In [ ]:

Analysis Report: Analysing e-commerce store sales

Author: PUNEETH G S
Date: 11 /12 /2024

In [ ]:

In [ ]:

Analysis of E - commerce Data set

source : kaggle

In [ ]:

this note explore the factor influencing the diffrent e - commerce category , discount and payment method relationships

containing attribute such as category ,price , discounts , final price and purchase date

In [ ]:

objective

- identifying key factors like category and discounts
- analysing diffrent purchase date
- check any seasonal relationship between the category product
- providing insights to manegers or shoopkeepers to stockup the product based analysis

In [ ]:

Dataset Description

- User_ID: A unique identifier for each user (e.g., a shortened version of a UUID)

- Product_ID: A unique identifier for each product (e.g., a shortened version of a UUID)
- Category: The product category (e.g., Electronics, Clothing, Sports, etc.)
- Price: The original price of the product before any discount is applied
- Discount (%): The discount percentage applied to the product
- Final_Price: The final price of the product after applying the discount
- Payment_Method: The method used for payment (e.g., Credit Card, UPI, Net Banking)
- Purchase_Date: The date when the transaction occurred, formatted as MM-DD-YYYY

In [ ]:

Methodology

1. Load and explore the dataset.
2. Perform data cleaning and preprocessing.
3. Analyze and visualize key metrics.
4. Draw conclusions and provide recommendations.

In [ ]:

In [10]:
```
# Let's Begin
```

Load the dataset

In [12]:
```
# import the necessary liabraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
plt.style.use("ggplot")

data = pd.read_csv("e-commerce.csv")
data.head()
```

Out[12]:

	User_ID	Product_ID	Category	Price (Rs.)	Discount (%)	Final_Price(Rs.)	Payment_Method
0	337c166f	f414122f-e	Sports	36.53	15	31.05	Net Banking
1	d38a19bf	fde50f9c-5	Clothing	232.79	20	186.23	Net Banking
2	d7f5f0b0	0d96fc90-3	Sports	317.02	25	237.76	Credit Card
3	395d4994	964fc44b-d	Toys	173.19	25	129.89	UPI
4	a83c145c	d70e2fc6-e	Beauty	244.80	20	195.84	Net Banking

In [ ]:

Data cleaning

In [14]:
```
#checking for any null value present
data.isnull().sum()
```

Out[14]:
```
User_ID             0
Product_ID          0
Category            0
Price (Rs.)         0
Discount (%)        0
Final_Price(Rs.)    0
Payment_Method      0
Purchase_Date       0
dtype: int64
```

In [15]:
```
#checking  any duplicate values present in the data set
data = data.drop_duplicates()
```

In [ ]:

In [16]:
```
# checking any NaN value present
data.isna().sum()
```

Out[16]:
```
User_ID             0
Product_ID          0
Category            0
Price (Rs.)         0
Discount (%)        0
Final_Price(Rs.)    0
Payment_Method      0
Purchase_Date       0
dtype: int64
```

In [ ]:

take away

there is no null value and NaN (Not a Number) present in the data set It clean and ready to analysize

In [ ]:

In [ ]:

data explorations

In [19]:
```
data.shape
```

Out[19]: (3660, 8)

In [20]:
```
## this data set contain 3360 rows and 8 columns bellow are the summary of attri
```

In [21]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3660 entries, 0 to 3659
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   User_ID           3660 non-null   object 
 1   Product_ID        3660 non-null   object 
 2   Category          3660 non-null   object 
 3   Price (Rs.)       3660 non-null   float64
 4   Discount (%)      3660 non-null   int64  
 5   Final_Price(Rs.)  3660 non-null   float64
 6   Payment_Method    3660 non-null   object 
 7   Purchase_Date     3660 non-null   object 
dtypes: float64(2), int64(1), object(5)
memory usage: 228.9+ KB
```

In [22]:
```
data.describe()
```

Out[22]:

	Price (Rs.)	Discount (%)	Final_Price(Rs.)
count	3660.000000	3660.000000	3660.000000
mean	254.800675	18.825137	206.906579
std	141.682621	14.731338	122.687844
min	10.090000	0.000000	5.890000
25%	134.012500	5.000000	104.512500
50%	253.845000	15.000000	199.185000
75%	377.595000	25.000000	304.117500
max	499.960000	50.000000	496.820000

key insights

- average price = 254.8
- minimum price = 10
- maximum price = 499.9
- minimum discounts = 0
- maximum discounts = 50
- Minimum final price = 5.89
- maximum final price = 496.82

In [ ]:

data Transformation

In [25]:
```
#conavrting to date - time
data['Purchase_Date'] = pd.to_datetime(data['Purchase_Date'], format='%d-%m-%Y')

data['day']= data['Purchase_Date'].dt.day

data['month']  = data['Purchase_Date'].dt.month

data['year'] = data['Purchase_Date'].dt.year

months = [
    "January", "February", "March", "April", "May", "June",
    "July", "August", "September", "October", "November", "December"
]

data['month'] = data['month'].map(lambda x: months[x - 1] if 1 <= x <= 12 else "
```

take aways

- convarted purchased_date into valid date and time
- created column day , month , year based on purchase_date

In [138…
```
data.to_excel("E-commerceUpdated.xlsx" , index=False)
```

In [27]:
```
# remaing the columns
data.rename(columns={"Price (Rs.)":"Price"} , inplace=True)

data.rename(columns={"Discount (%)":"Discount"} , inplace=True)

data.rename(columns={'Final_Price(Rs.)': "Final_Price"} , inplace=True)
```

In [ ]:

In [ ]:

1. What are the most popular purchase categories among users,and what discounts do they receive?

In [29]:
```
# grouping the data

category_discount = data.groupby(['Category'])["Discount"].mean().reset_index()\
                    .sort_values(by="Discount" , ascending=False)
```

In [30]:
```
category_discount['Discount'] = np.round(category_discount['Discount'] , 2)
```

In [31]:
```
#visualizing the discounts
plt.figure(figsize=(10,5))
bar=plt.bar(x=category_discount['Category'] , height=category_discount['Discount
height = category_discount['Discount']
plt.bar_label(bar , height)
plt.title(" Total Discount across diffrent Categories ")
plt.show()
```

key take aways

- Home and Kitchen items provide higher discounts, which significantly influence purchases, with Sports and Electronics following closely behind
- Toys and clothing categories provide less discount

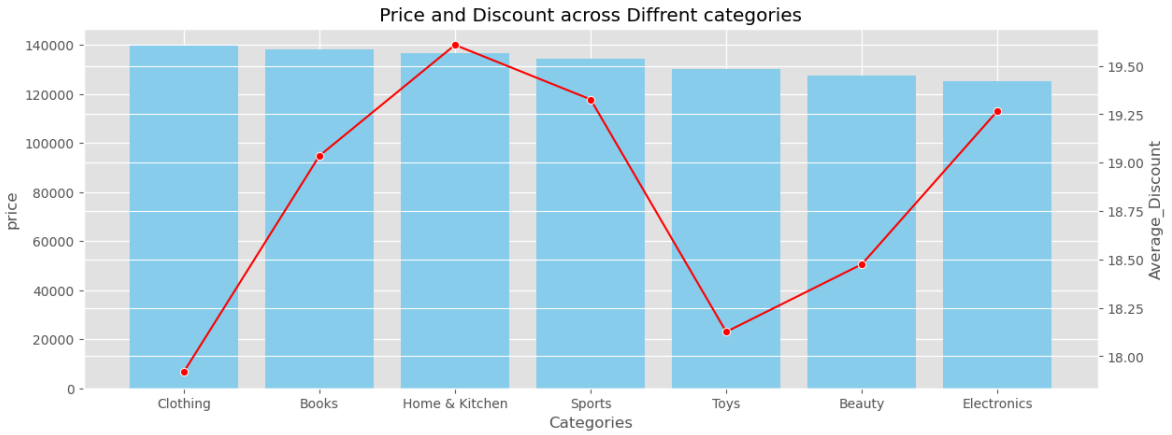
Question 2: Which categories are the most expensive, and what discounts do they offer ?

In [ ]:
```
```

In [34]:
```
#calculating the average discounts with Price

category_summary = data.groupby('Category').agg(
    Total_Price=('Price', 'sum'),
    Average_Discount=('Discount', 'mean')
).reset_index().sort_values(by="Total_Price" , ascending=False)
```

In [ ]:

In [35]:
```
# creating a twin plot

plt.figure(figsize=(14,5))
plt.bar(x=category_summary['Category'] , height=category_summary['Total_Price']
plt.ylabel("price")
plt.xlabel("Categories")

# creating a twin plot
ax = plt.gca().twinx()
sns.lineplot(data= category_summary , ax=ax ,  x="Category" , y="Average_Discoun
plt.title("Price and Discount across Diffrent categories")
plt.show()
```



take aways

- The clothing category has a higher total price and offers lower discounts, followed by the books and home and kitchen categories.
- The sports and electronics categories have lower total prices but provide higher discounts compared to the other categories.

In [ ]:

In [ ]:

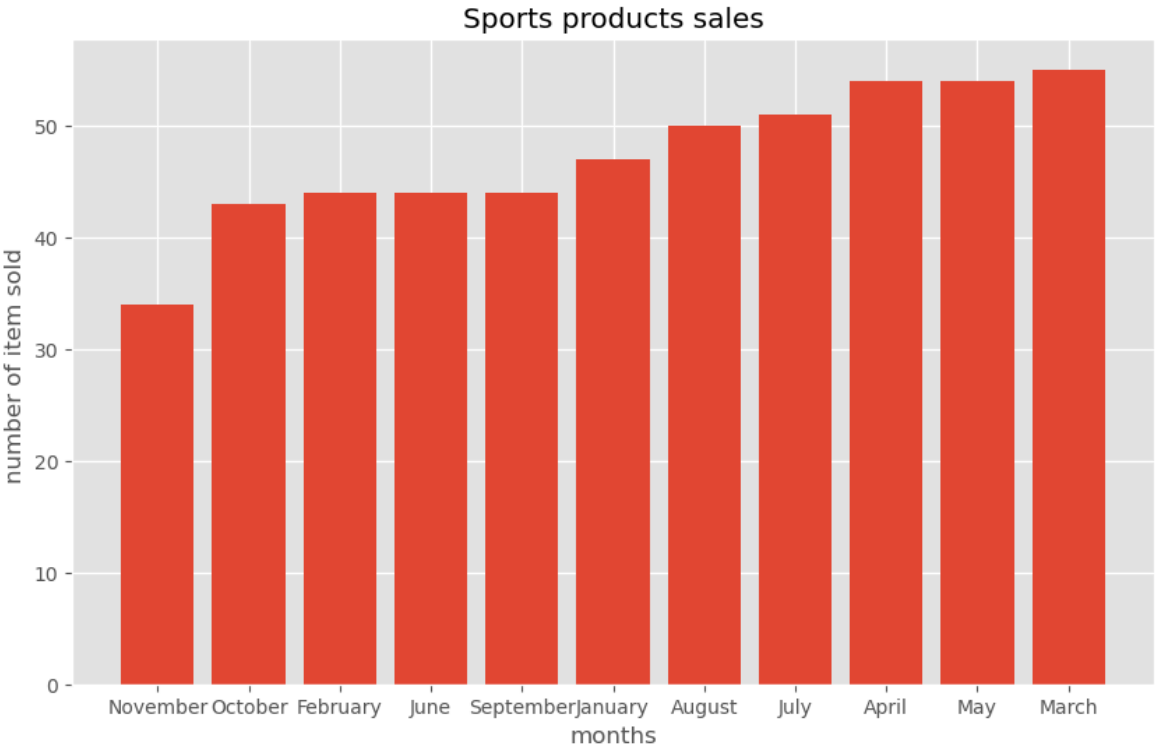
Is there any seasonal relationship between the categories? If so, how is it observed?

In [ ]:

In [38]:
```
# analysis of sports  categories

sports_summary = data.query("`Category`=='Sports'")
sports_summary['Price'].sum()
sports_summary['Discount'].mean()
Total_sports = sports_summary['Final_Price'].sum()
print(f"the total price of overall sports category is {Total_sports} ofter disco
```

```
# checking the trends
sports_summary

sports_insigts = sports_summary.groupby(['month']).agg(Total_price = ("Final_Pri
                                        count = ("month" , "count")).reset_index()
                                       sort_values(by="count" )

#visualize the data

plt.figure(figsize=(10,6))
plt.bar(x=sports_insigts['month'] , height=sports_insigts['count'] )
plt.xlabel("months")
plt.ylabel("number of item sold ")
plt.title("Sports products sales")
plt.show()
```

```
the total price of overall sports category is 108518.79000000001 ofter discount
```



take aways

- Sports products had the highest sales in March, followed by April, May, and July.
-

Sales of sports products were lowest in November, October, February, and September.

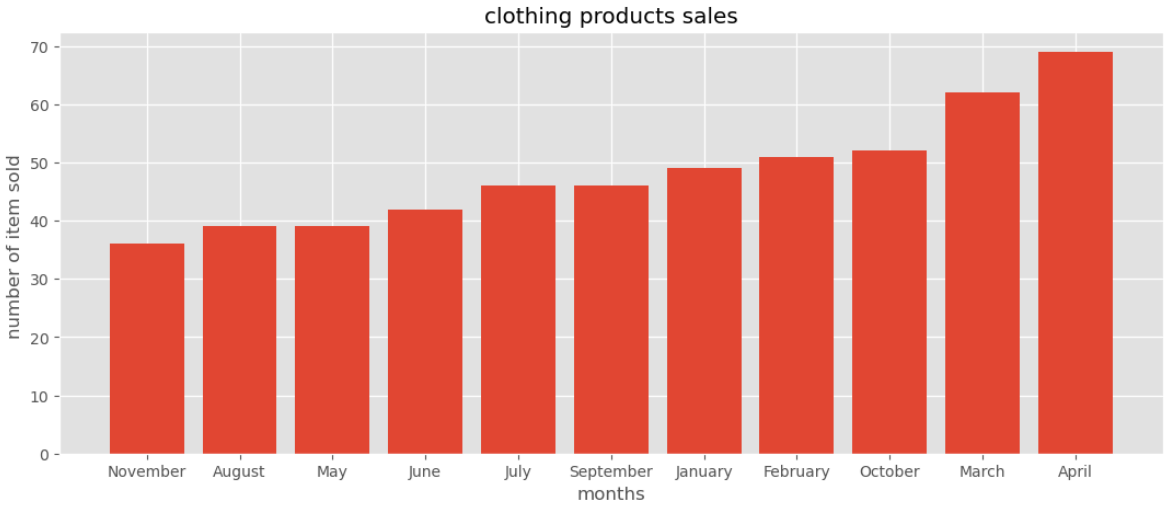
In [41]:
```
sports_insigts
```

Out[41]:

	month	Total_price	avg_dis	count
8	November	6684.64	20.441176	34
9	October	8730.58	18.953488	43
2	February	10225.61	18.863636	44
5	June	8237.01	15.000000	44
10	September	9594.97	17.500000	44
3	January	9981.53	17.978723	47
1	August	8928.81	23.000000	50
4	July	10840.61	19.901961	51
0	April	11756.67	23.148148	54
7	May	11395.04	21.203704	54
6	March	12143.32	15.909091	55

In [42]:
```
clothing_summary = data.query("`Category` == 'Clothing'")
```

In [43]:
```
clothing_insights  = clothing_summary.groupby(['month']).agg(Total_price = ("Fin
                                        count = ("month", "count")).reset_index(

# visualize data

plt.figure(figsize=(13,5))
plt.bar(x=clothing_insights['month'] , height=clothing_insights['count'] )
plt.xlabel("months")
plt.ylabel("number of item sold ")
plt.title("clothing products sales")
plt.show()
```



take aways

- there are more clothing products sold in month of April followed by March octobar , February

- There are less clothing product soled in month of navember , may , August

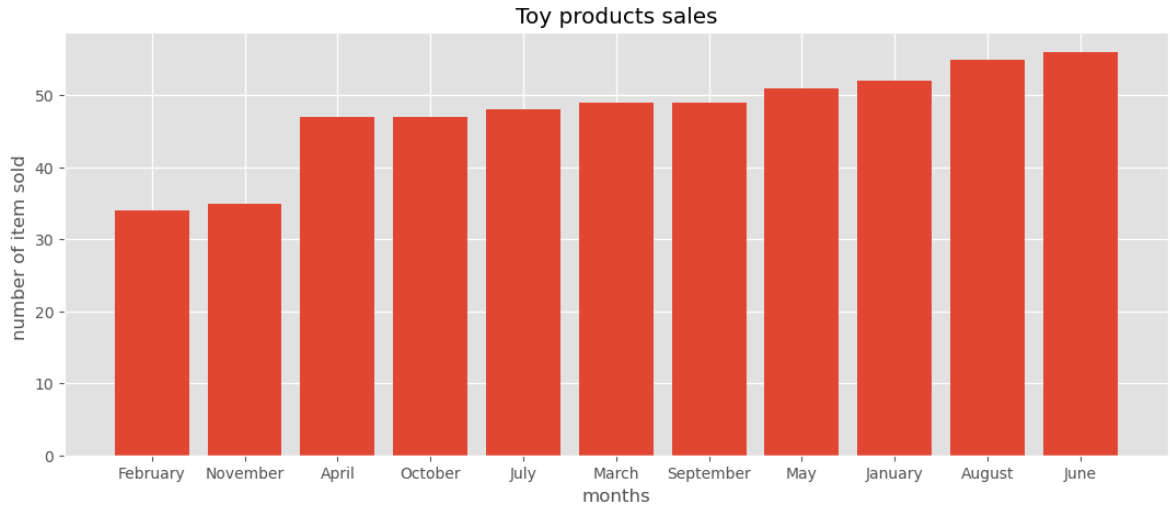
In [46]:
```
data['Category']
```

Out[46]:
```
0              Sports
1            Clothing
2              Sports
3                Toys
4              Beauty
             ...
3655           Beauty
3656             Toys
3657    Home & Kitchen
3658       Electronics
3659    Home & Kitchen
Name: Category, Length: 3660, dtype: object
```

In [47]:
```
# Filtering the data

Toys = data.query("`Category` == 'Toys'")
```

In [48]:
```
Toys_summary  = Toys.groupby(['month']).agg(Total_amount = ("Final_Price" ,"sum"
                              avg_dis = ("Discount" , "mean") , count = ("month","
                              .reset_index().sort_values(by="count" )
```

In [49]:
```
# visualize data

plt.figure(figsize=(13,5))
plt.bar(x=Toys_summary['month'] , height=Toys_summary['count'] )
plt.xlabel("months")
plt.ylabel("number of item sold ")
plt.title("Toy products sales")
plt.show()
```

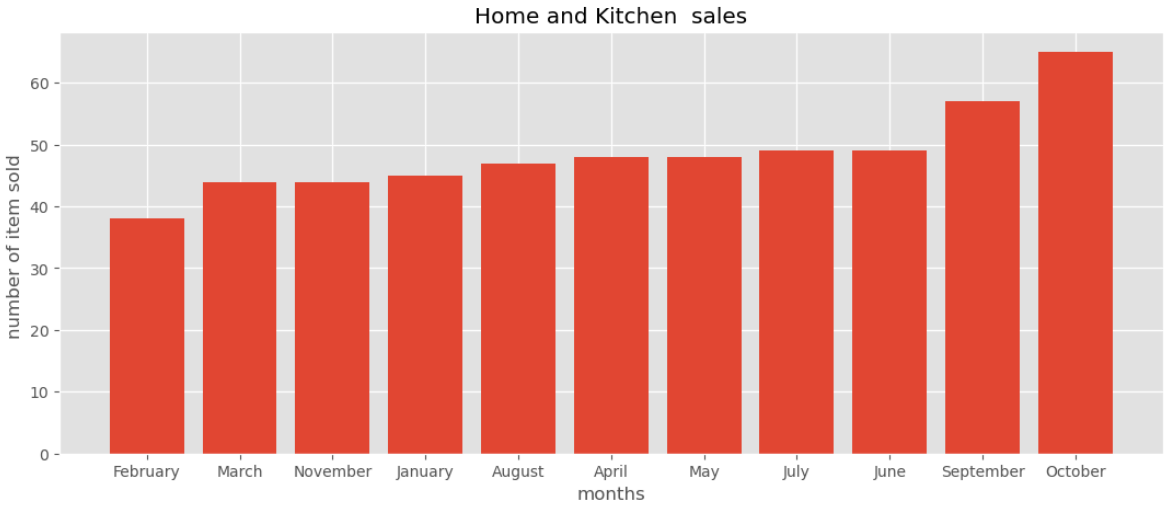


take aways

- More toys were sold in the months of June, August, January, and May.
- Fewer toys were sold in the months of February, November, and April.

In [ ]:

In [52]:
```
Books = data.query("`Category` == 'Books'")

Books_Summary =Books.groupby(['month']).agg(Total_amount = ("Final_Price" ,"sum"
                            avg_dis = ("Discount" , "mean") , count = ("month","
                            .reset_index().sort_values(by="count" )
# visualize data

plt.figure(figsize=(13,5))
plt.bar(x=Books_Summary['month'] , height=Books_Summary['count'] )
plt.xlabel("months")
plt.ylabel("number of item sold ")
plt.title("Home and Kitchen  sales")

plt.show()
```



Takeaways

- More books were sold in the months of October, September, June, and July.
-

Fewer books were sold in the months of February, March, and November.

In [ ]:

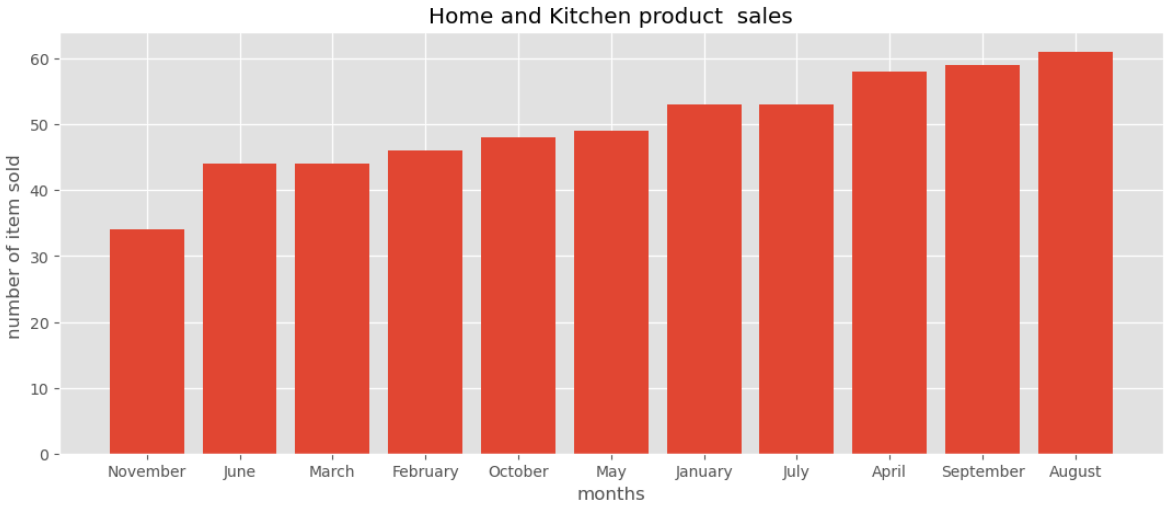
In [ ]:

In [ ]:

In [ ]:

In [54]:
```
# Home kitchen analysis

Home_kitchen = data.query("`Category` == 'Home & Kitchen'")

Home_kitchen_Summary = Home_kitchen.groupby(['month']).agg(Total_amount = ("Fina
                            avg_dis = ("Discount" , "mean") , count = ("month","
                            .reset_index().sort_values(by="count" )
```

```
# visualize data

plt.figure(figsize=(13,5))
plt.bar(x=Home_kitchen_Summary['month'] , height=Home_kitchen_Summary['count'] )
plt.xlabel("months")
plt.ylabel("number of item sold ")
plt.title("Home and Kitchen product  sales")

plt.show()
```



take aways

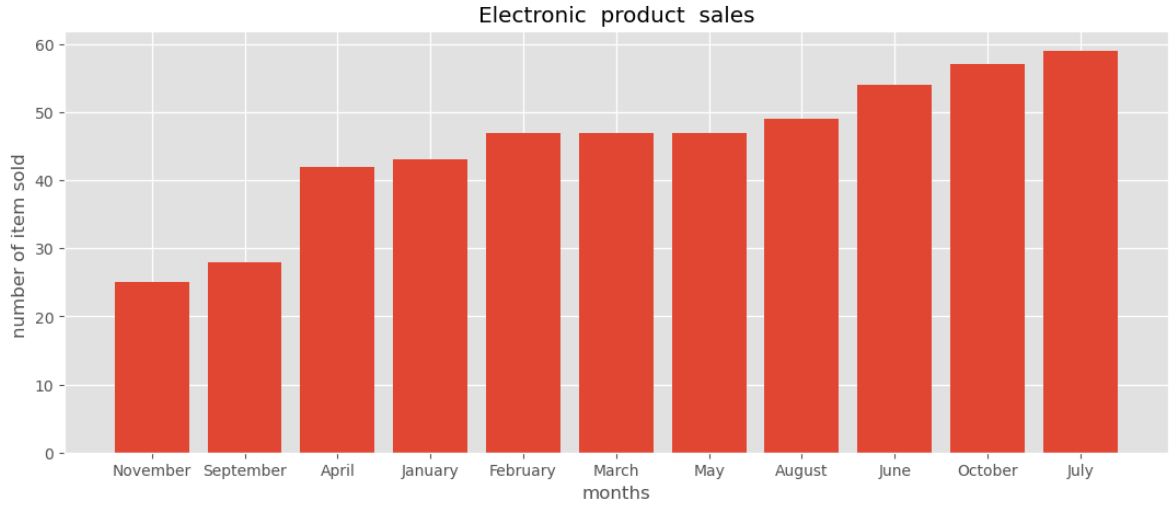
- More home and kitchen products were sold in the months of August, September, and April.
-

Fewer home and kitchen products were sold in the months of November, June, and March.

In [56]:
```
data['Category'].unique()
```

Out[56]:
```
array(['Sports', 'Clothing', 'Toys', 'Beauty', 'Books', 'Home & Kitchen',
       'Electronics'], dtype=object)
```

In [57]:
```
Electronics = data.query("`Category` == 'Electronics'")

Electronics_Summary = Electronics.groupby(['month']).agg(Total_amount = ("Final_
                            avg_dis = ("Discount" , "mean") , count = ("month","
                            .reset_index().sort_values(by="count" )
# visualize data

plt.figure(figsize=(13,5))
plt.bar(x=Electronics_Summary['month'] , height=Electronics_Summary['count'] )
plt.xlabel("months")
plt.ylabel("number of item sold ")
plt.title("Electronic  product  sales")

plt.show()
```



In [ ]:

Take aways

- More electronics products were sold in the months of July, October, and June.
-

Fewer electronics products were sold in the months of November, September, and April.

In [ ]:

In [ ]:

Conclusion

1. **Discount Trends:**
   - **Home and Kitchen items** offer the highest discounts, driving significant purchases, followed by **Sports** and **Electronics**.
   - **Clothing** and **Toys** provide lower discounts, which may affect their sales.
2. **Category Sales Performance:**
   - **Sports:** Peak sales in **March**, low in **November** and **October**.
   - **Clothing:** Best months are **April** and **March**, weakest in **November** and **August**.
   - **Toys:** Strong sales in **June** and **August**, weak in **February** and **November**.
   - **Books:** High sales in **October** and **September**, low in **March** and **November**.
   - **Home and Kitchen:** Strong in **August**, weak in **November** and **June**.
   - **Electronics:** Best sales in **July**, weak in **November** and **April**.
3. **Seasonality:**
   - Sales vary significantly by category and season, with **November** generally a slow month for most categories.

Recommendations

1. **Increase Discounts:** Focus on higher discounts for **Clothing** and **Toys** to boost sales.
2. **Seasonal Campaigns:** Target peak months for promotions (e.g., **Sports in March**, **Toys in June**, **Books in October**).
3. **Boost Low-Performance Months:** Address weak sales in **November** with strategic offers across categories.
4. **Optimize Inventory:** Align stock levels with peak sales months to meet demand efficiently.

These steps will help maximize sales and align promotions with consumer behavior.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Conclusion

1. **Discount Trends:**
   - **Home and Kitchen items** offer the highest discounts, driving significant purchases, followed by **Sports** and **Electronics**.
   - **Clothing** and **Toys** provide lower discounts, which may affect their sales.
2. **Category Sales Performance:**
   - **Sports:** Peak sales in **March**, low in **November** and **October**.
   - **Clothing:** Best months are **April** and **March**, weakest in **November** and **August**.
   - **Toys:** Strong sales in **June** and **August**, weak in **February** and **November**.
   - **Books:** High sales in **October** and **September**, low in **March** and **November**.
   - **Home and Kitchen:** Strong in **August**, weak in **November** and **June**.
   - **Electronics:** Best sales in **July**, weak in **November** and **April**.

3. **Seasonality:**
    - Sales vary significantly by category and season, with **November** generally a slow month for most c

gories.

Recommendations

1. **Increase Discounts:** Focus on higher discounts for **Clothing** and **Toys** to boost sales.
2. **Seasonal Campaigns:** Target peak months for promotions (e.g., **Sports in March**, **Toys in June**, **Books in October**).
3. **Boost Low-Performance Months:** Address weak sales in **November** with strategic offers across categories.
4. **Optimize Inventory:** Align stock levels with peak sales months to meet d with consumer behavior.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: