

Лабораторная работа №4

Работа с библиотеками Python

Полезные ссылки:

[Знакомимся с NumPy](#)

[Знакомимся с Pandas](#)

[Знакомимся с SciPy](#)

[Знакомимся с Matplotlib](#)

[И еще немного о Matplotlib](#)

[ТОП-6 библиотек для визуализации данных](#)

Установка пакетов (библиотек)

По умолчанию в Python в качестве менеджера пакетов используется `pip`. Чтобы установить библиотеку с помощью `pip` в командной строке нужно выполнить следующую команду:

```
pip install <package_name>
```

где `<package_name>` — имя требуемой библиотеки. Например, для загрузки библиотеки `pandas` используйте следующую команду:

```
pip install pandas
```

Эта команда установит последнюю версию библиотеки. Но если нужна какая-то конкретная версия, отличная от последней, ее можно указать после двойного равно:

```
pip install pandas==2.2.0
```

Если скачиваемый пакет имеет зависимости от других пакетов, они автоматически «подтянутся» вместе с установкой первого. Хотя в редких случаях это приходится делать вручную. Для просмотра всех установленных пакетов используйте:

```
pip list
```

Так же в реальных проектах на Python обычно создается файл со всеми зависимостями, используемыми в данном проекте. Типичное название такого файла — `requirements.txt`. Он содержит список всех требуемых для запуска проекта библиотек с указанием их версий. Чтобы установить все библиотеки из данного файла, используйте команду:

```
pip install -r requirements.txt
```

Создать свой файл с зависимостями проекта можно следующей командой:

```
pip freeze > requirements.txt
```

Эта команда создаст список из всех библиотек, которые установлены для используемого интерпретатора.

По умолчанию все пакеты скачиваются в основной интерпретатор, находящийся в директории с Python. Однако различные проекты требуют не только разного набора библиотек для запуска, но и использование различных версий одних и тех же библиотек. Это означает, что при чередовании работы между двумя и более проектами, вам придется каждый раз загружать нужные версии библиотек для проекта. Проект может требовать использования как более новых версий пакетов (с новым функционалом), так и более старых версий (где нужный функционал из библиотеки еще не удален). Также сами

библиотеки могут перестать корректно работать, если пакеты, от которых первые зависят, будут несовместимы по версиям.

Для решения данной проблемы для каждого проекта создается свое **виртуальное окружение** перед началом работы над ним.

Виртуальное окружение Python

Виртуальное окружение позволяет изолировать зависимости разных проектов, избегая конфликтов между библиотеками. Оно изолирует пакеты, требуемые для конкретного проекта, от основного интерпретатора. Для создания своего виртуального окружения выполните следующие действия:

1. В командной строке (или через проводник) создайте папку, в которой будет лежать проект:

```
mkdir my_ml_project
```

```
cd my_ml_project
```

2. В командной строке введите команду для создания виртуального окружения (вместо **<your_directory_name>** введите своё название, обычно используется `venv`):

```
python -m venv <your_directory_name>
```

3. Активируйте виртуальное окружение:

a. Windows: `<your_directory_name>\Scripts\activate`

b. Linux/MacOS: `source <your_directory_name>/bin/activate`

После выполнения последнего пункта командная строка примет следующий вид — в начале командной строки появится надпись формата “`(.<your_directory_name>)`”, в данном случае “`(.venv)`”:



```
(.venv) PS C:\Users\User\PycharmProjects\Lecture>
```

Теперь все ваши пакеты будут загружаться в папку виртуального окружения, при условии, что вы работаете в командной строке с запущенным виртуальным окружением.

Также, чтобы ваша IDE использовала зависимости из виртуального окружения требуется:

1. Для JupyterLab — запустить веб-сервер JupyterLab из активированного виртуального окружения.

2. Для PyCharm — при создании проекта в PyCharm, виртуальное окружение автоматически создается в той же папке. Для использования данного виртуального окружения либо активируйте его через командную строку, либо используете встроенный терминал в PyCharm, где виртуальное окружение уже запущено.

Часть 1. Mathplotlib

Задача 1.

Постройте графики следующих функций на интервале -360° до 360° .

$$f(x) = e^{\cos x} + \ln(\cos^2(0.6x) + 1) \cdot \sin x,$$

$$h(x) = -\ln((\cos x + \sin x)^2 + 2.5) + 10.$$

На оси абсцисс значения должны быть в градусах.

Пояснение. Для вычисления функций косинуса и синуса значения аргумента нужно перевести в радианы.

Задача 2.

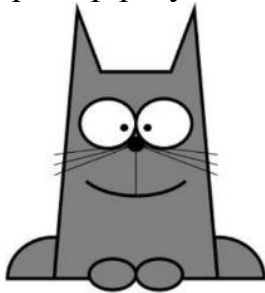
Постройте график функции на интервале $[-10; 10]$:

$$f(x) = \frac{5}{x^2 - 9}$$

Задача 3.

С помощью геометрических примитивов изобразите свое любимое животное.

Пример результата:



Часть 2. NumPy, SciPy

Задача 1.

Напишите программу для решения задачи:

Дан массив, в который занесены расходы человека на проезд по каждому месяцу в течение года. Сравнить, в какой период – зимний или летний – тратится больше денег на проезд. Вывести номера месяцев (начиная с 1), в которые расходы были наибольшими.

Задача 2.

Напишите программу для решения задачи:

Дорога из пункта А в пункт В состоит из нескольких участков, известны их длины. Для автомобиля известна средняя скорость его движения на каждом участке. Автомобиль въехал на дорогу в начале участка k и выехал с нее после проезда по участку p . Найти сколько километров проехал автомобиль по дороге, время движения и среднюю скорость.

Входные данные:

- строка, в которой через пробел перечислены длины всех участков дороги;
- строка, в которой через пробел перечислены средние скорости на участках;
- номер участка, на котором автомобиль въехал на дорогу;
- номер участка, после которого выехал.

Выходные данные:

- длина пути с k по p участок;
- время в пути;
- средняя скорость движения автомобиля по этим участкам.

Тест 1

Длины: 20 8 9 18 5 12 16 16 6 7

Скорости: 44 70 44 66 46 38 38 37 66 67

$k = 4$

$p = 7$

$S = 49$ км, $T = 1.28$ час, $V = 38.34$ км/ч

Задача 3.

Решить систему линейных уравнений матричным способом.

$$\begin{cases} -2x_1 - 8.5x_2 - 3.4x_3 + 3.5x_4 = -1.88, \\ 2.4x_2 + 8.2x_4 = -3.28, \\ 2.5x_1 + 1.6x_2 + 2.1x_3 + 3x_4 = -0.5, \\ 0.3x_1 - 0.4x_2 - 4.8x_3 + 4.6x_4 = -2.83. \end{cases}$$

Для этого нужно:

- сформировать матрицу A , состоящую из коэффициентов левых частей уравнений;
- сформировать вектор столбец B , каждый элемент которого – соответствующие значения правых частей уравнений;
- вычислить вектор-решение системы уравнений по формуле:
 $X = A^{-1} \cdot B$;
- вывести полученный вектор, округлив значения до одного знака после запятой.

Задача 4.

Напишите программу для вычисления определенного интеграла, двойного интеграла (интегралы выбрать на свое усмотрение).

Часть 3. Генерация синтетических данных

Генерация данных

Сгенерировать данные, описывающие итоги вступительной кампании за последние 5 лет, которые будут содержать следующую информацию: ФИО студента, год поступления, форма обучения, балл за ЦЭ/ЦТ, средний балл аттестата, общий балл при поступлении, специальность, на которую поступил студент, адрес регистрации, номер мобильного телефона.

Применяя различные библиотеки визуализации данных, отобразите динамику среднего балла за ЦТ/ЦЭ по каждому предмету, динамику среднего балла аттестата, динамику проходного балла, количество поступивших студентов по каждой специальности, статистику по формам обучения.

Часть 4. Анализ данных по продажам

Анализ данных продаж авиабилетов – крайне важная задача для любой авиакомпании.

Собранный из внутренних источников компании S7 датасет представляет из себя обработанный набор записей покупок авиабилетов по датам и их характеристики (способы оплаты, пассажиры, сегменты перелета, суммы и т. д.) за некоторый период времени.

Используя библиотеки Python, ответьте на следующие вопросы, приводя подтверждение в виде графиков, схем, дашбордов, таблиц и пр.:

- общие описательные статистики показателей;
- можно ли выделить какие-то определенные аэропорты;
- есть ли в данных сезонность; если да, то какие показатели от нее зависят;
- можно ли сделать какие-то выводы по статусу или типам пассажиров;
- есть ли какие-то особенности и нюансы при выборе способа оплаты;
- предсказание объемов продаж билетов, количества перелетов.

Часть 5. Анализ продаж

Используйте предложенный датасет, или же выберите самостоятельно, или же сгенерируйте свой и проведите анализ данных. Исходя из предложенного датасета, нужно проанализировать динамику продаж по каждому виду товара и по каждой точке реализации (количество, объем продаж, себестоимость, средняя цена, средние продажи на точку, рост/спад и т.д. за период), а также динамику общего товарооборота. Составьте прогноз продаж по каждому виду товара. Отрадите все показатели с помощью средств визуализации, сформировав полноценный отчет.

Указания к выполнению работы:

Создать для лабораторной работы ветку lab_4 от ветки main, после от ветки lab_4 создать 5 веток, согласно частям лабораторной работы. У каждой части должно быть свое виртуальное окружение с РАЗНЫМИ версиями библиотек и Python, обязательно создать файл requirements.txt для каждой части. Для одной из частей (на свой вкус) сначала создать файл requirements.txt со списком необходимых библиотек, а затем установить их, согласно указанному списку.