Зубарева Наталия БПИ195
Вариант 8

Задача №2. Тестирование последовательностей псевдослучайных чисел.

Датчики псевдослучайных чисел разрабатываются так, чтобы генерируемые ими последовательности можно было считать реализациями независимых случайных величин, равномерно распределённых на единичном отрезке. Ваше задание — реализовать такой датчик и проверить генерируемую последовательность на равномерность и независимость.

Выполните следующие шаги:

1. Рассчитайте 100 псевдослучайных чисел методом, соответствующим вашему варианту. Описание методов дано на второй и третьей страницах.
2. Приведите первые 10 чисел этой последовательности.
3. Постройте гистограмму с 10 столбцами для полученной последовательности.
4. Проверьте гипотезу о том, что последовательность имеет распределение R(0, 1) критерием хи-квадрат, разбив интервал [0; 1) на десять равных интервалов.
5. Повторите шаги 3 и 4 для последовательности длиной в 10000 чисел.
6. Изучите тест перестановок (он описан на третьей странице) и проверьте этим тестом первые 9999 чисел вашей последовательности, разбив их на тройки.

Используйте уровень значимости 5%.

Методы генерации псевдослучайных чисел и проверяющие по вариантам:

Степенной остаточный №2.

Назначаем начальное число $z_1 < 10000$. Последующие числа получаем из соотношения $z_i = ((z_{i-1}+17)^{2.2} \operatorname{div} 100) \bmod 10000$, div — целая часть от деления, mod — остаток от деления.

Полученная последовательность укладывается в пределы от 0 до 1 так: $x_i = \dfrac{z_i}{10000}$.

1. Требуется рассчитать 100 псевдослучайных чисел данным методом, z1 = 7724

Напишем функцию, создающую требуемый массив чисел на языке Matlab:

makestat.m

```
function stat = makestat(stat, size, first)
    stat(1) = first;
for i = 2:size
    prevval = stat(length(stat));
    newval = mod(fix(power(prevval + 17, 2.2)/100), 10000);
    stat(length(stat)+1) = newval;
end
```

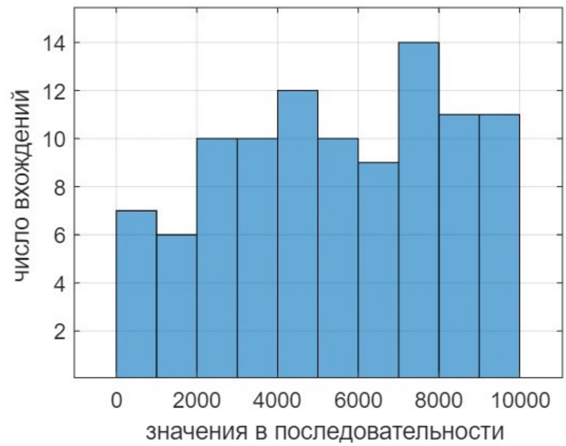
Теперь создадим переменную для массива и вызовем функцию для 100 элементов и первого числа 7724

```
>> stat = [];
>> stat = makestat(stat, 100, 7724)
```

2. Выведем первые 10 чисел последовательности:

```
>> stat(1:10)

ans =

      7724        2142        6449        7678        5348        3442         507        9605        6739        2669
```

3. Построим гистограмму из 10 столбцов для полученной последовательности:




```
>> histogram(stat, 10)
```

4. Проверим гипотезу о том, что последовательность имеет распределение R(0,1) критерием хи-квадрат.
Для этого разобьём интервал [0, 1) на десять равных интервалов $[x_i; x_{i+1})$, рассчитаем для равномерного распределения частоты попадания чисел в каждый интервал - η_i, затем уложим последовательность в нужный интервал с помощью описанной в степенном остаточном методе формулы и вычислим частоты попадания чисел в интервалы для чисел нормализованной последовательности - n_i.

Составим гипотезы:
H0: $\forall i \;\; n_i = \eta_i$ - основная гипотеза - частоты соответствуют частотам R(0,1)
HA: $\exists j$: $n_j \neq \eta_j$ - альтернативная гипотеза, любое отклонение от H0

Проверять гипотезу будем с помощью статистики:

$\chi^2 = \sum_{j=1}^{10} (n_j - \eta_j)^2/\eta_j \sim H0 \sim \chi^2(k-1)$, - распределение хи-квадрат с k-1 степенями свободы, где n_j - наблюдаемые частоты, η_j - ожидаемые частоты, k- число наблюдений = 10 в нашем случае (равно числу интервалов)

Мы должны отвергнуть гипотезу H0, если рассчитанная статистика будет больше значения квантили $\chi^2(k-1)$ уровня α, где α - уровень значимости, в нашем случае равен 0.05

Найдём в таблице критическую квантиль $\chi^2(k-1)$ α: $\chi^2(9)\ 0.05 = 16.919$

Теперь рассчитаем статистику. Для этого построим интервалы [xi; xi+1):
[0;0.1), [0.1;0.2), [0.2;0.3), [0.3;0.4), [0.4;0.5), [0.5;0.6), [0.6;0.7), [0.7;0.8), [0.8;0.9), [0.9;1),
вычислим ожидаемые частоты как (длина последовательности / число интервалов) = 100/10 = 10

Для нахождения наблюдаемых частот уложим последовательность в интервал [0,1) по формуле ns_i = s_i/10000 :

```
>> normstat = normalizestat(stat)

normstat =

  Columns 1 through 20

    0.7724    0.2142    0.6449    0.7678    0.5348    0.3442    0.0507    0.9605

  Columns 21 through 40

    0.0577    0.2656    0.6256    0.1755    0.0158    0.0860    0.9826    0.3691
```

normalizestat.m | countpermutations.m | countdiff.m

```
function normstat = normalizestat(stat)
    normstat = zeros(1, length(stat));
for i = 1:length(stat)
    normstat(i) = stat(i)/10000;
end
```

Для каждого из 10 равных интервалов найдём, сколько чисел в него попало

```
>> observedfreq = countfrequences(normstat)

observedfreq =

     7     6    10    10    12    10     9    14    11    11
```

countfrequences.m | normalizestat.m | countpermutations.m | countdiff.m | makesta

```
function freq = countfrequences(stat)
    freq = [0,0,0,0,0,0,0,0,0,0];
    for i = 1:length(stat)
        freq(fix(stat(i)*10)+1) = freq(fix(stat(i)*10)+1) +1;
    end
end
```

Теперь для каждого интервала найдём (nj- ηj)^2/ηj и просуммируем:

xi	xi+1	ηi	ni
0	0.1	10	7
0.1	0.2	10	6
0.2	0.3	10	10
0.3	0.4	10	10
0.4	0.5	10	12
0.5	0.6	10	10
0.6	0.7	10	9
0.7	0.8	10	14
0.8	0.9	10	11
0.9	1	10	11

countpermutations.m | makestat.m * | countdiff.m

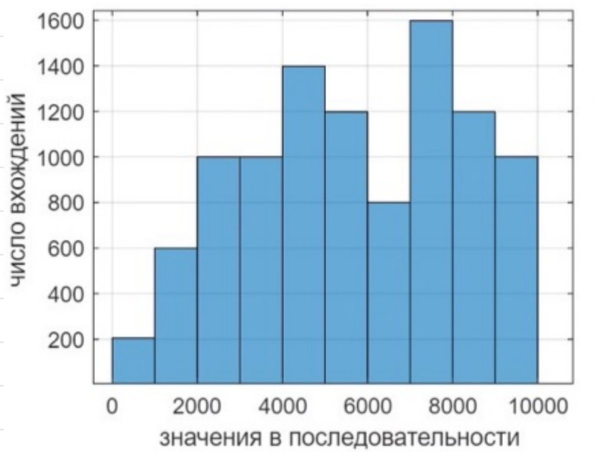
```
function res = countdiff(expected, observed)
    res = 0;
for i = 1:length(expected)
    res = res+ power((observed(i) - expected(i)), 2)/expected(i);
end
```

```
normalfreq =

    10    10    10    10    10    10    10    10    10    10

>> res = countdiff(normalfreq, observedfreq)

res =

    4.8000
```

χ^2 = 4.8
Сравним с критической квантилью χ^2(9) 0.05 = 16.919
4.8 < 16.919   χ^2 < χ^2(5) 0.05

На уровне значимости 5% мы не отвергаем гипотезу H0, и считаем что полученная последовательность из 100 чисел имеет распределение R(0,1)

5. Сделаем то же самое для последовательности длиной 10000 элементов:
Сгенерируем последовательность и построим гистограмму:




```
>> stat10000 = [];
>> stat10000 = makestat(stat10000, 10000, 7724);
>> histogram(stat10000, 10)
```

Проверим гипотезу о том, что последовательность имеет распределение R(0,1) критерием хи-квадрат.
Для этого разобьём интервал [0, 1) на десять равных интервалов $[x_i; x_{i+1})$, рассчитаем для R(0,1) частоты попадания чисел в каждый интервал - η_i, затем уложим последовательность в нужный интервал с помощью описанной в степенном остаточном методе формулы и вычислим частоты попадания чисел в интервалы для чисел нормализованной последовательности - n_i.

Составим гипотезы:
$H0: \forall i \;\; n_i = \eta_i$ - основная гипотеза - частоты соответствуют частотам R(0,1)
$HA: \exists j: n_j \neq \eta_j$ - альтернативная гипотеза, любое отклонение от H0

Проверять гипотезу будем с помощью статистики:

$\chi^2 = \sum_{j=1}^{10}(n_j - \eta_j)^2/\eta_j \sim H0 \sim \chi^2(k-1)$, - распределение хи-квадрат с k-1 степенями свободы, где n_j - наблюдаемые частоты, η_j - ожидаемые частоты, k- число наблюдений = 10 в нашем случае (равно числу интервалов)

Мы должны отвергнуть гипотезу H0, если рассчитанная статистика будет больше значения квантили $\chi^2(k-1)$ уровня α, где α - уровень значимости, в нашем случае равен 0.05

Найдём в таблице критическую квантиль $\chi^2(k-1)$ α: $\chi^2(9)$ 0.05 = 16.919

Теперь рассчитаем статистику. Для этого построим интервалы $[x_i; x_{i+1})$, вычислим ожидаемые частоты как длина последовательности / число интервалов = 10000/10 = 1000

Для нахождения наблюдаемых частот уложим последовательность в интервал [0,1):

```
>> normstat10000 = normalizestat(stat10000)

normstat10000 =

  Columns 1 through 20

    0.7724    0.2142    0.6449    0.7678    0.5348    0.3442
```

Для каждого из 10 равных интервалов найдём, сколько чисел в него попало

```
>> observedfreq10000 = countfrequences(normstat10000)

observedfreq10000 =

         205          600         1000         1000         1398         1198          801         1598         1199         1001

>> normalfreq10000 = [1000,1000,1000,1000,1000,1000,1000,1000,1000,1000]

normalfreq10000 =

        1000         1000         1000         1000         1000         1000         1000         1000         1000         1000
```

Теперь для каждого интервала найдём (nj- ηj)^2/ηj и просуммируем:

xi	xi+1	ηi	ni
0	0.1	1000	205
0.1	0.2	1000	600
0.2	0.3	1000	1000
0.3	0.4	1000	1000
0.4	0.5	1000	1398
0.5	0.6	1000	1198
0.6	0.7	1000	801
0.7	0.8	1000	1598
0.8	0.9	1000	1199
0.9	1	1000	1001

```
>> res = countdiff(normalfreq10000, observedfreq10000)

res =

   1.4264e+03
```

$\chi^2 = 1426.4$

Сравним с критической квантилью $\chi^2(9)\ 0.05 = 16.919$

$1426.4 > 16.919$ $\chi^2 > \chi^2(5)\ 0.05$

На уровне значимости 5% мы отвергаем гипотезу H0, и считаем что полученная последовательность из 10000 чисел не имеет распределение R(0,1)

6. Тест перестановок. Посчитаем количество перестановок каждого вида в последовательности из 9999 первых элементов полученной последовательности и проверим с помощью критерия хи-квадрат гипотезу о том, что числа независимы - об этом свидетельствует равновероятность появления каждого типа перестановок. Пусть $x_1, \dots x_k$ - перестановки из 3 чисел, $p_1, \dots p_k$ - вероятности их появления.

Составим гипотезы:
H_0: $p_1=p_2=p_3=p_4=p_5=p_6=1/6$ - основная гипотеза - перестановки равновероятны
H_A: $\exists\, j$: $p_j \neq 1/6$ - альтернативная гипотеза, любое отклонение от H_0

Проверять гипотезу будем с помощью статистики:

$\chi^2 = \sum_{j=1}^{k}(O_j - E_j)^2/E_j \sim H_0 \sim \chi^2(k-1)$, - распределение хи-квадрат с k-1 степенями свободы, где O_j - наблюдаемые частоты, E_j - ожидаемые частоты, k- число наблюдений = 6 в нашем случае

Мы должны отвергнуть гипотезу H_0, если рассчитанная статистика будет больше значения квантили $\chi^2(k-1)$ уровня α, где α - уровень значимости, в нашем случае равен 0.05

Найдём в таблице критическую квантиль $\chi^2(k-1)$ α: $\chi^2(5)$ 0.05 = 11.07

Теперь вычислим статистику.
Напишем функцию, которая будет разбивать последовательность на тройки и подсчитывать частоту каждой перестановки:

countpermutations.m * × makestat.m ×

```
function permutations = countpermutations(stat)
    permutations = [0,0,0,0,0,0];
for i = 1:length(stat)/3
    first = stat(3*i-2);
    second = stat(3*i-1);
    third = stat(3*i);

    if first>second
        if second> third
            permutations(6) = permutations(6)+1;
        elseif first>third
                permutations(5) = permutations(5)+1;
        else
             permutations(3) = permutations(3)+1;
        end
    else
       if second> third
          if first>third
               permutations(4) = permutations(4)+1;
          else
               permutations(2) = permutations(2)+1;
          end
       else
            permutations(1) = permutations(1)+1;
       end
    end
end
```

```
%permutations[1] - 123
%permutations[2] - 132
%permutations[3] - 213
%permutations[4] - 231
%permutations[5] - 312
%permutations[6] - 321
```

Вызвав эту функцию от последовательности получаем значения O1, ... O6

```
>> statpermutations = countpermutations(stat10000(1:9999))

statpermutations =

   601   667   467   465   666   467
```

Соответствующие Ej для всех j = 9999/18 = 555.5

Вычислим статистику $\chi^2 = \sum_{j=1}^{6}(O_j - E_j)^2/E_j$:

```
>> normalpermutations = [555.5, 555.5, 555.5, 555.5, 555.5, 555.5];
>> countdiff(normalpermutations, statpermutations)

ans =

   91.0306
```

$\chi^2 = 91.0306$ - значение статистики

Сравним χ^2 и критическую квантиль $\chi^2(5)$ 0.05 = 11.07

$91.0306 > 11.07$ $\chi^2 > \chi^2(5)$ 0.05

На уровне значимости 5% мы отвергаем гипотезу H0 о том, что перестановки распределены равновероятно

Согласно тесту перестановок получаем, что на уровне значимости 5% числа в последовательности зависимы.