

Задача 6.1

In [2]:

```
import numpy as np
import pandas as pd
import scipy.stats as sps
import matplotlib.pyplot as plt
%pylab inline
```

Populating the interactive namespace from numpy and matplotlib

In [3]:

```
file_obj = open('6.csv')
print file_obj.readline().strip()
print file_obj.readline().strip()
print file_obj.readline().strip()
```

```
lambda = 95
t_0 = 500
t = 100000
```

In [4]:

```
lambda_ = 95
t_0 = 500
t = 100000
```

In [5]:

```
data = np.array([float(line.strip()) for line in file_obj])
```

Найдём $E(N_t|N_s)$:

$$E(N_t|N_s) = E(N_t - N_s + N_s|N_s) = E(N_t - N_s|N_s) + E(N_s|N_s)$$

$$(N_t - N_s) \perp\!\!\!\perp N_s \Rightarrow E(N_t - N_s|N_s) = E(N_t - N_s)$$

$$N_t - N_s \sim \text{Pois}(\lambda, t - s) \Rightarrow E(N_t - N_s) = \lambda \cdot (t - s)$$

$N_s - F_{N_s}$ -измеримая случайная величина, значит $E(N_s|N_s) = N_s$. Тогда

$$E(N_t|N_s) = \lambda \cdot (t - s) + N_s$$

$E(N_t|N_s)$ для $s = k \cdot t_0$, где $0 \leq k \cdot t_0 \leq t$:

In [18]:

```
s = np.arange(0, t + t_0, t_0)
N_s = np.array([(data <= cur_t).sum(0) for cur_t in s])
result = 1 / float(lambda_) * (t - s) + N_s
```

Вывод программы, предсказывающей сколько серверов нужно докупить к моменту времени t:

In [28]:

```
for i in range(s.size):  
    print r'time = %d: %d' % (s[i], result[i])
```

```
time = 0: 1052  
time = 500: 1051  
time = 1000: 1055  
time = 1500: 1055  
time = 2000: 1059  
time = 2500: 1058  
time = 3000: 1056  
time = 3500: 1054  
time = 4000: 1055  
time = 4500: 1056  
time = 5000: 1060  
time = 5500: 1058  
time = 6000: 1055  
time = 6500: 1054  
time = 7000: 1055  
time = 7500: 1050  
time = 8000: 1050  
time = 8500: 1051  
time = 9000: 1056  
time = 9500: 1055  
time = 10000: 1058  
time = 10500: 1059  
time = 11000: 1061  
time = 11500: 1060  
time = 12000: 1060  
time = 12500: 1059  
time = 13000: 1056  
time = 13500: 1060  
time = 14000: 1064  
time = 14500: 1065  
time = 15000: 1066  
time = 15500: 1066  
time = 16000: 1067  
time = 16500: 1065  
time = 17000: 1067  
time = 17500: 1067  
time = 18000: 1063  
time = 18500: 1058  
time = 19000: 1060  
time = 19500: 1060  
time = 20000: 1061  
time = 20500: 1060  
time = 21000: 1064  
time = 21500: 1064  
time = 22000: 1065  
time = 22500: 1066  
time = 23000: 1069  
time = 23500: 1070  
time = 24000: 1070  
time = 24500: 1068  
time = 25000: 1068  
time = 25500: 1067
```

```
time = 26000: 1066
time = 26500: 1067
time = 27000: 1067
time = 27500: 1066
time = 28000: 1066
time = 28500: 1065
time = 29000: 1065
time = 29500: 1064
time = 30000: 1066
time = 30500: 1064
time = 31000: 1065
time = 31500: 1067
time = 32000: 1067
time = 32500: 1065
time = 33000: 1066
time = 33500: 1070
time = 34000: 1072
time = 34500: 1075
time = 35000: 1077
time = 35500: 1077
time = 36000: 1077
time = 36500: 1078
time = 37000: 1078
time = 37500: 1078
time = 38000: 1080
time = 38500: 1082
time = 39000: 1080
time = 39500: 1079
time = 40000: 1080
time = 40500: 1078
time = 41000: 1080
time = 41500: 1078
time = 42000: 1082
time = 42500: 1083
time = 43000: 1084
time = 43500: 1084
time = 44000: 1080
time = 44500: 1077
time = 45000: 1079
time = 45500: 1087
time = 46000: 1085
time = 46500: 1086
time = 47000: 1092
time = 47500: 1091
time = 48000: 1087
time = 48500: 1088
time = 49000: 1091
time = 49500: 1092
time = 50000: 1091
time = 50500: 1087
time = 51000: 1085
time = 51500: 1084
time = 52000: 1080
time = 52500: 1081
time = 53000: 1084
time = 53500: 1083
time = 54000: 1081
```

```
time = 54500: 1083
time = 55000: 1083
time = 55500: 1080
time = 56000: 1080
time = 56500: 1078
time = 57000: 1086
time = 57500: 1084
time = 58000: 1085
time = 58500: 1087
time = 59000: 1087
time = 59500: 1087
time = 60000: 1084
time = 60500: 1082
time = 61000: 1082
time = 61500: 1088
time = 62000: 1091
time = 62500: 1090
time = 63000: 1089
time = 63500: 1091
time = 64000: 1088
time = 64500: 1088
time = 65000: 1090
time = 65500: 1088
time = 66000: 1083
time = 66500: 1083
time = 67000: 1082
time = 67500: 1080
time = 68000: 1080
time = 68500: 1079
time = 69000: 1077
time = 69500: 1076
time = 70000: 1078
time = 70500: 1078
time = 71000: 1078
time = 71500: 1076
time = 72000: 1075
time = 72500: 1073
time = 73000: 1074
time = 73500: 1069
time = 74000: 1071
time = 74500: 1071
time = 75000: 1069
time = 75500: 1070
time = 76000: 1070
time = 76500: 1068
time = 77000: 1067
time = 77500: 1071
time = 78000: 1071
time = 78500: 1073
time = 79000: 1074
time = 79500: 1072
time = 80000: 1070
time = 80500: 1071
time = 81000: 1071
time = 81500: 1067
time = 82000: 1063
time = 82500: 1065
```

```
time = 83000: 1059
time = 83500: 1059
time = 84000: 1056
time = 84500: 1056
time = 85000: 1054
time = 85500: 1053
time = 86000: 1060
time = 86500: 1059
time = 87000: 1055
time = 87500: 1055
time = 88000: 1052
time = 88500: 1056
time = 89000: 1055
time = 89500: 1052
time = 90000: 1053
time = 90500: 1053
time = 91000: 1051
time = 91500: 1050
time = 92000: 1048
time = 92500: 1050
time = 93000: 1054
time = 93500: 1053
time = 94000: 1053
time = 94500: 1055
time = 95000: 1052
time = 95500: 1047
time = 96000: 1042
time = 96500: 1036
time = 97000: 1031
time = 97500: 1026
time = 98000: 1021
time = 98500: 1015
time = 99000: 1010
time = 99500: 1005
time = 100000: 1000
```

Разброс получившихся значений:

In [29]:

```
round(result.max() - result.min())
```

Out[29]:

93.0