

```
In [1]: # tips prediction
# first initiate the dataset
import pandas as pd
df = pd.read_csv("E:/tips.csv")

In [2]: df

Out[2]:
   total_bill  tip    sex  smoker  day  time  size
0      16.99  1.01  Female    No  Sun  Dinner    2
1      10.34  1.66    Male    No  Sun  Dinner    3
2      21.01  3.50    Male    No  Sun  Dinner    3
3      23.68  3.31    Male    No  Sun  Dinner    2
4      24.59  3.61  Female    No  Sun  Dinner    4
...
239  29.03  5.92    Male    No  Sat  Dinner    3
240  27.18  2.00  Female    Yes  Sat  Dinner    2
241  22.67  2.00    Male    Yes  Sat  Dinner    2
242  17.82  1.75    Male    No  Sat  Dinner    2
243  18.78  3.00  Female    No  Thur  Dinner    2

244 rows x 7 columns

In [3]: df.head()

Out[3]:
   total_bill  tip    sex  smoker  day  time  size
0      16.99  1.01  Female    No  Sun  Dinner    2
1      10.34  1.66    Male    No  Sun  Dinner    3
2      21.01  3.50    Male    No  Sun  Dinner    3
3      23.68  3.31    Male    No  Sun  Dinner    2
4      24.59  3.61  Female    No  Sun  Dinner    4

In [4]: df.tail()

Out[4]:
   total_bill  tip    sex  smoker  day  time  size
239  29.03  5.92    Male    No  Sat  Dinner    3
240  27.18  2.00  Female    Yes  Sat  Dinner    2
241  22.67  2.00    Male    Yes  Sat  Dinner    2
242  17.82  1.75    Male    No  Sat  Dinner    2
243  18.78  3.00  Female    No  Thur  Dinner    2

In [5]: df.shape

Out[5]:
(244, 7)

In [6]: df.columns

Out[6]:
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'], dtype='object')

In [7]: df["sex"].value_counts()

Out[7]:
Male      157
Female     87
Name: sex, dtype: int64

In [8]: df["tip"].value_counts()

Out[8]:
2.00     33
3.00     23
4.00     12
5.00     10
2.50     10
...
4.34      1
1.56      1
5.20      1
2.60      1
3.75      1
Name: tip, Length: 123, dtype: int64

In [9]: df["total_bill"].value_counts()

Out[9]:
13.42      3
13.03      2
15.98      2
17.92      2
10.07      2
...
24.75      1
23.16      1
28.97      1
22.49      1
18.78      1
Name: total_bill, Length: 229, dtype: int64

In [10]: df["size"].value_counts()

Out[10]:
2      156
3       38
4       37
5        5
1         4
6         4
Name: size, dtype: int64

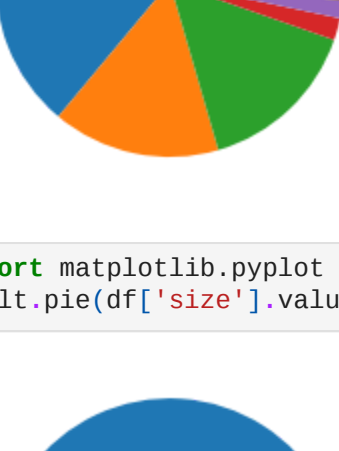
In [11]: df["size"]

Out[11]:
0      2
1      3
2      3
3      3
4      4
...
239     3
240     2
241     2
242     2
243     2
Name: size, Length: 244, dtype: int64

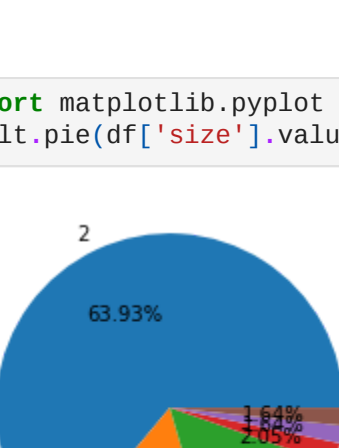
In [12]: import matplotlib.pyplot as plt
plt

Out[12]:
<module 'matplotlib.pyplot' from 'E:\anaconda\lib\site-packages\matplotlib\pyplot.py">

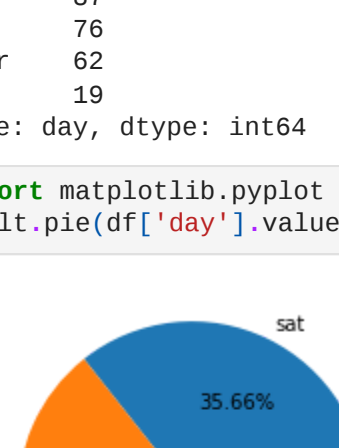
In [13]: import matplotlib.pyplot as plt
z=plt.pie(df["size"].value_counts())



In [14]: import matplotlib.pyplot as plt
z=plt.pie(df["size"].value_counts(),autopct="%0.2f%%")



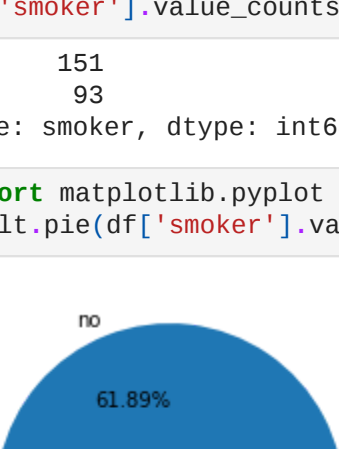
In [15]: import matplotlib.pyplot as plt
z=plt.pie(df["size"].value_counts(),labels=['2','3','4','5','1','6'],autopct="%0.2f%%")



In [17]: df["day"].value_counts()

Out[17]:
Sat      87
Sun      76
Thur     62
Fri      19
Name: day, dtype: int64

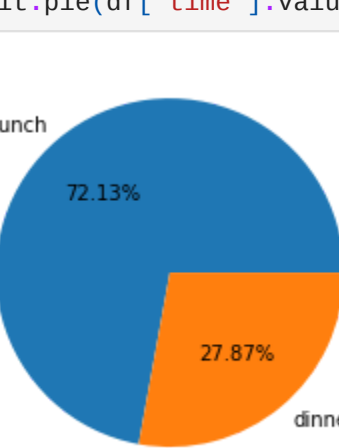
In [18]: import matplotlib.pyplot as plt
z=plt.pie(df["day"].value_counts(),labels=['sat','3sun','thur','fri'],autopct="%0.2f%%")



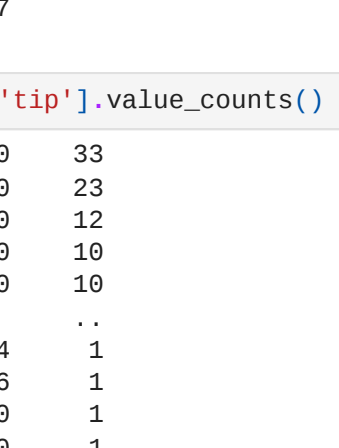
In [19]: df["smoker"].value_counts()

Out[19]:
No      151
Yes      93
Name: smoker, dtype: int64

In [20]: import matplotlib.pyplot as plt
z=plt.pie(df["smoker"].value_counts(),labels=['no','yes'],autopct="%0.2f%%")



In [21]: import matplotlib.pyplot as plt
z=plt.pie(df["time"].value_counts(),labels=['lunch','dinner'],autopct="%0.2f%%")



In [22]: df["total_bill"].max()

Out[22]:
50.81

In [23]: df["total_bill"].min()

Out[23]:
3.07

In [24]: df["tip"].value_counts()

Out[24]:
2.00     33
3.00     23
4.00     12
5.00     10
2.50     10
...
4.34      1
1.56      1
5.20      1
2.60      1
3.75      1
Name: tip, Length: 123, dtype: int64

In [25]: df["tip"].max()

Out[25]:
10.0

In [26]: df["tip"].min()

Out[26]:
1.0

In [27]: df["day"].unique()

Out[27]:
array(['Sun', 'Sat', 'Thur', 'Fri'], dtype=object)

In [28]: df["day"].value_counts()

Out[28]:
Sat      87
Sun      76
Thur     62
Fri      19
Name: day, dtype: int64

In [29]: df["time"].unique()

Out[29]:
array(['Dinner', 'Lunch'], dtype=object)

In [30]: df["time"].value_counts()

Out[30]:
Dinner    176
Lunch     68
Name: time, dtype: int64

In [31]: #which day more collection
g=df.groupby(df["day"])

In [32]: g.max()

Out[32]:
   total_bill  tip    sex  smoker  time  size
day
Fri      40.17  4.73    Male    Yes  Lunch    4
Sat      50.81 10.00    Male    Yes  Dinner    5
Sun      48.17  6.50    Male    Yes  Dinner    6
Thur     43.11  6.70    Male    Yes  Lunch    6

In [34]: #which day less collection
g.min()

Out[34]:
   total_bill  tip    sex  smoker  time  size
day
Fri       5.75  1.00  Female    No  Dinner    1
Sat       3.07  1.00  Female    No  Dinner    1
Sun       7.25  1.01  Female    No  Dinner    2
Thur       7.51  1.25  Female    No  Dinner    1

In [35]: g.max().sum()

Out[35]:
total_bill      182.26
tip              27.93
sex      MaleMaleMaleMale
smoker      YesYesYesYes
time      LunchDinnerDinnerLunch
size              21
dtype: object

In [36]: df["size"].max()

Out[36]:
6

In [37]: df.loc[df["size"]==6]

Out[37]:
   total_bill  tip    sex  smoker  day  time  size
125    29.80  4.2  Female    No  Thur  Lunch    6
141    34.30  6.7    Male    No  Thur  Lunch    6
143    27.05  5.0  Female    No  Thur  Lunch    6
156    48.17  5.0    Male    No  Sun  Dinner    6

In [38]: df["total_bill"].max()

Out[38]:
50.81

In [39]: df.loc[df["total_bill"]==50.81]

Out[39]:
   total_bill  tip    sex  smoker  day  time  size
170    50.81 10.0  Male    Yes  Sat  Dinner    3

In [40]: df["tip"].max()

Out[40]:
10.0

In [41]: df.loc[df["tip"]==10.0]

Out[41]:
   total_bill  tip    sex  smoker  day  time  size
170    50.81 10.0  Male    Yes  Sat  Dinner    3

In [42]: # only female data
df.loc[df["sex"]=="Female"]

Out[42]:
   total_bill  tip    sex  smoker  day  time  size
0      16.99  1.01  Female    No  Sun  Dinner    2
4      24.59  3.61  Female    No  Sun  Dinner    4
11     35.26  5.00  Female    No  Sun  Dinner    4
14     14.83  3.02  Female    No  Sun  Dinner    2
16     10.33  1.67  Female    No  Sun  Dinner    3
...
226    10.09  2.00  Female    Yes  Fri  Lunch    2
229    22.12  2.88  Female    Yes  Sat  Dinner    2
238    35.83  4.67  Female    No  Sat  Dinner    3
240    27.18  2.00  Female    Yes  Sat  Dinner    2
243    18.78  3.00  Female    No  Thur  Dinner    2

87 rows x 7 columns

In [43]: # only male data
df.loc[df["sex"]=="Male"]

Out[43]:
   total_bill  tip    sex  smoker  day  time  size
1      10.34  1.66    Male    No  Sun  Dinner    3
2      21.01  3.50    Male    No  Sun  Dinner    3
3      23.68  3.31    Male    No  Sun  Dinner    2
5      25.29  4.71    Male    No  Sun  Dinner    4
6       8.77  2.00    Male    No  Sun  Dinner    2
...
236    12.80  1.00    Male    Yes  Sat  Dinner    2
237    32.83  1.17    Male    Yes  Sat  Dinner    2
239    29.03  5.92    Male    No  Sat  Dinner    3
241    22.67  2.00    Male    Yes  Sat  Dinner    2
242    17.82  1.75    Male    No  Sat  Dinner    2

157 rows x 7 columns

In [44]: # machine learning
# convert the string values to numeric
from sklearn.preprocessing import LabelEncoder
le1=LabelEncoder()
le2=LabelEncoder()
le3=LabelEncoder()
le4=LabelEncoder()
df["sex"]=le1.fit_transform(df["sex"])
df["smoker"]=le2.fit_transform(df["smoker"])
df["day"]=le3.fit_transform(df["day"])
df["time"]=le4.fit_transform(df["time"])

In [45]: df

Out[45]:
   total_bill  tip    sex  smoker  day  time  size
0      16.99  1.01      0      0      2      0      2
1      10.34  1.66      1      0      2      0      3
2      21.01  3.50      1      0      2      0      3
3      23.68  3.31      1      0      2      0      2
4      24.59  3.61      0      0      2      0      4
...
239    29.03  5.92      1      0      1      0      3
240    27.18  2.00      0      1      1      0      2
241    22.67  2.00      1      1      1      0      2
242    17.82  1.75      1      0      1      0      2
243    18.78  3.00      0      0      3      0      2

244 rows x 7 columns

In [46]: # separate input and output
X=df.drop(columns="tip")
Y=df["tip"]

In [47]: # split the data for training and testing
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2)

In [48]: X_train

Out[48]:
   total_bill  sex  smoker  day  time  size
130     19.08      1      0      3      1      2
109     14.31      0      1      1      0      2
126     11.38      0      0      3      1      2
197     43.11      0      1      3      1      4
155     29.85      0      0      2      0      5
...
44      30.40      1      0      2      0      4
225     16.27      0      1      0      1      2
104     20.92      0      0      1      0      2
214     28.17      0      1      1      0      3
147     11.87      0      0      3      1      2

195 rows x 6 columns

In [49]: Y_train

Out[49]:
130     1.50
109     4.00
126     2.00
197     5.00
155     5.14
...
44     10.00
225     2.50
104     4.00
214     6.50
147     1.63
Name: tip, Length: 195, dtype: float64

In [50]: # create a ML model
from sklearn.neighbors import KNeighborsRegressor
k=KNeighborsRegressor(n_neighbors=5)
# train the model
K.fit(X_train,Y_train)

In [50]: KNeighborsRegressor()

In [51]: # test the model
Y_pred=K.predict(X_test)

In [52]: Y_pred

Out[52]:
array([2.562, 2.408, 2.686, 3.732, 1.462, 3.538, 2.02, 4.68, 2.556,
       2.116, 3.538, 2.686, 3.844, 4.454, 3.286, 5. , 4.088, 1.712,
       1.684, 2.676, 5. , 2.622, 2.066, 2.75, 4.024, 2.922, 1.77,
       4.084, 3.29, 2.448, 2.44, 2.064, 4.77, 3.386, 3.8 , 3.592,
       2.456, 2.684, 3.534, 1.93, 4.074, 2.238, 3.386, 1.684, 3.098,
       1.462, 4.094, 3.108, 3.214])

In [53]: Y_test.values

Out[53]:
array([ 2.47,  3. ,  3.51,  5. ,  1.25,  3.61,  1.5,  2. ,  2. ,
        1.8,  2. ,  2.3,  3.11,  5. ,  3.76, 10. ,  2.55,  2. ,
        1.5,  3.35,  6.73,  4. ,  2.5,  3.15,  2.92,  3.5,  1.07,
        3. ,  3. ,  4.3,  4. ,  3.5,  6. ,  6.5,  3.48,  4. ,
        3. ,  1.57,  3.18,  2.31,  4.71,  2.5,  2.71,  1.48,  3. ,
        1.45,  2.56,  4.29,  1.75])

In [54]: # find mean squared error
from sklearn.metrics import mean_squared_error
mse=mean_squared_error(Y_test,Y_pred)
mse

Out[54]:
1.525363591867345

In [55]: # r 2 score
from sklearn.metrics import r2_score
rse=r2_score(Y_test,Y_pred)
rse

Out[55]:
0.48779561822846966

In [56]: # predict for a new data
K.predict([[20,0,1,2,1,3]])

E:\anaconda\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but KNeighborsRegressor was fitted with feature names
  warnings.warn(
array([3.584])

In [58]: # create a new ML model
from sklearn.ensemble import RandomForestRegressor
R=RandomForestRegressor()
# train the model
R.fit(X_train,Y_train)

Out[58]:
RandomForestRegressor()

In [64]: # test the model
Y_pred_rf=R.predict(X_test)

In [65]: Y_pred

Out[65]:
array([2.562, 2.408, 2.686, 3.732, 1.462, 3.538, 2.02, 4.68, 2.556,
       2.116, 3.538, 2.686, 3.844, 4.454, 3.286, 5. , 4.088, 1.712,
       1.684, 2.676, 5. , 2.622, 2.066, 2.75, 4.024, 2.922, 1.77,
       4.084, 3.29, 2.448, 2.44, 2.064, 4.77, 3.386, 3.8 , 3.592,
       2.456, 2.684, 3.108, 3.214])

In [66]: Y_test.values

Out[66]:
array([ 2.47,  3. ,  3.51,  5. ,  1.25,  3.61,  1.5,  2. ,  2. ,
        1.8,  2. ,  2.3,  3.11,  5. ,  3.76, 10. ,  2.55,  2. ,
        1.5,  3.35,  6.73,  4. ,  2.5,  3.15,  2.92,  3.5,  1.07,
        3. ,  3. ,  4.3,  4. ,  3.5,  6. ,  6.5,  3.48,  4. ,
        3. ,  1.57,  3.18,  2.31,  4.71,  2.5,  2.71,  1.48,  3. ,
        1.45,  2.56,  4.29,  1.75])

In [68]: # find mean squared error
from sklearn.metrics import mean_squared_error
mse_rf=mean_squared_error(Y_test,Y_pred_rf)
mse_rf

Out[68]:
1.0817142291836732

In [69]: # r 2 score
from sklearn.metrics import r2_score
rse_rf=r2_score(Y_test,Y_pred_rf)
rse_rf

Out[69]:
0.588972284817285

In [ ]:
```