ГУАП

КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ассистент |  |  |  | М.Н. Шелест |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №2 |
| ДАТЧИКИ СЛУЧАЙНЫХ ЧИСЕЛ. ПОСТРОЕНИЕ ДИАГРАММ |
| по курсу: ПОСТРОЕНИЕ И АНАЛИЗ ГРАФОВЫХ МОДЕЛЕЙ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4917 |  |  |  | В.С. Тихонов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2022

Цель работы: изучение алгоритмов получения на ЭВМ чисел с заданным законом распределения и построения гистограмм.

Вариант 20

|  |  |
| --- | --- |
| Закон распределения | Параметры закона |
| Эрланговский |  |

Ход работы:

Выведем соотношение, позволяющее из чисел, сформированных базовым датчиком, получить числа с заданным законом распределения:

Напишем программу, реализующую датчик случайных чисел с заданным законом распределения и программу построения гистограммы выборки, сформированной созданным датчиком с учетом параметров, заданных в таблице (рис. 1).

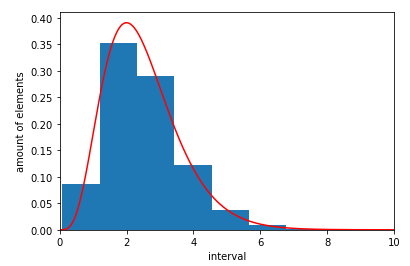


Рисунок 1 – Гистограмма выборки

При помощи программы построения гистограмм заполним таблицу 2. Зафиксируем xmin и xmax:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер интервала | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Число элементов в данным интервале | 13970 | 40276 | 42457 | 28735 | 14735 | 6587 | 2480 | 937 | 291 | 100 |

Xmin = 0.277

Xmax = 5.717

На основании таблицы 2 построить гистограмму распределения сформированной выборки (Рисунок 2):

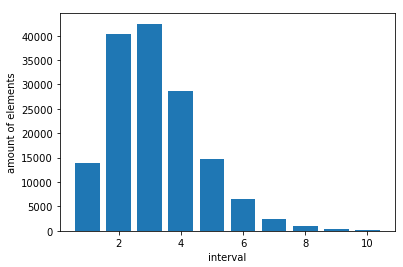


Рисунок 2 – Гистограмма распределения сформированной выборки

Построение графиков зависимости оценок математического ожидания и дисперсии от объема выборки (рис. 3, 4).

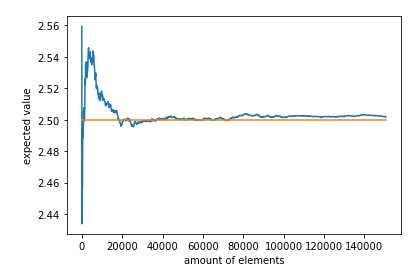


Рисунок 3 – График оценки математического ожидания от объема выборки

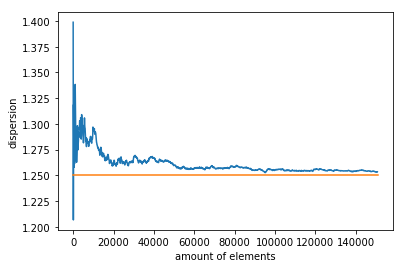


Рисунок 4 – График зависимости оценки дисперсии от объема выборки

Вывод: в ходе выполнения лабораторной работы были изучены алгоритмы получения на ЭВМ чисел с заданным законом распределения и построения гистограмм. Было выведено соотношение, на основе которого был разработан датчик случайных чисел с Эрланговским законом распределения и программа построения гистограммы выборки, сформированной созданным датчиком. Полученные результаты были проверены аналитически. Так же была исследована зависимость оценок математического ожидания и дисперсии от объема выборки.

Листинг 1 – Код программы

|  |
| --- |
| from matplotlib import pyplot as plt  import numpy as np  import scipy.stats as st  from scipy import integrate  import math  def rand\_gener(k, lmb, n):  res = []  for j in range(n):  a = 0  for i in range(k):  a = a + (-(1 / lmb) \* np.log(np.random.random()))  res.append(a)  return res  def gamma\_hist(k, lmb, n):  A, steps, mu, d = [], [], [], []  res = [0, 1000]  x\_min = x\_max = delta = 0  while min(res) < n: # N = N+1  A.append(rand\_gener(k, lmb, 1)[0]) # формирование случайного числа и добавление его к выборке А  if len(A) % 100 == 0:  mu.append(sum(A) / len(A)) # мат ожидание  d.append(np.var(A)) # дисперсия  if len(A) < 1000:  continue  elif x\_min <= A[-1] <= x\_max:  #  bar = int((A[-1] - x\_min) / delta)  res[((bar + 1) \* (bar <= len(res) - 1)) - 1] += 1  if 1 - sum(res) / len(A) < 0.01: # вероятность попадания случайной величины за границы данного интервала < 0.01  continue  x\_min = min(A) # левая граница минимальный элемент выборки  x\_max = max(A) # правая граница максимальный элемент выборки  delta = (x\_max - x\_min) / 10 # делта  res = np.zeros(10) # Количество попавших в res[i] интервал элементов выборки  # Запронение интервалов гисторгаммы ()  for i in A:  bar = int((i - x\_min) / delta)  res[((bar + 1) \* (bar <= len(res) - 1)) - 1] += 1  print(x\_min)  print(x\_max)  return A, mu, d, res  def main():  k, lmb = 5, 2  A, mu, d, res = gamma\_hist(k, lmb, 100)  print(res)  plt.xlim(0, 10)  x = np.linspace(0, 10, 10000)  plt.hist(A, density=True, bins=10, label="Values")  plt.plot(x, st.gamma.pdf(x, k, scale=1 / lmb), color="red")  plt.ylabel('amount of elements')  plt.xlabel('interval')  plt.show()  plt.plot([100 \* i for i in range(len(mu))], mu)  plt.plot([100, len(mu) \* 100], [k / lmb] \* 2)  plt.xlabel('amount of elements')  plt.ylabel('expected value')  plt.show()  plt.plot([100 \* i for i in range(len(d))], d)  plt.plot([100, len(d) \* 100], [k / (lmb \*\* 2)] \* 2)  plt.xlabel('amount of elements')  plt.ylabel('dispersion')  plt.show()    plt.bar([1, 2, 3, 4, 5, 6, 7, 8, 9, 10], res)  plt.ylabel('amount of elements')  plt.xlabel('interval')  plt.show()  if \_\_name\_\_ == "\_\_main\_\_":  main() |