

## 3\_2

March 28, 2016

```
In [2]: __author__ = 'Security'
import numpy as np
import scipy.stats as stats
%matplotlib inline
import matplotlib.pyplot as plt
```

```
In [18]: grid = np.arange(-1000, 1000, 0.1)
cauchySample = [float(line.rstrip('\n')) for line in open('Cauchy.txt', 'r')]
```

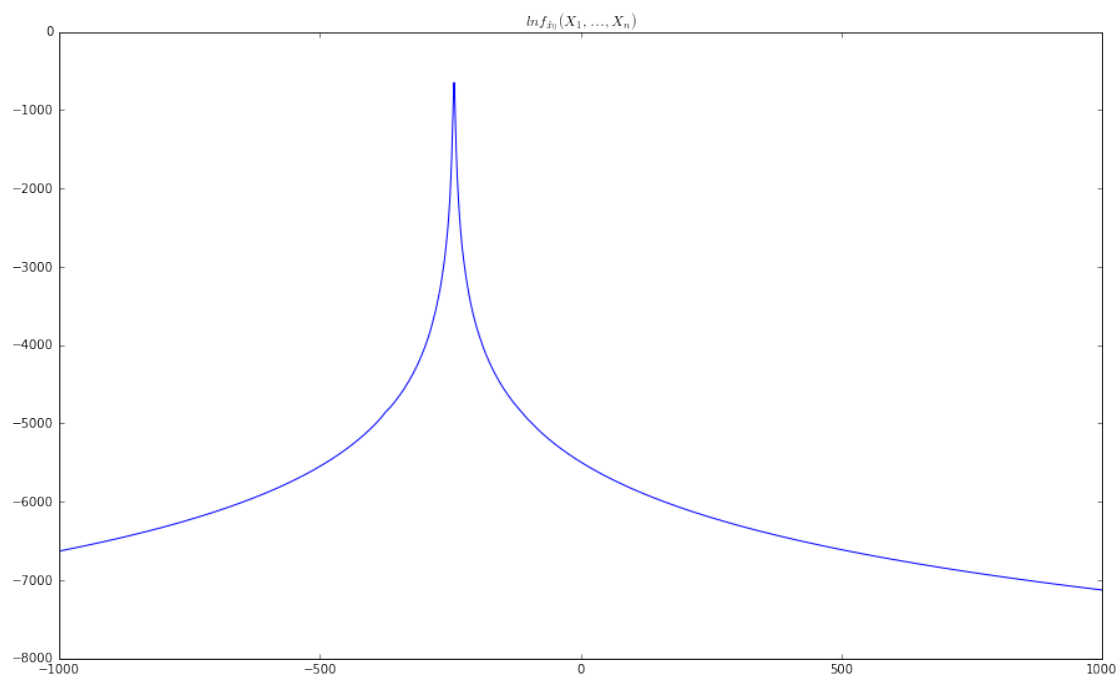
Здесь функция правдоподобия смещены на некоторую константу, но т.к. нас интересует аргумент, при котором достигается максимум, то она влияет ни на что не будет. Также заметим, что искать мы будем логарифм от этой функции, дабы облегчить вычисления.

```
In [19]: def likelihoodFunction(x0, sample):
return np.sum([-np.log(1 + (x - x0)**2) for x in sample])
```

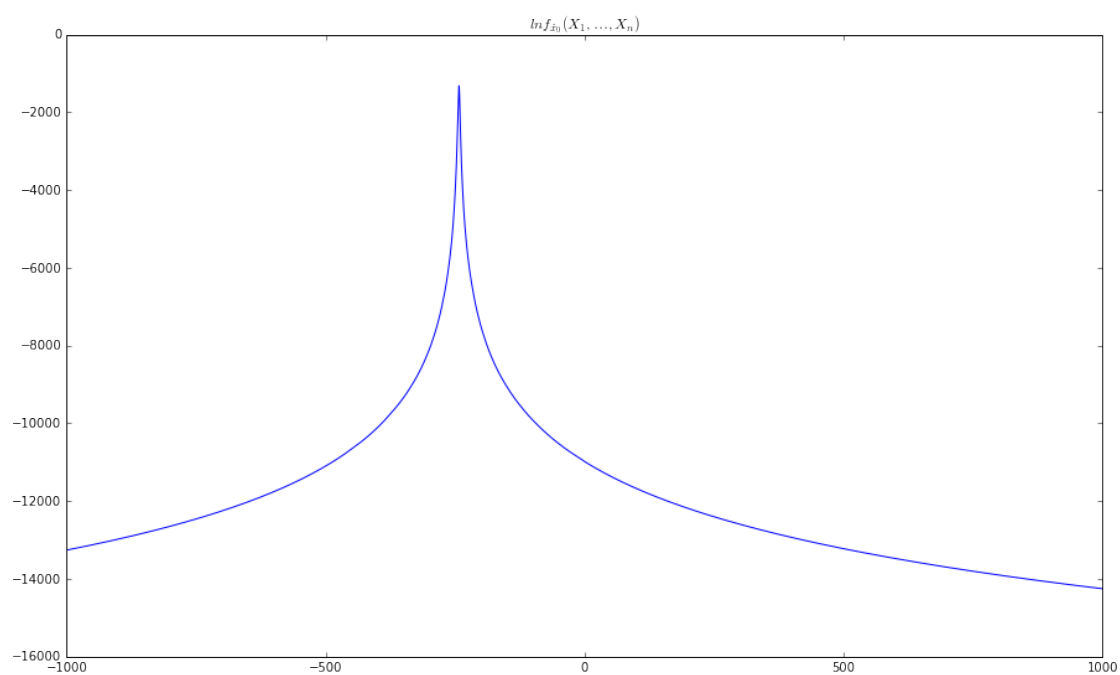
```
def likelihoodFunctionValues(sample, grid):
return [likelihoodFunction(x0, sample) for x0 in grid]
```

```
def estimationForCauchySample(sample, grid):
return grid[np.argmax(likelihoodFunctionValues(sample, grid))]
```

```
In [20]: for sample in [cauchySample[:int((len(cauchySample)/2))], cauchySample]:
plt.figure(figsize=(15, 9))
plt.title(r'$\ln f_{\hat{x}_{\{0\}}}(X_{\{1\}}, \ldots, X_{\{n\}})$')
plt.plot(grid, likelihoodFunctionValues(sample, grid))
plt.show()
print('Estimation for first {:} elements = {}'.format(len(sample), estimationForCauchySample(sample, grid)))
```



Estimation for first 500 elements = -242.7999999982783



Estimation for first 1000 elements = -242.6999999998278

Итак, оценка отклонения для первой половины элементов равна  $-242.7999999982783$ , а для всех  $-242.6999999998278$

In [ ]: