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

Лабораторна робота №1

з дисципліни «Вступ до машинного навчання»

Тема «Використання бібліотек Pandas та Matplotlib»

Варіант №17

Студента 4-го курсу НН ІАТЕ гр. ТР-12
Ковальова Олександра
Перевірив: вик. Ліскін В’ячеслав Олегович

КИЇВ 2025

Хід роботи.

Відкрити та зчитати файл з даними.

В роботі використовується Jupyter Notebook.

Було використано бібліотеку pandas для роботи з даними у Python. Спочатку відбувається імпорт цієї бібліотеки за допомогою ключового слова import. Далі, з метою зчитування даних з CSV-файлу, використовується функція read_csv. Вона приймає шлях до файлу, у цьому випадку "Input_Lab-1_Vehicle_Sales.csv", і повертає об'єкт типу DataFrame, який є зручною структурою даних для роботи з таблицями.

Результат виконання цієї функції зберігається в змінній df. DataFrame дозволяє зручно обробляти та аналізувати дані, виконувати операції фільтрації, агрегації та модифікації, що значно спрощує роботу з великими наборами даних.

```
1  import pandas as pd
2  df = pd.read_csv("data/Input_Lab-1_Vehicle_Sales.csv")
   [83]
```

Визначити та вивести кількість записів та кількість полів у кожному записі.

Було використано атрибут shape об'єкта DataFrame для визначення кількості записів та полів. Атрибут shape повертає кортеж, який містить два значення: перше значення є кількістю рядків (записів), а друге – кількістю стовпців (полів).

Змінна num_records отримує кількість рядків, що визначається через df.shape[0], де df – це DataFrame. Аналогічно, змінна num_fields отримує кількість стовпців через df.shape[1].

Після цього, за допомогою функції print, виводяться на екран значення кількості записів та полів, формуючи відповідні повідомлення.

```
1  num_records = df.shape[0]
2  num_fields = df.shape[1]
3  print(f"Кількість записів (рядків): {num_records}")
4  print(f"Кількість полів (стовпців): {num_fields}")
   [84]
    Кількість записів (рядків): 186
    Кількість полів (стовпців): 6
```

Вивести К+7 перших та 5К-3 останніх записів.

Було визначено значення K = 2. Для перших записів обчислено first_n = K + 7, що дає 9, і використано метод head(first_n) для виведення перших 9 рядків. Для останніх

записів обчислено last_n = 5 * K - 3, що дає 7, і використано метод tail(last_n) для виведення останніх 7 рядків. Результати виводяться через print та display.

Перші 9 записів:

9 rows 9 rows × 6 cols

	Year	Month	New	Used	Total Sales New	Total Sales Used
0	2002	JAN	31106	49927	$755015820	$386481929
1	2002	FEB	27520	50982	$664454223	$361353242
2	2002	MAR	34225	58794	$805666244	$419385387
3	2002	APR	36452	59817	$846368297	$433061150
4	2002	MAY	37359	60577	$855005784	$442569410
5	2002	JUN	36348	55415	$830251613	$414731166
6	2002	JUL	30367	55235	$700530891	$384673023
7	2002	AUG	38965	60310	$934484212	$455453720
8	2002	SEP	39740	55485	$950502055	$432112270

Останні 7 записів:

7 rows 7 rows × 6 cols

	Year	Month	New	Used	Total Sales New	Total Sales Used
179	2016	DEC	26205	56234	$926735097	$655654762
180	2017	JAN	28827	56583	$1000285431	$674078790
181	2017	FEB	22413	55989	$776222008	$592255229
182	2017	MAR	27838	72717	$926703144	$744746293
183	2017	APR	26135	62740	$867722291	$654981615
184	2017	MAY	28931	68296	$937804113	$718689038
185	2017	JUN	31835	67827	$1019153457	$720739448

Визначити та вивести тип полів кожного запису.

Було використано атрибут dtypes для визначення типів даних кожного стовпця в DataFrame. Результат показує, що стовпець Year має тип int64, стовпець Month – тип object, що в даному випадку означає, що в ньому зберігаються рядкові значення, стовпці New та Used мають тип int64, а стовпці Total Sales New та Total Sales Used мають тип object, що також вказує на наявність рядкових даних.

```
1  print(df.dtypes)
[86]
```

```
Year                int64
Month              object
New                 int64
Used                int64
Total Sales New    object
Total Sales Used   object
dtype: object
```

Привести поля, що відповідають обсягам продаж, до числового вигляду (показати, що це виконано).

Було використано методи str.replace та astype для перетворення полів, що містять обсяги продажів, до числового вигляду. Спочатку з полів Total Sales New та Total Sales Used видалено символи долара за допомогою str.replace("$", ""). Потім ці стовпці були перетворені в тип int, щоб представити обсяги продажів як цілі числа, використовуючи метод astype(int).

```
df["Total Sales New"] = df["Total Sales New"].str.replace("$", "").astype(int)
df["Total Sales Used"] = df["Total Sales Used"].str.replace("$", "").astype(int)
print("\nТипи даних після перетворення:")
print(df.dtypes)
print("\nПерші 5 записів після перетворення:")
print(df.head())
[87]
```

```
Типи даних після перетворення:
Year                 int64
Month               object
New                  int64
Used                 int64
Total Sales New      int64
Total Sales Used     int64
dtype: object

Перші 5 записів після перетворення:
   Year Month    New   Used  Total Sales New  Total Sales Used
0  2002   JAN  31106  49927        755015820         386481929
1  2002   FEB  27520  50982        664454223         361353242
2  2002   MAR  34225  58794        805666244         419385387
```

Ввести нові поля: Сумарний обсяг продаж автомобілів (нових та б/в) у кожний період;

Було додано нове поле Sum, яке обчислює сумарний обсяг продажів автомобілів (нових та б/в) для кожного періоду. Це досягнуто шляхом простого складання значень зі стовпців New та Used. Результат зберігається в новому стовпці Sum.

```
df["Sum"] = df["New"] + df["Used"]
print("\nПерші 5 записів після перетворення:")
display(df.head())
[88]
```

```
Перші 5 записів після перетворення:
```

5 rows × 7 cols — Static Output

	Year	Month	New	Used	Total Sales New	Total Sales Used	Sum
0	2002	JAN	31106	49927	755015820	386481929	81033
1	2002	FEB	27520	50982	664454223	361353242	78502
2	2002	MAR	34225	58794	805666244	419385387	93019
3	2002	APR	36452	59817	846368297	433061150	96269
4	2002	MAY	37359	60577	855005784	442569410	97936

Ввести нові поля: Сумарний дохід від продажу автомобілів (нових та б/в) у кожний період;

Було введено нове поле Total Sales Sum, яке обчислює сумарний дохід від продажу автомобілів (нових та б/в) для кожного періоду. Це досягнуто шляхом складання значень зі стовпців Total Sales New та Total Sales Used. Результат зберігається в новому стовпці Total Sales Sum.

```
df["Total Sales Sum"] = df["Total Sales New"] + df["Total Sales Used"]
print("\nПерші 5 записів після перетворення:")
display(df.head())
[89]
```

Перші 5 записів після перетворення:

5 rows × 8 cols — Static Output

	Year	Month	New	Used	Total Sales New	Total Sales Used	Sum	Total Sales Sum
0	2002	JAN	31106	49927	755015820	386481929	81033	1141497749
1	2002	FEB	27520	50982	664454223	361353242	78502	1025807465
2	2002	MAR	34225	58794	805666244	419385387	93019	1225051631
3	2002	APR	36452	59817	846368297	433061150	96269	1279429447
4	2002	MAY	37359	60577	855005784	442569410	97936	1297575194

Ввести нові поля: Різницю в обсязі продаж нових та б/в автомобілів у кожен період.

Було введено нове поле Difference, яке обчислює різницю в обсязі продаж нових та б/в автомобілів для кожного періоду. Для цього використано функцію abs(), яка обчислює абсолютне значення різниці між стовпцями New та Used.

```
df["Difference"] = abs(df["New"] - df["Used"])
print("\nПерші 5 записів після перетворення:")
display(df.head())
[90]
```

Перші 5 записів після перетворення:

5 rows × 9 cols — Static Output

Month	New	Used	Total Sales New	Total Sales Used	Sum	Total Sales Sum	Difference
JAN	31106	49927	755015820	386481929	81033	1141497749	18821
FEB	27520	50982	664454223	361353242	78502	1025807465	23462
MAR	34225	58794	805666244	419385387	93019	1225051631	24569
APR	36452	59817	846368297	433061150	96269	1279429447	23365
MAY	37359	60577	855005784	442569410	97936	1297575194	23218

Змінити порядок розташування полів таким чином: Рік, Місяць, Сумарний дохід, Дохід від продажу нових автомобілів, Дохід від продажу б/в автомобілів, Сумарний обсяг продаж, Обсяг продаж нових автомобілів, Обсяг продаж б/в автомобілів, Різниця між обсягами продаж нових та б/в автомобілів.

Було змінено порядок стовпців у DataFrame відповідно до заданого списку. Для цього створено новий список new_column_order, який містить бажаний порядок полів.

Потім DataFrame було перерозміщено за допомогою індексації, де df[new_column_order] застосовує новий порядок до стовпців.

```
new_column_order = [
    "Year", "Month", "Total Sales Sum", "Total Sales New",
    "Total Sales Used", "Sum", "New", "Used", "Difference"
]
df = df[new_column_order]
display(df.head())
```

[91]

5 rows × 9 cols — Static Output

	Year	Month	Total Sales Sum	Total Sales New	Total Sales Used	Sum	New	Used
0	2002	JAN	1141497749	755015820	386481929	81033	31106	49927
1	2002	FEB	1025807465	664454223	361353242	78502	27520	50982
2	2002	MAR	1225051631	805666244	419385387	93019	34225	58794
3	2002	APR	1279429447	846368297	433061150	96269	36452	59817
4	2002	MAY	1297575194	855005784	442569410	97936	37359	60577

Визначити та вивести: Рік та місяць, у які нових автомобілів було продано менше за б/в;

Було використано метод loc для фільтрації рядків, де обсяг продажу нових автомобілів менший за обсяг продажу б/в автомобілів. У результаті вибираються лише стовпці Year та Month, що містять відповідні дані про періоди. Результат зберігається в змінній low_new_sales і виводиться за допомогою функції display(), щоб показати роки та місяці, у які нових автомобілів було продано менше за б/в.

```
low_new_sales = df.loc[df["New"] < df["Used"], ["Year", "Month"]]
display(low_new_sales)
```

[92]

11 rows — 186 rows × 2 cols

	Year	Month
0	2002	JAN
1	2002	FEB
2	2002	MAR
3	2002	APR
4	2002	MAY
...	...	...
181	2017	FEB
182	2017	MAR
183	2017	APR
184	2017	MAY
185	2017	JUN

Визначити та вивести: Рік та місяць, коли сумарний дохід був мінімальним;

Було використано метод min() для визначення мінімального значення сумарного доходу з стовпця Total Sales Sum. Потім методом loc були вибрані рядки, де значення цього стовпця дорівнює мінімальному доходу, і відображено стовпці Year, Month та Total Sales Sum.

```
min_revenue = df["Total Sales Sum"].min()
min_revenue_date = df.loc[df["Total Sales Sum"] == min_revenue, ["Year", "Month", "Total Sales Sum"]]
display(min_revenue_date)
[93]
```

1 row · 1 rows × 3 cols

	Year	Month	Total Sales Sum
97	2010	FEB	701892920

Визначити та вивести: Рік та місяць, коли було продано найбільше б/в авто.

Було використано метод max() для визначення максимального значення обсягу продажу б/в автомобілів з стовпця Used. Потім методом loc були вибрані рядки, де значення цього стовпця дорівнює максимальному обсягу продажу б/в авто, і відображено стовпці Year, Month та Used.

```
max_sold_used = df["Used"].max()
date = df.loc[df["Used"] == max_sold_used, ["Year", "Month", "Used"]]
display(date)
[94]
```

1 row · 1 rows × 3 cols

	Year	Month	Used
170	2016	MAR	73163

Визначити та вивести: Сумарний обсяг продажу транспортних засобів за кожен рік;

Було використано метод groupby("Year") для групування даних за роками, а потім застосовано метод sum() для обчислення сумарного обсягу продажу транспортних засобів (стовпець Total Sales Sum) за кожен рік.

```
yearly_sales = df.groupby("Year")[["Total Sales Sum"]].sum()
display(yearly_sales)
[95]
```

16 rows · 16 rows × 1 cols

Year	Total Sales Sum
2002	14512764648
2003	15789219836
2004	16358504971
2005	16646537437
2006	16277344524
2007	15792131352

Визначити та вивести: Середній дохід від продажу б/в транспортних засобів в місяці M, де M – це порядковий номер у списку підгрупи за абеткою.

Було обчислено порядковий номер місяця за допомогою змінної VARIANT = 17. Потім створено список місяців months, і за допомогою значення M вибрано місяць, для якого обчислюється середній дохід від продажу б/в транспортних засобів.

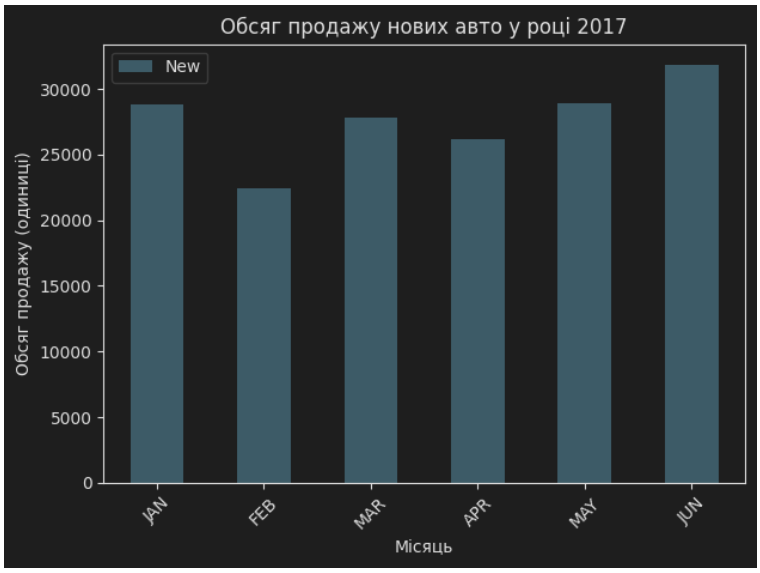
Застосовано метод groupby("Month") для групування даних за місяцями, після чого використано метод mean() для обчислення середнього доходу в стовпці Total Sales Used для кожного місяця.

```
VARIANT = 17
M = VARIANT - 12
months = ["JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL", "AUG", "SEP", "OCT", "NOV", "DEC"]
month = months[M - 1]

avg_used_car_revenue = df.groupby("Month")["Total Sales Used"].mean()
by_month = avg_used_car_revenue[month]
print("Середній дохід від продажу б/в транспортних засобів в місяці " + month + ": " + str(by_month) + "$")

[96]
Середній дохід від продажу б/в транспортних засобів в місяці MAY: 506301338.5$
```

Побудувати стовпчикову діаграму обсягу продаж нових авто у році 20YY, де дві останні цифри року визначаються як 17 – порядковий номер у списку підгрупи за абеткою.

Було обчислено рік для побудови діаграми за допомогою змінної VARIANT, де YEAR = 2017. Це значення використовується для фільтрації даних, де рік дорівнює 2017, і вибираються стовпці Month та New, що містять дані про обсяг продажу нових автомобілів у цьому році.

Для побудови стовпчикової діаграми використано бібліотеку matplotlib. Спочатку була налаштована фігура за допомогою plt.figure(figsize=(10, 6)), потім метод plot побудував стовпчикову діаграму, де по осі X виводяться місяці, а по осі Y – обсяг продажу нових автомобілів. Заголовок, підписи осей і обертання міток по осі X також були налаштовані для зручності. Використано plt.tight_layout() для коректного відображення всіх елементів графіка, а plt.show() відобразило діаграму.

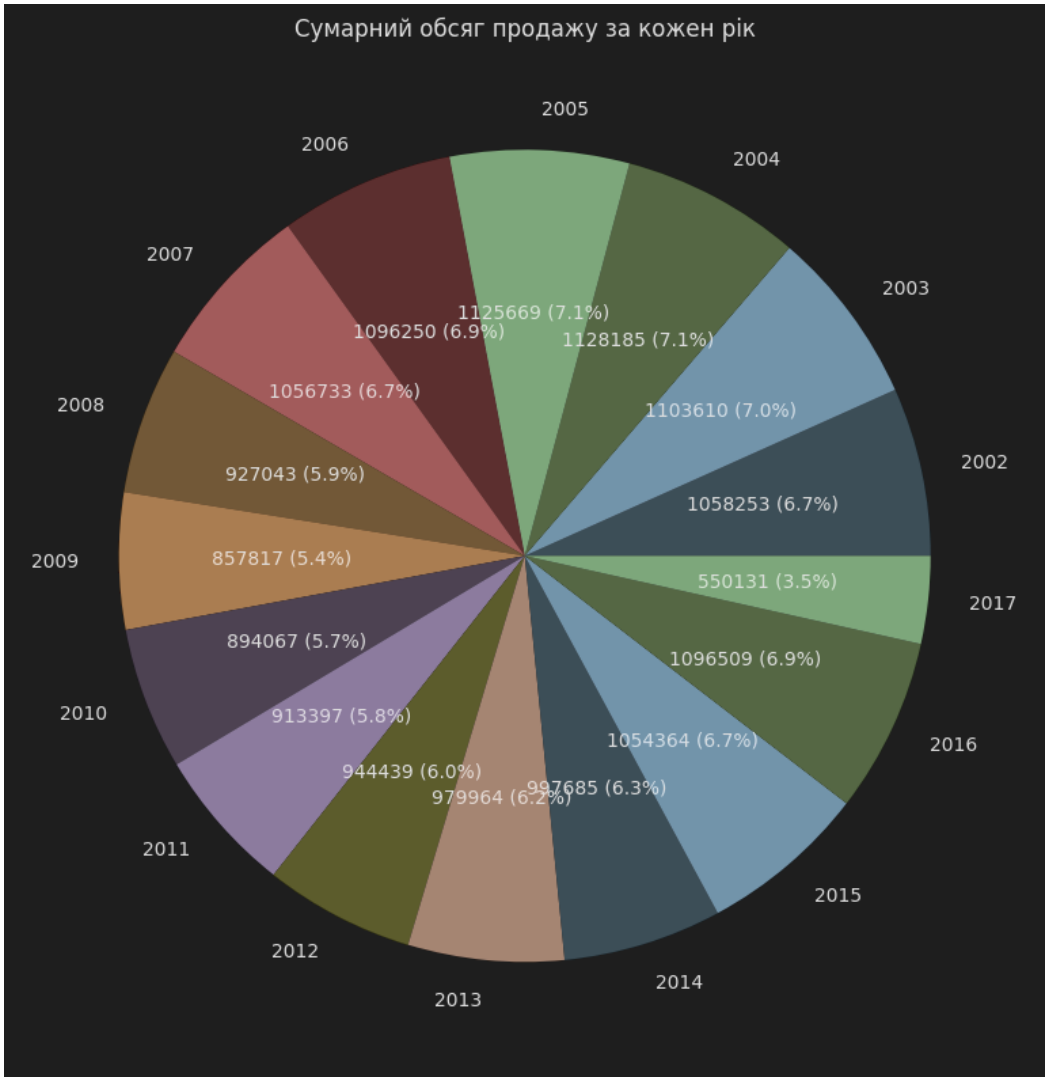


Побудувати кругову діаграму сумарного обсягу продаж за кожен рік.

Було використано метод groupby("Year") для групування даних за роками, після чого застосовано метод sum() для обчислення сумарного обсягу продажу за кожен рік. Результат зберігається в змінній yearly_sales.

Для побудови кругової діаграми використовувалась функція plt.pie(), яка відображає сумарний обсяг продажу за кожен рік. Аргумент autopct дозволяє відображати відсотки і абсолютні значення для кожного сегмента. Для відображення абсолютних значень була створена функція func(), яка обчислює абсолютне значення з відповідного відсотка.

Застосовано кольорову палітру з plt.cm.Paired.colors для діаграми, а plt.tight_layout() гарантує коректне розташування елементів. Використано plt.show() для відображення діаграми.



Побудувати на одному графіку:

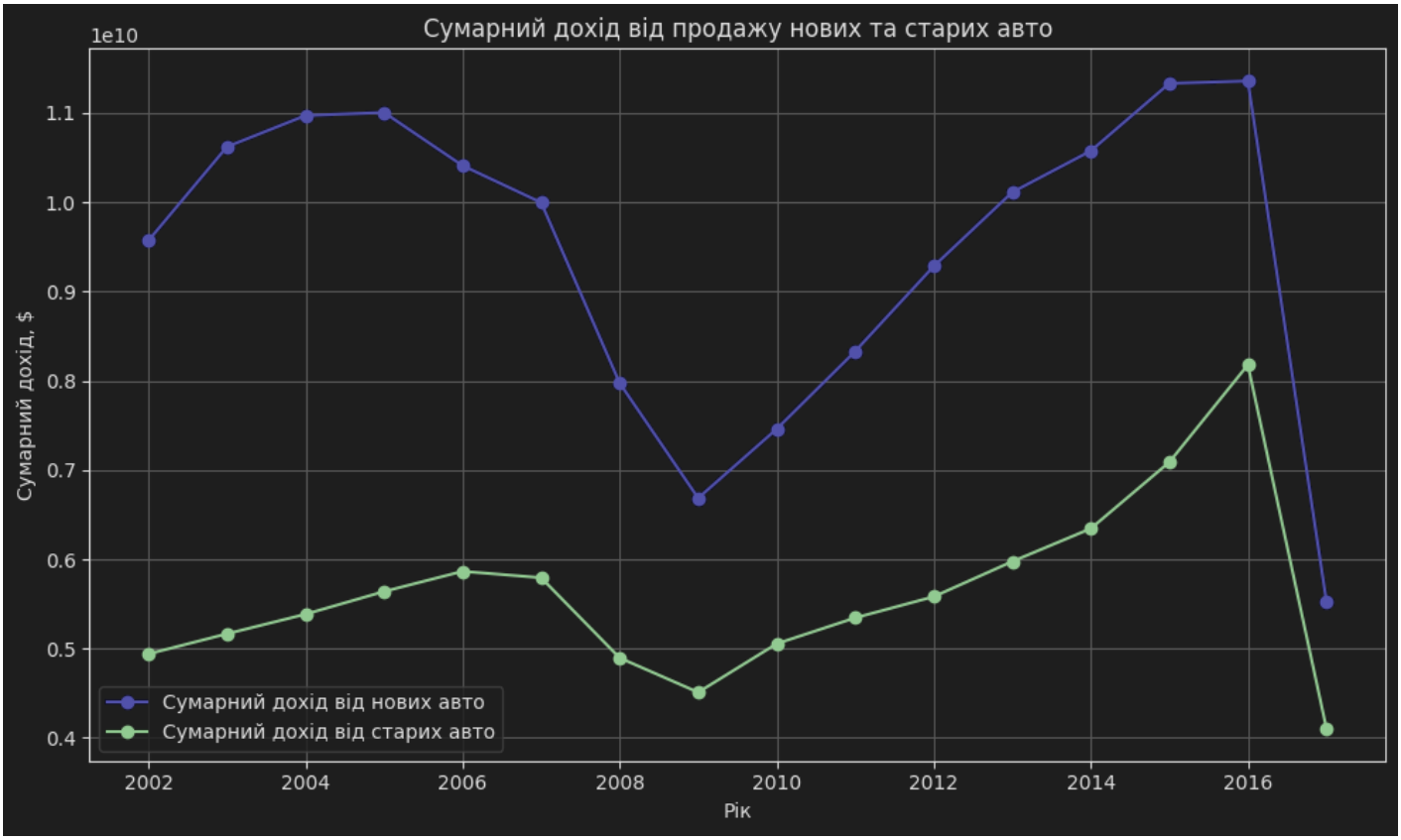
a. Сумарний дохід від продажу нових авто;

b. Сумарний дохід від продажу старих авто.

Було використано метод groupby("Year") для групування даних за роками, після чого застосовано метод sum() для обчислення сумарного доходу від продажу нових та старих автомобілів за кожен рік. Результат зберігається в змінній yearly_sales.

Для побудови графіка використовувалась функція plt.plot(), яка створює лінійні графіки для сумарного доходу від продажу нових та старих автомобілів. Лінії на графіку мають різні кольори (синій для нових авто та зелений для старих) і маркери o для позначення точок даних.

Заголовок графіка, підписи осей та легенда додані для кращого розуміння даних. Використано plt.grid(True) для додавання сітки на графік і plt.tight_layout() для коректного відображення елементів. Наприкінці використано plt.show() для відображення графіка.



Висновок: У процесі виконання лабораторної роботи було використано бібліотеки pandas і matplotlib для обробки та візуалізації даних. Зокрема, було здійснено зчитування, обробку та аналіз даних за допомогою pandas, включаючи роботу з типами даних, групування, обчислення нових полів та фільтрацію за умовами. Для візуалізації результатів застосовано matplotlib, що дозволило побудувати графіки, які наочно демонструють обсяг продажів і доходи. Таким чином, завдання дозволило набратися досвіду роботи з даними та їх представленням у вигляді графіків.