

Ram Gurjar 0905CS181133 Data Analytics Practical File

1. Write a Python Program to Get Total Price of all FuelType and show it using a line plot with the following Style properties Generated line plot must include following Style properties: –

Line Style dotted and Line-color should be red ⓘShow legend at the upper right location. ⓘX label name = Fuel Type ⓘY label name = Price ⓘAdd a circle marker. ⓘLine marker color as red ⓘLine width should be 3

In [2]:

```
import pandas as pd
df=pd.read_csv("https://raw.githubusercontent.com/gchoi/Dataset/master/ToyotaCorolla.csv")
```

Out[2]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	13500	23	46986	Diesel	90	1	0	2000	3	1165
1	13750	23	72937	Diesel	90	1	0	2000	3	1165
2	13950	24	41711	Diesel	90	1	0	2000	3	1165
3	14950	26	48000	Diesel	90	0	0	2000	3	1165
4	13750	30	38500	Diesel	90	0	0	2000	3	1170
...
1431	7500	69	20544	Petrol	86	1	0	1300	3	1025
1432	10845	72	19000	Petrol	86	0	0	1300	3	1015
1433	8500	71	17016	Petrol	86	0	0	1300	3	1015
1434	7250	70	16916	Petrol	86	1	0	1300	3	1015
1435	6950	76	1	Petrol	110	0	0	1600	5	1114

1436 rows × 10 columns

In [3]:

```
df2=df.groupby(['FuelType']).sum()
df2=df2.reset_index(level=0)
dic={"FuelType":df2["FuelType"],"Price":df2["Price"]}
df2=pd.DataFrame(dic)
df2
```

Out[3]:

	FuelType	Price
0	CNG	160160
1	Diesel	1750656
2	Petrol	13498648

In [4]:

```
import matplotlib.pyplot as plt
plt.plot(df2["FuelType"],df2["Price"],"r--",label="Price and FuelType",linewidth=3,marker='o',markerfacecolor='k')
plt.xlabel('FuelType',size=25)
plt.ylabel('Price',size=25)
plt.legend(loc='upper left')
plt.show()
```

2. Write a Python Program to Read all product sales data and show it using a multiline plot. Firstly, Insert a column Months in a DataFrame. Display the number of FuelType sold per month for each FuelType using multiline plots. (Separate Plotline for each FuelType).

In [5]:

```
import random
mon=["Jan","Feb","march","april","may","june","july","Aug","Sep","oct","nov","dec"]
sal=[1375,1689,1767,1777,1880,1887,1889,1983,1999,2080,2221,2793]
month=[]
sales=[]
j=0
for i in range(0,len(df["Price"])):
    month.append(mon[j])
    sales.append(sal[j])
    j=(j+1)%12
df["Month"]=month
df["Sales"]=sales
df
```

Out[5]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight	Month	Sales
0	13500	23	46986	Diesel	90	1	0	2000	3	1165	Jan	1375
1	13750	23	72937	Diesel	90	1	0	2000	3	1165	Feb	1689
2	13950	24	41711	Diesel	90	1	0	2000	3	1165	march	1767
3	14950	26	48000	Diesel	90	0	0	2000	3	1165	april	1777
4	13750	30	38500	Diesel	90	0	0	2000	3	1170	may	1800
...
1431	7500	69	20544	Petrol	86	1	0	1300	3	1025	april	1777
1432	10845	72	19000	Petrol	86	0	0	1300	3	1015	may	1800
1433	8500	71	17016	Petrol	86	0	0	1300	3	1015	june	1887
1434	7250	70	16916	Petrol	86	1	0	1300	3	1015	july	1889
1435	6950	76	1	Petrol	110	0	0	1600	5	1114	Aug	1903

1436 rows × 12 columns

In [6]:

```
import matplotlib.pyplot as plt
plt.plot(df["Month"],head(12),df["Sales"].head(12),label="Diesel")
plt.plot(df["Month"].iloc[17:29],df["FuelType"].iloc[17:29],label="Petrol")
plt.plot(df["Month"].tail(),df["Price"].tail(),label="CNG")
plt.legend()
plt.show()
```

3. Write a Python Program to Read 'Petrol' sales data of each month and show it using a scatter plot Firstly, Insert a column Months in a DataFrame. Also, add a grid in the plot. gridline style should “_”.

In [7]:

```
import numpy as np
mon=["Jan","Feb","march","april","may","june","july","Aug","Sep","oct","nov","dec"]
sales=[]
month=[]
j=0
for i in range(0,len(df["Price"])):
    month.append(random.choice(mon))
    sales.append(random.randint(1880,2080))
df["Month"]=month
df
```

Out[7]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight	Month	Sales
0	13500	23	46986	Diesel	90	1	0	2000	3	1165	june	1426
1	13750	23	72937	Diesel	90	1	0	2000	3	1165	june	1803
2	13950	24	41711	Diesel	90	1	0	2000	3	1165	april	1969
3	14950	26	48000	Diesel	90	0	0	2000	3	1165	Aug	1899
4	13750	30	38500	Diesel	90	0	0	2000	3	1170	Aug	1818
...
1431	7500	69	20544	Petrol	86	1	0	1300	3	1025	Feb	1112
1432	10845	72	19000	Petrol	86	0	0	1300	3	1015	Jan	1057
1433	8500	71	17016	Petrol	86	0	0	1300	3	1015	oct	1951
1434	7250	70	16916	Petrol	86	1	0	1300	3	1015	dec	1685
1435	6950	76	1	Petrol	110	0	0	1600	5	1114	Feb	1980

1436 rows × 12 columns

In [8]:

```
dfs=df.loc[df["FuelType"]=="Petrol"]
plt.scatter(dfs["Month"].head(60),dfs["Sales"].head(60))
plt.grid(linestyle = "-.-")
plt.show()
```

4. Write a Python Program to Read 'Petrol' and 'CNG' FuelType sales data and show it using the Heatmap. The bar chart should display the number of units sold per month for each product. Add a separate bar for each product in the same chart.

In [24]:

```
import seaborn as sns
dfs=df.loc[df["FuelType"] != "Diesel"]
dic={"Price":dfs["Price"],"Sales":dfs["Sales"]*12}
dfs=pd.DataFrame(dic)
sns.heatmap(dfs.head(100))
```

Out[24]:

<AxesSubplot:>

5: Write a Python Program to Read the total Price of each month and show it using the histogram to see most common Price ranges.

In [228]:

```
plt.hist(df["Price"])
plt.show()
```

Out[228]:

6. Write a Python Program to Calculate total Price data for last year for each FuelType and show it using a Pie chart.

In [221]:

```
x=df["FuelType"].value_counts().iloc[:].values
y=[]
for i in range(len(x)):
    y.append(df["FuelType"].value_counts().iloc[:].index[i])
plt.pie(x,labels=y,autopct="%3.2f%%")
plt.show()
```

Out[221]:

7. Write a Python Program to Read 'Diesel', 'Petrol' of all months and display it using the Subplot.

In [222]:

```
fig=plt.figure(figsize=(20,5))
ax1=plt.subplot2grid((2,4),(0,0))
ax2=plt.subplot2grid((2,4),(0,1))
ax1.scatter(df["Price"],df["Age"],c='c')
ax2.hist(df["Price"],color='m')
fig.tight_layout()
```

Out[222]:

8. Write a Python Program to Read all FuelType data and show it using the stack plot.

In [223]:

```
mon=["Jan","Feb","march","april","may","june","july","Aug","Sep","oct","nov","dec"]
Petrol=[0,3,4,3,4,3,5,2,5,7]
Diesel=[2,3,4,3,2,3,2,5,7,8,2,4]
CNG=[7,5,6,3,9,7,8,5,4,6,9]
labels=["Petrol","Diesel","CNG"]
plt.stackplot(mon,Petrol,Diesel,CNG,labels=label,color=['b','c','r'])
plt.xlabel('Month',size=25)
plt.ylabel('Sales per month',size=25)
plt.legend()
plt.show()
```

Out[223]:

9. Write a R program to create and display a DataFrame from a specified dictionary data which has the index labels.

In [28]:

```
exam_data = {'name': ['Dinesh', 'Suresh', 'Rahul', 'Ravi', 'Manoj', 'Hari', 'Yatharth', 'Saurabh', 'Kapil', 'Salini'], 'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], 'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']} labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']
```

Out[28]:

DataFrame <- dataframe from R.png

	name	score	attempts	qualify
a	Dinesh	12.5	1	yes
b	Suresh	9	3	no
c	Rahul	16.5	2	yes
d	Ravi	nan	3	no
e	Manoj	9	2	no
f	Hari	20	3	yes
g	Yatharth	14.5	1	yes
h	Saurabh	nan	1	no
i	Kapil	8	2	no
j	Salini	19	1	yes

10. Write a Python program to display a summary of the basic information about a specified DataFrame which is created by Toyota.csv and its data like index, columns, non null values of each column, memory usage etc.

In [224]:

```
dfo=pd.read_csv(r"C:\Users\rang9\toyota.csv")
dfo.info()
```

Out[224]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  ---
 0   Price      1197 non-null    int64
 1   Age        1197 non-null    int64
 2   KM         1197 non-null    int64
 3   FuelType   1194 non-null    object
 4   HP         1197 non-null    int64
 5   MetColor   1197 non-null    int64
 6   Automatic  1197 non-null    int64
 7   CC         1197 non-null    int64
 8   Doors      1197 non-null    int64
 9   Weight     1197 non-null    int64
dtypes: int64(9), object(1)
memory usage: 93.6+ KB
```

11. Write a Python program to get the first 3 rows of a given DataFrame where data frame create by using dictionary. { 'name': ['Dinesh', 'Suresh', 'Rahul', 'Ravi', 'Manoj', 'Hari', 'Yatharth', 'Saurabh', 'Kapil', 'Salini'], 'runs': [125, 129, 165, np.nan, 109, 120, 145, np.nan, 118, 119], 'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1], labels = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [225]:

```
data={'name': ['Dinesh', 'Suresh', 'Rahul', 'Ravi', 'Manoj', 'Hari', 'Yatharth', 'Saurabh', 'Kapil', 'Salini'],
      'runs': [125, 129, 165, np.nan, 109, 120, 145, np.nan, 118, 119],
      'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
      'labels': ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']}
data=pd.DataFrame(data,index=labels)
print(data)
dataf.iloc[:3]
```

Out[225]:

	name	runs	attempts
A	Dinesh	125.0	1
B	Suresh	129.0	3
C	Rahul	165.0	2

12. Write a Python program to select the rows where the 'FuelType' is missing, i.e. is NaN. Hint: print(df[df['FuelType'].isnull()])

In [226]:

```
dfo[dfo["FuelType"].isnull()]
```

Out[226]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
861	10950	65	64630	NaN	110	0	0	1600	5	1070
1082	8600	76	134889	NaN	72	1	0	2000	3	1115
1196	8750	79	89739	NaN	86	1	0	1300	5	1035

13. Write a R program to for linear and multi linear regression

In [29]:

```
import PIL
PIL.Image.open('linearReg in R.png', mode='r')
```

Out[29]:

```
1 # Create the predictor and response variables
2 x = c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
3 y = c(103, 81, 56, 81, 47, 57, 76, 72, 62, 48)
4 relation = lm(y~x)
5
6 # Give the chart file a name
7 png(file = "linearRegression.png")
8
9 # Plot the chart
10 plot(x,y,col = "blue",main = "Height & Weight Regression",
11 abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in cm")
12
13
```