In [1]:
```
import pandas as pd
import os
import psycopg2
from sqlalchemy import create_engine
```

In [3]:
```
file_paths = [r"C:\Users\hp\OneDrive\Desktop\IT_Services_Marketshare_Masked\IT_Services_Marketshare_2020 (742642).CSV",
              r"C:\Users\hp\OneDrive\Desktop\IT_Services_Marketshare_Masked\IT_Services_Marketshare_2020Q1.CSV",
              r"C:\Users\hp\OneDrive\Desktop\IT_Services_Marketshare_Masked\IT_Services_Marketshare_2021 (765402).CSV",
              r"C:\Users\hp\OneDrive\Desktop\IT_Services_Marketshare_Masked\IT_Services_Marketshare_2022 (787876).CSV",
              r"C:\Users\hp\OneDrive\Desktop\IT_Services_Marketshare_Masked\Services_Market_Share_2023 (808454).CSV"
             ]
```

In [4]:
```
#load and merging csv files
dataframes = [pd.read_csv(file) for file in file_paths]
df = pd.concat(dataframes, ignore_index=True)
```

In [5]:
```
#inspecting data
print(df.info())  # Checking column names, data types, and missing values
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2080794 entries, 0 to 2080793
Data columns (total 15 columns):
 #   Column                          Dtype  
---  ------                          -----  
 0   Year                            object 
 1   Super Region                    object 
 2   Region                          object 
 3   Country                         object 
 4   Vendor                          object 
 5   Service 1                       object 
 6   Service 2                       object 
 7   Service 3                       object 
 8   Vertical                        object 
 9   Ticker                          object 
 10  HQ Country                      object 
 11  VendorRevenue - USD             float64
 12  ConstantCurrency Revenue - USD  float64
 13  Vendor Name                     object 
 14  Vendor.1                        object 
dtypes: float64(2), object(13)
memory usage: 238.1+ MB
None
```

In [6]:
```
print(df.head())
```

```
      Year    Super Region          Region         Country    Vendor  \
0  2019 YR  Eastern Europe  Eastern Europe  Czech Republic  Vendor 2
1  2019 YR  Eastern Europe  Eastern Europe  Czech Republic  Vendor 2
2  2019 YR  Eastern Europe  Eastern Europe  Czech Republic  Vendor 2
3  2019 YR  Eastern Europe  Eastern Europe  Czech Republic  Vendor 2
4  2019 YR  Eastern Europe  Eastern Europe  Czech Republic  Vendor 2

     Service 1                                    Service 2  \
0  IT Services  Application Implementation & Managed Services
1  IT Services  Application Implementation & Managed Services
2  IT Services  Application Implementation & Managed Services
3  IT Services  Application Implementation & Managed Services
4  IT Services  Application Implementation & Managed Services

                    Service 3                         Vertical    Ticker  \
0  Application Implementation             Banking & Securities  Ticker 2
1  Application Implementation  Communications, Media & Services  Ticker 2
2  Application Implementation                        Education  Ticker 2
3  Application Implementation                       Government  Ticker 2
4  Application Implementation             Healthcare Providers  Ticker 2

      HQ Country  VendorRevenue - USD  ConstantCurrency Revenue - USD  \
0  United States             5.693025                        5.632195
1  United States             8.248331                        8.160199
2  United States             0.098793                        0.097737
3  United States             4.374420                        4.327679
4  United States             2.630113                        2.602010

  Vendor Name Vendor.1
0         NaN      NaN
1         NaN      NaN
2         NaN      NaN
3         NaN      NaN
4         NaN      NaN
```

In [7]:
```
# Count missing values
print(df.isnull().sum())
```

```
Year                                   0
Super Region                      444553
Region                                 0
Country                                0
Vendor                            331724
Service 1                              0
Service 2                              0
Service 3                              0
Vertical                               0
Ticker                                 0
HQ Country                         31068
VendorRevenue - USD                    0
ConstantCurrency Revenue - USD         0
Vendor Name                      1749070
Vendor.1                         1186930
dtype: int64
```

In [8]:
```
#lets hadle the missing values
df.fillna("Unknown", inplace=True)  # Replace missing values with "Unknown"es
```

In [13]:
```
df['Year'] = df['Year'].str.extract(r'(\d{4})')  # Extract four-digit year
df['Year'] = df['Year'].astype(int)  # Convert to integer type
```

In [14]:
```
print(df['year'].unique())  # Ensure only valid years remain
```

```
['2019' '2020' '2018' '2021' '2022' '2023']
```

In [15]:
```
#handling missing values
df.drop(columns=["Vendor Name", "Vendor.1"], inplace=True) #dropping becuase to many missing values
```

In [17]:
```
df["Super Region"].fillna("Unknown", inplace=True)
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_9912\982764173.py:1: FutureWarning: A value is trying to be set on a copy of a DataFra
me or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are set
ting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = d
f[col].method(value) instead, to perform the operation inplace on the original object.


  df["Super Region"].fillna("Unknown", inplace=True)
```

In [18]:
```
df = df.assign(
    **{
        "Super Region": df["Super Region"].fillna("Unknown"),
        "Vendor": df["Vendor"].fillna("Not Available"),
        "HQ Country": df["HQ Country"].fillna("Unknown"),
    }
)
```

In [19]:
```
#checking data quality
print(f"Duplicate Rows: {df.duplicated().sum()}")
```

```
Duplicate Rows: 104050
```

In [20]:
```
#dropping duplicate files
df = df.drop_duplicates()
```

In [21]:
```
#Check for Inconsistent Data Entries
print(df["Country"].unique())  # Check for inconsistencies in country names
print(df["Vendor"].unique()[:20])  # Sample vendor names
```

```
['Czech Republic' 'Hungary' 'Poland' 'Rest of Eastern Europe' 'India'
 'Indonesia' 'Malaysia' 'Rest of Emerging Asia/Pacific' 'Thailand'
 'Rest of Eurasia' 'Russia' 'China' 'Hong Kong' 'Taiwan' 'Japan'
 'Argentina' 'Brazil' 'Chile' 'Colombia' 'Mexico' 'Rest of Latin America'
 'Australia' 'New Zealand' 'Singapore' 'South Korea' 'Israel'
 'Rest of Middle East and North Africa' 'Saudi Arabia' 'Turkey' 'Canada'
 'United States' 'Rest of Sub-Saharan Africa' 'South Africa' 'Austria'
 'Belgium' 'Denmark' 'Finland' 'France' 'Germany' 'Greece' 'Ireland'
 'Italy' 'Netherlands' 'Norway' 'Portugal' 'Rest of Western Europe'
 'Spain' 'Sweden' 'Switzerland' 'United Kingdom' 'Rest of Europe']
['Vendor 2' 'Vendor 3' 'Vendor 8' 'Vendor 9' 'Vendor 10' 'Vendor 11'
 'Vendor 14' 'Vendor 15' 'Vendor 17' 'Vendor 18' 'Vendor 187' 'Vendor 24'
 'Vendor 157' 'Vendor 159' 'Vendor 271' 'Vendor 29' 'Vendor 188'
 'Vendor 290' 'Vendor 32' 'Vendor 189']
```

In [22]:
```
print(df.describe())  # Gives statistical summary of numerical columns
```

```
               Year  VendorRevenue - USD  ConstantCurrency Revenue - USD
count  1.976744e+06         1.976744e+06                    1.976744e+06
mean   2.020621e+03         5.895189e+00                    6.079768e+00
std    1.498219e+00         6.508654e+01                    6.563189e+01
min    2.018000e+03        -1.298388e+00                   -1.407164e+00
25%    2.019000e+03         1.773009e-02                    1.885000e-02
50%    2.021000e+03         1.566611e-01                    1.654410e-01
75%    2.022000e+03         1.132680e+00                    1.189012e+00
max    2.023000e+03         1.364897e+04                    1.364897e+04
```

In [23]:
```
# Removing Negative Revenue Values
print(df[df["VendorRevenue - USD"] < 0])  # See the negative values
print(df[df["ConstantCurrency Revenue - USD"] < 0])
```

```
         Year           Super Region                 Region  \
1848     2019         Eastern Europe         Eastern Europe
4867     2019         Eastern Europe         Eastern Europe
8449     2019         Eastern Europe         Eastern Europe
12696    2019         Eastern Europe         Eastern Europe
17219    2019  Emerging Asia/Pacific  Emerging Asia/Pacific
...       ...                    ...                    ...
2037101  2023                Unknown    Mature Asia/Pacific
2037102  2023                Unknown    Mature Asia/Pacific
2037103  2023                Unknown    Mature Asia/Pacific
2037104  2023                Unknown    Mature Asia/Pacific
2037105  2023                Unknown    Mature Asia/Pacific

                      Country     Vendor    Service 1  \
1848           Czech Republic  Vendor 76  IT Services
4867                  Hungary  Vendor 76  IT Services
8449                   Poland  Vendor 76  IT Services
12696    Rest of Eastern Europe  Vendor 76  IT Services
17219                   India  Vendor 76  IT Services
...                       ...        ...          ...
2037101             Singapore  Vendor 30     Services
2037102             Singapore  Vendor 30     Services
2037103             Singapore  Vendor 30     Services
2037104             Singapore  Vendor 30     Services
2037105             Singapore  Vendor 30     Services

                                          Service 2  \
1848     Application Implementation & Managed Services
4867     Application Implementation & Managed Services
8449     Application Implementation & Managed Services
12696    Application Implementation & Managed Services
17219    Application Implementation & Managed Services
...                                                ...
2037101                  Business Process Services
2037102                  Business Process Services
2037103                  Business Process Services
2037104                  Business Process Services
2037105                  Business Process Services

                          Service 3                          Vertical  \
1848     Application Implementation                   Wholesale Trade
```

```
4867     Application Implementation                      Wholesale Trade
8449     Application Implementation                      Wholesale Trade
12696    Application Implementation                      Wholesale Trade
17219    Application Implementation                      Wholesale Trade
...                             ...                                  ...
2037101   Business Process Services    Communications Media and Services
2037102   Business Process Services                           Healthcare
2037103   Business Process Services                            Insurance
2037104   Business Process Services  Manufacturing and Natural Resources
2037105   Business Process Services                               Retail

            Ticker     HQ Country  VendorRevenue - USD  \
1848     Ticker 76  United States        -5.777993e-02
4867     Ticker 76  United States        -1.556389e-02
8449     Ticker 76  United States        -5.918199e-02
12696    Ticker 76  United States        -9.580049e-02
17219    Ticker 76  United States        -4.164214e-01
...            ...            ...                  ...
2037101  Ticker 30          Japan        -1.660000e-08
2037102  Ticker 30          Japan        -5.770000e-09
2037103  Ticker 30          Japan        -3.500000e-09
2037104  Ticker 30          Japan        -3.560000e-08
2037105  Ticker 30          Japan        -4.630000e-08

         ConstantCurrency Revenue - USD  year
1848                          -0.057162  2019
4867                          -0.014701  2019
8449                          -0.058417  2019
12696                         -0.097721  2019
17219                         -0.395860  2019
...                                 ...   ...
2037101                        0.000000  2023
2037102                        0.000000  2023
2037103                        0.000000  2023
2037104                        0.000000  2023
2037105                        0.000000  2023

[142 rows x 14 columns]
         Year           Super Region                 Region  \
1848     2019         Eastern Europe         Eastern Europe
4867     2019         Eastern Europe         Eastern Europe
```

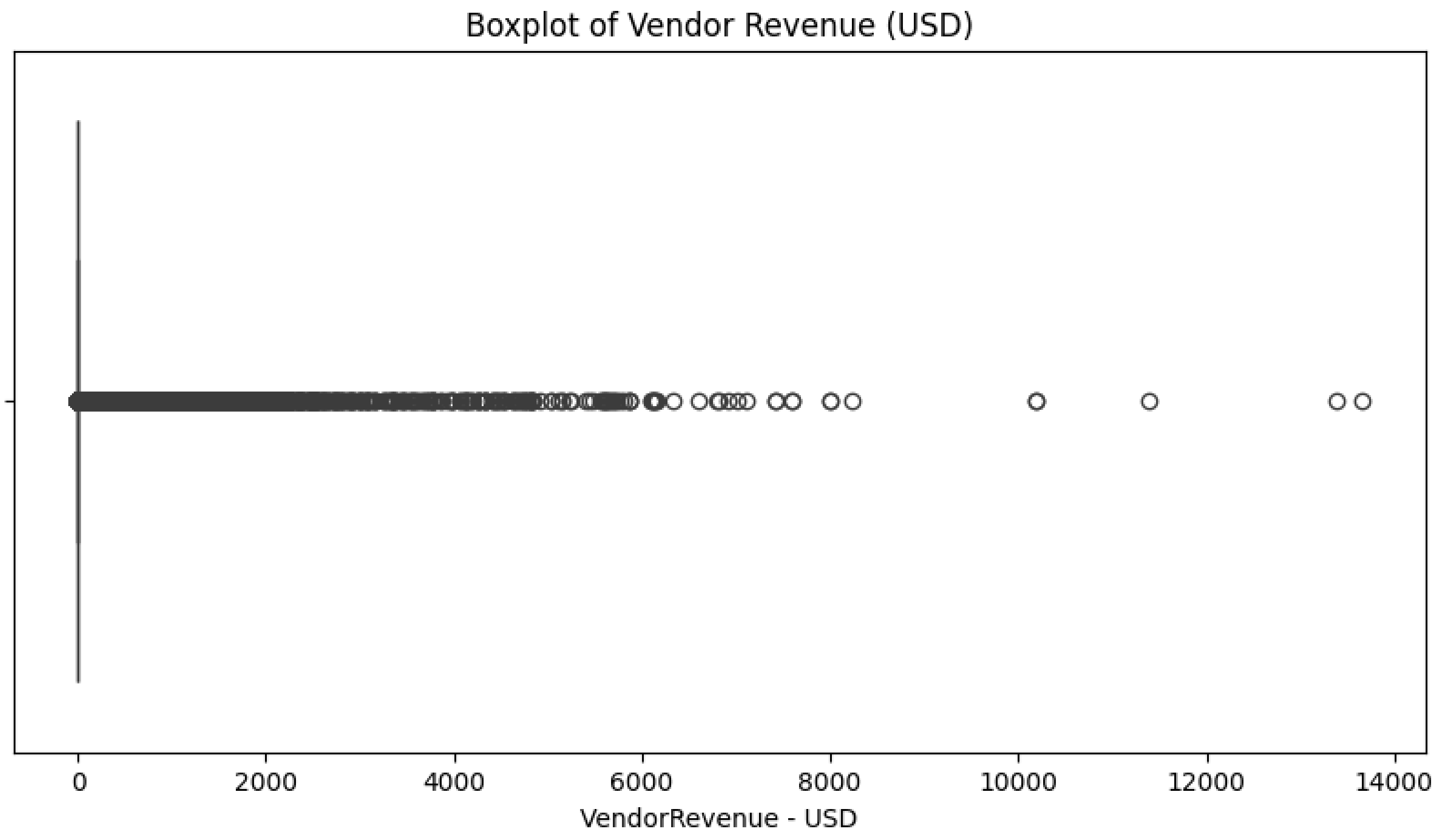
```
8449     2019         Eastern Europe         Eastern Europe
12696    2019         Eastern Europe         Eastern Europe
17219    2019  Emerging Asia/Pacific  Emerging Asia/Pacific
...       ...                    ...                    ...
1852679  2022                Unknown          North America
1952136  2023                Unknown                 Europe
1981551  2023                Unknown                 Europe
1981554  2023                Unknown                 Europe
1981584  2023                Unknown                 Europe

                        Country      Vendor    Service 1  \
1848             Czech Republic   Vendor 76  IT Services   
4867                    Hungary   Vendor 76  IT Services   
8449                     Poland   Vendor 76  IT Services   
12696    Rest of Eastern Europe   Vendor 76  IT Services   
17219                     India   Vendor 76  IT Services   
...                         ...         ...          ...   
1852679           United States  Vendor 149     Services   
1952136             Netherlands  Vendor 129     Services   
1981551                  Sweden  Vendor 129     Services   
1981554                  Sweden  Vendor 129     Services   
1981584                  Sweden  Vendor 129     Services   

                                                 Service 2  \
1848         Application Implementation & Managed Services   
4867         Application Implementation & Managed Services   
8449         Application Implementation & Managed Services   
12696        Application Implementation & Managed Services   
17219        Application Implementation & Managed Services   
...                                                    ...   
1852679                                         Consulting   
1952136    Application Implementation and Managed Services   
1981551    Application Implementation and Managed Services   
1981554    Application Implementation and Managed Services   
1981584  Infrastructure Implementation and Managed Serv...   

                              Service 3           Vertical      Ticker  \
1848         Application Implementation    Wholesale Trade   Ticker 76   
4867         Application Implementation    Wholesale Trade   Ticker 76   
8449         Application Implementation    Wholesale Trade   Ticker 76   
12696        Application Implementation    Wholesale Trade   Ticker 76   
```

```
17219             Application Implementation      Wholesale Trade   Ticker 76
...                                      ...                  ...         ...
1852679                Technology Consulting               Retail  Ticker 149
1952136  Application Managed Services (AMS)      Wholesale Trade  Ticker 129
1981551  Application Managed Services (AMS)  Power and Utilities  Ticker 129
1981554  Application Managed Services (AMS)      Wholesale Trade  Ticker 129
1981584       Infrastructure Implementation  Power and Utilities  Ticker 129

             HQ Country  VendorRevenue - USD  ConstantCurrency Revenue - USD  \
1848      United States            -0.057780                       -0.057162
4867      United States            -0.015564                       -0.014701
8449      United States            -0.059182                       -0.058417
12696     United States            -0.095800                       -0.097721
17219     United States            -0.416421                       -0.395860
...                 ...                  ...                             ...
1852679          France            -1.144791                       -1.144790
1952136  United Kingdom            -0.234358                       -0.246991
1981551  United Kingdom            -0.001468                       -0.001688
1981554  United Kingdom            -0.051403                       -0.059133
1981584  United Kingdom            -0.002433                       -0.002798

         year
1848     2019
4867     2019
8449     2019
12696    2019
17219    2019
...       ...
1852679  2022
1952136  2023
1981551  2023
1981554  2023
1981584  2023

[84 rows x 14 columns]
```

In [24]:
```
df = df[df["VendorRevenue - USD"] >= 0]
df = df[df["ConstantCurrency Revenue - USD"] >= 0]
```

In [25]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [26]:
```
#handle outliers
plt.figure(figsize=(10, 5))
sns.boxplot(x=df["VendorRevenue - USD"])
plt.title("Boxplot of Vendor Revenue (USD)")
plt.show()
```



In [28]:
```
upper_limit = df["VendorRevenue - USD"].quantile(0.99)
df["VendorRevenue - USD"] = df["VendorRevenue - USD"].clip(upper=upper_limit)
```

```
In [29]: #rechecking data
         df.info()
         df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1976602 entries, 0 to 2080793
Data columns (total 14 columns):
 #   Column                          Dtype  
---  ------                          -----  
 0   Year                            int32  
 1   Super Region                    object 
 2   Region                          object 
 3   Country                         object 
 4   Vendor                          object 
 5   Service 1                       object 
 6   Service 2                       object 
 7   Service 3                       object 
 8   Vertical                        object 
 9   Ticker                          object 
 10  HQ Country                      object 
 11  VendorRevenue - USD             float64
 12  ConstantCurrency Revenue - USD  float64
 13  year                            object 
dtypes: float64(2), int32(1), object(11)
memory usage: 218.7+ MB
```

Out[29]:

	Year	Super Region	Region	Country	Vendor	Service 1	Service 2	Service 3	Vertical	Ticker	HQ Country	VendorRevenue - USD	C
0	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Banking & Securities	Ticker 2	United States	5.693025	
1	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Communications, Media & Services	Ticker 2	United States	8.248331	
2	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Education	Ticker 2	United States	0.098793	
3	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Government	Ticker 2	United States	4.374420	
4	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Healthcare Providers	Ticker 2	United States	2.630113	

In [30]:
```
# Drop the duplicate 'year' column
df.drop(columns=['year'], inplace=True, errors='ignore')
```

In [31]:
```
# Checking for missing values again
print("Missing Values:\n", df.isnull().sum())
```

```
Missing Values:
 Year                                 0
Super Region                         0
Region                               0
Country                              0
Vendor                               0
Service 1                            0
Service 2                            0
Service 3                            0
Vertical                             0
Ticker                               0
HQ Country                           0
VendorRevenue - USD                  0
ConstantCurrency Revenue - USD       0
dtype: int64
```

In [32]:
```
# Checking for duplicate rows
print("Duplicate Rows:", df.duplicated().sum())
```

```
Duplicate Rows: 509
```

In [33]:
```
# Removing duplicate rows
df.drop_duplicates(inplace=True)
```

In [34]:
```
# Checking again for duplicate rows
print("Duplicate Rows:", df.duplicated().sum())
```

```
Duplicate Rows: 0
```

In [35]:
```
# Confirming changes
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1976093 entries, 0 to 2080793
Data columns (total 13 columns):
 #   Column                          Dtype  
---  ------                          -----  
 0   Year                            int32  
 1   Super Region                    object 
 2   Region                          object 
 3   Country                         object 
 4   Vendor                          object 
 5   Service 1                       object 
 6   Service 2                       object 
 7   Service 3                       object 
 8   Vertical                        object 
 9   Ticker                          object 
 10  HQ Country                      object 
 11  VendorRevenue - USD             float64
 12  ConstantCurrency Revenue - USD  float64
dtypes: float64(2), int32(1), object(10)
memory usage: 203.5+ MB
```

Out[35]:

	Year	Super Region	Region	Country	Vendor	Service 1	Service 2	Service 3	Vertical	Ticker	HQ Country	VendorRevenue - USD
0	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Banking & Securities	Ticker 2	United States	5.693025
1	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Communications, Media & Services	Ticker 2	United States	8.248331
2	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Education	Ticker 2	United States	0.098793
3	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Government	Ticker 2	United States	4.374420
4	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Healthcare Providers	Ticker 2	United States	2.630113

In [71]:

```python
from sqlalchemy import create_engine

DB_URL = "postgresql://postgres:Raftaar2810@localhost:5432/market_share"

try:
    engine = create_engine(DB_URL)
    conn = engine.connect()
    print("Connection successful!")
except Exception as e:
    print(f" Connection failed: {e}")
```

```
Connection successful!
```

In [51]:
```
#saving the clean dataset
df.to_csv("cleaned_data.csv", index=False)
```

In [52]:
```
df = pd.read_csv("cleaned_data.csv")
df.head()
```

Out[52]:

	Year	Super Region	Region	Country	Vendor	Service 1	Service 2	Service 3	Vertical	Ticker	HQ Country	VendorRevenue - USD
0	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Banking & Securities	Ticker 2	United States	5.693025
1	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Communications, Media & Services	Ticker 2	United States	8.248331
2	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Education	Ticker 2	United States	0.098793
3	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Government	Ticker 2	United States	4.374420
4	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Healthcare Providers	Ticker 2	United States	2.630113

In [69]:
```
from sqlalchemy import create_engine
DB_URL = "postgresql://postgres:Raftaar2810@localhost:5432/market_share"
engine = create_engine(DB_URL)
```

```
df = pd.read_csv("cleaned_data.csv")
df.to_sql("market_share_table", engine, if_exists="replace", index=False)
print(" Data successfully loaded into PostgreSQL!")
```

```
Data successfully loaded into PostgreSQL!
```

In [54]:

```
#loading data from postgresql in to pandas
import pandas as pd
from sqlalchemy import create_engine
DB_URL = "postgresql://postgres:Raftaar2810@localhost:5432/market_share"  # Database connection
engine = create_engine(DB_URL)
# Load data into a Pandas DataFrame
query = "SELECT * FROM market_share_table;"
df = pd.read_sql(query, engine)
df.head()
```

Out[54]:

	Year	Super Region	Region	Country	Vendor	Service 1	Service 2	Service 3	Vertical	Ticker	HQ Country	VendorRevenue - USD	C
0	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Banking & Securities	Ticker 2	United States	5.693025	
1	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Communications, Media & Services	Ticker 2	United States	8.248331	
2	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Education	Ticker 2	United States	0.098793	
3	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Government	Ticker 2	United States	4.374420	
4	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Healthcare Providers	Ticker 2	United States	2.630113	

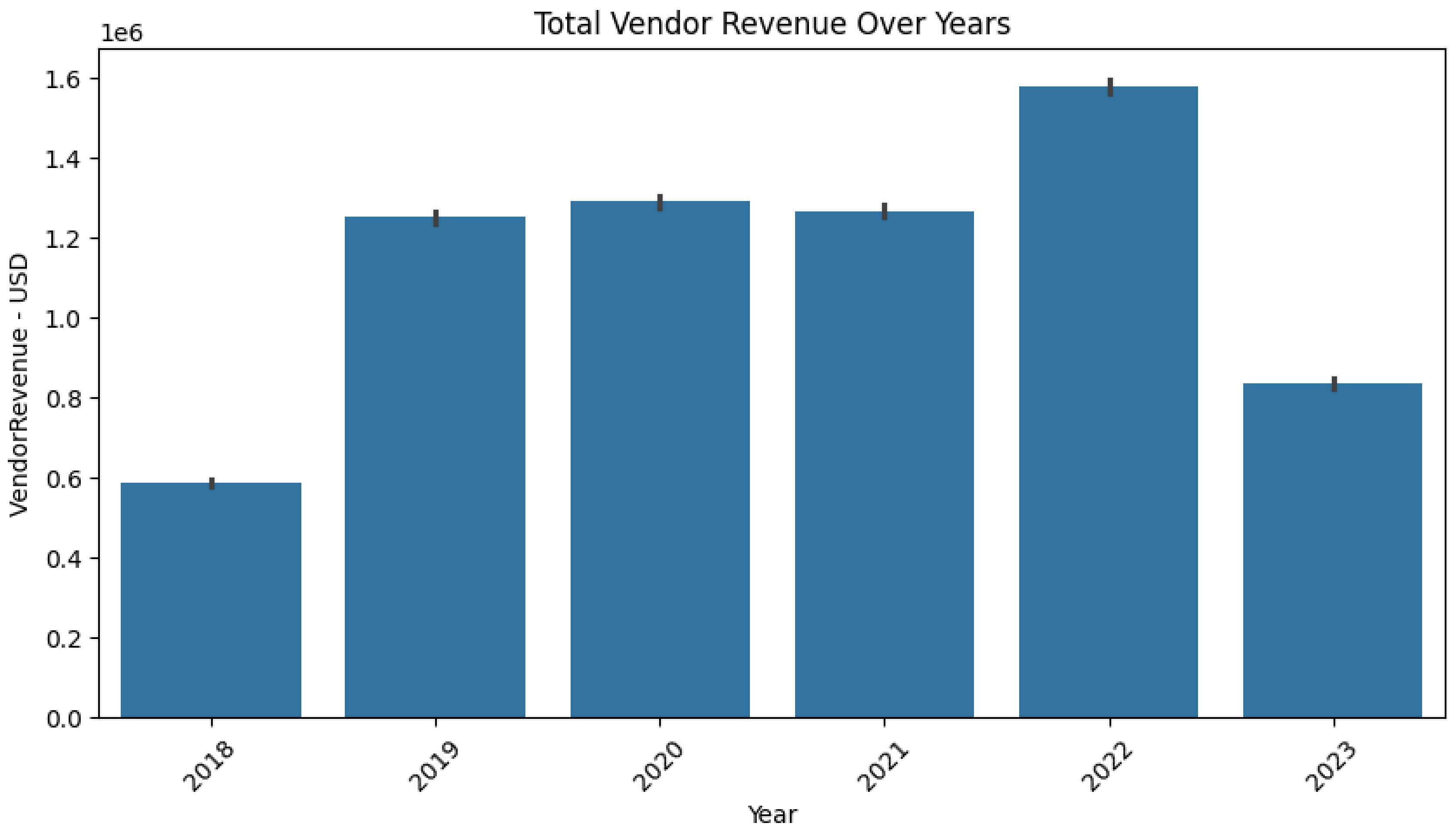
In [55]:
```
df.describe()
```

Out[55]:

	Year	VendorRevenue - USD	ConstantCurrency Revenue - USD
count	1.976093e+06	1.976093e+06	1.976093e+06
mean	2.020621e+03	3.446757e+00	5.961678e+00
std	1.498403e+00	1.234379e+01	6.392268e+01
min	2.018000e+03	0.000000e+00	0.000000e+00
25%	2.019000e+03	1.772821e-02	1.884700e-02
50%	2.021000e+03	1.565538e-01	1.653190e-01
75%	2.022000e+03	1.130908e+00	1.187183e+00
max	2.023000e+03	9.503382e+01	1.364897e+04

In [56]:

```
# sample visaulization:- Revenue Trend Over Years
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10, 5))
sns.barplot(x=df["Year"], y=df["VendorRevenue - USD"], estimator=sum)
plt.xticks(rotation=45)
plt.title("Total Vendor Revenue Over Years")
plt.show()
```



In [57]:
```
#Loading the Cleaned Data
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("cleaned_data.csv") # Load the cleaned dataset
df.head() # Verify data is loaded correctly
```

Out[57]:

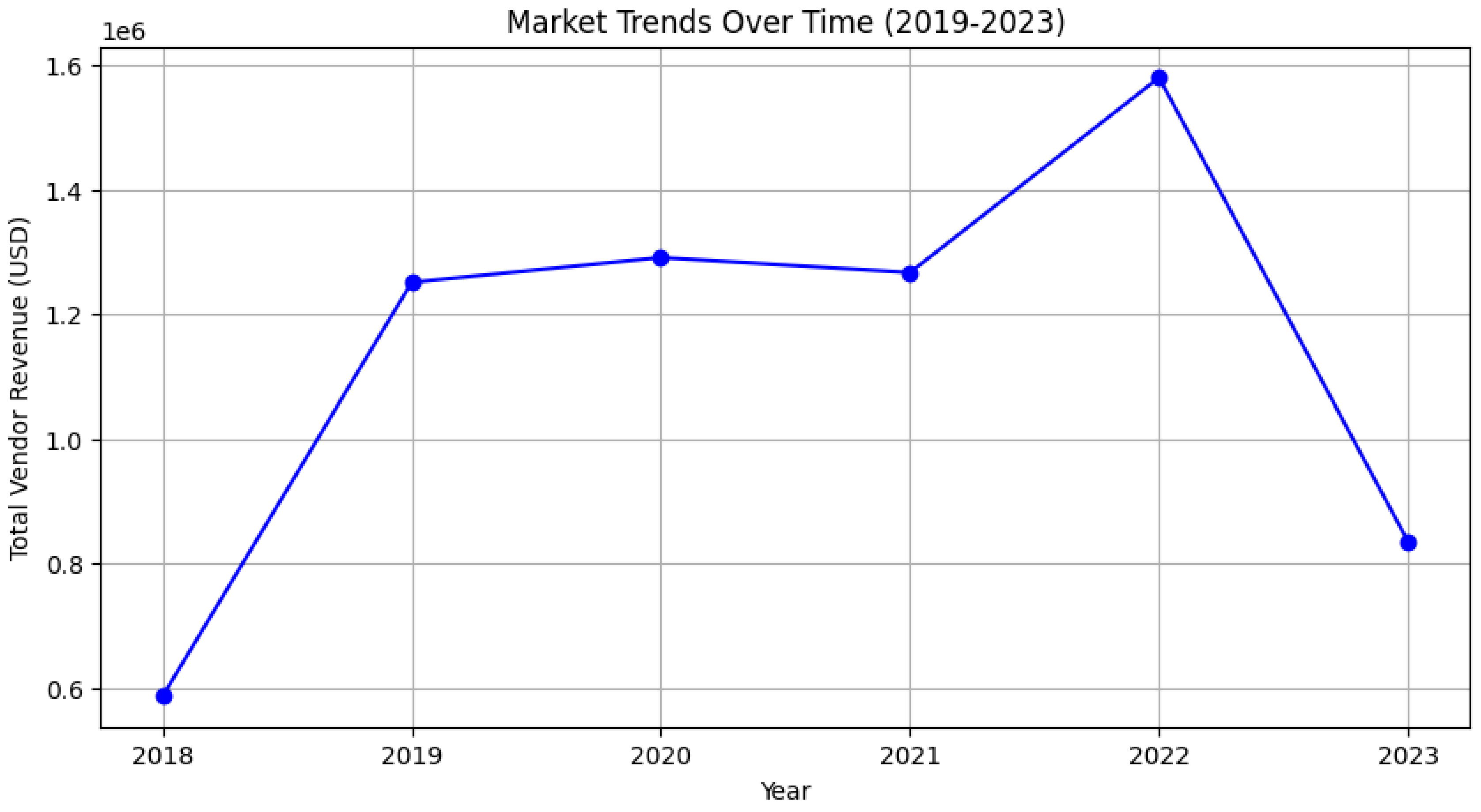
	Year	Super Region	Region	Country	Vendor	Service 1	Service 2	Service 3	Vertical	Ticker	HQ Country	VendorRevenue - USD	C
0	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Banking & Securities	Ticker 2	United States	5.693025	
1	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Communications, Media & Services	Ticker 2	United States	8.248331	
2	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Education	Ticker 2	United States	0.098793	
3	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Government	Ticker 2	United States	4.374420	
4	2019	Eastern Europe	Eastern Europe	Czech Republic	Vendor 2	IT Services	Application Implementation & Managed Services	Application Implementation	Healthcare Providers	Ticker 2	United States	2.630113	

In [58]:

```python
# Aggregate total revenue by year
revenue_by_year = df.groupby("Year")["VendorRevenue - USD"].sum().reset_index()
print(revenue_by_year) # Display the result
```

```
   Year  VendorRevenue - USD
0  2018         5.877803e+05
1  2019         1.250904e+06
2  2020         1.290538e+06
3  2021         1.266611e+06
4  2022         1.579407e+06
5  2023         8.358715e+05
```

In [59]:
```
# Plot the trend of total vendor revenue over years
plt.figure(figsize=(10, 5))
plt.plot(revenue_by_year["Year"], revenue_by_year["VendorRevenue - USD"], marker="o", linestyle="-", color="b")
plt.xlabel("Year")
plt.ylabel("Total Vendor Revenue (USD)")
plt.title("Market Trends Over Time (2019-2023)")
plt.grid(True)
plt.show()
```

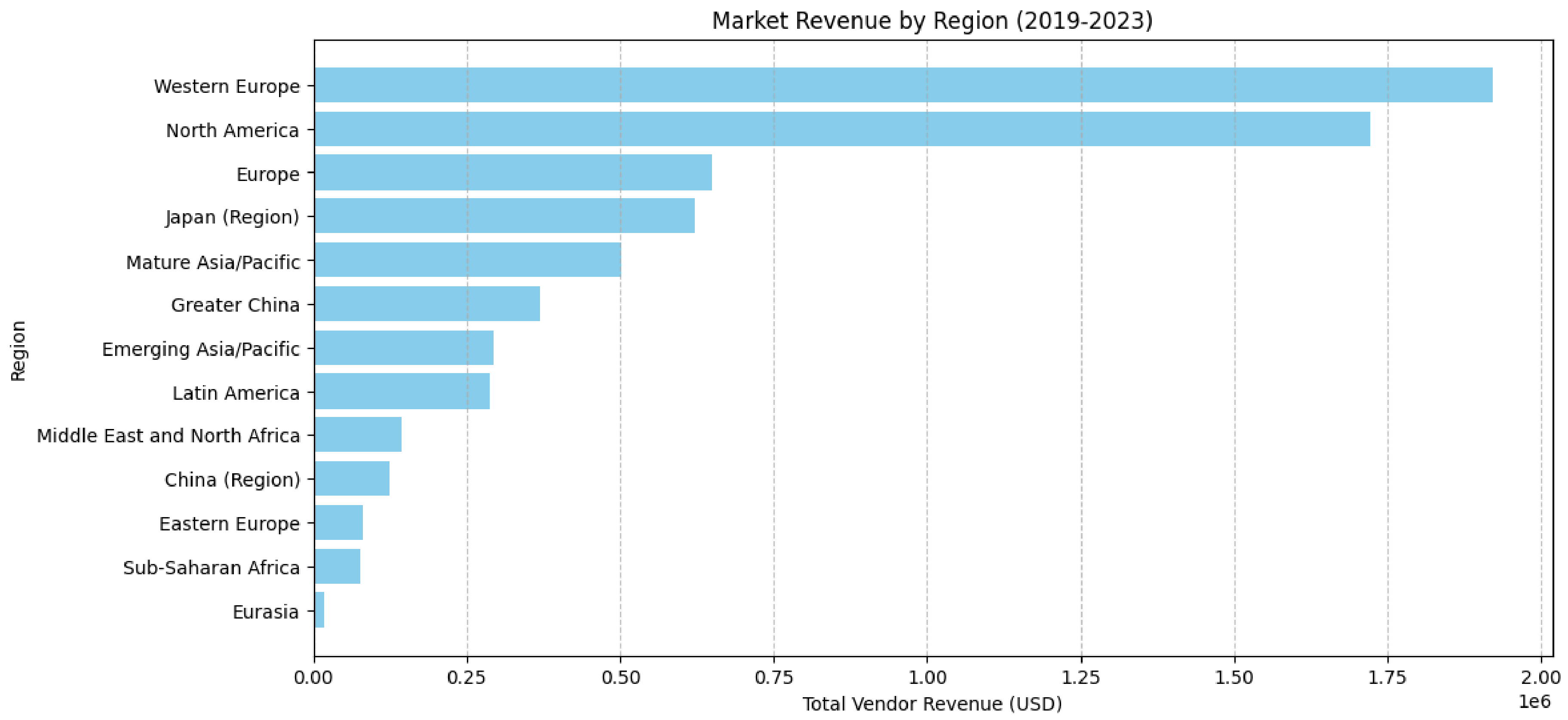


In [ ]:
```
# Revenue By region
```

In [60]:
```
# Aggregate total revenue by region
revenue_by_region = df.groupby("Region")["VendorRevenue - USD"].sum().reset_index()
# Sort by revenue for better visualization
revenue_by_region = revenue_by_region.sort_values(by="VendorRevenue - USD", ascending=False)
print(revenue_by_region.head(10))  # Show top 10 regions
```

```
                          Region  VendorRevenue - USD
12                Western Europe         1.922836e+06
10                 North America         1.722870e+06
4                         Europe         6.487239e+05
6                 Japan (Region)         6.218530e+05
8           Mature Asia/Pacific          5.022991e+05
5                  Greater China         3.690751e+05
2         Emerging Asia/Pacific          2.928379e+05
7                  Latin America         2.877018e+05
9   Middle East and North Africa         1.443784e+05
0                 China (Region)         1.234024e+05
```

In [61]:
```
# Plot revenue by region
plt.figure(figsize=(12, 6))
plt.barh(revenue_by_region["Region"], revenue_by_region["VendorRevenue - USD"], color="skyblue")
plt.xlabel("Total Vendor Revenue (USD)")
plt.ylabel("Region")
plt.title("Market Revenue by Region (2019-2023)")
plt.gca().invert_yaxis()  # Invert y-axis to show highest first
plt.grid(axis="x", linestyle="--", alpha=0.7)
plt.show()
```



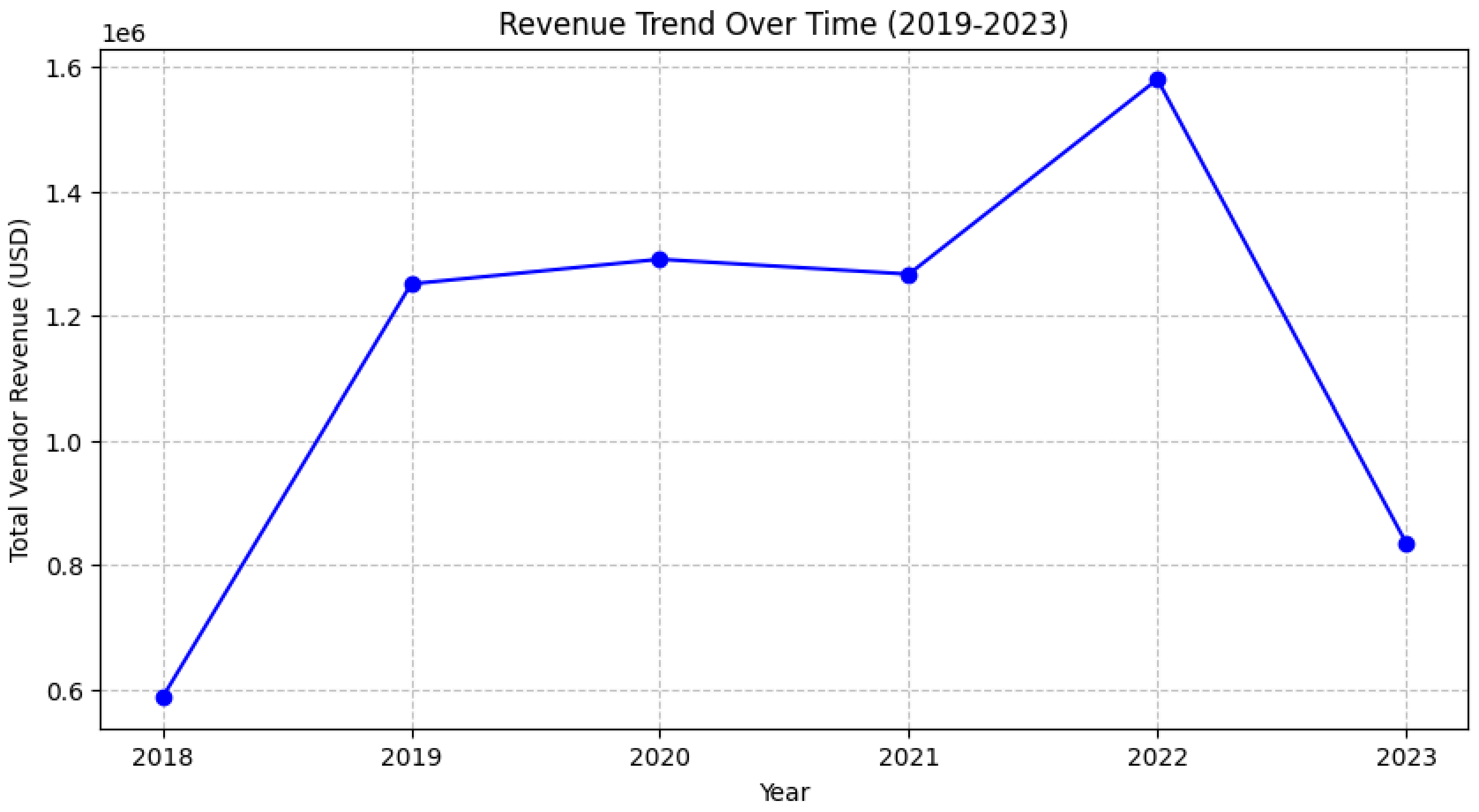
In [ ]:
```
# Revenue Trend Over Time
```

In [62]:
```
# Aggregate total revenue by year
revenue_by_year = df.groupby("Year")["VendorRevenue - USD"].sum().reset_index()
# Sort by year for proper trend visualization
revenue_by_year = revenue_by_year.sort_values(by="Year")
# Display the result
print(revenue_by_year)
```

```
   Year  VendorRevenue - USD
0  2018         5.877803e+05
1  2019         1.250904e+06
2  2020         1.290538e+06
3  2021         1.266611e+06
4  2022         1.579407e+06
5  2023         8.358715e+05
```

In [63]:

```
# Plot revenue trend over years
plt.figure(figsize=(10, 5))
plt.plot(revenue_by_year["Year"], revenue_by_year["VendorRevenue - USD"], marker="o", linestyle="-", color="blue")
plt.xlabel("Year")
plt.ylabel("Total Vendor Revenue (USD)")
plt.title("Revenue Trend Over Time (2019-2023)")
plt.xticks(revenue_by_year["Year"])  # Ensure all years are labeled
plt.grid(True, linestyle="--", alpha=0.7)
plt.show()
```

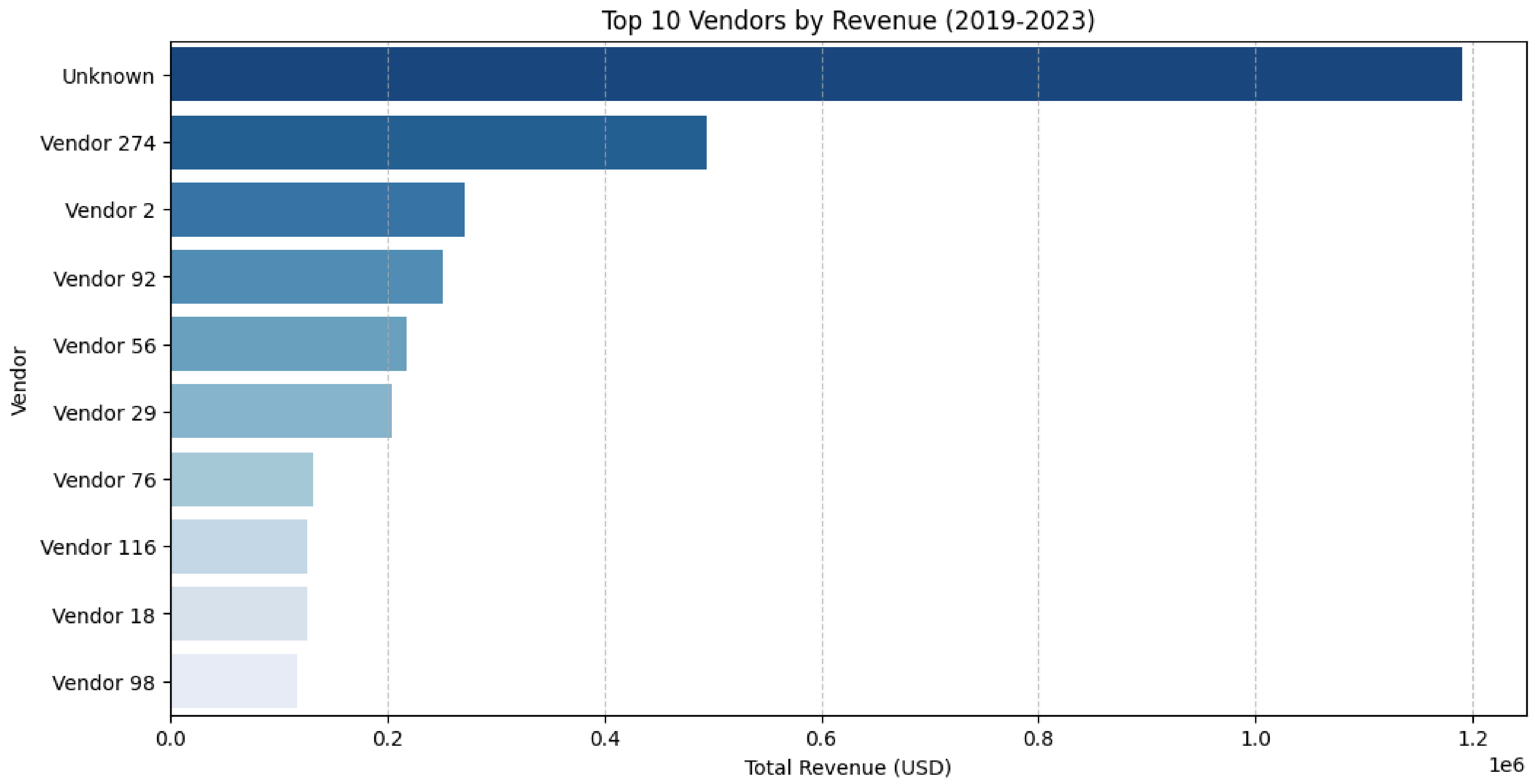


In [ ]:
```
#Market Share by Vendor
```

In [68]:
```
# Ensuring dataframe exists
if 'df' not in locals():
    print(" DataFrame 'df' not found! Make sure to load your dataset first.")
else:
    vendor_revenue = df.groupby("Vendor", as_index=False)["VendorRevenue - USD"].sum() # Group by Vendor and sum revenue
    vendor_revenue = vendor_revenue.sort_values(by="VendorRevenue - USD", ascending=False) # Sort vendors by revenue in descen
    print(vendor_revenue.head(10)) # Display the top 10 vendors
```

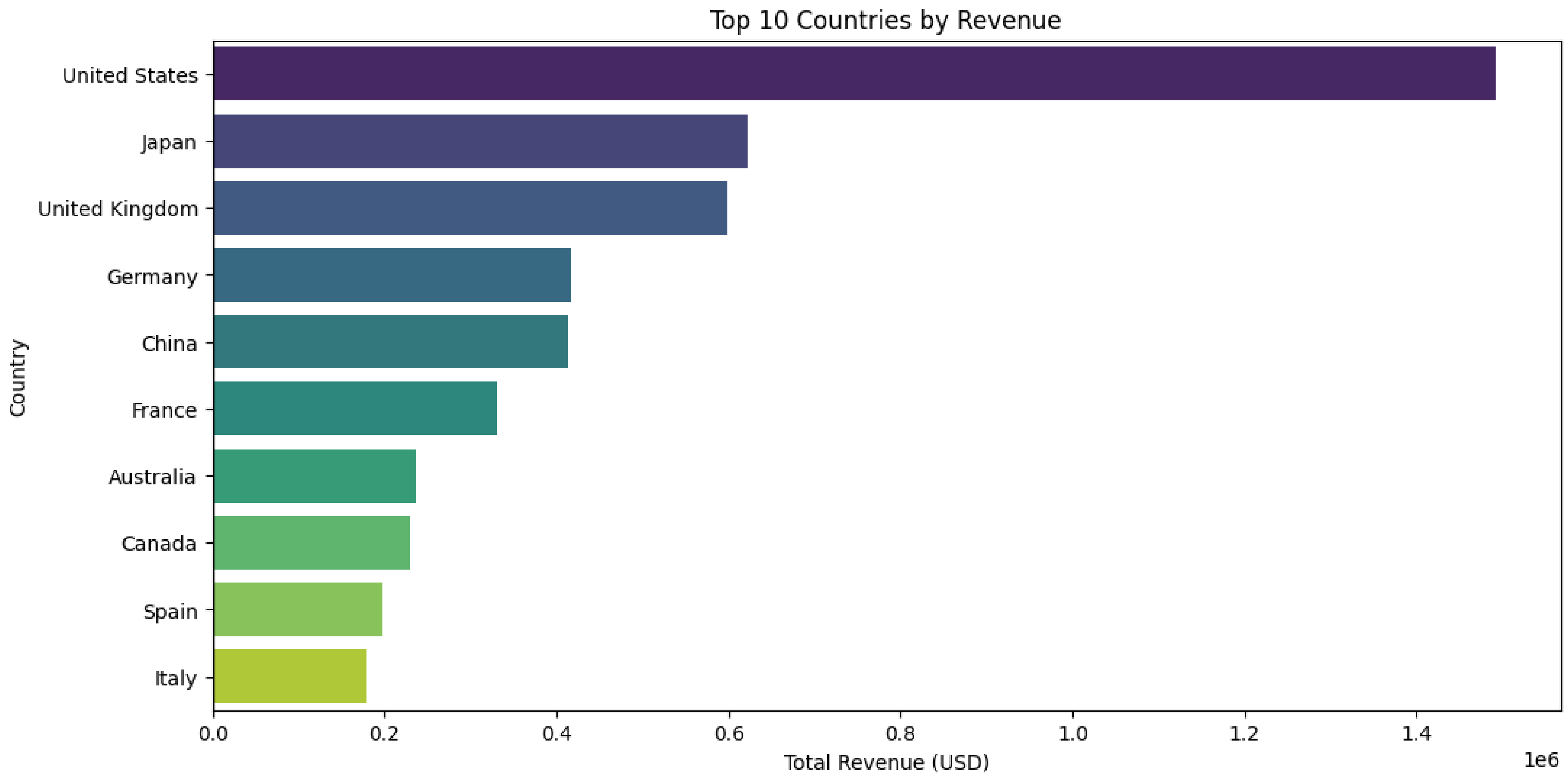
```
          Vendor  VendorRevenue - USD
0        Unknown         1.190275e+06
195   Vendor 274         4.946245e+05
112     Vendor 2         2.716997e+05
315    Vendor 92         2.510248e+05
275    Vendor 56         2.171571e+05
212    Vendor 29         2.046908e+05
297    Vendor 76         1.311397e+05
20    Vendor 116         1.269256e+05
90     Vendor 18         1.266250e+05
321    Vendor 98         1.166680e+05
```

In [73]:
```
top_vendors = vendor_revenue.head(10)
plt.figure(figsize=(12, 6))
sns.barplot(x="VendorRevenue - USD", y="Vendor", data=top_vendors, hue="Vendor", dodge=False, legend=False, palette="Blues_r")
plt.xlabel("Total Revenue (USD)")
plt.ylabel("Vendor")
plt.title("Top 10 Vendors by Revenue (2019-2023)")
plt.grid(axis="x", linestyle="--", alpha=0.7)
plt.show()
```



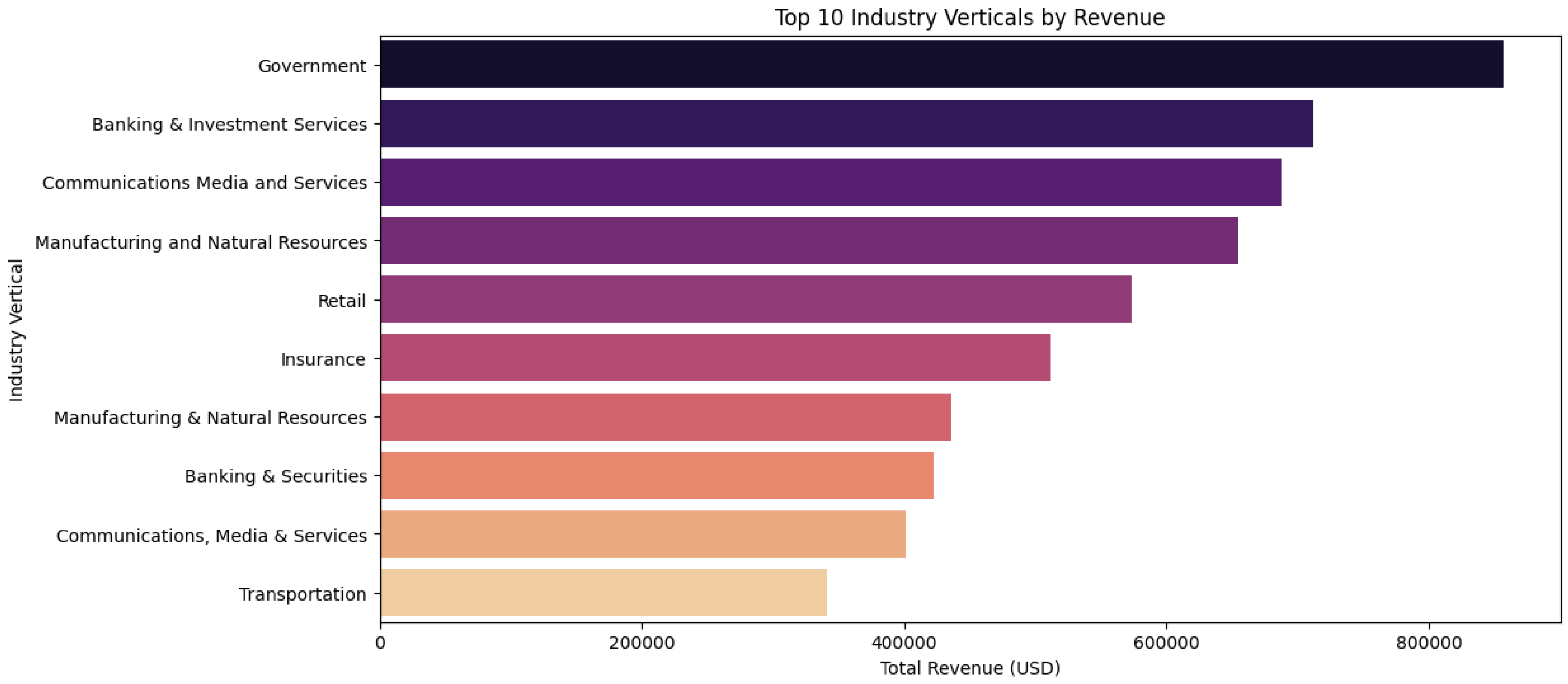
In [ ]:
```
#Revenue by Country
```

In [75]:
```
# Aggregate revenue by country
revenue_by_country = df.groupby("Country")["VendorRevenue - USD"].sum().reset_index()

# Sort in descending order and select top 10
top_countries = revenue_by_country.sort_values(by="VendorRevenue - USD", ascending=False).head(10)
plt.figure(figsize=(12, 6))
sns.barplot(x="VendorRevenue - USD", y="Country", data=top_countries, hue="Country", legend=False, palette="viridis")
plt.xlabel("Total Revenue (USD)")
plt.ylabel("Country")
plt.title("Top 10 Countries by Revenue")
plt.show()
```



In [ ]:
```
#Revenue By Industry Vertical
```

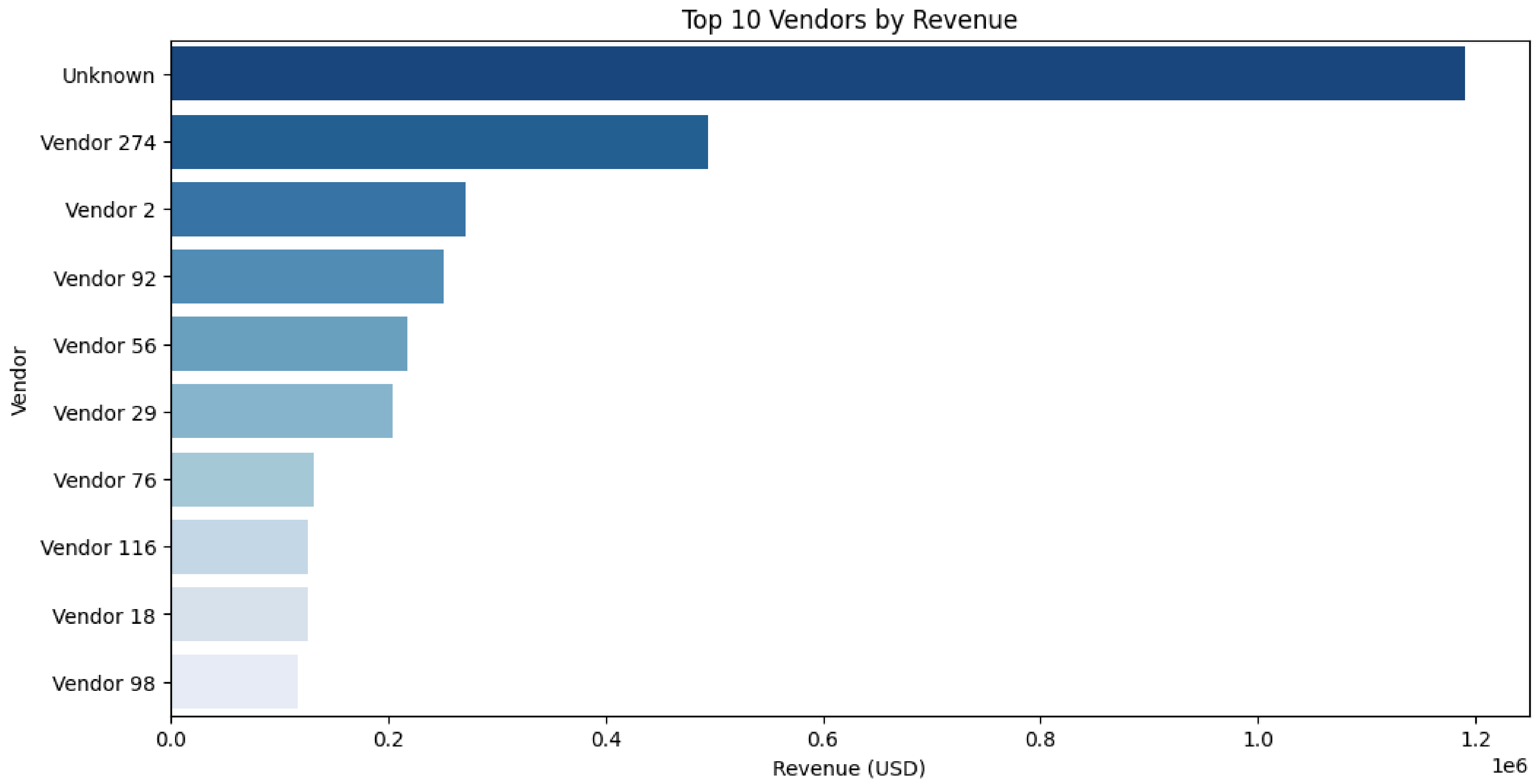
In [77]:
```
# Grouping data by Vertical and summing the revenue
vertical_revenue = df.groupby("Vertical")["VendorRevenue - USD"].sum().reset_index()
# Sorting to get the top 10 verticals
top_verticals = vertical_revenue.sort_values(by="VendorRevenue - USD", ascending=False).head(10)
plt.figure(figsize=(12, 6))
sns.barplot(x="VendorRevenue - USD", y="Vertical", data=top_verticals, hue="Vertical", palette="magma", legend=False)
plt.xlabel("Total Revenue (USD)")
plt.ylabel("Industry Vertical")
plt.title("Top 10 Industry Verticals by Revenue")
plt.show()
```



```
In [ ]: #Vendor performance across industries
```

```
In [89]: # Aggregate revenue by vendor
vendor_revenue = df.groupby("Vendor", as_index=False)["VendorRevenue - USD"].sum()
# Get top 10 vendors by revenue
top_vendors = vendor_revenue.sort_values(by="VendorRevenue - USD", ascending=False).head(10)
plt.figure(figsize=(12, 6))
sns.barplot(x="VendorRevenue - USD", y="Vendor", data=top_vendors, palette="Blues_r", hue=None, legend=False)
plt.xlabel("Revenue (USD)")
plt.ylabel("Vendor")
plt.title("Top 10 Vendors by Revenue")
plt.show()
```

```
C:\Users\hp\AppData\Local\Temp\ipykernel_9912\2229456751.py:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and se
t `legend=False` for the same effect.

  sns.barplot(x="VendorRevenue - USD", y="Vendor", data=top_vendors, palette="Blues_r", hue=None, legend=False)
```



```
In [ ]:
```