Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Отчет

по лабораторной работе №7

**ПОЛУЧЕНИЕ ИНТЕРВАЛЬНЫХ ОЦЕНОК ПАРАМЕТРОВ РАСПРЕДЕЛЕНИЙ**

Выполнили: Проверил:

ст. гр. 820603 Ярмолик В.И.

Дрозд В. А.

Ермаков Т. А.

Минск 2021

# Цель работы

Изучение задачи получения интервальных оценок параметров распределений.

Приобретение навыков получения интервальных оценок параметров распределений в системе *Matlab*.

# Теоретические сведения

Доверительный интервал для математического ожидании *α* нормальной генеральной совокупности при известной дисперсии

|  |  |
| --- | --- |
|  | (1) |

где

|  |  |
| --- | --- |
|  |  |
|  |  |

Доверительный интервал для математического ожидании *α* нормальной генеральной совокупности при неизвестной дисперсии

|  |  |
| --- | --- |
|  | (2) |

где

|  |  |  |
| --- | --- | --- |
|  |  |  |

Доверительный интервал для дисперсии нормальной генеральной совокупности при известном математическом ожидании *α*

|  |  |
| --- | --- |
|  | (3) |

где

|  |  |
| --- | --- |
|  |  |

Доверительный интервал для дисперсии нормальной генеральной совокупности при неизвестном математическом ожидании *α*

|  |  |
| --- | --- |
|  | (4) |

Пусть выполнено *n* независимых испытаний Бернулли, в результате которых событие *А* появилось *m* раз. Границы доверительного интервала для вероятности события *A* определяются выражением

|  |  |
| --- | --- |
|  | (5) |

Приведем средства *Matlab* для получения интервальных оценок параметров распределений:

*x = norminv(p, mu, sigma)* возвращает значение аргумента функции нормального распределения с математическим ожиданием *mu* и среднеквадратическим отклонением *sigma* по значению *p* функции распределения.

*x = chi2inv(p, n)* возвращает значение аргумента функции распределения хи-квадрат с *n* степенями свободы по значению *p* функции распределения.

*x = tinv(p, n)* возвращает значение аргумента функции распределения Стьюдента с *n*степенями свободы по значению *p* функции распределения.

*x = finv(p, m, n)* возвращает возвращает значение аргумента функции распределения Фишера с *m, n* степенями свободы по значению *p* функции распределения.

# Порядок выполнения работы

Смоделируем выборку из нормального распределения с самостоятельно выбранными значениями параметров и получим доверительные интервалы (1) ‒ (4) для этих параметров, создав собственные *m*-файлы-функции.

Ниже представлен код функции, позволяющей получить из нормального распределения выборочное среднее значения *xm*, выборочную дисперсию *s20* при известном математическом ожидании и выборочную дисперсию *s2* при неизвестном математическом ожидании.

*function [xm, s20, s2] = customnormfit(x,alpha)*

*xm = mean(x);*

*sum20 = 0;*

*sum2 = 0;*

*n = length(x);*

*for i = 1 : n*

*sum20 = sum20 + (x(i) - alpha) ^ 2;*

*sum2 = sum2 + (x(i) - xm) ^ 2;*

*end*

*s20 = sum20 / length(x);*

*s2 = sum2 / length(x);*

*end*

Для получения интервальных оценок параметров нормального распределения напишем исполняемый *m*-файл, который сформирует необходимые данные для анализа. Приведем код основной программы:

*n = 1000;*

*alpha = 2; sigma = 4;*

*x = ones(1,n);*

*for i = 1 : n*

*x(i) = normrnd(alpha,sigma);*

*end*

*[xm,s20,s2]=customnormfit(x,alpha);*

*y = [0.90 0.95 0.975];*

*for i = 1 : length(y)*

*%mathexpect if dispersion is known*

*confInt = norminv(1 - (1 - y(i)) / 2,0,1) \* ((sigma / n) ^ ( 1 / 2));*

*result(1,1) = xm - confInt;*

*result(1,2) = xm + confInt;*

*%mathexpect if dispersion is unknown*

*confInt = tinv(1 - (1 - y(i)) / 2,n) \* ((s2 / (n - 1)) ^ (1 / 2)) ;*

*result(2,1) = xm - confInt;*

*result(2,2) = xm + confInt;*

*%dispersion if mathexpect is known*

*result(3,1) = n \* s2 / chi2inv(1 - ( 1 - y(i)) / 2,n);*

*result(3,2) = n \* s2 / chi2inv(1 - ( 1 + y(i)) / 2,n);*

*%dispersion if mathexpect is unknown*

*result(4,1) = n \* s20 / chi2inv(1 - (1 - y(i)) / 2,n - 1);*

*result(4,2) = n \* s20 / chi2inv(1 - (1 + y(i)) / 2,n - 1);*

*[muhat,sigmahat,muci,sigmaci] = normfit(x,1 - y(i));*

*%Вывод:*

*fprintf('Matlab: ');*

*fprintf('%f\n',y(i));*

*fprintf('muhat: %.4f\n', muhat);*

*fprintf('sigmahat: %.4f\n', sigmahat);*

*fprintf('muci: %.4f\t%.4f\n', muci(1), muci(2));*

*fprintf('sigmaci: %.4f\t%.4f\n', sigmaci(1),sigmaci(2));*

*fprintf('sigmaci^2: %.4f\t%.4f\n', sigmaci(1)^2,sigmaci(2)^2);*

*fprintf('custom\_func: ');*

*fprintf('%f\n',y(i));*

*fprintf('I(a,sig)\t%.4f\t%.4f\n', result(1,1),result(1,2));*

*fprintf('I(a,s)\t%.4f\t%.4f\n', result(2,1),result(2,2));*

*fprintf('I(sig2,s02)\t%.4f\t%.4f\n', result(3,1),result(3,2));*

*fprintf('I(sig2,s2)\t%.4f\t%.4f\n', result(4,1),result(4,2));*

*end*

Покажем сравнение интервальных оценок параметров, найденных при помощи собственного алгоритма и средств *Matlab*.

В таблице 1 показано сравнение интервальных оценок при объеме выборки 1000.

Таблица 1 – Сравнение интервальных оценок нормального распределения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Собственный алгоритм | | *Matlab* | |
|  | | | |
|  |  |  |  |
|  | 1.8385 | 2.0466 | – | – |
|  | 1.7341 | 2.1509 | 1,7341 | 2,1509 |
|  | 14.8929 | 17.2544 | – | – |
|  | 14.9104 | 17.2759 | 14,9073 | 17,2723 |
|  |  | | | |
|  |  |  |  |
|  | 1.8186 | 2.0665 | – | – |
|  | 1.6941 | 2.1909 | 1,6941 | 2,1909 |
|  | 14.6899 | 17.5061 | – | – |
|  | 14.7070 | 17.5281 | 14,7040 | 17,5244 |
|  |  | | | |
|  |  |  |  |
|  | 1.8008 | 2.0843 | – | – |
|  | 1.6584 | 2.2267 | 1.6584 | 2,2267 |
|  | 14.5117 | 17.7351 | – | – |
|  | 14.5285 | 17.7574 | 14.5255 | 17.7537 |

Смоделируем выборку из биномиального распределения с самостоятельно выбранными значениями параметров и получим доверительный интервал (5) для вероятности *p* случайного события.

Приведем код основной программы:

*n = 1000;*

*m = 40;*

*p = m / n;*

*q = 1 - p;*

*y = [ 0.90 0.95 0.975];*

*for i = 1 : length(y)*

*confInt = norminv(1 - (1 - y(i)) / 2,0,1) \* ((p \* q / n) ^ ( 1 / 2));*

*result(1,1) = p - confInt;*

*result(1,2) = p + confInt;*

*[phat,pci] = binofit(m,n,1-y(i));*

*%Вывод:*

*fprintf('Matlab: ');*

*fprintf('%f\n',y(i));*

*fprintf('phat: %.4f\n', phat);*

*fprintf('pci: %.4f\t%.4f\n', pci(1),pci(2));*

*fprintf('custom\_func: ');*

*fprintf('%f\n',y(i));*

*fprintf('pci\t%.4f\t%.4f\n',result(1,1),result(1,2));*

*end*

В таблице 2 показаны интервальные оценки для биномиального распределения.

Таблица 2 − Сравнение интервальных оценок биномиального распределения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Собственный алгоритм | | *Matlab* | |
|  | | | |
|  |  |  |  |
|  | 0.0298 | 0.0502 | 0.0303 | 0.0518 |
|  |  | | | |
|  |  |  |  |
|  | 0.0279 | 0.0521 | 0.0287 | 0.0541 |
|  |  | | | |
|  |  |  |  |
|  | 0.0261 | 0.0539 | 0.0273 | 0.0562 |

# Выводы

В ходе выполнения лабораторной работы были изучены методы получения интервальных оценок параметров распределений, а также приобретены навыки получения интервальных оценок параметров распределений в системе *Matlab*. Полученные собственным способом интервальные оценки практически совпадают с оценками, полученными средствами *Matlab*.