Университет ИТМО

Практическая работа №5 по предмету

теория вероятностей

Выполнил: Патутин В.М

Преподаватель: Селина Е. Г.

Санкт-Петербург

2021 г

Листинг программы:

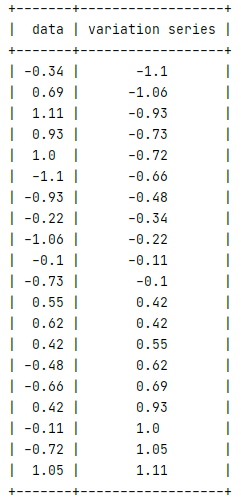
import math as m  
import sys  
  
import matplotlib.pyplot as plt  
from prettytable import PrettyTable  
  
  
def matematical\_expenctation(series, rate\_met):  
 vrb = 0  
 sum = 0  
 for i in range(len(series)):  
 vrb += series[i] \* rate\_met[i]  
 sum += rate\_met[i]  
 return vrb / sum  
  
  
def deviation(series, rate\_met):  
 vrb = 0  
 for i in range(len(series)):  
 vrb += series[i] \* rate\_met[i]  
 ret = 0  
 for i in range(len(series)):  
 ret += rate\_met[i] \* (series[i] - vrb) \*\* 2  
 ret = ret \*\* (1 / 2)  
 return ret  
  
  
*# input*file = open(**'input.txt'**, **'r'**)  
mass = file.readline().strip().replace(**","**, **"."**).split(**" "**)  
data = list()  
for i in range(0, 20):  
 try:  
 number = float(mass[i])  
 except ValueError as e:  
 print(**"Exception :"**, e)  
 sys.exit()  
 data.append(number)  
file.close()  
  
*# firste table*t = PrettyTable()  
  
t.add\_column(**"data"**, data)  
data.sort()  
t.add\_column(**"variation series"**, data)  
  
print(t)  
  
*# second table*t = PrettyTable()  
  
statistical\_series = list()  
for el in data:  
 if statistical\_series.count(el) == 0:  
 statistical\_series.append(el)  
statistical\_series.sort()  
t.add\_column(**"statistical series"**, statistical\_series)  
  
statistical\_series\_count = list()  
for el in statistical\_series:  
 statistical\_series\_count.append(data.count(el))  
t.add\_column(**"n"**, statistical\_series\_count)  
  
statistical\_series\_rate = list()  
for el in statistical\_series\_count:  
 statistical\_series\_rate.append(el / len(data))  
t.add\_column(**"rate"**, statistical\_series\_rate)  
  
print(t)  
print(**"Экстремальные значения:"**, statistical\_series[0], **";"**, statistical\_series[len(statistical\_series) - 1])  
print(**"Размах:"**, **'%-13.2f'** % (statistical\_series[len(statistical\_series) - 1] - statistical\_series[0]))  
print(**"Математическое ожидание:"**, **'%-13.2f'** % (matematical\_expenctation(statistical\_series, statistical\_series\_count)))  
print(**"Средне квадратичное отклонение:"**, **'%-13.2f'** % (deviation(statistical\_series, statistical\_series\_rate)))  
  
*# third table and graph №1*plt.subplot(5, 1, 1)  
plt.title(**"Эмпирическая функция распределения"**)  
print(**"Эмпирическая функция распределения:"**)  
print(**"**\t\t**0, X <="**, statistical\_series[0])  
sum\_f\_x = statistical\_series\_count[0]  
for i in range(len(statistical\_series\_count) - 1):  
 print(**'**\t\t**%-2.2f'** % (sum\_f\_x / 20), **";"**, statistical\_series[i], **"< X <="**, statistical\_series[i + 1])  
 x\_graph = list()  
 y\_graph = list()  
 x\_graph.append(statistical\_series[i])  
 x\_graph.append(statistical\_series[i + 1])  
 y\_graph.append(sum\_f\_x / 20)  
 y\_graph.append(sum\_f\_x / 20)  
 plt.plot(x\_graph, y\_graph)  
 sum\_f\_x += statistical\_series\_count[i + 1]  
print(**"**\t\t**1, X>"**, statistical\_series[-1])  
  
*# fourth table and graph №3*plt.subplot(5, 1, 5)  
plt.title(**"Гистограмма частот"**)  
  
t = PrettyTable()  
  
array = list()  
h = (data[-1] - data[0]) / (1 + m.log(len(data), 2))  
step = data[0] - h/2  
  
while step < data[-1]:  
 array.append(step)  
 step += h  
 array.append(step)  
  
integral\_series = list()  
for i in range(0, len(array), 2):  
 pair = list()  
 pair.append(array[i])  
 pair.append(array[i + 1])  
 integral\_series.append(pair)  
  
integral\_series\_count = list()  
integral\_series\_rate = list()  
array\_y = list()  
for i in range(0, len(array), 2):  
 kol = 0  
 for el in data:  
 if array[i] <= el <= array[i + 1]:  
 kol += 1  
 integral\_series\_count.append(kol)  
 integral\_series\_rate.append(kol / 20)  
 array\_y.append(kol / 20)  
 array\_y.append(kol / 20)  
  
x\_graph = list()  
y\_graph = list()  
for i in range(0, len(array), 2):  
 st = array[i]  
 end = array[i + 1]  
 while (st <= end):  
 x\_graph.append(st)  
 st += h  
  
plt.hist(data, x\_graph)  
t.add\_column(**"integral series"**, integral\_series)  
t.add\_column(**"n"**, integral\_series\_count)  
t.add\_column(**"rate"**, integral\_series\_rate)  
  
*# GRAPH 2*plt.subplot(5, 1, 3)  
plt.title(**"Полигон частот"**)  
x = list()  
y = list()  
for i in range(len(integral\_series\_count)):  
 x.append((float(integral\_series[i][0]) + float(integral\_series[i][0])) / 2)  
 y.append(integral\_series\_rate[i])  
  
plt.plot(x, y)  
  
print(t)  
plt.show()

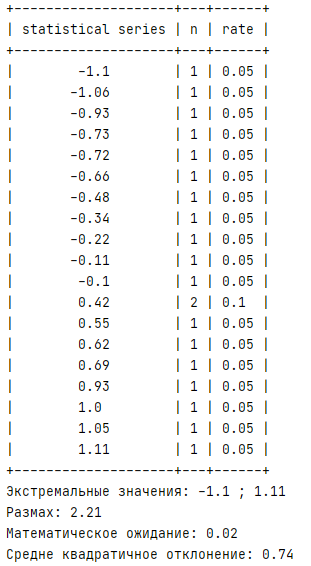
Пример работы программы:

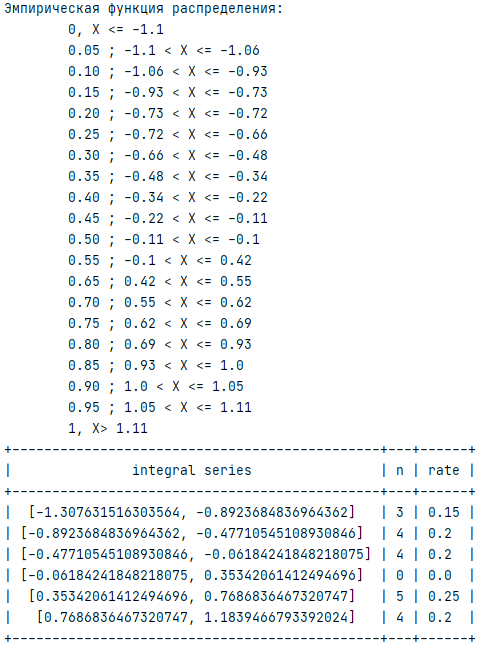
input:

-0.34 0.69 1.11 0.93 1.00 -1.1 -0.93 -0.22 -1.06 -0.1 -0.73 0.55 0.62 0.42 -0.48 -0.66 0.42 -0.11 -0.72 1.05

output:







Графики:

