Лабораторная работа 3

Борис Шапошников

Виктор Шарепо

Рамазан Рахматуллин

Обратная связь: bshaposhnikov01@gmail.com

Часть 1

Для случайной величины, распределенной по нормальному закону с параметрами (a, σ^2) и по равномерному распределению с параметрами (a, b), выполнить следующие действия.

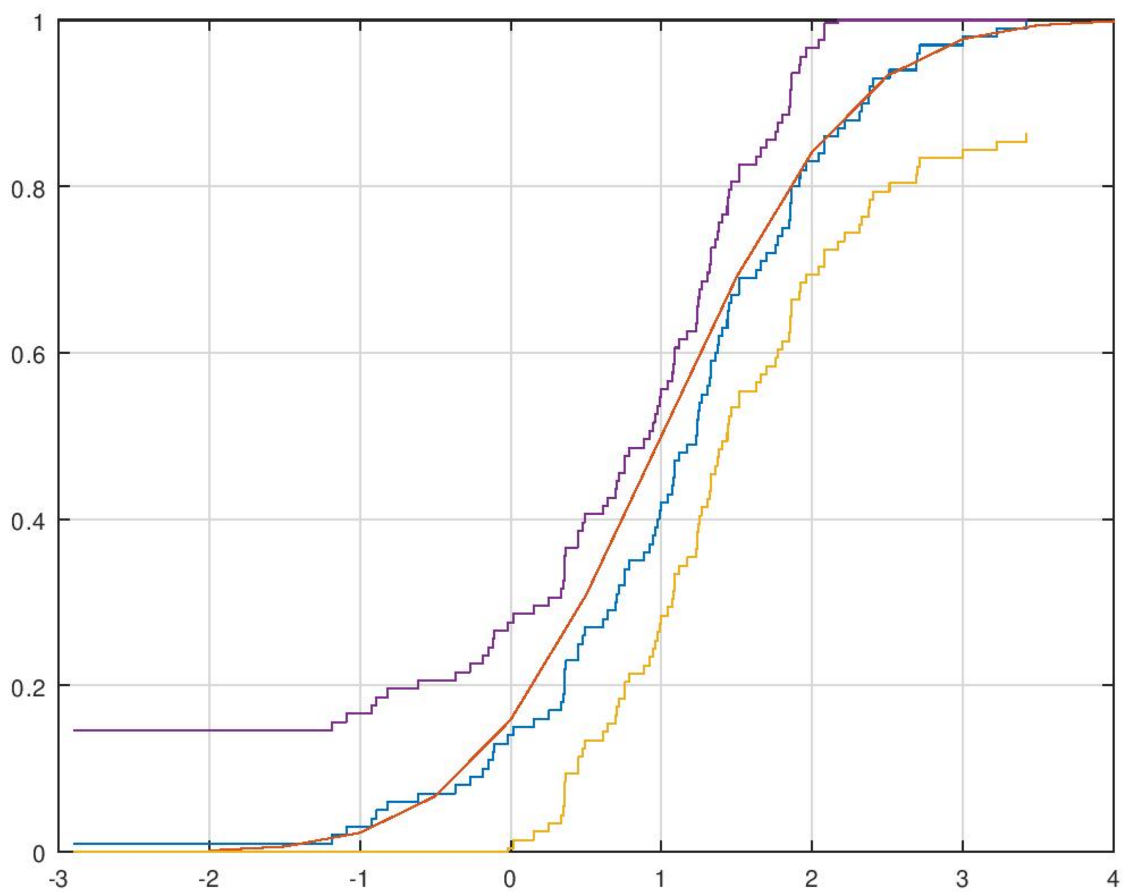
- Задать параметры распределения $X \sim N(a, \sigma^2)$, $X \sim U(a, b)$.
- Построить график $F_x(x)$, используя функцию normcdf.
- При n=100 построить выборку из генеральной совокупности X.
- По построенной выборке построить график эмпирической функции распределения $F_n(x)$, используя при построении встроенную функцию [a,b]=stairs(x,y) для построения кусочно- постоянной функции. Учесть при построении, что $F_n(x)$ изменяется на 1/n в каждой следующей точке выборки.
- Построить доверительную полосу надежности $\gamma = 0.95; u(\gamma) = 1.36$ (см. пособие стр.92-96).
- На этом же графике построить $F_n(x)$ и $F_x(x)$. Убедится, что функция распределения попадает (?) в доверительную полосу.

Код и графики:

- Нормальное распределение

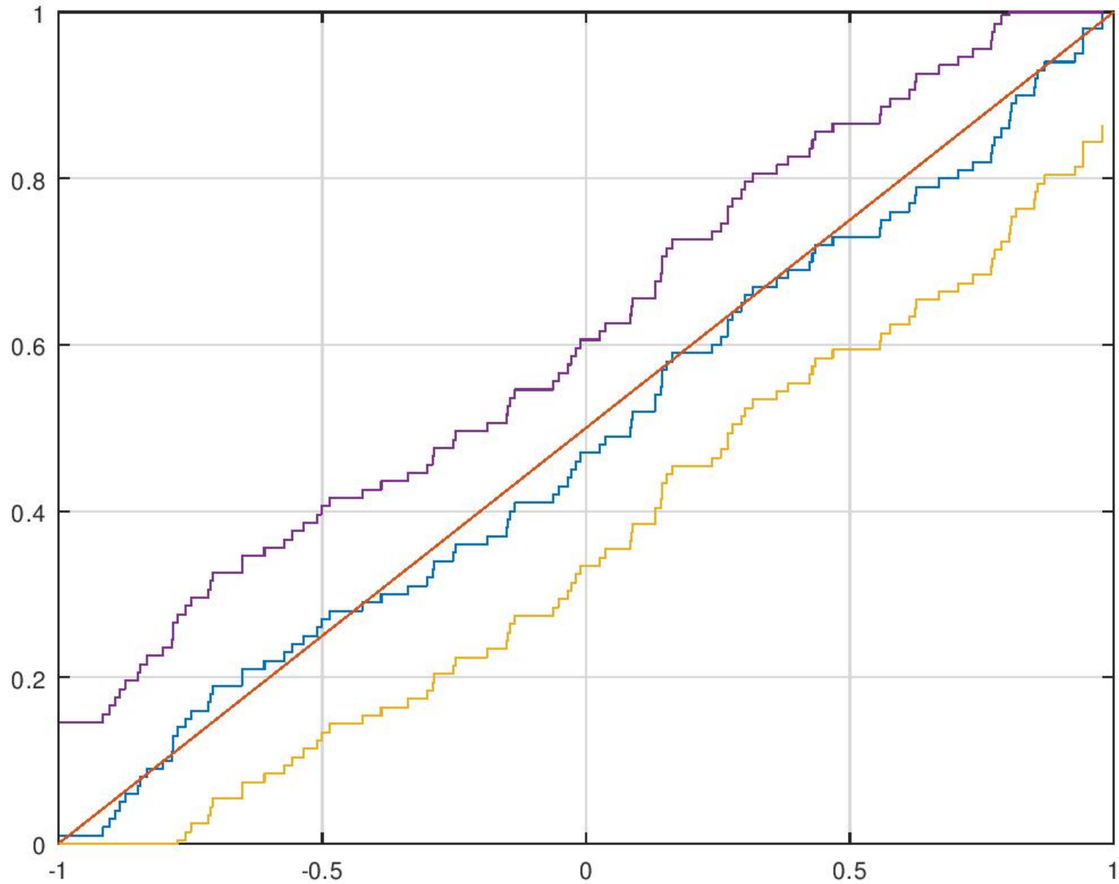
```
pkg load statistics;
clc;
clear;

```

```
E = 1;
sigma = 1;

n = 100;
m = 1;

v = sort(normrnd(E, sigma, n, m));

Fn = 1 / n : 1 / n : 1;
[a, b] = stairs(v, Fn);

x = E - 3 * sigma : 0.5 : E + 3 * sigma;
F = normcdf(x, E, sigma);

d = 1.36 / sqrt(n);
b1 = max(0, b - d);
b2 = min(1, b + d);
plot(a, b, x, F, a, b1, a, b2), grid;
```



- Равномерное распределение

```
pkg load statistics;
clc;
clear;

x = -1;
y = 1;

n = 100;
m = 1;
v = sort(unifrnd(x, y, n, m));

Fn = 1 / n : 1 / n : 1;
Fx = -1 : 1 / n : 1;
F = unifcdf(Fx, x, y);
[a, b] = stairs(v, Fn);


d = 1.36 / sqrt(n);

b1 = max(0, b - d);
b2 = min(1, b + d);
plot(a, b, Fx, F, a, b1, a, b2), grid;
```



Вывод из первой части: реальная функция распределения попала в доверительную полосу.

Часть 2

На основе критерия Колмогорова (если сможете, и на основе критерия Смирнова) провести проверку гипотез при $n = 10^4$ и $n = 10^6$ (см. пособие стр. 96-97).

- Нормальное распределение

```
pkg load statistics;
clc;
clear;

E = 1;
sigma = 1;
n = 10^4;
Fn = 1 / n : 1 / n : 1;
```

```

fprintf("Selection: %d\nI type error\n\n", n);
for m = 100:100:1000
  fprintf("m = %d\n", m);
  v = sort(normrnd(E, sigma, n, m));
  F = normcdf(v, E, sigma);
  diff = F - Fn';
  sup = max(max(abs(diff), abs(diff - 1 / n)));
  kolmogorov = sqrt(n) * sup;
  k = mean(kolmogorov > 1.36);

  smirnov = 1 / (12*n) + sum((diff + 1 / (2*n)) .^ 2);
  s = mean(smirnov > 0.46);
  fprintf("Kolmogorov: %d\nSmirnov: %d\n\n", k, s);
endfor

fprintf("===================================\n")
fprintf("Selection: %d\nII type error\n\n", n);
v = sort(normrnd(E, sigma, n, m));
for a = 0:0.01:0.1
  fprintf("addition = %d\n", a);
  F = normcdf(v, E, sigma + a);
  diff = F - Fn';
  sup = max(max(abs(diff), abs(diff - 1 / n)));
  kolmogorov = sqrt(n) * sup;
  k = 1 - mean(kolmogorov > 1.36);

  smirnov = 1 / (12*n) + sum((diff + 1 / (2*n)) .^ 2);
  s = 1 - mean(smirnov > 0.46);
  fprintf("Kolmogorov: %d\nSmirnov: %d\n\n", k, s);
endfor

n = 10^6;
Fn = 1 / n : 1 / n : 1;

fprintf("===================================\n")
fprintf("===================================\n")
fprintf("Selection: %d\nI type error\n\n", n);
for m = 100:100:1000
  fprintf("m = %d\n", m);
  v = sort(normrnd(E, sigma, n, m));
  F = normcdf(v, E, sigma);
  diff = F - Fn';
```

```
  sup = max(max(abs(diff), abs(diff - 1 / n)));
  kolmogorov = sqrt(n) * sup;
  k = mean(kolmogorov > 1.36);

  smirnov = 1 / (12*n) + sum((diff + 1 / (2*n)) .^ 2);
  s = mean(smirnov > 0.46);
  fprintf("Kolmogorov: %d\nSmirnov: %d\n\n", k, s);
endfor

fprintf("==================================\n")
fprintf("Selection: %d\nII type error\n\n", n);
v = sort(normrnd(E, sigma, n, m));
for a = 0:0.01:0.1
  fprintf("addition = %d\n", a);
  F = normcdf(v, E, sigma + a);
  diff = F - Fn';
  sup = max(max(abs(diff), abs(diff - 1 / n)));
  kolmogorov = sqrt(n) * sup;
  k = 1 - mean(kolmogorov > 1.36);

  smirnov = 1 / (12*n) + sum((diff + 1 / (2*n)) .^ 2);
  s = 1 - mean(smirnov > 0.46);
  fprintf("Kolmogorov: %d\nSmirnov: %d\n\n", k, s);
endfor
```

Вывод программы см. в приложенном файле normoutput.txt.

- Равномерное распределение

```
pkg load statistics;
clc;
clear;

n = 10^4;
Fn = 1 / n : 1 / n : 1;
x = -1;
y = 1;
Fx = -1 : 2 / n : 1;

fprintf("Selection: %d\nI type error\n\n", n);
for m = 100:100:1000
  fprintf("m = %d\n", m);
  v = sort(unifrnd(x, y, n, m));
  F = unifcdf(v, x, y);
  diff = F - Fn';
```

```
  sup = max(max(abs(diff), abs(diff - 1 / n)));
  kolmogorov = sqrt(n) * sup;
  k = mean(kolmogorov > 1.36);

  smirnov = 1 / (12*n) + sum((diff + 1 / (2*n)) .^ 2);
  s = mean(smirnov > 0.46);
  fprintf("Kolmogorov: %d\nSmirnov: %d\n\n", k, s);
endfor

fprintf("===================================\n")
fprintf("Selection: %d\nII type error\n\n", n);
v = sort(unifrnd(x, y, n, m));
for a = 0:0.01:0.1
  fprintf("addition = %d\n", a);
  F = unifcdf(v, x + 3 * a, y + 3 * a);
  diff = F - Fn';
  sup = max(max(abs(diff), abs(diff - 1 / n)));
  kolmogorov = sqrt(n) * sup;
  k = 1 - mean(kolmogorov > 1.36);

  smirnov = 1 / (12*n) + sum((diff + 1 / (2*n)) .^ 2);
  s = 1 - mean(smirnov > 0.46);
  fprintf("Kolmogorov: %d\nSmirnov: %d\n\n", k, s);
endfor

n = 10^6;
Fn = 1 / n : 1 / n : 1;

fprintf("===================================\n")
fprintf("===================================\n")
fprintf("Selection: %d\nI type error\n\n", n);
for m = 100:100:1000
  fprintf("m = %d\n", m);
  v = sort(unifrnd(x, y, n, m));
  F = unifcdf(v, x, y);
  diff = F - Fn';
  sup = max(max(abs(diff), abs(diff - 1 / n)));
  kolmogorov = sqrt(n) * sup;
  k = mean(kolmogorov > 1.36);

  smirnov = 1 / (12*n) + sum((diff + 1 / (2*n)) .^ 2);
  s = mean(smirnov > 0.46);
  fprintf("Kolmogorov: %d\nSmirnov: %d\n\n", k, s);
```

```
endfor

fprintf("===================================\n")
fprintf("Selection: %d\nII type error\n\n", n);
v = sort(unifrnd(x, y, n, m));
for a = 0:0.001:0.01
  fprintf("addition = %d\n", a);
  F = unifcdf(v, x + a, y + a);
  diff = F - Fn';
  sup = max(max(abs(diff), abs(diff - 1 / n)));
  kolmogorov = sqrt(n) * sup;
  k = 1 - mean(kolmogorov > 1.36);

  smirnov = 1 / (12*n) + sum((diff + 1 / (2*n)) .^ 2);
  s = 1 - mean(smirnov > 0.46);
  fprintf("Kolmogorov: %d\nSmirnov: %d\n\n", k, s);
endfor
```

Вывод программы см. в приложенном файле unioutput.txt.

Вывод из второй части:
Эмпирическая функция распределения хорошо оценивает реальную функцию распределения. Реальная функция распределения попадает в доверительный интервал. Ошибки первого рода (гипотеза, верна, но отвергается), как показывают тесты, для эмпирической функции распределения, случаются редко и при совпадении параметров основной функции распределения и той, из которой делаем выборку, верная гипотеза почти никогда не отвергается. При изменении параметров функции распределения, из которой делаем выборку, относительно основной, критерий согласия Колмогорова и Смирнова начинают резко отвергать гипотезу.