Лабораторная работа 4

Борис Шапошников

Виктор Шарепо

Рамазан Рахматуллин

Обратная связь: bshaposhnikov01@gmail.com

Часть 1

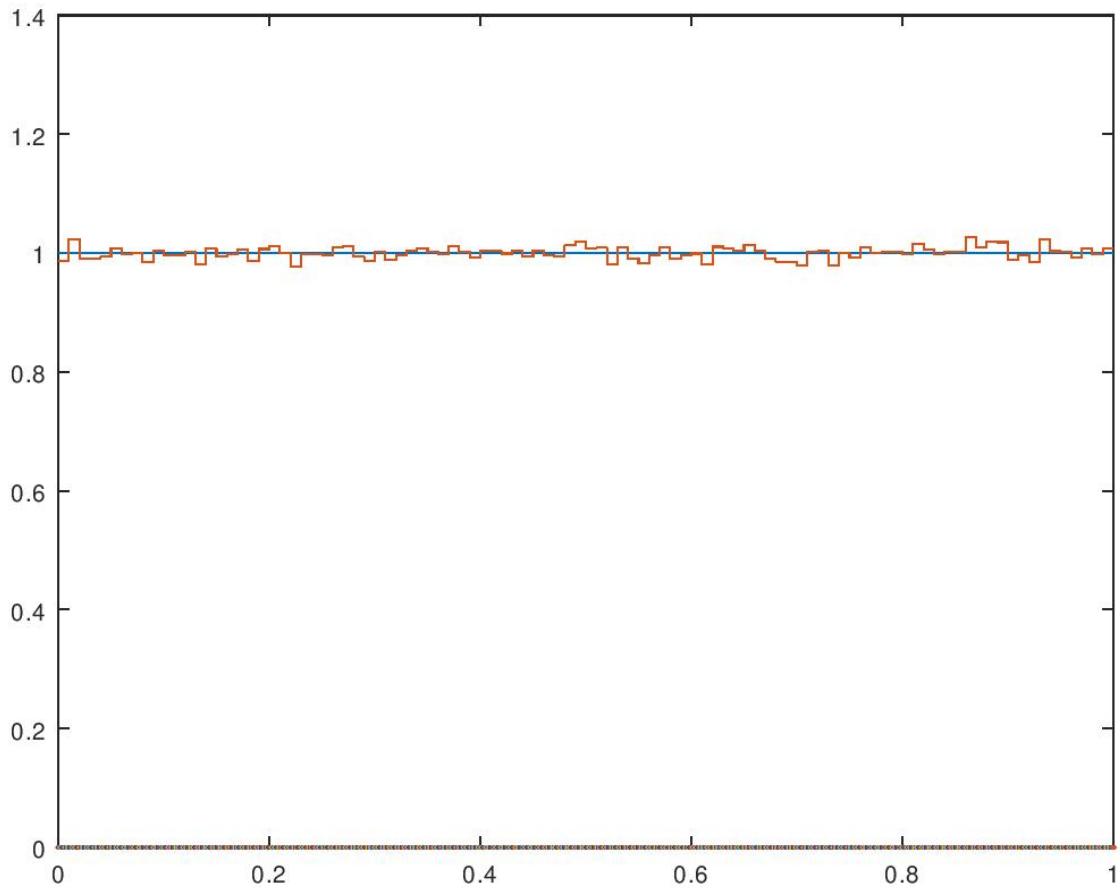
- Для равномерного и нормального распредлений выбрать параметры и построить графики плотности.
- Произвести выборку $n = 10^6$ из соответствующего распределения.
- Для $m = 100$ построить гистограмму.
- Объединить графики плотности и гистограмму на одном графике.

Код и графики:

- Равномерное распределение

```
clc;
clear;

pkg load statistics;
n = 1e6;
m = 1e2;

a = 0;
b = 1;
x = sort(unifrnd(a, b, n, 1));
y = hist(x, m);
y = y / n;

```

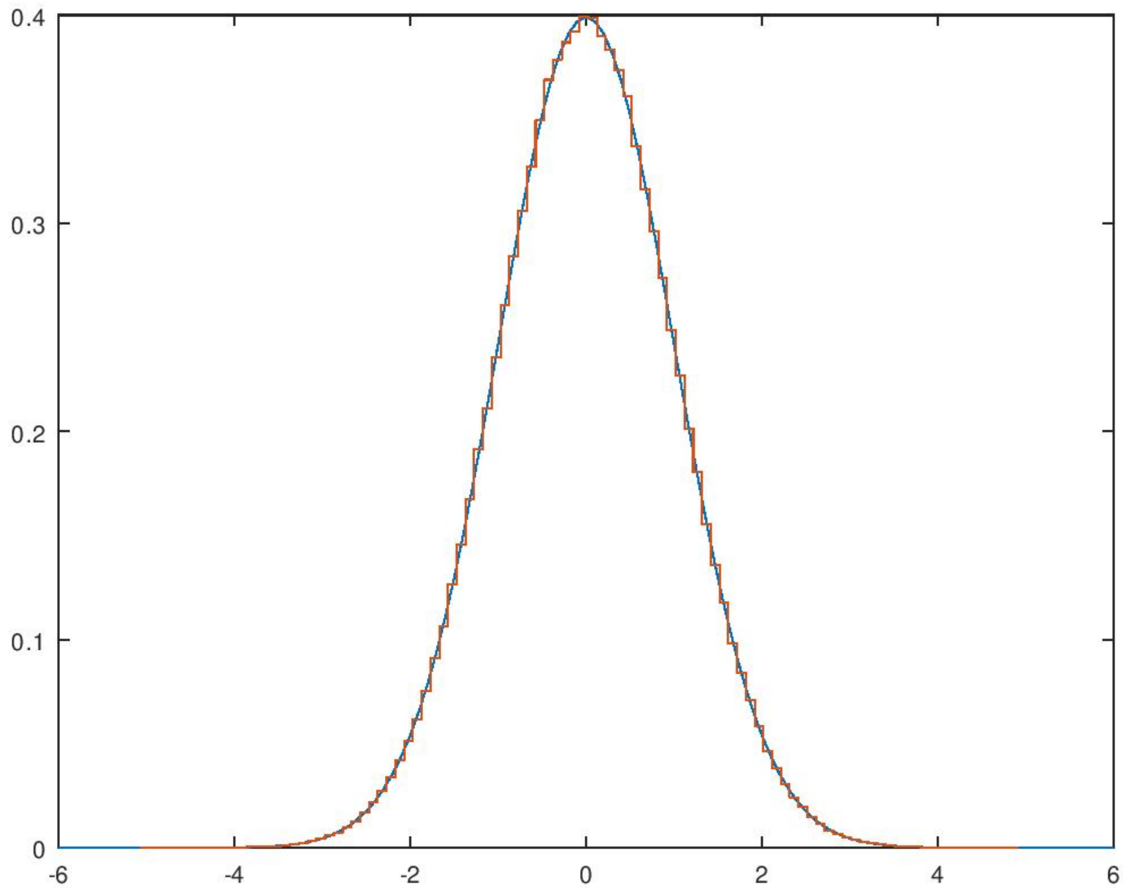
```
step = (x(n) - x(1)) / m
f = y / step

[stx, sty] = stairs(x(1) : step : x(end), [f, f(end)]);
interval = a:0.001:b
plot(interval, unifpdf(interval, a, b), stx, sty, interval, 0)
```



- Нормальное распределение

```
clc;
clear;

pkg load statistics;
n = 1e6;
m = 1e2;


x = sort(randn(1, n));
y = hist(x, m);
```

```
y = y / n;

step = (x(n) - x(1)) / m
f = y / step

[stx, sty] = stairs(x(1) : step : x(end), [f, f(end)]);
plot(-6:0.1:6, normpdf(-6:0.1:6), stx, sty)
```



Вывод из первой части

Построив гистограмму, мы получили эмпирическую оценку плотности распределения случайной величины. По графикам мы убедились, что гистограмма, построенная на основе выборки из распределения, достаточно хорошо приближает его плотность вероятности.

Часть 2

Проверка гипотез согласия с H_0 - тип распределения.

Цель работы - оценка ошибки первого рода.

1. Для $n = 10^6$ провести $m = 100$ проверок гипотез по критерию χ^2. Построенную статистику сравнить с порогом.

   - Равномерное распределение

```
clc;
clear;

pkg load statistics;

n = 1e6;
m = 1e2;

a = 0;
b = 1;

errorsI = 0;
gamma = 0.95;

for i = 1:m
  x = sort(unifrnd(a, b, n, 1));
  mi = min(x);
  ma = max(x);

  y = hist(x, m);
  step = (x(n) - x(1)) / m;

  y1 = unifcdf(x(1) : step : x(end) - step, mi, ma);
  y2 = unifcdf(x(1) + step : step : x(end), mi, ma);

  p = y2 - y1;
  hi2 = sum(((y - n * p) .^ 2) ./ (n * p));
  threshold = chi2inv(gamma, m - 3);
  if (hi2 >= threshold)
    errorsI++;
  endif
endfor

```

```
printf("Alpha: %d\n", 1 - gamma);
printf("Estimated alpha: %d\n", errorsI / m);
```

Результат

Alpha: 0.05

Estimated alpha: 0.05

- Нормальное распределение

```
clc;
clear;

pkg load statistics;

n = 1e6;
m = 1e2;

errorsI = 0;
gamma = 0.95;

for i = 1:m
  x = sort(normrnd(1, 3, n, 1));

  y = hist(x, m);
  step = (x(n) - x(1)) / m;

  E = mean(x);
  sigma = std(x);

  y1 = normcdf(x(1) : step : x(end) - step, E, sigma);
  y2 = normcdf(x(1) + step : step : x(end), E, sigma);

  p = y2 - y1;
  hi2 = sum(((y - n * p) .^ 2) ./ (n * p));
  threshold = chi2inv(gamma, m - 3);
  if (hi2 >= threshold)
    errorsI++;
  endif
endfor

printf("Alpha: %d\n", 1 - gamma);
printf("Estimated alpha: %d\n", errorsI / m);
```

Результат

```
Alpha: 0.05
Estimated alpha: 0.19
```

Вывод

Для равномерного распределения оценка $\alpha = \frac{H_1 count}{m}$ очень близка к теоретической $\alpha = 1 - \gamma$. Для нормального распределения это не так. Для него не всегда выполнено требоване $n_j >= 6$ для близких к $a = min(x)$ и $b = max(x)$ интервалов. ${\chi^2}_{n,m-1} = \sum_{j=1}^{m} \frac{(n_j - n*p_{j,0})^2}{n*p_{j,0}}$

2. Для нормального распределения провести объединение соседних интервалов для выполнения $n_j >= 6$.

```
clc;
clear;

pkg load statistics;

n = 1e6;
m = 1e2;

errorsI = 0;
gamma = 0.95;

for i = 1:m
  x = sort(normrnd(1, 3, n, 1));

  y = hist(x, m);
  step = (x(n) - x(1)) / m;

  E = mean(x);
  sigma = std(x);

  l = x(1)
  r = x(1) + step

  i = 1;
  hi2 = 0;
```

```
  seg_cnt = 0;
  while (i <= m)
    y_i = y(i);
    while (y_i < 6 && i++ < m)
      y_i += y(i);
      r += step
    endwhile
    y1 = normcdf(l, E, sigma);
    y2 = normcdf(r, E, sigma);
    p = y2 - y1;
    hi2 += (y_i - n * p) ^ 2 / (n * p);
    l = r;
    r += step;
    i++;
    seg_cnt++;
  endwhile

  threshold = chi2inv(gamma, seg_cnt - 3);
  if (hi2 >= threshold)
    errorsI++;
  endif
endfor

printf("Alpha: %d\n", 1 - gamma);
printf("Estimated alpha: %d\n", errorsI / m);
```

Результат

Alpha: 0.05

Estimated alpha: 0.04

Вывод

При объединении соседних интервалов для выполнения $n_j >= 6$ для нормального распределения оценка α близка к теоретической.

3. Испортить выборочные параметры и провести оценку β.

- Равномерное распределение

  Делаем изменение вида $U(a - eps, b + eps)$ для различных eps из

0..0.005.

```
clc;
clear;

pkg load statistics;

n = 1e6;
m = 1e2;

a = 0;
b = 1;

gamma = 0.95;

for eps = 0 : 0.001 : 0.005
  errorsII = 0;
  for i = 1:m
    x = sort(unifrnd(a, b, n, 1));
    mi = min(x) - eps;
    ma = max(x) + eps;

    y = hist(x, m);
    step = (x(n) - x(1)) / m;

    y1 = unifcdf(x(1) : step : x(end) - step, mi, ma);
    y2 = unifcdf(x(1) + step : step : x(end), mi, ma);

    p = y2 - y1;
    hi2 = sum(((y - n * p) .^ 2) ./ (n * p));
    threshold = chi2inv(gamma, m - 3);
    if (hi2 < threshold)
      errorsII++;
    endif
  endfor

  printf("Estimated beta for eps = %d: %d\n", eps, errorsII / m);
endfor
```

Результат

Estimated beta for eps = 0: 0.9

Estimated beta for eps = 0.001: 0.9

Estimated beta for eps = 0.002: 0.63

Estimated beta for eps = 0.003: 0.28

Estimated beta for eps = 0.004: 0

Estimated beta for eps = 0.005: 0

- Нормальное распределение

  Делаем изменение вида $N(a + eps, \sigma)$ для различных eps из 0..0.05

```
clc;
clear;

pkg load statistics;

n = 1e6;
m = 1e2;

gamma = 0.95;

for eps = 0 : 0.01 : 0.05
  errorsII = 0;

  for i = 1:m
    x = sort(normrnd(1, 3, n, 1));

    y = hist(x, m);
    step = (x(n) - x(1)) / m;

    E = mean(x) + eps;
    sigma = std(x);

    y1 = normcdf(x(1) : step : x(end) - step, E, sigma);
    y2 = normcdf(x(1) + step : step : x(end), E, sigma);

    p = y2 - y1;
    hi2 = sum(((y - n * p) .^ 2) ./ (n * p));
    threshold = chi2inv(gamma, m - 3);
    if (hi2 < threshold)
      errorsII++;
```

```
    endif
  endfor

  printf("Estimated beta for eps = %d: %d\n", eps, errorsII / m);
endfor
```

Результат

Estimated beta for eps = 0: 0.9

Estimated beta for eps = 0.01: 0.61

Estimated beta for eps = 0.02: 0.02

Estimated beta for eps = 0.03: 0

Estimated beta for eps = 0.04: 0

Estimated beta for eps = 0.05: 0

Вывод

Оценив ошибки второго рода для различных *eps*, убеждаемся в том, что при малых изменениях ошибка второго рода почти не меняется, однако при больших изменениях *eps* критерий χ^2 резко реагирует на это (ошибка второго рода с некоторого значения *eps* резко стремится к 0).