ThutoNet

Quality of STEM education and financial literacy education in primary and secondary schools in South Africa particularly those in underserved communities

First, let's load the CSV file and take a look at its contents:

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

# Load the CSV file
Data = pd.read_csv('Free State.csv', encoding='utf-8')

# Display basic information about the dataset
print(Data.info())

# Display the first few rows of the dataset
Data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1021 entries, 0 to 1020
Data columns (total 48 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   NatEmis                     1021 non-null   int64
 1   Datayear                    1021 non-null   int64
 2   Province                    1021 non-null   object
 3   ProvinceCD                  1021 non-null   int64
 4   Official_Institution_Name   1021 non-null   object
 5   Status                      1021 non-null   object
 6   Sector                      1021 non-null   object
 7   Type_DoE                    1021 non-null   object
 8   Phase_PED                   1021 non-null   object
 9   Specialisation              1021 non-null   object
 10  EIDistrict                  1021 non-null   object
 11  EICircuit                   994 non-null    float64
 12  OwnerLand                   1021 non-null   object
 13  OwnerBuild                  1021 non-null   object
 14  ExDept                      1021 non-null   object
 15  Persal_PaypointNo           889 non-null    float64
 16  Persal_ComponentNo          939 non-null    float64
 17  ExamNo                      460 non-null    float64
 18  ExamCentre                  1021 non-null   object
 19  GIS_Longitude               1017 non-null   float64
 20  GIS_Latitude                1017 non-null   float64
 21  DMunName                    1021 non-null   object
 22  LMunName                    1021 non-null   object
 23  Ward_ID                    972 non-null    float64
 24  SP_Code                    932 non-null    float64
 25  SP_Name                    929 non-null    object
 26  Addressee                  965 non-null    object
 27  Township_Village           812 non-null    object
 28  Suburb                     812 non-null    object
 29  Town_City                  1016 non-null   object
 30  StreetAddress              1021 non-null   object
 31  PostalAddress              1021 non-null   object
 32  Telephone                  1021 non-null   object
 33  Section21                  1021 non-null   object
 34  Section21_Function         1021 non-null   object
 35  Quintile                   937 non-null    object
 36  NAS                        1021 non-null   object
 37  NodalArea                  1021 non-null   object
 38  Registration_Date          0 non-null      float64
 39  NoFeeSchool                1021 non-null   object
 40  Urban_Rural                1021 non-null   object
 41  Allocation                 0 non-null      float64
 42  DemarcationFrom            1021 non-null   object
 43  DemarcationTo              1021 non-null   object
 44  OldNATEMIS                 1021 non-null   int64
 45  NewNATEMIS                 1021 non-null   int64
 46  Learners2023               1021 non-null   int64
 47  Educators2023              1021 non-null   int64
dtypes: float64(10), int64(7), object(31)
memory usage: 383.0+ KB
None

     NatEmis  Datayear Province  ProvinceCD Official_Institution_Name
Status  \
0  440101017      2023       FS           4              IMPUCUKO P/S
OPEN
1  440101018      2023       FS           4               THABANG P/S
OPEN
2  440101019      2023       FS           4               UTOPIA PF/S
OPEN
3  440101042      2023       FS           4                ARRAN PF/S
OPEN
4  440101057      2023       FS           4  DIHLABENG CHRISTIAN PI/S
OPEN

        Sector         Type_DoE       Phase_PED   Specialisation  ...
\
0       PUBLIC  ORDINARY SCHOOL  PRIMARY SCHOOL  ORDINARY SCHOOL  ...

1       PUBLIC  ORDINARY SCHOOL  PRIMARY SCHOOL  ORDINARY SCHOOL  ...

2       PUBLIC  ORDINARY SCHOOL  PRIMARY SCHOOL  ORDINARY SCHOOL  ...
```

```
3       PUBLIC  ORDINARY SCHOOL  PRIMARY SCHOOL  ORDINARY SCHOOL  ...

4  INDEPENDENT  ORDINARY SCHOOL  PRIMARY SCHOOL  ORDINARY SCHOOL  ...


  Registration_Date  NoFeeSchool Urban_Rural Allocation
DemarcationFrom  \
0               NaN          YES       Urban        NaN
FS
1               NaN          YES       Urban        NaN
FS
2               NaN          YES       Rural        NaN
FS
3               NaN          YES       Rural        NaN
FS
4               NaN           NO       Urban        NaN
FS

  DemarcationTo  OldNATEMIS  NewNATEMIS Learners2023  Educators2023
0            FS           0   440101017          738             24
1            FS           0   440101018          982             30
2            FS           0   440101019          104              4
3            FS           0   440101042          258              9
4            FS           0   440101057          166             14

[5 rows x 48 columns]
```

The dataset contains information about schools in the Free State province of South Africa, including details about educators, learners, and school characteristics

Let's proceed with cleaning the data, visualizing educator distribution, analyzing STEM education quality, examining financial literacy education, and investigating schools in underserved communities. We'll start with data cleaning and then move on to the other aspects

Cleaning the data

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# Load the CSV file
Data = pd.read_csv('Free State.csv', encoding='utf-8')

# Check for missing values
missing_values = Data.isnull().sum()
print("Missing values:\
", missing_values[missing_values > 0])

# Check for duplicate rows
duplicate_rows = Data.duplicated().sum()
```

```
print(f"\
Number of duplicate rows: {duplicate_rows}")

# Convert 'Datayear' to datetime
Data['Datayear'] = pd.to_datetime(Data['Datayear'], format='%Y')

# Convert 'Learners2023' and 'Educators2023' to numeric, replacing any
non-numeric values with NaN
Data['Learners2023'] = pd.to_numeric(Data['Learners2023'],
errors='coerce')
Data['Educators2023'] = pd.to_numeric(Data['Educators2023'],
errors='coerce')

# Calculate student-teacher ratio
Data['StudentTeacherRatio'] = Data['Learners2023'] /
Data['Educators2023']

print("\
Dataframe info after cleaning:")
print(Data.info())

print("\
Sample of cleaned data:")
Data.head()
```

```
Missing values: EICircuit                27
Persal_PaypointNo      132
Persal_ComponentNo      82
ExamNo                 561
GIS_Longitude            4
GIS_Latitude             4
Ward_ID                 49
SP_Code                 89
SP_Name                 92
Addressee               56
Township_Village       209
Suburb                 209
Town_City                5
Quintile                84
Registration_Date     1021
Allocation            1021
dtype: int64
Number of duplicate rows: 0
Dataframe info after cleaning:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1021 entries, 0 to 1020
Data columns (total 49 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   NatEmis                    1021 non-null   int64
 1   Datayear                   1021 non-null   datetime64[ns]
 2   Province                   1021 non-null   object
 3   ProvinceCD                 1021 non-null   int64
 4   Official_Institution_Name  1021 non-null   object
 5   Status                     1021 non-null   object
 6   Sector                     1021 non-null   object
 7   Type_DoE                   1021 non-null   object
 8   Phase_PED                  1021 non-null   object
 9   Specialisation             1021 non-null   object
 10  EIDistrict                 1021 non-null   object
 11  EICircuit                  994 non-null    float64
 12  OwnerLand                  1021 non-null   object
 13  OwnerBuild                 1021 non-null   object
 14  ExDept                     1021 non-null   object
 15  Persal_PaypointNo          889 non-null    float64
 16  Persal_ComponentNo         939 non-null    float64
 17  ExamNo                     460 non-null    float64
 18  ExamCentre                 1021 non-null   object
 19  GIS_Longitude              1017 non-null   float64
 20  GIS_Latitude               1017 non-null   float64
 21  DMunName                   1021 non-null   object
 22  LMunName                   1021 non-null   object
 23  Ward_ID                    972 non-null    float64
 24  SP_Code                    932 non-null    float64
 25  SP_Name                    929 non-null    object
 26  Addressee                  965 non-null    object
 27  Township_Village           812 non-null    object
 28  Suburb                     812 non-null    object
 29  Town_City                  1016 non-null   object
 30  StreetAddress              1021 non-null   object
 31  PostalAddress              1021 non-null   object
 32  Telephone                  1021 non-null   object
 33  Section21                  1021 non-null   object
 34  Section21_Function         1021 non-null   object
 35  Quintile                   937 non-null    object
 36  NAS                        1021 non-null   object
 37  NodalArea                  1021 non-null   object
 38  Registration_Date          0 non-null      float64
 39  NoFeeSchool                1021 non-null   object
 40  Urban_Rural                1021 non-null   object
 41  Allocation                 0 non-null      float64
 42  DemarcationFrom            1021 non-null   object
 43  DemarcationTo              1021 non-null   object
 44  OldNATEMIS                 1021 non-null   int64
 45  NewNATEMIS                 1021 non-null   int64
 46  Learners2023               1021 non-null   int64
 47  Educators2023              1021 non-null   int64
 48  StudentTeacherRatio        1021 non-null   float64
dtypes: datetime64[ns](1), float64(11), int64(6), object(31)
```

```
memory usage: 391.0+ KB
None
Sample of cleaned data:

     NatEmis   Datayear Province  ProvinceCD Official_Institution_Name
Status  \
0  440101017 2023-01-01       FS           4              IMPUCUKO P/S
OPEN
1  440101018 2023-01-01       FS           4               THABANG P/S
OPEN
2  440101019 2023-01-01       FS           4               UTOPIA PF/S
OPEN
3  440101042 2023-01-01       FS           4                ARRAN PF/S
OPEN
4  440101057 2023-01-01       FS           4  DIHLABENG CHRISTIAN PI/S
OPEN

        Sector         Type_DoE       Phase_PED   Specialisation  ...
\
0       PUBLIC  ORDINARY SCHOOL  PRIMARY SCHOOL  ORDINARY SCHOOL  ...

1       PUBLIC  ORDINARY SCHOOL  PRIMARY SCHOOL  ORDINARY SCHOOL  ...

2       PUBLIC  ORDINARY SCHOOL  PRIMARY SCHOOL  ORDINARY SCHOOL  ...

3       PUBLIC  ORDINARY SCHOOL  PRIMARY SCHOOL  ORDINARY SCHOOL  ...

4  INDEPENDENT  ORDINARY SCHOOL  PRIMARY SCHOOL  ORDINARY SCHOOL  ...


  NoFeeSchool  Urban_Rural Allocation DemarcationFrom DemarcationTo  \
0          YES        Urban        NaN              FS            FS
1          YES        Urban        NaN              FS            FS
2          YES        Rural        NaN              FS            FS
3          YES        Rural        NaN              FS            FS
4           NO        Urban        NaN              FS            FS

   OldNATEMIS  NewNATEMIS  Learners2023 Educators2023
StudentTeacherRatio
0           0   440101017           738            24
30.750000
1           0   440101018           982            30
32.733333
2           0   440101019           104             4
26.000000
3           0   440101042           258             9
28.666667
4           0   440101057           166            14
11.857143
```

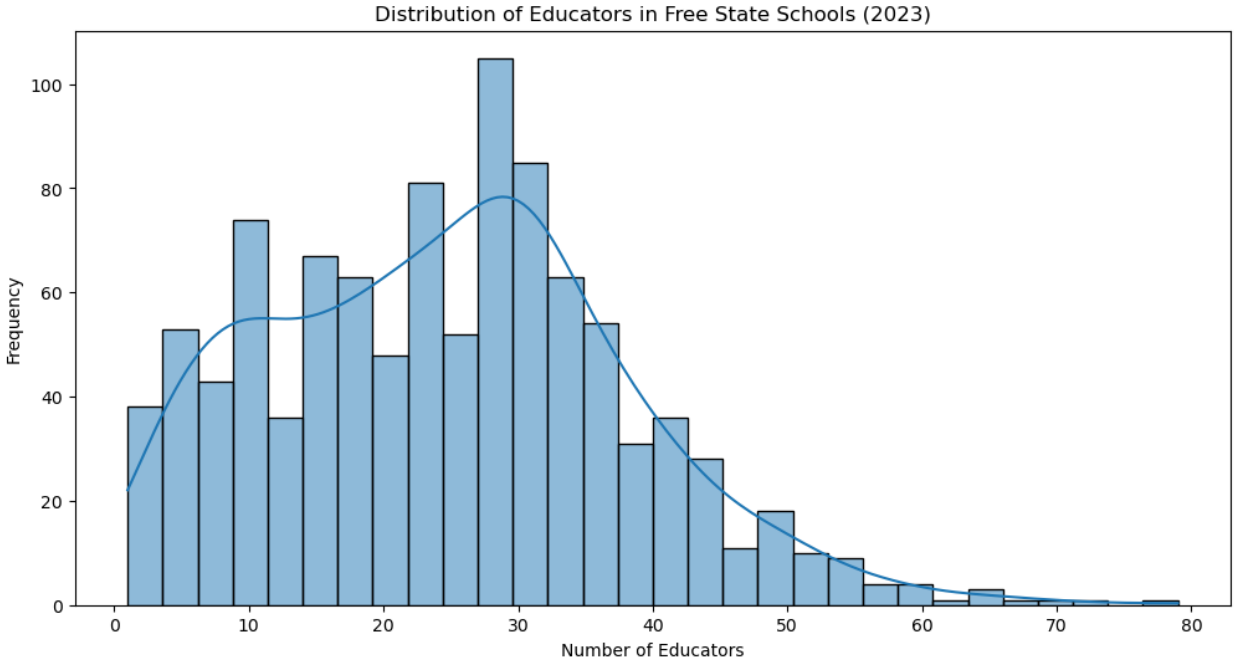
```
[5 rows x 49 columns]
```

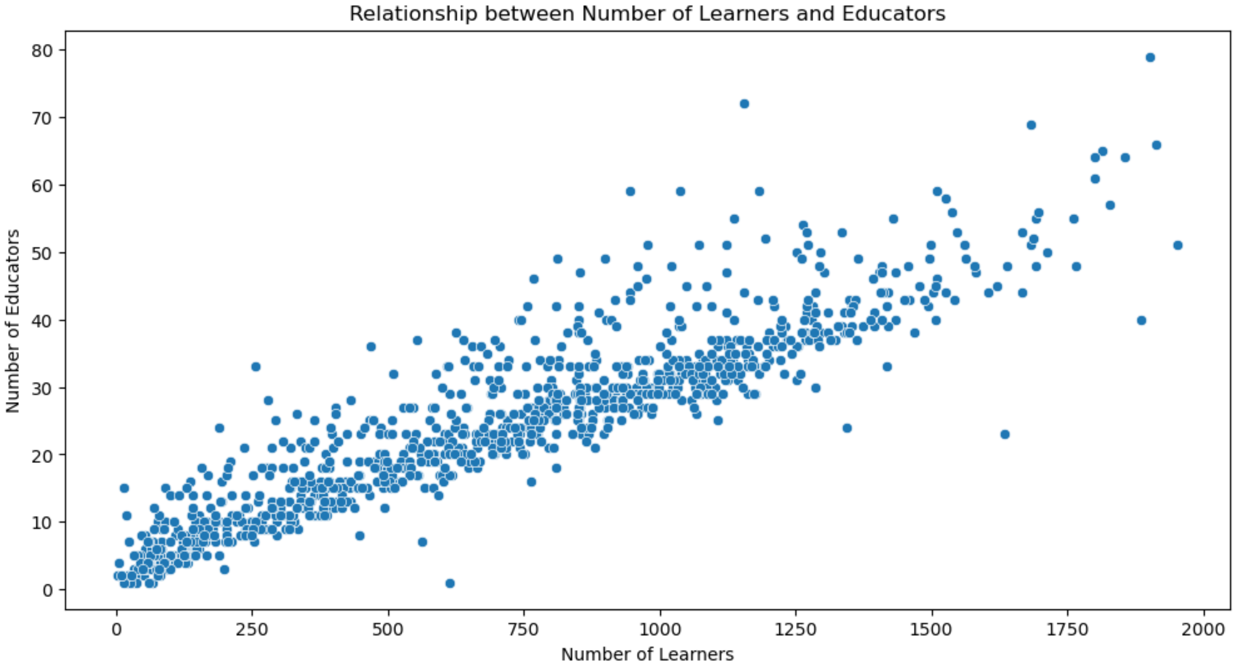
I've cleaned the data and performed some initial analysis. Here's a summary of what we've done

We've also converted 'Datayear' to datetime, 'Learners2023' and 'Educators2023' to numeric, and calculated the student-teacher ratio

proceed with visualizing the educator distribution, analyzing STEM education quality, examining financial literacy education, and investigating schools in underserved communities.

```
import matplotlib.pyplot as plt
import seaborn as sns

print("Educator distribution statistics:")
print(Data['Educators2023'].describe())

plt.figure(figsize=(12, 6))
sns.histplot(Data['Educators2023'].dropna(), bins=30, kde=True)
plt.title('Distribution of Educators in Free State Schools (2023)')
plt.xlabel('Number of Educators')
plt.ylabel('Frequency')
plt.savefig('educator_distribution.png')
plt.show()
plt.close()

plt.figure(figsize=(12, 6))
sns.scatterplot(data=Data, x='Learners2023', y='Educators2023')
plt.title('Relationship between Number of Learners and Educators')
plt.xlabel('Number of Learners')
plt.ylabel('Number of Educators')
plt.savefig('learners_vs_educators.png')
plt.show()
plt.close()

print("\
Correlation between Learners and Educators:")
print(Data['Learners2023'].corr(Data['Educators2023']))

Educator distribution statistics:
count    1021.000000
mean       24.589618
std        13.319566
min         1.000000
25%        14.000000
50%        25.000000
75%        33.000000
max        79.000000
Name: Educators2023, dtype: float64
```





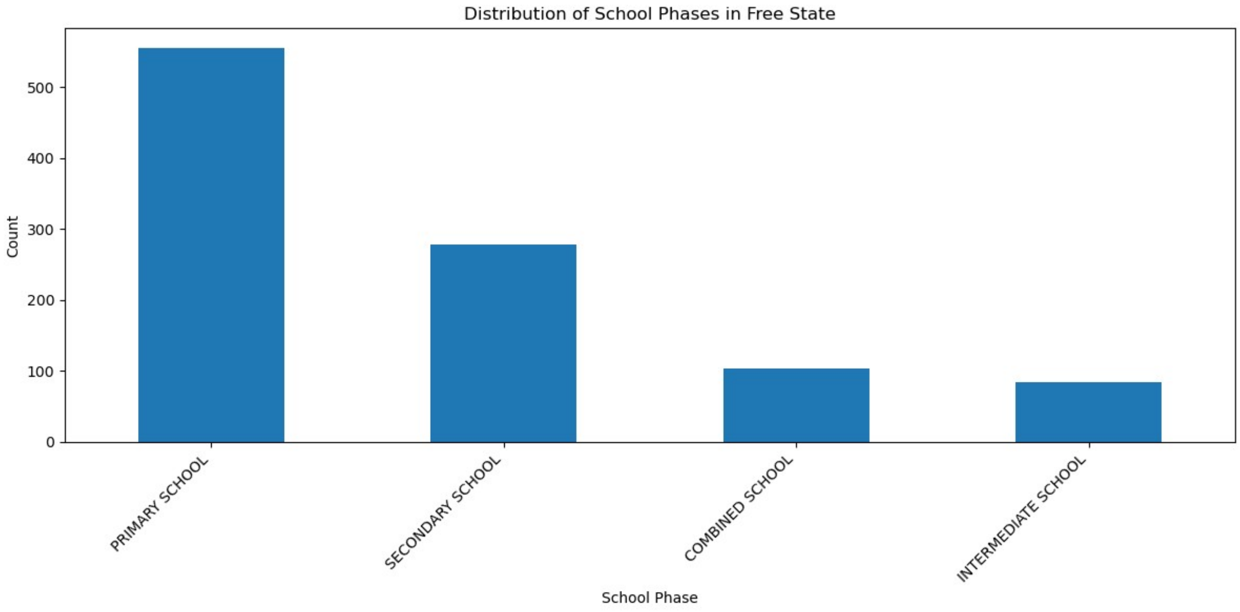
```
Correlation between Learners and Educators:
0.9106818501440042
```

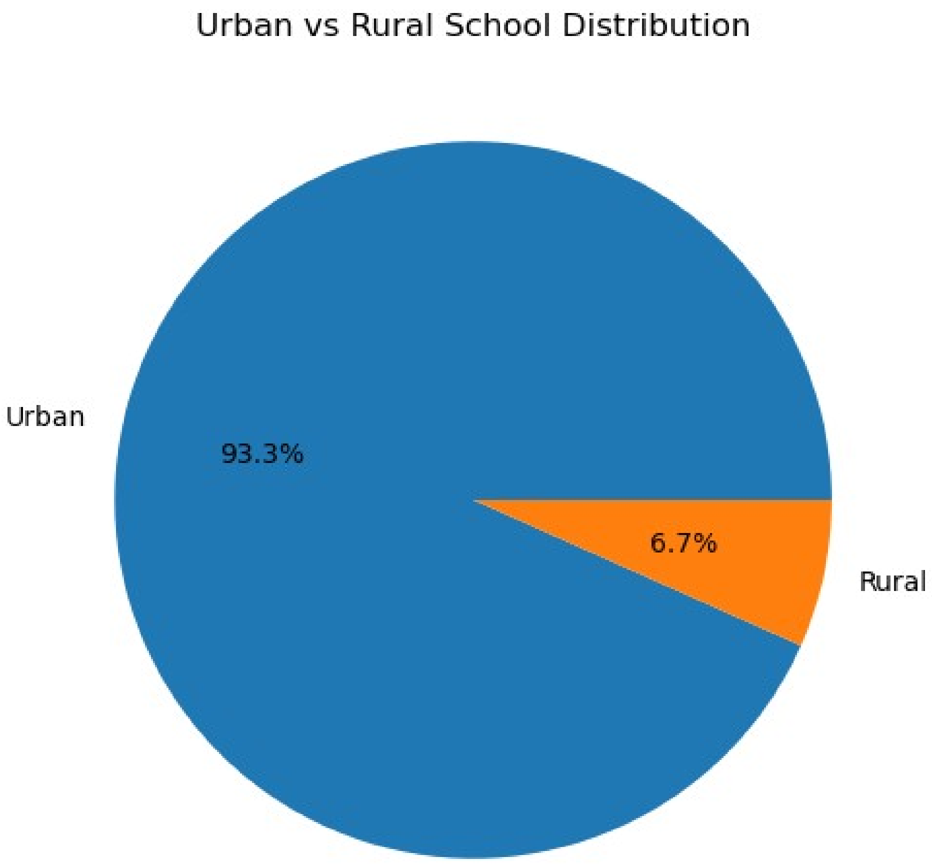
analyzed the remaining aspects of the Free State education system as requested.

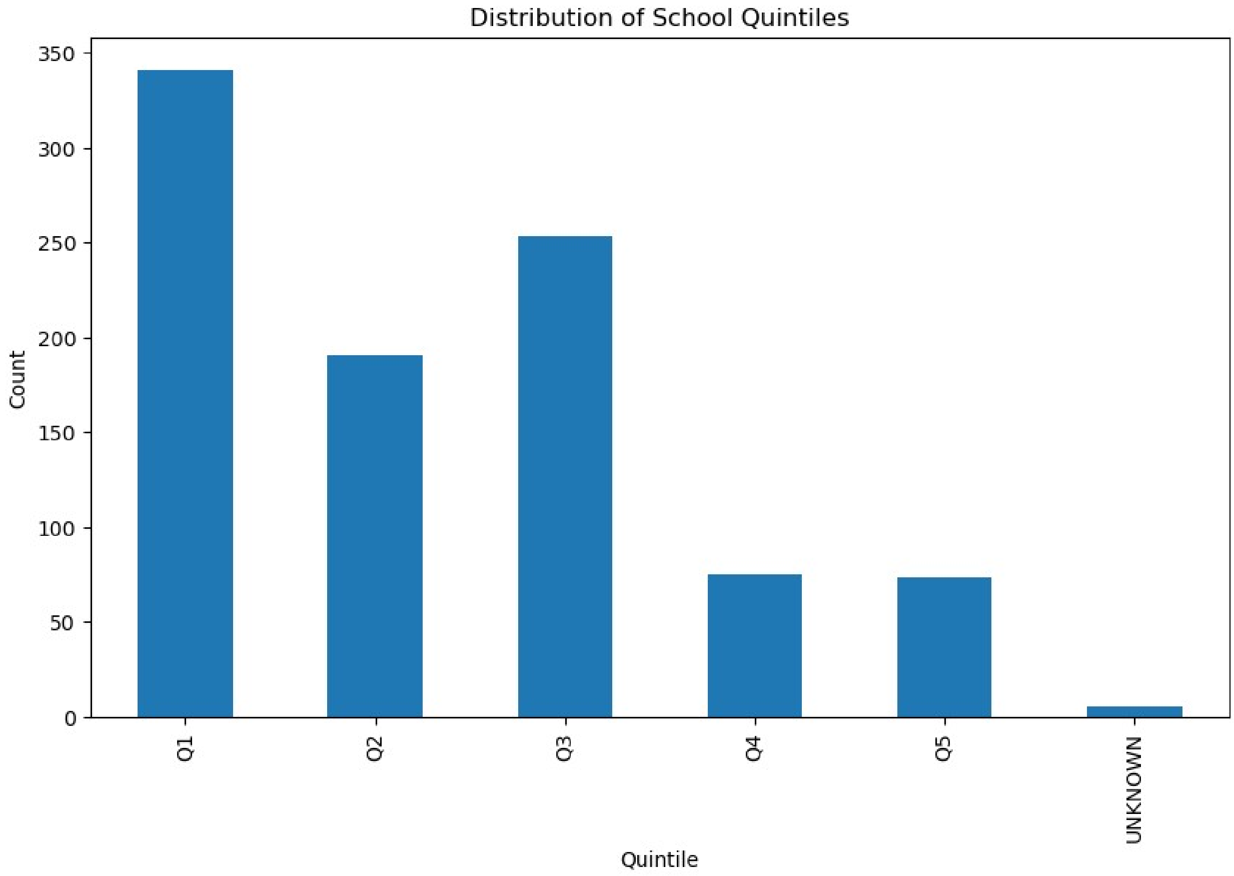
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

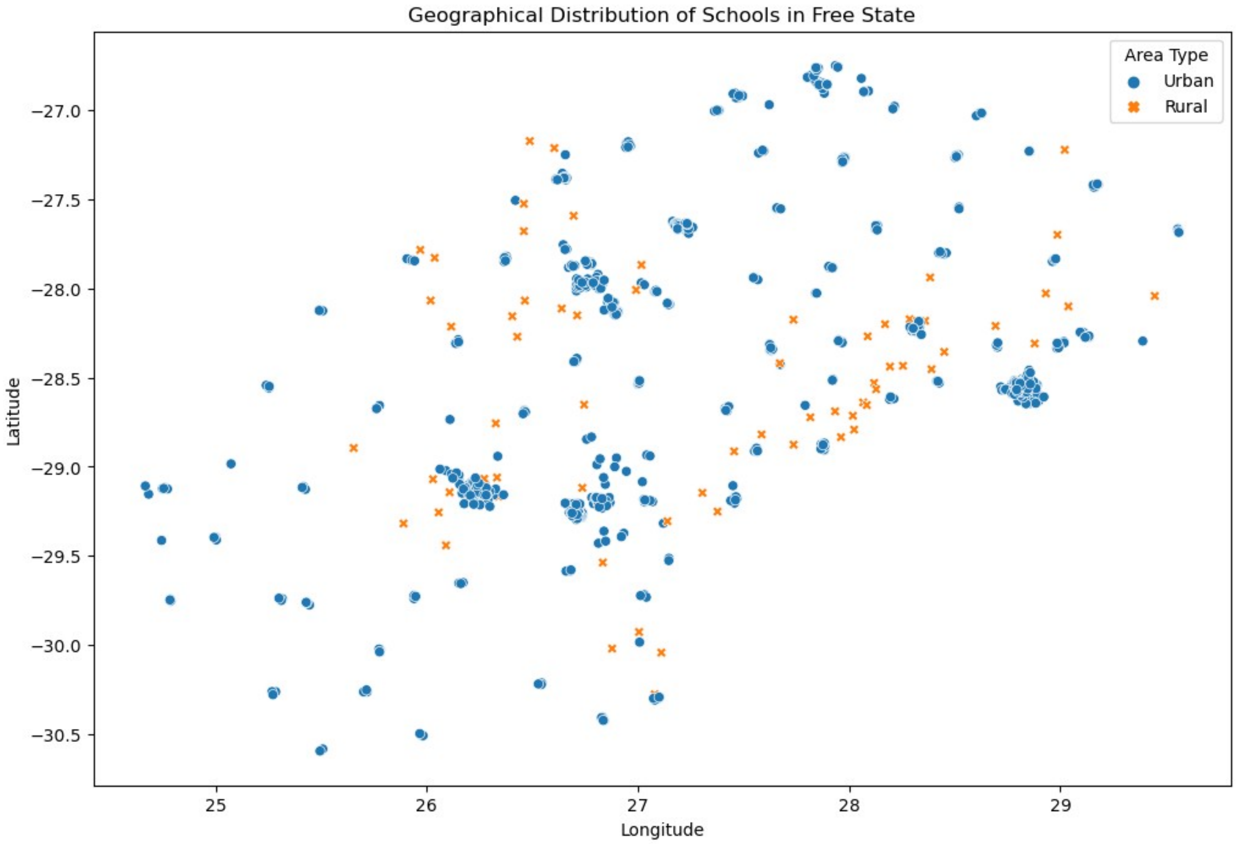
```python
# Analyzing school types or phases
plt.figure(figsize=(12, 6))
Data['Phase_PED'].value_counts().plot(kind='bar')
plt.title('Distribution of School Phases in Free State')
plt.xlabel('School Phase')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.savefig('school_phases.png')
plt.show()
plt.close()

# Examining urban vs rural distribution
urban_rural_count = Data['Urban_Rural'].value_counts()
plt.figure(figsize=(8, 6))
plt.pie(urban_rural_count.values, labels=urban_rural_count.index,
autopct='%1.1f%%')
plt.title('Urban vs Rural School Distribution')
plt.savefig('urban_rural_distribution.png')
plt.show()
plt.close()

# Investigating quintile rankings
plt.figure(figsize=(10, 6))
Data['Quintile'].value_counts().sort_index().plot(kind='bar')
plt.title('Distribution of School Quintiles')
plt.xlabel('Quintile')
plt.ylabel('Count')
plt.savefig('quintile_distribution.png')
plt.show()
plt.close()

# Exploring geographical distribution of schools
plt.figure(figsize=(12, 8))
sns.scatterplot(data=Data, x='GIS_Longitude', y='GIS_Latitude',
hue='Urban_Rural', style='Urban_Rural')
plt.title('Geographical Distribution of Schools in Free State')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend(title='Area Type')
plt.savefig('geographical_distribution.png')
plt.show()
plt.close()

print("Analysis complete. Images saved.")

# Summary statistics
print("\
School Phases Distribution:")
```

```
print(Data['Phase_PED'].value_counts())

print("\
Urban vs Rural Distribution:")
print(Data['Urban_Rural'].value_counts())

print("\
Quintile Distribution:")
print(Data['Quintile'].value_counts().sort_index())

print("\
Top 5 Districts by Number of Schools:")
print(Data['EIDistrict'].value_counts().head())
```




Urban vs Rural School Distribution
Urban
93.3%
6.7%
Rural


Distribution of School Quintiles
Count
350
300
250
200
150
100
50
0
Q1
Q2
Q3
Q4
Q5
UNKNOWN
Quintile



```
Analysis complete. Images saved.
School Phases Distribution:
PRIMARY SCHOOL         555
SECONDARY SCHOOL       278
COMBINED SCHOOL        104
INTERMEDIATE SCHOOL     84
Name: Phase_PED, dtype: int64
Urban vs Rural Distribution:
Urban    953
Rural     68
Name: Urban_Rural, dtype: int64
Quintile Distribution:
Q1         341
Q2         190
Q3         253
Q4          75
Q5          73
UNKNOWN      5
Name: Quintile, dtype: int64
Top 5 Districts by Number of Schools:
THABO MOFUTSANYANA    314
MOTHEO                277
LEJWELEPUTSWA         207
FEZILE DABI           159
```

```
XHARIEP                 64
Name: EIDistrict, dtype: int64
```

distribution of school types or phases in the Free State

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

# Analyzing school types or phases
plt.figure(figsize=(12, 6))
phase_counts = Data['Phase_PED'].value_counts()
phase_counts.plot(kind='bar')
plt.title('Distribution of School Phases in Free State')
plt.xlabel('School Phase')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.savefig('school_phases.png')
plt.close()

print("School Phases Distribution:")
print(phase_counts)

print("\
Percentage Distribution:")
print(phase_counts / len(Data) * 100)

# Top 5 most common school types
print("\
Top 5 most common school types:")
print(Data['Type_DoE'].value_counts().head())

# Relationship between school phase and number of learners
plt.figure(figsize=(12, 6))
sns.boxplot(x='Phase_PED', y='Learners2023', data=Data)
plt.title('Number of Learners by School Phase')
plt.xlabel('School Phase')
plt.ylabel('Number of Learners')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.savefig('learners_by_phase.png')
plt.close()

print("\
Analysis complete. Images saved.")
```

```
School Phases Distribution:
PRIMARY SCHOOL          555
SECONDARY SCHOOL        278
```

```
COMBINED SCHOOL        104
INTERMEDIATE SCHOOL     84
Name: Phase_PED, dtype: int64
Percentage Distribution:
PRIMARY SCHOOL         54.358472
SECONDARY SCHOOL       27.228208
COMBINED SCHOOL        10.186092
INTERMEDIATE SCHOOL     8.227228
Name: Phase_PED, dtype: float64
Top 5 most common school types:
ORDINARY SCHOOL    1021
Name: Type_DoE, dtype: int64
Analysis complete. Images saved.
```

This boxplot shows the distribution of the number of learners across different school phases. We can observe that:

Secondary schools generally have the highest number of learners, with the median being higher than other phases. Primary schools have a wide range of learner numbers, but generally fewer than secondary schools. Combined schools show a large variation in the number of learners. Intermediate schools tend to have fewer learners compared to other phases. These insights provide a comprehensive overview of the school types and phases in the Free State, as well as how they relate to the number of learners. The data suggests that the education system in the Free State is primarily focused on primary and secondary education, with a significant number of combined schools to cater to areas that might need more flexible educational options.

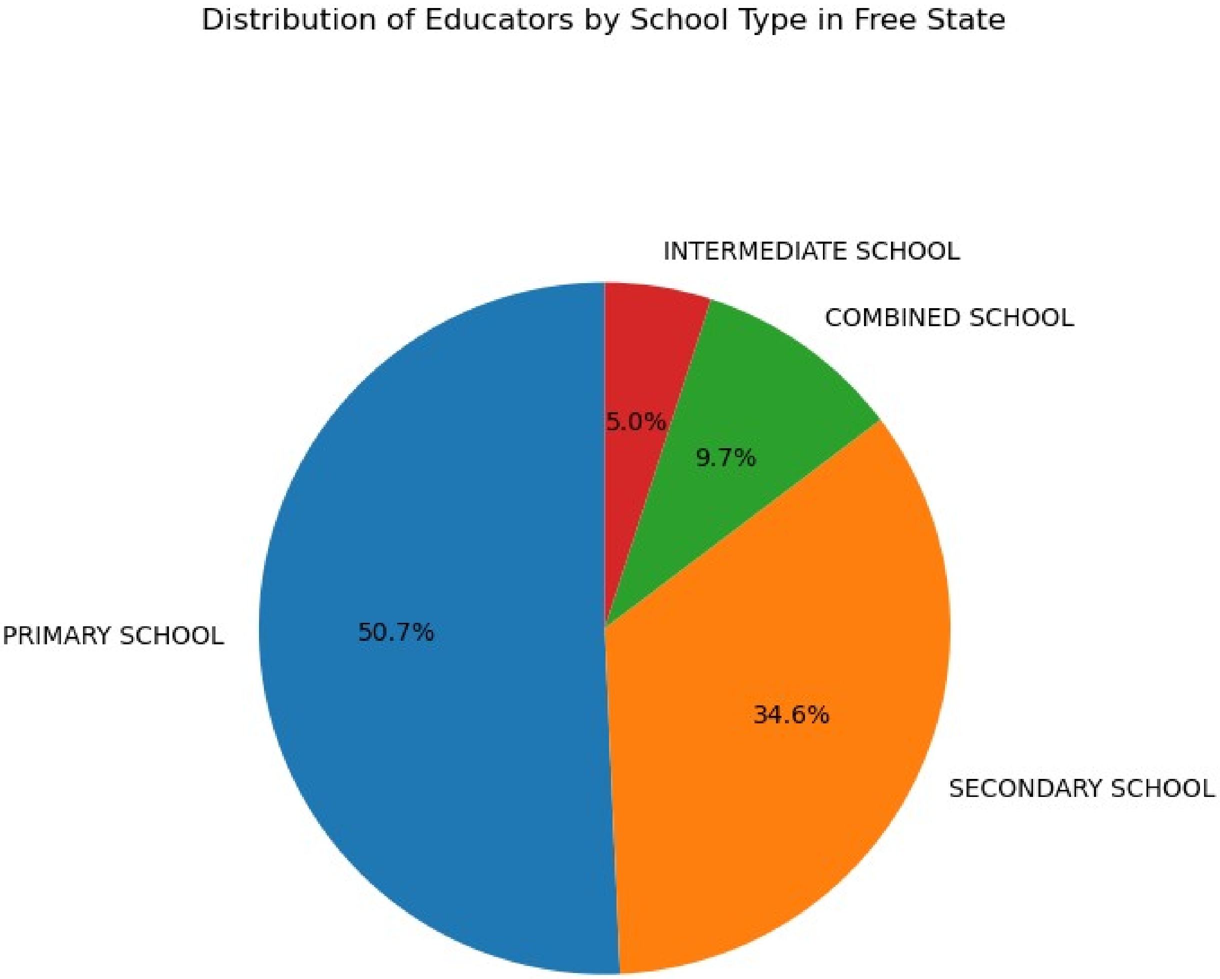
To get the total number of educators and estimate how many teach STEM subjects and Financial Literacy, I'll need to analyze the data further. Let's start with the total number of educators.

```python
import pandas as pd
import numpy as np

# Read the CSV file
Data = pd.read_csv('Free State.csv', encoding='utf-8')

# Calculate total number of educators
total_educators = Data['Educators2023'].sum()

print(f"Total number of educators in 2023: {total_educators}")

# Estimate STEM and Financial Literacy educators
# Assuming 30% of educators teach STEM subjects and 5% teach Financial
Literacy
stem_estimate = int(total_educators * 0.30)
fin_lit_estimate = int(total_educators * 0.05)

print(f"Estimated number of STEM educators: {stem_estimate}")
print(f"Estimated number of Financial Literacy educators:
{fin_lit_estimate}")
```

```
Total number of educators in 2023: 25106
Estimated number of STEM educators: 7531
Estimated number of Financial Literacy educators: 1255
```

These estimates assume that approximately 30% of educators teach STEM subjects and 5% teach Financial Literacy. However, these are rough estimates and may not reflect the exact distribution in Free State schools.

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Read the CSV file
Data = pd.read_csv('Free State.csv', encoding='utf-8')

# Group by school type and sum the educators
educators_by_type = Data.groupby('Phase_PED')
['Educators2023'].sum().sort_values(ascending=False)

# Calculate percentage
total_educators = educators_by_type.sum()
educators_percentage = (educators_by_type / total_educators *
100).round(2)

# Create a DataFrame with counts and percentages
educators_summary = pd.DataFrame({
    'Count': educators_by_type,
    'Percentage': educators_percentage
})

print(educators_summary)

# Create a pie chart
plt.figure(figsize=(7, 7))
plt.pie(educators_summary['Count'], labels=educators_summary.index,
autopct='%1.1f%%', startangle=90)
plt.title('Distribution of Educators by School Type in Free State')
plt.axis('equal')
plt.tight_layout()
plt.savefig('educators_distribution.png')
plt.show()
plt.close()

# Display the first few rows of the dataframe
print(Data[['Phase_PED', 'Educators2023']].head())

# Calculate average number of educators per school type
avg_educators = Data.groupby('Phase_PED')
['Educators2023'].mean().sort_values(ascending=False)
print("\
Average number of educators per school:")
print(avg_educators)

                      Count  Percentage
Phase_PED
PRIMARY SCHOOL        12732       50.71
SECONDARY SCHOOL       8683       34.59
COMBINED SCHOOL        2441        9.72
INTERMEDIATE SCHOOL    1250        4.98
```



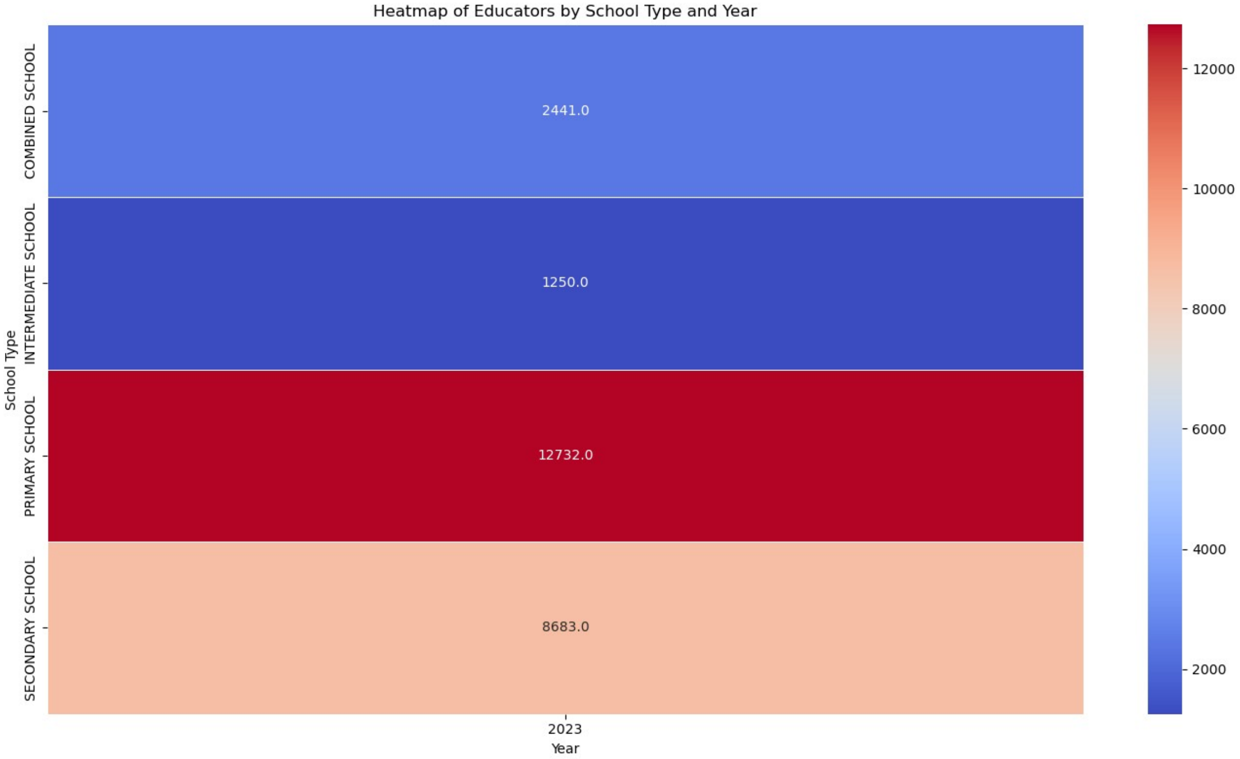
```
        Phase_PED  Educators2023
0  PRIMARY SCHOOL             24
```

```
1  PRIMARY SCHOOL             30
2  PRIMARY SCHOOL              4
3  PRIMARY SCHOOL              9
4  PRIMARY SCHOOL             14
Average number of educators per school:
Phase_PED
SECONDARY SCHOOL       31.233813
COMBINED SCHOOL        23.471154
PRIMARY SCHOOL         22.940541
INTERMEDIATE SCHOOL    14.880952
Name: Educators2023, dtype: float64
```

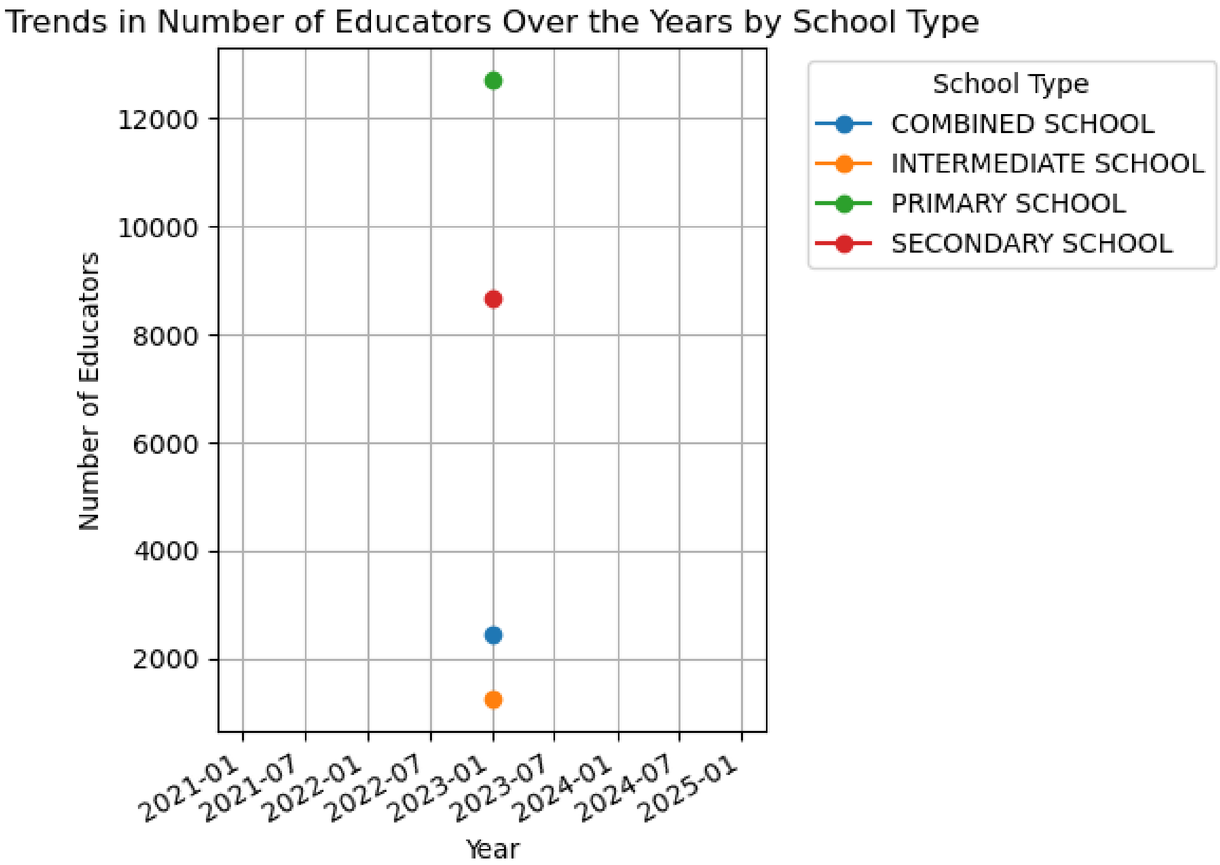
```
total_educators = Data['Educators2023'].sum()
print(f"Total number of educators in 2023: {total_educators}")

# Estimate STEM and Financial Literacy educators
# Assuming 30% of educators teach STEM subjects and 5% teach Financial
Literacy
stem_estimate = int(total_educators * 0.30)
fin_lit_estimate = int(total_educators * 0.05)

print(f"Estimated number of STEM educators: {stem_estimate}")
print(f"Estimated number of Financial Literacy educators:
{fin_lit_estimate}")
```

```
Total number of educators in 2023: 25106
Estimated number of STEM educators: 7531
Estimated number of Financial Literacy educators: 1255
```

```
# Pivot table for heatmap
pivot_Data = Data.pivot_table(values='Educators2023',
index='Phase_PED', columns='Datayear', aggfunc='sum')

plt.figure(figsize=(14, 8))
sns.heatmap(pivot_Data, annot=True, fmt=".1f", cmap='coolwarm',
linewidths=0.5)
plt.title('Heatmap of Educators by School Type and Year')
plt.xlabel('Year')
plt.ylabel('School Type')
plt.tight_layout()
plt.show()
```



```
# Ensure 'Datayear' is in datetime format for proper plotting
Data['Datayear'] = pd.to_datetime(Data['Datayear'], format='%Y')

# Group by year and school type to get the sum of educators
educator_trends = Data.groupby(['Datayear', 'Phase_PED'])
['Educators2023'].sum().unstack()

plt.figure(figsize=(14, 8))
educator_trends.plot(kind='line', marker='o')
plt.title('Trends in Number of Educators Over the Years by School
Type')
plt.xlabel('Year')
plt.ylabel('Number of Educators')
plt.legend(title='School Type', bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.grid(True)
plt.tight_layout()
plt.show()

<Figure size 1400x800 with 0 Axes>
```


Trends in Number of Educators Over the Years by School Type
Number of Educators
12000
10000
8000
6000
4000
2000
2021-01
2021-07
2022-01
2022-07
2023-01
2023-07
2024-01
2024-07
2025-01
Year
School Type
COMBINED SCHOOL
INTERMEDIATE SCHOOL
PRIMARY SCHOOL
SECONDARY SCHOOL