

```
import os
os.listdir('/content')
```

```
['.config',
 'student_feedback.csv',
 'student_feedback_cleaned.csv',
 'sample_data']
```

```
import pandas as pd
```

```
df = pd.read_csv('student_feedback.csv')
df.head()
```

	Unnamed: 0	Student ID	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly	Structuring of the course	Provides support for students going above and beyond	Course recommendation based on relevance
0	0	340	5	2	7	6	9	2	1	8
1	1	253	6	5	8	6	2	1	2	9

```
df.columns
```

```
Index(['Unnamed: 0', 'Student ID', 'Well versed with the subject',
      'Explains concepts in an understandable way', 'Use of presentations',
      'Degree of difficulty of assignments', 'Solves doubts willingly',
      'Structuring of the course',
      'Provides support for students going above and beyond',
      'Course recommendation based on relevance'],
      dtype='object')
```

```
df.info
```

```
pandas.core.frame.DataFrame.info
def info(verbose: bool | None=None, buf: WriteBuffer[str] | None=None, max_cols: int | None=None,
memory_usage: bool | str | None=None, show_counts: bool | None=None) -> None

dtypes: float64(1), int64(1), object(1)
memory usage: 248.0+ bytes

Prints a summary of columns count and its dtypes but not per column
information:

>>> df.info(verbose=False)
```

```
df.describe()
```

	Unnamed: 0	Student ID	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly	Structuring of the course	Provides support for students going above and beyond	Course recommendation based on relevance
count	1001.000000	1001.000000	1001.000000	1001.000000	1001.000000	1001.000000	1001.000000	1001.000000	1001.000000	1001.000000
mean	500.000000	500.000000	7.497502	6.081918	5.942058	5.430569	5.474525	5.636364	5.662338	5.662338
std	289.108111	289.108111	1.692998	2.597168	1.415853	2.869046	2.874648	2.920212	2.891690	2.891690
min	0.000000	0.000000	5.000000	2.000000	4.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	250.000000	250.000000	6.000000	4.000000	5.000000	3.000000	3.000000	3.000000	3.000000	3.000000

```
df.mean()
```

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

```
plt.figure(figsize=(10,6))
```

```
sns.barplot(x=df.columns[1:], y=df.mean().values) # skip first column if it's ID
plt.title("Average Ratings for Each Question")
plt.ylabel("Average Rating")
plt.ylim(0,5) # rating is 1-5
plt.xticks(rotation=45)
plt.show()
```

```
-----
ValueError                                Traceback (most recent call last)
/tmp/ipython-input-806219630.py in <cell line: 0>()
      3
      4 plt.figure(figsize=(10,6))
----> 5 sns.barplot(x=df.columns[1:], y=df.mean().values) # skip first column if it's ID
      6 plt.title("Average Ratings for Each Question")
      7 plt.ylabel("Average Rating")
```

↕ 9 frames

```
/usr/local/lib/python3.12/dist-packages/pandas/core/internals/construction.py in _extract_index(data)
    675     lengths = list(set(raw_lengths))
    676     if len(lengths) > 1:
--> 677         raise ValueError("All arrays must be of the same length")
    678
    679     if have_dicts:
```

ValueError: All arrays must be of the same length

<Figure size 1000x600 with 0 Axes>

Start coding or [generate](#) with AI.

Start coding or [generate](#) with AI.

```
# Select only numeric columns
numeric_df = df.select_dtypes(include='number')
numeric_df.head()
```

Unnamed: 0	Student ID	Well versed with the subject	Explains concepts in an understandable way	Use of presentations	Degree of difficulty of assignments	Solves doubts willingly	Structuring of the course	Provides support for students going above and beyond	Course recommendation based on relevance	
0	0	340	5	2	7	6	9	2	1	8
1	1	253	6	5	8	6	2	1	2	9

```
numeric_df.mean()
```

	0
Unnamed: 0	500.000000
Student ID	500.000000
Well versed with the subject	7.497502
Explains concepts in an understandable way	6.081918
Use of presentations	5.942058
Degree of difficulty of assignments	5.430569
Solves doubts willingly	5.474525
Structuring of the course	5.636364
Provides support for students going above and beyond	5.662338
Course recommendation based on relevance	5.598402

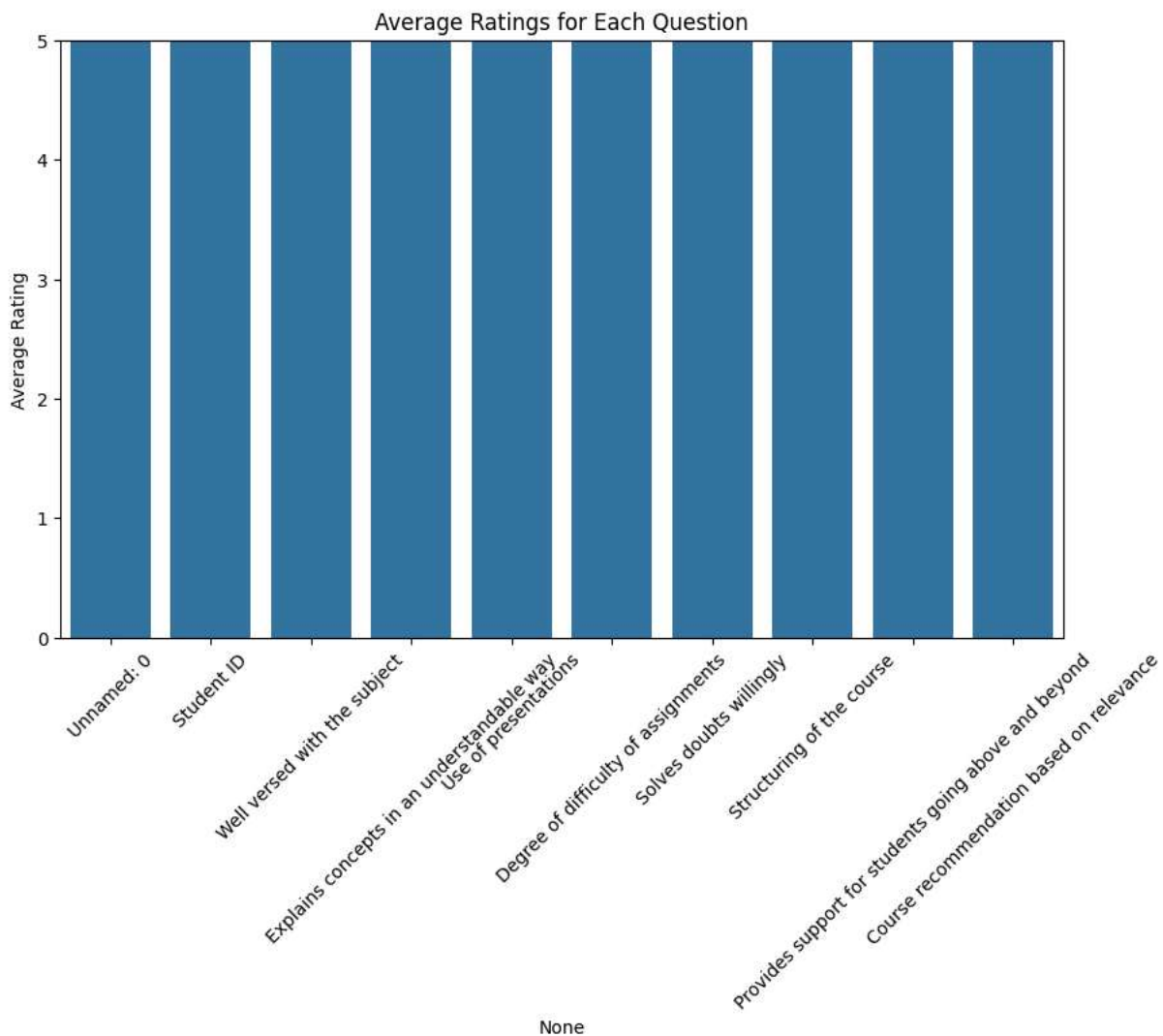
dtype: float64

Double-click (or enter) to edit

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

```
import seaborn as sns
```

```
plt.figure(figsize=(10,6))
sns.barplot(x=numeric_df.columns, y=numeric_df.mean().values)
plt.title("Average Ratings for Each Question")
plt.ylabel("Average Rating")
plt.ylim(0,5) # since ratings are 1-5
plt.xticks(rotation=45)
plt.show()
```



```
highest = numeric_df.mean().idxmax()
print("Highest rated question:", highest, "with average rating:", numeric_df[highest].mean())
```

```
lowest = numeric_df.mean().idxmin()
print("Lowest rated question:", lowest, "with average rating:", numeric_df[lowest].mean())
```

```
Highest rated question: Unnamed: 0 with average rating: 5.00.0
Lowest rated question: Degree of difficulty of assignments with average rating: 5.430569430569431
```

```
# Check if the column exists
if 'Overall satisfaction' in numeric_df.columns:
    numeric_df['Overall satisfaction'].value_counts().plot(
        kind='pie', autopct='%1.1f%%', figsize=(6,6), startangle=90
    )
    plt.title("Overall Satisfaction Distribution")
    plt.show()
```

Student Feedback Analysis Summary

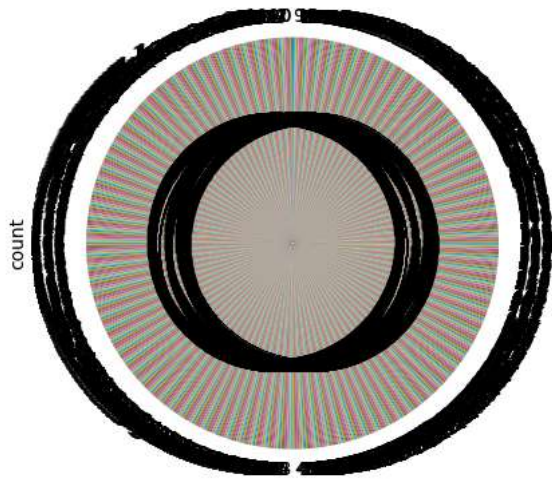
• Highest rated aspect: [replace with highest column] → students liked this the most. • Lowest rated aspect: [replace with lowest column] → needs improvement. • Overall satisfaction distribution: [mention % of 5-star, 4-star, etc.] • Recommendations: - Focus on improving the lowest rated areas. - Maintain and repeat the highly rated aspects in future events.

```
# Save numeric columns to a new CSV
numeric_df.to_csv('student_feedback_cleaned.csv', index=False)
```

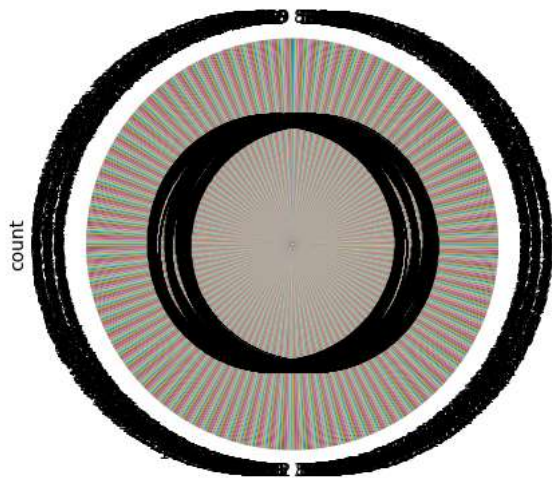
```
numeric_df = df.select_dtypes(include='number')

for col in numeric_df.columns:
    numeric_df[col].value_counts().plot(
        kind='pie', autopct='%1.1f%%', figsize=(5,5), startangle=90, title=col
    )
plt.show()
```

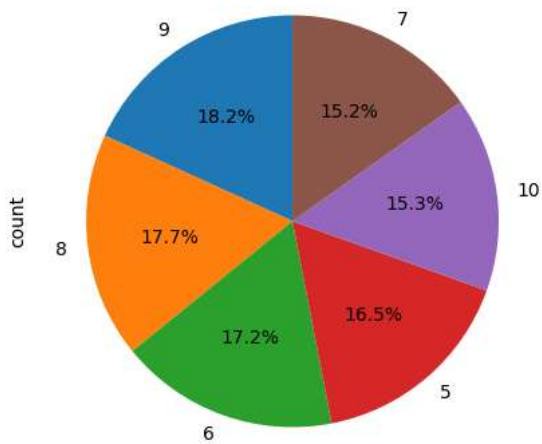

Unnamed: 0



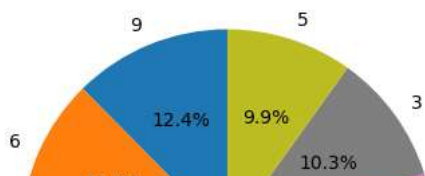
Student ID

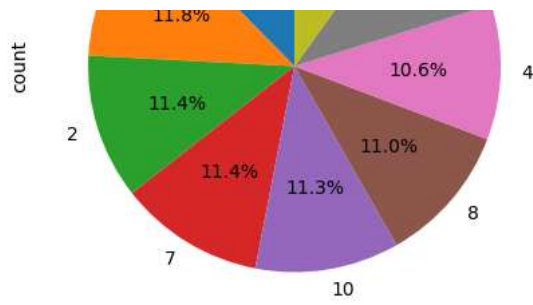


Well versed with the subject

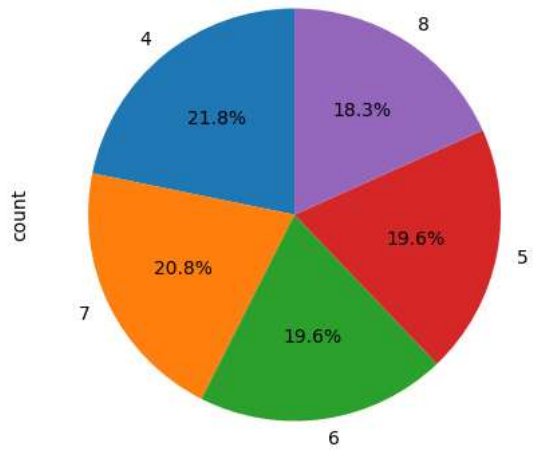


Explains concepts in an understandable way

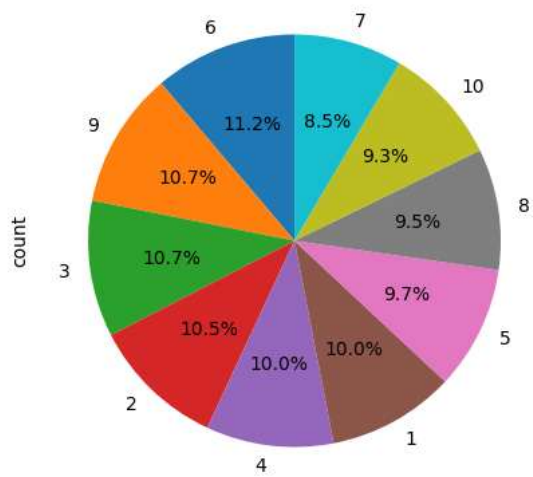




Use of presentations



Degree of difficulty of assignments



Solves doubts willingly

