Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Exe11 - Normal Distribution Exercise

Name: Phua Yan Han

IC Number: 050824070059

Date : 10/7/23

Introduction :

Conclusion :

Normal Distribution

The normal distribution is defined by the following probability density function, where μ is the population mean and σ^2 is the variance.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

If a random variable X follows the normal distribution, then we write:

$$X \sim N(\mu, \sigma^2)$$

In particular, the normal distribution with $\mu = 0$ and $\sigma = 1$ is called the `standard normal distribution` , and is denoted as `N(0,1`). It can be graphed as follows.

The normal distribution is important because of the **Central Limit Theorem**, which states that the population of all possible samples of size n from a population with mean μ and variance σ^2 approaches a normal distribution with mean μ and $\frac{\sigma^2}{n}$ when n approaches infinity.

Read and understand more about **Central Limit Theorem (CLT)** [here (https://statisticsbyjim.com/basics/central-limit-theorem/)](https://statisticsbyjim.com/basics/central-limit-theorem/)

Question 1

Suppose widgit weights produced at MS Widgit Works have weights that are normally distributed with mean 17.46 grams and variance 375.67 grams. What is the probability that a randomly chosen widgit weighs more then 19 grams?

In [11]:
```
from scipy.stats import norm
import math
from scipy.stats import stats
1-norm.cdf(19,17.46,math.sqrt(375.67))
```

Out[11]:
```
0.46833563578991133
```

Question 2

Suppose IQ scores are normally distributed with mean 100 and standard deviation 15. What is the 95th percentile of the distribution of IQ scores?

In [15]:
```
secret=norm.ppf(0.95)
100 + secret * 15
```

Out[15]:
```
124.67280440427209
```

Question 3

Suppose wages are normally distributed with a mean of 1900 and a standard deviation of 150.

1. What percentage of people have wages less than 1800?
2. What percentage of people have wages greater than 2100?
3. What percentage of people have wages between 1800 and 2100?
4. What wages separate the top 10% from the others?
5. What wages separate the lower 25% from the others?

In [27]:
```
# 1
secret1=norm.cdf(1800,1900,150)
print(secret1)
# 2
secret2=1-norm.cdf(2100,1900,150)
print(secret2)
# 3
print(norm.cdf(2100,1900,150)-secret1)
# 4
percentile = norm.ppf(0.90)
print(2100 + percentile * 150)
# 5
percentile2 = norm.ppf(0.25)
print(2100 + percentile2 * 150)
```

```
0.2524925375469229
0.09121121972586788
0.6562962427272092
2292.23273483169
1998.8265374705877
```

Question 4

Based on the Ages of Death during the Spanish Flu, 1918.

Demonstration of the central limit theorem, using the distribution of sample mean age at death in samples from a highly non-normal distribution: the frequency distribution of age at death in Switzerland in 1918 during the Spanish flu epidemic.

In [30]:
```
## Question 4

import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

path="http://whitlockschluter.zoology.ubc.ca/wp-content/data/chapter10/chap10e
flu = pd.read_csv(path)
flu
```

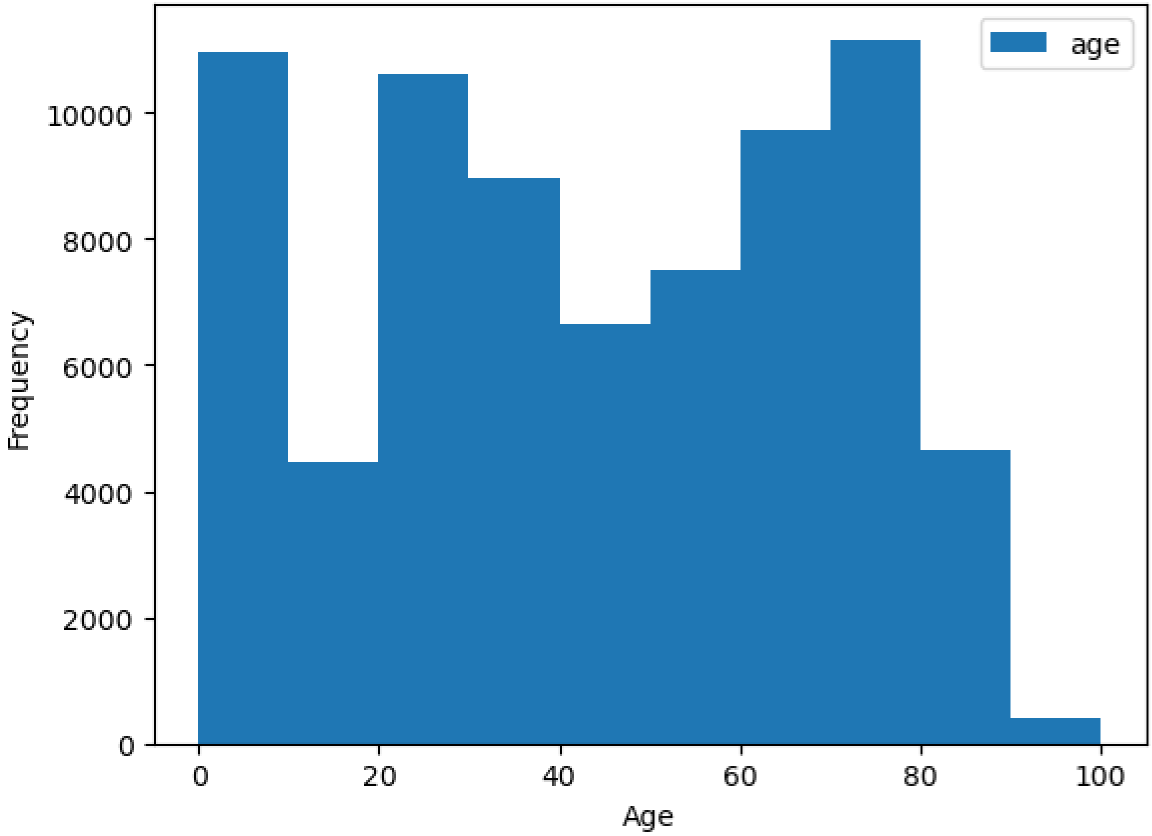
Out[30]:

	age
0	0
1	0
2	0
3	0
4	0
...	...
75029	98
75030	99
75031	99
75032	99
75033	100

75034 rows × 1 columns

Histogram showing the frequency distribution of ages at death in Switzerland in 1918 during the Spanish flu epidemic.

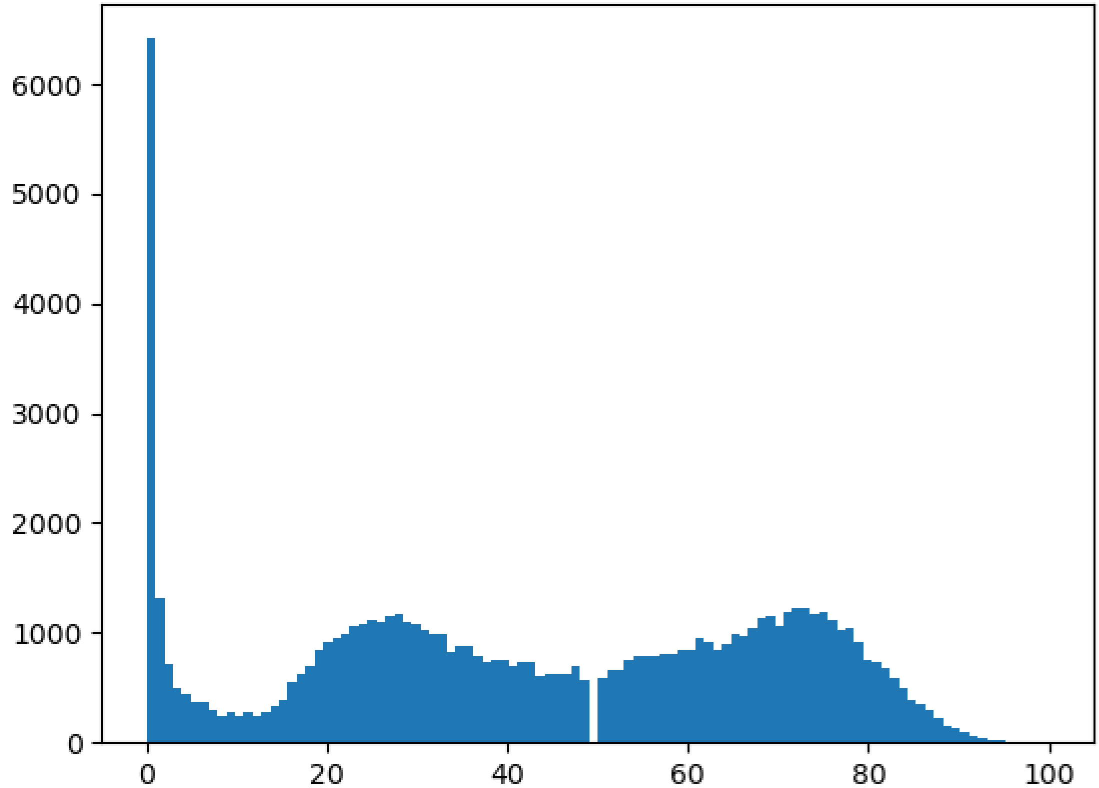
In [38]:
```
flu.plot.hist()
plt.xlabel('Age')
plt.show()
```



Histogram with better binning (0,102,2) and axis labels

In [41]:
```
plt.hist(flu['age'], bins=102)
```

Out[41]:
```
(array([6.409e+03, 1.307e+03, 7.230e+02, 5.040e+02, 4.450e+02, 3.680e+02,
        3.680e+02, 3.000e+02, 2.480e+02, 2.710e+02, 2.330e+02, 2.700e+02,
        2.380e+02, 2.810e+02, 3.350e+02, 3.960e+02, 5.440e+02, 6.200e+02,
        6.960e+02, 8.400e+02, 9.230e+02, 9.440e+02, 9.860e+02, 1.062e+03,
        1.076e+03, 1.117e+03, 1.105e+03, 1.144e+03, 1.166e+03, 1.100e+03,
        1.075e+03, 1.027e+03, 9.920e+02, 9.970e+02, 8.320e+02, 8.770e+02,
        8.720e+02, 7.850e+02, 7.330e+02, 7.570e+02, 7.510e+02, 6.890e+02,
        7.270e+02, 7.380e+02, 5.980e+02, 6.220e+02, 6.290e+02, 6.170e+02,
        6.970e+02, 5.740e+02, 0.000e+00, 5.910e+02, 6.610e+02, 6.560e+02,
        7.450e+02, 7.940e+02, 7.910e+02, 7.920e+02, 8.140e+02, 7.990e+02,
        8.490e+02, 8.480e+02, 9.470e+02, 9.090e+02, 8.390e+02, 8.910e+02,
        9.940e+02, 9.660e+02, 1.039e+03, 1.135e+03, 1.154e+03, 1.054e+03,
        1.190e+03, 1.229e+03, 1.221e+03, 1.171e+03, 1.189e+03, 1.112e+03,
        1.030e+03, 1.039e+03, 9.220e+02, 7.530e+02, 7.270e+02, 6.770e+02,
        5.790e+02, 5.010e+02, 3.920e+02, 3.540e+02, 2.990e+02, 2.240e+02,
        1.440e+02, 1.330e+02, 1.000e+02, 5.600e+01, 4.300e+01, 3.000e+01,
        1.600e+01, 1.100e+01, 6.000e+00, 6.000e+00, 3.000e+00, 1.000e+00]),
 array([  0.        ,   0.98039216,   1.96078431,   2.94117647,
          3.92156863,   4.90196078,   5.88235294,   6.8627451 ,
          7.84313725,   8.82352941,   9.80392157,  10.78431373,
         11.76470588,  12.74509804,  13.7254902 ,  14.70588235,
         15.68627451,  16.66666667,  17.64705882,  18.62745098,
         19.60784314,  20.58823529,  21.56862745,  22.54901961,
         23.52941176,  24.50980392,  25.49019608,  26.47058824,
         27.45098039,  28.43137255,  29.41176471,  30.39215686,
         31.37254902,  32.35294118,  33.33333333,  34.31372549,
         35.29411765,  36.2745098 ,  37.25490196,  38.23529412,
         39.21568627,  40.19607843,  41.17647059,  42.15686275,
         43.1372549 ,  44.11764706,  45.09803922,  46.07843137,
         47.05882353,  48.03921569,  49.01960784,  50.        ,
         50.98039216,  51.96078431,  52.94117647,  53.92156863,
         54.90196078,  55.88235294,  56.8627451 ,  57.84313725,
         58.82352941,  59.80392157,  60.78431373,  61.76470588,
         62.74509804,  63.7254902 ,  64.70588235,  65.68627451,
         66.66666667,  67.64705882,  68.62745098,  69.60784314,
         70.58823529,  71.56862745,  72.54901961,  73.52941176,
         74.50980392,  75.49019608,  76.47058824,  77.45098039,
         78.43137255,  79.41176471,  80.39215686,  81.37254902,
         82.35294118,  83.33333333,  84.31372549,  85.29411765,
         86.2745098 ,  87.25490196,  88.23529412,  89.21568627,
         90.19607843,  91.17647059,  92.15686275,  93.1372549 ,
         94.11764706,  95.09803922,  96.07843137,  97.05882353,
         98.03921569,  99.01960784, 100.        ]),
 <BarContainer object of 102 artists>)
```



Demonstrate the **central limit theorem**. Treat the age at death measurements from Switzerland in 1918 as the population. Take a large number of random samples, each of size n, from the population of age at death measurements and plot the sample means.

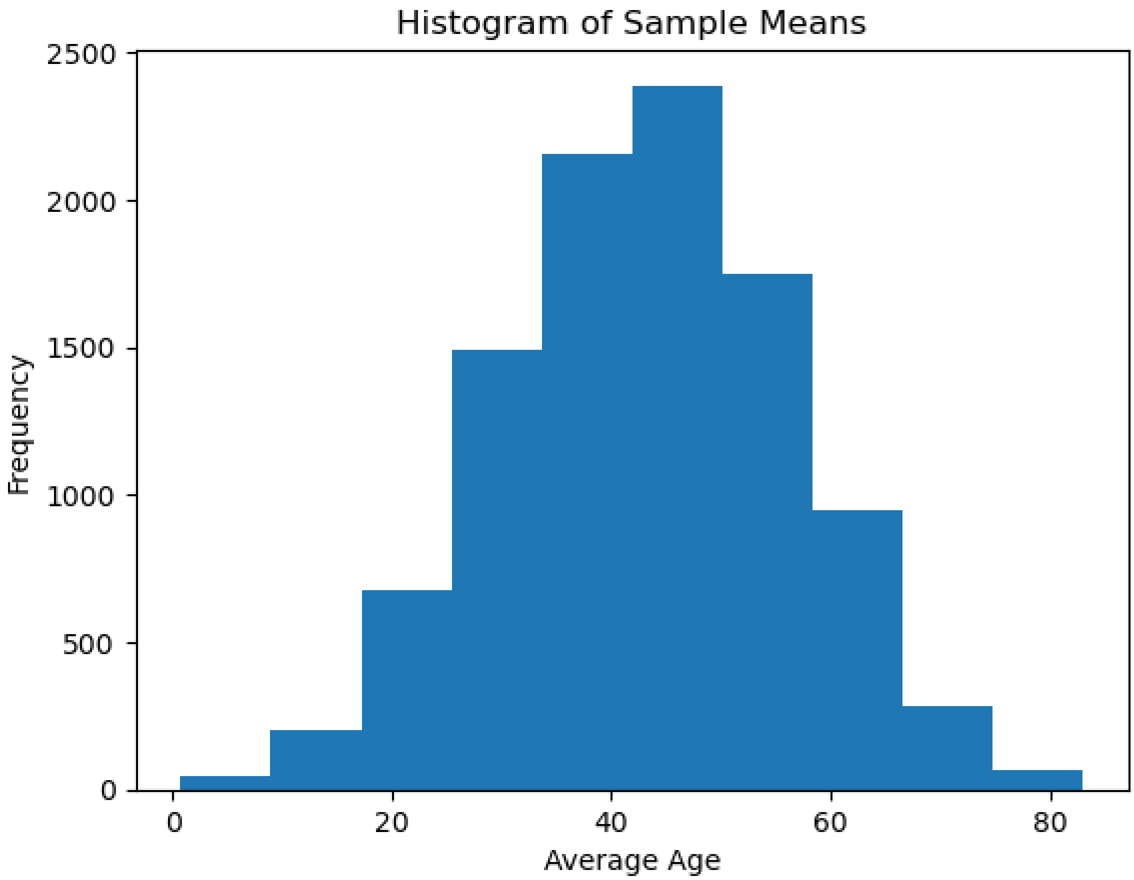
Note: your results won't be the identical to the one shown below, because 10,000 random samples is not large enough for extreme accuracy. Change the n below to another number and rerun to see the effects of sample size on the shape of the distribution of sample means.

Write a loop to sample 10000 times from 'Age'. Each time, collect 4 samples. Store the average age in a new variable, `age1` . Plot the histogram for `age1` .

In [46]:
```
import numpy as np
age_data = flu['age']
sample_means = []
num_samples = 10000
sample_size = 4

for _ in range(num_samples):
    sample = np.random.choice(age_data, size=sample_size)
    sample_mean = np.mean(sample)
    sample_means.append(sample_mean)

plt.hist(sample_means)
plt.xlabel('Average Age')
plt.ylabel('Frequency')
plt.title('Histogram of Sample Means')
plt.show()
```



Histogram of the sample means with more options

In [52]:
```
plt.hist(sample_means,bins=50,edgecolor='black',color='red')
plt.xlabel('Average Age')
plt.ylabel('Frequency')
plt.title('Histogram of Sample Means')
plt.show()
```