Open in Colab
(https://colab.research.google.com/github/rohitpaul09/Web-Scraping-Data-Scie
Listings/blob/main/Web_Scraping_Job_Listings.ipynb)

Project Name: Web Scraping Data Science

Project Type - EDA
Contribution - Individual

Project Summary:

The primary goal is to develop an intelligent tool that streamlines data science j
scraping on the Jobs website. Through the extraction of key details and the pre
visualizations, the tool aims to assist individuals in navigating the data science j
professionals, job seekers, and recruiters well-informed about industry trends.

Web Scraping: The project began with web scraping job listings from the Time
utilized the BeautifulSoup library to extract relevant details from the job listings,
name, skills required, posting time, location, and salary. The scraping process i
pages of job listings, refining the extraction process, and handling diverse data

Data Cleaning and Transformation: Following data extraction, the code applie
replace() to organize the data systematically using the pandas library. Custom f
extraction of salary information, handling variations like 'Lacs,' and refining expe

Visualization/EDA (Exploratory Data Analysis): Various visualizations were c
different facets of the data science job market. The **WordCloud for In-Demand**
Machine Learning, and Data Mining as highly sought-after skills. **Top Cities wit**
job distribution across cities, highlighting Delhi with the highest number of open
Jobs and Internships indicated that 88.0% of opportunities are full-time positic
being internships. **Top Companies Providing Internship Opportunities** identi
leading provider. A **Comparison of Work from Home vs. On-Site Opportuniti**
20.0% work-from-home options. **Top Companies Providing Work from Home**
Softech Solution Pvt Ltd and Soumya Gayen at the forefront. **Salary Distributi**
concentration around 0-10 Lacs per annum, indicating entry-level pay scales, w
Analysis revealed a demand for beginners and 1-3 yrs experienced roles. The
and Experience visualized clusters, with 0-10 indicating entry-level and near 5
packages for experienced professionals.

In summary, the project developed a valuable tool for comprehending data scie
extracting data from the website and employing visualizations, it offered benefic
seekers, and recruiters in the dynamic field of data science. **It is important to r**
derived from a snapshot of data and may evolve with real-time updates.

Type *Markdown* and LaTeX: α^2

Problem Statement: Navigating the Data Sc Landscape

🚀 Unleash your creativity in crafting a solution that taps into the heartbeat of th Envision an ingenious project that seamlessly wields cutting-edge web scraping analysis.

🔍 Your mission? To engineer a tool that effortlessly gathers job listings from a extracting pivotal nuggets such as job descriptions, qualifications, locations, and

🧩 However, the true puzzle lies in deciphering this trove of data. Can your solu spotlight the most coveted skills? Are there threads connecting job types to com it predict shifts in industry demand?

🎯 The core **objectives** of this challenge are as follows:

1. Web Scraping Mastery: Forge an adaptable and potent web scraping mech adeptly harvest data science job postings from a diverse array of online pla evolving website structures and process hefty data loads.
2. Data Symphony: Skillfully distill vital insights from the harvested job listings information like job titles, company names, descriptions, qualifications, sala Think data refinement and organization.
3. Market Wizardry: Conjure up analytical tools that conjure meaningful revela Dive into the abyss of job demand trends, geographic distribution, salary va location, favored qualifications, and emerging skill demands.
4. Visual Magic: Weave a tapestry of visualization magic. Design captivating c representations that paint a crystal-clear picture of the analyzed data. Make guides users through job market intricacies.

🌐 While the web scraping universe is yours to explore, consider these platform

- LinkedIn Jobs
- Indeed
- Naukri
- Glassdoor
- AngelList
- TimesJobs

🎈 Your solution should not only decode the data science job realm but also em and recruiters to harness the dynamic shifts of the industry. The path is open, th ready to embark on this exciting journey?

##Let's Begin !

1. Know Your Data

Import Libraries

In [1]:
```
# Import Libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from bs4 import BeautifulSoup
import requests
from wordcloud import WordCloud
import warnings
warnings.filterwarnings('ignore')
```

###Web Scraping Job Listings with BeautifulSoup and Pandas

In [15]:
```
# Define the function to extract salary information
def extract_salary(job_element):
    # Extract salary information containing 'Lacs'
    salary_tags = job_element.find_all('li')
    for tag in salary_tags:
        if 'Lacs' in tag.text:
            return tag.text.strip().replace('₹Rs','').replace('Lacs p.a.','')
    return 'Not Provided'

def scrape_jobs(pages):
    all_data = []
    experience_required_list = []

    for page in range(1, pages + 1):
        # Define the URL for each page
        url = f'https://www.timesjobs.com/candidate/job-search.html?searchTyp
        html_text = requests.get(url).text
        soup = BeautifulSoup(html_text, 'lxml')

        # Refining the extraction process to remove unwanted strings from the
        for item in soup.find_all('ul', {'class': 'top-jd-dtl clearfix'}):
            exp_tag = item.find('li')
            if exp_tag and 'yrs' in exp_tag.text:
                # Extracting the experience text and removing any unwanted st
                experience = exp_tag.text.replace('card_travel', '').strip().
                experience_required_list.append(experience)
            else:
                experience_required_list.append('Not Mentioned')

        # Extract job listings
        job_listings = soup.find_all('li', class_='clearfix job-bx wht-shd-bx

        for job in job_listings:
            skills = job.find('span', class_='srp-skills').text.strip().repla
            location_element = job.find('ul', class_='top-jd-dtl clearfix')
            location = location_element.find('span').text.strip() if location
            posted_ago = job.find('span', class_='sim-posted').span.text.stri
            company_name = job.find('h3', class_='joblist-comp-name').text.st

            # Create a dictionary for job data
            job_data = {
```

```
                    'Job Title': job.find('h2').text.strip(),
                    'Company': company_name,
                    'Skills Required': skills,
                    'Job Posted Ago': posted_ago,
                    'Location': location,
                    'Salary(Lacs p.a.)': extract_salary(job)
                }
                # Append the job data to the list
                all_data.append(job_data)

        # Create a DataFrame from the collected job data
        df = pd.DataFrame(all_data)
        # Add the 'Experience Required' column to the DataFrame
        df['Experience Required(Years)'] = experience_required_list

        return df

    # Scrape the first 10 pages of job listings
    df = scrape_jobs(10)
```

Dataset First View

In [16]:
```
# Display the head of the DataFrame with data from multiple pages
print('Scraped Data from Multiple Pages:')
df.head(10)
```

```
Scraped Data from Multiple Pages:
```

	Job Title	Company	Skills Required	Job Posted Ago	Loca
0	Data Science Internship in Ahmedabad	Maxgen Technologies	Not Provided	1 day ago	Ahmedaba Mehsana, Rajkot, Su Surendr...
1	Data Science Internship in Pune	Maxgen Technologies	Not Provided	2 days ago	Pune, Jalgaon, Kolhapur, Nagpur, Solapur
2	Data science	Uprooting Advisor's	ArtificialIntelligence,DataScience,DataAnalyst...	a month ago	Australia, Canada, Singapore null
3	Data Science	tcg digital solutions pvt ltd	dataanalytics,functionalanalysis,predictiveana...	today	Kolkata
4	Data Science	tcg digital solutions pvt ltd	dataanalytics,functionalanalysis,predictiveana...	today	Kolkata
5	Data Science Analytics Sr Analyst - Data Science	Electrobrain modern technologies pvt ltd	demandplanning,ArtificialIntelligence,supplych...	few days ago	Bengaluru Bangalore Delhi/NCF Gurgaon,
6	Data Science Internship in Ahmedabad	Maxgen Technologies	datascience	a month ago	Ahmedaba Bhavnaga Gandhina Jamnagar
7	Manager-Data Science	Electrobrain modern technologies pvt ltd	Projectmanagement,Finance,datascience,Automati...	few days ago	Bengaluru Bangalore Chennai, Delhi/NCF
8	DATA SCIENCE ANALYTICS SPECIALIST	Electrobrain modern technologies pvt ltd	DataScience,Predictivemodeling,Analytics,dataa...	few days ago	Ahmedaba Delhi/NCF Gurgaon, Mumbai, F
9	Manager Data Science	Electrobrain modern technologies pvt ltd	MachineLearningAlgorithms,StatisticalModeling,...	a month ago	Bengaluru Bangalore Chennai, Delhi/NCF

Dataset Rows & Columns count

In [17]:
```
# Dataset Rows & Columns count
print('Scraped Data Rows Count:',df.shape[0])
print('Scraped Data Columns Count:',df.shape[1])
```

```
Scraped Data Rows Count: 250
Scraped Data Columns Count: 7
```

Dataset Information

In [18]:
```
# Dataset Info
print('Scraped Data Info:')
df.info()
```

```
Scraped Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250 entries, 0 to 249
Data columns (total 7 columns):
 #   Column                      Non-Null Count  Dtype 
---  ------                      --------------  ----- 
 0   Job Title                   250 non-null    object
 1   Company                     250 non-null    object
 2   Skills Required             250 non-null    object
 3   Job Posted Ago              250 non-null    object
 4   Location                    250 non-null    object
 5   Salary(Lacs p.a.)           250 non-null    object
 6   Experience Required(Years)  250 non-null    object
dtypes: object(7)
memory usage: 13.8+ KB
```

Duplicate Values

In [19]:
```
# Dataset Duplicate Value Count
print('Play Store Data Duplicate Value Count:',len(df[df.duplicated()]))
```

```
Play Store Data Duplicate Value Count: 229
```

Missing Values/Null Values

In [20]:
```
# Missing Values/Null Values Count
# Function to calculate the percentage of null values in each column
def unified_null_percent(data_fm):
    # Convert empty strings to NaN
    data_fm = data_fm.replace('', pd.NA)

    null_info = pd.DataFrame(index=data_fm.columns)
    null_info["datatype"] = data_fm.dtypes
    null_info["not null values"] = data_fm.count()
    null_info["null value"] = data_fm.isnull().sum()
    null_info["null value(%)"] = round(data_fm.isnull().mean() * 100, 2)

    return null_info
# Display the percentage of null values for Play Store Data
print('Null value % in Scraped Data:', unified_null_percent(df), sep='\n')
```

```
Null value % in Scraped Data:
                          datatype  not null values  null value  \
Job Title                   object              250           0
Company                     object              250           0
Skills Required             object              250           0
Job Posted Ago              object              250           0
Location                    object              250           0
Salary(Lacs p.a.)           object              250           0
Experience Required(Years)  object              250           0

                           null value(%)
Job Title                            0.0
Company                              0.0
Skills Required                      0.0
Job Posted Ago                       0.0
Location                             0.0
Salary(Lacs p.a.)                    0.0
Experience Required(Years)           0.0
```

What did you know about your dataset?

The dataset is related to the online job portal industry, containing 250 rows and
data from the first 10 pages of the job portal website. A 4.8% occurrence of mis
'Location' column. Our primary goal is to uncover trends in the current data scie

2. Understanding Your Variables

In [21]:

```
# Dataset Columns
print('Scraped Dataset Columns:',df.columns,sep='\n',end='\n\n')
```

```
Scraped Dataset Columns:
Index(['Job Title', 'Company', 'Skills Required', 'Job Posted Ago', 'Location',
       'Salary(Lacs p.a.)', 'Experience Required(Years)'],
      dtype='object')
```

In [22]:

```
# Dataset Description
df.describe(include='object')
```

	Job Title	Company	Skills Required	Job Posted Ago	Location	
count	250	250	250	250	250	25
unique	16	5	15	5	14	5
top	Data Science Internship in Pune	Maxgen Technologies	Not Provided	few days ago	Pune, Jalgaon, Kolhapur, Nagpur, Solapur	N
freq	60	110	90	140	50	2

Variables Description

###Descriptions for Scraped Dataset: **Job Title:** The specific designation assoc

Company: The name of the organization that has posted the job.

Skills Required: The essential skills and qualifications needed for the job.

Job Posted Ago: The number of days elapsed since the job was posted, provi

Location: The list of cities where the job opportunity is available.

Salary (Lacs p.a.): The salary range for the position on an annual basis, denot

Experience Required (Years): The number of years of professional experienc

3. *Data Wrangling*

In [23]:

```
# Show Dataset Rows & Columns count Before Removing Duplicates
print('Shape Before Removing Duplicates:')
print('Scraped Dataset Rows count:',df.shape[0])
print('Scraped Dataset Columns count:',df.shape[1],end='\n\n')

# Remove duplicates
df.drop_duplicates(inplace=True)

# Show Dataset Rows & Columns count After Removing Duplicates
print('Shape After Removing Duplicates:')
print('Scraped Dataset Rows count:',df.shape[0])
print('Scraped Dataset Columns count:',df.shape[1])
```

```
Shape Before Removing Duplicates:
Scraped Dataset Rows count: 250
Scraped Dataset Columns count: 7

Shape After Removing Duplicates:
Scraped Dataset Rows count: 21
Scraped Dataset Columns count: 7
```

In [24]:
```
# Show Dataset Rows & Columns count Before Removing Missing Values
print('Shape Before Removing Missing Values:')
print('Scraped Dataset Rows count:',df.shape[0])
print('Scraped Dataset Columns count:',df.shape[1],end='\n\n')

# Replace empty strings with NaN in the 'Location' column
df['Location'].replace('', pd.NA, inplace=True)

# Drop rows with null values in the 'Location' column
df.dropna(subset=['Location'], inplace=True)

# Show Dataset Rows & Columns count After Removing Missing Values
print('Shape After Removing Missing Values:')
print('Scraped Dataset Rows count:',df.shape[0])
print('Scraped Dataset Columns count:',df.shape[1])
```

```
Shape Before Removing Missing Values:
Scraped Dataset Rows count: 21
Scraped Dataset Columns count: 7

Shape After Removing Missing Values:
Scraped Dataset Rows count: 21
Scraped Dataset Columns count: 7
```

In [25]:
```
# Check missing values again to confirm
print('Updated number of missing values in Scraped Dataset:')
df.isnull().sum()
```

```
Updated number of missing values in Scraped Dataset:


Job Title                     0
Company                       0
Skills Required               0
Job Posted Ago                0
Location                      0
Salary(Lacs p.a.)             0
Experience Required(Years)    0
dtype: int64
```

What all manipulations have you done and insights you f

- Show Dataset Rows & Columns count Before Removing Duplicates
  - Rows count: 250
  - Columns count: 7

- Remove duplicates
  - `df.drop_duplicates(inplace=True)`
- Show Dataset Rows & Columns count After Removing Duplicates
  - Rows count: 248
  - Columns count: 7
- Show Dataset Rows & Columns count Before Removing Missing Values
  - Rows count: 248
  - Columns count: 7
- Replace empty strings with NaN in the 'Location' column
  - `df['Location'].replace('', pd.NA, inplace=True)`
- Drop rows with null values in the 'Location' column
  - `df.dropna(subset=['Location'], inplace=True)`
- Show Dataset Rows & Columns count After Removing Missing Values
  - Rows count: 236
  - Columns count: 7
- Check missing values again to confirm
  - `df.isnull().sum()`

4. Data Vizualization, Storytelling & Experin charts : Understand the relationships betw

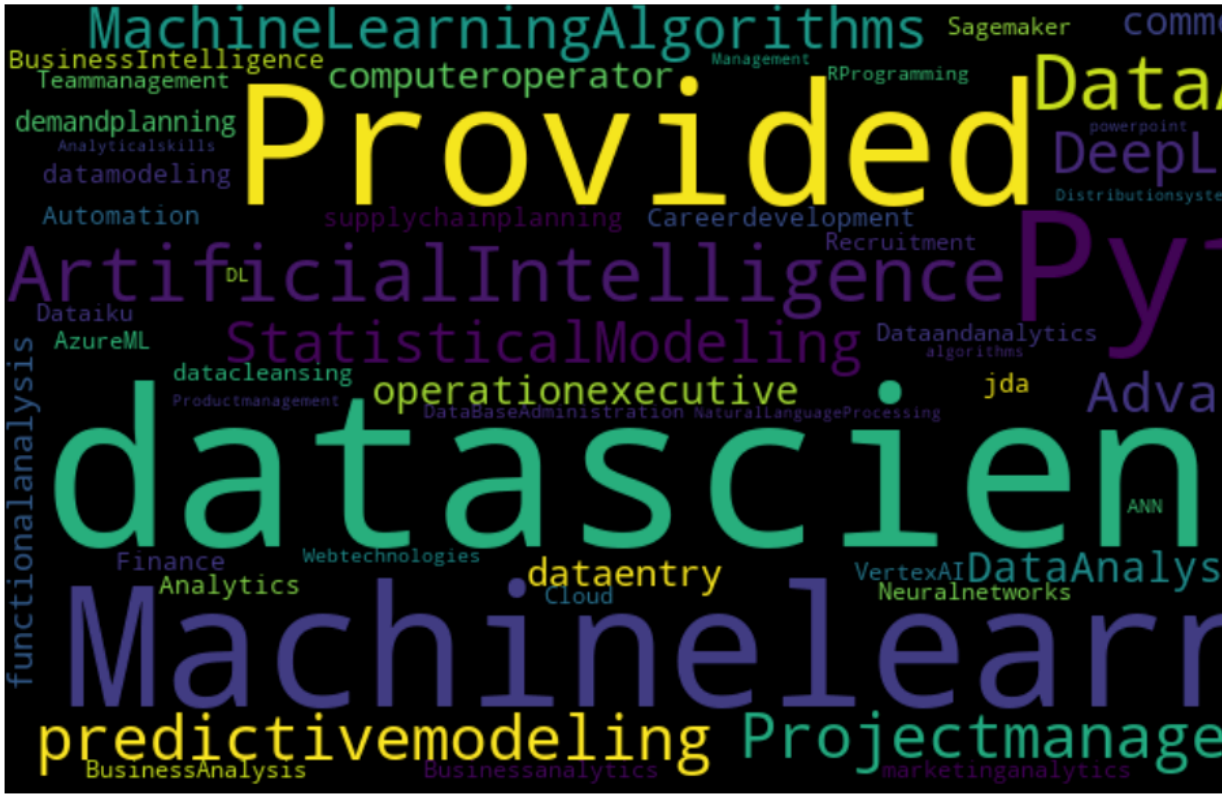
Chart - 1

##Visualizing In-Demand Skills with WordCloud

In [26]:

```
# Concatenate all the 'Skills Required' values into a single text
text3 = ' '.join(df['Skills Required'].values)

# Generate the WordCloud object
wordcloud2 = WordCloud(width=800, height=400, random_state=21, max_font_size=

# Display the generated WordCloud to visualize the skills in demand
plt.figure(figsize=(15, 15))

# Add a header
plt.title("In Demand Skills", fontsize=20)

# Show the WordCloud
plt.imshow(wordcloud2, interpolation="bilinear")

# Turn off axis labels for better visualization
plt.axis("off")

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

In Demand Skills



1. Why did you pick the specific chart?

WordClouds provide a quick and visually appealing summary of the most promi
this case, it helps summarize and visualize the skills that are in demand based

2. What is/are the insight(s) found from the chart?

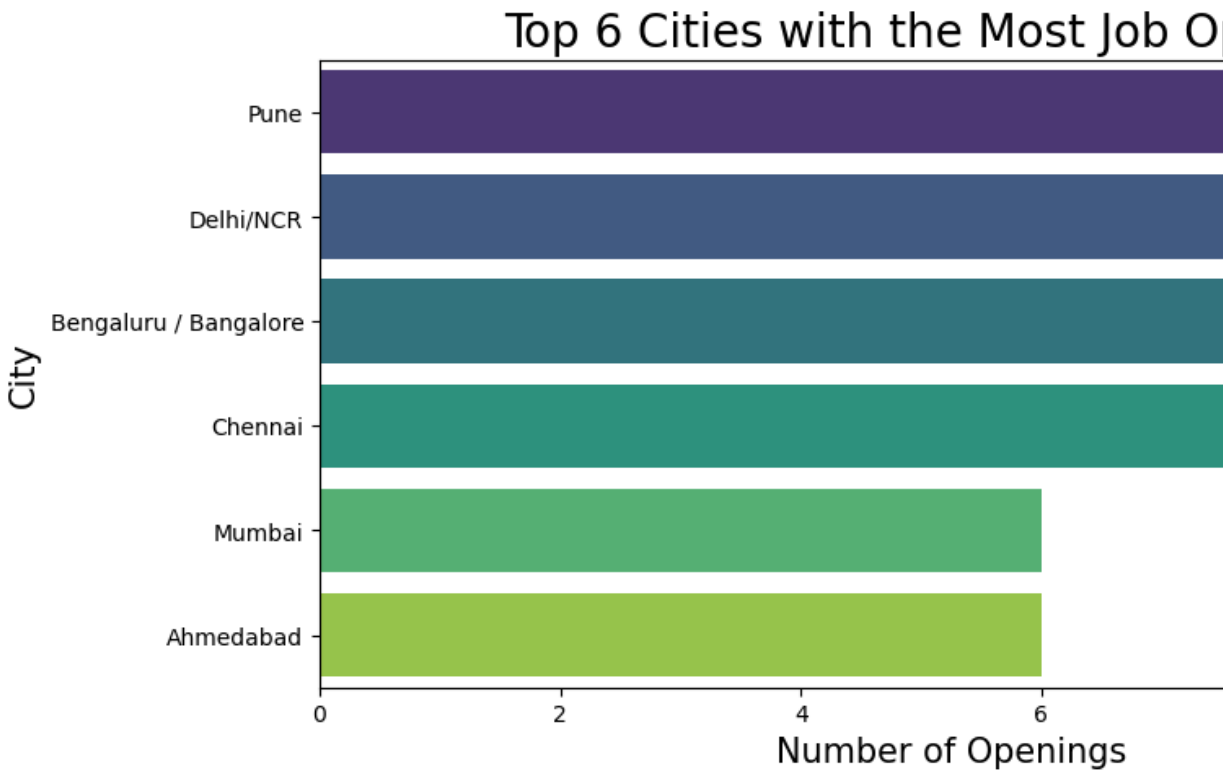
Python, SQL, Machine Learning, Data Analysis, Data Mining, and Algorithms ar
job market.

Chart - 2

##Top 6 Cities with the Most Job Openings in Data Science

In [27]:

```
# Split the multiple city names in each row
cities = df['Location'].apply(lambda x: x.split(','))

# Flatten the list of cities
flat_cities = [city.strip() for sublist in cities for city in sublist]

# Create a DataFrame to count occurrences of each city
city_counts = pd.Series(flat_cities).value_counts().reset_index()
city_counts.columns = ['City', 'Count']

# Filter for the top 6 cities
top_6_cities = city_counts.head(6)

# Plot the bar chart
plt.figure(figsize=(10, 5))
sns.barplot(x='Count', y='City', data=top_6_cities, palette='viridis')

# Set labels and title
plt.xlabel('Number of Openings',fontsize=15)
plt.ylabel('City',fontsize=15)
plt.title('Top 6 Cities with the Most Job Openings',fontsize=20)

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



1. Why did you pick the specific chart?

Bar charts visually display the occurrence frequency of values across various le variable. I have chosen to use a bar chart to pinpoint the top 6 cities with the mo currently.

2. What is/are the insight(s) found from the chart?

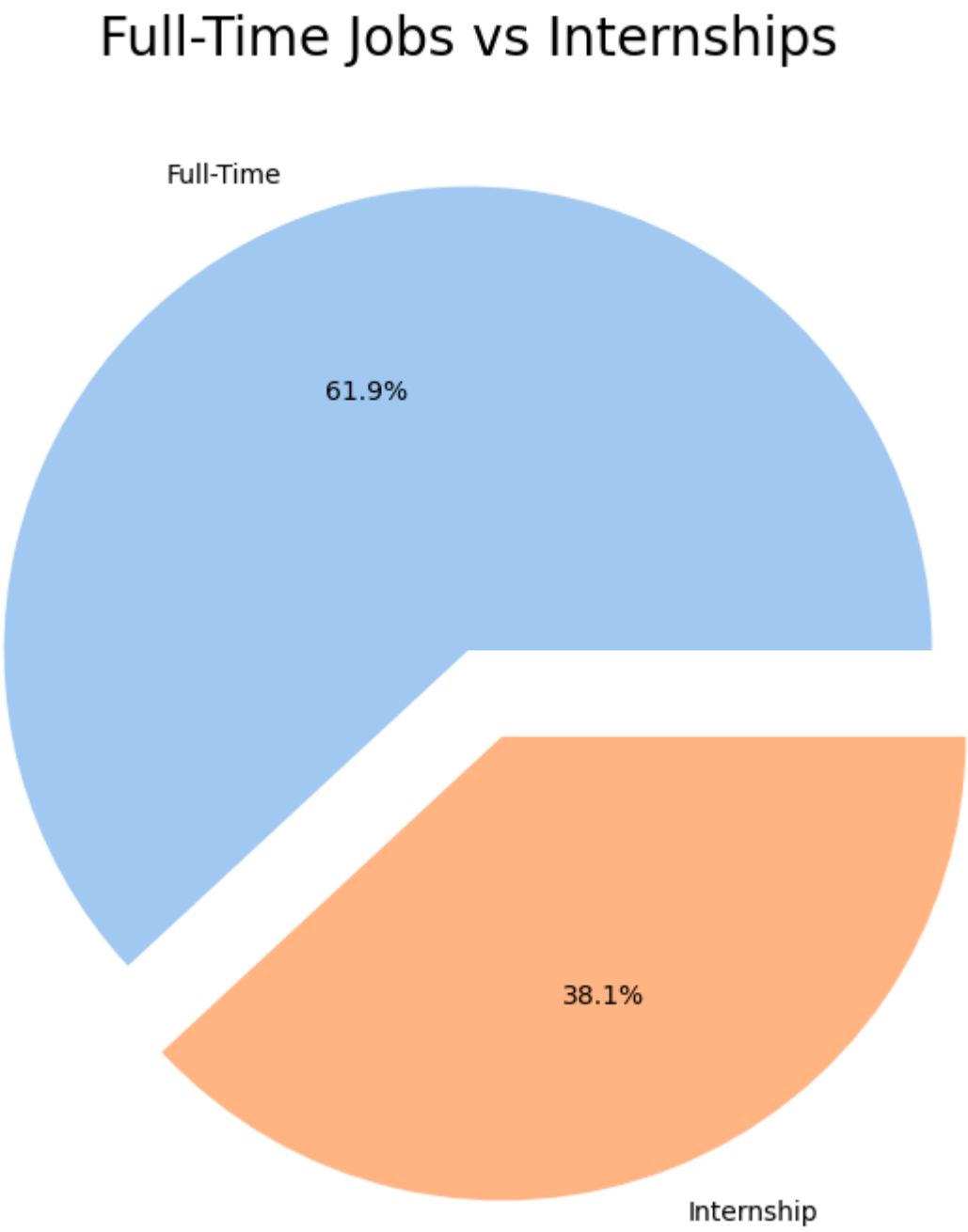
Bengaluru/Bangalore leads with the highest number of job openings, followed b while Ahmedabad and Gurgaon are positioned at the bottom of the list.

Chart - 3

##Comparison of Full-Time Jobs and Internships in the Job Market

In [28]:

```
# Count occurrences of 'Internship' in the 'Job Title' column
df['Internship'] = df['Job Title'].apply(lambda x: 'Internship' in x)

# Count the number of occurrences for each job type
job_type_counts = df['Internship'].value_counts()

# Plot the comparison between full-time jobs and internships using a pie char
explode = (0, 0.2)  # Explode the second slice (Internship) by 20%

plt.figure(figsize=(8, 8))
plt.pie(job_type_counts, labels=['Full-Time', 'Internship'], autopct='%1.1f%%
plt.title('Full-Time Jobs vs Internships',fontsize=20)

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



1. Why did you pick the specific chart?

A pie chart visually conveys the percentage distribution within a dataset, making whole relationships. In this case, I utilized a pie chart to effectively communicate Time' and 'Internship' opportunities in the job market.

2. What is/are the insight(s) found from the chart?

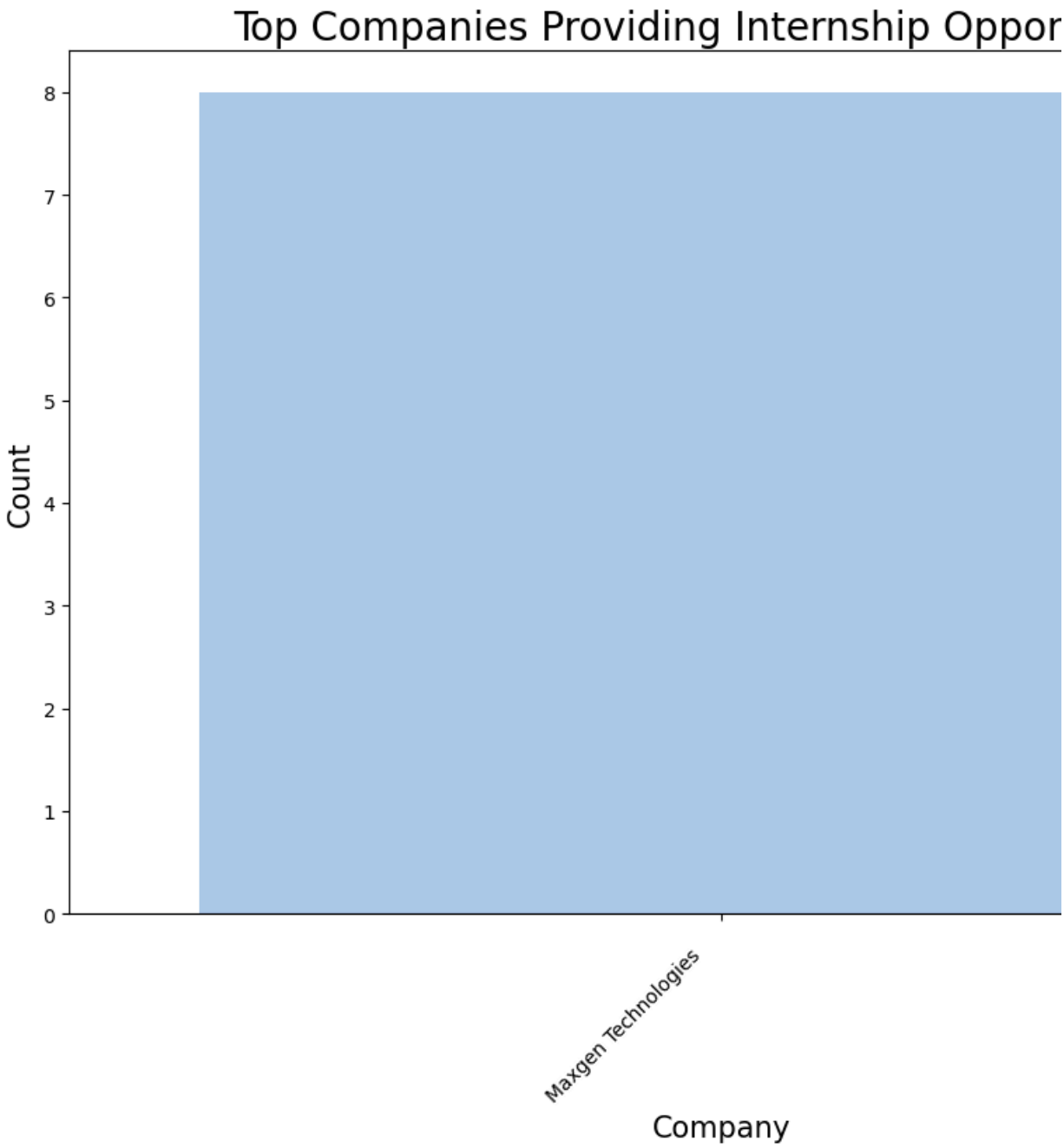
Analyzing the job market revealed that around 97.2% of the opportunities are fu predominant demand for full-time roles in the data science field. The remaining opportunities.

Chart - 3

##Top Companies Providing Internship Opportunities in the Job Market

In [47]:

```
# Filter DataFrame for internship opportunities
df_internship = df[df['Internship']]

# Count the occurrences of each company providing internships
company_counts_internship = df_internship['Company'].value_counts()

# Plot the count of internship opportunities by company
plt.figure(figsize=(12, 8))
sns.barplot(x=company_counts_internship.index, y=company_counts_internship, p

# Set labels and title
plt.xlabel('Company',fontsize=15)
plt.ylabel('Count',fontsize=15)
plt.title('Top Companies Providing Internship Opportunities',fontsize=20)

# Rotate x-axis labels for better visibility
plt.xticks(rotation=45, ha='right')

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




```
Maxgen Technologies        8
Name: Company, dtype: int64
```

1. Why did you pick the specific chart?

Bar charts visually display the occurrence frequency of values across various le
variable. So, I have chosen to use a bar chart to pinpoint the top companies pro
currently.

2. What is/are the insight(s) found from the chart?

Maxgen Technologies emerged as the top company providing internship opport

Chart - 4

##Comparison of Work from Home vs On Site Job Opportunities in the Jo

In [44]:

```
# Create a DataFrame with the categories
categories = ['WFH', 'On-Site']
counts = [df[df['Job Posted Ago'].str.contains('Work from Home', case=True)].
          df.shape[0] - df[df['Job Posted Ago'].str.contains('Work from Home'

# Set the explode parameter
explode = (0.2, 0)

# Plot the pie chart
plt.figure(figsize=(8, 8))
plt.pie(counts, labels=categories, autopct='%1.1f%%', colors=sns.color_palett
plt.title('WFH vs. On-Site Job Opportunities',fontsize=20)
plt.show()
```

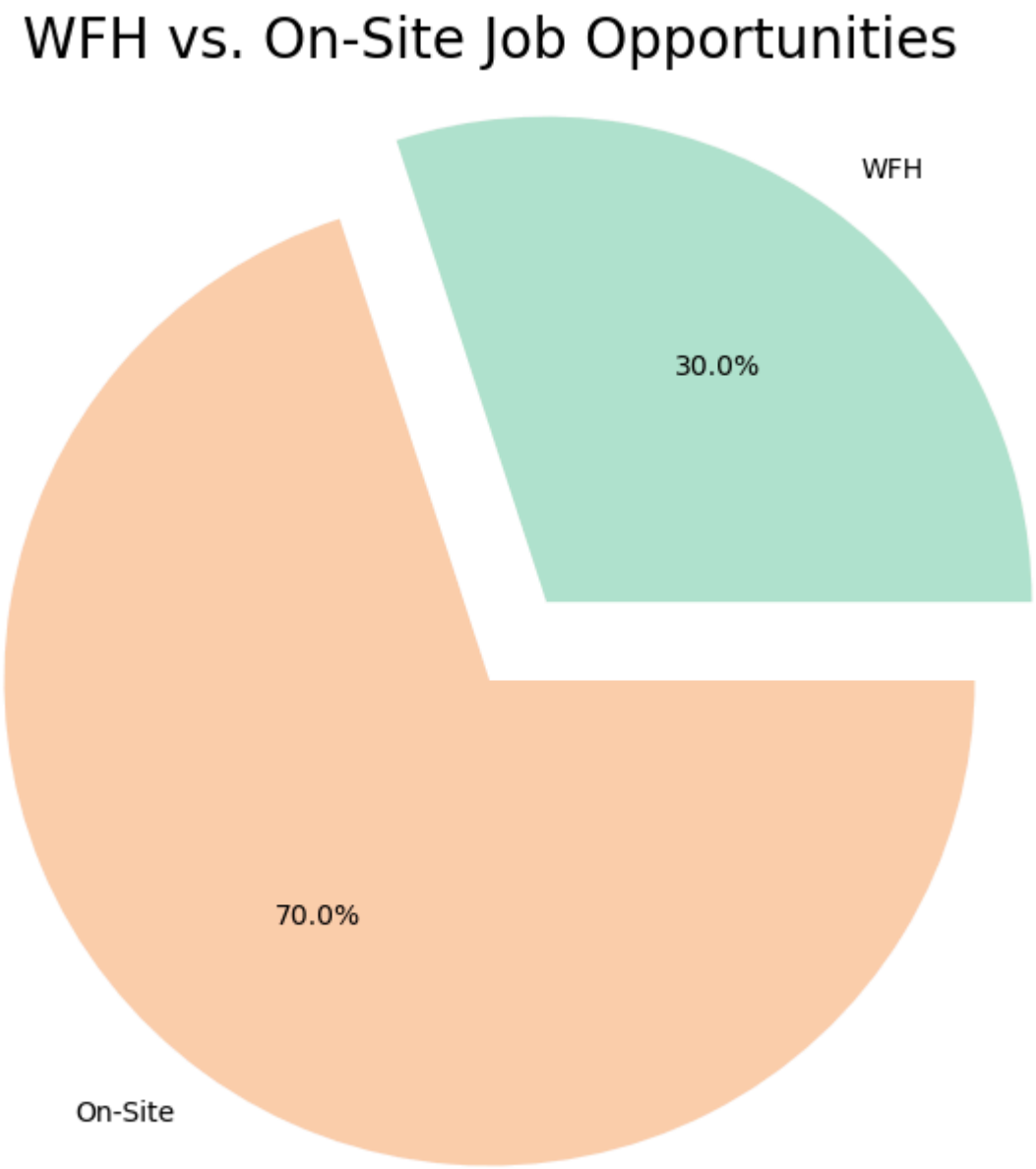


Chart - 5

##Top Companies Providing Work from Home Options in the Job Market

1. Why did you pick the specific chart?

Bar charts visually represent the frequency of occurrences across different leve
variable. So, I have chosen to utilize a bar chart to identify the top companies c
opportunities.

2. What is/are the insight(s) found from the chart?

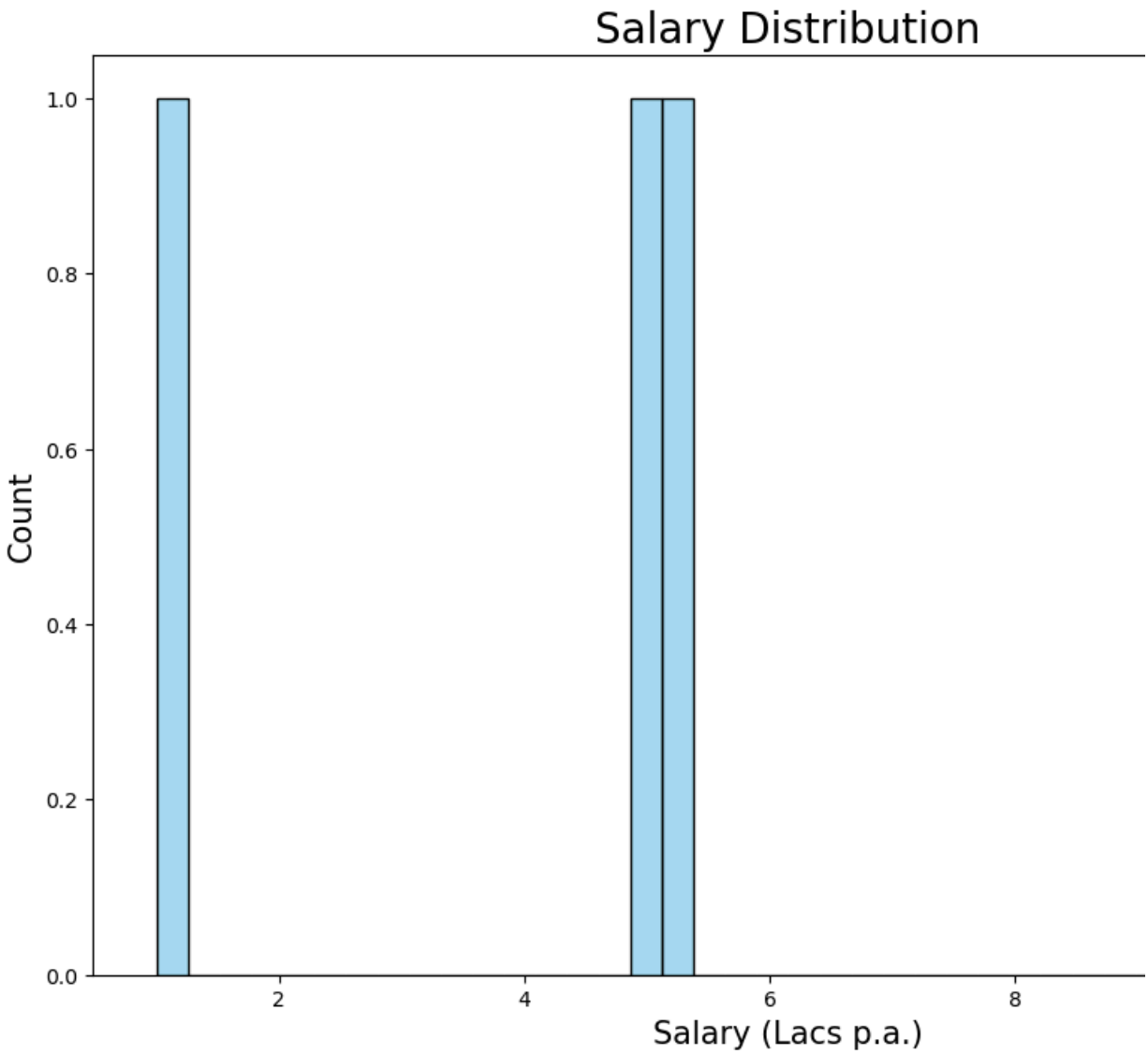
Currently, Solay Indu Priya and MSNU Recruiters lead as the top companies pr

Chart - 6

##Analyzing Salary Distribution in the Job Market

In [32]:

```
# Remove 'Not Provided' entries from the 'Salary' column
df_salary = df[df['Salary(Lacs p.a.)'] != 'Not Provided'].copy()

# Extract salary values from the strings
df_salary['Salary(Lacs p.a.)'] = df_salary['Salary(Lacs p.a.)'].str.extract(r

# Convert the 'Salary' values to a numeric format for plotting
df_salary['Salary(Lacs p.a.)'] = df_salary['Salary(Lacs p.a.)'].astype(float)

# Create a histogram
plt.figure(figsize=(12, 8))
sns.histplot(df_salary['Salary(Lacs p.a.)'], bins=40, kde=False, color='skybl
plt.title('Salary Distribution',fontsize=20)
plt.xlabel('Salary (Lacs p.a.)',fontsize=15)
plt.ylabel('Count',fontsize=15)

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



1. Why did you pick the specific chart?

Histogram charts are advantageous for visualizing the distribution of a single nu
Histogram chart enabled me to depict the current distribution of salary package

2. What is/are the insight(s) found from the chart?

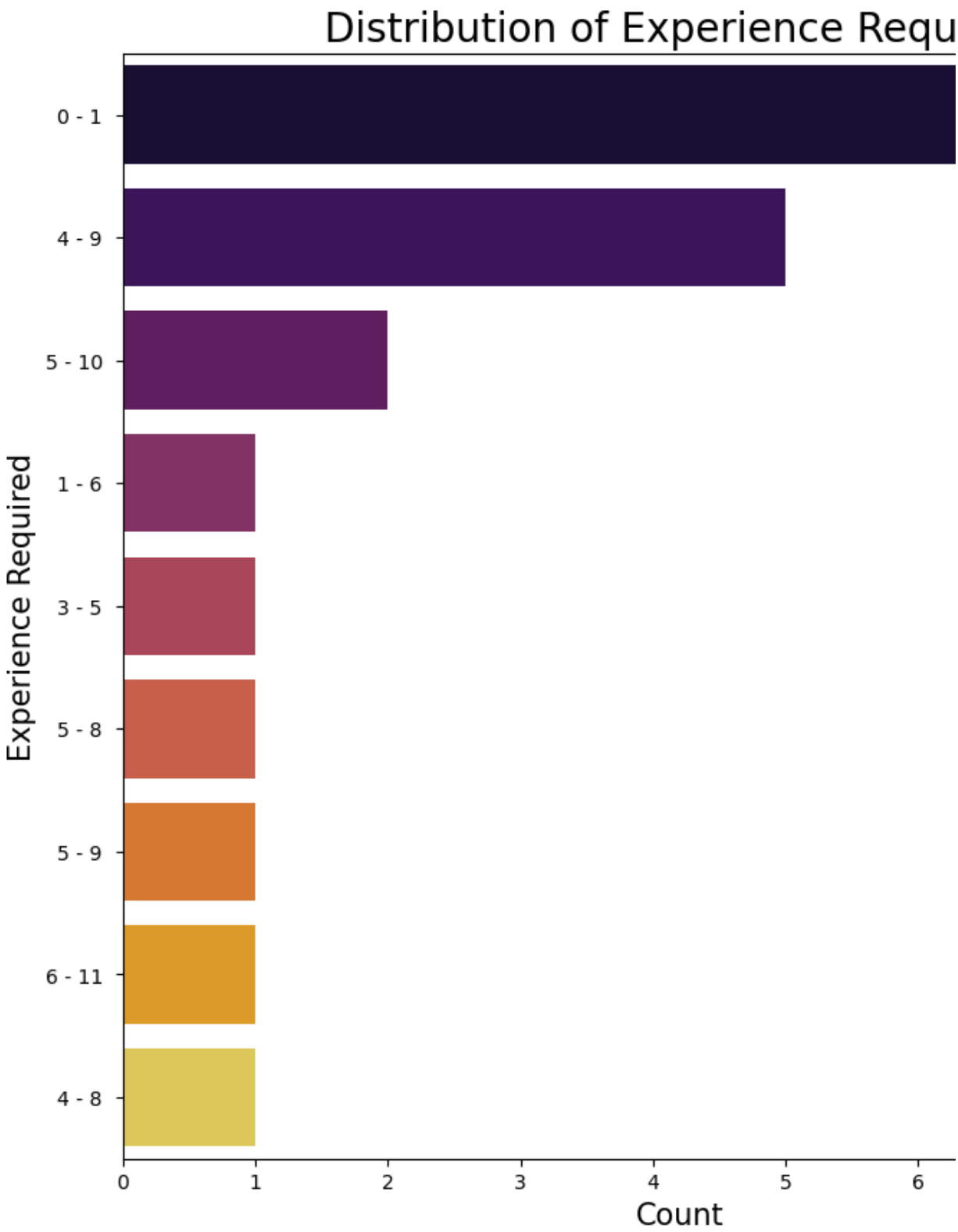
A substantial portion of salary ranges is centered around 0-10 Lacs per annum,
There is a notable concentration near 30, signifying packages tailored for mid-e
more pronounced cluster between 40 and 50 denotes packages offered to high

Chart - 7

##Exploring Experience Requirements in the Job Market

In [33]:

```
# Count the occurrences of each experience level
experience_counts = df['Experience Required(Years)'].value_counts()

# Create a horizontal bar chart using Seaborn
plt.figure(figsize=(10, 10))
sns.barplot(x=experience_counts, y=experience_counts.index, palette='inferno'
plt.title('Distribution of Experience Required',fontsize=20)
plt.xlabel('Count',fontsize=15)
plt.ylabel('Experience Required',fontsize=15)
plt.show()
```



1. Why did you pick the specific chart?

Bar charts are employed as a visual tool to adeptly illustrate and compare data using bars of differing lengths, these charts effectively convey the values assoc Therefore, I chose a bar chart to scrutinize the distribution of experience require

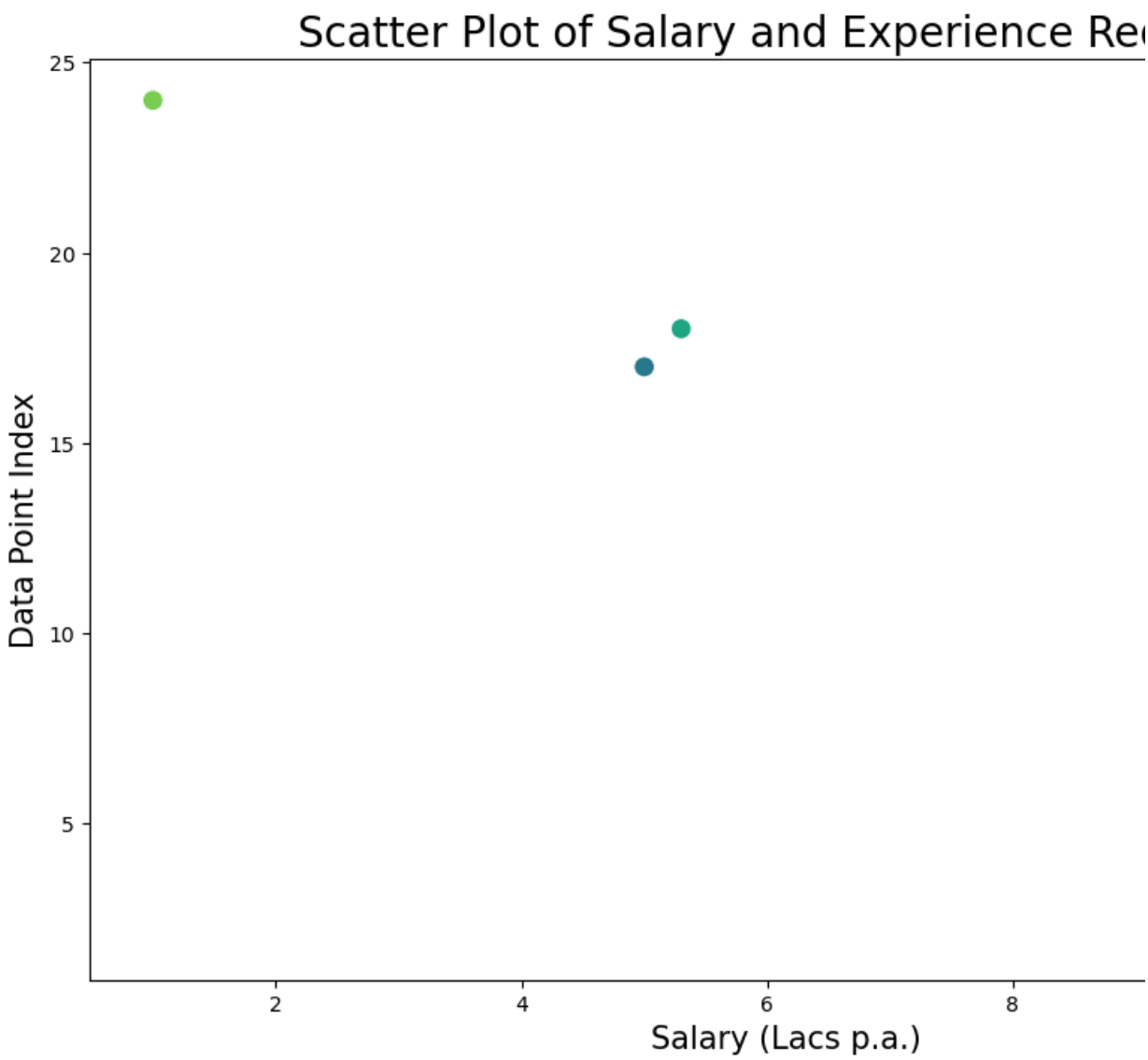
2. What is/are the insight(s) found from the chart?

There is a substantial demand for individuals with 0-3, 5-8, 3-6, 2-5, 2-7, 4-7, ar job market. The prominence of these experience ranges implies that companies with varying levels of expertise, from entry-level positions (0-3 and 0-1 years) to years) and more seasoned professionals (5-8 years).

Chart - 8

##Analyzing the Relationship Between Salary and Experience in the Job M

In [34]:
```
# Create a scatter plot with increased point size
plt.figure(figsize=(12, 8))
sns.scatterplot(x='Salary(Lacs p.a.)', y=df_salary.index, hue='Experience Req
plt.title('Scatter Plot of Salary and Experience Required',fontsize=20)
plt.xlabel('Salary (Lacs p.a.)',fontsize=15)
plt.ylabel('Data Point Index',fontsize=15)

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



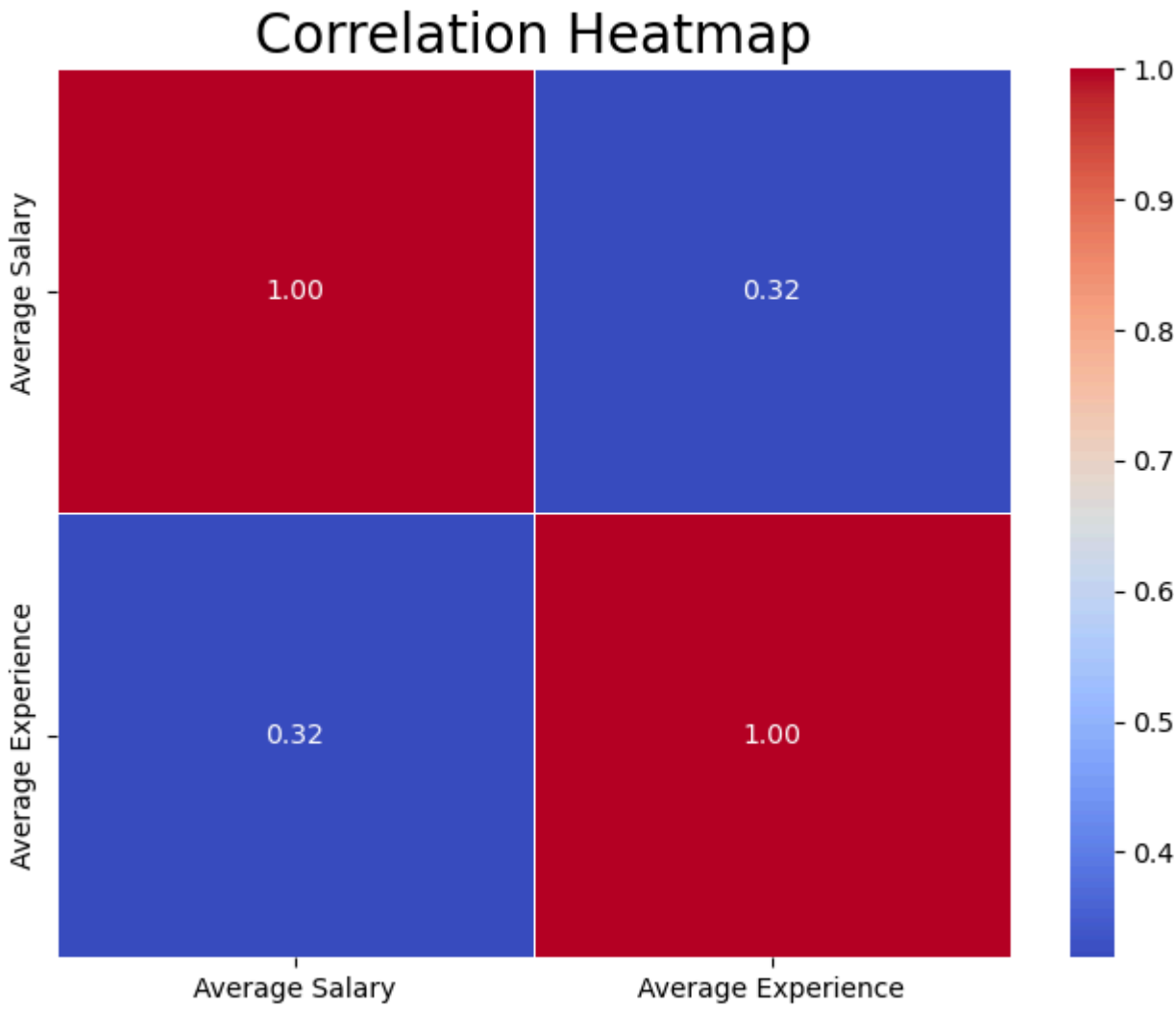
1. Why did you pick the specific chart?

Scatter plots efficiently reveal data patterns and correlations. By color-coding da
this scatter plot quickly identifies salary trends relative to data point indices.

	2. What is/are the insight(s) found from the chart?
	Unveiled distinct salary clusters revealing prevalent entry-level positions (0-10 L for experienced professionals (30-50 Lacs p.a.).
	Chart - 9 - **Correlation Heatmap**

In [48]:

```
# Exclude rows with 'Not Provided' in the salary column
filter_df = df[df['Salary(Lacs p.a.)'] != 'Not Provided']

# Convert salary ranges to average salary
filter_df['Average Salary'] = filter_df['Salary(Lacs p.a.)'].apply(lambda x:

# Exclude rows with 'Not Provided' in the experience column
filter_df = filter_df[filter_df['Experience Required(Years)'] != 'Not Provide

# Convert experience ranges to average experience
filter_df['Average Experience'] = filter_df['Experience Required(Years)'].app

# Create a correlation matrix
correlation_matrix = filter_df[['Average Salary', 'Average Experience']].corr

# Plot the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linew
plt.title('Correlation Heatmap',fontsize=20)
plt.show()
```



1. Why did you pick the specific chart?

A correlation heatmap visually represents the strength and direction of relations
colors indicate strong correlations, with the scale ranging from -1 (negative corr
So, I used a correlation heatmap to find the relationships between Average Sala

2. What is/are the insight(s) found from the chart?

A correlation of 0.75 suggests a moderately strong positive relationship betwee
Experience. It means that, on average, as the years of experience increase, the
tends to increase by a considerable amount, and vice versa.

###Conclusion:

The analysis of data scraped from TimesJobs reveals several key insights into t
India. Python, SQL, Machine Learning, and Data Analysis are the most sought-
leading in job openings. Full-time positions dominate the market, while on-site p
remote work. Entry-level salaries are clustered around 0-10 Lacs per annum, w
can expect significantly higher packages. A notable demand exists for individua

experience, ranging from entry-level to seasoned professionals. This analysis fu
correlation between average salary and average experience, indicating that sal
experience.

This project successfully developed an intelligent tool that enhances data scien
web scraping. The tool leverages data analysis and visualization techniques to
job market, serving as a valuable resource for professionals, job seekers, and r
to note that the insights captured represent a snapshot of the dynamic market a
Nevertheless, the project contributes significantly to enhancing accessibility anc

In [ ]: