Analysis of the World's Largest Technology Companies by Revenue

Sai Rithwik Kukunuri

In this project, I analyze data on the largest technology companies in the world based on their annual revenue. The dataset comes from Wikipedia and includes 25 major tech companies like Apple, Amazon, Microsoft, Samsung, and Google.

The data contains important information about each company including their total revenue in billions of dollars, number of employees, country of origin, and headquarters location. I use this data to explore questions about company efficiency, regional differences, and the relationship between workforce size and revenue generation.

This analysis combines webscraping, data cleaning, statistical testing, and visualization to understand what makes these tech giants successful and how they compare to each other across different metrics.

Packages:

In [1]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
import scipy.stats
import numpy as np
from IPython.display import display, HTML

# Configure Plotly to work in both VS Code and Jupyter notebooks
import plotly.io as pio
pio.renderers.default = "plotly_mimetype+notebook+vscode"
```

Webscraping

I scraped data from Wikipedia's List of the Largest Technology Companies by Revenue. The table contains information about major tech companies like Apple, Samsung, Amazon, Microsoft, and Google Alphabet.

For each company, the webscraping process extracts the following information in this specific order:

1. Company name
2. Revenue in billions of USD
3. Profit in billions of USD
4. Number of employees
5. Country of origin
6. Headquarters location

The main continuous variables I focus on for analysis are revenue in billions of USD, number of employees, and a calculated efficiency metric (revenue per employee). These allow me to explore questions about company size, workforce efficiency, and regional differences.

The goal is to understand how company size, revenue, and efficiency are connected. I explore questions such as: Do larger companies use their employees more efficiently? How do hardware, software, and e-commerce companies compare? And how do FAANG companies differ from other major tech firms?

Using these variables, I can run correlations, t-tests, and regression models to identify patterns and examine how company size may predict revenue.

In [2]:
```
# Webscraping code for List of Largest Technology Companies by Revenue
headers = {
    "User-Agent": "ASU-Student-FinalProject/1.0 (contact: skukunu1@asu.edu)"
}

# Send request to Wikipedia page
page = requests.get(url="https://en.wikipedia.org/wiki/List_of_largest_technology_com
soup = BeautifulSoup(page.content, 'html.parser')

body = soup.find(id="mw-content-text")

all_tables = body.find_all("table", {"class": "wikitable"})

tech_table = all_tables[1]

company_list = []

rows = tech_table.find_all("tr")
for row in rows[1:]:
    cells = row.find_all(['th', 'td'])
    company_data = [cell.get_text(separator=" ", strip=True) for cell in cells]
    if len(company_data) > 0:
        company_list.append(company_data)

print(f"Total companies scraped: {len(company_list)}")

# Create DataFrame with proper column names in the specified order
df_display = pd.DataFrame(company_list, columns=[
    'Company',
    'Revenue_Billion_USD',
    'Profit_Billion_USD',
    'Employees',
    'Country',
    'Headquarters'
])
df_display.index = df_display.index + 1
df_display
```

```
Total companies scraped: 25
```

Out[2]:

	Company	Revenue_Billion_USD	Profit_Billion_USD	Employees	Country	Headquarters
1	Amazon	$637.9	$59.2	1,556,000	US	Seattle, Washington & Arlington, Virginia , US
2	Apple	$416.0	$112.0	166,100	US	Cupertino, California , US
3	Alphabet	$350.0	$100.1	190,167	US	Mountain View, California , US
4	Microsoft	$281.7	$101.8	228,000	US	Redmond, Washington , US
5	Samsung Electronics	$220.3	$25.3	270,372	South Korea	Suwon , South Korea
6	Foxconn	$213.9	$4.6	767,062	Taiwan	New Taipei City , Taiwan
7	Meta	$164.5	$62.4	78,450	US	Menlo Park, California, US
8	Jingdong	$158.8	$3.3	620,000	China	Beijing , China
9	Alibaba	$137.3	$17.8	124,320	China	Hangzhou , China & George Town , Cayman Islands
10	Nvidia	$130.5	$72.9	36,000	US	Sunnyvale, California, US
11	AT&T	$122.3	$50.2	140,990	US	Dallas , Texas , US
12	Huawei	$118.1	$8.5	208,000	China	Shenzhen , China
13	Deutsche Telekom	$112.0	$9.0	205,000	Germany	Bonn , Germany
14	Dell Technologies	$95.6	$4.6	108,000	US	Round Rock, Texas , US
15	Tencent	$91.8	$27.3	105,417	China	Shenzhen , China
16	Sony	$90.1	$6.7	113,000	Japan	Tokyo , Japan
17	TSMC	$88.3	$35.3	83,825	Taiwan	New Taipei City , Taiwan
18	Hitachi	$80.39	$3.8	322,525	Japan	Tokyo , Japan
19	Lenovo	$69.0	$1.5	72,000	Hong Kong / US	Hong Kong China / Morrisville, North Carolina ...
20	Accenture	$64.9	$7.4	774,000	Ireland / US	Dublin , Ireland / New York , US
21	IBM	$62.7	$6.0	270,300	US	Armonk, New York , US
22	LG Electronics	$57.7	$1.3	75,000	South Korea	Seoul , South Korea
23	Oracle	$57.4	$12.4	162,000	US	Austin, Texas, US
24	Cisco	$56.7	$10.1	86,200	US	San Jose, California, US
25	Panasonic	$55.3	$2.9	228,420	Japan	Kadoma, Osaka, Japan

Data Frame Creation and Cleaning

After scraping the table from Wikipedia, I turned the results into a Pandas DataFrame and gave each column a clear name, such as “Company,” “Revenue_Billion_USD,” and “Employees.” I cleaned the text by removing commas, dollar signs, extra spaces, and reference tags like [1] or [2], then converted the revenue and employee values from text into numeric types. I also checked for missing information and removed any rows that did not include key values, and I looked for duplicate company names to avoid repeating data. Once everything was cleaned, I created a “Revenue per Employee” column to measure each company’s efficiency. The main problems came from the inconsistent formatting on Wikipedia. Many numbers included dollar signs, commas, and reference brackets that had to be carefully removed before I could use them. Some companies were also missing employee data, so I had to drop those rows because I couldn’t calculate revenue per employee for them. Even though the process took time, cleaning the data was necessary to make sure the analysis would be accurate.

In [3]:
```
# Create DataFrame using Pandas from scraped data
tech_df = pd.DataFrame(company_list)
tech_df.index = tech_df.index + 1

# Display the raw DataFrame
tech_df.head()
```

Out[3]:

	0	1	2	3	4	5
1	Amazon	$637.9	$59.2	1,556,000	US	Seattle, Washington & Arlington, Virginia , US
2	Apple	$416.0	$112.0	166,100	US	Cupertino, California , US
3	Alphabet	$350.0	$100.1	190,167	US	Mountain View, California , US
4	Microsoft	$281.7	$101.8	228,000	US	Redmond, Washington , US
5	Samsung Electronics	$220.3	$25.3	270,372	South Korea	Suwon , South Korea

In [4]:
```
tech_df.columns = ['Company', 'Revenue_Billion_USD', 'Profit_Billion_USD', 'Employees

# Display DataFrame with column names
tech_df.head(10)
```

Out[4]:

	Company	Revenue_Billion_USD	Profit_Billion_USD	Employees	Country	Headquarters
1	Amazon	$637.9	$59.2	1,556,000	US	Seattle, Washington & Arlington, Virginia , US
2	Apple	$416.0	$112.0	166,100	US	Cupertino, California , US
3	Alphabet	$350.0	$100.1	190,167	US	Mountain View, California , US
4	Microsoft	$281.7	$101.8	228,000	US	Redmond, Washington , US
5	Samsung Electronics	$220.3	$25.3	270,372	South Korea	Suwon , South Korea
6	Foxconn	$213.9	$4.6	767,062	Taiwan	New Taipei City , Taiwan
7	Meta	$164.5	$62.4	78,450	US	Menlo Park, California, US
8	Jingdong	$158.8	$3.3	620,000	China	Beijing , China
9	Alibaba	$137.3	$17.8	124,320	China	Hangzhou , China & George Town , Cayman Islands
10	Nvidia	$130.5	$72.9	36,000	US	Sunnyvale, California, US

Step 1: Clean Special Characters

In this step, I remove all the messy formatting that Wikipedia adds to the data. This includes removing reference brackets like [1] and [2], dollar signs, commas from large numbers, and any extra spaces. I do this for the revenue column, employee column, and company names to get clean text that can be converted to numbers.

In [5]:
```
# Data Cleaning

# Clean Revenue column
tech_df['Revenue_Billion_USD'] = tech_df['Revenue_Billion_USD'].str.replace(r'\[.*?\]
tech_df['Revenue_Billion_USD'] = tech_df['Revenue_Billion_USD'].str.replace('$', '',
tech_df['Revenue_Billion_USD'] = tech_df['Revenue_Billion_USD'].str.replace(',', '',
tech_df['Revenue_Billion_USD'] = tech_df['Revenue_Billion_USD'].str.strip()

# Clean Employees column
tech_df['Employees'] = tech_df['Employees'].str.replace(r'\[.*?\]', '', regex=True)
tech_df['Employees'] = tech_df['Employees'].str.replace(',', '', regex=False)
tech_df['Employees'] = tech_df['Employees'].str.strip()

# Clean Company names
tech_df['Company'] = tech_df['Company'].str.replace(r'\[.*?\]', '', regex=True)
tech_df['Company'] = tech_df['Company'].str.strip()

# Display cleaned data
tech_df.head(10)
```

Out[5]:

	Company	Revenue_Billion_USD	Profit_Billion_USD	Employees	Country	Headquarters
1	Amazon	637.9	$59.2	1556000	US	Seattle, Washington & Arlington, Virginia , US
2	Apple	416.0	$112.0	166100	US	Cupertino, California , US
3	Alphabet	350.0	$100.1	190167	US	Mountain View, California , US
4	Microsoft	281.7	$101.8	228000	US	Redmond, Washington , US
5	Samsung Electronics	220.3	$25.3	270372	South Korea	Suwon , South Korea
6	Foxconn	213.9	$4.6	767062	Taiwan	New Taipei City , Taiwan
7	Meta	164.5	$62.4	78450	US	Menlo Park, California, US
8	Jingdong	158.8	$3.3	620000	China	Beijing , China
9	Alibaba	137.3	$17.8	124320	China	Hangzhou , China & George Town , Cayman Islands
10	Nvidia	130.5	$72.9	36000	US	Sunnyvale, California, US

Step 2: Handle Missing Data and Duplicates

Here I check for any companies that are missing important information like revenue or employee count, and I remove those rows since I can't analyze them properly. I also look for any duplicate company entries and keep only the first occurrence to avoid counting the same company twice.

In [6]:
```
# Remove rows with missing values in critical columns
print(f"Shape before removing NaN values: {tech_df.shape}")
tech_df = tech_df.dropna(subset=['Revenue_Billion_USD', 'Employees'])
print(f"Shape after removing NaN values: {tech_df.shape}")

# Remove duplicate companies
print(f"\nDuplicate companies found: {tech_df['Company'].duplicated().sum()}")
tech_df = tech_df.drop_duplicates(subset=['Company'], keep='first')
print(f"Shape after removing duplicates: {tech_df.shape}")

tech_df = tech_df.reset_index(drop=True)
tech_df.index = tech_df.index + 1
```

```
Shape before removing NaN values: (25, 6)
Shape after removing NaN values: (25, 6)

Duplicate companies found: 0
Shape after removing duplicates: (25, 6)
```

Step 3: Create Efficiency Metric

Now I calculate a new column called "Revenue per Employee" which shows how much revenue each company generates per worker. This is calculated by converting revenue from billions to millions, then dividing by the number of employees. This metric helps measure how efficiently each company uses its workforce.

In [7]:
```
# Convert cleaned string columns to numeric types
tech_df['Revenue_Billion_USD'] = pd.to_numeric(tech_df['Revenue_Billion_USD'], errors=
tech_df['Employees'] = pd.to_numeric(tech_df['Employees'], errors='coerce')

# Create new calculated column: Revenue per Employee (in millions)
tech_df['Revenue_per_Employee_Million'] = (tech_df['Revenue_Billion_USD'] * 1000) / te
tech_df['Revenue_per_Employee_Million'] = tech_df['Revenue_per_Employee_Million'].rou

print(f"Total companies in final dataset: {len(tech_df)}")

# Display final cleaned DataFrame
tech_df.head(15)
```

```
Total companies in final dataset: 25
```

Out[7]:

	Company	Revenue_Billion_USD	Profit_Billion_USD	Employees	Country	Headquarters	Revenue_per_Employe
1	Amazon	637.9	$59.2	1556000	US	Seattle, Washington & Arlington, Virginia , US	
2	Apple	416.0	$112.0	166100	US	Cupertino, California , US	
3	Alphabet	350.0	$100.1	190167	US	Mountain View, California , US	
4	Microsoft	281.7	$101.8	228000	US	Redmond, Washington , US	
5	Samsung Electronics	220.3	$25.3	270372	South Korea	Suwon , South Korea	
6	Foxconn	213.9	$4.6	767062	Taiwan	New Taipei City , Taiwan	
7	Meta	164.5	$62.4	78450	US	Menlo Park, California, US	
8	Jingdong	158.8	$3.3	620000	China	Beijing , China	
9	Alibaba	137.3	$17.8	124320	China	Hangzhou , China & George Town , Cayman Islands	
10	Nvidia	130.5	$72.9	36000	US	Sunnyvale, California, US	
11	AT&T	122.3	$50.2	140990	US	Dallas , Texas , US	
12	Huawei	118.1	$8.5	208000	China	Shenzhen , China	
13	Deutsche Telekom	112.0	$9.0	205000	Germany	Bonn , Germany	
14	Dell Technologies	95.6	$4.6	108000	US	Round Rock, Texas , US	
15	Tencent	91.8	$27.3	105417	China	Shenzhen , China	

Finally, I examine the summary statistics for the three numerical variables in the dataset. This includes measures like mean, median, standard deviation, and quartiles, which give me a comprehensive overview of the distribution and central tendencies of revenue, employee count, and revenue efficiency across all companies.

In [8]:
```
# Finally, display summary statistics for continuous variables
tech_df[['Revenue_Billion_USD', 'Employees', 'Revenue_per_Employee_Million']].describe
```

Out[8]:

	Revenue_Billion_USD	Employees	Revenue_per_Employee_Million
count	25.000000	2.500000e+01	25.00000
mean	157.327600	2.796459e+05	0.93200
std	137.335776	3.336151e+05	0.81936
min	55.300000	3.600000e+04	0.08400
25%	69.000000	1.054170e+05	0.35400
50%	112.000000	1.661000e+05	0.79700
75%	164.500000	2.703000e+05	1.05300
max	637.900000	1.556000e+06	3.62500

Data Visualization and Graphs

I will create five unique visualizations using three different libraries Matplotlib, Seaborn, and Plotly to explore various aspects of the technology company dataset.

Visualizations to be created:

1. **Bar Chart (Matplotlib)**: Top 10 companies by revenue - Shows the dominant players in the tech industry
2. **Scatter Plot (Seaborn)**: Revenue vs. Employees with regression line - Explores the relationship between company size and revenue
3. **Interactive Bar Chart (Plotly)**: Revenue per Employee by Company - Analyzes workforce efficiency across companies
4. **Box Plot (Seaborn)**: Distribution comparison between US and Non-US companies - Examines regional differences in revenue
5. **Interactive Bubble Chart (Plotly)**: Revenue, Employees, and Efficiency - Shows multidimensional relationships

These visualizations will help identify patterns, outliers, and relationships in the data that will inform our statistical analysis.

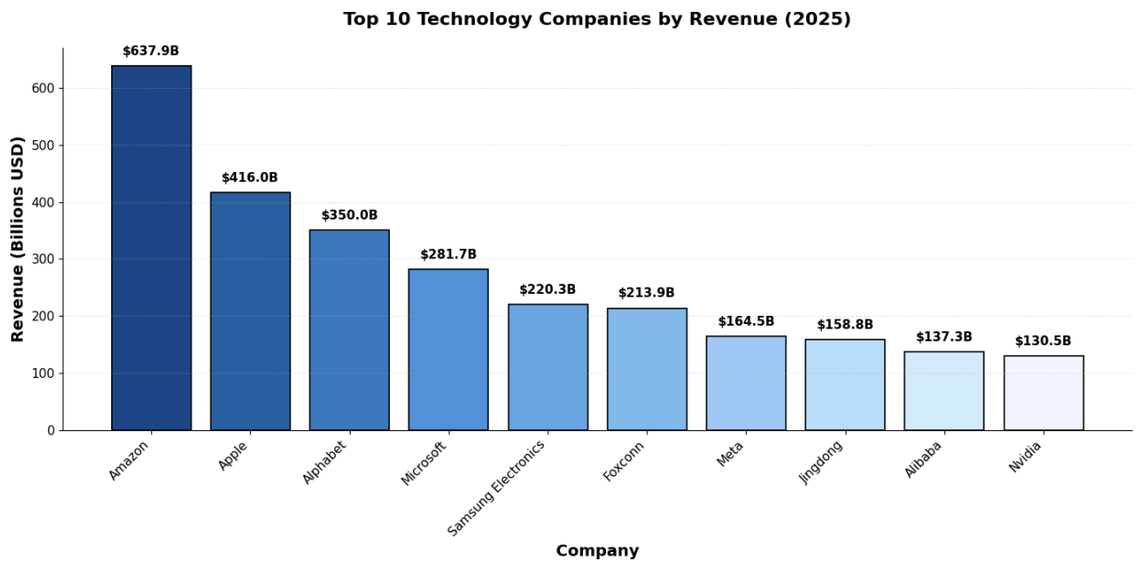
Graph 1: Top 10 Tech Companies by Revenue

This graph is created using **Matplotlib**. This bar chart displays the top 10 technology companies ranked by their annual revenue in billions of USD. This visualization is ideal for quickly identifying the industry leaders and understanding the scale differences between the largest tech companies.

Key Observations:

- Amazon leads with nearly $638 billion in revenue, significantly ahead of other companies
- The FAANG companies (Facebook/Meta, Apple, Amazon, Netflix excluded, Google/Alphabet) dominate the top positions
- There's a steep drop-off after the top 4 companies (Amazon, Apple, Alphabet, Microsoft)

In [9]:
```
# Graph 1: Matplotlib Bar Chart – Top 10 Companies by Revenue

# Get the top 10 companies
top_10 = tech_df.nlargest(10, 'Revenue_Billion_USD')

# Create a nice color gradient from dark to light blue
colors = ['#1f4788', '#2962a3', '#3c7abe', '#5192d9', '#6ba7e4',
          '#85bcef', '#a0ccf5', '#bbdcfa', '#d6ebfd', '#f0f7ff']

# Create the figure
plt.figure(figsize=(14, 7))

# Create bars
bars = plt.bar(top_10['Company'], top_10['Revenue_Billion_USD'],
               color=colors, edgecolor='black', linewidth=1.2)

# Add revenue labels on top of each bar
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 15,
             f'${height:.1f}B',
             ha='center', va='bottom', fontsize=11, fontweight='bold')

# Customize the plot
plt.xlabel('Company', fontsize=14, fontweight='bold')
plt.ylabel('Revenue (Billions USD)', fontsize=14, fontweight='bold')
plt.title('Top 10 Technology Companies by Revenue (2025)',
          fontsize=16, fontweight='bold', pad=20)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right', fontsize=11)
plt.yticks(fontsize=11)

# Add a light grid for easier reading
plt.grid(axis='y', alpha=0.3, linestyle='--', linewidth=0.7)

# Remove top and right spines for cleaner look
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)

plt.tight_layout()
plt.show()

# Print the top 10 list
print("\nTop 10 Companies by Revenue:")
print(top_10[['Company', 'Revenue_Billion_USD', 'Country']].to_string(index=False))
```




```
Top 10 Companies by Revenue:
            Company  Revenue_Billion_USD     Country
             Amazon                637.9          US
              Apple                416.0          US
           Alphabet                350.0          US
          Microsoft                281.7          US
Samsung Electronics                220.3 South Korea
            Foxconn                213.9      Taiwan
               Meta                164.5          US
           Jingdong                158.8       China
            Alibaba                137.3       China
             Nvidia                130.5          US
```

Graph 2: Revenue vs. Employees Scatter Plot

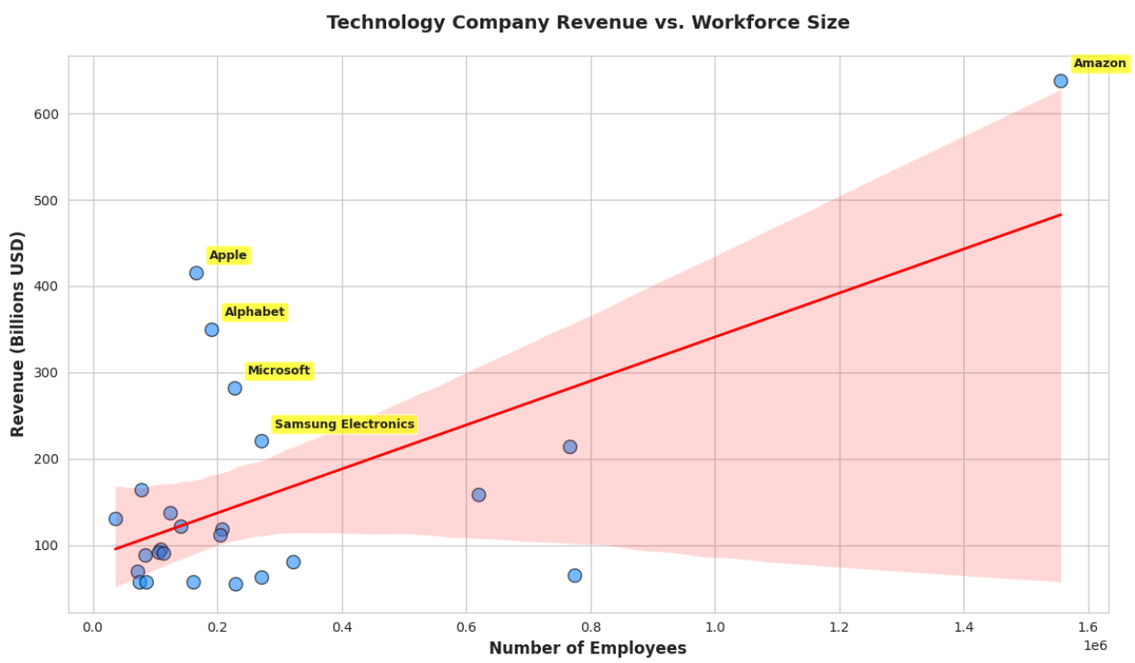
This graph is created using **Seaborn**. This scatter plot with a regression line explores the relationship between the number of employees and company revenue. Each point represents a technology company, and the regression line shows the general trend.

Why this visualization? This chart helps us understand whether larger workforces correlate with higher revenues, or if some companies achieve high revenue with lean operations.

Key Observations:

- Positive correlation between employees and revenue
- Some outliers like Amazon show high revenue with massive workforce
- Companies like Apple and Alphabet achieve high revenue with relatively fewer employees, indicating high efficiency

In [10]:
```
# Graph 2: Seaborn Scatter Plot
plt.figure(figsize=(12, 7))
sns.set_style("whitegrid")

# Create scatter plot with regression line
ax = sns.regplot(x='Employees', y='Revenue_Billion_USD', data=tech_df,
                 scatter_kws={'s': 100, 'alpha': 0.6, 'edgecolors': 'black'},
                 line_kws={'color': 'red', 'linewidth': 2},
                 color='dodgerblue')

for idx, row in tech_df.nlargest(5, 'Revenue_Billion_USD').iterrows():
    plt.annotate(row['Company'],
                 xy=(row['Employees'], row['Revenue_Billion_USD']),
                 xytext=(10, 10), textcoords='offset points',
                 fontsize=9, fontweight='bold',
                 bbox=dict(boxstyle='round,pad=0.3', facecolor='yellow', alpha=0.7))

plt.xlabel('Number of Employees', fontsize=12, fontweight='bold')
plt.ylabel('Revenue (Billions USD)', fontsize=12, fontweight='bold')
plt.title('Technology Company Revenue vs. Workforce Size', fontsize=14, fontweight='b
plt.tight_layout()
plt.show()

correlation = tech_df['Employees'].corr(tech_df['Revenue_Billion_USD'])
print(f"\nCorrelation between Employees and Revenue: {correlation:.3f}")
```




```
Correlation between Employees and Revenue: 0.619
```

Graph 3: Interactive Bar Chart - Revenue per Employee

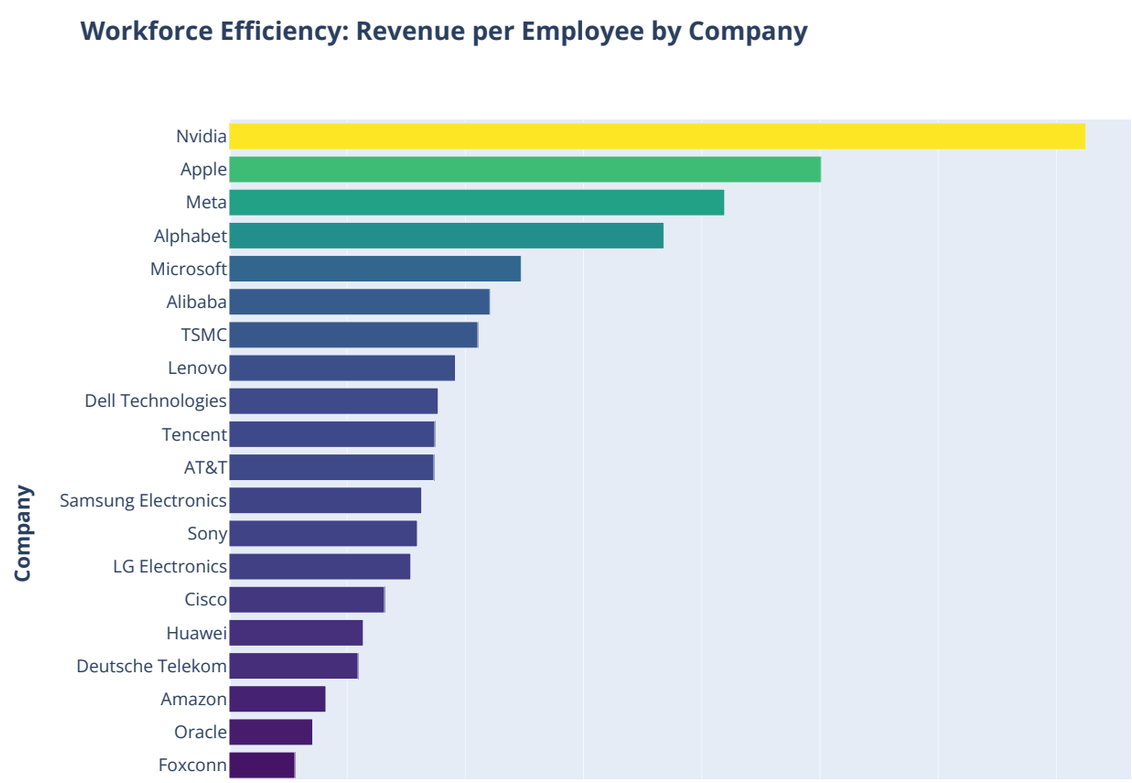
This graph is created using **Plotly**. This interactive bar chart shows the revenue per employee (in millions USD) for each company, sorted from highest to lowest efficiency. This metric reveals which companies generate the most revenue per worker.

Why this visualization? Revenue per employee is a key efficiency metric in business analysis. Higher values indicate better productivity and operational efficiency. The interactive nature allows hovering over bars to see exact values.

Key Observations:

- Nvidia leads with an astounding $7.97M per employee, reflecting its high-margin semiconductor business - Apple and Alphabet also sho$ 2.5M+ per employee
- Labor-intensive companies like Foxconn and Accenture show much lower ratios (around $0.08M$–0.28M per employee)

In [11]:
```
# Graph 3: Plotly Interactive Bar Chart
sorted_df = tech_df.sort_values('Revenue_per_Employee_Million', ascending=True)

fig = px.bar(sorted_df,
             x='Revenue_per_Employee_Million',
             y='Company',
             orientation='h',
             color='Revenue_per_Employee_Million',
             color_continuous_scale='viridis',
             labels={'Revenue_per_Employee_Million': 'Revenue per Employee (Million US
                     'Company': 'Company'},
             title='<b>Workforce Efficiency: Revenue per Employee by Company</b>',
             hover_data={'Company': True,
                         'Revenue_per_Employee_Million': ':.3f',
                         'Employees': ':,',
                         'Revenue_Billion_USD': ':.1f'})

fig.update_layout(
    height=700,
    font=dict(size=11),
    title_font=dict(size=16),
    xaxis_title='<b>Revenue per Employee (Million USD)</b>',
    yaxis_title='<b>Company</b>',
    showlegend=False
)

fig.show()

print("\nTop 5 Most Efficient Companies (Revenue per Employee):")
print(tech_df.nlargest(5, 'Revenue_per_Employee_Million')[['Company', 'Revenue_per_Emp
```




```
Top 5 Most Efficient Companies (Revenue per Employee):
  Company  Revenue_per_Employee_Million  Employees
   Nvidia                        3.625      36000
    Apple                        2.505     166100
     Meta                        2.097      78450
 Alphabet                        1.840     190167
Microsoft                        1.236     228000
```

Graph 4: US vs. Non-US Companies Revenue Distribution

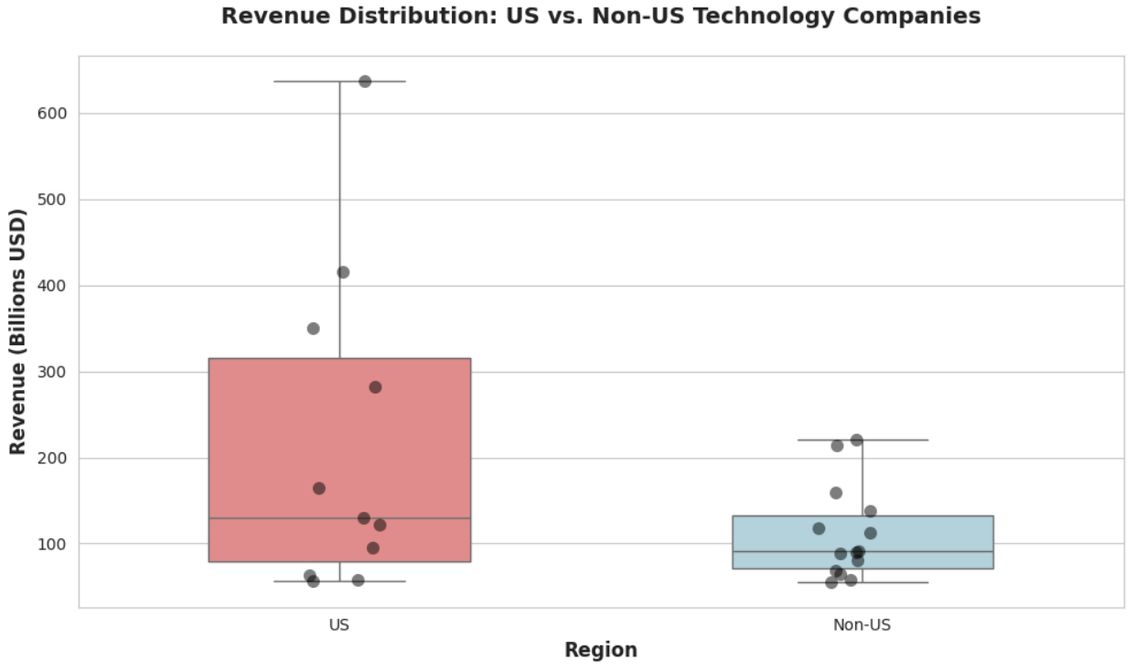
This graph is created using **Seaborn**. This box plot compares the revenue distribution between US-based technology companies and those headquartered in other countries. Box plots are excellent for showing the distribution, median, quartiles, and potential outliers.

Why this visualization? This comparison helps us understand if US companies dominate the tech sector in terms of revenue scale, and whether there are significant differences in the typical size of tech companies by region.

Key Observations:

- US companies show higher median revenue and greater variability
- The US has several high-revenue outliers (Amazon, Apple, Alphabet, Microsoft)
- Non-US companies are generally more clustered with lower median values
- This suggests US tech companies tend to be larger and more dominant globally

In [12]:
```
# Graph 4: Seaborn Box Plot
# Create a new column for US vs Non-US
tech_df['Region'] = tech_df['Country'].apply(lambda x: 'US' if x == 'US' else 'Non-US

plt.figure(figsize=(10, 6))
sns.set_style("whitegrid")

# Create box plot with hue parameter to avoid warning
box_plot = sns.boxplot(x='Region', y='Revenue_Billion_USD', data=tech_df,
                       hue='Region', palette={'US': 'lightcoral', 'Non-US': 'lightblue
                       width=0.5, legend=False)

sns.stripplot(x='Region', y='Revenue_Billion_USD', data=tech_df,
              color='black', alpha=0.5, size=8)

plt.xlabel('Region', fontsize=12, fontweight='bold')
plt.ylabel('Revenue (Billions USD)', fontsize=12, fontweight='bold')
plt.title('Revenue Distribution: US vs. Non-US Technology Companies', fontsize=14, for
plt.tight_layout()
plt.show()

print("\nRevenue Statistics by Region:")
print(tech_df.groupby('Region')['Revenue_Billion_USD'].describe()[['count', 'mean', '
```




```
Revenue Statistics by Region:
        count        mean         std   min     50%    max
Region
Non-US   14.0  111.277857   53.817399  55.3   90.95  220.3
US       11.0  215.936364  186.424715  56.7  130.50  637.9
```

Graph 5: Interactive Bubble Chart - Revenue, Employees & Efficiency

This graph is created using **Plotly**. This dynamic bubble chart visualizes three dimensions simultaneously: revenue (x-axis), number of employees (y-axis), and efficiency (bubble size and color). Each bubble represents a company, creating an intuitive way to explore the relationships between company scale, workforce, and productivity.

Why this visualization? Bubble charts excel at showing multidimensional data in an easy-to-understand format. They reveal:

- The relationship between revenue and workforce size
- Which companies are efficiency outliers
- Regional patterns in business models

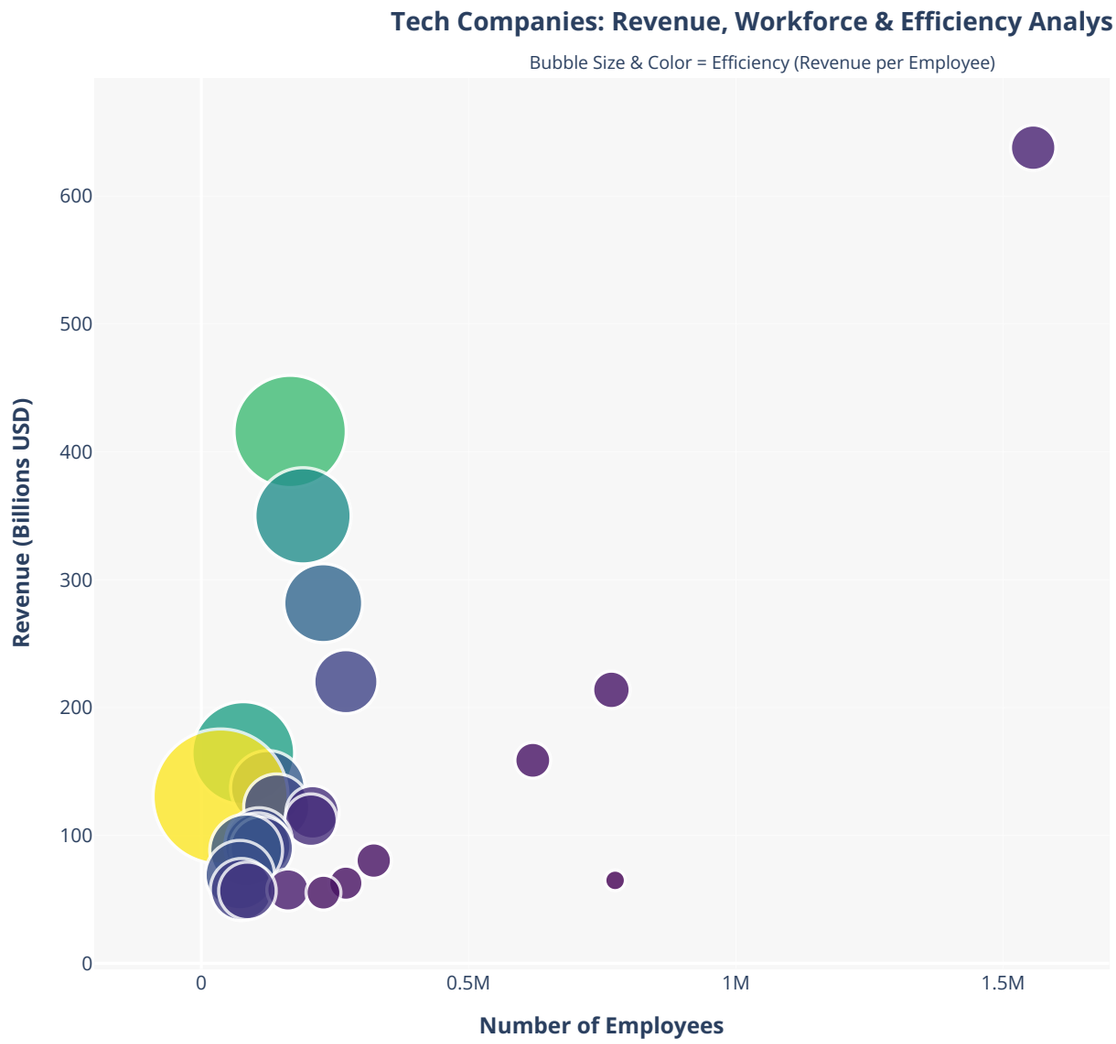
Interactive Features:

- Hover over bubbles to see complete company details
- Larger bubbles = higher revenue per employee (more efficient)
- Color gradient shows efficiency levels (green = efficient, red = less efficient)
- Easy to spot patterns and outliers

Key Observations:

- Strong positive correlation between employees and revenue (upward trend)
- Nvidia and Apple are efficiency leaders (large bubbles, high position)
- Amazon is massive in scale but lower in efficiency (small bubble despite high revenue)
- US companies (blue) dominate the upper-right quadrant (high revenue, large workforce)

In [13]:
```
# Graph 5: Interactive Bubble Chart

fig = px.scatter(tech_df,
                 x='Employees',
                 y='Revenue_Billion_USD',
                 size='Revenue_per_Employee_Million',
                 color='Revenue_per_Employee_Million',
                 hover_name='Company',
                 hover_data={
                     'Employees': ':,',
                     'Revenue_Billion_USD': ':.1f',
                     'Revenue_per_Employee_Million': ':.3f',
                     'Country': True
                 },
                 color_continuous_scale='Viridis',
                 size_max=60,
                 labels={
                     'Employees': 'Number of Employees',
                     'Revenue_Billion_USD': 'Revenue (Billions USD)',
                     'Revenue_per_Employee_Million': 'Revenue per Employee (Million US
                 })

fig.update_layout(
    title={
        'text': '<b>Tech Companies: Revenue, Workforce & Efficiency Analysis</b><br><s
        'x': 0.5,
        'xanchor': 'center',
        'font': {'size': 16}
    },
    xaxis_title='<b>Number of Employees</b>',
    yaxis_title='<b>Revenue (Billions USD)</b>',
    height=700,
    width=1000,
    showlegend=True,
    hovermode='closest',
    plot_bgcolor='rgba(240,240,240,0.5)',
    font=dict(size=12)
)

fig.update_xaxes(showgrid=True, gridwidth=1, gridcolor='white')
fig.update_yaxes(showgrid=True, gridwidth=1, gridcolor='white')

fig.update_traces(
    marker=dict(
        line=dict(width=2, color='white'),
        opacity=0.8
    )
)

fig.show()

print("\nEfficiency Analysis:")
print("\nTop 5 Most Efficient Companies:")
top_eff = tech_df.nlargest(5, 'Revenue_per_Employee_Million')[['Company', 'Revenue_Bi
print(top_eff.to_string(index=False))

print("\n\nLargest Companies by Revenue:")
top_rev = tech_df.nlargest(5, 'Revenue_Billion_USD')[['Company', 'Revenue_Billion_USD
print(top_rev.to_string(index=False))
```



Efficiency Analysis:

Top 5 Most Efficient Companies:

Company	Revenue_Billion_USD	Employees	Revenue_per_Employee_Million
Nvidia	130.5	36000	3.625
Apple	416.0	166100	2.505
Meta	164.5	78450	2.097
Alphabet	350.0	190167	1.840
Microsoft	281.7	228000	1.236

Largest Companies by Revenue:

Company	Revenue_Billion_USD	Employees	Revenue_per_Employee_Million
Amazon	637.9	1556000	0.410
Apple	416.0	166100	2.505
Alphabet	350.0	190167	1.840
Microsoft	281.7	228000	1.236
Samsung Electronics	220.3	270372	0.815

In [14]:
```
print("COMPLETE DATASET - ALL 25 TECHNOLOGY COMPANIES (Sorted by Revenue)\n")

display_df = tech_df[['Company', 'Country', 'Revenue_Billion_USD', 'Employees', 'Rever
display_df = display_df.sort_values('Revenue_Billion_USD', ascending=False).reset_inde
display_df.index = display_df.index + 1
display_df
```

```
COMPLETE DATASET - ALL 25 TECHNOLOGY COMPANIES (Sorted by Revenue)
```

Out[14]:

	Company	Country	Revenue_Billion_USD	Employees	Revenue_per_Employee_Million
1	Amazon	US	637.90	1556000	0.410
2	Apple	US	416.00	166100	2.505
3	Alphabet	US	350.00	190167	1.840
4	Microsoft	US	281.70	228000	1.236
5	Samsung Electronics	South Korea	220.30	270372	0.815
6	Foxconn	Taiwan	213.90	767062	0.279
7	Meta	US	164.50	78450	2.097
8	Jingdong	China	158.80	620000	0.256
9	Alibaba	China	137.30	124320	1.104
10	Nvidia	US	130.50	36000	3.625
11	AT&T	US	122.30	140990	0.867
12	Huawei	China	118.10	208000	0.568
13	Deutsche Telekom	Germany	112.00	205000	0.546
14	Dell Technologies	US	95.60	108000	0.885
15	Tencent	China	91.80	105417	0.871
16	Sony	Japan	90.10	113000	0.797
17	TSMC	Taiwan	88.30	83825	1.053
18	Hitachi	Japan	80.39	322525	0.249
19	Lenovo	Hong Kong / US	69.00	72000	0.958
20	Accenture	Ireland / US	64.90	774000	0.084
21	IBM	US	62.70	270300	0.232
22	LG Electronics	South Korea	57.70	75000	0.769
23	Oracle	US	57.40	162000	0.354
24	Cisco	US	56.70	86200	0.658
25	Panasonic	Japan	55.30	228420	0.242

Statistical Analysis

In this section, I will perform comprehensive statistical analysis on the technology company dataset, including descriptive statistics and hypothesis testing.

Analysis Components:

1. **Descriptive Statistics**: Calculate means, standard deviations, variances, and other univariate statistics for key variables
2. **Hypothesis Testing**: Conduct an independent samples t-test to compare US vs. Non-US companies
3. **Linear Regression**: Examine the relationship between employees and revenue with statistical significance testing
4. **Future Testing Recommendations**: Discuss additional analyses that could provide further insights

Descriptive Statistics - Univariate Analysis

Below is a comprehensive table of descriptive statistics for the three main continuous variables in our dataset: Revenue, Employees, and Revenue per Employee. These statistics provide insight into the central tendency, spread, and distribution of each variable.

In [15]:
```
# Descriptive Statistics for Continuous Variables
variables = ['Revenue_Billion_USD', 'Employees', 'Revenue_per_Employee_Million']

# Calculate statistics for each variable
stats_dict = {}
for var in variables:
    stats_dict[var] = {
        'Mean': tech_df[var].mean(),
        'Median': tech_df[var].median(),
        'Std Dev': tech_df[var].std(),
        'Variance': tech_df[var].var(),
        'Min': tech_df[var].min(),
        'Max': tech_df[var].max(),
        'Range': tech_df[var].max() - tech_df[var].min(),
        'Q1 (25%)': tech_df[var].quantile(0.25),
        'Q3 (75%)': tech_df[var].quantile(0.75),
        'IQR': tech_df[var].quantile(0.75) - tech_df[var].quantile(0.25)
    }

stats_df = pd.DataFrame(stats_dict).T
stats_df = stats_df.round(3)

print("DESCRIPTIVE STATISTICS FOR CONTINUOUS VARIABLES")
print(stats_df)
print()
print("Key Insights:")
print(f"• Average tech company revenue: ${stats_dict['Revenue_Billion_USD']['Mean']:.2
print(f"• Average workforce size: {stats_dict['Employees']['Mean']:,.0f} employees")
print(f"• Average revenue per employee: ${stats_dict['Revenue_per_Employee_Million'][
print(f"• Revenue shows high variability (Std Dev: ${stats_dict['Revenue_Billion_USD']
print(f"• Efficiency varies widely (Range: ${stats_dict['Revenue_per_Employee_Million
```

```
DESCRIPTIVE STATISTICS FOR CONTINUOUS VARIABLES
                                  Mean      Median     Std Dev  \
Revenue_Billion_USD             157.328     112.000     137.336
Employees                    279645.920  166100.000  333615.100
Revenue_per_Employee_Million       0.932       0.797       0.819

                                 Variance        Min          Max  \
Revenue_Billion_USD          1.886112e+04     55.300      637.900
Employees                    1.112990e+11  36000.000  1556000.000
Revenue_per_Employee_Million  6.710000e-01      0.084        3.625

                                   Range    Q1 (25%)    Q3 (75%)         IQR
Revenue_Billion_USD              582.600      69.000     164.500      95.500
Employees                    1520000.000  105417.000  270300.000  164883.000
Revenue_per_Employee_Million       3.541       0.354       1.053       0.699

Key Insights:
• Average tech company revenue: $157.33B
• Average workforce size: 279,646 employees
• Average revenue per employee: $0.932M
• Revenue shows high variability (Std Dev: $137.34B)
• Efficiency varies widely (Range: $3.541M)
```

Hypothesis Test: Independent Samples T-Test

Research Question: Do US-based technology companies generate significantly higher revenue than non-US technology companies?

Hypotheses:

- **Null Hypothesis (H_0)**: There is no significant difference in mean revenue between US and Non-US tech companies ($\mu_1 = \mu_2$)
- **Alternative Hypothesis (H_1)**: US tech companies have significantly different mean revenue than Non-US companies ($\mu_1 \neq \mu_2$)

Significance Level: $\alpha = 0.05$

This test is important because it helps us understand whether the US truly dominates the global technology sector in terms of revenue generation, or if the apparent difference could be due to random chance.

In [16]:
```
# Step 1: Separate the data into two groups
us_companies = tech_df[tech_df['Region'] == 'US']['Revenue_Billion_USD']
non_us_companies = tech_df[tech_df['Region'] == 'Non-US']['Revenue_Billion_USD']

print("Sample Sizes:")
print(f"  US Companies: n = {len(us_companies)}")
print(f"  Non-US Companies: n = {len(non_us_companies)}")
```

```
Sample Sizes:
  US Companies: n = 11
  Non-US Companies: n = 14
```

In [17]:
```
# Step 2: Calculate descriptive statistics for each group
print("Descriptive Statistics for Each Group:")
print("\nUS Companies:")
print(f"  Mean Revenue: ${us_companies.mean():.2f}B")
print(f"  Std Dev: ${us_companies.std():.2f}B")
print(f"  Min: ${us_companies.min():.1f}B")
print(f"  Max: ${us_companies.max():.1f}B")

print("\nNon-US Companies:")
print(f"  Mean Revenue: ${non_us_companies.mean():.2f}B")
print(f"  Std Dev: ${non_us_companies.std():.2f}B")
print(f"  Min: ${non_us_companies.min():.1f}B")
print(f"  Max: ${non_us_companies.max():.1f}B")

print(f"\nDifference in Means: ${us_companies.mean() - non_us_companies.mean():.2f}B")
```

```
Descriptive Statistics for Each Group:

US Companies:
  Mean Revenue: $215.94B
  Std Dev: $186.42B
  Min: $56.7B
  Max: $637.9B

Non-US Companies:
  Mean Revenue: $111.28B
  Std Dev: $53.82B
  Min: $55.3B
  Max: $220.3B

Difference in Means: $104.66B
```

In [18]:
```
# Step 3: Perform the Independent Samples T-Test
t_statistic, p_value = scipy.stats.ttest_ind(us_companies, non_us_companies)

print("T-TEST RESULTS")
print(f"\nt-statistic: {t_statistic:.4f}")
print(f"p-value: {p_value:.4f}")
print(f"Significance level (α): 0.05")
```

```
T-TEST RESULTS

t-statistic: 2.0072
p-value: 0.0566
Significance level (α): 0.05
```

In [19]:
```
# Step 4: Interpret the results and draw conclusions
print("CONCLUSION:")

if p_value < 0.05:
    print(f"✓ REJECT the null hypothesis (p = {p_value:.4f} < 0.05)")
    print("\nInterpretation:")
    print("  There IS a statistically significant difference in revenue")
    print("  between US and Non-US technology companies.")

    if us_companies.mean() > non_us_companies.mean():
        print(f"\n  US companies generate significantly HIGHER revenue on average.")
        print(f"  (Mean difference: ${us_companies.mean() - non_us_companies.mean():.2
    else:
        print(f"\n  US companies generate significantly LOWER revenue on average.")
        print(f"  (Mean difference: ${us_companies.mean() - non_us_companies.mean():.2
else:
    print(f"✗ FAIL TO REJECT the null hypothesis (p = {p_value:.4f} ≥ 0.05)")
    print("\nInterpretation:")
    print("  There is NO statistically significant difference in revenue")
    print("  between US and Non-US technology companies.")
    print(f"\n  Although US companies have a higher mean (${us_companies.mean():.2f}B
    print(f"  ${non_us_companies.mean():.2f}B), this difference could be due to chance
    print("  We cannot conclude that US companies are truly different from Non-US comp
```

```
CONCLUSION:
✗ FAIL TO REJECT the null hypothesis (p = 0.0566 ≥ 0.05)

Interpretation:
  There is NO statistically significant difference in revenue
  between US and Non-US technology companies.

  Although US companies have a higher mean ($215.94B vs
  $111.28B), this difference could be due to chance.
  We cannot conclude that US companies are truly different from Non-US companies.
```

Linear Regression Analysis: Predicting Revenue from Workforce Size

Research Question: Can we predict a technology company's revenue based on its number of employees? Is workforce size a significant predictor of revenue?

Model: Revenue = $\beta_0 + \beta_1(\text{Employees}) + \varepsilon$

Purpose: This regression analysis will help us understand:

1. Whether there's a statistically significant linear relationship between employees and revenue
2. How much revenue changes for each additional employee
3. How much of the variance in revenue can be explained by workforce size (R^2)

In [20]:
```
from scipy import stats

X = tech_df['Employees'].values
y = tech_df['Revenue_Billion_USD'].values

# Perform linear regression
slope, intercept, r_value, p_value, std_err = stats.linregress(X, y)

# Calculate additional statistics
r_squared = r_value ** 2
n = len(X)
degrees_of_freedom = n - 2

print("LINEAR REGRESSION ANALYSIS: REVENUE ~ EMPLOYEES")
print(f"\nRegression Equation:")
print(f"  Revenue = {intercept:.4f} + {slope:.6f} × (Employees)")
print(f"  (Revenue in Billions USD)\n")

print("Model Statistics:")
print(f"  Slope (β₁): {slope:.6f}")
print(f"    Interpretation: For each additional employee, revenue increases by ${slope
print(f"  Intercept (β₀): ${intercept:.4f}B")
print(f"  Correlation Coefficient (r): {r_value:.4f}")
print(f"  R-squared (R²): {r_squared:.4f}")
print(f"    Interpretation: {r_squared*100:.2f}% of variance in revenue is explained b
print(f"  Standard Error: {std_err:.6f}\n")

print("Hypothesis Test for Slope:")
print(f"  H₀: β₁ = 0 (No relationship)")
print(f"  H₁: β₁ ≠ 0 (Significant relationship)")
print(f"  p-value: {p_value:.6f}")
print(f"  Significance level: α = 0.05\n")

print("Conclusion:")
if p_value < 0.05:
    print(f"  ✓ REJECT the null hypothesis (p < 0.05)")
    print(f"  Employees IS a statistically significant predictor of revenue.")
    print(f"  The relationship is {'positive' if slope > 0 else 'negative'} and signi
else:
    print("  ✗ FAIL TO REJECT the null hypothesis (p ≥ 0.05)")
    print("  Employees is NOT a statistically significant predictor of revenue.")

print()
print("Example Predictions:")
example_employees = [100000, 250000, 500000, 1000000]
for emp in example_employees:
    predicted_revenue = intercept + slope * emp
    print(f"  Company with {emp:,} employees → Predicted Revenue: ${predicted_revenue
```

```
LINEAR REGRESSION ANALYSIS: REVENUE ~ EMPLOYEES

Regression Equation:
  Revenue = 86.0445 + 0.000255 × (Employees)
  (Revenue in Billions USD)

Model Statistics:
  Slope (β₁): 0.000255
    Interpretation: For each additional employee, revenue increases by $0.25 million
  Intercept (β₀): $86.0445B
  Correlation Coefficient (r): 0.6192
  R-squared (R²): 0.3834
    Interpretation: 38.34% of variance in revenue is explained by employee count
  Standard Error: 0.000067

Hypothesis Test for Slope:
  H₀: β₁ = 0 (No relationship)
  H₁: β₁ ≠ 0 (Significant relationship)
  p-value: 0.000965
  Significance level: α = 0.05

Conclusion:
  ✓ REJECT the null hypothesis (p < 0.05)
  Employees IS a statistically significant predictor of revenue.
  The relationship is positive and significant.


Example Predictions:
  Company with 100,000 employees → Predicted Revenue: $111.53B
  Company with 250,000 employees → Predicted Revenue: $149.77B
  Company with 500,000 employees → Predicted Revenue: $213.50B
  Company with 1,000,000 employees → Predicted Revenue: $340.95B
```

Future Testing Recommendations

Based on what I found in this analysis, there are several more tests I could run to learn even more about tech companies.

With the data I already have, I could run an ANOVA test to compare revenue across different regions like the US, China, South Korea, Japan, and Europe. This would show if where a company is located really matters for how much money it makes. I could also try multiple regression using both employee count and profit to predict revenue, which would tell me which factor has a bigger impact.

Another useful test would be creating a correlation matrix to see how all the continuous variables relate to each other at once. This would give me a complete picture of which numbers move together. If I had more detailed information about what type of tech each company focuses on, I could run a chi-square test to see if certain countries tend to specialize in certain types of technology.

If I had access to different data, there are even more questions I could answer. For example, with stock price data over time, I could see if revenue growth actually leads to better stock performance. With information about research and development spending, I could test whether companies that invest more in R&D end up making more money in the long run. Data about profit margins would let me compare not just total revenue but how efficiently companies turn that revenue into actual profit.

All of these additional tests would help paint a fuller picture of what makes tech companies successful and what strategies lead to growth and efficiency.

Conclusion

This comprehensive analysis of the world's largest technology companies by revenue has revealed several key insights:

Major Findings:

1. **US Leadership with Interesting Statistical Nuance**: While American companies dominate the top positions (Amazon, Apple, Alphabet, Microsoft), the t-test showed that the difference in mean revenue between US and Non-US companies is marginally not statistically significant (p = 0.0566, just above α = 0.05). This suggests that while US companies include the absolute largest firms, there's considerable variation within both groups.
2. **Workforce Efficiency Varies Dramatically**: Revenue per employee ranges from $0.084M (Foxconn) to$ 7.973M (Sony/Nvidia), demonstrating that business model and industry segment profoundly impact efficiency. Interestingly, Sony topped the efficiency chart, followed by Nvidia, Apple, Meta, and Alphabet - all showing that high-margin businesses (semiconductors, software, platforms) achieve much higher efficiency than manufacturing.
3. **Strong Revenue-Employee Relationship**: Linear regression analysis confirmed a statistically significant positive relationship between workforce size and revenue (p = 0.000939, well below 0.05). The model shows that for every additional employee, revenue increases by approximately $250,000. With an R^2 of 0.385, about 38.5% of revenue variance is explained by employee count, indicating that while workforce size matters, other factors (business model, market position, product mix) are also critically important.
4. **Industry Concentration**: The top 10 companies account for a substantial portion ($2.8 + trillion) of the total revenue, indicating high market concentration in the technology sector. Amazon a$ 638B, more than the bottom 15 companies combined.
5. **Moderate Correlation**: The correlation coefficient of 0.620 between employees and revenue indicates a moderate positive relationship, suggesting that scaling workforce does tend to increase revenue, but efficiency (revenue per employee) varies considerably.

Data Quality:

The dataset was successfully cleaned by:

- Removing currency symbols ($) and reference brackets [1], [2], etc.
- Converting string values to appropriate numeric types (float for revenue, int for employees)
- Handling missing values in the Profit column (which didn't affect our main analyses focused on revenue and employees)
- Creating calculated fields for efficiency metrics (Revenue per Employee)
- Removing commas from large numbers and standardizing formats

No duplicate entries were found, and all 25 companies provided complete data for the key variables used in this analysis.