

**Содержание**

[Введение 3](#_Toc45721587)

[Основная часть 4](#_Toc45721588)

[1 выбор платформы и данных 4](#_Toc45721589)

[1.1 Выбор платформы для разработки и необходимых библиотек 4](#_Toc45721590)

[1.1.1 Обоснование выбора платформы интерпретатора языка Python 5](#_Toc45721591)

[1.1.2 Выбор библиотек для анализа данных 5](#_Toc45721592)

[1.2 Получение данных 6](#_Toc45721593)

[1.2.1 Предварительный анализ данных 6](#_Toc45721594)

[1.2.2 Выбор формата хранения данных 7](#_Toc45721595)

[1.2.3 Очистка данных от цифрового мусора 7](#_Toc45721596)

[1.3 Выводы по разделу 8](#_Toc45721600)

[2 Сортировка данных 10](#_Toc45721601)

[2.1 Получить Dataset (данные) 10](#_Toc45721602)

[2.2 Рассчитать стандартные числовые характеристики для набора данных Series 13](#_Toc45721603)………………………………………………………11

[2.3 Визуализировать данные с помощью стандартных библиотек по заданным критериям ………………………………………………………12](#_Toc45721604)

[2.4 Сформировать Dataframe из данных Series и добавить к этим данным следующие столбцы 14](#_Toc45721605)

[2.5 Визуализировать данные, полученные в результате промежуточного анализа (вычислений) 18](#_Toc45721606)

[2.6 Выводы по разделу 20](#_Toc45721606)

[Заключение 21](#_Toc45721613)

[Список использованной литературы 21](#_Toc45721614)

**Введение**

Учебная практика (ознакомительная практика) относится к разделу Б2.О.01 блока «Практики» базового учебного плана основной профессиональной образовательной программы по направлению подготовки 09.03.03 Прикладная информатика и является видом учебной деятельности, направленной на ознакомление, формирование, закрепление, развитие практических навыков и компетенции в процессе выполнения определенных видов работ, связанных с будущей профессиональной деятельностью. Учебная практика проходила в Московском университете им. С.Ю. Витте (далее - Университете) на кафедре информационных систем

**Цель практики**

Приобретение первичных профессиональных навыков, практического опыта, закрепление, систематизация и расширение теоретических знаний по дисциплинам учебного плана при прохождении учебной практики в Университете.

**Задачи практики**

* Закрепление приобретенных теоретических знаний.
* Приобретение навыков научного поиска и практической работы с информационными источниками данных.
* Овладение практическими методиками формирования цели и задач для работы над групповым проектом, управления познавательной деятельностью.
* Приобретение практических навыков участия в командной работе, в групповых проектах, распределения ролей в условиях командного взаимодействия
* Приобретение практических навыков использования иностранного языка при анализе зарубежных информационных Интернет источников.
* Приобретение навыков использования современных информационных технологий и программных средств для решения задач анализа данных.
* Приобретение навыков инсталлирования программного обеспечения.
* Формирование практических навыков по основам алгоритмизации вычислительных процессов и программированию решения задач обработки данных их отладке и тестирования на выбранной языковой платформе.
* Получение навыков экспериментального исследования больших данных и практического решения информационных задач по обработке данных.
* Получения навыков проведения инженерных расчетов.
* Получение навыков самостоятельной работы.
* Получение навыков подготовки отчетной документации

Источниками информации явились открытые информационные базы, информационно-справочные документы, интернет-ресурсы, учебно-методические материалы, размещенные в Электронном Университете.

**ОСНОВНАЯ ЧАСТЬ**

# выбор платформы и данных

## Выбор платформы для разработки и необходимых библиотек

Python (в русском языке распространено название пито́н) — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты). Python имеет множество полезных библиотек, такие как Matplotlib, pandas, NumPy

### Обоснование выбора платформы интерпретатора языка Python

Python — один из самых популярных языков программирования в мире, в свежем рейтинге [TIOBE](http://www.tiobe.com/) он занимает 5 место. Так же это язык программирования общего назначения, нацеленный в первую очередь на повышение продуктивности самого программиста, нежели кода, который он пишет. Говоря простым человеческим языком, на Python можно написать практически что угодно (веб-/настольные приложения, игры, скрипты по автоматизации, комплексные системы расчёта, системы управления жизнеобеспечением и многое многое другое) без ощутимых проблем.

### Выбор библиотек для анализа данных

В нашей практической работе мы решили выбрать 3 библиотеки: matplotlib, numpy и pandas

Matplotlib — один из самых популярных пакетов Python, используемых для визуализации данных. Это кроссплатформенная библиотека для создания 2D графиков из данных в массивах. Matplotlib написан на Python и использует NumPy, числовое математическое расширение Python. Он предоставляет объектно-ориентированный API, который помогает встраивать графики в приложения, используя наборы инструментов Python GUI, такие как PyQt, WxPythonotTkinter. Он также может использоваться в оболочках Python и IPython, ноутбуках Jupyter и серверах веб-приложений.

Matplotlib имеет процедурный интерфейс под названием Pylab, который похож на MATLAB, проприетарный язык программирования, разработанный MathWorks. Matplotlib вместе с NumPy можно рассматривать как эквивалент MATLAB с открытым исходным кодом.

NumPy - один из самых фундаментальных пакетов в Python - универсальный пакет для обработки массивов. Он предоставляет высокопроизводительные объекты многомерных массивов и инструменты для работы с массивами. NumPy — это эффективный контейнер универсальных многомерных данных.

Что можно делать с помощью NumPy?

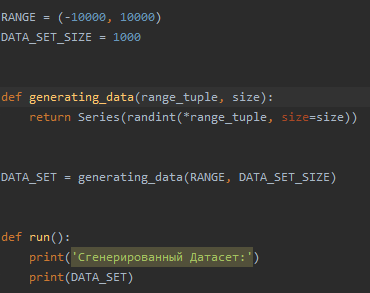
1. Основные операции с массивами: добавление, умножение, срез, выравнивание, изменение формы, индексирование массивов;
2. Расширенные операции с массивами: стековые массивы, разбиение на секции, широковещательные массивы;
3. Работа с DateTime или линейной алгеброй;
4. Основные нарезки и расширенное индексирование в NumPy Python.

Pandas — это библиотека Python с открытым исходным кодом, предоставляющая высокопроизводительный инструмент для обработки и анализа данных с использованием его мощных структур данных. Название Pandas происходит от слова Panel Data — эконометрика из многомерных данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами.

**1.2 Получение данных**

### 1.2.1 Предварительный анализ данных

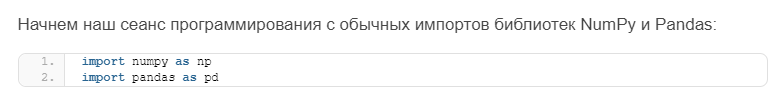
Анализ данных – преобразование данных в выводы, на основе которых будут приниматься решение и строиться действия с помощью людей, процессов и технологий. В своей проектной работе я решил получить данные для анализа следующим образом:



### 1.2.2 Выбор формата хранения данных

**Pandas** — более новый пакет, надстройка над библиотекой NumPy, обеспечивающий эффективную реализацию класса DataFrame.

**Объекты DataFrame** — многомерные массивы с метками для строк и столбцов, а также зачастую с неоднородным типом данных и/или пропущенными данными.



Помимо удобного интерфейса для **хранения маркированных данных**, библиотека **Pandas** реализует **множество операций для работы с данными** хорошо знакомых пользователям фреймворков баз данных и электронных таблиц.

**Объект Series** библиотеки Pandas — одномерный массив индексированных данных. Его можно создать из списка или массива следующим образом:



Как мы видели из предыдущего результата, объект Series служит адаптером как для последовательности значений, так и последовательности индексов, к которым можно получить доступ посредством атрибутов values и index.

### 1.2.3 Очистка данных от цифрового мусора

Очистка данных – это процесс обнаружения и поправки или удаления файлов, которые были повреждены или же неточных записей из набора этих записей, таблицы или же базы данных. Процесс подключает в себя выявление неполных, неверных, неточных или же несущественных данных, а также вслед за тем замену, удаление или изменение «загрязненных» данных.

Если сравнивать этот язык с другими известными нам языками, Python удалив какой-либо объект не сразу освобождает память обратно. Взамен сего, он пользуется вспомогательным менеджером памяти, предназначенным для малехоньких объектов (размер их составляет не больше чем 512 байт). Для работы с этими объектами он выделяет гигантские блоки памяти, в которых в последующем станет сберегается большое количество не больших объектов.

Как только один маленький объект удаляется — память из-под него не перебегает к операционной системе, Python оставляет её для свежих объектов с этим же объемом. В случае если в одном из выделенных блоков памяти не осталось объектов, то Python имеет возможность вызволить его операционной системе. Как правило, освобождение блоков доводится, когда скрипт делает большое количество временных объектов.

Метод подсчета ссылок — это одна из самых несложных техник для сборки ненужных данных. Объекты удаляются, как только они не имеют каких-либо ссылок. 

## 1.3 Выводы по разделу

Исходя из вышеуказанного, мы сделали вывод по выбору платформы и удобных библиотек для решения поставленной нам задачи (сортировка данных). Провели сортировку данных и очистку от цифрового мусора.

Таблица 1: Выводы по разделу 1

|  |  |  |
| --- | --- | --- |
| Выводы | Сформированные компетенции | |
| Код | Содержание компетенции |
| Осуществил поиск и системный анализ подходящих инструментов языка Python. | УК-1 | Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач. |
| Выбрал наиболее подходяшие библиотеки | УК-2 | Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений |
| Организовал получение данных и выполнил свою роль в их генерации | УК-3 | Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде |
| Написал отчёт на русском языке | УК-4 | Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах) |
| Воспринял многообразие и историю технологических решений по анализу данных | УК-5 | Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах |
| Спланировал работу по проекту | УК-6 | Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни |
| Подготовил рабочее место согласно требованиям | УК-7 | Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности |
| Выбрал библиотеки, реализующие наиболее оптимальные алгоритмы, тем самым минимизировав углеродный след от вычислений. | УК-8 | Способен создавать и поддерживать в повседневной жизни и в профессиональной деятельности безопасные условия жизнедеятельности для сохранения природной среды, обеспечения устойчивого развития общества, в том числе при угрозе и возникновении чрезвычайных ситуаций и военных конфликтов |
| Получил данные наиболее экономичным способом | УК-9 | Способен принимать обоснованные экономические решения в различных областях жизнедеятельности |
| Реализовал отчёт без плагиата и способов обхода проверок | УК-10 | Способен формировать нетерпимое отношение к коррупционному поведению |
| Осуществил сбор инженерных измерений для формирования модели | ОПК-1 | Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности; |
| Продемонстрировал понимание принципов работы выбранных библиотек | ОПК-2 | Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности; |
| Познакомился с массивом литературы и документацией по выбранным библиотекам, а также с лучшими практиками их применения, в том числе в плане безопасности | ОПК-3 | Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; |
| Документировал работу в виде данного отчёта | ОПК-4 | Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью; |
| Инсталлировал выбранные библиотеки | ОПК-5 | Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем; |
| Выполнил организационную подготовку к выполнению процесса анализа данных | ОПК-6 | Способен анализировать и разрабатывать организационно-технические и экономические процессы с применением методов системного анализа и математического моделирования; |
| Разработал практический алгоритм генерации данных | ОПК-7 | Способен разрабатывать алгоритмы и программы, пригодные для практического применения; |
| Реализовал подготовительные стадии жизненного цикла проекта о работе с данными | ОПК-8 | Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла; |

# 

# СОРТИРОВКА данных

## 2.1 Получить Dataset (данные)

**Pandas** — это высокоуровневая библиотека для анализа данных. Почему я её называю высокоуровневой, потому что построена она поверх более низкоуровневой библиотеки NumPy (написана на Си), что является большим плюсом в производительности. В экосистеме Python, pandas является наиболее продвинутой и быстроразвивающейся библиотекой для обработки и анализа данных.

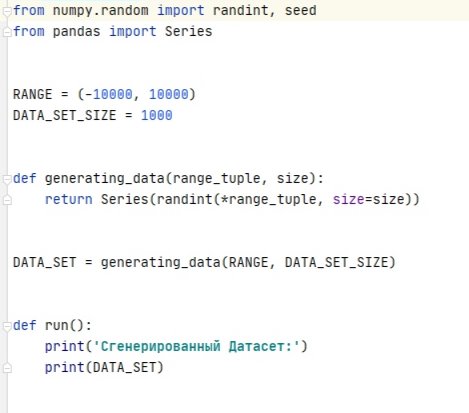
## DataFrame и Series

Чтобы эффективно работать с pandas, необходимо освоить самые главные структуры данных библиотеки: DataFrame и Series. Без понимания что они из себя представляют, невозможно в дальнейшем проводить качественный анализ.

### Series

Структура/объект Series представляет из себя объект, похожий на одномерный массив (Python список, например), но отличительной его чертой является наличие ассоциированных меток, т.н. индексов, вдоль каждого элемента из списка. Такая особенность превращает его в ассоциативный массив или словарь в Python.

Мы получили Dataset данные таким образом:



## 2.2 Рассчитать стандартные числовые характеристики для набора данных Series

Прежде чем приступить к выполнению задания №2, мы сначала импортируем DataSet , который был создан в задании №1 (см. строку №1).



После импортирования DataSet приступаем к заданию.  
1. Минимальное значение:

Находим минимальное значение созданного DataSet при помощи команды DATA\_SET.min( ) (см. строку №4).



2. Подсчет количества повторяющихся элементов:

Находим количество повторяющихся элементов , которое будем искать по формуле CD = S - nU , где CD - количество повторяющихся элементов DataSet , S - размер всего DataSet (все значения) , а nU - количество уникальных элементов DataSet (см. строку №6).  
  
3. Максимальное значение:



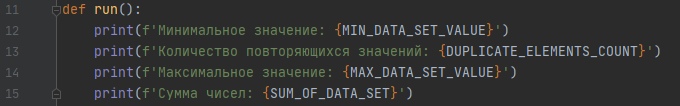
Находим максимальное значение созданного DataSet при помощи команды DATA\_SET.max() (см. строку №7).  
  
Аналогично заданию по поиску минимального значения (см. пункт №1 и строку №4).  
4. Сумма значений:



Находим сумму всех значений созданного DataSet (кол-во значений - 1000) при помощи команды DATA\_SET.sum() (см. строку №8).



Далее, при помощи команды def run() и команды print() , мы выводим все наши данные на печать в порядке , заданном в задании №2 (см. строки 11 - 15)

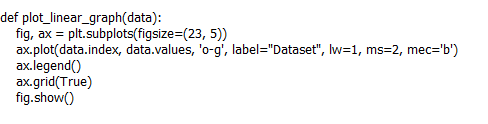


## 2.3 Визуализировать данные с помощью стандартных библиотек по заданным критериям

Линейный график отображает динамику по одному или нескольким показателям. Его удобно использовать, чтобы сравнить, как меняются со временем разные наборы данных. Например, посмотреть статистику посещений трех посадочных страниц.

При работе с линейными диаграммами упор делается на продолжение или последовательность значений (тренд), но все еще есть некоторая поддержка для сравнения отдельных значений с использованием маркеров данных.

Для построения линейного графика используется функция plot()

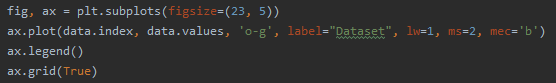


Гистограмма — это способ представления статистических данных в графическом виде – в виде столбчатой диаграммы. Она отображает распределение отдельных измерений параметров изделия или процесса.

Matplotlib — это популярная библиотека для визуализации данных, написанная на языке Python. Хоть пользоваться ей очень просто, настройка данных, параметров, графиков и отрисовки для каждого нового проекта.

Изображение в matplotlib создается путём последовательного вызова команд/функций этого модуля. Графические объекты (точки, линии, фигуры и т.д.) последовательно накладываются один на другой, если они занимают общие области на рисунке. Объектом самого высокого уровня при работе с matplotlib является рисунок (Figure). На нем располагаются одна или несколько областей рисования (Axes) и элементы оформления рисунка (заголовки, легенда и т.д.). Каждый объект Axes содержит две (или три) координатных оси Axis. Но основное назначение Axes состоит в том, что на него наносится графика: кривые, диаграммы, легенды и так далее.

Создать рисунок figure позволяет инструкция: plt.figure()



Чтобы текущий рисунок отразился на экране, нужно воспользоваться командой plt.show(). После ввода команды plt.show(), откроется окно Matplotlib с графиком.



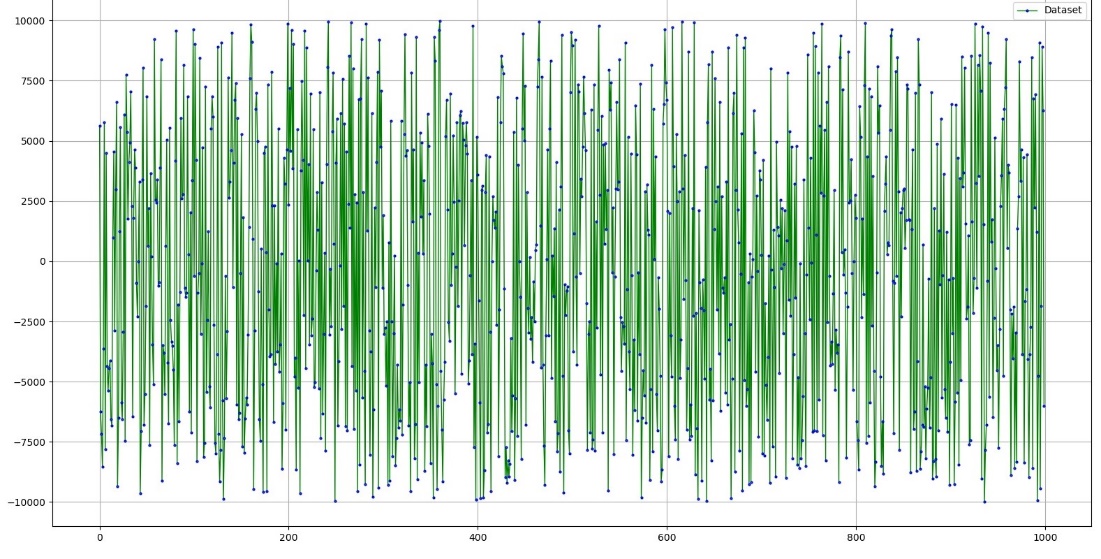
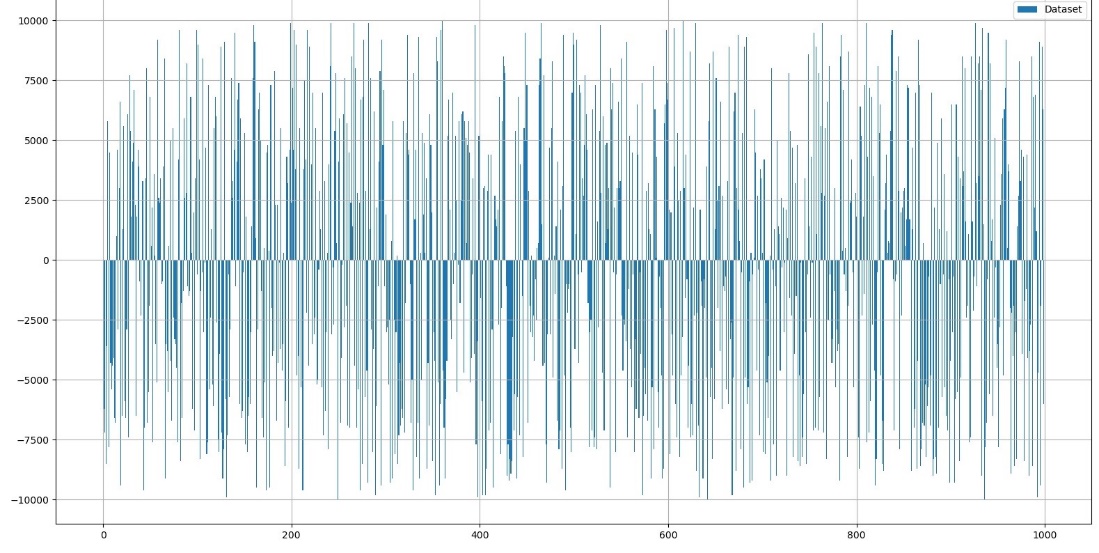
Matplotlib предоставляет функциональные возможности для визуализации гистограмм Python

Как было определено ранее, график гистограммы использует края бина на оси х и соответствующие частоты на оси у.

Оставаясь в научном стеке Python, Pandas + Series.histogram () +  pyplot.hist () , чтобы нарисовать гистограмму Matplotlib



Получились такие гистограммы:



## 2.4 Сформировать Dataframe из данных Series и добавить к этим данным следующие столбцы

Импортируем библиотеку DataFrame , а также Dataset , полученную из задания №1:

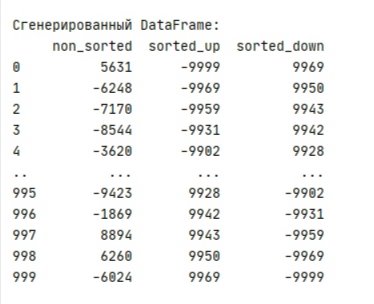
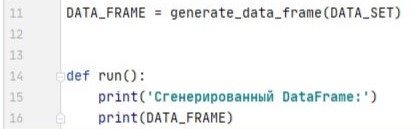


Далее мы приступаем к формированию DataFrame из данных Series и добавляем к ним столбцы.

Генерируем DataFrame , и выполняем сортировку следующим образом:



После мы даем имя DataFrame новому сгенерированному DataFrame из DataSet, а после выводим на экран получившиеся числа:

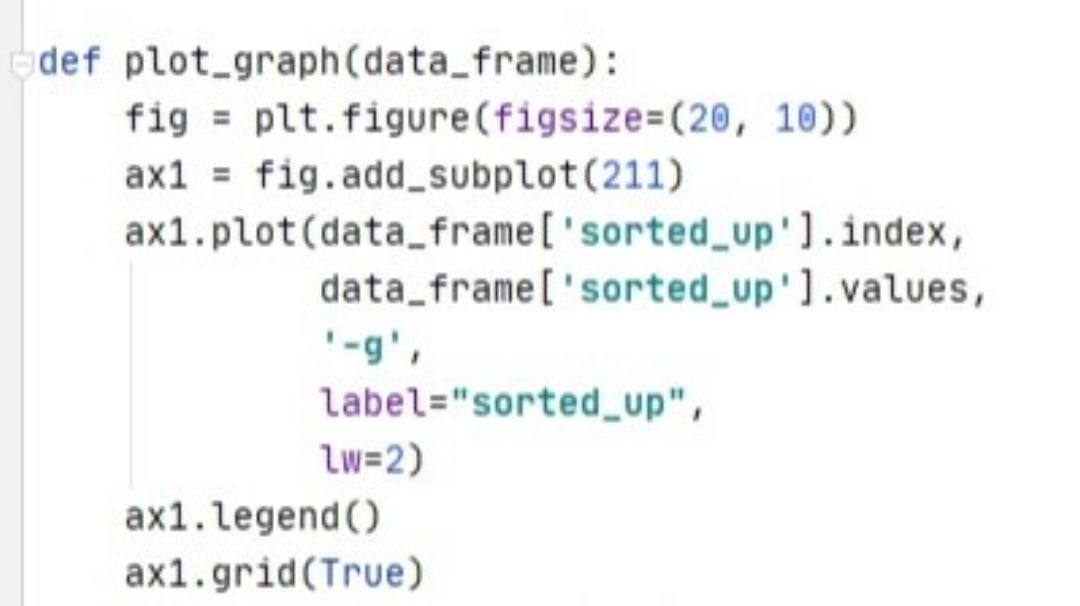


## 2.5 Визуализация данных полученные в результате промежуточного анализа

Для визуализации данных, полученных во время анализа, потребовалось прибегнуть к использованию сторонней библиотеки “Matplotlib”. Сама библиотека содержит в себе огромное количество способов для построения фигуры. Размер самой библиотеки насчитывает около 70000 строк кода.

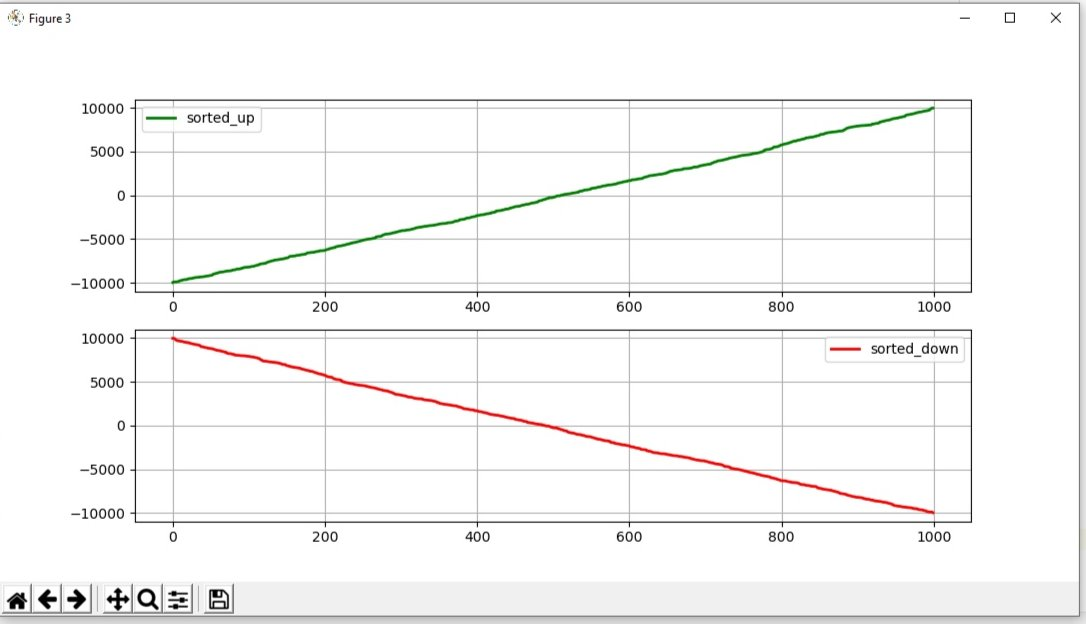
Возвращаясь к коду, мы подключаем инструмент pyplot, импортируя его из библиотеки “Matplotlib”. После импортируем “Data\_Frame” из “Task\_4”, чтобы получить данные из промежуточного анализа. Чтобы построить сам график мы использовали команду “plot\_graph()”, внутрь которого мы поместили объект **Figure** - важнейший внешний контейнер для графики “Matplotlib”. Ведь именно **Figure** содержит в себе множество объектов **Axes**, которые являются непосредственно графиками. С помощью команды “figsize=(20, 10)” мы задаем конкретную размерность окна(Figure). В строке “ax = fig.add\_subplot(211)” мы добавили к Figure область Axes. Функция “ax = fig.add\_subplot(211)” является более гибкой в использовании, нежели функция “fig.add\_axes”, которая считается правильной для библиотеки Matplotlib.

В этих строчках мы добавляем цвет линии, ее ширину, а также прописываем команды по добавлению легенды и сетки на график. **Legend — это** вспомогательный инструмент, который позволяет определить, что принадлежит определенному цвету линии. Легенда сделала наш график более информативным. Это все косметические инструменты, позволяющие нам сделать график более привлекательным.



Тоже самое мы проделываем с вторым графиком на сортировку по убыванию.

В заключении мы получаем два линейных графика, отсортированных по значению возрастания и убывания.



## 2.6 Выводы по разделу

Во время практической работы мы научились сортировке данных, а как раз: генерированию числовых данных, расчёту нормальных числовых данных (минимальное значение, количество повторяющихся значений, сумму чисел, максимальное значение), визуализации данных с помощью стандартных библиотек по данным аспектам, визуализации данных, приобретенных в итоге промежуточного анализа (вычислений).

Таблица 2: Выводы по разделу 2

|  |  |  |
| --- | --- | --- |
| Выводы | Сформированные компетенции | |
| Код | Содержание компетенции |
| Системно проанализировал данные | УК-1 | Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач. |
| Определили способы анализа данных и выполнил соответствующие задачи | УК-2 | Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений |
| Применил методы статистического анализа для обработки данных | ОПК-1 | Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности; |
| Продемонстрировал понимание принципов статистического анализа данных | ОПК-2 | Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности; |
| Проанализировал отражённые в данных организационно-технические и экономические процессы | ОПК-6 | Способен анализировать и разрабатывать организационно-технические и экономические процессы с применением методов системного анализа и математического моделирования; |
| Разработал несколько практических алгоритмов статистического анализа данных | ОПК-7 | Способен разрабатывать алгоритмы и программы, пригодные для практического применения; |
| Выполнил работы на стадиях разработки и тестирования жизненного цикла программы | ОПК-8 | Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла; |

**ЗАКЛЮЧЕНИЕ**

С помощью приобретённых первичных профессиональных навыков в учебном процессе мы научились работать в команде над групповым проектом, распределяя обязанности в условиях командного взаимодействия. Приобрели навыки использования современных информационных технологий и навыков использования иностранного языка. Так же овладели навыками инсталлирования программного обеспечения. Мы смогли улучшить свои практические навыки по основам алгоритмизации вычислительных процессов и программирования.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Книга “Pandas. Работа с данными”
2. [https://www.tutorialspoint.com/python\_pandas/python\_p..](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fwww.tutorialspoint.com%2Fpython_pandas%2Fpython_pandas_dataframe.htm)
3. [https://python-scripts.com/matplotlib](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fpython-scripts.com%2Fmatplotlib) (книга по Matplotlib)
4. [https://devpractice.ru/matplotlib-book/](https://vk.com/away.php?utf=1&to=https%3A%2F%2Fdevpractice.ru%2Fmatplotlib-book%2F). (Matplotlib)
5. [https://devpractice.ru/kniga-pandas/ (Книга](https://devpractice.ru/kniga-pandas/%20(Книга) Pandas).
6. Архитектура приложений с открытым исходным кодом", том 2 Автор: John Hunter, Michael Droettboom, перевод: А. Панин