Import Libraries

In [1]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

In [ ]:
```
```

Data Cleaning & Preparation

Import Company Data

In [2]:
```
df_c = pd.read_csv('companies.csv')
```

In [3]:
```
df_c.head(5)
```

Out[3]:

	company_id	name	description	company_size	state	country	city	zip_code	address	
0	1009	IBM	At IBM, we do more than work. We create. We cr...	7.0	NY	US	Armonk, New York	10504	International Business Machines Corp.	https://www.linkedin.cc
1	1016	GE HealthCare	Every day millions of people feel the impact o...	7.0	0	US	Chicago	0	-	https://www.linkedin.com/comp
2	1021	GE Power	GE Power, part of GE Vernova, is a world energ...	7.0	NY	US	Schenectady	12345	1 River Road	https://www.linkedin.com/cc
3	1025	Hewlett Packard Enterprise	Official LinkedIn of Hewlett Packard Enterpris...	7.0	Texas	US	Houston	77389	1701 E Mossy Oaks Rd Spring	https://www.linkedin.com/c
4	1028	Oracle	We're a cloud technology company that provides...	7.0	Texas	US	Austin	78741	2300 Oracle Way	https://www.linkedin.com

In [4]:
```
df_c.isna().sum()
```

Out[4]:
```
company_id         0
name               1
description      143
company_size    1105
state              8
country            0
city               1
zip_code          12
address           11
url                0
dtype: int64
```

In [5]:
```
#drop columns not needed
df_c.drop(columns= ['address','url','description'], inplace=True)
```

In [6]:
```
#drop companies with no name
df_c.dropna(subset=['name'], inplace=True)
```

In [7]:
```
df_c
```

Out[7]:

	company_id	name	company_size	state	country	city	zip_code
0	1009	IBM	7.0	NY	US	Armonk, New York	10504
1	1016	GE HealthCare	7.0	0	US	Chicago	0
2	1021	GE Power	7.0	NY	US	Schenectady	12345
3	1025	Hewlett Packard Enterprise	7.0	Texas	US	Houston	77389
4	1028	Oracle	7.0	Texas	US	Austin	78741
...	...	...	...	...	...	...	...
11356	3700144594	BYREDO	3.0	0	0	0	0
11357	3700144710	Pros2Plan, a division of Spinnaker SCA	2.0	CO	US	Boulder	80303
11358	3700147810	Ascendo Resources	3.0	FL	US	Coral Gables	33134
11359	3700150295	The Crox Group	2.0	Illinois	US	Lincolnwood	60712
11360	3700152513	Sidley Austin LLP	5.0	Illinois	US	Chicago	60603

11360 rows × 7 columns

Import Industry Data

In [8]:
```
df_i = pd.read_csv('company_industries.csv')
```

In [9]:
```
df_i.head(5)
```

Out[9]:

	company_id	industry
0	81149246	Higher Education
1	10033339	Information Technology & Services
2	6049228	Accounting
3	2641066	Electrical & Electronic Manufacturing
4	96649998	Marketing & Advertising

In [10]:
```
#merge companies and industry data
df_ci= pd.merge(df_c, df_i, on='company_id', how='inner')
```

In [11]:
```
df_ci
```

Out[11]:

	company_id	name	company_size	state	country	city	zip_code	industry
0	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services
1	1009	IBM	7.0	NY	US	Armonk, New York	10504	IT Services and IT Consulting
2	1016	GE HealthCare	7.0	0	US	Chicago	0	Hospital & Health Care
3	1016	GE HealthCare	7.0	0	US	Chicago	0	Hospitals and Health Care
4	1021	GE Power	7.0	NY	US	Schenectady	12345	Renewables & Environment
...	...	...	...	...	...	...	...	...
12550	101065652	AUX CLOUD COMMERCE(USA)INC	1.0	0	0	0	0	Appliances, Electrical, and Electronics Manufa...
12551	101068739	IZI1 Operations, LLC	1.0	Florida	US	St Petersberg	0	Truck Transportation
12552	101069729	SunSource Power	1.0	0	0	0	0	IT Services and IT Consulting
12553	101173981	Cultivate Charlottesville	1.0	VA	US	Charlottesville	22911	Non-profit Organizations
12554	101174062	Drag Story Hour	NaN	CA	US	San Francisco	0	Non-profit Organizations

12555 rows × 8 columns

In [12]:
```
df_ci['industry'].unique()
```

Out[12]:
```
array(['Information Technology & Services',
       'IT Services and IT Consulting', 'Hospital & Health Care',
       'Hospitals and Health Care', 'Renewables & Environment',
       'Renewable Energy Semiconductor Manufacturing',
       'Business Consulting and Services', 'Management Consulting',
       'Industrial Automation', 'Telecommunications', 'Computer Software',
       'Software Development', 'Financial Services', 'Consumer Goods',
       'Banking', 'Pharmaceuticals', 'Pharmaceutical Manufacturing',
       'Medical Equipment Manufacturing', 'Armed Forces',
       'Entertainment Providers', 'Entertainment',
       'Electrical & Electronic Manufacturing',
       'Appliances, Electrical, and Electronics Manufacturing',
       'Aviation and Aerospace Component Manufacturing', 'Oil & Energy',
       'Oil and Gas', 'Food and Beverage Services', 'Manufacturing',
       'Defense & Space', 'Defense and Space Manufacturing',
       'Food & Beverages', 'Human Resources', 'Human Resources Services',
       'Motor Vehicle Manufacturing', 'Semiconductors',
       'Semiconductor Manufacturing', 'Food Production',
       'Food and Beverage Manufacturing', 'Retail',
       'Transportation/Trucking/Railroad', 'Truck Transportation',
       'Biotechnology', 'Biotechnology Research', 'Medical Device',
       'Higher Education', 'Personal Care Product Manufacturing',
       'Package/Freight Delivery', 'Freight and Package Transportation',
       'Advertising Services', 'Staffing & Recruiting',
       'Staffing and Recruiting',
       'Broadcast Media Production and Distribution', 'Insurance',
       'Information Services', 'Automotive', 'Leisure, Travel & Tourism',
       'Mechanical Or Industrial Engineering',
       'Industrial Machinery Manufacturing', 'Airlines and Aviation',
       'Hospitality', 'Real Estate', 'Automation Machinery Manufacturing',
       'Computer and Network Security', 'Accounting',
       'Marketing & Advertising', 'Computer Games', 'Broadcast Media',
       'Chemicals', 'Chemical Manufacturing', 'Aviation & Aerospace',
       'Machinery Manufacturing', 'Government Administration',
       'Administration of Justice', 'Law Enforcement', 'Restaurants',
       'Wholesale Building Materials', 'Computer Hardware Manufacturing',
       'Education Management', 'Airlines/Aviation', 'Plastics',
       'Musicians', 'Mining', 'Translation & Localization',
       'Retail Apparel and Fashion',
       'Paper and Forest Product Manufacturing', 'Construction',
       'Packaging and Containers Manufacturing', 'Fundraising',
       'Education Administration Programs', 'Paper & Forest Products',
       'Public Relations & Communications',
       'Public Relations and Communications Services', 'Research',
       'Glass, Ceramics & Concrete',
       'Glass, Ceramics and Concrete Manufacturing', 'Sporting Goods',
       'Sporting Goods Manufacturing', 'Machinery',
       'Business Supplies & Equipment', 'Retail Office Equipment',
       'Printing', 'Printing Services', 'E-Learning Providers', 'Farming',
       'Utilities', 'International Trade & Development',
       'Outsourcing and Offshoring Consulting',
       'Health, Wellness & Fitness', 'Law Practice',
       'Book and Periodical Publishing', 'Spectator Sports',
       'Apparel & Fashion', 'Religious Institutions', 'Research Services',
       'Wellness and Fitness Services',
       'International Trade and Development', 'Photography',
       'Non-profit Organization Management', 'Logistics & Supply Chain',
       'Consumer Services', 'Luxury Goods & Jewelry',
       'Facilities Services', 'Cosmetics', 'Computer Hardware',
       'Investment Banking', 'Newspaper Publishing', 'Mining & Metals',
       'Security & Investigations', 'Beverage Manufacturing',
       'Retail Luxury Goods and Jewelry', 'Building Materials',
       'Executive Office', 'Internet',
       'Technology, Information and Internet', 'Consumer Electronics',
       'Computers and Electronics Manufacturing',
       'Architecture and Planning', 'Travel Arrangements',
       'Non-profit Organizations', 'Civil Engineering', 'Wholesale',
       'Environmental Services', 'Events Services',
       'Civic & Social Organization', 'Packaging & Containers',
       'Animation', 'Animation and Post-production',
       'Primary and Secondary Education', 'Sports',
       'International Affairs', 'Textile Manufacturing',
       'Museums, Historical Sites, and Zoos',
       'Gambling Facilities and Casinos', 'Newspapers',
       'Plastics Manufacturing', 'Furniture',
       'Furniture and Home Furnishings Manufacturing',
       'Primary/Secondary Education', 'Publishing', 'Mental Health Care',
       'Public Policy Offices', 'Design', 'Architecture & Planning',
       'Performing Arts', 'Think Tanks', 'Public Safety',
       'Museums & Institutions', 'Civic and Social Organizations',
       'Computer & Network Security', 'Libraries', 'Graphic Design',
       'Textiles', 'Security and Investigations', 'Wireless',
       'Design Services', 'Media Production', 'Market Research',
       'Individual & Family Services', 'Individual and Family Services',
```

```
'Outsourcing/Offshoring', 'Supermarkets', 'Retail Groceries',
'Executive Offices', 'Wine & Spirits', 'Investment Management',
'Railroad Equipment Manufacturing',
'Venture Capital and Private Equity Principals', 'Shipbuilding',
'Transportation, Logistics, Supply Chain and Storage',
'Legal Services', 'Translation and Localization',
'Government Relations', 'Music', 'Online Audio and Video Media',
'Medical Practices', 'Writing & Editing', 'Gambling & Casinos',
'E-learning', 'Artists and Writers', 'Recreational Facilities',
'Professional Training & Coaching',
'Government Relations Services', 'Legislative Office', 'Dairy',
'Public Policy', 'Online Media',
'Recreational Facilities & Services', 'Political Organization',
'Venture Capital & Private Equity', 'Computer Networking',
'Retail Art Supplies', 'Alternative Medicine',
'Writing and Editing', 'Commercial Real Estate',
'Veterinary Services', 'Philanthropy',
'Professional Training and Coaching', 'Import & Export', 'Fishery',
'Maritime Transportation', 'Military', 'Medical Practice',
'Wholesale Import and Export', 'Tobacco', 'Railroad Manufacture',
'Tobacco Manufacturing', 'Capital Markets', 'Judiciary',
'Veterinary', 'Alternative Dispute Resolution', 'Maritime',
'Leasing Non-residential Real Estate', 'Warehousing',
'Dairy Product Manufacturing', 'Fine Art'], dtype=object)
```

In [13]:
```
#rename rows for standardization
df_ci['industry']=df_ci['industry'].replace({'IT Services':'Information Technology & Services', 'Hospital & Health
                                             'Hospitals and Health Care':'Health Care', 'Renewables & Environment':'
                                              'Pharmaceutical':'Pharmaceuticals','Entertainment Providers':'Entertai
                                              'Appliances, Electrical, and Electronics Manufacturing':'Electricals &
                                              'Leasing Non-residential Real Estate':'Real Estate', 'Warehousing':'Rea
                                              'Food and Beverage Manufacturing':'Manufacturing','Transportation, Log
                                              'Railroad Equipment Manufacturing':'Manufacturing','Plastics Manufactu
                                               'Motor Vehicle Manufacturing':'Manufacturing', 'Semiconductors':'Elec
                                              'Higher Education':'Education','Personal Care Product Manufacturing':'
                                              'Mechanical Or Industrial Engineering':'Engineering','Political Organi
                                               'Chemical Manufacturing':'Manufacturing', 'Machinery Manufacturing':'
                                              'E-learning':'Education','Recreational Facilities & Services':'Recreat
                                               'Mental Health Care':'Health Care','Education Management':'Education'
                                                 'Business Supplies & Equipment':'Equipment', 'Retail Office Equipme
                                                  'Public Policy Offices':'Government', 'Design':'Architecture','Arc
                                                 'Research Services':'Research', 'Automotive': 'Automation','Industr
                                              'Wholesale Building Materials':'Building Materials', 'Computer Hardwar
```

In [14]:
```
df_ci
```

Out[14]:

	company_id	name	company_size	state	country	city	zip_code	industry
0	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services
1	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services
2	1016	GE HealthCare	7.0	0	US	Chicago	0	Health Care
3	1016	GE HealthCare	7.0	0	US	Chicago	0	Health Care
4	1021	GE Power	7.0	NY	US	Schenectady	12345	Renewable Energy
...	...	...	...	...	...	...	...	...
12550	101065652	AUX CLOUD COMMERCE(USA)INC	1.0	0	0	0	0	Electricals & Electronics
12551	101068739	IZI1 Operations, LLC	1.0	Florida	US	St Petersberg	0	Transportation
12552	101069729	SunSource Power	1.0	0	0	0	0	Information Technology & Services
12553	101173981	Cultivate Charlottesville	1.0	VA	US	Charlottesville	22911	Non-profit Organizations
12554	101174062	Drag Story Hour	NaN	CA	US	San Francisco	0	Non-profit Organizations

12555 rows × 8 columns

Import Company Speciality Data

In [15]:
```
df_cs=pd.read_csv('company_specialities.csv')
```

In [16]:
```
df_cs
```

Out[16]:

	company_id	speciality
0	81149246	Childrens Music Education
1	81149246	Foundational Music Theory
2	81149246	Child Music Lessons
3	81149246	social emotional learning
4	81149246	social emotional development
...	...	...
78400	2293632	LED Billboards
78401	2293632	Electronic Message Centers
78402	2293632	Digital Signage
78403	2293632	Outdoor Digital Signage
78404	373873	System Integrator for ERPs and n-tier web base...

78405 rows × 2 columns

In [17]:
```
#merge company specialty on previous merge
df_cis= pd.merge(df_ci, df_cs, on='company_id', how='inner')
```

In [18]:
```
df_cis
```

Out[18]:

	company_id	name	company_size	state	country	city	zip_code	industry	speciality
0	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cloud
1	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Mobile
2	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cognitive
3	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Security
4	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Research
...	...	...	...	...	...	...	...	...	...
89858	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Cross Training
89859	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Cardio Barre/HIIT Training
89860	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Aggie Owned and Operated
89861	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Highly Trained Teachers
89862	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Group Class, Individual Coaching

89863 rows × 9 columns

Import Employee Data

In [19]:
```
df_ec=pd.read_csv('employee_counts.csv')
```

In [20]:
```
df_ec
```

Out[20]:

	company_id	employee_count	follower_count	time_recorded
0	81149246	6	91	1.692645e+09
1	10033339	3	187	1.692645e+09
2	6049228	20	82	1.692645e+09
3	2641066	45	2336	1.692645e+09
4	96649998	0	2	1.692645e+09
...	...	...	...	...
14270	829374	2455	58753	1.699140e+09
14271	5574	8281	96163	1.699140e+09
14272	10135152	6322	31953	1.699140e+09
14273	373873	45	30298	1.699140e+09
14274	7428	6901	76462	1.699140e+09

14275 rows × 4 columns

Merge Company, Employee, Industry , Speciality Data to df1

In [21]:
```
df1= pd.merge(df_cis, df_ec, on='company_id', how='inner')
```

In [22]:
```
df1
```

Out[22]:

	company_id	name	company_size	state	country	city	zip_code	industry	speciality	employee_count	follower_count
0	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cloud	316130	16114398
1	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cloud	308001	15467710
2	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Mobile	316130	16114398
3	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Mobile	308001	15467710
4	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cognitive	316130	16114398
...	...	...	...	...	...	...	...	...	...	...	...
131432	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Cross Training	0	2
131433	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Cardio Barre/HIIT Training	0	2
131434	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Aggie Owned and Operated	0	2
131435	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Highly Trained Teachers	0	2
131436	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Group Class, Individual Coaching	0	2

131437 rows × 12 columns

In [23]:
```
df1.drop(columns=['follower_count'], inplace=True)
```

In [24]:
```
df1
```

Out[24]:

	company_id	name	company_size	state	country	city	zip_code	industry	speciality	employee_count	time_recorded
0	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cloud	316130	1.692851e+09
1	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cloud	308001	1.698964e+09
2	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Mobile	316130	1.692851e+09
3	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Mobile	308001	1.698964e+09
4	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cognitive	316130	1.692851e+09
...	...	...	...	...	...	...	...	...	...	...	...
131432	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Cross Training	0	1.699139e+09
131433	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Cardio Barre/HIIT Training	0	1.699139e+09
131434	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Aggie Owned and Operated	0	1.699139e+09
131435	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Highly Trained Teachers	0	1.699139e+09
131436	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Group Class, Individual Coaching	0	1.699139e+09

131437 rows × 11 columns

In [25]:
```
from datetime import datetime
#change format of date notations to actual date
df1['time_recorded'] = df1['time_recorded'].apply(lambda x: datetime.utcfromtimestamp(x).strftime('%d %B, %Y'))
```

In [26]:
```
df1
```

Out[26]:

	company_id	name	company_size	state	country	city	zip_code	industry	speciality	employee_count	time_recorded
0	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cloud	316130	24 August, 2023
1	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cloud	308001	02 November, 2023
2	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Mobile	316130	24 August, 2023
3	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Mobile	308001	02 November, 2023
4	1009	IBM	7.0	NY	US	Armonk, New York	10504	Information Technology & Services	Cognitive	316130	24 August, 2023
...	...	...	...	...	...	...	...	...	...	...	...
131432	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Cross Training	0	04 November, 2023
131433	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Cardio Barre/HIIT Training	0	04 November, 2023
131434	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Aggie Owned and Operated	0	04 November, 2023
131435	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Highly Trained Teachers	0	04 November, 2023
131436	101059464	Pure Barre College Station	1.0	Texas	US	College Station	77845	Health Care	Group Class, Individual Coaching	0	04 November, 2023

131437 rows × 11 columns

Import Job Industry

In [27]:
```
df_j=pd.read_csv('job_industries.csv')
```

In [28]:
```
df_j
```

Out[28]:

	job_id	industry_id
0	3378133231	68
1	3497509795	96
2	3690843087	47
3	3691775263	112
4	3691779379	80
...	...	...
44086	3757486249	47
44087	3757486249	43
44088	3757780487	104
44089	3757934256	80
44090	3757498232	14

44091 rows × 2 columns

Import Benefit Data

In [29]:
```
df_b=pd.read_csv('benefits.csv')
```

In [30]:
```
df_b
```

Out[30]:

	job_id	inferred	type
0	3690843087	0	Medical insurance
1	3690843087	0	Dental insurance
2	3690843087	0	401(k)
3	3690843087	0	Paid maternity leave
4	3690843087	0	Disability insurance
...	...	...	...
29320	3757934256	0	401(k)
29321	3757934256	0	Paid paternity leave
29322	3757934256	0	Paid maternity leave
29323	3757934256	0	Disability insurance
29324	3757498232	1	Medical insurance

29325 rows × 3 columns

In [31]:
```
#merge benefits on job industry
df_m= pd.merge(df_j, df_b, on='job_id', how='inner')
```

In [32]:
```
df_m
```

Out[32]:

	job_id	industry_id	inferred	type
0	3690843087	47	0	Medical insurance
1	3690843087	47	0	Dental insurance
2	3690843087	47	0	401(k)
3	3690843087	47	0	Paid maternity leave
4	3690843087	47	0	Disability insurance
...	...	...	...	...
39915	3757934256	80	0	401(k)
39916	3757934256	80	0	Paid paternity leave
39917	3757934256	80	0	Paid maternity leave
39918	3757934256	80	0	Disability insurance
39919	3757498232	14	1	Medical insurance

39920 rows × 4 columns

Import Job Skill Data

In [33]:
```
df_js=pd.read_csv('job_skills.csv')
```

In [34]:
```
df_js
```

Out[34]:

	job_id	skill_abr
0	3690843087	ACCT
1	3690843087	FIN
2	3691763971	MGMT
3	3691763971	MNFC
4	3691775263	MGMT
...	...	...
56586	3757780487	HCPR
56587	3757934256	DSGN
56588	3757934256	ART
56589	3757934256	IT
56590	3757498232	ADM

56591 rows × 2 columns

In [35]:
```
#merge job data on job skill
df_m= pd.merge(df_m, df_js, on='job_id', how='inner')
```

In [36]:
```
df_m
```

Out[36]:

	job_id	industry_id	inferred	type	skill_abr
0	3690843087	47	0	Medical insurance	ACCT
1	3690843087	47	0	Medical insurance	FIN
2	3690843087	47	0	Dental insurance	ACCT
3	3690843087	47	0	Dental insurance	FIN
4	3690843087	47	0	401(k)	ACCT
...	...	...	...	...	...
74427	3757934256	80	0	Paid maternity leave	IT
74428	3757934256	80	0	Disability insurance	DSGN
74429	3757934256	80	0	Disability insurance	ART
74430	3757934256	80	0	Disability insurance	IT
74431	3757498232	14	1	Medical insurance	ADM

74432 rows × 5 columns

Import Salary Data

In [37]:
```
df_s=pd.read_csv('salaries.csv')
```

In [38]:
```
df_s
```

Out[38]:

	salary_id	job_id	max_salary	med_salary	min_salary	pay_period	currency	compensation_type
0	1	3378133231	30.0	NaN	22.0	HOURLY	USD	BASE_SALARY
1	2	3690843087	65000.0	NaN	55000.0	YEARLY	USD	BASE_SALARY
2	3	3691794313	22.0	NaN	19.0	HOURLY	USD	BASE_SALARY
3	4	3691795389	70000.0	NaN	68000.0	YEARLY	USD	BASE_SALARY
4	5	3691797089	22.0	NaN	18.0	HOURLY	USD	BASE_SALARY
...	...	...	...	...	...	...	...	...
13347	13736	3756116147	210000.0	NaN	170000.0	YEARLY	USD	BASE_SALARY
13348	13737	3757723474	130000.0	NaN	110000.0	YEARLY	USD	BASE_SALARY
13349	13738	3757485745	NaN	48.0	NaN	HOURLY	USD	BASE_SALARY
13350	13739	3757490898	60000.0	NaN	45000.0	YEARLY	USD	BASE_SALARY
13351	13740	3757486249	115000.0	NaN	100000.0	YEARLY	USD	BASE_SALARY

13352 rows × 8 columns

In [39]:
```
#merge salary on previous job merge
df_m= pd.merge(df_m, df_s, on='job_id', how='inner')
```

In [40]:
```
df_m
```

Out[40]:

	job_id	industry_id	inferred	type	skill_abr	salary_id	max_salary	med_salary	min_salary	pay_period	currency	com
0	3690843087	47	0	Medical insurance	ACCT	2	65000.0	NaN	55000.0	YEARLY	USD	
1	3690843087	47	0	Medical insurance	FIN	2	65000.0	NaN	55000.0	YEARLY	USD	
2	3690843087	47	0	Dental insurance	ACCT	2	65000.0	NaN	55000.0	YEARLY	USD	
3	3690843087	47	0	Dental insurance	FIN	2	65000.0	NaN	55000.0	YEARLY	USD	
4	3690843087	47	0	401(k)	ACCT	2	65000.0	NaN	55000.0	YEARLY	USD	
...	...	...	...	...	...	...	...	...	...	...	...	...
40370	3757486249	43	0	Tuition assistance	ACCT	13740	115000.0	NaN	100000.0	YEARLY	USD	
40371	3757486249	43	0	Student loan assistance	CNSL	13740	115000.0	NaN	100000.0	YEARLY	USD	
40372	3757486249	43	0	Student loan assistance	ACCT	13740	115000.0	NaN	100000.0	YEARLY	USD	
40373	3757486249	43	0	Disability insurance	CNSL	13740	115000.0	NaN	100000.0	YEARLY	USD	
40374	3757486249	43	0	Disability insurance	ACCT	13740	115000.0	NaN	100000.0	YEARLY	USD	

40375 rows × 12 columns

Import Industry Data

In [41]:
```
df_i=pd.read_csv('industries.csv')
```

In [42]:
```
df_i
```

Out[42]:

	industry_id	industry_name
0	1	Defense and Space Manufacturing
1	3	Computer Hardware Manufacturing
2	4	Software Development
3	5	Computer Networking Products
4	6	Technology, Information and Internet
...	...	...
224	3240	Renewable Energy Power Generation
225	3241	Renewable Energy Equipment Manufacturing
226	3242	Engineering Services
227	3243	Services for Renewable Energy
228	3251	Climate Technology Product Manufacturing

229 rows × 2 columns

In [ ]:

In [43]:
```
#merge industry data on previous merge
df_m= pd.merge(df_m, df_i, on='industry_id', how='inner')
```

In [44]:
```
df_m
```

Out[44]:

	job_id	industry_id	inferred	type	skill_abr	salary_id	max_salary	med_salary	min_salary	pay_period	currency	com
0	3690843087	47	0	Medical insurance	ACCT	2	65000.00	NaN	55000.0	YEARLY	USD	
1	3690843087	47	0	Medical insurance	FIN	2	65000.00	NaN	55000.0	YEARLY	USD	
2	3690843087	47	0	Dental insurance	ACCT	2	65000.00	NaN	55000.0	YEARLY	USD	
3	3690843087	47	0	Dental insurance	FIN	2	65000.00	NaN	55000.0	YEARLY	USD	
4	3690843087	47	0	401(k)	ACCT	2	65000.00	NaN	55000.0	YEARLY	USD	
...	...	...	...	...	...	...	...	...	...	...	...	...
40370	3757480049	3195	1	401(k)	ADM	13018	NaN	12.0	NaN	HOURLY	USD	
40371	3757496520	1779	1	401(k)	ENG	13149	26.72	NaN	22.5	HOURLY	USD	
40372	3757496520	1779	1	401(k)	IT	13149	26.72	NaN	22.5	HOURLY	USD	
40373	3757496520	1779	1	Disability insurance	ENG	13149	26.72	NaN	22.5	HOURLY	USD	
40374	3757496520	1779	1	Disability insurance	IT	13149	26.72	NaN	22.5	HOURLY	USD	

40375 rows × 13 columns

Import Skills Data

In [45]:
```
df_sk=pd.read_csv('skills.csv')
```

In [46]:
```
df_sk
```

Out[46]:

	skill_abr	skill_name
0	PRCH	Purchasing
1	SUPL	Supply Chain
2	PR	Public Relations
3	SCI	Science
4	STRA	Strategy/Planning
5	WRT	Writing/Editing
6	QA	Quality Assurance
7	DIST	Distribution
8	PROD	Production
9	PRJM	Project Management
10	ADVR	Advertising
11	RSCH	Research
12	HR	Human Resources
13	LGL	Legal
14	PRDM	Product Management
15	MRKT	Marketing
16	EDU	Education
17	TRNG	Training
18	ANLS	Analyst
19	FIN	Finance
20	SALE	Sales
21	BD	Business Development
22	MGMT	Management
23	MNFC	Manufacturing
24	GENB	General Business
25	CUST	Customer Service
26	ENG	Engineering
27	OTHR	Other
28	CNSL	Consulting
29	ACCT	Accounting/Auditing
30	HCPR	Health Care Provider
31	DSGN	Design
32	ART	Art/Creative
33	IT	Information Technology
34	ADM	Administrative

Merge Job industry, Benefits, Salary , Job skill Industry data to df2

In [47]:
```
df2= pd.merge(df_m, df_sk, on='skill_abr', how='inner')
```

In [48]:
```
df2
```

Out[48]:

	job_id	industry_id	inferred	type	skill_abr	salary_id	max_salary	med_salary	min_salary	pay_period	currency	com
0	3690843087	47	0	Medical insurance	ACCT	2	65000.0	NaN	55000.0	YEARLY	USD	
1	3690843087	47	0	Dental insurance	ACCT	2	65000.0	NaN	55000.0	YEARLY	USD	
2	3690843087	47	0	401(k)	ACCT	2	65000.0	NaN	55000.0	YEARLY	USD	
3	3690843087	47	0	Paid maternity leave	ACCT	2	65000.0	NaN	55000.0	YEARLY	USD	
4	3690843087	47	0	Disability insurance	ACCT	2	65000.0	NaN	55000.0	YEARLY	USD	
...	...	...	...	...	...	...	...	...	...	...	...	
40370	3749346362	140	0	Tuition assistance	ADVR	7799	56000.0	NaN	55000.0	YEARLY	USD	
40371	3749346362	140	0	Vision insurance	ADVR	7799	56000.0	NaN	55000.0	YEARLY	USD	
40372	3749346362	140	0	Dental insurance	ADVR	7799	56000.0	NaN	55000.0	YEARLY	USD	
40373	3749346362	140	0	401(k)	ADVR	7799	56000.0	NaN	55000.0	YEARLY	USD	
40374	3757496465	1862	1	401(k)	ADVR	12276	190000.0	NaN	170000.0	YEARLY	USD	

40375 rows × 14 columns

Import Postings Data

In [49]:
```
df_jp=pd.read_csv('job_postings.csv')
```

In [50]:
```
df_jp
```

Out[50]:

	job_id	company_id	title	description	max_salary	med_salary	min_salary	pay_period	formatted_work
0	3757940104	553718.0	Hearing Care Provider	Overview\n\nHearingLife is a national hearing ...	NaN	5250.00	NaN	MONTHLY	Ful
1	3757940025	2192142.0	Shipping & Receiving Associate 2nd shift (Beav...	Metalcraft of Mayville\nMetalcraft of Mayville...	NaN	NaN	NaN	NaN	Ful
2	3757938019	474443.0	Manager, Engineering	\nThe TSUBAKI name is synonymous with excellen...	NaN	NaN	NaN	NaN	Ful
3	3757938018	18213359.0	Cook	descriptionTitle\n\n Looking for a great oppor...	NaN	22.27	NaN	HOURLY	Ful
4	3757937095	437225.0	Principal Cloud Security Architect (Remote)	Job Summary\nAt iHerb, we are on a mission to ...	275834.0	NaN	205956.0	YEARLY	Ful
...	...	...	...	...	...	...	...	...	
33241	133114754	77766802.0	Sales Manager	Are you a dynamic and creative marketing profe...	NaN	NaN	NaN	NaN	Ful
33242	108965123	NaN	Office Administrative Assistant	A fast-fashion wholesaler, is looking for a fu...	NaN	NaN	NaN	NaN	Ful
33243	102339515	52132271.0	Franchise Owner	DuctVentz is a dryer and A/C – heat vent clean...	NaN	NaN	NaN	NaN	Ful
33244	85008768	NaN	Licensed Insurance Agent	While many industries were hurt by the last fe...	52000.0	NaN	45760.0	YEARLY	Ful
33245	3958427	630152.0	Stylist/ Clorist	Karen Marie is looking for an awesome experien...	80000.0	NaN	35000.0	YEARLY	Ful

33246 rows × 28 columns

In [51]:
```
df3= pd.merge(df_jp, df2, on='job_id', how='inner')
```

In [52]:
```
df3
```

Out[52]:

	job_id	company_id	title	description	max_salary_x	med_salary_x	min_salary_x	pay_period_x	formatted_v
0	3757940104	553718.0	Hearing Care Provider	Overview\n\nHearingLife is a national hearing ...	NaN	5250.0	NaN	MONTHLY	
1	3757940104	553718.0	Hearing Care Provider	Overview\n\nHearingLife is a national hearing ...	NaN	5250.0	NaN	MONTHLY	
2	3757935381	19181907.0	Insights Analyst - Auto Industry	Who We Are\n\nEscalent is an award-winning dat...	64000.0	NaN	58000.0	YEARLY	
3	3757935381	19181907.0	Insights Analyst - Auto Industry	Who We Are\n\nEscalent is an award-winning dat...	64000.0	NaN	58000.0	YEARLY	
4	3757935381	19181907.0	Insights Analyst - Auto Industry	Who We Are\n\nEscalent is an award-winning dat...	64000.0	NaN	58000.0	YEARLY	
...	...	...	...	...	...	...	...	...	
40370	85008768	NaN	Licensed Insurance Agent	While many industries were hurt by the last fe...	52000.0	NaN	45760.0	YEARLY	
40371	85008768	NaN	Licensed Insurance Agent	While many industries were hurt by the last fe...	52000.0	NaN	45760.0	YEARLY	
40372	3958427	630152.0	Stylist/ Clorist	Karen Marie is looking for an awesome experien...	80000.0	NaN	35000.0	YEARLY	
40373	3958427	630152.0	Stylist/ Clorist	Karen Marie is looking for an awesome experien...	80000.0	NaN	35000.0	YEARLY	
40374	3958427	630152.0	Stylist/ Clorist	Karen Marie is looking for an awesome experien...	80000.0	NaN	35000.0	YEARLY	

40375 rows × 41 columns

In [53]:
```
from datetime import datetime

#change notation to actual date
df3['original_listed_time'] = df3['original_listed_time'].apply(lambda x: datetime.utcfromtimestamp(x / 1000).strf
```

In [54]:
```
df3['original_listed_time']
```

Out[54]:
```
0        2023-11-04
1        2023-11-04
2        2023-11-02
3        2023-11-02
4        2023-11-02
            ...
40370    2023-08-23
40371    2023-08-23
40372    2023-11-03
40373    2023-11-03
40374    2023-11-03
Name: original_listed_time, Length: 40375, dtype: object
```

In [55]:
```
df3['pay_period_x']
```

Out[55]:
```
0        MONTHLY
1        MONTHLY
2         YEARLY
3         YEARLY
4         YEARLY
          ...
40370     YEARLY
40371     YEARLY
40372     YEARLY
40373     YEARLY
40374     YEARLY
Name: pay_period_x, Length: 40375, dtype: object
```

In [56]:
```
df3.columns
```

Out[56]:
```
Index(['job_id', 'company_id', 'title', 'description', 'max_salary_x',
       'med_salary_x', 'min_salary_x', 'pay_period_x', 'formatted_work_type',
       'location', 'applies', 'original_listed_time', 'remote_allowed',
       'views', 'job_posting_url', 'application_url', 'application_type',
       'expiry', 'closed_time', 'formatted_experience_level', 'skills_desc',
       'listed_time', 'posting_domain', 'sponsored', 'work_type', 'currency_x',
       'compensation_type_x', 'scraped', 'industry_id', 'inferred', 'type',
       'skill_abr', 'salary_id', 'max_salary_y', 'med_salary_y',
       'min_salary_y', 'pay_period_y', 'currency_y', 'compensation_type_y',
       'industry_name', 'skill_name'],
      dtype='object')
```

In [57]:
```
#drop inwanted columns
df3=df3.drop(columns=['max_salary_y', 'med_salary_y','min_salary_y', 'pay_period_y','job_posting_url', 'applicatio
```

In [58]:
```
df3.columns
```

Out[58]:
```
Index(['job_id', 'company_id', 'title', 'description', 'max_salary_x',
       'med_salary_x', 'min_salary_x', 'pay_period_x', 'formatted_work_type',
       'location', 'applies', 'original_listed_time', 'remote_allowed',
       'views', 'application_type', 'expiry', 'formatted_experience_level',
       'skills_desc', 'sponsored', 'work_type', 'industry_id', 'type',
       'skill_abr', 'salary_id', 'currency_y', 'industry_name', 'skill_name'],
      dtype='object')
```

In [59]:
```
df3
```

Out[59]:

	job_id	company_id	title	description	max_salary_x	med_salary_x	min_salary_x	pay_period_x	formatted_v
0	3757940104	553718.0	Hearing Care Provider	Overview\n\nHearingLife is a national hearing ...	NaN	5250.0	NaN	MONTHLY	
1	3757940104	553718.0	Hearing Care Provider	Overview\n\nHearingLife is a national hearing ...	NaN	5250.0	NaN	MONTHLY	
2	3757935381	19181907.0	Insights Analyst - Auto Industry	Who We Are\n\nEscalent is an award-winning dat...	64000.0	NaN	58000.0	YEARLY	
3	3757935381	19181907.0	Insights Analyst - Auto Industry	Who We Are\n\nEscalent is an award-winning dat...	64000.0	NaN	58000.0	YEARLY	
4	3757935381	19181907.0	Insights Analyst - Auto Industry	Who We Are\n\nEscalent is an award-winning dat...	64000.0	NaN	58000.0	YEARLY	
...	...	...	...	...	...	...	...	...	
40370	85008768	NaN	Licensed Insurance Agent	While many industries were hurt by the last fe...	52000.0	NaN	45760.0	YEARLY	
40371	85008768	NaN	Licensed Insurance Agent	While many industries were hurt by the last fe...	52000.0	NaN	45760.0	YEARLY	
40372	3958427	630152.0	Stylist/ Clorist	Karen Marie is looking for an awesome experien...	80000.0	NaN	35000.0	YEARLY	
40373	3958427	630152.0	Stylist/ Clorist	Karen Marie is looking for an awesome experien...	80000.0	NaN	35000.0	YEARLY	
40374	3958427	630152.0	Stylist/ Clorist	Karen Marie is looking for an awesome experien...	80000.0	NaN	35000.0	YEARLY	

40375 rows × 27 columns

In [60]:
```
#final merge between fist df and new df
final_df= pd.merge(df3, df1, on='company_id', how='inner')
```

In [61]:
```
final_df.head(5)
```

Out[61]:

	job_id	company_id	title	description	max_salary_x	med_salary_x	min_salary_x	pay_period_x	formatted_work_t
0	3757940104	553718.0	Hearing Care Provider	Overview\n\nHearingLife is a national hearing ...	NaN	5250.0	NaN	MONTHLY	Full-t
1	3757940104	553718.0	Hearing Care Provider	Overview\n\nHearingLife is a national hearing ...	NaN	5250.0	NaN	MONTHLY	Full-t
2	3757940104	553718.0	Hearing Care Provider	Overview\n\nHearingLife is a national hearing ...	NaN	5250.0	NaN	MONTHLY	Full-t
3	3757940104	553718.0	Hearing Care Provider	Overview\n\nHearingLife is a national hearing ...	NaN	5250.0	NaN	MONTHLY	Full-t
4	3757940104	553718.0	Hearing Care Provider	Overview\n\nHearingLife is a national hearing ...	NaN	5250.0	NaN	MONTHLY	Full-t

5 rows × 37 columns

In [62]:
```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1107608 entries, 0 to 1107607
Data columns (total 37 columns):
 #   Column                      Non-Null Count    Dtype
---  ------                      --------------    -----
 0   job_id                      1107608 non-null  int64
 1   company_id                  1107608 non-null  float64
 2   title                       1107608 non-null  object
 3   description                 1107608 non-null  object
 4   max_salary_x                1017411 non-null  float64
 5   med_salary_x                90197 non-null    float64
 6   min_salary_x                1017411 non-null  float64
 7   pay_period_x                1107608 non-null  object
 8   formatted_work_type         1107608 non-null  object
 9   location                    1107608 non-null  object
 10  applies                     765426 non-null   float64
 11  original_listed_time        1107608 non-null  object
 12  remote_allowed              160246 non-null   float64
 13  views                       1004185 non-null  float64
 14  application_type            1107608 non-null  object
 15  expiry                      1107608 non-null  float64
 16  formatted_experience_level  974006 non-null   object
 17  skills_desc                 4516 non-null     object
 18  sponsored                   1107608 non-null  int64
 19  work_type                   1107608 non-null  object
 20  industry_id                 1107608 non-null  int64
 21  type                        1107608 non-null  object
 22  skill_abr                   1107608 non-null  object
 23  salary_id                   1107608 non-null  int64
 24  currency_y                  1107608 non-null  object
 25  industry_name               1107392 non-null  object
 26  skill_name                  1107608 non-null  object
 27  name                        1107608 non-null  object
 28  company_size                1079306 non-null  float64
 29  state                       1106798 non-null  object
 30  country                     1107608 non-null  object
 31  city                        1107608 non-null  object
 32  zip_code                    1105641 non-null  object
 33  industry                    1107608 non-null  object
 34  speciality                  1107608 non-null  object
 35  employee_count              1107608 non-null  int64
 36  time_recorded               1107608 non-null  object
dtypes: float64(9), int64(5), object(23)
memory usage: 312.7+ MB
```

In [63]:
```
#further drop of unwanted columns
final_df=final_df.drop(columns=['time_recorded','expiry','industry_name','work_type','location','description','ski
```

In [64]:
```
final_df
```

Out[64]:

	job_id	company_id	title	max_salary_x	med_salary_x	min_salary_x	pay_period_x	formatted_work_type	applies	orig
0	3757940104	553718.0	Hearing Care Provider	NaN	5250.0	NaN	MONTHLY	Full-time	NaN	
1	3757940104	553718.0	Hearing Care Provider	NaN	5250.0	NaN	MONTHLY	Full-time	NaN	
2	3757940104	553718.0	Hearing Care Provider	NaN	5250.0	NaN	MONTHLY	Full-time	NaN	
3	3757940104	553718.0	Hearing Care Provider	NaN	5250.0	NaN	MONTHLY	Full-time	NaN	
4	3757940104	553718.0	Hearing Care Provider	NaN	5250.0	NaN	MONTHLY	Full-time	NaN	
...	...	...	...	...	...	...	...	...	...	
1107603	3958427	630152.0	Stylist/ Clorist	80000.0	NaN	35000.0	YEARLY	Full-time	NaN	
1107604	3958427	630152.0	Stylist/ Clorist	80000.0	NaN	35000.0	YEARLY	Full-time	NaN	
1107605	3958427	630152.0	Stylist/ Clorist	80000.0	NaN	35000.0	YEARLY	Full-time	NaN	
1107606	3958427	630152.0	Stylist/ Clorist	80000.0	NaN	35000.0	YEARLY	Full-time	NaN	
1107607	3958427	630152.0	Stylist/ Clorist	80000.0	NaN	35000.0	YEARLY	Full-time	NaN	

1107608 rows × 26 columns

In [65]:
```
#rearrange columns
final_df=final_df[['job_id', 'company_id', 'name', 'title', 'industry','skill_name', 'type', 'max_salary_x','med_s
                   'application_type', 'formatted_experience_level','sponsored', 'currency_y','company_size', 'sta
```

In [66]:
```
final_df
```

Out[66]:

	job_id	company_id	name	title	industry	skill_name	type	max_salary_x	med_salary_x	min_salary_x	...
0	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	NaN	5250.0	NaN	...
1	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	NaN	5250.0	NaN	...
2	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	NaN	5250.0	NaN	...
3	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	NaN	5250.0	NaN	...
4	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	NaN	5250.0	NaN	...
...	...	...	...	...	...	...	...	...	...	...	...
1107603	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	NaN	35000.0	...
1107604	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	NaN	35000.0	...
1107605	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	NaN	35000.0	...
1107606	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	NaN	35000.0	...
1107607	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	NaN	35000.0	...

1107608 rows × 26 columns

In [67]:
```
final_df.isnull().sum()
```

Out[67]:
```
job_id                        0
company_id                    0
name                          0
title                         0
industry                      0
skill_name                    0
type                          0
max_salary_x              90197
med_salary_x            1017411
min_salary_x              90197
pay_period_x                  0
formatted_work_type           0
applies                  342182
original_listed_time          0
remote_allowed           947362
views                    103423
application_type              0
formatted_experience_level 133602
sponsored                     0
currency_y                    0
company_size              28302
state                       810
country                       0
city                          0
zip_code                   1967
employee_count                0
dtype: int64
```

In [68]:
```
final_df['application_type'].unique()
```

Out[68]:
```
array(['OffsiteApply', 'ComplexOnsiteApply', 'SimpleOnsiteApply'],
      dtype=object)
```

In [69]:
```
#change view count from notation to numbers
final_df['views'] = final_df['views'].apply(lambda x: '{:.0f}'.format(x))
```

In [70]:
```
#rename columns
final_df=final_df.rename(columns={'name':'Company_Name','type':'Benefits','formatted_work_type':'work_type','applic
                                  'max_salary_x':'max_salary', 'med_salary_x':'med_salary','min_salary_x':'min_sala
```

In [71]:
```
final_df
```

Out[71]:

	job_id	company_id	Company_Name	Job_Title	industry	skill_name	Benefits	max_salary	med_salary	min_salary	...
0	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	NaN	5250.0	NaN	...
1	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	NaN	5250.0	NaN	...
2	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	NaN	5250.0	NaN	...
3	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	NaN	5250.0	NaN	...
4	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	NaN	5250.0	NaN	...
...	...	...	...	...	...	...	...	...	...	...	...
1107603	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	NaN	35000.0	...
1107604	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	NaN	35000.0	...
1107605	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	NaN	35000.0	...
1107606	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	NaN	35000.0	...
1107607	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	NaN	35000.0	...

1107608 rows × 26 columns

In [72]:
```
final_df.isnull().sum()
```

Out[72]:
```
job_id                0
company_id            0
Company_Name          0
Job_Title             0
industry              0
skill_name            0
Benefits              0
max_salary        90197
med_salary      1017411
min_salary        90197
pay_period            0
work_type             0
No_of_Applicants  342182
Date_Posted           0
remote_allowed    947362
No_of_Views           0
application_type      0
Experience_level  133602
sponsored             0
Currency              0
company_size      28302
state               810
country               0
city                  0
zip_code           1967
employee_count        0
dtype: int64
```

In [73]:
```
final_df['max_salary'] = final_df['max_salary'].fillna(value=-1)
final_df['med_salary'] = final_df['med_salary'].fillna(value=-1)
final_df['min_salary'] = final_df['min_salary'].fillna(value=-1)
final_df['company_size'] = final_df['company_size'].fillna(value=-1)
final_df['remote_allowed'] = final_df['remote_allowed'].fillna(value=-1)
final_df['zip_code'] = final_df['zip_code'].fillna(value=-1)
final_df['No_of_Applicants'] = final_df['No_of_Applicants'].fillna(value=-1)
```

```
final_df['state'] = final_df['state'].fillna(value='Uknown')
final_df['Experience_level'] = final_df['Experience_level'].fillna(value='Uknown')
```

In [74]:
```
final_df.isnull().sum()
```

Out[74]:
```
job_id              0
company_id          0
Company_Name        0
Job_Title           0
industry            0
skill_name          0
Benefits            0
max_salary          0
med_salary          0
min_salary          0
pay_period          0
work_type           0
No_of_Applicants    0
Date_Posted         0
remote_allowed      0
No_of_Views         0
application_type    0
Experience_level    0
sponsored           0
Currency            0
company_size        0
state               0
country             0
city                0
zip_code            0
employee_count      0
dtype: int64
```

Exploratory Data Analysis

In [75]:
```
final_df
```

Out[75]:

	job_id	company_id	Company_Name	Job_Title	industry	skill_name	Benefits	max_salary	med_salary	min_salary	...
0	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	-1.0	5250.0	-1.0	...
1	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	-1.0	5250.0	-1.0	...
2	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	-1.0	5250.0	-1.0	...
3	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	-1.0	5250.0	-1.0	...
4	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	-1.0	5250.0	-1.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1107603	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	-1.0	35000.0	...
1107604	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	-1.0	35000.0	...
1107605	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	-1.0	35000.0	...
1107606	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	-1.0	35000.0	...
1107607	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	-1.0	35000.0	...

1107608 rows × 26 columns

In [76]:
```
import seaborn as nsns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

Top 10 Companies with the Most Job Postings

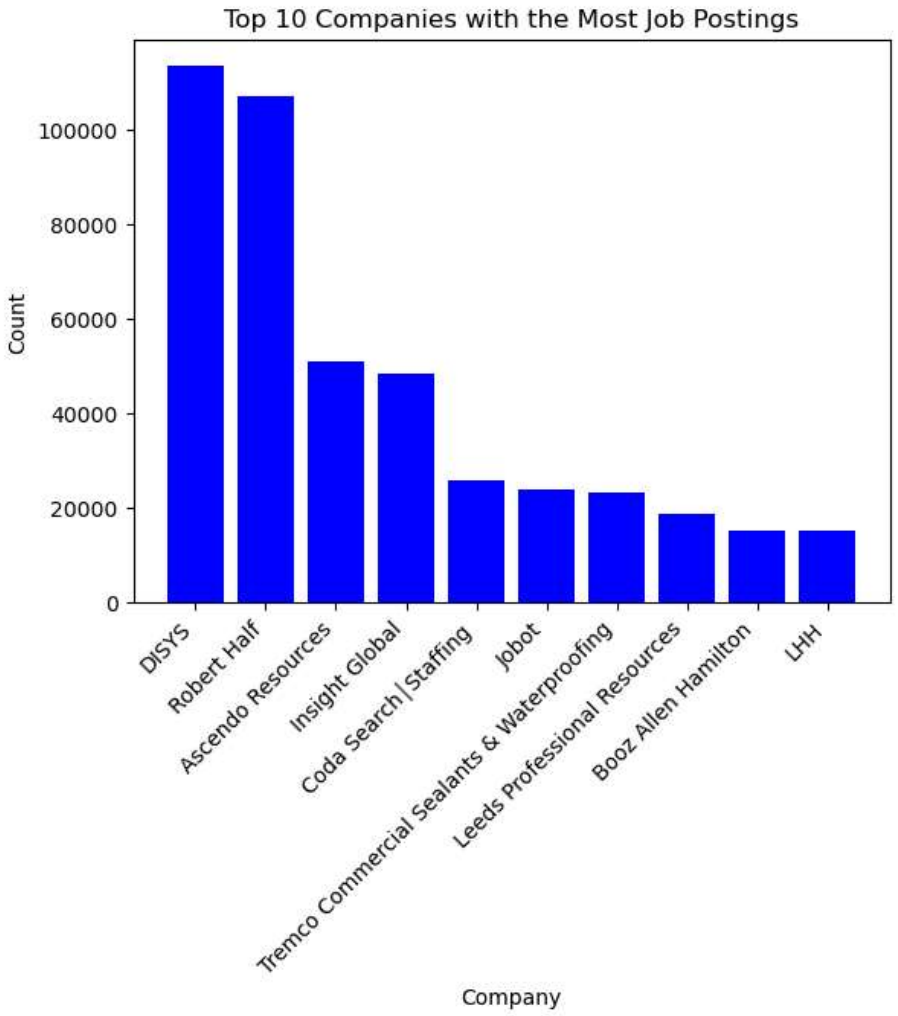
In [77]:
```
Company_posting_counts = final_df['Company_Name'].value_counts().sort_values(ascending=False)
print(Company_posting_counts.head(10))
```

```
Company_Name
DISYS                                    113400
Robert Half                              106932
Ascendo Resources                         50844
Insight Global                            48450
Coda Search|Staffing                      25920
Jobot                                     23936
Tremco Commercial Sealants & Waterproofing    23232
Leeds Professional Resources              18648
Booz Allen Hamilton                       15120
LHH                                       15050
Name: count, dtype: int64
```

In [78]:
```
plt.bar(Company_posting_counts.head(10).index,Company_posting_counts.head(10).values , color='blue')

plt.xlabel('Company')
plt.ylabel('Count')
plt.title('Top 10 Companies with the Most Job Postings')
plt.xticks(np.arange(len(Company_posting_counts.head(10).index)), rotation=45, ha='right')
plt.xticks(np.arange(len(Company_posting_counts.head(10).index)))

plt.show()
```



Top 10 Most Sought-After Positions

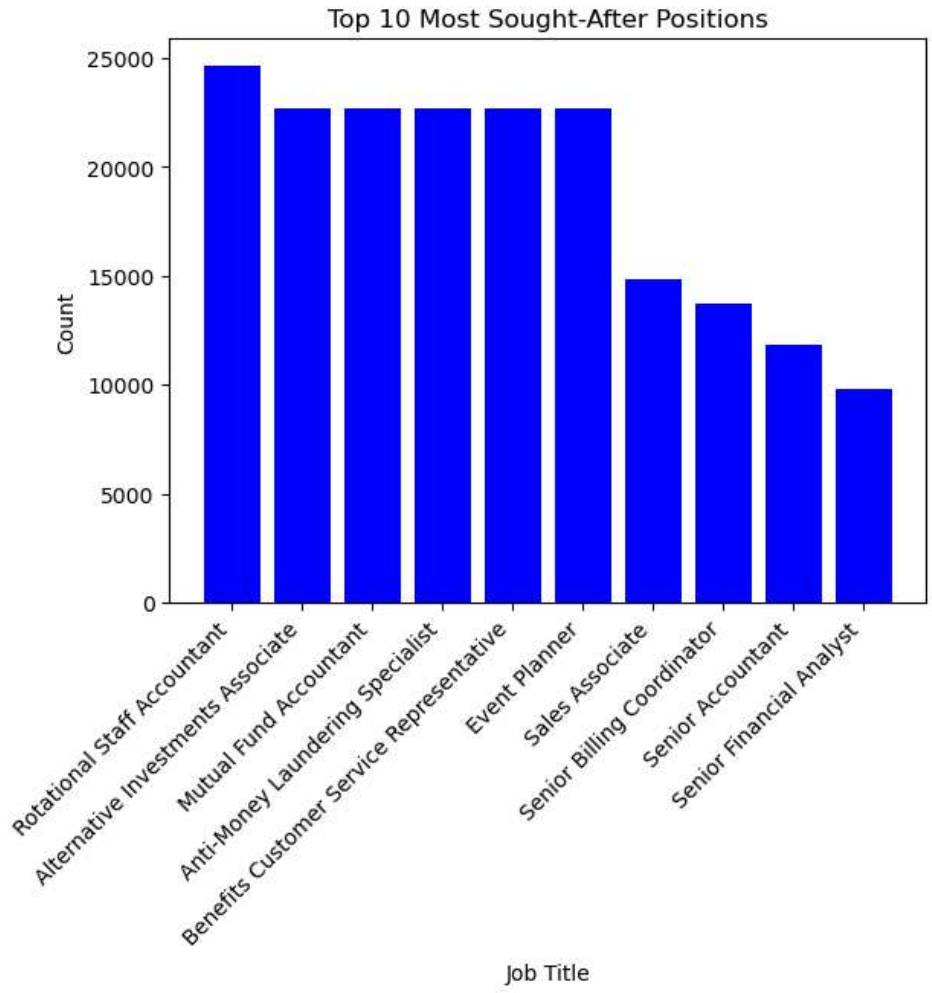
In [79]:
```
Job_Title_counts=final_df['Job_Title'].value_counts().sort_values(ascending=False)
Job_Title_counts.head(10)
```

Out[79]:
```
Job_Title
Rotational Staff Accountant                   24624
Alternative Investments Associate             22680
Mutual Fund Accountant                        22680
Anti-Money Laundering Specialist              22680
 Benefits Customer Service Representative     22680
Event Planner                                 22680
Sales Associate                               14848
Senior Billing Coordinator                    13680
Senior Accountant                             11815
Senior Financial Analyst                       9774
Name: count, dtype: int64
```

In [80]:
```
plt.bar(Job_Title_counts.head(10).index, Job_Title_counts.head(10).values, color='blue')

plt.xlabel('Job Title')
plt.ylabel('Count')
plt.title('Top 10 Most Sought-After Positions')
plt.xticks(np.arange(len(Job_Title_counts.head(10).index)), rotation=45, ha='right')
plt.xticks(np.arange(len(Job_Title_counts.head(10).index)))

plt.show()
```



In [112…
```
final_df['Job_Title'].unique
```

Out[112]:
```
<bound method Series.unique of 0          Hearing Care Provider
1          Hearing Care Provider
2          Hearing Care Provider
3          Hearing Care Provider
4          Hearing Care Provider
                  ...
1107603         Stylist/ Clorist
1107604         Stylist/ Clorist
1107605         Stylist/ Clorist
1107606         Stylist/ Clorist
1107607         Stylist/ Clorist
Name: Job_Title, Length: 1107608, dtype: object>
```

In [130…
```
analyst_jobs = final_df[final_df['Job_Title'].str.contains('Business analyst', case=False)]

# Print the job titles
print(analyst_jobs['Job_Title']).value_counts()
```

```
376184             Business Analyst, Senior
376185             Business Analyst, Senior
376186             Business Analyst, Senior
376187             Business Analyst, Senior
376188             Business Analyst, Senior
                        ...
1024836    Human Resources Business Analyst
1024837    Human Resources Business Analyst
1024838    Human Resources Business Analyst
1024839    Human Resources Business Analyst
1024840    Human Resources Business Analyst
Name: Job_Title, Length: 5406, dtype: object
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[130], line 4
      1 analyst_jobs = final_df[final_df['Job_Title'].str.contains('Business analyst', case=False)]
      3 # Print the job titles
----> 4 print(analyst_jobs['Job_Title']).value_counts()

AttributeError: 'NoneType' object has no attribute 'value_counts'
```

In [132…]:
```
industry_counts = final_df.groupby('industry')['No_of_Applicants'].sum().sort_values(ascending=False)
print(sum(industry_counts.head(10)))
```

```
11095934
```

Top 10 Industries with Job Openings

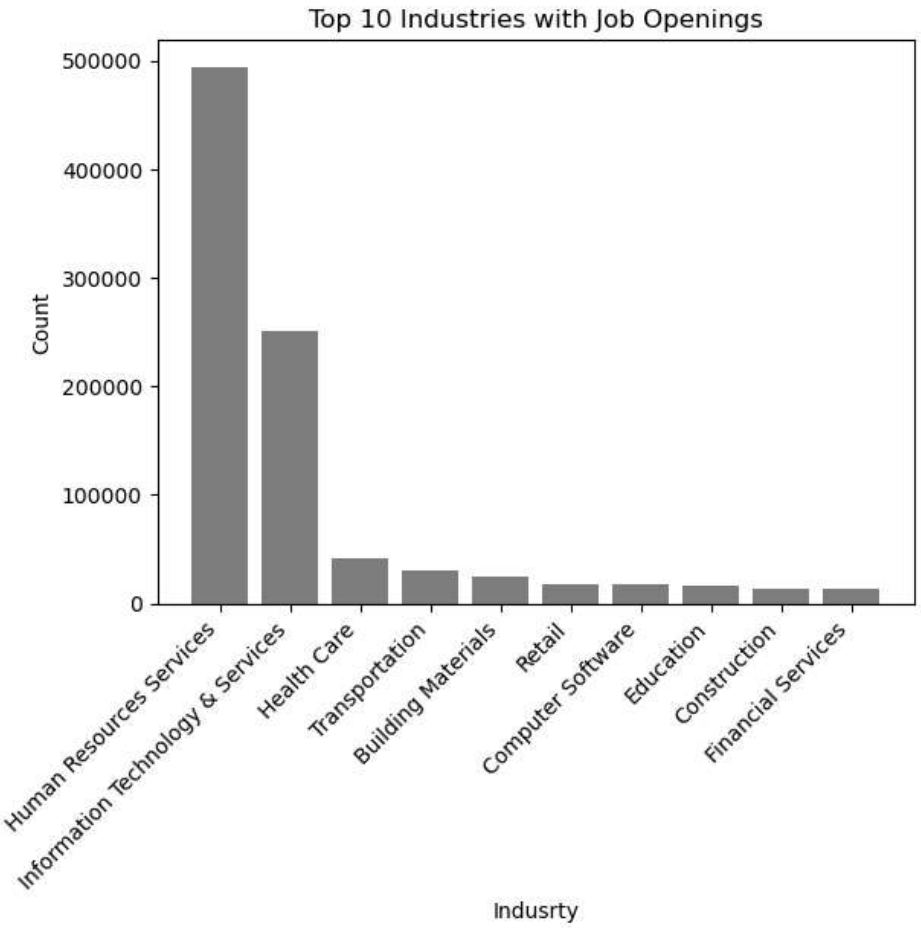
In [81]:
```
Industry_counts=final_df['industry'].value_counts().sort_values(ascending=False)
Industry_counts.head(10)
```

Out[81]:
```
industry
Human Resources Services             494216
Information Technology & Services    251491
Health Care                           41638
Transportation                        30479
Building Materials                    25276
Retail                                18025
Computer Software                     17197
Education                             16395
Construction                          13702
Financial Services                    13036
Name: count, dtype: int64
```

In [82]:
```
plt.bar(Industry_counts.head(10).index, Industry_counts.head(10).values , color='grey')

plt.xlabel('Indusrty')
plt.ylabel('Count')
plt.title('Top 10 Industries with Job Openings')
plt.xticks(np.arange(len(Industry_counts.head(10).index)), rotation=45, ha='right')
plt.xticks(np.arange(len(Industry_counts.head(10).index)))

plt.show()
```



Top 10 Industries With Most Applications

In [83]:
```
industry_counts = final_df.groupby('industry')['No_of_Applicants'].sum()

industry_counts.head(10)
```

Out[83]:
```
industry
 Manufacturing                608.0
Accounting                  28395.0
Advertising Services       253222.0
Airlines and Aviation         -48.0
Architecture                53294.0
Architecture and Planning     -80.0
Art                           336.0
Arts                          131.0
Automation                  39263.0
Aviation & Aerospace          836.0
Name: No_of_Applicants, dtype: float64
```

In [84]:
```
final_df['No_of_Applicants'] = final_df['No_of_Applicants'].astype(int)
```

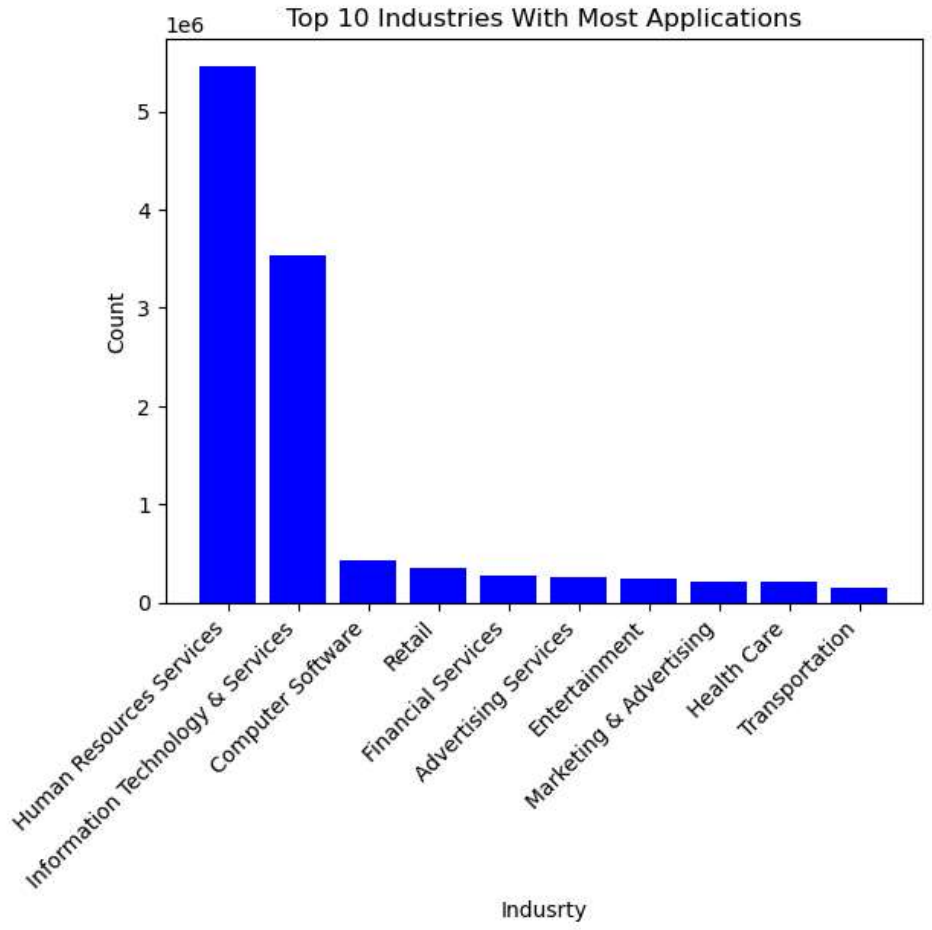
In [108…
```
industry_counts = final_df.groupby('industry')['No_of_Applicants'].sum().sort_values(ascending=False)
print(industry_counts.head(10))
```

```
industry
Human Resources Services             5465319
Information Technology & Services    3527657
Computer Software                     428750
Retail                                351082
Financial Services                    269261
Advertising Services                  253222
Entertainment                         242599
Marketing & Advertising               210039
Health Care                           204697
Transportation                        143308
Name: No_of_Applicants, dtype: int32
```

In [86]:
```
plt.bar(industry_counts.head(10).index, industry_counts.head(10).values, color='blue')

plt.xlabel('Indusrty')
plt.ylabel('Count')
plt.title('Top 10 Industries With Most Applications')
plt.xticks(np.arange(len(industry_counts.head(10).index)), rotation=45, ha='right')
plt.xticks(np.arange(len(industry_counts.head(10).index)))
```

```
plt.show()
```



Top Ten Job Opening Dates

In [87]:
```
Opening_date_counts=final_df['Date_Posted'].value_counts().sort_values(ascending=False)
Opening_date_counts.head(10)
```

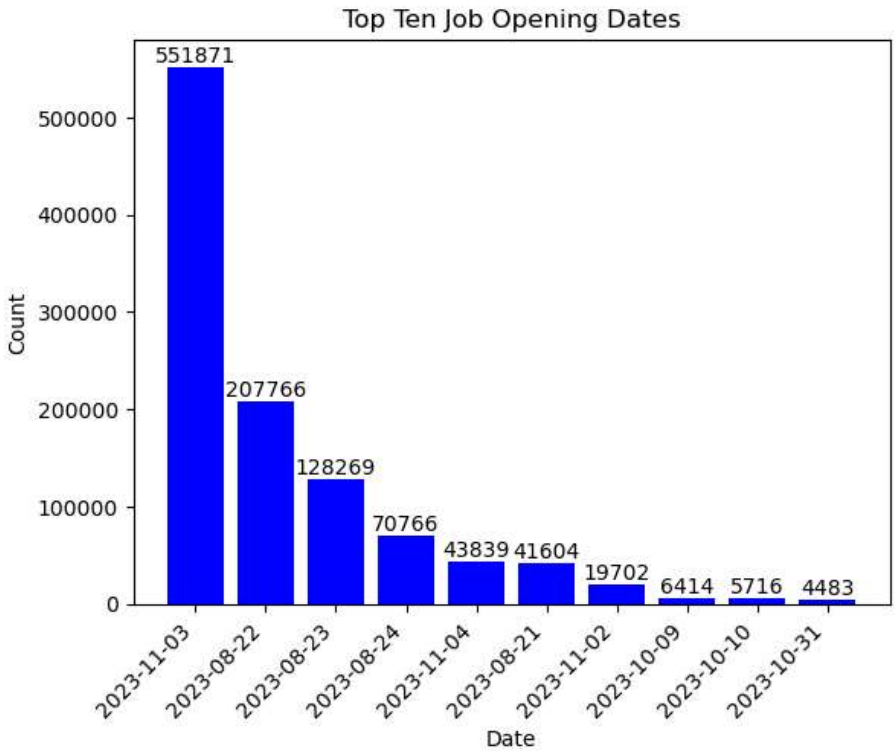
Out[87]:
```
Date_Posted
2023-11-03    551871
2023-08-22    207766
2023-08-23    128269
2023-08-24     70766
2023-11-04     43839
2023-08-21     41604
2023-11-02     19702
2023-10-09      6414
2023-10-10      5716
2023-10-31      4483
Name: count, dtype: int64
```

In [88]:
```
plt.bar(Opening_date_counts.head(10).index, Opening_date_counts.head(10).values, color='blue')

plt.xlabel('Date')
plt.ylabel('Count')
plt.title('Top Ten Job Opening Dates')
plt.xticks(np.arange(len(Opening_date_counts.head(10).index)), rotation=45, ha='right')
plt.xticks(np.arange(len(Opening_date_counts.head(10).index)))

for i, count in enumerate(Opening_date_counts.head(10).values):
    plt.text(i, count, str(count), ha='center', va='bottom')


plt.show()
```



In [89]:
```
final_df['Date_Posted'] = pd.to_datetime(final_df['Date_Posted'])

# Extract the months and count the occurrences
Opening_month_counts = final_df['Date_Posted'].dt.month.unique()

print(Opening_month_counts)
```

```
[11 10  8  7  9]
```

Industry Demand By Experience Level

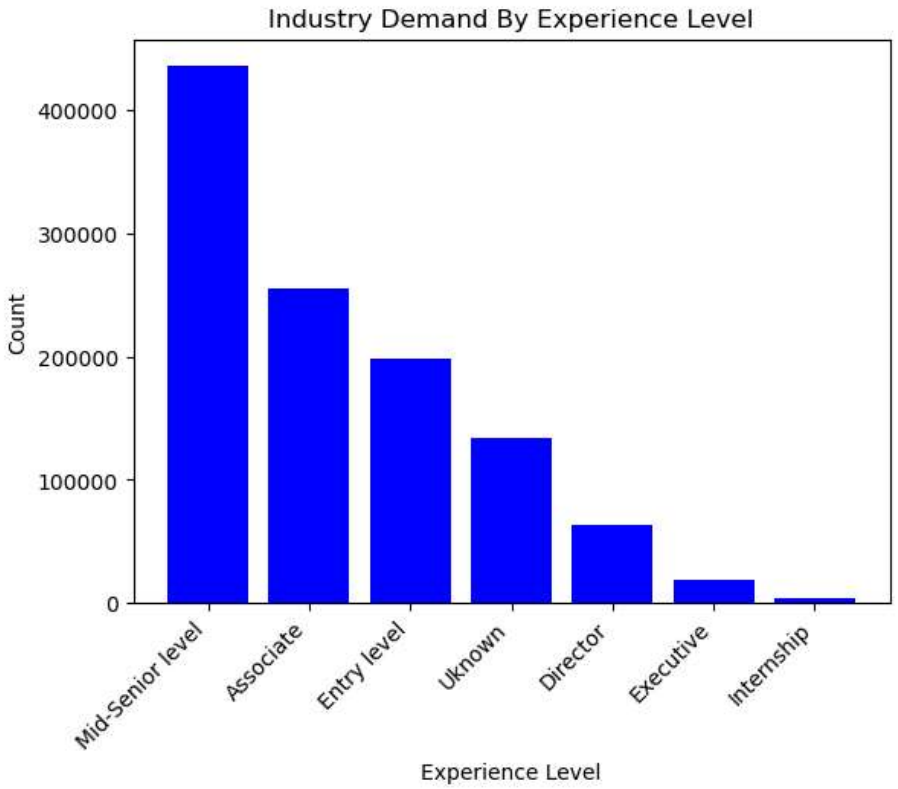
In [90]:
```
Experience_level_counts=final_df['Experience_level'].value_counts().sort_values(ascending=False)
Experience_level_counts
```

Out[90]:
```
Experience_level
Mid-Senior level    435425
Associate           254566
Entry level         197664
Uknown              133602
Director             63633
Executive            18399
Internship            4319
Name: count, dtype: int64
```

In [91]:
```
plt.bar(Experience_level_counts.index, Experience_level_counts.values, color='blue')

plt.xlabel('Experience Level')
plt.ylabel('Count')
plt.title('Industry Demand By Experience Level')
plt.xticks(np.arange(len(Experience_level_counts.index)), rotation=45, ha='right')
plt.xticks(np.arange(len(Experience_level_counts.index)))

plt.show()
```



In [92]:
```
applicant_Experience_counts = final_df.groupby('industry')['No_of_Applicants'].sum().sort_values(ascending=False)
print(industry_counts.head(10))
```

```
industry
Human Resources Services             5465319
Information Technology & Services    3527657
Computer Software                     428750
Retail                                351082
Financial Services                    269261
Advertising Services                  253222
Entertainment                         242599
Marketing & Advertising               210039
Health Care                           204697
Transportation                        143308
Name: No_of_Applicants, dtype: int32
```

In [131…
```
final_df
```

Out[131]:

	job_id	company_id	Company_Name	Job_Title	industry	skill_name	Benefits	max_salary	med_salary	min_salary	...
0	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	-1.0	5250.0	-1.0	...
1	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	-1.0	5250.0	-1.0	...
2	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	-1.0	5250.0	-1.0	...
3	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	-1.0	5250.0	-1.0	...
4	3757940104	553718.0	HearingLife	Hearing Care Provider	Retail	Other	Medical insurance	-1.0	5250.0	-1.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1107603	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	-1.0	35000.0	...
1107604	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	-1.0	35000.0	...
1107605	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	-1.0	35000.0	...
1107606	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	-1.0	35000.0	...
1107607	3958427	630152.0	Karen Marie Salon	Stylist/ Clorist	Consumer Services	Design	Medical insurance	80000.0	-1.0	35000.0	...

1107608 rows × 26 columns

In [104…

```
remote_counts=final_df['remote_allowed'].value_counts()
remote_counts.head(10)
```

Out[104]:

```
remote_allowed
-1.0    947362
 1.0    160246
Name: count, dtype: int64
```

In [106…

```
import plotly.graph_objects as go

plot_data=[
    go.Pie(
        labels=("Onsite","Remote"),
        values=final_df['remote_allowed'].value_counts(),
        marker=dict(colors=["Blue","Green"],
                    line=dict(color="white",
                              width=1.5)),

        rotation=90,
        hoverinfo= 'label+value+text',
        hole=.6)
]


plot_layout = go.Layout(dict(title='Work Location'))

fig = go.Figure(data=plot_data, layout=plot_layout)


fig.show()
```

Job Openings By Country

In [94]:
```
Country_counts=final_df['country'].value_counts().sort_values(ascending=False)
Country_counts.head(10)
```

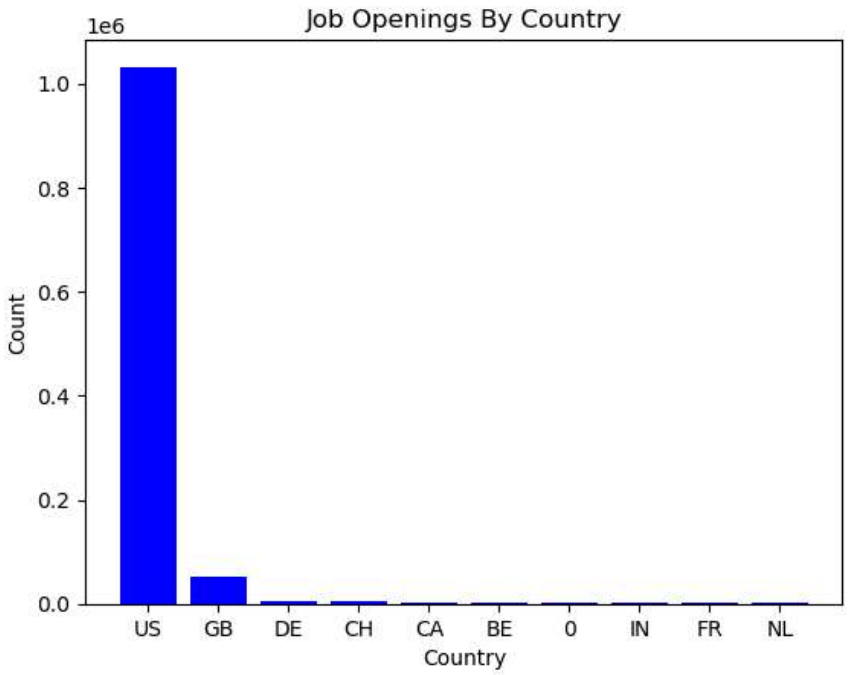
Out[94]:
```
country
US    1031929
GB      51467
DE       4085
CH       3999
CA       3478
BE       2840
0        2061
IN       1976
FR       1433
NL        698
Name: count, dtype: int64
```

In [95]:
```
plt.bar(Country_counts.head(10).index, Country_counts.head(10).values, color='blue')

plt.xlabel('Country')
plt.ylabel('Count')
plt.title('Job Openings By Country')

plt.xticks(np.arange(len(Country_counts.head(10).index)))

plt.show()
```



Distribution of Company Sizes

In [96]:
```
company_size_counts=final_df['company_size'].value_counts().sort_values(ascending=False)
company_size_counts.head(10)
```

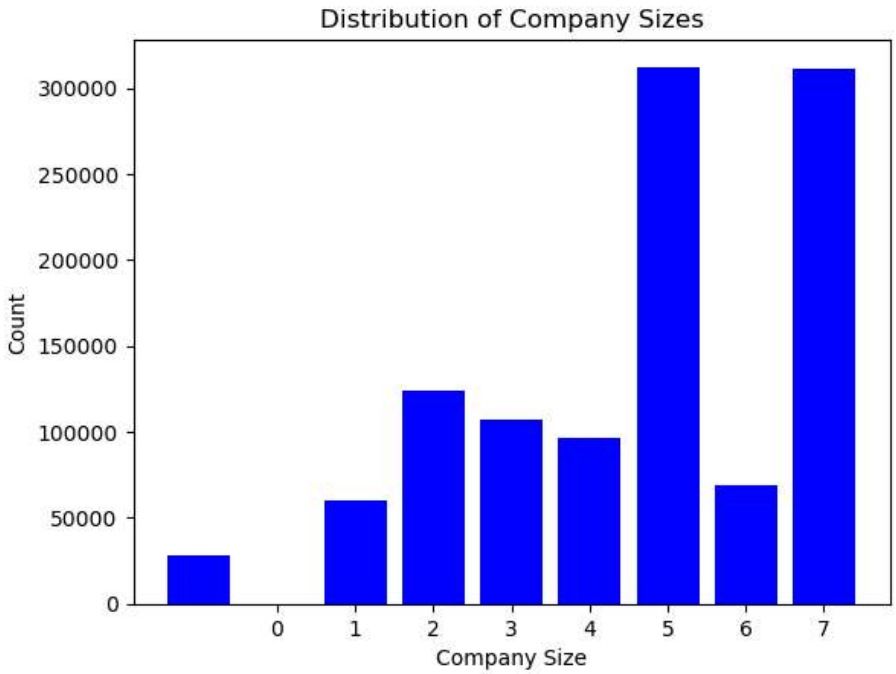
Out[96]:
```
company_size
 5.0    312306
 7.0    310869
 2.0    123713
 3.0    106872
 4.0     96512
 6.0     68612
 1.0     60422
-1.0     28302
Name: count, dtype: int64
```

In [97]:
```
plt.bar(company_size_counts.head(10).index, company_size_counts.head(10).values, color='blue')

plt.xlabel('Company Size')
plt.ylabel('Count')
plt.title('Distribution of Company Sizes')

plt.xticks(np.arange(len(company_size_counts.head(10).index)))

plt.show()
```



Top 10 Benefits Offered by Companies

In [98]:
```
benefits_counts=final_df['Benefits'].value_counts().sort_values(ascending=False)
benefits_counts.head(10)
```

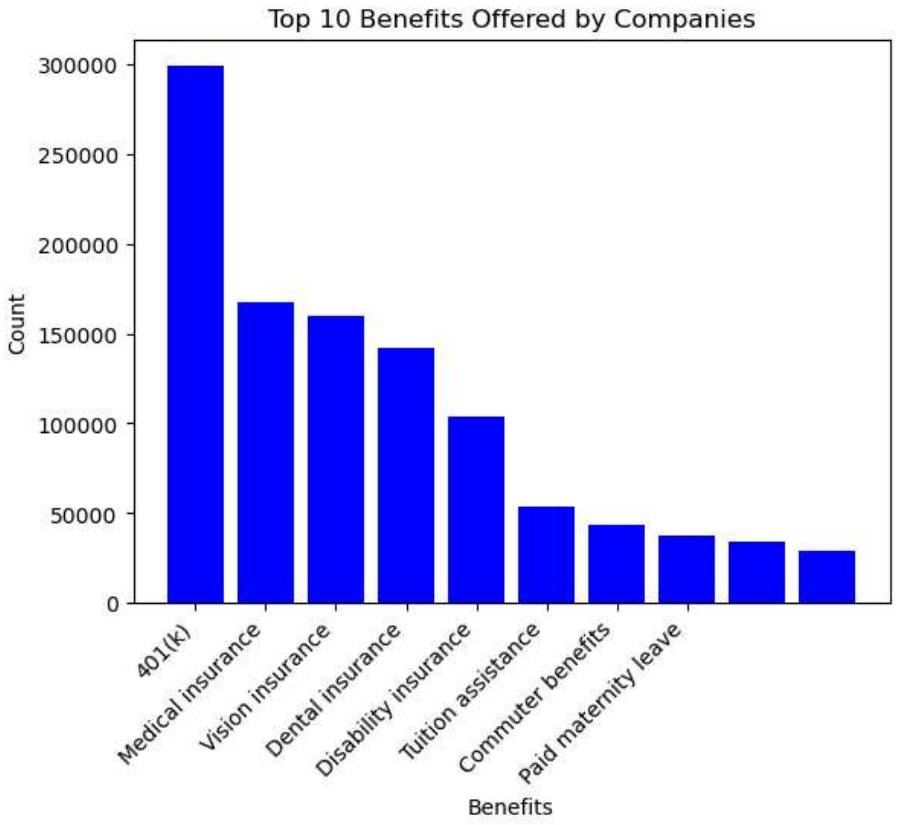
Out[98]:
```
Benefits
401(k)                  298921
Medical insurance       167411
Vision insurance        159764
Dental insurance        141732
Disability insurance    103643
Tuition assistance       53778
Commuter benefits        43247
Paid maternity leave     37549
Paid paternity leave     34153
Pension plan             28845
Name: count, dtype: int64
```

In [99]:
```
plt.bar(benefits_counts.head(10).index, benefits_counts.head(10).values, color='blue')

plt.xlabel('Benefits')
plt.ylabel('Count')
plt.title('Top 10 Benefits Offered by Companies')

plt.xticks(np.arange(len(company_size_counts.head(10).index)), rotation=45, ha='right')
plt.xticks(np.arange(len(company_size_counts.head(10).index)))

plt.show()
```



In [100…
```
InJobs=final_df.to_csv('LinkedInJobs.csv')
```

In [ ]: