**ПЕРВОЕ ВЫСШЕЕ ТЕХНИЧЕСКОЕ УЧЕБНОЕ ЗАВЕДЕНИЕ РОССИИ**



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОРНЫЙ УНИВЕРСИТЕТ**

Кафедра системного анализа и управления

**КУРСОВАЯ РАБОТА**

По дисциплине : Программирование и основы алгоритмизации систем управления

(наименование учебной дисциплины согласно учебному плану)

Тема работы: Разработка приложения для расчета рангового коэффициента корреляции Спирмена

Выполнил: студент гр. ИТУ-21 Шулепов К.В.

(шифр группы) (подпись) (Ф.И.О.)

Оценка:

Дата:

Проверил

руководитель работы: доцент Кухарова Т.В.

(должность) (подпись) (Ф.И.О.)

Санкт-Петербург

2023

СОДЕРЖАНИЕ

[Введение 4](#_Toc135040372)

[Описание предметной области 5](#_Toc135040373)

[Описание языка программирования 7](#_Toc135040374)

[Разработка приложения 8](#_Toc135040375)

[Тестирование приложения 15](#_Toc135040376)

[Заключение 20](#_Toc135040377)

[Библиографический список 21](#_Toc135040378)

[Приложение 22](#_Toc135040379)

# Введение

Ранговый коэффициент корреляции Спирмена - это статистический показатель, который используется для измерения силы связи между двумя переменными, которые измерены в ранговой шкале.

Данный коэффициент все еще является актуальным в научных исследованиях, особенно в области социальных наук, психологии и медицины. Он может быть использован для анализа различных взаимосвязей.

Ранговый коэффициент корреляции Спирмена также полезен в тех случаях, когда данные не соответствуют нормальному распределению или содержат выбросы. Он также может быть использован для анализа связи между переменными, измеренными в номинальной шкале.

Таким образом, ранговый коэффициент корреляции Спирмена остается актуальным и полезным показателем для анализа взаимосвязей между переменными в различных областях науки и приложений.

Цель работы – изучить алгоритм, способы расчёта рангового коэффициента корреляции Спирмена и разработать приложение для его расчёта.

Объект исследования – ранговый коэффициент корреляции Спирмена.

# Описание предметной области

Ранговый коэффициент корреляции Спирмена - это статистическая мера, используемая для измерения силы связи между двумя ранговыми переменными. Он может принимать значения от -1 до 1, где -1 означает полную обратную связь, 0 - отсутствие связи и 1 - полную прямую связь.

Коэффициент корреляции Спирмена был разработан британским статистиком Чарльзом Спирменом в начале 20-го века. Спирмен родился в 1863 году в Англии и учился в Оксфордском университете. Он сделал значительный вклад в различные области статистики, включая теорию рангов, корреляцию и факторный анализ.

В 1904 году Спирмен опубликовал статью, в которой он предложил новый метод оценки корреляции, который основывался на рангах данных, а не на их фактических значениях. Он заметил, что ранговая корреляция может быть более эффективной, чем обычная линейная корреляция, когда данные имеют асимметричное распределение или наличие выбросов.

Спирмен также предложил формулу для вычисления коэффициента корреляции Спирмена, которая в основном используется и по сей день. Спирмен получил много почетных наград за свою работу, в том числе, избрание его президентом Королевского статистического общества в 1933 году.

Ранговый коэффициент корреляции Спирмена основан на присвоении ранговым переменным ранговых значений, то есть каждому значению переменной присваивается ранг, отражающий его положение в отсортированном списке. Если два объекта имеют одинаковые значения, то им присваивается средний ранг.

Если значение коэффициента равно 0, то можно говорить об отсутствии связи между переменными. Значение 1 означает, что все значения одной переменной связаны с возрастающими значениями другой переменной, а значение -1 - обратная связь.

Ранговый коэффициент корреляции Спирмена часто используется в задачах, связанных с оценкой связи между ранговыми переменными, такими как оценка связи между ранговыми показателями качества продукта и ранговыми оценками клиентов. Он также используется для проверки гипотезы о том, что две ранговые переменные имеют одинаковое распределение.

Ранговый коэффициент корреляции Спирмена рассчитывается по следующему алгоритму:

Пусть имеется две выборки значений X и Y размера n.

Для каждой выборки вычисляем ранги элементов. Для этого:

1. Сортируем значения выборки по возрастанию.
2. Присваиваем первому значению ранг 1, второму - 2 и т.д.
3. Если в выборке есть повторяющиеся значения, то присваиваем им среднее значение рангов.
4. Вычисляем разности рангов для каждой пары соответствующих элементов X и Y.
5. Вычисляем сумму квадратов разностей рангов:
6. Суммируем квадраты разностей рангов для каждой пары соответствующих элементов X и Y.
7. Обозначим эту сумму .
8. Вычисляем ранговый коэффициент корреляции Спирмена по формуле:

Где:

r - ранговый коэффициент корреляции Спирмена

- сумма квадратов разностей рангов

n - размер выборок

Таким образом, ранговый коэффициент корреляции Спирмена измеряет степень линейной связи между двумя выборками, учитывая только порядок значений. Он часто используется в анализе данных для определения связи между двумя ранговыми переменными.

Он используется в различных областях, включая науку о данных, социологию, психологию, экономику и другие. Например, в науке о данных ранговый коэффициент корреляции Спирмена может использоваться для измерения корреляции между рейтингами двух продуктов, чтобы понять, насколько схожи предпочтения потребителей. В экономике он может использоваться для изучения связи между доходом и расходами.

Также он может использоваться в медицине для исследования связи между двумя медицинскими показателями, такими как вес и рост, для определения, существует ли между ними зависимость.

Однако, следует отметить, что если связь между переменными не является монотонной, то коэффициент корреляции Спирмена может быть низким, даже если между переменными существует связь. В таких случаях может быть полезно использовать другие методы, например, коэффициент корреляции Кендалла.

Также следует учитывать, что коэффициент корреляции Спирмена не измеряет причинно-следственную связь между переменными, а только степень их связи. Это означает, что даже если коэффициент корреляции высокий, это не означает, что одна переменная вызывает изменения в другой переменной.

# Описание языка программирования

Приложение в данной работе будет написано на языке программирования Python. Python - это интерпретируемый, объектно-ориентированный язык программирования, который был разработан в конце 1980-х годов Гвидо ван Россумом. Python имеет простой и читаемый синтаксис, который делает его легким в изучении и использовании для начинающих программистов. Он также имеет широкие возможности, которые делают его полезным во многих областях, таких как наука о данных, искусственный интеллект, веб-разработка, научные и инженерные вычисления, разработка игр и многие другие. Python является кроссплатформенным языком, который может быть использован на различных операционных системах, таких как Windows, macOS и Linux. Python также имеет обширную стандартную библиотеку, которая содержит множество полезных функций и инструментов для разработки приложений. Кроме того, Python имеет одно из самых больших сообществ разработчиков в мире, что делает его идеальным языком для разработки проектов в команде или для получения помощи и поддержки в различных вопросах.

Python имеет некоторые недостатки, которые могут привести к проблемам в некоторых ситуациях. Одним из главных недостатков Python является низкая производительность. Python, как интерпретируемый язык программирования, работает медленнее, чем компилируемые языки программирования, такие как C++ или Java. Это может привести к проблемам, особенно в приложениях, которые требуют обработки больших объемов данных или высокой производительности.

Еще одним недостатком Python является ограниченная поддержка многопоточности. Python поддерживает многопоточность, но из-за некоторых ограничений в реализации GIL (Global Interpreter Lock), который контролирует доступ к общей памяти, использование нескольких потоков в Python может быть менее эффективным, чем в некоторых других языках программирования.

Python также не является лучшим выбором для разработки мобильных приложений и игр, поскольку он может быть менее производительным, чем некоторые другие языки программирования, которые специально оптимизированы для этой цели.

Наконец, Python может иметь проблемы с управлением памятью, которые могут привести к утечкам памяти и другим проблемам, если код не написан правильно. Это может быть особенно проблематично для долгоживущих приложений, которые должны работать непрерывно.

Однако, несмотря на эти недостатки, Python остается одним из самых популярных и универсальных языков программирования, который может быть использован для множества задач, включая научные вычисления, разработку веб-приложений и создание искусственного интеллекта.

В данной работе приложение будет достаточно небольшим, поэтому Python со своей простотой кода будет идеальным для него языком программирования. Из-за обработки малого количества информации проблем со скоростью обработки данных быть не должно.

# Разработка приложения

Приложение для расчёта ранговой корреляции коэффициента Спирмена получится довольно компактным благодаря встроенной библиотеке Scipy, где есть необходимые функции для выполнения этой задачи. Код будет состоять из импорта библиотек, создания интерфейса и функции, выполняющей расчёт.

Разберем код по частям:

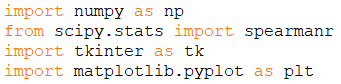


Рисунок 1 – Импорт библиотек

Код импортирует несколько модулей для использования в программе. numpy - библиотека для работы с массивами чисел и математическими операциями над ними.   
NumPy (Numeric Python) - это библиотека языка программирования Python для работы с многомерными массивами (в том числе матрицами) и высокоуровневыми математическими функциями, которые позволяют работать с данными быстро и удобно.

Массивы NumPy предоставляют эффективный способ хранения и манипулирования большими объемами данных. Они являются более быстрым и эффективным в использовании по сравнению с обычными списками Python. Они также предоставляют множество встроенных функций, которые упрощают выполнение математических операций и преобразований.

В NumPy существуют различные методы создания массивов, включая создание массивов из списка, массивов случайных чисел, массивов из файлов и многое другое. Также, NumPy предоставляет множество функций для работы с массивами, включая математические операции, операции сравнения, индексирование и срезы.

Библиотека NumPy часто используется в научных вычислениях, машинном обучении, обработке сигналов, обработке изображений, статистике и многих других областях.

scipy.stats - модуль для статистических функций, в данном случае используется функция spearmanr, которая позволяет рассчитать ранговый коэффициент корреляции Спирмена.

scipy.stats - это модуль библиотеки SciPy, который предоставляет функции для статистического анализа данных. Он включает множество распределений вероятности, статистических функций и тестов гипотез.

С помощью модуля scipy.stats можно рассчитывать различные статистические характеристики, такие как среднее значение, стандартное отклонение, дисперсию, корреляцию и многое другое. Кроме того, этот модуль также включает реализации многих распределений вероятности, таких как нормальное, экспоненциальное, бета-распределение и многие другие.

Модуль scipy.stats также включает функции для выполнения статистических тестов, таких как тест Стьюдента, анализ дисперсии (ANOVA), тест Колмогорова-Смирнова, тест Шапиро-Уилка и другие.

tkinter - библиотека для создания графического интерфейса пользователя.

Tkinter - это стандартная библиотека языка Python, которая предоставляет инструменты для создания графических интерфейсов пользователя (GUI). Tkinter основан на библиотеке Tcl/Tk, которая была написана на языке программирования Tcl и обеспечивает возможности создания интерфейсов для программ на разных языках программирования, включая Python.

С помощью Tkinter вы можете создавать окна, кнопки, текстовые поля, чекбоксы, радиокнопки, списки, меню и многое другое. Tkinter имеет простой синтаксис, позволяющий быстро создавать интерфейсы. Также она достаточно гибкая, чтобы удовлетворить потребности большинства пользователей.

Большинство методов и функций в Tkinter достаточно интуитивно понятны и могут быть использованы даже без глубокого знания программирования GUI. Кроме того, существует множество руководств и примеров, которые помогут вам начать работу с Tkinter.

matplotlib.pyplot - библиотека для визуализации данных, в данном случае используется для создания диаграммы рассеяния и регрессионной прямой.

matplotlib.pyplot - это модуль библиотеки Matplotlib, который предоставляет простой интерфейс для создания графиков в Python. Он обычно используется вместе с NumPy и другими библиотеками, которые предоставляют массивы данных.

matplotlib.pyplot позволяет создавать различные типы графиков, такие как линейные графики, гистограммы, круговые диаграммы, диаграммы рассеяния и многое другое. Он также обладает многими возможностями настройки, такими как изменение масштаба осей, изменение цветов и типов линий, добавление текста и меток и многое другое.

Основной элемент графика в matplotlib.pyplot - это фигура (Figure), на которой располагаются оси (Axes). На оси можно нарисовать различные элементы графика, такие как линии, точки, текст и метки. Кроме того, можно добавлять легенды, заголовки и метки осей для облегчения чтения графика.

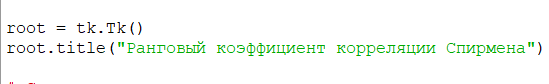


Рисунок 2 – Создание главного окна

Здесь создаётся главное окно приложения и установка заголовка окна. root = tk.Tk() - это создание главного окна приложения в библиотеке tkinter.

tk - это модуль библиотеки Tkinter (Tcl/Tk интерфейс для Python), который предоставляет интерфейс для создания графических пользовательских интерфейсов (GUI) в Python.

Tk() - это метод класса tkinter, который создает главное окно приложения. После этого, вы можете добавлять на это окно различные виджеты, такие как кнопки, текстовые поля, выпадающие списки и т.д. с помощью других методов класса.

В данном коде root - это объект окна, на котором мы можем размещать виджеты и задавать их свойства.

title - это метод класса tkinter, который устанавливает заголовок окна. В данном случае, заголовок устанавливается в "Ранговый коэффициент корреляции Спирмена". Это будет отображаться в заголовке окна приложения.

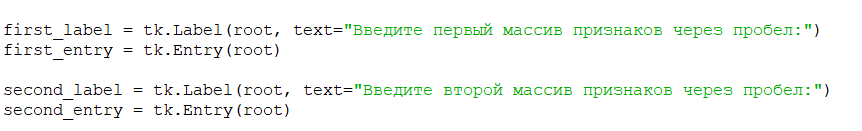


Рисунок 3 – Создание полей для ввода

Далее создаём два виджета Label и два виджета Entry. Label используется для отображения текста, а Entry используется для ввода данных пользователем.

first\_label = tk.Label(root, text="Введите первый массив признаков через пробел:") - это создание объекта Label (метки) на главном окне root.

tk.Label - это класс метки в библиотеке tkinter. Метка - это виджет, который может содержать текстовую информацию.

root - это родительский виджет, на котором будет размещен данный виджет. В данном случае, родительским виджетом является главное окно приложения.

text="Введите первый массив признаков через пробел:" - это текст, который будет отображаться на метке. В данном случае, на метке будет отображаться текст " Введите первый массив признаков через пробел:".

first\_entry = tk.Entry(root) - это создание объекта Entry (поля ввода) на главном окне root.

tk.Entry - это класс поля ввода в библиотеке tkinter. Поле ввода - это виджет, который позволяет пользователю вводить текстовую информацию.

Объект first\_entry - это переменная, которая будет использоваться для обращения к этому виджету и его свойствам в дальнейшем.

Две следующие строки кода – это выполнение аналогичных действий со вторым массивом признаков.

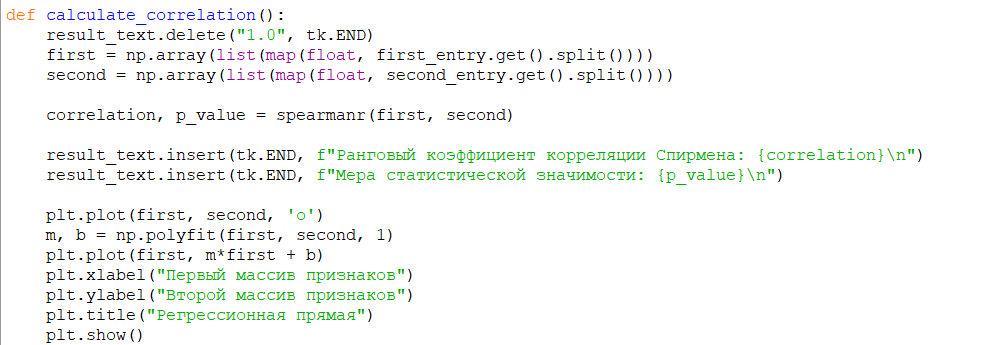


Рисунок 4 – Функция calculate\_correlation

Эта функция рассчитывает ранговый коэффициент корреляции Спирмена между двумя массивами признаков и строит регрессионную прямую для этих массивов.

result\_text.delete("1.0", tk.END) удаляет весь текст в поле для вывода результата, чтобы предыдущие расчёты не были рядом с новыми.

result\_text.delete("1.0", tk.END) - функция очищает виджет текста result\_text перед выводом нового результата.

first = np.array(list(map(float, first\_entry.get().split()))) - строка считывает содержимое поля ввода first\_entry и преобразует его в массив с помощью библиотеки NumPy. Она сначала вызывает метод get() для получения текстового содержимого поля ввода, затем метод split() для разбиения этого текста на список строк, используя пробелы в качестве разделителей. Наконец, map() вызывает функцию float() для преобразования каждой строки в числовое значение типа float. Результаты map() обрабатываются функцией list(), чтобы создать список из числовых значений, которые затем преобразуются в массив NumPy с помощью np.array().

second = np.array(list(map(float, second\_entry.get().split()))) - аналогичная строка для чтения и преобразования содержимого второго поля ввода second\_entry.

correlation, p\_value = spearmanr(first, second) - функция spearmanr() из модуля SciPy вычисляет ранговый коэффициент корреляции Спирмена и меру статистической значимости для двух массивов first и second. Результаты сохраняются в переменных correlation и p\_value.

Мера статистической значимости, также называемая p-значением, в коэффициенте корреляции Спирмена показывает вероятность того, что найденная корреляция является случайной, то есть что нет настоящей связи между двумя признаками. Обычно, если p-значение меньше 0,05, то находится статистически значимая корреляция, то есть связь между признаками не случайна. Если же p-значение больше 0,05, то корреляция считается незначительной. На экран мера статистической значимости выводится не будет, т.к. для каждой ситуации её требуемое значение может быть различным.

Функция spearmanr() в модуле scipy.stats вычисляет коэффициент корреляции Спирмена между двумя переменными. Она принимает два одномерных массива данных, и возвращает два значения: значение коэффициента корреляции и p-value.

Прежде чем вычислить коэффициент корреляции, spearmanr() сортирует каждый массив данных в порядке возрастания и присваивает каждому элементу его порядковый ранг. Затем вычисляется коэффициент корреляции между двумя массивами порядковых рангов, что позволяет избежать проблем, связанных с нарушением нормальности распределения или наличием выбросов.

P-value, возвращаемое spearmanr(), является вероятностью того, что наблюдаемая корреляция между двумя переменными может быть случайной. Если p-value меньше уровня значимости (обычно 0,05), то мы можем отклонить гипотезу о том, что корреляция случайна, и считать корреляцию статистически значимой. Если p-value больше уровня значимости, то мы не можем отклонить гипотезу о случайной корреляции.

Функция spearmanr() также имеет необязательные параметры, которые позволяют задать другие способы обработки по

ий или настройки для расчета p-value.

result\_text.insert(tk.END, f"Ранговый коэффициент корреляции Спирмена: {correlation}\n") - функция insert() добавляет результаты вычисления коэффициента корреляции в виджет текста result\_text.

result\_text.insert(tk.END, f"Мера статистической значимости: {p\_value}\n") - аналогичная строка для вывода меры статистической значимости.

Затем код строит график рассеяния линейной регрессии для двух массивов данных first и second.

plt.plot(first, second, 'o') строит график рассеяния, в котором каждой точке на графике соответствует пара значений из массивов first и second. Знак 'o' означает, что график будет точечный.

Функция plot в библиотеке matplotlib используется для создания линейных графиков. Она принимает два обязательных аргумента - x и y - массивы данных для координат оси X и Y соответственно.

Например, plt.plot(x, y) создаст линейный график, где значения x будут отображаться по оси X, а значения y - по оси Y.

Кроме того, функция plot также принимает необязательные аргументы, которые позволяют настроить внешний вид графика, такие как:

color - цвет линии графика

linestyle - стиль линии графика (например, пунктирный, сплошной, штрих-пунктирный и т.д.)

marker - тип маркера, который отображается на точках данных на графике (например, круг, треугольник, крест и т.д.)

label - метка, которая используется для обозначения линии графика в легенде

и многие другие.

Например, plt.plot(x, y, color='r', linestyle='--', marker='o', label='My Plot') создаст график с красной пунктирной линией, круглыми маркерами на точках данных и меткой "My Plot" в легенде.

После того, как график был нарисован, его можно настроить с помощью других функций matplotlib, таких как xlabel, ylabel, title, legend и т.д. Затем график можно отобразить с помощью функции show.

m, b = np.polyfit(first, second, 1) использует метод наименьших квадратов, чтобы найти коэффициенты угла наклона m и пересечения с осью y b линейной регрессии, которая наилучшим образом соответствует этим данным.

(x, y, deg) является функцией библиотеки NumPy, которая позволяет провести полиномиальную аппроксимацию (линейную или нелинейную) на основе заданных данных.

В данном случае x и y являются переменными, значения которых передаются в функцию. Аргумент deg указывает степень полинома, которым мы хотим аппроксимировать данные. В данном случае мы используем deg=1, чтобы получить линейную аппроксимацию.

Функция np.polyfit возвращает коэффициенты полинома, который соответствует линейной аппроксимации вида: y = mx + b, где m - наклон линии, а b - смещение (точка пересечения с осью y).

Таким образом, в строке m, b = np.polyfit(first, second, 1) мы передаем массивы first и second, которые содержат значения первого и второго признаков соответственно, и задаем deg=1 для получения линейной аппроксимации. Затем функция np.polyfit возвращает коэффициенты линейной аппроксимации, которые мы сохраняем в переменных m и b. После этого мы можем использовать эти коэффициенты, чтобы построить линию регрессии.

plt.plot(first, m\*first + b): Эта команда строит линейную регрессионную прямую, используя коэффициенты, найденные в предыдущем шаге.

plt.xlabel("Первый массив признаков") и plt.ylabel("Второй массив признаков"): Эти команды устанавливают подписи для осей x и y соответственно.

plt.title("Регрессионная прямая") устанавливает заголовок для графика.

plt.show() отображает график.

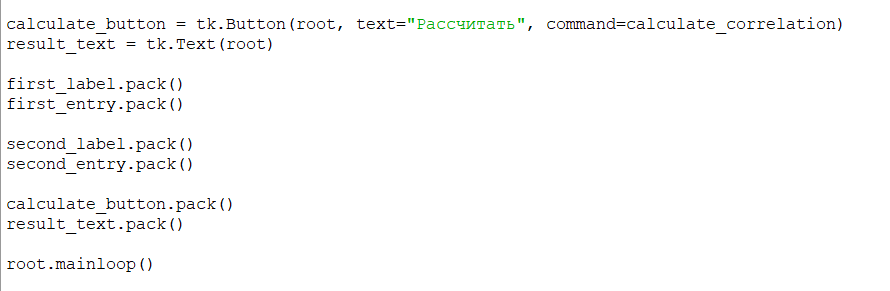


Рисунок 5 – Добавление элементов интерфейса

Этот код создает графический интерфейс (GUI) с помощью библиотеки Tkinter в Python с помощью метода pack().

Элементы интерфейса:

- first\_label - метка для первого ввода признаков;

- first\_entry - поле ввода для первого ввода признаков;

- second\_label - метка для второго ввода признаков;

- second\_entry - поле ввода для второго ввода признаков;

- calculate\_button - кнопка "Рассчитать";

- result\_text - текстовое поле для вывода результатов.

Виджеты добавляются на главное окно (root) с помощью метода .pack().

Когда пользователь нажимает на кнопку "Рассчитать", вызывается функция calculate\_correlation(), которая рассчитывает коэффициент корреляции и выводит результаты в текстовое поле result\_text. Также функция создает две диаграммы: диаграмму рассеяния и регрессионную прямую.

Наконец, окно отображается на экране с помощью метода .mainloop(). mainloop() - это метод tkinter, который запускает главный цикл обработки событий. Он обрабатывает все события, такие как нажатия кнопок, перемещения мыши и другие, и отвечает за отображение окна и его элементов. Этот метод блокирует выполнение дальнейшего кода до тех пор, пока пользователь не закроет окно или не завершится работа программы. Без вызова mainloop(), окно не будет отображаться и пользователь не сможет взаимодействовать с интерфейсом.

# Тестирование приложения

Для тестирования приложения можно решить задачу. Предположим, у нас есть данные о количестве часов, которые студенты проводят ежедневно на подготовку к экзамену и оценках, которые они получают на экзамене. Мы хотим узнать, существует ли связь между количеством времени, проводимым на подготовку к экзамену, и оценками на экзамене. Рассчитаем ранговый коэффициент корреляции Спирмена для этого. Предположим, что у нас есть следующие данные:

|  |  |  |
| --- | --- | --- |
| Студент | Кол-во часов | Оценка на экзамене |
| 1 | 10 | 90 |
| 2 | 5 | 75 |
| 3 | 15 | 95 |
| 4 | 8 | 80 |
| 5 | 12 | 85 |

Ранжирование данных по рангам:

|  |  |  |  |
| --- | --- | --- | --- |
| Студент | Кол-во часов(ранг) | Оценка на экзамене(ранг) |  |
| 1 | 3 | 4 | 1 |
| 2 | 1 | 1 | 0 |
| 3 | 5 | 5 | 0 |
| 4 | 2 | 2 | 0 |
| 5 | 4 | 3 | 1 |

Вычисляем ранговый коэффициент корреляции Спирмена по формуле:

Теперь решим данную задачу в приложении:

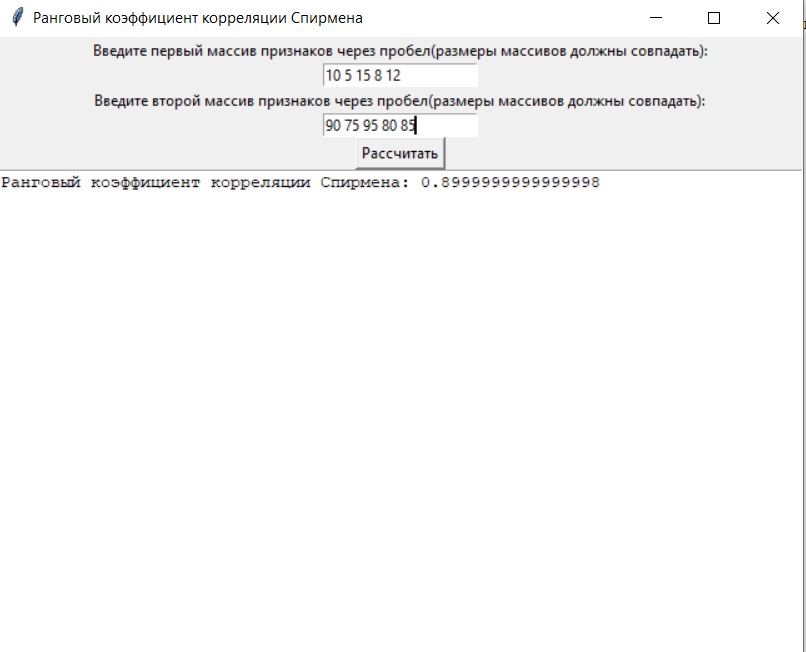


Рисунок 6 – Результат первого тестирования

Результат получился таким же с разностью в 10-15. Математический расчет может быть выполнен с большей точностью, чем метод в Python, который использует численные алгоритмы и округления для решения задачи.

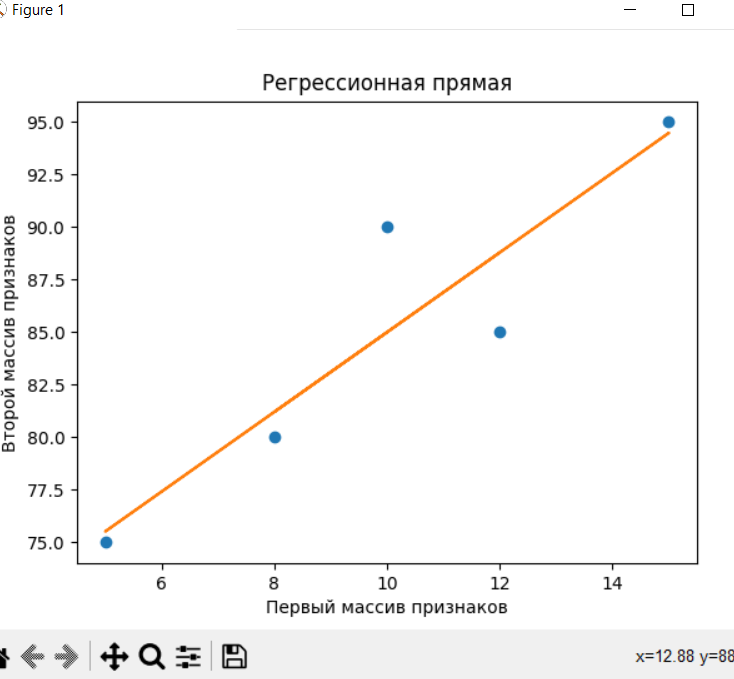


Рисунок 7 – График первого тестирования

Также получаем график, на котором видно что два массива признаков(часы и оценка) почти прямо пропорциональны друг другу.

Рассмотрим ещё одну задачу. У нас есть данные о температуре и количестве продаж мороженого в магазине в течение года. Мы можем использовать коэффициент корреляции Спирмена, чтобы определить, есть ли связь между этими переменными и насколько сильна эта связь.

Например, предположим, что мы имеем данные о температуре и продажах мороженого для 12 месяцев. Мы можем использовать коэффициент корреляции Спирмена, чтобы определить, есть ли связь между этими двумя переменными. Если коэффициент корреляции Спирмена близок к 1, то это означает, что есть положительная связь между температурой и количеством продаж мороженого, то есть чем выше температура, тем больше продаж мороженого. Если коэффициент корреляции Спирмена близок к -1, то это означает, что есть отрицательная связь между температурой и количеством продаж мороженого, то есть чем выше температура, тем меньше продаж мороженого. Если коэффициент корреляции близок к нулю, то это означает, что между этими переменными нет связи.

Предположим, что у нас есть следующие данные:

|  |  |  |
| --- | --- | --- |
| Месяц | Средняя температура.°C | Продажи мороженного |
| Январь | -12 | 142 |
| Февраль | -15 | 124 |
| Март | -5 | 186 |
| Апрель | 3 | 253 |
| Май | 12 | 354 |
| Июнь | 20 | 574 |
| Июль | 25 | 512 |
| Август | 23 | 578 |
| Сентябрь | 14 | 421 |
| Октябрь | 3 | 257 |
| Ноябрь | -1 | 142 |
| Декабрь | -10 | 96 |

Ранжирование данных по рангам:

|  |  |  |  |
| --- | --- | --- | --- |
| Месяц | Средняя температура(ранг) | Продажи мороженного(ранг) |  |
| Январь | 2 | 3.5 | 2,25 |
| Февраль | 1 | 2 | 1 |
| Март | 4 | 5 | 1 |
| Апрель | 6.5 | 6 | 0,25 |
| Май | 8 | 8 | 0 |
| Июнь | 10 | 11 | 1 |
| Июль | 12 | 10 | 4 |
| Август | 11 | 12 | 1 |
| Сентябрь | 9 | 9 | 0 |
| Октябрь | 6.5 | 7 | 0,25 |
| Ноябрь | 5 | 3.5 | 2,25 |
| Декабрь | 3 | 1 | 4 |

Вычисляем ранговый коэффициент корреляции Спирмена по формуле:

Результат работы приложения:

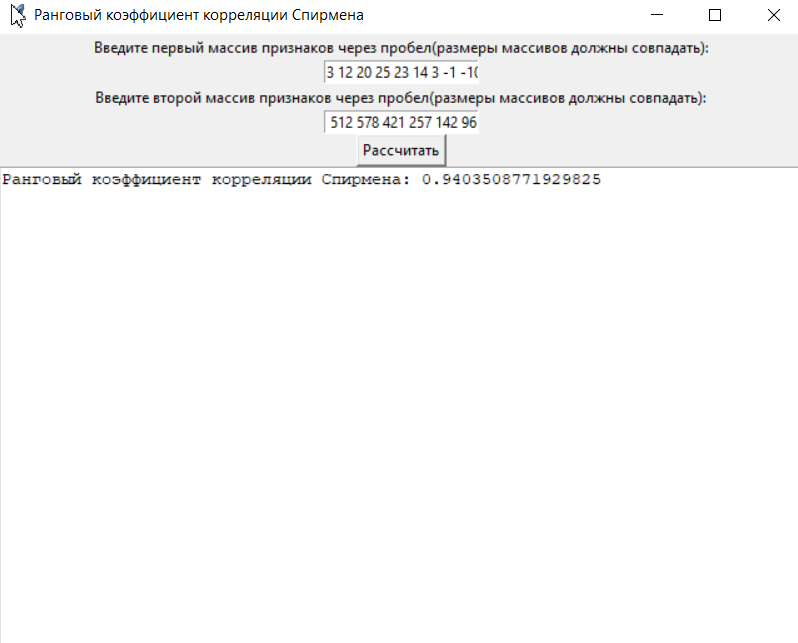


Рисунок 8 – Результат второго тестирования

Графический результат:

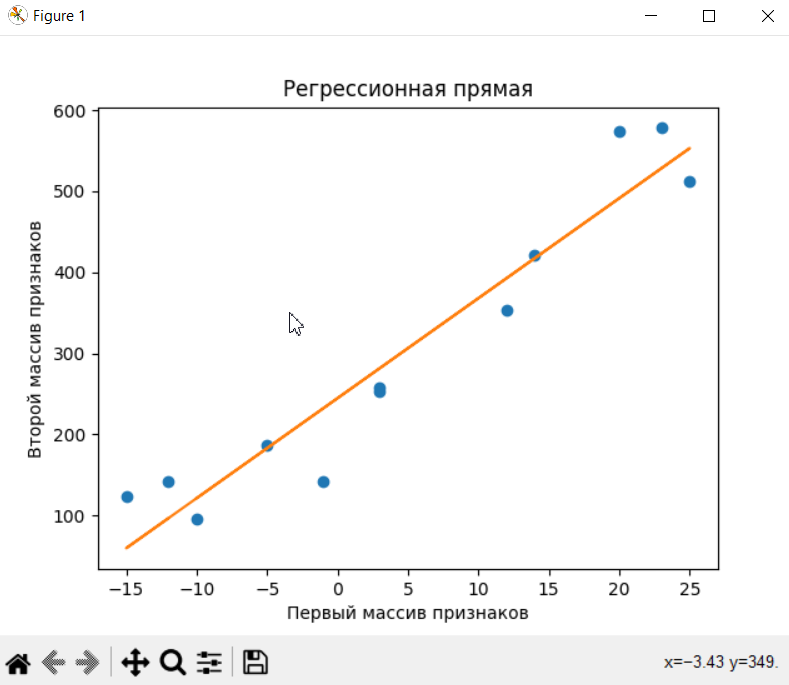


Рисунок 9 – График второго тестирования

Результат работы приложения снова сошёлся с математическими расчётами. Он показывает что связь между продажами мороженного и температурой связаны положительно. Это видно по таблице. После тестирования на двух задачах можно прийти к выводу, что программа работает корректно.

# Заключение

В ходе выполнения работы был изучен ранговый коэффицент корреляции Спирмена, сферы его применения, алгоритм расчёта и разработано приложение на языке Python для его расчёта.

В целом, ранговый коэффициент корреляции Спирмена является эффективным инструментом для измерения связи между двумя переменными, особенно в тех случаях, когда данные не соответствуют требованиям для использования других показателей корреляции. Однако стоит отметить, что ранговый коэффицент корреляции Спирмена эффективен только при прямой зависимости между признаками. Если зависимость между признаками имеет сложную логику, то данный способ расчёта взаимосвязи применять не нужно.

Код на Python получился небольшим, так как благодаря функциям с импортируемых библиотек не пришлось прописывать математический алгоритм расчёта. Также расчёт с помощью встроенных функций позволяет работать приложению быстрее, чем при расчёте с помощью математического алгоритма. Это будет иметь значение при анализе больших данных. Результаты расчёта не отличаются от математического способа.

Приложение можно впоследствии доработать в направлении обработок ошибок и удобства интерфейса.

# Библиографический список

1. [https://www.psychol-ok.ru/statistics/spearman/ [Электронный](https://www.psychol-ok.ru/statistics/spearman/%20%5bЭлектронный) **ресурс].** [Коэффициент ранговой корреляции Спирмена. (дата обращения: 07.05.2023)](http://www.yandex.ru/clck/jsredir?from=www.yandex.ru%3Bsearch%2F%3Bweb%3B%3B&text=&etext=2202.JUVXDXuxnyTehYcvK1RCkMfPDbSBI-e9UQ0VbXghBL-GfFBGixSML-mYjm-vOBtF5ZUftOfz91TOvsUK_nqJ0H0Yce8SbkYtHPVe74GquNxqbWNpanpzdGtsdmZvamh2.50348e191541f5dadb2a4fd84be2736a0ffcaa31&uuid=&state=RsWHKQP_fPE,&&cst=AxbTlK7nwx4MdSOJ1ERoFmWJV-TtvpnQsYUaKkPHsji3J2Y3qCDRJUTVJuAJpj0Iy1YjaShmVBG6d2xvq1JW74AIWCzEdSXIwomMY1RPHFBMvHPHfTXW6l0u-1oC4-VqksE2IqC7avnpi5xR4jcP_wBczPQ38_u1SHqrmM5YxVII5WTtyXF2Hthhno5KbUjvCs8iAyFVt_a828yWxwzD5UKijFaPXCSBr7BeOFJetInx9ePxKzF_HposcsdpsbFSI8YkKeYIA73Z7B3vJj5PueYQt3t2TmIdInZnBm7s52wvuzGiYnl38jL4NaIZxxrkc3sqIvCnvZrCUPMntJVV11rXVMJrr8WfE294hEW2hpgRMx6uFRRliqVkEB37lz4dEkLFL2jjFbZoupaApoUQ_J8Ksyh--wOwEefSwgHCeOulPQ7YtBv2kdsrYOi2oAb4d4wDubNt6RLCpxii2DIJu_u-EuFJ2-XvXCcE9oJAOnaVUCjPMA0F681Q3B6t2-eqm20qtfChst2SuYZvMabBvwaSOySn4Um2Zou8t84XuORUCNA5WU7ogZLlQqqUEenAETXdVzLtZ6Q5M1Vcu3fUgy-ZP4nWGnd9DVGdr_AKXU7NJkm3PDkNkkMpXX-PhdKcb190h4OgKB483mUa37yAz6WGfEIoDcmtcGUAkjxheUprC6jWyLnCHitLxoyYoVzpRK_iMI0mDAgpZM0jDY8x_5nVIWmwJn49aelz-NpKu4lXG183NXA6XCyMEJj0pfuFrxdRMGd0wPoZn8IOTcsJEvcZaTy7xUfpVflRCXZ28mNqCSEehsV2isyiGY0OpFrBXHWIuh2z1r6L9rOkTmCaRIJ1ZdoUj2E3XB_F98uJYXKWXiP3GhNiJx5p8Q3mMkJBE92S4Xwh2xbuwwMDPaEizufdob42E-KGxqWoxS3QarargkZujDmIGZPO0QsTeFEQfcxM6fJ1dXaBfLRsBYewxIN53k-MeE8vTt7OmmTgvhO27EwS4T3xX5MnMpGON3kk&data=VzFITjJTUER3MkI4MEY5djBaZUVGOEs2YmN4VmFHcTQtOC1MUmNkLWVfaGNVRzlJTGhBaVFDdEFQMTJ3cV9HSWM2YVlvbkpqY1pJRFM0cnZGRXdoQl9qX1NQaXgwX1pYVmd3SEI2Vmx3N2NMcDlaY1VMVjM4LUp3V3huSmtiMXNFXzJ3YVFHOGFfSSw,&sign=b25367a597d7c12479f7c84ae900317c&keyno=WEB_0&b64e=2&ref=mag21uLwzH-rYVS7bq28kWLw8trSEHW3dPBfrt6pjwERQqMFOez8XL-H_ux8xWNZmjH3usNhVk667CzZfjR_pK7OJqk1yODHhxvotVE_2WQcjgVLRcsksGRerdrcwjTAp2U0YQ1CS_ekScdTai4mt8xjiOxHoZRXo5qwolgFqHaYTz6Nf6hxUKcYKUcxt4GckZuKu3FV_l3Nwr0xP1D8A1pS4R2JgA40KVKG1nwntGnpnaAymofcvGReb08Y9ySGuWe6TM_q1YFWCKLlkiZr8TZ15M44XZ6O&l10n=ru&cts=1683593994433%40%40events%3D%5B%7B%22event%22%3A%22click%22%2C%22id%22%3A%221_9kraw02-01%22%2C%22cts%22%3A1683593994433%2C%22fast%22%3A%7B%22organic%22%3A1%7D%2C%22service%22%3A%22web%22%2C%22event-id%22%3A%22lhfke85dj2%22%7D%5D&mc=3.202819531114783&hdtime=5964.5" \t "_blank)
2. <https://medstatistic.ru/methods/methods9.html>. [Электронный ресурс] [Как рассчитать коэффициент Спирмена? (дата обращения: 07.05.2023)](http://www.yandex.ru/clck/jsredir?from=www.yandex.ru%3Bsearch%2F%3Bweb%3B%3B&text=&etext=2202.JUVXDXuxnyTehYcvK1RCkMfPDbSBI-e9UQ0VbXghBL-GfFBGixSML-mYjm-vOBtF5ZUftOfz91TOvsUK_nqJ0H0Yce8SbkYtHPVe74GquNxqbWNpanpzdGtsdmZvamh2.50348e191541f5dadb2a4fd84be2736a0ffcaa31&uuid=&state=RsWHKQP_fPE,&&cst=AxbTlK7nwx4MdSOJ1ERoFmWJV-TtvpnQsYUaKkPHsji3J2Y3qCDRJUTVJuAJpj0Iy1YjaShmVBG6d2xvq1JW74AIWCzEdSXIwomMY1RPHFBMvHPHfTXW6l0u-1oC4-VqksE2IqC7avnpi5xR4jcP_wBczPQ38_u1SHqrmM5YxVII5WTtyXF2Hthhno5KbUjvCs8iAyFVt_a828yWxwzD5UKijFaPXCSBr7BeOFJetInx9ePxKzF_HposcsdpsbFSI8YkKeYIA73Z7B3vJj5PueYQt3t2TmIdInZnBm7s52wvuzGiYnl38jL4NaIZxxrkc3sqIvCnvZrCUPMntJVV11rXVMJrr8WfE294hEW2hpgRMx6uFRRliqVkEB37lz4dEkLFL2jjFbZoupaApoUQ_J8Ksyh--wOwEefSwgHCeOulPQ7YtBv2kdsrYOi2oAb4d4wDubNt6RLCpxii2DIJu_u-EuFJ2-XvXCcE9oJAOnaVUCjPMA0F681Q3B6t2-eqm20qtfChst2SuYZvMabBvwaSOySn4Um2Zou8t84XuORUCNA5WU7ogZLlQqqUEenAETXdVzLtZ6Q5M1Vcu3fUgy-ZP4nWGnd9DVGdr_AKXU7NJkm3PDkNkkMpXX-PhdKcb190h4OgKB483mUa37yAz6WGfEIoDcmtcGUAkjxheUprC6jWyLnCHitLxoyYoVzpRK_iMI0mDAgpZM0jDY8x_5nVIWmwJn49aelz-NpKu4lXG183NXA6XCyMEJj0pfuFrxdRMGd0wPoZn8IOTcsJEvcZaTy7xUfpVflRCXZ28mNqCSEehsV2isyiGY0OpFrBXHWIuh2z1r6L9rOkTmCaRIJ1ZdoUj2E3XB_F98uJYXKWXiP3GhNiJx5p8Q3mMkJBE92S4Xwh2xbuwwMDPaEizufdob42E-KGxqWoxS3QarargkZujDmIGZPO0QsTeFEQfcxM6fJ1dXaBfLRsBYewxIN53k-MeE8vTt7OmmTgvhO27EwS4T3xX5MnMpGON3kk&data=VzFITjJTUER3MkI4MEY5djBaZUVGNTQwNHliRlY2RDBaMWV2OGNKNS04QnBhb2JCTGdtaThQNHFmTXRtclZlektiRkN2ZkdIdmZaSFVjNU1ncGlYd3JmQnFnOFBWcG5xbzlzeGNGTGhXSklzc2ZVUGcwaDIwbDA4OThZbFl1Y1NCenZKRW9qT2FEdyw,&sign=17ae20e06d4b983a28862e355eccfdd6&keyno=WEB_0&b64e=2&ref=mag21uLwzH-rYVS7bq28kWLw8trSEHW3dPBfrt6pjwERQqMFOez8XL-H_ux8xWNZmjH3usNhVk667CzZfjR_pK7OJqk1yODHhxvotVE_2WQcjgVLRcsksGRerdrcwjTAp2U0YQ1CS_ekScdTai4mt8xjiOxHoZRXo5qwolgFqHaYTz6Nf6hxUKcYKUcxt4GckZuKu3FV_l3Nwr0xP1D8A1pS4R2JgA40KVKG1nwntGnpnaAymofcvGReb08Y9ySGuWe6TM_q1YFWCKLlkiZr8TZ15M44XZ6O&l10n=ru&cts=1683594144714%40%40events%3D%5B%7B%22event%22%3A%22click%22%2C%22id%22%3A%222_9kraw02-01%22%2C%22cts%22%3A1683594144714%2C%22fast%22%3A%7B%22organic%22%3A1%7D%2C%22service%22%3A%22web%22%2C%22event-id%22%3A%22lhfkhg3ugc%22%7D%5D&mc=4.555487143479848&hdtime=156244.7" \t "_blank)
3. <https://skillbox.ru/media/code/biblioteka-numpy-vsye-chto-nuzhno-znat-novichku/>[Электронный ресурс]. Numpy. Гайд по библиотеке. (дата обращения: 07.05.2023)
4. <https://www.python.org/>.[Электронный ресурс]  [The official home of the Python Programming Language. (дата обращения: 07.05.2023)](http://www.yandex.ru/clck/jsredir?from=www.yandex.ru%3Bsearch%2F%3Bweb%3B%3B&text=&etext=2202.JAK7W_LsO7WuaijDPW2e1HpsbWFqaGNkeWZ1ZWJ4cm0.3495c3dd5954cc9a42e4af555f16c2f50acc4b92&uuid=&state=RsWHKQP_fPE,&&cst=AxbTlK7nwx4MdSOJ1ERoFmWJV-TtvpnQsYUaKkPHsji3J2Y3qCDRJUTVJuAJpj0Iy1YjaShmVBG6d2xvq1JW74AIWCzEdSXIwomMY1RPHFBMvHPHfTXW6l0u-1oC4-VqksE2IqC7avnpi5xR4jcP_wBczPQ38_u1SHqrmM5YxVII5WTtyXF2Hthhno5KbUjvCs8iAyFVt_a828yWxwzD5UKijFaPXCSBr7BeOFJetInx9ePxKzF_HposcsdpsbFSI8YkKeYIA73Z7B3vJj5PueYQt3t2TmIdInZnBm7s52wvuzGiYnl38jL4NaIZxxrkc3sqIvCnvZrCUPMntJVV11rXVMJrr8WfE294hEW2hpgRMx6uFRRliqVkEB37lz4dEkLFL2jjFbZoupaApoUQ_J8Ksyh--wOwEefSwgHCeOulPQ7YtBv2kdsrYOi2oAb4d4wDubNt6RLCpxii2DIJu_u-EuFJ2-XvXCcE9oJAOnaVUCjPMA0F681Q3B6t2-eqm20qtfChst2SuYZvMabBvwaSOySn4Um2Zou8t84XuORUCNA5WU7ogZLlQqqUEenAETXdVzLtZ6Q5M1Vcu3fUgy-ZP4nWGnd9DVGdr_AKXU7NJkm3PDkNkkMpXX-PhdKcb190h4OgKB483mUa37yAz6WGfEIoDcmtcGUAkjxheUprC6jWyLnCHitLxoyYoVzpRK_iMI0mDAgpZM0jDY8x_5nVIWmwJn49aelz-NpKu4lXG183NXA6XCyMEJj0pfuFrxdRMGd0wPoZn8IOTcsJEvcZaTy7xUfpVflRCXZ28mNqCSEehsV2isyiGY0OpFrBXHWIuh2z1r6L9rOkTmCaRIJ1ZdoUj2E3XB_F98uJYXKWXiP3GhNiJx5p8Q3mMkJBE92S4Xwh2xbuwwMDPaEizqi60KlMidk8ikoM183iBotqjgpF3c5eVYQymUzulo8FIokB_Lr8CS0K_UZIH5W1iPZba2iw6PuFY74S8U7vBXeW9un6JXqTWeau9Uy2DAdq&data=VzFITjJTUER3MkI4MEY5djBaZUVGeFRNNHBCcmxMS2I4TlNJbUdQNUVtRm5wLXJIVGx6N0l4M1J3OVRad2U1S0hrSjhOVDNUUFkyUVF6eXU5RUdnMGl3bVUzZ05keUdaQVVmMGxYWUp4X0Us&sign=0019b0dda6688d2ea3bff72f943a7077&keyno=WEB_0&b64e=2&ref=mag21uLwzH-rYVS7bq28kWLw8trSEHW3dPBfrt6pjwHe0xlffdthuC64FvziCL8M&l10n=ru&cts=1683594303873%40%40events%3D%5B%7B%22event%22%3A%22click%22%2C%22id%22%3A%222_fq98w02-01%22%2C%22cts%22%3A1683594303873%2C%22fast%22%3A%7B%22organic%22%3A1%7D%2C%22service%22%3A%22web%22%2C%22event-id%22%3A%22lhfkkuwx8b%22%7D%5D&mc=4.056020968057881&hdtime=20483.2" \t "_blank)
5. https://pythonru.com/uroki/obuchenie-python-gui-uroki-po-tkinter. [Электронный ресурс]. Обучение Python GUI (уроки по Tkinter). (дата обращения: 07.05.2023)
6. https://pythonru.com/biblioteki/pyplot-uroki. [Электронный ресурс]. Как строить графики с Pyplot / plt 2. (дата обращения: 07.05.2023)

# Приложение

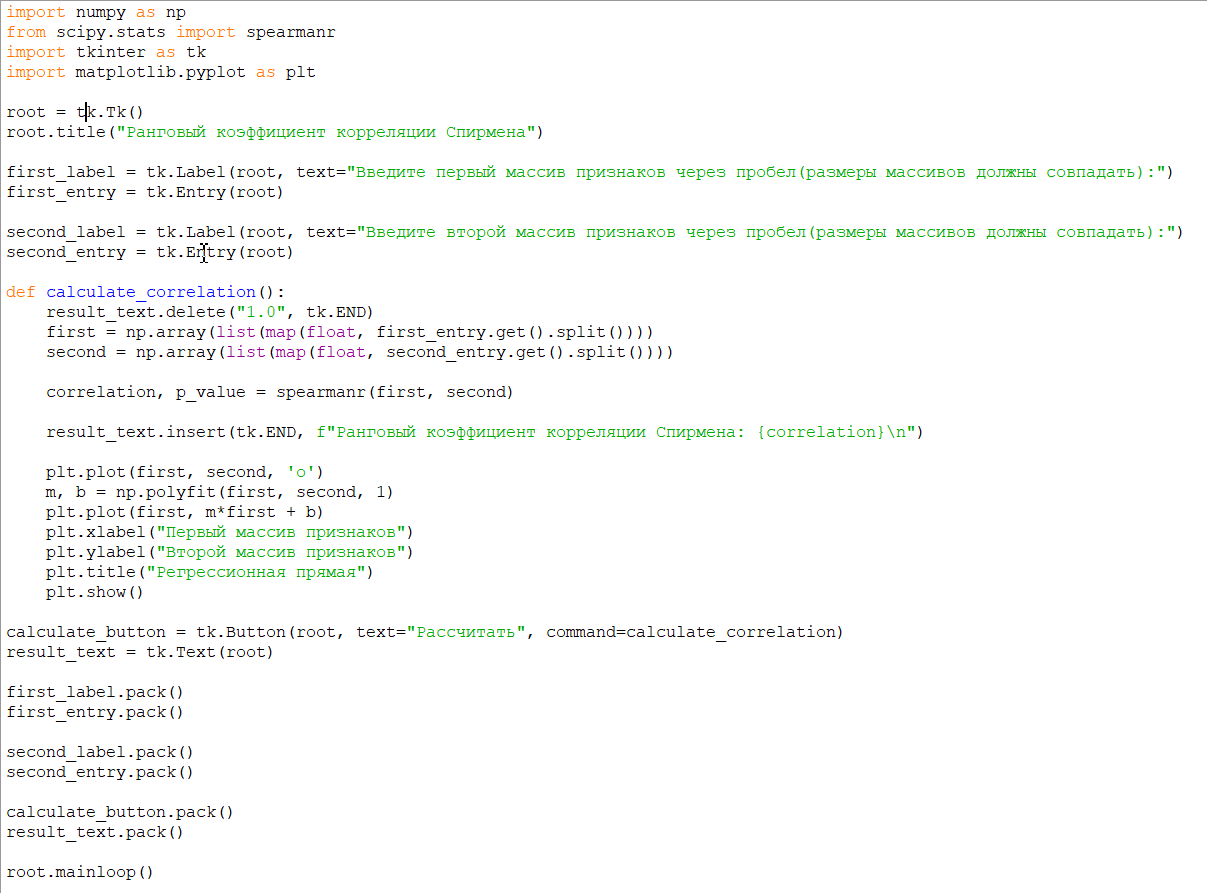


Рисунок 10 – Код программы.