In [1]:

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sts
%matplotlib inline

Выкладки

Сведение задачи к линейной модели

$$X_i = \beta_1 + i\beta_2 + \epsilon_0 + \epsilon_1 + \ldots + \epsilon_i | i \in \{0, 1, \ldots, n\} |$$

$$Y_i = X_i - X_{i-1} = \beta_2 + \epsilon_i | i \in \{1, ..., n\} |$$

$$Y_0 = \beta_1 + \epsilon_0$$

$$Z_{(n+1)\times 2} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{pmatrix}$$

$$(Z^T Z) = \begin{pmatrix} 1 & 0 \\ 0 & n \end{pmatrix}$$

$$(Z^T Z)^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{n} \end{pmatrix}$$

$$Z\hat{\theta} = \frac{1}{n} \begin{pmatrix} n & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{pmatrix} Y$$

$$||Y - Z\hat{\theta}||^2 = ||(E - \frac{1}{n} \begin{pmatrix} n & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 1 \end{pmatrix})Y||^2$$

ОНК для β_1 и β_2

$$\hat{\theta} = (Z^T Z)^{-1} Z^T Y = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix} Y = \frac{1}{n} \begin{pmatrix} n & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 1 \end{pmatrix} Y = \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix}$$

Несмещенная оценка для σ^2

$$\sigma^2 = \frac{\|Y - Z\hat{\theta}\|^2}{(n+1)-k} = \frac{1}{n-1}\|Y - Z\hat{\theta}\|^2$$

Оценка дисперсии отсчета времени

$$\epsilon_i = \epsilon_i^t \beta_2 \Rightarrow \sigma_t^2 = \frac{\sigma^2}{\beta_2}$$

Вычисления

Считывание данных

In [2]:

```
file_obj = open('Regression.csv', 'r')

# исходная выборка
X = np.array([])

for line in file_obj:
    line = line[:-1]
    X = np.append(X, float(line))
file_obj.close()
```

In [3]:

```
print X.size, X
                                                             120.4
1000 [
          83.7221
                       90.586
                                    98.7931
                                                107.9353
289
       128.1435
                                           168.3627
    136.9536
                 144.848
                              155.1632
                                                        180.2104
187.4253
    196.2578
                 204.3863
                              213.5371
                                           223.8599
                                                        232.1265
244.0747
    252.7152
                 263.7067
                              275.0745
                                           284.6497
                                                        295.322
303.6249
                 322.6255
                              329.9894
                                           337.8079
                                                        347.0841
    313.271
357.7277
    365.5541
                 378.2801
                              389.2964
                                           402.3789
                                                        413.0363
422,618
    433.872
                 446.1038
                              455.5501
                                           468.5738
                                                        476.5022
484.8989
    495.9649
                                           527.4497
                 507.8976
                              517.448
                                                       534.6791
543.689
                                                        596.2015
    552.807
                 562.4875
                              575.1784
                                           585.1601
607.9615
    618.0607
                 628.5974
                              637.27
                                           642.4356
                                                       652.3472
    2706
```

In [4]:

```
Y = np.zeros_like(X)
Y[0] = X[0]

for i in xrange(1, X.size):
    Y[i] = X[i] - X[i-1]
```

In [5]:

rint Y.si	ze, Y					
-		8639 8.	2071 9.	1422 12.	4936 7.	7146
		11.8477	7.2149	8.8325	8.1285	9.150
		8 6405	10 0015	11 3679	0 5752	10.672
		0.0403	10.9913	11.30/0	3.3/32	10.072
9.6461		7.3639	7.8185	9.2762	10.6436	7.826
_	12 0025	10 6574	0 5017	11 254	12 2210	0 446
		10.65/4	9.581/	11.254	12.2318	9.446
		11.066	11.9327	9.5504	10.0017	7.229
		12.6909	9.9817	11.0414	11.76	10.099
		0 0116	0 0314	10 0823	12 36/1	10.072
		9.9110	9.0014	10.0023	12.3041	10.072
		10.8974	11.7189	9.6636	11.0944	13.893
	8.9441	9.2825	10.3039	8.2481	11.52	12.009
	000 [83. .8101 7 10.3152 10.3228 8.2666 8.3029 9.6461 12.726 11.0163 13.0237 7.9284 9.0099 9.118 10.5367 8.6726 8.4658 9.3562 10.9897	.8101 7.8944 10.3152 13.1995 10.3228 8.2666 11.9482 8.3029 9.6461 9.3545 12.726 11.0163 13.0825 13.0237 7.9284 8.3967 9.0099 9.118 9.6805 10.5367 8.6726 5.1656 8.4658 9.3562 8.9073 10.9897 9.7399 8.9441	000 [83.7221 6.8639 88101 7.8944 10.3152 13.1995 11.8477 10.3228 8.2666 11.9482 8.6405 8.3029 9.6461 9.3545 7.3639 12.726 11.0163 13.0825 10.6574 13.0237 7.9284 8.3967 11.066 9.0099 9.118 9.6805 12.6909 10.5367 8.6726 5.1656 9.9116 8.4658 9.3562 8.9073 10.8974 10.9897 9.7399 8.9441 9.2825	000 [83.7221 6.8639 8.2071 9.8101 7.8944 10.3152 13.1995 11.8477 7.2149 10.3228 8.2666 11.9482 8.6405 10.9915 8.3029 9.6461 9.3545 7.3639 7.8185 12.726 11.0163 13.0825 10.6574 9.5817 13.0237 7.9284 8.3967 11.066 11.9327 9.0099 9.118 9.6805 12.6909 9.9817 10.5367 8.6726 5.1656 9.9116 9.0314 8.4658 9.3562 8.9073 10.8974 11.7189 10.9897 9.7399 8.9441 9.2825 10.3039	000 [83.7221 6.8639 8.2071 9.1422 128101 7.8944 10.3152 13.1995 11.8477 7.2149 8.8325 10.3228 8.2666 11.9482 8.6405 10.9915 11.3678 8.3029 9.6461 9.3545 7.3639 7.8185 9.2762 12.726 11.0163 13.0825 10.6574 9.5817 11.254 13.0237 7.9284 8.3967 11.066 11.9327 9.5504 9.0099 9.118 9.6805 12.6909 9.9817 11.0414 10.5367 8.6726 5.1656 9.9116 9.0314 10.0823 8.4658 9.3562 8.9073 10.8974 11.7189 9.6636 10.9897 9.7399 8.9441 9.2825 10.3039 8.2481	000 [83.7221 6.8639 8.2071 9.1422 12.4936 7.8101 7.8944 10.3152 13.1995 11.8477 7.2149 8.8325 8.1285 10.3228 8.2666 11.9482 8.6405 10.9915 11.3678 9.5752 8.3029 9.6461 9.3545 7.3639 7.8185 9.2762 10.6436 12.726 11.0163 13.0825 10.6574 9.5817 11.254 12.2318 13.0237 7.9284 8.3967 11.066 11.9327 9.5504 10.0017 9.0099 9.118 9.6805 12.6909 9.9817 11.0414 11.76 10.5367 8.6726 5.1656 9.9116 9.0314 10.0823 12.3641 8.4658 9.3562 8.9073 10.8974 11.7189 9.6636 11.0944 10.9897 9.7399 8.9441 9.2825 10.3039 8.2481 11.52

ОНК для β_1 и β_2

$$\hat{\theta} = (Z^T Z)^{-1} Z^T Y = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix} Y = \frac{1}{n} \begin{pmatrix} n & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 1 \end{pmatrix} Y = \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix}$$

In [6]:

```
beta_1 = Y[0]
beta_2 = np.mean(Y[1:])
```

In [7]:

```
print 'ОНК для beta_1 и beta_2:', '\nbeta_1 = ', beta_1, '\nbeta_2 = ', beta_2

OHK для beta_1 и beta_2:
beta_1 = 83.7221
beta_2 = 9.97799349349
```

Несмещенная оценка для σ^2

$$\sigma^2 = \frac{\|Y - Z\hat{\theta}\|^2}{(n+1)-k} = \frac{1}{n-1}\|Y - Z\hat{\theta}\|^2$$

In [8]:

```
Z = np.zeros(Y.size * 2).reshape(Y.size, 2)
Z[0][0] = 1
for i in xrange(1, Z.size/2):
        Z[i][1] = 1
```

In [9]:

```
print Z
```

```
[[ 1. 0.]
 [ 0. 1.]
 [ 0. 1.]
 ...,
 [ 0. 1.]
 [ 0. 1.]
 [ 0. 1.]
```

In [10]:

```
theta = np.linalg.inv((np.transpose(Z).dot(Z))).dot(np.transpose(Z)).dot(Y)
```

In [11]:

```
print theta
```

[83.7221 9.97799349]

```
In [12]:
```

```
Y_Z_Theta = Y - Z.dot(theta)
```

In [13]:

```
sigma_2 = float(Y_Z_Theta.dot(np.transpose(Y_Z_Theta)))/(Y.size - 2)
```

In [14]:

```
print 'Несмещенная оценка сигма в квадрате равна ', sigma_2
```

Несмещенная оценка сигма в квадрате равна 2.70470552438

Оценка дисперсии отсчета времени

```
In [15]:
```

```
sigma_2_t = float(sigma_2)/(beta_2 * beta_2)
```

In [16]:

```
print 'Оценка дисперсии отсчета времени ', sigma_2_t
```

Оценка дисперсии отсчета времени 0.027166491595

Вывод

С вероятностью порядка 0.99 ошибка отсчета времени лежит в интервале:

```
In [17]:
```

```
print '[', -sigma_2_t*3, ', ', sigma_2_t*3, ']'
[ -0.0814994747849 ,  0.0814994747849 ]
```

Так как измерения проводятся с интервалом 1 секунда(по показаниям сечтика), то относительная погрешность равна:

In [18]:

```
print sigma_2_t*6, '%'
```

0.16299894957 %

Данная погрешность является заметной, однако приемлемой для эксперементов.