SUMESH R - 20104169

Basic Analysis using NumPy and Pandas

Import Libraries

In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
from numpy import cov
from scipy.stats import pearsonr
from scipy.stats import spearmanr
```

Import Dataset

In [4]:
```
data = pd.read_csv("4_drug200.csv")
```

In [5]:
```
display(data)
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
...	...	...	...	...	...	...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

To display top 10 rows

In [6]:
```
data.head(10)
```

Out[6]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
5	22	F	NORMAL	HIGH	8.607	drugX
6	49	F	NORMAL	HIGH	16.275	drugY
7	41	M	LOW	HIGH	11.037	drugC
8	60	M	NORMAL	HIGH	15.171	drugY
9	43	M	LOW	NORMAL	19.368	drugY

to display last 5 rows

In [7]:
```
data.tail()
```

Out[7]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

statistical summary

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

Out[8]:

	Age	Na_to_K
count	200.000000	200.000000
mean	44.315000	16.084485
std	16.544315	7.223956
min	15.000000	6.269000
25%	31.000000	10.445500
50%	45.000000	13.936500

	Age	Na_to_K
75%	58.000000	19.380000
max	74.000000	38.247000

To print number of elements

In [9]:
```
data.size
```

Out[9]: 1200

to print number of row and cols

In [10]:
```
data.shape
```

Out[10]: (200, 6)

to find missing values

In [11]:
```
data.isna()
```

Out[11]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
195	False	False	False	False	False	False
196	False	False	False	False	False	False
197	False	False	False	False	False	False
198	False	False	False	False	False	False
199	False	False	False	False	False	False

200 rows × 6 columns

fill null values with a constant

In [12]:
```
data.fillna(5)
```

Out[12]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
...	...	...	...	...	...	...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

mean

In [13]:
```
data.mean()
```

Out[13]:
```
Age        44.315000
Na_to_K    16.084485
dtype: float64
```

median

In [14]:
```
data.median()
```

Out[14]:
```
Age        45.0000
Na_to_K    13.9365
dtype: float64
```

mode

In [15]:
```
data.mode()
```

Out[15]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	47.0	M	HIGH	HIGH	12.006	drugY
1	NaN	NaN	NaN	NaN	18.295	NaN

cumsum

In [16]:
```
data.cumsum()
```

Out[16]:

	Age	Sex	
0	23	F	
1	70	FM	
2	117	FMM	
3	145	FMMF	
4	206	FMMFF	
...	...	...	
195	8732	FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...	HIGHLOWLOWNORMALLOW
196	8748	FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...	HIGHLOWLOWNORMALLOW
197	8800	FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...	HIGHLOWLOWNORMALLOW
198	8823	FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...	HIGHLOWLOWNORMALLOW
199	8863	FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...	HIGHLOWLOWNORMALLOW

200 rows × 6 columns

min

In [17]:
```
data.min()
```

Out[17]:
```
Age              15
Sex               F
BP             HIGH
Cholesterol    HIGH
Na_to_K       6.269
Drug          drugA
dtype: object
```

max

In [18]:
```
data.max()
```

Out[18]:
```
Age              74
Sex               M
BP           NORMAL
Cholesterol  NORMAL
Na_to_K      38.247
Drug          drugY
dtype: object
```

sum

In [19]:
```
data.sum()
```

Out[19]:
```
Age                                                          8863
Sex            FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...
BP             HIGHLOWLOWNORMALLOWNORMALNORMALLOWNORMALLOWLOW...
Cholesterol    HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHNORMALHIGH...
Na_to_K                                                  3216.897
Drug           drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugCdrugYd...
dtype: object
```

count

In [20]:
```
data.count()
```

Out[20]:
```
Age            200
Sex            200
BP             200
Cholesterol    200
Na_to_K        200
Drug           200
dtype: int64
```

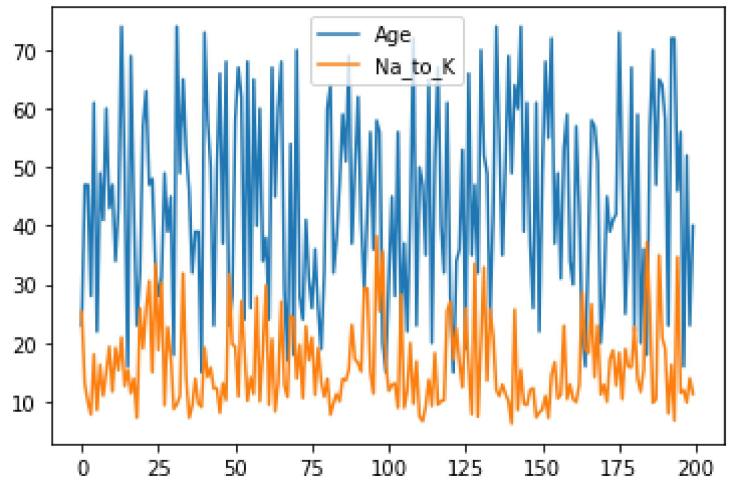
to select a particular columns

In [21]:
```
df=pd.DataFrame(data[['Age','Na_to_K']])
import matplotlib.pyplot as plt
```

line plot

In [22]:
```
df.plot.line()
```

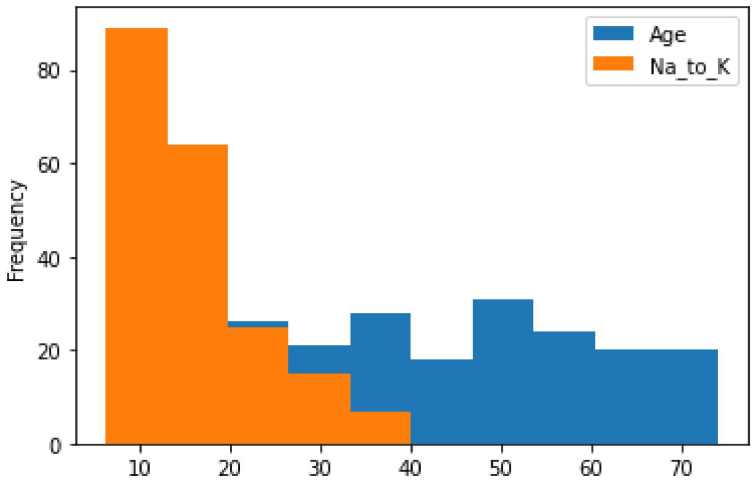
Out[22]:
```
<AxesSubplot:>
```



histogram

In [23]:
```
df.plot.hist()
```

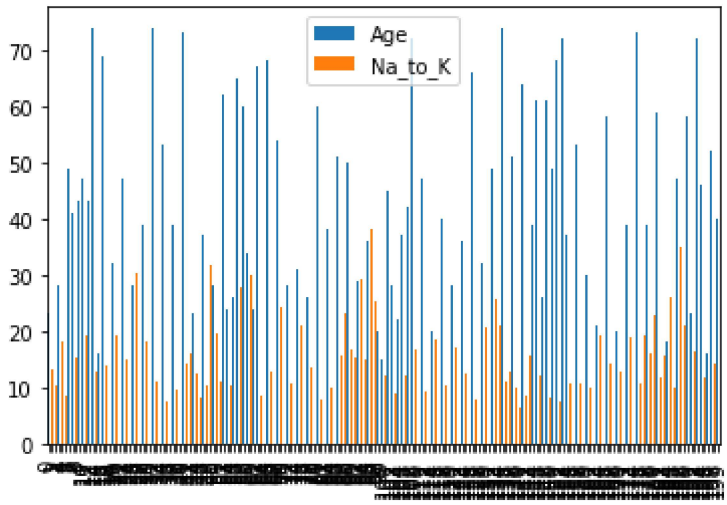
Out[23]: <AxesSubplot:ylabel='Frequency'>



bar chart

In [24]:
```
df.plot.bar()
```

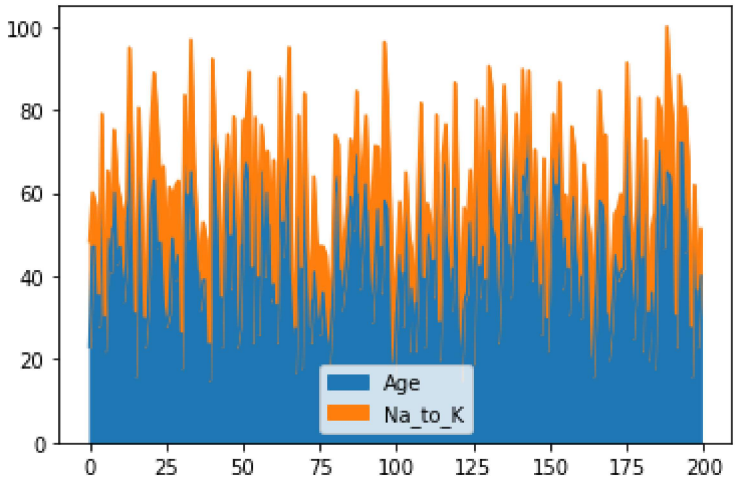
Out[24]: <AxesSubplot:>



area plot

In [25]:
```
df.plot.area()
```

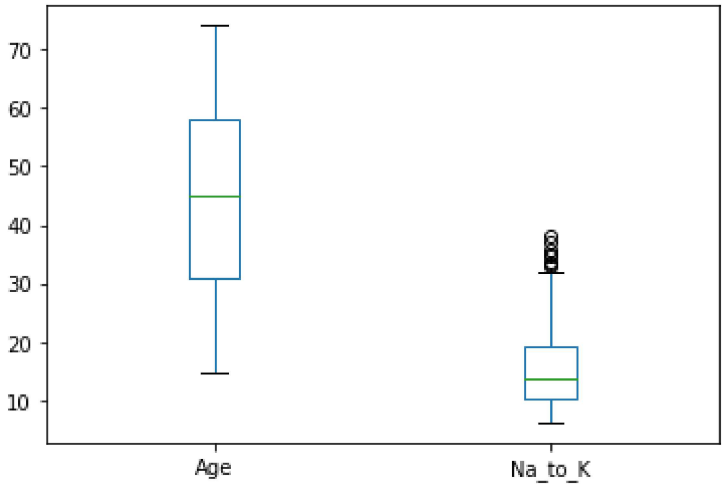
Out[25]: <AxesSubplot:>



box plot

In [26]:
```
df.plot.box()
```

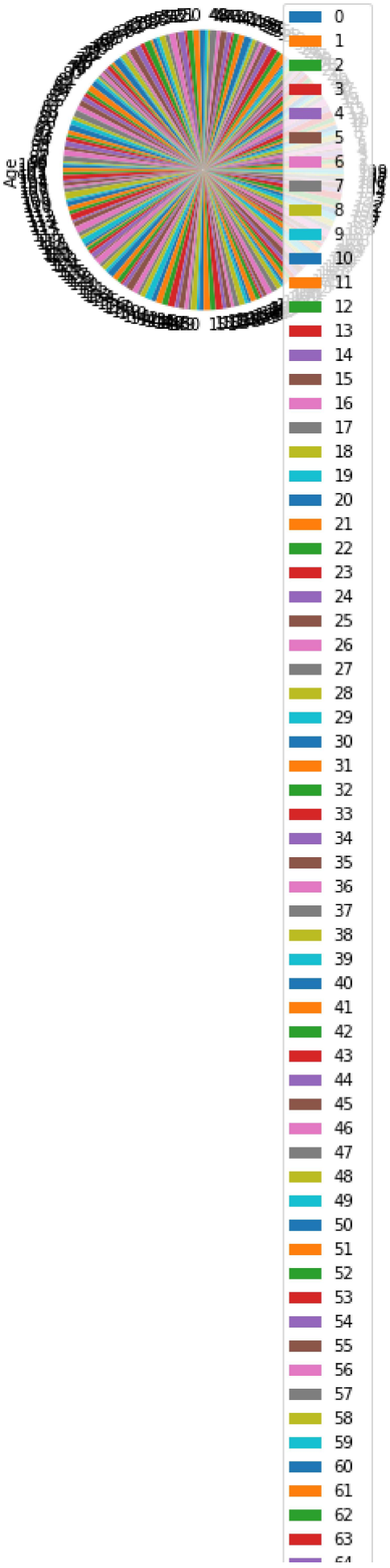
Out[26]: <AxesSubplot:>

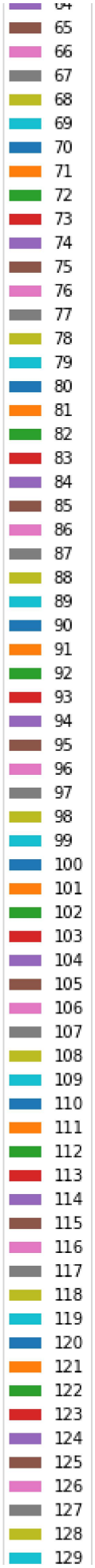


pie plot

In [27]:
```
df.plot.pie(y="Age")
```

Out[27]: <AxesSubplot:ylabel='Age'>

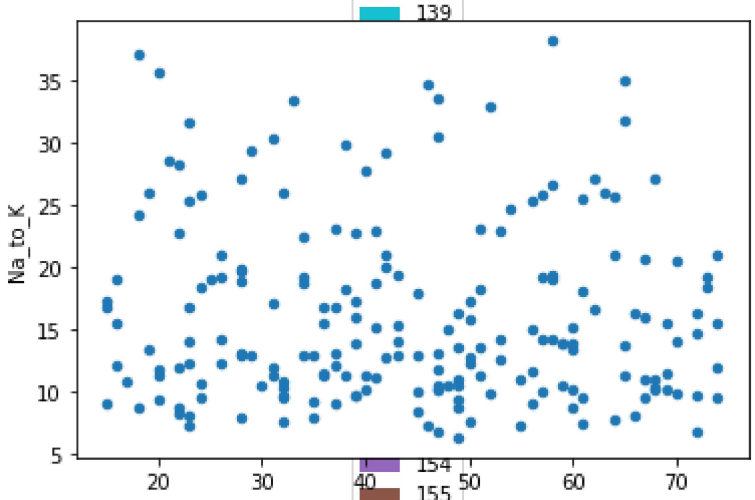

Age
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63


64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

scatter plot

In [28]:
```
df.plot.scatter(x="Age",y="Na_to_K")
```

Out[28]:
```
<AxesSubplot:xlabel='Age', ylabel='Na_to_K'>
```



covariance

In [29]:
```
cov(data["Age"],data["Na_to_K"])
```

Out[29]:
```
array([[273.71434673,  -7.54375153],
       [ -7.54375153,  52.18553348]])
```

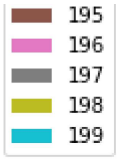
correlation

In [30]:
```
spearmanr(data["Age"],data["Na_to_K"])
```

Out[30]:
```
SpearmanrResult(correlation=-0.047273882688479915, pvalue=0.5062200581387418)
```

In [31]:
```
pearsonr(data["Age"],data["Na_to_K"])
```

Out[31]:
```
(-0.06311949726772592, 0.3745786399034559)
```


195
196
197
198
199