In [1]:

```
%pylab inline

import pandas
import scipy.stats as sps
import numpy as np
import matplotlib.pyplot as plt
import math
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [2]:

```
# читаем данные из файла
X = [float(x) for x in open('/Users/evgeniatveritinova1/Downloads/file3.txt').
size = len(X)
```

$X_i = \beta_1 + i\beta_2 + \varepsilon_0 + \dots + \varepsilon_i$, для $i = 0 \dots n$

Рассмотрим Y_i:

$Y_0 = X_0 = \beta_1 + \varepsilon_0$

$Y_i = X_i - X_{i-1} = \beta_2 + \varepsilon_i$, для $i = 0 \dots n$

Тогда для Y имеем линейную модель $Z = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{pmatrix}$

In [3]:

```
Z = [[0,1] for i in range(size)]
Z[0] = [1,0]
Z = np.matrix(Z)
Y = [X[k] - X[k-1] for k in range(size)]
Y[0] = X[0]
Y = np.matrix(Y).T
```

$\hat{\beta} = \binom{\hat{\beta}_1}{\hat{\beta}_2} = (Z^T Z)^{-1} Z^T Y = \begin{pmatrix} X_0 \\ \frac{X_n - X_1}{n} \end{pmatrix}$

In [4]:

```
beta_est = (Z.T * Z).I * Z.T * Y
print beta_est
```

```
[[ 681.456492  ]
 [   4.65300028]]
```

$\hat{\sigma}^2 = \frac{1}{n-1}|Y - Z\hat{\beta}| = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - X_{i-1} - \frac{X_n - X_1}{n})^2$

In [5]:

```
sigma_est = ((Y - Z * beta_est).T * (Y - Z * beta_est)) / (size - 1)
print sigma_est
```

```
[[ 0.08815892]]
```

$\hat{\sigma}_t^2 = \frac{\hat{\sigma}^2}{\beta_2^2}$

In [6]:

```
sigma_time_est = sigma_est / (beta_est[1]**2)
print sigma_time_est
```

```
[[ 0.00407193]]
```

Так как $\hat{\sigma}_t^2 \ll 1$ и $\hat{\sigma}_t^2 \ll Y_i$, то можно сделать вывод, что линейная модель применима в данном случае для получения достаточно точных оценок