Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского»**

**Институт Информационных технологий, математики и механики**

**Кафедра: программной инженерии**

Отчет по учебной практике:

Тема:

«Оценка параметров распределения потоков сложной структуры»

**Выполнил:** студент группы 381603-3

Кумин

Алексей Александрович

Подпись

**Научный руководитель:**

Ассистент

Евгений Владимирович

Кудрявцев

Профессор

Федоткин Михаил Андреевич

Подпись

Нижний Новгород

2020 г

Содержание

[1 Постановка задачи 3](#_Toc43444317)

[2 Описание и необходимая обработка данных 4](#_Toc43444318)

[3 Теоретические сведения 5](#_Toc43444319)

[3.1 Алгоритм нелокального описания потоков сложной структуры 5](#_Toc43444320)

[3.1.1 Нелокальный способ 1[2] 5](#_Toc43444321)

[3.1.2 Нелокальный способ 3[2] 5](#_Toc43444322)

[3.2 Проверка гипотез о независимости с помощью критерия Валлиса-Мура 6](#_Toc43444323)

[3.3 Случайные величины, оценка их параметров и проверка гипотез о распределении с помощью критерия «хи-квадрат» 7](#_Toc43444324)

[3.3.1 Смещенное Пуассоновское распределение 7](#_Toc43444325)

[3.3.2 Геометрическое распределение 7](#_Toc43444326)

[3.3.3 Смещенное показательное распределение 7](#_Toc43444327)

[3.3.4 Смесь распределений двух случайных величин 7](#_Toc43444328)

[3.3.5 Проверка гипотез о распределении с помощью критерия «хи-квадрат» 8](#_Toc43444329)

[4 Эксперименты и результаты 9](#_Toc43444330)

[4.1 Алгоритм нелокального описания 9](#_Toc43444331)

[4.2 Поиск распределений и оценка параметров 11](#_Toc43444332)

[4.2.1 Гипотезы для количества заявок в пачке требований 11](#_Toc43444333)

[4.2.2 Гипотеза для интервалов между пачками требований 13](#_Toc43444334)

[4.2.3 Разбиение выборки на две части и валидация параметров распределения 14](#_Toc43444335)

[5 Заключение 15](#_Toc43444336)

[6 Ссылки 16](#_Toc43444337)

[7 Реализация наиболее важных методов 17](#_Toc43444338)

[7.1 Добавление колонки дней в таблицу 17](#_Toc43444339)

[7.2 Разбиение данных на две выборки по времени и по количеству заявок в промежутках времени 17](#_Toc43444340)

[7.3 Разбиение данных на две выборки по времени и по количеству заявок в промежутках времени 18](#_Toc43444341)

[7.4 Оценка параметров распределения для выборки событий 19](#_Toc43444342)

[7.5 Критерий «хи-квадрат» для выборки событий 19](#_Toc43444343)

[7.6 Оценка параметров для промежутков времени между группами 20](#_Toc43444344)

[7.7 Критерий «хи-квадрат» промежутков времени между группами 20](#_Toc43444345)

[7.8 Отрисовка гистограмм и функций распределения 20](#_Toc43444346)

# Постановка задачи

В классической теории массового обслуживания рассматриваются только такие входные потоки, в которых случайные расстояния между заявками независимы и одинаково распределены. На практике нередко приходится сталкиваться с ситуацией, когда интервалы между заявками зависимы и имеют разное распределение. На это могут повлиять, например, погодные условия, катастрофы, политические события и т.д. Это не позволят сделать выводы о вероятностной структуре входного потока.

В данной работе требуется построить алгоритм определения временной и пространственной характеристики потока требований, на примере выборки террористических актов [1], с помощью нелокального описания входных потоков неоднородных требований [2].

# Описание и необходимая обработка данных

База данных террористических актов по всему миру [1] представляет собой таблицу из 181691 событий, имеющих 135 характеристик. Для данной работы требуется только 5 признаков – год, месяц, день, код страны, название страны (Рис. 1).

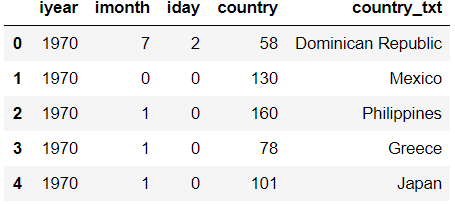
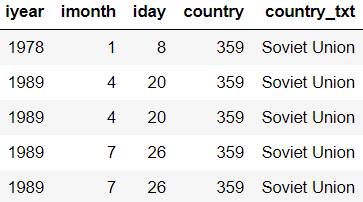


Рис. 1

Можно исследовать террористические акты по всем странам одновременно, а можно для каждый странны отдельно. В дальнейшем сравниваются данные для России и СССР с США (Рис. 2).



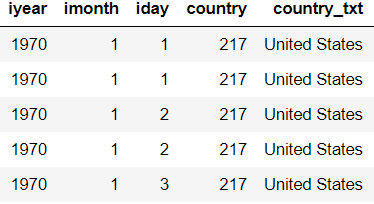


Рис.2

Примем первый акт как начальный, для каждого последующего события добавим количество дней со дня начала первого акта (Рис. 3) и получим выборку для потока требований определяемую добавленным столбцом.

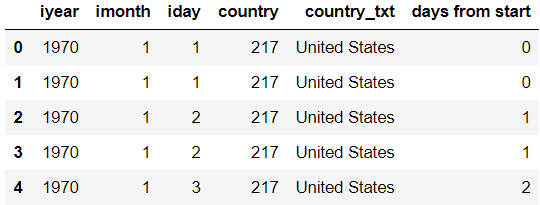


Рис. 3

# Теоретические сведения

## Алгоритм нелокального описания потоков сложной структуры

Входные потоки, как правило, описываются в виде случайной последовательности моментов поступления -ого требования или с помощью случайного процесса . Однако, случайные интервалы между последовательными заявками часто оказываются зависимыми и имеющими различные функции распределения, что не позволяет найти конечномерные распределения процесса.

### Нелокальный способ 1[2]

Для формализации потока применим нелокальный способ 1[2]. Преобразуем исходный поток отельных заявок в поток групп , где – число требований, поступивших на независимых интервалах согласно некоторому заданному принципу.

Зададим параметр близости , тогда величины будут определяться как

Реализация данного метода представлена в разделе 7.

### Нелокальный способ 3[2]

Поток делится на группы поэтапно, на этапе с номером получается векторная случайная последовательность . Параметры метода: натуральное число и постоянные величины . Рекуррентные формулы для определения моментов имеют вид:

Несложно заметить, что для формирования нулевого уровня используется первый способ разбиения, далее на каждом этапе редуцируется по одной пачке до тех пор, пока множества не окажутся пустыми. Данный способ предоставляет исследователю более детальное управление разбиением за счет большего числа параметров.

## Проверка гипотез о независимости с помощью критерия Валлиса-Мура

Применяя указанный выше метод, можно разбить исходный поток на группы так, что случайные последовательности и будут составлены из независимых и одинаково распределенных случайных величин. Для этого применим фазово-частотный критерий Валлиса-Мура [2]. Далее описан алгоритм в общем случае, будем говорить о повторной выборке объёма и наблюдаемых значениях .

Критерий состоит в определении статистики , считающей случайное число фаз. Для всех ненулевых значений выпишем последовательно полученные знаки. Фаза – перемена знака при последовательном обходе массива выписанных знаков. Посчитав общее число перемен знаков, получим и построим статистику

Если выдвинутая гипотеза о независимости и одинаковом распределении случайных величин верна, то последовательность сходится к стандартному нормальному закону. Пороговое значение определяется при заданном уровне значимости из равенства где функция распределения стандартной нормальной случайной величины. Гипотеза отвергается если ,

## Случайные величины, оценка их параметров и проверка гипотез о распределении с помощью критерия «хи-квадрат»

### Смещенное Пуассоновское распределение

В общем случае, – дискретная случайная величина имеет смещенное распределение Пуассона, если

Для нашего случая потребуем, чтобы , тогда

Соответственно, параметр можно оценить с помощью метода моментов:

### Геометрическое распределение

– дискретная случайная величина имеет геометрическое распределение, если:

Оценка методом моментов для параметрa:

### Смещенное показательное распределение

– непрерывная случайная величина имеет смещенное показательное распределение, если:

Оценка методом моментов для параметров:

### Смесь распределений двух случайных величин

– дискретная случайная величина имеет смешанное распределение двух случайных величин, если:

Из-за сложности нахождения параметров распределения с помощью критерия максимального правдоподобия и неточной оценки методом моментов (приходится вычислять моменты 3 степени и отклонения в выборке могут давать большую погрешность между истинными и вычисляемыми моментами) будем оценивать параметры следующим образом.

Необходимо, во-первых, отсортировать и откинуть несколько последних значений отсортированной выборки, которые могут являться отклонениями. Во-вторых, нужно разделить выборку на две части. Вычислим параметры первого распределения с помощью первой выборки, а второго соответственно с помощью второй методом моментов (. Решая уравнение получим оценку для параметра смеси распределений. Количество отбрасываемых значений и параметр разбиения выборки задается пользователем.

Так ,например, для смеси геометрического и смещенного пуассоновского распределений будет справедлива формула:

### Проверка гипотез о распределении с помощью критерия «хи-квадрат»

Для дискретных случайных величин критерий будет выглядеть следующим образом: требуется разбить выборку на частей так, чтобы вероятности, что значение случайной величины принадлежит части , были примерно равны между собой и примерно равны величине . Далее сформируем из выборки массив количество элементов выборки, попавшее в часть . Необходимо посчитать значение статистики

Значение статистики при имеет распределение с степенями свободы, где количество оцениваемых параметров. Следовательно, если получить значение статистики, не превосходящее некоторого порогового значения величины с степенями свободы, то исследуемые статистические данные можно считать совместимыми с гипотезой о распределении.

В случае непрерывной случайной величины необходимо разбить выборку на промежутки так, чтобы вероятности попадания в них были примерно равны между собой, посчитать количество элементов выборки попавших в каждый промежуток, а далее следовать алгоритму, описанному для дискретного случая.

# Эксперименты и результаты

## Алгоритм нелокального описания

Сначала была произведена работа над разбиением исходного потока на поток групп при помощи нелокального метода для России и США, и с помощью критерия Валлиса-Мура проверена гипотеза о независимости.

Нелокальный метод разбиения показал хорошие результаты: для России c параметром , и уровне значимости для критерия Валлиса-Мура были сформированы выборки и проверены гипотезы о независимости и одинаковом распределении (Рис. 4).

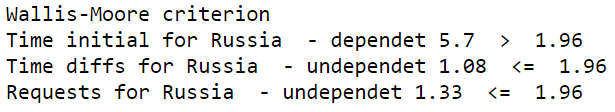
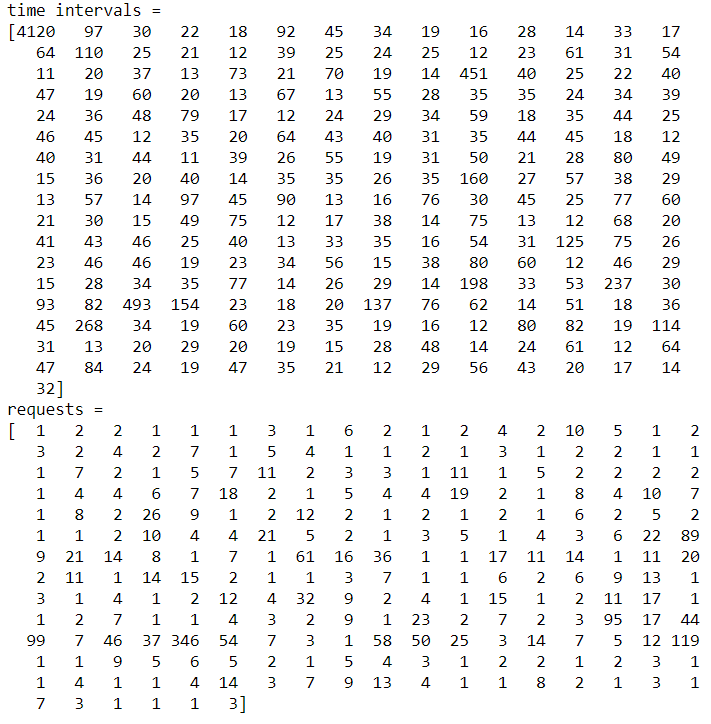
**

Рис. 4

Для США с параметром результаты показаны на Рис. 5

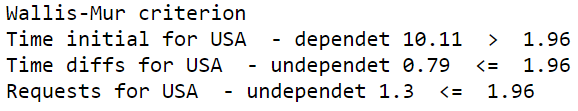
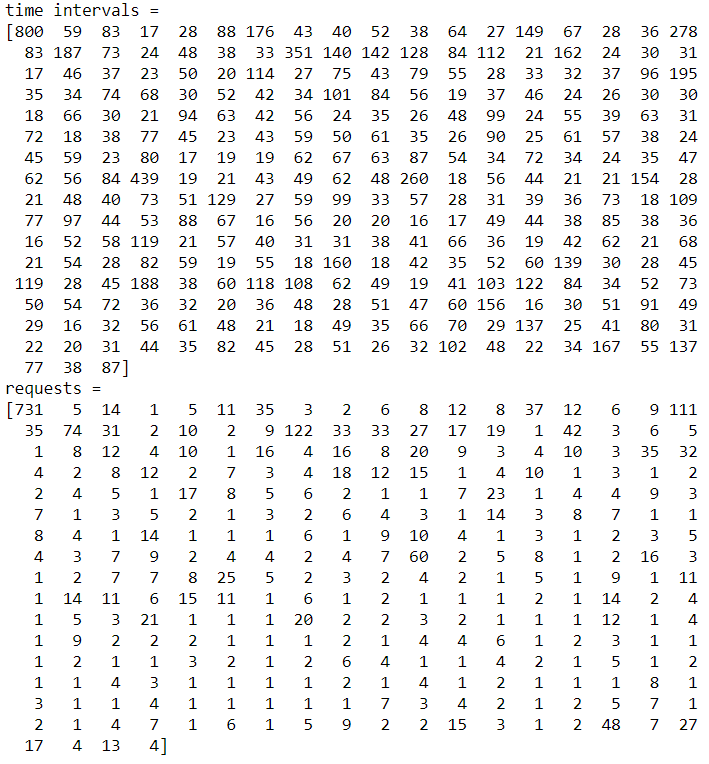


Рис. 5

Заметим, что для шага в случае выборки США, критерий Валлиса-Мура отвергает гипотезу о независимых распределениях (Рис. 6)

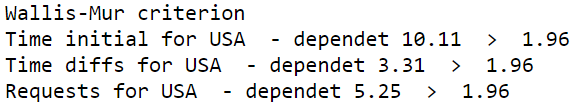


Рис.6

## Поиск распределений и оценка параметров

### Гипотезы для количества заявок в пачке требований

Первый эксперимент заключался в предположении, что распределение количества требований в пачке подчиняется смеси пуассоновских распределений. Дело в том, что просто геометрическое или пуассоновское распределение плохо описывает поток количества требований из-за наличия выбросов в выборке и есть два варианта: рассмотреть смесь распределений и отбросить выбросы. Будем придерживаться первого варианта, так как количество выбросов в выборке в среднем довольно большое относительно размера выборки(80-90%). Для оценки параметров необходимо разделить выборку на две части как было описано ранее. Для данных России с параметром разделения 10, количеством отбрасываемых значений из выборки равным 2 и параметром разбиения для критерия «хи-квадрат» имеем результаты, представленные на Рис. 7

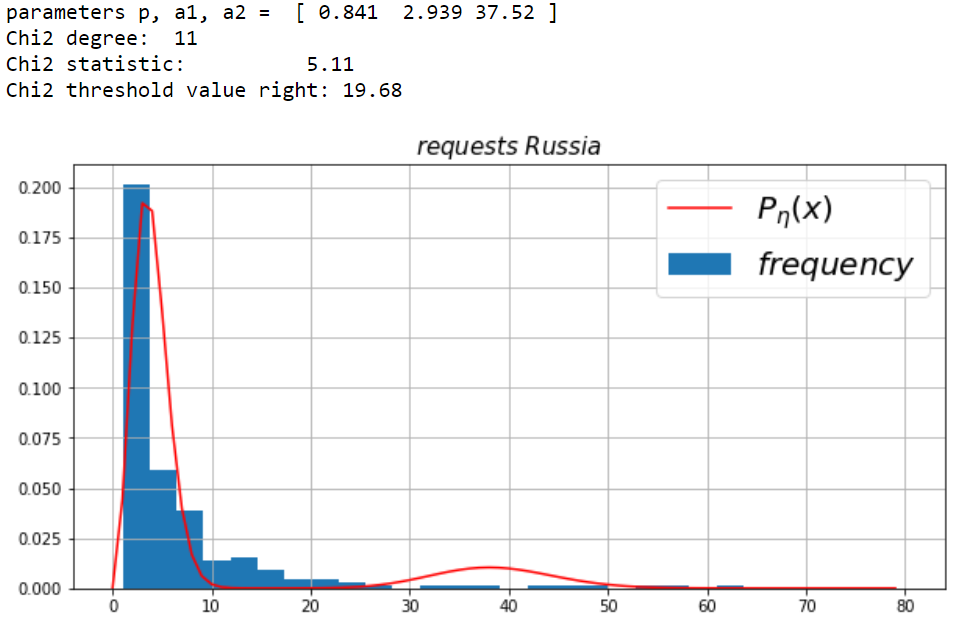


Рис. 7

Из графиков видно, что данное распределение не очень хорошо описывает исследуемую случайную величину, поэтому выдвинем другую гипотезу. Пусть она подчиняется смеси геометрического и Пуассоновского распределений, с параметром разбиения 30. Тогда получим следующие результаты (Рис. 8)

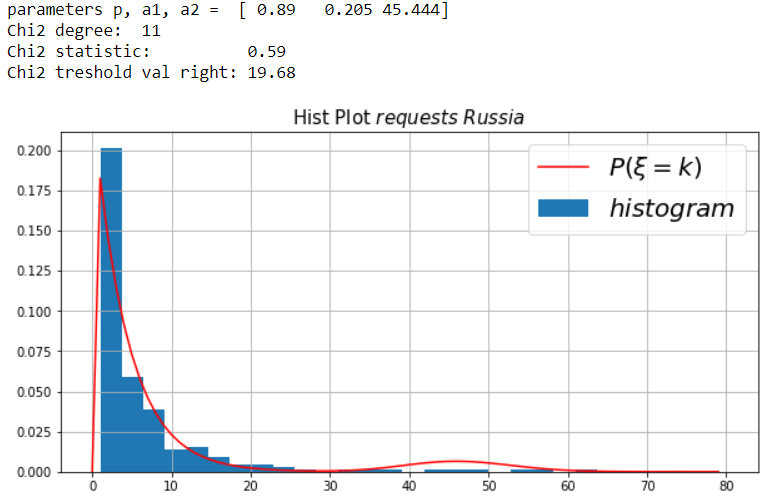


Рис. 8

Видно, что данное распределение наиболее точно описывает исследуемую случайную величину.

Рассмотрим теперь случай выборки для США. Выдвинем гипотезу о смеси геометрического и пуассоновских распределений с параметром разделения 30 без удаления элементов для данной выборки. Результаты представлены на Рис. 9.

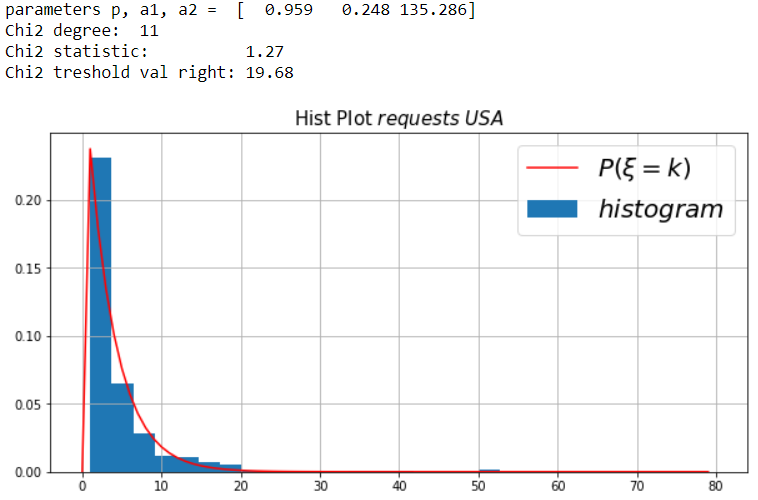
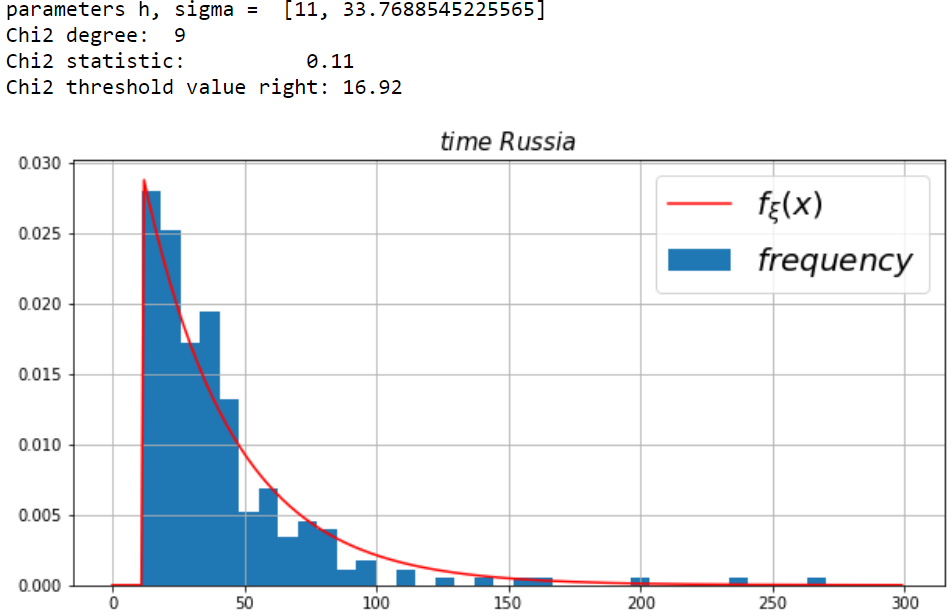


Рис. 9

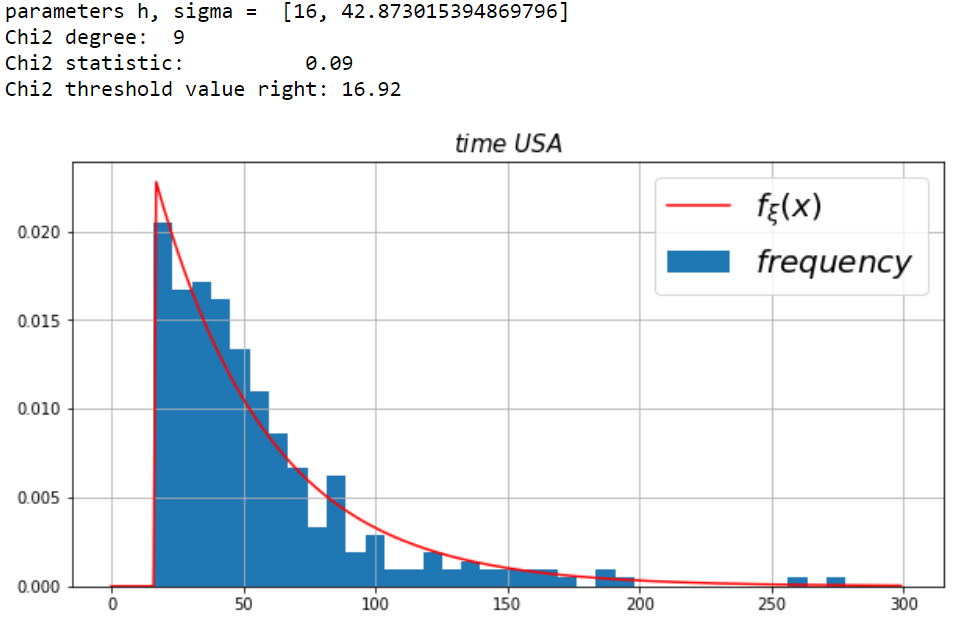
Можно заметить, что для России и США случайные величины количества заявок в пачке требований очень похожи друг на друга и геометрическое распределение преобладает в смеси.

### Гипотеза для интервалов между пачками требований

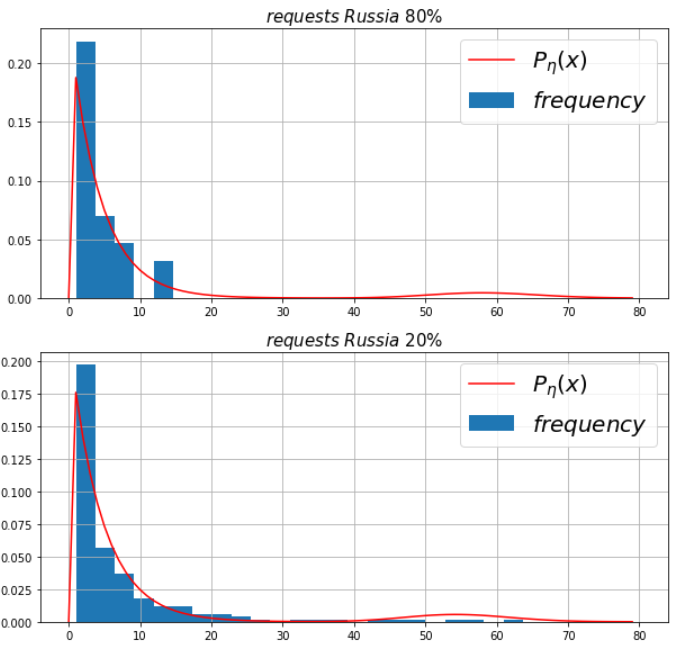
Будем предполагать, что интервалы времени между пачками требований подчиняются смещенному показательному распределению, описанному ранее, и оценим параметр минимальным значением из выборки и, для более точной оценки другого параметра с помощью математического ожидания, удалим из отсортированной выборки несколько последний значений. Тогда после удаления 2х элементов для выборки России имеем следующий результат:



Для выборки США при аналогичных условиях имеем следующий результат:



### Разбиение выборки на две части и валидация параметров распределения

Проведем эксперимент: разделим выборку на две части – 80% и 20%, подберем параметры на большей выборке и примем их в качестве параметров для меньшей части выборки. Для данных России был получен следующий результат:  


Параметры подходят для меньшей выборки, что говорит о том, что количество требований в пачке не зависит от времени прихода пачки.

# Заключение

С помощью алгоритма нелокального описания сложного случайного процесса, удалось исследовать и выдвинуть гипотезы о распределении для потока случайных событий, а именно, террористических актов, происходящих по всему миру. Установлено, что поток террористических актов имеет сложную структуру и не может быть описан конечномерными распределениями. Зависимости между происшествиями объясняются воздействием политических событий, природных катастроф, научно-техническим прогрессом и др.

Нелокальное описание случайного потока террористических событий, представляющее собой последовательность групп актов, идущих друг за другом через измеряемые промежутки времени, позволило описать процесс с помощью вероятностных распределений. Для России и США установлено, что количество актов в группе подчиняется в большей степени геометрическому распределению, а промежутки времени между группами – смещенному показательному распределению.

# Ссылки

1. <https://www.kaggle.com/START-UMD/gtd>
2. Кудрявцев Е.В., Федоткин М.А «Изучение характеристик транспортного потока большой плотности», 2013
3. Гнеденко Б.В., Коваленко И.Н. «Введение в теорию массового обслуживания» М.: Наука, 1966. — 432 с.
4. Федоткин М.А. «Модели в теории вероятностей» ФИЗМАТЛИТ 2012. – 608 с.

# Реализация наиболее важных методов

Программная среда для исследований была реализована на языке Python с помощью общеизвестных библиотек numpy, mathplotlib, pandas, scipy, datetime.

## Добавление колонки дней в таблицу

def compute\_and\_add\_days(df, name):

df = df.reset\_index(drop = True)

n\_cols = df.shape[1]

n\_rows = df.shape[0]

ret = df;

uncorrects = [[] , 0];

row\_days = []

row0 = df.loc[0];

start = date(row0[0], row0[1], row0[2])

for i in range(0, n\_rows):

row = ret.loc[i]

if ((row[2] < 1) or (row[2] > 31)): #or (row[1] < 1) or (row[1] > 12)):

ret = ret.drop(i, axis = 0)

continue

curr = date(row[0], row[1], row[2])

row\_days.append((curr - start).days)

ret[name] = row\_days

ret = ret.reset\_index(drop = True)

ret = ret.sort\_values(by=[name])

ret = ret.reset\_index(drop=True)

return ret

## Разбиение данных на две выборки по времени и по количеству заявок в промежутках времени

def first\_method\_reduce(df, step):

n\_cols = df.shape[1]

n\_rows = df.shape[0]

ret = []

i = 0

while (i < (n\_rows - 1)):

row = [df.loc[i][n\_cols - 1], i]

while ((df.loc[i + 1][n\_cols - 1] - df.loc[i][n\_cols - 1] <= step) and (i < (n\_rows - 2))):

i += 1

i += 1

ret.append([row[0], i - row[1]])

ret[-1][1] += 1

ret = np.array(ret)

return (ret)

def third\_method\_reduce(df, step\_params):

step\_0 = step\_params[0]

step\_1 = step\_params[1]

step\_2 = step\_params[2]

d\_step = step\_params[3]

df = first\_method\_reduce(df, step\_0)

ret = []

i = 0

s = 0

k = 0

while (s != -1):

s = -1

n\_rows = len(df)

s\_tmp1 = 1000000

s\_tmp2 = 1000000

i = 0

while (i < n\_rows-2):

time\_req\_1 = df[i]

time\_req\_2 = df[i + 1]

if (time\_req\_1[1] <= d\_step and time\_req\_2[1] == d\_step+1 and time\_req\_2[0] - time\_req\_1[0] < step\_1):

s\_tmp1 = i

break

i += 1

i = 0

while (i < n\_rows-2):

time\_req\_1 = df[i]

time\_req\_2 = df[i + 1]

if (time\_req\_1[1] <= d\_step and time\_req\_2[1] <= d\_step and time\_req\_2[0] - time\_req\_1[0] < step\_2):

s\_tmp2 = i

break

i += 1

s = min(s\_tmp1, s\_tmp2)

if (s > 0 and s < 1000000):

i = 0

tmp\_df = []

while (i <= s):

tmp\_df.append(df[i])

i+=1

if (i == s + 1):

tmp\_df[s][0] = df[i][0]

tmp\_df[s][1] += df[i][1]

while (i < n\_rows-1):

tmp\_df.append(df[i + 1])

i+=1

df = np.array(tmp\_df)

else:

break

return (df)

def make\_reduce(sample, method, step, country):

reduce = method(sample, step)

print("Reduce data for", country, "with days from start, shape: " , len(reduce))

time\_smp = np.diff(reduce[:,0])

requests\_smp = reduce[:, 1]

print("time intervals = ")

print(time\_smp)

print("requests = ")

print(requests\_smp)

return time\_smp, requests\_smp

## Разбиение данных на две выборки по времени и по количеству заявок в промежутках времени

def num\_sign\_changes(arr):

i = 0

sign = arr[0]

count = 0

while i < len(arr) - 1:

if arr[i] != 0:

if sign == 0:

sign = arr[i]

elif sign != arr[i]:

count += 1

sign = arr[i]

i +=1

return count

def Wallis\_Murr\_number(sample):

sample\_diff = np.diff(sample)

gamma = num\_sign\_changes(np.sign(sample\_diff)) - 2

n = len(sample)

return np.abs((gamma - (2\*n - 7)/3)\*np.sqrt(90/ (16\*n - 29)))

def Wallis\_Murr\_crit(sample, alfa, str):

Z1 = Wallis\_Murr\_number(sample)

f = norm(0, 1)

treshold\_val = f.ppf(alfa/2)

if (Z1 <= -treshold\_val):

print(str, "is undependet", round(Z1, 2), " <= ", -round(treshold\_val,2))

else:

print(str, "is dependet", round(Z1, 2), " > ", -round(treshold\_val, 2))

## Оценка параметров распределения для выборки событий

def GMM\_Poisson(sample):

a1 = np.average(sample)

return a1 - 1

def GMM\_geom(sample):

a1 = np.average(sample)

return 1/a1

def GMM\_mix\_distrib(sample, a2, a3):

a1 = np.average(sample)

equations = lambda x: (x\*a2 + (1 - x)\*a3 - a1)

return fsolve(equations, 0.9)

def process\_sample(sample, delimiter, missed\_vals):

sample1 = (np.sort(sample))[0:len(sample) - missed\_vals]

ret1 = sample1[sample1 <= delimiter]

ret2 = sample1[sample1 > delimiter]

return ret1, ret2

## Критерий «хи-квадрат» для выборки событий

def func\_sum(inter, func):

sum = 0

for i in np.arange(inter[0], inter[1] + 1):

sum += func(i)

return sum

def divide\_time(time, r, func):

tmp = np.bincount(time)

interv = len(time)//r

intervp = 1/r

m = np.zeros(r)

p = np.zeros(r)

j = 0

for i in range(0, r):

if (j < len(tmp)):

m[i] += tmp[j]

p[i] += func(j)

j += 1

while (p[i] < intervp) and (j < len(tmp)):

intervp = (1-sum(p))/(r - i)

m[i] += tmp[j]

p[i] += func(j)

j += 1

p[len(p) - 1] = 1 - sum(p[0:len(p) - 1])

return m, p

def Chi\_squre\_number(m, p):

ret = 0

r = len(m)

n = np.sum(m)

for i in range(0, r):

ret += (m[i] - n\*p[i])\*\*2 / n\*p[i]

return ret

def Chi\_2\_crit(sample, r, P, alfa, num\_param):

m, p = divide\_time(sample, r, P)

chi\_stat = Chi\_squre\_number(m, p)

chi\_2 = chi2(r - 1 - num\_param)

print("Chi2 degree: ", r - 1 - num\_param)

chi\_2\_tres1 = chi\_2.ppf(1 - alfa)

print("Chi2 statistic: ", round(chi\_stat, 2))

print("Chi2 treshold val right:", round(chi\_2\_tres1, 2))

## Оценка параметров для промежутков времени между группами

def GMM\_shift\_exp(sample, missed\_vals):

tmp = np.sort(time\_arr1)[0:len(time\_arr1)-missed\_vals]

a1 = np.average(tmp)

a2 = np.std(tmp)

return [ a1 - a2, a2]

## Критерий «хи-квадрат» промежутков времени между группами

def divide\_time1(time, r, func):

tmp1 = np.bincount(time)

interv = len(time)//r

intervp = 1/r

m = np.zeros(r)

p = np.zeros(r + 1)

bounds = np.zeros(r + 1)

j = 0

h = 0.25

interval\_sum = 0

for i in range(1, r + 1):

while (p[i] < intervp) and (func(interval\_sum) < func(max(time))):

interval\_sum +=h

p[i] = func(interval\_sum) - func(bounds[i - 1])

bounds[i] = interval\_sum

tmp = time[time < bounds[i]]

m[i - 1] = len(tmp[tmp >= bounds[i - 1]])

p[r] = 1 -sum(p[:r])

p = p[1:]

return m, p

def Chi\_2\_crit1(sample, r, P, alfa, num\_param):

m, p = divide\_time1(sample, r, P)

chi\_stat = Chi\_squre\_number(m, p)

chi\_2 = chi2(r - 1 - num\_param)

print("Chi2 degree: ", r - 1 - num\_param)

chi\_2\_tres1 = chi\_2.ppf(1 - alfa)

print("Chi2 statistic: ", round(chi\_stat, 2))

print("Chi2 treshold val right:", round(chi\_2\_tres1, 2))

## Отрисовка гистограмм и функций распределения

def plot\_hist(sample, r\_bound, P, ax, title, n=0):

x = np.arange(0, r\_bound)

Fx = []

for i in x:

Fx.append(P(i))

if (n == 0):

n = round(np.math.sqrt(len(sample)))

bins = np.linspace(sample.min(), r\_bound, n)

ax.hist(sample, bins, density=1)

ax.plot(x, Fx, color="red")

ax.legend([r'$f\_\xi(x)$', r'$histogram$'], fontsize=20)

ax.set\_title("Hist Plot " + title , fontsize = 15)

ax.grid()