**Федеральное государственное образовательное**

**бюджетное учреждение**

**высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ**

**ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ**

**ФЕДЕРАЦИИ»**

**(Финансовый университет)**

**Факультет**

**информационных технологий и анализа больших данных**

**Кафедра «Бизнес-информатика»**

**Домашнее задание № 5**

«Решение задач методом наименьших квадратов.»

Студенты группы БИ20-8:

Луканина Полина

Аверкин Никита

Филимонова Арина

Совин Владимир

Горшков Георгий

Киселева Евгения

Руководитель:

Аксенов Дмитрий Андреевич

**Москва 2022**

### Оглавление

[Оглавление 2](#_Toc102046239)

[1. Постановка задачи (физическая модель) 4](#_Toc102046240)

[2. Математическая модель 5](#_Toc102046241)

[3. Алгоритмы решения задачи 7](#_Toc102046242)

[3.1. Решение задачи о нахождении выигрышной стратегии для 7](#_Toc102046243)

[антагонистической игры 7](#_Toc102046244)

[3.1.1. Описание входных данных. 7](#_Toc102046245)

[3.1.2. Описание алгоритма решения 8](#_Toc102046246)

[3.1.3. Описание выходных данных 9](#_Toc102046247)

[4. Варианты использования системы 12](#_Toc102046248)

[4.1. ВИ 13](#_Toc102046249)

[5. Архитектура решения 15](#_Toc102046250)

[5.1 Алгоритм 1 15](#_Toc102046251)

[5.2 Функции обработки информации 16](#_Toc102046252)

[5.3 Функции вывода информации 31](#_Toc102046253)

[6 Тестирование 31](#_Toc102046254)

[6.1. Тестирование Датасета №1: 31](#_Toc102046255)

[6.1.1. Метод Python: 31](#_Toc102046256)

[6.1.2. Метод Excel: 35](#_Toc102046257)

[6.2. Тестирование Датасета №2: 38](#_Toc102046258)

[6.2.1. Метод Python: 38](#_Toc102046259)

[6.2.2. Метод Excel: 42](#_Toc102046260)

[6.3. Тестирование Датасета №3: 44](#_Toc102046261)

[6.3.1. Метод Python: 44](#_Toc102046262)

[6.3.2. Метод Excel: 48](#_Toc102046263)

[6.4. Тестирование Датасета №4: 51](#_Toc102046264)

[6.4.1. Метод Python: 51](#_Toc102046265)

[6.4.2. Метод Excel: 54](#_Toc102046266)

[6.5. Тестирование Датасета №5: 57](#_Toc102046267)

[6.5.1. Метод Python: 57](#_Toc102046268)

[6.5.2. Метод Excel: 60](#_Toc102046269)

[7. Заключение 63](#_Toc102046270)

# Постановка задачи (физическая модель)

Одна юридическая компания помогает с процедурой банкротства для физических лиц и индивидуальных предпринимателей. Ведь порой это единственный способ погасить долг перед кредитором.

Руководители компании хотят выяснить, тенденцию количества банкротств в России в ближайшие несколько лет и основываясь на этих данных рассчитать нужно ли сокращать/увеличивать количество юристов в компании и на сколько. Мы должны им в этом помочь.

По нашему запросу нам предоставили отчет за последние 7 лет по количеству банкротств.

Таблица 1. Отчет о банкротствах за 7 лет

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 |
| Количество граждан и индивидуальных предпринимателей, признанных банкротами | 870 | 19 574 | 29 827 | 43 984 | 68 980 | 163 235 | 192 846 |

# Математическая модель

Одной из задач аппроксимации является описание в виде функциональной зависимости связи между величинами, заданными таблично или в виде набора точек (xk, yk), где k = 0, ..., n. При аппроксимации желательно получить относительно простую функциональную зависимость, которая позволила бы «сгладить» экспериментальные погрешности. Эта функциональная зависимость должна с достаточной точностью соответствовать исходной табличной зависимости.

В качестве критерия точности чаще всего используют критерий наименьших квадратов.

Так, требуется найти аппроксимирующую функцию y = f (x, a, b, …) такую, чтобы график проходил как можно ближе к экспериментальным точкам, для чего необходимо найти коэффициенты a, b, …

Для каждой точки, описываемой 𝑥𝑘 и 𝑦𝑘, применяется следующая функция – наименьшее расстояние до каждой точки:

Для нахождения значений 𝑎 и 𝑏 запишем функцию 𝑓(𝑎, 𝑏):

Найдем частные производные 𝑓 (𝑎, 𝑏) по 𝑎 и по 𝑏:

Раскроем скобки внутри суммы и вынесем множители за скобки:

Приравняем частные производные к 0:

Запишем среднее арифметическое 𝑥𝑘 и yk, а также 𝑥𝑘 2 и 𝑦𝑘𝑥𝑘 в виде:

Подставляя средние арифметические в уравнения, умножаем на 𝑛 и получаем:

Решая систему линейных уравнений относительно 𝑎 и 𝑏, получаем ответ:

Для дальнейшего выстраивания доверительных интервалов и определения результатов прогнозирования используется дисперсия, которая вычисляется по формуле:

Дисперсия описывает разброс данных в выборке. Если дисперсия вокруг тренда невелика, то результатам прогнозирования можно доверять. Если дисперсия велика – то вероятность того, что прогноз сбудется, мала.

Ошибка прогноза вычисляется из прогнозируемого диапазона и находится как квадратный корень из дисперсии:

# Алгоритмы решения задачи

Алгоритмы решения реализованы с помощью программного кода в Python.

## 3.1. Решение задачи о нахождении выигрышной стратегии для

## антагонистической игры

### 3.1.1. Описание входных данных.

Формат входных данных определяется тем, что программа принимает только CSV файл, который содержит в себе массивы точек в формате [] и [x, y].

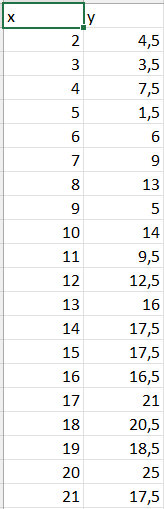


Рисунок 1. Пример входных данных в CSV файле



Рисунок 2. Пример заполнения пути к файлу

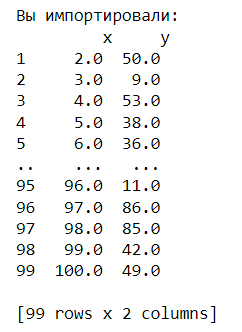


Рисунок 3. Пример импортированных данных в программу

### 3.1.2. Описание алгоритма решения

После того как данные введены, программе необходимо преобразовать данные для дальнейшего использования.

Шаг 1: рассчитываем среднее значение массива.

Шаг 2: рассчитываем ковариации по выборе.

Шаг 3: рассчитываем дисперсию по выборке.

Шаг 4: рассчитываем аппроксимирующую линию и коэффициенты.

Шаг 5: формируем уравнение линейной аппроксимации.

Шаг 6: строим график линейной аппроксимации и строим прямые доверительные интервалы.

Шаг 7: рассчитываем дисперсию по выборке.

Шаг 8: рассчитываем свободные члены.

Шаг 9: формируем массив из суммы степеней переменной.

Шаг 10: рассчитываем коэффициенты с квадратичной аппроксимирующей функцией.

Шаг 11: формируем уравнения квадратичной аппроксимации.

Шаг 12: рассчитываем аппроксимирующей прямой.

Шаг 13: строим график квадратичной аппроксимации и прямые доверительные интервалы.

Шаг 14: рассчитываем квадратичную аппроксимацию и строим график функции.

### 3.1.3. Описание выходных данных

В конце программа выдает выборку с выбросами и для нее аппроксимирующую линейную функцию, аппроксимирующую квадратную функцию, а также выборку без выбросов и для нее аппроксимирующую линейную функцию, аппроксимирующую квадратную функцию. Более того, на выходе вы получите 5 графиков: график аппроксимации, графики линейной и квадратичной аппроксимации с выбросами и также график линейной и квадратичной аппроксимации без выбросов.

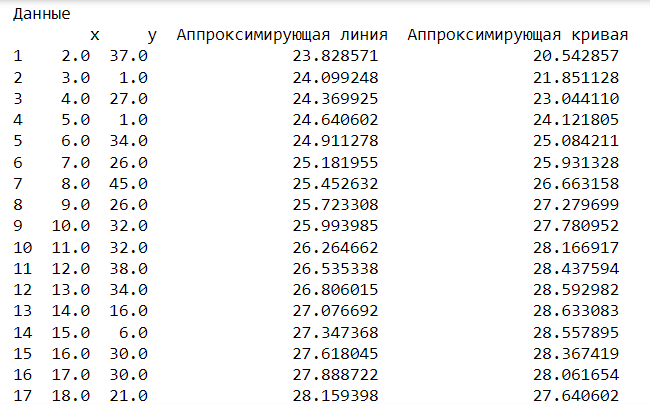


Рисунок 4. Пример выходных данных

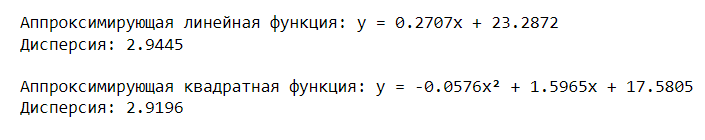


Рисунок 5. Линейная и квадратная функция

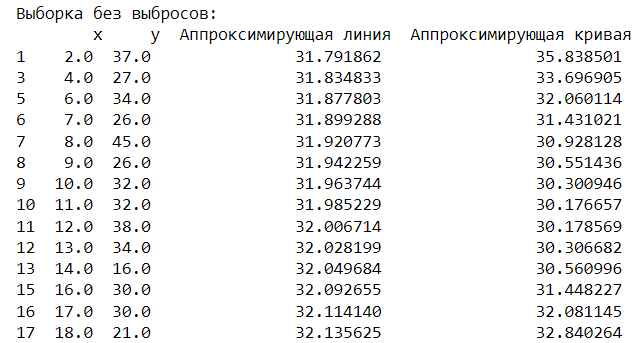


Рисунок 6. Выборка без выбросов



Рисунок 7. Линейная и квадратная функция

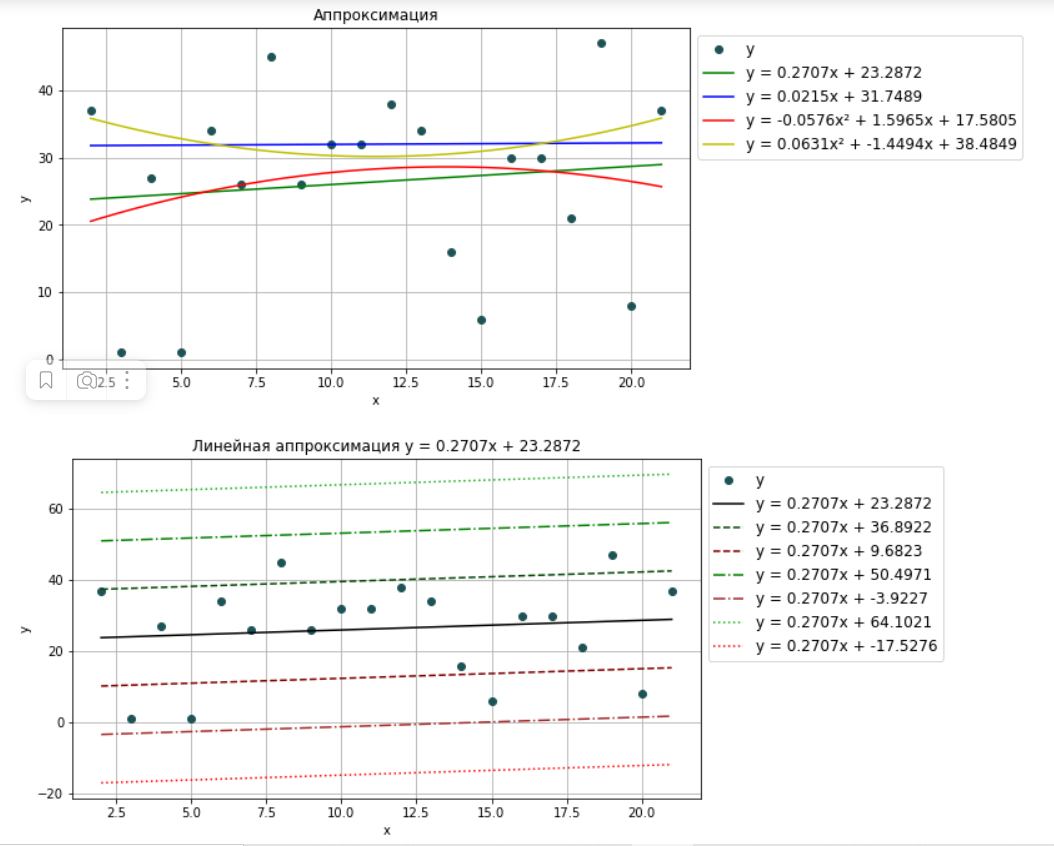


Рисунок 8. Визуализация

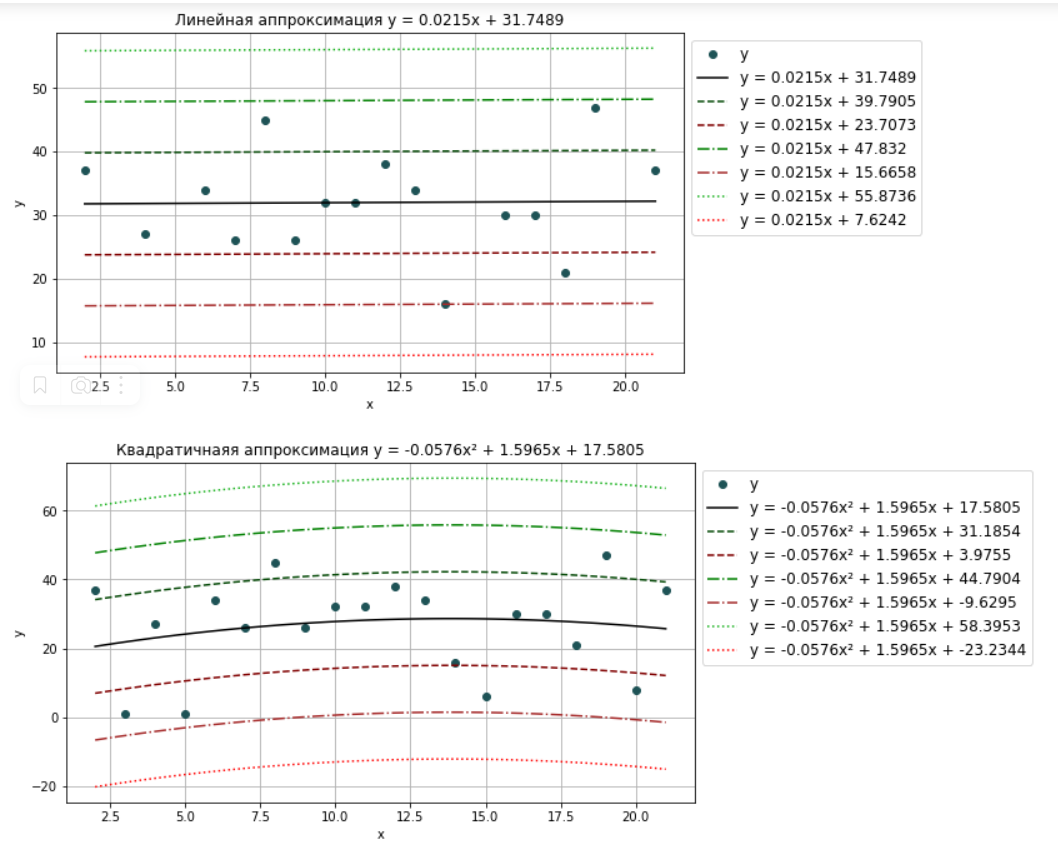


Рисунок 9. Визуализация

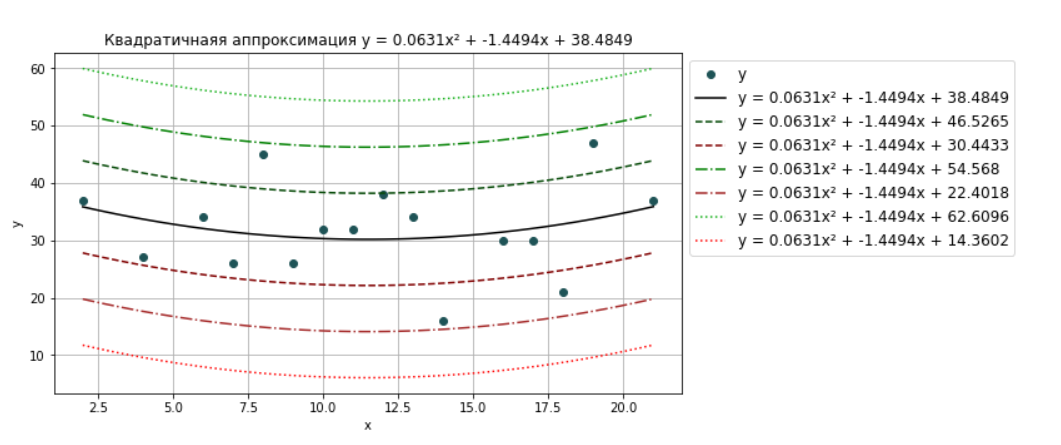


Рисунок 10. Визуализация

# Варианты использования системы

В нашей системе есть один вариант использования.

## 4.1. ВИ

Данный вариант использования включает в себя ввод данных с помощью файла csv. Для того, чтобы ввести путь к файлу необходимо запустить программу.

После этого появляется окно, в котором вводим путь к csv файлу. Например:



Рисунок 11. Выбор местонахождения csv файла

Далее программа показывает содержимое файла таким образом:

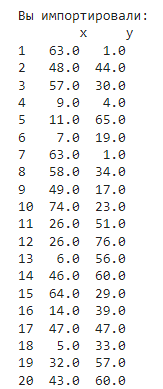


Рисунок . Данные csv файла

Программа преобразует полученные данные и выводит значения аппроксимирующей линейной функции и аппроксимирующей квадратной функции

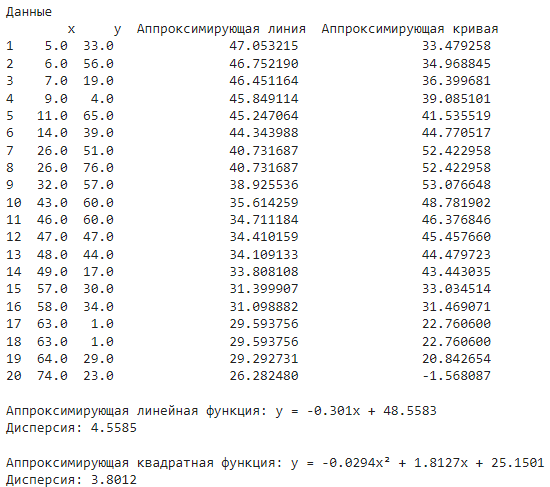


Рисунок . Преобразованные данные

После этого, программа находит точки выброса и исключает их из выборки:



Рисунок . Выборка без выбросов

# Архитектура решения

Для решения задачи использовались методы (функции), которые можно разделить на 3 принципиальных кода.

## Алгоритм 1

**5.1.1.** **Функции считывания информации**

**Входные данные:**

* Нет входных данных

**Выходные данные:**

* A – массив данных в формате data frame.

**Переменные, затрагиваемые в ходе работы:**

* csv –
* A –
* x –
* y –
* columns –
* index –

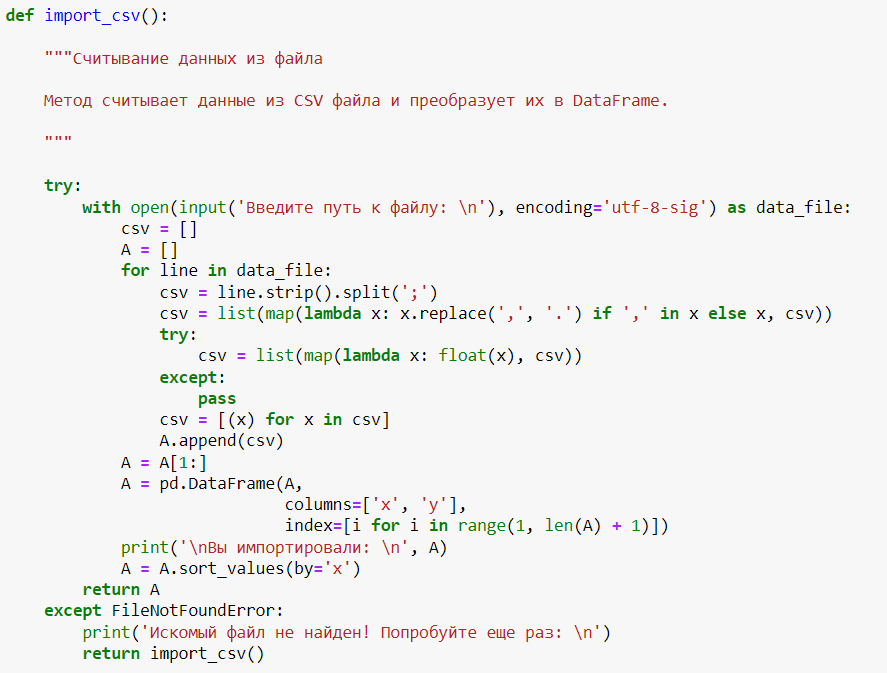


Рисунок 15. Часть программного кода, отвечающего за функцию считывания информации

## 5.2 Функции обработки информации

После того, как вы введете все необходимые данные, программа их получит и начнет первичную обработку.

Поскольку ввод данных осуществляется с помощью файла csv, то для начала файл необходимо открыть и прочитать программе, а затем разделить данные и также занести их в пустой словарь. Более того, программа образует список с названиями колонок массива данных, а также список с названиями строк.

После всех необходимых операций с данными реализуются важные функции, рассчитывающие аппроксимацию линейной и квадратичной функции с выбросами и без них.

В функции mean\_value рассчитывается среднее значение массива с помощью метода mean ().

В функции covariance рассчитывается ковариация по выборке. Это осуществляется за счет вычитания среднего значения массива из передаваемых переменных в функцию.

В функции var\_value происходит расчет дисперсии по выборке.

В функции approxim\_line рассчитывается аппроксимирующая линия и коэффициенты. Более того, метод формирует массив аппроксимирующей линии и добавляет его в виде столбца в таблицу DataFrame.

В функции equation\_linal\_approxim формируется уравнение линейной аппроксимации.

В функции graph\_linal\_approxim осуществляется построение линейной аппроксимации и прямые доверительные интервалы.

В функции dispersion рассчитывается дисперсия по выборке.

В функции free\_value рассчитываются свободные члены.

В функции x\_degree происходит формирование массива и суммы степеней переменной.

В функции matrix осуществляется расчет коэффициентов с квадратичной аппроксимацией функции.

В функции equation\_quadratic\_approxim происходит формирование уравнения квадратичной аппроксимации

В функции approxim\_curve рассчитывается аппроксимирующая кривая.

В функции delete\_release удаляются выбросы. После их удаления функция возвращает выборку без них.

В функции graph\_quadratic\_approxim происходит повторение графика квадратичной аппроксимации и прямые доверительные интервалы. В этой функции заключена вторая fun\_sigma, в которой идет построение линии доверительного интервала.

В функции graph\_all\_func строятся графики линейной и нелинейной аппроксимации в исходном виде и после удаления выбросов. В данной функции заключены еще две: linal – в которой сроятся график функции линейной аппроксимации; quadratic – строится график квадратичной функции. Далее следует подробное описание входных, выходных и переменных, затрагивается в ходе работы.

**Функция mean\_value:**

**Входные данные:**

* A – массив данных в формате DataFrame.

**Выходные данные:**

* Отсутствуют

**Переменные, затрагиваемые в ходе работы:**

* value – переменная, принимающая значения внутри функции.
* A – массив данных в формате DataFrame.

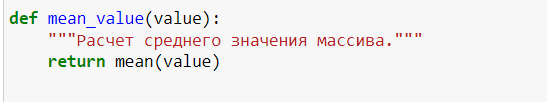
****

Рисунок 16. Часть кода, отвечающая за функцию

**Функция covariance:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* х – переменная, принимающая значения в функции.
* y – переменная, принимающая значения в функции.
* dx – ковариация по первой выборке.
* dy – ковариация по второй выборке.

**Часть программного кода, отвечающая за функцию:**

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 17. Часть программного кода, отвечающая за функцию

**Функция var\_valuе:**

**Входные данные:**

* A – массив данных в формате DataFrame.

**Выходные данные:**

* Отсутствуют

**Переменные, затрагиваемые в ходе работы:**

* value – переменная, принимающая значения внутри функции.
* A – массив данных в формате DataFrame.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 18. Часть кода, отвечающая за функцию

**Функция approxim\_line:**

**Входные данные:**

* Отсутствуют

**Выходные данные:**

* Отсутствуют

**Переменные, затрагиваемые в ходе работы:**

* column\_x\_mean – среднее значение первого столбца массива данных.
* column\_y\_mean - среднее значение второго столбца массива данных.
* k - коэффициент.
* b – коэффициент.
* approxim\_line\_ - значение аппроксимирующей линии.
* num – значение аппроксимирующей линии.
* х – название столбцов массива данных.
* y – название столбцов массива данных.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 19. Часть кода, отвечающая за функцию

**Функция equation\_linal\_approxim:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* b – переменная, принимающая значения в функции.
* k – переменная, принимающая значения в функции.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 20. Часть программного кода, отвечающая за функцию

**Функция fun\_sigma:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* b - переменная, принимающая значения в функции.
* color - переменная, принимающая значения в функции.
* linestyle - переменная, принимающая значения в функции.
* label - переменная, принимающая значения в функции.
* l – переменная, принимающая значения в функции.
* y — это координаты графика функции вида kx + b по оси ординат

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 21. Часть программного кода, отвечающая за функцию

**Функция graph\_linal\_approxim:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* l – переменная, принимающая значения в функции.
* k – переменная, принимающая значения в функции.
* b – переменная, принимающая значения в функции.
* y — это координаты графика функции вида kx + b по оси ординат

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 22. Часть программного кода, отвечающая за функцию

**Функция dispersion:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* num - переменная, принимающая значения в функции.
* column - переменная, принимающая значения в функции.
* sigma – значение дисперсии по выборке.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 23. Часть программного кода, отвечающая за функцию

**Функция x\_degree:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* num - переменная, принимающая значения в функции.

**Изображение выглядит как текст, антенна

Автоматически созданное описание**

Рисунок 24. Часть программного кода, отвечающая за функцию

**Функция matrix :**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* x — это массив, состоящий из сумм степеней x (вызывается функция x\_degree)
* matrix - произведение обратной матрицы matrix и массива (в качестве второго множителя вызывается функция free\_values)
* num - переменная, принимающая значения в функции.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 25. Часть программного кода, отвечающая за функцию

**Функция equation\_quadratic\_approxim:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* а – переменная, принимающая значения в функции.
* b – переменная, принимающая значения в функции.
* c – переменная, принимающая значения в функции.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 26. Часть программного кода, отвечающая за функцию

**Функция approxim\_curve:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* а – константы квадратичной функции вида ax^2 + bx + c.
* b – константы квадратичной функции вида ax^2 + bx + c.
* c – константы квадратичной функции вида ax^2 + bx + c.
* approxim\_curve\_ - значение аппроксимирующей кривой.
* num - переменная, принимающая значения в функции.

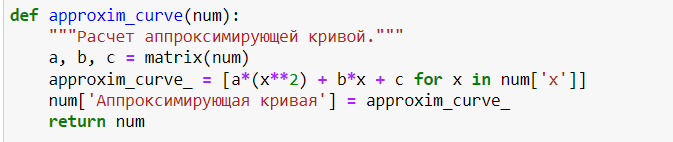
****

Рисунок 27. Часть программного кода, отвечающая за функцию

**Функция delete\_release:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* num – переменная, принимающая значения в функции.
* num\_without\_release – выборка без выбросов.
* upper\_line – аппроксимирующая линия.
* bottom\_line – аппроксимирующая линия.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 28. Часть программного кода, отвечающая за функцию

**Функция graph\_quadratic\_approxim:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* num – переменная, принимающая значения в функции.
* l – переменная, принимающая значения в функции.
* y – координаты графика функции вида ax^2 + bx + c по оси ординат
* а – константы квадратичной функции вида ax^2 + bx + c.
* b – константы квадратичной функции вида ax^2 + bx + c.
* c – константы квадратичной функции вида ax^2 + bx + c.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 29. Часть программного кода, отвечающая за функцию

**Функция graph\_all\_func:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* Num - переменная, принимающая значения в функции.
* k - переменная, принимающая значения в функции.
* b - переменная, принимающая значения в функции.
* num\_without\_release - переменная, принимающая значения в функции.
* k\_cl - переменная, принимающая значения в функции.
* b\_cl - переменная, принимающая значения в функции.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 30. Часть программного кода, отвечающая за функцию

**Функция linal:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* k - переменная, принимающая значения в функции.
* y – координаты графика функции вида ax^2 + bx + c по оси ординат
* b - переменная, принимающая значения в функции.
* l – переменная, принимающая значения в функции.
* color - переменная, принимающая значения в функции.

**Функция quadratic:**

**Входные данные:**

* Отсутствуют.

**Выходные данные:**

* Отсутствуют.

**Переменные, затрагиваемые в ходе работы:**

* y – координаты графика функции вида ax^2 + bx + c по оси ординат
* ak - переменная, принимающая значения в функции.
* bk - переменная, принимающая значения в функции.
* ck – переменная, принимающая значения в функции.
* color - переменная, принимающая значения в функции.

## 5.3 Функции вывода информации

Метод вывода информации

Что делает: осуществляет вывод необходимой информации

Вывод информации осуществляется с помощью функции graphs\_approxim.

В данном методе осуществляется непосредственно вызов функций с помощью метода print ().

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 31. Часть кода, отвечающая за вывод данных

# Тестирование

Проведём тестирование нашей программы и сравним полученные показатели, чтобы сделать вывод о предпочтительном варианте использования нашей программы или Excel под условия заказчика.

## 6.1. Тестирование Датасета №1:

### 6.1.1. Метод Python:

Импортируем датасет в Python и получаем результат.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 32. импорт

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 33. аппроксимация 1

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 34. аппроксимация 2

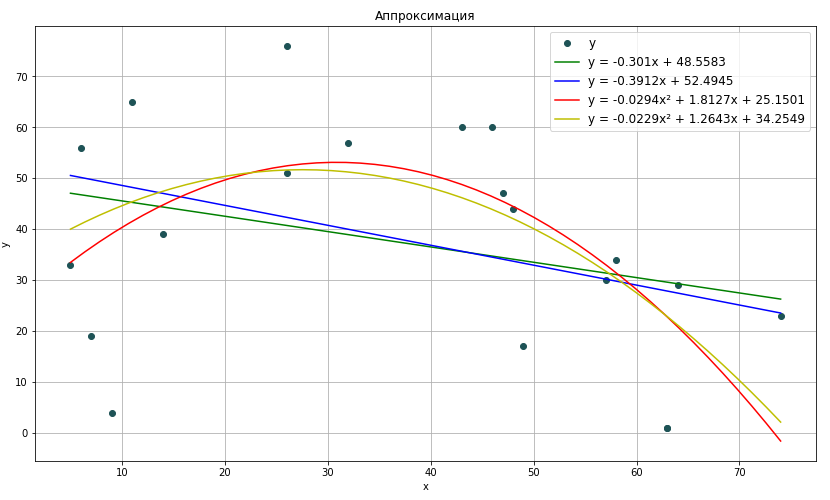


Рисунок 35.Аппроксимация

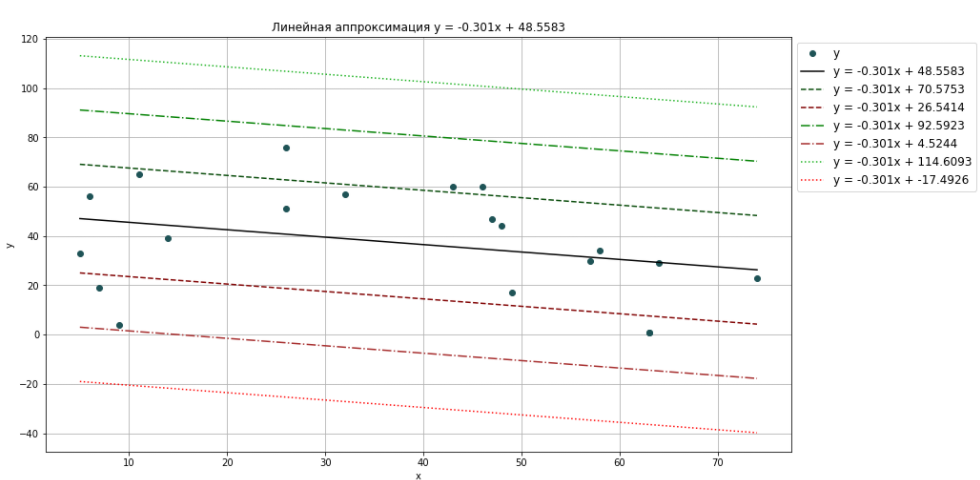


Рисунок 36. линейная аппроксимация 1



Рисунок 37. линейная аппроксимация 2

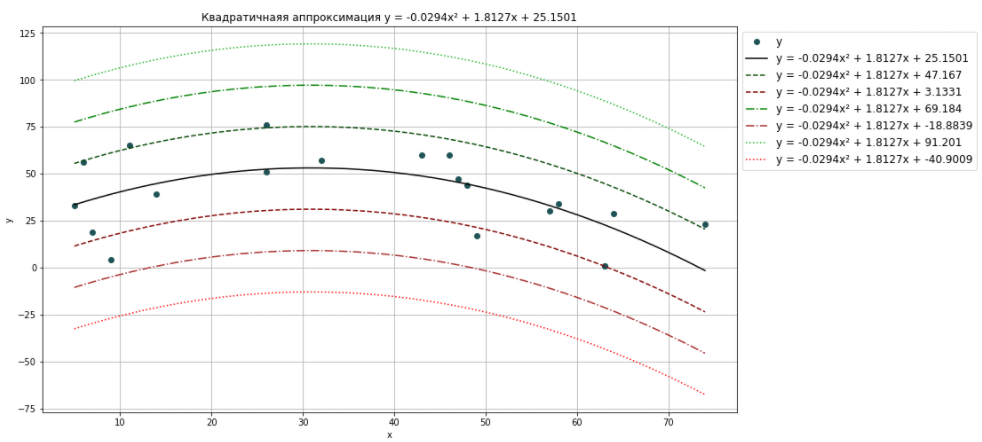


Рисунок 38. квадратная аппроксимация 1

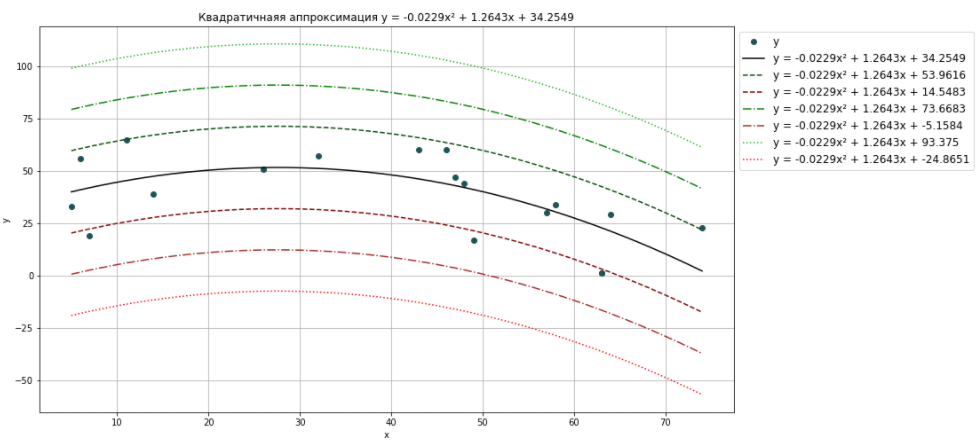


Рисунок 39. квадратная аппроксимация 2

### 6.1.2. Метод Excel:

Вводим данные датасета в поля X и Y и получаем результат.

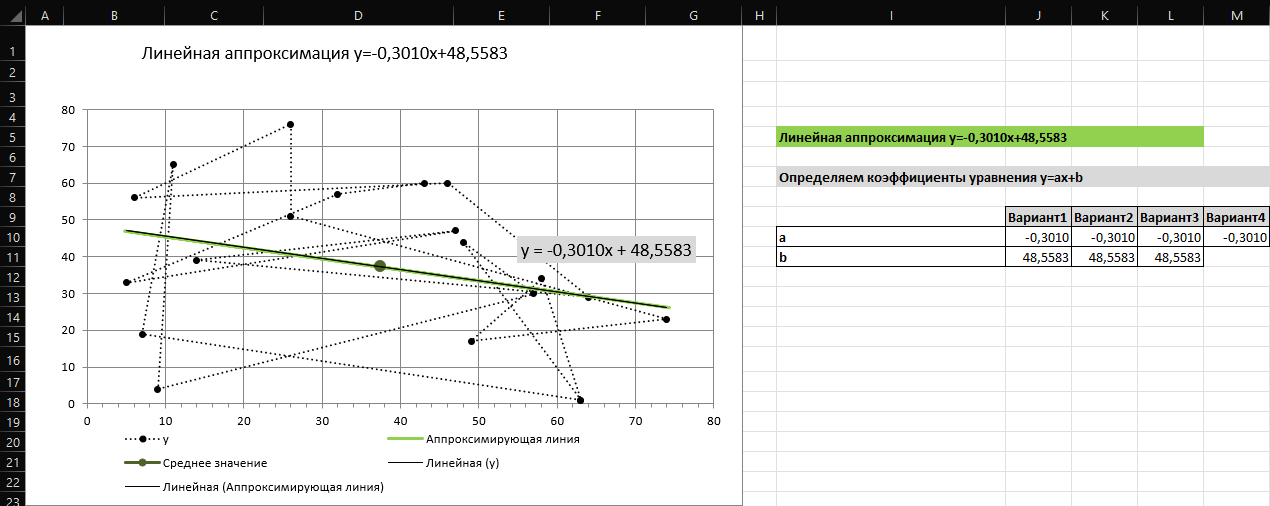


Рисунок 40. линейная аппроксимация

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 41. аппроксимирующая линия, дисперсия и отклонения

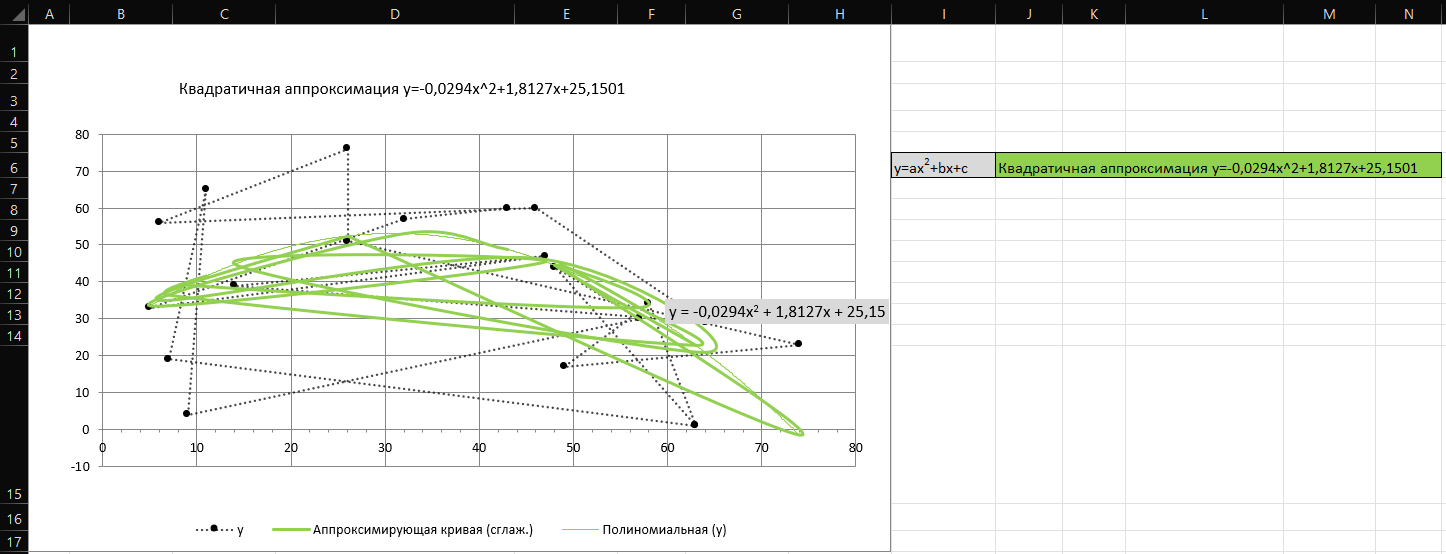


Рисунок 42. квадратичная аппроксимация

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 43. матрица

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 44, аппроксимирующая кривая, отклонения и дисперсия

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 45. решение с помощью линейной функции

## 6.2. Тестирование Датасета №2:

### 6.2.1. Метод Python:

Импортируем датасет в Python и получаем результат.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 46. импорт

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 47. аппроксимация 1

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 48. аппроксимация 2

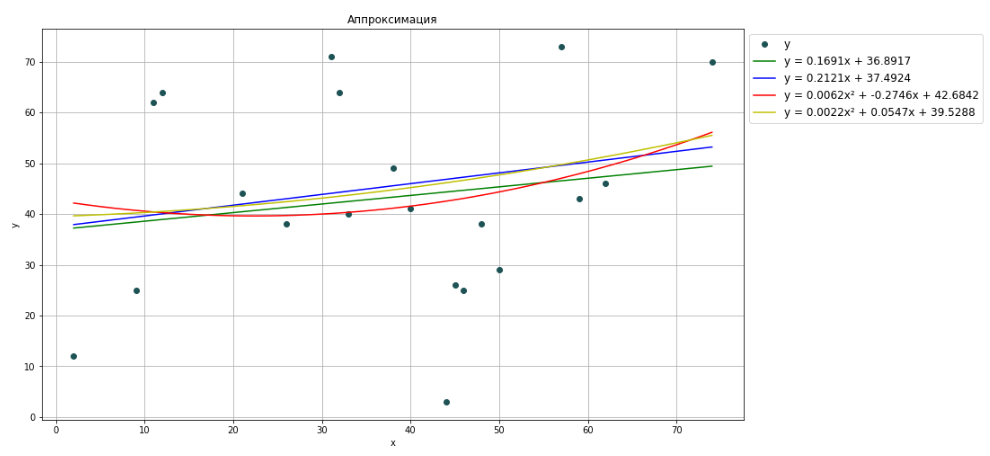


Рисунок 49. аппроксимация

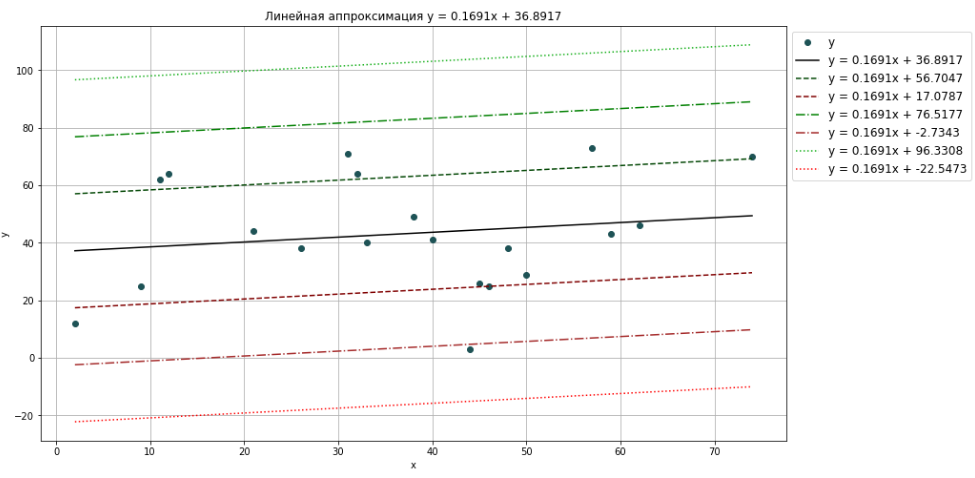


Рисунок 50. линейная аппроксимация 1

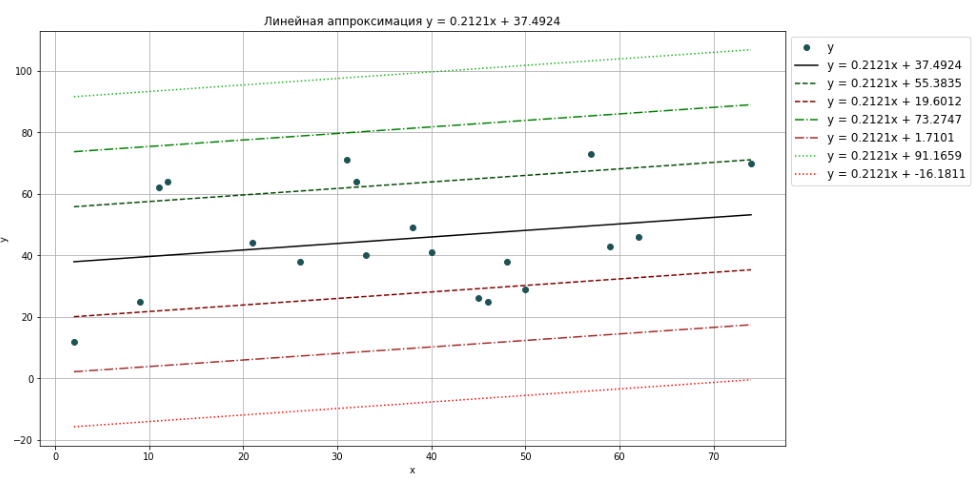


Рисунок 51. линейная аппроксимация 2

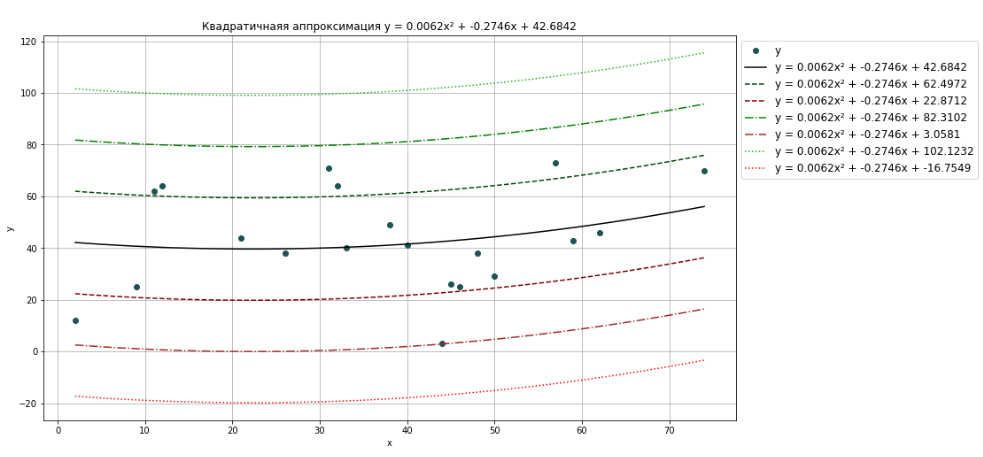


Рисунок 52. квадратичная аппроксимация 1

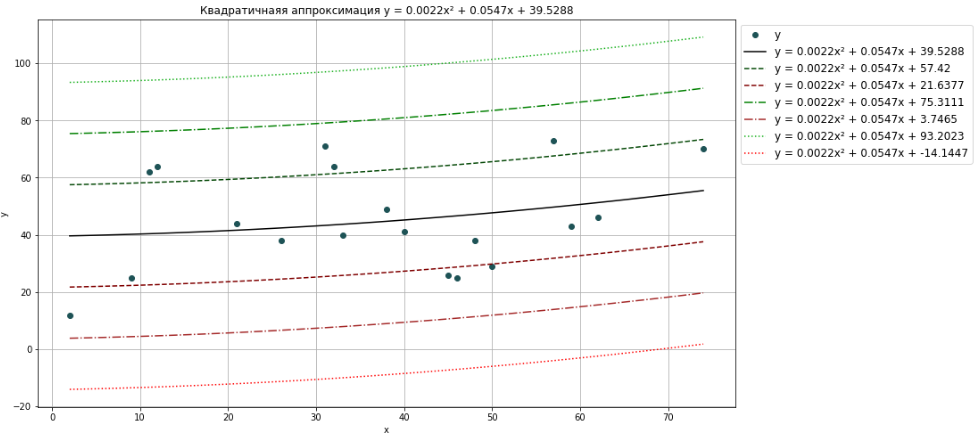


Рисунок 53. квадратичная аппроксимация 2

### 6.2.2. Метод Excel:

Вводим данные датасета в поля X и Y и получаем результат.

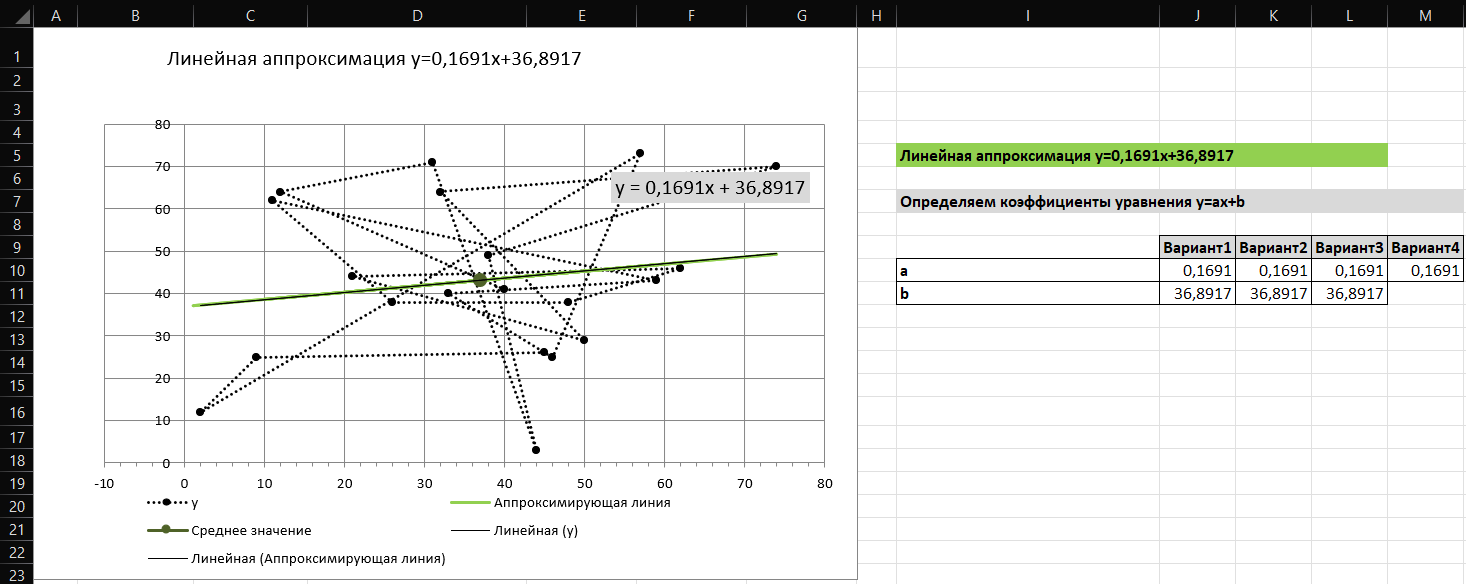


Рисунок 54. линейная аппроксимация

Изображение выглядит как текст, стол

Автоматически созданное описание

Рисунок 55. аппроксимирующая кривая, отклонения и дисперсия

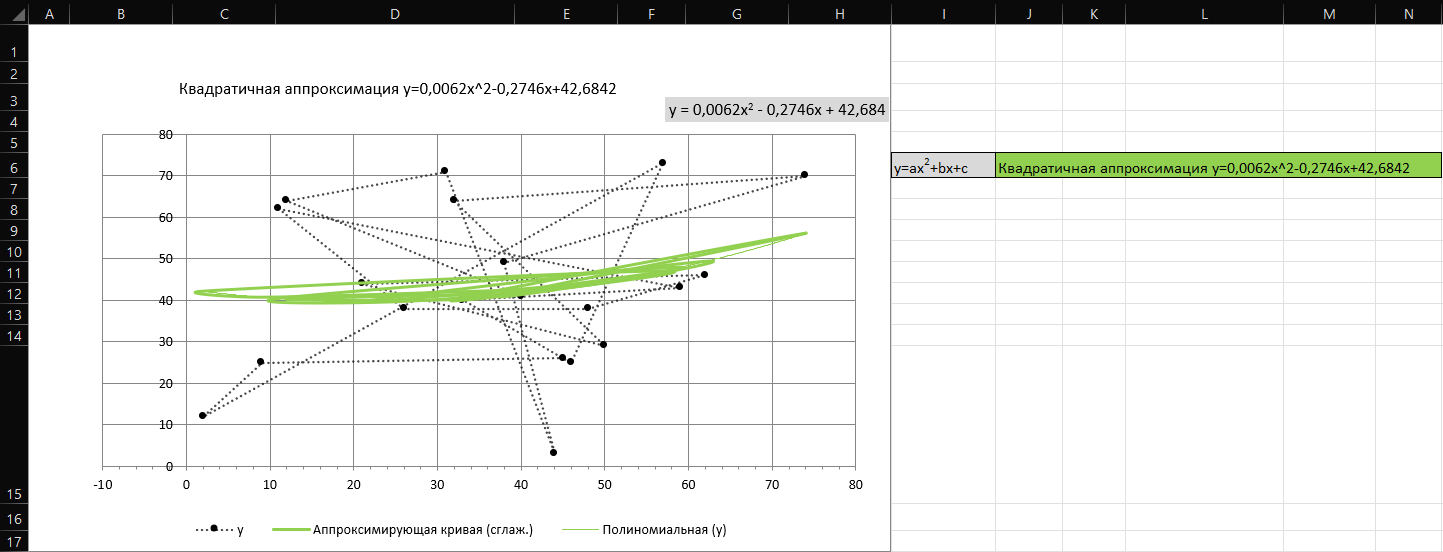


Рисунок 56. квадратичная аппроксимация

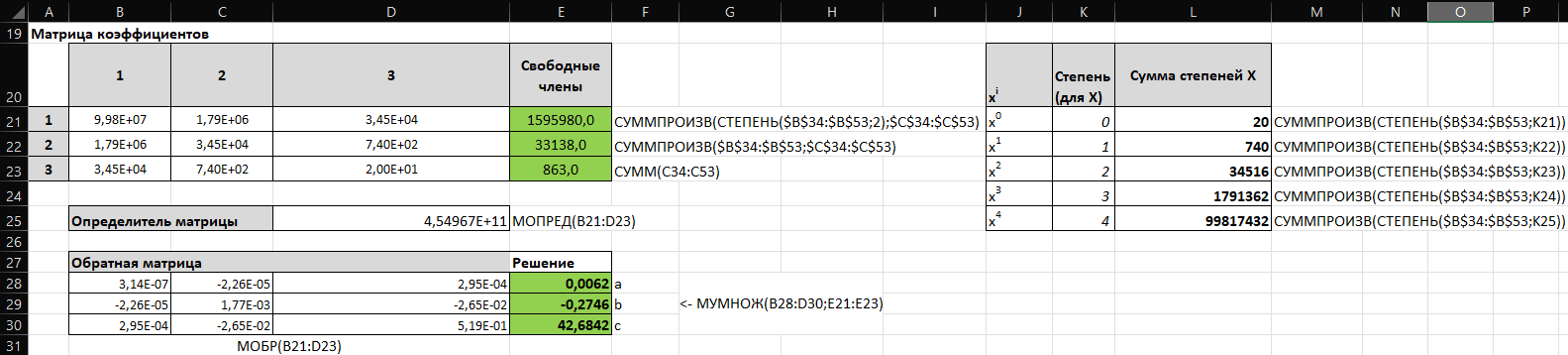


Рисунок 57. матрица

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 58. аппроксимирующая кривая, отклонения и дисперсия

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 59. решение с помощью функции ЛИНЕЙН

## 6.3. Тестирование Датасета №3:

### 6.3.1. Метод Python:

Импортируем датасет в Python и получаем результат.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 60. импорт

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 61. аппроксимация 1

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 62. аппроксимация 2

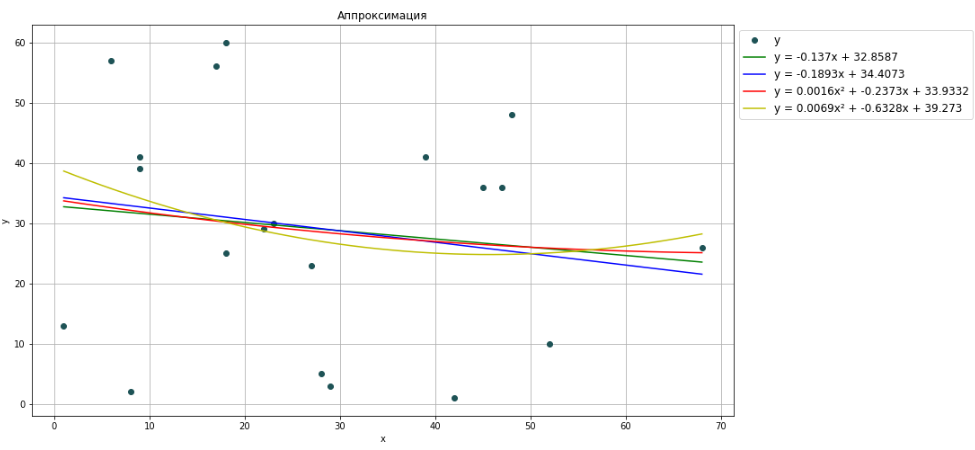


Рисунок 63. аппроксимация

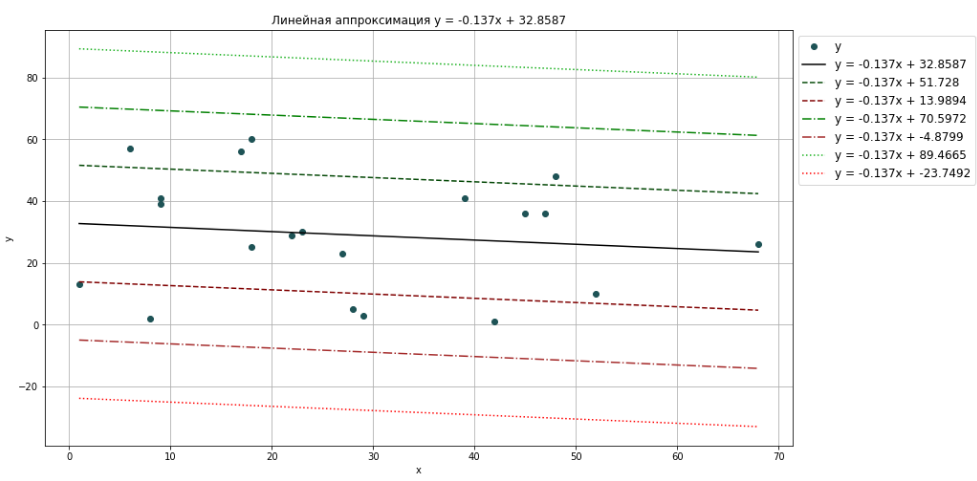


Рисунок 64. линейная аппроксимация 1

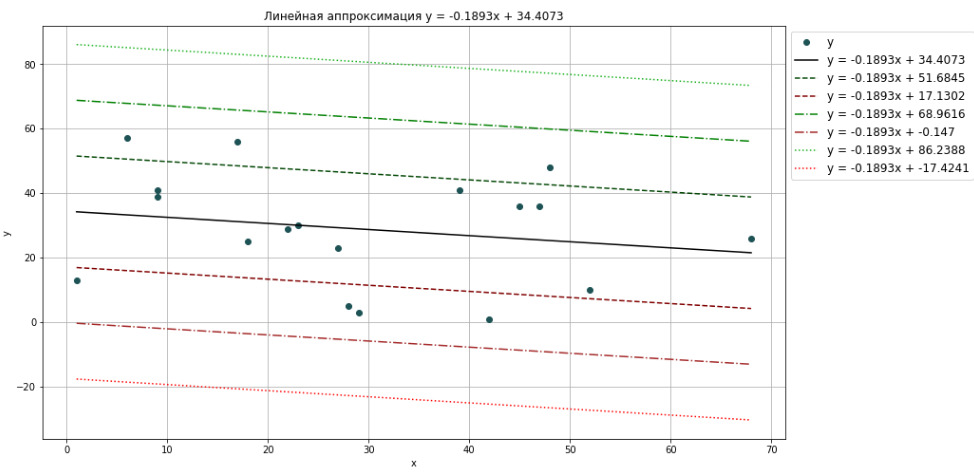


Рисунок 65. линейная аппроксимация 2

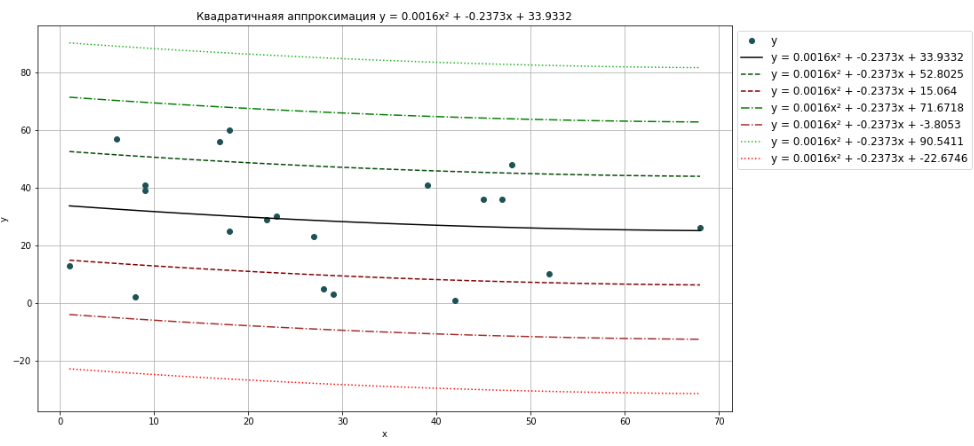


Рисунок 66. квадратичная аппроксимация 1

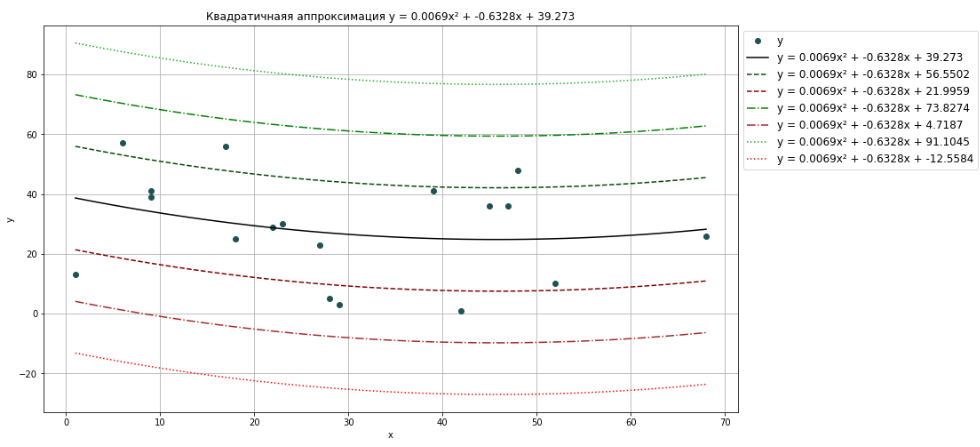


Рисунок 67. квадратичная аппроксимация 2

### 6.3.2. Метод Excel:

Вводим данные датасета в поля X и Y и получаем результат.

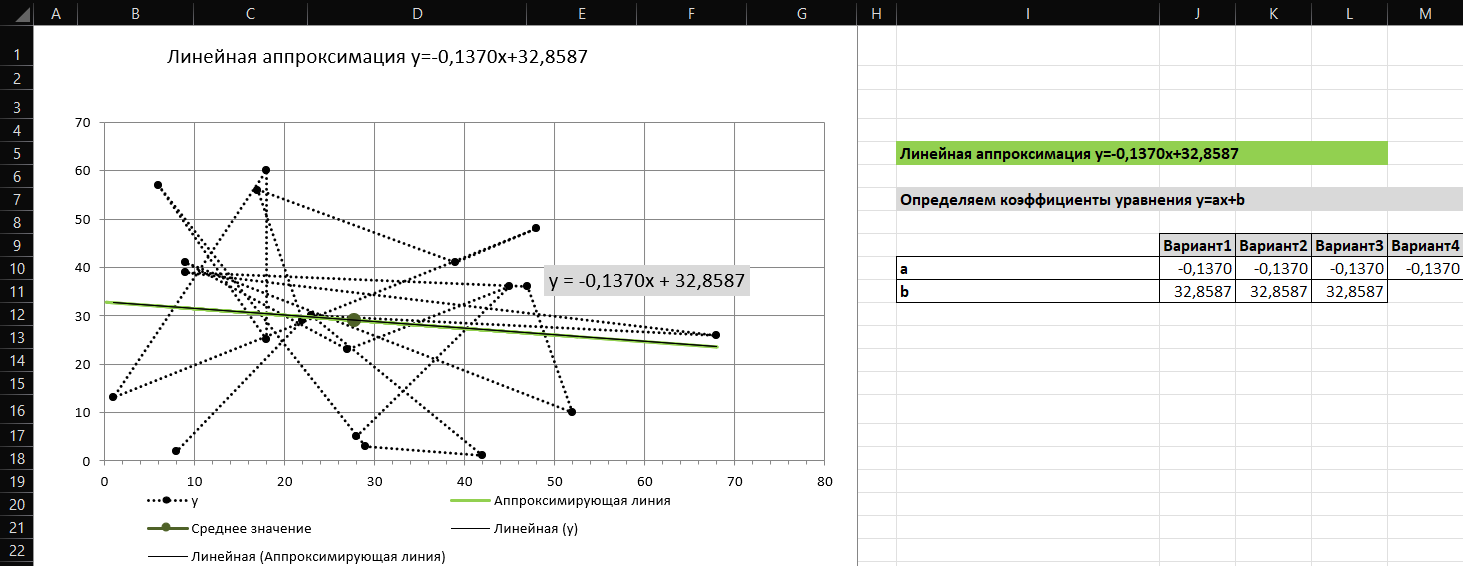


Рисунок 68. линейная аппроксимация

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 69. аппроксимирующая линия, отклонения и дисперсия

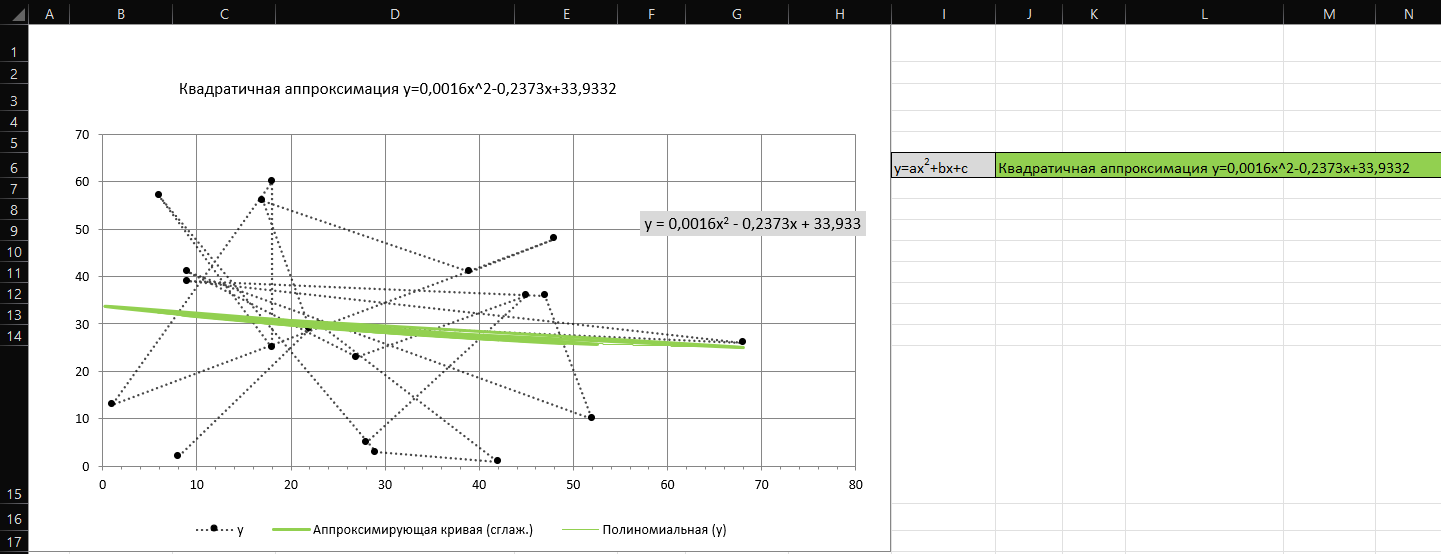


Рисунок 70. квадратичная аппроксимация

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 71. матрица

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 72. аппроксимирующая прямая, отклонения и дисперсия

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 73. Решение с помощью ЛИНЕЙН

## 6.4. Тестирование Датасета №4:

### 6.4.1. Метод Python:

Импортируем датасет в Python и получаем результат.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 74. импорт

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 75. аппроксимация 1

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 76. аппроксимация 2

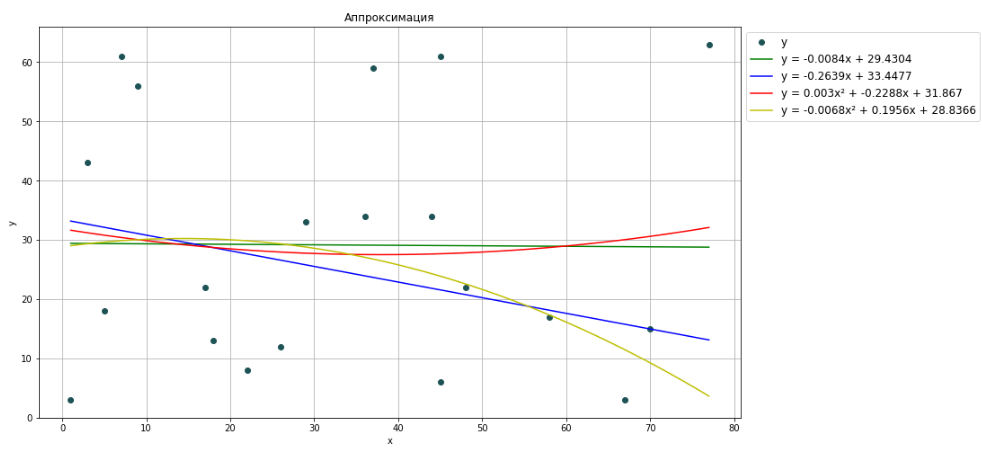


Рисунок 77. аппроксимация

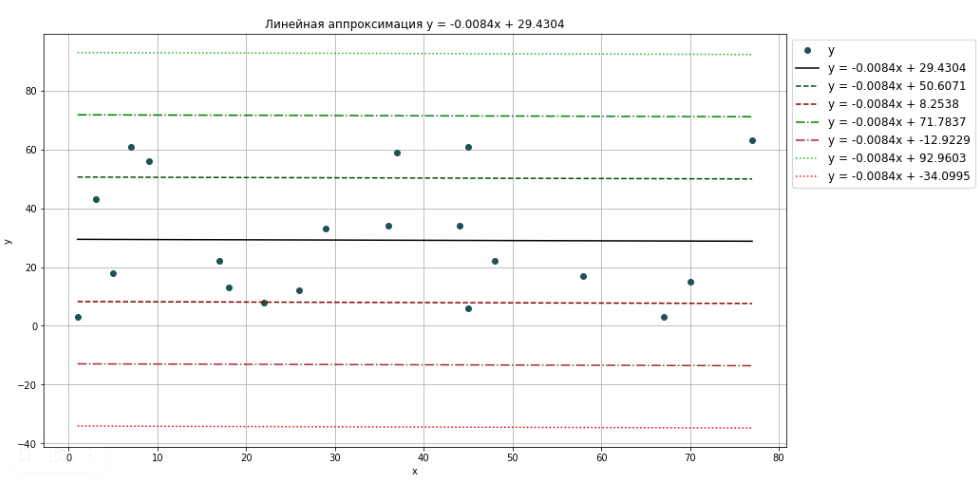
****

Рисунок 78. линейная аппроксимация 1

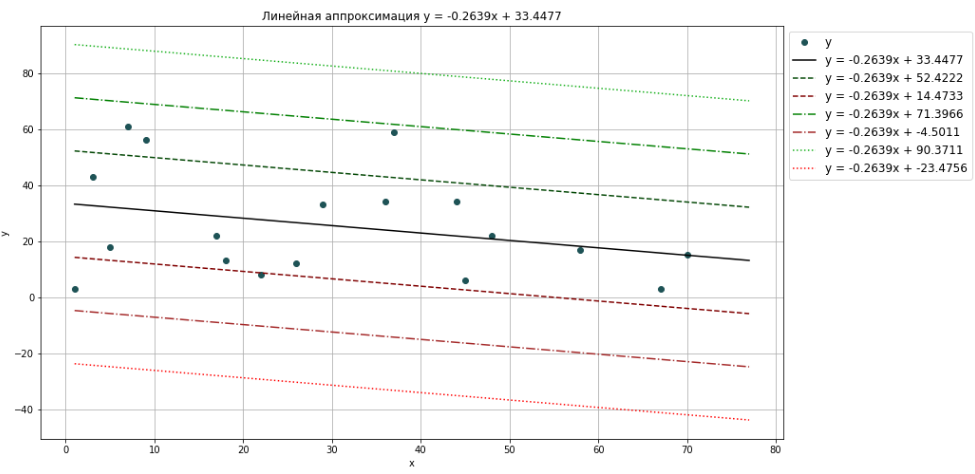


Рисунок 79. линейная аппроксимация 2

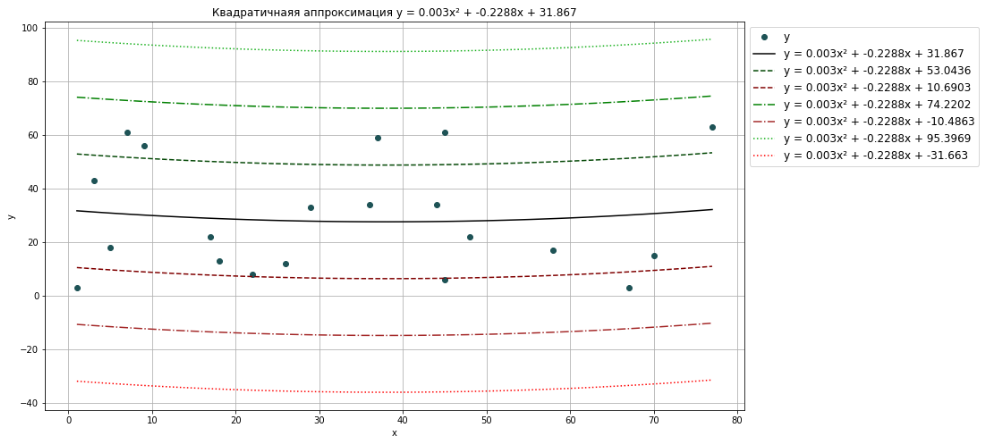


Рисунок 80. квадратичная аппроксимация 1

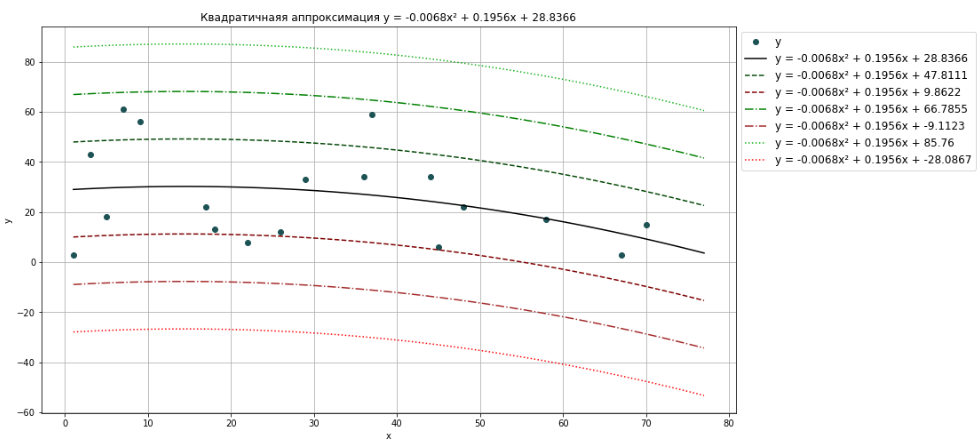


Рисунок 81. квадратичная аппроксимация 2

### 6.4.2. Метод Excel:

Вводим данные датасета в поля X и Y и получаем результат.

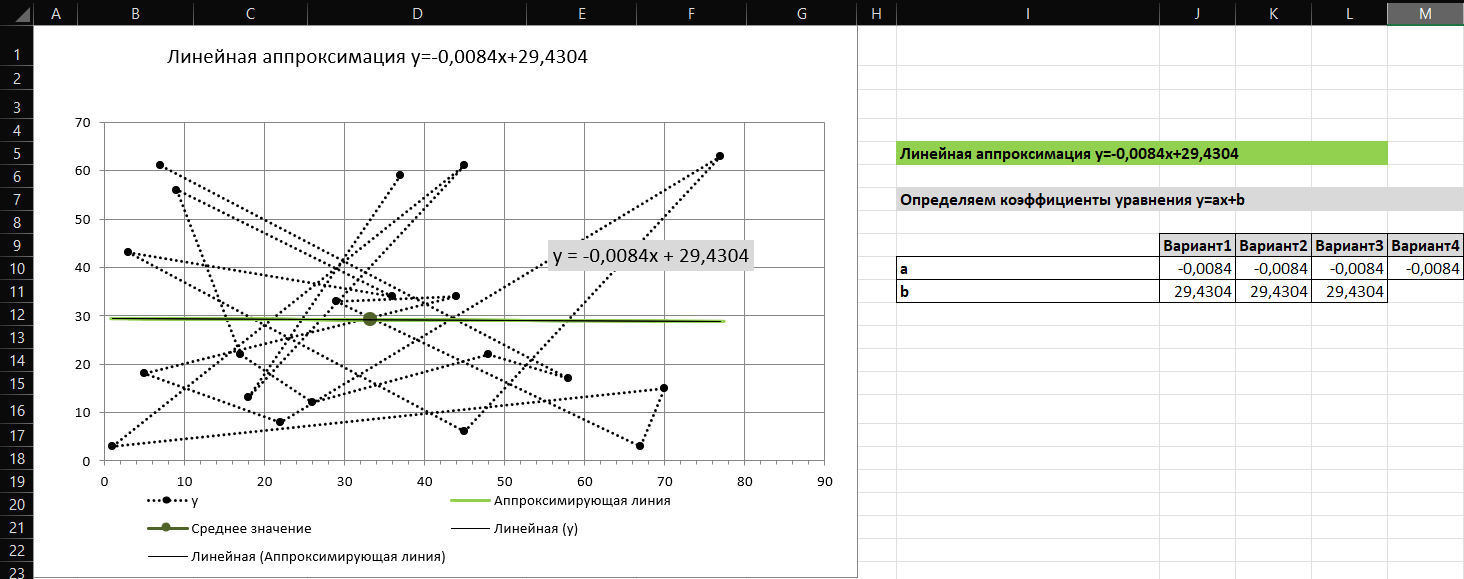


Рисунок 82. линейная аппроксимация

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 83. аппроксимирующая линия, отклонения и дисперсия

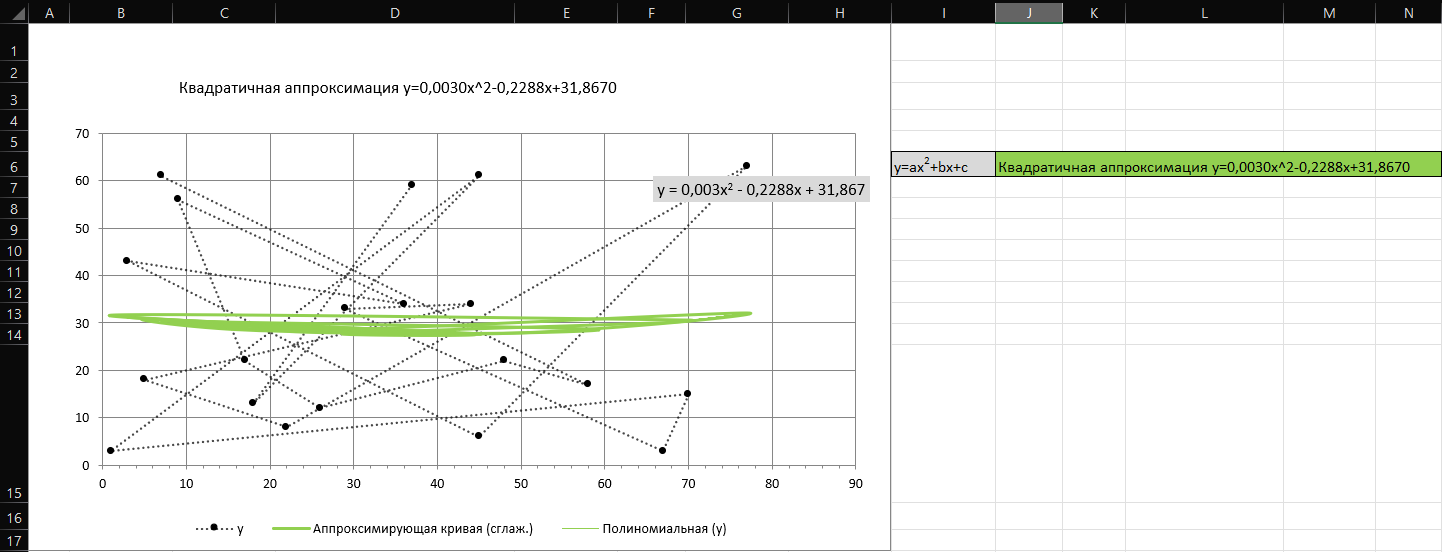


Рисунок 84. квадратичная аппроксимация

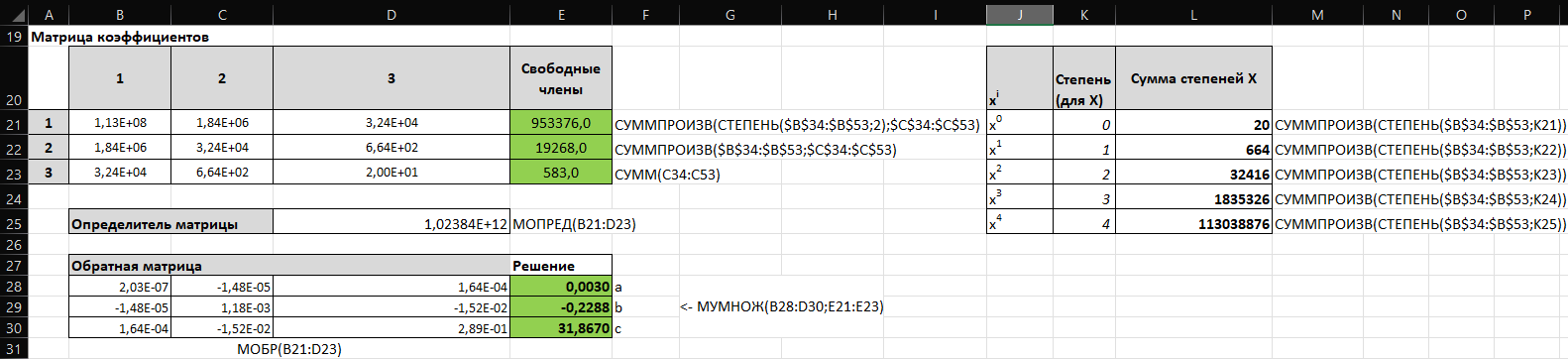


Рисунок 85. матрицы

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 86. аппроксимирующая кривая, отклонения и дисперсия



Рисунок 87. решение с помощью функции линейн

## 6.5. Тестирование Датасета №5:

### 6.5.1. Метод Python:

Импортируем датасет в Python и получаем результат.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 88. импорт

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 89. аппроксимация 1

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 90. аппроксимация 2

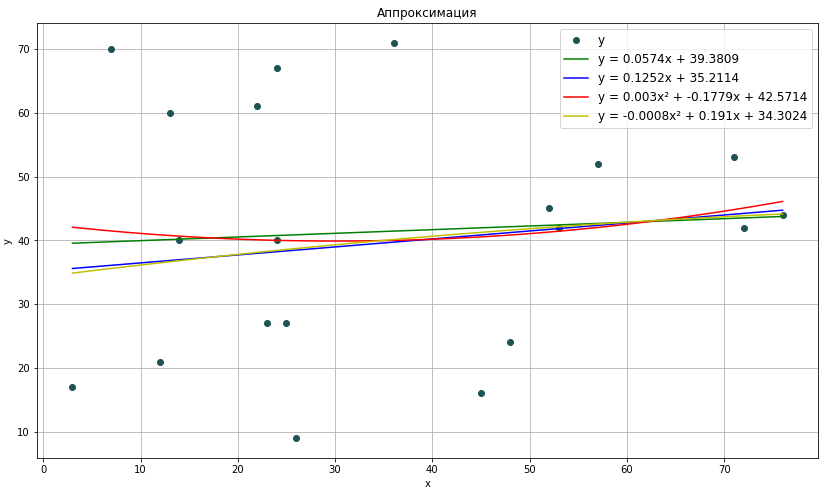


Рисунок 91. аппроксимация

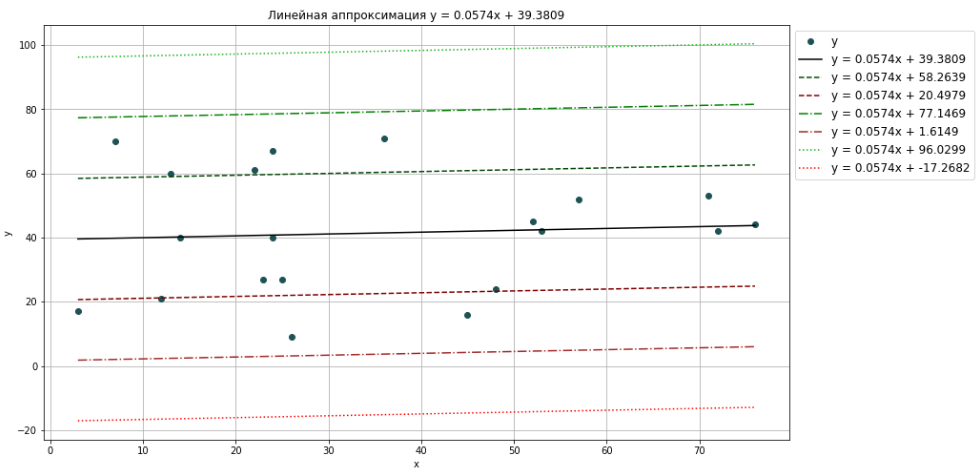


Рисунок 92. линейная аппроксимация 1

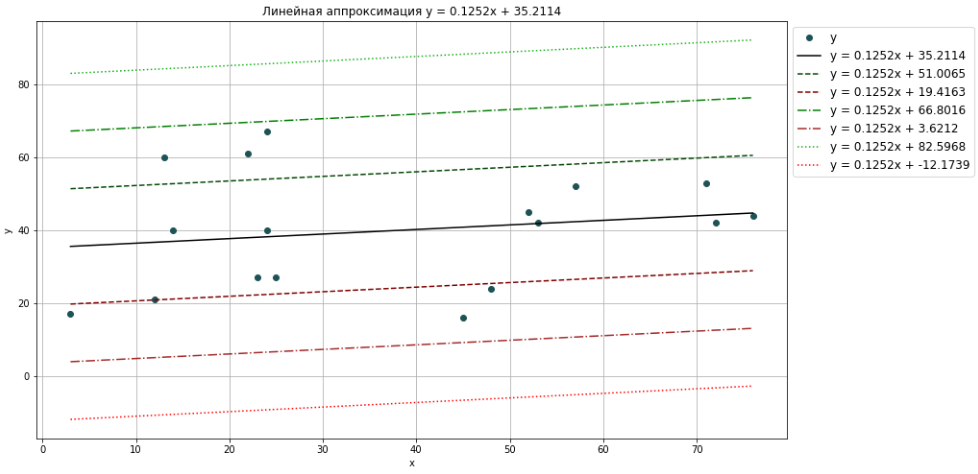


Рисунок 93. линейная аппроксимация 2

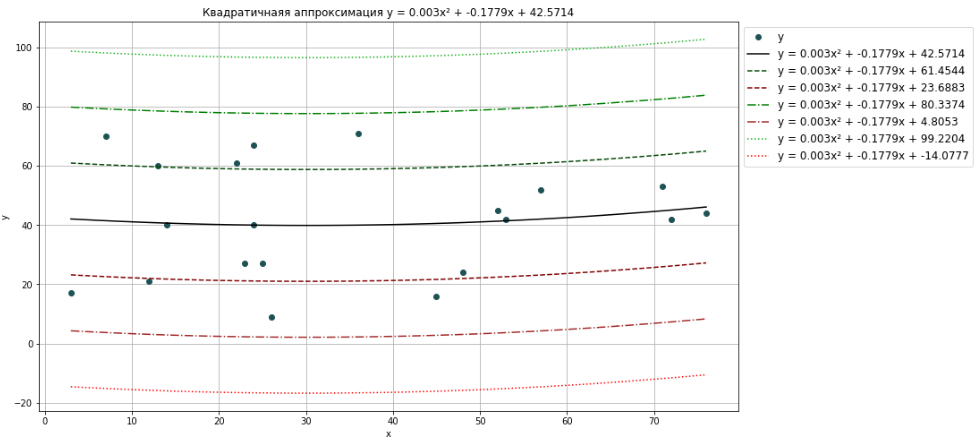


Рисунок 94. квадратичная аппроксимация 1

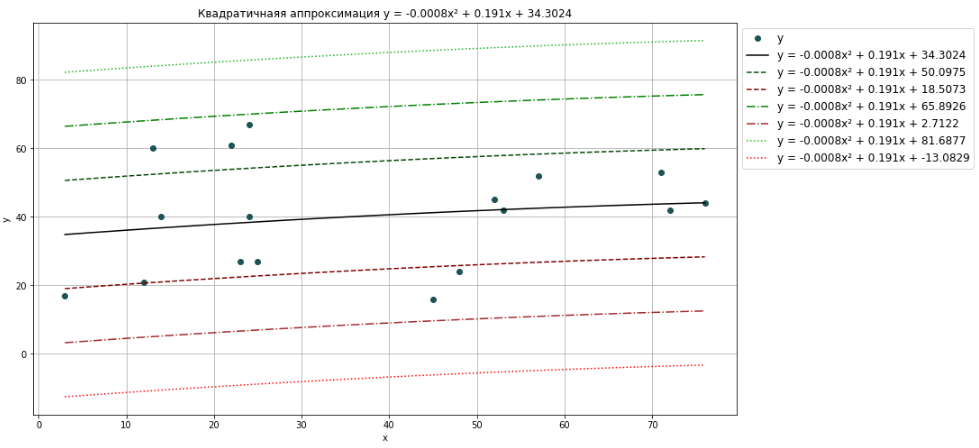


Рисунок 95. квадратичная аппроксимация 2

### 6.5.2. Метод Excel:

Вводим данные датасета в поля X и Y и получаем результат.

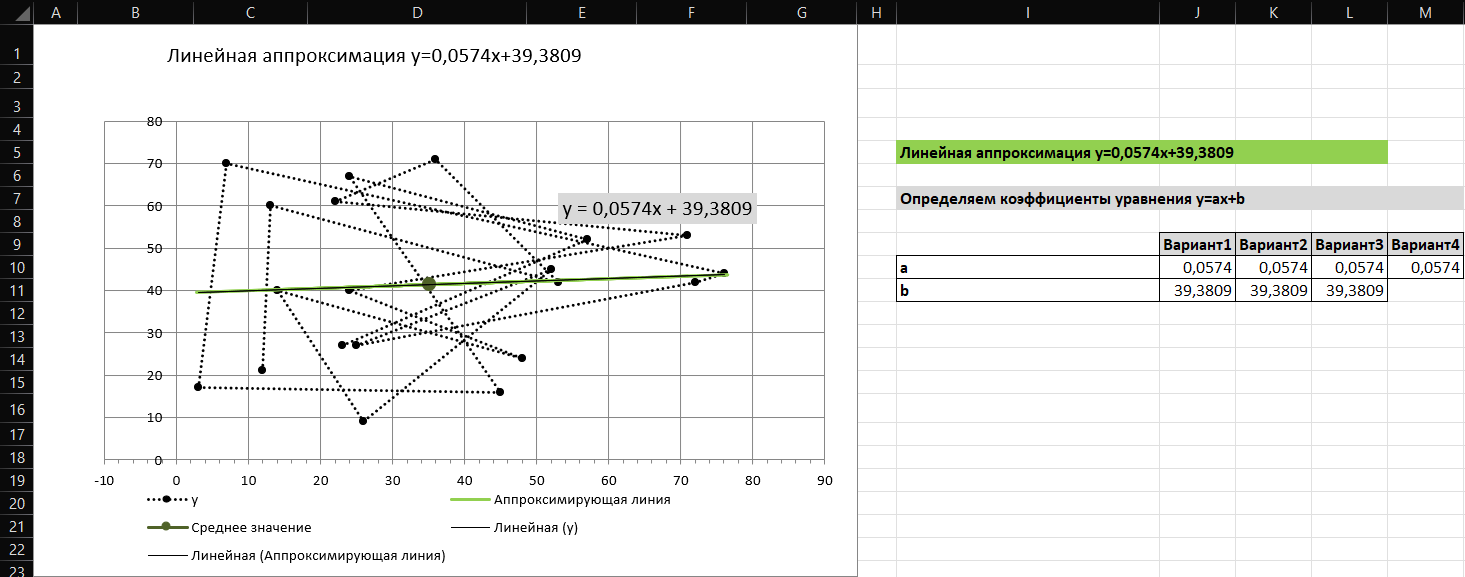


Рисунок 96. линейная аппроксимация

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 97. аппроксимирующая линия, отклонения и дисперсия

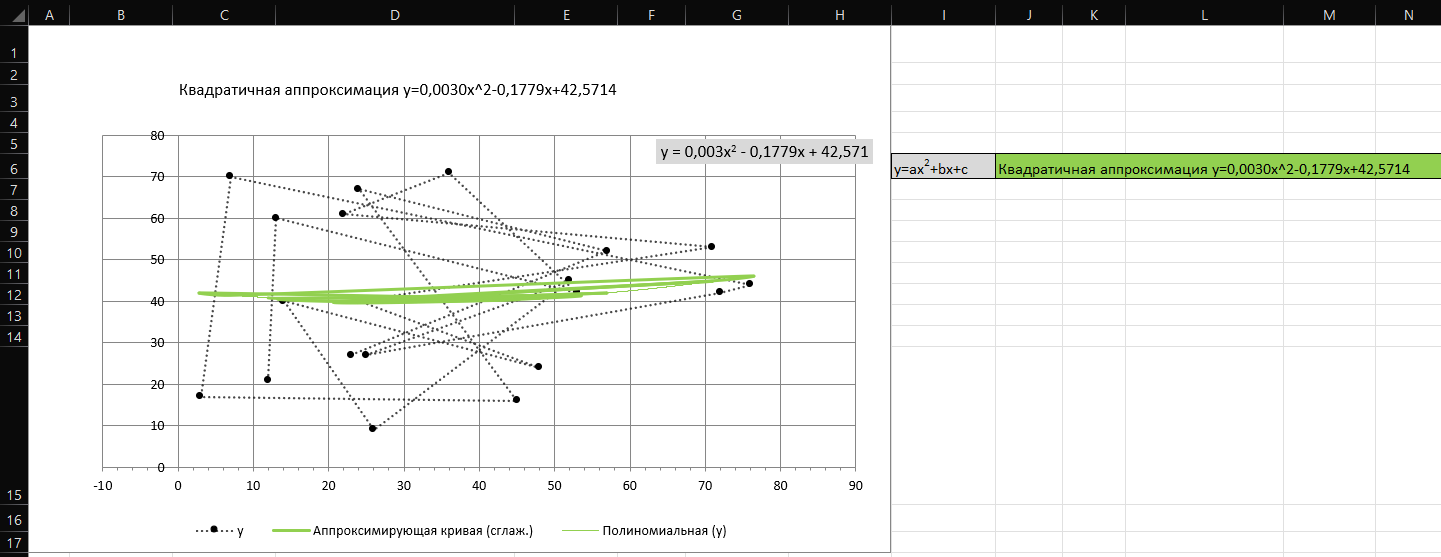


Рисунок 98. квадратичная аппроксимация

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 99. матрица

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 100. аппроксимирующая кривая, отклонения и дисперсия

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 101. решение с помощью функции линейн

# 7. Заключение

Наш представленный код решает поставленную задачу. На основании тестирования данного алгоритма можно сделать вывод о том, что Python выводит самое оптимальное решение достаточно быстро и наглядно.

Теперь сравним два алгоритма по критериям: эффективности, скорости использования алгоритма, простоты использования, надёжности в разрезе человеческого фактора и точности предоставляемого решения.

Таблица 2. Сравнение двух методов

|  |  |  |
| --- | --- | --- |
| **Критерий** | **Python** | **Excel** |
| Эффективность | Высокая | Высокая |
| Скорость использования алгоритма | Высокая | Высокая |
| Простота использования | Высокая | Средняя |
| Надёжность (человеческий фактор) | Высокая | Средняя |
| Точность | Высокая | Высокая |

Мы считаем, что представленный рукописный код на языке Python лучше, потому что удобнее, быстрее и проще, имеет функцию импорта исходных данных, а также более наглядную визуализацию относительно Excel. Улучшением кода может послужить добавление времени выполнения запроса, более детальной выводимой информации, ручного ввода данных, в том числе генерация рандомных данных.