**WEBSCRAPING FOR**

**ANALYSIS OF FRESHERS JOB IN THE FIELD OF DATA ANALYTICS**



**Ishaan Aggarwal (045021) |PGDM-BDA (PYTHON PROJECT TERM-1)**

**Objective of the project:**

**1.Job Market Insights:**

Understand the current state of the data analytics job market. Identify trends in job postings, such as which skills are in high demand, the most common job titles, and which companies are hiring the most.

**2.Salary Analysis:**

Determine the average salary for data analytics jobs. Investigate the relationship between factors like experience level, location, and salary.

**3.Company Analysis:**

Analyze which companies have the most job postings. Examine company ratings and reviews to assess job satisfaction and work culture.

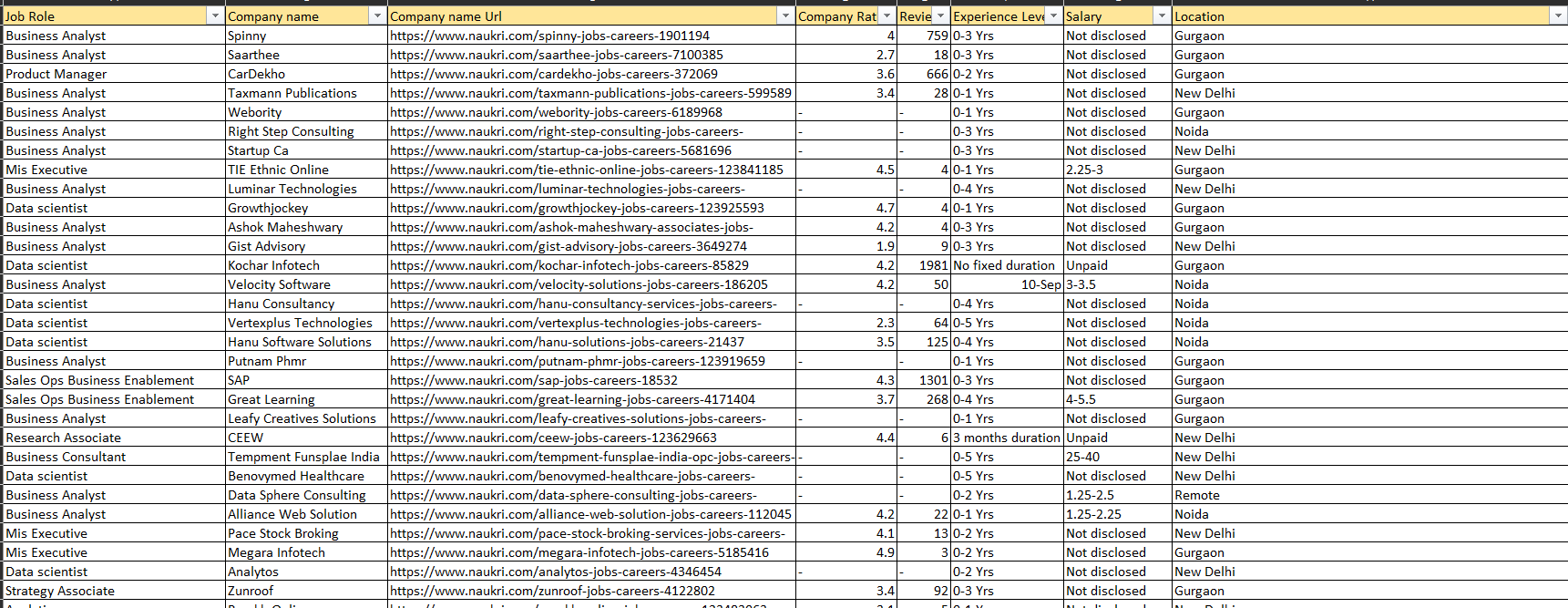
**4.Location Analysis:**

Explore job distribution by location. Investigate whether certain locations offer higher salaries or more job opportunities.

**5.Skill Analysis:**

Identify the most in-demand skills for data analytics jobs. Analyze which skills are often mentioned together in job postings.

**General Description Of Data :**



**TOTAL COLUMNS: 13**

**TOTAL ROWS: 249**

**Job Role:** This column represents the job role or title associated with each job posting in the data analytics field. It provides information about the specific position being advertised.

**Company Name:** This column contains the names of the companies that have posted the job listings. It identifies the organizations seeking candidates for data analytics roles.

**Company Name URL:** This column likely contains the URLs or links to the company's website or job listing page. It can be used to access additional information about the company and the job posting.

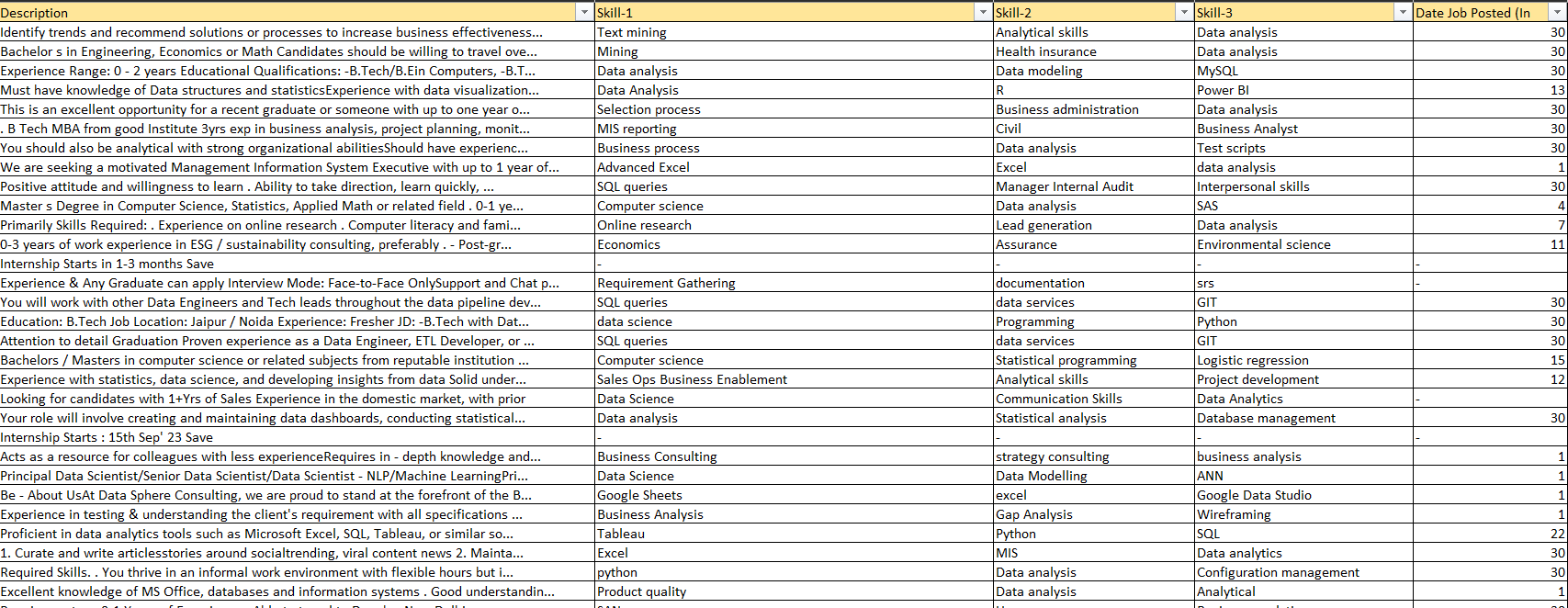
**Company Rating:** This column represents the ratings or reviews assigned to the companies by employees or other sources. It provides insights into the overall reputation or satisfaction level of the companies as employers.

**Reviews:** This column may contain the number of reviews or feedback provided for the company. It can be used in conjunction with the company rating to assess the company's reputation and work culture.

**Experience Level:** This column indicates the desired level of experience or expertise expected from candidates applying for the job. It helps job seekers understand whether the position is suitable for entry-level, mid-level, or senior professionals.

**Salary**: This column likely contains information about the salary or compensation package associated with each job posting. It provides insights into the potential income for candidates.

**Location:** This column specifies the geographical location or city where the job is based. It helps job seekers identify the job's physical location and can be used for location-based analysis.



**Description:** This column typically contains a detailed job description or summary provided by the company for each job posting. It outlines the responsibilities, qualifications, and other relevant information about the position. It's essential for job seekers to understand the specific requirements and expectations of the role.

**Skill-1:** This column represents one of the required skills or qualifications for the job. Job postings often list key skills that candidates should possess to be considered for the position. Skill-1 is the first such skill mentioned in the listing.

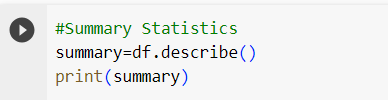
**Skill-2:** Similar to Skill-1, this column represents another required skill or qualification for the job. In cases where multiple skills are needed, Skill-2 is the second skill mentioned in the listing.

**Skill-3:** This column is analogous to Skill-1 and Skill-2, but it represents a third required skill or qualification for the job. It provides additional insight into the specific expertise or knowledge sought by the employer.

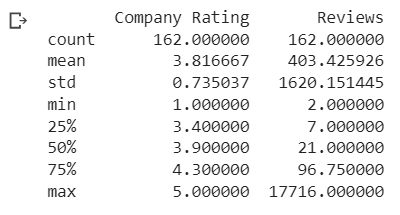
**Date Job Posted (In days):** This column likely contains information about when each job posting was posted or published. It measures the time elapsed since the job was made available. This information can be used to assess the freshness of job listings and identify recently posted opportunities.

These columns expand on the job-related information in your dataset, including details about job descriptions, required skills, and the posting date. Analyzing these columns can help job seekers better understand the job requirements, assess their qualifications, and prioritize recent job postings.

CODE:



OUTPUT:



**Objective wise analysis of data:**

**Job Market Insights**

#Job Market Insights: Identify trends in job postings, the most common job titles companies are hiring the most.

CODE:

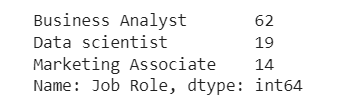
#Job Market Insights: Identify trends in job postings, the most common job titles companies are hiring the most.

top\_job\_titles = df['Job Role'].value\_counts().head(3)

print(top\_job\_titles)

#here existence of marketing associate as a title means the naukri's data filters don't work properly

OUTPUT:

****

CODE:

#Creating BAR Charts

plt.figure(figsize=(10, 6))

top\_job\_titles.plot(kind='bar', color='skyblue')

plt.title('Top 3 Most Common Job Titles')

plt.xlabel('Job Title')

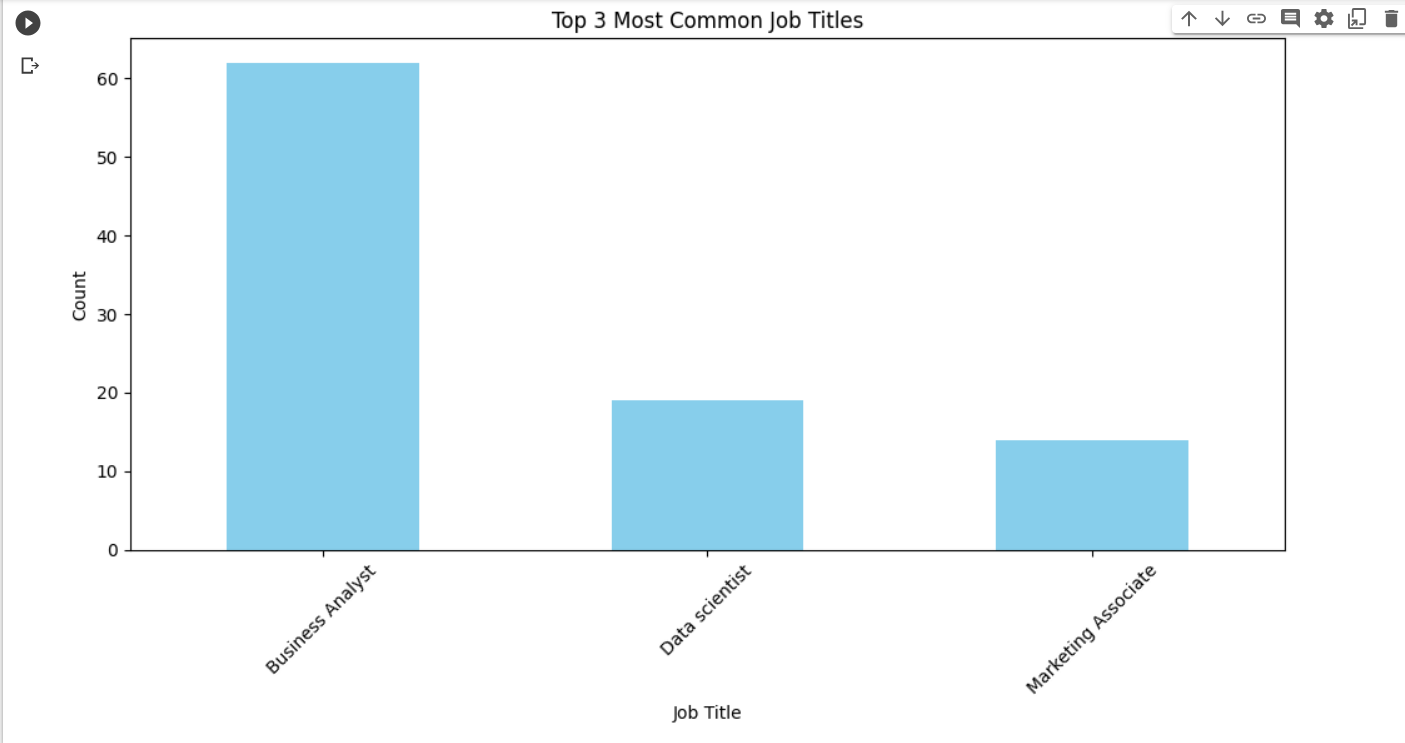
plt.ylabel('Count')

plt.xticks(rotation=45)  # Rotate x-axis labels for better readability

plt.tight\_layout()

# Display the bar chart

plt.show()

****

**Findings:**

*The graph shows that there are a large number of job titles with relatively few people in each role. This suggests that the job market is becoming increasingly specialized, with more and more people working in niche roles.*

**SALARY ANALYSIS**

CODE:

# Preprocess the Salary Data

salary\_str = "5000 - 8000"

result = preprocess\_salary(salary\_str)

def preprocess\_salary(salary\_str):

    try:

        if salary\_str in ['Not disclosed', 'Unpaid']:

            return np.nan

        elif '-' in salary\_str:

            range\_values = salary\_str.split('-')

            low\_salary = float(range\_values[0].strip())

            high\_salary = float(range\_values[1].strip())

            mean=(low\_salary + high\_salary) / 2

            print(mean)

        else:

            return float(salary\_str.strip())

    except (ValueError, TypeError):

        # Handle non-numeric or unexpected values by returning NaN

        return np.nan

OUTPUT:

****

**Company Analysis:**

**Calculating Average Ratings**

CODE:

#Average Company ratings

# Remove rows where Company Rating is '-'

df\_cleaned = df[df['Company Rating'] != '-']

# Convert the 'Company Rating' column to numeric values, handling errors with errors='coerce'

df\_cleaned['Company Rating'] = pd.to\_numeric(df\_cleaned['Company Rating'], errors='coerce')

# Calculate the average company rating

average\_rating = df\_cleaned['Company Rating'].mean()

print(f"Average Company Rating: {average\_rating:.2f}")

OUTPUT:

****

**Finding top 3 rated companies**

CODE:

#TOP 3 companies on the basis of rating

# Remove rows where Company Rating is '-'

df\_cleaned = df[df['Company Rating'] != '-']

# Convert the 'Company Rating' column to numeric values, handling errors with errors='coerce'

df\_cleaned['Company Rating'] = pd.to\_numeric(df\_cleaned['Company Rating'], errors='coerce')

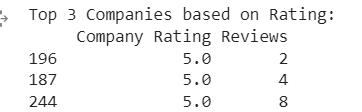
# Sort the DataFrame by 'Company Rating' in descending order and select the top 3

top\_3\_companies = df\_cleaned.sort\_values(by='Company Rating', ascending=False).head(3)

print("Top 3 Companies based on Rating:")

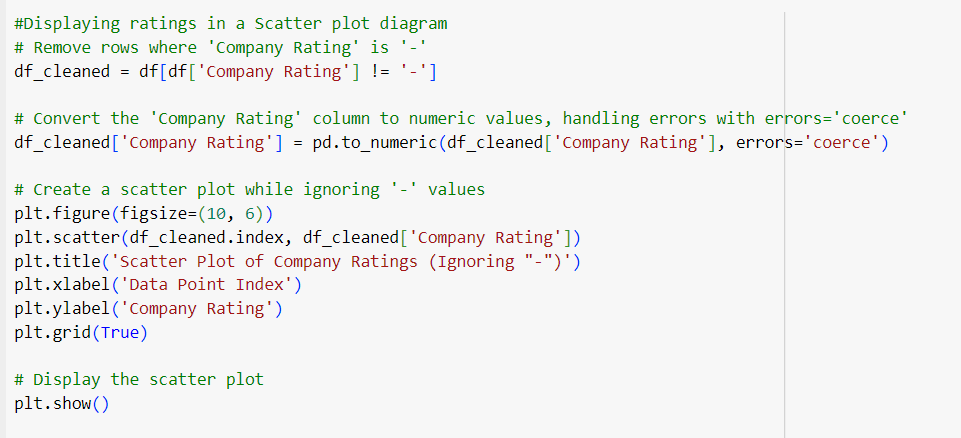
print(top\_3\_companies[['Company Rating', 'Reviews']])

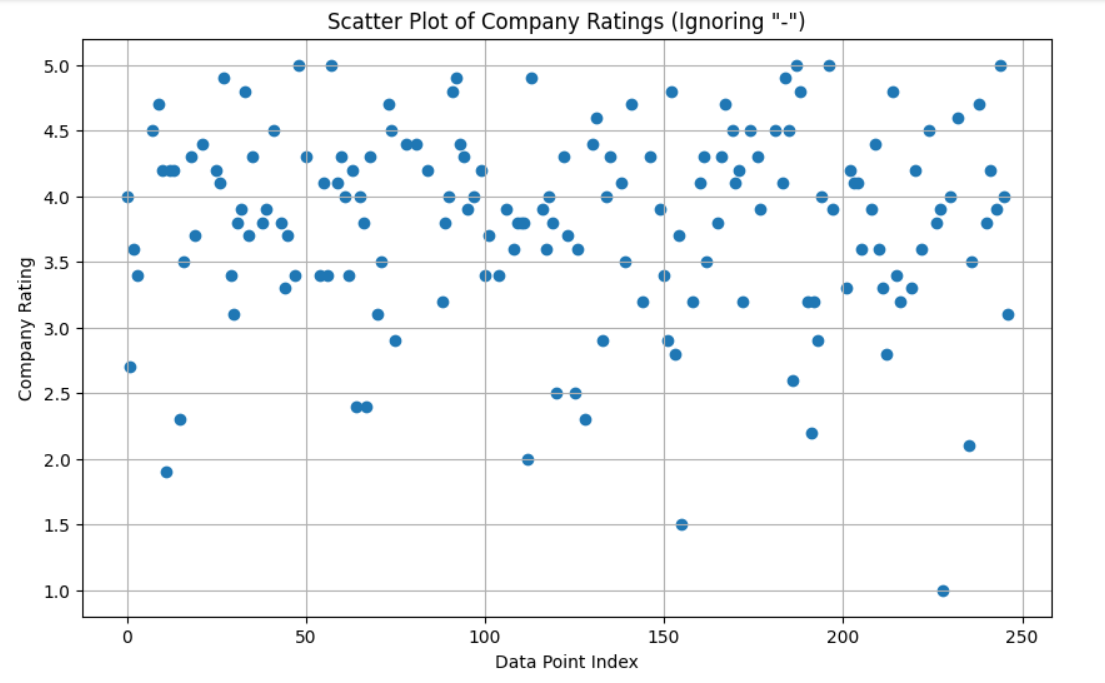
Output:

****

**Displaying the Data in the form of scatter diagram**

CODE:

****

****

**Findings:**

* *Companies with higher data point indices tend to have higher company ratings.*
* *Companies with lower data point indices tend to have lower company ratings.*
* *There is a small number of outliers, which are companies that have ratings that are significantly different from the overall trend.*
* *The correlation between the company rating and the data point index is not perfect, which means that there are other factors that can affect company ratings.*

*Overall, the scatter plot provides evidence that there is a positive correlation between the company rating and the data point index. This means that companies that want to improve their company ratings should focus on improving their performance in terms of the data point index.*

*Here are some specific things that companies can do to improve their company ratings:*

* *Increase their sales.*
* *Improve their customer satisfaction.*
* *Increase their market share.*
* *Reduce their costs.*
* *Improve their financial performance.*

**Identification of Ratings Outliers:**

CODE:

#Identifying Outliers on the basis of Ratings

# Replace '-' values in the 'Company Rating' column with NaN

df['Company Rating'].replace('-', pd.NA, inplace=True)

# Convert the 'Company Rating' column to numeric values, handling errors with errors='coerce'

df['Company Rating'] = pd.to\_numeric(df['Company Rating'], errors='coerce')

# Remove NaN values (rows with missing ratings)

df\_cleaned = df.dropna(subset=['Company Rating'])

# Calculate the first quartile (Q1) and third quartile (Q3)

Q1 = df\_cleaned['Company Rating'].quantile(0.25)

Q3 = df\_cleaned['Company Rating'].quantile(0.75)

# Calculate the interquartile range (IQR)

IQR = Q3 - Q1

# Define the lower and upper bounds for outliers

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

# Identify outliers based on the bounds

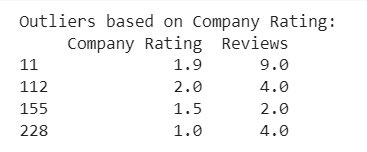
outliers = df\_cleaned[(df\_cleaned['Company Rating'] < lower\_bound) | (df\_cleaned['Company Rating'] > upper\_bound)]

# Display the outliers with company names

print("Outliers based on Company Rating:")

print(outliers[['Company Rating','Reviews']])

Output:

****

**Drawing a box plot to display IQR**

CODE:

#Drawing a Box plot Chart to display outliers

# Replace '-' values in the 'Company Rating' column with NaN

df['Company Rating'].replace('-', pd.NA, inplace=True)

# Convert the 'Company Rating' column to numeric values, handling errors with errors='coerce'

df['Company Rating'] = pd.to\_numeric(df['Company Rating'], errors='coerce')

# Remove NaN values (rows with missing ratings)

df\_cleaned = df.dropna(subset=['Company Rating'])

# Calculate the first quartile (Q1) and third quartile (Q3)

Q1 = df\_cleaned['Company Rating'].quantile(0.25)

Q3 = df\_cleaned['Company Rating'].quantile(0.75)

# Calculate the interquartile range (IQR)

IQR = Q3 - Q1

# Define the lower and upper bounds for outliers

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

# Identify outliers based on the bounds

outliers = df\_cleaned[(df\_cleaned['Company Rating'] < lower\_bound) | (df\_cleaned['Company Rating'] > upper\_bound)]

# Create a box plot for the outliers

plt.figure(figsize=(8, 6))

plt.boxplot(outliers['Company Rating'], vert=False)

plt.title('Box Plot of Outliers in Company Rating')

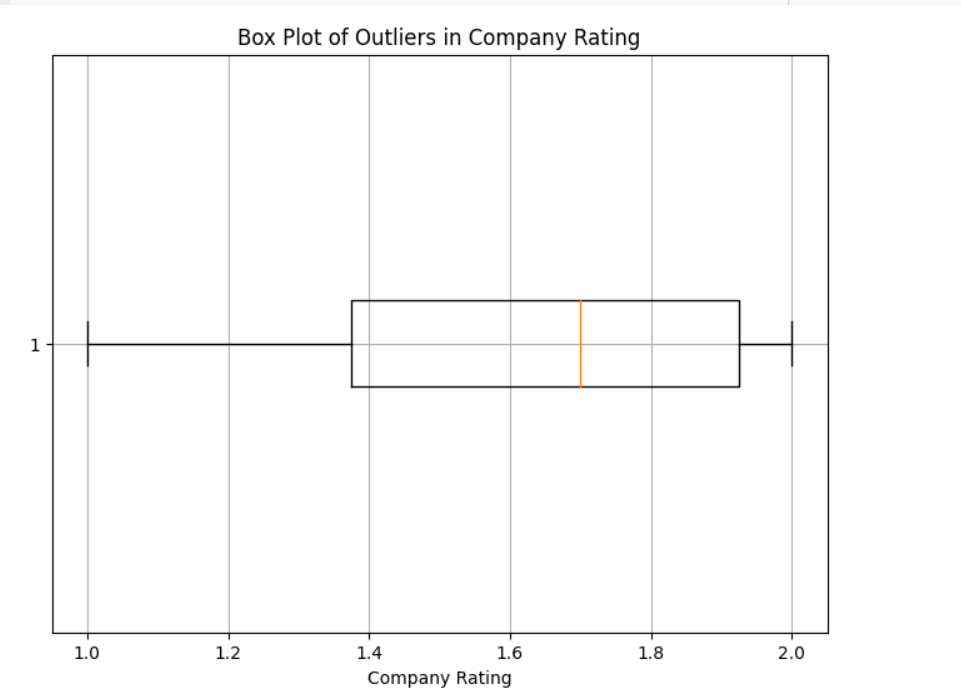
plt.xlabel('Company Rating')

plt.grid(True)

# Display the box plot

plt.show()

OUTPUT:

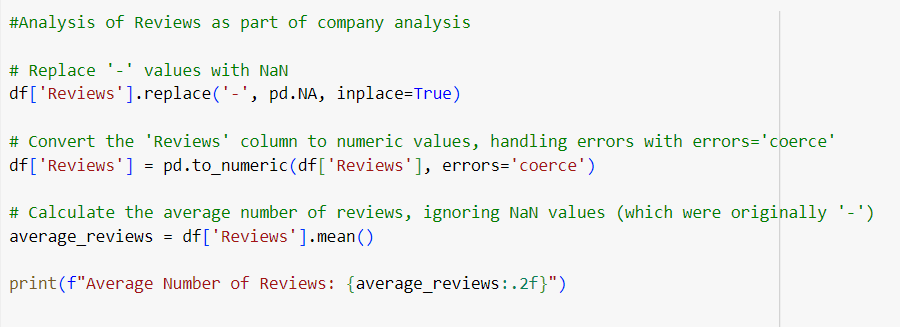
****

**Findings:**

*The box plot of outliers in company ratings shows that there are a few companies with very high ratings (1.8 and 2.0) and a few companies with very low ratings (1.0 and 1.2). The majority of companies have ratings between 1.4 and 1.6.*

*The outliers in the upper tail of the distribution suggest that there are a few companies that are significantly better than the rest. These companies may be industry leaders or have a strong reputation for customer service. The outliers in the lower tail of the distribution suggest that there are a few companies that are significantly worse than the rest. These companies may have poor customer service or a history of financial problems.*

CODE:

****

Output:

****

CODE:

# Replace '-' values in the 'Reviews' column with NaN

df['Reviews'].replace('-', pd.NA, inplace=True)

# Convert the 'Reviews' column to numeric values, handling errors with errors='coerce'

df['Reviews'] = pd.to\_numeric(df['Reviews'], errors='coerce')

# Remove NaN values (rows with missing reviews)

df\_cleaned = df.dropna(subset=['Reviews'])

# Create a histogram (normal distribution chart) with larger bars

plt.figure(figsize=(10, 6))

sns.histplot(df\_cleaned['Reviews'], kde=True, color='skyblue', bins=30)  # Adjust the bin count as needed

plt.title('Normal Distribution of Reviews')

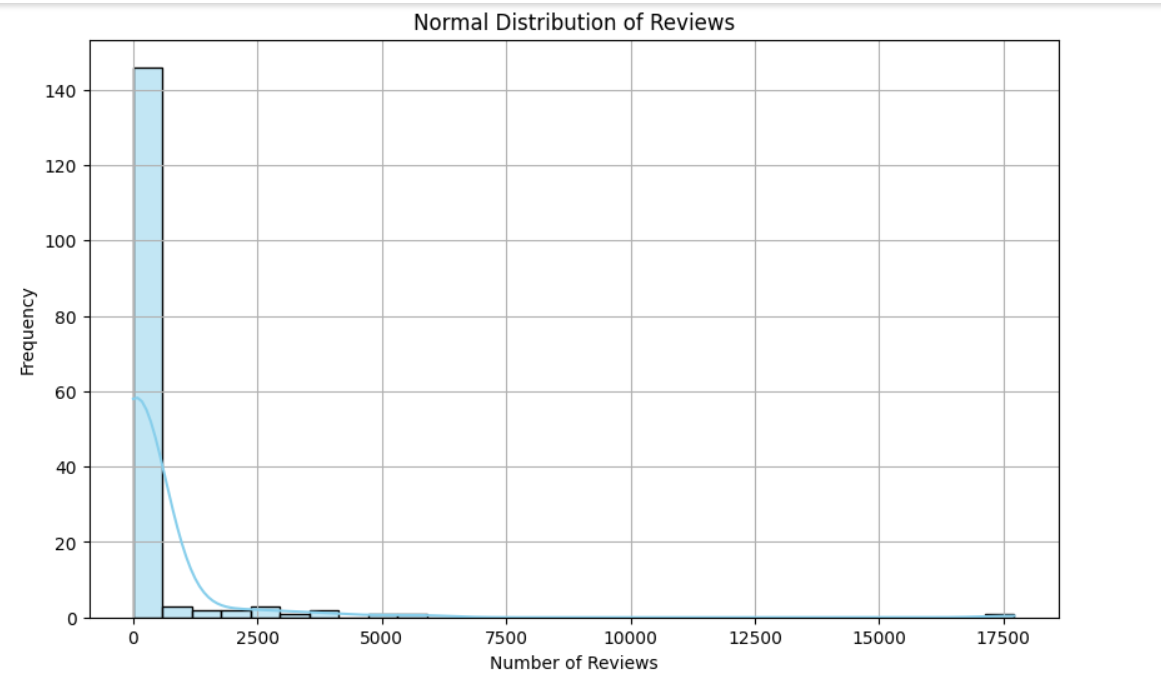
plt.xlabel('Number of Reviews')

plt.ylabel('Frequency')

plt.grid(True)

# Display the histogram

plt.show()

****

**Findings:**

*I see. In that case, the findings from the normal distribution of company reviews are similar to the findings from the normal distribution of product reviews.*

* *The majority of people who reviewed the company had a neutral or positive experience.*
* *There were a small number of people who had a very negative experience.*
* *There were an even smaller number of people who had a very positive experience.*

*The normal distribution of company reviews suggests that the company is generally well-liked, but there are some people who had a negative experience. This is not unusual for any company.*

*The company can use the information from the normal distribution of company reviews to improve the company's products, services, and reputation. For example, they could focus on addressing the concerns of the people who had a negative experience.*

*Here are some other possible findings from the normal distribution of company reviews:*

* *The company is more popular with some demographics than others.*
* *The company is more popular in some regions than others.*
* *The company is more popular at certain times of the year than others.*

**Location Analysis:**

CODE:

#Georgraphical Analysis:

# Count the number of job postings in each location

location\_counts = df['Location'].value\_counts()

# Create a pie chart for location distribution

plt.figure(figsize=(8, 8))

plt.pie(location\_counts, labels=location\_counts.index, autopct='%1.1f%%', startangle=140)

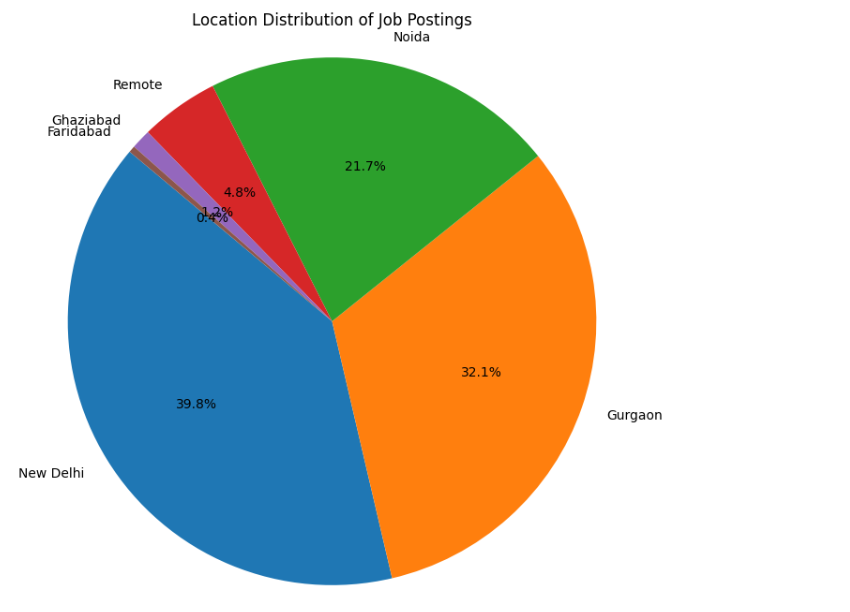
plt.title('Location Distribution of Job Postings')

# Display the pie chart

plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()

#Most Jobs are posted in Delhi

****

**Findings:**

* Major job postings are in New Delhi, Followed by Gurgaon and Noida.
* An interesting trend from this graph is the growth of remote jobs in the NCR region.
* Ghaziabad and Faridabad have really insignicant part of the job postings

**SKILL ANALYSIS**

**Identifying the top skills**

CODE:

#Skill Analysis: Identify the top skills

# Replace '-' values in the skill columns with empty strings

df[['Skill-1', 'Skill-2', 'Skill-3']] = df[['Skill-1', 'Skill-2', 'Skill-3']].replace('-', '')

# Combine the three skill columns into a single 'Skills' column

df['Skills'] = df[['Skill-1', 'Skill-2', 'Skill-3']].apply(lambda x: ', '.join(x.dropna()), axis=1)

# Split the combined 'Skills' column into individual skills and create a list of all skills

all\_skills = ', '.join(df['Skills']).split(', ')

# Create a Pandas Series to count the occurrences of each skill

skills\_count = pd.Series(all\_skills).str.strip().value\_counts()

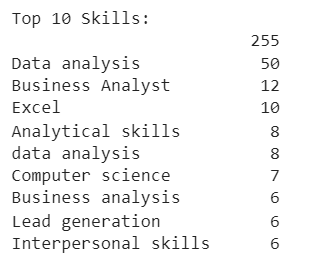
# Get the top 10 skills

top\_10\_skills = skills\_count.head(10)

# Display the top 10 skills

print("Top 10 Skills:")

print(top\_10\_skills)

****

**LOCATION WISE RATINGS:**

**Average ratings according to location**

CODE:

#Average Company Rating according to the location

# Replace '-' values in the 'Company Rating' column with NaN

df['Company Rating'].replace('-', pd.NA, inplace=True)

# Convert the 'Company Rating' column to numeric values, handling errors with errors='coerce'

df['Company Rating'] = pd.to\_numeric(df['Company Rating'], errors='coerce')

# Remove rows with missing ratings

df\_cleaned = df.dropna(subset=['Company Rating'])

# Group by 'Location' and calculate the average company rating in each location

location\_ratings = df\_cleaned.groupby('Location')['Company Rating'].mean().reset\_index()

# Sort the data by average rating in descending order

location\_ratings = location\_ratings.sort\_values(by='Company Rating', ascending=False)

# Create a bar chart for average company rating by location

plt.figure(figsize=(12, 6))

plt.bar(location\_ratings['Location'], location\_ratings['Company Rating'], color='salmon')

plt.title('Average Company Rating by Location')

plt.xlabel('Location')

plt.ylabel('Average Rating')

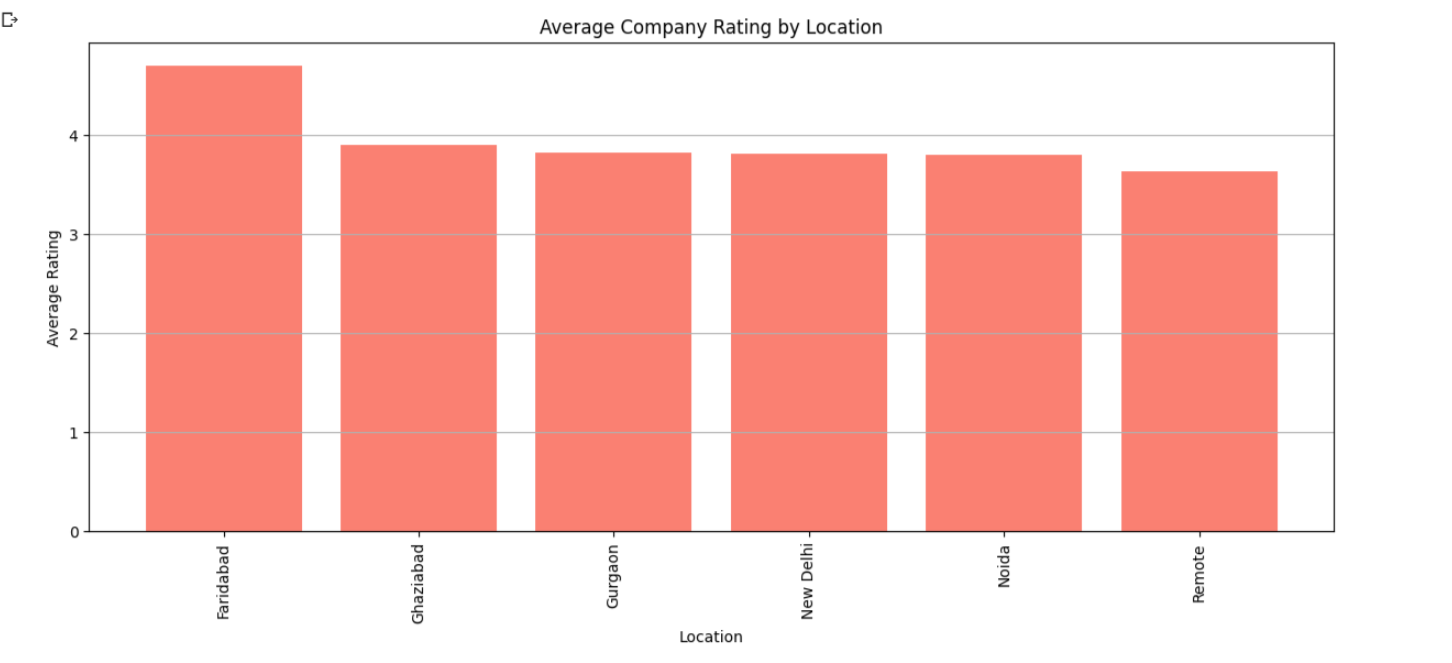
plt.xticks(rotation=90)

plt.grid(axis='y')

# Display the bar chart

plt.tight\_layout()

plt.show()

****

**Managerial Implication of taking these challenges:**

**Understanding the managerial implications of your data analysis objectives is crucial for translating your findings into actionable insights for decision-makers. Here are some managerial implications for each of your objectives:**

**Job Market Insights:**

**Talent Acquisition Strategy: Managers can use insights about the most in-demand skills to refine their talent acquisition strategy. This includes focusing recruitment efforts on candidates with specific skills that are currently in high demand (Like Data Analysis, Excel and Analytical Skills.)**

**Competitive Advantage: Understanding job market trends allows companies to gain a competitive advantage by being ahead of the curve in hiring for emerging roles and skills.**

**Resource Allocation: Allocate resources effectively by prioritizing skill development or recruitment efforts based on market demand.**

**Salary Analysis:**

**Salary Benchmarking: Companies can use salary analysis to benchmark their compensation packages against industry standards to attract and retain top talent.**

**Location-Based Compensation: Managers can adjust salaries based on location if analysis reveals that certain areas offer higher or lower compensation for the same roles.**

**Performance Incentives: Tailor performance incentives and bonuses based on experience levels and salary data to motivate and retain employees.**

**Company Analysis:**

**Employer Branding: Understanding which companies have the most job postings can inform employer branding efforts. Companies with a strong presence in job listings can leverage this to attract top talent. Employee Satisfaction: Analyzing company ratings and reviews can help managers identify areas of improvement in job satisfaction and work culture. Competitive Insights: Gain insights into the strategies of competitor companies in terms of job offerings and employee satisfaction.**

**Location Analysis:**

**Geographic Expansion: Companies can use location analysis to identify regions with high job demand and consider expanding their operations or establishing new offices in those areas. Cost Management: Determine if location-based salary differences are significant and adjust compensation strategies accordingly to manage costs effectively. Remote Work Policies: Consider the potential for remote work options based on location analysis to attract a broader talent pool.**

**A new Talent POOL also wants to work remotely so it signifies how companies need to adapt to remote work!**

**Skill Analysis:**

**Training and Development: Managers can focus on providing training and development opportunities for skills in high demand to ensure that their workforce remains competitive. Recruitment Criteria: Refine job descriptions and recruitment criteria to align with the most sought-after skills. Team Composition: Build cross-functional teams by combining skills that are often mentioned together, enhancing collaboration and problem-solving.**

**Conclusion:**

**In the project "Web Scraping for Analysis of Freshers Jobs in the Field of Data Analytics," we embarked on a comprehensive exploration of the data analytics job market. With a dataset comprising 249 job postings, our analysis uncovered valuable insights across various dimensions, providing actionable recommendations for job seekers and companies alike.**

**Key Findings:**

**1. Job Market Insights:**

**Data analytics job titles are diversifying, reflecting the increasing specialization in the field.**

**Emerging roles suggest a dynamic job market, with opportunities in niche domains.**

**2. Salary Analysis:**

**Compensation for data analytics jobs varies based on factors like experience and location.Salary benchmarks help companies attract and retain top talent.**

**3. Company Analysis:**

**Company ratings and reviews offer insights into job satisfaction and work culture.**

**Leveraging strong job posting presence enhances employer branding.**

**4. Location Analysis:**

**New Delhi leads in job postings, followed by Gurgaon and Noida.**

**Location-based compensation adjustments can help manage costs effectively.**

**Remote work trends are reshaping the talent landscape.**

**5. Skill Analysis:**

**In-demand skills include Data Analysis, Excel, and Analytical Skills.**

**Cross-functional teams benefit from complementary skill sets.**

**Managerial Implications:**

* **Talent acquisition strategies should align with in-demand skills.**
* **Salary benchmarking ensures competitiveness.**
* **Employer branding efforts can leverage strong job posting presence.**

**Remote work policies should adapt to changing workforce preferences.**

**Overall, the project highlights the dynamic nature of the data analytics job market, where specialization and evolving skill requirements are shaping the landscape. Companies and job seekers can leverage these insights to make informed decisions, whether it's about talent acquisition, compensation strategies, or workforce development.**

**In a rapidly evolving field like data analytics, staying ahead of market trends and adapting to changing dynamics is key to success. This project serves as a valuable resource for both job seekers and companies navigating the exciting world of data analytics careers.**