**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**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 8382 |  | Звегинцева Е.Н. |
| Преподаватель |  | Середа А.-В.И. |

Санкт-Петербург

2022

**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. 8382 |  | Мирончик П.Д. |
| Преподаватель |  | Середа А.-В.И. |

Санкт-Петербург

2022

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |
| --- | --- | --- |
| Студентка Звегинцева Е.Н. | | |
| Группа 8382 | | |
| 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 записки:  «Аннотация», «Содержание», «Введение», «Заключение», «Список использованных источников». | | |
| Предполагаемый объем пояснительной записки:  Не менее 20 страниц. | | |
| Дата выдачи задания: 05.04.2022 | | |
| Дата сдачи работы: 10.04.2022 | | |
| Дата защиты работы: 12.04.2022 | | |
| Студентка |  | Звегинцева Е.Н. |
| Преподаватель |  | Середа А.-В.И. |

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |
| --- | --- | --- |
| Студент Мирончик П.Д. | | |
| Группа 8382 | | |
| 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 записки:  «Аннотация», «Содержание», «Введение», «Заключение», «Список использованных источников». | | |
| Предполагаемый объем пояснительной записки:  Не менее 20 страниц. | | |
| Дата выдачи задания: 05.04.2022 | | |
| Дата сдачи работы: 10.04.2022 | | |
| Дата защиты работы: 12.04.2022 | | |
| Студент |  | Мирончик П.Д. |
| Преподаватель |  | Середа А.-В.И. |

**Аннотация**

В данной курсовой работе проводится статистический анализ двумерной выборки. Исследование включает в себя выравнивание статистических рядов, нахождение точечных и интервальных статистических оценок, построение регрессионных кривых, проверку статистических гипотез о нормальном распределении выборки, о равенстве коэффициента корреляции нулю с помощью критерия Пирсона. Осуществляется кластерный анализ выборки с использованием двух различных методов.

**Summary**

In this course work is a statistical analysis of a two-dimensional sample. The study includes the alignment of statistical series, finding point and interval statistical estimates, the construction of regression curves, testing statistical hypotheses about the normal distribution of the sample, about the equality of the correlation coefficient to zero using Pearson's test. Cluster analysis of the sample using two different methods is carried out.

**содержание**

|  |  |  |
| --- | --- | --- |
|  | Введение | 8 |
| 1. | Выравнивание статистических рядов | 9 |
| 1.1. | Основные теоретические положения | 9 |
| 1.2.  1.3.  1.4.  1.5. | Формирование и первичная обработка выборки. Ранжированный и интервальный ряды  Нахождение точечных оценок параметров распределения  Нахождение интервальных оценок параметров распределения. Проверка статистической гипотезы о нормальном распределении  Выводы | 13  20  23  25 |
| 2. | Корреляционный и регрессионный анализ | 27 |
| 2.1. | Основные теоретические положения | 27 |
| 2.1.1  2.1.2 | Корреляционный анализ  Регрессионный анализ | 27  29 |
| 2.2.  2.3.  2.4. | Элементы корреляционного анализа. Проверка статистической гипотезы о равенстве коэффициента корреляции нулю  Элементы регрессионного анализа. Выборочные прямые. среднеквадратической регрессии. Корреляционные отношения.  Выводы | 31  37  42 |
| 3. | Кластерный анализ | 44 |
| 3.1. | Основные теоретические положения | 44 |
| 3.2.  3.3.  3.4. | Метод k-средних  Метод поиска сгущений  Выводы | 47  56  68 |
|  | Заключение | 69 |
|  | Список использованных источников | 70 |
|  | Приложение А. Программа для формирования и первичной обработки выборки, построения, ранжированного и интервального рядов  Приложение Б. Программа для нахождения точечных оценок параметров распределения  Приложение В. Программа для нахождения интервальных оценок параметров распределения и проверки статистической гипотезы о нормальном распределении  Приложение Г. Программа для нахождения элементов корреляционного анализа и проверки статистической гипотезы о равенстве коэффициента корреляции нулю  Приложение Д. Программа для нахождения элементов регрессионного анализа и построения выборочные прямых среднеквадратической регрессии, поиска корреляционного отношения  Приложение Е. Программа для метода k-cредних  Приложение Ж. Программа для метода поиска сгущений | 71  74  77  79  82  86  89 |

**введение**

В ходе выполнения данной работы необходимо осуществить статистический анализ двумерной выборки. Для проведения вычислений используется язык Python. Необходимо получить практические навыки нахождения точечных статистических оценок параметров распределения случайных величин, а также проверить гипотезу о нормальности распределения этих величин.

Для выявления зависимости между величинами, необходимо осуществить корреляционный и регрессионные анализы. Также необходимо освоить основные понятия статистических гипотез, провести проверку на их справедливость для данной выборки и ознакомиться с методом наименьших квадратов, вычислить критерий Пирсона, рассчитать корреляционные отношения и построить корреляционные прямые, а также прямые среднеквадратической регрессии.

Необходимо реализовать метод k-means и метод поиска сгущений для осуществления кластерного анализа, используемого для более глубокого анализа данных.

**1. выравнивание статистических рядов**

**1.1. Основные теоретические положения**

Ранжированный ряд– это распределение отдельных единиц совокупности в порядке возрастания или убывания исследуемого признака. Ранжирование позволяет легко разделить количественные данные по группам, сразу обнаружить наименьшее и наибольшее значения признака, выделить значения, которые чаще всего повторяются.

Вариационный ряд– ранжированный ряд, в котором объединены одинаковые элементы; можно представить как таблицу, включающую в себя уникальные значения выборки  и количество таких значений в выборке. Элементы вариационного ряда называются вариантами.

Интервальный ряд распределения – это таблица, состоящая из двух столбцов (строк) – интервалов варьирующего признака и числа единиц совокупности, попадающих в данный интервал (частот - ), или долей этого числа в общей численности совокупностей (частостей - ).

Полигоном частот называют ломанную, отрезки которой соединяют точки . Для построения полигона частот на оси абсцисс откладывают варианты , а на оси ординат – соответствующие им частоты . Точки соединяют отрезками прямых и получают полигон частот.

Гистограммой частот (частостей) называется ступенчатая фигура, состоящая из прямоугольников с основаниями, равными интервалам значений и высотами, равными отношению частот (или частостей) к шагу:

Эмпирической функцией распределениявеличины называют функцию,

, где

Xi – i-ый элемент выборки. Т.е. значение функции в точке x равно числу элементов выборки, деленному на объем выборки. В данной работе эмпирическая функция строится для интервального ряда абсолютных и относительных частот, поэтому вместо отдельных элементов выборки в формулу подставляются сформированные интервалы, а в случае относительных частот не осуществляется деление на N.

Математическим ожиданием дискретной случайной величины называется сумма произведений ее возможных значений на соответствующие им вероятности:

Дисперсией случайной величины называется математическое ожидание квадрата ее отклонения от ее математического ожидания:

Среднеквадратическим отклонением случайной величины Х (стандартом) называется квадратный корень из ее дисперсии:

Асимметрией, или коэффициентом асимметрии, называется числовая характеристика, определяемая выражением:

где – центральный эмпирический момент третьего порядка,  *–* исправленнаявыборочная дисперсия.

Центральным моментом порядка  случайной величины *X* называется математическое ожидание величины:

Исправленная выборочная дисперсияопределяется по формуле:

где выборочная дисперсия.

Эксцессом называется численная характеристика случайной величины, которая определяется выражением:

Для нормального закона . Отсюда следует, что для нормального закона. Эксцесс показывает, как быстро уменьшается плотность распределения вблизи её максимального значения.

Мода дискретной случайной величины – это наиболее вероятное значение этой случайной величины. Модой непрерывной случайной величины называется ее значение, при котором плотность вероятности максимальна.

Медиана случайной величины *X* – это такое ее значение , для которого выполнено равенство

Доверительным называют интервал, который с заданной надежностью покрывает заданный параметр.

Интервальной оценкой математического ожидания по выборочной среднем при неизвестном среднем квадратическом отклонении генеральной совокупности служит доверительный интервал:

,

где

– статистическая оценка математического ожидания;

– исправленная выборочная дисперсия;

– объём выборки;

– из таблицы.

Интервальной оценкой среднеквадратического отклонения по исправленной выборочной дисперсии служит доверительный интервал:

где

– исправленная выборочная дисперсия;

­– из таблицы.

Критерий Пирсона, или критерий (Хи-квадрат), применяют для проверки гипотезы о соответствии эмпирического распределения предполагаемому теоретическому распределению .

Метод позволяет оценить статистическую значимость различий двух или нескольких относительных показателей (частот, долей).

Теоретические частоты вычисляются по формуле:

где

.

Следует привести теоретические частоты к функции Лапласа.

Если , то примет следующий вид:

Для данной задачи . Преобразуя формулу получим:

,

где – функция ошибок.

Если - гипотеза принимается, иначе () – гипотезу отвергают.

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

Генеральная совокупность состоит из данных наблюдений относительно объемного веса при влажности и модуля упругости при сжатии вдоль волокон древесины резонансной ели, представленных преподавателем. Из нее была сгенерирована выборка из 114 наблюдений.

Полученная исходная выборка представлена в табл. 1.2.1

Таблица 1.2.1 – Исходная выборка

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 405 | 107.5 | 24 | 418 | 125.7 | 47 | 528 | 163.4 | 70 | 476 | 143.0 | 93 | 391 | 98.2 |
| 2 | 521 | 154.9 | 25 | 477 | 139.7 | 48 | 462 | 135.7 | 71 | 464 | 143.2 | 94 | 482 | 150.1 |
| 3 | 382 | 98.1 | 26 | 468 | 142.0 | 49 | 453 | 119.5 | 72 | 449 | 124.5 | 95 | 453 | 126.4 |
| 4 | 362 | 111.7 | 27 | 545 | 145.3 | 50 | 393 | 103.2 | 73 | 331 | 84.6 | 96 | 465 | 140.9 |
| 5 | 444 | 130.0 | 28 | 441 | 140.8 | 51 | 487 | 146.0 | 74 | 444 | 121.4 | 97 | 458 | 121.7 |
| 6 | 465 | 114.8 | 29 | 543 | 155.4 | 52 | 430 | 95.3 | 75 | 423 | 104.1 | 98 | 482 | 139.9 |
| 7 | 513 | 163.6 | 30 | 474 | 132.5 | 53 | 371 | 91.9 | 76 | 446 | 130.3 | 99 | 448 | 125.6 |
| 8 | 362 | 84.3 | 31 | 321 | 86.1 | 54 | 441 | 126.1 | 77 | 451 | 128.6 | 100 | 411 | 112.9 |
| 9 | 550 | 147.9 | 32 | 438 | 120.7 | 55 | 495 | 150.9 | 78 | 434 | 140.4 | 101 | 452 | 119.7 |
| 10 | 453 | 138.2 | 33 | 409 | 121.0 | 56 | 510 | 129.4 | 79 | 469 | 131.5 | 102 | 504 | 143.8 |
| 11 | 482 | 141.2 | 34 | 451 | 124.3 | 57 | 496 | 141.7 | 80 | 452 | 131.0 | 103 | 454 | 131.1 |
| 12 | 431 | 125.0 | 35 | 462 | 125.2 | 58 | 525 | 162.1 | 81 | 397 | 108.6 | 104 | 422 | 122.9 |
| 13 | 437 | 129.2 | 36 | 541 | 162.3 | 59 | 377 | 96.0 | 82 | 470 | 124.0 | 105 | 443 | 137.4 |
| 14 | 352 | 87.7 | 37 | 460 | 140.7 | 60 | 512 | 169.9 | 83 | 351 | 102.9 | 106 | 510 | 153.9 |
| 15 | 351 | 89.0 | 38 | 358 | 98.3 | 61 | 382 | 113.9 | 84 | 430 | 104.3 | 107 | 448 | 125.0 |
| 16 | 498 | 164.0 | 39 | 353 | 98.0 | 62 | 480 | 153.9 | 85 | 541 | 146.8 | 108 | 442 | 123.4 |
| 17 | 510 | 162.3 | 40 | 508 | 159.0 | 63 | 391 | 107.5 | 86 | 467 | 140.5 | 109 | 423 | 115.9 |
| 18 | 547 | 154.7 | 41 | 379 | 94.6 | 64 | 448 | 137.3 | 87 | 394 | 112.1 | 110 | 504 | 145.3 |
| 19 | 477 | 146.0 | 42 | 458 | 128.0 | 65 | 540 | 156.7 | 88 | 406 | 110.1 | 111 | 471 | 143.9 |
| 20 | 428 | 131.6 | 43 | 548 | 162.3 | 66 | 421 | 126.9 | 89 | 477 | 135.8 | 112 | 493 | 154.5 |
| 21 | 458 | 124.4 | 44 | 447 | 117.5 | 67 | 453 | 124.2 | 90 | 424 | 119.0 | 113 | 544 | 166.7 |
| 22 | 470 | 146.7 | 45 | 446 | 128.4 | 68 | 441 | 122.8 | 91 | 475 | 143.6 | 114 | 460 | 122.4 |
| 23 | 499 | 144.5 | 46 | 483 | 130.3 | 69 | 452 | 116.1 | 92 | 352 | 84.9 |  |  |  |

Далее выборка была сокращена до величины и была упорядочена по возрастанию для получения ранжированного ряда. Результат представлен на рис. 1.2.1

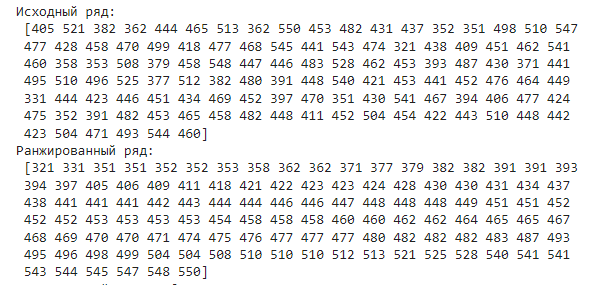


Рисунок 1.2.1 –выполнение формирования ранжированного ряда из исходного программой.

Отсюда можно сделать вывод, что наименьшее значение в выборке , а наибольшее значение .

Преобразование полученной выборки в вариационный ряд с абсолютными частотами, путем подсчета уникальных значений для нашей величины, представлено на рис.1.2.2. Также были рассчитаны относительные частоты, как доля значений от общего числа наблюдений, что можно наблюдать на рис. 1.2.3

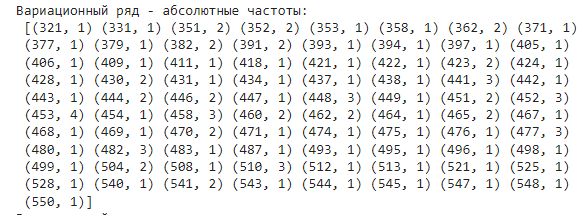
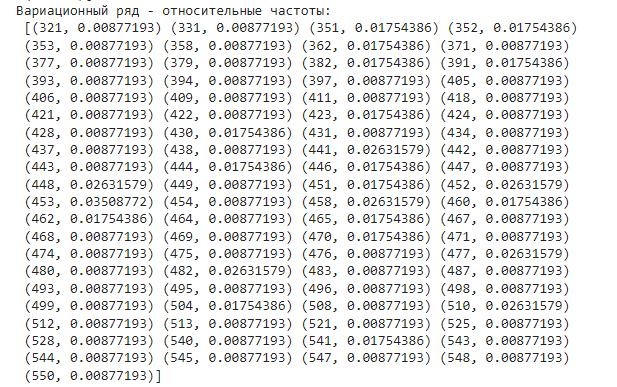


Рисунок 1.2.2 – Вариационный ряд с абсолютными частотами

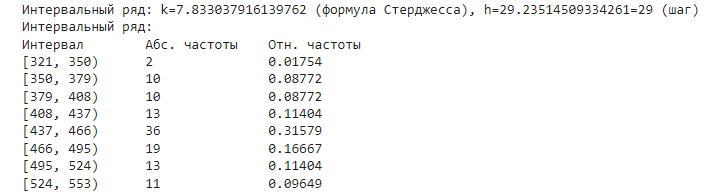
 Рисунок 1.2.3 – Вариационный ряд с относительными частотами

Наиболее часто встречающаяся величина -

Далее был сформирован интервальный ряд. Были рассчитаны коэффициент k по формуле Стерджесса для определения количества интервалов и ширина интервала h.

где – объем выборки

Аналогично предыдущему пункту для каждого интервала были вычислены абсолютные и относительные частоты. Полученные значения можно увидеть на рис. 1.2.4

Рисунок 1.2.4 – Интервальный ряд, вычисленный программой

Для проверки достоверности вычислений, можно сложить все абсолютные частоты, получив 114, что соответствует объему выборки, а при сложении относительных частот получим 1. Наибольшее количество наблюдений мы наблюдаем в интервале [437, 466)

Строим полигоны, представленные на рис. 1.2.5 и 1.2.6, применительно к интервальному ряду для абсолютных и относительных частот раздельно.

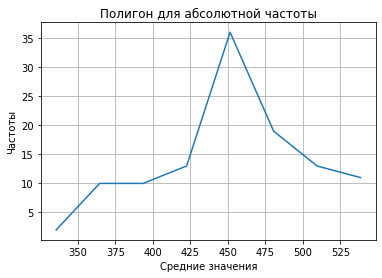


Рисунок 1.2.5 – Полигон для абсолютной частоты

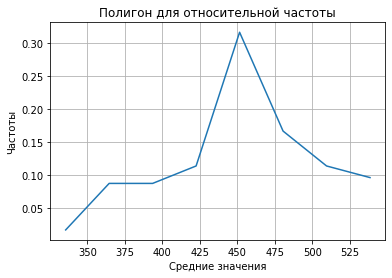


Рисунок 1.2.6 – Полигон для относительной частоты

Гистограммы, построенные применительно к интервальному ряду для абсолютных и относительных частот представлены на рис. 1.2.7 и 1.2.8.

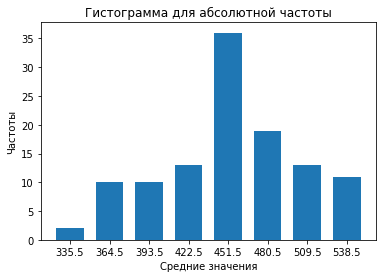


Рисунок 1.2.7 – Гистограмма для абсолютной частоты

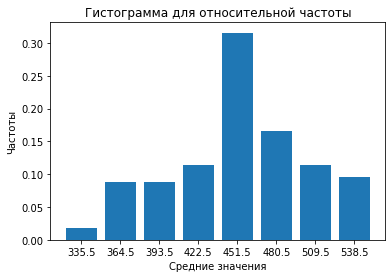


Рисунок 1.2.8 – Гистограмма для относительной частоты

Как мы можем заметить, и гистограммы, и полигоны, для абсолютных и относительных частот совпадают по форме, в связи с тем, что при преобразовании, отношения величин сохраняются.

Из данных графиков можно предположить, что выборка сформирована под действием нормального закона.

Далее строим эмпирические функции распределения применительно к интервальному ряду для абсолютных и относительных частот, представленные на рис. 1.2.9 и 1.2.10.

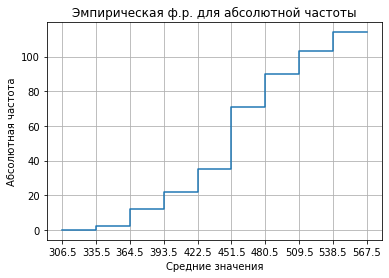


Рисунок 1.2.9 – График эмпирической функции распределения для абсолютной частоты

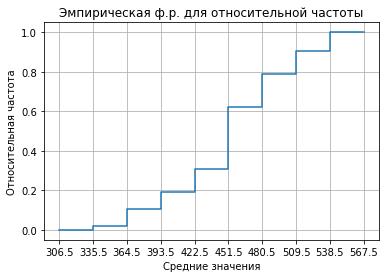


Рисунок 1.2.10 – График эмпирической функции распределения для относительной частоты

**1.3. Нахождение точечных оценок параметров распределения.**

Вычислим среднее значение для каждого интервала и с помощью абсолютной частоты найдем условные варианты по формуле:

где ширина интервала, – середина интервала с наибольшей частотой.

Для нахождения условных моментов нужно вычислить величины Результаты вычислений представлены в табл. 1.3.1.

Таблица 1.3.1 – Результат вычислений

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 335.5 | 2 | -4 | -8.0 | 32.0 | -128.0 | 512.0 |
| 364.5 | 10 | -3 | -30.0 | 90.0 | -270.0 | 810.0 |
| 393.5 | 10 | -2 | -20.0 | 40.0 | -80.0 | 160.0 |
| 422.5 | 13 | -1 | -13.0 | 13.0 | -13.0 | 13.0 |
| 451.5 | 36 | 0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 480.5 | 19 | 1 | 19.0 | 19.0 | 19.0 | 19.0 |
| 509.5 | 13 | 2 | 26.0 | 52.0 | 104.0 | 208.0 |
| 538.5 | 11 | 3 | 33.0 | 99.0 | 297.0 | 891.0 |

Также вычислим величину для каждого интервала (табл.1.3.2) для проведения проверки.

Таблица 1.3.2- Величина для интервалов

|  |
| --- |
|  |
| 162,0 |
| 160,0 |
| 10,0 |
| 0,0 |
| 36,0 |
| 304,0 |
| 1053,0 |
| 2816,0 |

4541

Как мы видим, значения совпали, значит все вычислено верно.

Далее мы находим условные моменты, по формуле:

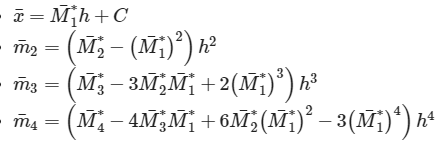
Условный эмпирический момент 1 порядка: 0.06140350877192982

Условный эмпирический момент 2 порядка: 3.026315789473684

Условный эмпирический момент 3 порядка: -0.6228070175438597

Условный эмпирический момент 4 порядка: 22.92105263157895

Используя вычисленные условные моменты, вычислим центральные эмпирические моменты и начальный эмпирический момент 1го порядка(выборочное среднее), по формулам:



Начальный эмпирический момент 1го порядка: 453.280701754386

Центральный эмпирический момент 2го порядка: 2541.960680209295

Центральный эмпирический момент 3го порядка: -28774.708304309555

Центральный эмпирический момент 4го порядка: 16368209.866476053

Вычислим выборочное среднее и дисперсию, используя стандартные формулы:

Результаты совпали с вычисленными с помощью условных моментов, следовательно формулы с условными моментами использованы корректно.

Найдем исправленную выборочную дисперсию и несмещенную оценку стандартного отклонения:

Далее вычислим асимметрию:

Вычислим эксцесс:

Можно увидеть, что основной центр тяжести графика немного смещен вправо, в связи с маленьким по модулю коэффициентом асимметрии. По эксцессу можно сделать вывод, что пик распределения случайной величины не такой острый, как в случае нормального распределения. Это можно наблюдать на рис.1.3.1

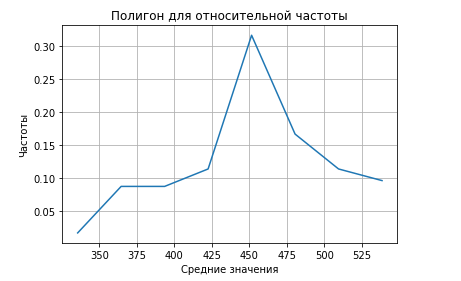


Рисунок 1.3.1 – полигон для относительной частоты из п.1.2

Далее найдем моду вариационного ряда по формуле:

Далее найдем медиану вариационного ряда по формуле:

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

Определим доверительный интервал для мат. ожидания по формуле:

, где

– статистическая оценка математического ожидания, – исправленная выборочная дисперсия, – объём выборки;

Для доверительной точности 0.95 и объема выборки n=114 было выбрано значение 1.980 по приложению 6.

При  γ=0.95: (443.8897332769508, 462.67167023182117)

Для доверительной точности 0.99 и объема выборки n=114 было выбрано значение t=2.617 по приложению 6.

При  γ=0.99: (440.8684974587052, 465.69290605006677)

Определим доверительный интервал для среднеквадратического отклонения по формуле:

, где

– исправленная выборочная дисперсия

Для доверительной точности 0.95 и объема выборки n=114 было выбрано значение q=0.143 по приложению 7

При  γ=0.95: (43.39887183768498, 57.88204260265337)

Для доверительной точности 0.99 и объема выборки n=114 было выбрано значение q=0.198 по приложению 7

При  γ=0.99: (40.61364669057568, 60.66726774976267)

По результатам видно, что большая доверительная вероятность обеспечивается за счет увеличения доверительного интервала, что делает оценку менее точной в обоих случаях (при вычислении доверительного интервала и для мат.ожидания, и для СКО).

Проверим гипотезу о нормальном распределении исследуемой случайной величины с помощью критерия Пирсона

Формулируем нулевую гипотезу: величина ν имеет нормальное распределение с математическим ожиданием  и среднеквадратическим отклонением *S*;

Вычислим теоретические вероятности и частоты попадания в каждый интервал. Результаты представлены в табл.1.4.1.

Таблица 1.4.1 – Вычисления теоретических вероятностей и частоты попадания

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| 321 | 350 | 2 | -2,61215 | -2,03949 | -0,4955 | -0,4793 | 0,016202 | 1,847017 |
| 350 | 379 | 10 | -2,03949 | -1,46683 | -0,4793 | -0,42879 | 0,050511 | 5,758279 |
| 379 | 408 | 10 | -1,46683 | -0,89416 | -0,42879 | -0,31438 | 0,114406 | 13,04231 |
| 408 | 437 | 13 | -0,89416 | -0,3215 | -0,31438 | -0,12608 | 0,188299 | 21,46612 |
| 437 | 466 | 36 | -0,3215 | 0,251169 | -0,12608 | 0,099158 | 0,225241 | 25,67746 |
| 466 | 495 | 19 | 0,251169 | 0,823833 | 0,099158 | 0,294983 | 0,195825 | 22,32402 |
| 495 | 524 | 13 | 0,823833 | 1,396498 | 0,294983 | 0,418718 | 0,123735 | 14,10577 |
| 524 | 553 | 11 | 1,396498 | 1,969163 | 0,418718 | 0,475533 | 0,056815 | 6,476919 |

С использованием полученных частот вычислим по формуле:

Вычисленные результаты представлены в табл. 1.4.2

Таблица 1.4.2 - Вычисления

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 2 | 1,847017 | 0,012671199 |
| 10 | 5,758279 | 3,124578747 |
| 10 | 13,04231 | 0,709661414 |
| 13 | 21,46612 | 3,338991969 |
| 36 | 25,67746 | 4,149742667 |
| 19 | 22,32402 | 0,49494192 |
| 13 | 14,10577 | 0,08668296 |
| 11 | 6,476919 | 3,158640428 |
|  |  | 15,0759113 |

Сравним полученные значения с табличным значением .

Нулевая гипотеза принимается в случае, когда

В нашем случае мы отвергаем нулевую гипотезу, так как , т.е.. Следовательно величина ν не имеет нормальное распределение с математическим ожиданием   и среднеквадратическим отклонением *S*.

**1.5. Выводы.**

Из двумерной генеральной совокупности была сформирована выборка, которая была приведена к ранжированному, вариационному и интервальному видам. Используя полученный интервальный ряд построен полигон, гистограмма и эмпирическая функция распределения для абсолютных и относительных частот.

По гистограмме относительных частот было сделано предположение о формировании выборки под действием нормального закона.

Были выполнены вычисления точечных оценок мат.ожидания, дисперсии, коэффициентов асимметрии и эксцесса методом моментов. Корректность используемого метода моментов была проверена с помощью сравнения значений, полученных с помощью условных вариант и стандартных формул. Были вычислены мода и медиана распределения. Также были сделаны выводы о небольшом сдвиге в правую сторону из-за отрицательной величины асимметрии, а также более пологом пике из-за отрицательной величины эксцесса.

Также были получены границы доверительных интервалов для математического ожидания и СКО случайной величины с доверительными вероятностями 0,95 и 0,99. Из полученных результатов можно сделать вывод, что размер доверительного интервала увеличивается при увеличении доверительной вероятности.

Также была проверена гипотеза о нормальном распределении исследуемой случайной величины с помощью критерия Пирсона . Был сделан вывод, что нулевая гипотеза отвергается, т.к. , следовательно, исследуемая случайная величина не принадлежит нормальному закону распределения.

**2. корреляционный и регрессионный анализ**

**2.1. Основные теоретические положения.**

**2.1.1 Корреляционный анализ.**

Рассмотрим систему двух случайных величин . Эти случайные величины могут быть независимыми, если плотность их совместного распределения равна произведению плотностей распределения каждой из величин:

Иначе между величинами существует *статистическая зависимость*, частным случаем которой является *корреляционная зависимость*. Корреляционной называют статистическую зависимость, при которой изменение значения одной величины приводит к изменению математического ожидания другой величины. Функции регрессии:

Корреляционный момент:

Линейный коэффициент корреляции:

Для коэффициента корреляции справедливо соотношение:

Если =0, величины X и Y некоррелированы. Иначе они коррелированы. Из того, что X и Y коррелированы следует, что они зависимы, но не наоборот: зависимые величины могут быть как коррелированными, так и некоррелированными. При этом независимые величины всегда некоррелированы.

Выборочный коэффициент корреляции можно вычислить по формуле:

С помощью преобразования Фишера перейдём к случайной величине :

Распределение при неограниченном возрастании объёма выборки асимптотически нормальное со значением СКО:

В результате доверительный интервал для генеральной совокупности с доверительной вероятностью определяется по следующей схеме:

1. По формуле (1) вычисляется выборочное значение .
2. По формуле (2) вычисляется значение .
3. Интервал для генерального значения представляется в виде:

где значение должно удовлетворять условию:

1. Для пересчёта интервала в доверительных интервал для коэффициента корреляции с тем же значением необходимо воспользоваться обратным преобразованием Фишера:

Так как  – случайная величина, то, что r¯xy≠0, не означает, что коэффициент корреляции rxy для генеральной совокупности также не равен нулю. Необходимо выполнить проверку статистической гипотезы. Нулевая и альтернативная гипотезы формулируются следующим образом:

В качестве критерия проверки статистической гипотезы о значимости выборочного коэффициента корреляции можно принять случайную величину:

имеющую распределение по закону Стьюдента с степенями свободы. Критическая область для данного критерия двусторонняя.

Проверка гипотезы осуществляется по стандартной схеме:

1. По формуле (3) вычисляется значение .
2. По заданному уровню значимости и значению из таблицы определяется значение .
3. Если – нет оснований отвергать гипотезу .

Если – основная гипотеза с выборочными данными должна быть отвергнута.

**2.1.2 Регрессионный анализ.**

Метод наименьших квадратов (МНК) — метод, основанный на поиске минимума суммы квадратов отклонений значений некоторых функций от заданного множества значений. МНК является одним из основных методов регрессионного анализа и применяется для оценки параметров регрессионных моделей на основе выборочных данных.



Т.е. необходимо найти такие значения переменных, чтобы значения функций были близки к .

Регрессионный анализ – это статистический метод исследования влияния одной или нескольких независимых переменных на зависимую переменную .

Линейные функции выборочной среднеквадратической регрессии:

где - выборочное среднее , соответственно, – выборочный коэффициент корреляции, и – статистические оценки среднеквадратических отклонений для выборок , .

Для оценки корреляционной зависимости между случайными величинами в общем, а не только линейной, может быть использовано так называемое корреляционной отношение.

Чтобы рассчитать выборочное корреляционное отношение нужно рассчитать внутригрупповую и межгрупповую дисперсии. Оценку общей дисперсии можно представить, как сумму:

Чтобы рассчитать выборочное корреляционное отношение к нужно рассчитать внутригрупповую и межгрупповую дисперсии.

Внутригрупповая дисперсия вычисляется по формуле (среднее дисперсий внутри каждой группы):

Межгрупповая дисперсия вычисляется по формуле (дисперсии средних значений внутри каждой группы относительно ):

Выборочное корреляционное отношение к определяется в соответствии с выражением:

И определяется следующими свойствами:

* – нет оснований предположить, что между величинами X и Y существует корреляционная зависимость
* – выборочные данные дают основание предположить, что между величинами X и Y существует функциональная зависимость

– выборочные данные согласованы с предположением, что случайные величины X и Y связаны линейной корреляционной зависимостью.

**2.2. Элементы корреляционного анализа. Проверка статистической гипотезы о равенстве коэффициента корреляции нулю.**

Имеющаяся генеральная совокупность экспериментальных данных представлена в табл. 2.2.1. Объём выборки: 114.

Таблица 2.2.1 – Генеральная совокупность экспериментальных данных

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 405 | 107.5 | 24 | 418 | 125.7 | 47 | 528 | 163.4 | 70 | 476 | 143.0 | 93 | 391 | 98.2 |
| 2 | 521 | 154.9 | 25 | 477 | 139.7 | 48 | 462 | 135.7 | 71 | 464 | 143.2 | 94 | 482 | 150.1 |
| 3 | 382 | 98.1 | 26 | 468 | 142.0 | 49 | 453 | 119.5 | 72 | 449 | 124.5 | 95 | 453 | 126.4 |
| 4 | 362 | 111.7 | 27 | 545 | 145.3 | 50 | 393 | 103.2 | 73 | 331 | 84.6 | 96 | 465 | 140.9 |
| 5 | 444 | 130.0 | 28 | 441 | 140.8 | 51 | 487 | 146.0 | 74 | 444 | 121.4 | 97 | 458 | 121.7 |
| 6 | 465 | 114.8 | 29 | 543 | 155.4 | 52 | 430 | 95.3 | 75 | 423 | 104.1 | 98 | 482 | 139.9 |
| 7 | 513 | 163.6 | 30 | 474 | 132.5 | 53 | 371 | 91.9 | 76 | 446 | 130.3 | 99 | 448 | 125.6 |
| 8 | 362 | 84.3 | 31 | 321 | 86.1 | 54 | 441 | 126.1 | 77 | 451 | 128.6 | 100 | 411 | 112.9 |
| 9 | 550 | 147.9 | 32 | 438 | 120.7 | 55 | 495 | 150.9 | 78 | 434 | 140.4 | 101 | 452 | 119.7 |
| 10 | 453 | 138.2 | 33 | 409 | 121.0 | 56 | 510 | 129.4 | 79 | 469 | 131.5 | 102 | 504 | 143.8 |
| 11 | 482 | 141.2 | 34 | 451 | 124.3 | 57 | 496 | 141.7 | 80 | 452 | 131.0 | 103 | 454 | 131.1 |
| 12 | 431 | 125.0 | 35 | 462 | 125.2 | 58 | 525 | 162.1 | 81 | 397 | 108.6 | 104 | 422 | 122.9 |
| 13 | 437 | 129.2 | 36 | 541 | 162.3 | 59 | 377 | 96.0 | 82 | 470 | 124.0 | 105 | 443 | 137.4 |
| 14 | 352 | 87.7 | 37 | 460 | 140.7 | 60 | 512 | 169.9 | 83 | 351 | 102.9 | 106 | 510 | 153.9 |
| 15 | 351 | 89.0 | 38 | 358 | 98.3 | 61 | 382 | 113.9 | 84 | 430 | 104.3 | 107 | 448 | 125.0 |
| 16 | 498 | 164.0 | 39 | 353 | 98.0 | 62 | 480 | 153.9 | 85 | 541 | 146.8 | 108 | 442 | 123.4 |
| 17 | 510 | 162.3 | 40 | 508 | 159.0 | 63 | 391 | 107.5 | 86 | 467 | 140.5 | 109 | 423 | 115.9 |
| 18 | 547 | 154.7 | 41 | 379 | 94.6 | 64 | 448 | 137.3 | 87 | 394 | 112.1 | 110 | 504 | 145.3 |
| 19 | 477 | 146.0 | 42 | 458 | 128.0 | 65 | 540 | 156.7 | 88 | 406 | 110.1 | 111 | 471 | 143.9 |
| 20 | 428 | 131.6 | 43 | 548 | 162.3 | 66 | 421 | 126.9 | 89 | 477 | 135.8 | 112 | 493 | 154.5 |
| 21 | 458 | 124.4 | 44 | 447 | 117.5 | 67 | 453 | 124.2 | 90 | 424 | 119.0 | 113 | 544 | 166.7 |
| 22 | 470 | 146.7 | 45 | 446 | 128.4 | 68 | 441 | 122.8 | 91 | 475 | 143.6 | 114 | 460 | 122.4 |
| 23 | 499 | 144.5 | 46 | 483 | 130.3 | 69 | 452 | 116.1 | 92 | 352 | 84.9 |  |  |  |

Полученные данные при обработке выборки величины *E*, для удобства представлены в табл. 2.2.2 ниже, без дублирования вычислений.

Таблица 2.2.2 - Точечные статистические оценки параметров распределения

|  |  |
| --- | --- |
| Математическое ожидание | 129.8438 |
| Дисперсия | 439.691 |
| Среднеквадратичное отклонение | 20.9688 |
| Исправленное среднеквадратичное отклонение | 21.06138 |
| Асимметрия | -0.10855 |
| Эксцесс | -1.22854 |
| Мода | 124.175 |
| Медиана | 129.5941 |

Построим двумерный интервальный вариационный ряд (рис. 2.2.1)

Область значений, принимаемых каждой из величин, разбивается на 8 интервалов, вычисляются середины этих интервалов.

В таблице j-ая строка соответствует j-ому интервалу величины ν, i-ый столбец – i-ому интервалу величины E. В ячейки таблицы записывается абсолютная частота одновременного попадания значения ν двумерной случайной величины (ν,E) в j-ый интервал и значения E этой величины – в i-ый интервал.

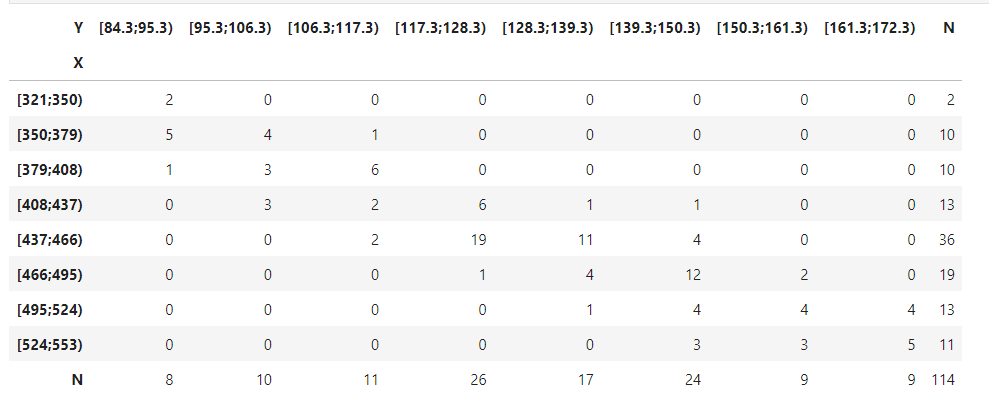


Рисунок 2.2.1 – таблица двумерного интервального ряда.

Основываясь на двумерном интервальном вариационном ряде построим корреляционную таблицу. Для этого внесем суммарные абсолютные частоты для каждого из интервалов величин (значения совпали, с представленными в лабораторной работе 1, следовательно вычисления верны). Результат представлен на рис. 2.2.2.

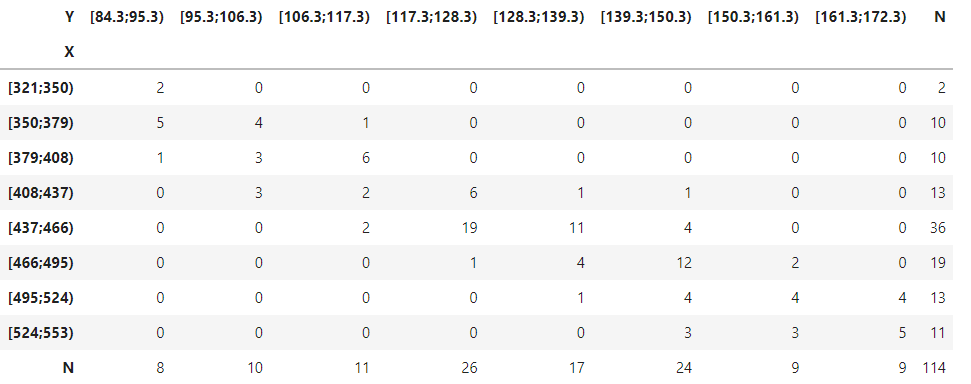


Рисунок 2.2.2 – корреляционная таблица двумерного интервального ряда.

Далее преобразуем середины интервалов в условные варианты, для упрощения вычислений. В качестве условного нуля примем середин интервала с наибольшей частотой. На рис. 2.2.3 представлена таблица, с интервалами, замененными на условные варианты



Рисунок 2.2.3 – таблица двумерного интервального ряда с условными вариантами.

Вычислим выборочный коэффициент корреляции при помощи условных вариант.

Значение выборочного коэффициента корреляции вычисляется по формуле:

где - выборочные средние для условных вариант, – неисправленные среднеквадратические отклонения условных вариант.

Для вычисления построим вспомогательную таблицу (умножим абсолютные частоты на условные варианты). Результаты представлены на рис.2.2.4.



Рисунок 2.2.4 – вспомогательная таблица двумерного интервального ряда.

Вычислим выборочный коэффициент корреляции с помощью стандартной формулы

Значение выборочного коэффициента корреляции вычисляется по формуле:

где – неисправленные среднеквадратические отклонения условных вариант. В начале лабораторной работы приведены значения исправленных СКО , которые вычисляются как:



Тогда формула примет вид:

Значение совпало с предыдущим, значит вычисления проведены верно. Данное значение близко к единице и положительно, значит можно сделать вывод о том, что данные сильно коррелируют и при возрастании одной величины произойдет возрастание и второй.

Построим доверительный интервал для коэффициента корреляции при уровне значимости .

Используем *Z* преобразование Фишера. Где распределение :

СКО для распределения:

Для построения доверительного интервала для коэффициента корреляции сделаем обратное преобразование Фишера:

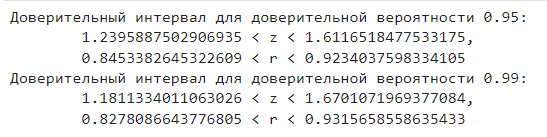


Рисунок 2.2.5 – Вычисления доверительных интервалов

Можно заметить, что при увеличении уровня надежности получили более широкий доверительный интервал.

Проверим гипотезу о равенстве нулю коэффициента корреляции. Вычислим по формуле:

(для уровня значимости )

Исходя из того, что , гипотеза о равенстве нулю коэффициента корреляции отвергается.

**2.3. Элементы регрессионного анализа. Выборочные прямые. среднеквадратической регрессии. Корреляционные отношения.**

Зависимость между двумя величинами можно наблюдать на диаграмме рассеяния, представленной на рис.2.3.1.

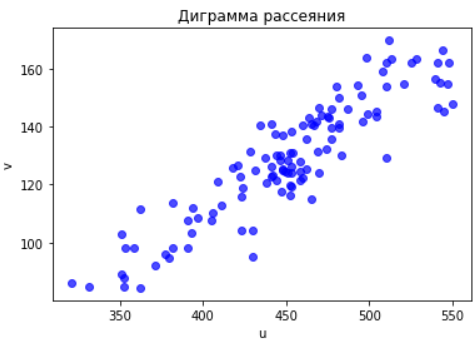


Рисунок 2.3.1 – Диаграмма рассеяния

Далее построим уравнение выборочной прямой среднеквадратичной регрессии, а также уравнение среднеквадратичной регрессии для величин *E* и *ν*:

Рассчитаем эти значения (см. рис. 2.3.2)



Рисунок 2.3.2 – Вычисления с помощью программы

Данные регрессионные прямые можно отобразить на графике, представленном на рис.3. Точкой отмечена координата, которая является пересечением наших прямых.

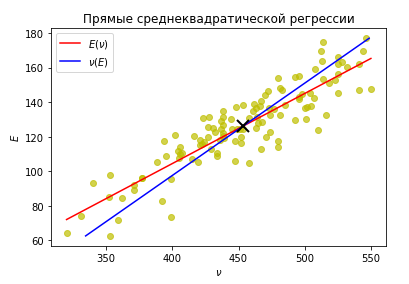


Рисунок 2.3.3 – график прямых среднеквадратичной регрессии

Далее можно найти дисперсию отклонений для величин *E* и  *v*, данная остаточная дисперсия равна:

Вычислим данные значения:

Можно наблюдать, что остаточная дисперсия величины *E* сильно меньше величины *v*, что можно наблюдать на рис.3, где по оси *E* точки меньше отступают от прямой *E(v)*, чем у второй величины.

Составим корреляционную таблицу для нахождения выборочного корреляционного отношения. Для фиксированных значений величин рассматриваем наблюдения (*E*, *v*) и (*v*, *E*) как группу и вычисляем средние значения величины в этой группе. Представим в таблице значения величины в этой группе (рис. 2.3.4).

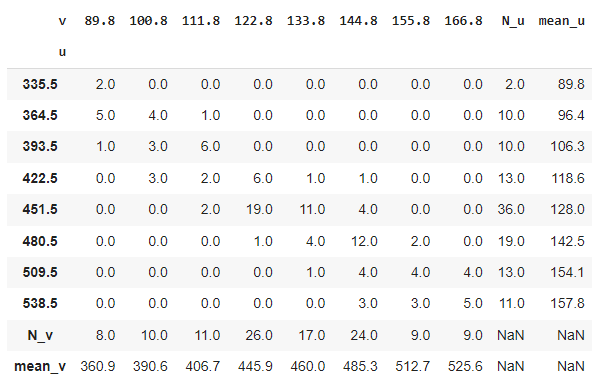


Рисунок 2.3.4 – Корреляционная таблица

Для каждой величины вычислим внутрегрупповую и межгрупповую дисперсии, а также их сумму (рис. 2.3.5).

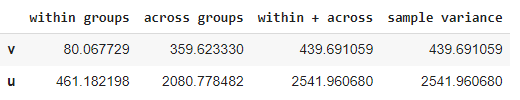


Рисунок 2.3.5 –Таблица вычислений внутригрупповой и межгрупповой дисперсии

Как мы можем наблюдать, сумма равна выборочной дисперсии, следовательно вычисления произведены верно.

Далее вычислим корреляционные отношения *E* к *v*, *v* к *E.*

Выборочное корреляционное отношение к определяется в соответствии с выражением:



= 0.904378

= 0.904750

Выборочное корреляционное отношение должно быть не меньше модуля выборочного коэффициента корреляции, который в нашем случае равен:

Как мы видим, значения корреляционных отношений соответствуют ожиданиям и близки к 1, следовательно между величинами присутствует сильная корреляционная зависимость.

Далее необходимо построить параболическую корреляционную кривую, определив значения коэффициентов с помощью МНК.

Для определения коэффициентов корреляционной кривой параболического вида была решена следующая система уравнений:

Система была решена с помощью написанной программы на языке Python матричным способом.В результате работы программы были получены следующие значения коэффициентов:



В связи с тем, что коэффициент *a* достаточно мал, то кривые будут близки к прямым, что можно увидеть на рис. 2.3.6.

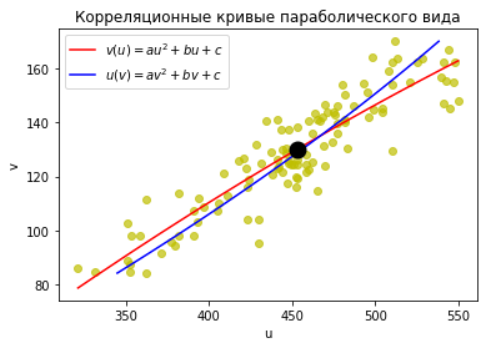


Рисунок 2.3.6 - Корреляционные кривые параболического вида

Согласно варианту построим корреляционную кривую функции

.

Для этого воспользуемся методом МНК и выразим с его помощью искомые переменные. После расчета на исходных данных получим следующие значения коэффициентов кривой обратной функции:



Построим график для найденных коэффициентов.

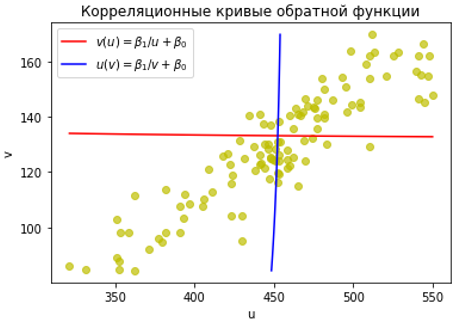


Рисунок 2.3.7 - Полученная кривая на множестве выборки

Видно, что полученные прямые очень плохо приближают значения распределения – вероятно, это связано с тем, что используемая функция не подходит для рассматриваемых данных, т.к. между распределениями величин E и v наблюдается прямая зависимость.

**2.4. Выводы.**

Был проведен корреляционный анализ распределения случайных величин. Был вычислен линейный коэффициент корреляции двумя различными способами. Его близость к 1 дает нам сделать вывод о сильной корреляционной зависимости меду величинами. Также были построены доверительные интервалы с различными вероятностями и проведена проверка, в ходе которой мы отвергли нулевую гипотезу.

Из перечисленного можно сделать вывод, что выборочный коэффициент корреляции значимо отличен от нуля.

Была построена выборочная прямая линейной регрессии, коэффициенты которой вычислены с помощью метода наименьших квадратов. Было вычислено корреляционное отношение величины *E* к *ν* и величины *ν* к *E*, что подтвердило сильную корреляционную зависимость величин. Были построены корреляционные кривые параболического вида и обратной функции. Также, с помощью метода наименьших квадратов, были рассчитаны коэффициенты кривых.

**3. Кластерный анализ**

**3.1. Основные теоретические положения**

Кластерный анализ – многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы.

К характеристикам кластера относятся в частности: центр, радиус; среднеквадратическое отклонение; размер кластера.

Центр кластера – это среднее геометрическое место точек, принадлежащих кластеру, в пространстве данных.

Радиус кластера – максимальное расстояние точек, принадлежащих кластеру, от центра кластера.

Нормировка, т.е. стандартизация, переменных применяется для того, чтобы характеристики имели один масштаб, в следствии чего было возможно корректное разбиение на кластеры. В данной работе для нормировки была использована формула

Где стандартное отклонение переменной, а – ее среднее значение.

Евклидово расстояние (способ определения расстояния между наблюдениями):

Алгоритм -means – это наиболее популярный метод кластеризации, который разделяет определенный набор данных на заданное пользователем число кластеров .  В начале классификации задается число k классов и выбираются k точек, которые будут служить центрами кластеров. Для каждого наблюдения из исходной выборки вычисляются расстояния до центров кластеров. Наблюдение распределяется в кластер, центр которого находится ближе всего к нему.

Возможны две разновидности метода *k* -means. Первая предполагает пересчет центра кластера после каждого изменения его состава, а вторая —лишь после завершения цикла.

Метод поиска сгущений является еще одним итеративным методом кластерного анализа. Основная идея метода заключается в построении гиперсферы заданного радиуса, которая перемещается в пространстве классификационных признаков в поисках локальных сгущений объектов.

Алгоритм можно описать так:

1. В качестве начального центра кластера выбирается наблюдение из выборки еще не распределенных наблюдений, на расстоянии R от которого находится наибольшее число других наблюдений;
2. Формируется кластер относительно текущего центра;
3. В качестве нового центра выбирается среднее геометрическое место точек, входящих в кластер
4. Повторяем пункты 2 и 3, пока состав нового кластера не совпадет с составом кластера, полученного на предыдущей итерации.
5. Добавляем полученные данные (кластер и его центр) и запоминаем выборку нераспределенных элементов.

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 расстояние между -м и -м объектами, то в качестве нижней границы значения радиуса сферы можно выбрать минимальное из таких расстояний, а в качестве верхней границы - максимальное:

Тогда, если начинать работу алгоритма с

и при каждом его повторении увеличивать значение на некоторую величину, то в конечном итоге можно найти значения радиусов, которые приводят к устойчивому разбиению на кластеры.

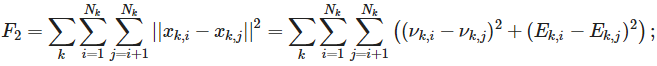
Наилучшим разбиением считается такое, при котором достигается экстремальное (минимальное или максимальное) значение выбранного функционала качества.

В качестве таких функционалов могут быть использованы:

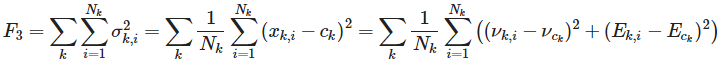
* Сумма по всем кластерам квадратов расстояний элементов кластеров до центров соответствующих кластеров:



* Сумма по всем кластерам внутрикластерных расстояний между элементами кластеров:



* Сумма по всем кластерам внутрикластерных дисперсий (относительно центров кластеров):



Оптимальным следует считать разбиение, при котором сумма внутрикластерных (внутригрупповых) дисперсий будет минимальной.

**3.2. Метод k-средних**

Нам нужно корректно реализовать методы кластерного анализа, для чего мы нормализуем нашу выборку по формуле:

Где стандартное отклонение переменной, а – ее среднее значение.

Нормализированное множество представлено на рис.3.2.1 в таблице, а также на рис. 3.2.2 на диаграмме рассеяния.

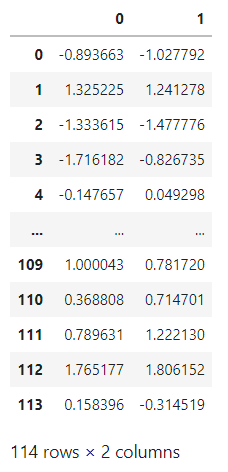


Рисунок 3.2.1 – таблица, сгенерированная программой

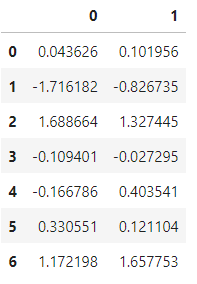


Рисунок 3.2.2 – диаграмма рассеяния нормализованных величин

Далее мы определяем верхнюю оценку количества кластеров:



Среди наблюдаемых значений выборки отбираются случайным образом начальные центры кластеров, в связи с тем, что проблематично определить визуально, где они точно находятся. Далее эти центры были отмечены крестиками на рис. 3.2.3.



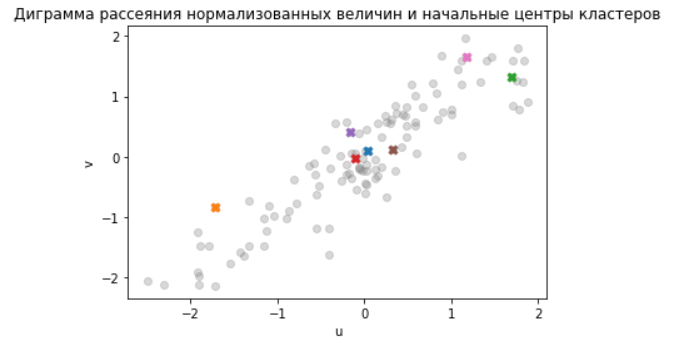
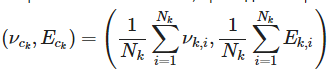


Рисунок 3.2.3 – Диаграмма рассеяния с начальными центрами классов

Далее были реализованы функции для осуществления метода кластеризации *k-means:*

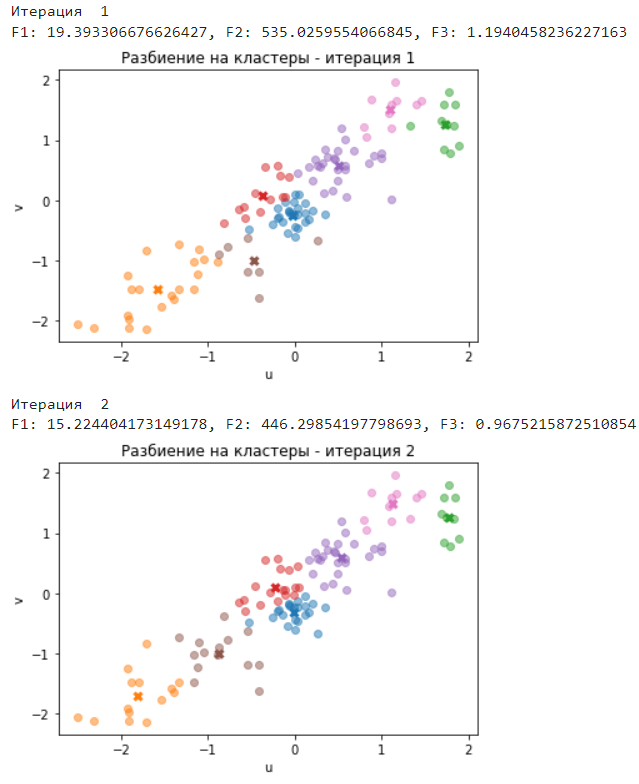
* Функция, определяющая индекс кластера, котору принадлежит данное наблюдение (С помощью евклидова расстояния выбирается наименьшее расстояние).
* Функция, осуществляющая пересчет центров кластеров (среднее геометрическое место точек, принадлежащих кластеру). В случае пустоты кластера – центр не пересчитывается.



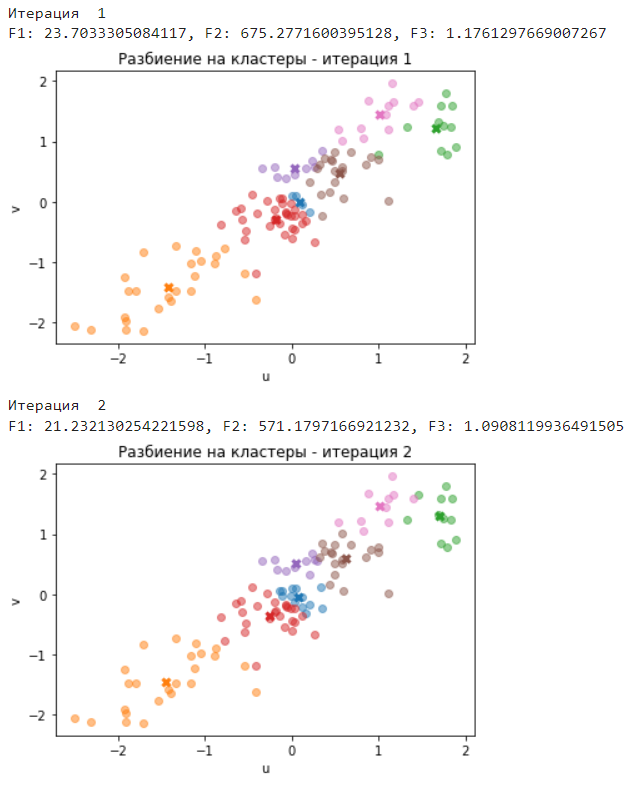
* Функция-перебор элементов для распределения в кластеры
* Три функции для функционалов качества полученного разбиения. Т.е. функции для сумм по всем кластерам квадратов расстояний элементов кластеров до центров соответствующих кластеров, внутрикластерных расстояний между элементами кластеров и внутрикластерных дисперсий.

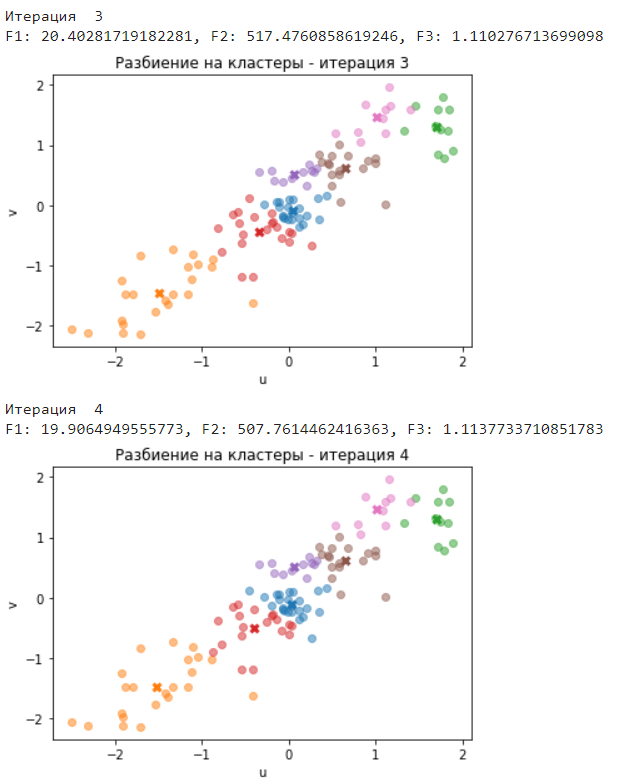
Диаграмма рассеяния выводится на каждом шаге, с пометками центров кластеров, вместе со значениями функционалов качества. При совпадении результата с предыдущей итерацией – алгоритм завершает работу.

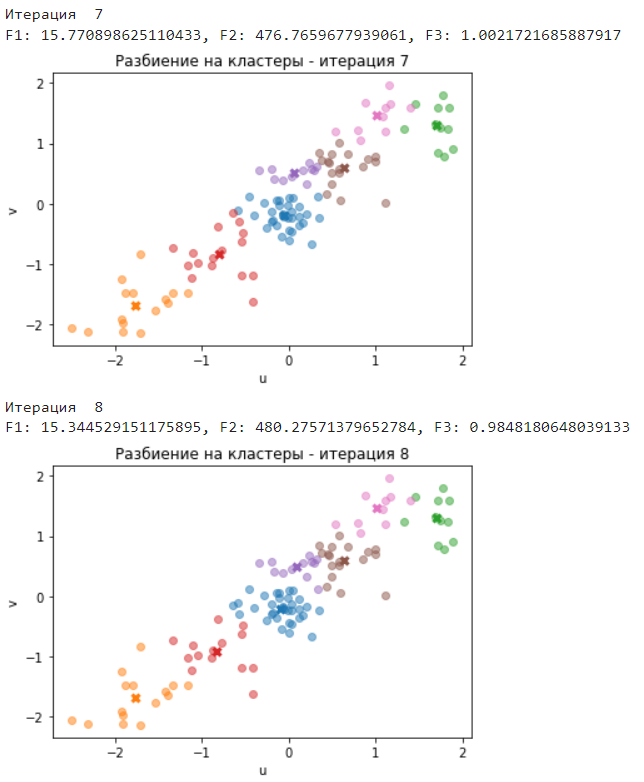
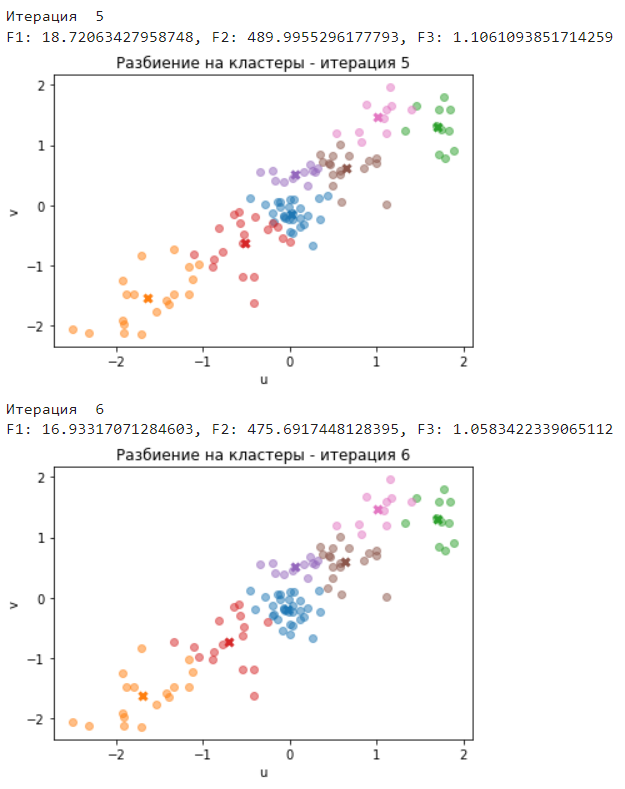
Работа алгоритма с пересчетом центров кластеров после каждого изменения состава кластеров:

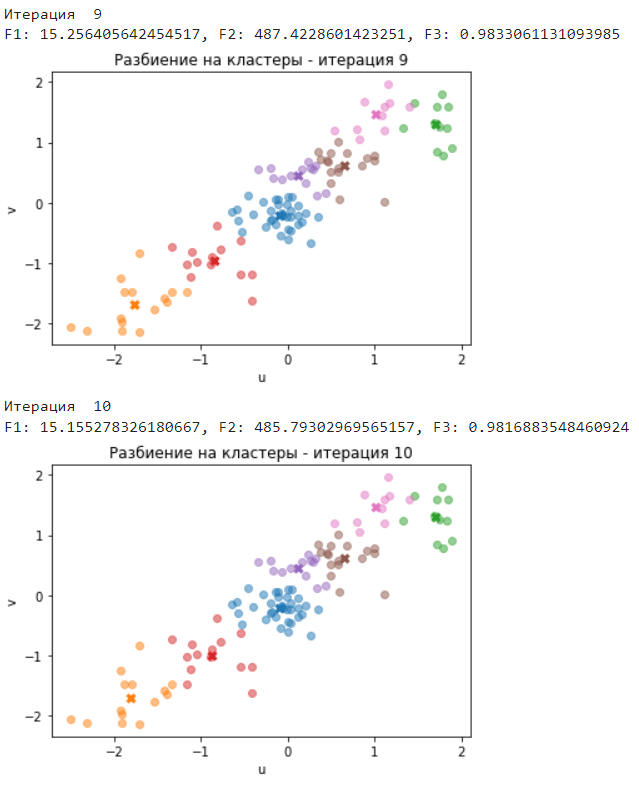
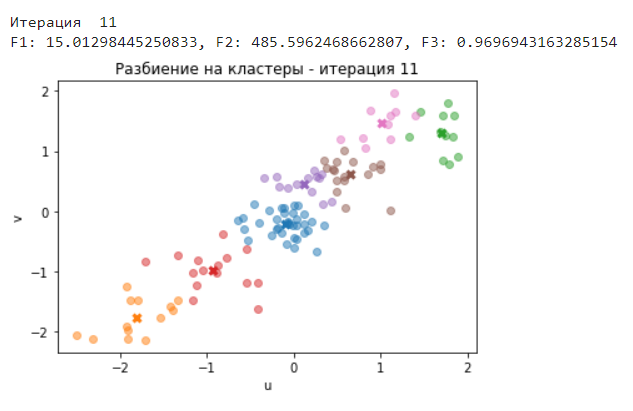


Работа алгоритма с пересчетом центров кластеров после просмотра всех выборочных значений:







Как мы можем заметить, алгоритм, пересчитывающий центры после каждого изменения состава кластеров, работает гораздо быстрее, но результат полученный двумя алгоритмами различается. В первом случае мы получили немного большее расстояние от элементов до центров кластеров (функционал 1), но меньшее внутрикластерное расстояние и внутрикластерные дисперсии, относительно центров (функционал 2 и 3).

**3.3. Метод поиска сгущений**

Реализуем алгоритм поиска сгущений. Отобразим полученные кластеры, выделим каждый кластер разным цветом, отметим центроиды.

Определим нижнюю и верхнюю границы радиуса сферы:

Сам алгоритм поиска сгущений описан в основных теоретических положениях. Визуально представим его на примере радиуса, входящего в промежуток, *R*=1.0719558085668415 (при дальнейшей проверке устойчивости соединений, мы опустим промежуточные шаги, для удобства прочтения отчета).

|  |  |
| --- | --- |
| Формирование 1го кластера | |
| *1.1*  *1.3*  *1.5* | *1.2*  *1.4*  *1.6* |

|  |  |
| --- | --- |
| Формирование 2 кластера | |
| *2.1*  *2.3*  *2.5* | *2.2*  *2.4* |

|  |  |
| --- | --- |
| Формирование 3го кластера | |
| *3.1* | *3.2* |

|  |  |
| --- | --- |
| Формирование 4го кластера | |
|  |  |
| *4.1* | *4.2* |

Результат кластеризации представлен на рис. 3.3.1.

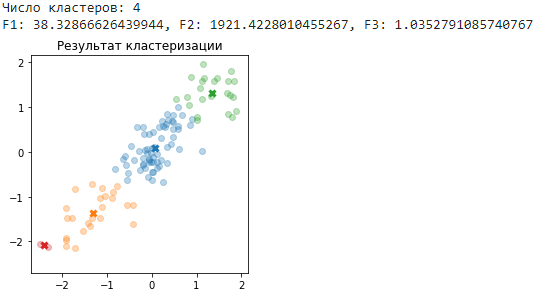


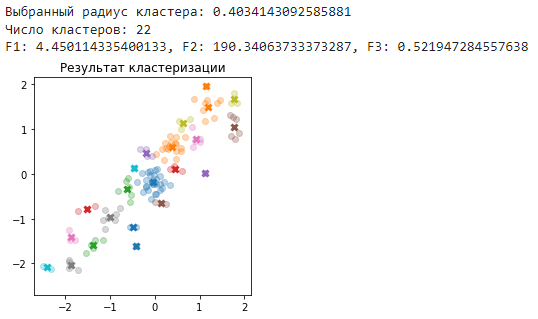
Рисунок 3.3.1 – Результат кластеризации

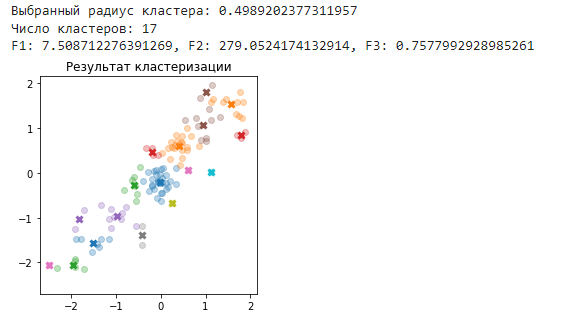
Таблица 3.3.1 – Центры вычисленных кластеров и количество элементов в них

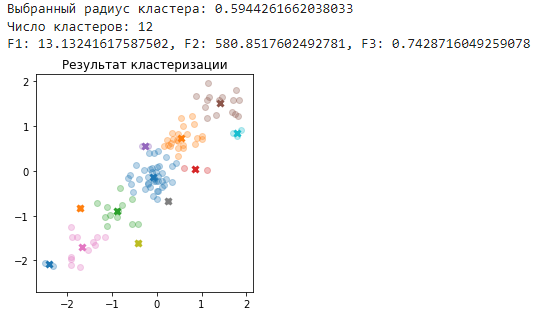
|  |  |  |
| --- | --- | --- |
| Номер  кластера | Центр кластера | Количество элементов в кластере |
| 1 | (0.08217701526223979, 0.09878930409319236) | 65 |
| 2 | (-1.3057191761513063, -1.3567031373700817) | 24 |
| 3 | (1.339363393379832, 1.3180793373874455) | 23 |
| 4 | (-2,1754394392808516 ; -2,465932878978102) | 2 |

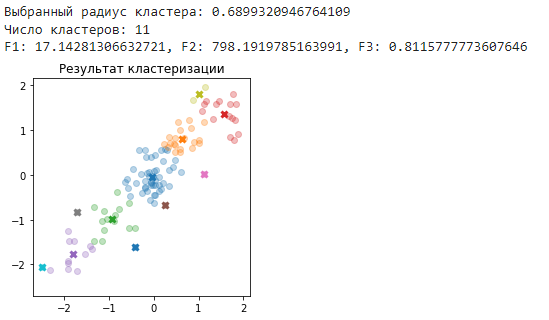
Проанализируем устойчивость метода поиска сгущений для разных радиусов. Ближайшие к границам значения в дальнейших примерах мы опустим, так как при них создавались либо слишком большие кластеры, куда входила почти вся выборка, либо слишком маленькие, куда входило 1-3 элемента, что является нецелесообразной кластеризацией.

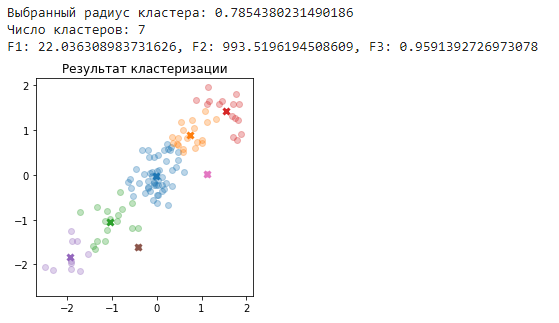
Ниже приведены вычисления разбиения на кластеры с постепенным увеличением радиуса. Для каждого из которых приведено 3 функционала качества.

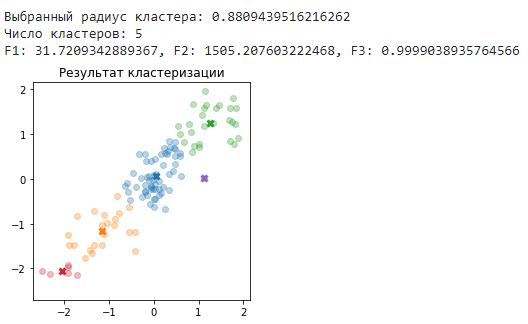


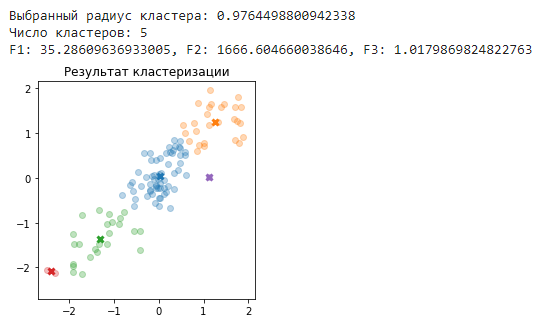


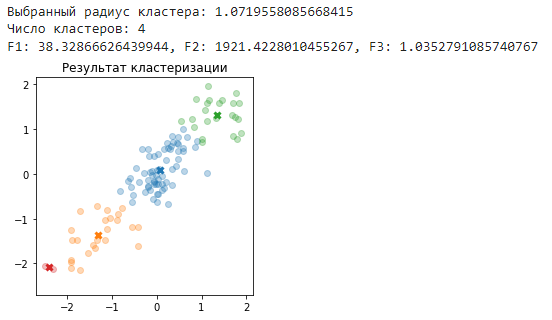


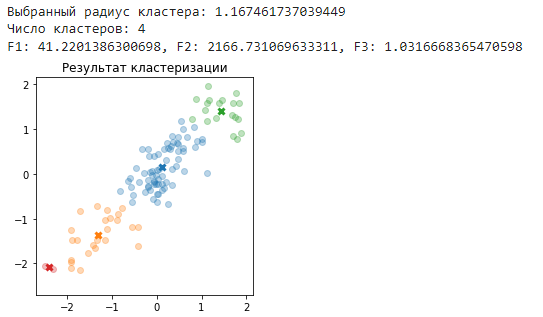


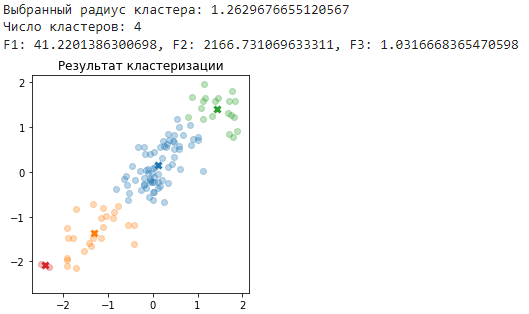






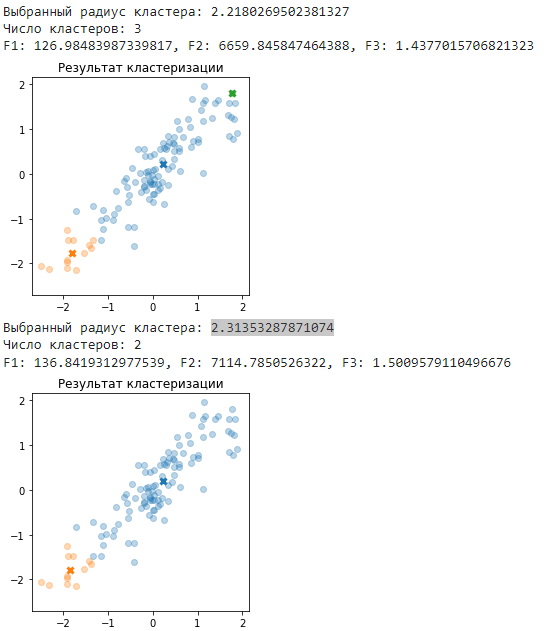








Далее присутствует промежуток радиусов, которые разделяют выборку на 3 кластера, вплоть до радиуса *R=* *2.31353287871074*



Оценив приведенные результаты можно сделать вывод, что малые радиусы окружности порождают неустойчивые разбиения. Можно сделать предположение, что это вызвано очень малым расстоянием наблюдений, близких к выборочным средним. При всем перечисленном, значения функционалов качества до *R=* *0.6899320946764109* меньше, чем было получено при использовании метода k-means. Можно увидеть, что при увеличении радиуса увеличиваются значения функционалов качества, но при этом мы наблюдаем более устойчивые разбиения.

**3.4. Выводы.**

Были освоены основные понятия кластерного анализа.С помощью метода *k-means* было осуществлено распределение наблюдений по семи кластерам. Были реализованы две вариации алгоритма. Алгоритм с пересчетом центров кластеров после каждого изменения их состава - сходится гораздо быстрее (2 итерации) и захватывает значения имеющие меньшее внутрикластерное расстояние и внутрикластерные дисперсии, относительно центров. Алгоритм с пересчетом центров кластеров после просмотра всех элементов выборки захватывает значения менее удаленные от центра выборки.

Был реализован алгоритм метода поиска сгущений. Для применения алгоритма были применены различные радиусы окружностей для проверки соединений на устойчивость. В следствии чего, было сделано предположение о том, что при увеличении радиуса, увеличивается и устойчивость соединения. Но в связи с не таким плотным расположением по краям и сгущением около выборочных средних, значение радиуса сильно влияет на состав кластеров. Функционалы качества показывают нам, что качество разбиения значительно снижается от увеличения радиуса, не смотря на устойчивость разбиений.

При сравнении реализованных методов кластеризации, можно сказать о том, что при больших значениях радиуса окружности, метод поиска сгущений дает худшие функционалы качества, чем при методе k-means. Также этот метод является более медленным, чем k-means. Преимущество k-means в том, что можно оценить число кластеров. Лишь при малых значениях радиуса окружности можно наблюдать меньшие показатели функционалов качества, чем было получено при использовании метода k-means. Из чего можно сделать вывод, что метод k-means дает лучшие результаты для исследуемой выборки.

**заключение**

Был проведен статистический анализ двумерной выборки. При выполнении задания на выравнивание статистических рядов были построены статистические, ранжированные, вариационные и интервальные ряды, графически построены полигоны частот, гистограммы, эмпирические функции распределения двухмерной выборки, найдены выборочные оценки: среднего, дисперсии, исправленной дисперсии, СКВО, асимметрии, эксцесса, медианы и моды, построены доверительные интервалы для математического ожидания и СКО, проверена гипотеза о нормальном законе по критерию Пирсона, результат которой показал, что исследуемое распределение не принадлежит нормальному закону распределения. Для отдельных параметров, например, эксцесса, была выполнена проверка путем выполнения расчетов другими методами, демонстрирующая корректность сделанных вычислений.

В ходе проведения корреляционного и регрессионного анализа для второго признака был выполнен статический анализ, аналогичный выполненному ранее для первого признака. Была построена корреляционная таблица, найдены оценка и доверительный интервал коэффициента корреляции, проверена и отвергнута гипотеза о равенстве коэффициента корреляции нулю. Рассчитаны и отображены на одном графике уравнения прямых среднеквадратической регрессии и найдены оценки корреляционных отношений. Далее были построены и отображены уравнения выборочных кривых для параболической среднеквадратической регрессии, а также уравнения кривых обратной функции, выбранной в соответствии с вариантом работы.

В разделе кластерного анализа было в первую очередь нормализовано множество точек признаков. Для нормализованного множества был реализован алгоритм k-средних; для этого было предварительно грубо оценено количество кластеров. Алгоритм k-средних реализован в двух вариантах: с пересчетом центров кластеров после каждого изменения их составов и с пересчетом центров после завершения просмотра всех данных. Для каждого шага метода были рассчитаны функционалы качества разбиений, что позволило убедиться в улучшении результатов разбиения при проходе по итерациям. Также была выполнена кластеризация методом поиска сгущений. По результатам кластеризации были составлены графики, а также рассчитаны функционалы качества аналогично алгоритму k-средних. Для рассматриваемой выборки проведена оценка чувствительности метода к погрешностям, которая показала, что устойчивость возрастает по мере увеличения радиусов кластеров, но при этом одновременно значительно ухудшаются значения функционалов качества. Сравнение методов k-средних и метода поиска сгущений показало, что метод k-средних показывает лучшие результаты на исследуемой выборке, являясь более быстрым и точным.

**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., Парилина 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. Кластеризация // machinelearning.ru URL: http://www.machinelearning.ru/wiki/index.php?title=Линейная\_регрессия (дата обращения: 07.04.2021).
10. Учебные материалы к курсу СМОЭД // МОЭВМ Вики. URL: http://se.moevm.info/doku.php/courses:statistical\_methods\_of\_experimental\_d ata\_handling:materials (дата обращения: 10.04.2022)

**приложение А**

**Программа для формирования и первичной обработки выборки, построения, ранжированного и интервального рядов.**

**lr1.py**

from tkinter import ROUND

import numpy as np

from math import \*

import matplotlib.pyplot as plt

ROUND\_ACC = 5

def m\_round(a):

return round(a, ROUND\_ACC)

def m\_np\_round(a):

return np.round(a, ROUND\_ACC)

def make\_spaced(\_data):

\_n = len(\_data)

\_k = 1 + 3.322 \* log10(\_n)

\_max = max(\_data)

\_min = min(\_data)

\_h = (\_max - \_min) / \_k

print("Интервальный ряд: k={} (формула Спенсера), h={}={} (шаг)".format(\_k, \_h, round(\_h)))

\_h = round(\_h)

\_from, \_upto = \_min, \_min + \_h

\_spaces = []

\_abs\_freqs = []

\_rel\_freqs = []

\_space\_count = 0

for i in \_data:

if i >= \_upto:

\_spaces.append([\_from, \_upto])

\_abs\_freqs.append(\_space\_count)

\_rel\_freqs.append(\_space\_count / \_n)

\_from = \_upto

\_upto = \_from + \_h

\_space\_count = 1

else:

\_space\_count += 1

\_spaces.append([\_from, \_upto])

\_abs\_freqs.append(\_space\_count)

\_rel\_freqs.append(\_space\_count / \_n)

return np.array(\_spaces), np.array(\_abs\_freqs), np.array(\_rel\_freqs)

source\_dataset = np.genfromtxt('dataset.csv', delimiter=',')

source\_len = len(source\_dataset)

data = source\_dataset[:,0].astype(int)

print("Исходный ряд:\n", data)

data = np.sort(data)

ranked\_data = np.array(data)

print("Ранжированный ряд:\n", ranked\_data)

uniq, abs\_freqs = np.unique(data, return\_counts=True)

abs\_freqs\_table = np.array([tuple([uniq[i], abs\_freqs[i]]) for i in range(len(uniq))], dtype='i, i')

print("Вариационный ряд - абсолютные частоты:\n", abs\_freqs\_table)

rel\_freqs = (abs\_freqs / source\_len).astype(float)

rel\_freqs\_table = np.array([tuple([uniq[i], rel\_freqs[i]]) for i in range(len(uniq))], dtype='i, f')

print("Вариационный ряд - относительные частоты:\n", rel\_freqs\_table)

spaces, spaces\_abs\_freq, spaces\_rel\_freq = make\_spaced(data)

print("Интервальный ряд:\n{}\t{}\t{}".format("Интервал", "Абс. частоты", "Отн. частоты"))

for i in range(len(spaces)):

print("[{}, {})\t{}\t\t{}".format(spaces[i][0], spaces[i][1], spaces\_abs\_freq[i], m\_round(spaces\_rel\_freq[i])))

spaces\_median = np.median(spaces, axis=1)

spaces\_median\_str = [str(i) for i in spaces\_median]

plt.plot(spaces\_median, spaces\_abs\_freq)

plt.title("Полигон для абсолютной частоты")

plt.xlabel("Средние значения")

plt.ylabel("Частоты")

plt.grid()

plt.show()

plt.plot(spaces\_median, spaces\_rel\_freq)

plt.title("Полигон для относительной частоты")

plt.xlabel("Средние значения")

plt.ylabel("Частоты")

plt.grid()

plt.show()

plt.bar(spaces\_median\_str, spaces\_abs\_freq, width=0.7)

plt.title("Гистограмма для абсолютной частоты")

plt.xlabel("Средние значения")

plt.ylabel("Частоты")

plt.show()

plt.bar(spaces\_median\_str, spaces\_rel\_freq)

plt.title("Гистограмма для относительной частоты")

plt.xlabel("Средние значения")

plt.ylabel("Частоты")

plt.show()

space\_width = spaces\_median[1] - spaces\_median[0]

spaces\_median\_t = np.array([spaces\_median[0] - space\_width, \*spaces\_median, spaces\_median[-1] + space\_width])

spaces\_distrib = [0, 0]

t = 0

for i in spaces\_abs\_freq:

t += i

spaces\_distrib.append(t)

plt.step(spaces\_median\_t, spaces\_distrib)

plt.title("Эмпирическая ф.р. для абсолютной частоты")

plt.xlabel("Средние значения")

plt.ylabel("Абсолютная частота")

plt.xticks(spaces\_median\_t)

plt.grid()

plt.show()

spaces\_distrib = [0, 0]

t = 0

for i in spaces\_abs\_freq:

t += i / np.sum(spaces\_abs\_freq)

spaces\_distrib.append(t)

plt.step(spaces\_median\_t, spaces\_distrib)

plt.title("Эмпирическая ф.р. для относительной частоты")

plt.xlabel("Средние значения")

plt.ylabel("Относительная частота")

plt.xticks(spaces\_median\_t)

plt.grid()

plt.show()

**приложение Б**

**Программа для** **нахождения точечных оценок параметров распределения.**

**lr2.py**

from lr1 import spaces, spaces\_abs\_freq, data, h

import pandas as pd

import numpy as np

from math import sqrt

df = pd.DataFrame()

df['Средние значения'] = list(

map(lambda space: (space[0] + space[1]) / 2, spaces))

df['Частоты'] = spaces\_abs\_freq

C = df.iloc[4, 0]

df['Условные варианты'] = df['Средние значения'].apply(lambda x: (x - C) / h)

# условные эмпирические моменты, M с чертой и звездочкой

moments = []

for i in range(1, 5):

col = 'nu{}'.format(i)

df[col] = df.iloc[:, 1:3].apply(lambda x: x[0] \* x[1] \*\* i, axis=1)

moments.append(df[col].sum() / len(data))

df['Проверка'] = df.iloc[:, 1:3].apply(lambda x: x[0] \* ((x[1]+1)\*\*4), axis=1)

print(df)

for i, m in enumerate(moments):

print("Условный эмпирический момент {} порядка: {}".format(i + 1, m))

sum1 = df['nu4'].sum() + df['nu3'].sum()\*4 + df['nu2'].sum()\*6 + df['nu1'].sum()\*4 + df['Частоты'].sum()

sum2 = df['Проверка'].sum()

print("Проверка:", sum1, sum2, "если сошлось - все ок")

print(" --- Вычисляем через условные моменты:")

start\_moment\_1\_usl = moments[0]\*h + C

print('Начальный эмпирический момент 1го порядка: ', start\_moment\_1\_usl)

central\_moment\_2\_usl = (moments[1] - moments[0]\*\*2)\*(h\*\*2)

print('Центральный эмпирический момент 2го порядка: ', central\_moment\_2\_usl)

central\_moment\_3\_usl = (moments[2] - 3\*moments[1]

\* moments[0] + 2\*(moments[0]\*\*3))\*(h\*\*3)

print('Центральный эмпирический момент 3го порядка: ', central\_moment\_3\_usl)

central\_moment\_4\_usl = (moments[3] - 4\*moments[2]\*moments[0] +

6\*moments[1]\*(moments[0]\*\*2) - 3\*(moments[0]\*\*4))\*(h\*\*4)

print('Центральный эмпирический момент 4го порядка: ', central\_moment\_4\_usl)

print(' --- Вычисляем по стандартной формуле:')

start\_moment\_1\_emp = df.iloc[:, :2].apply(

lambda x: x[0]\*x[1], axis=1).sum() / len(data)

print('Начальный эмпирический момент 1го порядка: ', start\_moment\_1\_emp)

central\_moment\_2\_emp = df.iloc[:, :2].apply(lambda x: (

(x[0] - start\_moment\_1\_emp)\*\*2)\*x[1], axis=1).sum() / len(data)

print('Центральный эмпирический момент 2го порядка: ', central\_moment\_2\_emp)

print(" --- ")

s = np.sqrt((len(df)/(len(df)-1)) \* central\_moment\_2\_emp)

asim = central\_moment\_3\_usl / (s\*\*3)

print('Асимметрия: ', asim)

ecs = central\_moment\_4\_usl / (s\*\*4) - 3

print('Эксцесс: ', ecs)

space\_moda\_index = np.argmax(spaces\_abs\_freq)

m2, m1, m3 = spaces\_abs\_freq[space\_moda\_index], 0, 0

if space\_moda\_index > 0:

m1 = spaces\_abs\_freq[space\_moda\_index - 1]

if space\_moda\_index < len(spaces\_abs\_freq):

m3 = spaces\_abs\_freq[space\_moda\_index + 1]

moda = spaces[space\_moda\_index][0] + h \* ((m2 - m1) / (2 \* m2 - m1 - m3))

print("Мода:", moda)

median\_index = 4

median\_lower\_freqs = np.sum(spaces\_abs\_freq[:median\_index])

median = \

spaces[median\_index][0] \

+ h \* ((0.5 \* len(data) - median\_lower\_freqs) /

spaces\_abs\_freq[median\_index])

print("Медиана:", median)

print(" --- ")

x\_middle = df.iloc[:, 0:2].apply(lambda x: x[0] \* x[1], axis=1).sum() / len(data)

print("Мат. ожидание:", x\_middle)

disp = df.iloc[:, 0:2].apply(lambda x: (x[0] - x\_middle)\*\*2 \* x[1], axis=1).sum() / len(data)

print("Дисперсия:", disp)

sko = sqrt(disp)

print("СКО:", sko)

N = len(data)

S2 = N/(N-1)\*disp

print("Исправленная выборочная дисперсия: S^2={}, СКО S=sqrt(S^2)={}".format(S2, sqrt(S2)))

**приложение В**

**Программа для** **нахождения интервальных оценок параметров распределения и проверки статистической гипотезы о нормальном распределении.**

**lr3.py**

**from** lr2 **import** x\_middle, disp, sko, S2, N, spaces, spaces\_abs\_freq

**import** math

**import** pandas **as** pd

**import** numpy **as** np

**from** math **import** sqrt

**from** scipy **import** special

*# функция Лапласа*

**def** laplas(x):

**return** special**.**erf(x**/**2**\*\***0.5)**/**2

S **=** sqrt(S2)

print(pd**.**DataFrame({

"Интервал": map(**lambda** x: "[{},{})"**.**format(x[0], x[1]), spaces),

"Середина": map(**lambda** x: ((x[0] **+** x[1]) **/** 2)**.**round(2), spaces),

"Частоты": spaces\_abs\_freq

}))

print("N: {}"**.**format(N))

print("Матожидание: {}"**.**format(x\_middle))

print("Дисперсия: {}"**.**format(disp))

print("СКО: {}"**.**format(sko))

print("Исправленная дисперсия: {}"**.**format(S2))

print("Исправленное СКО: {}"**.**format(S))

\_gamma **=** 0.99

\_t **=** 2.617

\_t **=** S **\*** \_t **/** np**.**sqrt(N)

\_i1 **=** x\_middle **-** \_t

\_i2 **=** x\_middle **+** \_t

print("Для надежности {} и объема выборки n={} было выбрано значение t={} по приложению 6"**.**format(\_gamma, N, \_t))

print("Доверительный интервал для матожидания: ({}, {})"**.**format(\_i1, \_i2))

\_x **=** list(map(**lambda** x: x[0], spaces))

\_x**.**append(spaces[**-**1][1])

df **=** pd**.**DataFrame()

df['x\_i'] **=** \_x[:**-**1]

df['x\_(i+1)'] **=** \_x[1:]

df['n\_i'] **=** spaces\_abs\_freq

df['z\_i'] **=** df**.**iloc[:, 0]**.**apply(**lambda** x: (x **-** x\_middle) **/** S)

df['z\_(i+1)'] **=** df**.**iloc[:, 1]**.**apply(**lambda** x: (x **-** x\_middle) **/** S)

df['F(z\_i)'] **=** df**.**iloc[:, 3]**.**apply(**lambda** x: laplas(x))

df['F(z\_(i+1))'] **=** df**.**iloc[:, 4]**.**apply(**lambda** x: laplas(x))

df['p\_i'] **=** df**.**iloc[:, 5:7]**.**apply(**lambda** x: x[1] **-** x[0], axis**=**1)

df['n\_i1'] **=** df**.**iloc[:, 7]**.**apply(**lambda** x: N **\*** x)

df

df1 **=** pd**.**DataFrame()

df1['n\_i'] **=** spaces\_abs\_freq

df1['n\_i1'] **=** df['n\_i1']

df1['(n\_i - n\_i1)^2 / n\_i1'] **=** df1**.**iloc[:, :]**.**apply(**lambda** x: (x[0] **-** x[1])**\*\***2 **/** x[1], axis**=**1)

X2\_seen **=** df1**.**iloc[:,2]**.**sum()

print("X2\_наб = {}"**.**format(X2\_seen))

print("Число стееней свободы k = {} - 3 = {}"**.**format(len(spaces), len(spaces)**-**3))

print("Для числа степеней свободы 5 X\_крит = {}"**.**format(11.07))

df1

**приложение Г**

**Программа для** **нахождения элементов корреляционного анализа и проверки статистической гипотезы о равенстве коэффициента корреляции нулю.**

**lr4.py**

**import** import\_ipynb

**import** lr3\_x **as** X

**import** lr3\_y **as** Y

**import** pandas **as** pd

**import** numpy **as** np

**import** math

**def** localize\_space(space):

**return** "[{};{})"**.**format(space[0], space[1])

**def** \_value\_in\_space(value, space):

**return** space[0] **<=** value **and** value **<** space[1]

\_source **=** pd**.**read\_csv('dataset.csv', header**=None**)

cor **=** pd**.**DataFrame()

*#cor[0] = [localize\_space(X.spaces[i]) for i in range(len(X.spaces))]*

**for** \_y, space\_y **in** enumerate(Y**.**spaces):

\_col **=** []

**for** space\_x **in** X**.**spaces:

\_count **=** 0

**for** row **in** \_source**.**iterrows():

**if** \_value\_in\_space(row[1][0], space\_x) **and** \_value\_in\_space(row[1][1], space\_y):

\_count **+=** 1

\_col**.**append(\_count)

cor[\_y] **=** \_col

cor

\_rowsIndex **=** pd**.**Index([**\***[localize\_space(i) **for** i **in** X**.**spaces], "N"], name**=**"X")

\_colsIndex **=** pd**.**Index([**\***[localize\_space(i) **for** i **in** Y**.**spaces], "N"], name**=**"Y")

cor1 **=** cor**.**copy()

cor1[len(cor1)] **=** [cor1**.**iloc[i,:]**.**sum() **for** i **in** range(len(X**.**spaces))]

\_sums **=** [cor1**.**iloc[:,i]**.**sum() **for** i **in** range(len(Y**.**spaces) **+** 1)]

cor1 **=** pd**.**concat([cor1, pd**.**DataFrame([\_sums])])

cor1**.**index **=** \_rowsIndex

cor1**.**columns **=** \_colsIndex

cor1

**import** lr2\_x

**import** lr2\_y

v\_x **=** np**.**array(list(map(int, np**.**round(lr2\_x**.**df["Условные варианты"]))))

v\_y **=** np**.**array(list(map(int, np**.**round(lr2\_y**.**df["Условные варианты"]))))

\_rowsIndex **=** pd**.**Index([**\***v\_x, "n\_u"], name**=**"u")

\_colsIndex **=** pd**.**Index([**\***v\_y, "n\_v"], name**=**"v")

cor2 **=** cor1**.**copy()

cor2**.**index **=** \_rowsIndex

cor2**.**columns **=** \_colsIndex

cor2

cor3 **=** cor2**.**copy()

**for** i **in** range(len(v\_x)):

cor3**.**iloc[i,:**-**1] **\*=** v\_x[i]

**for** i **in** range(len(v\_y)):

cor3**.**iloc[:**-**1,i] **\*=** v\_y[i]

cor3**.**iloc[:,**-**1] **=** cor3**.**iloc[:,:**-**1]**.**sum(axis**=**1)

cor3**.**iloc[**-**1,:] **=** cor3**.**iloc[:**-**1,:]**.**sum(axis**=**0)

cor3

\_N **=** lr2\_x**.**N

\_sum **=** cor3["n\_v"]["n\_u"]

\_counts\_x **=** np**.**array(list(cor2**.**iloc[:**-**1,**-**1]))

\_counts\_y **=** np**.**array(list(cor2**.**iloc[**-**1,:**-**1]))

\_x\_mean **=** sum(\_counts\_x **\*** v\_x) **/** \_N

\_y\_mean **=** sum(\_counts\_y **\*** v\_y) **/** \_N

\_x\_s **=** np**.**sqrt((sum(\_counts\_x **\*** v\_x**\*\***2) **/** \_N **-** \_x\_mean**\*\***2))

\_y\_s **=** np**.**sqrt((sum(\_counts\_y **\*** v\_y**\*\***2) **/** \_N **-** \_y\_mean**\*\***2))

r\_variants **=** (\_sum **-** \_N **\*** \_x\_mean **\*** \_y\_mean) **/** (\_N **\*** \_x\_s **\*** \_y\_s)

r\_variants

**def** space\_middle(space):

**return** (space[1] **-** space[0]) **/** 2 **+** space[0]

\_middles\_x **=** list(map(space\_middle, X**.**spaces))

\_middles\_y **=** list(map(space\_middle, Y**.**spaces))

cor4 **=** cor2**.**copy()

**for** i **in** range(len(v\_x)):

cor4**.**iloc[i,:**-**1] **\*=** \_middles\_x[i]

**for** i **in** range(len(v\_y)):

cor4**.**iloc[:**-**1,i] **\*=** \_middles\_y[i]

cor4**.**iloc[:,**-**1] **=** cor4**.**iloc[:,:**-**1]**.**sum(axis**=**1)

cor4**.**iloc[**-**1,:] **=** cor4**.**iloc[:**-**1,:]**.**sum(axis**=**0)

\_rowsIndex **=** pd**.**Index([**\***\_middles\_x, "n\_u"], name**=**"u")

\_colsIndex **=** pd**.**Index([**\***\_middles\_y, "n\_v"], name**=**"v")

cor4**.**index **=** \_rowsIndex

cor4**.**columns **=** \_colsIndex

\_N **=** lr2\_x**.**N

\_sum **=** cor4["n\_v"]["n\_u"]

r\_standard **=** (\_sum **-** \_N **\*** lr2\_x**.**x\_middle **\*** lr2\_y**.**x\_middle) **/** (\_N **\*** math**.**sqrt(lr2\_x**.**S2) **\*** math**.**sqrt(lr2\_y**.**S2))

r\_standard

**from** scipy.stats **import** norm

\_N **=** X**.**N

**def** \_trust\_interval(\_g):

\_q **=** 1 **-** \_g

\_l **=** norm**.**ppf(1 **-** \_q**/**2)

\_z **=** 0.5 **\*** np**.**log((1 **+** r\_standard) **/** (1 **-** r\_standard))

\_sigma\_z **=** 1 **/** np**.**sqrt(\_N **-** 3)

\_z\_space **=** [\_z **-** \_l **\*** \_sigma\_z, \_z **+** \_l **\*** \_sigma\_z]

\_r\_space **=** np**.**tanh(\_z\_space)

print("Доверительный интервал для доверительной вероятности {}: \n\t{} < z < {}, \n\t{} < r < {}"**.**format(\_g, **\***\_z\_space, **\***\_r\_space))

\_trust\_interval(0.95)

\_trust\_interval(0.99)

**from** scipy.stats **import** t

\_a **=** 0.05

\_N **=** X**.**N

\_t\_obs **=** r\_standard **\*** np**.**sqrt((\_N**-**2) **/** (1 **-** r\_standard**\*\***2))

\_t\_crit **=** t**.**ppf(1 **-** \_a**/**2, df**=**(\_N**-**2))

print("Если T\_наблюдаемое < T\_критическое, то гипотеза принимается. В нашем случае {})

**приложение Д**

**Программа для нахождения элементов регрессионного анализа и построения выборочные прямых среднеквадратической регрессии, поиска корреляционного отношения.**

**lr5.py**

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** sys**,** os

saved\_stdout **=** sys**.**stdout

sys**.**stdout **=** open(os**.**devnull, 'w')

**import** import\_ipynb

**from** lr4 **import** r\_standard **as** r, cor, space\_middle

**import** lr3\_x **as** X

**import** lr3\_y **as** Y

sys**.**stdout **=** saved\_stdout;

source **=** pd**.**read\_csv('dataset.csv', header**=None**)

N **=** len(source)

source**.**head()

params **=** pd**.**DataFrame(data**=**{

"mean": [X**.**x\_middle, Y**.**x\_middle],

"variance": [X**.**disp, Y**.**disp],

"sd": [X**.**S, Y**.**S],

}, index**=**["u","v"])

print(params)

plt**.**scatter(source[0], source[1], alpha**=**0.7, color**=**"b")

plt**.**xlabel("u")

plt**.**ylabel("v")

plt**.**title("Диграмма рассеяния")

plt**.**show()

params["reg\_coeff"] **=** [

r **\*** params**.**loc["u", "sd"] **/** params**.**loc["v", "sd"],

r **\*** params**.**loc["v", "sd"] **/** params**.**loc["u", "sd"]

]

print(params)

**def** reg\_X(y):

**return** params**.**loc["u", "mean"] **+** params**.**loc["u", "reg\_coeff"] **\*** (y **-** params**.**loc["v", "mean"])

**def** reg\_Y(x):

**return** params**.**loc["v", "mean"] **+** params**.**loc["v", "reg\_coeff"] **\*** (x **-** params**.**loc["u", "mean"])

plt**.**scatter(source[0], source[1], alpha**=**0.7, color**=**"y")

plt**.**plot(source[0]**.**sort\_values(), reg\_Y(source[0]**.**sort\_values()), label**=**"v(u)", color**=**"r")

plt**.**plot(reg\_X(source[1]**.**sort\_values()), source[1]**.**sort\_values(), label**=**"u(v)", color**=**"b")

plt**.**plot(params**.**loc["u", "mean"], params**.**loc["v", "mean"], color**=**"k", marker**=**"o", mew**=**2, ms**=**12)

plt**.**xlabel("u")

plt**.**ylabel("v")

plt**.**title("Прямые среднеквадратической регрессии")

plt**.**legend()

plt**.**show()

params["err\_var"] **=** params["sd"]**\*\***2 **\*** (1 **-** r**\*\***2)

print(params)

cor1 **=** cor**.**copy()

u\_middles **=** list(map(**lambda** x: round(x, 1), map(space\_middle, X**.**spaces)))

v\_middles **=** list(map(**lambda** x: round(x, 1), map(space\_middle, Y**.**spaces)))

N\_u **=** cor1**.**sum(axis**=**1)

N\_v **=** cor1**.**sum(axis**=**0)

mean\_u **=** np**.**sum(cor1 **\*** v\_middles, axis**=**1) **/** N\_u

mean\_v **=** np**.**sum(cor1**.**transpose() **\*** u\_middles, axis**=**1) **/** N\_v

cor1["N\_u"] **=** N\_u

cor1["mean\_u"] **=** mean\_u

cor1 **=** pd**.**concat([cor1, pd**.**DataFrame([N\_v, mean\_v])])

cor1**.**columns **=** pd**.**Index([**\***v\_middles, "N\_u", "mean\_u"], name**=**"v")

cor1**.**index **=** pd**.**Index([**\***u\_middles, "N\_v", "mean\_v"], name**=**"u")

cor1 **=** cor1**.**round(1)

cor1**.**iloc[:**-**2,:**-**2] **=** cor1**.**iloc[:**-**2,:**-**2]**.**round()

cor1

var\_v\_u **=** np**.**sum(cor **\*** (

np**.**tile(v\_middles, (len(mean\_u), 1)) **-**

np**.**tile(mean\_u, (len(v\_middles), 1))**.**transpose()

)**\*\***2,

axis**=**1) **/** N\_u

var\_v\_in **=** sum(var\_v\_u **\*** N\_u) **/** N

var\_v\_across **=** sum(N\_u **\*** (mean\_u **-** params**.**loc["v", "mean"]) **\*\*** 2) **/** N

var\_u\_v **=** np**.**sum(cor**.**transpose() **\*** (

np**.**tile(u\_middles, (len(mean\_v), 1)) **-**

np**.**tile(mean\_v, (len(u\_middles), 1))**.**transpose()

)**\*\***2,

axis**=**1) **/** N\_v

var\_u\_in **=** sum(var\_u\_v **\*** N\_v) **/** N

var\_u\_across **=** sum(N\_v **\*** (mean\_v **-** params**.**loc["u", "mean"]) **\*\*** 2) **/** N

variances **=** pd**.**DataFrame(data**=**{

"within groups": [var\_v\_in, var\_u\_in],

"across groups": [var\_v\_across, var\_u\_across],

"within + across": [var\_v\_in **+** var\_v\_across, var\_u\_in **+** var\_u\_across],

"sample variance": [params**.**loc["v", "variance"], params**.**loc["u", "variance"]]

}, index**=**["v", "u"])

variances

eta\_v\_u **=** np**.**sqrt(variances**.**loc['v', "across groups"] **/** params**.**loc["v", "variance"])

eta\_u\_v **=** np**.**sqrt(variances**.**loc['u', "across groups"] **/** params**.**loc["u", "variance"])

print(eta\_v\_u)

print(eta\_u\_v)

left\_v\_u **=** [

[sum(source[0]**\*\***4), sum(source[0]**\*\***3), sum(source[0]**\*\***2)],

[sum(source[0]**\*\***3), sum(source[0]**\*\***2), sum(source[0])],

[sum(source[0]**\*\***2), sum(source[0]), N]]

right\_v\_u **=** [

sum(source[1] **\*** source[0]**\*\***2),

sum(source[1] **\*** source[0]),

sum(source[1])]

a\_v, b\_v, c\_v **=** np**.**linalg**.**solve(left\_v\_u, right\_v\_u)

left\_u\_v **=** [

[sum(source[1]**\*\***4), sum(source[1]**\*\***3), sum(source[1]**\*\***2)],

[sum(source[1]**\*\***3), sum(source[1]**\*\***2), sum(source[1])],

[sum(source[1]**\*\***2), sum(source[1]), N]]

right\_u\_v **=** [

sum(source[0] **\*** source[1]**\*\***2),

sum(source[0] **\*** source[1]),

sum(source[0])]

a\_u, b\_u, c\_u **=** np**.**linalg**.**solve(left\_u\_v, right\_u\_v)

parab\_coeffs **=** pd**.**DataFrame(data**=**{

"a": [a\_v, a\_u],

"b": [b\_v, b\_u],

"c": [c\_v, c\_u]},

index**=**["v(u)", "u(v)"])

parab\_coeffs

parab\_v **=** **lambda** u: a\_v **\*** u**\*\***2 **+** b\_v **\*** u **+** c\_v

parab\_u **=** **lambda** v: a\_u **\*** v**\*\***2 **+** b\_u **\*** v **+** c\_u

plt**.**scatter(source[0], source[1], alpha**=**0.7, color**=**"y")

plt**.**plot(source[0]**.**sort\_values(), parab\_v(source[0]**.**sort\_values()), label**=**"$v(u) = a u^2 + b u + c$", color**=**"r")

plt**.**plot(parab\_u(source[1]**.**sort\_values()), source[1]**.**sort\_values(), label**=**"$u(v) = a v^2 + b v + c$", color**=**"b")

plt**.**plot(params**.**loc["u", "mean"], params**.**loc["v", "mean"], color**=**"k", marker**=**"o", mew**=**2, ms**=**12)

plt**.**xlabel("u")

plt**.**ylabel("v")

plt**.**title("Корреляционные кривые параболического вида")

plt**.**legend()

plt**.**show()

**приложение Е**

**Программа для метода k-cредних**

**lr6.py**

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** import\_ipynb

**from** lr4 **import** r\_standard **as** r, cor, space\_middle

**import** lr3\_x **as** X

**import** lr3\_y **as** Y

source **=** pd**.**read\_csv('dataset.csv', header**=None**)

N **=** len(source)

source**.**head()

normalized **=** source**.**copy()

normalized[0] **=** (normalized[0] **-** normalized[0]**.**mean()) **/** normalized[0]**.**std()

normalized[1] **=** (normalized[1] **-** normalized[1]**.**mean()) **/** normalized[1]**.**std()

normalized

plt**.**scatter(normalized[0], normalized[1], alpha**=**0.5, color**=**"gray")

plt**.**xlabel("u")

plt**.**ylabel("v")

plt**.**title("Диграмма рассеяния нормализованных величин")

plt**.**show()

k **=** int(np**.**floor(np**.**sqrt(N**/**2)))

centers **=** normalized**.**sample(n**=**k, random\_state**=**2298)**.**reset\_index(drop**=True**)

centers

plt**.**scatter(normalized[0], normalized[1], alpha**=**0.3, color**=**"gray")

**for** i **in** range(k):

plt**.**plot(centers**.**iloc[i, 0], centers**.**iloc[i, 1], marker**=**"x", mew**=**3, ms**=**6)

plt**.**xlabel("u")

plt**.**ylabel("v")

plt**.**title("Диграмма рассеяния нормализованных величин и начальные центры кластеров")

plt**.**show()

**def** find\_cluster\_index(\_point, \_centers):

**return** np**.**argmin(np**.**linalg**.**norm(\_point **-** \_centers, axis**=**1))

**def** calculate\_centers(\_centers, \_clusters):

\_centers **=** \_centers**.**copy()

**for** i **in** range(len(\_clusters)):

**if** len(\_clusters[i]) **>** 0:

\_centers**.**loc[i, :] **=** np**.**mean(\_clusters[i], axis**=**0)

**return** \_centers

**def** make\_clusters(\_data, \_centers, \_mode**=**'after\_each'):

\_clusters **=** [[] **for** \_ **in** range(len(\_centers))]

**for** i **in** range(len(\_data)):

\_clust\_ind **=** find\_cluster\_index(\_data**.**loc[i, :], \_centers)

\_clusters[\_clust\_ind]**.**append(list(\_data**.**loc[i, :]))

**if** \_mode **==** 'after\_each':

\_centers **=** calculate\_centers(\_centers, \_clusters)

**if** \_mode **==** 'after\_all':

\_centers **=** calculate\_centers(\_centers, \_clusters)

**return** \_clusters, \_centers

**def** f1(\_clusters, \_centers):

\_sum **=** 0

**for** i **in** range(len(\_clusters)):

\_sum **+=** np**.**sum((np**.**array(\_clusters[i]) **-** np**.**array(\_centers**.**loc[i, :]))**\*\***2)

**return** \_sum

**def** f2(\_clusters):

\_sum **=** 0

**for** i **in** range(len(\_clusters)):

**for** j **in** range(len(\_clusters[i])):

**for** k **in** range(j**+**1, len(\_clusters[i])):

\_sum **+=** np**.**linalg**.**norm(np**.**array(\_clusters[i][j]) **-** np**.**array(\_clusters[i][k]))

**return** \_sum

**def** f3(\_clusters, \_centers):

\_sum **=** 0

**for** i **in** range(len(\_clusters)):

\_sum **+=** np**.**sum((np**.**array(\_clusters[i]) **-** np**.**array(\_centers**.**loc[i, :]))**\*\***2) **/** len(\_clusters[i])

**return** \_sum

**def** draw\_clusters(\_clusters, \_centers, \_index):

**for** i **in** range(len(\_clusters)):

plt**.**scatter(np**.**array(\_clusters[i])[:, 0], np**.**array(\_clusters[i])[:, 1], alpha**=**0.5)

plt**.**plot(\_centers**.**loc[i, 0], \_centers**.**loc[i, 1], marker**=**"x", mew**=**3, ms**=**6)

plt**.**xlabel("u")

plt**.**ylabel("v")

plt**.**title("Разбиение на кластеры - итерация " **+** str(\_index))

plt**.**show()

centers1 **=** centers**.**copy()

clusters1 **=** **None**

next\_clusters1, centers1 **=** make\_clusters(normalized, centers1, 'after\_each')

counter1 **=** 1

**while** (clusters1 **!=** next\_clusters1):

clusters1 **=** next\_clusters1**.**copy()

print("Итерация ", counter1)

print('F1: {}, F2: {}, F3: {}'**.**format(

f1(next\_clusters1, centers1),

f2(next\_clusters1),

f3(next\_clusters1, centers1)))

draw\_clusters(next\_clusters1, centers1, counter1)

next\_clusters1, centers1 **=** make\_clusters(normalized, centers1, 'after\_each')

counter1 **+=** 1

centers2 **=** centers**.**copy()

clusters2 **=** **None**

next\_clusters2, centers2 **=** make\_clusters(normalized, centers2, 'after\_all')

counter2 **=** 1

**while** (clusters2 **!=** next\_clusters2):

clusters2 **=** next\_clusters2**.**copy()

print("Итерация ", counter2)

print('F1: {}, F2: {}, F3: {}'**.**format(

f1(next\_clusters2, centers2),

f2(next\_clusters2),

f3(next\_clusters2, centers2)))

draw\_clusters(next\_clusters2, centers2, counter2)

next\_clusters2, centers2 **=** make\_clusters(normalized, centers2, 'after\_all')

counter2 **+=** 1

**приложение Ж**

**Программа для метода поиска сгущений.**

**lr7.py**

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**from** math **import** **\***

source **=** pd**.**read\_csv('dataset.csv', header**=None**)

U **=** 'u'

V **=** 'v'

source**.**columns **=** [U, V]

N **=** len(source)

source**.**head()

normalized **=** source**.**copy()

normalized[U] **=** (normalized[U] **-** normalized[U]**.**mean()) **/** normalized[U]**.**std()

normalized[V] **=** (normalized[V] **-** normalized[V]**.**mean()) **/** normalized[V]**.**std()

normalized

X\_LIM **=** (np**.**min(normalized[U]) **-** 0.2, np**.**max(normalized[U]) **+** 0.2)

Y\_LIM **=** (np**.**min(normalized[V]) **-** 0.2, np**.**max(normalized[V]) **+** 0.2)

LIM **=** (np**.**min([**\***X\_LIM, **\***Y\_LIM]), np**.**max([**\***X\_LIM, **\***Y\_LIM]))

plt**.**scatter(normalized[U], normalized[V], alpha**=**0.8, color**=**"gray")

plt**.**xlabel(U)

plt**.**ylabel(V)

plt**.**title("Диграмма рассеяния нормализованных величин")

plt**.**xlim(LIM)

plt**.**ylim(LIM)

plt**.**gca()**.**set\_aspect('equal')

plt**.**show()

**def** \_find\_point\_distances(\_points, \_point):

**return** np**.**linalg**.**norm(

\_points[(\_points **!=** \_point)**.**apply(**lambda** x: x**.**any(), axis**=**1)] **-** \_point,

axis**=**1)

**def** find\_min\_distance(\_points):

**return** np**.**min(\_points**.**apply(

**lambda** x: np**.**min(\_find\_point\_distances(\_points, x)),

axis**=**1))

**def** find\_max\_distance(\_points):

**return** np**.**max(\_points**.**apply(

**lambda** x: np**.**max(\_find\_point\_distances(\_points, x)),

axis**=**1))

**def** find\_initial\_center\_index(\_points):

**return** np**.**argmin(\_points**.**apply(

**lambda** x: np**.**sum(\_find\_point\_distances(\_points, x)),

axis**=**1))

**class** Cluster(object):

**def** \_\_init\_\_(self, \_points, \_radius):

self**.**points **=** \_points

self**.**center **=** \_points**.**iloc[find\_initial\_center\_index(\_points), :]

self**.**radius **=** \_radius

self**.**center\_history **=** [self**.**center]

**def** correct\_center(self):

\_inner\_points **=** np**.**linalg**.**norm(self**.**points **-** self**.**center, axis**=**1) \

**<=** self**.**radius

\_new\_center **=** self**.**points[\_inner\_points]**.**mean(axis**=**0)

**if** (\_new\_center **!=** self**.**center)**.**any():

self**.**center **=** \_new\_center

self**.**center\_history**.**append(self**.**center)

**return** **True**

**return** **False**

**def** filter\_inner\_points(self, \_center **=** **None**):

**if** \_center **is** **None**:

\_center **=** self**.**center

\_inner\_points **=** np**.**linalg**.**norm(self**.**points **-** \_center, axis**=**1) \

**<** self**.**radius

**return** self**.**points[\_inner\_points]

**def** filter\_outer\_points(self, \_center**=None**):

**if** \_center **is** **None**:

\_center **=** self**.**center

\_inner\_points **=** np**.**linalg**.**norm(self**.**points **-** \_center, axis**=**1) \

**<** self**.**radius

**return** self**.**points[**~**\_inner\_points]

**def** f1(self):

\_inner\_points **=** self**.**filter\_inner\_points()

**return** np**.**sum(np**.**linalg**.**norm(\_inner\_points **-** self**.**center, axis**=**1) **\*\*** 2)

**def** f2(self):

\_inner\_points **=** self**.**filter\_inner\_points()

\_sum **=** 0

**for** i **in** range(len(\_inner\_points) **-** 1):

\_sum **+=** np**.**sum(np**.**linalg**.**norm(\_inner\_points**.**iloc[i**+**1:,:] **-** \_inner\_points**.**iloc[i,:], axis**=**1))

**return** \_sum

**def** f3(self):

\_inner\_points **=** self**.**filter\_inner\_points()

**return** np**.**sum((np**.**array(\_inner\_points) **-** np**.**array(self**.**center)) **\*\*** 2) **/** len(\_inner\_points)

**def** \_\_str\_\_(self):

**return** '{{Center({:.4f}; {:.4f}), Points({})}}'**.**format(**\***list(self**.**center), len(self**.**filter\_inner\_points()))

**def** clusterise(\_points, \_r):

\_clusters **=** []

**while** len(\_points) **>** 0:

\_cluster **=** Cluster(\_points, \_r)

**while** \_cluster**.**correct\_center():

**pass**

\_clusters**.**append(\_cluster)

\_points **=** \_cluster**.**filter\_outer\_points()

**return** \_clusters

In [183]:

R\_MIN **=** find\_min\_distance(normalized)

R\_MAX **=** find\_max\_distance(normalized)

print("Минимальный радиус: {}, максимальный радиус: {}"**.**format(R\_MIN, R\_MAX))

**def** draw\_cluster\_history(\_cluster):

\_history **=** \_cluster**.**center\_history

**for** center **in** \_history:

\_ax **=** plt**.**gca()

\_inner\_points **=** \_cluster**.**filter\_inner\_points(center)

\_outer\_points **=** \_cluster**.**filter\_outer\_points(center)

\_ax**.**scatter(\_inner\_points[U],

\_inner\_points[V], alpha**=**0.5, color**=**"gray")

\_ax**.**scatter(\_outer\_points[U],

\_outer\_points[V], alpha**=**0.2, color**=**"gray")

\_ax**.**add\_patch(plt**.**Circle(list(center), \_cluster**.**radius, alpha**=**0.4))

\_ax**.**plot(center[U], center[V], marker**=**"o", ms**=**6)

\_ax**.**set\_xlim(LIM)

\_ax**.**set\_ylim(LIM)

\_ax**.**set\_aspect('equal')

plt**.**show()

**def** visualize\_clusters\_history(\_clusters):

print("Сформировано кластеров: {}"**.**format(len(\_clusters)))

i **=** 1

**for** cluster **in** \_clusters:

print("Формирование {} кластера"**.**format(i))

draw\_cluster\_history(cluster)

i **+=** 1

**def** visualize\_clusters(\_clusters):

plt**.**title("Результат кластеризации")

**for** cluster **in** \_clusters:

\_inner\_points **=** cluster**.**filter\_inner\_points()

plt**.**scatter(\_inner\_points[U], \_inner\_points[V], alpha**=**0.3)

plt**.**plot(cluster**.**center[U], cluster**.**center[V],

marker**=**"x", mew**=**3, ms**=**6)

plt**.**xlim(LIM)

plt**.**ylim(LIM)

plt**.**gca()**.**set\_aspect('equal')

plt**.**show()

**def** show\_clusters\_quality(\_clusters):

\_f1 **=** np**.**sum(list(map(**lambda** x: x**.**f1(), \_clusters)))

\_f2 **=** np**.**sum(list(map(**lambda** x: x**.**f2(), \_clusters)))

\_f3 **=** np**.**sum(list(map(**lambda** x: x**.**f3(), \_clusters)))

print("F1: {}, F2: {}, F3: {}"**.**format(\_f1, \_f2, \_f3))

STEP **=** (R\_MAX **-** R\_MIN) **/** 60

r **=** R\_MIN **+** STEP

**while** r **<** R\_MAX:

print("Выбранный радиус кластера: {}"**.**format(r))

\_clusters **=** clusterise(normalized, r)

*#visualize\_clusters\_history(\_clusters)*

print("Число кластеров: {}"**.**format(len(\_clusters)))

show\_clusters\_quality(\_clusters)

visualize\_clusters(\_clusters)

r **+=** STEP

**if** len(\_clusters) **==** 1:

**break**