Importing the libraries and reading the file

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
```

```
In [2]: df = pd.read_csv('monthly.csv')
        df
```

```
Out[2]:
        months     total_sales  total_quantity  total_amount
 0      August     11957        13442           2244412.31
 1      March      15154        17006           2809063.30
 2      July       14291        16069           2646899.69
 3      June       13556        15255           2578293.30
 4      October    20284        22705           3736884.05
 5      April      18289        20568           3396059.11
 6      December   24989        28121           4619297.12
 7      September  11629        13119           2098816.70
 8      May        16554        18657           3144584.80
 9      January     9681        10868           1815335.12
10      November   17580        19808           3198909.23
11      February   11986        13461           2203481.24
```

Arranging the months column

In [37]:

```
months = {
    1:'January',
    2:'February',
    3:'March',
    4:'April',
    5:'May',
    6:'June',
    7:'July',
    8:'August',
    9:'September',
    10:'October',
    11:'November',
    12:'December'
}
data_ = []
for month in months:
    for i in df.values:
        if (months[month].lower() == (i[0]).lower()):
            data_.append(i)

df = pd.DataFrame(data_,columns =['months','total_sales','total_quantity','total_amount'])
df
```

Out[37]:

	months	total_sales	total_quantity	total_amount
0	January	9681	10868	1815335.12
1	February	11986	13461	2203481.24
2	March	15154	17006	2809063.30
3	April	18289	20568	3396059.11
4	May	16554	18657	3144584.80
5	June	13556	15255	2578293.30
6	July	14291	16069	2646899.69
7	August	11957	13442	2244412.31
8	September	11629	13119	2098816.70
9	October	20284	22705	3736884.05

	months	total_sales	total_quantity	total_amount
10	November	17580	19808	3198909.23

Adding a column

In [49]:

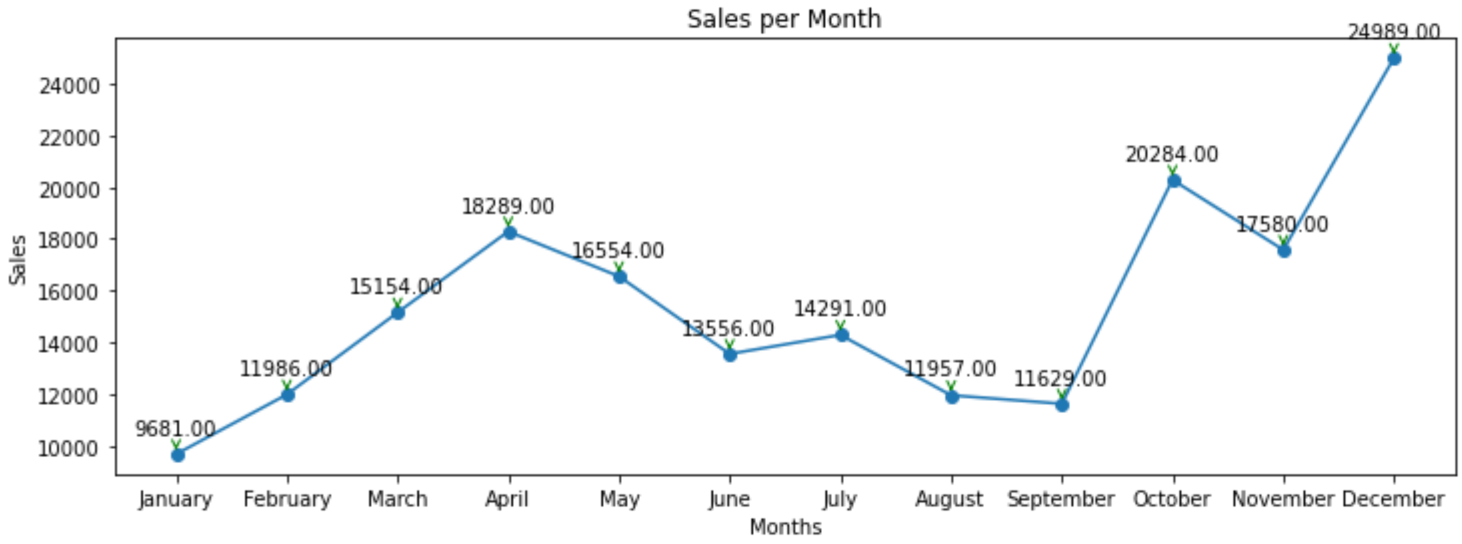
```
amt_per_sale = []

for i in df.values:
    amt_per_sale.append(round(i[3]/i[2],2))

df['avg_amt_collected_per_quantity_sold'] = amt_per_sale
df
```

Out[49]:

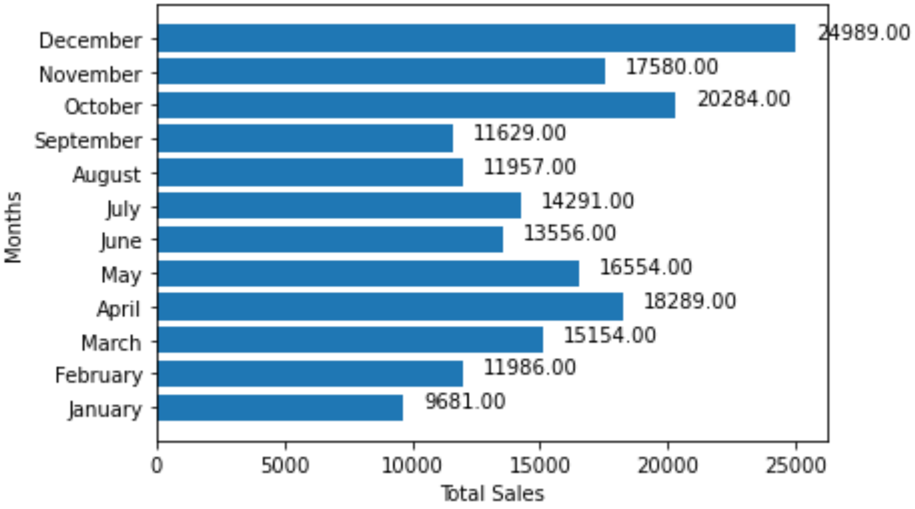
	months	total_sales	total_quantity	total_amount	avg_amt_collected_per_quantity_sold
0	January	9681	10868	1815335.12	167.03
1	February	11986	13461	2203481.24	163.69
2	March	15154	17006	2809063.30	165.18
3	April	18289	20568	3396059.11	165.11
4	May	16554	18657	3144584.80	168.55
5	June	13556	15255	2578293.30	169.01
6	July	14291	16069	2646899.69	164.72
7	August	11957	13442	2244412.31	166.97
8	September	11629	13119	2098816.70	159.98
9	October	20284	22705	3736884.05	164.58
10	November	17580	19808	3198909.23	161.50
11	December	24989	28121	4619297.12	164.27

Data Visualization | Monthly sales | line Graph, Bar graph and Pie Chart

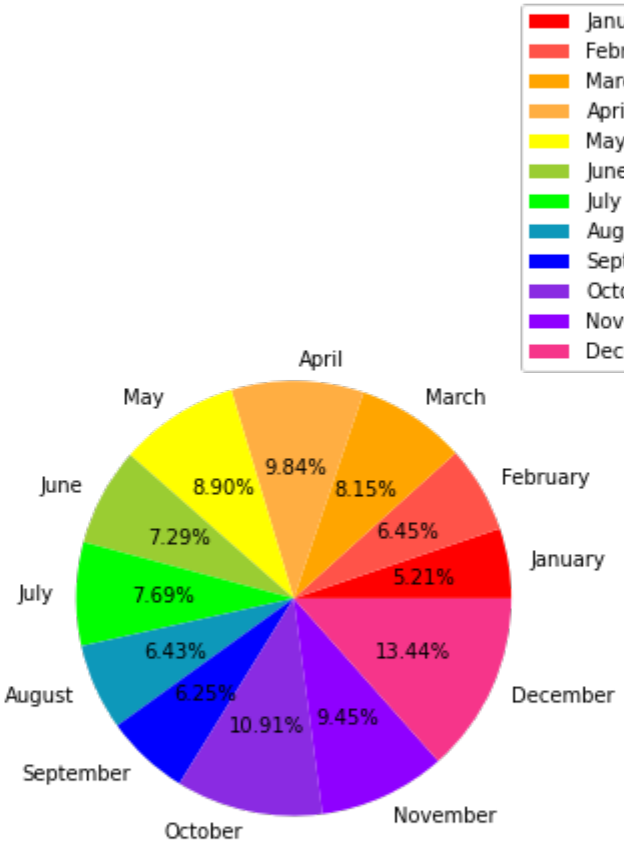
```
In [79]: fig,axs = plt.subplots(figsize =(12,4))
         plt.plot(df['months'],df['total_sales'],marker ='o')
         plt.title('Sales per Month')
         plt.xlabel('Months')
         plt.ylabel('Sales')
         for x,y in zip(df['months'],df['total_sales']):

             label = "{:.2f}".format(y)

             plt.annotate(label,                                            # this is the value which we want to la
                          (x,y),                                            # x and y is the points location where
                          textcoords = "offset points",
                          xytext     = (0,10),                              # this for the distance between the po
                          ha         = 'center',                            # and the text label
                          arrowprops = dict(arrowstyle="->", color='green'))
         plt.show()
```



```
In [138]: plt.barh(df['months'],df['total_sales'])
          for x,y in zip(df['total_sales'],df['months']):

              label = "{:.2f}".format(x)

              plt.annotate(label,                                                 # this is the value which we want to la
                           (x,y),                                                 # x and y is the points location where
                           textcoords = "offset points",
                           xytext     = (10,0),                                   # this for the distance between the po
          #                  ha         = 'left',                                 # and the text label
          #                  arrowprops = dict(arrowstyle="->", color='green')
                          )
          plt.xlabel('Total Sales')
          plt.ylabel('Months')
          plt.show()
```

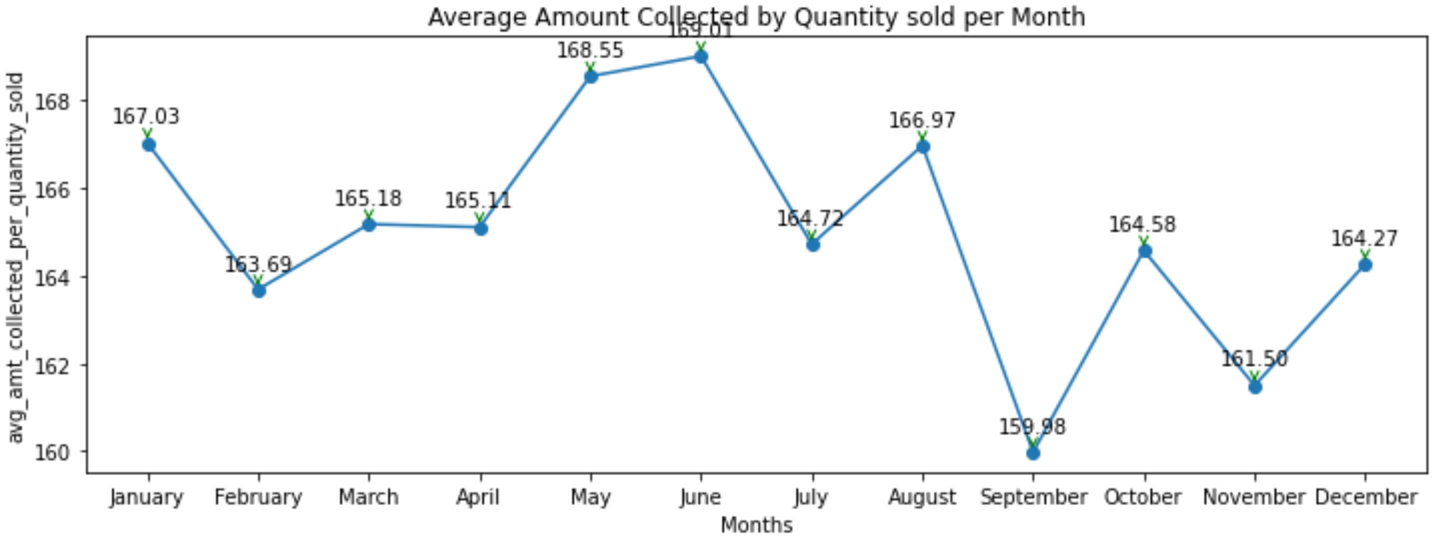


```
In [139]: fig,axs = plt.subplots(figsize =(12,5))
          plt.pie(df['total_sales'], labels = df['months'],autopct = '%1.2f%%',colors =['#FF0000','#FF5349','#
          plt.legend(loc = 'lower left',borderaxespad=25)
          plt.show()
```



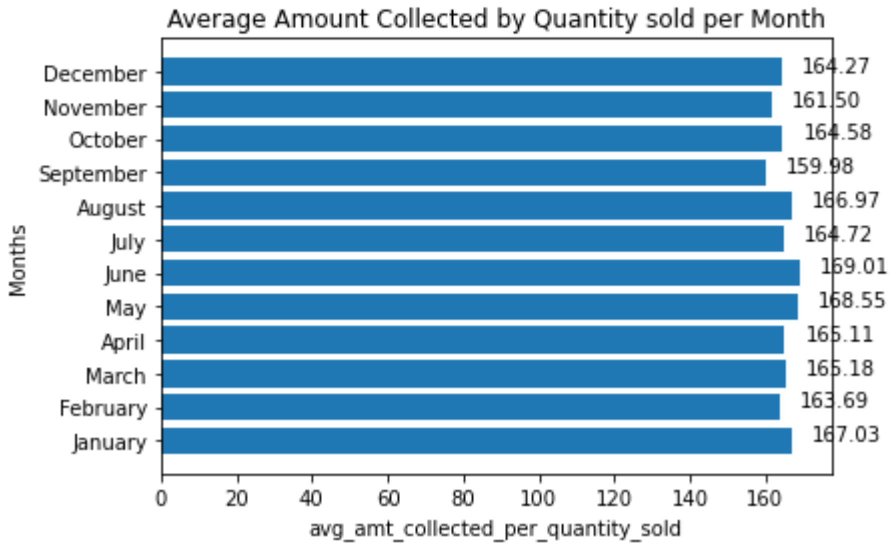
Data Visualization | Average amount collected per quantity sold | line chart ,bar graph and pie

```
In [146]: fig,axs = plt.subplots(figsize =(12,4))
          plt.plot(df['months'],df['avg_amt_collected_per_quantity_sold'],marker ='o')
          plt.title('Average Amount Collected by Quantity sold per Month')
          plt.xlabel('Months')
          plt.ylabel('avg_amt_collected_per_quantity_sold')
          for x,y in zip(df['months'],df['avg_amt_collected_per_quantity_sold']):

              label = "{:.2f}".format(y)

              plt.annotate(label,                                             # this is the value which we want to la
                           (x,y),                                             # x and y is the points location where
                           textcoords = "offset points",
                           xytext     = (0,10),                               # this for the distance between the po
                           ha         = 'center',                             # and the text label
                           arrowprops = dict(arrowstyle="->", color='green'))
          plt.show()
```

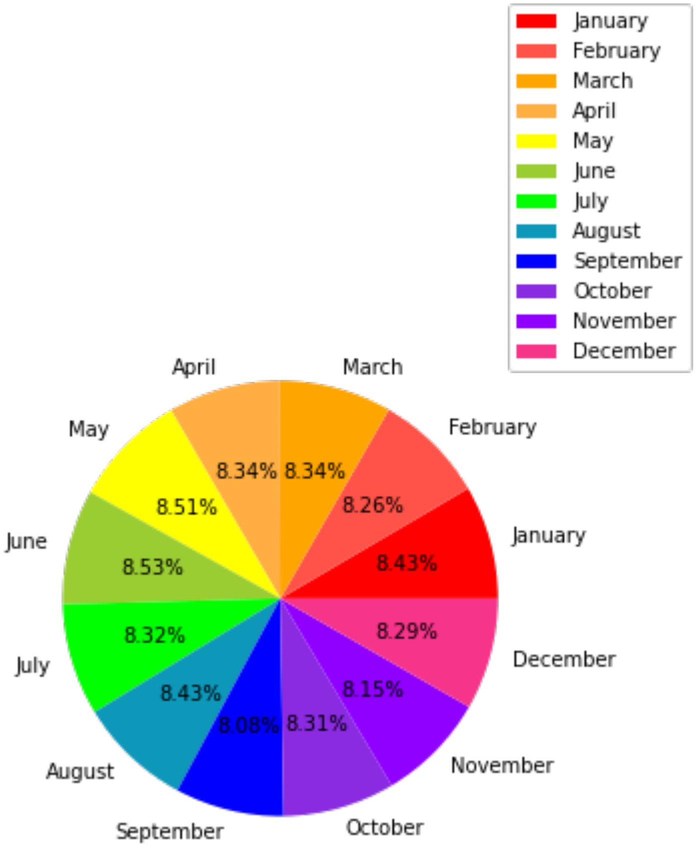


In [147]:

```
plt.barh(df['months'],df['avg_amt_collected_per_quantity_sold'])
for x,y in zip(df['avg_amt_collected_per_quantity_sold'],df['months']):

    label = "{:.2f}".format(x)

    plt.annotate(label,                                                # this is the value which we want to la
                 (x,y),                                                # x and y is the points location where
                 textcoords = "offset points",
                 xytext     = (10,0),                                  # this for the distance between the po
#                  ha         = 'left',                                # and the text label
#                  arrowprops = dict(arrowstyle="->", color='green')
                )
plt.title('Average Amount Collected by Quantity sold per Month')
plt.xlabel('avg_amt_collected_per_quantity_sold')
plt.ylabel('Months')
plt.show()
```



```
In [140]: fig,axs = plt.subplots(figsize =(12,5))
          plt.pie(df['avg_amt_collected_per_quantity_sold'], labels = df['months'],autopct = '%1.2f%%',colors
          plt.legend(loc = 'lower left',borderaxespad=25)
          plt.show()
```


January
February
March
April
May
June
July
August
September
October
November
December
April
March
May
February
June
January
December
July
November
August
September
October