

**МИНОБРНАУКИ РОССИИ  
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра математического обеспечения и применения ЭВМ**

**КУРСОВАЯ РАБОТА  
по дисциплине «Статистические методы обработки экспериментальных  
данных»**

**ТЕМА: ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И КОМПЬЮТЕРНОЕ ИССЛЕДОВАНИЕ  
АЛГОРИТМОВ ОБРАБОТКИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ**

Студент гр. 7381

---

Тарасенко Е.А.

Преподаватель

---

Середа В.И.

Санкт-Петербург

2021

**ЗАДАНИЕ**  
**НА КУРСОВУЮ РАБОТУ**

Студент Тарасенко Е.А.

Группа 7381

Тема работы: Программная реализация и компьютерное исследование алгоритмов обработки экспериментальных данных

Исходные данные:

Из представленной генеральной совокупности формируется выборка заданного объема по одному из представленных в таблице признаков. Необходимо провести выравнивание статистических рядов, выполнить корреляционный, регрессионный и кластерный анализы.

Содержание пояснительной записи:

Содержание, Введение, Выравнивание статистических рядов, Корреляционный и регрессионный анализ, Кластерный анализ, Заключение, Список используемых источников.

Предполагаемый объем пояснительной записи:

Не менее 20 страниц.

Дата выдачи задания: 06.04.2021

Дата сдачи реферата: 13.04.2021

Дата защиты реферата: 20.04.2021

Студент

---

Тарасенко Е.А.

Преподаватель

---

Середа В.И.

## **АННОТАЦИЯ**

В рамках данной курсовой работы с помощью разработанных программных средств была сформирована из заданной генеральной совокупности двумерная выборка, для которой впоследствии по каждому из двух параметров были получены статистические, ранжированные, вариационные и интервальные ряды.

С учетом выравнивания полученных рядов над выборкой был произведен корреляционный и регрессионный, а также кластерный анализ. По результатам каждого из анализов сделаны выводы.

## **SUMMARY**

As part of this course work, using the developed software, a two-dimensional sample was formed from a given general population, for which statistical, ranked, variational and interval series were subsequently obtained for each of the two parameters.

Taking into account the alignment of the obtained series over the sample, correlation and regression, and cluster analysis was performed. Conclusions are drawn from the results of each of the analyzes.

# СОДЕРЖАНИЕ

Введение	6
1. Выравнивание статистических рядов	7
1.1. Основные теоретические положения	7
1.2. Формирование и первичная обработка выборки. Ранжированный и интервальный ряды	10
1.3. Нахождение точечных оценок параметров распределения	16
1.4. Нахождение интервальных оценок параметров распределения. Проверка статистической гипотезы о нормальном распределении	17
1.5. Выводы	20
2. Корреляционный и регрессионный анализ	21
2.1. Основные теоретические положения	21
2.2. Проверка статистической гипотезы о равенстве коэффициента корреляции нулю	25
2.3. Выборочные прямые среднеквадратической регрессии. Корреляционные отношения	33
2.4. Выводы	37
3. Кластерный анализ	39
3.1. Основные теоретические положения	39
3.2. Метод $k$ -средних	42
3.3. Метод поиска сгущений	52
3.4. Выводы	59
Заключение	60
Список использованных источников	61
Приложение А. Исходный код программы для выравнивания статистических рядов	62

Приложение Б. Исходный код программы для корреляционного и регрессионного анализа	68
Приложение В. Исходный код программы для кластерного анализа	77

## **ВВЕДЕНИЕ**

Целью данной работы является разработка программных средств и исследование с их помощью, полученной из заданной генеральной совокупности, двумерной выборки.

Над полученной выборкой необходимо было произвести корреляционный, регрессионный и кластерный анализ. Для выполнения поставленной задачи по ней были сформированы (для каждого из двух рассматриваемых параметров) статистические, ранжированные, вариационные и интервальные ряды.

Формирование рядов происходило для обоих из рассматриваемых параметров выборки. Для интервальных рядов были построены полигоны и гистограммы частот, а также графики эмпирических функций распределения. Регрессионный и корреляционный анализ сопровождался построением выборочных прямых среднеквадратической регрессии и выборочных корреляционных кривых соответственно. В рамках кластерного анализа были исследованы алгоритмы  $k$ -средних и поиска сгущений.

# 1. ВЫРАВНИВАНИЕ СТАТИСТИЧЕСКИХ РЯДОВ

На первом этапе работы было произведено формирование выборки из генеральной совокупности и выполнена подготовка выборочных данных к проведению статистического анализа. После проведенной подготовки были найдены точечные статистические оценки выборочной средней, выборочной дисперсии, исправленной выборочной дисперсии, выборочного СКВО, асимметрии и эксцесса, моды и медианы. Также была выполнена проверка гипотезы о нормальном законе по критерию Пирсона. Все необходимые вычисления производились при помощи, разработанной на языке Python, программы, исходный код которой представлен в приложении А.

## 1.1. Основные теоретические положения

*Выборка. Последовательность действий при предварительной обработке выборки.*

Виды выборок: *повторная* и *бесповторная*. Способы формирования: *с разбиением генеральной совокупности на части* и *без*. *Статистический ряд* – последовательность элементов выборки, расположенных в порядке их получения (наблюдения). *Ранжированный ряд* – последовательность элементов выборки, расположенных в порядке возрастания их значений. *Вариационный ряд* – получается из ранжированного ряда в результате объединения одинаковых элементов.

*Построение интервального ряда.*

Количество интервалов:  $k = 1 + 3.31 \cdot \lg N$ . Полученное значение округляется до нечетного целого. Ширина интервала:  $h = \frac{x_{\max} - x_{\min}}{k}$ . Границы интервалов:  $[x_{\min} + (i - 1) \cdot h; x_{\min} + i \cdot h], i = 1, 2, \dots, k - 1$ , где  $i$  – номер интервала. Последний интервал:  $[x_{\min} + (k - 1) \cdot h; x_{\min} + k \cdot h]$ . Также необходима частота попадания значений вариант в интервалы и середины интервалов  $\tilde{x}_i, i = \overline{1, k}$ .

### *Эмпирическая функция распределения.*

Эмпирической функцией распределения называют функцию  $F^*(x)$ , определяющую для каждого значения  $x$  относительную частоту события  $X < x$ . График  $F^*(x)$  представляет собой лестничный график, длина каждой ступеньки которого равна длине соответствующего интервала, а высота – отношению накопленной частоты до середины этого интервала к объему выборки, т. е.

$$F^*(\tilde{x}_i) = \frac{m_i^{\text{нак.}}}{N}; m_i^{\text{нак.}} = \sum_{j=1}^{i-1} m_j; i = 1, 2, \dots, k.$$

### *Начальные и центральные моменты.*

Начальный эмпирический момент  $k$ -го порядка:  $\bar{M}_k = \frac{1}{N} \sum n_j x_j^k$ . В частности:  $\bar{x}_{\text{в}} = \bar{M}_1 = \frac{1}{N} \sum n_j x_j$ . Центральный эмпирический момент  $k$ -го порядка:  $\bar{m}_k = \frac{1}{N} \sum n_j (x_j - \bar{x}_{\text{в}})^k$ . В частности:  $D_{\text{в}} = \bar{m}_2 = \frac{1}{N} \sum n_j (x_j - \bar{x}_{\text{в}})^2$ . Исправленная оценка дисперсии:  $s^2 = \frac{N}{N-1} D_{\text{в}}$ . Статистические оценки СКВО:  $\sigma_{\text{в}} = \sqrt{D_{\text{в}}}$ ;  $s = \sqrt{s^2}$ . Статистические оценки асимметрии и эксцесса:  $\bar{A}_s = \frac{\bar{m}_3}{s^3}$ ;  $E = \frac{\bar{m}_4}{s^4} - 3$ . Теоретические моменты распределения СВ приравниваются соответствующим эмпирическим моментам того же порядка (метод моментов К. Пирсона); эмпирические моменты являются несмещенными оценками соответствующих им теоретических).

### *Условные эмпирические моменты. Связь условных эмпирических моментов с эмпирическими начальными и центральными моментами.*

Для упрощения вычислений эмпирических моментов вводят в рассмотрение так называемые условные варианты  $u_j = \frac{x_j - C}{h}$ , где  $C$  – условный ноль (значение варианты интервального ряда, являющуюся средней или близкой к средней по значению в этом ряду). Все условные варианты – целые числа. Вводятся в рассмотрение условные моменты  $k$ -го порядка:  $\bar{M}_k^* = \frac{1}{N} \sum n_j \left( \frac{x_j - C}{h} \right)^k = \frac{1}{N} \sum n_j u_j^k$ . Легко показать справедливость следующих соотношений:

$$\begin{aligned}\bar{x}_B &= \bar{M}_1 = \bar{M}_1^* h + C; \bar{m}_2 = (\bar{M}_2^* - (\bar{M}_1^*)^2) h^2 \\ \bar{m}_3 &= (\bar{M}_3^* - 3\bar{M}_2^*\bar{M}_1^* + 2(\bar{M}_1^*)^3) h^3 \\ \bar{m}_4 &= (\bar{M}_4^* - 4\bar{M}_3^*\bar{M}_1^* + 6\bar{M}_2^*(\bar{M}_1^*)^2 - 3(\bar{M}_1^*)^4) h^4\end{aligned}$$

*Доверительный интервал для оценки математического ожидания при неизвестном СКВО.*

Для выборки объема  $N$  значений нормально распределенной случайной величины  $X$  с неизвестным значением  $\sigma = \sigma(X)$  справедливо соотношение:

$$P\left(\bar{x}_B - \frac{t_{\gamma}S}{\sqrt{N}} < a < \bar{x}_B + \frac{t_{\gamma}S}{\sqrt{N}}\right) = \gamma, \quad \text{где } \bar{x}_B \text{ и } S \text{ — выборочное среднее и «исправленное» выборочное СКВО — статистические оценки } a \text{ — математического ожидания и } \sigma \text{ — СКВО случайной величины } X; t = \frac{\bar{x}_B - a}{S/\sqrt{N}} \text{ (CB, распределенная по закону } S(t, N) \text{ — закону Стьюдента с } k = N - 1 \text{ степенями свободы). В результате требуемый доверительный интервал, покрывающий неизвестное значение параметра } a \text{ с надежностью } \gamma: \left(\bar{x}_B - \frac{t_{\gamma}S}{\sqrt{N}}, \bar{x}_B + \frac{t_{\gamma}S}{\sqrt{N}}\right). \text{ Здесь значение } t_{\gamma}(N) \text{ — табличная величина.}$$

*Доверительный интервал для оценки СКВО.*

Для выборки объема  $N$  значений нормально распределенной случайной величины  $X$  доверительный интервал, покрывающий с надежностью  $\gamma$  неизвестное значение параметра  $\sigma$ :  $P(S - \delta < \sigma < S + \delta) = \gamma$  или  $S - \delta < \sigma < S + \delta$ , где  $S$  — «исправленной» выборочное СКВО — статистическая оценка  $\sigma$  — СКВО случайной величины  $X$ . Если  $q = \delta/S$ , то  $S(1 - q) < \sigma < S(1 + q)$ . При  $q < 1$ :  $\frac{1}{S(1+q)} < \frac{1}{\sigma} < \frac{1}{S(1-q)}$ . Умножим последнее неравенство на  $S\sqrt{N-1}$ :  $\frac{\sqrt{N-1}}{(1+q)} < \frac{S\sqrt{N-1}}{\sigma} < \frac{\sqrt{N-1}}{(1-q)}$ . Если  $\chi = \frac{S\sqrt{N-1}}{\sigma}$ , то:  $\frac{\sqrt{N-1}}{(1+q)} < \chi < \frac{\sqrt{N-1}}{(1-q)}$ . Потребуем, чтобы вероятность выполнения двойного неравенства равнялась  $\gamma$ .  $q(\gamma, N)$  — табличная величина. В результате требуемый доверительный интервал  $(S(1 - q), S(1 + q))$ , показывающий неизвестное значение параметра  $\sigma$  с надежностью  $\gamma$ , будет построен.

### *Проверка статистической гипотезы о законе распределения.*

Гипотеза  $H_0$  – выборочные данные представляют значения случайной величины, распределенной по нормальному закону распределения. В качестве критерия проверки гипотезы будет использоваться критерий Пирсона  $\chi^2$ . По нему вычисляется «наблюденное» значение случайной величины  $\chi^2$ :  $\chi^2_{\text{набл}} = \sum_{i=1}^K \frac{(n_i - n'_i)^2}{n'_i}$ , где  $n'_i$  – теоретические частоты – частоты, с которыми нормально распределенная величина с параметрами  $\bar{x}_B$  и  $S$  попадала бы в  $i$ -й интервал интервального ряда при проведении  $N$  испытаний. По числу степеней свободы ( $k = K - 3$ ) и уровню значимости вычисляется значение  $\chi^2_{\text{крит}} = \chi^2(\alpha, k)$  – табличная величина. Область принятия гипотезы  $H_0$  определяется условием:  $\chi^2_{\text{набл}} \leq \chi^2_{\text{крит}}$ . Для вычисления значения  $\chi^2_{\text{набл}}$  необходимо вычислить значения «теоретических» частот  $n'_i$ . Для этого можно воспользоваться следующей последовательностью действий:

1. Пересчет границ интервалов интервального ряда. Первый интервал:  $(z_1 = -\infty)$ ; последний интервал:  $(z_{K+1} = +\infty)$ ;  $z_i = \frac{x_i - \bar{x}_B}{S}, i = 2, 3, \dots, K$ .
2. Вычисление вероятностей попадания случайной величины в каждый  $i$ -й интервал интервального ряда:  $p_i = \Phi(z_{i+1}) - \Phi(z_i), i = 1, 2, 3, \dots, K$ . Здесь  $\Phi(z)$  – функция Лапласа (табличная величина):  $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_0^z e^{-\frac{t^2}{2}} dt$ ;  $\Phi(-\infty) = -0.5$ ;  $\Phi(+\infty) = 0.5$ .
3. Вычисление значений «теоретических» частот:  $n'_i = N \cdot p_i, i = 1, 2, 3, \dots, K$ .

## **1.2. Формирование и первичная обработка выборки. Ранжированный и интервальный ряды**

В качестве генеральной совокупности были взяты экспериментальные данные, содержащие информацию (за 2015 год) о показаниях различных датчиков при газовых выбросах на одной из турецких электростанций (7384 элемента, рис. 1.1).

Так как каждый элемент представляет собой набор показаний независимых датчиков при газовых выбросах, то целесообразно рассматривать способы формирования выборки, не предполагающие разбиение генеральной совокупности на части. Также отсутствует какая-либо информация об оборудовании, на котором производилось измерение. Из-за достаточно большого объема данных слишком малы шансы повторного попадания элемента в выборку при «отборе с возвратом». Поэтому для выполнения работы была сформирована бесповторная случайная выборка объемом в  $N = 100$  элементов, полученная без разбиения исходной генеральной совокупности на части (рис. 1.2).

Полученная выборка была упорядочена в хронологическом порядке. Полученный в итоге статистический ряд, представленный на рис. 1.3. Данные из статистического ряда, пересортированные в порядке возрастания значений, составляют ранжированный ряд (рис. 1.4). Путем объединения одинаковых элементов ранжированного ряда был получен вариационный ряд. На рис. 1.5 он представлен с соответствующими абсолютными и относительными частотами вхождения каждой варианты в выборку.

1	AT, AP, AH, AFDP, GTEP, TIT, TAT, TEY, CDP, CO, NOx
2	<b>1.9532, 1020.1, 84.985, 2.5304, 20.116, 1048.7, 544.92, 116.27, 10.799, 7.4491, 113.25</b>
3	<b>1.2191, 1020.1, 87.523, 2.3937, 18.584, 1045.5, 548.5, 109.18, 10.347, 6.4684, 112.02</b>
4	<b>0.94915, 1022.2, 78.335, 2.7789, 22.264, 1068.8, 549.95, 125.88, 11.256, 3.6335, 88.147</b>
5	<b>1.0075, 1021.7, 76.942, 2.817, 23.358, 1075.2, 549.63, 132.21, 11.702, 3.1972, 87.078</b>
6	<b>1.2858, 1021.6, 76.732, 2.8377, 23.483, 1076.2, 549.68, 133.58, 11.737, 2.3833, 82.515</b>
7	<b>1.8319, 1021.7, 76.411, 2.841, 23.495, 1076.4, 549.92, 133.58, 11.829, 2.0812, 81.193</b>
8	<b>2.074, 1022, 75.974, 2.7981, 22.945, 1073.7, 549.98, 131.53, 11.687, 2.2529, 83.171</b>
9	<b>1.7824, 1022.6, 73.535, 2.8327, 23.337, 1075.7, 550.01, 133.18, 11.745, 3.735, 85.749</b>
10	<b>1.593, 1023.2, 72.873, 2.8729, 23.654, 1078.5, 550.06, 135.38, 11.772, 3.6398, 86.491</b>
11	<b>1.6819, 1023.8, 72.441, 2.9058, 23.463, 1077.9, 550.12, 134.86, 11.742, 3.5866, 86.328</b>
12	<b>1.9002, 1024.5, 71.376, 2.9126, 23.562, 1078.2, 550.12, 134.98, 11.77, 3.5605, 84.117</b>
13	<b>1.7797, 1025.1, 68.528, 2.8725, 23.276, 1077, 550.03, 134.21, 11.782, 3.6902, 85.317</b>
14	<b>1.1722, 1025.3, 65.626, 2.85, 23.215, 1077.3, 550.09, 134.44, 11.873, 3.1766, 86.431</b>
15	<b>0.72327, 1025.4, 64.393, 2.8395, 23.101, 1076.2, 550.15, 133.82, 11.797, 2.562, 84.708</b>
16	<b>0.48348, 1025.8, 64.144, 2.8746, 23.299, 1078.2, 549.96, 135.36, 11.897, 2.1854, 84.104</b>
17	<b>0.42953, 1025.9, 64.504, 2.866, 23.069, 1077, 549.73, 134.47, 11.805, 2.4286, 83.869</b>
18	<b>0.48238, 1026.1, 63.519, 2.5481, 20.185, 1037.1, 535.4, 115.92, 10.86, 12.659, 118.27</b>
19	<b>0.059447, 1026, 52.124, 2.4651, 18.478, 1048.4, 548.78, 111.2, 10.384, 4.6896, 104.56</b>
20	<b>-0.21598, 1026.7, 51.978, 2.5301, 19.438, 1050.4, 546.71, 115.46, 10.767, 7.0983, 116.96</b>
21	<b>0.0167, 1026.8, 54.805, 3.3399, 27.408, 1085.6, 540.72, 148.05, 12.79, 3.8632, 75.64</b>
22	<b>0.60101, 1026.7, 55.028, 4.0702, 24.002, 1060.507, 71.160, 22.14.750, 2.070, 50.254</b>

Рисунок 1.1 – Исходная генеральная совокупность (файл gt\_2015.csv)

1	<b>16.659,1021.4,69.377,3.6937,23.827,1077.8,550.05,130.48,11.791,2.374,51.495</b>
2	<b>30.528,1009.3,51.416,4.5247,31.219,1099.8,541.22,149.06,13.506,1.682,46.159</b>
3	<b>20.941,1014.9,73.346,3.3241,23.555,1072.9,550.4,126.33,11.515,2.8664,52.072</b>
4	<b>22.047,1008.4,63.769,4.259,31.101,1099.9,541.52,152.55,13.396,1.4137,52.404</b>
5	<b>32.231,1009.1,39.002,4.4552,30.431,1099.9,543.85,148.91,13.277,1.6531,56.062</b>
6	<b>26.307,1016.9,44.445,4.336,30.952,1099.9,541.48,151.3,13.448,1.7587,49.11</b>
7	<b>30.292,1012.2,60.139,4.5506,29.519,1100.0,546.67,147.02,13.07,2.0121,50.392</b>
8	<b>21.405,1014.5,73.099,3.5732,25.925,1086.8,550.05,136.48,12.202,1.916,54.255</b>
9	<b>14.349,1007.2,80.061,4.5422,34.514,1100.0,530.14,160.16,14.101,1.7899,51.932</b>
10	<b>15.929,1019.2,59.913,3.3327,27.995,1083.0,550.1,135.04,12.012,0.84619,59.182</b>
11	<b>4.3862,1023.2,82.002,3.415,22.995,1074.1,549.9,130.14,11.714,3.4533,71.095</b>
12	<b>14.357,1021.0,46.549,3.3704,29.802,1078.1,549.95,131.6,11.769,3.1666,67.437</b>
13	<b>18.752,1009.5,77.188,2.905,20.824,1056.0,549.73,114.68,10.79,3.4933,51.248</b>
14	<b>20.431,1012.5,85.636,3.0984,19.546,1050.2,550.01,109.96,10.493,4.5747,48.34</b>
15	<b>21.345,1016.6,69.022,2.7117,19.365,1049.1,549.89,108.34,10.447,4.4993,53.227</b>
16	<b>0.32864,1020.8,79.49,2.8,20.006,1053.2,548.69,115.66,10.824,8.127,106.02</b>
17	<b>2.7318,1019.7,83.997,3.2741,22.956,1073.0,550.21,131.16,11.607,3.6737,69.206</b>
18	<b>9.7406,1018.0,73.716,3.0919,21.91,1065.8,549.93,123.92,11.276,3.5225,62.773</b>
19	<b>8.9086,1014.7,76.151,3.1966,23.199,1073.2,549.58,129.86,11.625,3.2612,62.75</b>
20	<b>29.053,1010.4,56.689,4.3808,30.297,1100.1,544.22,148.53,13.296,1.1639,48.653</b>
21	<b>20.616,1019.3,74.517,4.2231,30.352,1100.1,542.41,151.23,13.418,2.0811,50.016</b>
~	<b>22.74,1013.3,55.89,3.7203,29.353,1095.3,549.56,141.48,12.713,0.7157,60.208</b>

Рисунок 1.2 – Сформированная бесповторная случайная выборка (файл sample.csv)

1	<b>0.94915,1022.2,78.335,2.7789,22.264,1068.8,549.95,125.88,11.256,3.6335,88.147</b>
2	<b>1.0693,1031.1,63.791,4.1693,34.575,1100.1,529.0,167.5,14.33,2.3791,60.097</b>
3	<b>4.9344,1025.6,81.903,4.1364,33.694,1099.9,530.76,164.64,14.195,2.6888,58.278</b>
4	<b>17.505,991.4,52.45,3.2861,28.27,1073.8,550.09,128.78,11.703,3.5697,64.867</b>
5	<b>17.542,995.67,59.628,3.2987,24.362,1077.1,549.91,131.34,11.81,3.4994,61.826</b>
6	<b>14.287,1008.6,65.791,3.016,26.678,1067.7,549.98,124.41,11.399,1.6836,64.547</b>
7	<b>5.9876,1006.0,77.561,3.0831,24.475,1073.4,550.07,131.11,11.614,2.9033,73.543</b>
8	<b>2.2188,1020.2,68.068,2.9267,22.706,1071.8,550.0,130.19,11.451,3.5693,83.93</b>
9	<b>9.359,1007.4,87.486,2.712,19.592,1047.7,549.52,111.51,10.448,5.6001,75.788</b>
10	<b>6.912,1010.7,85.53,2.6612,20.21,1043.4,548.26,109.49,10.221,5.953,78.847</b>
11	<b>0.32864,1020.8,79.49,2.8,20.006,1053.2,548.69,115.66,10.824,8.127,106.02</b>
12	<b>10.659,1021.9,66.465,3.4305,27.334,1086.5,550.03,139.16,12.172,2.7484,66.689</b>
13	<b>12.665,1004.9,84.381,2.7753,24.801,1048.3,550.22,110.73,10.439,3.4359,58.913</b>
14	<b>14.349,1007.2,80.061,4.5422,34.514,1100.0,530.14,160.16,14.101,1.7899,51.932</b>
15	<b>18.129,1004.2,62.09,3.3528,28.337,1081.5,549.96,134.03,11.939,3.7119,63.463</b>
16	<b>14.036,1013.1,36.163,3.3728,30.851,1085.9,550.22,137.92,12.179,2.0295,70.302</b>
17	<b>16.167,1019.6,45.368,4.3257,37.625,1100.0,536.93,155.05,13.747,3.6825,62.981</b>
18	<b>15.929,1019.2,59.913,3.3327,27.995,1083.0,550.1,135.04,12.012,0.84619,59.182</b>
19	<b>22.151,1010.7,46.756,3.5803,30.856,1084.4,550.35,133.99,12.179,3.362,61.995</b>
20	<b>14.357,1021.0,46.549,3.3704,29.802,1078.1,549.95,131.6,11.769,3.1666,67.437</b>
21	<b>22.74,1013.3,55.89,3.7203,29.353,1095.3,549.56,141.48,12.713,0.7157,60.208</b>
~	<b>22.74,1013.3,55.89,3.7203,29.353,1095.3,549.56,141.48,12.713,0.7157,60.208</b>

Рисунок 1.3 – Полученный статистический ряд (файл stat\_range.csv)

1	<b>0.32864,1020.8,79.49,2.8,20.006,1053.2,548.69,115.66,10.824,8.127,106.02</b>
2	<b>0.94915,1022.2,78.335,2.7789,22.264,1068.8,549.95,125.88,11.256,3.6335,88.147</b>
3	<b>1.0693,1031.1,63.791,4.1693,34.575,1100.1,529.0,167.5,14.33,2.3791,60.097</b>
4	<b>2.2188,1020.2,68.068,2.9267,22.706,1071.8,550.0,130.19,11.451,3.5693,83.93</b>
5	<b>2.7318,1019.7,83.997,3.2741,22.956,1073.0,550.21,131.16,11.607,3.6737,69.206</b>
6	<b>3.773,1031.3,86.165,2.9532,19.32,1030.8,536.56,109.21,10.427,13.64,97.987</b>
7	<b>4.3862,1023.2,82.002,3.415,22.995,1074.1,549.9,130.14,11.714,3.4533,71.095</b>
8	<b>4.9344,1025.6,81.903,4.1364,33.694,1099.9,530.76,164.64,14.195,2.6888,58.278</b>
9	<b>5.8837,1028.7,94.2,3.9831,23.563,1076.9,550.11,131.41,11.771,3.3134,64.738</b>
10	<b>5.9876,1006.0,77.561,3.0831,24.475,1073.4,550.07,131.11,11.614,2.9033,73.543</b>
11	<b>6.1588,1022.6,70.759,4.5285,33.1,1100.0,530.46,164.5,14.15,2.3735,50.436</b>
12	<b>6.912,1010.7,85.53,2.6612,20.21,1043.4,548.26,109.49,10.221,5.953,78.847</b>
13	<b>7.1137,1024.9,82.008,4.475,32.102,1100.0,534.63,161.52,13.964,2.1146,48.446</b>
14	<b>7.3283,1011.3,82.977,3.2802,23.618,1075.3,549.79,131.94,11.618,2.8995,64.237</b>
15	<b>8.0977,1023.4,73.573,4.4425,32.1,1099.9,534.77,160.72,13.943,2.1016,50.292</b>
16	<b>8.9086,1014.7,76.151,3.1966,23.199,1073.2,549.58,129.86,11.625,3.2612,62.75</b>
17	<b>9.359,1007.4,87.486,2.712,19.592,1047.7,549.52,111.51,10.448,5.6001,75.788</b>
18	<b>9.7406,1018.0,73.716,3.0919,21.91,1065.8,549.93,123.92,11.276,3.5225,62.773</b>
19	<b>10.529,1018.9,82.532,2.591,18.599,1046.3,549.9,109.08,10.226,5.8061,64.789</b>
20	<b>10.659,1021.9,66.465,3.4305,27.334,1086.5,550.03,139.16,12.172,2.7484,66.689</b>
21	<b>11.23,1019.5,50.965,3.2959,23.643,1076.7,549.8,131.63,11.77,2.9711,62.291</b>
...	...

Рисунок 1.4 – Ранжированный ряд (файл ranked\_range.csv)

1	<b>0.32864,1020.8,79.49,2.8,20.006,1053.2,548.69,115.66,10.824,8.127,106.02</b>	1	<b>1.0,0.01</b>
2	<b>0.94915,1022.2,78.335,2.7789,22.264,1068.8,549.95,125.88,11.256,3.6335,88.147</b>	2	<b>1.0,0.01</b>
3	<b>1.0693,1031.1,63.791,4.1693,34.575,1100.1,529.0,167.5,14.33,2.3791,60.097</b>	3	<b>1.0,0.01</b>
4	<b>2.2188,1020.2,68.068,2.9267,22.706,1071.8,550.0,130.19,11.451,3.5693,83.93</b>	4	<b>1.0,0.01</b>
5	<b>2.7318,1019.7,83.997,3.2741,22.956,1073.0,550.21,131.16,11.607,3.6737,69.206</b>	5	<b>1.0,0.01</b>
6	<b>3.773,1031.3,86.165,2.9532,19.32,1030.8,536.56,109.21,10.427,13.64,97.987</b>	6	<b>1.0,0.01</b>
7	<b>4.3862,1023.2,82.002,3.415,22.995,1074.1,549.9,130.14,11.714,3.4533,71.095</b>	7	<b>1.0,0.01</b>
8	<b>4.9344,1025.6,81.903,4.1364,33.694,1099.9,530.76,164.64,14.195,2.6888,58.278</b>	8	<b>1.0,0.01</b>
9	<b>5.8837,1028.7,94.2,3.9831,23.563,1076.9,550.11,131.41,11.771,3.3134,64.738</b>	9	<b>1.0,0.01</b>
10	<b>5.9876,1006.0,77.561,3.0831,24.475,1073.4,550.07,131.11,11.614,2.9033,73.543</b>	10	<b>1.0,0.01</b>
11	<b>6.1588,1022.6,70.759,4.5285,33.1,1100.0,530.46,164.5,14.15,2.3735,50.436</b>	11	<b>1.0,0.01</b>
12	<b>6.912,1010.7,85.53,2.6612,20.21,1043.4,548.26,109.49,10.221,5.953,78.847</b>	12	<b>1.0,0.01</b>
13	<b>7.1137,1024.9,82.008,4.475,32.102,1100.0,534.63,161.52,13.964,2.1146,48.446</b>	13	<b>1.0,0.01</b>
14	<b>7.3283,1011.3,82.977,3.2802,23.618,1075.3,549.79,131.94,11.618,2.8995,64.237</b>	14	<b>1.0,0.01</b>
15	<b>8.0977,1023.4,73.573,4.4425,32.1,1099.9,534.77,160.72,13.943,2.1016,50.292</b>	15	<b>1.0,0.01</b>
16	<b>8.9086,1014.7,76.151,3.1966,23.199,1073.2,549.58,129.86,11.625,3.2612,62.75</b>	16	<b>1.0,0.01</b>
17	<b>9.359,1007.4,87.486,2.712,19.592,1047.7,549.52,111.51,10.448,5.6001,75.788</b>	17	<b>1.0,0.01</b>
18	<b>9.7406,1018.0,73.716,3.0919,21.91,1065.8,549.93,123.92,11.276,3.5225,62.773</b>	18	<b>1.0,0.01</b>
19	<b>10.529,1018.9,82.532,2.591,18.599,1046.3,549.9,109.08,10.226,5.8061,64.789</b>	19	<b>1.0,0.01</b>
20	<b>10.659,1021.9,66.465,3.4305,27.334,1086.5,550.03,139.16,12.172,2.7484,66.689</b>	20	<b>1.0,0.01</b>
21	<b>11.23,1019.5,50.965,3.2959,23.643,1076.7,549.8,131.63,11.77,2.9711,62.291</b>	21	<b>1.0,0.01</b>
...	...	...	...

Рисунок 1.5 – Вариационный ряд и соответствующая информация о частотах (файлы var\_range.csv и var\_range\_freq.csv соответственно)

В данном случае (по причине отсутствия повторений) вариационный ряд совпадает с ранжированным.

Для формирования интервального ряда был рассмотрен только первый показатель экспериментальных данных (температура окружающей среды, AT – Ambient temperature). На основании данных выборки были получены значения

количества интервалов  $k$  (количество интервалов было округлено до целого нечетного значения) и их ширина  $h$  по формулам:

$$k = 1 + 3.31 \cdot \lg N \approx 7; h = \frac{x_{\max} - x_{\min}}{k} \approx 4.69448,$$

где  $N = 100$  – объем рассматриваемой выборки, а  $x_{\min} = 0.32864$  и  $x_{\max} = 33.19$  – соответственно минимальное и максимальное значения параметра.

Для каждого из 7-ти интервалов были рассчитаны границы, середина, абсолютная и относительная частоты. Полученный интервальный ряд представлен в таблице 1.1.

Таблица 1.1 – Полученный интервальный ряд

Номер интервала	Границы интервала	Середина интервала	Частота попадания в интервал	Относительная частота
1	[0.329; 5.023)	2.676	8	0.08
2	[5.023; 9.718)	7.37	9	0.09
3	[9.718; 14.412)	12.065	17	0.17
4	[14.412; 19.107)	16.759	16	0.16
5	[19.107; 23.801)	21.454	25	0.25
6	[23.801; 28.496)	26.148	15	0.15
7	[28.496; 33.19]	30.843	10	0.1

Границы интервалов вычислялись по формулам:  $[x_{\min} + (i - 1) \cdot h; x_{\min} + i \cdot h], i = 1, 2, \dots, k - 1$ , где  $i$  – номер интервала. Границы последнего интервала:  $[x_{\min} + (k - 1) \cdot h; x_{\min} + k \cdot h]$ .

На основании интервального ряда были построены графики и гистограммы абсолютных и относительных частот (рис. 1.6 и 1.7 соответственно). Также для каждого интервала были вычислены эмпирические функции распределения (рис. 1.8).

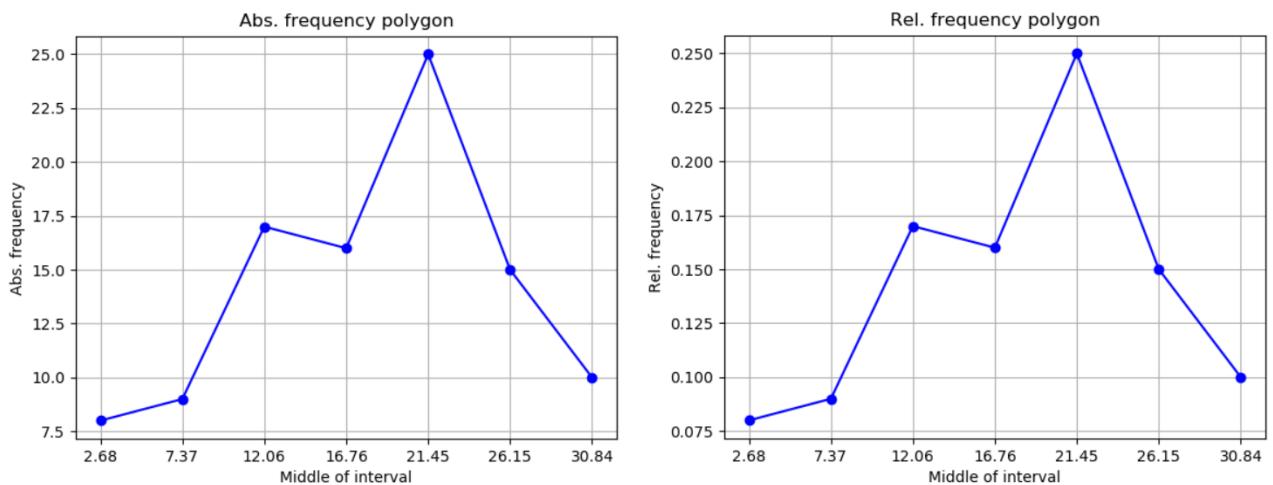


Рисунок 1.6 – Полигоны абсолютных и относительных частот  
интервального ряда

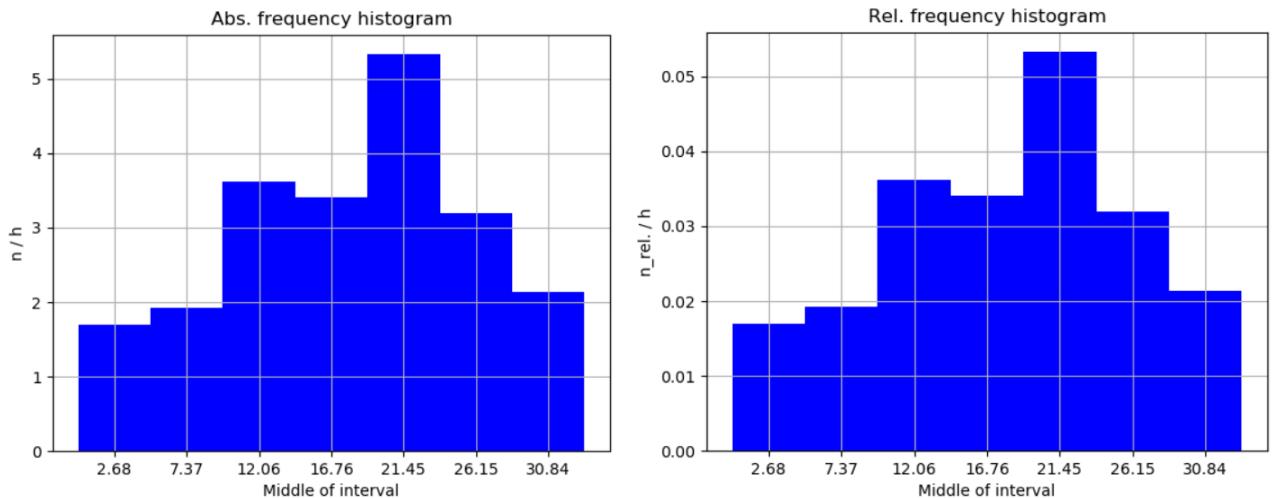


Рисунок 1.7 – Гистограммы абсолютных и относительных частот  
интервального ряда

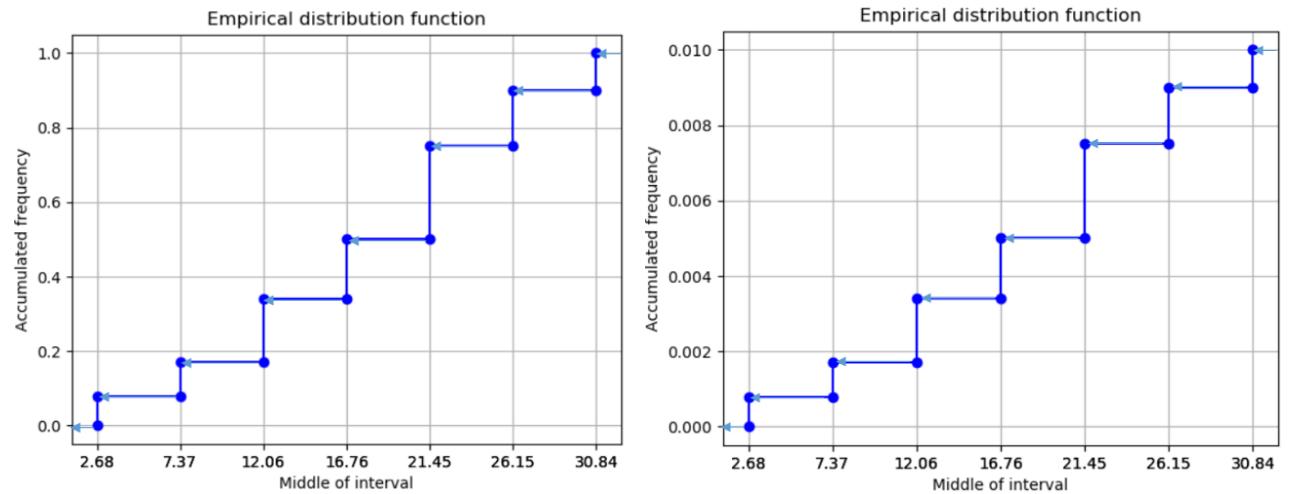


Рисунок 1.8 – Графики эмпирических функций распределения  
абсолютных и относительных частот

В связи с нормированием частот по объему выборки, полученные графики для относительных частот «сжаты» по сравнению с аналогичными для абсолютных (вертикально масштабированы). Это обусловлено тем, что все точки на них находятся в промежутке ординат, который в  $N = 100$  раз меньше соответствующего промежутка на графиках абсолютных частот.

### 1.3. Нахождение точечных оценок параметров распределения

Для имеющегося интервального ряда (см. табл. 1.1) были вычислены условные варианты:  $u_j = \frac{x_j - C}{h}$ , где  $j$  – индекс интервала,  $h$  – ширина интервала ( $h \approx 4.69448$ ), а  $C$  – условный ноль ( $C = x_4 = 16.75932$ ). В качестве вариант в данном случае выступают середины интервалов. С помощью найденных условных вариантов были вычислены условные эмпирические моменты  $\bar{M}_k^*$ , а с помощью условных эмпирических моментов – центральные эмпирические моменты  $\bar{m}_k$  ( $N = 100$  – объем выборки,  $n_j$  – абсолютная частота на  $j$ -м интервале). Результаты вычислений представлены в табл. 1.2.

Таблица 1.2 – Условные эмпирические и центральные эмпирические моменты

$k$	1	2	3	4
$\bar{M}_k^*$	0.26	3	1.1	18.84
$\bar{m}_k$	17.98	64.62	-124.65	9178.91

Были найдены выборочное среднее  $\bar{x}_{\text{в}}$  и дисперсия  $D_{\text{в}}$  с помощью условных вариантов:

$$\bar{x}_{\text{в}} = \bar{M}_1^* h + C \approx 17.9798848; D_{\text{в}} = \bar{m}_2 \approx 64.62464898$$

Эти величины также были вычислены при помощи стандартных формул:

$$\bar{x}_{\text{в}} = \frac{1}{N} \sum n_j x_j = 17.9798848; D_{\text{в}} = \frac{1}{N} \sum n_j (x_j - \bar{x}_{\text{в}})^2 \approx 64.62464898$$

Полученные разными способами значения практически совпадают. Более точные значения приведены на рис. 1.9. Незначительные различия в результатах вычислений могут быть обусловлены слишком высокой точностью языка Python,

из-за чего переменные в этом языке могут не настолько точно интерпретироваться пользовательскими компьютерами.

```
Выборочное среднее, вычисленное с помощью усл. вариант: 17.979884799999997
Дисперсия, вычисленная с помощью усл. вариант: 64.62464898020092
Выборочное среднее, вычисленное с помощью стандартной формулы: 17.9798848
Дисперсия, вычисленная с помощью стандартной формулы: 64.62464898020094
```

Рисунок 1.9 – Программные результаты вычислений выборочного среднего и дисперсии

С помощью полученных величин была найдена исправленная оценка дисперсии, а также статистические оценки СКО (среднеквадратичных отклонений):

$$s^2 = \frac{N}{N-1} D_B \approx 65.27742321; \sigma_B = \sqrt{D_B} \approx 8.03894576; s = \sqrt{s^2} \approx 8.07944449$$

Далее были найдены статистические оценки коэффициентов асимметрии и эксцесса:

$$\bar{A}_s = \frac{\bar{m}_3}{s^3} \approx -0.23634708; E = \frac{\bar{m}_4}{s^4} - 3 \approx -0.84590488$$

В связи с тем, что  $\bar{A}_s < 0$  и  $|\bar{A}_s| < 0.25$ , распределение можно охарактеризовать как «скошенное влево» и незначительно асимметричное. Так как  $E < 0$ , то говорят, что эмпирическое распределение низкое и пологое относительно «эталонного» нормального распределения.

Модой данного распределения будет являться 5-я варианта (в данном случае – середина 5-го интервала), т. к. именно у этого интервала наибольшая абсолютная частота вхождения выборочных данных ( $n_5 = 25$ ). Т. о.,  $M_0 \approx 21.4538$ . Медиана рассматриваемого распределения – середина 4-го (центрального) интервала:  $m_e = 16.75932$ .

#### **1.4. Нахождение интервальных оценок параметров распределения. Проверка статистической гипотезы о нормальном распределении**

Сначала был вычислен доверительный интервал для математического ожидания при неизвестном СКВО:  $\left(\bar{x}_B - \frac{t_{\gamma/2}}{\sqrt{N}}, \bar{x}_B + \frac{t_{\gamma/2}}{\sqrt{N}}\right)$ , где  $\bar{x}_B = 17.9798848$ .

Исправленное выборочное СКВО –  $S \approx 8.079444$ ; объем рассматриваемой выборки –  $N = 100$ ; коэффициент Стьюдента  $t_\gamma = 1.984$  для заданных  $\gamma$  и  $N$ . Таким образом, полученный доверительный интервал:

$$(16.376923, 19.582847)$$

То есть, с вероятностью  $\gamma = 0.95$  данный интервал покрывает истинное значение математического ожидания рассматриваемого распределения.

Далее были вычислены границы доверительного интервала для среднеквадратичного отклонения:  $(S(1 - q), S(1 + q))$ . Так как для заданных  $\gamma$  и  $N$  табличная величина  $q = 0.143$ , то искомый интервал будет иметь вид:

$$(6.924084, 9.234805)$$

То есть, с вероятностью  $\gamma = 0.95$  данный интервал покрывает истинное значение среднеквадратичного отклонения рассматриваемого распределения.

Также была проверена гипотеза о нормальности имеющегося распределения с помощью критерия Пирсона. Для этого был рассмотрен полученный ранее в работе интервальный ряд (см. табл. 1.1).

Для каждого из имеющихся ( $K = 7$ ) интервалов были рассчитаны соответствующие значения  $z_i = \frac{x_i - \bar{x}_B}{S}$ ,  $i = 2, 3, \dots, K$  и, аналогично,  $z_{i+1}$ , причем,  $z_1 = -\infty$  и  $z_{K+1} = +\infty$ .

После пересчета границ интервалов были найдены (на основании новых значений границ) соответствующие значения функций Лапласа ( $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_0^z e^{-\frac{t^2}{2}} dt$ ):  $\Phi(z_i)$  и  $\Phi(z_{i+1})$ . Величина  $\Phi(z)$  – табличная ( $\Phi(-\infty) = -0.5$ ;  $\Phi(+\infty) = 0.5$ ).

Далее на основании имеющихся значений  $\Phi(z_i)$  и  $\Phi(z_{i+1})$  были найдены значения вероятностей попаданий случайной величины в каждый  $i$ -й интервал интервального ряда:  $p_i = \Phi(z_{i+1}) - \Phi(z_i)$ ,  $i = 1, 2, 3, \dots, K$ .

По полученным вероятностям были рассчитаны теоретические частоты  $n'_i = N \cdot p_i$ ,  $i = 1, 2, 3, \dots, K$ , после чего все необходимые для дальнейшего выполнения поставленной задачи величины были занесены в табл. 1.3.

Таблица 1.3 – Все величины, необходимые для проверки гипотезы о нормальности распределения с помощью критерия Пирсона

Номер интервала $K$	Значение $x_i$	Значение $x_{i+1}$	Значение $n_i$	Значение $z_i$	Значение $z_{i+1}$	Значение $\Phi(z_i)$	Значение $\Phi(z_{i+1})$	Значение $p_i$	Значение $n'_i$
1	0.329	5.023	8	$-\infty$	-1.604	-0.5	-0.4452	0.0548	5.48
2	5.023	9.718	9	-1.604	-1.023	-0.4452	-0.3461	0.0991	9.91
3	9.718	14.412	17	-1.023	-0.442	-0.3461	-0.17	0.1761	17.61
4	14.412	19.107	16	-0.442	0.139	-0.17	0.0557	0.2257	22.57
5	19.107	23.801	25	0.139	0.72	0.0557	0.2642	0.2085	20.85
6	23.801	28.496	15	0.72	1.302	0.2642	0.4032	0.139	13.9
7	28.496	33.19	10	1.302	$+\infty$	0.4032	0.5	0.0968	9.68

На основании теоретических частот было рассчитано наблюдаемое значение критерия Пирсона:  $\chi^2_{\text{набл}} = \sum_{i=1}^K \frac{(n_i - n'_i)^2}{n'_i} \approx 4.099662$ . Также было найдено табличное значение критической точки по заданному уровню значимости  $\alpha = 0.05$  и числу степеней свободы  $k = K - 3 = 4$ :  $\chi^2_{\text{крит}} = \chi^2(\alpha, k) = 9.5$ .

Таким образом, выполняется (исходя из данных табл. 1.3) условия  $\sum_{i=1}^K n'_i = N$  и  $\chi^2_{\text{набл}} \leq \chi^2_{\text{крит}}$ , так как  $4.099662 \leq 9.5$ . Следовательно, на уровне значимости  $\alpha = 0.05$  предполагаемая гипотеза о нормальности распределения принимается.

## **1.5. Выводы**

В рамках данного раздела была составлена из заданной генеральной совокупности и подготовлена к статистическому анализу случайная бесповторная выборка. Из построенной выборочной последовательности были получены статистический, ранжированный, вариационный и интервальный ряды. Для последнего были построены эмпирические функции, графические полигоны и гистограммы для абсолютных и относительных частот.

Также были найдены точечные статистические оценки выборочной средней, выборочной дисперсии, исправленной выборочной дисперсии, выборочного СКВО, асимметрии и эксцесса, моды и медианы. Значения выборочного среднего и дисперсии, полученные при помощи стандартной формулы и условных вариант, практически совпали. Незначительные различия обусловлены слишком высокой точностью выбранного языка программирования. Исследуемое распределение обладает незначительной асимметрией; оно низкое и пологое относительно «эталонного» нормального распределения.

Была выполнена проверка гипотезы о нормальном законе по критерию Пирсона. Выполнение условия  $\chi^2_{\text{набл}} \leq \chi^2_{\text{крит}}$ , которое было достигнуто в ходе выполнения работы, позволяет принять гипотезу о нормальности рассматриваемого распределения на уровне значимости  $\alpha = 0.05$ .

## 2. КОРРЕЛЯЦИОННЫЙ И РЕГРЕССИОННЫЙ АНАЛИЗ

В ходе данного этапа выполнения работы была сформирована двумерная выборка, для которой был произведен корреляционный и регрессионный анализ. Все вычисления так же, как и на предыдущем этапе, производились при помощи разработанной на языке программирования Python программы (исходный код представлен в приложении Б).

### 2.1. Основные теоретические положения

*Корреляционная зависимость, корреляционный момент, коэффициент корреляции. Статистическая оценка коэффициента корреляции.*

Корреляционной называют статистическую зависимость двух случайных величин, при которой изменение значения одной из случайных величин приводит к изменению математического ожидания другой случайной величины:  $M(X/y) = q_1(y); M(Y/x) = q_2(x)$ . Функции  $q_1(y)$  и  $q_2(x)$  – функциями регрессии. Корреляционный момент:  $\mu_{xy} = M\{[x - M(X)] \cdot [y - M(Y)]\}$ . Коэффициент корреляции:  $r_{xy} = \frac{\mu_{xy}}{\sigma_x \sigma_y}; |r_{xy}| \leq 1$ . Случайные величины  $X$  и  $Y$  называют коррелированными (и, как следствие, зависимыми), если их корреляционный момент или их коэффициент корреляции отличен от нуля то есть  $X$  и  $Y$  коррелированные  $\Rightarrow$  СВ  $X$  и  $Y$  зависимы;  $X$  и  $Y$  независимы  $\Rightarrow$  СВ  $X$  и  $Y$  некоррелированные.

Значение  $\bar{r}_{xy}$  – статистической оценки  $r_{xy}$  – коэффициента корреляции можно вычислить по формуле:  $\bar{r}_{xy} = \frac{\sum_{i=1}^{K_y} \sum_{j=1}^{K_x} n_{ij} y_i x_j - N \bar{x}_b \bar{y}_b}{N S_x S_y}$ . При  $N > 50$  в случае нормального распределения системы случайных величин  $\{X, Y\}$  для оценки значения  $r_{xy}$  можно использовать соотношение:

$$\bar{r}_{xy} - 3 \frac{1 - \bar{r}_{xy}^2}{\sqrt{N}} \leq r_{xy} \leq \bar{r}_{xy} + 3 \frac{1 + \bar{r}_{xy}^2}{\sqrt{N}}$$

*Доверительный интервал для выборочного коэффициента корреляции.*

С помощью преобразования Фишера перейдем к случайной величине  $z$ :

$$\bar{z} = 0.5 \ln \frac{1+\bar{r}_{xy}}{1-\bar{r}_{xy}} = 1.1513 \lg \frac{1+\bar{r}_{xy}}{1-\bar{r}_{xy}}. \quad \text{Распределение } z \text{ при неограниченном}$$

возрастании объема выборки асимптотически нормальное со значением СКВО:

$\bar{\sigma}_z = \frac{1}{\sqrt{N-3}}$ . В результате, доверительный интервал для  $r_{xy}$  генеральной совокупности с доверительной вероятностью  $\gamma$  определяют по схеме:

1. Вычисляют выборочное значение  $\bar{z}$ ;
2. Вычисляют значение  $\bar{\sigma}_z$ ;
3. Доверительный интервал для генерального значения представляется в виде:  $(\bar{z} - \lambda(\gamma)\bar{\sigma}_z, \bar{z} + \lambda(\gamma)\bar{\sigma}_z)$ ;  $\Phi[\lambda(\gamma)] = \frac{\gamma}{2}$ ;

4. Для пересчета интервала в доверительный интервал для коэффициента корреляции с тем же значением  $\gamma$  необходимо воспользоваться обратным преобразованием Фишера:  $r = \text{th}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = \frac{e^{2z} - 1}{e^{2z} + 1}$ .

*Проверка гипотезы о значимости выборочного коэффициента корреляции.*

При выборке объема  $N$  значений двумерной нормально распределенной случайной величины  $\{X, Y\}$  и  $\bar{r}_{xy} \neq 0$  ( $\bar{r}_{xy}$  – случайная величина) не утверждается, что  $r_{xy}$  – коэффициент корреляции для генеральной совокупности тоже отличен от нуля. Возникает необходимость проверить гипотезу  $H_0: r_{xy} = 0$ .

Альтернативной будет гипотеза  $H_1: r_{xy} \neq 0$ . Если основная гипотеза  $H_0$  отвергается, то это означает, что выборочный коэффициент корреляции  $\bar{r}_{xy}$  значимо отличается от нуля (значим). В противном случае -  $\bar{r}_{xy}$  не значим.

Критерия проверки статистической гипотезы о значимости выборочного коэффициента корреляции – случайная величина:  $T = \bar{r}_{xy}\sqrt{N-2}/\sqrt{1-\bar{r}_{xy}^2}$ . При справедливости нулевой гипотезы  $H_0$  случайная величина  $T$  распределена по закону Стьюдента с  $k = N - 2$  степенями свободы. Проверка гипотезы осуществляется по схеме:

1. По представленной выше формуле вычисляется значение  $T_{\text{набл}}$ ;
2. Определяется табличное значение  $t_{\text{крит}}(\alpha, k)$ ;
3. Если  $|T_{\text{набл}}| \leq t_{\text{крит}}(\alpha, k)$  – нет оснований отвергать гипотезу  $H_0$ ;

Если  $|T_{\text{набл}}| > t_{\text{крит}}(\alpha, k)$  – основная гипотеза  $H_0$  с выборочными данными должна быть отвергнута.

### *Метод наименьших квадратов.*

Метод наименьших квадратов (МНК) – метод, основанный на поиске минимума суммы квадратов отклонений значений некоторых функций от заданного множества значений.

Рассмотрим, например, следующую ситуацию:

- Задано  $m$  значений  $y_i, i = 1, 2, \dots, m$ ;
- Задано  $m$  функций  $f_i(x_1, x_2, \dots, x_n), i = 1, 2, \dots, m; n < m$ ;
- Требуется найти такие значения  $x_1^*, x_2^*, \dots, x_n^*$ , при которых сумма квадратов отклонений значений функций  $f_i(x_1, x_2, \dots, x_n)$  от  $y_i, i = 1, 2, \dots, m$  была бы минимально возможной.

По сути требуется найти решение системы уравнений:  $f_i(x_1, x_2, \dots, x_n) = y_i, i = 1, 2, \dots, m$  или (что эквивалентно) решить оптимизационную задачу:  $\phi(x_1^*, x_2^*, \dots, x_n^*) = \min_{x \in X} \{\sum_{i=1}^m (f_i(x_1, x_2, \dots, x_n) - y_i)^2\}$ .

Рассмотрим другую ситуацию:

- Задано  $m$  пар значений  $(x_i, y_i), i = 1, 2, \dots, m$ ;
- Задана функция  $f(x, a, b, c)$ ;
- Требуется найти такие значения  $a^*, b^*, c^*$ , при которых сумма квадратов отклонений значений функции  $f(x, a, b, c)$  от  $y_i$ , при  $x = x_i, i = 1, 2, \dots, m$  была бы минимально возможной.

Здесь требуется найти решение системы уравнений:  $f(x, a, b, c) = y_i, i = 1, 2, \dots, m$  или (что эквивалентно) решить оптимизационную задачу:  $\phi(a^*, b^*, c^*) = \min_{a,b,c} \{\sum_{i=1}^m (f(x_i, a, b, c) - y_i)^2\}$ .

### *Выборочные прямые среднеквадратической регрессии.*

В случае, когда известна только двумерная выборка значений случайных величин  $X$  и  $Y$ , возможно построение только выборочных прямых среднеквадратической регрессии. Уравнения выборочных прямых среднеквадратической регрессии:

$$\bar{y}_x = \bar{y}_{\text{в}} + \bar{r}_{xy} \frac{S_y}{S_x} (x - \bar{x}_{\text{в}}); \bar{x}_y = \bar{x}_{\text{в}} + \bar{r}_{xy} \frac{S_x}{S_y} (y - \bar{y}_{\text{в}})$$

### *Корреляционное отношение.*

Каждую из групп данных (из имеющейся двумерной выборки) можно (при определенных условиях) рассматривать как отдельную выборку и определить для каждой группы групповую выборочную среднюю и групповую выборочную дисперсию. Внутригрупповая дисперсия  $D_{\text{вн гр}}$  – взвешенная по объемам групп среднее арифметическое групповых дисперсий. Межгрупповая дисперсия  $D_{\text{межгр}}$  – дисперсия условных (групповых) средних  $\bar{x}_{y_i}$  относительно выборочной средней  $\bar{x}_{\text{в}}$ . Оценка общей дисперсии  $X$ :  $D_{\text{общ}} = D_{\text{межгр}} + D_{\text{вн гр}}$ .

Если  $D_{\text{межгр}} = 0$ , то все условные (групповые) средние равны и не зависят от выборочных значений  $y$ . В этом случае между исследуемыми случайными величинами не имеет место корреляционная зависимость. Если  $D_{\text{вн гр}} = 0$ , то в каждой из групп (в данном случае в каждой строке корреляционной таблицы) имеется только одно значение  $X$ . В этом случае между исследуемыми случайными величинами функциональная зависимость.

Определим  $\bar{\eta}_{xy}$  – выборочное корреляционное отношение  $X$  к  $Y$ :  $\bar{\eta}_{xy} = \frac{\bar{\sigma}_{\bar{x}y}}{\bar{\sigma}_x} = \frac{\sqrt{D_{\text{межгр}}}}{\sqrt{D_{\text{общ}}}}$ . Аналогично для  $\bar{\eta}_{yx} = \frac{\bar{\sigma}_{\bar{y}x}}{\bar{\sigma}_y}$ .

*Свойства выборочного корреляционного отношения* (имеют место как для  $\bar{\eta}_{yx}$ , так и для  $\bar{\eta}_{xy}$ ):

1.  $0 \leq \bar{\eta} \leq 1$ ;
2.  $\bar{\eta} = 0$  – случайные величины  $X$  и  $Y$  не связаны корреляционной зависимостью;

3.  $\bar{\eta} = 1$  – случайные величины  $X$  и  $Y$  связаны функциональной зависимостью;

4.  $\bar{\eta} \geq |\bar{r}_{xy}|$ ;

5.  $\bar{\eta} = |\bar{r}_{xy}|$  – случайные величины  $X$  и  $Y$  связаны линейной корреляционной зависимостью;

6. Корреляционное отношение является количественной мерой тесноты корреляционной зависимости между случайными величинами  $X$  и  $Y$ . Вместе с тем, кроме п.5, оно не позволяет определить характер корреляционной зависимости.

*Построение уравнений выборочных кривых для параболической среднеквадратической регрессии.*

Выборочное уравнение регрессии  $Y$  на  $X$ :  $\bar{y}_x = ax^2 + bx + c$ . Значения коэффициентов  $a$ ,  $b$  и  $c$  определим из системы, полученной с помощью МНК:

$$\begin{cases} \left( \sum_{i=1}^m n_{x_i} x_i^4 \right) a + \left( \sum_{i=1}^m n_{x_i} x_i^3 \right) b + \left( \sum_{i=1}^m n_{x_i} x_i^2 \right) c = \sum_{i=1}^m n_{x_i} \bar{y}_{x_i} x_i^2 \\ \left( \sum_{i=1}^m n_{x_i} x_i^3 \right) a + \left( \sum_{i=1}^m n_{x_i} x_i^2 \right) b + \left( \sum_{i=1}^m n_{x_i} x_i \right) c = \sum_{i=1}^m n_{x_i} \bar{y}_{x_i} x_i \\ \left( \sum_{i=1}^m n_{x_i} x_i^2 \right) a + \left( \sum_{i=1}^m n_{x_i} x_i \right) b + Nc = \sum_{i=1}^m n_{x_i} \bar{y}_{x_i} \end{cases}$$

## 2.2. Проверка статистической гипотезы о равенстве коэффициента корреляции нулю

Был составлен интервальный ряд по, отличному от рассматриваемого до этого, измерению. Все необходимые действия для формирования рядов проводились относительно уже второго измерения (давления окружающей среды, AP – Ambient pressure) в то время, как до этого рассматривалась температура среды (Ambient temperature, AT). Уже имеющийся статистический ряд (см. рис. 1.3), полученный путем упорядочивания случайной бесповторной выборки объема  $N = 100$  в порядке получения измерений, был преобразован в новый ранжированный ряд, в котором элементы выборки расположены в

порядке возрастания значений уже второго измерения. Новый ранжированный ряд представлен на рис. 2.1.

1	<b>17.505, 991.4, 52.45, 3.2861, 28.27, 1073.8, 550.09, 128.78, 11.703, 3.5697, 64.867</b>
2	<b>17.542, 995.67, 59.628, 3.2987, 24.362, 1077.1, 549.91, 131.34, 11.81, 3.4994, 61.826</b>
3	<b>19.302, 1002.5, 91.297, 3.4731, 24.192, 1076.4, 550.32, 129.45, 11.787, 1.5438, 44.221</b>
4	<b>16.055, 1004.0, 83.661, 3.1968, 21.224, 1060.4, 549.72, 118.81, 11.019, 3.3532, 54.957</b>
5	<b>18.129, 1004.2, 62.09, 3.3528, 28.337, 1081.5, 549.96, 134.03, 11.939, 3.7119, 63.463</b>
6	<b>12.665, 1004.9, 84.381, 2.7753, 24.801, 1048.3, 550.22, 110.73, 10.439, 3.4359, 58.913</b>
7	<b>29.18, 1005.3, 54.655, 4.4693, 30.931, 1099.9, 542.83, 148.97, 13.375, 1.3316, 53.928</b>
8	<b>25.755, 1005.7, 66.673, 4.4612, 30.51, 1100.1, 543.85, 148.52, 13.298, 1.8928, 58.907</b>
9	<b>5.9876, 1006.0, 77.561, 3.0831, 24.475, 1073.4, 550.07, 131.11, 11.614, 2.9033, 73.543</b>
10	<b>24.761, 1006.1, 41.086, 4.2744, 30.924, 1099.9, 541.79, 152.8, 13.396, 1.4347, 54.649</b>
11	<b>32.648, 1006.1, 51.63, 4.2385, 29.29, 1099.7, 548.41, 144.44, 12.955, 2.8659, 50.615</b>
12	<b>32.852, 1006.5, 45.314, 4.4436, 29.869, 1100.0, 546.53, 146.77, 13.055, 1.8117, 56.64</b>
13	<b>23.474, 1006.6, 79.67, 4.3696, 30.062, 1100.1, 544.99, 148.51, 13.245, 1.583, 57.285</b>
14	<b>20.925, 1007.0, 60.806, 4.2693, 31.2, 1099.9, 541.15, 153.06, 13.467, 1.2973, 53.532</b>
15	<b>14.349, 1007.2, 80.061, 4.5422, 34.514, 1100.0, 530.14, 160.16, 14.101, 1.7899, 51.932</b>
16	<b>27.733, 1007.2, 56.531, 4.009, 25.032, 1080.7, 550.02, 129.63, 12.012, 2.0733, 54.622</b>
17	<b>23.175, 1007.3, 84.395, 4.4342, 30.265, 1100.1, 544.0, 148.84, 13.193, 0.70905, 53.523</b>
18	<b>24.503, 1007.3, 74.401, 3.823, 24.514, 1078.6, 549.93, 129.76, 11.826, 2.1178, 44.868</b>
19	<b>9.359, 1007.4, 87.486, 2.712, 19.592, 1047.7, 549.52, 111.51, 10.448, 5.6001, 75.788</b>
20	<b>33.19, 1007.5, 43.443, 4.6012, 29.647, 1099.9, 546.94, 146.86, 13.062, 1.6753, 56.48</b>
21	<b>18.426, 1008.1, 73.549, 3.0869, 19.609, 1049.4, 550.33, 109.97, 10.46, 4.1164, 52.278</b>

Рисунок 2.1 – Ранжированный ряд (файл ranked\_range\_2.csv)

Далее путем удаления повторяющихся элементов выборки и запоминания их частот был сформирован вариационный ряд (вместе с соответствующей таблицей абсолютных и относительных частот представлен на рис. 2.2). Так как выборка бесповторная, то вариационный и ранжированный ряды совпадают.

1	<b>17.505, 991.4, 52.45, 3.2861, 28.27, 1073.8, 550.09, 128.78, 11.703, 3.5697, 64.867</b>	1	<b>1.0, 0.01</b>
2	<b>17.542, 995.67, 59.628, 3.2987, 24.362, 1077.1, 549.91, 131.34, 11.81, 3.4994, 61.826</b>	2	<b>1.0, 0.01</b>
3	<b>19.302, 1002.5, 91.297, 3.4731, 24.192, 1076.4, 550.32, 129.45, 11.787, 1.5438, 44.221</b>	3	<b>1.0, 0.01</b>
4	<b>16.055, 1004.0, 83.661, 3.1968, 21.224, 1060.4, 549.72, 118.81, 11.019, 3.3532, 54.957</b>	4	<b>1.0, 0.01</b>
5	<b>18.129, 1004.2, 62.09, 3.3528, 28.337, 1081.5, 549.96, 134.03, 11.939, 3.7119, 63.463</b>	5	<b>1.0, 0.01</b>
6	<b>12.665, 1004.9, 84.381, 2.7753, 24.801, 1048.3, 550.22, 110.73, 10.439, 3.4359, 58.913</b>	6	<b>1.0, 0.01</b>
7	<b>29.18, 1005.3, 54.655, 4.4693, 30.931, 1099.9, 542.83, 148.97, 13.375, 1.3316, 53.928</b>	7	<b>1.0, 0.01</b>
8	<b>25.755, 1005.7, 66.673, 4.4612, 30.51, 1100.1, 543.85, 148.52, 13.298, 1.8928, 58.907</b>	8	<b>1.0, 0.01</b>
9	<b>5.9876, 1006.0, 77.561, 3.0831, 24.475, 1073.4, 550.07, 131.11, 11.614, 2.9033, 73.543</b>	9	<b>1.0, 0.01</b>
10	<b>24.761, 1006.1, 41.086, 4.2744, 30.924, 1099.9, 541.79, 152.8, 13.396, 1.4347, 54.649</b>	10	<b>1.0, 0.01</b>
11	<b>32.648, 1006.1, 51.63, 4.2385, 29.29, 1099.7, 548.41, 144.44, 12.955, 2.8659, 50.615</b>	11	<b>1.0, 0.01</b>
12	<b>32.852, 1006.5, 45.314, 4.4436, 29.869, 1100.0, 546.53, 146.77, 13.055, 1.8117, 56.64</b>	12	<b>1.0, 0.01</b>
13	<b>23.474, 1006.6, 79.67, 4.3696, 30.062, 1100.1, 544.99, 148.51, 13.245, 1.583, 57.285</b>	13	<b>1.0, 0.01</b>
14	<b>20.925, 1007.0, 60.806, 4.2693, 31.2, 1099.9, 541.15, 153.06, 13.467, 1.2973, 53.532</b>	14	<b>1.0, 0.01</b>
15	<b>14.349, 1007.2, 80.061, 4.5422, 34.514, 1100.0, 530.14, 160.16, 14.101, 1.7899, 51.932</b>	15	<b>1.0, 0.01</b>
16	<b>27.733, 1007.2, 56.531, 4.009, 25.032, 1080.7, 550.02, 129.63, 12.012, 2.0733, 54.622</b>	16	<b>1.0, 0.01</b>
17	<b>23.175, 1007.3, 84.395, 4.4342, 30.265, 1100.1, 544.0, 148.84, 13.193, 0.70905, 53.523</b>	17	<b>1.0, 0.01</b>
18	<b>24.503, 1007.3, 74.401, 3.823, 24.514, 1078.6, 549.93, 129.76, 11.826, 2.1178, 44.868</b>	18	<b>1.0, 0.01</b>
19	<b>9.359, 1007.4, 87.486, 2.712, 19.592, 1047.7, 549.52, 111.51, 10.448, 5.6001, 75.788</b>	19	<b>1.0, 0.01</b>
20	<b>33.19, 1007.5, 43.443, 4.6012, 29.647, 1099.9, 546.94, 146.86, 13.062, 1.6753, 56.48</b>	20	<b>1.0, 0.01</b>
21	<b>18.426, 1008.1, 73.549, 3.0869, 19.609, 1049.4, 550.33, 109.97, 10.46, 4.1164, 52.278</b>	21	<b>1.0, 0.01</b>

Рисунок 2.2 – Вариационный ряд и его таблица частот (файлы var\_range\_2.csv и var\_range\_freq\_2.csv соответственно)

На основе полученных данных был сформирован новый интервальный ряд для второго измерения с количеством интервалов  $k = 1 + 3.31 \cdot \lg N \approx 7$  и шириной интервала  $h = \frac{y_{max} - y_{min}}{k} \approx 5.785714$ , где  $y_{min} = 991.4$ , а  $y_{max} = 1031.9$ . Для каждого интервала были рассчитаны границы, середина, абсолютная и относительная частоты. Полученный интервальный ряд представлен в таблице 2.1. Границы интервалов вычислялись по формулам:  $[y_{min} + (i - 1) \cdot h; y_{min} + i \cdot h], i = 1, 2, \dots, k - 1$ , где  $i$  – номер интервала. Границы последнего интервала:  $[y_{min} + (k - 1) \cdot h; y_{min} + k \cdot h]$ .

Таблица 2.1 – Интервальный ряд

Номер инт-ла	Границы интервала	Середина интервала	Частота попадания в интервал	Относительная частота
1	[991.4; 997.186)	994.293	2	0.02
2	[997.186; 1002.971)	1000.079	1	0.01
3	[1002.971; 1008.757)	1005.864	23	0.23
4	[1008.757; 1014.543)	1011.65	28	0.28
5	[1014.543; 1020.329)	1017.436	26	0.26
6	[1020.329; 1026.114)	1023.221	15	0.15
7	[1026.114; 1031.9]	1029.007	5	0.05

На основании интервального ряда были построены полигон и гистограмма абсолютных частот. Также для каждого интервала была вычислена эмпирическая функция. Аналогичные действия были проведены для относительных частот. Совмещенные иллюстрации результатов работы представлены на рис. 2.3 – 2.5.

Далее к полученному интервальному ряду были добавлены условные варианты:  $u_j = \frac{y_j - C}{h}$ , где  $j$  – индекс интервала,  $h$  – его ширина, а  $C$  – условный ноль ( $C = y_4 = 1011.65$ ). В качестве вариант выступают середины интервалов. Внешний вид интервального ряда с соответствующими условными вариантами представлен в таблице 2.2.

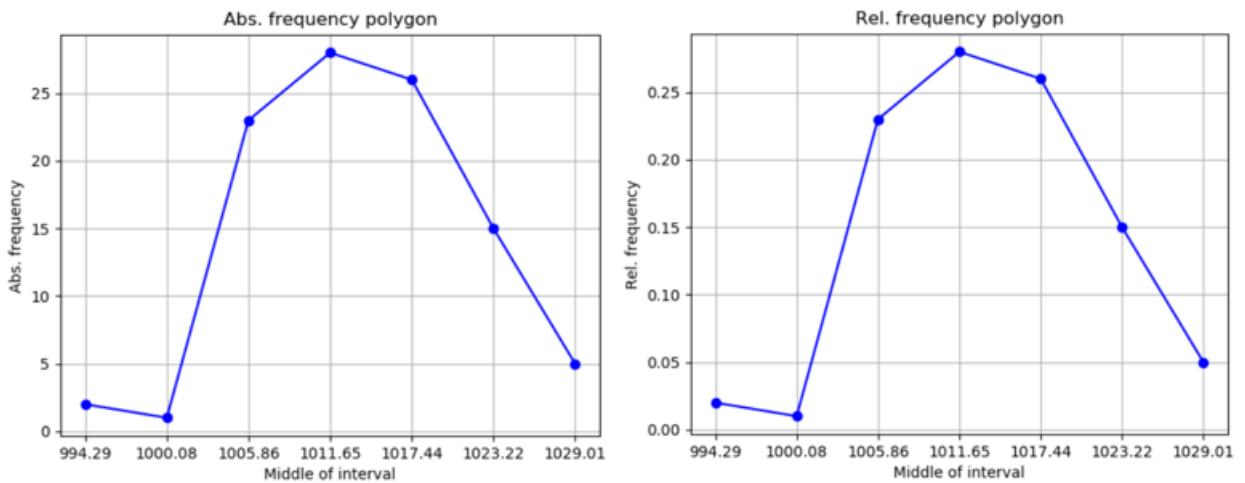


Рисунок 2.3 – Полигоны абсолютных и относительных частот  
интервального ряда

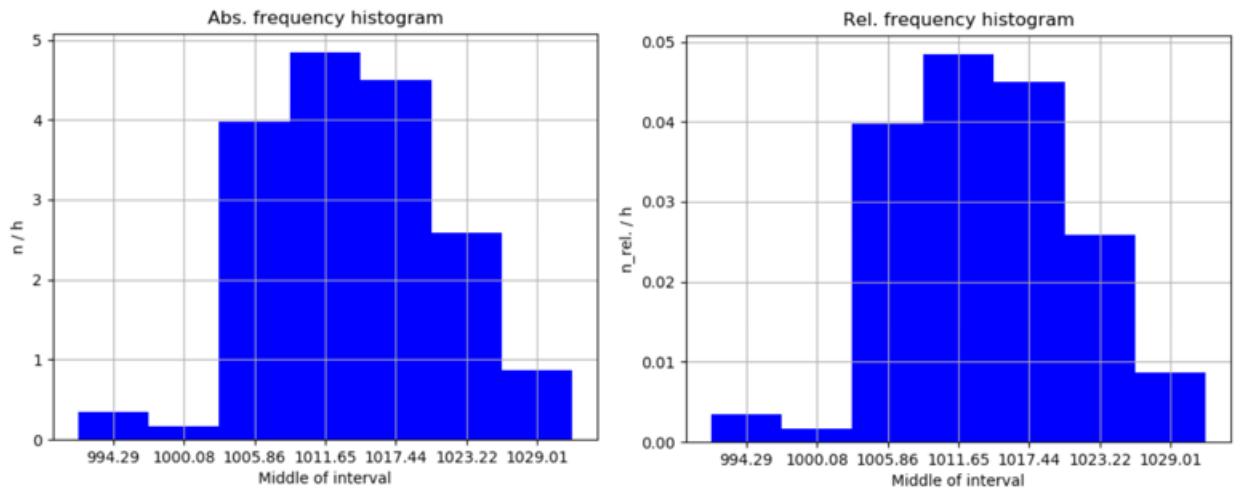


Рисунок 2.4 – Гистограммы абсолютных и относительных частот  
интервального ряда

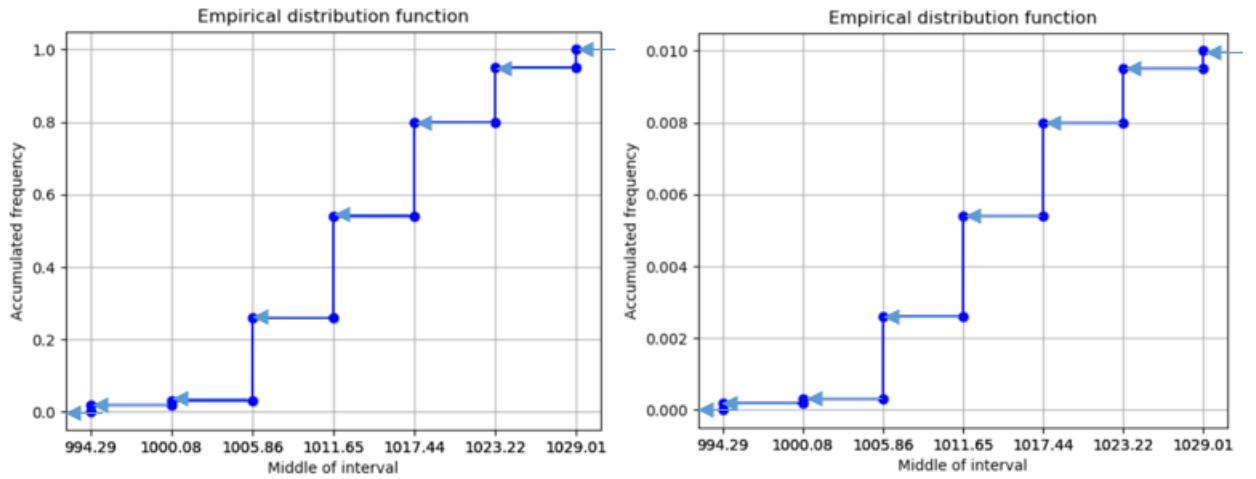


Рисунок 2.5 – Графики эмпирических функций распределения  
абсолютных и относительных частот

Таблица 2.2 – Интервальный ряд с условными вариантами

Номер инт-ла	Границы интервала	Середина интервала	Частота попадания в инт-л	Относит. частота	Условная варианта
1	[991.4; 997.186)	994.293	2	0.02	-3
2	[997.186; 1002.971)	1000.079	1	0.01	-2
3	[1002.971; 1008.757)	1005.864	23	0.23	-1
4	[1008.757; 1014.543)	1011.65	28	0.28	0
5	[1014.543; 1020.329)	1017.436	26	0.26	1
6	[1020.329; 1026.114)	1023.221	15	0.15	2
7	[1026.114; 1031.9]	1029.007	5	0.05	3

Далее были вычислены условные эмпирические моменты  $\bar{M}_k^*$ , а с помощью них – центральные эмпирические моменты  $\bar{m}_k$  ( $N$  – объем выборки,  $n_j$  – абсолютная частота на  $j$ -м интервале). Результаты вычислений представлены в таблице 2.3.

Таблица 2.3 – Условные эмпирические и центральные эмпирические моменты

$k$	1	2	3	4
$\bar{M}_k^*$	0.4	1.76	1.96	8.72
$\bar{m}_k$	1013.964	53.559	-4.648	8064.313

Также были найдены выборочное среднее  $\bar{y}_B$  и дисперсия  $D_B$  с помощью условныхvariant:

$$\bar{y}_B = \bar{M}_1^* h + C \approx 1013.964286; D_B = \bar{m}_2 \approx 53.559184$$

Эти величины также были вычислены при помощи стандартных формул:

$$\bar{y}_B = \frac{1}{N} \sum n_j y_j \approx 1013.964286; D_B = \frac{1}{N} \sum n_j (y_j - \bar{y}_B)^2 \approx 53.559184$$

Полученные разными способами значения практически совпадают. Незначительные различия в результатах вычислений могут быть обусловлены

слишком высокой точностью языка Python, из-за чего переменные в этом языке могут неточно интерпретироваться пользовательскими компьютерами.

С помощью полученных величин была найдена исправленная оценка дисперсии, а также статистические оценки СКО (среднеквадратичных отклонений):

$$s^2 = \frac{N}{N-1} D_{\text{в}} \approx 54.100186; \sigma_{\text{в}} = \sqrt{D_{\text{в}}} \approx 7.318414; s_y = \sqrt{s^2} \approx 7.355283$$

Далее были найдены статистические оценки коэффициентов асимметрии и эксцесса:

$$\bar{A}_s = \frac{\bar{m}_3}{s^3} \approx -0.011681; E = \frac{\bar{m}_4}{s^4} - 3 \approx -0.244694$$

Модой данного распределения будет являться 4-я варианта (середина 4-го интервала), так как именно у этого интервала наибольшая абсолютная частота вхождения выборочных данных ( $n_4 = 28$ ). Медиана рассматриваемого распределения – также середина 4-го (центрального) интервала. Таким образом,  $M_0 = m_e = 1011.65$ .

В связи с тем, что  $\bar{A}_s < 0$  и  $|\bar{A}_s| < 0.25$ , распределение можно охарактеризовать как «скошенное влево» и незначительно асимметричное. Так как  $E < 0$ , то говорят, что эмпирическое распределение низкое и пологое относительно «эталонного» нормального распределения.

В результате совмещения полученного интервального ряда для второго параметра и ряда из предыдущего раздела (для первого параметра) была построена таблица 2.4. По уже двумерному интервальному ряду была построена корреляционная таблица 2.5. В ячейках располагаются частоты вхождения в выборку элементов, чьи первые два параметра соответственно принадлежат интервалам, расположенным в строках и столбцах таблицы. Для проверки построенной корреляционной таблицы необходимо просуммировать все абсолютные частоты по первому ( $x$ ) и второму ( $y$ ) параметру. В обоих этих случаях сумма соответствующих частот должна равняться объему выборки. Так как это условие выполняется (что видно из таблицы), то она построена верно.

Таблица 2.4 – Двумерный интервальный ряд

№ инт-ла $i$	Интервальный ряд 1-го параметра				Интервальный ряд 2-го параметра			
	Границы	$x_i$	$n_i$	$\tilde{n}_i$	Границы	$y_i$	$n_i$	$\tilde{n}_i$
1	[0.329; 5.023)	2.676	8	0.08	[991.4; 997.186)	994.293	2	0.02
2	[5.023; 9.718)	7.37	9	0.09	[997.186; 1002.971)	1000.079	1	0.01
3	[9.718; 14.412)	12.065	17	0.17	[1002.971; 1008.757)	1005.864	23	0.23
4	[14.412; 19.107)	16.759	16	0.16	[1008.757; 1014.543)	1011.65	28	0.28
5	[19.107; 23.801)	21.454	25	0.25	[1014.543; 1020.329)	1017.436	26	0.26
6	[23.801; 28.496)	26.148	15	0.15	[1020.329; 1026.114)	1023.221	15	0.15
7	[28.496; 33.19]	30.843	10	0.1	[1026.114; 1031.9]	1029.007	5	0.05

Таблица 2.5 – Корреляционная таблица двумерного интервала

Интервальный ряд 2-го пар-па ( $y$ ; середины инт-лов)		Интервальный ряд 1-го пар-па ( $x$ ; середины инт-лов)							Суммарная частота $n_y$
		$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	
$y_1$	994.293	0	0	0	2	0	0	0	2
$y_2$	1000.079	0	0	0	0	1	0	0	1
$y_3$	1005.864	0	2	3	3	6	5	4	23
$y_4$	1011.65	0	2	2	3	9	6	6	28
$y_5$	1017.436	2	1	6	5	8	4	0	26
$y_6$	1023.221	4	3	4	3	1	0	0	15
$y_7$	1029.007	2	1	2	0	0	0	0	5
Суммарная частота $n_x$		8	9	17	16	25	15	10	$N = 100$

По данным таблицы была вычислена статистическая оценка коэффициента корреляции:  $\bar{r}_{xy} = \frac{\sum_{i=1}^{K_y} \sum_{j=1}^{K_x} n_{ij} y_i x_j - N \bar{x}_B \bar{y}_B}{N S_x S_y} \approx -0.4725896$ . Здесь  $n_{ij}, y_i, x_j$  – данные из корреляционной таблицы;  $N = 100$ ;  $\bar{x}_B = 17.9798848$  – выборочное среднее из 2-й работы;  $\bar{y}_B = 1013.964286$ ;  $S_x = 8.07944449$ ;  $S_y = 7.355283$ . Таким образом, оценка коэффициента корреляции  $r_{xy}$ :  $\bar{r}_{xy} - 3 \frac{1 - \bar{r}_{xy}^2}{\sqrt{N}} \leq r_{xy} \leq \bar{r}_{xy} + 3 \frac{1 + \bar{r}_{xy}^2}{\sqrt{N}}$ ;  $-0.705587 \leq r_{xy} \leq -0.105587$ .

Исходя из полученной оценки коэффициента был сделан вывод о том, что между переменными может иметь место умеренная или незначительная отрицательная корреляция. То есть поведение выходной переменной будет противоположным поведению входной (если значение  $x$  возрастает,  $y$  – убывает, и наоборот) в незначительной степени.

Далее был найден доверительный интервал для коэффициента корреляции. Для этого с помощью преобразования Фишера был выполнен переход к случайной величине  $z$ :  $\bar{z} = 0.5 \ln \frac{1 + \bar{r}_{xy}}{1 - \bar{r}_{xy}} = 1.1513 \lg \frac{1 + \bar{r}_{xy}}{1 - \bar{r}_{xy}} \approx -0.513403$ . Значение СКВО распределения  $z$ :  $\bar{\sigma}_z = \frac{1}{\sqrt{N-3}} \approx 0.101535$ . При известном значении функции Лапласа  $\Phi[\lambda(\gamma)] = \frac{\gamma}{2} = 0.475$  (отсюда  $\lambda(\gamma) = 1.96$ ) был найден доверительный интервал для  $r_{xy}$  генеральной совокупности с доверительной вероятностью  $\gamma$ :

$$(\bar{z} - \lambda(\gamma) \bar{\sigma}_z, \bar{z} + \lambda(\gamma) \bar{\sigma}_z)$$

$$(-0.712411, -0.314395)$$

Пересчет границ интервала в доверительный интервал для коэффициента корреляции с тем же значением  $\gamma$  было использовано обратное преобразование Фишера:  $r = \text{th}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = \frac{e^{2z} - 1}{e^{2z} + 1}$ . Таким образом,  $(-0.612186, -0.30443)$ .

Полученный доверительный интервал с вероятностью  $\gamma = 0.95$  покрывает истинное значение коэффициента корреляции  $r_{xy}$  и уточняет полученные ранее оценочные данные.

Также была осуществлена проверка статистической гипотезы о равенстве нулю значения  $r_{xy}$  при заданном уровне значимости  $\alpha = 0.05$ . В качестве критерия была принята случайная величина:  $T_{\text{набл}} = \frac{\bar{r}_{xy}\sqrt{N-2}}{\sqrt{1-\bar{r}_{xy}^2}} \approx -5.308624$ . При  $\alpha = 0.05$  и количестве степеней свободы  $k = N - 2 = 98$ :  $t_{\text{крит}}(\alpha, k) = 1.984$ ;  $|T_{\text{набл}}| > t_{\text{крит}}(\alpha, k)$ .

Следовательно, на уровне значимости  $\alpha = 0.05$  предполагаемая гипотеза о равенстве нулю значения  $r_{xy}$  отвергается, а это значит, что коэффициент корреляции значим (его значение значимо отличается от нуля).

### 2.3. Выборочные прямые среднеквадратической регрессии. Корреляционные отношения

Сначала было построено графическое представление рассматриваемой двумерной выборки с помощью имеющегося двумерного интервального ряда (см. табл. 2.4). Построенный график приведен на рис. 2.6.

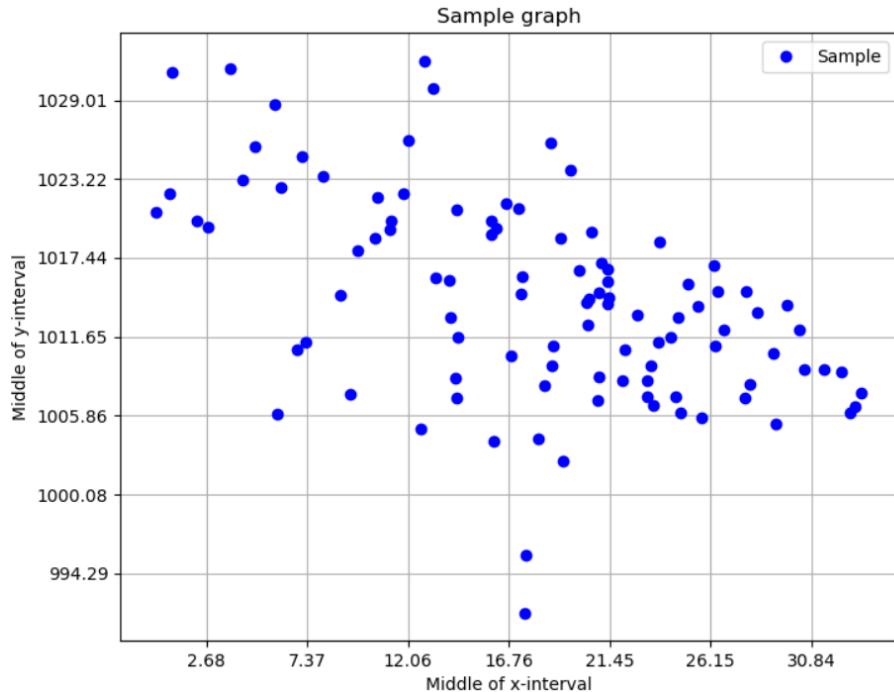


Рисунок 2.6 – Графическое представление двумерной выборки

Для данной выборки были построены уравнения средней квадратичной регрессии  $x$  на  $y$  и  $y$  на  $x$ :  $\bar{y}_x = \bar{y}_B + \bar{r}_{xy} \frac{S_y}{S_x} (x - \bar{x}_B)$ ;  $\bar{x}_y = \bar{x}_B + \bar{r}_{xy} \frac{S_x}{S_y} (y - \bar{y}_B)$ .

Здесь  $\bar{y}_B = 1013.964286$ ,  $\bar{x}_B = 17.979885$  – выборочные средние, рассчитанные ранее;  $S_y = 7.355283$ ,  $S_x = 8.079444$  – исправленные выборочные СКВО;  $\bar{r}_{xy} = -0.47259$  – статистическая оценка коэффициента корреляции. Прямые, описываемые этими уравнениями, были добавлены к графическому представлению выборки. Результат приведен на рис. 2.7.

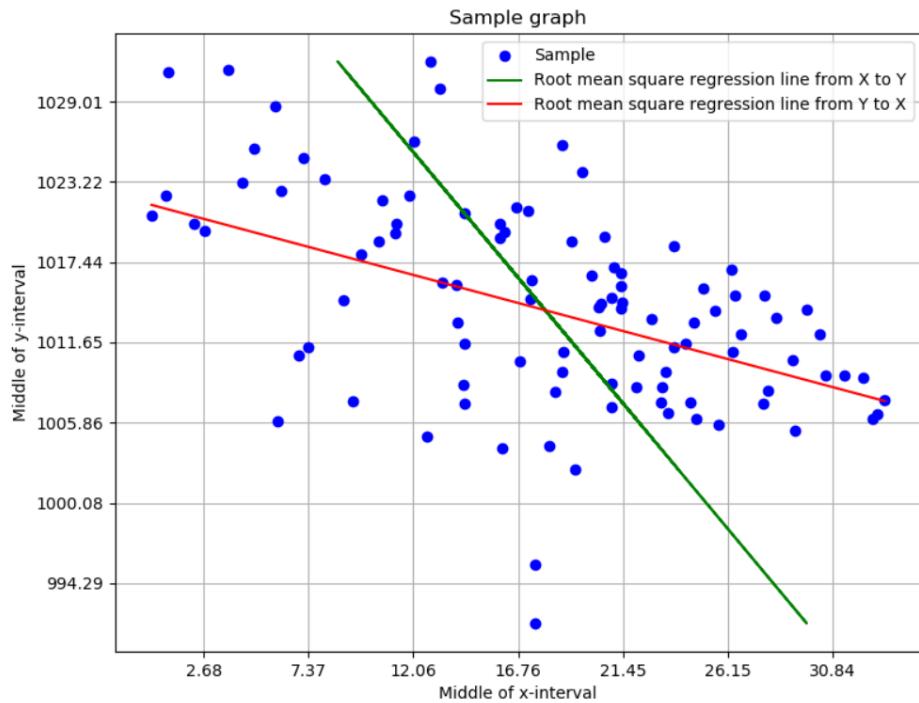


Рисунок 2.7 – Графическое представление двумерной выборки с учетом выборочных прямых среднеквадратической регрессии

Так как  $0.3 \leq |\bar{r}_{xy}| \leq 0.5$  и  $\bar{r}_{xy} < 0$ , то связь  $X$  и  $Y$  умеренная обратная, что подтверждает полученный график: с увеличением значений абсцисс происходит незначительное уменьшение значений ординат и наоборот.

Далее была составлена корреляционная таблица для нахождения выборочного корреляционного отношения (табл. 2.6) на основе составленной ранее корреляционной таблицы (табл. 2.5). Здесь значения  $\bar{x}_{y_i}$  и  $\bar{y}_{x_i}$  – усредненные по соответствующей суммарной частоте ( $n_{y_i}$  или  $n_{x_i}$ ) суммы произведений частот из ячеек таблицы и соответствующих им середин интервалов ( $x_i$  или  $y_i$ ). Вычисления групповых дисперсий производились по стандартной формуле:  $D_{xy} = \frac{1}{n_{y_i}} \sum n_{ij} (x_j - \bar{x}_{y_i})^2$  или  $D_{yx} = \frac{1}{n_{x_i}} \sum n_{ij} (y_j - \bar{y}_{x_i})^2$ .

Таблица 2.6 – Корреляционная таблица для нахождения выборочного корреляционного отношения

Интервальный ряд 2-го пар-ра ( $y$ ; середины инт-лов)		Интервальный ряд 1-го пар-ра ( $x$ ; середины инт-лов)									
		$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$n_{y_i}$	$\bar{x}_{y_i}$	Значение $D_{xy}$
$y_1$	994.293	0	0	0	2	0	0	0			
$y_2$	1000.079	0	0	0	0	1	0	0			
$y_3$	1005.864	0	2	3	3	6	5	4	23	21.046	51.575
$y_4$	1011.65	0	2	2	3	9	6	6	28	22.292	45.735
$y_5$	1017.436	2	1	6	5	8	4	0	26	17.12	43.946
$y_6$	1023.221	4	3	4	3	1	0	0	15	10.187	34.673
$y_7$	1029.007	2	1	2	0	0	0	0	5	7.37	17.631
Значение $n_{x_i}$		8	9	17	16	25	15	10	$N = 100$		
Значение $\bar{y}_{x_i}$		1023.2	1016.7	1017.4	1012.3	1012.1	1011.2	1009.3			
Значение $D_{xy}$		62.816	51.196	78.979	29.243	19.936	8.034				

Полученные групповые дисперсии были применены для вычисления внутригрупповых, взвешенных по объемам групп средних арифметических групповых дисперсий:  $D_{\text{вн гр } x_y} \approx 42.176461$ ;  $D_{\text{вн гр } y_x} \approx 39.43704$ . Также были найдены межгрупповые дисперсии – дисперсии условных (групповых) средних  $\bar{x}_{y_i}$  и  $\bar{y}_{x_i}$  относительно выборочных средних  $\bar{x}_B$  и  $\bar{y}_B$ :  $D_{\text{меж гр } x_y} \approx 22.448188$ ;  $D_{\text{меж гр } y_x} \approx 14.122143$ . Общие дисперсии являются суммами

соответствующих      внутригрупповых      и      межгрупповых:       $D_{\text{общ } x_y} \approx 64.624649$ ;  $D_{\text{общ } y_x} \approx 53.559184$ .

На основе полученных данных были найдены значения  $\bar{\eta}_{xy}$  – выборочных корреляционных отношений:

$$\bar{\eta}_{xy} = \frac{\bar{\sigma}_{\bar{x}_y}}{\bar{\sigma}_x} = \frac{\sqrt{D_{\text{меж гр } xy}}}{\sqrt{D_{\text{общ } x_y}}} \approx 0.589375; \bar{\eta}_{yx} = \frac{\bar{\sigma}_{\bar{y}_x}}{\bar{\sigma}_y} = \frac{\sqrt{D_{\text{меж гр } y_x}}}{\sqrt{D_{\text{общ } y_x}}} \approx 0.513492$$

Таким образом, при  $|r_{xy}| = 0.47259$  неравенства  $\eta_{xy} \geq |r_{xy}|$  и  $\eta_{yx} \geq |r_{xy}|$  выполняются.

Для заданной выборки была построена корреляционная кривая параболического вида  $y = \beta_2 x^2 + \beta_1 x + \beta_0$ . Коэффициенты найдены из системы, полученной методом наименьших квадратов:

$$\begin{cases} \left( \sum_{i=1}^m n_{x_i} x_i^4 \right) \beta_2 + \left( \sum_{i=1}^m n_{x_i} x_i^3 \right) \beta_1 + \left( \sum_{i=1}^m n_{x_i} x_i^2 \right) \beta_0 = \sum_{i=1}^m n_{x_i} \bar{y}_{x_i} x_i^2 \\ \left( \sum_{i=1}^m n_{x_i} x_i^3 \right) \beta_2 + \left( \sum_{i=1}^m n_{x_i} x_i^2 \right) \beta_1 + \left( \sum_{i=1}^m n_{x_i} x_i \right) \beta_0 = \sum_{i=1}^m n_{x_i} \bar{y}_{x_i} x_i \\ \left( \sum_{i=1}^m n_{x_i} x_i^2 \right) \beta_2 + \left( \sum_{i=1}^m n_{x_i} x_i \right) \beta_1 + N \beta_0 = \sum_{i=1}^m n_{x_i} \bar{y}_{x_i} \end{cases}$$

Рассчитанные коэффициенты:  $\beta_2 \approx 0.012392; \beta_1 \approx -0.856304; \beta_0 \approx 1024.553489$ . Также была простроена корреляционная кривая дробно-линейной функции  $-y = \frac{1}{\beta_1 x + \beta_0}$ . Для этого была произведена замена переменных:  $\beta_1 x + \beta_0 = \frac{1}{y}; X = x; Y = \frac{1}{y}$ . Таким образом,  $\beta_1 X + \beta_0 = Y; f(\beta_1, \beta_0) = \sum_{i=1}^m n_{x_i} (\beta_1 X + \beta_0 - Y)^2$ .

$$\begin{cases} \frac{\partial f(\beta_1, \beta_0)}{\partial \beta_1} = 2 \sum_{i=1}^m n_{x_i} (\beta_1 X + \beta_0 - Y) \cdot X = 0 \\ \frac{\partial f(\beta_1, \beta_0)}{\partial \beta_0} = 2 \sum_{i=1}^m n_{x_i} (\beta_1 X + \beta_0 - Y) = 0 \end{cases}$$

$$\left\{ \begin{array}{l} \left( \sum_{i=1}^m n_{x_i} X_i^2 \right) \beta_1 + \left( \sum_{i=1}^m n_{x_i} X_i \right) \beta_0 = \sum_{i=1}^m n_{x_i} \bar{Y}_{x_i} X_i \\ \left( \sum_{i=1}^m n_{x_i} X_i \right) \beta_1 + N \beta_0 = \sum_{i=1}^m n_{x_i} \bar{Y}_{x_i} \end{array} \right.$$

$$\left\{ \begin{array}{l} \left( \sum_{i=1}^m n_{x_i} x_i^2 \right) \beta_1 + \left( \sum_{i=1}^m n_{x_i} x_i \right) \beta_0 = \sum_{i=1}^m \frac{n_{x_i} x_i}{\bar{y}_{x_i}} \\ \left( \sum_{i=1}^m n_{x_i} x_i \right) \beta_1 + N \beta_0 = \sum_{i=1}^m \frac{n_{x_i}}{\bar{y}_{x_i}} \end{array} \right.$$

Рассчитанные коэффициенты:  $\beta_1 \approx 0; \beta_0 \approx 0.000979$ . Построенные параболическая кривая и кривая дробно-линейной функции приведены на рис. 2.8.

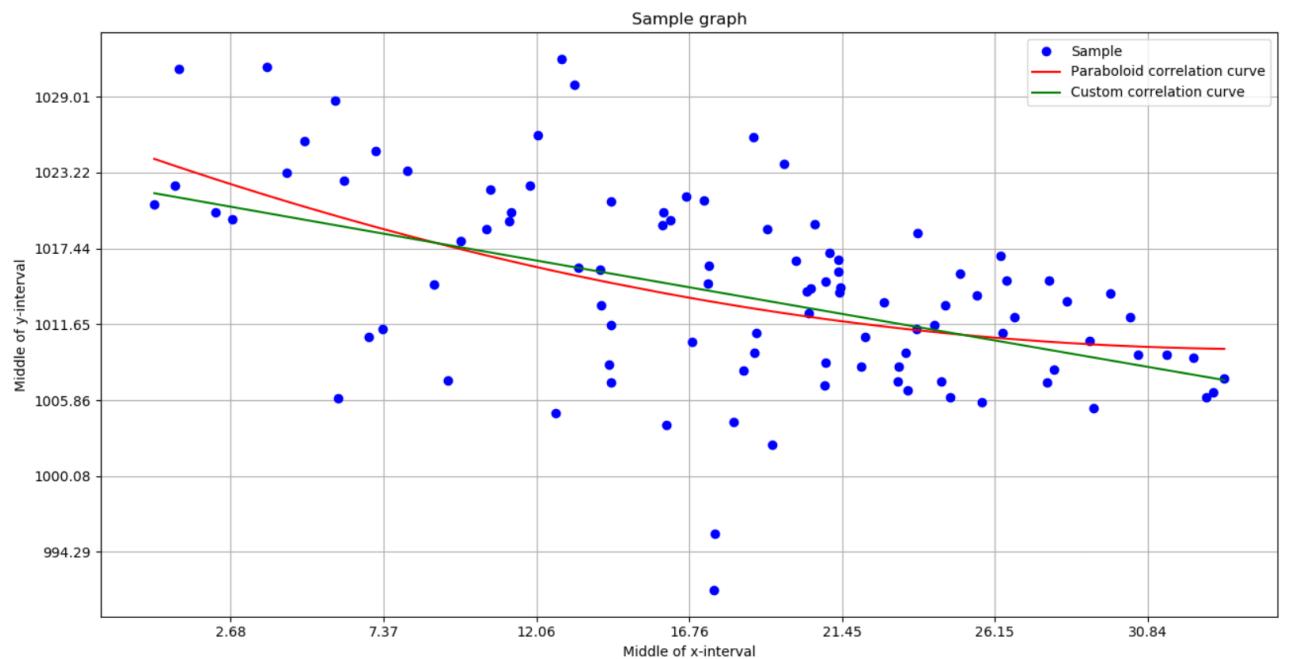


Рисунок 3 – Корреляционные кривые параболического вида и дробно-линейной функции

#### 2.4. Выводы

В рамках данного раздела была сформирована двумерная выборка, для которой был произведен корреляционный и регрессионный анализ.

Найдена оценка корреляционных отношений; построены выборочные прямые и корреляционные кривые среднеквадратической регрессии. Для

рассматриваемой двумерной выборки была создана корреляционная таблица, по ней вычислена оценка коэффициента корреляции и доверительный интервал, который с вероятностью  $\gamma = 0.95$  покрывает истинное значение коэффициента корреляции  $r_{xy}$  и уточняет полученные оценочные данные.

Была проведена проверка гипотезы о значимости коэффициента корреляции. На уровне значимости  $\alpha = 0.05$  предполагаемая гипотеза о равенстве нулю значения  $r_{xy}$  отвергается, а это значит, что коэффициент корреляции значим (его значение значимо отличается от нуля).

### 3. КЛАСТЕРНЫЙ АНАЛИЗ

В данном разделе был произведен кластерный анализ рассматриваемой двумерной выборки. Были рассмотрены два алгоритма:  $k$ -средних и поиска сгущений. На каждом этапе алгоритмов рассчитывались оценки качества разбиения. Алгоритм  $k$ -средних был исследован в двух своих вариациях: пересчет центроид производился после каждого изменения состава кластера и после завершения каждого шага процедуры. Расчеты производились при помощи программы (исходный код Python представлен в приложении В).

#### 3.1. Основные теоретические положения

*Метод  $k$ -средних.*

Пусть имеется  $n$  наблюдений, каждое из которых характеризуется  $m$  признаками  $X_1, X_2, \dots, X_n$ . Эти наблюдения необходимо разбить на  $k$  кластеров. Из  $n$  точек исследуемой совокупности отбираются случайным образом или задаются, исходя из каких-либо априорных соображений,  $k$  точек (объектов). Эти точки принимаются за эталоны – центры кластеров. Каждому эталону присваиваются порядковый номер, который одновременно является и номером кластера.

На первом шаге из оставшихся  $(n - k)$  объектов извлекается точка  $X_i$  с координатами  $(x_{i1}, x_{i2}, \dots, x_{in})$  и проверяется, к каждому из эталонов (центров) она находится ближе всего. Для этого используется одна из метрик, например, евклидово расстояние  $d_{Eij} = \left( \sum_{k=1}^m (x_k^{(i)} - x_k^{(j)})^2 \right)^{\frac{1}{2}}$ . Проверяемый объект присоединяется к тому центру (эталону), которому соответствует минимальное из расстояний. Эталон заменяется новым (корректируется), пересчитанным с учетом присоединенной точки (вычисляется новое значение среднего арифметического всех включенных в кластер элементов), вес кластера (количество объектов, входящих в кластер) увеличивается на единицу. Если встречаются два и более минимальных расстояния, то  $i$ -й объект присоединяют к центру (кластеру) с наименьшим порядковым номером.

*На следующем шаге* выбирают точку  $X_{i+1}$  и для нее повторяются все процедуры. Таким образом, через  $(n - k)$  шагов все точки (объекты) совокупности окажутся отнесенными к одному из  $k$  кластеров. Цикл процедуры завершается, но на этом метод работу не заканчивает.

*Для того, чтобы добиться устойчивости* все кластеры считаются пустыми с центрами (эталонами), полученными в конце предыдущего цикла. Все точки  $X_1, X_2, \dots, X_n$  снова последовательно подсоединяются к этим кластерам по рассмотренным правилам. Цикл повторяется. По его завершению новое разбиение сравнивается с полученным в предыдущем цикле. Если они совпадают, работа алгоритма завершается. В противном случае цикл снова повторяется.

*Окончательное разбиение* имеет центры тяжести, которые, как правило, не совпадают с первоначальными эталонами. Каждая точка  $X_i (i = 1, 2, \dots, n)$  будет относится к тому кластеру, расстояние до центра которого от этой точки минимально.

*Возможны две разновидности* метода  $k$ -средних. *Первая* предполагает пересчет центра кластера после каждого изменения его состава, как рассмотрено выше, а *вторая* – лишь после завершения цикла. В обоих случаях итеративный алгоритм этого метода минимизирует дисперсию внутри каждого кластера, хотя в явном виде такой критерий оптимизации не используется.

Перед началом работы метода целесообразно нормировать характеристики объектов:  $\hat{X} = \frac{x - \bar{x}_B}{s_x}$ ;  $\hat{Y} = \frac{y - \bar{y}_B}{s_y}$ . Если изначально нет разумных соображений по необходимому количеству кластеров для разбиения, рекомендуется первоначально создать 2 кластера, затем 3, 4, 5 и т. д., сравнивая результаты.

### *Оценка качества многомерной классификации.*

Для оценки полученных результатов кластеризации используются:

1. Сумма квадратов расстояний до центров кластеров  $F_1 = \sum_{k=1}^K \sum_{i=1}^{N_k} d^2(X_i^{(k)}, \bar{X}^{(k)}) \Rightarrow \min;$

2. Сумма внутрикластерных расстояний между объектами  $F_2 = \sum_{k=1}^K \sum_{X_i, X_j \in S_k} d^2(X_i, X_j) \Rightarrow \min$ ;

3. Сумма внутрикластерных дисперсий  $F_3 = \sum_{k=1}^K \sum_{j=1}^{N_k} \sigma_{ij}^2 \Rightarrow \min$ .

Здесь  $\sigma_{ij}^2$  – дисперсия  $j$ -й переменной в  $k$ -м кластере. Оптимальным следует считать разбиение, при котором сумма внутрикластерных (внутригрупповых) дисперсий будет минимальной.

### *Метод поиска сгущений.*

*Основная идея метода* заключается в построении гиперсферы заданного радиуса, которая перемещается в пространстве классификационных признаков в поисках локальных сгущений объектов. *Метод поиска сгущений требует*, прежде всего, *вычисления матрицы расстояний* (или матрицы мер сходства) между объектами и *выбора первоначального центра сферы*.

*Как правило, на первом шаге* центром сферы служит объект (точка), в ближайшей (заданной) окрестности которого расположено наибольшее число соседей. На основе заданного радиуса сферы ( $R$ ) определяется совокупность точек внутри этой сферы, и для них вычисляются координаты центра (вектор средних для попавших в сферу значений признаков).

*Когда очередной пересчет* координат центра сферы приводит к такому же результату, как и на предыдущем шаге, перемещение сферы прекращается, а точки, попавшие в нее, образуют кластер, и из дальнейшего процесса кластеризации исключаются.

*Перечисленные процедуры повторяются* для всех оставшихся точек. Работа алгоритма завершается за конечное число шагов, и все точки оказываются распределенными по кластерам. Число образовавшихся кластеров заранее неизвестно и сильно зависит от заданного радиуса сферы.

*Для оценки устойчивости* полученного разбиения целесообразно повторить процесс кластеризации несколько раз для различных значений радиуса сферы, изменяя каждый раз радиус на небольшую величину.

*Существуют различные способы выбора начального радиуса сферы.* В частности, если обозначить через  $d_{ij}$  расстояние между  $i$ -м и  $j$ -м объектами, то в качестве нижней границы значения радиуса сферы можно выбрать минимальное из таких расстояний, а в качестве верхней границы – максимальное:  $R_{min} = \min_{i,j} d_{ij}$ ;  $R_{max} = \max_{i,j} d_{ij}$ . Тогда, если начинать алгоритма работу с  $R = R_{min} + \delta$ ;  $\delta > 0$  и при каждом его повторении увеличивать значение  $\delta$  на некоторую величину, то в конечном итоге можно найти значения радиусов, которые приводят к устойчивому разбиению на кластеры. Следует отметить следующие существенные при реализации метода поиска сгущений моменты:

1. В случае разномасштабности квалификационных признаков необходимо проведение их нормировки перед началом работы метода;
2. Возможны два варианта реализации метода. Одним из них не предполагает изменения заданного значения радиуса сферы до завершения кластеризации, а другой – предполагает изменение этого радиуса в процессе кластеризации при начале построения очередной сферы;
3. В отличие от метода  $k$ -средних метод поиска сгущений не требует задания количества кластеров, на которые предполагается разбить исходное множество объектов;
4. Качество полученного в результате применения метода итогового разбиения на кластеры оцениваются, как и метода  $k$ -средних, с помощью введенных на предыдущей лекции критериев качества разбиения  $F_1$ ,  $F_2$  и  $F_3$ ;

Получение в результате кластеризации пересекающихся кластеров (наличие спорных объектов) в принципе является неудовлетворительным результатом. На практике в этом случае необходимо скорректировать процесс, либо выбрать другой метод кластеризации.

### 3.2. Метод $k$ -средних

Перед началом работы алгоритма рассматриваемое множество точек было нормализовано по обоим из интересующих параметров. Исходная и нормализованная выборки представлены на рис. 3.1.

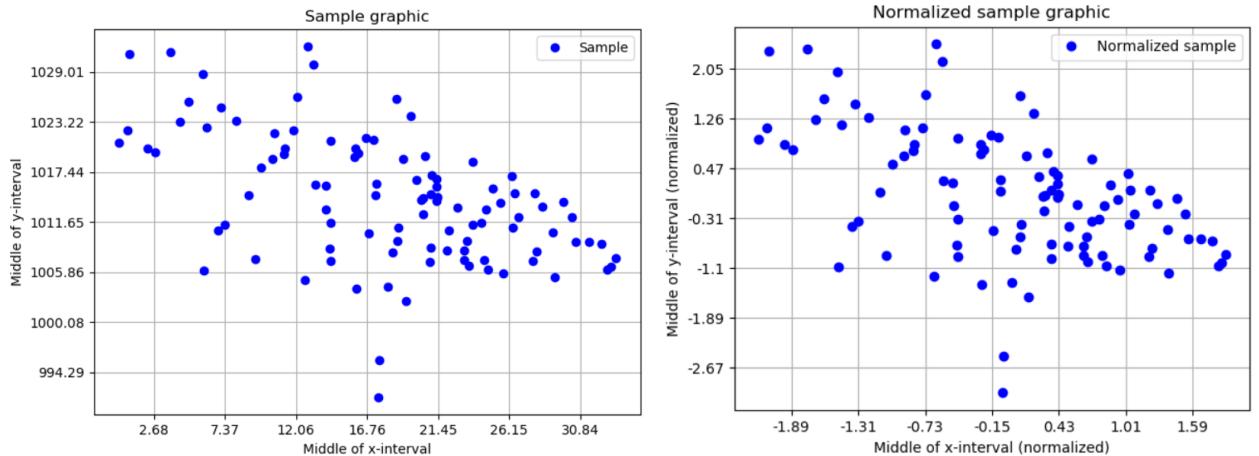


Рисунок 3.1 – Исходная двумерная выборка и нормализованная  
соответственно

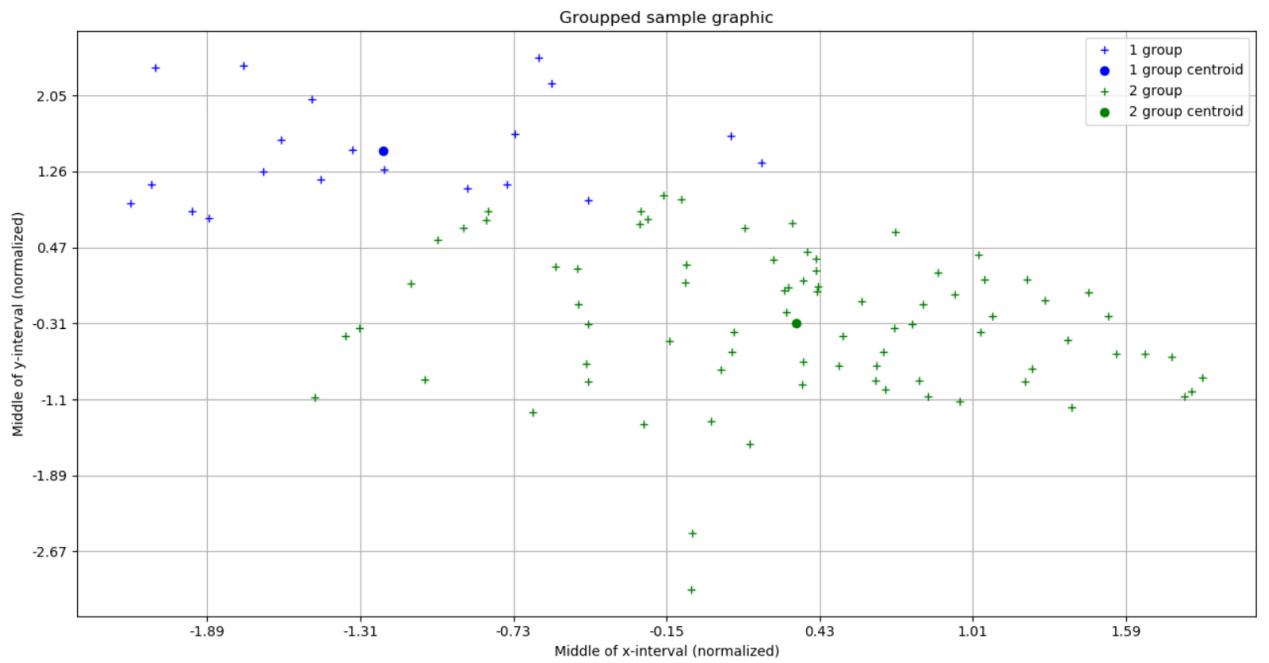
При объеме выборки  $N = 100$  была найдена верхняя оценка количества кластеров по формуле:  $\bar{k} = \lfloor \sqrt{N/2} \rfloor \approx 7$ . Так как по графику распределения точек нельзя сказать, на сколько кластеров будет целесообразнее разделить данные, то в работе были последовательно рассмотрены значения количества кластеров от  $k = 2$  до  $k = \bar{k}$ .

Также, учитывая то, что алгоритм  $k$ -means предполагает два варианта обновления центроид каждого кластера (после добавления каждого нового элемента или по окончании каждого цикла работы алгоритма), в данной работе для каждого значения  $k$  были рассмотрены оба случая.

Для каждого варианта запуска (разбиения)  $k$ -means рассчитывались оценки качества: сумма квадратов расстояний до центров кластеров ( $F_1 = \sum_{k=1}^K \sum_{i=1}^{N_k} d^2(X_i^{(k)}, \bar{X}^{(k)}) \Rightarrow \min$ ); сумма внутрикластерных расстояний между объектами ( $F_2 = \sum_{k=1}^K \sum_{X_i, X_j \in S_k} d^2(X_i, X_j) \Rightarrow \min$ ) и сумма внутрикластерных дисперсий ( $F_3 = \sum_{k=1}^K \sum_{j=1}^{N_k} \sigma_{ij}^2 \Rightarrow \min$ ). В качестве расстояния между точками использовалось Евклидово.

Результаты разбиений на кластеры рассматриваемой выборки при различных режимах обновления центроид и различных количествах кластеров представлены на рис. 3.2 – 3.13 (также к рисункам прилагаются листинги

программы – вычисление оценок качества разбиения на каждом цикле работы алгоритма).



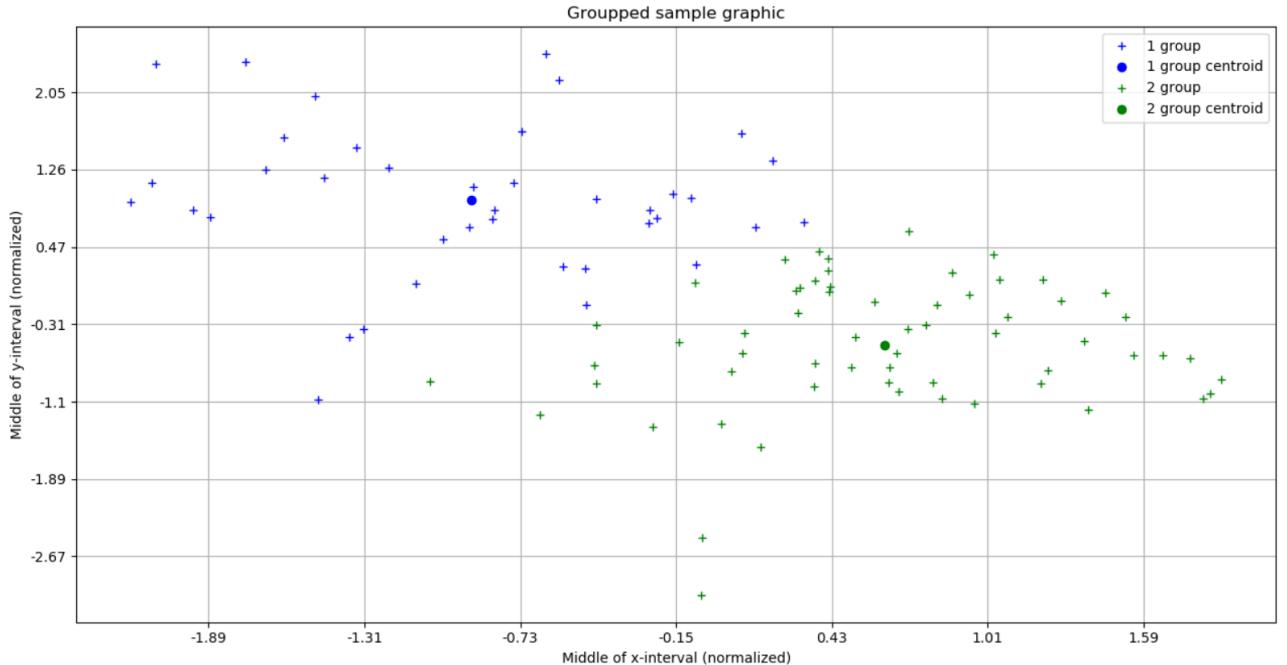
$$F1 = 112.28111418951862 \quad F2 = 16138.535016232947 \quad F3 = 7.073763306197124$$

$$F1 = 112.28111418951862 \quad F2 = 16138.535016232941 \quad F3 = 21.739595454064045$$

$$F1 = 112.28111418951862 \quad F2 = 16138.535016232941 \quad F3 = 21.739595454064045$$

Рисунок 3.2 – Полученные кластеры и оценки качества разбиения при  $k = 2$  и с пересчетом центров после добавления каждой новой точки в соответствующий

кластер

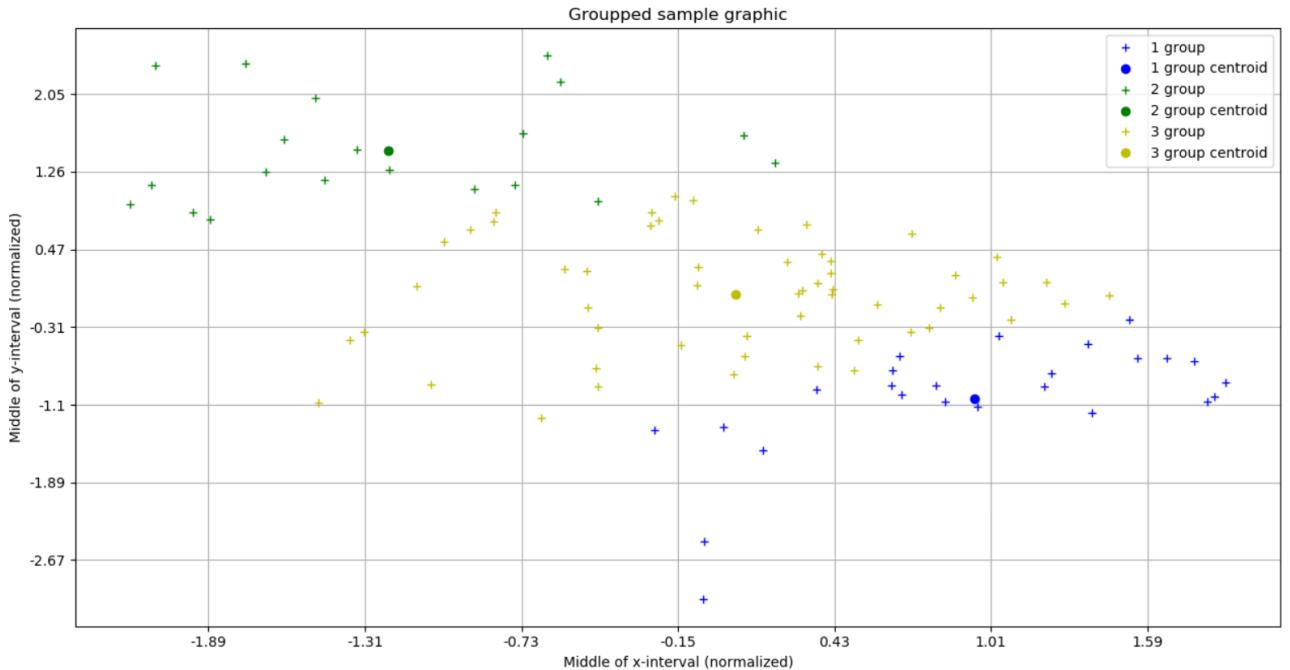


```

F1 = 109.9367274184755 F2 = 11875.485781401863 F3 = 12.440553275247607
F1 = 99.84038571771173 F2 = 9911.003428977845 F3 = 16.779303539264873
F1 = 95.81524527468498 F2 = 9544.098818021166 F3 = 13.893185700948386
F1 = 94.65164630854345 F2 = 9565.427015083491 F3 = 11.903512857311313
F1 = 94.37421092984074 F2 = 9687.910560834713 F3 = 16.340287298549722
F1 = 94.37421092984074 F2 = 9687.910560834713 F3 = 16.340287298549722

```

Рисунок 3.3 – Полученные кластеры и оценки качества разбиения при  $k = 2$  и с пересчетом центров только после завершения очередного цикла алгоритма

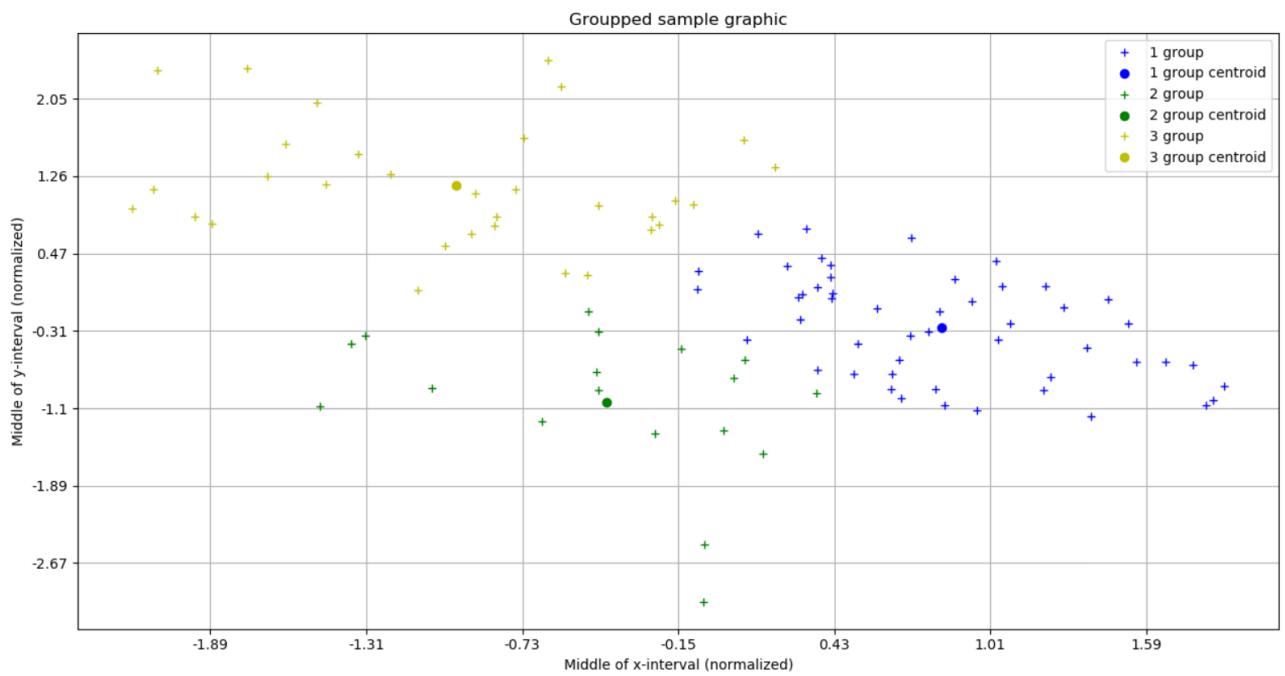


```

F1 = 74.40574468763717 F2 = 5618.766287594447 F3 = 6.999661445386962
F1 = 94.41865965167057 F2 = 10541.95394300813 F3 = 32.82017237370578
F1 = 79.50633257328448 F2 = 6518.249084956827 F3 = 27.3629783574601
F1 = 79.50633257328448 F2 = 6518.249084956827 F3 = 27.3629783574601

```

Рисунок 3.4 – Полученные кластеры и оценки качества разбиения при  $k = 3$  и с пересчетом центров после добавления каждой новой точки в соответствующий кластер



$$F_1 = 70.95813313889262 \quad F_2 = 4704.533635355706 \quad F_3 = 4.06535305090484$$

$$F_1 = 69.257761666282736 \quad F_2 = 4732.634877926893 \quad F_3 = 9.028659151025115$$

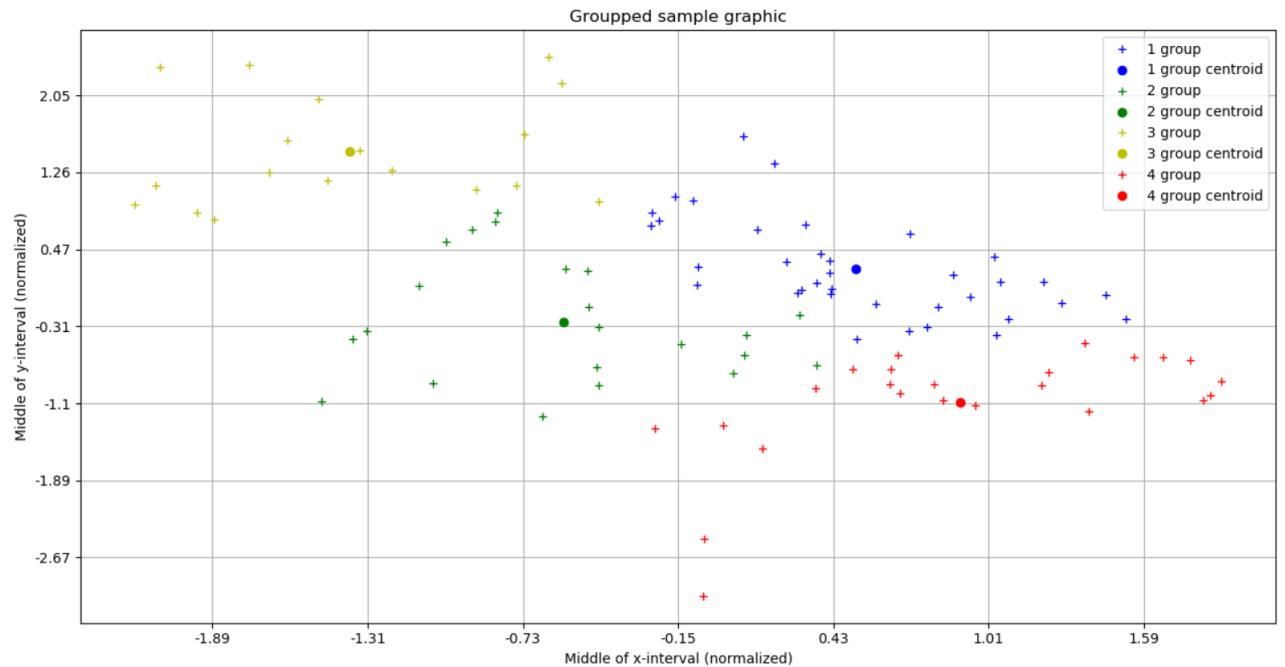
$$F_1 = 68.23550786663162 \quad F_2 = 4808.797366653004 \quad F_3 = 7.903555910894296$$

$$F_1 = 68.0370493108753 \quad F_2 = 4846.493514587161 \quad F_3 = 7.600743162838363$$

$$F_1 = 67.94942337184327 \quad F_2 = 4900.513974145244 \quad F_3 = 7.281470887151613$$

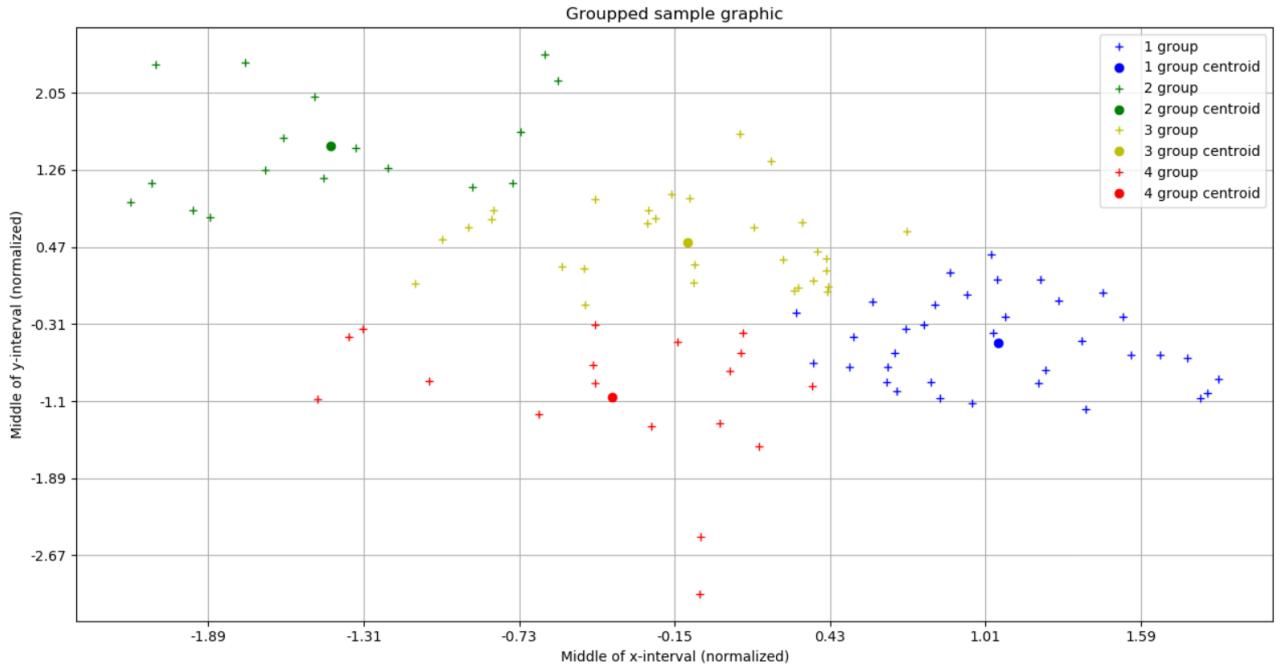
$$F_1 = 67.94942337184327 \quad F_2 = 4900.513974145244 \quad F_3 = 7.281470887151613$$

Рисунок 3.5 – Полученные кластеры и оценки качества разбиения при  $k = 3$  и с пересчетом центров только после завершения очередного цикла алгоритма



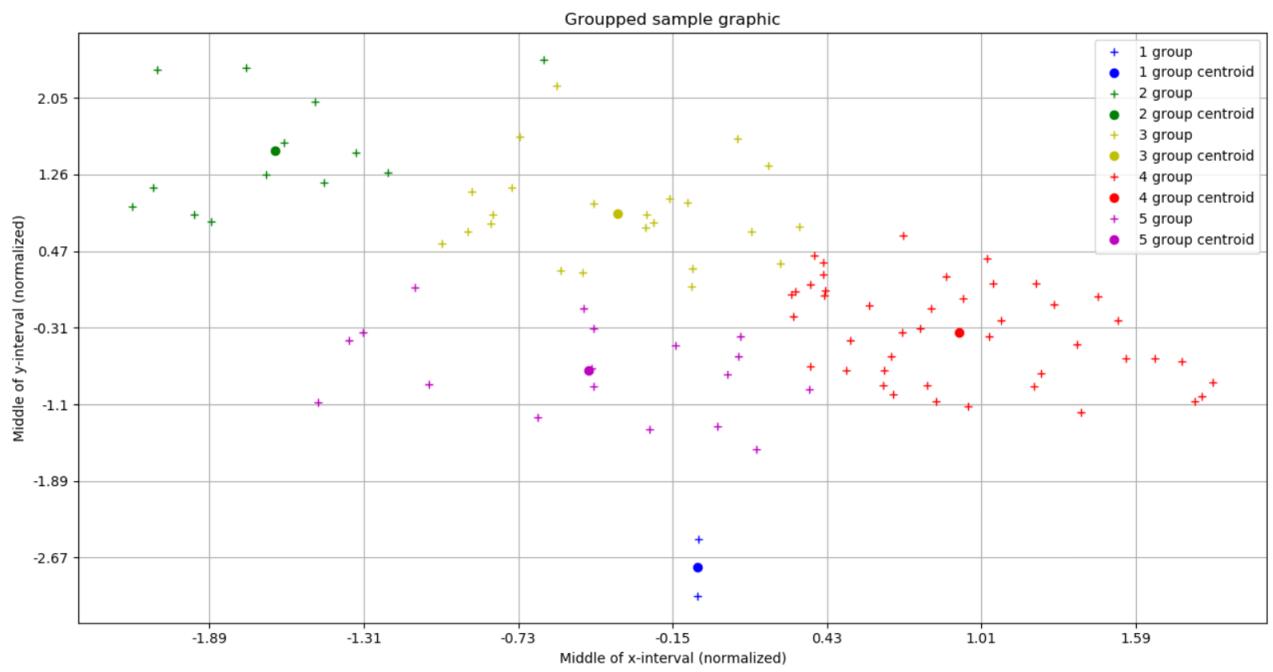
$F1 = 58.28926024128469$   $F2 = 3061.607185477403$   $F3 = 5.743824522497709$   
 $F1 = 60.29578140646611$   $F2 = 3115.7778113798845$   $F3 = 13.547694963099735$   
 $F1 = 60.29578140646611$   $F2 = 3115.7778113798845$   $F3 = 13.547694963099735$

Рисунок 3.6 – Полученные кластеры и оценки качества разбиения при  $k = 4$  и с пересчетом центров после добавления каждой новой точки в соответствующий кластер



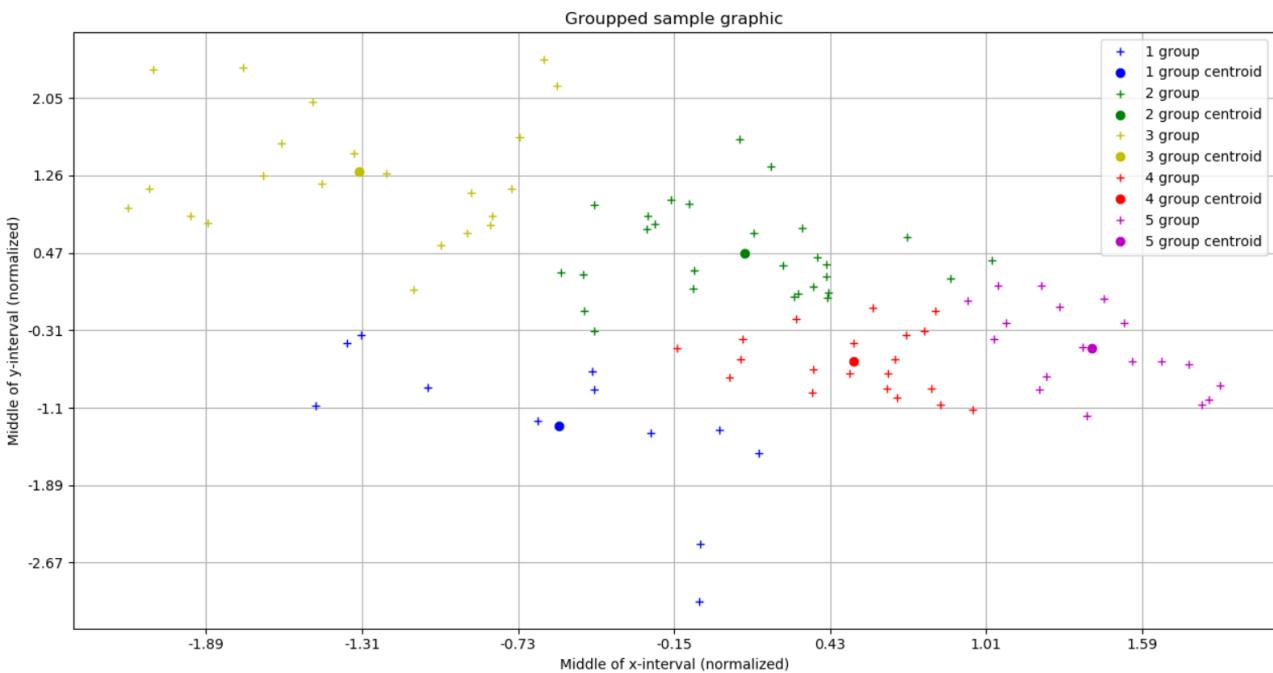
$F1 = 60.19763712970449$   $F2 = 3492.749683367008$   $F3 = 17.07889154119203$   
 $F1 = 55.104778073521814$   $F2 = 2931.4646506313006$   $F3 = 7.703223421438579$   
 $F1 = 52.719666937519996$   $F2 = 2699.4351296982913$   $F3 = 6.703642779439124$   
 $F1 = 50.615656809451366$   $F2 = 2578.812535485243$   $F3 = 5.75454177680596$   
 $F1 = 50.1494255589069$   $F2 = 2565.376352111106$   $F3 = 5.492154080801662$   
 $F1 = 49.96453825075999$   $F2 = 2521.648821009176$   $F3 = 5.752303568645811$   
 $F1 = 49.96453825075999$   $F2 = 2521.648821009176$   $F3 = 5.752303568645811$

Рисунок 3.7 – Полученные кластеры и оценки качества разбиения при  $k = 4$  и с пересчетом центров только после завершения очередного цикла алгоритма



$F_1 = 57.75008219271205$   $F_2 = 2982.7759929693966$   $F_3 = 10.996411954835654$   
 $F_1 = 47.49376452527104$   $F_2 = 2618.0057642216093$   $F_3 = 4.010004322425181$   
 $F_1 = 45.36023820793403$   $F_2 = 2706.2960521233545$   $F_3 = 3.15214501586247$   
 $F_1 = 45.36023820793403$   $F_2 = 2706.2960521233545$   $F_3 = 3.15214501586247$

Рисунок 3.8 – Полученные кластеры и оценки качества разбиения при  $k = 5$  и с пересчетом центров после добавления каждой новой точки в соответствующий кластер

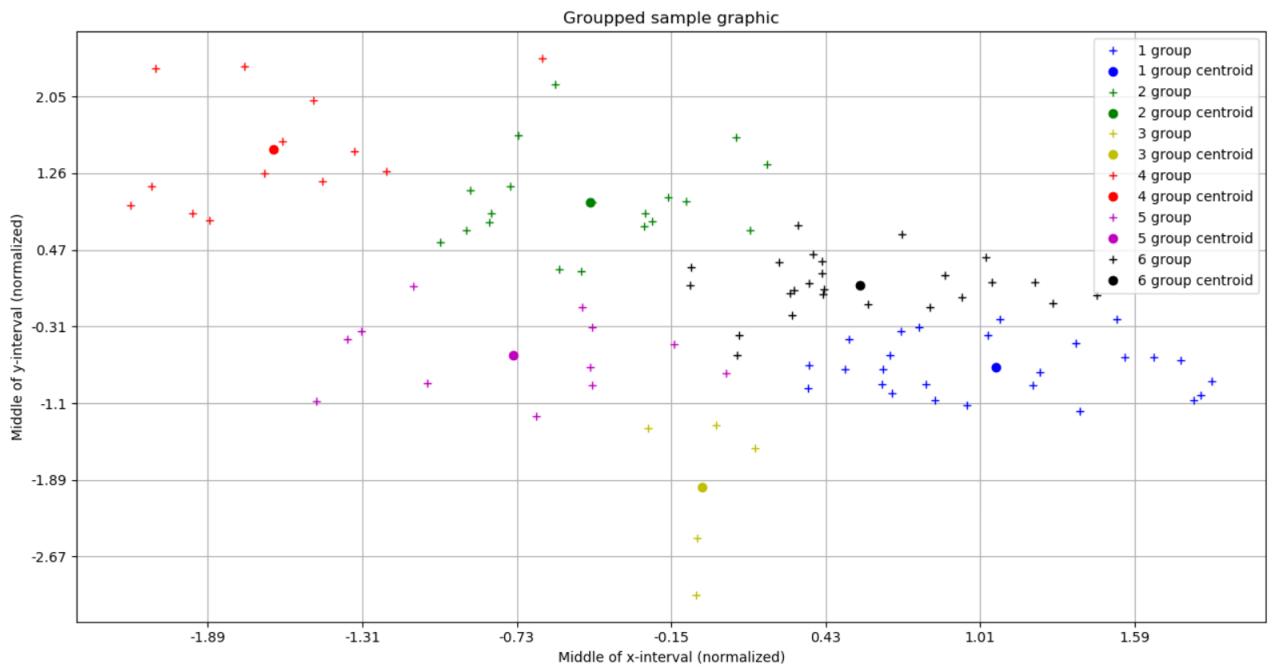


```

F1 = 47.29339943287443 F2 = 2305.34772287817 F3 = 1.6189140102919717
F1 = 45.86386577367037 F2 = 2066.659454848685 F3 = 3.844631503499008
F1 = 44.74449235030803 F2 = 1895.2334463695765 F3 = 3.7636151930360566
F1 = 44.20267782944892 F2 = 1822.7400121269998 F3 = 3.501013736366148
F1 = 43.89790439900459 F2 = 1790.5556132483246 F3 = 3.776408913879036
F1 = 43.8163781144112 F2 = 1785.4565039982847 F3 = 3.791249383860155
F1 = 43.8163781144112 F2 = 1785.4565039982847 F3 = 3.791249383860155

```

Рисунок 3.9 – Полученные кластеры и оценки качества разбиения при  $k = 5$  и с пересчетом центров только после завершения очередного цикла алгоритма

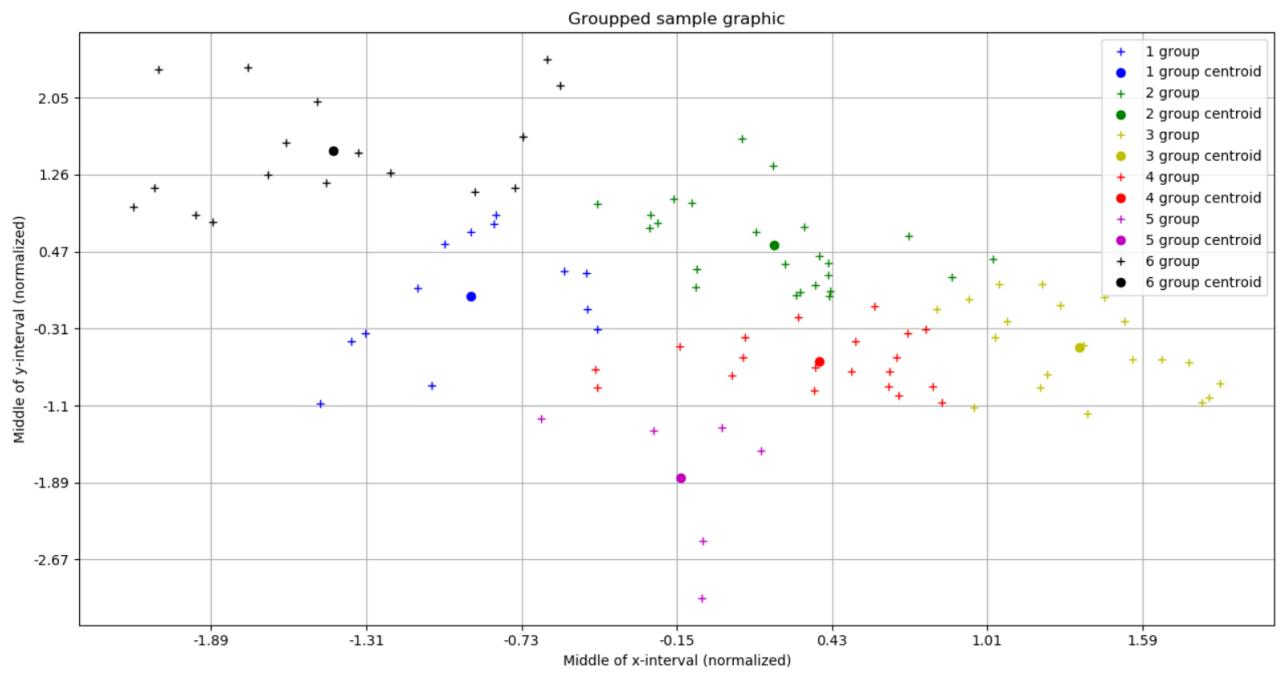


```

F1 = 36.18383673216885 F2 = 1408.1105985620982 F3 = 5.319017593696869
F1 = 34.52867716350937 F2 = 1287.312426470701 F3 = 2.8314800310405364
F1 = 34.52867716350937 F2 = 1287.312426470701 F3 = 2.8314800310405364

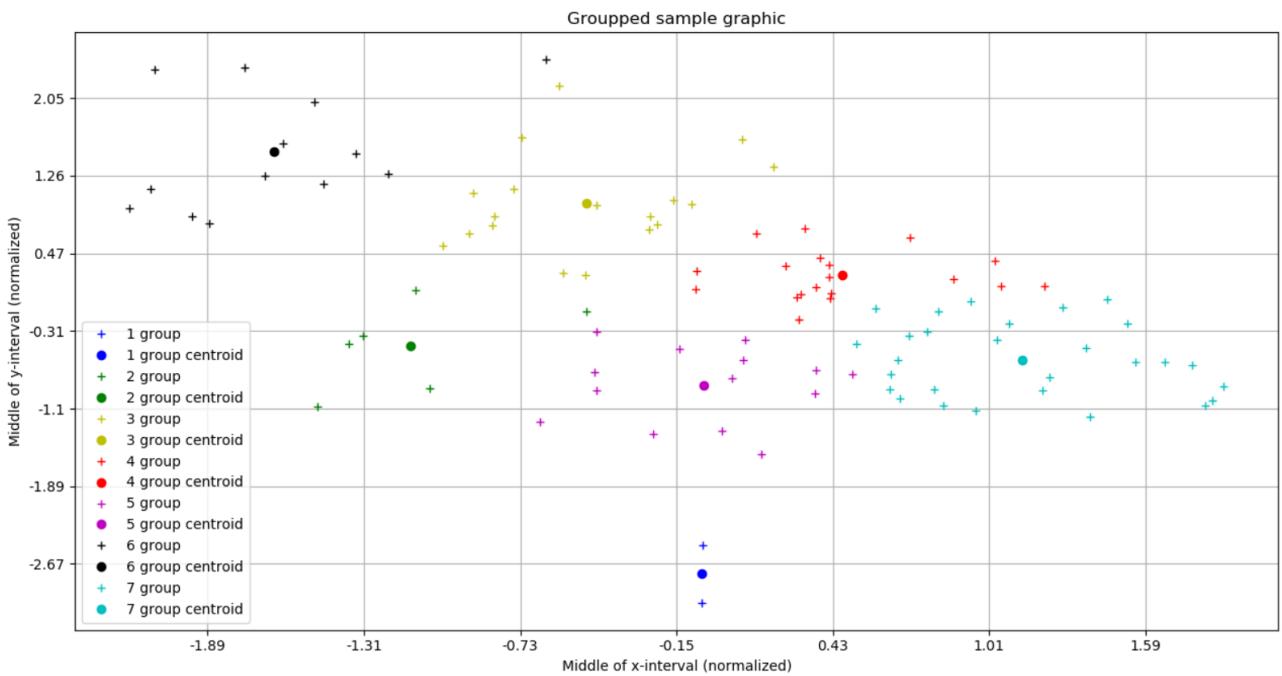
```

Рисунок 3.10 – Полученные кластеры и оценки качества разбиения при  $k = 6$  и с пересчетом центров после добавления каждой новой точки в соответствующий кластер



$F1 = 65.43448436143987$   $F2 = 3398.6917645566764$   $F3 = 7.440363557805173$   
 $F1 = 53.308578467566015$   $F2 = 2109.349364359881$   $F3 = 11.633657072224652$   
 $F1 = 44.684080654987866$   $F2 = 1634.5375089068043$   $F3 = 10.3933019900155$   
 $F1 = 42.167632298803305$   $F2 = 1486.5119708616521$   $F3 = 9.219369566027279$   
 $F1 = 38.64451276237526$   $F2 = 1419.918183857191$   $F3 = 7.138005557515912$   
 $F1 = 37.62872609090315$   $F2 = 1350.721129872909$   $F3 = 5.743631585715163$   
 $F1 = 37.02591053027983$   $F2 = 1331.1952577979282$   $F3 = 5.420342487741767$   
 $F1 = 36.77983374399376$   $F2 = 1298.9653303259936$   $F3 = 5.469305989977952$   
 $F1 = 36.71461346033792$   $F2 = 1292.6627067022262$   $F3 = 5.62614196375646$   
 $F1 = 36.71461346033792$   $F2 = 1292.6627067022262$   $F3 = 5.62614196375646$

Рисунок 3.11 – Полученные кластеры и оценки качества разбиения при  $k = 6$  и с пересчетом центров только после завершения очередного цикла алгоритма

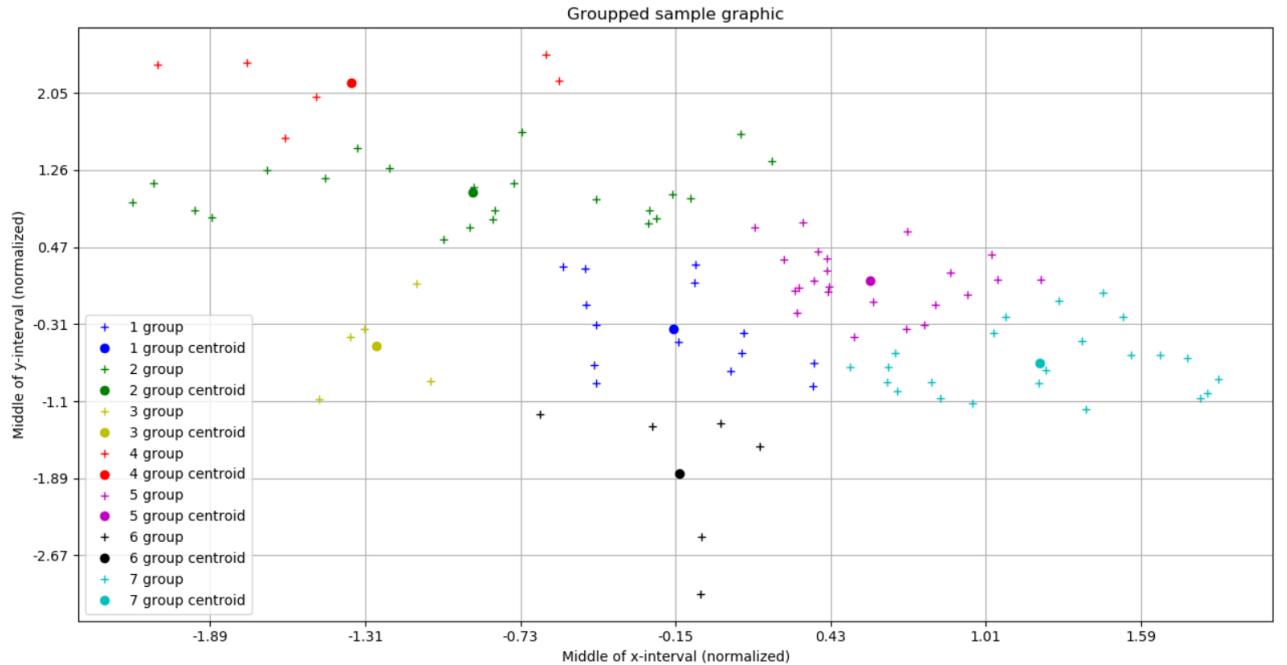


```

F1 = 38.37302601869447 F2 = 1273.0853275707905 F3 = 3.4915946648511085
F1 = 30.010659508000742 F2 = 1069.9820779713286 F3 = 2.8765490437402246
F1 = 29.567381144950573 F2 = 1034.4271800980396 F3 = 1.989874266617261
F1 = 29.509490307045876 F2 = 1054.4071453066 F3 = 1.8372962869140899
F1 = 30.138959888466594 F2 = 1130.7173240034008 F3 = 1.824055965279629
F1 = 30.138959888466594 F2 = 1130.7173240034008 F3 = 1.824055965279629

```

Рисунок 3.12 – Полученные кластеры и оценки качества разбиения при  $k = 7$  и с пересчетом центров после добавления каждой новой точки в соответствующий кластер



```

F1 = 41.80199244335502 F2 = 1729.588512999395 F3 = 2.716584911411287
F1 = 37.91303601643676 F2 = 1506.1992702611144 F3 = 5.581179006668578
F1 = 36.17793843695938 F2 = 1404.159353310337 F3 = 5.638330700232615
F1 = 35.5698775263453 F2 = 1347.5944283964538 F3 = 5.573608216335875
F1 = 35.00057978313879 F2 = 1307.817800699795 F3 = 5.591404311198676
F1 = 35.00057978313879 F2 = 1307.817800699795 F3 = 5.591404311198676

```

Рисунок 3.13 – Полученные кластеры и оценки качества разбиения при  $k = 7$  и с пересчетом центров только после завершения очередного цикла алгоритма

Стоит отметить, что в ходе работы алгоритма  $k$ -means оценки качества разбиения гарантированно минимизировались с каждым новым циклом при пересчете центров кластеров по их окончании (каждое следующее значение  $F_1$ ,  $F_2$  или  $F_3$  на новом цикле было практически всегда гарантированно меньше предыдущего). При больших значениях количества кластеров эта тенденция прослеживалась и при другом режиме работы. Наименьшие значения оценок

были достигнуты при наибольшем из допустимых значений количества кластеров для разбиения (а именно  $k = 7$ ).

Таким образом, наилучшим вариантом разбиения можно считать кластеризацию при  $k = 7$  с пересчетом центров кластеров после добавления каждой новой точки в соответствующий кластер.

### 3.3. Метод поиска сгущений

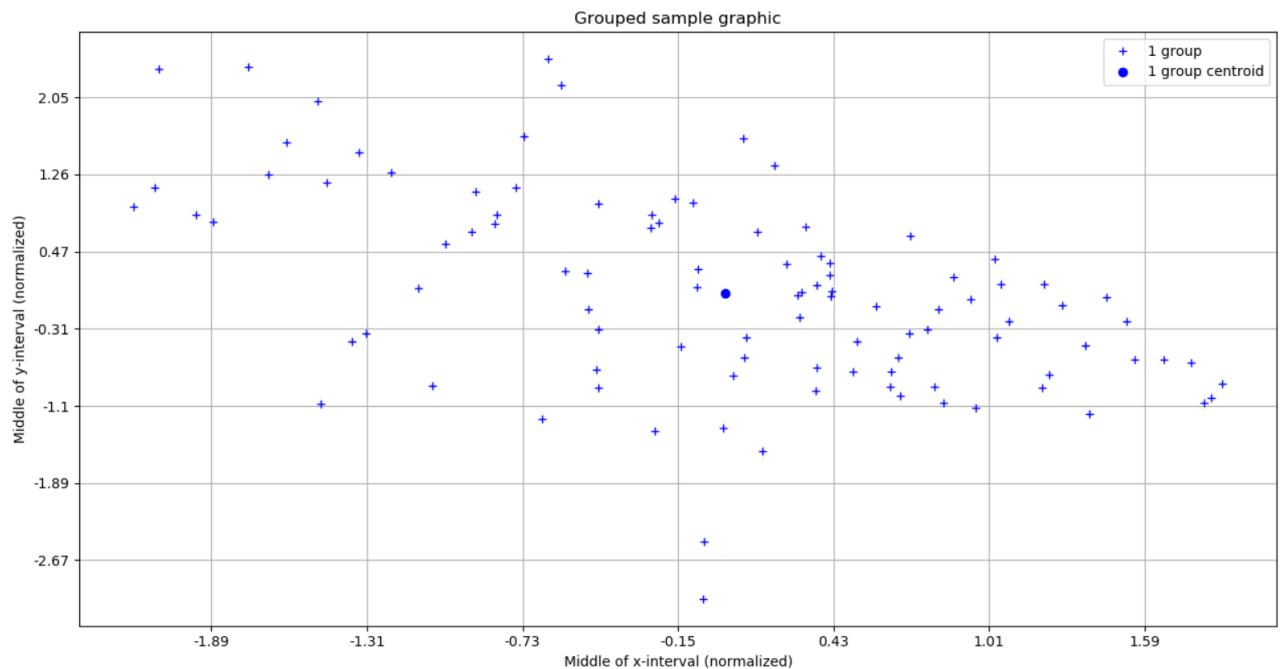
Для работы метода поиска сгущений необходим начальный радиус для формирования гиперсфер  $R$ . В начале работы алгоритма рассчитывается так называемая матрица расстояний, которая содержит дистанции между всеми парами точек на множестве (дистанции вычисляются как Евклидово расстояние). Исходя из значений элементов матрицы можно ограничить диапазон интересующих значений вводимого пользователем радиуса:  $R_{min} = \min_{i,j} d_{ij}; R_{max} = \max_{i,j} d_{ij}; R = R_{min} + \delta$  или  $R = R_{max} - \delta; \delta > 0$ .

Для рассматриваемой выборки:  $R_{min} = 0.0; R_{max} = 5.768104$ .

В рамках тестирования алгоритма с целью поиска наилучшего варианта разбиения было рассмотрено несколько значений  $R$ . Для определения каждого кластера используется это значение в качестве стартового радиуса гиперсферы, а после достижения устойчивости кластерного набора корректируется, изменяясь на небольшую величину, в случае обнаружения пересечений (наличие «спорных точек» – объектов, принадлежащих сразу нескольким кластерам). При достаточно малых значениях радиуса в кластер может попасть только одна точка. Таким образом, объясняется одно из свойств метода поиска сгущений: меньшие значения радиуса порождают большее количество кластеров в результате разбиений.

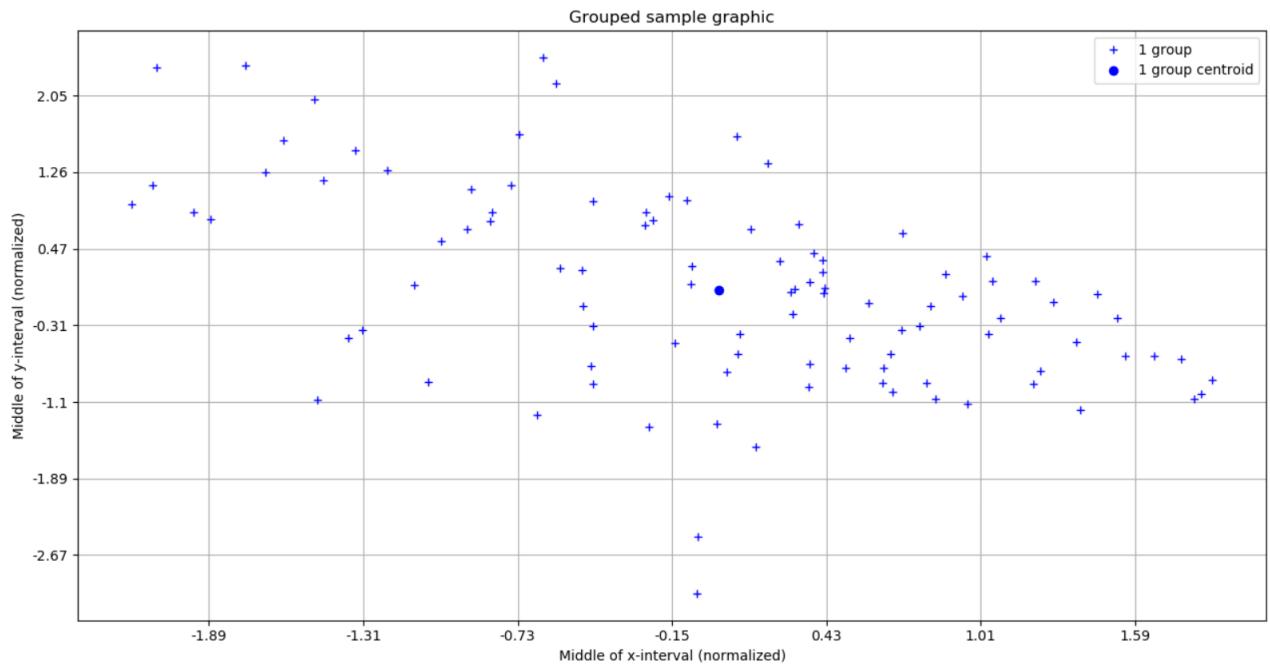
В ходе выполнения поставленной задачи, рассматривались различные значения радиусов от  $R_{max} = \max_{i,j} d_{ij} = 5.768104$  до минимально возможного значения, при котором количество кластеров не будет превышать полученную ранее верхнюю оценку  $\bar{k} = 7$ . Для каждого варианта разбиения были вычислены

оценки его качества: сумма квадратов расстояний до центров кластеров ( $F_1 = \sum_{k=1}^K \sum_{i=1}^{N_k} d^2(X_i^{(k)}, \bar{X}^{(k)}) \Rightarrow \min$ ), сумма внутрикластерных расстояний между объектами ( $F_2 = \sum_{k=1}^K \sum_{X_i, X_j \in S_k} d^2(X_i, X_j) \Rightarrow \min$ ) и сумма внутрикластерных дисперсий ( $F_3 = \sum_{k=1}^K \sum_{j=1}^{N_k} \sigma_{ij}^2 \Rightarrow \min$ ). В качестве расстояния между точками использовалось Евклидово. На рис. 3.14 – 3.20 приведены результаты работы алгоритма для разных значений  $R$  (графическая иллюстрация полученного разбиения и листинг программы с оценками его качества).



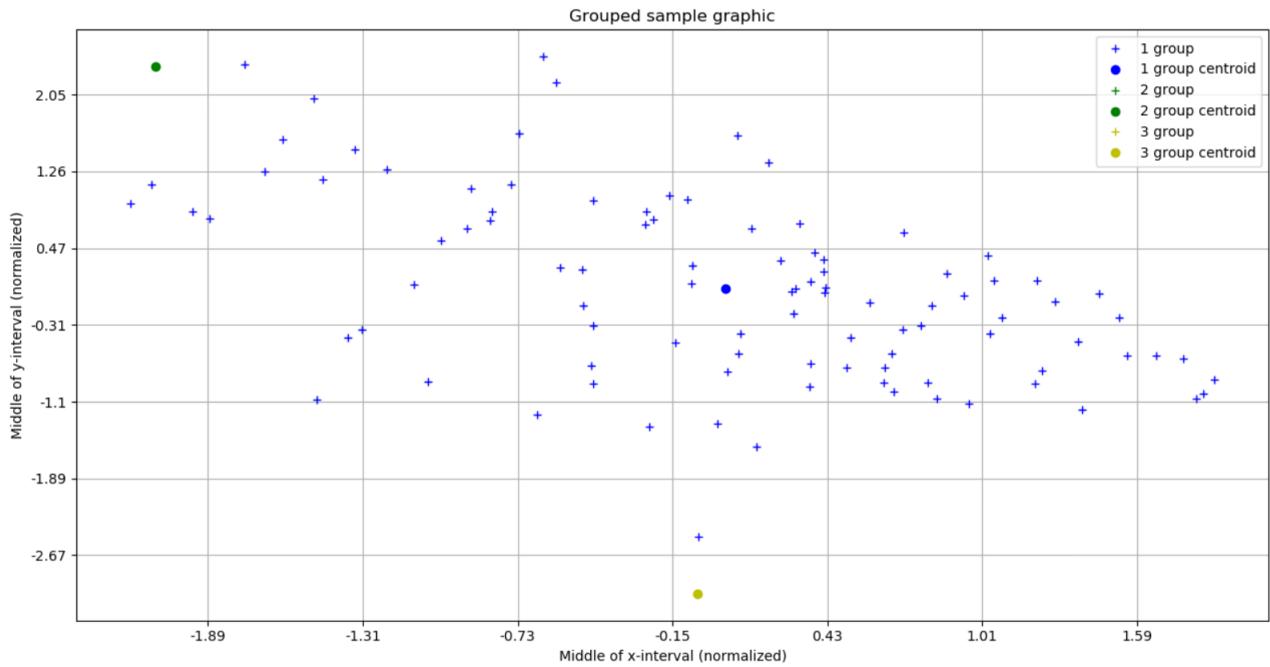
`F1 = 202.83607936379065 F2 = 40567.215872758235 F3 = 56.71578126180981`

Рисунок 3.14 – Результаты работы программы для  $R = 5.5$



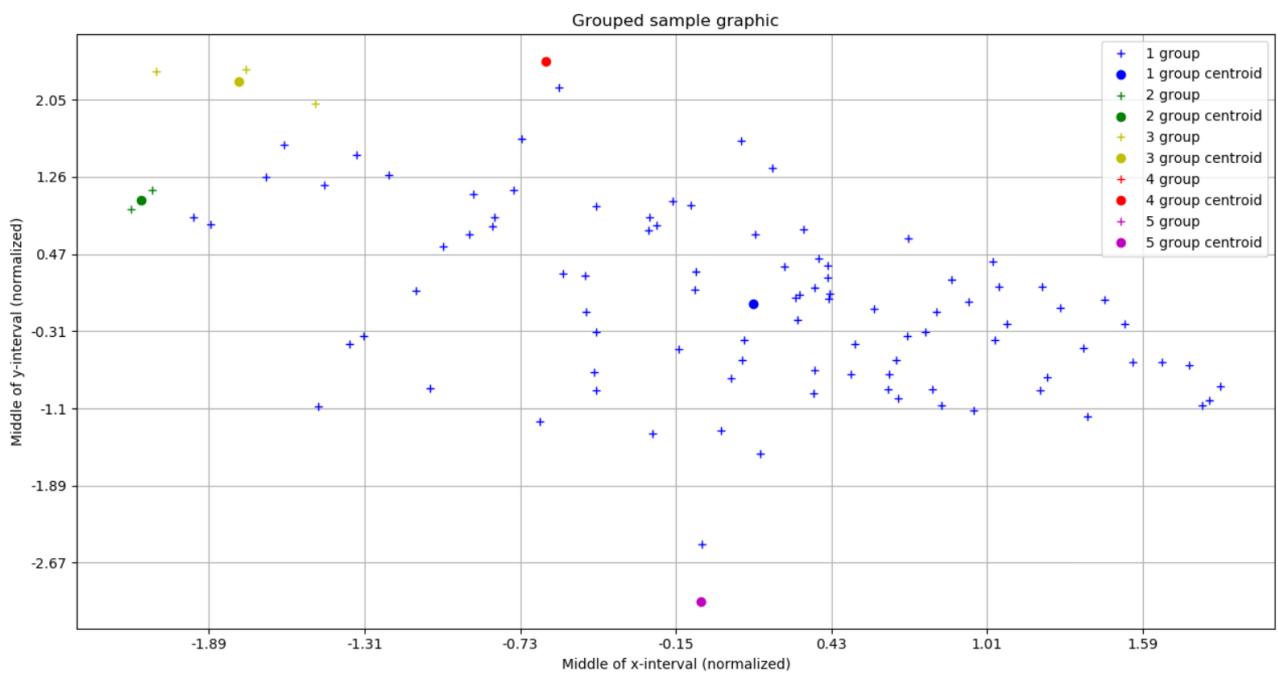
$F_1 = 202.83607936379065 \quad F_2 = 40567.215872758235 \quad F_3 = 56.71578126180981$

Рисунок 3.15 – Результаты работы программы для  $R = 4$



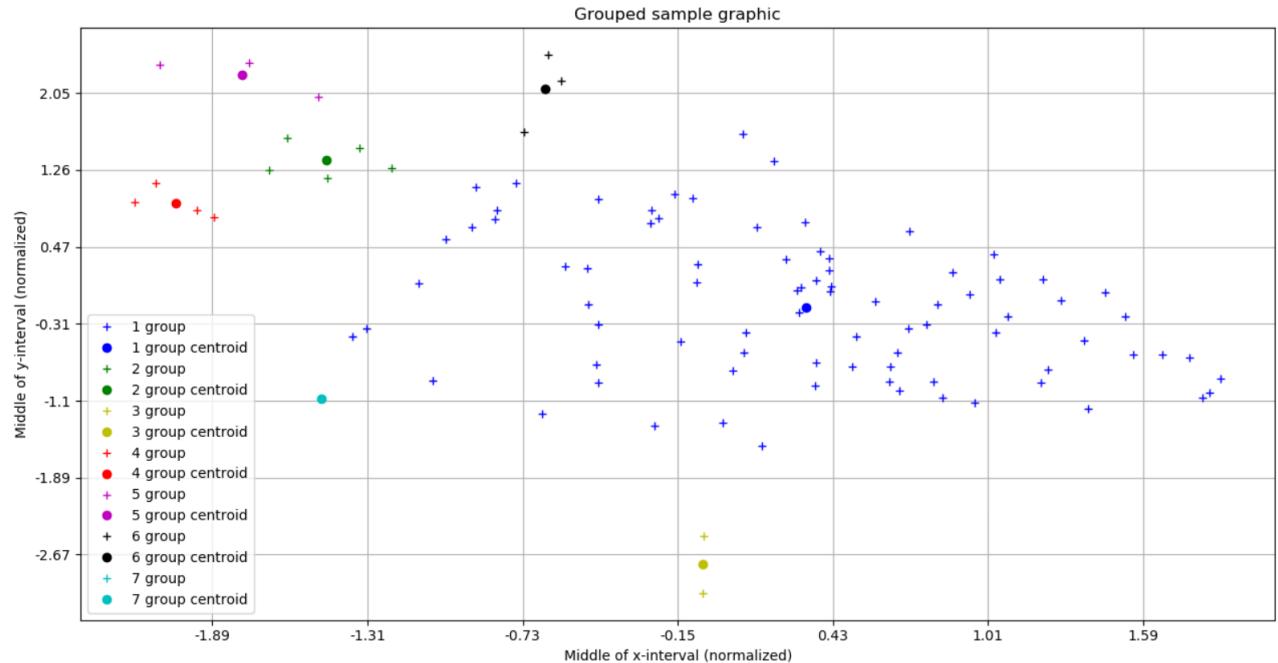
$F_1 = 183.37244264294694 \quad F_2 = 35940.99875801764 \quad F_3 = 50.37182277736108$

Рисунок 3.16 – Результаты работы программы для  $R = 3$



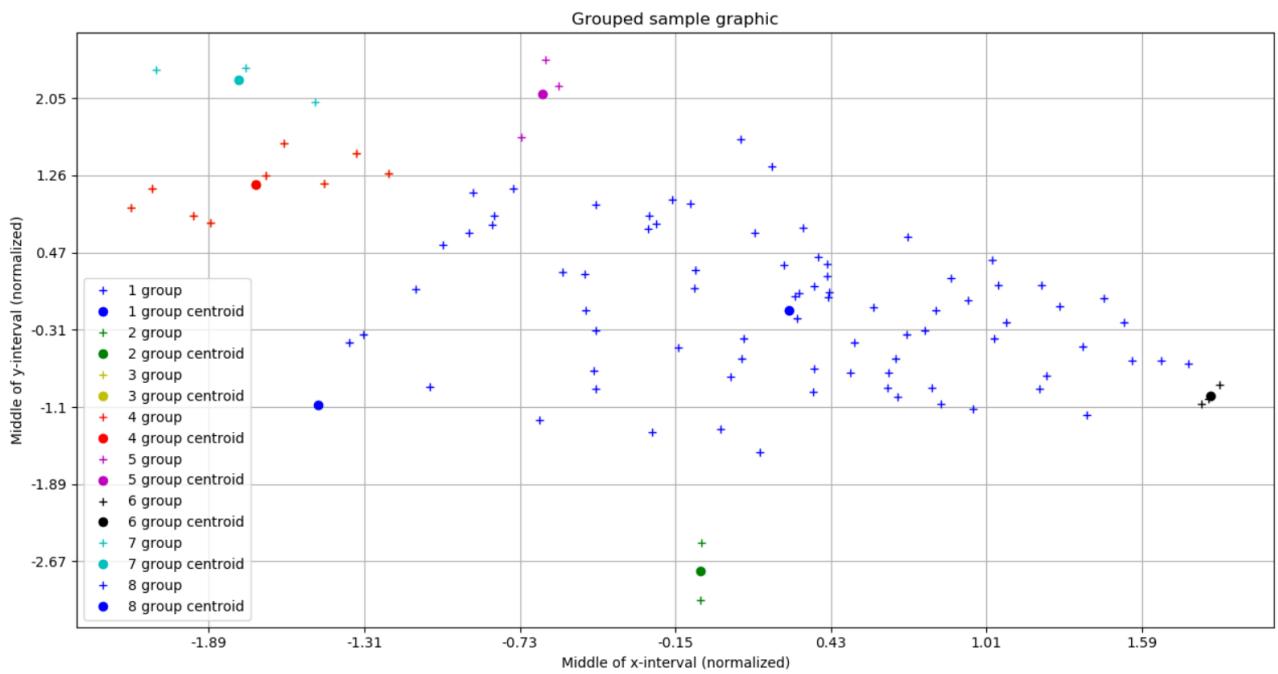
$F_1 = 149.640645637864 \quad F_2 = 27783.278324308005 \quad F_3 = 39.03109793538321$

Рисунок 3.17 – Результаты работы программы для  $R = 2.5$



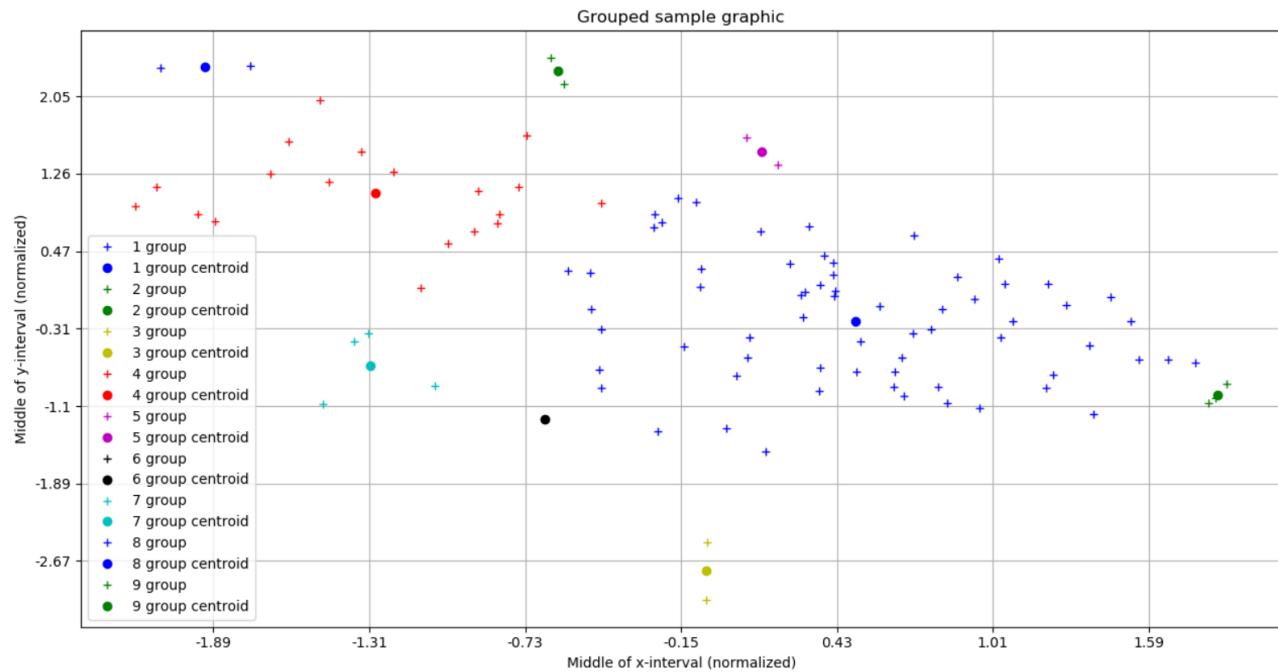
$F_1 = 95.00998788879119 \quad F_2 = 15403.082920865832 \quad F_3 = 12.608845320300269$

Рисунок 3.18 – Результаты работы программы для  $R = 2$



$$F_1 = 88.30773418437253 \quad F_2 = 13408.67583889652 \quad F_3 = 11.304186818676474$$

Рисунок 3.19 – Результаты работы программы для  $R = 1.75$



$$F_1 = 56.707232103660836 \quad F_2 = 6525.745513276366 \quad F_3 = 3.783723480300598$$

Рисунок 3.20 – Результаты работы программы для  $R = 1.5$

Значения оценок качества разбиений  $F_1$ ,  $F_2$  и  $F_3$  минимизируются с увеличением количества требуемых кластеров (уменьшением стартового радиуса гиперсферы для их формирования). Из интересующих значений  $R$  (не превышающих верхнюю оценку радиуса  $R_{max}$  и позволяющих получить

количество кластеров в пределах соответствующей верхней оценки  $\bar{k}$  наименьшими результатами вычислений  $F_1$ ,  $F_2$  и  $F_3$  обладает разбиение при  $R = 2$ . Значения  $F_1 \approx 95.009988$ ,  $F_2 \approx 15403.082921$  и  $F_3 \approx 12.608845$  при таком разбиении превышают значения, полученные при помощи алгоритма  $k$ -средних (при таком же количестве кластеров  $k = 7$ ),  $F_1 \approx 30.13896$ ,  $F_2 \approx 1130.717324$  и  $F_3 \approx 1.824056$ . Это означает, что для выполнения кластеризации рассматриваемой двумерной выборки предпочтительнее использовать алгоритм  $k$ -средних.

Полученную конфигурацию необходимо было проверить на устойчивость, для этого алгоритм был запущен заново при  $R = 2$ . Теперь в качестве стартовых центроид для каждого кластера будут назначаться одни и те же точки (при одинаковых разбиениях), итоги разбиений будут сохраняться для анализа, а сам радиус – изменяться на небольшую величину. В данных условиях, если при рассмотренных радиусах составы кластеров будут идентичны, то разбиение можно будет считать устойчивым. Результаты проверки на устойчивость приведены на рис. 3.21 – 3.26.

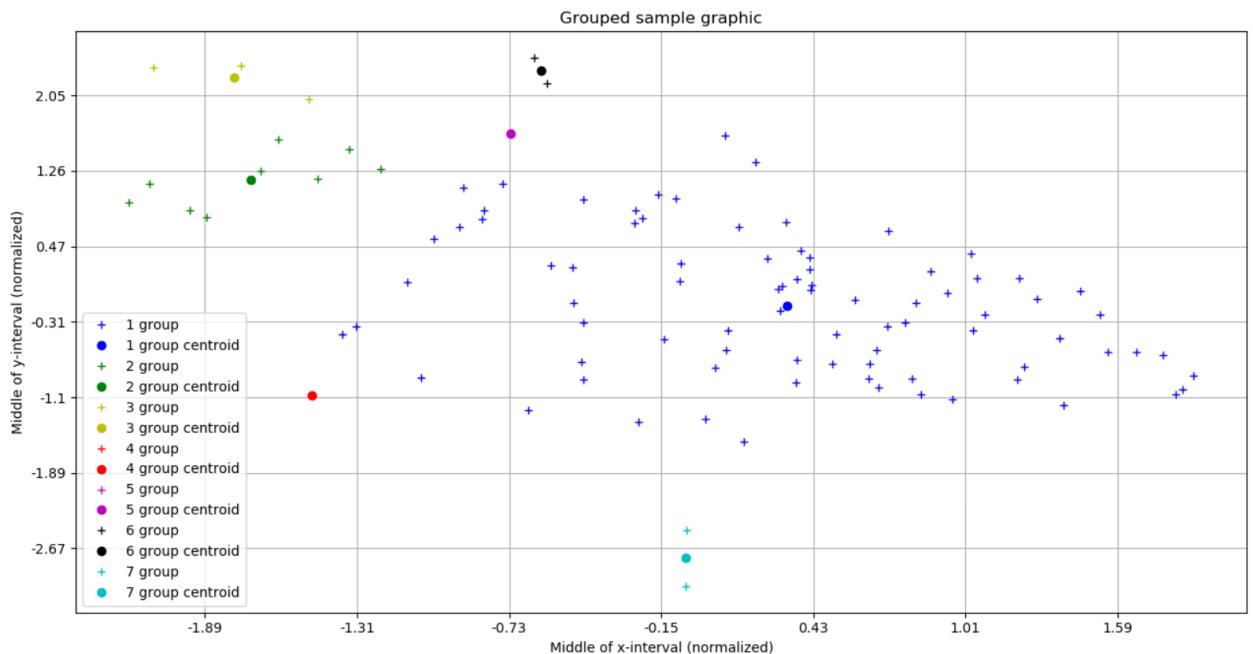


Рисунок 3.21 – Итоги кластеризации ( $R = 2$ )

```

1 - 11,13,15,16,17,18,19,20,21,22,24,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,45,46,47,
2 - 48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,
3 - 78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,
4 - 0,1,3,4,6,7,10,12,14,
5 - 2,5,8,
6 - 9,
7 - 23,
8 - 25,26,
9 - 43,44,

```

F1 = 95.85622442598142 F2 = 15425.040901653054 F3 = 12.648540523032386

Рисунок 3.22 – Состав кластеров и листинг при  $R = 2.01$

```

1 - 11,13,15,16,17,18,19,20,21,22,24,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,45,46,47,
2 - 48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,
3 - 78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,
4 - 0,1,3,4,6,7,10,12,14,
5 - 2,5,8,
6 - 9,
7 - 23,
8 - 25,26,
9 - 43,44,

```

F1 = 95.85622442598142 F2 = 15425.040901653054 F3 = 12.648540523032386

Рисунок 3.23 – Состав кластеров и листинг при  $R = 2.005$

```

1 - 11,13,15,16,17,18,19,20,21,22,24,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,45,46,47,
2 - 48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,
3 - 78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,
4 - 0,1,3,4,6,7,10,12,14,
5 - 2,5,8,
6 - 9,
7 - 23,
8 - 25,26,
9 - 43,44,

```

F1 = 95.85622442598142 F2 = 15425.040901653054 F3 = 12.648540523032386

Рисунок 3.24 – Состав кластеров и листинг при  $R = 2$

```

1 - 11,13,15,16,17,18,19,20,21,22,24,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,45,46,47,
2 - 48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,
3 - 78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,
4 - 0,1,3,4,6,7,10,12,14,
5 - 2,5,8,
6 - 9,
7 - 23,
8 - 25,26,
9 - 43,44,

```

F1 = 95.85622442598142 F2 = 15425.040901653054 F3 = 12.648540523032386

Рисунок 3.25 – Состав кластеров и листинг при  $R = 1.995$

```

1 - 11,13,15,16,17,18,19,20,21,22,24,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,45,46,47,
2 - 48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,
3 - 78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,
4 - 0,1,3,4,6,7,10,12,14,
5 - 2,5,8,
6 - 9,
7 - 23,
8 - 25,26,
9 - 43,44,

```

F1 = 95.85622442598142 F2 = 15425.040901653054 F3 = 12.648540523032386

Рисунок 3.26 – Состав кластеров и листинг при  $R = 1.99$

При изменении радиуса на небольшую величину, состав кластеров и показания листинга программы (оценки качества разбиения) остаются такими же, как и при исходном запуске с  $R = 2$ . Это позволяет сделать вывод об устойчивости разбиения к погрешностям.

### 3.4. Выводы

В данном разделе был произведен кластерный анализ рассматриваемой двумерной выборки. В ходе выполнения анализа были реализованы алгоритм  $k$ -средних в двух вариациях и метод поиска сгущений.

По итогам выполнения кластеризации было установлено, что наилучшим вариантом для разбиения является алгоритм  $k$ -средних с пересчетом центроид после добавления каждой новой точки в соответствующий кластер. При таком режиме были получены наименьшие значения оценок качества разбиения (суммы квадратов расстояний до центров кластеров ( $F_1 \approx 30.13896$ ), суммы внутрикластерных расстояний между объектами ( $F_2 \approx 1130.717324$ ) и суммы внутрикластерных дисперсий ( $F_3 \approx 1.824056$ )).

Для обоих алгоритмов наилучшим количеством кластеров можно считать  $k = 7$ . Наилучшее разбиение методом поиска сгущений было проверено на устойчивость. При незначительном изменении стартового радиуса гиперсфер состав кластеров не изменился, следовательно, разбиение устойчиво.

## ЗАКЛЮЧЕНИЕ

В рамках данной работы были разработаны и протестированы программные средства для формирования двумерной выборки из заданной генеральной совокупности и подготовки ее к проведению статистического анализа. Полученная выборка подверглась корреляционному и регрессионному, а также кластерному анализу. Регрессионный и корреляционный анализ сопровождался построением выборочных прямых среднеквадратической регрессии и выборочных корреляционных кривых соответственно. В рамках кластерного анализа были исследованы алгоритмы  $k$ -средних и поиска сгущений.

Исследуемое распределение обладает незначительной асимметрией; оно низкое и пологое относительно «эталонного» нормального распределения. Гипотеза о нормальности распределения принята на уровне значимости  $\alpha = 0.05$  (так как  $\chi^2_{\text{набл}} \leq \chi^2_{\text{крит}}$ ), а гипотеза о равенстве нулю коэффициента корреляции отвергнута (следовательно, он значим).

По итогам выполнения кластеризации было установлено, что самым оптимальным вариантом для разбиения является алгоритм  $k$ -средних с пересчетом центроид после добавления каждой новой точки в соответствующий кластер. Для обоих алгоритмов наилучшим количеством кластеров можно считать  $k = 7$ . Наилучшее разбиение методом поиска сгущений было проверено на устойчивость.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Смирнов Н.А., Экало А.В. Методы обработки экспериментальных данных: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2009.
2. Белоногов А.М., Попов Ю.И., Посредник О.В. Статистическая обработка результатов физического эксперимента [Комплект]: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2009.
3. Егоров В.А. и др. Анализ однородных статистических данных: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2005.
4. Морозов В.В., Соботковский Б.Е., Шейнман И.Л. Методы обработки результатов физического эксперимента: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2004.
5. Буре В.М., Парилова Е.М., Свиркин М.В. Математическая статистика. СПб.: факультет ПМ ПУ СПбГУ, 2007.
6. Митин И.В., Русаков В.С. Анализ и обработка экспериментальных данных. М.: Физический факультет МГУ, 2006.
7. Кобзарь А.И. Прикладная математическая статистика. М.: Физматлит, 2006.
8. Котельников Р.Б. Анализ результатов наблюдений. М.: Энергоатомиздат, 1986.
9. NumPy Documentation // NumPy. URL: <https://numpy.org/doc/stable/contents.html> (дата обращения: 08.04.2021).
10. Pyplot tutorial // Matplotlib. URL: [https://matplotlib.org/2.0.2/users/pyplot\\_tutorial.html](https://matplotlib.org/2.0.2/users/pyplot_tutorial.html) (дата обращения: 09.04.2021).

**ПРИЛОЖЕНИЕ А**  
**ИСХОДНЫЙ КОД ПРОГРАММЫ ДЛЯ ВЫРАВНИВАНИЯ**  
**СТАТИСТИЧЕСКИХ РЯДОВ**

```
import matplotlib.pyplot as plt
import numpy as np
import random, math

def save_current_sample(sample, file_path):
    file = open(file_path, 'w')
    for i in range(sample.shape[0]):
        for j in range(sample.shape[1]):
            file.write(str(sample[i][j]))
            if j != (sample.shape[1] - 1): file.write(',')
        if i != (sample.shape[0] - 1): file.write('\n')
    file.close()

N = 100
data_matrix = np.ndarray((N, 11))

# 1
gen_data_file_path = input('Enter gen. data file path: ')
data_sampling_mode = input('Enter data sampling mode (random / random_unique / mechanical): ')
gen_data_file = open(gen_data_file_path, 'r')
lines = gen_data_file.readlines()
gen_data_matrix = np.ndarray((len(lines) - 1,
                             data_matrix.shape[1]))
for line in range(len(lines)):
    if line == 0: continue
    elements = lines[line].split(',')
    for i in range(data_matrix.shape[1]): gen_data_matrix[line - 1][i] = float(elements[i])
gen_data_file.close()

data_ind_seq = np.ndarray([N])
if data_sampling_mode == 'mechanical':
    for i in range(N): data_ind_seq[i] = int(i * gen_data_matrix.shape[0] / N)
elif data_sampling_mode == 'random':
    for i in range(N): data_ind_seq[i] = random.randint(0, gen_data_matrix.shape[0])
```

```

elif data_sampling_mode == 'random_unique':
    for i in range(N):
        data_ind_seq[i] = random.randint(0,
gen_data_matrix.shape[0])
        if i > 0:
            while data_ind_seq[i] in data_ind_seq[0:i]:
                data_ind_seq[i] = random.randint(0,
gen_data_matrix.shape[0])
    for i in range(data_matrix.shape[0]):
        for j in range(data_matrix.shape[1]):
            data_matrix[i][j] =
gen_data_matrix[int(data_ind_seq[i])][j]
    save_current_sample(data_matrix, './sample.csv')

# 2.a
data_ind_seq.sort()
for i in range(data_matrix.shape[0]):
    for j in range(data_matrix.shape[1]):
        data_matrix[i][j] =
gen_data_matrix[int(data_ind_seq[i])][j]
    save_current_sample(data_matrix, './stat_range.csv')

data_ind_seq = np.lexsort((data_matrix[:, 10], data_matrix[:, 9],
data_matrix[:, 8], data_matrix[:, 7],
data_matrix[:, 6], data_matrix[:, 5],
data_matrix[:, 4], data_matrix[:, 3],
data_matrix[:, 2], data_matrix[:, 1],
data_matrix[:, 0]))
ranked_sample = data_matrix.copy()
for i in range(data_matrix.shape[0]):
    for j in range(data_matrix.shape[1]):
        ranked_sample[i][j] = data_matrix[int(data_ind_seq[i])][j]
    save_current_sample(ranked_sample, './ranked_range.csv')

# 2.b
var_sample = np.ndarray((0, ranked_sample.shape[1]))
var_sample_freq = np.ndarray((0, 2))
for i in range(ranked_sample.shape[0]):
    if i == 0 or not np.array_equal(ranked_sample[i],
ranked_sample[i - 1]):
        var_sample = np.vstack((var_sample, ranked_sample[i]))
        var_sample_freq = np.vstack((var_sample_freq, [1, 1 / N]))
    else:
        var_sample_freq[var_sample_freq.shape[0] - 1][0] += 1
        var_sample_freq[var_sample_freq.shape[0] - 1][1] =
var_sample_freq[var_sample_freq.shape[0] - 1][0] / N

```

```

save_current_sample(var_sample, './var_range.csv')
save_current_sample(var_sample_freq, './var_range_freq.csv')

k = 1 + 3.31 * np.log10(N)
if int(k) % 2 == 1: k = int(k)
else: k = int(k) + 1
h = (max(var_sample[:, 0]) - min(var_sample[:, 0])) / k
print('k =', k, 'h =', h)

# 2.c
interval_range = np.ndarray((0, 5))
for i in range(1, k + 1):
    interval_range = np.vstack((interval_range, [min(var_sample[:, 0]) + (h * (i - 1)), min(var_sample[:, 0]) + (h * i),
                                                (min(var_sample[:, 0]) + (h * (i - 1)) + min(var_sample[:, 0]) + (h * i)) / 2, 0, 0]))
current_int_ind = 0
for i in range(len(var_sample)):
    if var_sample[i][0] >= interval_range[current_int_ind][1]:
        current_int_ind += 1
    if current_int_ind >= k: current_int_ind = k - 1
    interval_range[current_int_ind][3] += var_sample_freq[i][0]
    interval_range[current_int_ind][4] += var_sample_freq[i][1]
save_current_sample(interval_range, './interval_range.csv')

# 2.d
plt.clf()
plt.title('Abs. frequency polygon')
plt.plot(interval_range[:, 2], interval_range[:, 3], 'bo-')
plt.xticks(interval_range[:, 2], np.around(interval_range[:, 2], 2))
plt.xlabel('Middle of interval'); plt.ylabel('Abs. frequency')
plt.grid()
plt.show()

plt.clf()
plt.title('Rel. frequency polygon')
plt.plot(interval_range[:, 2], interval_range[:, 4], 'bo-')
plt.xticks(interval_range[:, 2], np.around(interval_range[:, 2], 2))
plt.xlabel('Middle of interval'); plt.ylabel('Rel. frequency')
plt.grid()
plt.show()

# 2.e
plt.clf()

```

```

plt.title('Abs. frequency histogram')
plt.bar(interval_range[:, 2], height=interval_range[:, 3] / h,
width=h, color='b', tick_label=np.around(interval_range[:, 2], 2))
plt.xlabel('Middle of interval'); plt.ylabel('n / h')
plt.grid()
plt.show()

plt.clf()
plt.title('Rel. frequency histogram')
plt.bar(interval_range[:, 2], height=interval_range[:, 4] / h,
width=h, color='b', tick_label=np.around(interval_range[:, 2], 2))
plt.xlabel('Middle of interval'); plt.ylabel('n_rel. / h')
plt.grid()
plt.show()

# 2.f
plt.clf()
plt.title('Empirical distribution function')
points = np.ndarray((0, 2))
for i in range(k):
    accumulated_freq = 0
    for j in range(i): accumulated_freq += (interval_range[j][3] /
N)
    points = np.vstack((points, [[interval_range[i][2],
accumulated_freq,
[interval_range[i][2],
accumulated_freq + (interval_range[i][3] / N)]]))
plt.plot(points[:, 0], points[:, 1], 'bo-')
plt.xticks(points[:, 0], np.around(points[:, 0], 2))
plt.xlabel('Middle of interval'); plt.ylabel('Accumulated
frequency')
plt.grid()
plt.show()

plt.clf()
plt.title('Empirical distribution function')
points = np.ndarray((0, 2))
for i in range(k):
    accumulated_freq = 0
    for j in range(i): accumulated_freq += (interval_range[j][4] /
N)
    points = np.vstack((points, [[interval_range[i][2],
accumulated_freq,
[interval_range[i][2],
accumulated_freq + (interval_range[i][4] / N)])))
plt.plot(points[:, 0], points[:, 1], 'bo-')

```

```

plt.xticks(points[:, 0], np.around(points[:, 0], 2))
plt.xlabel('Middle of interval'); plt.ylabel('Accumulated
frequency')
plt.grid()
plt.show()

# 2.g
conditional_options = np.arange(interval_range.shape[0])
C = interval_range[interval_range.shape[0] // 2][2]
N = 100.0
h = 0.0
for interval in range(interval_range.shape[0]):
    h = interval_range[interval][1] - interval_range[interval][0]
    u = int(((interval_range[interval][2] - C) / h + 0.1).round())
    conditional_options[interval] = u
print('Условные варианты:', conditional_options)

M = np.zeros(4)
for k in range(4):
    for j in range(interval_range.shape[0]):
        M[k] += interval_range[j][3] * pow(conditional_options[j], k + 1) / N
print('Условные эмпирические моменты: M1 =', M[0], 'M2 =', M[1],
'M3 =', M[2], 'M4 =', M[3])
m1 = M[0] * h + C
m2 = (M[1] - pow(M[0], 2)) * pow(h, 2)
m3 = (M[2] - 3 * M[1] * M[0] + 2 * pow(M[0], 3)) * pow(h, 3)
m4 = (M[3] - 4 * M[2] * M[0] + 6 * M[1] * pow(M[0], 2) - 3 *
pow(M[0], 4)) * pow(h, 4)
print('Центральные эмпирические моменты: m1 =', m1, 'm2 =', m2,
'm3 =', m3, 'm4 =', m4)

x_sample_mean_u = m1; D_u = m2
print('Выборочное среднее, вычисленное с помощью усл. вариант:', x_sample_mean_u)
print('Дисперсия, вычисленная с помощью усл. вариант:', D_u)
x_sample_mean = 0.0
for i in range(interval_range.shape[0]): x_sample_mean +=
interval_range[i][3] * interval_range[i][2] / N
D = 0.0
for i in range(interval_range.shape[0]): D += interval_range[i][3] *
pow(interval_range[i][2] - x_sample_mean, 2) / N
print('Выборочное среднее, вычисленное с помощью стандартной
формулы:', x_sample_mean)
print('Дисперсия, вычисленная с помощью стандартной формулы:', D)
s_2 = N * D / (N - 1); sigma = math.sqrt(D); S = math.sqrt(s_2)

```

```

print('Исправленная оценка дисперсии:', s_2, '\nСтат. оценки СКО:
sigma =', sigma, 's =', S)

As = m3 / pow(S, 3); E = m4 / pow(S, 4) - 3
print('Стат. оценка коэффи. асимметрии:', As, '\nСтат. оценка
экспессса:', E)

# 2.h
M0 = interval_range[np.argmax(interval_range[:, 3])][2]; m_e = C
print('Мода:', M0, 'Медиана:', m_e)

# 2.i
t_Student = 1.984
print('Дов. инт-л для мат. ожидания: (' , x_sample_mean - t_Student
* S / math.sqrt(N), ';',
      x_sample_mean + t_Student * S / math.sqrt(N), ')')
q = 0.143
print('Дов. инт-л для оценки СКВО: (' , S * (1 - q), ';', S * (1 +
q), ')')

# 2.j
Phi_z = [-0.5, -0.4452, -0.3461, -0.17, 0.0557, 0.2642, 0.4032,
0.5]
Hi_2_watch = 0.0
print('x_i | x_i+1 | n_i | z_i | z_i+1 | Φ(z_i) | Φ(z_i+1) | p_i |
n`_i')
for i in range(interval_range.shape[0]):
    x_i = interval_range[i][0]; x_next = interval_range[i][1]; n_i =
interval_range[i][3]
    if i == 0: z_i = '-inf'
    else: z_i = (x_i - x_sample_mean) / S
    if i == interval_range.shape[0] - 1: z_next = '+inf'
    else: z_next = (x_next - x_sample_mean) / S
    p_i = Phi_z[i + 1] - Phi_z[i]; n_i_ = N * p_i
    print(x_i, ' | ', x_next, ' | ', n_i, ' | ', z_i, ' | ',
z_next, ' | ', Phi_z[i], ' | ', Phi_z[i + 1], ' | ', p_i, ' | ',
n_i_)
    Hi_2_watch += pow(n_i - n_i_, 2) / n_i_
print('χ2набл. =', Hi_2_watch)
k = interval_range.shape[0] - 3
print('χ2крит. ( alpha = 0.05; k =', k, ') = 9.5')

```

**ПРИЛОЖЕНИЕ Б**  
**ИСХОДНЫЙ КОД ПРОГРАММЫ ДЛЯ КОРРЕЛЯЦИОННОГО И**  
**РЕГРЕССИОННОГО АНАЛИЗА**

```
import matplotlib.pyplot as plt
import numpy as np
import math

def save_current_sample(sample, file_path):
    file = open(file_path, 'w')
    for i in range(sample.shape[0]):
        for j in range(sample.shape[1]):
            file.write(str(sample[i][j]))
            if j != (sample.shape[1] - 1): file.write(',')
        if i != (sample.shape[0] - 1): file.write('\n')
    file.close()

N = 100

# 1
sample_file = open('./stat_range.csv', 'r')
lines = sample_file.readlines()
stat_range = np.ndarray((len(lines), 11))
for line in range(len(lines)):
    elements = lines[line].split(',')
    for i in range(stat_range.shape[1]): stat_range[line][i] =
float(elements[i])
sample_file.close()

data_ind_seq = np.lexsort((stat_range[:, 10], stat_range[:, 9],
stat_range[:, 8], stat_range[:, 7],
stat_range[:, 6], stat_range[:, 5],
stat_range[:, 4], stat_range[:, 3],
stat_range[:, 2], stat_range[:, 0],
stat_range[:, 1]))
ranked_sample = stat_range.copy()
for i in range(stat_range.shape[0]):
    for j in range(stat_range.shape[1]):
        ranked_sample[i][j] = stat_range[int(data_ind_seq[i])][j]
    save_current_sample(ranked_sample, './ranked_range_2.csv')

var_sample = np.ndarray((0, ranked_sample.shape[1]))
```

```

var_sample_freq = np.ndarray((0, 2))
for i in range(ranked_sample.shape[0]):
    if i == 0 or not np.array_equal(ranked_sample[i],
    ranked_sample[i - 1]):
        var_sample = np.vstack((var_sample, ranked_sample[i]))
        var_sample_freq = np.vstack((var_sample_freq, [1, 1 / N]))
    else:
        var_sample_freq[var_sample_freq.shape[0] - 1][0] += 1
        var_sample_freq[var_sample_freq.shape[0] - 1][1] =
var_sample_freq[var_sample_freq.shape[0] - 1][0] / N
save_current_sample(var_sample, './var_range_2.csv')
save_current_sample(var_sample_freq, './var_range_freq_2.csv')

interval_range_y = np.ndarray((0, 5))
k = 1 + 3.31 * np.log10(N)
if int(k) % 2 == 1:
    k = int(k)
else:
    k = int(k) + 1
h = (max(var_sample[:, 1]) - min(var_sample[:, 1])) / k
print('k =', k, 'h =', h)
for i in range(1, k + 1):
    int_sample = np.vstack((interval_range_y, [min(var_sample[:, 1]) + (h * (i - 1)), min(var_sample[:, 1]) + (h * i),
                                                (min(var_sample[:, 1]) + (h * (i - 1)) + min(var_sample[:, 1]) + (h * i)) / 2,
                                                0, 0]))
current_int_ind = 0
for i in range(len(var_sample)):
    if var_sample[i][1] >= interval_range_y[current_int_ind][1]:
        current_int_ind += 1
        if current_int_ind >= k: current_int_ind = k - 1
        interval_range_y[current_int_ind][3] += var_sample_freq[i][0]
        interval_range_y[current_int_ind][4] += var_sample_freq[i][1]
save_current_sample(interval_range_y, './interval_range_2.csv')

plt.clf()
plt.title('Abs. frequency polygon')
plt.plot(interval_range_y[:, 2], interval_range_y[:, 3], 'bo-')
plt.xticks(interval_range_y[:, 2], np.around(interval_range_y[:, 2], 2))
plt.xlabel('Middle of interval'); plt.ylabel('Abs. frequency')
plt.grid()
plt.show()

plt.clf()

```

```

plt.title('Rel. frequency polygon')
plt.plot(interval_range_y[:, 2], interval_range_y[:, 4], 'bo-')
plt.xticks(interval_range_y[:, 2], np.around(interval_range_y[:, 2], 2))
plt.xlabel('Middle of interval'); plt.ylabel('Rel. frequency')
plt.grid()
plt.show()

plt.clf()
plt.title('Abs. frequency histogram')
plt.bar(interval_range_y[:, 2], height=interval_range_y[:, 3] / h,
width=h, color='b', tick_label=np.around(interval_range_y[:, 2], 2))
plt.xlabel('Middle of interval'); plt.ylabel('n / h')
plt.grid()
plt.show()

plt.clf()
plt.title('Rel. frequency histogram')
plt.bar(interval_range_y[:, 2], height=interval_range_y[:, 4] / h,
width=h, color='b', tick_label=np.around(interval_range_y[:, 2], 2))
plt.xlabel('Middle of interval'); plt.ylabel('n_rel. / h')
plt.grid()
plt.show()

plt.clf()
plt.title('Empirical distribution function')
points = np.ndarray((0, 2))
for i in range(k):
    accumulated_freq = 0
    for j in range(i): accumulated_freq += (interval_range_y[j][3] /
/ N)
    points = np.vstack((points, [[interval_range_y[i][2],
accumulated_freq,
[interval_range_y[i][2],
accumulated_freq + (interval_range_y[i][3] / N)]]))
plt.plot(points[:, 0], points[:, 1], 'bo-')
plt.xticks(points[:, 0], np.around(points[:, 0], 2))
plt.xlabel('Middle of interval'); plt.ylabel('Accumulated
frequency')
plt.grid()
plt.show()

plt.clf()
plt.title('Empirical distribution function')

```

```

points = np.ndarray((0, 2))
for i in range(k):
    accumulated_freq = 0
    for j in range(i): accumulated_freq += (interval_range_y[j][4] / N)
    points = np.vstack((points, [[interval_range_y[i][2], accumulated_freq],
                                 [interval_range_y[i][2], accumulated_freq + (interval_range_y[i][4] / N)]]))
plt.plot(points[:, 0], points[:, 1], 'bo-')
plt.xticks(points[:, 0], np.around(points[:, 0], 2))
plt.xlabel('Middle of interval'); plt.ylabel('Accumulated frequency')
plt.grid()
plt.show()

# 2
conditional_options = np.arange(k)
C = interval_range_y[k // 2][2]
for interval in range(k):
    u = int(((interval_range_y[interval][2] - C) / h + 0.1).round())
    conditional_options[interval] = u
print('Условные варианты:', conditional_options)

M = np.zeros(4)
for i in range(4):
    for j in range(k):
        M[i] += interval_range_y[j][3] * pow(conditional_options[j], i + 1) / N
print('Условные эмпирические моменты: M1 =', M[0], 'M2 =', M[1],
      'M3 =', M[2], 'M4 =', M[3])

m1 = M[0] * h + C
m2 = (M[1] - pow(M[0], 2)) * pow(h, 2)
m3 = (M[2] - 3 * M[1] * M[0] + 2 * pow(M[0], 3)) * pow(h, 3)
m4 = (M[3] - 4 * M[2] * M[0] + 6 * M[1] * pow(M[0], 2) - 3 * pow(M[0], 4)) * pow(h, 4)
print('Центральные эмпирические моменты: m1 =', m1, 'm2 =', m2,
      'm3 =', m3, 'm4 =', m4)

y_sample_mean_u = m1; D_u = m2
print('Выборочное среднее, вычисленное с помощью усл. вариант:', y_sample_mean_u)
print('Дисперсия, вычисленная с помощью усл. вариант:', D_u)

```

```

y_sample_mean = 0.0
for i in range(k): y_sample_mean += interval_range_y[i][3] *
interval_range_y[i][2] / N
D = 0.0
for i in range(k): D += interval_range_y[i][3] *
pow(interval_range_y[i][2] - y_sample_mean, 2) / N
print('Выборочное среднее, вычисленное с помощью стандартной
формулы:', y_sample_mean)
print('Дисперсия, вычисленная с помощью стандартной формулы:', D)

s_2 = N * D / (N - 1); sigma = math.sqrt(D); s_y = math.sqrt(s_2)
print('Исправленная оценка дисперсии:', s_2, '\nСтат. оценки СКО:
sigma =', sigma, 's =', s_y)

As = m3 / pow(s_y, 3); E = m4 / pow(s_y, 4) - 3
print('Стат. оценка коэффи. асимметрии:', As, '\nСтат. оценка
экспессса:', E)

M0 = interval_range_y[np.argmax(interval_range_y[:, 3])][2]; m_e =
C
print('Мода:', M0, 'Медиана:', m_e)

# 3.a
interval_range_file = open('./interval_range.csv', 'r')
lines = interval_range_file.readlines()
interval_range_x = np.ndarray((len(lines), 5))
for line in range(len(lines)):
    elements = lines[line].split(',')
    for i in range(interval_range_x.shape[1]):
        interval_range_x[line][i] = float(elements[i])
interval_range_file.close()
x_sample_mean = 17.979884799999997
s_x = 8.079444486616898

corr_table = np.zeros((interval_range_y.shape[0],
interval_range_x.shape[0]))
for i in range(interval_range_y.shape[0]):
    for j in range(interval_range_x.shape[0]):
        for sample_element in range(N):
            if (interval_range_x[j][0] <=
stat_range[sample_element][0] <= interval_range_x[j][1]) \
                and (interval_range_y[i][0] <=
stat_range[sample_element][1] <= interval_range_y[i][1]):
                corr_table[i][j] += 1
save_current_sample(corr_table, './corr_table.csv')

```

```

# 3.b
r_xy = 0.0
for i in range(interval_range_y.shape[0]):
    for j in range(interval_range_x.shape[0]):
        r_xy += (corr_table[i][j] * interval_range_y[i][2] *
interval_range_x[j][2])
r_xy -= (N * x_sample_mean * y_sample_mean)
r_xy /= (N * s_x * s_y)
print('r_ =', r_xy, '\n', r_xy - 3 * ((1 - pow(r_xy, 2)) /
(math.sqrt(N))), '<= r <=', r_xy + 3 * ((1 + pow(r_xy, 2)) /
(math.sqrt(N)) )

# 3.c
z_ = 1.1513 * np.log10((1 + r_xy) / (1 - r_xy))
sigma_z = 1 / math.sqrt(N - 3)
print('z_ =', z_, 'sigma_z =', sigma_z)
lambda_gamma = 1.96
z_left = z_ - lambda_gamma * sigma_z; z_right = z_ + lambda_gamma
* sigma_z
print('Дов. инт-л для ген. значения: (', z_left, ';', z_right,
')
r_left = (np.exp(2 * z_left) - 1) / (np.exp(2 * z_left) + 1);
r_right = (np.exp(2 * z_right) - 1) / (np.exp(2 * z_right) + 1)
print('Дов. инт-л для ген. значения коэф. корреляции: (',
r_left, ';', r_right, ')')

# 3.d
T_watch = (r_xy * math.sqrt(N - 2)) / math.sqrt(1 - pow(r_xy, 2))
print('T_набл =', T_watch)

# 3.e
sample_file = open('./var_range.csv', 'r')
lines = sample_file.readlines()
sample = np.ndarray((len(lines), 11))
for line in range(len(lines)):
    elements = lines[line].split(',')
    for i in range(sample.shape[1]): sample[line][i] =
float(elements[i])
sample_file.close()

plt.clf()
plt.title('Sample graph')
plt.plot(sample[:, 0], sample[:, 1], 'bo', label='Sample')
plt.xticks(interval_range_x[:, 2], np.around(interval_range_x[:, 2], 2))
plt.yticks(interval_range_y[:, 2], np.around(interval_range_y[:, 2], 2))

```

```

2], 2))
plt.xlabel('Middle of x-interval'); plt.ylabel('Middle of y-
interval')
plt.legend()
plt.grid()
plt.show()

plt.clf()
plt.title('Sample graph')
plt.plot(sample[:, 0], sample[:, 1], 'bo', label='Sample')
plt.plot((sample[:, 1] - y_sample_mean) * r_xy * s_x / s_y +
x_sample_mean, sample[:, 1],
          'g-', label='Root mean square regression line from X to
Y')
plt.plot(sample[:, 0], (sample[:, 0] - x_sample_mean) * r_xy * s_y /
s_x + y_sample_mean,
          'r-', label='Root mean square regression line from Y to
X')
plt.xticks(interval_range_x[:, 2], np.around(interval_range_x[:, 2], 2))
plt.yticks(interval_range_y[:, 2], np.around(interval_range_y[:, 2], 2))
plt.xlabel('Middle of x-interval'); plt.ylabel('Middle of y-
interval')
plt.legend()
plt.grid()
plt.show()

# 3.f
k_x = 7; k_y = 7
x_y = np.zeros(k_y); y_x = np.zeros(k_x)
n_y = np.array([2, 1, 23, 28, 26, 15, 5]); n_x = np.array([8, 9,
17, 16, 25, 15, 10])
for i in range(k_y):
    for j in range(k_x):
        x_y[i] += corr_table[i][j] * interval_range_x[j][2] /
n_y[i]
for i in range(k_x):
    for j in range(k_y):
        y_x[i] += corr_table[j][i] * interval_range_y[j][2] /
n_x[i]
print('x_y =', x_y, '\ny_x =', y_x)

Dxy = np.zeros(k_y); Dyx = np.zeros(k_x)
for i in range(k_y):
    for j in range(k_x):

```

```

        Dxy[i] += corr_table[i][j] * pow(interval_range_x[j][2] -
x_y[i], 2) / n_y[i]
    for i in range(k_x):
        for j in range(k_y):
            Dyx[i] += corr_table[j][i] * pow(interval_range_y[j][2] -
y_x[i], 2) / n_x[i]
    print('D_xy =', Dxy, '\nD_yx =', Dyx)

D_in_gr_xy = 0.0; D_in_gr_yx = 0.0
for i in range(k_y): D_in_gr_xy += Dxy[i] * n_y[i] / N
for i in range(k_x): D_in_gr_yx += Dyx[i] * n_x[i] / N
print('D_in_gr_xy =', D_in_gr_xy, '\nD_in_gr_yx =', D_in_gr_yx)

D_between_gr_xy = 0.0; D_between_gr_yx = 0.0
for i in range(k_y): D_between_gr_xy += n_y[i] * pow(x_y[i] -
x_sample_mean, 2) / N
for i in range(k_x): D_between_gr_yx += n_x[i] * pow(y_x[i] -
y_sample_mean, 2) / N
print('D_between_gr_xy =', D_between_gr_xy, '\nD_between_gr_yx =',
D_between_gr_yx)

D_general_xy = D_in_gr_xy + D_between_gr_xy; D_general_yx =
D_in_gr_yx + D_between_gr_yx
print('D_general_xy =', D_general_xy, '\nD_general_yx =',
D_general_yx)

sigma_xy = math.sqrt(D_between_gr_xy); sigma_x =
math.sqrt(D_general_xy); eta_xy = sigma_xy / sigma_x
sigma_yx = math.sqrt(D_between_gr_yx); sigma_y =
math.sqrt(D_general_yx); eta_yx = sigma_yx / sigma_y
print('eta_xy =', eta_xy, '\neta_yx =', eta_yx)

# 3.g
matrix = np.zeros((3, 4))
for i in range(k_x):
    matrix[0][0] += n_x[i] * pow(interval_range_x[i][2], 4)
    matrix[0][1] += n_x[i] * pow(interval_range_x[i][2], 3)
    matrix[0][2] += n_x[i] * pow(interval_range_x[i][2], 2)
    matrix[1][2] += n_x[i] * interval_range_x[i][2]
    matrix[0][3] += n_x[i] * pow(interval_range_x[i][2], 2) *
y_x[i]
    matrix[1][3] += n_x[i] * interval_range_x[i][2] * y_x[i]
    matrix[2][3] += n_x[i] * y_x[i]
matrix[1][0] = matrix[0][1]; matrix[1][1] = matrix[0][2];
matrix[2][0] = matrix[0][2]; matrix[2][1] = matrix[1][2];
matrix[2][2] = N

```

```

print(matrix)

determinant = np.linalg.det(matrix[:, 0:3])
a = np.linalg.det(matrix[:, [3, 1, 2]]) / determinant
b = np.linalg.det(matrix[:, [0, 3, 2]]) / determinant
c = np.linalg.det(matrix[:, [0, 1, 3]]) / determinant
print(a, b, c)
paraboloid_curve = a * sample[:, 0] * sample[:, 0] + b * sample[:, 0] + c

matrix = np.zeros((2, 3))
for i in range(k_x):
    matrix[0][0] += n_x[i] * pow(interval_range_x[i][2], 2)
    matrix[0][1] += n_x[i] * interval_range_x[i][2]
    matrix[0][2] += n_x[i] * interval_range_x[i][2] / y_x[i]
    matrix[1][2] += n_x[i] / y_x[i]
matrix[1][0] = matrix[0][1]; matrix[1][1] = N
print(matrix)

b = (matrix[0][2] * matrix[1][0] - matrix[0][0] * matrix[1][2]) /
(matrix[0][1] * matrix[1][0] - matrix[1][1] * matrix[0][0])
a = (matrix[1][2] - b * matrix[1][1]) / matrix[1][0]
print(a, b)

plt.clf()
plt.title('Sample graph')
plt.plot(sample[:, 0], sample[:, 1], 'bo', label='Sample')
plt.plot(sample[:, 0], paraboloid_curve, 'r-', label='Paraboloid correlation curve')
plt.plot(sample[:, 0], 1 / (a * sample[:, 0] + b), 'g-', label='Custom correlation curve')
plt.xticks(interval_range_x[:, 2], np.around(interval_range_x[:, 2], 2))
plt.yticks(interval_range_y[:, 2], np.around(interval_range_y[:, 2], 2))
plt.xlabel('Middle of x-interval'); plt.ylabel('Middle of y-interval')
plt.legend()
plt.grid()
plt.show()

```

## ПРИЛОЖЕНИЕ В

### ИСХОДНЫЙ КОД ПРОГРАММЫ ДЛЯ КЛАСТЕРНОГО АНАЛИЗА

```
import matplotlib.pyplot as plt
import numpy as np
import random, math

def distance(a, b, m=2):
    s = 0
    for k_m in range(m): s += pow(a[k_m] - b[k_m], 2)
    return math.sqrt(s)

def change_centroids(groups, centroids, k):
    for j in range(k):
        s = [0, 0]; count = 0
        for n in range(N):
            if groups[j][n] == -1: break
            s[0] += sample[groups[j][n]][0]; s[1] += sample[groups[j][n]][1]; count += 1
        if count != 0: centroids[j] = [s[0] / count, s[1] / count]

def change_centroid(group_array):
    s = [0, 0]; count = 0
    for n in group_array:
        if n == -1: break
        s[0] += sample[n][0]; s[1] += sample[n][1]
        count += 1
    return s[0] / count, s[1] / count

def groups_equal(arr1, arr2):
    for i in range(arr1.shape[0]):
        for j in range(arr1.shape[1]):
            if arr1[i][j] != arr2[i][j]: return False
    return True

def k_means(k, freq_centroids_changing):
    print('k =', k, 'freq_centroids_changing =',
freq_centroids_changing)
    centroids = np.ndarray((k, 2))
    groups = np.full((k, N), -1)
```

```

old_groups = np.full((k, N), -1)
first_stage = True
for i in range(k):
    index = random.randint(0, N - 1)
    while index in groups: index = random.randint(0, N - 1)
    groups[i][0] = index
    centroids[i] = sample[index]

while first_stage or not groups_equal(old_groups, groups):
    if first_stage: first_stage = False
    else:
        old_groups = groups.copy()
        groups = np.full((k, N), -1)
        for i in range(N):
            if i not in groups:
                min_distance = None; group_number = -1
                for j in range(k):
                    if group_number == -1 or distance(sample[i],
centroids[j]) < min_distance:
                        min_distance = distance(sample[i],
centroids[j])
                        group_number = j
                for j in range(N):
                    if groups[group_number][j] == -1:
                        groups[group_number][j] = i
                        break
                if freq_centroids_changing:
change_centroids(groups, centroids, k)
            if not freq_centroids_changing: change_centroids(groups,
centroids, k)

F1 = 0; F2 = 0; F3 = 0
for i in range(k):
    cluster_n = 0
    for j in range(N):
        if groups[i][j] == -1: break
        cluster_n += sample_freq[groups[i][j]][0]
        F1 += pow(distance(sample[groups[i][j]],
centroids[i]), 2)
        F3 += pow(np.sum((sample[groups[i][j]] -
centroids[i]) * (sample[groups[i][j]] - centroids[i]) *
sample_freq[groups[i][j]][0])) /
cluster_n, 2)
        for jj in range(N):
            if j == jj: continue
            if groups[i][jj] == -1: break

```

```

        F2 += pow(distance(sample[groups[i][j]], 
sample[groups[i][jj]]), 2)
        print('F1 =', F1, 'F2 =', F2, 'F3 =', F3)

    styles = ['b', 'g', 'y', 'r', 'm', 'k', 'c']
    plt.clf()
    plt.title('Grouped sample graphic')
    for i in range(k):
        plt.plot([sample[j][0] for j in groups[i][:] if j != -1],
[sample[j][1] for j in groups[i][:] if j != -1],
            styles[i] + '+', label=str(i + 1) + ' group')
        plt.plot(centroids[i][0], centroids[i][1], styles[i] +
'o', label=str(i + 1) + ' group centroid')
        plt.xticks((interval_range_x[:, 2] - x_sample_mean) / s_x,
np.around((interval_range_x[:, 2] - x_sample_mean) / s_x, 2))
        plt.yticks((interval_range_y[:, 2] - y_sample_mean) / s_y,
np.around((interval_range_y[:, 2] - y_sample_mean) / s_y, 2))
        plt.xlabel('Middle of x-interval (normalized)');
    plt.ylabel('Middle of y-interval (normalized)')
    plt.legend()
    plt.grid()
    plt.show()

def concentration_searching(radius):
    groups = np.full((N, N), -1)
    centroids = np.ndarray((N, 2))
    k = 0

    while True:
        start_center = random.randint(0, N - 1); local_radius =
radius
        while start_center in groups: start_center =
random.randint(0, N - 1)

        while True:
            centroids[k][0] = sample[start_center][0];
            centroids[k][1] = sample[start_center][1]
            first_stage = True
            old_group = groups[k].copy()
            while first_stage or not groups_equal(old_group,
groups[k]):
                if not first_stage:
                    old_group = groups[k].copy()
                    groups[k] = np.full(N, -1)
                else: first_stage = False

```

```

        for i in range(N):
            if distance(sample[i], centroids[k]) <=
local_radius:
            for j in range(N):
                if groups[k][j] == -1:
                    groups[k][j] = i
                    break
            centroids[k][0], centroids[k][1] =
change_centroid(groups[k])

            crossing = False
            for i in range(N):
                if groups[k][i] != -1 and ((k > 1 and groups[k][i]
in groups[0:k - 1, :]) or (k == 1 and groups[k][i] in groups[0])):
                    if local_radius > delta: local_radius -= delta
                    groups[k] = np.full(N, -1)
                    crossing = True
                    break
            if not crossing: break

            k += 1
            complete = True
            for i in range(N):
                if i not in groups: complete = False
            if complete: break

F1 = 0; F2 = 0; F3 = 0
for i in range(k):
    cluster_n = 0
    for j in range(N):
        if groups[i][j] == -1: break
        cluster_n += sample_freq[groups[i][j]][0]
        F1 += pow(distance(sample[groups[i][j]],
centroids[i]), 2)
        F3 += pow(np.sum((sample[groups[i][j]] - centroids[i])
* (sample[groups[i][j]] - centroids[i]) *
sample_freq[groups[i][j]][0]) /
cluster_n, 2)
        for jj in range(N):
            if j == jj: continue
            if groups[i][jj] == -1: break
            F2 += pow(distance(sample[groups[i][j]],
sample[groups[i][jj]]), 2)
    print('F1 =', F1, 'F2 =', F2, 'F3 =', F3)

styles = ['b', 'g', 'y', 'r', 'm', 'k', 'c']

```

```

plt.clf()
plt.title('Grouped sample graphic')
for i in range(k):
    plt.plot([sample[j][0] for j in groups[i][:] if j != -1],
[sample[j][1] for j in groups[i][:] if j != -1],
            styles[i % len(styles)] + '+', label=str(i + 1) +
' group')
    plt.plot(centroids[i][0], centroids[i][1], styles[i % len(styles)] + 'o',
label=str(i + 1) + ' group centroid')
    plt.xticks((interval_range_x[:, 2] - x_sample_mean) / s_x,
np.around((interval_range_x[:, 2] - x_sample_mean) / s_x, 2))
    plt.yticks((interval_range_y[:, 2] - y_sample_mean) / s_y,
np.around((interval_range_y[:, 2] - y_sample_mean) / s_y, 2))
    plt.xlabel('Middle of x-interval (normalized)');
plt.ylabel('Middle of y-interval (normalized)')
    plt.legend()
    plt.grid()
    plt.show()

```

```

# 1
N = 100
k_ = int(math.sqrt(N / 2))
x_sample_mean = 17.979884799999997
s_x = 8.079444486616898
y_sample_mean = 1013.9642857142858
s_y = 7.355282831323163

sample_file = open('./var_range.csv', 'r')
lines = sample_file.readlines()
sample = np.ndarray((len(lines), 2))
for line in range(len(lines)):
    elements = lines[line].split(',')
    for i in range(2): sample[line][i] = float(elements[i])
sample_file.close()

sample_file = open('./var_range_freq.csv', 'r')
lines = sample_file.readlines()
sample_freq = np.ndarray((len(lines), 2))
for line in range(len(lines)):
    elements = lines[line].split(',')
    for i in range(2): sample_freq[line][i] = float(elements[i])
sample_file.close()

```

```

sample_file = open('./interval_range.csv', 'r')
lines = sample_file.readlines()
interval_range_x = np.ndarray((len(lines), 5))
for line in range(len(lines)):
    elements = lines[line].split(',')
    for i in range(interval_range_x.shape[1]):
interval_range_x[line][i] = float(elements[i])
sample_file.close()

sample_file = open('./interval_range_2.csv', 'r')
lines = sample_file.readlines()
interval_range_y = np.ndarray((len(lines), 5))
for line in range(len(lines)):
    elements = lines[line].split(',')
    for i in range(interval_range_y.shape[1]):
interval_range_y[line][i] = float(elements[i])
sample_file.close()

plt.clf()
plt.title('Sample graphic')
plt.plot(sample[:, 0], sample[:, 1], 'bo', label='Sample')
plt.xticks(interval_range_x[:, 2], np.around(interval_range_x[:, 2], 2))
plt.yticks(interval_range_y[:, 2], np.around(interval_range_y[:, 2], 2))
plt.xlabel('Middle of x-interval'); plt.ylabel('Middle of y-interval')
plt.legend()
plt.grid()
plt.show()

# 2
sample[:, 0] = (sample[:, 0] - x_sample_mean) / s_x; sample[:, 1]
= (sample[:, 1] - y_sample_mean) / s_y

plt.clf()
plt.title('Normalized sample graphic')
plt.plot(sample[:, 0], sample[:, 1], 'bo', label='Normalized sample')
plt.xticks((interval_range_x[:, 2] - x_sample_mean) / s_x,
np.around((interval_range_x[:, 2] - x_sample_mean) / s_x, 2))
plt.yticks((interval_range_y[:, 2] - y_sample_mean) / s_y,
np.around((interval_range_y[:, 2] - y_sample_mean) / s_y, 2))
plt.xlabel('Middle of x-interval (normalized)');
plt.ylabel('Middle of y-interval (normalized)')
plt.legend()

```

```

plt.grid()
plt.show()

# 3 - 5
for i in range(2, k_ + 1):
    k_means(i, True); k_means(i, False)

# 6 - 8
D = np.zeros((N, N))
for i in range(N):
    for j in range(N):
        if i != j: D[i][j] = distance(sample[i], sample[j])
delta = 0.01
concentration_searching(float(input('Enter a 1st cluster sphere
radius (' + str(np.min(D)) + ';' + str(np.max(D)) + '): ')))

```