WEB SCRAPING

Presented By : Ishank Verma

www.linkedin.com/in/ishank-verma

OBJECTIVE

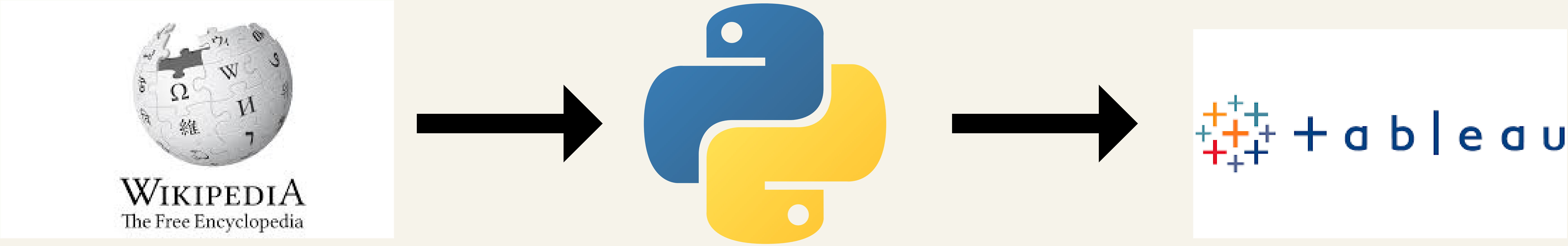
The project aims to scrape data from Wikipedia on the largest technology companies by revenue, extract key metrics like total revenue, number of employees, and revenue per employee, clean and analyze the data to identify trends and insights, and present these findings in an actionable format for strategic decision-making in the technology sector.

WEB SCRAPING

Web scraping is like having a robot that collects information from websites automatically. It reads web pages, finds the data you want (like prices or news articles), and puts it all together for you to use. Just remember to follow the rules and be respectful of website owners' rights when scraping data.

There are lots of web scraping techniques, and here I am using BeautifulSoup, a popular tool for parsing HTML and extracting data from web pages.

DATA FLOW

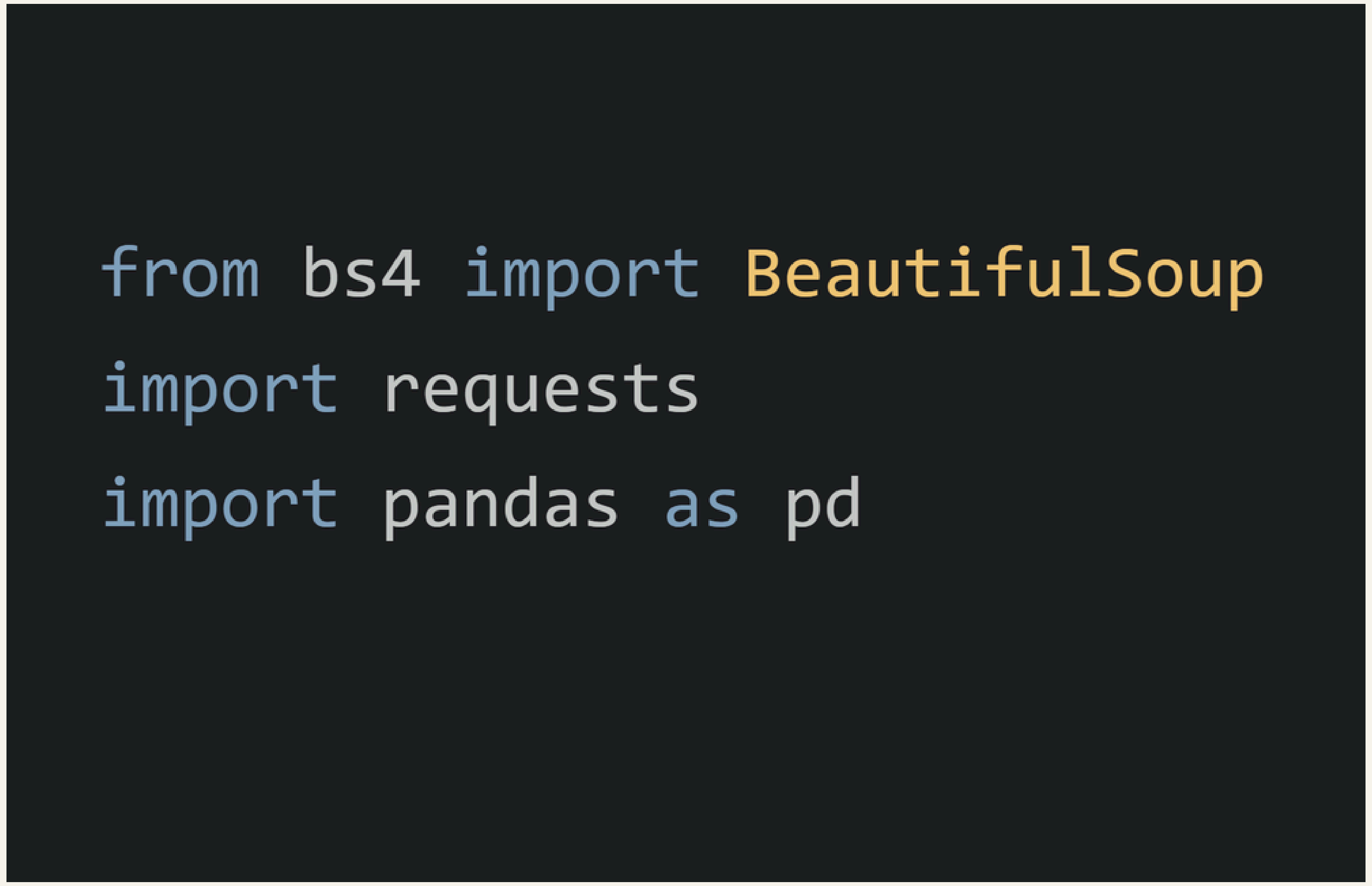


Data Source: Wikipedia

Web Scraping with python

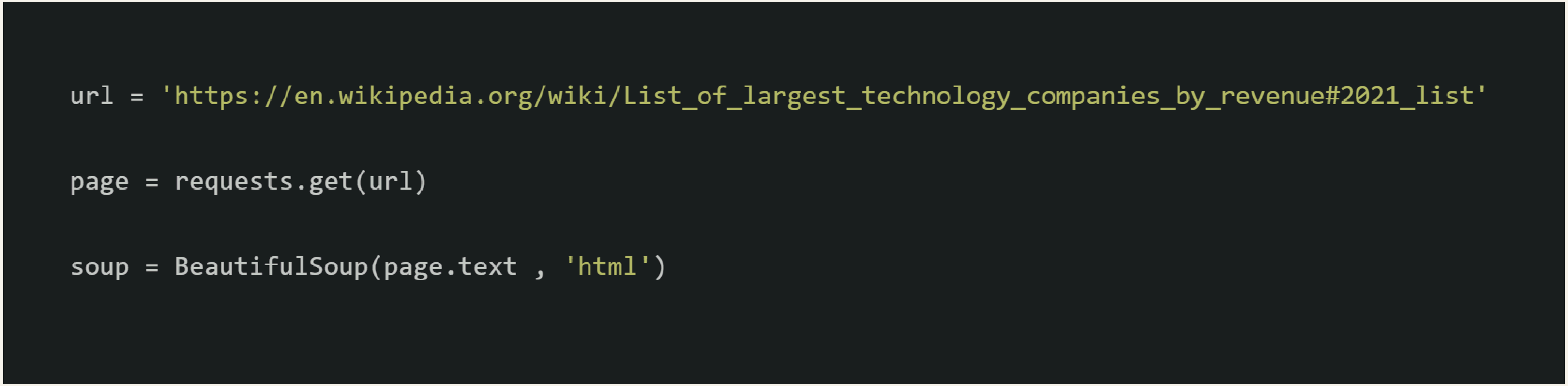
Data Visualization

LIBRARIES



```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

1. **Beautiful Soup(bs4):** Used for parsing HTML and XML documents during web scraping
2. **Requests:** Facilitates making HTTP requests to fetch web page content.
3. **Pandas (pd) :** Handles data manipulation and analysis, organizing scraped data into structured formats.

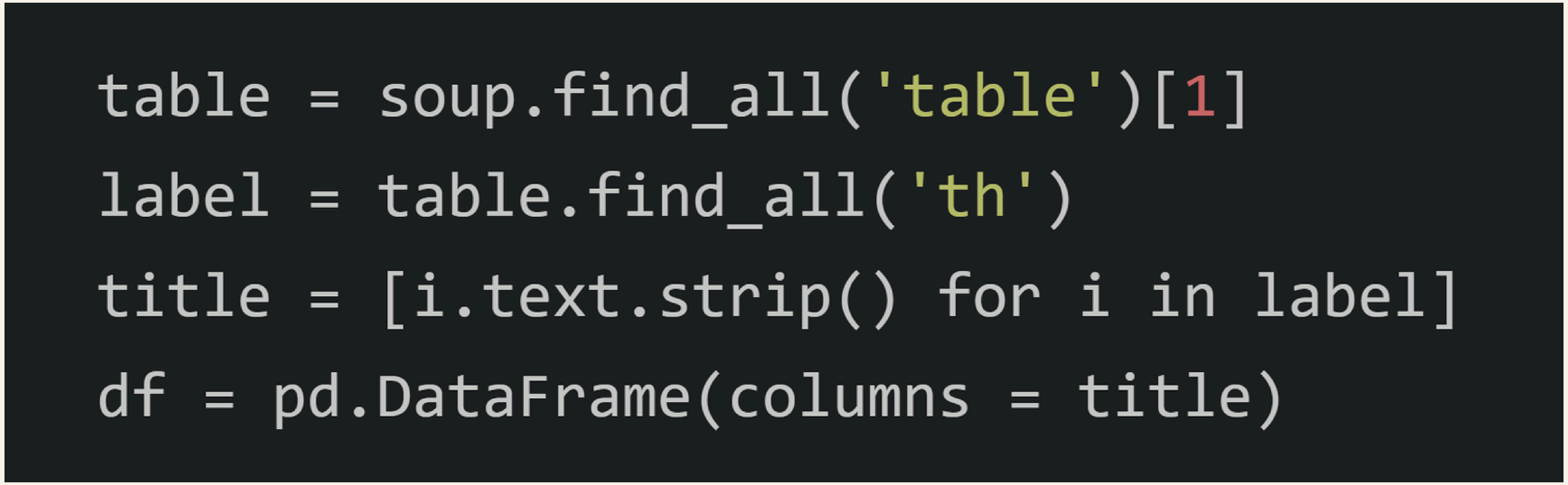


```
url = 'https://en.wikipedia.org/wiki/List_of_largest_technology_companies_by_revenue#2021_list'

page = requests.get(url)

soup = BeautifulSoup(page.text , 'html')
```

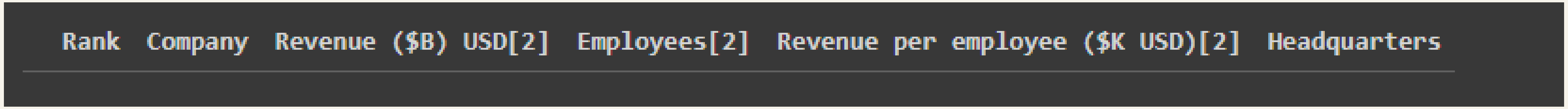
This code uses the Requests library to make an HTTP request and the BeautifulSoup library to parse the HTML content of a Wikipedia page that lists the top technology companies by revenue



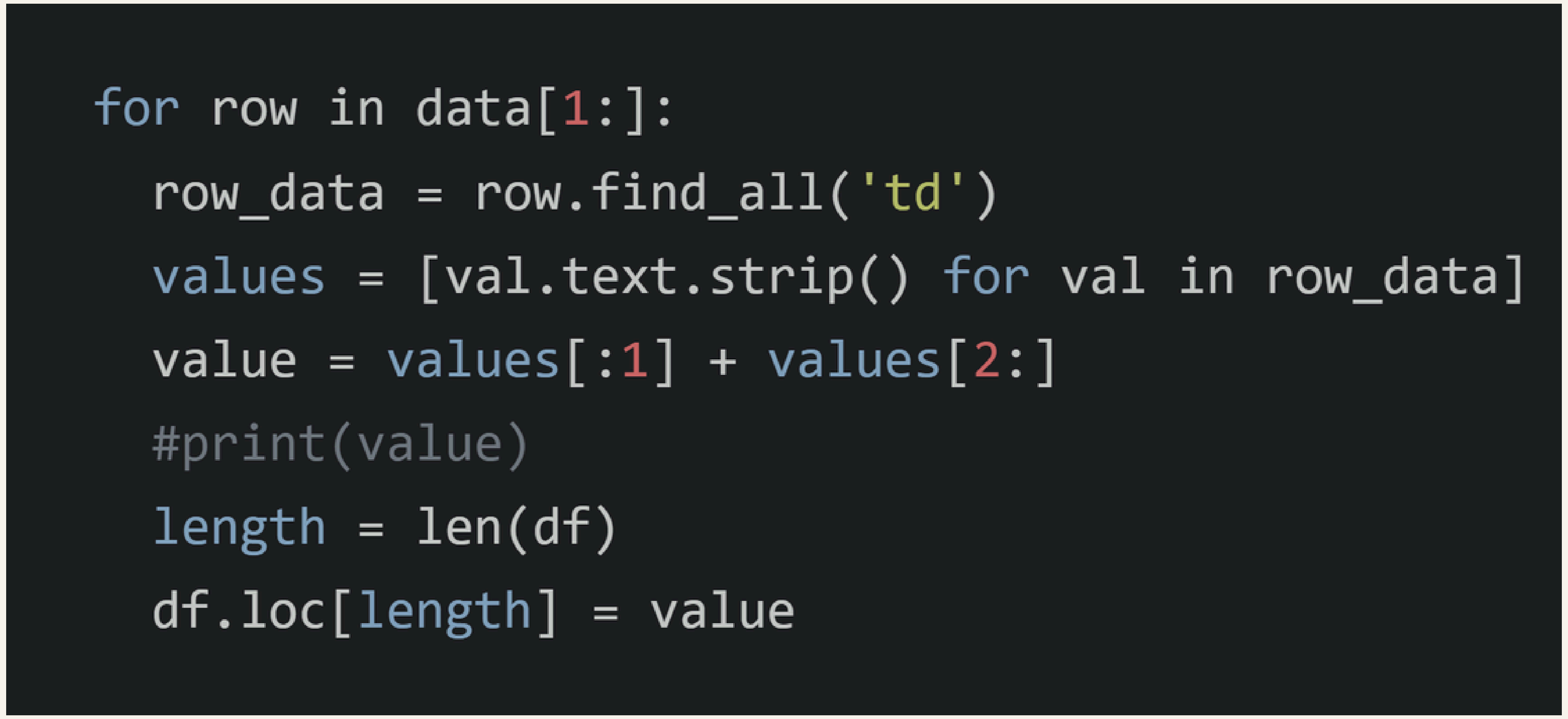
```
table = soup.find_all('table')[1]
label = table.find_all('th')
title = [i.text.strip() for i in label]
df = pd.DataFrame(columns = title)
```

This code finds the second table on a webpage and creates a pandas DataFrame using its column headers.

Output:



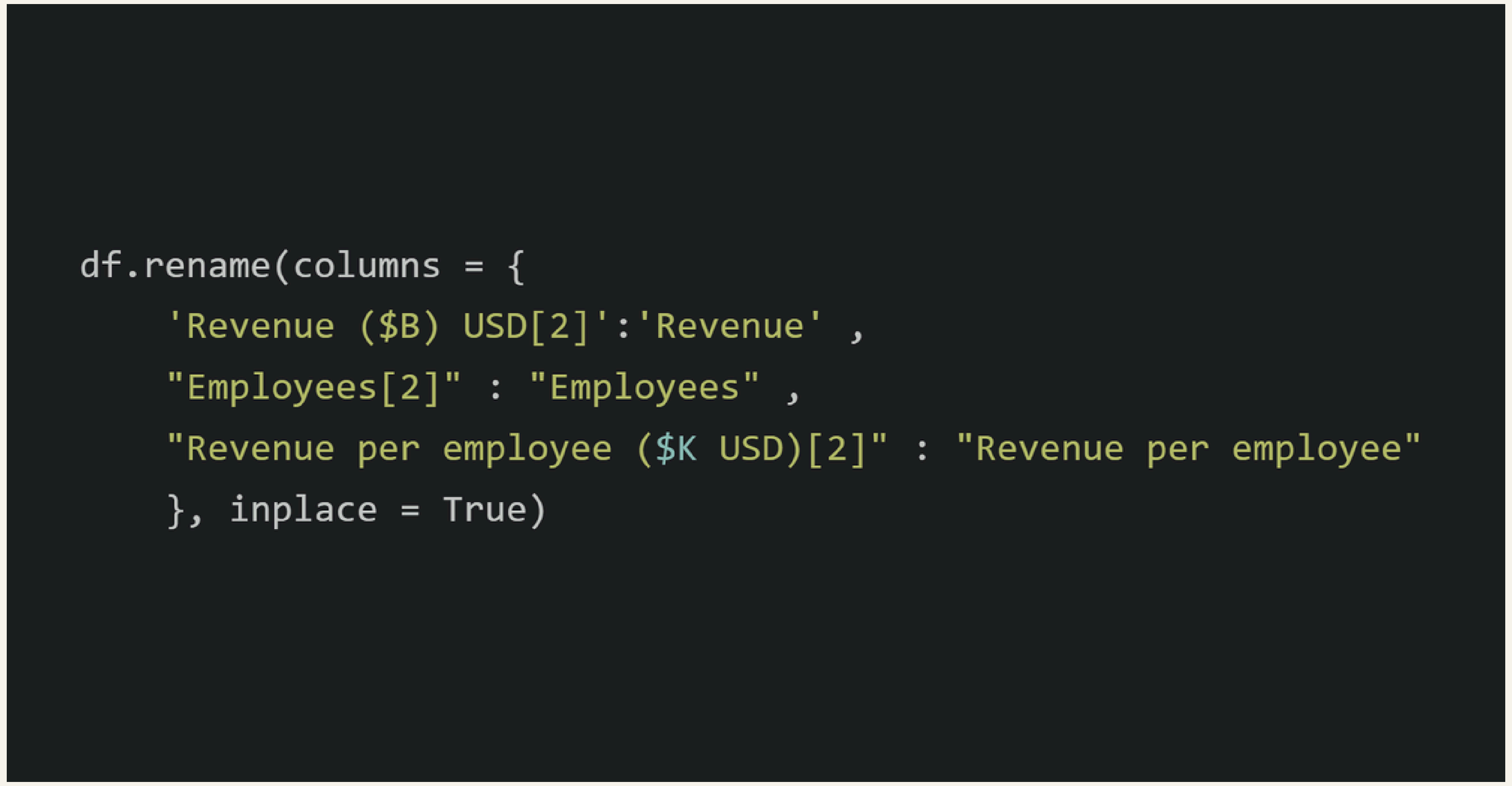
```
Rank  Company  Revenue ($B) USD[2]  Employees[2]  Revenue per employee ($K USD)[2]  Headquarters
```



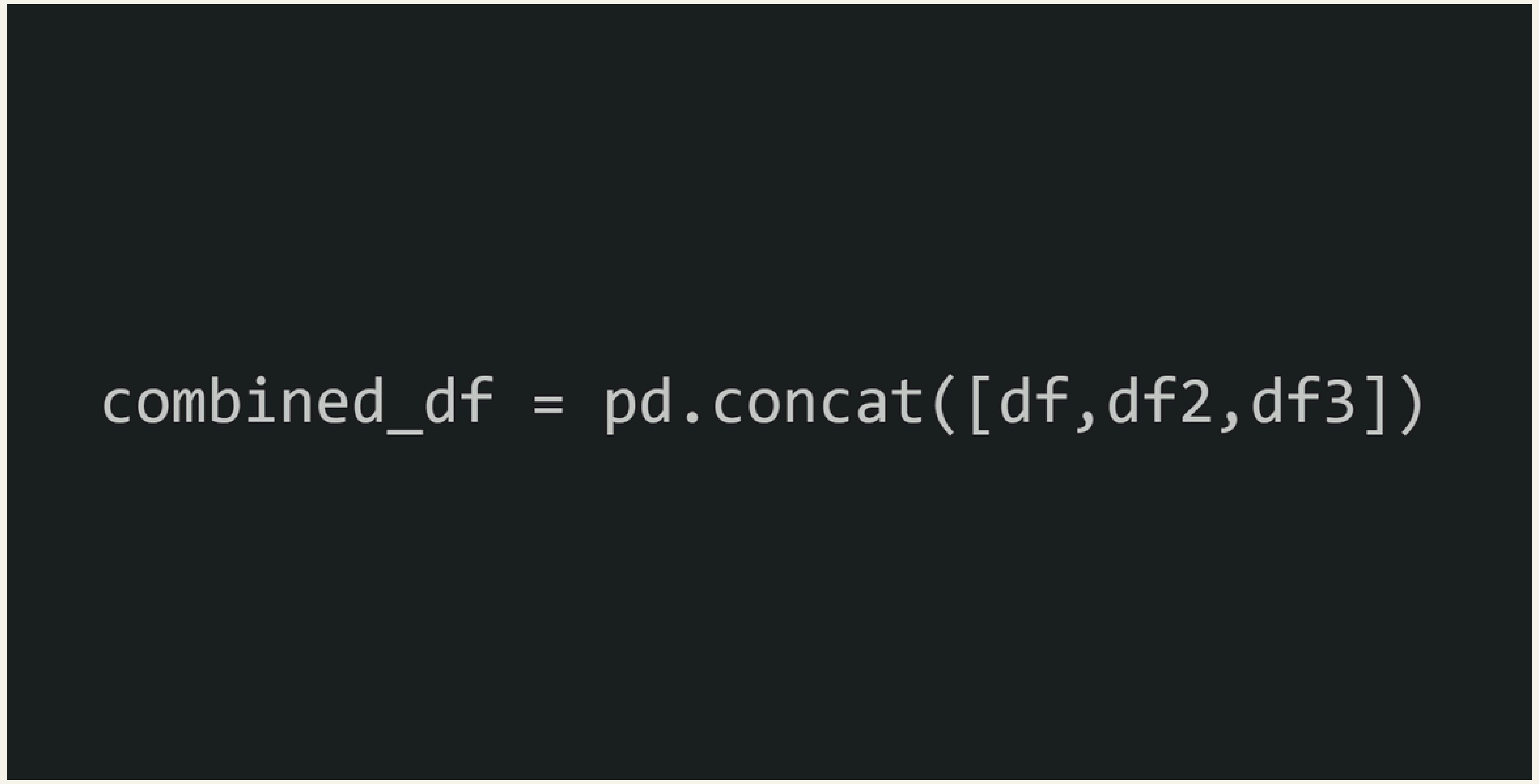
```
for row in data[1:]:
  row_data = row.find_all('td')
  values = [val.text.strip() for val in row_data]
  value = values[:1] + values[2:]
  #print(value)
  length = len(df)
  df.loc[length] = value
```

The code snippet processes table rows, excluding the second row that typically contains images or unrelated content, by extracting and formatting data from table cells before appending it to a pandas DataFrame for analysis or visualization.

Rename columns with meaningful names.



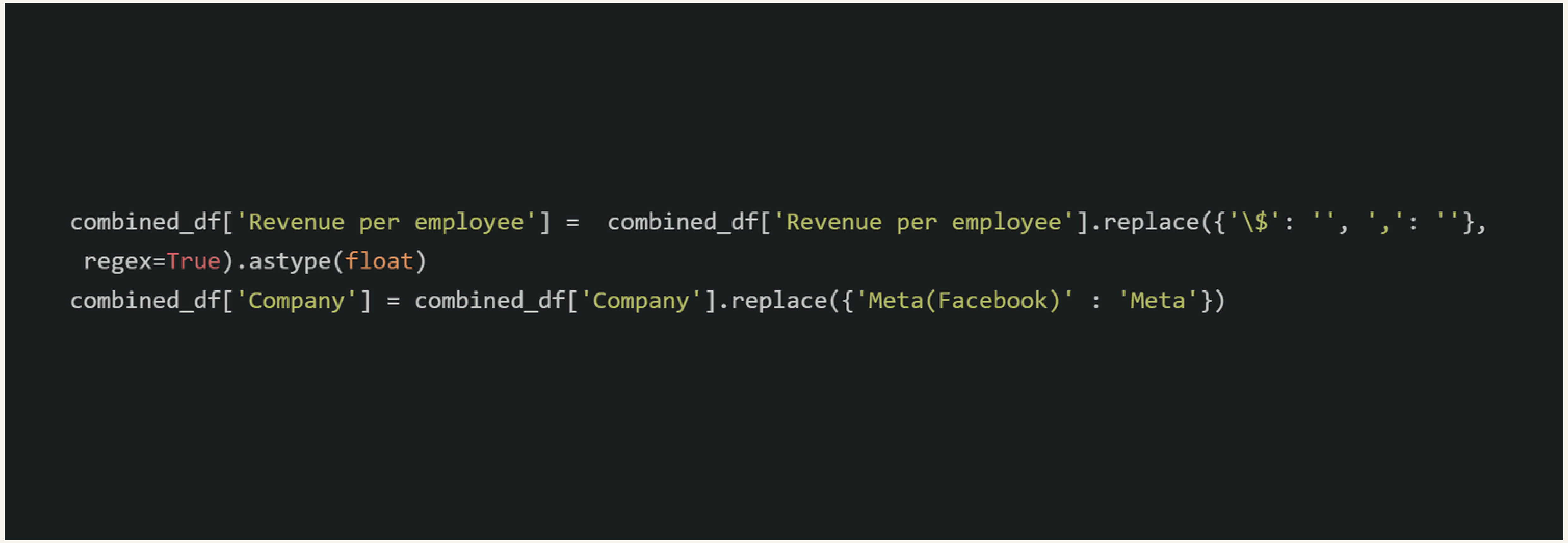
```
df.rename(columns = {
    'Revenue ($B) USD[2]':'Revenue' ,
    "Employees[2]" : "Employees" ,
    "Revenue per employee ($K USD)[2]" : "Revenue per employee"
    }, inplace = True)
```



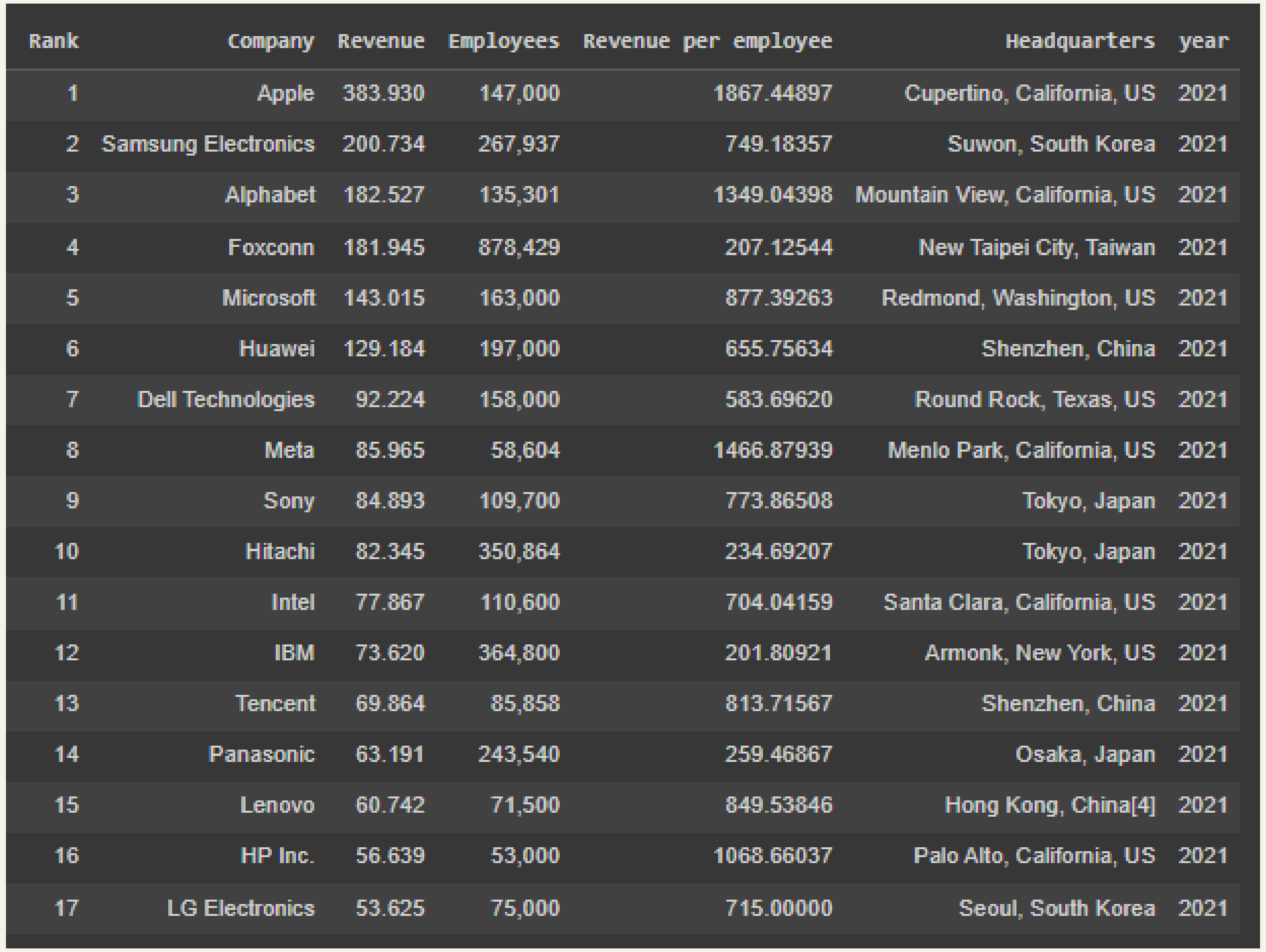
```
combined_df = pd.concat([df,df2,df3])
```

concatenates the DataFrames df, df2, and df3 into a single DataFrame, combined_df, merging data from multiple years into one unified dataset for comprehensive analysis.

The code cleans the 'Revenue per employee' column by removing symbols, converts it to float, and standardizes the 'Company' column for consistent analysis.



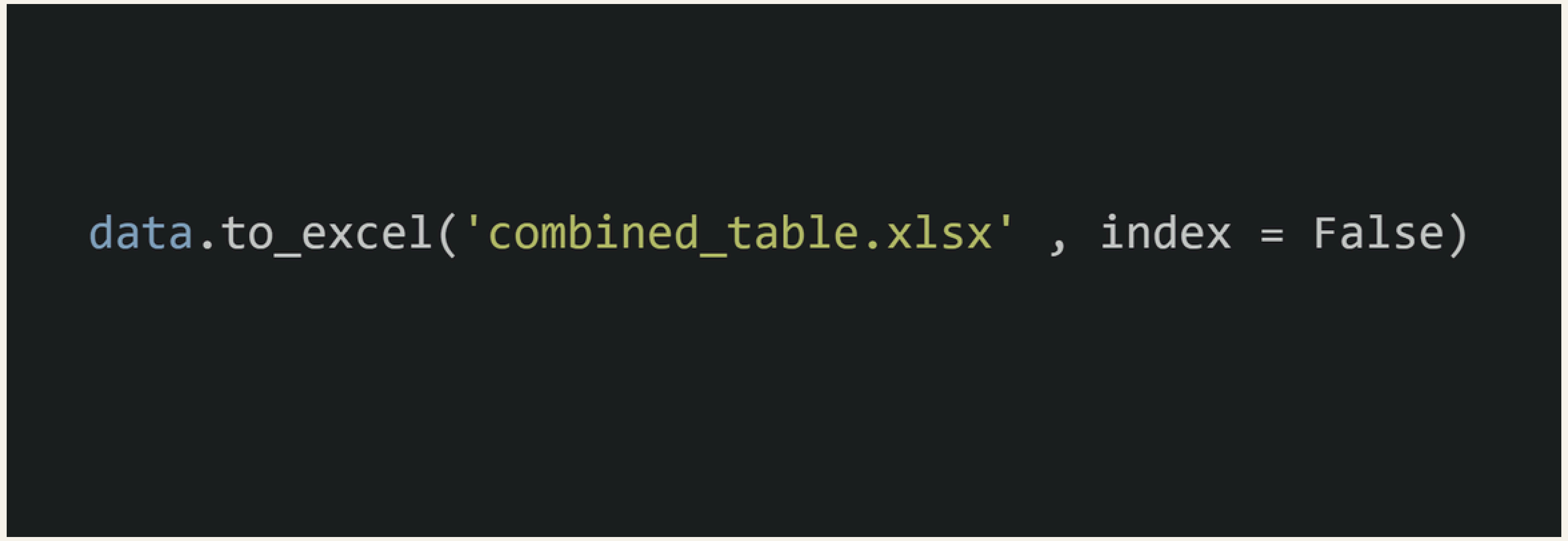
```
combined_df['Revenue per employee'] =  combined_df['Revenue per employee'].replace({'\$': '', ',': ''},
 regex=True).astype(float)
combined_df['Company'] = combined_df['Company'].replace({'Meta(Facebook)' : 'Meta'})
```



Rank	Company	Revenue	Employees	Revenue per employee	Headquarters	year
1	Apple	383.930	147,000	1867.44897	Cupertino, California, US	2021
2	Samsung Electronics	200.734	267,937	749.18357	Suwon, South Korea	2021
3	Alphabet	182.527	135,301	1349.04398	Mountain View, California, US	2021
4	Foxconn	181.945	878,429	207.12544	New Taipei City, Taiwan	2021
5	Microsoft	143.015	163,000	877.39263	Redmond, Washington, US	2021
6	Huawei	129.184	197,000	655.75634	Shenzhen, China	2021
7	Dell Technologies	92.224	158,000	583.69620	Round Rock, Texas, US	2021
8	Meta	85.965	58,604	1466.87939	Menlo Park, California, US	2021
9	Sony	84.893	109,700	773.86508	Tokyo, Japan	2021
10	Hitachi	82.345	350,864	234.69207	Tokyo, Japan	2021
11	Intel	77.867	110,600	704.04159	Santa Clara, California, US	2021
12	IBM	73.620	364,800	201.80921	Armonk, New York, US	2021
13	Tencent	69.864	85,858	813.71567	Shenzhen, China	2021
14	Panasonic	63.191	243,540	259.46867	Osaka, Japan	2021
15	Lenovo	60.742	71,500	849.53846	Hong Kong, China[4]	2021
16	HP Inc.	56.639	53,000	1068.66037	Palo Alto, California, US	2021
17	LG Electronics	53.625	75,000	715.00000	Seoul, South Korea	2021

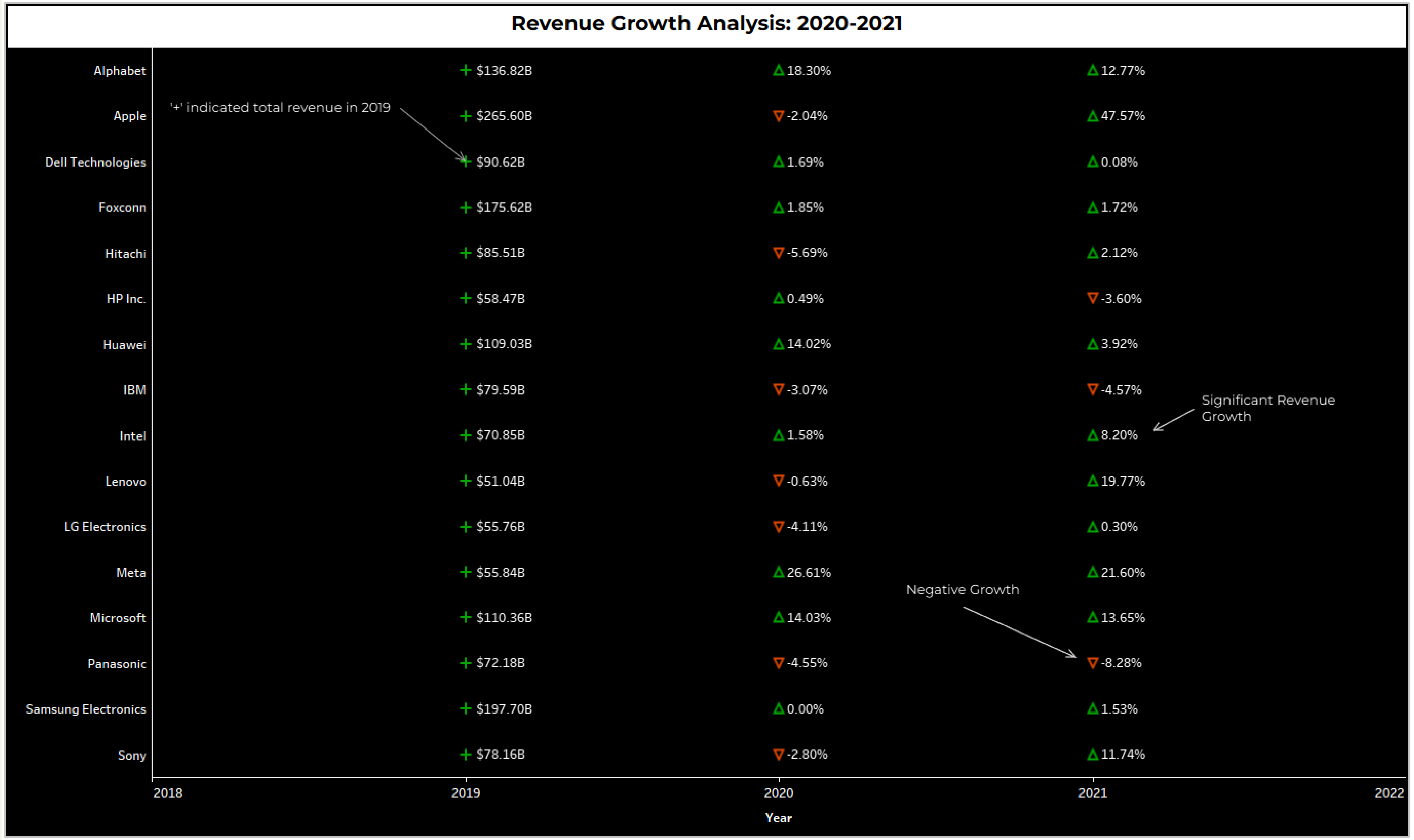
This table represents data for the year 2019, extracted using web scraping. The same approach was applied to extract tables for additional years from the same webpage, providing comparative insights into the companies' performance over time.

Save the DataFrame data to an Excel file named combined_table.xlsx without including the index column for storage or sharing.



```
data.to_excel('combined_table.xlsx' , index = False)
```

VISUAL DEPICTION



Revenue Growth Analysis: 2020-2021

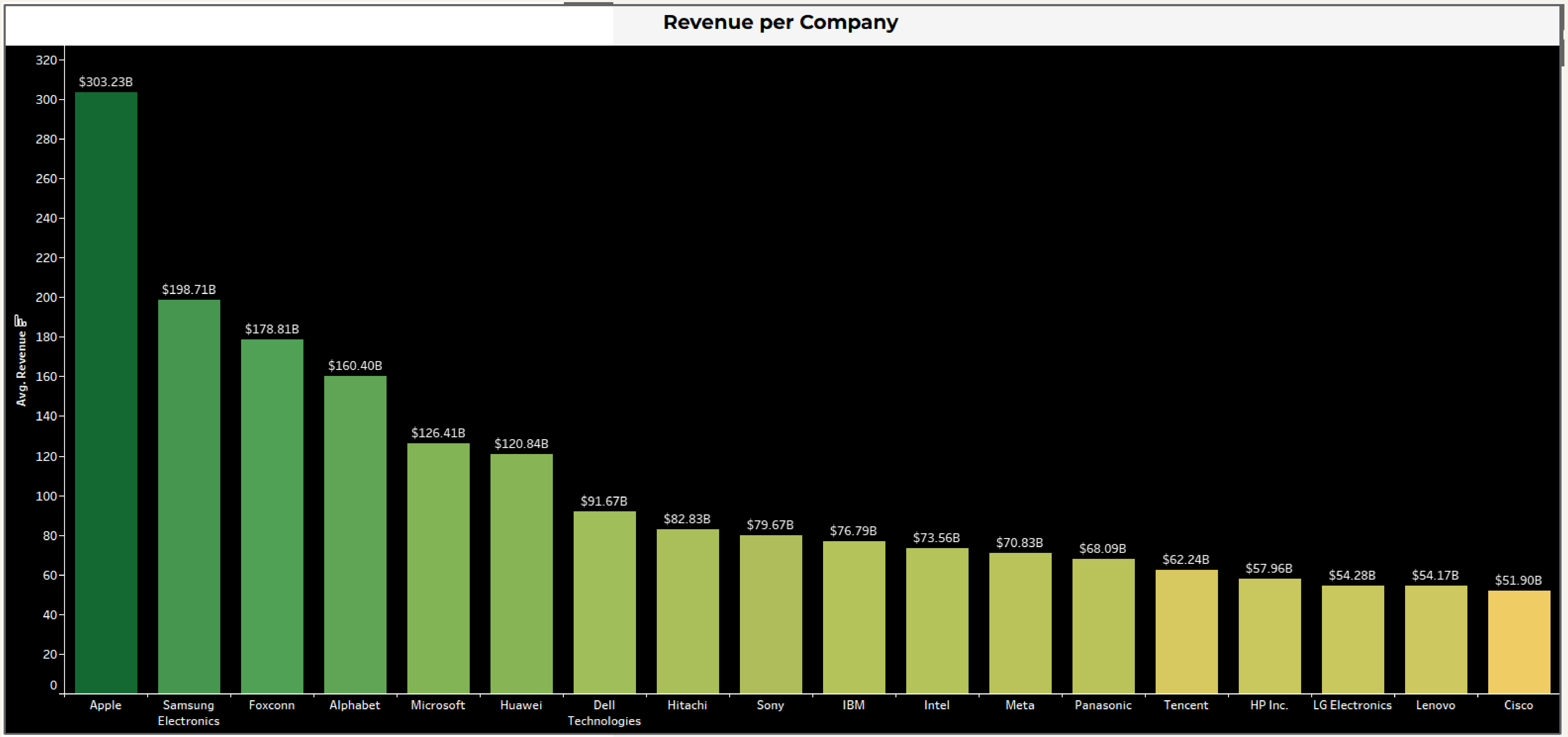
	2019	2020	2021
Alphabet	+ $136.82B	△ 18.30%	△ 12.77%
Apple	+ $265.60B	▽ -2.04%	△ 47.57%
Dell Technologies	+ $90.62B	△ 1.69%	△ 0.08%
Foxconn	+ $175.62B	△ 1.85%	△ 1.72%
Hitachi	+ $85.51B	▽ -5.69%	△ 2.12%
HP Inc.	+ $58.47B	△ 0.49%	▽ -3.60%
Huawei	+ $109.03B	△ 14.02%	△ 3.92%
IBM	+ $79.59B	▽ -3.07%	▽ -4.57%
Intel	+ $70.85B	△ 1.58%	△ 8.20%
Lenovo	+ $51.04B	▽ -0.63%	△ 19.77%
LG Electronics	+ $55.76B	▽ -4.11%	△ 0.30%
Meta	+ $55.84B	△ 26.61%	△ 21.60%
Microsoft	+ $110.36B	△ 14.03%	△ 13.65%
Panasonic	+ $72.18B	▽ -4.55%	▽ -8.28%
Samsung Electronics	+ $197.70B	△ 0.00%	△ 1.53%
Sony	+ $78.16B	▽ -2.80%	△ 11.74%

'+' indicated total revenue in 2019

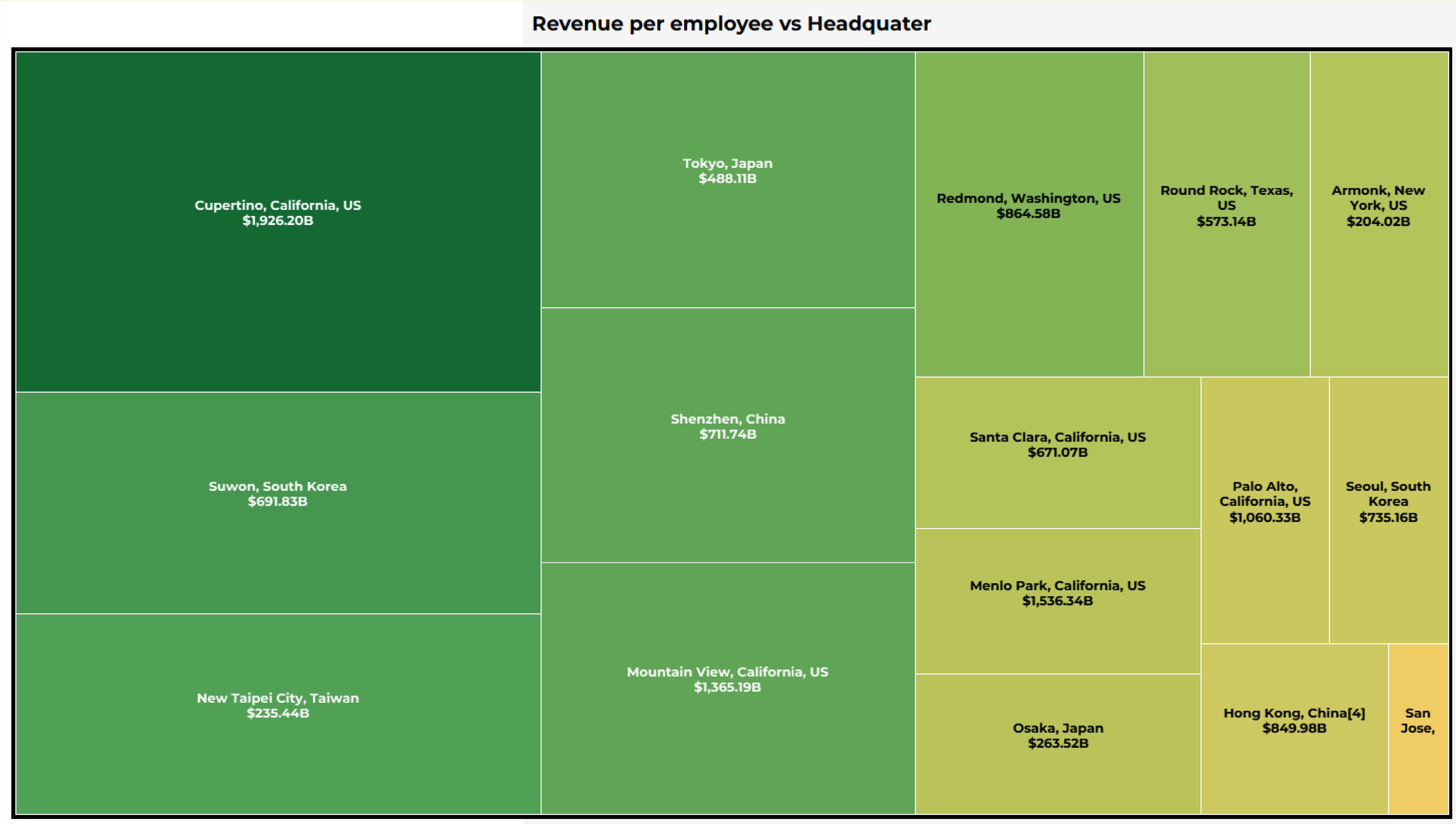
Significant Revenue Growth

Negative Growth

VISUAL DEPICTION



VISUAL DEPICTION



Revenue per employee vs Headquater

CONCLUSION

Web scraping the list of technology companies from Wikipedia provided valuable revenue insights from 2019 to 2021. This data-focused method highlights the resilience of technology and supports strategic decision-making.

The data not only highlights the growth and resilience of the technology sector but also emphasizes the importance of leveraging web scraping as a powerful tool for gathering actionable business intelligence.

THANK YOU

"From Scraping to Insights: Illuminating Data"

www.linkedin.com/in/ishank-verma