In [1]:
```
import pandas as pd
import numpy as np
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, ipl
init_notebook_mode(connected=True)
import holoviews as hv
import plotly.graph_objects as go
import plotly.express as pex
hv.extension('bokeh')
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
import networkx as nx
```

Q1A: Trade between top five exporters and importers (Sankey)

In [2]:
```
def detail_trade(country_name, year):
    # Read all information in the sheet
    country = pd.read_excel('Q1.xlsx', sheet_name = country_name)
    # Read specific columns in the sheet
    country_export = pd.read_excel('Q1.xlsx', sheet_name = country_name, us
    # Choose specific year (There are two years in the sheet, including 201
    country_export = country_export.loc[country_export['Year'] == year]
    # Sort from highest to lowest
    country_export = country_export.sort_values(by=['Export (US$ Thousand)'
    # Get the first ten countries with highesting numbers
    country_export = country_export[:10]
    # Rename column names
    country_export_new = country_export.rename(columns={'Export (US$ Thousa
    # Reset the index
    country_export_new = country_export_new.reset_index()
    # Add new column, named "Measure"
    country_export_new['Measure'] = ['Export' for i in range(len(country_ex

    # Do the same thing for import
    country_import = pd.read_excel('Q1.xlsx', sheet_name = country_name, us
    country_import = country_import.loc[country_import['Year'] == year]
    country_import = country_import.sort_values(by=['Import (US$ Thousand)'
    country_import = country_import[:10]
    country_import_new = country_import.rename(columns={'Import (US$ Thousa
    country_import_new = country_import_new.reset_index()
    country_import_new['Measure'] = ['Import' for i in range(len(country_im

    # Combine two table including the exporting and importing for specific
    total = country_export_new.append(country_import_new)
    # Reset index
    total = total.reset_index()
    # Drop columns
    total = total.drop(columns=['level_0', 'index', 'Reporter Name', 'Year'
    # Reorder columns
    total = total[['Measure', 'Country', 'Value']]

    return total
```

In [3]:
```
total_usa = detail_trade('United States', 2019)
```

In [4]:
```
# Draw the graph
hv.Sankey(total_usa)
sankey1 = hv.Sankey(total_usa, kdims=["Measure", "Country"], vdims=["Value"

sankey1.opts(cmap='Colorblind',label_position='left',
                                edge_color='Country', edge_line_width=0,
                                node_alpha=1.0, node_width=40, node_sort=T
                                width=800, height=600, bgcolor="snow",
                                title="Trade in United States in 2019")
```

Out[4]:

In [5]:
```
total_china = detail_trade('China', 2019)
```

In [6]:
```
# Draw the graph
hv.Sankey(total_china)
sankey1 = hv.Sankey(total_china, kdims=["Measure", "Country"], vdims=["Valu

sankey1.opts(cmap='Colorblind',label_position='left',
                                    edge_color='Country', edge_line_width=0,
                                    node_alpha=1.0, node_width=40, node_sort=T
                                    width=800, height=600, bgcolor="snow",
                                    title="Trade in China in 2019")
```

Out[6]:

In [7]:
```
exporters = ['China', 'United States', 'Germany', 'Japan', 'Korea, Rep.']
exporters_name = ['China_ex', 'United States_ex', 'Germany_ex', 'Japan_ex',
importers = ['United States', 'China', 'Germany', 'United Kingdom', 'France
importers_name = ['United States_im', 'China_im', 'Germany_im', 'United Kin
```

In [8]:
```
exporter = pd.read_excel('Q1_new.xlsx', sheet_name = 'Exporter_2019')
exporter = exporter.rename(columns={'Reporter Name': 'Exporter', 'Partner N
importer = pd.read_excel('Q1_new.xlsx', sheet_name = 'Importer_2019')
importer = importer.rename(columns={'Reporter Name': 'Importer', 'Partner N
importer = importer[['Exporter', 'Importer', 'Value']]
```

In [9]:
```
total_trade = exporter.append(importer)
total_trade = total_trade.dropna(subset=['Value'])
```

In [10]:
```
total_trade['Exporter'] = total_trade['Exporter'].replace(exporters, export
total_trade['Importer'] = total_trade['Importer'].replace(importers, import
```

In [11]:
```
for index, row in total_trade.iterrows():
    if row['Exporter'] not in exporters_name:
        total_trade['Exporter'] = total_trade['Exporter'].replace([row['Exp

for index, row in total_trade.iterrows():
    if row['Importer'] not in importers_name:
        total_trade['Importer'] = total_trade['Importer'].replace([row['Imp
```

In [12]:
```
sankey1 = hv.Sankey(total_trade, kdims=["Exporter", "Importer"], vdims=["Va

sankey1.opts(cmap='Colorblind',label_position='left',
                                    edge_color='Importer', edge_line_width=0,
                                    node_alpha=1.0, node_width=40, node_sort=T
                                    width=1000, height=600, bgcolor="snow",
                                    title="Trade in 2019 for Top Five Exportin
```

Out[12]:

Q1B: Trade between top 10 gdp countries (Weighted graph)

This is the other way to represent the data of exports and imports. However, we didn't show in the presentaion and report result, since in the future we would like to improve the presentation of these two weighted graphs by adding the arrows and having clear explanation of the amount of either importing or exporting for these seleced countres.

In [13]:
```
data = pd.read_excel('top10.xlsx')
datanew = data[["Country","ExportCountry","Export"]]
datanew
```

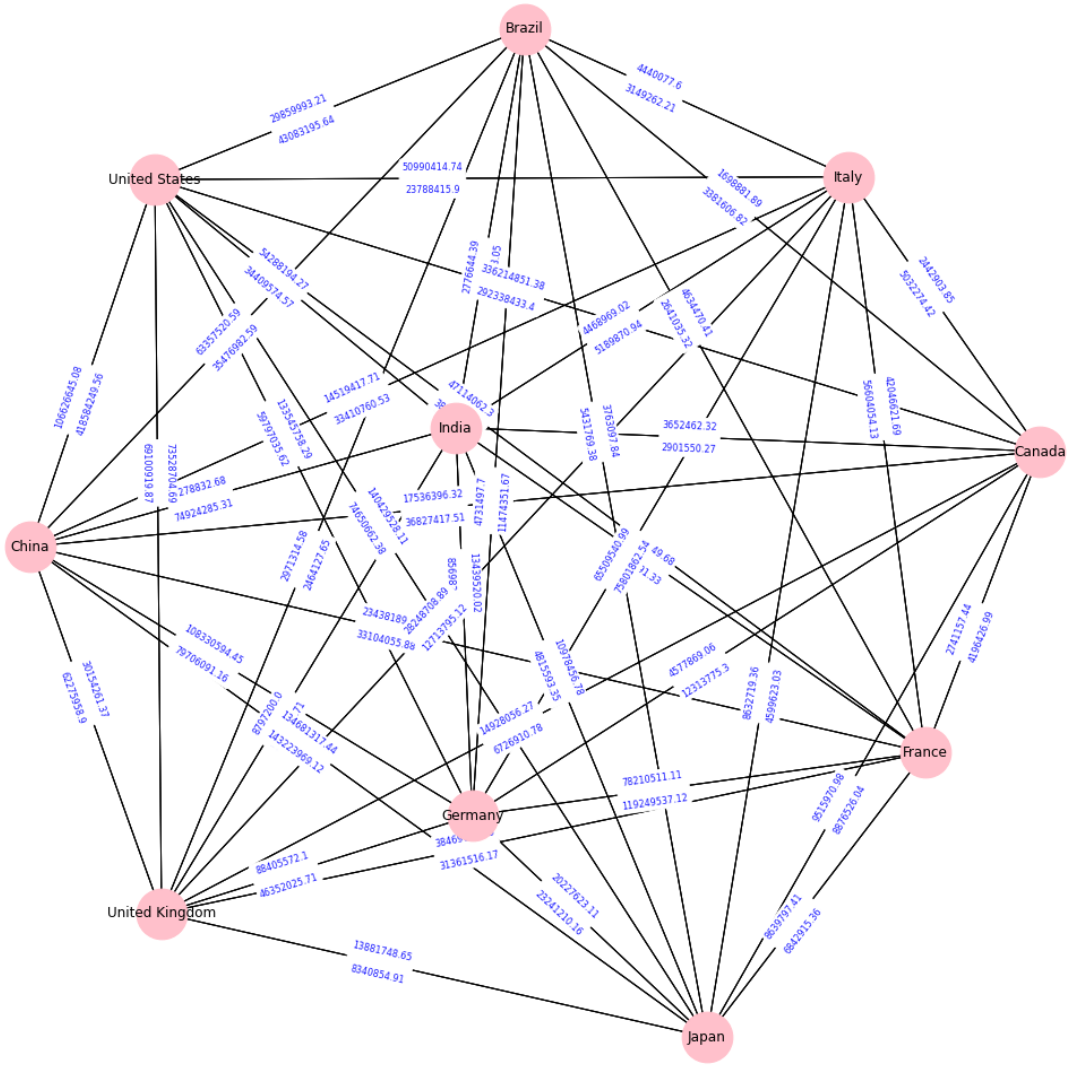
Out[13]:

	Country	ExportCountry	Export
0	Brazil	Canada	3381606.82
1	Brazil	China	63357520.59
2	Brazil	France	2641035.32
3	Brazil	Germany	4731497.70
4	Brazil	India	2776644.39
...	...	...	...
85	United States	Germany	59797035.62
86	United States	India	34409574.57
87	United States	Italy	23788415.90
88	United States	Japan	74650662.38
89	United States	United Kingdom	69100919.87

90 rows × 3 columns

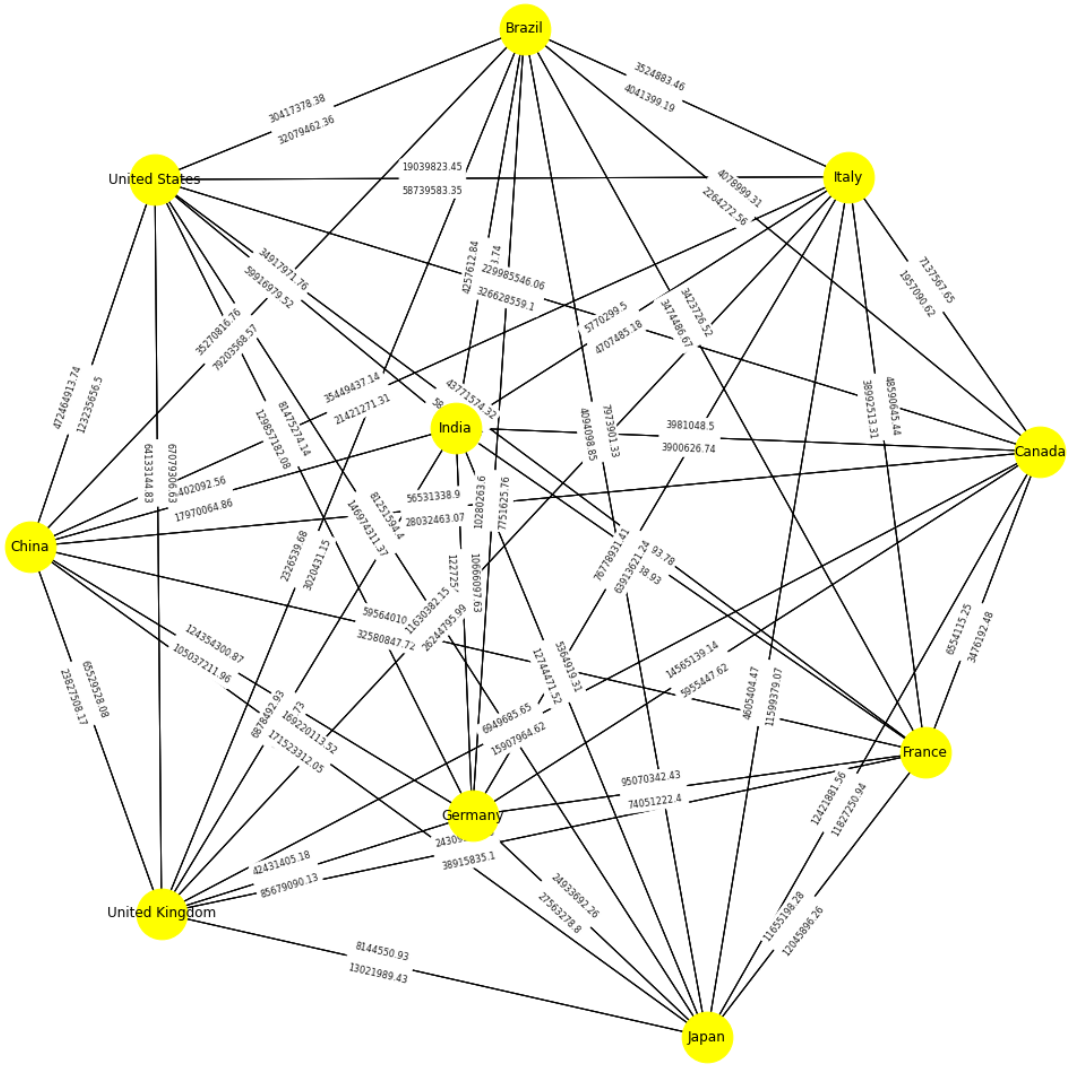
In [14]:
```
G = nx.DiGraph()
for index, row in datanew.iterrows():
    G.add_node(row['Country'])
    G.add_node(row['ExportCountry'])
    G.add_edge(row['Country'], row['ExportCountry'], length = row['Export']
```

In [15]:
```
plt.figure(figsize=(15,15))
pos = nx.spring_layout(G,seed=20)
nx.draw(G, pos, with_labels=True, connectionstyle='arc3,rad=0')

edge_labels = dict([((u, v,), f'{d["length"]}\n\n{G.edges[(v,u)]["length"]}
                for u, v, d in G.edges(data=True) if pos[u][0] > pos[v][0]]



nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_color='b
nx.draw_networkx_nodes(G, pos, node_size=2000, node_color = 'pink')
nx.draw_networkx_edges(G,pos,alpha=0.5,edge_color='black')




#plt.tight_layout()
plt.title("Top 10 GDP Countries' Export Trade ")
plt.show()
```



In [16]:
```
data = pd.read_excel('top10.xlsx')
Import = data[["Country","ExportCountry","Import"]]

G = nx.DiGraph()
for index, row in Import.iterrows():
    G.add_node(row['Country'])
    G.add_node(row['ExportCountry'])
    G.add_edge(row['Country'], row['ExportCountry'], length = row['Import']
```

In [17]:
```
plt.figure(figsize=(15,15))
pos = nx.spring_layout(G,seed=20)
nx.draw(G, pos, with_labels=True, connectionstyle='arc3,rad=0')

edge_labels = dict([((u, v,), f'{d["length"]}\n\n{G.edges[(v,u)]["length"]}
                for u, v, d in G.edges(data=True) if pos[u][0] > pos[v][0]]



nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, font_color='b
nx.draw_networkx_nodes(G, pos, node_size=2000, node_color = 'yellow')
nx.draw_networkx_edges(G,pos,alpha=0.5,edge_color='black')




#plt.tight_layout()
plt.title("Top 10 GDP Countries' Import Trade ")
plt.show()
```



Q2: Analyze the value of imports & exports, 16 categories, 10 countries

In [18]:
```
df2017 = pd.read_excel("Q2_2017.xlsx")
df2018 = pd.read_excel("Q2_2018.xlsx")
df2019 = pd.read_excel("Q2_2019.xlsx")
```

In [19]:
```
df2017.head()
```

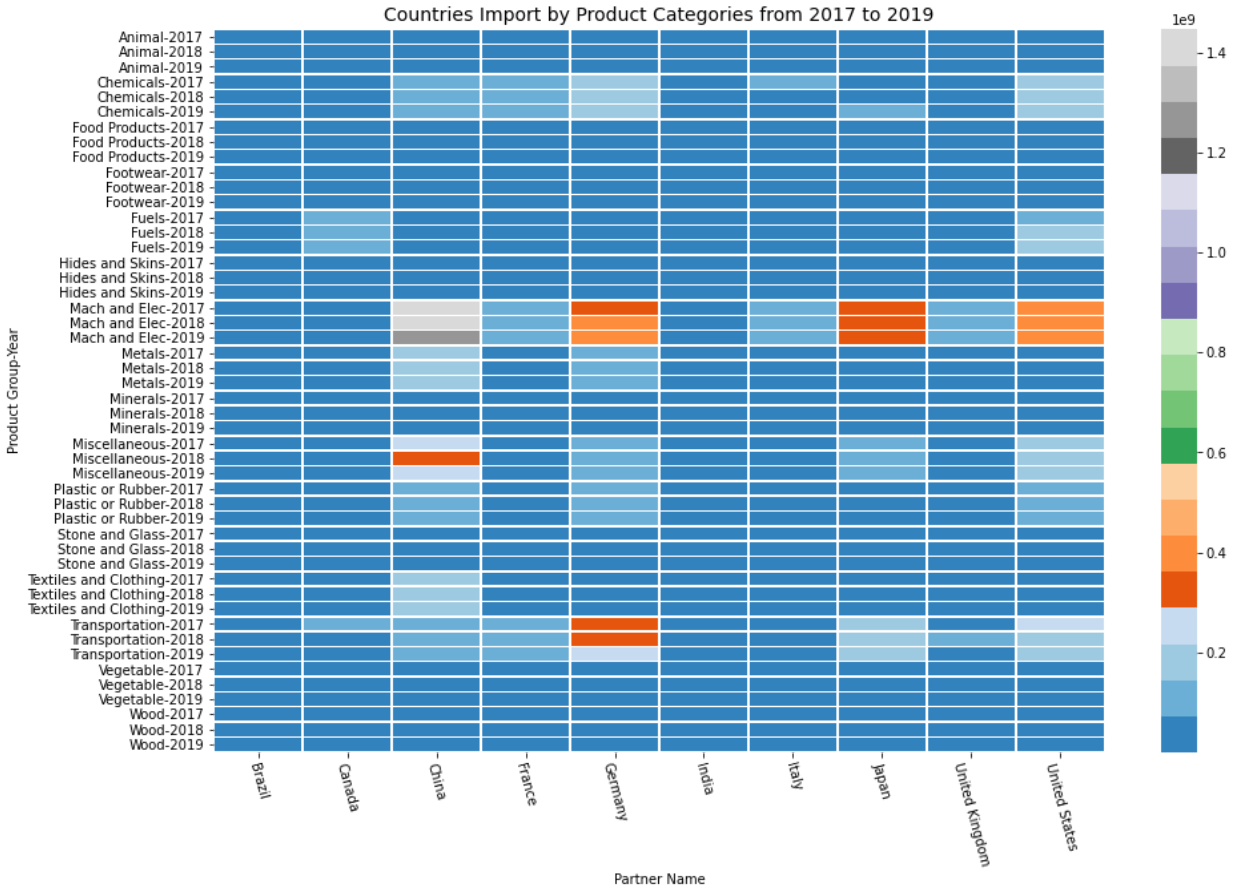
Out[19]:

	Reporter Name	Partner Name	Year	Product Group	Export (US$ Thousand)	Import (US$ Thousand)
0	World	United States	2017	Animal	3.271923e+07	2.776733e+07
1	World	United States	2017	Chemicals	2.688639e+08	1.903602e+08
2	World	United States	2017	Food Products	8.013343e+07	4.503055e+07
3	World	United States	2017	Footwear	3.087877e+07	1.185737e+06
4	World	United States	2017	Fuels	1.558189e+08	1.307280e+08

Heatmap: Changes and Trends

In [20]:
```
# Countries import from 2017-2019 heatmap
df_2017_2018_2019 = pd.concat([df2017,df2018,df2019])

df_country_prod_imp = df_2017_2018_2019.pivot_table(index=["Product Group",

plt.figure(figsize=(15,10))
sns.heatmap(df_country_prod_imp, cmap="tab20c",linewidths=.5)
plt.xticks(rotation=-75)
plt.title("Countries Import by Product Categories from 2017 to 2019",fontsi
```

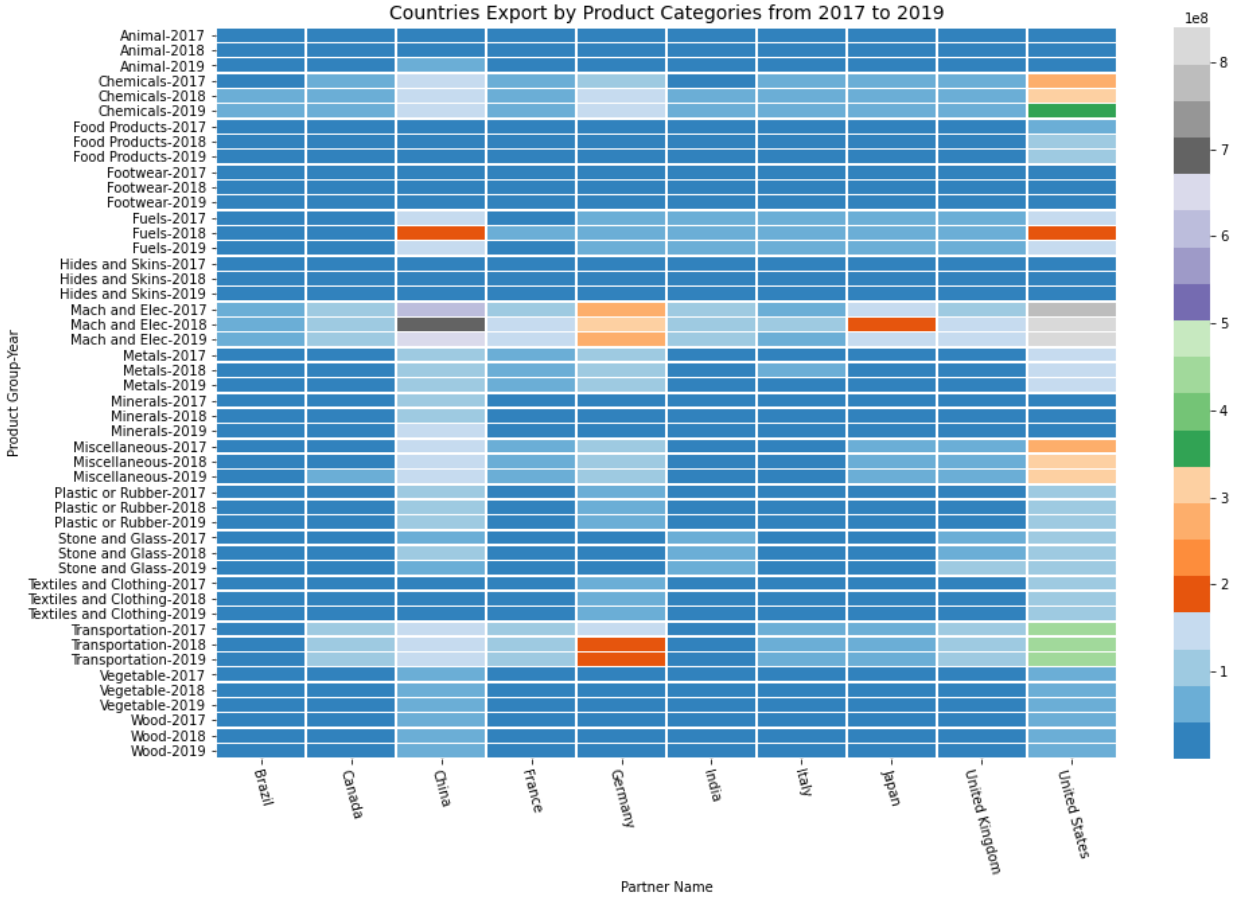
Out[20]:
```
Text(0.5, 1.0, 'Countries Import by Product Categories from 2017 to 201
9')
```




```
- Import
- Trend & influence of tariff
- Categories of Mach & Elec and transportation should be a metter of
concern
- Mach & Elec: China: downward trend, highest
               Germany: upward,highest
- Transportation: Germany: down
-------------------------------------------------------------------------
Why heatmap: remarkable in color: which categories and countries
compared to line chart and bar chart
line chart: too many lines, hard to tell
bar chart: differencies in
```

In [21]:
```
# Countries export from 2017-2019 heatmap
df_2017_2018_2019 = pd.concat([df2017,df2018,df2019])

df_country_prod_exp = df_2017_2018_2019.pivot_table(index=["Product Group",

plt.figure(figsize=(15,10))
sns.heatmap(df_country_prod_exp, cmap="tab20c",linewidths=.5)
plt.xticks(rotation=-75)
plt.title("Countries Export by Product Categories from 2017 to 2019",fontsi
```

Out[21]:
```
Text(0.5, 1.0, 'Countries Export by Product Categories from 2017 to 201
9')
```




```
- Export
- Categories:
    Chemical: USA: increase, highesr
    Mach & Elec: China: up then down, higher then most of the countries
                 USA: up,highest
    Miscellaneous: USA:up, highest
    Transportation: Germany: up
                    Usa: highest
```

Comparison of export among countries in 2018

and 2019

In [22]:
```
df2018_exp = df2018[["Partner Name","Year","Product Group","Export (US$ Tho
df2019_exp = df2019[["Partner Name","Year","Product Group","Export (US$ Tho
df2019_exp["Export (US$ Thousand)"] = df2019_exp["Export (US$ Thousand)"]*(

df2018_2019_exp = pd.concat([df2018_exp,df2019_exp ])
df2018_2019_exp.head()
```

Out[22]:

	Partner Name	Year	Product Group	Export (US$ Thousand)
0	United States	2018	Animal	3.319013e+07
1	United States	2018	Chemicals	3.190755e+08
2	United States	2018	Food Products	8.796040e+07
3	United States	2018	Footwear	3.298597e+07
4	United States	2018	Fuels	1.848361e+08

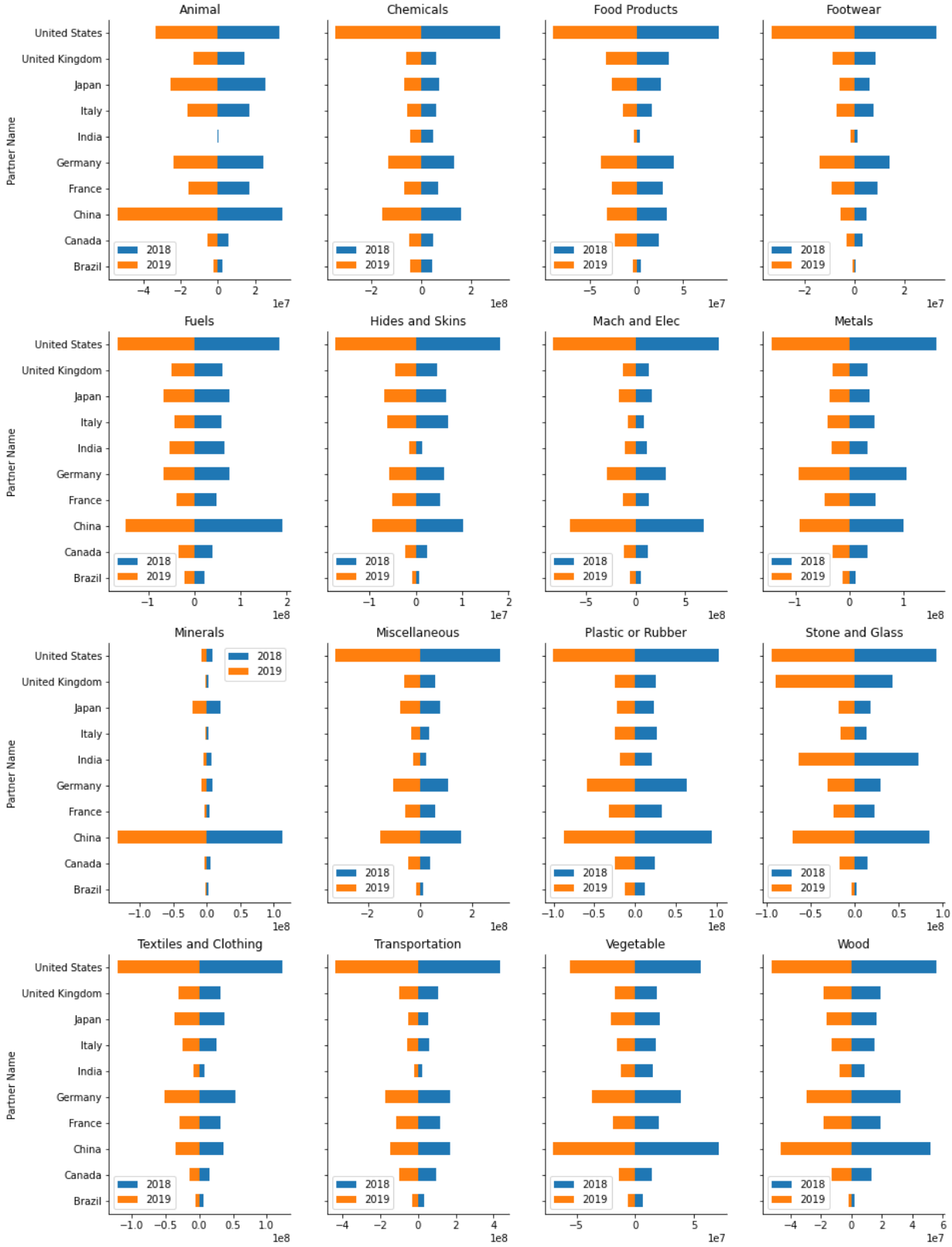
In [23]:
```
df2018_2019_exp_piv = df2018_2019_exp.pivot_table(index="Partner Name",colu
df2018_2019_exp_piv
```

Out[23]:

Product Group	Animal		Chemicals		Food Products		
Year	2018	2019	2018	2019	2018	2019	
Partner Name							
Brazil	2506623.24	-2400707.78	4.245714e+07	-4.275616e+07	4591824.79	-4199421.86	64
Canada	5503902.79	-5613622.18	4.862393e+07	-4.865184e+07	24099363.35	-23787571.66	309
China	34788272.26	-54192877.17	1.610071e+08	-1.587524e+08	32184724.86	-31656304.90	480
France	16735662.79	-15797783.42	6.772859e+07	-6.702447e+07	27894631.80	-27139176.83	908
Germany	24567795.22	-23811105.69	1.319325e+08	-1.327650e+08	39845411.13	-39143404.74	1421
India	204622.45	-207026.28	4.687901e+07	-4.548054e+07	3125548.59	-2918894.34	121
Italy	16976949.03	-16440973.24	5.973384e+07	-5.830004e+07	16097890.80	-15432784.08	781
Japan	25702199.07	-25325328.80	7.029119e+07	-6.762307e+07	25759121.67	-27173445.97	602
United Kingdom	14393798.08	-13295555.74	6.121933e+07	-5.897131e+07	34152790.05	-32834074.66	851
United States	33190126.28	-33847179.13	3.190755e+08	-3.456493e+08	87960396.35	-89719182.63	3298

10 rows × 32 columns

In [24]:
```
titles = [x[0] for x in df2018_2019_exp_piv.columns][::2]

fig,axs = plt.subplots(4,4,figsize=(15,22))
axs = axs.flatten()

for i in range(0,16):
    tmp_df = df2018_2019_exp_piv.iloc[:,i*2:(i+1)*2].copy()
    tmp_df.columns = ["2018","2019"]
    tmp_df.plot(kind="barh",sharey=True,
                                                stacked=True,
                                                legend=False,
                                                title=titles[i],ax=axs[i],
                                                )
    sns.despine()
    axs[i].legend(fontsize=10)
```




```
Remarkable categories and countries:
Strength and weakness
- China: Mineral, prominent
         Vegetables
         Animal
         wood
- USA: large in each categories
        Textile
```

Comparison of import among countries in 2018 and 2019

In [25]:
```
df2018_imp = df2018[["Partner Name","Year","Product Group","Import (US$ Tho
df2019_imp = df2019[["Partner Name","Year","Product Group","Import (US$ Tho
df2019_imp["Import (US$ Thousand)"] = df2019_imp["Import (US$ Thousand)"]*(

df2018_2019_imp = pd.concat([df2018_imp,df2019_imp ])
df2018_2019_imp.head()
```

Out[25]:

	Partner Name	Year	Product Group	Import (US$ Thousand)
0	United States	2018	Animal	2.725376e+07
1	United States	2018	Chemicals	1.992497e+08
2	United States	2018	Food Products	4.543039e+07
3	United States	2018	Footwear	1.027420e+06
4	United States	2018	Fuels	1.866005e+08

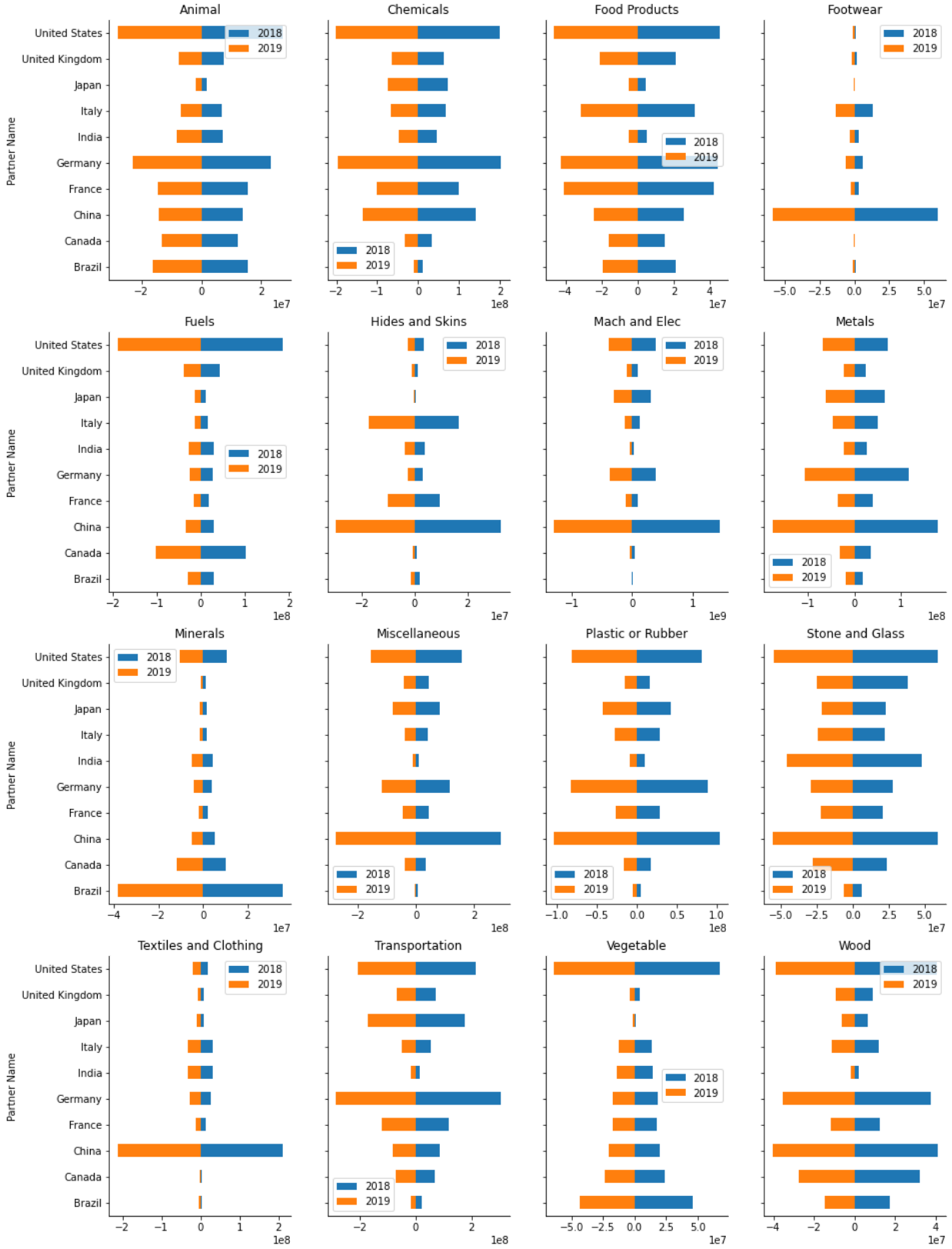
In [26]:
```
df2018_2019_imp_piv = df2018_2019_imp.pivot_table(index="Partner Name",colu
df2018_2019_imp_piv
```

Out[26]:

Product Group	Animal		Chemicals		Food Products		
Year	2018	2019	2018	2019	2018	2019	
Partner Name							
Brazil	15606572.62	-16315503.88	1.224826e+07	-1.131585e+07	21167159.01	-19531441.12	124
Canada	12311063.03	-13290351.48	3.263724e+07	-3.289952e+07	15249785.52	-16081800.84	13
China	13906857.69	-14117630.79	1.418938e+08	-1.346791e+08	25401544.51	-24090965.84	6037
France	15357902.72	-14513304.61	9.810821e+07	-1.020266e+08	42050243.12	-40984708.23	303
Germany	23056377.95	-22920338.51	2.030625e+08	-1.969241e+08	44291652.31	-42456662.09	605
India	7085532.08	-8360596.95	4.455538e+07	-4.768041e+07	5240238.69	-5166647.45	355
Italy	6794929.11	-6868738.41	6.803527e+07	-6.800381e+07	31751588.33	-31415604.66	1305
Japan	1758446.32	-1793970.15	7.214194e+07	-7.478562e+07	4347000.48	-4705111.86	27
United Kingdom	7463848.87	-7514938.62	6.343358e+07	-6.337713e+07	21273807.85	-20915778.98	179
United States	27253762.22	-28000361.18	1.992497e+08	-2.011104e+08	45430388.25	-46271482.93	102

10 rows × 32 columns

In [27]:
```
titles = [x[0] for x in df2018_2019_imp_piv.columns][::2]

fig,axs = plt.subplots(4,4,figsize=(15,22))
axs = axs.flatten()

for i in range(0,16):
    tmp_df = df2018_2019_imp_piv.iloc[:,i*2:(i+1)*2].copy()
    tmp_df.columns = ["2018","2019"]
    tmp_df.plot(kind="barh",sharey=True,
                                                stacked=True,
                                                legend=False,
                                                title=titles[i],ax=axs[i],
                                                )
    sns.despine()
    axs[i].legend(fontsize=10)
```




```
Import
Strength and weakness
Remarkable categories and countries:
- China: footwear, Hides and Skins, Mach and Elec, Metals, Miscellaneous,
Textiles
- Brazil: Minerals
- USA: Vege, Fueks
```

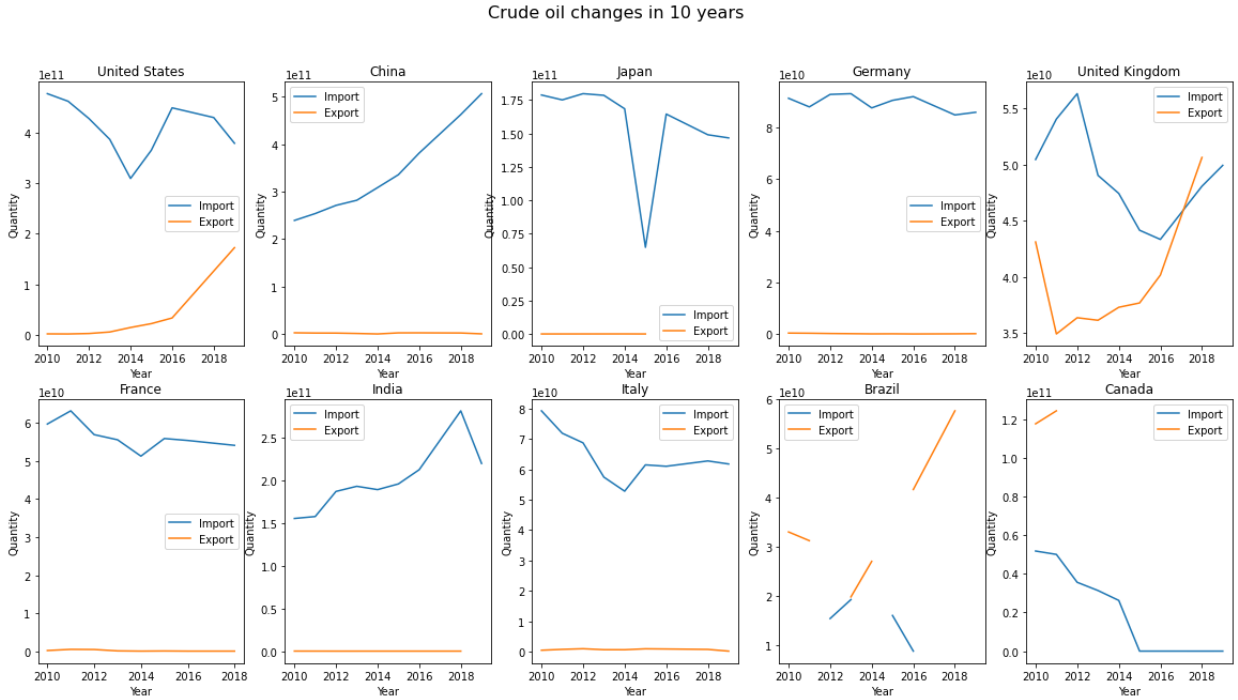
Comparison between import and export in one country in 2018 and 2019

In [28]:
```
# export
df2018_exp=df2018[["Partner Name","Year","Product Group","Export (US$ Thous
df2019_exp=df2019[["Partner Name","Year","Product Group","Export (US$ Thous
df2019_exp["Export (US$ Thousand)"]=df2019_exp["Export (US$ Thousand)"]*(-1

df2018_2019_exp=pd.concat([df2018_exp,df2019_exp ])
df2018_2019_exp_piv=df2018_2019_exp.pivot_table(index="Product Group",colum

# import
df2018_imp=df2018[["Partner Name","Year","Product Group","Import (US$ Thous
df2019_imp=df2019[["Partner Name","Year","Product Group","Import (US$ Thous
df2019_imp["Import (US$ Thousand)"]=df2019_imp["Import (US$ Thousand)"]*(-1

df2018_2019_imp=pd.concat([df2018_imp,df2019_imp ])
df2018_2019_imp_piv=df2018_2019_imp.pivot_table(index="Product Group",colum

# prepare title
titles=[x[0] for x in df2018_2019_exp_piv.columns][::2]


for i in range(0,10):
    fig,axs=plt.subplots(1,2,figsize=(6,5))

    # plot export
    tmp_df_exp=df2018_2019_exp_piv.iloc[:,i*2:(i+1)*2].copy()
    tmp_df_exp.columns=["2018","2019"]
    tmp_df_exp.plot(kind="barh",sharey=True,
                                                stacked=True,
                                                legend=False,
                                                title="{} Export".format(t
                                                )

    sns.despine()
    axs[0].legend(fontsize=8,loc="upper right")

    # plot import
    tmp_df_imp=df2018_2019_imp_piv.iloc[:,i*2:(i+1)*2].copy()
    tmp_df_imp.columns=["2018","2019"]
    tmp_df_imp.plot(kind="barh",sharey=True,
                                                stacked=True,
                                                legend=False,
                                                title="{} Import".format(t
                                                )

    sns.despine()
    axs[1].legend(fontsize=8,loc="upper right")
    plt.show()
```


```
Brainstorm for this part:
- Export vs import in the same country, also does the countries rely on
import? Features?
- relationship between export and import
- difference between developed countries and developing countries
Attention: quantity difference
```

Q3: Analysis Of The Trade For Crude Oil Among Ten Years

In [29]:
```
plt.figure(figsize=(20,10))


# use pandas to read the excel file into a dataframe
data = pd.read_excel("Q3.xlsx", engine="openpyxl", sheet_name="By-HS6Produc

country = ["United States","China","Japan","Germany","United Kingdom","Fran

# Line Plot for each country
num = 1
for c in country:
    # first determine import and export quantity over years for each countr
    # find rows for each country, separate by import and export, then extra
    Import = data.loc[(data["Reporter"] == c) & (data["TradeFlow"] == "Impo
    Export = data[(data["Reporter"] == c) & (data["TradeFlow"] == "Export")

    # get currect axis, draw import and export on the same graph
    plt.subplot(2, 5, num)
    ax = plt.gca()
    Import.plot(kind = 'line', x = "Year", y = "Quantity", ax = ax, label="
    Export.plot(kind = 'line', x = "Year", y = "Quantity", ax = ax, label="
    ax.set_ylabel("Quantity")
    ax.set_title(c)
    num += 1
plt.suptitle("Crude oil changes in 10 years", size=16)
plt.show()
```



In [30]:
```
# Bar chart to compare all countries' import and export quantities in 2019
# find all rows where year is 2019, separate by import and export
data_2019_import = data[(data["Year"] == 2019) & (data["TradeFlow"] == "Imp
data_2019_export = data[(data["Year"] == 2019) & (data["TradeFlow"] == "Exp

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(20,10))

data_2019_import.plot.bar(x = "Reporter", y = "Quantity", rot = 70, title =
data_2019_export.plot.bar(x = "Reporter", y = "Quantity", rot = 70, title =
plt.suptitle("Top 10 countries import and export in 2019", size=16)
plt.show()
```