Data Analysis and Visualisation in Python & Pandas

In this project, we used Pandas to extract and analyse specific data from the student.csv dataset. Our first step was to read the data from the CSV file into a Pandas DataFrame using Colab notebook. To quickly explore the dataset, we printed the first 5 rows, providing a snapshot of the data. Additionally, we used the code to get the information and summary statistics for the dataframe, which helped us better understand the data's structure and attributes.

Throughout the project, we utilized a wide range of Python and Pandas functionalities to manipulate and analyse data efficiently. These included basic techniques such as indexing and slicing, which allowed us to access and extract specific portions of data. We also employed aggregation and grouping methods to summarise and analyse data based on specific criteria. Additionally, we leveraged more advanced operations, such as pivot tables, which enabled us to reorganise and summarise data in a more insightful and comprehensive manner, facilitating deeper analysis and understanding.

We also explored important positional arguments to define the order of parameters in functions, which helped streamline our functions. Furthermore, we utilised algorithms and arithmetic operators for data manipulation and analysis, along with shortcut operators for efficient coding. Conditional statements allowed us to perform operations based on specific conditions, such as sorting in descending order, average marks by gender and filtering students with more than 60 marks.

To assist with the visual analysis of the data, we employed several visualisation techniques, including histograms, bar plots, scatter plots, and box plots. These visualisations were instrumental in uncovering patterns, distributions, and relationships within the data, allowing us to gain deeper insights.

Please download the student.csv dataset here.

Exercise 1: Loading and Exploring the Data

1. Question: "Write the code to read a CSV file into a Pandas DataFrame."
2. Question: "Write the code to display the first 5 rows of the DataFrame."
3. Question: "Write the code to get the information about the DataFrame."
4. Question: "Write the code to get summary statistics for the DataFrame."

1. #code to read a CSV file into a Pandas DataFrame

```
import pandas as pd
dataframe = pd.read_csv('student.csv')
```

2. #code to display the first 5 rows of the DataFrame.

```
dataframe.head()
```

	id		name	class	mark	gender
0	1		John Deo	Four	75	female
1	2		Max Ruin	Three	85	male
2	3		Arnold	Three	55	male
3	4		Krish Star	Four	60	female
4	5		John Mike	Four	60	female

3. #code to get the information about the DataFrame.

```
dataframe.info()
```

```
#   Column Non-Null Count Dtype
 0   id      35 non-null     int64
 1   name    34 non-null     object
 2   class   34 non-null     object
 3   mark    35 non-null     int64
 4   gender  33 non-null     object
```

4.#code to get summary statistics for the DataFrame.

```
dataframe.describe()
```

	id	mark
count	35.000000	35.000000
mean	18.000000	74.657143
std	10.246951	16.401117
min	1.000000	18.000000
25%	9.500000	62.500000
50%	18.000000	79.000000
75%	26.500000	88.000000
max	35.000000	96.000000

Exercise 2: Indexing and Slicing

1. Question: "Write the code to select the 'name' column."
2. Question: "Write the code to select the 'name' and 'mark' columns."
3. Question: "Write the code to select the first 3 rows."
4. Question: "Write the code to select all rows where the 'class' is 'Four'."

1. #code to select the 'name' column
```
dataframe['name']
```

2. #code to select the 'name' and 'mark' columns
```
dataframe[['name','mark']].head(5)
```

	name	mark
0	John Deo	75
1	Max Ruin	85
2	Arnold	55
3	Krish Star	60
4	John Mike	60

3. #code to display the first 3 rows
```
dataframe.head(3)
```

	id	name	class	mark	gender
0	1	John Deo	Four	75	female
1	2	Max Ruin	Three	85	male
2	3	Arnold	Three	55	male

4. #code to select all rows where the 'class' is 'Four'
```
dataframe[dataframe['class']=='Four']
```

	id	name	class	mark	gender
0	1	John Deo	Four	75	female
3	4	Krish Star	Four	60	female
4	5	John Mike	Four	60	female
5	6	Alex John	Four	55	male
9	10	Big John	Four	55	female
15	16	Gimmy	Four	88	male
20	21	Babby John	Four	69	female
30	31	Marry Toeey	Four	88	male

Exercise 3: Data Manipulation

1. Question: "Write the code to add a new column 'passed' that indicates whether the student passed (mark >= 60)."
2. Question: "Write the code to rename the 'mark' column to 'score'."
3. Question: "Write the code to drop the 'passed' column."

1. #code to add a new column 'passed' that indicates whether the student passed (mark >= 60).

```
dataframe['passed'] = dataframe['mark'] >= 60
dataframe
```

	id	name	class	mark	gender	passed
0	1	John Deo	Four	75	female	True
1	2	Max Ruin	Three	85	male	True
2	3	Arnold	Three	55	male	False
3	4	Krish Star	Four	60	female	True
4	5	John Mike	Four	60	female	True
5	6	Alex John	Four	55	male	False
6	7	My John Rob	Fifth	78	male	True
7	8	Asruid	Five	85	male	True
8	9	Tes Qry	Six	78	NaN	True
9	10	Big John	Four	55	female	False
10	11	Ronald	Six	89	female	True

2. #code to rename the 'mark' column to 'score'

```
dataframe.rename(columns={'mark':'score'}, inplace=True)
dataframe
```

	id	name	class	score	gender
0	1	John Deo	Four	75	female
1	2	Max Ruin	Three	85	male
2	3	Arnold	Three	55	male

3. #code to drop the 'passed' column

```
dataframe.drop('passed', axis=1, inplace=True)
dataframe
```

	id	name	class	mark	gender
0	1	John Deo	Four	75	female
1	2	Max Ruin	Three	85	male
2	3	Arnold	Three	55	male

Exercise 4: Aggregation and Grouping

1. Question: "Write the code to group the DataFrame by the 'class' column and calculate the mean 'mark' for each group."
2. Question: "Write the code to count the number of students in each class."
3. Question: "Write the code to calculate the average mark for each gender."

1. #code to group the DataFrame by the 'class' column and calculate the mean 'mark' for each group

```
dataframe.groupby('class')['mark'].mean()
```

class	mark
Eight	79.000000
Fifth	78.000000
Five	80.000000
Four	68.750000
Nine	41.500000
Seven	77.600000
Six	82.571429
Three	73.666667

2. #code to count the number of students in each class

```
dataframe['class'].value_counts()
```

class	count
Seven	10
Four	8
Six	7
Three	3
Five	2
Nine	2
Fifth	1
Eight	1

3. #code to calculate the average mark for each gender

```
dataframe.groupby('gender')['mark'].mean()
```

gender	mark
female	77.312500
male	71.588235

Exercise 5: Advanced Operations

1. Question: "Write the code to create a pivot table with 'class' as rows, 'gender' as columns, and 'mark' as values."
2. Question: "Write the code to create a new column 'grade' where marks >= 85 are 'A', 70-84 are 'B', 60-69 are 'C', and below 60 are 'D'."
3. Question: "Write the code to sort the DataFrame by 'mark' in descending order."

1. #code to create a pivot table with 'class' as rows, 'gender' as columns, and 'mark' as values
```
dataframe.pivot_table(index='class', columns='gender',values='mark')
```

2. #code to create a new column 'grade' where marks >= 85 are 'A', 70-84 are 'B', 60-69 are 'C', and below 60 are 'D'
```
dataframe['grade'] = pd.cut(dataframe['mark'], bins=[0,59,69,84,100], labels=['D','C','B','A'])
dataframe
```

	id	name	class	mark	gender	grade
0	1	John Deo	Four	75	female	B
1	2	Max Ruin	Three	85	male	A
2	3	Arnold	Three	55	male	D
3	4	Krish Star	Four	60	female	C
4	5	John Mike	Four	60	female	C
5	6	Alex John	Four	55	male	D

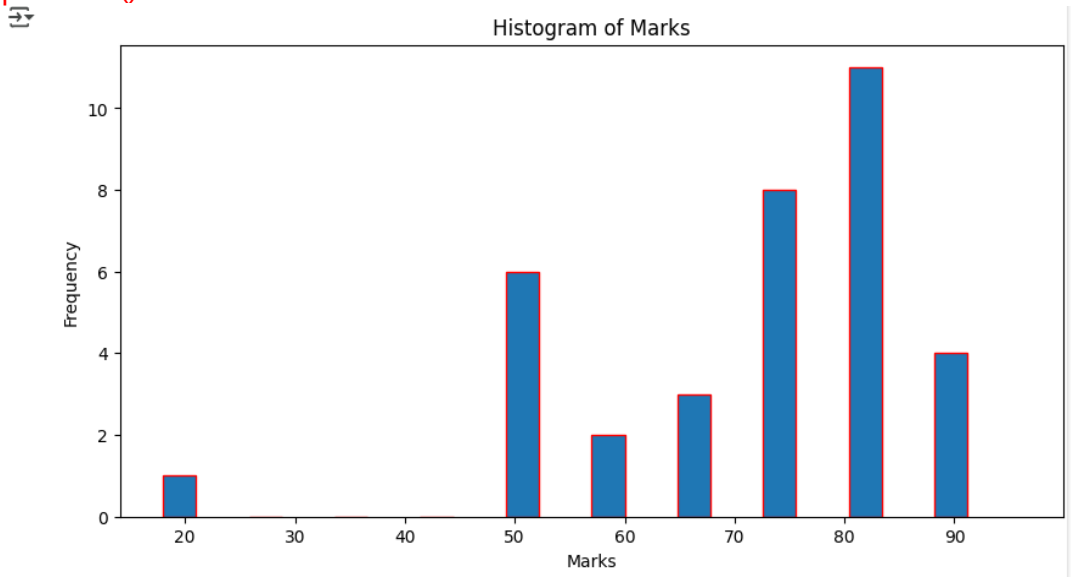
3. #code to sort the DataFrame by 'mark' in descending order
```
dataframe.sort_values(by='mark', ascending=False)
```

	id	name	class	mark	gender	grade
32	33	Kenn Rein	Six	96	female	A
11	12	Recky	Six	94	female	A
31	32	Binn Rott	Seven	90	female	A
10	11	Ronald	Six	89	female	A
24	25	Giff Tow	Seven	88	male	A

Exercise 6: Exporting and visualising the data

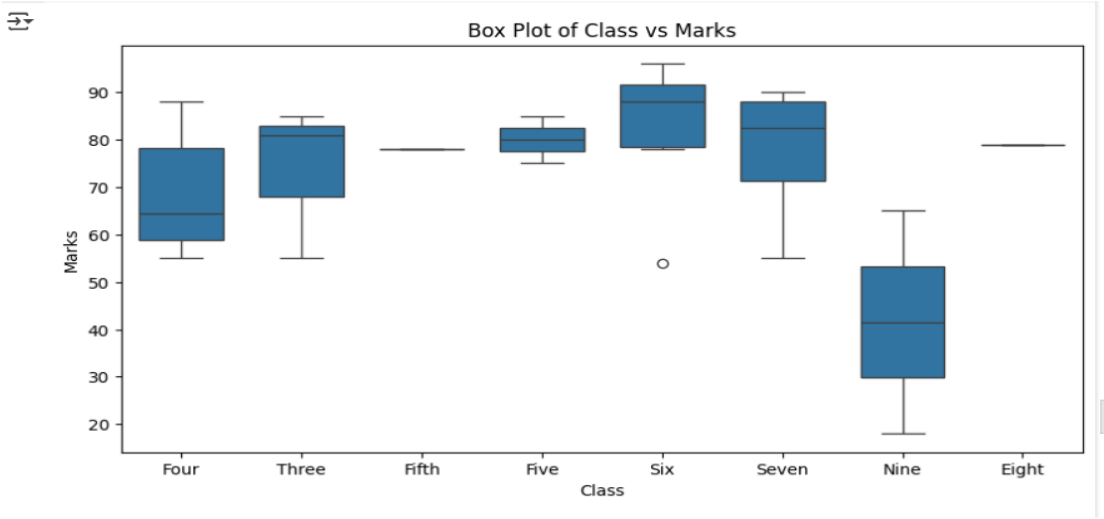
```
#code to save the DataFrame with the new 'grade' column to a new CSV file.
dataframe.to_csv('student_with_grade.csv', index=False)

#Histogram plot
plt.figure(figsize=(10,5))
plt.hist(df['mark'], bins=10,  edgecolor='red',width=3)
plt.title('Histogram of Marks')
plt.xlabel('Marks')
plt.ylabel('Frequency')
plt.show()
```

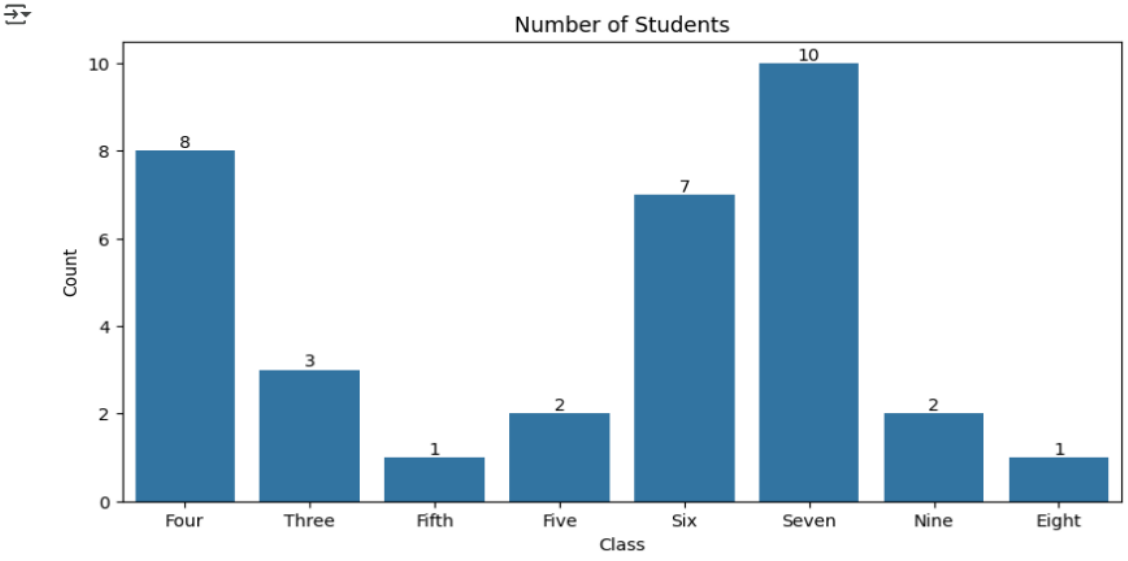



```
#Scatter plot
df['gender']=df['gender'].astype(str)
plt.figure(figsize=(10,8))
plt.scatter(df['gender'], df['mark'], alpha=0.3, color='green', marker='x', s=50)
plt.title('Scatter Plot of Gender vs Marks')
plt.xlabel('Gender')
plt.ylabel('Marks')
plt.show()
```

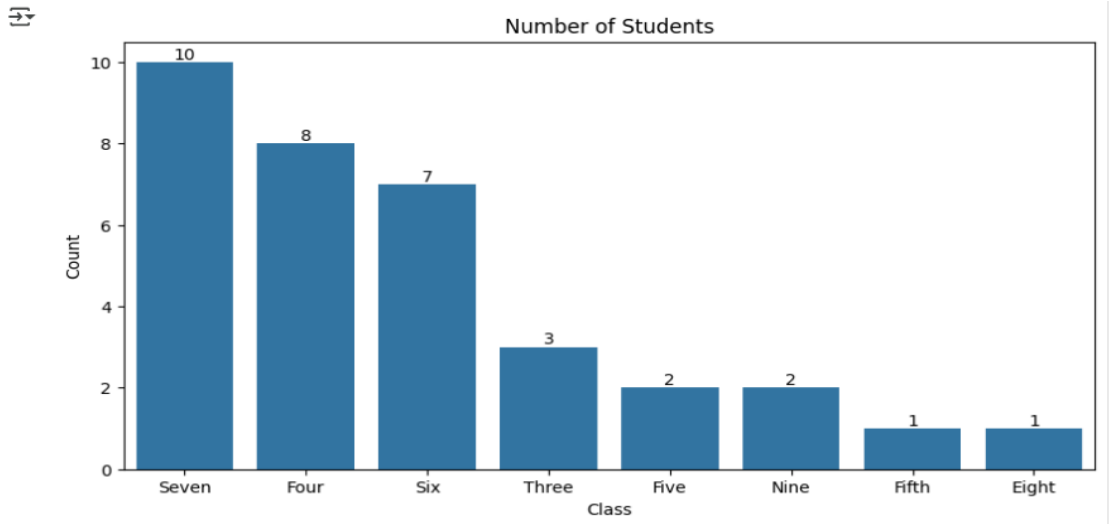
```
#Box Plot
plt.figure(figsize=(10,5))
sns.boxplot(x='class', y='mark', data=df, width=0.7)
plt.title('Box Plot of Class vs Marks')
plt.xlabel('Class')
plt.ylabel('Marks')
plt.show()
```




```
#Count Plot
plt.figure(figsize=(10,5))
std=sns.countplot(x='class',data=df)
std.bar_label(std.containers[0])
plt.title('Number of Students')
plt.xlabel('Class')
plt.ylabel('Count')
plt.show()
```



```
#To sort in descending order
plt.figure(figsize=(10,5))
std=sns.countplot(x='class',data=df, order=df['class'].value_counts().index)
std.bar_label(std.containers[0])
plt.title('Number of Students')
plt.xlabel('Class')
plt.ylabel('Count')
plt.show()
```




```
#Average Marks by Gender
plt.figure(figsize=(10,6))
avg_marks_gender=df.groupby('gender')['mark'].mean().reset_index()
sns.barplot(x='gender',y='mark',data=avg_marks_gender)
plt.title('Average Marks by Gender')
plt.xlabel('Gender')
plt.ylabel('Average Marks')
plt.show()
```