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

data = pd.read_csv('C:/Users/Sony Vaio/Downloads/data.csv')
```

In [2]:
```
m=data['Maxpulse']
n=data['Calories']
c=data['Pulse']
len(m)
```

Out[2]:
```
169
```

In [3]:
```
data
```

Out[3]:

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0
...	...	...	...	...
164	60	105	140	290.8
165	60	110	145	300.0
166	60	115	145	310.2
167	75	120	150	320.4
168	75	125	150	330.4

169 rows × 4 columns

In [4]:
```
len(data['Maxpulse'])
```

Out[4]:
```
169
```

In [5]:
```
data['Maxpulse']
```

Out[5]:
```
0      130
1      145
2      135
3      175
4      148
      ...
164    140
165    145
166    145
167    150
168    150
Name: Maxpulse, Length: 169, dtype: int64
```

In [6]:
```
m.sum()
```

Out[6]:
```
22654
```

In [ ]:
```
```

In [7]:
```
np.mean(m)
```

Out[7]:
```
134.0473372781065
```

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

Out[8]:
```
count    169.000000
mean     134.047337
std       16.450434
min      100.000000
25%      124.000000
50%      131.000000
75%      141.000000
max      184.000000
Name: Maxpulse, dtype: float64
```

In [9]:
```
M=[0,3,4,1,6,7]
mu=np.mean(M)
var_list=[]
s=0
t=0
for i in M:
        s=s+i**2/len(M)
        t=t+i/len(M)
        l=t**2
        res=np.sqrt(s-l)

   # var_list.append((np.sqrt((mu**2)/sum(M)-sum(M)**2)))
print(s)
print(l)
print(res)
print(np.std(M))
```

```
18.5
12.25
2.5
2.5
```

In [10]:
```
#       $$ sigmaf(x) = 1 $$
```

In [11]:
```
r=np.random.randint(0,10)
n=50
p=0.5
b=np.random.binomial(n,p,50)
```

In [12]:
```
p=b/50
p
```

Out[12]:
```
array([0.5 , 0.48, 0.4 , 0.36, 0.46, 0.46, 0.56, 0.76, 0.44, 0.52, 0.56,
       0.6 , 0.48, 0.52, 0.4 , 0.54, 0.5 , 0.5 , 0.5 , 0.46, 0.52, 0.58,
       0.62, 0.42, 0.52, 0.46, 0.3 , 0.58, 0.28, 0.52, 0.42, 0.62, 0.42,
       0.5 , 0.6 , 0.58, 0.34, 0.48, 0.46, 0.6 , 0.4 , 0.4 , 0.54, 0.44,
       0.54, 0.5 , 0.44, 0.54, 0.46, 0.6 ])
```

In [13]:
```
y=p/np.sum(p)
```

In [14]:
```
np.sum(y)
```

Out[14]:
```
1.0
```

In [15]:
```
np.sum(y)
```

Out[15]:
```
1.0
```

In [16]:
```
mu=0
for i in range(len(b)):
    mu=mu+b[i]*p[i]
mu
```

Out[16]:
```
628.28
```

In [17]:
```
var=0
for i in range(len(b)):
    var=var+(b[i]-mu)**2*p[i]
var
```

Out[17]:
```
8969066.040512001
```

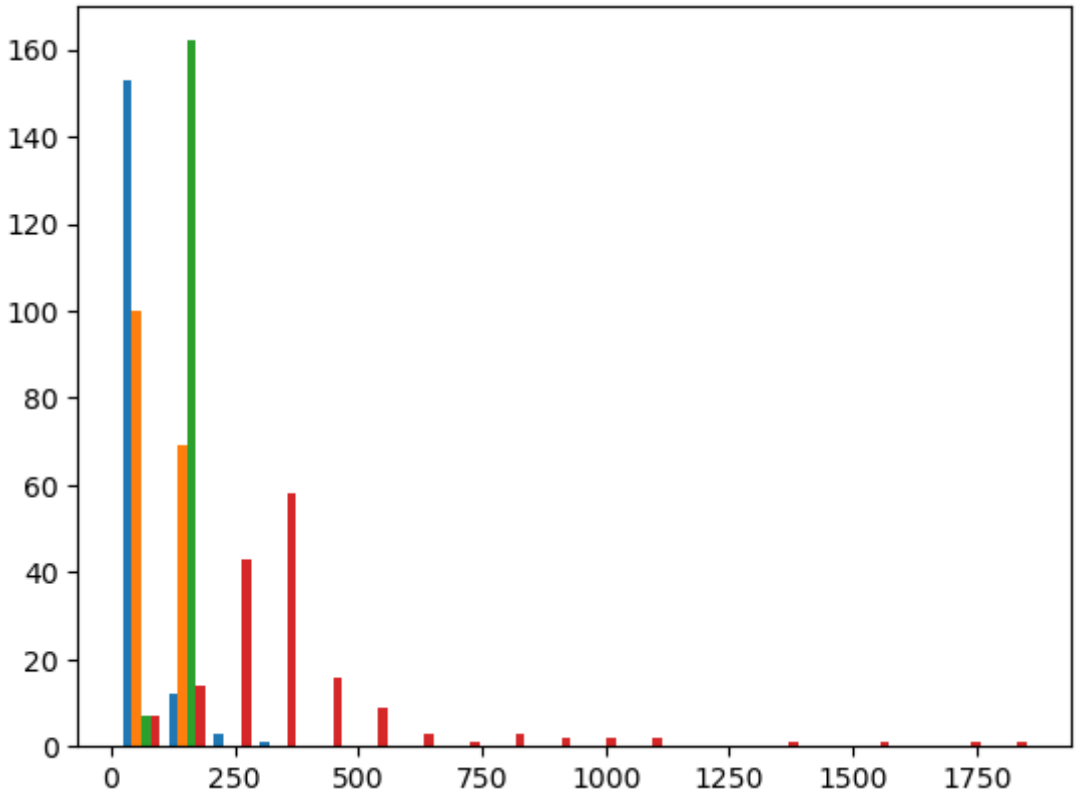
In [18]:
```
np.mean(b)
```

Out[18]:
```
24.68
```

In [19]:
```
np.var(b)
```

Out[19]:
```
19.1776
```

In [20]:
```
plt.hist(data,bins=20)
plt.show()
```



In [21]:
```
plt.scatter(m,n,c=c)
plt.xlabel('maxplus')
plt.ylabel('calories')
```

```
plt.colorbar(label='Pulse')
plt.show()
```

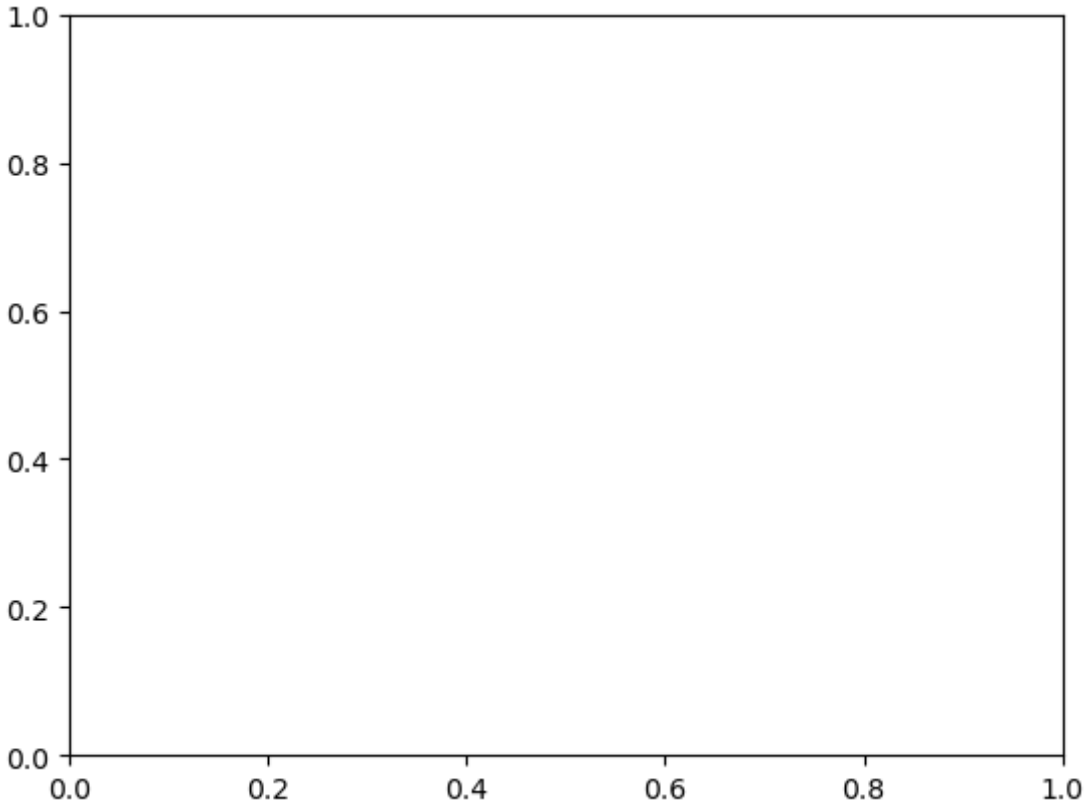
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[21], line 1
----> 1 plt.scatter(m,n,c=c)
      2 plt.xlabel('maxplus')
      3 plt.ylabel('calories')

File ~\anaconda3\lib\site-packages\matplotlib\pyplot.py:2835, in scatter(x, y, s, c, marker, cmap, norm, vmin, vmax, alpha, linewidths, edgecolors, plotnonfinite, data, **kwargs)
   2830 @_copy_docstring_and_deprecators(Axes.scatter)
   2831 def scatter(
   2832         x, y, s=None, c=None, marker=None, cmap=None, norm=None,
   2833         vmin=None, vmax=None, alpha=None, linewidths=None, *,
   2834         edgecolors=None, plotnonfinite=False, data=None, **kwargs):
-> 2835     __ret = gca().scatter(
   2836         x, y, s=s, c=c, marker=marker, cmap=cmap, norm=norm,
   2837         vmin=vmin, vmax=vmax, alpha=alpha, linewidths=linewidths,
   2838         edgecolors=edgecolors, plotnonfinite=plotnonfinite,
   2839         **({"data": data} if data is not None else {}), **kwargs)
   2840     sci(__ret)
   2841     return __ret

File ~\anaconda3\lib\site-packages\matplotlib\__init__.py:1442, in _preprocess_data.<locals>.inner(ax, data, *args, **kwargs)
   1439 @functools.wraps(func)
   1440 def inner(ax, *args, data=None, **kwargs):
   1441     if data is None:
-> 1442         return func(ax, *map(sanitize_sequence, args), **kwargs)
   1444     bound = new_sig.bind(ax, *args, **kwargs)
   1445     auto_label = (bound.arguments.get(label_namer)
   1446                   or bound.kwargs.get(label_namer))

File ~\anaconda3\lib\site-packages\matplotlib\axes\_axes.py:4584, in Axes.scatter(self, x, y, s, c, marker, cmap, norm, vmin, vmax, alpha, linewidths, edgecolors, plotnonfinite, **kwargs)
   4582 y = np.ma.ravel(y)
   4583 if x.size != y.size:
-> 4584     raise ValueError("x and y must be the same size")
   4586 if s is None:
   4587     s = (20 if mpl.rcParams['_internal.classic_mode'] else
   4588          mpl.rcParams['lines.markersize'] ** 2.0)

ValueError: x and y must be the same size
```



In [ ]:

In [ ]:
```
def bank(L,x):
    y = ((L**x)*np.exp(-L))/np.math.factorial(x)
    return y
```

In [ ]:
```
bank(6,5)
```

In [ ]:
```
def tedist():
    x=280
    m=300
    s=50
    n=15
    k=s/np.sqrt(n)
    t=(x-m)/k
    return t
```

In [ ]:
```
tedist()
```

centrallimittheorem

In [ ]:
```
msize=[]
x = np.random.uniform(size=(1000))
#plt.hist(x)
s=[]
for i in range(1000):
    #sam=np.random.choice(x,100).mean()
    sam=np.random.uniform(0,10,59).mean()
    s.append(sam)

plt.hist(s)
```

In [ ]:
```
x = np.random.binomial(n=10, p=0.5, size=2000)
y=np.mean(x)
plt.hist(y,bins=20)
plt.show()
print(x)
print(y)
```

In [ ]:
```
msize=[]
x = np.random.binomial(10,0.5,10)
#plt.hist(x)
s=[]
for i in range(1000):
    #sam=np.random.choice(x,100).mean()
    sam=np.random.binomial(10,0.5,59).mean()
    s.append(sam)

plt.hist(s)
```

In [ ]:
```

```

In [ ]:
```

```