Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет телекоммуникаций и информатики»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №0

по дисциплине «Моделирование»

Выполнил:
Студент гр. ИВ-622
Корягина В.А

Проверила:
Ассистент Кафедры ВС
Петухова Я.В.

Новосибирск 2020

СОДЕРЖАНИЕ

Постановка задачи

Необходимо взять готовую реализацию трех генераторов псевдослучайных чисел на и убедиться в их равномерном распределении, используя такие параметры как критерий Пирсона и автокорреляция.

Теоретические сведения

Пусть X - исследуемая случайная величина. Требуется проверить гипотезу H_0 о том, что данная случайная величина подчиняется закону распределения *F(x)*. Для этого необходимо произвести выборку из n независимых наблюдений над случайной величиной X: $X^n = (x_1, \dots, x_n)$, $x_i \in [a, b]$, $\forall_i = 1 \dots n$, и по ней построить эмпирический закон распределения *F'(x)* случайной величины *X*.

Гипотеза $H_0^{'}: X^n$ порождается ранее упомянутой функцией *F'(x)*. Разделим [*a*,*b*] на *k* непересекающихся интервалов $\left[a_i, b_i\right], i = 1 \dots k$.
Пусть n_j — количество наблюдений в j-м интервале;
$p_j = F\left(b_j\right) - F(a_j)$ - вероятность попадания наблюдения в j-й интервал при выполнении гипотезы $H_0^{'}$;
$E_j = np_j$ - ожидаемое число попаданий в j-й интервал; тогда распределение χ^2 с числом степеней свободы k−1 будет иметь следующую статистику:

$$\chi^2 = \sum_{i=1}^{k} \frac{(n_i - p_i N)^2}{p_i N} \sim \chi^2_{k-1}$$

В зависимости от значения критерия χ^2 гипотеза H_0 может приниматься либо отвергаться:

- $\chi^2_1 < \chi^2 < \chi^2_2$ - гипотеза H_0 выполняется.
- $\chi^2 \leq \chi^2_1$ — попадает в левый «хвост» распределения. Следовательно, теоретические и практические значения очень близки. Если, к примеру, происходит проверка генератора случайных чисел, который сгенерировал *n* чисел из отрезка [0,1] и выборка X^n распределена равномерно на [0,1], то генератор нельзя назвать случайным (гипотеза случайности не выполняется), т.к. выборка распределена слишком равномерно, но гипотеза H_0 выполняется.

- $\chi^2 \geq \chi_2^2$ — попадает в правый «хвост» распределения, гипотеза H_0 отвергается.

Автокорреляция уровней ряда – корреляционная между последовательными уровнями одного и того же ряда динамики (сдвинутыми на определенный промежуток времени L – лаг). Автокорреляция может быть измерена коэффициентом автокорреляции.

Лаг (сдвиг во времени) определяет порядок коэффициента автокорреляции. Если L = 1, то имеем коэффициент автокорреляции 1-го порядка $r_{t,t-1}$, если L = 2, то коэффициент автокорреляции 2-го порядка $r_{t,t-2}$ и т.д.

Рассчитав несколько коэффициентов автокорреляции, можно определить лаг (I), при котором автокорреляция ($r_{t,t-L}$) наиболее высокая, выявив тем самым структуру временного ряда. Если наиболее высоким оказывается значение $r_{t,t-1}$, то исследуемый ряд содержит только тенденцию. Если наиболее высоким оказался $r_{t,t-L}$, то ряд содержит (помимо тенденции) колебания периодом L. Если ни один из (l=1;L) не является значимым, можно сделать одно из двух предположений:

- либо ряд не содержит тенденции и циклических колебаний, а его уровень определяется только случайной компонентой;
- либо ряд содержит сильную нелинейную тенденцию, для выявления которой нужно провести дополнительный анализ.

Значение (по модулю)	**Интерпретация**
до 0,2	очень слабая корреляция
до 0,5	слабая корреляция
до 0,7	средняя корреляция
до 0,9	высокая корреляция
свыше 0,9	очень высокая корреляция

Автокорреляционная функция – предназначена для оценки корреляции между смещенными копиями последовательностей.

$$a(\tau) = \frac{\sum_{i=1}^{N-\tau} (x_i - Ex)\,(x_{i+\tau} - Ex)}{(N-\tau) * S^2(x)}, \text{ где}$$

Ex - математическое ожидание — среднее значение случайной величины при стремлении количества выборок или количества её измерений (иногда говорят — количества испытаний) к бесконечности:

$$E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i$$

$S^2(x)$ - выборочная дисперсия случайной величины — это оценка теоретической дисперсии распределения, рассчитанная на основе данных выборки:

$$S^2(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - x')^2 = \frac{1}{N} \sum_{i=1}^{N} x_i^2 - (x')^2$$

$x_{i+\tau}$ - множество значений другой случайной величины (полученной из значений прошлой случайной величины, но с некоторым смещением);

n - мощность множества случайных величин;

τ - смещение последовательности.

Ход работы

Необходимо определить исследуемые генераторы случайных чисел:

- Генератор из стандартной библиотеки языка программирования Java (java.util.Random)
- Генератор однородных псевдослучайных значений, применимых для использования в изолированных параллельных вычислениях языка Java (java.util.SplittableRandom)
- Криптографически сильный генератор случайных чисел языка Java (java.security.SecureRandom)

Для расчета значения критерия Пирса необходимо выбрать количество интервалов k, количество генерируемых значений N и генератор случайных чисел.

При запуске программы происходит генерация чисел, их подсчет количеств попаданий на интервалы, после чего рассчитывается “хи-квадрат” и автокорреляция.

Далее построим таблицу расчета критерия Пирса и автокорреляция при варьировании значений количества интервалов k и количества генерируемых значений N:

Генератор случайных чисел	**k = 10 000 N = 1 000**	**k = 1 000 000 N = 1 000**	**k = 1 000 000 N = 100**
Java Random	$\chi^2_{эксп} = 1005$	$\chi^2_{эксп} = 980,93$	$\chi^2_{эксп} = 128$
Splittable Random	$\chi^2_{эксп} = 1007,4$	$\chi^2_{эксп} = 1012,23$	$\chi^2_{эксп} = 117,174$
Secure Random	$\chi^2_{эксп} = 959$	$\chi^2_{эксп} = 1059,416$	$\chi^2_{эксп} = 114,93$
$\chi^2_{таб}$	$\chi^2_{таб} = 1142.848$	$\chi^2_{таб} = 1142.848$	$\chi^2_{таб} = 134,642$

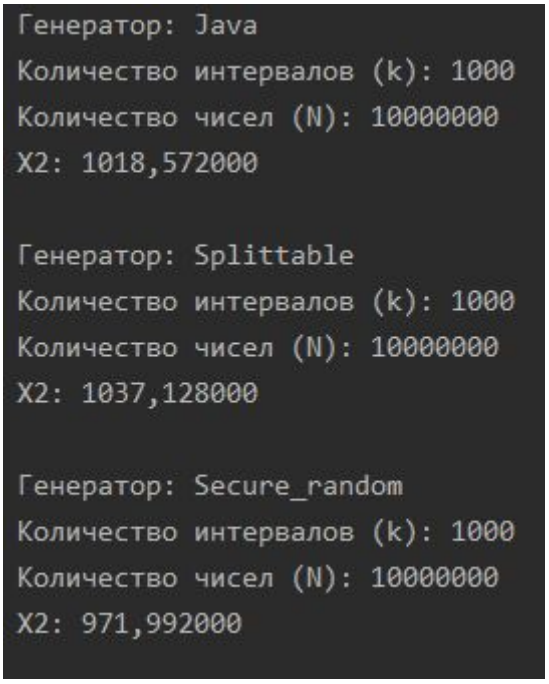

Рисунок 1 - Результат выполнения программы расчета критерия Пирса и автокорреляции (k = 1000, N = 1 000 000)

Для расчета коэффициента корреляции определяется параметр τ (смещение в последовательности от 1 до половины наших интервалов).

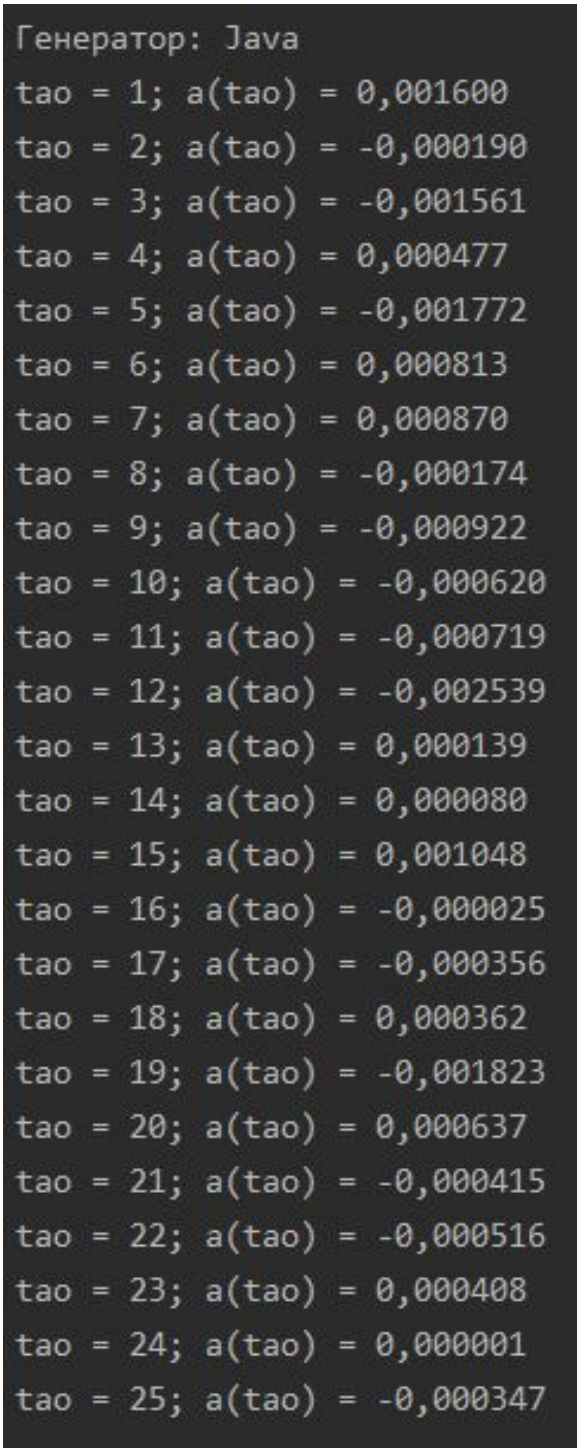

Рисунок 2 - Результат расчета коэффициента корреляции генератором Java при изменении τ . (k = 1000, N = 1 000 000)

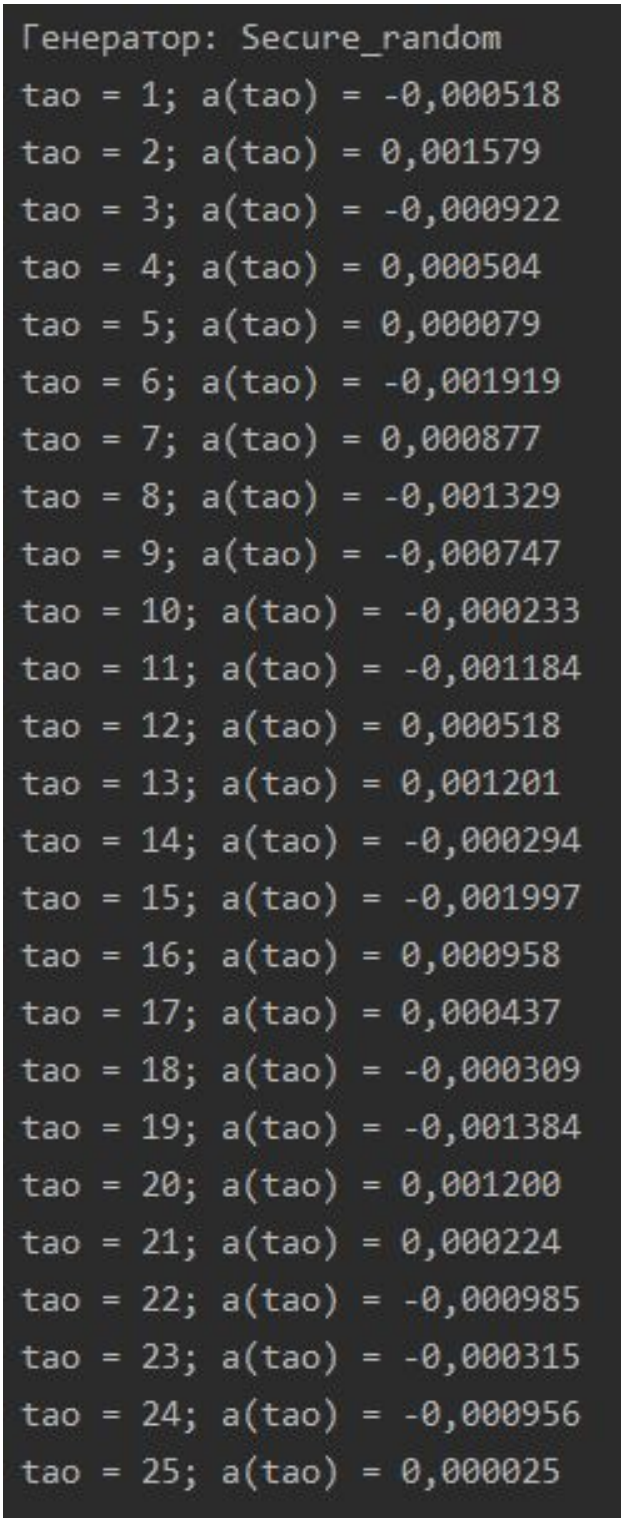

Рисунок 3 - Результат расчета коэффициента корреляции генератором Secure Random при изменении τ . (k = 1000, N = 1 000 000)

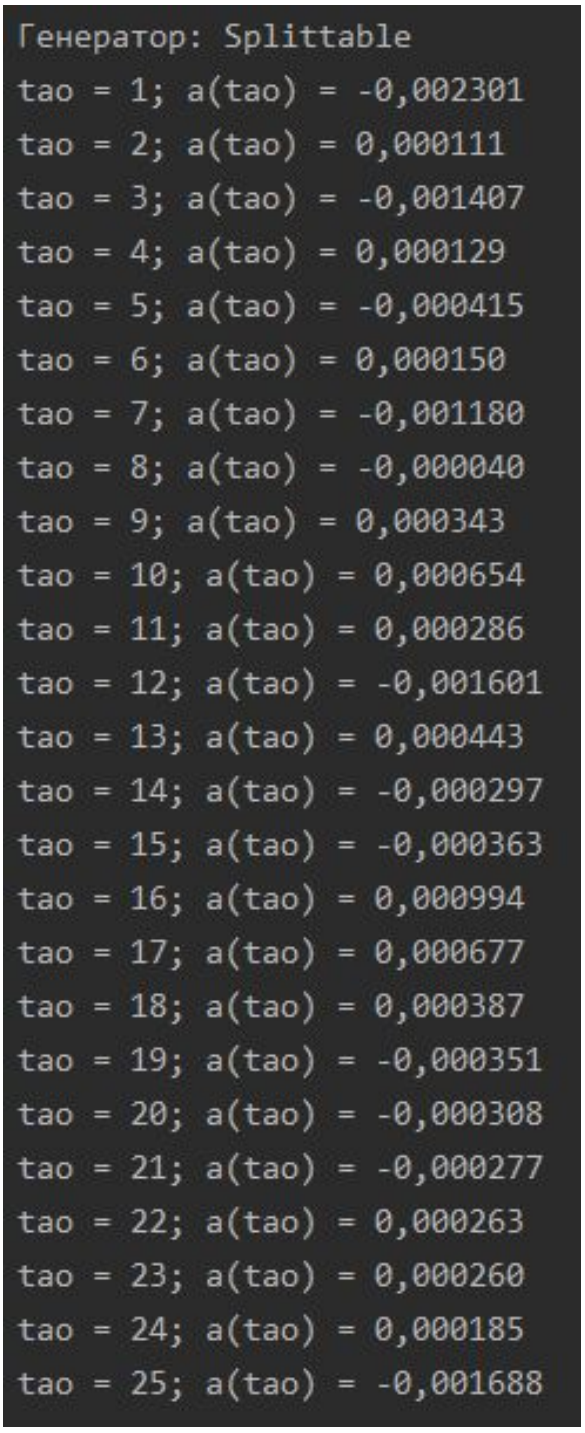

Рисунок 3 - Результат расчета коэффициента корреляции генератором Splittable Randpm при изменении τ . (k = 1000, N = 1 000 000)

Заключение

В рамках лабораторной работы были проведены эксперименты с генераторами случайных чисел языка Java. С помощью данных генераторов был произведен анализ равномерности распределения по критерию Пирса и автокорреляции.

По результатам экспериментов видно, что критерии «хи-квадрат» не превышает табличного значения $\chi^2_{эксп} < \chi^2_{таб}$ и можно сделать вывод о том, что гипотезы о равновероятном распределении в генераторах случайных чисел принимается. Если бы $\chi^2_{эксп}$ значение попало в критическую область (была бы равна или больше чем $\chi^2_{таб}$), то гипотеза была бы отклонена. Следовательно, для всех трех различных генераторов гипотеза о равномерном распределении принимается.

По результатам эксперимента расчета критерия Пирса имеется следующая зависимость: чем больше отклонение, тем менее распределенным становится распределение. Также значение критерия Пирсона, не превышало критического значения, следовательно, гипотезу о равномерном распределении нельзя отвергнуть.

В каждом из наших проведенных экспериментов автокорреляционная функция при изменении параметра τ (смещение в последовательности от 1 до половины наших интервалов) приближена к нулю, что говорит нам об очень слабой силе корреляции. Она показывает зависимости между данными крайне мала и также можно отметить, что числа генерируются случайным образом.

Приложение

```
import java.util.*
import kotlin.math.pow

fun generateGraph(k: Int, N: Int, randomMethod: RandomMethod) {
    check(N / k > 5) { "Недостаточное количесто генерируемых чисел!" }
    var graph = mutableMapOf<Int, Int>()
    repeat(N) {
        val interval = (randomMethod.generate() / (1.0 / k)).toInt() + 1
        graph[interval] = graph[interval]?.plus(1) ?: 1
    }
    graph = graph.toSortedMap()
    val x2 = calculateX2(graph, k, N).double()
    println("""
        Генератор: ${randomMethod.name.toLowerCase().capitalize()}
        Количество интервалов (k): $k
        Количество чисел (N): ${N}0
        X2: $x2

    """.trimIndent())
}

private fun calculateX2(nList: Map<Int, Int>, k: Int, n: Int) = nList.map{(_, ni)->
    val pi = 1.0 / k
    (ni - pi * n).pow(2.0) / (pi * n)
}.sum()

fun calculateCov(N: Int, k: Int, randomMethod: RandomMethod) {
    println("Генератор: ${randomMethod.name.toLowerCase().capitalize()}")
    val taoMax = 25
    val aCov = ACov(N, randomMethod)
    (1..taoMax).map { tao ->
        val cov = aCov.calculateByTao(tao)
        println("tao = $tao; a(tao) = ${cov.double()}")
    }
}
```

```
class ACov(N: Int, randomMethod: RandomMethod) {
    private val list = mutableListOf<Double>()
    private val xAverage: Double
    private val s2: Double

    init {
        var xSum = 0.0
        var x2sum = 0.0
        repeat(N) {
            val random = randomMethod.generate()
            list.add(random)
            x2sum += random * random
            xSum += random
        }
        xAverage = xSum / N
        s2 = (x2sum / N - xAverage.pow(2.0))
    }

    fun calculateByTao(tao: Int) = (0 until list.size - tao).map { i ->
        (list[i] - xAverage) * (list[i + tao] - xAverage)
    }.sum() / ((list.size - tao) * s2)
}

enum class RandomMethod {
    JAVA, SPLITTABLE, SECURE;

    fun generate() = when (this) {
        SECURE -> SecureRandom().nextDouble()
        JAVA -> Random().nextDouble()
        SPLITTABLE -> SplittableRandom().nextDouble()
    }
}

fun main() {
    val k = 1000
    val N = 1_000_000
    RandomMethod.values().forEach { random ->
        generateGraph(k = k, N = N, randomMethod = random)
        calculateCov(k = k, N = N, randomMethod = random)
    }
}
```