1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт кибербезопасности и защиты информации

**К У Р С О В А Я Р А Б О Т А**

1. «**Распределение Коши**»
2. по дисциплине «Теория вероятностей и математическая статистика»
3. Выполнил
4. студент гр. 4851004/10001 Гуторова Л.С.

<*подпись*>

1. Проверил:
2. преподаватель Лаврова Д.С.

<*подпись*>

1. Санкт-Петербург
2. 2023

# ЦЕЛЬ РАБОТЫ

Цель данной курсовой работы заключается в изучении и анализе распределения Коши, его свойств и особенностей, а также в применении данного распределения на заданной выборке.

Для выполнения работы необходимо использовать математический аппарат теории вероятностей и математической статистики, а также программные средства для моделирования и анализа данных. Jupyter Notebook используется для обработки заданной выборки.

# ПОСТАНОВКА ЗАДАЧ

1. Провести исследование распределения из варианта: история его появления, его назначение, применение в научных исследованиях;
2. Знакомство с jupyter notebook;
3. Понятие эмпирической функции распределения;
4. Понятие гистограммы;
5. Описание параметров распределения;
6. Понятие точечных оценок;
7. Понятие интегральный оценок;
8. Понятие статистических критериев.

# ОПисание решения

## Исследование распределения Коши

### О распределении

Распределение Коши в теории вероятностей (также называемое в физике распределением Лоренца и распределением Брейта – Вигнера) – класс абсолютно непрерывных распределений. Случайная величина, имеющая распределение Коши, является стандартным примером величины, не имеющей математического ожидания и дисперсии, о чем свидетельствует [8].

### Об истории появления

Распределение Коши было введено французским математиком Огюстеном Луи Коши в 1827 году. Он исследовал распределение, которое возникает при делении двух нормально распределенных случайных величин, и обнаружил, что оно имеет тяжелые хвосты и не имеет математического ожидания и дисперсии.

Сначала распределение было названо "распределением Лапласа", в честь французского математика Пьера-Симона Лапласа, который также исследовал это распределение. Однако позже оно было переименовано в "распределение Коши" в честь его создателя.

### Применение в научных исследованиях

Распределение Коши является одним из распределений вероятности, которое часто применяется в научных исследованиях. Оно используется в различных областях, таких как физика, экономика, биология, медицина и т.д.

Одним из примеров применения распределения Коши является его использование в статистической физике для описания распределения скоростей молекул в газах. Также распределение Коши может быть использовано для анализа экспериментальных данных в биологии, например, для анализа распределения размеров клеток.

В экономике распределение Коши может быть использовано для анализа распределения доходов населения или цен на товары и услуги. В медицине распределение Коши может быть использовано для анализа распределения длительности жизни пациентов или времени ожидания на прием к врачу.

Также распределение Коши может быть использовано в математической статистике для оценки параметров других распределений вероятности, например, нормального или экспоненциального распределений.

Таким образом, распределение Коши является важным инструментом для анализа данных в различных научных исследованиях и может быть использовано для моделирования различных явлений в природе и обществе.

## Знакомство с jupyter notebook

Для реализации данного пункта была настроена среда интерактивной разработки jupyter notebook. Код разработанной программы представлен в приложении А.

### Считывание из файла

Функция scan\_file выполняет считывание значений из файла в список list\_elem. Далее, двигаясь по файлу, записывается значение без пробелов в переменную num. После эта переменная заносится в конец списка.

### Сумма элементов выборки

Сумма элементов выборки подсчитывается с помощью функции sum. Эта функция пробегается по списку list\_elem и значения суммируются в переменную num. Далее функция возвращает эту сумму.

### Выборочное среднее

Выборочное среднее подсчитывается с помощью функции sample\_mean(). Сначала подсчитывается сумма всех значение из пункта 3.2.2, а потом эта сумма делится на количество элементов выборки. Количество элементов выборки подсчитывается с помощью функции get\_len().

### Медиана

Для начала выборка сортируется. Далее если количество элементов нечетное число, то медиана – это значение с номером n/2. Иначе среднее арифметическое значение элементов с номером n//2 и n//2-1.

### Мода

Подсчитывается количество каждых уникальных значений. Далее находим максимум и таким образом получаем моду. Если все значения равны 1, то значение моды не определено.

### Размах выборки

Изначально были подсчитаны максимальное и минимальное значения выборки. В функции находится разница между этими значениями.

### Смещенная дисперсия

Вычисляется следующая сумма значений: из значения вычитается выборочное среднее, подсчитанное заранее. Далее это значение возводится в квадрат. После эта сумма делится на количество элементов.

То есть выполняется формула: .

### Несмещенная дисперсия

Аналогично предыдущему пункту, только деление выполняется с делителем n-1.

То есть выполняется формула: .

### Выборочный начальный момент k-ого порядка

k=3. Начальный момент считался по похожей формуле, как смещенная дисперсия. Однако между квадрата в формуле стоит степень k.

### Значения

После работы программы были получены следующие значения:

Сумма выборки равна -5122.298736999999

Среднее выборчное равно -17.07432912333333

Медиана равна -18.1089855

Мода равна None

Размах выборки равен -469.90322199999997

Смещенная дисперсия равна 565.533110843405

Несмещенная дисперсия равна 567.4245259298378

Выборочный начальный момент 3-ого порядка равен 192315.21920248197

Выборочный центральный момент 3-ого порядка равен 192315.21920248197

Мода равна None, так как она не определена. Значения ни разу не повторились.

## Понятие эмпирической функции распределения

### Определение эмпирической функции

Пусть Функция – функция распределения случайно величины . Другими словами, функцией распределения случайной величины называется, определяющая для каждого наперед заданного значения вероятность того, что случайная величина примет значение меньшее, чем .

Для распределения Коши функция распределения имеет следующий вид: .

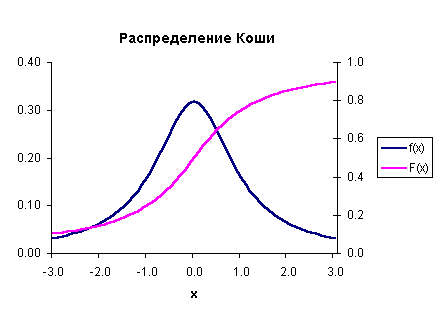


Рисунок 1 – Теоретические графики функций распределения Коши

Эмпирической функцией распределения называют такую функцию, которая определяет для каждого значения частоту событий и предназначена для оценки теоретической функции распределения генеральной совокупности в математической статистике.

Итак, для выборки из генеральной совокупности с теоретической функцией распределения эмпирическая функция распределения – , где число элементов выборки , меньших .

Был построен график эмпирической функции с помощью jupyter notebook.

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Рисунок 2 – График эмпирической функции, полученной по данной выборке

А также графики эмпирической функции для случайных выборок из данной выборки.

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок 3 – График эмпирической функции, полученной по подвыборкам

### Вывод

Видно, что графики теоретической и практической эмпирической функции совпадают. Если брать подвыборки данной выборки, то можно заметить, что чем меньше элементов было взято, тем хуже график отображает распределение Коши. График, где 200 элементов практически совпадает с теоретическим графиком.

Можно заметить, что у данной выборки есть смещение. Этот параметр несложно заметить, обратив внимание, что резкий рост функции возникает в районе числа -18, а не в районе 0, как это происходит у теоретической функции.

## Понятие гистограммы

### Гистограмма

Гистограмма распределения Коши является графическим представлением выборки данных из этого распределения. Она строится путем разбиения диапазона значений на интервалы и подсчета количества значений выборки, попадающих в каждый интервал.

Если функция непрерывна и имеет производную на всем промежутке изменения случайной величины, то называется функцией плотности распределения случайной величины.

Для распределения Коши функция плотности распределения имеет следующий вид: , где – коэффициент масштаба, – коэффициент сдвига.

Гистограмма – это кусочно-постоянная функция , принимающая на интервале значение , где  — число элементов выборки, попавших в , , ; .

### Алгоритм построения гистограммы

Чтобы построить гистограмму для распределения Коши нужно:

1. Определить интервалы гистограммы. Для распределения Коши интервалы должны быть симметричны относительно медианы.
2. Разделить интервалы на равные части, называемые ячейками.
3. Подсчитать количество значений выборки, попавших в каждую ячейку.
4. Отобразить результаты на гистограмме, где по оси X отложены интервалы, а по оси Y – количество значений, попавших в каждый интервал.
5. Нормировать гистограмму, чтобы ее площадь равнялась единице. Для этого нужно разделить каждое значение на общее количество значений выборки, умноженное на ширину ячейки.
6. Построить график плотности вероятности распределения Коши на том же графике, чтобы сравнить его с гистограммой.

### Функция для построения гистограммы

Функция hist() выполняет определение параметров для построения гистограммы. Для начала выборка сортируется. После выбирается максимальное и минимальное значение из выборки. Для начала определяется ширина столбца. Для этого из максимального значения вычитается минимальное значение и делится на количество столбцов. Далее через цикл проверяем входят ли значение в текущий промежуток.

Далее нормируем значения. Для этого делим на количество элементов, умноженное на ширину столбца. Через функцию linspace() возвращают промежуток значений, то есть значения y.

### Построение гистограммы

Для тех же выборок, которые были выделены для эмпирической функции, были построены гистограммы.

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок 4 – Гистограмма распределения Коши с 10 элементами выборки

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

Рисунок 5 – Гистограмма распределения Коши с 100 элементами выборки

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

Рисунок 6 – Гистограмма распределения Коши с 200 элементами выборки

### Вывод

Полученная гистограмма очень похожа на теоретическую функции плотности. При маленьком количестве элементов выборки гистограмма очень плохо отражает функцию плотности, но при большом количества эта гистограмма достаточно сильно совпадает с теоретической функции.

Видно, что при данной выборке есть параметр смещения. Примерно график смещен на -17.

## Описание параметров распределения

### Функция распределения

Для распределения Коши функция распределения имеет следующий вид: , где – коэффициент масштаба, – коэффициент сдвига.

В варианте используется только один параметр, поэтому возьмем , поэтому функция имеет вид: .

### Параметр смещения

Параметр смещения (loc) в распределении Коши определяет точку, в которой находится пик распределения. Он является медианой распределения и определяет центральную точку симметрии. При изменении значения параметра смещения форма распределения не меняется, но сдвигается вправо или влево относительно оси x.

### Функции распределение

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 7 – Функция распределения при случайном смещении

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 8 – Функция распределения при случайном смещении

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Рисунок 9 – Функция распределения при случайном смещении

### Вывод

Из полученных графиков можно сделать вывод, что параметр loc(смещение) отвечает за смещение функции распределения вдоль оси x.

Для данной выборки смещение очень влияет на график функции. При смещении в промежутке [-20, -15] получается корректный график функции распределения. При остальных величинах смещения видно, что график не похож на теоретический.

## Понятие точечных оценок

### Метод моментов

Выборочный момент порядка 𝑘 является состоятельной оценкой теоретического момента. Данный факт лежит в основе метода моментов. Момент порядка 𝑘 – математическое ожидание случайной величины в степени k:

### Метод максимального правдоподобия

Другим наиболее распространенным методов нахождения оценок является метод максимального правдоподобия. Пусть теоретическая функция распределения зависит от параметра 𝜃. С помощью вероятности или плотности составим функцию правдоподобия Если случайные величины (элементы выборки) дискретные, то функция правдоподобия представляет собой произведение вероятностей, а если величины (элементы выборки) непрерывные – произведение плотностей.

Таким образом, функция правдоподобия – вероятность (для дискретного случая) или плотность (для непрерывного случая) совместного распределения случайных величин , если бы значение неизвестного параметра действительно равнялось 𝜃. В качестве оценки 𝜃 неизвестного параметра следует выбрать то значение, которое дало бы максимум функции правдоподобия. Оценка метода максимального правдоподобия 𝜃̂ – то значение, которое обеспечивает максимум функции правдоподобия.

### Вывод по методам

Известно, что распределение Коши не имеет ни математического ожидания, ни дисперсии. Поэтому для оценки его параметров приходится модифицировать излагаемые в учебниках и различных пособиях методы. Нахождение оценок параметров loc, исходя из множества независимых измерений (𝑥1, 𝑥2, … , 𝑥𝑛 ) представляет некоторые трудности. Это объясняется тем, что стандартные методы получения оценок – метод максимального правдоподобия и метод моментов почти не работают.

Математическое ожидание распределения Коши не существует, так как расходится интеграл

Изображение выглядит как черный

Автоматически созданное описание

Рисунок 10 – Математическое ожидание распределения Коши

Расходится он из-за того, что подынтегральная функция ведёт себя на бесконечности как . Поэтому не существуют ни дисперсия, ни моменты более высоких порядков этого распределения. Для распределения Коши моменты целых порядков не существуют (точнее говоря, они равны ∞).

Согласно [8], Михаил Леонидович Шинкеев обошёл эту трудность, рассматриваяпри 0 < k < 1, где математическое ожидание существует и равно . Для *λ* оценка получается в следующем виде:

Возьмём , подсчитаем *Mk* и получим искомую оценку параметра сдвига: .

Метод максимального правдоподобия для распределения Коши даёт следующее алгебраическое уравнение: , которое не решается в явной форме относительно параметра x0, поэтому найти точное значение оценки параметра сдвига не представляется возможным. Однако, всё ещё можно найти значение параметра сдвига с помощью данного метода для заданной точности. Для нахождения значения этого параметра с точностью до 4 знаков после запятой была написана функция на языке программирования Python, которая не дала результата.

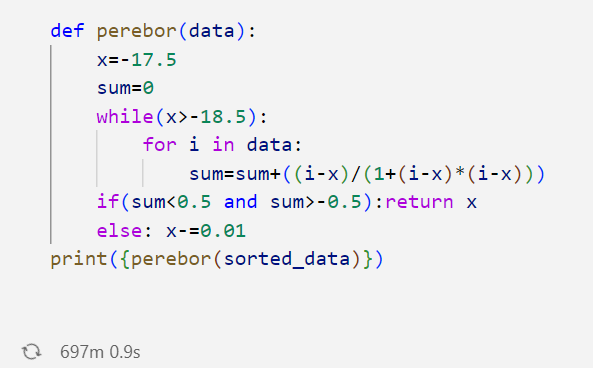


Рисунок 11 – Перебор не дал результата

Для распределения Коши существуют другие методы для оценка параметра смещения. Далее будут рассмотрены некоторые из них.

### Оценка усеченного среднего

Усеченное среднее является полезным средством оценки, поскольку оно менее чувствительно к выбросам. Чтобы составить данную статистику требуется отбросить наименьшие и наибольшие значения, а далее найти среднее выборочное.

Была составлена функция, которая подсчитывает усеченное среднее: Усеченное среднее равно -17.94381544999999 при отбрасывании 6 элементов

Усеченное среднее равно -18.07860151333332 при отбрасывании 4 элементов

Усеченное среднее равно -17.968541209999987 при отбрасывании 2 элементов

### Медиана

Иногда для распределения Коши смещение можно оценить как медиану. Медиана равна -18.1089855.

### Свойства медианы

Медиана как оценка является несмещенной, состоятельной и не эффективной.

### Графики

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок 12 – Сравнение полученных теоретических функций с эмпирической функцией

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

Рисунок 13 – Сравнение полученных теоретических функций с эмпирической функцией

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

Рисунок 14 – Сравнение полученных теоретических функций с эмпирической функцией

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок 15 – Сравнение полученных теоретических функций с эмпирической функцией

### Вывод

Оценка смещения, полученная с помощью усечения 2 и 6 элементов, сильно отличается от эмпирической. Однако при отсечении 4 элементов значение становится ближе к медиане, и большая часть графики совпадают. Как видно, в эмпирическом графике есть много точек, которые смещены относительно теоретических функций, однако данные оценки дают схожие результаты и стремятся к эмпирической.

## Понятие интервальных оценок

### Определение интервальных оценок

Интервальные оценки позволяют оценить диапазон возможных значений параметров с определенной вероятностью. Для оценки параметров выборки с помощью интервальных оценок можно использовать методы, основанные на теории вероятностей и статистике.

Важно учитывать, что интервальные оценки дают только приблизительные значения параметров выборки, и точность этих оценок зависит от размера выборки и уровня доверия, выбранного для построения интервала.

Требуется оценить параметр смещения распределения выборки с помощью интервальной оценки с уровнями доверия , .

Так как известно значение исправленной выборочной дисперсии, то можно воспользоваться формулой для вычисления интервальной оценки параметра смещения:

### Расчет значений

Для уровней доверия , по таблице значений функции Лапласа определим и :

Учитывания, что , получим:

Для точности подсчета интервала была создана подпрограмма на языке python. Она представлена в приложении А.

Левая часть интервала при уровне доверия 0,95 равна: -19.769892180662954

Правая часть интервала при уровне доверия 0,95 равна: -14.37876606600371

Левая часть интервала при уровне доверия 0,99 равна: -20.622570290634567

Правая часть интервала при уровне доверия 0,99 равна:

-13.526087956032095

Интервальная оценка параметра смещения для уровня доверия

Интервальная оценка параметра смещения для уровня доверия

Соотнесем полученные интервальные и точечные оценки. В ходе выполнения предыдущего пункта была получены следующие оценки параметра *loc*:

Метод моментов:

Медиана:

Усеченное среднее:

Можно заметить, что данные значения действительно попадают на оба полученных выше интервала для уровней доверия и .

### Вывод

Была произведена интервальная оценка смещения распределения Коши. С помощью разработанной программы удалось найти значения для концов интервала для уровней доверия, равным 0,95 и 0,99. Таким образом, полученные в пункте 3.6 значения смещения входят в данные интервалы. Это говорит о том, что составленные интервалы достаточно корректны.

## Понятие статистических критериев

### Гипотезы о параметрах распределениях.

Для проверки нулевой гипотезы о равенстве смещения распределения Коши заданной оценке ( ) можно использовать критерий отношения правдоподобия.

Для этого необходимо сравнить значения логарифма правдоподобия для нулевой гипотезы и альтернативной гипотезы

При применении метода моментов оценка параметра смещения распределения Коши равна выборочной медиане. При применении метода максимального правдоподобия оценка параметра смещения равна выборочному среднему.

Для распределения Коши логарифм правдоподобия имеет вид:

где n - размер выборки, γ - масштаб распределения, - i-ое значение выборки.

Вычислим значения логарифма правдоподобия для нулевой и альтернативной гипотез:

:

:

Значение статистики отношения правдоподобия равно:

В случае оценки параметра сдвига, полученной в результате применения метода максимального правдоподобия:

–для уровня значимости

Для уровня значимости =0,05 и количества степеней свободы k=1 (так как проверяем только один параметр) можно использовать критическое значение распределения хи-квадрат с уровнем значимости 0,05 и 1 степенью свободы: =3,84.

Если значение статистики LR больше критического значения, то нулевая гипотеза отвергается.

Вычислим значение статистики LR для каждого метода оценки смещения: LR =0. По хи-квадрату не отвергается. По другому критерию отвергается.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Принята по результатам расчета** | **Вывод** |
| **Верна на самом деле** |  |  | Вероятность правильного решения |
|  | Ошибка первого рода (ложная тревога, ложное срабатывание). Вероятность – |
|  |  | Ошибка второго рода  Вероятность ошибки |
|  | Вероятность правильного решения |

**Для уровня значимости :**

Вероятность ошибки первого рода определяет заданный уровень значимости *α*. Поскольку *α* = 0,05, следовательно, вероятность ошибки первого рода равна 5%.

Вероятность ошибки второго рода:

Таким образом, вероятность ошибки второго рода для уровня значимости α = 0,05 равна 5,96%.

### Гипотезы о виде распределения

#### Критерий Колмогорова:

Была составлена нулевая гипотеза H0: выборка подчиняется распределению Коши.

Была построена эмпирическую функцию распределения (ЭФР) по выборке в пункте 3.3.

Далее произведено сравнение ЭФР с теоретической функцией распределения Коши с помощью статистики Колмогорова:

где - теоретическая функция распределения Коши, - ЭФР, max - максимальное отклонение между двумя функциями.

Произведено сравнение значения статистики D с критическим значением при уровне значимости α=0,05. =0,52, вычисляется по таблице Колмогорова-Смирнова.

Если D > , то нулевая гипотеза отвергается и можно сделать вывод, что выборка не подчиняется распределению Коши.

Для данной выборки при loc=median была реализована программа, которая подсчитала наибольшую разность эмпирической и теоретической функций: 0.01674

Таким образом D<, а это значит, что данная выборка подчиняется распределению Коши.

#### Критерий

Выборка была разделена на 8 интервалов.

Теоретическая вероятность попадания в каждый из интервалов:

Статистика критерия имеет вид: , где частота попадания в интервал.

Получившееся значение = 12.3575. Это значение не превышает критическое значения , значит данная выборка подчиняется распределению Коши.

### Гипотезы об однородности выборок

Свойства распределения Коши резко отличны от свойств распределения Гаусса, Лапласа и других экспоненциальных распределений.

Распределение Коши описывает отношение двух нормально распределённых случайных величин. Распределение Коши является *бесконечно делимым*: сумма независимых случайных величин, распределённых по Коши, также распределена по Коши. Этим же свойством обладают *нормальное*, *хи-квадрат* и *гамма-распределения*. Распределение Коши напоминает внешне *нормальное распределение*, однако имеет более тяжёлые хвосты. Кроме этого, график функции распределения Коши напоминает *распределения Стьюдента*. С возрастанием числа степеней свободы *k* распределение Стьюдента переходит в нормальное распределение и в большей мере напоминает распределение Коши. Также, соответствующие графики для *хи-квадрат* *распределения* и распределения Коши будут также схожи, если увеличить число степеней свободы *k*.

Таким образом, в качестве распределений, близких по виду к распределению Коши, можно отметить *нормальное распределение*, *распределение Стьюдента* и *хи-квадрат распределение*.

*Метод обратного преобразования* – способ генерации случайных величин с заданной функцией распределения, путём модификации работы генератора равномерно распределённых чисел.

Рассмотрим функцию распределения интересующего нас закона распределения и попробуем выразить обратную функцию:

Теперь вспоминаем, что функция распределения F(x) всегда принимает значения из интервала . Тогда, если сгенерировать случайную величину , равномерно распределённую на отрезке , то через обратную функцию можно получить значение , принимающие все значения из области определения. Причём возвращаемые значения случайной величины подчинены заданному закону распределения. Это можно объяснить тем, что вероятность получить значения на участках, где функция быстро возрастает, больше, чем на участках, где функция распределения возрастает медленно.

Получается, что метод обратного преобразования деформирует равномерное распределение в соответствии с интересующим нас законом распределения. Если мы генерируем равномерное , то, действительно, полученная методом обратного преобразования случайная величина x имеет распределение .

Получим нормальное распределение методом обратного преобразования.

Полученная по этому выражению случайная величина будет подчинена нормальному закону распределения.

Функция – функция ошибок, определяемая как

При попытке выразить обратную функцию возникает проблема, характеризующаяся недостаточными знаниями для работы с .

Применим аналогичные рассуждения для работы с хи-квадрат распределением. Соответствующая функция распределения: , где – полная гамма-функция, – неполная гамма-функция.

Если вещественная часть комплексного числа *z* положительна, то гамма-функция определяется через абсолютно сходящийся интеграл:

Как и в случае выше, здесь возникают сложности при работе с данными функциями распределения, поэтому не удаётся выразить с помощью метода обратного преобразования обратную функцию в явном виде. Соответственно, без данных функций сгенерировать выборку объёма *n* не удаётся. Тем не менее, при оценке параметров распределения Коши мы воспользовались конечной формулой перехода,, поэтому ранее уже была составлена выборка *X2*, подчинённая закону нормального распределения.

Если попытаться применить упомянутый выше метод для распределения Стьюдента, то мы вновь столкнёмся с обозначенной ранее проблемой. Функция распределения в этом случае выглядит следующим образом:

Данная функция распределения содержит в себе как гамма-функцию, так и гипергеометрическую функцию, для работы с которыми теоретических знаний и практических навыков недостаточно.

В связи с проблемой, описанной выше, построить критерий для проверки гипотезы вида: против альтернативы не удаётся.

Общий подход к построению k-выборочных критериев однородности законов заключается в следующих этапах:

1. Формулировка нулевой гипотезы о равенстве распределений в k выборках. Нулевая гипотеза предполагает, что все k выборок были взяты из одного и того же распределения.

2. Выбор статистики, которая будет использоваться для проверки нулевой гипотезы. Статистика должна быть такой, чтобы ее распределение было известно при верности нулевой гипотезы.

3. Определение критической области для статистики. Критическая область определяется таким образом, чтобы вероятность ошибки первого рода была равна заранее заданному уровню значимости.

4. Вычисление значения статистики для каждой выборки и определение ее попадания в критическую область. Если значение статистики попадает в критическую область, то нулевая гипотеза отвергается.

5. Интерпретация результатов и принятие решения о равенстве или неравенстве распределений в k выборках.

Примерами k-выборочных критериев однородности являются критерий Колмогорова-Смирнова, критерий Вилкоксона-Манна-Уитни, критерий Фридмана и др.

Соответственно, исследовать сходимость к предельным распределениям статистики критерия в данной ситуации также не представляется возможным. Оценить мощность критерия в данном случае мы также не можем. Тем не менее, алгоритм оценки мощности критерия уже был ранее использован в пункте “Гипотезы о параметрах распределения”.

### Нормальное распределение

Критерий Колмогорова-Смирнова.

0: Выборки из одной генеральной совокупности.

1: Выборки из разных генеральных совокупностей.

Эмпирическое значение критерия: , где количество элементов в первой выборке, во второй, .

Если , то принимается гипотеза 0, в другом случае 1.

Для уровня значимости 0,05 критическое значение (0,05) = 1,36.

Эмпирическое значение превосходит критическое, следовательно выборки не были произведены из одной генеральной совокупности.

# ВЫВОД

В данной курсовой работе было рассмотрено распределение Коши, которое является одним из важных распределений в статистике и математической физике. Были рассмотрены основные свойства этого распределения, его плотность вероятности, функция распределения.

Одним из важных результатов работы является построение эмпирической, теоретической функции, а также гистограммы. По этим построениям были сделаны выводы о неизвестном параметре смещения, а также тщательно изучено само распределение и его свойства.

Также были рассмотрены некоторые методы оценки параметров распределения Коши на основе выборки данных, такие как метод максимального правдоподобия и метод моментов, интервальные оценки, критерии и Колмогорова. Было показано, что эти методы могут быть применены для оценки параметра смещения распределения Коши.

В целом, данная курсовая работа является полезным исследованием распределения Коши, которое может быть использовано в различных областях науки и техники. Результаты работы могут быть использованы для дальнейшего исследования этого распределения и его применений.

Список используемых источников

1. Н. И. Чернова. Математическое ожидание случайно величины. URL: <https://tvims.nsu.ru/chernova/tv/lec/node42.html#3166> (дата обращения: 15.06.2023)
2. Н. И. Чернова. Математические ожидания и дисперсии стандартных распределений. URL: <https://tvims.nsu.ru/chernova/tv/lec/node46.html> (дата обращения: 15.06.2023)
3. Википедия – свободная энциклопедия. URL: <https://ru.wikipedia.org/wiki/Распределение_Коши> (дата обращения: 15.06.2023)
4. ВикиБриф. Распределение Коши – Cauchy distribution. URL: <https://ru.wikibrief.org/wiki/Cauchy_distribution> (дата обращения: 15.06.2023)
5. ВикиБриф. Усеченное среднее – Truncated mean. URL: <https://ru.wikibrief.org/wiki/Truncated_mean> (дата обращения: 15.06.2023)
6. ВикиБриф. Медиана - Median. URL: https://ru.wikibrief.org/wiki/Median (дата обращения: 15.06.2023)
7. Галкин В.М., Ерофеева Л.Н., Лещева С.В. Оценки параметра распределения Коши (2014) // Труды Нижегородского государственного университета им. Р.Е. Алексеева № 2 (104). URL: <https://www.nntu.ru/frontend/web/ngtu/files/nauka/izdaniya/trudy/2014/02/314-319.pdf> (дата обращения: 15.06.2023).
8. Шинкеев М.Л. Оценка параметров распределения Коши // Национальный исследовательский Томский политехнический университет. – Томск. URL: <https://www.nntu.ru/frontend/web/ngtu/files/nauka/izdaniya/trudy/2014/02/314-319.pdf> (дата обращения: 15.06.2023).
9. Проверка статистических гипотез. URL: <https://e.vyatsu.ru/pluginfile.php/462632/mod_resource/content/2/Теоретический%20материал__проверка%20статгипотез_пункт%203.6.1.pdf> (дата обращения: 15.06.2023)/

ПРИЛОЖЕНИЕ А

Листинг программы «Распределение Коши»

import math

import random

#Сканирование выборки из файла

def scan\_file(filename):

    numbers = []

    with open(filename, "r") as file:

        for line in file:

            try:

                num = float(line.strip())

                numbers.append(num)

            except ValueError:

                print(f"Ignoring invalid number: {line.strip()}")

    return numbers

list\_elem = scan\_file("var\_29\_cauchy.csv")

MIN\_MERGE = 32

#Длина выборки

def get\_len(data):

    len = 0

    for i in data:

        len += 1

    return len

def calcMinRun(n):

    r = 0

    while n >= MIN\_MERGE:

        r |= n & 1

        n >>= 1

    return n + r

def insertionSort(arr, left, right):

    for i in range(left + 1, right + 1):

        j = i

        while j > left and arr[j] < arr[j - 1]:

            arr[j], arr[j - 1] = arr[j - 1], arr[j]

            j -= 1

def merge(arr, l, m, r):

    len1, len2 = m - l + 1, r - m

    left, right = [], []

    for i in range(0, len1):

        left.append(arr[l + i])

    for i in range(0, len2):

        right.append(arr[m + 1 + i])

    i, j, k = 0, 0, l

    while i < len1 and j < len2:

        if left[i] <= right[j]:

            arr[k] = left[i]

            i += 1

        else:

            arr[k] = right[j]

            j += 1

        k += 1

    while i < len1:

        arr[k] = left[i]

        k += 1

        i += 1

    while j < len2:

        arr[k] = right[j]

        k += 1

        j += 1

#Сортировка

def timsort(data):

    arr = list(data)

    n = get\_len(arr)

    minRun = calcMinRun(n)

    for start in range(0, n, minRun):

        end = min(start + minRun - 1, n - 1)

        insertionSort(arr, start, end)

    size = minRun

    while size < n:

        for left in range(0, n, 2 \* size):

            mid = min(n - 1, left + size - 1)

            right = min((left + 2 \* size - 1), (n - 1))

            if mid < right:

                merge(arr, left, mid, right)

        size = 2 \* size

    return arr

#Максимум

def find\_maximum(arr):

    if get\_len(arr) == 0:

        return None

    max\_value = arr[0]

    for i in range(1, get\_len(arr)):

        if arr[i] > max\_value:

            max\_value = arr[i]

    return max\_value

#Минимум

def find\_minimum(data):

    if get\_len(data) == 0:

        return None

    min\_value = data[0]

    for i in range(1, get\_len(data)):

        if data[i] < min\_value:

            min\_value = data[i]

    return min\_value

#Максимум из отсортированного списка

def get\_max(data, len):

    return data[len-1]

#Минимум из отсортированного списка

def get\_min(data):

    return data[0]

#Сумма элементов выборки

def get\_sum(data):

    num = 0

    for i in data:

        num += i

    return num

#Выборочное среднее

def get\_sample\_mean(sum, len):

    return (sum/len)

#Медиана

def get\_median(sorted\_data, len):

    if len % 2 == 0:

        return (sorted\_data[len // 2 - 1] + sorted\_data[len // 2]) / 2

    else:

        return sorted\_data[len // 2]

#Мода

def get\_mode(data):

    frequency = {}

    for item in data:

        frequency[item] = frequency.get(item, 0) + 1

    max\_frequency = find\_maximum(list(frequency.values()))

    if max\_frequency == 1:

        return None

    mode = [item for item, freq in frequency.items() if freq == max\_frequency]

    return mode

#Размах выборки

def get\_sample\_range(max, min):

    return (max - min)

#Смещенная дисперсия

def get\_biased\_variance(len, sample\_mean, data):

    if len == 0:

        return 0

    variance = get\_sum((x - sample\_mean) \*\* 2 for x in data) / len

    return variance

#Несмещенная дисперсия

def get\_unbiased\_variance(data, len, sample\_mean):

    if len <= 1:

        return 0

    variance = get\_sum((x - sample\_mean) \*\* 2 for x in data) / (len - 1)

    return variance

#Начальный момент

def get\_sample\_moment(data, len, sample\_mean, k=3):

    if len == 0:

        return 0

    moment = get\_sum((x - sample\_mean) \*\* k for x in data) / len

    return moment

#Центральный момент

def get\_sample\_central\_moment(data, len, sample\_mean, k=3):

    if len == 0:

        return 0

    central\_moment = get\_sum((x - sample\_mean) \*\* k for x in data) / len

    return central\_moment

#Вывод статистик

sorted\_data=timsort(list\_elem)

len=get\_len(list\_elem)

elem\_min=get\_min(sorted\_data)

elem\_max=get\_max(sorted\_data, len)

sum = get\_sum(list\_elem)

sample\_mean = get\_sample\_mean(sum, len)

median = get\_median(sorted\_data, len)

mode = get\_mode(list\_elem)

sample\_range = get\_sample\_range(elem\_min, elem\_max)

biased\_variance = get\_biased\_variance(len, sample\_mean, list\_elem)

unbiased\_variance = get\_unbiased\_variance(list\_elem, len, sample\_mean)

sample\_moment = get\_sample\_moment(list\_elem, len, sample\_mean)

sample\_central\_moment = get\_sample\_central\_moment(list\_elem, len, sample\_mean)

print(f"Сумма выборки равна {sum}")

print(f"Среднее выборчное равно {sample\_mean}")

print(f"Медиана равна {median}")

print(f"Мода равна {mode}")

print(f"Размах выборки равен {sample\_range}")

print(f"Смещенная дисперсия равна {biased\_variance}")

print(f"Несмещенная дисперсия равна {unbiased\_variance}")

print(f"Выборочный начальный момент 3-ого порядка равен {sample\_moment}")

print(f"Выборочный центральный момент 3-ого порядка равен {sample\_central\_moment}")

#Выбираем случайные элементы из выборки

def select\_random\_elements(arr, x):

    if x > get\_len(arr):

        raise ValueError("The number of elements to select is greater than the array length.")

    random\_elements = random.sample(arr, x)

    return random\_elements

sample\_10 = select\_random\_elements(list\_elem, 10)

sample\_100 = select\_random\_elements(list\_elem, 100)

sample\_200 = select\_random\_elements(list\_elem, 200)

samples = ["sample\_10", "sample\_100", "sample\_200"]

import csv

#Сохранение их в файл csv

def save\_to\_csv(data, filename):

    with open(filename, 'w', newline='') as csvfile:

        writer = csv.writer(csvfile)

        for value in data:

            writer.writerow([value])

samples = ["sample\_10", "sample\_100", "sample\_200"]

for sample in samples:

    save\_to\_csv(eval(sample), f"{sample}.csv")

#График эмпирической функции

import numpy as np

import scipy.stats as sps

import matplotlib.pyplot as plt

blue = '#0099CC'

def emp\_func(x):

    sort\_list=timsort(x)

    n = get\_len(x)

    y = np.zeros(n)

    for i in range(n):

        y[i] = i / float(n)

    return sort\_list, y

#График по всей выборки

def func\_emp():

    x\_ecdf, y\_ecdf = emp\_func(sorted\_data)

    plt.plot(x\_ecdf, y\_ecdf, marker='.', linestyle='none')

    plt.xlabel('Значение')

    plt.xlim([-25, -10])

    plt.ylabel('F(x)')

    plt.title('Эмпирическая функция распределения')

    plt.show()

#График по составленным выборкам

def func\_emp\_samples():

    for sample in samples:

        x\_sub\_ecdf, y\_sub\_ecdf = emp\_func(eval(sample))

        plt.plot(x\_sub\_ecdf, y\_sub\_ecdf, marker='.', linestyle='none', label=f'Подвыборка размера {get\_len(x\_sub\_ecdf)}')

    plt.legend(loc='lower right')

    plt.xlim([-25, -10])

    plt.xlabel('Значение')

    plt.ylabel('F(x)')

    plt.title('Эмпирическая функция распределения для подвыборок')

    plt.show()

func\_emp()

func\_emp\_samples()

#Функция определения гистограммы

def hist\_func(x, colum):

    hist = np.zeros(colum)

    sorted\_x=timsort(x)

    x\_max=float(get\_max(sorted\_x, get\_len(x)))

    x\_min=float(get\_min(sorted\_x))

    width = (x\_max - x\_min) / colum

    for value in x:

        index = int((value - x\_min) / width)

        if index == colum:

            index -= 1

        hist[index] += 1

    hist = hist / (get\_len(x) \* width)

    bins = np.linspace(x\_min, x\_max, colum+1)

    return bins, hist

#Построение гистограммы

def build\_hist(i, sample):

    colors = ['r', 'g', 'b']

    titles = ['Гистограмма распределения Коши с 10 элементами выборки',

          'Гистограмма распределения Коши со 100 элементами выборки',

          'Гистограмма распределения Коши с 200 элементами выборки']

    bins, hist = hist\_func(sample, 50)

    plt.bar(bins[:-1], hist,

                color=colors[i], alpha=0.5,

                width=(bins[1:]-bins[:-1]),

                label="Количество элементов выборки: {}".format(get\_len(sample)))

    plt.title(titles[i])

    plt.xlabel('x')

    plt.ylabel('p(x)')

    plt.legend()

    plt.show()

i=0

sample[0]

for sample in samples:

    build\_hist(i, eval(sample))

    i+=1

#Теоретическая функкция распределения Коши

def cauchy(x, x0):

    sort\_list=timsort(x)

    n = get\_len(x)

    y = np.zeros(n)

    for i in range(n):

        y[i] = 1 / math.pi \* math.atan(sorted\_data[i] - x0) + 0.5

    return sort\_list, y

#Построение теоретического графика

def build\_graph(arr):

    for i in range(3):

        x\_ecdf, y\_ecdf = cauchy(sorted\_data, arr[i])

        plt.plot(x\_ecdf, y\_ecdf, marker='.', label=f'loc={arr[i]}')

    plt.xlabel('Значение')

    plt.xlim([-50,50])

    plt.ylabel('F(x)')

    plt.title('Функция распределения со случайным смещением')

    plt.legend()

    plt.show()

arr1=[-17,-7,1]

arr2=[-30,-20,-10]

arr3=[-30,0,30]

build\_graph(arr1)

build\_graph(arr2)

build\_graph(arr3)

#Усеченное среднее

def truncated\_mean(k):

    data=np.zeros(get\_len(sorted\_data))

    for i in range(get\_len(sorted\_data)-(k\*2)):

        data[i]=sorted\_data[i+k]

    sum\_data=get\_sum(data)

    sample\_mean\_data=get\_sample\_mean(sum\_data,get\_len(data))

    print(f"Усеченное среднее равно {sample\_mean\_data}")

    return sample\_mean\_data

sample\_mean\_data=truncated\_mean(2)

#Сравнение полученных функций

def func\_estimate(arr):

    for i in range(2):

        x\_ecdf, y\_ecdf = cauchy(sorted\_data, arr[i])

        plt.plot(x\_ecdf, y\_ecdf, marker='.', label=f'loc={arr[i]}')

    x\_ecdf, y\_ecdf = emp\_func(sorted\_data)

    plt.plot(x\_ecdf, y\_ecdf, marker='.', label=f'Эмперическая функция распределения')

    plt.xlabel('Значение')

    plt.xlim([-20,-15])

    plt.ylabel('F(x)')

    plt.title('Сравнение полученных теоретических функций с эмпирической')

    plt.legend()

    plt.show()

arr=[-17.420171177624205, sample\_mean\_data]

func\_estimate(arr)

sqrt1=unbiased\_variance\*\*0.5

sqrt2=len\*\*0.5

#Интервальные оценки

def formula\_minus(C):

    x=float(C)\*sqrt1

    y=x/sqrt2

    return(sample\_mean - y)

def formula\_plus(C):

    return(sample\_mean + ((C\*sqrt1)/sqrt2))

print(f"Левая часть интервала при уровне доверия 0,95 равна:{formula\_minus(1.96)}")

print(f"Правая часть интервала при уровне доверия 0,95 равна:{formula\_plus(1.96)}")

print(f"Левая часть интервала при уровне доверия 0,99 равна:{formula\_minus(2.58)}")

print(f"Правая часть интервала при уровне доверия 0,99 равна:{formula\_plus(2.58)}")

def formula\_hypo(len, data, loc):

    meow=len\*math.log(math.pi)

    nya=len\*math.log(1)

    sum=0

    for i in data:

        sum=math.log((i-loc)\*\*2)

    L=-meow-nya-sum

    return L

L0\_mediana=formula\_hypo(len, sorted\_data, median)

L1\_sample\_mean=formula\_hypo(len, sorted\_data, median)

LR=-2\*math.log(L0\_mediana/L1\_sample\_mean)

print(f"Коэффициент правдоподобия: {LR}")

#Разность эмперической и теоретической функции

def max\_dif(emp, func):

    for i in range(len):

        emp[i]=func[i]-emp[i]

    return emp

sort\_m, func=cauchy(sorted\_data, median)

sort\_k, emp=emp\_func(sorted\_data)

max=max\_dif(emp, func)

max\_sort=timsort(max)

print(f"Наибольшая разность эмпирической и теоретической функций: {get\_max(max\_sort, get\_len(max\_sort))}")

#Критерий хи-квадрат

def cauchy\_solo(x, x0):

    y = 1 / math.pi \* math.atan(x - x0) + 0.5

    return y

def x\_2(data, n, arr):

    sum=0

    for i in range(8):

        if(i==0): continue

        n[i]=n[i]/len

        F=cauchy\_solo(data[arr[i]], median)-cauchy\_solo(data[arr[i-1]], median)

        sum=sum+((n[i]-F)\*\*2)/F

    return sum

arr=[0,10,66,101,182,207,269,284,199]

n=[10,56,35,81,25,62,15,16]

x2=len\*x\_2(sorted\_data,n,arr)

print(f"Хи-квадрат: {x2}")

#Метод моментов

def moment(data):

    sum=0

    for i in data:

        sum=sum+(abs(i))\*\*(1/25)

    sum=(1/len)\*sum

    return sum

sum=(moment(sorted\_data)\*math.cos((math.pi\*(1/25))/2))\*\*25

print(f"loc={sum}")

#Перебор метода максимального правдоподобия

def perebor(data):

    x=-17.5

    sum=0

    while(x>-18.5):

        for i in data:

            sum=sum+((i-x)/(1+(i-x)\*(i-x)))

    if(sum<0.5 and sum>-0.5):return x

    else: x-=0.01

print({perebor(sorted\_data)})