Российский университет дружбы народов Научный факультет

Моделирование беспроводных сетей

Лабораторная работа №1

Подготовлено студентом:

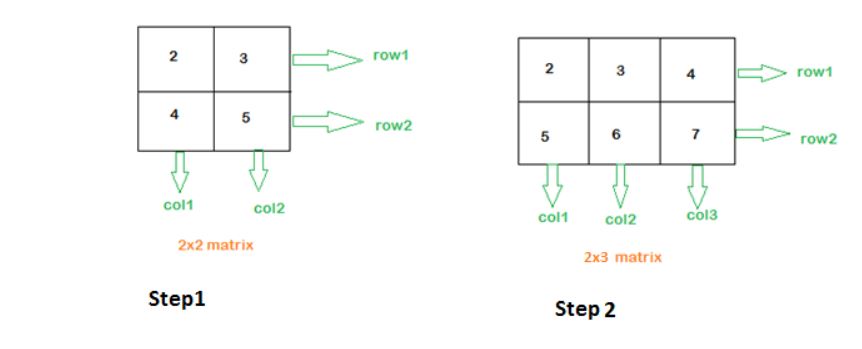
Елиенис Санчес Родригес.

Преподаватель: Вячеслав Бегишев

Массив - это упорядоченный набор данных (как примитивов, так и объектов в зависимости от языка). Массивы (массивы) используются для хранения нескольких значений в одной переменной, в отличие от переменных, которые могут хранить только одно значение (для каждой переменной).

Когда мы говорим о массивах в Python, мы имеем в виду специализированный двумерный прямоугольный массив данных, оба из которых хранятся в строках и столбцах. Внутри этого массива могут быть данные в виде чисел, строк, символов, выражений и т. Д. матрица - одна из важных структур данных, которую можно использовать в математических и научных расчетах.

Данные в матрице выглядят примерно так, как показано на этом графике



[https://cms.rootstack.com/sites/default/files/inline-images/Matriz python.JPG](https://cms.rootstack.com/sites/default/files/inline-images/Matriz%20python.JPG)

Первый Шаг:

Здесь показана матрица 2x2, в которой видны две строки и два столбца. Данные в этой матрице представлены в виде чисел, где в первой строке отмечены значения 2, 3, а во второй строке-значения 4, 5. В столбцах находится первый столбец со значениями 3 и 4 и второй столбец со значениями 3 и 5.

Второй Шаг:

Отображается немного другая матрица размером 2 x 3 с двумя строками и тремя столбцами. Данные в первой строке имеют значения 2, 3, 4, а во второй строке-значения 5, 6, 7. Первый столбец имеет значения 2,5, второй столбец имеет значения 3,6, а третий столбец имеет значения 4,7.

Точно так же вы можете хранить свои данные в массиве nxn в Python. При сложении, вычитании, умножении и т. Д. Можно выполнять множество операций.

**Задача 1**

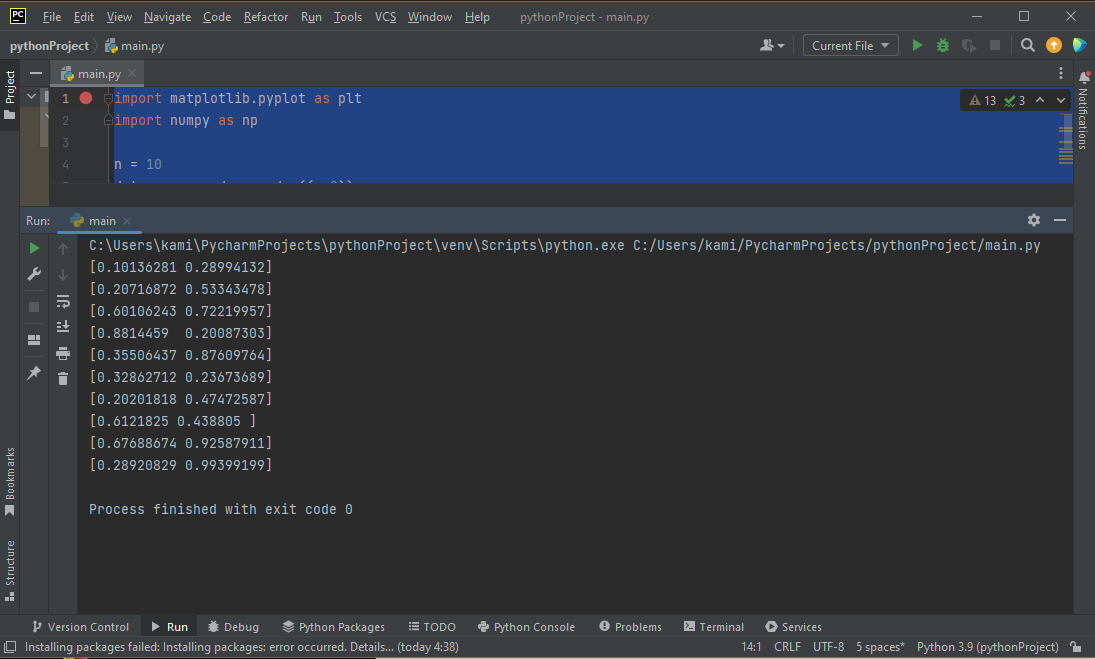
Постройте матрицу с количеством строк 10 и столбцов 2, переменные

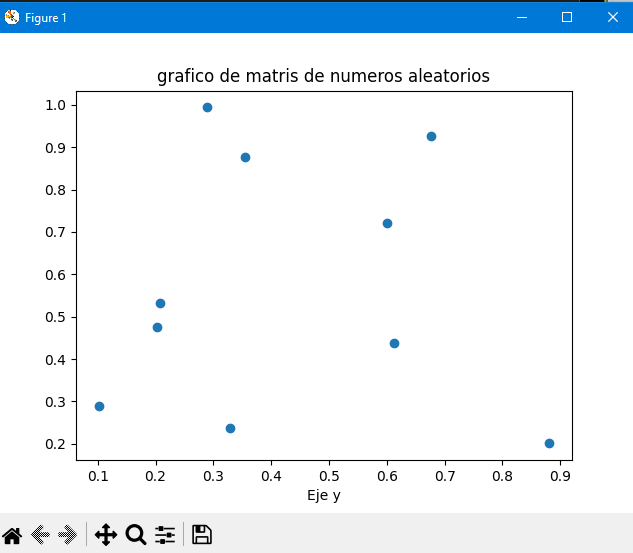
заполняются случайными числами с равномерным распределением.

Полученные в результате сгенерированные случайные числа представляют на графике в

виде точек.

import matplotlib.pyplot as plt  
import numpy as np  
  
n = 10  
data = np.random.random((n,2))  
colors = [ "red","green"]  
color\_filt = []  
for line in data:  
 print(line)  
  
 plt.scatter(data[:,0], data[:,1])  
 plt.xlabel("Eje x")  
 plt.xlabel("Eje y")  
 plt.title("grafico de matris de numeros aleatorios")  
 plt.show()





Python определяет набор функций, которые используются для генерации или управления случайными числами с помощью случайного модуля. Функции в случайном модуле основаны на функции генератора псевдослучайных чисел random() , которая генерирует случайное число с плавающей запятой от 0,0 до 1,0. Этот конкретный тип функций используется во многих играх, лотереях или любых приложениях, требующих генерации случайных чисел.

Гистограммы позволяют получить общее представление о существующем распределении в выборке данных. Для чего вы рисуете столбцы, связанные с диапазоном значений, высота которых пропорциональна частоте их появления

В математике, в частности в области статистики, ожидаемое значение (также называемое ожидаемым значением, средним по совокупности или средним) случайной величины

{\displaystyle X} X , это число

{\displaystyle \mathbb {E} [X]}{\displaystyle \mathbb {E} [X]}

o {\displaystyle {\text{E}}[X]}{\displaystyle {\text{E}}[X]}

который формализует идею среднего значения случайного явления. Это концепция, аналогичная среднему арифметическому набора данных.

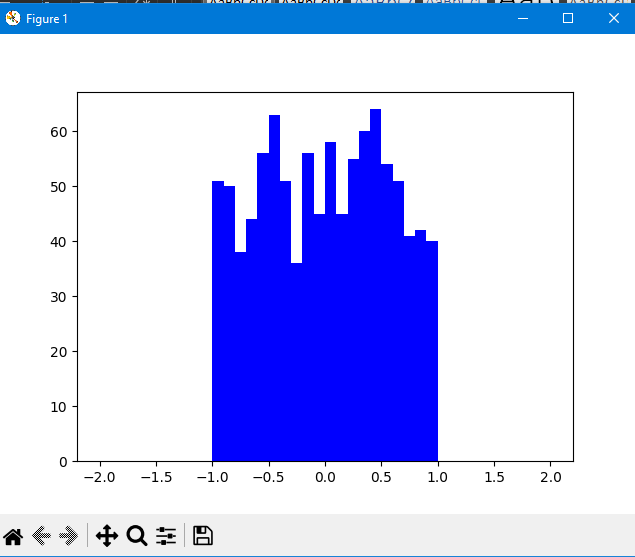
Чтобы вычислить дисперсию в Python, вы можете использовать функцию var () модуля numpy. Аргумент x - это список или массив, содержащий элементы дистрибутива. Функция var () возвращает статистическую дисперсию распределения.

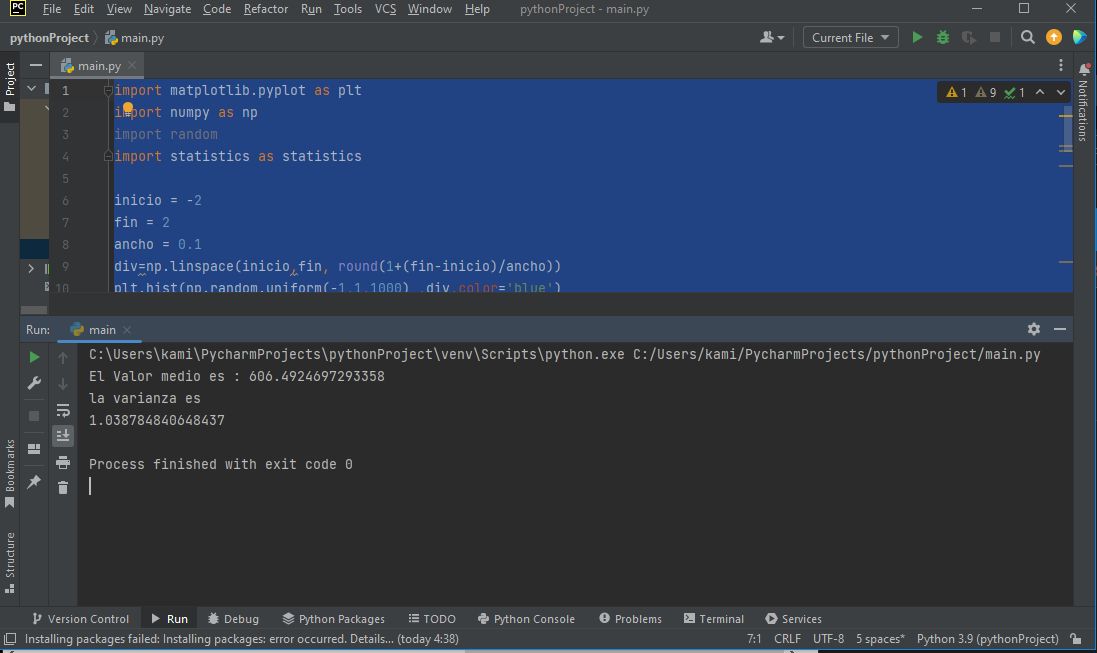
Задача 2

Сгенерируйте 1000 случайных чисел с любым известным распределением

и постройте свою гистограмму, математическое ожидание, дисперсию.

import matplotlib.pyplot as plt  
import numpy as np  
import random  
import statistics as statistics  
  
inicio = -2  
fin = 2  
ancho = 0.1  
div=np.linspace(inicio,fin, round(1+(fin-inicio)/ancho))  
plt.hist(np.random.uniform(-1,1,1000) ,div,color='blue')  
plt.show()  
  
import random  
  
num = 4  
  
strt\_num = 1  
  
end\_num = 1000  
  
res = 0.0  
for \_ in range(num):  
 res += random.uniform(strt\_num, end\_num)  
  
res = res / num  
  
print("El Valor medio es : " + str(res))  
  
y = [random.gauss(5, 1) for \_ in range(1000)]  
statistics.variance(y)  
print("la varianza es")  
print(statistics.variance(y))





Случайные точки

Область, в которой будут создаваться случайные точки, может быть определена с помощью обязательных многоугольных, точечных или линейных объектов или с помощью окна обязательного расширения.

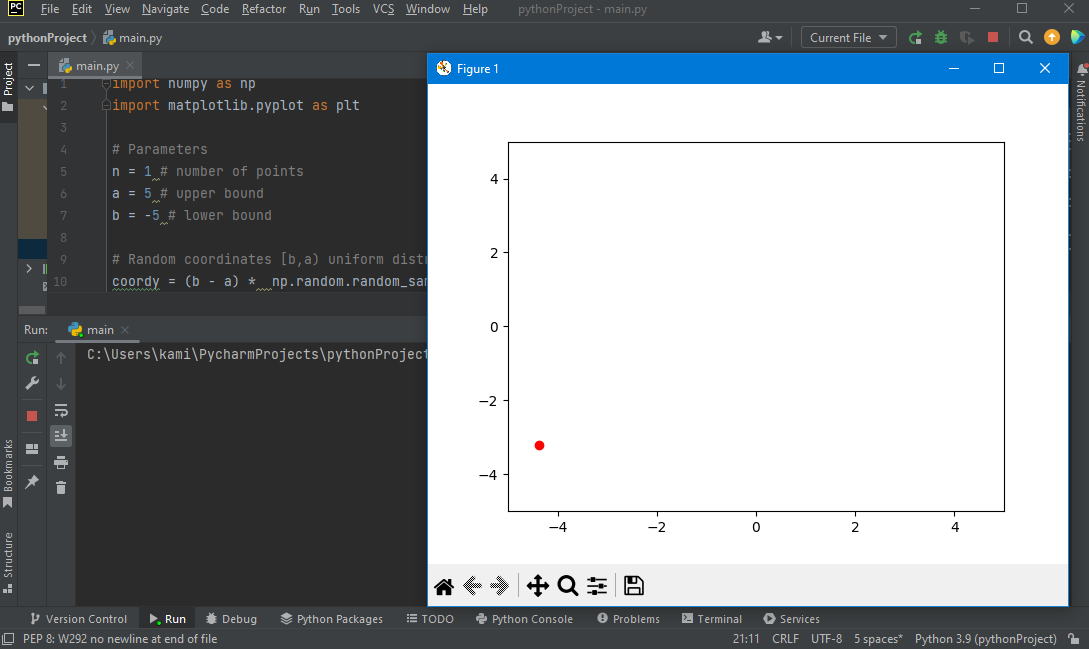
Параметр Количество точек может быть задан как число или как числовое поле обязательного класса объектов, содержащее значения количества случайных точек, которые должны быть расположены в каждом объекте. Параметр поля действителен только для обязательных объектов многоугольника или линии. Если количество точек указано в виде числа, это количество случайных точек будет сгенерировано внутри или вдоль каждого объекта в связанном классе объектов.

**Задача 3**

Создать случайную точку, равномерно распределенную по квадрату

со стороной а.

import numpy as np  
import matplotlib.pyplot as plt  
  
# Parameters  
n = 1 # number of points  
a = 5 # upper bound  
b = -5 # lower bound  
  
# Random coordinates [b,a) uniform distributed  
coordy = (b - a) \* np.random.random\_sample((n,)) + a # generar aleatorio y  
coordx = (b - a) \* np.random.random\_sample((n,)) + a # generate aleatorio x  
  
# Create limits (x,y)=((-5,5),(-5,5))  
plt.xlim((b,a))  
plt.ylim((b,a))  
  
# Plot points  
for i in range(n):  
 plt.plot(coordx[i],coordy[i],'ro')  
  
plt.show()



Вывод

На протяжении многих лет Python поднимался по карьерной лестнице, завоевывая огромную популярность и заставляя дрожать те языки, которые, казалось, останутся навсегда. Это простой в освоении язык, который идеально подходит для начинающих программистов. Он имеет большое разнообразие и бесконечное количество применений, имеет очень активное сообщество, что гарантирует, что язык будет постоянно обновляться с течением времени, и что появятся новые библиотеки, которые позволят нам сэкономить время и труд.

Кроме того, как мы уже видели, Python предлагает множество преимуществ для всех своих пользователей, в том числе: его большую простоту, гибкость и простоту обучения. Благодаря прямому синтаксису и хорошему использованию пробелов с отступами, их очень легко выучить, прочитать и даже поделиться. В настоящее время в его онлайн-репозитории насчитывается более 145 000 библиотек, которые удовлетворяют практически любые потребности, от возможности разработки видеоигр до разработки игр.

была продемонстрирована реализация различных типов функций, которые способны моделировать построение массивов с помощью кода Python. Генерация случайных чисел, построение графиков и вероятностные вычисления, такие как среднее и медиана полученных данных

Python всегда был хорошим союзником системных администраторов и операционных групп, вплоть до замены сценариев, написанных на bash. Но это не только ограничивается этим, но и является пионером в мире сканирования и сканирования, где мы можем получать информацию с веб-страниц очень простым и полностью автоматизированным способом

Библиография

Perez P, B., & Acosta Velarde, R. (2019). Mejora en la seguridad python . Ciencia Digital, 2(3), 61-74. https://inventwithpython.com/cracking/chapter7.html

Cabrera R, Juan , python-transposition (2017). método transposición. Ciencia Digital, 8(5),

https://blog.finxter.com/python-transposition-algorithm/

S Paulina . (2017). Programacion de 0 aplicada. https://github.com/indetectablesnet/

manuals/blob/master/Programacion/Python/Algoritmos%20y%20Programaci%C3%B3n%201/algoritmos-programacion-1.pdf

Cley R . Naranjo, Velarde,Q. (2016). Secretos informáticos . Ciencia Digital, 62-

https://studylib.es/doc/9203034/la-biblia-de-los-c%C3%B3digos-secretos---herv%C3%A9-lehning