In [ ]:

Glassdoor Job Market Analysis -

Project Type - EDA

Contribution - Individual

Name - Sandeep Prajapat

Project Summary -

This project analyzes Glassdoor job postings to uncover trends in salaries, business size impact, industry distribution, and key factors influencing job compensation. The study explores salary variations across job titles, company ratings, revenue levels, and geographic locations. Additionally, it examines skill demand and sector-wise job availability. Data visualizations, including bar charts, scatter plots, and heatmaps, are used to present insights. The findings help job seekers and employers understand salary expectations and industry trends.

GitHub Link -

Github link

Problem Statement

In today's competitive job market, understanding salary trends and job opportunities is crucial for both job seekers and employers. Salaries vary significantly based on factors such as job title, company size, industry, location, and required skills. However, without proper data analysis, identifying these trends remains challenging.

- This project aims to analyze Glassdoor job postings to:
- Identify salary patterns across different job roles, industries, and locations.
- Examine the impact of company size and revenue on salary levels.
- Explore the relationship between company ratings and compensation.
- Determine the most in-demand skills in the job market.

Define Your Business Objective?

The objective of this project is to provide data-driven insights into job market trends, helping job seekers, employers, and industry analysts make informed decisions. By analyzing Glassdoor job postings, the study aims to:

For Job Seekers

- Understand salary expectations based on job title, company size, and location.
- Identify the most in-demand skills to enhance career prospects.
- Compare job opportunities across industries and geographic regions.

For Employers & Recruiters

- Benchmark salary offerings to stay competitive in the market.
- Identify hiring trends and industry demand for specific roles.
- Assess how company ratings and size impact talent acquisition.

For Industry Analysts & Policymakers

- Evaluate employment trends across different business sectors.
- Analyze workforce distribution and salary growth patterns.
- Provide insights into skill demand for workforce development strategies.

General Guidelines : -

1. Well-structured, formatted, and commented code is required.

2. Exception Handling, Production Grade Code & Deployment Ready Code will be a plus. Those students will be awarded some additional credits.

   The additional credits will have advantages over other students during Star Student selection.

   ```
   [ Note: - Deployment Ready Code is defined as, the whole .ipynb notebook should be executable in one go
             without a single error logged. ]
   ```

3. Each and every logic should have proper comments.

4. You may add as many number of charts you want. Make Sure for each and every chart the following format should be answered.

```
# Chart visualization code
```

- Why did you pick the specific chart?
- What is/are the insight(s) found from the chart?
- Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

5. You have to create at least 20 logical & meaningful charts having important insights.

[Hints : - Do the Vizualization in a structured way while following "UBM" Rule.

U - Univariate Analysis,

B - Bivariate Analysis (Numerical - Categorical, Numerical - Numerical, Categorical - Categorical)

M - Multivariate Analysis]

Let's Begin !

1. Know Your Data

Import Libraries

In [1]:
```
# Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

Dataset Loading

In [2]:
```
# Load Dataset
glassdoor_jobs = pd.read_csv(r"C:\Users\sande\Downloads\glassdoor_jobs.csv")
```

Dataset First View

In [3]:
```
# Dataset First Look
glassdoor_jobs.head()
```

Out[3]:

	Unnamed: 0	Job Title	Salary Estimate	Job Description	Rating	Company Name	Location	Headquarters	Size	Founded	Type of ownership
0	0	Data Scientist	$53K-91K (Glassdoor est.)	Data Scientist\nLocation: Albuquerque, NM\nEdu...	3.8	Tecolote Research\n3.8	Albuquerque, NM	Goleta, CA	501 to 1000 employees	1973	Company - Private
1	1	Healthcare Data Scientist	$63K-112K (Glassdoor est.)	What You Will Do:\n\nI. General Summary\n\nThe...	3.4	University of Maryland Medical System\n3.4	Linthicum, MD	Baltimore, MD	10000+ employees	1984	Other Organization
2	2	Data Scientist	$80K-90K (Glassdoor est.)	KnowBe4, Inc. is a high growth information sec...	4.8	KnowBe4\n4.8	Clearwater, FL	Clearwater, FL	501 to 1000 employees	2010	Company - Private
3	3	Data Scientist	$56K-97K (Glassdoor est.)	*Organization and Job ID**\nJob ID: 310709\n\n...	3.8	PNNL\n3.8	Richland, WA	Richland, WA	1001 to 5000 employees	1965	Government
4	4	Data Scientist	$86K-143K (Glassdoor est.)	Data Scientist\nAffinity Solutions / Marketing...	2.9	Affinity Solutions\n2.9	New York, NY	New York, NY	51 to 200 employees	1998	Company - Private

2. Data Wrangling & Data Cleaning

In [4]:
```
glassdoor_jobs.columns
```

Out[4]:
```
Index(['Unnamed: 0', 'Job Title', 'Salary Estimate', 'Job Description',
       'Rating', 'Company Name', 'Location', 'Headquarters', 'Size', 'Founded',
       'Type of ownership', 'Industry', 'Sector', 'Revenue', 'Competitors'],
      dtype='object')
```

Data Wrangling Code

In [ ]:
```
#  1. Renaming Columns
glassdoor_jobs.columns = glassdoor_jobs.columns.str.replace(' ', '_')

#  2. To convert hourly wages to annual salary, hourly salary should be multiply by 2080 (number of work hours in a year: 40 hour

glassdoor_jobs['Salary_Estimate'] = glassdoor_jobs['Salary_Estimate'].str.replace('$25-$28','$52K-$58K')
glassdoor_jobs['Salary_Estimate'] = glassdoor_jobs['Salary_Estimate'].str.replace('$10-$17','$21K-$35K')
glassdoor_jobs['Salary_Estimate'] = glassdoor_jobs['Salary_Estimate'].str.replace('$21-$29','$44K-$60K')
glassdoor_jobs['Salary_Estimate'] = glassdoor_jobs['Salary_Estimate'].str.replace('$21-$34','$44K-$71K')
glassdoor_jobs['Salary_Estimate'] = glassdoor_jobs['Salary_Estimate'].str.replace('$18-$25','$37K-$52K')
glassdoor_jobs['Salary_Estimate'] = glassdoor_jobs['Salary_Estimate'].str.replace('$24-$39','$50K-$81K')
glassdoor_jobs['Salary_Estimate'] = glassdoor_jobs['Salary_Estimate'].str.replace('$17-$24','$35K-$50K')
glassdoor_jobs['Salary_Estimate'] = glassdoor_jobs['Salary_Estimate'].str.replace('$27-$47','$56K-$98K')
glassdoor_jobs['Salary_Estimate'] = glassdoor_jobs['Salary_Estimate'].str.replace('$15-$25','$31K-$52K')

#  3. Remove  "$", "K", "(Glassdoor est.)", "Employer Provided Salary:", "(Employer est.)", "Per Hour" and extra spaces
glassdoor_jobs.Salary_Estimate = glassdoor_jobs.Salary_Estimate.str.replace('(Glassdoor est.)', ' ').str.replace('Employer Provic

#  4. Create Min_Salary and Max_Salary columns
glassdoor_jobs[['Min_Salary', 'Max_Salary']] = glassdoor_jobs['Salary_Estimate'].str.split('-', expand=True)

glassdoor_jobs['Min_Salary'] = pd.to_numeric(glassdoor_jobs['Min_Salary'], errors='coerce')
glassdoor_jobs['Max_Salary'] = pd.to_numeric(glassdoor_jobs['Max_Salary'], errors='coerce')

#  5. Create Avg_Salary column
glassdoor_jobs['Avg_Salary'] = (glassdoor_jobs.Min_Salary + glassdoor_jobs.Max_Salary)/2

# 6. Extract the state abbreviation (last part after the comma)
glassdoor_jobs['Job_State'] = glassdoor_jobs['Location'].str.split(', ').str[-1]

#  7. Converting state name in state abbrevation that are not in abbreaviation form
glassdoor_jobs['Job_State'] = glassdoor_jobs.Job_State.str.replace('Oregon', 'OR').str.replace('New Jersey', 'NJ').str.replace('V

#  8. Converting negative Rating  to 0
glassdoor_jobs['Rating'] = np.where(glassdoor_jobs.Rating < 0, 0, glassdoor_jobs.Rating)

#  9. Categorizing all job titles
def categorize_job_title(title):
    title_lower = title.lower()

    if any(keyword in title_lower for keyword in ['data analyst', 'data & analytics','analytics manager', 'bi & platform analytic
        return 'Data Analyst'
    elif any(keyword in title_lower for keyword in ['data scientist', 'machine learning scientist', 'research scientist', 'ai sci
        return 'Data Scientist'
    elif any(keyword in title_lower for keyword in ['data engineer', 'data architect', 'data systems specialist','data management
        return 'Data Engineer'
    elif any(keyword in title_lower for keyword in ['machine learning', 'deep learning', 'ml engineer']):
        return 'Machine Learning Engineer'
    else:
        return 'Other'

glassdoor_jobs['new_job_title'] = glassdoor_jobs['Job_Title'].apply(categorize_job_title)

#  10. Converting -1 Founded year to 2025
glassdoor_jobs['Founded'] = np.where(glassdoor_jobs.Founded < 0, 2025, glassdoor_jobs.Founded)

#  11. Create Year_of_Operation column
glassdoor_jobs['Year_of_Operation'] = 2025 - glassdoor_jobs.Founded

#  12. Converting -1 Industry to 'Other Industrie'
glassdoor_jobs.Industry = glassdoor_jobs.Industry.str.replace('-1', 'Other Industries')

#  13. Converting -1 Headquarters to 'Other'
glassdoor_jobs.Headquarters = glassdoor_jobs.Headquarters.str.replace('-1', 'Other')

#  14. Converting -1 Type_of_ownership to 'Other Organization'
glassdoor_jobs.Type_of_ownership = glassdoor_jobs.Type_of_ownership.str.replace('-1','Other Organization' )

#  15. Converting -1 Sector to 'Other Sector'
glassdoor_jobs.Sector = glassdoor_jobs.Sector.str.replace('-1','Other Sector' )

#  16. Create Business_Size column as per employees count
size_mapping = {
    '1 to 50 employees': 'Small Business',
    '51 to 200 employees': 'Medium Business',
    '201 to 500 employees': 'Large Business',
    '501 to 1000 employees': 'Large Business',
    '1001 to 5000 employees': 'Enterprise Business',
    '5001 to 10000 employees': 'Enterprise Business',
    '10000+ employees': 'Enterprise Business',
    '-1': 'Unknown'
}
```

```
glassdoor_jobs['Business_Size'] = glassdoor_jobs['Size'].replace(size_mapping)

#  17. Remove Unknown / Non-Applicable','Unknown and replace -1 to unknown also Ensure all values are strings
glassdoor_jobs['Revenue'] = glassdoor_jobs['Revenue'].str.replace('-1','Unknown').str.replace('Unknown / Non-Applicable','Unknowr

#  18. Creating new column Python as per skill Python in Job description
def check_python(skill):
    if pd.isna(skill):
        return 0
    return 1 if 'python' in skill.lower() else 0

glassdoor_jobs['Python'] = glassdoor_jobs['Job_Description'].apply(check_python)

glassdoor_jobs['Python'].value_counts()

#  19. Creating new column Excel as per skill Excel in Job description
def check_excel(skill):
    if pd.isna(skill):
        return 0
    return 1 if 'excel' in skill.lower() else 0

glassdoor_jobs['Excel'] = glassdoor_jobs['Job_Description'].apply(check_excel)

glassdoor_jobs['Excel'].value_counts()

#  20. Creating new column Tableau as per skill Tableau in Job description
def check_tableau(skill):
    if pd.isna(skill):
        return 0
    return 1 if 'tableau' in skill.lower() else 0

glassdoor_jobs['Tableau'] = glassdoor_jobs['Job_Description'].apply(check_tableau)

glassdoor_jobs['Tableau'].value_counts()

#  21. Creating new column power_BI as per skill Power BI in Job description
def check_power_bi(skill):
    if pd.isna(skill):
        return 0
    return 1 if 'power bi' in skill.lower() else 0

glassdoor_jobs['Power_BI'] = glassdoor_jobs['Job_Description'].apply(check_power_bi)

glassdoor_jobs['Power_BI'].value_counts()

#  22. Creating new column SQL as per skill SQL in Job description
def check_SQL(skill):
    if pd.isna(skill):
        return 0
    return 1 if 'sql' in skill.lower() else 0

glassdoor_jobs['SQL'] = glassdoor_jobs['Job_Description'].apply(check_SQL)

glassdoor_jobs['SQL'].value_counts()


#  23. Creating new column AWS as per skill AWS in Job description
def check_AWS(skill):
    if pd.isna(skill):
        return 0
    return 1 if 'aws' in skill.lower() else 0

glassdoor_jobs['AWS'] = glassdoor_jobs['Job_Description'].apply(check_AWS)

glassdoor_jobs['AWS'].value_counts()


# 24. Creating new column Hadoop as per skill Hadoop in Job description
def check_Hadoop(skill):
    if pd.isna(skill):
        return 0
    return 1 if 'hadoop' in skill.lower() else 0

glassdoor_jobs['Hadoop'] = glassdoor_jobs['Job_Description'].apply(check_Hadoop)

glassdoor_jobs['Hadoop'].value_counts()

# 25. Creating new column Azure as per skill Azure in Job description
def check_Azure(skill):
    if pd.isna(skill):
        return 0
    return 1 if 'azure' in skill.lower() else 0

glassdoor_jobs['Azure'] = glassdoor_jobs['Job_Description'].apply(check_Azure)

glassdoor_jobs['Azure'].value_counts()

# 26.  Creating new column Spark as per skill Spark in Job description
def check_Spark(skill):
```

```
    if pd.isna(skill):
        return 0
    return 1 if 'spark' in skill.lower() else 0

glassdoor_jobs['Spark'] = glassdoor_jobs['Job_Description'].apply(check_Spark)

glassdoor_jobs['Spark'].value_counts()

# 27. Creating new column Machine_Learning as per skill Machine_Learning in Job description
def check_Machine_Learning(skill):
    if pd.isna(skill):
        return 0
    return 1 if 'machine learning' in skill.lower() else 0

glassdoor_jobs['Machine_Learning'] = glassdoor_jobs['Job_Description'].apply(check_Machine_Learning)

glassdoor_jobs['Machine_Learning'].value_counts()

# 28. Creating new column Matlab as per skill Matlab in Job description
def check_Matlab(skill):
    if pd.isna(skill):
        return 0
    return 1 if 'matlab' in skill.lower() else 0

glassdoor_jobs['Matlab'] = glassdoor_jobs['Job_Description'].apply(check_Matlab)

glassdoor_jobs['Matlab'].value_counts()

# 29. Dropping unnecessary columns
glassdoor_jobs.drop(['Unnamed:_0','Salary_Estimate','Job_Title', 'Job_Description', 'Location','Size','Competitors'] ,axis = 1, i
```

Data Cleaning Summary

1. Column Formatting

- Renamed column names by replacing spaces with underscores for consistency.

2. Salary Processing

- Converted hourly wages to annual salaries.
- Removed unnecessary characters like "$", "K", "Per Hour", "(Glassdoor est.),etc.".
- Extracted Min_Salary, Max_Salary, and computed Avg_Salary.

3. Location Standardization

- Extracted state abbreviations from job locations.
- Converted full state names to abbreviations where needed.

4. Handling Missing or Inconsistent Values

- Set negative ratings (-1) to 0.
- Replaced -1 values in Founded, Industry, Headquarters, Type_of_ownership, and Sector with meaningful labels.
- Standardized revenue values, replacing unknown entries with "Unknown".

5. Derived and Categorical Variables

- Created Year_of_Operation based on company founding year.
- Classified businesses into Small, Medium, Large, and Enterprise categories based on employee count.
- Categorized job titles into Data Analyst, Data Scientist, Data Engineer, Machine Learning Engineer, and Other.

6. Skill Extraction from Job Descriptions

- Identified presence of key skills (Python, SQL, Excel, Tableau, Power BI, AWS, Hadoop, Azure, Spark, Machine Learning, and MATLAB) in job descriptions.

7. Dropping Unnecessary Columns

- Removed irrelevant or redundant columns (Unnamed:_0, Salary_Estimate, Job_Title, Job_Description, Location, Size, Competitors).

Dataset Rows & Columns count

In [6]:
```
# Dataset Rows & Columns count
glassdoor_jobs.shape
```

Out[6]:
```
(956, 26)
```

Dataset Information

In [7]:
```
# Dataset Info
glassdoor_jobs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 956 entries, 0 to 955
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Rating             956 non-null    float64
 1   Company_Name       956 non-null    object 
 2   Headquarters       956 non-null    object 
 3   Founded            956 non-null    int64  
 4   Type_of_ownership  956 non-null    object 
 5   Industry           956 non-null    object 
 6   Sector             956 non-null    object 
 7   Revenue            956 non-null    object 
 8   Min_Salary         742 non-null    float64
 9   Max_Salary         956 non-null    int64  
 10  Avg_Salary         742 non-null    float64
 11  Job_State          956 non-null    object 
 12  new_job_title      956 non-null    object 
 13  Year_of_Operation  956 non-null    int64  
 14  Business_Size      956 non-null    object 
 15  Python             956 non-null    int64  
 16  Excel              956 non-null    int64  
 17  Tableau            956 non-null    int64  
 18  Power_BI           956 non-null    int64  
 19  SQL                956 non-null    int64  
 20  AWS                956 non-null    int64  
 21  Hadoop             956 non-null    int64  
 22  Azure              956 non-null    int64  
 23  Spark              956 non-null    int64  
 24  Machine_Learning   956 non-null    int64  
 25  Matlab             956 non-null    int64  
dtypes: float64(3), int64(14), object(9)
memory usage: 194.3+ KB
```

Duplicate Values

In [8]:
```
# Dataset Duplicate Value Count
glassdoor_jobs.duplicated().sum()
```

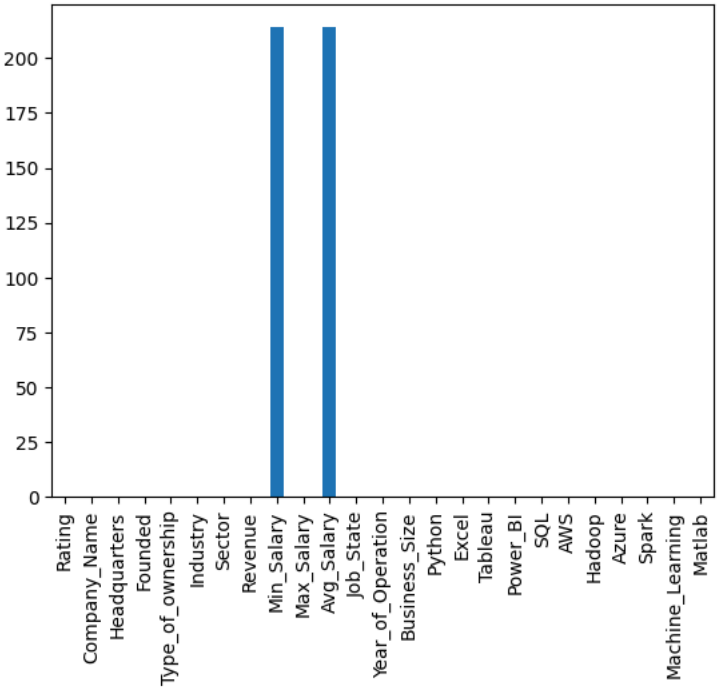
Out[8]:
```
np.int64(366)
```

Missing Values/Null Values

In [9]:
```
# Missing Values/Null Values Count
glassdoor_jobs.isnull().sum()
glassdoor_jobs.isna().sum()
```

Out[9]:
```
Rating                 0
Company_Name           0
Headquarters           0
Founded                0
Type_of_ownership      0
Industry               0
Sector                 0
Revenue                0
Min_Salary           214
Max_Salary             0
Avg_Salary           214
Job_State              0
new_job_title          0
Year_of_Operation      0
Business_Size          0
Python                 0
Excel                  0
Tableau                0
Power_BI               0
SQL                    0
AWS                    0
Hadoop                 0
Azure                  0
Spark                  0
Machine_Learning       0
Matlab                 0
dtype: int64
```

In [39]:
```
# Visualizing the missing values
glassdoor_jobs.isnull().sum().plot(kind='bar')
plt.show()
```



What did you know about your dataset?

The dataset consists of 956 job postings sourced from Glassdoor, providing key information about companies, job roles, salaries, and required skills. It includes 26 columns, covering the following aspects:

1. Company Information

Company_Name: Name of the company.

Headquarters: Location of the company's main office.

Founded: Year the company was established.

Type_of_ownership: Public, private, government, etc.

Revenue: Company revenue range.

Business_Size: Company size based on revenue.

2. Job Details

new_job_title: Job title after standardization.

Sector: Business sector (e.g., Technology, Healthcare).

Industry: Specific industry classification.

Job_State: Location of the job posting.

3. Salary Information

Min_Salary: Minimum salary offered (in thousands and in $).

Max_Salary: Maximum salary offered (in thousands and in $).

Avg_Salary: Average of min and max salary.

4. Company Ratings & Experience

Rating: Company rating (out of 5).

Year_of_Operation: Number of years since the company was founded.

5. Skill Requirements (Binary: 1 = required, 0 = not required)

Python, SQL, Excel, Tableau, Power_BI, AWS, Hadoop, Azure, Spark, Machine_Learning, Matlab.

3. Understanding Your Variables

In [10]:
```
# Dataset Columns
glassdoor_jobs.columns
```

Out[10]:
```
Index(['Rating', 'Company_Name', 'Headquarters', 'Founded',
       'Type_of_ownership', 'Industry', 'Sector', 'Revenue', 'Min_Salary',
       'Max_Salary', 'Avg_Salary', 'Job_State', 'new_job_title',
       'Year_of_Operation', 'Business_Size', 'Python', 'Excel', 'Tableau',
       'Power_BI', 'SQL', 'AWS', 'Hadoop', 'Azure', 'Spark',
       'Machine_Learning', 'Matlab'],
      dtype='object')
```

In [12]:
```
# Dataset Describe
glassdoor_jobs.describe()
```

Out[12]:

	Rating	Founded	Min_Salary	Max_Salary	Avg_Salary	Year_of_Operation	Python	Excel	Tableau	Power_BI	
count	956.000000	956.000000	742.000000	956.000000	742.000000	956.000000	956.000000	956.000000	956.000000	956.000000	956.
mean	3.636820	1980.172594	74.772237	99.746862	101.499326	44.827406	0.518828	0.508368	0.196653	0.055439	0.
std	0.920064	50.683191	30.926942	66.289661	37.463851	50.683191	0.499907	0.500192	0.397676	0.228956	0.
min	0.000000	1744.000000	15.000000	1.000000	15.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	3.300000	1968.000000	52.000000	59.000000	73.500000	14.000000	0.000000	0.000000	0.000000	0.000000	0.
50%	3.800000	1999.000000	69.500000	110.000000	97.500000	26.000000	1.000000	1.000000	0.000000	0.000000	1.
75%	4.200000	2011.000000	91.000000	143.000000	122.500000	57.000000	1.000000	1.000000	0.000000	0.000000	1.
max	5.000000	2025.000000	202.000000	306.000000	254.000000	281.000000	1.000000	1.000000	1.000000	1.000000	1.

Variables Description

1. Company & Job Info

Company_Name, Headquarters, Founded – Basic company details.

Type_of_ownership, Industry, Sector – Business classification.

Revenue, Business_Size – Company financials.

Job_State – Location of job posting.

2. Salary Data

Min_Salary, Max_Salary, Avg_Salary – Salary range (in thousands).

3. Company Ratings & Experience

Rating – Glassdoor rating (out of 5).

Year_of_Operation – Years since founding.

4. Job Role & Skills

new_job_title – Standardized job titles.

Python, SQL, Excel, Tableau, Power_BI, AWS, Hadoop, Azure, Spark, Machine_Learning, Matlab – Required skills (1 = Yes, 0 = No).

Check Unique Values for each variable.

In [14]:
```
# Check Unique Values for each variable.
glassdoor_jobs.nunique()
```

Out[14]:
```
Rating                32
Company_Name         448
Headquarters         235
Founded              109
Type_of_ownership     12
Industry              63
Sector                25
Revenue               13
Min_Salary           114
Max_Salary           161
Avg_Salary           221
Job_State             40
new_job_title          5
Year_of_Operation    109
Business_Size          5
Python                 2
Excel                  2
Tableau                2
Power_BI               2
SQL                    2
AWS                    2
Hadoop                 2
Azure                  2
Spark                  2
Machine_Learning       2
Matlab                 2
dtype: int64
```

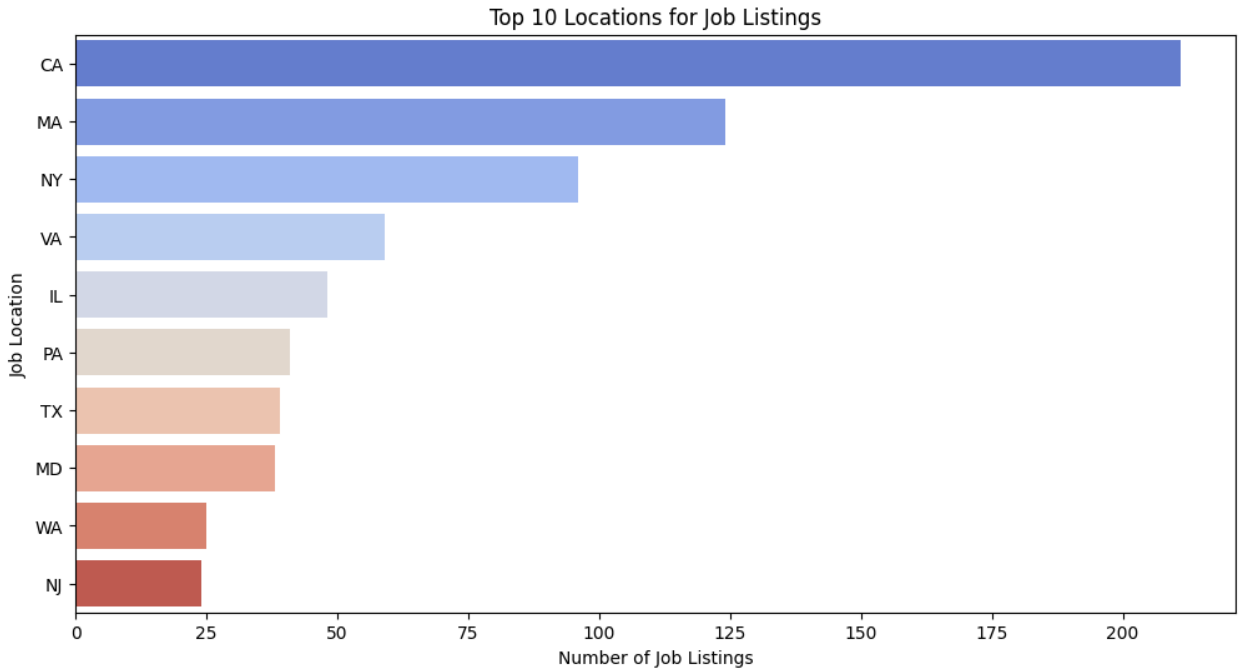
4. Data Vizualization, Storytelling & Experimenting with charts : Understand the relationships between variables

Chart 1. Top 10 Locations for Job Listings

In [43]:
```
# Chart - 1 Top 10 Locations for Job Listings

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

top_locations = glassdoor_jobs["Job_State"].value_counts().nlargest(10)
sns.barplot(x=top_locations.values, y=top_locations.index, hue=top_locations.index, palette="coolwarm", legend=False)
plt.xlabel("Number of Job Listings")
plt.ylabel("Job Location")
plt.title("Top 10 Locations for Job Listings")
plt.show()
```



1. Why did you pick the specific chart?

- A bar chart is ideal for categorical data like job locations.
- It clearly shows the top 10 states with the highest number of job listings.
- Easy comparison: It helps compare job availability across different locations.

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

- Tech hubs (e.g., California, New York, Texas) have the highest job postings.
- Smaller states may have fewer job listings, potentially due to lower industry presence.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

These insights help businesses and job seekers make better decisions:

For Employers:

- Companies can target high-demand locations for recruitment.
- Helps in workforce planning and expansion strategies.

For Job Seekers:

- Candidates can focus on locations with more job opportunities.
- Helps in relocation decisions based on job market trends.

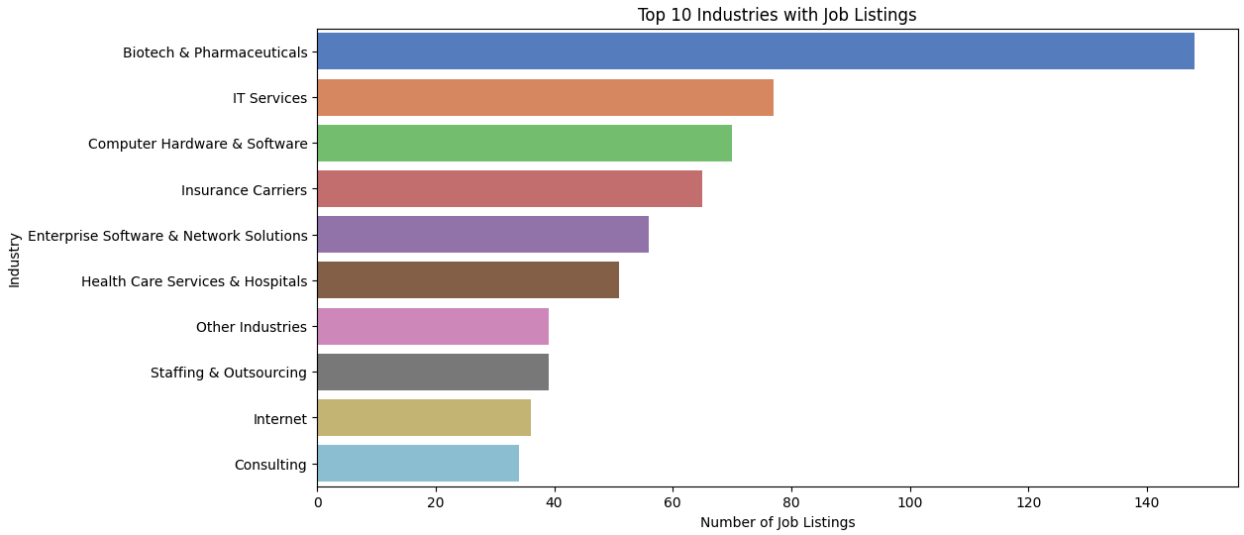
Insights That Lead to Negative Growth?

- States with low job listings may indicate slow economic growth, fewer industries, or lack of investment.
- Companies in low-job states may struggle to attract talent, leading to business stagnation.

Chart 2. Top 10 Industries with Job Listings

In [42]:
```
# Chart - 2 Top 10 Industries with Job Listings

plt.figure(figsize=(12, 6))
top_industries = glassdoor_jobs["Industry"].value_counts().nlargest(10)
sns.barplot(x=top_industries.values, y=top_industries.index, hue=top_industries.index, palette="muted", legend=False )
plt.xlabel("Number of Job Listings")
plt.ylabel("Industry")
plt.title("Top 10 Industries with Job Listings")
plt.show()
```



1. Why did you pick the specific chart?

- A bar chart is best for categorical data, making it easy to compare industries.
- It highlights the top 10 industries with the most job listings.
- The horizontal format ensures readability for industry names.

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

- Certain industries (e.g., Technology, Finance, Healthcare) have the highest number of job postings.
- Tech and Finance sectors dominate job postings, reflecting their growth and demand for skilled workers.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

The insights help both businesses and job seekers:

For Employers:

- Companies can benchmark hiring trends and adjust recruitment strategies accordingly.
- Helps HR teams to focus on competitive hiring in high-demand industries.

For Job Seekers:

- Candidates can target industries with high job availability.
- Encourages professionals to upskill in trending industries for better career opportunities.

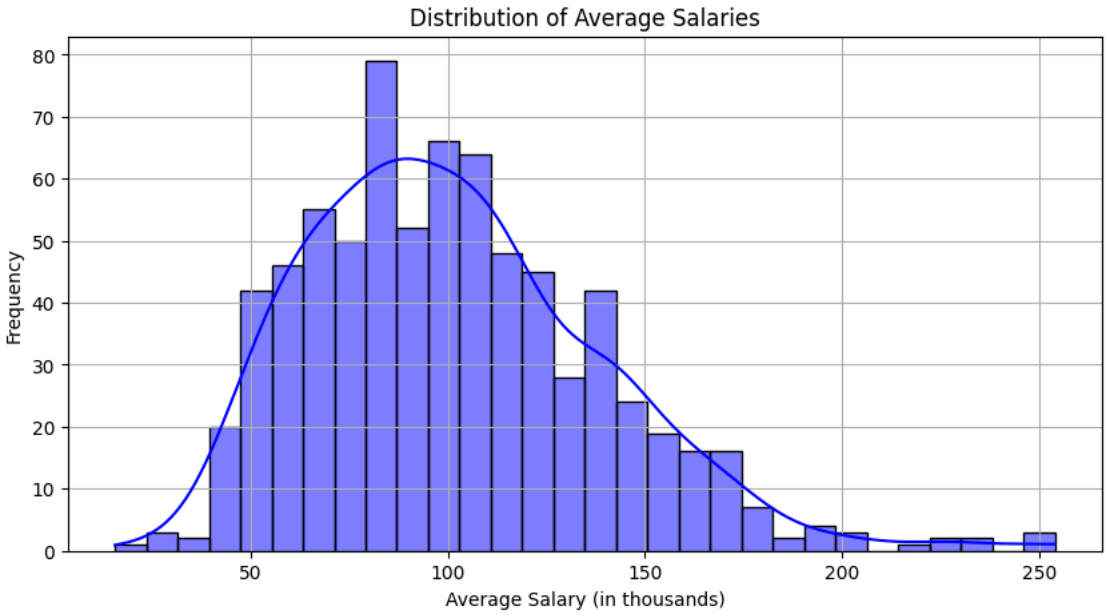
Insights That Lead to Negative Growth?

- Industries with low job postings may struggle with slow growth, automation, or declining demand.

Chart 3. Distribution of Average Salaries

In [17]:
```
# Chart - 3 Distribution of Average Salaries

plt.figure(figsize=(10, 5))
sns.histplot(glassdoor_jobs['Avg_Salary'].dropna(), bins=30, kde=True, color='blue')
plt.xlabel("Average Salary (in thousands)")
plt.ylabel("Frequency")
plt.title("Distribution of Average Salaries")
plt.grid(True)
plt.show()
```



1. Why did you pick the specific chart?

- A histogram is best for analyzing numerical data distributions like salaries.
- The KDE (Kernel Density Estimate) line helps visualize the overall salary trend.

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

- Most job postings offer salaries within a specific range, with a peak around $70K-$110K.
- Higher salaries ($200K+) are rare.

3. Will the gained insights help creating a positive business impact?

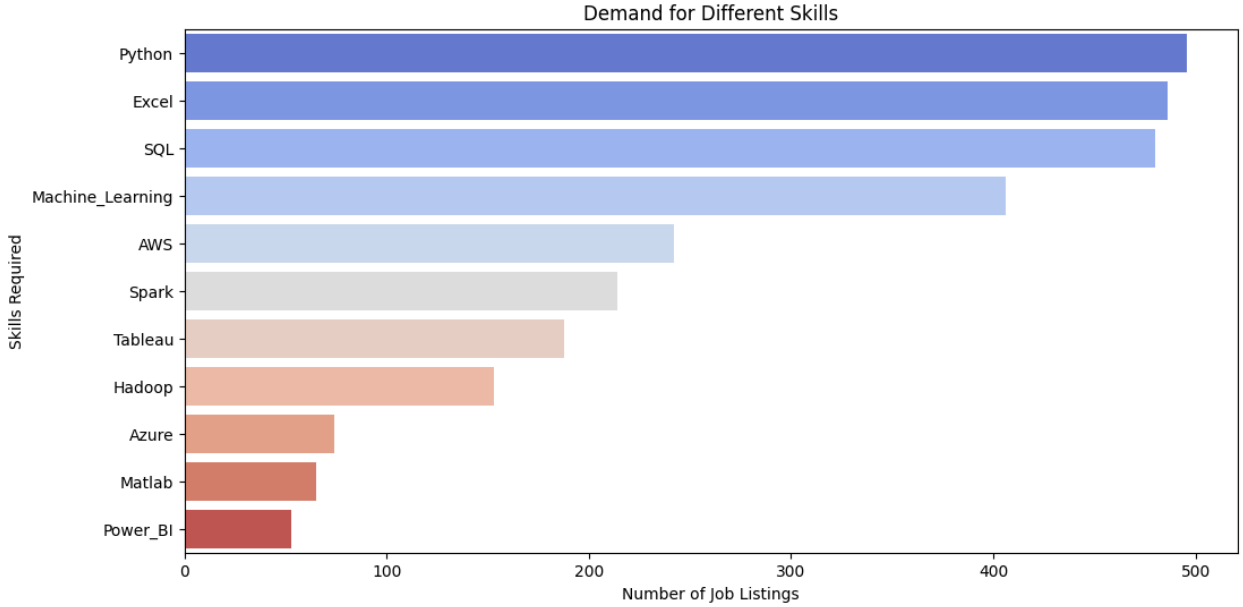
Are there any insights that lead to negative growth? Justify with specific reason.

- Helps job seekers set realistic salary expectations.
- Employers can position salary offerings competitively.

Chart 4. Demand for Different Skills

In [41]:
```
# Chart - 4 Demand for Different Skills

plt.figure(figsize=(12, 6))
skills = ["Python", "SQL", "Tableau", "Excel", "Power_BI", "AWS", "Hadoop", "Azure", "Spark", "Machine_Learning", "Matlab"]
skill_counts = glassdoor_jobs[skills].sum().sort_values(ascending = False)
sns.barplot(x=skill_counts.values, y=skill_counts.index, hue=skill_counts.index, palette="coolwarm", legend=False)
plt.xlabel("Number of Job Listings")
plt.ylabel("Skills Required")
plt.title("Demand for Different Skills")
plt.show()
```



1. Why did you pick the specific chart?

- A bar chart is best for categorical data, making it easy to compare skills.
- It helps visualize which skills are most required in job postings.

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

- Most in-demand skills: Python, Excel and SQL appear in the highest number of job listings.
- Data Visualization tools (Tableau, Power BI) are frequently required.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

Understanding skill demand helps:

For Job Seekers:

- Helps professionals focus on learning in-demand skills.
- Assists in career planning and choosing the right skillset for better job opportunities.

For Employers & Recruiters:

- Helps in targeted hiring by prioritizing essential skills.
- Ensures training programs align with market demand.

For Educational Institutions:

- Can design courses based on the most in-demand skills.

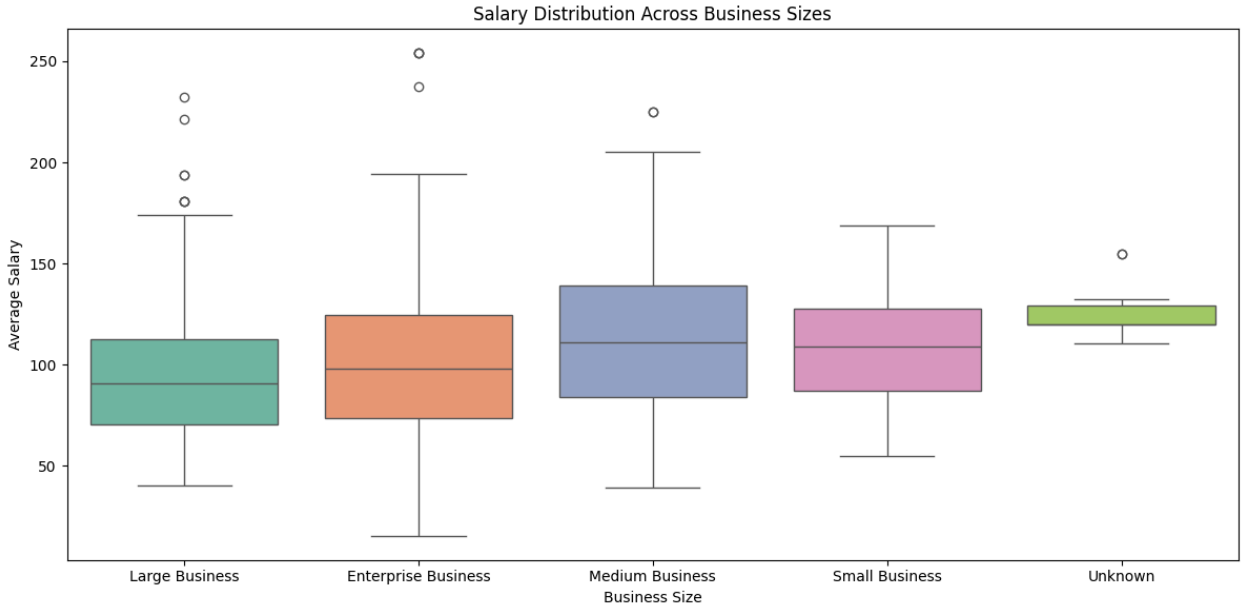
Chart 5. Salary Distribution Across Business Sizes

In [19]:
```
# Chart - 5 Salary Distribution Across Business Sizes

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

sns.boxplot(x=glassdoor_jobs["Business_Size"], y=glassdoor_jobs["Avg_Salary"], hue=glassdoor_jobs["Business_Size"], palette="Set2
plt.xlabel("Business Size")
plt.ylabel("Average Salary")
plt.title("Salary Distribution Across Business Sizes")

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



1. Why did you pick the specific chart?

- A box plot is ideal for showing salary distribution across different business sizes.
- It displays median, quartiles, and outliers, helping understand salary variation.

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

- Enterprise businesses have the widest salary range, suggesting high variation in pay across roles.
- Medium-sized businesses offer competitive salaries, often close to large enterprises.
- Small businesses have the lowest median salary.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

The insights help:

For Employers:

- Helps in salary benchmarking to stay competitive.
- Medium and small businesses can adjust salaries to attract top talent.

For Job Seekers:

- Helps in choosing the right company size based on salary expectations.
- Candidates can negotiate better salaries based on market insights.

Insights That Lead to Negative Growth?

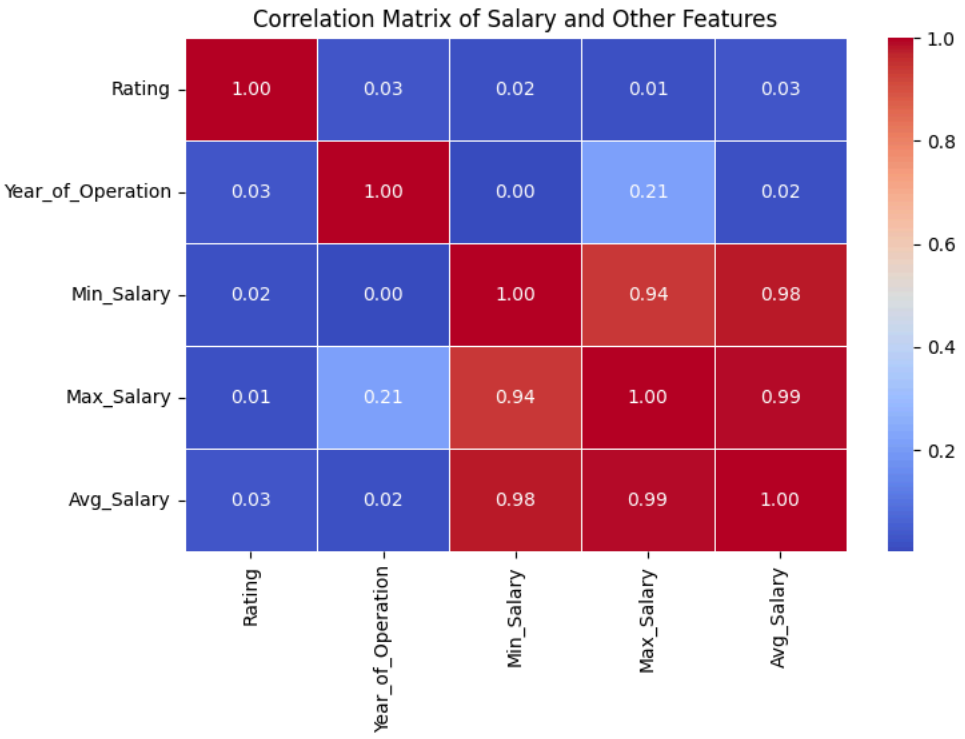
- If small companies consistently offer lower salaries, they may face talent shortages.
- High-skilled professionals may prefer larger firms, impacting small business growth.

Chart 6. Correlation between salary and other numerical features

In [ ]:
```
# Chart - 6 Correlation between salary and other numerical features

numeric_features = ["Rating", "Year_of_Operation", "Min_Salary", "Max_Salary", "Avg_Salary"]
correlation_matrix = glassdoor_jobs[numeric_features].corr()

plt.figure(figsize=(8, 5))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Correlation Matrix of Salary and Other Features")
plt.show()
```



1. Why did you pick the specific chart?

- A heatmap effectively visualizes correlations between numerical variables.
- It uses color gradients to highlight strong and weak relationships.
- Helps in quickly identifying patterns without requiring complex statistical analysis.

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

Weak or No Correlation Between Rating and Salary

- Higher-rated companies don't necessarily offer higher salaries.
- Other factors (company size, industry, location) influence salaries more.

Year of Operation has Weak Salary Correlation

- Older companies do not always pay more than newer ones.
- Startups and newer firms may offer competitive salaries to attract talent.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

These insights can help in:

For Employers:

- Helps new companies stay competitive by offering market-aligned salaries.

For Job Seekers:

- Encourages candidates to evaluate salary offers beyond company rating.
- Shows that young companies can be attractive employers based on salary competitiveness.

For HR & Recruitment:

- Salary decisions should focus on market demand rather than company age or rating.

Insights That Lead to Negative Growth?

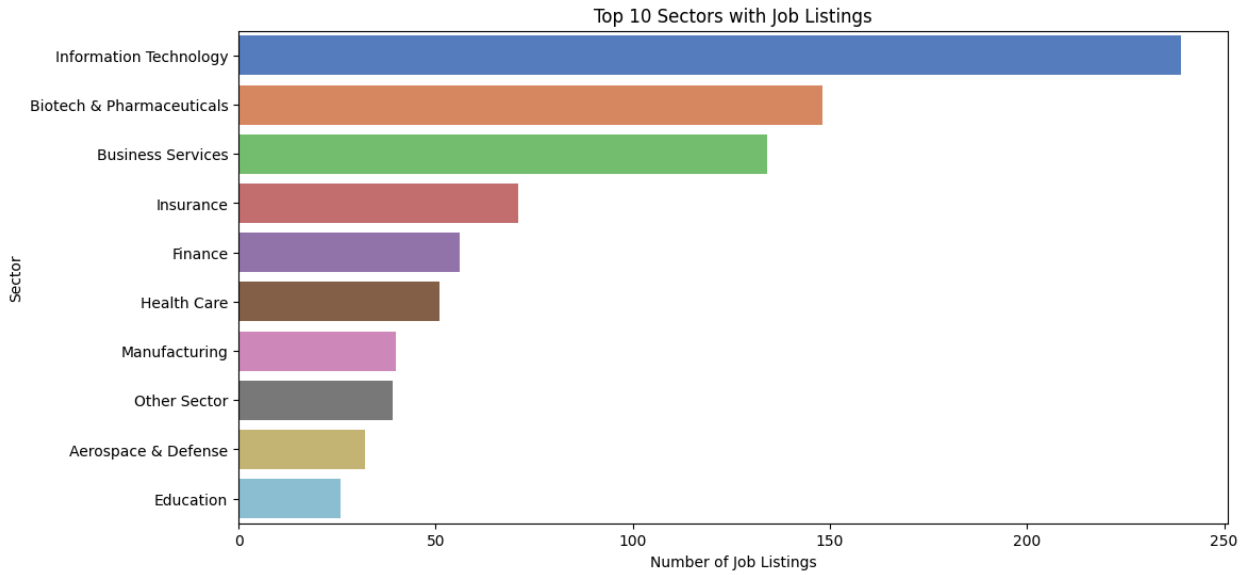
- Companies may prioritize culture, benefits, or work-life balance instead of salary.

Chart 7. Top 10 Sectors with Job Listings

In [27]:
```
# Chart - 7 Top 10 Sectors with Job Listings

plt.figure(figsize=(12, 6))
top_sectors = glassdoor_jobs["Sector"].value_counts().nlargest(10)
sns.barplot(x=top_sectors.values, y=top_sectors.index, hue=top_sectors.index, palette="muted", legend=False)
plt.xlabel("Number of Job Listings")
```

```
plt.ylabel("Sector")
plt.title("Top 10 Sectors with Job Listings")
plt.show()
```



1. Why did you pick the specific chart?

- A bar chart is best for categorical data, making it easy to compare skills.
- It provides a clear ranking of the top 10 sectors with the most job postings.
- Helps in quickly identifying which industries are hiring the most.

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

- Technology, Finance, and Healthcare sectors have the highest job demand.
- Sectors like Retail and Manufacturing have lower job postings, possibly due to automation or slower industry growth.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

The insights help:

For Employers & Recruiters:

- Helps companies understand competition for talent in high-demand sectors.

For Job Seekers:

- Encourages candidates to focus on high-growth industries for better career opportunities.
- Helps professionals upskill in trending sectors like tech and finance.

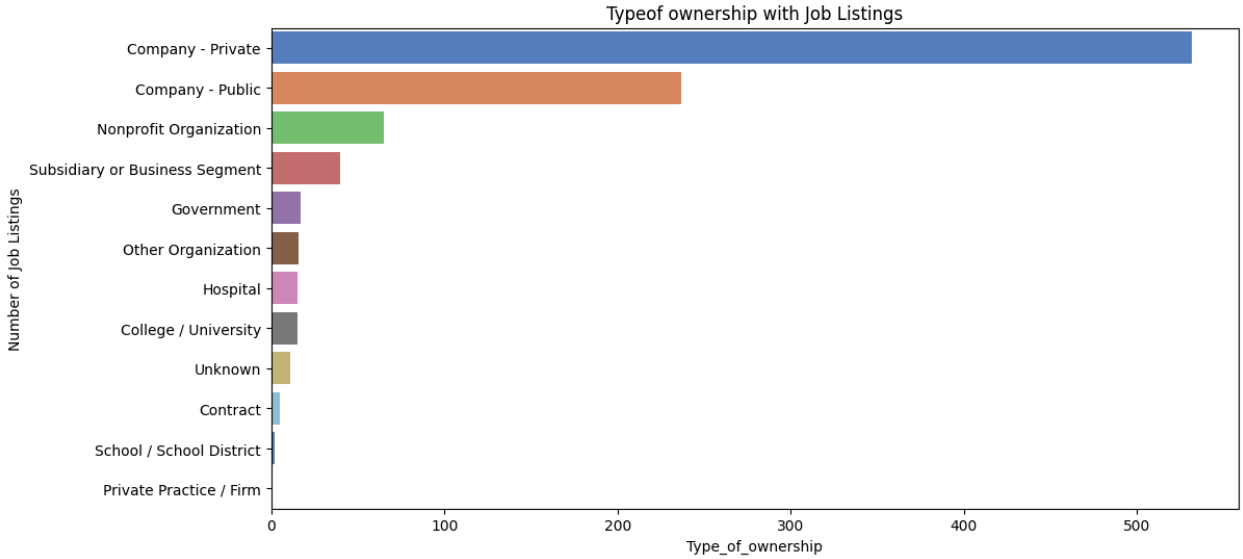
For Industry:

- Helpsinstitutions allocate resources to fast-growing sectors.
- Supports education & training programs to align with industry demands.

Chart 8. Typeof ownership with Job Listings

In [29]:
```
# Chart - 8 visualization code

plt.figure(figsize=(12, 6))
top_sectors = glassdoor_jobs["Type_of_ownership"].value_counts()
sns.barplot(x=top_sectors.values, y=top_sectors.index, hue=top_sectors.index, palette="muted", legend=False)
plt.xlabel("Type_of_ownership")
plt.ylabel("Number of Job Listings")
plt.title("Typeof ownership with Job Listings")
plt.show()
```



1. Why did you pick the specific chart?

- It provides a clear comparison of job postings across different ownership types (e.g., Public, Private, Government).
- Helps in understanding which ownership type dominates hiring.

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

- Private companies have the highest number of job postings, indicating they are the biggest employers.
- Public companies also contribute significantly.
- Government job listings are relatively lower, suggesting limited hiring or slower job creation.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

For Employers & Investors:

- Helps investors understand which ownership types drive employment growth.
- Encourages businesses to explore job market trends in private vs. public sectors.

For Job Seekers:

- Helps candidates choose between stable public-sector jobs or high-growth private-sector roles.
- Encourages professionals to upskill based on hiring trends in different ownership types.

Insights That Lead to Negative Growth

- A balanced job market requires both private-sector growth and stable government employment to ensure long-term economic sustainability.

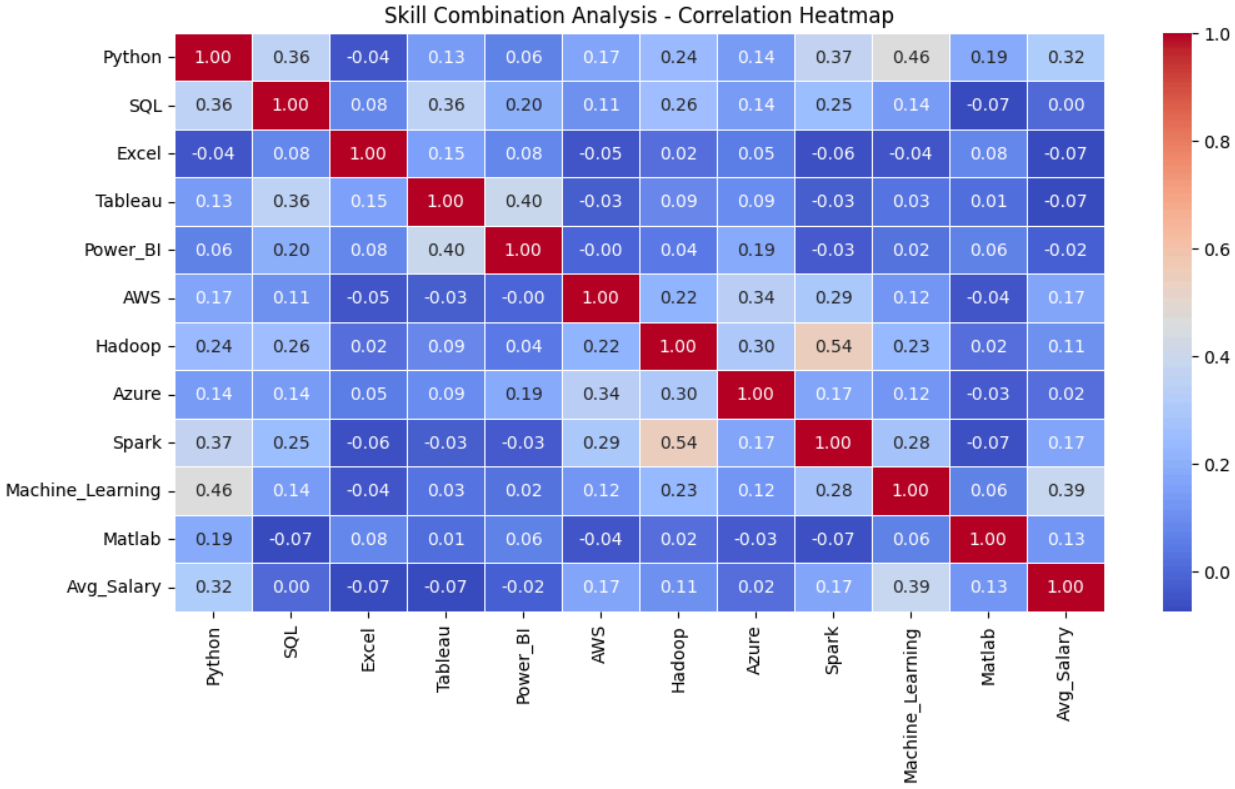
Chart 9. Skill Combination Analysis

In [77]:
```
# Chart - 9 Skill Combination Analysis

skill_columns = ["Python", "SQL", "Excel", "Tableau", "Power_BI", "AWS", "Hadoop", "Azure", "Spark", "Machine_Learning", "Matlab"

skill_corr = glassdoor_jobs[skill_columns].corr()

plt.figure(figsize=(12, 6))
sns.heatmap(skill_corr, annot=True, cmap="coolwarm", linewidths=0.5, fmt=".2f")
plt.title("Skill Combination Analysis - Correlation Heatmap")
plt.show()
```



1. Why did you pick the specific chart?

- A heatmap is ideal for visualizing correlations between multiple skills and salary in a compact format.

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

- SQL, Python, and Excel are often required together.
- AWS, Azure, and Spark show high correlation, indicating demand for cloud computing skills.
- Machine Learning, AWS, and Spark likely have a stronger positive correlation with Avg_Salary, indicating higher salaries for advanced technical roles.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

These insights help:

For Job Seekers:

- Helps professionals focus on learning high-paying skills (e.g., AWS, Machine Learning).
- Encourages combining complementary skills (e.g., Python + SQL for data jobs).

For Employers & Recruiters:

- Guides skill-based hiring decisions, ensuring the right mix of expertise.
- Helps optimize training programs by focusing on in-demand skills.

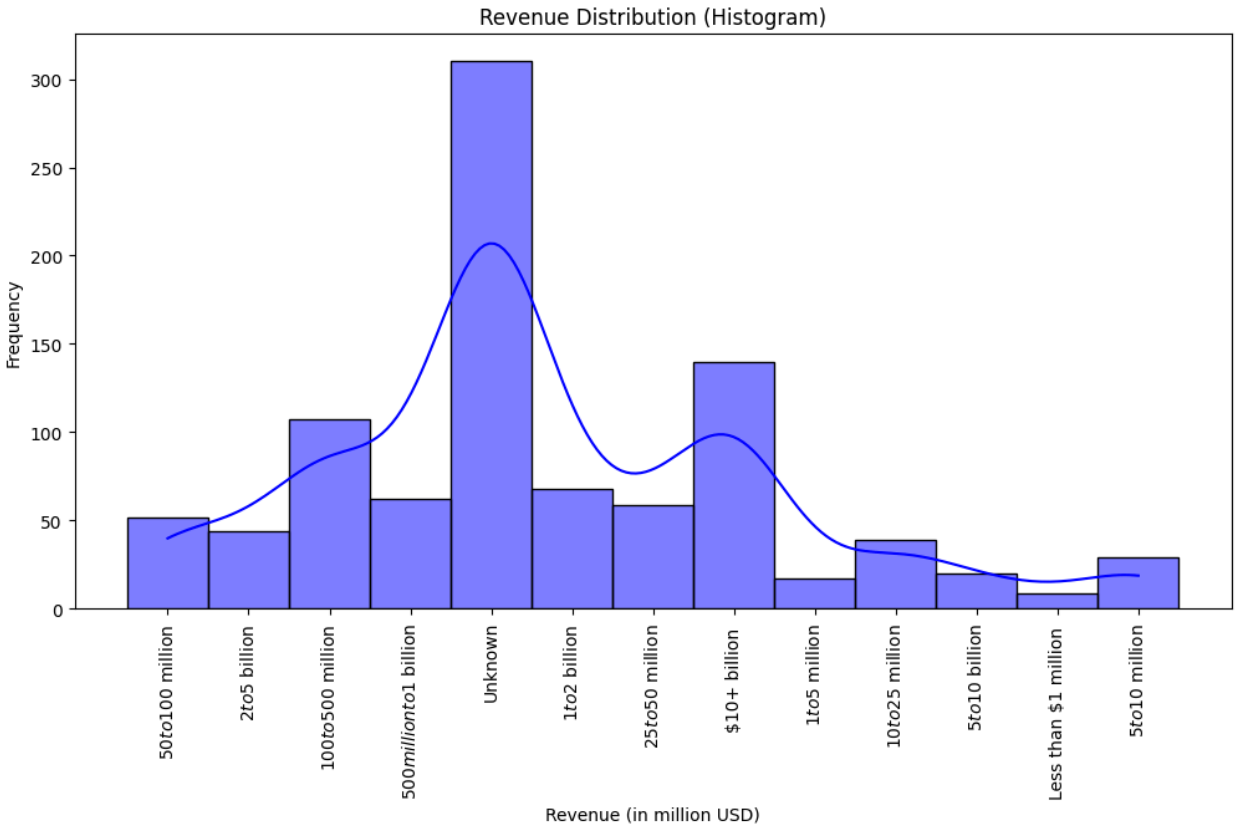
For Training Institutes & Educators:

- Helps design upskilling programs based on salary impact.
- Supports curriculum alignment with employer demands.

Chart 10. Revenue Distribution

In [40]:
```
# Chart - 10 visualization code

plt.figure(figsize=(12, 6))
sns.histplot(glassdoor_jobs["Revenue"], bins=20, kde=True, color="blue")
plt.xlabel("Revenue (in million USD)")
plt.ylabel("Frequency")
plt.title("Revenue Distribution (Histogram)")
plt.xticks(rotation=90)
plt.show()
```



1. Why did you pick the specific chart?

- A histogram is the best way to visualize the distribution of numerical data like revenue.
- The KDE (Kernel Density Estimate) line helps identify trends and patterns in revenue values.

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

- Majority of job postings belong to mid-revenue ($1M-$500M) companies.

3. Will the gained insights help creating a positive business impact?

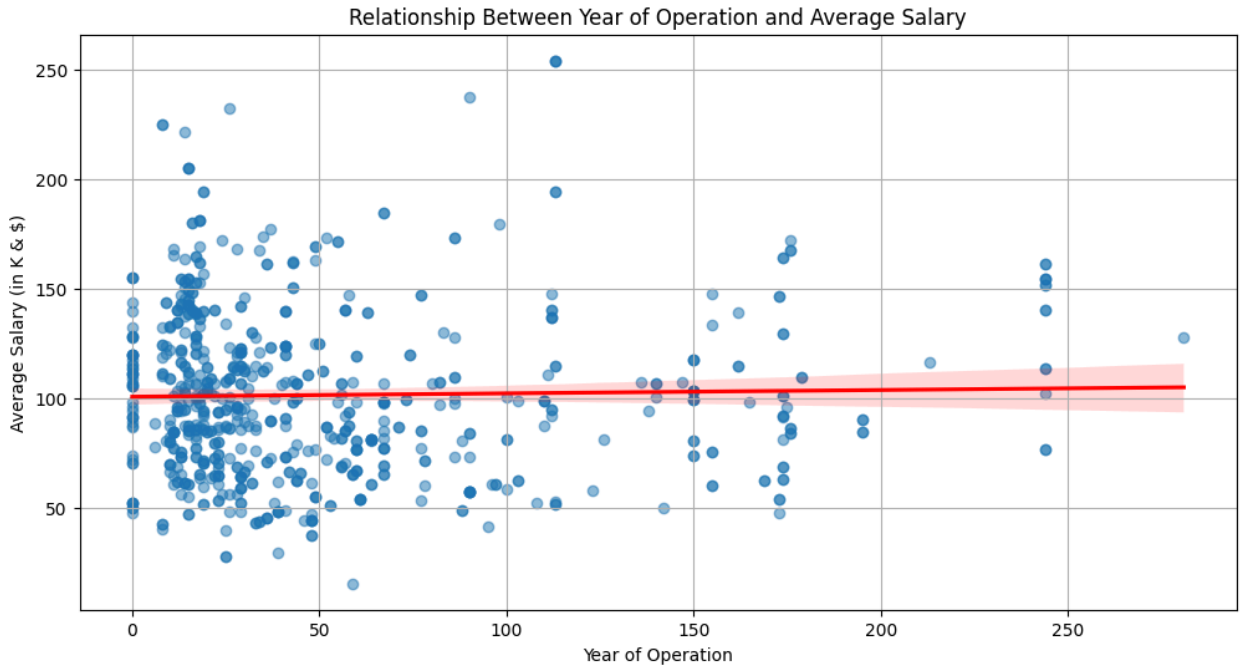
Are there any insights that lead to negative growth? Justify with specific reason.

- Small and mid-sized firms remain significant job providers.

Chart 11. Relationship Between Year of Operation and Average Salary

In [34]:
```
# Chart - 11 visualization code

plt.figure(figsize=(12, 6))
sns.regplot(x=glassdoor_jobs["Year_of_Operation"], y=glassdoor_jobs["Avg_Salary"], scatter_kws={"alpha": 0.5}, line_kws={"color":
plt.xlabel("Year of Operation")
plt.ylabel("Average Salary (in K & $)")
plt.title("Relationship Between Year of Operation and Average Salary")
plt.grid(True)
plt.show()
```



1. Why did you pick the specific chart?

- A regression plot is ideal for analyzing relationships between two continuous variables.
- It helps in understanding how the number of years a company has been operating affects average salary.

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

- Companies established in the last 20 years often offer competitive salaries.
- Well-established companies (50+ years) may offer stable but moderate salary growth.

3. Will the gained insights help creating a positive business impact?

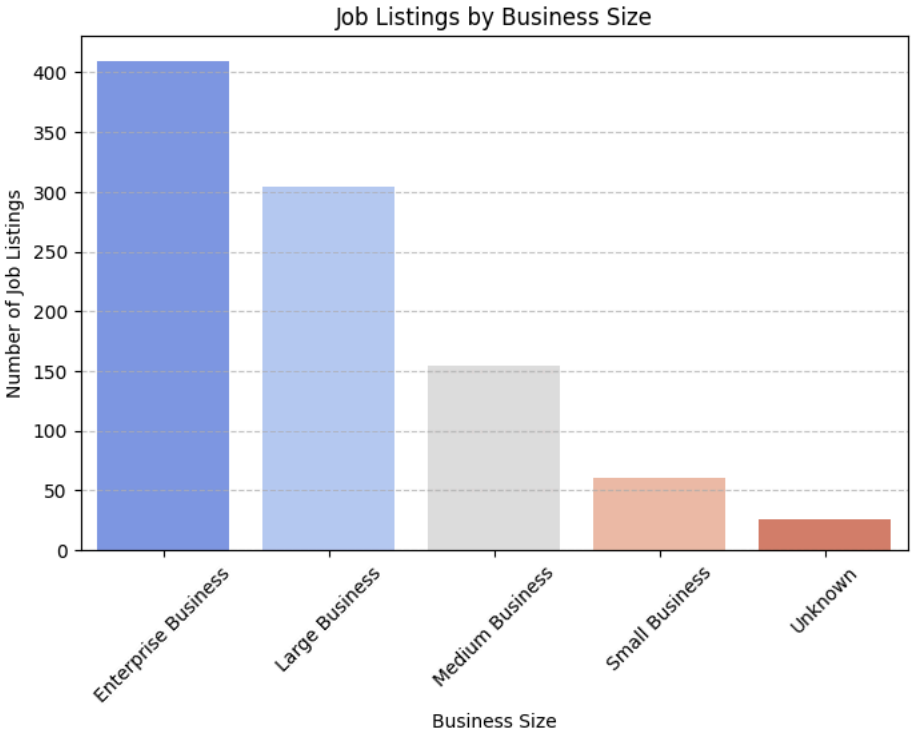
Are there any insights that lead to negative growth? Justify with specific reason.

- Job seekers should not disregard startups, as they may offer lucrative salaries.
- Established firms may focus more on benefits over base salary.

Chart 12. Job Listings by Business Size

In [57]:
```
# Chart - 12 visualization code

plt.figure(figsize=(8, 5))
business_size_counts = glassdoor_jobs["Business_Size"].value_counts()
sns.barplot(x=business_size_counts.index, y=business_size_counts.values,hue=business_size_counts.index, legend=False, palette="co
plt.xlabel("Business Size")
plt.ylabel("Number of Job Listings")
plt.title("Job Listings by Business Size")
plt.xticks(rotation=45)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()
```



1. Why did you pick the specific chart?

- A bar chart is ideal for categorical data, making it easy to compare job listings across different business sizes.

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

- Enterprise & Large Businesses Have the Most Job Listings.
- Medium-Sized Companies Also Post a Good Number of Jobs.
- Small Businesses Have Fewer Job Listings.

3. Will the gained insights help creating a positive business impact?

Are there any insights that lead to negative growth? Justify with specific reason.

These insights help:

For Employers & Recruiters

- Small and medium businesses can adjust recruitment strategies to stay competitive.

For Job Seekers

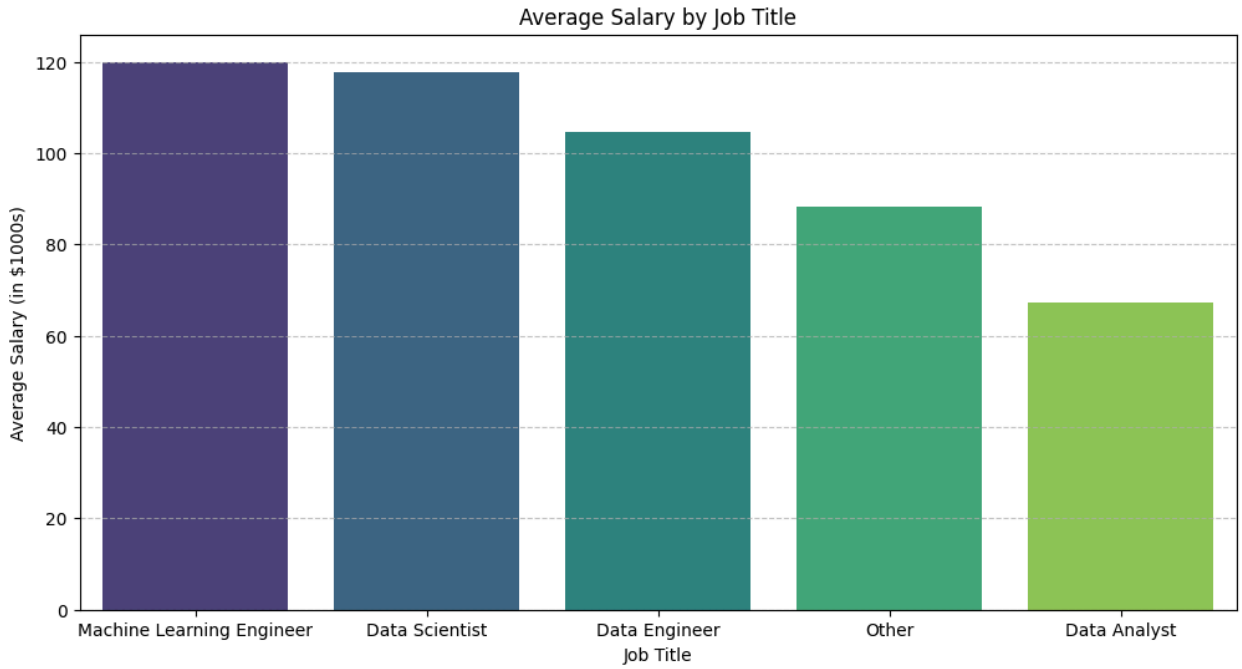
- Encourages candidates to target companies based on hiring trends.

Chart 13. Average Salary by Job Title

In [79]:
```
# Average Salary by Job Title

avg_salary_by_job = glassdoor_jobs.groupby("new_job_title")["Avg_Salary"].mean().sort_values(ascending=False)

# Create bar plot
plt.figure(figsize=(12, 6))
sns.barplot(x=avg_salary_by_job.index, y=avg_salary_by_job.values, hue=avg_salary_by_job.index, legend=False, palette="viridis")
plt.xlabel("Job Title")
plt.ylabel("Average Salary (in $1000s)")
plt.title("Average Salary by Job Title")
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()
```



1. Why did you pick the specific chart?

- A bar chart is the best choice for categorical data, making it easy to compare average salaries across different job titles.
- The sorted order helps in quickly identifying high-paying vs. low-paying jobs.

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

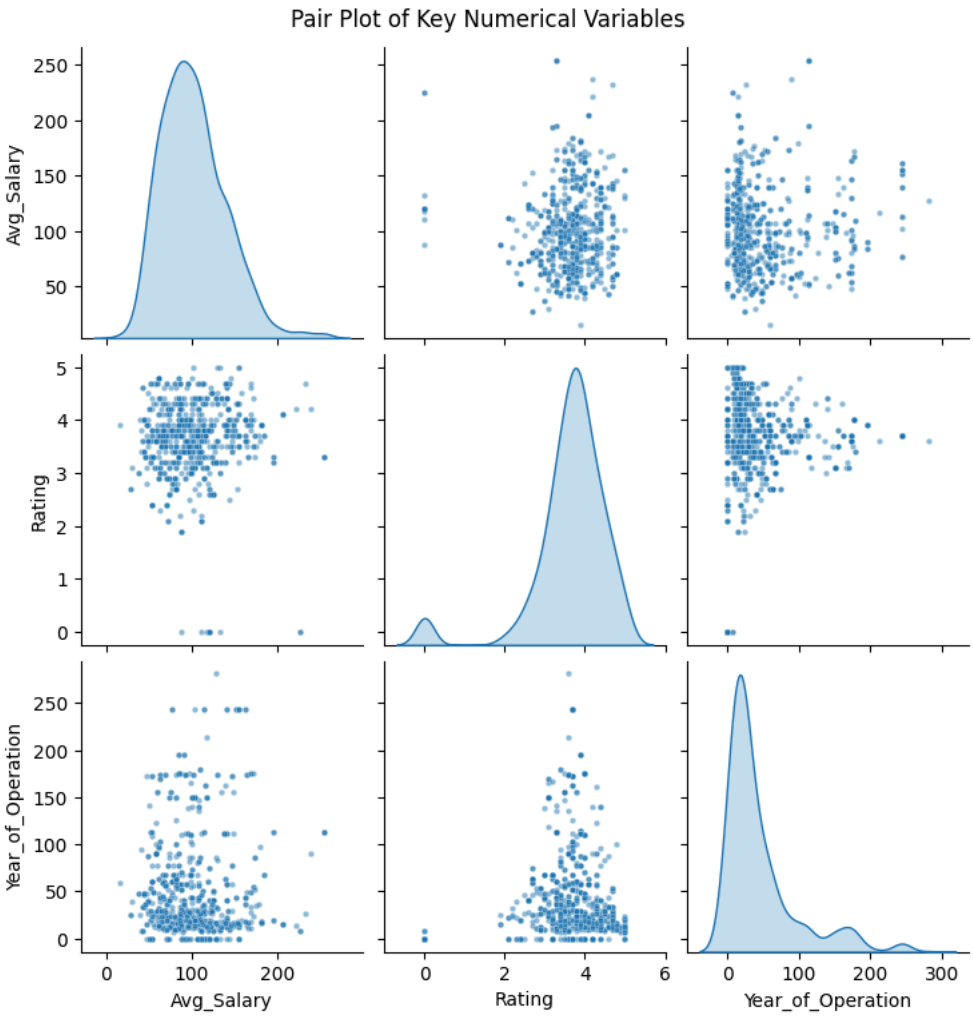
- Roles like Machine Learning Engineer, Data Scientist, and Data Engineer have higher average salaries.
- Data Analysts tend to have lower average salaries.

Chart 14. Pair Plot

In [51]:
```
# Pair Plot visualization code

glassdoor_jobs_pairplot = glassdoor_jobs[["Avg_Salary", "Rating", "Year_of_Operation"]].copy()

sns.pairplot(glassdoor_jobs_pairplot, diag_kind="kde", plot_kws={'alpha': 0.5, 's': 10})
plt.suptitle("Pair Plot of Key Numerical Variables", y=1.02)
plt.show()
```



1. Why did you pick the specific chart?

- A pair plot is ideal for analyzing relationships between multiple numerical variables in a single visualization.

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

- Rating and salary do not show a clear correlation, supporting the idea that salaries are influenced more by industry and job role.
- Helps job seekers understand that company rating alone is not a good salary predictor.
- Employers can use operational years and industry-specific benchmarks to determine salary structures.

5. Solution to Business Objective

What do you suggest the client to achieve Business Objective ?

Explain Briefly.

For Job Seekers:

- Prioritize skill development in Python, SQL, and data visualization tools (Tableau, Power BI) to increase employability.
- Focus on high-demand industries like Tech, Finance, and Healthcare.
- Consider geographic location for better job opportunities.

For Employers & Recruiters:

- Offer competitive salaries aligned with industry benchmarks to attract top talent.
- Invest in training programs to develop necessary skills within the workforce.
- Consider salary restructuring based on business size and industry demands.

For Industry Analysts & Policymakers:

- Support workforce development initiatives to bridge skill gaps.

- Encourage policy changes that promote fair salary practices across industries.

Conclusion

This project successfully provides insights into the job market using Glassdoor job postings. It highlights salary trends, industry demand, and skill importance, benefiting job seekers, employers, and industry analysts.

Key Takeaways:

- Salaries vary more by job role and skills than by company rating.
- Tech, Finance, and Healthcare industries dominate job postings.
- Python, SQL, and Excel remain the most sought-after skills.
- Startups and mid-sized businesses offer competitive salaries, making them viable job options.

Hurrah! You have successfully completed your EDA Capstone Project !!!

In [ ]:

In [ ]: