School Presentation Appendix

March 10, 2024

By Pedro Huet López

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

psat_path = r"C:\Users\pedro\Downloads\Data␣
  ↪excercise\PSAT\assessment_psatsat_citywide_2023.xlsx"

columns_to_keep = ["School Year", "Category", "Category Description", "Test",␣
  ↪"# Students", "Average Math Score", "% Meeting College Readiness Benchmark"]
psatecondata = pd.read_excel(psat_path, header=0, sheet_name=2,␣
  ↪usecols=columns_to_keep)
mask = (psatecondata['Test'] == 'SAT') & (psatecondata['School Year'] ==␣
  ↪'2020-2021')
new_rows = psatecondata[mask].copy()
new_rows['Test'] = 'Combined'
psatecondata = pd.concat([psatecondata, new_rows], ignore_index=True)
psatecondata = psatecondata.loc[psatecondata['Test'].isin(['Combined', 'SAT'])]
psatecondata = psatecondata.loc[psatecondata['Category'].isin(['Economically␣
  ↪Disadvantaged', 'Race or Ethnicity', 'Race or Ethnicity/Gender', 'Students␣
  ↪with Disabilities'])]
```

[2]:
```
school_year_order = ["2017-2018", "2018-2019", "2019-2020", "2020-2021",␣
  ↪"2021-2022", "2022-2023"]

psatecondata['School Year'] = pd.Categorical(psatecondata['School Year'],␣
  ↪categories=school_year_order, ordered=True)
psatecondata['School Year Code'] = psatecondata['School Year'].cat.codes + 1
psatecondata = psatecondata.sort_values(by='School Year')
psatecondata.reset_index(drop=True, inplace=True)
print(psatecondata)
print(psatecondata.dtypes)
```

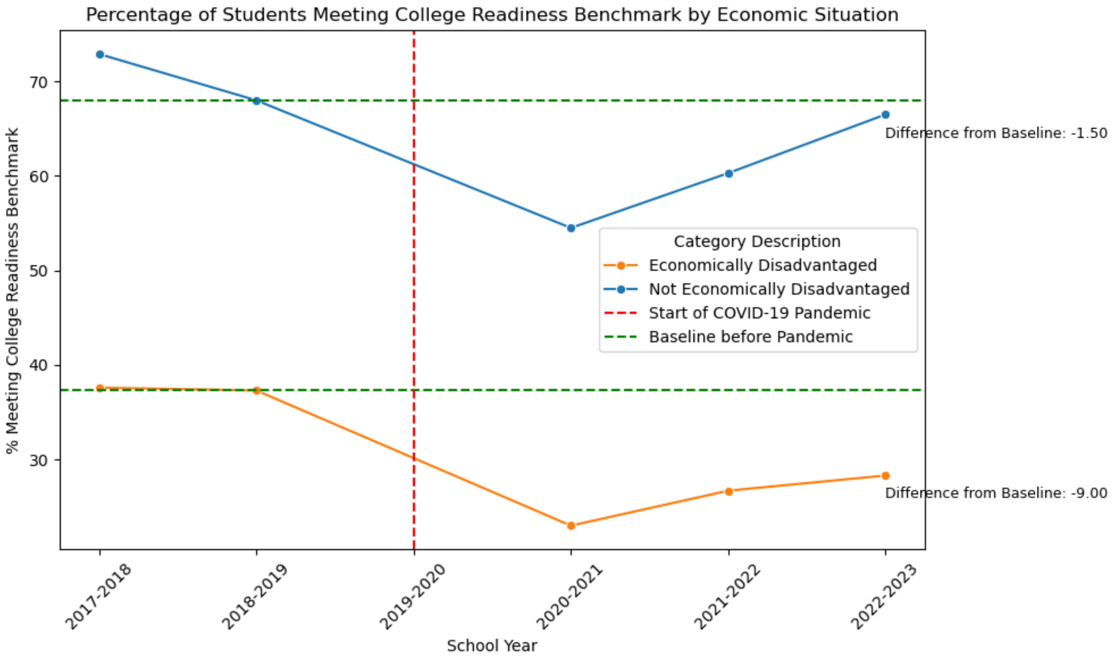
```
    School Year      Test                   Category  \
0     2017-2018  Combined          Race or Ethnicity
1     2017-2018  Combined   Race or Ethnicity/Gender
```

```
2     2017-2018  Combined    Race or Ethnicity/Gender
3     2017-2018  Combined    Race or Ethnicity/Gender
4     2017-2018       SAT    Race or Ethnicity/Gender
..          …       …                         …
317   2022-2023  Combined    Race or Ethnicity/Gender
318   2022-2023       SAT    Race or Ethnicity/Gender
319   2022-2023  Combined    Race or Ethnicity/Gender
320   2022-2023  Combined    Race or Ethnicity/Gender
321   2022-2023       SAT  Economically Disadvantaged

                 Category Description  # Students Average Math Score  \
0        HAWAIIAN OR PACIFIC ISLANDER          89
1                AMERICAN INDIAN/MALE         111
2    ASIAN OR PACIFIC ISLANDER/FEMALE           3
3      ASIAN OR PACIFIC ISLANDER/MALE           6
4                        ASIAN/FEMALE         478                585
..                                …         …                …
317               HISPANIC_NON-BINARY          28
318                      MULTI/FEMALE         139                541
319                      MULTI/FEMALE         397
320                   HISPANIC/FEMALE       14160
321                                NO        4585                536

    % Meeting College Readiness Benchmark  School Year Code
0                                      73                 1
1                                      55                 1
2                                                         1
3                                                         1
4                                    74.3                 1
..                                    …               …
317                                  53.6                 6
318                                  66.2                 6
319                                  69.8                 6
320                                  37.1                 6
321                                  61.7                 6

[322 rows x 8 columns]
School Year                              category
Test                                       object
Category                                   object
Category Description                       object
# Students                                  int64
Average Math Score                         object
% Meeting College Readiness Benchmark      object
School Year Code                             int8
dtype: object
```

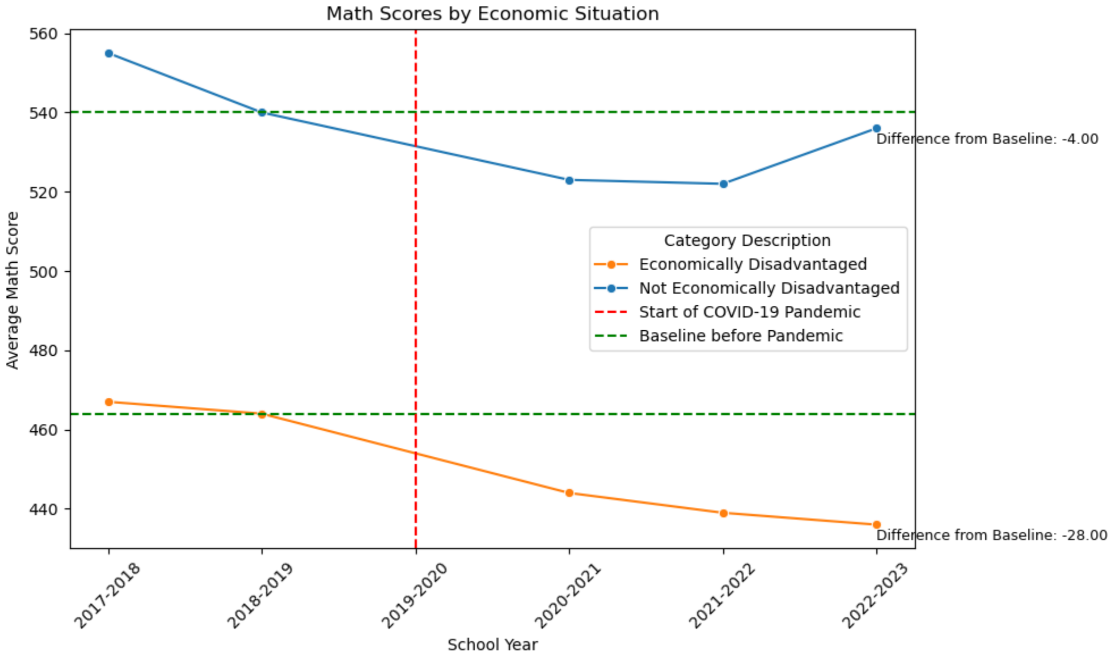
[3]:

```
economically_data = psatecondata[(psatecondata['Category'] == 'Economically␣
 ↪Disadvantaged') & (psatecondata['Test'] == 'Combined')]
economically_data.loc[economically_data['Category Description'] == 'YES',␣
 ↪'Category Description'] = 'Economically Disadvantaged'
economically_data.loc[economically_data['Category Description'] == 'NO',␣
 ↪'Category Description'] = 'Not Economically Disadvantaged'

category_descriptions = economically_data['Category Description'].unique()

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

green_line_label_added = False

for category_description in category_descriptions:
    category_data = economically_data[economically_data['Category Description']␣
 ↪== category_description]
    category_data = category_data.sort_values('School Year')
    sns.lineplot(data=category_data, x='School Year', y='% Meeting College␣
 ↪Readiness Benchmark', marker='o', label=category_description, errorbar=None)

    y_value_second_observation = category_data.iloc[1]['% Meeting College␣
 ↪Readiness Benchmark']

    if not green_line_label_added:
        plt.axhline(y=y_value_second_observation, color='green',␣
 ↪linestyle='--', label="Baseline before Pandemic")
        green_line_label_added = True
    else:
        plt.axhline(y=y_value_second_observation, color='green', linestyle='--')

    fifth_x_point = category_data.iloc[4]['School Year']
    fifth_x_difference = category_data.iloc[4]['% Meeting College Readiness␣
 ↪Benchmark'] - y_value_second_observation

    plt.text(fifth_x_point, category_data.iloc[4]['% Meeting College Readiness␣
 ↪Benchmark'] - 2, f'Difference from Baseline: {fifth_x_difference:.2f}',␣
 ↪ha='left', va='center', fontsize=9, color='black', rotation=0,␣
 ↪bbox=dict(facecolor='none', edgecolor='none'))

plt.axvline(x=2, color='red', linestyle='--', label="Start of COVID-19␣
 ↪Pandemic")

handles, labels = plt.gca().get_legend_handles_labels()

order = [2, 0, 3, 1]
```

```
plt.legend([handles[idx] for idx in order],[labels[idx] for idx in order],␣
 ↪title='Category Description')

plt.title('Percentage of Students Meeting College Readiness Benchmark by␣
 ↪Economic Situation')
plt.xlabel('School Year')
plt.ylabel('% Meeting College Readiness Benchmark')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

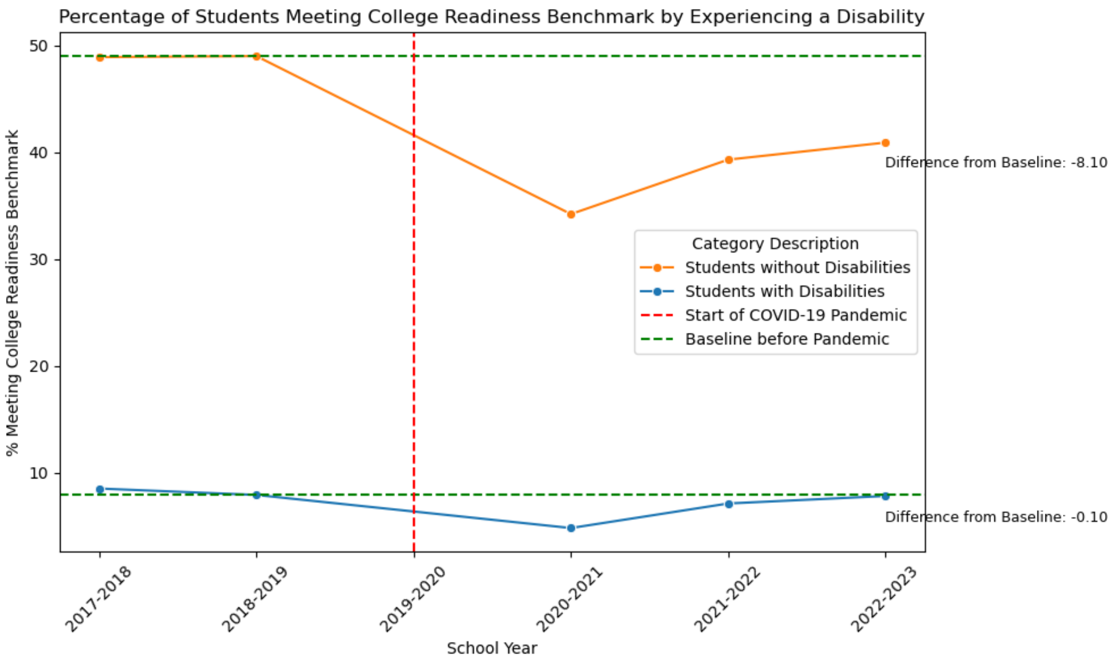


```
[4]: economically_data2 = psatecondata[(psatecondata['Category'] == 'Economically␣
 ↪Disadvantaged') & (psatecondata['Test'] == 'SAT')]
economically_data2.loc[economically_data2['Category Description'] == 'YES',␣
 ↪'Category Description'] = 'Economically Disadvantaged'
economically_data2.loc[economically_data2['Category Description'] == 'NO',␣
 ↪'Category Description'] = 'Not Economically Disadvantaged'

category_descriptions = economically_data2['Category Description'].unique()

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

green_line_label_added = False

for category_description in category_descriptions:
    category_data = economically_data2[economically_data2['Category␣
 ↪Description'] == category_description]
    category_data = category_data.sort_values('School Year')
    sns.lineplot(data=category_data, x='School Year', y='Average Math Score',␣
 ↪marker='o', label=category_description, errorbar=None)

    y_value_second_observation = category_data.iloc[1]['Average Math Score']

    if not green_line_label_added:
        plt.axhline(y=y_value_second_observation, color='green',␣
 ↪linestyle='--', label="Baseline before Pandemic")
        green_line_label_added = True
    else:
        plt.axhline(y=y_value_second_observation, color='green', linestyle='--')

    fifth_x_point = category_data.iloc[4]['School Year']
    fifth_x_difference = category_data.iloc[4]['Average Math Score'] -␣
 ↪y_value_second_observation

    plt.text(fifth_x_point, category_data.iloc[4]['Average Math Score'] - 3,␣
 ↪f'Difference from Baseline: {fifth_x_difference:.2f}', ha='left',␣
 ↪va='center', fontsize=9, color='black', rotation=0,␣
 ↪bbox=dict(facecolor='none', edgecolor='none'))

plt.axvline(x=2, color='red', linestyle='--', label="Start of COVID-19␣
 ↪Pandemic")

handles, labels = plt.gca().get_legend_handles_labels()

order = [2, 0, 3, 1]

plt.legend([handles[idx] for idx in order],[labels[idx] for idx in order],␣
 ↪title='Category Description')

plt.title('Math Scores by Economic Situation')
plt.xlabel('School Year')
plt.ylabel('Average Math Score')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



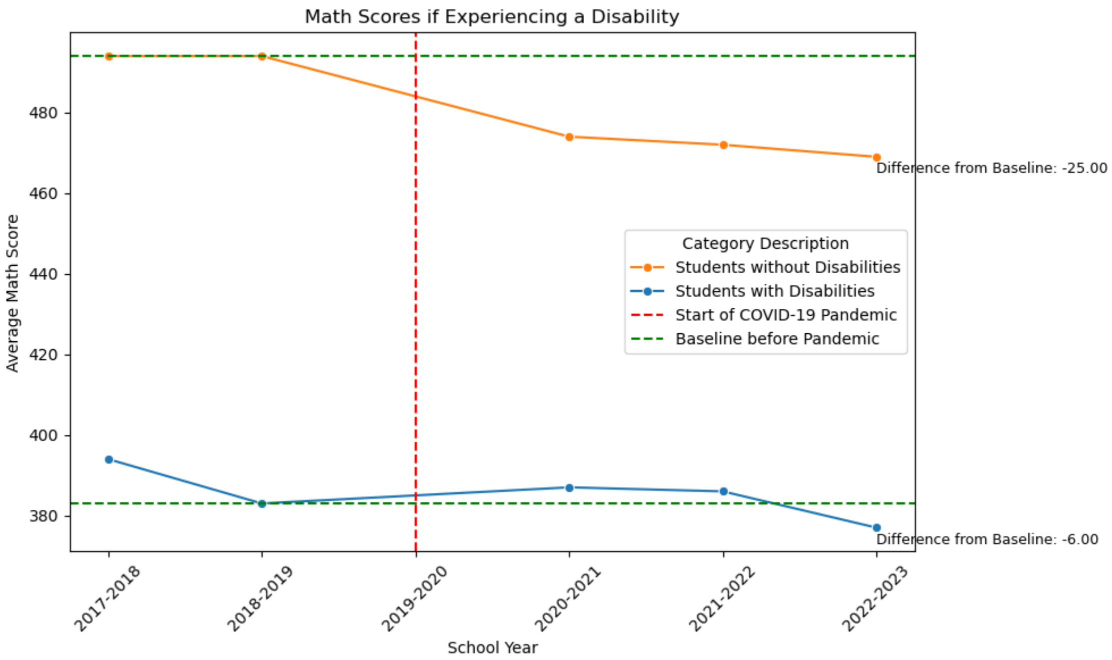
[5]:
```
economically_data = psatecondata[(psatecondata['Category'] == 'Students with Disabilities') & (psatecondata['Test'] == 'Combined')]
economically_data.loc[economically_data['Category Description'] == 'YES', 'Category Description'] = 'Students with Disabilities'
economically_data.loc[economically_data['Category Description'] == 'NO', 'Category Description'] = 'Students without Disabilities'

category_descriptions = economically_data['Category Description'].unique()

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

green_line_label_added = False

for category_description in category_descriptions:
    category_data = economically_data[economically_data['Category Description'] == category_description]
    category_data = category_data.sort_values('School Year')
    sns.lineplot(data=category_data, x='School Year', y='% Meeting College Readiness Benchmark', marker='o', label=category_description, errorbar=None)

    y_value_second_observation = category_data.iloc[1]['% Meeting College Readiness Benchmark']

    if not green_line_label_added:
```

```python
        plt.axhline(y=y_value_second_observation, color='green', linestyle='--', label="Baseline before Pandemic")
        green_line_label_added = True
    else:
        plt.axhline(y=y_value_second_observation, color='green', linestyle='--')

    fifth_x_point = category_data.iloc[4]['School Year']
    fifth_x_difference = category_data.iloc[4]['% Meeting College Readiness Benchmark'] - y_value_second_observation

    plt.text(fifth_x_point, category_data.iloc[4]['% Meeting College Readiness Benchmark'] - 2, f'Difference from Baseline: {fifth_x_difference:.2f}', ha='left', va='center', fontsize=9, color='black', rotation=0, bbox=dict(facecolor='none', edgecolor='none'))

plt.axvline(x=2, color='red', linestyle='--', label="Start of COVID-19 Pandemic")

handles, labels = plt.gca().get_legend_handles_labels()

order = [2, 0, 3, 1]

plt.legend([handles[idx] for idx in order],[labels[idx] for idx in order], title='Category Description')

plt.title('Percentage of Students Meeting College Readiness Benchmark by Experiencing a Disability')
plt.xlabel('School Year')
plt.ylabel('% Meeting College Readiness Benchmark')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



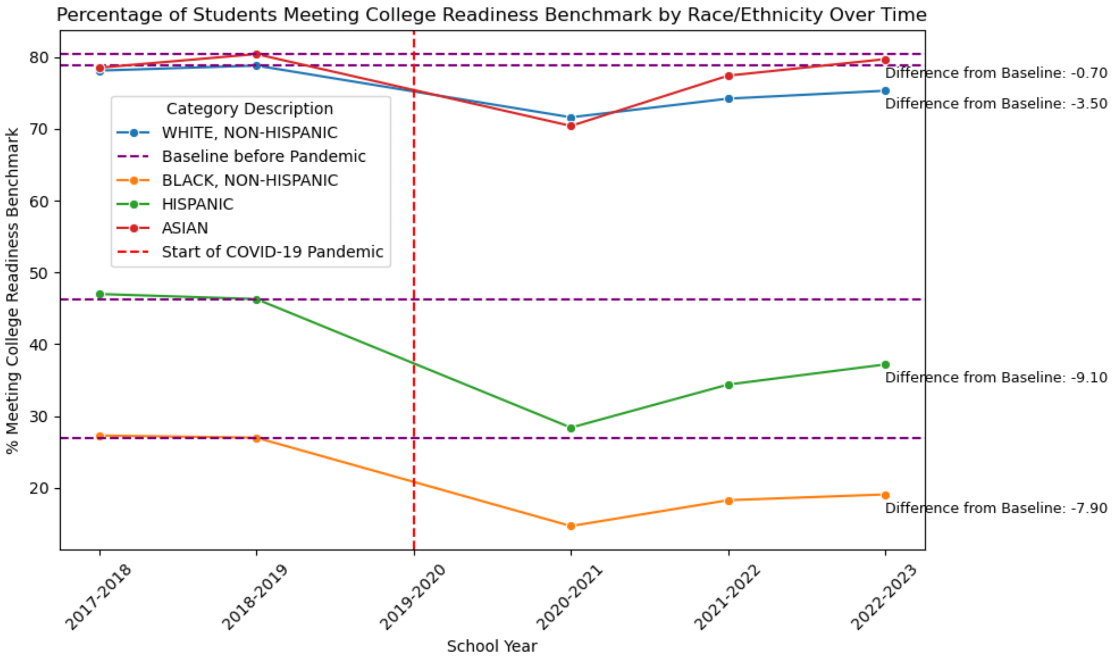
```
[6]: economically_data2 = psatecondata[(psatecondata['Category'] == 'Students with␣
      ↪Disabilities') & (psatecondata['Test'] == 'SAT')]
     economically_data2.loc[economically_data2['Category Description'] == 'YES',␣
      ↪'Category Description'] = 'Students with Disabilities'
     economically_data2.loc[economically_data2['Category Description'] == 'NO',␣
      ↪'Category Description'] = 'Students without Disabilities'

     category_descriptions = economically_data2['Category Description'].unique()

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

     green_line_label_added = False

     for category_description in category_descriptions:
         category_data = economically_data2[economically_data2['Category␣
      ↪Description'] == category_description]
         category_data = category_data.sort_values('School Year')
         sns.lineplot(data=category_data, x='School Year', y='Average Math Score',␣
      ↪marker='o', label=category_description, errorbar=None)

         y_value_second_observation = category_data.iloc[1]['Average Math Score']

         if not green_line_label_added:
             plt.axhline(y=y_value_second_observation, color='green',␣
      ↪linestyle='--', label="Baseline before Pandemic")
```

```
        green_line_label_added = True
    else:
        plt.axhline(y=y_value_second_observation, color='green', linestyle='--')

    fifth_x_point = category_data.iloc[4]['School Year']
    fifth_x_difference = category_data.iloc[4]['Average Math Score'] -␣
 ↪y_value_second_observation

    plt.text(fifth_x_point, category_data.iloc[4]['Average Math Score'] - 3,␣
 ↪f'Difference from Baseline: {fifth_x_difference:.2f}', ha='left',␣
 ↪va='center', fontsize=9, color='black', rotation=0,␣
 ↪bbox=dict(facecolor='none', edgecolor='none'))

plt.axvline(x=2, color='red', linestyle='--', label="Start of COVID-19␣
 ↪Pandemic")

handles, labels = plt.gca().get_legend_handles_labels()

order = [2, 0, 3, 1]

plt.legend([handles[idx] for idx in order],[labels[idx] for idx in order],␣
 ↪title='Category Description')

plt.title('Math Scores if Experiencing a Disability')
plt.xlabel('School Year')
plt.ylabel('Average Math Score')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
[7]: perform_by_race = psatecondata[(psatecondata['Category'] == 'Race or␣
      ↪Ethnicity') & (psatecondata['Test'] == 'Combined')]

     races = ['WHITE, NON-HISPANIC', 'BLACK, NON-HISPANIC', 'HISPANIC', 'ASIAN']

     perform_by_race.loc[perform_by_race['Category Description'].isin(races),␣
      ↪'Category Description'] = perform_by_race['Category Description']

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

     green_line_label_added = False

     for category_description in races:
         category_data = perform_by_race[perform_by_race['Category Description'] ==␣
      ↪category_description]
         category_data = category_data.sort_values('School Year')
         sns.lineplot(data=category_data, x='School Year', y='% Meeting College␣
      ↪Readiness Benchmark', marker='o', label=category_description, errorbar=None)

         y_value_second_observation = category_data.iloc[1]['% Meeting College␣
      ↪Readiness Benchmark']

         if not green_line_label_added:
             plt.axhline(y=y_value_second_observation, color='purple',␣
      ↪linestyle='--', label="Baseline before Pandemic")
```

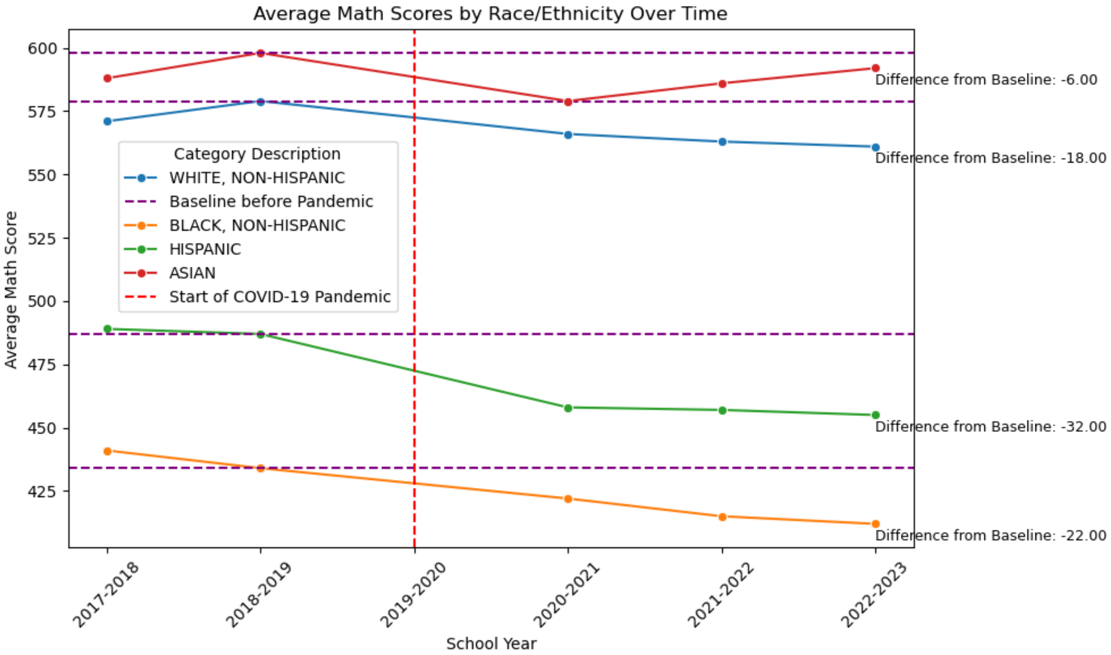
```
        green_line_label_added = True
    else:
        plt.axhline(y=y_value_second_observation, color='purple',␣
 ↪linestyle='--')

    fifth_x_point = category_data.iloc[4]['School Year']
    fifth_x_difference = category_data.iloc[4]['% Meeting College Readiness␣
 ↪Benchmark'] - y_value_second_observation

    plt.text(fifth_x_point, category_data.iloc[4]['% Meeting College Readiness␣
 ↪Benchmark']  - 2, f'Difference from Baseline: {fifth_x_difference:.2f}',␣
 ↪ha='left', va='center', fontsize=9, color='black', rotation=0,␣
 ↪bbox=dict(facecolor='none', edgecolor='none'))

plt.axvline(x=2, color='red', linestyle='--', label="Start of COVID-19␣
 ↪Pandemic")

plt.legend(title='Category Description', bbox_to_anchor=(0.05, 0.89),␣
 ↪loc='upper left')

plt.title('Percentage of Students Meeting College Readiness Benchmark by Race/
 ↪Ethnicity Over Time')
plt.xlabel('School Year')
plt.ylabel('% Meeting College Readiness Benchmark')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



[8]:

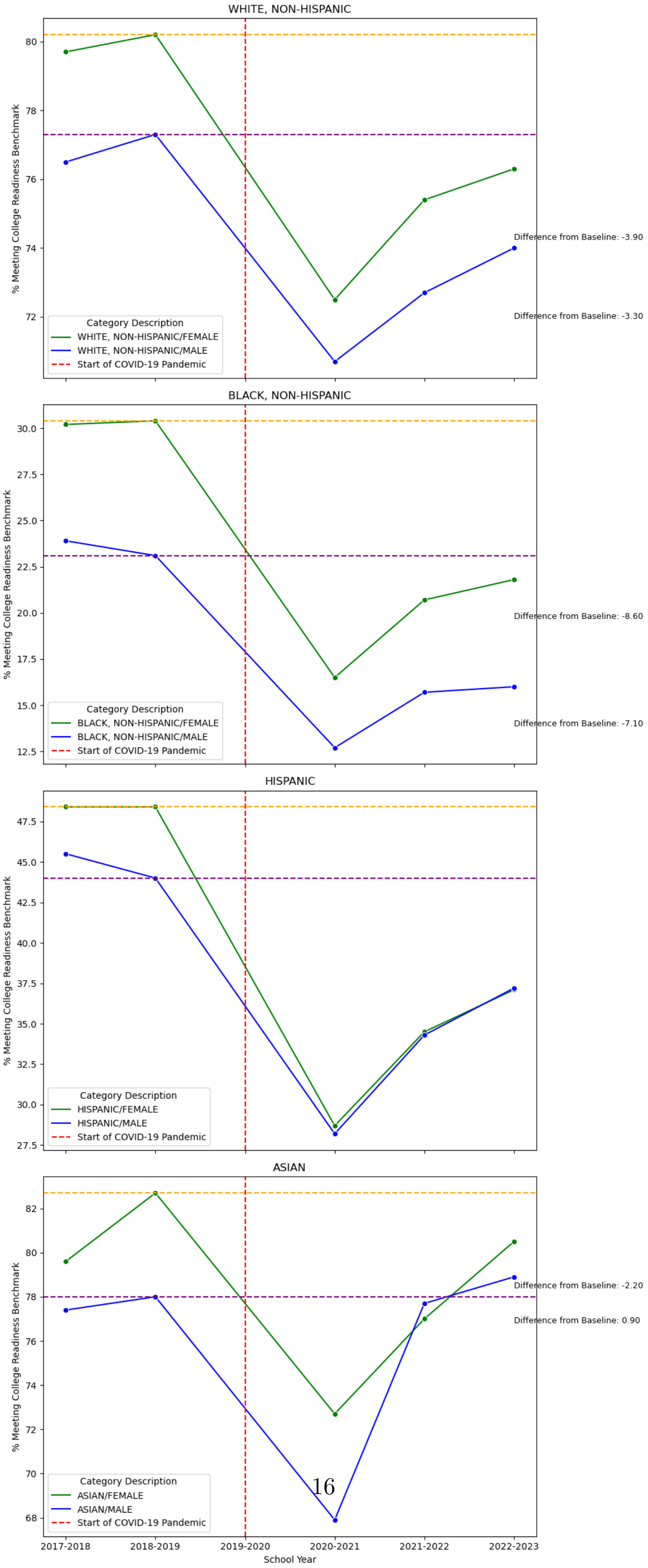
```
perform_by_race = psatecondata[(psatecondata['Category'] == 'Race or␣
 ↪Ethnicity') & (psatecondata['Test'] == 'SAT')]

races = ['WHITE, NON-HISPANIC', 'BLACK, NON-HISPANIC', 'HISPANIC', 'ASIAN']

perform_by_race.loc[perform_by_race['Category Description'].isin(races),␣
 ↪'Category Description'] = perform_by_race['Category Description']

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

green_line_label_added = False

for category_description in races:
    category_data = perform_by_race[perform_by_race['Category Description'] ==␣
 ↪category_description]
    category_data = category_data.sort_values('School Year')
    sns.lineplot(data=category_data, x='School Year', y='Average Math Score',␣
 ↪marker='o', label=category_description, errorbar=None)

    y_value_second_observation = category_data.iloc[1]['Average Math Score']

    if not green_line_label_added:
        plt.axhline(y=y_value_second_observation, color='purple',␣
 ↪linestyle='--', label="Baseline before Pandemic")
        green_line_label_added = True
    else:
        plt.axhline(y=y_value_second_observation, color='purple',␣
 ↪linestyle='--')

    fifth_x_point = category_data.iloc[4]['School Year']
    fifth_x_difference = category_data.iloc[4]['Average Math Score'] -␣
 ↪y_value_second_observation

    plt.text(fifth_x_point, category_data.iloc[4]['Average Math Score'] -5,␣
 ↪f'Difference from Baseline: {fifth_x_difference:.2f}', ha='left',␣
 ↪va='center', fontsize=9, color='black', rotation=0,␣
 ↪bbox=dict(facecolor='none', edgecolor='none'))

plt.axvline(x=2, color='red', linestyle='--', label="Start of COVID-19␣
 ↪Pandemic")

plt.legend(title='Category Description', bbox_to_anchor=(0.05, 0.80),␣
 ↪loc='upper left')
```

```
plt.title('Average Math Scores by Race/Ethnicity Over Time')
plt.xlabel('School Year')
plt.ylabel('Average Math Score')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
[9]: perform_by_race_gen = psatecondata[(psatecondata['Category'] == 'Race or␣
      ↪Ethnicity/Gender') & (psatecondata['Test'] == 'Combined')]

     fig, axes = plt.subplots(4, 1, figsize=(10, 24), sharex=True)

     gender_colors = {'MALE': 'green', 'FEMALE': 'blue'}

     race_groups = {'WHITE, NON-HISPANIC': ['WHITE, NON-HISPANIC/MALE', 'WHITE,␣
      ↪NON-HISPANIC/FEMALE'],
                    'BLACK, NON-HISPANIC': ['BLACK, NON-HISPANIC/MALE', 'BLACK,␣
      ↪NON-HISPANIC/FEMALE'],
                    'HISPANIC': ['HISPANIC/MALE', 'HISPANIC/FEMALE'],
                    'ASIAN': ['ASIAN/MALE', 'ASIAN/FEMALE']}

     baseline_colors = {'WHITE, NON-HISPANIC/MALE': 'purple', 'WHITE, NON-HISPANIC/
      ↪FEMALE': 'orange',
                        'BLACK, NON-HISPANIC/MALE': 'purple', 'BLACK, NON-HISPANIC/
      ↪FEMALE': 'orange',
                        'HISPANIC/MALE': 'purple', 'HISPANIC/FEMALE': 'orange',
                        'ASIAN/MALE': 'purple', 'ASIAN/FEMALE': 'orange'}

     for race, race_gender_combinations in race_groups.items():
         race_data = pd.concat([perform_by_race_gen[perform_by_race_gen['Category␣
      ↪Description'] == rgc] for rgc in race_gender_combinations])
         race_data = race_data.sort_values(['Category Description', 'School Year'])

         sns.lineplot(data=race_data, x='School Year', y='% Meeting College␣
      ↪Readiness Benchmark', hue='Category Description', marker='o',␣
      ↪ax=axes[list(race_groups.keys()).index(race)], palette=[gender_colors[rgc.
      ↪split('/')[1]] for rgc in race_gender_combinations])

         baseline_values = race_data.groupby('Category Description').nth(1)
         for gender, baseline_value in baseline_values.iterrows():
             color = baseline_colors[gender]
             axes[list(race_groups.keys()).index(race)].axhline(y=baseline_value['%␣
      ↪Meeting College Readiness Benchmark'], color=color, linestyle='--',␣
      ↪label=None if gender == 'WHITE, NON-HISPANIC/MALE' else None)

         axes[list(race_groups.keys()).index(race)].axvline(x=2, color='red',␣
      ↪linestyle='--', label="Start of COVID-19 Pandemic")

         if race != 'HISPANIC':
             fifth_x_points = race_data.groupby('Category Description').nth(4)
             for gender, fifth_x_point in fifth_x_points.iterrows():
                 baseline_value = baseline_values.loc[gender, '% Meeting College␣
      ↪Readiness Benchmark']
                 fifth_x_difference = fifth_x_point['% Meeting College Readiness␣
      ↪Benchmark'] - baseline_value
                 axes[list(race_groups.keys()).index(race)].
      ↪text(fifth_x_point['School Year'], fifth_x_point['% Meeting College␣
      ↪Readiness Benchmark'] - 2,
                              f'Difference from Baseline: {fifth_x_difference:.2f}',␣
      ↪ha='left', va='center', fontsize=9, color='black',
                              rotation=0, bbox=dict(facecolor='none',␣
      ↪edgecolor='none'))

         axes[list(race_groups.keys()).index(race)].set_title(f'{race}')
         axes[list(race_groups.keys()).index(race)].set_ylabel('% Meeting College␣
      ↪Readiness Benchmark')
         axes[list(race_groups.keys()).index(race)].legend(title='Category␣
      ↪Description')

     axes[-1].set_xlabel('School Year')
```

```
plt.tight_layout()
plt.show()
```


WHITE, NON-HISPANIC
% Meeting College Readiness Benchmark
80
78
76
74
72
Difference from Baseline: -3.90
Difference from Baseline: -3.30
Category Description
WHITE, NON-HISPANIC/FEMALE
WHITE, NON-HISPANIC/MALE
Start of COVID-19 Pandemic
BLACK, NON-HISPANIC
% Meeting College Readiness Benchmark
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5
Difference from Baseline: -8.60
Difference from Baseline: -7.10
Category Description
BLACK, NON-HISPANIC/FEMALE
BLACK, NON-HISPANIC/MALE
Start of COVID-19 Pandemic
HISPANIC
% Meeting College Readiness Benchmark
47.5
45.0
42.5
40.0
37.5
35.0
32.5
30.0
27.5
Category Description
HISPANIC/FEMALE
HISPANIC/MALE
Start of COVID-19 Pandemic
ASIAN
% Meeting College Readiness Benchmark
82
80
78
76
74
72
70
68
Difference from Baseline: -2.20
Difference from Baseline: 0.90
Category Description
ASIAN/FEMALE
ASIAN/MALE
Start of COVID-19 Pandemic
2017-2018
2018-2019
2019-2020
2020-2021
2021-2022
2022-2023
School Year

[10]:

```
perform_by_race_gen = psatecondata[(psatecondata['Category'] == 'Race or␣
 ↪Ethnicity/Gender') & (psatecondata['Test'] == 'SAT')]

gender_colors = {'MALE': 'green', 'FEMALE': 'blue'}

race_groups = {'WHITE, NON-HISPANIC': ['WHITE, NON-HISPANIC/MALE', 'WHITE,␣
 ↪NON-HISPANIC/FEMALE'],
               'BLACK, NON-HISPANIC': ['BLACK, NON-HISPANIC/MALE', 'BLACK,␣
 ↪NON-HISPANIC/FEMALE'],
               'HISPANIC': ['HISPANIC/MALE', 'HISPANIC/FEMALE'],
               'ASIAN': ['ASIAN/MALE', 'ASIAN/FEMALE']}

for race, race_gender_combinations in race_groups.items():
    race_data = pd.concat([perform_by_race_gen[perform_by_race_gen['Category␣
 ↪Description'] == rgc] for rgc in race_gender_combinations])
    race_data = race_data.sort_values(['Category Description', 'School Year'])

    fig, ax = plt.subplots(figsize=(10, 6))

    sns.lineplot(data=race_data, x='School Year', y='Average Math Score',␣
 ↪hue='Category Description', marker='o', palette=[gender_colors[rgc.split('/
 ↪')[1]] for rgc in race_gender_combinations], ax=ax)

    baseline_values = race_data.groupby('Category Description').nth(1)
    for gender, baseline_value in baseline_values.iterrows():
        ax.axhline(y=baseline_value['Average Math Score'], color='purple',␣
 ↪linestyle='--', label="Baseline before Pandemic" if gender == 'WHITE,␣
 ↪NON-HISPANIC/MALE' else None)

    ax.axvline(x=2, color='red', linestyle='--', label="Start of COVID-19␣
 ↪Pandemic")

    fifth_x_points = race_data.groupby('Category Description').nth(4)
    for gender, fifth_x_point in fifth_x_points.iterrows():
        baseline_value = baseline_values.loc[gender, 'Average Math Score']
        fifth_x_difference = fifth_x_point['Average Math Score'] -␣
 ↪baseline_value
        ax.text(fifth_x_point['School Year'], fifth_x_point['Average Math␣
 ↪Score'] - 2,
                     f'Difference from Baseline: {fifth_x_difference:.2f}',␣
 ↪ha='left', va='center', fontsize=9, color='black',
                     rotation=0, bbox=dict(facecolor='none', edgecolor='none'))

    ax.set_title(f'{race}')
    ax.set_ylabel('Average Math Score')
    ax.legend(title='Category Description')
    ax.set_xlabel('School Year')
    plt.grid(False)

    # Save the plot as an image
    plt.savefig(f'{race}_plot.png')

    # Close the plot to release memory
    plt.close()

# After generating all the plots, insert them into a presentation manuall
```

[11]:
```
lefr_path = r"C:\Users\pedro\Downloads\Data excercise\Limited English Frequency␣
 ↪Report\demographics_lepiepfrm_20thday_sy2024_finalv2.xlsx"

student_typ_data23 = pd.read_excel(lefr_path, header=2, sheet_name=0)
student_typ_data23 = student_typ_data23.iloc[:-2]
student_typ_data23 = student_typ_data23[student_typ_data23['Grade'].
 ↪isin(['District Total', 'Ninth Grade', 'Tenth Grade', 'Eleventh Grade'])]
student_typ_data23 = student_typ_data23.drop(columns=[student_typ_data23.
 ↪columns[7], student_typ_data23.columns[5], student_typ_data23.columns[3]])
student_typ_data23['Year'] = 2023

melted_data23 = pd.melt(student_typ_data23, id_vars=['Grade', 'Year'],␣
 ↪value_vars=['N', 'N.1', 'N.2'], var_name='Student_Type', value_name='Count')
melted_data23['Student_Type'] = melted_data23['Student_Type'].replace({'N':␣
 ↪'Bilingual', 'N.1': 'Diverse Learners', 'N.2': 'Economically Disadvantaged'})

print(melted_data23)
```

```
             Grade  Year                Student_Type     Count
0   District Total  2023                   Bilingual   79833.0
1      Ninth Grade  2023                   Bilingual    5964.0
2      Tenth Grade  2023                   Bilingual    5813.0
3   Eleventh Grade  2023                   Bilingual    4764.0
4   District Total  2023            Diverse Learners   51910.0
5      Ninth Grade  2023            Diverse Learners    3924.0
6      Tenth Grade  2023            Diverse Learners    4262.0
7   Eleventh Grade  2023            Diverse Learners    4022.0
8   District Total  2023  Economically Disadvantaged  228466.0
9      Ninth Grade  2023  Economically Disadvantaged   19069.0
10     Tenth Grade  2023  Economically Disadvantaged   21667.0
11  Eleventh Grade  2023  Economically Disadvantaged   20440.0
```

[12]:
```
lefr_path2 = r"C:\Users\pedro\Downloads\Data excercise\Limited English␣
 ↪Frequency Report\demographics_lepsped_20thday_2023"
```

```
student_typ_data22 = pd.read_excel(lefr_path, header=2, sheet_name=0)
student_typ_data22 = student_typ_data22.iloc[:-2]
student_typ_data22 = student_typ_data22[student_typ_data22['Grade'].
 ↪isin(['District Total', 'Ninth Grade', 'Tenth Grade', 'Eleventh Grade'])]
student_typ_data22 = student_typ_data22.drop(columns=[student_typ_data22.
 ↪columns[7], student_typ_data22.columns[5], student_typ_data22.columns[3]])
student_typ_data22['Year'] = 2022

melted_data22 = pd.melt(student_typ_data22, id_vars=['Grade', 'Year'],␣
 ↪value_vars=['N', 'N.1', 'N.2'], var_name='Student_Type', value_name='Count')
melted_data22['Student_Type'] = melted_data22['Student_Type'].replace({'N':␣
 ↪'Bilingual', 'N.1': 'Diverse Learners', 'N.2': 'Economically Disadvantaged'})

print(melted_data22)
```

```
             Grade  Year                Student_Type     Count
0   District Total  2022                   Bilingual   79833.0
1      Ninth Grade  2022                   Bilingual    5964.0
2      Tenth Grade  2022                   Bilingual    5813.0
3   Eleventh Grade  2022                   Bilingual    4764.0
4   District Total  2022            Diverse Learners   51910.0
5      Ninth Grade  2022            Diverse Learners    3924.0
6      Tenth Grade  2022            Diverse Learners    4262.0
7   Eleventh Grade  2022            Diverse Learners    4022.0
8   District Total  2022  Economically Disadvantaged  228466.0
9      Ninth Grade  2022  Economically Disadvantaged   19069.0
10     Tenth Grade  2022  Economically Disadvantaged   21667.0
11  Eleventh Grade  2022  Economically Disadvantaged   20440.0
```

[13]:

```
lefr_path3 = r"C:\Users\pedro\Downloads\Data excercise\Limited English␣
 ↪Frequency Report\demographics_lepsped_2022_v10272021.xls"

student_typ_data21 = pd.read_excel(lefr_path3, header=1, sheet_name=0)
student_typ_data21 = student_typ_data21.iloc[:-2]
student_typ_data21 = student_typ_data21[student_typ_data21['Grade'].
 ↪isin(['District Total', 'Ninth Grade', 'Tenth Grade', 'Eleventh Grade'])]
student_typ_data21 = student_typ_data21.drop(columns=[student_typ_data21.
 ↪columns[7], student_typ_data21.columns[5], student_typ_data21.columns[3]])
student_typ_data21['Year'] = 2021

melted_data21 = pd.melt(student_typ_data21, id_vars=['Grade', 'Year'],␣
 ↪value_vars=['N', 'N.1', 'N.2'], var_name='Student_Type', value_name='Count')
melted_data21['Student_Type'] = melted_data21['Student_Type'].replace({'N':␣
 ↪'Bilingual', 'N.1': 'Diverse Learners', 'N.2': 'Economically Disadvantaged'})

print(melted_data21)
```

```
              Grade  Year                 Student_Type     Count
0    District Total  2021                    Bilingual   69268.0
1       Ninth Grade  2021                    Bilingual    4544.0
2       Tenth Grade  2021                    Bilingual    3502.0
3    Eleventh Grade  2021                    Bilingual    3126.0
4    District Total  2021             Diverse Learners   48749.0
5       Ninth Grade  2021             Diverse Learners    4342.0
6       Tenth Grade  2021             Diverse Learners    4369.0
7    Eleventh Grade  2021             Diverse Learners    4306.0
8    District Total  2021  Economically Disadvantaged  230496.0
9       Ninth Grade  2021  Economically Disadvantaged   19725.0
10      Tenth Grade  2021  Economically Disadvantaged   20289.0
11   Eleventh Grade  2021  Economically Disadvantaged   20107.0
```

[14]:

```python
lefr_path4 = r"C:\Users\pedro\Downloads\Data excercise\Limited English Frequency Report\demographics_lepsped_2021_v10072020.xls"

student_typ_data20 = pd.read_excel(lefr_path4, header=1, sheet_name=0)
student_typ_data20 = student_typ_data20.iloc[:-2]
student_typ_data20 = student_typ_data20[student_typ_data20['Grade'].isin(['District Total', 'Ninth Grade', 'Tenth Grade', 'Eleventh Grade'])]
student_typ_data20 = student_typ_data20.drop(columns=[student_typ_data20.columns[7], student_typ_data20.columns[5], student_typ_data20.columns[3]])
student_typ_data20['Year'] = 2020

melted_data20 = pd.melt(student_typ_data20, id_vars=['Grade', 'Year'], value_vars=['N', 'N.1', 'N.2'], var_name='Student_Type', value_name='Count')
melted_data20['Student_Type'] = melted_data20['Student_Type'].replace({'N': 'Bilingual', 'N.1': 'Diverse Learners', 'N.2': 'Economically Disadvantaged'})

print(melted_data20)
```

```
              Grade  Year                 Student_Type     Count
0    District Total  2020                    Bilingual   63313.0
1       Ninth Grade  2020                    Bilingual    3369.0
2       Tenth Grade  2020                    Bilingual    3285.0
3    Eleventh Grade  2020                    Bilingual    2873.0
4    District Total  2020             Diverse Learners   49655.0
5       Ninth Grade  2020             Diverse Learners    4218.0
6       Tenth Grade  2020             Diverse Learners    4498.0
7    Eleventh Grade  2020             Diverse Learners    4081.0
8    District Total  2020  Economically Disadvantaged  217392.0
9       Ninth Grade  2020  Economically Disadvantaged   17078.0
10      Tenth Grade  2020  Economically Disadvantaged   17174.0
11   Eleventh Grade  2020  Economically Disadvantaged   16172.0
```

```
[15]: lefr_path5 = r"C:\Users\pedro\Downloads\Data excercise\Limited English␣
       ↪Frequency Report\demographics_lepsped_2020_10202020.xls"

      student_typ_data19 = pd.read_excel(lefr_path5, header=1, sheet_name=0)
      student_typ_data19 = student_typ_data19.iloc[:-2]
      student_typ_data19 = student_typ_data19[student_typ_data19['Grade'].
       ↪isin(['District Total', 'Ninth Grade', 'Tenth Grade', 'Eleventh Grade'])]
      student_typ_data19 = student_typ_data19.drop(columns=[student_typ_data19.
       ↪columns[7], student_typ_data19.columns[5], student_typ_data19.columns[3]])
      student_typ_data19['Year'] = 2019

      melted_data19 = pd.melt(student_typ_data19, id_vars=['Grade', 'Year'],␣
       ↪value_vars=['N', 'N.1', 'N.2'], var_name='Student_Type', value_name='Count')
      melted_data19['Student_Type'] = melted_data19['Student_Type'].replace({'N':␣
       ↪'Bilingual', 'N.1': 'Diverse Learners', 'N.2': 'Economically Disadvantaged'})

      print(melted_data19)
```

```
             Grade  Year                Student_Type     Count
0   District Total  2019                   Bilingual   69012.0
1      Ninth Grade  2019                   Bilingual    3249.0
2      Tenth Grade  2019                   Bilingual    3061.0
3   Eleventh Grade  2019                   Bilingual    2668.0
4   District Total  2019            Diverse Learners   51691.0
5      Ninth Grade  2019            Diverse Learners    4362.0
6      Tenth Grade  2019            Diverse Learners    4419.0
7   Eleventh Grade  2019            Diverse Learners    3925.0
8   District Total  2019  Economically Disadvantaged  271179.0
9      Ninth Grade  2019  Economically Disadvantaged   20569.0
10     Tenth Grade  2019  Economically Disadvantaged   21735.0
11  Eleventh Grade  2019  Economically Disadvantaged   20401.0
```

```
[16]: lefr_path6 = r"C:\Users\pedro\Downloads\Data excercise\Limited English␣
       ↪Frequency Report\demographics_lepsped_2019_10202020.xls"

      student_typ_data18 = pd.read_excel(lefr_path6, header=1, sheet_name=0)
      student_typ_data18 = student_typ_data18.iloc[:-2]
      student_typ_data18 = student_typ_data18[student_typ_data18['Grade'].
       ↪isin(['District Total', 'Ninth Grade', 'Tenth Grade', 'Eleventh Grade'])]
      student_typ_data18 = student_typ_data18.drop(columns=[student_typ_data18.
       ↪columns[7], student_typ_data18.columns[5], student_typ_data18.columns[3]])
      student_typ_data18['Year'] = 2018

      melted_data18 = pd.melt(student_typ_data18, id_vars=['Grade', 'Year'],␣
       ↪value_vars=['N', 'N.1', 'N.2'], var_name='Student_Type', value_name='Count')
      melted_data18['Student_Type'] = melted_data18['Student_Type'].replace({'N':␣
       ↪'Bilingual', 'N.1': 'Diverse Learners', 'N.2': 'Economically Disadvantaged'})
```

```
print(melted_data18)
```

```
            Grade  Year                Student_Type     Count
0   District Total  2018                   Bilingual   69282.0
1      Ninth Grade  2018                   Bilingual    2992.0
2      Tenth Grade  2018                   Bilingual    2977.0
3   Eleventh Grade  2018                   Bilingual    2513.0
4   District Total  2018            Diverse Learners   50772.0
5      Ninth Grade  2018            Diverse Learners    4387.0
6      Tenth Grade  2018            Diverse Learners    4613.0
7   Eleventh Grade  2018            Diverse Learners    3811.0
8   District Total  2018  Economically Disadvantaged  276836.0
9      Ninth Grade  2018  Economically Disadvantaged   21937.0
10     Tenth Grade  2018  Economically Disadvantaged   23409.0
11  Eleventh Grade  2018  Economically Disadvantaged   21022.0
```

```
[17]: lefr_path7 = r"C:\Users\pedro\Downloads\Data excercise\Limited English␣
       ↪Frequency Report\demographics_lepsped_2018_10202020.xls"

      student_typ_data17 = pd.read_excel(lefr_path7, header=1, sheet_name=0)
      student_typ_data17 = student_typ_data17.iloc[:-2]
      student_typ_data17 = student_typ_data17[student_typ_data17['Grade'].
       ↪isin(['District Total', 'Ninth Grade', 'Tenth Grade', 'Eleventh Grade'])]
      student_typ_data17 = student_typ_data17.drop(columns=[student_typ_data17.
       ↪columns[7], student_typ_data17.columns[5], student_typ_data17.columns[3]])
      student_typ_data17['Year'] = 2017

      melted_data17 = pd.melt(student_typ_data17, id_vars=['Grade', 'Year'],␣
       ↪value_vars=['N', 'N.1', 'N.2'], var_name='Student_Type', value_name='Count')
      melted_data17['Student_Type'] = melted_data17['Student_Type'].replace({'N':␣
       ↪'Bilingual', 'N.1': 'Diverse Learners', 'N.2': 'Economically Disadvantaged'})

      print(melted_data17)
```

```
            Grade  Year                Student_Type     Count
0   District Total  2017                   Bilingual   67834.0
1      Ninth Grade  2017                   Bilingual    2695.0
2      Tenth Grade  2017                   Bilingual    2798.0
3   Eleventh Grade  2017                   Bilingual    2300.0
4   District Total  2017            Diverse Learners   50917.0
5      Ninth Grade  2017            Diverse Learners    4454.0
6      Tenth Grade  2017            Diverse Learners    4465.0
7   Eleventh Grade  2017            Diverse Learners    3888.0
8   District Total  2017  Economically Disadvantaged  288572.0
9      Ninth Grade  2017  Economically Disadvantaged   22413.0
10     Tenth Grade  2017  Economically Disadvantaged   23655.0
11  Eleventh Grade  2017  Economically Disadvantaged   21953.0
```

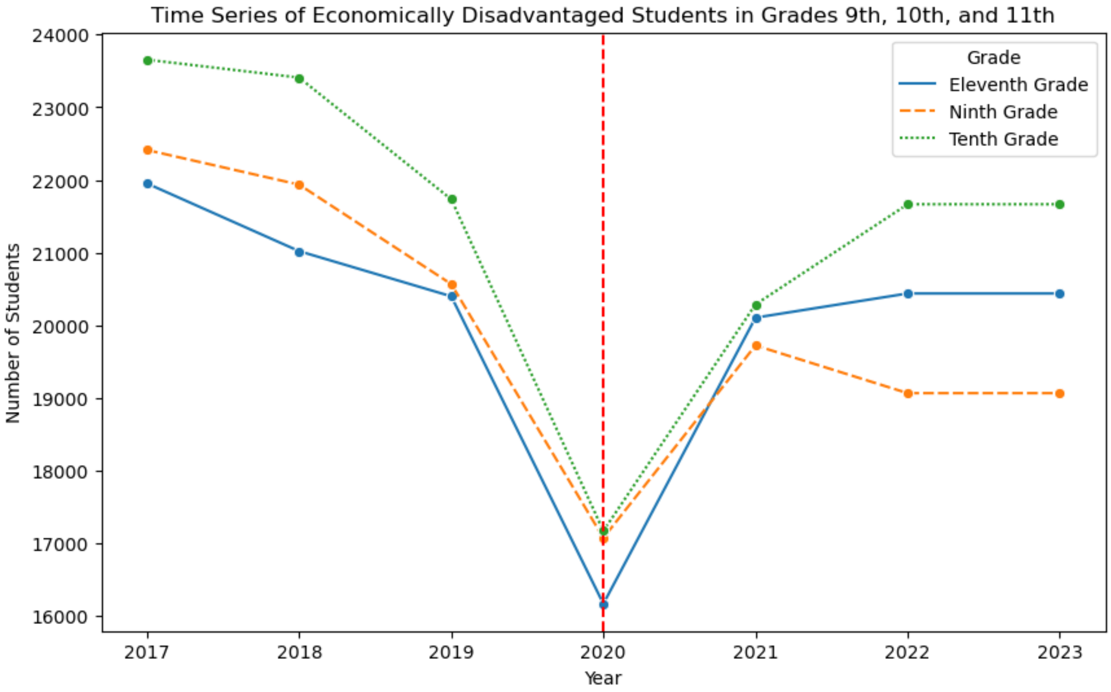
[18]:

```python
dataframes = [ melted_data17, melted_data18, melted_data19, melted_data20, melted_data21, melted_data22, melted_data23]

combined_data = pd.concat(dataframes, ignore_index=True)
print(combined_data)
```

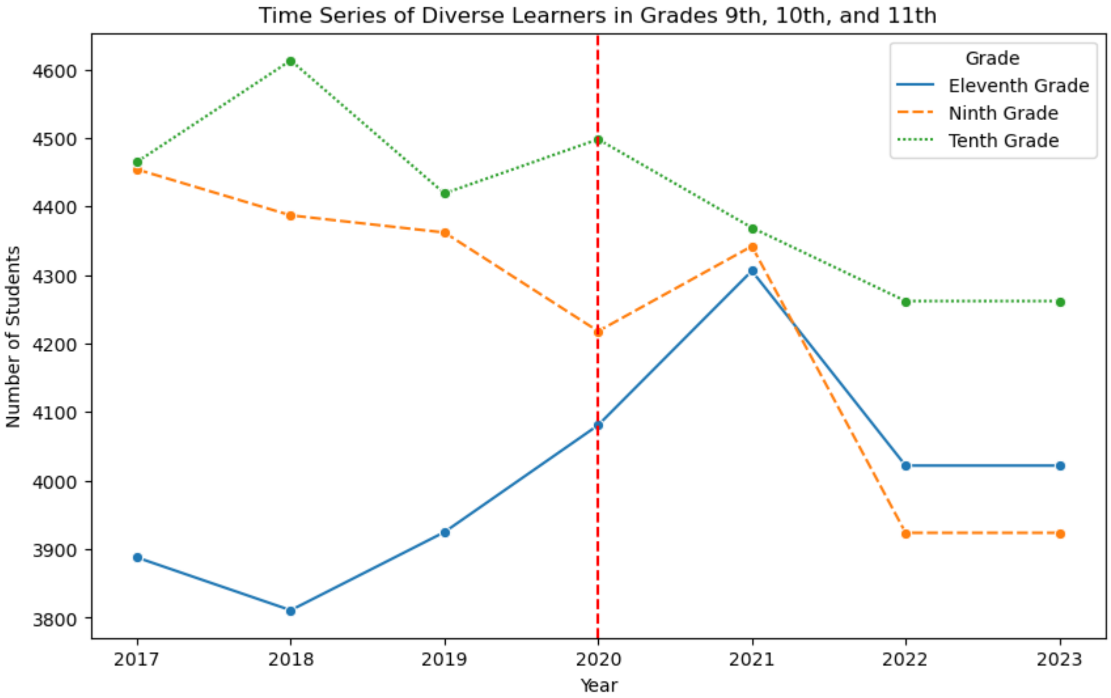
```
            Grade  Year                Student_Type     Count
0  District Total  2017                   Bilingual   67834.0
1     Ninth Grade  2017                   Bilingual    2695.0
2     Tenth Grade  2017                   Bilingual    2798.0
3  Eleventh Grade  2017                   Bilingual    2300.0
4  District Total  2017            Diverse Learners   50917.0
..            …     …                         …         …
79 Eleventh Grade  2023            Diverse Learners    4022.0
80 District Total  2023  Economically Disadvantaged  228466.0
81    Ninth Grade  2023  Economically Disadvantaged   19069.0
82    Tenth Grade  2023  Economically Disadvantaged   21667.0
83 Eleventh Grade  2023  Economically Disadvantaged   20440.0

[84 rows x 4 columns]
```

[19]:

```python
disadvantaged_students = combined_data[(combined_data['Student_Type'] == 'Economically Disadvantaged') & (combined_data['Grade'] != 'District Total')]
disadvantaged_students_grouped = disadvantaged_students.groupby(['Year', 'Grade'])['Count'].sum().unstack()

plt.figure(figsize=(10, 6))
ax = sns.lineplot(data=disadvantaged_students_grouped, marker='o')
plt.title('Time Series of Economically Disadvantaged Students in Grades 9th, 10th, and 11th')
plt.xlabel('Year')
plt.ylabel('Number of Students')
plt.legend(title='Grade')

plt.grid(False)

plt.axvline(x=2020, color='red', linestyle='--')

plt.show()
```

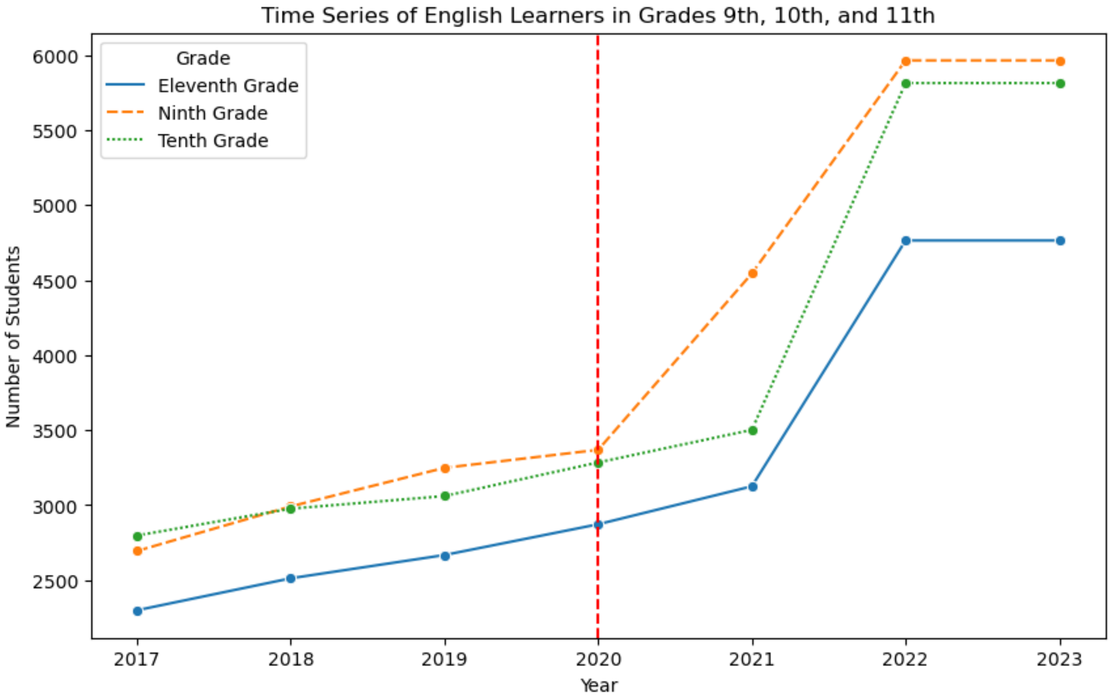


[20]:
```
diverse_learners = combined_data[(combined_data['Student_Type'] == 'Diverse␣
    ↪Learners') & (combined_data['Grade'] != 'District Total')]
diverse_learners_grouped = diverse_learners.groupby(['Year', 'Grade'])['Count'].
    ↪sum().unstack()

plt.figure(figsize=(10, 6))
ax = sns.lineplot(data=diverse_learners_grouped, marker='o')
plt.title('Time Series of Diverse Learners in Grades 9th, 10th, and 11th')
plt.xlabel('Year')
plt.ylabel('Number of Students')
plt.legend(title='Grade')

plt.grid(False)

plt.axvline(x=2020, color='red', linestyle='--')

plt.show()
```



[21]:

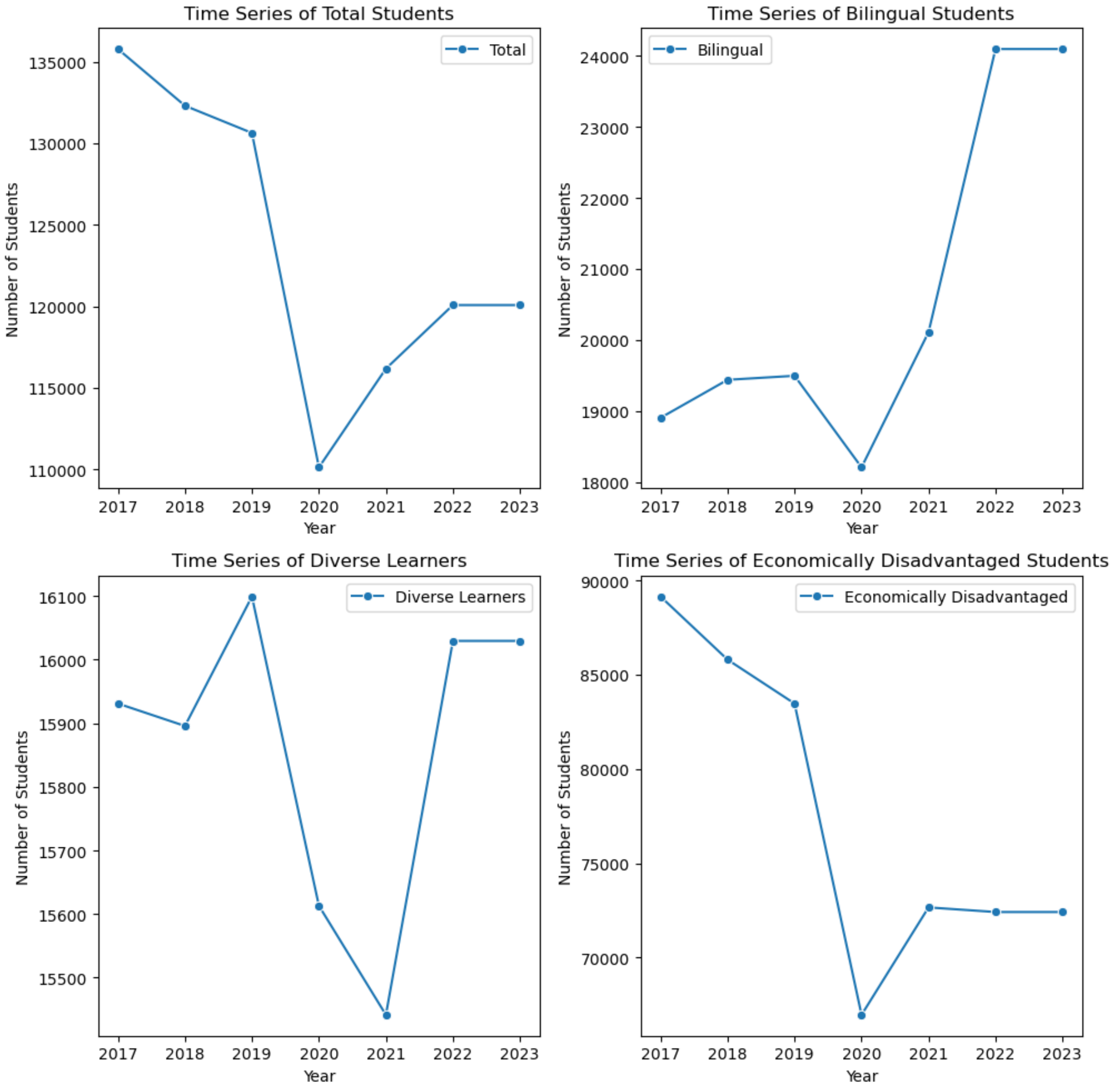
```
bilingual_students = combined_data[(combined_data['Student_Type'] == 'Bilingual') & (combined_data['Grade'] != 'District Total')]
bilingual_students_grouped = bilingual_students.groupby(['Year', 'Grade'])['Count'].sum().unstack()

plt.figure(figsize=(10, 6))
ax = sns.lineplot(data=bilingual_students_grouped, marker='o')
plt.title('Time Series of English Learners in Grades 9th, 10th, and 11th')
plt.xlabel('Year')
plt.ylabel('Number of Students')
plt.legend(title='Grade')

plt.grid(False)

plt.axvline(x=2020, color='red', linestyle='--')

plt.show()
```



[22]:

```
total_students = combined_data[combined_data['Grade'] == 'District Total']
bilingual_students = combined_data[combined_data['Student_Type'] == 'Bilingual']
diverse_students = combined_data[combined_data['Student_Type'] == 'Diverse␣
 ↪Learners']
disadvantaged_students = combined_data[combined_data['Student_Type'] ==␣
 ↪'Economically Disadvantaged']

fig, axes = plt.subplots(2, 2, figsize=(10, 10))

sns.lineplot(data=total_students, x='Year', y='Count', marker='o',␣
 ↪label='Total', err_style=None, ax=axes[0, 0])
axes[0, 0].set_title('Time Series of Total Students')
axes[0, 0].set_xlabel('Year')
axes[0, 0].set_ylabel('Number of Students')
axes[0, 0].legend()
axes[0, 0].grid(False)

sns.lineplot(data=bilingual_students, x='Year', y='Count', marker='o',␣
 ↪label='Bilingual', err_style=None, ax=axes[0, 1])
axes[0, 1].set_title('Time Series of Bilingual Students')
axes[0, 1].set_xlabel('Year')
axes[0, 1].set_ylabel('Number of Students')
axes[0, 1].legend()
axes[0, 1].grid(False)
```

```
sns.lineplot(data=diverse_students, x='Year', y='Count', marker='o',␣
 ↪label='Diverse Learners', err_style=None, ax=axes[1, 0])
axes[1, 0].set_title('Time Series of Diverse Learners')
axes[1, 0].set_xlabel('Year')
axes[1, 0].set_ylabel('Number of Students')
axes[1, 0].legend()
axes[1, 0].grid(False)

sns.lineplot(data=disadvantaged_students, x='Year', y='Count', marker='o',␣
 ↪label='Economically Disadvantaged', err_style=None, ax=axes[1, 1])
axes[1, 1].set_title('Time Series of Economically Disadvantaged Students')
axes[1, 1].set_xlabel('Year')
axes[1, 1].set_ylabel('Number of Students')
axes[1, 1].legend()
axes[1, 1].grid(False)

plt.tight_layout()
plt.show()
```


Time Series of Total Students
Total
135000
130000
125000
120000
115000
110000
Number of Students
2017
2018
2019
2020
2021
2022
2023
Year
Time Series of Bilingual Students
Bilingual
24000
23000
22000
21000
20000
19000
18000
Number of Students
2017
2018
2019
2020
2021
2022
2023
Year
Time Series of Diverse Learners
Diverse Learners
16100
16000
15900
15800
15700
15600
15500
Number of Students
2017
2018
2019
2020
2021
2022
2023
Year
Time Series of Economically Disadvantaged Students
Economically Disadvantaged
90000
85000
80000
75000
70000
Number of Students
2017
2018
2019
2020
2021
2022
2023
Year

[ ]: