Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Лабораторна робота №4

Аналіз даних з використанням мови Python

Тема: Візуалізація даних за допомогою matplotlib та Seaborn

Варіант: 1

Виконав

студент групи ІП-11:

Панченко С. В.

Перевірив:

Тимофєєва Ю. С

Київ 2023

ЗМІСТ

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Ознайомитись з основними діаграмами та графіками, що використовуються при аналізі даних. Навчитись будувати їх за допомогою бібліотек matplotlib та Seaborn.

2 ЗАВДАННЯ

Варіант 1.

Файл diamonds.csv.

1. Побудувати стовпчикові діаграми, на яких відобразити а) кількість діамантів кожного з класів якості; б) максимальну ціну діамантів кожного класу якості; в) середню глибину діамантів різного класу якості з різною якістю кольору.
2. Побудувати гістограму глибини діамантів у відсотках (depth), загальну і для кожного класу якості.
3. Побудувати діаграму розмаху параметру table (загальну і в залежності від якості кольору), визначити чи присутні викиди.
4. За допомогою діаграм розсіювання зробити висновки щодо залежності між а) довжиною і шириною; б) глибиною у % і глибиною у мм. Порахувати коефіцієнт кореляції за допомогою відповідних функцій.

3 ВИКОНАННЯ

3.1 Побудувати стовпчикові діаграми, на яких відобразити 1. кількість діамантів кожного з класів якості; 2. максимальну ціну діамантів кожного класу якості; 3. середню глибину діамантів різного класу якості з різною якістю кольору.

Для початку імпортуємо модулі pandas, numpy, seaborn та matplotlib. Завантажимо датафрейм та виведемо його вміст. Видалимо колонку індексів, оскільки вона за замовчуванням створюється при конструюванні датафрейму.

In [68]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv('data/diamonds.csv')
df.drop(['Unnamed: 0'], inplace=True, axis=1)
df
```

Out[68]:

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	'Ideal'	'E'	'SI2'	61.5	55.0	326	3.95	3.98	2.43
1	0.21	'Premium'	'E'	'SI1'	59.8	61.0	326	3.89	3.84	2.31
2	0.23	'Good'	'E'	'VS1'	56.9	65.0	327	4.05	4.07	2.31
3	0.29	'Premium'	'I'	'VS2'	62.4	58.0	334	4.20	4.23	2.63
4	0.31	'Good'	'J'	'SI2'	63.3	58.0	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...	...
1509	0.81	'Very Good'	'G'	'VS2'	63.1	58.0	2994	5.88	5.84	3.70
1510	1.24	'Premium'	'J'	'I1'	61.9	55.0	2994	6.92	6.85	4.26
1511	0.81	'Premium'	'G'	'VS2'	62.0	58.0	2994	5.95	5.92	3.68
1512	0.81	'Premium'	'D'	'SI2'	61.7	58.0	2994	5.97	5.93	3.67
1513	0.73	'Ideal'	'D'	'SI1'	61.4	56.0	2995	5.78	5.82	3.56

1514 rows × 10 columns

Рисунок 3.1 - Імпортування модулів та завантаження датасету

Побудуємо діаграму кількості діамантів кожного з класів якості. Для цього використаємо метод pd.Series.hist.

```
In [69]: df.cut.hist()
```

```
Out[69]: <AxesSubplot: >
```

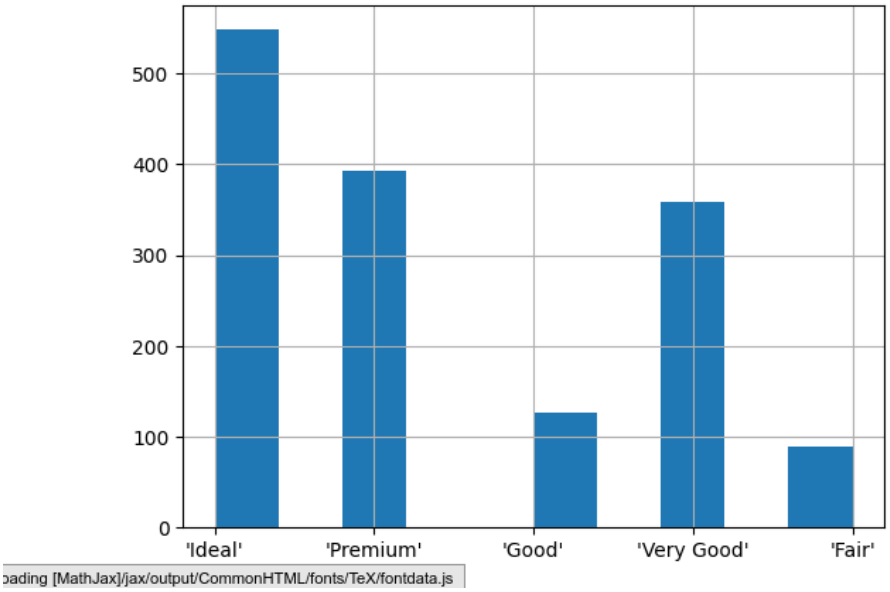

Рисунок 3.2 - Діаграма кількості діамантів кожного з класів якості

Побудуємо діаграму максимальних ціни діамантів кожного класу якості. Зробимо це за допомогою функції sns.barplot, де в аргументи передамо класи якості, у значення їхню максимальну ціну. Максимальну ціну знайдемо, згрупувавши датафрейм та застосувавши функцію max.

```
In [70]: dd = df.groupby(by='cut').agg(max)
         dd
```

Out[70]:

cut	carat	color	clarity	depth	table	price	x	y	z
'Fair'	1.50	'J'	'VVS2'	69.5	70.0	2993	7.26	7.09	4.70
'Good'	1.04	'J'	'VVS2'	65.2	65.0	2990	6.67	6.60	4.03
'Ideal'	1.02	'J'	'VVS2'	63.0	60.0	2995	6.53	6.50	3.99
'Premium'	1.27	'J'	'VVS2'	63.0	62.0	2994	7.12	7.05	4.26
'Very Good'	1.24	'J'	'VVS2'	64.5	64.0	2994	6.85	6.92	4.26

Рисунок 3.3 - Знаходження максимальних цін по кожному класу

Тепер за цією інформацією нарешті побудуємо діаграму. Зробимо обмеження для осі ординат за допомогою методу set_ylim, передавши в нього кортеж з локального мінімально та максимального значення глобальних максимальних значень по класах.

```
In [71]: ax = sns.barplot(x=dd.index, y=dd.price)
         ax.set_ylim((dd.price.min()-3, dd.price.max()+3))

Out[71]: (2987.0, 2998.0)
```

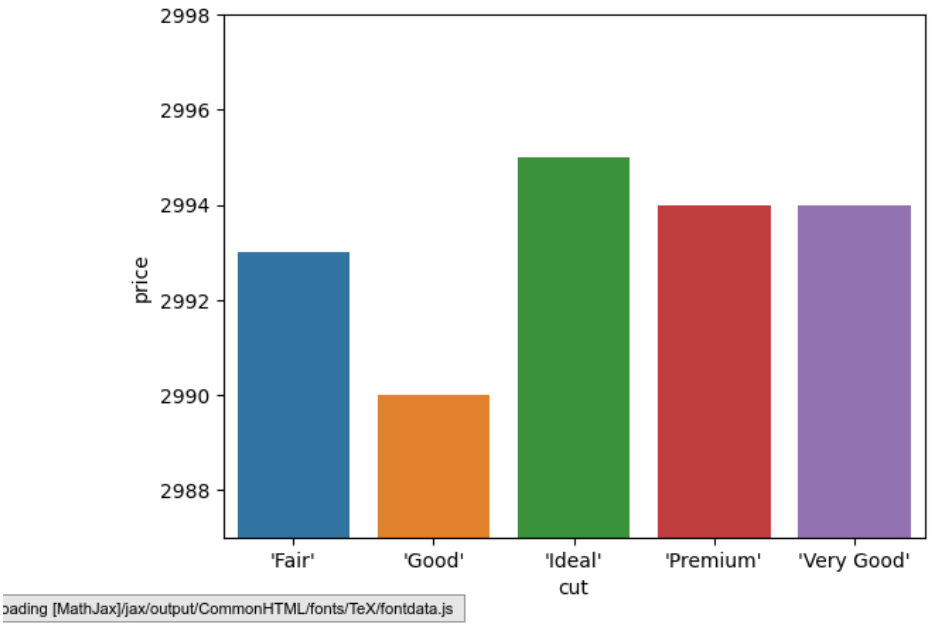

Рисунок 3.4 - Діаграма максимальних ціни діамантів кожного класу якості

Побудуємо діаграму середньої глибини діамантів різного класу якості з різною якістю кольору. Також застосуємо функцію sns.barplot, але передамо в параметр за замовчуванням hue - колір.

```
In [72]: dd = df.groupby(by=['cut', 'color']).agg(np.mean)
         dd_index = dd.index.to_frame()
         ax = sns.barplot(x=dd_index.iloc[:, 0], y=dd.z, hue=dd_index.iloc[:, 1])
         ax.set_ylim((dd.z.min()-0.1, dd.z.max()+0.1))
```

```
/tmp/ipykernel_12148/1613217633.py:1: FutureWarning: The default value of numeric_only in
DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will default to Fal
se. Either specify numeric_only or select only columns which should be valid for the funct
ion.
  dd = df.groupby(by=['cut', 'color']).agg(np.mean)
```

```
Out[72]: (3.0557142857142856, 4.187777777777777)
```

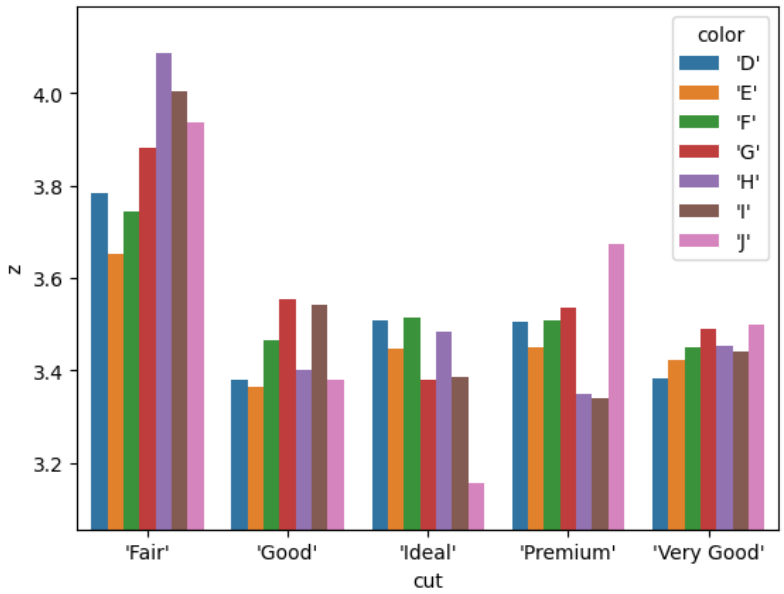

Рисунок 3.5 - Діаграма середньої глибини діамантів різного класу якості з різною якістю кольору

3.2 Побудувати гістограму глибини діамантів у відсотках (depth), загальну і для кожного класу якості.

Спочатку подивимося на загальний розподіл, використавши функцію sns.histplot.

```
In [73]: sns.histplot(df['depth'])

Out[73]: <AxesSubplot: xlabel='depth', ylabel='Count'>
```

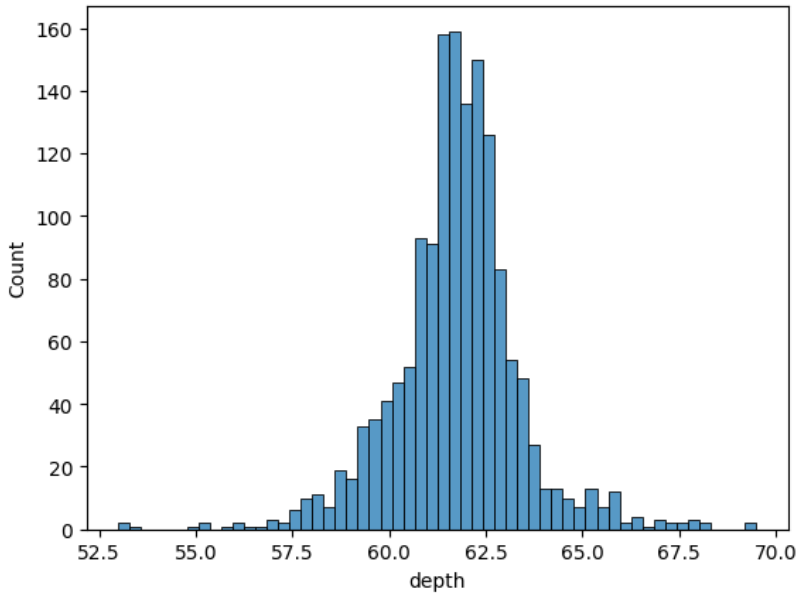

Рисунок 3.6 - Діаграма загального розподілу глибини діамантів у відсотках

Для побудови гістограми застосуємо об'єкт класу sns.FacetGrid, у конструктор якого передаємо необхідні стовпчики. Після цього застосуємо метод map, передавши в нього функцію гістограми sns.histplot та необхідну колонку.

```
In [74]: fig = sns.FacetGrid(df[['depth', 'cut']], col='cut',
                             col_wrap=3, height=3)
         fig.map(sns.histplot, 'depth')

Out[74]: <seaborn.axisgrid.FacetGrid at 0x7f0350f019f0>
```

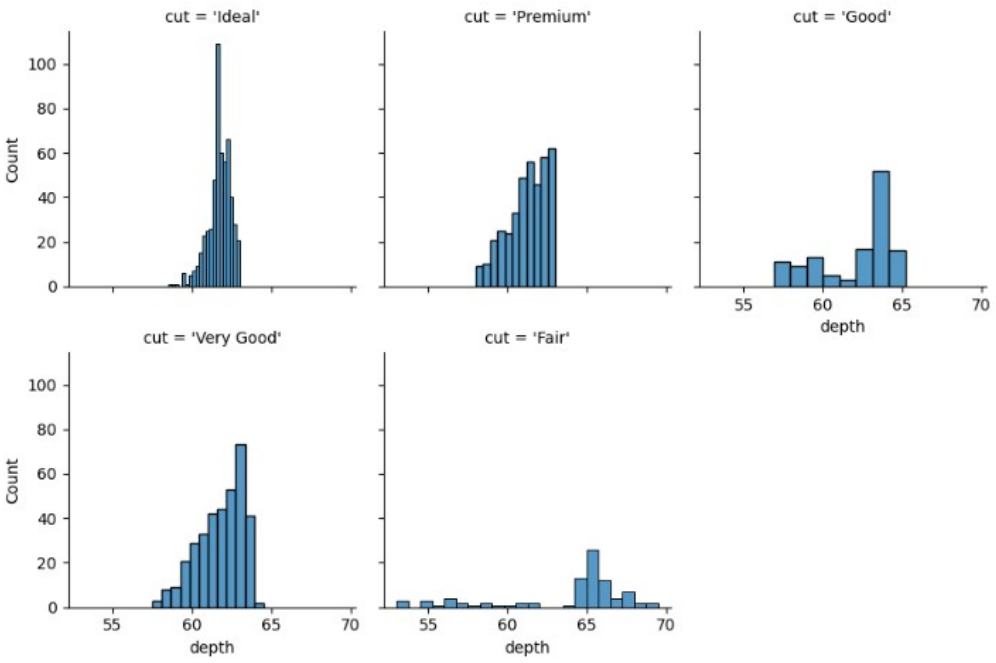

Рисунок 3.7 - Діаграма розподілу глибини діамантів у відсотках для кожного класу якості

3.3 Побудувати діаграму розмаху параметру table (загальну і в залежності від якості кольору), визначити чи присутні викиди.

Побудуємо діаграму розмаху, застосувавши функцію sns.boxplot. Як бачимо, існують викиди, що не попадають у діапазон. Детальніше їх розглянемо у наступному параграфі.

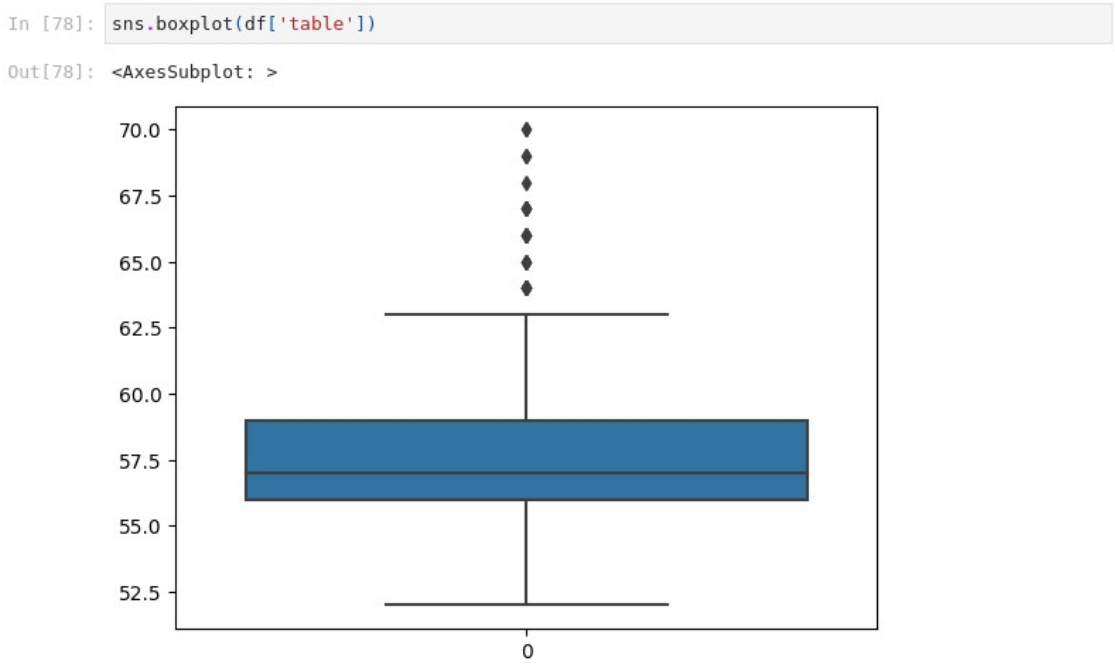

Рисунок 3.8 - Діаграма розмаху параметру table

Побудуємо діаграму розмаху, застосувавши функцію sns.boxplot. Як ми побачимо, що для 'E', 'I', 'H', 'F', 'G', 'D' існують значення, що попадають поза межі T-shaped whiskers, тобто вони не попадають у межу, яка більша у 1.5 рази інтерквантирального діапазону.

```
In [77]: sns.boxplot(x=df['table'], y=df['color'])

Out[77]: <AxesSubplot: xlabel='table', ylabel='color'>
```

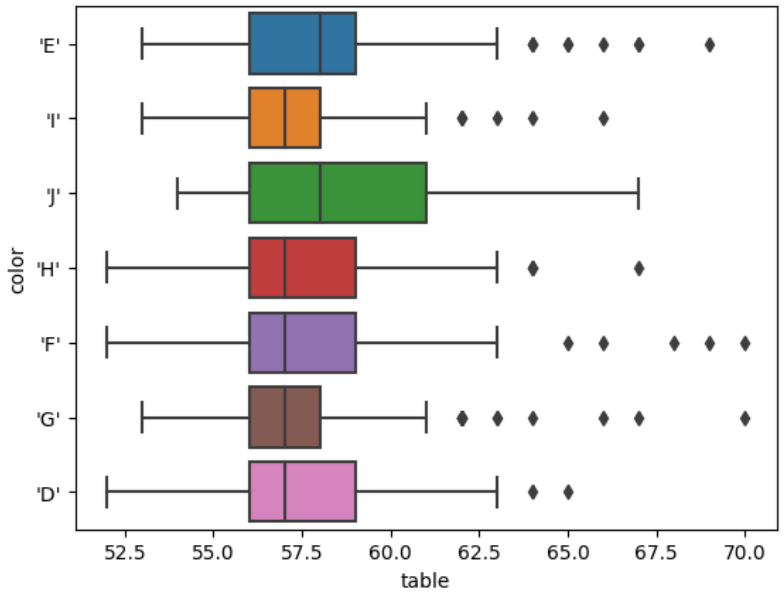

Рисунок 3.9 - Діаграма розмаху параметру table від якості кольору

3.4 За допомогою діаграм розсіювання зробити висновки щодо залежності між 1. довжиною і шириною; 2. глибиною у % і глибиною у мм. Порахувати коефіцієнт кореляції за допомогою відповідних функцій.

Побудуємо діаграму розсіювання за допомогою функції sns.scatterplot. Для початку зобразимо залежність довжини від ширини. Бачимо, що проглядається чітка лінійна залежність між даними величинами.

```
In [80]: sns.scatterplot(data=df, x='x', y='y')
Out[80]: <AxesSubplot: xlabel='x', ylabel='y'>
```

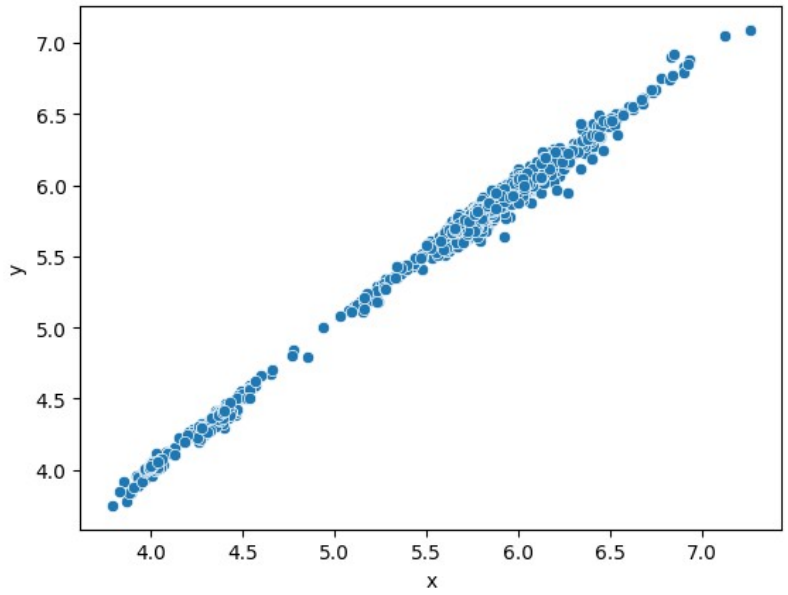

Рисунок 3.10 - Діаграма розсіювання довжини від ширини

Зобразимо діаграму розсіювання між глибиною у % та глибиною у мм. Зараз складно зробити висновок про залежність величин, тому поглянемо на матрицю кореляцій.

```
In [81]: sns.scatterplot(data=df, x='depth', y='z')
Out[81]: <AxesSubplot: xlabel='depth', ylabel='z'>
```

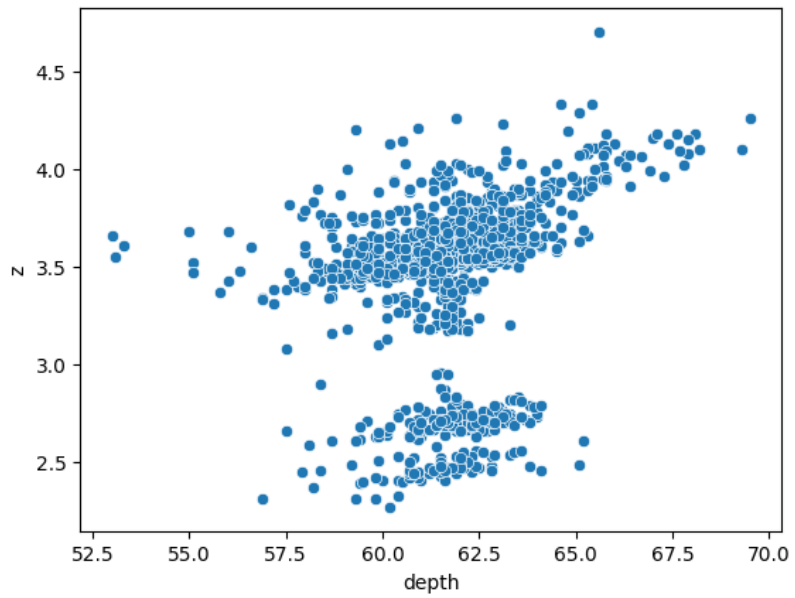

Рисунок 3.11 - Діаграма розсіювання між глибиною у % та глибиною у мм

Побудуємо діаграму кореляцій величин між собою за допомогою методу pandas.DataFrame.corr, результат якого передамо до функції sns.heatmap. Побачимо,що кореляція між довжиною та шириною дорівнює одиниці. Однак залежність абсолютної глибини до відносної - 0.23, що свідчить про те, що ці величини мають зв'язок. Низька кореляція пояснюється тим, що глибина у відсотках залежить також від інших глибин діамантів, у той час як абсолютна глибина показує конкретне число.

Рисунок 3.12 - Матриця кореляцій

4 ВИСНОВОК

Під час виконання даної лабораторної роботи я ознайомився з основними діаграмами та графіками, що використовуються при аналізі даних, навчився будувати їх за допомогою бібліотек matplotlib та Seaborn.

У першому завданні побудував стовпчасті діаграми за допомогою метода pandas.DataFrame.hist та функції sns.barplot.

У другому завданні показав гістограму розсіювання глибини діамантів у відсотках за допомогою функції sns.histplot.

У третьому завданні показав діаграми розмаху за допомогою функції sns.boxplot та побачив, що деякі значення є викидами.

У четвертому завданні побудував діаграми розсіювання за допомогою функції sns.scatterplot та побачив, що існує чітка лінійна залежність між довжиною та шириною і її коефіцієнт кореляції дорівнює одиниці; також існує кореляція у 0.23 між абсолютною та відносною глибиною, що пояснюється тим що відносна залежить не тільки від даного каменя, а й глибин інших.