1. Import all the required libraries

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Import the data

In [15]:
```
mart = pd.read_csv(r"C:\Users\lenovo\Downloads\supermarket_sales - Sheet1.csv")
mart.head()
```

Out[15]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2

In [16]:
```
mart.shape
```

Out[16]:
```
(1000, 17)
```

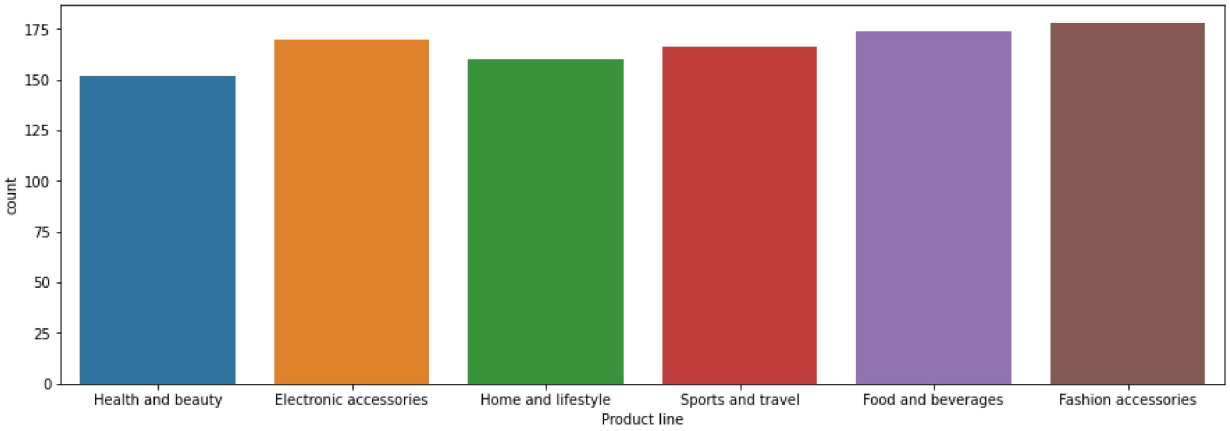
COUNT PLOT

3.Create a basic COUNTPLOT to get number of transactions for each pf the productline and change the figure size

In [24]:
```
plt.figure(figsize=(15,5))
sns.countplot(x='Product line',data=mart)

#TIP: press shift + tab on 'countplot' to see the parameters
```

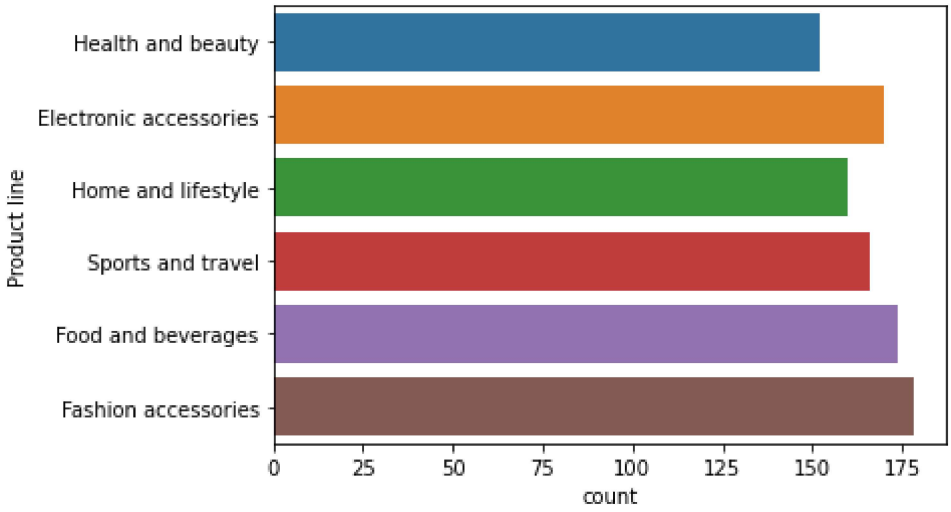
Out[24]:
```
<AxesSubplot:xlabel='Product line', ylabel='count'>
```



4.Make it horizontal barplot

In [25]:
```
sns.countplot(y='Product line',data=mart)
```

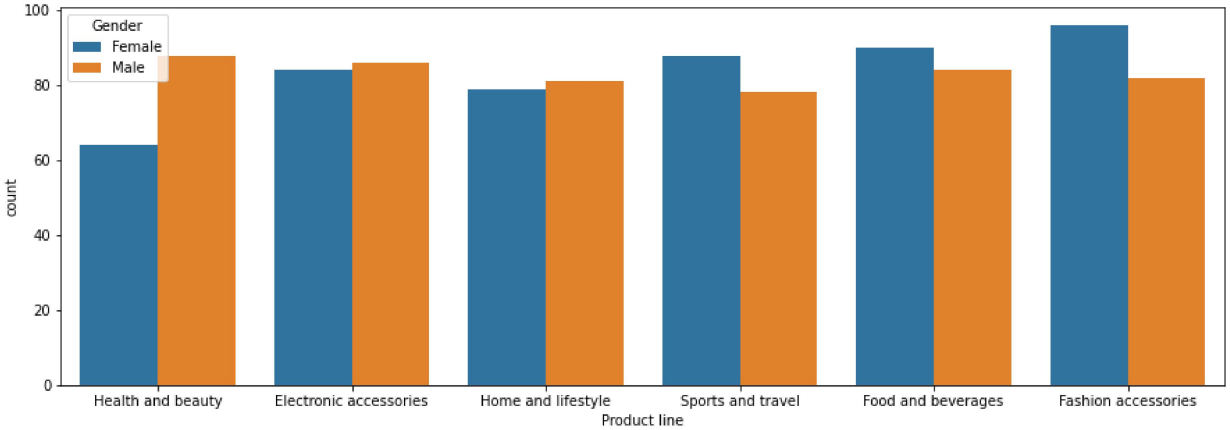
Out[25]:
```
<AxesSubplot:xlabel='count', ylabel='Product line'>
```



5. Add hue to get the count on two categories i.e, Product Line and Gender

In [27]:
```
plt.figure(figsize=(15,5))
sns.countplot(x='Product line',data=mart,hue='Gender')
```

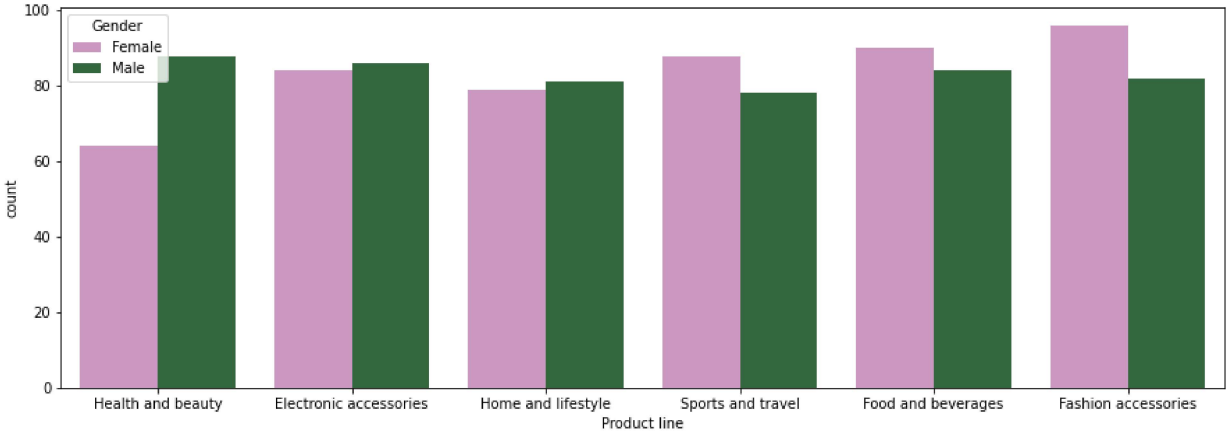
Out[27]:
```
<AxesSubplot:xlabel='Product line', ylabel='count'>
```



6. Using different color palette

In [31]:
```
plt.figure(figsize=(15,5))
sns.countplot(x='Product line', hue='Gender',data=mart,palette='cubehelix_r')
```

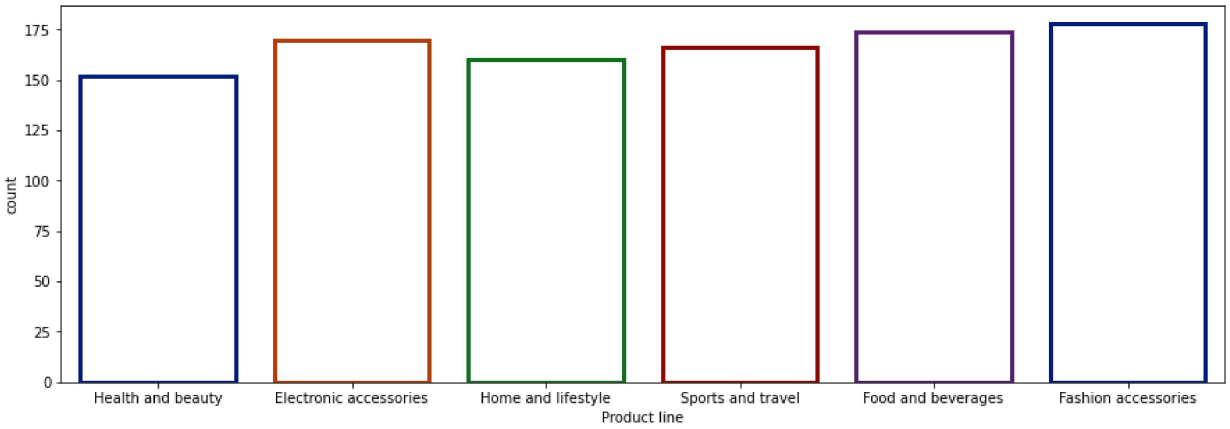
Out[31]:
```
<AxesSubplot:xlabel='Product line', ylabel='count'>
```



7. Change style using facecolor, linewidth and edge color

In [34]:
```
plt.figure(figsize=(15,5))
sns.countplot(x='Product line', data=mart
              ,facecolor=(0,0,0,0)
              ,linewidth=3
              ,edgecolor=sns.color_palette('dark',5))
```

Out[34]:
```
<AxesSubplot:xlabel='Product line', ylabel='count'>
```



BAR PLOT

A bar plot represents an estimate of central tendency for a numeric variable with height of each rectangle and provides some indication of the uncertainity around that estimate using error bars.

1. Display the seaborn dataset names and load one of them for example

In [36]:
```
sns.get_dataset_names()
```

Out[36]:
```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'exercise',
 'flights',
 'fmri',
 'gammas',
 'geyser',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'taxis',
 'tips',
 'titanic']
```

In [38]:
```
titanic = sns.load_dataset('titanic')
titanic
```

Out[38]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True
...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True

891 rows × 15 columns

2. Create a basic BARPLOT

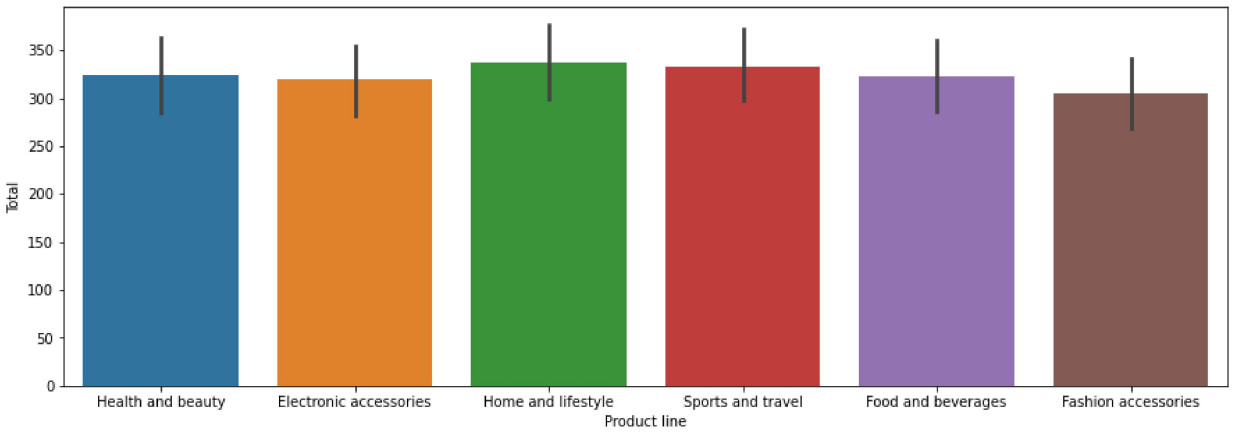
In [45]:
```
mart.head()
```

Out[45]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2

In [46]:
```
plt.figure(figsize=(15,5))
sns.barplot(x='Product line',y='Total', data=mart)
```

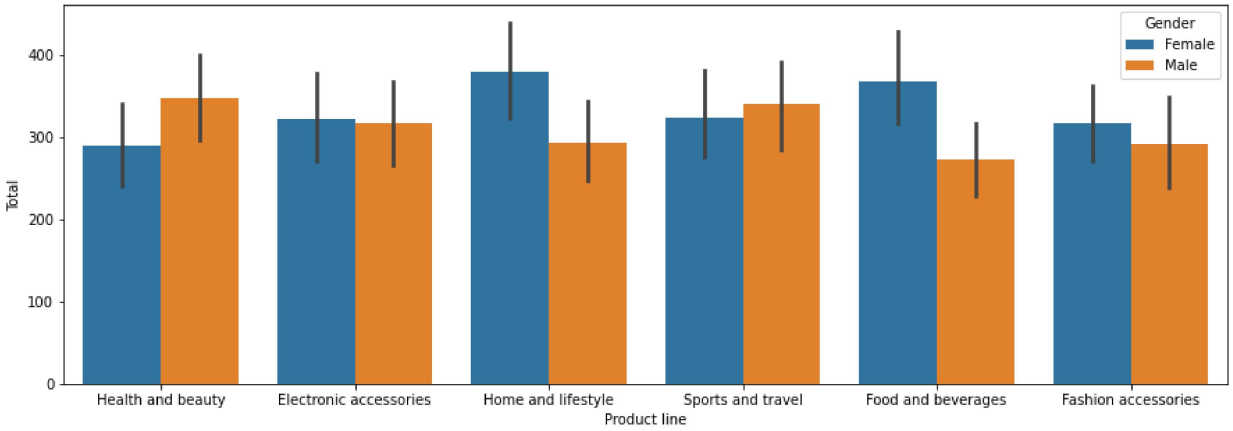
Out[46]:
```
<AxesSubplot:xlabel='Product line', ylabel='Total'>
```



3.Add HUE to the barplot

In [48]:
```
plt.figure(figsize=(15,5))
sns.barplot(x='Product line',y='Total',hue='Gender',data=mart)
```

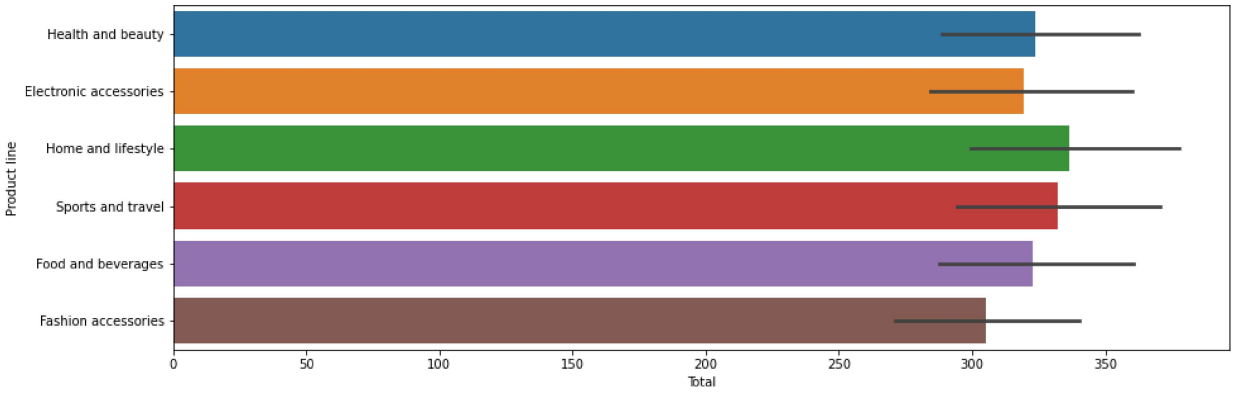
Out[48]:
```
<AxesSubplot:xlabel='Product line', ylabel='Total'>
```



4. Make the barplot Horizontal

In [50]:
```
plt.figure(figsize=(15,5))
sns.barplot(y='Product line',x='Total',data=mart)
```

Out[50]:
```
<AxesSubplot:xlabel='Total', ylabel='Product line'>
```



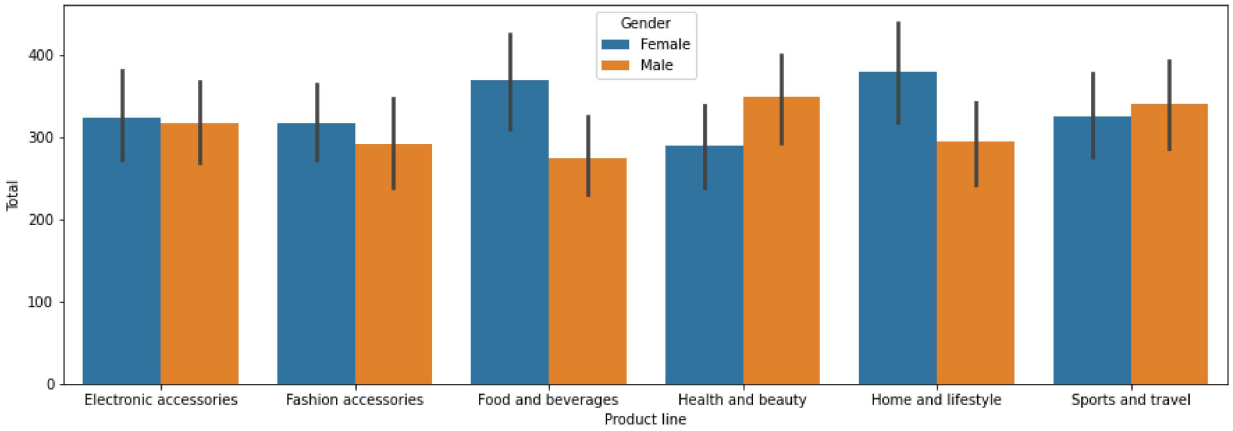
5. Plot the bars in a given order

In [53]:
```
x=mart['Product line'].sort_values()
x.unique()
```

Out[53]:
```
array(['Electronic accessories', 'Fashion accessories',
       'Food and beverages', 'Health and beauty', 'Home and lifestyle',
       'Sports and travel'], dtype=object)
```

In [60]:
```
plt.figure(figsize=(15,5))
sns.barplot(
    x='Product line',
    y='Total',
    hue='Gender',
    data=mart,
    order=['Electronic accessories','Fashion accessories','Food and beverages','Health
```

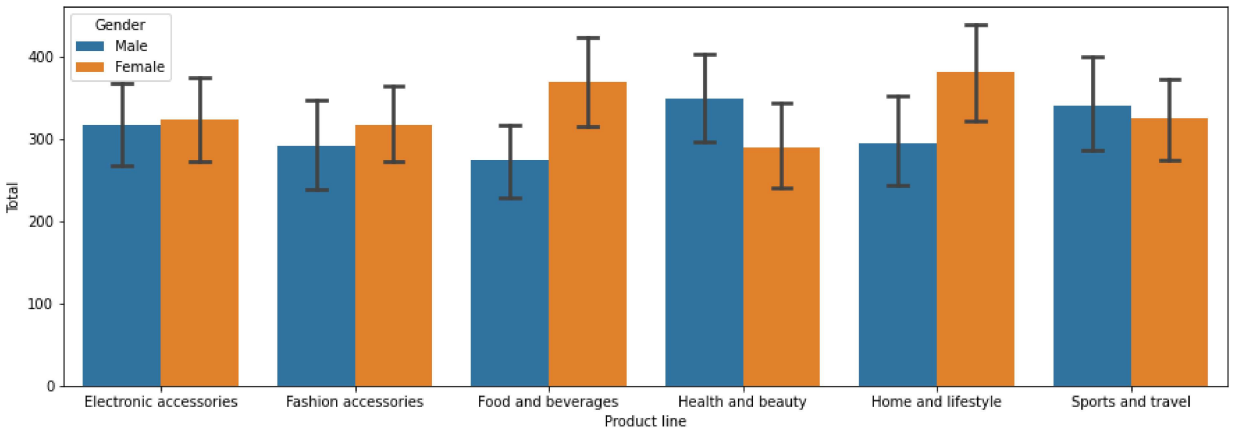
Out[60]:
```
<AxesSubplot:xlabel='Product line', ylabel='Total'>
```



6. Add CAP on the error bar

In [66]:
```
plt.figure(figsize=(15,5))
sns.barplot(
    x='Product line',
    y='Total',
    hue='Gender',
    hue_order=['Male','Female'],
    data=mart,
    order=['Electronic accessories','Fashion accessories','Food and beverages','Health
    capsize=0.1
    )
```

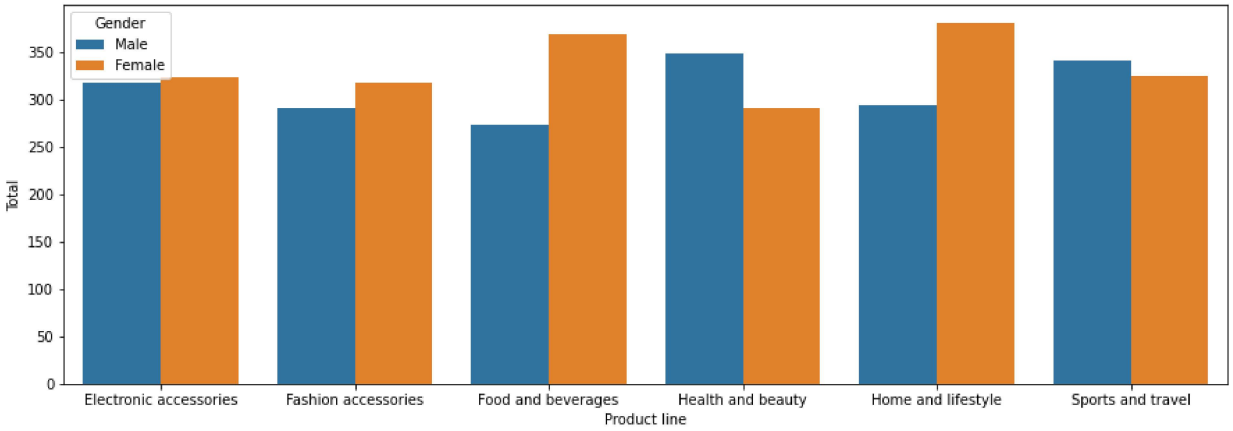
Out[66]:
```
<AxesSubplot:xlabel='Product line', ylabel='Total'>
```



7. Remove the ERROR bar using ci (class interval)

In [67]:
```
plt.figure(figsize=(15,5))
sns.barplot(
    x='Product line',
    y='Total',
    hue='Gender',
    hue_order=['Male','Female'],
    data=mart,
    order=['Electronic accessories','Fashion accessories','Food and beverages','Health
    ci=None
    )
```

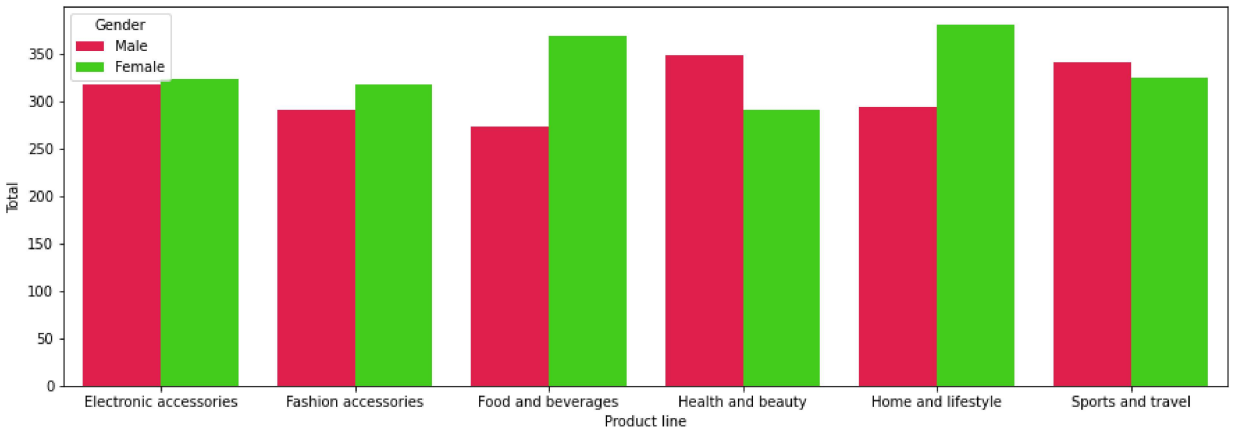
Out[67]:
```
<AxesSubplot:xlabel='Product line', ylabel='Total'>
```



8. Change bar color using Palette attribute

In [72]:
```
plt.figure(figsize=(15,5))
sns.barplot(
    x='Product line',
    y='Total',
    hue='Gender',
    hue_order=['Male','Female'],
    data=mart,
    order=['Electronic accessories','Fashion accessories','Food and beverages','Health
    ci=None,
    palette='prism_r'
    )
```

Out[72]:
```
<AxesSubplot:xlabel='Product line', ylabel='Total'>
```



9. Change bar color using Color attribute

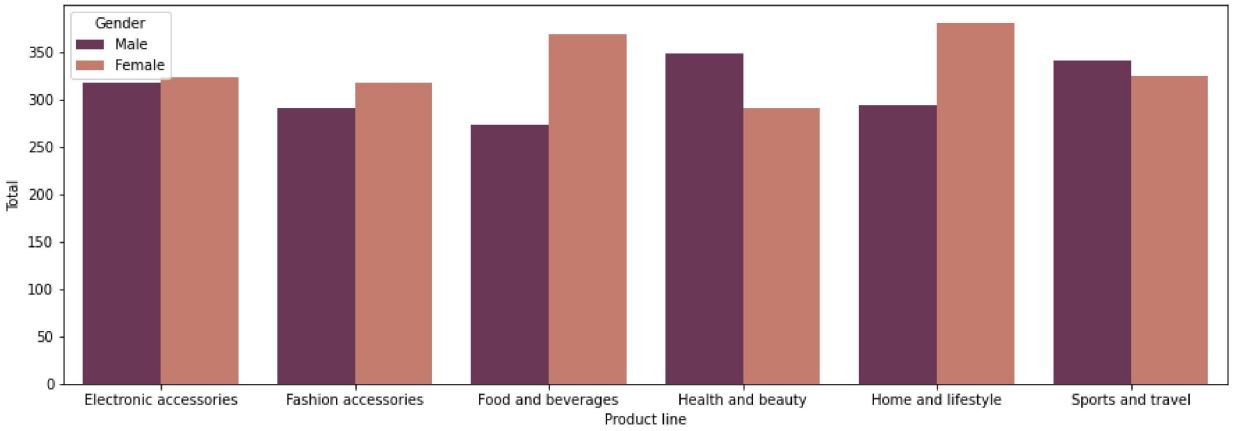
In [76]:
```
plt.figure(figsize=(15,5))
sns.barplot(
    x='Product line',
    y='Total',
    data=mart,
    order=['Electronic accessories','Fashion accessories','Food and beverages','Health
    ci=None,
    color='yellow'
    )
```

Out[76]:
```
<AxesSubplot:xlabel='Product line', ylabel='Total'>
```

10. Using Saturation parameter

In [83]:
```
plt.figure(figsize=(15,5))
sns.barplot(
    x='Product line',
    y='Total',
    hue='Gender',
    hue_order=['Male','Female'],
    data=mart,
    order=['Electronic accessories','Fashion accessories','Food and beverages','Health
    ci=None,
    palette='rocket',
    saturation=0.5
    )
```

Out[83]:
```
<AxesSubplot:xlabel='Product line', ylabel='Total'>
```

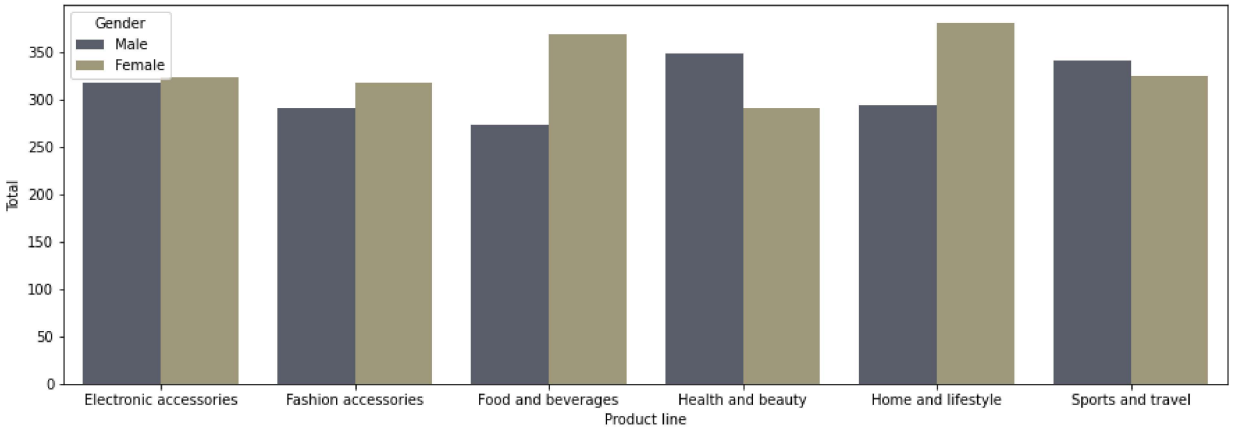


11. Change default Aggregation method using ESTIMATOR parameter

In [87]:
```
plt.figure(figsize=(15,5))
sns.barplot(
    x='Product line',
    y='Total',
    hue='Gender',
    hue_order=['Male','Female'],
    data=mart,
    order=['Electronic accessories','Fashion accessories','Food and beverages','Health
    ci=None,
    palette='cividis',
```

```
    estimator=np.mean #or sum | np.median
    )
```

Out[87]:
```
<AxesSubplot:xlabel='Product line', ylabel='Total'>
```



BOX PLOT

This provides a summary of supplied data, which includes the information like:

Mean | Media | Minimum | Maximum | 1st Quartile | 3rd Quartile | Outliers

In [89]:
```
mart.head()
```

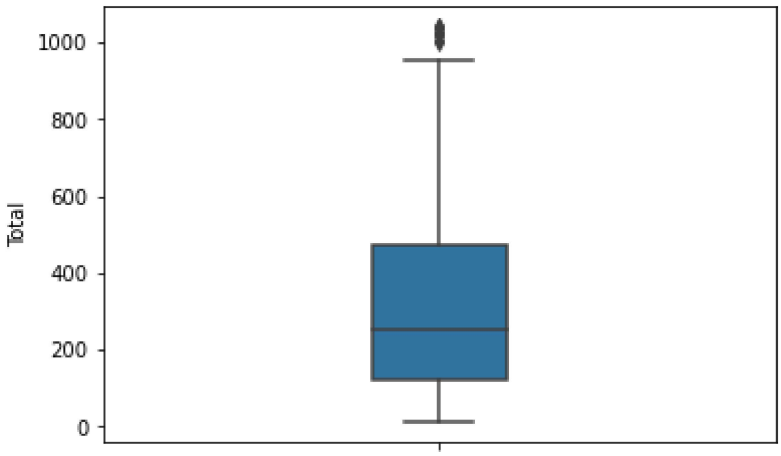
Out[89]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2

1. Create a basic Box Plot on one NUMERIC Vatiable

In [95]:
```
sns.boxplot(y='Total',data=mart, width=0.2)
```

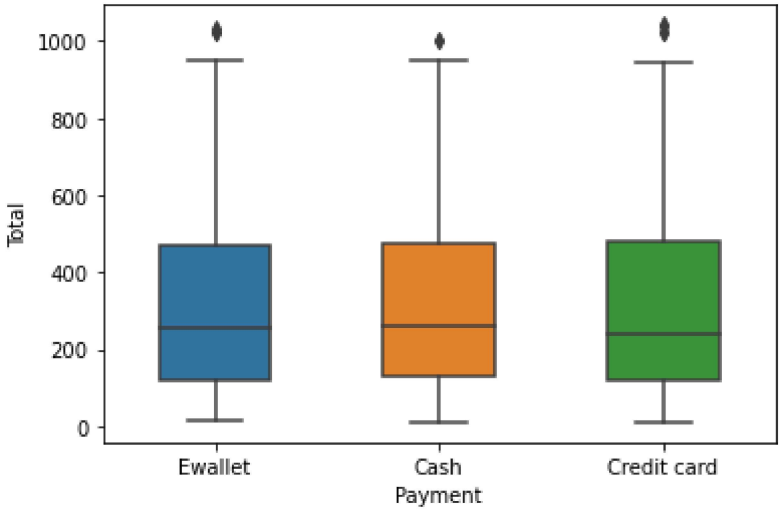
Out[95]: <AxesSubplot:ylabel='Total'>



2. Create a basic BOX plot on one numeric variable by a Categorical Variable

In [98]:
```
sns.boxplot(x='Payment',y='Total',data=mart,width=0.5)
```

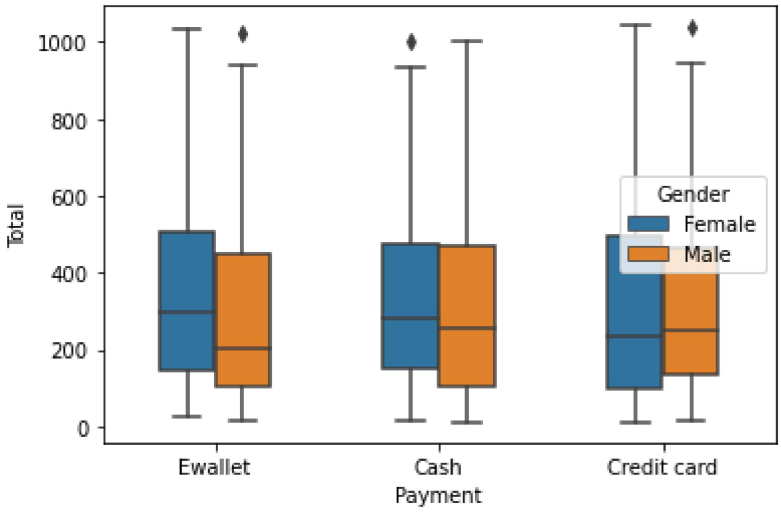
Out[98]: <AxesSubplot:xlabel='Payment', ylabel='Total'>



3. Create a basic BOX plot on one numerical variable by TWO Categorical variable using HUE attribute

In [100…
```
sns.boxplot(x='Payment',y='Total',hue='Gender',data=mart,width=0.5)
```

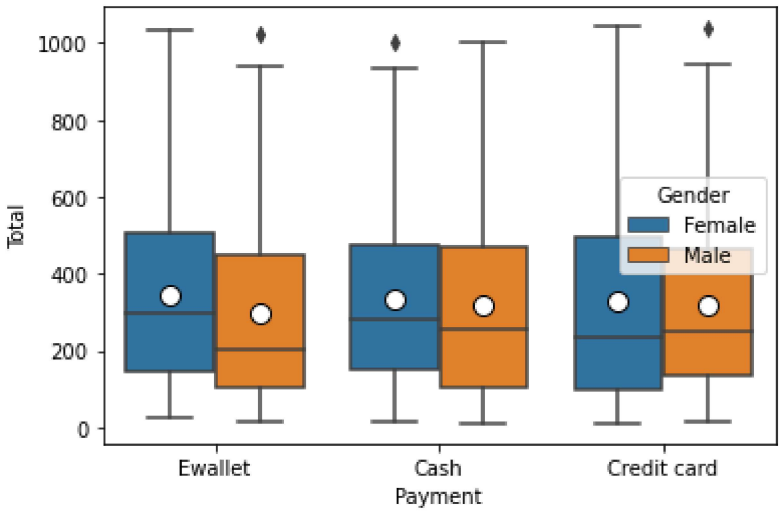
Out[100]: <AxesSubplot:xlabel='Payment', ylabel='Total'>



4. Add MEAN marker in the box plot using 'showmeans' attribute and change its style using meanprops

In [105…
```
sns.boxplot(x='Payment',
            y='Total',
            showmeans=True,
            meanprops={"marker":"o",
                       "markerfacecolor":"white",
                        "markersize":10,
                        "markeredgecolor":"black"},
            data=mart,
            hue='Gender')
```

Out[105]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```

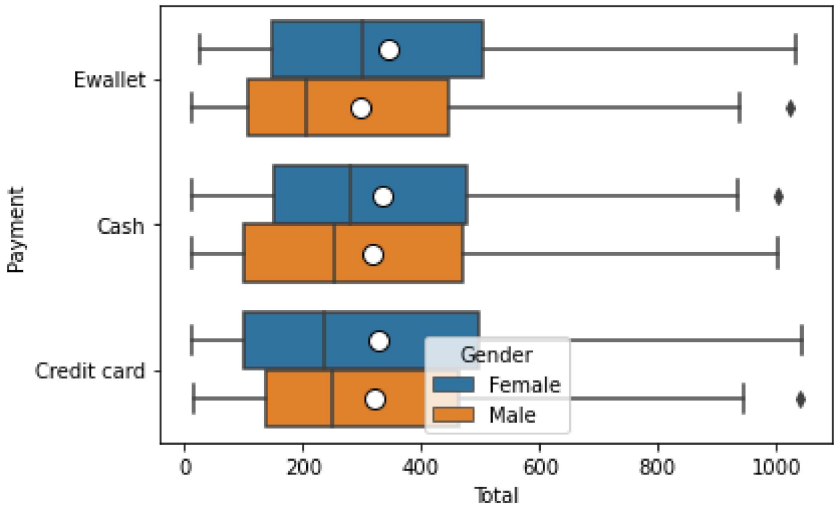


5. Make HORIZONTAL box plot

In [107…
```
sns.boxplot(x='Total',
            y='Payment',
            showmeans=True,
            meanprops={"marker":"o",
                       "markerfacecolor":"white",
                        "markersize":10,
                        "markeredgecolor":"black"},
```

```
            data=mart,
            hue='Gender')
```

Out[107]: <AxesSubplot:xlabel='Total', ylabel='Payment'>

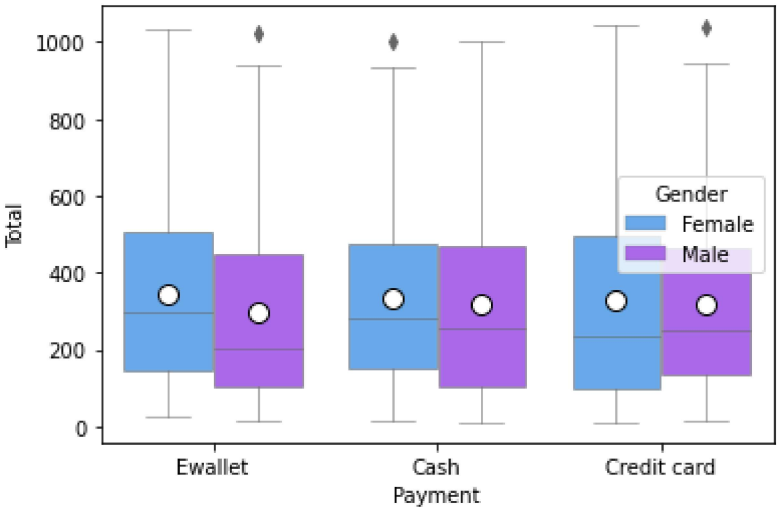


6. Change Palette, Line width etc..

In [110…

```
sns.boxplot(y='Total',
            x='Payment',
            showmeans=True,
            meanprops={"marker":"o",
                       "markerfacecolor":"white",
                        "markersize":10,
                        "markeredgecolor":"black"},
            data=mart,
            hue='Gender',
            palette='cool',
            linewidth=0.5
           )
```

Out[110]: <AxesSubplot:xlabel='Payment', ylabel='Total'>



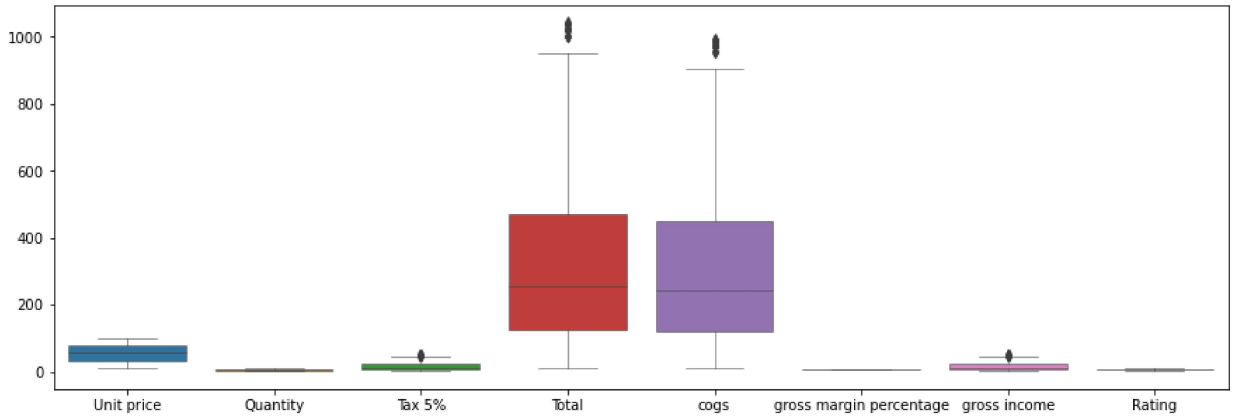
7.Create box plot for EACH OF THE NUMERIC VARIABLE in the dataframe

In [113…

```
plt.figure(figsize=(15,5))
```

```
sns.boxplot(data=mart,linewidth=0.5)
```

Out[113]: <AxesSubplot:>



VIOLIN PLOT

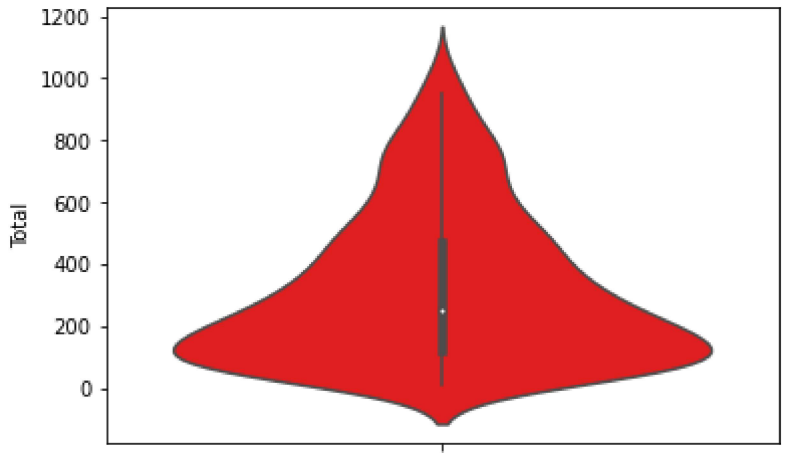
A violin plot is a method of plotting numeric data. It is similar to a box plot, with the addition of a rotated kernel density plot on each side.

Violin plots are similar to box plots, except that they also show the probability density of the data at different values, usually smoothed by a kernel density estimator. Typically a violin plot will include all the data that is in a box plot: a marker for the median of the data; a box or marker indicating the interquartile range; and possibly all sample points, if the number of samples is not too high.

1. Create a basic VIOLIN plot

In [115…
```
sns.violinplot(y='Total',data=mart,color='red')
```

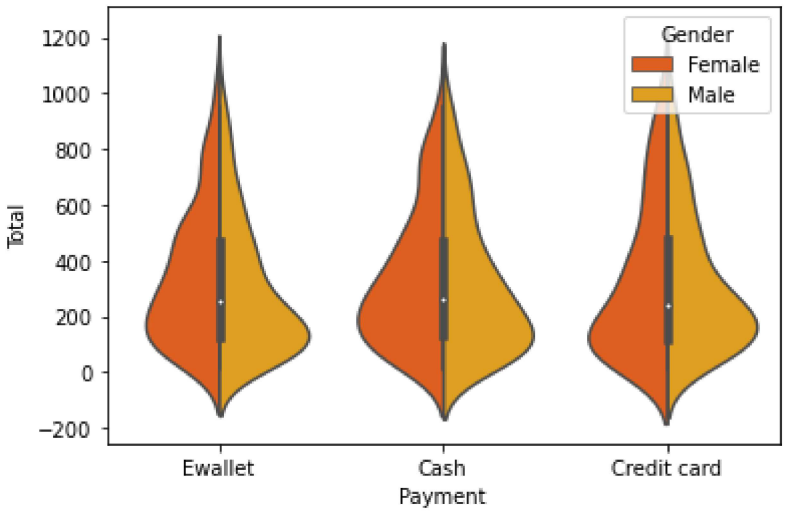
Out[115]: <AxesSubplot:ylabel='Total'>



2.Create a VIOLINPLOT on two categorical and one numeric variable and use split

In [118…
```
sns.violinplot(x='Payment',y='Total',hue='Gender',split=True,data=mart,palette='autumr
```

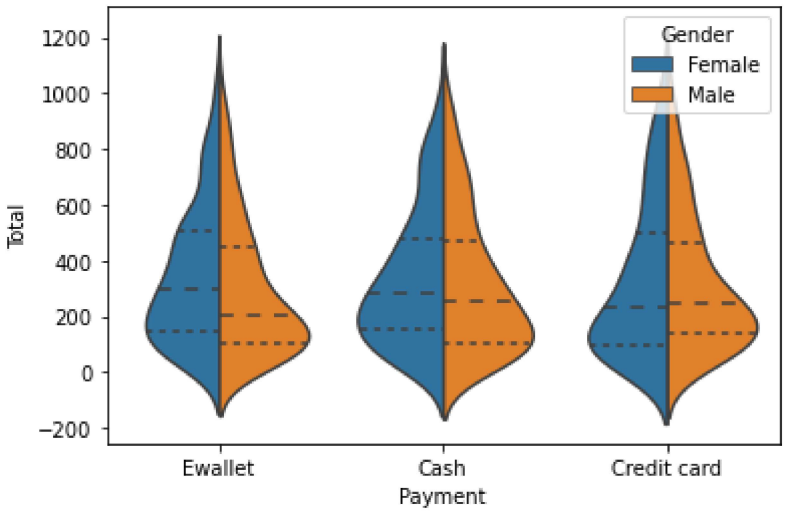
Out[118]: <AxesSubplot:xlabel='Payment', ylabel='Total'>



3. Change the box in the VIOLINPLOT to horizontal lines

In [119…
```
sns.violinplot(x='Payment',y='Total',hue='Gender',split=True,inner='quartile',data=mar
```

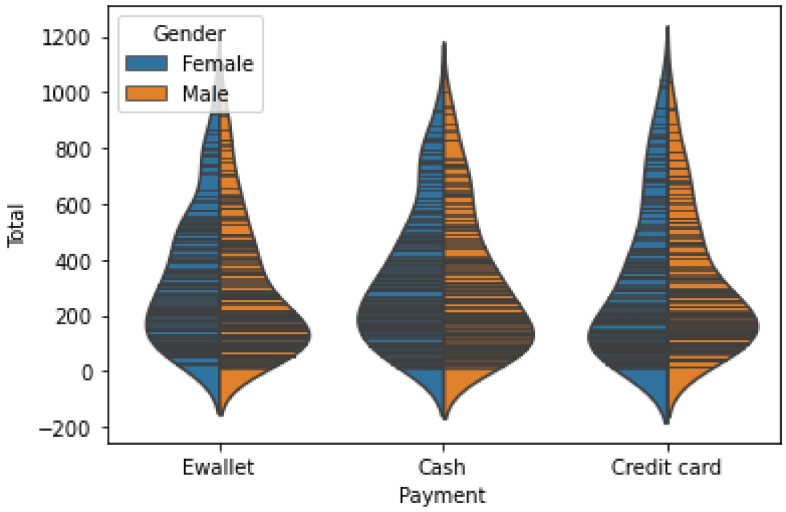
Out[119]: <AxesSubplot:xlabel='Payment', ylabel='Total'>



4. Draw line for each observation in a VIOLIN Plot

In [120…
```
sns.violinplot(x='Payment',y='Total',hue='Gender',split=True,inner='stick',data=mart)
```

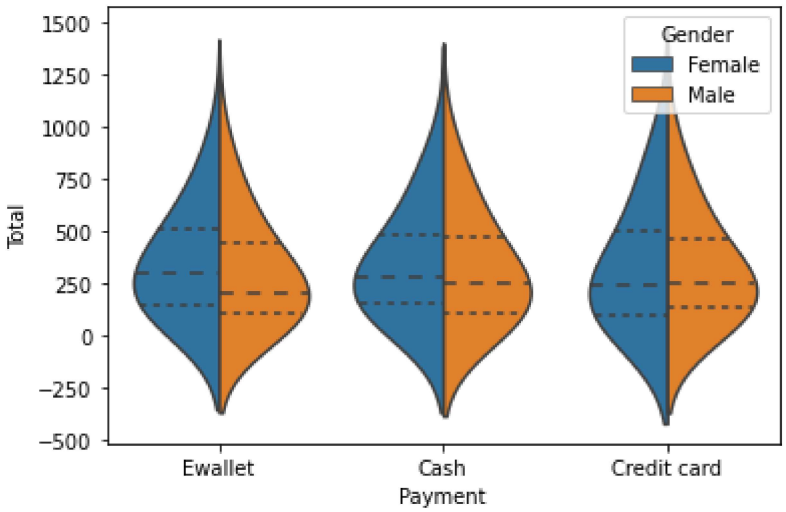
Out[120]: <AxesSubplot:xlabel='Payment', ylabel='Total'>



5. Change the amount of smoothing using bw attribute

In [126…
```
sns.violinplot(x='Payment',y='Total',hue='Gender',bw=0.8, #band width(bw)
               split=True,inner='quartile',data=mart)
```

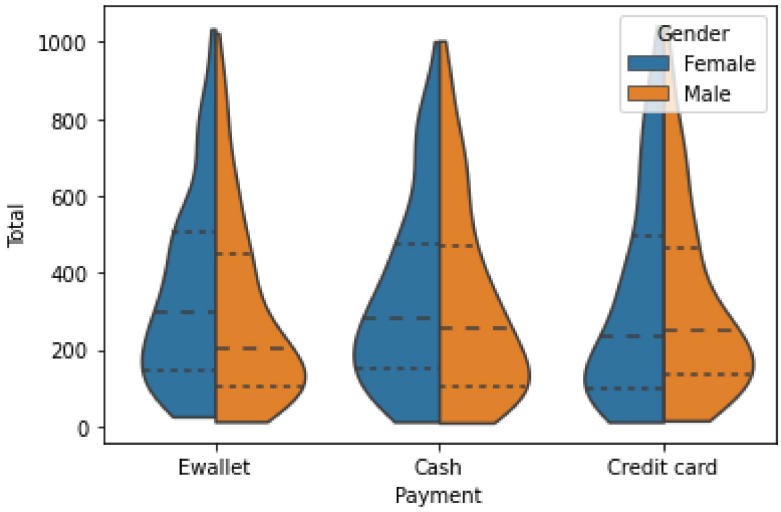
Out[126]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```



6. Cut out the extreme values

In [127…
```
sns.violinplot(x='Payment',y='Total',hue='Gender',split=True,inner='quartile',data=mar
```

Out[127]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```



STRIP PLOT

A strip plot is a graphical data an.alysis technique for summarizing a univariate data set

Use Strip Plot when there is lesser amount of data

A strip plot can be drawn on its own, but it is also a good complement to a box or violin plot in cases where you want to show "all observations" along with some representation of the underlying distribution.

In [131…
```
mart.head()
```

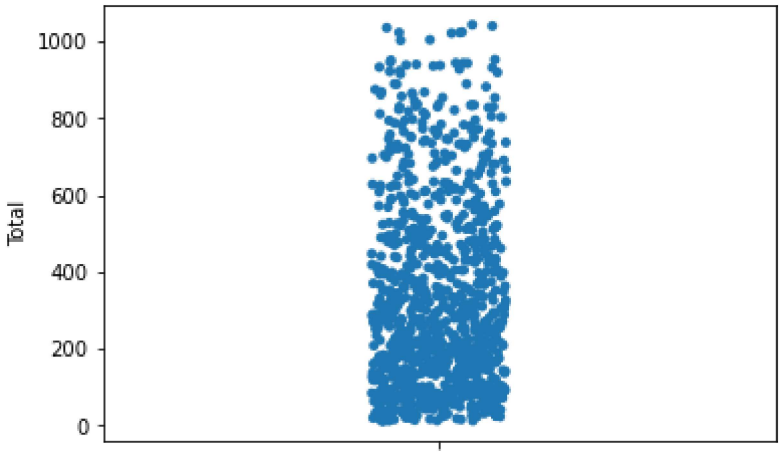
Out[131]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2

1. Create a basic STRIP PLOT

In [133…
```
sns.stripplot(data=mart, y='Total')
```

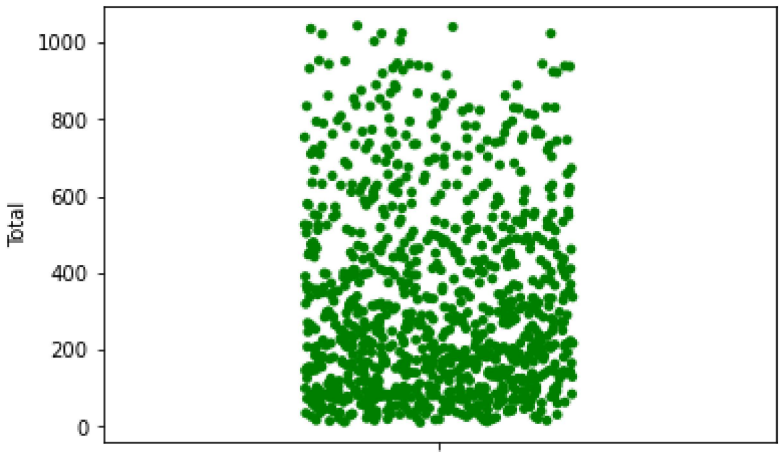
Out[133]:
```
<AxesSubplot:ylabel='Total'>
```



2. Expand markers in STRIP PLOT using jitter

In [137…
```
sns.stripplot(y='Total',data=mart,jitter=0.2,color='green')
```

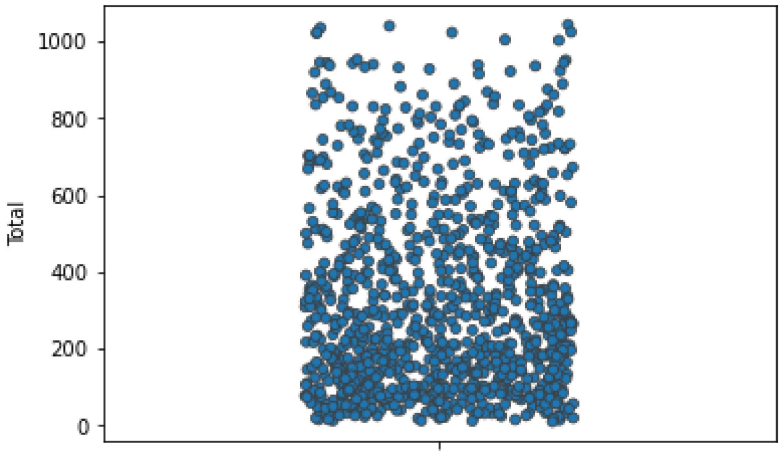
Out[137]:
```
<AxesSubplot:ylabel='Total'>
```



3. Draw line around the points using LINEWIDTH

In [138…
```
sns.stripplot(y='Total',data=mart,jitter=0.2,linewidth=0.8)
```

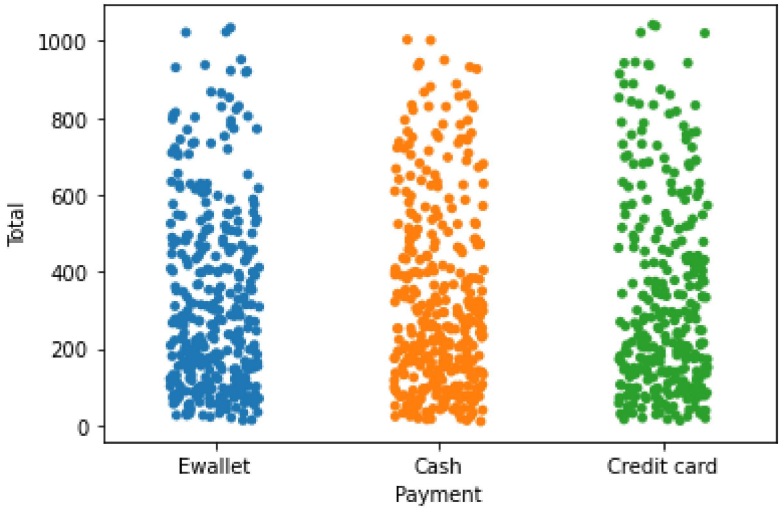
Out[138]:
```
<AxesSubplot:ylabel='Total'>
```



4. Include one categorical variable

In [140…
```
sns.stripplot(y='Total',x='Payment',data=mart,jitter=0.2)
```

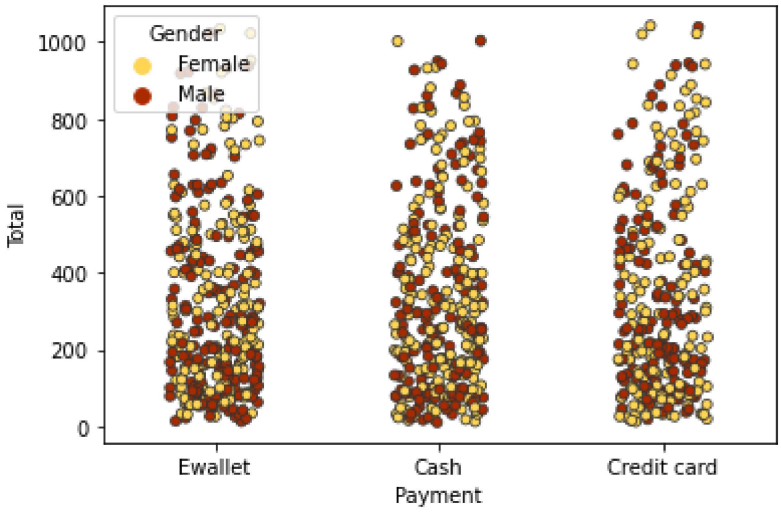
Out[140]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```



5. Include third categorical variable with hue

In [145…
```
sns.stripplot(y='Total',x='Payment',hue='Gender',data=mart,jitter=0.2,palette='afmhot_
```

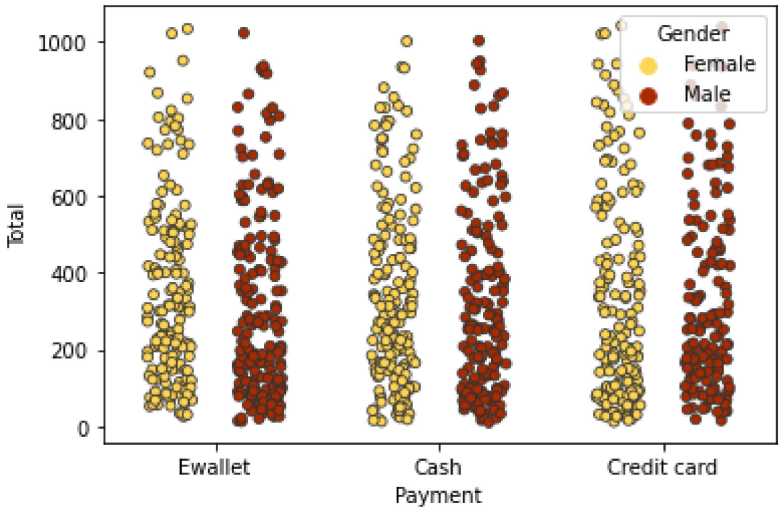
Out[145]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```



6. Seperate each level of hue using DODGE

In [146…
```
sns.stripplot(y='Total',x='Payment',hue='Gender',data=mart,jitter=0.2,palette='afmhot_
```

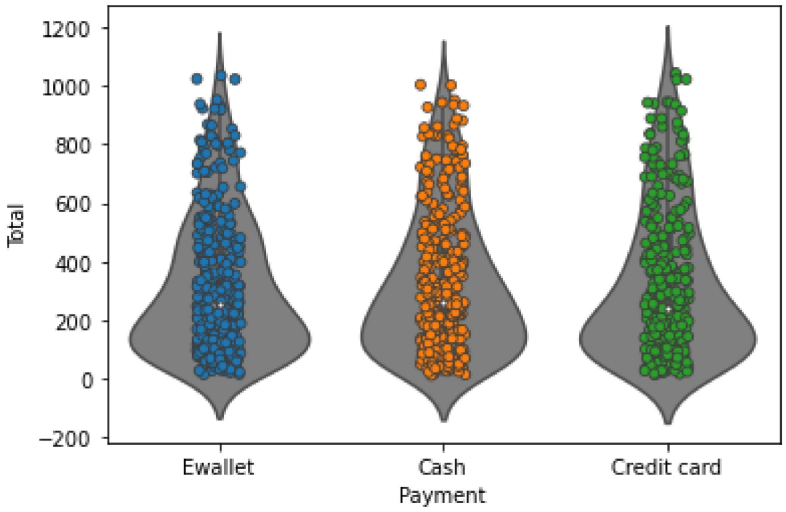
Out[146]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```



7.Draw strips on top of VIOLIN PLOT

In [152…
```
sns.stripplot(y='Total',x='Payment',jitter=0.1,data=mart,linewidth=0.8)
sns.violinplot(y='Total',x='Payment',data=mart,color='grey')
```

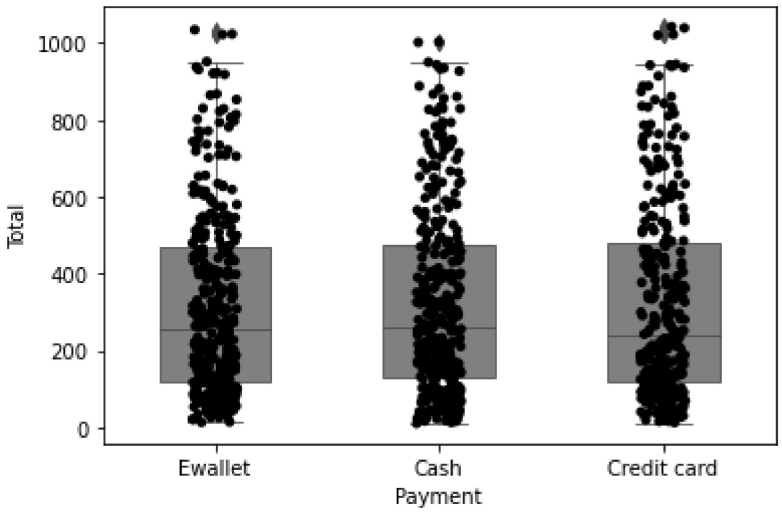
Out[152]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```



8. Draw the strips on top of BOX PLOT

In [168…
```
sns.stripplot(y='Total',x='Payment',data=mart,jitter=0.1,color='black')
sns.boxplot(y='Total',x='Payment',data=mart,linewidth=0.8,width=0.5,color='grey')
```

Out[168]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```



SWARM PLOT

Why do we draw a swarm plot?

To represent each of the data point on the plot

[1,1,1,1,2,2,2,2,3,3,3,3,4,4,4]

Why can't we do this on strip plot?

same data points overlaps in strip plots

Used for small set of data.

In [169…
```
mart.head()
```

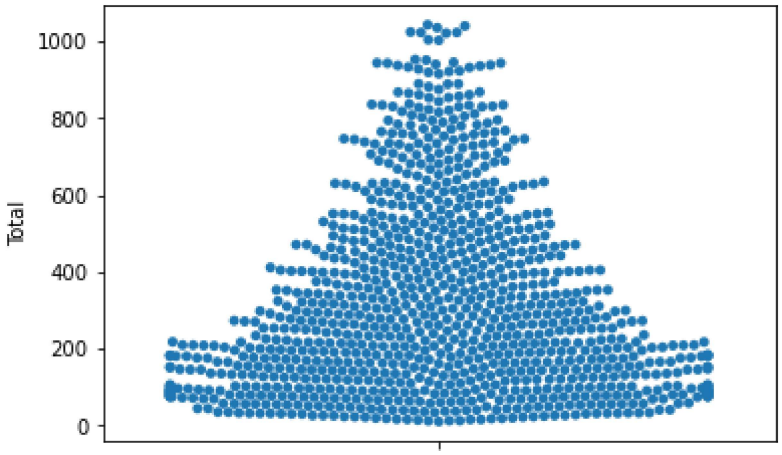
Out[169]:

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2

1. Create a basic swarm plot on Total column

In [172…
```
sns.swarmplot(data=mart,y='Total')
```

Out[172]:
```
<AxesSubplot:ylabel='Total'>
```

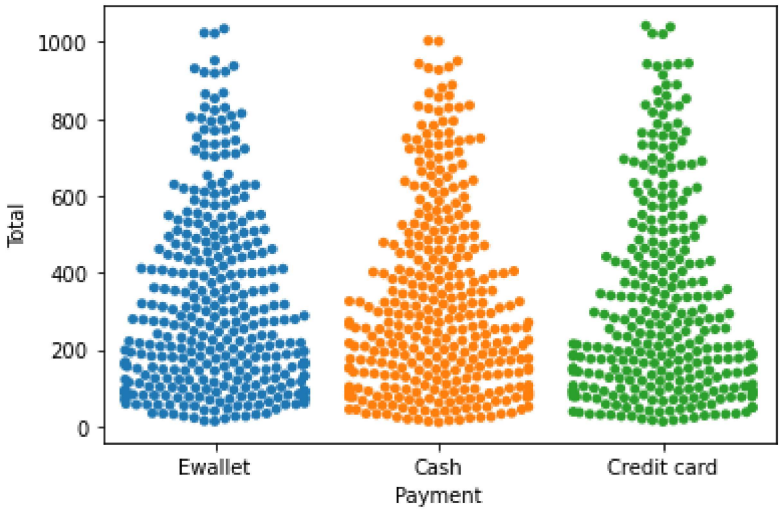


2. Create SWARM PLOT group by Categories

In [173…
```
sns.swarmplot(x='Payment',y='Total',data=mart)
```

```
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 9.6% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 7.3% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 8.4% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
```

Out[173]: <AxesSubplot:xlabel='Payment', ylabel='Total'>



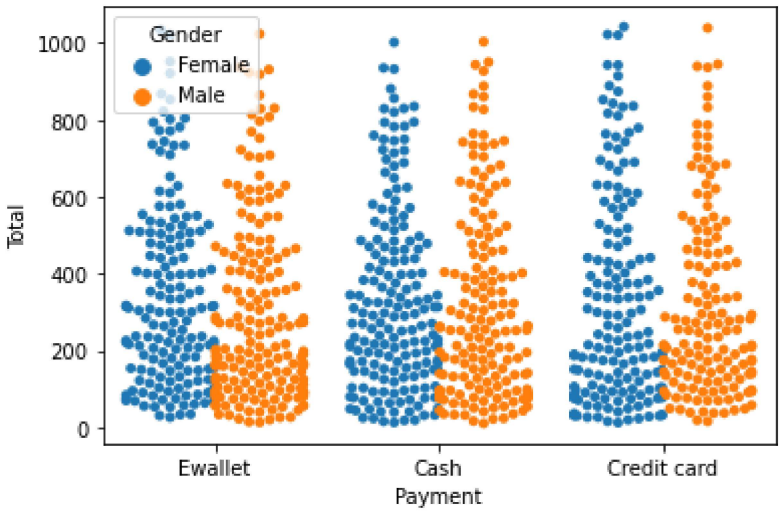
3. Showing HUES seperately on the categorical axis

In [174…

```
sns.swarmplot(x='Payment',y='Total',data=mart,hue='Gender',split=True)
```

```
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:3002: UserWarning: The `split`
parameter has been renamed to `dodge`.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 9.4% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 20.0% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 7.9% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 11.4% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 12.9% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 10.8% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
```

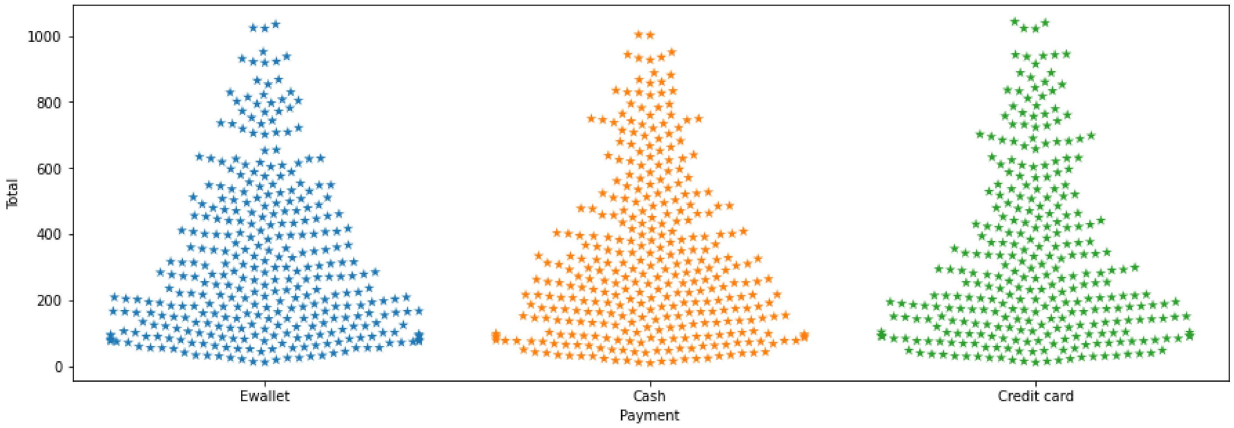
Out[174]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```



4. Styling a swarm- Change the marker,size,color,edgecolor...

In [185…
```
plt.figure(figsize=(15,5))
sns.swarmplot(x='Payment',y='Total',data=mart,marker='*',size=8)
```

Out[185]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```

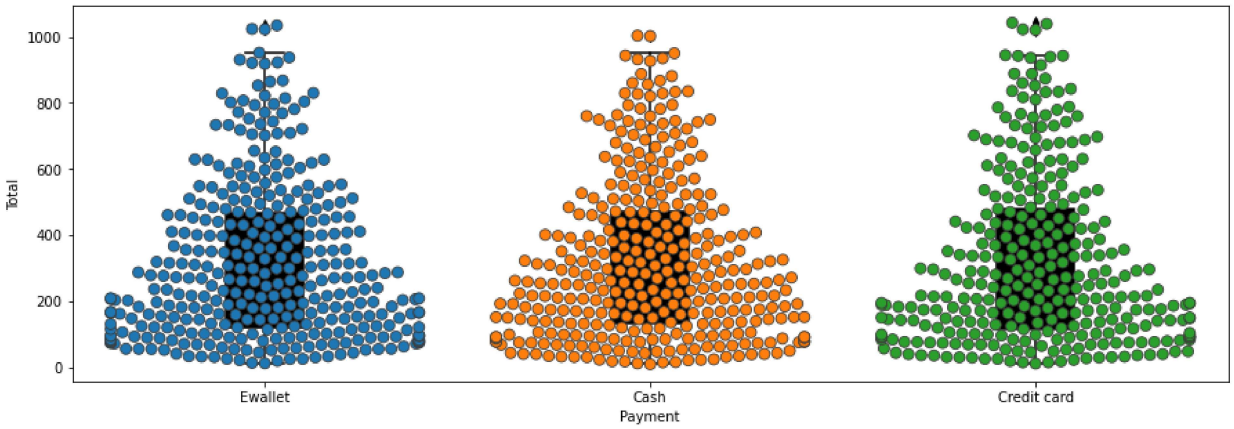


5. Overlay a swarm plot on top of box plot

In [191…
```
plt.figure(figsize=(15,5))
sns.boxplot(x='Payment',y='Total',data=mart,color='black',width=0.2)
sns.swarmplot(x='Payment',y='Total',data=mart,marker='o',size=8,linewidth=0.8)
```

```
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 7.8% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 6.4% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
```

Out[191]:
```
<AxesSubplot:xlabel='Payment', ylabel='Total'>
```

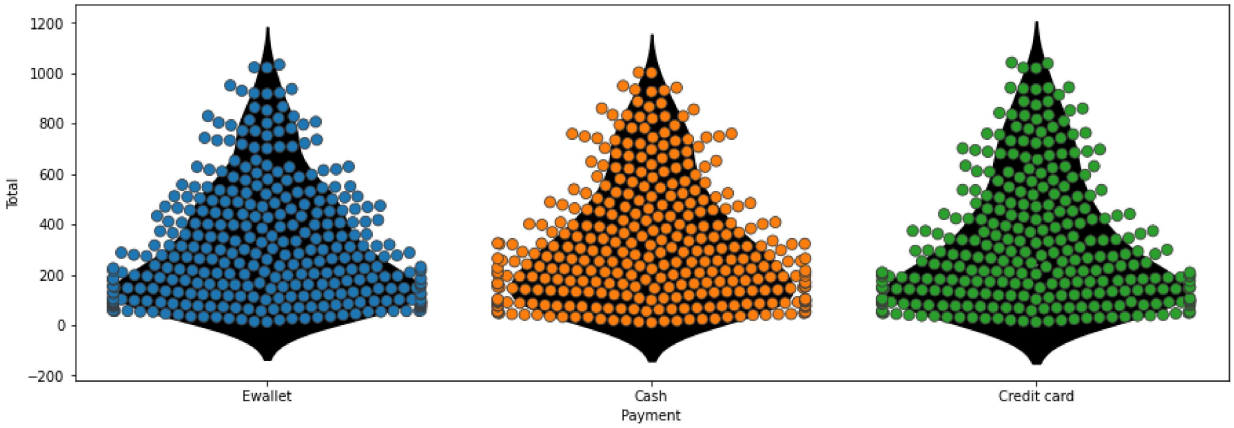


6. Overlay a swarm plot on top of violin plot

In [193…
```
plt.figure(figsize=(15,5))
sns.violinplot(x='Payment',y='Total',data=mart,color='black',width=0.8)
sns.swarmplot(x='Payment',y='Total',data=mart,marker='o',size=8,linewidth=0.8)
```

```
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 14.5% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 11.9% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
D:\Anaconda_\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 13.5% of the
points cannot be placed; you may want to decrease the size of the markers or use stri
pplot.
  warnings.warn(msg, UserWarning)
```

Out[193]: <AxesSubplot:xlabel='Payment', ylabel='Total'>



CAT PLOT

If you are working with data that involves any categorical variables like survey responses, your best tools to visualize and compare different features of your data would be categorical plots

In [194…

```
mart.head()
```

Out[194]:

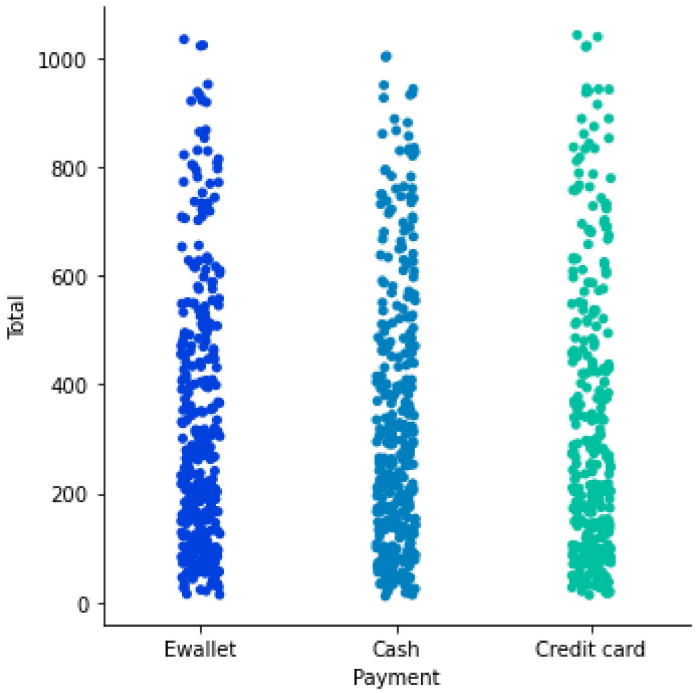
	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	1/
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2

1. Create a basic CATPLOT

By default it will create the stip plot

In [196…
```
sns.catplot(data=mart,x='Payment',y='Total',palette='winter')
```

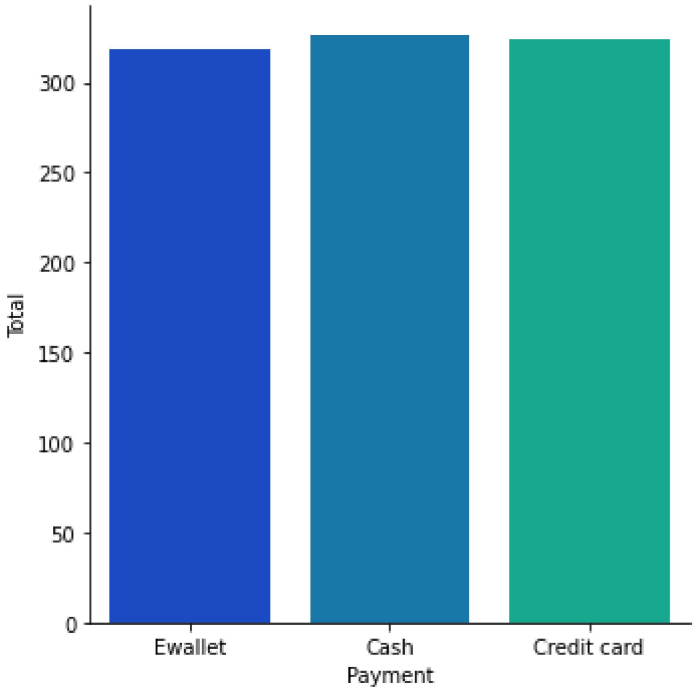
Out[196]:
```
<seaborn.axisgrid.FacetGrid at 0x1d07edb5670>
```



2. Change the type or kind of the plot in catplot using kind parameter

In [206…
```
sns.catplot(data=mart,x='Payment',y='Total',palette='winter',kind='bar',ci=None)
```

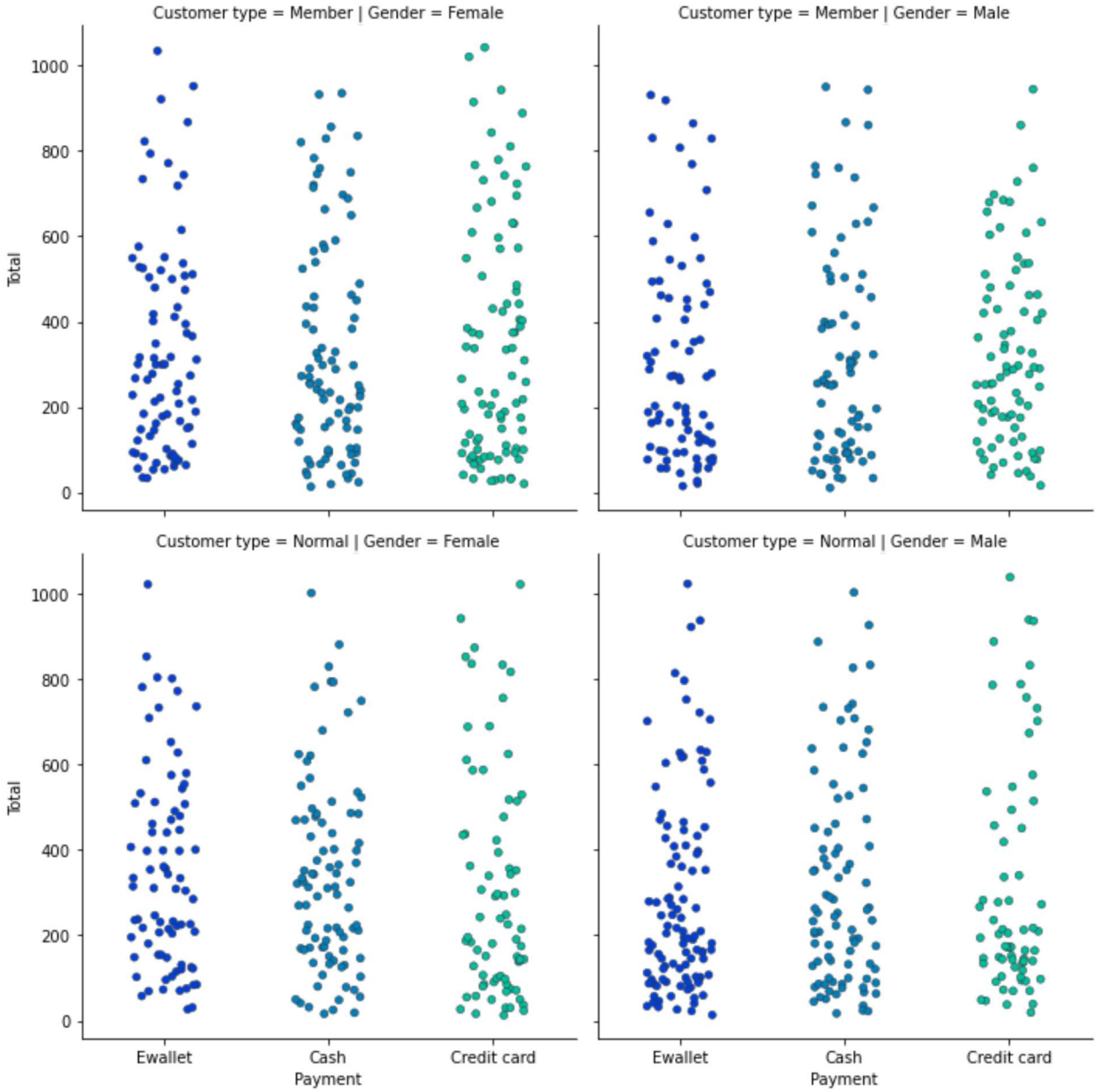
Out[206]: <seaborn.axisgrid.FacetGrid at 0x1d0019803a0>



3. Creating a separate plot for each of the category in supllied categorical data to COL agruement

In [204…
```
sns.catplot(data=mart,x='Payment',y='Total',col='Gender',row='Customer type',jitter=0.
```

Out[204]: <seaborn.axisgrid.FacetGrid at 0x1d001350070>


Customer type = Member | Gender = Female
Customer type = Member | Gender = Male
Customer type = Normal | Gender = Female
Customer type = Normal | Gender = Male
Total
0
200
400
600
800
1000
Ewallet
Cash
Credit card
Payment

In [ ]: