```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files student_feedback.csv
student_feedback.csv(text/csv) - 25879 bytes, last modified: 12/17/2025 - 100% done
Saving student_feedback.csv to student_feedback (1).csv
```

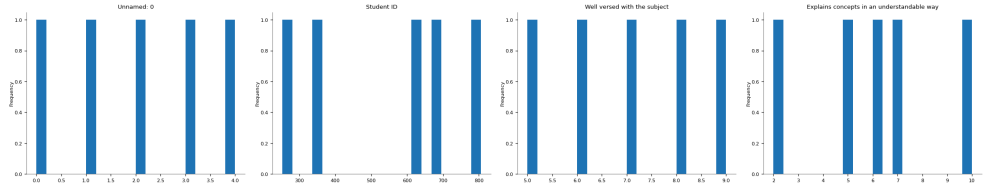
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from textblob import TextBlob
```

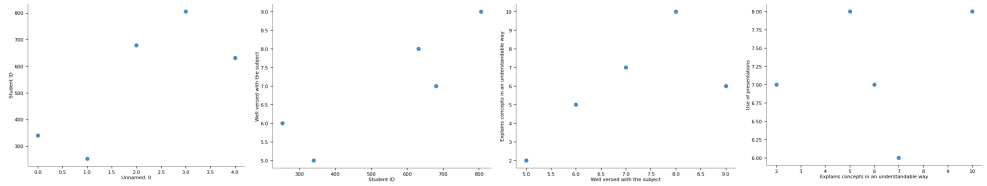
```
df = pd.read_csv('student_feedback.csv')
df.head()
```

	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
2	2	680	7	7	6	5	4	2	3	1
3	3	806	9	6	7	1	5	9	4	6
4	4	632	8	10	8	4	6	6	9	9

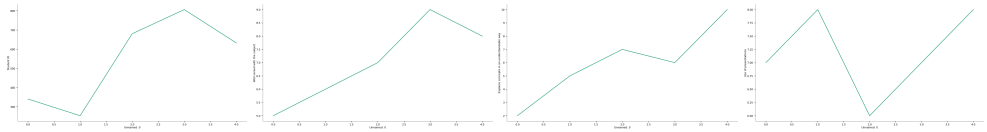
Distributions



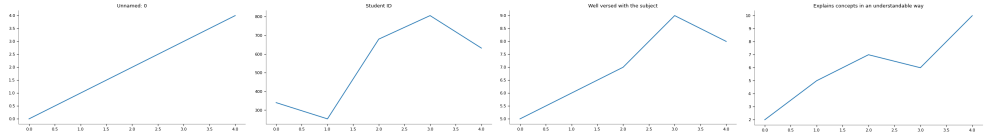
2-d distributions



Time series



Values



Next steps: Generate code with df New interactive sheet

```
df.info()
df.columns
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1001 entries, 0 to 1000
Data columns (total 10 columns):
 #   Column                                            Non-Null Count  Dtype
---  ------                                            --------------  -----
 0   Unnamed: 0                                        1001 non-null   int64
 1   Student ID                                        1001 non-null   int64
 2   Well versed with the subject                      1001 non-null   int64
 3   Explains concepts in an understandable way        1001 non-null   int64
 4   Use of presentations                              1001 non-null   int64
 5   Degree of difficulty of assignments               1001 non-null   int64
 6   Solves doubts willingly                           1001 non-null   int64
 7   Structuring of the course                         1001 non-null   int64
 8   Provides support for students going above and beyond  1001 non-null   int64
 9   Course recommendation based on relevance          1001 non-null   int64
dtypes: int64(10)
memory usage: 78.3 KB
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')
```

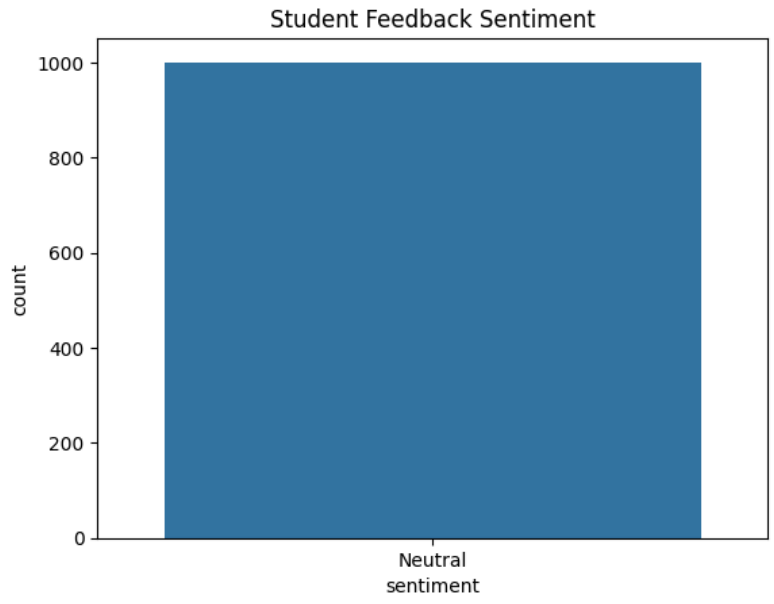
```
df.dropna(inplace=True)
```

```
df.rename(columns={"What did you like about the event?":"feedback"},
inplace=True)
```

```
def get_sentiment(text):
  return TextBlob(text).sentiment.polarity
```

```
df["sentiment_score"] = df["feedback"].apply(get_sentiment)
```

```
sns.countplot(x="sentiment", data=df)
plt.title("Student Feedback Sentiment")
plt.show()
```




```
if 'feedback' in df.columns:
    print("Column 'feedback' exists in the DataFrame.")
else:
    print("Column 'feedback' does NOT exist in the DataFrame.")
```

```
Column 'feedback' exists in the DataFrame.
```

```
Start coding or generate with AI.
```

```
def sentiment_label(score):
    if score > 0:
        return "Positive"
    elif score < 0:
        return "Negative"
    else:
        return "Neutral"

df["sentiment"] = df["sentiment_score"].apply(sentiment_label)
```

```
import pandas as pd

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

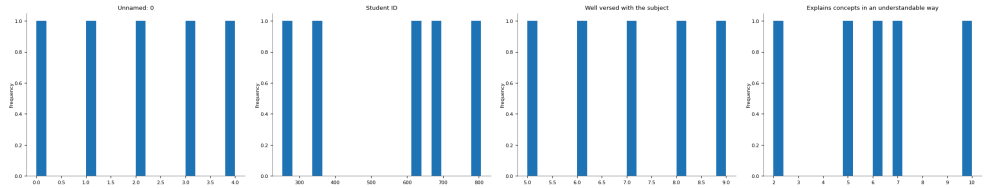
1 to 5 of 5 entries Filter

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

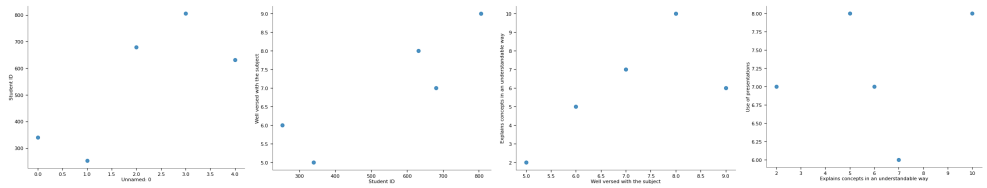
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

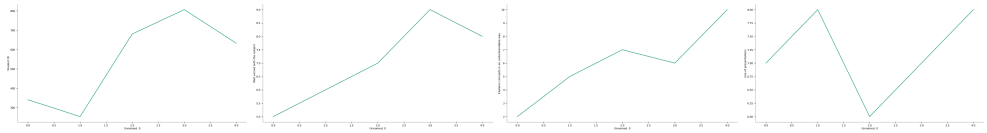
Distributions



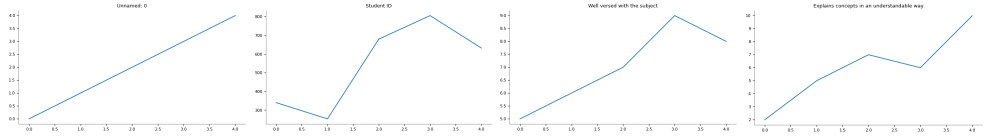
2-d distributions



Time series



Values

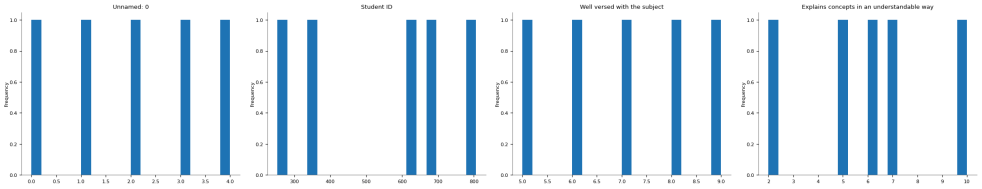


```
df = pd.read_csv('student_feedback.csv')
```

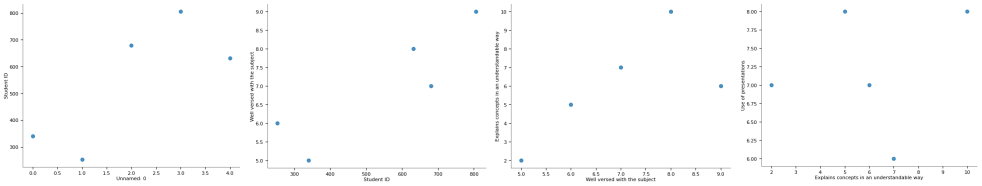
```
df.head()
```

	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
2	2	680	7	7	6	5	4	2	3	1
3	3	806	9	6	7	1	5	9	4	6
4	4	632	8	10	8	4	6	6	9	9

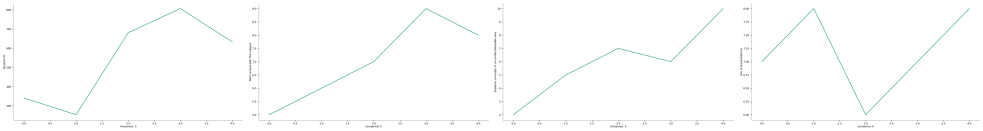
Distributions



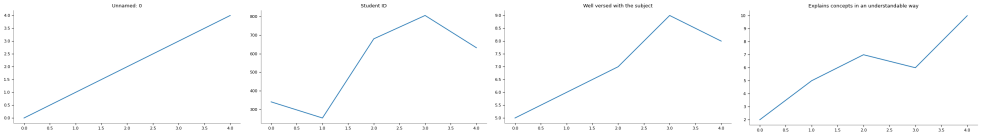
2-d distributions



Time series



Values



Next steps: Generate code with df | New interactive sheet

```
df["feedback"]=""
```

```
df["feedback"]= "Good event"
```

```
df["feedback"] = "This is a placeholder feedback text for all students."
```

First, let's create a function to generate feedback text based on the numerical rating columns.

```
def generate_feedback(row):
    feedback_parts = []

    # Instructor's subject knowledge
    subject_knowledge = row['Well versed with the subject']
    if subject_knowledge >= 8:
        feedback_parts.append("Instructor showed excellent
    elif subject_knowledge >= 5:
        feedback_parts.append("Instructor was well versed
    else:
        feedback_parts.append("Instructor needs to improve

    # Explains concepts
    explains_concepts = row['Explains concepts in an unders
```

```
    if explains_concepts >= 8:
        feedback_parts.append("Concepts were explained very
    elif explains_concepts >= 5:
        feedback_parts.append("Concepts were explained ade
    else:
        feedback_parts.append("Concepts could be explained

    # Use of presentations
    presentations = row['Use of presentations']
    if presentations >= 8:
        feedback_parts.append("Presentations were very effe
    elif presentations >= 5:
        feedback_parts.append("Presentations were used appr
    else:
        feedback_parts.append("Presentations could be more

    # Difficulty of assignments
    assignments_difficulty = row['Degree of difficulty of a
    if assignments_difficulty >= 8:
        feedback_parts.append("Assignments were challenging
    elif assignments_difficulty >= 5:
        feedback_parts.append("Assignments had a reasonable
    else:
        feedback_parts.append("Assignments were too easy.")

    # Solves doubts
    solves_doubts = row['Solves doubts willingly']
    if solves_doubts >= 8:
        feedback_parts.append("Instructor was very helpful
    elif solves_doubts >= 5:
        feedback_parts.append("Instructor was willing to so
    else:
        feedback_parts.append("Instructor was not very help

    # Structuring of the course
    course_structuring = row['Structuring of the course']
    if course_structuring >= 8:
        feedback_parts.append("The course was very well str
    elif course_structuring >= 5:
        feedback_parts.append("The course structuring was a
    else:
        feedback_parts.append("The course structure needs i

    # Support for students
    student_support = row['Provides support for students go
    if student_support >= 8:
        feedback_parts.append("Excellent support provided f
    elif student_support >= 5:
        feedback_parts.append("Good support for students se
    else:
        feedback_parts.append("More support needed for moti

    # Course recommendation
    course_recommendation = row['Course recommendation base
    if course_recommendation >= 8:
        feedback_parts.append("Highly recommend this course
    elif course_recommendation >= 5:
        feedback_parts.append("The course is relevant and r
    else:
        feedback_parts.append("Relevance of the course coul

    return " ".join(feedback_parts)
```

Now, let's apply this function to create the 'feedback' column with actual text data.

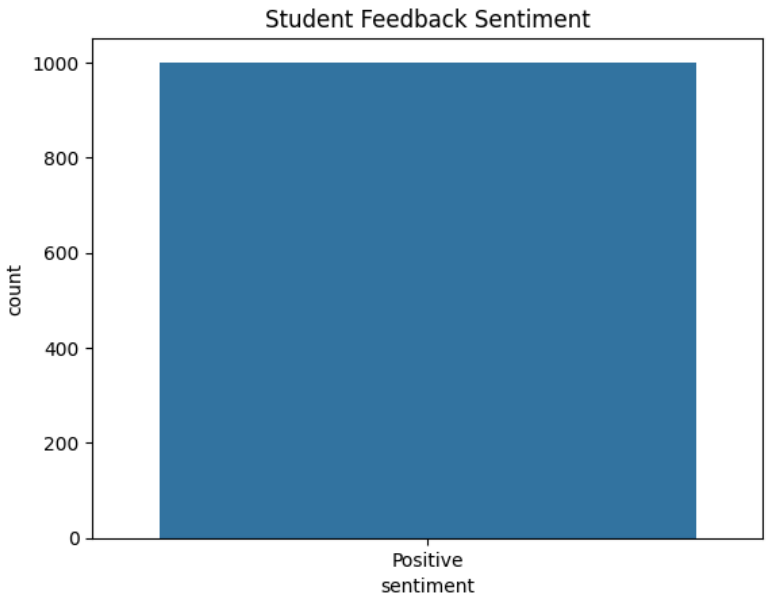
```
df["feedback"] = df.apply(generate_feedback, axis=1)
display(df[['Student ID', 'feedback']].head())
```

	Student ID	feedback
0	340	Instructor was well versed with the subject. C...
1	253	Instructor was well versed with the subject. C...
2	680	Instructor was well versed with the subject. C...
3	806	Instructor showed excellent subject knowledge....
4	632	Instructor showed excellent subject knowledge....

With the 'feedback' column now containing actual text, let's recalculate the sentiment scores and labels, and then display the updated sentiment distribution.

Gemini

```
ING: Runtime no longer has a reference to this dataframe, please re-run this cell and try again.
df["sentiment_score"] = df["feedback"].apply(get_sentiment)
df["sentiment"] = df["sentiment_score"].apply(sentiment_lab
sns.countplot(x="sentiment", data=df)
plt.title("Student Feedback Sentiment")
plt.show()
```




```
df.head()
```

Next steps: Generate code with df New interactive sheet

	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

```
from google.colab import files
uploaded = files.upload()
```

Choose Files updated_ev...edback.csv

updated_event_feedback.csv(text/csv) - 383858 bytes, last modified: 12/17/2025 - 100% done

```
Saving updated_event_feedback.csv to updated_event_feedback (1).csv
```

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

```
from textblob import TextBlob
```

```
df = pd.read_csv("updated_event_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
2	2	680	7	7	6	5	4	2	3	1
3	3	806	9	6	7	1	5	9	4	6
4	4	632	8	10	8	4	6	6	9	9

Next steps: Generate code with df New interactive sheet

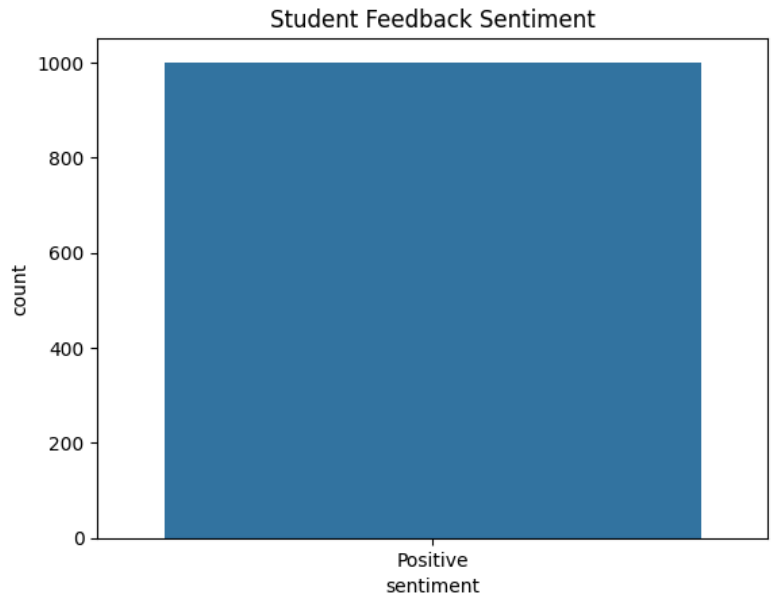
```
def get_sentiment(text):
    return TextBlob(text).sentiment.polarity
```

```
df["sentiment_score"] = df["feedback"].apply(get_sentiment)
```

```
def sentiment_label(score):
    if score > 0:
        return "Positive"
    elif score < 0:
        return "Negative"
    else:
        return "Neutral"

df["sentiment"] = df["sentiment_score"].apply(sentiment_label)
```

```
sns.countplot(x="sentiment", data=df)
plt.title("Student Feedback Sentiment")
plt.show()
```




```
df.head()
```

```
df["Overall Rating"] = ""
```

```
df["Overall Rating"] ="4,3,2,1"
```

```
df.head()
```

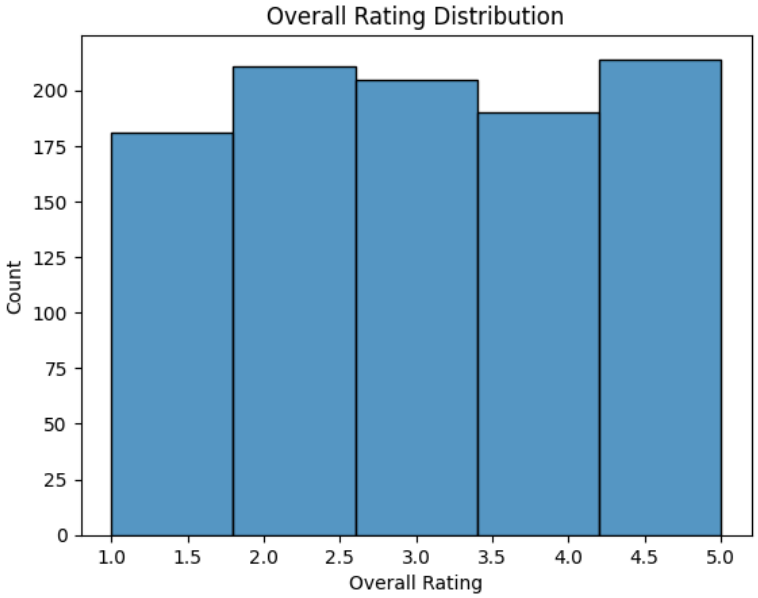
	Unnamed: 0	Student ID	Well versed with	Explains concepts in an understandable	Use of presentations	Degree of difficulty of	Solves doubts	Structuring of the	Provides support for students going beyond	Course recommendation based on	
1	1	253	6	5	8	6	2	1	2	9	v th
2	2	680	7	7	6	5	4	2	3	1	v th
3	3	806	9	6	7	1	5	9	4	6	kn
4	4	632	8	10	8	4	6	6	9	9	kn

Next steps: Generate code with df | New interactive sheet

Next steps: Generate code with df New interactive sheet

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
2	1	2	9	ve th
4	2	3	1	ve th
5	9	4	6	kno
6	6	9	9	kno

```
sns.histplot(df["Overall Rating"], bins=5)
plt.title("Overall Rating Distribution")
plt.show()
```



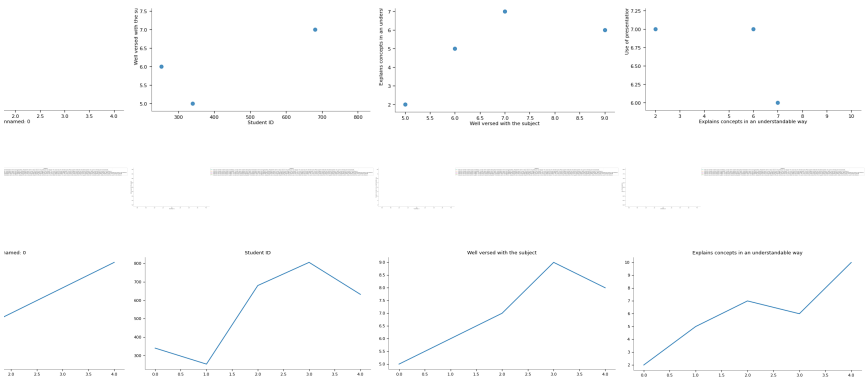

```
df["Overall Rating"].mean()
```

```
np.float64(3.044955044955045)
```

```
import numpy as np

# create overall rating from 1 to 5
df["Overall Rating"] = np.random.randint(1, 6, size=len(df))

df.head()
```



ibutions

```
FutureWarning:

.ette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and se

        <string>:5: FutureWarning:

.ette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and se

        <string>:5: FutureWarning:
```

```
.ette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and se
```

```
<string>:5: FutureWarning:
.ette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and se
```

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	
340	5	2	7	6	9	2	1	8	ve th
253	6	5	8	6	2	1	2	9	ve th
680	7	7	6	5	4	2	3	1	ve th
806	9	6	7	1	5	9	4	6	

Next steps: Generate code with df | New interactive sheet

```
import numpy as np

# create overall rating from 1 to 5
df["Overall Rating"] = np.random.randint(1, 6, size=len(df))

df.head()
```

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	
340	5	2	7	6	9	2	1	8	ve th
253	6	5	8	6	2	1	2	9	ve th
680	7	7	6	5	4	2	3	1	ve th
806	9	6	7	1	5	9	4	6	