**Poisson Distribution**

The **Poisson distribution**is the probability distribution of independent event occurrences in an interval. If *λ*is the [mean](http://www.r-tutor.com/elementary-statistics/numerical-measures/mean) occurrence per interval, then the probability of having *x*occurrences within a given interval is:

       x -λ
f(x) = λ-e-- where x = 0,1,2,3,...
        x!


**Problem**

If there are twelve cars crossing a bridge per minute on average, find the probability of having seventeen or more cars crossing the bridge in a particular minute.

**Solution**

The probability of having *sixteen or less*cars crossing the bridge in a particular minute is given by the function ppois.

> ppois(16, lambda=12)   # lower tail   
[1] 0.89871

Hence the probability of having seventeen or more cars crossing the bridge in a minute is in the *upper tail*of the probability density function.

> ppois(16, lambda=12, lower=FALSE)   # upper tail   
[1] 0.10129

from scipy.stats import poisson

import seaborn as sb

data\_binom = poisson.rvs(mu=4, size=10000)

ax = sb.distplot(data\_binom,

kde=True,

color='green',

hist\_kws={"linewidth": 25,'alpha':1})

ax.set(xlabel='Poisson', ylabel='Frequency')

from collections import Counter

filename = 'myfile.txt'

with open(filename) as fn:

# Read each line

ln = fn.readline()

# Keep count of lines

lncnt = 1

while ln:

print("Line {}: {}".format(lncnt, ln.strip()))

ln = fn.readline()

lncnt += 1

with open(r'myfile.txt') as f:

p = Counter(f.read().split())

print(p)

import nltk

word\_data = "It originated from the idea that there are readers who prefer learning new skills from the comforts of their drawing rooms"

nltk\_tokens = nltk.word\_tokenize(word\_data)

print (nltk\_tokens)

import nltk

sentence\_data = "Sun rises in the east. Sun sets in the west."

nltk\_tokens = nltk.sent\_tokenize(sentence\_data)

print (nltk\_tokens)

Stemming Algorithm

In the areas of Natural Language Processing we come across situation where two or more words have a common root. For example, the three words - agreed, agreeing and agreeable have the same root word agree. A search involving any of these words should treat them as the same word which is the root word. So it becomes essential to link all the words into their root word. The NLTK library has methods to do this linking and give the output showing the root word.

The below program uses the Porter Stemming Algorithm for stemming.

import nltk

from nltk.stem.porter import PorterStemmer

porter\_stemmer = PorterStemmer()

word\_data = "It originated from the idea that there are readers who prefer learning new skills from the comforts of their drawing rooms"

# First Word tokenization

nltk\_tokens = nltk.word\_tokenize(word\_data)

#Next find the roots of the word

for w in nltk\_tokens:

print "Actual: %s Stem: %s" % (w,porter\_stemmer.stem(w))