Міністерство освіти і науки України
НТУУ «КПІ ім. Ігоря Сікорського»
Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці

Лабораторна робота №2

з дисципліни «Вступ до інтелектуального аналізу даних»

Тема «Вступ до графічних методів у статистиці»

Варіант №19

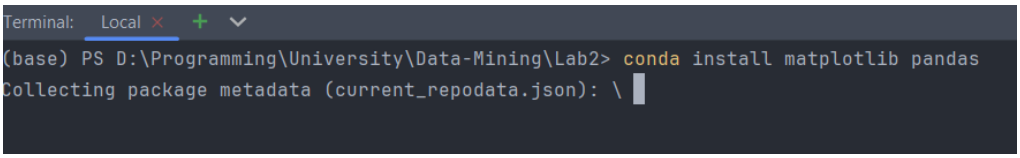
Студента 3-го курсу НН ІАТЕ гр. ТР-12
Ковальова Олександра
Перевірив: д.т.н., проф. Путренко В. В.

КИЇВ 2024

Мета: Ознайомитись з прикладом створення графіків за допомогою бібліотеки MatPlotLib. Ознайомитись з бібліотекою Pandas. Виконати поставлене завдання.

Хід роботи

Встановимо потрібні пакети (бібліотеки) за допомогою пакетного менеджеру conda:



Завдання: побудувати 4 діаграми, а саме кругову, стовпчасту діаграму, гістограму, та діаграму «ящик з вусами».

Код до кожної діаграми знаходиться в окремій комірці Jupyter Notebook, що дозволяє модифікувати код до однієї діаграми не зачіпаючи інші.

Для початку треба імпортувати пакети, підключитись до бази даних, створити об’єкт курсор:

In [ ]:
```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

db_filename = 'data/cereals.db'
conn = sqlite3.connect(db_filename)
c = conn.cursor()
```

У першій комірці знаходиться код, який відповідає за здобуття потрібних даних, конвертацію у потрібний формат, та побудову кругової діаграми. Спочатку виконується запит – результатом є кількість продуктів відповідно кожного виробника. Далі ці дані за допомогою бібліотеки Pandas конвертуються в псевдо-табличний формат, який сприймається бібліотеками для побудови графіків (у нашому випадку – MatPlotLib). Після цього у псевдоніму plt, який відповідає модулю pyplot MatPlotLib, викликається метод pie. Використовується для побудови кругових діаграм. Аргументи: значення (кількість записів відповідно виробників), підписи (виробники), тінь від графіку (є чи ні). Далі йде вказівка, що потрібні рівні співвідношення сторін осей, тобто графік повинен бути саме круговим, а не овальним. І наприкінці використовується метод show(), щоб вивести графік на екран.

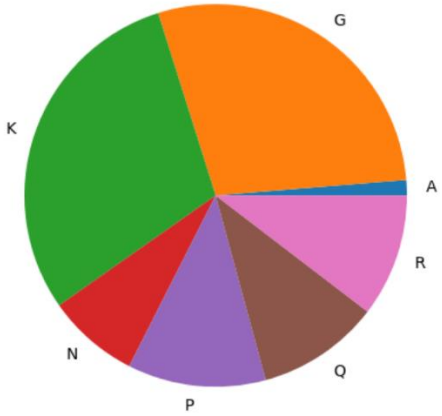
In [3]:
```
c.execute('''SELECT
Manufacturer, count(*) FROM
cereals GROUP BY Manufacturer''')

counts = c.fetchall()

manuStats = (pd.DataFrame.from_records(counts,
                                       columns=['manufacturer', 'value']))

plt.pie(manuStats['value'],
        labels=manuStats['manufacturer'],
        shadow=False)
plt.axis('equal')
plt.show()
```

Результат:



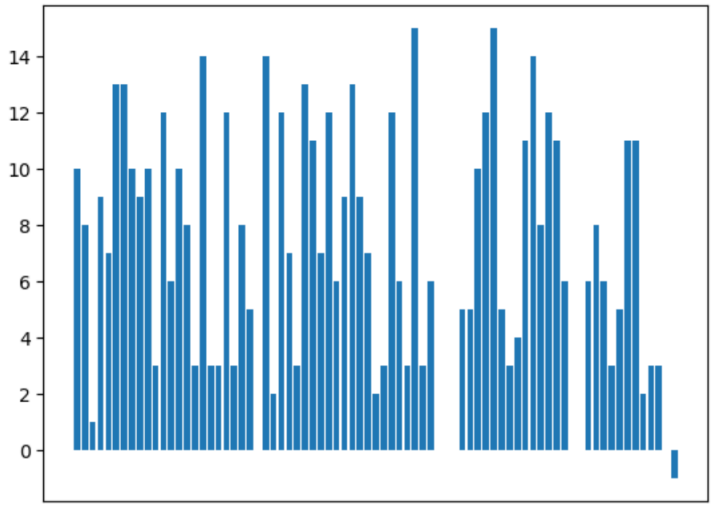
Друга комірка потрібна для генерації стовпчастої діаграми. Робимо SQL запит до таблиці, будуємо Pandas дата-фрейм, використовуємо метод bar для побудови. Аргументи – список з чисел, від 0 до довжини масиву значень (координати X), та самі значення. Використовуємо метод xticks з порожнім списком, щоб прибрати усі підписи та мітки з осі X. Виводимо графік на екран.

```
In [5]: c.execute('''SELECT Cereal,
        Sugars FROM cereals''')
        sugars = c.fetchall()

        sugarFrame = (pd.DataFrame.
                      from_records(sugars,columns=['Cereal', 'Sugar']))

        plt.bar(
            range(len(sugarFrame['Sugar'])),
            sugarFrame['Sugar'])
        plt.xticks([])
        plt.show()
```

Результат:



Можна побачити відхилення в кінці графіку. Це явище можна пояснити тим, що у виробника Q, в продукті «Quaker Oatmeal» значення кількості цукру дорівнює -1.

Третя комірка потрібна для коду, з якого в результаті буде отримана гістограма. Вони використовуються для візуалізації розподілу даних.

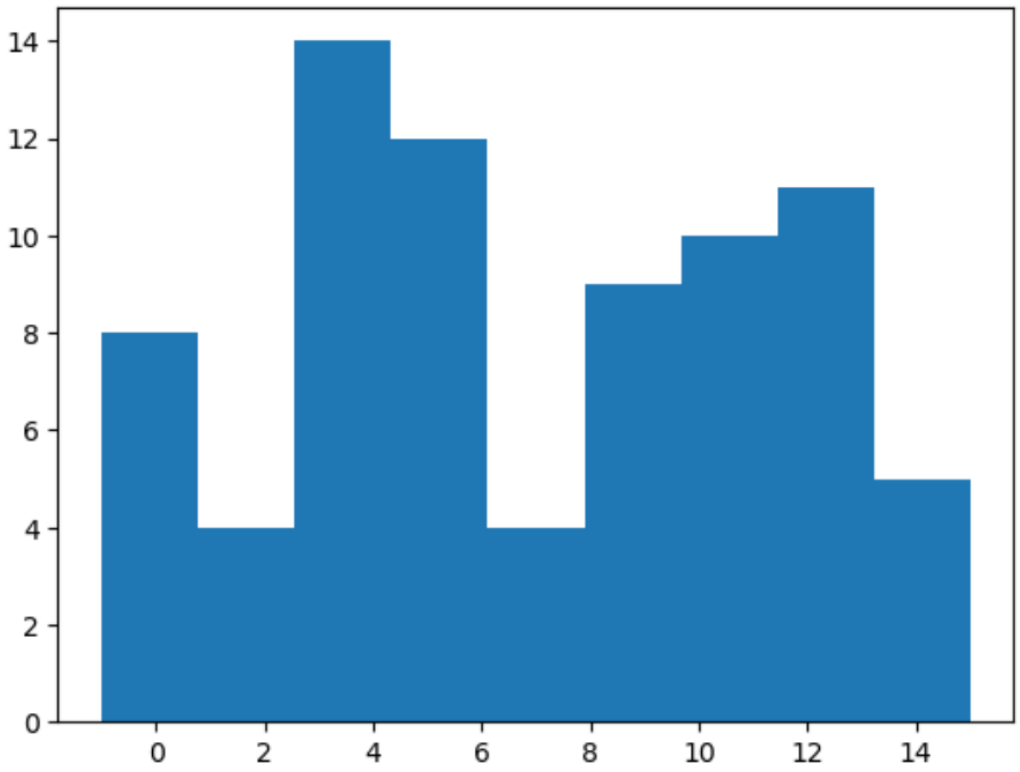
Спочатку отримаємо значення цукру в грамах з усіх продуктів. Будуємо датафрейм. За допомогою методу hist буде побудована гістограма, за всіма значеннями, отриманими раніше. Бібліотека має всі необхідні засоби для обрахунку. Також, використовується параметр bins, який вказує кількість інтервалів (або "бінів"), на які буде розбито діапазон значень даних при побудові гістограми. Інтервали визначають ширину та кількість стовпців, які будуть відображені на гістограмі. У даному випадку використовується правило $\sqrt{N}$, яке потрібне для визначення оптимальної кількості бінів (інтервалів). Так як кількість записів у базі даних дорівнює 77, то найближчим цілим значенням кореню є 9.

```
In [6]: c.execute('''SELECT Sugars
        FROM cereals''')
        sugar = c.fetchall()

        sugarFrame = (pd.DataFrame.
                      from_records(sugar, columns=['Sugar']))

        plt.hist(sugarFrame['Sugar'],
                 bins=9)
        plt.show()
```

Результат:



Остання комірка потрібна для побудови діаграми «ящик з вусами». Ця діаграма є потужним інструментом для візуалізації статистичних характеристик розподілу даних та надає компактну та інформативну інформацію про розподіл числових даних.

Основні застосування включають:

- Діапазон значень: Ящик представляє інтерквартильний розмах (IQR), який визначається від верхнього (Q3) до нижнього (Q1) квартиля. Величина цього ящика вказує на розкид даних в центральній частині розподілу.
- Вуси: Вуси розширюються від краю ящика до максимального та мінімального значень даних, які не ввійшли в діапазон викидів. Вони дозволяють оцінювати розмах всіх даних та ідентифікувати можливі викиди.
- Середнє або медіана: Лінія в середині ящика може представляти середнє арифметичне або медіану розподілу, залежно від вибору користувача або характеристик даних.
- Викиди: Точки за межами вусів вказують на індивідуальні значення, які можуть бути викидами або аномаліями в розподілі.
- Порівняння груп: Діаграма дозволяє порівнювати розподіли даних між різними групами чи категоріями.
- Симетрія та асиметрія: Форма та розташування ящика можуть вказувати на симетричний чи асиметричний характер розподілу.

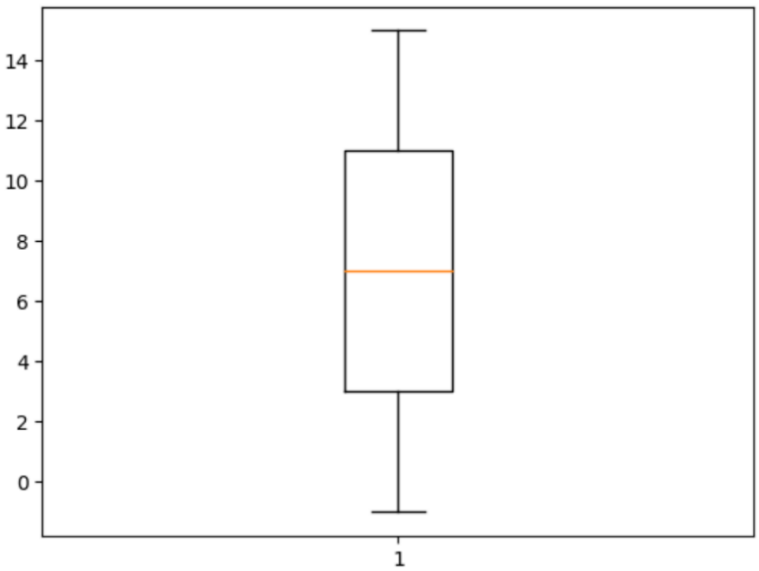
Процес побудови діаграми не відрізняється від попередніх. У бібліотеці є всі необхідні засоби для розрахунків.

```
In [8]: c.execute('''SELECT Manufacturer, Sugars
        FROM cereals''')
        sugarByMan = c.fetchall()

        sugarBoxFrame = (pd.DataFrame.from_records(
            sugarByMan, columns=['Manufacturer', 'Sugar']))

        plt.boxplot(sugarFrame['Sugar'])
        plt.show()
```

Результат:



Після побудови діаграми, можна сказати, що найменше значення серед вибірки дорівнює -1, максимальне – 15. Близько 50% спостережень знаходяться між 3 та 11. Медіана дорівнює 7. Викидів (аномалій) не спостерігається.

Висновок: Під час виконання лабораторної роботи були набуті практичні навички роботи з діаграмами різних типів: круговою, стовпчастою, гістограмою, ящиком з вусами. Були виконані певні зразкові задачі з використанням бібліотек Pandas та MatPlotLib.

Програмний код

Notebook.ipynb:

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

db_filename = 'data/cereals.db'
conn = sqlite3.connect(db_filename)
c = conn.cursor()

c.execute('''SELECT
Manufacturer, count(*) FROM
cereals GROUP BY Manufacturer''')
counts = c.fetchall()
manuStats = (pd.DataFrame.from_records(counts,
                                       columns=['manufacturer', 'value']))

plt.pie(manuStats['value'],
        labels=manuStats['manufacturer'],
        shadow=False)
plt.axis('equal')
plt.show()

c.execute('''SELECT Cereal,
Sugars FROM cereals''')
sugars = c.fetchall()
sugarFrame = (pd.DataFrame.
              from_records(sugars,columns=['Cereal', 'Sugar']))
plt.bar(
    range(len(sugarFrame['Sugar'])),
    sugarFrame['Sugar'])
plt.xticks([])
plt.show()

c.execute('''SELECT Sugars
FROM cereals''')
sugar = c.fetchall()
sugarFrame = (pd.DataFrame.
              from_records(sugar, columns=['Sugar']))

plt.hist(sugarFrame['Sugar'],
         bins=9)
plt.show()
c.execute('''SELECT Manufacturer, Sugars
FROM cereals''')
sugarByMan = c.fetchall()
sugarBoxFrame = (pd.DataFrame.from_records(
    sugarByMan, columns=['Manufacturer', 'Sugar']))

plt.boxplot(sugarFrame['Sugar'])
plt.show()
```