Sales and Product Demand Analysis

In [1]:
```
# Leveraging Python for Insightful Sales and Product Analysis
# Exploratory Analysis
# Statistical Analysis
# predictive Analysis
```

In [3]:
```
# Importing Librarys

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import OneHotEncoder
import warnings
warnings.filterwarnings("ignore")
```

Load the data from CSV files and merge files

In [7]:
```
# Load the dataset
product = pd.read_csv("C:/Users/HP/Desktop/NEW PORTFOLIO  WORK/Toys 4 tool project data
```

In [8]:
```
# Load the dataset
sales = pd.read_csv("C:/Users/HP/Desktop/NEW PORTFOLIO  WORK/Toys 4 tool project data se
```

In [9]:
```
# Load the dataset
stores = pd.read_csv("C:/Users/HP/Desktop/NEW PORTFOLIO  WORK/Toys 4 tool project data s
```

In [15]:
```
# Merge the three tables on both 'product_id' and 'store_id' columns
merged_data = pd.merge(pd.merge(sales, product, on='Product_ID', how='inner'), stores, o
```

Exploratory data analysis of data

In [ ]:
```
# Perform basic exploratory analysis to understand the data
```

In [17]:
```
print(merged_data.head())
```

```
   Sale_ID        Date  Store_ID  Product_ID  Units      Product_Name  \
0        1  2022-01-01        24           4      1  Chutes & Ladders
1       66  2022-01-01        24           4      1  Chutes & Ladders
2       98  2022-01-01        24           4      1  Chutes & Ladders
3      128  2022-01-01        24           4      1  Chutes & Ladders
4      146  2022-01-01        24           4      1  Chutes & Ladders

  Product_Category Product_Cost Product_Price                    Store_Name  \
0            Games       $9.99        $12.99   Maven Toys Aguascalientes 1
1            Games       $9.99        $12.99   Maven Toys Aguascalientes 1
2            Games       $9.99        $12.99   Maven Toys Aguascalientes 1
3            Games       $9.99        $12.99   Maven Toys Aguascalientes 1
4            Games       $9.99        $12.99   Maven Toys Aguascalientes 1

       Store_City Store_Location Store_Open_Date
0  Aguascalientes       Downtown      2010-07-31
```

```
1  Aguascalientes       Downtown      2010-07-31
2  Aguascalientes       Downtown      2010-07-31
3  Aguascalientes       Downtown      2010-07-31
4  Aguascalientes       Downtown      2010-07-31
```

In [18]:
```
merged_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 829262 entries, 0 to 829261
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   Sale_ID           829262 non-null  int64 
 1   Date              829262 non-null  object
 2   Store_ID          829262 non-null  int64 
 3   Product_ID        829262 non-null  int64 
 4   Units             829262 non-null  int64 
 5   Product_Name      829262 non-null  object
 6   Product_Category  829262 non-null  object
 7   Product_Cost      829262 non-null  object
 8   Product_Price     829262 non-null  object
 9   Store_Name        829262 non-null  object
 10  Store_City        829262 non-null  object
 11  Store_Location    829262 non-null  object
 12  Store_Open_Date   829262 non-null  object
dtypes: int64(4), object(9)
memory usage: 88.6+ MB
```

In [25]:
```
# Remove '$' symbol and convert 'Product_Cost' and 'Product_Price' columns to float
merged_data['Product_Cost'] = merged_data['Product_Cost'].str.replace('$', '').astype(fl
merged_data['Product_Price'] = merged_data['Product_Price'].str.replace('$', '').astype(
```

In [27]:
```
merged_data.shape
```

Out[27]:
```
(829262, 13)
```

In [29]:
```
#Extracting month, week, day name and year from the date column
merged_data['month']=pd.to_datetime(merged_data['Store_Open_Date']).dt.month
merged_data['day_name']=pd.to_datetime(merged_data['Store_Open_Date']).dt.day_name()
merged_data['year']=pd.to_datetime(merged_data['Store_Open_Date']).dt.year
```

In [30]:
```
merged_data.head(10)
```

Out[30]:

	Sale_ID	Date	Store_ID	Product_ID	Units	Product_Name	Product_Category	Product_Cost	Product_Price	
0	1	2022-01-01	24	4	1	Chutes & Ladders	Games	9.99	12.99	Ag
1	66	2022-01-01	24	4	1	Chutes & Ladders	Games	9.99	12.99	Ag
2	98	2022-01-01	24	4	1	Chutes & Ladders	Games	9.99	12.99	Ag
3	128	2022-01-01	24	4	1	Chutes & Ladders	Games	9.99	12.99	Ag
4	146	2022-01-01	24	4	1	Chutes & Ladders	Games	9.99	12.99	Ag
5	258	2022-	24	4	1	Chutes &	Games	9.99	12.99	

		01-01				Ladders				Ag
6	271	2022-01-01	24	4	1	Chutes & Ladders	Games	9.99	12.99	Ag
7	302	2022-01-01	24	4	1	Chutes & Ladders	Games	9.99	12.99	Ag
8	394	2022-01-01	24	4	1	Chutes & Ladders	Games	9.99	12.99	Ag
9	703	2022-01-01	24	4	1	Chutes & Ladders	Games	9.99	12.99	Ag

In [31]:
```
merged_data.tail(10)
```

Out[31]:

	Sale_ID	Date	Store_ID	Product_ID	Units	Product_Name	Product_Category	Product_Cost	Product_Pri
829252	792603	2023-09-04	13	28	1	Playfoam	Art & Crafts	3.99	10.
829253	792827	2023-09-04	13	28	1	Playfoam	Art & Crafts	3.99	10.
829254	792861	2023-09-04	13	28	1	Playfoam	Art & Crafts	3.99	10.
829255	792908	2023-09-04	13	28	1	Playfoam	Art & Crafts	3.99	10.
829256	793137	2023-09-04	13	28	1	Playfoam	Art & Crafts	3.99	10.
829257	793189	2023-09-04	13	28	1	Playfoam	Art & Crafts	3.99	10.
829258	820255	2023-09-25	13	28	1	Playfoam	Art & Crafts	3.99	10.
829259	820290	2023-09-25	13	28	1	Playfoam	Art & Crafts	3.99	10.
829260	820877	2023-09-25	13	28	1	Playfoam	Art & Crafts	3.99	10.
829261	821027	2023-09-25	13	28	1	Playfoam	Art & Crafts	3.99	10.

Statistical Analysis

In [26]:
```
print(merged_data.describe())
```

```
            Sale_ID       Store_ID     Product_ID          Units  \
count  829262.000000  829262.000000  829262.000000  829262.000000
mean   414631.500000      25.277034      15.014149       1.315103
std    239387.463803      14.352573       9.869417       0.830701
min         1.000000       1.000000       1.000000       1.000000
25%    207316.250000      13.000000       6.000000       1.000000
50%    414631.500000      26.000000      14.000000       1.000000
75%    621946.750000      38.000000      24.000000       1.000000
```

```
max    829262.000000      50.000000      35.000000      30.000000
```

```
        Product_Cost  Product_Price
count  829262.000000  829262.000000
mean        9.976460      13.772327
std         7.817749       8.664794
min         1.990000       2.990000
25%         3.990000       6.990000
50%         7.990000      12.990000
75%        11.990000      15.990000
max        34.990000      39.990000
```

In [43]:
```
correlation_matrix = merged_data[['Units', 'Product_Cost', 'Product_Price']].corr()
print(correlation_matrix)
```

```
                Units  Product_Cost  Product_Price
Units        1.000000     -0.083454      -0.096340
Product_Cost -0.083454      1.000000       0.961101
Product_Price -0.096340     0.961101       1.000000
```

Predictive Analysis of Store Sales

In [56]:
```
# Group data by 'Store_Name' and 'year' and calculate the count of sales
sales_by_store_year = merged_data.groupby(['Store_Name', 'year']).size().reset_index(nam

# Sort the result by 'year' in ascending order
sales_by_store_year = sales_by_store_year.sort_values(by='year', ascending=True)

# Compute the difference between each year and 2016
sales_by_store_year['Years of sales'] = abs(sales_by_store_year['year'] - 2016)

# Find the last year in which a product was sold for each store
last_year_sold = merged_data.groupby('Store_Name')['year'].max().reset_index()
last_year_sold.columns = ['Store_Name', 'Last_Sold_Year']

# Merge the last year sold information with the sales_by_store_year DataFrame
sales_by_store_year = pd.merge(sales_by_store_year, last_year_sold, on='Store_Name', how

print(sales_by_store_year)
```

```
                      Store_Name  year  Sales_Count  Years of sales  \
0        Maven Toys Guadalajara 1  1992        15926              24
1          Maven Toys Monterrey 1  1995        15698              21
2        Maven Toys Guadalajara 2  1999        16331              17
3           Maven Toys Saltillo 1  2000        18924              16
4             Maven Toys La Paz 1  2001        13217              15
5           Maven Toys Mexicali 1  2003        16864              13
6          Maven Toys Monterrey 2  2003        21300              13
7   Maven Toys Ciudad de Mexico 1  2004        24482              12
8            Maven Toys Pachuca 1  2004        14969              12
9         Maven Toys Cuernavaca 1  2005        13643              11
10          Maven Toys Campeche 1  2005        17695              11
11          Maven Toys Chetumal 1  2006        14644              10
12          Maven Toys Mexicali 2  2006        16991              10
13            Maven Toys Toluca 1  2007        23533               9
14  Maven Toys Tuxtla Gutierrez 1  2007        14618               9
15        Maven Toys Guanajuato 1  2007        18157               9
16   Maven Toys San Luis Potosi 1  2007        15499               9
17            Maven Toys Puebla 1  2008        15776               8
18            Maven Toys Merida 1  2008        14875               8
19          Maven Toys Santiago 1  2009        16111               7
20         Maven Toys Zacatecas 1  2009        13501               7
21            Maven Toys Oaxaca 1  2010        13957               6
22    Maven Toys Aguascalientes 1  2010        14588               6
```

23	Maven Toys Guanajuato	2	2010	16494	6
24	Maven Toys Campeche	2	2010	12805	6
25	Maven Toys Chihuahua	1	2010	13998	6
26	Maven Toys Ciudad Victoria	1	2010	16034	6
27	Maven Toys Xalapa	1	2011	14998	5
28	Maven Toys Guadalajara	3	2011	23384	5
29	Maven Toys Puebla	2	2011	16764	5
30	Maven Toys Ciudad de Mexico	2	2012	29024	4
31	Maven Toys Hermosillo	1	2012	15264	4
32	Maven Toys Villahermosa	1	2013	16324	3
33	Maven Toys Monterrey	3	2013	16049	3
34	Maven Toys Morelia	1	2013	14956	3
35	Maven Toys Chilpancingo	1	2013	14592	3
36	Maven Toys Ciudad de Mexico	3	2013	19551	3
37	Maven Toys Toluca	2	2014	12776	2
38	Maven Toys Chihuahua	2	2014	16580	2
39	Maven Toys Hermosillo	2	2014	18018	2
40	Maven Toys Durango	1	2014	14110	2
41	Maven Toys Xalapa	2	2014	18809	2
42	Maven Toys Puebla	3	2014	14868	2
43	Maven Toys Hermosillo	3	2014	16553	2
44	Maven Toys Ciudad de Mexico	4	2015	17668	1
45	Maven Toys Monterrey	4	2015	16276	1
46	Maven Toys Guadalajara	4	2015	18739	1
47	Maven Toys Saltillo	2	2016	14166	0
48	Maven Toys Culiacan	1	2016	14594	0
49	Maven Toys Guanajuato	3	2016	14569	0

	Last_Sold_Year
0	1992
1	1995
2	1999
3	2000
4	2001
5	2003
6	2003
7	2004
8	2004
9	2005
10	2005
11	2006
12	2006
13	2007
14	2007
15	2007
16	2007
17	2008
18	2008
19	2009
20	2009
21	2010
22	2010
23	2010
24	2010
25	2010
26	2010
27	2011
28	2011
29	2011
30	2012
31	2012
32	2013
33	2013
34	2013
35	2013
36	2013

```
37            2014
38            2014
39            2014
40            2014
41            2014
42            2014
43            2014
44            2015
45            2015
46            2015
47            2016
48            2016
49            2016
```

In [58]:
```python
# Select specific columns to display
columns_to_display = ['Store_Name', 'year','Sales_Count', 'Last_Sold_Year']

# Print the selected columns
print(sales_by_store_year[columns_to_display])
```

```
                      Store_Name  year  Sales_Count  Last_Sold_Year
0        Maven Toys Guadalajara 1  1992        15926            1992
1          Maven Toys Monterrey 1  1995        15698            1995
2        Maven Toys Guadalajara 2  1999        16331            1999
3           Maven Toys Saltillo 1  2000        18924            2000
4             Maven Toys La Paz 1  2001        13217            2001
5           Maven Toys Mexicali 1  2003        16864            2003
6          Maven Toys Monterrey 2  2003        21300            2003
7   Maven Toys Ciudad de Mexico 1  2004        24482            2004
8            Maven Toys Pachuca 1  2004        14969            2004
9         Maven Toys Cuernavaca 1  2005        13643            2005
10          Maven Toys Campeche 1  2005        17695            2005
11          Maven Toys Chetumal 1  2006        14644            2006
12          Maven Toys Mexicali 2  2006        16991            2006
13            Maven Toys Toluca 1  2007        23533            2007
14  Maven Toys Tuxtla Gutierrez 1  2007        14618            2007
15        Maven Toys Guanajuato 1  2007        18157            2007
16   Maven Toys San Luis Potosi 1  2007        15499            2007
17            Maven Toys Puebla 1  2008        15776            2008
18            Maven Toys Merida 1  2008        14875            2008
19          Maven Toys Santiago 1  2009        16111            2009
20         Maven Toys Zacatecas 1  2009        13501            2009
21            Maven Toys Oaxaca 1  2010        13957            2010
22    Maven Toys Aguascalientes 1  2010        14588            2010
23        Maven Toys Guanajuato 2  2010        16494            2010
24          Maven Toys Campeche 2  2010        12805            2010
25         Maven Toys Chihuahua 1  2010        13998            2010
26   Maven Toys Ciudad Victoria 1  2010        16034            2010
27            Maven Toys Xalapa 1  2011        14998            2011
28       Maven Toys Guadalajara 3  2011        23384            2011
29            Maven Toys Puebla 2  2011        16764            2011
30  Maven Toys Ciudad de Mexico 2  2012        29024            2012
31        Maven Toys Hermosillo 1  2012        15264            2012
32      Maven Toys Villahermosa 1  2013        16324            2013
33         Maven Toys Monterrey 3  2013        16049            2013
34           Maven Toys Morelia 1  2013        14956            2013
35      Maven Toys Chilpancingo 1  2013        14592            2013
36  Maven Toys Ciudad de Mexico 3  2013        19551            2013
37            Maven Toys Toluca 2  2014        12776            2014
38         Maven Toys Chihuahua 2  2014        16580            2014
39        Maven Toys Hermosillo 2  2014        18018            2014
40           Maven Toys Durango 1  2014        14110            2014
41            Maven Toys Xalapa 2  2014        18809            2014
42            Maven Toys Puebla 3  2014        14868            2014
43        Maven Toys Hermosillo 3  2014        16553            2014
44  Maven Toys Ciudad de Mexico 4  2015        17668            2015
```

```
45          Maven Toys Monterrey 4  2015        16276           2015
46       Maven Toys Guadalajara 4   2015        18739           2015
47          Maven Toys Saltillo 2   2016        14166           2016
48          Maven Toys Culiacan 1   2016        14594           2016
49        Maven Toys Guanajuato 3   2016        14569           2016
```

Maven Toys Ciudad de Mexico 2 sold porduct 29024
Maven Toys Guadalajara 3 2011 sold porduct 23384
Maven Toys Ciudad de Mexico 3 sold porduct 19551
Maven Toys Hermosillo 2 sold porduct 18018
Maven Toys Xalapa 2 sold porduct 18809
Maven Toys Ciudad de Mexico 4 sold porduct 17668

Over the span of six years from 2010 to 2016, these five companies excelled in generating significant sales volumes. Establishing strong relationships with these stores during this period proved instrumental in driving successful sales outcomes. However, it's important to note that these sales were achieved within a one-year timeframe for each store.

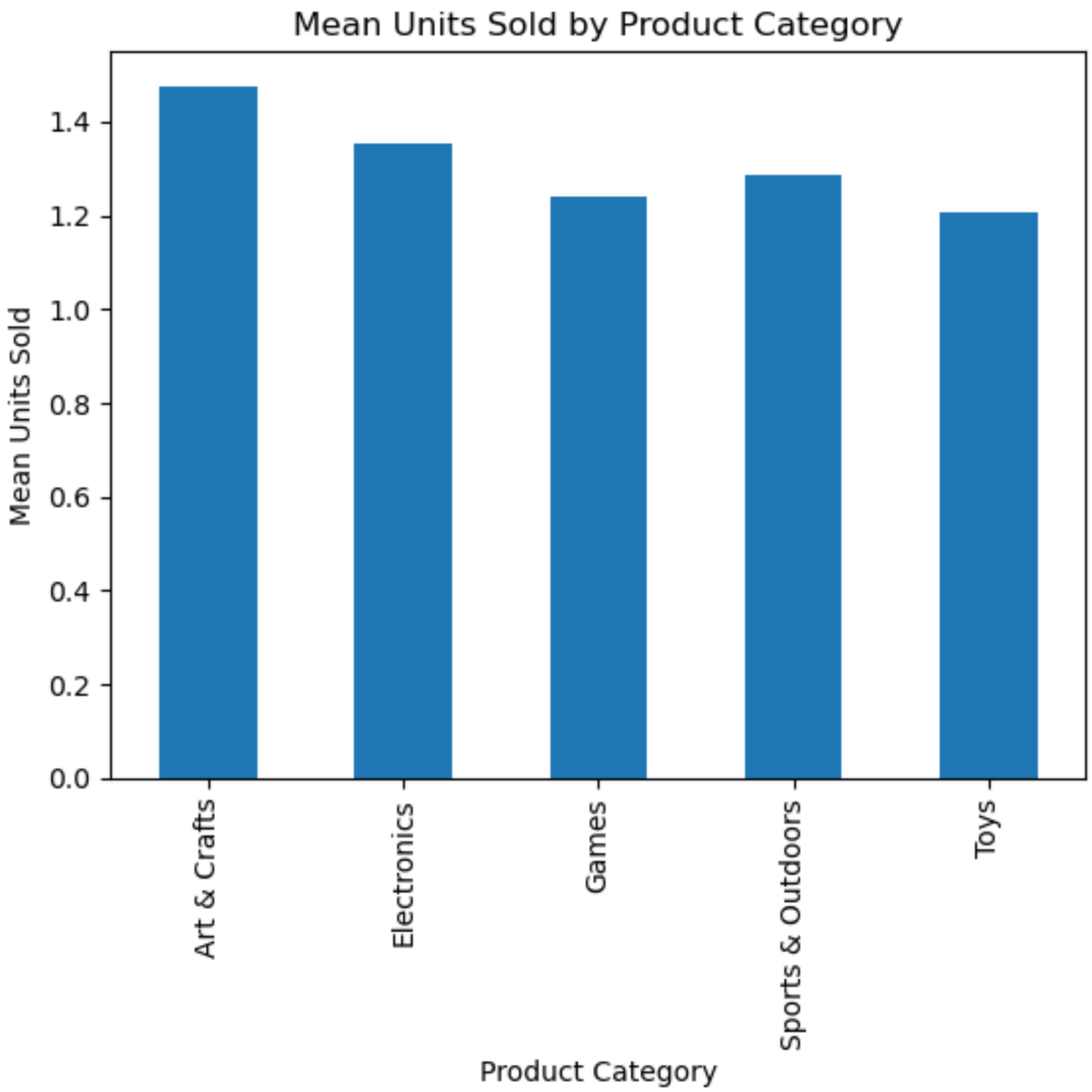
Product Demand Prediction Analyses

In [59]:
```
# Step 2: Statistical Analysis
descriptive_stats = merged_data.groupby(['Product_Name', 'Product_Category']).agg({'Unit
correlation_matrix = merged_data.corr()
```

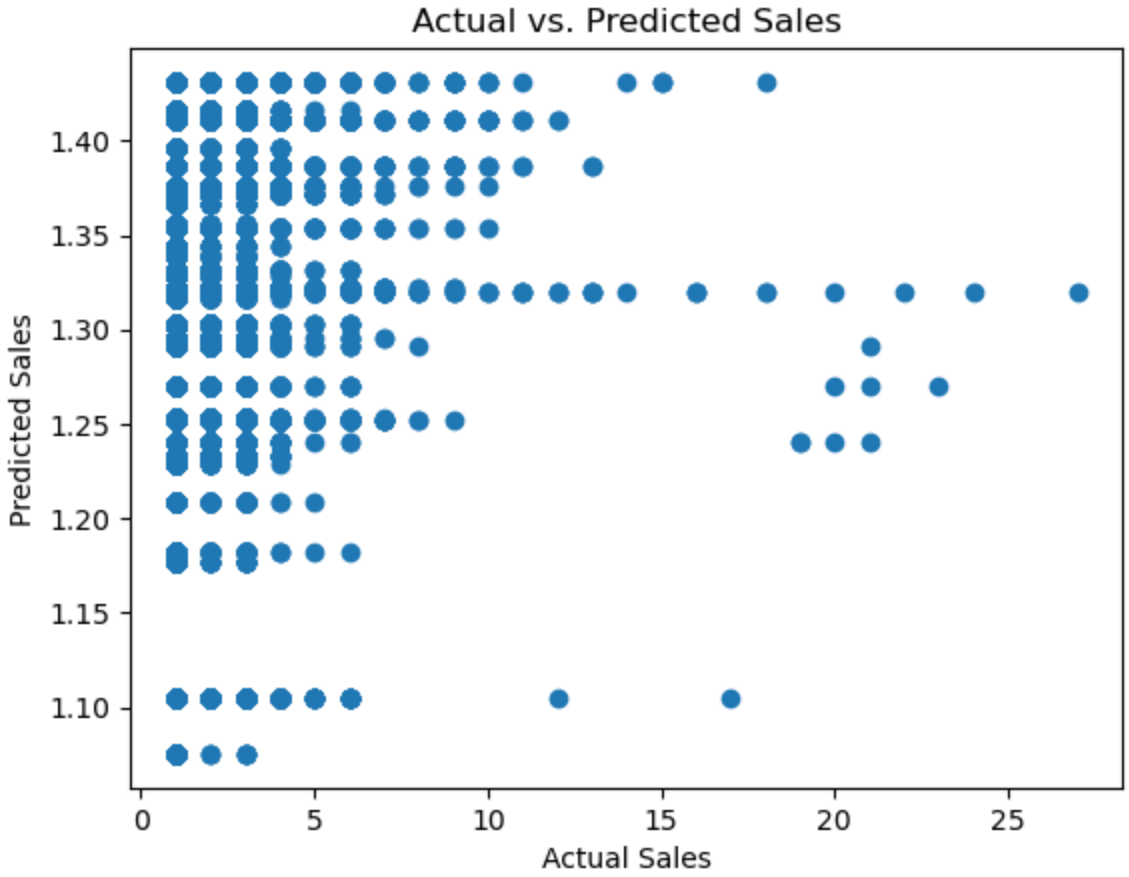
In [61]:
```
# Step 3: Sales Prediction
X = merged_data[['Product_Cost', 'Product_Price']]
y = merged_data['Units']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42

model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
```

In [62]:
```
# Step 4: Visualization
# Example: Bar plot of mean units sold by product category
mean_units_by_category = merged_data.groupby('Product_Category')['Units'].mean()
mean_units_by_category.plot(kind='bar', xlabel='Product Category', ylabel='Mean Units So
plt.show()
```



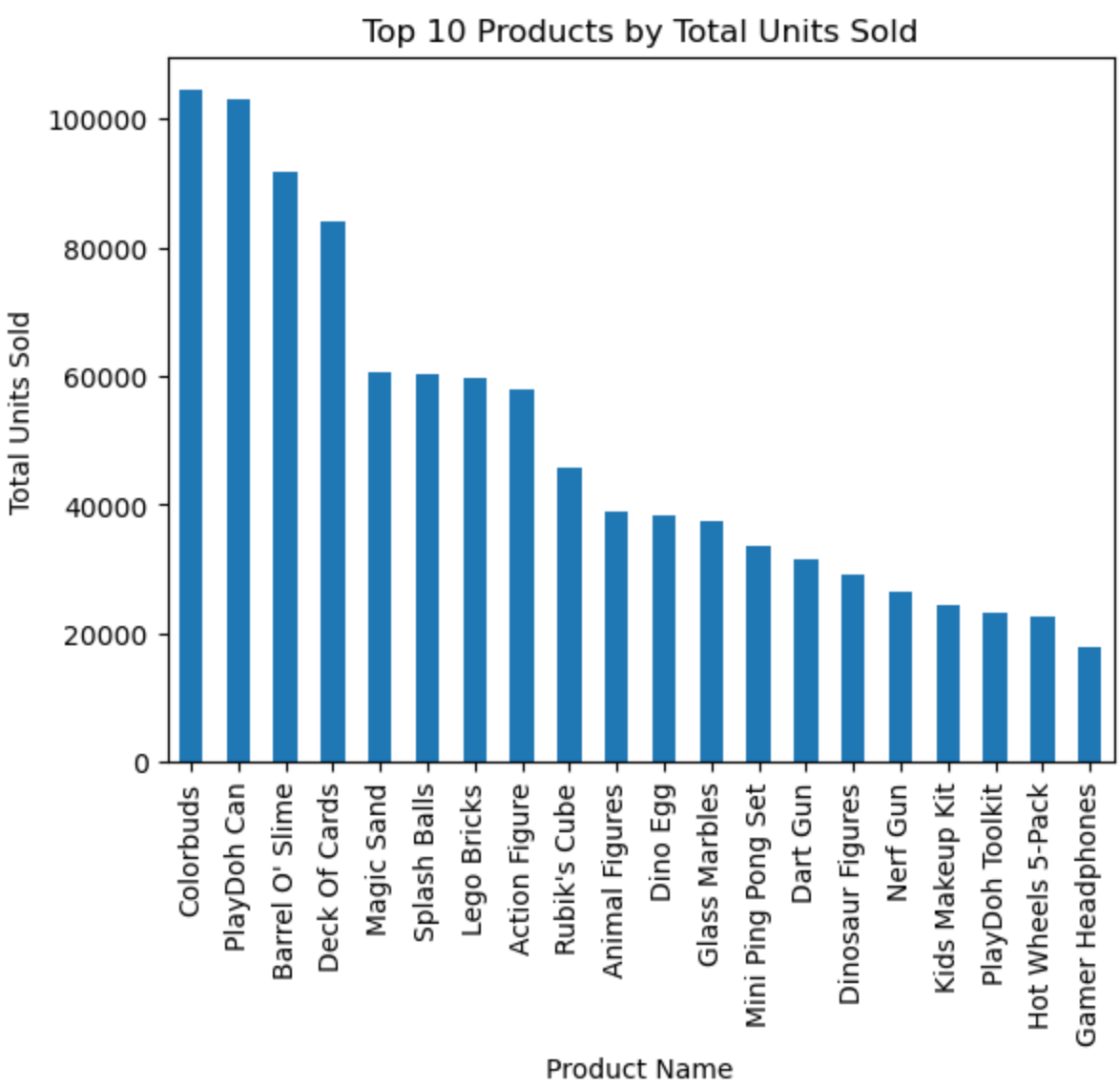
In [63]:
```
# Example: Scatter plot of actual vs. predicted sales
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Sales')
plt.ylabel('Predicted Sales')
plt.title('Actual vs. Predicted Sales')
plt.show()
```



In [64]:
```
# Step 2: Aggregate data to calculate demand metrics
product_demand = merged_data.groupby('Product_Name').agg({
    'Units': 'sum',  # Total units sold
    'Product_Price': 'mean'  # Average price
})
product_demand = product_demand.rename(columns={'Units': 'Total_Units_Sold', 'Product_Pr
```

In [66]:
```
 #Step 3: Sort products by total units sold or any other metric of interest
product_demand = product_demand.sort_values(by='Total_Units_Sold', ascending=False)

# Step 4: Visualize demand
product_demand.head(20).plot(kind='bar', y='Total_Units_Sold', legend=None)
plt.xlabel('Product Name')
plt.ylabel('Total Units Sold')
plt.title('Top 10 Products by Total Units Sold')
plt.show()
```



Findings from the data analysis reveal that across five categories - art and craft, sports and outdoor toys, games, toys, and electronics - there exists slight variation in demand. However, all categories demonstrate consistent demand, indicating a diverse market landscape.

Moreover, products labeled as colourbuds,playdoh can,barrel o' slime and deck of card emerge as the top-selling items, suggesting their popularity among consumers. This data underscores the importance of these products in driving sales within the market.