

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

## Курсовой проект

Тема: **"Робастные оценки параметров регрессии при наличии  
группированной выборки"**

АНДРЕЙ РУМЯНЦЕВ, 7 ГРУППА

Научный руководитель  
Агеева Е. С.

Минск, 2017

# 1 Введение

Существует несколько подходов для оценки параметров регрессии, но далеко не все устойчивы к возникновению аномальных наблюдений. В реальной жизни аномальные наблюдения возникают постоянно, поэтому большинство методов просто неприменимо. В прошлом веке в работах Хьюбера была заложена теория робастного оценивания. Были предложены следующие робастные оценки:

- М-Оценки
- R-Оценки
- L-Оценки

М-оценки – некоторое подобие оценок максимального правдоподобия (ММП-оценки – частный случай), L-оценки строятся на основе линейных комбинаций порядковых статистик, R-оценки – на основе ранговых статистик. В данном курсовом проекте я буду моделировать функцию регрессии с аномальными наблюдениями, анализировать точность методов и находить для разных методов так называемый ”breakpoint” – процент аномальных наблюдений, при котором увеличение количества наблюдений не повысит точность методов.

## 2 Теоретические сведения

На данном этапе будем работать с линейной регрессией:

$$y_i = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_n x_{in} + \epsilon_i, \quad (1)$$

Где  $y_i$  –  $i$ -е наблюдение из  $N$  наблюдений,  $x_i$  регрессоры,  $\{\alpha, \beta_k, k = \overline{1, n}\}$  – параметры регрессии, а  $\epsilon_i$  – ошибка, распределение которой подчиняется нормальному закону с нулевым ожиданием и дисперсией  $\sigma^2$ .

В нашей задаче считаем параметры  $\{\alpha, \beta_k, k = \overline{1, n}\}$  неизвестными, их нам и требуется найти.

Теперь рассмотрим некоторые методы оценки параметров регрессии:

### 2.1 Метод Наименьших Квадратов

### 2.2 М-оценки

### 2.3 L-оценки

## 3 Моделирование регрессии на языке Python

Подключим необходимые библиотеки:

```

import numpy as np
import matplotlib.pyplot as plt
from random import random
import pylab
import scipy
from outliers import smirnov_grubbs as grubbs
from matplotlib.backends.backend_pdf import PdfPages
from statsmodels.robust.scale import mad
import theano
import theano.tensor as T
import statsmodels.api as sm
import statsmodels.formula.api as smf

```

Заведем константы для моделирования: количество наблюдений, процент аномальных наблюдений, и параметры регрессии, используемые в моделировании:

```

SAMPLE_QUANTITY=100
OUTLIER_PERCENTAGE = 10.0
regressionParameters = np.matrix([100,4]).T

```

Проинициализируем результирующий вектор y:

```

y_points = np.zeros(shape = SAMPLE_QUANTITY)

```

Теперь моделируем y:

```

x_points = np.zeros(shape=[SAMPLE_QUANTITY,len(regressionParameters)])
y_points = np.zeros(shape = SAMPLE_QUANTITY)
# plt.plot(x_points,y_points,'ro')
# # plt.hist(y_points,bins="auto")
# plt.show()
for i in range(0,SAMPLE_QUANTITY):
    if random()>OUTLIER_PERCENTAGE/100:
        x_points[i] = np.append(np.ones(1),np.random.uniform(-5,5,size = len(regressionParameters)))
        # print(x_points[i])
        y_points[i]=(x_points[i]*regressionParameters)+np.random.normal(0,4)
    else:
        x_points[i] = np.append(np.ones(1),np.random.uniform(-5,5,size = len(regressionParameters)))
        y_points[i]=np.random.normal(100,10, size=1)
plt.plot(x_points.T[1],y_points,'ro')
plt.show()

```

# Содержание

<b>1</b>	<b>Введение</b>	<b>1</b>
<b>2</b>	<b>Теоретические сведения</b>	<b>1</b>
2.1	Метод Наименьших Квадратов . . . . .	1
2.2	М-оценки . . . . .	1
2.3	L-оценки . . . . .	1
<b>3</b>	<b>Моделирование регрессии на языке Python</b>	<b>1</b>