*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 № \_3\_**

**Название лабораторной работы: Моделирование выборки из абсолютно непрерывного закона распределения методом обратных функций.**

**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**

1. Для данного методом обратных функций смоделируйте выборку из закона распределения с заданной плотностью .

2. Для полученной выборки найдите гистограмму относительных частот. Постройте на одном рисунке графики теоретической плотности и гистограмму относительных частот.

3. Вычислите выборочное среднее и выборочную дисперсию и сравните с истинными значениями этих характеристик.

4. Используя неравенство Dvoretzky-Kiefer-Wolfowitz, постройте 90% доверительный интервал для функции распределения .

**Исходные данные:**

Объем выбокри: n=120

Плотность вероятности: – распределение Парето

**Моделирование выборки**

Дана плотность вероятности распределения закона Парето , находим функцию распределения для данного закона Парето:

Вычисляем обратную к F(x) функцию:

Сгенерируем n случайных чисел из интервала (0,1):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 0.488555 | 0.239591 | 0.340821 | 0.336837 | 0.405022 | 0.018034 |
| 2 | 0.359453 | 0.072098 | 0.757447 | 0.992371 | 0.556311 | 0.095661 |
| 3 | 0.459099 | 0.720099 | 0.045583 | 0.53232 | 0.852092 | 0.614168 |
| 4 | 0.081241 | 0.334455 | 0.510657 | 0.32521 | 0.966653 | 0.630458 |
| 5 | 0.49207 | 0.501789 | 0.640284 | 0.913976 | 0.655989 | … |

Пересчитываем значения с помощью обратной функции и получаем выборку X:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 51.704704 | 50.689457 | 51.052832 | 51.037453 | 51.315074 | 50.045517 |
| 2 | 51.126075 | 50.187423 | 53.669762 | 63.803812 | 52.073416 | 50.25201 |
| 3 | 51.560143 | 53.29473 | 50.116773 | 51.936486 | 55.013645 | 52.43848 |
| 4 | 50.212278 | 51.028304 | 51.819037 | 50.993119 | 59.267576 | 52.551705 |
| 5 | 51.722536 | 51.772524 | 52.622565 | 56.524798 | 52.740153 | … |

**Первоначальная обработка полученных статистических данных**

Число интервалов для построения гистограммы относительных частот вычислим по формуле Стёрджиса:

Подсчет частот и относительных частот попадания элементов выборки в каждый интервал:

|  |  |  |
| --- | --- | --- |
| Интервал | Частота | Относительная частота |
| [50; 52.23) | 75 | 0.625 |
| [52.23; 54.46) | 23 | 0.192 |
| [54.46; 56.69) | 12 | 0.1 |
| [56.69; 58.92) | 4 | 0.033 |
| [58.92; 61.15) | 3 | 0.025 |
| [61.15; 63.38) | 1 | 0.008 |
| [63.38; 65.62] | 2 | 0.017 |

Гистограмма относительных частот и график теоретической плотности распределения закона Парето:

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

По графику делаем вывод, что частоты полученного распределения соответствуют теоретическому распределению.

**Эмпирические и теоретические характеристики**

Математическое ожидание:

Выборочное среднее:

Разность теоретического и эмпирического среднего:

Дисперсия:

Выборочная дисперсия:

Отношение выборочной и теоретической дисперсии:

Из полученных данных делаем вывод, что расхождение выборочного среднего и математического ожидания мало, а отношение дисперсия близко к единице, следовательно, смоделированную выборку можно считать удовлетворительной.

**Построение доверительного интервала для теоретической функции распределения**

Построим доверительный интервал (L(x); R(x)) при помощи неравенства Дворецкого-Кифера-Волфовица. Этот интервал указывает границы, внутри которых заключена функция распределения F(x) с вероятностью 0,9.

Таким образом, если , , , то с вероятностью 1 – α:

,

где

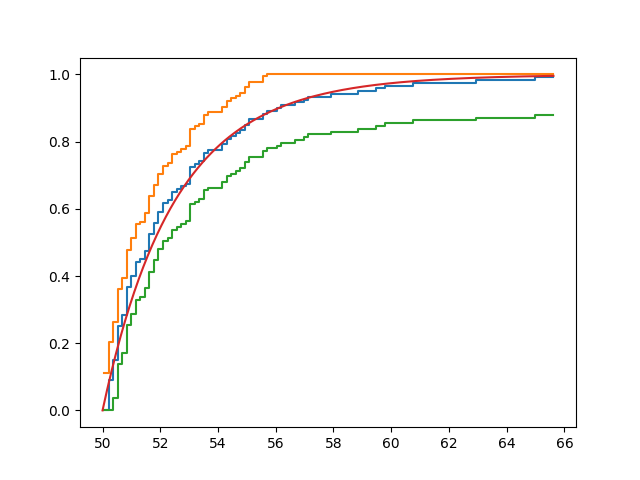
,

Имеем:

Эмпирическая функция:

Границы интервала:

Графическая иллюстрация построенного доверительного интервала:



**Вывод:** в ходе проделанной лабораторной работы было проделано моделирование выборки из абсолютно непрерывного закона распределения методом обратных функций. Полученных эмпирические харакетристики смоделированной выборки близки к теоретическим, что свидетельствует об удовлетворительном качестве моделирования.

**Приложение**

Программный код на языке Python.

**import** random  
**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**import** math  
  
  
**def** reverse\_pareto(y: float) -> float:  
 **return** 50 / (1 - y) \*\* (1 / 20)  
  
  
**def** ind(z: float) -> bool:  
 **if** z > 0:  
 **return** 1  
 **else**:  
 **return** 0  
  
  
**def** f\_emp(z: float, n: int, X: list[float]) -> float:  
 res = 0  
 **for** i **in** range(n):  
 res += ind(z - X[i])  
  
 **return** res  
  
  
*# Константы варианта, дано распределение Парето*n = 120 *# объем выборки  
# Параметры распределения*k = 20  
xm = 50  
  
*# Моделируем числа*X = []  
**for** i **in** range(n):  
 X.append(round(random.random(), 6))  
  
print(X)  
  
*# Пересчитываем с помощью обратной функции*Y = []  
**for** i **in** range(len(X)):  
 Y.append(round(reverse\_pareto(X[i]), 6))  
  
print(Y)  
  
*# Обработка данных*Y\_exp = k \* xm / (k - 1)  
D = ((xm / (k - 1)) \*\* 2) \* k / (k - 2)  
print(**f'Математическое ожидание: {**round(Y\_exp, 6)**}'**)  
print(**f'Дисперсия: {**round(D, 6)**}'**)  
  
Y\_avg = sum(Y) / n  
S2 = sum([(item - Y\_avg) \*\* 2 **for** item **in** Y]) / (n - 1)  
print(**f'Выборочное среднее: {**Y\_avg**}'**)  
print(**f'Выборочная дисперсия: {**S2**}'**)  
print(**f'Сравнение: {**round(abs(Y\_avg - Y\_exp), 6)**}'**)  
print(**f'Сравнение: {**round((D / S2) \*\* (1 / 2), 6)**}'**)  
  
delta = (max(Y) - 50) / 7  
rel\_freq = [0, 0, 0, 0, 0, 0, 0]  
**for** item **in** Y:  
 i = 0  
 marg = delta  
 **while** marg < max(Y):  
 **if** item <= 50 + marg:  
 rel\_freq[i] += 1  
 **break  
 else**:  
 marg += delta  
 i += 1  
  
print(max(Y))  
print(rel\_freq)  
  
freq = [rel\_freq[i] / 120 **for** i **in** range(len(rel\_freq))]  
print(freq)  
  
intervals = np.linspace(50, max(Y), 8)  
  
*#plt.bar(intervals[:-1], np.array(rel\_freq) / n, width=intervals[1] - intervals[0], align='edge', edgecolor='black')*plt.hist(Y, bins=7, density=**True**, edgecolor=**'black'**)  
*# Генерация данных*x = np.linspace(xm, max(Y), 1000) *# Диапазон значений x*pdf = (k \* (xm \*\* k)) / (x \*\* (k + 1)) *# Плотность распределения  
  
# Построение графика*plt.plot(x, pdf, color=**'red'**)  
plt.xlabel(**'Значения'**)  
plt.ylabel(**'Относительная частота'**)  
plt.title(**'Гистограмма относительных частот и теоретическая плотность распределения'**)  
plt.legend()  
plt.show()  
  
*# Построение эмпирической ф-ии распределения и нер-во Дворецкий-Кифер-Волфовиц*gamma = 0.1  
epsilon = math.sqrt(-1 / (2 \* n) \* math.log((gamma / 2), math.e))  
print(**f'Эпсилон: {**epsilon**}'**)  
  
x = np.linspace(50, max(Y), 300)  
f = 1 - (xm / x) \*\* k  
  
z\_values = np.linspace(min(Y), max(Y), 100)  
  
*# Вычисление значений эмпирической функции распределения*f\_emp\_values = [f\_emp(z, n, Y) / n **for** z **in** z\_values]  
f\_emp\_values\_pe = [f\_emp(z, n, Y) / n + epsilon **for** z **in** z\_values]  
f\_emp\_values\_me = [f\_emp(z, n, Y) / n - epsilon **for** z **in** z\_values]  
  
f\_emp\_values = np.clip(f\_emp\_values, 0, 1)  
f\_emp\_values\_pe = np.clip(f\_emp\_values\_pe, 0, 1)  
f\_emp\_values\_me = np.clip(f\_emp\_values\_me, 0, 1)  
  
*# Построение ступенчатого графика*plt.step(z\_values, f\_emp\_values, where=**'post'**)  
plt.step(z\_values, f\_emp\_values\_pe, where=**'post'**)  
plt.step(z\_values, f\_emp\_values\_me, where=**'post'**)  
plt.plot(x, f)  
plt.legend()  
plt.show()