

In [9]:

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

Считывание данных

In [10]:

```
file_obj = open('13.csv', 'r')

# массив данных
Y = np.array([])
X = np.array([])

# приведение данных к float
for line in file_obj:
    line = line.split(',')
    Y = np.append(Y, float(line[0]))
    X = np.append(X, float(line[1]))
    X = np.append(X, float(line[2]))
    X = np.append(X, float(line[3]))
    X = np.append(X, float(line[4]))
file_obj.close()
X = X.reshape(X.size/4, 4)
Y = Y.reshape(Y.size, 1)
```

ОНК для θ

In [11]:

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

Печать θ

In [12]:

```
for t in theta:
    print round(t,3)
```

```
295.92
-2.293
4.293
-29282.885
```

Печать $\det|X^T X|$

In [13]:

```
print np.linalg.det(np.transpose(X).dot(X))
```

0.671186903151

Поскольку det отличен от нуля, то исходные данные не являются линейно зависимыми.

Оценки по λ

In [14]:

```
E = np.eye(X.size/Y.size)
N = 100
step = 0.1
lam = np.arange(0, N, step)
mark = np.array([])
```

In [*]:

```
for i in xrange(Y.size):
    cur_X = np.delete(X, i, 0)
    cur_Y = np.delete(Y, i, 0)
    X_T_X = np.transpose(cur_X).dot(cur_X)
    X_T_Y = np.transpose(cur_X).dot(cur_Y)

    for l in xrange(lam.size):
        mark = np.append(mark, np.linalg.inv( X_T_X - lam[l]*E ).dot(X_T_Y))

mark = mark.reshape(Y.size, lam.size, X.size/Y.size)
```

Не знаю почему, но у меня этот шаг выполнялся очень долго и я так и не дождался результатов.

Дисперсии оценок для каждого λ

In [*]:

```
disp_lam = np.zeros(lam.size * X.size/Y.size).reshape(lam.size, X.size/Y.size)

for l in xrange(lam.size):
    for i in xrange(X.size/Y.size):
        disp_lam[l][i] = np.var(mark[:Y.size + 1, l, i])
```

Графики дисперсий от λ

In [*]:

```
plt.figure(figsize=(16, 8 * X.size/Y.size))

for i in xrange(X.size/Y.size):
    plt.subplot(X.size/Y.size, 1, i + 1)
    plt.plot(lam, disp_lam[:lam.size + 1, i], label='$X_{' + str(i) + '}$')

    plt.ylim((0, 0.5)) # размеры графика по вертикальной оси
    plt.xlabel('$n$') # название горизонтальной оси (аналогично plt.ylabel)
    plt.title('Dispersion$(\\lambda)$ for ' + '$X_{' + str(i) + '}$') # имя
    plt.grid() # добавляем сетку
    plt.legend()

plt.show()
```

In []: