PROJECT REPORT (PHASE-1)

Analyzing New York State's Graduation Rate

INTRODUCTION:

The New York State Education Department has released the "Graduation Rate in New York 2017-2018" dataset, which contains graduation rates and dropouts data for public high schools in the state. The dataset aims to provide valuable insights into the graduation rates and dropouts of students in public high schools across the state. Dropout rates in areas with high dropout rates and higher graduation rates in places with low graduation rates. The model we developed will project the typical graduation and dropout rates for the years 2017 to 2018, based on the data provided to it. 2019 dropout rates throughout all of New York state.

Data source: The dataset is taken from https://data.nysed.gov/downloads.php

```
In [117]: dm.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 116446 entries, 0 to 116445
Data columns (total 36 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   REPORT_SCHOOL_YEAR     116446 non-null  object
 1   AGGREGATION_INDEX      116446 non-null  int64
 2   AGGREGATION_TYPE       116446 non-null  object
 3   AGGREGATION_CODE       116446 non-null  float64
 4   AGGREGATION_NAME       116446 non-null  object
 5   LEA_BEDS               73587 non-null   float64
 6   LEA_NAME               73587 non-null   object
 7   NRC_CODE               112240 non-null  float64
 8   NRC_DESC               112240 non-null  object
 9   COUNTY_CODE            115878 non-null  float64
 10  COUNTY_NAME            115878 non-null  object
 11  NYC_IND                115878 non-null  float64
 12  BOCES_CODE             111744 non-null  float64
 13  BOCES_NAME             111744 non-null  object
 14  MEMBERSHIP_CODE        116446 non-null  int64
 15  MEMBERSHIP_KEY         116446 non-null  int64
 16  MEMBERSHIP_DESC        116446 non-null  object
 17  SUBGROUP_CODE          116446 non-null  int64
 18  SUBGROUP_NAME          116446 non-null  object
 19  ENROLL_CNT             116446 non-null  object
 20  GRAD_CNT               116446 non-null  object
 21  GRAD_PCT               116446 non-null  object
```

Data Cleaning/Processing:

1. Missing Data: The data frame shows the missing values that need to be removed. This helps identify columns further cleaning or investigation.

In [118]:
```
## 1 Missing data##
```

In [119]:
```
total = dm.shape[0]
missing_columns = [colu for colu in dm.columns if dm[colu].isnull().sum() > 0]
for colu in missing_columns:
    null_count = dm[colu].isnull().sum()
    per = (null_count/total) * 100
    print(f"{colu}: {null_count} ({round(per, 3)}%)")
total
```

```
LEA_BEDS: 42859 (36.806%)
LEA_NAME: 42859 (36.806%)
NRC_CODE: 4206 (3.612%)
NRC_DESC: 4206 (3.612%)
COUNTY_CODE: 568 (0.488%)
COUNTY_NAME: 568 (0.488%)
NYC_IND: 568 (0.488%)
BOCES_CODE: 4702 (4.038%)
BOCES_NAME: 4702 (4.038%)
```

Out[119]: 116446

In [120]:
```
# Now removing the missing data rows in columns
```

In [121]:
```
dm =dm.dropna(subset=['COUNTY_CODE','COUNTY_NAME','BOCES_CODE','BOCES_NAME'])
dm
```

Out[121]:

	REPORT_SCHOOL_YEAR	AGGREGATION_INDEX	AGGREGATION_TYPE	AGGREGATION_CODE	AGGREGATION_NAME	LEA_BEDS	LEA_NAME	NRC_(
640	2017-18	3	District	1.010001e+10	ALBANY CITY SCHOOL DISTRICT	NaN	NaN	
641	2017-18	3	District	1.010001e+10	ALBANY CITY SCHOOL DISTRICT	NaN	NaN	
642	2017-18	3	District	1.010001e+10	ALBANY CITY SCHOOL DISTRICT	NaN	NaN	

2.Duplicate values or Rows: The data frame has been checked for Duplicate values or rows has been removed. This helps to ensure quality data and accuracy.

```
In [122]: ## 2 Duplicates##
```

```
In [123]: print(f"Number of duplicate rows: {dm.duplicated().sum()}")

Number of duplicate rows: 455
```

```
In [124]: #removing Duplicates
dm = dm.drop_duplicates(keep=False)
print(f"Number of duplicate rows: {dm.duplicated().sum()}")

Number of duplicate rows: 0
```

3. Remove Unwanted Columns: Removing unwanted columns can simplify and streamline data analysis. This will help to protect sensitive data.

```
In [125]: ## 3 Remove useless columns##
```

```
In [126]: dm = dm.drop(['AGGREGATION_CODE','LEA_BEDS','NYC_IND','MEMBERSHIP_CODE','MEMBERSHIP_KEY','LEA_NAME','NRC_CODE','NRC_DESC','MEMBE
dm
```

Out[126]:

	REPORT_SCHOOL_YEAR	AGGREGATION_INDEX	AGGREGATION_TYPE	AGGREGATION_NAME	COUNTY_CODE	COUNTY_NAME	BOCES_CODE	
640	2017-18	3	District	ALBANY CITY SCHOOL DISTRICT	1.0	ALBANY	190.0	S SCHO
641	2017-18	3	District	ALBANY CITY SCHOOL DISTRICT	1.0	ALBANY	190.0	S SCHO

4. Remove Special Characters: Removing special characters that can help standardize and clean data and makes the data more uniform. This will help to analyze the data easier.

```
In [128]: ## 4 Eliminating special characters (%)  from Data ##
```

```
In [129]: dm['GED_PCT'].replace(regex=True,inplace=True,to_replace=r'\D',value=r'')
dm['DROPOUT_PCT'].replace(regex=True,inplace=True,to_replace=r'\D',value=r'')
dm['GRAD_PCT'].replace(regex=True,inplace=True,to_replace=r'\D',value=r'')
dm['STILL_ENR_PCT'].replace(regex=True,inplace=True,to_replace=r'\D',value=r'')
dm['LOCAL_PCT'].replace(regex=True,inplace=True,to_replace=r'\D',value=r'')
dm
```

Out[129]:

	REPORT_SCHOOL_YEAR	AGGREGATION_INDEX	AGGREGATION_TYPE	AGGREGATION_NAME	COUNTY_CODE	COUNTY_NAME	BOCES_CODE
640	2017-18	3	District	ALBANY CITY SCHOOL DISTRICT	1.0	ALBANY	190.0
641	2017-18	3	District	ALBANY CITY SCHOOL DISTRICT	1.0	ALBANY	190.0
642	2017-18	3	District	ALBANY CITY SCHOOL DISTRICT	1.0	ALBANY	190.0
643	2017-18	3	District	ALBANY CITY SCHOOL DISTRICT	1.0	ALBANY	190.0

5. Changing the order of Data Frame columns: The use of changing the columns is to arrange the data frame columns in the order that is most useful for data analysis or visualization.

In [130]:
```
## 5 Changing the order of DataFrame columns##
```

In [131]:
```
cols = dm.columns.tolist()
cols = cols[-1:] + cols[:-1]
dm = dm[cols]
dm
```

Out[131]:

	DROPOUT_PCT	REPORT_SCHOOL_YEAR	AGGREGATION_INDEX	AGGREGATION_TYPE	AGGREGATION_NAME	COUNTY_CODE	COUNTY_NAME	B
640	25	2017-18	3	District	ALBANY CITY SCHOOL DISTRICT	1.0	ALBANY	
641	21	2017-18	3	District	ALBANY CITY SCHOOL DISTRICT	1.0	ALBANY	
642	29	2017-18	3	District	ALBANY CITY SCHOOL DISTRICT	1.0	ALBANY	
643		2017-18	3	District	ALBANY CITY SCHOOL DISTRICT	1.0	ALBANY	

6. Improving the presentation of columns by formatting: The use of formatting the presentation of columns in data frame is make the data frame easier to view.

In [132]:
```
## 6 improving the presentation of columns by formatting
```

In [133]:
```
dm['COUNTY_NAME'] = dm['COUNTY_NAME'].str.capitalize()
dm['BOCES_NAME'] = dm['BOCES_NAME'].str.capitalize()
dm['AGGREGATION_NAME'] = dm['AGGREGATION_NAME'].str.capitalize()
dm
```

Out[133]:

	DROPOUT_PCT	REPORT_SCHOOL_YEAR	AGGREGATION_INDEX	AGGREGATION_TYPE	AGGREGATION_NAME	COUNTY_CODE	COUNTY_NAME	B
640	25	2017-18	3	District	Albany city school district	1.0	Albany	
641	21	2017-18	3	District	Albany city school district	1.0	Albany	
642	29	2017-18	3	District	Albany city school district	1.0	Albany	
643		2017-18	3	District	Albany city school district	1.0	Albany	

7. Changing subgroup to proper values: The letters M and F, which stand for male and female, respectively, are among the entries in the subgroup name field. I renamed them M and F with the appropriate values to prevent misunderstandings.

In [134]:
```
## 7 To avoid confusion, change M and F to the proper values(0,1) in the SUBGROUP NAME column.
# F=0
# M=1
dm['SUBGROUP_NAME'] =dm['SUBGROUP_NAME'].replace(['F'],'0')
dm['SUBGROUP_NAME'] = dm['SUBGROUP_NAME'].replace(['M'],'1')
dm
```

Out[134]:

AGGREGATION_NAME	COUNTY_CODE	COUNTY_NAME	BOCES_CODE	BOCES_NAME	SUBGROUP_CODE	...	ENROLL_CNT	GRAD_CNT	GRAD_PCT	LOCAL_(
Albany city school district	1.0	Albany	190.0	Albany-schenectady-schoharie(capital region)	1	...	660	465	70	
Albany city school district	1.0	Albany	190.0	Albany-schenectady-schoharie(capital region)	2	...	335	255	76	
Albany city school district	1.0	Albany	190.0	Albany-schenectady-schoharie(capital region)	3	...	325	210	65	
Albany city school district	1.0	Albany	190.0	Albany-schenectady-schoharie(capital region)	4	...	-	-		

8. Indexing: Here, performing to create a data frame proper index order, which implies restoring the index after conversion is finished will guarantee precise ordering.

In [135]:
```
## 8 indexing ##
```

In [136]:
```
dm.reset_index(drop=True, inplace=True)
dm
```

Out[136]:

	DROPOUT_PCT	REPORT_SCHOOL_YEAR	AGGREGATION_INDEX	AGGREGATION_TYPE	AGGREGATION_NAME	COUNTY_CODE	COUNTY_NAME	BOCE
0	25	2017-18	3	District	Albany city school district	1.0	Albany	
1	21	2017-18	3	District	Albany city school district	1.0	Albany	
2	29	2017-18	3	District	Albany city school district	1.0	Albany	
3		2017-18	3	District	Albany city school district	1.0	Albany	
4	25	2017-18	3	District	Albany city school district	1.0	Albany	

9. Converting Data Types: To facilitate further processing, most of the numerical fields' datatypes were changed from object to float in the dataset's numerical columns.

```
In [ ]: ## 9 Converting the datatypes
```

```
In [144]:
dm['ENROLL_CNT'] = pd.to_numeric(dm['ENROLL_CNT'],errors='coerce')
dm['GRAD_CNT'] = pd.to_numeric(dm['GRAD_CNT'],errors='coerce')
dm['GRAD_PCT'] = pd.to_numeric(dm['GRAD_PCT'],errors='coerce')
dm['LOCAL_CNT'] = pd.to_numeric(dm['LOCAL_CNT'],errors='coerce')
dm['LOCAL_PCT'] = pd.to_numeric(dm['LOCAL_PCT'],errors='coerce')
dm['GED_CNT'] = pd.to_numeric(dm['GED_CNT'],errors='coerce')
dm['GED_PCT'] = pd.to_numeric(dm['GED_PCT'],errors='coerce')
dm['DROPOUT_CNT'] = pd.to_numeric(dm['DROPOUT_CNT'],errors='coerce')
dm['DROPOUT_PCT'] = pd.to_numeric(dm['DROPOUT_PCT'],errors='coerce')
dm['GRAD_CNT'] = pd.to_numeric(dm['GRAD_CNT'],errors='coerce')
dm.dtypes
dm
```

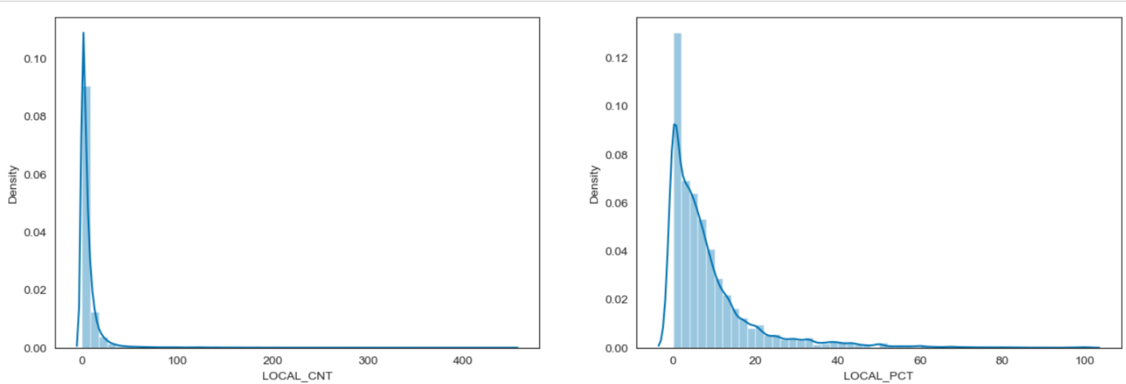
Out[144]:

	DROPOUT_PCT	REPORT_SCHOOL_YEAR	AGGREGATION_INDEX	AGGREGATION_TYPE	AGGREGATION_NAME	COUNTY_CODE	COUNTY_NAME	BOCE
0	25.0	2017-18	3	District	Albany city school district	1.0	Albany	
1	21.0	2017-18	3	District	Albany city school district	1.0	Albany	
2	29.0	2017-18	3	District	Albany city school district	1.0	Albany	

10 Finding outliners using Z-score-based technique: The data points in a data frame are normally distributed, the Z-score-based technique can be used to identify outliers. A data point that deviates significantly from the mean is referred to as an outlier.

```
In [273]: ## 10 Finding outliners using Z-score-based technique!##
```

```
In [274]: plt.figure(figsize=(16,5))
plt.subplot(1,2,1)
sns.distplot(dm['LOCAL_CNT'])
plt.subplot(1,2,2)
sns.distplot(dm['LOCAL_PCT'])
plt.show()
```



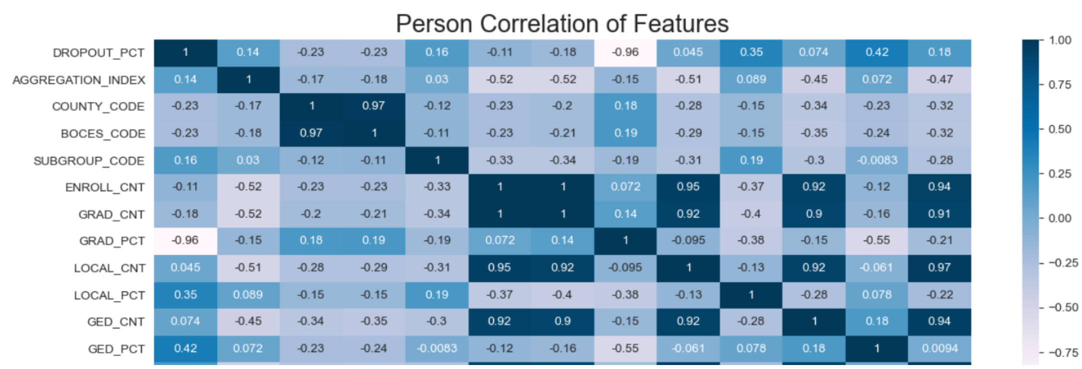
Exploratory Data Analysis:

1.Correlation using pearson method: It is a statistical method that measures the strength and direction of the relationship between variables, where variables are continuous.

```
In [281]: ##finding correlation using pearson method
```

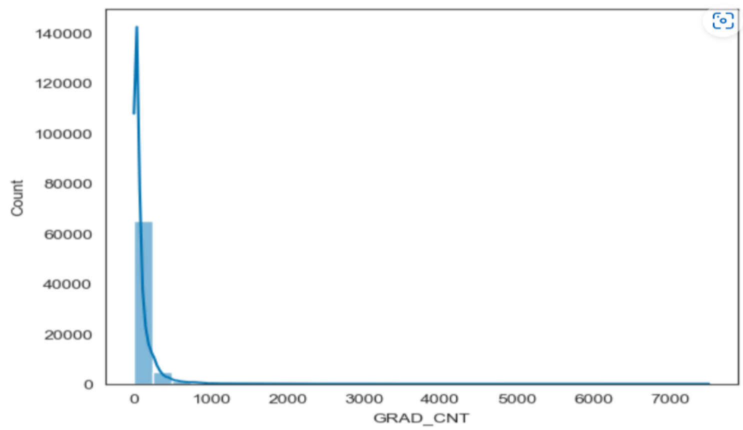
```
In [282]: corr = dm.corr(method = 'pearson')
          colormap = plt.cm.PuBu
          plt.figure(figsize=(15,5))
          plt.title("Pearson Correlation of Features", y = 1, size = 20)
          sns.heatmap(corr.astype(float).corr(), linecolor = "white", cmap = colormap, annot = True)

Out[282]: <AxesSubplot:title={'center':'Person Correlation of Features'}>
```



2. Hist Plot: The histogram is used to calculate the total number of items. (GRAD CNT) is the determinant. This will help us with our value analysis for each of the dependent variables, as well as for a large number of them.

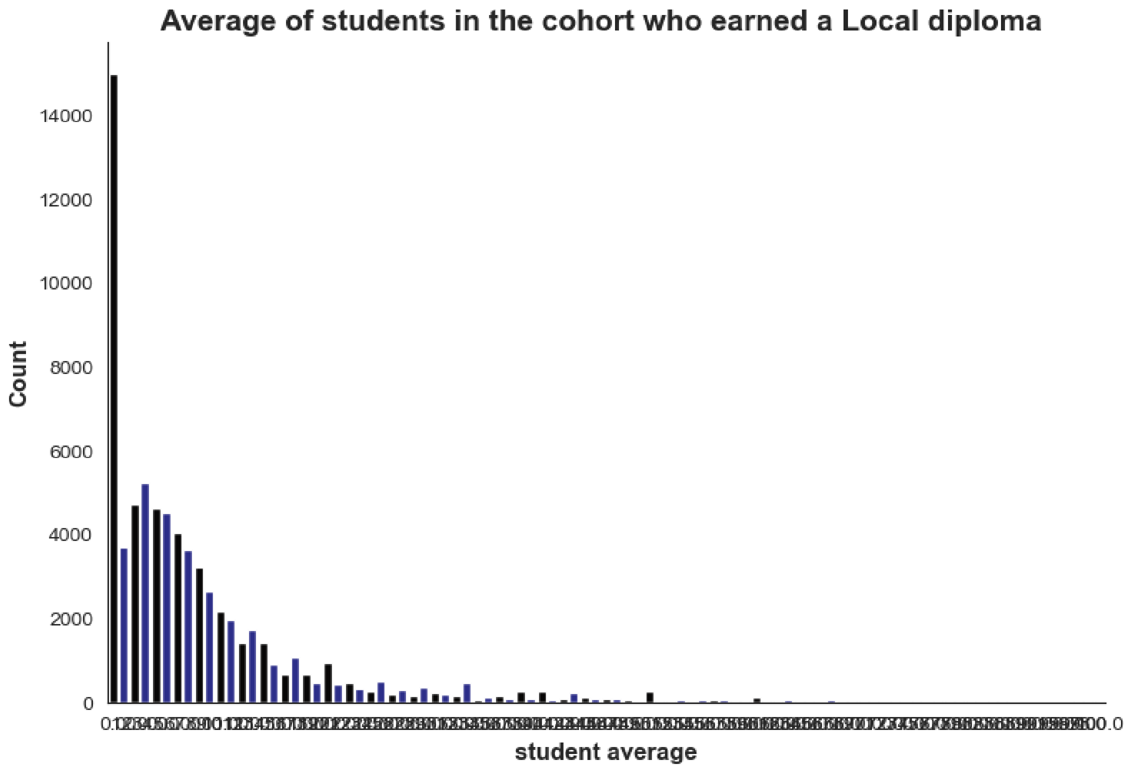
```
sns.histplot(dm['GRAD_CNT'],kde=True,bins=30)
plt.show()
```



3. Visualization: Visualization checking the column's visual representation to see how many are above or below normal. As a result, we now know how many there are generally.

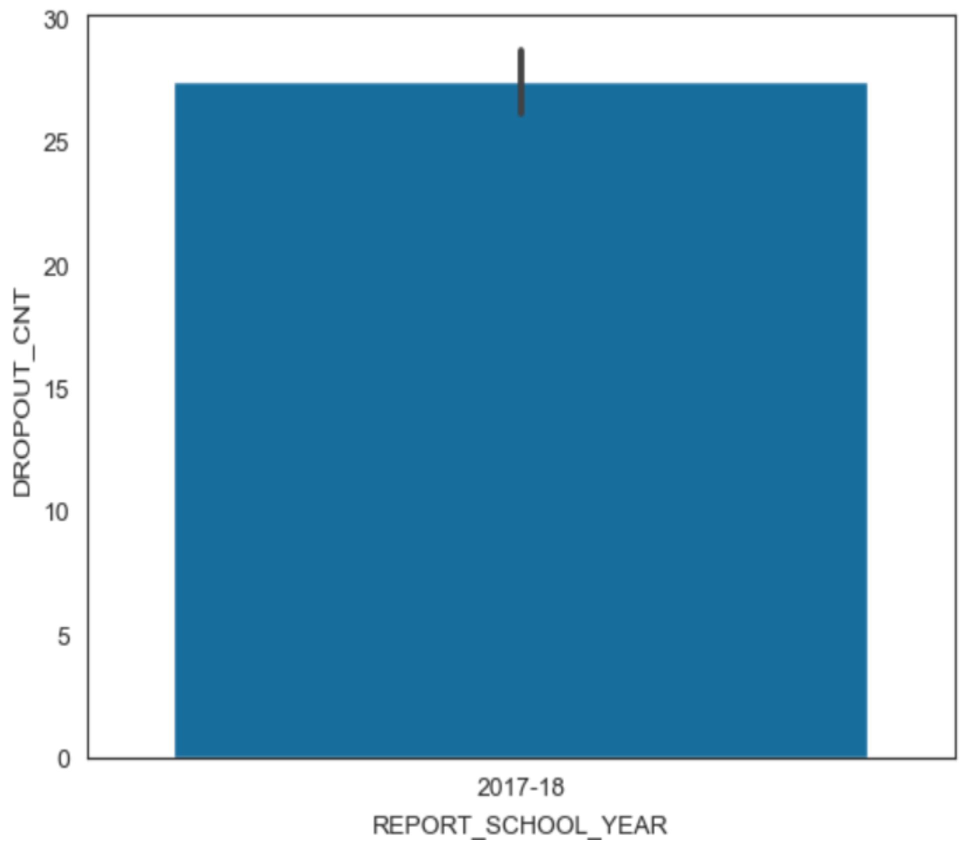
Out[285]:

	DROPOUT_PCT	AGGREGATION_INDEX	COUNTY_CODE	BOCES_CODE	SUBGROUP_CODE	ENROLL_CNT	GRAD_CNT	GRAD_PCT	LOCAL_CNT
count	71743.000000	110834.000000	110834.000000	110834.000000	110834.000000	71743.000000	71743.000000	71743.000000	71743.000000
mean	8.458456	3.655728	34.974692	3431.541937	9.143458	121.950950	102.340200	81.963648	6.918097
std	11.011031	0.475132	17.635556	1700.622960	5.202226	264.486269	216.540825	19.560251	16.739034
min	0.000000	3.000000	1.000000	190.000000	1.000000	5.000000	0.000000	0.000000	0.000000
25%	1.000000	3.000000	26.000000	2490.000000	5.000000	23.000000	18.000000	76.000000	1.000000
50%	5.000000	4.000000	33.000000	3090.000000	9.000000	56.000000	45.000000	88.000000	3.000000
75%	12.000000	4.000000	49.000000	4590.000000	13.000000	122.000000	105.000000	95.000000	7.000000
max	100.000000	4.000000	68.000000	6690.000000	18.000000	9114.000000	7500.000000	100.000000	452.000000

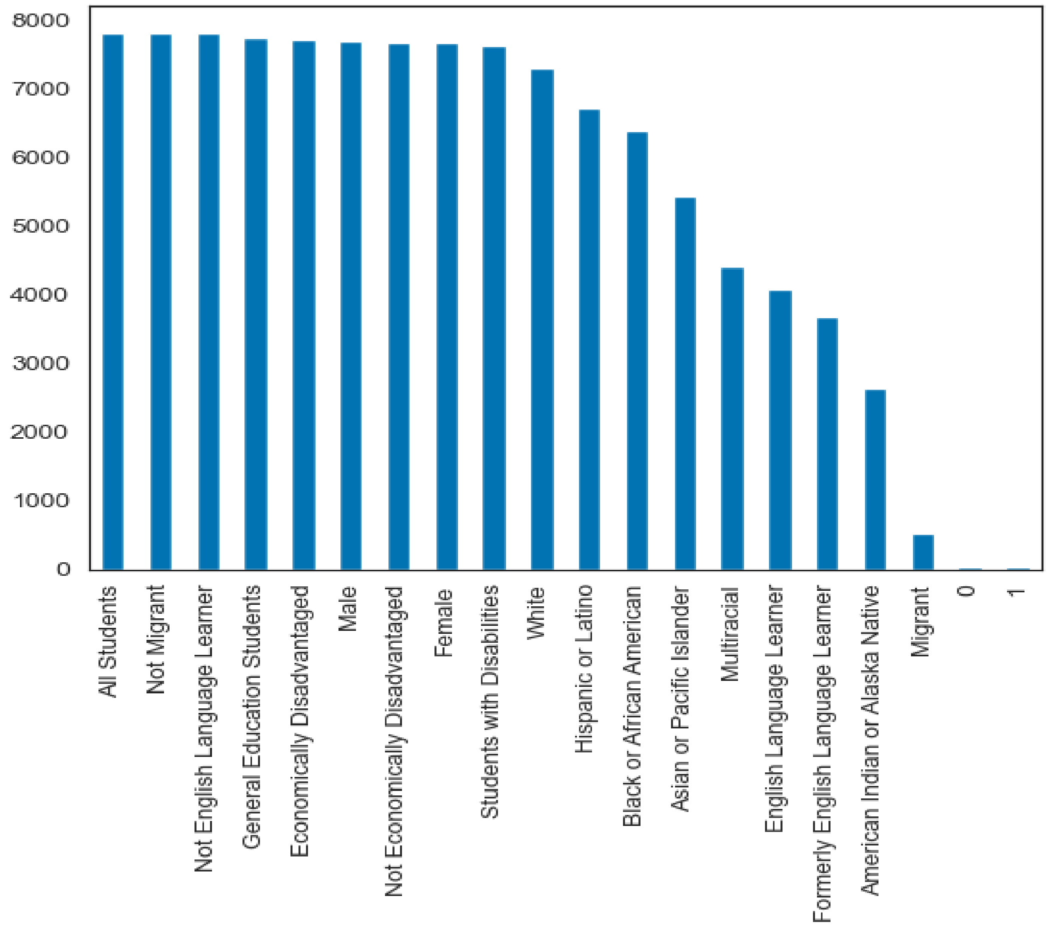


4. Dropout rate in 2018: The dropout rate for that category in a separate column. Make sure that the data is properly formatted and labeled.

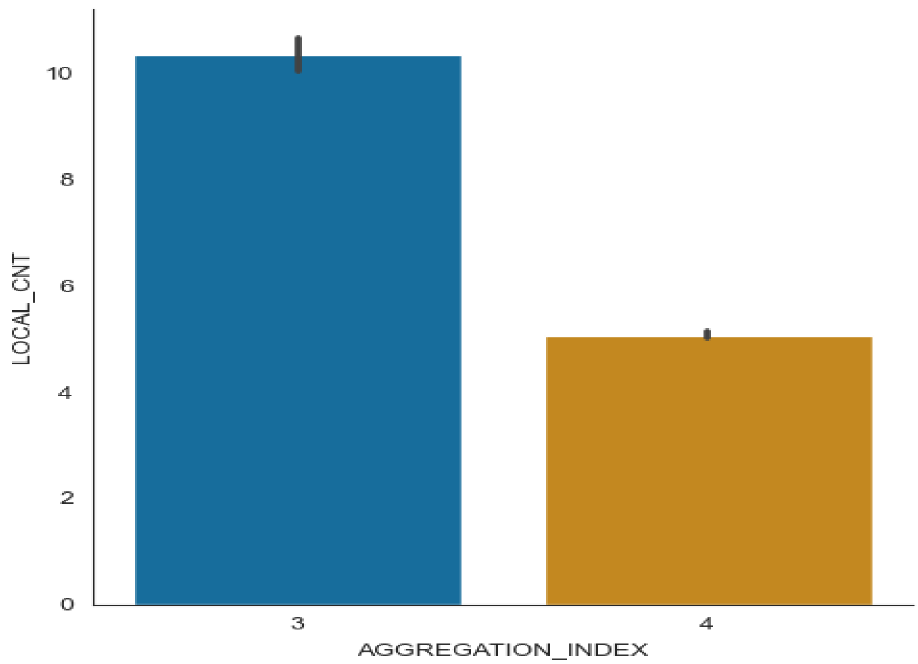
In [299]:
```
sns.barplot(x='REPORT_SCHOOL_YEAR',y='DROPOUT_CNT',data=dm,estimator=np.std)
plt.show()
```



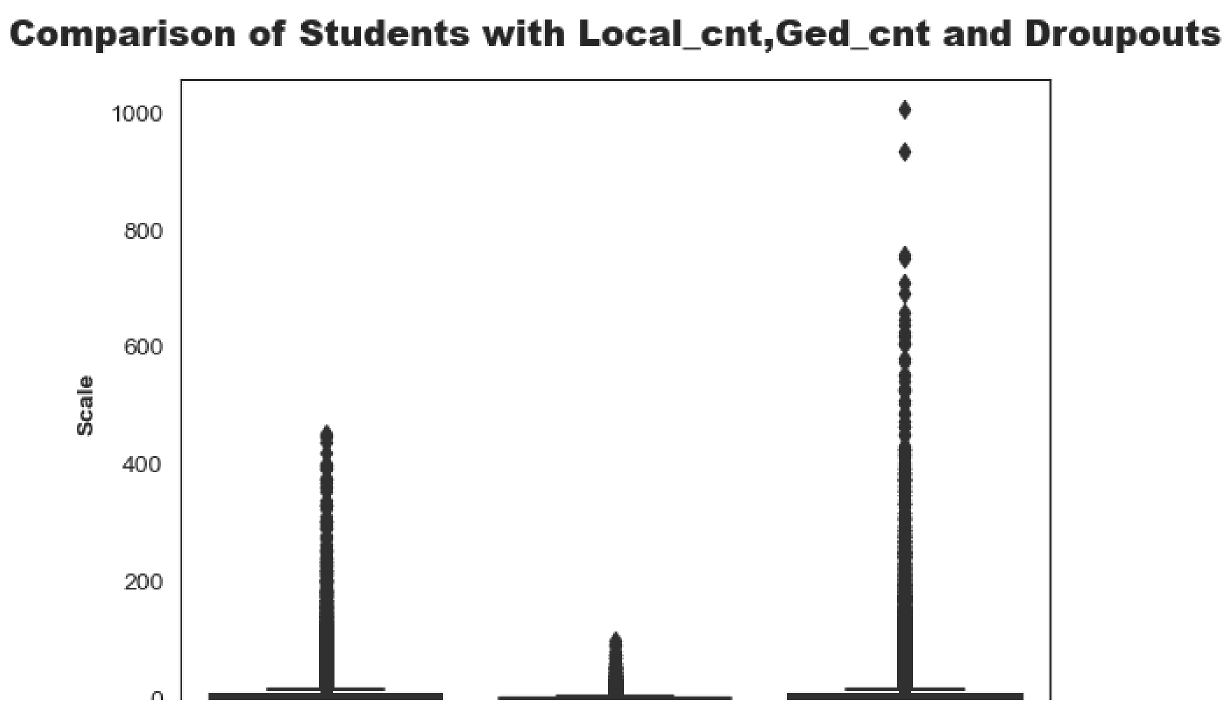
5.Bar Graph: A bar graph has subgroup name on the y-axis and all values on the x-axis. Plotting dropout cnt on the y-axis with the maximum subgroup name and the minimum subgroup name and which year has the highest dropout cnt and lowest dropout cnt will reveal which year exhibits the highest and lowest count.



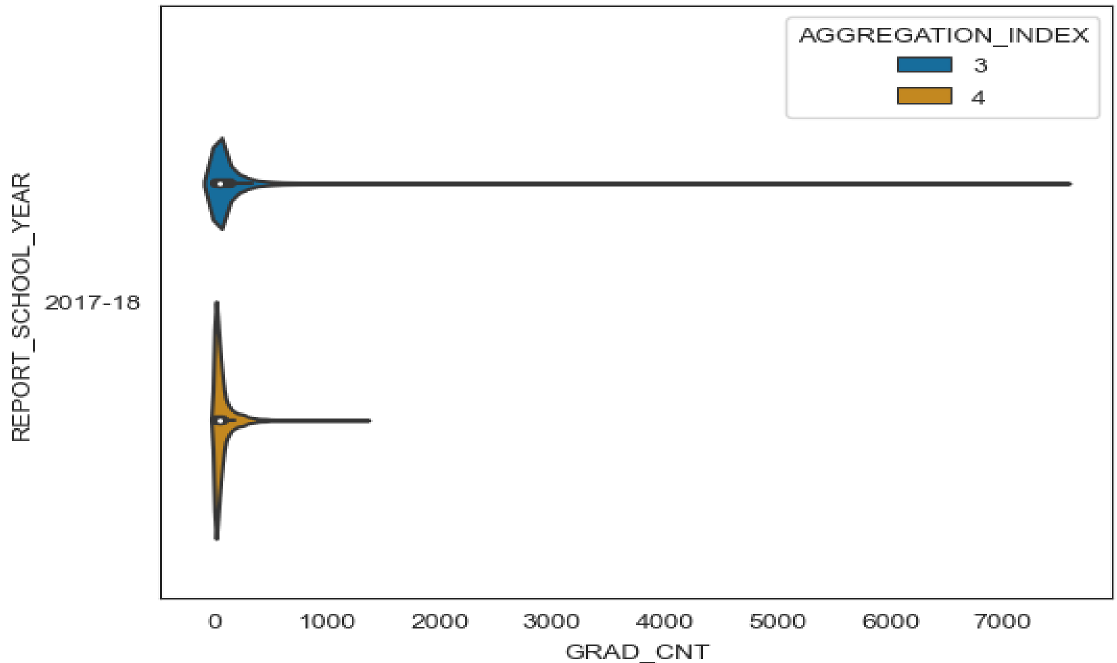
6. Catplot: Determine the relationship among the local_cnt and the aggregation index using a cat plot. To analyze this relationship, we use the cat plot.



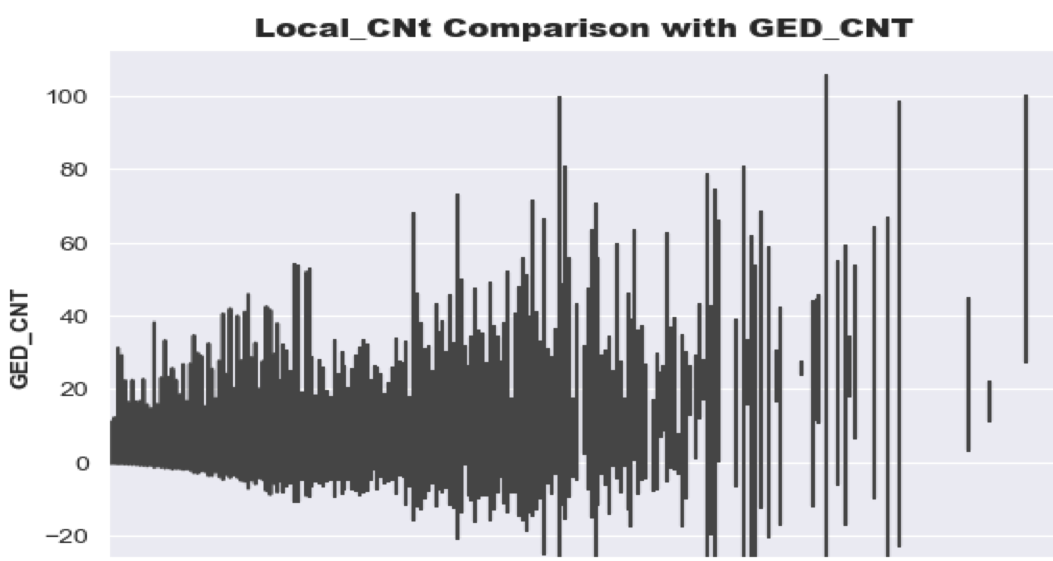
7. Comparison of Students with Local_cnt,Ged_cnt and Droupouts: Here comparing the students with scale on Y-axis and Local_cnt,Ged_cnt and Droupouts on X-axis.



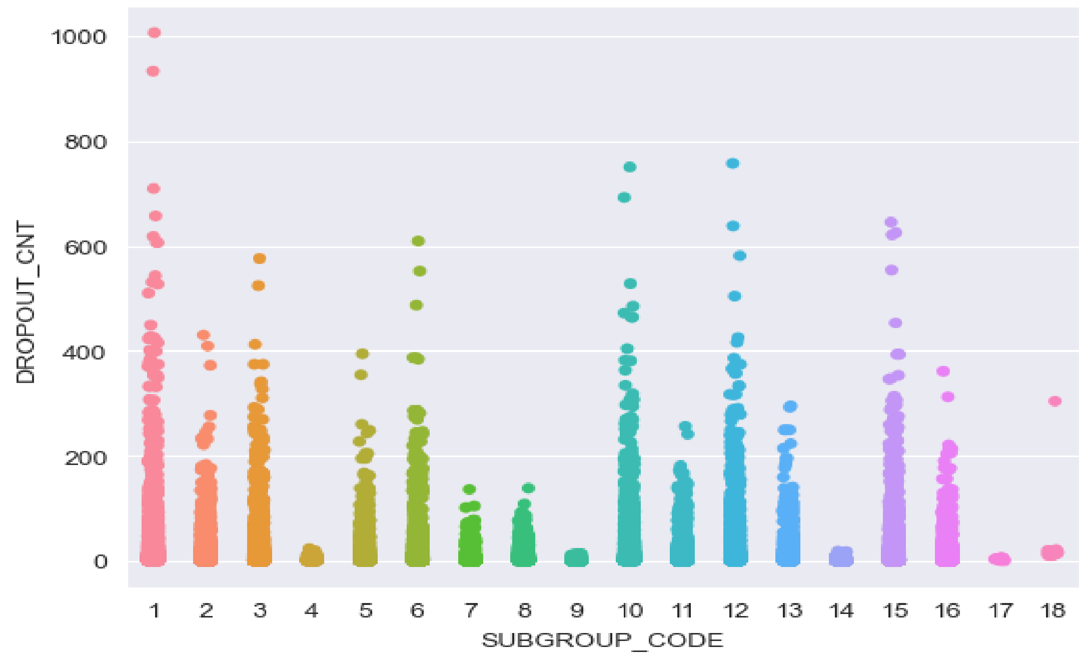
8. Violin Plot: A violin plot is a type of data visualization that shows how a continuous variable is distributed across different categories or groups. On X-axis Grad_cnt and on Y-axis Report_school Year.



9. Distinction in Local_CNt Comparison with Ged_Cnt: It shows comparison b/w Local Cnt and Ged_cnt



10.Strip Plot: It is basically a scatter plot that differentiates different categories. So, the data that corresponds to each category is shown as a scatter plot, and all the observations and collected data that are visualized are shown, side-by-side on a single graph.



11. Heat Maps: Heatmaps are most useful for identifying patterns in large amounts of data at a glance.