**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра ИС**

**ОТЧЁТ**

**по практической работе №1**

**по дисциплине «Статистический анализ»**

**Тема: Формирование и первичная обработка выборки.**

**Ранжированные и интервальные ряды.**

**Вариант № 8**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент гр. 9372 | |  | Иванов Р.С. |
| Преподаватель |  |  | Сучков А.И. |

Санкт-Петербург

2021

**Цель работы.**

Ознакомление с основными правилами формирования выборки и подготовки выборочных данных к статистическому анализу.

**Основные теоретические положения.**

Для изучения формы эмпирического распределения проводят группировку данных. Результаты группировки представляют в виде таблиц и графиков.

Аналитическая группировка данных предназначена для анализа корреляционной взаимосвязи. Такая группировка заключается в разбиении диапазона возможных значений на интервалы и подсчете итогов по каждой группе. Для каждого из интервалов необходимо вычислить следующие показатели:

– ***ni*** – частота (количество элементов выборки, попадающих в данный интервал);

– ***pi*** – относительная частота, частость (доля числа элементов в данном интервале от объема выборки);

**Постановка задачи.**

Осуществить формирование репрезентативной выборки заданного объема из имеющейся генеральной совокупности экспериментальных данных.

Осуществить последовательное преобразование полученной выборки в ранжированный, вариационный и интервальный ряды. Применительно к интервальному ряду построить и отобразить графически полигон, гистограмму эмпирическую функцию распределения для абсолютных и относительных частот, а также кумуляту. Полученные результаты содержательно проинтерпретировать.

**Выполнение работы.**

Во время выполнения работы был написан код на языке Python, выполняющий поставленную задачу. Реализовано чтение и дальнейшая обработка данных из Price\_Mileage.csv, одобренного преподавателем. Для этого была использована библиотека csv.

Выборка была сформирована во время выполнения практической работы 0 и использована в ходе выполнения этой практической работы. Объём выборки составлял 115 позиций.

Ранжированный ряд был получен с помощью встроенной функции sort. Часть результата представлена в табл.1.

Таблица 1 – Начало ранжированного ряда

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3641 | 5359 | 5775 | 6530 | 6743 | 6748 | 7250 | 7415 | 9445 | 10145 |
| 10447 | 10944 | 11346 | 12265 | 13162 | 13239 | 14870 | 15340 | 17355 | 17432 |
| 17946 | 19164 | 19235 | 19420 | 19636 | 20575 | 21304 | 23095 | 23783 | 23987 |
| 24722 | 26026 | 26353 | 26549 | 26996 | 27327 | 28280 | 28307 | 30229 | … |

Вариационный ряд составлен с помощью библиотеки nltk. Пример указан в табл. 2. Так как в качестве выборки был взят пробег машин, вариационный ряд не позволяет сделать никаких содержательных выводов.

Таблица 2 – Начало вариационного ряда

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Xi | 3641 | 5359 | 5775 | 6530 | 6743 | 6748 | 7250 | 7415 | 9445 | … |
| ni | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | … |
| pi | 0.009 | 0.009 | 0.009 | 0.009 | 0.009 | 0.009 | 0.009 | 0.009 | 0.009 | … |

Для интервального ряда вычислен размах вариации в соответствие с формулой (1).

(1)

Вычислено рекомендуемое количество интервалов в соответствие с формулой (2).

(2)

Берется интервалов на 1 больше, так как за берется не , а рассчитывается по рекомендованной преподавателем формуле (3).

(3)

где h – длина каждого интервала нашего ряда, высчитанная по формуле (4).

(4)

Пример построения интервального ряда в табл. 3.

Таблица 3 – Интервальный ряд

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Xi | (0; 27023] | (27023; 54046] | (54046; 81069] | (81069; 108092] | (108092; 135115] | (135115; 162138] | (162138; 189161] | (189161; 216184] | (216184; 243207] |
| ni | 35 | 44 | 14 | 13 | 4 | 1 | 3 | 0 | 1 |
| pi | 0.304 | 0.383 | 0.122 | 0.113 | 0.035 | 0.009 | 0.026 | - | 0.009 |

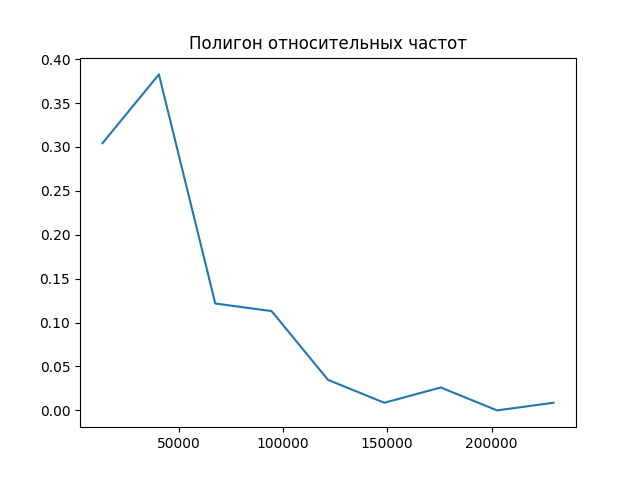
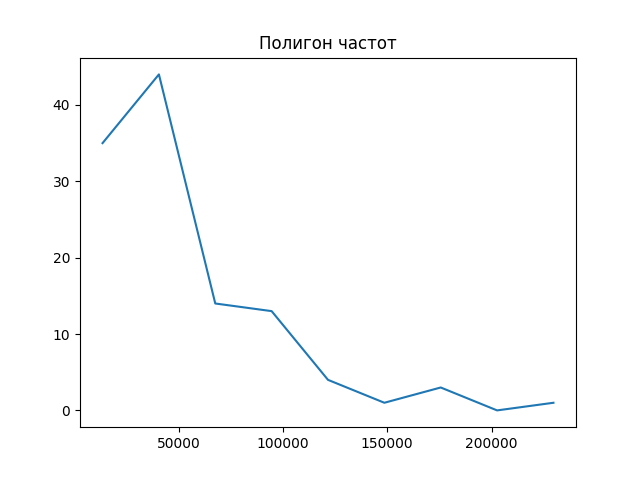
Также были высчитаны центры интервалов, для удобного отображения на графиках. Полигоны частот абсолютных и относительных рассчитаны с помощью библиотеки *matplotlib* (как и все последующие графики), функцией *plot*. Записаны в файлы ***poly.png*** и ***poly\_otn.png*** соответственно. Примеры полигонов см. на рис. 1 и рис. 2.

Рисунок 1 - poly\_otn.png

Рисунок 2 – poly.png

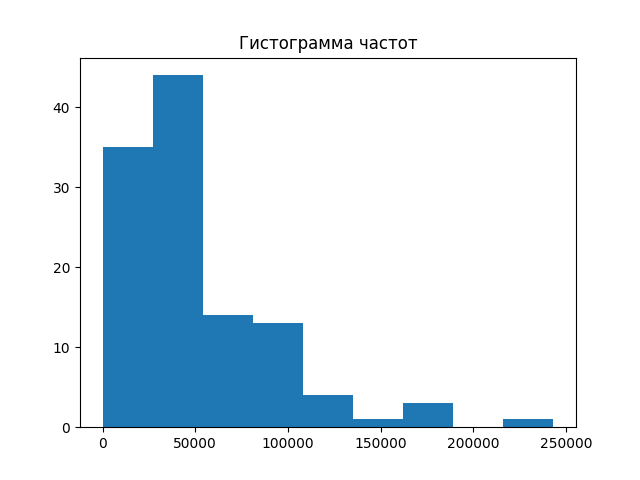
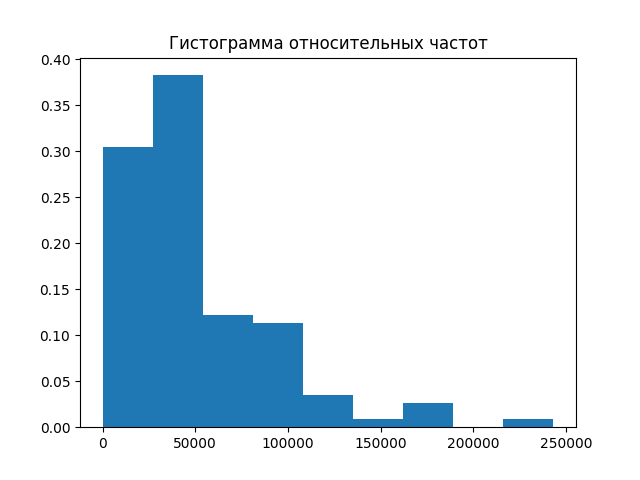
 Гистограммы частот абсолютных и относительных рассчитаны с помощью функции *bar*. Файлы записаны как ***gist.png***, ***gist\_otn.png*** соответственно. Примеры гистограммы см. на рис. 3 и рис. 4.

Рисунок 3 - gist.png

Рисунок 4 - gist\_otn.png

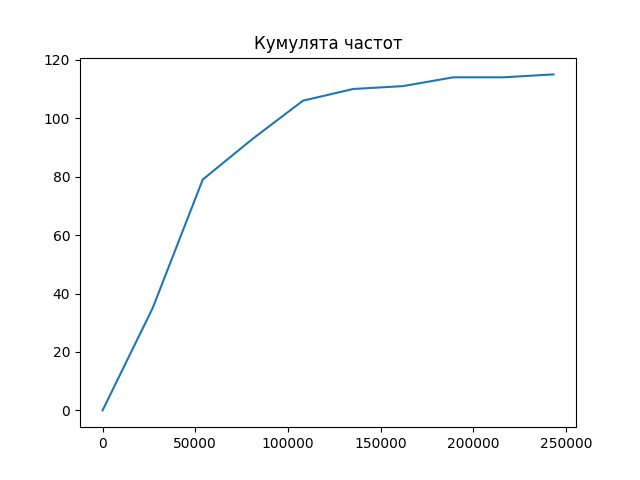
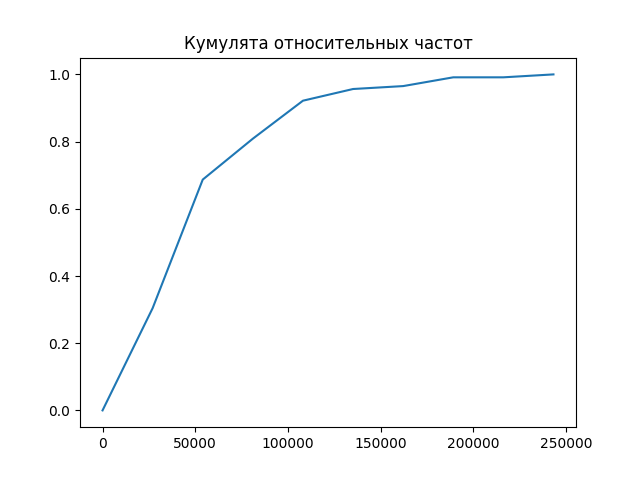
 Для отображения кумуляты и эмпирической функции были высчитаны накопленные относительные и абсолютные частоты. Файлы кумуляты записаны как ***cum.png*** и ***cum\_otn.png*** для абсолютных и относительных частот соответственно. Примеры кумулят отображены на рис. 5 и рис. 6.

Рисунок 6 - cum\_otn.png

Файлы эмпирической функции распределения записаны ***emp\_func\_otn.png*** и ***emp\_func.png*** для относительных и абсолютных частот соответственно. Примеры см. на рис. 7 и рис. 8.

Рисунок 5 - cum.png

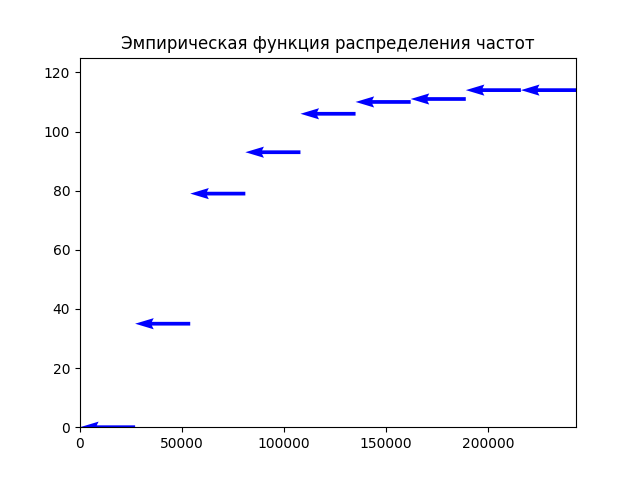
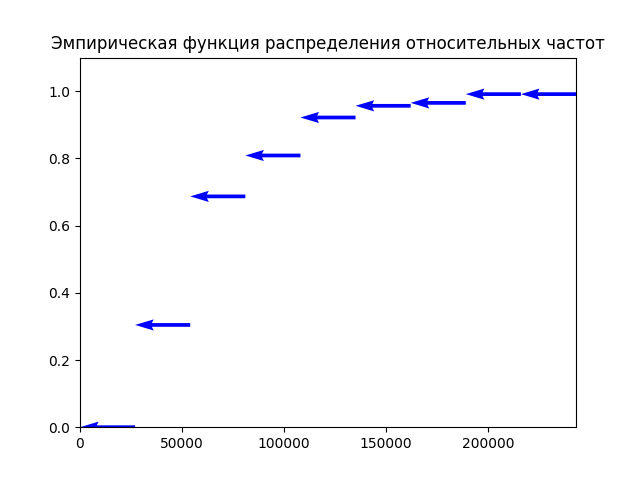
 Для удобной интерпретации информации все файлы, a также значения ранжированного, вариационного и интервального рядов сгруппированы в ***xlsx*** файле с помощью библиотеки *openpyxl.* Исходный код программы см. в приложении А.

Рисунок 6 - emp\_func.png

Рисунок 8 - emp\_func\_otn.png

**Вывод.**

Изучены основные правила формирования выборки и подготовки выборочных данных к статистическому анализу. Обнаружено, что графики относительных и абсолютных частот не отличаются своим поведением. Изучили способы построения кумуляты, полигонов, эмпирических функций и гистограмм.

**ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

import matplotlib.pyplot as plt  
import math  
import csv  
from openpyxl import Workbook  
from openpyxl.drawing.image import Image  
  
sample = []

with open('Price-Mileage.csv') as csv\_file: # Читаем выборку из файла 0 работы  
 spam\_reader = csv.reader(csv\_file, quotechar='|')  
 for row in spam\_reader:  
 x, y = row[0].split(';')  
 if y.isdigit(): sample.append(int(y))  
  
sample.sort() # Используя встроенную функцию сортировки получаем ранжированный ряд  
  
R = sample[len(sample) - 1] - sample[0] # Размах  
print("R =", R)  
  
k = round(1 + math.log2(len(sample))) # Число интервалов(Формула Стёрджеса)  
print("k = ", k)  
  
h = round(R / k) # Длина интервала  
print("h =", h)  
  
k += 1 # Иначе интервалы не покроют выборку  
  
x0 = sample[0] - h / 2 # Начало первого частичного интервала  
if x0 < 0:  
 x0 = 0  
  
print("x0 =", x0)  
  
interval = []  
variation = []  
x = x0  
  
# Получаем вариационный ряд  
for i in range(len(sample)):  
 a = 0  
 a += 1  
 if i == len(sample) - 1:  
 variation.append([sample[i], a, a/len(sample)])  
 break  
 if sample[i] != sample[i+1]:  
 variation.append([sample[i], a, a/len(sample)])  
  
a = 0  
  
for i in range(k):  
 interval.append([(x, x + h), 0, 0])  
 x += h  
  
# Получаем интервальный ряд  
for i in sample:  
 for j in range(k):  
 if interval[j][0][0] < i <= interval[j][0][1]:  
 interval[j][1] += 1  
 break  
  
for i in interval:  
 i[2] = i[1] / len(sample)  
  
print(interval)  
  
middle\_int = []  
accum\_freq = []  
accum\_afreq = []  
a = 0  
b = 0  
  
# Вычисляем серидины интервалов и их накопленные частоты  
for i in range(len(interval)):  
 a = a + interval[i][2]  
 b = b + interval[i][1]  
 middle\_int.append(interval[i][0][0] + h / 2)  
 accum\_afreq.append(a)  
 accum\_freq.append(b)  
  
x = middle\_int  
y = [interval[i][1] for i in range(len(interval))]  
plt.plot(x, y)  
plt.title("Полигон частот")  
plt.savefig('poly.png')  
plt.show()  
plt.clf()  
  
y = [interval[i][2] for i in range(len(interval))]  
plt.plot(x, y)  
plt.title("Полигон относительных частот")  
plt.savefig("poly\_otn.png")  
plt.show()  
plt.clf()  
  
x = [x0 + i \* h + h / 2 for i in range(k)]  
y = [interval[i][1] for i in range(len(interval))]  
plt.bar(x, y, width=h)  
plt.title("Гистограмма частот")  
plt.savefig("gist.png")  
plt.show()  
plt.clf()  
  
y = [interval[i][2] for i in range(len(interval))]  
plt.bar(x, y, width=h)  
plt.title("Гистограмма относительных частот")  
plt.savefig("gist\_otn.png")  
plt.show()  
plt.clf()  
  
X = []  
Y = accum\_afreq  
U = []  
V = accum\_afreq  
for i in range(k):  
 U.append(x0 + h \* i)  
 X.append(x0 + h \* i + h)  
plt.quiver(X, Y, -h, 0, angles='xy', scale\_units='xy', scale=1, color='b')  
plt.title("Эмпирическая функция распределения относительных частот")  
plt.xlim(0, 250000)  
plt.ylim(0, 1.1)  
plt.savefig("emp\_func\_otn.png")  
plt.show()  
plt.clf()  
  
Y = accum\_freq  
V = accum\_freq  
plt.quiver(X, Y, -h, 0, angles='xy', scale\_units='xy', scale=1, color='b')  
plt.title("Эмпирическая функция распределения частот")  
plt.xlim(0, 250000)  
plt.ylim(0, len(sample) + 10)  
plt.savefig("emp\_func.png")  
plt.show()  
plt.clf()  
plt.plot(middle\_int, accum\_freq)  
plt.title("Кумулята частот")  
plt.savefig("cum.png")  
plt.clf()  
plt.plot(middle\_int, accum\_afreq)  
plt.title("Кумулята относительных частот")  
plt.savefig("cum\_otn.png")  
plt.show()  
plt.clf()  
  
wb = Workbook()  
filename = "output.xlsx"  
ws = wb.active  
ws.title = "Result1"  
ws["A1"] = "Ранжированный"  
ws.append(sample) # Делаю так исключительно ради удобства, можно запариться еще больше  
ws["A4"] = "Вариационный ряд"  
ws.append(variation[i][0] for i in range(len(variation)))  
ws["A6"] = "Частоты"  
ws.append([variation[i][1] for i in range(len(variation))])  
ws["A8"] = "Относительные частоты"  
ws.append([variation[i][2] for i in range(len(variation))])  
ws["A12"] = "Интервальный ряд"  
ws.append([str(interval[i][0]) for i in range(len(interval))])  
ws["A14"] = "Частоты"  
ws.append([interval[i][1] for i in range(len(interval))])  
ws["A16"] = "Относительные частоты"  
ws.append([interval[i][2] for i in range(len(interval))])  
ws.add\_image(Image("poly.png"), "A18")  
ws.add\_image(Image("poly\_otn.png"), "K18")  
ws.add\_image(Image("gist.png"), "A43")  
ws.add\_image(Image("gist\_otn.png"), "K43")  
ws.add\_image(Image("emp\_func.png"), "A69")  
ws.add\_image(Image("emp\_func\_otn.png"), "K69")  
ws.add\_image(Image("cum.png"), "A95")  
ws.add\_image(Image("cum\_otn.png"), "K95")  
wb.save(filename=filename)