A. Data Loading

Load the dataset using **pandas**

In [1]:
```
import pandas as pd
```

In [12]:
```
df = pd.read_csv("student-mat.csv", delimiter=";")
```

Display the first few rows using .head().

In [22]:
```
df.head()
```

Out[22]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15	
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6	

5 rows × 33 columns

B. Data Exploration

In [23]:
```
df.columns
```

Out[23]:
```
Index(['school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu',
       'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime',
       'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery',
       'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc',
       'Walc', 'health', 'absences', 'G1', 'G2', 'G3'],
      dtype='object')
```

In [24]:
```
df.describe()
```

Out[24]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	Dalc	Walc	
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.
mean	16.696203	2.749367	2.521519	1.448101	2.035443	0.334177	3.944304	3.235443	3.108861	1.481013	2.291139	3.
std	1.276043	1.094735	1.088201	0.697505	0.839240	0.743651	0.896659	0.998862	1.113278	0.890741	1.287897	1.
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.000000	3.000000	2.000000	1.000000	1.000000	3.
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.000000	4.000000	3.000000	3.000000	1.000000	2.000000	4.
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	5.000000	4.000000	4.000000	2.000000	3.000000	5.
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.

In [28]:
```
df.shape
```

Out[28]:
```
(395, 33)
```

In [25]:
```
df.isnull().sum()
```

Out[25]:
```
school        0
sex           0
age           0
address       0
famsize       0
Pstatus       0
Medu          0
Fedu          0
Mjob          0
Fjob          0
reason        0
guardian      0
traveltime    0
studytime     0
failures      0
schoolsup     0
famsup        0
paid          0
activities    0
nursery       0
higher        0
internet      0
romantic      0
famrel        0
freetime      0
goout         0
Dalc          0
Walc          0
health        0
absences      0
G1            0
G2            0
G3            0
dtype: int64
```

C. Data Cleaning

In [20]:
```
df['sex'].unique()
```

Out[20]:
```
array(['F', 'M'], dtype=object)
```

In [21]:
```
df['school'].unique()
```

Out[21]:
```
array(['GP', 'MS'], dtype=object)
```

In [26]:
```
df.dtypes
```

Out[26]:
```
school        object
sex           object
age            int64
address       object
famsize       object
Pstatus       object
Medu           int64
Fedu           int64
Mjob          object
Fjob          object
reason        object
guardian      object
traveltime     int64
studytime      int64
failures       int64
schoolsup     object
famsup        object
paid          object
activities    object
nursery       object
higher        object
internet      object
romantic      object
famrel         int64
freetime       int64
goout          int64
Dalc           int64
Walc           int64
health         int64
absences       int64
G1             int64
G2             int64
G3             int64
dtype: object
```

In [28]:
```
df.shape
```

Out[28]: (395, 33)

In [29]:
```
df.drop_duplicates()
```

Out[29]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
390	MS	M	20	U	LE3	A	2	2	services	services	...	5	5	4	4	5	4	11	9
391	MS	M	17	U	LE3	T	3	1	services	services	...	2	4	5	3	4	2	3	14
392	MS	M	21	R	GT3	T	1	1	other	other	...	5	5	3	3	3	3	3	10
393	MS	M	18	R	LE3	T	3	2	services	other	...	4	4	1	3	4	5	0	11
394	MS	M	19	U	LE3	T	1	1	other	at_home	...	3	2	3	3	3	5	5	8

395 rows × 33 columns

D. Data Analysis Questions

1. What is the average score in math (G3)?

In [36]:
```
round(df['G3'].mean(),2)
```

Out[36]: 10.42

Avg of score in Math G3 is 10.42

2. How many students scored above 15 in their final grade (G3)?

In [56]:
```
df[df['G3'] > 15].shape[0]
```

Out[56]: 40

In [57]:
```
df[df['G3'] > 15]['G3'].count()
```

Out[57]: 40

3. Is there a correlation btw study time (study time) and the final grade (G3)?

In [58]:
```
correlation = df['studytime'].corr(df['G3'])
print(correlation)
```

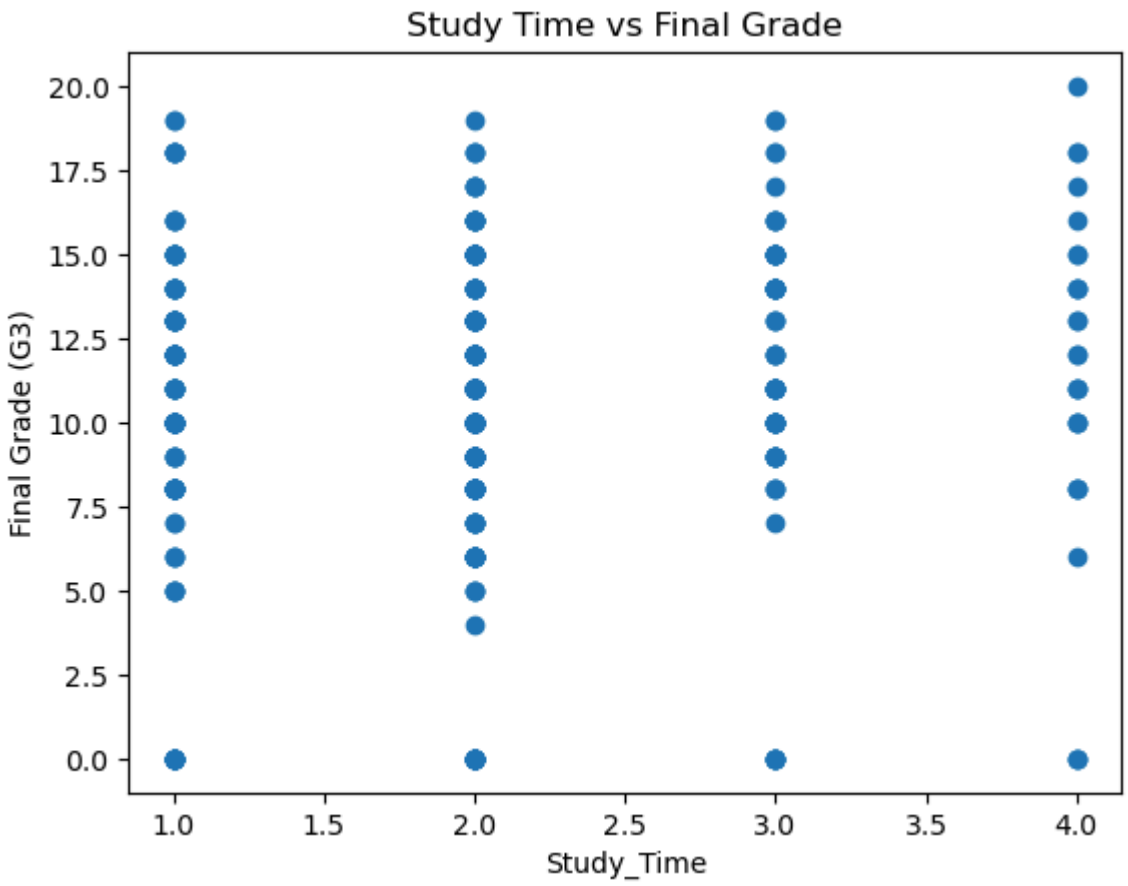
```
0.09781968965319636
```

Interpretation:

- If correlation > 0, there is a positive relationship (more study time leads to higher grades).
- If correlation < 0, there is a negative relationship (more study time leads to lower grades).
- If correlation ≈ 0, there is no strong relationship between study time and final grades.

Our answer is >0, So positive correlation

In [61]:
```
import matplotlib.pyplot as plt

plt.scatter(df['studytime'], df['G3'])
plt.xlabel('Study_Time')
plt.ylabel('Final Grade (G3)')
plt.title('Study Time vs Final Grade')
plt.show()
```



4. Which gender has a higher average final grade (G3) ?

In [64]:
```
df.groupby('sex')['G3'].mean()
```

Out[64]:
```
sex
F     9.966346
M    10.914439
Name: G3, dtype: float64
```

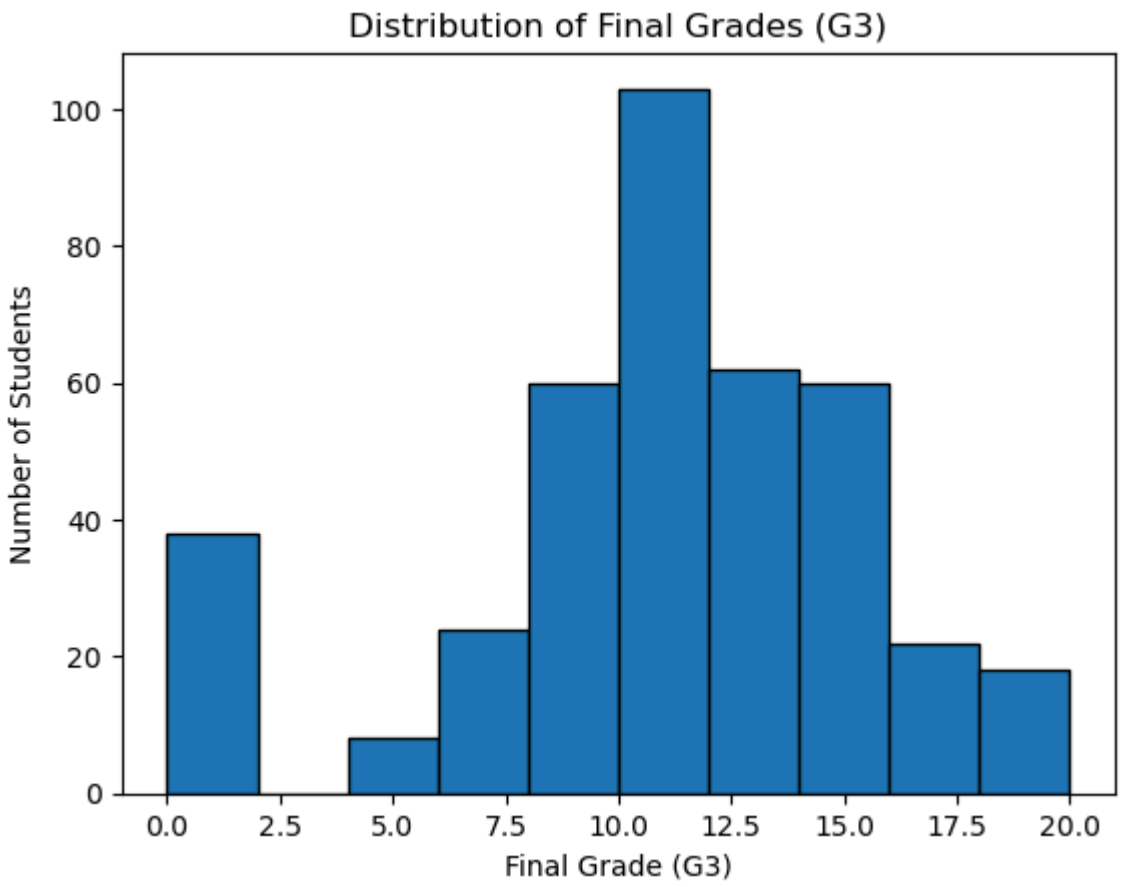
From the results, **males (M)** have a higher average final grade (`G3`) than **females (F)**`:

- **Females (F):** 9.97
- **Males (M):** 10.91

This means, on average, male students scored higher in their final grade (`G3`) compared to female students.

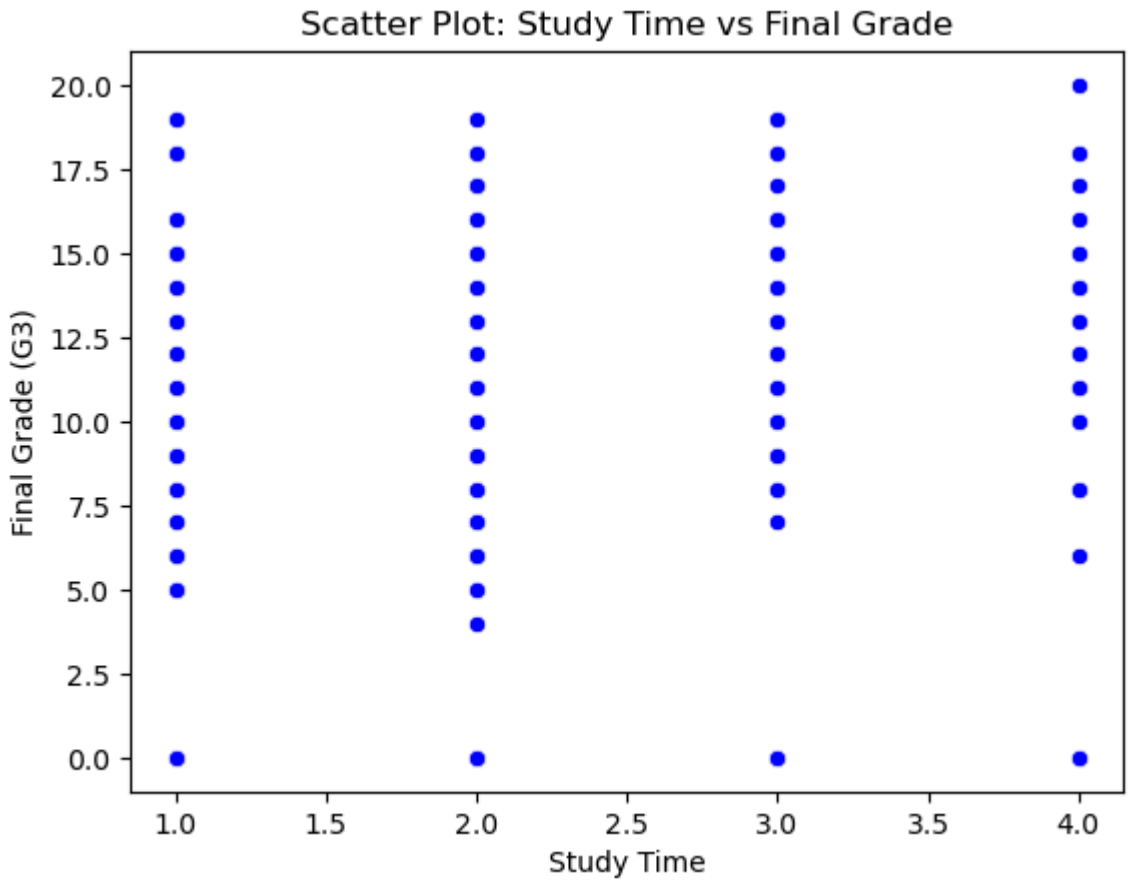
E. Data Visualization

In [67]:
```
import matplotlib.pyplot as plt

plt.hist(df['G3'], bins=10, edgecolor='black')
plt.xlabel('Final Grade (G3)')
plt.ylabel('Number of Students')
plt.title('Distribution of Final Grades (G3)')
plt.show()
```



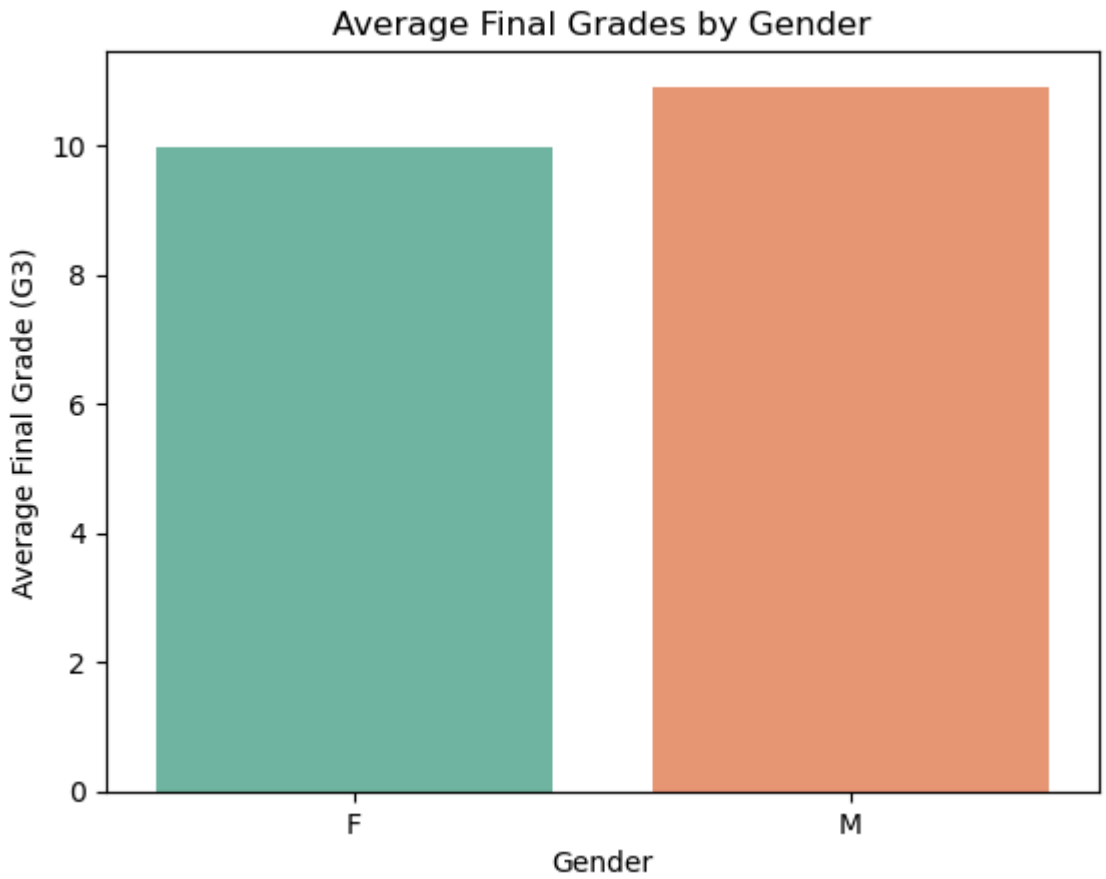
2. Scatter plot between study time (study time) and final grade (G3)

In [69]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(x='studytime', y='G3', data=df, color='blue')
plt.xlabel('Study Time')
plt.ylabel('Final Grade (G3)')
plt.title('Scatter Plot: Study Time vs Final Grade')
plt.show()
```



3. Bar chart comparing the average scores of male and female students

In [70]:
```
import seaborn as sns
import matplotlib.pyplot as plt

avg_scores = df.groupby('sex')['G3'].mean().reset_index()
sns.barplot(x='sex', y='G3', data=avg_scores, palette='Set2')
plt.xlabel('Gender')
plt.ylabel('Average Final Grade (G3)')
plt.title('Average Final Grades by Gender')
plt.show()
```



In [ ]: