telangana-growth-analysis

September 14, 2023

1 Analyse Growth And Present Insights To The Telangana Government

About the project: - Telangana is one of the fastest-growing states in India and one of the states with an open data policy. (They have published all their data online)
- Peter Pandey is an aspiring data analyst looking for a project with real-time data to add to his portfolio. He wanted to analyse Telangana's growth among different sectors quantitatively and provide useful Insights to the Telangana government that would help them to make data-informed decisions that would further support the growth of the state
Resource: https://codebasics.io/challenge/codebasics-resume-project-challenge

Objective: - Explore the Stamp Registration, Transportation and Ts-Ipass Datasets. - Analyze trends and patterns, categories and time period. - Identify growth opportunities and areas needing attention. - Find correlation among these departments and report the overall growth of the state through insights and relevant visuals such as shape maps.

Audiences (Stackholders): - From Telangana Government 1. Kalvakuntla Taraka Rama Rao / KTR Minister for IT, Industries and Municipal Administration: https://www.linkedin.com/in/ktramarao 2. Jayesh Ranjan Secretary, Information Technology (IT); Government of Telangana: https://www.linkedin.com/in/jayesh-ranjan-37415963 3. Dileep Konatham Director at Government of Telangana https://www.linkedin.com/in/dileep-konatham-2624b91b5 4. Venu Panjarla Open Data Telangana https://www.linkedin.com/in/venupanjarla - From Codebasics Team 1. Dhaval Patel Founder, Codebasics https://www.linkedin.com/in/dhavalsays/ 2. Hemanand Vadivel Co-Founder, Codebasics https://www.linkedin.com/in/hemvad/ codebasics.io

Use these hashtags so that we can track your work:
#codebasicsresumeprojectchallenge #OpenDataTelangana #CRPC7
Other Hashtags You Can Use
#GovernmentofTelangana #HappeningHyderabad

Primary Questions: - Stamp Registration 1. 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. 2. 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? 3. Is there any alteration of e-Stamp challan count and document registration count pattern since the implementation of e-Stamp challan? If so, what suggestions would you propose to the government? 4. Categorize districts into three segments based on their stamp registration revenue generation during the fiscal year 2021 to 2022.

- Transportation 5. 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) 6. 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. 7. 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) - Ts-Ipass (Telangana State Industrial Project Approval and Self Certification System) 8. List down the top 5 sectors that have witnessed the most significant investments in FY 2022. 9. 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? 10. Is there any relationship between district investments, vehicles sales and stamps revenue within the same district between FY 2021 and 2022? 11. Are there any particular sectors that have shown substantial investment in multiple districts between FY 2021 and 2022? 12. Can we identify any seasonal patterns or cyclicality in the investment trends for specific sectors? Do certain sectors experience higher investments during particular months?

Author: Bao Thai
Linkedin: https://www.linkedin.com/in/bao-quoc-508008181/
Gmail: QuocBao363@gmail.com

1.1 Import needed packages and libraries

[223]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

1.2 Import dataset

[224]:
```
dim_date = pd.read_csv("C:/Users/Admin/Desktop/Projects/TelanganaGrowthAnalysis/
  ↪dataset/dim_date.csv")
date_data = dim_date.copy()
date_data.head()
```

[224]:

	month	Mmm	quarter	fiscal_year
0	2019-04-01	Apr	Q1	2019
1	2019-05-01	May	Q1	2019
2	2019-06-01	Jun	Q1	2019
3	2019-07-01	Jul	Q2	2019
4	2019-08-01	Aug	Q2	2019

[225]:
```
date_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 4 columns):
```

```
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   month        48 non-null     object
 1   Mmm          48 non-null     object
 2   quarter      48 non-null     object
 3   fiscal_year  48 non-null     int64
dtypes: int64(1), object(3)
memory usage: 1.6+ KB
```

[226]:
```
date_data.isna().sum()
```

[226]:
```
month          0
Mmm            0
quarter        0
fiscal_year    0
dtype: int64
```

[227]:
```
date_data.duplicated().sum()
```

[227]:
```
0
```

[228]:
```
dim_districts = pd.read_csv("C:/Users/Admin/Desktop/Projects/
  ↪TelanganaGrowthAnalysis/dataset/dim_districts.csv")
districts = dim_districts.copy()
districts.head()
```

[228]:

	dist_code	district
0	19_1	Adilabad
1	22_2	Bhadradri Kothagudem
2	21_1	Hanumakonda
3	16_1	Hyderabad
4	20_2	Jagtial

[229]:
```
districts.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 33 entries, 0 to 32
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   dist_code  33 non-null     object
 1   district   33 non-null     object
dtypes: object(2)
memory usage: 660.0+ bytes
```

[230]:
```
districts.isna().sum()
```

[230]:
```
dist_code    0
district     0
dtype: int64
```

[231]:
```
districts.duplicated().sum()
```

[231]:
```
0
```

[232]:
```
fact_stamps = pd.read_csv("C:/Users/Admin/Desktop/Projects/
 ↪TelanganaGrowthAnalysis/dataset/fact_stamps.csv")
stamps = fact_stamps.copy()
stamps.head()
```

[232]:

	dist_code	month	documents_registered_cnt	documents_registered_rev	\
0	14_1	2019-04-01	4533	59236363	
1	17_3	2019-04-01	4151	41508762	
2	20_3	2019-04-01	2116	23674170	
3	21_5	2019-04-01	1089	15915285	
4	23_1	2019-04-01	6133	82593256	

	estamps_challans_cnt	estamps_challans_rev
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0

[233]:
```
stamps.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1504 entries, 0 to 1503
Data columns (total 6 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   dist_code                1504 non-null   object
 1   month                    1504 non-null   object
 2   documents_registered_cnt 1504 non-null   int64
 3   documents_registered_rev 1504 non-null   int64
 4   estamps_challans_cnt     1504 non-null   int64
 5   estamps_challans_rev     1504 non-null   int64
dtypes: int64(4), object(2)
memory usage: 70.6+ KB
```

[234]:
```
stamps.isna().sum()
```

[234]:
```
dist_code                0
month                    0
```

```
documents_registered_cnt    0
documents_registered_rev    0
estamps_challans_cnt        0
estamps_challans_rev        0
dtype: int64
```

[235]:
```
stamps.duplicated().sum()
```

[235]:
```
0
```

[236]:
```
fact_TS_iPASS = pd.read_csv("C:/Users/Admin/Desktop/Projects/
 ↪TelanganaGrowthAnalysis/dataset/fact_TS_iPASS.csv")
ts_ipass = fact_TS_iPASS.copy()
ts_ipass.head()
```

[236]:

	dist_code	month	sector	investment in cr	\
0	14_1	01-04-2019	Engineering	2.3200	
1	19_1	01-04-2019	Engineering	0.6250	
2	20_3	01-04-2019	Wood and Leather	0.2000	
3	20_3	01-04-2019	Textiles	0.2675	
4	21_5	01-04-2019	Electrical and Electronic Products	0.1200	

	number_of_employees
0	15
1	13
2	8
3	27
4	5

[237]:
```
ts_ipass.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5753 entries, 0 to 5752
Data columns (total 5 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   dist_code            5753 non-null   object
 1   month                5753 non-null   object
 2   sector               5753 non-null   object
 3   investment in cr     5753 non-null   float64
 4   number_of_employees  5753 non-null   int64
dtypes: float64(1), int64(1), object(3)
memory usage: 224.9+ KB
```

[238]:
```
ts_ipass.isna().sum()
```

[238]:
```
dist_code              0
month                  0
sector                 0
investment in cr       0
number_of_employees    0
dtype: int64
```

[239]:
```
ts_ipass.duplicated().sum()
```

[239]:
```
0
```

[240]:
```
fact_transport = pd.read_csv("C:/Users/Admin/Desktop/Projects/
 ↪TelanganaGrowthAnalysis/dataset/fact_transport.csv")
transport = fact_transport.copy()
transport.head()
```

[240]:

	dist_code	month	fuel_type_petrol	fuel_type_diesel	\
0	15_1	2019-04-01	17910	3011	
1	18_2	2019-04-01	3066	306	
2	20_3	2019-04-01	1577	215	
3	21_3	2019-04-01	1961	281	
4	21_7	2019-04-01	1552	309	

	fuel_type_electric	fuel_type_others	vehicleClass_MotorCycle	\
0	76	22	15308	
1	6	0	2995	
2	0	0	1546	
3	2	0	1939	
4	0	0	1512	

	vehicleClass_MotorCar	vehicleClass_AutoRickshaw	vehicleClass_Agriculture	\
0	4429	0	4	
1	142	49	64	
2	79	29	21	
3	72	72	48	
4	76	69	109	

	vehicleClass_others	seatCapacity_1_to_3	seatCapacity_4_to_6	\
0	1278	16110	4182	
1	128	3156	189	
2	117	1683	104	
3	113	2082	146	
4	95	1696	145	

	seatCapacity_above_6	Brand_new_vehicles	Pre-owned_vehicles	\
0	717	19542	1477	
1	33	3322	56	

```
2                      5                1751                    41
3                     16                2209                    35
4                     20                1820                    41

   category_Non-Transport  category_Transport
0                   19856                1163
1                    3203                 175
2                    1648                 144
3                    2075                 169
4                    1701                 160
```

[241]:
```
transport.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1440 entries, 0 to 1439
Data columns (total 18 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   dist_code                  1440 non-null   object
 1   month                      1440 non-null   object
 2   fuel_type_petrol           1440 non-null   int64
 3   fuel_type_diesel           1440 non-null   int64
 4   fuel_type_electric         1440 non-null   int64
 5   fuel_type_others           1440 non-null   int64
 6   vehicleClass_MotorCycle    1440 non-null   int64
 7   vehicleClass_MotorCar      1440 non-null   int64
 8   vehicleClass_AutoRickshaw  1440 non-null   int64
 9   vehicleClass_Agriculture   1440 non-null   int64
 10  vehicleClass_others        1440 non-null   int64
 11  seatCapacity_1_to_3        1440 non-null   int64
 12  seatCapacity_4_to_6        1440 non-null   int64
 13  seatCapacity_above_6       1440 non-null   int64
 14  Brand_new_vehicles         1440 non-null   int64
 15  Pre-owned_vehicles         1440 non-null   int64
 16  category_Non-Transport     1440 non-null   int64
 17  category_Transport         1440 non-null   int64
dtypes: int64(16), object(2)
memory usage: 202.6+ KB
```

[242]:
```
transport.isna().sum()
```

[242]:
```
dist_code                  0
month                      0
fuel_type_petrol           0
fuel_type_diesel           0
fuel_type_electric         0
fuel_type_others           0
```

```
vehicleClass_MotorCycle      0
vehicleClass_MotorCar        0
vehicleClass_AutoRickshaw    0
vehicleClass_Agriculture     0
vehicleClass_others          0
seatCapacity_1_to_3          0
seatCapacity_4_to_6          0
seatCapacity_above_6         0
Brand_new_vehicles           0
Pre-owned_vehicles           0
category_Non-Transport       0
category_Transport           0
dtype: int64
```

[243]:
```
transport.duplicated().sum()
```

[243]:
```
0
```

1.3 Merge dataset

1.3.1 For analyzing about Stamp Registration

[244]:
```
stamps_df_temp = pd.merge(districts, stamps, on = 'dist_code', how = 'inner')
stamps_df_temp.head()
```

[244]:
```
  dist_code  district       month  documents_registered_cnt  \
0      19_1  Adilabad  2019-04-01                      1043
1      19_1  Adilabad  2019-05-01                      1092
2      19_1  Adilabad  2019-06-01                       824
3      19_1  Adilabad  2019-07-01                      1186
4      19_1  Adilabad  2019-08-01                      1124

   documents_registered_rev  estamps_challans_cnt  estamps_challans_rev
0                  12672655                     0                     0
1                  15177385                     0                     0
2                  11443194                     0                     0
3                  17853370                     0                     0
4                  14955291                     0                     0
```

[245]:
```
stamps_df = pd.merge(date_data, stamps_df_temp, on = 'month', how = 'inner')
stamps_df.head()
```

[245]:
```
        month  Mmm quarter  fiscal_year dist_code              district  \
0  2019-04-01  Apr      Q1         2019      19_1              Adilabad
1  2019-04-01  Apr      Q1         2019      22_2  Bhadradri Kothagudem
2  2019-04-01  Apr      Q1         2019      21_1           Hanumakonda
3  2019-04-01  Apr      Q1         2019      16_1             Hyderabad
```

```
4  2019-04-01  Apr      Q1         2019      20_2               Jagtial

   documents_registered_cnt  documents_registered_rev  estamps_challans_cnt  \
0                      1043                  12672655                     0
1                       542                  12610700                     0
2                      4186                 120047007                     0
3                      5268                 724238098                     0
4                      1943                  26031616                     0

   estamps_challans_rev
0                     0
1                     0
2                     0
3                     0
4                     0
```

1.3.2 For analyzing about Transportation

[246]:
```
transport_temp = pd.merge(districts, transport, on = 'dist_code' ,how = 'inner')
```

[247]:
```
transport_df = pd.merge(date_data, transport_temp, on = 'month', how = 'inner')
transport_df.head()
```

[247]:
```
        month  Mmm quarter  fiscal_year dist_code             district  \
0  2019-04-01  Apr      Q1         2019      19_1             Adilabad
1  2019-04-01  Apr      Q1         2019      22_2  Bhadradri Kothagudem
2  2019-04-01  Apr      Q1         2019      16_1            Hyderabad
3  2019-04-01  Apr      Q1         2019      20_2              Jagtial
4  2019-04-01  Apr      Q1         2019      21_3              Jangoan

   fuel_type_petrol  fuel_type_diesel  fuel_type_electric  fuel_type_others  \
0              1758               326                   0                 2
1              3527               669                   2                 4
2             24309              2957                 284              1540
3              2786               440                   2                 0
4              1961               281                   2                 0

   …  vehicleClass_AutoRickshaw  vehicleClass_Agriculture  \
0  …                         77                        13
1  …                        231                        74
2  …                       1518                         0
3  …                        106                       121
4  …                         72                        48

   vehicleClass_others  seatCapacity_1_to_3  seatCapacity_4_to_6  \
0                  183                 1823                  184
1                  254                 3683                  467
```

```
2                 1742                23082                  5306
3                  146                 2951                   245
4                  113                 2082                   146

   seatCapacity_above_6  Brand_new_vehicles  Pre-owned_vehicles  \
0                    79                2061                  25
1                    52                4096                 106
2                   698               27121                1969
3                    30                3120                 108
4                    16                2209                  35

   category_Non-Transport  category_Transport
0                    1832                 254
1                    3725                 477
2                   25973                3117
3                    2986                 242
4                    2075                 169

[5 rows x 22 columns]
```

1.3.3 For analyzing about Ts Ipass

[248]:
```
ts_ipass_temp = pd.merge(districts, ts_ipass, on = 'dist_code', how = 'inner')
ts_ipass_temp
```

[248]:
```
     dist_code             district       month  \
0         19_1             Adilabad  01-04-2019
1         19_1             Adilabad  01-04-2019
2         19_1             Adilabad  01-04-2019
3         19_1             Adilabad  01-07-2019
4         19_1             Adilabad  01-07-2019
…          …                    …          …
5748      23_3  Yadadri Bhuvanagiri  01-03-2023
5749      23_3  Yadadri Bhuvanagiri  01-03-2023
5750      23_3  Yadadri Bhuvanagiri  01-03-2023
5751      23_3  Yadadri Bhuvanagiri  01-03-2023
5752      23_3  Yadadri Bhuvanagiri  01-03-2023

                                                 sector  investment in cr  \
0                                           Engineering            0.6250
1     Cement, Cement & Concrete Products, Fly Ash Br…           19.3702
2                                       Food Processing            0.0980
3                                             Beverages            0.1300
4     Cement, Cement & Concrete Products, Fly Ash Br…            0.7300
…                                                    …                 …
5748  Fertlizers Organic and Inorganic,Pesticides,In…            1.2500
5749                                    Food Processing           15.4500
```

```
5750                      Pharmaceuticals and Chemicals            1.2800
5751                                   Plastic and Rubber            0.4500
5752                                            Textiles            0.6629

      number_of_employees
0                      13
1                      99
2                       7
3                       3
4                      10
…                     …
5748                   24
5749                   24
5750                   18
5751                    9
5752                   13

[5753 rows x 6 columns]
```

[249]:
```
ts_ipass_temp['month'] = pd.to_datetime(ts_ipass_temp['month'],␣
↪format='%d-%m-%Y').dt.strftime('%Y-%m-%d')
```

[250]:
```
ts_ipass_df = pd.merge(date_data, ts_ipass_temp, on = 'month', how = 'inner')
ts_ipass_df.head()
```

[250]:
```
         month  Mmm quarter  fiscal_year dist_code              district  \
0  2019-04-01  Apr      Q1         2019      19_1              Adilabad
1  2019-04-01  Apr      Q1         2019      19_1              Adilabad
2  2019-04-01  Apr      Q1         2019      19_1              Adilabad
3  2019-04-01  Apr      Q1         2019      22_2  Bhadradri Kothagudem
4  2019-04-01  Apr      Q1         2019      22_2  Bhadradri Kothagudem

                                              sector  investment in cr  \
0                                        Engineering            0.6250
1  Cement, Cement & Concrete Products, Fly Ash Br…           19.3702
2                                    Food Processing            0.0980
3                                  Paper and Printing            0.2000
4                                          Beverages            0.0600

   number_of_employees
0                   13
1                   99
2                    7
3                    6
4                    3
```

1.4 Insight Analysis

1.4.1 Stamp Registration

1. 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.

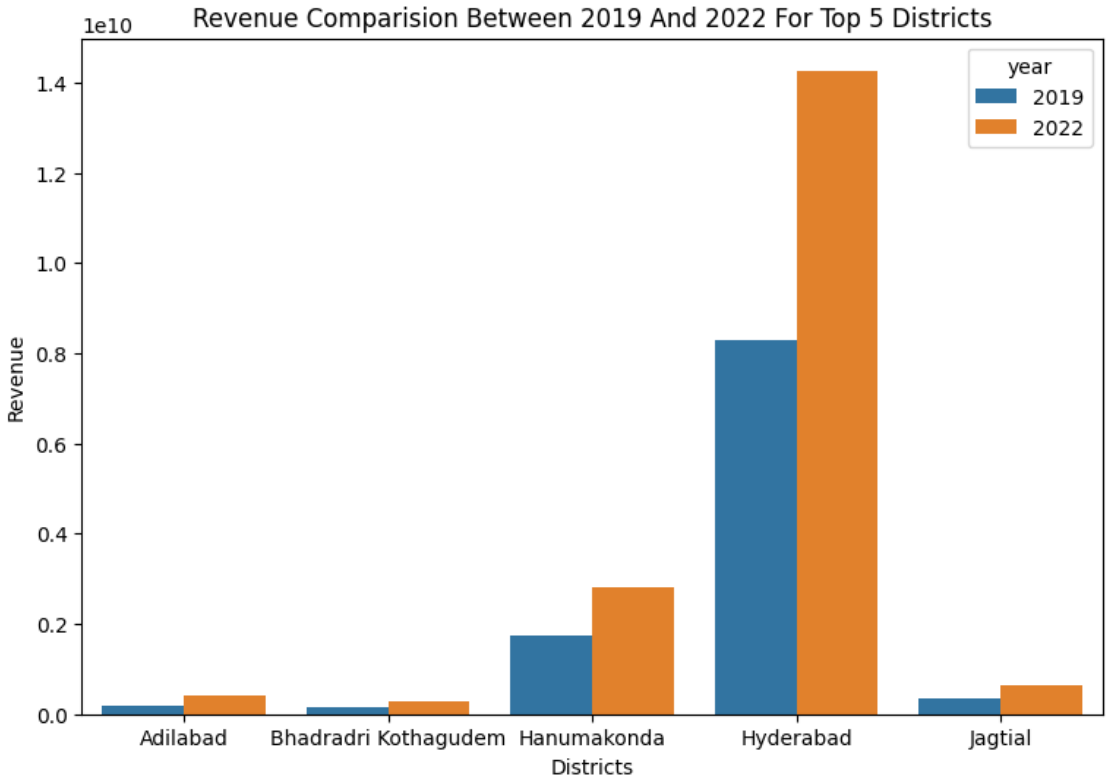
[251]:
```
revenue = stamps_df[stamps_df['fiscal_year'].isin([2019, 2022])].
 ↪groupby(['district','fiscal_year'])['documents_registered_rev'].sum().
 ↪unstack()
revenue['growth'] = ((revenue[2022] - revenue[2019])/revenue[2019])
revenue['growth'].sort_values(ascending = False)
top5_growth_districts = revenue.head(5)
top5_growth_districts
```

[251]:
fiscal_year	2019	2022	growth
district			
Adilabad	178082233	396840845	1.228413
Bhadradri Kothagudem	162466947	292643450	0.801249
Hanumakonda	1738280378	2817238587	0.620704
Hyderabad	8299406278	14266012441	0.718920
Jagtial	357209544	628898536	0.760587

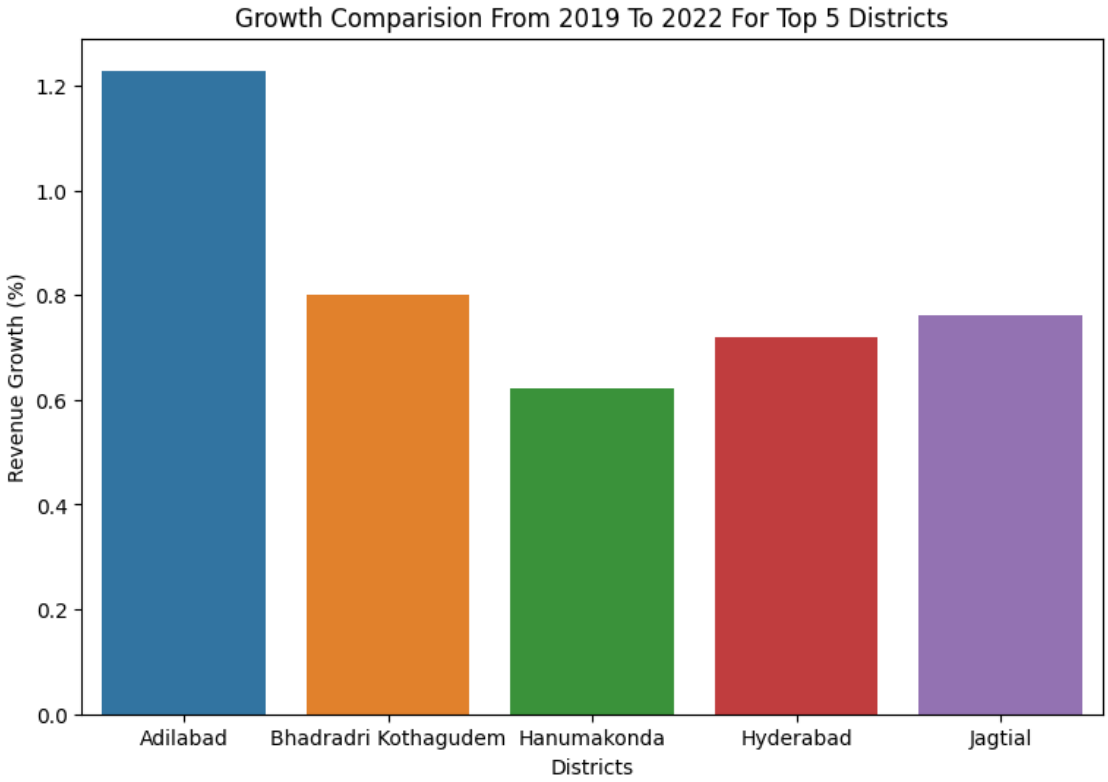
[252]:
```
top5_revenue_districts = top5_growth_districts.reset_index().melt(id_vars =␣
 ↪'district', value_vars = [2019,2022], var_name = 'year', value_name =␣
 ↪'revenue')
top5_revenue_districts
```

[252]:
	district	year	revenue
0	Adilabad	2019	178082233
1	Bhadradri Kothagudem	2019	162466947
2	Hanumakonda	2019	1738280378
3	Hyderabad	2019	8299406278
4	Jagtial	2019	357209544
5	Adilabad	2022	396840845
6	Bhadradri Kothagudem	2022	292643450
7	Hanumakonda	2022	2817238587
8	Hyderabad	2022	14266012441
9	Jagtial	2022	628898536

[253]:
```
plt.figure(figsize = (9,6))
sns.barplot(x= top5_revenue_districts['district'], y =␣
 ↪top5_revenue_districts['revenue'], hue = top5_revenue_districts['year'])
plt.title('Revenue Comparision Between 2019 And 2022 For Top 5 Districts')
plt.xlabel('Districts')
plt.ylabel('Revenue')
plt.show()
```



[254]:
```
plt.figure(figsize = (9,6))
sns.barplot(x= top5_growth_districts.index, y = top5_growth_districts['growth'])
plt.title('Growth Comparision From 2019 To 2022 For Top 5 Districts')
plt.xlabel('Districts')
plt.ylabel('Revenue Growth (%)')
plt.show()
```



The revenue generated from document registration varies across districts in Telangana. Here are the top 5 districts that showed the highest document registration revenue growth between FY 2019 and 2022:

1. **Adilabad**: The revenue increased from **178,082,233** in 2019 to **396,840,845** in 2022, showing a growth of approximately **123%**.
2. **Bhadradri Kothagudem**: The revenue increased from **162,466,947** in 2019 to **292,643,450** in 2022, showing a growth of approximately **80%**.
3. **Hanumakonda**: The revenue increased from **1,738,280,378** in 2019 to **2,817,238,587** in 2022, showing a growth of approximately **62%**.
4. **Hyderabad**: The revenue increased from **8,299,406,278** in 2019 to **14,266,012,441** in 2022, showing a growth of approximately **72%**.
5. **Jagtial**: The revenue increased from **357,209,544** in 2019 to **628,898,536** in 2022, showing a growth of approximately **76%**.

These figures indicate significant growth in document registration revenue across these districts over the period of three fiscal years.

2. 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?

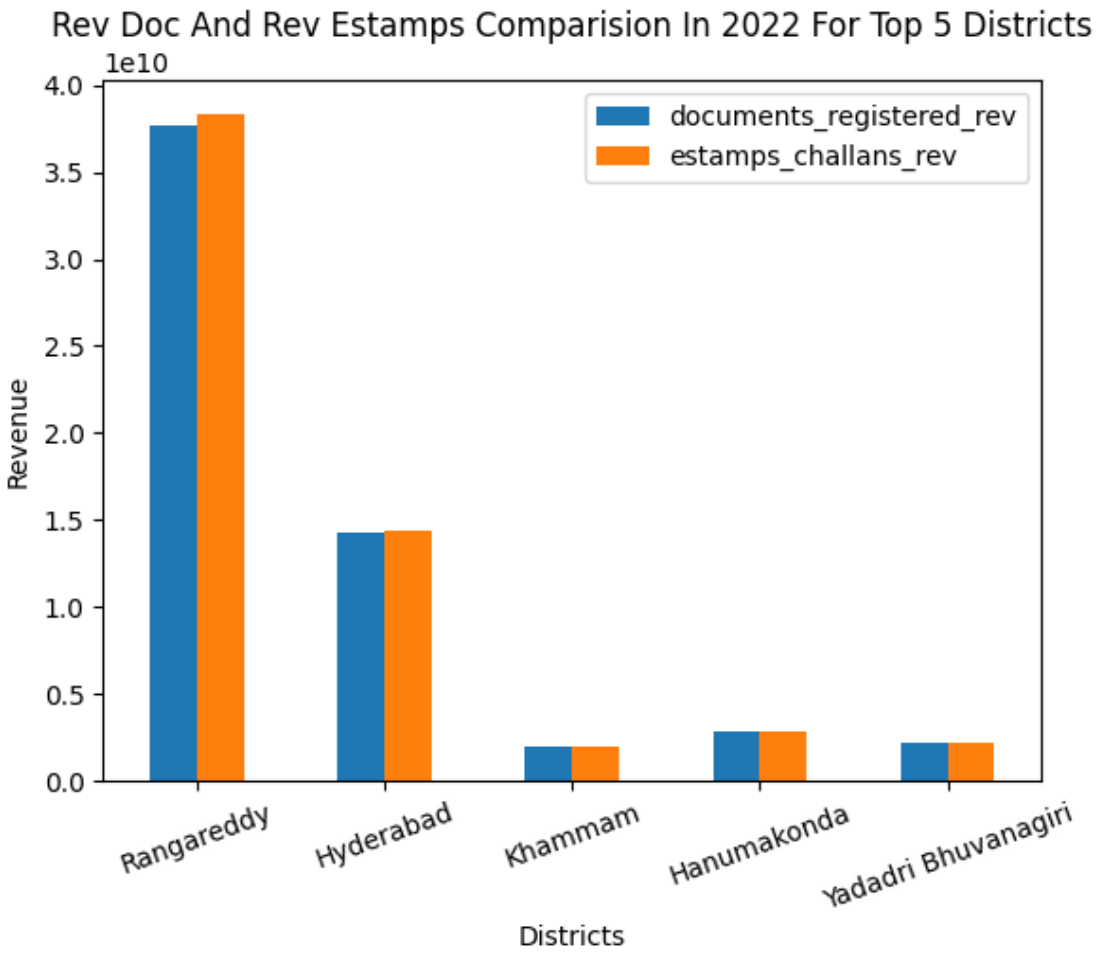
[255]:
```
rev_stamp_doc_2022 = stamps_df[stamps_df['fiscal_year'].isin([2022])].
 ↪groupby(stamps_df['district'])[['documents_registered_rev','estamps_challans_rev']].
 ↪sum()
rev_stamp_doc_2022 = rev_stamp_doc_2022.reset_index()
rev_stamp_doc_2022['different_revenue'] =␣
 ↪rev_stamp_doc_2022['estamps_challans_rev'] - ␣
 ↪rev_stamp_doc_2022['documents_registered_rev']
rev_stamp_doc_2022 = rev_stamp_doc_2022.sort_values(by = 'different_revenue',␣
 ↪ascending = False)
top5_diff_rev_2022 = rev_stamp_doc_2022.head(5)
top5_diff_rev_2022
```

[255]:

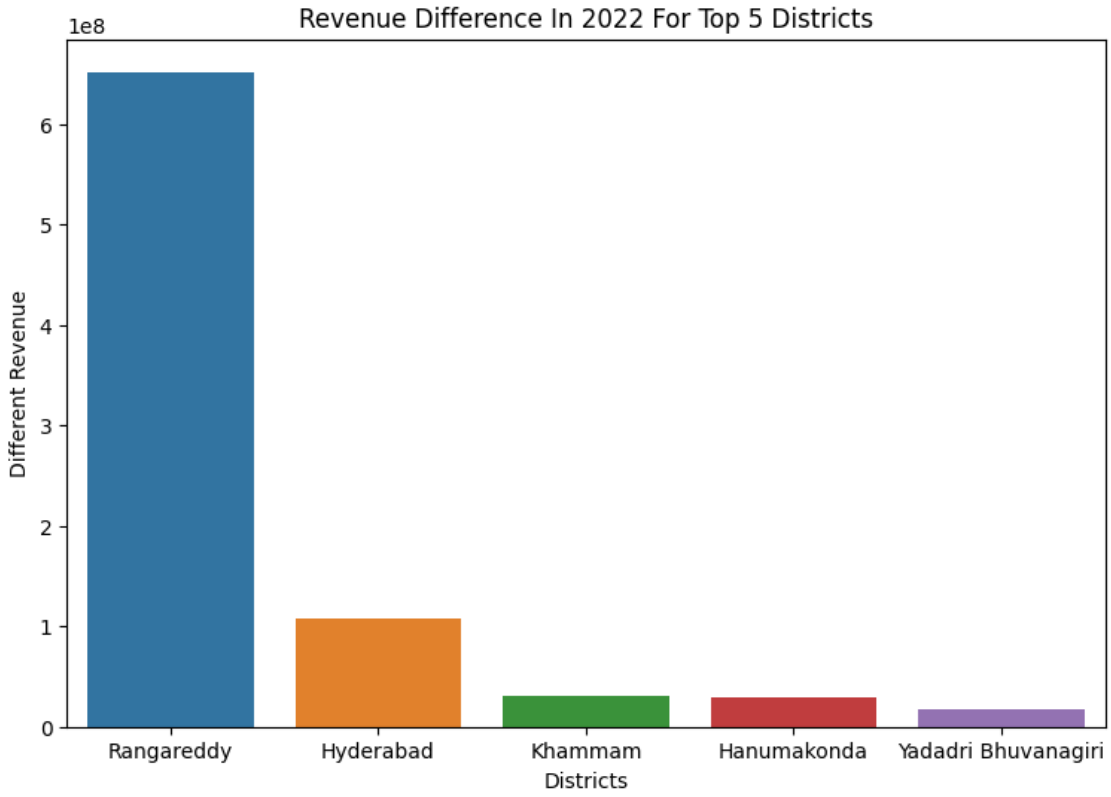
	district	documents_registered_rev	estamps_challans_rev	different_revenue
24	Rangareddy	37697750946	38349357618	651606672
3	Hyderabad	14266012441	14374315032	108302591
9	Khammam	1971647539	2002220314	30572775
2	Hanumakonda	2817238587	2846856844	29618257
31	Yadadri Bhuvanagiri	2167480603	2185270667	17790064

[256]:
```
plt.figure(figsize = (9,6))
top5_diff_rev_2022.plot(x = 'district', y =␣
 ↪['documents_registered_rev','estamps_challans_rev'], kind = 'bar')
plt.title('Rev Doc And Rev Estamps Comparision In 2022 For Top 5 Districts')
plt.xlabel('Districts')
plt.ylabel('Revenue')
plt.xticks(rotation = 20)
plt.show()
```

```
<Figure size 900x600 with 0 Axes>
```



[257]:
```
plt.figure(figsize = (9,6))
sns.barplot(x = top5_diff_rev_2022['district'], y =␣
 ↪top5_diff_rev_2022['different_revenue'])
plt.title('Revenue Difference In 2022 For Top 5 Districts')
plt.xlabel('Districts')
plt.ylabel('Different Revenue')
plt.show()
```



The revenue generated from document registration and e-stamp challans varies across districts. Here are the top 5 districts where e-stamps revenue contributes significantly more to the revenue than the documents in FY 2022:

1. **Rangareddy**: The revenue from documents registered was **37,697,750,946**, while the revenue from e-stamps challans was **38,349,357,618**. The difference in revenue is **651,606,672**.
2. **Hyderabad**: The revenue from documents registered was **14,266,012,441**, while the revenue from e-stamps challans was **14,374,315,032**. The difference in revenue is **108,302,591**.
3. **Khammam**: The revenue from documents registered was **1,971,647,539**, while the revenue from e-stamps challans was **2,002,220,314**. The difference in revenue is **30,572,775**.
4. **Hanumakonda**: The revenue from documents registered was **2,817,238,587**, while the revenue from e-stamps challans was **2,846,856,844**. The difference in revenue is **29,618,257**.
5. **Yadadri Bhuvanagiri**: The revenue from documents registered was **2,167,480,603**, while the revenue from e-stamps challans was **2,185,270,667**. The difference in revenue is **17,790,064**.

These figures indicate that e-stamp challans contribute significantly more to the total revenue than document registration in these districts for FY 2022.

3. Is there any alteration of e-Stamp challan count and document registration count pattern since the implementation of e-Stamp challan? If so, what suggestions would you propose to the government?

[258]:
```
estamps_yearly = stamps_df.
 ↪groupby(stamps_df['fiscal_year'])[['estamps_challans_cnt','estamps_challans_rev']].
 ↪sum()
estamps_yearly = estamps_yearly.reset_index()
estamps_yearly
```

[258]:

	fiscal_year	estamps_challans_cnt	estamps_challans_rev
0	2019	0	0
1	2020	496132	27866550225
2	2021	1173978	85665322428
3	2022	1254961	107149881605

[259]:
```
documents_registered_yearly = stamps_df.
 ↪groupby(stamps_df['fiscal_year'])[['documents_registered_cnt','documents_registered_rev']].
 ↪sum()
documents_registered_yearly = documents_registered_yearly.reset_index()
documents_registered_yearly
```

[259]:

	fiscal_year	documents_registered_cnt	documents_registered_rev
0	2019	1614417	62410774226
1	2020	943893	42047114334
2	2021	1104580	84075980256
3	2022	1207073	106695404941

[260]:
```
comparison = pd.merge(estamps_yearly, documents_registered_yearly, on =␣
 ↪'fiscal_year', how = 'inner')
comparison
```

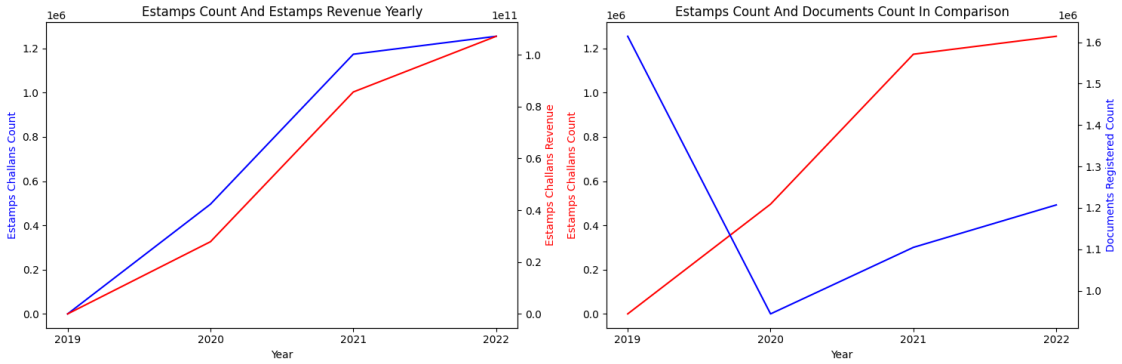
[260]:

	fiscal_year	estamps_challans_cnt	estamps_challans_rev	\
0	2019	0	0	
1	2020	496132	27866550225	
2	2021	1173978	85665322428	
3	2022	1254961	107149881605	

	documents_registered_cnt	documents_registered_rev
0	1614417	62410774226
1	943893	42047114334
2	1104580	84075980256
3	1207073	106695404941

[261]:
```
fig, (ax1, ax3) = plt.subplots(1,2, figsize=(15,5))

#Estamps Challan count and Estamps Challan revenue
ax2 = ax1.twinx()
ax1.plot(estamps_yearly['fiscal_year'], estamps_yearly['estamps_challans_cnt'],␣
 ↪color='blue')
```

```
ax2.plot(estamps_yearly['fiscal_year'], estamps_yearly['estamps_challans_rev'],␣
 ↪color='red')

#Estamps Challan cound and Document count in comparison
ax4 = ax3.twinx()
ax3.plot(comparison['fiscal_year'], comparison['estamps_challans_cnt'], color =␣
 ↪'red')
ax4.plot(comparison['fiscal_year'], comparison['documents_registered_cnt'],␣
 ↪color = 'blue')

#Title for plots
ax1.set_title('Estamps Count And Estamps Revenue Yearly')
ax1.set_xlabel('Year')
ax1.set_ylabel('Estamps Challans Count', color='blue')
ax2.set_ylabel('Estamps Challans Revenue', color='red')
ax1.set_xticks(estamps_yearly['fiscal_year'].unique())

ax3.set_title('Estamps Count And Documents Count In Comparison')
ax3.set_xlabel('Year')
ax3.set_ylabel('Estamps Challans Count', color = 'red')
ax4.set_ylabel('Documents Registered Count', color = 'blue')
ax3.set_xticks(comparison['fiscal_year'].unique())

plt.tight_layout()
plt.show()
```



From the charts above we can see that since the implementation of e-Stamp challan in 2020: - The count of e-Stamp challans has been increasing year by year. - The revenue from e-Stamp challans has also been increasing significantly. - The count of documents registered decreased in 2020 compared to 2019 but started increasing again in the following years. - The revenue from documents registered also shows a similar pattern.

Based on these observations: - It seems that the implementation of e-Stamp challan has been successful and well-received as indicated by the increasing counts and revenues. - However, it's

important to ensure that document registration does not decrease as it's also a significant source of revenue.

As for suggestions to the government: - Promote Awareness: Continue to promote awareness about the benefits and convenience of e-Stamp challans to encourage more people to use this service. - Improve Accessibility: Ensure that the process of obtaining e-Stamp challans is easy and accessible to everyone across all districts. - Monitor Trends: Keep monitoring the trends in both e-Stamp challan and document registration counts and revenues to identify any potential issues or opportunities for improvement. - Maintain Balance: While promoting e-Stamp challans, also ensure that document registration services are not neglected. Both services are important and contribute significantly to the revenue

4. Categorize districts into three segments based on their stamp registration revenue generation during the fiscal year 2021 to 2022.

[262]:
```
stamps_df_2021_2022 = stamps_df[stamps_df['fiscal_year'].isin([2021,2022])]
stamps_df_2021_2022['estamps_challans_rev'].describe()
```

[262]:
```
count    7.360000e+02
mean     2.619772e+08
std      6.474970e+08
min      7.237980e+05
25%      2.633567e+07
50%      4.749806e+07
75%      1.221248e+08
max      4.294273e+09
Name: estamps_challans_rev, dtype: float64
```

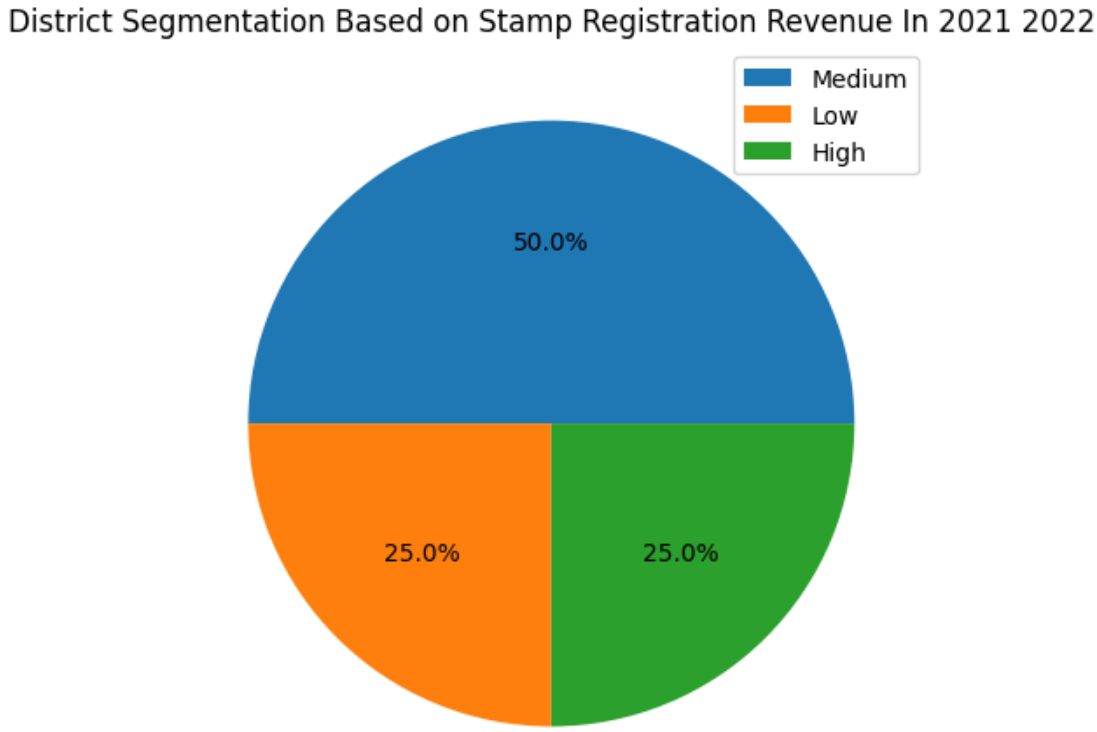
[263]:
```
q1 = stamps_df_2021_2022['estamps_challans_rev'].quantile(0.25)
q2 = stamps_df_2021_2022['estamps_challans_rev'].quantile(0.50)
q3 = stamps_df_2021_2022['estamps_challans_rev'].quantile(0.75)
```

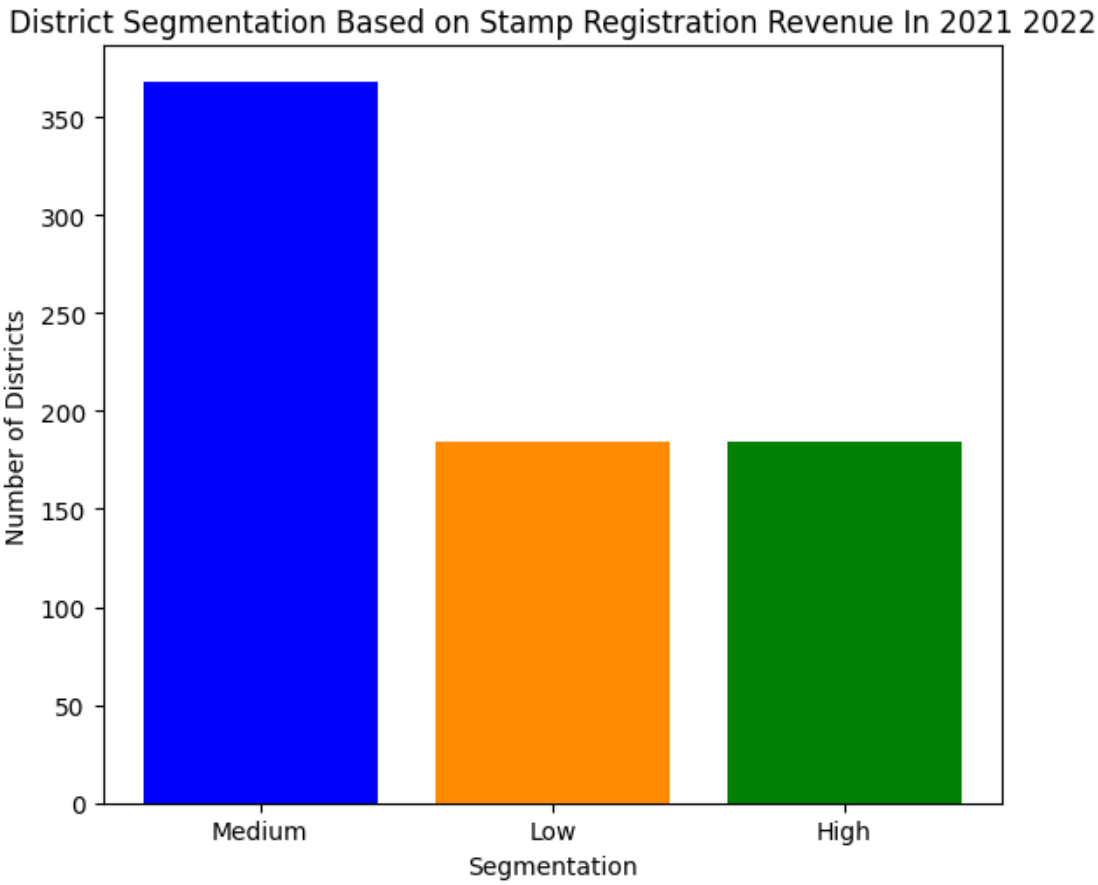
[264]:
```
stamps_df_2021_2022['segment'] = pd.
 ↪cut(stamps_df_2021_2022['estamps_challans_rev'], [0, q1, q3, float('inf')],␣
 ↪labels=['Low', 'Medium', 'High'])
segment_counts = stamps_df_2021_2022['segment'].value_counts()
segment_counts
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_856\1692301134.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  stamps_df_2021_2022['segment'] =
pd.cut(stamps_df_2021_2022['estamps_challans_rev'], [0, q1, q3, float('inf')],
labels=['Low', 'Medium', 'High'])
```

[264]:
```
Medium    368
Low       184
High      184
Name: segment, dtype: int64
```

[265]:
```
fig, ax = plt.subplots(2,1, figsize = (6,10))

ax[0].pie(segment_counts, autopct='%1.1f%%')
ax[0].set_title('District Segmentation Based on Stamp Registration Revenue In␣
 ↪2021 2022')
ax[0].legend(labels = segment_counts.index)

ax[1].bar(segment_counts.index, segment_counts.values, color = ['blue',␣
 ↪'darkorange','green'])
ax[1].set_title('District Segmentation Based on Stamp Registration Revenue In␣
 ↪2021 2022')
ax[1].set_xlabel('Segmentation')
ax[1].set_ylabel('Number of Districts')

plt.tight_layout()
plt.show()
```


District Segmentation Based on Stamp Registration Revenue In 2021 2022
Medium
Low
High
50.0%
25.0%
25.0%


District Segmentation Based on Stamp Registration Revenue In 2021 2022
Number of Districts
0
50
100
150
200
250
300
350
Medium
Low
High
Segmentation

The districts have been categorized into three segments based on their stamp registration revenue generation during the fiscal year 2021 to 2022. The segments are as follows:

- **High:** 184 districts
- **Medium:** 368 districts
- **Low:** 184 districts

1.4.2 Transportation

5. 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)

- Analyzing total sales Fuel-Typed in 2019:

[266]:
```
grouped_2019_by_quar_month = transport_df[transport_df['fiscal_year'] == 2019].
↪groupby(['quarter','Mmm']).sum()
grouped_2019_by_quar_month['total_sales_fuel_typed'] = (
    grouped_2019_by_quar_month['fuel_type_petrol'] +
    grouped_2019_by_quar_month['fuel_type_diesel'] +
    grouped_2019_by_quar_month['fuel_type_electric'] +
    grouped_2019_by_quar_month['fuel_type_others'])
grouped_2019_by_quar_month = grouped_2019_by_quar_month.reset_index()
grouped_2019_by_quar_month.head()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_856\532430309.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  grouped_2019_by_quar_month = transport_df[transport_df['fiscal_year'] ==
2019].groupby(['quarter','Mmm']).sum()
```

[266]:

	quarter	Mmm	fiscal_year	fuel_type_petrol	fuel_type_diesel	\
0	Q1	Apr	60570	140694	23098	
1	Q1	Jun	60570	141494	29331	
2	Q1	May	60570	143436	25605	
3	Q2	Aug	60570	126787	23525	
4	Q2	Jul	60570	131449	23550	

	fuel_type_electric	fuel_type_others	vehicleClass_MotorCycle	\
0	477	1742	131322	
1	328	1967	131213	
2	287	1998	132566	
3	256	2783	116938	
4	383	2717	123110	

```
   vehicleClass_MotorCar  vehicleClass_AutoRickshaw  vehicleClass_Agriculture  \
0                  17868                       4821                      1997
1                  18434                       5795                      4112
2                  18914                       5054                      2784
3                  17768                       6792                      1281
4                  15159                       6740                      2158

   vehicleClass_others  seatCapacity_1_to_3  seatCapacity_4_to_6  \
0                10003               140624                22048
1                13566               144697                23320
2                12008               143843                23211
3                10572               125607                24175
4                10930               132811                21907

   seatCapacity_above_6  Brand_new_vehicles  Pre-owned_vehicles  \
0                  3296              158951                7060
1                  4873              165108                8012
2                  4181              163618                7708
3                  3516              145140                8211
4                  3282              149925                8174

   category_Non-Transport  category_Transport  total_sales_fuel_typed
0                  151922               14089                  166011
1                  154501               18619                  173120
2                  155076               16250                  171326
3                  136555               16796                  153351
4                  141025               17074                  158099
```

[267]:
```
pivot_2019_by_month = pd.pivot_table(grouped_2019_by_quar_month, index = 'Mmm',␣
↪values = 'total_sales_fuel_typed')
pivot_2019_by_month = pivot_2019_by_month.reset_index()
```

[268]:
```
months_order = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',␣
↪'Oct', 'Nov', 'Dec']
pivot_2019_by_month['Mmm'] = pd.Categorical(pivot_2019_by_month['Mmm'],␣
↪categories=months_order, ordered=True)
pivot_2019_by_month = pivot_2019_by_month.sort_values('Mmm')
pivot_2019_by_month
```

[268]:
```
   Mmm  total_sales_fuel_typed
4  Jan                  162447
3  Feb                  184469
7  Mar                  155629
0  Apr                  166011
8  May                  171326
6  Jun                  173120
```

```
5    Jul                  158099
1    Aug                  153351
11   Sep                  123221
10   Oct                  261064
9    Nov                  180614
2    Dec                  154220
```

[269]:
```
pivot_2019_by_quar = pd.pivot_table(grouped_2019_by_quar_month, index =␣
↪'quarter', values = 'total_sales_fuel_typed')
pivot_2019_by_quar = pivot_2019_by_quar.reset_index()
pivot_2019_by_quar
```

[269]:
```
  quarter  total_sales_fuel_typed
0      Q1           170152.333333
1      Q2           144890.333333
2      Q3           198632.666667
3      Q4           167515.000000
```

- Analyzing total sales Fuel-Typed in 2020:

[270]:
```
grouped_2020_by_quar_month = transport_df[transport_df['fiscal_year'] == 2020].
↪groupby(['quarter','Mmm']).sum()
grouped_2020_by_quar_month['total_sales_fuel_typed'] = (
    grouped_2020_by_quar_month['fuel_type_petrol'] +
    grouped_2020_by_quar_month['fuel_type_diesel'] +
    grouped_2020_by_quar_month['fuel_type_electric'] +
    grouped_2020_by_quar_month['fuel_type_others'])
grouped_2020_by_quar_month = grouped_2020_by_quar_month.reset_index()
grouped_2020_by_quar_month.head()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_856\3824301240.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  grouped_2020_by_quar_month = transport_df[transport_df['fiscal_year'] ==
2020].groupby(['quarter','Mmm']).sum()
```

[270]:
```
  quarter  Mmm  fiscal_year  fuel_type_petrol  fuel_type_diesel  \
0      Q1  Apr        60600               341              1652
1      Q1  Jun        60600            140052             18478
2      Q1  May        60600             58963              9491
3      Q2  Aug        60600            124697             13220
4      Q2  Jul        60600            119936             16687

   fuel_type_electric  fuel_type_others  vehicleClass_MotorCycle  \
0                   0                 1                      337
1                 177               137                   127355
```

```
2                  50                62                  54769
3                 240               239                 108038
4                 253               205                 105946

   vehicleClass_MotorCar  vehicleClass_AutoRickshaw  vehicleClass_Agriculture  \
0                      6                          3                       952
1                  15406                        868                      7628
2                   4848                        248                      4809
3                  19577                        561                      3441
4                  16946                        764                      5357

   vehicleClass_others  seatCapacity_1_to_3  seatCapacity_4_to_6  \
0                  696                 1981                    9
1                 7587               141939                15153
2                 3892                63266                 4722
3                 6779               117615                18516
4                 8068               118730                16319

   seatCapacity_above_6  Brand_new_vehicles  Pre-owned_vehicles  \
0                     4                1980                  14
1                  1752              151995                6849
2                   578               66335                2231
3                  2253              130997                7399
4                  2030              130381                6700

   category_Non-Transport  category_Transport  total_sales_fuel_typed
0                    1332                 662                    1994
1                  150848                7996                  158844
2                   64677                3889                   68566
3                  131690                6706                  138396
4                  128991                8090                  137081
```

[271]:
```
pivot_2020_by_month = pd.pivot_table(grouped_2020_by_quar_month, index = 'Mmm',␣
↪values = 'total_sales_fuel_typed')
pivot_2020_by_month = pivot_2020_by_month.reset_index()
```

[272]:
```
months_order = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',␣
↪'Oct', 'Nov', 'Dec']
pivot_2020_by_month['Mmm'] = pd.Categorical(pivot_2020_by_month['Mmm'],␣
↪categories=months_order, ordered=True)
pivot_2020_by_month = pivot_2020_by_month.sort_values('Mmm')
pivot_2020_by_month
```

[272]:
```
   Mmm  total_sales_fuel_typed
4  Jan                  158645
3  Feb                  152989
7  Mar                  155028
```

```
0    Apr                   1994
8    May                  68566
6    Jun                 158844
5    Jul                 137081
1    Aug                 138396
11   Sep                 139619
10   Oct                 203828
9    Nov                 190147
2    Dec                 153072
```

[273]:
```
pivot_2020_by_quar = pd.pivot_table(grouped_2020_by_quar_month, index =␣
↪'quarter', values = 'total_sales_fuel_typed')
pivot_2020_by_quar = pivot_2020_by_quar.reset_index()
pivot_2020_by_quar
```

[273]:
```
  quarter  total_sales_fuel_typed
0      Q1            76468.000000
1      Q2           138365.333333
2      Q3           182349.000000
3      Q4           155554.000000
```

- Analyzing total sales Fuel-Typed in 2021:

[274]:
```
grouped_2021_by_quar_month = transport_df[transport_df['fiscal_year'] == 2021].
↪groupby(['quarter','Mmm']).sum()
grouped_2021_by_quar_month['total_sales_fuel_typed'] = (
    grouped_2021_by_quar_month['fuel_type_petrol'] +
    grouped_2021_by_quar_month['fuel_type_diesel'] +
    grouped_2021_by_quar_month['fuel_type_electric'] +
    grouped_2021_by_quar_month['fuel_type_others'])
grouped_2021_by_quar_month = grouped_2021_by_quar_month.reset_index()
grouped_2021_by_quar_month.head()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_856\2052142697.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  grouped_2021_by_quar_month = transport_df[transport_df['fiscal_year'] ==
2021].groupby(['quarter','Mmm']).sum()
```

[274]:
```
  quarter  Mmm  fiscal_year  fuel_type_petrol  fuel_type_diesel  \
0      Q1  Apr        60630            107965             17006
1      Q1  Jun        60630            113917             21934
2      Q1  May        60630             57942             10888
3      Q2  Aug        60630            118016             16013
4      Q2  Jul        60630            117253             19452
```

```
   fuel_type_electric  fuel_type_others  vehicleClass_MotorCycle  \
0                 938               845                    88954
1                1336               422                    97573
2                 351               284                    47877
3                1970              1538                    96739
4                1931              1205                    97707

   vehicleClass_MotorCar  vehicleClass_AutoRickshaw  vehicleClass_Agriculture  \
0                  24216                       1066                      4482
1                  21029                        479                      9593
2                  13185                        199                      4021
3                  28710                       1481                      3083
4                  26240                       1051                      5561

   vehicleClass_others  seatCapacity_1_to_3  seatCapacity_4_to_6  \
0                 8036                100413                23601
1                 8935                115255                20001
2                 4183                 55557                12710
3                 7528                106498                27539
4                 9282                111278                24944

   seatCapacity_above_6  Brand_new_vehicles  Pre-owned_vehicles  \
0                  2736              118576                8178
1                  2351              129332                8277
2                  1196               65067                4398
3                  3498              127472               10069
4                  3612              130057                9784

   category_Non-Transport  category_Transport  total_sales_fuel_typed
0                  118095                8659                  126754
1                  128499                9110                  137609
2                   65287                4178                   69465
3                  128989                8552                  137537
4                  129944                9897                  139841
```

[275]:
```
pivot_2021_by_month = pd.pivot_table(grouped_2021_by_quar_month, index = 'Mmm',␣
↪values = 'total_sales_fuel_typed')
pivot_2021_by_month = pivot_2021_by_month.reset_index()
```

[276]:
```
months_order = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',␣
↪'Oct', 'Nov', 'Dec']
pivot_2021_by_month['Mmm'] = pd.Categorical(pivot_2021_by_month['Mmm'],␣
↪categories=months_order, ordered=True)
pivot_2021_by_month = pivot_2021_by_month.sort_values('Mmm')
pivot_2021_by_month
```

[276]:

	Mmm	total_sales_fuel_typed
4	Jan	115575
3	Feb	131554
7	Mar	138385
0	Apr	126754
8	May	69465
6	Jun	137609
5	Jul	139841
1	Aug	137537
11	Sep	109592
10	Oct	173020
9	Nov	143488
2	Dec	127485

[277]:
```
pivot_2021_by_quar = pd.pivot_table(grouped_2021_by_quar_month, index =␣
↪'quarter', values = 'total_sales_fuel_typed')
pivot_2021_by_quar = pivot_2021_by_quar.reset_index()
pivot_2021_by_quar
```

[277]:

	quarter	total_sales_fuel_typed
0	Q1	111276.000000
1	Q2	128990.000000
2	Q3	147997.666667
3	Q4	128504.666667

- Analyzing total sales Fuel-Typed in 2022:

[278]:
```
grouped_2022_by_quar_month = transport_df[transport_df['fiscal_year'] == 2022].
↪groupby(['quarter','Mmm']).sum()
grouped_2022_by_quar_month['total_sales_fuel_typed'] = (
    grouped_2022_by_quar_month['fuel_type_petrol'] +
    grouped_2022_by_quar_month['fuel_type_diesel'] +
    grouped_2022_by_quar_month['fuel_type_electric'] +
    grouped_2022_by_quar_month['fuel_type_others'])
grouped_2022_by_quar_month = grouped_2022_by_quar_month.reset_index()
grouped_2022_by_quar_month.head()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_856\3627280342.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  grouped_2022_by_quar_month = transport_df[transport_df['fiscal_year'] ==
2022].groupby(['quarter','Mmm']).sum()
```

[278]:

	quarter	Mmm	fiscal_year	fuel_type_petrol	fuel_type_diesel	\
0	Q1	Apr	60660	109884	17983	
1	Q1	Jun	60660	113121	23838	

```
2      Q1  May        60660            104266             19572
3      Q2  Aug        60660            112693             16524
4      Q2  Jul        60660             97171             19582

   fuel_type_electric  fuel_type_others  vehicleClass_MotorCycle  \
0                5712              2385                    95782
1                3956              2926                    94599
2                3696              2304                    86691
3                6711              3839                    95882
4                4425              2793                    80980

   vehicleClass_MotorCar  vehicleClass_AutoRickshaw  vehicleClass_Agriculture  \
0                  25142                       1697                      2417
1                  26659                       2137                      5487
2                  24642                       1799                      3702
3                  28016                       2952                      1958
4                  25062                       2086                      3326

   vehicleClass_others  seatCapacity_1_to_3  seatCapacity_4_to_6  \
0                10934               107353                24780
1                14961               110832                27237
2                13008               100677                24608
3                10969               105162                29706
4                12517                93294                25614

   seatCapacity_above_6  Brand_new_vehicles  Pre-owned_vehicles  \
0                  3803              125111               10861
1                  5603              130625               13218
2                  4443              119272               10570
3                  4710              125355               14422
4                  4795              111350               12621

   category_Non-Transport  category_Transport  total_sales_fuel_typed
0                  123883               12089                  135964
1                  127259               16584                  143841
2                  115491               14351                  129838
3                  126238               13539                  139767
4                  109736               14235                  123971
```

[279]:
```
pivot_2022_by_month = pd.pivot_table(grouped_2022_by_quar_month, index = 'Mmm',␣
↪values = 'total_sales_fuel_typed')
pivot_2022_by_month = pivot_2022_by_month.reset_index()
pivot_2022_by_month
```

[279]:
```
   Mmm  total_sales_fuel_typed
0  Apr                  135964
1  Aug                  139767
```

```
2   Dec                  73214
3   Feb                  66431
4   Jan                 127824
5   Jul                 123971
6   Jun                 143841
7   Mar                 151416
8   May                 129838
9   Nov                  72127
10  Oct                 110900
11  Sep                 138399
```

[280]:
```
months_order = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',␣
 ↪'Oct', 'Nov', 'Dec']
pivot_2022_by_month['Mmm'] = pd.Categorical(pivot_2022_by_month['Mmm'],␣
 ↪categories=months_order, ordered=True)
pivot_2022_by_month = pivot_2022_by_month.sort_values('Mmm')
```

[281]:
```
pivot_2022_by_quar = pd.pivot_table(grouped_2022_by_quar_month, index =␣
 ↪'quarter', values = 'total_sales_fuel_typed')
pivot_2022_by_quar = pivot_2022_by_quar.reset_index()
pivot_2022_by_quar
```

[281]:
```
  quarter  total_sales_fuel_typed
0      Q1           136547.666667
1      Q2           134045.666667
2      Q3            85413.666667
3      Q4           115223.666667
```

- Analyzing total sales Fuel-Typed yearly:

[282]:
```
grouped_by_yearly = transport_df.groupby(['fiscal_year']).sum()
grouped_by_yearly['total_sales_fuel_typed'] = (
     grouped_by_yearly['fuel_type_petrol'] +
     grouped_by_yearly['fuel_type_diesel'] +
     grouped_by_yearly['fuel_type_electric'] +
    grouped_by_yearly['fuel_type_others'])
grouped_by_yearly = grouped_by_yearly.reset_index()
grouped_by_yearly.head()
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_856\4130530342.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  grouped_by_yearly = transport_df.groupby(['fiscal_year']).sum()
```

[282]:
```
   fiscal_year  fuel_type_petrol  fuel_type_diesel  fuel_type_electric  \
0         2019           1669884            341790                4143
```

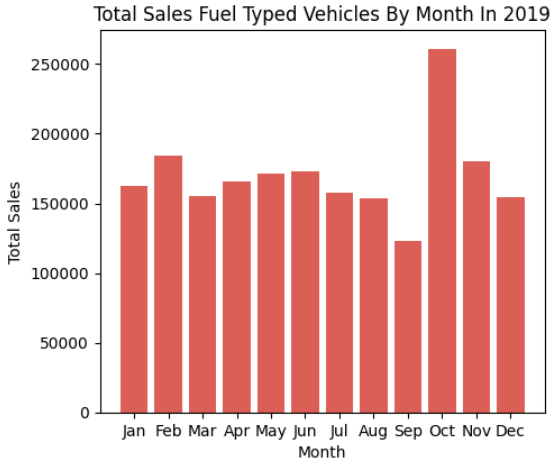
```
1          2020           1433801            212145                5980
2          2021           1300723            199346               31004
3          2022           1118857            196103               62532

   fuel_type_others  vehicleClass_MotorCycle  vehicleClass_MotorCar  \
0             27754                  1524671                 233632
1              6283                  1232661                 246120
2             19232                  1097546                 288537
3             36200                   955109                 268141

   vehicleClass_AutoRickshaw  vehicleClass_Agriculture  vehicleClass_others  \
0                      75003                     45797               164462
1                      11919                     64991               102580
2                      17524                     49754                96951
3                      30699                     35172               124639

   seatCapacity_1_to_3  seatCapacity_4_to_6  seatCapacity_above_6  \
0              1695324               304299                 43088
1              1391270               239278                 27627
2              1231864               280517                 37757
3              1082286               282856                 47322

   Brand_new_vehicles  Pre-owned_vehicles  category_Non-Transport  \
0             1936747              106824                 1811731
1             1566049               92228                 1551106
2             1440068              110249                 1441048
3             1274737              139033                 1262986

   category_Transport  total_sales_fuel_typed
0              231840                 2043571
1              107171                 1658209
2              109269                 1550305
3              150784                 1413692
```

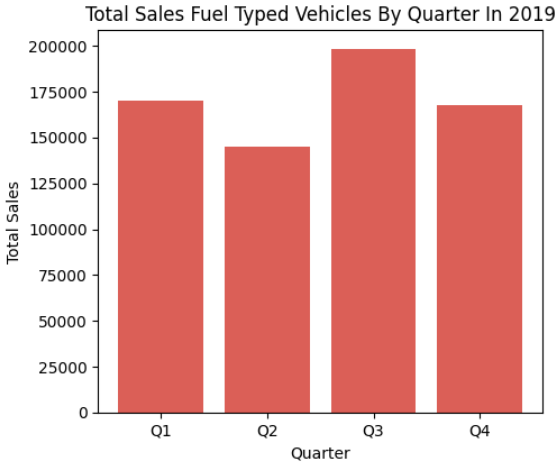
- Visualizing monthly total sales and quarterly total sales from 2019 to 2022:

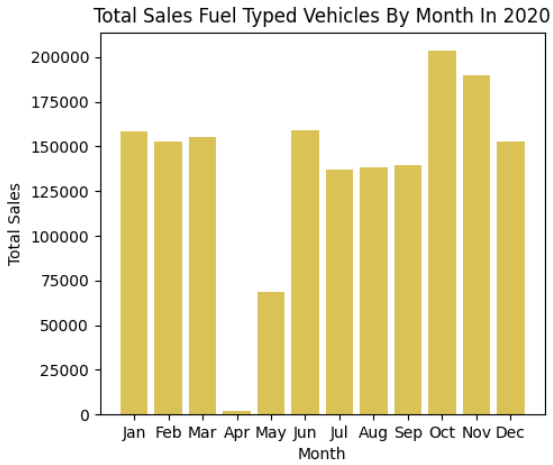
[283]:
```
fig , ax = plt.subplots(4, 2, figsize =(10,17))

#colors for plot
color_0 = sns.color_palette('hls', 8)[0]
color_1 = sns.color_palette('hls', 8)[1]
color_2 = sns.color_palette('hls', 8)[2]
color_3 = sns.color_palette('hls', 8)[3]

#Total sales monthly and quarterly in 2019
ax[0,0].bar(pivot_2019_by_month['Mmm'],␣
 ↪pivot_2019_by_month['total_sales_fuel_typed'], color = color_0)
```

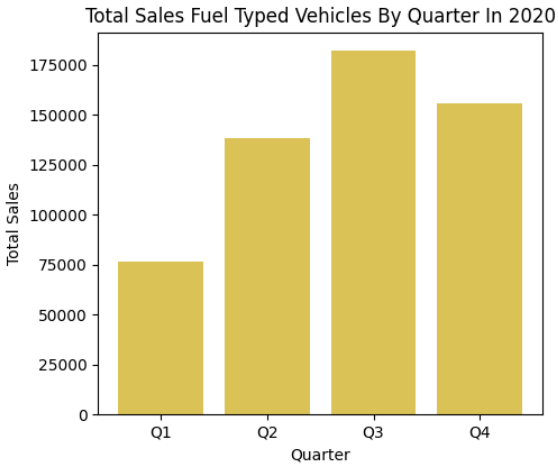
```
ax[0,1].bar(pivot_2019_by_quar['quarter'],␣
 ↪pivot_2019_by_quar['total_sales_fuel_typed'], color = color_0)

#Total sales monthly and quarterly in 2020
ax[1,0].bar(pivot_2020_by_month['Mmm'],␣
 ↪pivot_2020_by_month['total_sales_fuel_typed'], color = color_1)
ax[1,1].bar(pivot_2020_by_quar['quarter'],␣
 ↪pivot_2020_by_quar['total_sales_fuel_typed'], color = color_1)

#Total sales monthly and quarterly in 2021
ax[2,0].bar(pivot_2021_by_month['Mmm'],␣
 ↪pivot_2021_by_month['total_sales_fuel_typed'], color = color_2)
ax[2,1].bar(pivot_2021_by_quar['quarter'],␣
 ↪pivot_2021_by_quar['total_sales_fuel_typed'], color = color_2)

#Total sales monthly and quarterly in 2022
ax[3,0].bar(pivot_2022_by_month['Mmm'],␣
 ↪pivot_2022_by_month['total_sales_fuel_typed'], color = color_3)
ax[3,1].bar(pivot_2022_by_quar['quarter'],␣
 ↪pivot_2022_by_quar['total_sales_fuel_typed'], color = color_3)


#Title for plots
#2019
ax[0,0].set_title('Total Sales Fuel Typed Vehicles By Month In 2019')
ax[0,0].set_xlabel('Month')
ax[0,0].set_ylabel('Total Sales')

ax[0,1].set_title('Total Sales Fuel Typed Vehicles By Quarter In 2019')
ax[0,1].set_xlabel('Quarter')
ax[0,1].set_ylabel('Total Sales')

#2020
ax[1,0].set_title('Total Sales Fuel Typed Vehicles By Month In 2020')
ax[1,0].set_xlabel('Month')
ax[1,0].set_ylabel('Total Sales')

ax[1,1].set_title('Total Sales Fuel Typed Vehicles By Quarter In 2020')
ax[1,1].set_xlabel('Quarter')
ax[1,1].set_ylabel('Total Sales')

#2021
ax[2,0].set_title('Total Sales Fuel Typed Vehicles By Month In 2021')
ax[2,0].set_xlabel('Month')
ax[2,0].set_ylabel('Total Sales')
```

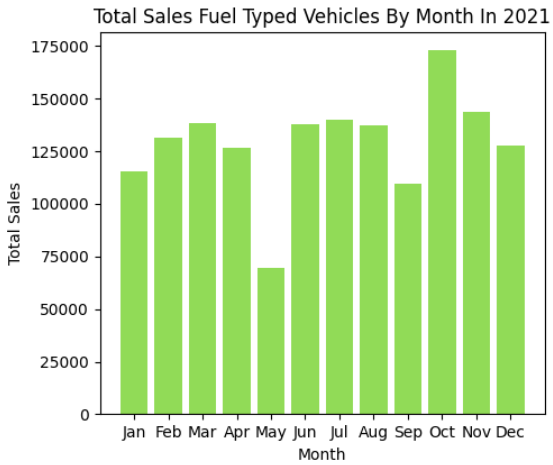
```
ax[2,1].set_title('Total Sales Fuel Typed Vehicles By Quarter In 2021')
ax[2,1].set_xlabel('Quarter')
ax[2,1].set_ylabel('Total Sales')

#2022
ax[3,0].set_title('Total Sales Fuel Typed Vehicles By Month In 2022')
ax[3,0].set_xlabel('Month')
ax[3,0].set_ylabel('Total Sales')

ax[3,1].set_title('Total Sales Fuel Typed Vehicles By Quarter In 2022')
ax[3,1].set_xlabel('Quarter')
ax[3,1].set_ylabel('Total Sales')


plt.tight_layout()
plt.show()
```

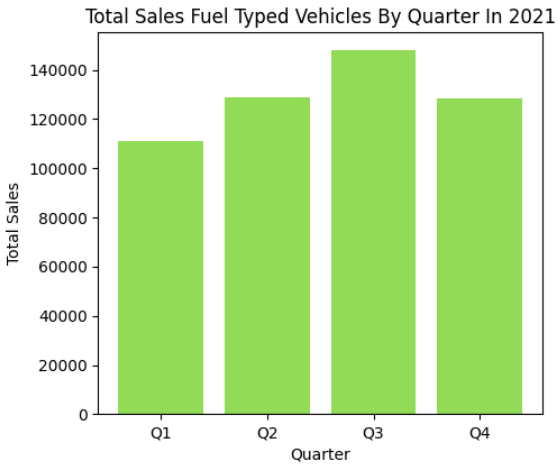

Total Sales Fuel Typed Vehicles By Month In 2019
Total Sales
Month
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
0 50000 100000 150000 200000 250000

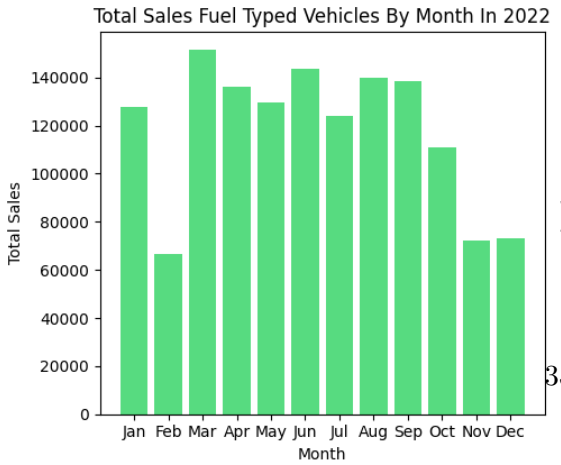

Total Sales Fuel Typed Vehicles By Quarter In 2019
Total Sales
Quarter
Q1 Q2 Q3 Q4
0 25000 50000 75000 100000 125000 150000 175000 200000

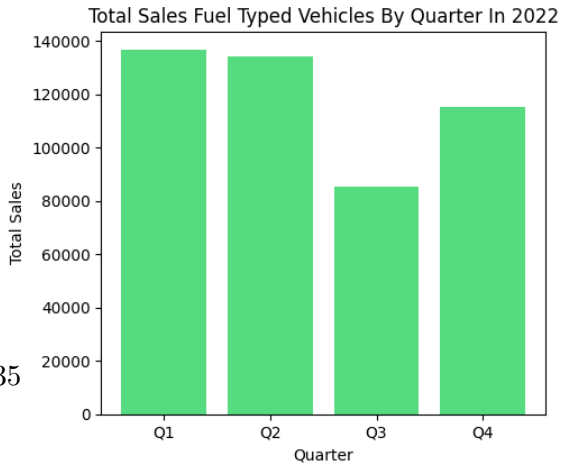

Total Sales Fuel Typed Vehicles By Month In 2020
Total Sales
Month
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
0 25000 50000 75000 100000 125000 150000 175000 200000


Total Sales Fuel Typed Vehicles By Quarter In 2020
Total Sales
Quarter
Q1 Q2 Q3 Q4
0 25000 50000 75000 100000 125000 150000 175000


Total Sales Fuel Typed Vehicles By Month In 2021
Total Sales
Month
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
0 25000 50000 75000 100000 125000 150000 175000

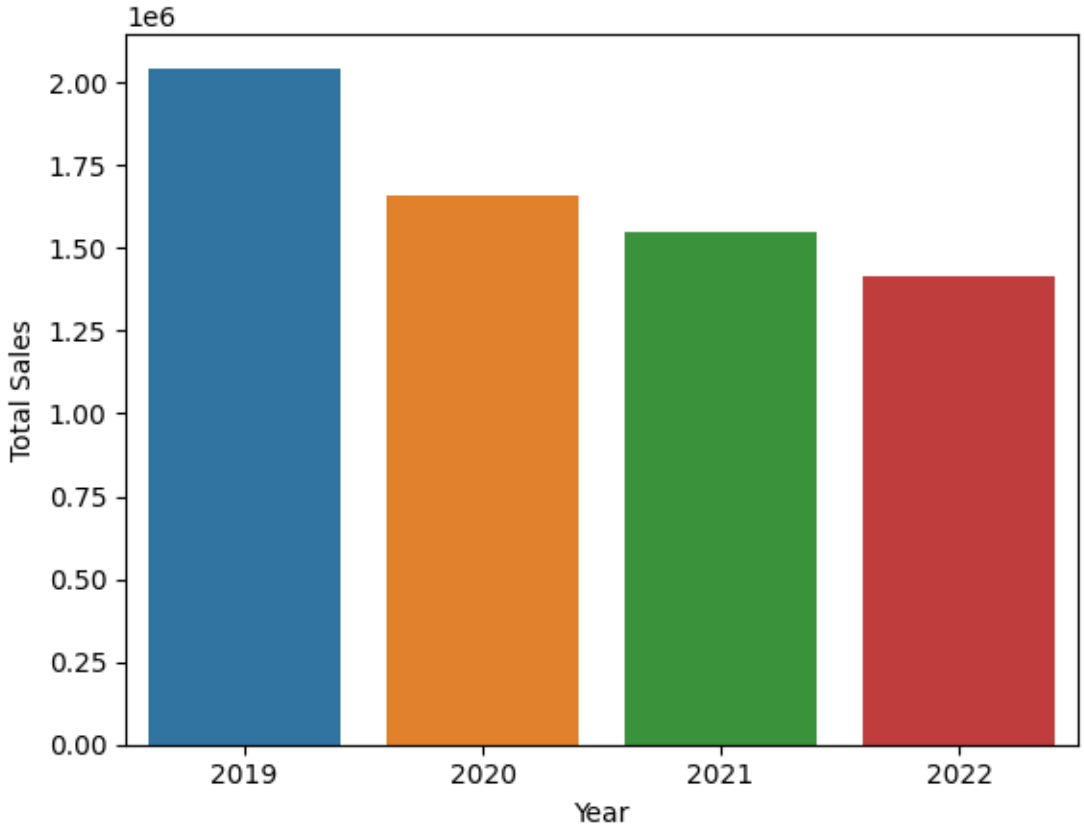

Total Sales Fuel Typed Vehicles By Quarter In 2021
Total Sales
Quarter
Q1 Q2 Q3 Q4
0 20000 40000 60000 80000 100000 120000 140000


Total Sales Fuel Typed Vehicles By Month In 2022
Total Sales
Month
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
0 20000 40000 60000 80000 100000 120000 140000


Total Sales Fuel Typed Vehicles By Quarter In 2022
Total Sales
Quarter
Q1 Q2 Q3 Q4
0 20000 40000 60000 80000 100000 120000 140000

- Visualizing total sales Fuel-Typed yearly:

[284]:
```
sns.barplot(x = grouped_by_yearly['fiscal_year'], y =␣
 ↪grouped_by_yearly['total_sales_fuel_typed'])
plt.xlabel('Year')
plt.ylabel('Total Sales')
plt.show()
```



Monthly Sales: - In 2019, the highest sales were observed in October (261,064) and the lowest in September (123,221). - In 2020, the highest sales were observed in October (203,828) and the lowest in April (1,994). - In 2021, the highest sales were observed in October (173,020) and the lowest in May (69,465). - In 2022, the highest sales were observed in March (151,416) and the lowest in February (66,431).

Quarterly Sales: - In all years from 2019 to 2022, the first quarter (Q1: Jan-Mar) and third quarter (Q3: Jul-Sep) generally had higher sales compared to the second quarter (Q2: Apr-Jun) and fourth quarter (Q4: Oct-Dec).

Yearly Sales: - The total sales of fuel-typed vehicles decreased each year from 2019 to 2022.

From these observations, it can be inferred that vehicle sales tend to be higher at the beginning and middle of the year. The drop in sales during April 2020 could be attributed to the global pandemic situation.

As for the driving factors behind these trends: 1. **Seasonal Factors**: Vehicle sales often increase during festive seasons or when new models are launched. 2. **Economic Factors**: The state of the economy can greatly influence vehicle sales. For instance, a strong economy will encourage consumers to purchase new vehicles. 3. **Policy Changes**: Government policies such as changes in tax rates or introduction of new regulations can also impact vehicle sales.

To increase vehicle sales, it would be beneficial for the government to introduce incentives during low-sales periods. Additionally, promoting fuel-efficient vehicles could attract environmentally conscious consumers.

6 . 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.

[285]:
```
transport_2022 = transport_df[transport_df['fiscal_year'] == 2022]
vehicle_class = ['vehicleClass_MotorCycle', 'vehicleClass_MotorCar',␣
 ↪'vehicleClass_AutoRickshaw', 'vehicleClass_Agriculture']
district_vehicle_df = transport_2022.groupby('district')[vehicle_class].sum()
district_vehicle_df['predominant_vehicle_class'] = district_vehicle_df.
 ↪idxmax(axis=1)
district_vehicle_df
```

[285]:

district	vehicleClass_MotorCycle	vehicleClass_MotorCar	\
Adilabad	10410	1416	
Bhadradri Kothagudem	17022	3157	
Hyderabad	206819	51447	
Jagtial	13639	2055	
Jangoan	8617	1177	
Jayashankar Bhupalpally	9225	1343	
Jogulamba Gadwal	10300	899	
Kamareddy	15184	2153	
Karimnagar	19411	4818	
Khammam	27385	5880	
Kumurambheem Asifabad	6433	479	
Mahabubabad	11046	1155	
Mahabubnagar	17977	3300	
Mancherial	11603	2218	
Medak	11663	2347	
Medchal_Malkajgiri	164626	61071	
Nagarkurnool	12229	2066	
Nalgonda	27942	4927	
Nirmal	10984	1581	
Nizamabad	28105	5824	
Peddapalli	10818	2205	

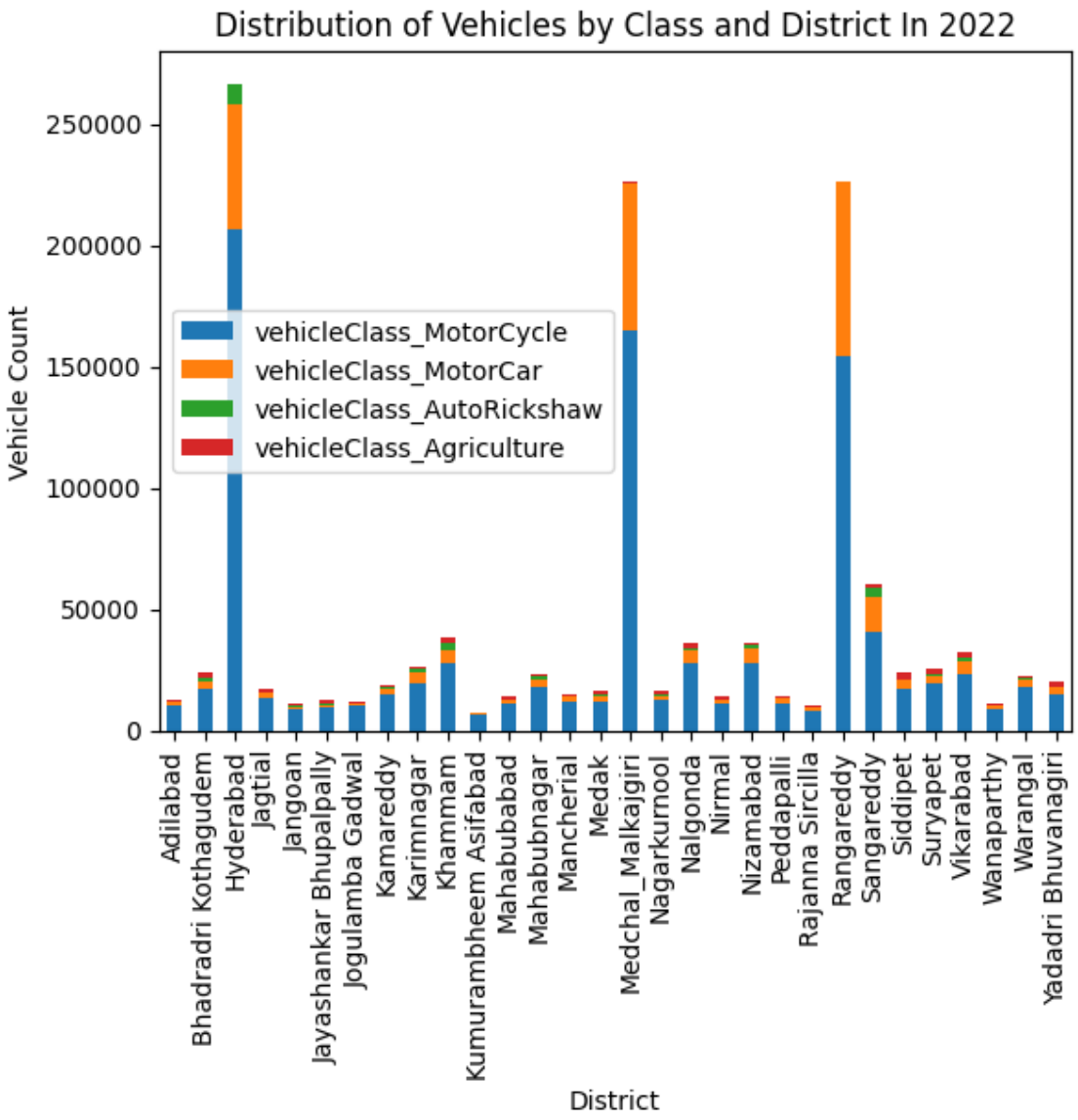
Rajanna Sircilla	8007	1376
Rangareddy	154186	71832
Sangareddy	40784	14565
Siddipet	17401	3254
Suryapet	19165	3038
Vikarabad	23116	5211
Wanaparthy	8771	1194
Warangal	17770	2937
Yadadri Bhuvanagiri	14471	3216

district	vehicleClass_AutoRickshaw	vehicleClass_Agriculture	\
Adilabad	368	467	
Bhadradri Kothagudem	1388	2013	
Hyderabad	8397	20	
Jagtial	272	858	
Jangoan	397	1225	
Jayashankar Bhupalpally	218	1928	
Jogulamba Gadwal	116	868	
Kamareddy	379	748	
Karimnagar	1042	999	
Khammam	2690	2121	
Kumurambheem Asifabad	286	160	
Mahabubabad	680	906	
Mahabubnagar	1130	937	
Mancherial	550	400	
Medak	610	1579	
Medchal_Malkajgiri	83	261	
Nagarkurnool	366	1556	
Nalgonda	964	2599	
Nirmal	295	989	
Nizamabad	1011	1181	
Peddapalli	382	705	
Rajanna Sircilla	148	675	
Rangareddy	69	449	
Sangareddy	3381	1570	
Siddipet	524	2437	
Suryapet	739	2188	
Vikarabad	2046	1814	
Wanaparthy	372	985	
Warangal	1331	681	
Yadadri Bhuvanagiri	465	1853	

district	predominant_vehicle_class
Adilabad	vehicleClass_MotorCycle
Bhadradri Kothagudem	vehicleClass_MotorCycle

```
Hyderabad                 vehicleClass_MotorCycle
Jagtial                   vehicleClass_MotorCycle
Jangoan                   vehicleClass_MotorCycle
Jayashankar Bhupalpally   vehicleClass_MotorCycle
Jogulamba Gadwal          vehicleClass_MotorCycle
Kamareddy                 vehicleClass_MotorCycle
Karimnagar                vehicleClass_MotorCycle
Khammam                   vehicleClass_MotorCycle
Kumurambheem Asifabad     vehicleClass_MotorCycle
Mahabubabad               vehicleClass_MotorCycle
Mahabubnagar              vehicleClass_MotorCycle
Mancherial                vehicleClass_MotorCycle
Medak                     vehicleClass_MotorCycle
Medchal_Malkajgiri        vehicleClass_MotorCycle
Nagarkurnool              vehicleClass_MotorCycle
Nalgonda                  vehicleClass_MotorCycle
Nirmal                    vehicleClass_MotorCycle
Nizamabad                 vehicleClass_MotorCycle
Peddapalli                vehicleClass_MotorCycle
Rajanna Sircilla          vehicleClass_MotorCycle
Rangareddy                vehicleClass_MotorCycle
Sangareddy                vehicleClass_MotorCycle
Siddipet                  vehicleClass_MotorCycle
Suryapet                  vehicleClass_MotorCycle
Vikarabad                 vehicleClass_MotorCycle
Wanaparthy                vehicleClass_MotorCycle
Warangal                  vehicleClass_MotorCycle
Yadadri Bhuvanagiri       vehicleClass_MotorCycle
```

[286]:
```
district_vehicle_df.plot(kind='bar', stacked=True)
plt.title('Distribution of Vehicles by Class and District In 2022')
plt.xlabel('District')
plt.ylabel('Vehicle Count')
plt.show()
```



The distribution of vehicles by vehicle class across different districts in FY 2022 is as follows:

- **MotorCycle**: This class of vehicles is the most predominant across all districts. The districts with the highest number of motorcycles are Hyderabad (206,819), Medchal_Malkajgiri (164,626), and Rangareddy (154,186).
- **MotorCar**: This class of vehicles is also popular but not as much as motorcycles. The districts with the highest number of motorcars are Hyderabad (51,447), Medchal_Malkajgiri (61,071), and Rangareddy (71,832).
- **AutoRickshaw**: This class of vehicles is less common compared to motorcycles and motorcars. The districts with the highest number of auto rickshaws are Hyderabad (8,397), Sangareddy (3,381), and Khammam (2,690).
- **Agriculture**: This class of vehicles is least common among the four classes. The districts

with the highest number of agriculture vehicles are Bhadradri Kothagudem (2,013), Nalgonda (2,599), and Siddipet (2,437).

From this analysis, it's clear that the MotorCycle class is the most preferred vehicle class across all districts. However, the preference for other vehicle classes like MotorCar, AutoRickshaw, and Agriculture varies from district to district. Factors influencing these preferences could include the district's economic status, road infrastructure, and local needs or customs.

7. 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)

[287]:
```
fuel_type_cols = ['fuel_type_petrol','fuel_type_diesel', 'fuel_type_electric']
sales_2021 = transport_df[transport_df['fiscal_year'] == 2021].
 ↪groupby('district')[fuel_type_cols].sum()
sales_2021['total_sales'] = (
    sales_2021['fuel_type_petrol'] +
    sales_2021['fuel_type_diesel'] +
    sales_2021['fuel_type_electric'] )
```

[288]:
```
sales_2022 = transport_df[transport_df['fiscal_year'] == 2022].
 ↪groupby('district')[fuel_type_cols].sum()
sales_2022['total_sales'] = (
    sales_2022['fuel_type_petrol'] +
    sales_2022['fuel_type_diesel'] +
    sales_2022['fuel_type_electric'] )
```

[289]:
```
sales_growth = (sales_2022 - sales_2021)/sales_2021
sales_growth
```

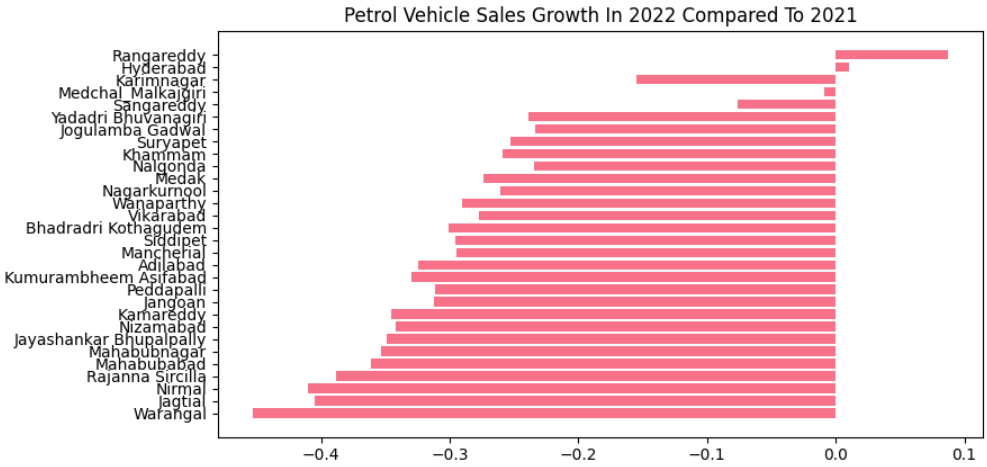
[289]:

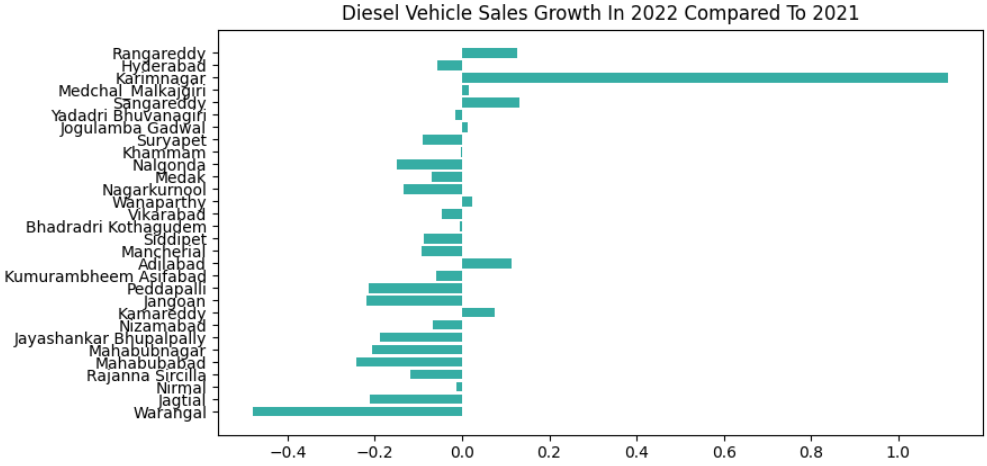
district	fuel_type_petrol	fuel_type_diesel	\
Adilabad	-0.324253	0.112925	
Bhadradri Kothagudem	-0.300868	-0.004705	
Hyderabad	0.010266	-0.055485	
Jagtial	-0.405252	-0.212744	
Jangoan	-0.312288	-0.219968	
Jayashankar Bhupalpally	-0.349253	-0.188268	
Jogulamba Gadwal	-0.233724	0.011693	
Kamareddy	-0.345884	0.074642	
Karimnagar	-0.154777	1.114422	
Khammam	-0.258599	-0.003475	
Kumurambheem Asifabad	-0.329608	-0.060523	
Mahabubabad	-0.360821	-0.242895	
Mahabubnagar	-0.353178	-0.205507	
Mancherial	-0.294770	-0.092593	
Medak	-0.274055	-0.068753	
Medchal_Malkajgiri	-0.009042	0.016168	

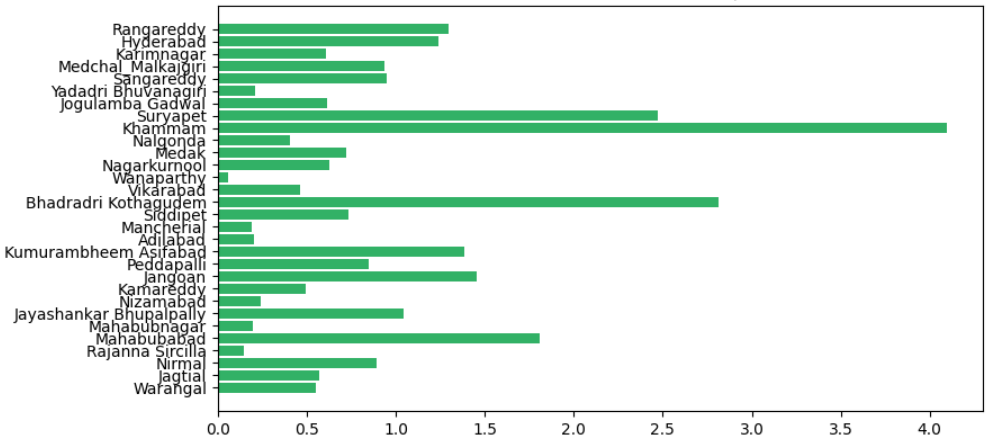
Nagarkurnool	-0.260474	-0.133635
Nalgonda	-0.234440	-0.148936
Nirmal	-0.410514	-0.014160
Nizamabad	-0.342295	-0.066632
Peddapalli	-0.311325	-0.212873
Rajanna Sircilla	-0.388045	-0.119423
Rangareddy	0.087322	0.126745
Sangareddy	-0.076069	0.132064
Siddipet	-0.295256	-0.086597
Suryapet	-0.253223	-0.089589
Vikarabad	-0.277004	-0.045884
Wanaparthy	-0.290268	0.021966
Warangal	-0.453198	-0.480322
Yadadri Bhuvanagiri	-0.239242	-0.014965

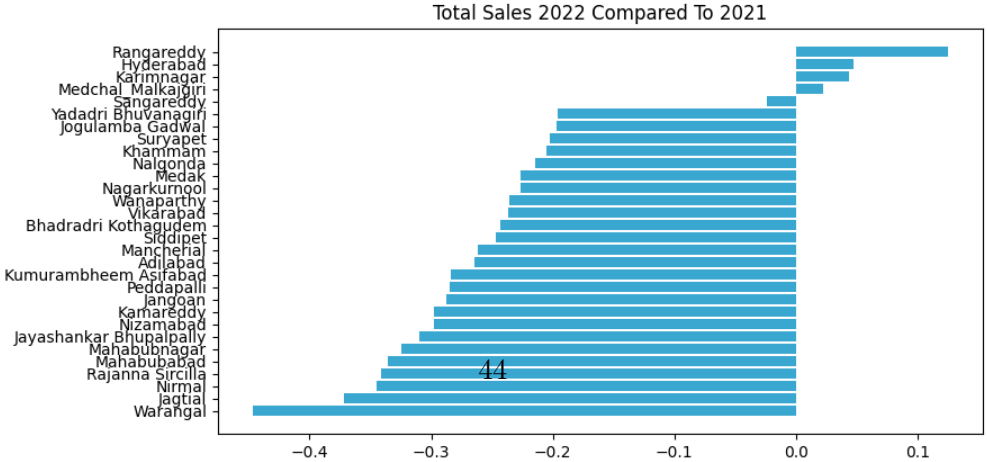
district	fuel_type_electric	total_sales
Adilabad	0.202166	-0.264841
Bhadradri Kothagudem	2.815789	-0.243000
Hyderabad	1.242804	0.047385
Jagtial	0.569682	-0.371537
Jangoan	1.452055	-0.287681
Jayashankar Bhupalpally	1.045045	-0.309701
Jogulamba Gadwal	0.615385	-0.196928
Kamareddy	0.491803	-0.297479
Karimnagar	0.607955	0.044180
Khammam	4.093960	-0.204871
Kumurambheem Asifabad	1.382353	-0.283556
Mahabubabad	1.805556	-0.336049
Mahabubnagar	0.194444	-0.324853
Mancherial	0.191919	-0.261468
Medak	0.724324	-0.226472
Medchal_Malkajgiri	0.935648	0.022199
Nagarkurnool	0.628571	-0.226863
Nalgonda	0.405063	-0.213982
Nirmal	0.892958	-0.344545
Nizamabad	0.240524	-0.298083
Peddapalli	0.844749	-0.284687
Rajanna Sircilla	0.149171	-0.341179
Rangareddy	1.299088	0.125237
Sangareddy	0.950578	-0.023889
Siddipet	0.737079	-0.247028
Suryapet	2.471503	-0.202778
Vikarabad	0.464752	-0.236598
Wanaparthy	0.054795	-0.236111
Warangal	0.548926	-0.446918
Yadadri Bhuvanagiri	0.209607	-0.195586

[290]:
```
sales_growth_2022 = sales_growth.sort_values('total_sales', ascending = True)

fig, ax = plt.subplots(4, 1, figsize=(9, 17))

#Colors
color_0 = sns.color_palette('husl', 8)
color_1 = sns.color_palette('husl', 8)
color_2 = sns.color_palette('husl', 8)

#Plots
ax[0].barh(sales_growth_2022.index, sales_growth_2022['fuel_type_petrol'],␣
 ↪color = color_0[0])
ax[1].barh(sales_growth_2022.index, sales_growth_2022['fuel_type_diesel'],␣
 ↪color = color_1[4])
ax[2].barh(sales_growth_2022.index, sales_growth_2022['fuel_type_electric'],␣
 ↪color = color_2[3])
ax[3].barh(sales_growth_2022.index, sales_growth_2022['total_sales'], color =␣
 ↪color_2[5])

#Title for plots
ax[0].set_title('Petrol Vehicle Sales Growth In 2022 Compared To 2021')
ax[1].set_title('Diesel Vehicle Sales Growth In 2022 Compared To 2021')
ax[2].set_title('Electric Vehicle Sales Growth In 2022 Compared To 2021')
ax[3].set_title('Total Sales 2022 Compared To 2021')

plt.tight_layout()
plt.show()
```

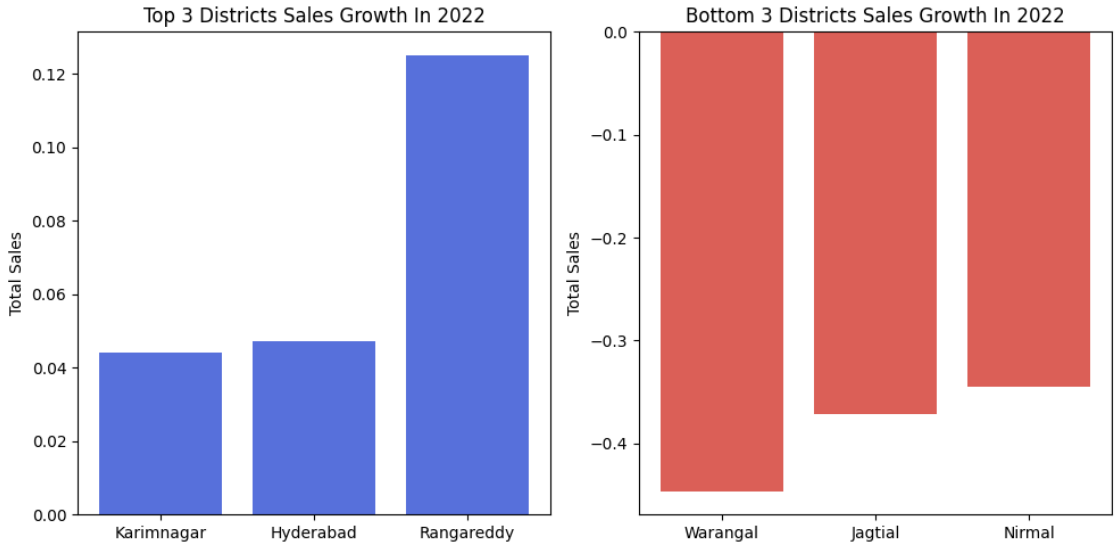

Petrol Vehicle Sales Growth In 2022 Compared To 2021
Rangareddy
Hyderabad
Karimnagar
Medchal_Malkajgiri
Sangareddy
Yadadri Bhuvanagiri
Jogulamba Gadwal
Suryapet
Khammam
Nalgonda
Medak
Nagarkurnool
Wanaparthy
Vikarabad
Bhadradri Kothagudem
Siddipet
Mancherial
Adilabad
Kumurambheem Asifabad
Peddapalli
Jangoan
Kamareddy
Nizamabad
Jayashankar Bhupalpally
Mahabubnagar
Mahabubabad
Rajanna Sircilla
Nirmal
Jagtial
Warangal
−0.4
−0.3
−0.2
−0.1
0.0
0.1


Diesel Vehicle Sales Growth In 2022 Compared To 2021
Rangareddy
Hyderabad
Karimnagar
Medchal_Malkajgiri
Sangareddy
Yadadri Bhuvanagiri
Jogulamba Gadwal
Suryapet
Khammam
Nalgonda
Medak
Nagarkurnool
Wanaparthy
Vikarabad
Bhadradri Kothagudem
Siddipet
Mancherial
Adilabad
Kumurambheem Asifabad
Peddapalli
Jangoan
Kamareddy
Nizamabad
Jayashankar Bhupalpally
Mahabubnagar
Mahabubabad
Rajanna Sircilla
Nirmal
Jagtial
Warangal
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


Electric Vehicle Sales Growth In 2022 Compared To 2021
Rangareddy
Hyderabad
Karimnagar
Medchal_Malkajgiri
Sangareddy
Yadadri Bhuvanagiri
Jogulamba Gadwal
Suryapet
Khammam
Nalgonda
Medak
Nagarkurnool
Wanaparthy
Vikarabad
Bhadradri Kothagudem
Siddipet
Mancherial
Adilabad
Kumurambheem Asifabad
Peddapalli
Jangoan
Kamareddy
Nizamabad
Jayashankar Bhupalpally
Mahabubnagar
Mahabubabad
Rajanna Sircilla
Nirmal
Jagtial
Warangal
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Total Sales 2022 Compared To 2021
Rangareddy
Hyderabad
Karimnagar
Medchal_Malkajgiri
Sangareddy
Yadadri Bhuvanagiri
Jogulamba Gadwal
Suryapet
Khammam
Nalgonda
Medak
Nagarkurnool
Wanaparthy
Vikarabad
Bhadradri Kothagudem
Siddipet
Mancherial
Adilabad
Kumurambheem Asifabad
Peddapalli
Jangoan
Kamareddy
Nizamabad
Jayashankar Bhupalpally
Mahabubnagar
Mahabubabad
Rajanna Sircilla
Nirmal
Jagtial
Warangal
−0.4
−0.3
−0.2
−0.1
0.0
0.1

[291]:
```
top_3_districts = sales_growth_2022.tail(3)
bottom_3_districts = sales_growth_2022.head(3)

fig, ax = plt.subplots(1, 2, figsize=(10, 5))
color_0 = sns.color_palette('hls', 8)[0]
color_1 = sns.color_palette('hls', 8)[5]
# Plot data for top 3 districts
ax[0].bar(top_3_districts.index, top_3_districts['total_sales'], color =␣
 ↪color_1)
ax[0].set_title('Top 3 Districts Sales Growth In 2022')
ax[0].set_ylabel('Total Sales')

# Plot data for bottom 3 districts
ax[1].bar( bottom_3_districts.index, bottom_3_districts['total_sales'], color =␣
 ↪color_0)
ax[1].set_title('Bottom 3 Districts Sales Growth In 2022')
ax[1].set_ylabel('Total Sales')

plt.tight_layout()
plt.show()
```



Top 3 Districts (Highest Growth) 1. **Rangareddy**: The total sales growth was approximately **12.52%**. The growth in petrol, diesel, and electric vehicles was about **8.73%**, **12.67%**, and **129.91%** respectively. 2. **Hyderabad**: The total sales growth was approximately **4.74%**. The growth in petrol, diesel, and electric vehicles was about **1.03%**, **-5.55%**, and **124.28%** respectively. 3. **Karimnagar**: The total sales growth was approximately **2.22%**. The growth in petrol, diesel,

and electric vehicles was about **-0.90%**, **1.62%**, and **93.56%** respectively.

Bottom 3 Districts (Lowest Growth) 1. **Warangal**: The total sales growth was approximately **-44.69%**. The growth in petrol, diesel, and electric vehicles was about **-45.32%**, **-48.03%**, and **54.89%** respectively. 2. **Jagtial**: The total sales growth was approximately **-37.15%**. The growth in petrol, diesel, and electric vehicles was about **-40.53%**, **-21.27%**, and **56.97%** respectively. 3. **Nirmal**: The total sales growth was approximately **-34.45%**. The growth in petrol, diesel, and electric vehicles was about **-41.05%**, **-1.42%**, and **89.30%** respectively.

These figures indicate that while some districts have seen significant growth in vehicle sales, others have experienced a decline during FY 2022 compared to FY 2021.

1.4.3 Ts-Ipass (Telangana State Industrial Project Approval and Self Certification System)

8. List down the top 5 sectors that have witnessed the most significant investments in FY 2022.

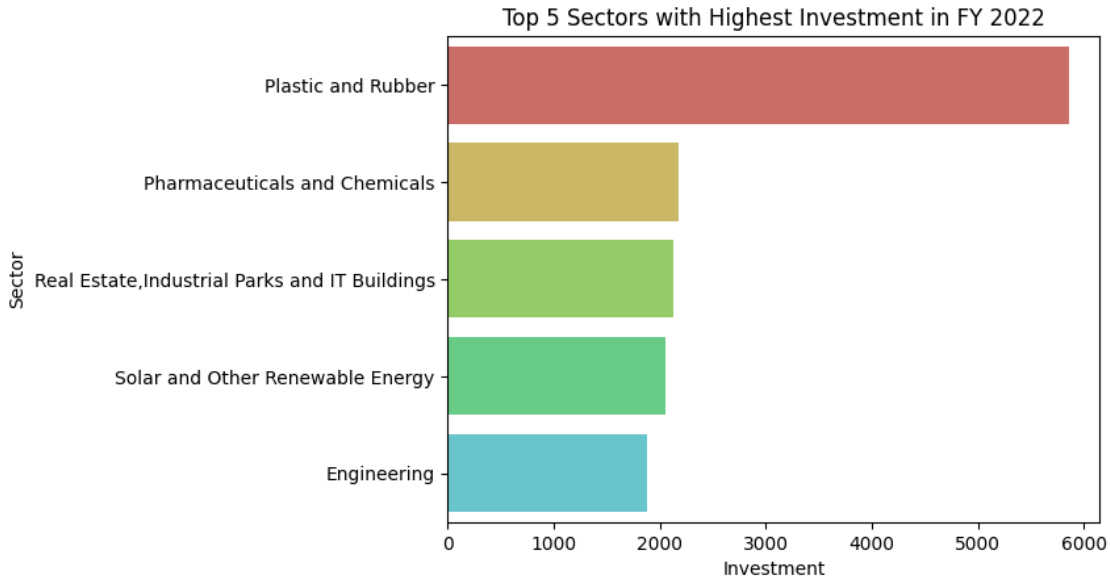
[292]:
```
ts_ipass_fy_2022 = ts_ipass_df[ts_ipass_df['fiscal_year'] == 2022]
```

[293]:
```
sector_invest_2022 = ts_ipass_fy_2022.groupby('sector')['investment in cr'].
↪sum()
top_5_sector_invest_2022 = sector_invest_2022.sort_values(ascending = False).
↪head(5)
top_5_sector_invest_2022 = top_5_sector_invest_2022.reset_index()
top_5_sector_invest_2022
```

[293]:

	sector	investment in cr
0	Plastic and Rubber	5855.6095
1	Pharmaceuticals and Chemicals	2181.6342
2	Real Estate,Industrial Parks and IT Buildings	2127.2963
3	Solar and Other Renewable Energy	2052.9850
4	Engineering	1877.4533

[294]:
```
colors = sns.color_palette('hls', 8)
sns.barplot(x = top_5_sector_invest_2022['investment in cr'], y =␣
↪top_5_sector_invest_2022['sector'], orient = 'h', palette = colors)
plt.title('Top 5 Sectors with Highest Investment in FY 2022')
plt.xlabel('Investment')
plt.ylabel('Sector')
plt.show()
```



The top 5 sectors that have witnessed the most significant investments in FY 2022 are:

1. **Plastic and Rubber**: 5855.6095 cr
2. **Pharmaceuticals and Chemicals**: 2181.6342 cr
3. **Real Estate, Industrial Parks and IT Buildings**: 2127.2963 cr
4. **Solar and Other Renewable Energy**: 2052.9850 cr
5. **Engineering**: 1877.4533 cr

These sectors have attracted the highest investments, indicating strong growth and development potential.

9. 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?

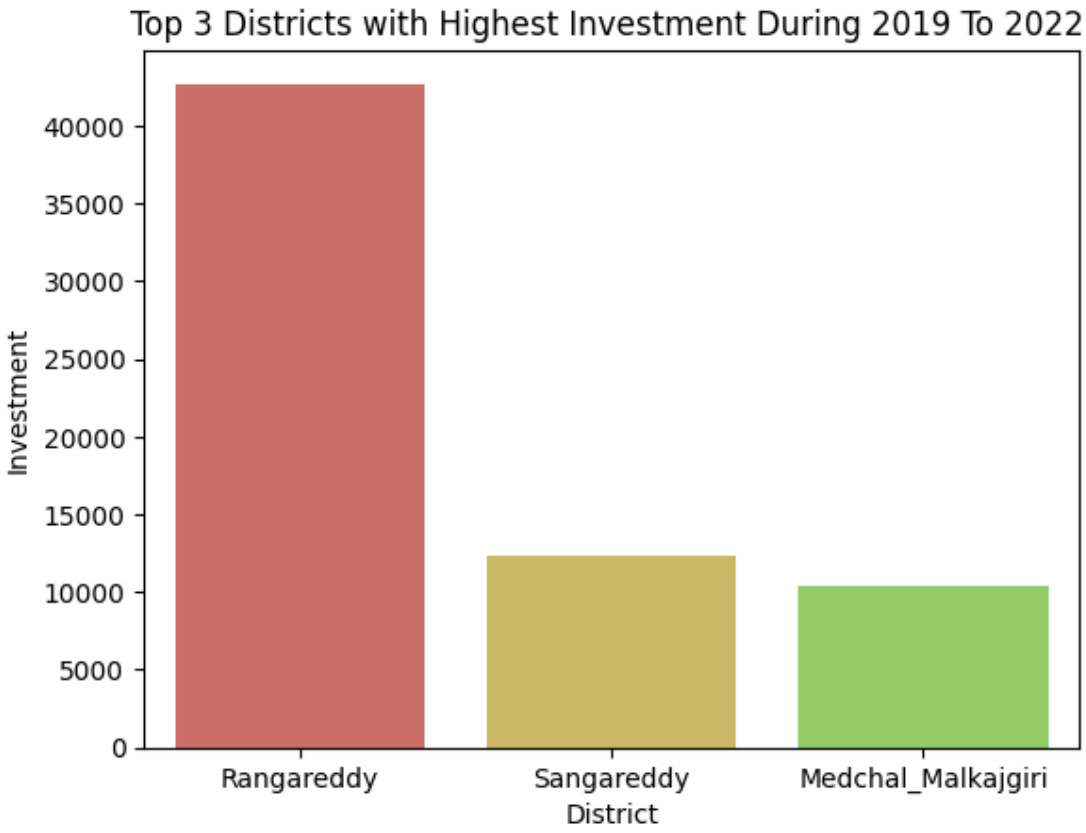
[295]:
```
distric_invest_2019_2022 = ts_ipass_df.groupby('district')['investment in cr'].
↪sum()
top_3_distric_invest_2019_2022 = distric_invest_2019_2022.sort_values(ascending␣
↪= False).head(3)
top_3_distric_invest_2019_2022 = top_3_distric_invest_2019_2022.reset_index()
top_3_distric_invest_2019_2022
```

[295]:

	district	investment in cr
0	Rangareddy	42706.3320
1	Sangareddy	12366.7556
2	Medchal_Malkajgiri	10394.5610

[296]:
```
colors = sns.color_palette('hls', 8)
```

```
sns.barplot(x = top_3_distric_invest_2019_2022['district'], y =␣
 ↪top_3_distric_invest_2019_2022['investment in cr'], palette = colors)
plt.title('Top 3 Districts with Highest Investment During 2019 To 2022 ')
plt.xlabel('District')
plt.ylabel('Investment')
plt.show()
```



Factors could have led to substantial investments in these particular districts, it could be a variety of factors such as:

- **Infrastructure**: Districts with better infrastructure such as roads, ports, airports, and utilities often attract more investment.
- **Government Policies**: Favorable government policies such as tax incentives, subsidies, and ease of doing business can also attract investment.
- **Availability of Skilled Labor**: Districts with a large pool of skilled labor can attract industries that require such skills.
- **Market Access**: Districts that are close to major markets or have good connectivity to them can attract more investment.

10. Is there any relationship between district investments, vehicles sales and stamps revenue within the same district between FY 2021 and 2022?

[297]:
```
stamps_df_2021_2022 = stamps_df[stamps_df['fiscal_year'].isin([2021,2022])]
transport_df_2021_2022 = transport_df[transport_df['fiscal_year'].
 ↪isin([2021,2022])]
ts_ipass_df_2021_2022 = ts_ipass_df[ts_ipass_df['fiscal_year'].
 ↪isin([2021,2022])]
```

[298]:
```
transport_df_2021_2022['total_vehicle_sales'] =␣
 ↪transport_df_2021_2022[['fuel_type_petrol', 'fuel_type_diesel',␣
 ↪'fuel_type_electric', 'fuel_type_others', 'vehicleClass_MotorCycle',␣
 ↪'vehicleClass_MotorCar', 'vehicleClass_AutoRickshaw',␣
 ↪'vehicleClass_Agriculture', 'vehicleClass_others']].sum(axis=1)
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_856\77516591.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  transport_df_2021_2022['total_vehicle_sales'] =
transport_df_2021_2022[['fuel_type_petrol', 'fuel_type_diesel',
'fuel_type_electric', 'fuel_type_others', 'vehicleClass_MotorCycle',
'vehicleClass_MotorCar', 'vehicleClass_AutoRickshaw',
'vehicleClass_Agriculture', 'vehicleClass_others']].sum(axis=1)
```

[299]:
```
#District investment
invest_by_dist = ts_ipass_df_2021_2022.groupby('district')['investment in cr'].
 ↪sum()

#Vehicle Sales
total_vehicle_sales_by_dist = transport_df_2021_2022.
 ↪groupby('district')['total_vehicle_sales'].sum()

#stamps revenue and document revenue
document_rev = stamps_df_2021_2022.
 ↪groupby('district')['documents_registered_rev'].sum()
estamps_rev = stamps_df_2021_2022.groupby('district')['estamps_challans_rev'].
 ↪sum()
```

[300]:
```
# Combine the Series into a DataFrame
combined_df = pd.concat([invest_by_dist, total_vehicle_sales_by_dist,␣
 ↪document_rev, estamps_rev], axis=1)

# Rename the columns
combined_df.columns = ['investment', 'total_vehicle_sales',␣
 ↪'documents_registered_rev', 'estamps_challans_rev']
```

[301]:
```
combined_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 33 entries, Adilabad to Yadadri Bhuvanagiri
Data columns (total 4 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   investment               33 non-null     float64
 1   total_vehicle_sales      30 non-null     float64
 2   documents_registered_rev  32 non-null     float64
 3   estamps_challans_rev     32 non-null     float64
dtypes: float64(4)
memory usage: 1.3+ KB
```

[302]:
```
combined_df.isna().sum()
```

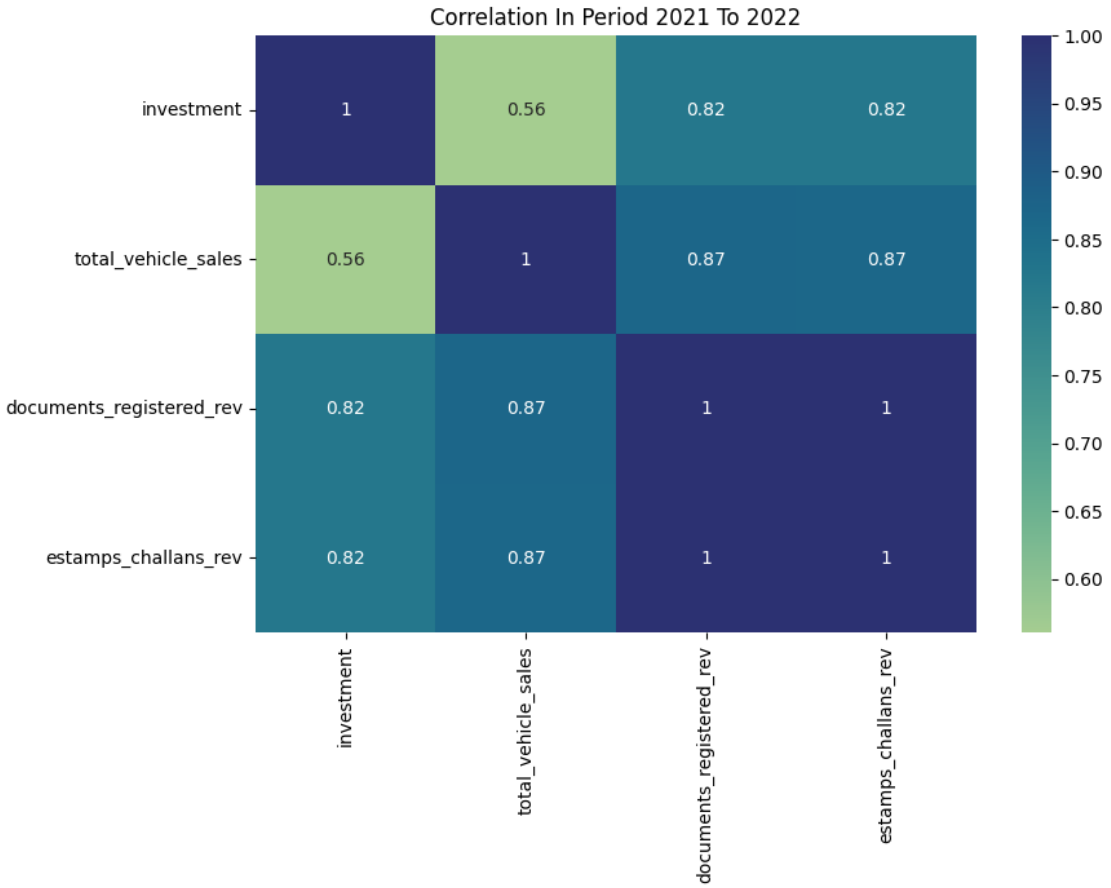
[302]:
```
investment                 0
total_vehicle_sales        3
documents_registered_rev   1
estamps_challans_rev       1
dtype: int64
```

[303]:
```
combined_df = combined_df.fillna(combined_df.mean())
```

[304]:
```
combined_df.isna().sum()
```

[304]:
```
investment                 0
total_vehicle_sales        0
documents_registered_rev   0
estamps_challans_rev       0
dtype: int64
```

[305]:
```
color = sns.color_palette("crest", as_cmap=True)
plt.figure(figsize = (9,6))
sns.heatmap(combined_df.corr(),cmap = color ,annot = True)
plt.title('Correlation In Period 2021 To 2022')
plt.show()
```



- Investment and Total Vehicle Sales: The correlation coefficient is approximately 0.56, indicating a moderate positive relationship. This suggests that districts with higher investments tend to have higher vehicle sales.

- Investment and Documents Registered Revenue: The correlation coefficient is approximately 0.82, indicating a strong positive relationship. This suggests that districts with higher investments tend to have higher revenue from registered documents.

- Investment and E-stamp Challans Revenue: The correlation coefficient is approximately 0.82, indicating a strong positive relationship. This suggests that districts with higher investments tend to have higher revenue from e-stamp challans.

- Total Vehicle Sales and Documents Registered Revenue: The correlation coefficient is approximately 0.87, indicating a strong positive relationship. This suggests that districts with higher vehicle sales tend to have higher revenue from registered documents.

- Total Vehicle Sales and E-stamp Challans Revenue: The correlation coefficient is approximately 0.87, indicating a strong positive relationship. This suggests that districts with higher vehicle sales tend to have higher revenue from e-stamp challans.

- Documents Registered Revenue and E-stamp Challans Revenue: The correlation coefficient is approximately 1.00, indicating a very strong positive relationship. This suggests that

districts with higher revenue from registered documents also tend to have higher revenue from e-stamp challans.

11. Are there any particular sectors that have shown substantial investment in multiple districts between FY 2021 and 2022?

Ficsal year 2021:

[307]:
```
ts_ipass_df_2021 = ts_ipass_df[ts_ipass_df['fiscal_year'] == 2021]
sector_by_dist_2021_temp = ts_ipass_df_2021.groupby('sector')['district'].
 ↪unique()
sector_by_dist_2021_temp = sector_by_dist_2021_temp.reset_index()
sector_by_dist_2021_temp['num_district'] = sector_by_dist_2021_temp['district'].
 ↪apply(len)
sector_by_dist_2021_temp
```

[307]:

	sector	district	num_district
0	Agro based incl Cold Storages	[Jagtial, Jangoan, Kamareddy, Karimnagar, Kham…	28
1	Automobile	[Nirmal, Rangareddy]	2
2	Beverages	[Adilabad, Jagtial, Jangoan, Khammam, Medak, S…	21
3	Cement, Cement & Concrete Products, Fly Ash Br…	[Adilabad, Bhadradri Kothagudem, Jagtial, Jang…	28
4	Electrical and Electronic Products	[Medak, Medchal_Malkajgiri, Sangareddy, Warang…	10
5	Engineering	[Hyderabad, Jagtial, Karimnagar, Khammam, Manc…	30
6	Fertlizers Organic and Inorganic,Pesticides,In…	[Yadadri Bhuvanagiri, Rangareddy, Medak, Sanga…	10
7	Food Processing	[Jagtial, Jangoan, Jogulamba Gadwal, Kamareddy…	33
8	Granite and Stone Crushing	[Karimnagar, Khammam, Nalgonda, Nizamabad, Raj…	24
9	Others	[Mancherial, Nizamabad, Rangareddy, Sangareddy…	27
10	Paper and Printing	[Medak, Medchal_Malkajgiri, Nagarkurnool, Yada…	17
11	Pharmaceuticals and Chemicals		
12	Plastic and Rubber		
13	R&D		
14	Real Estate,Industrial Parks and IT Buildings		
15	Solar and Other Renewable Energy		
16	Textiles		
17	Wood and Leather		

```
11  [Jagtial, Medak, Medchal_Malkajgiri, Rangaredd…          24
12  [Jogulamba Gadwal, Mancherial, Medchal_Malkajg…          23
13  [Medchal_Malkajgiri, Sangareddy, Mahabubnagar,…           6
14              [Sangareddy, Rangareddy, Karimnagar]            3
15  [Yadadri Bhuvanagiri, Medchal_Malkajgiri, Sang…           7
16  [Jogulamba Gadwal, Narayanpet, Rajanna Sircill…          16
17  [Jagtial, Kamareddy, Sangareddy, Suryapet, Vik…          20
```

[308]:
```
total_invest_by_sector_2021_temp = ts_ipass_df_2021.
 ↪groupby('sector')['investment in cr'].sum()
total_invest_by_sector_2021_temp = total_invest_by_sector_2021_temp.
 ↪reset_index()
total_invest_by_sector_2021_temp
```

[308]:
```
                                              sector  investment in cr
0                       Agro based incl Cold Storages          325.2291
1                                          Automobile            5.0000
2                                           Beverages         1843.6802
3   Cement, Cement & Concrete Products, Fly Ash Br…          120.9007
4                  Electrical and Electronic Products          176.4594
5                                         Engineering          967.5172
6   Fertlizers Organic and Inorganic,Pesticides,In…           23.4525
7                                     Food Processing          999.3621
8                          Granite and Stone Crushing          400.7498
9                                              Others          647.6261
10                                Paper and Printing          205.9040
11                     Pharmaceuticals and Chemicals         6860.9321
12                                Plastic and Rubber         2059.6245
13                                               R&D          876.7164
14     Real Estate,Industrial Parks and IT Buildings         2015.2619
15                   Solar and Other Renewable Energy          164.0238
16                                           Textiles          162.3720
17                                Wood and Leather            34.0320
```

[309]:
```
sector_dist_investment_2021 = pd.merge(sector_by_dist_2021_temp,␣
 ↪total_invest_by_sector_2021_temp, on = 'sector', how = 'inner')
sector_dist_investment_2021
```

[309]:
```
                                              sector  \
0                       Agro based incl Cold Storages
1                                          Automobile
2                                           Beverages
3   Cement, Cement & Concrete Products, Fly Ash Br…
4                  Electrical and Electronic Products
5                                         Engineering
6   Fertlizers Organic and Inorganic,Pesticides,In…
7                                     Food Processing
```

8	Granite and Stone Crushing			
9	Others			
10	Paper and Printing			
11	Pharmaceuticals and Chemicals			
12	Plastic and Rubber			
13	R&D			
14	Real Estate,Industrial Parks and IT Buildings			
15	Solar and Other Renewable Energy			
16	Textiles			
17	Wood and Leather			

	district	num_district	\
0	[Jagtial, Jangoan, Kamareddy, Karimnagar, Kham…	28	
1	[Nirmal, Rangareddy]	2	
2	[Adilabad, Jagtial, Jangoan, Khammam, Medak, S…	21	
3	[Adilabad, Bhadradri Kothagudem, Jagtial, Jang…	28	
4	[Medak, Medchal_Malkajgiri, Sangareddy, Warang…	10	
5	[Hyderabad, Jagtial, Karimnagar, Khammam, Manc…	30	
6	[Yadadri Bhuvanagiri, Rangareddy, Medak, Sanga…	10	
7	[Jagtial, Jangoan, Jogulamba Gadwal, Kamareddy…	33	
8	[Karimnagar, Khammam, Nalgonda, Nizamabad, Raj…	24	
9	[Mancherial, Nizamabad, Rangareddy, Sangareddy…	27	
10	[Medak, Medchal_Malkajgiri, Nagarkurnool, Yada…	17	
11	[Jagtial, Medak, Medchal_Malkajgiri, Rangaredd…	24	
12	[Jogulamba Gadwal, Mancherial, Medchal_Malkajg…	23	
13	[Medchal_Malkajgiri, Sangareddy, Mahabubnagar,…	6	
14	[Sangareddy, Rangareddy, Karimnagar]	3	
15	[Yadadri Bhuvanagiri, Medchal_Malkajgiri, Sang…	7	
16	[Jogulamba Gadwal, Narayanpet, Rajanna Sircill…	16	
17	[Jagtial, Kamareddy, Sangareddy, Suryapet, Vik…	20	

	investment in cr
0	325.2291
1	5.0000
2	1843.6802
3	120.9007
4	176.4594
5	967.5172
6	23.4525
7	999.3621
8	400.7498
9	647.6261
10	205.9040
11	6860.9321
12	2059.6245
13	876.7164
14	2015.2619

```
15          164.0238
16          162.3720
17           34.0320
```

[310]:
```
sector_by_district_2021 = sector_dist_investment_2021.sort_values(by =␣
↪'num_district', ascending = True)
top_10_sector_by_district_2021 = sector_by_district_2021.tail(10)
top_10_sector_by_district_2021
```

[310]:
```
                                          sector  \
17                              Wood and Leather
2                                      Beverages
12                           Plastic and Rubber
8                      Granite and Stone Crushing
11                   Pharmaceuticals and Chemicals
9                                        Others
3   Cement, Cement & Concrete Products, Fly Ash Br…
0                     Agro based incl Cold Storages
5                                    Engineering
7                              Food Processing

                                           district  num_district  \
17  [Jagtial, Kamareddy, Sangareddy, Suryapet, Vik…            20
2   [Adilabad, Jagtial, Jangoan, Khammam, Medak, S…            21
12  [Jogulamba Gadwal, Mancherial, Medchal_Malkajg…            23
8   [Karimnagar, Khammam, Nalgonda, Nizamabad, Raj…            24
11  [Jagtial, Medak, Medchal_Malkajgiri, Rangaredd…            24
9   [Mancherial, Nizamabad, Rangareddy, Sangareddy…            27
3   [Adilabad, Bhadradri Kothagudem, Jagtial, Jang…            28
0   [Jagtial, Jangoan, Kamareddy, Karimnagar, Kham…            28
5   [Hyderabad, Jagtial, Karimnagar, Khammam, Manc…            30
7   [Jagtial, Jangoan, Jogulamba Gadwal, Kamareddy…            33

    investment in cr
17           34.0320
2          1843.6802
12         2059.6245
8           400.7498
11         6860.9321
9           647.6261
3           120.9007
0           325.2291
5           967.5172
7           999.3621
```

[311]:
```
invest_amount_2021 = sector_dist_investment_2021.sort_values(by = 'investment␣
↪in cr', ascending = True)
```

```
top_10_invest_amount_2021 = invest_amount_2021.tail(10)
top_10_invest_amount_2021
```

[311]:

	sector	district	num_district	investment in cr
0	Agro based incl Cold Storages	[Jagtial, Jangoan, Kamareddy, Karimnagar, Kham…	28	325.2291
8	Granite and Stone Crushing	[Karimnagar, Khammam, Nalgonda, Nizamabad, Raj…	24	400.7498
9	Others	[Mancherial, Nizamabad, Rangareddy, Sangareddy…	27	647.6261
13	R&D	[Medchal_Malkajgiri, Sangareddy, Mahabubnagar,…	6	876.7164
5	Engineering	[Hyderabad, Jagtial, Karimnagar, Khammam, Manc…	30	967.5172
7	Food Processing	[Jagtial, Jangoan, Jogulamba Gadwal, Kamareddy…	33	999.3621
2	Beverages	[Adilabad, Jagtial, Jangoan, Khammam, Medak, S…	21	1843.6802
14	Real Estate,Industrial Parks and IT Buildings	[Sangareddy, Rangareddy, Karimnagar]	3	2015.2619
12	Plastic and Rubber	[Jogulamba Gadwal, Mancherial, Medchal_Malkajg…	23	2059.6245
11	Pharmaceuticals and Chemicals	[Jagtial, Medak, Medchal_Malkajgiri, Rangaredd…	24	6860.9321

Fiscal year 2022:

[312]:
```
ts_ipass_df_2022 = ts_ipass_df[ts_ipass_df['fiscal_year'] == 2022]
sector_by_dist_2022_temp = ts_ipass_df_2022.groupby('sector')['district'].
 ↪unique()
sector_by_dist_2022_temp = sector_by_dist_2022_temp.reset_index()
sector_by_dist_2022_temp['num_district'] = sector_by_dist_2022_temp['district'].
 ↪apply(len)
```

```
sector_by_dist_2022_temp
```

[312]:

	sector	district	num_district
0	Agro based incl Cold Storages	[Jagtial, Kamareddy, Karimnagar, Khammam, Maha…	25
1	Automobile	[Sangareddy, Rangareddy, Medchal_Malkajgiri]	3
2	Beverages	[Bhadradri Kothagudem, Jangoan, Khammam, Mahab…	24
3	Cement, Cement & Concrete Products, Fly Ash Br…	[Hanumakonda, Jangoan, Karimnagar, Khammam, Ku…	31
4	Electrical and Electronic Products	[Medchal_Malkajgiri, Nirmal, Rangareddy, Medak…	8
5	Engineering	[Hanumakonda, Hyderabad, Jagtial, Kamareddy, K…	27
6	Fertlizers Organic and Inorganic,Pesticides,In…	[Hanumakonda, Sangareddy, Yadadri Bhuvanagiri,…	8
7	Food Processing	[Bhadradri Kothagudem, Hanumakonda, Jagtial, J…	33
8	Granite and Stone Crushing	[Bhadradri Kothagudem, Hanumakonda, Kamareddy,…	21
9	Industrial Parks and IT Buildings	[Rangareddy]	1
10	Others	[Jangoan, Karimnagar, Khammam, Mahabubnagar, M…	29
11	Paper and Printing	[Bhadradri Kothagudem, Jagtial, Medchal_Malkaj…	16
12	Pharmaceuticals and Chemicals	[Kamareddy, Mahabubnagar, Medak, Medchal_Malka…	18
13	Plastic and Rubber	[Karimnagar, Medak, Medchal_Malkajgiri, Nalgon…	18
14	R&D	[Medchal_Malkajgiri, Sangareddy, Siddipet, Yad…	8
15	Real Estate,Industrial Parks and IT Buildings	[Karimnagar, Rangareddy, Yadadri Bhuvanagiri]	3
16	Solar and Other Renewable Energy	[Jogulamba Gadwal, Mahabubnagar, Medak, Nirmal…	11
17	Textiles	[Jangoan, Mancherial, Rajanna Sircilla, Hanuma…	17
18	Wood and Leather	[Mahabubabad, Siddipet, Karimnagar, Medchal_Ma…	19

[313]:
```
total_invest_by_sector_2022_temp = ts_ipass_df_2022.
↪groupby('sector')['investment in cr'].sum()
```

```
total_invest_by_sector_2022_temp = total_invest_by_sector_2022_temp.
 ↪reset_index()
total_invest_by_sector_2022_temp
```

[313]:

	sector	investment in cr
0	Agro based incl Cold Storages	1263.5502
1	Automobile	1567.4271
2	Beverages	476.7425
3	Cement, Cement & Concrete Products, Fly Ash Br…	1142.7468
4	Electrical and Electronic Products	484.4655
5	Engineering	1877.4533
6	Fertlizers Organic and Inorganic,Pesticides,In…	34.2244
7	Food Processing	1455.9435
8	Granite and Stone Crushing	946.7192
9	Industrial Parks and IT Buildings	280.4090
10	Others	1040.3561
11	Paper and Printing	1251.3714
12	Pharmaceuticals and Chemicals	2181.6342
13	Plastic and Rubber	5855.6095
14	R&D	1484.9027
15	Real Estate,Industrial Parks and IT Buildings	2127.2963
16	Solar and Other Renewable Energy	2052.9850
17	Textiles	176.1469
18	Wood and Leather	63.9763

[314]:
```
sector_dist_investment_2022 = pd.merge(sector_by_dist_2022_temp,␣
 ↪total_invest_by_sector_2022_temp, on = 'sector', how = 'inner')
sector_dist_investment_2022
```

[314]:

	sector
0	Agro based incl Cold Storages
1	Automobile
2	Beverages
3	Cement, Cement & Concrete Products, Fly Ash Br…
4	Electrical and Electronic Products
5	Engineering
6	Fertlizers Organic and Inorganic,Pesticides,In…
7	Food Processing
8	Granite and Stone Crushing
9	Industrial Parks and IT Buildings
10	Others
11	Paper and Printing
12	Pharmaceuticals and Chemicals
13	Plastic and Rubber
14	R&D
15	Real Estate,Industrial Parks and IT Buildings
16	Solar and Other Renewable Energy

```
17                                          Textiles
18                                  Wood and Leather

                                                district  num_district  \
0   [Jagtial, Kamareddy, Karimnagar, Khammam, Maha…              25
1        [Sangareddy, Rangareddy, Medchal_Malkajgiri]               3
2   [Bhadradri Kothagudem, Jangoan, Khammam, Mahab…              24
3   [Hanumakonda, Jangoan, Karimnagar, Khammam, Ku…              31
4   [Medchal_Malkajgiri, Nirmal, Rangareddy, Medak…               8
5   [Hanumakonda, Hyderabad, Jagtial, Kamareddy, K…              27
6   [Hanumakonda, Sangareddy, Yadadri Bhuvanagiri,…               8
7   [Bhadradri Kothagudem, Hanumakonda, Jagtial, J…              33
8   [Bhadradri Kothagudem, Hanumakonda, Kamareddy,…              21
9                                       [Rangareddy]               1
10  [Jangoan, Karimnagar, Khammam, Mahabubnagar, M…              29
11  [Bhadradri Kothagudem, Jagtial, Medchal_Malkaj…              16
12  [Kamareddy, Mahabubnagar, Medak, Medchal_Malka…              18
13  [Karimnagar, Medak, Medchal_Malkajgiri, Nalgon…              18
14  [Medchal_Malkajgiri, Sangareddy, Siddipet, Yad…               8
15      [Karimnagar, Rangareddy, Yadadri Bhuvanagiri]               3
16  [Jogulamba Gadwal, Mahabubnagar, Medak, Nirmal…              11
17  [Jangoan, Mancherial, Rajanna Sircilla, Hanuma…              17
18  [Mahabubabad, Siddipet, Karimnagar, Medchal_Ma…              19

    investment in cr
0          1263.5502
1          1567.4271
2           476.7425
3          1142.7468
4           484.4655
5          1877.4533
6            34.2244
7          1455.9435
8           946.7192
9           280.4090
10         1040.3561
11         1251.3714
12         2181.6342
13         5855.6095
14         1484.9027
15         2127.2963
16         2052.9850
17          176.1469
18           63.9763
```

[315]:
```
sector_by_district_2022 = sector_dist_investment_2022.sort_values(by =␣
↪'num_district', ascending = True)
```

```
top_10_sector_by_district_2022 = sector_by_district_2022.tail(10)
top_10_sector_by_district_2022
```

[315]:

	sector	district	num_district	investment in cr
13	Plastic and Rubber	[Karimnagar, Medak, Medchal_Malkajgiri, Nalgon…	18	5855.6095
12	Pharmaceuticals and Chemicals	[Kamareddy, Mahabubnagar, Medak, Medchal_Malka…	18	2181.6342
18	Wood and Leather	[Mahabubabad, Siddipet, Karimnagar, Medchal_Ma…	19	63.9763
8	Granite and Stone Crushing	[Bhadradri Kothagudem, Hanumakonda, Kamareddy,…	21	946.7192
2	Beverages	[Bhadradri Kothagudem, Jangoan, Khammam, Mahab…	24	476.7425
0	Agro based incl Cold Storages	[Jagtial, Kamareddy, Karimnagar, Khammam, Maha…	25	1263.5502
5	Engineering	[Hanumakonda, Hyderabad, Jagtial, Kamareddy, K…	27	1877.4533
10	Others	[Jangoan, Karimnagar, Khammam, Mahabubnagar, M…	29	1040.3561
3	Cement, Cement & Concrete Products, Fly Ash Br…	[Hanumakonda, Jangoan, Karimnagar, Khammam, Ku…	31	1142.7468
7	Food Processing	[Bhadradri Kothagudem, Hanumakonda, Jagtial, J…	33	1455.9435

[316]:
```
invest_amount_2022 = sector_dist_investment_2022.sort_values(by = 'investment␣
↪in cr', ascending = True)
top_10_invest_amount_2022 = invest_amount_2022.tail(10)
top_10_invest_amount_2022
```

[316]:

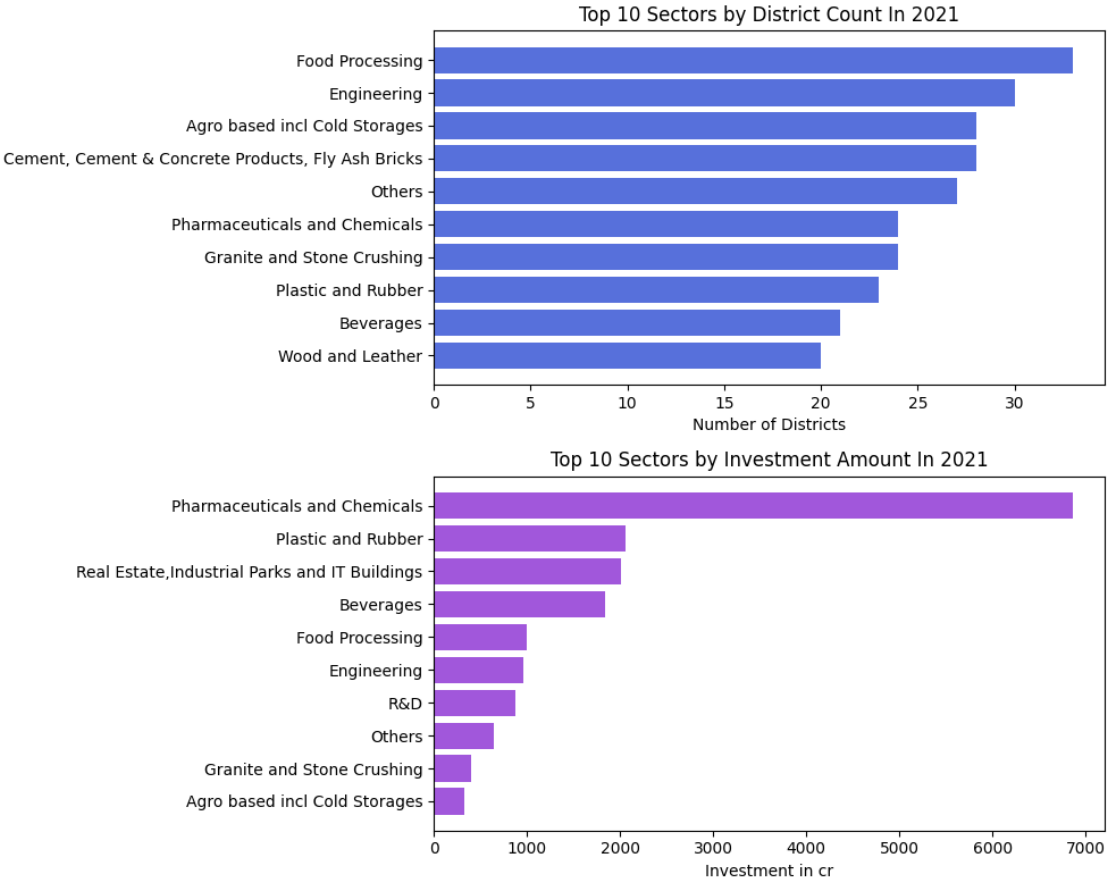
	sector
11	Paper and Printing
0	Agro based incl Cold Storages

```
7                                  Food Processing
14                                             R&D
1                                       Automobile
5                                      Engineering
16                 Solar and Other Renewable Energy
15  Real Estate,Industrial Parks and IT Buildings
12                    Pharmaceuticals and Chemicals
13                                Plastic and Rubber

                                              district  num_district  \
11  [Bhadradri Kothagudem, Jagtial, Medchal_Malkaj…            16
0   [Jagtial, Kamareddy, Karimnagar, Khammam, Maha…            25
7   [Bhadradri Kothagudem, Hanumakonda, Jagtial, J…            33
14  [Medchal_Malkajgiri, Sangareddy, Siddipet, Yad…             8
1        [Sangareddy, Rangareddy, Medchal_Malkajgiri]             3
5   [Hanumakonda, Hyderabad, Jagtial, Kamareddy, K…            27
16  [Jogulamba Gadwal, Mahabubnagar, Medak, Nirmal…            11
15      [Karimnagar, Rangareddy, Yadadri Bhuvanagiri]             3
12  [Kamareddy, Mahabubnagar, Medak, Medchal_Malka…            18
13  [Karimnagar, Medak, Medchal_Malkajgiri, Nalgon…            18

    investment in cr
11         1251.3714
0          1263.5502
7          1455.9435
14         1484.9027
1          1567.4271
5          1877.4533
16         2052.9850
15         2127.2963
12         2181.6342
13         5855.6095
```

Visualize Sectors have invested in a number of districts, and the total investment by each sector for fiscal years 2021 and 2022

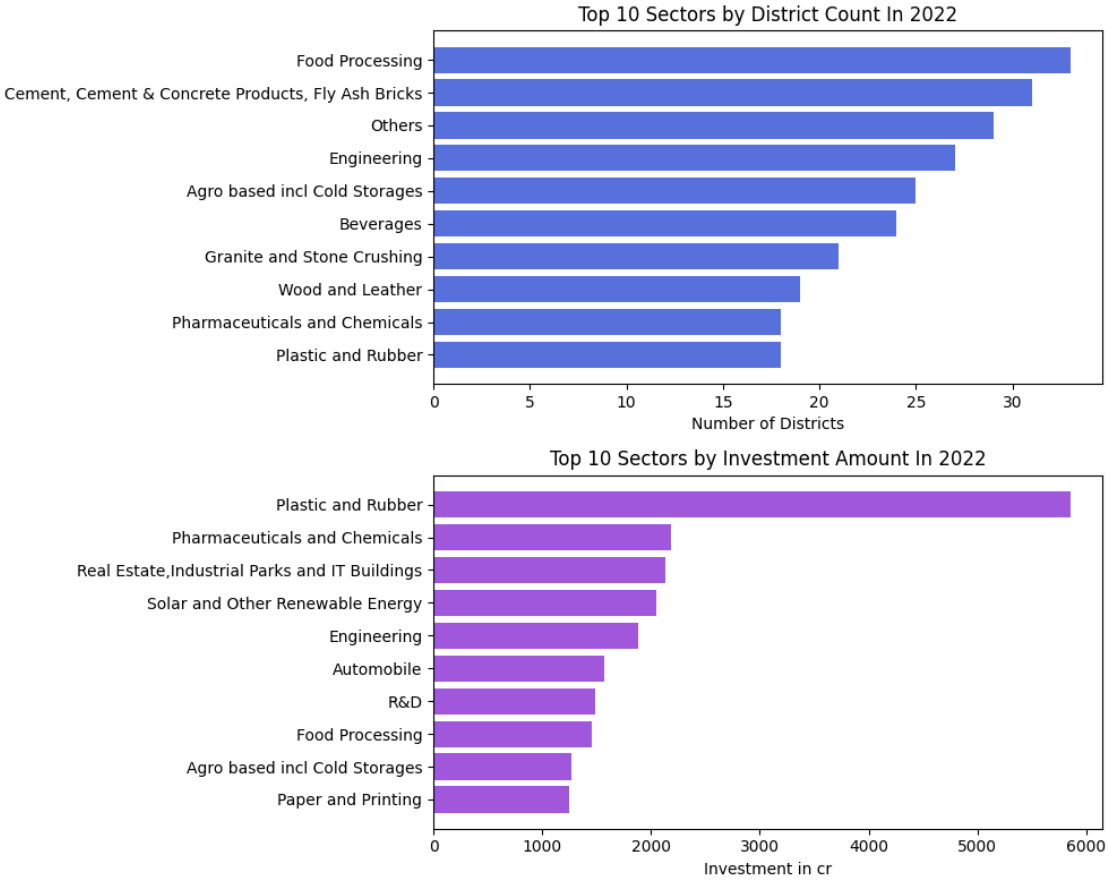
[317]:
```
fig, ax = plt.subplots(2,figsize = (10,8))

colors = sns.color_palette('hls', 8)

ax[0].barh(top_10_sector_by_district_2021['sector'],␣
 ↪top_10_sector_by_district_2021['num_district'], color= colors[5])
ax[0].set_xlabel('Number of Districts')
ax[0].set_title('Top 10 Sectors by District Count In 2021')
```

```
ax[1].barh(top_10_invest_amount_2021['sector'],␣
 ↪top_10_invest_amount_2021['investment in cr'], color=colors[6])
ax[1].set_xlabel('Investment in cr')
ax[1].set_title('Top 10 Sectors by Investment Amount In 2021')


plt.tight_layout()
plt.show()
```



[318]:
```
fig, ax = plt.subplots(2,figsize = (10,8))

colors = sns.color_palette('hls', 8)

ax[0].barh(top_10_sector_by_district_2022['sector'],␣
 ↪top_10_sector_by_district_2022['num_district'], color= colors[5])
ax[0].set_xlabel('Number of Districts')
ax[0].set_title('Top 10 Sectors by District Count In 2022')
```

```
ax[1].barh(top_10_invest_amount_2022['sector'],␣
 ↪top_10_invest_amount_2022['investment in cr'], color=colors[6])
ax[1].set_xlabel('Investment in cr')
ax[1].set_title('Top 10 Sectors by Investment Amount In 2022')


plt.tight_layout()
plt.show()
```



- **Pharmaceuticals and Chemicals** sector has consistently been a major player in terms of investment, with an investment of 6860.9321 cr in 2021 and 2181.6342 cr in 2022. However, there was a significant decrease in investment from 2021 to 2022.
- **The Plastic and Rubber** sector saw a significant increase in investment from 2059.6245 cr in 2021 to 5855.6095 cr in 2022, making it the sector with the highest investment in 2022.
- **The Food Processing** sector consistently had the highest number of districts involved, with 33 districts in 2021 and the same number in 2022.
- **The Engineering** sector also saw a consistent presence across the years, with an investment of 967.5172 cr across 30 districts in 2021, and an increased investment of 1877.4533 cr across

27 districts in 2022.

- New sectors like **R&D, Automobile, and Solar and Other Renewable Energy** have emerged in the top 10 sectors by investment amount in 2022, which were not present in the top 10 list of 2021.
- Some sectors like **Beverages, and Granite and Stone Crushing** have seen a decrease in their investments from 2021 to 2022.

These trends indicate a dynamic shift in the sectors that are attracting investments year over year, with some sectors consistently performing well, while others are emerging as new areas of interest.

12. Can we identify any seasonal patterns or cyclicality in the investment trends for specific sectors? Do certain sectors experience higher investments during particular months?

[319]:
```
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct',␣
 ↪'Nov', 'Dec']

grouped = ts_ipass_df.groupby(['sector', 'Mmm'])['investment in cr'].sum().
 ↪reset_index()
grouped['Mmm'] = pd.Categorical(grouped['Mmm'], categories=months, ordered=True)

pivot_df = grouped.pivot(index='Mmm', columns='sector', values='investment in␣
 ↪cr')


pivot_df.plot(subplots=True, layout=(-1, 2), figsize=(14, 22))
plt.tight_layout()
plt.show()
```


Agro based incl Cold Storages
Automobile
Beverages
Cement, Cement & Concrete Products, Fly Ash Bricks
Electrical and Electronic Products
Engineering
Fertlizers Organic and Inorganic,Pesticides,Insecticides, and Other Related
Food Processing
Granite and Stone Crushing
Industrial Parks and IT Buildings
Others
Paper and Printing
Pharmaceuticals and Chemicals
Plastic and Rubber
R&D
Real Estate,Industrial Parks and IT Buildings
Solar and Other Renewable Energy
Textiles
Thermal Power Plant
Wood and Leather
Jan
Mar
May
Jul
Sep
Nov
Mmm

- The **Agro based incl Cold Storages** sector seems to have higher investments in the months of June and March.
- The **Automobile** sector shows a significant investment in January.
- The **Beverages** sector has high investments in January and February.
- The **Cement, Cement & Concrete Products, Fly Ash Bricks** sector shows a peak in August.
- The **Engineering** sector has higher investments in February and October.
- The **Fertlizers Organic and Inorganic,Pesticides,Insecticides, and Other Related** sector shows a significant peak in December.
- The **Food Processing** sector has relatively consistent investments throughout the year with slight peaks in March and September.
- The **Granite and Stone Crushing** sector shows a peak in October.
- The **Pharmaceuticals and Chemicals** sector shows high investments in December, September, and February.
- The **Plastic and Rubber** sector has a significant peak in April.

Thank you for reading my project

Author: Bao Thai
Linkedin: https://www.linkedin.com/in/bao-quoc-508008181/
Gmail: QuocBao363@gmail.com