**from** scipy.stats **import** binom

**import** matplotlib.pyplot **as** pp

In [2]:

n **=** 10

p **=** 0.8

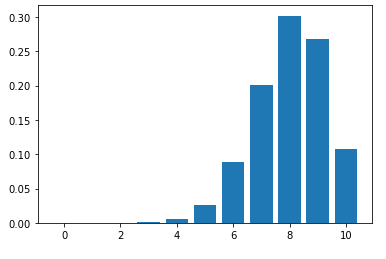
v **=** list(range(n**+**1))

dist **=** [binom**.**pmf(r,n,p) **for** r **in** v]

pp**.**bar(v,dist)

pp**.**show()

output:



In [5]:

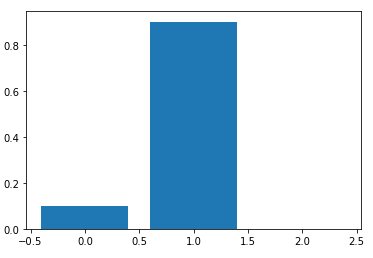
**from** scipy.stats **import** bernoulli

bd **=** bernoulli(0.9)

x **=** [0,1,2]

pp**.**bar(x,bd**.**pmf(x))

pp**.**show()

output: 

In [6]:

**import** matplotlib.pyplot **as** pp

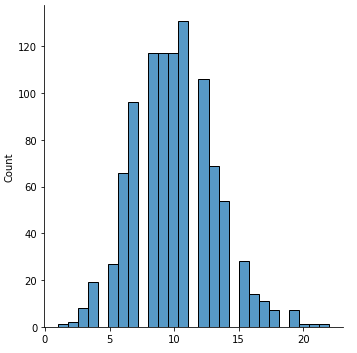
**import** seaborn **as** sn

**from** numpy **import** random

sn**.**displot(random**.**poisson(lam**=**10,size**=**1000))

pp**.**show()

output:



In [7]:

**import** numpy **as** np

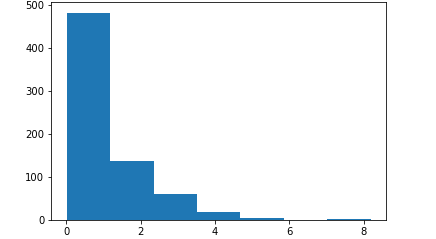
**import** matplotlib.pyplot **as** pp

exp **=** np**.**random**.**exponential(1,700)

count,bins,ignored **=** pp**.**hist(exp,7)

pp**.**show()

output:



In [13]:

**import** matplotlib.pyplot **as** pp

**import** numpy **as** np

mu,sigma **=** 0.5,0.1

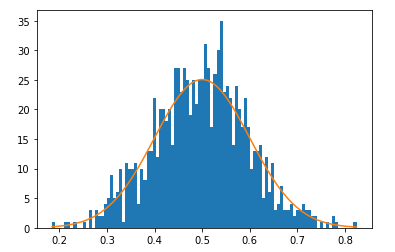
s **=** np**.**random**.**normal(mu,sigma,1000)

count,bins,ignored **=** pp**.**hist(s,100)

pp**.**plot(bins, 1**/**sigma**\***np**.**sqrt(2**\***np**.**pi)**\***np**.**exp(**-**(bins**-**mu)**\*\***2**/**(2**\***sigma**\*\***2)))

pp**.**show()

output:



DATASET 1

**import** numpy **as** np

**import** pandas **as** pd

In [31]:

data **=** pd**.**read\_csv(r"C:\Users\user\Desktop\7\_uber.csv")

In [34]:

df**=**pd**.**DataFrame(data)

df1**=**df**.**loc[[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]]

df1

Out[34]:

|  | **Unnamed: 0** | **key** | **fare\_amount** | **pickup\_datetime** | **pickup\_longitude** | **pickup\_latitude** | **dropoff\_longitude** | **dropoff\_latitude** | **passenger\_count** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 24238194 | 2015-05-07 19:52:06.0000003 | 7.5 | 2015-05-07 19:52:06 UTC | -73.999817 | 40.738354 | -73.999512 | 40.723217 | 1 |
| **1** | 27835199 | 2009-07-17 20:04:56.0000002 | 7.7 | 2009-07-17 20:04:56 UTC | -73.994355 | 40.728225 | -73.994710 | 40.750325 | 1 |
| **2** | 44984355 | 2009-08-24 21:45:00.00000061 | 12.9 | 2009-08-24 21:45:00 UTC | -74.005043 | 40.740770 | -73.962565 | 40.772647 | 1 |
| **3** | 25894730 | 2009-06-26 08:22:21.0000001 | 5.3 | 2009-06-26 08:22:21 UTC | -73.976124 | 40.790844 | -73.965316 | 40.803349 | 3 |
| **4** | 17610152 | 2014-08-28 17:47:00.000000188 | 16.0 | 2014-08-28 17:47:00 UTC | -73.925023 | 40.744085 | -73.973082 | 40.761247 | 5 |
| **5** | 44470845 | 2011-02-12 02:27:09.0000006 | 4.9 | 2011-02-12 02:27:09 UTC | -73.969019 | 40.755910 | -73.969019 | 40.755910 | 1 |
| **6** | 48725865 | 2014-10-12 07:04:00.0000002 | 24.5 | 2014-10-12 07:04:00 UTC | -73.961447 | 40.693965 | -73.871195 | 40.774297 | 5 |
| **7** | 44195482 | 2012-12-11 13:52:00.00000029 | 2.5 | 2012-12-11 13:52:00 UTC | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1 |
| **8** | 15822268 | 2012-02-17 09:32:00.00000043 | 9.7 | 2012-02-17 09:32:00 UTC | -73.975187 | 40.745767 | -74.002720 | 40.743537 | 1 |
| **9** | 50611056 | 2012-03-29 19:06:00.000000273 | 12.5 | 2012-03-29 19:06:00 UTC | -74.001065 | 40.741787 | -73.963040 | 40.775012 | 1 |
| **10** | 2205147 | 2015-05-22 17:32:27.0000004 | 6.5 | 2015-05-22 17:32:27 UTC | -73.974388 | 40.746952 | -73.988586 | 40.729805 | 1 |
| **11** | 6379048 | 2011-05-23 22:15:00.00000086 | 8.5 | 2011-05-23 22:15:00 UTC | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1 |
| **12** | 31892535 | 2011-05-17 14:03:00.000000158 | 3.3 | 2011-05-17 14:03:00 UTC | -73.966378 | 40.804440 | -73.965890 | 40.807133 | 5 |
| **13** | 13012786 | 2011-06-25 11:19:00.000000102 | 10.9 | 2011-06-25 11:19:00 UTC | -73.953352 | 40.767382 | -73.972510 | 40.796137 | 1 |
| **14** | 48411337 | 2010-04-06 22:20:27.0000004 | 6.9 | 2010-04-06 22:20:27 UTC | -73.973370 | 40.755193 | -73.978265 | 40.766375 | 1 |
| **15** | 46272151 | 2012-02-21 09:33:00.00000028 | 9.7 | 2012-02-21 09:33:00 UTC | -73.990718 | 40.751920 | -73.973053 | 40.744230 | 2 |

In [35]:

x**=**df1**.**mean()

print(x)

Unnamed: 0 3.078507e+07

fare\_amount 9.331250e+00

pickup\_longitude -6.472908e+01

pickup\_latitude 3.565660e+01

dropoff\_longitude -6.472372e+01

dropoff\_latitude 3.566895e+01

passenger\_count 1.937500e+00

dtype: float64

In [36]:

y**=**df1**.**median()

print(y)

Unnamed: 0 2.986387e+07

fare\_amount 8.100000e+00

pickup\_longitude -7.397388e+01

pickup\_latitude 4.074493e+01

dropoff\_longitude -7.397076e+01

dropoff\_latitude 4.075858e+01

passenger\_count 1.000000e+00

dtype: float64

In [37]:

z**=**df1**.**mode()

print(z)

Unnamed: 0 key fare\_amount \

0 2205147 2009-06-26 08:22:21.0000001 9.7

1 6379048 2009-07-17 20:04:56.0000002 NaN

2 13012786 2009-08-24 21:45:00.00000061 NaN

3 15822268 2010-04-06 22:20:27.0000004 NaN

4 17610152 2011-02-12 02:27:09.0000006 NaN

5 24238194 2011-05-17 14:03:00.000000158 NaN

6 25894730 2011-05-23 22:15:00.00000086 NaN

7 27835199 2011-06-25 11:19:00.000000102 NaN

8 31892535 2012-02-17 09:32:00.00000043 NaN

9 44195482 2012-02-21 09:33:00.00000028 NaN

10 44470845 2012-03-29 19:06:00.000000273 NaN

11 44984355 2012-12-11 13:52:00.00000029 NaN

12 46272151 2014-08-28 17:47:00.000000188 NaN

13 48411337 2014-10-12 07:04:00.0000002 NaN

14 48725865 2015-05-07 19:52:06.0000003 NaN

15 50611056 2015-05-22 17:32:27.0000004 NaN

pickup\_datetime pickup\_longitude pickup\_latitude \

0 2009-06-26 08:22:21 UTC 0.0 0.0

1 2009-07-17 20:04:56 UTC NaN NaN

2 2009-08-24 21:45:00 UTC NaN NaN

3 2010-04-06 22:20:27 UTC NaN NaN

4 2011-02-12 02:27:09 UTC NaN NaN

5 2011-05-17 14:03:00 UTC NaN NaN

6 2011-05-23 22:15:00 UTC NaN NaN

7 2011-06-25 11:19:00 UTC NaN NaN

8 2012-02-17 09:32:00 UTC NaN NaN

9 2012-02-21 09:33:00 UTC NaN NaN

10 2012-03-29 19:06:00 UTC NaN NaN

11 2012-12-11 13:52:00 UTC NaN NaN

12 2014-08-28 17:47:00 UTC NaN NaN

13 2014-10-12 07:04:00 UTC NaN NaN

14 2015-05-07 19:52:06 UTC NaN NaN

15 2015-05-22 17:32:27 UTC NaN NaN

dropoff\_longitude dropoff\_latitude passenger\_count

0 0.0 0.0 1.0

1 NaN NaN NaN

2 NaN NaN NaN

3 NaN NaN NaN

4 NaN NaN NaN

5 NaN NaN NaN

6 NaN NaN NaN

7 NaN NaN NaN

8 NaN NaN NaN

9 NaN NaN NaN

10 NaN NaN NaN

11 NaN NaN NaN

12 NaN NaN NaN

13 NaN NaN NaN

14 NaN NaN NaN

15 NaN NaN NaN

In [38]:

a**=**df1**.**sum()

print(a)

Unnamed: 0 492561150

key 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

fare\_amount 149.3

pickup\_datetime 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

pickup\_longitude -1035.665286

pickup\_latitude 570.505594

dropoff\_longitude -1035.579463

dropoff\_latitude 570.703221

passenger\_count 31

dtype: object

In [39]:

b**=**df1**.**describe()

print(b)

Unnamed: 0 fare\_amount pickup\_longitude pickup\_latitude \

count 1.600000e+01 16.000000 16.000000 16.000000

mean 3.078507e+07 9.331250 -64.729080 35.656600

std 1.640473e+07 5.413775 25.267654 13.918933

min 2.205147e+06 2.500000 -74.005043 0.000000

25% 1.716318e+07 6.200000 -73.991627 40.735822

50% 2.986387e+07 8.100000 -73.973879 40.744926

75% 4.530630e+07 11.300000 -73.959423 40.755372

max 5.061106e+07 24.500000 0.000000 40.804440

dropoff\_longitude dropoff\_latitude passenger\_count

count 16.000000 16.000000 16.000000

mean -64.723716 35.668951 1.937500

std 25.265569 13.923754 1.611159

min -74.002720 0.000000 1.000000

25% -73.980845 40.740104 1.000000

50% -73.970765 40.758578 1.000000

75% -73.962921 40.774476 2.250000

max 0.000000 40.807133 5.000000

In [40]:

b**=**df1**.**cumsum()

print(b)

Unnamed: 0 key \

0 24238194 2015-05-07 19:52:06.0000003

1 52073393 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

2 97057748 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

3 122952478 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

4 140562630 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

5 185033475 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

6 233759340 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

7 277954822 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

8 293777090 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

9 344388146 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

10 346593293 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

11 352972341 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

12 384864876 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

13 397877662 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

14 446288999 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

15 492561150 2015-05-07 19:52:06.00000032009-07-17 20:04:56...

fare\_amount pickup\_datetime \

0 7.5 2015-05-07 19:52:06 UTC

1 15.2 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC

2 28.1 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

3 33.4 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

4 49.4 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

5 54.3 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

6 78.8 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

7 81.3 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

8 91.0 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

9 103.5 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

10 110.0 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

11 118.5 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

12 121.8 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

13 132.7 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

14 139.6 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

15 149.3 2015-05-07 19:52:06 UTC2009-07-17 20:04:56 UTC...

pickup\_longitude pickup\_latitude dropoff\_longitude dropoff\_latitude \

0 -73.999817 40.738354 -73.999512 40.723217

1 -147.994172 81.466579 -147.994222 81.473542

2 -221.999215 122.207349 -221.956787 122.246189

3 -295.975339 162.998193 -295.922103 163.049538

4 -369.900362 203.742278 -369.895185 203.810785

5 -443.869381 244.498188 -443.864204 244.566695

6 -517.830828 285.192153 -517.735399 285.340992

7 -517.830828 285.192153 -517.735399 285.340992

8 -591.806015 325.937920 -591.738119 326.084529

9 -665.807080 366.679707 -665.701159 366.859541

10 -739.781468 407.426659 -739.689745 407.589346

11 -739.781468 407.426659 -739.689745 407.589346

12 -813.747846 448.231099 -813.655635 448.396479

13 -887.701198 488.998481 -887.628145 489.192616

14 -961.674568 529.753674 -961.606410 529.958991

15 -1035.665286 570.505594 -1035.579463 570.703221

passenger\_count

0 1

1 2

2 3

3 6

4 11

5 12

6 17

7 18

8 19

9 20

10 21

11 22

12 27

13 28

14 29

15 31

In [41]:

c**=**df1**.**count()

print(c)

Unnamed: 0 16

key 16

fare\_amount 16

pickup\_datetime 16

pickup\_longitude 16

pickup\_latitude 16

dropoff\_longitude 16

dropoff\_latitude 16

passenger\_count 16

dtype: int64

In [42]:

d**=**df1**.**min()

print(d)

Unnamed: 0 2205147

key 2009-06-26 08:22:21.0000001

fare\_amount 2.5

pickup\_datetime 2009-06-26 08:22:21 UTC

pickup\_longitude -74.005043

pickup\_latitude 0.0

dropoff\_longitude -74.00272

dropoff\_latitude 0.0

passenger\_count 1

dtype: object

In [43]:

e**=**df1**.**max()

print(e)

Unnamed: 0 50611056

key 2015-05-22 17:32:27.0000004

fare\_amount 24.5

pickup\_datetime 2015-05-22 17:32:27 UTC

pickup\_longitude 0.0

pickup\_latitude 40.80444

dropoff\_longitude 0.0

dropoff\_latitude 40.807133

passenger\_count 5

dtype: object

In [44]:

**from** numpy **import** cov

In [47]:

print(cov(df1['passenger\_count'],df1['fare\_amount']))

[[ 2.59583333 3.70208333]

[ 3.70208333 29.30895833]]

**import** matplotlib.pyplot **as** pp

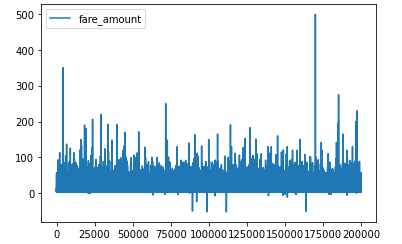
In [55]:

df**=**data[['key','fare\_amount']]

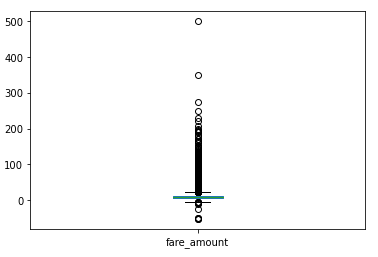
df**.**plot**.**line()

Out[55]:

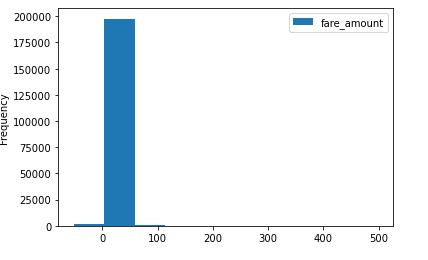
<AxesSubplot:>



df**.**plot**.**box()



df**.**plot**.**hist()



df1**=**data[['passenger\_count','dropoff\_latitude']]

df1**.**plot**.**area()



df1**.**isna()

Out[64]:

|  | **passenger\_count** | **dropoff\_latitude** |
| --- | --- | --- |
| **0** | False | False |
| **1** | False | False |
| **2** | False | False |
| **3** | False | False |
| **4** | False | False |
| **...** | ... | ... |
| **199995** | False | False |
| **199996** | False | False |
| **199997** | False | False |
| **199998** | False | False |
| **199999** | False | False |

200000 rows × 2 columns

In [66]:

df1**.**dropna

Out[66]:

<bound method DataFrame.dropna of passenger\_count dropoff\_latitude

0 1 40.723217

1 1 40.750325

2 1 40.772647

3 3 40.803349

4 5 40.761247

... ... ...

199995 1 40.740297

199996 1 40.739620

199997 2 40.692588

199998 1 40.695415

199999 1 40.768793

[200000 rows x 2 columns]>

DATASET 2

**import** numpy **as** np

**import** pandas **as** pd

In [67]:

data **=** pd**.**read\_csv(r"C:\Users\user\Desktop\3\_Fitness-1.csv")

In [69]:

df**=**pd**.**DataFrame(data)

df1**=**df**.**loc[[0,1,2,3,4,5,6,7,8]]

df1

Out[69]:

|  | **Row Labels** | **Sum of Jan** | **Sum of Feb** | **Sum of Mar** | **Sum of Total Sales** |
| --- | --- | --- | --- | --- | --- |
| **0** | A | 5.62% | 7.73% | 6.16% | 75 |
| **1** | B | 4.21% | 17.27% | 19.21% | 160 |
| **2** | C | 9.83% | 11.60% | 5.17% | 101 |
| **3** | D | 2.81% | 21.91% | 7.88% | 127 |
| **4** | E | 25.28% | 10.57% | 11.82% | 179 |
| **5** | F | 8.15% | 16.24% | 18.47% | 167 |
| **6** | G | 18.54% | 8.76% | 17.49% | 171 |
| **7** | H | 25.56% | 5.93% | 13.79% | 170 |
| **8** | Grand Total | 100.00% | 100.00% | 100.00% | 1150 |

In [70]:

x**=**df1**.**mean()

print(x)

Sum of Total Sales 255.555556

dtype: float64

In [71]:

y**=**df1**.**median()

print(y)

Sum of Total Sales 167.0

dtype: float64

In [72]:

z**=**df1**.**mode()

print(z)

Row Labels Sum of Jan Sum of Feb Sum of Mar Sum of Total Sales

0 A 100.00% 10.57% 100.00% 75

1 B 18.54% 100.00% 11.82% 101

2 C 2.81% 11.60% 13.79% 127

3 D 25.28% 16.24% 17.49% 160

4 E 25.56% 17.27% 18.47% 167

5 F 4.21% 21.91% 19.21% 170

6 G 5.62% 5.93% 5.17% 171

7 Grand Total 8.15% 7.73% 6.16% 179

8 H 9.83% 8.76% 7.88% 1150

In [73]:

a**=**df1**.**sum()

print(a)

Row Labels ABCDEFGHGrand Total

Sum of Jan 5.62%4.21%9.83%2.81%25.28%8.15%18.54%25.56%100...

Sum of Feb 7.73%17.27%11.60%21.91%10.57%16.24%8.76%5.93%1...

Sum of Mar 6.16%19.21%5.17%7.88%11.82%18.47%17.49%13.79%1...

Sum of Total Sales 2300

dtype: object

In [74]:

b**=**df1**.**describe()

print(b)

Sum of Total Sales

count 9.000000

mean 255.555556

std 337.332963

min 75.000000

25% 127.000000

50% 167.000000

75% 171.000000

max 1150.000000

In [75]:

b**=**df1**.**cumsum()

print(b)

Row Labels Sum of Jan \

0 A 5.62%

1 AB 5.62%4.21%

2 ABC 5.62%4.21%9.83%

3 ABCD 5.62%4.21%9.83%2.81%

4 ABCDE 5.62%4.21%9.83%2.81%25.28%

5 ABCDEF 5.62%4.21%9.83%2.81%25.28%8.15%

6 ABCDEFG 5.62%4.21%9.83%2.81%25.28%8.15%18.54%

7 ABCDEFGH 5.62%4.21%9.83%2.81%25.28%8.15%18.54%25.56%

8 ABCDEFGHGrand Total 5.62%4.21%9.83%2.81%25.28%8.15%18.54%25.56%100...

Sum of Feb \

0 7.73%

1 7.73%17.27%

2 7.73%17.27%11.60%

3 7.73%17.27%11.60%21.91%

4 7.73%17.27%11.60%21.91%10.57%

5 7.73%17.27%11.60%21.91%10.57%16.24%

6 7.73%17.27%11.60%21.91%10.57%16.24%8.76%

7 7.73%17.27%11.60%21.91%10.57%16.24%8.76%5.93%

8 7.73%17.27%11.60%21.91%10.57%16.24%8.76%5.93%1...

Sum of Mar Sum of Total Sales

0 6.16% 75

1 6.16%19.21% 235

2 6.16%19.21%5.17% 336

3 6.16%19.21%5.17%7.88% 463

4 6.16%19.21%5.17%7.88%11.82% 642

5 6.16%19.21%5.17%7.88%11.82%18.47% 809

6 6.16%19.21%5.17%7.88%11.82%18.47%17.49% 980

7 6.16%19.21%5.17%7.88%11.82%18.47%17.49%13.79% 1150

8 6.16%19.21%5.17%7.88%11.82%18.47%17.49%13.79%1... 2300

In [76]:

c**=**df1**.**count()

print(c)

Row Labels 9

Sum of Jan 9

Sum of Feb 9

Sum of Mar 9

Sum of Total Sales 9

dtype: int64

In [77]:

d**=**df1**.**min()

print(d)

Row Labels A

Sum of Jan 100.00%

Sum of Feb 10.57%

Sum of Mar 100.00%

Sum of Total Sales 75

dtype: object

In [78]:

e**=**df1**.**max()

print(e)

Row Labels H

Sum of Jan 9.83%

Sum of Feb 8.76%

Sum of Mar 7.88%

Sum of Total Sales 1150

dtype: object

In [79]:

**from** numpy **import** cov

In [81]:

print(cov(df1['Sum of Total Sales'],df1['Sum of Total Sales']))

[[113793.52777778 113793.52777778]

[113793.52777778 113793.52777778]]

In [51]:

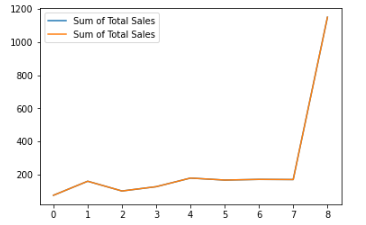
**import** matplotlib.pyplot **as** pp

In [82]:

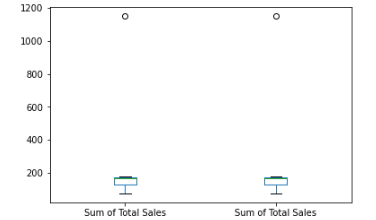
df**=**data[['Sum of Total Sales','Sum of Total Sales']]

df**.**plot**.**line()

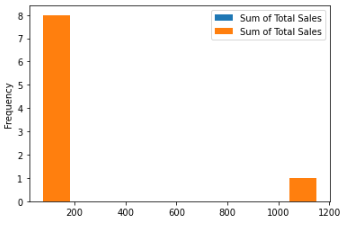
Out[82]:



df**.**plot**.**box()

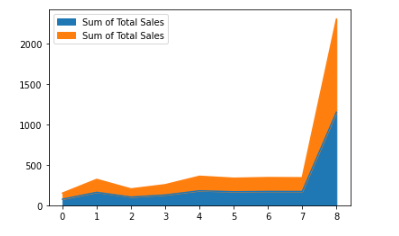


df**.**plot**.**hist()



df1**=**data[['Sum of Total Sales','Sum of Total Sales']]

df1**.**plot**.**area()



df1**.**isna()

| **Sum of Total Sales** | **Sum of Total Sales** |
| --- | --- |
| **0** | False | False |
| **1** | False | False |
| **2** | False | False |
| **3** | False | False |
| **4** | False | False |
| **5** | False | False |
| **6** | False | False |
| **7** | False | False |
| **8** | False | False |

In [87]:

df1**.**dropna

Out[87]:

<bound method DataFrame.dropna of Sum of Total Sales Sum of Total Sales

0 75 75

1 160 160

2 101 101

3 127 127

4 179 179

5 167 167

6 171 171

7 170 170

8 1150 1150>

DATASET 3

**import** numpy **as** np

**import** pandas **as** pd

In [88]:

data **=** pd**.**read\_csv(r"C:\Users\user\Desktop\4\_drug200.csv")

In [89]:

df**=**pd**.**DataFrame(data)

df1**=**df**.**loc[[0,1,2,3,4,5,6,7,8]]

df1

Out[89]:

|  | **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 |

In [90]:

x**=**df1**.**mean()

print(x)

Age 42.000000

Na\_to\_K 13.943667

dtype: float64

In [91]:

y**=**df1**.**median()

print(y)

Age 47.000

Na\_to\_K 13.093

dtype: float64

In [92]:

z**=**df1**.**mode()

print(z)

Age Sex BP Cholesterol Na\_to\_K Drug

0 47.0 F LOW HIGH 7.798 drugY

1 NaN NaN NORMAL NaN 8.607 NaN

2 NaN NaN NaN NaN 10.114 NaN

3 NaN NaN NaN NaN 11.037 NaN

4 NaN NaN NaN NaN 13.093 NaN

5 NaN NaN NaN NaN 15.171 NaN

6 NaN NaN NaN NaN 16.275 NaN

7 NaN NaN NaN NaN 18.043 NaN

8 NaN NaN NaN NaN 25.355 NaN

In [93]:

a**=**df1**.**sum()

print(a)

Age 378

Sex FMMFFFFMM

BP HIGHLOWLOWNORMALLOWNORMALNORMALLOWNORMAL

Cholesterol HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGH

Na\_to\_K 125.493

Drug drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugCdrugY

dtype: object

In [94]:

b**=**df1**.**describe()

print(b)

Age Na\_to\_K

count 9.000000 9.000000

mean 42.000000 13.943667

std 14.756355 5.523274

min 22.000000 7.798000

25% 28.000000 10.114000

50% 47.000000 13.093000

75% 49.000000 16.275000

max 61.000000 25.355000

In [95]:

b**=**df1**.**cumsum()

print(b)

Age Sex BP \

0 23 F HIGH

1 70 FM HIGHLOW

2 117 FMM HIGHLOWLOW

3 145 FMMF HIGHLOWLOWNORMAL

4 206 FMMFF HIGHLOWLOWNORMALLOW

5 228 FMMFFF HIGHLOWLOWNORMALLOWNORMAL

6 277 FMMFFFF HIGHLOWLOWNORMALLOWNORMALNORMAL

7 318 FMMFFFFM HIGHLOWLOWNORMALLOWNORMALNORMALLOW

8 378 FMMFFFFMM HIGHLOWLOWNORMALLOWNORMALNORMALLOWNORMAL

Cholesterol Na\_to\_K \

0 HIGH 25.355

1 HIGHHIGH 38.448

2 HIGHHIGHHIGH 48.562

3 HIGHHIGHHIGHHIGH 56.360

4 HIGHHIGHHIGHHIGHHIGH 74.403

5 HIGHHIGHHIGHHIGHHIGHHIGH 83.010

6 HIGHHIGHHIGHHIGHHIGHHIGHHIGH 99.285

7 HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGH 110.322

8 HIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGHHIGH 125.493

Drug

0 drugY

1 drugYdrugC

2 drugYdrugCdrugC

3 drugYdrugCdrugCdrugX

4 drugYdrugCdrugCdrugXdrugY

5 drugYdrugCdrugCdrugXdrugYdrugX

6 drugYdrugCdrugCdrugXdrugYdrugXdrugY

7 drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugC

8 drugYdrugCdrugCdrugXdrugYdrugXdrugYdrugCdrugY

In [96]:

c**=**df1**.**count()

print(c)

Age 9

Sex 9

BP 9

Cholesterol 9

Na\_to\_K 9

Drug 9

dtype: int64

In [97]:

d**=**df1**.**min()

print(d)

Age 22

Sex F

BP HIGH

Cholesterol HIGH

Na\_to\_K 7.798

Drug drugC

dtype: object

In [98]:

e**=**df1**.**max()

print(e)

Age 61

Sex M

BP NORMAL

Cholesterol HIGH

Na\_to\_K 25.355

Drug drugY

dtype: object

In [99]:

**from** numpy **import** cov

In [101]:

print(cov(df1['Age'],df1['Na\_to\_K']))

[[217.75 8.970125 ]

[ 8.970125 30.50655575]]

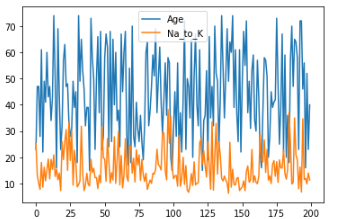
In [102]:

**import** matplotlib.pyplot **as** pp

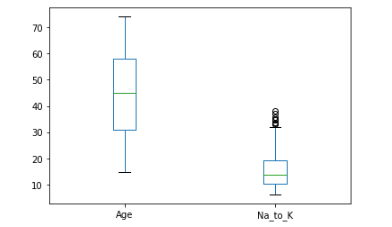
In [103]:

df**=**data[['Age','Na\_to\_K']]

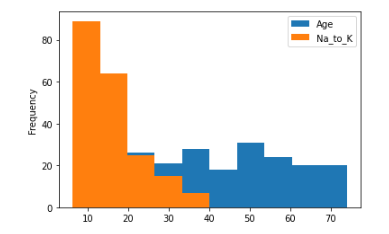
df**.**plot**.**line()



df**.**plot**.**box()

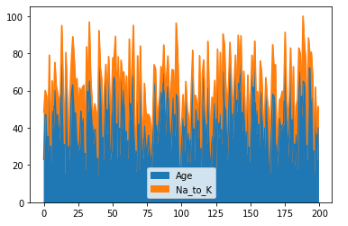


df**.**plot**.**hist()



df1**=**data[['Age','Na\_to\_K']]

df1**.**plot**.**area()



df1**.**isna()

Out[109]:

|  | **Age** | **Na\_to\_K** |
| --- | --- | --- |
| **0** | False | False |
| **1** | False | False |
| **2** | False | False |
| **3** | False | False |
| **4** | False | False |
| **...** | ... | ... |
| **195** | False | False |
| **196** | False | False |
| **197** | False | False |
| **198** | False | False |
| **199** | False | False |

200 rows × 2 columns

In [110]:

df1**.**dropna

Out[110]:

<bound method DataFrame.dropna of 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 x 2 columns]>