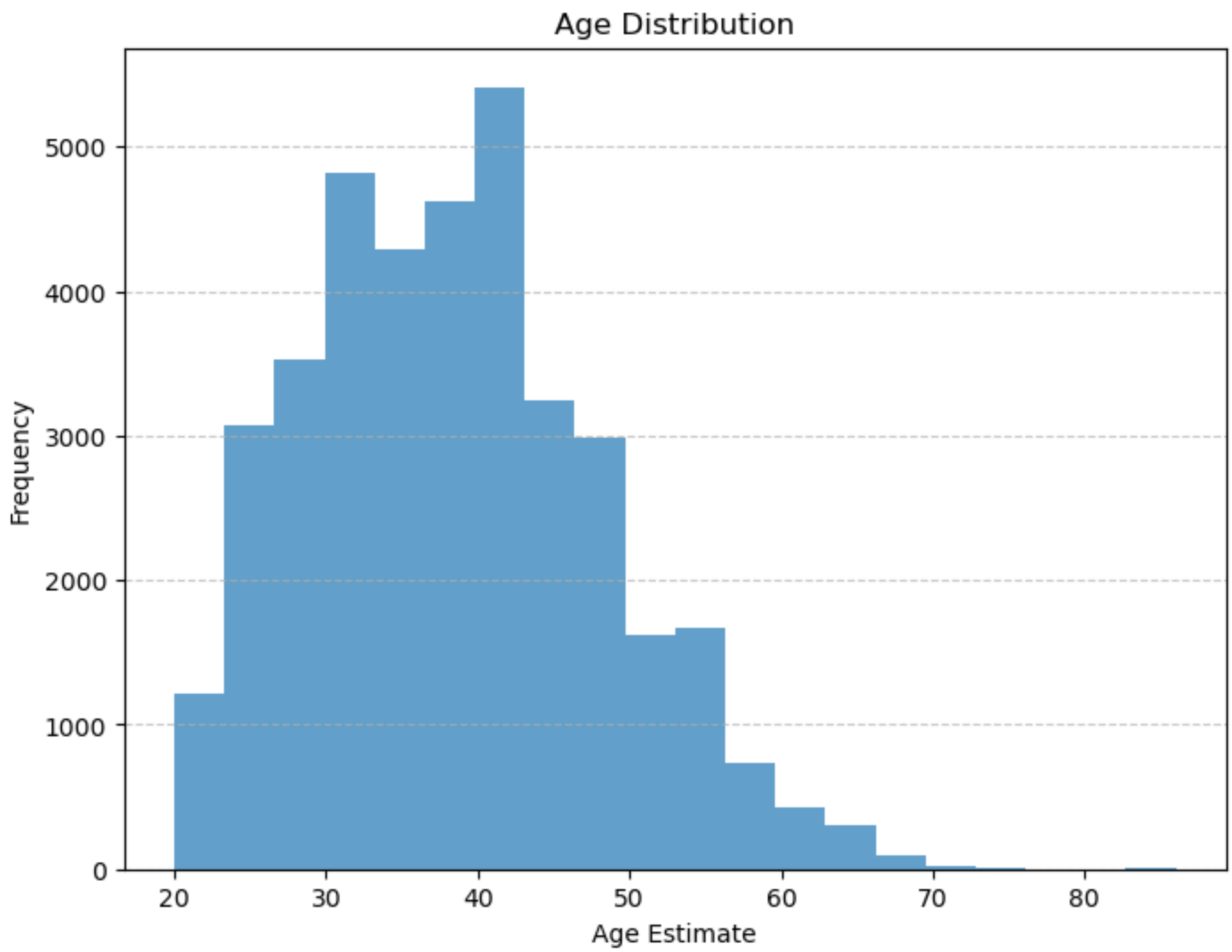
In [42]:

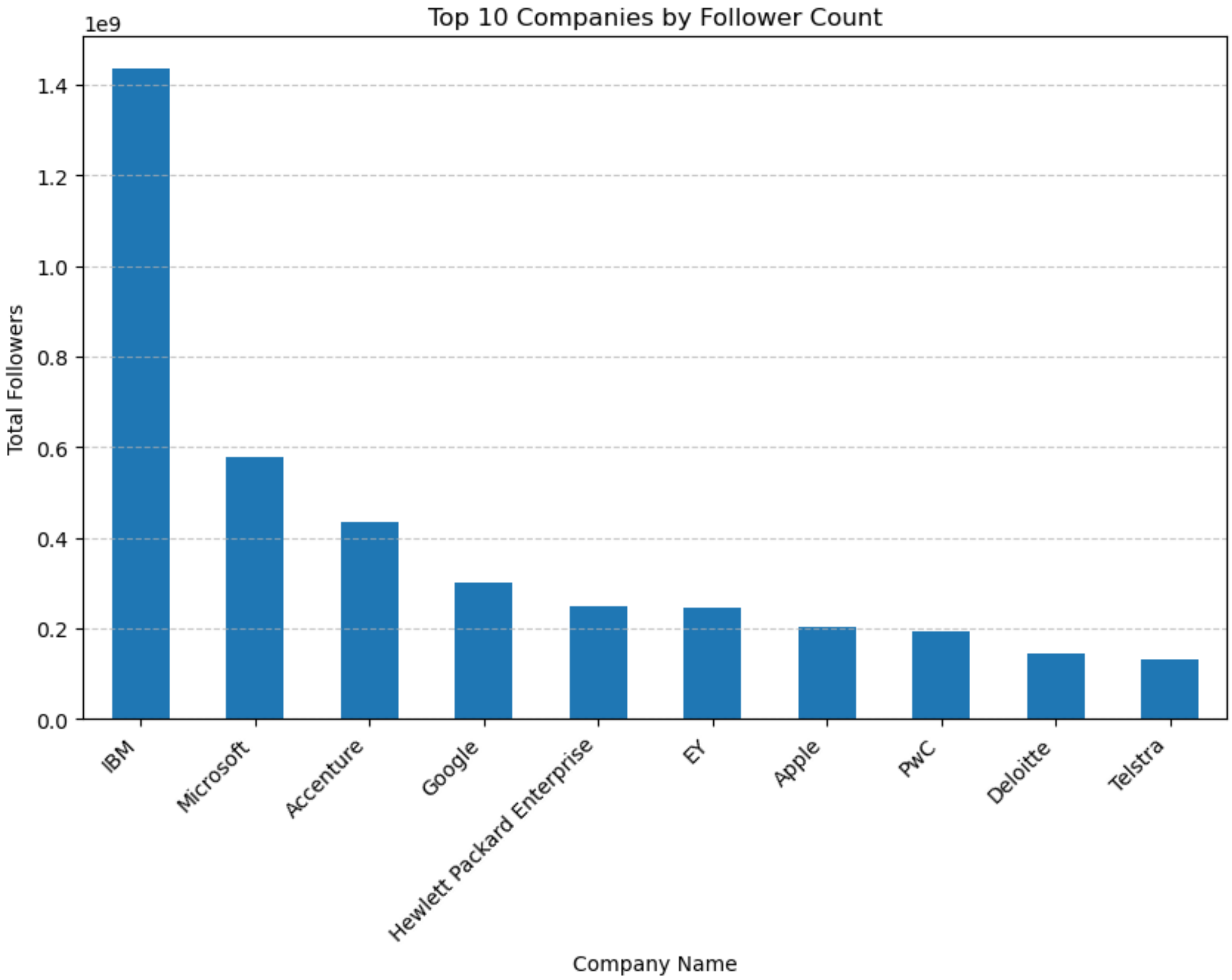
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler


# Load the dataset
file_path = '/Users/lasyatummala/Downloads/dump.csv'
data = pd.read_csv(file_path)

# Step 1: Drop unnecessary columns
columns_to_drop = [
    "Unnamed: 0", "companyHasLogo", "companyUrn",
    "memberUrn", "posLocation", "posLocationCode", "positionId"
]
data_cleaned = data.drop(columns=columns_to_drop, errors='ignore')

# Step 2: Drop rows with missing values in important columns
important_columns = [
    "ageEstimate", "companyName", "companyStaffCount",
    "companyFollowerCount", "connectionsCount",
    "country", "mbrTitle", "posTitle", "startDate"
]
data_cleaned = data_cleaned.dropna(subset=important_columns)

# Step 3: Standardize column names
data_cleaned.columns = [col.lower().replace(" ", "_") for col in data_cleane

# Step 4: Convert date columns to datetime
data_cleaned['startdate'] = pd.to_datetime(data_cleaned['startdate'], errors

# Step 5: Reset index for the cleaned dataset
data_cleaned = data_cleaned.reset_index(drop=True)

# Save the cleaned data to a new file (optional)
data_cleaned.to_csv('cleaned_dataset.csv', index=False)

# Display the first few rows of the cleaned dataset
print(data_cleaned.head())
```

```
   ageestimate  companyfollowercount       companyname  companystaffcount  \
0         41.0              198859.0  Commonwealth Bank            32905.0
1         41.0              198859.0  Commonwealth Bank            32905.0
2         41.0               10047.0           CommSec              619.0
3         41.0              198859.0  Commonwealth Bank            32905.0
4         30.0              300723.0            PayPal            22522.0

                   companyurl  connectionscount country     enddate  \
0  http://www.commbank.com.au/             500.0      au         NaN
1  http://www.commbank.com.au/             500.0      au  2014-06-01
2    http://www.commsec.com.au             500.0      au  2012-12-01
3  http://www.commbank.com.au/             500.0      au  2008-07-01
4        http://www.paypal.com             500.0      au         NaN

   followable  followerscount genderestimate  \
0         1.0           506.0           male
1         1.0           506.0           male
2         1.0           506.0           male
3         1.0           506.0           male
4         1.0           951.0         female

                                        haspicture  ispremium  \
0                                              NaN        0.0
1                                              NaN        0.0
2                                              NaN        0.0
3                                              NaN        0.0
4  RTMZ0-46bTjK4V_MGFDG6i5g0yZmFp5oS0S9liWvpWg.jpg        0.0

               mbrlocation             mbrlocationcode  \
0  Sydney Area, Australia  urn:li:fs_region:(au,4910)
1  Sydney Area, Australia  urn:li:fs_region:(au,4910)
2  Sydney Area, Australia  urn:li:fs_region:(au,4910)
3  Sydney Area, Australia  urn:li:fs_region:(au,4910)
4  Sydney Area, Australia  urn:li:fs_region:(au,4910)

                                    mbrtitle                     postitle  \
0  Portfolio Executive at Commonwealth Bank          Portfolio Executive
1  Portfolio Executive at Commonwealth Bank  Solution Delivery Executive
2  Portfolio Executive at Commonwealth Bank              Project Manager
3  Portfolio Executive at Commonwealth Bank              Project Manager
4          Senior Marketing Manager, PayPal     Senior Marketing Manager

   startdate  avgmemberposduration  avgcompanyposduration
0 2014-07-01              760.5000               989.9361
1 2013-11-01              760.5000               989.9361
2 2008-08-01              760.5000               747.2308
3 2007-02-01              760.5000               989.9361
4 2017-01-01              395.2857               683.3496
```

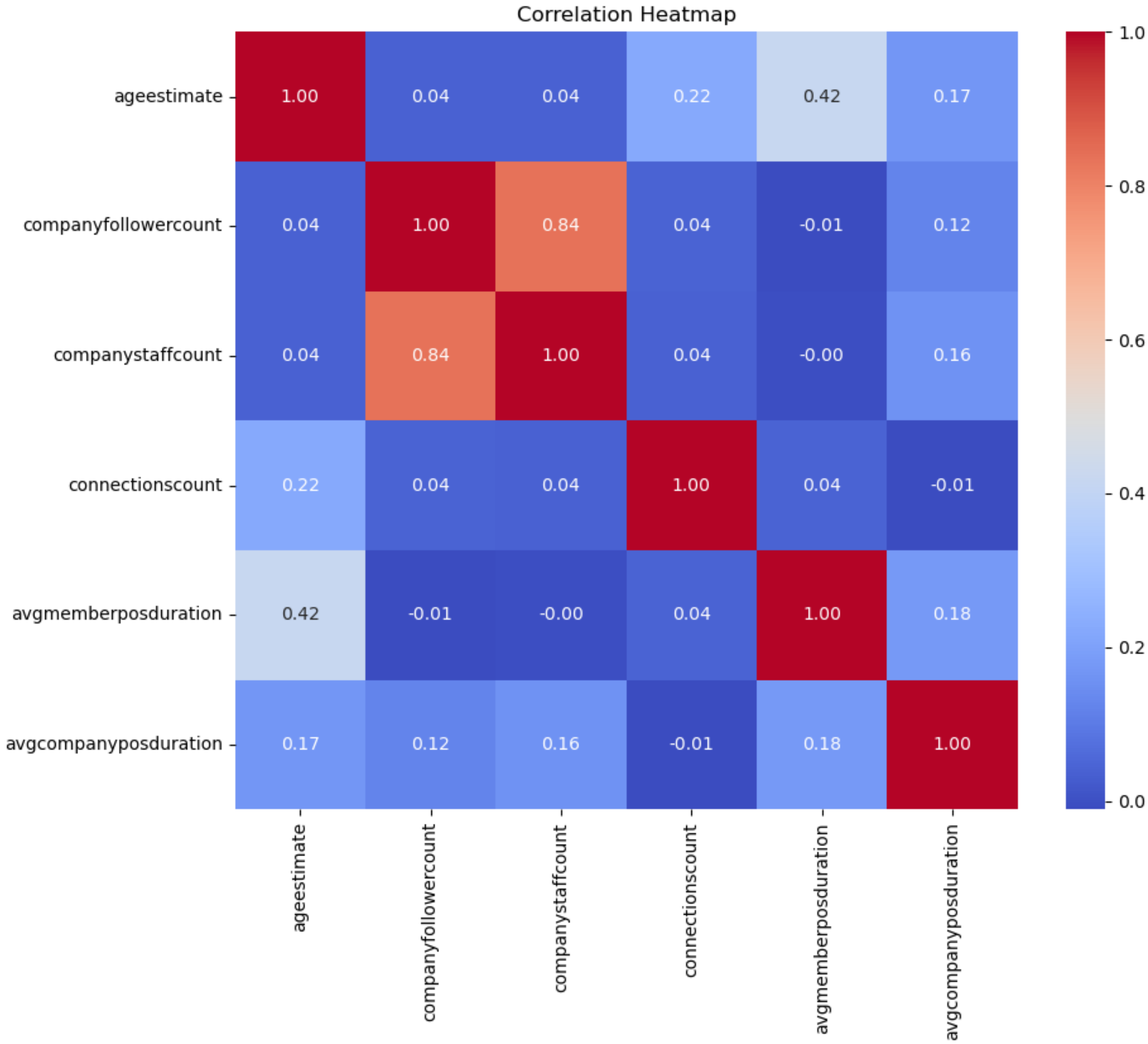
In [43]:
```
# Summary statistics
summary_stats = data_cleaned.describe()

# Age distribution
plt.figure(figsize=(8, 6))
data_cleaned['ageestimate'].plot(kind='hist', bins=20, alpha=0.7)
plt.title("Age Distribution")
plt.xlabel("Age Estimate")
plt.ylabel("Frequency")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



In [44]:
```
# Top 10 companies by follower count
top_companies = data_cleaned.groupby('companyname')['companyfollowercount'].

plt.figure(figsize=(10, 6))
top_companies.plot(kind='bar')
plt.title("Top 10 Companies by Follower Count")
plt.xlabel("Company Name")
plt.ylabel("Total Followers")
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

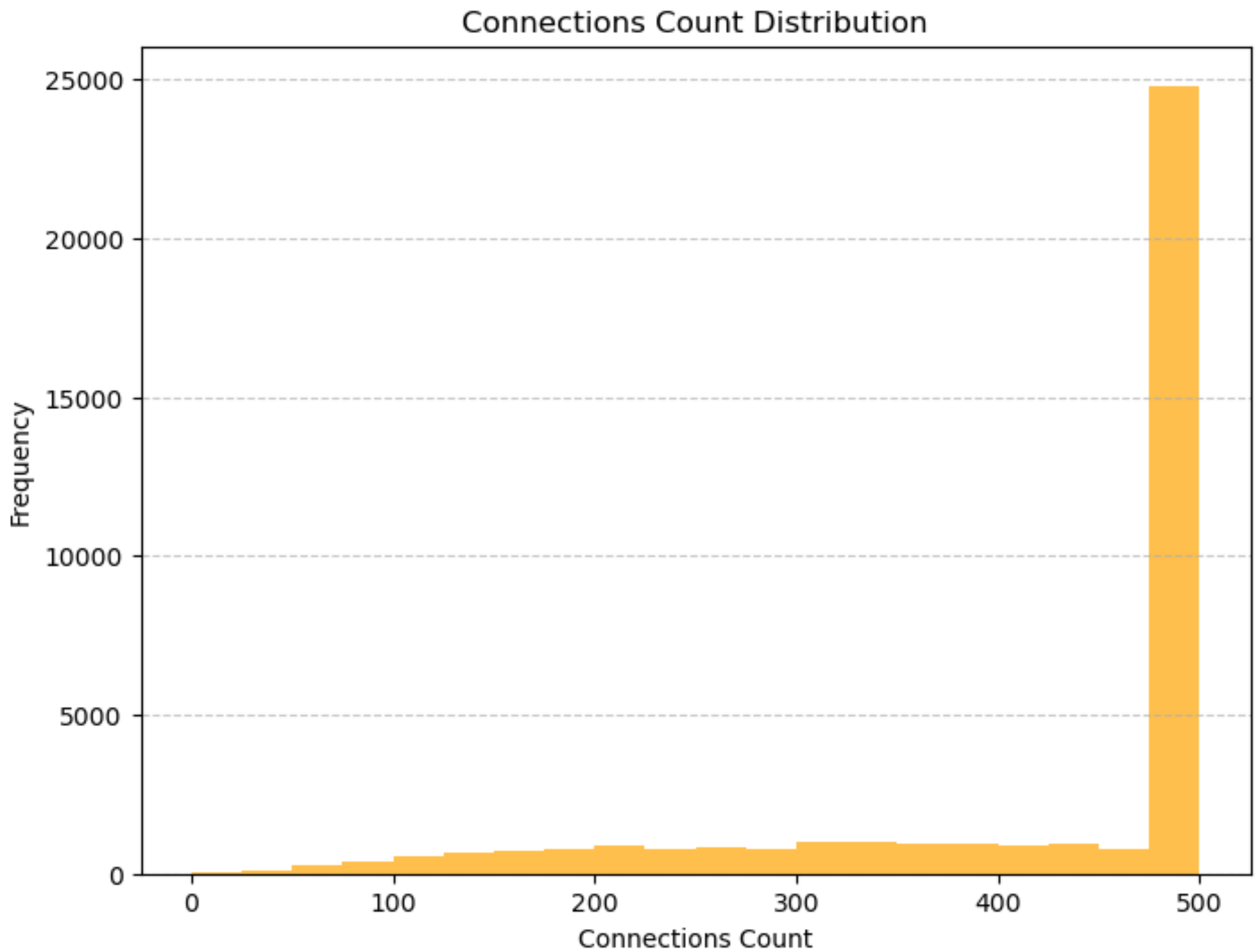


In [45]:
```
# Correlation heatmap for numerical columns
numerical_cols = ['ageestimate', 'companyfollowercount', 'companystaffcount'
                  'connectionscount', 'avgmemberposduration', 'avgcompanypos
correlation_matrix = data_cleaned[numerical_cols].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



In [46]:

```
# Connections count distribution
plt.figure(figsize=(8, 6))
data_cleaned['connectionscount'].plot(kind='hist', bins=20, alpha=0.7, color
plt.title("Connections Count Distribution")
plt.xlabel("Connections Count")
plt.ylabel("Frequency")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



In [47]:
```
# Display summary statistics
print(summary_stats)
```

```
       ageestimate  companyfollowercount  companystaffcount  \
count  38056.000000          3.805600e+04       38056.000000   
mean      38.447971          2.142387e+05       20630.486835   
std        9.608200          6.943557e+05       67827.487532   
min       20.000000          0.000000e+00           0.000000   
25%       31.000000          1.490000e+03         102.000000   
50%       38.000000          1.437050e+04        1324.500000   
75%       45.000000          1.263610e+05        9808.000000   
max       86.000000          7.502740e+06      568533.000000   

      connectionscount    followable  followerscount    ispremium  \
count      38056.000000  38056.000000    38056.000000 38056.000000   
mean         424.637166      0.949706     1183.745060     0.129441   
std          122.964646      0.218554     2958.116725     0.335692   
min            0.000000      0.000000        0.000000     0.000000   
25%          370.750000      1.000000      352.000000     0.000000   
50%          500.000000      1.000000      652.000000     0.000000   
75%          500.000000      1.000000     1186.000000     0.000000   
max          500.000000      1.000000   161922.000000     1.000000   

      avgmemberposduration  avgcompanyposduration  
count          37846.000000           37421.000000  
mean             874.844241             887.609754  
std              634.315739             312.327584  
min                0.000000             -91.000000  
25%              502.714300             731.191900  
50%              730.750000             898.134700  
75%             1068.618050            1037.745100  
max            15492.500000            9497.000000  
```

In [48]:

```python
 data_cleaned['tenure_years'] = (
    pd.to_datetime('today') - data_cleaned['startdate']
).dt.days / 365
```

In [49]:

```python
data_cleaned['pos_follower_ratio'] = (
    data_cleaned['companyfollowercount'] / data_cleaned['companystaffcount']
).fillna(0)
```

In [50]:

```python
data_cleaned = pd.get_dummies(data_cleaned, columns=['country'], drop_first=
```

In [51]:

```python
company_summary = data_cleaned.groupby('companyname').agg({
    'companyfollowercount': 'mean',
    'companystaffcount': 'mean',
    'connectionscount': 'mean'
}).reset_index()
```

In [52]:

```python
scaler = MinMaxScaler()
data_cleaned[['connectionscount', 'avgcompanyposduration']] = scaler.fit_tra
    data_cleaned[['connectionscount', 'avgcompanyposduration']]
)
```

In [58]:
```
# Top individuals by connections
top_individuals = data_cleaned.nlargest(10, 'connectionscount')[['mbrtitle',

# Average connections by industry or company
avg_connections_by_company = data_cleaned.groupby('companyname')['connection
```

In [59]:
```
# Top companies by followers per staff
top_influential_companies = data_cleaned.nlargest(10, 'pos_follower_ratio')[

# Largest companies by staff size
largest_companies = data_cleaned.groupby('companyname')['companystaffcount']
```

In [60]:
```
# Average tenure by company
avg_tenure_by_company = data_cleaned.groupby('companyname')['tenure_years'].

# Companies with lowest tenure
low_tenure_companies = data_cleaned.groupby('companyname')['tenure_years'].m
```

In [61]:
```
# Geographic distribution based on `mbrlocation`
location_distribution = data_cleaned['mbrlocation'].value_counts()

# Display the top 10 locations
print("Top 10 Locations by Number of Profiles:")
print(location_distribution.head(10))
```

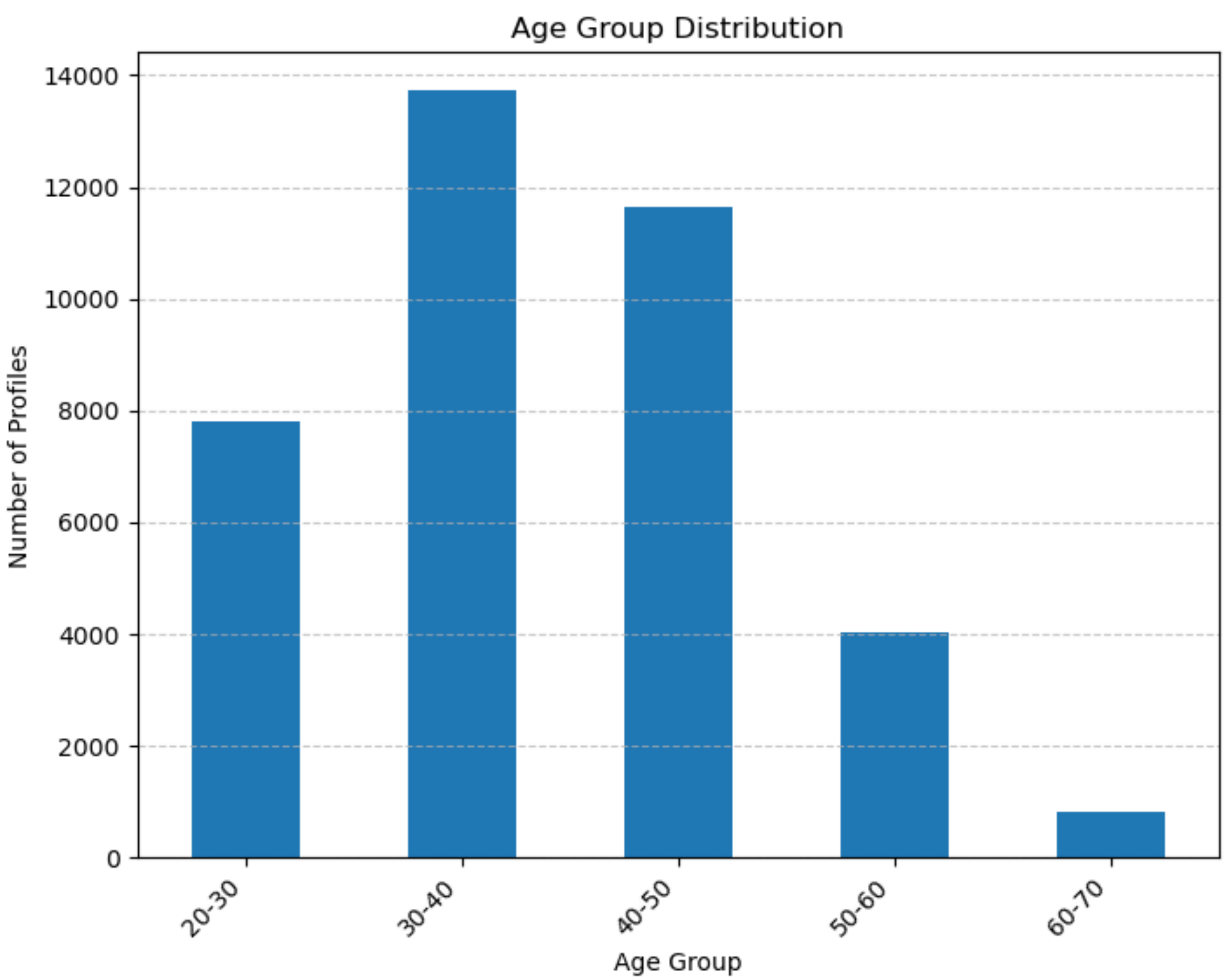
```
Top 10 Locations by Number of Profiles:
Sydney, Australia                         8702
Melbourne, Australia                      6835
Sydney Area, Australia                    4595
Melbourne Area, Australia                 2763
Brisbane, Australia                       2624
Perth, Australia                          1562
Melbourne, Victoria, Australia            1243
Adelaide, Australia                       1126
Sydney, New South Wales, Australia         753
Canberra, Australia                        752
Name: mbrlocation, dtype: int64
```

In [62]:
```
# Top job titles
top_job_titles = data_cleaned['mbrtitle'].value_counts().nlargest(10)

# Top job titles by company
job_titles_by_company = data_cleaned.groupby('companyname')['mbrtitle'].appl
```

In [63]:
```
# Companies with high follower growth potential
growth_potential = data_cleaned[data_cleaned['pos_follower_ratio'] < 1].nlar
```

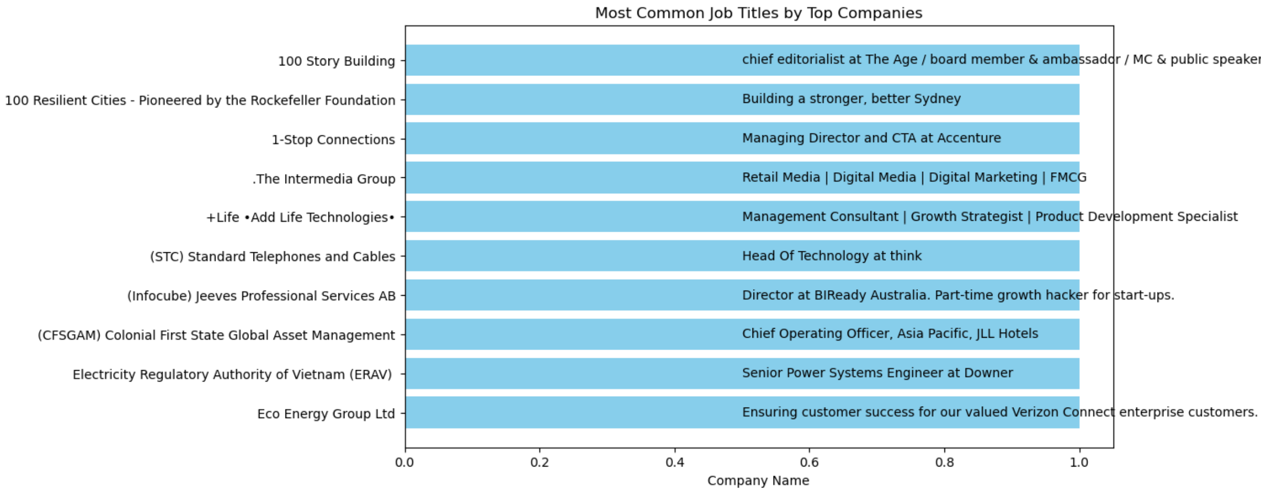
In [64]:

```
# Define age bins and labels
age_bins = [20, 30, 40, 50, 60, 70]
age_labels = ['20-30', '30-40', '40-50', '50-60', '60-70']

# Create the `age_group` column
data_cleaned['age_group'] = pd.cut(data_cleaned['ageestimate'], bins=age_bin

# Count the distribution of age groups
age_group_distribution = data_cleaned['age_group'].value_counts().sort_index

# Plot the age group distribution
plt.figure(figsize=(8, 6))
age_group_distribution.plot(kind='bar')
plt.title("Age Group Distribution")
plt.xlabel("Age Group")
plt.ylabel("Number of Profiles")
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



In [65]:
```python
# Analyze the most common job titles by company
job_titles_by_company = data_cleaned.groupby('companyname')['mbrtitle'].appl

# Display the top 10 companies with their most common job titles
top_companies = job_titles_by_company.head(10)
print("Most Common Job Titles by Company:")
print(top_companies)
# Analyze the most common job titles by company
job_titles_by_company = data_cleaned.groupby('companyname')['mbrtitle'].appl

# Select top companies
top_companies = job_titles_by_company.head(10)

# Visualize using horizontal bars with annotations
plt.figure(figsize=(10, 6))
y_positions = range(len(top_companies))
plt.barh(y_positions, [1] * len(top_companies), color='skyblue')  # Dummy nu
plt.yticks(y_positions, top_companies.index)
plt.xlabel("Company Name")
plt.title("Most Common Job Titles by Top Companies")

# Add annotations for job titles
for i, (company, job_title) in enumerate(zip(top_companies.index, top_compan
    plt.text(0.5, i, job_title, va='center', ha='left', fontsize=10, color='

plt.show()
```

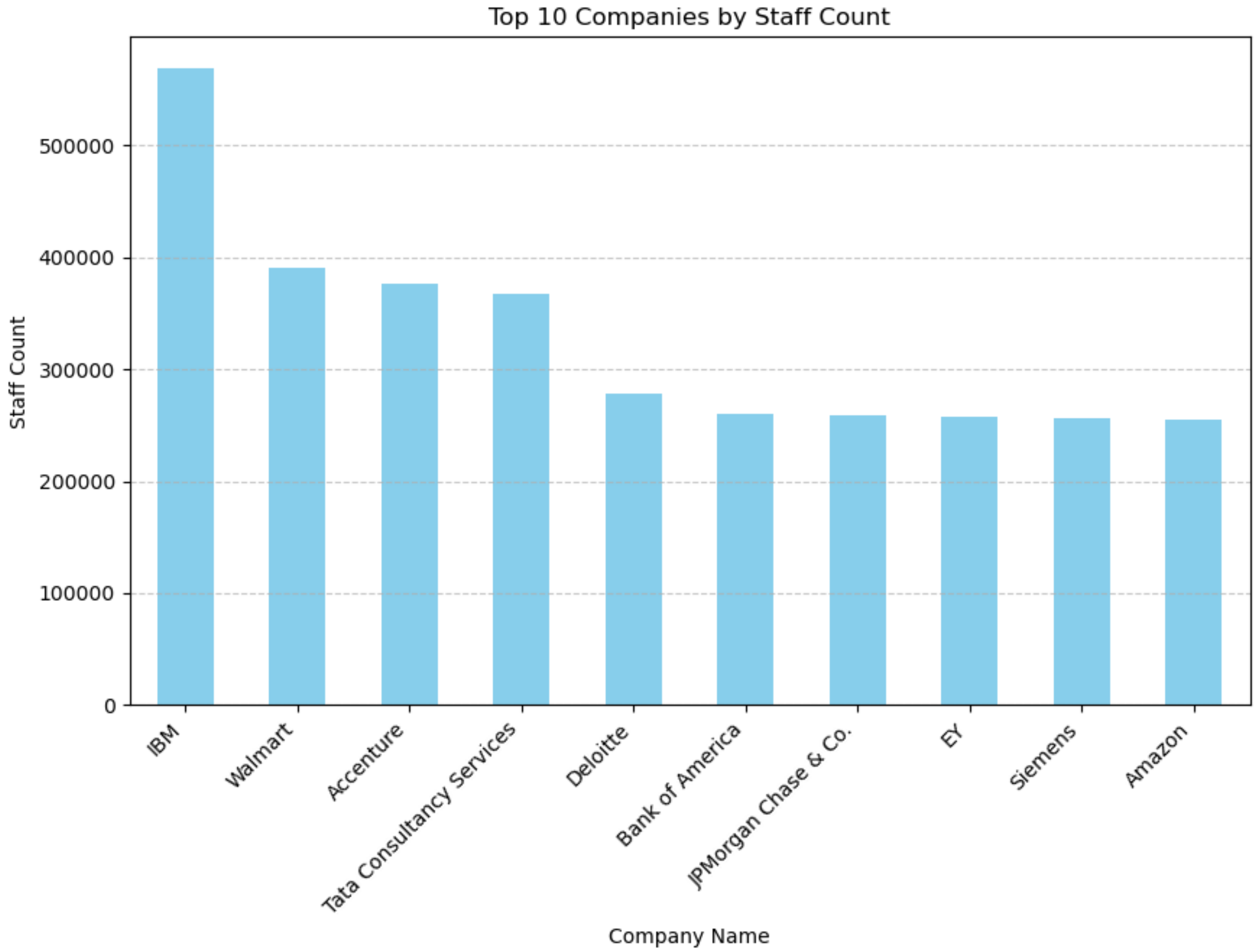
```
Most Common Job Titles by Company:
companyname
 Eco Energy Group Ltd                                                 Ensuring c
ustomer success for our valued Veriz...
 Electricity Regulatory Authority of Vietnam (ERAV)
Senior Power Systems Engineer at Downer
(CFSGAM) Colonial First State Global Asset Management                 Chief Oper
ating Officer, Asia Pacific, JLL Hotels
(Infocube) Jeeves Professional Services AB                            Director a
t BIReady Australia. Part-time growt...
(STC) Standard Telephones and Cables
Head Of Technology at think
+Life •Add Life Technologies•                                         Management
Consultant | Growth Strategist | Pr...
.The Intermedia Group                                                 Retail Med
ia | Digital Media | Digital Marketi...
1-Stop Connections
Managing Director and CTA at Accenture
100 Resilient Cities - Pioneered by the Rockefeller Foundation
Building a stronger, better Sydney
100 Story Building                                                    chief edit
orialist at The Age / board member &...
Name: mbrtitle, dtype: object
```



In [66]:

```
# Analyze the top companies by staff count
top_companies_by_staff = data_cleaned.groupby('companyname')['companystaffcc

# Display the top companies by staff count
print("Top 10 Companies by Staff Count:")
print(top_companies_by_staff)

# Visualization of the top companies by staff count
plt.figure(figsize=(10, 6))
top_companies_by_staff.plot(kind='bar', color='skyblue')
plt.title("Top 10 Companies by Staff Count")
plt.xlabel("Company Name")
plt.ylabel("Staff Count")
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

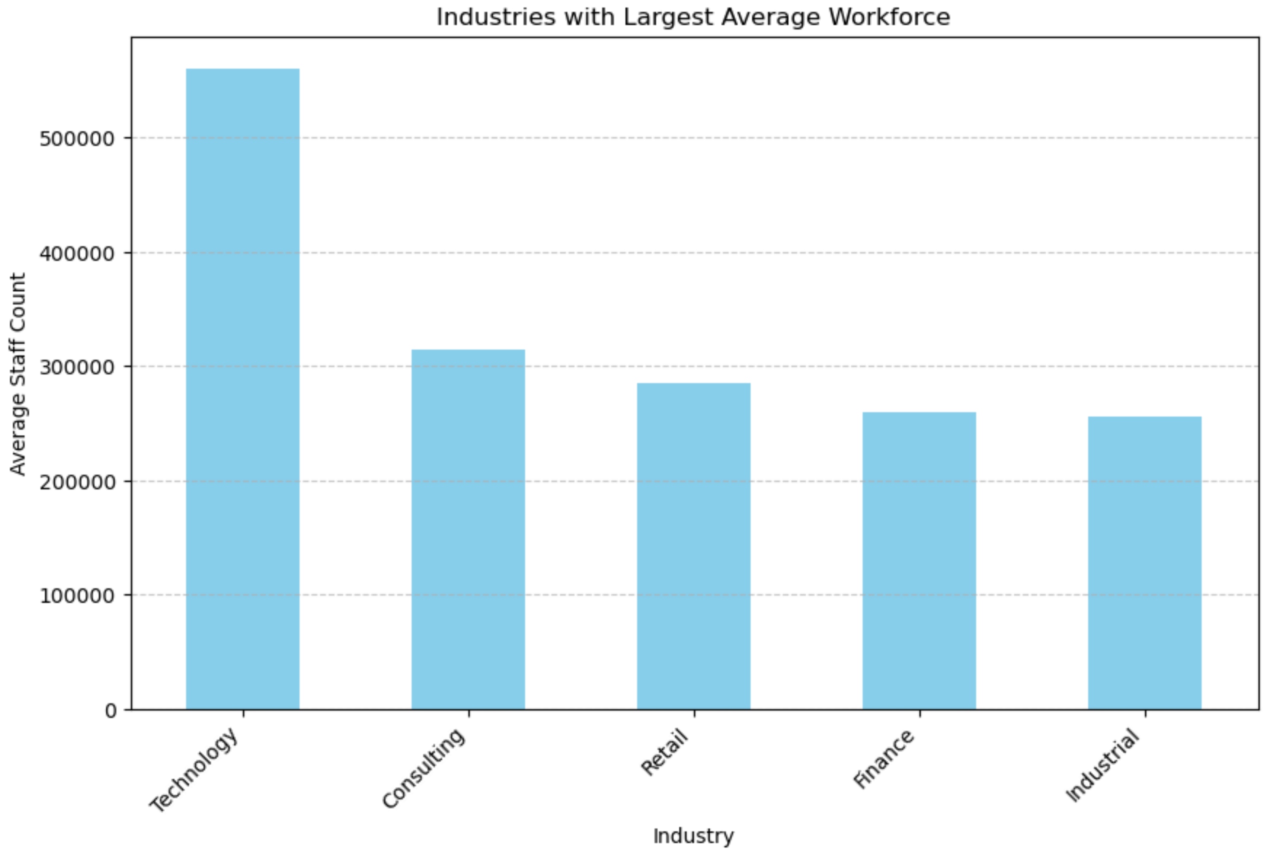
```
Top 10 Companies by Staff Count:
companyname
IBM                            568533.0
Walmart                        391155.0
Accenture                      377002.0
Tata Consultancy Services      367421.0
Deloitte                       277621.0
Bank of America                259914.0
JPMorgan Chase & Co.           258692.0
EY                             257899.0
Siemens                        255714.0
Amazon                         254637.0
Name: companystaffcount, dtype: float64
```


Top 10 Companies by Staff Count
Staff Count
500000
400000
300000
200000
100000
0
IBM
Walmart
Accenture
Tata Consultancy Services
Deloitte
Bank of America
JPMorgan Chase & Co.
EY
Siemens
Amazon
Company Name

In [67]:

```python
# Example of adding an industry column based on company names
industry_mapping = {
    "IBM": "Technology",
    "Walmart": "Retail",
    "Accenture": "Consulting",
    "Tata Consultancy Services": "Technology",
    "Deloitte": "Consulting",
    "Bank of America": "Finance",
    "JPMorgan Chase & Co.": "Finance",
    "EY": "Consulting",
    "Siemens": "Industrial",
    "Amazon": "Retail",
    # Add more mappings as needed
}

# Add industry column to the dataset
data_cleaned['industry'] = data_cleaned['companyname'].map(industry_mapping)

# Group by industry and calculate the average workforce size
industry_workforce = data_cleaned.groupby('industry')['companystaffcount'].m

# Display the industries with the largest workforce trends
print("Industries with the Largest Average Workforce Trends:")
print(industry_workforce)

# Visualization of the largest workforce trends by industry
plt.figure(figsize=(10, 6))
industry_workforce.plot(kind='bar', color='skyblue')
plt.title("Industries with Largest Average Workforce")
plt.xlabel("Industry")
plt.ylabel("Average Staff Count")
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

```
Industries with the Largest Average Workforce Trends:
industry
Technology    560276.202346
Consulting    314062.050847
Retail        284974.333333
Finance       259136.363636
Industrial    255714.000000
Name: companystaffcount, dtype: float64
```



In [68]:

```
# Ensure the startdate is in datetime format
data_cleaned['startdate'] = pd.to_datetime(data_cleaned['startdate'], errors

# Extract the year from the start date
data_cleaned['start_year'] = data_cleaned['startdate'].dt.year

# Group by year to calculate average workforce size
workforce_growth = data_cleaned.groupby('start_year')['companystaffcount'].m

# Drop missing or invalid years
workforce_growth = workforce_growth.dropna()

# Display workforce growth over time
print("Workforce Growth Over Time:")
print(workforce_growth)

# Visualize workforce growth over time
plt.figure(figsize=(10, 6))
workforce_growth.plot(kind='line', marker='o', color='skyblue')
plt.title("Workforce Growth Over Time")
plt.xlabel("Year")
plt.ylabel("Average Staff Count")
plt.grid(axis='both', linestyle='--', alpha=0.7)
plt.show()
```

```
Workforce Growth Over Time:
start_year
1962    50737.000000
1966     1146.000000
1968     4492.000000
1969     4328.500000
1971      485.333333
1972    29730.500000
1973     1204.500000
1974    15248.000000
1975     8400.250000
1976     9341.000000
1977     7652.833333
1978    80469.000000
1979    20512.500000
1980     8903.450000
1981     8003.700000
1982    70205.700000
1983    16614.944444
1984    25818.222222
1985    40177.971429
1986    18142.372093
1987    34606.263158
1988    37399.666667
1989    49641.202247
1990    36468.106796
1991    25483.402299
1992    25694.908257
1993    35487.591837
1994    30789.560440
1995    37026.720165
1996    31699.748000
1997    31393.634868
1998    40480.300771
1999    35794.944206
2000    35108.055821
2001    34406.182421
2002    24794.685990
2003    28706.696237
2004    26349.639796
2005    24207.219965
2006    29688.677803
2007    27040.354423
2008    23791.145152
2009    20778.027194
2010    22757.423181
2011    23086.206074
2012    18822.564781
2013    15839.935123
2014    14666.341441
2015    15481.156184
2016    11683.760738
2017    14564.990114
2018    15071.706099
2019     5948.244898
Name: companystaffcount, dtype: float64
```

Workforce Growth Over Time
Average Staff Count
Year
80000
70000
60000
50000
40000
30000
20000
10000
0
1960
1970
1980
1990
2000
2010
2020

In [ ]: