import libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statistics as sts
```

load dataset

In [118…
```
mart=pd.read_csv(r'C:\Users\lenovo\Downloads\supermarket_sales - Sheet1.csv')
mart=mart[['Gender','Payment','Unit price','Quantity','Total','gross income','Branch']
mart.head()
```

Out[118]:

	Gender	Payment	Unit price	Quantity	Total	gross income	Branch
0	Female	Ewallet	74.69	7	548.9715	26.1415	A
1	Female	Cash	15.28	5	80.2200	3.8200	C
2	Male	Credit card	46.33	7	340.5255	16.2155	A
3	Male	Ewallet	58.22	8	489.0480	23.2880	A
4	Male	Ewallet	86.31	7	634.3785	30.2085	A

HIST PLOT

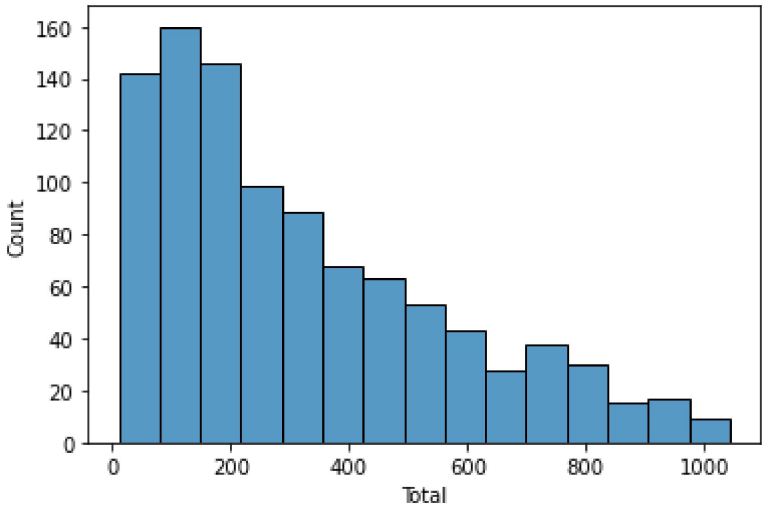
A histogram is a classical visualization tool that represents the distribution of one or more variables by counting the number of observations that fall within the discrete bins.

This function can normalize the statistic computed within each bin to estimate frequency, density or probability mass, and it can add a smooth curve obtained using a kernel density estimate, similar to # kdeplot()

1. Creating a basic HISTOGRAM on horizontal/vertical axis

In [4]:
```
sns.histplot(x='Total',data=mart)
```

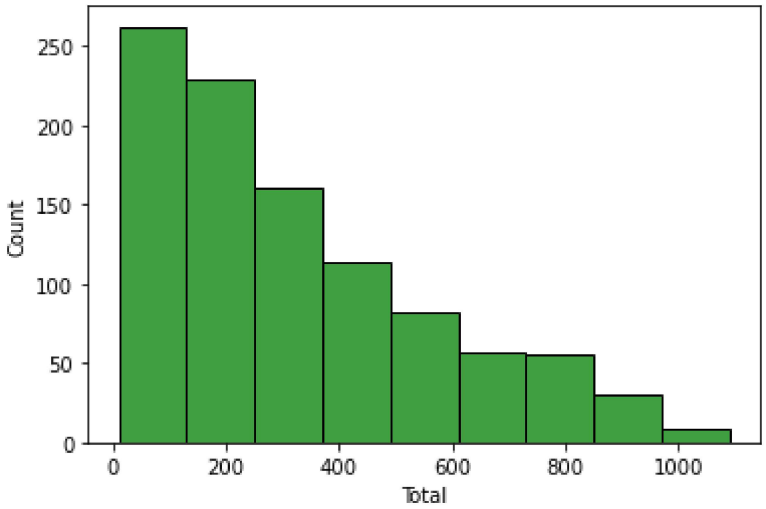
Out[4]:
```
<AxesSubplot:xlabel='Total', ylabel='Count'>
```



2. Change the width of Bins in a HISTOGRAM

In [8]:
```
sns.histplot(x='Total',data=mart,binwidth=120,color='green')
```

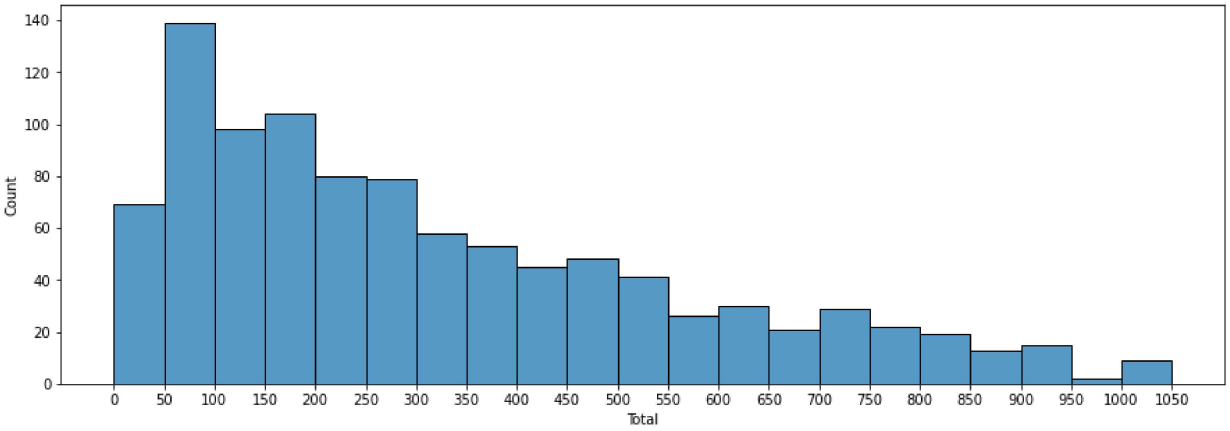
Out[8]:
```
<AxesSubplot:xlabel='Total', ylabel='Count'>
```



3. Changing number of Bins and Interval on X-Axis in a HISTOGRAM PLOT

In [15]:
```
plt.figure(figsize=(15,5))
sns.histplot(x='Total',data=mart,bins=np.arange(0,1100,50))
plt.xticks(np.arange(0,1100,50))
```

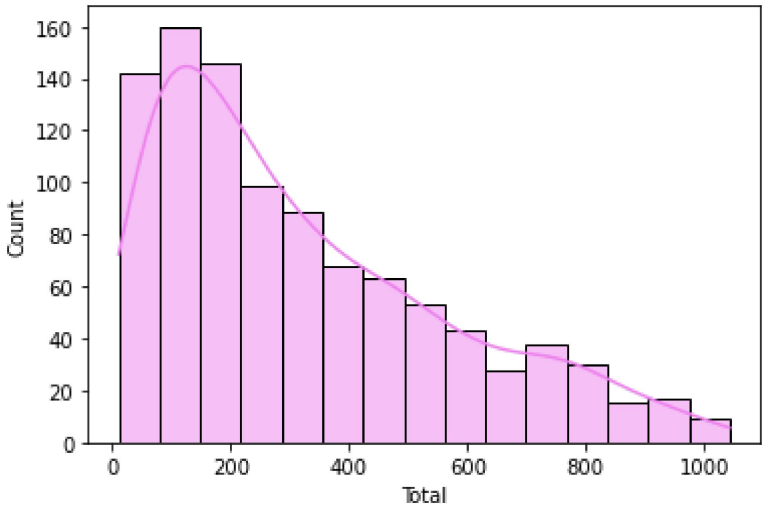
Out[15]:
```
([<matplotlib.axis.XTick at 0x1eb320dc2b0>,
  <matplotlib.axis.XTick at 0x1eb320dc280>,
  <matplotlib.axis.XTick at 0x1eb320f70d0>,
  <matplotlib.axis.XTick at 0x1eb32117b20>,
  <matplotlib.axis.XTick at 0x1eb3211d2b0>,
  <matplotlib.axis.XTick at 0x1eb3211da00>,
  <matplotlib.axis.XTick at 0x1eb32122190>,
  <matplotlib.axis.XTick at 0x1eb3211d670>,
  <matplotlib.axis.XTick at 0x1eb321173d0>,
  <matplotlib.axis.XTick at 0x1eb31ef4f40>,
  <matplotlib.axis.XTick at 0x1eb320ed880>,
  <matplotlib.axis.XTick at 0x1eb32132cd0>,
  <matplotlib.axis.XTick at 0x1eb3213b460>,
  <matplotlib.axis.XTick at 0x1eb3213bbb0>,
  <matplotlib.axis.XTick at 0x1eb321403d0>,
  <matplotlib.axis.XTick at 0x1eb3213bc10>,
  <matplotlib.axis.XTick at 0x1eb320f7070>,
  <matplotlib.axis.XTick at 0x1eb321221f0>,
  <matplotlib.axis.XTick at 0x1eb32140a30>,
  <matplotlib.axis.XTick at 0x1eb3214a1c0>,
  <matplotlib.axis.XTick at 0x1eb3214a910>,
  <matplotlib.axis.XTick at 0x1eb3214d0a0>],
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```



4. Including a KDE in a HISTOGRAM PLOT

In [17]:
```
sns.histplot(x='Total',data=mart,kde=True,color='violet')
```

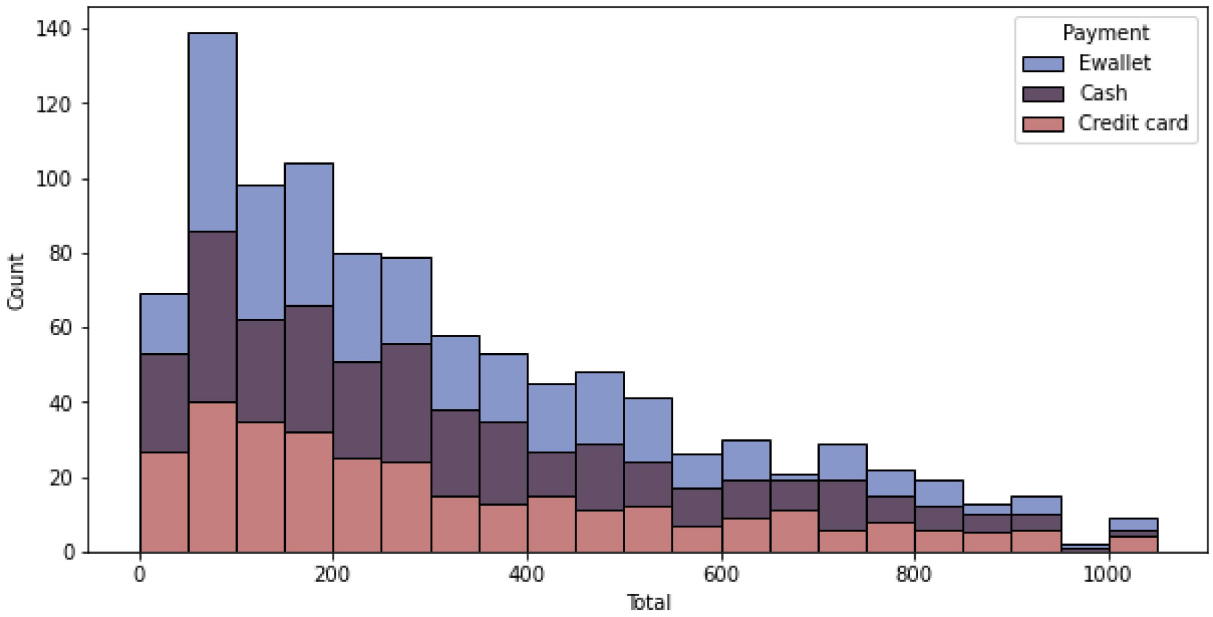
Out[17]:
```
<AxesSubplot:xlabel='Total', ylabel='Count'>
```



5. Creating a stack plot using seaborn| Stack the HISTOGRAM PLOT on a categorical variable in hue

In [28]:
```
plt.figure(figsize=(10,5))
sns.histplot(data=mart,x='Total',bins=np.arange(0,1100,50),hue='Payment',multiple='sta
```

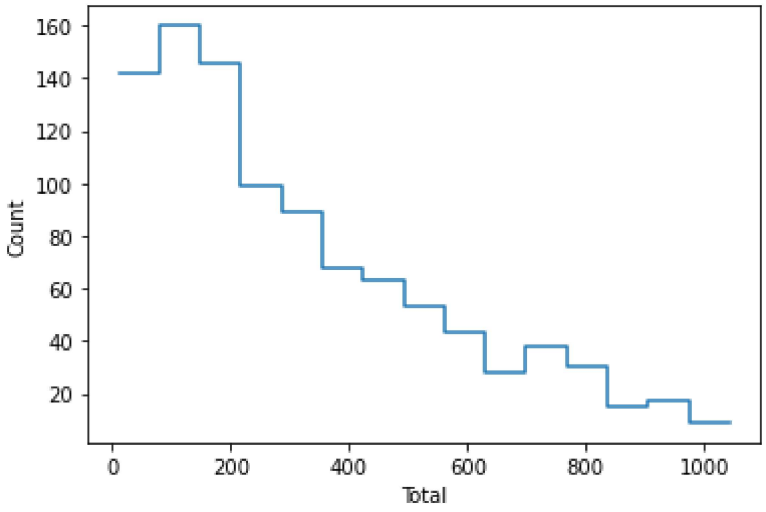
Out[28]:
```
<AxesSubplot:xlabel='Total', ylabel='Count'>
```



6. Creating a step plot using HIST PLOT in seaborn and making it no fill| creating a poly plot

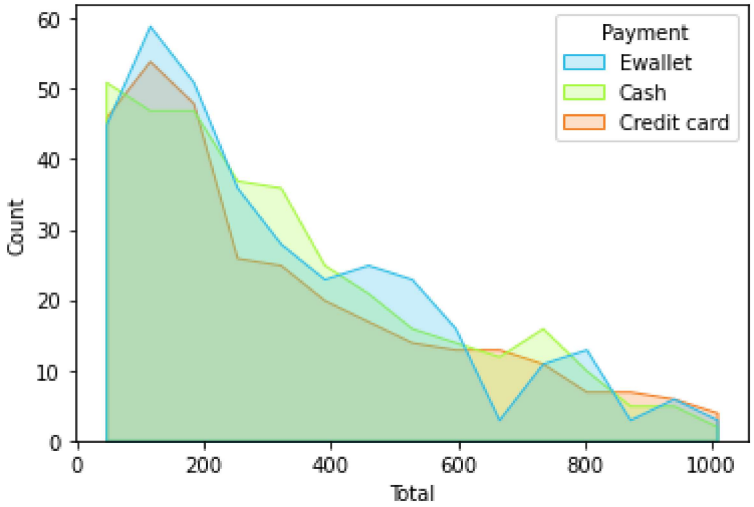
In [30]:
```
sns.histplot(x='Total',data=mart,element='step',fill=False)
```

Out[30]:
```
<AxesSubplot:xlabel='Total', ylabel='Count'>
```



In [35]:
```
sns.histplot(x='Total',data=mart,element='poly',hue='Payment',palette='turbo')
```

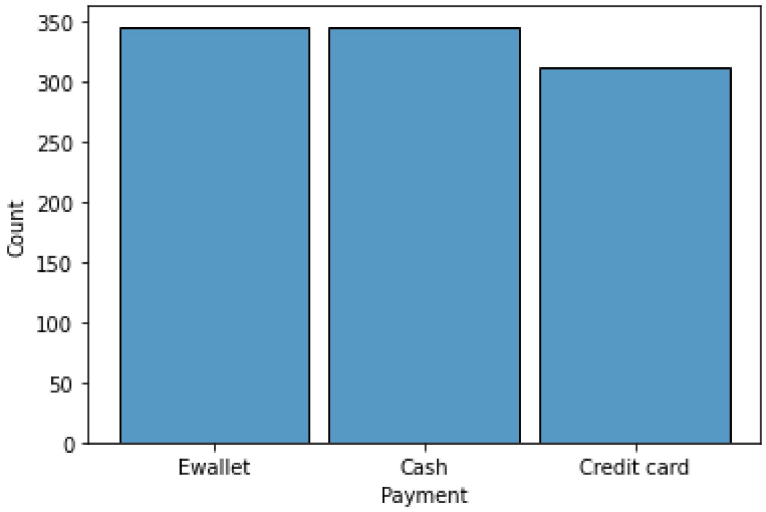
Out[35]:
```
<AxesSubplot:xlabel='Total', ylabel='Count'>
```



7. Creating a HISTOGRAM for a categorical variable and shrink it

In [40]:
```
sns.histplot(x='Payment',data=mart,shrink=0.9)
```

Out[40]:
```
<AxesSubplot:xlabel='Payment', ylabel='Count'>
```

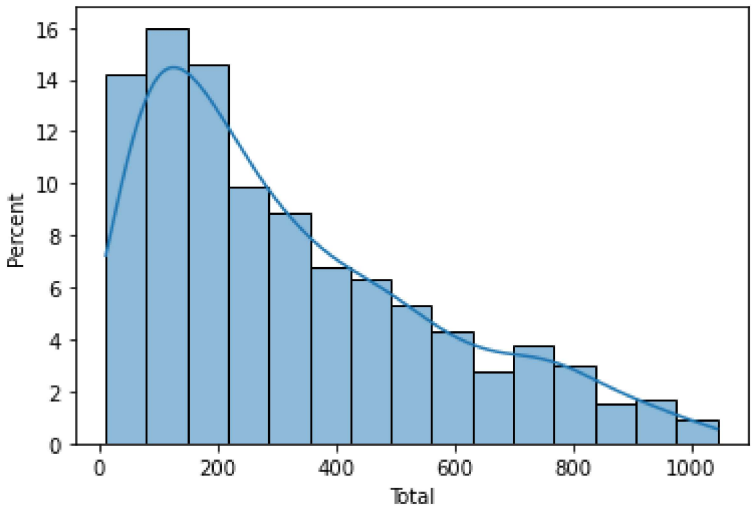


8. Using different stats or calculation or aggregation in a HIST PLOT

In [47]:
```
sns.histplot(x='Total',data=mart,stat='percent',kde=True)

#stat can take values ['count', 'frequency', 'density', 'probability', 'proportion',
```

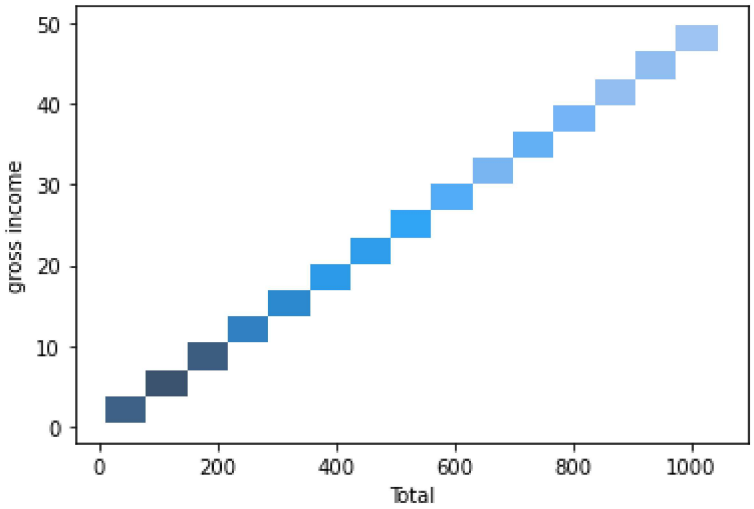
Out[47]:
```
<AxesSubplot:xlabel='Total', ylabel='Percent'>
```



9. Creating a bivariate HISTOGRAM using HISTPLOT

In [48]:
```
sns.histplot(x='Total',y='gross income',data=mart)
```

Out[48]:
```
<AxesSubplot:xlabel='Total', ylabel='gross income'>
```



KDE PLOT

A kernel density estimate (KDE) plot is a method for visualizing the distribution of observations in a dataset, analagous to a histogram. KDE represents the data using a continuous probability density curve in one or more dimensions.

In [49]:
```
mart.head()
```

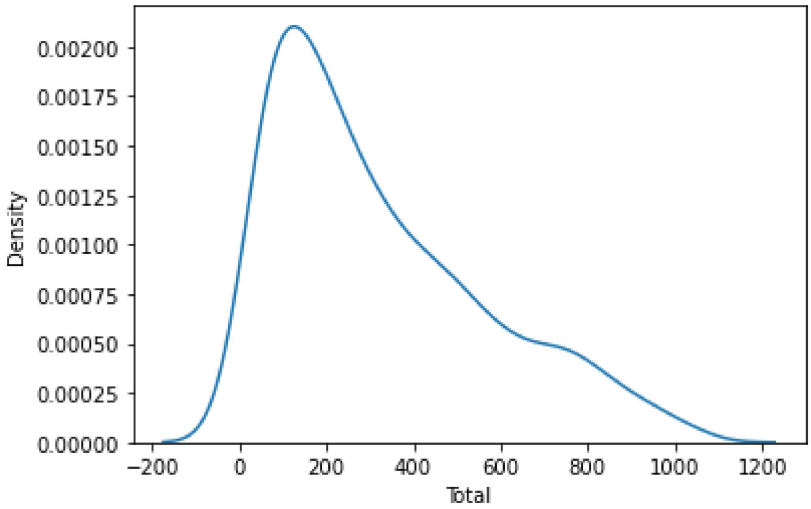
Out[49]:

	Gender	Payment	Unit price	Quantity	Total	gross income
0	Female	Ewallet	74.69	7	548.9715	26.1415
1	Female	Cash	15.28	5	80.2200	3.8200
2	Male	Credit card	46.33	7	340.5255	16.2155
3	Male	Ewallet	58.22	8	489.0480	23.2880
4	Male	Ewallet	86.31	7	634.3785	30.2085

1. Creating a basic KDE Plot

In [50]:
```
sns.kdeplot(x='Total',data=mart)
```

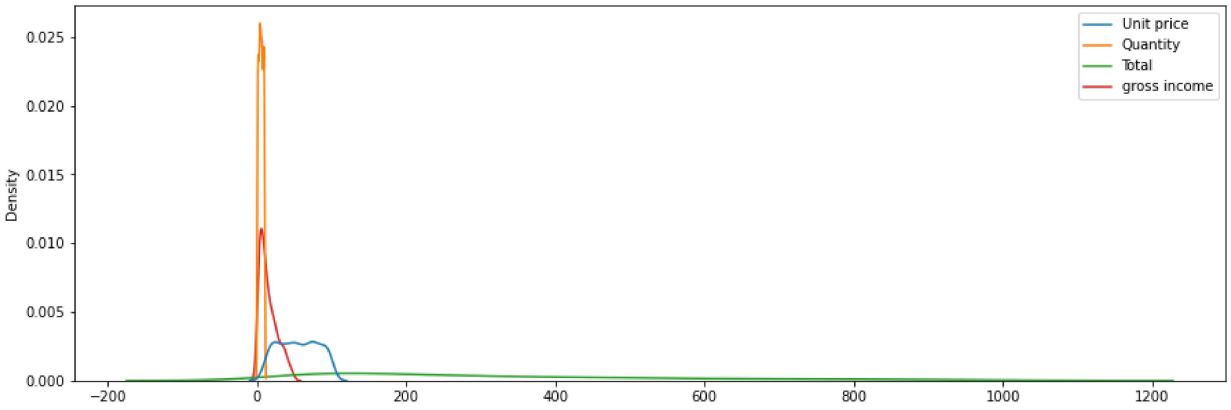
Out[50]:
```
<AxesSubplot:xlabel='Total', ylabel='Density'>
```



2. Creating a KDE Plot for all the numeric variables in a dataframe

In [52]:
```
plt.figure(figsize=(15,5))
sns.kdeplot(data=mart)
```

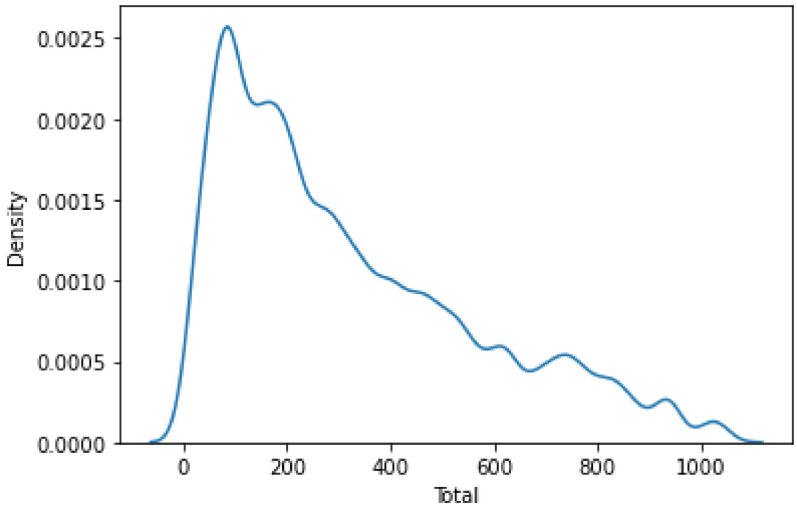
Out[52]:
```
<AxesSubplot:ylabel='Density'>
```



3. Adjusting the smoothness using bw_adjust arguement

In [57]:
```
sns.kdeplot(x='Total',data=mart,bw_adjust=0.4)
```

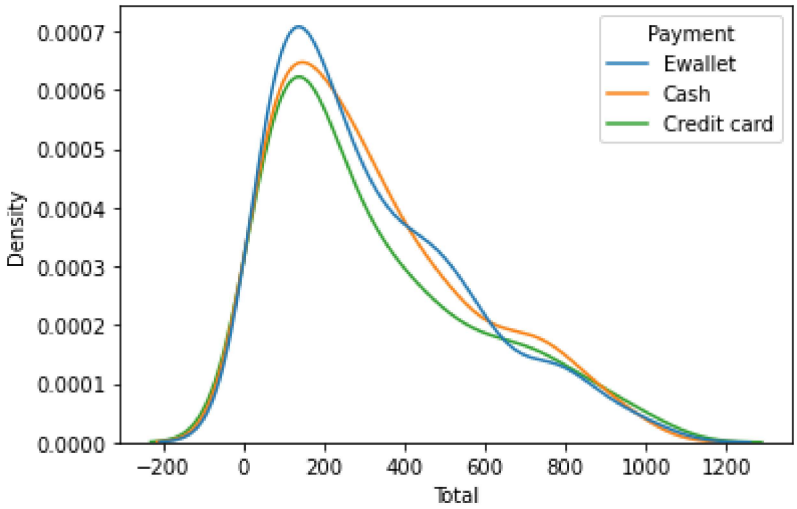
Out[57]:
```
<AxesSubplot:xlabel='Total', ylabel='Density'>
```



4. Grouping on a categorical variable with HUE in KDE Plot

In [58]:
```
sns.kdeplot(x='Total',data=mart,hue='Payment')
```

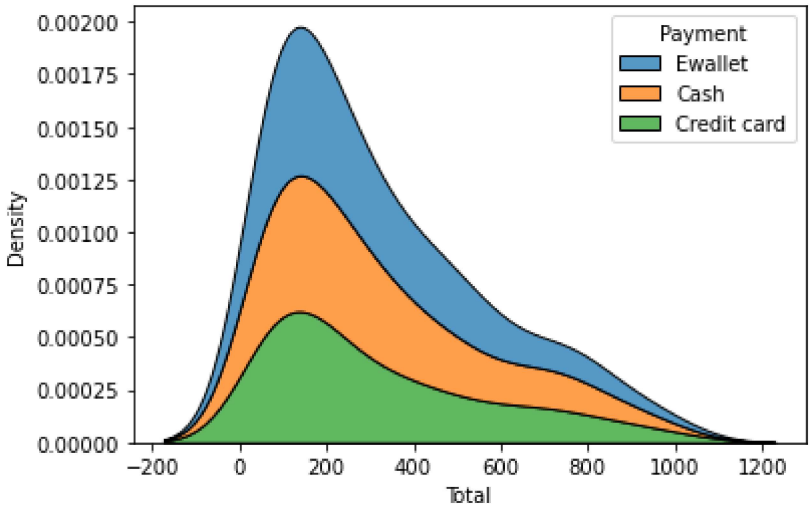
Out[58]:
```
<AxesSubplot:xlabel='Total', ylabel='Density'>
```



5. Creating a stacked KDE Plot

In [59]:
```
sns.kdeplot(x='Total',data=mart,hue='Payment',multiple='stack')
```

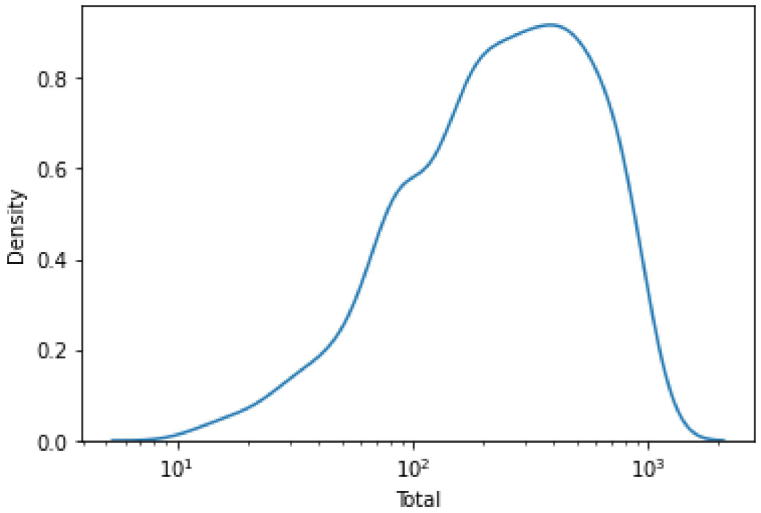
Out[59]:
```
<AxesSubplot:xlabel='Total', ylabel='Density'>
```



6. Using log scalling in the KDE plot

In [60]:
```
sns.kdeplot(x='Total',data=mart,log_scale=True)
```

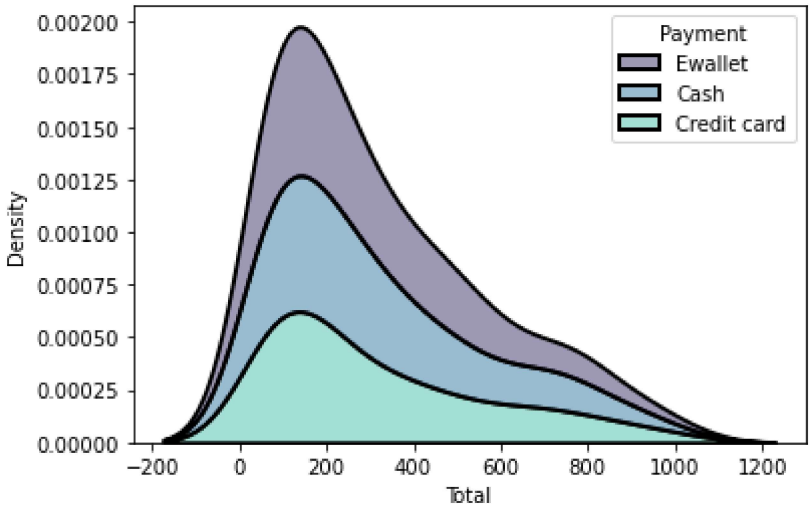
Out[60]:
```
<AxesSubplot:xlabel='Total', ylabel='Density'>
```



7. Styling a KDE plot by changing linewidth, color palettes, alpha etc...

In [66]:
```
sns.kdeplot(x='Total',data=mart,hue='Payment',multiple='stack',linewidth=2,palette='ma
```

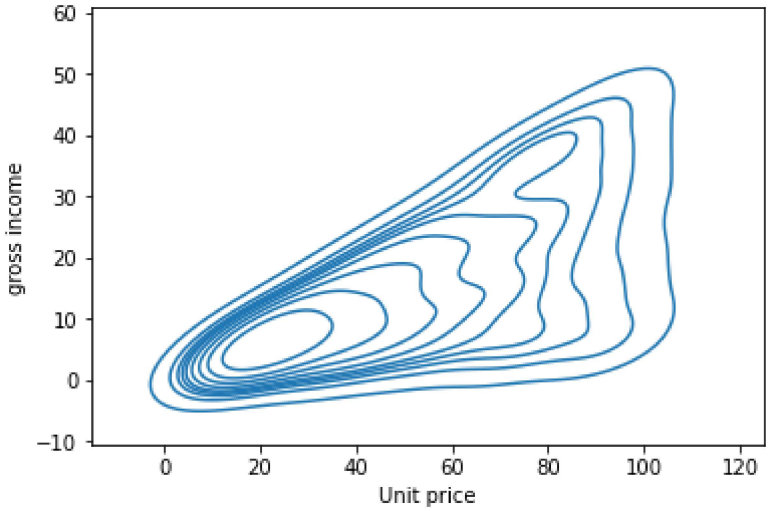
Out[66]:
```
<AxesSubplot:xlabel='Total', ylabel='Density'>
```



8. Creating a bivariate KDE Plot

In [67]:
```
sns.kdeplot(x='Unit price',y='gross income',data=mart)
```

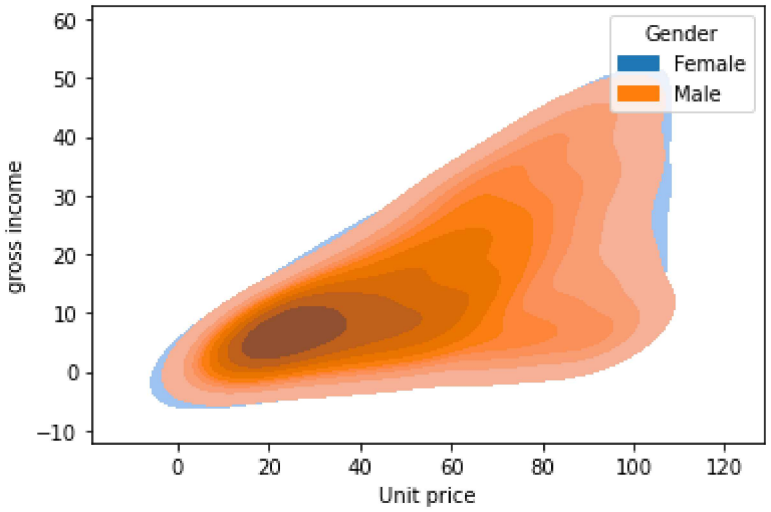
Out[67]:
```
<AxesSubplot:xlabel='Unit price', ylabel='gross income'>
```



9. Grouping a bivariate KDE Plot basis on a categorical variable and changing the counters

In [69]:
```
sns.kdeplot(x='Unit price',y='gross income',data=mart,hue='Gender',fill=True)
```

Out[69]:
```
<AxesSubplot:xlabel='Unit price', ylabel='gross income'>
```



RUG PLOT

Plot marginal distributions by drawing ticks along the x and y axes

This function is inteded to complement other plots showing the location of individual observations in an unobstrusive way

In [70]:
```
mart.head()
```

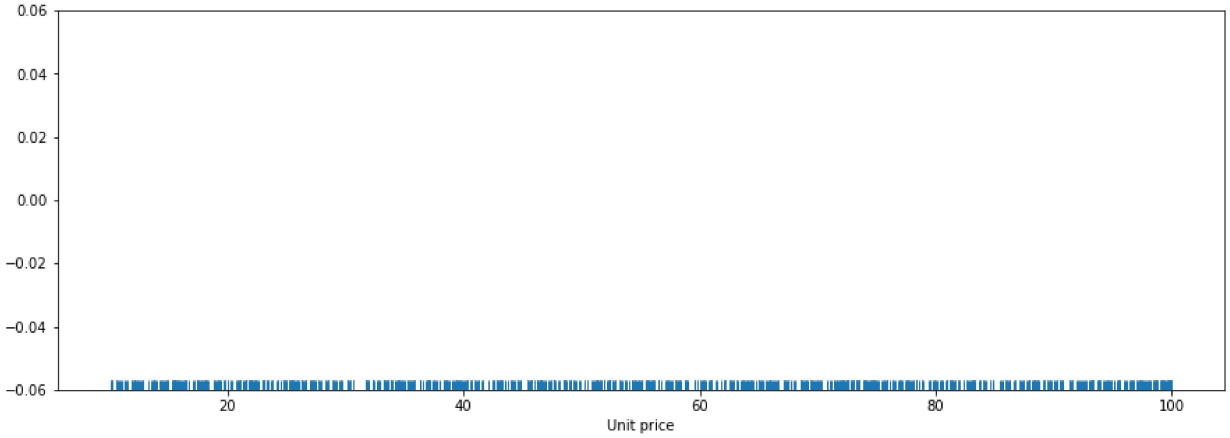
Out[70]:

	Gender	Payment	Unit price	Quantity	Total	gross income
0	Female	Ewallet	74.69	7	548.9715	26.1415
1	Female	Cash	15.28	5	80.2200	3.8200
2	Male	Credit card	46.33	7	340.5255	16.2155
3	Male	Ewallet	58.22	8	489.0480	23.2880
4	Male	Ewallet	86.31	7	634.3785	30.2085

1. Creating a basic RUG Plot using one numeric variable

In [71]:
```
plt.figure(figsize=(15,5))
sns.rugplot(data=mart,x='Unit price')
```

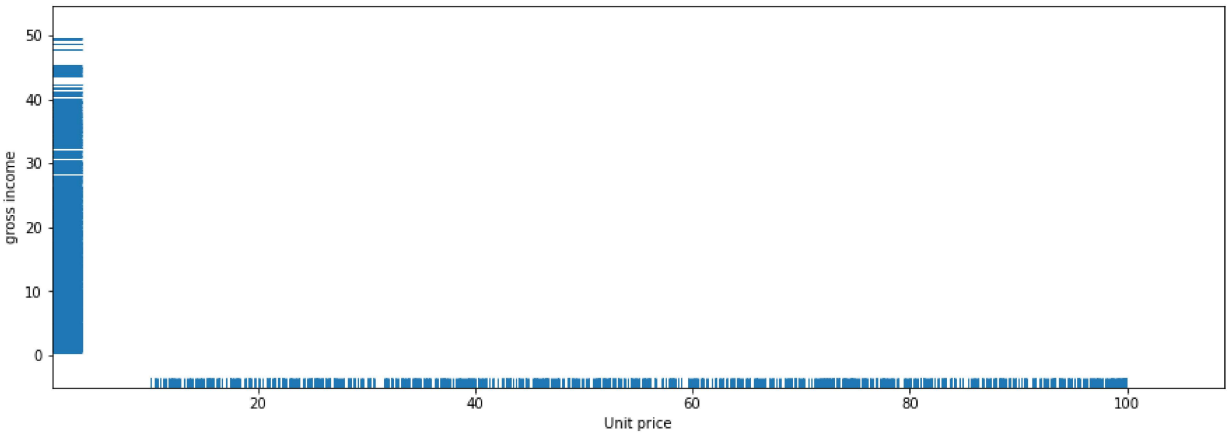
Out[71]:
```
<AxesSubplot:xlabel='Unit price'>
```



2. Creating a basic RUG Plot using two numeric variable

In [72]:
```
plt.figure(figsize=(15,5))
sns.rugplot(x='Unit price',y='gross income',data=mart)
```

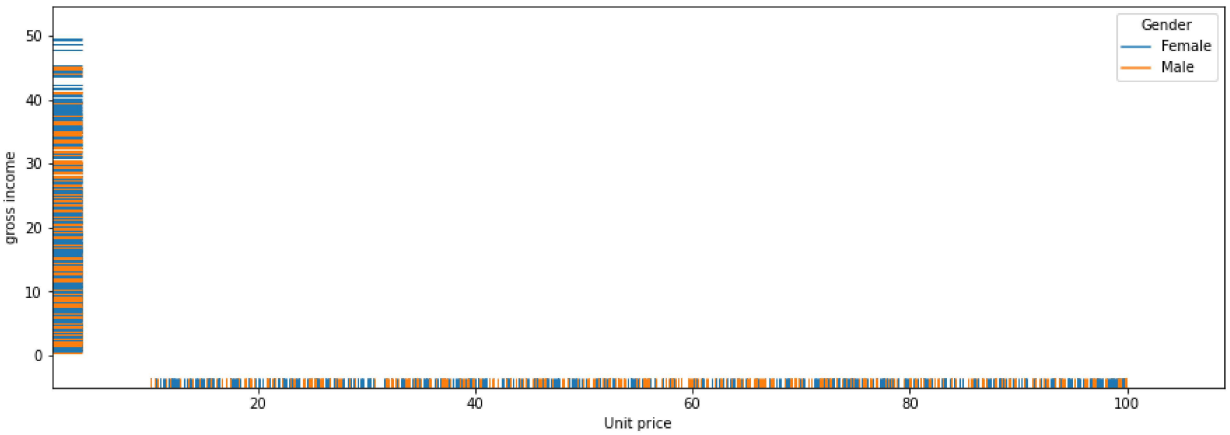
Out[72]:
```
<AxesSubplot:xlabel='Unit price', ylabel='gross income'>
```



3. Grouping the RUG Plot ticks by categories/categorical variable using Hue

In [73]:
```
plt.figure(figsize=(15,5))
sns.rugplot(x='Unit price',y='gross income',data=mart,hue='Gender')
```

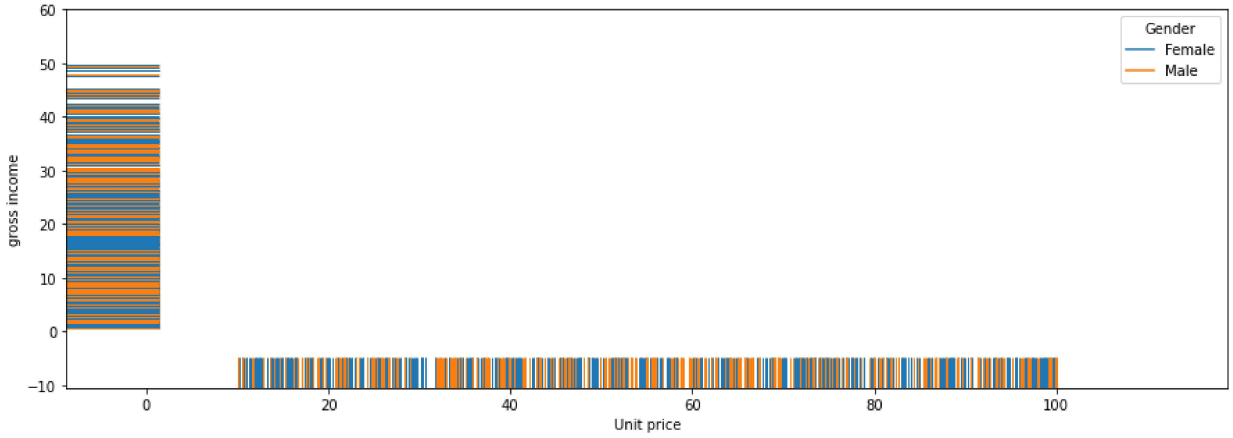
Out[73]:
```
<AxesSubplot:xlabel='Unit price', ylabel='gross income'>
```



4. Adjusting the height of RUGs/Ticks in a RUG Plot

In [75]:
```
plt.figure(figsize=(15,5))
sns.rugplot(x='Unit price',y='gross income',data=mart,hue='Gender',height=0.08)
```

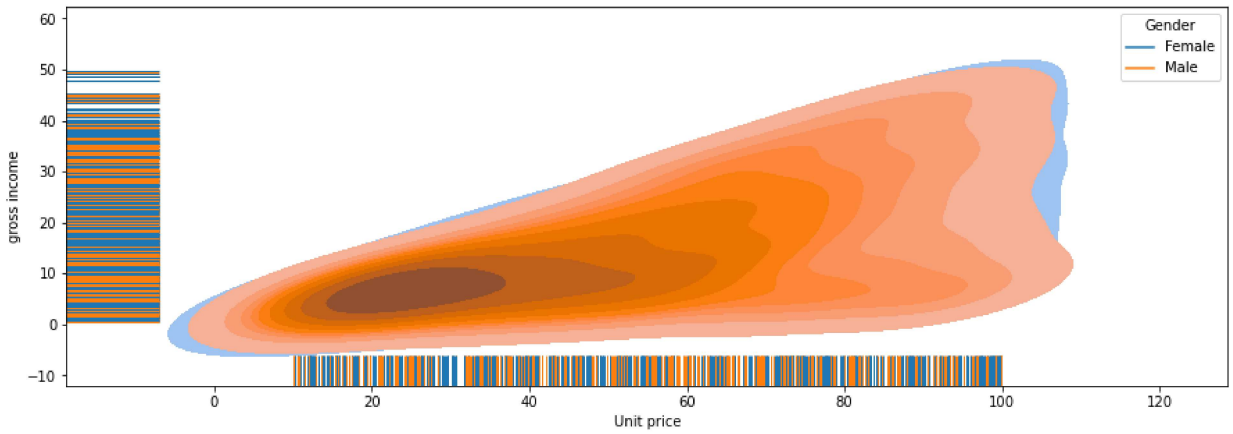
Out[75]:
```
<AxesSubplot:xlabel='Unit price', ylabel='gross income'>
```



5. Combining a RUG Plot with a KDE Plot

In [84]:
```
plt.figure(figsize=(15,5))
sns.kdeplot(x='Unit price',y='gross income',data=mart,hue='Gender',fill=True)
sns.rugplot(x='Unit price',y='gross income',data=mart,hue='Gender',height=0.08)
```

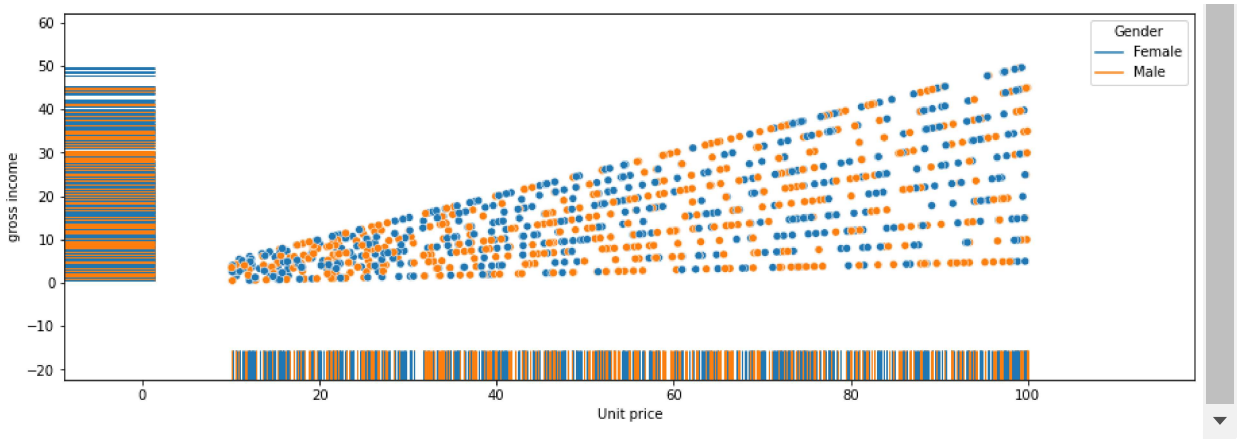
Out[84]:
```
<AxesSubplot:xlabel='Unit price', ylabel='gross income'>
```



6. Combining a RUG Plot with a Scatter Plot

In [85]:
```
plt.figure(figsize=(15,5))
sns.scatterplot(x='Unit price',y='gross income',data=mart,hue='Gender')
sns.rugplot(x='Unit price',y='gross income',data=mart,hue='Gender',height=0.08)
```

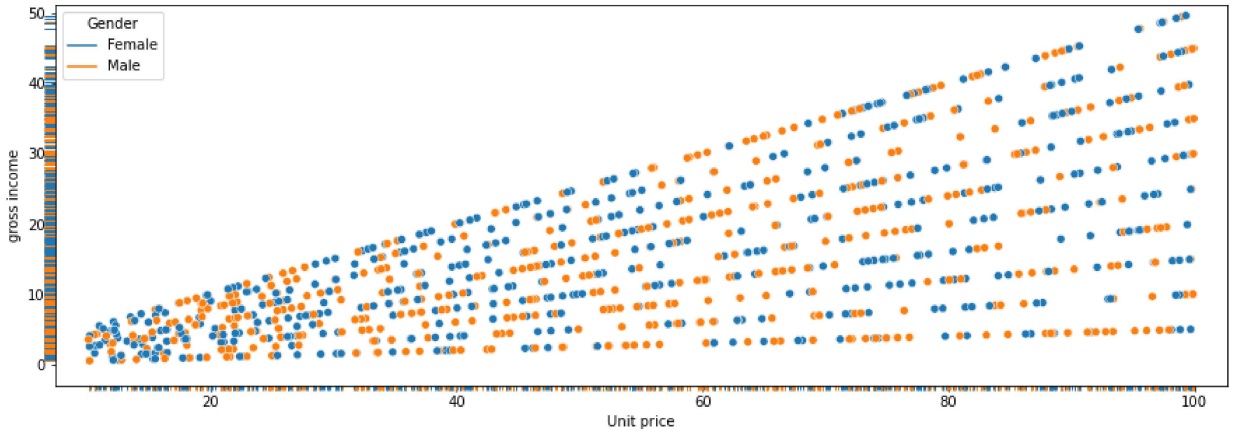
Out[85]:
```
<AxesSubplot:xlabel='Unit price', ylabel='gross income'>
```



7. Showing Rugs/Ticks out side of the Plot

In [87]:
```
plt.figure(figsize=(15,5))
sns.scatterplot(x='Unit price',y='gross income',data=mart,hue='Gender')
sns.rugplot(x='Unit price',y='gross income',data=mart,hue='Gender',height=-0.01,clip_c
```

Out[87]:
```
<AxesSubplot:xlabel='Unit price', ylabel='gross income'>
```



ECDF PLOT

[Emprical Cumulative Distribution Function]

Represents the proportion or count of observations falling below each unique values in a dataset

In [88]:
```
mart.head()
```

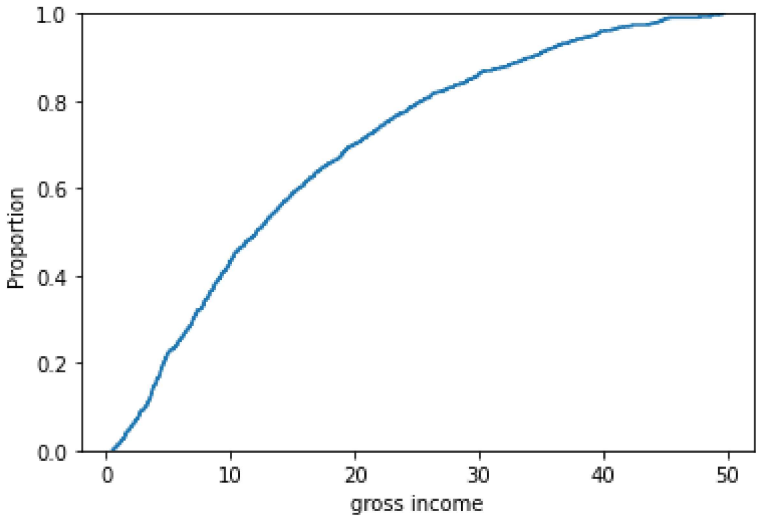
Out[88]:

	Gender	Payment	Unit price	Quantity	Total	gross income
0	Female	Ewallet	74.69	7	548.9715	26.1415
1	Female	Cash	15.28	5	80.2200	3.8200
2	Male	Credit card	46.33	7	340.5255	16.2155
3	Male	Ewallet	58.22	8	489.0480	23.2880
4	Male	Ewallet	86.31	7	634.3785	30.2085

1. Creating basic ECDF plot based on one numeric variable

In [90]:
```
sns.ecdfplot(x='gross income',data=mart)
```

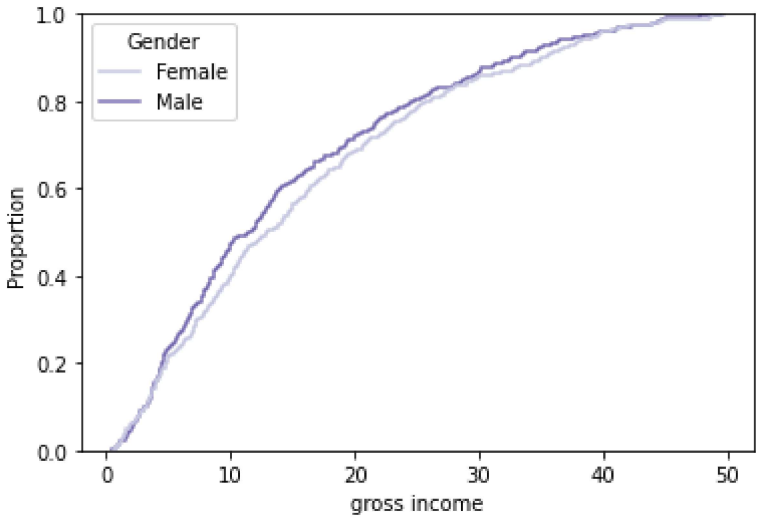
Out[90]:
```
<AxesSubplot:xlabel='gross income', ylabel='Proportion'>
```



2. Categorizing/Grouping the ECDF plot by a categorical variable value

In [93]:
```
sns.ecdfplot(data=mart,x='gross income',hue='Gender',palette='Purples')
```

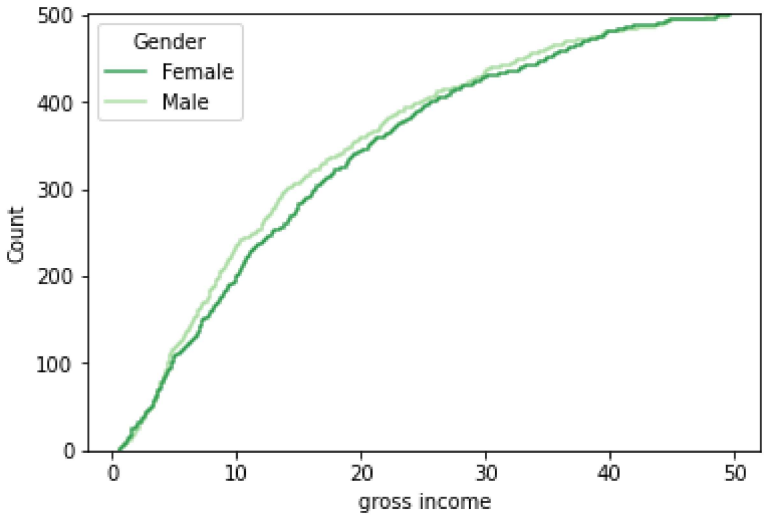
Out[93]:
```
<AxesSubplot:xlabel='gross income', ylabel='Proportion'>
```



3. Showing count in ECDF plot|Switching the stat from Proportation to Count in a ECDF plot

In [95]:
```
sns.ecdfplot(data=mart,x='gross income',hue='Gender',palette='Greens_r',stat='count')
```

Out[95]:
```
<AxesSubplot:xlabel='gross income', ylabel='Count'>
```



DIS PLOT

DISTRIBUTION PLOT

In [119…
```
mart.head()
```

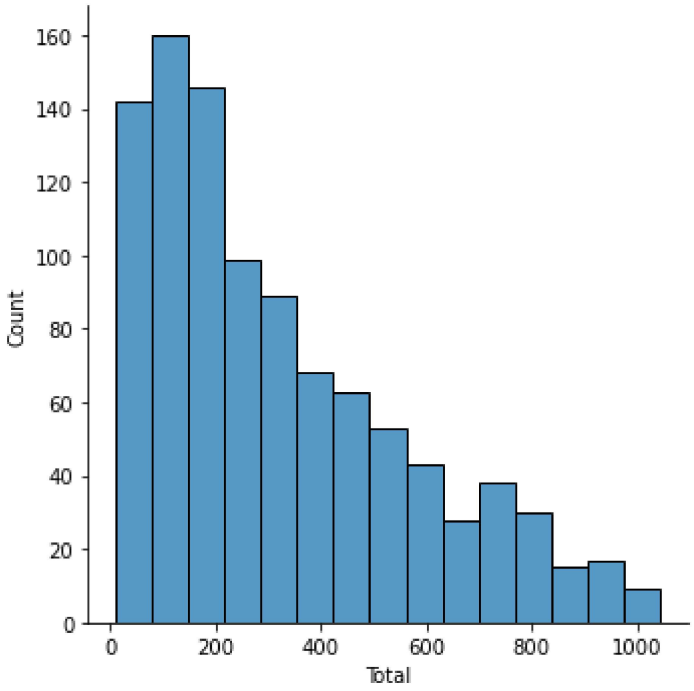
Out[119]:

	Gender	Payment	Unit price	Quantity	Total	gross income	Branch
0	Female	Ewallet	74.69	7	548.9715	26.1415	A
1	Female	Cash	15.28	5	80.2200	3.8200	C
2	Male	Credit card	46.33	7	340.5255	16.2155	A
3	Male	Ewallet	58.22	8	489.0480	23.2880	A
4	Male	Ewallet	86.31	7	634.3785	30.2085	A

1. Creating a basic Dis Plot

In [97]:
```
sns.displot(data=mart,x='Total') #default histogram
```

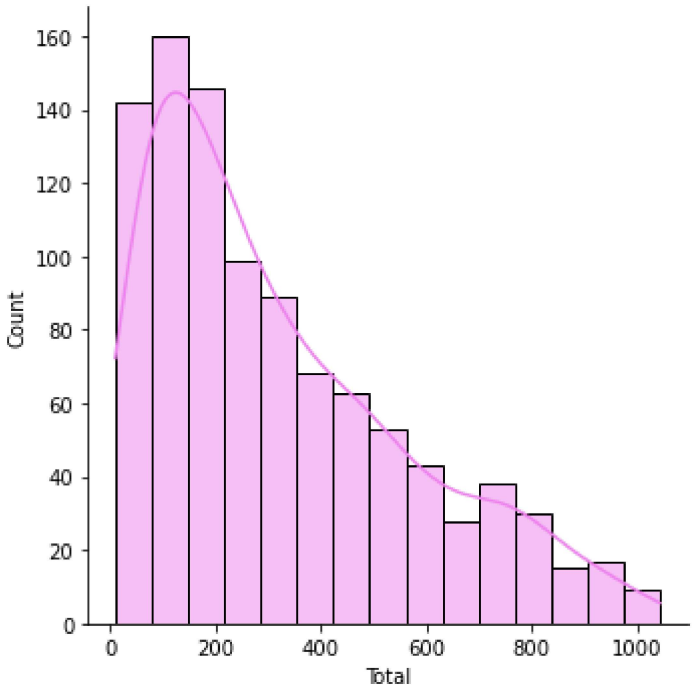
Out[97]:
```
<seaborn.axisgrid.FacetGrid at 0x1eb3b4c0ee0>
```



2. Including/Combining a KDE Plot with Histogram using Dis Plot

In [103…
```
sns.displot(data=mart,x='Total',kde=True,color='violet')
```

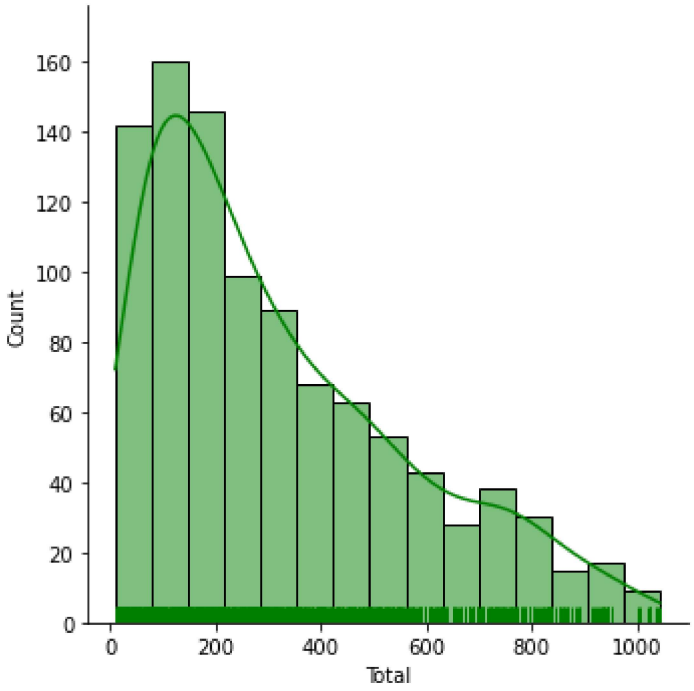
Out[103]:
```
<seaborn.axisgrid.FacetGrid at 0x1eb3c8645b0>
```



3. Including/Combining a Rug Plot with Histogram & KDE using Dis Plot|Creating a Histogram, KDE and Rug plot altogether in one plot using Dis Plot

In [106…
```
sns.displot(data=mart,x='Total',kde=True,rug=True,color='green')
```

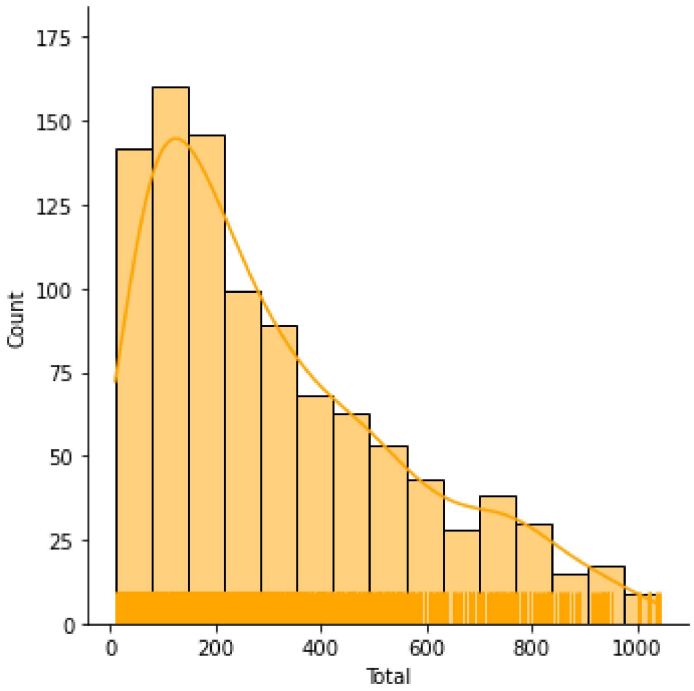
Out[106]:
```
<seaborn.axisgrid.FacetGrid at 0x1eb3ca1c460>
```



4. Make adjustments only in Rug Plot included in a Displot

In [109…
```
sns.displot(data=mart,x='Total',kde=True,rug=True,rug_kws={'height':0.05},color='orang
```

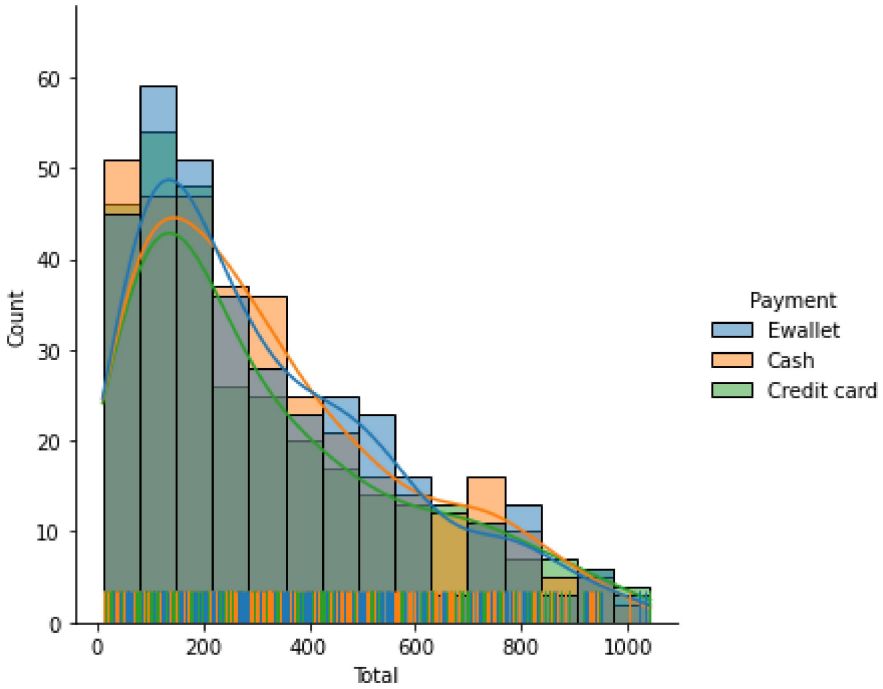
Out[109]:
```
<seaborn.axisgrid.FacetGrid at 0x1eb3caf9490>
```



5. Grouping a Dis Plot basis on a categorical variable

In [115…
```
sns.displot(data=mart,x='Total',kde=True,rug=True,rug_kws={'height':0.05},hue='Payment
```

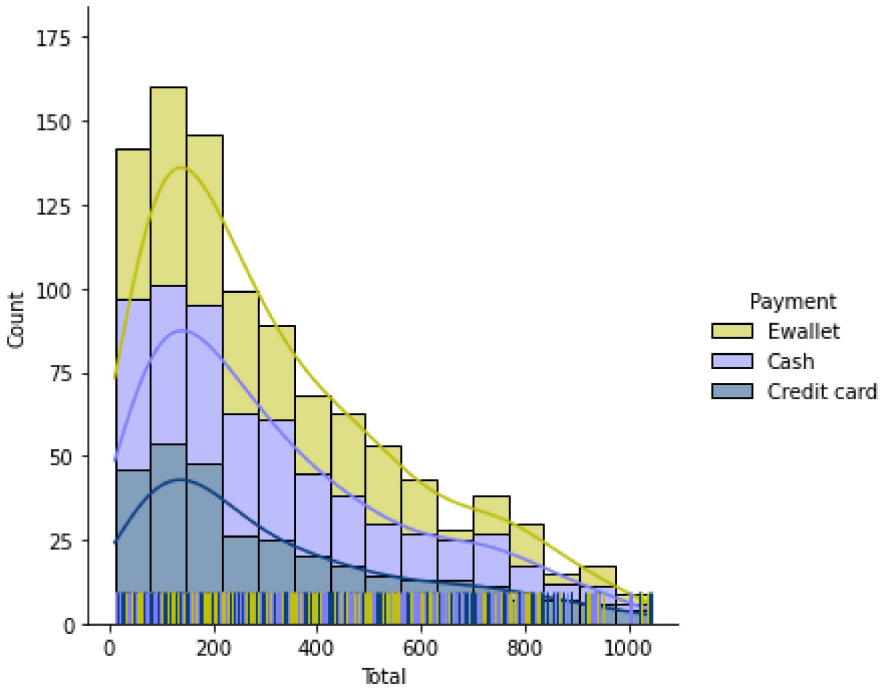
Out[115]:
```
<seaborn.axisgrid.FacetGrid at 0x1eb3cd1e070>
```



6. Making Stack plot in a Dis Plot

In [114…
```
sns.displot(data=mart,x='Total',kde=True,rug=True,rug_kws={'height':0.05},hue='Payment
```

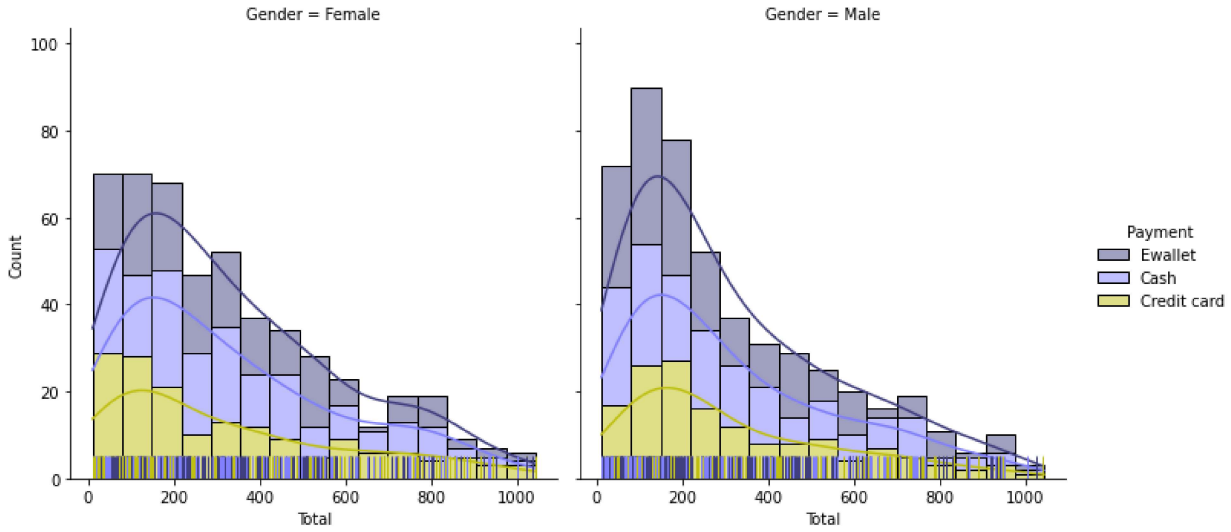
Out[114]:
```
<seaborn.axisgrid.FacetGrid at 0x1eb3cda7040>
```



7. Separating a Displot in multiple columns basis on the category in supplied categorical variable

In [117…
```
sns.displot(data=mart,x='Total',kde=True,rug=True,rug_kws={'height':0.05},hue='Payment
```

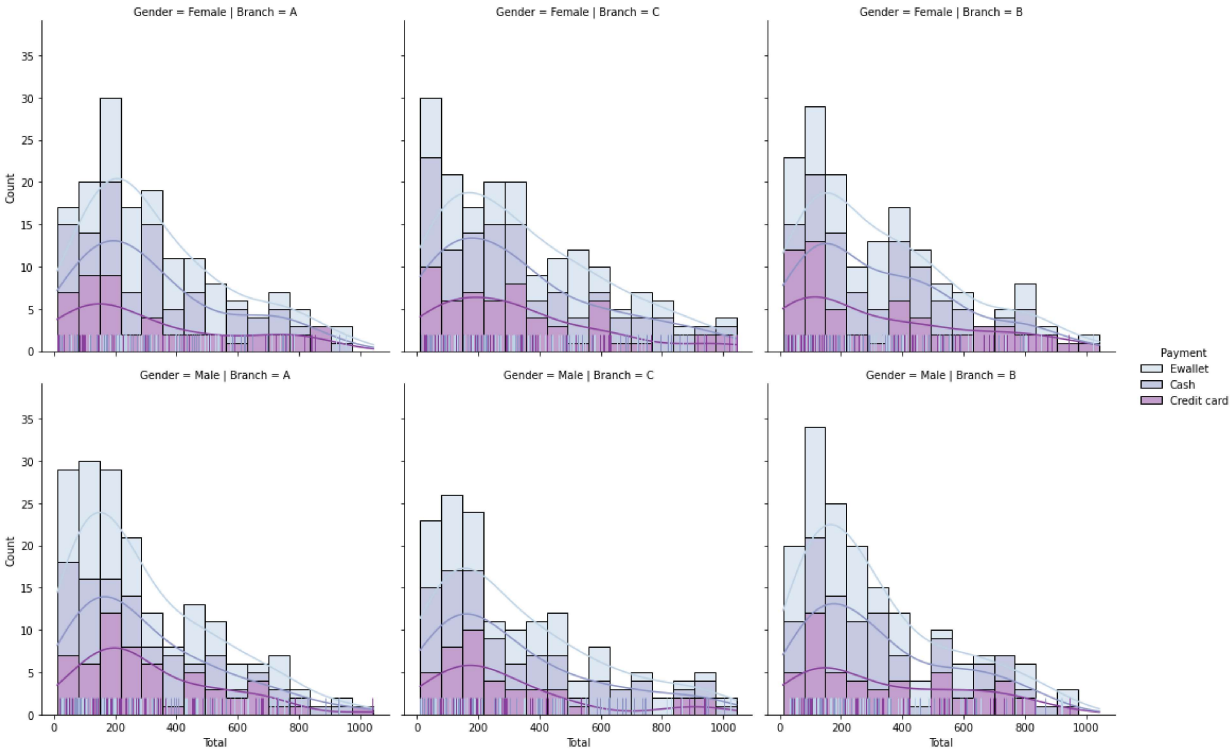
Out[117]:
```
<seaborn.axisgrid.FacetGrid at 0x1eb3e1a4ca0>
```



8. Separating a Displot in multiple rows basis on the category in supplied categorical variable

In [124…
```
sns.displot(data=mart,x='Total',kde=True,rug=True,rug_kws={'height':0.05},hue='Payment
```

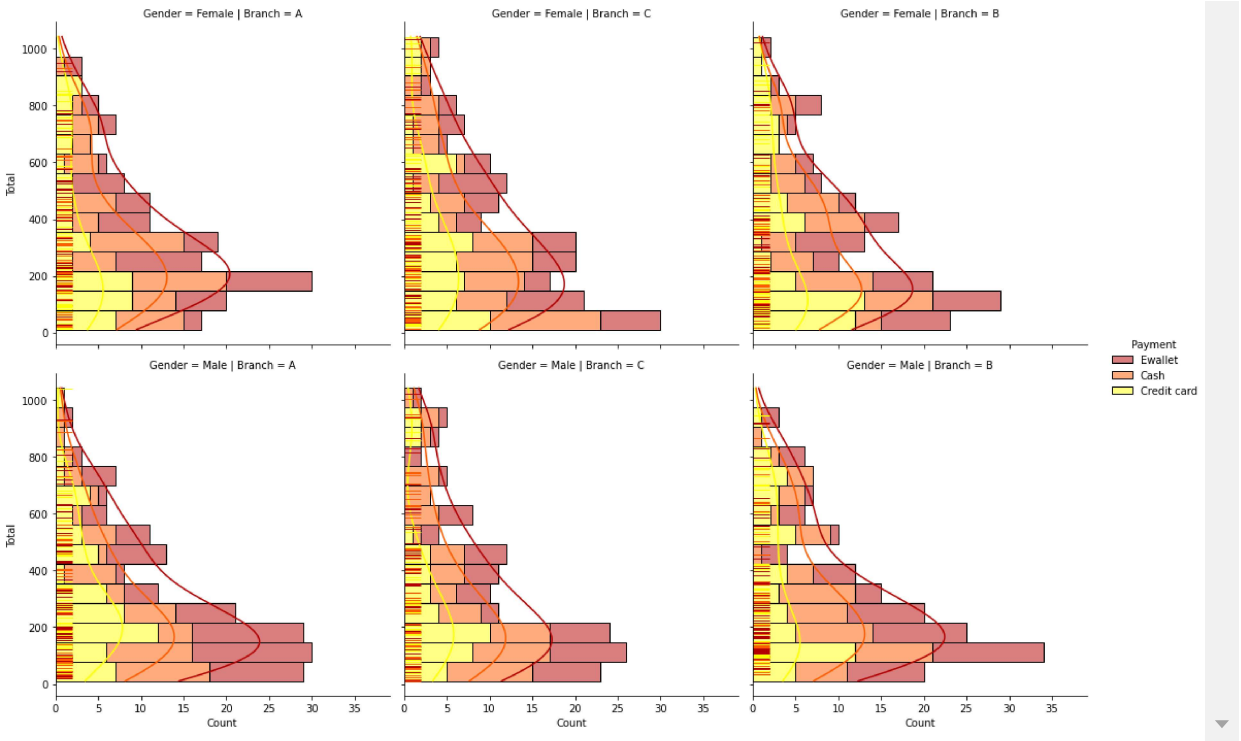
Out[124]:
```
<seaborn.axisgrid.FacetGrid at 0x1eb3e642d00>
```



9. Creating Bar Plots using Dis plot in Seaborn

In [127…
```
#change the Total to y axes
sns.displot(data=mart,y='Total',kde=True,rug=True,rug_kws={'height':0.05},hue='Payment
```

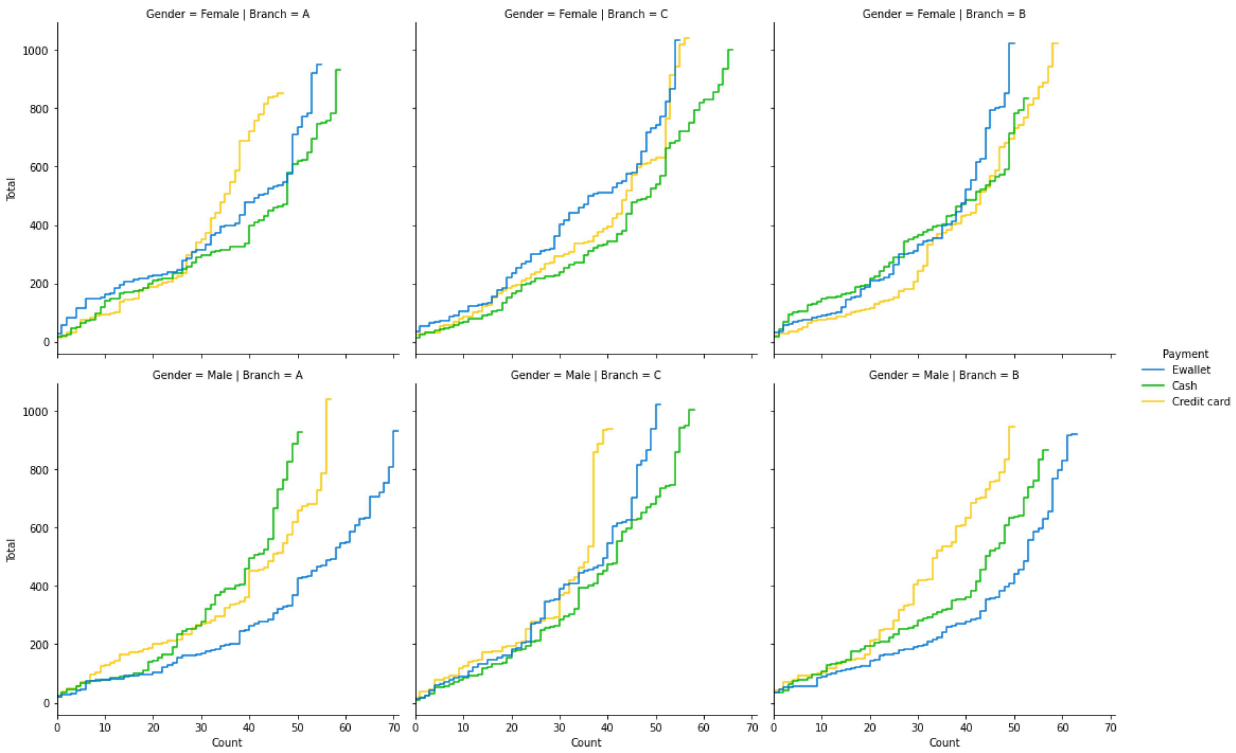
Out[127]:
```
<seaborn.axisgrid.FacetGrid at 0x1eb41be6fd0>
```



10. Creating ECDF Plot using Displot in Seaborn

In [134…
```
sns.displot(data=mart,y='Total',kind='ecdf',hue='Payment',palette='nipy_spectral',col=
```

Out[134]:
```
<seaborn.axisgrid.FacetGrid at 0x1eb45aacf70>
```



JOINT PLOT

Draw a plot of two variables with bivariate and univariate graphs

The plot consists of one relational plot and two marginal distribution plots(on x and y axes)

In [136…
```
iris = sns.load_dataset('iris')
iris.head()
```

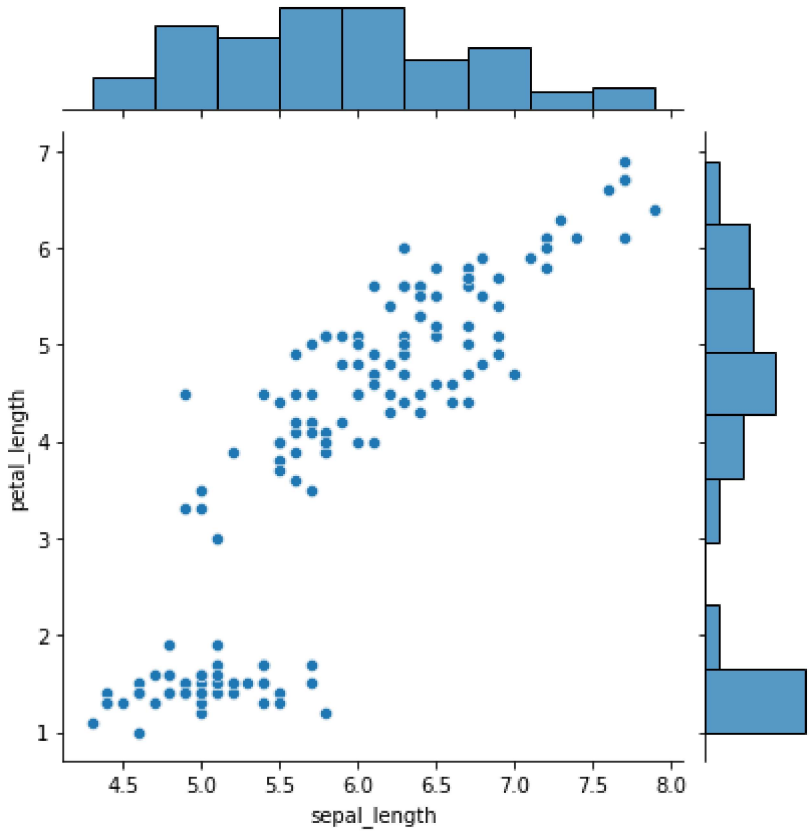
Out[136]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

1. Creating a basic JOINT PLOT

In [137…
```
sns.jointplot(data=iris,x='sepal_length',y='petal_length')
```

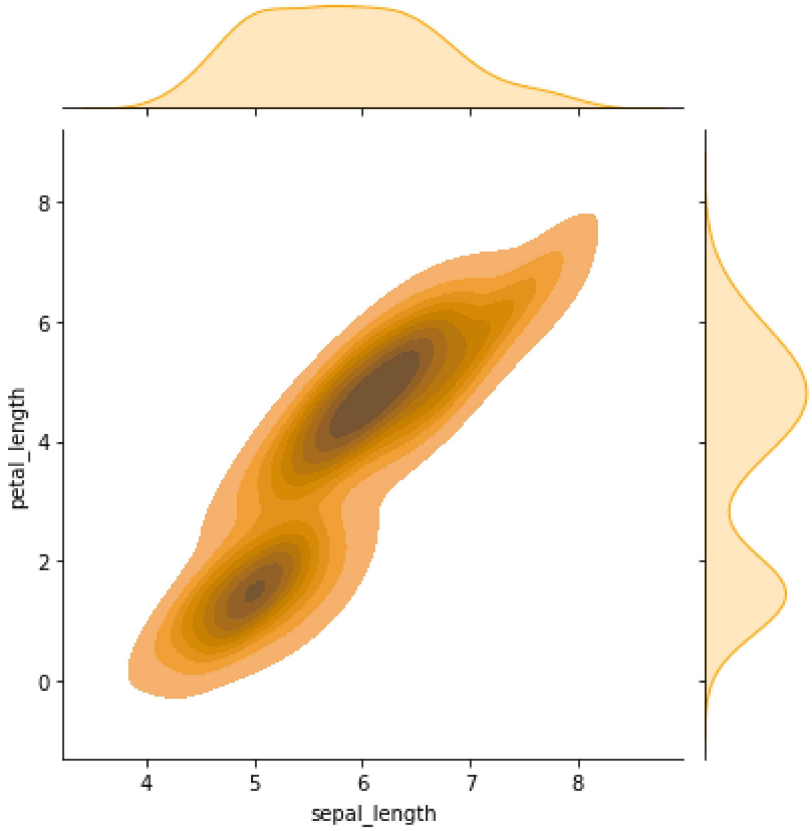
Out[137]:
```
<seaborn.axisgrid.JointGrid at 0x1eb439103d0>
```



2. Changing the Plot kind in a JOINT PLOT to SCATTER, KDE, HIST, HEX, REG or RESID plot

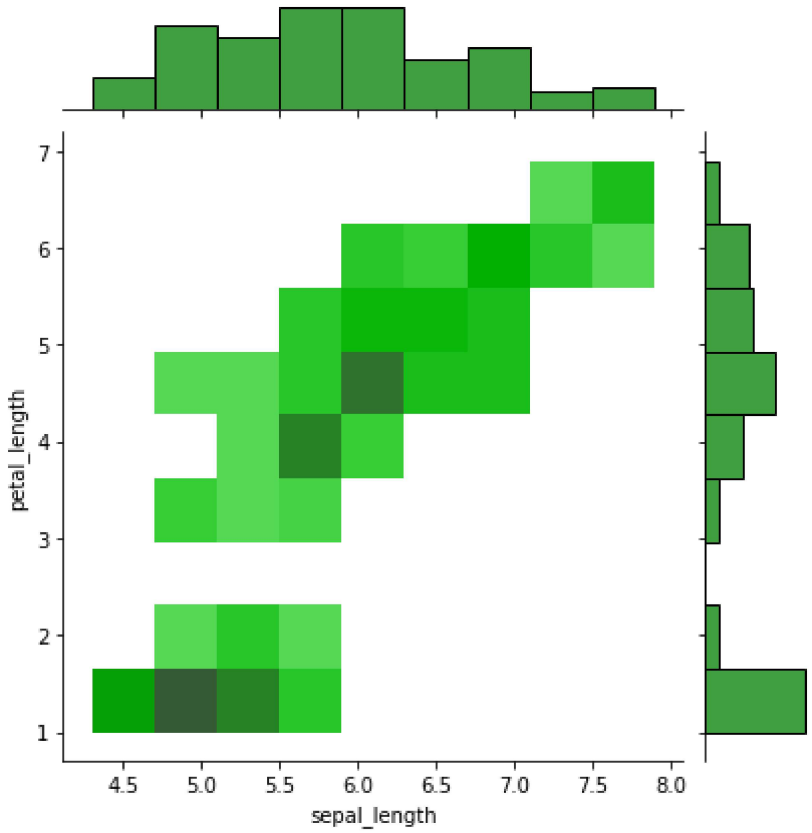
In [138…
```
sns.jointplot(data=iris,x='sepal_length',y='petal_length',kind='kde',fill=True,color='
```

Out[138]:
```
<seaborn.axisgrid.JointGrid at 0x1eb46ad4f10>
```



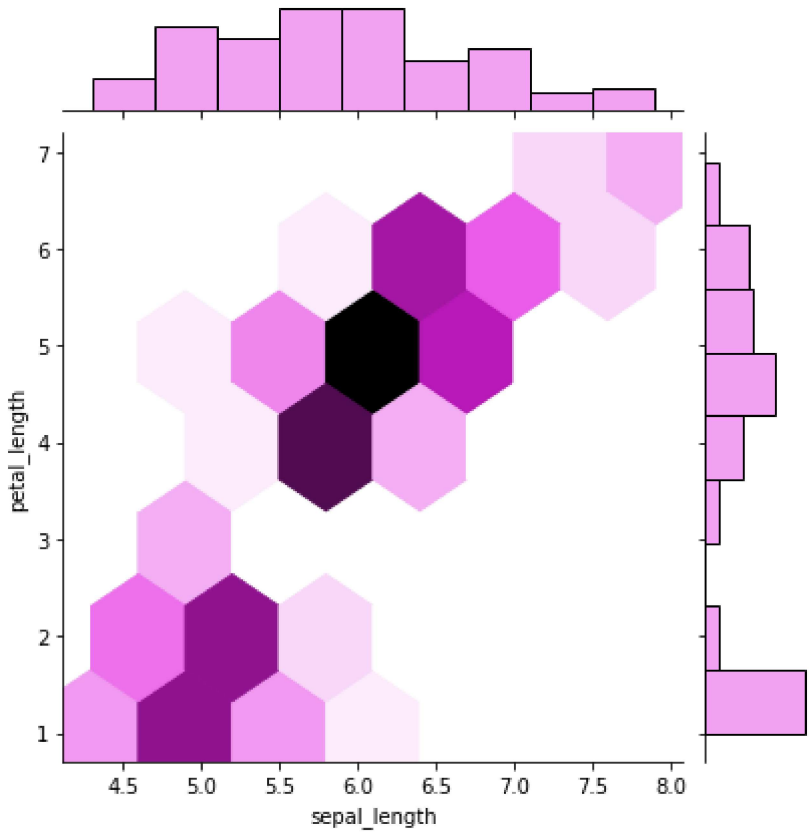
In [141…
```
sns.jointplot(data=iris,x='sepal_length',y='petal_length',kind='hist',color='green')
```

Out[141]:
```
<seaborn.axisgrid.JointGrid at 0x1eb488a7eb0>
```



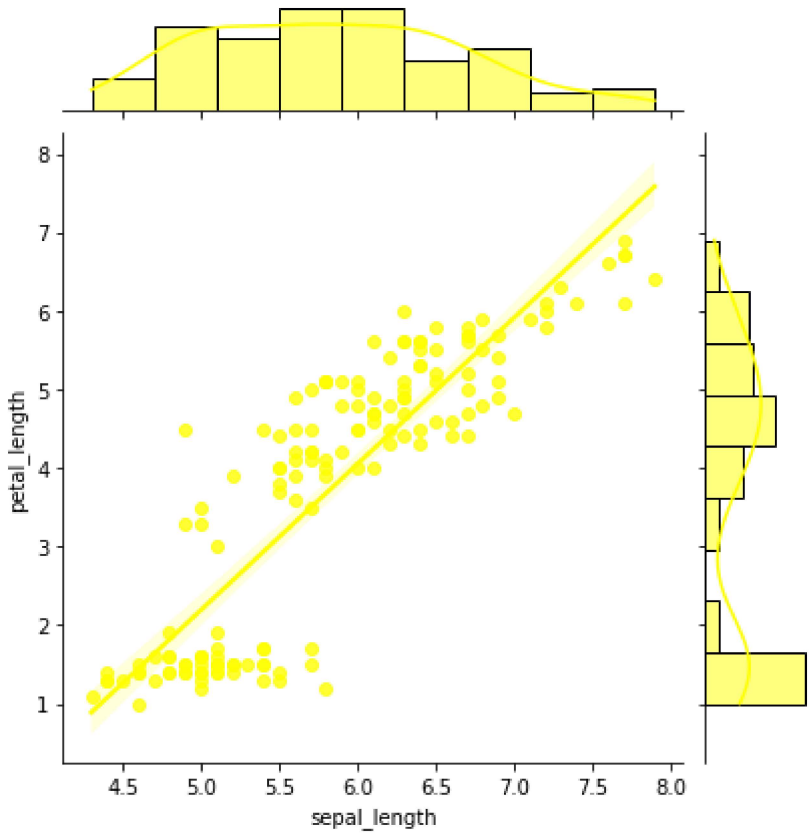
In [142…
```
sns.jointplot(data=iris,x='sepal_length',y='petal_length',kind='hex',color='violet')
```

Out[142]: <seaborn.axisgrid.JointGrid at 0x1eb489c2100>



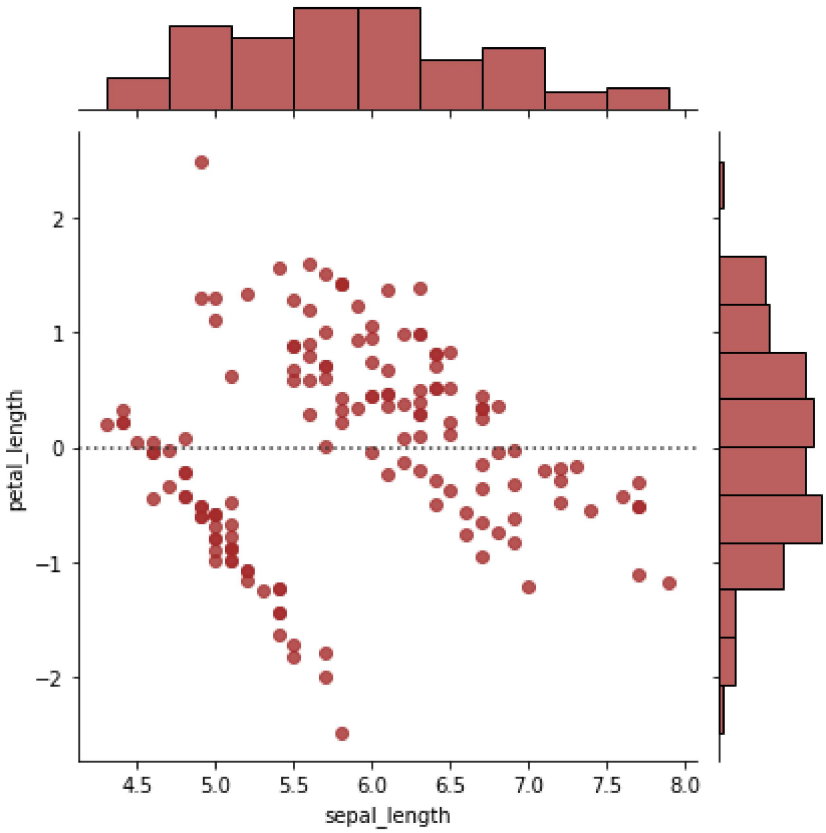
In [144…
```
sns.jointplot(data=iris,x='sepal_length',y='petal_length',kind='reg',color='yellow')
```

Out[144]: <seaborn.axisgrid.JointGrid at 0x1eb488f29d0>



In [147…
```
sns.jointplot(data=iris,x='sepal_length',y='petal_length',kind='resid',color='brown')
```

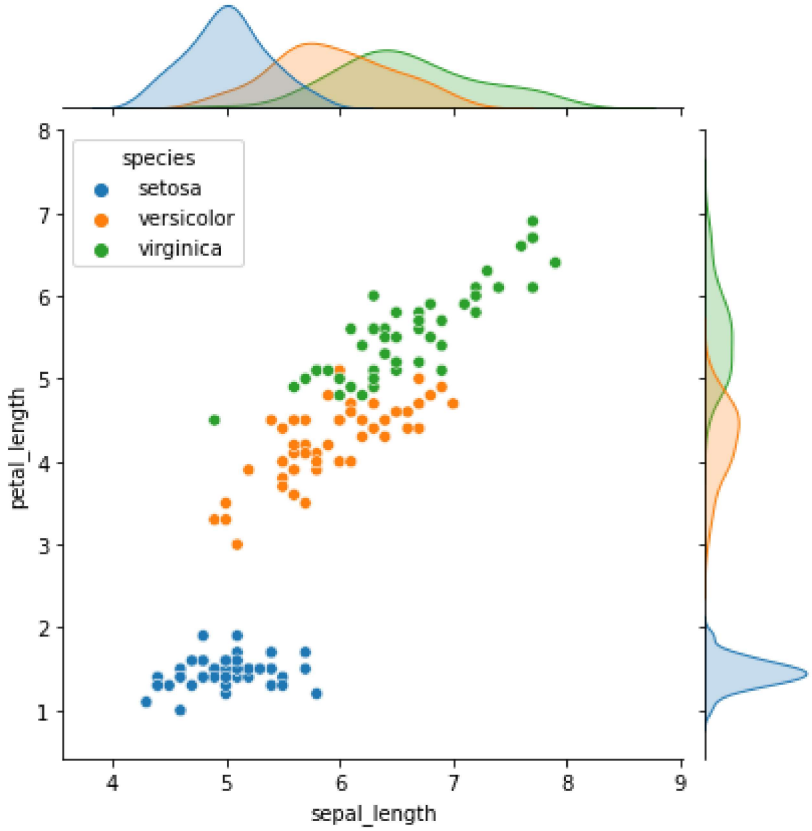
Out[147]:
```
<seaborn.axisgrid.JointGrid at 0x1eb48ca8d60>
```



3. Categorization in a JOINT PLOT basis on the value of a categorical variable using Hue arguement

In [148…
```
sns.jointplot(data=iris,x='sepal_length',y='petal_length',hue='species')
```

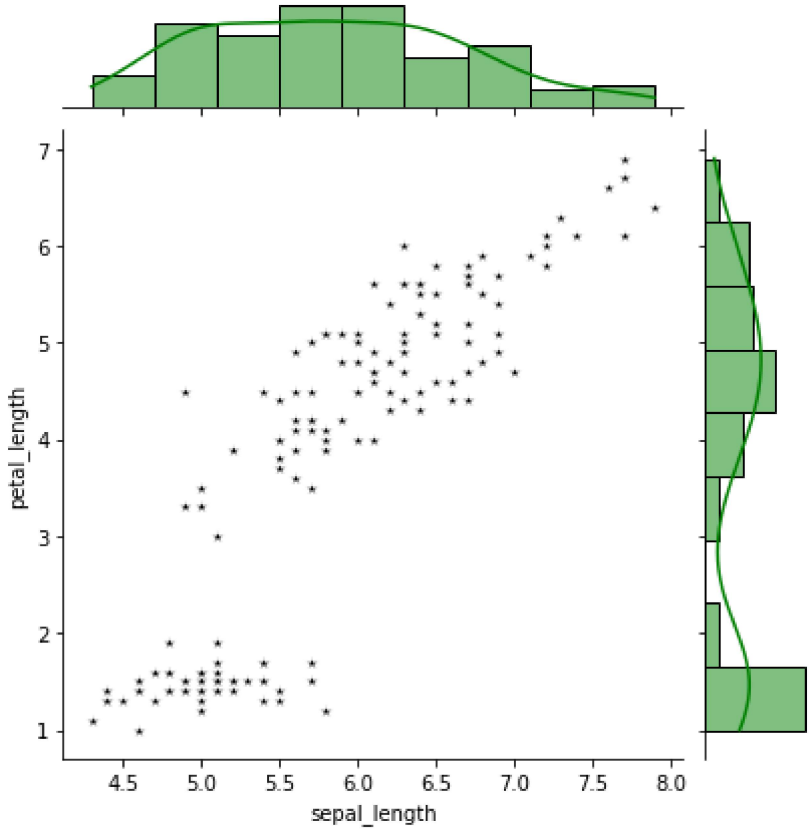
Out[148]:
```
<seaborn.axisgrid.JointGrid at 0x1eb48da3be0>
```



4. Changing the format of individual plots in a JOINT PLOT|Changing the format/style of Relational and Marginal Distribution plot separatly in a JOINT PLOT

In [157…
```
sns.jointplot(data=iris,x='sepal_length',y='petal_length',
              joint_kws=dict(marker='*',color='black'),
              marginal_kws=dict(color='green',kde=True))
```

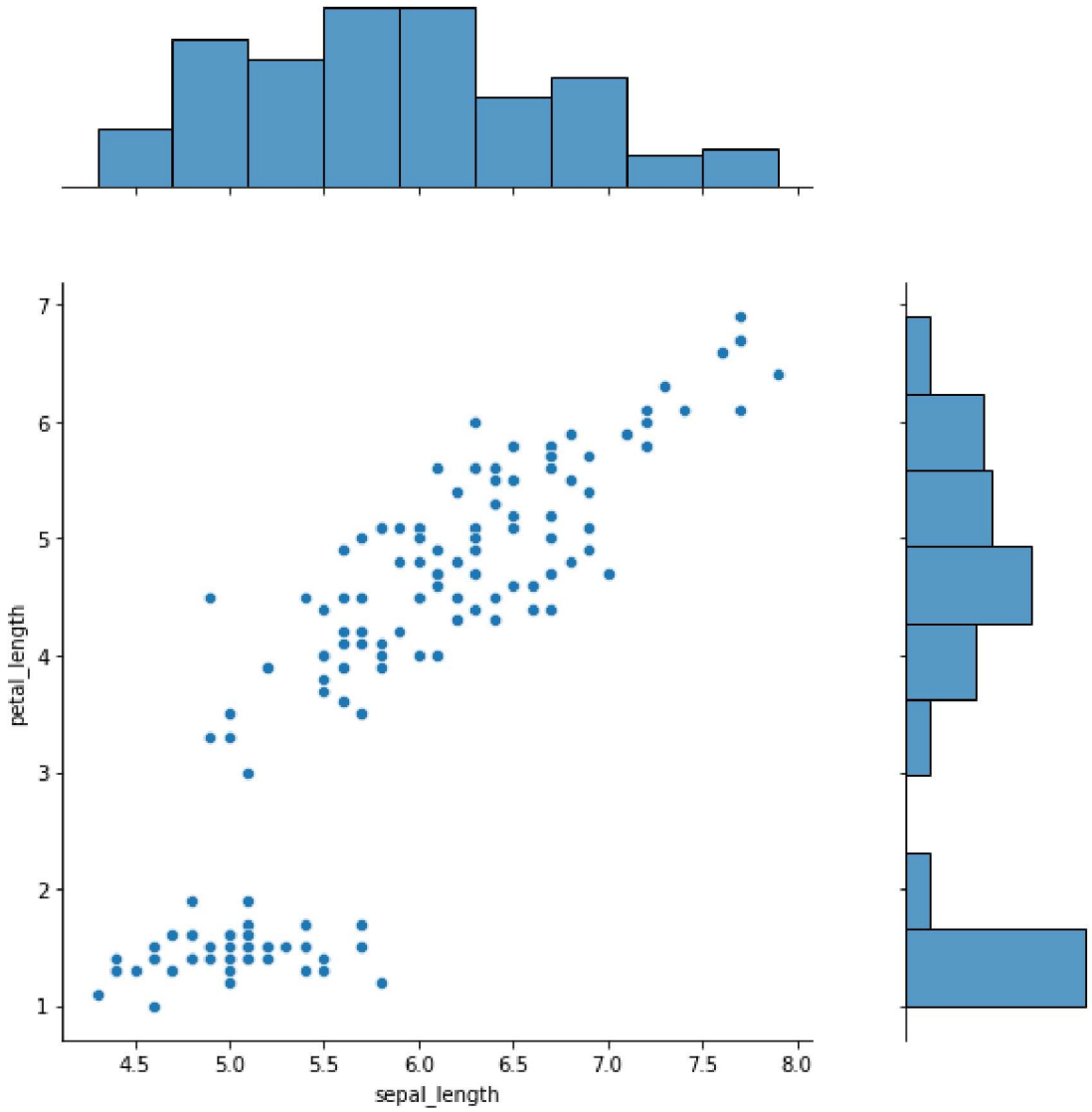
Out[157]:
```
<seaborn.axisgrid.JointGrid at 0x1eb4a5e79d0>
```



5. Adjusting Height, Ratio and Space in a JOINT PLOT

In [164…
```
sns.jointplot(data=iris,x='sepal_length',y='petal_length',
              height=8,
              ratio=3,
              space=0.5)
```

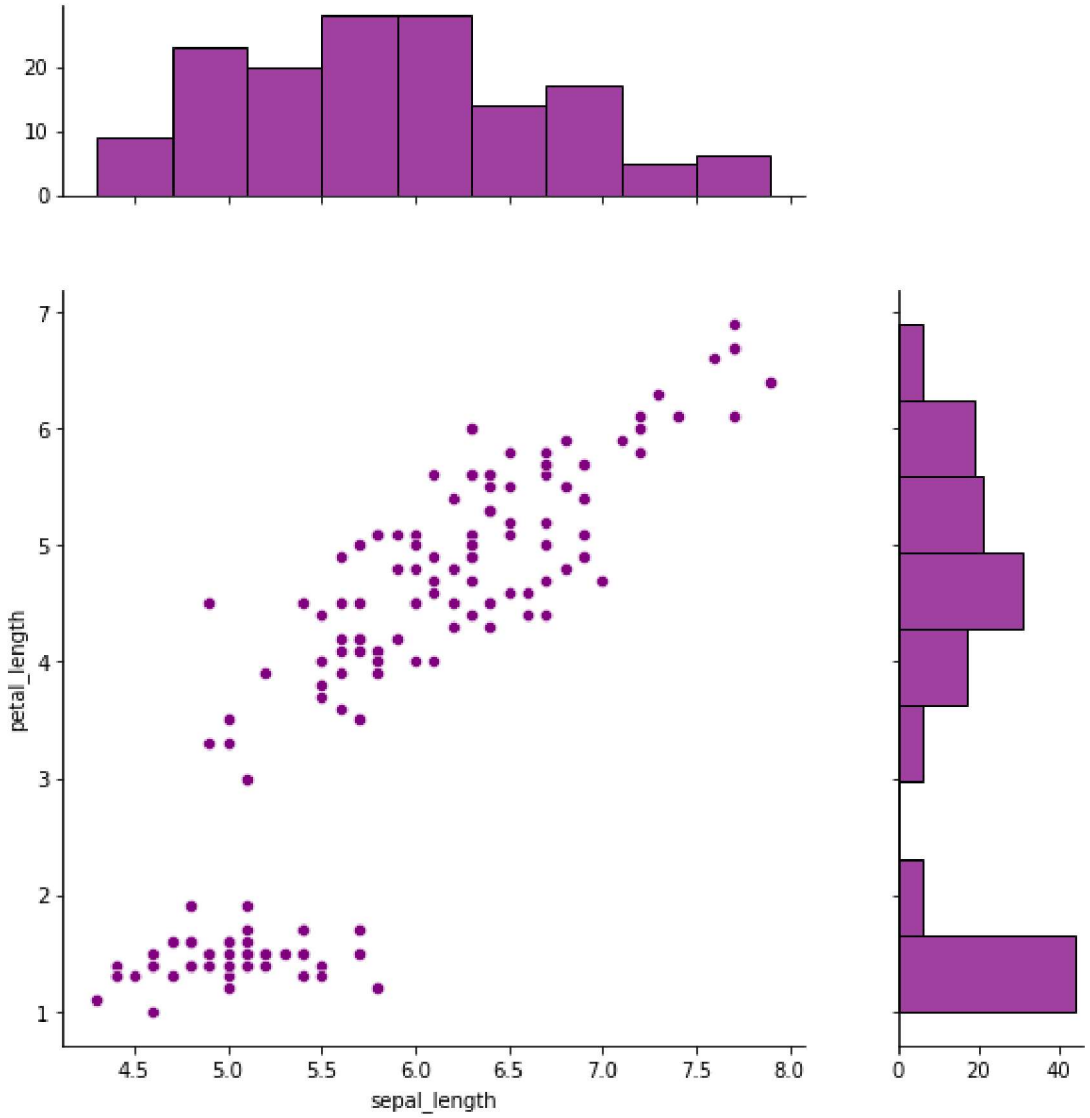
Out[164]:
```
<seaborn.axisgrid.JointGrid at 0x1eb4c881490>
```



6. Adding Y axis to the Marginal Distribution Plot in a JOINT PLOT| Showing/Hiding the Marginal Ticks of a Distribution Plot in a JOINT PLOT

In [167…
```
sns.jointplot(data=iris,x='sepal_length',y='petal_length',
              height=8,
              ratio=3,
              space=0.5,
              marginal_ticks=True,
             color='purple')
```

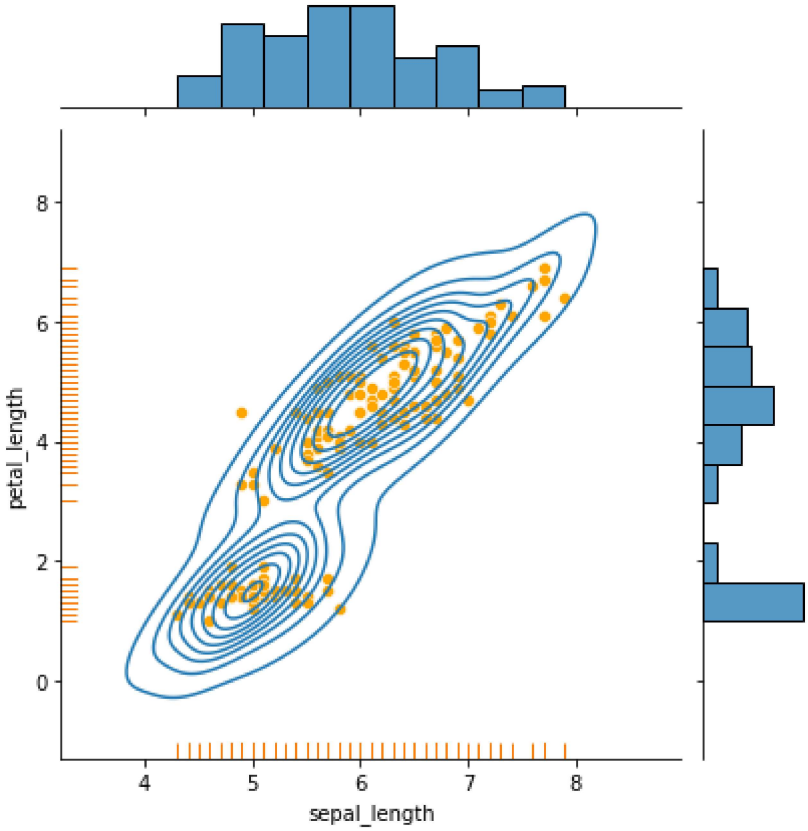
Out[167]:
```
<seaborn.axisgrid.JointGrid at 0x1eb4c640730>
```



7. Creating KDE and RUG Plot on top of the JOINT PLOT

In [172…
```
l = sns.jointplot(data=iris,x='sepal_length',y='petal_length',joint_kws=dict(color='or
l.plot_joint(sns.kdeplot)
l.plot_joint(sns.rugplot)
```

Out[172]:
```
<seaborn.axisgrid.JointGrid at 0x1eb4eb92dc0>
```



In [ ]: