Exercise 9.1 (May 29, 2020)

-B9TB1707

Question:

- A, B, C and D are the last 4 digits from your student number (see Excercise 4.1)
- In an area of a country, it is known that earthquakes occur 0.7*(A+1) times in B+1 days in an average sense since the dawn of the history
- However, there were 10+C+D earthquakes in the last four weeks
- Calculate the probability of 10+C+D and more earthquakes occurring in four consecutive weeks

Solution:

My code for the solution is as follows:

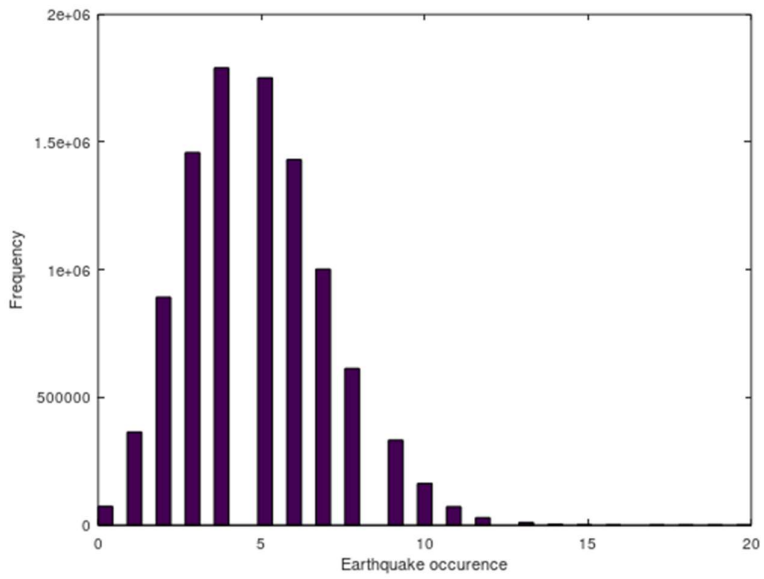
CAPS_09_B9TB1707_9.1.m

```
A = 1,B = 7,C = 0,D = 7;
l = .7*(A+1)*28/(B+1);
t = 10+C+D;
occurence=randp(l,10000000,1);
hist(occurence,45);
xlabel('Earthquake occurence');
ylabel('Frequency');
n=occurence>=t;
p=sum(n)/length(n);
printf("The probability is earthquakes occuring %d times or more in 28 days is %d",t,p);
```

The output is as follows:

```
Command Window
>> The probability is earthquakes occuring 17 times or more in 28 days is 1.4e-05
```

Z+ Z- Insert Text Axes Grid Autoscale



Where A = 1, B = 7, C = 0, D = 7.
The probability that the event happened k times in this specified interval is given by;

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

And the expected value of the event defined $E[X]$ is equal to λ.
This is the Poisson distribution equation, and I will use this equation to solve this problem.

How it works:

1. Line 1 declares the variables A, B, C, D.
2. Line 2 declares the variable to hold the value of lambda for the Poisson equation.
3. Line 3 declares the variable to hold the unusual occurrences of earthquakes, the threshold value.
4. A Poisson distribution is generated by randp function of length 10000000
5. Line 5 generates a histogram based on the Poisson distribution.
6. Line 6 and 7 labels the histogram.
7. Line 8 filters the data in the distribution and keeps only the values equal to or greater than the threshold, by using a simple relational operatorLin.
8. Line 9 calculates the probability
9. Line 10 displays the results.

Conclusion:

The program above uses the randp() function to generate the passion distribution. One could use an iterative approach to solve this problem, but I feel that this way is too inefficient and cumbersome to code, so I stuck with the randp() function.