```
#PROJECT ON PANDAS FUNCTONS
```

```
import pandas as pd

data = {
    "StudentID": range(301, 321),
    "Name": [
        "Alice Brown", "Ben Johnson", "Clara Davis", "Daniel Lee", "Eva Wilson",
        "Frank Miller", "Grace Taylor", "Henry Clark", "Irene White", "Jack Lewis",
        "Karen Young", "Liam Scott", "Mia Adams", "Noah Carter", "Olivia Green",
        "Paul Walker", "Quinn Rivera", "Ryan Hall", "Sophia King", "Thomas Allen"
    ],
    "Major": [
        "Computer Science", "Mechanical Eng.", "Psychology", "Business Mgmt.", "Biology",
        "Computer Science", "English Literature", "Mathematics", "Chemistry", "Business Mgmt.",
        "Biology", "Computer Science", "Mathematics", "Mechanical Eng.", "Psychology",
        "Business Mgmt.", "English Literature", "Biology", "Mathematics", "Computer Science"
    ],
    "GPA": [
        3.9, 3.3, 3.5, 3.2, 3.8, 2.9, 3.6, 3.7, 3.4, 3.1,
        3.9, 3.6, 3.2, 3.0, 3.8, 3.5, 3.7, 2.8, 3.9, 3.4
    ],
    "State": [
        "California", "Texas", "Florida", "Illinois", "New York", "California", "Ohio",
        "Texas", "Washington", "Florida", "California", "New York", "Texas", "Illinois",
        "Florida", "Ohio", "Texas", "California", "Washington", "Illinois"
    ]
}

df = pd.DataFrame(data)
```

```
df
```

```
    StudentID          Name               Major  GPA       State
0         301   Alice Brown    Computer Science  3.9  California
1         302   Ben Johnson     Mechanical Eng.  3.3       Texas
2         303   Clara Davis          Psychology  3.5     Florida
3         304    Daniel Lee      Business Mgmt.  3.2    Illinois
4         305    Eva Wilson             Biology  3.8    New York
5         306  Frank Miller    Computer Science  2.9  California
6         307  Grace Taylor  English Literature  3.6        Ohio
7         308   Henry Clark         Mathematics  3.7       Texas
8         309   Irene White           Chemistry  3.4  Washington
9         310    Jack Lewis      Business Mgmt.  3.1     Florida
10        311   Karen Young             Biology  3.9  California
11        312    Liam Scott    Computer Science  3.6    New York
12        313     Mia Adams         Mathematics  3.2       Texas
13        314   Noah Carter     Mechanical Eng.  3.0    Illinois
14        315  Olivia Green          Psychology  3.8     Florida
15        316   Paul Walker      Business Mgmt.  3.5        Ohio
16        317  Quinn Rivera  English Literature  3.7       Texas
17        318     Ryan Hall             Biology  2.8  California
18        319   Sophia King         Mathematics  3.9  Washington
19        320  Thomas Allen    Computer Science  3.4    Illinois
```

Next steps: Generate code with df New interactive sheet

```
#Display the first 5 rows of the dataset to get an overview of the data

df.head(5)
```

	StudentID	Name	Major	GPA	State
0	301	Alice Brown	Computer Science	3.9	California
1	302	Ben Johnson	Mechanical Eng.	3.3	Texas
2	303	Clara Davis	Psychology	3.5	Florida
3	304	Daniel Lee	Business Mgmt.	3.2	Illinois
4	305	Eva Wilson	Biology	3.8	New York

Next steps: Generate code with df New interactive sheet

```
#What happens if you run df.tail(10)? How many rows are shown?

df.tail(10)
```

	StudentID	Name	Major	GPA	State
10	311	Karen Young	Biology	3.9	California
11	312	Liam Scott	Computer Science	3.6	New York
12	313	Mia Adams	Mathematics	3.2	Texas
13	314	Noah Carter	Mechanical Eng.	3.0	Illinois
14	315	Olivia Green	Psychology	3.8	Florida
15	316	Paul Walker	Business Mgmt.	3.5	Ohio
16	317	Quinn Rivera	English Literature	3.7	Texas
17	318	Ryan Hall	Biology	2.8	California
18	319	Sophia King	Mathematics	3.9	Washington
19	320	Thomas Allen	Computer Science	3.4	Illinois

```
#get a summary specifically for GPA

df["GPA"].describe()
```

	GPA
count	20.000000
mean	3.460000
std	0.342437
min	2.800000
25%	3.200000
50%	3.500000
75%	3.725000
max	3.900000

dtype: float64

```
#check the total number of rows and columns in the dataset

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   StudentID  20 non-null     int64  
 1   Name       20 non-null     object 
 2   Major      20 non-null     object 
 3   GPA        20 non-null     float64
 4   State      20 non-null     object 
dtypes: float64(1), int64(1), object(3)
memory usage: 932.0+ bytes
```

```
#Verify that your dataset has 20 students and 5 columns

df.shape
```

```
(20, 5)
```

```
#Selection & Indexing
#How do you select the Name and GPA columns from the DataFrame?

df[['Name', 'GPA']]
```

	Name	GPA
0	Alice Brown	3.9
1	Ben Johnson	3.3
2	Clara Davis	3.5
3	Daniel Lee	3.2
4	Eva Wilson	3.8
5	Frank Miller	2.9
6	Grace Taylor	3.6
7	Henry Clark	3.7
8	Irene White	3.4
9	Jack Lewis	3.1
10	Karen Young	3.9
11	Liam Scott	3.6
12	Mia Adams	3.2
13	Noah Carter	3.0
14	Olivia Green	3.8
15	Paul Walker	3.5
16	Quinn Rivera	3.7
17	Ryan Hall	2.8
18	Sophia King	3.9
19	Thomas Allen	3.4

```
#How do you select the row for the student with StudentID 308?


df[df['StudentID'] == 308]
```

	StudentID	Name	Major	GPA	State
7	308	Henry Clark	Mathematics	3.7	Texas

```
#How do you select the first 3 rows and the first 2 columns?

df.iloc[:3, :2]
```

	StudentID	Name
0	301	Alice Brown
1	302	Ben Johnson
2	303	Clara Davis

```
#Filtering & Conditional Selection
#How do you find all students with a GPA greater than 3.5?

df[df['GPA'] > 3.5]
```

	StudentID	Name	Major	GPA	State
0	301	Alice Brown	Computer Science	3.9	California
4	305	Eva Wilson	Biology	3.8	New York
6	307	Grace Taylor	English Literature	3.6	Ohio
7	308	Henry Clark	Mathematics	3.7	Texas
10	311	Karen Young	Biology	3.9	California
11	312	Liam Scott	Computer Science	3.6	New York
14	315	Olivia Green	Psychology	3.8	Florida
16	317	Quinn Rivera	English Literature	3.7	Texas
18	319	Sophia King	Mathematics	3.9	Washington

```
#How do you filter students whose Major is Computer Science?

df[df['Major'] == 'Computer Science']
```

	StudentID	Name	Major	GPA	State
0	301	Alice Brown	Computer Science	3.9	California
5	306	Frank Miller	Computer Science	2.9	California
11	312	Liam Scott	Computer Science	3.6	New York
19	320	Thomas Allen	Computer Science	3.4	Illinois

```
#How do you select students from California with a GPA above 3.0?

df[(df['State'] == 'California') & (df['GPA'] > 3.0)]
```

	StudentID	Name	Major	GPA	State
0	301	Alice Brown	Computer Science	3.9	California
10	311	Karen Young	Biology	3.9	California

```
#How do you find students whose name starts with “A” or “B”?

df[df['Name'].str.startswith(('A','B'))]
```

	StudentID	Name	Major	GPA	State
0	301	Alice Brown	Computer Science	3.9	California
1	302	Ben Johnson	Mechanical Eng.	3.3	Texas

```
#Sorting
#How do you sort the DataFrame by GPA in descending order?

df.sort_values(by='GPA', ascending=False)
```

	StudentID	Name	Major	GPA	State
0	301	Alice Brown	Computer Science	3.9	California
10	311	Karen Young	Biology	3.9	California
18	319	Sophia King	Mathematics	3.9	Washington
14	315	Olivia Green	Psychology	3.8	Florida
4	305	Eva Wilson	Biology	3.8	New York
7	308	Henry Clark	Mathematics	3.7	Texas
16	317	Quinn Rivera	English Literature	3.7	Texas
11	312	Liam Scott	Computer Science	3.6	New York
6	307	Grace Taylor	English Literature	3.6	Ohio
15	316	Paul Walker	Business Mgmt.	3.5	Ohio
2	303	Clara Davis	Psychology	3.5	Florida
19	320	Thomas Allen	Computer Science	3.4	Illinois
8	309	Irene White	Chemistry	3.4	Washington
1	302	Ben Johnson	Mechanical Eng.	3.3	Texas
3	304	Daniel Lee	Business Mgmt.	3.2	Illinois
12	313	Mia Adams	Mathematics	3.2	Texas
9	310	Jack Lewis	Business Mgmt.	3.1	Florida
13	314	Noah Carter	Mechanical Eng.	3.0	Illinois
5	306	Frank Miller	Computer Science	2.9	California
17	318	Ryan Hall	Biology	2.8	California

```
#How do you sort students first by State and then by GPA?

df.sort_values(by=['State','GPA'])
```

	StudentID	Name	Major	GPA	State
17	318	Ryan Hall	Biology	2.8	California
5	306	Frank Miller	Computer Science	2.9	California
0	301	Alice Brown	Computer Science	3.9	California
10	311	Karen Young	Biology	3.9	California
9	310	Jack Lewis	Business Mgmt.	3.1	Florida
2	303	Clara Davis	Psychology	3.5	Florida
14	315	Olivia Green	Psychology	3.8	Florida
13	314	Noah Carter	Mechanical Eng.	3.0	Illinois
3	304	Daniel Lee	Business Mgmt.	3.2	Illinois
19	320	Thomas Allen	Computer Science	3.4	Illinois
11	312	Liam Scott	Computer Science	3.6	New York
4	305	Eva Wilson	Biology	3.8	New York
15	316	Paul Walker	Business Mgmt.	3.5	Ohio
6	307	Grace Taylor	English Literature	3.6	Ohio
12	313	Mia Adams	Mathematics	3.2	Texas
1	302	Ben Johnson	Mechanical Eng.	3.3	Texas
7	308	Henry Clark	Mathematics	3.7	Texas
16	317	Quinn Rivera	English Literature	3.7	Texas
8	309	Irene White	Chemistry	3.4	Washington
18	319	Sophia King	Mathematics	3.9	Washington

```
#Aggregation & Grouping
#how do you calculate the average GPA for the entire dataset?
```

```
df['GPA'].mean()
```

```
np.float64(3.46)
```

```
#How do you find the highest GPA for each Major?

df.groupby('Major')['GPA'].max()
```

```
                    GPA
           Major
         Biology    3.9
   Business Mgmt.   3.5
       Chemistry    3.4
 Computer Science   3.9
English Literature  3.7
     Mathematics    3.9
 Mechanical Eng.    3.3
      Psychology    3.8

dtype: float64
```

```
#How do you count the number of students in each State?

df['State'].value_counts()
```

```
             count
       State
  California    4
       Texas    4
     Florida    3
    Illinois    3
    New York    2
        Ohio    2
  Washington    2

dtype: int64
```

```
#how do you calculate the average GPA for students grouped by Major?

df.groupby('Major')['GPA'].mean()
```

```
                         GPA
           Major
         Biology    3.500000
   Business Mgmt.   3.266667
       Chemistry    3.400000
 Computer Science   3.450000
English Literature  3.650000
     Mathematics    3.600000
 Mechanical Eng.    3.150000
      Psychology    3.650000

dtype: float64
```

```
#Adding & Modifying Columns
#How do you increase every GPA by 0.1 for all students?
```

```
df['GPA'] = df['GPA'] + 0.1
df
```

```
    StudentID          Name                Major  GPA       State
0         301   Alice Brown     Computer Science  4.0  California
1         302   Ben Johnson     Mechanical Eng.   3.4       Texas
2         303   Clara Davis           Psychology  3.6     Florida
3         304    Daniel Lee      Business Mgmt.   3.3    Illinois
4         305    Eva Wilson              Biology  3.9    New York
5         306  Frank Miller     Computer Science  3.0  California
6         307  Grace Taylor   English Literature  3.7        Ohio
7         308   Henry Clark          Mathematics  3.8       Texas
8         309   Irene White            Chemistry  3.5  Washington
9         310    Jack Lewis      Business Mgmt.   3.2     Florida
10        311   Karen Young              Biology  4.0  California
11        312    Liam Scott     Computer Science  3.7    New York
12        313     Mia Adams          Mathematics  3.3       Texas
13        314   Noah Carter     Mechanical Eng.   3.1    Illinois
14        315  Olivia Green           Psychology  3.9     Florida
15        316   Paul Walker      Business Mgmt.   3.6        Ohio
16        317  Quinn Rivera   English Literature  3.8       Texas
17        318     Ryan Hall              Biology  2.9  California
18        319   Sophia King          Mathematics  4.0  Washington
19        320  Thomas Allen     Computer Science  3.5    Illinois
```

Next steps: Generate code with df New interactive sheet

```
#How do you create a new column Pass that is True if GPA ≥ 3.0 and False otherwise?

df['Pass'] = df['GPA'] >= 3.0
df
```

	StudentID	Name	Major	GPA	State	Pass
0	301	Alice Brown	Computer Science	4.0	California	True
1	302	Ben Johnson	Mechanical Eng.	3.4	Texas	True
2	303	Clara Davis	Psychology	3.6	Florida	True
3	304	Daniel Lee	Business Mgmt.	3.3	Illinois	True
4	305	Eva Wilson	Biology	3.9	New York	True
5	306	Frank Miller	Computer Science	3.0	California	True
6	307	Grace Taylor	English Literature	3.7	Ohio	True
7	308	Henry Clark	Mathematics	3.8	Texas	True
8	309	Irene White	Chemistry	3.5	Washington	True
9	310	Jack Lewis	Business Mgmt.	3.2	Florida	True
10	311	Karen Young	Biology	4.0	California	True
11	312	Liam Scott	Computer Science	3.7	New York	True
12	313	Mia Adams	Mathematics	3.3	Texas	True
13	314	Noah Carter	Mechanical Eng.	3.1	Illinois	True
14	315	Olivia Green	Psychology	3.9	Florida	True
15	316	Paul Walker	Business Mgmt.	3.6	Ohio	True
16	317	Quinn Rivera	English Literature	3.8	Texas	True
17	318	Ryan Hall	Biology	2.9	California	False
18	319	Sophia King	Mathematics	4.0	Washington	True
19	320	Thomas Allen	Computer Science	3.5	Illinois	True

Next steps: Generate code with df New interactive sheet

```
#How do you create a new column Honor that shows "Yes" if GPA ≥ 3.7, otherwise "No"?

df['Honor'] = df['GPA'].apply(lambda x: 'Yes' if x >= 3.7 else 'No')
df
```

	StudentID	Name	Major	GPA	State	Pass	Honor
0	301	Alice Brown	Computer Science	4.0	California	True	Yes
1	302	Ben Johnson	Mechanical Eng.	3.4	Texas	True	No
2	303	Clara Davis	Psychology	3.6	Florida	True	No
3	304	Daniel Lee	Business Mgmt.	3.3	Illinois	True	No
4	305	Eva Wilson	Biology	3.9	New York	True	Yes
5	306	Frank Miller	Computer Science	3.0	California	True	No
6	307	Grace Taylor	English Literature	3.7	Ohio	True	Yes
7	308	Henry Clark	Mathematics	3.8	Texas	True	Yes
8	309	Irene White	Chemistry	3.5	Washington	True	No
9	310	Jack Lewis	Business Mgmt.	3.2	Florida	True	No
10	311	Karen Young	Biology	4.0	California	True	Yes
11	312	Liam Scott	Computer Science	3.7	New York	True	Yes
12	313	Mia Adams	Mathematics	3.3	Texas	True	No
13	314	Noah Carter	Mechanical Eng.	3.1	Illinois	True	No
14	315	Olivia Green	Psychology	3.9	Florida	True	Yes
15	316	Paul Walker	Business Mgmt.	3.6	Ohio	True	No
16	317	Quinn Rivera	English Literature	3.8	Texas	True	Yes
17	318	Ryan Hall	Biology	2.9	California	False	No
18	319	Sophia King	Mathematics	4.0	Washington	True	Yes
19	320	Thomas Allen	Computer Science	3.5	Illinois	True	No

Next steps: Generate code with df New interactive sheet

```
#String Operations
#How do you extract the first names from the Name column?

df['First_Name'] = df['Name'].str.split().str[0]
df
```

	StudentID	Name	Major	GPA	State	Pass	Honor	First_Name
0	301	Alice Brown	Computer Science	4.0	California	True	Yes	Alice
1	302	Ben Johnson	Mechanical Eng.	3.4	Texas	True	No	Ben
2	303	Clara Davis	Psychology	3.6	Florida	True	No	Clara
3	304	Daniel Lee	Business Mgmt.	3.3	Illinois	True	No	Daniel
4	305	Eva Wilson	Biology	3.9	New York	True	Yes	Eva
5	306	Frank Miller	Computer Science	3.0	California	True	No	Frank
6	307	Grace Taylor	English Literature	3.7	Ohio	True	Yes	Grace
7	308	Henry Clark	Mathematics	3.8	Texas	True	Yes	Henry
8	309	Irene White	Chemistry	3.5	Washington	True	No	Irene
9	310	Jack Lewis	Business Mgmt.	3.2	Florida	True	No	Jack
10	311	Karen Young	Biology	4.0	California	True	Yes	Karen
11	312	Liam Scott	Computer Science	3.7	New York	True	Yes	Liam
12	313	Mia Adams	Mathematics	3.3	Texas	True	No	Mia
13	314	Noah Carter	Mechanical Eng.	3.1	Illinois	True	No	Noah
14	315	Olivia Green	Psychology	3.9	Florida	True	Yes	Olivia
15	316	Paul Walker	Business Mgmt.	3.6	Ohio	True	No	Paul
16	317	Quinn Rivera	English Literature	3.8	Texas	True	Yes	Quinn
17	318	Ryan Hall	Biology	2.9	California	False	No	Ryan
18	319	Sophia King	Mathematics	4.0	Washington	True	Yes	Sophia

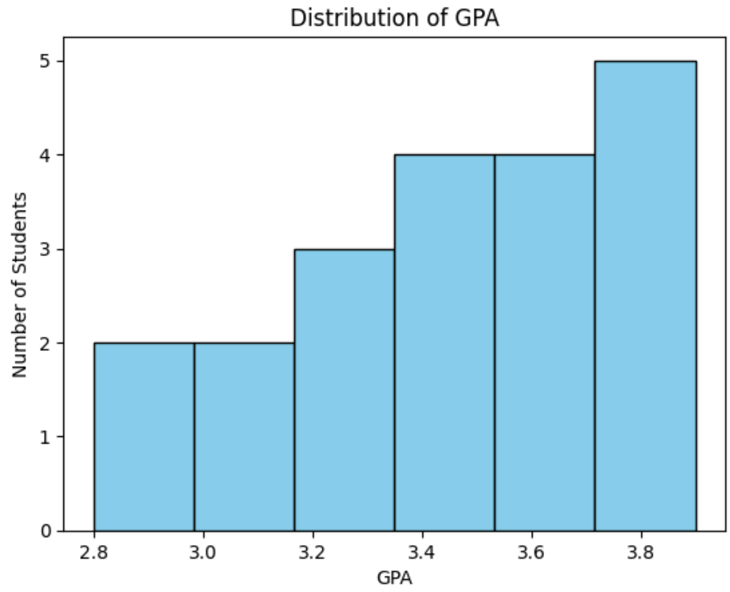
```
import pandas as pd

data = {
    "StudentID": range(301, 321),
    "Name": [
        "Alice Brown", "Ben Johnson", "Clara Davis", "Daniel Lee", "Eva Wilson",
        "Frank Miller", "Grace Taylor", "Henry Clark", "Irene White", "Jack Lewis",
        "Karen Young", "Liam Scott", "Mia Adams", "Noah Carter", "Olivia Green",
        "Paul Walker", "Quinn Rivera", "Ryan Hall", "Sophia King", "Thomas Allen"
    ],
    "Major": [
        "Computer Science", "Mechanical Eng.", "Psychology", "Business Mgmt.", "Biology",
        "Computer Science", "English Literature", "Mathematics", "Chemistry", "Business Mgmt.",
        "Biology", "Computer Science", "Mathematics", "Mechanical Eng.", "Psychology",
        "Business Mgmt.", "English Literature", "Biology", "Mathematics", "Computer Science"
    ],
    "GPA": [
        3.9, 3.3, 3.5, 3.2, 3.8, 2.9, 3.6, 3.7, 3.4, 3.1,
        3.9, 3.6, 3.2, 3.0, 3.8, 3.5, 3.7, 2.8, 3.9, 3.4
    ],
    "State": [
        "California", "Texas", "Florida", "Illinois", "New York", "California", "Ohio",
        "Texas", "Washington", "Florida", "California", "New York", "Texas", "Illinois",
        "Florida", "Ohio", "Texas", "California", "Washington", "Illinois"
    ]
}

df = pd.DataFrame(data)
```

```
df
```

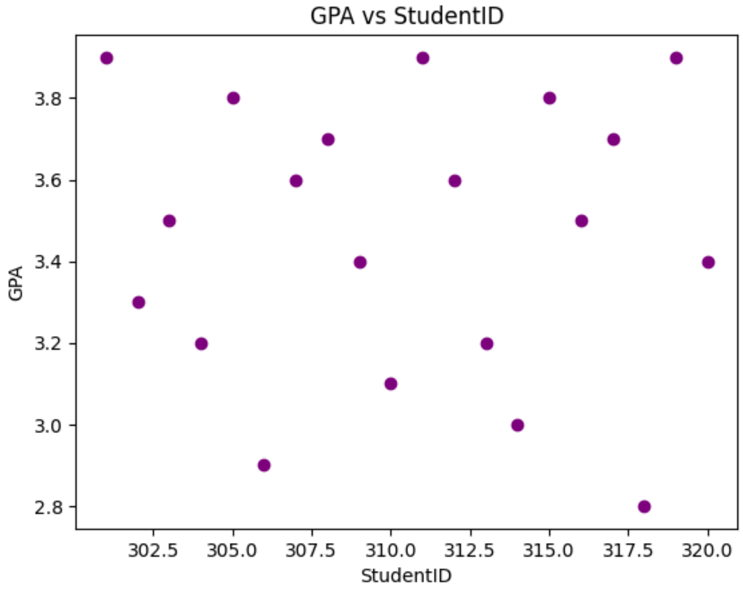
```
    StudentID          Name               Major  GPA       State
 0        301   Alice Brown    Computer Science  3.9  California
 1        302   Ben Johnson     Mechanical Eng.  3.3       Texas
 2        303   Clara Davis          Psychology  3.5     Florida
 3        304    Daniel Lee      Business Mgmt.  3.2    Illinois
 4        305    Eva Wilson             Biology  3.8    New York
 5        306  Frank Miller    Computer Science  2.9  California
 6        307  Grace Taylor  English Literature  3.6        Ohio
 7        308   Henry Clark         Mathematics  3.7       Texas
 8        309   Irene White           Chemistry  3.4  Washington
 9        310    Jack Lewis      Business Mgmt.  3.1     Florida
10        311   Karen Young             Biology  3.9  California
11        312    Liam Scott    Computer Science  3.6    New York
12        313     Mia Adams         Mathematics  3.2       Texas
13        314   Noah Carter     Mechanical Eng.  3.0    Illinois
14        315  Olivia Green          Psychology  3.8     Florida
15        316   Paul Walker      Business Mgmt.  3.5        Ohio
16        317  Quinn Rivera  English Literature  3.7       Texas
17        318     Ryan Hall             Biology  2.8  California
18        319   Sophia King         Mathematics  3.9  Washington
19        320  Thomas Allen    Computer Science  3.4    Illinois
```

Next steps: Generate code with df New interactive sheet

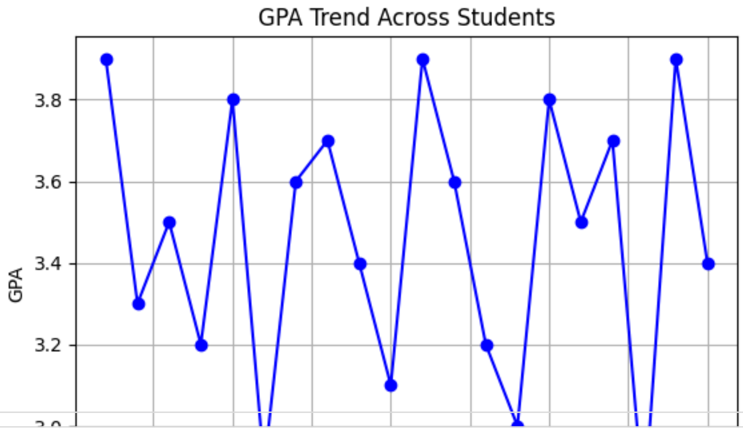
```
#GRAPHS
#HISTOGRAM GRAPH
#Plot a histogram of the students' GPA.

import matplotlib.pyplot as plt

plt.hist(df['GPA'], bins=6, color='skyblue', edgecolor='black')
plt.title('Distribution of GPA')
plt.xlabel('GPA')
plt.ylabel('Number of Students')
plt.show()
```



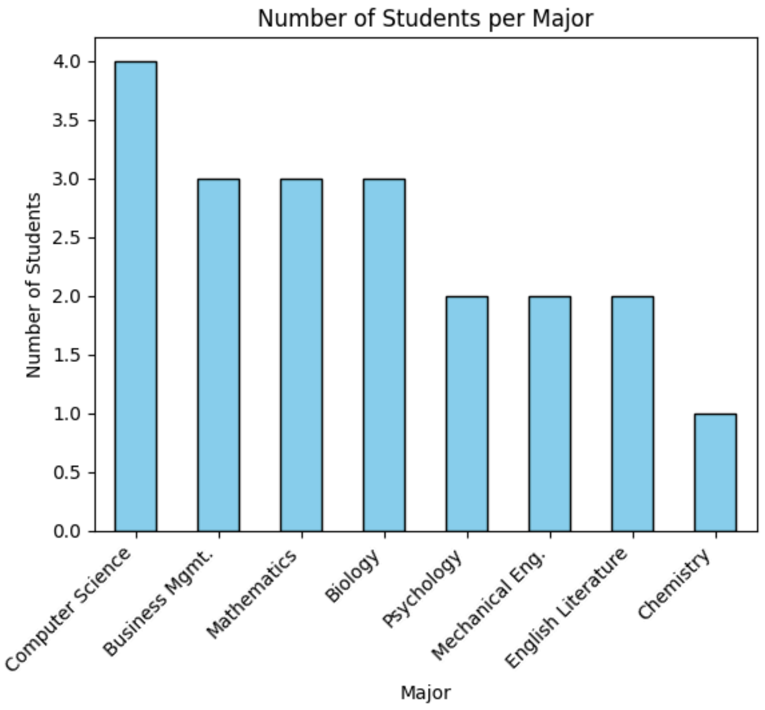
```
#SCATTER GRAPH
#Plot GPA vs StudentID to see individual performance trends.

plt.scatter(df['StudentID'], df['GPA'], color='purple')
plt.title('GPA vs StudentID')
plt.xlabel('StudentID')
plt.ylabel('GPA')
plt.show()
```



```
#LINE GRAPH
#Plot GPA against StudentID as a line plot to see GPA trends across students.

import matplotlib.pyplot as plt

plt.plot(df['StudentID'], df['GPA'], marker='o', color='blue', linestyle='-')
plt.title('GPA Trend Across Students')
plt.xlabel('StudentID')
plt.ylabel('GPA')
plt.grid(True)
plt.show()
```



```
#BAR GRAPH
# Count students per Major

major_counts = df['Major'].value_counts()

# Plot bar graph
major_counts.plot(kind='bar', color='skyblue', edgecolor='black')
plt.title('Number of Students per Major')
plt.xlabel('Major')
plt.ylabel('Number of Students')
plt.xticks(rotation=45, ha='right')
plt.show()
```


Number of Students per Major
Number of Students
Computer Science
Business Mgmt.
Mathematics
Biology
Psychology
Mechanical Eng.
English Literature
Chemistry
Major