# cheg304 homework4 question1

AUTHOR
udid 702687390

PUBLISHED
February 28, 2025

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.special import factorial

df = pd.read_csv('question1_data.txt', sep=',', index_col=1)
df
```

| | Number of Employees |
|---|---|
| **Number of Accidents** | |
| 0 | 447 |
| 1 | 132 |
| 2 | 42 |
| 3 | 21 |
| 4 | 5 |

computing the mean and variance

```
mean = np.dot(df.index, df['Number of Employees']) / np.sum(df['Number of Employees'])
print(f'mean: {mean:.3f} incidents')

variance = np.dot((df.index - mean)**2, df['Number of Employees']) / (np.sum(df['Number o
print(f'variance: {variance:.3f} incidents^2')
```

```
mean: 0.462 incidents
variance: 0.667 incidents^2
```

for the poisson, $\lambda$ should be the EV (mean) of the distribution

for the negative binomial we know the mean is

$$\mu = \frac{k(1-p)}{p}$$

and the variance is

$$\sigma^2 = \frac{k(1-p)}{p^2}$$

and now, since we calcualted the mean and the variance, we have 2 equations with 2 unknowns. this yields

$$k = \frac{\mu^2}{\sigma^2 - \mu} \qquad p = \frac{\mu}{\sigma^2}$$

```
k = (mean**2)/(variance - mean)
print(f'k: {k:.3f}')

p = (mean/variance)
print(f'p: {p:.3f}')

k = np.round(k)
print(f'k (rounded to int): {k:.0f}')
```

```
k: 1.043
p: 0.693
k (rounded to int): 1
```

and now computing the respective $\chi^2$

```
def poisson(x, lambd):
    return (lambd**x) * np.exp(-lambd) / factorial(x)

def choose(top: int, bottom: int):
    return factorial(top) / (factorial(bottom) * factorial(top-bottom))

def negative_binom(x, k, p):
    return choose(x+k-1, k-1) * (p**k) * ((1-p)**x)

n = np.sum(df['Number of Employees'])

poisson_predictions = poisson(np.array(df.index), mean) * n
print('poisson predictions\n',np.round(poisson_predictions,0))

nb_predictions = negative_binom(np.array(df.index), k, p) * n
print('negative binom predictions\n',np.round(nb_predictions,0))

print('\n ----- chi squared stuffs ----- \n')
chi_poisson = np.sum((df['Number of Employees'] - poisson_predictions)**2 / poisson_predi
print(f'chi squared for poisson: {chi_poisson:.2f}')

chi_nb = np.sum((df['Number of Employees'] - nb_predictions)**2 / nb_predictions)
print(f'chi squared for negative binomial: {chi_nb:.2f}')
```

```
poisson predictions
 [408. 188.  44.   7.   1.]
negative binom predictions
 [448. 138.  42.  13.   4.]

 ----- chi squared stuffs -----

chi squared for poisson: 74.26
chi squared for negative binomial: 5.45
```

the negative binomial has a much smaller $\chi^2$ value. this lines up well with the theory since our poisson distribution has one value for mean and variance, but the mean and the variance of the historical data were not the same, so it makes sense that a distribution with 2 parameters describing it captures this extra information.

```
# this is filler text
```