Notebook

April 12, 2019

Local date & time is : 04/12/2019 15:14:17 PDT

1 newpage

In [25]:
```
#each interval is poisson distribution with mu = 40. find P(X >= 30) for each interval and mul

(1-stats.poisson.cdf(29, 40))**2
```

Out[25]: 0.9154113546571047

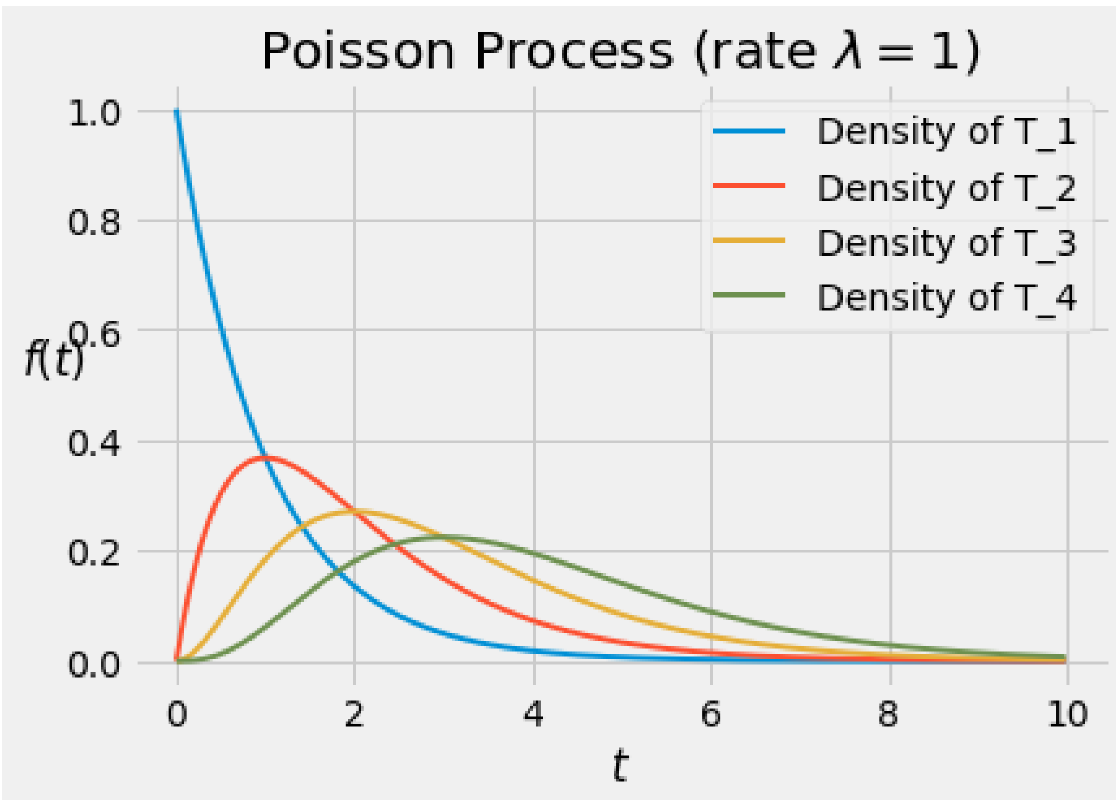
2 newpage

Distribution of G2 is same as distribution of T1 which is exponential distribution with parameter lambda.

It is because each Ti is independent from the other. So the waiting time for T2 from T1 is same distribution as T1.

It is a exponential distribution with parameter n*lambda

In [30]:
```
t = np.arange(0, 10, 0.01)
for r in np.arange(1, 5):
    f = stats.gamma.pdf(t, r, scale = 1)
    plt.plot(t, f, lw=2, label='Density of T_'+str(r))  # plot the density of T_r
plt.xlabel('$t$')
plt.ylabel('$f(t)$', rotation=0)
plt.legend()
plt.title('Poisson Process (rate $\lambda = 1$)');
```



In [32]:
```
# lambda = 1, t = 15/2
# P(T10 > t)
stats.poisson.cdf(9, 7.5)
```

Out[32]: 0.7764076130197146

In [33]:
```
#consider second call as origin
#r = 3, lambda = 1, t = 7/2
#P(Tr < 7/2)
1- stats.poisson.cdf(2, 3.5)
```

Out[33]: 0.6791528011378658

3 newpage

For each $t > 0$, the total number of points $N_{(0,t)}$ has the Poisson λt distribution, and the number of blue points $B_{(0,t)}$ has the Poisson pλt distribution.

We know from above that N(0,t) has poissonλt distribution. And considering blue points as the number of successes in Bernoulli trials, number of blue points will have Poisson pλt distribution.

Yes. Following from our argument part 1, that disjoint time intervals are independent for the total number of arrivals, since blue points are also part of the total arrivals, blue points will also be indenpendent if intervals are disjoint

The blue process is a Poisson process with rate pλper unit time.

$R_{(0,t)}$ has the poisson (1-p)λt distribution.

They are independent. The for poissonization of binomial, the distribution of success and failure are independent

The blue process and the red process are independent Poisson processes. The blue process has rate p$\lambda$and the red process has rate (1-p)λ.

```
In [35]: #lambda = 3, t = 5
         #Number of cell phone = poisson distribution with 3*5*0.05
         #Number of other items = poisson distribution with 3*5*0.95
         #P(cell >= 1)*p(other <= 10)
         (1- stats.poisson.pmf(0, 3*5*0.05)) * (stats.poisson.cdf(10, 3*5*0.95))

Out[35]: 0.08426146935195485
```

```
In [59]: #P(other = 15)*P(cell = 1)
         0.95**15*0.05

Out[59]: 0.023164561507987652
```

```
In [62]: """Among the first 22 items that arrive, there should be less than 3 cell phones."""

         stats.binom.cdf(2, 22, 0.05)

Out[62]: 0.905176954081569
```

4 newpage

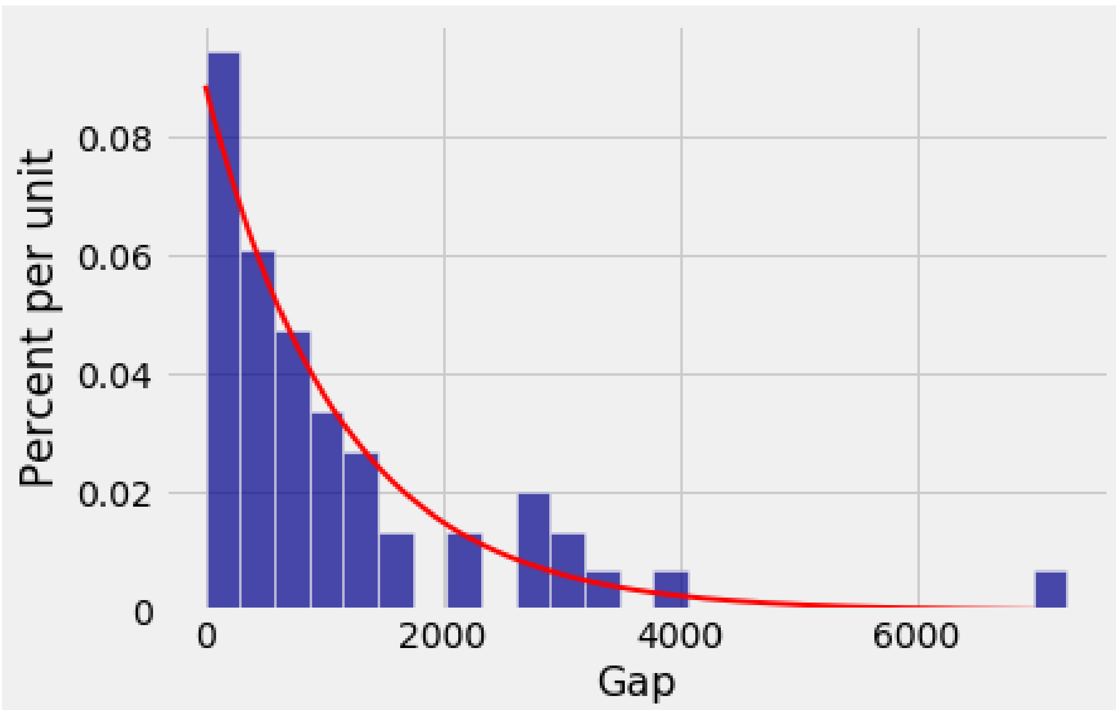
In [66]:
```
gaps = quakes.column('Gap')
mean_gap = np.mean(gaps)

lam = 1 / np.mean(gaps)

mean_gap, lam
```

Out[66]: (1128.8235294117646, 0.0008858780614903596)

In [67]:
```
quakes.hist('Gap', bins=25)
t = np.arange(7000)
f = stats.expon.pdf(t, scale = 1/lam)
plt.plot(t, f, color='red', lw=2);
```



In [68]:
```
# P(at least one quake within a year from now)
# t = 365, lambda = lam
1-stats.poisson.pmf(0, 365*lam)
```

Out[68]: 0.2762762299248045

In [69]:
```
# number of quakes with magnitude at least 6.0
num_big = sum(quakes.column('Magnitude') > 6)

# proportion of quakes with magnitude at least 6.0
prop_big = num_big / 51

num_big, prop_big
```

Out[69]: (35, 0.6862745098039216)

In [76]:
```
# P(1 quake of magnitude at least 6.0 and at least one quake of magnitude in [4.9, 6.0))
big = prop_big*365*lam
small = (1-prop_big)*365*lam

stats.poisson.pmf(1, big)*(1-stats.poisson.pmf(0, small))
```

Out[76]: 0.017146209255629666

In [81]:
```
# number of years n such that
# P(at least quake of magnitude 6.0 will happen in n years) = 0.99
# want 1 - stats.poisson.pmf(0, 365*n*lam*prop_big) = 0.99
# stats.poisson.pmf(0, 365*n*lam*prop_big) = e^(-365*n*lam*prop_big) = 0.01
n = np.log(0.01) / (-365*lam*prop_big)
n
```

Out[81]: 20.75300568354868