Analyzing Top Largest Companies in the United States by Revenue

In [1]:

```
from bs4 import BeautifulSoup
from selenium import webdriver
import requests
import pandas as pd
import matplotlib.pyplot as plt
import time
```

Collecting Data Using BeautifulSoup

In [2]:

```
url = "https://en.wikipedia.org/wiki/List_of_largest_companies_in_the_Uni
page = requests.get(url)
soup = BeautifulSoup(page.text,'html')
```

In [3]:

```
soup.find('table',class_ = 'wikitable sortable')
```

Out[3]:

```
<table class="wikitable sortable">
<caption>
</caption>
<tbody><tr>
<th>Rank
</th>
<th>Name
</th>
<th>Industry
</th>
<th>Revenue <br/>(USD millions)
</th>
<th>Revenue growth
</th>
<th>Employees
</th>
<th>Headquarters
</th></tr>
<tr>
```

In [4]:

```
table = soup.find_all('table')[1]
```

In [5]:

```
world_titles=table.find_all('th')
```

In [6]:
```
world_table_titles = [title.text.strip() for title in world_titles]
```

In [7]:
```
d = pd.DataFrame(columns = world_table_titles)
d
```

Out[7]:

Rank	Name	Industry	Revenue (USD millions)	Revenue growth	Employees	Headquarters

In [8]:
```
columndata=table.find_all('tr')
```

In [9]:

```
for row in columndata[1:]:
    row_data = row.find_all('td')
    individual_row_data = [data.text.strip() for data in row_data]
    print(individual_row_data)

    length = len(d)
    d.loc[length]= individual_row_data
```

['1', 'Walmart', 'Retail', '611,289', '6.7%', '2,100,000', 'Bentonville, Arkansas']
['2', 'Amazon', 'Retail and cloud computing', '513,983', '9.4%', '1,540,000', 'Seattle, Washington']
['3', 'ExxonMobil', 'Petroleum industry', '413,680', '44.8%', '62,000', 'Spring, Texas']
['4', 'Apple', 'Electronics industry', '394,328', '7.8%', '164,000', 'Cupertino, California']
['5', 'UnitedHealth Group', 'Healthcare', '324,162', '12.7%', '400,000', 'Minnetonka, Minnesota']
['6', 'CVS Health', 'Healthcare', '322,467', '10.4%', '259,500', 'Woonsocket, Rhode Island']
['7', 'Berkshire Hathaway', 'Conglomerate', '302,089', '9.4%', '383,000', 'Omaha, Nebraska']
['8', 'Alphabet', 'Technology and cloud computing', '282,836', '9.8%', '156,000', 'Mountain View, California']
['9', 'McKesson Corporation', 'Health', '276,711', '4.8%', '48,500', 'Irving, Texas']
['10', 'Chevron Corporation', 'Petroleum industry', '246,252', '51.6%', '43,846', 'San Ramon, California']
['11', 'Cencora', 'Pharmacy wholesale', '238,587', '11.5%', '41,500', 'Chesterbrook, Pennsylvania']
['12', 'Costco', 'Retail', '226,954', '15.8%', '304,000', 'Issaquah, Washington']
['13', 'Microsoft', 'Technology and cloud computing', '198,270', '18.0%', '221,000', 'Redmond, Washington']
['14', 'Cardinal Health', 'Healthcare', '181,364', '11.6%', '46,035', 'Dublin, Ohio']
['15', 'Cigna', 'Health insurance', '180,516', '3.7%', '70,231', 'Bloomfield, Connecticut']
['16', 'Marathon Petroleum', 'Petroleum industry', '180,012', '27.6%', '17,800', 'Findlay, Ohio']
['17', 'Phillips 66', 'Petroleum industry', '175,702', '53.0%', '13,000', 'Houston, Texas']
['18', 'Valero Energy', 'Petroleum industry', '171,189', '58.0%', '9,743', 'San Antonio, Texas']
['19', 'Ford Motor Company', 'Automotive industry', '158,057', '15.9%', '173,000', 'Dearborn, Michigan']
['20', 'The Home Depot', 'Retail', '157,403', '4.1%', '471,600', 'Atlanta, Georgia']
['21', 'General Motors', 'Automotive industry', '156,735', '23.4%', '167,000', 'Detroit, Michigan']
['22', 'Elevance Health', 'Healthcare', '156,595', '13.0%', '102,200', 'Indianapolis, Indiana']
['23', 'JPMorgan Chase', 'Financial services', '154,792', '21.7%', '293,723', 'New York City, New York']
['24', 'Kroger', 'Retail', '148,258', '7.5%', '430,000', 'Cincinnati, Ohio']
['25', 'Centene', 'Healthcare', '144,547', '14.7%', '74,300', 'St. Louis, Missouri']
['26', 'Verizon Communications', 'Telecommunications', '136,835', '2.4%', '117,100', 'New York City, New York']
['27', 'Walgreens Boots Alliance', 'Pharmaceutical industry', '132,703', '10.7%', '262,500', 'Deerfield, Illinois']
['28', 'Fannie Mae', 'Financials', '121,596', '19.7%', '8,000', 'Washington, D.C.']
['29', 'Comcast', 'Telecommunications', '121,427', '4.3%', '186,000', 'P

hiladelphia, Pennsylvania']
['30', 'AT&T', 'Conglomerate and telecomunications', '120,741', '28.5%', '160,700', 'Dallas, Texas']
['31', 'Meta Platforms', 'Technology', '116,609', '1.1%', '86,482', 'Menlo Park, California']
['32', 'Bank of America', 'Financials', '115,053', '22.6%', '216,823', 'Charlotte, North Carolina']
['33', 'Target Corporation', 'Retail', '109,120', '2.9%', '440,000', 'Minneapolis, Minnesota']
['34', 'Dell Technologies', 'Technology', '102,301', '4.4%', '133,000', 'Round Rock, Texas']
['35', 'Archer Daniels Midland', 'Food industry', '101,556', '19.1%', '41,181', 'Chicago, Illinois']
['36', 'Citigroup', 'Financials', '101,078', '26.6%', '238,104', 'New York City, New York']
['37', 'United Parcel Service', 'Transportation', '100,338', '3.1%', '404,700', 'Atlanta, Georgia']
['38', 'Pfizer', 'Pharmaceutical industry', '100,330', '23.4%', '83,000', 'New York City, New York']
['39', "Lowe's", 'Retail', '97,059', '0.8%', '244,500', 'Mooresville, North Carolina']
['40', 'Johnson & Johnson', 'Pharmaceutical industry', '94,943', '1.2%', '152,700', 'New Brunswick, New Jersey']
['41', 'FedEx', 'Transportation', '93,512', '11.4%', '518,249', 'Memphis, Tennessee']
['42', 'Humana', 'Health insurance', '92,870', '11.8%', '67,100', 'Louisville, Kentucky']
['43', 'Energy Transfer Partners', 'Petroleum industry', '89,876', '33.3%', '12,565', 'Dallas, Texas']
['44', 'State Farm', 'Financials', '89,328', '8.6%', '60,519', 'Bloomington, Illinois']
['45', 'Freddie Mac', 'Financials', '86,717', '31.6%', '7,819', 'McLean, Virginia']
['46', 'PepsiCo', 'Beverage', '86,859', '8.7%', '315,000', 'Purchase, New York']
['47', 'Wells Fargo', 'Financials', '82,859', '0.5%', '238,000', 'San Francisco, California']
['48', 'The Walt Disney Company', 'Media', '82,722', '22.7%', '195,800', 'Burbank, California']
['49', 'ConocoPhillips', 'Petroleum industry', '82,156', '69.9%', '9,500', 'Houston, Texas']
['50', 'Tesla', 'Automotive and energy', '81,462', '51.4%', '127,855', 'Austin, Texas']
['51', 'Procter & Gamble', 'Consumer products manufacturing', '80,187', '5.3%', '106,000', 'Cincinnati, Ohio']
['52', 'United States Postal Service', 'Logistics', '78,620', '2.0%', '576,000', 'Washington, D.C.']
['53', 'Albertsons', 'Retail', '77,650', '8.0%', '198,650', 'Boise, Idaho']
['54', 'General Electric', 'Conglomerate', '76,555', '3.2%', '172,000', 'Boston, Massachusetts']
['55', 'MetLife', 'Financials', '69,898', '1.7%', '45,000', 'New York City, New York']
['56', 'Goldman Sachs', 'Financials', '68,711', '5.7%', '48,500', 'New York City, New York']
['57', 'Sysco', 'Food service', '68,636', '33.8%', '70,510', 'Houston, Texas']

['58', 'Bunge Limited', 'Food industry', '67,232', '13.7%', '23,000', 'W
hite Plains, New York']
['59', 'RTX Corporation', 'Conglomerate', '67,074', '4.2%', '182,000',
'Arlington County, Virginia']
['60', 'Boeing', 'Aerospace and defense', '66,608', '6.9%', '156,000',
'Arlington County, Virginia']
['61', 'StoneX Group', 'Financials', '66,036', '55.3%', '4,000[2]', 'New
York City, New York']
['62', 'Lockheed Martin', 'Aerospace and defense', '65,984', '1.6%', '11
6,000', 'Bethesda, Maryland']
['63', 'Morgan Stanley', 'Financials', '65,936', '7.9%', '82,427', 'New
York City, New York']
['64', 'Intel', 'Technology', '63,054', '20.1%', '131,900', 'Santa Clar
a, California']
['65', 'HP', 'Technology', '62,983', '0.8%', '58,000', 'Palo Alto, Calif
ornia']
['66', 'TD Synnex', 'Infotech', '62,344', '97.2%', '28,500', 'Clearwate
r, Florida']
['67', 'IBM', 'Technology and cloud computing', '60,530', '16.3%', '303,
100', 'Armonk, New York']
['68', 'HCA Healthcare', 'Healthcare', '60,233', '2.5%', '250,500', 'Nas
hville, Tennessee']
['69', 'Prudential Financial', 'Financials', '60,050', '15.3%', '39,58
3', 'Newark, New Jersey']
['70', 'Caterpillar', 'Machinery', '59,427', '16.6%', '109,100', 'Deerfi
eld, Illinois']
['71', 'Merck & Co.', 'Pharmaceutical industry', '59,283', '15.8%', '68,
000', 'Kenilworth, New Jersey']
['72', 'World Fuel Services', 'Petroleum industry and logistics', '59,04
3', '88.4%', '5,214', 'Miami, Florida']
['73', 'New York Life Insurance Company', 'Insurance', '58,445', '14.
2%', '15,050', 'New York City, New York']
['74', 'Enterprise Products', 'Petroleum industry', '58,186', '42.6%',
'7,300', 'Houston, Texas']
['75', 'AbbVie', 'Pharmaceutical industry', '58,054', '3.3%', '50,000',
'Lake Bluff, Illinois']
['76', 'Plains All American Pipeline', 'Petroleum industry', '57,342',
'36.3%', '4,100', 'Houston, Texas']
['77', 'Dow Chemical Company', 'Chemical industry', '56,902', '3.5%', '3
7,800', 'Midland, Michigan']
['78', 'AIG', 'Insurance', '56,437', '8.4%', '26,200', 'New York City, N
ew York']
['79', 'American Express', 'Financial', '55,625', '27.3%', '77,300', 'Ne
w York City, New York']
['80', 'Publix', 'Retail', '54,942', '13.5%', '242,000', 'Lakeland, Flor
ida']
['81', 'Charter Communications', 'Telecommunications', '54,022', '4.5%',
'101,700', 'Stamford, Connecticut']
['82', 'Tyson Foods', 'Food processing', '53,282', '13.2%', '142,000',
'Springdale, Arkansas']
['83', 'John Deere', 'Agriculture manufacturing', '52,577', '19.4%', '8
2,239', 'Moline, Illinois']
['84', 'Cisco', 'Telecom hardware manufacturing', '51,557', '3.5%', '83,
300', 'San Jose, California']
['85', 'Nationwide Mutual Insurance Company', 'Financial', '51,450', '8.
6%', '24,791', 'Columbus, Ohio']
['86', 'Allstate', 'Insurance', '51,412', '3.4%', '54,250', 'Northfield

```
Township, Cook County, Illinois']
['87', 'Delta Air Lines', 'Airline', '50,582', '69.2%', '95,000', 'Atlan
ta, Georgia']
['88', 'Liberty Mutual', 'Insurance', '49,956', '3.6%', '50,000', 'Bosto
n, Massachusetts']
['89', 'TJX', 'Retail', '49,936', '2.9%', '329,000', 'Framingham, Massac
husetts']
['90', 'Progressive Corporation', 'Insurance', '49,611', '4.0%', '55,06
3', 'Mayfield Village, Ohio']
['91', 'American Airlines', 'Airline', '48,971', '63.9%', '129,700', 'Fo
rt Worth, Texas']
['92', 'CHS', 'Agriculture cooperative', '47,194', '24.3%', '10,014', 'I
nver Grove Heights, Minnesota']
['93', 'Performance Food Group', 'Food processing', '47,194', '61.6%',
'34,825', 'Richmond, Virginia']
['94', 'PBF Energy', 'Petroleum industry', '46,830', '71.8%', '3,616',
'Parsippany–Troy Hills, New Jersey']
['95', 'Nike', 'Apparel', '46,710', '4.9%', '79,100', 'Beaverton, Orego
n']

['96', 'Best Buy', 'Retail', '46,298', '10.6%', '71,100', 'Richfield, Mi
nnesota']
['97', 'Bristol-Myers Squibb', 'Pharmaceutical industry', '46,159', '0.
5%', '34,300', 'New York City, New York']
['98', 'United Airlines', 'Airline', '44,955', '82.5%', '92,795', 'Chica
go, Illinois']
['99', 'Thermo Fisher Scientific', 'Laboratory instruments', '44,915',
'14.5%', '130,000', 'Waltham, Massachusetts']
['100', 'Qualcomm', 'Technology', '44,200', '31.7%', '51,000', 'San Dieg
o, California']
```

Collected Data in DataFrame

In [10]:

```
d
```

Out[10]:

	Rank	Name	Industry	Revenue (USD millions)	Revenue growth	Employees	Headquarters
0	1	Walmart	Retail	611,289	6.7%	2,100,000	Bentonville, Arkansas
1	2	Amazon	Retail and cloud computing	513,983	9.4%	1,540,000	Seattle, Washington
2	3	ExxonMobil	Petroleum industry	413,680	44.8%	62,000	Spring, Texas
3	4	Apple	Electronics industry	394,328	7.8%	164,000	Cupertino, California
4	5	UnitedHealth Group	Healthcare	324,162	12.7%	400,000	Minnetonka, Minnesota
...	...	...	...	...	...	...	...
95	96	Best Buy	Retail	46,298	10.6%	71,100	Richfield, Minnesota
96	97	Bristol-Myers Squibb	Pharmaceutical industry	46,159	0.5%	34,300	New York City, New York
97	98	United Airlines	Airline	44,955	82.5%	92,795	Chicago, Illinois
98	99	Thermo Fisher Scientific	Laboratory instruments	44,915	14.5%	130,000	Waltham, Massachusetts
99	100	Qualcomm	Technology	44,200	31.7%	51,000	San Diego, California

100 rows × 7 columns

Data Transformation

In [11]:

```
# Convert the values in the 'Revenue (USD millions)' column to string fir
d['Revenue (USD millions)'] = d['Revenue (USD millions)'].astype(str)
# Remove commas from the strings and convert to float
d['Revenue (USD millions)'] = d['Revenue (USD millions)'].str.replace(','
```

In [12]:

```
# Convert the values in the 'Revenue growth' column to string first
d['Revenue growth'] = d['Revenue growth'].astype(str)
# Remove percentage sign from the strings and convert to float
d['Revenue growth'] = d['Revenue growth'].str.rstrip('%').astype(float)
```

In [13]:

```
# Convert the values in the 'Employees' column to string first
d['Employees'] = d['Employees'].astype(str)
# Remove non-numeric characters, such as brackets, and commas from the st
d['Employees'] = d['Employees'].str.replace(r'\D', '', regex=True)
# Convert to float
d['Employees'] = d['Employees'].astype(float)
```

In [14]:

```
# Create empty columns for City and State
d['City'] = ''
d['State'] = ''

# Split the 'Headquarters' column into separate columns for city and stat
headquarters_split = d['Headquarters'].str.split(', ', n=1, expand=True)

# Assign values to City and State columns based on split results
d['City'] = headquarters_split[0]
d['State'] = headquarters_split[1].fillna('')  # Fill NaN values with emp
```

In [15]:

```
d
```

Out[15]:

	Rank	Name	Industry	Revenue (USD millions)	Revenue growth	Employees	Headquarters	
0	1	Walmart	Retail	611289.0	6.7	2100000.0	Bentonville, Arkansas	Ber
1	2	Amazon	Retail and cloud computing	513983.0	9.4	1540000.0	Seattle, Washington	
2	3	ExxonMobil	Petroleum industry	413680.0	44.8	62000.0	Spring, Texas	
3	4	Apple	Electronics industry	394328.0	7.8	164000.0	Cupertino, California	Cu
4	5	UnitedHealth Group	Healthcare	324162.0	12.7	400000.0	Minnetonka, Minnesota	Minı
...	...	...	...	...	...	...	...	
95	96	Best Buy	Retail	46298.0	10.6	71100.0	Richfield, Minnesota	F
96	97	Bristol-Myers Squibb	Pharmaceutical industry	46159.0	0.5	34300.0	New York City, New York	N
97	98	United Airlines	Airline	44955.0	82.5	92795.0	Chicago, Illinois	(
98	99	Thermo Fisher Scientific	Laboratory instruments	44915.0	14.5	130000.0	Waltham, Massachusetts	V
99	100	Qualcomm	Technology	44200.0	31.7	51000.0	San Diego, California	Sa

100 rows × 9 columns

In [16]:

```
d['City'].isnull().value_counts()
```

Out[16]:

```
False    100
Name: City, dtype: int64
```

In [17]:

```python
# Get unique values from the 'Industry' column
unique_categories = d['Industry'].unique()

# Split combined categories if necessary
categories = set()  # Using a set to ensure uniqueness
for category in unique_categories:
    if ' and ' in category:
        # Split combined categories
        split_categories = category.split(' and ')
        categories.update(split_categories)
    else:
        categories.add(category)

# Convert set to list for easier handling
categories_list = list(categories)
print(categories_list)
```

```
['Transportation', 'Laboratory instruments', 'Financials', 'Telecom hard
ware manufacturing', 'telecomunications', 'defense', 'Media', 'logistic
s', 'Food processing', 'Apparel', 'Financial', 'Telecommunications', 'Ma
chinery', 'Agriculture cooperative', 'Automotive industry', 'Electronics
industry', 'Conglomerate', 'Healthcare', 'Automotive', 'Agriculture manu
facturing', 'Insurance', 'Logistics', 'cloud computing', 'Pharmacy whole
sale', 'Infotech', 'Food service', 'Petroleum industry', 'Health', 'Aero
space', 'Financial services', 'Beverage', 'Retail', 'Consumer products m
anufacturing', 'Health insurance', 'Chemical industry', 'energy', 'Food
industry', 'Airline', 'Pharmaceutical industry', 'Technology']
```

In [18]:

```python
split_categories
```

Out[18]:

```
['Petroleum industry', 'logistics']
```

In [19]:

```
d
```

Out[19]:

	Rank	Name	Industry	Revenue (USD millions)	Revenue growth	Employees	Headquarters	
0	1	Walmart	Retail	611289.0	6.7	2100000.0	Bentonville, Arkansas	Ber
1	2	Amazon	Retail and cloud computing	513983.0	9.4	1540000.0	Seattle, Washington	
2	3	ExxonMobil	Petroleum industry	413680.0	44.8	62000.0	Spring, Texas	
3	4	Apple	Electronics industry	394328.0	7.8	164000.0	Cupertino, California	Cu
4	5	UnitedHealth Group	Healthcare	324162.0	12.7	400000.0	Minnetonka, Minnesota	Minı
...	...	...	...	...	...	...	...	
95	96	Best Buy	Retail	46298.0	10.6	71100.0	Richfield, Minnesota	F
96	97	Bristol-Myers Squibb	Pharmaceutical industry	46159.0	0.5	34300.0	New York City, New York	N
97	98	United Airlines	Airline	44955.0	82.5	92795.0	Chicago, Illinois	(
98	99	Thermo Fisher Scientific	Laboratory instruments	44915.0	14.5	130000.0	Waltham, Massachusetts	V
99	100	Qualcomm	Technology	44200.0	31.7	51000.0	San Diego, California	Sa

100 rows × 9 columns

In [20]:

```
split_categories = d['Industry'].str.split(' and ', expand=True)
```

In [21]:

```
split_categories
```

Out[21]:

	0	1
0	Retail	None
1	Retail	cloud computing
2	Petroleum industry	None
3	Electronics industry	None
4	Healthcare	None
...	...	...
95	Retail	None
96	Pharmaceutical industry	None
97	Airline	None
98	Laboratory instruments	None
99	Technology	None

100 rows × 2 columns

In [22]:
```
split_categories[0].value_counts()
```

Out[22]:
```
Retail                              11
Petroleum industry                  11
Financials                          11
Technology                           8
Pharmaceutical industry              6
Healthcare                           6
Insurance                            5
Conglomerate                         4
Telecommunications                   3
Airline                              3
Automotive industry                  2
Health insurance                     2
Food industry                        2
Transportation                       2
Food processing                      2
Financial                            2
Aerospace                            2
Telecom hardware manufacturing       1
Machinery                            1
Agriculture manufacturing            1
Agriculture cooperative              1
Chemical industry                    1
Apparel                              1
Media                                1
Infotech                             1
Food service                         1
Logistics                            1
Consumer products manufacturing      1
Automotive                           1
Beverage                             1
Financial services                   1
Pharmacy wholesale                   1
Health                               1
Electronics industry                 1
Laboratory instruments               1
Name: 0, dtype: int64
```

In [23]:
```
split_categories[1].value_counts()
```

Out[23]:
```
cloud computing      4
defense              2
telecomunications    1
energy               1
logistics            1
Name: 1, dtype: int64
```

In [24]:

```
# Concatenate the split categories with the original DataFrame
d = pd.concat([d, split_categories], axis=1)
```

In [25]:

```
d
```

Out[25]:

	Rank	Name	Industry	Revenue (USD millions)	Revenue growth	Employees	Headquarters	
0	1	Walmart	Retail	611289.0	6.7	2100000.0	Bentonville, Arkansas	Ber
1	2	Amazon	Retail and cloud computing	513983.0	9.4	1540000.0	Seattle, Washington	
2	3	ExxonMobil	Petroleum industry	413680.0	44.8	62000.0	Spring, Texas	
3	4	Apple	Electronics industry	394328.0	7.8	164000.0	Cupertino, California	Cu
4	5	UnitedHealth Group	Healthcare	324162.0	12.7	400000.0	Minnetonka, Minnesota	Minı
...	...	...	...	...	...	...	...	
95	96	Best Buy	Retail	46298.0	10.6	71100.0	Richfield, Minnesota	F
96	97	Bristol-Myers Squibb	Pharmaceutical industry	46159.0	0.5	34300.0	New York City, New York	N
97	98	United Airlines	Airline	44955.0	82.5	92795.0	Chicago, Illinois	(
98	99	Thermo Fisher Scientific	Laboratory instruments	44915.0	14.5	130000.0	Waltham, Massachusetts	V
99	100	Qualcomm	Technology	44200.0	31.7	51000.0	San Diego, California	Sa

100 rows × 11 columns

In [26]:

```
# Get dummies for each split category
dummies = pd.get_dummies(d[[0, 1]], prefix='', prefix_sep='')
```

In [27]:

```
# Concatenate the dummies with the original DataFrame
d = pd.concat([d, dummies], axis=1)
```

In [28]:

```
d
```

Out[28]:

	Rank	Name	Industry	Revenue (USD millions)	Revenue growth	Employees	Headquarters	
0	1	Walmart	Retail	611289.0	6.7	2100000.0	Bentonville, Arkansas	Ber
1	2	Amazon	Retail and cloud computing	513983.0	9.4	1540000.0	Seattle, Washington	
2	3	ExxonMobil	Petroleum industry	413680.0	44.8	62000.0	Spring, Texas	
3	4	Apple	Electronics industry	394328.0	7.8	164000.0	Cupertino, California	Cu
4	5	UnitedHealth Group	Healthcare	324162.0	12.7	400000.0	Minnetonka, Minnesota	Minı
...	...	...	...	...	...	...	...	
95	96	Best Buy	Retail	46298.0	10.6	71100.0	Richfield, Minnesota	F
96	97	Bristol-Myers Squibb	Pharmaceutical industry	46159.0	0.5	34300.0	New York City, New York	N
97	98	United Airlines	Airline	44955.0	82.5	92795.0	Chicago, Illinois	(
98	99	Thermo Fisher Scientific	Laboratory instruments	44915.0	14.5	130000.0	Waltham, Massachusetts	V
99	100	Qualcomm	Technology	44200.0	31.7	51000.0	San Diego, California	Sa

100 rows × 51 columns

In [29]:

```
d1 = d.drop(['Industry', 0, 1,'Headquarters'], axis=1)
```

Transfromed Data

In [30]:

```
d1
```

Out[30]:

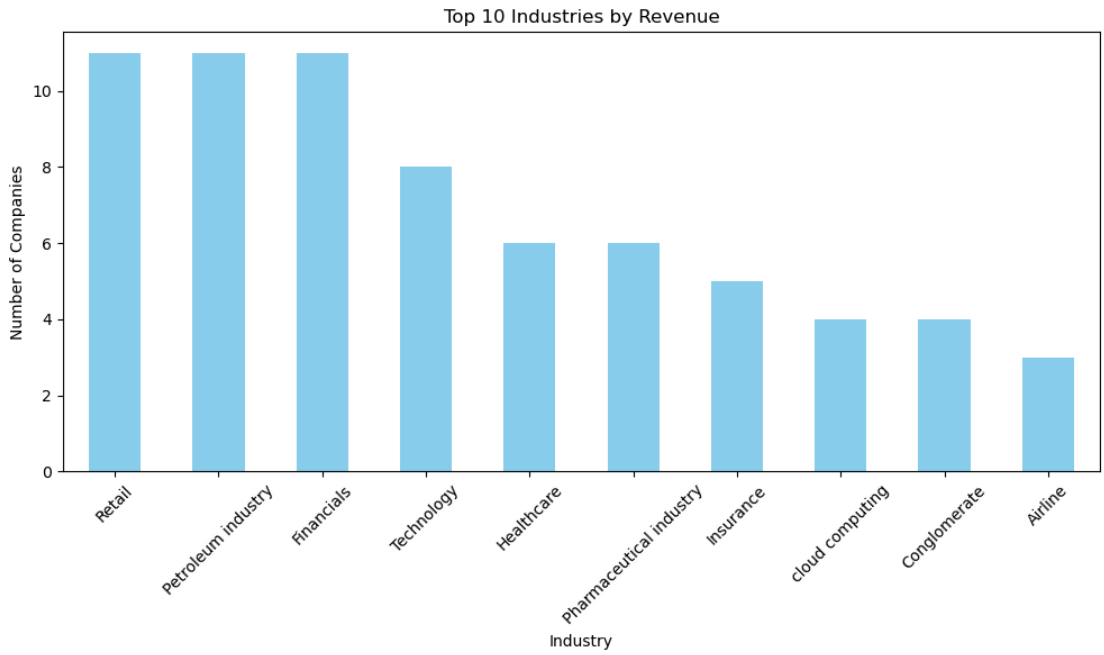
	Rank	Name	Revenue (USD millions)	Revenue growth	Employees	City	State	Aerospa
0	1	Walmart	611289.0	6.7	2100000.0	Bentonville	Arkansas	
1	2	Amazon	513983.0	9.4	1540000.0	Seattle	Washington	
2	3	ExxonMobil	413680.0	44.8	62000.0	Spring	Texas	
3	4	Apple	394328.0	7.8	164000.0	Cupertino	California	
4	5	UnitedHealth Group	324162.0	12.7	400000.0	Minnetonka	Minnesota	
...	...	...	...	...	...	...	...	
95	96	Best Buy	46298.0	10.6	71100.0	Richfield	Minnesota	
96	97	Bristol-Myers Squibb	46159.0	0.5	34300.0	New York City	New York	
97	98	United Airlines	44955.0	82.5	92795.0	Chicago	Illinois	
98	99	Thermo Fisher Scientific	44915.0	14.5	130000.0	Waltham	Massachusetts	
99	100	Qualcomm	44200.0	31.7	51000.0	San Diego	California	

100 rows × 47 columns

Analyzing Data

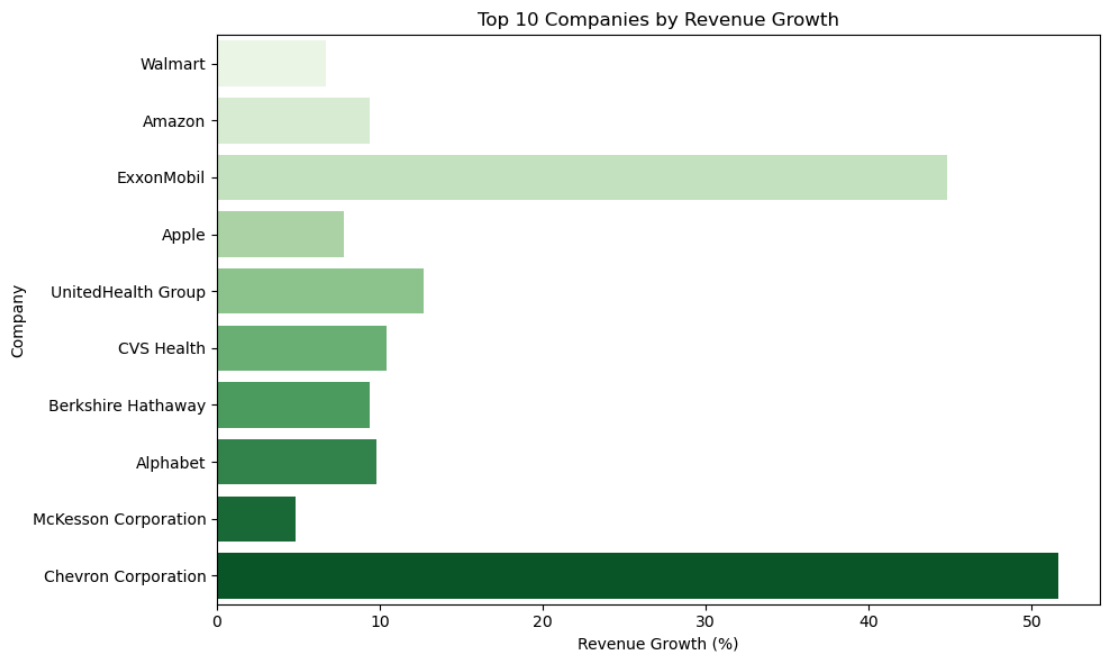
In [31]:

```
# Plotting revenue for top 10 industries
top_industries = d1.iloc[:, 7:].sum().sort_values(ascending=False).head(1
plt.figure(figsize=(10, 6))
top_industries.plot(kind='bar', color='skyblue')
plt.title('Top 10 Industries by Revenue')
plt.xlabel('Industry')
plt.ylabel('Number of Companies')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



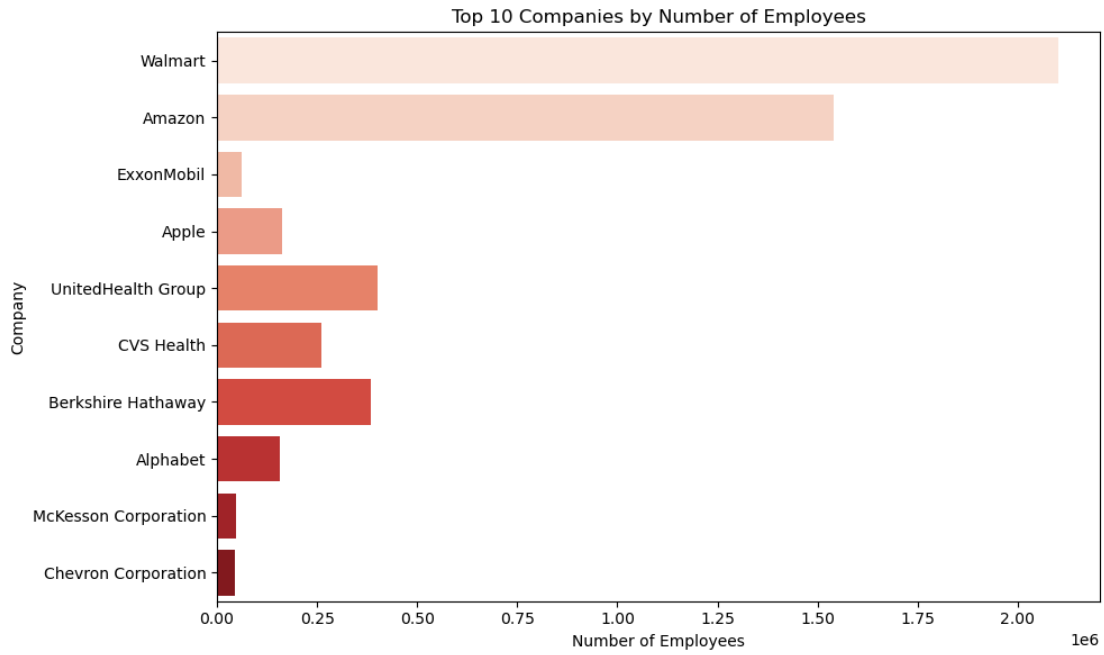
In [32]:

```
import seaborn as sns
# Plotting revenue growth for top 10 companies
top_companies_growth = d1[['Name', 'Revenue growth']].head(10)
plt.figure(figsize=(10, 6))
sns.barplot(x='Revenue growth', y='Name', data=top_companies_growth, pale
plt.title('Top 10 Companies by Revenue Growth')
plt.xlabel('Revenue Growth (%)')
plt.ylabel('Company')
plt.tight_layout()
plt.show()
```



In [33]:

```
# Plotting number of employees for top 10 companies
top_companies_employees = d1[['Name', 'Employees']].head(10)
plt.figure(figsize=(10, 6))
sns.barplot(x='Employees', y='Name', data=top_companies_employees, palett
plt.title('Top 10 Companies by Number of Employees')
plt.xlabel('Number of Employees')
plt.ylabel('Company')
plt.tight_layout()
plt.show()
```



Perfroming Clustering Algorithms

In [34]:

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# Selecting features for clustering
features = d1[['Revenue (USD millions)', 'Revenue growth', 'Employees']]
scaler = StandardScaler()

# Normalizing the features
features_scaled = scaler.fit_transform(features)
```

In [35]:

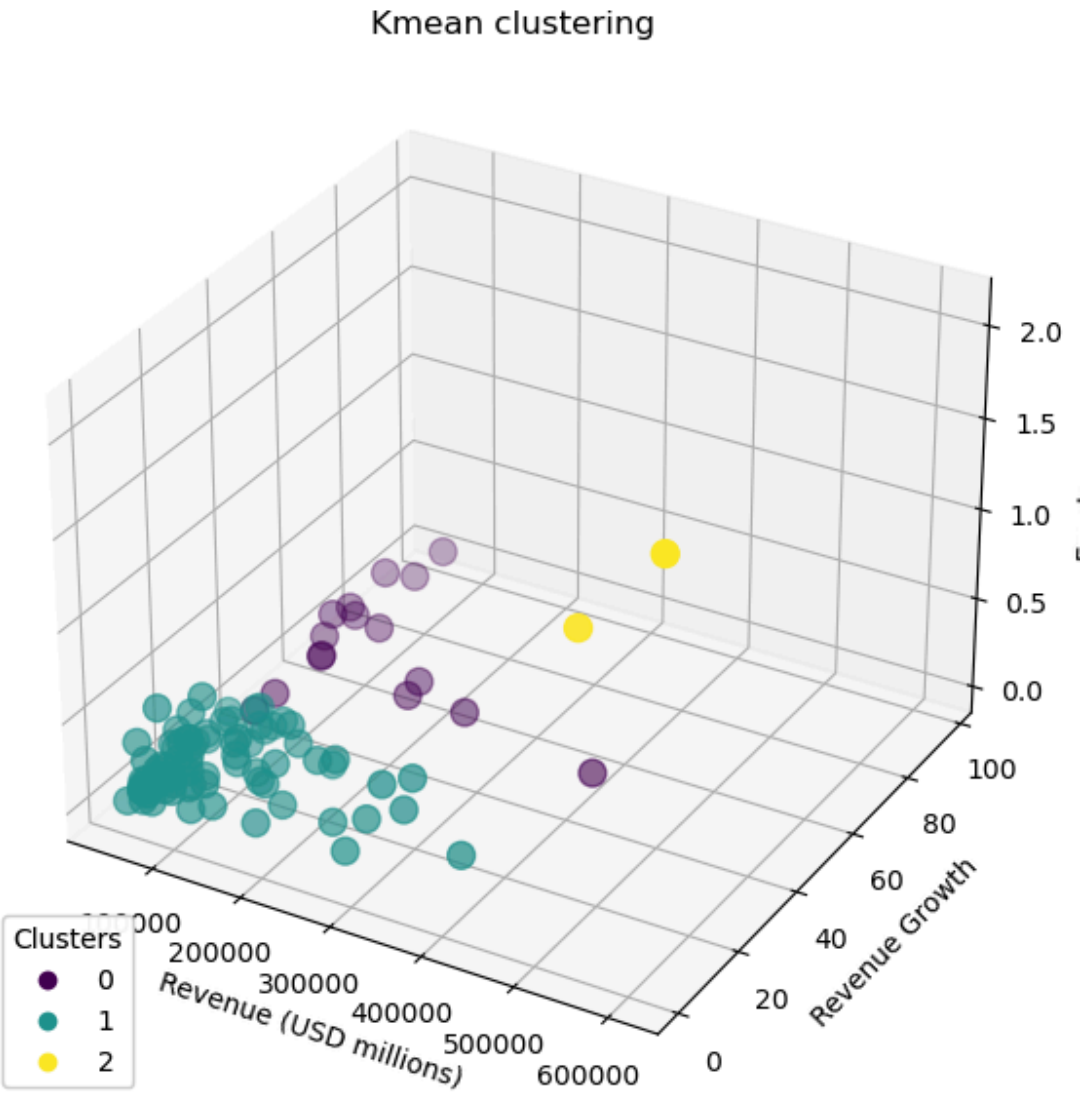
```python
# Applying K-means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
d1['Cluster'] = kmeans.fit_predict(features_scaled)
```

```
C:\Users\amrap\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:14
16: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warn
ing
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\amrap\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:14
40: UserWarning: KMeans is known to have a memory leak on Windows with M
KL, when there are less chunks than available threads. You can avoid it
by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [36]:

```
# Check the centroids of the clusters
centroids = scaler.inverse_transform(kmeans.cluster_centers_)
print(pd.DataFrame(centroids, columns=['Revenue (USD millions)', 'Revenue

# Count number of companies in each cluster
print(d1['Cluster'].value_counts())

# Plotting clusters

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Creating a 3D plot
fig = plt.figure(figsize=(20, 7))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot of your data
scatter = ax.scatter(d1['Revenue (USD millions)'],
                     d1['Revenue growth'],
                     d1['Employees'],
                     c=d1['Cluster'], cmap='viridis', s=100)

# Labels and title
ax.set_xlabel('Revenue (USD millions)')
ax.set_ylabel('Revenue Growth')
ax.set_zlabel('Employees')
plt.title('Kmean clustering')

# Legend with cluster labels
legend1 = ax.legend(*scatter.legend_elements(),
                    loc="lower left", title="Clusters")
ax.add_artist(legend1)

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

```
   Revenue (USD millions)  Revenue growth     Employees
0           106995.250000       62.343750  4.418725e+04
1           108773.780488       11.512195  1.523867e+05
2           562636.000000        8.050000  1.820000e+06
1    82
0    16
2     2
Name: Cluster, dtype: int64
```


Kmean clustering
Clusters
0
1
2
200000
300000
400000
500000
600000
Revenue (USD millions)
0
20
40
60
80
100
Revenue Growth
0.0
0.5
1.0
1.5
2.0

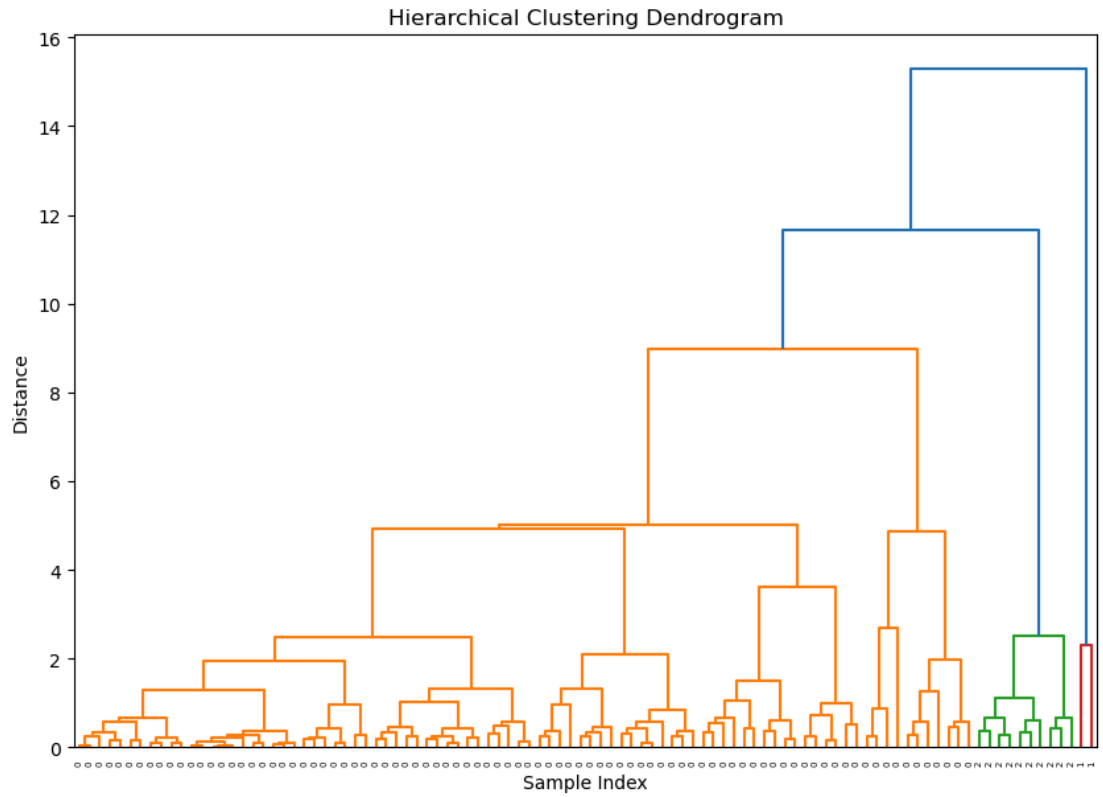
In [37]:

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.cluster.hierarchy import dendrogram, linkage

# Assuming 'd1' is your DataFrame from previous steps
features = d1[['Revenue (USD millions)', 'Revenue growth', 'Employees']]
scaler = StandardScaler()

# Normalize the features
features_scaled = scaler.fit_transform(features)

# Using Agglomerative Clustering
agg_clust = AgglomerativeClustering(n_clusters=3, affinity='euclidean', l
d1['Cluster'] = agg_clust.fit_predict(features_scaled)
```

```
C:\Users\amrap\anaconda3\Lib\site-packages\sklearn\cluster\_agglomerativ
e.py:1006: FutureWarning: Attribute `affinity` was deprecated in version
1.2 and will be removed in 1.4. Use `metric` instead
  warnings.warn(
```

In [38]:

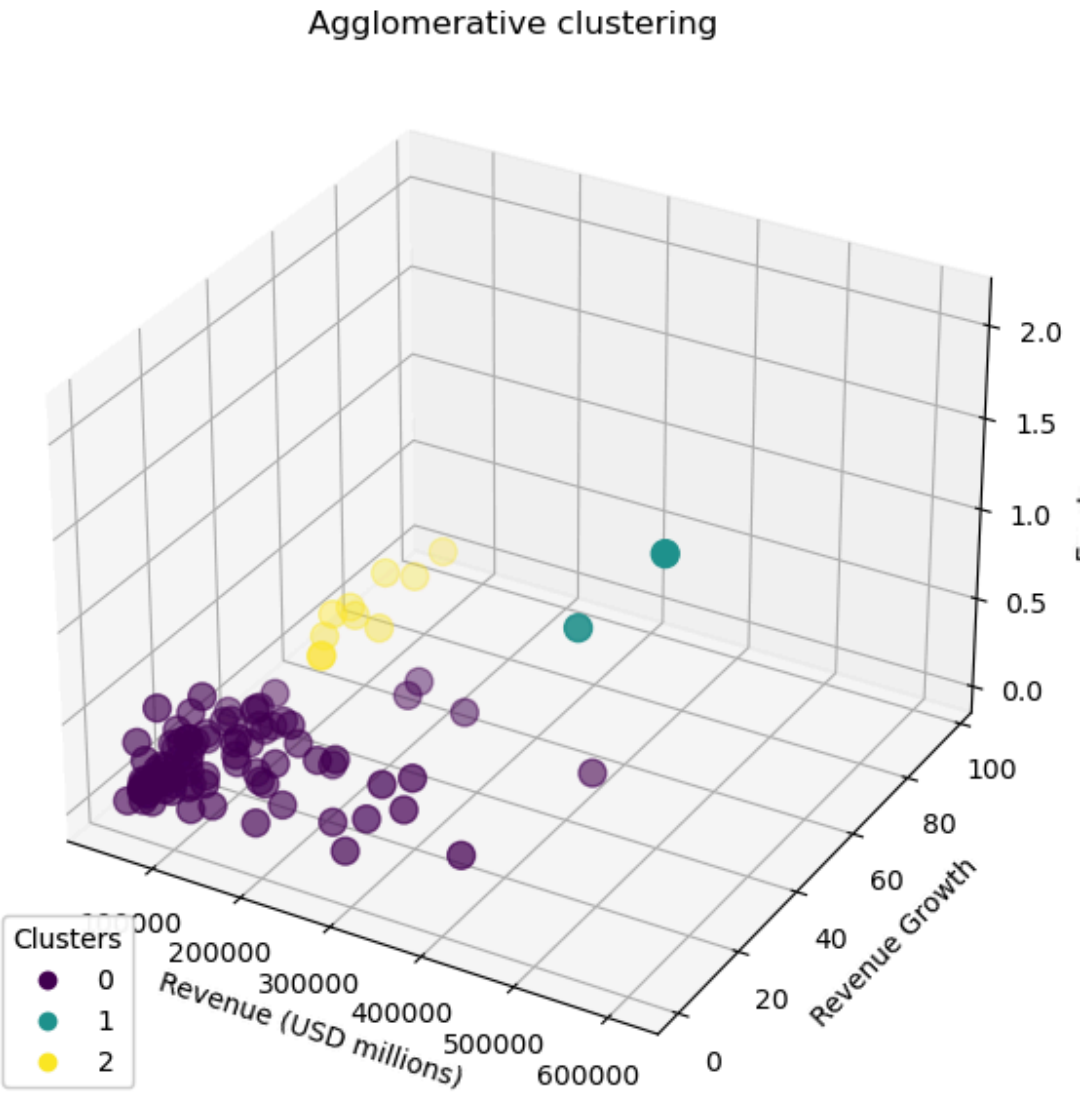
```
# Create the linkage matrix
linked = linkage(features_scaled, 'ward')

plt.figure(figsize=(10, 7))
dendrogram(linked, orientation='top', labels=d1['Cluster'].values, distan
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Sample Index')
plt.ylabel('Distance')
plt.show()
```



In [39]:

```python
# Check how many companies fall into each cluster
print(d1['Cluster'].value_counts())

# Plotting clusters
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Creating a 3D plot
fig = plt.figure(figsize=(20, 7))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot of your data
scatter = ax.scatter(d1['Revenue (USD millions)'],
                     d1['Revenue growth'],
                     d1['Employees'],
                     c=d1['Cluster'], cmap='viridis', s=100)

# Labels and title
ax.set_xlabel('Revenue (USD millions)')
ax.set_ylabel('Revenue Growth')
ax.set_zlabel('Employees')
plt.title('Agglomerative clustering')

# Legend with cluster labels
legend1 = ax.legend(*scatter.legend_elements(),
                    loc="lower left", title="Clusters")
ax.add_artist(legend1)

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

```
0    88
2    10
1     2
Name: Cluster, dtype: int64
```



In [40]:

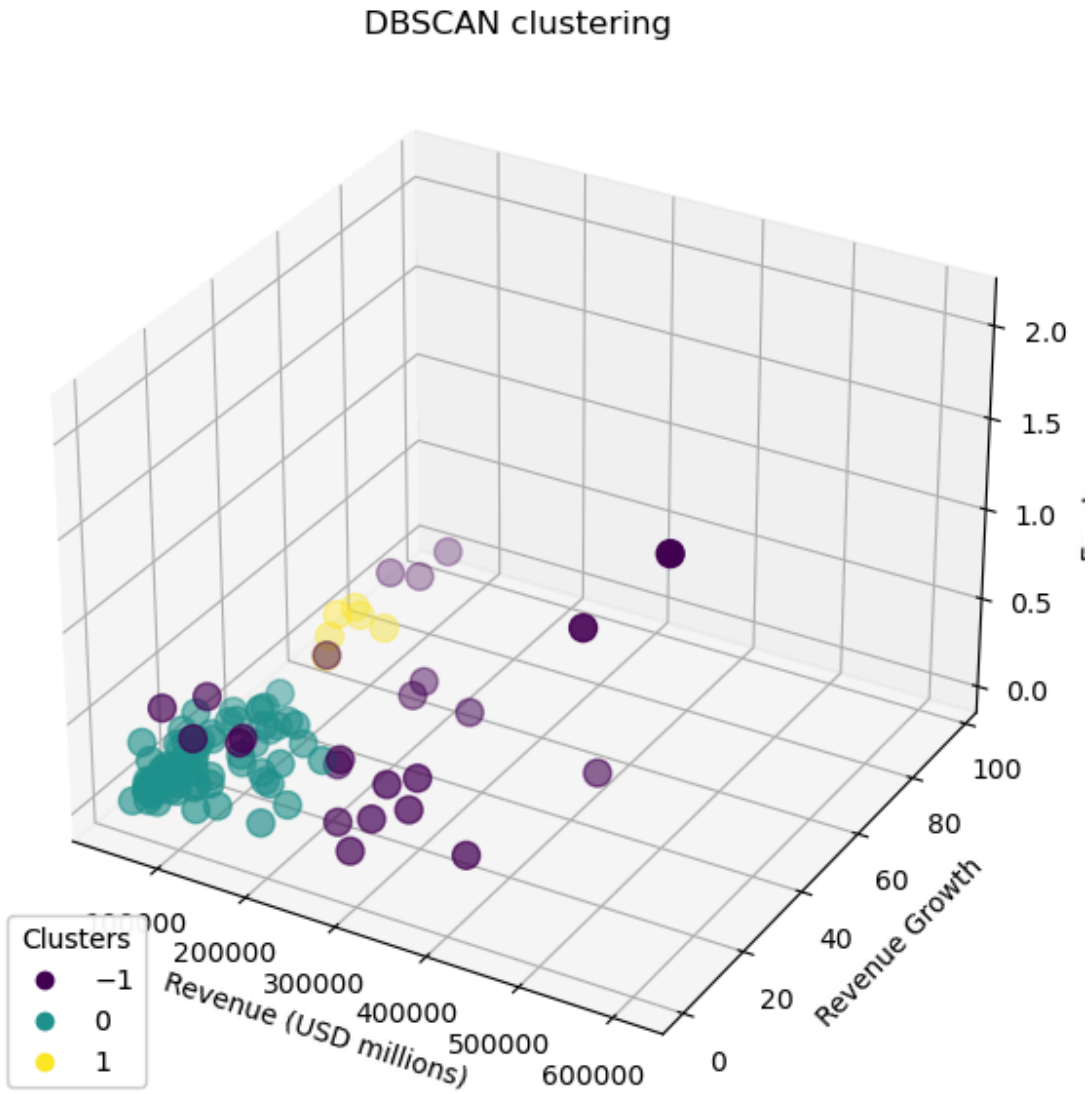
```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming 'd1' is your DataFrame and we're considering the following fea
features = d1[['Revenue (USD millions)', 'Revenue growth', 'Employees']]
scaler = StandardScaler()

# Normalize the features
features_scaled = scaler.fit_transform(features)
```

In [41]:

```
# Applying DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=5)
d1['Cluster'] = dbscan.fit_predict(features_scaled)
```

In [42]:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
# Creating a 3D plot
fig = plt.figure(figsize=(20, 7))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot of your data
scatter = ax.scatter(d1['Revenue (USD millions)'],
                     d1['Revenue growth'],
                     d1['Employees'],
                     c=d1['Cluster'], cmap='viridis', s=100)

# Labels and title
ax.set_xlabel('Revenue (USD millions)')
ax.set_ylabel('Revenue Growth')
ax.set_zlabel('Employees')
plt.title('DBSCAN clustering')

# Legend with cluster labels
legend1 = ax.legend(*scatter.legend_elements(),
                    loc="lower left", title="Clusters")
ax.add_artist(legend1)

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



In [ ]:

```

```

DBSCAN Clustering Algorithm on Different features

In [43]:

```
#CLUSTERING ON INDUSTRIES AND REVNEUES
# Get the names of these top industries
top_industry_names = top_industries.index.tolist()
```

In [44]:

```
top_industries_df = d1[top_industry_names]
```

In [45]:

```
top_industries_df
```

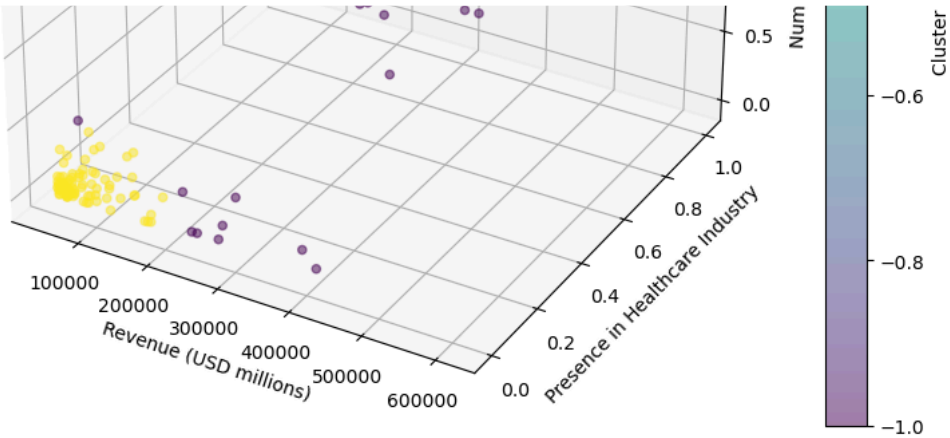
Out[45]:

	Retail	Petroleum industry	Financials	Technology	Healthcare	Pharmaceutical industry	Insurance	com
0	1	0	0	0	0	0	0	
1	1	0	0	0	0	0	0	
2	0	1	0	0	0	0	0	
3	0	0	0	0	0	0	0	
4	0	0	0	0	1	0	0	
...	...	...	...	...	...	...	...	
95	1	0	0	0	0	0	0	
96	0	0	0	0	0	1	0	
97	0	0	0	0	0	0	0	
98	0	0	0	0	0	0	0	
99	0	0	0	1	0	0	0	

100 rows × 10 columns

In [46]:

```
#CLUSTERING ON INDUSTRIES AND REVNEUES
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN
from mpl_toolkits.mplot3d import Axes3D

# Scaling the features
scaler = StandardScaler()

# Loop over each industry column in top_industries_df
for industry_name in top_industries_df.columns:
    # Prepare the feature matrix for clustering
    X = pd.concat([d1[['Revenue (USD millions)', 'Employees']], top_indus
    X_scaled = scaler.fit_transform(X)

    # Apply DBSCAN
    dbscan = DBSCAN(eps=0.5, min_samples=5)
    clusters = dbscan.fit_predict(X_scaled)

    # Setup for 3D plotting
    fig = plt.figure(figsize=(10, 8))
    ax = fig.add_subplot(111, projection='3d')

    # Creating the scatter plot
    scatter = ax.scatter(X['Revenue (USD millions)'], X[industry_name], X

    # Labelling axes
    ax.set_xlabel('Revenue (USD millions)')
    ax.set_ylabel(f'Presence in {industry_name} Industry')
    ax.set_zlabel('Number of Employees')

    # Adding a title
    ax.set_title(f'3D DBSCAN Clustering of Companies by Revenue, {industr

    # Adding a color bar
    color_bar = fig.colorbar(scatter, ax=ax)
    color_bar.set_label('Cluster Label')

    # Show plot
    plt.show()
```



3D DBSCAN Clustering of Companies by Revenue, Pharmaceutical industry, and Employees

A Dashboard to display analyzed data

In [57]:

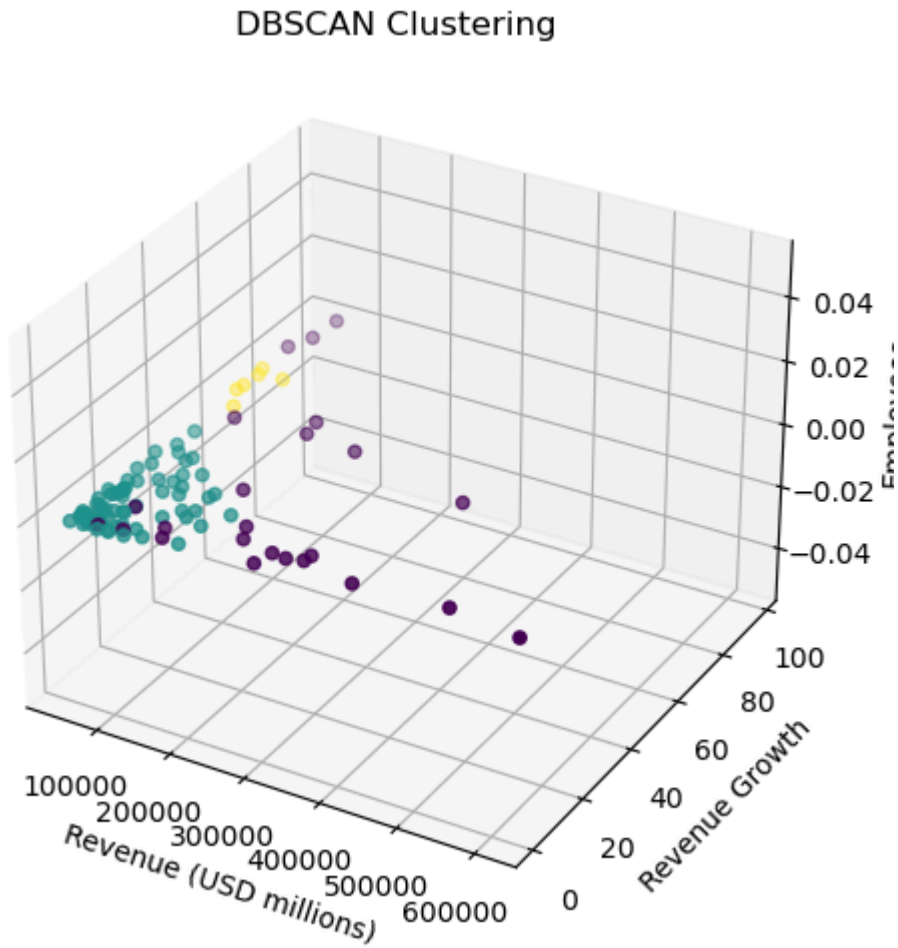
```
###final
import tkinter as tk
from tkinter import ttk
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, DBSCAN
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D

class IntegratedDataDashboard(tk.Tk):
    def __init__(self, df):
        super().__init__()
        self.title("Integrated Data Visualization Dashboard")
        self.geometry("1200x800")
        self.data = df
        self.top_industries_df = top_industries_df
       # self.agglo = ['Agglo histo']

        # Main title for the dashboard
        self.main_title = tk.Label(self, text="Data Visualization Dashboa
        self.main_title.pack(pady=(10, 5))

        # Subtitle for the dashboard
        self.subtitle = tk.Label(self, text="Explore data visualizations
        self.subtitle.pack(pady=(0, 20))

        # Label for the plot dropdown
        self.plot_label = tk.Label(self, text="Select a Visualization or
        self.plot_label.pack(pady=(10, 2))

        # Dropdown to choose the plot type
        self.plot_options = [
            'Top 10 Industries', 'Top 10 Revenue Growth', 'Top 10 Employe
            'K-Means Clustering','Agglomerative Clustering', 'DBSCAN Clus
        ]
        self.selected_option = tk.StringVar(self)
        self.plot_dropdown = ttk.Combobox(self, textvariable=self.selecte
        self.plot_dropdown.pack(pady=20)
        self.plot_dropdown.bind('<<ComboboxSelected>>', self.update_plot)

        # Dropdown for selecting industries if needed
        self.industry_var = tk.StringVar(self)
        self.industry_dropdown = ttk.Combobox(self, textvariable=self.ind
        self.industry_dropdown.pack(pady=20)
        self.industry_dropdown.bind("<<ComboboxSelected>>", self.plot_ind

        # Dropdown for agglometric
        #self.agglo_var = tk.StringVar(self)
        #self.agglo_dropdown = ttk.Combobox(self, textvariable=self.agglo
        #self.agglo_dropdown.pack(pady=20)
        #self.agglo_dropdown.bind("<<ComboboxSelected>>", self.agglo_his)

        # Prepare the figure and canvas for plotting
        self.fig, self.ax = plt.subplots()
        self.canvas = FigureCanvasTkAgg(self.fig, self)
```

```
        self.canvas_widget = self.canvas.get_tk_widget()
        self.canvas_widget.pack(fill=tk.BOTH, expand=True)



    def update_plot(self, event=None):
        plot_type = self.selected_option.get()
        self.ax.clear()
        #self.fig.clf()

        if plot_type == 'Industry Specific DBSCAN':
            self.fig.clf()
            self.industry_dropdown['values'] = self.top_industries_df.col
            self.industry_dropdown.update()
        elif plot_type == 'Top 10 Industries':
            self.plot_top_industries()
        elif plot_type == 'Top 10 Revenue Growth':
            self.plot_revenue_growth()
        elif plot_type == 'Top 10 Employees':
            self.plot_employees()
        elif plot_type == 'K-Means Clustering':
            self.fig.clf()
            self.plot_kmeans()
        elif plot_type == 'Agglomerative Clustering':
            self.fig.clf()
            self.plot_aggcl()
           # if plot_type == 'Agglomerative Clustering':
              #  self.agglo_dropdown['values'] = self.agglo.tolist()
              #  self.aaglo_dropdown.update()
        elif plot_type == 'DBSCAN Clustering':
            self.fig.clf()
            self.plot_dbscan()
        self.canvas.draw()

    def plot_top_industries(self):
        data = self.data.iloc[:, 7:].sum().sort_values(ascending=False).h
        self.ax.bar(data.index, data.values, color='skyblue')
        self.ax.set_title('Top 10 Industries by Revenue')
        self.ax.set_xlabel('Industry')
        self.ax.set_ylabel('Revenue (USD millions)')

    def plot_revenue_growth(self):
        data = self.data[['Name', 'Revenue growth']].head(10)
        sns.barplot(ax=self.ax, x='Revenue growth', y='Name', data=data,
        self.ax.set_title('Top 10 Companies by Revenue Growth')

    def plot_employees(self):
        data = self.data[['Name', 'Employees']].head(10)
        sns.barplot(ax=self.ax, x='Employees', y='Name', data=data, palet
        self.ax.set_title('Top 10 Companies by Number of Employees')

    def plot_kmeans(self):
        features = self.data[['Revenue (USD millions)', 'Revenue growth',
        scaler = StandardScaler()
        features_scaled = scaler.fit_transform(features)
        kmeans = KMeans(n_clusters=3, random_state=42)
        clusters = kmeans.fit_predict(features_scaled)
```

```
        self.ax.clear()
        self.ax = self.fig.add_subplot(111, projection='3d')
        self.ax.scatter(features['Revenue (USD millions)'], features['Rev
        self.ax.set_title('K-Means Clustering')
        self.ax.set_xlabel('Revenue (USD millions)')
        self.ax.set_ylabel('Revenue Growth')
        self.ax.set_zlabel('Employees')
        self.canvas.draw()

    def plot_aggcl(self):
        features = self.data[['Revenue (USD millions)', 'Revenue growth',
        scaler = StandardScaler()
        features_scaled = scaler.fit_transform(features)
        agg_c = AgglomerativeClustering(n_clusters=3, affinity='euclidean
        clusters = agg_c.fit_predict(features_scaled)
        self.ax.clear()
        self.ax = self.fig.add_subplot(111, projection='3d')
        self.ax.scatter(features['Revenue (USD millions)'], features['Rev
        self.ax.set_title('K-Means Clustering')
        self.ax.set_xlabel('Revenue (USD millions)')
        self.ax.set_ylabel('Revenue Growth')
        self.ax.set_zlabel('Employees')
        self.canvas.draw()

    def agglo_his(self):
        linked = linkage(features_scaled, 'ward')
        dendrogram(linked, orientation='top', labels=clusters.values, dis
        self.ax.set_title('Hierarchical Clustering Dendrogram')
        self.ax.set_xlabel('Index')
        self.ax.set_xlabel('Distance')
        self.canvas.draw()


    def plot_dbscan(self):
        features = self.data[['Revenue (USD millions)', 'Revenue growth',
        scaler = StandardScaler()
        features_scaled = scaler.fit_transform(features)
        dbscan = DBSCAN(eps=0.5, min_samples=5)
        clusters = dbscan.fit_predict(features_scaled)
        self.ax.clear()
        self.ax = self.fig.add_subplot(111, projection='3d')
        self.ax.scatter(features['Revenue (USD millions)'], features['Rev
        self.ax.set_title('DBSCAN Clustering')
        self.ax.set_xlabel('Revenue (USD millions)')
        self.ax.set_ylabel('Revenue Growth')
        self.ax.set_zlabel('Employees')
        self.canvas.draw()


    def plot_industry_dbscan(self, event):
        industry_name = self.industry_var.get()
        X = pd.concat([self.data[['Revenue (USD millions)', 'Employees']]
        scaler = StandardScaler()
        X_scaled = scaler.fit_transform(X)
        dbscan = DBSCAN(eps=1.5, min_samples=2)
        clusters = dbscan.fit_predict(X_scaled)
        self.ax.clear()
```

```
        self.ax = self.fig.add_subplot(111, projection='3d')
        self.ax.scatter(X['Revenue (USD millions)'], X['Employees'], X[in
        self.ax.set_xlabel('Revenue (USD millions)')
        self.ax.set_ylabel('Employees')
        self.ax.set_zlabel(f'Presence in {industry_name}')
        self.ax.set_title(f'3D DBSCAN Clustering in {industry_name}')
        self.canvas.draw()

if __name__ == '__main__':
    df = d1
    app = IntegratedDataDashboard(df)
    app.mainloop()
```

```
C:\Users\amrap\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:14
16: FutureWarning: The default value of `n_init` will change from 10 to
'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warn
ing
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\amrap\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:14
40: UserWarning: KMeans is known to have a memory leak on Windows with M
KL, when there are less chunks than available threads. You can avoid it
by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\amrap\anaconda3\Lib\site-packages\sklearn\cluster\_agglomerativ
e.py:1006: FutureWarning: Attribute `affinity` was deprecated in version
1.2 and will be removed in 1.4. Use `metric` instead
  warnings.warn(
```


DBSCAN Clustering
0.04
0.02
0.00
−0.02
−0.04
100
80
60
40
20
0
Revenue Growth
100000
200000
300000
400000
500000
600000
Revenue (USD millions)

In [ ]: