In [2]:
```
import pandas as pd
```

In [3]:
```
data=pd.read_csv("E:\Public_School_Characteristics_2020-21.csv")
data.head(10)
```

```
C:\Users\SAHIL SINGH\AppData\Local\Temp\ipykernel_29428\2609331481.py:1: DtypeWarn
ing: Columns (11) have mixed types. Specify dtype option on import or set low_memo
ry=False.
  data=pd.read_csv("E:\Public_School_Characteristics_2020-21.csv")
```

Out[3]:

	X	Y	OBJECTID	NCESSCH	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NA
0	-86.206200	34.260200	1	10000500870	2020-2021	AL	100005	AL-101	Albert
1	-86.204900	34.262200	2	10000500871	2020-2021	AL	100005	AL-101	Albert
2	-86.220100	34.273300	3	10000500879	2020-2021	AL	100005	AL-101	Albert
3	-86.221806	34.252700	4	10000500889	2020-2021	AL	100005	AL-101	Albert
4	-86.193300	34.289800	5	10000501616	2020-2021	AL	100005	AL-101	Albert
5	-86.221800	34.253300	6	10000502150	2020-2021	AL	100005	AL-101	Albert
6	-86.254100	34.533700	7	10000600193	2020-2021	AL	100006	AL-048	Mars Cou
7	-86.142000	34.362500	8	10000600872	2020-2021	AL	100006	AL-048	Mars Cou
8	-86.270400	34.406900	9	10000600876	2020-2021	AL	100006	AL-048	Mars Cou
9	-86.321259	34.176234	10	10000600877	2020-2021	AL	100006	AL-048	Mars Cou

10 rows × 79 columns

Checking for Duplicate

In [4]:
```
data[data.duplicated()]
```

Out[4]:

X	Y	OBJECTID	NCESSCH	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NAME	SCH_NAME	...

0 rows × 79 columns

In [5]:
```
data.duplicated().sum()
```

Out[5]:
```
0
```

No duplicates found.

Cleaning the Data

In [6]:
```
data.isnull().values.any()
```

Out[6]:
```
True
```

In [7]:
```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
data.isnull().sum()
```

Out[7]:

X	0
Y	0
OBJECTID	0
NCESSCH	0
SURVYEAR	0
STABR	0
LEAID	0
ST_LEAID	0
LEA_NAME	0
SCH_NAME	0
LSTREET1	3
LSTREET2	100144
LCITY	0
LSTATE	0
LZIP	0
LZIP4	0
PHONE	0
CHARTER_TEXT	0
MAGNET_TEXT	0
VIRTUAL	0
GSLO	0
GSHI	0
SCHOOL_LEVEL	0
TITLEI	0
STITLEI	0
STATUS	0
SCHOOL_TYPE_TEXT	0
SY_STATUS_TEXT	0
ULOCALE	0
NMCNTY	0
TOTFRL	23758
FRELCH	26605
REDLCH	26605
PK	69021
KG	47185
G01	46781
G02	46742
G03	46790
G04	47000
G05	48296
G06	63502
G07	68572
G08	68344
G09	73736
G10	73875
G11	73903
G12	73991
G13	100579
UG	92569
AE	100540
TOTMENROL	100722
TOTFENROL	100722
TOTAL	2071
MEMBER	2071
FTE	9502
STUTERATIO	1216
AMALM	31433
AMALF	31687
AM	24373
ASALM	17784
ASALF	18097
AS	13828
BLALM	11599
BLALF	12288

```
BL                    8706
HPALM                40038
HPALF                40567
HP                   34906
HIALM                 4686
HIALF                 4907
HI                    3698
TRALM                 8943
TRALF                 9128
TR                    6700
WHALM                 4354
WHALF                 4607
WH                    3784
LATCOD                   0
LONCOD                   0
dtype: int64
```

The above table shows us how many null values are there in each attribute.

Removing Null Values:

However the total number of rows in the tab 100722. We will remove the attributes having 90% of data as 'null'. Droping Lunch data as well as it is not usefull in this analysis.

In [8]:
```
attributes_to_remove = ['LSTREET2', 'TOTMENROL', 'TOTFENROL','TOTFRL','FRELCH','REI
df = data.drop(attributes_to_remove, axis=1)
```

In [9]:
```
df.head(5)
```

Out[9]:

	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NAME	SCH_NAME	LSTREET1	LCITY	LSTATE
0	2020-2021	AL	100005	AL-101	Albertville City	Albertville Middle School	600 E Alabama Ave	Albertville	AL
1	2020-2021	AL	100005	AL-101	Albertville City	Albertville High School	402 E McCord Ave	Albertville	AL
2	2020-2021	AL	100005	AL-101	Albertville City	Albertville Intermediate School	901 W McKinney Ave	Albertville	AL
3	2020-2021	AL	100005	AL-101	Albertville City	Albertville Elementary School	145 West End Drive	Albertville	AL
4	2020-2021	AL	100005	AL-101	Albertville City	Albertville Kindergarten and PreK	257 Country Club Rd	Albertville	AL

Grade values can be zero. therefore, replacing NaN with 0. Moreover, there can be zero number of students from different race in some schools. so it can be xero as well.

In [10]:
```
attributes_to_replace = ['KG','G01','G02','G03','G04','G05','G06','G07','G08','G09
df[attributes_to_replace] = df[attributes_to_replace].fillna(0)
```

In [11]:
```
df.head(3)
```

Out[11]:

	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NAME	SCH_NAME	LSTREET1	LCITY	LSTATE
0	2020-2021	AL	100005	AL-101	Albertville City	Albertville Middle School	600 E Alabama Ave	Albertville	AL
1	2020-2021	AL	100005	AL-101	Albertville City	Albertville High School	402 E McCord Ave	Albertville	AL
2	2020-2021	AL	100005	AL-101	Albertville City	Albertville Intermediate School	901 W McKinney Ave	Albertville	AL

The total number of students and members cannot be zero. So, removing rows that have NAN in their total and Member Attribute.

In [12]:
```
df = df.dropna(subset=['TOTAL','MEMBER','FTE','STUTERATIO'])
```

In [13]:
```
df.isnull().sum()
```

Out[13]:

SURVYEAR	0
STABR	0
LEAID	0
ST_LEAID	0
LEA_NAME	0
SCH_NAME	0
LSTREET1	3
LCITY	0
LSTATE	0
LZIP	0
LZIP4	0
CHARTER_TEXT	0
MAGNET_TEXT	0
VIRTUAL	0
GSLO	0
GSHI	0
SCHOOL_LEVEL	0
TITLEI	0
STITLEI	0
STATUS	0
SCHOOL_TYPE_TEXT	0
SY_STATUS_TEXT	0
ULOCALE	0
NMCNTY	0
PK	0
KG	0
G01	0
G02	0
G03	0
G04	0
G05	0
G06	0
G07	0
G08	0
G09	0
G10	0
G11	0
G12	0
G13	0
UG	0
AE	0
TOTAL	0
MEMBER	0
FTE	0
STUTERATIO	0
AMALM	0
AMALF	0
AM	0
ASALM	0
ASALF	0
AS	0
BLALM	0
BLALF	0
BL	0
HPALM	0
HPALF	0
HP	0
HIALM	0
HIALF	0
HI	0
TRALM	0
TRALF	0
TR	0
WHALM	0

```
WHALF                0
WH                   0
LATCOD               0
LONCOD               0
dtype: int64
```

All NaN are removed.

Discriptive Visualization

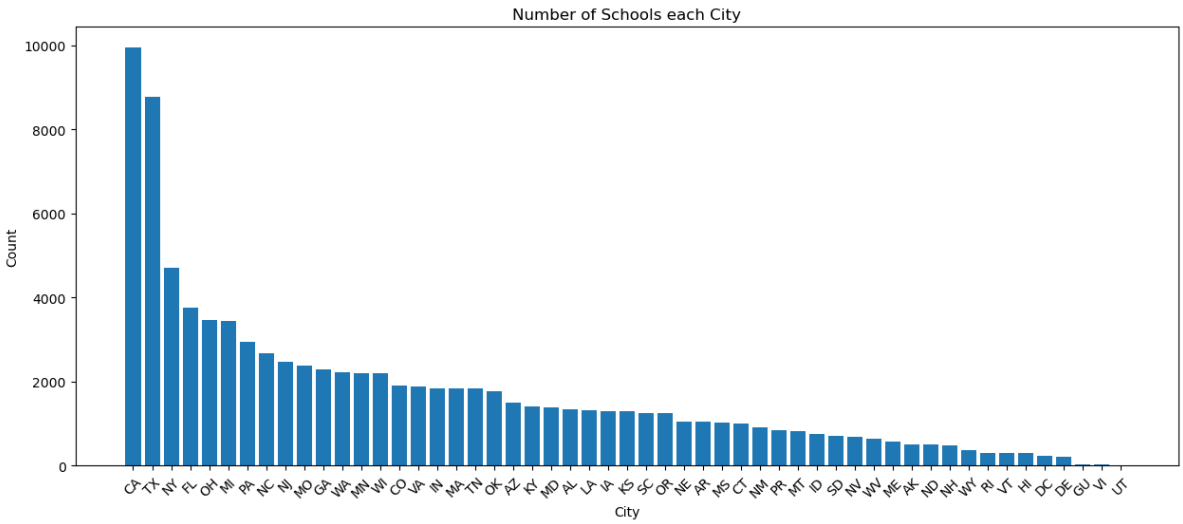
In [14]:
```
import matplotlib.pyplot as plt
```

In [15]:
```
df.head(4)
```

Out[15]:

	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NAME	SCH_NAME	LSTREET1	LCITY	LSTATE
0	2020-2021	AL	100005	AL-101	Albertville City	Albertville Middle School	600 E Alabama Ave	Albertville	AL
1	2020-2021	AL	100005	AL-101	Albertville City	Albertville High School	402 E McCord Ave	Albertville	AL
2	2020-2021	AL	100005	AL-101	Albertville City	Albertville Intermediate School	901 W McKinney Ave	Albertville	AL
3	2020-2021	AL	100005	AL-101	Albertville City	Albertville Elementary School	145 West End Drive	Albertville	AL

In [16]:
```
charter_counts = df['LSTATE'].value_counts()

labels = charter_counts.index
values = charter_counts.values

plt.figure(figsize=(15, 6))

plt.bar(labels, values)

plt.title('Number of Schools each City')
plt.xlabel('City')
plt.ylabel('Count')
plt.xticks(rotation=45)

plt.show()
```



In [17]:
```
filtered_df = df[df['CHARTER_TEXT'] == 'N']
filtered_df.head(10)
```

Out[17]:

	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NAME	SCH_NAME	LSTREET1	LCITY	LSTA
52271	2020-2021	MT	3000002	MT-0327	Somers Elem	Lakeside Elementary School	255 Adams Street	Lakeside	
52272	2020-2021	MT	3000002	MT-0327	Somers Elem	Somers Middle School	315 School Addition Rd	Somers	
52273	2020-2021	MT	3000003	MT-0402	Cut Bank Elem	Glacier Elementary School	451 Tipville Road	Cut Bank	
52274	2020-2021	MT	3000003	MT-0402	Cut Bank Elem	Anna Jeffries Elementary	105 2nd Street NW	Cut Bank	
52275	2020-2021	MT	3000003	MT-0402	Cut Bank Elem	Cut Bank Middle School	101 3rd Avenue SE	Cut Bank	
52276	2020-2021	MT	3000003	MT-0402	Cut Bank Elem	Hidden Lake Elementary	100 Welch Road	Cut Bank	
52277	2020-2021	MT	3000003	MT-0402	Cut Bank Elem	Zenith Elementary	Zenith Colony Road	Cut Bank	
52278	2020-2021	MT	3000003	MT-0402	Cut Bank Elem	H C Davis Elementary	15 2nd Avenue SE	Cut Bank	
52279	2020-2021	MT	3000003	MT-0402	Cut Bank Elem	Horizon Elementary	100 Horizon Road	Cut Bank	
52280	2020-2021	MT	3000004	MT-1222	Mountain View Elem	Mountain View Elementary	Pardue Road-Seville	Cut Bank	

In [18]:
```
df.head(4)
```

Out[18]:

	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NAME	SCH_NAME	LSTREET1	LCITY	LSTATE
0	2020-2021	AL	100005	AL-101	Albertville City	Albertville Middle School	600 E Alabama Ave	Albertville	AL
1	2020-2021	AL	100005	AL-101	Albertville City	Albertville High School	402 E McCord Ave	Albertville	AL
2	2020-2021	AL	100005	AL-101	Albertville City	Albertville Intermediate School	901 W McKinney Ave	Albertville	AL
3	2020-2021	AL	100005	AL-101	Albertville City	Albertville Elementary School	145 West End Drive	Albertville	AL

In [19]:
```
df['CHARTER_TEXT'].unique()
```

Out[19]:
```
array(['No', 'Yes', 'N'], dtype=object)
```

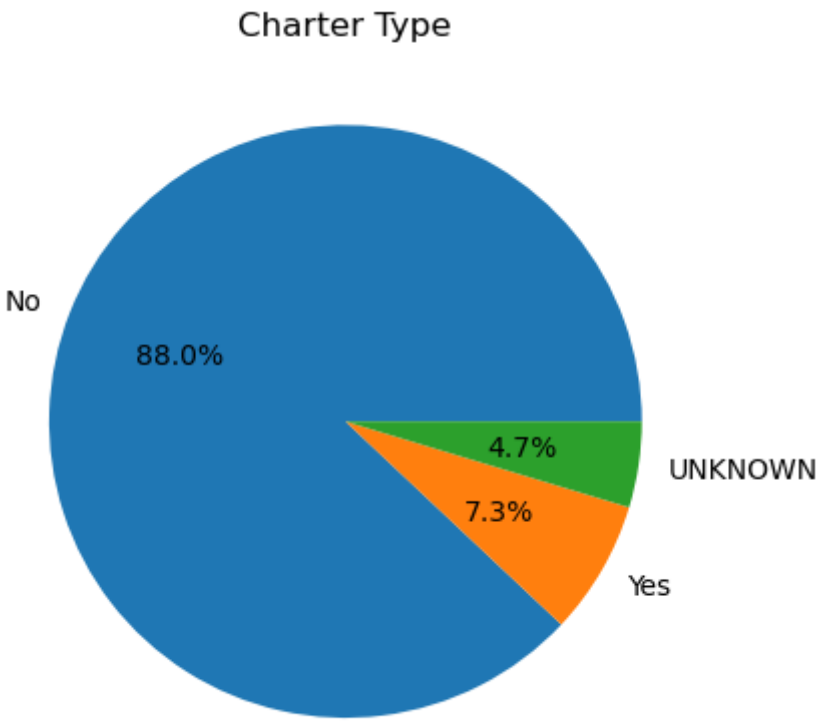
In [20]:
```
df['CHARTER_TEXT'] = df['CHARTER_TEXT'].replace('N', 'UNKNOWN')
```

In [21]:
```
df.loc[df['CHARTER_TEXT'] == 'N',]
```

Out[21]:

	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NAME	SCH_NAME	LSTREET1	LCITY	LSTATE	LZIP

In [22]:
```
charter_counts = df['CHARTER_TEXT'].value_counts()
labels = charter_counts.index
sizes = charter_counts.values

plt.pie(sizes, labels=labels, autopct='%1.1f%%')

plt.title('Charter Type')

plt.show()
```



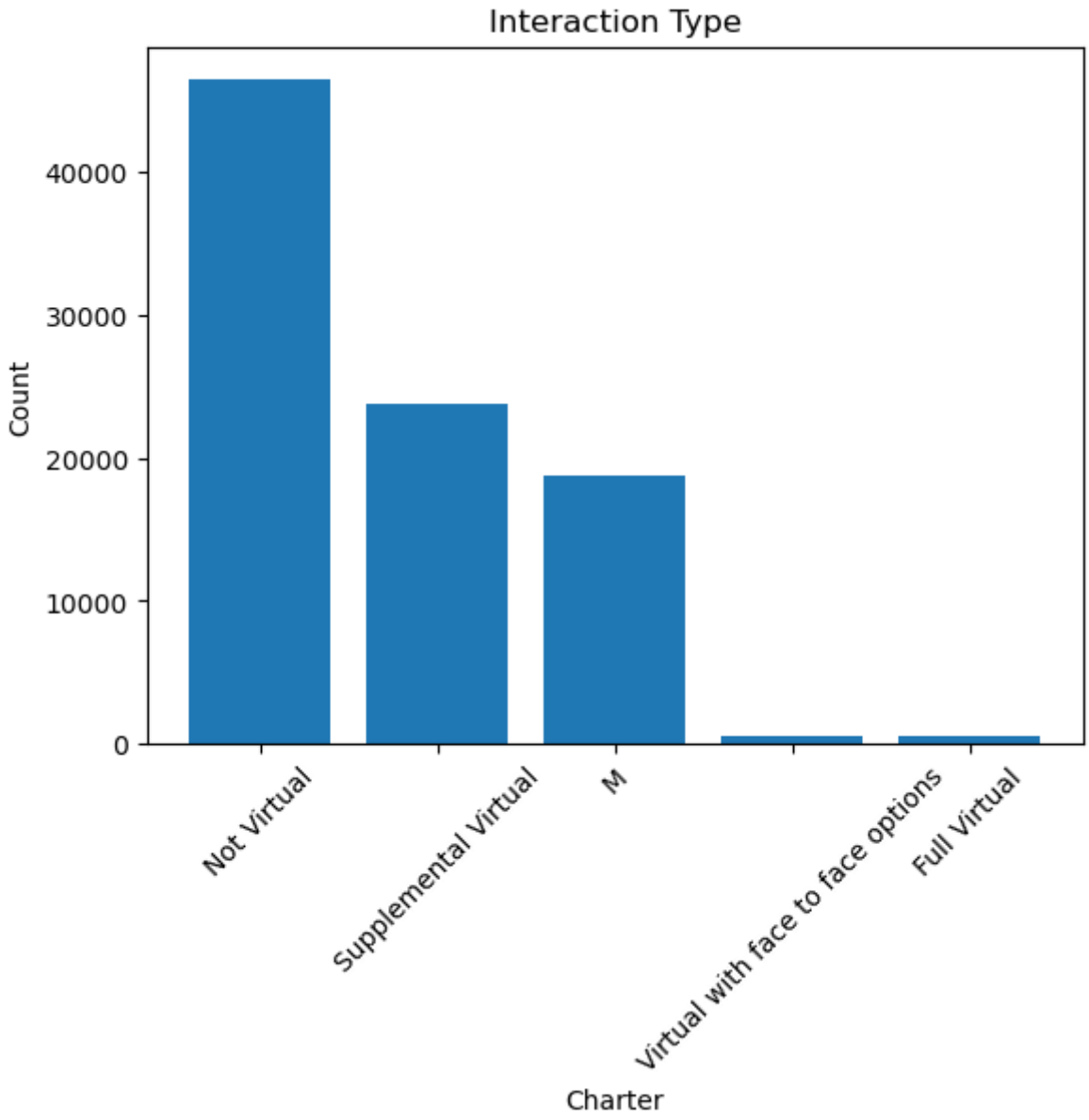
In [23]:
```
df['CHARTER_TEXT'].unique()
```

Out[23]:
```
array(['No', 'Yes', 'UNKNOWN'], dtype=object)
```

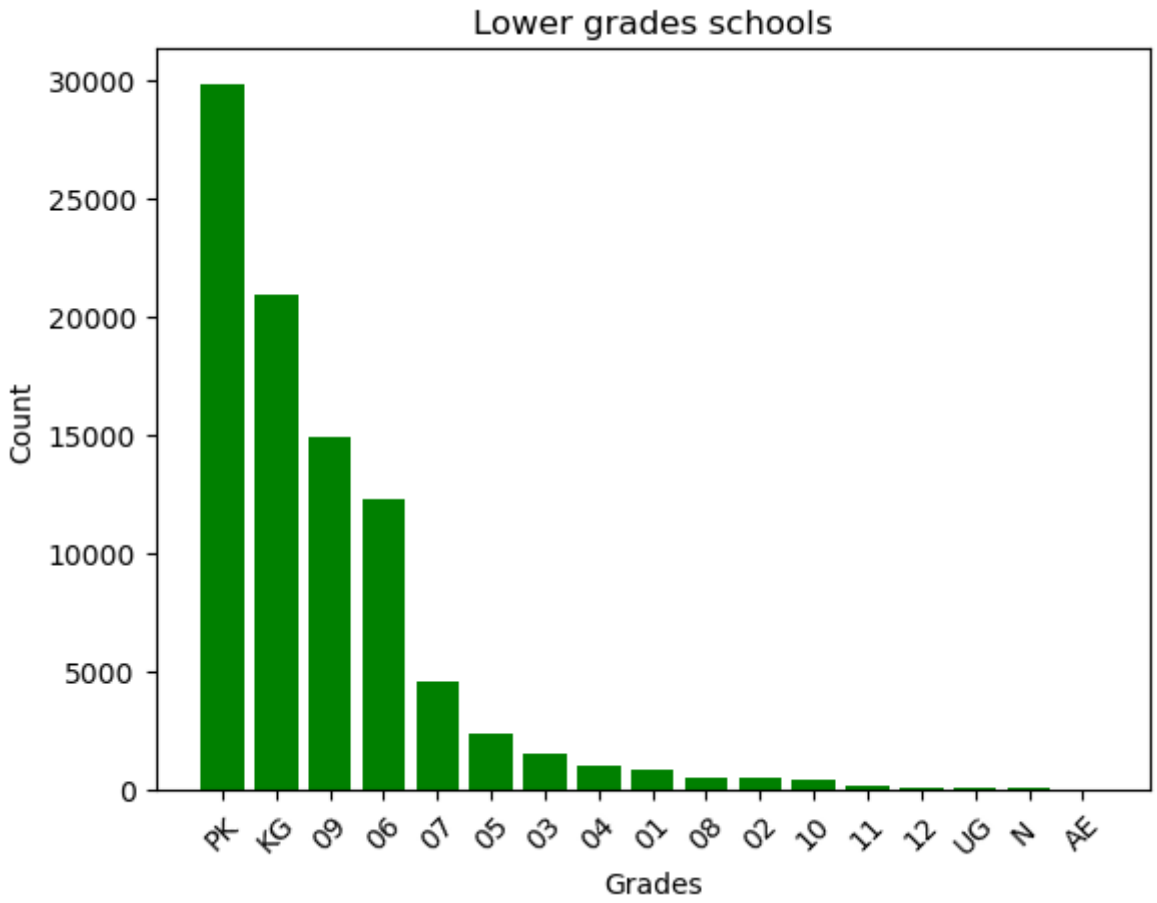
In [24]:
```
df['VIRTUAL'].unique()
```

Out[24]:
```
array(['Not Virtual', 'Full Virtual', 'Supplemental Virtual',
       'Virtual with face to face options', 'M'], dtype=object)
```

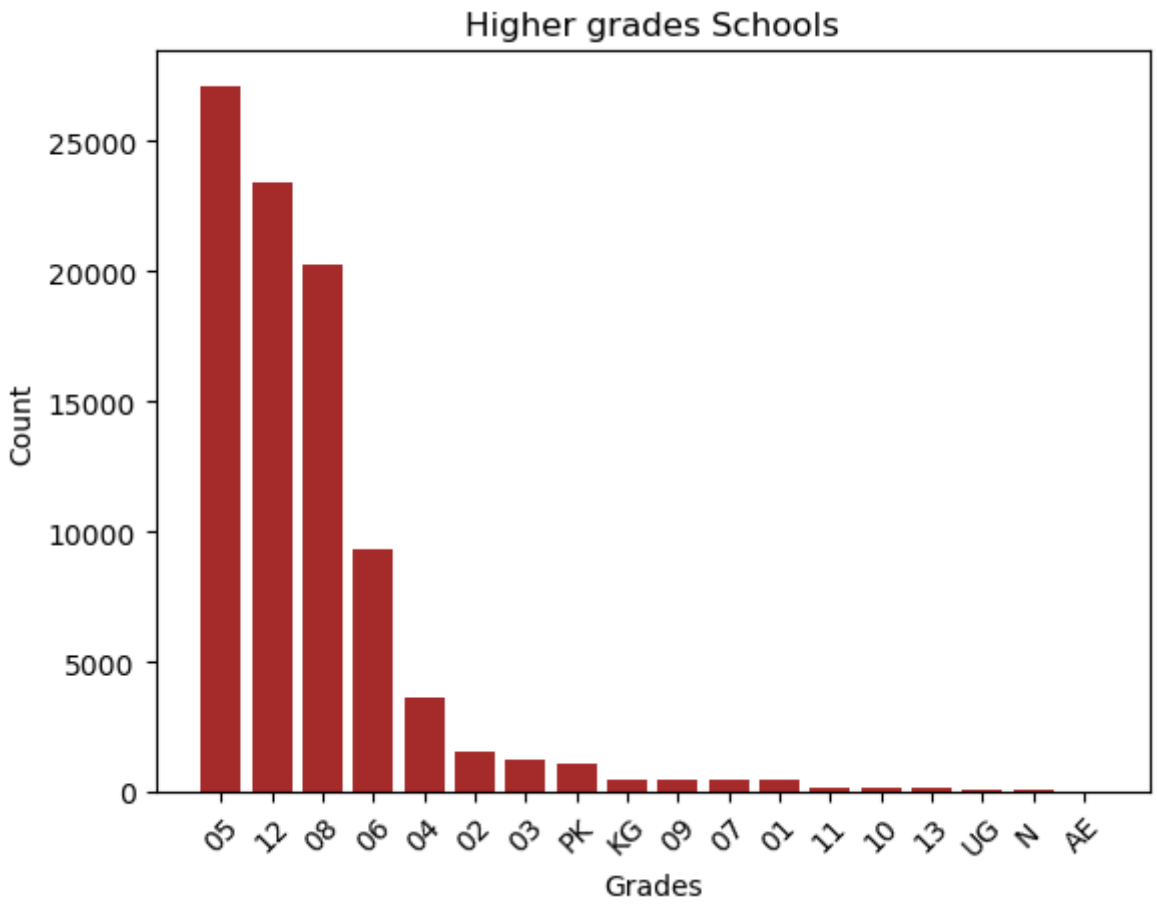
In [25]:
```
charter_counts = df['VIRTUAL'].value_counts()

labels = charter_counts.index
values = charter_counts.values

plt.bar(labels, values)

plt.title('Interaction Type')
plt.xlabel('Charter')
plt.ylabel('Count')
plt.xticks(rotation=45)

plt.show()
```



In [26]:
```
charter_counts = df['GSLO'].value_counts()

labels = charter_counts.index
values = charter_counts.values
bar_color = 'green'

plt.bar(labels, values, color=bar_color)

plt.title('Lower grades schools')
plt.xlabel('Grades')
plt.ylabel('Count')
plt.xticks(rotation=45)

plt.show()
```



In [27]:
```
charter_counts = df['GSHI'].value_counts()

labels = charter_counts.index
values = charter_counts.values
bar_color = 'brown'

plt.bar(labels, values, color=bar_color)

plt.title('Higher grades Schools')
plt.xlabel('Grades')
plt.ylabel('Count')
plt.xticks(rotation=45)

plt.show()
```



In [28]:
```
df.head(4)
```

Out[28]:

	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NAME	SCH_NAME	LSTREET1	LCITY	LSTATE
0	2020-2021	AL	100005	AL-101	Albertville City	Albertville Middle School	600 E Alabama Ave	Albertville	AL
1	2020-2021	AL	100005	AL-101	Albertville City	Albertville High School	402 E McCord Ave	Albertville	AL
2	2020-2021	AL	100005	AL-101	Albertville City	Albertville Intermediate School	901 W McKinney Ave	Albertville	AL
3	2020-2021	AL	100005	AL-101	Albertville City	Albertville Elementary School	145 West End Drive	Albertville	AL

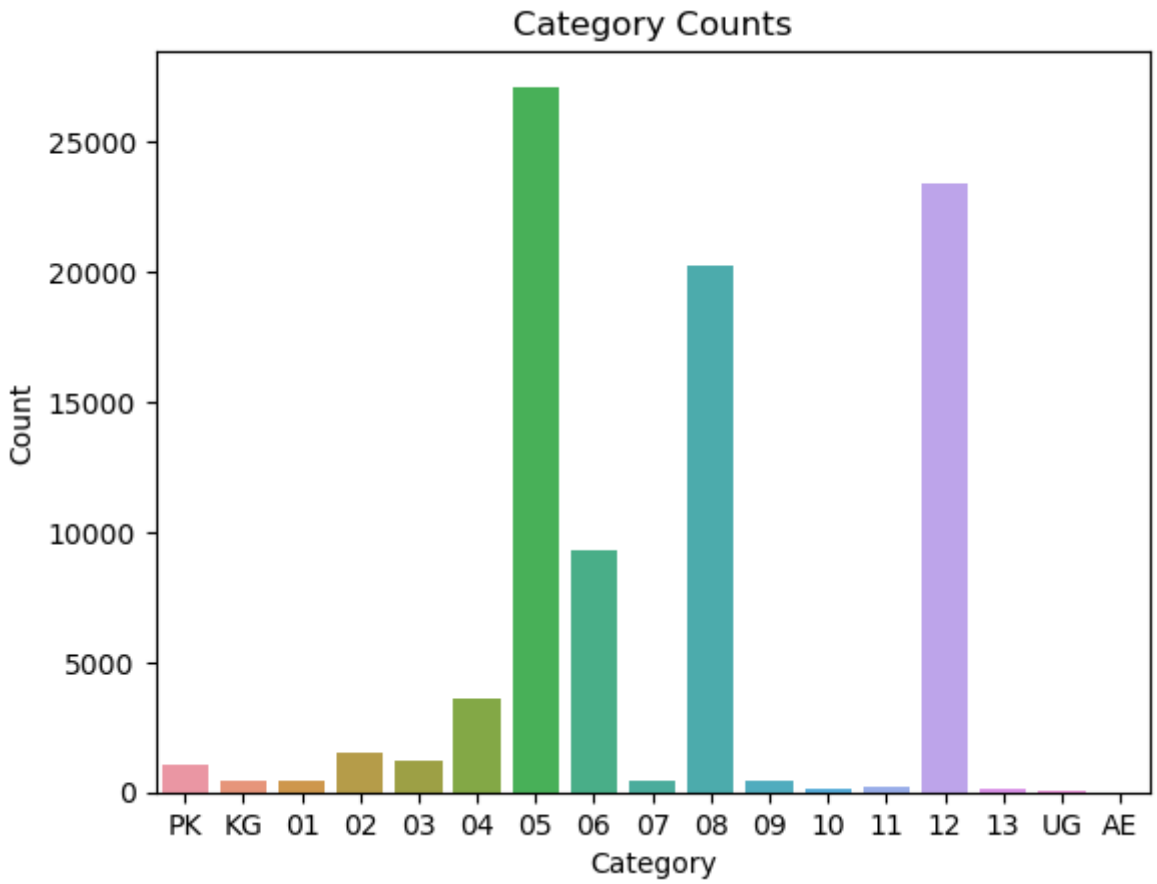
In [29]:
```
import seaborn as sns
```

In [30]:
```
df['GSHI'].unique()
```

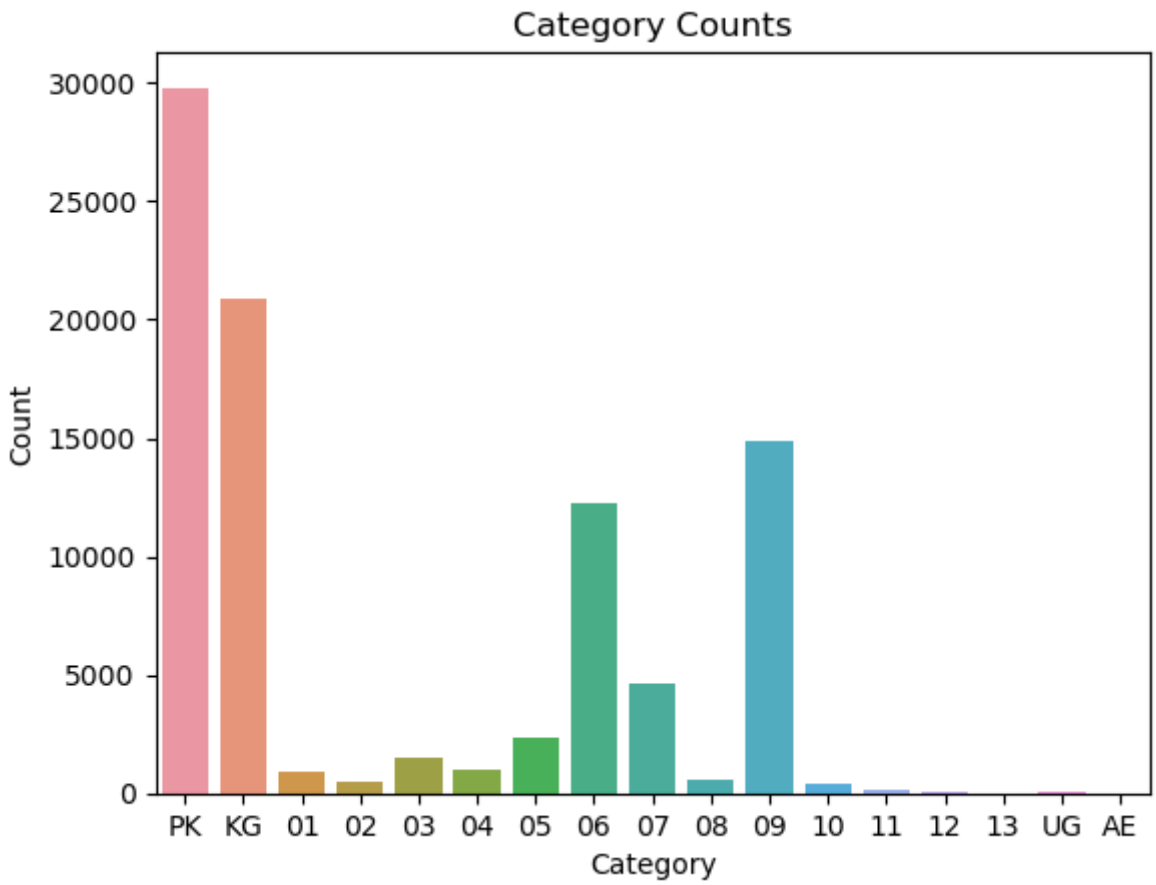
Out[30]:
```
array(['08', '12', '06', '04', 'KG', '02', '05', '03', '01', '09', '10',
       '07', 'PK', '11', 'UG', 'N ', '13', 'AE'], dtype=object)
```

In [31]:
```
custom_order = ['PK','KG','01','02','03','04','05','06','07','08','09','10','11','

df['GSHI'] = pd.Categorical(df['GSHI'], categories=custom_order, ordered=True)

sorted_df = df.sort_values('GSHI')
```

```
sns.countplot(data=sorted_df, x='GSHI')

plt.title('Category Counts')
plt.xlabel('Category')
plt.ylabel('Count')

plt.show()
```



In [32]:
```
df['GSLO'] = pd.Categorical(df['GSLO'], categories=custom_order, ordered=True)

sorted_df = df.sort_values('GSLO')

sns.countplot(data=sorted_df, x='GSLO')

plt.title('Category Counts')
plt.xlabel('Category')
plt.ylabel('Count')

plt.show()
```



In [33]:
```
df.head(6)
```

Out[33]:

	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NAME	SCH_NAME	LSTREET1	LCITY	LSTATE
0	2020-2021	AL	100005	AL-101	Albertville City	Albertville Middle School	600 E Alabama Ave	Albertville	AL
1	2020-2021	AL	100005	AL-101	Albertville City	Albertville High School	402 E McCord Ave	Albertville	AL
2	2020-2021	AL	100005	AL-101	Albertville City	Albertville Intermediate School	901 W McKinney Ave	Albertville	AL
3	2020-2021	AL	100005	AL-101	Albertville City	Albertville Elementary School	145 West End Drive	Albertville	AL
4	2020-2021	AL	100005	AL-101	Albertville City	Albertville Kindergarten and PreK	257 Country Club Rd	Albertville	AL
5	2020-2021	AL	100005	AL-101	Albertville City	Albertville Primary School	1100 Horton Rd	Albertville	AL

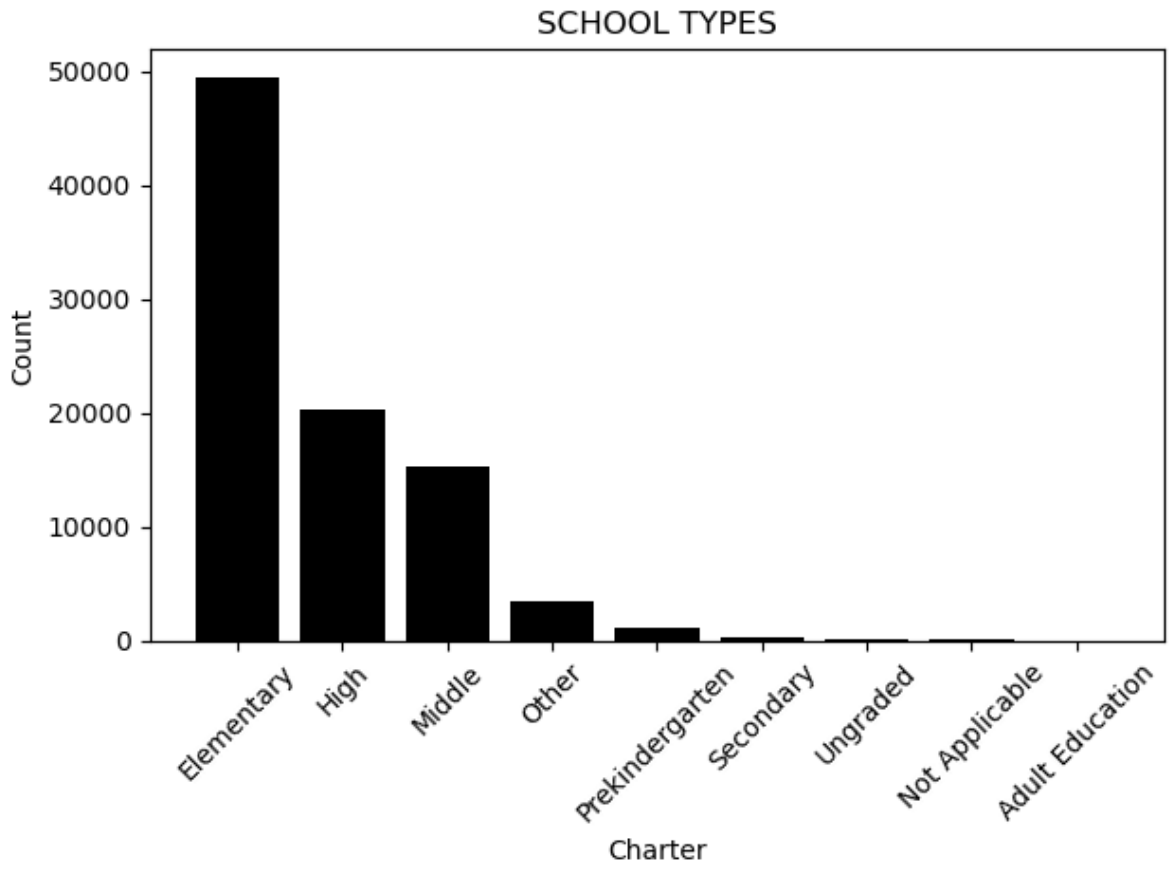
In [34]:
```
df['GSLO'].head(20)
```

Out[34]:
```
0     07
1     09
2     05
3     03
4     PK
5     01
6     05
7     06
9     03
10    09
11    03
12    09
13    PK
16    PK
17    06
18    PK
19    06
20    PK
21    PK
22    PK
Name: GSLO, dtype: category
Categories (17, object): ['PK' < 'KG' < '01' < '02' ... '12' < '13' < 'UG' < 'AE']
```

In [35]:
```
df['SCHOOL_LEVEL'].unique()
```

Out[35]:
```
array(['Middle', 'High', 'Elementary', 'Other', 'Secondary',
       'Prekindergarten', 'Ungraded', 'Not Applicable', 'Adult Education'],
      dtype=object)
```

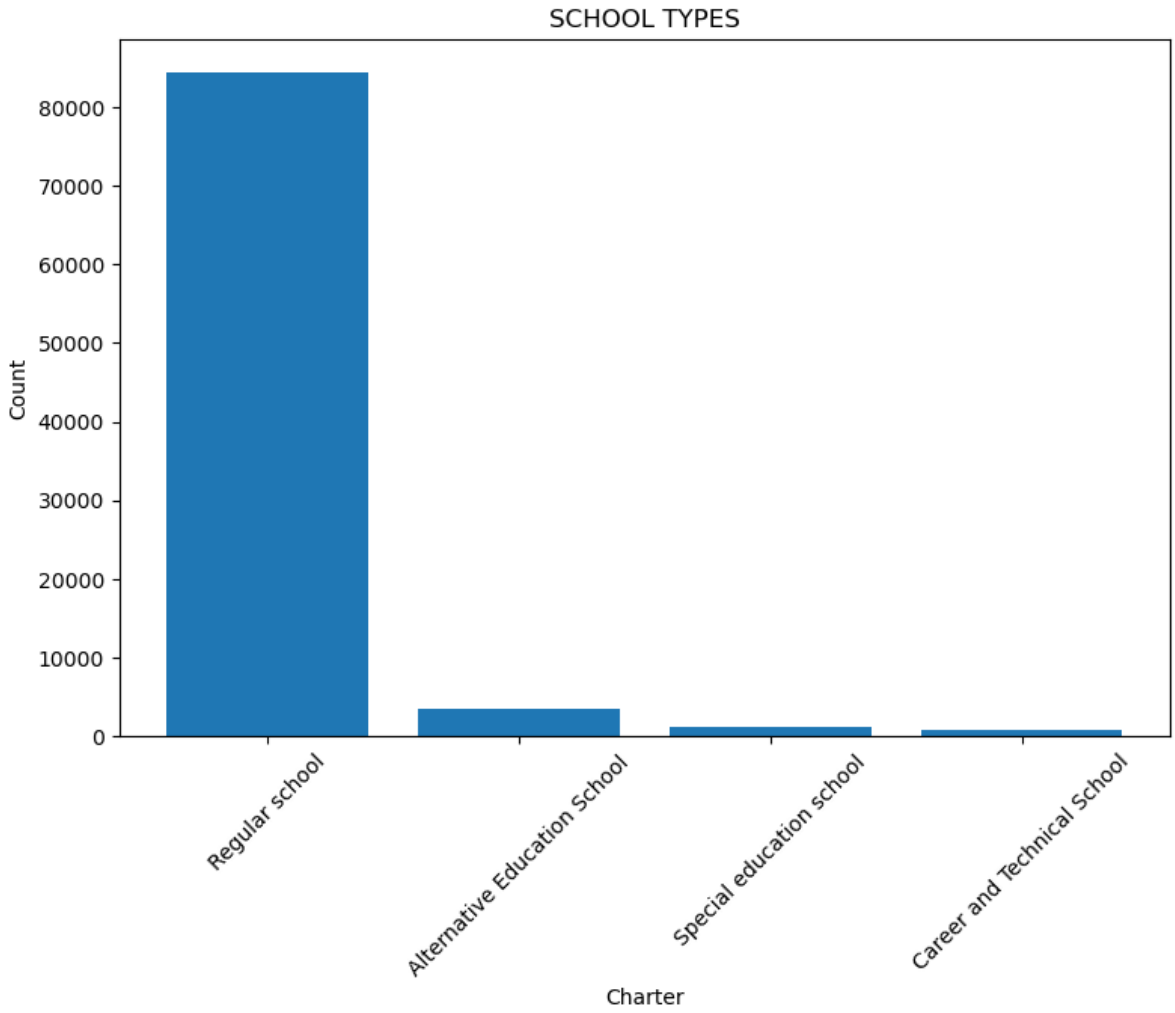
In [36]:
```
charter_counts = df['SCHOOL_LEVEL'].value_counts()

labels = charter_counts.index
values = charter_counts.values
bar_color='black'
plt.bar(labels, values,color=bar_color)

plt.title('SCHOOL TYPES')
plt.xlabel('Charter')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.tight_layout()

plt.show()
```



In [37]:
```
df['SCHOOL_TYPE_TEXT'].unique()
```

Out[37]:
```
array(['Regular school', 'Special education school',
       'Alternative Education School', 'Career and Technical School'],
      dtype=object)
```

In [38]:
```
charter_counts = df['SCHOOL_TYPE_TEXT'].value_counts()

labels = charter_counts.index
values = charter_counts.values

plt.figure(figsize=(9, 6))

plt.bar(labels, values)

plt.title('SCHOOL TYPES')
plt.xlabel('Charter')
plt.ylabel('Count')
plt.xticks(rotation=45)

plt.show()
```



In [39]:
```
attribute_columns = ['KG','G01','G02','G03','G04','G05','G06','G07','G08','G09','G

sum_values = df[attribute_columns].sum()

print(sum_values)
```

```
KG     3189874.0
G01    3326469.0
G02    3332491.0
G03    3353836.0
G04    3404069.0
G05    3440254.0
G06    3533804.0
G07    3640888.0
G08    3667536.0
G09    3793267.0
G10    3675358.0
G11    3481369.0
G12    3420053.0
G13       1943.0
UG      122316.0
AE        4595.0
PK      990539.0
dtype: float64
```

In [40]:
```
df.head(2)
```

Out[40]:

	SURVYEAR	STABR	LEAID	ST_LEAID	LEA_NAME	SCH_NAME	LSTREET1	LCITY	LSTATE
0	2020-2021	AL	100005	AL-101	Albertville City	Albertville Middle School	600 E Alabama Ave	Albertville	AL
1	2020-2021	AL	100005	AL-101	Albertville City	Albertville High School	402 E McCord Ave	Albertville	AL

In [41]:
```
correlation_matrix = df.corr()
```

In [42]:
```
correlation_matrix=correlation_matrix.round(2)
```

In [43]:
```
correlation_matrix
```

Out[43]:

	LEAID	LZIP	STATUS	PK	KG	G01	G02	G03	G04	G05	G06	G07	G0
LEAID	1.00	-0.21	0.01	0.09	-0.06	-0.04	-0.04	-0.04	-0.05	-0.04	-0.02	-0.02	-0.0
LZIP	-0.21	1.00	0.00	-0.03	0.02	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.0
STATUS	0.01	0.00	1.00	-0.00	-0.01	-0.01	-0.02	-0.02	-0.01	-0.01	-0.00	-0.01	-0.0
PK	0.09	-0.03	-0.00	1.00	0.32	0.27	0.24	0.19	0.17	0.11	-0.13	-0.14	-0.1
KG	-0.06	0.02	-0.01	0.32	1.00	0.89	0.84	0.71	0.65	0.49	-0.15	-0.22	-0.2
G01	-0.04	-0.01	-0.01	0.27	0.89	1.00	0.93	0.78	0.70	0.53	-0.16	-0.24	-0.2
G02	-0.04	-0.01	-0.02	0.24	0.84	0.93	1.00	0.83	0.74	0.55	-0.16	-0.24	-0.2
G03	-0.04	-0.01	-0.02	0.19	0.71	0.78	0.83	1.00	0.89	0.67	-0.15	-0.23	-0.2
G04	-0.05	-0.01	-0.01	0.17	0.65	0.70	0.74	0.89	1.00	0.77	-0.11	-0.22	-0.2
G05	-0.04	-0.01	-0.01	0.11	0.49	0.53	0.55	0.67	0.77	1.00	0.08	-0.14	-0.1
G06	-0.02	-0.01	-0.00	-0.13	-0.15	-0.16	-0.16	-0.15	-0.11	0.08	1.00	0.75	0.7
G07	-0.02	-0.01	-0.01	-0.14	-0.22	-0.24	-0.24	-0.23	-0.22	-0.14	0.75	1.00	0.9
G08	-0.02	-0.01	-0.01	-0.14	-0.23	-0.24	-0.24	-0.23	-0.22	-0.15	0.73	0.97	1.0
G09	-0.02	-0.00	-0.01	-0.12	-0.22	-0.23	-0.24	-0.23	-0.23	-0.21	-0.12	-0.09	-0.0
G10	-0.02	-0.00	-0.02	-0.12	-0.22	-0.24	-0.24	-0.24	-0.23	-0.21	-0.13	-0.10	-0.1
G11	-0.02	-0.00	-0.02	-0.12	-0.23	-0.24	-0.24	-0.24	-0.23	-0.21	-0.13	-0.10	-0.1
G12	-0.03	0.00	-0.02	-0.12	-0.23	-0.24	-0.25	-0.24	-0.24	-0.22	-0.13	-0.11	-0.1
G13	0.01	-0.02	-0.00	-0.01	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.01	-0.01	-0.0
UG	0.05	-0.15	-0.00	-0.00	-0.03	-0.03	-0.03	-0.03	-0.03	-0.02	-0.01	-0.01	-0.0
AE	0.00	0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.0
TOTAL	-0.06	-0.01	-0.03	-0.00	0.14	0.15	0.15	0.16	0.16	0.17	0.28	0.29	0.3
MEMBER	-0.06	-0.01	-0.03	-0.00	0.14	0.15	0.15	0.16	0.16	0.17	0.28	0.29	0.3
FTE	0.05	-0.17	-0.04	0.01	0.09	0.11	0.11	0.12	0.12	0.13	0.25	0.26	0.2
STUTERATIO	-0.02	0.03	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0
AMALM	0.02	0.11	-0.01	0.03	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.0
AMALF	0.01	0.10	-0.01	0.03	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.0
AM	0.02	0.11	-0.01	0.03	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.0
ASALM	-0.07	0.04	-0.01	-0.04	0.06	0.07	0.07	0.08	0.08	0.08	0.11	0.12	0.1
ASALF	-0.07	0.04	-0.01	-0.04	0.06	0.07	0.07	0.08	0.08	0.08	0.11	0.12	0.1
AS	-0.07	0.04	-0.01	-0.04	0.06	0.07	0.07	0.08	0.08	0.08	0.11	0.12	0.1
BLALM	-0.03	-0.19	-0.01	0.04	0.07	0.08	0.08	0.08	0.08	0.08	0.15	0.15	0.1
BLALF	-0.03	-0.18	-0.01	0.03	0.07	0.08	0.08	0.08	0.08	0.08	0.15	0.15	0.1
BL	-0.03	-0.18	-0.01	0.04	0.07	0.08	0.08	0.08	0.08	0.08	0.15	0.15	0.1
HPALM	-0.01	0.10	-0.01	-0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.0
HPALF	-0.01	0.10	-0.01	-0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.03	0.03	0.0
HP	-0.01	0.10	-0.01	-0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.0

	LEAID	LZIP	STATUS	PK	KG	G01	G02	G03	G04	G05	G06	G07	G0
HIALM	-0.08	0.19	-0.02	0.01	0.08	0.08	0.08	0.09	0.09	0.09	0.17	0.18	0.1
HIALF	-0.08	0.19	-0.02	0.01	0.09	0.09	0.09	0.09	0.09	0.10	0.17	0.18	0.1
HI	-0.08	0.19	-0.02	0.01	0.09	0.08	0.09	0.09	0.09	0.09	0.17	0.18	0.1
TRALM	-0.05	0.07	-0.02	0.07	0.24	0.24	0.24	0.23	0.23	0.22	0.20	0.20	0.2
TRALF	-0.04	0.06	-0.01	0.06	0.24	0.23	0.23	0.23	0.23	0.22	0.20	0.20	0.2
TR	-0.04	0.07	-0.02	0.07	0.24	0.24	0.24	0.24	0.23	0.23	0.20	0.21	0.2
WHALM	0.02	-0.13	-0.02	-0.03	0.08	0.08	0.08	0.08	0.09	0.11	0.19	0.20	0.2
WHALF	0.02	-0.13	-0.02	-0.03	0.09	0.08	0.08	0.09	0.09	0.11	0.19	0.20	0.2
WH	0.02	-0.13	-0.02	-0.03	0.08	0.08	0.08	0.09	0.09	0.11	0.19	0.20	0.2
LATCOD	0.05	-0.07	0.00	-0.09	-0.11	-0.11	-0.11	-0.12	-0.12	-0.10	-0.07	-0.06	-0.0
LONCOD	0.21	0.87	0.00	0.07	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.0

In [46]:
```
from sklearn.model_selection import train_test_split
```

In [47]:
```
X = df.drop('TOTAL', axis=1)
y = df['TOTAL']
```

In [48]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta
```

In [49]:
```
print("Training Features Shape:", X_train.shape)
print("Testing Features Shape:", X_test.shape)
print("Training Labels Shape:", y_train.shape)
print("Testing Labels Shape:", y_test.shape)
```

```
Training Features Shape: (71972, 67)
Testing Features Shape: (17993, 67)
Training Labels Shape: (71972,)
Testing Labels Shape: (17993,)
```

In [ ]:

In [ ]: