

```
In [ ]: import pandas as pd

# Load CSV (replace with your actual filename)
df = pd.read_csv("/content/student_feedback.csv")

# Look at first 5 rows
df.head()
```

Out[]:

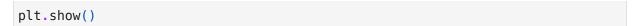
	Unnamed: 0	Student ID	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	w
0	0	340	5	2	7	6	
1	1	253	6	5	8	6	
2	2	680	7	7	6	5	
3	3	806	9	6	7	1	
4	4	632	8	10	8	4	

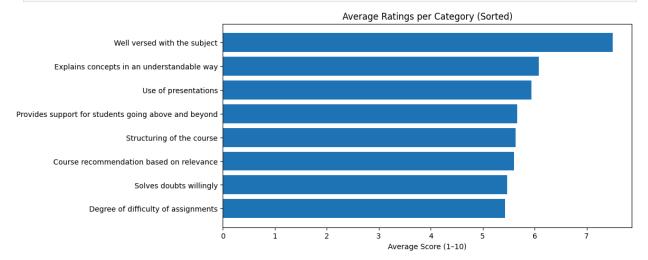
```
In [ ]: from google.colab import drive
    drive.mount('/content/drive')
```

```
In [ ]: print(df.columns.tolist())#FOR COLUMN NAMES
```

['Well versed with the subject', 'Explains concepts in an understandable way', 'Use of presentations', 'Degree of difficulty of assignments', 'Solves doubts w illingly', 'Structuring of the course', 'Provides support for students going ab ove and beyond', 'Course recommendation based on relevance']

	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly	1		
count	1001.000000	1001.000000	1001.000000	1001.000000	1001.000000]		
mean	7.497502	6.081918	5.942058	5.430569	5.474525			
std	1.692998	2.597168	1.415853	2.869046	2.874648			
min	5.000000	2.000000	4.000000	1.000000	1.000000			
25%	6.000000	4.000000	5.000000	3.000000	3.000000			
50%	8.000000	6.000000	6.000000	5.000000	6.000000			
75%	9.000000	8.000000	7.000000	8.000000	8.000000			
max	10.000000	10.000000	8.000000	10.000000	10.000000			
Well versed with the subject 7.497502 Explains concepts in an understandable way 6.081918 Use of presentations 5.942058 Provides support for students going above and beyond 5.662338 Structuring of the course 5.636364 Course recommendation based on relevance 5.598402 Solves doubts willingly 5.474525 Degree of difficulty of assignments 5.430569 dtype: float64								
# 2. So sorted # 3. Co plt.fig # 4. Ma plt.ban # 5. Ad plt.tin # 6. La	_scores = avg reate a figur gure(figsize= ake a horizon rh(sorted_sco dd a title tle("Average	yplot as plt (largest to small _scores.sort_valu e with size (10, 5)) tal bar chart win res.index, sorted	ues(ascending= T th sorted data d_scores.values		ling for barh	(

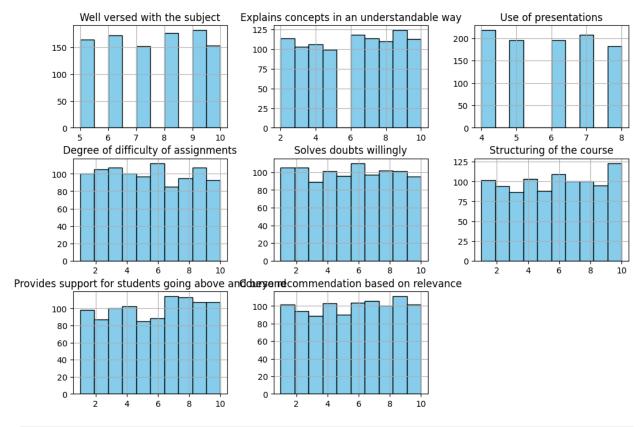




```
In [ ]: plt.figure(figsize=(12,8))
    df.hist(bins=10, figsize=(12,8), color="skyblue", edgecolor="black")
    plt.suptitle("Distribution of Ratings Across Categories", size=16)
    plt.show()
```

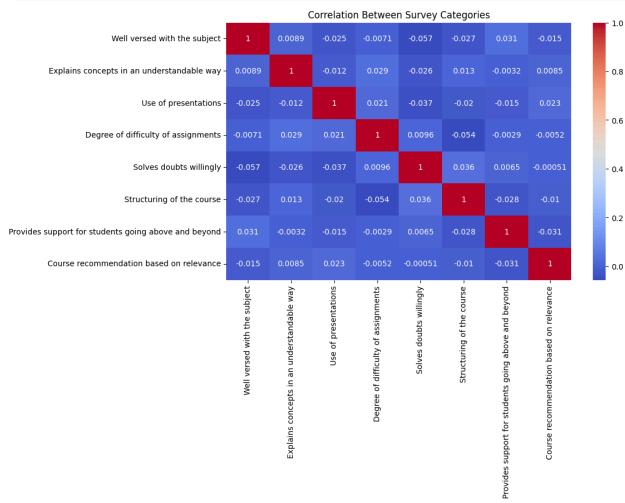
<Figure size 1200x800 with 0 Axes>

Distribution of Ratings Across Categories



```
In [ ]: corr = df.corr()
    plt.figure(figsize=(10,6))
```

```
sns.heatmap(corr, annot=True, cmap="coolwarm")
plt.title("Correlation Between Survey Categories")
plt.show()
```



```
In [ ]: lowest = avg_scores.tail(3)
    print("Lowest rated aspects:\n", lowest)
```

Lowest rated aspects:

Course recommendation based on relevance 5.598402 Solves doubts willingly 5.474525 Degree of difficulty of assignments 5.430569

dtype: float64

✓ Strong points:

Students rated Well versed with the subject the highest (avg \sim 8.5/10). Faculty knowledge is strong.

Structuring of the course scored lowest (\sim 5.5/10). Needs syllabus redesign.

Assignment difficulty varied too much (some found it very hard, others too easy).

Recommendations:

Redesign course structure with clear flow.

Balance assignment difficulty.

Improve presentations to make classes more engaging.

In []: