

186-Mayuri

Yadav

May 10, 2023

```
[1]: file=open('stud_info.csv','r')
info_dataset=[]
while True:
    data=file.readline()
    if data:
        info_dataset.append(data.replace("\n", "").split(','))
    else:
        break
print(info_dataset)
```

```
[['Roll No', 'name', 'Gender', 'DOB'], ['1', 'John', 'Male', '05-04-1988'],
['2', 'Mayur', 'Male', '04-05-1987'], ['3', 'Mangesh', 'Male', '25-05-1989'],
['4', 'Jessica', 'Female', '12-08-1990'], ['5', 'Jennifer', 'Female',
'02-09-1989'], ['6', 'Ramesh', 'Male', '03-09-1989'], ['7', 'Suresh', 'Male',
'04-09-1990'], ['8', 'Ganesh', 'Male', '05-10-1989'], ['9', 'Komal', 'Female',
'06-09-1989'], ['10', 'Mayuri', 'Female', '07-02-1988']]
```

[2]:

```
R
o
l
l
```

[3]: for row in

```
info_dataset[1:]:
RollNo.append(row
[0])
```

[4]:

```
p
r
i
```

```
['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
['John', 'Mayur', 'Mangesh', 'Jessica', 'Jennifer', 'Ramesh', 'Suresh',
'Ganesh', 'Komal', 'Mayuri']
['Male', 'Male', 'Male', 'Female', 'Female', 'Male', 'Male', 'Male', 'Female',
'Female']
```

```

'05-04-1988', '04-05-1987', '25-05-1989', '12-08-1990', '02-09-1989',
'03-09-1989', '04-09-1990', '05-10-1989', '06-09-1989', '07-02-1988']

```

```

[5]: file=open('student_marks.csv', 'r')
marks_dataset=[]
while True:
    data=file.readline()
    if data:
        marks_dataset.append(data.replace("\n",

```

```

[['Roll', 'Maths', 'Physics', 'Chemistry', 'Total', 'Percentage'], ['1', '55',
'45', '56', '156', '52.00'], ['2', '75', '55', '55', '185', '61.67'], ['3',
'25', '54', '89', '168', '56.00'], ['4', '78', '55', '86', '219', '73.00'],
['5', '58', '96', '78', '232', '77.33'], ['6', '88', '78', '58', '224',
'74.67'], ['7', '56', '89', '69', '214', '71.33'], ['8', '54', '55', '88',
'197', '65.67'], ['9', '46', '66', '65', '177', '59.00'], ['10', '89', '87',
'54', '230', '76.67']]

```

```

[6]: M
a
t
h

```

```

[7]: for row in
marks_dataset[1:]
:
Maths.append(row[
1])

```

```

[8]: print(Maths)
print(Physics)

```

```

['55', '75', '25', '78', '58', '88', '56', '54', '46', '89']
['45', '55', '54', '55', '96', '78', '89', '55', '66', '87']
['56', '55', '89', '86', '78', '58', '69', '88', '65', '54']
['156', '185', '168', '219', '232', '224', '214', '197', '177', '230']
['52.00', '61.67', '56.00', '73.00', '77.33', '74.67', '71.33', '65.67',
'59.00', '76.67']

```

```
[9]: file=open('stud_placement.csv','r')
placement_dataset=[]
while True:
    data=file.readline()
    if data:
        placement_dataset.append(data.replace("\n",
        ""))
```

```
[['Roll No', 'Company', 'JobRole', 'Package'], ['1', 'Infosys', 'Data Analyst', '10.2'], ['2', 'TCS', 'Java Developer', '9.6'], ['3', 'TCS', 'Data Scientist', '12.60'], ['4', 'Infosys', 'Data Analyst', '10.2'], ['5', 'Oracle', 'Java Developer', '9.6'], ['6', 'Oracle', 'Data Scientist', '12.60'], ['7', 'TCS', 'Tester', '6.50'], ['8', 'Infosys', 'Tester', '6.51'], ['9', 'Mindtree', 'Database Admin', '8.30'], ['10', 'Mindtree', 'Database Admin', '8.31']]
```

```
[10]: C
o
m
```

```
[11]: for row in
        placement_dataset
        [1:]:
```

```
[12]: p
r
i
```

```
['Infosys', 'TCS', 'TCS', 'Infosys', 'Oracle', 'Oracle', 'TCS', 'Infosys', 'Mindtree', 'Mindtree']
['Data Analyst', 'Java Developer', 'Data Scientist', 'Data Analyst', 'Java Developer', 'Data Scientist', 'Tester', 'Tester', 'Database Admin', 'Database Admin']
['10.2', '9.6', '12.60', '10.2', '9.6', '12.60', '6.50', '6.51', '8.30', '8.31']
```

```
[14]: studentdata=[]
studentdata.append(RollNo)
studentdata.append(Name)
studentdata.append(Gender)
studentdata.append(DO
```

```

studentdata.append(Company)
studentdata.append(JobRole)
studentdata.append(Package)
print(studentdata)

```

```

[['1', '2', '3', '4', '5', '6', '7', '8', '9', '10'], ['John', 'Mayur',
'Mangesh', 'Jessica', 'Jennifer', 'Ramesh', 'Suresh', 'Ganesh', 'Komal',
'Mayuri'], ['Male', 'Male', 'Male', 'Female', 'Female', 'Male', 'Male', 'Male',
'Female', 'Female'], ['05-04-1988', '04-05-1987', '25-05-1989', '12-08-1990',
'02-09-1989', '03-09-1989', '04-09-1990', '05-10-1989', '06-09-1989',
'07-02-1988'], ['55', '75', '25', '78', '58', '88', '56', '54', '46', '89'],
['45', '55', '54', '55', '96', '78', '89', '55', '66', '87'], ['56', '55', '89',
'86', '78', '58', '69', '88', '65', '54'], ['156', '185', '168', '219', '232',
'224', '214', '197', '177', '230'], ['52.00', '61.67', '56.00', '73.00',
'77.33', '74.67', '71.33', '65.67', '59.00', '76.67'], ['Infosys', 'TCS', 'TCS',
'Infosys', 'Oracle', 'Oracle', 'TCS', 'Infosys', 'Mindtree', 'Mindtree'], ['Data
Analyst', 'Java Developer', 'Data Scientist', 'Data Analyst', 'Java Developer',
'Data Scientist', 'Tester', 'Tester', 'Database Admin', 'Database Admin'],
['10.2', '9.6', '12.60', '10.2', '9.6', '12.60', '6.50', '6.51', '8.30',
'8.31']]

```

```
[15]: fw=open("StudentDetails.csv", "w")
```

```
[16]: data_to_write=[]
      for i in
          range(len(student
data[0])):
          row=list()
          for j in
              range(len(stu

```

```

['1, John, Male, 05-04-1988, 55, 45, 56, 156, 52.00, Infosys, Data Analyst, 10.2, \n']
['1, John, Male, 05-04-1988, 55, 45, 56, 156, 52.00, Infosys, Data Analyst, 10.2, \n',
'2, Mayur, Male, 04-05-1987, 75, 55, 55, 185, 61.67, TCS, Java Developer, 9.6, \n']
['1, John, Male, 05-04-1988, 55, 45, 56, 156, 52.00, Infosys, Data Analyst, 10.2, \n',
'2, Mayur, Male, 04-05-1987, 75, 55, 55, 185, 61.67, TCS, Java Developer, 9.6, \n',
'3, Mangesh, Male, 25-05-1989, 25, 54, 89, 168, 56.00, TCS, Data Scientist, 12.60, \n']
['1, John, Male, 05-04-1988, 55, 45, 56, 156, 52.00, Infosys, Data Analyst, 10.2, \n',
'2, Mayur, Male, 04-05-1987, 75, 55, 55, 185, 61.67, TCS, Java Developer, 9.6, \n',
'3, Mangesh, Male, 25-05-1989, 25, 54, 89, 168, 56.00, TCS, Data Scientist, 12.60, \n',
'4, Jessica, Female, 12-08-1990, 78, 55, 86, 219, 73.00, Infosys, Data Analyst, 10.2, \n']
['1, John, Male, 05-04-1988, 55, 45, 56, 156, 52.00, Infosys, Data Analyst, 10.2, \n',
'2, Mayur, Male, 04-05-1987, 75, 55, 55, 185, 61.67, TCS, Java Developer, 9.6, \n',
'3, Mangesh, Male, 25-05-1989, 25, 54, 89, 168, 56.00, TCS, Data Scientist, 12.60, \n',
'4, Jessica, Female, 12-08-1990, 78, 55, 86, 219, 73.00, Infosys, Data Analyst, 10.2, \n',

```

```
' 5, Jennifer, Female, 02-09-1989, 58, 96, 78, 232, 77. 33, Oracle, Java Developer, 9. 6, \n' ]
[' 1, John, Male, 05-04-1988, 55, 45, 56, 156, 52. 00, Infosys, Data Analyst, 10. 2, \n',
' 2, Mayur, Male, 04-05-1987, 75, 55, 55, 185, 61. 67, TCS, Java Developer, 9. 6, \n',
' 3, Mangesh, Male, 25-05-1989, 25, 54, 89, 168, 56. 00, TCS, Data Scientist, 12. 60, \n',
' 4, Jessica, Female, 12-08-1990, 78, 55, 86, 219, 73. 00, Infosys, Data Analyst, 10. 2, \n',
' 5, Jennifer, Female, 02-09-1989, 58, 96, 78, 232, 77. 33, Oracle, Java Developer, 9. 6, \n',
' 6, Ramesh, Male, 03-09-1989, 88, 78, 58, 224, 74. 67, Oracle, Data Scientist, 12. 60, \n' ]
[' 1, John, Male, 05-04-1988, 55, 45, 56, 156, 52. 00, Infosys, Data Analyst, 10. 2, \n',
' 2, Mayur, Male, 04-05-1987, 75, 55, 55, 185, 61. 67, TCS, Java Developer, 9. 6, \n',
' 3, Mangesh, Male, 25-05-1989, 25, 54, 89, 168, 56. 00, TCS, Data Scientist, 12. 60, \n',
' 4, Jessica, Female, 12-08-1990, 78, 55, 86, 219, 73. 00, Infosys, Data Analyst, 10. 2, \n',
' 5, Jennifer, Female, 02-09-1989, 58, 96, 78, 232, 77. 33, Oracle, Java Developer, 9. 6, \n',
' 6, Ramesh, Male, 03-09-1989, 88, 78, 58, 224, 74. 67, Oracle, Data Scientist, 12. 60, \n',
' 7, Suresh, Male, 04-09-1990, 56, 89, 69, 214, 71. 33, TCS, Tester, 6. 50, \n' ]
[' 1, John, Male, 05-04-1988, 55, 45, 56, 156, 52. 00, Infosys, Data Analyst, 10. 2, \n',
' 2, Mayur, Male, 04-05-1987, 75, 55, 55, 185, 61. 67, TCS, Java Developer, 9. 6, \n',
' 3, Mangesh, Male, 25-05-1989, 25, 54, 89, 168, 56. 00, TCS, Data Scientist, 12. 60, \n',
' 4, Jessica, Female, 12-08-1990, 78, 55, 86, 219, 73. 00, Infosys, Data Analyst, 10. 2, \n',
' 5, Jennifer, Female, 02-09-1989, 58, 96, 78, 232, 77. 33, Oracle, Java Developer, 9. 6, \n',
' 6, Ramesh, Male, 03-09-1989, 88, 78, 58, 224, 74. 67, Oracle, Data Scientist, 12. 60, \n',
' 7, Suresh, Male, 04-09-1990, 56, 89, 69, 214, 71. 33, TCS, Tester, 6. 50, \n',
' 8, Ganesh, Male, 05-10-1989, 54, 55, 88, 197, 65. 67, Infosys, Tester, 6. 51, \n' ]
[' 1, John, Male, 05-04-1988, 55, 45, 56, 156, 52. 00, Infosys, Data Analyst, 10. 2, \n',
' 2, Mayur, Male, 04-05-1987, 75, 55, 55, 185, 61. 67, TCS, Java Developer, 9. 6, \n',
' 3, Mangesh, Male, 25-05-1989, 25, 54, 89, 168, 56. 00, TCS, Data Scientist, 12. 60, \n',
' 4, Jessica, Female, 12-08-1990, 78, 55, 86, 219, 73. 00, Infosys, Data Analyst, 10. 2, \n',
' 5, Jennifer, Female, 02-09-1989, 58, 96, 78, 232, 77. 33, Oracle, Java Developer, 9. 6, \n',
' 6, Ramesh, Male, 03-09-1989, 88, 78, 58, 224, 74. 67, Oracle, Data Scientist, 12. 60, \n',
' 7, Suresh, Male, 04-09-1990, 56, 89, 69, 214, 71. 33, TCS, Tester, 6. 50, \n',
' 8, Ganesh, Male, 05-10-1989, 54, 55, 88, 197, 65. 67, Infosys, Tester, 6. 51, \n',
' 9, Komal, Female, 06-09-1989, 46, 66, 65, 177, 59. 00, Mindtree, Database Admin, 8. 30, \n' ]
[' 1, John, Male, 05-04-1988, 55, 45, 56, 156, 52. 00, Infosys, Data Analyst, 10. 2, \n',
' 2, Mayur, Male, 04-05-1987, 75, 55, 55, 185, 61. 67, TCS, Java Developer, 9. 6, \n',
' 3, Mangesh, Male, 25-05-1989, 25, 54, 89, 168, 56. 00, TCS, Data Scientist, 12. 60, \n',
' 4, Jessica, Female, 12-08-1990, 78, 55, 86, 219, 73. 00, Infosys, Data Analyst, 10. 2, \n',
' 5, Jennifer, Female, 02-09-1989, 58, 96, 78, 232, 77. 33, Oracle, Java Developer, 9. 6, \n',
' 6, Ramesh, Male, 03-09-1989, 88, 78, 58, 224, 74. 67, Oracle, Data Scientist, 12. 60, \n',
' 7, Suresh, Male, 04-09-1990, 56, 89, 69, 214, 71. 33, TCS, Tester, 6. 50, \n',
' 8, Ganesh, Male, 05-10-1989, 54, 55, 88, 197, 65. 67, Infosys, Tester, 6. 51, \n',
' 9, Komal, Female, 06-09-1989, 46, 66, 65, 177, 59. 00, Mindtree, Database Admin, 8. 30, \n',
' 10, Mayuri, Female, 07-02-1988, 89, 87, 54, 230, 76. 67, Mindtree, Database
Admin, 8. 31, \n' ]
```

```
[17]: fw.writelines(data_to_write)
```

```
[18]: fw.close()
```

```
[19]: print("Math
Marks=", Maths)
print("Phyics
Marks=", Physics)
print("Chemistry
Marks=", Chemistry)
math=[int(i) for i in
Maths]
physics=[int(i) for i
in Physics]
```

```
Math Marks= ['55', '75', '25', '78', '58', '88', '56', '54', '46', '89']
Phyics Marks= ['45', '55', '54', '55', '96', '78', '89', '55', '66', '87']
Chemistry Marks= ['56', '55', '89', '86', '78', '58', '69', '88', '65', '54']
Sum of Marks= [156, 185, 168, 219, 232, 224, 214, 197, 177, 230]
Average Marks= [156, 185, 168, 219, 232, 224, 214, 197, 177, 230]
```

```
[20]: print("Maximum Marks=", max(avg))
```

```
Maximum Marks= 232
```

```
[21]: print("Minimum Marks=", min(avg))
```

```
Minimum Marks= 156
```

```
[22]: print("Total No of Student=", len(studentdata[0]))
```

```
Total No of Student= 10
```

```
[23]: per=[]
for i in range(len(sum_of_marks)):
    per.append(round((100*sum_of_marks[
i]/(sum_of_marks[0]))))
```

```
Percentage= [57.78, 68.52, 62.22, 81.11, 85.93, 82.96, 79.26, 72.96, 65.56,
85.19]
```

## assignment2

May 27, 2023

```
[ ]: product_details=[]
supplier_details=dict()
customer_details=[]
gender={}
fp1= open("sales.csv","r")
data=fp1.readline()
while(True):
    data=fp1.readline()
    if not data:
        break
    print(data)
    data= data.replace("\n","")
    temp= data.split(",")
    product_details.append(temp[1])
    customer_details.append(temp[3])
    supplier_details.update({temp[0]:temp[2]})
    gender.update({temp[3]:temp[4]})
```

P00001,Lenovo laptop,Raka Ele.,Kaustobh Mahajan,male

P00002,Samsung Laptop,Vijay Sales,Siddhi kivale,female

P00003,Realmi 10pro,Gada Ele.,Sanket Kandalkar,male

P00004,Oppo f21,Surya Ele.,Yash mali,male

P00005,Lenovo laptop,Raka Ele.,Yash Bagul,male

P00006,Samsung M31,Gada Ele.,Siddhi kivale,female

P00007,LG TV 32\*,Vijay Sales,Sanket Kandalkar,male

P00008,Oppo f21,Surya Ele.,Kaustobh Mahajan,male

P00009,Lenovo laptop,Raka Ele.,Yash mali,male

P00010,Samsung M31,Gada Ele.,Siddhi kivale,female

P00011,LG TV 32\*,Surya Ele.,Sanket Kandalkar,male

P00012,Lenovo laptop,Raka Ele.,Kaustobh Mahajan,male

P00013,Samsung M31,Surya Ele.,Yash mali,male

P00014,Realmi 10pro,Raka Ele.,Siddhi kivale,female

P00015,Lenovo laptop,Gada Ele.,Tanuja Mali,female

P00016,Oppo f21,Vijay Sales,Kaustobh Mahajan,male

P00017,LG TV 32\*,Deshmukh Sales,Sanket Kandalkar,male

P00018,Lenovo laptop,Raka Ele.,Siddhi kivale,female

P00019,Samsung M21,Deshmukh Sales,Kaustobh Mahajan,male

P00020,LG TV 32\*,Gada Ele.,Yash mali,male

```
[ ]: fp1.close()
```

```
[ ]: customer_details= tuple(customer_details)
print(type(customer_details))
print("\nproduct_details\n",product_details,end='')
print("\ncustomer_details\n",customer_details,end='')
print("\nsupplier_details\n",supplier_details,end='')
print("\ngender\n",gender,end='')
```

```
<class 'tuple'>
```

```
product_details
```

```
['Lenovo laptop', 'Samsung Laptop', 'Realmi 10pro', 'Oppo f21', 'Lenovo
laptop', 'Samsung M31', 'LG TV 32*', 'Oppo f21', 'Lenovo laptop', 'Samsung M31',
'LG TV 32*', 'Lenovo laptop', 'Samsung M31', 'Realmi 10pro', 'Lenovo laptop',
'Oppo f21', 'LG TV 32*', 'Lenovo laptop', 'Samsung M21', 'LG TV 32*']
```

```
customer_details
```

```
('Kaustobh Mahajan', 'Siddhi kivale', 'Sanket Kandalkar', 'Yash mali', 'Yash
Bagul', 'Siddhi kivale', 'Sanket Kandalkar', 'Kaustobh Mahajan', 'Yash mali',
'Siddhi kivale', 'Sanket Kandalkar', 'Kaustobh Mahajan', 'Yash mali', 'Siddhi
kivale', 'Tanuja Mali', 'Kaustobh Mahajan', 'Sanket Kandalkar', 'Siddhi
kivale', 'Kaustobh Mahajan', 'Yash mali')
```

```
supplier_details
```

```
{'P00001': 'Raka Ele.', 'P00002': 'Vijay Sales', 'P00003': 'Gada Ele.',
'P00004': 'Surya Ele.', 'P00005': 'Raka Ele.', 'P00006': 'Gada Ele.', 'P00007':
'Vijay Sales', 'P00008': 'Surya Ele.', 'P00009': 'Raka Ele.', 'P00010': 'Gada
Ele.', 'P00011': 'Surya Ele.', 'P00012': 'Raka Ele.', 'P00013': 'Surya Ele.',
```



```
'P00014': 'Raka Ele.', 'P00015': 'Gada Ele.', 'P00016': 'Vijay Sales', 'P00017':
'Deshmukh Sales', 'P00018': 'Raka Ele.', 'P00019': 'Deshmukh Sales', 'P00020':
'Gada Ele.'}
```

gender

```
{'Kaustobh Mahajan': 'male', 'Siddhi kivale': 'female', 'Sanket Kandalkar':
'male', 'Yash mali': 'male', 'Yash Bagul': 'male', 'Tanuja Mali': 'female'}
```

```
[ ]: frequency= {}
for item in product_details:
    if item in frequency:
        frequency[item] += 1
    else:
        frequency[item] = 1
print(frequency)
marklist= sorted(frequency.items(), key=lambda x: x[1],reverse=True)
sortdict = dict(marklist)
print(sortdict)
print('The most popular product for sales',list(sortdict.
↳keys())[0],'sold',list(sortdict.values())[0],'times')
```

```
{'Lenovo laptop': 6, 'Samsung Laptop': 1, 'Realmi 10pro': 2, 'Oppo f21': 3,
'Samsung M31': 3, 'LG TV 32*': 4, 'Samsung M21': 1}
```

```
{'Lenovo laptop': 6, 'LG TV 32*': 4, 'Oppo f21': 3, 'Samsung M31': 3, 'Realmi
10pro': 2, 'Samsung Laptop': 1, 'Samsung M21': 1}
```

The most popular product for sales Lenovo laptop sold 6 times

```
[ ]: from collections import Counter
counter = dict(Counter(list(supplier_details.values())))
sorted_counter = sorted(counter.items(), key= lambda x:x[1],reverse=True)
sorted_counter = dict(sorted_counter)
print('The most popular product for sales',list(sorted_counter.keys())[0],
↳'sold',list(sorted_counter.values())[0],'Items')
```

The most popular product for sales Raka Ele. sold 6 Items

```
[ ]: frequency= {}
for item in customer_details:
    if item in frequency:
        frequency[item] += 1
    else:
        frequency[item] = 1
print('Frequency is as below:\n',frequency)
marklist= sorted(frequency.items(), key=lambda x: x[1],reverse=True)
sortdict = dict(marklist)
print('\nSorted dict is as below:\n',sortdict)
print('\n\nThe customer who buys most of the products',list(sortdict.
↳keys())[0],'buy',list(sortdict.values())[0],'Items')
```

Frequency is as below:

```
{'Kaustoobh Mahajan': 5, 'Siddhi kivale': 5, 'Sanket Kandalkar': 4, 'Yash mali': 4, 'Yash Bagul': 1, 'Tanuja Mali': 1}
```

Sorted dict is as below:

```
{'Kaustoobh Mahajan': 5, 'Siddhi kivale': 5, 'Sanket Kandalkar': 4, 'Yash mali': 4, 'Yash Bagul': 1, 'Tanuja Mali': 1}
```

The customer who buys most of the products Kaustoobh Mahajan buy 5 Items

```
[ ]: from collections import Counter
counter = dict(Counter(customer_details))
sorted_counter = sorted(counter.items(), key= lambda x:x[1],reverse=True)
sorted_counter = dict(sorted_counter)
print('The customer who buys most of the products',list(sorted_counter.
↳keys())[0], 'buy',list(sorted_counter.values())[0], 'Items')
```

The customer who buys most of the products Kaustoobh Mahajan buy 5 Items

```
[ ]: from collections import Counter
counter = dict(Counter(customer_details))
names = list(counter.keys())
print(names)
male = 0
female = 0
for name in names:
    if gender[name]=='male':
        male = male+1
    if gender[name]=='female':
        female=female+1
print('Total no of Male=',male)
print('Total no of Female=',female)
```

```
['Kaustoobh Mahajan', 'Siddhi kivale', 'Sanket Kandalkar', 'Yash mali', 'Yash Bagul', 'Tanuja Mali']
Total no of Male= 4
Total no of Female= 2
```

Mayuri Sunil Yadav

Roll no:- 186

```
import numpy as np
array1=np.array([[1,2,3],[4,5,6],[7,8,9]])
array1
```

Output

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

```
array2=np.array([[11,12,13],[14,15,16],[17,18,19]])
array2
```

Output

```
array([[11, 12, 13],
       [14, 15, 16],
       [17, 18, 19]])
```

## 1. Matrix Operation

### 1.1 Addition

```
resultarray=array1+array2
print("\nUsing Operator:\n",resultarray)
resultarray=np.add(array1,array2)
print("\nUsing Numpy Function:\n",resultarray)
```

Output

```
Using Operator:
[[12 14 16]
 [18 20 22]
 [24 26 28]]
```

Using Numpy Function:

```
[[12 14 16]
 [18 20 22]
 [24 26 28]]
```

## 1.2. Subtraction

```
resultarray=array1-array2
print("\nUsing Operator:\n",resultarray)
resultarray=np.subtract(array1,array2)
print("\nUsing Numpy Function:\n",resultarray)
```

### Output

Using Operator:

```
[[ -10  -10  -10]
 [ -10  -10  -10]
 [ -10  -10  -10]]
```

Using Numpy Function:

```
[[ -10  -10  -10]
 [ -10  -10  -10]
 [ -10  -10  -10]]
```

## 1.3. Multiplication

```
resultarray=array1*array2
print("\nUsing Operator:\n",resultarray)
resultarray=np.multiply(array1,array2)
print("\nUsing Numpy Function:\n",resultarray)
```

### Output

Using Operator:

```
[[ 11  24  39]
 [ 56  75  96]
 [119 144 171]]
```

Using Numpy Function:

```
[[ 11  24  39]
 [ 56  75  96]
 [119 144 171]]
```

## 1.4. Division

```
resultarray=array1/array2
print("\nUsing Operator:\n",resultarray)
resultarray=np.divide(array1,array2)
print("\nUsing Numpy Function:\n",resultarray)
```

### Output

Using Operator:

```
[[0.09090909 0.16666667 0.23076923]
 [0.28571429 0.33333333 0.375    ]
 [0.41176471 0.44444444 0.47368421]]
```

Using Numpy Function:

```
[[0.09090909 0.16666667 0.23076923]
 [0.28571429 0.33333333 0.375    ]
 [0.41176471 0.44444444 0.47368421]]
```

## 1.5. Mod

```
resultarray=array1%array2
print("\nUsing Operator:\n",resultarray)
resultarray=np.mod(array1,array2)
print("\nUsing Numpy Function:\n",resultarray)
```

### Output

Using Operator:

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Using Numpy Function:

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

## 1.6. dot Product

```
resultarray=np.dot(array1,array2)
print("",resultarray)
```

## Output

```
[[ 90  96 102]
 [216 231 246]
 [342 366 390]]
```

## .7. Transpose

```
resultarray=np.transpose(array1)
print(resultarray)
#Or
resultarray=array1.transpose()
print(resultarray)
```

## Output

```
[[1 4 7]
 [2 5 8]
 [3 6 9]]
[[1 4 7]
 [2 5 8]
 [3 6 9]]
```

## 2. Horizontal and vertical stacking of Numpy Arrays

---

### 2.1. Horizontal Stacking

```
resultarray=np.hstack((array1,array2))
resultarray
```

## Output

```
array([[ 1,  2,  3, 11, 12, 13],
       [ 4,  5,  6, 14, 15, 16],
       [ 7,  8,  9, 17, 18, 19]])
```

### 2.2. Vertical Stacking

```
resultarray=np.vstack((array1,array2))
resultarray
```

## Output

```
array([[ 1, 2, 3],
       [ 4, 5, 6],
       [ 7, 8, 9],
       [11, 12, 13],
       [14, 15, 16],
       [17, 18, 19]])
```

## 3.Custom sequence generation

---

### 3.1. Range

```
import numpy as np
nparray=np.arange(0,12,1).reshape(3,4)
nparray
```

## Output

```
array([[ 0, 1, 2, 3],
       [ 4, 5, 6, 7],
       [ 8, 9, 10, 11]])
```

### 3.2. Linearly Separable

```
nparray=np.linspace(start=0,stop=24,num=12).reshape(3,4)
nparray
```

## Output

```
array([[ 0. , 2.18181818, 4.36363636, 6.54545455],
       [ 8.72727273, 10.90909091, 13.09090909, 15.27272727],
       [17.45454545, 19.63636364, 21.81818182, 24. ]])
```

### 3.3. Empty Array

```
nparray=np.empty((3,3),int)
nparray
```

#### Output

```
array([[ 11, 24, 39],
       [ 56, 75, 96],
       [119, 144, 171]])
```

### 3.4. Empty Like Some other array

```
nparray=np.empty_like(array1)
nparray
```

#### Output

```
array([[ 90, 96, 102],
       [216, 231, 246],
       [342, 366, 390]])
```

---

### 3.5. Identity Matrix

```
nparray=np.identity(3)
nparray
```

#### Output

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

## 4. Arithmetic and Statistical Operations, Mathematical Operations, Bitwise Operators

---



## 4.1. Arithmetic Operation

```
array1=np.array([1,2,3,4,5])
array2=np.array([11,12,13,14,15])
print(array1)
print(array2)
```

### Output

```
[1 2 3 4 5]
[11 12 13 14 15]
```

```
# Addition
print(np.add(array1,array2))
# Subtraction
print(np.subtract(array1,array2))
# Multiplication
print(np.multiply(array1,array2))
# Division
print(np.divide(array1,array2))
```

### Output

```
[12 14 16 18 20]
[-10 -10 -10 -10 -10]
[11 24 39 56 75]
[0.09090909 0.16666667 0.23076923 0.28571429 0.33333333]
```

---

## 4.2. Statistical and Mathematical Operations

```
array1=np.array([1,2,3,4,5,9,6,7,8,9,9])
# Standard Deviation
print(np.std(array1))
#Minimum
print(np.min(array1))
#Summation
print(np.sum(array1))
#Median
print(np.median(array1))
#Mean
print(np.mean(array1))
#Mode
```

```

from scipy import stats
print("Most Frequent element=",stats.mode(array1)[0])
print("Number of Occarances=",stats.mode(array1)[1])
# Variance
print(np.var(array1))

```

## Output

```

2.7990553306073913
1
63
6.0
5.7272727272727275
Most Frequent element= [9]
Number of Occarances= [3]
7.834710743801653

```

## 4.3. Bitwise Operations

```

array1=np.array([1,2,3],dtype=np.uint8)
array2=np.array([4,5,6])
# AND
resultarray=np.bitwise_and(array1,array2)
print(resultarray)
# OR
resultarray=np.bitwise_or(array1,array2)
print(resultarray)
#LeftShift
resultarray=np.left_shift(array1,2)
print(resultarray)
#RightShift
resultarray=np.right_shift(array1,2)
print(resultarray)

```

## Output

```

[0 0 2]
[5 7 7]
[ 4 8 12]

```

[0 0 0]

```
### You can get Binary Representation of Number #####  
print(np.binary_repr(10,8))  
resultarray=np.left_shift(10,2)  
print(resultarray)  
print(np.binary_repr(np.left_shift(10,2),8))
```

Output

```
00001010  
40  
00101000
```

## 5.Copying and viewing arrays

---

### 5.1 Copy

```
array1=np.arange(1,10)  
print(array1)  
newarray=array1.copy()  
print(newarray)  
##modification in Original Array  
array1[0]=100  
print(array1)  
print(newarray)
```

Output

```
[1 2 3 4 5 6 7 8 9]  
[1 2 3 4 5 6 7 8 9]  
[100 2 3 4 5 6 7 8 9]  
[1 2 3 4 5 6 7 8 9]
```

### 5.2 View

```
array1=np.arange(1,10)  
print(array1)  
newarray=array1.view()
```

```
print(newarray)
##modification in Original Array
array1[0]=100
print(array1)
print(newarray)
```

## Output

```
[1 2 3 4 5 6 7 8 9]
[1 2 3 4 5 6 7 8 9]
[100  2  3  4  5  6  7  8  9]
[100  2  3  4  5  6  7  8  9]
```

## 6. Searching

```
array1=np.array([[1,2,3,12,5,7],[94,5,6,7,89,44],[7,8,9,11,13,14]])
print(array1)
```

## Output

```
[[ 1  2  3 12  5  7]
 [94  5  6  7 89 44]
 [ 7  8  9 11 13 14]]
```

```
np.sort(array1,axis=0)
```

## Output

```
array([[ 1,  2,  3,  7,  5,  7],
       [ 7,  5,  6, 11, 13, 14],
       [94,  8,  9, 12, 89, 44]])
```

```
np.sort(array1,axis=1)
```

## Output

```
array([[ 1,  2,  3,  5,  7, 12],
       [ 5,  6,  7, 44, 89, 94],
```

```
[ 7, 8, 9, 11, 13, 14]])
```

## 7. Searching

```
array1=np.array([1,2,3,12,5,7])  
np.searchsorted(array1,7,side="left")#Perform Search After sorting
```

Output

3

## 8. Counting

```
array1=np.array([1,2,3,12,5,7,0])  
print(np.count_nonzero(array1))#Return total Non Zero element  
print(np.nonzero(array1))#Return Index  
print(array1.size)#Total Element
```

Output

```
6  
(array([0, 1, 2, 3, 4, 5], dtype=int64),)  
7
```

## 9. Data Stacking

```
array1=np.array(np.arange(1,5).reshape(2,2))  
print(array1)  
array2=np.array(np.arange(11,15).reshape(2,2))  
print(array2)
```

Output

```
[[1 2]  
 [3 4]]  
[[11 12]  
 [13 14]]
```

```
newarray=np.stack([array1,array2],axis=0)
```

```
print(newarray)
```

Output

```
[[1 2]
 [3 4]]
[[11 12]
 [13 14]]
```

```
newarray=np.stack([array1,array2],axis=1)
```

```
print(newarray)
```

Output

```
[[1 2]
 [11 12]]
[[3 4]
 [13 14]]
```

## 10. Append

```
array1=np.arange(1,10).reshape(3,3)
```

```
print(array1)
```

```
array2=np.arange(21,30).reshape(3,3)
```

```
print(array2)
```

Output

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[21 22 23]
 [24 25 26]
 [27 28 29]]
```

```
np.append(array1,array2,axis=0)
```

Output

```
array([[ 1, 2, 3],
       [ 4, 5, 6],
       [ 7, 8, 9],
       [21, 22, 23],
       [24, 25, 26],
       [27, 28, 29]])
```

```
np.append(array1,array2,axis=1)
```

Output

```
array([[ 1, 2, 3, 21, 22, 23],
       [ 4, 5, 6, 24, 25, 26],
       [ 7, 8, 9, 27, 28, 29]])
```

## 11. Concat

```
array1=np.arange(1,10).reshape(3,3)
print(array1)
array2=np.arange(21,30).reshape(3,3)
print(array2)
```

Output

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[21 22 23]
 [24 25 26]
 [27 28 29]]
```

```
np.concatenate((array1,array2),axis=0)
```

Output

```
array([[ 1, 2, 3],
       [ 4, 5, 6],
       [ 7, 8, 9],
       [21, 22, 23],
       [24, 25, 26],
       [27, 28, 29]])
```

```
np.concatenate((array1,array2),axis=1)
```

Output

```
array([[ 1, 2, 3, 21, 22, 23],
       [ 4, 5, 6, 24, 25, 26],
       [ 7, 8, 9, 27, 28, 29]])
```

```
import numpy as np

# using loadtxt()
arr =
np.loadtxt("F:\\ISO\\EDS\\NOTES\\dataset\\testmarks1.csv",delimiter=";",s
kiprows=1)
print(type(arr))
arr.shape
```

Output

```
<class 'numpy.ndarray'>
(10, 5)
```

```
EDS=arr[:,1]
print(EDS)
```

Output



[43.05 43.47 42.24 39.24 40.9 39.47 41.68 42.19 44.75 46.95]

```
SON=arr[:,2]
```

```
print(SON)
```

Output

[27.79 28.52 28.16 26.16 26.03 26.31 25.63 27.61 28.35 28.88]

**NAME :Mayuri Sunil  
Yadav  
CLASS : A  
BATCH : A4  
ROLL NO : 186**

```
import pandas as pd
import numpy as np
f1 = open("F:\grainsales.csv","r")
data = pd.read_csv(f1)
df = pd.DataFrame(data)
maindata = df
df['Sales'].describe()
df=df.groupby('Months').sum()
df=df.sort_values(by= ['Sales'], ascending=False) df.head(1)
print("Best Month for Sales: July")
print("Revenue Earned was: 16000000")
df
maindata
df = df.groupby("GrainName").sum()
df = df.sort_values(by=["Sales"], ascending = False)
df.head (1)
print("Most Sold Grain is: Wheat")
print("The Best Month for sales is July and this product has occurred in July
so this is most sold product with highest sales")
df
maindata
df= df.groupby("City").sum()
df = df.sort_values (by = ['Sales'], ascending= False)
df.head (1)
print("'Asansole' Has sold highest no. of products")
df
maindata
df = df.groupby('State').sum()
df = df.sort_values (by = ['Sales'], ascending = False) print("West
Bengal has highest sales")

Best Month for Sales: July
Revenue Earned was: 16000000
```

Most Sold Grain is: Wheat

The Best Month for sales is July and this product has occurred in July so this is most sold product with highest sales

'Asansole' Has sold highest no. of products

West Bengal has highest sales.

Name - Mayuri Sunil Yadav

Roll no-186

Batch-A4

Prn no-202201070125

## ASSIGNMENT -5

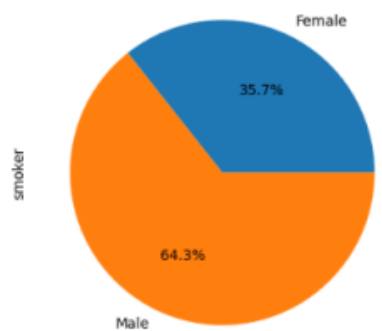
```
[ ] import pandas as pd
import matplotlib.pyplot as plt
d = pd.read_csv('/content/tips.csv')
print(d)
```

	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]

```
import pandas as pd
import matplotlib.pyplot as plt
d = pd.read_csv('/content/tips.csv')
#print(d)
t1 = d.groupby("sex").count()
t1["smoker"].plot(kind = "pie", autopct = '%1.1f%%')
plt.show()
print(t1)
```

☐



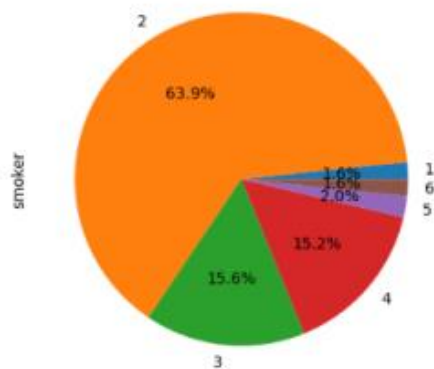
	total_bill	tip	smoker	day	time	size
sex						
Female	87	87	87	87	87	87
Male	157	157	157	157	157	157

```

import pandas as pd
import matplotlib.pyplot as plt
d = pd.read_csv('/content/tips.csv')
#print(d)
t1 = d.groupby("size").count()
t1["smoker"].plot(kind = "pie",autopct = '%1.1f%%')
plt.show()
print(t1)

```

✓

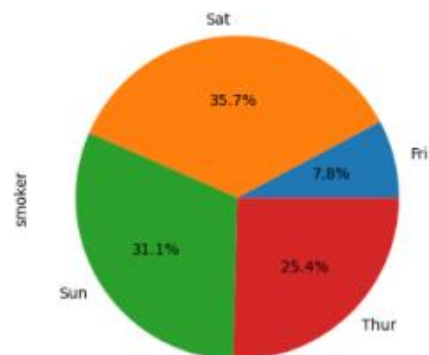


```

import pandas as pd
import matplotlib.pyplot as plt
d = pd.read_csv('/content/tips.csv')
#print(d)
t1 = d.groupby("day").count()
t1["smoker"].plot(kind = "pie",autopct = '%1.1f%%')
plt.show()
print(t1)

```

✓

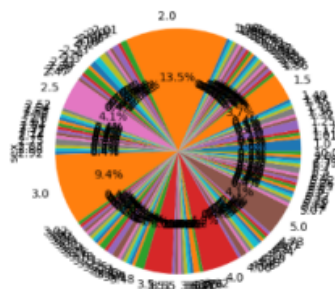


```

import pandas as pd
import matplotlib.pyplot as plt
d = pd.read_csv('/content/tips.csv')
#print(d)
t1 = d.groupby("tip").count()
t1["sex"].plot(kind = "pie",autopct = '%1.1f%%')
plt.show()
print(t1)

```

☐



tip	total_bill	sex	smoker	day	time	size
1.00	4	4	4	4	4	4
1.01	1	1	1	1	1	1
1.10	1	1	1	1	1	1
1.17	1	1	1	1	1	1
1.25	3	3	3	3	3	3
...	...	...	...	...	...	...
6.70	1	1	1	1	1	1
6.73	1	1	1	1	1	1
7.58	1	1	1	1	1	1
9.00	1	1	1	1	1	1
10.00	1	1	1	1	1	1

[123 rows x 6 columns]

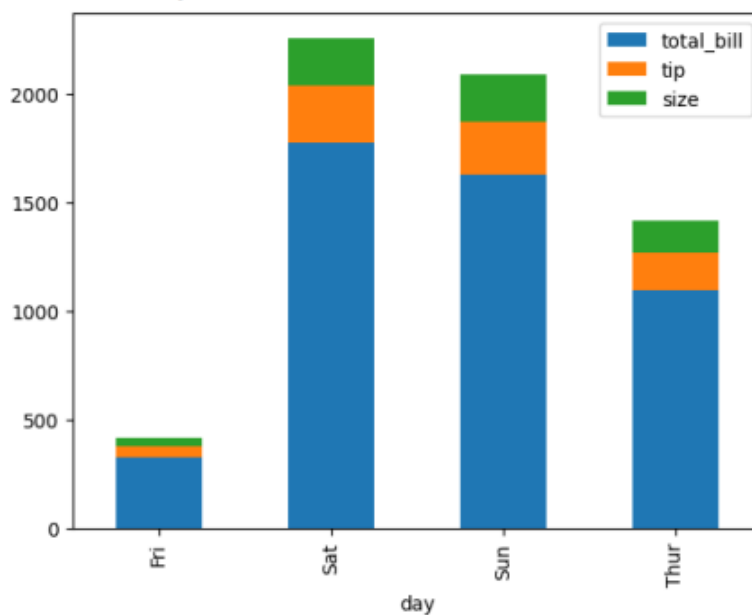
```

▶ t2 = d.groupby("day").sum("total_bill")
t2.plot(kind = "bar",stacked = True)

```

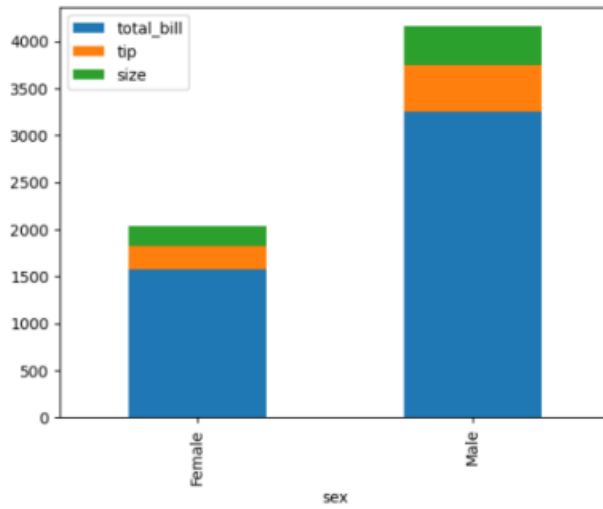
☐

<Axes: xlabel='day'>



```
t2 = d.groupby("sex").sum("smoker")
t2.plot(kind = "bar", stacked = True)
```

<Axes: xlabel='sex'>



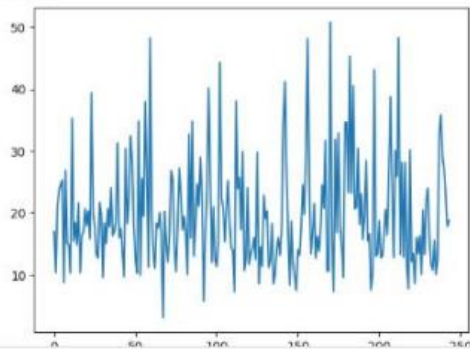
```
[ ] import matplotlib.pyplot as plt
d = pd.read_csv('/content/tips.csv')
#print(d)
d1 = d.groupby("total_bill").count()
print(d1)
```

```
total_bill  tip  sex  smoker  day  time  size
3.07        1    1      1      1      1      1
5.75        1    1      1      1      1      1
7.25        2    2      2      2      2      2
7.51        1    1      1      1      1      1
7.56        1    1      1      1      1      1
...
45.35       1    1      1      1      1      1
48.17       1    1      1      1      1      1
48.27       1    1      1      1      1      1
48.33       1    1      1      1      1      1
50.81       1    1      1      1      1      1
```

[229 rows x 6 columns]

```
d["total_bill"].plot()
```

<Axes: >



```

import pandas as pd
import matplotlib.pyplot as plt
d = pd.read_csv('/content/tips.csv')
print(d)
d1 = d.groupby("size").count()
print(d1)

```

```

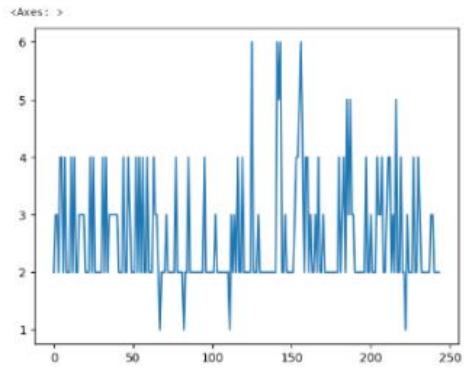
total_bill  tip  sex  smoker  day  time
size
1          4    4    4        4    4    4
2       156  156  156       156  156  156
3        38   38   38        38   38   38
4        37   37   37        37   37   37
5         5    5    5         5    5    5
6         4    4    4         4    4    4

```

```

d["size"].plot()

```

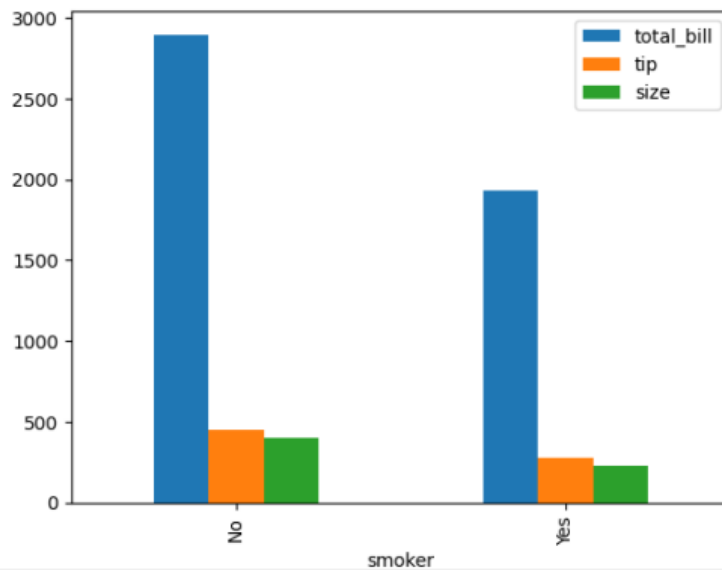


```

t2 = d.groupby("smoker").sum("tip")
t2.plot(kind = "bar",stacked = False)

```

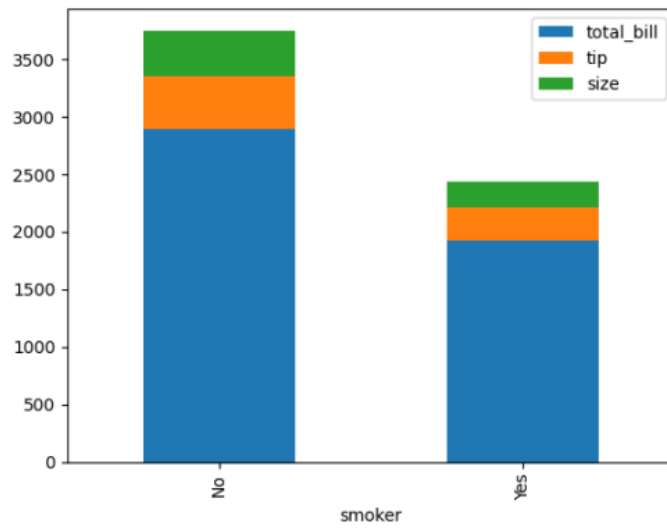
<Axes: xlabel='smoker'>



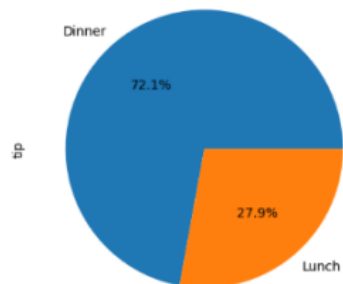


```
t2 = d.groupby("smoker").sum("time")
t2.plot(kind = "bar",stacked = True)
```

<Axes: xlabel='smoker'>



```
[ ] import pandas as pd
import matplotlib.pyplot as plt
d = pd.read_csv('/content/tips.csv')
#print(d)
t1 = d.groupby("time").count()
t1["tip"].plot(kind = "pie",autopct = '%1.1f%%')
plt.show()
print(t1)
```



time	total_bill	tip	sex	smoker	day	size
Dinner	176	176	176	176	176	176
Lunch	68	68	68	68	68	68