

# The Challenges and opportunities of online teaching during COVID-19 pandemic.

## Loading the Data

```
In [27]: #Importing the necessary libraries
import pandas as pd
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import statsmodels.api as sm
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
print("Libraies successfully imported")
```

Libraies successfully imported

```
In [9]: # Load the dataset
data = pd.read_csv('C:\\Users\\n\\Downloads\\online_teaching_survey_data.csv')

# Show the first few rows of the dataset
data.head()
```

Out[9]:

fact	one_on_one_discussions	parent_support	additional_support	engagement_strategies	method_
sted	Once a week	Parents have played a crucial role in supporti...	To improve my online teaching, I would benefit...	To keep the students engaged in online, I used...	sign ada t I
sted	Twice a month	Parents where encouraging their children to pa...	Teachers to be trained on usage of tools to m...	I would be calling names who are distracted an...	
sted	Once a week	I got 50%parents support	My colleague support	Some special mudras and finger games used in m...	Yi intera si
sted	Daily	Parents helped their level a lot.	More Technical knowledge.	Will Interact often.	Ye cl
sted	Daily	Not all the parents are supportive to the chil...	No additional support, in every aspects we fin...	Using pictures,power point presentation multip...	Yes, ok and lea differen

Type *Markdown* and LaTeX:  $\alpha^2$

```
In [4]: # Get summary statistics and check for missing values
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   grade_level_teach                     14 non-null     object
1   adaptation_level                      14 non-null     object
2   curriculum_coverage                  14 non-null     object
3   student_engagement                   14 non-null     object
4   learning_outcomes                    14 non-null     object
5   technology_used                       14 non-null     object
6   tech_issues                          14 non-null     object
7   admin_support                        14 non-null     object
8   transition_difficulty                 14 non-null     int64
9   stress_impact                        14 non-null     object
10  one_on_one_discussions                14 non-null     object
11  parent_support                        14 non-null     object
12  additional_support                    14 non-null     object
13  engagement_strategies                 14 non-null     object
14  method_change                        14 non-null     object
15  tech_access                           14 non-null     object
16  socioeconomic_impact                  14 non-null     object
17  future_practices                      14 non-null     object
18  sustainability_belief                 14 non-null     object
19  additional_comments                   13 non-null     object
dtypes: int64(1), object(19)
memory usage: 2.3+ KB
```

```
In [5]: # Get descriptive statistics for numeric columns (if any)
data.describe()
```

Out[5]:

	transition_difficulty
count	14.000000
mean	3.142857
std	0.662994
min	2.000000
25%	3.000000
50%	3.000000
75%	3.000000
max	5.000000

```
In [6]: # Check unique values in categorical columns
for column in data.columns:
    print(f"{column}: {data[column].unique()}")
```

```

grade_level_teach: ['Secondary' 'Pre- Primary' 'Primary']
adaptation_level: ['Moderately' 'With difficulty' 'Easily']
curriculum_coverage: ['Effectively' 'Neutral' 'Very effectively']
student_engagement: ['Lower' 'About the same' 'Much higher']
learning_outcomes: ['No significant change' 'Greatly decreased' 'Slightly improved']
technology_used: ['Teams' 'Zoom']
tech_issues: ['Internet connectivity' 'Software problems' 'Hardware issues']
admin_support: ['Well supported' 'Neutral']
transition_difficulty: [3 5 4 2]
stress_impact: ['Less affected' 'More affected' 'No change' '']
one_on_one_discussions: ['Once a week' 'Twice a month' 'Daily' 'Never']
parent_support: ['Parents have played a crucial role in supporting remote learning by managing schedules and creating conducive learning environment.'
'Parents were encouraging their children to participate when questions were raised and in activities and helped in home work '
'I got 50%parents support ' 'Parents helped their level a lot.'
'Not all the parents are supportive to the children, each parent have different difficulty in their children's remote learning.'
'Well supportive '
'Parents were cooperative, especially slow learner's parents. " 'Great'
'Neutral support' 'Some were supportable and some didn't'
'Parents well supported' 'No cooperation from all the parents '
'Parents who were helpful in helping children effectively ' 'Moderately']
additional_support: ['To improve my online teaching, I would benefit from more advanced digital tools for interactive learning '
'Teachers to be trained on usage of tools to make online teaching easier and interesting.'
'My colleague support ' 'More Technical knowledge.'
'No additional support, in every aspects we find difficulty.\n'
'Ambient environment '
'More teaching tools would probably help us to to improve like access to online resources .'
'Technical knowledge ' 'Proper technology use, and device '
'Parent's involvement" ' Connectivity with students individually'
'Teaching aids'
'Slides based on the contents along with all necessary resource materials can be provided '
'Iscribe pad']
engagement_strategies: ['To keep the students engaged in online, I used interactive tools like quizzes ,encouraged active participation through discussions and group activities. Regular feedback and personalized support also helped maintain their interest'
'I would be calling names who are distracted and put questions or ask them to continue from the last line i read'
'Some special mudras and finger games used in my class.'
'Will Interact often.'
'Using pictures,power point presentation multiple choice questioning method, sending the materials in pdf format, etc.,'
'Activities '
'To keep them engaged, I taught the topics in PowerPoint rather than using textbooks, some online worksheets and debates which made them more interesting. And the end of the class, I asked them how they felt (a kind of feedback).
'
'Discussion on current affairs '
'Mock test, games related to the subject '
'Making the class interesting and excited with some jokes in between'

```

```

'Keep asking questions' 'Quiz time'
'During the time of instruction we show them demo video to keep them engaged
'
' I ask to show the workout answers in cam itself ']
method_change: ['Yes, I significantly adapted my teaching methods for online
classes. Incorporated more multimedia resources ,interactive tools to accommo
date students attention span .Regular feedback and flexibility became essenti
al'
'Yes' 'Yes,extra interaction of students.' 'Yes I have changed.'
'Yes, observing and learning is different in every catagory. Like just liste
ning is one way and other way is observing and learning,, '
'yes ' 'Not really more, indeed sometimes! ' 'Yes ofcourse' 'Yea' 'Yes.']]
tech_access: ['Yes ,I have reliable access to the necessary technology'
'At first no, then later made necessary arrangements ' 'We had' 'Yes'
'Yes, not at full support.' 'yes' 'Yes, but faced issues sometimes'
'Not up to the level']
socioeconomic_impact: ['Students from lower income families often faced chall
enges such as limited access to devices, poor internet connectivity a, and la
ck of a conducive learning environment, which affected their engagement and
learning outcomes'
'Parents argument and forcing child to answer first.'
'70%students affected '
'Nowadays all are having smart phone so no major effect.'
'During pandemic, economically some families are weak and they are not affor
dable to provide gadgets to their children.'
'It has affected. But with growing and competitive prices of the service pro
vider, it is getting lesser '
'Students from low economic background severely affected indeed it was chall
enging for the teachers to make them participate in the sessions. '
'Financial restrictions for getting gadgets and network connectivity '
'Yes, concentration while using the gadgets'
'They are unable to concentrate because of noises surrounded by them, no sep
erate rooms for them, not having mobiles.'
'In many ways as per their financial ability'
'Lack of online facilities '
'Positive learning environment like family background, financial support, qu
ality of internet connection play an important role making the learning effec
tively '
"It's true. Siblings of different grades were more affect due to lack of gad
gets."]
future_practices: ['I will integrate more digital tools and online resources
into my teaching.'
'I will be more flexible and adaptable in my teaching strategies.']]
sustainability_belief: ['Yes, but only as a supplement to traditional classro
om teaching.'
'No, it is not sustainable and should not replace traditional teaching.'
'It is sustainable in certain contexts, but not universally.']]
additional_comments: ['Online education in India has had a mixed impact. Whil
e it provide continuity in learning, technological challenges like poor inter
net, lack of access to devices ,and unfamiliarity with digital platforms were
significant barriers, especially for students from rural or low income group.
Additionally ,the absence of face -to -face interaction affected student enga
gement and motivation.\nto improve ,there should be access to affordable inte
rnet and devices for students, \n increased teac
her training in pedagogy and user friendly platforms that cater to socioecon
omic needs'
'Parents were complaining that their child was not encouraged, questioned

```

always only few students were focused.'

'The biggest advantage is students can record all classes and then refer later.\nTeachers learning capacity increase in technology side(not only subject knowledge).'

'Need concentration in students side.'

'During pandemic, education through online is new and it is difficult to carry to the economically poor children, and poor network connection in the remote areas found very difficult. Online teaching is costlier for the vulnerable weaker section of the society.\n\n\n'

'The online education has impacted a lot. With growing advent of AI, the role of effective teaching is diminishing. I would recommend for some effective techniques which will be a touch of traditional teaching too'

'Many teachers lack the confidence and training to use technology in their teaching. My first and foremost suggestions for the teachers from non technical background need to be trained first by the experts and have the access to the technology. To improve active learning that help students retain and transfer skills and knowledge. \n '

'Updating technical knowledge. ' nan 'Parents monitor on children'

'Technical issues. . students deviated in other apps.. . '

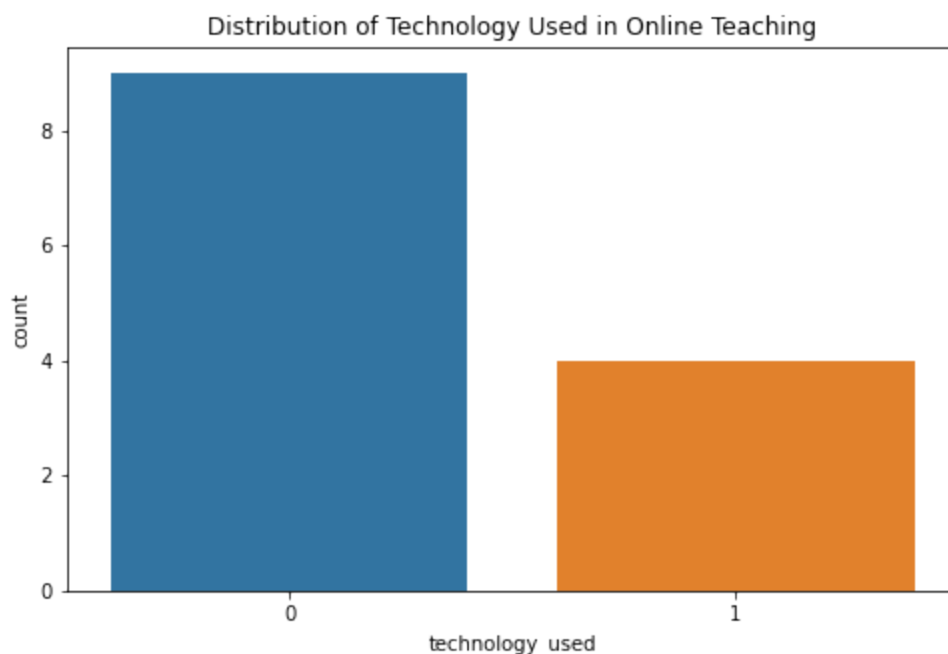
'Hands on learning will be more effective '

'Intermittent physical classes can be given to the online students to have an effective learning experience '

'Online teaching during pandemic is very new one for most of the teacher And for students also, they have to use to it.so we can conduct online classes during rainy holiday or other unplanned holidays. ']

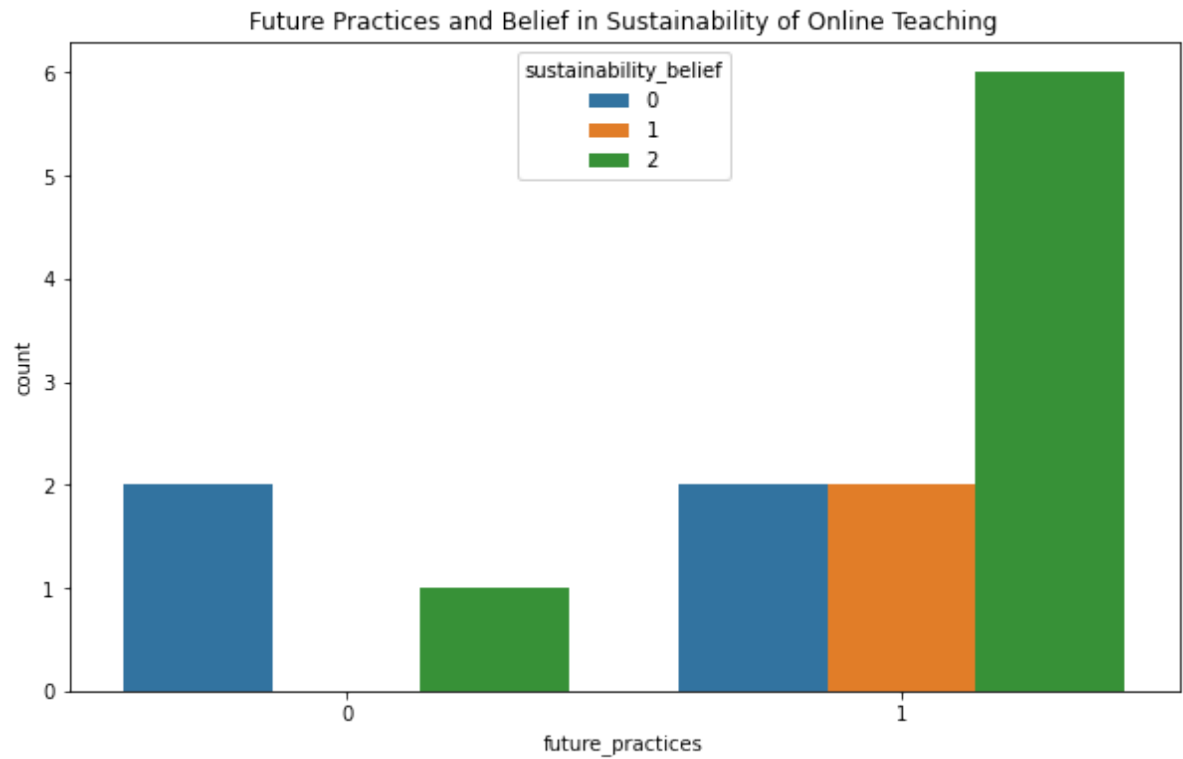
### a) Distribution of Technology Used

```
In [20]: # Plot the distribution of technology used
plt.figure(figsize=(8, 5))
sns.countplot(x='technology_used', data=data_cleaned)
plt.title('Distribution of Technology Used in Online Teaching')
plt.show()
```



**b) Future Practices in Online Teaching**

```
In [21]: # Plot future practices and sustainability belief
plt.figure(figsize=(10, 6))
sns.countplot(x='future_practices', hue='sustainability_belief', data=data_clean)
plt.title('Future Practices and Belief in Sustainability of Online Teaching')
plt.show()
```



## Data Cleaning and Preprocessing

```
In [7]: # Check for missing values
missing_values = data.isnull().sum()

# Drop columns with excessive missing values or irrelevant to the analysis (if
data_cleaned = data.dropna()

# Convert categorical variables to numeric using Label Encoding or One-Hot Encod

encoder = LabelEncoder()
for column in data_cleaned.columns:
    if data_cleaned[column].dtype == 'object':
        data_cleaned[column] = encoder.fit_transform(data_cleaned[column])

# View cleaned data
data_cleaned.head()
```

C:\Users\n\AppData\Local\Temp\ipykernel\_10676\518156326.py:13: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.  
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy))

```
data_cleaned[column] = encoder.fit_transform(data_cleaned[column])
```

Out[7]:

	grade_level_teach	adaptation_level	curriculum_coverage	student_engagement	learning_outcor
0	2	1	0	1	
1	0	2	1	1	
2	2	1	1	1	
3	2	1	1	1	
4	2	2	1	1	



## Exploratory Data Analysis (EDA)

In [60]: `data_cleaned.describe()`

Out[60]:

	grade_level_teach	adaptation_level	curriculum_coverage	student_engagement	learning_outcomes
count	13.000000	13.000000	13.000000	13.000000	13.000000
mean	1.692308	1.000000	0.615385	1.076923	1.000000
std	0.630425	0.707107	0.650444	0.493548	0.500000
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.000000	1.000000	0.000000	1.000000	1.000000
50%	2.000000	1.000000	1.000000	1.000000	1.000000
75%	2.000000	1.000000	1.000000	1.000000	2.000000
max	2.000000	2.000000	2.000000	2.000000	2.000000

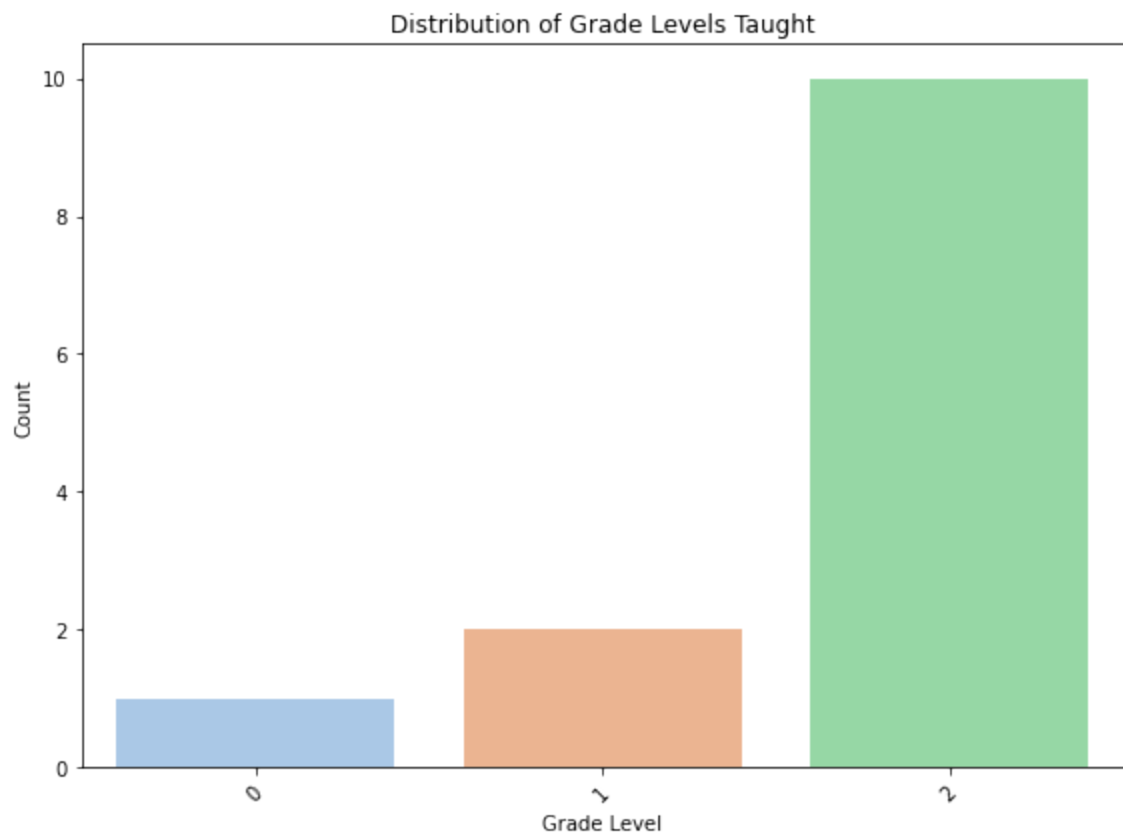
In [53]: `#Get the summary statistics`  
`data_cleaned.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13 entries, 0 to 13
Data columns (total 20 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   grade_level_teach                    13 non-null    int32
 1   adaptation_level                     13 non-null    int64
 2   curriculum_coverage                  13 non-null    int64
 3   student_engagement                  13 non-null    int64
 4   learning_outcomes                   13 non-null    int64
 5   technology_used                      13 non-null    int32
 6   tech_issues                         13 non-null    int32
 7   admin_support                       13 non-null    int32
 8   transition_difficulty                13 non-null    int64
 9   stress_impact                       13 non-null    int64
10   one_on_one_discussions               13 non-null    int32
11   parent_support                      13 non-null    int64
12   additional_support                   13 non-null    int32
13   engagement_strategies                13 non-null    int32
14   method_change                       13 non-null    int32
15   tech_access                         13 non-null    int64
16   socioeconomic_impact                 13 non-null    int64
17   future_practices                     13 non-null    int32
18   sustainability_belief                 13 non-null    int32
19   additional_comments                  13 non-null    int32
dtypes: int32(11), int64(9)
memory usage: 2.1 KB
```

## Visualizations

### 1: Histogram of Grade Levels

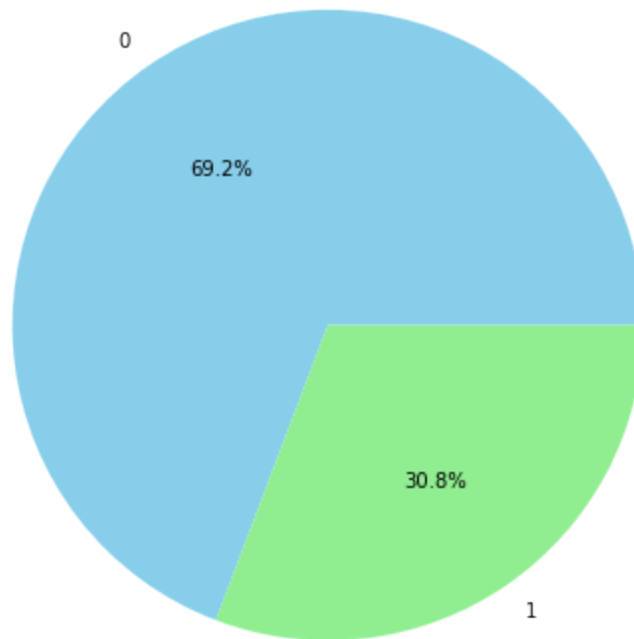
```
In [54]: # Matching the Loaded the dataset
df = data_cleaned
# -----
plt.figure(figsize=(8, 6))
sns.countplot(x='grade_level_teach', data=df, palette='pastel')
plt.title('Distribution of Grade Levels Taught')
plt.xlabel('Grade Level')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



### Pie Chart of Technology Used

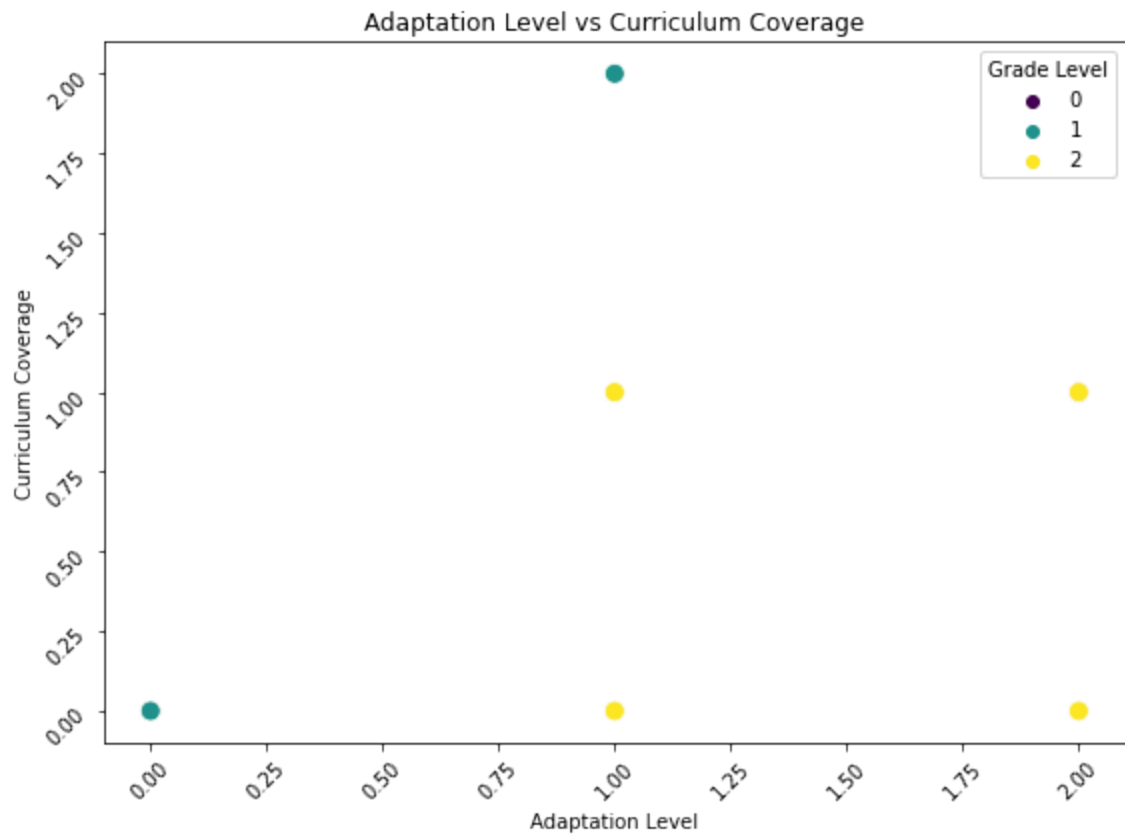
```
In [56]: # ---- ----
tech_counts = df['technology_used'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(tech_counts, labels=tech_counts.index, autopct='%1.1f%%', colors=['skyblue', 'lightgreen'])
plt.title('Technology Platforms Used for Online Teaching')
plt.tight_layout()
plt.show()
```

Technology Platforms Used for Online Teaching



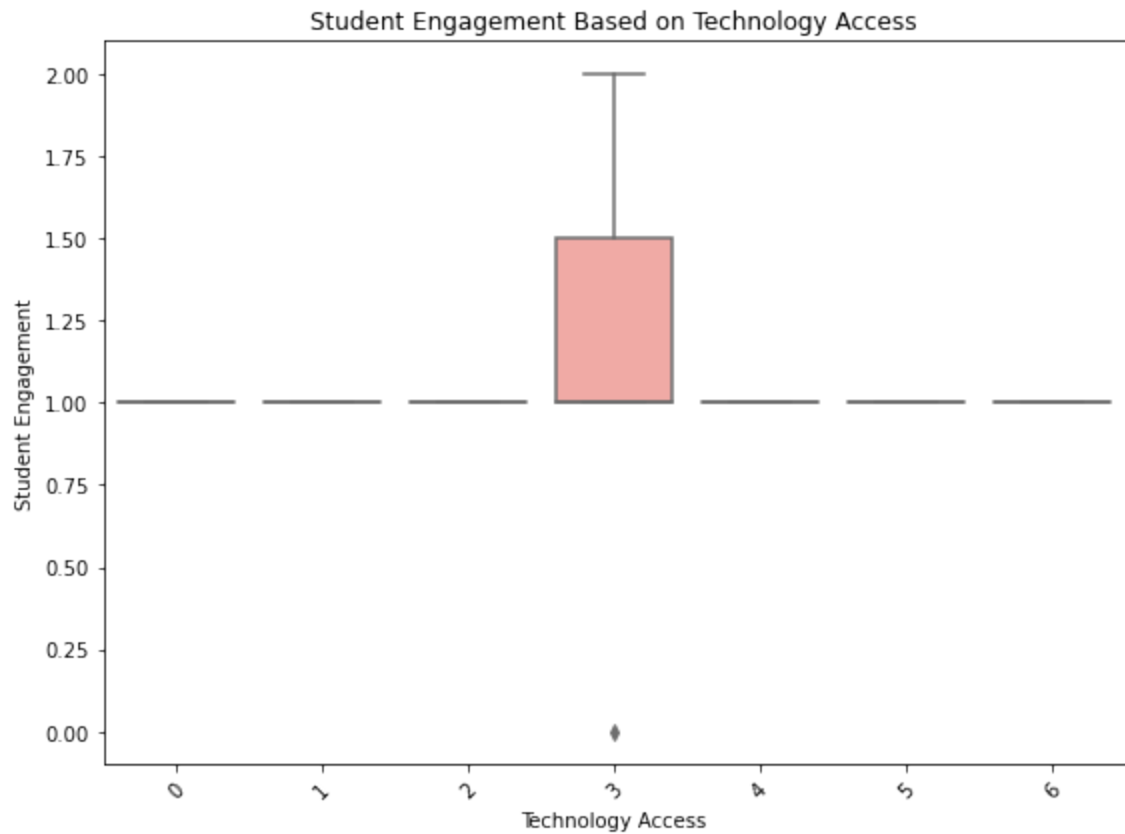
Scatter Plot of Adaptation Level vs Curriculum Coverage

```
In [57]: # ----  
plt.figure(figsize=(8, 6))  
sns.scatterplot(x='adaptation_level', y='curriculum_coverage', hue='grade_level')  
plt.title('Adaptation Level vs Curriculum Coverage')  
plt.xlabel('Adaptation Level')  
plt.ylabel('Curriculum Coverage')  
plt.legend(title='Grade Level')  
plt.xticks(rotation=45)  
plt.yticks(rotation=45)  
plt.tight_layout()  
plt.show()
```



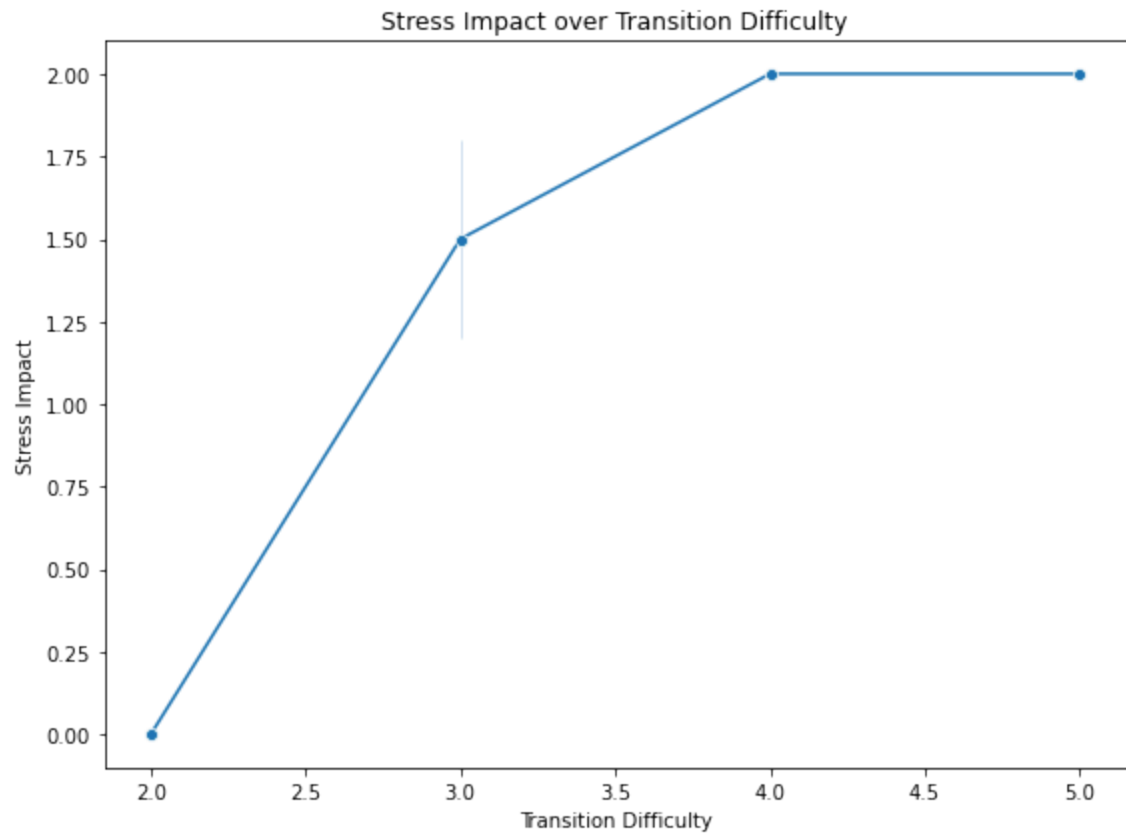
Box Plot of Student Engagement by Tech Access

```
In [58]: # ---- ----
plt.figure(figsize=(8, 6))
sns.boxplot(x='tech_access', y='student_engagement', data=df, palette='pastel')
plt.title('Student Engagement Based on Technology Access')
plt.xlabel('Technology Access')
plt.ylabel('Student Engagement')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Line Plot of Stress Impact over Transition Difficulty

```
In [61]: # ---- ----  
plt.figure(figsize=(8, 6))  
sns.lineplot(x='transition_difficulty', y='stress_impact', data=df, marker='o')  
plt.title('Stress Impact over Transition Difficulty')  
plt.xlabel('Transition Difficulty')  
plt.ylabel('Stress Impact')  
plt.tight_layout()  
plt.show()
```



## Analysis: Key Research Objectives

```
In [28]: # Load the dataset
df = data_cleaned

# Convert categorical variables to numerical using LabelEncoder
le = LabelEncoder()

# Encoding categorical columns
df['tech_access'] = le.fit_transform(df['tech_access'])
df['student_engagement'] = le.fit_transform(df['student_engagement'])
df['socioeconomic_impact'] = le.fit_transform(df['socioeconomic_impact'])
df['learning_outcomes'] = le.fit_transform(df['learning_outcomes'])
df['parent_support'] = le.fit_transform(df['parent_support'])
df['stress_impact'] = le.fit_transform(df['stress_impact'])
df['adaptation_level'] = le.fit_transform(df['adaptation_level'])
df['curriculum_coverage'] = le.fit_transform(df['curriculum_coverage'])
```

## 1. Impact of Technology Access on Student Engagement

```
In [40]: # 1. Impact of Technology Access on Student Engagement
X1 = df[['tech_access']]
y1 = df['student_engagement']
X1 = sm.add_constant(X1)
model1 = sm.OLS(y1, X1).fit()
print("Technology Access on Student Engagement")
print(model1.summary())

# Plot technology access vs. student engagement
plt.figure(figsize=(10, 6))
sns.boxplot(x='tech_access', y='student_engagement', data=data_cleaned)
plt.title('Impact of Technology Access on Student Engagement')
plt.show()
```



## Technology Access on Student Engagement

## OLS Regression Results

```

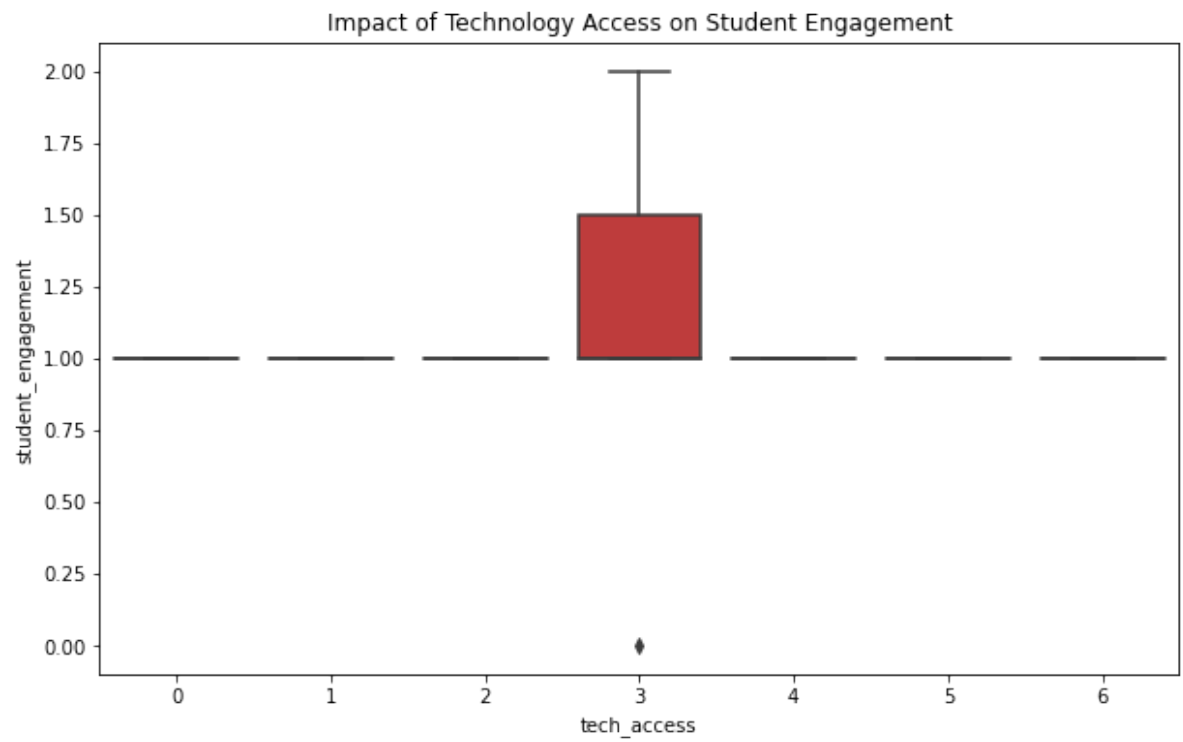
=====
=
Dep. Variable:      student_engagement    R-squared:                -0.00
0
Model:                OLS    Adj. R-squared:                -0.09
1
Method:                Least Squares    F-statistic:                -1.671e-1
5
Date:                Sat, 05 Oct 2024    Prob (F-statistic):                1.0
0
Time:                20:58:58    Log-Likelihood:                -8.746
2
No. Observations:                13    AIC:                21.4
9
Df Residuals:                11    BIC:                22.6
2
Df Model:                1
Covariance Type:                nonrobust
=====
==
               coef      std err          t      P>|t|      [0.025      0.97
5]
-----
--
const           1.0769        0.325        3.310      0.007        0.361        1.7
93
tech_access  1.943e-16        0.097    1.99e-15      1.000       -0.214        0.2
14
=====
=
Omnibus:                3.444    Durbin-Watson:                2.05
3
Prob(Omnibus):                0.179    Jarque-Bera (JB):                0.97
0
Skew:                0.231    Prob(JB):                0.61
6
Kurtosis:                4.256    Cond. No.                8.1
6
=====
=

```

## Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

C:\Users\n\anaconda3\lib\site-packages\scipy\stats\stats.py:1541: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=13  
 warnings.warn("kurtosistest only valid for n>=20 ... continuing ")



## 2. Role of Technology Used in Engagement

```
In [30]: # 2. Role of Technology Used in Engagement
X2 = df[['technology_used']]
y2 = df['student_engagement']
X2 = sm.add_constant(X2)
model2 = sm.OLS(y2, X2).fit()
print("Technology Used on Student Engagement")
print(model2.summary())
```

## Technology Used on Student Engagement

## OLS Regression Results

```

=====
=
Dep. Variable:      student_engagement    R-squared:                0.05
9
Model:                OLS    Adj. R-squared:            -0.02
6
Method:              Least Squares    F-statistic:            0.692
3
Date:                Sat, 05 Oct 2024    Prob (F-statistic):      0.42
3
Time:                20:14:36    Log-Likelihood:        -8.349
4
No. Observations:      13    AIC:                    20.7
0
Df Residuals:          11    BIC:                    21.8
3
Df Model:                1
Covariance Type:      nonrobust
=====

```

```

=====
=====
              coef    std err          t      P>|t|      [0.025
0.975]
-----
-----
const              1.0000      0.167      6.000      0.000      0.633
1.367
technology_used    0.2500      0.300      0.832      0.423     -0.411
0.911
=====

```

```

=
Omnibus:                2.977    Durbin-Watson:            2.31
8
Prob(Omnibus):          0.226    Jarque-Bera (JB):        0.73
4
Skew:                   0.296    Prob(JB):                0.69
3
Kurtosis:               4.002    Cond. No.                2.4
2
=====
=

```

## Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```

C:\Users\n\anaconda3\lib\site-packages\scipy\stats\stats.py:1541: UserWarning:
kurtosistest only valid for n>=20 ... continuing anyway, n=13
  warnings.warn("kurtosistest only valid for n>=20 ... continuing ")

```

### 3. Influence of Socioeconomic Status on Learning Outcomes

```
In [41]: # 3. Influence of Socioeconomic Status on Learning Outcomes
X3 = df[['socioeconomic_impact']]
y3 = df['learning_outcomes']
X3 = sm.add_constant(X3)
model3 = sm.OLS(y3, X3).fit()
print("Socioeconomic Status on Learning Outcomes")
print(model3.summary())
```

## Socioeconomic Status on Learning Outcomes

## OLS Regression Results

```

=====
=
Dep. Variable:      learning_outcomes  R-squared:                0.01
1
Model:              OLS  Adj. R-squared:            -0.07
8
Method:             Least Squares  F-statistic:              0.127
2
Date:               Sat, 05 Oct 2024  Prob (F-statistic):        0.72
8
Time:               21:00:07  Log-Likelihood:          -14.96
1
No. Observations:   13  AIC:                    33.9
2
Df Residuals:       11  BIC:                    35.0
5
Df Model:           1
Covariance Type:    nonrobust
=====
=====

```

```

=====
=====
              coef      std err          t      P>|t|      [0.025
0.975]
-----
-----
const              1.2857      0.436      2.950      0.013      0.327
2.245
socioeconomic_impact -0.0220      0.062     -0.357      0.728     -0.158
0.114
=====
=

```

```

Omnibus:              1.738  Durbin-Watson:              2.48
3
Prob(Omnibus):         0.419  Jarque-Bera (JB):              0.98
2
Skew:                  -0.309  Prob(JB):                      0.61
2
Kurtosis:              1.804  Cond. No.                      13.
6
=====
=

```

## Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```

C:\Users\n\anaconda3\lib\site-packages\scipy\stats\stats.py:1541: UserWarning:
kurtosistest only valid for n>=20 ... continuing anyway, n=13
  warnings.warn("kurtosistest only valid for n>=20 ... continuing ")

```

## 4. Parent Support vs Student Engagement

```
In [42]: # 4. Parent Support vs Student Engagement
X4 = df[['parent_support']]
y4 = df['student_engagement']
X4 = sm.add_constant(X4)
model4 = sm.OLS(y4, X4).fit()
print("Parent Support vs Student Engagement")
print(model4.summary())

# Plot parent support vs. student engagement
plt.figure(figsize=(10, 6))
sns.boxplot(x='parent_support', y='student_engagement', data=data_cleaned)
plt.title('Parent Support and Student Engagement')
plt.show()
```

## Parent Support vs Student Engagement

## OLS Regression Results

```

=====
=
Dep. Variable:      student_engagement    R-squared:                0.27
1
Model:                OLS    Adj. R-squared:                0.20
4
Method:                Least Squares    F-statistic:                4.08
2
Date:                Sat, 05 Oct 2024    Prob (F-statistic):        0.068
4
Time:                21:00:15    Log-Likelihood:            -6.694
5
No. Observations:        13    AIC:                        17.3
9
Df Residuals:            11    BIC:                        18.5
2
Df Model:                1
Covariance Type:        nonrobust
=====
=====

```

```

=====
=====
              coef      std err          t      P>|t|      [0.025
0.975]
-----
-----
const          0.6813      0.231      2.953      0.013      0.173
1.189
parent_support  0.0659      0.033      2.021      0.068     -0.006
0.138
=====
=====

```

```

=
Omnibus:                0.869    Durbin-Watson:                1.77
0
Prob(Omnibus):          0.648    Jarque-Bera (JB):                0.44
0
Skew:                   0.431    Prob(JB):                        0.80
3
Kurtosis:               2.734    Cond. No.                        13.
6
=====
=

```

## Notes:

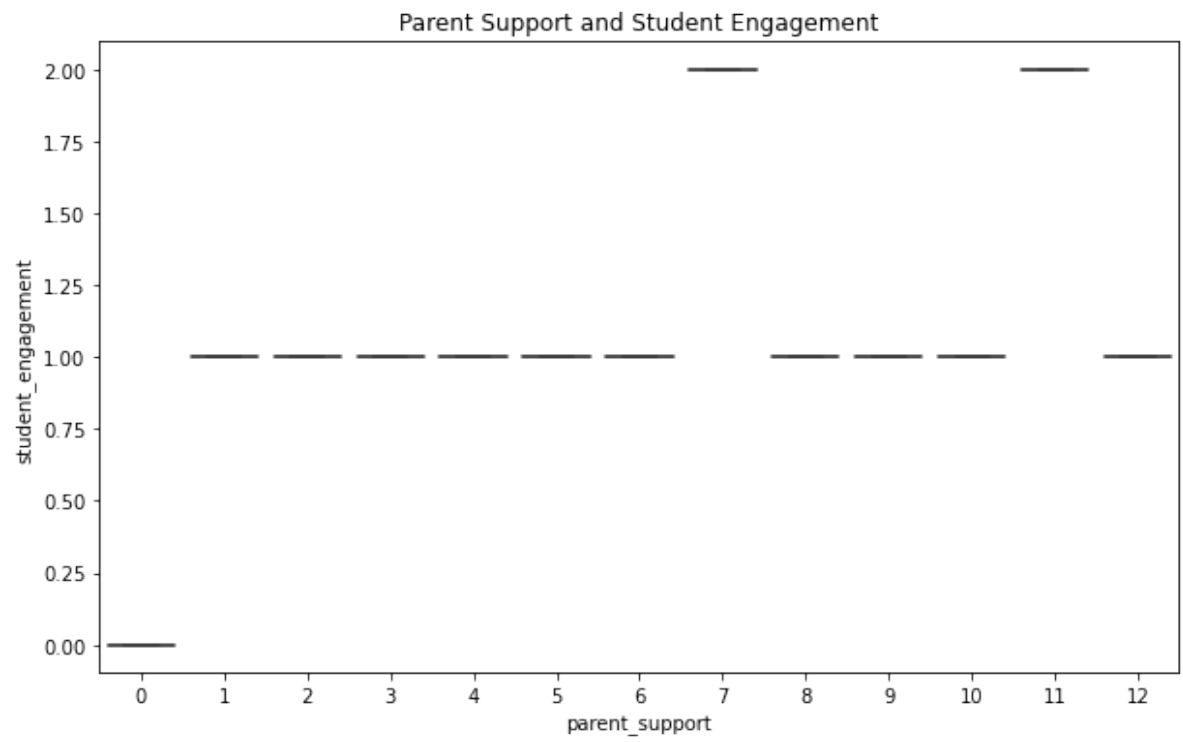
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```

C:\Users\n\anaconda3\lib\site-packages\scipy\stats\stats.py:1541: UserWarning:
kurtosistest only valid for n>=20 ... continuing anyway, n=13
  warnings.warn("kurtosistest only valid for n>=20 ... continuing ")

```





## 5 Stress Impact on Teacher Adaptation and Curriculum Coverage

```
In [43]: # 5. Stress Impact on Teacher Adaptation and Curriculum Coverage
X5 = df[['stress_impact']]
y5_1 = df['adaptation_level']
y5_2 = df['curriculum_coverage']

X5 = sm.add_constant(X5)
model5_1 = sm.OLS(y5_1, X5).fit()
model5_2 = sm.OLS(y5_2, X5).fit()

print("Stress Impact on Teacher Adaptation")
print(model5_1.summary())
print("Stress Impact on Curriculum Coverage")
print(model5_2.summary())
```

## Stress Impact on Teacher Adaptation

## OLS Regression Results

```
=====
=
Dep. Variable:      adaptation_level    R-squared:                0.12
7
Model:                OLS    Adj. R-squared:                0.04
8
Method:              Least Squares    F-statistic:              1.60
7
Date:                Sat, 05 Oct 2024    Prob (F-statistic):      0.23
1
Time:                21:00:30    Log-Likelihood:          -12.53
4
No. Observations:      13    AIC:                      29.0
7
Df Residuals:          11    BIC:                      30.2
0
Df Model:                1
Covariance Type:      nonrobust
=====
```

```
=====
====
              coef    std err          t      P>|t|      [0.025    0.
975]
-----
----
const          1.5588      0.481      3.244      0.008      0.501
2.617
stress_impact  -0.3824      0.302     -1.268      0.231     -1.046
0.282
=====
```

```
=====
=
Omnibus:                0.041    Durbin-Watson:              2.30
1
Prob(Omnibus):          0.979    Jarque-Bera (JB):            0.12
9
Skew:                  -0.064    Prob(JB):                    0.93
8
Kurtosis:              2.530    Cond. No.                     5.3
9
=====
```

## Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

## Stress Impact on Curriculum Coverage

## OLS Regression Results

```
=====
==
Dep. Variable:      curriculum_coverage    R-squared:                0.0
64
Model:                OLS    Adj. R-squared:            -0.0
21
Method:              Least Squares    F-statistic:              0.75
71
Date:                Sat, 05 Oct 2024    Prob (F-statistic):      0.4
```

```

03
Time:                21:00:30    Log-Likelihood:            -11.9
02
No. Observations:    13    AIC:                27.
80
Df Residuals:        11    BIC:                28.
93
Df Model:            1
Covariance Type:    nonrobust
=====
====
              coef      std err          t      P>|t|      [0.025      0.
975]
-----
----
const          0.2500      0.458      0.546      0.596      -0.758
1.258
stress_impact   0.2500      0.287      0.870      0.403      -0.382
0.882
=====
=
Omnibus:                0.535    Durbin-Watson:            1.78
9
Prob(Omnibus):          0.765    Jarque-Bera (JB):            0.58
5
Skew:                   0.294    Prob(JB):                  0.74
6
Kurtosis:               2.143    Cond. No.                  5.3
9
=====
=

```

#### Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```

C:\Users\n\anaconda3\lib\site-packages\scipy\stats\stats.py:1541: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=13
  warnings.warn("kurtosistest only valid for n>=20 ... continuing ")
C:\Users\n\anaconda3\lib\site-packages\scipy\stats\stats.py:1541: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=13
  warnings.warn("kurtosistest only valid for n>=20 ... continuing ")

```

## 6. Socioeconomic Challenges and Learning Outcomes

```
In [44]: # 6. Socioeconomic Challenges and Learning Outcomes
X6 = df[['socioeconomic_impact']]
y6 = df['learning_outcomes']
X6 = sm.add_constant(X6)
model6 = sm.OLS(y6, X6).fit()
print("Socioeconomic Challenges and Learning Outcomes")
print(model6.summary())

# Plot socioeconomic impact vs. Learning outcomes
plt.figure(figsize=(10, 6))
sns.scatterplot(x='socioeconomic_impact', y='learning_outcomes', data=data_clean)
plt.title('Socioeconomic Impact on Learning Outcomes')
plt.show()
```

## Socioeconomic Challenges and Learning Outcomes

## OLS Regression Results

```
=====
=
Dep. Variable:      learning_outcomes    R-squared:                0.01
1
Model:              OLS    Adj. R-squared:            -0.07
8
Method:             Least Squares    F-statistic:              0.127
2
Date:               Sat, 05 Oct 2024    Prob (F-statistic):       0.72
8
Time:               21:00:38    Log-Likelihood:          -14.96
1
No. Observations:   13    AIC:                    33.9
2
Df Residuals:       11    BIC:                    35.0
5
Df Model:            1
Covariance Type:    nonrobust
=====
```

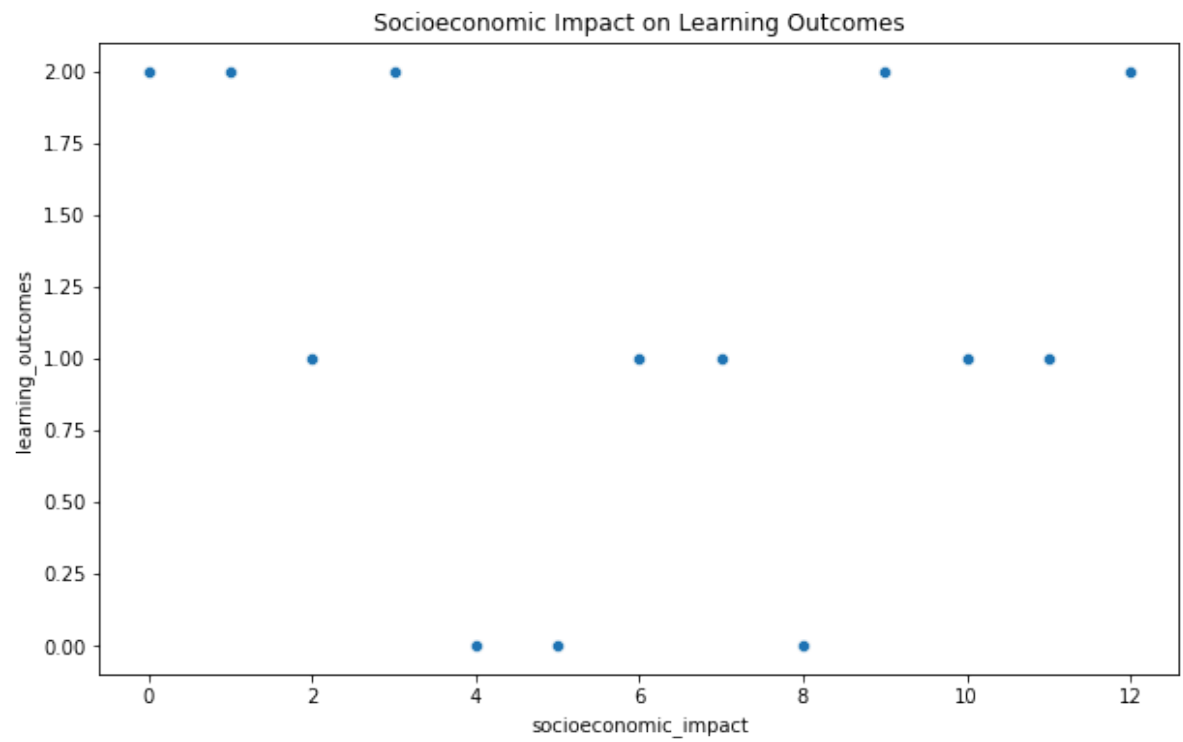
```
=====
=====
                                coef      std err          t      P>|t|      [0.025
0.975]
-----
-----
const                1.2857        0.436        2.950      0.013      0.327
2.245
socioeconomic_impact -0.0220        0.062       -0.357      0.728     -0.158
0.114
=====
```

```
=====
=
Omnibus:              1.738    Durbin-Watson:              2.48
3
Prob(Omnibus):        0.419    Jarque-Bera (JB):            0.98
2
Skew:                 -0.309    Prob(JB):                    0.61
2
Kurtosis:              1.804    Cond. No.                     13.
6
=====
=
```

## Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
C:\Users\n\anaconda3\lib\site-packages\scipy\stats\stats.py:1541: UserWarning:
kurtosistest only valid for n>=20 ... continuing anyway, n=13
warnings.warn("kurtosistest only valid for n>=20 ... continuing ")
```



## 7. Influence of Stress and Parental Support on Teachers' Adaptation and Student Engagement

```
In [45]: # 7. Influence of Stress and Parental Support on Teachers' Adaptation and Student Engagement
X7 = df[['stress_impact', 'parent_support']]
y7_1 = df['adaptation_level']
y7_2 = df['student_engagement']

X7 = sm.add_constant(X7)
model7_1 = sm.OLS(y7_1, X7).fit()
model7_2 = sm.OLS(y7_2, X7).fit()

print("Stress and Parental Support on Teachers' Adaptation")
print(model7_1.summary())
print("Stress and Parental Support on Student Engagement")
print(model7_2.summary())
```



### Stress and Parental Support on Teachers' Adaptation OLS Regression Results

```
=====
=
Dep. Variable:      adaptation_level    R-squared:                0.13
5
Model:              OLS                Adj. R-squared:          -0.03
7
Method:            Least Squares       F-statistic:             0.783
2
Date:              Sat, 05 Oct 2024    Prob (F-statistic):      0.48
3
Time:              21:02:03           Log-Likelihood:          -12.47
5
No. Observations:      13             AIC:                    30.9
5
Df Residuals:         10             BIC:                    32.6
4
Df Model:              2
Covariance Type:      nonrobust
=====
```

```
=====
=====
              coef      std err          t      P>|t|      [0.025
0.975]
-----
-----
const          1.4832      0.560        2.648      0.024      0.235
2.731
stress_impact  -0.3981      0.319       -1.247      0.241     -1.109
0.313
parent_support  0.0164      0.054        0.304      0.768     -0.104
0.137
=====
```

```
=====
=
Omnibus:          0.068    Durbin-Watson:          2.25
2
Prob(Omnibus):    0.967    Jarque-Bera (JB):          0.19
0
Skew:             -0.129    Prob(JB):                  0.90
9
Kurtosis:         2.467    Cond. No.                  22.
3
=====
=
```

#### Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

### Stress and Parental Support on Student Engagement OLS Regression Results

```
=====
=
Dep. Variable:      student_engagement    R-squared:                0.48
6
Model:              OLS                Adj. R-squared:          0.38
3
Method:            Least Squares       F-statistic:             4.73
```

```

1
Date:                Sat, 05 Oct 2024    Prob (F-statistic):        0.035
8
Time:                21:02:03    Log-Likelihood:        -4.418
0
No. Observations:    13    AIC:                14.8
4
Df Residuals:        10    BIC:                16.5
3
Df Model:            2
Covariance Type:    nonrobust
=====
=====
=====
              coef      std err          t      P>|t|      [0.025
0.975]
-----
-----
const          1.1373      0.301       3.773      0.004      0.466
1.809
stress_impact  -0.3517      0.172      -2.048      0.068     -0.734
0.031
parent_support  0.0756      0.029       2.597      0.027      0.011
0.140
=====
=
Omnibus:                19.275    Durbin-Watson:           1.89
3
Prob(Omnibus):          0.000    Jarque-Bera (JB):        16.54
0
Skew:                   1.928    Prob(JB):                0.00025
6
Kurtosis:               6.959    Cond. No.                22.
3
=====
=

```

## Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

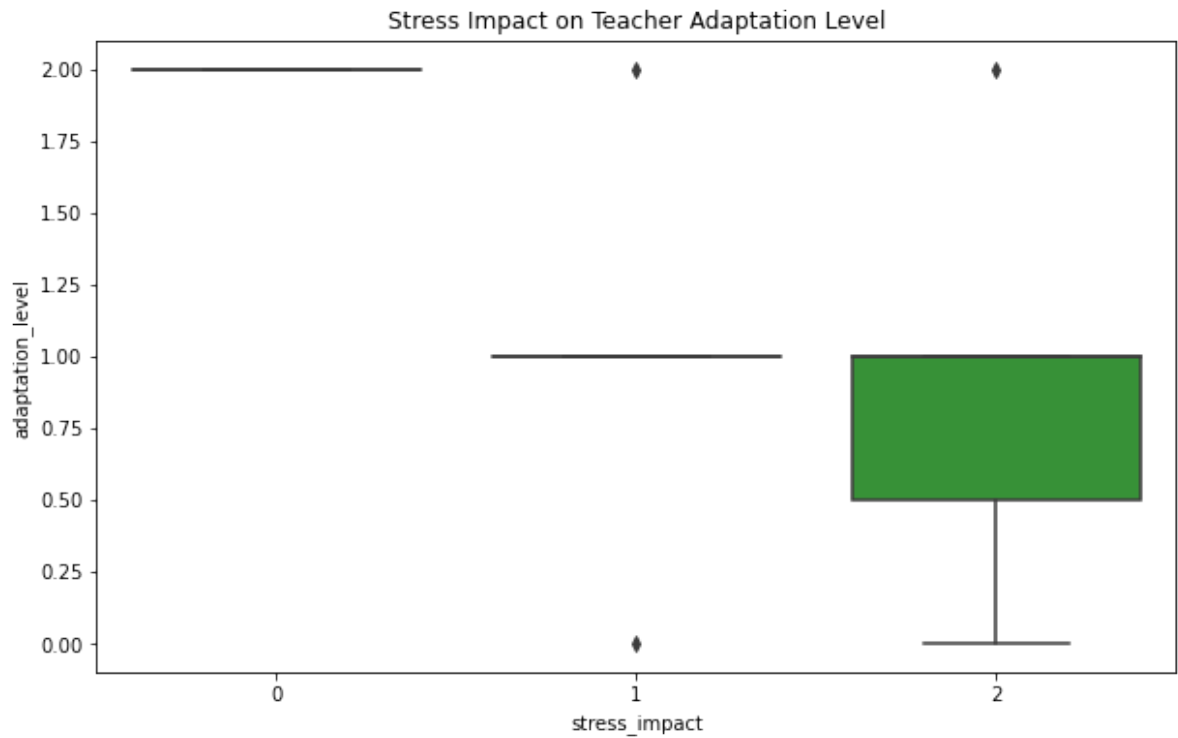
```

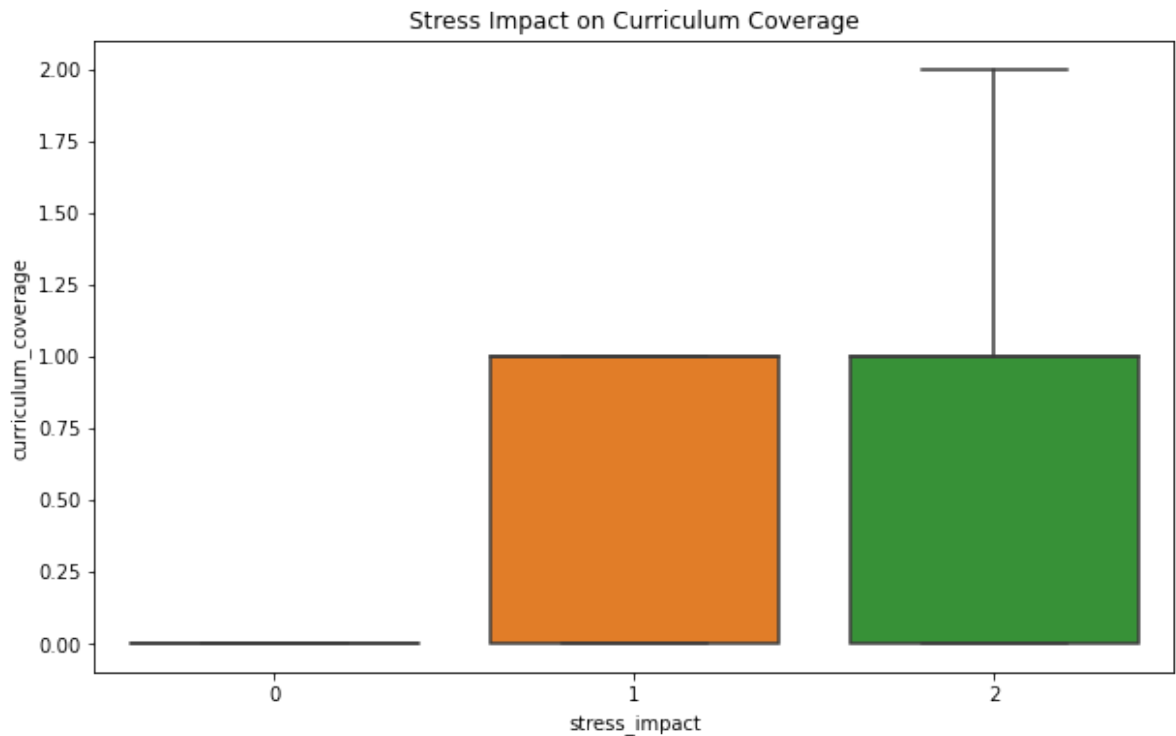
C:\Users\n\anaconda3\lib\site-packages\scipy\stats\stats.py:1541: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=13
  warnings.warn("kurtosistest only valid for n>=20 ... continuing ")
C:\Users\n\anaconda3\lib\site-packages\scipy\stats\stats.py:1541: UserWarning: kurtosistest only valid for n>=20 ... continuing anyway, n=13
  warnings.warn("kurtosistest only valid for n>=20 ... continuing ")

```

```
In [46]: # Plot stress impact vs. adaptation Level
plt.figure(figsize=(10, 6))
sns.boxplot(x='stress_impact', y='adaptation_level', data=data_cleaned)
plt.title('Stress Impact on Teacher Adaptation Level')
plt.show()

# Plot stress impact vs. curriculum coverage
plt.figure(figsize=(10, 6))
sns.boxplot(x='stress_impact', y='curriculum_coverage', data=data_cleaned)
plt.title('Stress Impact on Curriculum Coverage')
plt.show()
```





## Correlation Analysis

```
In [23]: # Compute correlation matrix
correlation_matrix = data_cleaned.corr()
print(correlation_matrix)
```

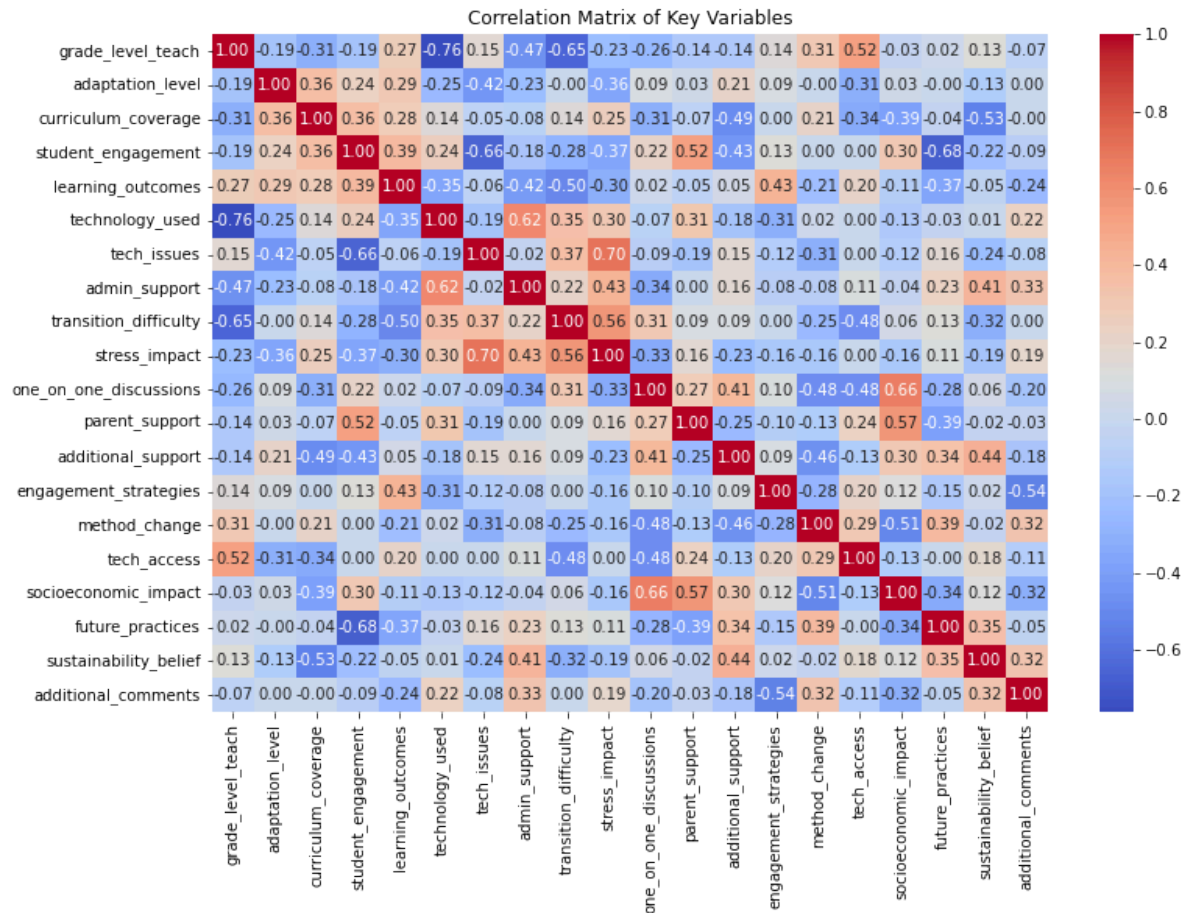
stress_impact	0.450577	5.658859e-01	1.000000e+00
one_on_one_discussions	-0.342940	3.100328e-01	-3.311221e-01
parent_support	0.000000	9.320546e-02	1.620509e-01
additional_support	0.164957	9.320546e-02	-2.268713e-01
engagement_strategies	-0.082479	6.208731e-17	-1.620509e-01
method_change	-0.082593	-2.488929e-01	-1.571513e-01
tech_access	0.105140	-4.752565e-01	3.669513e-17
socioeconomic_impact	-0.041239	6.213698e-02	-1.620509e-01
future_practices	0.225374	1.273429e-01	1.107019e-01
sustainability_belief	0.413223	-3.213525e-01	-1.885670e-01
additional_comments	0.329914	7.013567e-17	1.944611e-01

	one_on_one_discussions	parent_support \
grade_level_teach	-0.258065	-0.135769
adaptation_level	0.093470	0.030261
curriculum_coverage	-0.312653	-0.065795
student_engagement	0.216322	0.520266
learning_outcomes	0.019050	-0.053452
technology_used	-0.074083	0.311805
tech_issues	-0.091654	-0.192879

```
In [24]: # Compute correlation matrix
correlation_matrix = data_cleaned.corr()

# Plot a heatmap of the correlation matrix
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix of Key Variables')
plt.show()
```



# THE END