

```
In [19]: import scipy.stats as sps
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
```

**Адаптировать задачу из предыдущего раздела (УМО II) для случая, когда параметр  $\lambda$  неизвестен, и его нужно оценивать (даже в начале при отсутствии информации) по мере поступления новой информации (с помощью байесовской оценки)**

```
In [23]: data = np.loadtxt('6_1.txt')
```

**Посчитаем среднее время между ближайшими поломками**

```
In [49]: Xi = data[1:] - data[:-1]
mean = Xi.mean()
```

**Оценим лямбду тупо из выборочного среднего (т.к. матожидание -  $1/\lambda$ )**

$$\Rightarrow \lambda = \frac{1}{mean}$$

```
In [72]: lambda_ = 1 / mean
lambda_
```

```
Out[72]: 0.34580538073172418
```

**Матожидание гамма распределения:  $\Gamma(\alpha, \beta)$  - это  $\alpha/\beta$**

**Апостериорное распределение -  $\Gamma(\alpha_{prior} + n, \beta_{prior} + \sum x_i)$**

**Поскольку особо таких знаний о том как часто ломаются сервера у нас нет, то просто, попробуем взять параметры априорного распределения из выборки -  $\alpha/\beta = \lambda$ , где лямбда - это 1 делённая на среднее, которое мы посчитали (т.к. матожидание экспоненциального распределения -  $\frac{1}{\lambda}$ )**

```
In [45]: prior_alpha = 1
prior_beta = lambda_
```

**Типа подобрали параметры априорного распределения**

```
In [73]: est.insert(0,lambda_)
```

**Оценили параметр без данных****Выводим результаты**

```
In [95]: N = np.zeros(61);

for s in range(0, 61) :
    for time_of_breakage in data :
        if (time_of_breakage > s) :
            break
        N[s]+=1;
est = [ (prior_alpha + N[n]) / (prior_beta + Xi[:N[n]].sum()) for n in range (1, 61)]
est.insert(0, lambda_)

print("Time \t CondMean \t BreakIfNeeded \t lambda")
for s in range(0, 61) :
    break_occured
    if (s != 0 and N[s] != N[s-1]) :
        break_occured = "break"
    else :
        break_occured = ""
    print(s, "\t", N[s] + est[s]*(60 - s), "\t", break_occured, "\t\t", est[s])
```

Time	CondMean	BreakIfNeeded	lambda
0	20.7483228439		0.345805380732
1	170.6162		2.8918
2	61.6122237757	break	1.04503834096
3	73.137206602	break	1.24802116846
4	65.3294742673	break	1.1130263262
5	71.7372371851	break	1.23158613064
6	28.7627154149	break	0.440050285461
7	28.3226651294		0.440050285461
8	27.882614844		0.440050285461
9	27.4425645585		0.440050285461
10	27.0025142731		0.440050285461
11	26.5624639876		0.440050285461
12	26.1224137021		0.440050285461
13	25.6823634167		0.440050285461
14	25.2423131312		0.440050285461
15	29.5764702004	break	0.501699337787
16	29.0747708626		0.501699337787
17	29.8288819629	break	0.507648417743
18	29.3212335452		0.507648417743
19	26.7112513833	break	0.407591497153
20	26.3036598861		0.407591497153
21	25.896068389		0.407591497153
22	25.4884768918		0.407591497153
23	25.0808853947		0.407591497153
24	24.6732938975		0.407591497153
25	24.2657024003		0.407591497153
26	23.8581109032		0.407591497153
27	23.450519406		0.407591497153
28	24.2264173222	break	0.413325541319
29	23.8130917809		0.413325541319
30	25.1155014975	break	0.43718338325
31	24.5364488738	break	0.397808581855
32	24.1386402919		0.397808581855
33	23.7408317101		0.397808581855
34	23.3430231282		0.397808581855
35	22.9452145464		0.397808581855
36	22.5474059645		0.397808581855
37	21.9570448036	break	0.345958469721
38	21.6110863339		0.345958469721
39	21.2651278642		0.345958469721
40	20.9191693944		0.345958469721
41	20.5732109247		0.345958469721
42	20.227252455		0.345958469721
43	19.8812939853		0.345958469721
44	19.5353355155		0.345958469721
45	20.0759483075	break	0.338396553836
46	19.7375517537		0.338396553836
47	19.3991551999		0.338396553836
48	19.060758646		0.338396553836
49	19.7046718506	break	0.336788350051
50	19.3678835005		0.336788350051
51	19.0310951505		0.336788350051
52	19.7382253496	break	0.342278168703
53	19.3959471809		0.342278168703
54	20.146536703	break	0.35775611716
55	20.8129143007	break	0.362582860145
56	20.4503314406		0.362582860145
57	21.0828127382	break	0.360937579413
58	20.7218751588		0.360937579413
59	20.3609375794		0.360937579413
60	21.0 break		0.378125083195

/home/avk/Programs/anaconda3/lib/python3.5/site-packages/ipykernel/\_\_main\_\_  
 \_\_.py:8: DeprecationWarning: using a non-integer number instead of an integer will result in an error in the future

**Если иметь какие-либо начальные знания о распределении, то можно оценить параметр байесовской оценкой при малом количестве данных гораздо точнее, в этом я убедился в предыдущей задаче.**

**Чтобы дать более точную оценку, нужно знать что-то ещё о серверах**