Assignment 2

Christina Bisol

20046973

Q1 Poisson probability function as limit of the binomial probability

Defining Variables and set up:

$$n = N * p$$

where n=rate of success, N=number of trials, p=probability of success

$$p = \frac{n}{N}$$

solving for p

$$P_{binomial} = \frac{N!}{k!(N-k)!}p^k(1-p)^{N-k}$$

Now substituting p in and taking the limit as n goes to infinity:

$$\lim_{N\to\infty}(\frac{N!}{k!(N-k)!})(\frac{n}{N})^k(1-\frac{n}{N})^{N-k}$$

Removing the constants

$$n^k$$

and

$$\frac{1}{k!}$$

And using power laws on

$$(1-\frac{n}{N})^{N-k}$$

Yields:

$$(\frac{n^k}{k!})\lim_{n\to\infty}(\frac{N!}{k!(N-k)!})(\frac{1}{n^k})(1-\frac{n}{N})^N(1-\frac{n}{N})^{-k}$$

Taking the limit of each term

1st 2 terms

$$\lim_{N\to\infty}(\frac{N!}{k!(N-k)!})(\frac{1}{n^k})$$

Expanding the factorials:

$$\lim_{N\to\infty}\frac{N(N-1)(N-2)\ldots(N-k)(N-k-1)\ldots(1)}{(N-k)(N-k-1)\ldots(1)}(\frac{1}{N^k})$$

simplifying:

$$\lim_{N\to\infty}\frac{N(N-1)(N-2)\ldots(N-k+1)}{N^k}$$

I cancelled out n-k terms, we have k terms left so this can be rewritten as

$$\lim_{N\to\infty}(\frac{N}{N})(\frac{N-1}{N})\ldots(\frac{N-k+1}{N})=1$$

b/c each term approaches 1 as N approaches infinity

2nd term

$$\lim_{N\to\infty}(1-\frac{n}{N})^N$$

Using defintion of e:

$$e=\lim_{N\to\infty}(1+\frac{1}{x})^x$$

we can define x so we can use this limit:

$$x=-\frac{N}{n}$$

Substituting it in and solving:

$$\lim_{N\to\infty}(1+\frac{1}{x})^{x(-n)}=e^{-n}$$

Last term

$$\lim_{N\to\infty}(1-\frac{n}{N})^{-k}$$

but we know that

$$\lim_{N\to\infty}\frac{n}{N}=0$$

which leaves us with

$$\lim_{N\to\infty}(1)^{-k}=1^{-k}$$

Combining these limits:

$$(\frac{n^k}{k!})\lim_{n\to\infty}(\frac{N!}{k!(N-k)!})(\frac{1}{n^k})(1-\frac{n}{N})^N(1-\frac{n}{N})^{-k}=(\frac{n^k}{k!})*(1)*(e^{-n})$$

Which can be rewritten as:

$$P(n,k) = (\frac{n^k e^{-n}}{k!})$$

This is the Poisson distribution

I used the following website for assistance

[https://medium.com/@andrew.chamberlain/deriving-the-poisson-distribution-from-the-binomial-distribution-840cc1668239 (https://medium.com/@andrew.chamberlain/deriving-the-poisson-distribution-from-the-binomial-distribution-840cc1668239)](https://medium.com/@andrew.chamberlain/deriving-the-poisson-distribution-from-the-binomial-distribution-840cc1668239)

In [1]:
```
import numpy as np
import scipy.stats as stats #for binomial distribution
```

In [2]:
```
#Q2a)
#use binomail distribution b/c discrete number of throws
p_dice=1/6 #probability of rolling a 5
throws=15 #number of throws
ka=3 #want less than 4 times and cdf is less than and equal to
Pa=stats.binom.cdf(ka,throws,p_dice)*100 #multiply by 100 for percentage
print("Probability of rolling a 5 less than 4 times in 15 throws is {:.2f}%".f
ormat(Pa))
```

```
Probability of rolling a 5 less than 4 times in 15 throws is 76.85%
```

In [3]:
```
#Q2b)
avg=15 #expected average car accidents per week in Kingston
#making the average daily by dividing by N=7 (days per week)
avgN=15/7 #average car accidents per day in kingston
k_Q2b=np.array([1,2,3]) #array with desired number of car accidents per day
#not discrete so will use poisson (also dont have a probability p for binomia
l)
prbQ2b=stats.poisson.pmf(k_Q2b,avgN)*100 #probability as a percentage
print("Probability of 1 to 3 (inclusive) car accidents occuring in Kingston on
 a given day is {:.2f}%".format(prbQ2b.sum()))
```

```
Probability of 1 to 3 (inclusive) car accidents occuring in Kingston on a giv
en day is 71.32%
```

In [4]:
```
#Q2c)
#normal distribution
mu=90 #min average vaiting time
sigma=25 #min standard deviation of the waiting time
#want probability of waiting for more than 2 hours= 120 min

#so integrative from 121 to positive infiinity (i.e. x>120)
pQc=stats.norm.sf(120,mu,sigma) #following code in example 7.3 of the txtbk
#sf is greater than, so does not include 120 min
print("Probability of a waiting time greater than 2h is {:.2f}%".format(100*pQ
c))
```

```
Probability of a waiting time greater than 2h is 11.51%
```

In [5]:
```
#Q2d) Binomial distribution bc discrete number of diseases
k_partd=[1,2,3,4,5,6,7,8,9,10] #array to acccount for each of the 10 diseasese
disease=10 #number of rare diseases (N)
pd=0.05 # 5% probability of false positivity
#We want the cumulative probability
prob2d = stats.binom.pmf(k_partd,disease,pd)
prob2d = prob2d*100 #making it to percentage
prob2d = sum(prob2d)
print("The probability that Dr. Nitran tests positive for 1 or more diseases i
s {:.1f} %".format(prob2d))
```

```
The probability that Dr. Nitran tests positive for 1 or more diseases is 40.1
%
```

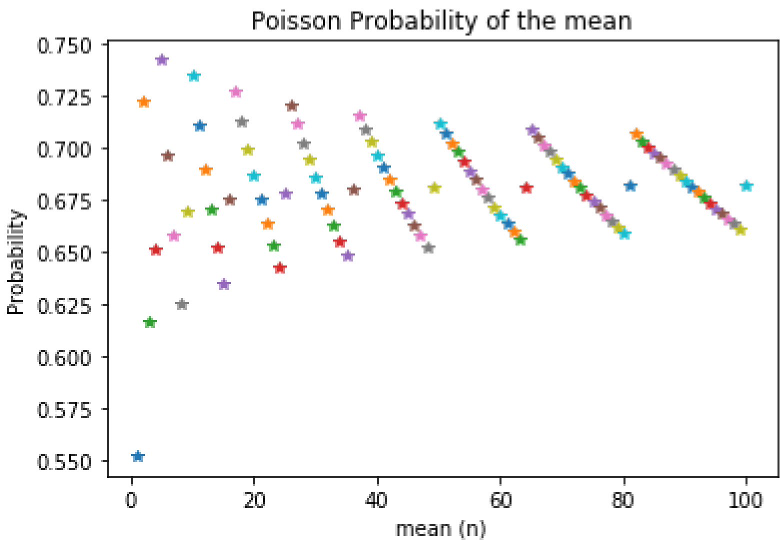
In [7]:
```
#Q2e)
#Assuming normal distribution for the heights
#so given that 60% of the data lies within the range 158 & 178 we can see:
mean=(158+178)/2
#done bc normal so data is symmetric about the mean and evenly distributed

#We can assume 80% of the population has a heigh of 178cm or less
#Using a z-score table, 178cm is 0.84 standard deviations away from the mean
#this is 10cm above the mean as 0.84(std)=10
std=10/0.84 #solving for the standard deviation in cm

#now solving for the problem:

pQ2e=stats.norm.sf(194,mean,std)
#normal distibution, using sf because want height 195 and above
#sf only determines greater than, so I subtracted one
ans2e=pQ2e*100 #multiply by 100 to get percentage

print("Probability of someone being 195cm or taller is {:.2f}%".format(ans2e))
```

```
Probability of someone being 195cm or taller is 1.45%
```

Sample z-score table includes:

[https://statistics.laerd.com/statistical-guides/img/normal-table-large.png (https://statistics.laerd.com/statistical-guides/img/normal-table-large.png)](https://statistics.laerd.com/statistical-guides/img/normal-table-large.png)

In [8]:
```
#Q2f)
import matplotlib.pyplot as mp
import pylab as pl #will use to plot
#experimentally showing
#running a loop with all values of the mean in the range (1-100)
for n in np.arange(1,101):
    inf=n-np.sqrt(n) #lower bound of n so will use cdf
    sup=n+np.sqrt(n) #upper bound of n so will use sf
    pQ2f=1-stats.poisson.cdf(inf,n)-stats.poisson.sf(sup,n)
    #plot
    mp.plot(n,pQ2f,'*')
    mp.xlabel("mean (n)")
    mp.ylabel("Probability")
    mp.title("Poisson Probability of the mean")
    mp.grid()
mp.show()
```



In [9]:
```
#Q3a)
pq3=0 #the probability of an event
i=0 #counter/number of events
#need i to be float like in a range bc it can be a decimal as specified in the
 question
while (pq3<=0.005):#while the probability is less than 0.5%
#b/c 100%-99.5%=0.5%
    i+=0.001 #increase the number of events by decimal
    pq3= stats.poisson.sf(7,i) #determine the poisson prob when k is 8 or more
and N=i
    #sf if for greater than which means it will start at 8 as desired
print("The maximum number of expected background events is {:.1f}".format(i))
```

```
The maximum number of expected background events is 2.6
```

In [10]:
```
#Q3b)
#using survival factor to find significance
s=stats.poisson.sf(7,4.6) #poisson probability for k>=8, N=4.6
ans=100*(1-s) #turning the significance into a percentage
print("The significance of finding if there were 4.6 background events is {:.1
f}%".format(ans))
```

```
The significance of finding if there were 4.6 background events is 90.5%
```

In [11]:
```
#Q3c)
B=0.1 #average background rate of 0.1 events per year
#this is mean so can use poisson
sr=0 #sr is the signal rate in events per year (lambda)
#set to be as low as possible
t=2 #experiment runs for 2 years
e=1 #and detected 1 event

p3c=stats.poisson.cdf(e,B*t)

#need the standard deviation of this mean based on this data
#determine sr they can reject w 90% confidnece ...aka largest sr given backgro
und rate that resuts in 90% probabiliy of 1 event or less being observed
j=0 #number of background events/counter (also needs to be float)
while (p3c>=0.9): #while the probability is less than 90%
    sr+=0.0001
    p3c=stats.poisson.cdf(e,(B+sr)*t)
sr-=0.0001
print("They can reject a value of {:.2f} at a 90% confidence".format(sr))
```

```
They can reject a value of 0.17 at a 90% confidence
```