Gala Groceries Exploratory Data Analysis

Necessary imports

In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
%matplotlib inline
warnings.filterwarnings('ignore')
```

Loading Data

In [3]:
```
df = pd.read_csv('sample_sales_data.csv')
df
```

Out[3]:

	Unnamed: 0	transaction_id	timestamp	product_id	category	customer_type	unit_price	quantity	total	payment_type
0	0	a1c82654-c52c-45b3-8ce8-4c2a1efe63ed	2022-03-02 09:51:38	3bc6c1ea-0198-46de-9ffd-514ae3338713	fruit	gold	3.99	2	7.98	e-wallet
1	1	931ad550-09e8-4da6-beaa-8c9d17be9c60	2022-03-06 10:33:59	ad81b46c-bf38-41cf-9b54-5fe7f5eba93e	fruit	standard	3.99	1	3.99	e-wallet
2	2	ae133534-6f61-4cd6-b6b8-d1c1d8d90aea	2022-03-04 17:20:21	7c55cbd4-f306-4c04-a030-628cbe7867c1	fruit	premium	0.19	2	0.38	e-wallet
3	3	157cebd9-aaf0-475d-8a11-7c8e0f5b76e4	2022-03-02 17:23:58	80da8348-1707-403f-8be7-9e6deeccc883	fruit	gold	0.19	4	0.76	e-wallet
4	4	a81a6cd3-5e0c-44a2-826c-aea43e46c514	2022-03-05 14:32:43	7f5e86e6-f06f-45f6-bf44-27b095c9ad1d	fruit	basic	4.49	2	8.98	debit card
...	...	...	...	...	...	...	...	...	...	...
7824	7824	6c19b9fc-f86d-4526-9dfe-d8027a4d13ee	2022-03-03 18:22:09	bc6187a9-d508-482b-9ca6-590d1cc7524f	cleaning products	basic	14.19	2	28.38	e-wallet
7825	7825	1c69824b-e399-4b79-a5e7-04a3a7db0681	2022-03-04 19:14:46	707e4237-191c-4cc9-85af-383a6c1cb2ab	cleaning products	standard	16.99	1	16.99	credit card
7826	7826	79aee7d6-1405-4345-9a15-92541e9e1e74	2022-03-03 14:00:09	a9325c1a-2715-41df-b7f4-3078fa5ecd97	cleaning products	basic	14.19	2	28.38	credit card
7827	7827	e5cc4f88-e5b7-4ad5-bc1b-12a828a14f55	2022-03-04 15:11:38	707e4237-191c-4cc9-85af-383a6c1cb2ab	cleaning products	basic	16.99	4	67.96	cash
7828	7828	afd70b4f-ee21-402d-8d8f-0d9e13c2bea6	2022-03-06 13:50:36	d6ccd088-11be-4c25-aa1f-ea87c01a04db	cleaning products	non-member	14.99	4	59.96	debit card

7829 rows × 10 columns

EDA

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7829 entries, 0 to 7828
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Unnamed: 0      7829 non-null   int64  
 1   transaction_id  7829 non-null   object 
 2   timestamp       7829 non-null   object 
 3   product_id      7829 non-null   object 
 4   category        7829 non-null   object 
 5   customer_type   7829 non-null   object 
 6   unit_price      7829 non-null   float64
 7   quantity        7829 non-null   int64  
 8   total           7829 non-null   float64
 9   payment_type    7829 non-null   object 
dtypes: float64(2), int64(2), object(6)
memory usage: 611.8+ KB
```

In [5]:
```
df.shape
```

Out[5]:
```
(7829, 10)
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Unnamed: 0        0
transaction_id    0
timestamp         0
product_id        0
category          0
customer_type     0
unit_price        0
quantity          0
total             0
payment_type      0
dtype: int64
```

Descriptive statistics

In [7]:
```
df.describe()
```

Out[7]:

	Unnamed: 0	unit_price	quantity	total
count	7829.000000	7829.000000	7829.000000	7829.000000
mean	3914.000000	7.819480	2.501597	19.709905
std	2260.181962	5.388088	1.122722	17.446680
min	0.000000	0.190000	1.000000	0.190000
25%	1957.000000	3.990000	1.000000	6.570000
50%	3914.000000	7.190000	3.000000	14.970000
75%	5871.000000	11.190000	4.000000	28.470000
max	7828.000000	23.990000	4.000000	95.960000

In [8]:
```
df.corr()
```

Out[8]:

	Unnamed: 0	unit_price	quantity	total
Unnamed: 0	1.000000	0.623392	0.003927	0.483878
unit_price	0.623392	1.000000	0.024588	0.792018
quantity	0.003927	0.024588	1.000000	0.521926
total	0.483878	0.792018	0.521926	1.000000

In [9]:
```
# highest sales items

pd.options.display.float_format = '{:,.0f}'.format
sales_items = df.groupby(by='category').mean().sort_values(by='total', ascending=False).head(30)
sales_items.reset_index(inplace=True)
sales_items
```

Out[9]:

	category	Unnamed: 0	unit_price	quantity	total
0	medicine	7,033	17	2	43
1	seafood	5,253	16	3	43
2	kitchen	7,346	15	2	38
3	meat	4,936	15	2	37
4	beverages	3,926	13	3	33
5	cleaning products	7,682	13	2	32
6	baby products	6,462	12	2	30
7	pets	6,654	11	2	26
8	frozen	2,400	10	3	25
9	cheese	4,598	9	3	23
10	personal care	6,823	9	3	23
11	dairy	4,264	8	3	20
12	baked goods	5,601	8	2	19
13	refrigerated items	2,056	7	3	17
14	condiments and sauces	3,685	7	2	17
15	canned foods	2,872	6	3	16
16	baking	5,954	5	2	13
17	packaged foods	3,341	5	3	13
18	spices and herbs	2,594	3	2	8
19	fruit	498	3	2	6
20	vegetables	1,420	2	3	6
21	snacks	6,218	2	2	6

In [10]:
```
df[["customer_type", "total"]].groupby(['customer_type'], as_index=False).mean().sort_values(by='total', ascending=False)
```

Out[10]:

	customer_type	total
3	premium	20
2	non-member	20
1	gold	20
4	standard	20
0	basic	19

In [11]:
```
df.nunique()
```

Out[11]:
```
Unnamed: 0        7829
transaction_id    7829
timestamp         7738
product_id         300
category            22
customer_type        5
unit_price          64
quantity             4
total              256
payment_type         4
dtype: int64
```

In [12]:
```
df['transaction_id'].value_counts()
```

Out[12]:
```
a1c82654-c52c-45b3-8ce8-4c2a1efe63ed    1
6532e258-95fd-4eb5-8c67-2bfb879a8fec    1
6fce2af3-47a0-4755-99c9-0cefb5ab6f41    1
6476e388-3990-471f-b415-3ee59ae18832    1
10afe89b-c45b-49a2-b0be-dec89a4c3f80    1
                                       ..
a9abe5ac-99d5-4d8b-bbbd-c2a207642849    1
6b0b23e8-412b-4665-8cc4-3e37f0d9e195    1
711a4162-1985-4f5a-94ca-137cfacaeadf    1
7d1e9010-dbaf-4770-a467-f31477910f7a    1
afd70b4f-ee21-402d-8d8f-0d9e13c2bea6    1
Name: transaction_id, Length: 7829, dtype: int64
```

In [13]:
```
df['customer_type'].value_counts()
```

Out[13]:
```
non-member    1601
standard      1595
premium       1590
basic         1526
gold          1517
Name: customer_type, dtype: int64
```

In [14]:
```
df['payment_type'].value_counts()
```

Out[14]:
```
cash           2027
credit card    1949
e-wallet       1935
debit card     1918
Name: payment_type, dtype: int64
```

In [15]:
```
df['total'].value_counts().mean()
```

Out[15]:
```
30.58203125
```

In [16]:
```
df['total'].value_counts().mode()
```

Out[16]:
```
0    4
Name: total, dtype: int64
```

In [17]:
```
df['total'].value_counts().median()
```

Out[17]:
```
24.0
```

Visualisation

In [18]:
```
correlation = df.corr()
```

In [19]:
```
correlation.style.background_gradient(cmap='coolwarm')
```

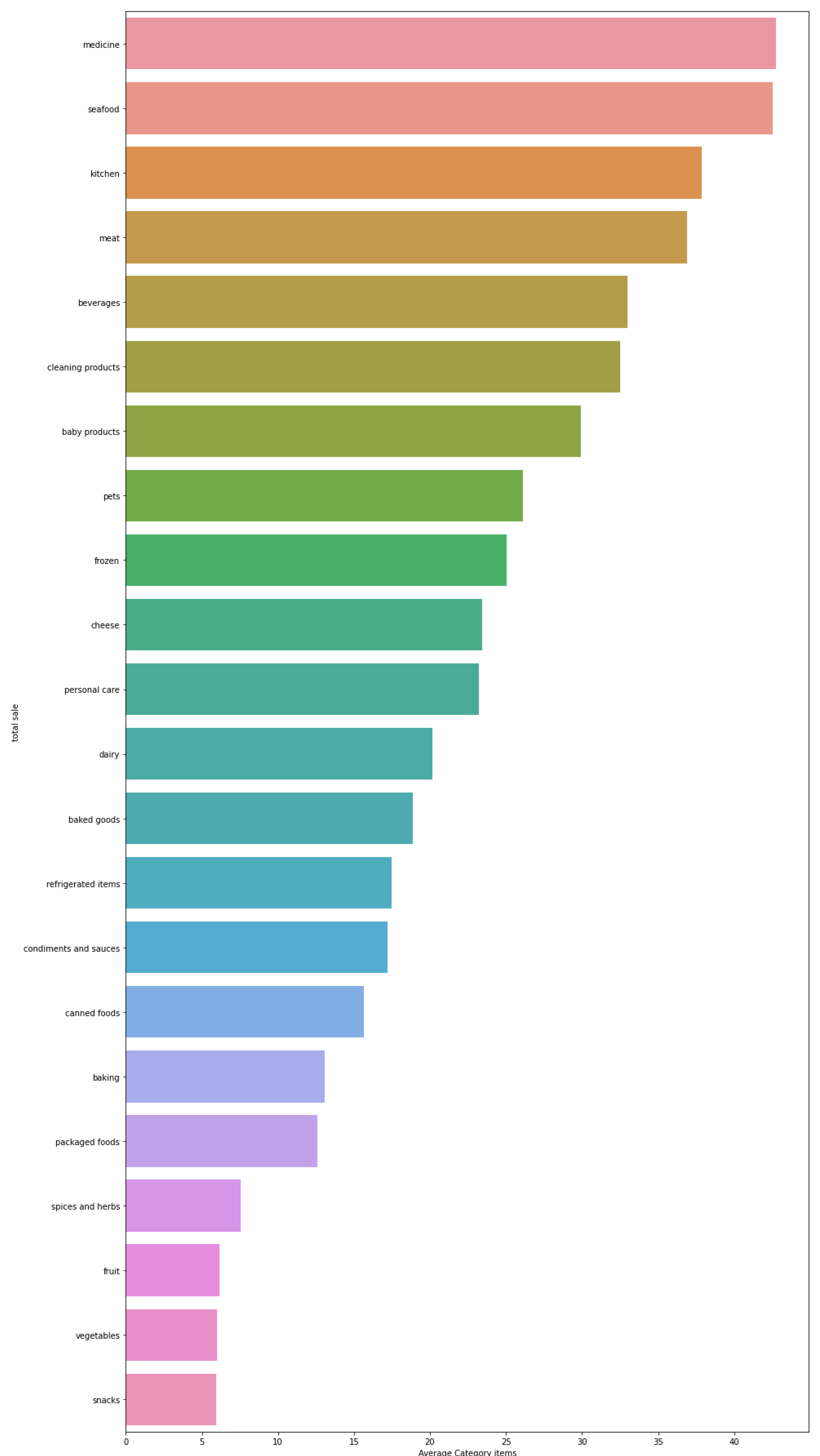
Out[19]:

	Unnamed: 0	unit_price	quantity	total
Unnamed: 0	1.000000	0.623392	0.003927	0.483878
unit_price	0.623392	1.000000	0.024588	0.792018
quantity	0.003927	0.024588	1.000000	0.521926
total	0.483878	0.792018	0.521926	1.000000

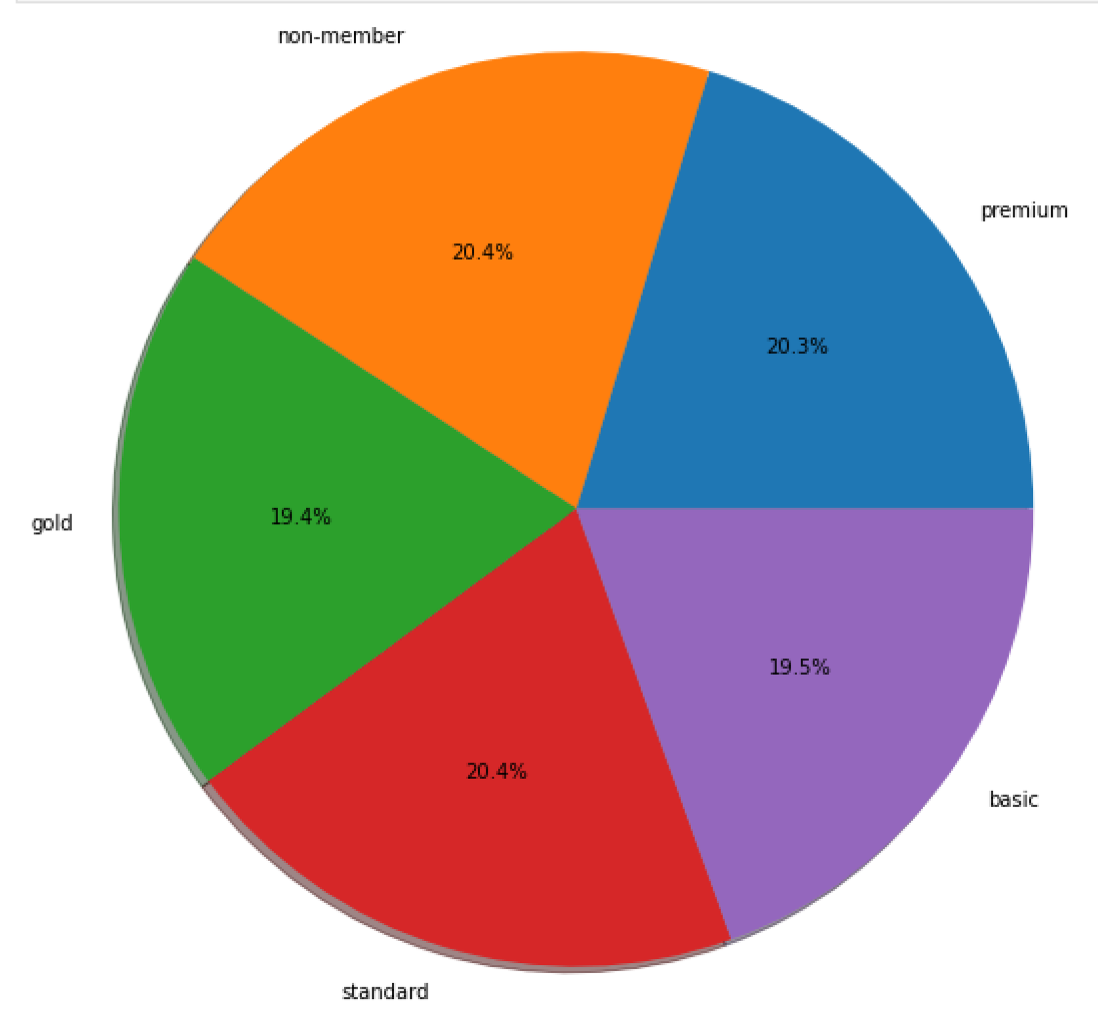
In [20]:
```
plt.figure(figsize=(15, 32))

sales_item = df.groupby(['category'])['total'].mean().sort_values(ascending=False)
title = [ tit for tit in sales_item.index]
sns.barplot(x=sales_item, y=title)
plt.xlabel('Average Category items')
plt.ylabel('total sale')
```

Out[20]:
```
Text(0, 0.5, 'total sale')
```



In [21]:
```
customer_type = ['premium', 'non-member', 'gold','standard', 'basic']
data = [1590,1601,1517,1595,1526]
fig = plt.figure(figsize =(10, 7))
plt.pie(data,labels=customer_type, shadow=True, autopct='%1.1f%%',radius=1.5)
plt.show()
```



END

In [ ]: