

Лабораторная работа 4, ТВМС

Бочарников Андрей, М3238

Ковешников Глеб, М3238

Шишкин Алексей, М3238

23 апреля 2020 г.

Формулировка

Для случайной величины, распределенной по нормальному закону с параметрами (a, σ^2) , выполнить следующие действия:

1. Задать параметры распределения $X \sim N(a, \sigma^2)$.
 2. Построить выборку генеральной совокупности X .
 3. Построить график гистограммы.
 4. Проверить гипотезу о виде распределения по критерию хи-квадрат.
- Аналогично для $X \sim U(a, b)$ - равномерно распределенной на $[a, b]$ случайной величины.

Входные данные

- Размер выборки для построения гистограммы: $n = 10^6$
- Размер выборки для проверки критерия χ^2 : $n = 10^4$
- Параметры нормального распределения: $\sigma = 1, \mu = 1$
- Параметры равномерного распределения: $a = 20, b = 80$
- $\alpha = 0.05$
- Количество тестов – 10^3

Программа 1

Нормальное распределение.

В первой части программы строится гистограмма с использованием функции *hist* и выводится на экран вместе с графиком функции распределения.

Во второй части производится 4 запуска тестов:

1. Выборка генерируется с помощью нормального распределения; нижняя граница на количество элементов попавших в интервал отсутствует; проверяется гипотеза о нормальном распределении; оценивается вероятность ошибки первого рода.
2. Выборка генерируется с помощью нормального распределения; соседние интервалы объединяются, чтобы количество элементов попавших на каждый интервал было не меньше 6; проверяется гипотеза о нормальном распределении; оценивается вероятность ошибки первого рода.
3. Всё делается аналогично второму пункту, но математическое ожидание и среднеквадратичное отклонение увеличиваются на 0.005. Считается ошибка второго рода.
4. Всё делается аналогично пункту 3, но математическое ожидание и среднеквадратичное отклонение увеличиваются на 0.05. Считается ошибка второго рода.

3.1 Исходный код

```
pkg load statistics  
  
clc ;
```

```

clear all;

function res = test_Chi2_1(tests, n, m)
    res = 0;
    mu = 1;
    sigma = 1;
    alpha = 0.05;
    for t = 1 : tests
        X = normrnd(mu, sigma, n, 1);

        l = min(X);
        r = max(X);

        delta = (r - l) / m;
        cnt_in_bucket = hist(X, m);

        #Выборочное среднее
        E = mean(X);
        #Выборочная отклонение
        Sqrt_D = std(X);

        P = [];
        for i = 1 : m
            P(i) = normcdf(l + delta * i, E, Sqrt_D) - normcdf(l + delta * (i - 1),
                E, Sqrt_D);
        endfor

        hi2 = sum(((cnt_in_bucket - n .* P) .^ 2) ./ (n .* P));
        res = res + (hi2 >= chi2inv(1 - alpha, m - 1 - 2));
    endfor
    printf("Normal_distribution_satisfies_the_hypothesis_about_normal_
        distribution\n")
    printf("For_alpha=%d, probability_of_type_I_error_is_%d\n", alpha, res /
        tests)
endfunction

function res = test_Chi2_2(tests, n, m)
    res = 0;
    mu = 1;
    sigma = 1;
    alpha = 0.05;
    for t = 1 : tests
        X = normrnd(mu, sigma, n, 1);

        l = min(X);
        r = max(X);

        delta = (r - l) / m;
        cnt_in_bucket = hist(X, m);
        walls = [];
        for i = 1 : m
            cur_l = (i - 1) * delta + l;
            cur_r = cur_l + delta;
            walls(i, 1) = cur_l;
            walls(i, 2) = cur_r;
        endfor

        #f means fixed, nj >= 6
        f_cnt_in_bucket = [];
        f_walls = [];
        f_m = 0;
        for i = 1 : m
            if (i == 1 || f_cnt_in_bucket(f_m) >= 6)

```

```

        f_m = f_m + 1;
        f_walls(f_m, 1) = walls(i, 1);
        f_cnt_in_bucket(f_m) = 0;
    endif
    f_walls(f_m, 2) = walls(i, 2);
    f_cnt_in_bucket(f_m) = f_cnt_in_bucket(f_m) + cnt_in_bucket(i);
endfor

#Выборочное среднее
E = mean(X);
#Выборочная отклонение
SQRT_D = std(X);

P = [];
for i = 1 : f_m
    P(i) = normcdf(f_walls(i, 2), E, SQRT_D) - normcdf(f_walls(i, 1), E,
        SQRT_D);
endfor

hi2 = sum((f_cnt_in_bucket - n .* P) .^ 2) ./ (n .* P);
res = res + (hi2 >= chi2inv(1 - alpha, f_m - 1 - 2));
endfor
printf("Normal_distribution_satisfies_the_hypothesis_about_normal_
distribution\n")
printf("Sample_grouped_to_met_n_j>=6\n")
printf("For_alpha=%d, probability_of_type_I_error_is_%d\n", alpha, res /
tests)
endfunction

function res = test_Chi2_3(tests, n, m, d)
    res = 0;
    mu = 1;
    sigma = 1;
    alpha = 0.05;
    for t = 1 : tests
        X = normrnd(mu, sigma, n, 1);

        l = min(X);
        r = max(X);

        delta = (r - l) / m;
        cnt_in_bucket = hist(X, m);
        walls = [];
        for i = 1 : m
            cur_l = (i - 1) * delta + l;
            cur_r = cur_l + delta;
            walls(i, 1) = cur_l;
            walls(i, 2) = cur_r;
        endfor

        #f means fixed, nj >= 6
        f_cnt_in_bucket = [];
        f_walls = [];
        f_m = 0;
        for i = 1 : m
            if (i == 1 || f_cnt_in_bucket(f_m) >= 6)
                f_m = f_m + 1;
                f_walls(f_m, 1) = walls(i, 1);
                f_cnt_in_bucket(f_m) = 0;
            endif
            f_walls(f_m, 2) = walls(i, 2);
            f_cnt_in_bucket(f_m) = f_cnt_in_bucket(f_m) + cnt_in_bucket(i);
        endfor
    endfor

```

```

#Выборочное среднее
E = mean(X) + d;
#Выборочная отклонение
SQRT_D = std(X) + d;

P = [];
for i = 1 : f_m
    P(i) = normcdf(f_walls(i, 2), E, SQRT_D) - normcdf(f_walls(i, 1), E,
        SQRT_D);
endfor

hi2 = sum((f_cnt_in_bucket - n .* P) .^ 2) ./ (n .* P);
res = res + (hi2 < chi2inv(1 - alpha, f_m - 1 - 2));
endfor
printf("Normal_distribution_satisfies_the_hypothesis_about_normal_
distribution\n")
printf("Sample_mean_and_variance_changed_by_%d\n", d)
printf("For_alpha=%d, probability_of_type_II_error_is_%d\n", alpha, res /
    tests)
endfunction

n = 10 ^ 6;
mu = 1;
sigma = 1;
m = 10 ^ 2;
tests = 10 ^ 3;

X = normrnd(mu, sigma, n, 1);

l = min(X);
r = max(X);
delta = (r - l) / m;
[y_coords x_coords] = hist(X, m);

x_coords_for_normpdf = 1:0.1:r;
bar(x_coords, y_coords / (n * delta));
hold on;
plot(x_coords_for_normpdf, normpdf(x_coords_for_normpdf, mu, sigma));

printf("Sample_size=%d\n", n)
printf("Length_of_intervals=%d\n", delta)
printf("Number_of_intervals=%d\n", m)

printf("\n")

# PART 2

test_Chi2_1(10 ^ 3, 10 ^ 4, m);

printf("\n")

test_Chi2_2(10 ^ 3, 10 ^ 4, m);

printf("\n")

test_Chi2_3(10 ^ 3, 10 ^ 4, m, 0.005);

printf("\n")

test_Chi2_3(10 ^ 3, 10 ^ 4, m, 0.05);

printf("\n")

```

3.2 Выходные данные

График гистограммы - график 1 в приложении после вывода.

Размер выборки = 1000000

Выбранная длина интервалов = 0.0965317

Количество интервалов = 100

Нормальное распределение проходит проверку гипотезы о нормальном распределении

Для $\alpha = 0.05$, вероятность ошибки первого рода получается 0.307

Нормальное распределение проходит проверку гипотезы о нормальном распределении

Данные сгруппированы, чтобы выполнялось $n_j \geq 6$

Для $\alpha = 0.05$, вероятность ошибки первого рода получается 0.065

Нормальное распределение проходит проверку гипотезы о нормальном распределении

Выборочное среднее и выборочная дисперсия изменены на 0.005

Для $\alpha = 0.05$, вероятность ошибки второго рода получается 0.955

Нормальное распределение проходит проверку гипотезы о нормальном распределении

Выборочное среднее и выборочная дисперсия изменены на 0.05

Для $\alpha = 0.05$, вероятность ошибки второго рода получается 0.001

Программа 2

Равномерное распределение.

В первой части программы строится гистограмма с использованием функции *hist* и выводится на экран вместе с графиком функции распределения.

Во второй части производится три запуска тестов на проверку гипотез:

1. Выборка генерируется с помощью равномерного распределения; нижняя граница на количество элементов попавших в интервал отсутствует; проверяется гипотеза о равномерном распределении; оценивается вероятность ошибки первого рода.
2. Всё делается аналогично пункту 1, но сдвигаются границы распределения на 0.05 влево и вправо. Оценивается вероятность ошибки второго рода.
3. Всё делается аналогично пункту 1, но сдвигаются границы распределения на 3 влево и вправо. Оценивается вероятность ошибки второго рода.

4.1 Исходный код

```
pkg load statistics

clc;
clear all;

function res = test_Chi2_1(tests, n, m)
    res = 0;
    a = 20;
    b = 80;
    alpha = 0.05;
    for t = 1 : tests
        X = unifrnd(a, b, n, 1);

        l = min(X);
        r = max(X);

        delta = (r - l) / m;
        cnt_in_bucket = hist(X, m);
```

```

P = [];
for i = 1 : m
    P(i) = unifcdf(l + i * delta, l, r) - unifcdf(l + (i - 1) * delta, l, r);
endfor

hi2 = sum(((cnt_in_bucket - n .* P) .^ 2) ./ (n .* P));
res = res + (hi2 >= chi2inv(1 - alpha, m - 1 - 2));
endfor
printf("Uniform_distribution_satisfies_the_hypothesis_about_uniform_
distribution\n");
printf("For_alpha=%d, probability_of_type_I_error_is_%d\n", alpha, res /
tests);
endfunction

function res = test_Chi2_2(tests, n, m, d)
    res = 0;
    a = 20;
    b = 80;
    alpha = 0.05;
    for t = 1 : tests
        X = unifrnd(a, b, n, 1);

        l = min(X);
        r = max(X);

        delta = (r - l) / m;
        cnt_in_bucket = hist(X, m);

        P = [];
        for i = 1 : m
            P(i) = unifcdf(l + i * delta, l - d, r + d) - unifcdf(l + (i - 1) *
                delta, l - d, r + d);
        endfor

        hi2 = sum(((cnt_in_bucket - n .* P) .^ 2) ./ (n .* P));
        res = res + (hi2 < chi2inv(1 - alpha, m - 1 - 2));
    endfor
    printf("Uniform_distribution_satisfies_the_hypothesis_about_uniform_
distribution\n");
    printf("Left_and_right_borders_changed_by_%d\n", d)
    printf("For_alpha=%d, probability_of_type_II_error_is_%d\n", alpha, res /
        tests)
endfunction

n = 10 ^ 6;
m = 10 ^ 2;
a = 20;
b = 80;

# PART 1

X = unifrnd(a, b, n, 1);

l = min(X);
r = max(X);
delta = (r - l) / m;
[y_coords x_coords] = hist(X, m);

real_y = 1 / (b - a);
bar(x_coords, y_coords / (n * delta));
hold on;
plot([a b], [real_y real_y], "linewidth", 1);

```

```

printf("Sample_size_=%d\n", n);
printf("Length_of_intervals_=%d\n", delta);
printf("Number_of_intervals_=%d\n", m);

printf("\n");

# PART 2

test_Chi2_1(10 ^ 3, 10 ^ 4, m);

printf("\n");

test_Chi2_2(10 ^ 3, 10 ^ 4, m, 0.05);

printf("\n");

test_Chi2_2(10 ^ 3, 10 ^ 4, m, 3);

printf("\n");

```

4.2 Выходные данные

График гистограммы - график 2 после вывода.

Размер выборки = 1000000

Выбранная длина интервалов = 0.599997

Количество интервалов = 100

Равномерное распределение проходит проверку гипотезы о равномерном распределении

Для $\alpha = 0.05$, вероятность ошибки первого рода получается 0.068

Равномерное распределение проходит проверку гипотезы о равномерном распределении

Левая и правая граница изменены на 0.05

Для $\alpha = 0.05$, вероятность ошибки второго рода получается 0.933

Равномерное распределение проходит проверку гипотезы о равномерном распределении

Левая и правая граница изменены на 3

Для $\alpha = 0.05$, вероятность ошибки второго рода получается 0

Вывод

1. Как видно по рисунку, график гистограммы хорошо приближает график плотности распределения.
2. В случае равномерного распределения вероятность ошибки первого рода теста по критерию хи-квадрат для $\alpha = 0.05$, сходится к 0.05. В случае нормального распределения с целью получения корректных результатов потребовалось объединение соседних интервалов, чтобы на каждый попадало минимум 6 элементов выборки.
3. Для оценки вероятности ошибки 2 рода было проведено две проверки: с относительно небольшим и заметным изменением выборочных параметров. В первом случае по критерию хи-квадрат получается что ложная гипотеза принимается с вероятностью 0.955 и 0.933 в нормальном и равномерном случаях соотв., то есть хи квадрат не замечает разницы после изменения параметром на такое маленькое значение, во втором – вероятность что ложная гипотеза принимается близка к нулю – критерий хи квадрат "заметил" изменения параметра.



