Deena 20104016

Basic Analysis using Numpy and Pandas

Import Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from numpy import cov
from scipy.stats import pearsonr
from scipy.stats import spearmanr
```

Importing Dataset

In [2]:
```
df=pd.read_csv("4_drug200.csv")
df
```

Out[2]:

	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 first 10 rows

In [3]:
```
df.head(10)
```

Out[3]:

	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 [4]:
```
df.tail(5)
```

Out[4]:

	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

Satistical Summary

In [5]:
```
df.mean()
```

Out[5]:
```
Age        44.315000
Na_to_K    16.084485
dtype: float64
```

In [6]:
```
df.median()
```

Out[6]:
```
Age        45.0000
Na_to_K    13.9365
dtype: float64
```

In [7]:
```
df.mode()
```

Out[7]:

	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

In [8]:
```
df.cumsum()
```

Out[8]:

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

200 rows × 6 columns

In [9]:
```
df.count()
```

Out[9]:
```
Age            200
Sex            200
BP             200
Cholesterol    200
Na_to_K        200
Drug           200
dtype: int64
```

In [10]:
```
df.max()
```

Out[10]:
```
Age              74
Sex               M
BP           NORMAL
Cholesterol  NORMAL
Na_to_K      38.247
Drug          drugY
dtype: object
```

In [11]:
```
df.min()
```

Out[11]:
```
Age              15
Sex               F
BP             HIGH
Cholesterol    HIGH
Na_to_K       6.269
Drug          drugA
dtype: object
```

In [12]:
```
df.sum()
```

Out[12]:
```
Age                                                          8863
Sex            FMMFFFFMMMFFMFFFMMMFMMMFFFMFFMFMMFMMMMFMFFMMFF...
BP             HIGHLOWLOWNORMALLOWNORMALNORMALLOWNORMALLOWLOW...
Cholesterol    HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHNORMALHIGH...
Na_to_K                                                  3216.897
Drug           drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugCdrugYd...
dtype: object
```

In [13]:
```
cov(df['Age'],df['Na_to_K'])
```

Out[13]:
```
array([[273.71434673,  -7.54375153],
       [ -7.54375153,  52.18553348]])
```

In [14]:
```
df.describe()
```

Out[14]:

	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
75%	58.000000	19.380000
max	74.000000	38.247000

pearsonr

In [15]:
```
pearsonr(df['Age'],df['Na_to_K'])
```

Out[15]:
```
(-0.06311949726772592, 0.3745756399034559)
```

spearmanr

In [16]:
```
spearmanr(df['Age'],df['Na_to_K'])
```

Out[16]:
```
SpearmanrResult(correlation=-0.047273882688479915, pvalue=0.5062200581387418)
```

To find shape and size

In [17]:
```
df.shape
```

Out[17]:
```
(200, 6)
```

In [18]:
```
df.size
```

Out[18]:
```
1200
```

To fill the null values

In [19]:
```
df.isna()
```

Out[19]:

	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

To fill missing values

In [20]:
```
df.dropna()
```

Out[20]:

	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

coloumns

In [21]:
```
df.columns
```

Out[21]:
```
Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')
```

to print a particular coloumn

In [22]:
```
data=df[['Age', 'Na_to_K']]
data
```

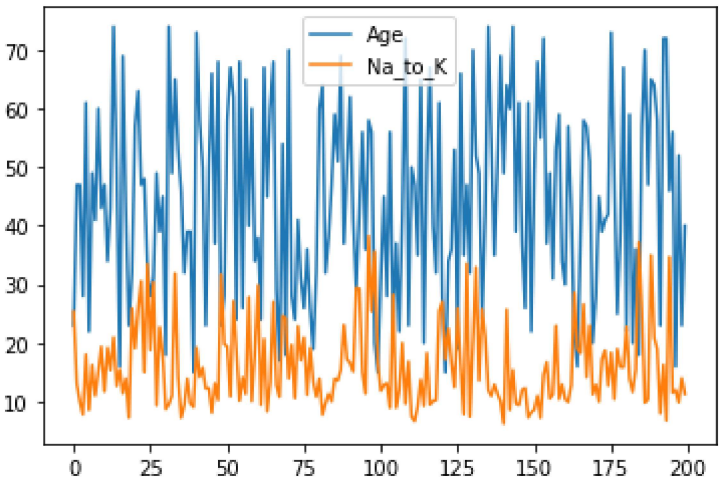
	Age	Na_to_K
0	23	25.355
1	47	13.093
2	47	10.114
3	28	7.798
4	61	18.043
...	...	...
195	56	11.567
196	16	12.006
197	52	9.894
198	23	14.020
199	40	11.349

200 rows × 2 columns

line plot

In [23]:
```
data.plot.line()
```

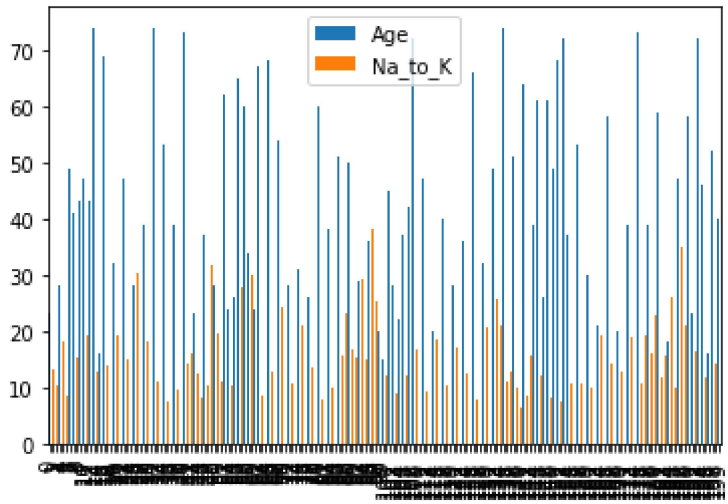
Out[23]: <AxesSubplot:>



bar plot

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

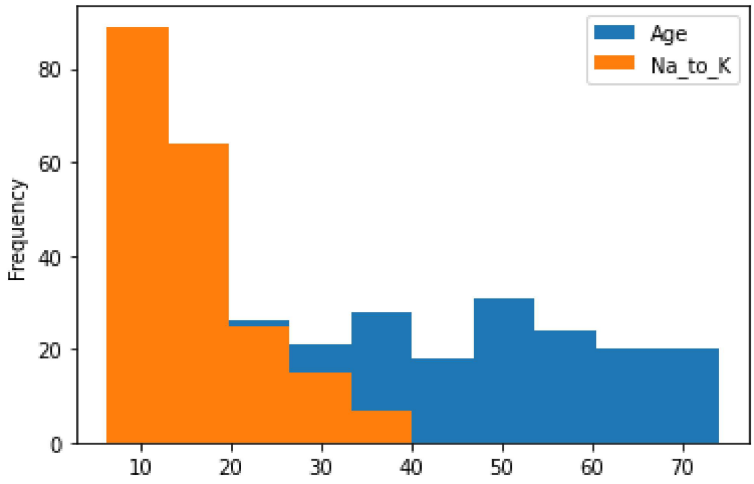
Out[24]: <AxesSubplot:>



hist plot

In [25]:
```
data.plot.hist()
```

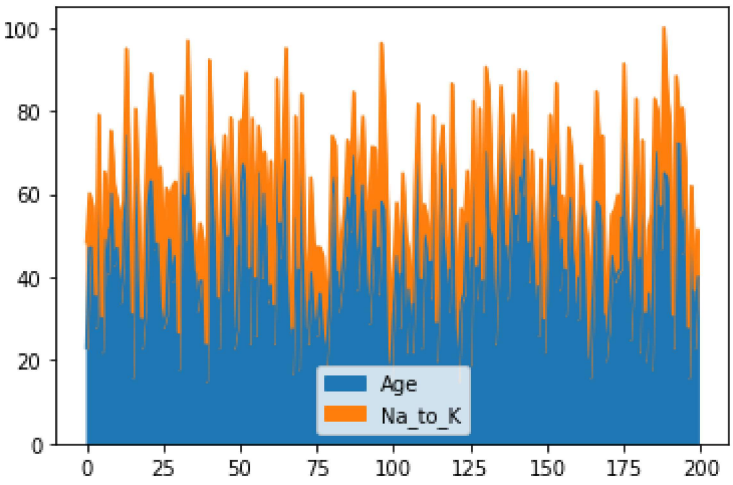
Out[25]: <AxesSubplot:ylabel='Frequency'>



Area plot

In [26]:
```
data.plot.area()
```

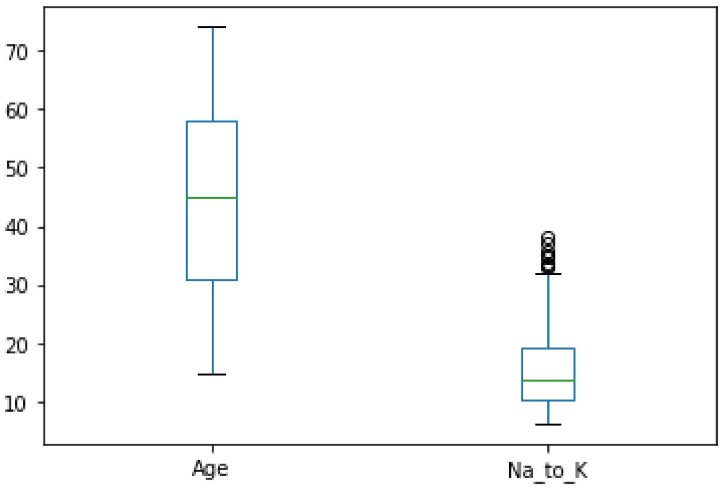
Out[26]: <AxesSubplot:>



Box plot

In [27]:
```
data.plot.box()
```

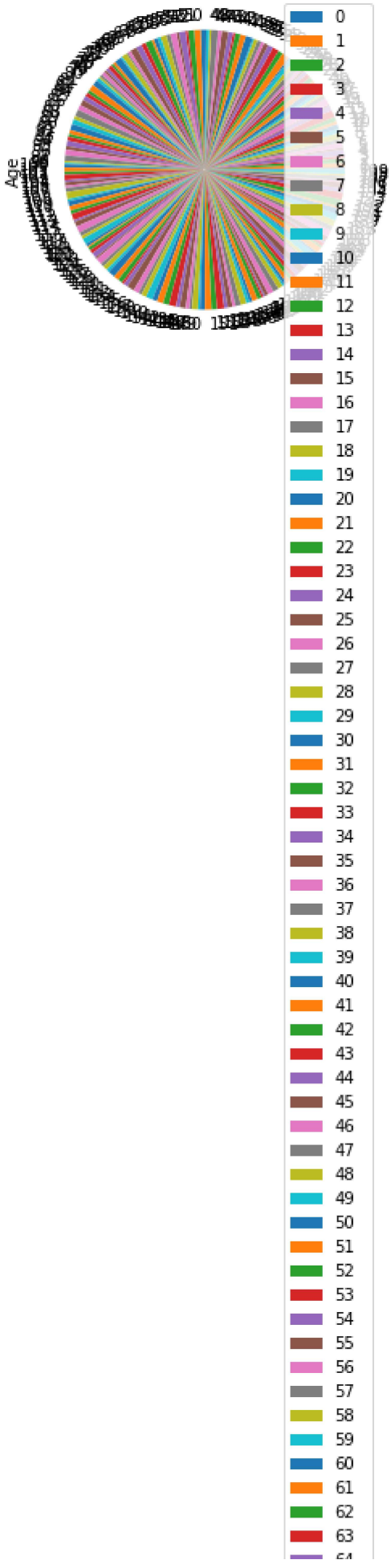
Out[27]: <AxesSubplot:>

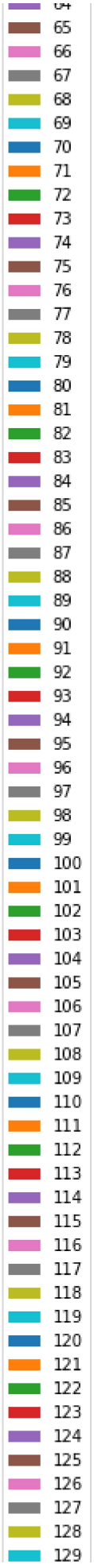


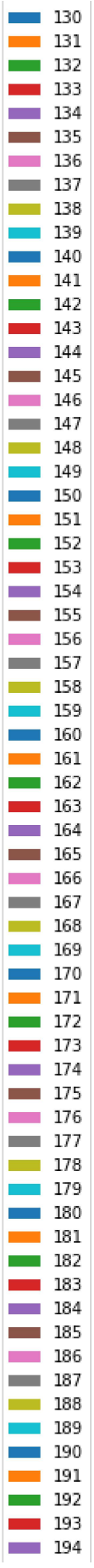
pie plot

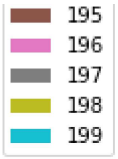
In [28]:
```
data.plot.pie(y='Age')
```

Out[28]:
```
<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


130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194



In [29]:
```
data.plot.scatter(x= 'Age',y='Na_to_K')
```

Out[29]: <AxesSubplot:xlabel='Age', ylabel='Na_to_K'>