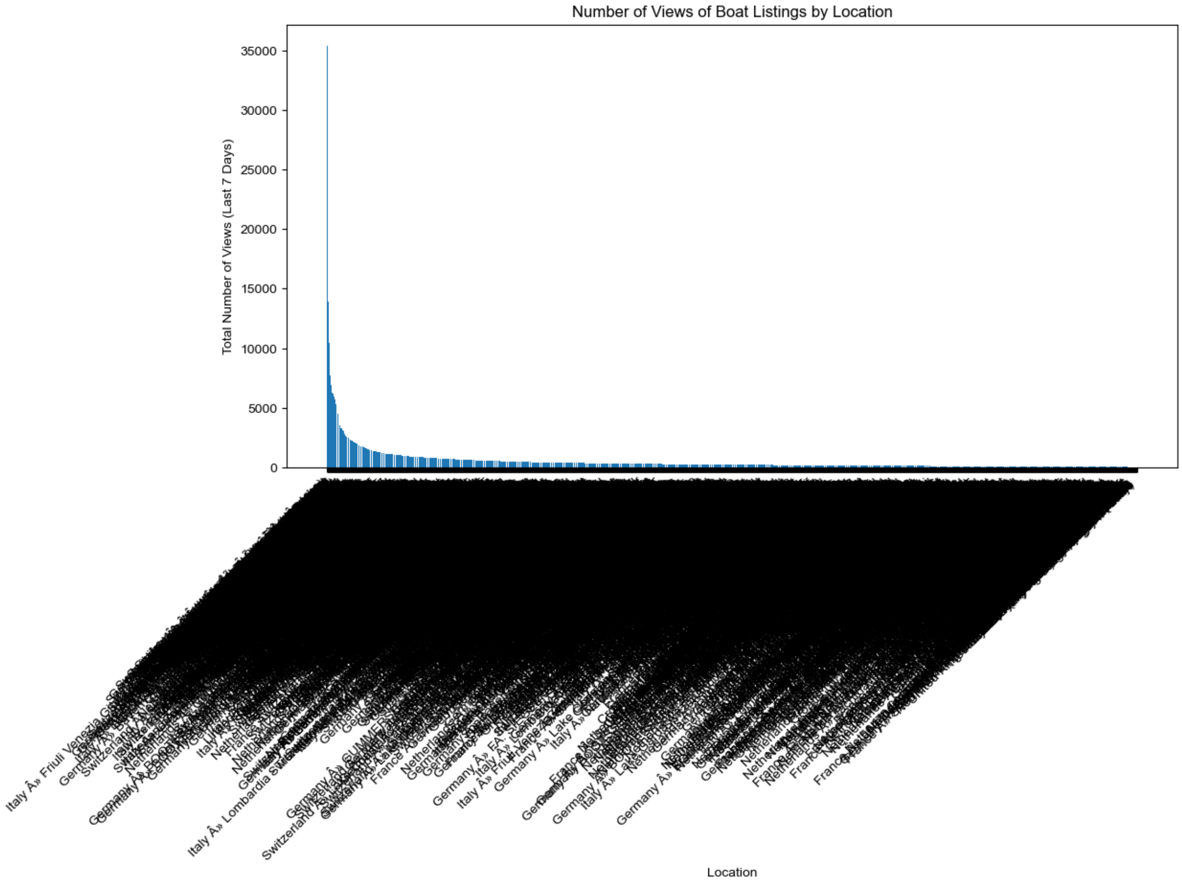
```
# Create a bar plot to visualize the average prices by material
plt.figure(figsize=(12, 6))
plt.bar(material_prices['Material'], material_prices['Price'])
plt.xlabel('Material')
plt.ylabel('Average Price')
plt.title('Average Price of Boats by Material')
plt.xticks(rotation=45, ha='right')
plt.show()
```



In [35]:

```
# Group the data by location and calculate the total number of views for each locat
location_views = df.groupby('Location')['Number of views last 7 days'].sum().reset_

# Sort the data by total views in descending order
location_views = location_views.sort_values(by='Number of views last 7 days', ascen

# Create a bar plot to visualize the number of views by location
plt.figure(figsize=(12, 6))
plt.bar(location_views['Location'], location_views['Number of views last 7 days'])
plt.xlabel('Location')
plt.ylabel('Total Number of Views (Last 7 Days)')
plt.title('Number of Views of Boat Listings by Location')
plt.xticks(rotation=45, ha='right')
plt.show()
```

```
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 156 (\x9c) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 150 (\x96) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 152 (\x98) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 159 (\x9f) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 142 (\x8e) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 128 (\x80) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 153 (\x99) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 137 (\x89) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 132 (\x84) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 133 (\x85) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 134 (\x86) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\hrobi\Downloads\ANACONDA\lib\site-packages\IPython\core\pylabtools.py:15
2: UserWarning: Glyph 9 (	) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



In [ ]: