

In [41]: # Assignment 2-1

```
numbers = [19,3,15,7,11]

print("\nCreating a bar chart from numbers")
print(f'Index{"Value":>8} bar')

for index, value in enumerate(numbers):
    print(f'{index: > 5} {value: > 8} {"*"* value}')
```

```
Creating a bar chart from numbers
Index  Value bar
0      19 *****
1      3 ***
2      15 *****
3      7 ****
4      11 *****
```

In line 5:

f'...' tells Python it's a formatted string (f-string).

"Value":>8 means "take the string "Value" and put it inside a field 8 characters aligned to the right."

This ensures the word "Value" starts under the numbers in the next rows, so everything lines up like a table.

In line 7 & 8:

enumerate(numbers) gives you both index (0,1,2,...) and value (19,3,15,...).

print{index: > 5} this makes small numbers (0, 1, 2, ...) line up neatly.

{value: > 8} same idea, but for the actual number in the list.

{"\* "\* value} makes a bar by repeating "\*" value times. e.g. if value=11, it produces "\*\*\*\*\*".

The repeated \* gives the "bar" effect.

In [1]: #Assignment 2-2

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

df = pd.read_csv('scores-2.csv')
```

In [16]: df

Out[16]:

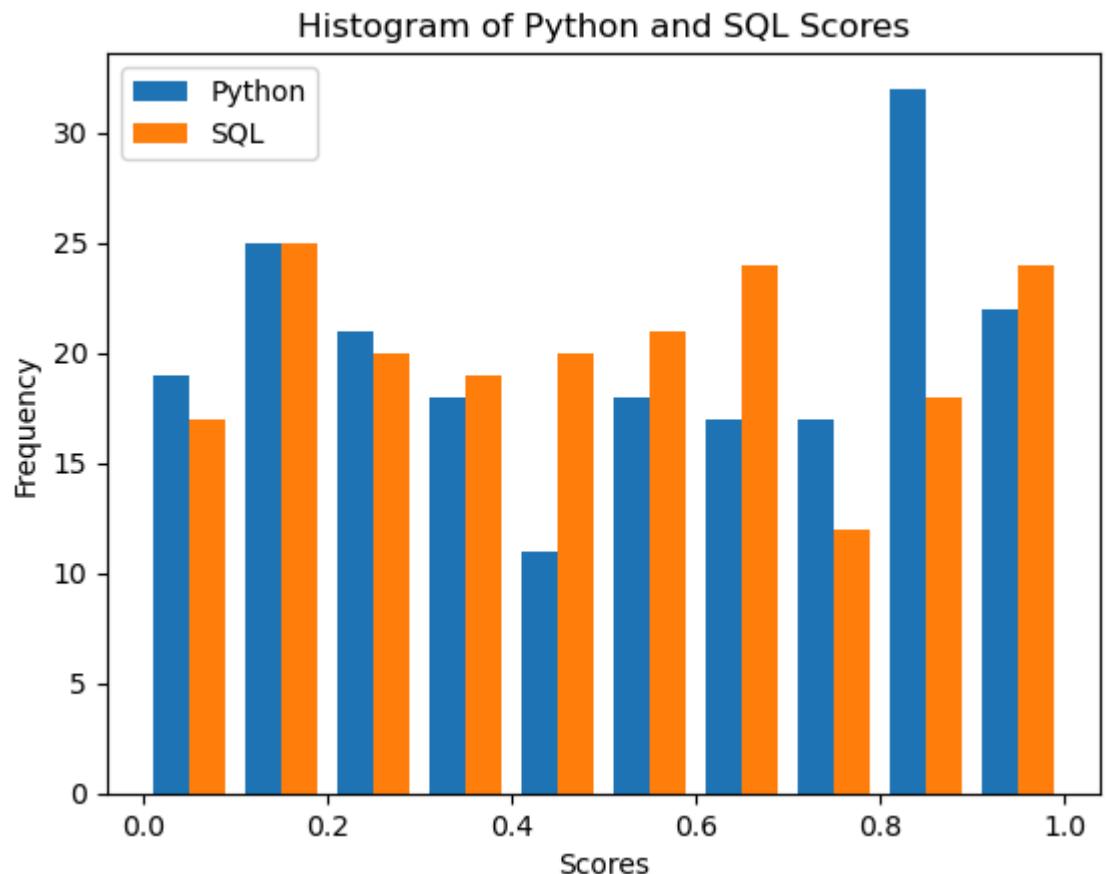
	Python	Sql	ML	Tableau	Excel	Student Placed
0	0.80	0.57	0.63	0.50	0.34	Yes
1	0.81	0.90	0.62	0.71	0.92	No
2	0.49	0.69	0.62	0.64	0.41	No
3	0.40	0.94	0.60	0.26	0.47	No
4	0.31	0.87	1.00	0.23	0.99	No
...	...	...	...	...	...	...
195	0.47	0.13	0.78	0.10	0.11	Yes
196	0.14	0.49	0.19	0.33	0.85	Yes
197	0.95	0.65	0.62	0.74	0.61	Yes
198	0.60	0.99	0.66	0.71	0.93	Yes
199	0.87	0.23	0.97	0.45	0.42	Yes

200 rows × 6 columns

In [7]: #1. Use Matplotlib to draw histograms for 'Python' and 'Sql' together :

```
plt.hist([df['Python'], df['Sql']], label=["Python", "SQL"])

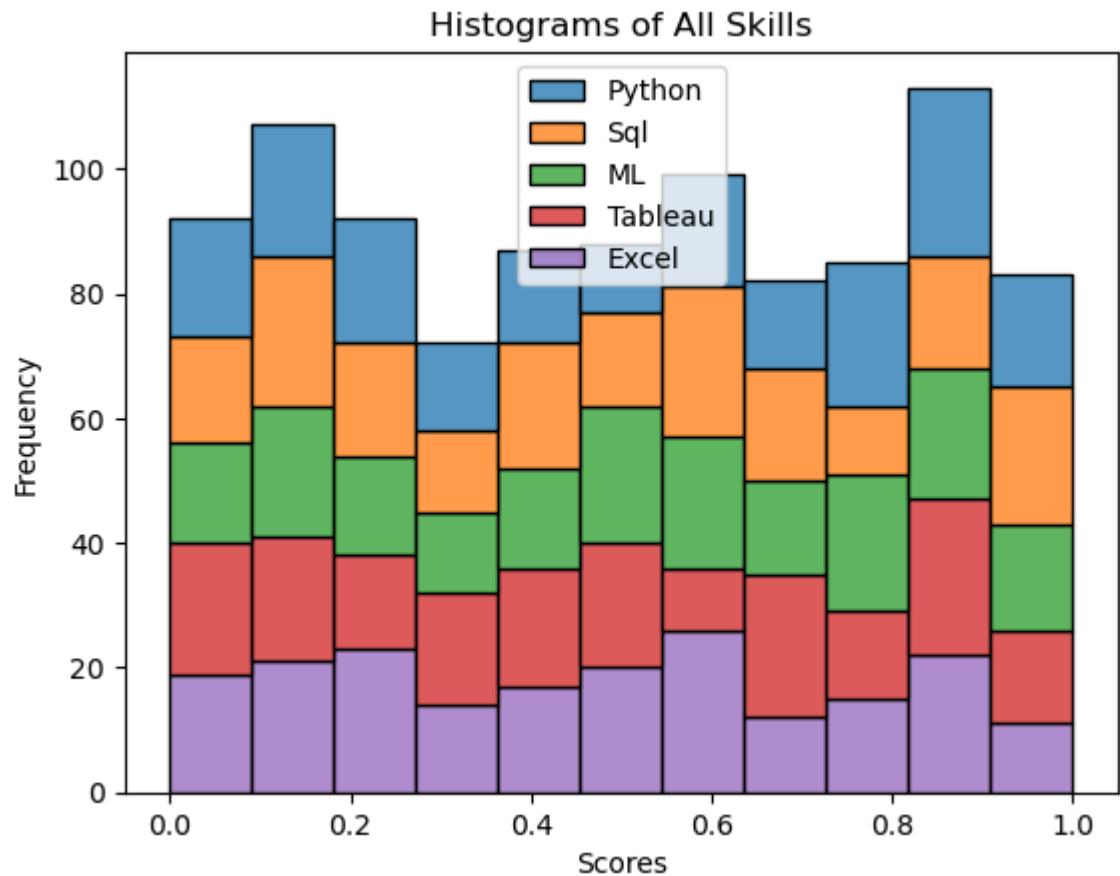
plt.title("Histogram of Python and SQL Scores")
plt.xlabel("Scores")
plt.ylabel("Frequency")
plt.legend()
plt.show()
```



In [40]: #2. Use seaborn to draw Histograms (within a single graph) for all the skills

```
sns.histplot(data=df,multiple="stack")

plt.title("Histograms for All Data")
plt.xlabel("Scores")
plt.ylabel("Frequency")
plt.show()
```



In [ ]: