In [1]:
```
%pwd
```

Out[1]:
```
'C:\\Users\\PC\\Downloads\\GL_P'
```

Problems on Probability

1. A multinational farm conducted a job satisfaction survey on a sample of its employees. The table below summarizes the result of the survey. Assuming that the sample drawn is a random sample from its entire employee population, the HR would like to know the probability of the following events.

In [4]:
```
from IPython.display import Image
Image(filename='Inferential.jpg', width = 500, height = 500)
#mage(filename='Inferential', width = 500, height = 500)
```

Out[4]:

	Job Satisfaction				
Education Level	Satisfied	Neutral	Dissatisfied	Highly Dissatisfied	Total
Did not complete high school	10	20	30	40	100
High school graduate	20	30	25	50	125
Some college	30	60	35	25	150
College Graduate	120	40	30	10	200
Post-graduate	60	15	0	0	75
Total	240	165	120	125	650

What proportion of employees is satisfied?

In [5]:
```
Total_Satisfied = 240
Total_Emp = 650
Prob_Satisfied = round(Total_Satisfied/Total_Emp,4)*100
print('Proportion of employees satisfied. is %1.1f' % Prob_Satisfied + '%')
```

```
Proportion of employees satisfied. is 36.9%
```

What proportion of employees is at a risk of attrition? (Assume a dissatisfied employee will attrite eventually.)

In [15]:
```
Prob_dissatisfied=round(120/650,3)
print(Prob_dissatisfied)
prob_highly_dissatisfied=round(125/650,3)
print(prob_highly_dissatisfied)
Prob_atrisk=(Prob_dissatisfied+prob_highly_dissatisfied)*100
print('Proportion of employees at risk. is %1.1f' % Prob_atrisk + '%')
print(f"Proportion of employees at risk is {Prob_atrisk}'%'")
```

```
0.185
0.192
Proportion of employees at risk. is 37.7%
Proportion of employees at risk is 37.7'%'
```

In [10]:
```
Prob_atrisk
```

Out[10]:
```
37.7
```

If a dissatisfied employee attrite with probability 0.4 within a year, what is the probability of an employee to attrite in a year?

In [16]:
```
_of_attrite = 0.4
:alculated the prob of disatisfied employee in last question
_atrisk = .377

'r(An employee will attrite) = Pr(Employee dissatisfied AND Employee attrite) = Pr(Employee dissatisfied) * Pr(Employee attrite|Di
_attrite = (Prob_of_attrite * Prob_atrisk)*100
:('Probability of employees attrite. is %1.2f' % Prob_attrite +'%')
```

```
Probability of employees attrite. is 15.08%
```

If an employee is chosen randomly, what is the probability that he/she at least a high school graduate

In [17]:
```
r(At least High School) = Pr(High school) + Pr(Some college) + Pr(College Grad) + Pr(Post-Grad) = 1 - Pr(Did not complete school)
dnt_comp_high_school = 100
tal=650
ob_Atleast_high_school=(1-round(didnt_comp_high_school/Total,4))*100
int('Probability of an employee being at least a high school graduate is %1.1f' % Prob_Atleast_high_school +'%')
```

```
Probability of an employee being at least a high school graduate is 84.6%
```

What is the satisfaction rate among employees who are college graduate or more?

In [18]:
```
#will be denoted by Pr(Satisfied | College Grad or Post Grad)
satisfied_college_grad=120
satisfied_postgrad=60
Total_college_grad=200
Total_postgrad=75
Prob_satisfied=round((satisfied_college_grad+satisfied_postgrad)/(Total_college_grad+Total_postgrad),3)*100
print('Probability of satisfaction rate among employees who are college graduate or more . is %1.1f' % Prob_satisfied +'%')
```

```
Probability of satisfaction rate among employees who are college graduate or more . is 65.5%
```

Among the employees who have at least some college education, what proportion is not dissatisfied?

In [19]:
```
Not dissatisfied | at least some college)
not_disatisfied_some_college= 30+60+120+40+60+15
l_college=150+200+75
_not_dissatisfied_some_college=round(emp_not_disatisfied_some_college/Total_college,3)*100
t('Probability of not dissatisfied who have atleast some college education . is %1.1f' % Prob_not_dissatisfied_some_college +'%')
```

```
Probability of not dissatisfied who have atleast some college education . is 76.5%
```

Among the employees who are not satisfied, what proportion are college graduate or more?

In [20]:
```
#Pr(College Grad or Post-grad | Not satisfied)
emp_collegegrad_post_grad_not_satsified=40+30+10+15
Total_not_satisfied=650-240
Prob_Collge_ormore=round(emp_collegegrad_post_grad_not_satsified/Total_not_satisfied,3)*100
print('Probability of college or more . is %1.1f' % Prob_Collge_ormore +'%')
```

```
Probability of college or more . is 23.2%
```

Among the employees who have not had any college education, what proportion are not highly dissatisfied?

In [22]:
```
#Pr(Not highly dissatisfied | No college education)
emp_no_college=100+125
emp_not_highly_dissatisfied_nocollege=225-90

Prob_Not_higlydistasified=round(emp_not_highly_dissatisfied_nocollege/emp_no_college,3)*100
print('Probability of college or more . is %1.1f' % Prob_Not_higlydistasified +'%')
```

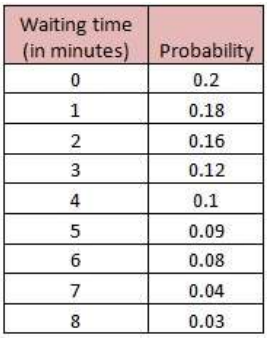
```
Probability of college or more . is 60.0%
```

Problem 2 on Probability

A multinational bank is concerned about the waiting time of its customers before they use the ATM for their transactions. A study of a random sample of 500 customers reveals the following probability distribution. The below table shows the distribution.

In [23]:
```
from IPython.display import Image
Image(filename='Problem2.jpg',width=200,height=100)
```

Out[23]:



Waiting time (in minutes)	Probability
0	0.2
1	0.18
2	0.16
3	0.12
4	0.1
5	0.09
6	0.08
7	0.04
8	0.03

What is the probability that a customer will wait for more than 5 minutes?

In [24]:
```
#P(x>5)= P(6)+P(7)+p(8)
Prob_greater_than5=0.08 +0.04+0.03
print('Probability of waiting time graeter than 5 is %1.2f' % Prob_greater_than5 )
```

```
Probability of waiting time graeter than 5 is 0.15
```

What proportion of the customers do not have to wait at all?

In [25]:
```
#Pr(Customers do not wait) = Pr(X = 0) i.e. we will take the prob at 0 i.e. 0.2
print('Probability of customers do not wait. is %1.2f' % 0.2 )
```

```
Probability of customers do not wait. is 0.20
```

What is the expected waiting time for a customer?

In [26]:
```
#Expected waiting time will be summation of product of waiting time and its respective probability

Prob_expected_time=0* 0.2 + 1* 0.18 + 2*0.16 + 3* 0.12 +4 * 0.1 + 5 * 0.09 +6* 0.08 +7*0.04  +8* 0.03
print('Probability of expected waiting time for customers is %1.2f' % Prob_expected_time )
```

```
Probability of expected waiting time for customers is 2.71
```

Problems on Binomial and Normal Distributions

In an inspection of automobiles 60% of all automobiles had emissions that do not meet the pollution requirement. For a random sample of 15 automobiles, compute the following

Probability that all 15 fails the inspection

Exactly 8 fails the inspection

Seven or less passes the inspection

Expected number of automobiles to pass the inspection and its variance.

In [27]:
```
import numpy as np
import scipy.stats as stats
```

In [28]:
```
# Probability that all 15 fails the inspection

p=0.6
n=15
k=15
binomial = stats.binom.pmf(k,n,p)
print('Probability that all 15 fails the inspection is %1.5f' % binomial )
```

```
Probability that all 15 fails the inspection is 0.00047
```

In [29]:
```
# Probability Exactly 8 fails the inspection

p=0.6
n=15
k=8
binomial = stats.binom.pmf(k,n,p)
print('Probability Exactly 8 fails the inspection is %1.5f' % binomial )
```

```
Probability Exactly 8 fails the inspection is 0.17708
```

In [30]:
```
# Probability Seven or less passes the inspection

p=0.6
n=15
k=7
##To  answer this we need to calculate cumulative probability
binomial = stats.binom.cdf(k,n,p)
#since we have used the p has passing prob, to calculate not passing we just need tosubtract it frim 1
binomial_notPassing=1-binomial

print('Probability Seven or less passes the inspection is %1.5f' % binomial_notPassing )
```

```
Probability Seven or less passes the inspection is 0.78690
```

In [31]:
```
#Alternatively, probability that an automobile will pass pollution test is 0.4.
# Probability Seven or less passes the inspection , taking q=0.4, k=7,n=15

print('Probability Seven or less passes the inspection is %1.5f' % stats.binom.cdf(7,15,0.4))
```

```
Probability Seven or less passes the inspection is 0.78690
```

In [32]:
```
#Expected number of automobiles to pass the inspection and its variance.
#E(X) = np
p=0.6
n=15
probability_of_passing_inspection=0.4
Expected_automobiles= n*probability_of_passing_inspection
print('Expected number of automobiles is %1.5f' % Expected_automobiles )
Var_auto = n*p*(1- p)
print('variance is %1.5f' % Var_auto )
```

```
Expected number of automobiles is 6.00000
variance is 3.60000
```

Over a long period of time in a large multinomial corporation, 10% of all sales trainees are rated as outstanding, 75% are rated as excellent, 10% are rated as satisfactory and 5% are considered unsatisfactory. Find the following probabilities for a sample of 10 trainees selected at random

Two are rated as outstanding

Two or more are rated as outstanding

Eight of the ten are rated either outstanding or excellent

None of the trainees are rated as unsatisfactory

This is a binomial problem.

In [33]:
```
#Two are rated as outstanding

p=0.1
n=10
k=2
binomial = stats.binom.pmf(k,n,p)
print('Probability of two are rated as outstanding is %1.5f' % binomial )
```

```
Probability of two are rated as outstanding is 0.19371
```

In [34]:
```
#Two or more are rated as outstanding.

#For this we will use cummulative probability
p=0.1
n=10
k=1
##To  answer this we need to calculate cumulative probability
binomial = stats.binom.cdf(k,n,p)
#since we have calculated for 1 or less, for two or more we will subtract this prob from 1
print('Probability of two or more are rated as outstanding is %1.5f' % (1-binomial))
```

```
Probability of two or more are rated as outstanding is 0.26390
```

In [35]:
```
#Eight of the ten are rated either outstanding or excellent
#prob of excellent & outstanding is 75+10
p=0.85
n=10
k=8
binomial = stats.binom.pmf(k,n,p)
print('Probability of eight out of ten is excellent & outstanding is %1.5f' % binomial )
```

```
Probability of eight out of ten is excellent & outstanding is 0.27590
```

In [36]:
```
#None of the trainees are rated as unsatisfactory\#prob of unsatisfactory is 5
p=0.05
n=10
k=0
binomial = stats.binom.pmf(k,n,p)
print('Probability of no trainees are unsatisfactory is %1.5f' % binomial )
```

```
Probability of no trainees are unsatisfactory is 0.59874
```

3. A labor union's examining board for the selection of trainees has a record of admitting 70% of all applicants who meet a basic set of criteria. Five members of a minority group recently came before the board and four out of five were rejected. Find the probability that one or fewer would be accepted if the record is really 0.7. Did the board apply a lower probability of acceptance when reviewing the five members of the minority group?

In [37]:
```
p=0.7
n=5
k=1
binomial = stats.binom.cdf(k,n,p)
print('Probability that 1 or fewer were admitted is %1.5f' % binomial )
```

```
Probability that 1 or fewer were admitted is 0.03078
```

Since this probability is very small, though not impossible, this event is not likely to occur. Hence our conclusion that in this instance, probability of acceptance does not seem to be 70%, but lower.

4. According to the Telecommunication Industry the average monthly cell phone bill is Rs. 850 with a standard deviation of Rs. 150.

What is the probability that a randomly selected cell phone bill is more than Rs 1200?

What is the probability that a randomly selected cell phone bill is between Rs 750 and Rs 1200?

What is the probability that a randomly selected cell phone bill is no more than Rs 650?

What is the amount above which lies top 15% of cell phone bills?

What is the amount below which lies bottom 25% of cell phone bills?

In [39]:
```
#Whatis the probability that a randomly selected cell phone bill is more than Rs 1200?

z = (1200 - 850)/150
```

In [40]:
```
z
```

Out[40]:
```
2.3333333333333335
```

In [45]:
```
1-stats.norm.cdf(2.3333333333333335)# normal-cdf(cumulative density function or continous area under the curve)
```

Out[45]:
```
0.009815328628645315
```

In [46]:
```
0.009815328628645315 * 100
```

Out[46]:
```
0.9815328628645315
```

the probability that a randomly selected cell phone bill is more than Rs 1200 is 0.9815328628645315 %

In [47]:
```
# What is the probability that a randomly selected cell phone bill is between Rs 750 and Rs 1200?

z1 = (1200 - 850)/150
z2 = (750 - 850)/150
```

In [49]:
```
z1
```

Out[49]:
```
2.3333333333333335
```

In [50]:
```
z2
```

Out[50]:
```
-0.6666666666666666
```

In [51]:
```
stats.norm.cdf(z1)-stats.norm.cdf(z2)
```

Out[51]:
```
0.7376921338244318
```

In [52]:
```
0.7376921338244318 * 100
```

Out[52]:
```
73.76921338244318
```

73.77 % is the probability that a randomly selected cell phone bill is between Rs 750 and Rs 1200.

In [56]:
```
#What is the probability that a randomly selected cell phone bill is no more than Rs 650?

z =  (650-850)/150
```

In [57]:
```
stats.norm.cdf(z)* 100
```

Out[57]:
```
9.121121972586788
```

9.12 % is the probability that a randomly selected cell phone bill is no more than Rs 650

In [58]:
```
#What is the amount above which lies top 15% of cell phone bills?

#Let the amount be M. Pr(X ≥ M) = 15% => 1 - Pr(X < M) = 0.85 . To calculate this we will use the percent point function i.e ppf

#z = (0.85-850)/150
```

In [60]:
```
stats.norm.ppf(0.85,loc=850,scale=150)
```

Out[60]:
```
1005.4650084240685
```

In [61]:
```
#What is the amount below which lies bottom 25% of cell phone bills?

stats.norm.ppf(0.25,loc=850,scale=150)
```

Out[61]:
```
748.8265374705877
```

5. For on-campus recruitment Ms. Z has sat for tests by Company A and Company B. For both tests her score is 50. It is known that for Company A, scores have a normal distribution with mean 40 and standard deviation 15 whereas for Company B, scores have a normal distribution with mean 45 and standard deviation 10. Relatively speaking in which test has Ms. Z done better?

In [62]:
```
#Z-score from Company A:
Z1 = (50-40)/15

#Z-score from Company B:
Z2 = (50-45)/10

print("Z1:",Z1)
print("Z2:",Z2)
```

```
Z1: 0.6666666666666666
Z2: 0.5
```

A higher score on Z scale is indicative of a better performance. Hence Ms. Z has done better in the test for Company A

Problem on Bayes Theorem

A computer component is given scores (A, B, C) after production.

On an average, 70% components were given a score of A, 18% were given a score of B and 12% a score of C.

It was found that 2% of the components that were given a score of A, 10% that were given a score of B and 18% that were given a score of C, eventually failed.

If you randomly pickup a failed component, what is the probability that it had received a score of B?

Solution:

The probablity will be given by the below formula:

It can be written as:

P(B | F) = P(B ∩ F)* P(B) / P(F)

and the formula can be break down into:

P(B | F) = P(B) * P(F | B) / (P(A) * P(F | A) + P(B) * P(F | B) + P(C) * P(F | C))

In [63]:
```
Prob_of_B=0.18
Prob_of_Fail_given_B=0.10
Prob_of_A=0.70
Prob_of_Fail_given_A=0.02
Prob_of_C=0.12
Prob_of_Fail_given_C=0.18

Prob_of_Fail=(0.02*0.70 + 0.18*0.10 + 0.12 * 0.18)
Prob_of_B_given_Fail= (Prob_of_B*Prob_of_Fail_given_B)/Prob_of_Fail

print('Probability that a failed component will score B is %1.5f' % Prob_of_B_given_Fail)
```

```
Probability that a failed component will score B is 0.33582
```

Problems on Poisson Distribution

A bank is interested in studying the number of people who use the ATM located outside during night hours. On an average, 1.6 customers walk up to the ATM during a 10-minute interval, b/w 9pm and midnight.

Find the probability of exactly 3 customers using the ATM in a 10-minute interval.

In [64]:
```
n = 3
rate = 1.6
poisson = stats.poisson.pmf(n,rate)* 100
print('Probability of exactly 3 customers using the ATM in a 10-minute interval is %1.2f' % poisson )
```

```
Probability of exactly 3 customers using the ATM in a 10-minute interval is 13.78
```

In [65]:
```
### What is the probability of 3 or fewer people?
```

In [66]:
```
n = np.arange(0,4)
rate = 1.6
poisson = stats.poisson.pmf(n,rate) * 100

##To calculatee the probability of 3 or fewer we will add the prob of 0,1,2 and 3 people

poisson[0]+poisson[1]+poisson[2]+poisson[3]
```

Out[66]:
```
92.11865127702812
```

In [ ]:
```

```