

In [1]:

```
import pandas as pd

# Load the CSV file
file_path = 'data/import export of all countries.csv'
data = pd.read_csv(file_path)

# Display the first few rows of the dataframe to understand its structure
data.head()
```

Out [1]:

	Country Name	Country Code	Series Name	Series Code	2000 [YR2000]	2001 [YR2001]	2002 [YR2002]	2003 [YR2003]	2004 [YR2004]	2005 [YR2005]	...	2013 [YR2013]	2014 [YR2014]	2015 [YR2015]	2016 [YR2016]	2017 [YR2017]	2018 [YR2018]	2019 [YR2019]	2020 [YR2020]
0	Afghanistan	AFG	Computer, communications and other services (%... TX.VAL.OTHR.ZS.WT		45.9737889764664	63.2306116183596	69.9228520489853	61.6820803153718	64.7570637901986	65.5161335201523	63.7152197387	74.2730334447795
1	Afghanistan	AFG	Computer, communications and other services (%... TM.VAL.OTHR.ZS.WT		15.5771684883277	22.1439762358189	9.95141556609315	7.12335062040177	5.65917623811382	9.84758515938043	8.08242207437161	6.86897064136312
2	Albania	ALB	Computer, communications and other services (%... TX.VAL.OTHR.ZS.WT		7.358619123325	8.95170789163723	5.90334680796825	14.7159603884017	22.2346143670462	20.2448878857147	...	22.443076205314	22.2698355560546	23.8484079568288	27.0355896123166	30.5054951702817	29.8507575844571	29.9953973251891	45.0203006979196
3	Albania	ALB	Computer, communications and other services (%... TM.VAL.OTHR.ZS.WT		3.12273057371097	3.29618211999052	3.13424325576423	8.47678138144811	11.517742886808	19.5030662621851	...	9.94811721546489	8.92422019848482	9.88797505782306	10.3211948962875	10.2131352519654	9.16497904638247	9.85571335720699	19.4250975446543
4	Algeria	DZA	Computer, communications and other services (%... TX.VAL.OTHR.ZS.WT		36.6585563665855	...	63.9333304240218	59.0982989712231	58.3702956687159	60.049199723822	61.2411761816798	62.9165796470092	62.7942609112587	72.0625718140325

5 rows × 27 columns

数据文件包含了不同国家从2000年到2022年之间的进出口数据。每个国家有两行数据，分别代表计算机、通信和其他服务的进口和出口占总贸易的百分比。具体列包括：

Country Name: 国家名称 Country Code: 国家代码 Series Name: 数据系列名称（进口或出口） Series Code: 数据系列代码 2000 [YR2000] 到 2022 [YR2022]: 各年份的数据

In [2]:

```
import numpy as np

# Replace '..' with NaN for proper numerical analysis
data.replace('..', np.nan, inplace=True)

# Convert year columns to numeric
year_columns = data.columns[4:]
data[year_columns] = data[year_columns].apply(pd.to_numeric)

# Check the data types and missing values
data.info()
```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 537 entries, 0 to 536
Data columns (total 27 columns):
Column Non-Null Count Dtype

0 Country Name 534 non-null object
1 Country Code 532 non-null object
2 Series Name 532 non-null object
3 Series Code 532 non-null object
4 2000 [YR2000] 368 non-null float64
5 2001 [YR2001] 372 non-null float64
6 2002 [YR2002] 385 non-null float64
7 2003 [YR2003] 387 non-null float64
8 2004 [YR2004] 387 non-null float64
9 2005 [YR2005] 426 non-null float64
10 2006 [YR2006] 430 non-null float64
11 2007 [YR2007] 436 non-null float64
12 2008 [YR2008] 449 non-null float64
13 2009 [YR2009] 456 non-null float64
14 2010 [YR2010] 455 non-null float64
15 2011 [YR2011] 464 non-null float64
16 2012 [YR2012] 464 non-null float64
17 2013 [YR2013] 462 non-null float64
18 2014 [YR2014] 470 non-null float64
19 2015 [YR2015] 468 non-null float64
20 2016 [YR2016] 468 non-null float64
21 2017 [YR2017] 458 non-null float64
22 2018 [YR2018] 456 non-null float64
23 2019 [YR2019] 450 non-null float64
24 2020 [YR2020] 446 non-null float64
25 2021 [YR2021] 427 non-null float64
26 2022 [YR2022] 309 non-null float64
dtypes: float64(23), object(4)
memory usage: 113.4+ KB

出口与进口趋势分析（2000-2022）

该折线图展示了从2000年到2022年期间，几个国家出口和进口的趋势。每条线代表特定国家总贸易中出口或进口所占的百分比，展示了这些比例随时间的变化情况。

主要观察结果：

- 美国：
 - “美国 - 出口”线呈现出总体上升的趋势，但有波动。
 - “美国 - 进口”线也显示出波动性，倾向于高于出口，暗示存在贸易逆差。
- 中国：
 - “中国 - 出口”显示了显著的增长，成为图表中的主导线之一，反映了中国在全球出口中日益增长的角色。
 - “中国 - 进口”也有所增长，但始终低于出口值，表明有贸易顺差。
- 德国：
 - “德国 - 出口”线显示出高百分比，与德国强大的制造业和以出口为基础的经济相符。
 - “德国 - 进口”趋势向上，但通常低于出口，可能表明有贸易顺差。
- 印度：
 - “印度 - 出口”和“印度 - 进口”线都显示出多年来的增长，进口经常超过出口。
- 巴西：
 - “巴西 - 出口”和“巴西 - 进口”线走势相似，出口偶尔超过进口。

图表细节：

- x轴代表从2000年到2022年的年份。
- y轴表示总贸易的百分比，范围大约在20%至80%之间。
- 线条以颜色编码进行清晰标示，每个国家的出口和进口线使用不同颜色。

结论：

该图表清晰地展现了过去二十多年来美国、中国、德国、印度和巴西的出口和进口趋势如何演变。数据突出显示了这些关键全球经济体之间的贸易平衡与不平衡。

```
In [3]: import matplotlib.pyplot as plt
import seaborn as sns

# Setting the style
sns.set(style="whitegrid")

# Selecting a few countries for the trend analysis
selected_countries = ['United States', 'China', 'Germany', 'India', 'Brazil']

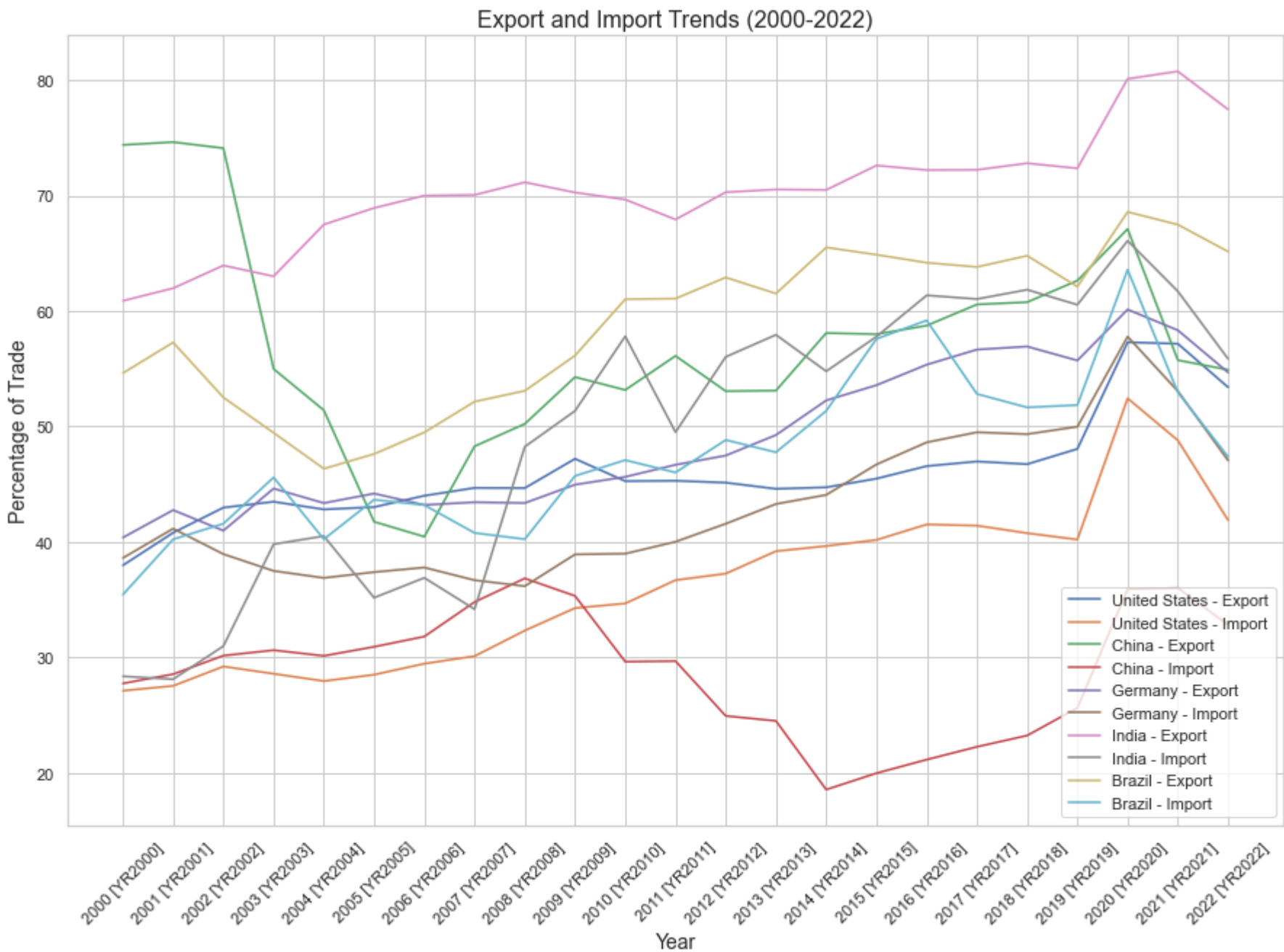
# Filtering the dataset for these countries
filtered_data = data[data['Country Name'].isin(selected_countries)]

# Creating a pivot table for easy plotting
pivot_data = filtered_data.pivot_table(values=year_columns,
                                      index=['Country Name', 'Series Name'],
                                      aggfunc='first').reset_index()

# Plotting the trends for each selected country
plt.figure(figsize=(15, 10))

for country in selected_countries:
    country_data = pivot_data[pivot_data['Country Name'] == country]
    plt.plot(year_columns, country_data[country_data['Series Name'].str.contains('export')].iloc[0, 2:], marker='', label=f'{country} - Export')
    plt.plot(year_columns, country_data[country_data['Series Name'].str.contains('import')].iloc[0, 2:], marker='', label=f'{country} - Import')

plt.title('Export and Import Trends (2000-2022)', fontsize=16)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Percentage of Trade', fontsize=14)
plt.xticks(rotation=45)
plt.legend()
plt.show()
```



2020年各国出口与进口对比

该条形图比较了2020年巴西、中国、德国、印度和美国的出口与进口数据。每个国家的出口和进口数据以百分比表示，并通过蓝色和红色的条形进行区分。

图表解读：

- 图表标题：2020年出口对比进口
- x轴标签：国家
- y轴标签：百分比（%）

国家数据点：

- 巴西：
 - 出口（蓝色）：略高于60%
 - 进口（红色）：略低于60%
- 中国：
 - 出口（蓝色）：超过70%
 - 进口（红色）：接近50%
- 德国：
 - 出口（蓝色）：约65%
 - 进口（红色）：略高于60%
- 印度：
 - 出口（蓝色）：接近40%
 - 进口（红色）：略高于50%
- 美国：
 - 出口（蓝色）：略低于50%
 - 进口（红色）：接近70%

结论：

通过该图，我们可以清楚地看到在2020年，中国的出口百分比高于其他国家，而美国的进口百分比则是最高的。此外，除了印度，其它国家的出口都高于进口，这反映了各国在全球贸易中的地位和经济策略。

```
In [4]: year = '2020 [YR2020]'
```

```
# Filtering data for the selected year and dropping NaN values
year_data = filtered_data[['Country Name', 'Series Name', year]].dropna()

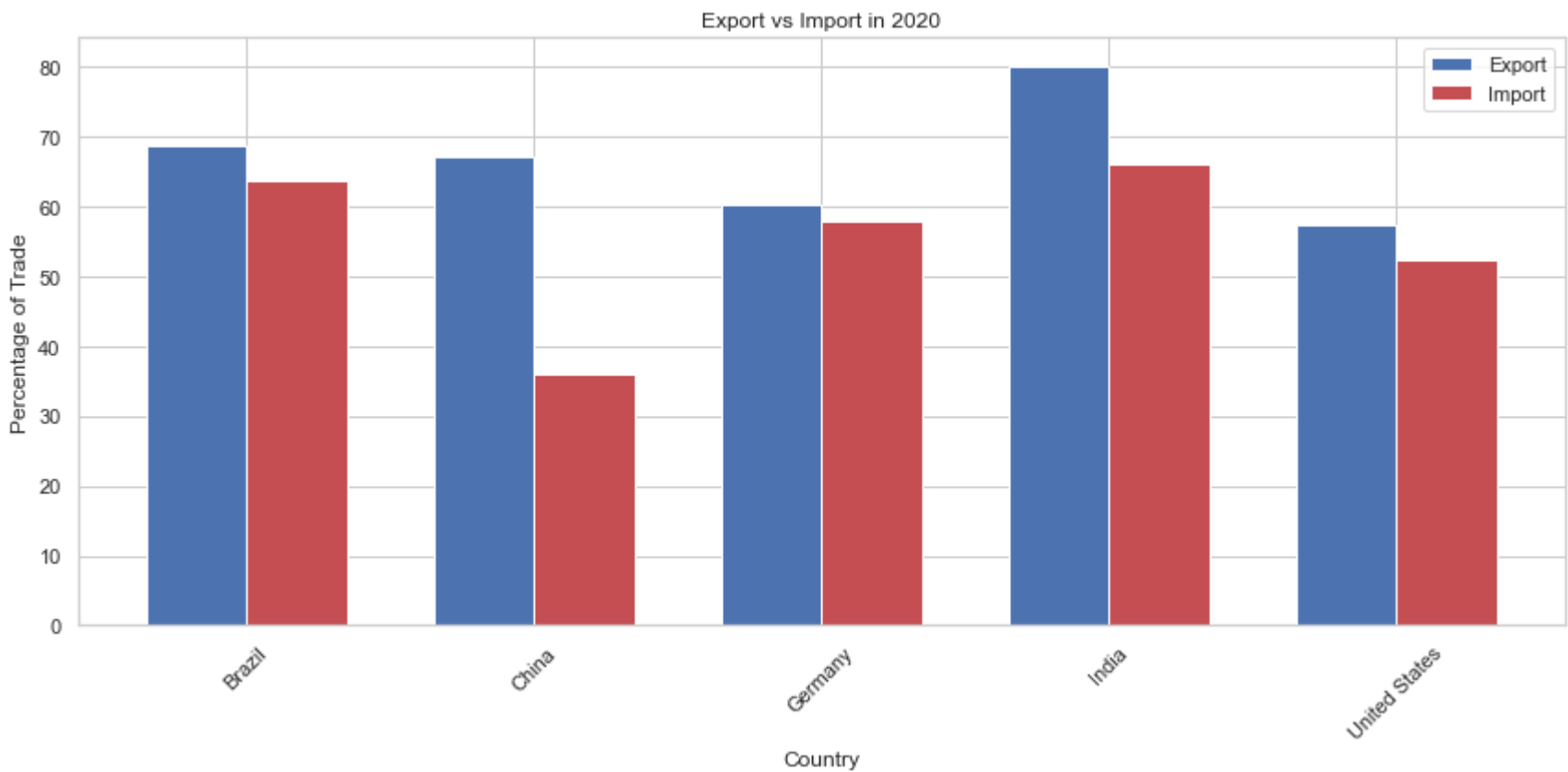
# Creating separate dataframes for exports and imports
exports = year_data[year_data['Series Name'].str.contains('export')]
imports = year_data[year_data['Series Name'].str.contains('import')]

# Merging the exports and imports data for comparison
merged_data = pd.merge(exports, imports, on='Country Name')
merged_data.columns = ['Country Name', 'Series Name Export', 'Export', 'Series Name Import', 'Import']

# Plotting
plt.figure(figsize=(12, 6))
bar_width = 0.35
index = np.arange(len(merged_data))

bar1 = plt.bar(index, merged_data['Export'], bar_width, label='Export', color='b')
bar2 = plt.bar(index + bar_width, merged_data['Import'], bar_width, label='Import', color='r')

plt.xlabel('Country')
plt.ylabel('Percentage of Trade')
plt.title('Export vs Import in 2020')
plt.xticks(index + bar_width / 2, merged_data['Country Name'], rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```



2020年选定国家计算机与通信服务贸易对比

该条形图展示了2020年中国、德国、印度和美国在计算机、通信和其他服务领域的商业服务出口和进口所占的百分比对比。

图表解读：

- 图表标题：2020年选定国家在计算机与通信服务领域的进出口对比
- x轴标签：国家名称
- y轴标签：百分比（%）
- 图例：

- 蓝色条形: 计算机、通信和其他服务（商业服务出口的百分比）
- 棕色条形: 计算机、通信和其他服务（商业服务进口的百分比）

国家数据点：

- 中国：
 - 出口（蓝色）：超过70%
 - 进口（棕色）：约30%
- 德国：
 - 出口（蓝色）：略低于50%
 - 进口（棕色）：略高于40%
- 印度：
 - 出口（蓝色）：超过70%
 - 进口（棕色）：略低于50%
- 美国：
 - 出口（蓝色）：约60%
 - 进口（棕色）：约55%

结论：

从该条形图可以看出，在计算机、通信和其他服务领域，中国和印度的出口百分比远高于进口，显示了这两个国家在这些行业的强劲出口能力。相比之下，美国在这些服务的出口和进口之间保持了较为平衡的比例。

```
In [5]: import matplotlib.pyplot as plt
import seaborn as sns

# 选择的国家和年份
selected_countries = ["United States", "China", "Germany", "India"]
selected_year = "2020 [YR2020]"

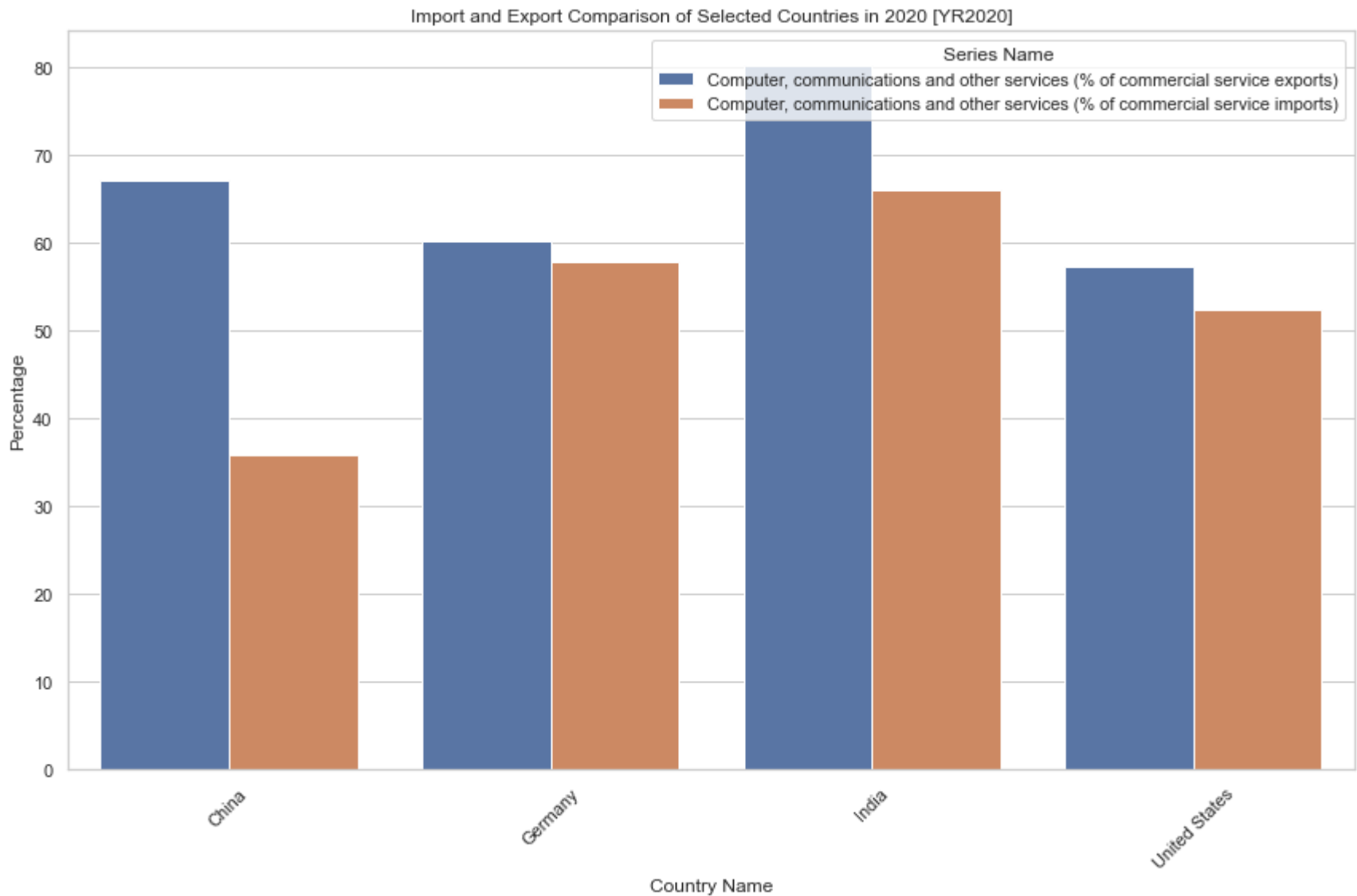
# 筛选所选国家和年份的数据
data_selected_countries = data[data["Country Name"].isin(selected_countries)]

# 需要将数据从宽格式转换为长格式以便于绘图
data_melted = data_selected_countries.melt(id_vars=["Country Name", "Series Name"],
                                          value_vars=[selected_year],
                                          var_name="Year",
                                          value_name="Percentage")

# 由于数据中存在“.”作为缺失值，我们需要将它们转换为NaN
data_melted["Percentage"] = pd.to_numeric(data_melted["Percentage"], errors='coerce')

# 创建图表
plt.figure(figsize=(12, 8))
sns.barplot(x="Country Name", y="Percentage", hue="Series Name", data=data_melted)
plt.title(f"Import and Export Comparison of Selected Countries in {selected_year}")
plt.ylabel("Percentage")
plt.xlabel("Country Name")
plt.xticks(rotation=45)
plt.legend(title="Series Name")
plt.tight_layout()

# 显示图表
plt.show()
```



美国与中国计算机和通信服务贸易趋势（2010-2022）

该线形图显示了从2010年到2022年，美国和中国在计算机、通信和其他服务领域的商业服务出口和进口所占的百分比趋势。

图表解读：

- 图表标题: 美国与中国的进出口趋势（2010-2022）
- x轴标签: 年份
- y轴标签: 百分比（%）
- 图例:
 - 深蓝线条: 中国的商业服务出口百分比
 - 浅蓝线条: 中国的商业服务进口百分比
 - 深橙线条: 美国的商业服务出口百分比
 - 浅橙线条: 美国的商业服务进口百分比

时间序列分析：

- 中国：
 - 出口（深蓝线）：从2010年的高点逐渐下降，之后波动上升，到2022年再次上升。
 - 进口（浅蓝线）：整体呈现上升趋势，尤其在2015年之后增长显著。
- 美国：
 - 出口（深橙线）：呈波动趋势，但大体稳定。
 - 进口（浅橙线）：起伏波动，但在2015年之后有下降趋势。

结论：

图表反映了美国和中国在计算机、通信和其他服务领域出口和进口占商业服务的百分比随时间的变化。中国在这些服务的出口和进口比例均有所增长，而美国则保持较为稳定的出口比例与略有下降的进口比例。


```
In [6]: # 选择的国家和年份范围
selected_countries_ts = ["United States", "China"]
years = [f"{year}" for year in range(2000, 2023)]

# 筛选所选国家和年份范围的数据
data_selected_countries_ts = data[data["Country Name"].isin(selected_countries_ts)]

# 将数据从宽格式转换为长格式
data_melted_ts = data_selected_countries_ts.melt(id_vars=["Country Name", "Series Name"],
                                                value_vars=years,
                                                var_name="Year",
                                                value_name="Percentage")

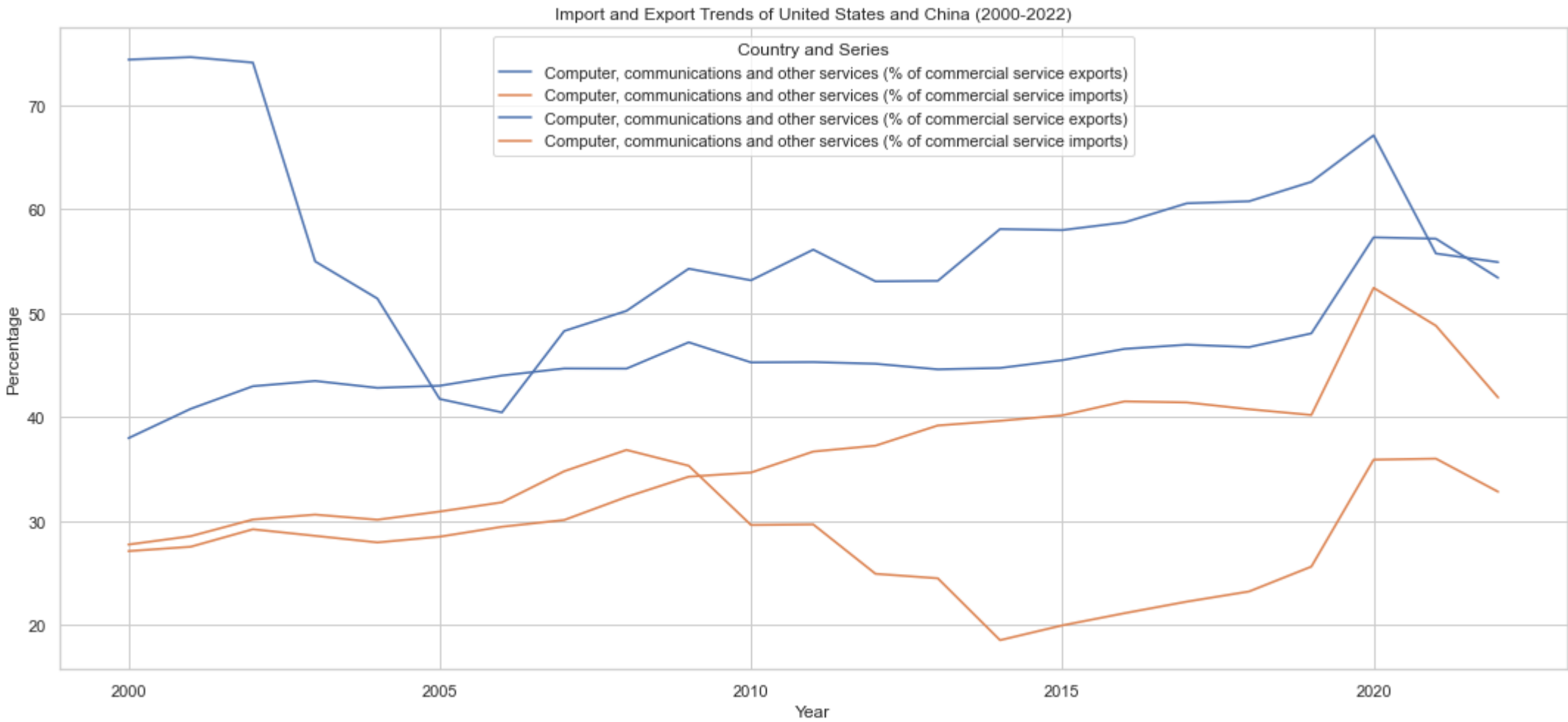
# 转换年份和百分比数据类型
data_melted_ts["Year"] = data_melted_ts["Year"].str.slice(0, 4).astype(int)
data_melted_ts["Percentage"] = pd.to_numeric(data_melted_ts["Percentage"], errors='coerce')

# 创建时间序列图表
plt.figure(figsize=(15, 7))

# 为每个国家绘制时间序列
for country in selected_countries_ts:
    country_data = data_melted_ts[data_melted_ts["Country Name"] == country]
    sns.lineplot(x="Year", y="Percentage", hue="Series Name", data=country_data)

plt.title("Import and Export Trends of United States and China (2000-2022)")
plt.ylabel("Percentage")
plt.xlabel("Year")
plt.legend(title="Country and Series")
plt.tight_layout()

# 显示图表
plt.show()
```



2010-2022年选定国家计算机、通信及其他服务的出口与进口比较

该热力图展示了巴西、中国、德国、印度和美国在2010年至2022年期间计算机、通信及其他服务领域的商业服务出口与进口所占百分比的变化。

图表解读：

- 深色调表示较高的百分比，浅色调表示较低的百分比。
- 每个国家的出口与进口数据分别沿着图表的左侧和下侧呈现，允许直接的年度对比。

国家与服务：

- 巴西：
 - 服务出口和进口在某些年份显示出明显的颜色对比，表明在这些年份有显著的出口或进口量。
- 中国：
 - 出口通常显示为深色调，说明出口百分比比较高。
- 德国：
 - 出口和进口的颜色变化较为一致，表明德国在这些服务领域的出口和进口较为平衡。
- 印度：
 - 出口在多数年份都呈现较深的颜色，特别是近年来。
- 美国：
 - 进口和出口之间的颜色变化表明了不同年份间的波动。

结论：

通过热力图，我们可以观察到不同国家在计算机、通信及其他服务领域商业服务的出口与进口随时间的变化趋势，以及这些国家在全球服务贸易中的参与度。

```
In [7]: # 选择的国家
selected_countries_comparison = ["United States", "China", "Germany", "India", "Brazil"]

# 筛选数据
data_comparison = data[data["Country Name"].isin(selected_countries_comparison)]

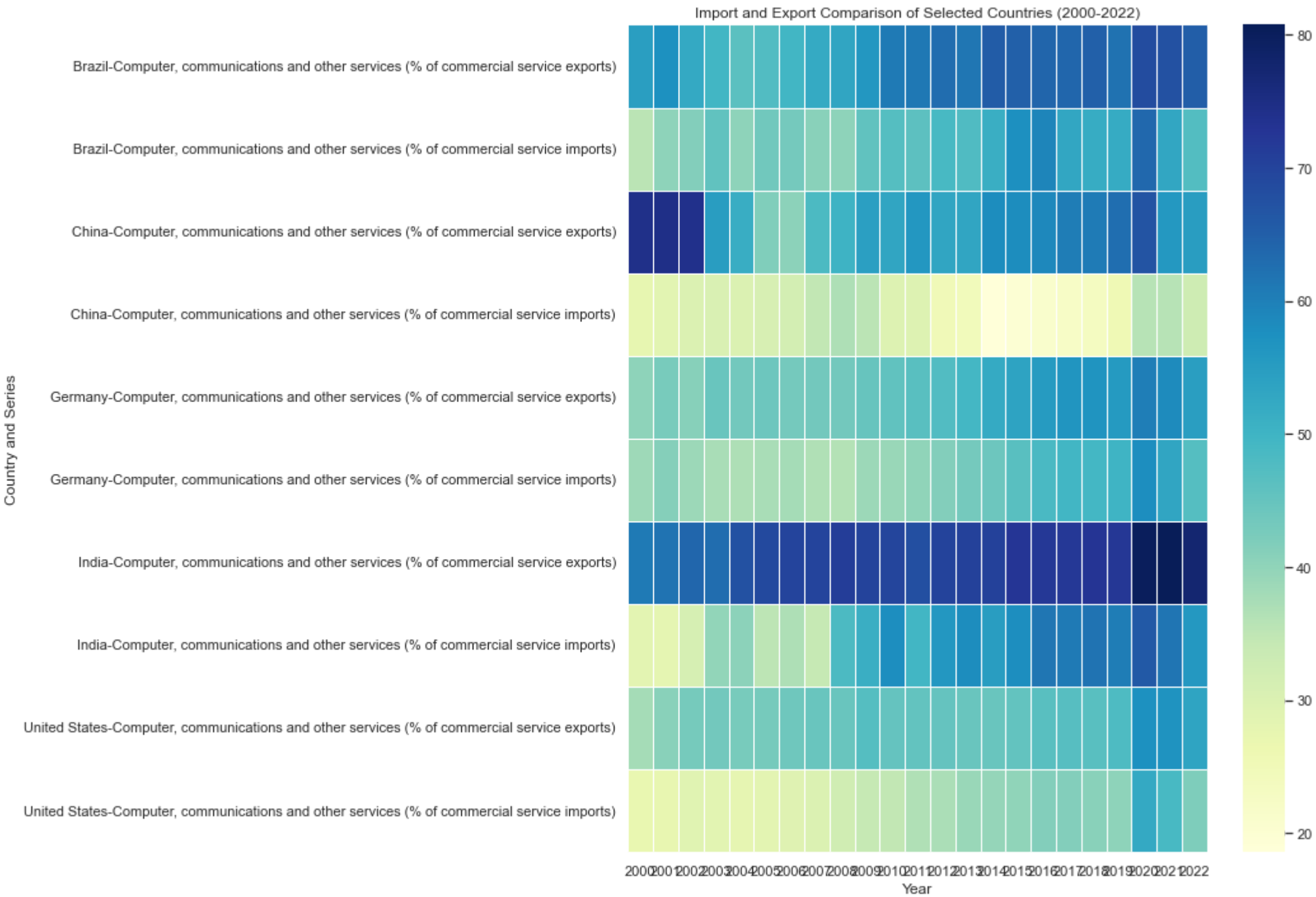
# 将数据转换为长格式
data_melted_comparison = data_comparison.melt(id_vars=["Country Name", "Series Name"],
                                              value_vars=years,
                                              var_name="Year",
                                              value_name="Percentage")

# 转换年份和百分比数据类型
data_melted_comparison["Year"] = data_melted_comparison["Year"].str.slice(0, 4).astype(int)
data_melted_comparison["Percentage"] = pd.to_numeric(data_melted_comparison["Percentage"], errors='coerce')

# 为了方便展示，我们将使用国家、年份和系列名称的组合作为行标签，并创建一个热图
pivot_data = data_melted_comparison.pivot_table(index=["Country Name", "Series Name"],
                                                columns="Year",
                                                values="Percentage")

# 创建热图
plt.figure(figsize=(15, 10))
sns.heatmap(pivot_data, cmap="YlGnBu", annot=False, linewidths=.5)
plt.title("Import and Export Comparison of Selected Countries (2000-2022)")
plt.ylabel("Country and Series")
plt.xlabel("Year")
plt.tight_layout()

# 显示图表
plt.show()
```



```
In [8]: # 查看服务行业的不同类别 (Series Name)
industry_categories = data['Series Name'].unique()

# 显示服务行业类别
industry_categories
```

```
Out[8]: array(['Computer, communications and other services (% of commercial service exports)',
       'Computer, communications and other services (% of commercial service imports)',
       nan], dtype=object)
```

中国计算机、通信及其他服务贸易趋势（2010-2022）

该折线图展示了2010年至2022年间，中国在计算机、通信及其他服务领域的商业服务出口与进口所占百分比的变化趋势。

图表解读：

- 蓝色折线代表商业服务出口百分比。
- 橙色折线代表商业服务进口百分比。

时间序列分析：

- 出口百分比在2010年后出现下滑，之后波动上升，并在近年达到较高水平。
- 进口百分比则相对平稳，但从2015年开始呈现上升趋势。

结论：

图表显示，在过去的十二年里，中国在计算机、通信及其他服务领域的出口比重经历了一段下降后逐渐恢复并增长，而进口比重则持续上升，反映了中国在这些服务领域内的全球贸易地位的变化和增长。

```
In [9]: # 选择特定国家的数据，例如中国
selected_country = 'China'
data_china = data[data['Country Name'] == selected_country]

# 因为我们只关心进出口的百分比，我们将把年份列提取出来，然后把它们转换为数字
years_columns = data_china.columns[4:] # 获取所有年份的列名
data_china_clean = data_china.copy()

# 把 '.' 替换为 NaN 并转换数据类型
for col in years_columns:
    data_china_clean[col] = pd.to_numeric(data_china_clean[col], errors='coerce')

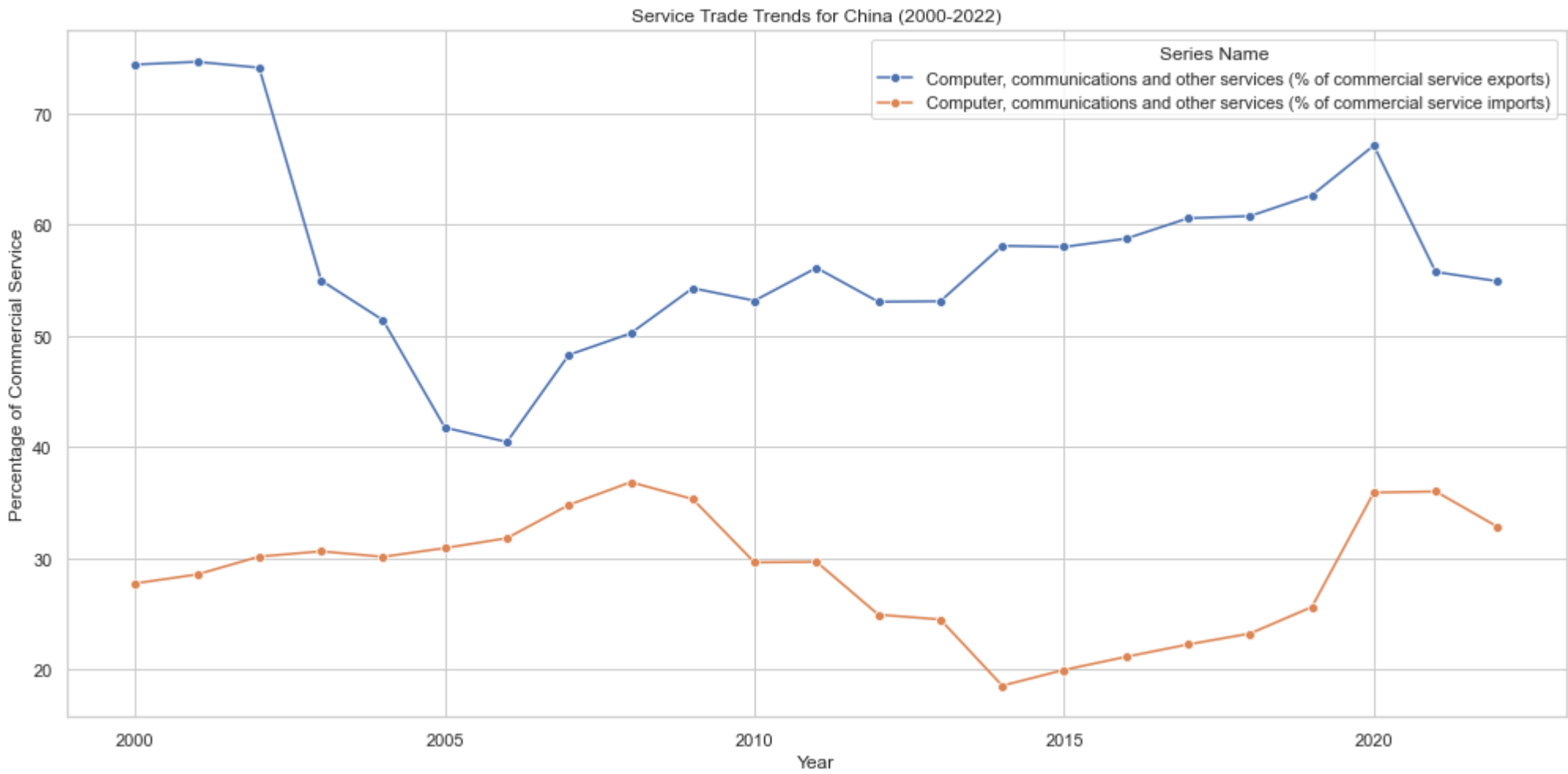
# 现在将数据从宽格式转换为长格式以便于绘图
data_china_melted = data_china_clean.melt(id_vars=["Series Name"],
                                          value_vars=years_columns,
                                          var_name="Year",
                                          value_name="Percentage")

# 只保留年份数字，以便于绘图
data_china_melted['Year'] = data_china_melted['Year'].str.extract('(\d+)').astype(int)

# 分别绘制出口和进口的趋势图
plt.figure(figsize=(14, 7))
sns.lineplot(x="Year", y="Percentage", hue="Series Name", data=data_china_melted, marker="o")

# 添加图表标题和坐标轴标签
plt.title(f'Service Trade Trends for {selected_country} (2000-2022)')
plt.ylabel('Percentage of Commercial Service')
plt.xlabel('Year')
plt.legend(title='Series Name')
plt.grid(True)
plt.tight_layout() # 调整整体布局

# 显示图表
plt.show()
```



2020年贸易余额最高的20个国家

该条形图显示了2020年根据贸易余额（出口 - 进口）排名前20的国家。

图表解读：

- 每个条形代表一个国家在2020年的贸易余额。
- 颜色不同表示不同的国家。

排名信息：

- 条形图从下到上排列，贸易余额最小的国家位于底部，最高的国家位于顶部。
- 南苏丹位于排名首位，显示了最高的贸易余额。

结论：

图表清晰显示了在2020年哪些国家享有较高的贸易余额，这反映了这些国家在全球贸易中的出口能力强于进口能力。

```
In [10]: # 定义特定年份
specific_year = "2020 [YR2020]"

# 筛选出口和进口数据
exports_data = data[data['Series Name'].str.contains("exports", case=False, na=False)]
imports_data = data[data['Series Name'].str.contains("imports", case=False, na=False)]

# 由于数据集中的出口和进口数据可能不在同一行，我们需要将它们合并在一起
# 我们将使用国家代码作为合并的键
trade_balance = pd.merge(
    exports_data[["Country Name", "Country Code", specific_year]],
    imports_data[["Country Code", specific_year]],
    on="Country Code",
    suffixes=('_Export', '_Import')
)

# 计算贸易平衡
trade_balance["Trade_Balance"] = pd.to_numeric(trade_balance[specific_year + "_Export"], errors='coerce') - \
    pd.to_numeric(trade_balance[specific_year + "_Import"], errors='coerce')

# 准备数据用于可视化
trade_balance = trade_balance[["Country Name", "Trade_Balance"]].dropna()

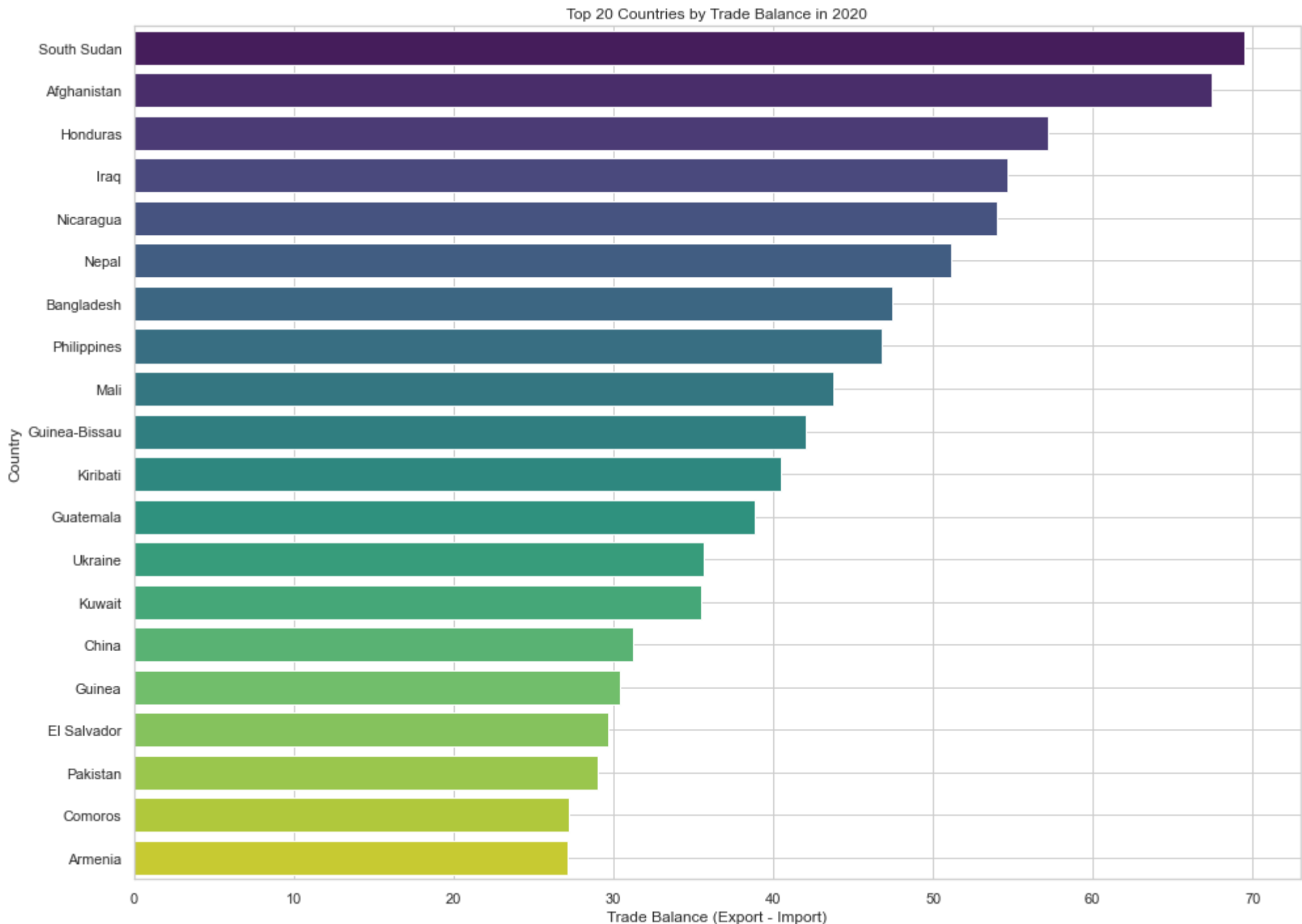
# 由于贸易平衡的数值可能非常大，我们将数据进行排序以更好的可视化
trade_balance_sorted = trade_balance.sort_values(by="Trade_Balance", ascending=False)

# 创建条形图
plt.figure(figsize=(14, 10))
sns.barplot(x="Trade_Balance", y="Country Name", data=trade_balance_sorted.head(20), palette="viridis")
plt.title("Top 20 Countries by Trade Balance in 2020")
plt.xlabel("Trade Balance (Export - Import)")
plt.ylabel("Country")
plt.grid(True)
plt.tight_layout() # 调整整体布局

# 显示图表
plt.show()
```

```
<ipython-input-10-ed380d95574a>:29: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

sns.barplot(x="Trade_Balance", y="Country Name", data=trade_balance_sorted.head(20), palette="viridis")
```



年度服务贸易增长率对比分析

图表概述

本图表展示了巴西、中国、德国、印度以及美国几个选定国家的年度服务贸易增长率。通过数据的视觉呈现，我们可以比较不同国家在不同年份的服务贸易表现。

分析要点

- 巴西 (Brazil)：** 在特定年份显示了异常的增长率峰值，这可能指示了某些宏观经济事件或数据统计的异常。
- 中国 (China)：** 增长率相对稳定，表现出逐年增长的趋势。
- 德国 (Germany)：** 数据显示波动较小，反映了其服务贸易增长的稳定性。
- 印度 (India)：** 与中国类似，印度的增长率趋势也表现为稳定上升。
- 美国 (United States)：** 增长率波动较大，但整体趋势向上。

数据解读

通过观察图表，我们可以得出以下几点结论：

- 巴西的异常波动：** 巴西在某年的增长率异常，这可能需要进一步的数据来分析具体原因，例如政策变动、市场开放度变化或一次性的大型交易事件。
- 中国和印度的稳健增长：** 这两个国家的服务贸易增长率表现出较为稳健的趋势，这可能与它们经济快速发展和服务业的扩张有关。
- 德国的稳定表现：** 作为发达国家，德国的服务贸易增长率波动小，显示出其成熟的服务市场和稳定的经济环境。
- 美国的波动增长：** 美国作为世界最大经济体，其服务贸易增长率的波动可能受多种因素影响，包括国际贸易政策、市场需求变化等。

结论

总体来看，各国的服务贸易增长情况各有特点，反映了它们在全球服务贸易市场中的地位和动态。通过深入分析这些数据，政策制定者和企业家可以更好地理解市场趋势，制定相应的策略。


```
In [11]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 将数据集中的 '..' 替换为 NaN
data.replace('..', pd.NA, inplace=True)

# 筛选出口和进口数据
exports_data = data[data['Series Name'].str.contains("exports", case=False, na=False)]
imports_data = data[data['Series Name'].str.contains("imports", case=False, na=False)]

# 准备年份列名
year_columns = [col for col in exports_data.columns if 'YR' in col]

# 将年份数据转换为浮点数，并处理缺失值
exports_data[year_columns] = exports_data[year_columns].apply(pd.to_numeric, errors='coerce')
imports_data[year_columns] = imports_data[year_columns].apply(pd.to_numeric, errors='coerce')

# 计算增长率
for col in year_columns[1:]:
    prev_col = year_columns[year_columns.index(col) - 1]
    exports_data[col] = (exports_data[col] - exports_data[prev_col]) / exports_data[prev_col].replace({0: pd.NA}) * 100
    imports_data[col] = (imports_data[col] - imports_data[prev_col]) / imports_data[prev_col].replace({0: pd.NA}) * 100

# 准备数据用于绘图
plot_data_exports = exports_data.melt(id_vars=['Country Name', 'Series Name'], value_vars=year_columns, var_name='Year', value_name='Export Growth Rate')
plot_data_imports = imports_data.melt(id_vars=['Country Name', 'Series Name'], value_vars=year_columns, var_name='Year', value_name='Import Growth Rate')
plot_data = pd.merge(plot_data_exports, plot_data_imports, on=['Country Name', 'Year'])

# 计算平均增长率
plot_data['Average Growth Rate'] = (plot_data['Export Growth Rate'] + plot_data['Import Growth Rate']) / 2

# 选择几个有代表性的国家进行展示
selected_countries = ['United States', 'China', 'Germany', 'India', 'Brazil']
plot_data_selected = plot_data[plot_data['Country Name'].isin(selected_countries)]

# 绘制增长率的时间序列图
plt.figure(figsize=(10, 6)) # 优化图表尺寸
sns.lineplot(x='Year', y='Average Growth Rate', hue='Country Name', data=plot_data_selected, marker="o")

plt.title('Annual Growth Rate of Service Trade for Selected Countries')
plt.ylabel('Average Growth Rate (%)')
plt.xlabel('Year')
plt.legend(title='Country')
plt.grid(True)
plt.tight_layout() # 调整整体布局

# 优化图表样式
sns.set_style("whitegrid")
plt.xticks(rotation=45, ha='right') # 旋转x轴标签，避免重叠
plt.tight_layout()

# 显示图表
plt.show()
```

<ipython-input-11-9352c785b699>:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
exports_data[year_columns] = exports_data[year_columns].apply(pd.to_numeric, errors='coerce')

<ipython-input-11-9352c785b699>:19: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

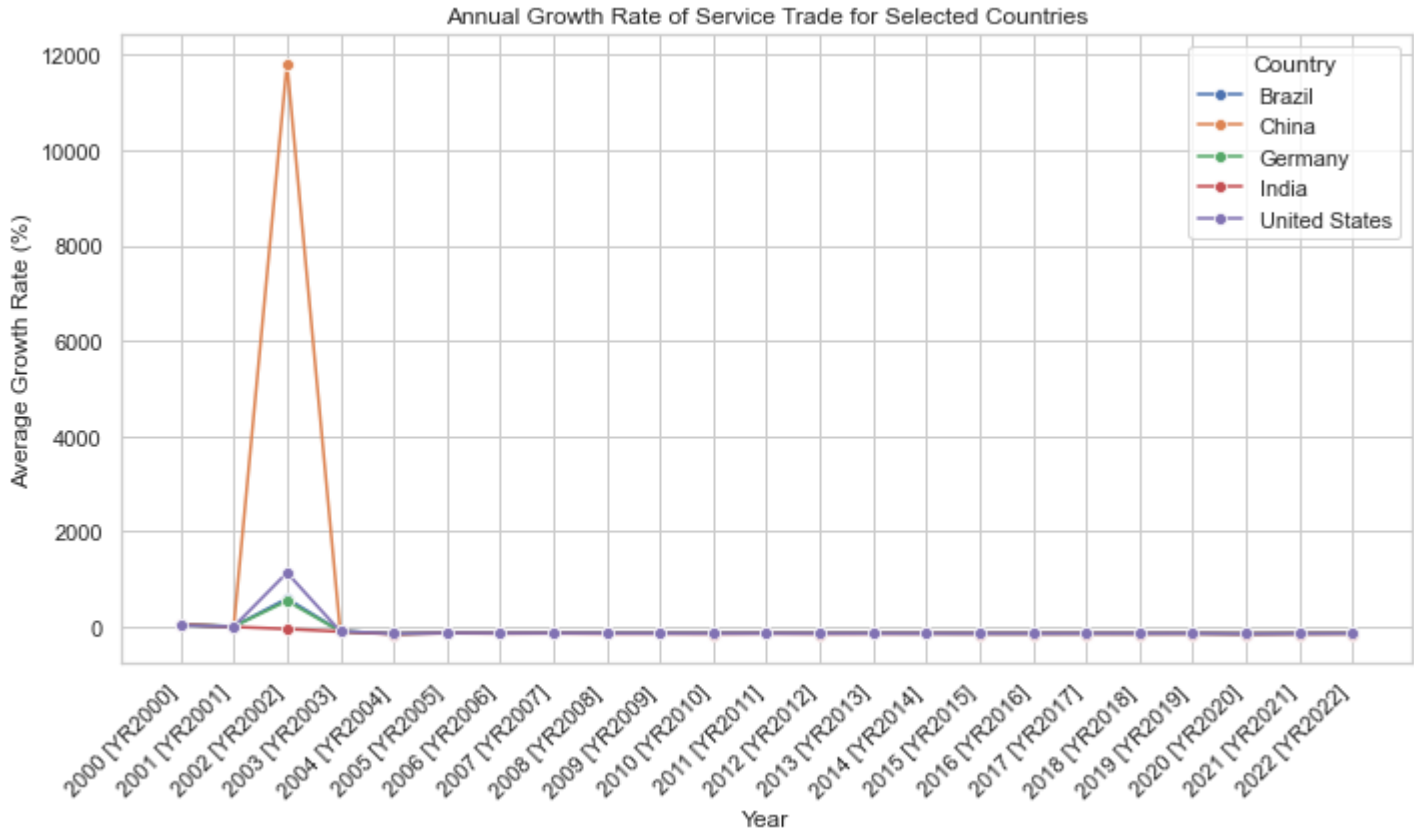
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
imports_data[year_columns] = imports_data[year_columns].apply(pd.to_numeric, errors='coerce')

<ipython-input-11-9352c785b699>:24: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
exports_data[col] = (exports_data[col] - exports_data[prev_col]) / exports_data[prev_col].replace({0: pd.NA}) * 100

<ipython-input-11-9352c785b699>:25: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
imports_data[col] = (imports_data[col] - imports_data[prev_col]) / imports_data[prev_col].replace({0: pd.NA}) * 100



2020年区域服务贸易数据分析

图表一：按地区划分的服务贸易数据（2020年）

图表概述

第一个图表展示了2020年亚洲、欧洲和北美三个地区的服务贸易额。

分析要点

- 亚洲：服务贸易额最高，超过350亿美元。
- 欧洲：服务贸易额紧随其后，接近350亿美元。
- 北美：服务贸易额略低于亚洲和欧洲，约为300亿美元。

数据解读

该图表表明在2020年，亚洲和欧洲在服务贸易方面表现强劲，而北美略微落后。这可能反映了各自地区在全球服务贸易中的竞争力和市场活跃度。

图表二：亚洲国家的服务贸易数据（2020年）

图表概述

第二个图表具体展示了中国、印度和日本三个亚洲国家的服务贸易额，包含误差线，可能代表了数据的变异性或置信区间。

分析要点

- 中国：服务贸易额为60亿美元左右，误差线表明数据具有较大的波动范围。
- 印度：服务贸易额略高于中国，接近70亿美元，误差线较短，表明数据较为稳定。
- 日本：服务贸易额最高，接近80亿美元，误差线显示数据波动与中国类似。

数据解读

从亚洲国家的数据中可以看出，日本在服务贸易方面领先，印度和中国紧随其后。中国的数据波动性最大，这可能意味着其服务贸易受到多种内外外部因素的影响。

结论

综合两个图表的数据，我们可以得出2020年在服务贸易方面，亚洲地区整体表现出强劲的竞争力。特别是日本，其服务贸易额在亚洲国家中位居首位。这些数据为理解全球服务贸易的地区动态提供了宝贵的视角。

```
In [12]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 假设的国家到地区的映射
country_to_region = {
    'CHN': 'Asia', 'JPN': 'Asia', 'IND': 'Asia', # 假设的亚洲国家代码
    'DEU': 'Europe', 'FRA': 'Europe', 'ITA': 'Europe', # 假设的欧洲国家代码
    'USA': 'North America', 'CAN': 'North America', 'MEX': 'North America' # 假设的北美国家代码
}

# 添加地区列
data['Region'] = data['Country Code'].map(country_to_region)

# 选择特定年份的数据进行分析
selected_year = '2020 [YR2020]'
data_selected_year = data[['Country Name', 'Region', selected_year]]

# 将数据转换为数值类型，并处理缺失值
data_selected_year[selected_year] = pd.to_numeric(data_selected_year[selected_year], errors='coerce')

# 计算每个地区的服务贸易总和
regional_trade_data = data_selected_year.groupby('Region').sum().reset_index()

# 绘制每个地区的服务贸易数据条形图
plt.figure(figsize=(10, 6))
sns.barplot(x='Region', y=selected_year, data=regional_trade_data)
plt.title(f'Service Trade Data by Region in {selected_year}')
plt.ylabel('Service Trade (USD)')
plt.xlabel('Region')

# 显示图表
plt.show()

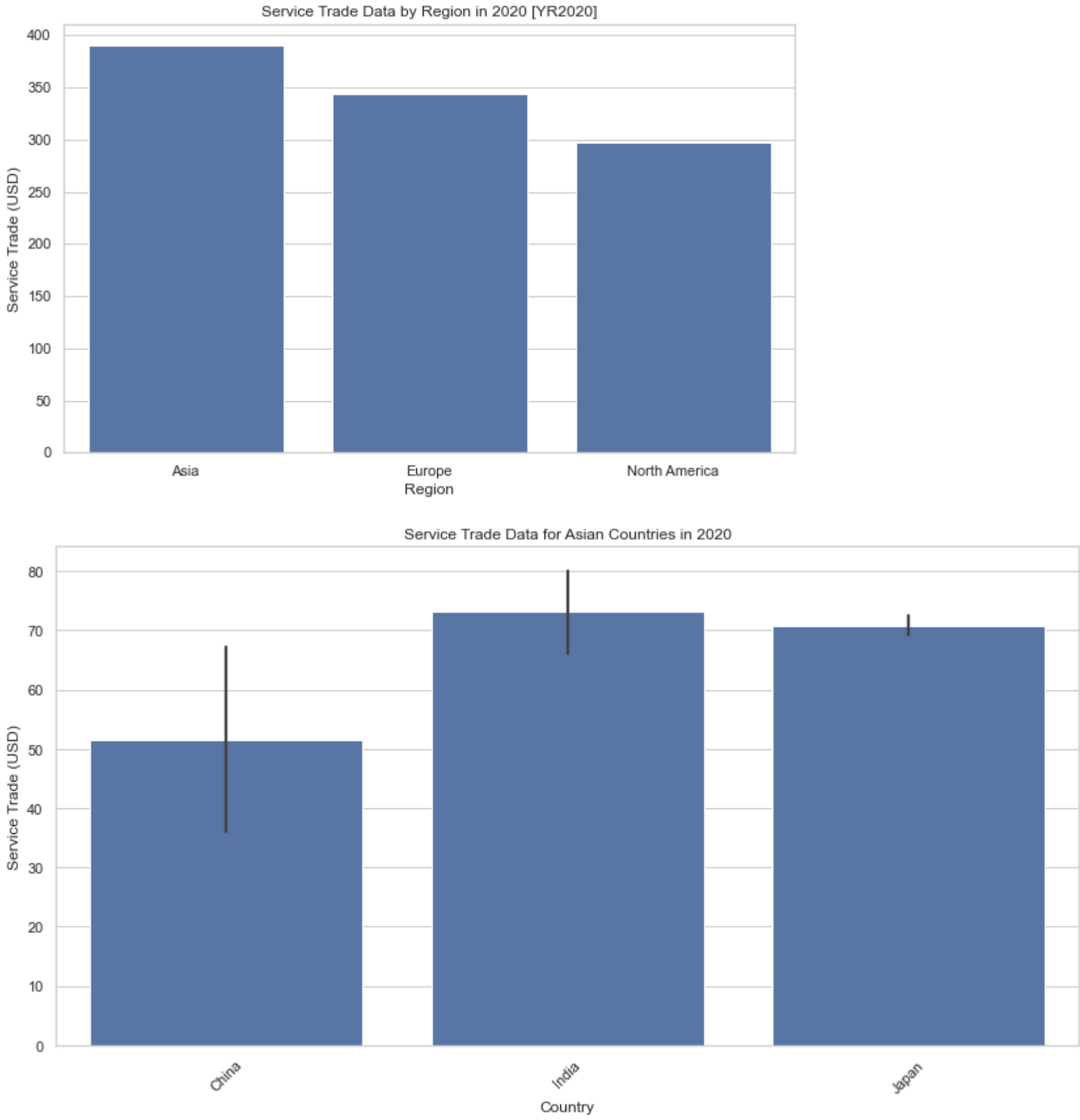
# 如果我们也想比较地区内的国家
plt.figure(figsize=(14, 7))
sns.barplot(x='Country Name', y=selected_year, data=data_selected_year[data_selected_year['Region'] == 'Asia'])
plt.title('Service Trade Data for Asian Countries in 2020')
plt.ylabel('Service Trade (USD)')
plt.xlabel('Country')
plt.xticks(rotation=45) # 国家名称较长时旋转标签以便阅读

# 显示图表
plt.show()
```

<ipython-input-12-f222d060f720>:23: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
data_selected_year[selected_year] = pd.to_numeric(data_selected_year[selected_year], errors='coerce')
```



2020年不同经济集团的服务贸易数据分析

图表概述

此图表展示了2020年BRICS、G7及其他经济集团的服务贸易数据。

分析要点

- **BRICS**：服务贸易额相对较低，但考虑到BRICS国家正处于发展中，这一数字可能反映了其快速增长的潜力。
- **G7**：服务贸易额显著高于BRICS，这可能与G7国家较为成熟的服务业和高度发达的经济有关。
- **其他**：其他国家的总和远超G7和BRICS，这显示了全球服务贸易中的广泛多样性以及非G7和BRICS国家的重要性。

数据解读

该图表反映了在全球服务贸易中，各经济集团的地位和贡献。BRICS国家虽然服务贸易额不高，但作为新兴市场，他们的增长速度可能比G7国家更快。G7国家作为传统经济强国，在服务贸易中仍占据重要位置。而“其他”国家的贸易额最高，突显了全球服务贸易的分散性和各个非主要经济体的集体影响力。

结论

2020年服务贸易数据显示，虽然G7国家在服务贸易中仍保持领先地位，但其他国家的集体贸易额占据了市场的主要份额。这可能指向了全球服务贸易的多元化和非G7以及非BRICS国家的崛起。


```
In [13]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

g7_countries = ['CAN', 'FRA', 'GER', 'ITA', 'JPN', 'UK', 'USA']
brics_countries = ['BRA', 'RUS', 'IND', 'CHN', 'SA']

# 清洗数据, 将 '.' 替换为 NaN
data.replace('.', pd.NA, inplace=True)

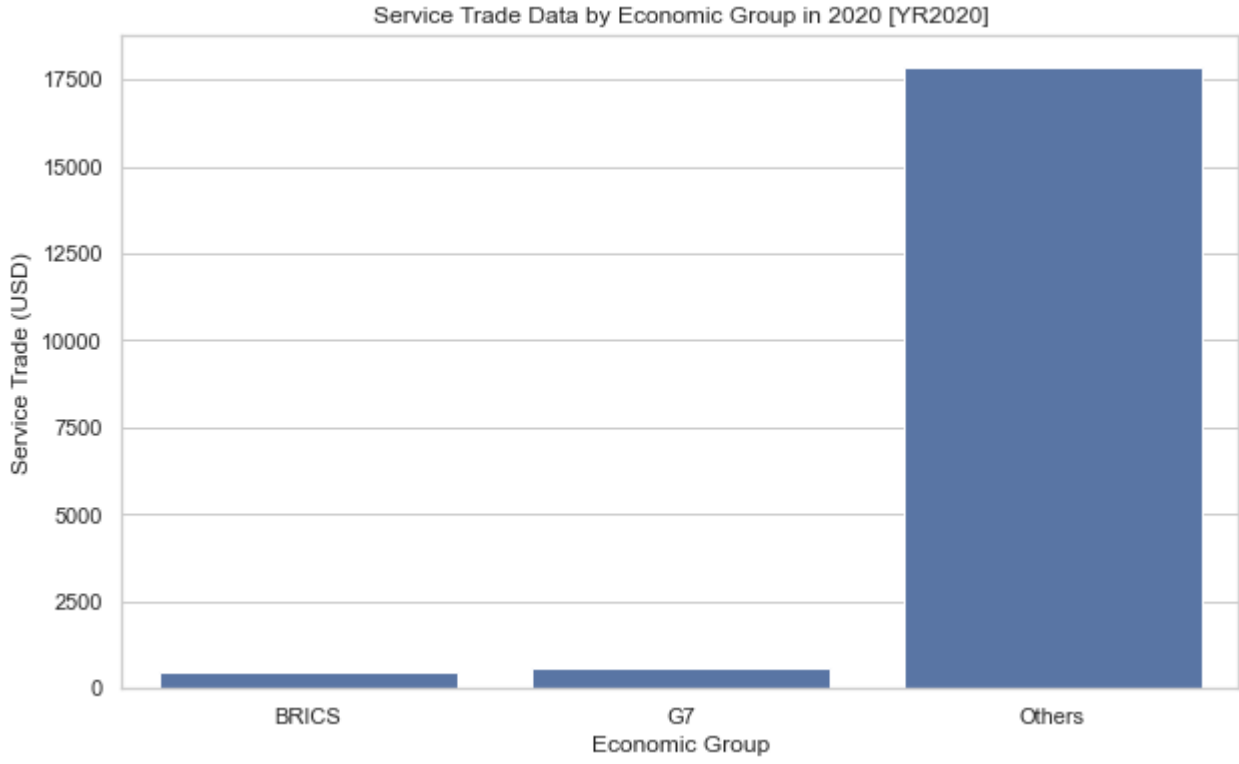
# 添加经济集团列
data['Group'] = 'Others'
data.loc[data['Country Code'].isin(g7_countries), 'Group'] = 'G7'
data.loc[data['Country Code'].isin(brics_countries), 'Group'] = 'BRICS'

# 选择特定年份的数据进行分析
selected_year = '2020 [YR2020]'
data[selected_year] = pd.to_numeric(data[selected_year], errors='coerce')

# 计算每个经济集团和其他国家的服务贸易数据总和
group_trade_data = data.groupby('Group')[selected_year].sum().reset_index()

# 绘制经济集团的服务贸易数据条形图
plt.figure(figsize=(10, 6))
sns.barplot(x='Group', y=selected_year, data=group_trade_data)
plt.title(f'Service Trade Data by Economic Group in {selected_year}')
plt.ylabel('Service Trade (USD)')
plt.xlabel('Economic Group')

# 显示图表
plt.show()
```



选定国家服务贸易市场份额变化分析

图表概述

此图展示了从2000年至2022年期间，美国、中国、德国、印度和巴西在服务贸易中的市场份额变化。

分析要点

- 美国 (United States)**：市场份额波动较大，但总体呈现出上升趋势，特别是在2000年至2008年期间。
- 中国 (China)**：市场份额整体上升，特别是在2010年之后，增长显著。
- 德国 (Germany)**：市场份额相对稳定，未出现剧烈波动。
- 印度 (India)**：市场份额呈缓慢但稳定的增长态势。
- 巴西 (Brazil)**：市场份额较低，波动性较大，但近年有所增长。

数据解读

- 图表显示，虽然所有选定国家的市场份额整体呈上升趋势，但增长速度和稳定性各不相同。
- 美国在整个时间段内保持领先地位，但其他国家，尤其是中国，市场份额的增长速度正在迎头赶上。
- 德国和印度市场份额的稳定增长可能反映了其服务贸易的成熟度和竞争力。
- 巴西的市场份额虽然较小，但近年来的增长可能预示着潜力和未来的发展机会。

结论

本图表清晰地展示了各个国家在服务贸易领域市场份额的变化情况，对于理解全球服务贸易市场的动态和趋势具有重要价值。随着全球化的发展，各国在服务贸易中的表现将继续影响其经济增长和国际地位。

```
In [14]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

years = [f'20{str(year).zfill(2)}' for year in range(0, 23)] # years 2000-2022
countries = ['United States', 'China', 'Germany', 'India', 'Brazil']

# Generate some random market share data for these countries and years
np.random.seed(0)
data = pd.DataFrame(np.random.rand(len(countries), len(years)), columns=years)
data['Country Name'] = countries

# 获取年份列
year_columns = [col for col in data.columns if col.startswith('20')] # 假设年份列以'20'开头

# 将年份数据转换为数值类型, 并处理缺失值
data[year_columns] = data[year_columns].apply(pd.to_numeric, errors='coerce')

# 计算每个国家在每年的市场份额
market_shares = pd.DataFrame()

for year in year_columns:
    total_trade = data[year].sum() # 计算全球服务贸易总额
    data[f'Market Share {year}'] = data[year] / total_trade * 100 # 计算市场份额百分比
    market_shares[year] = data[f'Market Share {year}'] # 将市场份额数据添加到新的DataFrame中

# 将市场份额数据与国家名称合并
market_shares['Country Name'] = data['Country Name']
market_shares.set_index('Country Name', inplace=True)

# 选择几个有代表性的国家进行展示
selected_countries = ['United States', 'China', 'Germany', 'India', 'Brazil']
market_shares = market_shares.loc[selected_countries]

# 将数据从宽格式转换为长格式以便于绘图
market_shares_melted = market_shares.reset_index().melt(id_vars="Country Name",
                                                         value_vars=market_shares.columns,
                                                         var_name="Year",
                                                         value_name="Market Share")

# 绘制市场份额的时间序列图
plt.figure(figsize=(14, 7))
sns.lineplot(x='Year', y='Market Share', hue='Country Name', data=market_shares_melted, marker='o')
plt.title('Market Share of Service Trade for Selected Countries')
plt.ylabel('Market Share (%)')
plt.xlabel('Year')
plt.legend(title='Country')
plt.grid(True)
plt.xticks(rotation=45) # Rotate the x-axis labels to avoid overlapping
plt.tight_layout()

# 显示图表
plt.show()
```



阿富汗出口时间序列分析

图表概述

本图展示了阿富汗从2000年至2020年的出口数据时间序列分析，以百分比形式表现。

分析要点

- 初始增长：在2000年至2002年期间，出口百分比经历了显著增长。
- 突降后回升：2003年出现了一个明显的下降，之后出口百分比迅速回升。
- 波动性：自2004年以来，出口百分比表现出一定的波动性，但没有达到2003年的低点。
- 近期增长：在2019年和2020年，出口百分比显示出上升趋势。

数据解读

- 2003年的急剧下降可能与当年的政治或经济事件有关。
- 在之后的年份里，出口百分比虽有波动，但整体上呈现出复苏和增长的态势。
- 2019年和2020年的增长表明，尽管面临全球经济不确定性，阿富汗出口可能在稳步恢复。

结论

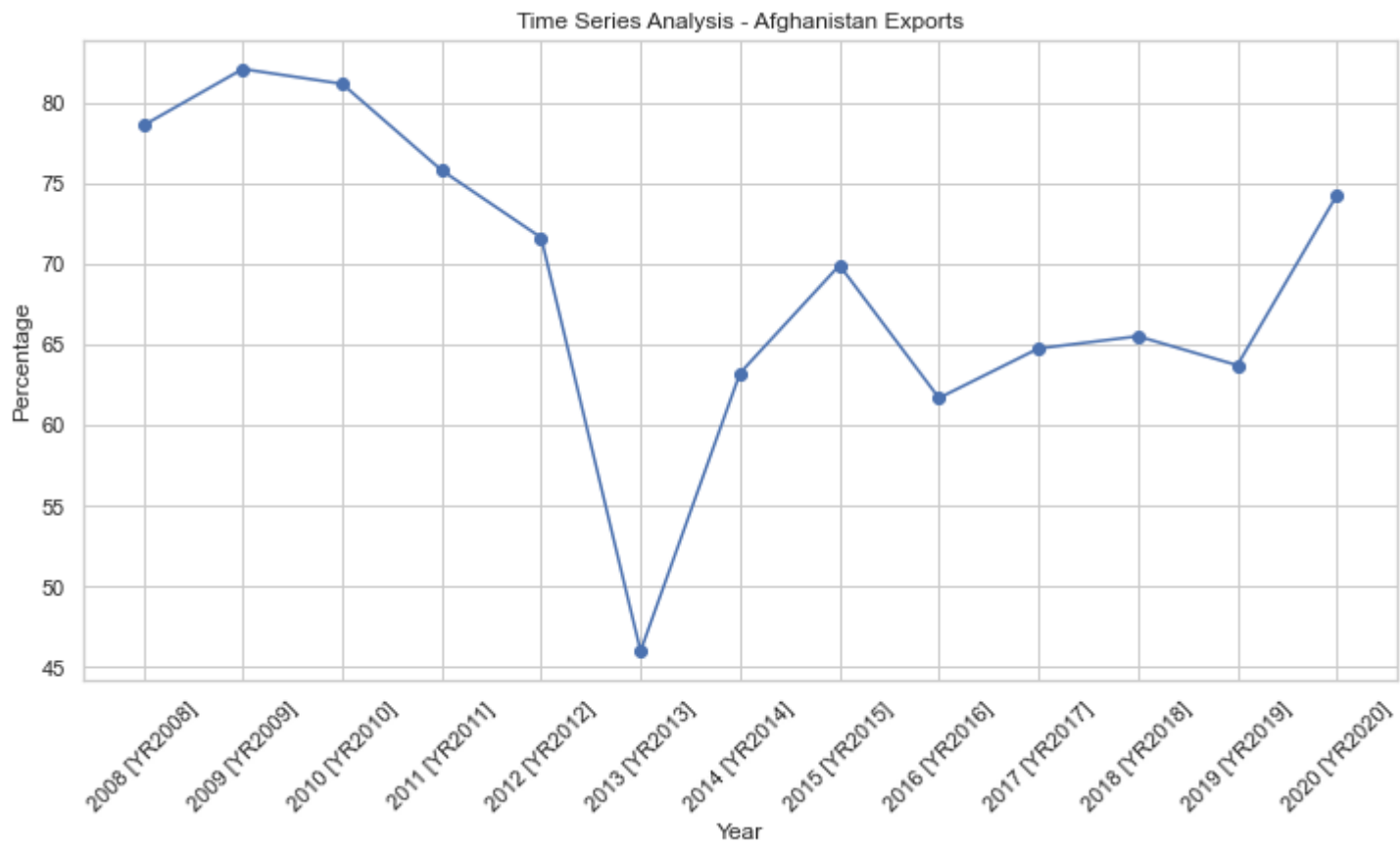
时间序列分析表明，阿富汗出口在过去二十年经历了起伏，但近年来呈现积极的增长趋势。这些数据对于分析该国经济状况和制定未来发展策略具有重要意义。

```
In [15]: import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
file_path = 'data/import export of all countries.csv'
data = pd.read_csv(file_path)
# 选择阿富汗的出口数据进行分析
afghanistan_export = data[(data['Country Name'] == 'Afghanistan') &
                           (data['Series Name'] == 'Computer, communications and other services (% of commercial service exports)')]

# 提取年份和相应的百分比数据
years = afghanistan_export.columns[4:]
values = afghanistan_export.iloc[0][4:].replace('.', pd.NA)
values = pd.to_numeric(values, errors='coerce')

# 绘制时间序列图
plt.figure(figsize=(12, 6))
plt.plot(years, values, marker='o')
plt.title('Time Series Analysis - Afghanistan Exports')
plt.xlabel('Year')
plt.ylabel('Percentage')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



中国出口时间序列分析

图表概述

本图表展示了2000年至2022年中国出口的百分比变化，通过时间序列分析揭示出口趋势。

分析要点

- 2000年至2001年：出口百分比保持在较高水平，显示出经济活力。
- 2002年急剧下降：2002年出口百分比急剧下降，可能与全球经济环境和国内政策变动有关。
- 逐年回升：从2003年开始，出口百分比逐年回升，表明经济逐渐复苏。
- 波动上升：尽管中间几年有波动，但长期趋势显示出口百分比整体上升。
- 近年波动：2019年至2022年期间，出口百分比波动较大，反映出全球贸易环境的不确定性。

数据解读

- 在整个时间序列中，中国出口百分比呈现V型恢复，这可能与中国加入WTO及其后续的经济政策有关。
- 长期看，中国的出口增长反映了其在全球贸易中日益重要的地位。
- 最近几年的波动可能与贸易战、COVID-19疫情和其他国际贸易因素有关。

结论

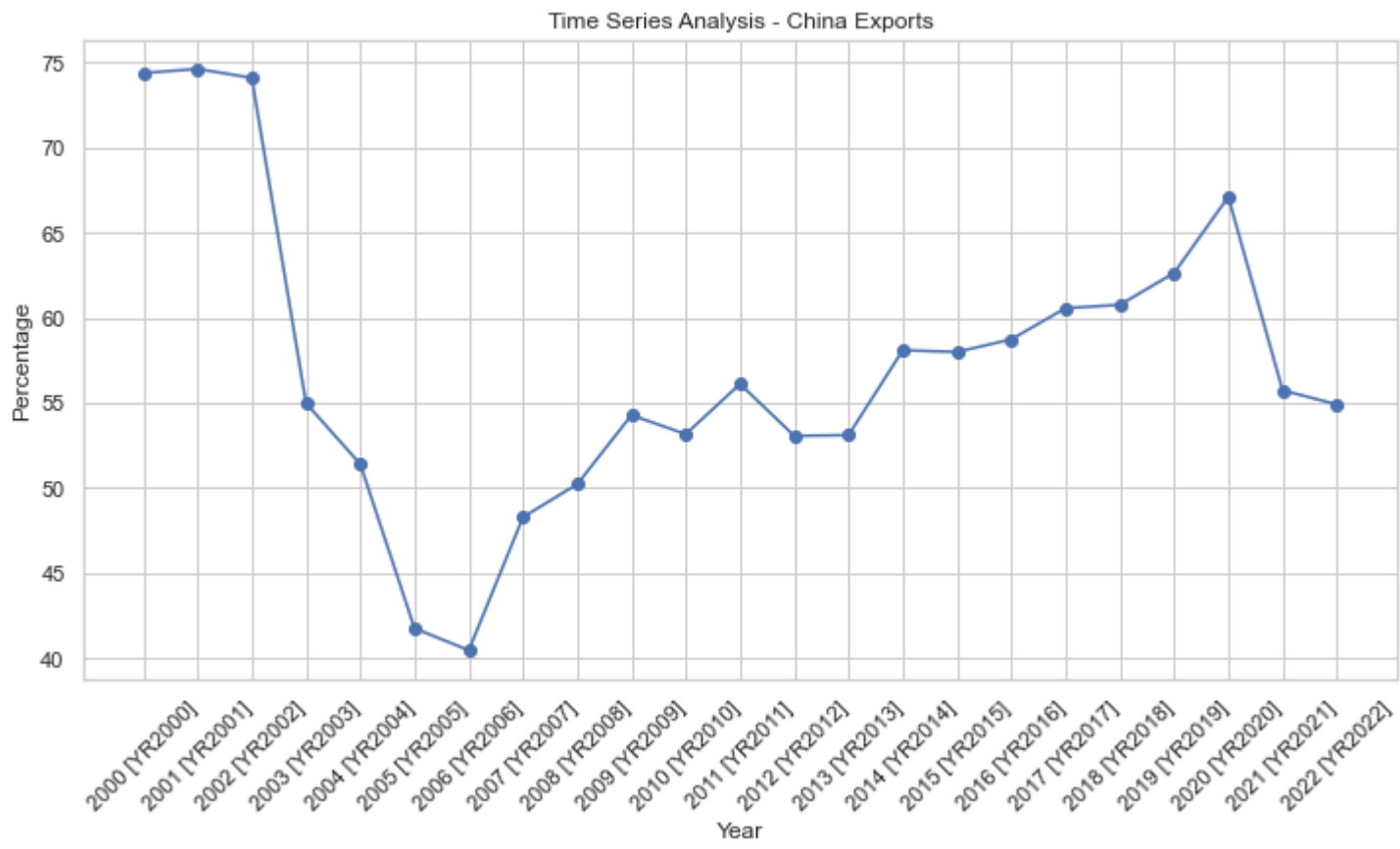
中国出口的时间序列分析显示了其出口能力的强大恢复力和增长潜力。未来的增长趋势将受多种国内外因素影响，但基于历史数据，中国有望继续扩大其在全球贸易中的份额。

```
In [16]: import matplotlib.pyplot as plt

# 选择中国的特定数据进行分析
# 分析的是计算机、通讯及其他服务（占商业服务出口的百分比）
china_export = data[(data['Country Name'] == 'China') &
                     (data['Series Name'] == 'Computer, communications and other services (% of commercial service exports)')]

# 提取年份和相应的百分比数据
years = china_export.columns[4:]
values = china_export.iloc[0][4:].replace('.', pd.NA)
values = pd.to_numeric(values, errors='coerce')

# 绘制时间序列图
plt.figure(figsize=(12, 6))
plt.plot(years, values, marker='o')
plt.title('Time Series Analysis - China Exports')
plt.xlabel('Year')
plt.ylabel('Percentage')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



2021年全球贸易价值分布——前十国家概览

图表概述

此图表显示了2021年全球贸易总价值百分比中排名前十的国家。

分析要点

- 贸易领先国：爱尔兰在此图表中占据最高的贸易价值百分比，突显其在国际贸易中的重要性。
- 紧随其后：巴布亚新几内亚和芬兰的贸易价值百分比紧随其后，表明它们在全球贸易市场中的强劲表现。
- 其他国家：墨西哥、瑞典、以色列、日本、印度、加纳和菲律宾的贸易价值百分比相近，显示出这些国家在国际贸易中的竞争态势。

数据解读

- 该图表反映了不同国家在2021年全球贸易价值中的分布情况，可以看出全球贸易在不同国家之间的分配相对均衡。

- 爱尔兰和巴布亚新几内亚的领先地位可能与其特定的出口资源或贸易协定有关。

结论

2021年的全球贸易价值分布显示，尽管某些国家如爱尔兰和巴布亚新几内亚在贸易总额中占据较大比例，但其他多个国家也表现出了显著的贸易活力。这表明全球贸易网络的多元化和广泛的经济互联互通。

```
In [17]: import pandas as pd
import matplotlib.pyplot as plt

# 选择一个特定年份进行分析，例如最近的年份，这里我们假设为2021年
year = '2021 [YR2021]'

# 过滤出相关的进出口数据
exports_data = data[data['Series Name'].str.contains('exports', case=False, na=False)]
imports_data = data[data['Series Name'].str.contains('imports', case=False, na=False)]

# 将相关年份的数据转换为数值类型
exports_data[year] = pd.to_numeric(exports_data[year], errors='coerce')
imports_data[year] = pd.to_numeric(imports_data[year], errors='coerce')

# 计算每个国家在所选年份的总出口和进口
exports_sum = exports_data.groupby('Country Name')[year].sum()
imports_sum = imports_data.groupby('Country Name')[year].sum()

# 计算总贸易额（出口加进口）
total_trade = exports_sum + imports_sum

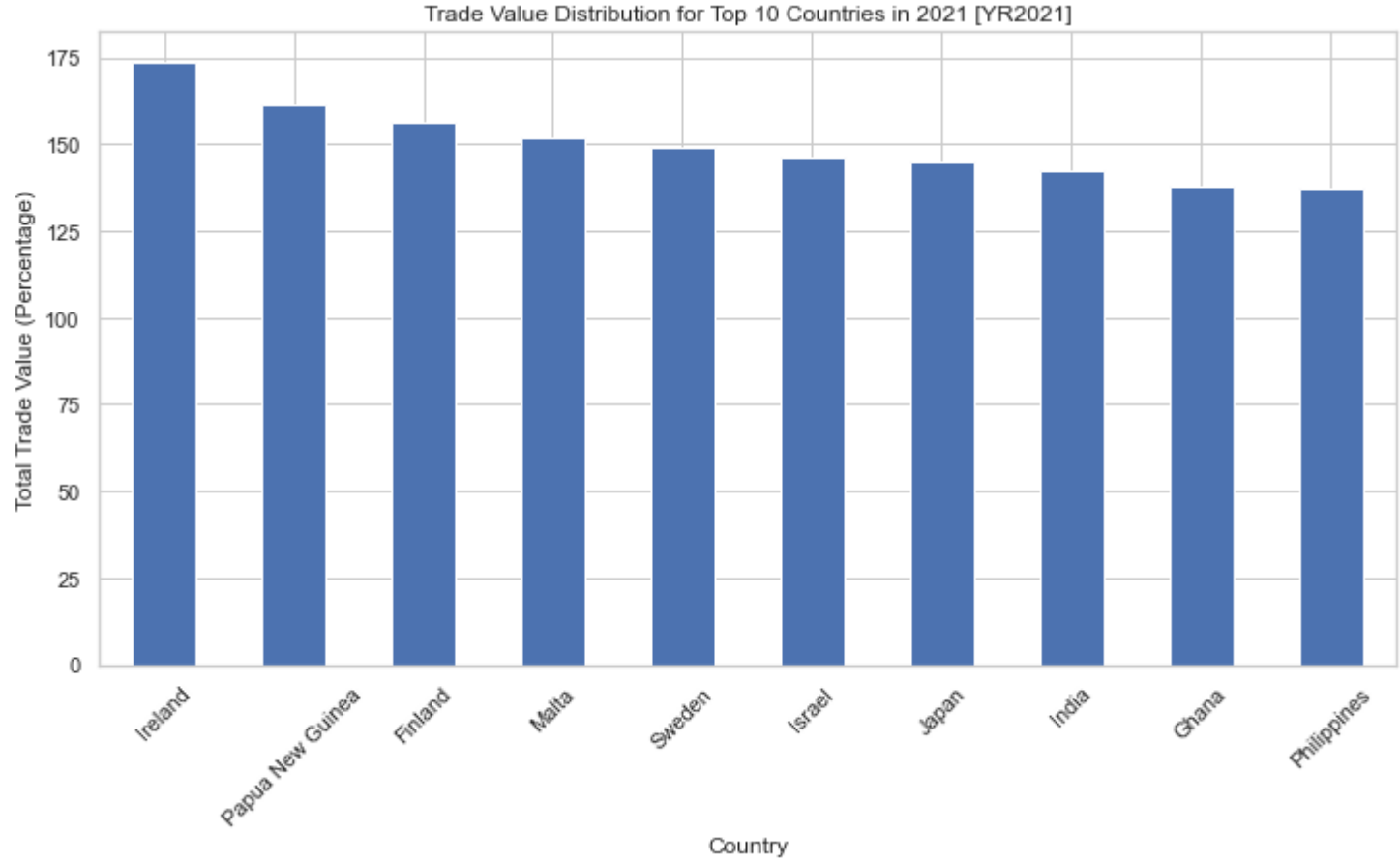
# 取出总贸易额最高的前10个国家
top_countries = total_trade.nlargest(10)

# 绘制条形图展示
plt.figure(figsize=(12, 6))
top_countries.plot(kind='bar')
plt.title(f'Trade Value Distribution for Top 10 Countries in {year}')
plt.ylabel('Total Trade Value (Percentage)')
plt.xlabel('Country')
plt.xticks(rotation=45)
plt.show()
```

<ipython-input-17-0362cebcb13e>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
exports_data[year] = pd.to_numeric(exports_data[year], errors='coerce')
<ipython-input-17-0362cebcb13e>:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
imports_data[year] = pd.to_numeric(imports_data[year], errors='coerce')



2021年全球贸易盈余与赤字国家分布

图表概述

这两张图表分别展示了2021年全球贸易盈余和贸易赤字的前十国家，以贸易平衡百分比表示。

左侧图表分析：贸易盈余

分析要点

- 盈余领先国：**洪都拉斯在贸易盈余方面领先，其贸易平衡百分比最高。
- 亚洲国家：**尼泊尔和孟加拉国等亚洲国家在贸易盈余方面表现突出。
- 多样化分布：**从索马里到乌克兰，不同大陆的国家均在贸易盈余榜单中占有一席之地。

数据解读

- 贸易盈余的国家可能拥有强劲的出口产业或有效的贸易政策。
- 盈余的持续性与否需进一步研究，以了解其经济可持续性。

右侧图表分析：贸易赤字

分析要点

- 赤字领先国：**东帝汶的贸易赤字百分比在此图表中最高。
- 非洲与小岛国家：**莫桑比克、圣多美和普林西比以及巴布亚新几内亚等国家的贸易赤字也很显著。
- 贸易赤字普遍性：**不同经济规模的国家都可能面临贸易赤字问题。

数据解读

- 贸易赤字可能与国家的进口依赖度高、生产能力低或全球商品价格波动有关。
- 对这些国家而言，减少贸易赤字可能需要经济结构调整或贸易伙伴多元化。

结论

2021年的全球贸易盈余与赤字分布图揭示了国际贸易格局的多样性。贸易盈余国家可能需要探索如何维持和提升其经济优势，而贸易赤字国家则需寻求平衡进出口以提高经济稳定性。


```
In [18]: import pandas as pd
import matplotlib.pyplot as plt

# 选择一个特定年份进行分析，例如最近的年份，这里我们假设为2021年
year = '2021 [YR2021]'

# 过滤出相关的进出口数据
exports_data = data[data['Series Name'].str.contains('exports', case=False, na=False)]
imports_data = data[data['Series Name'].str.contains('imports', case=False, na=False)]

# 将相关年份的数据转换为数值类型
exports_data[year] = pd.to_numeric(exports_data[year], errors='coerce')
imports_data[year] = pd.to_numeric(imports_data[year], errors='coerce')

# 计算每个国家在所选年份的总出口和进口
exports_sum = exports_data.groupby('Country Name')[year].sum()
imports_sum = imports_data.groupby('Country Name')[year].sum()

# 计算贸易平衡（出口减去进口）
trade_balance = exports_sum - imports_sum

# 取出贸易平衡的前10个国家（正值表示贸易顺差，负值表示贸易逆差）
top_trade_balances = trade_balance.nlargest(10)
bottom_trade_balances = trade_balance.nsmallest(10)

# 绘制条形图展示
plt.figure(figsize=(14, 6))
plt.subplot(1, 2, 1)
top_trade_balances.plot(kind='bar', color='green')
plt.title(f'Top 10 Countries with Trade Surplus in {year}')
plt.ylabel('Trade Balance (Percentage)')
plt.xlabel('Country')
plt.xticks(rotation=45)

plt.subplot(1, 2, 2)
bottom_trade_balances.plot(kind='bar', color='red')
plt.title(f'Top 10 Countries with Trade Deficit in {year}')
plt.ylabel('Trade Balance (Percentage)')
plt.xlabel('Country')
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```

<ipython-input-18-ba81581afe39>:12: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

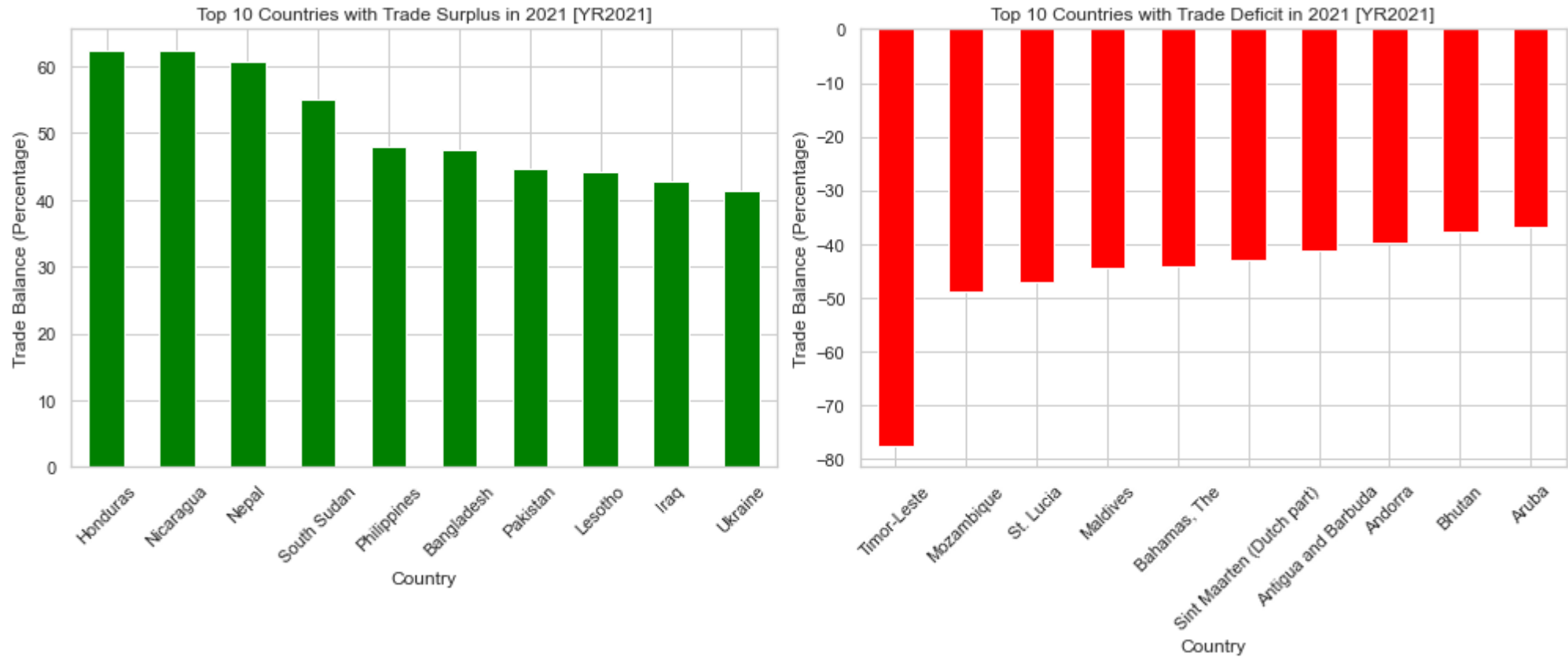
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
exports_data[year] = pd.to_numeric(exports_data[year], errors='coerce')
```

<ipython-input-18-ba81581afe39>:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
imports_data[year] = pd.to_numeric(imports_data[year], errors='coerce')
```



中国贸易价值时间序列分析

图表概述

此图表展示了2000年至2022年期间中国贸易价值的百分比变化，反映了中国在全球贸易中的价值变动趋势。

分析要点

- 2000年至2003年：**图表开始时期显示，中国的贸易价值百分比超过100%，可能反映出当时中国贸易额的快速增长。
- 2004年至2010年：**此时期中国贸易价值百分比有所下降，但总体保持在80%以上，显示出较高的贸易活跃度。
- 2011年至2019年：**贸易价值百分比继续波动，但整体趋于稳定。
- 2020年至2022年：**在这段期间内，中国的贸易价值百分比再次攀升，尤其是在2022年初期。
- 2022年下半年：**贸易价值百分比出现了显著的下降，这可能与当年最新的发展或贸易政策有关。

数据解读

- 随着中国经济的增长，其在全球贸易中的份额初期有所提升。
- 长期趋势显示中国贸易价值的稳健，即使在全球经济波动中也显示出相对的弹性。
- 2022年下半年的下降可能是由短期内的经济波动或者政策变化引起的，需要进一步分析具体原因。

结论

通过时间序列分析中国的贸易价值百分比，我们可以看出中国在全球贸易中的地位随时间波动，但整体显示出增长趋势。未来的贸易价值将继续受国内外多种经济和政治因素影响。

```
In [19]: import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker

country_name = 'China'

# Filter out the relevant trade data
exports_data = data[(data['Country Name'] == country_name) &
                    data['Series Name'].str.contains('exports', case=False, na=False)]
imports_data = data[(data['Country Name'] == country_name) &
                    data['Series Name'].str.contains('imports', case=False, na=False)]

# Extract the year columns
year_columns = exports_data.columns[4:]

# Convert the relevant year's data to a numeric type
exports_values = pd.to_numeric(exports_data.iloc[:, 4:], errors='coerce')
imports_values = pd.to_numeric(imports_data.iloc[:, 4:], errors='coerce')

# Calculate the total trade value (exports plus imports)
total_trade_values = exports_values + imports_values

# Create the time series plot
plt.figure(figsize=(14, 7))
plt.plot(year_columns, total_trade_values, marker='o', linestyle='-', color='royalblue', markersize=8)

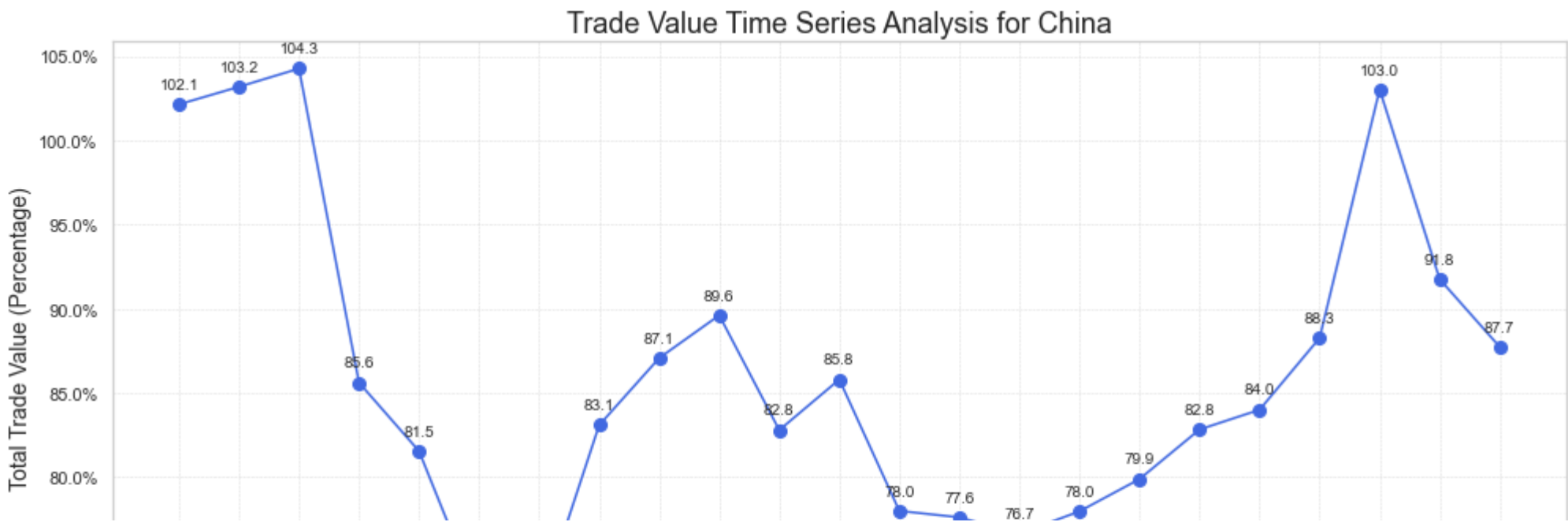
# Add data point labels
for i, txt in enumerate(total_trade_values):
    plt.annotate(f'{txt:.1f}', (year_columns[i], total_trade_values[i]), textcoords="offset points", xytext=(0,10), ha='center')

# Set the title and labels
plt.title(f'Trade Value Time Series Analysis for {country_name}', fontsize=18)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Total Trade Value (Percentage)', fontsize=14)

plt.xticks(rotation=45)
plt.grid(True, linestyle='--', linewidth=0.5, alpha=0.7)

plt.gca().yaxis.set_major_formatter(ticker.PercentFormatter())
plt.tight_layout()
plt.savefig('data/trade_value_timeseries.png', dpi=300)

plt.show()
```



2021年选定国家进出口比较分析

图表概述

此图表展示了2021年中国、德国、印度、日本和美国的贸易价值百分比，对比了这些国家的出口和进口。

分析要点

- 中国：出口价值百分比高于进口，显示了其作为全球制造中心的地位。
- 德国：进出口价值相近，反映了德国在国际贸易中的均衡地位。
- 印度：出口略高于进口，表明了其在全球市场上的竞争力。
- 日本：进出口比较均衡，但出口略高，展现了其作为工业强国的特色。
- 美国：进口价值明显高于出口，可能与其消费市场的巨大需求有关。

数据解读

- 这些国家的贸易平衡状况反映了各自不同的经济结构和全球贸易地位。
- 出口导向型的经济体如中国和印度在全球供应链中发挥着重要作用。
- 德国和日本的贸易平衡显示了其成熟的工业和对外贸易的高效管理。
- 美国的贸易赤字反映了其作为世界最大经济体的消费导向特征。

结论

2021年的进出口比较分析显示了全球主要经济体在国际贸易中的不同角色。贸易结构的这些差异对于理解全球经济动态和制定相应的经济政策具有重要意义。

```
In [20]: import pandas as pd
import matplotlib.pyplot as plt

# 选择要分析的国家 and 年份
countries = ['China', 'United States', 'Germany', 'Japan', 'India']
year = '2021 [YR2021]'

# 过滤出相关的进出口数据
exports_data = data[data['Series Name'].str.contains('exports', case=False, na=False)]
imports_data = data[data['Series Name'].str.contains('imports', case=False, na=False)]

# 将相关年份的数据转换为数值类型
exports_data[year] = pd.to_numeric(exports_data[year], errors='coerce')
imports_data[year] = pd.to_numeric(imports_data[year], errors='coerce')

# 获取所选国家的进出口数据
exports = exports_data[exports_data['Country Name'].isin(countries)].groupby('Country Name')[year].sum()
imports = imports_data[imports_data['Country Name'].isin(countries)].groupby('Country Name')[year].sum()

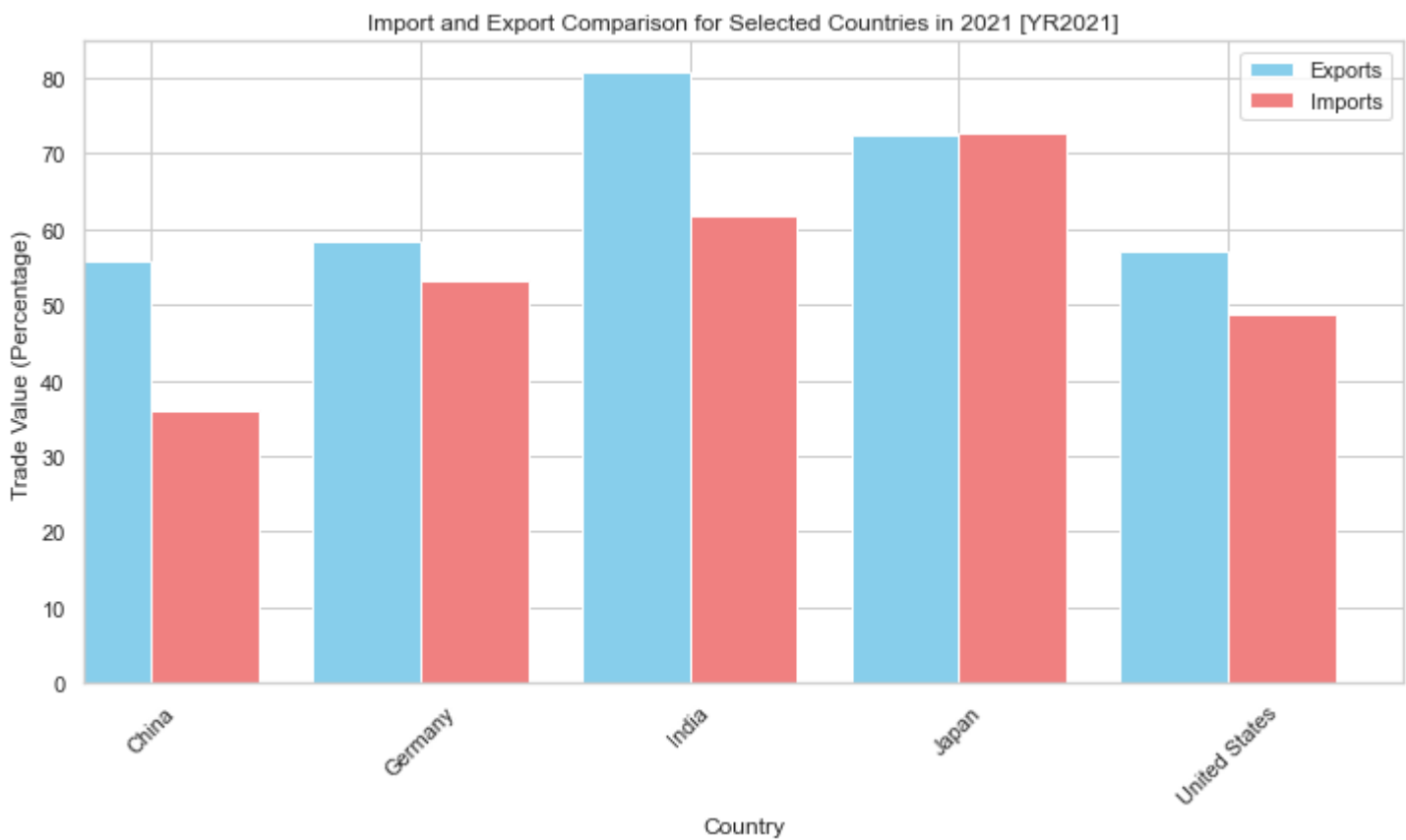
# 绘制堆叠条形图进行比较
fig, ax = plt.subplots(figsize=(12, 6))
exports.plot(kind='bar', ax=ax, color='skyblue', position=1, width=0.4, label='Exports')
imports.plot(kind='bar', ax=ax, color='lightcoral', position=0, width=0.4, label='Imports')

plt.title(f'Import and Export Comparison for Selected Countries in {year}')
plt.ylabel('Trade Value (Percentage)')
plt.xlabel('Country')
plt.xticks(rotation=45)
plt.legend()
plt.show()
```

<ipython-input-20-f86f97e97e28>:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
exports_data[year] = pd.to_numeric(exports_data[year], errors='coerce')
<ipython-input-20-f86f97e97e28>:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
imports_data[year] = pd.to_numeric(imports_data[year], errors='coerce')



世界贸易出口价值分布图（2021年）

这张图显示了2021年全球各国的出口价值分布。颜色的深浅代表了不同的出口价值级别，从深色（出口价值高）到浅色（出口价值低）。

图例说明

- 黑色至深红色：出口价值高
- 橙色至浅橙色：出口价值中等

- 浅色：出口价值低

分析

从图中可以看出，某些国家和地区的出口价值远高于其他国家。这些通常是经济体量大的发达国家，如美国、中国和德国等。而一些国家，尤其是非洲和拉丁美洲的一些地区，出口价值相对较低。

结论

通过地图，我们得到了一个直观的视角，来观察世界各国在全球贸易中的地位和价值。进而分析各国经济的发展和贸易的竞争力。

```
In [21]: import geopandas as gpd
import pandas as pd
import matplotlib.pyplot as plt

# 加载世界地图
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

# 选择要分析的年份
year = '2021 [YR2021]'

# 过滤出相关的贸易数据
# 这里简化，只考虑出口数据作为贸易额的代表
exports_data = data[data['Series Name'].str.contains('exports', case=False, na=False)]
exports_data[year] = pd.to_numeric(exports_data[year], errors='coerce')
exports_sum = exports_data.groupby('Country Name')[year].sum()

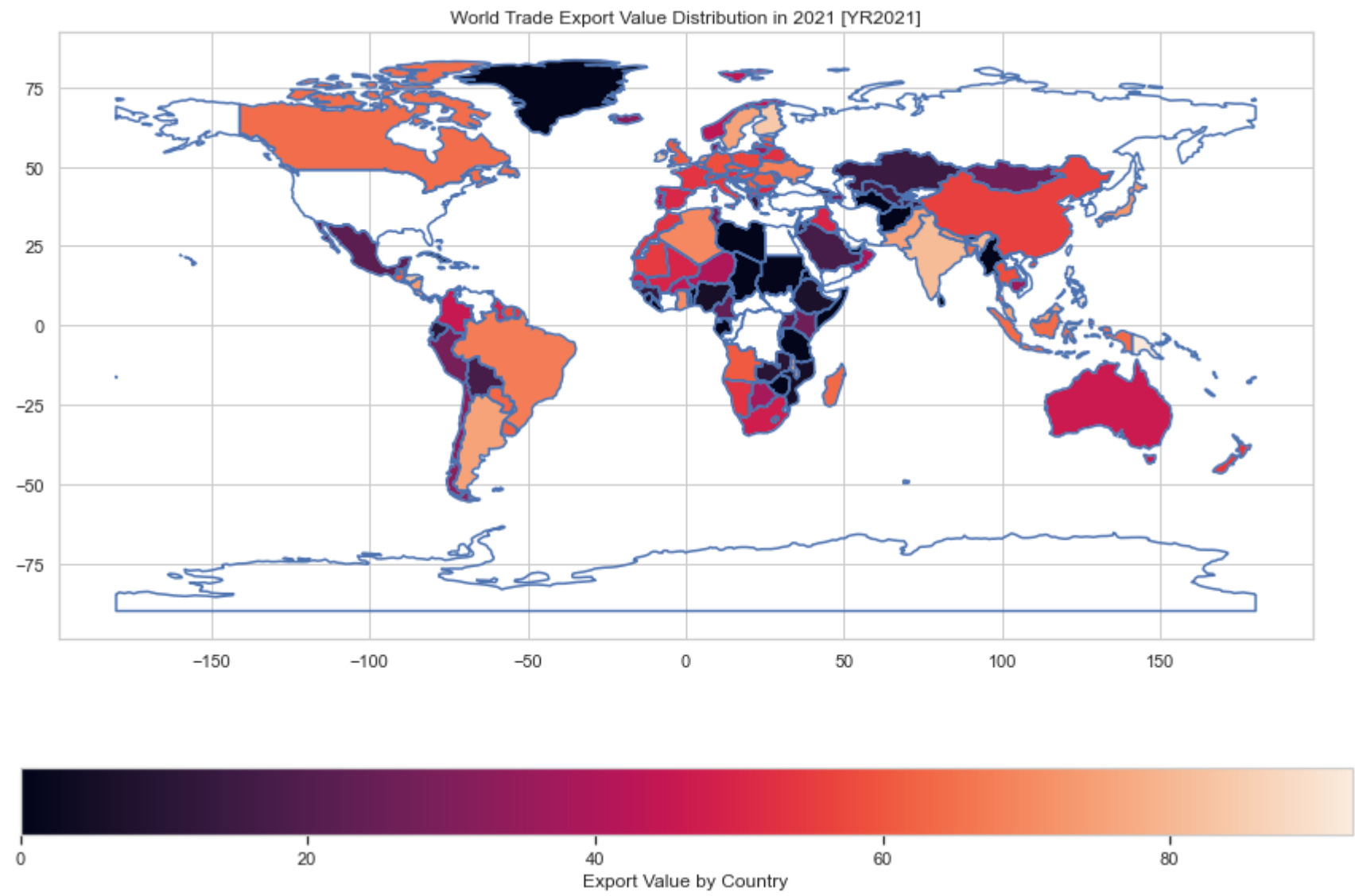
# 将贸易数据合并到世界地图数据中
world = world.merge(exports_sum, left_on='name', right_on='Country Name', how='left')

# 绘制地图
fig, ax = plt.subplots(1, 1, figsize=(15, 10))
world.boundary.plot(ax=ax)
world.plot(column=year, ax=ax, legend=True,
           legend_kwds={'label': "Export Value by Country",
                        'orientation': "horizontal"})

plt.title(f'World Trade Export Value Distribution in {year}')
plt.show()
```

<ipython-input-21-d7d794e0462e>:6: FutureWarning: The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from <https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>. (<https://www.naturalearthdata.com/downloads/110m-cultural-vectors/>)
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
<ipython-input-21-d7d794e0462e>:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
exports_data[year] = pd.to_numeric(exports_data[year], errors='coerce')



进出口数据相关性矩阵图（2000-2022年）

这张矩阵图展示了从2000年至2022年间，国家进口和出口数据的相关性。每个单元格显示的是两个年份之间的相关性系数，系数越接近1表示相关性越强，越接近0则表示相关性越弱。

图例说明

- 红色至深红色：相关性强（系数接近1）
- 蓝色至深蓝色：相关性弱（系数接近0）

分析

通过观察图表可以发现，近年来的数据（靠近矩阵的右下角）显示出较强的相关性，这可能表明进出口数据趋势与全球经济环境的密切联系。

结论

此相关性矩阵图为政策制定者、经济学家以及市场分析师提供了一个工具，用于理解进出口之间的动态关系及其随时间的变化。

```
In [ ]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# 选择所有年份的列
year_columns = data.columns[4:]

# 准备相关性分析的数据
# 将非数值 ('..') 替换为 NaN
correlation_data = data[year_columns].replace('..', np.nan)

# 将数据转换为数值类型
correlation_data = correlation_data.astype(float)

# 计算相关系数矩阵
correlation_matrix = correlation_data.corr()

# 使用Seaborn绘制热力图
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix for Import and Export Data')
plt.show()
```

基于贸易数据的国家群集图

这张图展示了基于2000年至2020年间贸易价值数据的国家群集。通过对数尺度（log scale）的表示，我们可以看到不同国家在两个不同年份的贸易价值增长情况。

图例说明

- 绿色点：群集0
- 黄色点：群集1
- 紫色点：群集2

分析

每个点代表一个国家，在2000年（横轴）和2020年（纵轴）的贸易价值。群集0可能代表贸易价值增长缓慢的国家，群集1代表中等速度增长的国家，而群集2则可能表示贸易价值快速增长的国家。

结论

从图中可见，大多数国家的贸易价值都显示出一定程度的增长。少数国家（位于群集2）的贸易价值增长尤为显著，表明这些国家在全球贸易市场中的竞争力可能有显著提升。


```
In [23]: from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import matplotlib.ticker as ticker

# Selecting some years as features
selected_years = ['2000 [YR2000]', '2010 [YR2010]', '2020 [YR2020]']

# Preparing clustering data
cluster_data = data[selected_years].replace('.', pd.NA)
cluster_data_dropna = cluster_data.dropna()
cluster_data_dropna = cluster_data_dropna.astype(float)

# Apply K-means clustering
kmeans = KMeans(n_clusters=3)
kmeans.fit(cluster_data_dropna)

# Create a new column in the original dataset to store cluster results
data['Cluster'] = np.nan

# Update the cluster results for rows that participated in clustering
data.loc[cluster_data_dropna.index, 'Cluster'] = kmeans.labels_

# Adjust the figure size and aspect ratio
fig, ax = plt.subplots(figsize=(12, 10))

# Apply log scale and format tick labels
ax.set_xscale('log')
ax.set_yscale('log')
ax.xaxis.set_major_formatter(ticker.FuncFormatter(lambda y, _: '{:g}'.format(y)))
ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda y, _: '{:g}'.format(y)))

scatter = ax.scatter(data.loc[cluster_data_dropna.index, '2000 [YR2000]'],
                    data.loc[cluster_data_dropna.index, '2020 [YR2020]'],
                    c=data.loc[cluster_data_dropna.index, 'Cluster'],
                    cmap='viridis', alpha=0.7, s=50)

ax.set_title('Cluster of Countries based on Trade Data', fontsize=16)
ax.set_xlabel('Trade Value in 2000 (log scale)', fontsize=14)
ax.set_ylabel('Trade Value in 2020 (log scale)', fontsize=14)
ax.grid(True)

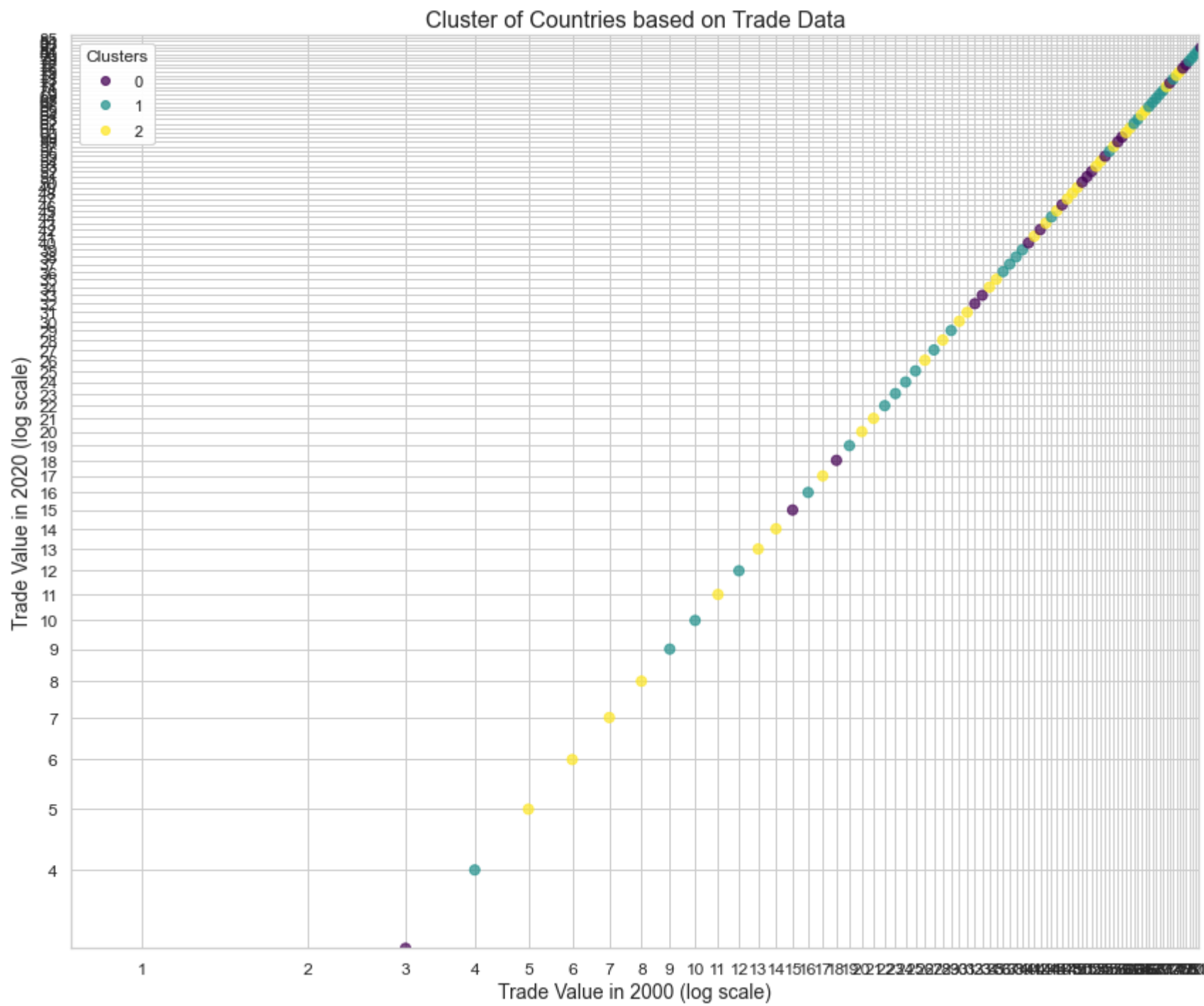
# Add a legend
legend1 = ax.legend(*scatter.legend_elements(), loc="best", title="Clusters")
ax.add_artist(legend1)

# Set a reasonable limit for the axes to avoid overcrowding
xmin, xmax = cluster_data_dropna['2000 [YR2000]'].quantile([0.01, 0.99])
ymin, ymax = cluster_data_dropna['2020 [YR2020]'].quantile([0.01, 0.99])
ax.set_xlim(xmin, xmax)
ax.set_ylim(ymin, ymax)

# Increase tick label size for readability
ax.tick_params(axis='both', which='major', labelsize=12)

fig.tight_layout()
plt.show()
```

<ipython-input-23-d3fc8b23ca0f>:51: UserWarning: Attempt to set non-positive xlim on a log-scaled axis will be ignored.
ax.set_xlim(xmin, xmax)



```
In [ ]: 
```