

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
In [2]: data=pd.read_csv(r"C:\Users\user\Desktop\Vicky\4_drug200.csv")
```

```
In [3]: data.head()
```

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

```
In [4]: data.tail()
```

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

```
In [5]: data.isna()
```

```
Out[5]:
```

	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

```
In [8]: data.fillna(value=5)
```

```
Out[8]:
```

	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

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

```
Out[9]: (200, 6)
```

```
In [14]: data.size
```

```
Out[14]: 1200
```

```
In [15]: pd.isna(data)
```

```
Out[15]:
```

	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

```
In [20]: data.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

200 rows × 6 columns

In [21]: `data.describe()`

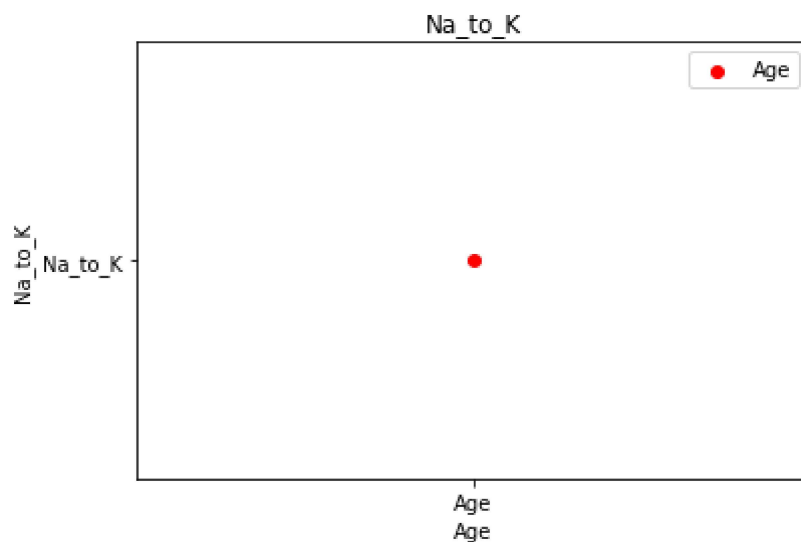
Out[21]:

	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

```
In [92]: x="Age"
y="Na_to_K"
plt.scatter(x,y,label="Age",color="r")

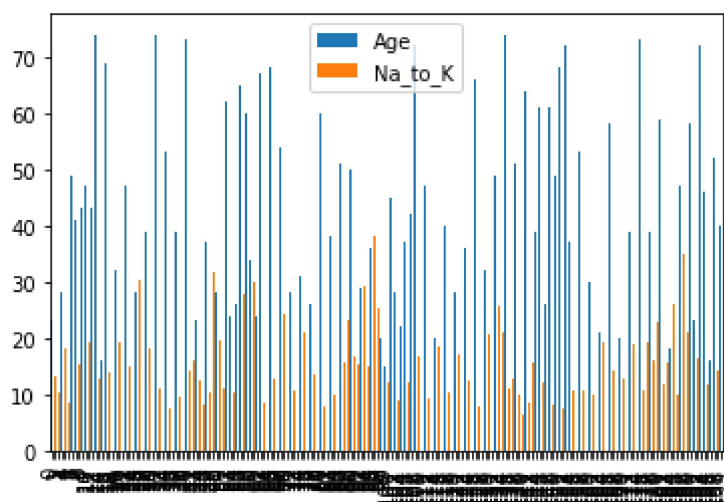
plt.title("Na_to_K")
plt.xlabel("Age")
plt.ylabel("Na_to_K")
plt.legend()
plt.show
```

Out[92]: <function matplotlib.pyplot.show(close=None, block=None)>



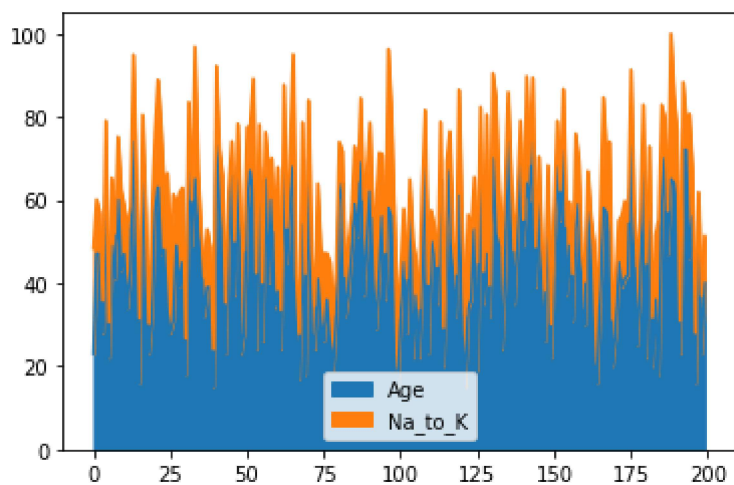
```
In [34]: data.plot.bar()
```

Out[34]: <AxesSubplot:>



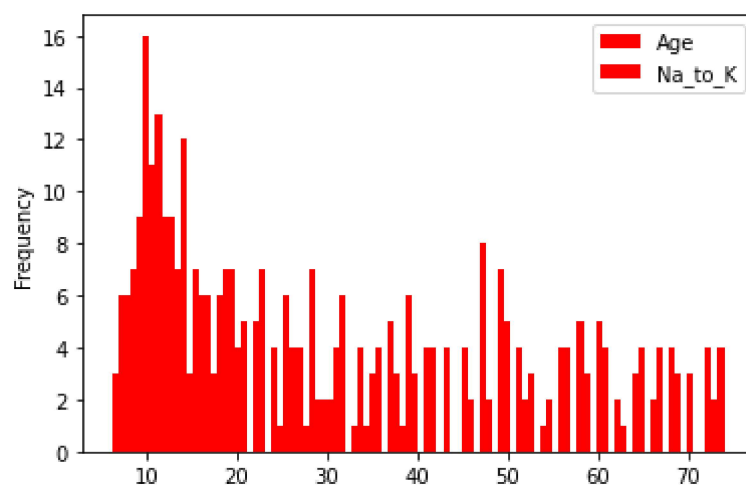
```
In [38]: data.plot.area()
```

Out[38]: <AxesSubplot:>



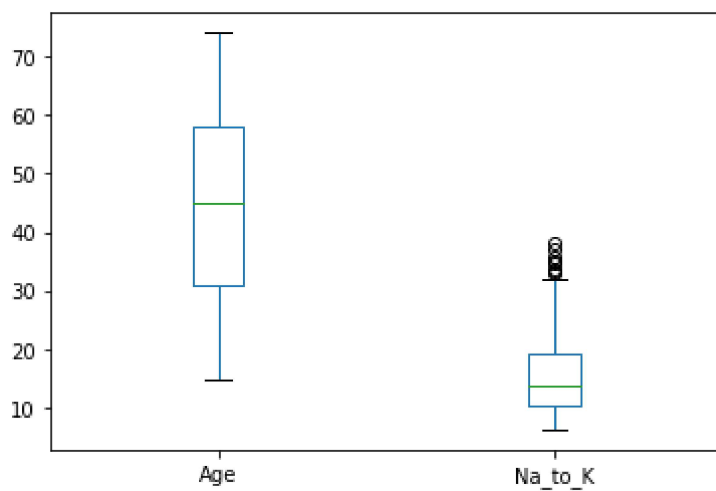
```
In [89]: data.plot.hist(bins=100,color="r")
```

```
Out[89]: <AxesSubplot:ylabel='Frequency'>
```

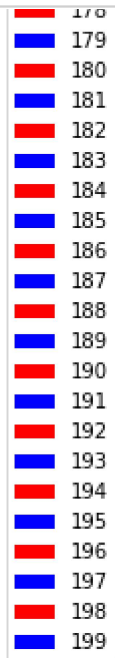


```
In [40]: data.plot.box()
```

```
Out[40]: <AxesSubplot:>
```



```
In [57]: cols=["r","b"]
data.plot.pie(y="Age",shadow=True,startangle=90,colors=cols)
```

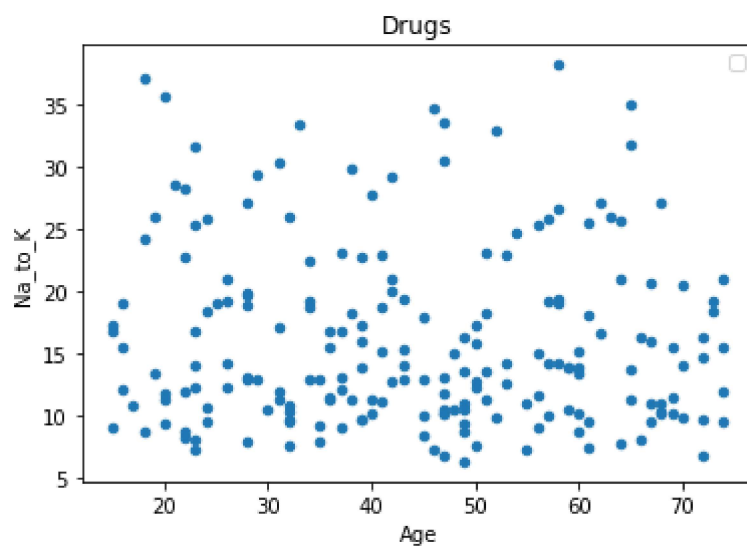


```
In [52]: data.plot.scatter(x="Age",y="Na_to_K")
```

```
plt.title("Drugs")
plt.legend()
```

No handles with labels found to put in legend.

```
Out[52]: <matplotlib.legend.Legend at 0x24478142550>
```



```
In [58]: from numpy import linalg as la
```

```
In [60]: data.mean()
```

```
Out[60]: Age      44.315000
Na_to_K    16.084485
dtype: float64
```

```
In [61]: data.median()
```

```
Out[61]: Age      45.0000
Na_to_K    13.9365
dtype: float64
```

```
In [62]: data.mode()
```

```
Out[62]:
```

	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 [63]: data.std()
```

```
Out[63]: Age      16.544315
Na_to_K      7.223956
dtype: float64
```

```
In [64]: data.var()
```

```
Out[64]: Age      273.714347
Na_to_K     52.185533
dtype: float64
```

```
In [65]: data.max()
```

```
Out[65]: Age      74
Sex      M
BP      NORMAL
Cholesterol  NORMAL
Na_to_K    38.247
Drug      drugY
dtype: object
```

```
In [66]: data.min()
```

```
Out[66]: Age      15
Sex      F
BP      HIGH
Cholesterol  HIGH
Na_to_K    6.269
Drug      drugA
dtype: object
```



```
In [67]: data1=data[["Age", "Na_to_K"]]  
data1
```

Out[67]:

	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

```
In [68]: data1.sum()
```

Out[68]: Age 8863.000
Na_to_K 3216.897
dtype: float64

```
In [69]: data.cumsum()
```

Out[69]:

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

200 rows × 6 columns

```
In [70]: data1.cumsum()
```

Out[70]:

	Age	Na_to_K
0	23	25.355
1	70	38.448
2	117	48.562
3	145	56.360
4	206	74.403
...
195	8732	3169.628
196	8748	3181.634
197	8800	3191.528
198	8823	3205.548
199	8863	3216.897

200 rows × 2 columns

```
In [71]: from scipy.stats import spearmanr  
         from scipy.stats import pearsonr
```

```
In [72]: print(spearmanr(data,data1))
```

```
SpearmanrResult(correlation=array([[ 1.          ,  0.10446701,  0.06283512, -
0.06881114 , -0.04727388,
        -0.02040842,  1.          , -0.04727388],
 [ 0.10446701,  1.          , -0.00552946, -0.00881101, -0.14587075,
        -0.09109896,  0.10446701, -0.14587075],
 [ 0.06283512, -0.00552946,  1.          , -0.1354242 , -0.15091871,
        0.1910008 ,  0.06283512, -0.15091871],
 [-0.06881114 , -0.00881101, -0.1354242 ,  1.          , -0.01282312,
        0.04802219, -0.06881114 , -0.01282312],
 [-0.04727388, -0.14587075, -0.15091871, -0.01282312,  1.          ,
        0.775643 , -0.04727388,  1.          ],
 [-0.02040842, -0.09109896,  0.1910008 ,  0.04802219,  0.775643 ,
        1.          , -0.02040842,  0.775643 ],
 [ 1.          ,  0.10446701,  0.06283512, -0.06881114 , -0.04727388,
        -0.02040842,  1.          , -0.04727388],
 [-0.04727388, -0.14587075, -0.15091871, -0.01282312,  1.          ,
        0.775643 , -0.04727388,  1.          ]]), pvalue=array([[0.00000000e+
00, 1.40978395e-01, 3.76735771e-01, 3.32950156e-01,
        5.06220058e-01, 7.74235282e-01, 0.00000000e+00, 5.06220058e-01],
 [1.40978395e-01, 0.00000000e+00, 9.38059728e-01, 9.01451523e-01,
        3.93001141e-02, 1.99517379e-01, 1.40978395e-01, 3.93001141e-02],
 [3.76735771e-01, 9.38059728e-01, 0.00000000e+00, 5.58761447e-02,
        3.29109301e-02, 6.74424545e-03, 3.76735771e-01, 3.29109301e-02],
 [3.32950156e-01, 9.01451523e-01, 5.58761447e-02, 0.00000000e+00,
        8.56982221e-01, 4.99504731e-01, 3.32950156e-01, 8.56982221e-01],
 [5.06220058e-01, 3.93001141e-02, 3.29109301e-02, 8.56982221e-01,
        0.00000000e+00, 1.95525864e-41, 5.06220058e-01, 0.00000000e+00],
 [7.74235282e-01, 1.99517379e-01, 6.74424545e-03, 4.99504731e-01,
        1.95525864e-41, 0.00000000e+00, 7.74235282e-01, 1.95525864e-41],
 [0.00000000e+00, 1.40978395e-01, 3.76735771e-01, 3.32950156e-01,
        5.06220058e-01, 7.74235282e-01, 0.00000000e+00, 5.06220058e-01],
 [5.06220058e-01, 3.93001141e-02, 3.29109301e-02, 8.56982221e-01,
        0.00000000e+00, 1.95525864e-41, 5.06220058e-01, 0.00000000e+00]]))
```

```
In [75]: from numpy import mean,std,cov
from numpy.random import randn,seed
from matplotlib import pyplot
```

In [76]: `print(cov(data,data1))`

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-76-81961316810f> in <module>
----> 1 print(cov(data,data1))

<__array_function__ internals> in cov(*args, **kwargs)

C:\ProgramData\Anaconda3\lib\site-packages\numpy\lib\function_base.py in cov
(m, y, rowvar, bias, ddof, fweights, aweights, dtype)
    2426         if not rowvar and y.shape[0] != 1:
    2427             y = y.T
-> 2428         X = np.concatenate((X, y), axis=0)
    2429
    2430         if ddof is None:

<__array_function__ internals> in concatenate(*args, **kwargs)

ValueError: all the input array dimensions for the concatenation axis must match exactly, but along dimension 1, the array at index 0 has size 6 and the array at index 1 has size 2
```

In [77]: `data.fillna(value=5)`

Out[77]:

	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

In [80]: `data.count()`

```
Out[80]: Age          200
Sex          200
BP           200
Cholesterol  200
Na_to_K      200
Drug         200
dtype: int64
```

In [81]: `data.count`

```
Out[81]: <bound method DataFrame.count of      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 x 6 columns]>
```

In [82]: `data.columns`

```
Out[82]: Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')
```

In [85]: `data.iloc[3]`

```
Out[85]: Age          28
Sex          F
BP          NORMAL
Cholesterol    HIGH
Na_to_K        7.798
Drug          drugX
Name: 3, dtype: object
```

```
In [86]: data.dropna(axis=1,how="any")
```

```
Out[86]:
```

	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

```
In [87]: data.index
```

```
Out[87]: RangeIndex(start=0, stop=200, step=1)
```

```
In [ ]:
```