python assignment modul 6

June 23, 2023

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

1 Module 6: Data Visualization Assignment 1

You work in XYZ Corporation as a Data Analyst. Your corporation has told you to visualize the mtcars.csv dataset with various plots. Dataset Link Tasks To Be Performed: 1. Start off by importing the cars.csv file in the jupyter notebook. 2. Generate a line plot graph for the column ‘model’ and ‘hp’ a. Map the ‘model’ column on the x-axis b. Map the ‘hp’ column on the y-axis c. Provide the x-axis label as Models of the cars d. Provide the y-axis label as Horse-Power of Cars e. Set the title as Model Names

```
[3]: data=pd.read_csv('cars-1.csv')
```

```
[4]: data
```

```
[4]:                   model   mpg  cyl   disp   hp  drat     wt   qsec  vs  am  \
     0             Mazda RX4  21.0    6  160.0  110  3.90  2.620  16.46   0   1
     1         Mazda RX4 Wag  21.0    6  160.0  110  3.90  2.875  17.02   0   1
     2            Datsun 710  22.8    4  108.0   93  3.85  2.320  18.61   1   1
     3        Hornet 4 Drive  21.4    6  258.0  110  3.08  3.215  19.44   1   0
     4     Hornet Sportabout  18.7    8  360.0  175  3.15  3.440  17.02   0   0
     5               Valiant  18.1    6  225.0  105  2.76  3.460  20.22   1   0
     6            Duster 360  14.3    8  360.0  245  3.21  3.570  15.84   0   0
     7             Merc 240D  24.4    4  146.7   62  3.69  3.190  20.00   1   0
     8              Merc 230  22.8    4  140.8   95  3.92  3.150  22.90   1   0
     9              Merc 280  19.2    6  167.6  123  3.92  3.440  18.30   1   0
     10            Merc 280C  17.8    6  167.6  123  3.92  3.440  18.90   1   0
     11           Merc 450SE  16.4    8  275.8  180  3.07  4.070  17.40   0   0
     12           Merc 450SL  17.3    8  275.8  180  3.07  3.730  17.60   0   0
     13          Merc 450SLC  15.2    8  275.8  180  3.07  3.780  18.00   0   0
     14   Cadillac Fleetwood  10.4    8  472.0  205  2.93  5.250  17.98   0   0
     15  Lincoln Continental  10.4    8  460.0  215  3.00  5.424  17.82   0   0
     16    Chrysler Imperial  14.7    8  440.0  230  3.23  5.345  17.42   0   0
     17             Fiat 128  32.4    4   78.7   66  4.08  2.200  19.47   1   1
```

```
18          Honda Civic  30.4    4   75.7   52  4.93  1.615  18.52   1   1
19       Toyota Corolla  33.9    4   71.1   65  4.22  1.835  19.90   1   1
20        Toyota Corona  21.5    4  120.1   97  3.70  2.465  20.01   1   0
21     Dodge Challenger  15.5    8  318.0  150  2.76  3.520  16.87   0   0
22          AMC Javelin  15.2    8  304.0  150  3.15  3.435  17.30   0   0
23           Camaro Z28  13.3    8  350.0  245  3.73  3.840  15.41   0   0
24     Pontiac Firebird  19.2    8  400.0  175  3.08  3.845  17.05   0   0
25            Fiat X1-9  27.3    4   79.0   66  4.08  1.935  18.90   1   1
26        Porsche 914-2  26.0    4  120.3   91  4.43  2.140  16.70   0   1
27         Lotus Europa  30.4    4   95.1  113  3.77  1.513  16.90   1   1
28       Ford Pantera L  15.8    8  351.0  264  4.22  3.170  14.50   0   1
29         Ferrari Dino  19.7    6  145.0  175  3.62  2.770  15.50   0   1
30        Maserati Bora  15.0    8  301.0  335  3.54  3.570  14.60   0   1
31           Volvo 142E  21.4    4  121.0  109  4.11  2.780  18.60   1   1

    gear  carb
0      4     4
1      4     4
2      4     1
3      3     1
4      3     2
5      3     1
6      3     4
7      4     2
8      4     2
9      4     4
10     4     4
11     3     3
12     3     3
13     3     3
14     3     4
15     3     4
16     3     4
17     4     1
18     4     2
19     4     1
20     3     1
21     3     2
22     3     2
23     3     4
24     3     2
25     4     1
26     5     2
27     5     2
28     5     4
29     5     6
30     5     8
```

```
31     4     2
```

[5]:
```
data.head()
```

[5]:
```
                model   mpg  cyl   disp   hp  drat     wt   qsec  vs  am  gear  \
0           Mazda RX4  21.0    6  160.0  110  3.90  2.620  16.46   0   1     4
1       Mazda RX4 Wag  21.0    6  160.0  110  3.90  2.875  17.02   0   1     4
2          Datsun 710  22.8    4  108.0   93  3.85  2.320  18.61   1   1     4
3      Hornet 4 Drive  21.4    6  258.0  110  3.08  3.215  19.44   1   0     3
4   Hornet Sportabout  18.7    8  360.0  175  3.15  3.440  17.02   0   0     3

   carb
0     4
1     4
2     1
3     1
4     2
```

[6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 12 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   model   32 non-null     object
 1   mpg     32 non-null     float64
 2   cyl     32 non-null     int64
 3   disp    32 non-null     float64
 4   hp      32 non-null     int64
 5   drat    32 non-null     float64
 6   wt      32 non-null     float64
 7   qsec    32 non-null     float64
 8   vs      32 non-null     int64
 9   am      32 non-null     int64
 10  gear    32 non-null     int64
 11  carb    32 non-null     int64
dtypes: float64(5), int64(6), object(1)
memory usage: 3.1+ KB
```

[8]:
```
data.describe().T
```

[8]:
```
      count        mean         std     min        25%      50%     75%  \
mpg    32.0   20.090625    6.026948  10.400   15.42500   19.200   22.80
cyl    32.0    6.187500    1.785922   4.000    4.00000    6.000    8.00
disp   32.0  230.721875  123.938694  71.100  120.82500  196.300  326.00
hp     32.0  146.687500   68.562868  52.000   96.50000  123.000  180.00
drat   32.0    3.596563    0.534679   2.760    3.08000    3.695    3.92
```

```
wt      32.0    3.217250    0.978457   1.513    2.58125    3.325    3.61
qsec    32.0   17.848750    1.786943  14.500   16.89250   17.710   18.90
vs      32.0    0.437500    0.504016   0.000    0.00000    0.000    1.00
am      32.0    0.406250    0.498991   0.000    0.00000    0.000    1.00
gear    32.0    3.687500    0.737804   3.000    3.00000    4.000    4.00
carb    32.0    2.812500    1.615200   1.000    2.00000    2.000    4.00

          max
mpg    33.900
cyl     8.000
disp  472.000
hp    335.000
drat    4.930
wt      5.424
qsec   22.900
vs      1.000
am      1.000
gear    5.000
carb    8.000
```

[9]:

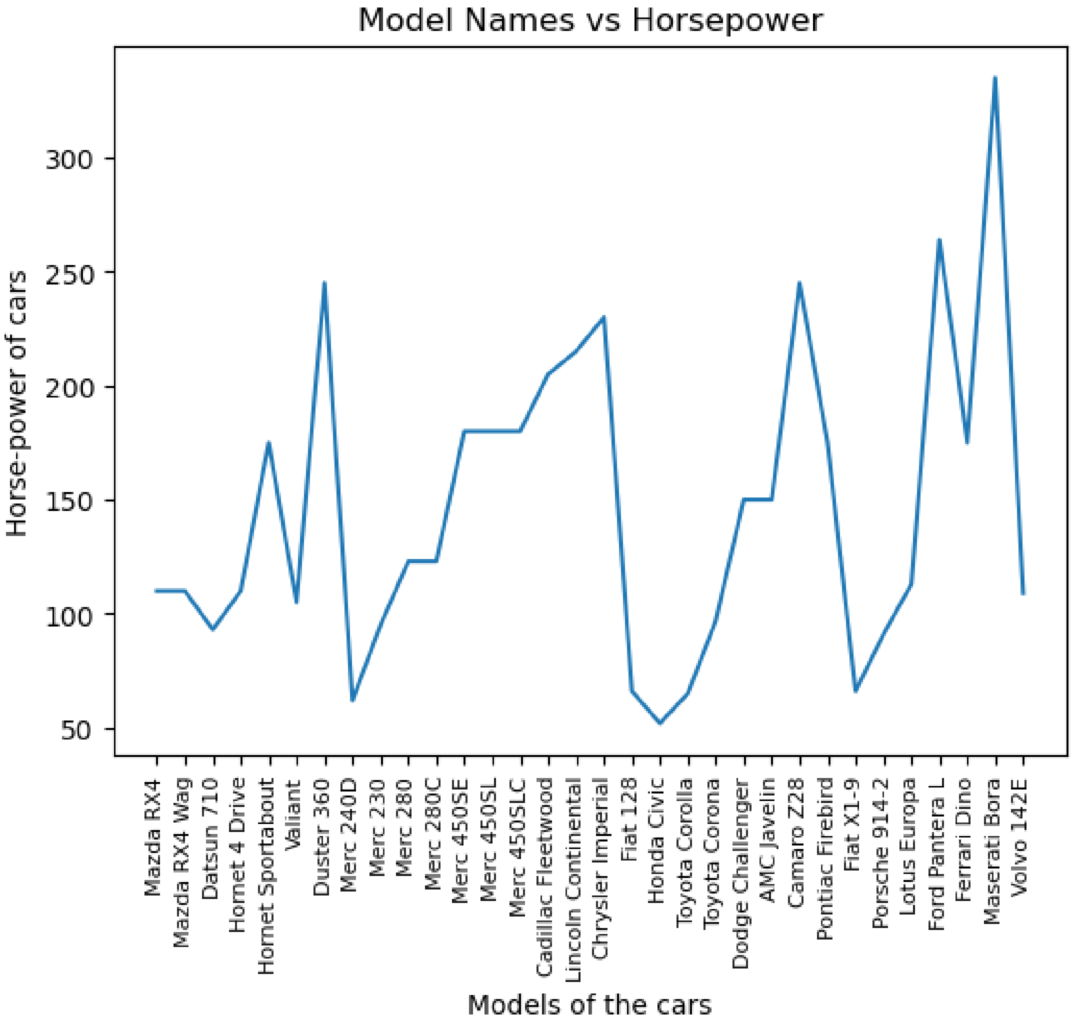
```python
data['model'].unique()
```

[9]:

```
array(['Mazda RX4', 'Mazda RX4 Wag', 'Datsun 710', 'Hornet 4 Drive',
       'Hornet Sportabout', 'Valiant', 'Duster 360', 'Merc 240D',
       'Merc 230', 'Merc 280', 'Merc 280C', 'Merc 450SE', 'Merc 450SL',
       'Merc 450SLC', 'Cadillac Fleetwood', 'Lincoln Continental',
       'Chrysler Imperial', 'Fiat 128', 'Honda Civic', 'Toyota Corolla',
       'Toyota Corona', 'Dodge Challenger', 'AMC Javelin', 'Camaro Z28',
       'Pontiac Firebird', 'Fiat X1-9', 'Porsche 914-2', 'Lotus Europa',
       'Ford Pantera L', 'Ferrari Dino', 'Maserati Bora', 'Volvo 142E'],
      dtype=object)
```

[15]:

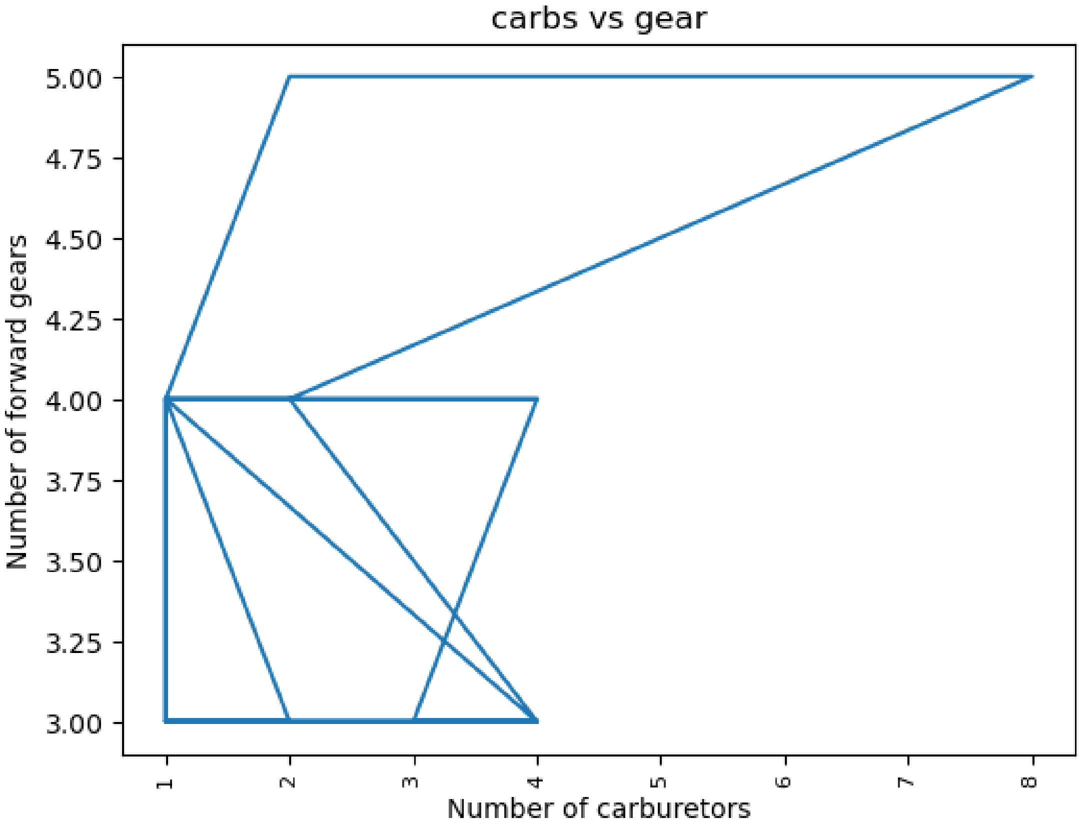
```python
plt.plot(data['model'], data['hp'])
plt.title('Model Names vs Horsepower')
plt.xlabel('Models of the cars')
plt.ylabel('Horse-power of cars')
plt.xticks(rotation='vertical',size=8)
plt.show()
```



2 Module 6: Data Visualization assignment 2

Python for Data Science Certification Course Problem Statement: You work in XYZ Corporation as a Data Analyst. Your corporation has told you to visualize the mtcars.csv dataset with various plots. Dataset Link Tasks To Be Performed: 1. Generate a bar plot graph for the columns 'carbs' and 'gear' a. Map the 'carbs' onto the x-axis. b. Map the 'gear' onto the y-axis. c. Provide the x-axis label as Number of carburetors. d. Provide the y-axis label as Number of forward gears. e. Set the title as carbs vs gear

[20]:
```
plt.plot(data['carb'], data['gear'])
plt.title('carbs vs gear')
plt.xlabel('Number of carburetors')
plt.ylabel('Number of forward gears')
plt.xticks(rotation='vertical',size=8)
plt.show()
```

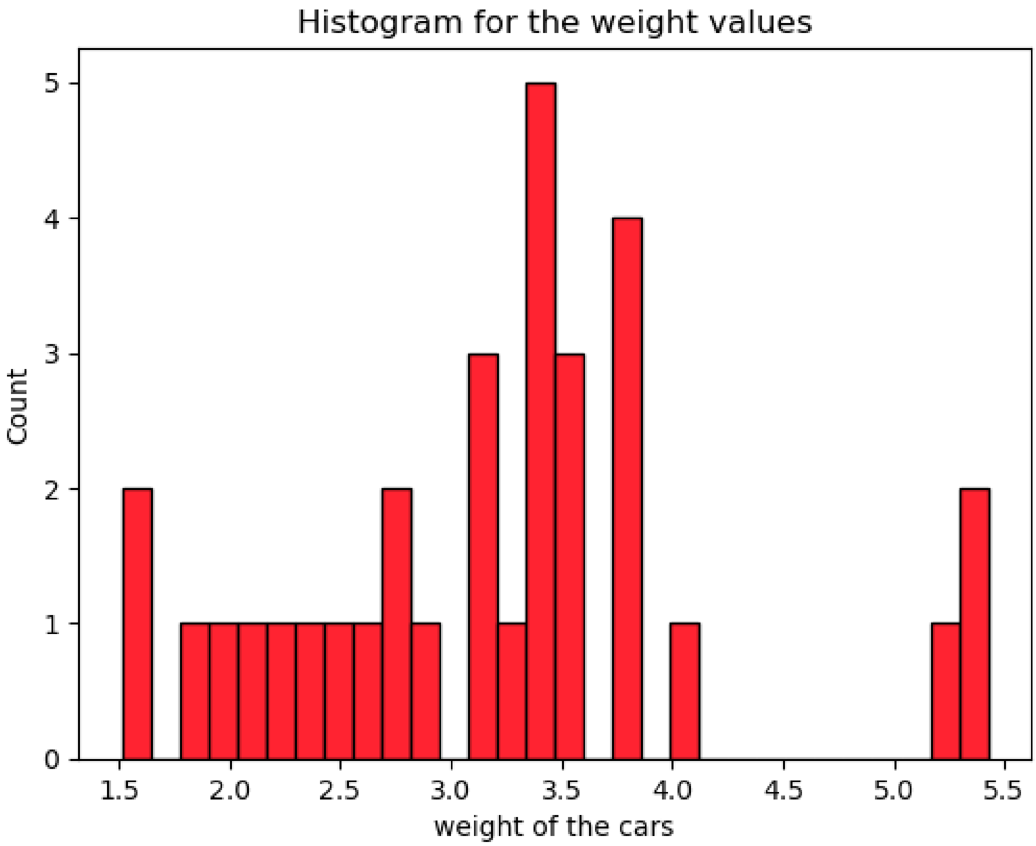


3 Module 6: Data Visualization Assignment 3

Problem Statement: You work in XYZ Corporation as a Data Analyst. Your corporation has told you to visualize the mtcars.csv dataset with various plots. Dataset Link Tasks To Be Performed: 1. Plot a histogram for the column ‘wt’ a. Map the ‘wt’ onto the x-axis b. Provide the x-axis label as ‘weight of the cars’ c. Provide the y-axis label as ‘Count’ d. Set the number of bins as 30 e. Set the title as ’Histogram fo

```
[21]: plt.hist(data['wt'],bins=30,edgecolor="#000000",color='#ff2331')
      plt.xlabel('weight of the cars')
      plt.ylabel('Count')
      plt.title('Histogram for the weight values')
```

```
[21]: Text(0.5, 1.0, 'Histogram for the weight values')
```



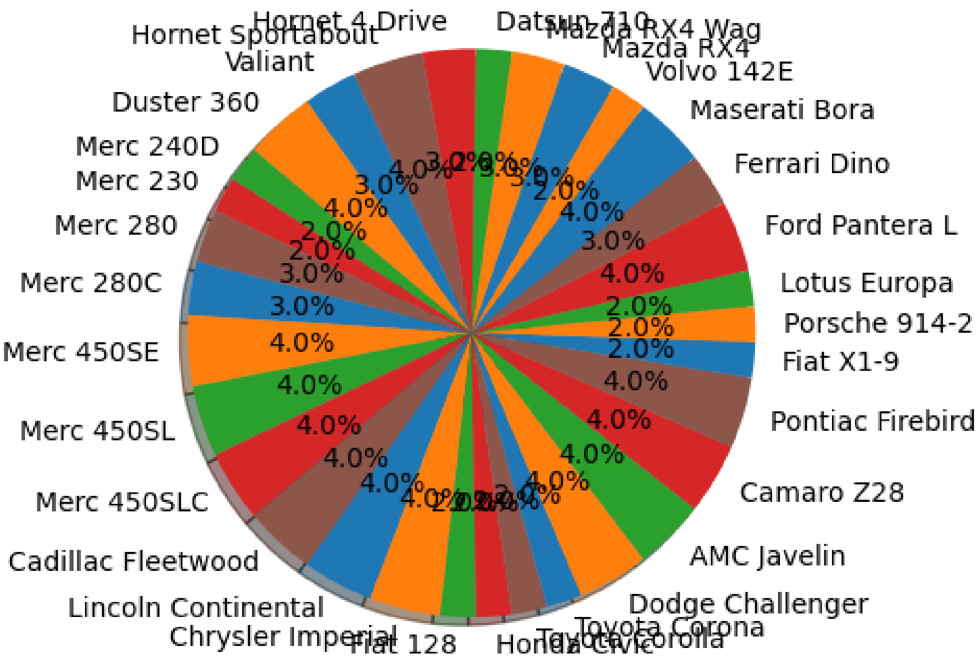
4 Module 6: Data Visualization Assignment 4

Problem Statement: You work in XYZ Corporation as a Data Analyst. Your corporation has told you to visualize the mtcars.csv dataset with various plots. Dataset Link Tasks To Be Performed: 1. Plot a pie chart for columns: 'cyl' and 'model' form the mtcars.csv data frame.

[22]:

```
models_cars = data["model"]
cyl_data= data["cyl"]
colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#8c564b"]

plt.pie(cyl_data , labels=models_cars , colors=colors,
autopct='%1.1f%%', shadow=True, startangle=60)
plt.show()
```



5 Module 6: Data Visualization Assignment 5

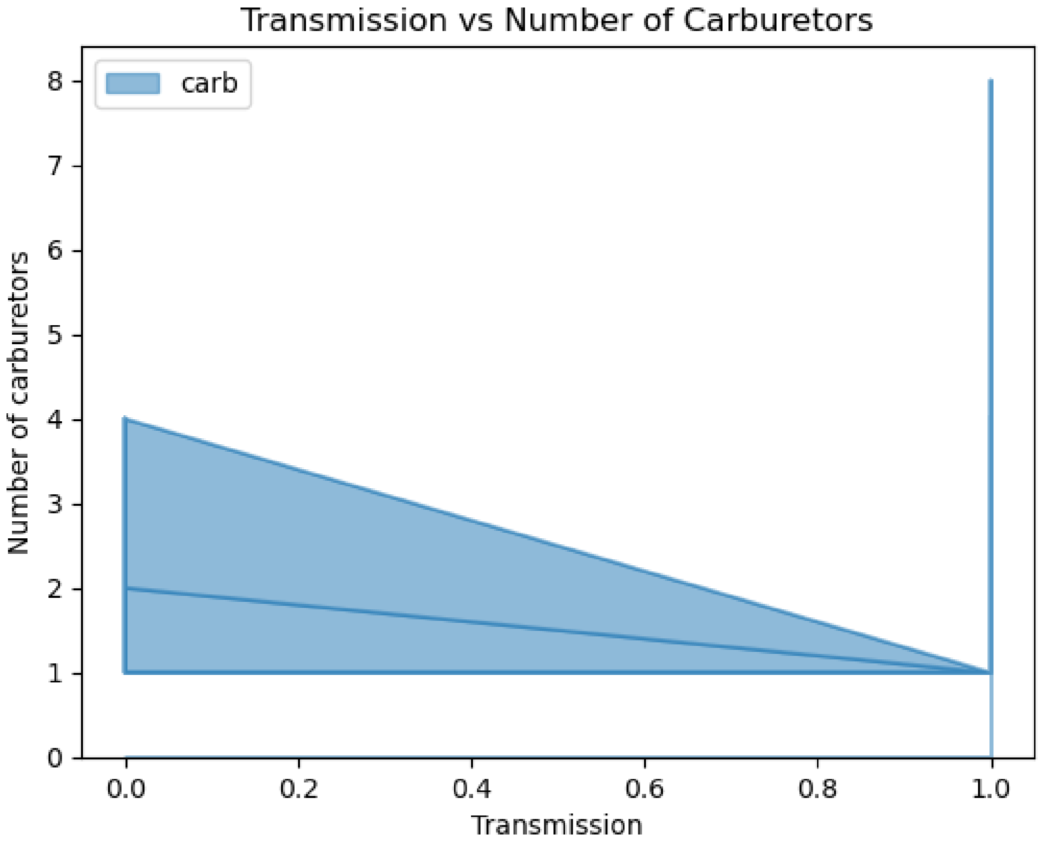
Problem Statement: You work in XYZ Corporation as a Data Analyst. Your corporation has told you to visualize the mtcars.csv dataset with various plots. Dataset Link Tasks To Be Performed: 1. Plot the area chart for the columns: ‘am’ and ‘carb’ a. Set the ‘am’ on the x-axis b. Set the ‘carb’ on the y-axis c. Provide the x-axis label as Transmission d. Provide the y-axis label as Number of carburetors e. Provide the title as Transmission vs number of carburetory.

[23]:

```
data.plot.area(x='am', y='carb', stacked=False)

plt.xlabel('Transmission')
plt.ylabel('Number of carburetors')
plt.title('Transmission vs Number of Carburetors')

plt.show()
```



6 Module 6: Data Visualization Case Study

Case Study Problem Statement: Consider yourself to be Sam who is a data scientist. He has been approached by a telecom company to build some aesthetic graphs to make better sense of the customer data. Tasks To Be Performed: 1. Sam has to build a bar-plot for the ‘Contract’ column a. Set the x-axis label to be ‘Contract Type of customer’ b. Set the y-axis label to be ‘Count’ c. Set the title of the plot to be ‘Distribution of Contract’ d. Assign ‘orange’ color to all the bars 2. Sam has to build a histogram for the ‘MonthlyCharges’ column a. Set the x-axis label to be ‘Monthly Charges Incurred’ b. Set the y-axis label to be ‘Count’ c. Set the title of the plot to be ‘Distribution of Monthly Charges’ d. Assign ‘forestgreen’ color to the bins 3. Sam has to build a scatter-plot between ‘TotalCharges’ & ‘tenure’. ‘TotalCharges’ should be on the y-axis and ‘tenure’ should be on the x-axis a. Set the x-axis label to be ‘Tenure of the customer’ b. Set the y-axis label to be ‘Total chargesIncurred’ c. Set the title of the plot to be ‘Total Charges vs Tenure’ d. Assign ‘indigo’ color to the points 4. Sam has to build a box-plot between ‘MonthlyCharges’ & ‘PaymentMethod’. ‘MonthlyCharges’ should be on the y-axis and ‘PaymentMethod’ should be on the x-axis a. Set the x-axis label to be ‘Payment Method of customer’ b. Set the y-axis label to be ‘Monthly ChargesIncurred’ c. Set the title of plot to be ‘Monthly Charges vs. Payment Method’ d. Assign ‘olive’ color to the box-plots

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

```
[26]: data=pd.read_csv('Customer Churn.csv')
```

```
[27]: data.head(10)
```

```
[27]:    customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
      0  7590-VHVEG  Female              0     Yes         No       1           No
      1  5575-GNVDE    Male              0      No         No      34          Yes
      2  3668-QPYBK    Male              0      No         No       2          Yes
      3  7795-CFOCW    Male              0      No         No      45           No
      4  9237-HQITU  Female              0      No         No       2          Yes
      5  9305-CDSKC  Female              0      No         No       8          Yes
      6  1452-KIOVK    Male              0      No        Yes      22          Yes
      7  6713-OKOMC  Female              0      No         No      10           No
      8  7892-POOKP  Female              0     Yes         No      28          Yes
      9  6388-TABGU    Male              0      No        Yes      62          Yes

            MultipleLines InternetService OnlineSecurity  … DeviceProtection  \
      0  No phone service             DSL             No  …               No
      1                No             DSL            Yes  …              Yes
      2                No             DSL            Yes  …               No
      3  No phone service             DSL            Yes  …              Yes
      4                No     Fiber optic             No  …               No
      5               Yes     Fiber optic             No  …              Yes
      6               Yes     Fiber optic             No  …               No
      7  No phone service             DSL            Yes  …               No
      8               Yes     Fiber optic             No  …              Yes
      9                No             DSL            Yes  …               No

        TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
      0          No          No              No  Month-to-month              Yes
      1          No          No              No        One year               No
      2          No          No              No  Month-to-month              Yes
      3         Yes          No              No        One year               No
      4          No          No              No  Month-to-month              Yes
      5          No         Yes             Yes  Month-to-month              Yes
      6          No         Yes              No  Month-to-month              Yes
      7          No          No              No  Month-to-month               No
      8         Yes         Yes             Yes  Month-to-month              Yes
      9          No          No              No        One year               No

                     PaymentMethod MonthlyCharges  TotalCharges Churn
      0           Electronic check          29.85         29.85    No
```

```
1              Mailed check          56.95        1889.5    No
2              Mailed check          53.85        108.15   Yes
3 Bank transfer (automatic)          42.30       1840.75    No
4          Electronic check          70.70        151.65   Yes
5          Electronic check          99.65         820.5   Yes
6   Credit card (automatic)          89.10        1949.4    No
7              Mailed check          29.75         301.9    No
8          Electronic check         104.80       3046.05   Yes
9 Bank transfer (automatic)          56.15       3487.95    No

[10 rows x 21 columns]
```

[28]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

[29]:

```
data.describe()
```

[29]:

```
       SeniorCitizen       tenure  MonthlyCharges
count    7043.000000  7043.000000     7043.000000
mean        0.162147    32.371149       64.761692
```

```
std          0.368612    24.559481       30.090047
min          0.000000     0.000000       18.250000
25%          0.000000     9.000000       35.500000
50%          0.000000    29.000000       70.350000
75%          0.000000    55.000000       89.850000
max          1.000000    72.000000      118.750000
```

[ ]:

```

```

[30]:

```
data['customerID']=pd.to_numeric(data['customerID'],errors='coerce')
data['TotalCharges']=pd.to_numeric(data['TotalCharges'],errors='coerce')
```

[39]:

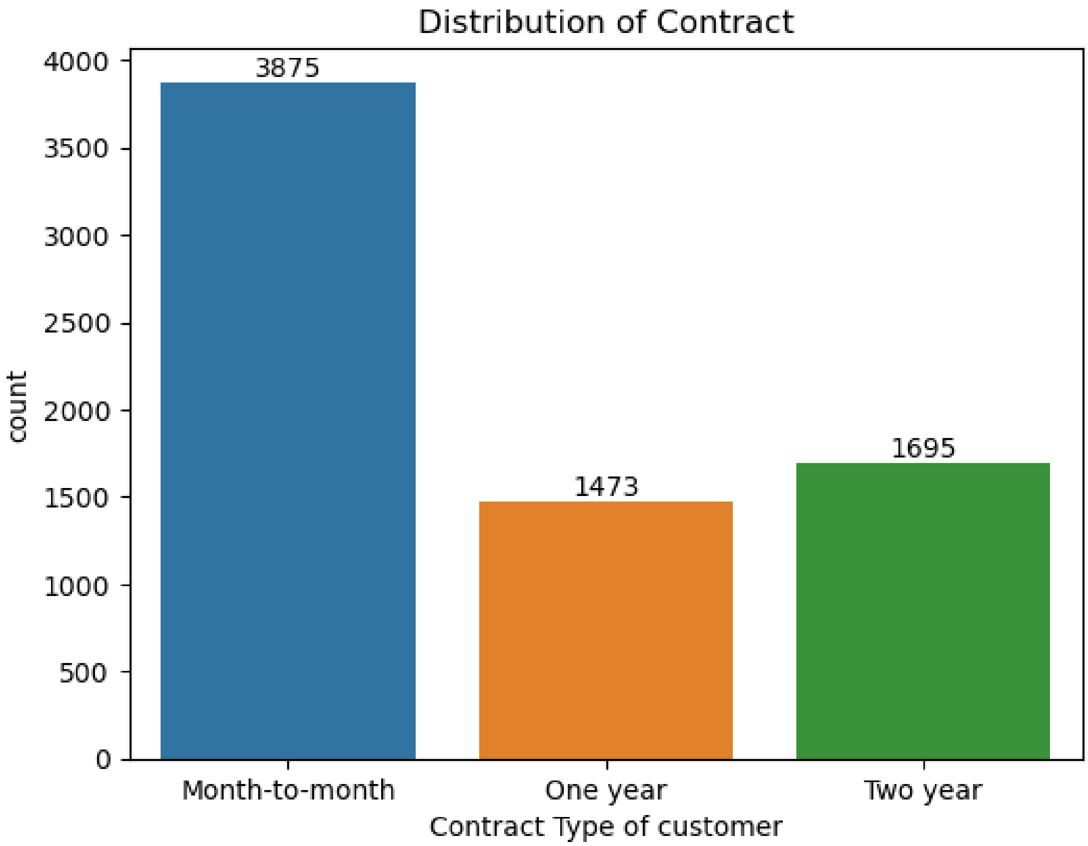
```
data.columns
```

[39]:

```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```

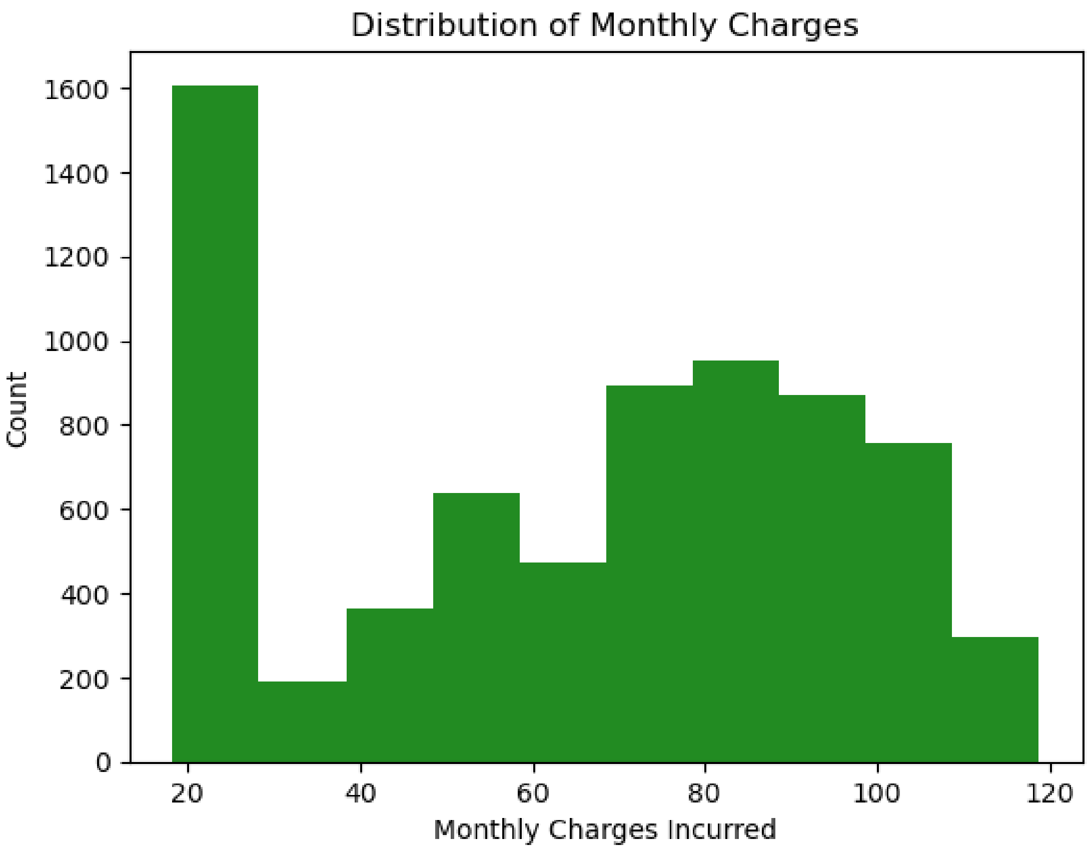
[45]:

```
##1,
ax = sns.countplot(x = 'Contract',data = data)
for bars in ax.containers:
    ax.bar_label(bars)
plt.xlabel('Contract Type of customer')
plt.ylabel('count')
plt.title('Distribution of Contract')
plt.show()
```



[46]:

```
##2,
count = [10,20,30,40,50,60,70]
plt.hist(data['MonthlyCharges'],bins=10, color='forestgreen')
plt.xlabel('Monthly Charges Incurred')
plt.ylabel('Count')
plt.title('Distribution of Monthly Charges')
plt.show()
```

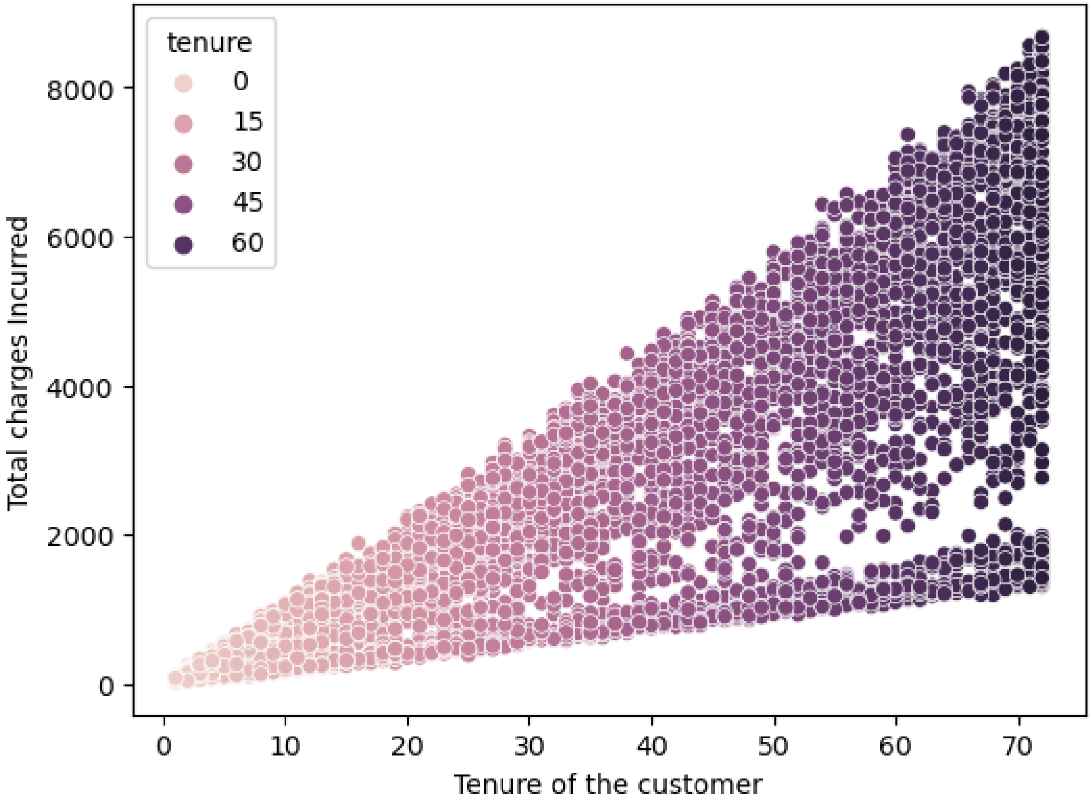


```
[55]: ##3,
      sns.scatterplot(x=data['tenure'], y=data['TotalCharges'],␣
       ↪hue='tenure',data=data)

      # Set the x-axis label to be 'Tenure of the customer'
      plt.xlabel('Tenure of the customer')

      # Set the y-axis label to be 'Total charges Incurred'
      plt.ylabel('Total charges Incurred')

      # Show the plot
      plt.show()
```

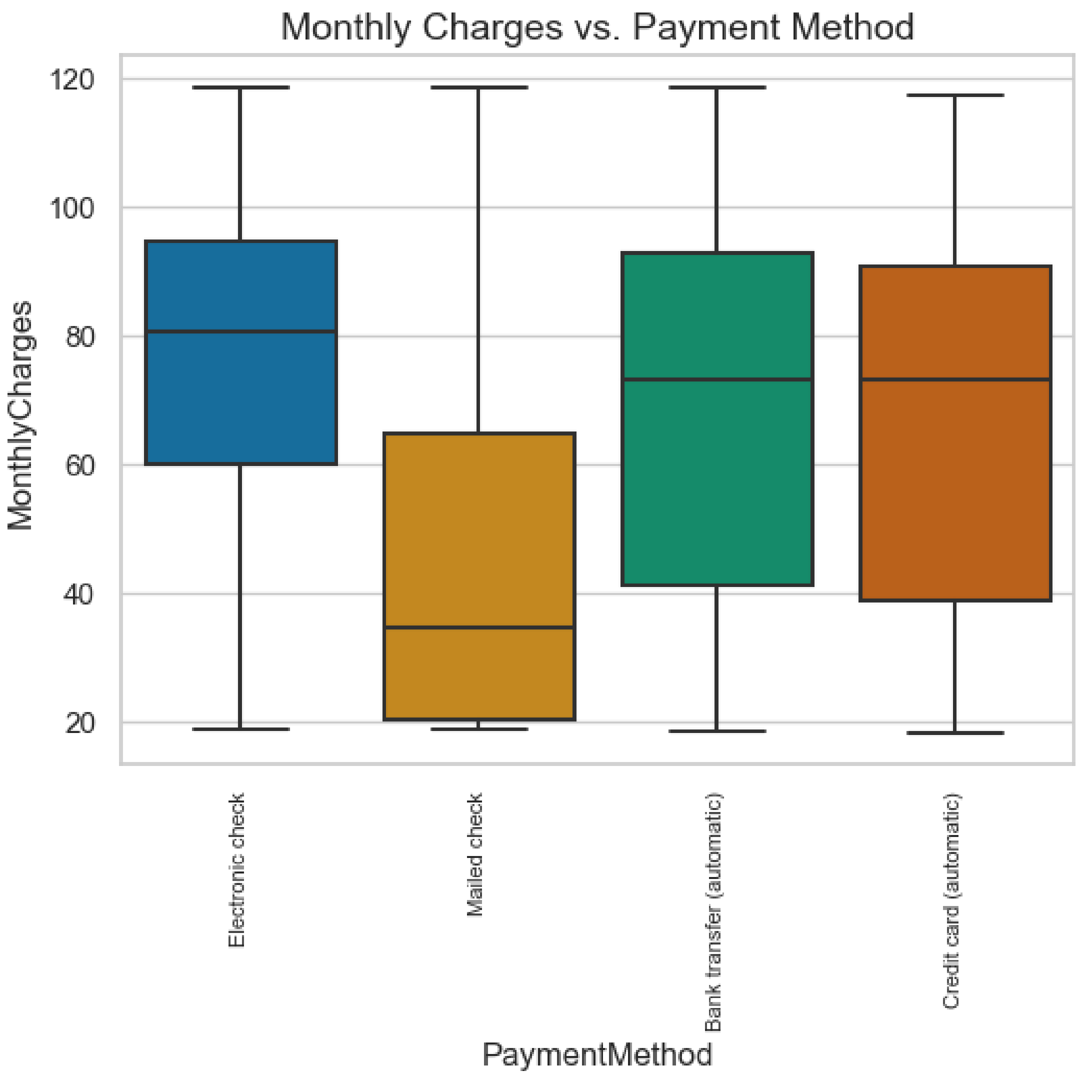


[74]:

```
```

[70]:

```
##4,
sns.boxplot(x=data["PaymentMethod"], y=data["MonthlyCharges"], data=data,␣
↪palette="colorblind")
plt.xlabel='Payment Method of customer'
plt.ylabel='Monthly Charges Incurred'
plt.title('Monthly Charges vs. Payment Method', fontsize=14)
plt.xticks(rotation='vertical',size=8)
plt.show()
```


Monthly Charges vs. Payment Method
MonthlyCharges
120
100
80
60
40
20
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)
PaymentMethod

```
[75]:
```

```
[77]:
```

```
[ ]:
```