Evaluating Kinder Readiness using MCLASS and TXKEA

Purpose

Extract, analyze, and present data that will help the Early Childhood Education team understand the extent to which the difference in readiness as measured by the two tests is due to differences in the underlying populations of students taking each (as opposed to differences in test design and scoring). As mentioned above, your response should include the following three components:

1) A SQL query you have written to aggregate and extract necessary data from the database,

2) A data cleaning/analysis script,

3) A brief narrative which describes key findings with data visualizations

SQL Query

```
WITH test_data AS (

SELECT student_id,

district_id,

composite_level,

assessment_edition,

CASE WHEN

composite_level IN

('Well Below Benchmark',

'Below Benchmark') THEN 0

ELSE 1 END

AS passed,

'MCLASS' AS test_taken

FROM MCLASS

WHERE assessment_edition = 'DIB%'

UNION ALL

SELECT student_id,
```

```
district_id,

language,

lit_screening_benchmark,

CASE WHEN

lit_screening_benchmark IN

('Montior',

'Support') THEN 0

ELSE 1 END

AS passed,

'TXKEA' AS test_taken

FROM TXKEA

WHERE language = 'English'

)

SELECT test_data.student_id,

test_data.district_id,

test_data.test_taken,

test_data.passed,

DEMO.ethnicity,

DEMO.eco,

DEMO.spec_ed,

DEMO.el

FROM test_data

LEFT JOIN DEMO

ON

test_data.student_id = DEMO.student_id ;
```

Data Cleaning and Prep

Import the Libraries and Load the dataset

In [1]:
```
# import libraries for data manipulation
import numpy as np
import pandas as pd

# import libraries for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('test_data.csv')
```

```
C:\Users\Cristi Mar\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3444: D
typeWarning: Columns (0) have mixed types.Specify dtype option on import or set low_memo
ry=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

In [3]:
```
df.head() #view the first few rows
```

Out[3]:

	student_id	district_id	test_taken	passed	ethnicity	eco	sped	el
0	885938600	53405.0	TXKEA	1	White	YES	NO	NO
1	871944576	798403.0	TXKEA	1	Black or African American	YES	NO	NO
2	818725252	53405.0	TXKEA	1	White	NO	NO	NO
3	702015143	800409.0	TXKEA	1	White	YES	YES	NO
4	717968813	48403.0	TXKEA	1	Two or more races	YES	NO	NO

Evaluate data for errors and cleaning

In [4]:
```
df.info() # Check the datatypes of columns
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150712 entries, 0 to 150711
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   student_id   150712 non-null  object
 1   district_id  145415 non-null  float64
 2   test_taken   150712 non-null  object
 3   passed       150712 non-null  int64
 4   ethnicity    150712 non-null  object
 5   eco          150712 non-null  object
 6   sped         150712 non-null  object
 7   el           150712 non-null  object
dtypes: float64(1), int64(1), object(6)
memory usage: 9.2+ MB
```

In [5]:
```
df.isnull().sum() #Checking to find null values
```

Out[5]:
```
student_id        0
district_id    5297
```

```
test_taken        0
passed            0
ethnicity         0
eco               0
sped              0
el                0
dtype: int64
```

In [6]:
```
df.describe(include='all').T #Look at the columns for possible outliers
```

Out[6]:

	count	unique	top	freq	mean	std	min	25%	
student_id	150712.0	150025.0	5315199254.0	4.0	NaN	NaN	NaN	NaN	
district_id	145415.0	NaN	NaN	NaN	516617.655696	367590.16558	1403.0	74408.0	77
test_taken	150712	2	TXKEA	94563	NaN	NaN	NaN	NaN	
passed	150712.0	NaN	NaN	NaN	0.665149	0.47194	0.0	0.0	
ethnicity	150712	7	Hispanic/Latino	69047	NaN	NaN	NaN	NaN	
eco	150712	2	YES	89425	NaN	NaN	NaN	NaN	
sped	150712	2	NO	139409	NaN	NaN	NaN	NaN	
el	150712	2	NO	131058	NaN	NaN	NaN	NaN	

In [7]:
```
df['district_id'] = df['district_id'].astype(object)
```

In [8]:
```
df.describe(include='all').T #Recheck values after recasting 'district_id' as object
```

Out[8]:

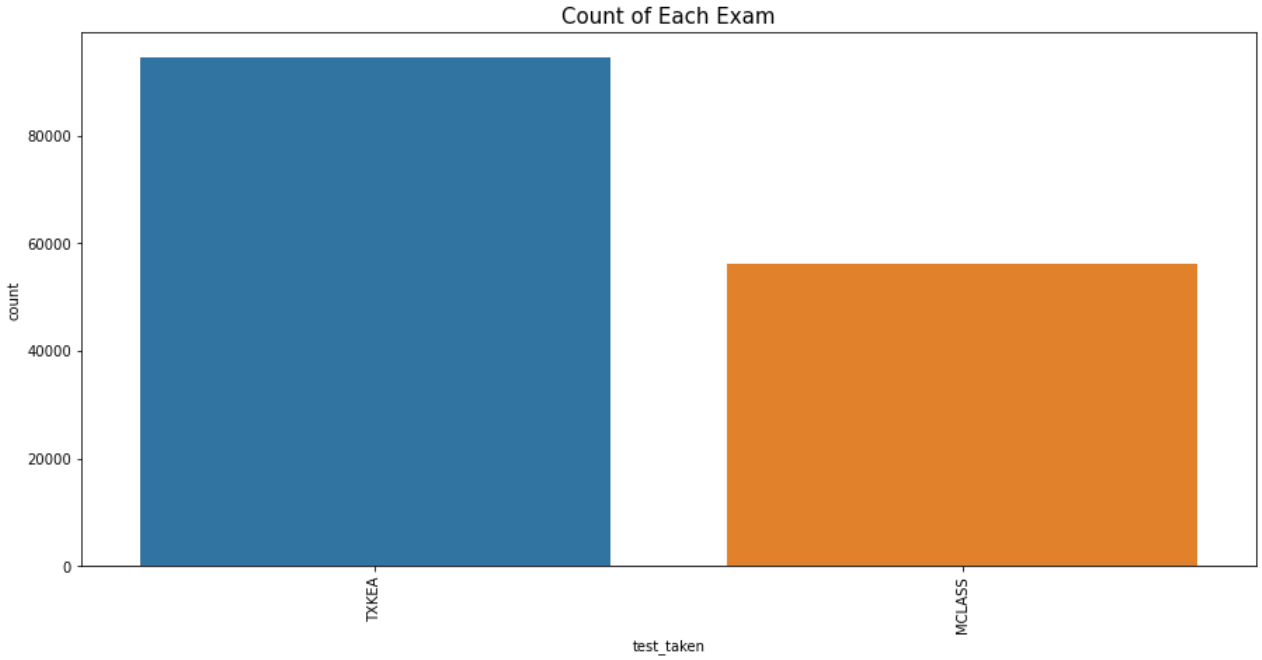
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
student_id	150712.0	150025.0	5315199254.0	4.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
district_id	145415.0	735.0	801102.0	5832.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
test_taken	150712	2	TXKEA	94563	NaN	NaN	NaN	NaN	NaN	NaN	NaN
passed	150712.0	NaN	NaN	NaN	0.665149	0.47194	0.0	0.0	1.0	1.0	1.0
ethnicity	150712	7	Hispanic/Latino	69047	NaN	NaN	NaN	NaN	NaN	NaN	NaN
eco	150712	2	YES	89425	NaN	NaN	NaN	NaN	NaN	NaN	NaN
sped	150712	2	NO	139409	NaN	NaN	NaN	NaN	NaN	NaN	NaN
el	150712	2	NO	131058	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Analysis and Visualizations

Univariate Analysis

In [18]:

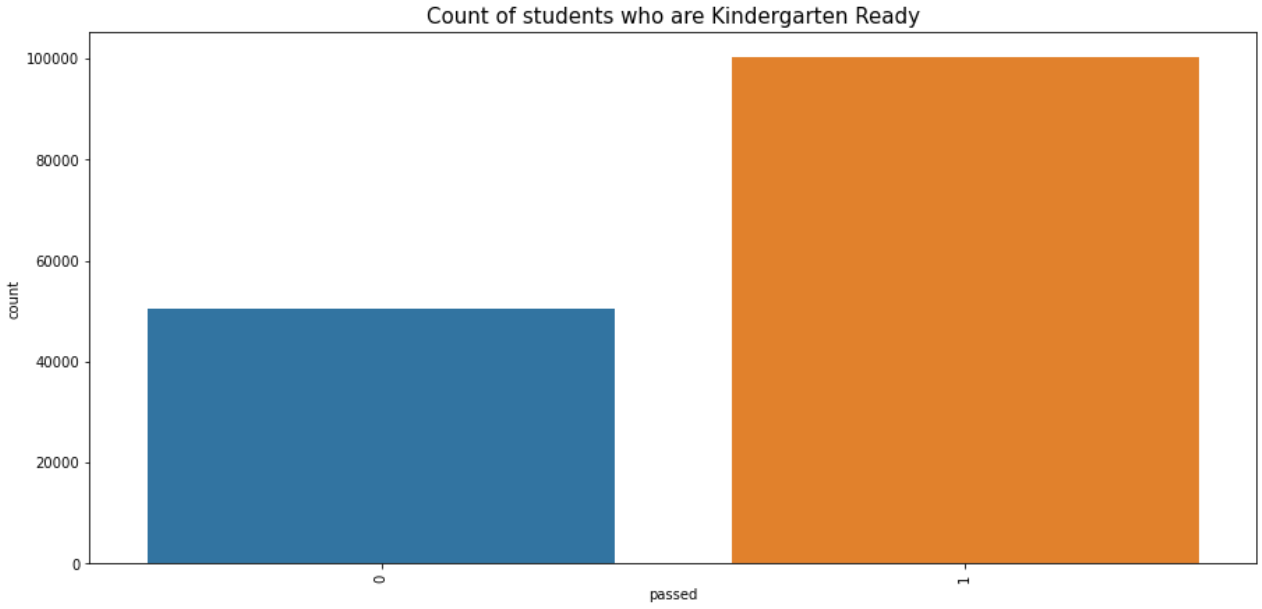
```
plt.figure(figsize = (15, 7))
a=sns.countplot(data=df, x = 'test_taken')
plt.xticks(rotation=90)
a.set_title("Count of Each Exam", fontsize=15)
plt.show()
```



We can see more students took the TXKEA than the MCLASS by about 30000

In [19]:

```
plt.figure(figsize = (15, 7))
b=sns.countplot(data=df, x = 'passed')
plt.xticks(rotation=90)
b.set_title("Count of students who are Kindergarten Ready", fontsize=15)
plt.show()
```

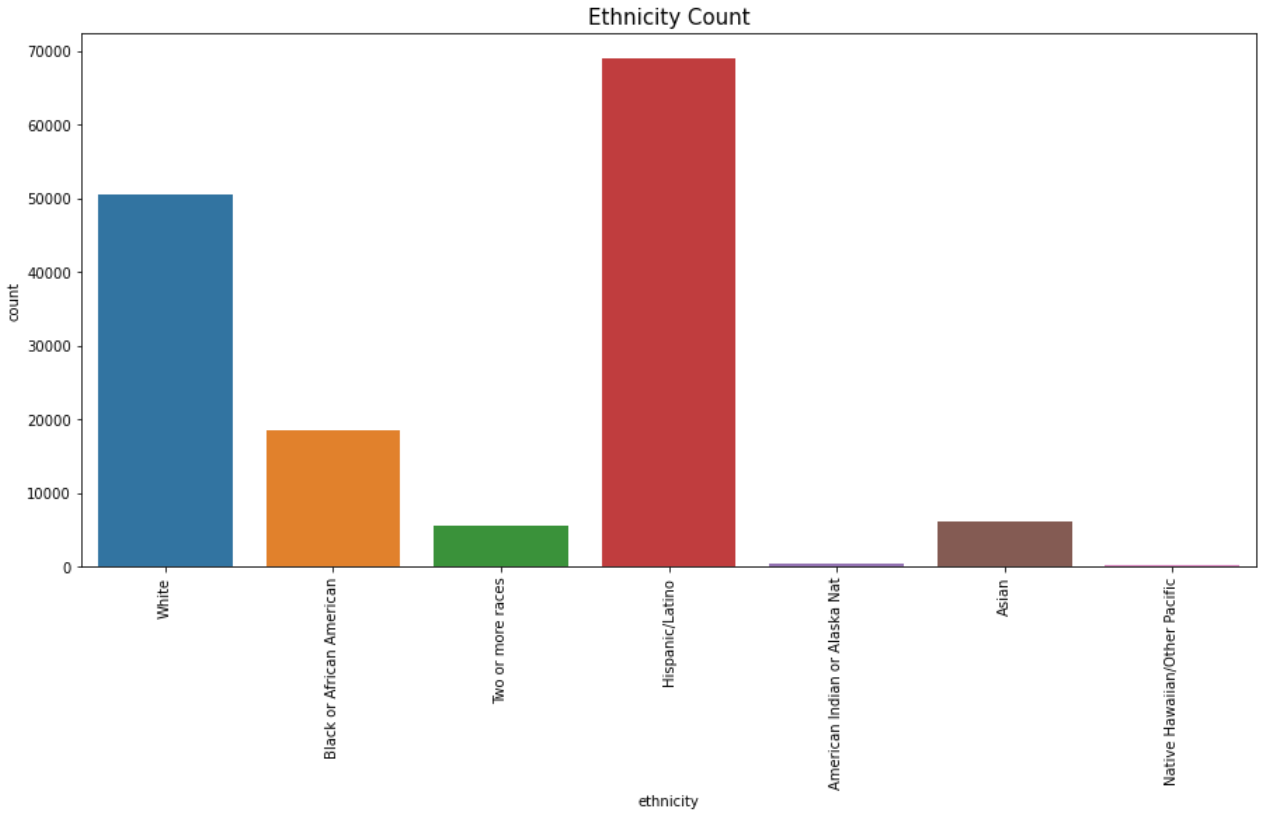


About a 1/3 of the PreK students didn't meet kindergarten readiness.

In [20]:

```
plt.figure(figsize = (15, 7))
```

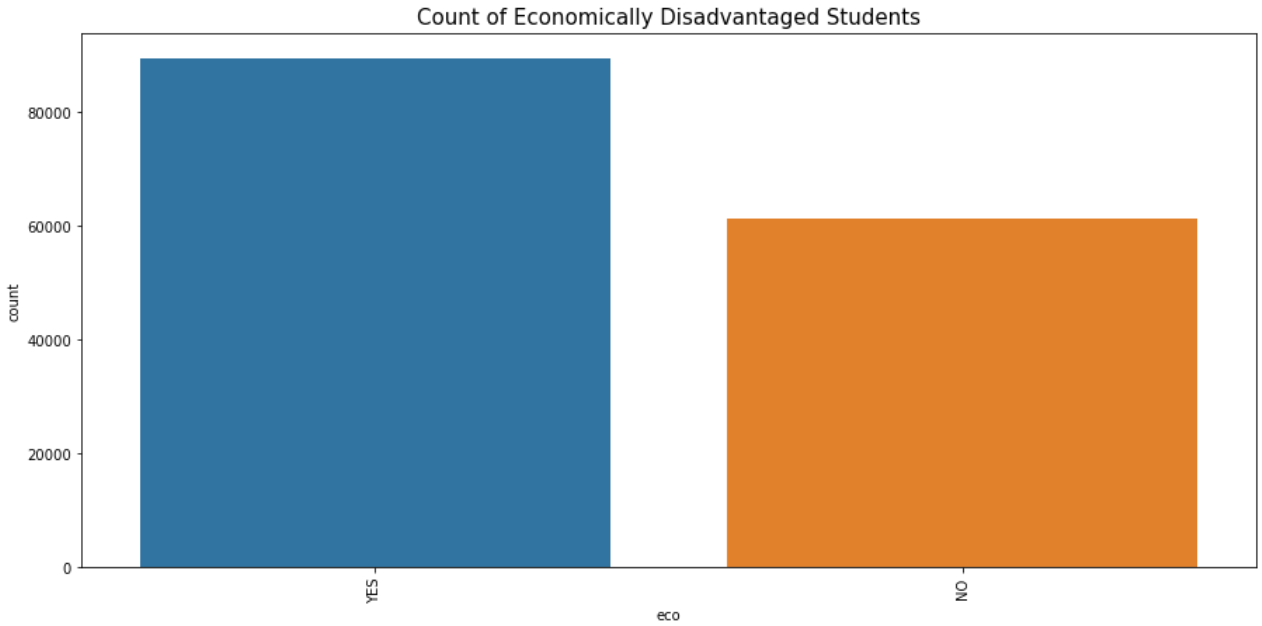
```
c=sns.countplot(data=df, x = 'ethnicity')
plt.xticks(rotation=90)
c.set_title("Ethnicity Count", fontsize=15)
plt.show()
```



We see that Hispanic is the predominant ethnicity.

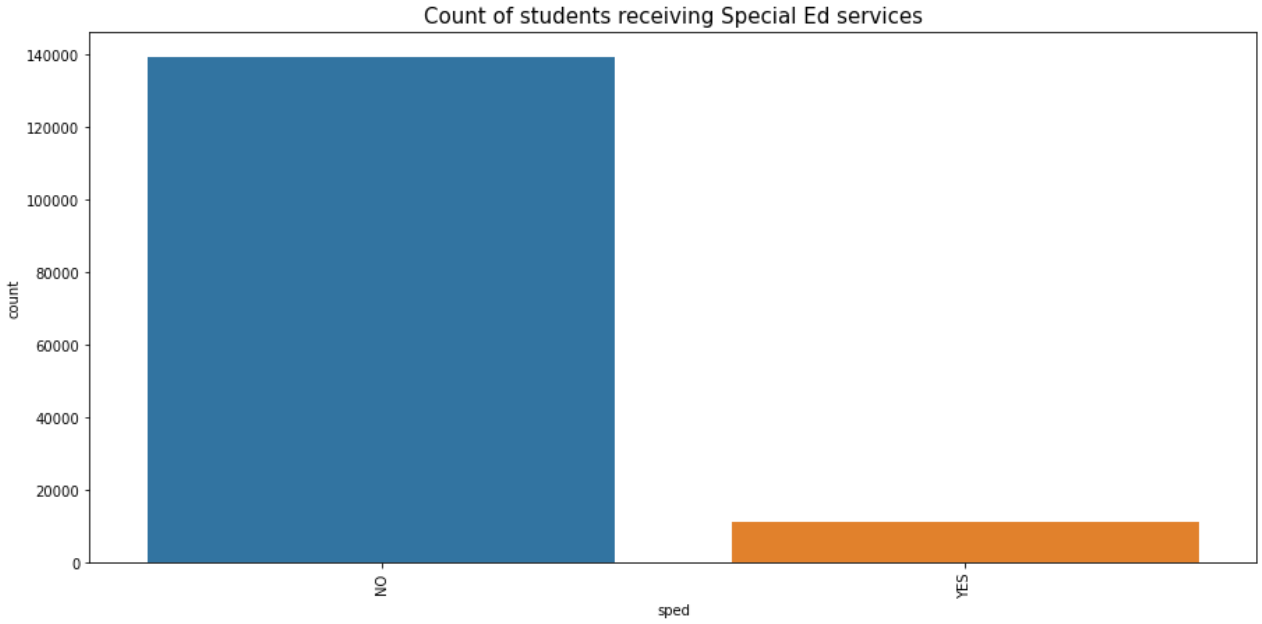
In [21]:

```
plt.figure(figsize = (15, 7))
d=sns.countplot(data=df, x = 'eco')
plt.xticks(rotation=90)
d.set_title("Count of Economically Disadvantaged Students", fontsize=15)
plt.show()
```



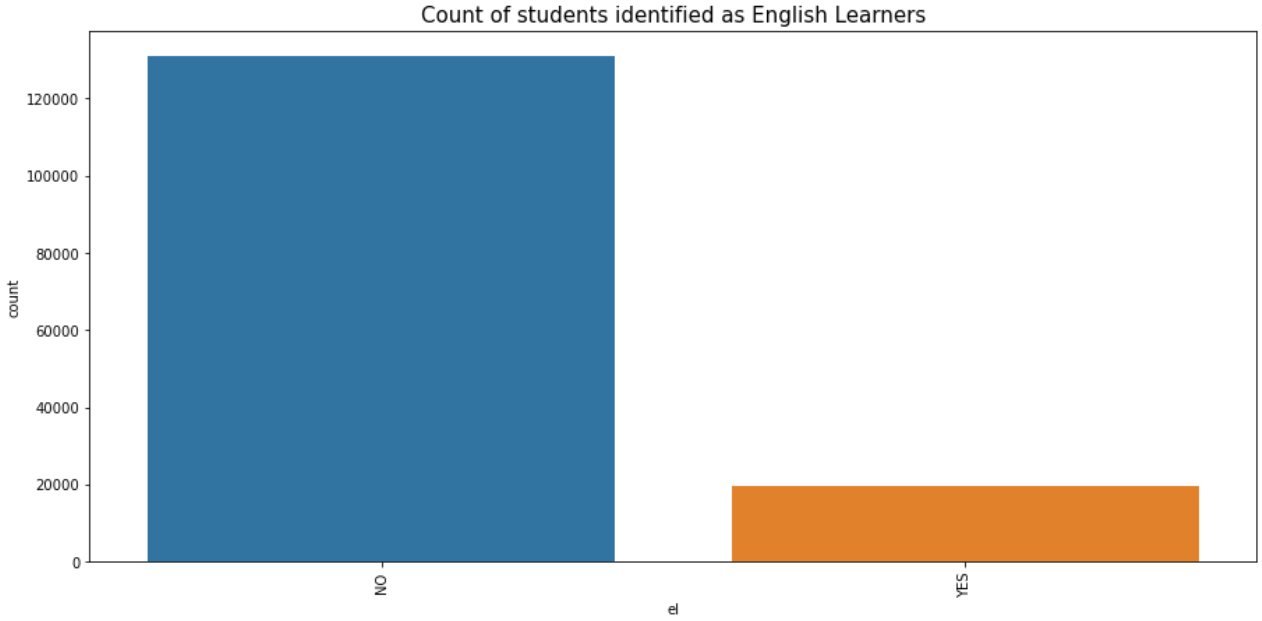
We have about 40% EcoDis rate for PreK.

In [22]:
```
plt.figure(figsize = (15, 7))
e=sns.countplot(data=df, x = 'sped')
plt.xticks(rotation=90)
e.set_title("Count of students receiving Special Ed services", fontsize=15)
plt.show()
```



We see 12% of students receive SPED services.

In [23]:
```
plt.figure(figsize = (15, 7))
f=sns.countplot(data=df, x = 'el')
plt.xticks(rotation=90)
f.set_title("Count of students identified as English Learners", fontsize=15)
plt.show()
```



We see about 13% of the students are ELs who took the exam in English.

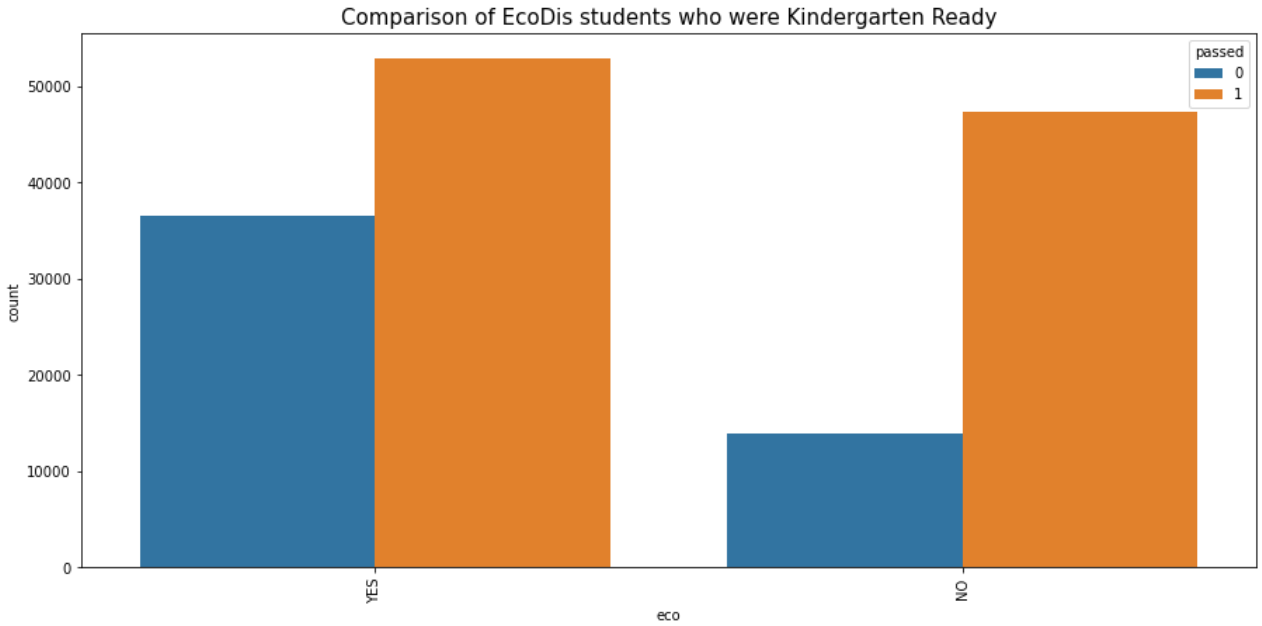
Multivariate Analysis

In [27]:
```
df_pass_eco=df.groupby(['eco', 'passed'])['student_id'].count().sort_values(ascending =
## creates a df where students are aggregated by '1' Kinder Ready and '0' Not Ready
df_pass_eco
```

Out[27]:

	eco	passed	student_id
0	YES	1	52871
1	NO	1	47375
2	YES	0	36554
3	NO	0	13912

In [32]:
```
plt.figure(figsize = (15, 7))
g=sns.countplot(data=df, x = 'eco', hue='passed')
plt.xticks(rotation=90)
g.set_title("Comparison of EcoDis students who were Kindergarten Ready", fontsize=15)
plt.show()
```



We can see that if a student is Economically Disadvantaged then you were much more likely to not be Kindergarten ready. Nearly 41% of EcoDis students were not ready as compared to 23% of students who were not EcoDis

In [36]:
```
df_pass_eth=df.groupby(['ethnicity', 'passed'])['student_id'].count().reset_index()
## creates a df where students are aggregated by '1' Kinder Ready and '0' Not Ready
df_pass_eth
```

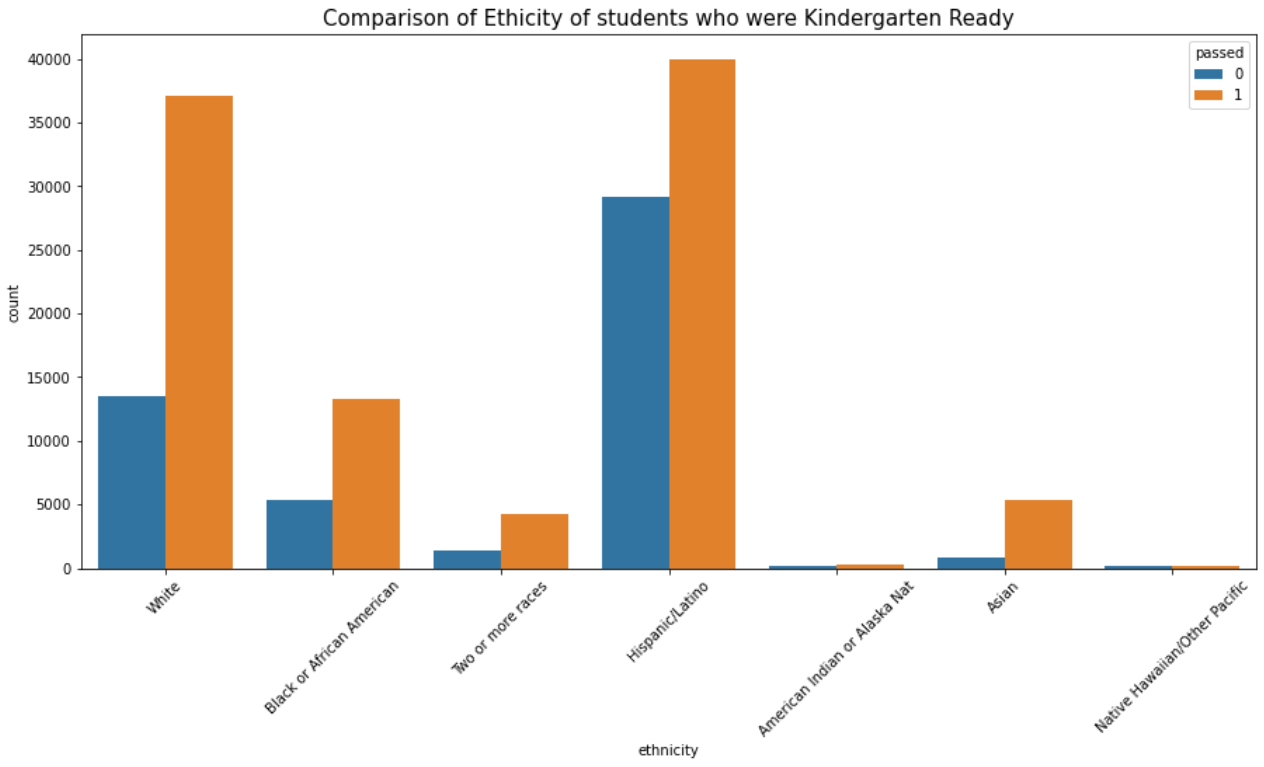
Out[36]:

	ethnicity	passed	student_id
0	American Indian or Alaska Nat	0	159

	ethnicity	passed	student_id
1	American Indian or Alaska Nat	1	274
2	Asian	0	844
3	Asian	1	5340
4	Black or African American	0	5364
5	Black or African American	1	13231
6	Hispanic/Latino	0	29108
7	Hispanic/Latino	1	39939
8	Native Hawaiian/Other Pacific	0	120
9	Native Hawaiian/Other Pacific	1	126
10	Two or more races	0	1387
11	Two or more races	1	4214
12	White	0	13484
13	White	1	37122

In [41]:

```
plt.figure(figsize = (15, 7))
h=sns.countplot(data=df, x = 'ethnicity', hue='passed')
plt.xticks(rotation=45)
h.set_title("Comparison of Ethicity of students who were Kindergarten Ready", fontsize=
plt.show()
```



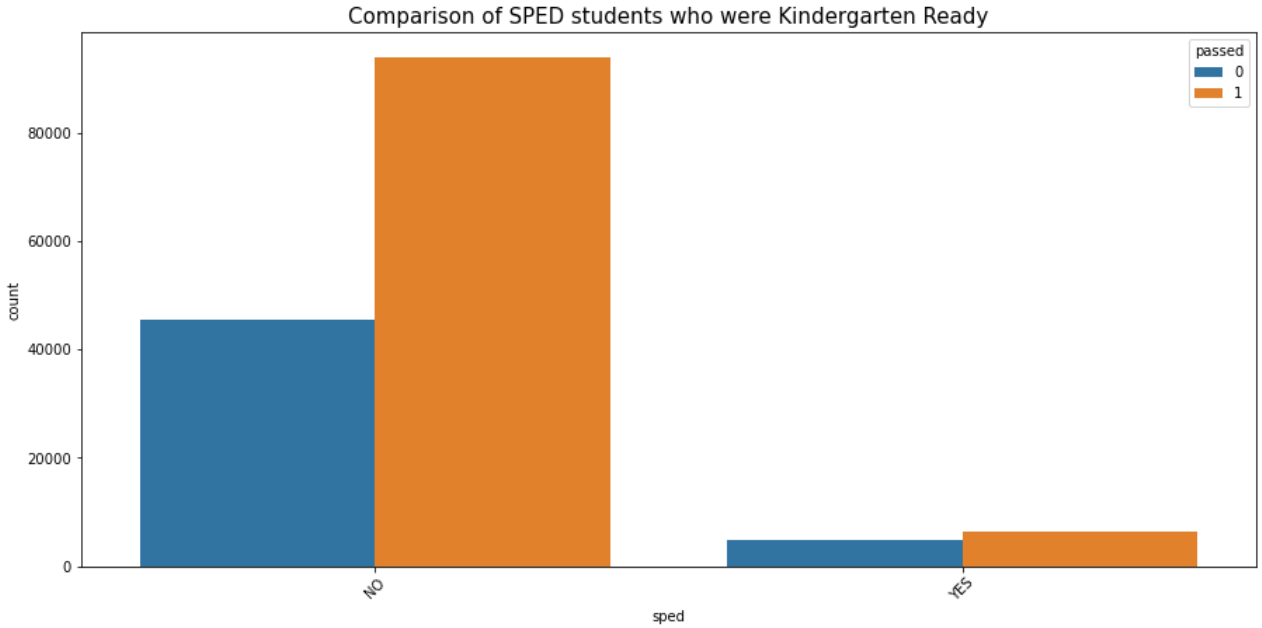
Native Hawaiian, Hispanic and Native Americans had the highest percentage of students not ready for Kinder.

In [39]:
```
df_pass_sped=df.groupby(['sped', 'passed'])['student_id'].count().reset_index()
## creates a df where students are aggregated by '1' Kinder Ready and '0' Not Ready
df_pass_sped
```

Out[39]:

	sped	passed	student_id
0	NO	0	45511
1	NO	1	93898
2	YES	0	4955
3	YES	1	6348

In [43]:
```
plt.figure(figsize = (15, 7))
i=sns.countplot(data=df, x = 'sped', hue='passed')
plt.xticks(rotation=45)
i.set_title("Comparison of SPED students who were Kindergarten Ready", fontsize=15)
plt.show()
```



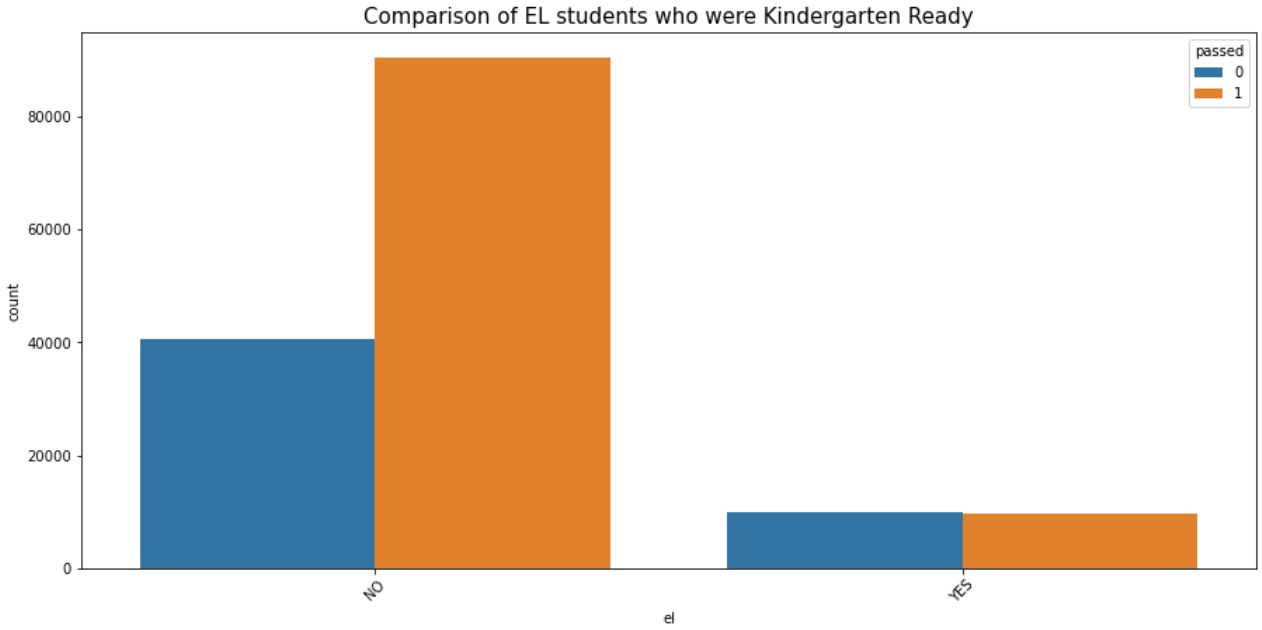
Students who needed SPED services were at 44% rate of not ready for Kinder as compared to the 33% who did not receive services.

In [42]:
```
df_pass_el=df.groupby(['el', 'passed'])['student_id'].count().reset_index()
## creates a df where students are aggregated by '1' Kinder Ready and '0' Not Ready
df_pass_el
```

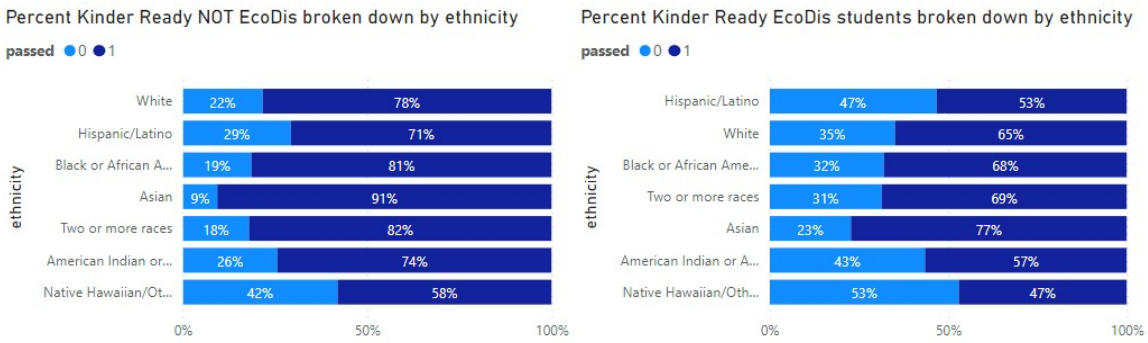
Out[42]:

	el	passed	student_id
0	NO	0	40577
1	NO	1	90481
2	YES	0	9889
3	YES	1	9765

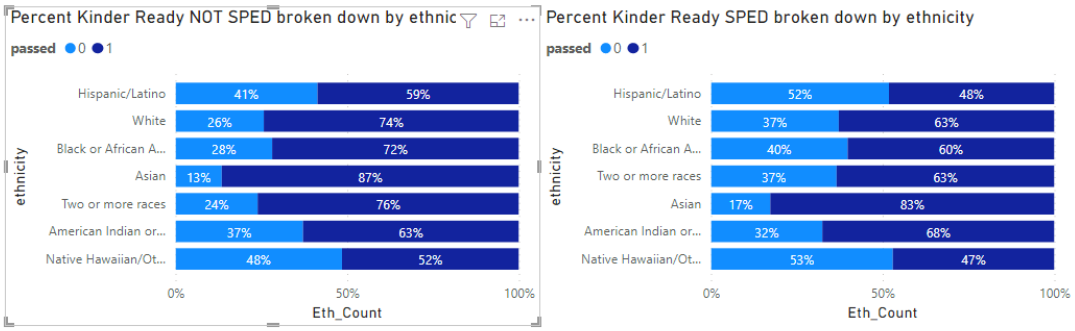
In [44]:
```
plt.figure(figsize = (15, 7))
j=sns.countplot(data=df, x = 'el', hue='passed')
plt.xticks(rotation=45)
j.set_title("Comparison of EL students who were Kindergarten Ready", fontsize=15)
plt.show()
```



1 in 2 EL students were not Kinder ready as compared to 31% of non EL students.



We can breakdown the information further and see how being Economically disadvantaged had a detrimental effect to Kinder Readiness across all ethnicities.



Again if you were offered SPED services you were more likely to not be ready for kinder for almost every race.

Overall Findings and Next Steps

- Economically Disadvantaged students were less likely to be Kinder ready as were SPED and EL students.
- We know that Hispanics make up the largest ethnictiy and have the highest counts of EcoDis, SPED and EL.
- I would like to create a map of districts with their corresponding counts for Kinder readiness broken down for a demographic groups. This can be used fund intervention for districts that have more students labeled as EcoDis, SPED, and EL.