Session 9: Simulation Modeling I

1. Generating Samples using scipy.stats

[1]:
```
from scipy.stats import norm
import numpy as np
dist=norm(100,30)
dist.rvs()
```

```
87.21850625120882
```

[2]:
```
demand=dist.rvs(size=10000)
demand
```

```
array([102.83514787, 105.12600165,  89.09890044, ...,  98.66273053,
        43.925369  ,  75.18899873])
```

[3]:
```
demand.mean()
```

```
99.65021791057396
```

[4]:
```
demand.std()
```

```
30.05704039862063
```

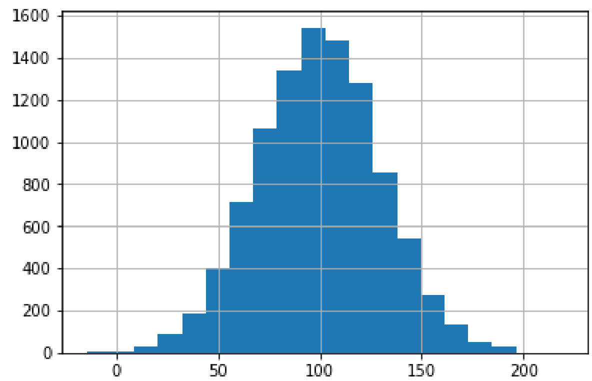
[5]:
```
demand<100
```

```
array([False, False,  True, ...,  True,  True,  True])
```

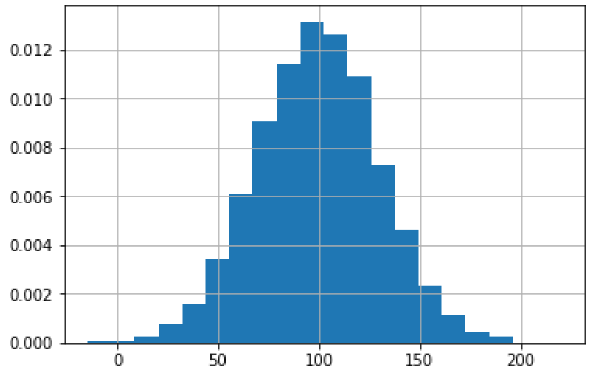
[6]:
```
(demand<100).mean()
```

```
0.5035
```

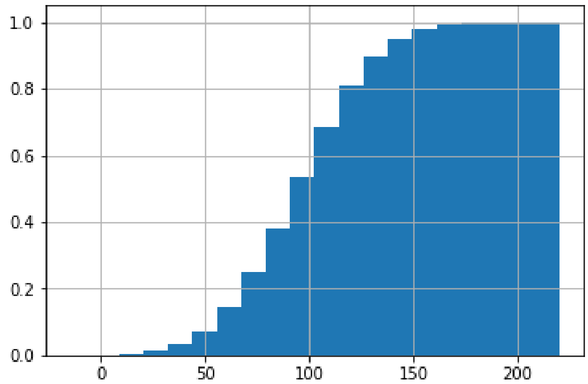
[25]:
```
import pandas as pd
demand=pd.Series(demand)
demand.hist(bins=20)
```



[8]:
```
# Density (Histogram normalized so that area under curve is one)
demand.hist(bins=20,density=True)
```



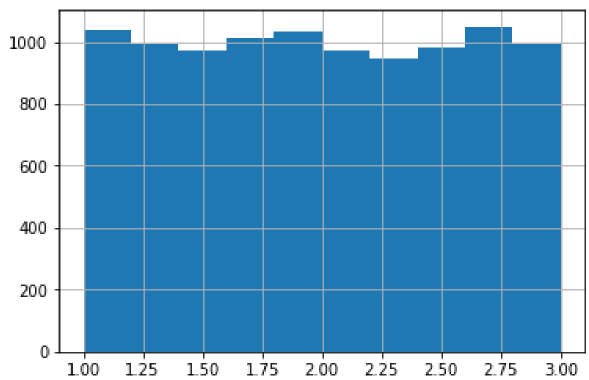
```
[9]: # Empirical CDF
     demand.hist(bins=20,density=True,cumulative=True)
```



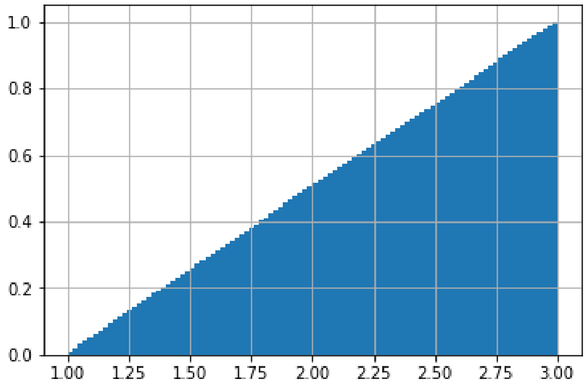
Q1. Sampling from Common Distributions

a) Generate 10000 samples of a uniform distribution between 1 and 3 and plot the histogram, as well as the empirical CDF. Calculate the mean and standard deviation of the samples, as well as the proportion between 2 and 2.5 (inclusive).

```
[10]:
```



```
[11]:
```



[12]:

```
1.9973519125605361
```

[13]:

```
0.5797250400567447
```

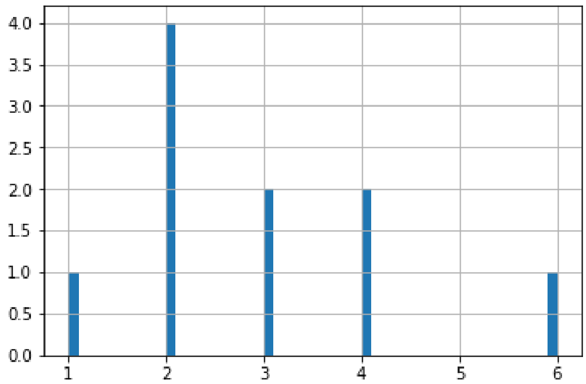
[14]:

```
0.2421
```

Q1-b: Generate 100 samples of a binomial distrubution with $n = 10$ and $p = 0.3$. Calculate the mean and standard deviation of the sample and compare with what it should be from the formula. Plot a histogram with 50 bins. Repeat with 10,000 samples.

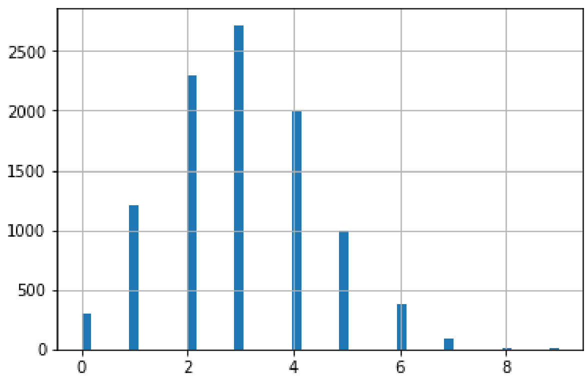
[15]:

```
Sample mean: 2.9
Sample std: 1.449137674618944
Theoretical mean: 3.0
Theoretical std: 1.4491376746189437
```



[16]:

```
Sample mean: 2.999
Sample std: 1.455819680143839
Theoretical mean: 3.0
Theoretical std: 1.4491376746189437
```



2. Generating Samples using np.random.choice

```
[17]: import numpy as np
      values=[1,2.5,3.5]
      probs=[0.3,0.5,0.2]
      np.random.choice(values,p=probs)
```

```
2.5
```

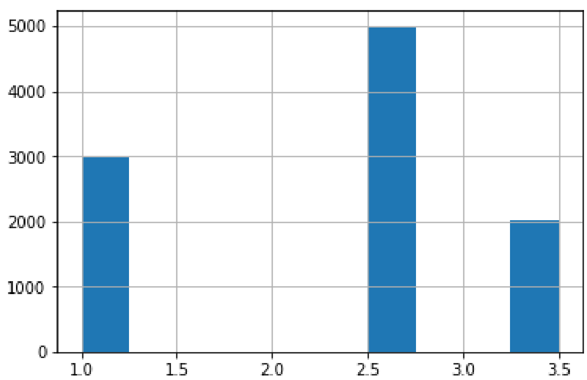
```
[18]: np.random.choice(values,p=probs,size=10)
```

```
array([3.5, 2.5, 1. , 3.5, 2.5, 1. , 2.5, 2.5, 3.5, 2.5])
```

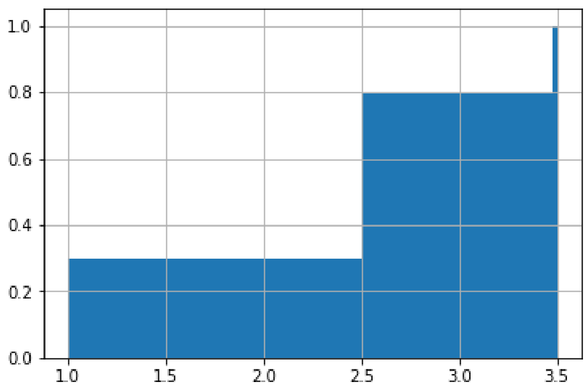
```
[19]: np.random.choice(['A','B','C'],p=[0.5,0.3,0.2],size=10)
```

```
array(['C', 'B', 'A', 'C', 'B', 'A', 'A', 'A', 'A', 'B'], dtype='<U1')
```

```
[20]: samples=pd.Series(np.random.choice(values,p=probs,size=10000))
      samples.hist()
```




```
[21]: samples.hist(bins=100,density=True,cumulative=True)
```



Q2. Sampling from Mixture of Distributions

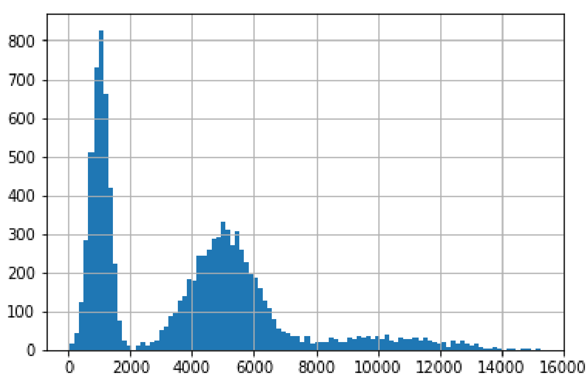
You are tasked with forecasting demand for a new product. Based on past data and your knowledge of the product, you estimate that the product quality will be Amazing with probability 0.1, Mediocre with probability 0.5, and Terrible with probability 0.4. You model the demand as normally distributed, with mean and standard deviation depending on the product quality as follows.

Prod. Quality:	Amazing	Mediocre	Terrible
μ	10000	5000	1000
σ	2000	1000	300

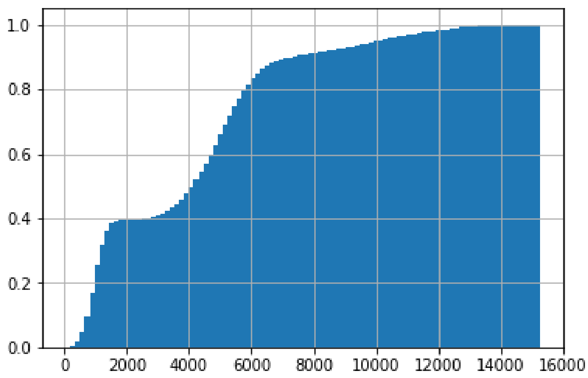
Create a `Series` called "forecast" with 10,000 samples of the demand forecast, and compute the mean and standard deviation of the samples, as well as the probability that demand is more than 6000. Finally, plot a histogram of the samples with 100 bins, as well as the empirical CDF.

[22]:

```
Sample mean: 3947.671852127712
Sample standard deviation: 2965.1276604886903
Probability demand more than 6000: 0.1798
```



[23]:



Q3. Sampling from More Complex Distributions

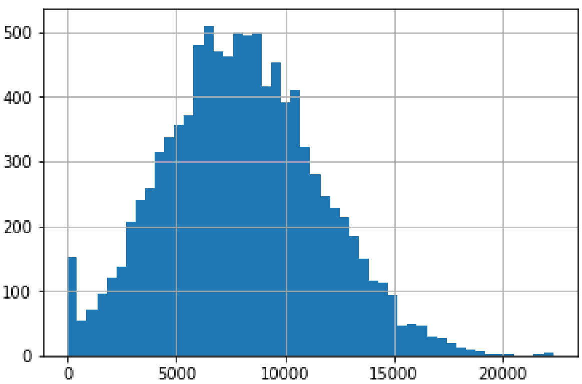
Nadeem is a car salesperson who faces the following incentive scheme at the dealership where he works. For each month, there is a "target profit" that the dealership sets for the month. If he makes more profit for the dealership that month than the target, then he receives a 20% bonus on the amount of profit over the target. However, if he does not meet the target, he receives no bonus. For example, if the target is 80,000 and he makes 100,000 of profit, then he receives a 4,000 bonus that month. However, if he makes 70,000, then he receives zero bonus that month. Nadeem would like to understand the distribution of his monthly bonus.

Nadeem estimates that the number of cars he sells is binomial distributed with $n = 200$ and $p = 0.2$. On every car he sells, the amount of profit he makes for the dealership is normally distributed with $\mu = 3000$ and $\sigma = 1000$, and the profit from each car is independent of another.

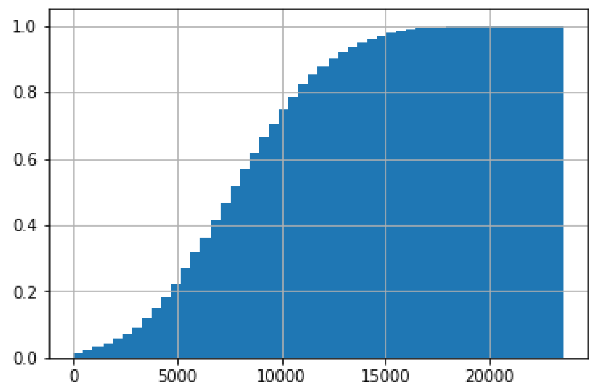
Create a `Series` called "monthlyBonus" with 10,000 samples of his monthly bonus. Compute the mean, the standard deviation, the probability the bonus is less than 5000, and plot a histogram with 50 bins as well as the empirical CDF.

[11]:

```
Mean is 8021.3351587025045
Standard deviation is 3583.7727955434984
Probability less than 5000 is 0.2058
```



[4]:



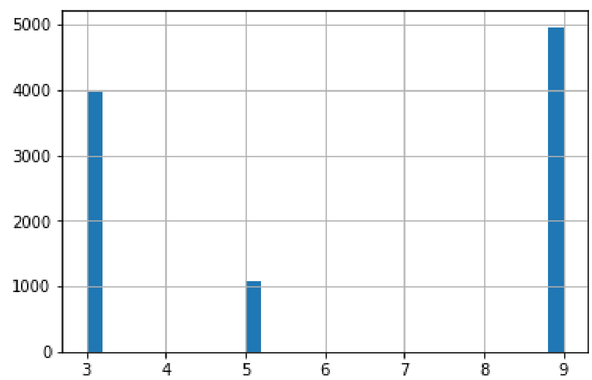
(Bonus) This question asks you to illustrate the Central Limit Theorem (CLT) by example. Consider the following distribution,

$$X = \begin{cases} 3 & \text{with probability } 0.4, \\ 5 & \text{with probability } 0.1, \\ 9 & \text{with probability } 0.5. \end{cases}$$

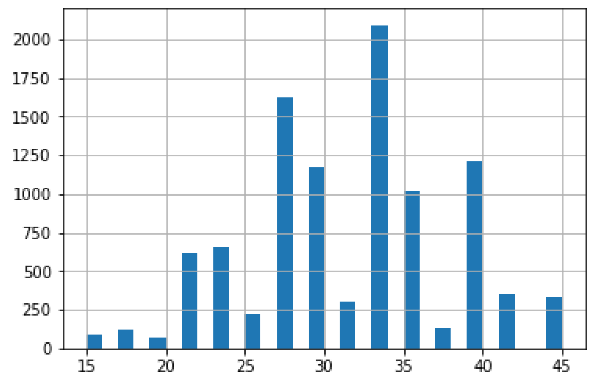
Define Y_n to be the sum of n independent random variables with the above distribution. Create 10000 samples of Y_1, Y_5, Y_{30}, Y_{100}, and Y_{1000} and plot their histograms (with 30 bins). You should be able to see the histograms converging to a Bell curve as n increases.

The Central Limit Theorem (CLT) says that this phenomenon always happens, regardless of the distribution of X. Moreover, it still takes place even if the n independent random variables do not have the same distribution, as long as each term in the sum is "small" relative to the whole. (For a precise mathematical formulation of the CLT in the case with non-identical random variables, search for Lyapunov or Lindeberg CLT on Wikipedia.)

[5]:

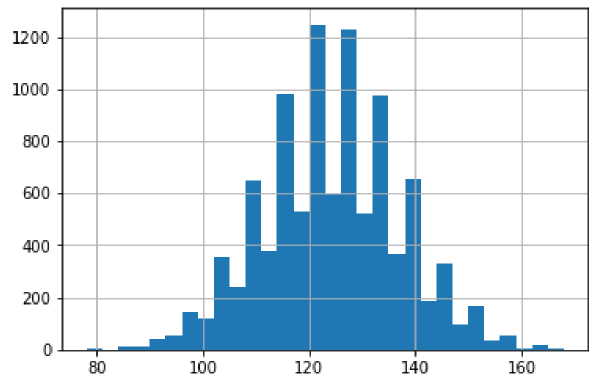


[6]:



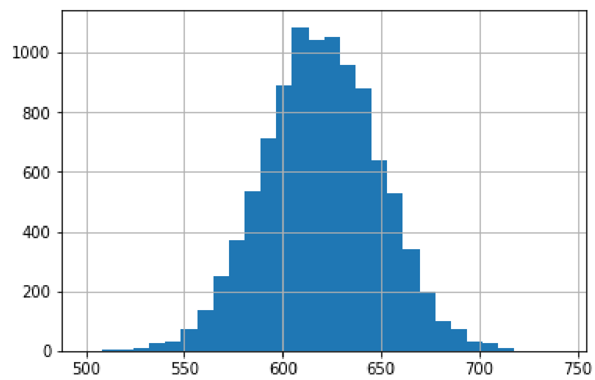
2000
1750
1500
1250
1000
750
500
250
0
15
20
25
30
35
40
45

[7]:



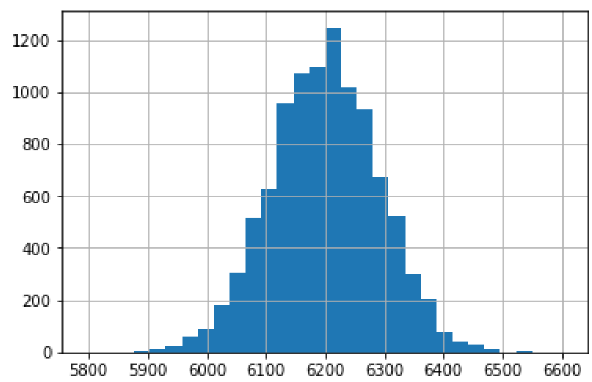
1200
1000
800
600
400
200
0
80
100
120
140
160

[8]:



1000
800
600
400
200
0
500
550
600
650
700
750

[10]:



1200
1000
800
600
400
200
0
5800
5900
6000
6100
6200
6300
6400
6500
6600