Statistics |Histogram Implementation

- **Dr. Mostafa Elhosseini**
- https://youtube.com/drmelhosseini (https://youtube.com/drmelhosseini)

In [1]:

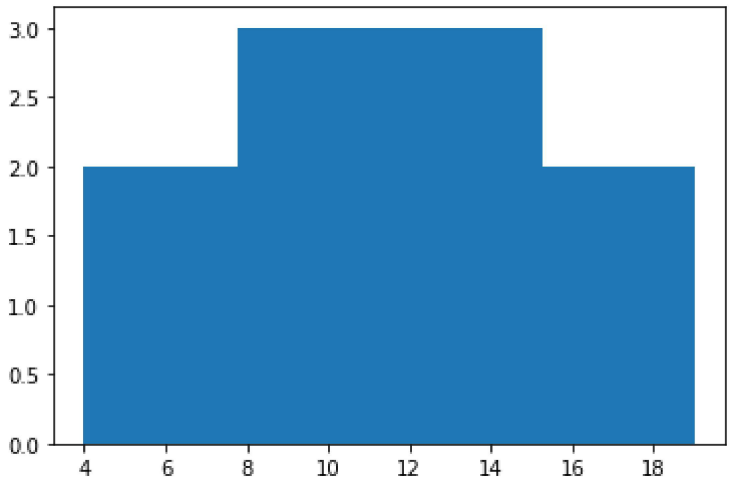
```
import pandas as pd
import numpy as np
import sklearn
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
x = np.array([4,5,8,8,9,13,13,13,17,19])
```

In [3]:

```
n,bins,patches = plt.hist(x,bins = 4)
```



How to Get endpoints of the Buckets (bins)

- Calculate first (max - min)/#bins = (19-4)/4 = 3.75
- First bin range = min to (min+...) but not included

In [4]:

```
n
```

Out[4]:

```
array([2., 3., 3., 2.])
```

In [5]:

```
bins
```

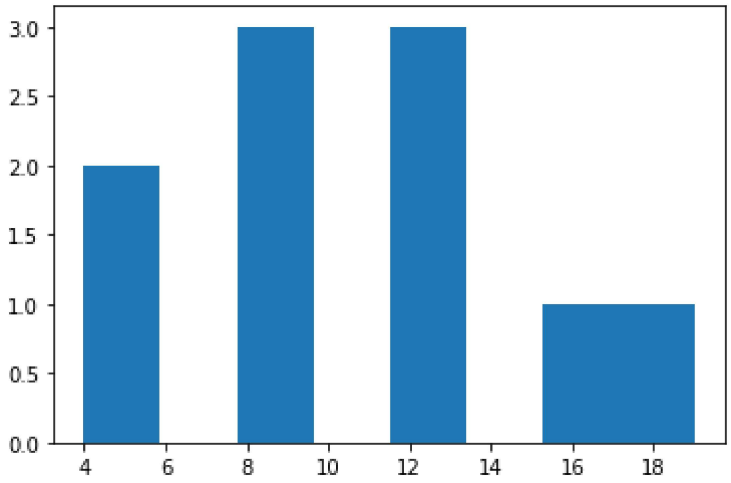
Out[5]:

```
array([ 4.  ,  7.75, 11.5 , 15.25, 19.  ])
```

Changing No. of Buckets

In [6]:

```
n,bins,patches = plt.hist(x,bins = 8)    # (19-4)/8
```



In [7]:

```
n
```

Out[7]:

```
array([2., 0., 3., 0., 3., 0., 1., 1.])
```

In [8]:

```
bins
```

Out[8]:

```
array([ 4.   ,  5.875,  7.75 ,  9.625, 11.5  , 13.375, 15.25 , 17.125,
       19.   ])
```

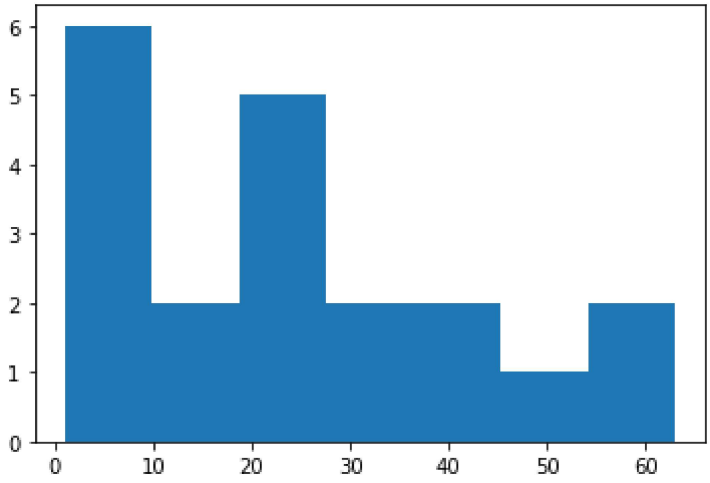
Example - Restaurant

In [9]:

```
x = np.array([1,3,27,32,5,63,26,25,18,16,4,45,29,19,22,51,58,9,42,6])
```

In [10]:

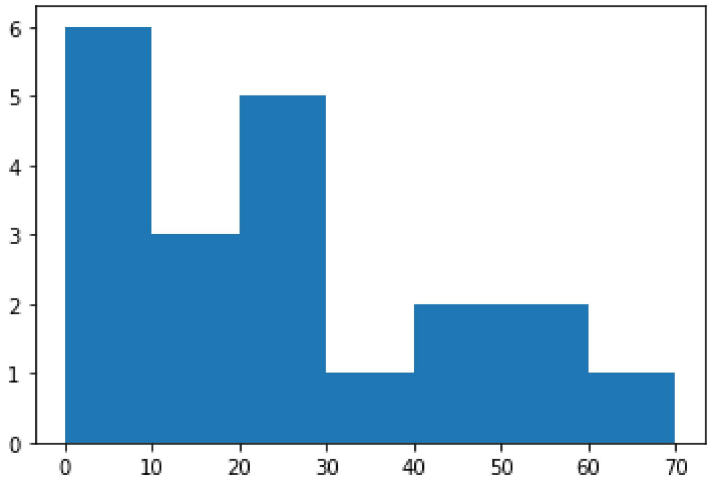
```
n,bins,patches = plt.hist(x,bins = 7)    #
```



Manually Assign Bucket Endpoints

In [11]:

```
n,bins,patches = plt.hist(x,bins = [0,10,20,30,40,50,60,70])    # same result as in hand wr
```



In [12]:

```
n
```

Out[12]:

```
array([6., 3., 5., 1., 2., 2., 1.])
```

In [13]:

```
bins
```

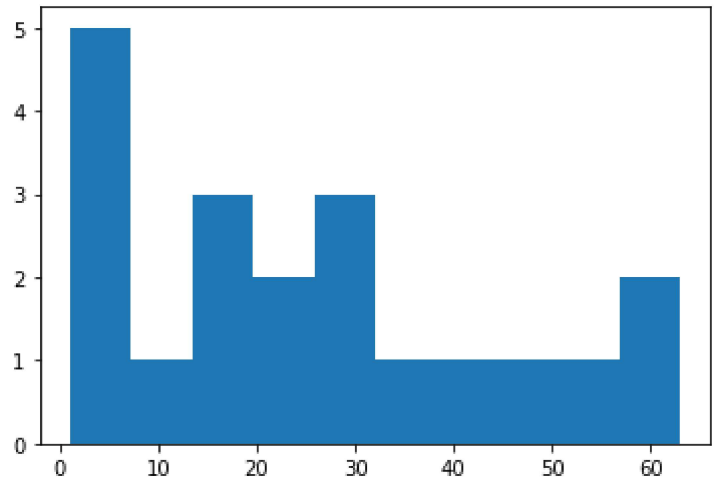
Out[13]:

```
array([ 0, 10, 20, 30, 40, 50, 60, 70])
```

Get The Optimal Setting For The bins

In [14]:

```
n,bins,patches = plt.hist(x)
```



In [15]:

```
n
```

Out[15]:

```
array([5., 1., 3., 2., 3., 1., 1., 1., 1., 2.])
```

In [16]:

```
bins      # 10
```

Out[16]:

```
array([ 1. ,  7.2, 13.4, 19.6, 25.8, 32. , 38.2, 44.4, 50.6, 56.8, 63. ])
```

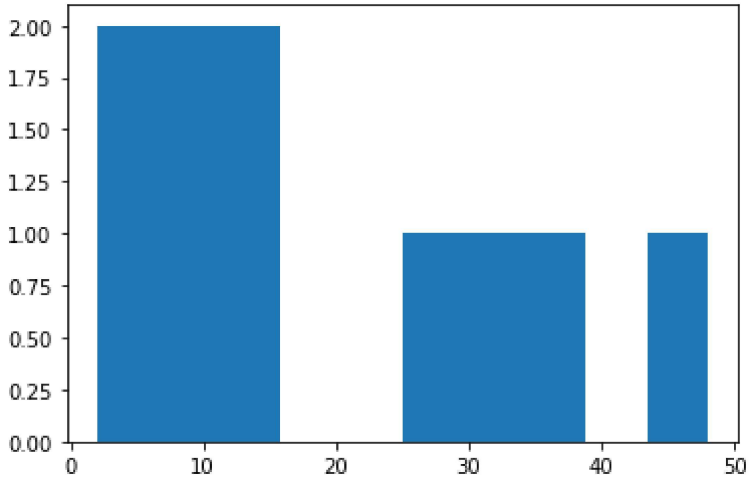
Example

In [17]:

```
x = np.random.randint(0,50,10)
```

In [18]:

```
n,bins,patches = plt.hist(x)
```



In [19]:

```
x
```

Out[19]:

```
array([ 2, 48, 28,  3, 12, 32,  8,  9, 36, 14])
```

In [20]:

```
n
```

Out[20]:

```
array([2., 2., 2., 0., 0., 1., 1., 1., 0., 1.])
```

In [21]:

```
bins
```

Out[21]:

```
array([ 2. ,  6.6, 11.2, 15.8, 20.4, 25. , 29.6, 34.2, 38.8, 43.4, 48. ])
```

Example

In [22]:

```
x = np.array([4,5,8,8,9,13,13,13,17,19])
```

In [23]:

```
n, bins = np.histogram(x,bins=4)
```

In [24]:

```
x
```

Out[24]:

```
array([ 4,  5,  8,  8,  9, 13, 13, 13, 17, 19])
```

In [25]:

```
n
```

Out[25]:

```
array([2, 3, 3, 2], dtype=int64)
```

In [26]:

```
bins
```

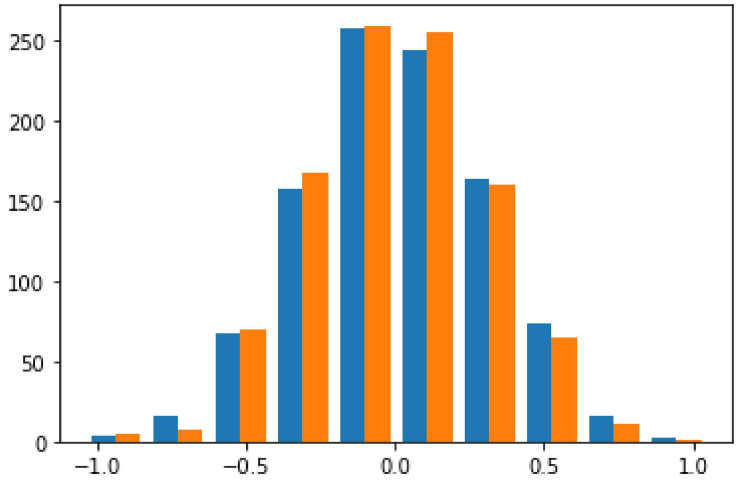
Out[26]:

```
array([ 4.  ,  7.75, 11.5 , 15.25, 19.  ])
```

Plot Two Histogram

In [27]:

```
x = 0.3*np.random.randn(1000)
y = 0.3*np.random.randn(1000)
n, bins, patches = plt.hist([x, y])
```



Plotting histogram of Iris data using Pandas

In [28]:

```
from sklearn.datasets import load_iris
import pandas as pd
```

In [29]:

```
data = load_iris().data
```

In [30]:

```
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width']
```

In [31]:

```
dataset = pd.DataFrame(data,columns=names)
```

In [32]:

```
dataset.head()
```

Out[32]:

	sepal-length	sepal-width	petal-length	petal-width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [33]:

```
dataset.columns[0]
```

Out[33]:

```
'sepal-length'
```

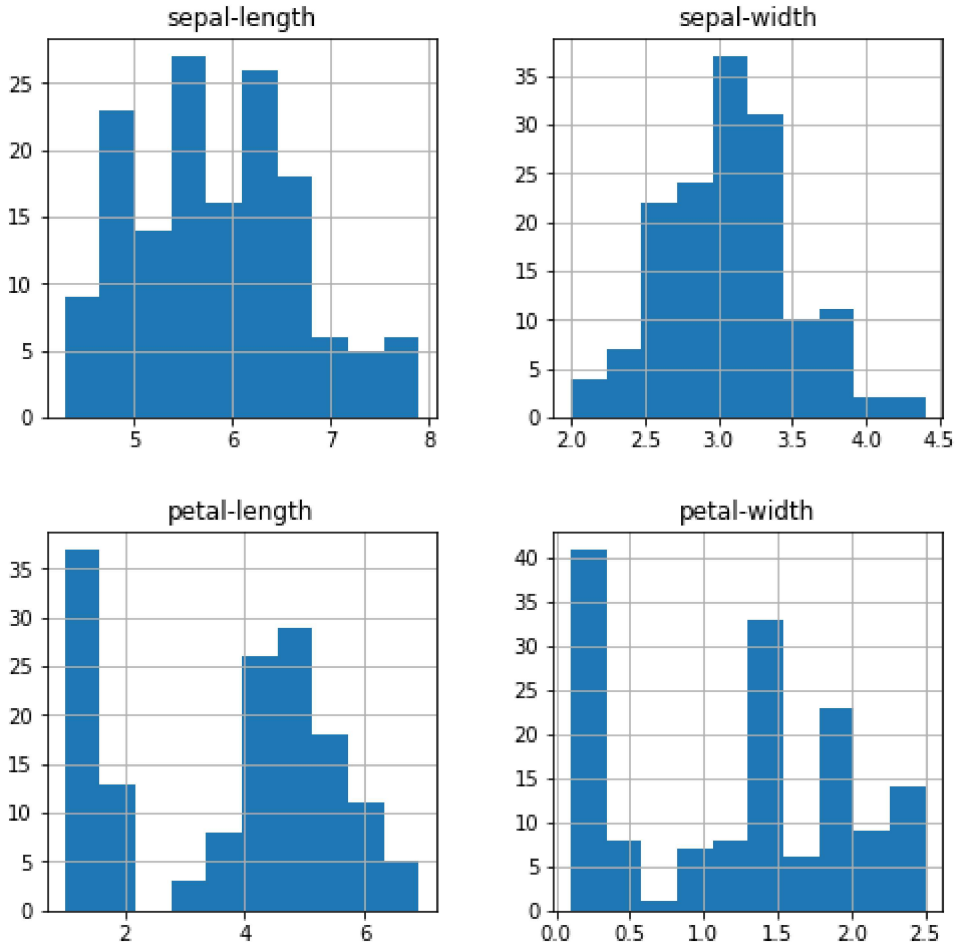
In [34]:

```
fig = plt.figure(figsize = (8,8))
ax = fig.gca()
dataset.hist(ax=ax)
```

```
<ipython-input-34-d001eeb18978>:3: UserWarning: To output multiple subplots,
the figure containing the passed axes is being cleared
  dataset.hist(ax=ax)
```

Out[34]:

```
array([[<AxesSubplot:title={'center':'sepal-length'}>,
        <AxesSubplot:title={'center':'sepal-width'}>],
       [<AxesSubplot:title={'center':'petal-length'}>,
        <AxesSubplot:title={'center':'petal-width'}>]], dtype=object)
```



Dataset

- The best way to avoid mistakes in buying clothes through online shops.
- You can check your predictions using this dataset
- The Dataset consists of 4 columns:
  - Weight (in kgs)
  - Age
  - Height (in cm)
  - Size
- The output parameter is size and input parameters are Weight, Age, and Height.

In [35]:

```
df = pd.read_csv("ClothesSizePrediction.csv")
```

In [36]:

```
df.head()
```

Out[36]:

	weight	age	height	size
0	62	28.0	172.72	XL
1	59	36.0	167.64	L
2	61	34.0	165.10	M
3	65	27.0	175.26	L
4	62	45.0	172.72	M

Histogram

- Grouping Observation into ranges
- Increase or decrease bins, gives us a better idea of what the distribution looks like

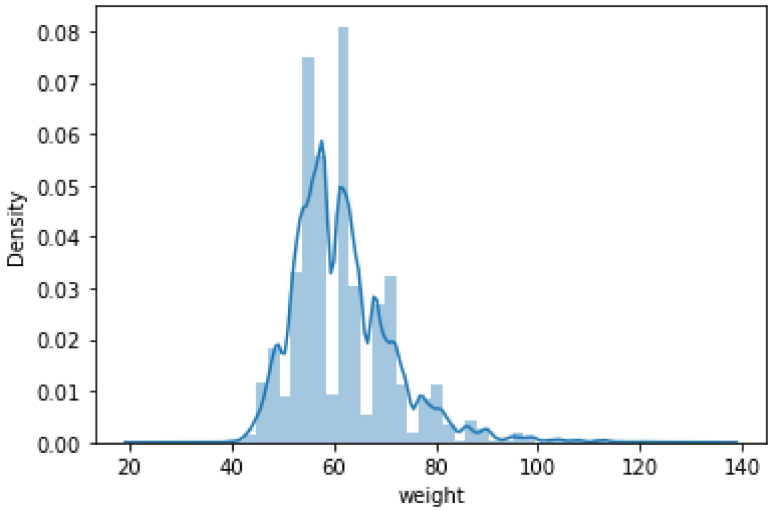
In [37]:

```
sns.distplot(df["weight"])
```

```
C:\Users\melho\anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[37]:

```
<AxesSubplot:xlabel='weight', ylabel='Density'>
```



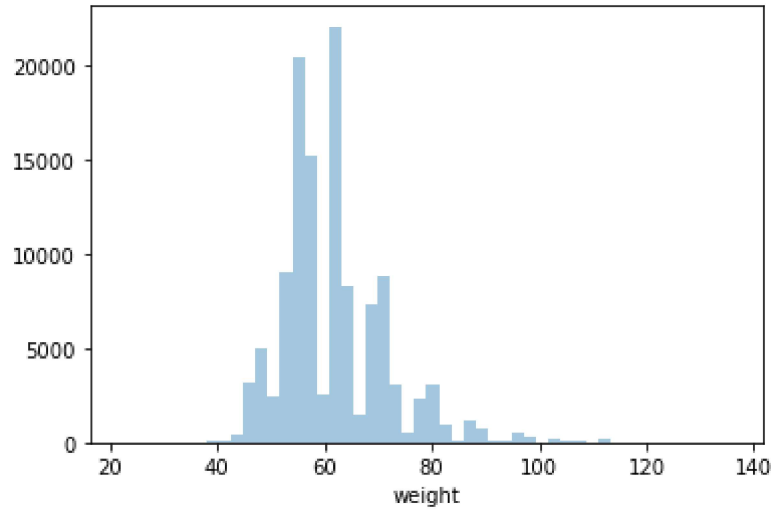
In [38]:

```
sns.distplot(df["weight"],kde = False)  # kernal density function
```

```
C:\Users\melho\anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[38]:

```
<AxesSubplot:xlabel='weight'>
```

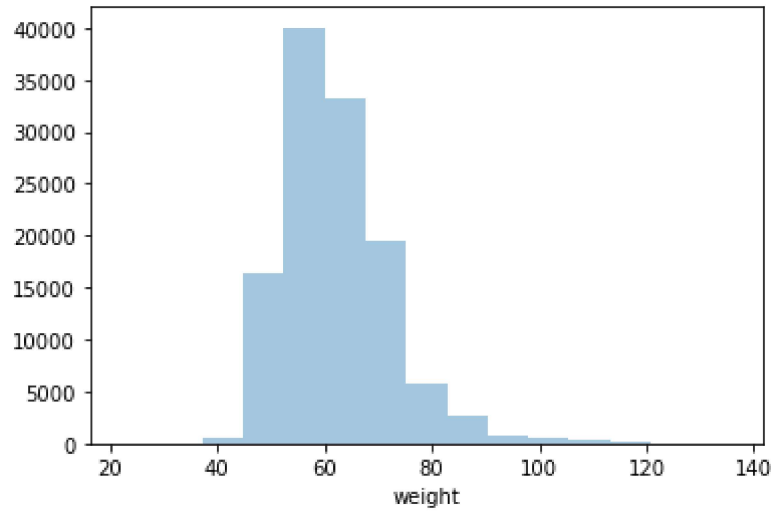


In [39]:

```
sns.distplot(df["weight"],kde = False,bins = 15)
```

Out[39]:

```
<AxesSubplot:xlabel='weight'>
```



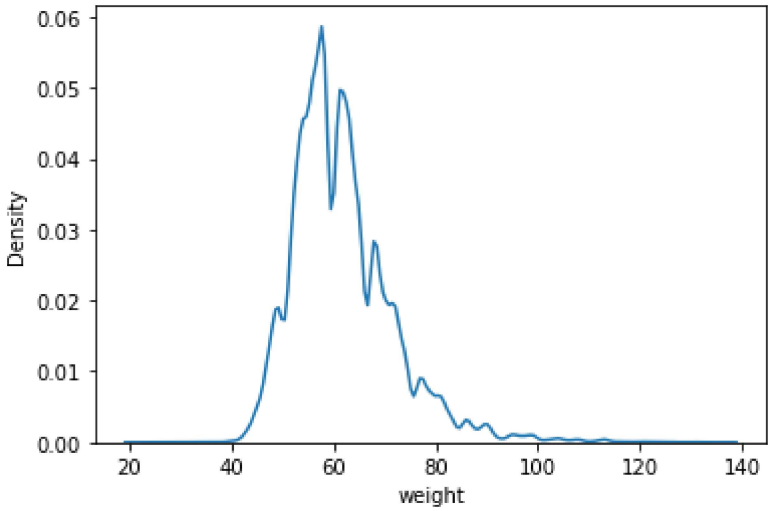
In [40]:

```
sns.distplot(df["weight"], hist = False )
```

```
C:\Users\melho\anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `kdeplot` (an axes-level function for
kernel density plots).
  warnings.warn(msg, FutureWarning)
```

Out[40]:

```
<AxesSubplot:xlabel='weight', ylabel='Density'>
```

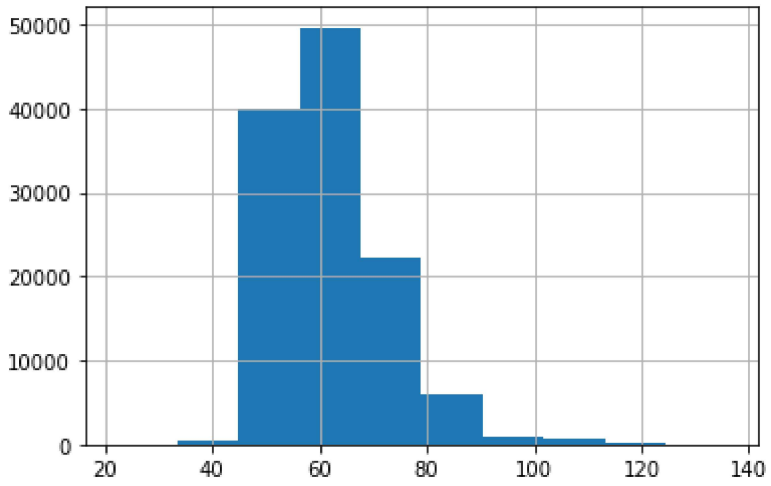


In [41]:

```
df["weight"].hist()
```

Out[41]:

<AxesSubplot:>

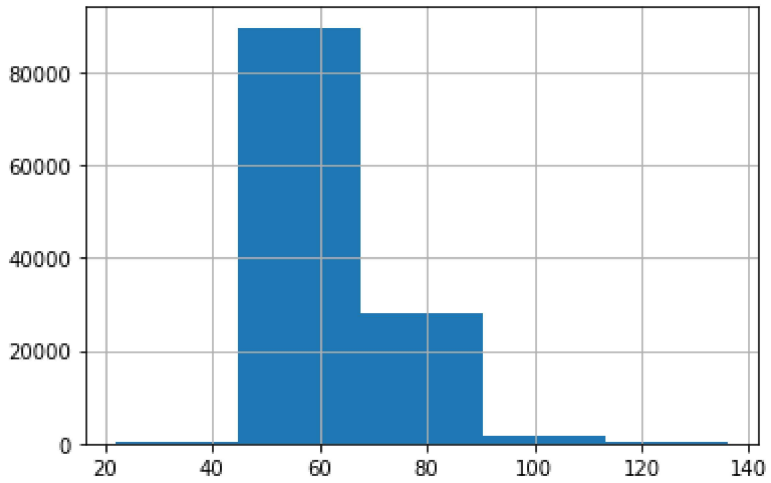


In [42]:

```
df["weight"].hist(bins = 5)
```

Out[42]:

<AxesSubplot:>

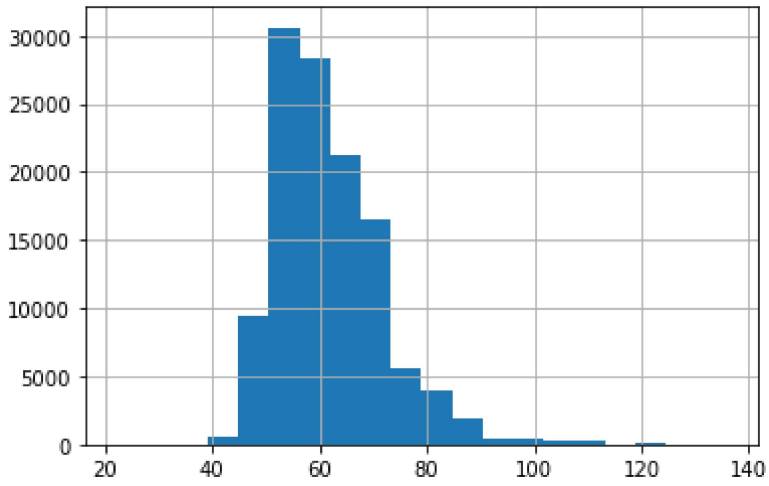


In [43]:

```
df["weight"].hist(bins = 20)
```

Out[43]:

<AxesSubplot:>

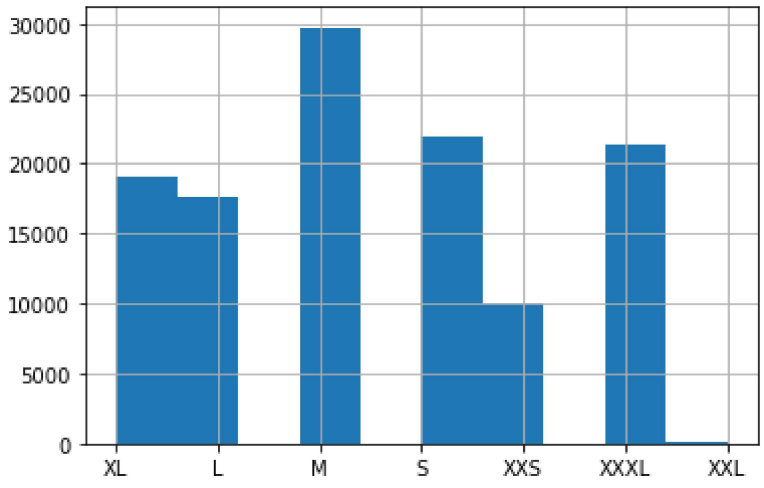


In [44]:

```
df["size"].hist()
```

Out[44]:

<AxesSubplot:>



References

- seaborn: statistical data visualization: https://seaborn.pydata.org/index.html (https://seaborn.pydata.org/index.html)
- A Waiter's Tips: https://www.kaggle.com/jsphyg/tipping (https://www.kaggle.com/jsphyg/tipping)