1. Load the dataset into a pandas DataFrame (Book1.csv)
2. View the first few rows of the dataset
3. Check the dimensions of the dataset
4. Check the column names
5. Check the data types of each column
6. Get summary statistics of numerical columns
7. Get summary statistics of numerical columns with 0.58 and 0.78 percentiles
8. Get summary statistics of object columns
9. information on all columns
10. Check for missing values
11. Fill in missing values by the mean of columns. (temporary)
12. Drop rows with any missing values
13. Set index as roll number in the data frame
14. Removing duplicates if duplicates
15. Removing duplicates if duplicates in Maths and Phy
16. find the number of students who have greater than 20 marks in maths, greater than 20 marks in Phy, and greater than 18 marks in Chem and print the data frame of it
17. Which grade level has the highest number of students scoring in each range of 'Phy' scores?
18. How do the scores in "Phy", "Chem", "Maths", "Eng", and "Com" vary across different grade levels? Writing appropriate code for visualization
19. Which pairs of columns have the highest positive correlation? Also, give the correlation
20. How many students have greater than 80 percent?
21. How many students have greater than 100 total and also get 20 or above in maths and Phy?
22. Which subject(s) have the highest median score based on the position of the box?
23. In which areas does the filled area plot show the "Phy" scores being higher than the "Eng" scores?
24. Are there any specific patterns or trends in the distribution of the data points between maths and total?

1	``` import re text = "The quick brown fox jumps over the lazy dog." result = re.findall(r'\w{4}', text) result ```	
2	``` import re text = "The code is ABC123XYZ and XYZ789." pattern = r'[A-Z]{3}\d{3}[A-Z]{3}' result = re.findall(pattern, text) result ```	
3	``` import re text = "The password is P@ssw0rd." pattern = r'[A-Za-z0-9]+' result = re.search(pattern, text) print(result.group()) ```	
4	``` import re text = "Hello, how are you?" pattern = r'\w{3}\W+[^\s]' result = re.sub(pattern, "###", text) (result) ```	
5	``` import re text = "The quick brown fox jumps over the lazy dog." pattern = r'\s[a-z]+\s' result = re.split(pattern, text) result ```	

| 6 | ```
import re
text = "The_quick_brown@fox*jumps#over$the^lazy&dog."
pattern = r'[a-zA-Z]+'
result = re.split(pattern, text)
len(result[0])
``` | |
| 7 | ```
import pandas as pd
import numpy as np
df=pd.DataFrame([[0,1,2,np.nan,5],[2,0,1,5,np.nan],[5,0,1,np.nan,5]])
df=df.drop_duplicates(subset=[1,2])
df=df.drop_duplicates(subset=[4])
df.dropna(thresh=2,axis=1,inplace=True)
df.shape
``` | |
| 8 | ```
import pandas as pd
import numpy as np
df = pd.DataFrame([[0, 1, 2, np.nan, 5], [2, 0, 1, 5, np.nan], [5, 0, 1,
np.nan, 5]])
df = df.fillna(method="bfill")
df= df.fillna(5.0)
df=df.drop_duplicates(subset=[3,4],keep="last")
print(df.loc[0])
``` | |
| 9 | ```
import pandas as pd
import numpy as np
df = pd.DataFrame({"A":[0, 1, 2, np.nan, 5],"B": [2, 0, 1, 5,
np.nan],"C": [5, 0, 1, np.nan, 5]})
df=df.drop_duplicates(subset=["C"])
df=df.fillna(1.0)
df=df.drop_duplicates(subset=["B","C"])
print(df.shape)
``` | |
| 10 | ```
import pandas as pd
import numpy as np
df=pd.DataFrame([[0,1,2,np.nan,5],[2,0,1,5,np.nan],[5,0,1,np.nan,5]])
print(df.iloc[2,4])
``` | |