Homework 2

Problem 1:

The problem is to find a distribution which can describe the number of car accidents in a given hour of the day.

We know that "Poisson Distribution" is created to expresses the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant rate and independently of the time since the last event.

Consider our problem and we can find that it could be described by Poisson Distribution. So we can use it to solve our problem.

$$P(x = k) = e^{-\lambda} * \frac{\lambda^k}{k!}$$

This is the distribution of Poisson Distribution. We can find here is to parameters λ and k. λ is the average which we want to solve out.

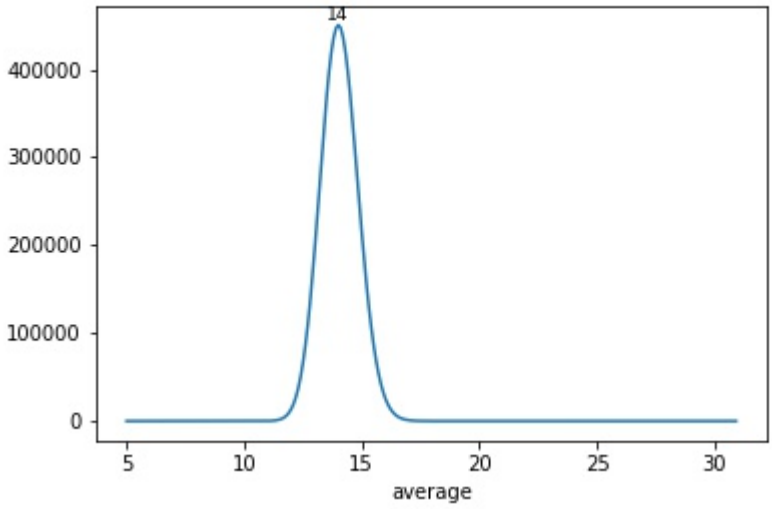
Look at those data and I guess λ is between 5 and 30 and with each λ, we can calculate $P(x = k)$.

```
def poisson(k,lam):
    return ((lam**k*np.exp(-
lam)/np.math.factorial(k)))
lam_span = np.arange(5,31,0.1)
data =
np.array([16,24,16,12,16,11,14,15,9,14,7])
prob = np.zeros((len(data),len(lam_span)))
for i in np.arange(0,len(lam_span)):
    for j in np.arange(0,len(data)):
        prob[j][i] =
poisson(data[j],lam_span[i])
```

Then multiply all $P(x = k)$ under the same λ.

```
mul = np.ones((len(lam_span)))
for i in np.arange(0,len(lam_span)):
    for j in np.arange(0,len(data)):
        mul[i] *= prob[j][i]
```

Then plot the image of mul by λ

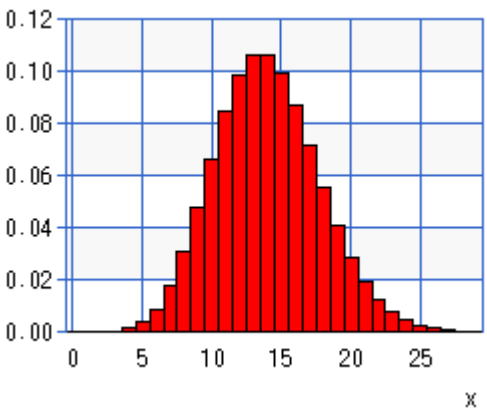


We can find the peak is $\lambda = 14$

So I conclude that the average accident rate is 14.

And the distribution is :

$$P(x = k) = e^{-14} * \frac{14^k}{k!}$$



Problem2:

```
import numpy as np
import matplotlib.pyplot as plt

def calPar(miu,sig):

    def N(x,miu,sig):
        return
1/np.math.sqrt(2*np.pi*sig**2)*np.exp(-(x-
miu)**2/2/sig**2)

    miu_span =
np.linspace(0.1,miu.max(),1000)
    mul = np.ones((len(miu_span)))
    prob =
np.zeros((len(miu),len(miu_span)))
    for i,m in enumerate(miu):
        for j,n in enumerate(miu_span):
            prob[i][j] =
N(miu[i],miu_span[j],sig[i])
    for i,m in enumerate(miu_span):
        for j,n in enumerate(miu):
            mul[i] *= prob[j][i]
    miu_ave =
miu_span[np.where(mul==np.max(mul))]
```

```
    sig_span =
np.linspace(0.1,10*miu.var(),1000)
    mul2 = np.ones((len(sig_span)))
    prob2 =
np.zeros((len(sig),len(sig_span)))
    for i,m in enumerate(sig):
        for j,n in enumerate(sig_span):
            prob2[i][j] =
N(miu[i],miu_ave,sig_span[j])
    for i,m in enumerate(sig_span):
        for j,n in enumerate(sig):
            mul2[i] *= prob2[j][i]
    sig_ave =
sig_span[np.where(mul2==np.max(mul2))]
    return (miu_ave,sig_ave)
```