In [3]:

```python
## import python liabraries
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:

```python
## read the dataset
df= pd.read_excel("C:/Users/pv11379/Downloads/Example (2).xlsx")
```

In [106…

```python
## copy original dataset in data
data=df.copy()
```

In [5]:

```python
## dataset top 2 rows with head
df.head(2)
```

Out[5]:

	Customer	Age	Sex	Groceries	Choco-bars	Type	Satisfied	Bulk
0	1	36	Male	180	3	White	4	3
1	2	45	Male	180	4	Milk	3	2

In [6]:

```python
## dataset bottom 2 rows with head
df.tail(2)
```

Out[6]:

	Customer	Age	Sex	Groceries	Choco-bars	Type	Satisfied	Bulk
48	49	36	Male	180	4	White	3	3
49	50	24	Male	180	2	Dark	4	3

In [7]:

```python
## dataset information  - 50 rows and 8 columns
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Customer    50 non-null     int64 
 1   Age         50 non-null     int64 
 2   Sex         50 non-null     object
 3   Groceries   50 non-null     int64 
 4   Choco-bars  50 non-null     int64 
 5   Type        50 non-null     object
 6   Satisfied   50 non-null     int64 
 7   Bulk        50 non-null     int64 
dtypes: int64(6), object(2)
memory usage: 3.3+ KB
```

In [8]:

```
## dataset columns details
column= df.columns
column
```

Out[8]:

```
Index(['Customer ', 'Age ', 'Sex', 'Groceries', 'Choco-bars', 'Type ',
       'Satisfied ', 'Bulk'],
      dtype='object')
```

convert dataset columns name into lower case

In [9]:

```
mod_col=[]
for i in column:
    a=i.lower()
    mod_col.append(a)
```

In [10]:

```
print(mod_col)
```

```
['customer ', 'age ', 'sex', 'groceries', 'choco-bars', 'type ', 'satisfied ', 'bulk']
```

remove spaces in dataset columns

In [11]:

```
mod_col1=[]
for i in mod_col:
    b=i.strip()
    mod_col1.append(b)
```

In [12]:

```
print(mod_col1)
```

```
['customer', 'age', 'sex', 'groceries', 'choco-bars', 'type', 'satisfied', 'bulk']
```

In [13]:

```
## assign new columns name to dataset
df.columns=mod_col1
```

In [14]:

```
df.columns
```

Out[14]:

```
Index(['customer', 'age', 'sex', 'groceries', 'choco-bars', 'type',
       'satisfied', 'bulk'],
      dtype='object')
```

In [15]:

```
df.head(2)
```

Out[15]:

	customer	age	sex	groceries	choco-bars	type	satisfied	bulk
0	1	36	Male	180	3	White	4	3
1	2	45	Male	180	4	Milk	3	2

dataset shape information

In [16]:

```
df.shape
print("dadataset total rows are- ",df.shape[0])
print("dadataset total columns are- ",df.shape[1])
```

```
dadataset total rows are-  50
dadataset total columns are-  8
```

In [17]:

```
##dataset indexing
df.index
```

Out[17]:

```
RangeIndex(start=0, stop=50, step=1)
```

numerical analysis of dataset int columns

In [18]:

```
df.describe()
```

Out[18]:

	customer	age	groceries	choco-bars	satisfied	bulk
count	50.00000	50.000000	50.000000	50.00000	50.000000	50.000000
mean	25.50000	30.400000	196.200000	3.42000	2.720000	2.580000
std	14.57738	8.342172	34.040192	1.12649	1.050559	1.144463
min	1.00000	18.000000	150.000000	2.00000	1.000000	1.000000
25%	13.25000	23.000000	180.000000	2.00000	2.000000	1.250000
50%	25.50000	29.500000	200.000000	3.00000	3.000000	3.000000
75%	37.75000	37.750000	220.000000	4.00000	4.000000	3.750000
max	50.00000	45.000000	250.000000	5.00000	4.000000	4.000000

In [19]:

```
##numerical analysis of dataset all columns
df.describe(include='all')
```

Out[19]:

	customer	age	sex	groceries	choco-bars	type	satisfied	bulk
count	50.00000	50.000000	50	50.000000	50.00000	50	50.000000	50.000000
unique	NaN	NaN	2	NaN	NaN	3	NaN	NaN
top	NaN	NaN	Male	NaN	NaN	Dark	NaN	NaN
freq	NaN	NaN	26	NaN	NaN	26	NaN	NaN
mean	25.50000	30.400000	NaN	196.200000	3.42000	NaN	2.720000	2.580000
std	14.57738	8.342172	NaN	34.040192	1.12649	NaN	1.050559	1.144463
min	1.00000	18.000000	NaN	150.000000	2.00000	NaN	1.000000	1.000000
25%	13.25000	23.000000	NaN	180.000000	2.00000	NaN	2.000000	1.250000
50%	25.50000	29.500000	NaN	200.000000	3.00000	NaN	3.000000	3.000000
75%	37.75000	37.750000	NaN	220.000000	4.00000	NaN	4.000000	3.750000
max	50.00000	45.000000	NaN	250.000000	5.00000	NaN	4.000000	4.000000

null values in dataset columns

In [21]:

```
df.isnull().sum()
## no null values in dataset
```

Out[21]:

```
customer      0
age           0
sex           0
groceries     0
choco-bars    0
type          0
satisfied     0
bulk          0
dtype: int64
```

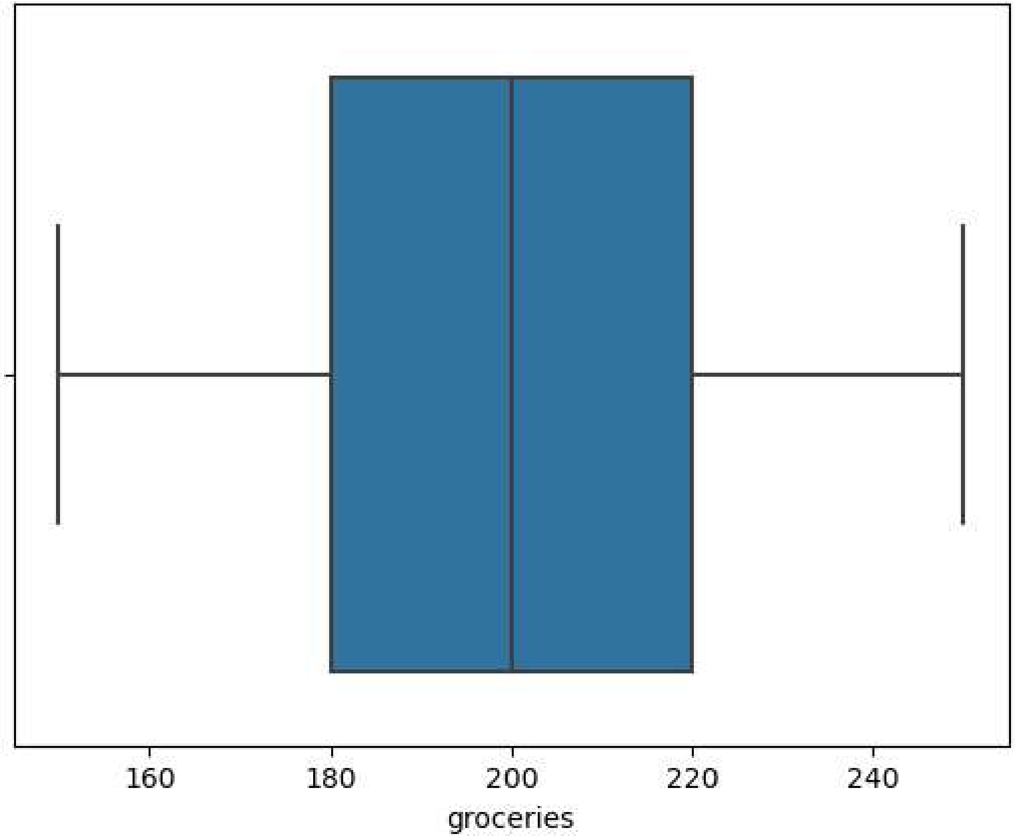
find out any outliers in groceries

In [22]:

```
sns.boxplot(x=df['groceries'])
## no outliers in groceries
```

Out[22]:

```
<Axes: xlabel='groceries'>
```



In [130…

```
## total male and female in dataset
df['sex'].value_counts()
```

Out[130]:

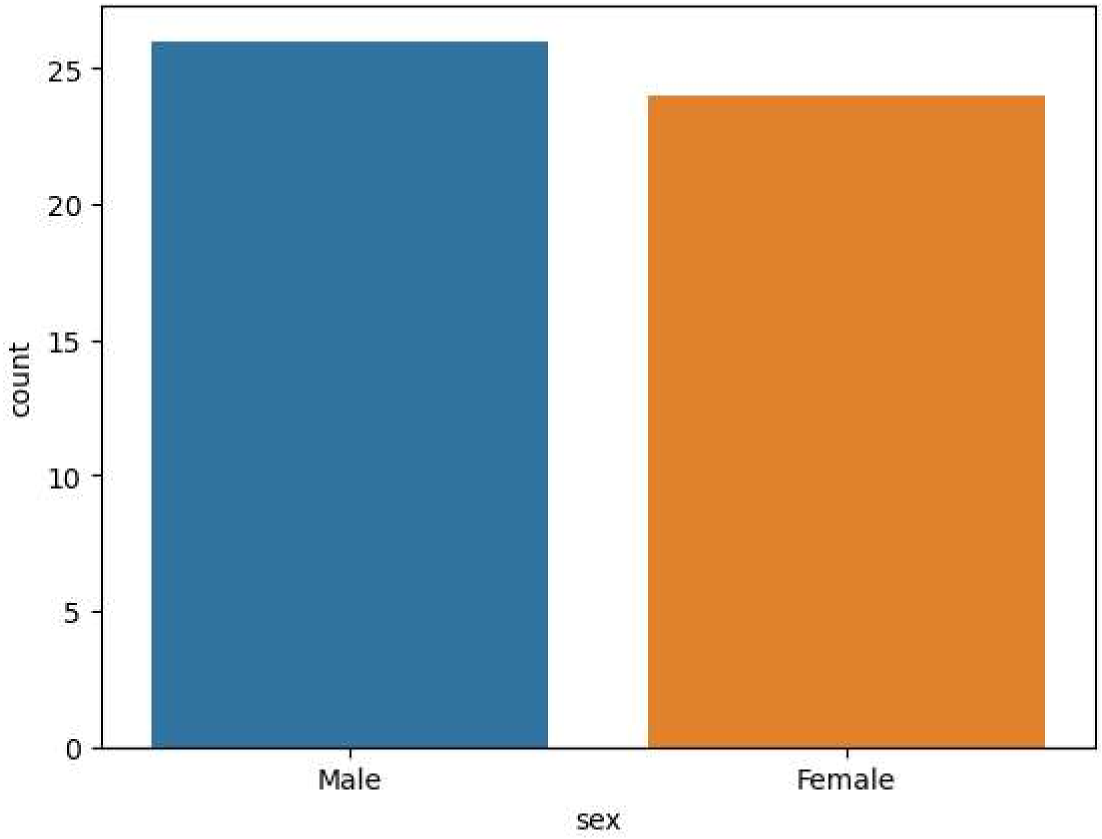
```
Male      26
Female    24
Name: sex, dtype: int64
```

In [23]:

```
## total male and female gender in dataset using seaborn
sns.countplot(x=df['sex'])
```

Out[23]:

```
<Axes: xlabel='sex', ylabel='count'>
```



In [70]:

```
## total dark,white and milk in dataset
df['type'].value_counts()
```

Out[70]:

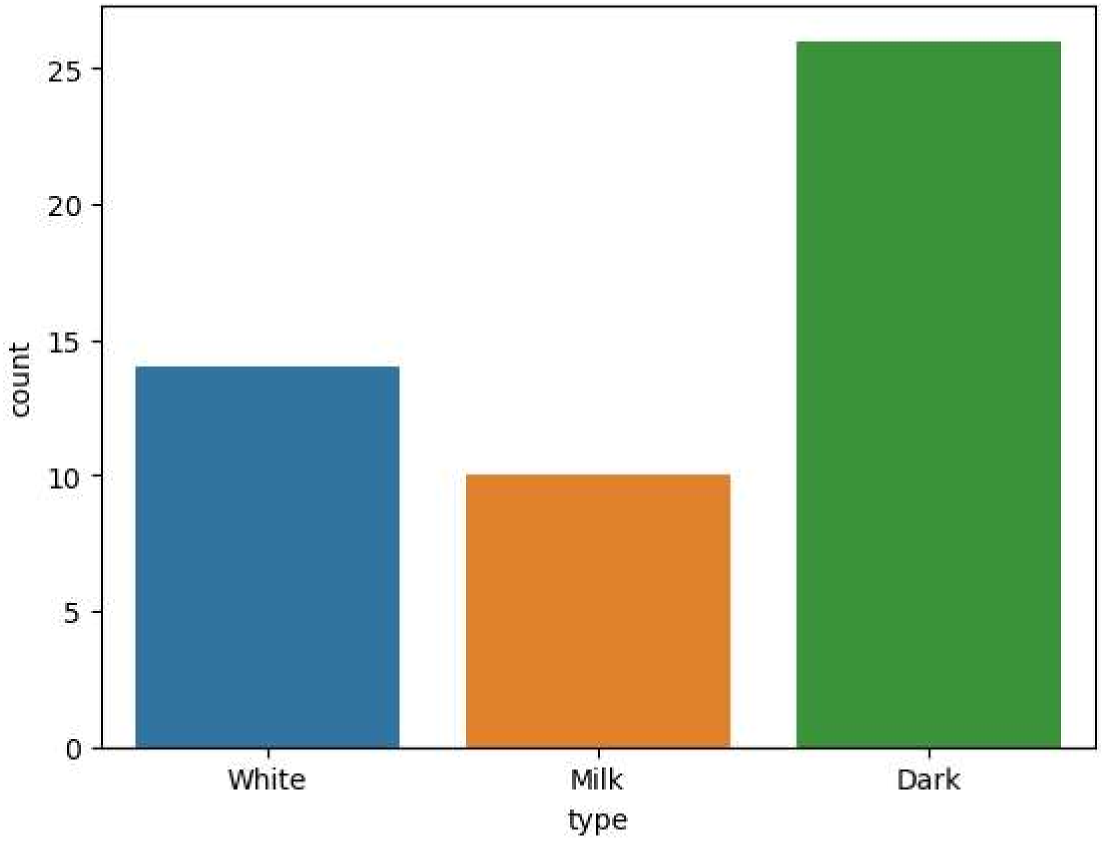
```
Dark     26
White    14
Milk     10
Name: type, dtype: int64
```

In [24]:

```
##total dark,whote and milk in dataset using seaborn
sns.countplot(x=df['type'])
```

Out[24]:

```
<Axes: xlabel='type', ylabel='count'>
```



In [79]:

```
## total groceries spends
print("Total groceries spend is - ",df['groceries'].sum())
```

```
Total groceries spend is -  9810
```

In [25]:

```
## sex wise groceries amount
df.groupby('sex')['groceries'].sum()
```

Out[25]:

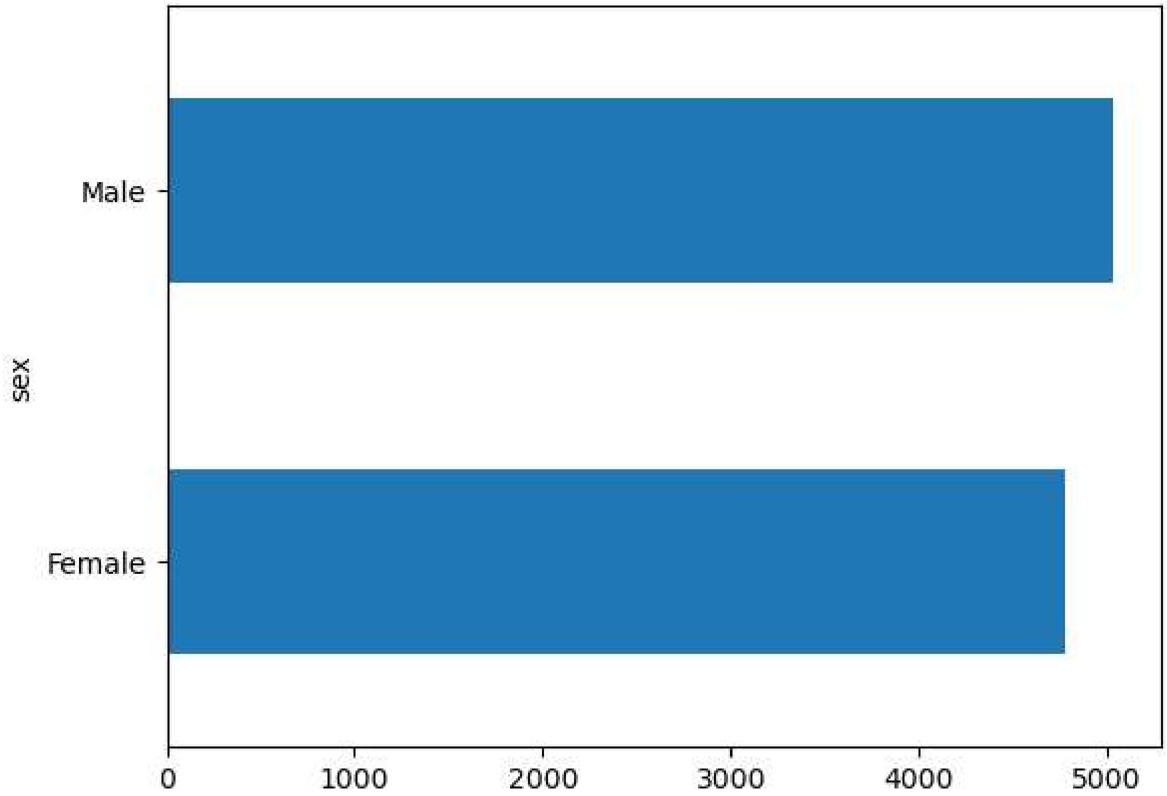
```
sex
Female    4780
Male      5030
Name: groceries, dtype: int64
```

In [26]:

```
## sex wise groceries spend bar char
df.groupby('sex')['groceries'].sum().plot(kind='barh')
```

Out[26]:

```
<Axes: ylabel='sex'>
```

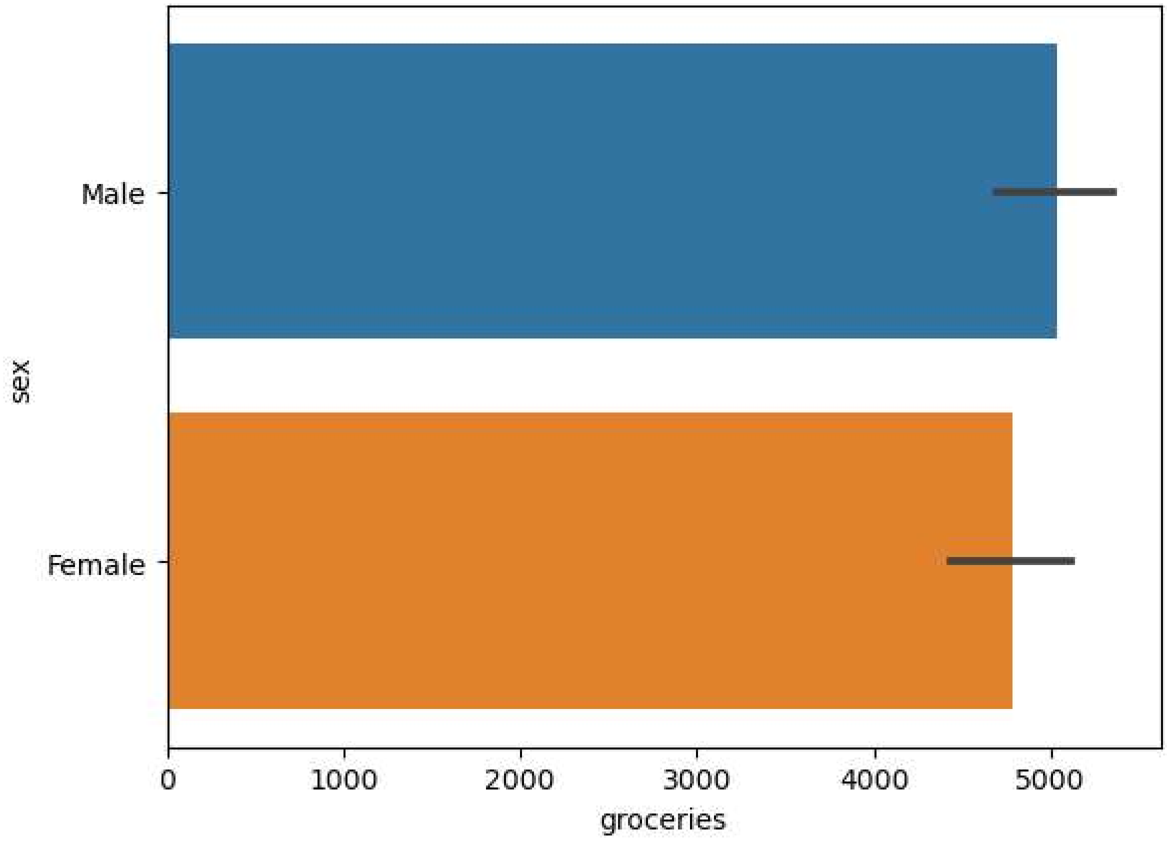


In [28]:

```
## sex wise groceries using seaborn
sns.barplot(x=df['groceries'],y=df['sex'],estimator='sum')
```

Out[28]:

```
<Axes: xlabel='groceries', ylabel='sex'>
```



In [35]:

```
## sex wise average groceries spend
sex_avg_groceries=df.groupby('sex')['groceries'].mean()
print(round(sex_avg_groceries),0)
```

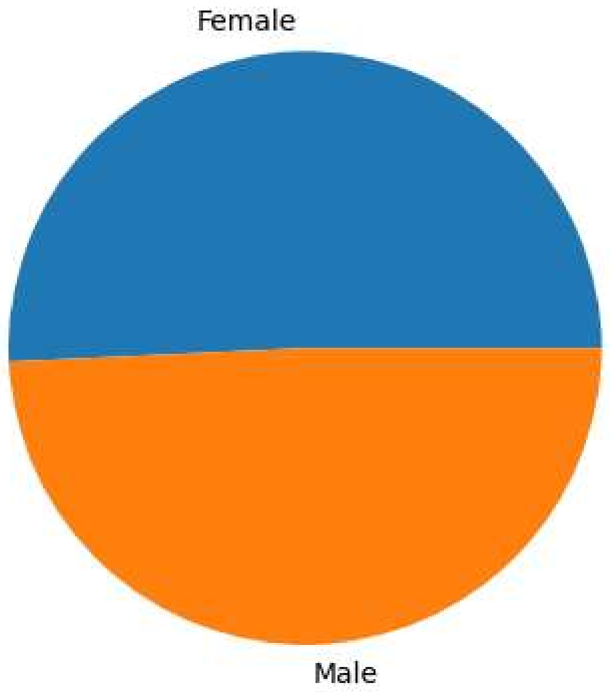
```
sex
Female    199.0
Male      193.0
Name: groceries, dtype: float64 0
```

In [36]:

```
## pie chart for sex wise average groceries spend
plt.pie(sex_avg_groceries.values,labels=sex_avg_groceries.index)
```

Out[36]:

```
([<matplotlib.patches.Wedge at 0x1cda1ac20d0>,
  <matplotlib.patches.Wedge at 0x1cda307b6d0>],
 [Text(-0.025104995935312163, 1.099713480493482, 'Female'),
  Text(0.025104995935312274, -1.099713480493482, 'Male')])
```

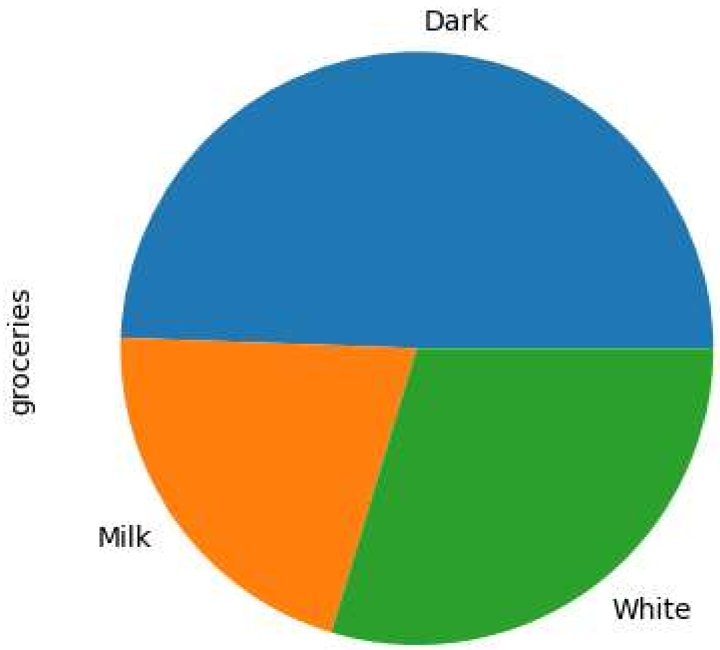


```
In [84]: ## type wise groceries spend
         df.groupby(['type'])['groceries'].sum()
```

```
Out[84]: type
         Dark     4850
         Milk     2050
         White    2910
         Name: groceries, dtype: int64
```

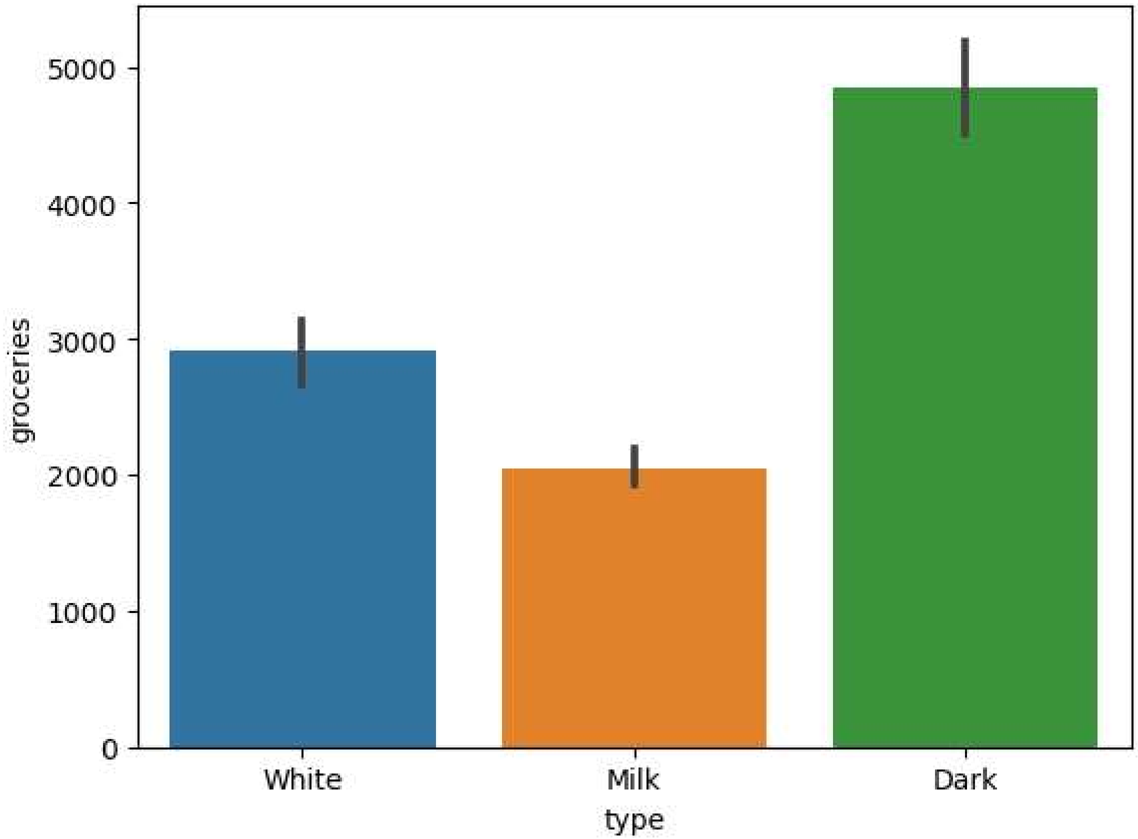
```
In [85]: df.groupby(['type'])['groceries'].sum().plot(kind='pie')
```

```
Out[85]: <Axes: ylabel='groceries'>
```



```
In [38]: sns.barplot(x=df['type'],y=df['groceries'],estimator='sum')
```

```
Out[38]: <Axes: xlabel='type', ylabel='groceries'>
```



In [88]:

```
##sex and type wise total groceries spend
df.groupby(['sex','type'])['groceries'].sum()
```

Out[88]:

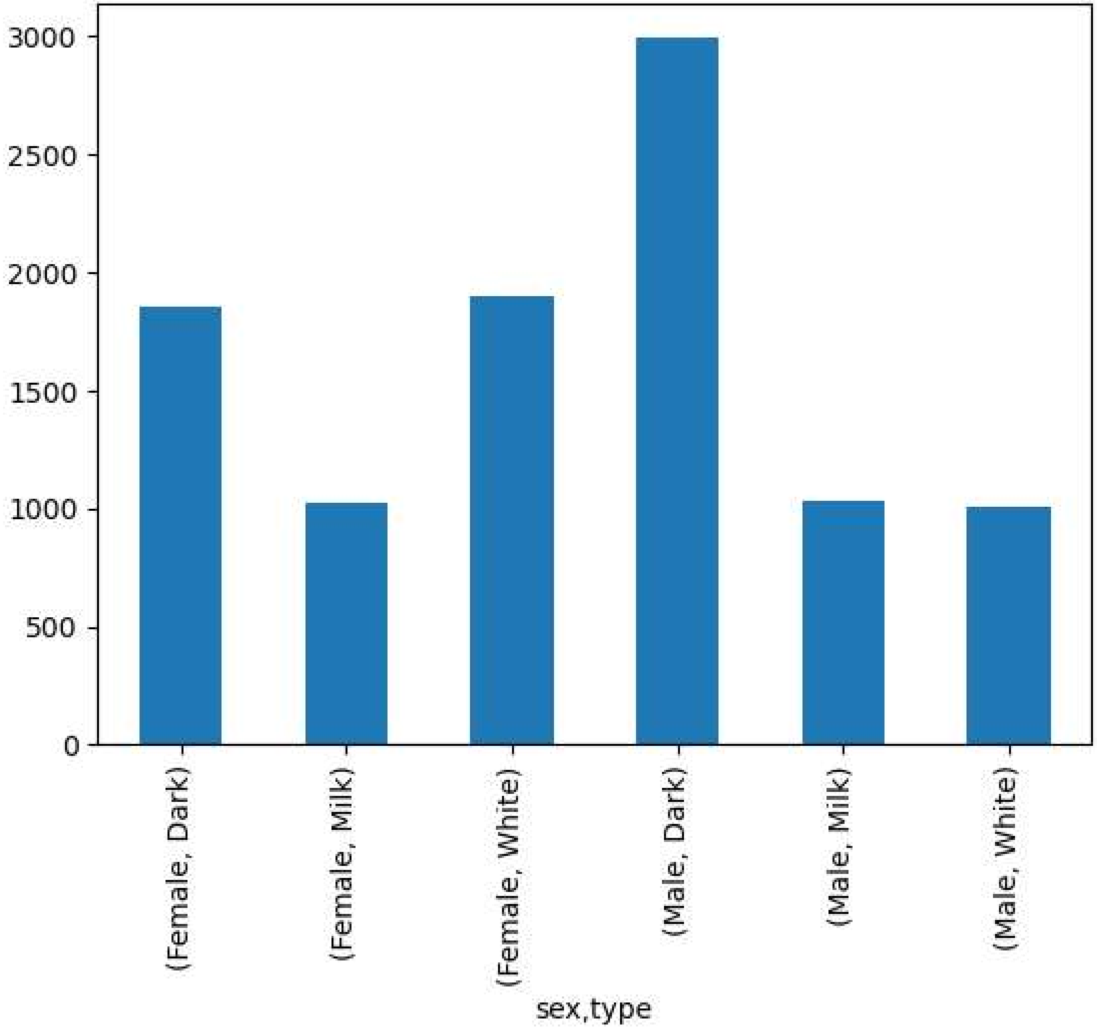
```
sex     type 
Female  Dark     1860
        Milk     1020
        White    1900
Male    Dark     2990
        Milk     1030
        White    1010
Name: groceries, dtype: int64
```

In [89]:

```
df.groupby(['sex','type'])['groceries'].sum().plot(kind='bar')
```

Out[89]:

```
<Axes: xlabel='sex,type'>
```

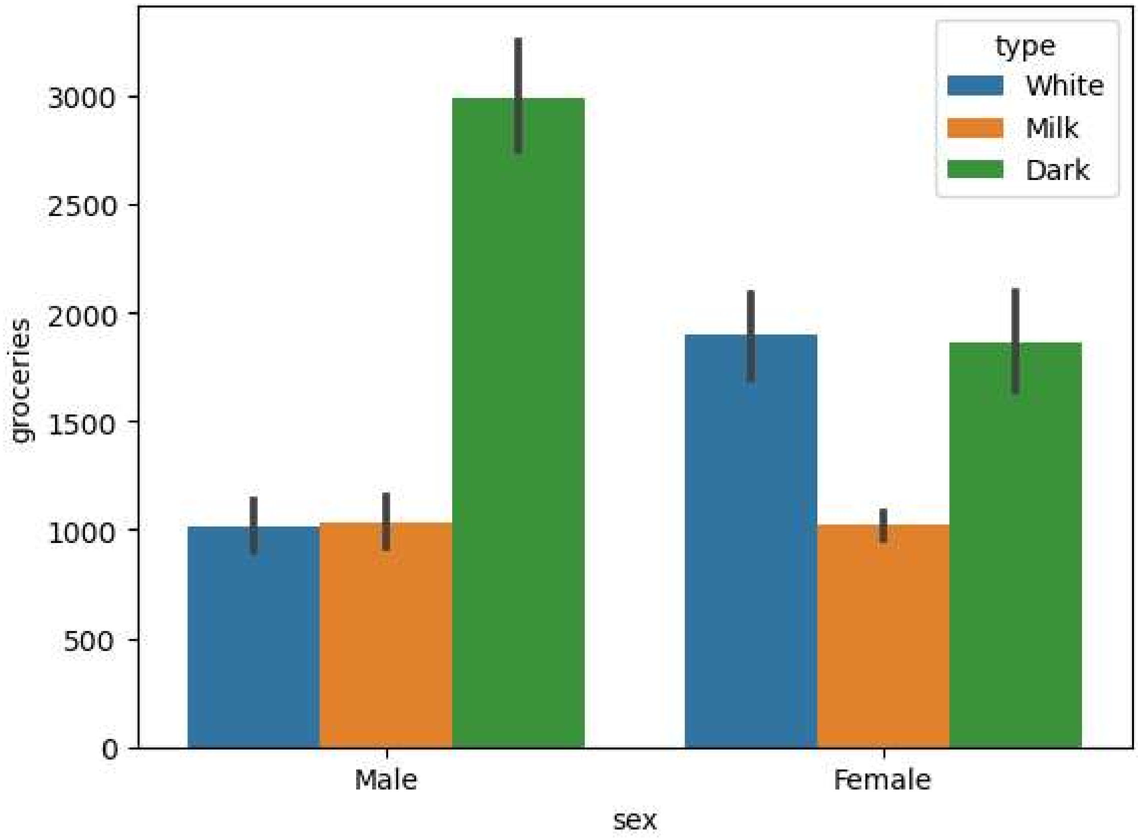


In [39]:

```python
sns.barplot(x=df['sex'],y=df['groceries'],hue=df['type'],estimator='sum')
```

Out[39]:

```
<Axes: xlabel='sex', ylabel='groceries'>
```



In [42]:

```
## correlation in dataset
corrdf= df.corr()
print(corrdf)
```

```
            customer       age  groceries  choco-bars  satisfied      bulk
customer    1.000000 -0.032389  -0.065598    0.136085  -0.141257  0.078901
age        -0.032389  1.000000   0.019835   -0.118140   0.038656 -0.097474
groceries  -0.065598  0.019835   1.000000   -0.021395   0.163671  0.241078
choco-bars  0.136085 -0.118140  -0.021395    1.000000  -0.295230  0.044640
satisfied  -0.141257  0.038656   0.163671   -0.295230   1.000000 -0.048885
bulk        0.078901 -0.097474   0.241078    0.044640  -0.048885  1.000000
```

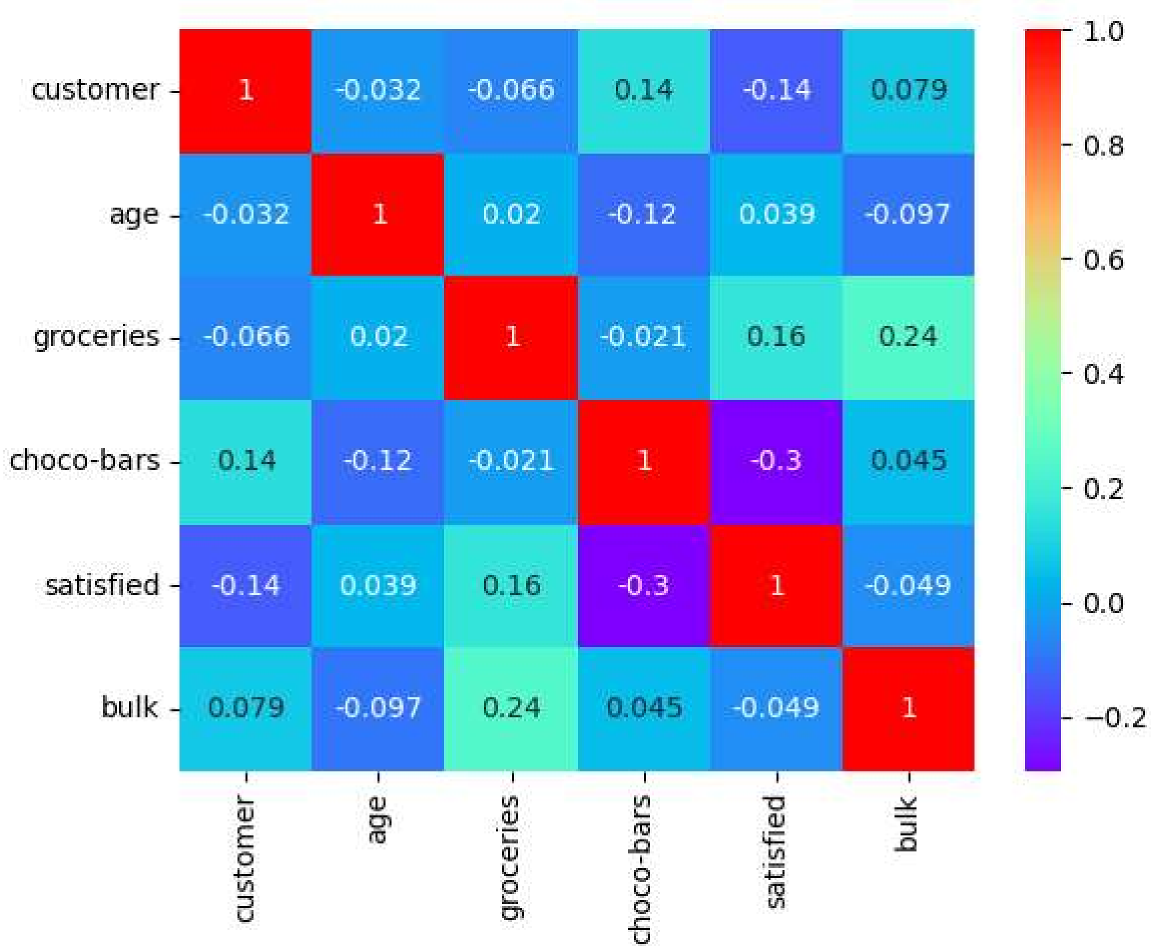
```
C:\Users\pv11379\AppData\Local\Temp\ipykernel_19448\2374362197.py:2: FutureWarning: The default value of numeric_only i
n DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  corrdf= df.corr()
```

In [43]:

```
## heatmap for correlation
sns.heatmap(corrdf,annot=True,cmap='rainbow')
## found no any strong correlation in dataset
```

Out[43]:

```
<Axes: >
```



In [46]:

```
## satisfied rating wise count
df['satisfied'].value_counts()
```

Out[46]:

```
4    15
2    15
3    13
1     7
Name: satisfied, dtype: int64
```

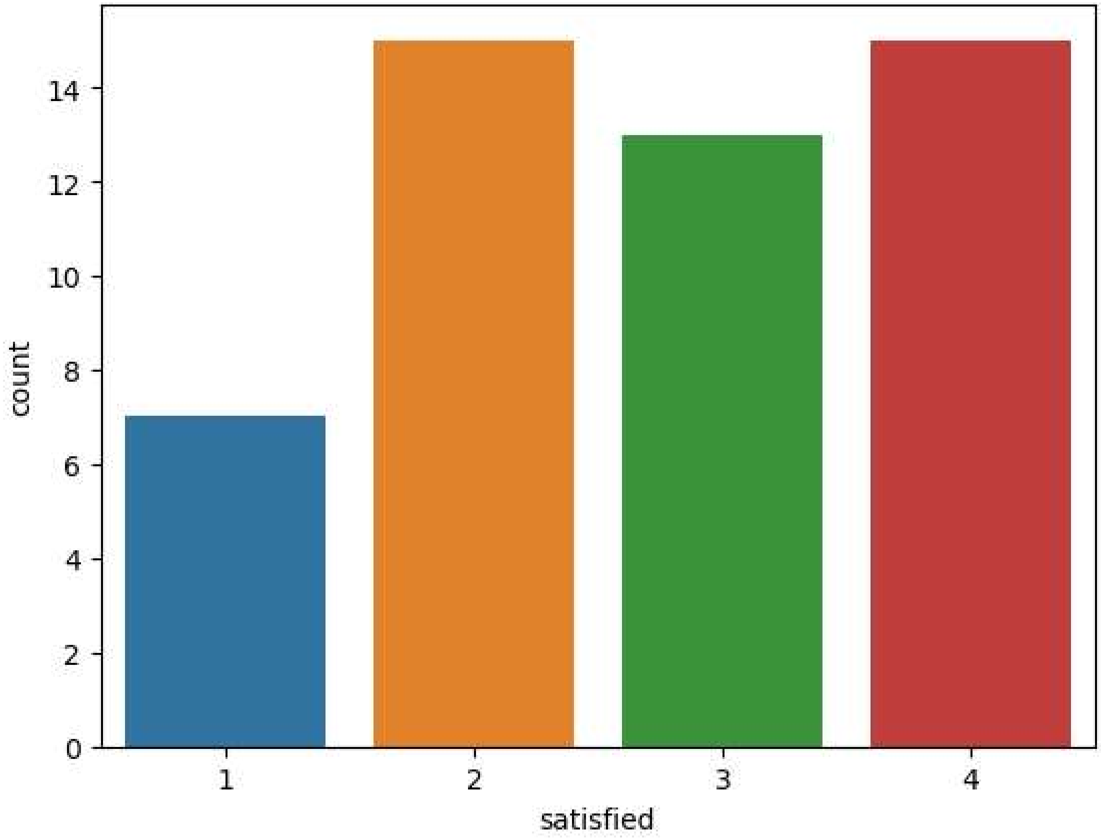
In [45]:

```
sns.countplot(x=df['satisfied'])

## 4 and 2 rating are highest in ratings
```

Out[45]:

```
<Axes: xlabel='satisfied', ylabel='count'>
```



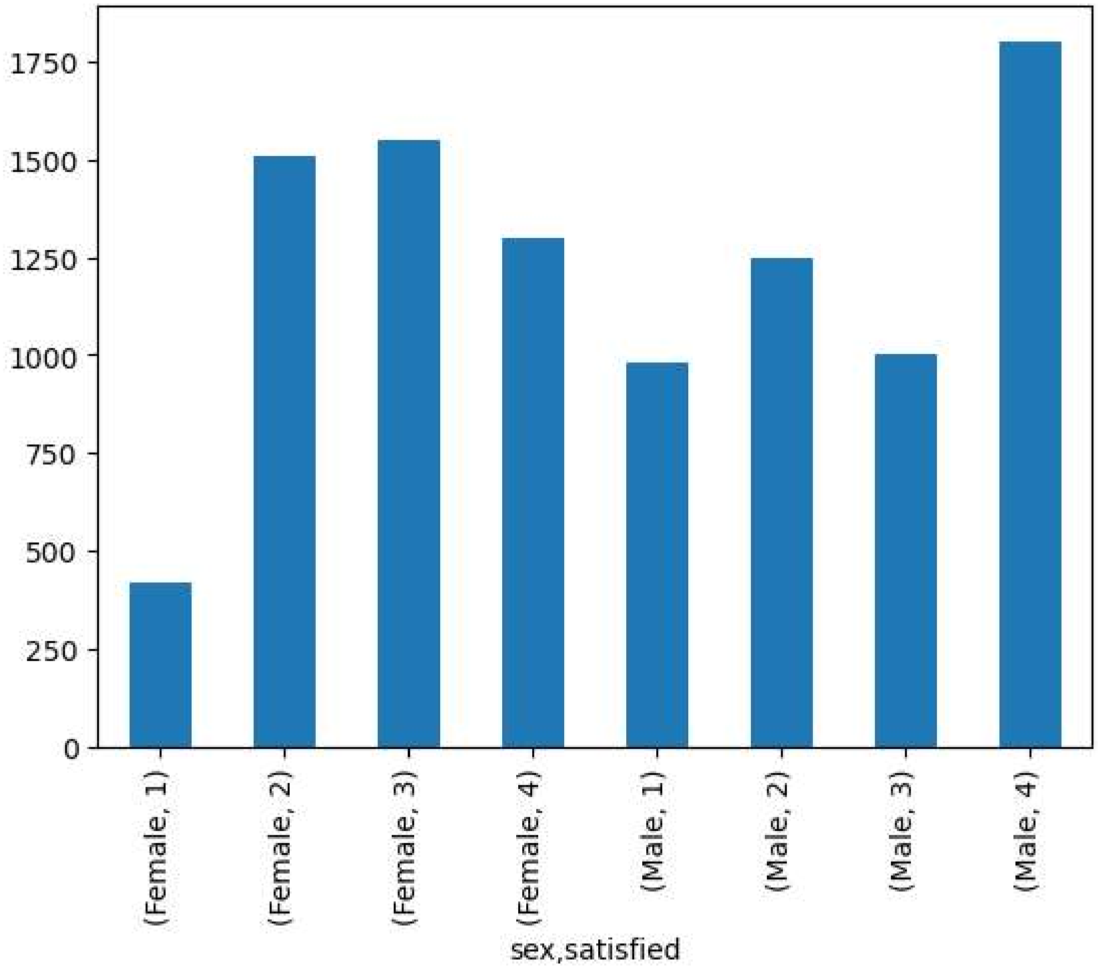
In [56]:

```
## rating wise grooceries spend by male and female
df.groupby(['sex','satisfied'])['groceries'].sum()
```

```
Out[56]: sex     satisfied
         Female  1             420
                 2            1510
                 3            1550
                 4            1300
         Male    1             980
                 2            1250
                 3            1000
                 4            1800
         Name: groceries, dtype: int64
```

```
In [57]: df.groupby(['sex','satisfied'])['groceries'].sum().plot(kind='bar')
```

```
Out[57]: <Axes: xlabel='sex,satisfied'>
```

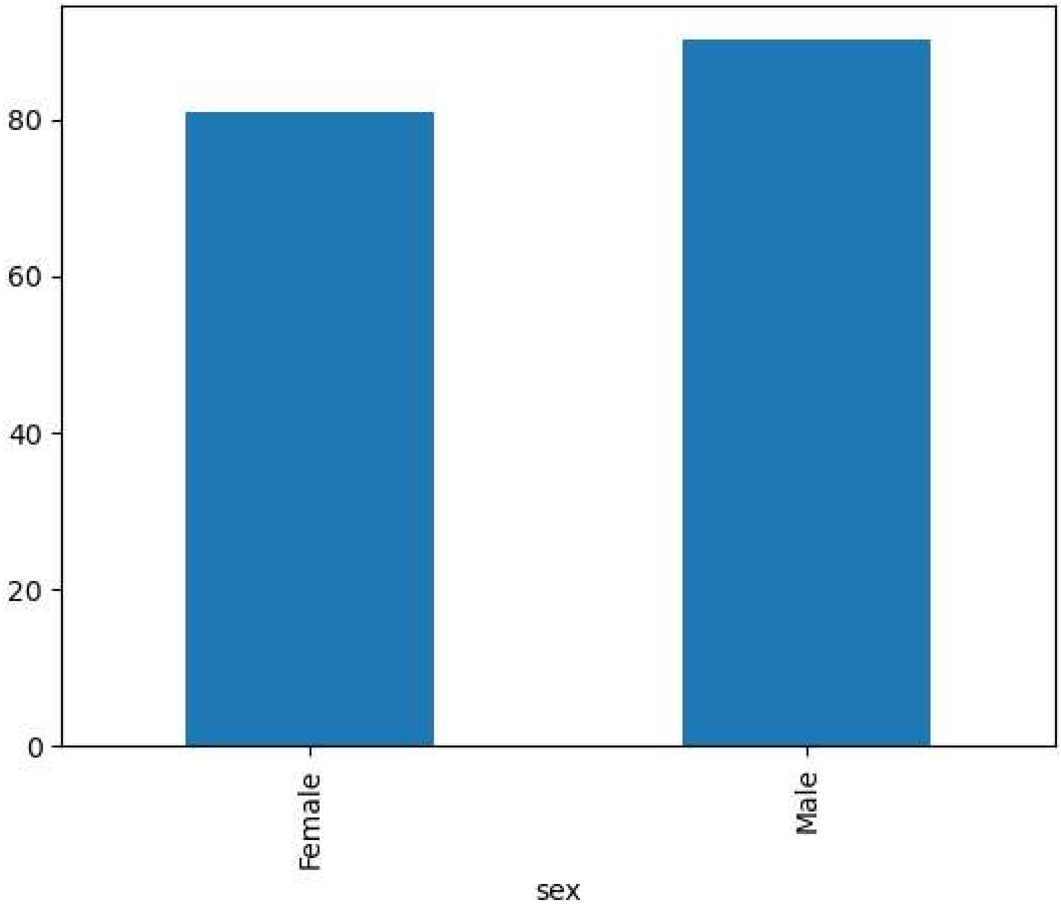


In [59]:
```
## average age of customer
print("average age of customer is - ",round(df['age'].mean(),0))
```

```
average age of customer is -  30.0
```

In [65]:
```
## total choco-bar purchased by male and female
df.groupby('sex')['choco-bars'].sum().plot(kind='bar')
```

Out[65]:
```
<Axes: xlabel='sex'>
```

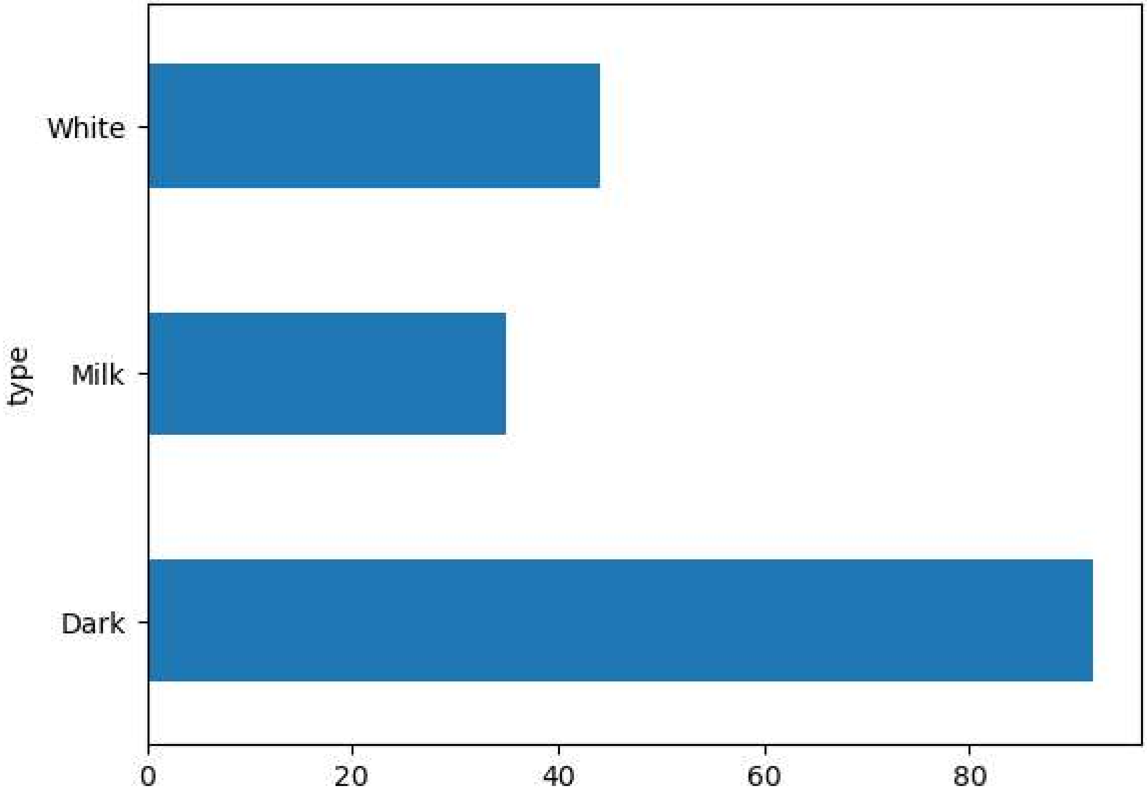


In [67]:

```
## type wise choco-bar purchased
df.groupby('type')['choco-bars'].sum().plot(kind='barh')
## highest purchased choco-bar is dark chocobar.
```

Out[67]:

```
<Axes: ylabel='type'>
```



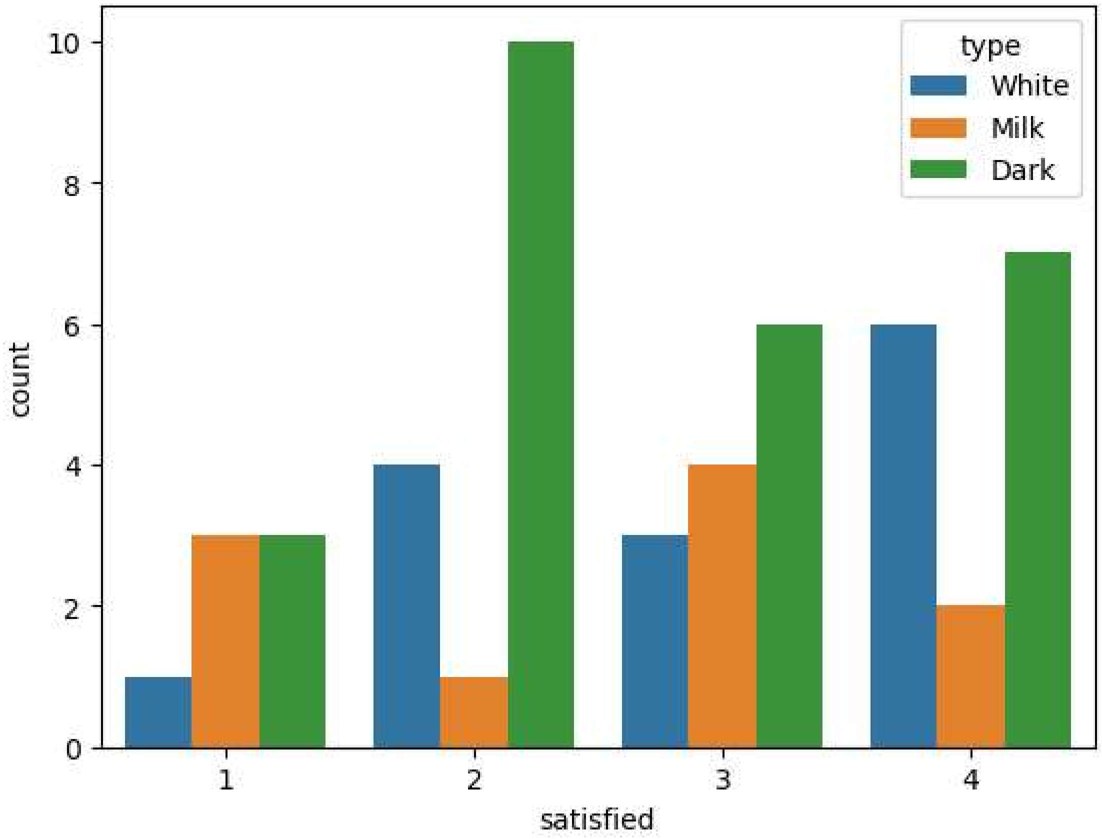
highest rating choco-bar

In [105…

```
sns.countplot(x=df['satisfied'],hue=df['type'])
## highest(4) rating in dark choco-bar
```

Out[105]:

```
<Axes: xlabel='satisfied', ylabel='count'>
```



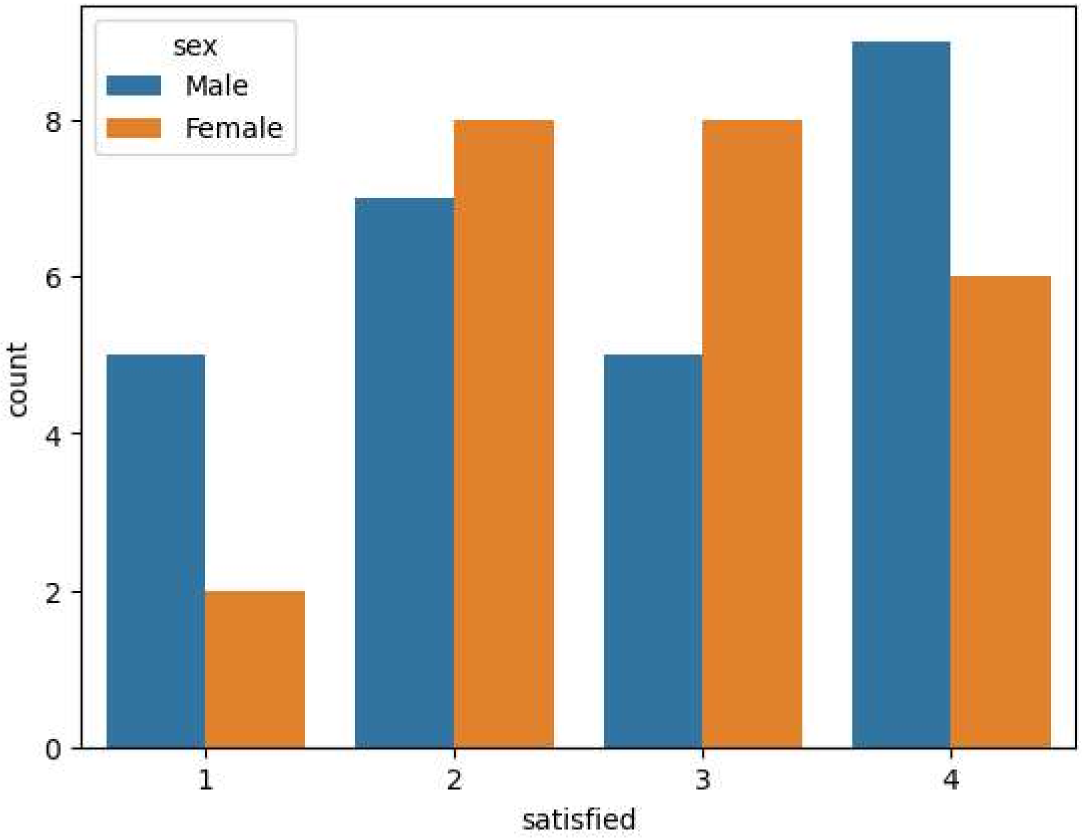
sex wise ratings given

In [76]:

```
sns.countplot(x=df['satisfied'],hue=df['sex'])
## male has given highest rating
```

Out[76]:

```
<Axes: xlabel='satisfied', ylabel='count'>
```



Top five age who purchased max groceries

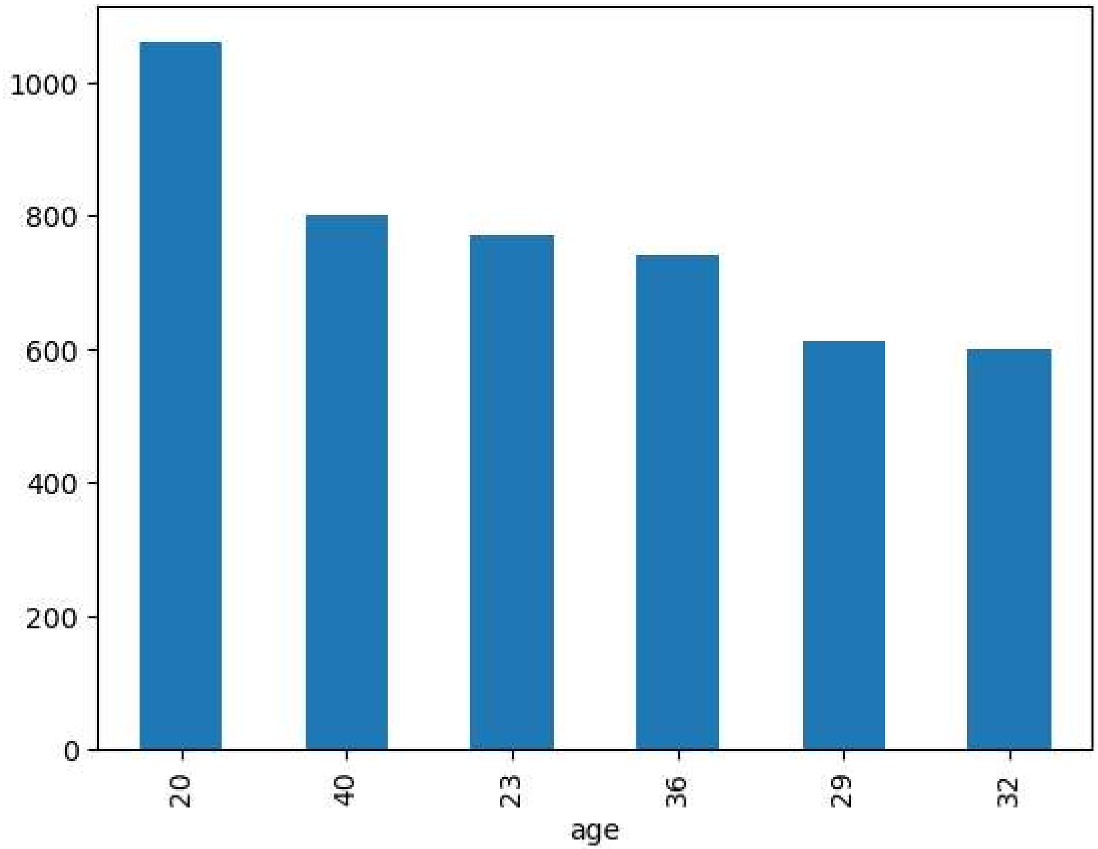
In [107…

```
age_gro=df.groupby('age')['groceries'].sum()
print("top five age category who purchased more groceries are-\n",(age_gro.sort_values(ascending=False).iloc[0:6]))
```

```
top five age category who purchased more groceries are-
 age
20    1060
40     800
23     770
36     740
29     610
32     600
Name: groceries, dtype: int64
```

```
In [104…  age_gro.sort_values(ascending=False).iloc[0:6].plot(kind='bar')
```

```
Out[104]: <Axes: xlabel='age'>
```



```
In [ ]:
```