Code and Outputs

[Data Cleaning]

Penalties

```
# merging the files since 2020 -2021 has different column names , dropping the columns not required

# ------------> starting with 2015 to 2019
# Convert 'Date_Column' to datetime format
penality_df['filedate'] = pd.to_datetime(penality_df['filedate'])
# Extract year component
penality_df['YearOfFileDate'] = penality_df['filedate'].dt.year
# Specify columns you want to drop
columns_to_drop = ['provname','address','city','zip','filedate','payden_strt_dt']
P_df = penality_df.drop(columns=columns_to_drop)

#---------------> for year 2020 to 2021
Covid_df['Processing Date'] = pd.to_datetime(Covid_df['Processing Date'])
# Extract year component
Covid_df['YearOfFileDate'] = Covid_df['Processing Date'].dt.year
Covid_df.drop(columns= ['Provider Name', 'Provider Address',
       'Provider City' ,'Provider Zip Code', 'Location','Processing Date','Penalty Date','Payment Denial Start Date'],inplace=True)
# Displaying the DataFrame after dropping columns
print("\nDataFrame after dropping columns and converting into years for filedate columns:")
P_df.head(5)
Covid_df.head(5)
```

```
DataFrame after dropping columns and converting into years for filedate columns:
```

	Federal Provider Number	Provider State	Penalty Type	Fine Amount	Payment Denial Length in Days	YearOfFileDate
0	15015	AL	Fine	22903.0	NaN	2020
1	15015	AL	Fine	6708.0	NaN	2020
2	15019	AL	Fine	78677.0	NaN	2020
3	15028	AL	Fine	6500.0	NaN	2020
4	15031	AL	Fine	31639.0	NaN	2020

```
# Dictionary mapping old column names to new column names
new_column_names = {'Federal Provider Number':'provnum', 'Provider State':'state',
       'Penalty Type':'pnlty_type', 'Fine Amount':'fine_amt',
       'Payment Denial Length in Days':'payden_days'}

# Rename the columns
Covid_df.rename(columns=new_column_names, inplace=True)
```

```
# concenating all them together
df_Penalities_2015to2021=pd.concat([P_df, Covid_df], ignore_index=True)
df_Penalities_2015to2021.head(5)
# shape of new dataframe is : (71488, 8)
```

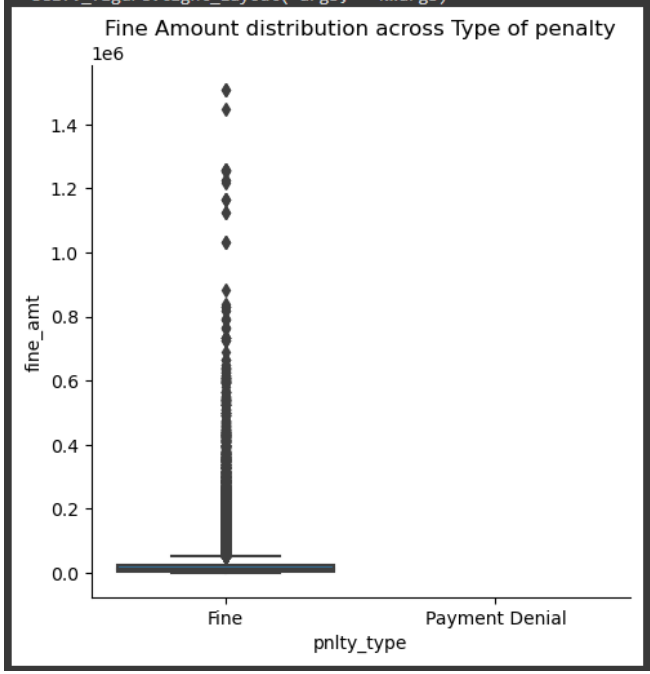
	provnum	state	pnlty_date	pnlty_type	fine_amt	payden_days	YearOfFileDate
0	15019	AL	2014-10-02	Fine	6692.0	NaN	2015
1	15037	AL	2015-05-21	Fine	13813.0	NaN	2015
2	15053	AL	2014-05-16	Fine	142870.0	NaN	2015
3	15053	AL	2014-05-16	Payment Denial	NaN	5.0	2015
4	15060	AL	2015-04-16	Fine	58273.0	NaN	2015

```
# counting the missing values in columns
missing_values_count = df_Penalities_2015to2021.isnull().sum()
missing_values_count
#total_missing = missing_values_count.sum()
#total_missing
```

```
provnum              0
state                0
pnlty_date       32786
pnlty_type           0
fine_amt         10856
payden_days      60635
YearOfFileDate       0
dtype: int64
```

```
#detecting outlier
sns.catplot(data=df_Penalities_2015to2021,x='pnlty_type',y='fine_amt', kind = 'box')
plt.title('Fine Amount distribution across Type of penalty')
plt.show()
```

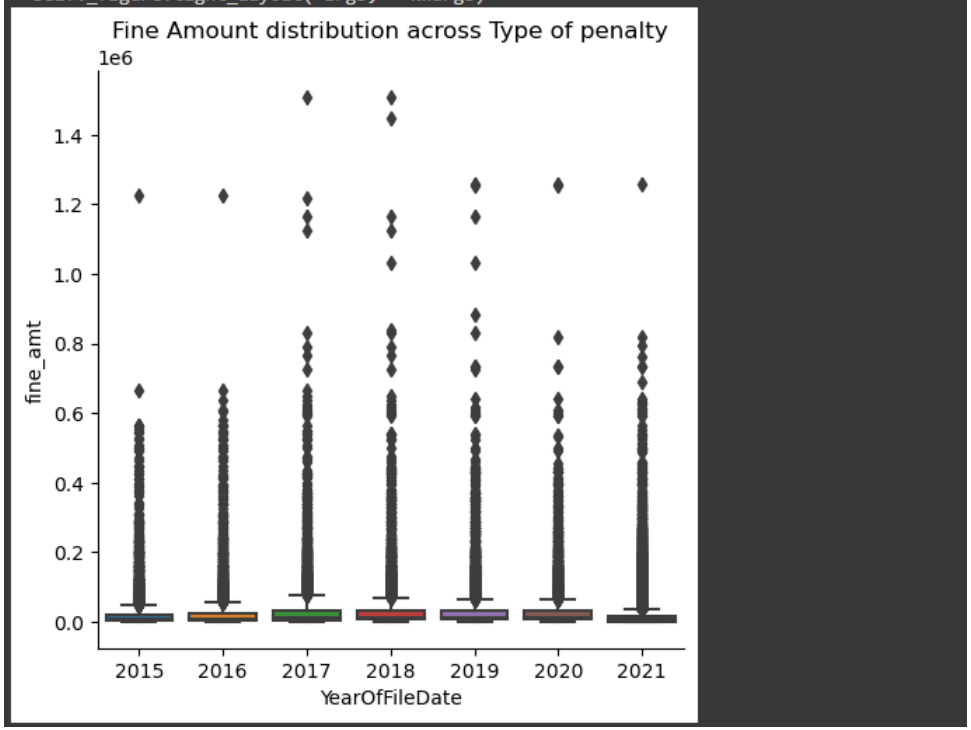
```
c:\Users\kanak\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The
  self._figure.tight_layout(*args, **kwargs)
```



df_Penalities_2015to2021['fine_amt'].fillna(df_Penalities_2015to2021['fine_amt'].mean(), inplace=True) We should not run this line as we found fine amount is only exist null when we have payment denial so not good to replace them !

```
#detecting outlier
sns.catplot(data=df_Penalities_2015to2021,x='YearOfFileDate',y='fine_amt', kind = 'box')
plt.title('Fine Amount distribution across Type of penalty')
plt.show()
```

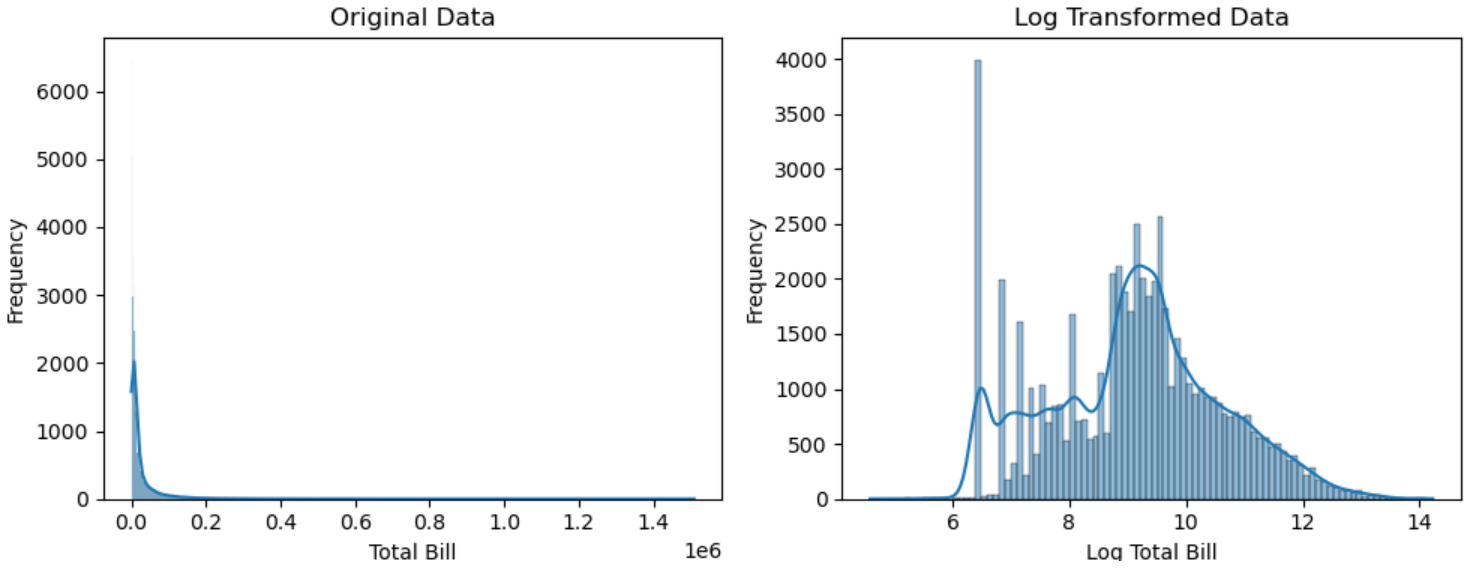
```
c:\Users\kanak\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figu
  self._figure.tight_layout(*args, **kwargs)
```



Although there are many outlier, but i think fine amt gives exact picture of how well the nursing homes are : thus we are not using below code to remove outliers ___ from scipy.stats.mstats import winsorize winsorized_values = winsorize(df_Penalities_2015to2021['fine_amt'], limits=[0.03, 0.03])

---Replace original column 'A' with winsorized values--- df_Penalities_2015to2021['payden_days'] = winsorized_values

```
total_bill = df_Penalities_2015to2021['fine_amt']
# Plotting histograms of original and transformed data
plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
sns.histplot(total_bill, kde=True)
plt.xlabel('Total Bill')
plt.ylabel('Frequency')
plt.title('Original Data')
plt.subplot(1, 2, 2)
sns.histplot(np.log(total_bill), kde=True)
plt.xlabel('Log Total Bill')
plt.ylabel('Frequency')
plt.title('Log Transformed Data')
# Display the plots
plt.tight_layout()
plt.show()
```



Merged Cost Reports

```
#standardizing column names
cr2020.rename(columns={'Provider CCN': 'Provider_CCN', 'Facility Name': 'Facility_Name',
                       'Street Address': 'Street_Address', 'State Code': 'State_Code', 'Zip Code': 'Zip_Code',
                       'Medicare CBSA Number': 'Medicare_CBSA_Number', 'Rural versus Urban': 'Rural_versus_Urban',
                       'Fiscal Year Begin Date': 'Fiscal_Year_Begin_Date','Fiscal Year End Date': 'Fiscal_Year_End_Date',
                       'Type of Control': 'Type_of_Control', 'Total Days Title V': 'Total_Days_Title_V',
                       'Total Days Title XVIII': 'Total_Days_Title_XVIII', 'Total Days Title XIX': 'Total_Days_Title_XIX',
                       'Total Days Other': 'Total_Days_Other', 'Total Days Total': 'Total_Days_Total',
                       'Number of Beds': 'Number_of_Beds', 'Total Bed Days Available': 'Total_Bed_Days_Available',
                       'Total Discharges Title V': 'Total_Discharges_Title_V','Total Discharges Title XVIII': 'Total_Discharges_Title_XVIII',
                       'Total Discharges Title XIX': 'Total_Discharges_Title_XIX', 'Total Discharges Title Other': 'Total_Discharges_Title_Other',
                       'Total Discharges Total': 'Total_Discharges_Total', 'SNF Average Length of Stay Title V': 'SNF_Average_Length_stay_V',
                       'SNF Average Length of Stay Title XVIII': 'SNF_Average_Length_stay_XVIII',
                       'SNF Average Length of Stay Title XIX': 'SNF_Average_Length_stay_XIX', 'SNF Average Length of Stay Total': 'SNF_Average_Length_of_stay_Tot',
                       'SNF Admissions Title V': 'SNF_Admissions_Title_V', 'SNF Admissions Title XVIII': 'SNF_Admissions_Title_XVIII',
                       'SNF Admissions Title XIX': 'SNF_Admissions_Title_XIX','SNF Admissions Total': 'SNF_Admissions_Total','SNF Days Title V': 'SNF_Days_Title_V',
                       'SNF Days Title XVIII': 'SNF_Days_Title_XVIII','SNF Days Title XIX': 'SNF_Days_Title_XIX', 'SNF Days Other': 'SNF_Days_Other',
                       'SNF Days Total': 'SNF_Days_Total','SNF Number of Beds':'SNF_Number_of_Beds','SNF Bed Days Available':'SNF_Bed_Days_Available',
                       'SNF Discharges Title V':'SNF_Discharges_Title_V', 'SNF Discharges Title XVIII':'SNF_Discharges_Title_XVIII',
                       'SNF Discharges Title XIX':'SNF_Discharges_Title_XIX','SNF Discharges Title Other':'SNF_Discharges_Title_Other',
                       'SNF Discharges Total':'SNF_Discharges_Total','Total RUG Days':'Total_RUG_Days', 'Total Salaries From Worksheet A':'Total_Salaries_From_Worksheet_A',
                       'Overhead Non-Salary Costs':'Overhead_Non_Salary_Costs', 'Total Costs': 'Total_Costs', 'Wage-related Costs (core)':'Wage_related_Costs_core',
                       'Total Salaries (adjusted)':'Total_Salaries_adjusted', 'Cash on hand and in banks': 'Cash_on_hand_and_in_banks',
                       'Accounts Receivable':'Accounts_Receivable', 'Total Current Assets':'Total_current_assets','Fixed equipment':'Fixed_equipment',
                       'Major movable equipment':'Major_movable_equipment','Total fixed Assets':'Total_fixed_assets','Other Assets':'Other_Assets',
                       'Total other Assets':'Total_other_Assets','Total Assets':'Total_Assets','Accounts payable':'Accounts_payable',
                       'Salaries, wages, and fees payable':'Salaries_wages_and_fees_payable','Other current liabilities':'Other_current_liabilities',
                       'Total current liabilities':'Total_current_liabilities','Total liabilities':'Total_liabilities','General fund balance':'General_fund_balance',
                       'Total fund balances':'Total_fund_balances','Total General Inpatient Care Services Revenue':'Total_General_Inpatient_Revenue',
                       'Inpatient Revenue':'Inpatient_Revenue','Gross Revenue':'Gross_Revenue', 'Net Patient Revenue':'Net_Patient_Revenue',
                       'Less Total Operating Expense':'Less_Total_Operating_Expense','Net Income from service to patients':'Net_Income_from_patients','Net Income':'Net_Income',
                       'Inpatient PPS Amount':'Inpatient_PPS_Amount'}, inplace=True)
```

```
#standardizing column names
cr2021.rename(columns={'Provider CCN': 'Provider_CCN', 'Facility Name': 'Facility_Name',
                       'Street Address': 'Street_Address', 'State Code': 'State_Code', 'Zip Code': 'Zip_Code',
                       'Medicare CBSA Number': 'Medicare_CBSA_Number', 'Rural versus Urban': 'Rural_versus_Urban',
                       'Fiscal Year Begin Date': 'Fiscal_Year_Begin_Date','Fiscal Year End Date': 'Fiscal_Year_End_Date',
                       'Type of Control': 'Type_of_Control', 'Total Days Title V': 'Total_Days_Title_V',
                       'Total Days Title XVIII': 'Total_Days_Title_XVIII', 'Total Days Title XIX': 'Total_Days_Title_XIX',
                       'Total Days Other': 'Total_Days_Other', 'Total Days Total': 'Total_Days_Total',
                       'Number of Beds': 'Number_of_Beds', 'Total Bed Days Available': 'Total_Bed_Days_Available',
                       'Total Discharges Title V': 'Total_Discharges_Title_V','Total Discharges Title XVIII': 'Total_Discharges_Title_XVIII',
                       'Total Discharges Title XIX': 'Total_Discharges_Title_XIX', 'Total Discharges Title Other': 'Total_Discharges_Title_Other',
                       'Total Discharges Total': 'Total_Discharges_Total', 'SNF Average Length of Stay Title V': 'SNF_Average_Length_stay_V',
                       'SNF Average Length of Stay Title XVIII': 'SNF_Average_Length_stay_XVIII',
                       'SNF Average Length of Stay Title XIX': 'SNF_Average_Length_stay_XIX', 'SNF Average Length of Stay Total': 'SNF_Average_Length_of_stay_Tot',
                       'SNF Admissions Title V': 'SNF_Admissions_Title_V', 'SNF Admissions Title XVIII': 'SNF_Admissions_Title_XVIII',
                       'SNF Admissions Title XIX': 'SNF_Admissions_Title_XIX','SNF Admissions Total': 'SNF_Admissions_Total','SNF Days Title V': 'SNF_Days_Title_V',
                       'SNF Days Title XVIII': 'SNF_Days_Title_XVIII','SNF Days Title XIX': 'SNF_Days_Title_XIX', 'SNF Days Other': 'SNF_Days_Other',
                       'SNF Days Total': 'SNF_Days_Total','SNF Number of Beds':'SNF_Number_of_Beds','SNF Bed Days Available':'SNF_Bed_Days_Available',
                       'SNF Discharges Title V':'SNF_Discharges_Title_V', 'SNF Discharges Title XVIII':'SNF_Discharges_Title_XVIII',
                       'SNF Discharges Title XIX':'SNF_Discharges_Title_XIX','SNF Discharges Title Other':'SNF_Discharges_Title_Other',
                       'SNF Discharges Total':'SNF_Discharges_Total','Total RUG Days':'Total_RUG_Days', 'Total Salaries From Worksheet A':'Total_Salaries_From_Worksheet_A',
                       'Overhead Non-Salary Costs':'Overhead_Non_Salary_Costs', 'Total Costs': 'Total_Costs', 'Wage-related Costs (core)':'Wage_related_Costs_core',
                       'Total Salaries (adjusted)':'Total_Salaries_adjusted', 'Cash on hand and in banks': 'Cash_on_hand_and_in_banks',
                       'Accounts Receivable':'Accounts_Receivable', 'Total Current Assets':'Total_current_assets','Fixed equipment':'Fixed_equipment',
                       'Major movable equipment':'Major_movable_equipment','Total fixed Assets':'Total_fixed_assets','Other Assets':'Other_Assets',
                       'Total other Assets':'Total_other_Assets','Total Assets':'Total_Assets','Accounts payable':'Accounts_payable',
                       'Salaries, wages, and fees payable':'Salaries_wages_and_fees_payable','Other current liabilities':'Other_current_liabilities',
                       'Total current liabilities':'Total_current_liabilities','Total liabilities':'Total_liabilities','General fund balance':'General_fund_balance',
                       'Total fund balances':'Total_fund_balances','Total General Inpatient Care Services Revenue':'Total_General_Inpatient_Revenue',
                       'Inpatient Revenue':'Inpatient_Revenue','Gross Revenue':'Gross_Revenue', 'Net Patient Revenue':'Net_Patient_Revenue',
                       'Less Total Operating Expense':'Less_Total_Operating_Expense','Net Income from service to patients':'Net_Income_from_patients','Net Income':'Net_Income',
                       'Inpatient PPS Amount':'Inpatient_PPS_Amount'}, inplace=True)
```

IncomeStatement

```
# Read in Data, Create List with all Variable with greater than 90% missing values and drop from df
df = pd.read_csv(r"C:\Users\joshd\OneDrive\Desktop\ \BANA 620\PROJECT\Data\MergedCostReports.csv")

badData = df[['Total_Days_Title_V','Total_Discharges_Title_V','SNF_Average_Length_stay_V','SNF_Admissions_Title_V',
         'SNF_Days_Title_V','SNF_Discharges_Title_V','NF Number of Beds','NF Number of Beds','NF Bed Days Available',
         'NF Days Title V','NF Days Title XIX','NF Days Other','NF Days Total','NF Discharges Title V','NF Discharges Title XIX',
         'NF Discharges Title Other','NF Discharges Total','NF Average Length of Stay Title V','NF Average Length of Stay Title XIX',
         'NF Average Length of Stay Total','NF Admissions Title V','NF Admissions Title XIX','NF Admissions Other',
         'NF Admissions Total','Nursing and Allied Health Education Activities','Outpatient Revenue','Unsecured Loans',
         'Minor equipment depreciable','Temporary Investments','Notes Receivable','Buildings','Inventory',
            'Prepaid expenses','Leasehold improvements','Payroll taxes payable','Deferred income','Contract Labor',
                    'Less: Allowances for uncollectible notes and accounts receivable','Total Other Income']]

df['Year'] = pd.DatetimeIndex(df['Fiscal_Year_End_Date']).year
df.drop(['Fiscal_Year_End_Date'], axis=1, inplace=True)
df.drop(['Fiscal_Year_Begin_Date'], axis=1, inplace=True)

df.drop(badData, axis=1, inplace=True)
```

```
# combine columns with similar names

df['Total_Income'] = df['Total_Income'].combine_first(df['Total Income'])
df.drop('Total Income', axis=1, inplace=True)

df['Total_Liab_and_fund_balances'] = df['Total_Liab_and_fund_balances'].combine_first(df['Total Liabilities and fund balances'])
df.drop('Total Liabilities and fund balances', axis=1, inplace=True)

df['Total_fixed_Assets'] = df['Total_fixed_Assets'].combine_first(df['Total_fixed_assets'])
df.drop('Total_fixed_assets', axis=1, inplace=True)

df['SNF_Admissions_Other'] = df['SNF_Admissions_Other'].combine_first(df['SNF Admissions Other'])
df.drop('SNF Admissions Other', axis=1, inplace=True)

df['SNF_Number_of_beds'] = df['SNF_Number_of_beds'].combine_first(df['SNF_Number_of_Beds'])
df.drop('SNF_Number_of_Beds', axis=1, inplace=True)

df['SNF_bed_Days_Available'] = df['SNF_bed_Days_Available'].combine_first(df['SNF_Bed_Days_Available'])
df.drop('SNF_Bed_Days_Available', axis=1, inplace=True)
```

```
# Create new df with only Property Details

propData = df[['Year','Provider_CCN','Facility_Name','Street_Address','City','State_Code','Zip_Code','County','Medicare_CBSA_Number','Type_of_Control','Rural_versus_Urban']]
```

```
# Create new df with only Balance Sheet related factors

balanceSheetData = df[['Year','Provider_CCN','Accounts_Receivable','Accounts_payable','Cash_on_hand_and_in_banks','General_fund_balance',
                       'Major_movable_equipment','Total_Assets','Total_Liab_and_fund_balances','Total_current_assets','Total_current_liabilities',
                       'Total_fixed_Assets','Total_fund_balances','Total_liabilities','Total_other_Assets','Fixed_equipment','Other current assets',
                       'Land','Land improvements','Investments','Notes and Loans Payable (short term)','Mortgage payable','Notes Payable',
                       'Total long term liabilities','Other long term liabilities','Other_Assets','Other_current_liabilities']]
```

```
# Create new df with only Income Statement related factors

incomeStatementData = df[['Year','Gross_Revenue','Inpatient_PPS_Amount','Inpatient_Revenue','Less_Total_Operating_Expense','Less_discounts_on_patients',
                      'Net_Income','Net_Income_from_patients','Net_Patient_Revenue','Overhead_Non_Salary_Costs','Salaries_wages_and_fees_payable',
                      'Total_Costs','Total_General_Inpatient_Revenue','Total_Income','Total_RUG_Days','Total_Salaries_From_Worksheet_A','Total_Salaries_adjusted',
                      'Wage_related_Costs_core','Total Charges','Allowable Bad Debts']]
```

```
# Create new df with only Rent Roll related factors

rentRollData = df[['Year','Provider_CCN','Number_of_Beds','SNF_Admissions_Other','SNF_Admissions_Title_XIX','SNF_Admissions_Title_XVIII',
                   'SNF_Admissions_Total','SNF_Average_Length_of_stay_Tot','SNF_Average_Length_stay_XIX','SNF_Average_Length_stay_XVIII',
                   'SNF_Days_Other','SNF_Days_Title_XIX','SNF_Days_Title_XVIII','SNF_Days_Total','SNF_Discharges_Title_Other','SNF_Discharges_Title_XIX',
                   'SNF_Discharges_Title_XVIII','SNF_Discharges_Total','SNF_Number_of_beds','SNF_bed_Days_Available','Total_Bed_Days_Available',
                   'Total_Days_Other','Total_Days_Title_XIX','Total_Days_Title_XVIII','Total_Days_Total','Total_Discharges_Title_Other',
                   'Total_Discharges_Title_XIX','Total_Discharges_Title_XVIII','Total_Discharges_Total']]
```

Property

```
#Standardize ZipCodes
def clean_zip(Zip_Code):
    # Remove non-digit characters
    cleaned_zip = ''.join(c for c in str(Zip_Code) if c.isdigit())
    # Take the first 5 digits
    standardized_zip = cleaned_zip[:5]
    return standardized_zip

# Clean the zip code column
propertyDataClean.loc[:,'Zip_Code'] = propertyDataClean['Zip_Code'].apply(clean_zip)

print(propertyDataClean['Zip_Code'])
```

```
0         35801
1         21224
2         46052
3         47303
4         47804
          ...
106264    98118
106265    75231
106266    76903
106267    78727
106268    77979
Name: Zip_Code, Length: 104026, dtype: object
```

Rent Roll

```
rentRollData['Year'].unique()
```

```
array([2015., 2016., 2014.,   nan, 2017., 2018., 2019., 2020., 2021.,
       2022.])
```

```
# finding count of null years
nan_values = rentRollData[rentRollData['Year'].isna()]
nan_values

# the above code give 30 columns and 2072 rows and even this data set has all values as null , next we will drop all those values
```

	Year	State_Code	Provider_CCN	Number_of_Beds	SNF_Admissions_Other	SNF_Admissions_Title_XIX	SNF_Admissions_Title_XVIII	SNF_Admissions_Total	SNF_Average_Length_of_stay_Tot	SNF_Average_Length_stay_XIX	...
236	NaN	NY	335532	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
306	NaN	MN	245547	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
356	NaN	OR	385164	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
390	NaN	SD	435062	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
496	NaN	RI	415089	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
...	...	...	...	...	...	...	...	...	...	...	...
103924	NaN	IA	165791	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
103966	NaN	NM	325120	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
104027	NaN	OR	385283	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
104089	NaN	WY	535021	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...
104812	NaN	SD	435135	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...

2072 rows × 30 columns

```
rentRollData = rentRollData.dropna(subset=['Year'])

# Checking for null values again
nan_values = rentRollData[rentRollData['Year'].isna()]
nan_values
```

Year	State_Code	Provider_CCN	Number_of_Beds	SNF_Admissions_Other

0 rows × 30 columns

```
rentRollData['Years'].unique()
```

```
array([2015, 2016, 2014, 2017, 2018, 2019, 2020, 2021, 2022])
```

```
# Drop rows where the 'year' column is equal to 2014
rentRollData = rentRollData[rentRollData['Years'] != 2014]
```

```
rentRollData.describe()
```

rentRollData.describe()

	Provider_CCN	Number_of_Beds	SNF_Admissions_Other	SNF_Admissions_Title_XIX	SNF_Admissions_Title_XVIII	SNF_Admissions_Total	SNF_Average_Length_of_stay_Tot	SNF_Average_Length_stay_XIX	SNF_Average_Length_stay_XVIII	SNF_Days_Other	...
count	104067.000000	1.037370e+05	101795.000000	90505.000000	102822.000000	103156.000000	103768.000000	91248.000000	103431.000000	103190.000000	...
mean	297528.697435	2.364933e+02	108.274375	45.826882	124.562292	271.428429	172.270675	592.405074	48.170132	8776.106086	...
std	175817.649479	3.739494e+04	141.463675	118.125764	148.785116	302.898947	1009.074984	2751.344743	69.657783	9281.823456	...
min	15009.000000	1.000000e+00	1.000000	1.000000	1.000000	1.000000	0.040000	0.090000	0.030000	1.000000	...
25%	155325.000000	7.800000e+01	27.000000	12.000000	40.000000	102.000000	82.910000	220.520000	29.760000	3315.000000	...
50%	265800.000000	1.080000e+02	66.000000	28.000000	83.000000	202.000000	127.230000	352.765000	39.020000	6149.000000	...
75%	395765.000000	1.380000e+02	142.000000	58.000000	161.000000	362.000000	196.342500	598.505000	53.590000	10731.000000	...
max	745001.000000	1.204380e+07	16797.000000	22390.000000	19132.000000	40155.000000	316646.870000	754822.080000	15196.000000	403352.000000	...

8 rows × 29 columns

```
# counting the missing values in columns
missing_values_count = rentRollData.isnull().sum()
missing_values_count
#total_missing = missing_values_count.sum()
#total_missing
```

```
State_Code                          0
Provider_CCN                        0
Number_of_Beds                    330
SNF_Admissions_Other             2272
SNF_Admissions_Title_XIX        13562
SNF_Admissions_Title_XVIII       1245
SNF_Admissions_Total              911
SNF_Average_Length_of_stay_Tot    299
SNF_Average_Length_stay_XIX     12819
SNF_Average_Length_stay_XVIII     636
SNF_Days_Other                    877
SNF_Days_Title_XIX              11226
SNF_Days_Title_XVIII              132
SNF_Days_Total                     20
SNF_Discharges_Title_Other       1769
SNF_Discharges_Title_XIX        12732
SNF_Discharges_Title_XVIII        624
SNF_Discharges_Total              297
SNF_Number_of_beds                345
SNF_bed_Days_Available            342
Total_Bed_Days_Available          337
Total_Days_Other                  283
Total_Days_Title_XIX            10481
Total_Days_Title_XVIII            131
Total_Days_Total                   10
Total_Discharges_Title_Other      1655
Total_Discharges_Title_XIX      11963
Total_Discharges_Title_XVIII      624
Total_Discharges_Total            284
Years                               0
dtype: int64
```

```
#finding what percentage of values in our dataset were missing to give us better
# sense of scale this problem
total_cells = np.product(rentRollData.shape)
total_missing = missing_values_count.sum()

# percent of data that is missing
(total_missing/total_cells) * 100

# the output is  94% indicates  the data set has value
```

```
2.7612339486420607
```

```
#correlation matrix
correlation_matrix = df_numerical_data.corr()
```

```
correlation_matrix
```

	Years	Provider_CCN	Number_of_Beds	SNF_Admissions_Other	SNF_Admissions_Title_XIX	SNF_Admissions_Title_XVIII	SNF_Admissions_Total	SNF_Average_Length_of_stay_Tot	SNF_Average_Length_stay_XIX	SNF_Average_Length_stay_XVIII	...
Years	1.000000	-0.006244	-0.001742	0.006209	-0.015830	-0.095347	-0.046875	0.003572	0.003707	0.037398	...
Provider_CCN	-0.006244	1.000000	-0.003464	-0.011621	-0.016633	-0.044118	-0.031260	0.004283	-0.006128	0.026247	...
Number_of_Beds	-0.001742	-0.003464	1.000000	0.004587	0.000991	0.005461	0.005115	-0.000354	0.000076	-0.000910	...
SNF_Admissions_Other	0.006209	-0.011621	0.004587	1.000000	0.456837	0.492037	0.855859	-0.049892	-0.032574	-0.117108	...
SNF_Admissions_Title_XIX	-0.015830	-0.016633	0.000991	0.456837	1.000000	0.209378	0.696737	-0.013706	-0.036725	-0.011536	...
SNF_Admissions_Title_XVIII	-0.095347	-0.044118	0.005461	0.492037	0.209378	1.000000	0.759380	-0.053439	-0.028137	-0.130843	...
SNF_Admissions_Total	-0.046875	-0.031260	0.005115	0.855859	0.696737	0.759380	1.000000	-0.055176	-0.042446	-0.122053	...
SNF_Average_Length_of_stay_Tot	0.003572	0.004283	-0.000354	-0.049892	-0.013706	-0.053439	-0.055176	1.000000	0.917700	0.059428	...
SNF_Average_Length_stay_XIX	0.003707	-0.006128	0.000076	-0.032574	-0.036725	-0.028137	-0.042446	0.917700	1.000000	0.031015	...
SNF_Average_Length_stay_XVIII	0.037398	0.026247	-0.000910	-0.117108	-0.011536	-0.130843	-0.122053	0.059428	0.031015	1.000000	...
SNF_Days_Other	-0.046604	-0.016856	0.000573	0.336324	0.036935	0.222750	0.257355	0.002576	-0.005194	-0.024358	...
SNF_Days_Title_XIX	-0.021623	-0.019973	0.000509	0.074227	0.082512	0.079889	0.103146	0.903635	0.837365	0.006982	...
SNF_Days_Title_XVIII	-0.090713	-0.044012	0.006900	0.383943	0.081624	0.774031	0.584909	-0.050552	-0.015455	-0.064785	...
SNF_Days_Total	-0.029991	-0.021757	0.001493	0.183076	0.092621	0.179880	0.213008	0.832755	0.791056	-0.003563	...
SNF_Discharges_Title_Other	-0.005475	-0.011314	0.003075	0.497737	0.101812	0.314567	0.403220	-0.034123	-0.032607	-0.076753	...
SNF_Discharges_Title_XIX	-0.011925	-0.036884	-0.000744	0.231104	0.362230	0.210966	0.342541	-0.029342	-0.082248	-0.011709	...
SNF_Discharges_Title_XVIII	-0.008516	-0.004368	0.000490	0.030148	0.006738	0.055550	0.043830	-0.004530	-0.001371	-0.103350	...
SNF_Discharges_Total	-0.009194	-0.006514	0.000836	0.097791	0.027421	0.094052	0.102295	-0.010086	-0.006770	-0.083337	...
SNF_Number_of_beds	-0.001765	-0.003443	0.999982	0.004587	0.001013	0.005443	0.005130	-0.000350	0.000089	-0.000904	...
SNF_bed_Days_Available	-0.013142	-0.010467	0.257958	0.186041	0.094134	0.197686	0.216032	-0.002810	0.001060	-0.013075	...
Total_Bed_Days_Available	0.013702	-0.010442	0.264074	0.182840	0.090919	0.194023	0.210411	-0.003602	0.001817	-0.013600	...
Total_Days_Other	-0.031965	-0.003829	0.000477	0.226783	0.002764	0.163243	0.189085	-0.006110	0.003794	-0.034722	...
Total_Days_Title_XIX	-0.022993	-0.022530	0.000501	0.073782	0.082283	0.079083	0.102650	0.902660	0.836608	0.006900	...
Total_Days_Title_XVIII	-0.087560	-0.042711	0.006618	0.374957	0.078579	0.757594	0.571746	-0.049105	-0.014209	-0.064272	...
Total_Days_Total	-0.032251	-0.019993	0.001471	0.181276	0.084361	0.188080	0.208403	0.800066	0.776122	-0.011607	...
Total_Discharges_Title_Other	-0.000773	-0.011384	0.003068	0.496993	0.100698	0.314380	0.402541	-0.034345	-0.032607	-0.076852	...
Total_Discharges_Title_XIX	-0.013842	-0.039760	-0.000747	0.226604	0.360876	0.206202	0.337558	-0.029269	-0.082465	-0.011084	...
Total_Discharges_Title_XVIII	-0.008520	-0.004372	0.000490	0.030148	0.006737	0.055555	0.043832	-0.004530	-0.001370	-0.103360	...
Total_Discharges_Total	-0.009444	-0.007273	0.000828	0.097135	0.027216	0.097524	0.103339	-0.010266	-0.006891	-0.077074	...

29 rows × 29 columns

[Insights]

Penalties

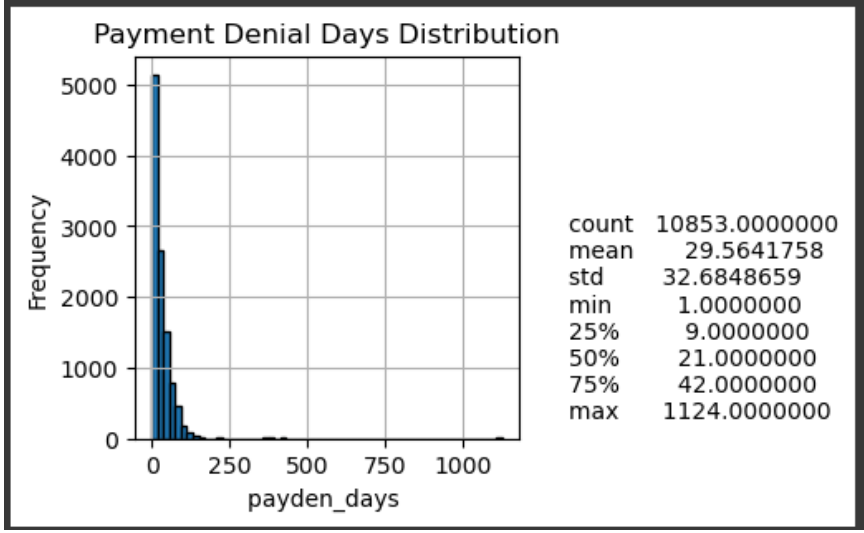
```
r=P_df[['YearOfFileDate','fine_amt','payden_days',]]
r.corr()
```

	YearOfFileDate	fine_amt	payden_days
YearOfFileDate	1.000000	0.055532	-0.042135
fine_amt	0.055532	1.000000	NaN
payden_days	-0.042135	NaN	1.000000

```
#penality_df.describe()
df_Penalities_2015to2021.describe()
```

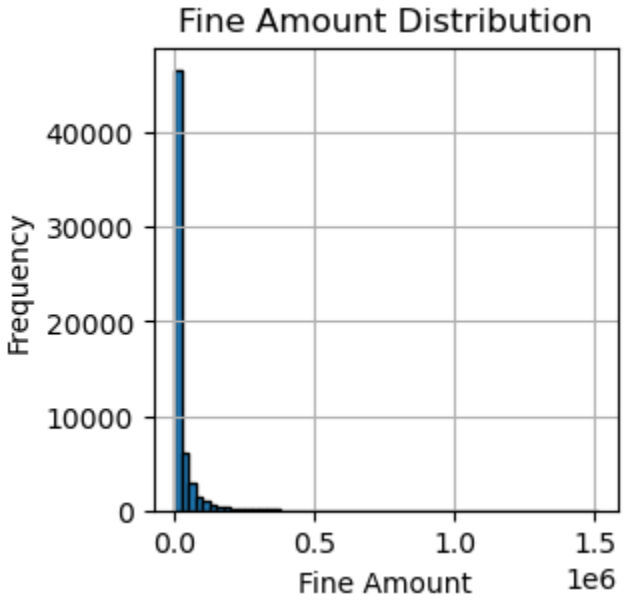
	fine_amt	payden_days	YearOfFileDate
count	6.063200e+04	10856.000000	71488.000000
mean	2.629427e+04	29.410648	2018.840966
std	5.734782e+04	33.960032	2.094308
min	9.800000e+01	-526.000000	2015.000000
25%	3.250000e+03	9.000000	2017.000000
50%	9.750000e+03	21.000000	2019.000000
75%	2.252000e+04	42.000000	2021.000000
max	1.508727e+06	1124.000000	2021.000000

```python
# Checking the distribution of fine amount and payden_days
pd.options.display.float_format = '{:.7f}'.format
def desc_num_feature_payden_days_beforechange(feature_name, bins=60, edgecolor='k', **kwargs):
    fig, ax = plt.subplots(figsize=(3,3))
    df_Penalities_2015to2021['payden_days'].hist(bins=bins, edgecolor=edgecolor, ax=ax,**kwargs)
    ax.set_title("Payment Denial Days Distribution", size=12)
    plt.xlabel('payden_days ')
    plt.ylabel('Frequency')
    desc_text=df_Penalities_2015to2021['payden_days'].describe().to_string()
    plt.figtext(1,0.15, desc_text,size=10)

desc_num_feature_payden_days_beforechange(df_Penalities_2015to2021['payden_days'])
```



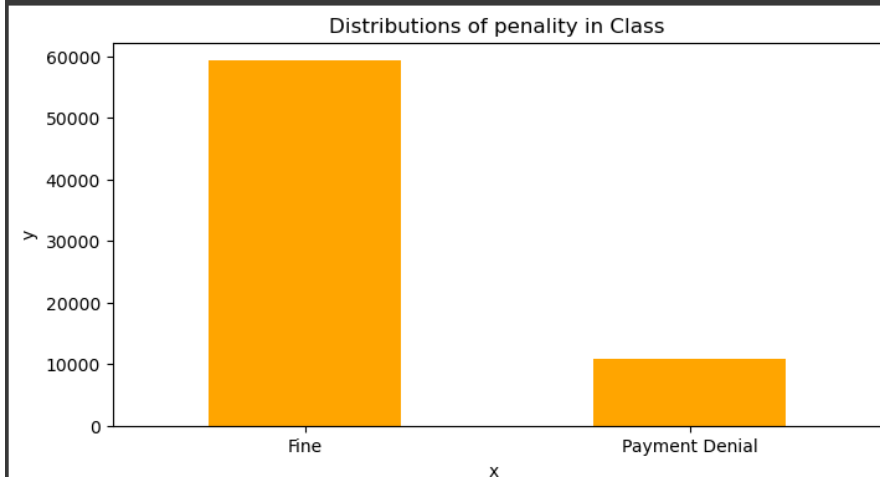

```
count   10853.0000000
mean       29.5641758
std        32.6848659
min         1.0000000
25%         9.0000000
50%        21.0000000
75%        42.0000000
max      1124.0000000
```

```python
# Checking the distribution of fine amount
pd.options.display.float_format = '{:.7f}'.format
def desc_num_feature_fineamt_beforechange(feature_name, bins=60, edgecolor='k', **kwargs):
    fig, ax = plt.subplots(figsize=(3,3))
    df_Penalities_2015to2021['fine_amt'].hist(bins=bins, edgecolor=edgecolor, ax=ax,**kwargs)
    ax.set_title("Fine Amount Distribution", size=12)
    #plt.xticks([0.0, 0.2,0.4,0.6,0.8,1.0,1.2,1.4], ['0k','2k','4k','6k','8k' ,'10k','12k','14k'])  # Replace 0.0 with 0 and 1.4 with 14000
    plt.xlabel('Fine Amount')
    plt.ylabel('Frequency')
    desc_text=df_Penalities_2015to2021['fine_amt'].describe().to_string()
    plt.figtext(1,0.15, desc_text,size=10)

desc_num_feature_fineamt_beforechange(df_Penalities_2015to2021['fine_amt'])
```



count	60632.0000000
mean	26294.2651735
std	57347.8238215
min	98.0000000
25%	3250.0000000
50%	9750.0000000
75%	22520.0000000
max	1508727.0000000

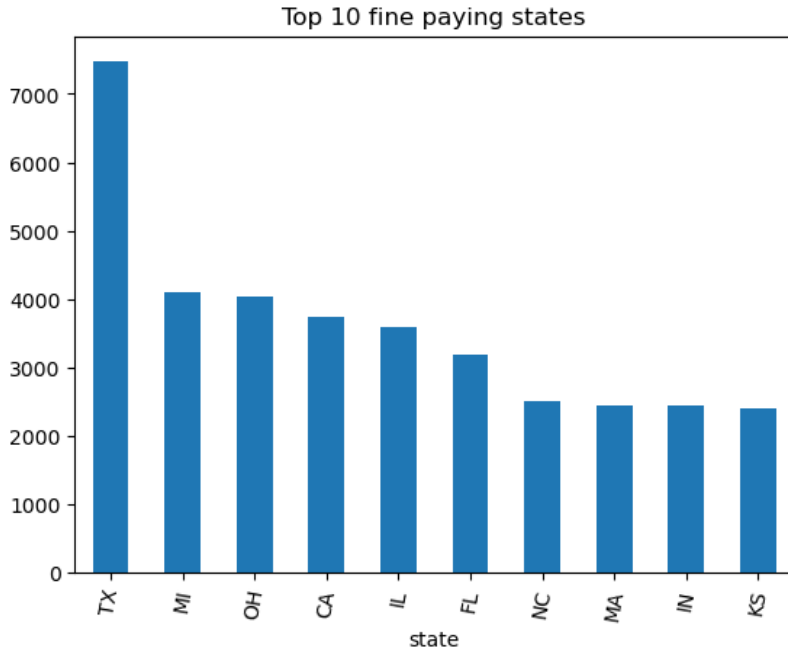
```
# visulaize the value_counts results using bar chart
PenaltyType=df_Penalities_2015to2021['pnlty_type'].value_counts() # storing the value
plt.figure(figsize=(8,4))
PenaltyType.plot(kind='bar',color='orange')
plt.title('Distributions of penality in Class')

plt.xlabel('x ')
plt.ylabel('y')
plt.xticks(rotation=0)

plt.show()
```




```
# Create the bar chart
plt.figure(figsize=(12,6))
# countplot each label and have the value quantity of it in the label, using hue will distrbute among gender
sns.countplot(data=df_Penalities_2015to2021,x='state', hue='pnlty_type',
              palette={'Fine':'green', 'Payment Denial':'red'})
plt.xticks(rotation=60)
plt.show()
```

O

```
# top fine paying states
df_Penalities_2015to2021['state'].value_counts().head(10).plot(kind='bar', rot=80,title='Top 10 fine paying states')
plt.show()
```

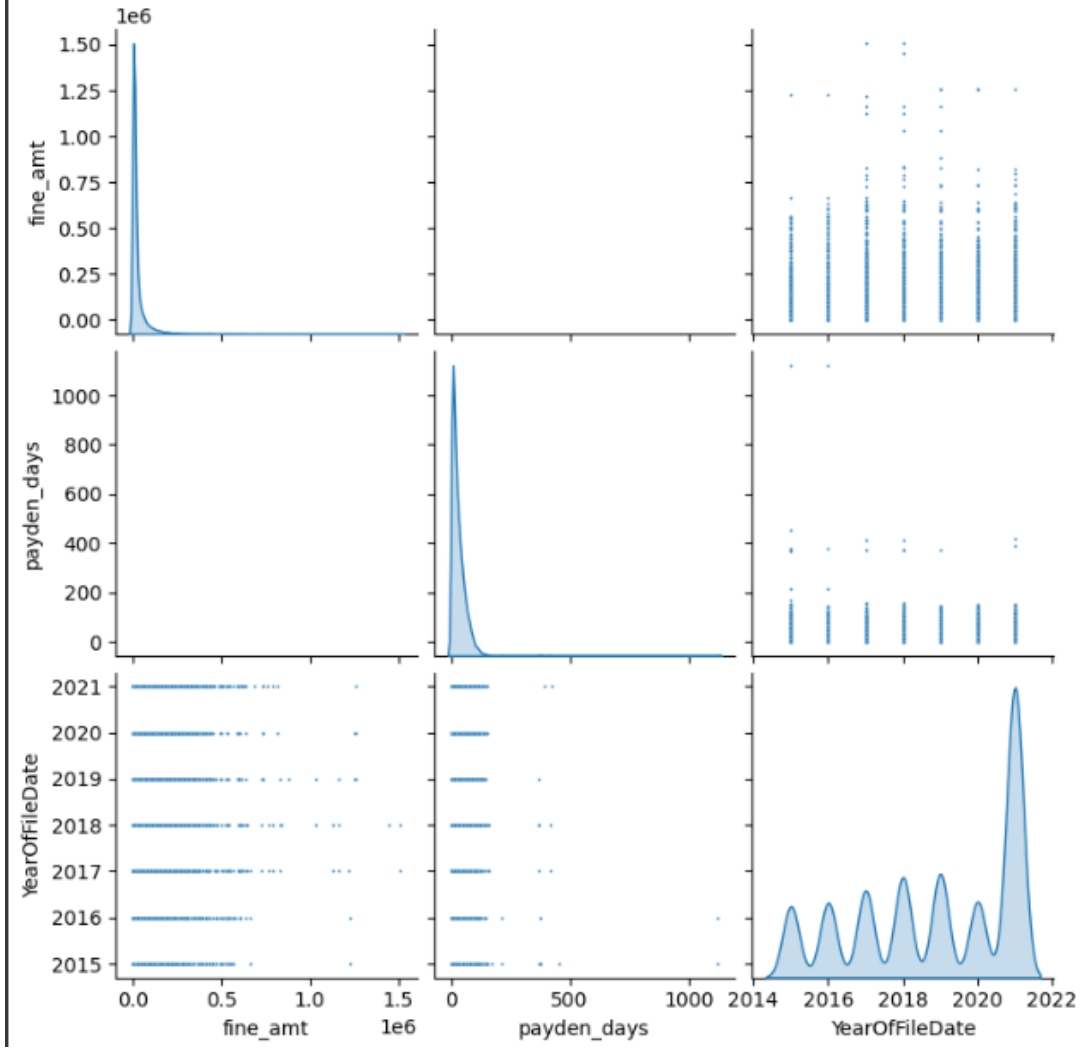



```
df_Penalities_2015to2021['fine_amt'].describe()
```

```
count     59222.0000000
mean      26810.8507480
std       57924.4636965
min          98.0000000
25%        3250.0000000
50%        9750.0000000
75%       23205.0000000
max     1508727.0000000
Name: fine_amt, dtype: float64
```

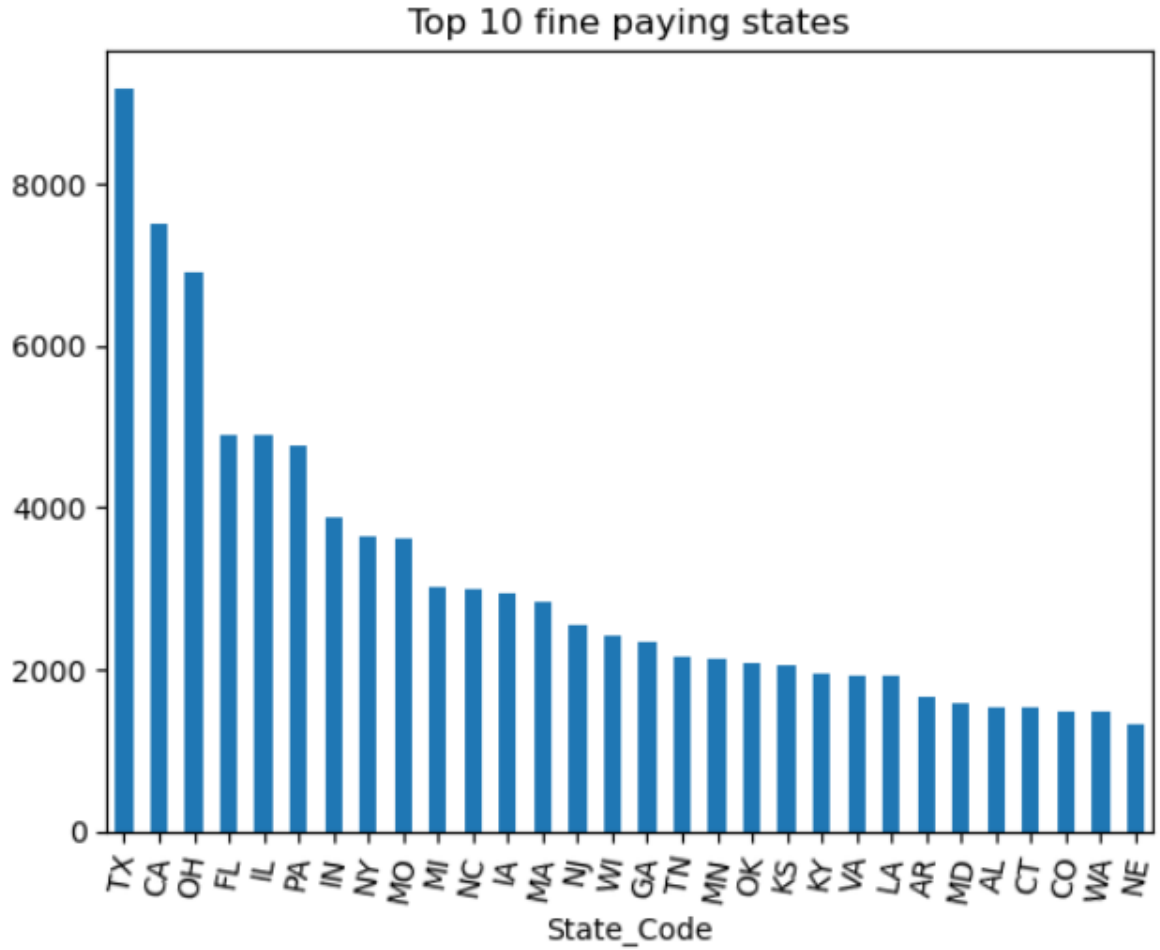
```
sns.pairplot(df_Penalities_2015to2021, plot_kws={"s": 2},diag_kind='kde')
```

```
c:\Users\kanak\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout
  self._figure.tight_layout(*args, **kwargs)
<seaborn.axisgrid.PairGrid at 0x16992373910>
```

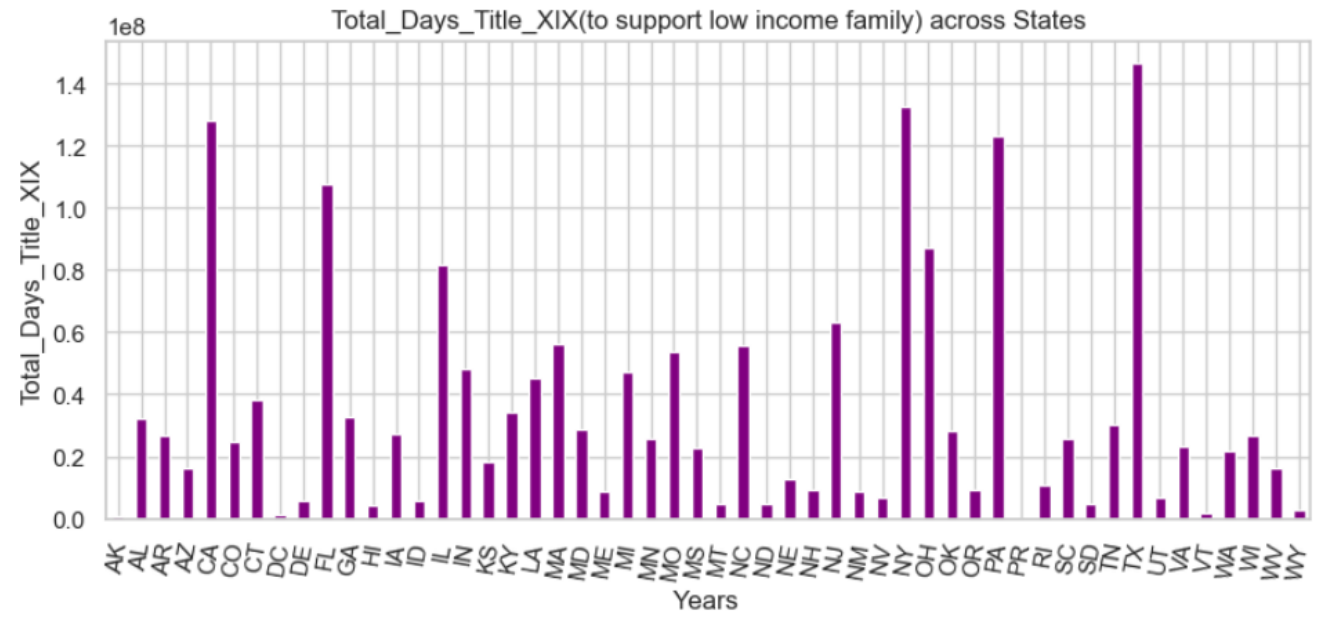


Rent Roll

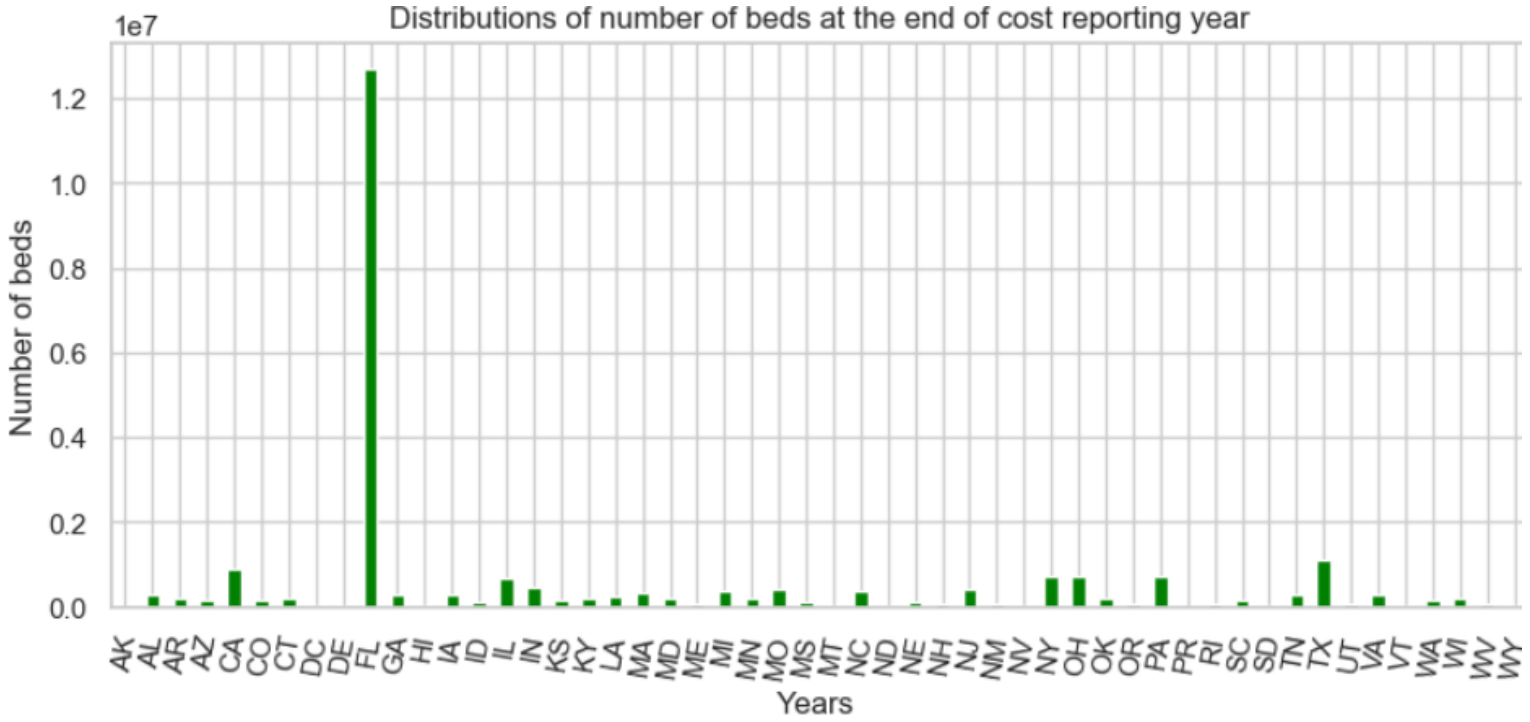
```
# top fine paying states
rentRollData['State_Code'].value_counts().head(30).plot(kind='bar', rot=80,title='Top 10 fine paying states')
plt.show()
```



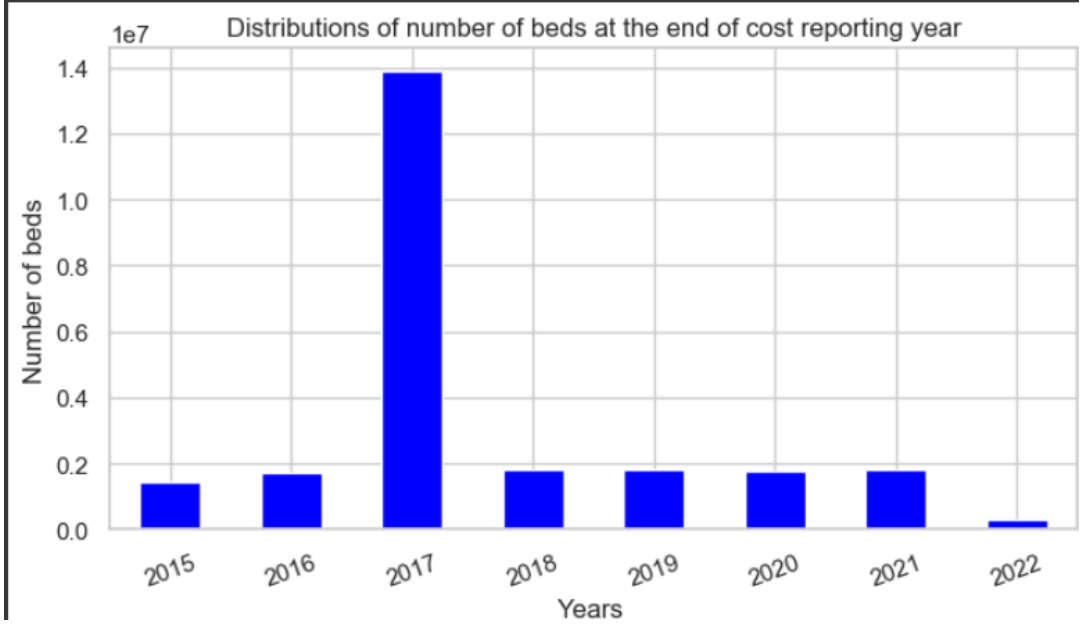
```
TotalDays_title19 = rentRollData.groupby('State_Code')['Total_Days_Title_XIX'].sum()
# indicates that there are low income familes are more in Texas

sns.set_theme(style="whitegrid")

# Create the chart

plt.figure(figsize=(10,4))
TotalDays_title19.plot(kind='bar',color='purple')
plt.title('Total_Days_Title_XIX(to support low income family) across States')

plt.xlabel('Years ')
plt.ylabel('Total_Days_Title_XIX')
plt.xticks(rotation=80)
plt.show()
```



```
plt.figure(figsize=(10,4))
beds_by_state.plot(kind='bar',color='green')
plt.title('Distributions of number of beds at the end of cost reporting year')

plt.xlabel('Years ')
plt.ylabel('Number of beds')
plt.xticks(rotation=80)
plt.show()
```

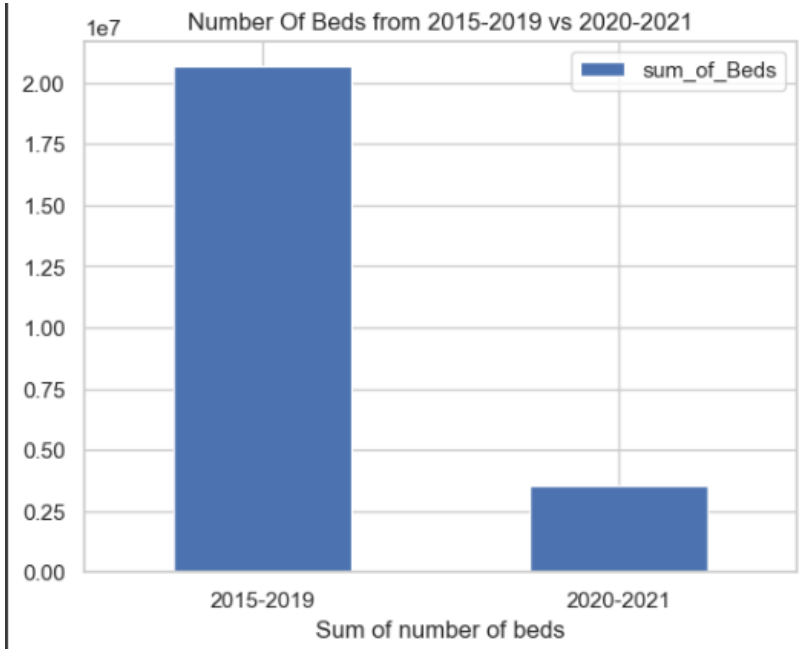


```
plt.figure(figsize=(8,4))
beds_by_year.plot(kind='bar',color='blue')
plt.title('Distributions of number of beds at the end of cost reporting year')

plt.xlabel('Years ')
plt.ylabel('Number of beds')
plt.xticks(rotation=20)
plt.show()
```



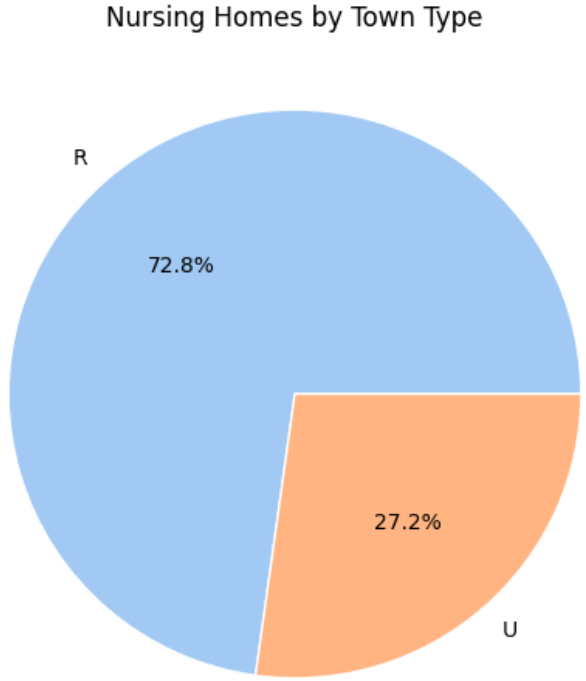
```
# splitting data in pre and post covid

# Create two separate DataFrames for years 2015 and 2016, and the remaining years
per_Covid = rentRollData[rentRollData['Years'].isin([2015, 2016, 2017,2018,2019])]
post_Covid = rentRollData[rentRollData['Years'].isin([2020, 2021])]
```

```
# Calculate the sum  for DataFrame 1 and DataFrame 2
sum_df1 = per_Covid['Number_of_Beds'].sum()
sum_df2 = post_Covid['Number_of_Beds'].sum()

# Create a new DataFrame to hold the sum of fine amounts
sum_of_Beds = pd.DataFrame({'TimeFrame_2015-2021': ['2015-2019', '2020-2021'],
                            'sum_of_Beds': [sum_df1, sum_df2]})

# Plot the sum of fine amounts
sum_of_Beds.plot(kind='bar', x='TimeFrame_2015-2021', y='sum_of_Beds', rot=0, title='Number Of Beds from 2015-2019 vs 2020-2021')
plt.xlabel('Sum of number of beds')
plt.ylabel('')
plt.show()
```



Property

```
types = ['R', 'U']
townTypes = propertyDataClean['Rural_versus_Urban'].value_counts()

#creating a pie chart
plt.figure(figsize=(6,6))
plt.pie(townTypes, labels=types, autopct='%1.1f%%', colors=sns.color_palette('pastel'), wedgeprops={'edgecolor':'white'})
plt.title('Nursing Homes by Town Type')

plt.show()
```

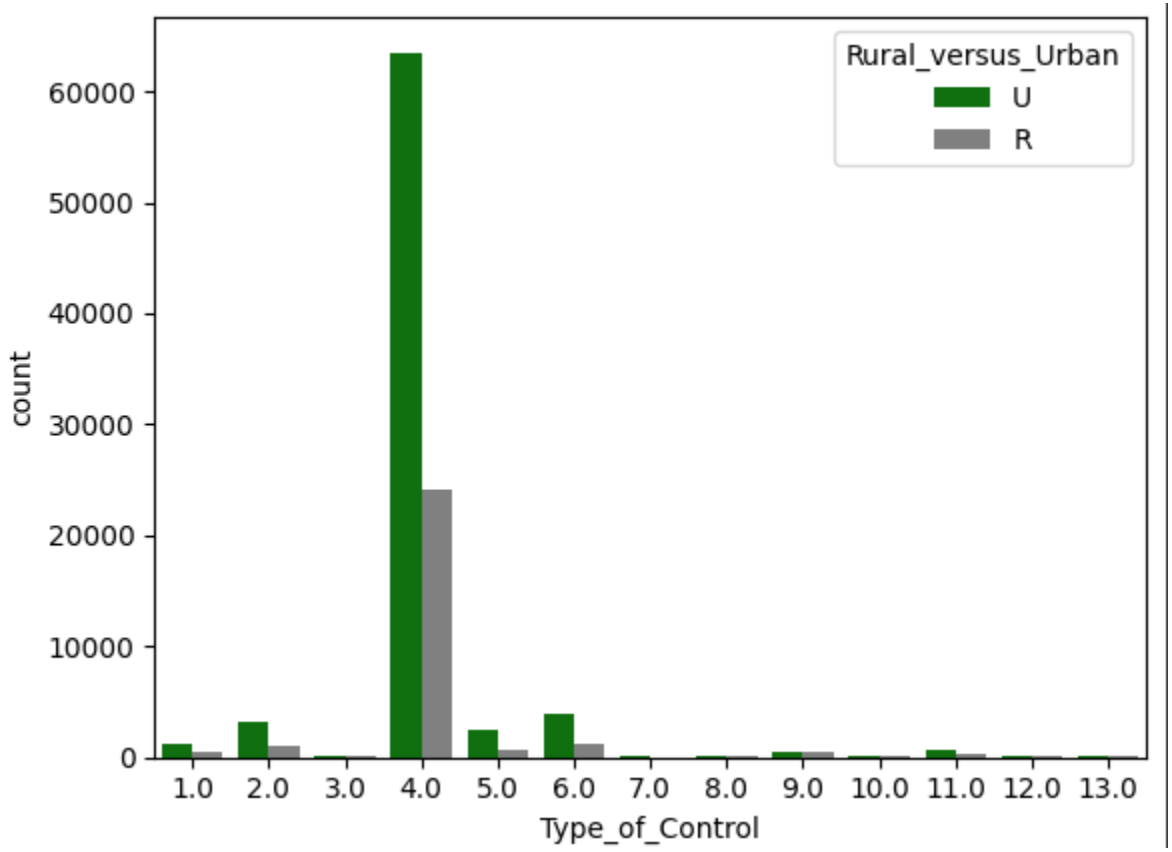


72.8 % of Nursing Home Facilities are located in Rural areas while 27 are located in Urban areas.

```python
propertyDataClean['Type_of_Control'].value_counts(normalize=True)
# 84% of nursing homes are category: 4- Proprietary-Corporation
# 4.7% of nursing homes are category: 6- Proprietary-Other
# 4% of nursing homes are category: 2- Volentary/Non-Profit Other
# 2.9% of nursing homes are category: 5- Proprietary Partnership
# 1.5% of nursing homes are category: 1- Volentary/Non-Profit Church
```

```
Type_of_Control
4.0     0.841770
6.0     0.047825
2.0     0.040009
5.0     0.029656
1.0     0.015313
9.0     0.009296
11.0    0.007489
13.0    0.002269
8.0     0.002086
10.0    0.001730
3.0     0.001432
12.0    0.000827
7.0     0.000298
Name: proportion, dtype: float64
```

```
sns.countplot(data=propertyDataClean, x='Type_of_Control', hue='Rural_versus_Urban',
palette={'R':'grey','U':'green'})
plt.show()
#the majority of Nuring Home Corporations are located in Urban Areas
```



Majority of Nursing Home Facilities in Urban and Rural areas are Corporations.