 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА**

з дисципліни «Обробка та аналіз біомедичних даних»

**Виконав:**

студент гр. БС-82

Французьонок І.С.

**Перевірив:**

доц. Носовець О.К

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

зміст

Вступ………………………………………………………………………………………..4

1. Основи роботи з Jupyter Notebook…………………………………………...................5

1.1. Теоретичні відомості……………………………………………………...…....5

1.2. Практична реалізація……………………………………………………......….6

1.3. Висновки до розділу……………………………………………………….......11

2. Візуалізація даних за допомогою бібліотеки Matplotlib…………………………......12

2.1. Теоретичні відомості…………………………………………………………..12

2.2. Практична реалізація………………………………………………………......13

2.3. Висновки до розділу…………………………………………………………...23

3. Візуалізація даних за допомогою бібліотеки Seaborn………………………………..24

3.1. Теоретичні відомості…………………………………………………………..24

3.2. Практична реалізація………………………………………………………......25

3.3. Висновки до розділу…………………………………………………………...31

4. Перевірка даних на нормальність за допомогою критерію Шапіро-Уілка…………32

4.1. Теоретичні відомості…………………………………………………………..32

4.2. Практична реалізація………………………………………………………......33

4.3. Висновки до розділу…………………………………………………………...38

5. Критерій Пірсона для порівняння декількох груп за розподіленням ознаки…….…39

5.1. Теоретичні відомості…………………………………………………………..39

5.2. Практична реалізація………………………………………………………......40

5.3. Висновки до розділу…………………………………………………………...43

6. Парний t–критерій Стьюдента…………………………………………………………44

6.1. Теоретичні відомості…………………………………………………………..44

6.2. Практична реалізація………………………………………………………......45

6.3. Висновки до розділу…………………………………………………………...48

7. t-критерій Стьюдента для незалежних вибірок……………………………………….49

7.1. Теоретичні відомості…………………………………………………………..49

7.2. Практична реалізація………………………………………………………......50

7.3. Висновки до розділу…………………………………………………………...54

8. Дисперсійний аналіз……………………………………………………………………55

8.1. Теоретичні відомості…………………………………………………………..55

8.2. Практична реалізація………………………………………………………......56

8.3. Висновки до розділу…………………………………………………………...60

**Вступ**

Мета розрахункової роботи: систематизація, засвоєння, поглиблення знань з дисципліни «Обробка та аналіз біомедичних даних», отримання на практиці навички аналізу та обробки даних, приміняючи різні методи.

Аналіз даних - розділ математики, що займається розробкою методів обробки даних незалежно від їх природи. Аналіз даних можна вважати прикладним розділом математичної статистики, проте потрібно наголосити, що аналіз даних охоплює обробку як кількісних, так і якісних даних. Можна виділити такі етапи аналізу даних: отримання даних, обробка, аналіз та інтерпретація результатів обробки.

Обробка даних виконуватиметься за допомогою мови Python y середовищі Jupyter Notebook з використанням бібліотек для візуалізації (matplotlib, seaborn) та роботи з даними (numpy, scipy, тощо).

**1. Основи роботи з Jupyter Notebook**

* 1. Теоретичні відомості

Jupyter Notebook – це командна оболонка для інтерактивних обчислень. Цей інструмент може використовуватися не лише на Python, але й на інших мовах програмування: Julia, R, Haskell і Ruby. Він часто використовується для роботи з даними, статистичним моделюванням і машинним навчанням. Робота в цій оболонці дуже зручна, оскільки можна дивитись результати роботи окремих шматків коду.

**pandas — програмна бібліотека, написана для мови програмування Python для маніпулювання даними та їхнього аналізу. Надає структури даних та операції для маніпулювання чисельними таблицями та часовими рядами.**

**Щоб ефективно працювати з pandas, необхідно освоїти структуру даних бібліотеки – DataFrame(таблична структура даних).**

**Задля коректного імпортування даних з робочої директорії використовувують os.**

**os – модуль, що надає безліч функцій для роботи з операційною системою, причому їх поведінка, як правило, не залежить від ОС.**

**Модуль рandas містить безліч функцій, які допомагають при роботі з даними, наприклад:**

* **head – вивід даних з початку;**
* **tail – вивід даних з кінця;**
* **describe – для опису різних типів даних;**
* **shape – відображення кількості стовпців та рядків (спостережень)**
  1. практична реалізація



Рис. 1.1. Завантаження даних в ручну



Рис. 1.2. Завантаження даних з CSV файлу

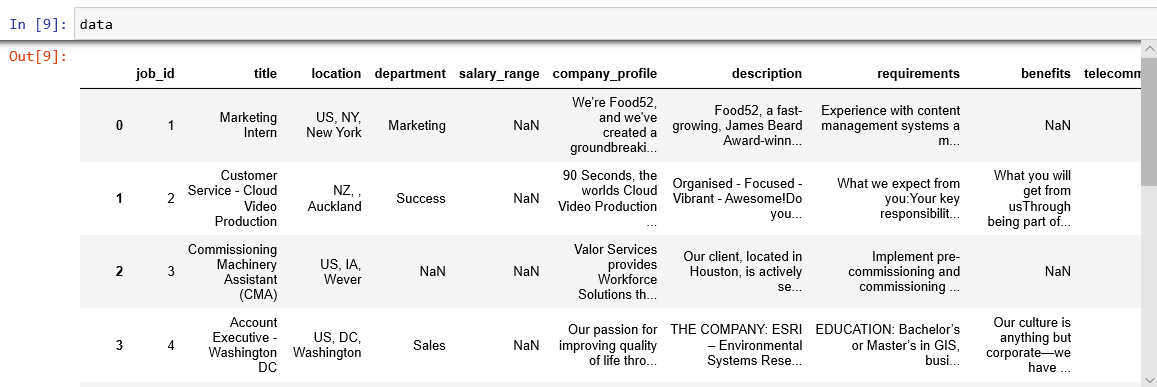


Рис. 1.3. Відображення завантажених даних

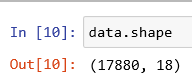


Рис. 1.4. Відображення кількості стовпців та рядків

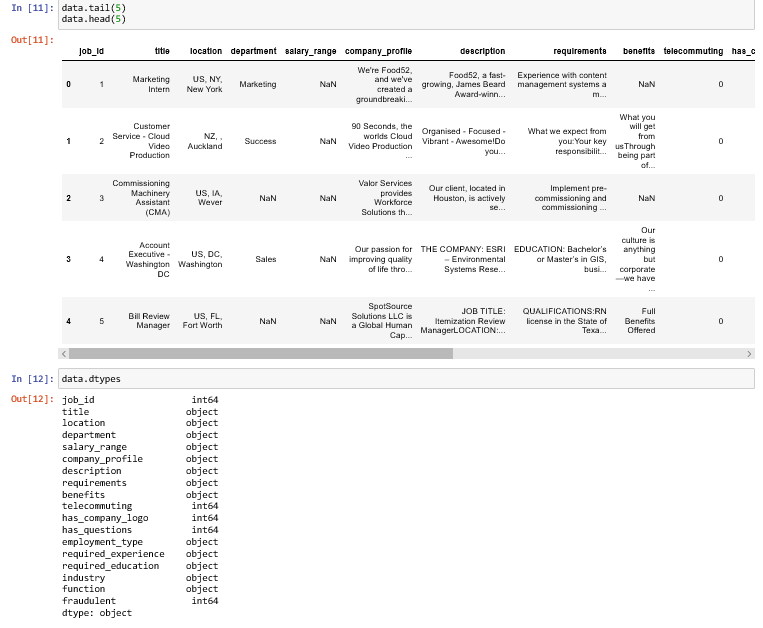


Рис. 1.5. Відображення п’яти перших і п’яти останніх рядків

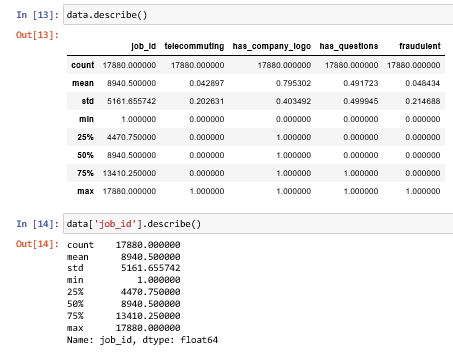


Рис. 1.6. Використання describe

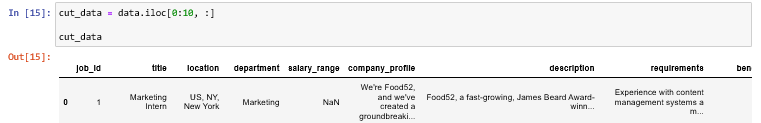


Рис. 1.7. Створення нового масиву який містить лише перші 10 рядків

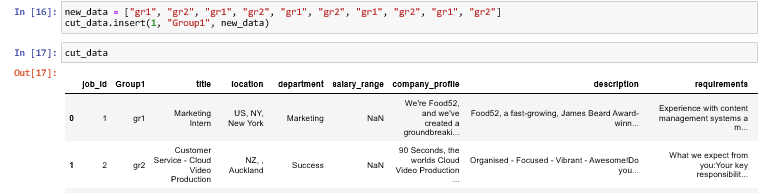


Рис. 1.8. Додається новий стовпчик

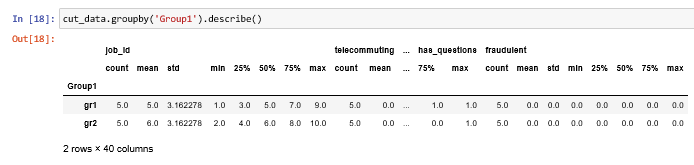


Рис. 1.9. Розділення на групи



Рис. 1.10. Збереження файлів

* 1. ВИСНОВКИ ДО РОЗДІЛУ

У ході виконанні даної роботи було ознайомлення з Jupyter Notebook та бібліотекою pandas. Було згенеровано дані в ручну та загружено з файлу. Використовувались функції head, tail, describe.

Jupyter Notebook є зручною для обробки та аналізу даних, має простий і зручний інтерфейс, дозволяє запусками код частинами і виводить результат виконання, це значно спрощує пошук помилок.

1. **Візуалізація даних за допомогою бібліотеки Matplotlib**
   1. Теоретичні відомості

**Matplotlib - це популярна бібліотека для візуалізації даних, написана на мові Python.** Matplotlib є гнучким, легко конфігурованим пакетом, який разом з NumPy, SciPy і Python надає можливості, подібні до MATLAB.

Дана бібілотека містить безліч функцій:

* *plot* – побудова графіку;
* *subplot* – побудова декількох графіків;
* *tight\_layout* – задля того щоб декілька графіків не накладалися один на одного.

У випадку, якщо необхідно побудувати графік певної функції, однієї бібліотеки matplotlib буде недостатньо, треба підключити це одну – numpy.

**Numpy** — розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами.

Для побудови графіків matplotlib дозволяє використовувати один з двох підходів:

* функціональний метод – звична побудова графіків;
* об’єктно-орієнтований підхід – кожен графік розглядається як об’єкт з власними методами та атрибутами.

Matplotlib має велику кількість функцій, які дозволяють будувати різні графіки:

* scatter – будує точкову діаграму;
* hist – будує гістограму;
* boxplot – будує box-plot діаграму.
  1. Практична реалізація



Рис. 2.1 Імпортування бібліотеки

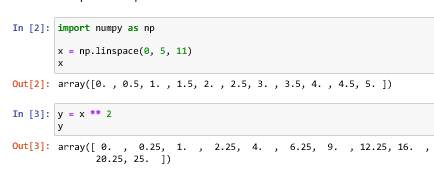
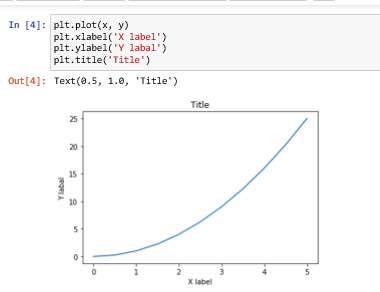
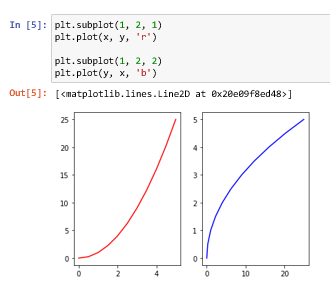
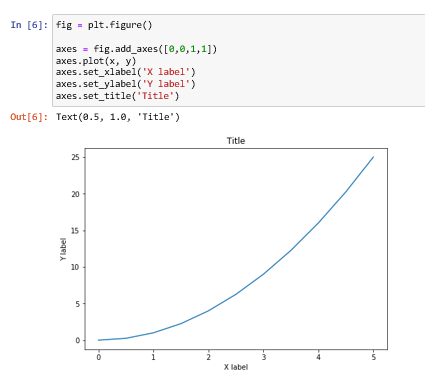
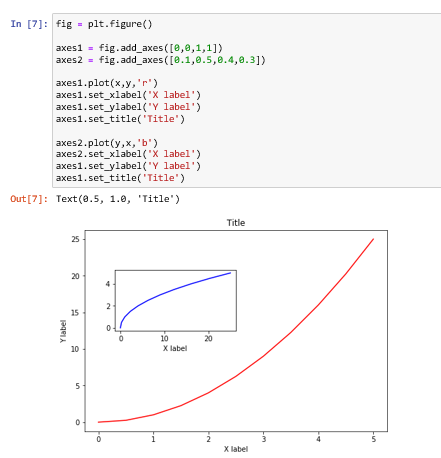


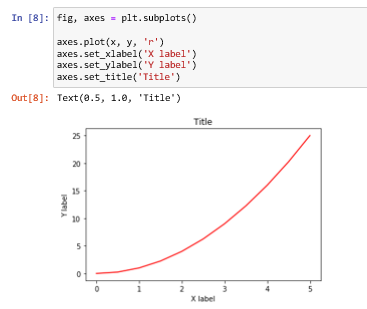
Рис. 2.2. Створення вектора











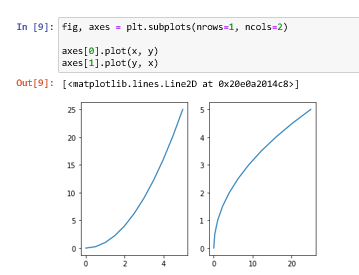


Рис. 2.3 Побудова графіків

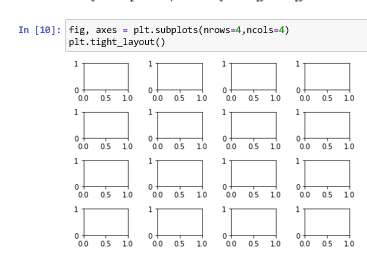


Рис. 2.4. Часткове накладання графіків

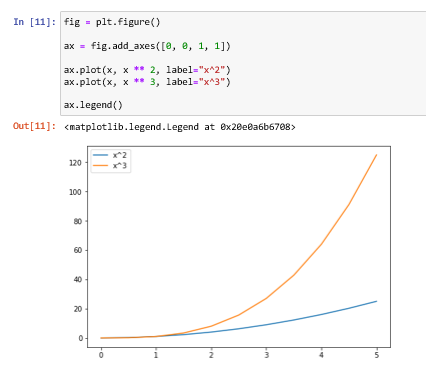


Рис. 2.5. Додавання легенди

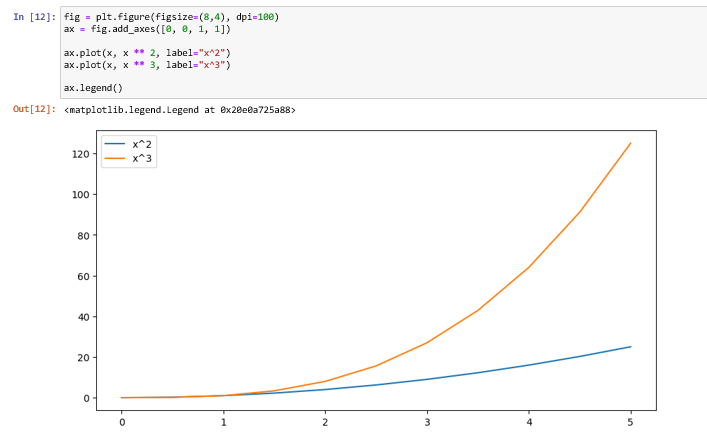


Рис. 2.6. Зміна розміру графіку і кольору

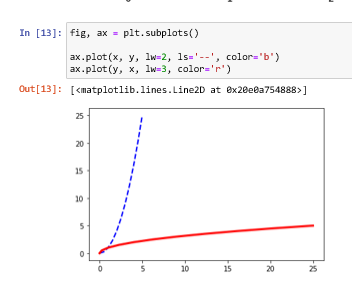


Рис. 2.7. Тип та ширина ліній

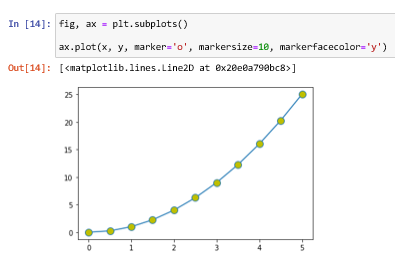


Рис. 2.8. Розмір та колір маркерів

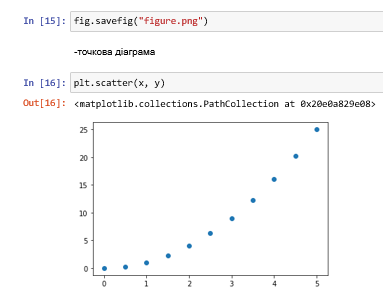


Рис. 2.9. Точкова діаграма

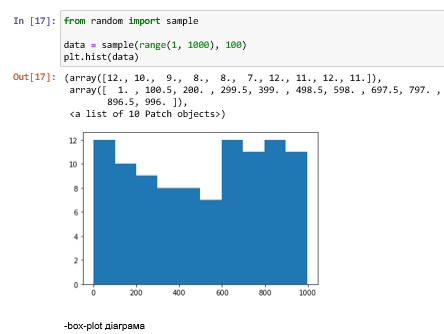


Рис. 2.10. Гістограма

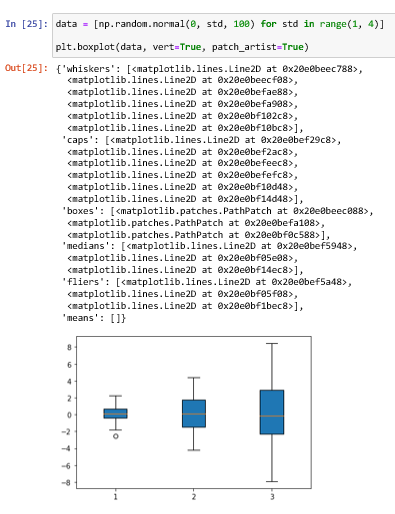


Рис. 2.11. box-plot діаграма

2.3.Висновки до розділу

У ході виконання даної роботи було проведено ознайомлення з бібліотекою matplotlib. Ця бібліотека дозволяє візуалізувати дані, вона має багато вбудованих функцій, що дозволяє легко керувати графіками.

У роботі було ознайомлено з різними методами побудови графіків: побудова декількох графіків поруч, побудова одночасно декількох графіків, додавання легенди до графіку, при створені графіка задавались різні характеристики.

Використання вищезгаданих бібліотек є невід’ємною частиною в аналізі та обробці даних.

1. **Візуалізація даних за допомогою бібліотеки Seaborn**
   1. Теоретичні відомості

**Seaborn** - бібліотека візуалізації даних Python, заснована на matplotlib. Вона так само як і matplotlib взаємодіє з такими бібліотеками як numpy i scipy.

Графіки для аналізу кількісних даних:

* *distplot* – одночасно показує гістограму і графік щільності розподілу;
* *jointplot –* показує спільний розподіл за двома змінним та їх взаємозв’язок;
* *rugplot –* показує те саме, що і графік щільності розподілу, тільки в одновимірної формі. Чим щільніше розташовані лінії, тим вище щільність

Можна також візуалізувати відносні розподіли між парами змінних за допомогою методів:

* *PairGrid –* показує відносини між усіма парами змінних;
* *pairplot –* будує попарні взаємозв’язки між змінними в повному наборі

даних;

* *FacetGrid –* дозволяє візуалізувати спільний розподіл окремих ознак кількох змінних.

***Візуалізація категоріальних даних***

У seaborn вбудовані функції для візуалізації категоріальних даних в наступних форматах:

* *boxplot –* показує медіану (або, якщо потрібно, середнє), нижній і верхній квартили, мінімальне і максимальне значення вибірки і викиди;
* *violinplot –* використовується для вивчення форми розподілу (як і boxplot);
* *stripplot –* представляє собою діаграму розсіювання, де одна за змінних

категоріальна;

* *swarmplot –* рівно те ж саме, що і stripplot з тією лише різницею, що точки не накладаються один на одного;
* *barplot –* дозволяють отримувати сукупні дані по кількісним даним в

залежності від категоріальної змінної;

* *countplot –* те ж саме, що і *barplot*, тільки функція вже явно задана, і вона вважає кількість значень в кожній категорії.
  1. практична реалізація



Рис. 3.1. Імпортування бібліотеки

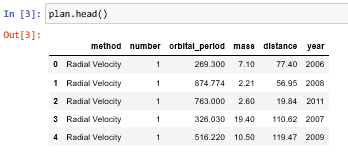
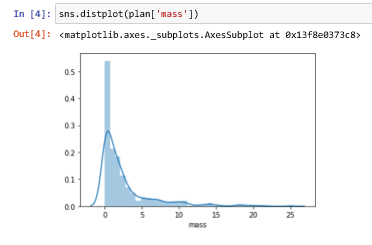
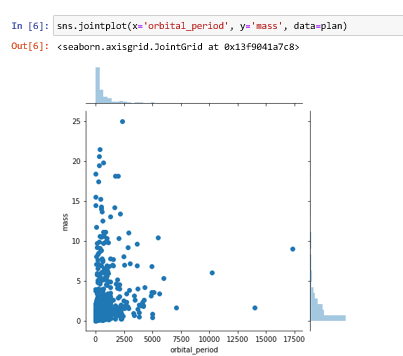


Рис. 3.2. Завантаження даних





\

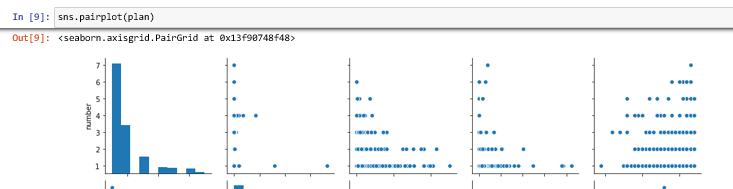
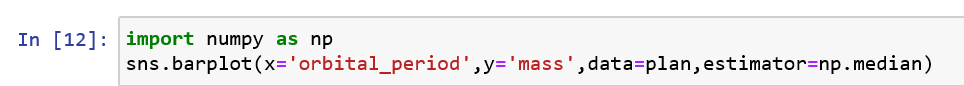
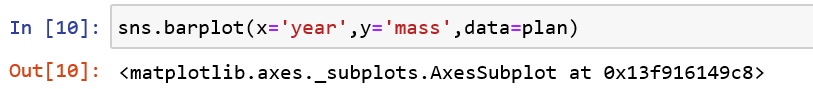


Рис. 3.3. Графіки для аналізу кількісних даних



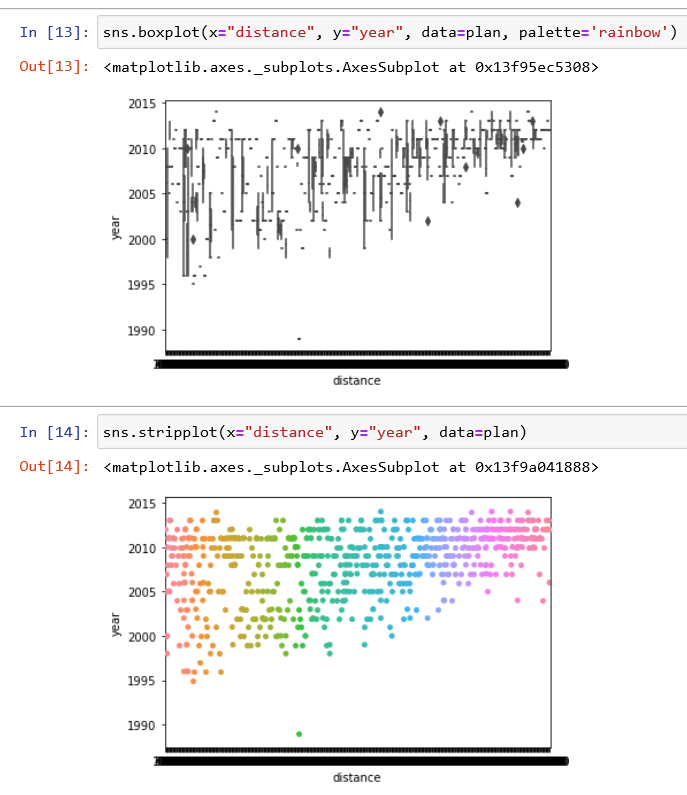


Рис. 3.3. Графіки для аналізу категоріальних даних

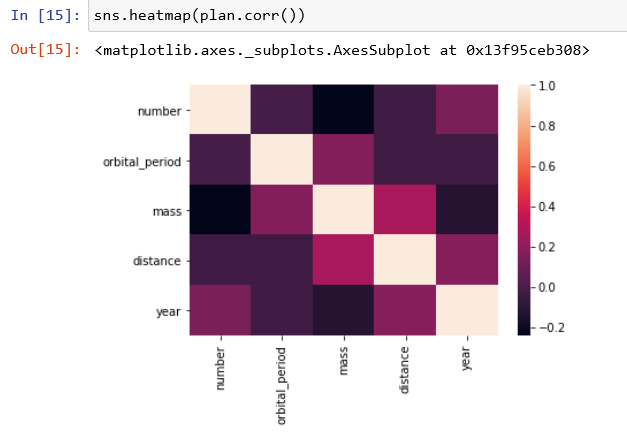


Рис. 3.4. Графік для аналізу взаємозв’язку даних

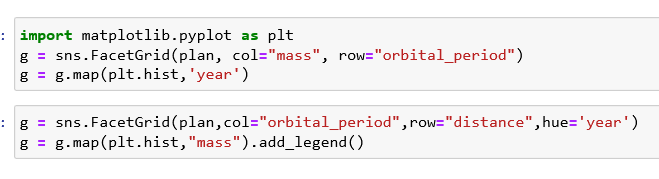


Рис. 3.5. Графік для відображення даних у вигляді сітки

3.3.Висновки до розділу

У ході виконання даної роботи було проведено ознайомлення з бібліотекою seaborn. Ця бібліотека слугує для візуалізації даних, має різноманіття спеціальних графіків, які вона може побудувати. Дану бібліотеку доречно використовувати, коли необхідно побудувати графік розподілу даних, та для визначення залежності між двома або більше змінними.

1. **Перевірка даних на нормальність за допомогою критерію** **Шапіро-Уілка**
   1. Теоретичні відомості

Критерій Шапіро-Уілка використовується для перевірки гіпотези про нормальний розподіл. У ряді дослідів, особливо в медичних, чисельність вибірки мала. Спеціально для перевірки нормальності розподілу малих, чисельністю від трьох до п'ятдесяти елементів, вибірок Шапіро і Уілк розробили критерій . Якщо W статистика значима, то гіпотеза про нормальний розподіл значень змінної відкидається. Критерій Шапіро-Уілка W привілейований критерій нормальності, так як його властивості мають більшу потужність перед широким вибором альтернативних критеріїв нормальності.

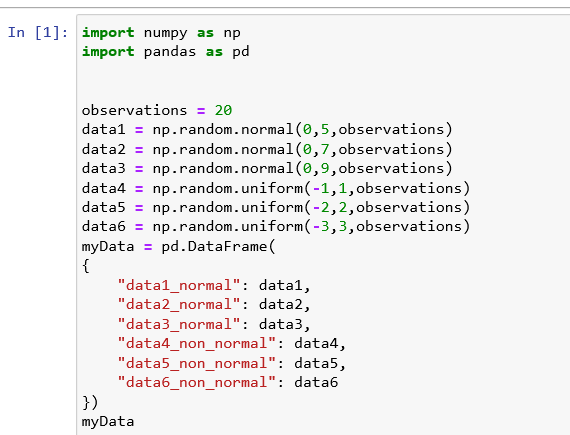
Нехай є вибірка Тоді , де

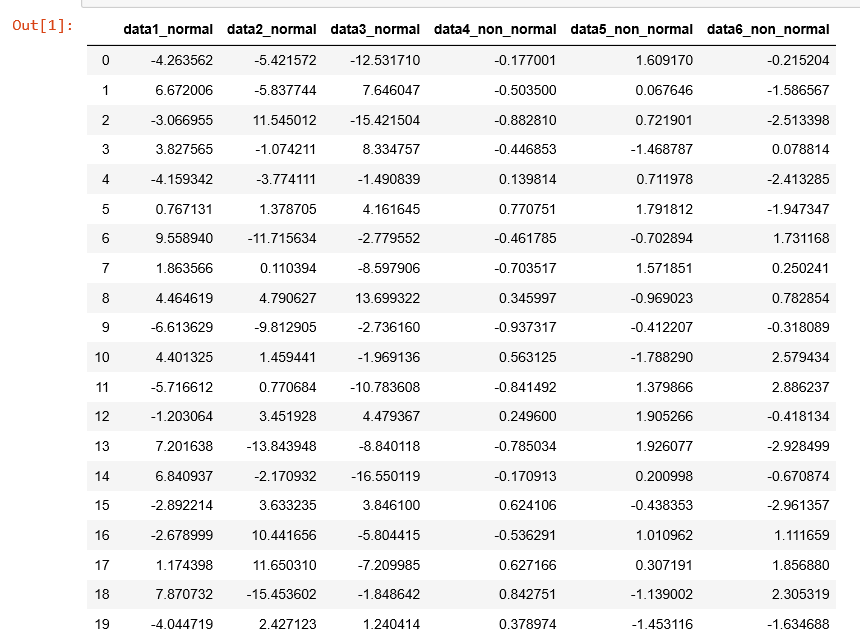
Значення *k* в останній формулі визначається наступним чином:

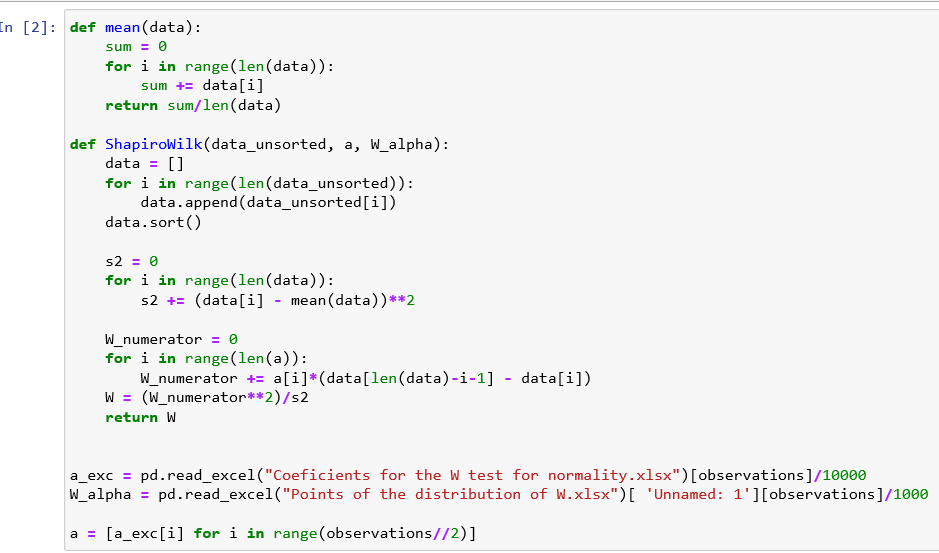
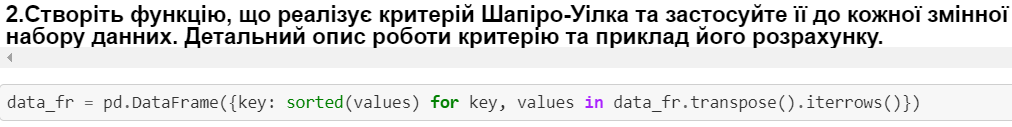
*k=n/2*, якщо n парне;

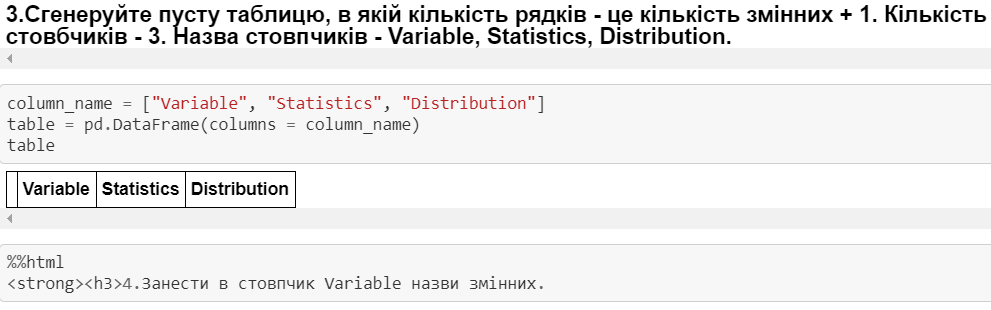
*k=(n-1)/2*, якщо n не парне.

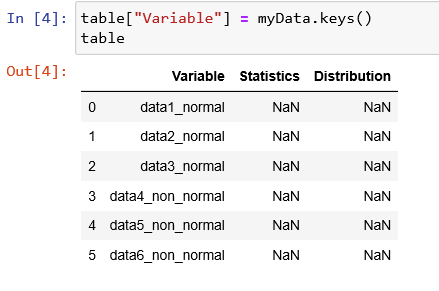
{ – відомі константи.

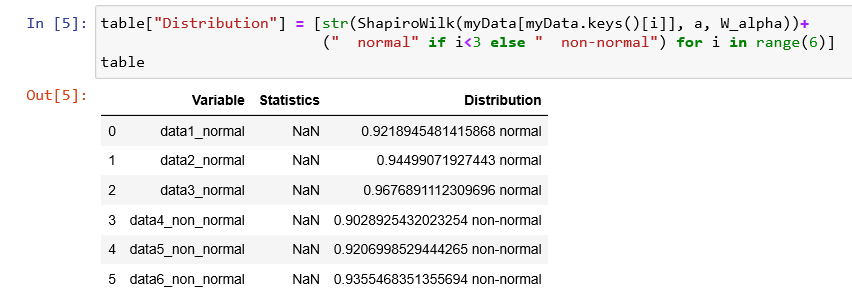
4.2. практична реалізація





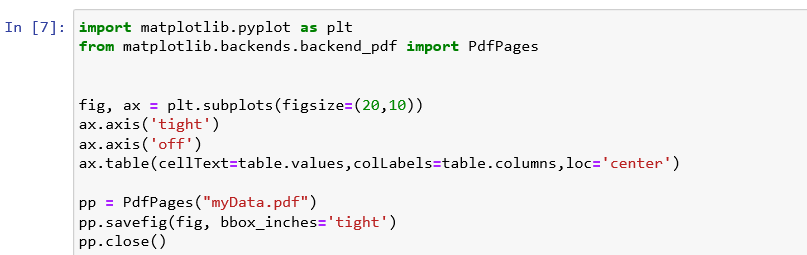








Зберегти в пдф



4.3. Висновки до розділу

У ході виконання роботи був використаний критерій Шапіро-Уілка, для оцінки розподілу даних, який побудований на відношенні оцінки дисперсії до її звичайної оцінки методом максимальної правдоподібності.

Елементи описової статистики є необхідною складовою, адже вони надають інформацію про набір спостережень. Медіана, мода, середнє значення, тощо є важливими компонентами аналізу даних та визначення їх розподілу.

1. **Критерій Пірсона для порівняння декількох груп за розподіленням ознаки**
   1. Теоретичні відомості

Критерій згоди Пірсона — один з найвідоміших критеріїв тому його часто і називають просто «критерій хі-квадрат». Використовується для перевірки гіпотези про закон розподілу. Це об'єктивна оцінка близькості емпіричних розподілів до теоретичних. При цьому порівнюються частоти названих рядів розподілу, виявляються розбіжності між ними і визначається вірогідність цих розбіжностей. Значення параметра - квадрат зростає із збільшенням різниці між частотами. Величина також залежить від числа ступенів вільності. Чим менше значення , тим вищі його ймовірність і вірогідність.

При використанні необхідно пам'ятати про достатньо велике число одиниць вибірки (n > 50) і величини частот (n > 5).

У якості нульвої гіпотези критерію Пірсона слугує вираз «ознаки не пов’язані».

розраховується за формулою , де

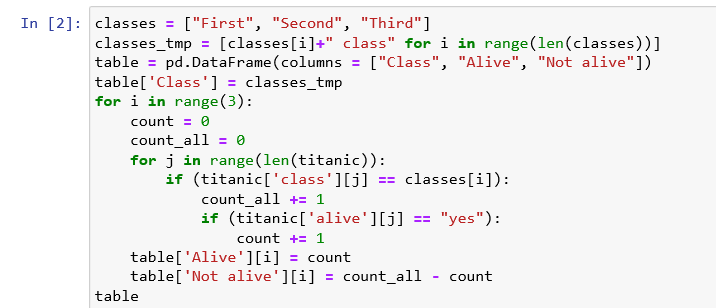
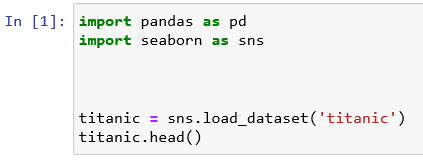
*-* значення, що досягаються шляхом вирівнювання величин з збереженням відсоткового відношення всередині груп.

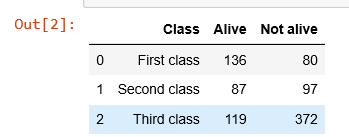
Після знаходження значення необхідно визначити ступінь вільності критерію за формулою *r=m-1,* де *m-* кількість градацій ознаки, що визначається у таблиці спряженості.

У випадку, коли , приймають гіпотезу, що розбіжності між значеннями не є статистично значимими. В іншому випадку приймають протилежну гіпотезу.

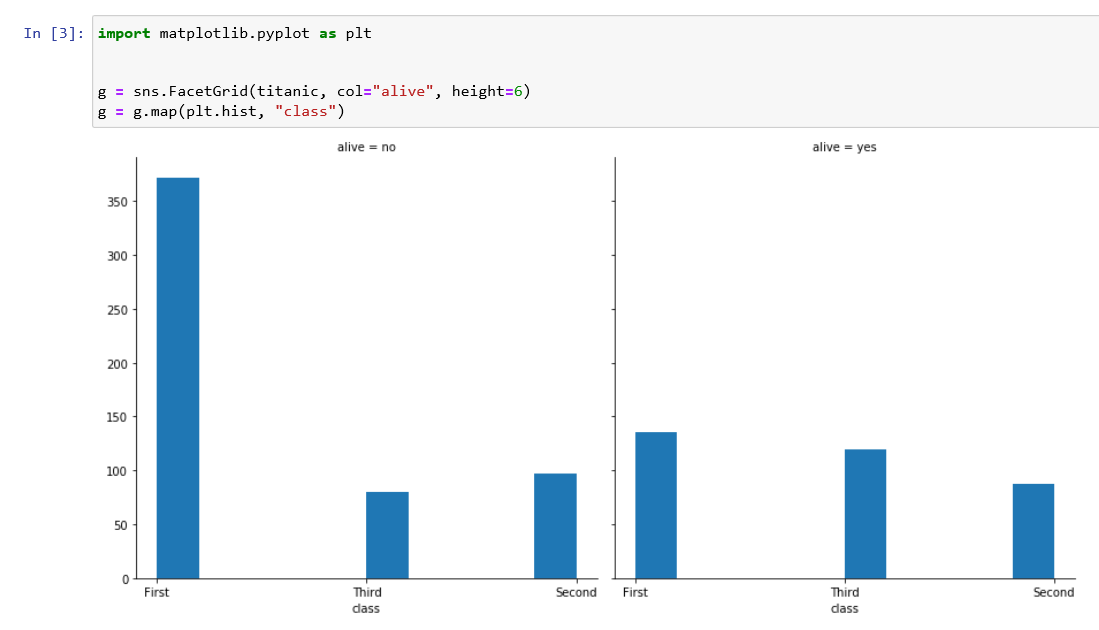
5.2. практична реалізація

**1. Обрати реальні дані з kaggle або сгенерувати випадкові дані, що містять розподіл деякої ознаки в двох групах.**

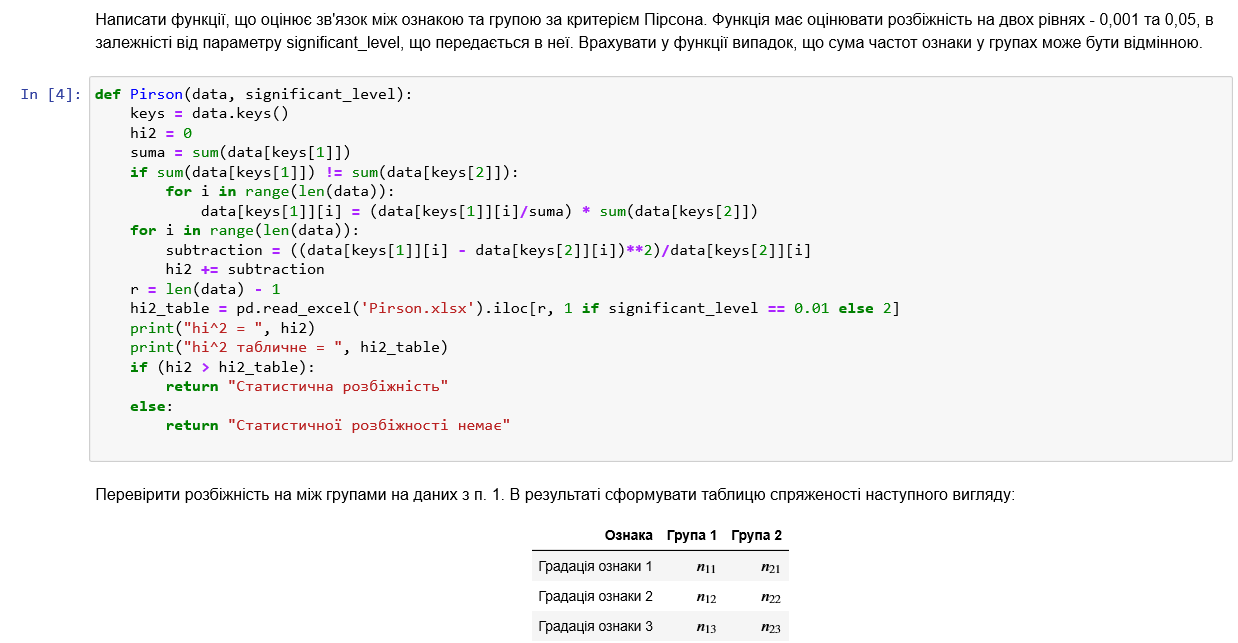




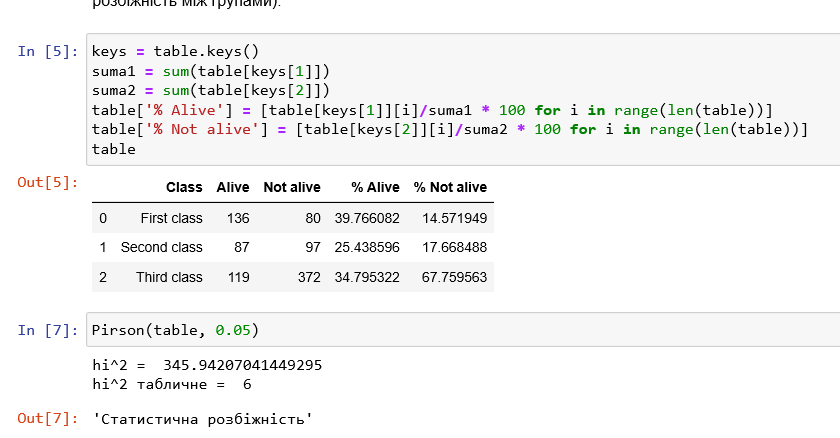
1. **Побудувати графік, що відображає розподіл ознак в групах.**



**3. Написати функції, що оцінює зв'язок між ознакою та групою за критерієм Пірсона. Функція має оцінювати розбіжність на двох рівнях - 0,001 та 0,05, в залежністі від параметру significant\_level, що передається в неї. Врахувати у функції випадок, що сума частот ознаки у групах може бути відмінною.**



**4. Перевірити розбіжність на між групами на даних з п. 1.**



5.3. Висновки до розділу

У ході виконання роботи було виконано порівняння теоретичних та практичних значень, що були згенеровані, за допомогою критерія Пірсона. Було використано для обробки даних візуалізацію розподілу даних та таблицю спряженості. Було реалізовано функцію що оцінює зв'язок між ознакою та групою за критерієм Пірсона.

1. **Парний t –критерій Стьюдента**
   1. Теоретичні відомості

t-критерій Стьюдента - загальна назва для класу методів статистичної перевірки гіпотез (статистичних критеріїв), заснованих на розподілі Стьюдента. Найбільш часті випадки застосування t-критерію пов'язані з перевіркою рівності середніх значень у двох вибірках.

Для застосування даного критерію необхідно, щоб вихідні дані мали нормальний розподіл. У разі застосування двохвибіркового критерію для незалежних вибірок також необхідне дотримання умови рівності дисперсій. Існують, альтернативи критерію Стьюдента для ситуації з нерівними дисперсіями.

У якості нульвої гіпотези слугує вираз «парні групи рівні»

Для перевірки чи є зв’язок між групами використовують коефіцієнт кореляції Пірсона й обчислють за формулою:

Далі розраховують кількість рівнів свободи  𝑑𝑓=𝑛−2, і за отриманими значеннями роблять висновок, про градацію зв’язку.

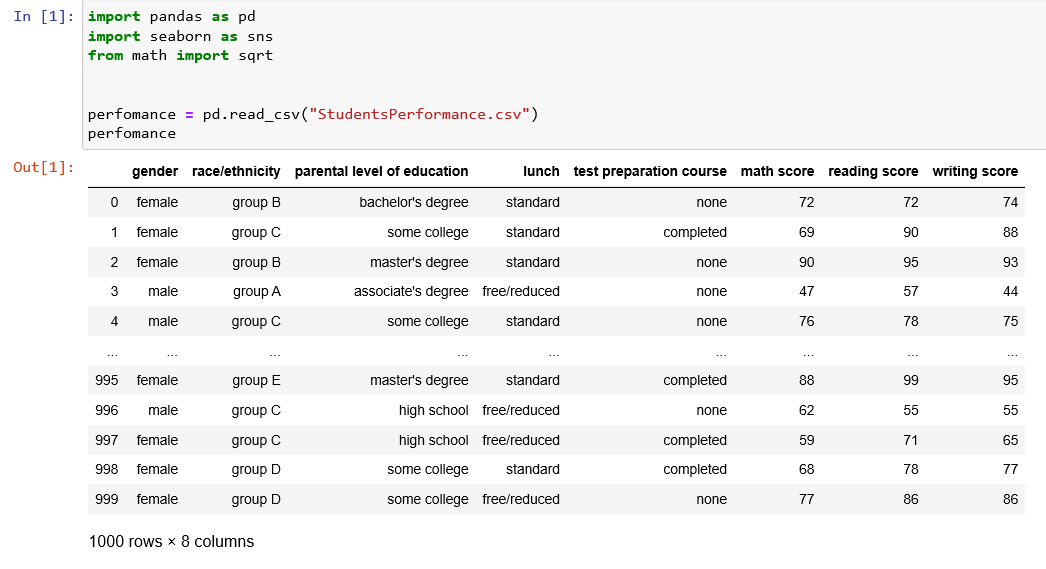
Після цього можна розрахувати критерій Стьюдента за формулою:

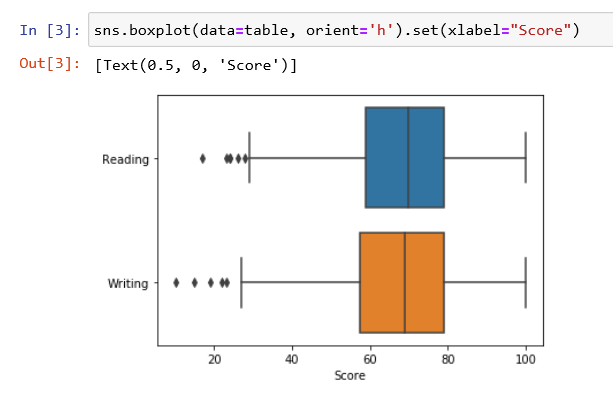
– середнє значення різниці між значеннями;

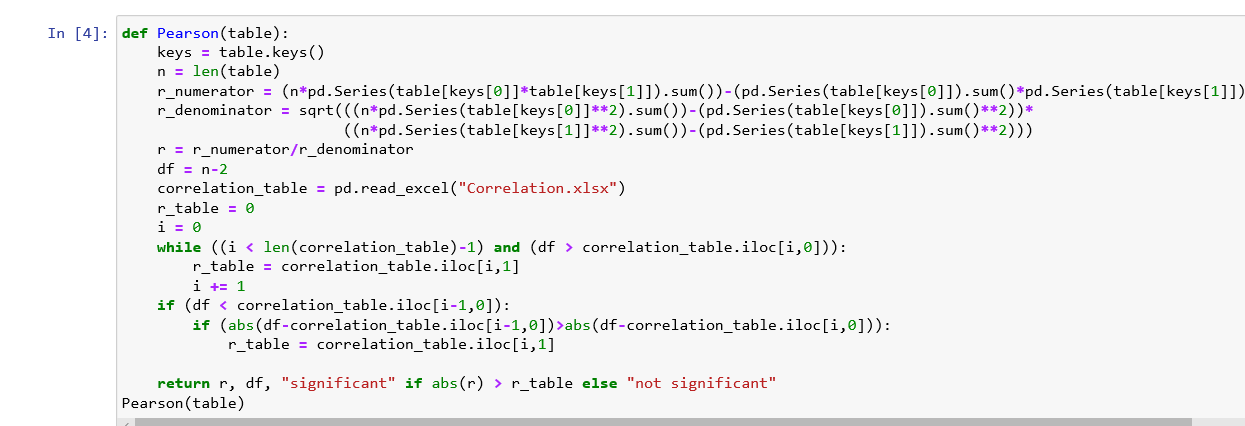
стандарне відхилення різниці між заченнями.

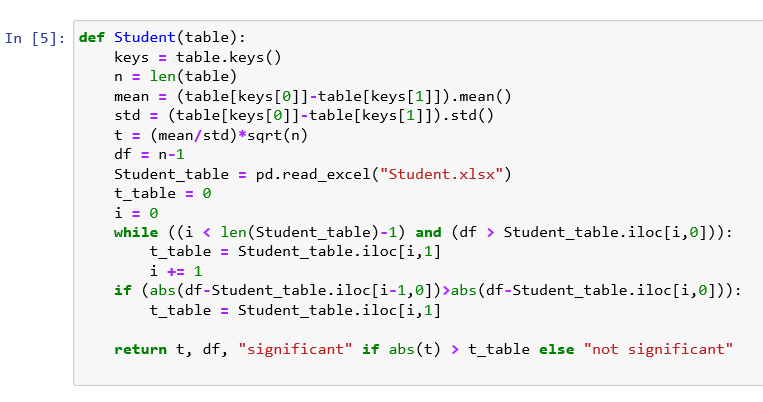
У випадку, коли , приймають гіпотезу про відмінність в середніх. В іншому випадку приймають протилежну гіпотезу.

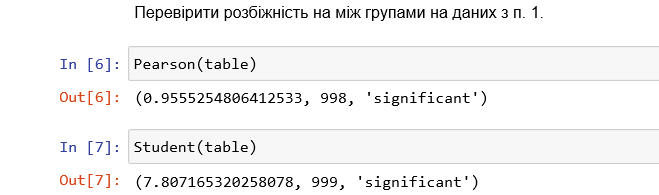
6.2. практична реалізація

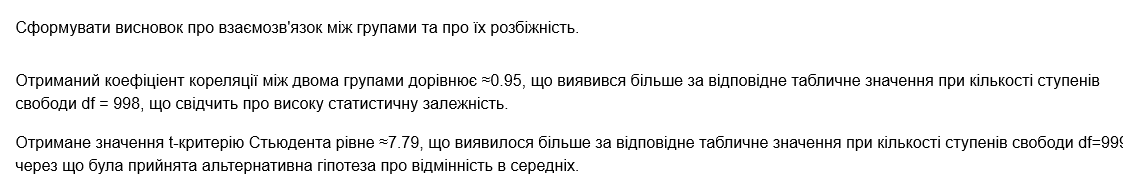












* 1. Висновки до розділу

У ході виконання роботи було використано t–критерієм Стьюдента та розроблено алгоритм його реалізації. Дані було згенеровано в ручну. Була написана функція для перевірки взаємозв’язку використовувався коефіцієнт кореляції Пірсона і функція для визначення відмінностей між групами за критерієм Стьюдента.

На основі згенерованих даних було зроблено висновок, що відсутня залежність між групами, значення коефіцієнту Стьюдента менше ніж його критичне значення.

1. **t -критерій Стьюдента для незалежних вибірок**
   1. Теоретичні відомості

t-критерій Стьюдента для незалежних вибірок застосовується для порівняння середніх значень двох незалежних між собою вибірок.

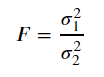
Умови застосування:

* Порівнянні значення не становлять пару корелюють значень
* Розподіл ознак в кожній вибірці відповідає нормальному розподілу
* Дисперсії ознаки в вибірках приблизно рівні (перевіряється за допомогою критерію F-Фішера)

Нульова гіпотеза:



Генеральні дисперсії оцінюються на основі вибірок, а сам критерій є відношенням однієї дисперсії до іншої:



У залежності від того, чи доведена чи ні рівність дисперсій використовують одну з двох формул обчислення критерія Стьюдента:

- у випадку коли дисперсії рівні;

- у випадку коли дисперсії не рівні.

 - середньоарифметичне для двох груп;

- стандартне відхилення;

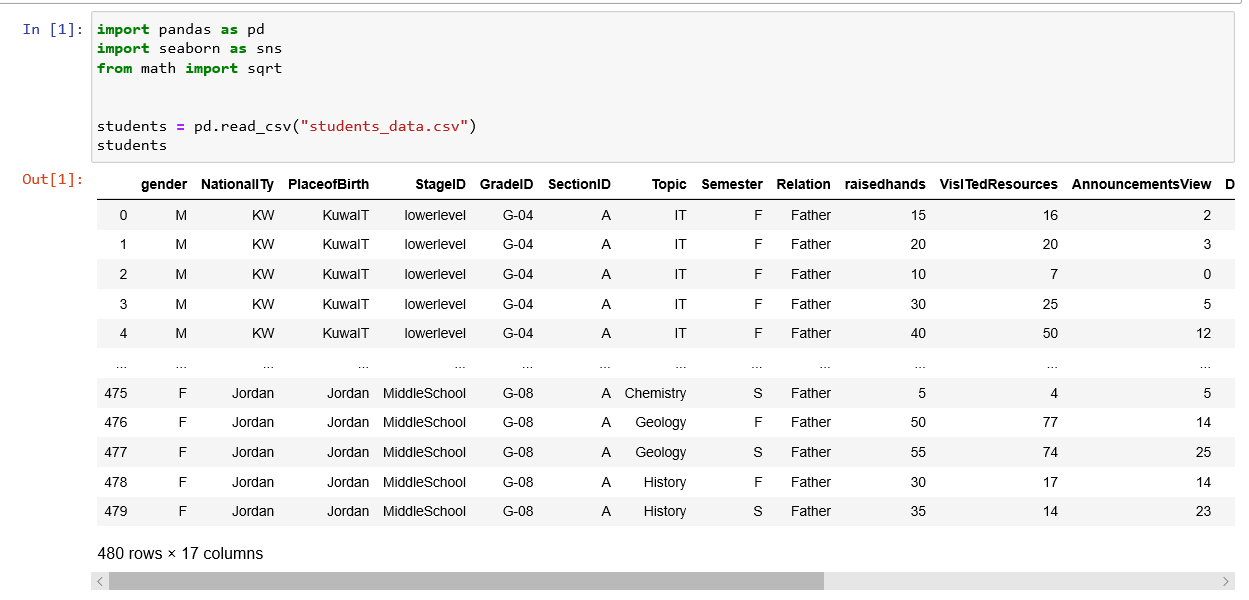
- кількість спостережень в вибірках.

Далі необхідно розрахувати кількість ступенів свободи за формулою 𝑑𝑓=𝑁1+𝑁2−2 і визначити теоретичне значення критерію.

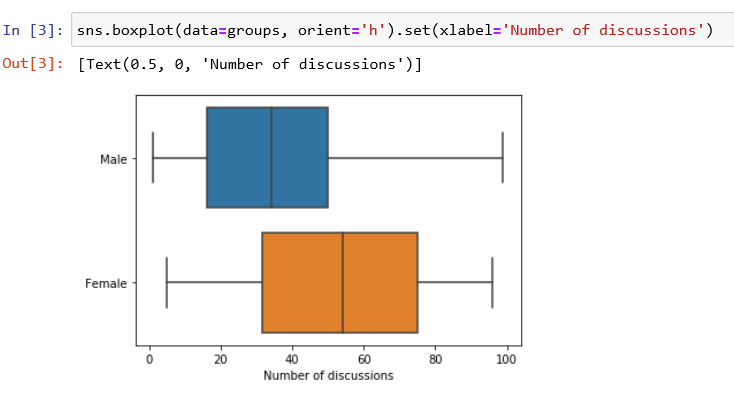
У випадку коли практичне значення більше ніж теоретичне, роблять висновок про статистичну відмінність середніх. У іншому випадку – навпаки.

7.2. практична реалізація

1. **Обрати реальні дані з kaggle або сгенерувати випадкові дані. Дані мають відповідати вимогам викладеним вище.**



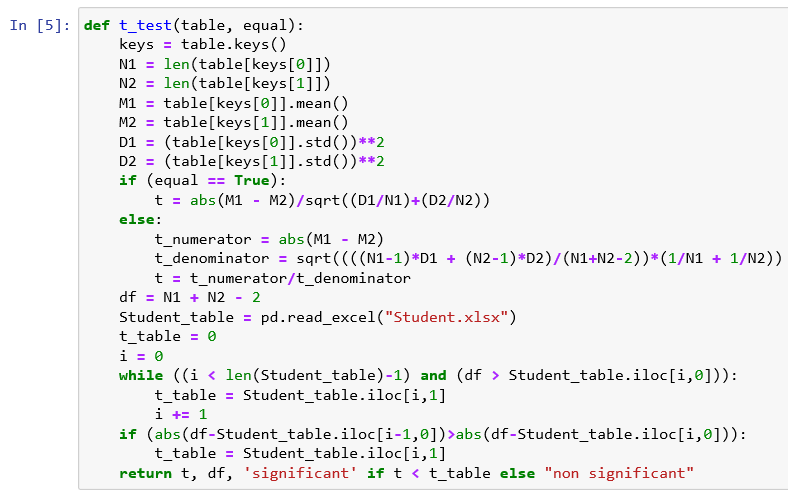
1. **Побудувати графік, що відображає розкид даних в групах.**



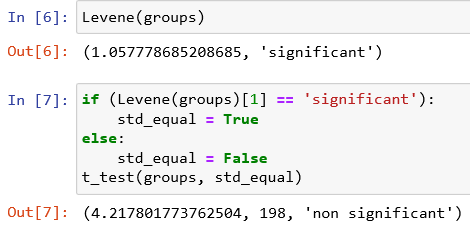
1. **Написати функції:**
2. **для оцінки вімінностей дисперсій між групами за критерієм Лівеня. Функція має повертати значення тесту та результат оцінки рівності дисперсій (significant або not significant).**



1. **для визначення відмінності між групами за критерієм Стьюдента в залежності від рівності/відмінності дисперсій. Функція має повертати значення критерію Стьюдента,значення ступеня свободи та результат оцінки значимості зв'зку (significant або not significant).**



1. **Перевірити розбіжність на між групами на даних з п. 1.**



7.3. Висновки до розділу

У ході виконання роботи було використано t-критерій Стьюдента для незалежних вибірок. Дані було згенеровано в ручну. Була написана функція для оцінки вімінностей дисперсій між групами за критерієм Лівеня і функція для для визначення відмінності між групами за критерієм Стьюдента в залежності від рівності/відмінності дисперсій.

1. **Дисперсійний аналіз**
   1. Теоретичні відомості

Дисперсійний аналізявляє собою статистичний метод аналізу результатів, які залежать від якісних ознак.

Кожен фактор може бути дискретною чи неперервною випадковою змінною, яку розділяють на декілька сталих рівнів (градацій, інтервалів). Якщо кількість вимірювань (проб, даних) на всіх рівнях кожного з факторів однакова, то дисперсійний аналіз називають рівномірним, інакше — нерівномірним.

Сутність цього аналізу полягає в тому, що загальну дисперсію досліджуваної ознаки розділяють на окремі компоненти, які обумовлені впливом певних конкретних чинників. Істотність їх впливу на цю ознаку здійснюється методом дисперсійного аналізу.

У якості нульової гіпотези слугує твердження: «середні незалежних груп рівні».

Для того щоб обчислити, необхідно виконати наступні кроки:

* Розрахувати внутрішньогрупову дисперсію 𝑆𝑆факт= -;
* Розрахувати загальногрупову дисперсію 𝑆𝑆заг=;
* Розрахувати випадкову (залишкову) величину  𝑆𝑆вип =𝑆𝑆заг−𝑆𝑆факт
* Визначити кількість ступенів свободи:

𝑑факт=𝑐−1

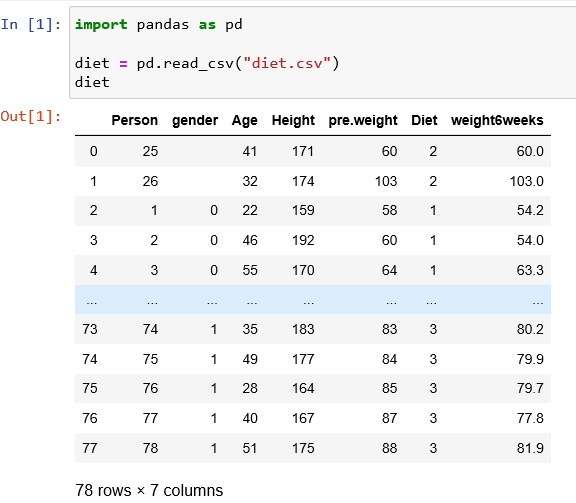
𝑑заг=𝑁−1

𝑑вип=𝑑заг−𝑑факт

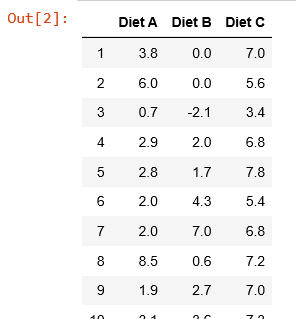
* Розрахувати 𝐹емпір за формулою 𝐹емпір=
* Співставити теоретичне та практичне значення критерію. Якщо практичне більше, тоді гіпотеза про рівність середніх видхиляється.

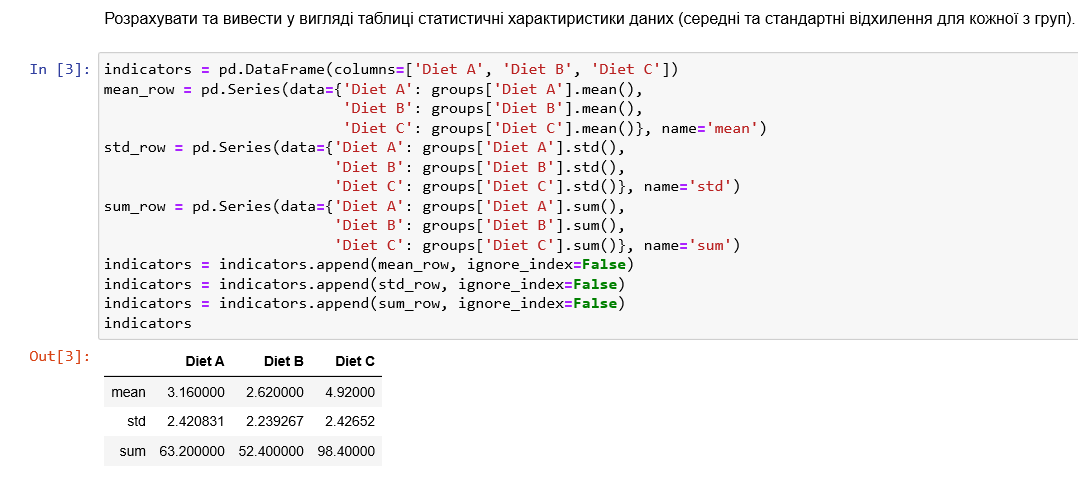
8.2. практична реалізація

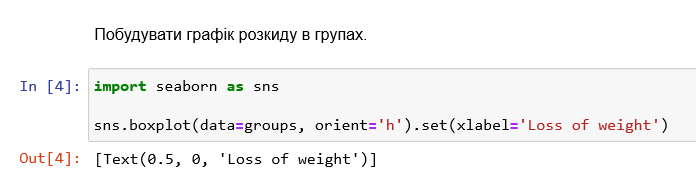
1. **Розрахувати та вивести у вигляді таблиці статистичні характиристики даних (середні та стандартні відхилення для кожної з груп).**









1. 
2. **Побудувати функцію, що реалізовує однофакторний дисперсійний аналіз.**

**def ANOVA(groups):**

**keys = groups.keys()**

**c = len(groups.columns)**

**Tc = [groups[keys[i]].sum() for i in range(c)]**

**n = len(groups)**

**N = n\*c**

**sq\_xi\_sum = 0**

**for i in range(c):**

**for j in range(1,n+1):**

**sq\_xi\_sum += (groups[keys[i]][j])\*\*2**

**SS\_fact = (sum(Tc[i]\*\*2 for i in range(len(Tc)))/n) - ((sum(Tc)\*\*2)/N)**

**SS\_zag = sq\_xi\_sum - (sum(Tc)\*\*2)/N**

**SS\_vyp = SS\_zag - SS\_fact**

**d\_fact = c - 1**

**d\_zag = N - 1**

**d\_vyp = d\_zag - d\_fact**

**MS\_fact = SS\_fact/d\_fact**

**MS\_vyp = SS\_vyp/d\_vyp**

**F\_emp = MS\_fact/MS\_vyp**

**Fisher\_table = pd.read\_excel("Fisher.xlsx")**

**df\_index1 = 0**

**df\_index2 = 0**

**for i in range(1,Fisher\_table.shape[0]-1):**

**if (Fisher\_table.iloc[i,0] <= d\_vyp):**

**df\_index1 = i**

**if (Fisher\_table.iloc[Fisher\_table.shape[0]-2,0] < d\_vyp):**

**df\_index1 = Fisher\_table.shape[0]-1**

**for i in range(1,Fisher\_table.shape[1]-1):**

**if (Fisher\_table.iloc[0,i] <= d\_fact):**

**df\_index2 = i**

**if (Fisher\_table.iloc[0,Fisher\_table.shape[1]-2] < d\_fact):**

**df\_index2 = Fisher\_table.shape[1]-1**

**F\_kryt = Fisher\_table.iloc[df\_index1, df\_index2]**

**table\_cols = ['SS(факт)','SS(заг)','SS(вип)','d(факт)','d(заг)','d(вип)','MS(факт)','MS(вип)','F(емпір)','F(крит)']**

**res\_table = pd.DataFrame([SS\_fact,SS\_zag,SS\_vyp,d\_fact,d\_zag,d\_vyp,MS\_fact,MS\_vyp,F\_emp,F\_kryt],**

**columns=['Results'],index=table\_cols)**

**significant = True if abs(F\_emp) < F\_kryt else False**

**if (significant == True):**

**print('F(емпір) < F(крит)')**

**print('significant')**

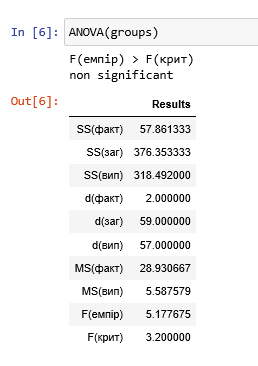
**else:**

**print('F(емпір) > F(крит)')**

**print('non significant')**

**return res\_table**

1. **Вивести результати розрахунку кожного з етапів вказаних вище у вигляді таблиці.**



8.3. Висновки до розділу

У ході виконання роботи було розглянуто дисперсійний аналіз та реалізовано алгоритм його розрахунку. Дані було згенеровано в ручну. Було реалізовано функцію, що реалізовує однофакторний дисперсійний аналіз.

Було зроблено висновок, що нульова гіпотеза про рівність середніх відхилина.