*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Изображение выглядит как эмблема, герб, нашивка, символ  Автоматически созданное описание** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(национальный исследовательский университет)***  ***(МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ФУНДАМЕНТАЛЬНЫЕ НАУКИ\_\_\_\_\_\_\_\_

**КАФЕДРА** ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И МАТЕМАТИЧЕСКАЯ ФИЗИКА (ФН11)\_

**НАПРАВЛЕНИЕ ПОДГОТОВКИ** МАТЕМАТИКА И КОМПЬЮТЕРНЫЕ НАУКИ (02.03.01)

**Отчет**

**по лабораторной работе № \_4\_**

**Название лабораторной работы: Интервальные оценки для параметров биноминального закона**

**Вариант № 2**

**Дисциплина:** Теория вероятности и математическая статистика

Студент группы ФН11-52Б **\_\_\_\_\_\_\_\_\_\_\_\_\_ Кожемякин Г.А.**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_ Облакова Т.В.**

(Подпись, дата) (И.О. Фамилия)

Москва, 2024

**Задание**

Используя выборку, сгенерированную вами в задаче 2 и считая параметр неизвестным (дано), постройте для уровней доверия , 0.95 и 0.98 симметричные интервальные оценки Клоппера-Пирсона для вероятности успеха в одном испытании .

Для тех же уровней найдите по ЦПТ приближенные доверительные интервалы для .

Сравните полученные результаты и убедитесь, что полученные интервалы содержат истинное значение параметра.

Для одного из значений постройте совмещенные графики функций распределения биномиальных законов , , .

Сформулируйте выводы.

**Исходные данные**

p = 0.4, k = 10, n = 120

Выборка: X = [5, 3, 3, 6, 3, 2, 3, 3, 4, 6, 3, 3, 4, 2, 5, 5, 7, 5, 1, 3, 4, 3, 5, 2, 3, 4, 4, 5, 1, 5, 3, 4, 3, 6, 2, 1, 2, 6, 2, 5, 1, 2, 6, 5, 2, 4, 5, 1, 4, 6, 7, 3, 5, 3, 4, 3, 3, 3, 2, 7, 4, 6, 4, 5, 3, 5, 2, 3, 3, 2, 3, 5, 3, 6, 3, 5, 7, 1, 4, 3, 2, 6, 5, 4, 4, 4, 4, 8, 4, 4, 3, 3, 4, 6, 3, 5, 4, 3, 4, 2, 6, 3, 4, 5, 3, 3, 7, 5, 4, 2, 5, 4, 5, 1, 5, 6, 4, 6, 3, 4]

**Решение задачи**

Пусть выборка из биномиального закона .

Интервальная оценка Клоппера-Пирсона строится на основе статистики , теоретическая функция распределения которой биномиальная, то есть имеет вид

Найдем при каждом такие значения и , что

Тогда .

Значения и представляют собой квантили уровней и для биномиального закона .

По условию параметр p подразумевается неизвестным, в то время как значение k=10 считается. В результате моделирования получено значение выборочного среднего и определена статистика К, представляющая собой суммарное число успехов: , что соответствует .

Ниже представлены графики квантилей для в зависимости от р и отмечено значение статистики K, полученное выше.

Изображение выглядит как текст, линия, График, диаграмма

Автоматически созданное описание

Численно найдем точки пересечения графиков квантилей с горизонтальной прямой К=466 и получим приближенные значения .

k = 10  
n = len(X) *# n = 120***def** func\_alpha(p):  
 **return** binom.ppf(alpha / 2, n \* k, p) - K  
  
  
**def** func\_1\_minus\_alpha(p):  
 **return** binom.ppf(1 - alpha / 2, n \* k, p) - K  
  
*# Численное нахождение точек пересечения квантилей с К=466*result\_alpha = root\_scalar(func\_alpha, bracket=[0, 1], method=**'bisect'**)  
result\_1\_minus\_alpha = root\_scalar(func\_1\_minus\_alpha, bracket=[0, 1], method=**'bisect'**)  
  
p\_alpha = result\_alpha.root  
p\_1\_minus\_alpha = result\_1\_minus\_alpha.root  
  
print(**f'Границы доверительного интервала для К=466: {**p\_1\_minus\_alpha**}, {**p\_alpha**}'**)

Полученные значения .

Далее, согласно общему принципу построения доверительных интервалов, множество надо эквивалентным образом переписать в виде , что приводит к уравнениям Клоппера-Пирсона для и:

Уравнения (3)-(4) можно решать с использованием неполной бэта-функции

связанной с биномиальным законом формулой:

Тогда уравнения (3)-(4) перепишутся в виде:

Решим уравнения Клоппера – Пирсона с использованием встроенной в модуле scipy Python функции beta. В качестве ее параметров задаем уровень и для .

*# Решение уравнений Клоппера-Пирсона для разных уровней значимости*alphas = [0.1, 0.05, 0.02]  
**for** a **in** alphas:  
 lower\_bound = beta.ppf(a / 2, K, n \* k - K + 1)  
 upper\_bound = beta.ppf(1 - a / 2, K + 1, n \* k - K)  
 print(**f'Решение уравнений Клоппера-Пирсона, alpha={**a**}: {**lower\_bound**}, {**upper\_bound**}'**)

Полученные границы интервалов:

Решение уравнений Клоппера-Пирсона, =0.1: 0.3649961603817708, 0.4120768678314399

Решение уравнений Клоппера-Пирсона, =0.05: 0.360643933610191, 0.41657144632431164

Решение уравнений Клоппера-Пирсона, =0.02: 0.3556032016402592, 0.4218084678016434

Далее найдем границы доверительных интервалов для тех же значений , что и при решении уравнений Клоппера – Пирсона, но теперь с использованием центральной предельной теоремы.

alphas = [0.1, 0.05, 0.02]  
*# Решение с использованием ЦПТ***for** a **in** alphas:  
 sigma = np.sqrt(x\_mean \* (k - x\_mean) / (n \* k)) / k  
 z\_alpha = norm.ppf(a / 2)  
 lower\_bound\_clt = x\_mean / k + z\_alpha \* sigma  
 upper\_bound\_clt = x\_mean / k - z\_alpha \* sigma  
 print(**f'Границы интервалов по ЦПТ, alpha={**a**}: {**lower\_bound\_clt**}, {**upper\_bound\_clt**}'**)

Полученные границы интервалов:

Границы интервалов по ЦПТ, =0.1: 0.3651915738256997, 0.4114750928409669

Границы интервалов по ЦПТ, =0.05: 0.3607582260832659, 0.4159084405834007

Границы интервалов по ЦПТ, =0.02: 0.3556035012376193, 0.4210631654290473

Полученные результаты задачи нахождения границ доверительных интервалов для различных уровней значимости с использованием уравнений Клоппера – Пирсона и центральной предельной теоремы представим в таблице ниже.

|  |  |  |
| --- | --- | --- |
| Уровень доверия | Доверительный интервал для р | |
| По Клопперу - Пирсону | По ЦПТ |
| 0,9 | (0,365; 0,412) | (0,365; 0,411) |
| 0,95 | (0,361; 0,417) | (0,361; 0,416) |
| 0,99 | (0,357; 0,422) | (0,356; 0,421) |
| Истинное значение р | р = 0,4 | |

Из таблицы следует, что с увеличением уровня доверия доверительные интервалы для параметра р расширяются и при этом всегда содержат истинное значение параметра. Как следствие, график истинной функции распределения F(x, p) расположен между графиками F(x,) и F(x,), что проиллюстрировано рисунком ниже.

Изображение выглядит как текст, линия, диаграмма, снимок экрана

Автоматически созданное описание

Можно также заметить, что разница между доверительными интервалами Клоппера – Пирсона и приближенными доверительными интервалами очень незначительна, что объясняется большим объемом выборки.

**Выводы:** в ходе проделанной лабораторной работы были получены доверительные интервалы для значения параметра р биноминального распределения с помощью двух методов: по Клопперу – Пирсону и с использованием центральной предельной теоремы. Полученные результаты согласуются с истинным значением параметра р, границы доверительных интервалов найденные разными методами мало отличаются ввиду большого объема выборки, а сами доверительные интервалы становятся шире с увеличением степени доверия. Таким образом, показана хорошая применимость данных методов для нахождения границ доверительных интервалов для оценки значения неизвестного параметра распределения.

**Приложение**

В приложении представлен программный код на языке Python, с помощью которого была выполнена лабораторная работа.

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**from** scipy.stats **import** binom  
**from** scipy.optimize **import** root\_scalar  
**from** scipy.stats **import** beta  
**from** scipy.stats **import** norm  
  
  
**def** func\_alpha(p):  
 **return** binom.ppf(alpha / 2, n \* k, p) - K  
  
  
**def** func\_1\_minus\_alpha(p):  
 **return** binom.ppf(1 - alpha / 2, n \* k, p) - K  
  
  
*# Выборка X*X = [5, 3, 3, 6, 3, 2, 3, 3, 4, 6, 3, 3, 4, 2, 5, 5, 7, 5, 1, 3, 4, 3, 5, 2, 3, 4, 4, 5, 1, 5, 3, 4, 3, 6, 2, 1, 2, 6, 2, 5, 1, 2, 6, 5, 2, 4, 5, 1, 4, 6, 7, 3, 5, 3, 4, 3, 3, 3, 2, 7, 4, 6, 4, 5, 3, 5, 2, 3, 3, 2, 3, 5, 3, 6, 3, 5, 7, 1, 4, 3, 2, 6, 5, 4, 4, 4, 4, 8, 4, 4, 3, 3, 4, 6, 3, 5, 4, 3, 4, 2, 6, 3, 4, 5, 3, 3, 7, 5, 4, 2, 5, 4, 5, 1, 5, 6, 4, 6, 3, 4]  
  
k = 10  
n = len(X)  
print(n)  
  
*# Выборочное среднее*x\_mean = np.mean(X)  
print(**f'Выборочное среднее: {**x\_mean**}'**)  
  
*# Статистика К - суммарное число успехов*K = np.sum(X)  
print(**f'Суммарное число успехов: {**K**}'**)  
print(**f'Проверка: {**x\_mean \* n**}'**)  
  
*# Генерация квантилей*alpha = 0.01  
p = np.linspace(0, 1, 1000)  
quantiles\_alpha = binom.ppf(alpha, n \* k, p)  
quantiles\_1\_minus\_alpha = binom.ppf(1 - alpha, n \* k, p)  
  
*# Построение графика*plt.figure(figsize=(8, 6))  
plt.plot(p, quantiles\_alpha, label=**r'$u(p, \alpha)$'**)  
plt.plot(p, quantiles\_1\_minus\_alpha, label=**r'$u(p, 1-\alpha)$'**)  
plt.axhline(K, color=**'black'**, linestyle=**'--'**)  
plt.text(0.05, K + 20, **f'K={**K**}'**, fontsize=12, verticalalignment=**'center'**)  
plt.title(**r'Квантили биноминального распределения для $\alpha=0.01$'**, fontsize=14)  
plt.legend()  
plt.grid(**True**)  
plt.show()  
  
*# Численное нахождение точек пересечения квантилей с К=466*result\_alpha = root\_scalar(func\_alpha, bracket=[0, 1], method=**'bisect'**)  
result\_1\_minus\_alpha = root\_scalar(func\_1\_minus\_alpha, bracket=[0, 1], method=**'bisect'**)  
  
p\_alpha = result\_alpha.root  
p\_1\_minus\_alpha = result\_1\_minus\_alpha.root  
  
print(**f'Границы доверительного интервала для К=466: {**p\_1\_minus\_alpha**}, {**p\_alpha**}'**)  
*# Вывод - ОК, т.к. истинное значение p=0.4 опадает в интервал  
  
# Решение уравнений Клоппера-Пирсона для разных уровней значимости*alphas = [0.1, 0.05, 0.02]  
**for** a **in** alphas:  
 lower\_bound = beta.ppf(a / 2, K, n \* k - K + 1)  
 upper\_bound = beta.ppf(1 - a / 2, K + 1, n \* k - K)  
 print(**f'Решение уравнений Клоппера-Пирсона, alpha={**a**}: {**lower\_bound**}, {**upper\_bound**}'**)  
  
*# Решение с использованием ЦПТ***for** a **in** alphas:  
 sigma = np.sqrt(x\_mean \* (k - x\_mean) / (n \* k)) / k  
 z\_alpha = norm.ppf(a / 2)  
 lower\_bound\_clt = x\_mean / k + z\_alpha \* sigma  
 upper\_bound\_clt = x\_mean / k - z\_alpha \* sigma  
 print(**f'Границы интервалов по ЦПТ, alpha={**a**}: {**lower\_bound\_clt**}, {**upper\_bound\_clt**}'**)  
  
  
*# Функции распределения B(k, p), B(k, p1), B(k, p2)*p = 0.4  
p1 = 0.365  
p2 = 0.412  
x\_values = np.linspace(0, k, 1000)  
  
binom\_cdf\_p = binom.cdf(x\_values, k, p)  
binom\_cdf\_p1 = binom.cdf(x\_values, k, p1)  
binom\_cdf\_p2 = binom.cdf(x\_values, k, p2)  
  
plt.plot(x\_values, binom\_cdf\_p, label=**r'$B(k=10, p=0.4)$'**)  
plt.plot(x\_values, binom\_cdf\_p1, label=**r'$B(k=10, p=0.365)$'**)  
plt.plot(x\_values, binom\_cdf\_p2, label=**r'$B(k=10, p=0.412)$'**)  
plt.title(**r'Функции распределения $B(k, p)$ для $p=0.4$, $p\_1=0.365$, и $p\_2=0.412$'**, fontsize=14)  
plt.legend()  
plt.grid(**True**)  
plt.show()