In [1]:
```
# python packages are imported
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# ... csv file are converted  into pandas.....
dim_date = pd.read_csv("dim_date.csv")
dim_districts = pd.read_csv("dim_districts.csv")
fact_stamps = pd.read_csv("fact_stamps.csv")
fact_transport = pd.read_csv("fact_transport.csv")
fact_ts_ipass = pd.read_csv("fact_TS_iPASS.csv")
```

In [3]:
```
# ... All the data set file are  printed....
print(dim_date.columns)
print(dim_districts.columns)
print(fact_stamps.columns)
print(fact_transport.columns)
print(fact_ts_ipass.columns)
```

```
Index(['month', 'Mmm', 'quarter', 'fiscal_year'], dtype='object')
Index(['dist_code', 'district'], dtype='object')
Index(['dist_code', 'month', 'documents_registered_cnt',
       'documents_registered_rev', 'estamps_challans_cnt',
       'estamps_challans_rev'],
      dtype='object')
Index(['dist_code', 'month', 'fuel_type_petrol', 'fuel_type_diesel',
       'fuel_type_electric', 'fuel_type_others', 'vehicleClass_MotorCycle',
       'vehicleClass_MotorCar', 'vehicleClass_AutoRickshaw',
       'vehicleClass_Agriculture', 'vehicleClass_others',
       'seatCapacity_1_to_3', 'seatCapacity_4_to_6', 'seatCapacity_above_6',
       'Brand_new_vehicles', 'Pre-owned_vehicles', 'category_Non-Transport',
       'category_Transport'],
      dtype='object')
Index(['dist_code', 'month', 'sector', 'investment in cr',
       'number_of_employees'],
      dtype='object')
```

In [47]:
```
# merged dim_district and Fact_stamps data on dist_code as primary key and Outer join is used.....
dataset_1= dim_districts.merge(fact_stamps, on = "dist_code", how = "outer")
# to know any null values
dataset_1.isna()
dataset_1.isna().any()
```

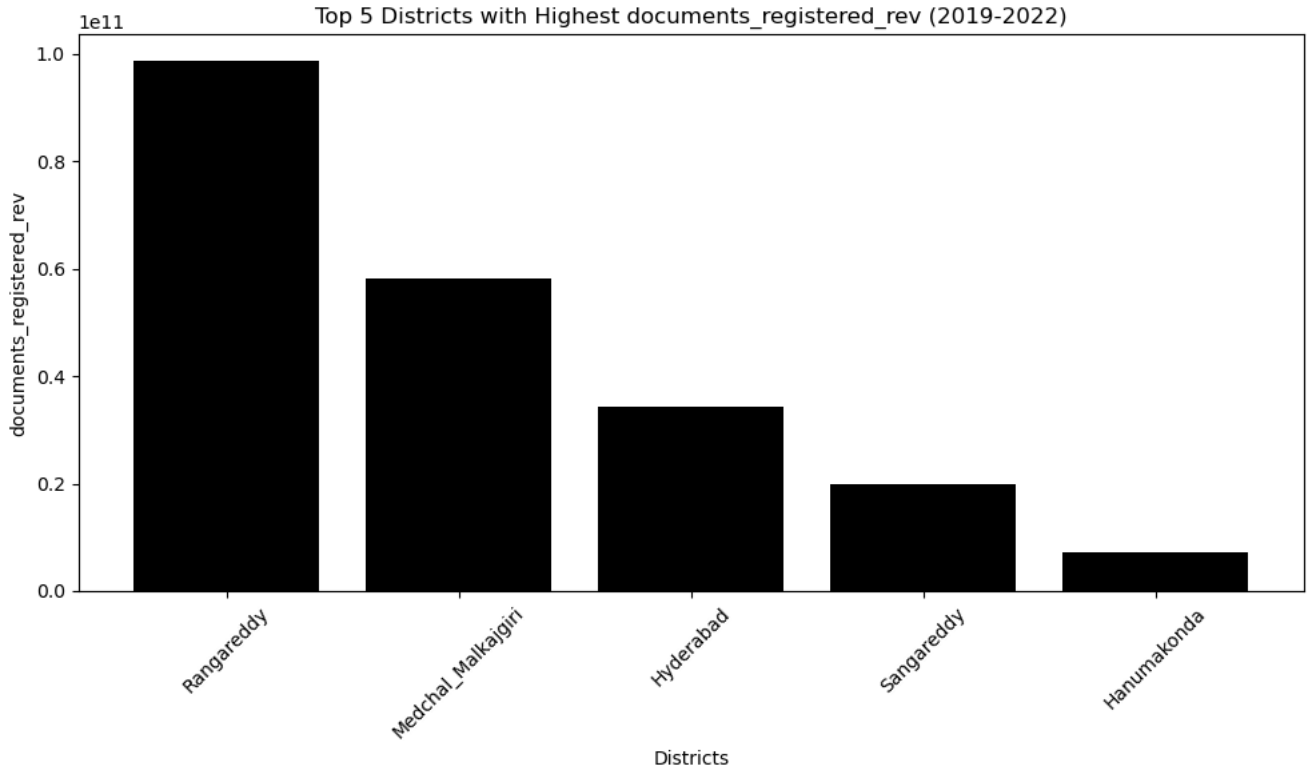
Out[47]:
```
dist_code                 False
district                  False
month                      True
documents_registered_cnt   True
documents_registered_rev   True
estamps_challans_cnt       True
estamps_challans_rev       True
dtype: bool
```

In [48]:
```
'''How does the revenue generated from document registration vary across districts in Telangana?
List down the top 5 districts that showed the highest document registration revenue growth between FY 2019 and 2022.'''
```

Out[48]:
```
'How does the revenue generated from document registration vary across districts in Telangana? \nList down the top 5
districts that showed the highest document registration revenue growth between FY 2019 and 2022.'
```

In [8]:
```
# Convert the 'month' column to a datetime format
dataset_1['month'] = pd.to_datetime(dataset_1['month'])
# Filter the DataFrame for data between 2019 and 2022
data_years_2019_to_2022 = dataset_1[(dataset_1['month'].dt.year >= 2019) & (dataset_1['month'].dt.year <= 2022)]
# Select specific columns
Dataset_q1 = data_years_2019_to_2022[['district', 'documents_registered_rev']]
# Group by 'district' and sum the 'documents_registered_rev'
result1_1_df = Dataset_q1.groupby('district')['documents_registered_rev'].sum().reset_index()
# Sort the DataFrame by 'documents_registered_rev' in descending order and take the top 5 districts
result1_1_df = result1_1_df.sort_values('documents_registered_rev', ascending=False).head(5)
print("The top 5 districts that showed the highest document registration revenue growth between FY 2019 and 2022.")
print(result1_1_df)
# Create a bar plot
plt.figure(figsize=(10, 6))
plt.bar(result1_1_df['district'], result1_1_df['documents_registered_rev'], color='black')
plt.title('Top 5 Districts with Highest documents_registered_rev (2019-2022)')
plt.xlabel('Districts')
plt.ylabel('documents_registered_rev')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

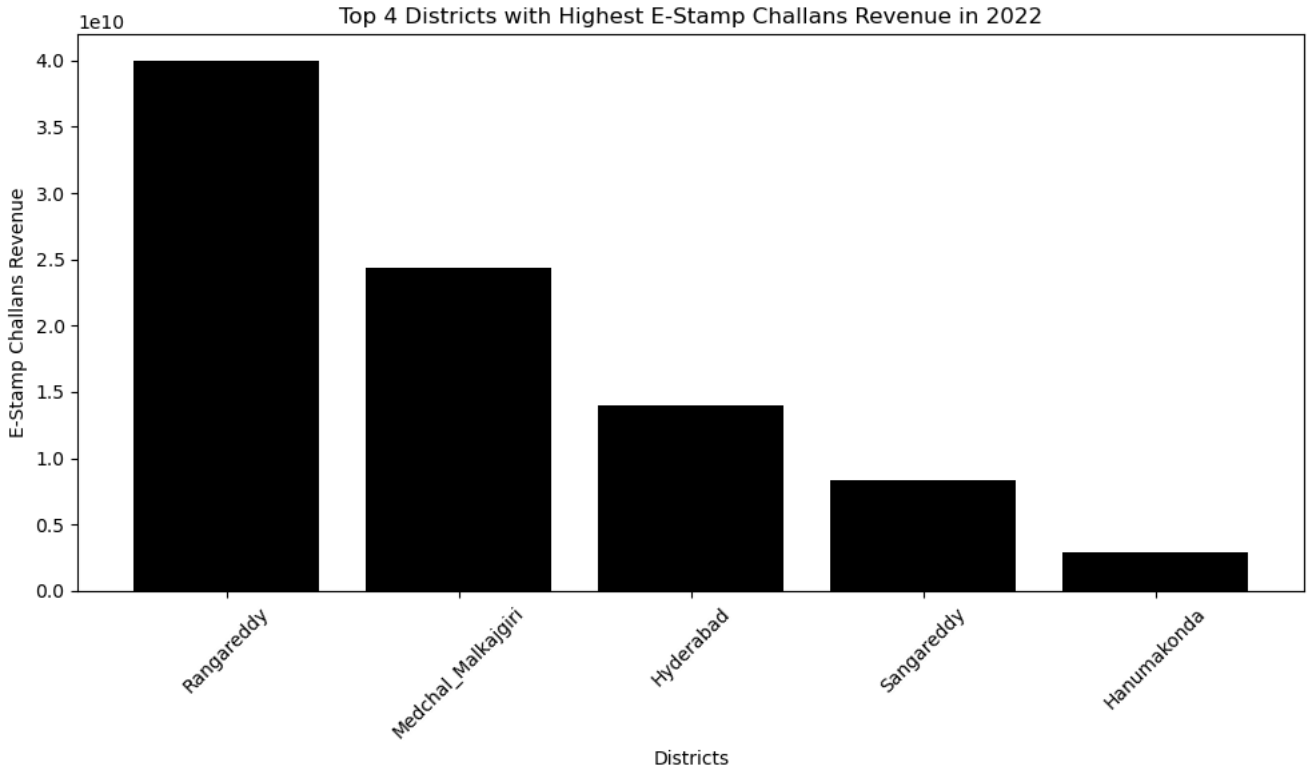
```
The top 5 districts that showed the highest document registration revenue growth between FY 2019 and 2022.
              district  documents_registered_rev
24          Rangareddy              9.868082e+10
15  Medchal_Malkajgiri              5.823680e+10
3             Hyderabad              3.436081e+10
25           Sangareddy              1.986397e+10
2           Hanumakonda              7.259725e+09
```



In [9]:
```
#Question2
#How does the revenue generated from document registration compare to the revenue generated
#from e-stamp challans across districts? List down the top 5 districts where e-stamps revenue contributes
#significantly more to the revenue than the documents in FY 2022?
```

In [10]:
```
#the top 5 districts where e-stamps revenue contributes significantly more to the revenue than the documents in FY 2022
# Filter the DataFrame for data from the year 2022
data_2022 = dataset_1[dataset_1['month'].dt.year == 2022]
# Select specific columns and create a DataFrame
Dataset_q2 = data_2022[['district', 'estamps_challans_rev']]
# Group by 'district' and sum the 'estamps_challans_rev'
result1_2_df = Dataset_q2.groupby('district')['estamps_challans_rev'].sum().reset_index()
# Sort the DataFrame by 'estamps_challans_rev' in descending order and take the top 4 districts
result1_2_df = result1_2_df.sort_values('estamps_challans_rev', ascending=False).head(5)
print("The top 5 districts where e-stamps revenue contributes significantly more to the revenue than the documents in F
print(result1_2_df.head(4))
# Create a bar plot
plt.figure(figsize=(10, 6))
plt.bar(result1_2_df['district'], result1_2_df['estamps_challans_rev'], color='black')
plt.title('Top 4 Districts with Highest E-Stamp Challans Revenue in 2022')
plt.xlabel('Districts')
plt.ylabel('E-Stamp Challans Revenue')
plt.xticks(rotation=45)
# Show the plot
plt.tight_layout()
plt.show()
```

```
The top 5 districts where e-stamps revenue contributes significantly more to the revenue than the documents in FY 202
2?
              district  estamps_challans_rev
24          Rangareddy          3.995544e+10
15  Medchal_Malkajgiri          2.439412e+10
3             Hyderabad          1.395908e+10
25          Sangareddy          8.371738e+09
```



In [11]:
```
#Question_4
#Categorize districts into three segments based on their stamp registration revenue generation during
#the fiscal year 2021 to 2022.
```

In [12]:
```
#Categorize districts into three segments based on their stamp registration revenue generation
#during the fiscal year 2021 to 2022.
# Assuming you have a DataFrame named dataset_1
# Convert the 'month' column to a datetime format
dataset_1['month'] = pd.to_datetime(dataset_1['month'])

# Filter the DataFrame for data between 2019 and 2022
data_years_2021_to_2022 = dataset_1[(dataset_1['month'].dt.year >= 2021) & (dataset_1['month'].dt.year <= 2022)]

# Select specific columns
Dataset_q4 = data_years_2021_to_2022[['district', 'estamps_challans_rev']]

# Group by 'district' and sum the 'documents_registered_rev'
result1_4_df = Dataset_q4.groupby('district')['estamps_challans_rev'].sum().reset_index()
result1_4_df_11 = result1_4_df.sort_values('estamps_challans_rev', ascending=False)
category_A = result1_4_df_11.iloc[0:4]
category_B = result1_4_df_11.iloc[4:17]
category_C = result1_4_df_11.iloc[17:]
print(category_A)
print(category_B)
print(category_C)
```

```
             district  estamps_challans_rev
24          Rangareddy          6.962464e+10
15  Medchal_Malkajgiri          4.168092e+10
3             Hyderabad          2.390250e+10
25          Sangareddy          1.432089e+10
                district  estamps_challans_rev
2           Hanumakonda          4.930453e+09
9               Khammam          3.652373e+09
31  Yadadri Bhuvanagiri          3.520680e+09
21            Nizamabad          2.619173e+09
18              Nalgonda          2.615991e+09
8             Karimnagar          2.543394e+09
12         Mahabubnagar          1.956398e+09
27              Suryapet          1.792893e+09
26              Siddipet          1.637765e+09
22            Peddapalli          1.105364e+09
14                 Medak          1.073720e+09
13            Mancherial          1.065763e+09
4                Jagtial          1.015855e+09
                 district  estamps_challans_rev
28               Vikarabad           861877771.0
17            Nagarkurnool           849387306.0
7                Kamareddy           842152448.0
5                  Jangoan           706701596.0
23       Rajanna Sircilla           696745057.0
29             Wanaparthy           696178925.0
0                Adilabad           693556767.0
11             Mahabubabad           676293913.0
6         Jogulamba Gadwal           613831979.0
20                 Nirmal           607442237.0
1    Bhadradri Kothagudem           532548442.0
19             Narayanpet           452592565.0
30               Warangal           415047454.0
16                 Mulugu           352409622.0
10  Kumurambheem Asifabad           124753459.0
```

In [49]:
```
#Preparing for the Dataset_2
dataset_2= dim_districts.merge(fact_transport, on = "dist_code", how = "outer")
dataset_2.isna()
dataset_2.isna().any()
```

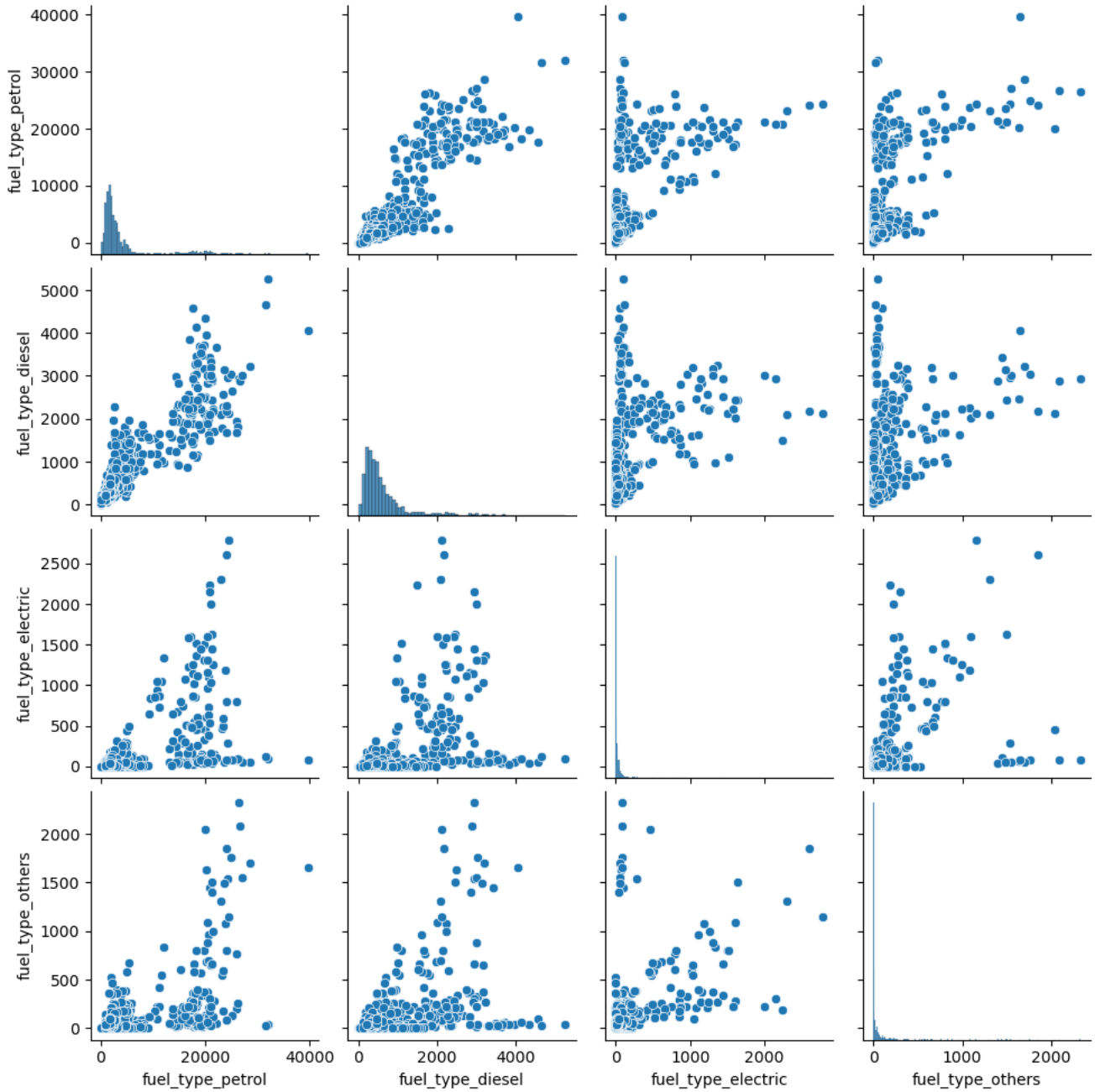
Out[49]:
```
dist_code                   False
district                    False
month                        True
fuel_type_petrol             True
fuel_type_diesel             True
fuel_type_electric           True
fuel_type_others             True
vehicleClass_MotorCycle      True
vehicleClass_MotorCar        True
vehicleClass_AutoRickshaw    True
vehicleClass_Agriculture     True
vehicleClass_others          True
seatCapacity_1_to_3          True
seatCapacity_4_to_6          True
seatCapacity_above_6         True
Brand_new_vehicles           True
Pre-owned_vehicles           True
category_Non-Transport       True
category_Transport           True
dtype: bool
```

In [52]:
```
# Question5
#Investigate whether there is any correlation between vehicle sales and specific months or seasons
#in different districts. Are there any months or seasons that consistently show higher or lower sales rate,
#and if yes, what could be the driving factors? (Consider Fuel-Type category only)
dataset_2['month'] = pd.to_datetime(dataset_2['month'])
dataset_2 = dataset_2[(dataset_2['month'].dt.year >= 2019)&(dataset_2['month'].dt.year <= 2023)]
Dataset2_q1 = dataset_2[['district', 'fuel_type_petrol','fuel_type_diesel','fuel_type_electric','fuel_type_others','mon
```

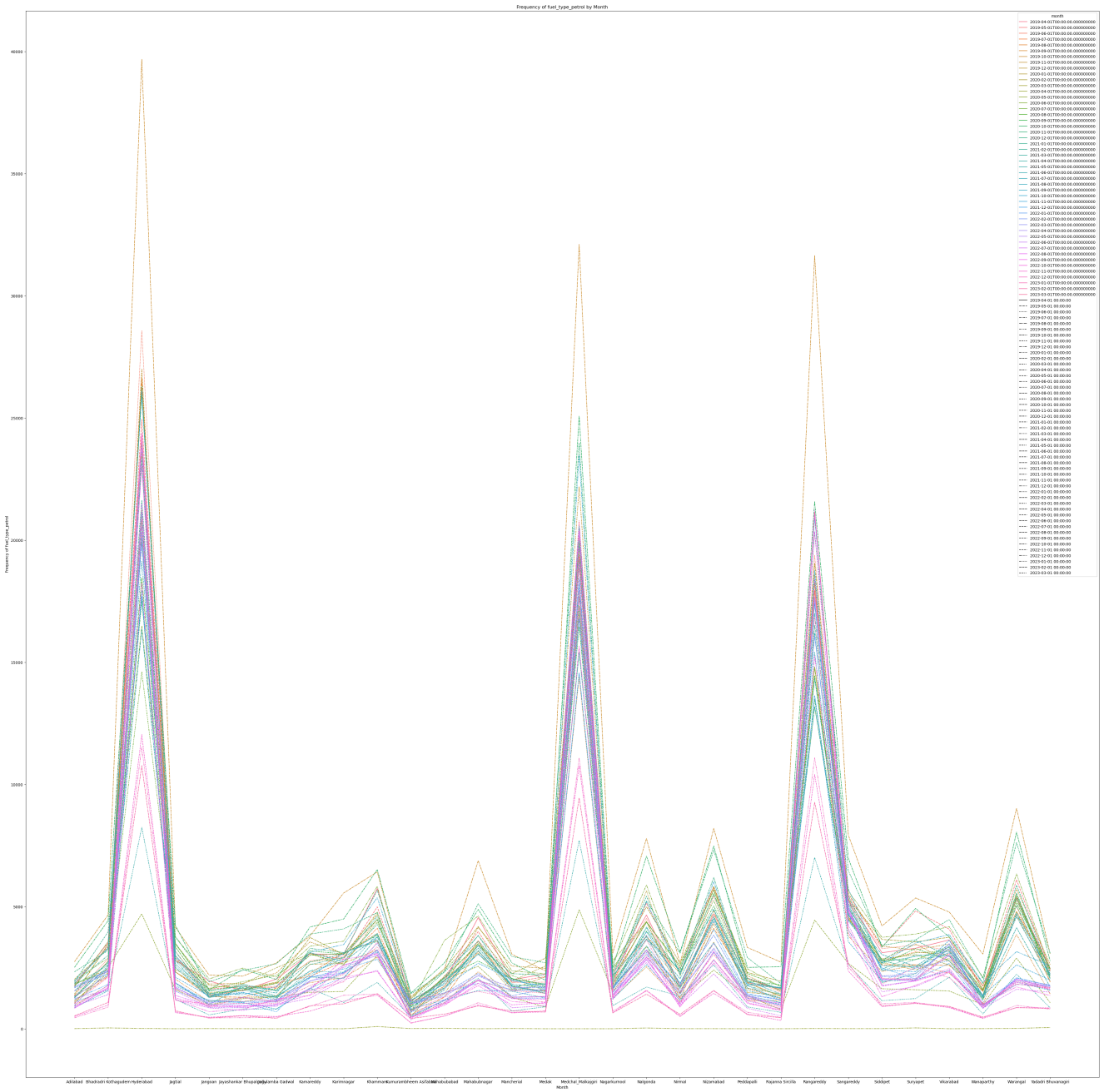
In [56]:
```
# Assuming Dataset2_q1 is a DataFrame with columns 'district', 'month', and fuel type columns
# Grouping by 'district' and 'month' and aggregating the fuel types
Dataset2_q1= Dataset2_q1.groupby(['district', 'month'])[['fuel_type_petrol', 'fuel_type_diesel', 'fuel_type_electric',

# Sorting the resulting DataFrame by index
Dataset2_q1 = Dataset2_q1.sort_index()

# Creating a pairplot
sns.pairplot(data=Dataset2_q1, vars=["fuel_type_petrol", "fuel_type_diesel", "fuel_type_electric", "fuel_type_others"])

# Printing a subset of the grouped DataFrame
Dataset2_q1.reset_index()[["district", "month", "fuel_type_petrol"]]
corr_matrix= Dataset2_q1.corr()
corr_matrix
```

Out[56]:

	fuel_type_petrol	fuel_type_diesel	fuel_type_electric	fuel_type_others
fuel_type_petrol	1.000000	0.892273	0.598413	0.633747
fuel_type_diesel	0.892273	1.000000	0.471648	0.495163
fuel_type_electric	0.598413	0.471648	1.000000	0.530573
fuel_type_others	0.633747	0.495163	0.530573	1.000000

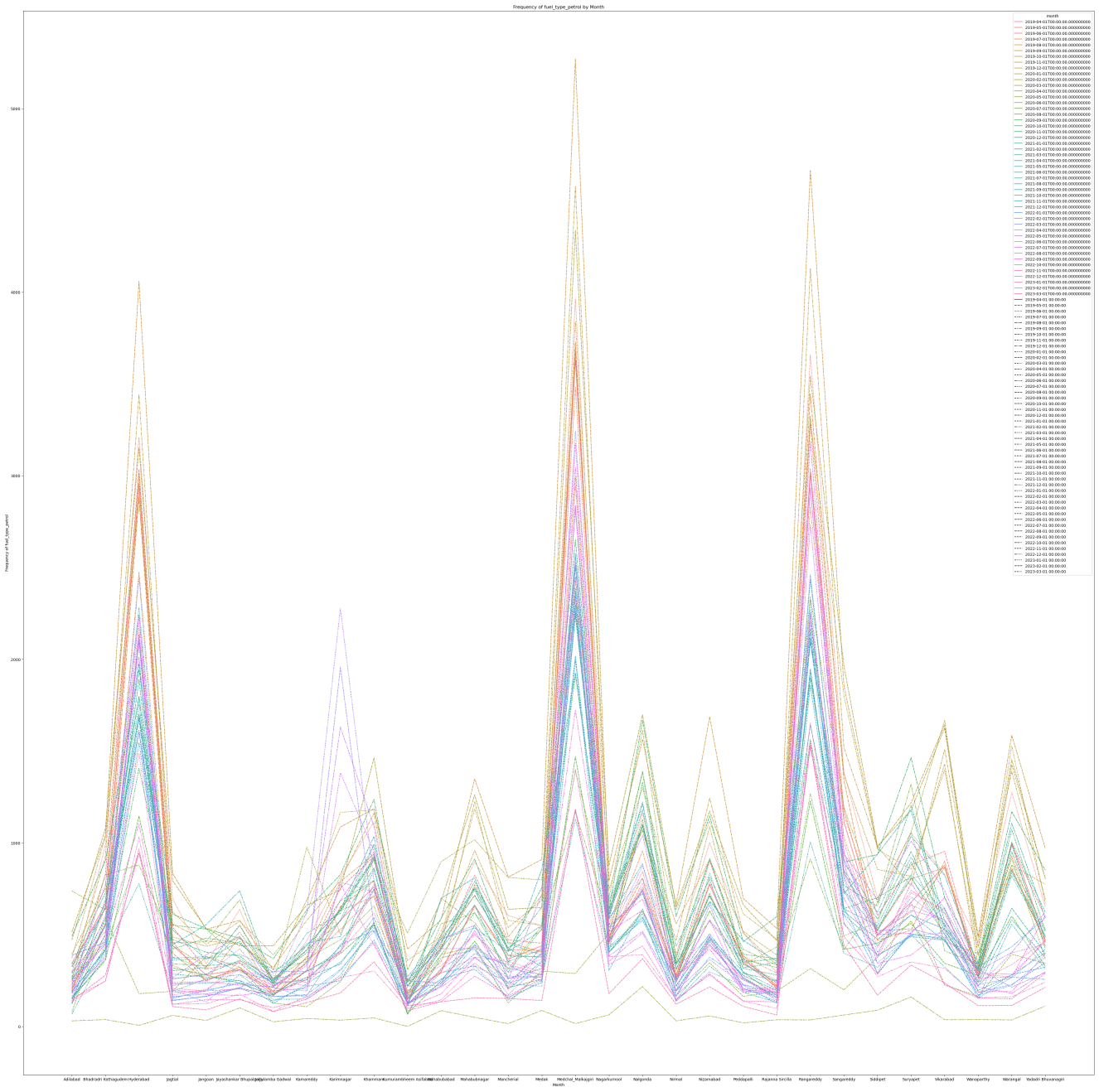


In [57]:
```
# Create a pivot table to count occurrences of 'fuel_type_petrol' for each combination of 'district' and 'month'
Dataset2_q1_pivot_table_1 = Dataset2_q1.pivot_table(index='district', columns='month', values='fuel_type_petrol')
# Plotting with Seaborn  # Optional: Set the figure size
plt.figure(figsize=(50,50))
sns.lineplot(data=Dataset2_q1_pivot_table_1)  # Transpose the pivot table for proper plotting
# Labeling axes and adding a title
plt.xlabel('Month')
plt.ylabel('Frequency of fuel_type_petrol')
plt.title('Frequency of fuel_type_petrol by Month')
# Show the plot
plt.show()
# Display the pivot table
```

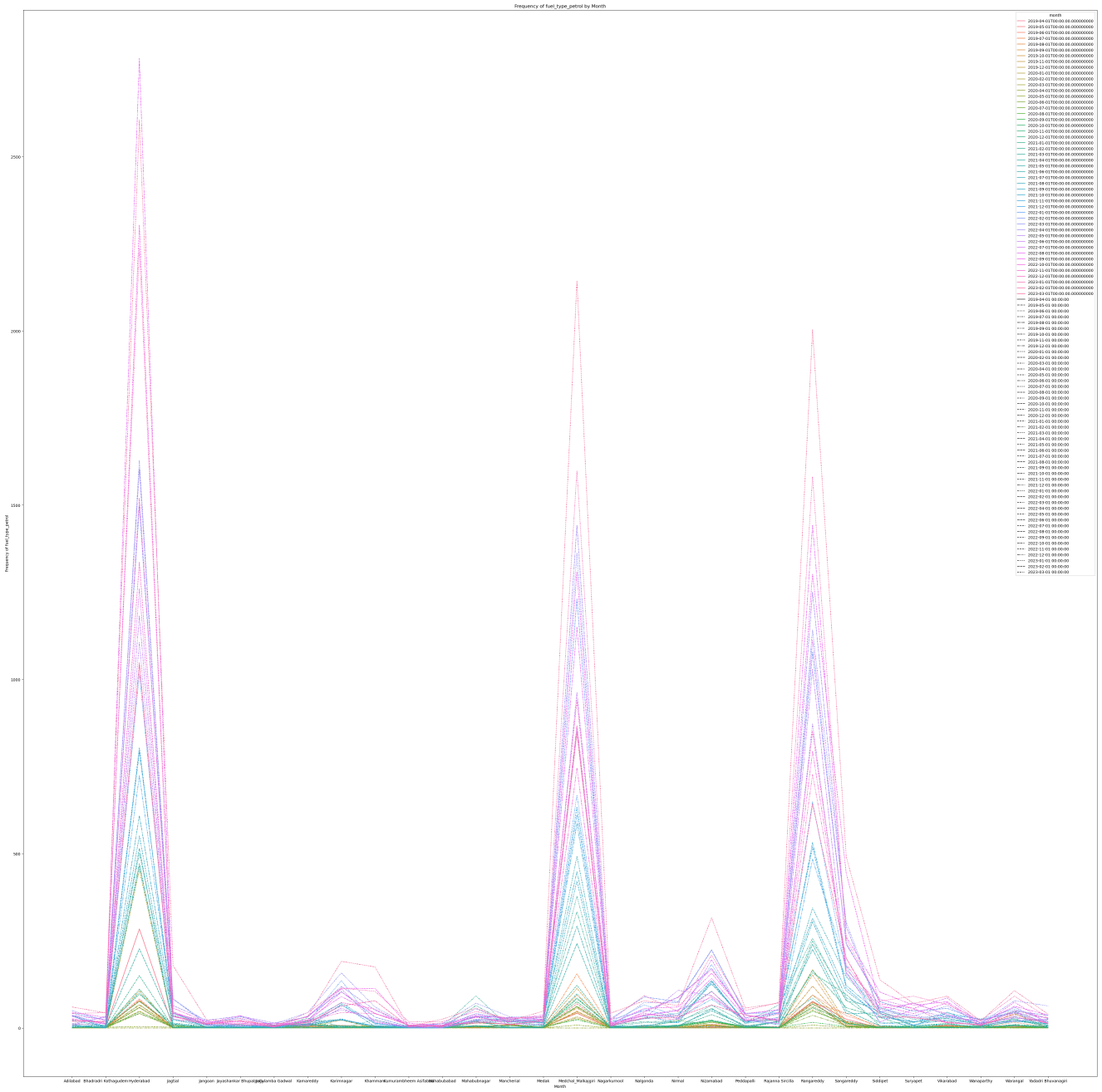


In [58]:

```
# Create a pivot table to count occurrences of 'fuel_type_petrol' for each combination of 'district' and 'month'
Dataset2_q1_pivot_table_2 = Dataset2_q1.pivot_table(index='district', columns='month', values='fuel_type_diesel')
# Plotting with Seaborn  # Optional: Set the figure size
plt.figure(figsize=(50,50))
sns.lineplot(data=Dataset2_q1_pivot_table_2)  # Transpose the pivot table for proper plotting

# Labeling axes and adding a title
plt.xlabel('Month')
plt.ylabel('Frequency of fuel_type_petrol')
plt.title('Frequency of fuel_type_petrol by Month')

# Show the plot
plt.show()
```



In [59]:
```
# Create a pivot table to count occurrences of 'fuel_type_petrol' for each combination of 'district' and 'month'
Dataset2_q1_pivot_table_3 = Dataset2_q1.pivot_table(index='district', columns='month', values='fuel_type_electric')
# Plotting with Seaborn  # Optional: Set the figure size
plt.figure(figsize=(50,50))
sns.lineplot(data=Dataset2_q1_pivot_table_3)  # Transpose the pivot table for proper plotting
# Labeling axes and adding a title
plt.xlabel('Month')
plt.ylabel('Frequency of fuel_type_petrol')
plt.title('Frequency of fuel_type_petrol by Month')
# Show the plot
plt.show()
```



In [22]:
```
ts? Are there any districts with a predominant preference for a specific vehicle class? Consider FY 2022 for analysis.'
```

Out[22]: 'How does the distribution of vehicles vary by vehicle class (MotorCycle, MotorCar, AutoRickshaw, Agriculture) across different districts? Are there any districts with a predominant preference for a specific vehicle class? Consider FY 2022 for analysis.'

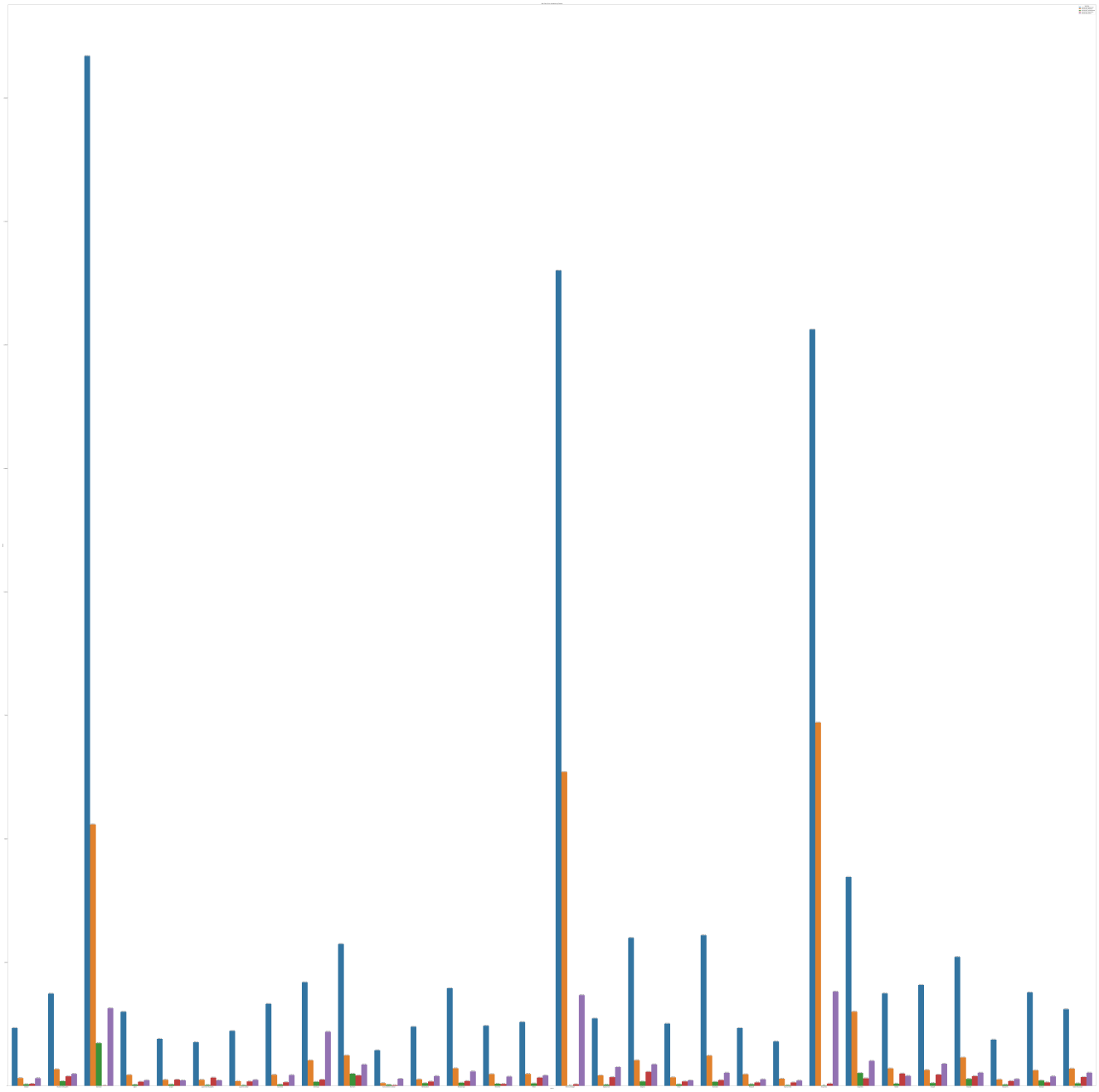
In [60]:
```
#Question-6
data_2022 = dataset_2[dataset_2['month'].dt.year == 2022]

# Select specific columns and create a DataFrame
Dataset_q2 = data_2022[['district','vehicleClass_MotorCycle','vehicleClass_MotorCar','vehicleClass_AutoRickshaw','vehic
# Group by 'district' and sum the 'estamps_challans_rev'
result2_2_df = Dataset_q2.groupby('district')['vehicleClass_MotorCycle','vehicleClass_MotorCar','vehicleClass_AutoRicks
```

```
/var/folders/qx/_z5tt86j3vggw6r55m0t2q_r0000gn/T/ipykernel_52125/2749530164.py:7: FutureWarning: Indexing with multip
le keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
  result2_2_df = Dataset_q2.groupby('district')['vehicleClass_MotorCycle','vehicleClass_MotorCar','vehicleClass_AutoR
ickshaw','vehicleClass_Agriculture','vehicleClass_others'].sum().reset_index()
```

In [24]:

```
# Melt the DataFrame to have variables in a single column
melted_data = pd.melt(result2_2_df, id_vars=['district'], value_vars=['vehicleClass_MotorCycle','vehicleClass_MotorCar'

# Create a bar plot with Seaborn
plt.figure(figsize=(200,200))  # Optional: Set the figure size

sns.barplot(data=melted_data, x='district', y='value', hue='Variable', ci=None)

# Set labels and title
plt.xlabel('District')
plt.ylabel('Values')
plt.title('Bar Plot of Four Variables by District')

# Show the legend
plt.legend(title='Variables', loc='upper right')

# Add numeric labels on top of the bars
for p in plt.gca().patches:
    height = p.get_height()
    plt.annotate(f'{height:.2f}', (p.get_x() + p.get_width() / 2, height),
                 ha='center', va='bottom')

# Show the plot
plt.show()
```

```
/var/folders/qx/_z5tt86j3vggw6r55m0t2q_r0000gn/T/ipykernel_52125/1156895206.py:7: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(data=melted_data, x='district', y='value', hue='Variable', ci=None)
```



In [61]:
```
#Question7
#List down the top 3 and bottom 3 districts that have shown the highest and lowest vehicle sales growth during FY 2022
#compared to FY 2021?(Consider and compare categories: Petrol, Diesel and Electric)

#the top 5 districts where e-stamps revenue contributes significantly more to the revenue than the documents in FY 2022
# Filter the DataFrame for data from the year 2022
dataset_2_3_2022 = dataset_2[dataset_2['month'].dt.year == 2022]
# Select specific columns and create a DataFrame
Dataset2_q3 = dataset_2_3_2022[['district', 'fuel_type_petrol']]
result2_3_df = Dataset2_q3.groupby('district')['fuel_type_petrol'].sum().reset_index()
# FY-2021.........
dataset_2_3_2021 = dataset_2[dataset_2['month'].dt.year == 2021]
# Select specific columns and create a DataFrame
Dataset2_q3_1 = dataset_2_3_2021[['district', 'fuel_type_petrol']]
result2_3_1_df = Dataset2_q3_1.groupby('district')['fuel_type_petrol'].sum().reset_index()
Dataset2_q3_petrol=result2_3_df.merge(result2_3_1_df, on = "district", how = "outer",suffixes=('_2022', '_2021'))
#converting into pivot_table
Dataset2_q3_petrol_pivot_table = Dataset2_q3_petrol.pivot_table(index='district', values=['fuel_type_petrol_2022','fuel
```

In [26]:
```
top_and_bottom_petrol = Dataset2_q3_petrol_pivot_table['fuel_type_petrol_2021'] - Dataset2_q3_petrol_pivot_table['fuel_
top_and_bottom_petrol_df = pd.DataFrame({
    'District': top_and_bottom_petrol.index,  # Assuming the districts are in the index
    'Petrol Difference': top_and_bottom_petrol
})
# Print the new DataFrame and top 3  district with highest petrol sales.
top_and_bottom_petrol_df = top_and_bottom_petrol_df.drop(columns='District')
top_petrol_df=top_and_bottom_petrol_df.sort_values('Petrol Difference', ascending= False)
top_petrol_df=top_petrol_df.iloc[0:3]
print(top_petrol_df)

## top 3 districrs with lowest petrol sales.........
bottom_petrol_df=top_and_bottom_petrol_df.sort_values('Petrol Difference', ascending=True)
bottom_petrol_df=bottom_petrol_df.iloc[0:3]
print(bottom_petrol_df)
```

```
           Petrol Difference
district
Warangal             25381.0
Nizamabad            18846.0
Khammam              13922.0
                     Petrol Difference
district
Rangareddy                     -11637.0
Kumurambheem Asifabad            2985.0
Jogulamba Gadwal                 3191.0
```

In [62]:
```
#..... COMPARING THE PRICES OF DISEAL OF 2021 AND 2022........

# Filter the DataFrame for data from the year 2022
dataset_2_3_2022 = dataset_2[dataset_2['month'].dt.year == 2022]
# Select specific columns and create a DataFrame
Dataset2_q3_d = dataset_2_3_2022[['district', "fuel_type_diesel"]]
result2_3_d_df = Dataset2_q3_d.groupby('district')['fuel_type_diesel'].sum().reset_index()
# FY-2021........
dataset_2_3_2021 = dataset_2[dataset_2['month'].dt.year == 2021]
# Select specific columns and create a DataFrame
Dataset2_q3_d1 = dataset_2_3_2021[['district', 'fuel_type_diesel']]
result2_3_d1_df = Dataset2_q3_d1.groupby('district')['fuel_type_diesel'].sum().reset_index()
Dataset2_q3_d1_diesel=result2_3_d1_df.merge(result2_3_d_df, on = "district", how = "outer",suffixes=('_2022', '_2021'))
#converting into pivot_table
Dataset2_q3_diesel_pivot_table = Dataset2_q3_d1_diesel.pivot_table(index='district', values=['fuel_type_diesel_2022','f
#print(Dataset2_q3_diesel_pivot_table)
```

In [63]:
```
top_and_bottom_Diesel = Dataset2_q3_diesel_pivot_table['fuel_type_diesel_2022'] - Dataset2_q3_diesel_pivot_table['fuel_
top_and_bottom_Diesel_df = pd.DataFrame({
    'District': top_and_bottom_Diesel.index,  # Assuming the districts are in the index
    'Diesel Difference': top_and_bottom_Diesel
})
# Print the new DataFrame and top 3  district with highest petrol sales.
top_and_bottom_Diesel_df = top_and_bottom_Diesel_df.drop(columns='District')
print("The Top 3 Districts with Highest Diesel Values")
top_diesel_df=top_and_bottom_Diesel_df.sort_values('Diesel Difference', ascending= False)
top_diesel_df=top_diesel_df.iloc[0:3]
print(top_diesel_df)

print("The Top 3 Districts with Lowest Diesel Values")
## top 3 districts with lowest Diesel sales.........
bottom_diesel_df=top_and_bottom_Diesel_df.sort_values('Diesel Difference', ascending=True)
bottom_diesel_df=bottom_diesel_df.iloc[0:3]
print(bottom_diesel_df)
```

```
The Top 3 Districts with Highest Diesel Values
             Diesel Difference
district
Warangal                4686.0
Nalgonda                2301.0
Mahabubnagar            2120.0
The Top 3 Districts with Lowest Diesel Values
                    Diesel Difference
district
Karimnagar                    -5863.0
Rangareddy                    -4913.0
Medchal_Malkajgiri            -2523.0
```

In [64]:
```
#..... COMPARING THE PRICES OF ELECTRIC OF 2021 AND 2022........

# Filter the DataFrame for data from the year 2022
dataset_2_3_2022 = dataset_2[dataset_2['month'].dt.year == 2022]
# Select specific columns and create a DataFrame
Dataset2_q3_e = dataset_2_3_2022[['district', "fuel_type_electric"]]
result2_3_e_df = Dataset2_q3_e.groupby('district')['fuel_type_electric'].sum().reset_index()
# FY-2021........
dataset_2_3_2021 = dataset_2[dataset_2['month'].dt.year == 2021]
# Select specific columns and create a DataFrame
Dataset2_q3_e1 = dataset_2_3_2021[['district', 'fuel_type_electric']]
result2_3_e1_df = Dataset2_q3_e1.groupby('district')['fuel_type_electric'].sum().reset_index()
Dataset2_q3_electric=result2_3_e1_df.merge(result2_3_e_df, on = "district", how = "outer",suffixes=('_2022', '_2021'))
#converting into pivot_table
Dataset2_q3_electric_pivot_table = Dataset2_q3_electric.pivot_table(index='district', values=['fuel_type_electric_2022'
```

In [65]:
```
top_and_bottom_electric = Dataset2_q3_electric_pivot_table['fuel_type_electric_2022'] - Dataset2_q3_electric_pivot_tabl
top_and_bottom_electric_df = pd.DataFrame({
    'District': top_and_bottom_electric.index,  # Assuming the districts are in the index
    'Electric Difference': top_and_bottom_electric
})
# Print the new DataFrame and top 3  district with highest petrol sales.
top_and_bottom_electric_df = top_and_bottom_electric_df.drop(columns='District')
print("The Top 3 Districts with Highest Diesel Values")
top_electric_df=top_and_bottom_electric_df.sort_values('Electric Difference', ascending= False)
top_electric_df=top_electric_df.iloc[0:3]
print(top_electric_df)

print("The Top 3 Districts with Lowest Diesel Values")
## top 3 districts with lowest Diesel sales..........
bottom_electric_df=top_and_bottom_electric_df.sort_values('Electric Difference', ascending=True)
bottom_electric_df=bottom_electric_df.iloc[0:3]
print(bottom_electric_df)
```

```
The Top 3 Districts with Highest Diesel Values
                     Electric Difference
district
Jogulamba Gadwal                   -55.0
Kumurambheem Asifabad              -57.0
Rajanna Sircilla                   -63.0
The Top 3 Districts with Lowest Diesel Values
                  Electric Difference
district
Hyderabad                    -12365.0
Rangareddy                    -8535.0
Medchal_Malkajgiri            -7730.0
```

In [66]:
```
#.......Ts-Ipass (Telangana State Industrial Project Approval and Self Certification System).......
```

In [68]:
```
#.....Ts-Ipass (Telangana State Industrial Project Approval and Self Certification System)......

dataset_3= dim_districts.merge(fact_ts_ipass, on = "dist_code", how = "outer")
#IS THERE ANY NULL VALUES OR NOT
dataset_3.isna().any()
dataset_3['month'] = pd.to_datetime(dataset_3['month'])
data_3_2022 = dataset_3[dataset_3['month'].dt.year == 2022]
```

In [69]:
```
result3_1_df = data_3_2022.groupby('sector')['investment in cr'].sum().reset_index()
print('#......List down the top 5 sectors that have witnessed the most significant investments in FY 2022.....')
print(result3_1_df.sort_values('investment in cr', ascending= False).head(5))
```

```
#......List down the top 5 sectors that have witnessed the most significant investments in FY 2022.....
                                        sector  investment in cr
14  Real Estate,Industrial Parks and IT Buildings         3990.2522
12                             Plastic and Rubber         3699.1197
11                Pharmaceuticals and Chemicals         2649.1294
15              Solar and Other Renewable Energy         2201.5980
2                                    Beverages         1848.1920
```

In [70]:
```
..... List down the top 3 districts that have attracted the most significant sector investments during FY 2019 to 2022?
taset_3['month'] = pd.to_datetime(dataset_3['month'])
ta_3_years_2019_to_2022 = dataset_3[(dataset_3['month'].dt.year >= 2019) & (dataset_3['month'].dt.year <= 2022)]
```

In [39]:
```
result3_2_df = data_3_years_2019_to_2022.groupby(['district','sector'])['investment in cr'].sum().reset_index()
print("#..... List down the top 3 districts that have attracted the most significant sector investments during FY 201
print(result3_2_df.sort_values('investment in cr',ascending=False).head(3))
```

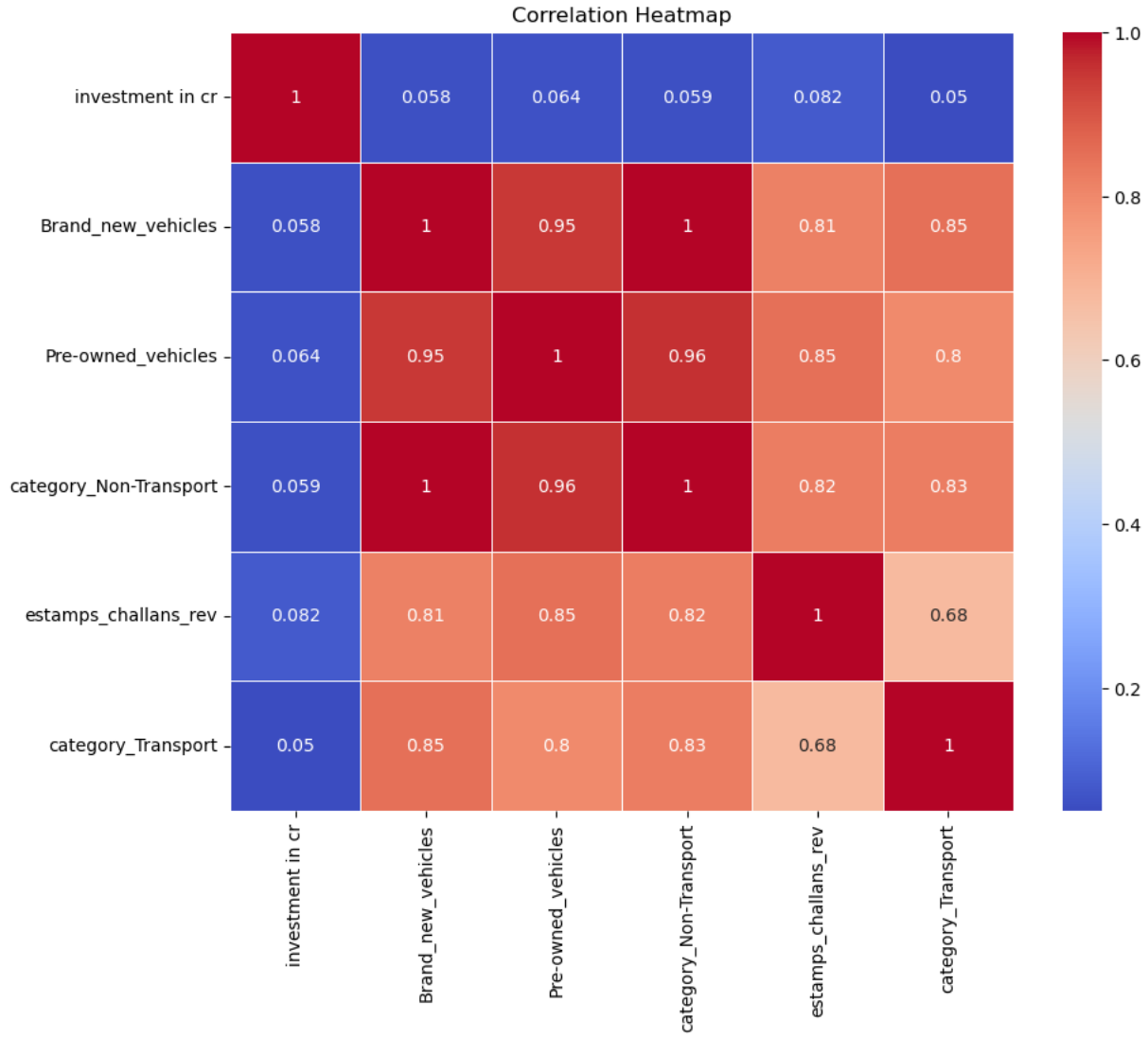
```
#..... List down the top 3 districts that have attracted the most significant sector investments during FY 2019 to
2022? What factors could have led to the substantial investments in these particular districts?....
       district                                             sector  \
331  Rangareddy      Real Estate,Industrial Parks and IT Buildings
296  Peddapalli  Fertlizers Organic and Inorganic,Pesticides,In...
329  Rangareddy                                  Plastic and Rubber

     investment in cr
331        28970.2729
296         5254.2800
329         4390.2628
```

In [71]:
```
#...Is there any relationship between district investments, vehicles sales and stamps revenue
#within the same district between FY 2021 and 2022?
#fact_transport = pd.read_csv("fact_transport.csv")
#fact_ts_ipass = pd.read_csv("fact_TS_iPASS.csv")
Data_3_3Q = fact_ts_ipass.merge(fact_transport, on = "dist_code", how = "outer")
Data_3_3Q = Data_3_3Q.merge(dim_districts, on = "dist_code", how = "outer")
#Data_3_3Q['month'] = pd.to_datetime(Data_3_3Q['month_x'])
Data_3_3Q = Data_3_3Q.merge(fact_stamps, on = "dist_code", how = "outer")
```

In [74]:
```
Data_3_3Q_N=Data_3_3Q[['district','investment in cr','Brand_new_vehicles','Pre-owned_vehicles','category_Non-Transport'
Data_3_3Q_N['month'] = pd.to_datetime(Data_3_3Q_N['month'])
Data_3_3Q_C= Data_3_3Q_N[(Data_3_3Q_N['month'].dt.year >= 2021) & (Data_3_3Q_N['month'].dt.year <= 2022)]
```

In [73]:
```
# Calculate the correlation matrix
correlation_matrix = Data_3_3Q_C.corr()
# Create a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')
# Show the heatmap
plt.show()
```

```
/var/folders/qx/_z5tt86j3vggw6r55m0t2q_r0000gn/T/ipykernel_52125/1408876288.py:2: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid column
s or specify the value of numeric_only to silence this warning.
  correlation_matrix = Data_3_3Q_C.corr()
```

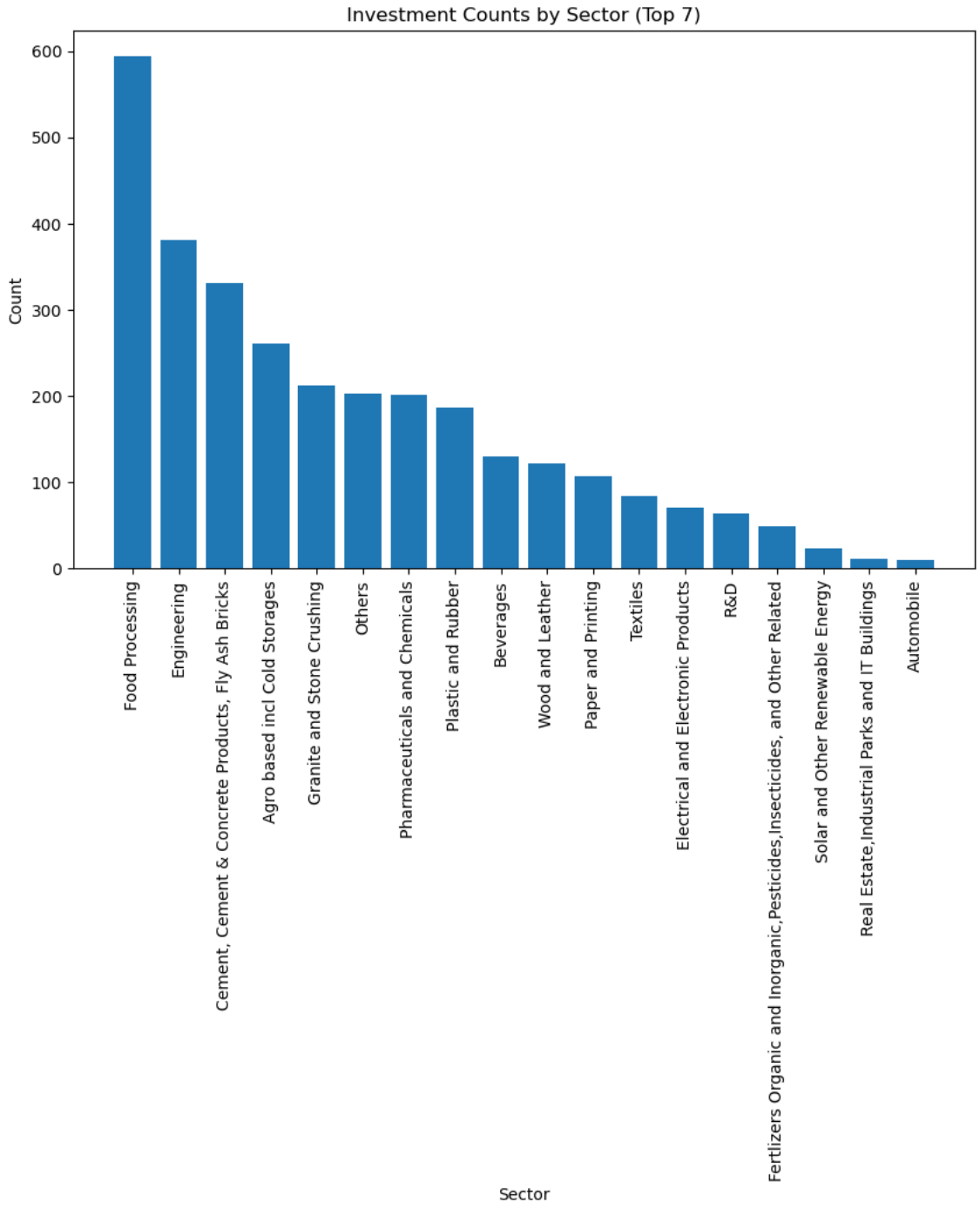


In [75]:
```
Data_3_4Q = fact_ts_ipass
Data_3_4Q=Data_3_4Q.merge(dim_districts, on='dist_code',how='outer')
```

In [76]:
```
Data_3_4Q['month'] = pd.to_datetime(Data_3_4Q['month'])
Data_3_4Q_2021_to_2022 = Data_3_4Q[(Data_3_4Q['month'].dt.year >= 2021) & (Data_3_4Q['month'].dt.year <= 2022)]

# Group by 'sector' and 'dist_code', and count the number of occurrences
Data_3_4Q_2021_to_2022 = Data_3_4Q_2021_to_2022.groupby('sector')['district'].size().reset_index(name='count').sort_val
# Sort the DataFrame by the count column in descending order

# Print the top 700 rows
print(Data_3_4Q_2021_to_2022.head(7))
```

```
                                       sector  count
7                                Food Processing    594
5                                    Engineering    380
3   Cement, Cement & Concrete Products, Fly Ash Br...    331
0                       Agro based incl Cold Storages    261
8                          Granite and Stone Crushing    212
9                                          Others    203
11                      Pharmaceuticals and Chemicals    201
```

In [77]:
```
plt.figure(figsize=(10, 6))
plt.bar(Data_3_4Q_2021_to_2022['sector'], Data_3_4Q_2021_to_2022['count'])

# Set labels and title
plt.xlabel('Sector')
plt.ylabel('Count')
plt.title('Investment Counts by Sector (Top 7)')

# Rotate x-axis labels for better readability
plt.xticks(rotation=90)

# Show the plot
plt.show()
```



In [ ]:

In [ ]: