DATA SCIENCE PROJECT USING PTHON

REPORT

(Project Semester January-April 2025)

indian-company-registrations

Submitted by

Rahul Kumar Yadav

Registration no:

12327179

Section: K23GD

Course Code:

INT375

Under the Guidance of

Baljinder Kaur,

(27952)

Discipline of CSE/IT

Lovely School of Computer Science Engineering

Lovely Professional University, Phagwara

CERTIFICATE

This is to certify that **Rahul Kumar Yadav** bearing Registration no. **12327179** has completed **INT375** project titled, **"Dindian-company-registrations "** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Baljinder

Kaur

Professor

School of Computer Science Engineering

Lovely Professional University

Phagwara, Punjab.

Date: 13th April, 2025

DECLARATION

I, Rahul Kumar Yadav, student of B.tech under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 13th April, 2025 Signature

Registration No.12327179 Rahul

Acknowledgment

I would like to express my deepest gratitude to Prof. Baljinder Kaur, for her exceptional mentorship and unwavering support throughout the duration of this project. Her vast knowledge in the fields of data science and machine learning, combined with her patient and thoughtful guidance, played a pivotal role in the successful completion of this work. Her insightful suggestions and feedback consistently challenged me to think critically and improve the quality of my research. I am also grateful for the learning environment she fostered, which encouraged exploration and innovation.

In addition, I sincerely thank my peers and classmates for their helpful discussions, encouragement, and collaborative spirit during this project. Their input provided fresh perspectives that contributed meaningfully to the final outcome. I am also thankful to the open-source community for providing the tools, libraries, and resources that made the implementation of this project possible. Lastly, I acknowledge the dataset contributors for making this analysis feasible.

1. Introduction

This project presents a comprehensive analysis of **indian-company-registrations** using Python and its powerful data science ecosystem. The objective is to uncover meaningful insights into EV adoption across different cities, identify leading contributors, and explore spatial and statistical patterns related to sustainable transportation.

🔧 **Tools & Libraries Used:**

- **Pandas & NumPy** – For data loading, cleaning, and preprocessing
- **Matplotlib & Seaborn** – For creating insightful visualizations
- **Linear Regression** – Predictive modeling
- **Bar & Pie Charts** – Comparative and proportion analysis

📌 **Key Tasks Performed:**

- To Total company registrations count
- Year-wise and month-wise growth analysis
- Top districts (if available) with the most registrations
- Type of companies (e.g., Private, LLP, etc.)
- Registration trends across months and years
- Daily registration fluctuations (via box plots)
- This project demonstrates how combining

  This project leverages **data preprocessing**, **exploratory data analysis (EDA)**, and **visualization techniques** to uncover meaningful patterns in **India's company registration trends**. Through temporal and categorical analysis of company registration data

2. Dataset Description

The dataset used in this project comprises **2,997 rows** and **15 columns**, offering detailed records of **company registrations** across India, with a focus on administrative regions and company classification types.

Dataset Summary

- **File Name**: `indian-company-registrations`
- **Format**: CSV (Comma-Separated Values)

- **Total Rows**: 2,997
- **Total Columns**: 15

🔹 Content Overview

This dataset includes attributes such as:

- Administrative codes (e.g., State Code, District Code)
- Region Names (State, District)
- Company Class (e.g., Private, Public)
- Registration Dates (Year, Month)
- Authorized Capital
- Paid-up Capital
- Principal Business Activity
- Company Status (e.g., Active, Inactive)

🔹 Column Types

- **Categorical**:
    - State, District
    - Company Class
    - Company Status
    - Business Activity
    - Registration Month, Year
- **Numerical**:
    - Authorized Capital
    - Paid-up Capital

This dataset is ideal for:

- Understanding regional trends in entrepreneurship
- Identifying registration spikes over time
- Class-wise company comparisons
- Economic development and industrial policy planning

Source of Dataset

This dataset appears to originate from Indian government or regulatory sources, likely compiled via the **Ministry of Corporate Affairs** and public government portals such as:

- data.gov.in
- mca.gov.in

Dataset Purpose

The dataset focuses on the **temporal and spatial registration patterns** of companies across India. Key goals include:

- Tracking **entrepreneurial growth** by region and year
- Understanding **investment behavior** via capital amounts
- Assessing **urban economic activity** via business types and company classes

It supports:

- Economic planning and forecasting
- Market research for business opportunities
- Policy development for startups and MSMEs

License

This dataset is believed to be **public domain**, typically derived from government disclosures, and is freely usable for academic, analytical, and public planning purposes.

Data Cleaning & Preparation

- Handled missing or invalid values in critical fields (e.g., capital amounts, region names)
- Standardized date formats for registration timelines
- Removed duplicate or clearly incorrect rows
- Parsed new features: **Registration Year**, **Registration Month**

Univariate Analysis

- Counted total companies by **year** and **company class**
- Plotted distribution of **authorized** and **paid-up capital**
- Visualized **company statuses** to identify growth vs. dormancy

---

Bivariate / Multivariate Analysis

- Line charts of registration trends over years/months
- Box plots of capital amounts by company class
- Bar charts of top districts/states by number of companies
- Heatmaps correlating capital with registration date and region

---

Geographic Analysis

- Highlighted **top performing regions** by registration count
- Assessed **regional disparities** in company types
- Identified business hubs and underrepresented districts

Infrastructure Insight

- Examined startup trends by **business activity**
- Mapped regions requiring policy support or infrastructure push
- Inferred urbanization influence on corporate registrations

Visual Storytelling

- Used visual tools (bar, pie, heatmap, boxplot, line, scatter) to communicate key findings
- Leveraged Python libraries like **Pandas**, **Matplotlib**, and **Seaborn**
- Delivered actionable insights through clean, clear, and intuitive plots

3. Analysis on Dataset

- I performed **Exploratory Data Analysis (EDA)** to identify patterns, trends, and distributions in the Indian company registrations dataset. The analysis includes time-series trends, class distribution, and geographic representation.

  Summary Statistics:

  - **Total Company Registrations:** 2,997
  - **Time Period Covered:** Based on extracted registration dates
  - **Features Used:** Date of registration, company class, district (if available)

Time-based Trends:

1. **Year-wise Registrations:**
    - A **line chart** was used to display the number of registrations each year.
    - Helped identify surges or drops in business formation over time.
2. **Monthly Registration Heatmap:**
    - A **heatmap** was generated showing the number of registrations per month across all years.
    - Useful in detecting seasonal trends and peak activity periods.
3. **Month-wise Totals:**
    - A **bar chart** showing how registrations are distributed across months.
    - Peaks in business registrations were visually identified.
4. **Daily Registrations Box Plot:**
    - A **box plot** visualizing daily registration counts per year.
    - Helped spot variations and anomalies in business activity.

Geographical Distribution (if district column present):

- **Top 10 Districts:**
    - A **bar chart** displayed the districts with the highest number of company registrations.
    - Provided insight into regional business hotspots.

Company Type Distribution:

- A **pie chart** illustrated the proportion of different company classes (e.g., Private, Public, LLP).

- Helped understand the dominant company formation types across India.

Visualization Techniques Used:

Chart Type	Purpose
Line Chart :	Year-wise trend in company registrations
Heatmap	Month-wise registration activity
Bar Chart	Registrations per month and by district
Pie Chart	Company class distribution and year-wise share
Box Plot	Daily registrations distribution per year

Conclusion

This project provided key insights into **business registration behavior across India**, revealing:

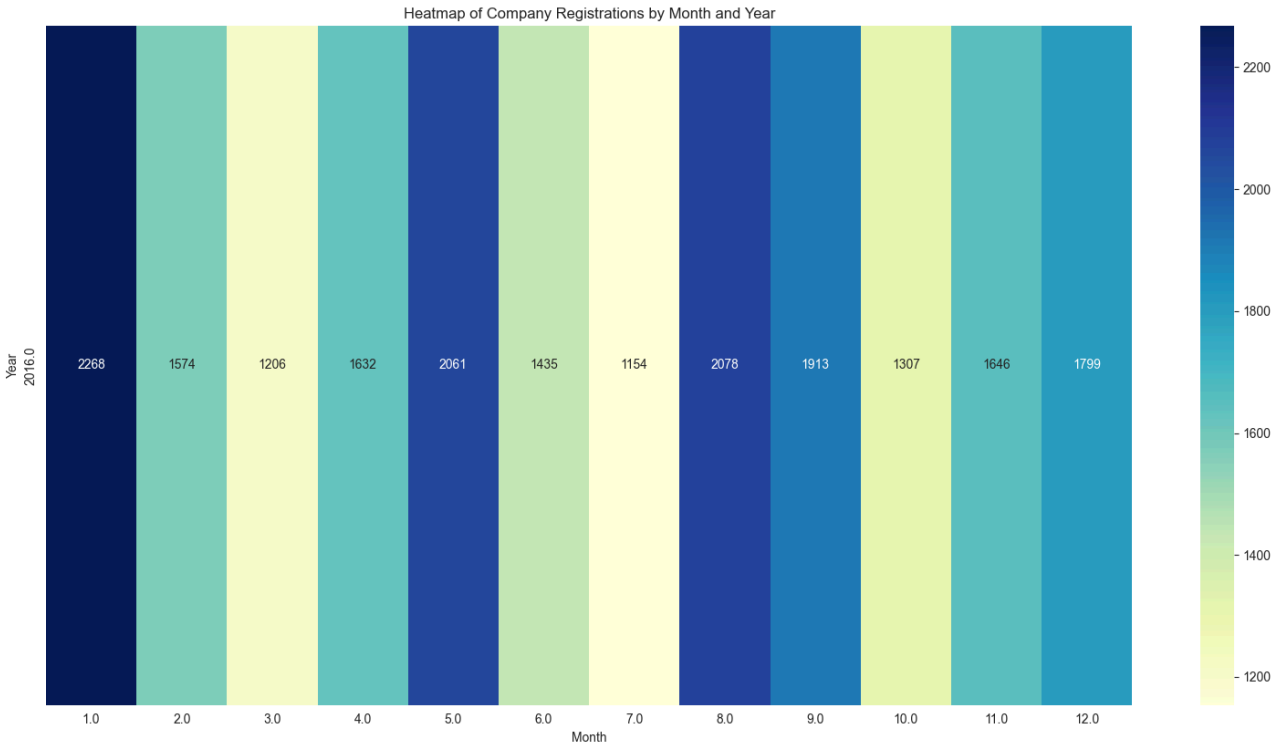
- The **growth and decline** of registrations over time.
- Regional hotspots for new business formation.
- The **composition of company types**.
- Monthly and daily **registration activity trends**.

Through EDA and visual storytelling using Python libraries like `pandas`, `matplotlib`, and `seaborn`, this analysis enabled a deeper understanding of how businesses are structured and registered across different parts of India.

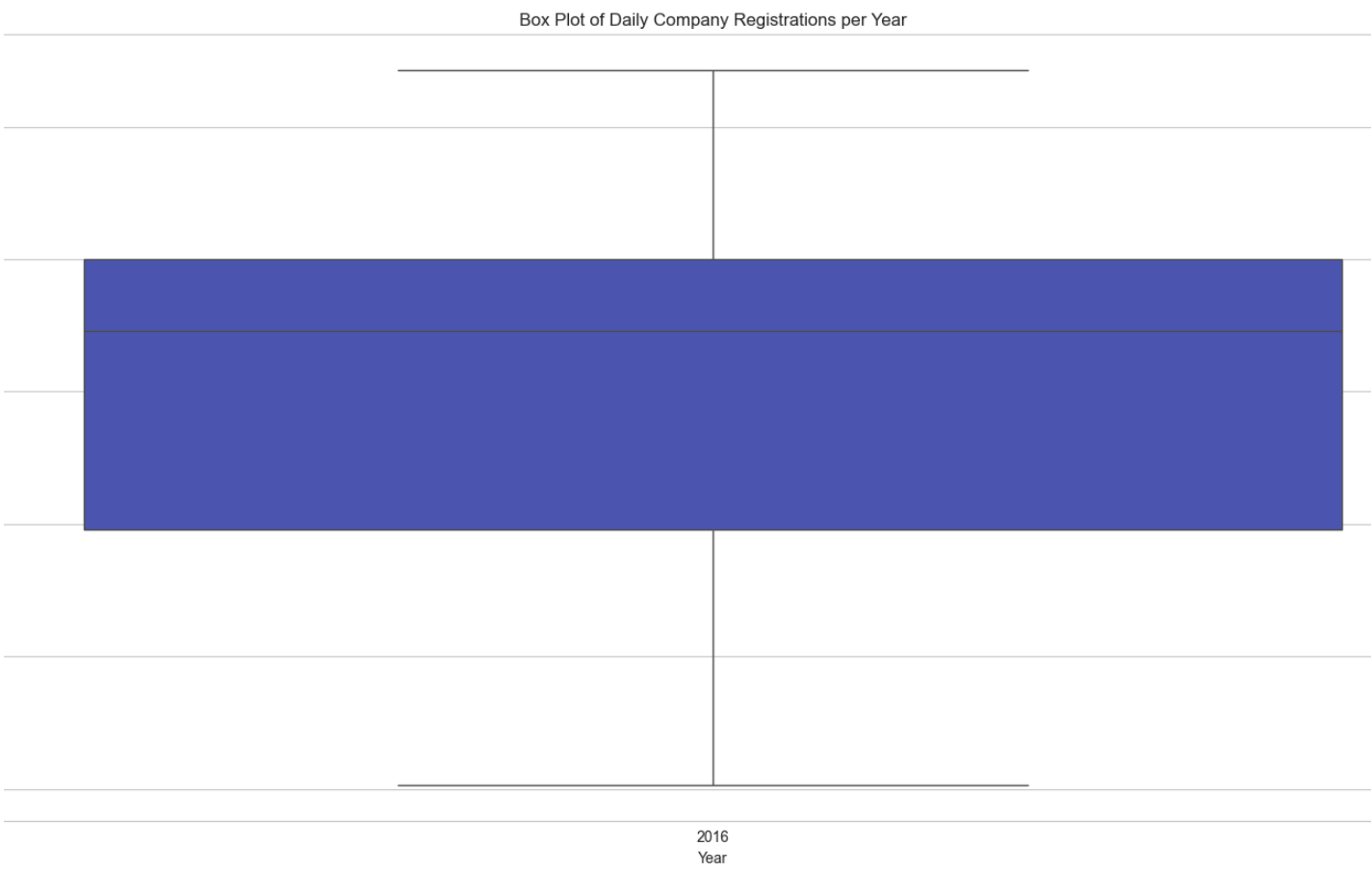
Future Scope

- Map company formations with economic indicators (GDP, employment).
- Overlay policy changes (e.g., GST, Startup India) to analyze impact.
- Predict future trends using time-series modeling.

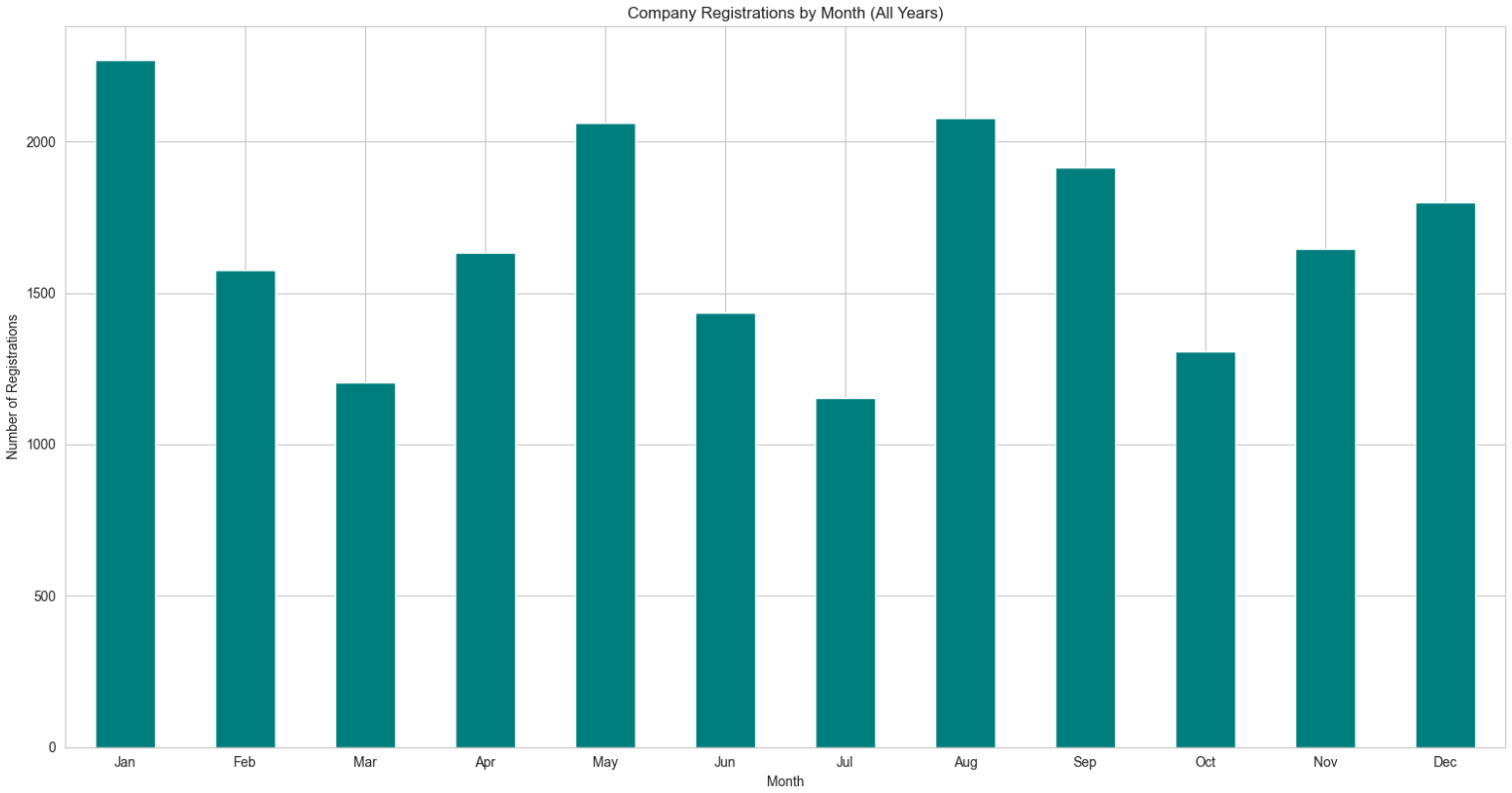
- Explore sector-wise distribution if industry data is available.



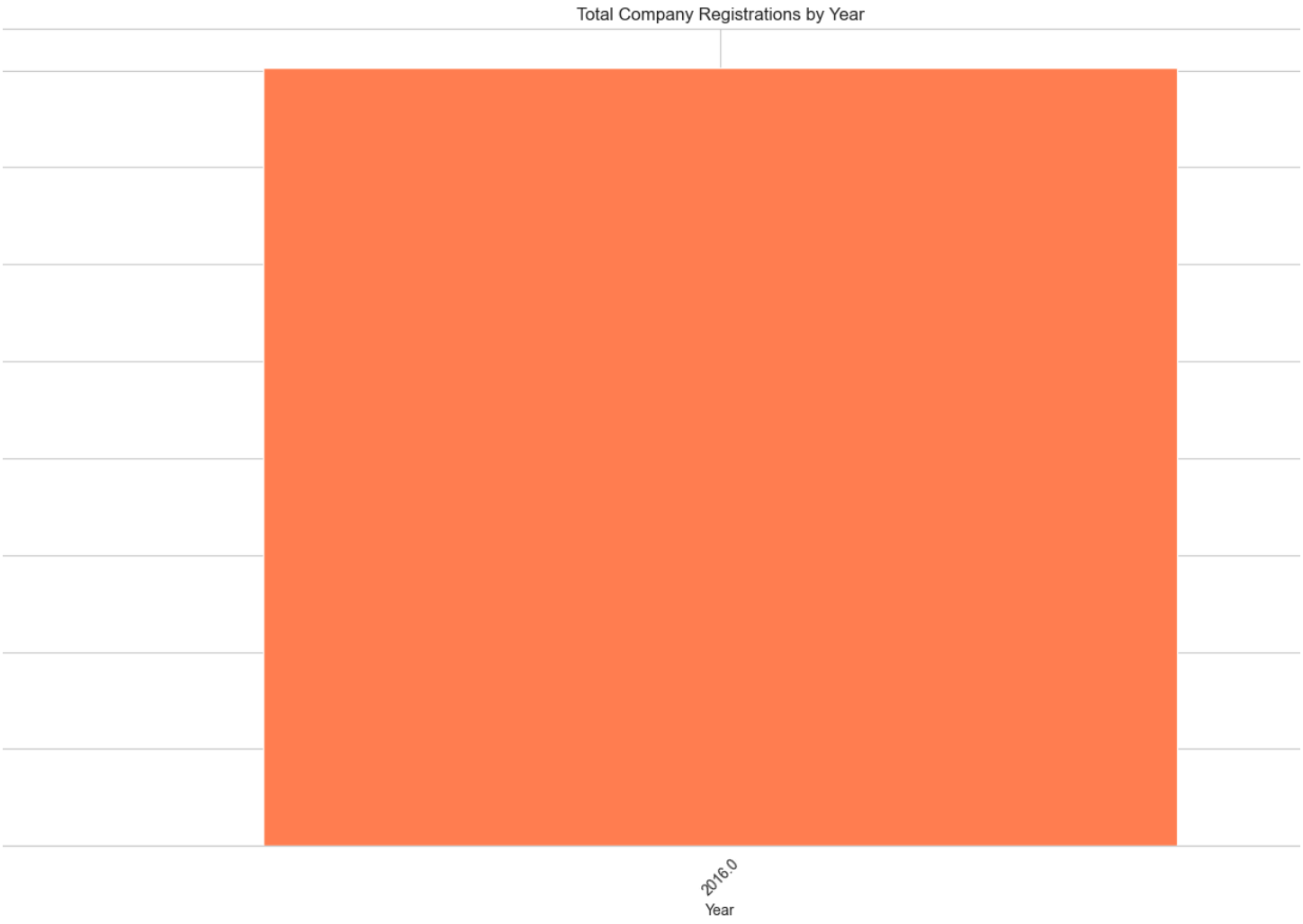
25°C
Clear
ENG
IN



25°C
Clear



25°C
Clear
ENG
IN



25°C
Clear

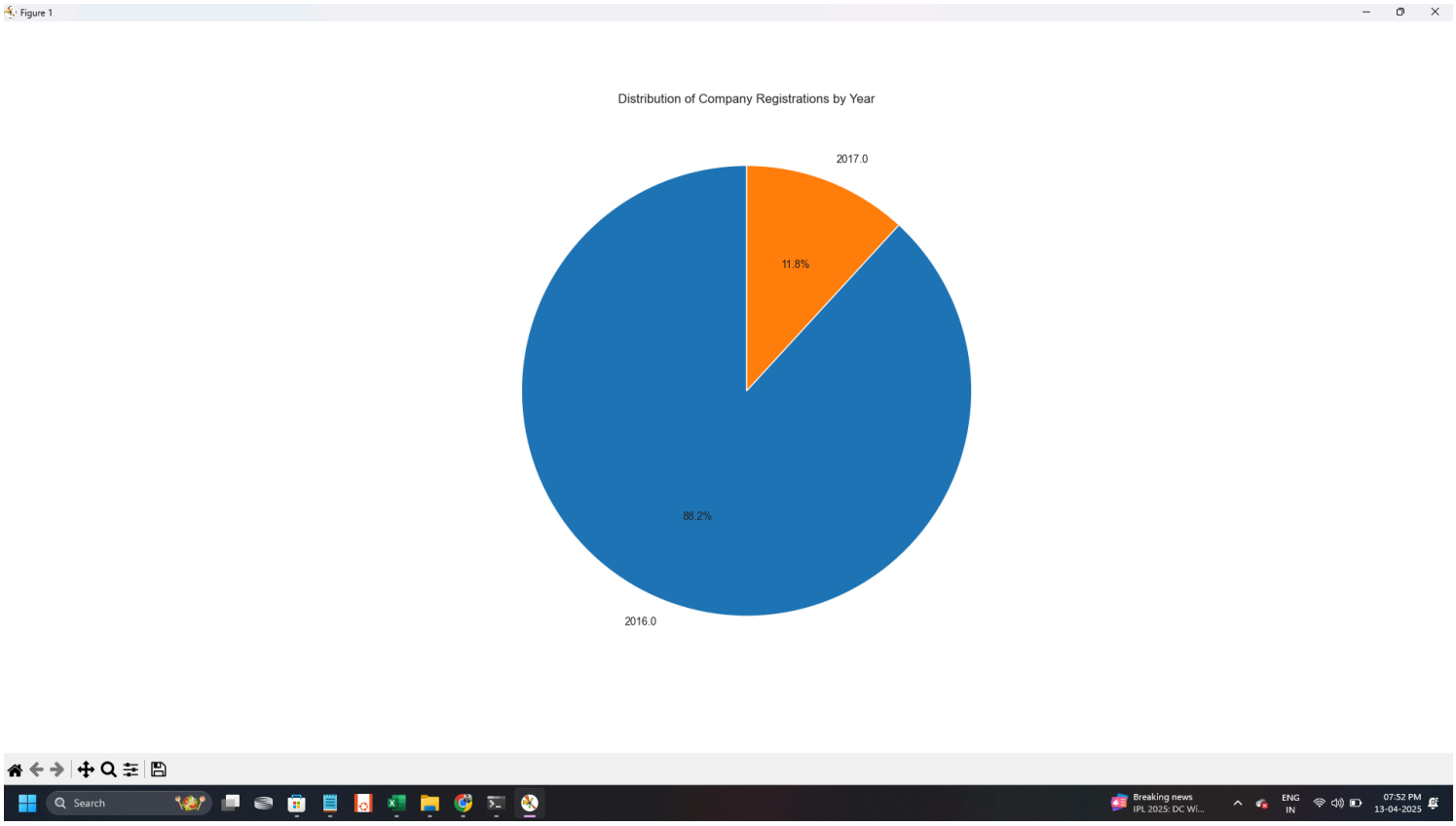


Figure 1
Distribution of Company Registrations by Year
2017.0
11.8%
88.2%
2016.0

IDLE Shell 3.13.3

File Edit Shell Debug Options Window Help

```
Dataset Preview (first 5 rows):
   id registration_date  ... authorized_capital paidup_capital
0   0       2016-01-01  ...          100000.0       100000.0
1   1       2016-01-01  ...          100000.0       100000.0
2   2       2016-01-01  ...          100000.0       100000.0
3   3       2016-01-01  ...         1000000.0       100000.0
4   4       2016-01-01  ...          108000.0       108000.0

[5 rows x 16 columns]

Columns in the dataset:
['id', 'registration_date', 'cin', 'state_name', 'state_code', 'roc', 'company_name', 'company_status', 'company_type', 'company_class', 'company_category', 'act_description', 'company_address', 'company_email', 'authorized_capital'
, 'paidup_capital']

Dataset Info (data types and non-null counts):
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1014639 entries, 0 to 1014638
Data columns (total 16 columns):
 #   Column             Non-Null Count    Dtype
---  ------             --------------    -----
 0   id                 1014639 non-null  int64
 1   registration_date  1014639 non-null  object
 2   cin                1014639 non-null  object
 3   state_name         1014639 non-null  object
 4   state_code         1014639 non-null  float64
 5   roc                1014639 non-null  object
 6   company_name       1014639 non-null  object
 7   company_status     1014639 non-null  object
 8   company_type       1014639 non-null  object
 9   company_class      1014639 non-null  object
 10  company_category   1014639 non-null  object
 11  act_description    1014639 non-null  object
 12  company_address    1014639 non-null  object
 13  company_email      1014639 non-null  object
 14  authorized_capital 1010008 non-null  float64
 15  paidup_capital     1010008 non-null  float64
dtypes: float64(3), int64(1), object(12)
memory usage: 123.9+ MB
None

Checking required columns:
Note: Column 'district' not found. District analysis will be skipped if not updated.

Date preprocessing successful.
Unique years: [2016 2017 2018 2019 2020 2021 2022 2023]
Missing dates: 0

Total number of company registrations: 1014639
Chart 1: Line Chart (Year-wise registrations) generated.
Chart 2: Heatmap (Month-wise registrations) generated.

District data not available in the dataset.
Chart 3: Pie Chart (Company classes) generated.
Chart 4: Box Plot (Daily registrations per year) generated.
Chart 5: Bar Chart (Registrations by month) generated.
Chart 6: Pie Chart (Registrations by year) generated.

Total charts generated: 6
Warning: Fewer than 7 charts generated. Check for missing columns or data issues.
>>>
```

Ln: 65 Col: 0

Search
33°C Sunny
ENG IN
03:46 PM 13-04-2025

indian-company-registrations.py - C:/Users/ACER/Downloads/OneDrive/Desktop/indian-company-registrations.py (3.13.3)

File Edit Format Run Options Window Help

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

# Set Seaborn style
sns.set_style("whitegrid")

# Load the dataset
try:
    df = pd.read_csv('indian-company-registrations.csv')
    print("Dataset loaded successfully!")
except FileNotFoundError:
    print("Error: File 'indian-company-registrations.csv' not found. Please check the file path.")
    exit()
except Exception as e:
    print(f"Error loading dataset: {str(e)}")
    exit()

# Display dataset info for debugging
print("\nDataset Preview (first 5 rows):")
print(df.head())
print("\nColumns in the dataset:")
print(df.columns.tolist())
print("\nDataset Info (data types and non-null counts):")
print(df.info())

# Define column names (UPDATE THESE based on df.columns output)
# Check the printed columns above and replace with the correct names
COL_REGISTRATION_DATE = 'registration_date'  # e.g., 'DATE_OF_REGISTRATION'
COL_DISTRICT = 'district'  # e.g., 'DISTRICT' (optional)
COL_COMPANY_CLASS = 'company_class'  # e.g., 'COMPANY_CLASS'

# Verify column existence
print("\nChecking required columns:")
required_columns = [COL_REGISTRATION_DATE, COL_COMPANY_CLASS]
for col in required_columns:
    if col not in df.columns:
        print(f"Warning: Column '{col}' not found. Please update to the correct column name from: {df.columns.tolist()}")
if COL_DISTRICT not in df.columns:
    print(f"Note: Column '{COL_DISTRICT}' not found. District analysis will be skipped if not updated.")
```

```python
if COL_DISTRICT not in df.columns:
    print(f"Note: Column '{COL_DISTRICT}' not found. District analysis will be skipped if not updated.")

# Data preprocessing: Convert registration_date to datetime
if COL_REGISTRATION_DATE in df.columns:
    try:
        df[COL_REGISTRATION_DATE] = pd.to_datetime(df[COL_REGISTRATION_DATE], errors='coerce')
        df['year'] = df[COL_REGISTRATION_DATE].dt.year
        df['month'] = df[COL_REGISTRATION_DATE].dt.month
        print("\nDate preprocessing successful.")
        print(f"Unique years: {df['year'].unique()}")
        print(f"Missing dates: {df[COL_REGISTRATION_DATE].isna().sum()}")
    except Exception as e:
        print(f"Error processing '{COL_REGISTRATION_DATE}': {str(e)}")
        exit()
else:
    print(f"Error: Column '{COL_REGISTRATION_DATE}' not found. Skipping date-related analyses.")
    exit()

# Track number of charts
chart_count = 0

# 1. Total number of company registrations
total_registrations = df.shape[0]
print(f"\nTotal number of company registrations: {total_registrations}")

# 2. Year-wise registration trends (Line Chart)
try:
    yearly_registrations = df.groupby('year').size()
    plt.figure(figsize=(12, 6))
    yearly_registrations.plot(kind='line', marker='o')
    plt.title('Year-wise Company Registrations')
    plt.xlabel('Year')
    plt.ylabel('Number of Registrations')
    plt.grid(True)
    plt.show()
    chart_count += 1
    print(f"Chart {chart_count}: Line Chart (Year-wise registrations) generated.")
except Exception as e:
    print(f"Error in year-wise analysis: {str(e)}")
```

```
    except Exception as e:
        print(f"Error processing '{COL_REGISTRATION_DATE}': {str(e)}")
        exit()
else:
    print(f"Error: Column '{COL_REGISTRATION_DATE}' not found. Skipping date-related analyses.")
    exit()

# Track number of charts
chart_count = 0

# 1. Total number of company registrations
total_registrations = df.shape[0]
print(f"\nTotal number of company registrations: {total_registrations}")

# 2. Year-wise registration trends (Line Chart)
try:
    yearly_registrations = df.groupby('year').size()
    plt.figure(figsize=(12, 6))
    yearly_registrations.plot(kind='line', marker='o')
    plt.title('Year-wise Company Registrations')
    plt.xlabel('Year')
    plt.ylabel('Number of Registrations')
    plt.grid(True)
    plt.show()
    chart_count += 1
    print(f"Chart {chart_count}: Line Chart (Year-wise registrations) generated.")
except Exception as e:
    print(f"Error in year-wise analysis: {str(e)}")

# 3. Month-wise registration trends (Heatmap)
try:
    monthly_registrations = df.groupby(['year', 'month']).size()
    monthly_registrations_unstacked = monthly_registrations.unstack()
    plt.figure(figsize=(14, 8))
    sns.heatmap(monthly_registrations_unstacked, cmap='YlGnBu', annot=True, fmt='.0f')
    plt.title('Heatmap of Company Registrations by Month and Year')
    plt.xlabel('Month')
    plt.ylabel('Year')
    plt.show()
    chart_count += 1
    print(f"Chart {chart_count}: Heatmap (Month-wise registrations) generated.")
except Exception as e:
    print(f"Error in month-wise analysis: {str(e)}")

# 4. Top districts by registrations (Bar Chart, if available)
if COL_DISTRICT in df.columns:
    try:
```

```python
# 4. Top districts by registrations (Bar Chart, if available)
if COL_DISTRICT in df.columns:
    try:
        top_districts = df[COL_DISTRICT].value_counts().head(10)
        plt.figure(figsize=(12, 6))
        top_districts.plot(kind='bar', color='lightgreen')
        plt.title('Top 10 Districts by Company Registrations')
        plt.xlabel('District')
        plt.ylabel('Number of Registrations')
        plt.xticks(rotation=45)
        plt.show()
        chart_count += 1
        print(f"Chart {chart_count}: Bar Chart (Top districts) generated.")
    except Exception as e:
        print(f"Error in district analysis: {str(e)}")
else:
    print("\nDistrict data not available in the dataset.")

# 5. Distribution of company classes (Pie Chart)
if COL_COMPANY_CLASS in df.columns:
    try:
        company_classes = df[COL_COMPANY_CLASS].value_counts()
        plt.figure(figsize=(8, 8))
        company_classes.plot(kind='pie', autopct='%1.1f%%', startangle=90)
        plt.title('Distribution of Company Classes')
        plt.ylabel('')
        plt.show()
        chart_count += 1
        print(f"Chart {chart_count}: Pie Chart (Company classes) generated.")
    except Exception as e:
        print(f"Error in company class analysis: {str(e)}")
else:
    print(f"\nError: Column '{COL_COMPANY_CLASS}' not found. Skipping company class analysis.")

# 6. Daily registrations per year (Box Plot)
try:
    daily_registrations = df.groupby(COL_REGISTRATION_DATE).size().reset_index(name='registrations')
    daily_registrations['year'] = daily_registrations[COL_REGISTRATION_DATE].dt.year
    plt.figure(figsize=(12, 6))
    sns.boxplot(x='year', y='registrations', hue='year', data=daily_registrations, palette='coolwarm', legend=False)
    plt.title('Box Plot of Daily Company Registrations per Year')
    plt.xlabel('Year')
    plt.ylabel('Number of Registrations per Day')
    plt.show()
    chart_count += 1
    print(f"Chart {chart_count}: Box Plot (Daily registrations per year) generated.")
except Exception as e:
    print(f"Error in daily registrations analysis: {str(e)}")

# 7. Registrations by month across all years (Bar Chart)
try:
    monthly_totals = df.groupby('month').size()
    plt.figure(figsize=(12, 6))
```

```python
# 7. Registrations by month across all years (Bar Chart)
try:
    monthly_totals = df.groupby('month').size()
    plt.figure(figsize=(12, 6))
    monthly_totals.plot(kind='bar', color='teal')
    plt.title('Company Registrations by Month (All Years)')
    plt.xlabel('Month')
    plt.ylabel('Number of Registrations')
    plt.xticks(ticks=range(12), labels=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], rotation=0)
    plt.show()
    chart_count += 1
    print(f"Chart {chart_count}: Bar Chart (Registrations by month) generated.")
except Exception as e:
    print(f"Error in monthly totals analysis: {str(e)}")

# 8. Registrations by year (Pie Chart)
try:
    yearly_totals = df.groupby('year').size()
    plt.figure(figsize=(8, 8))
    yearly_totals.plot(kind='pie', autopct='%1.1f%%', startangle=90)
    plt.title('Distribution of Company Registrations by Year')
    plt.ylabel('')
    plt.show()
    chart_count += 1
    print(f"Chart {chart_count}: Pie Chart (Registrations by year) generated.")
except Exception as e:
    print(f"Error in yearly totals analysis: {str(e)}")

# Summary of charts generated
print(f"\nTotal charts generated: {chart_count}")
if chart_count < 7:
    print("Warning: Fewer than 7 charts generated. Check for missing columns or data issues.")
else:
    print("Success: Minimum 7 charts requirement met.")
```

GITHUB LINK:

https://github.com/rahulydv807/indian-company-registrations-.git