

Homework 1 – First visit in Kaggle data

| 資訊三乙 E14084117 黃子峻

1. Dataset description 資料集與資料欄位描述

資料集名稱—**Global Commodity Trade Statistics**

1.1 基本資料集描述

1.2 資料特性描述

2. Data analysis 與產出

2.1 現有分析方法 Notebook 1 介紹—**Deep Analysis of China Trade Market**

2.1.1 Import Python modules and dataset

2.1.2 Read dataset

2.1.3 Analysis trade in China

2.1.4 China Import 1992 vs 2016

2.1.5 China Export 1992 vs 2016

2.1.6 China VS World

2.2 現有分析方法 Notebook 2 介紹—**Sheeps vs Goats**

2.2.1 Module import

2.2.2 Read dataset

2.2.3 Deal with missing values

2.2.4 Split dataframe

2.2.5 Plotting the weight in kgs of imported of Sheep & Goats

2.2.6 Analysis of sheep

2.3 Notebook 1 與 Notebook 2 方法比較

2.4 資料分析價值與可能產出

2.4.1 現有分析方法的產出

2.4.1 其他分析可能的產出與價值

3. My insight

1. Dataset description 資料集與資料欄位描述

資料集名稱—**Global Commodity Trade Statistics**

Global Commodity Trade Statistics

Three decades of global trade flows

k <https://www.kaggle.com/unitednations/global-commodity-trade-statistics>



1.1 基本資料集描述

這個資料集由 **United Nations Statistics Division** 提供，原資料來



自 <http://data.un.org/Explorer.aspx>。

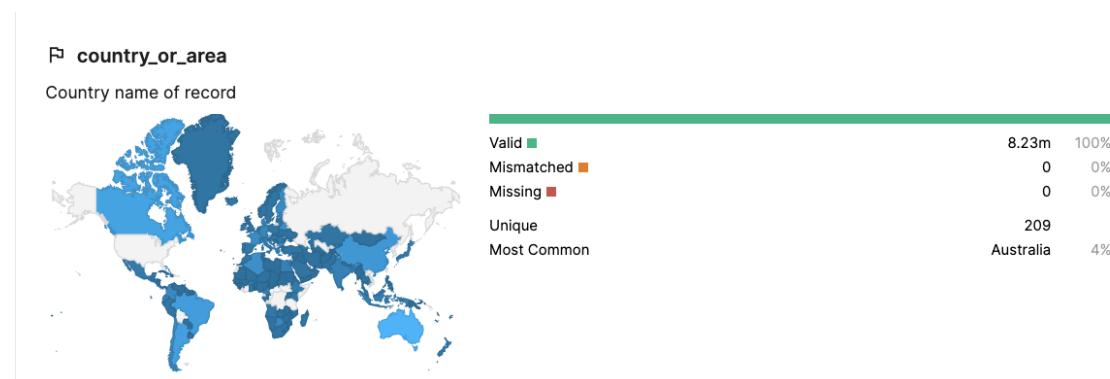
- 這個資料集涵蓋了過去近 30 年（1988 ~ 2016）地球上大多數國家的 5,000 種商品的進出口量。
- 對於如此長時間且大量的統計資料可能存在一些奇怪的錯誤數據，在介紹頁面中有提到「英國的茶葉進口看似相當準確，但阿富汗在 2016 年只輸出 51 隻羊值得懷疑」，因此即便資料來源於知名的世界組織，在使用數據時仍然需要注意數據的合理性。
- **License** – available free of charge and may be copied freely, duplicated and further distributed provided that UNdata is cited as the reference.
- 資料集 **commodity_trade_statistics_data.csv** 為 csv 格式，共 1.23 GB
- 欄位描述



下列圖表取自 Kaggle 的基本分析功能

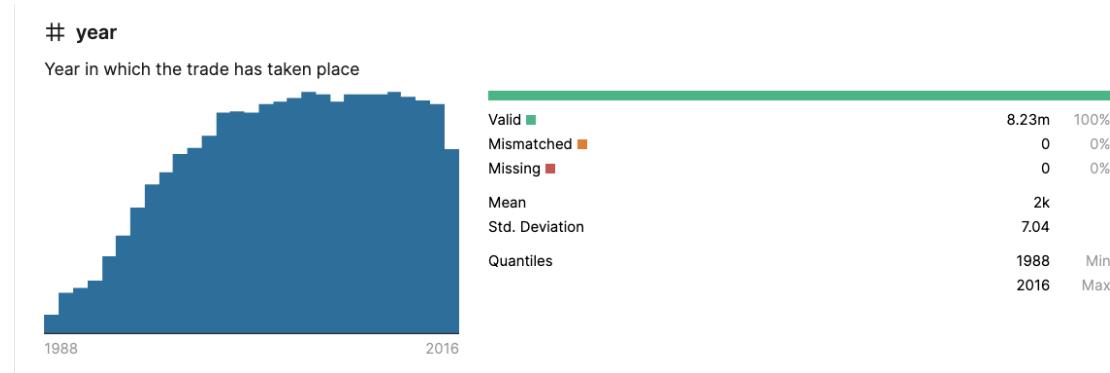
- **country_or_area (str)**

- 209 種國家或地區名稱，以開頭大寫的英文字串（含空白，非縮寫）呈現
- 無 Mismatched 和 Missing 資料，最多筆 Australia 的資料



- **year (int)**

- 西元年份（1988 ~ 2016），下圖以資料筆數對年份做圖，交易量有增長的趨勢。
- 無 Mismatched 和 Missing 資料

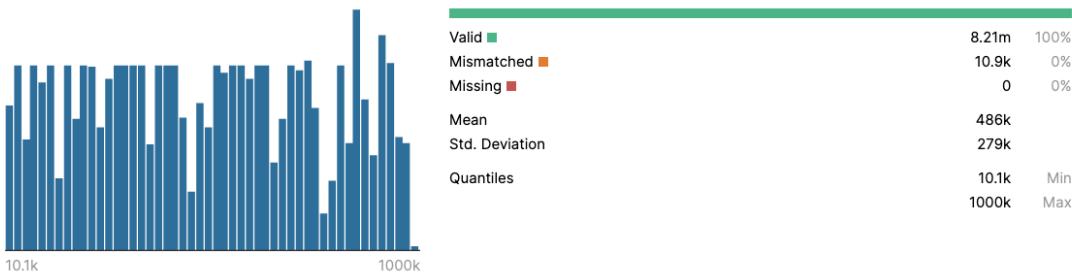


- **comm_code (int)**

- 根據世界海關組織：統一商品描述和編碼系統通常稱為 Harmonized System 或簡稱 HS，是世界海關組織 (WCO) 制定的多用途國際產品命名法。它包括大約 5,000 多個商品種類。
- 每個編碼都由一個六位數的代碼標識，由定義明確的規則支持，以實現統一分類。
- 有關更多信息，請參見此處<http://www.wcoomd.org/en/topics/nomenclature/overview/what-is-the-harmonized-system.aspx>
- 注意有 10.9k 筆 Mismatched 資料，但相對於 8.21m 筆有效資料來說，占比接近 0%，但若 Mismatched 資料非平均分佈也許會造成影響（此處看不出 Mismatched 的定義）
- 無資料 Missing

comm_code

Per the World Customs Organization: The Harmonized Commodity Description and Coding System generally referred to as "Harmonized System" or simply "HS" is a multipurpose international product nomenclature developed by the World Customs Organization (WCO). It comprises about 5,000 commodity groups; each identified by a six digit code, arranged in a legal and logical structure and is supported by well-defined rules to achieve uniform classification. For more, see here:
<http://www.wcoomd.org/en/topics/nomenclature/overview/what-is-the-harmonized-system.aspx>



◦ commodity (str)

- 對於對應的 **comm_code** 的描述，例如 “Horses, live pure-bred breeding”
- 無 Mismatched 和 Missing 資料
- 5031 種描述，與 **comm_code** 所標示的商品分類數量（大約 5,000 個）接近

A commodity

The description of a particular commodity code, i.e. "Horses, live pure-bred breeding"

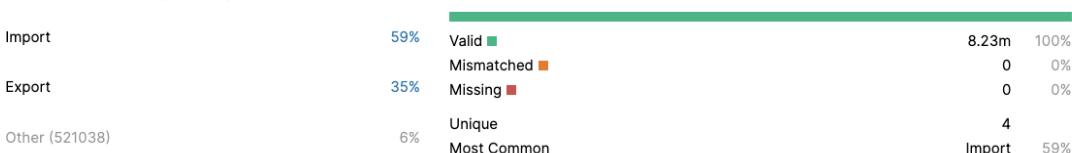


◦ flow (str)

- 以 “Export” / “Import” 標示進出口，分別占 35% 和 59%，出口資料遠多於進口資料，其餘 6% 為兩種其他標註，分別為 “Re-Export” 和 “Re-Import”
- Re-Export：又稱轉口貿易 (Entrepot Trade)，是指出口商通過中間商與進口商發生買賣關係，而後將貨物直接從出口國運往進口國的貿易方式。
- Re-Import：是指將貨物進口到以前從該國出口的國家。由於一個國家的價格因國家而異，因此再進口商可能會在另一個國家購買貨物，然後再進口這些貨物，以降低貨物在該國的銷售價格。
- 無 Mismatched 和 Missing 資料

A flow

Flow of trade i.e. Export, Import

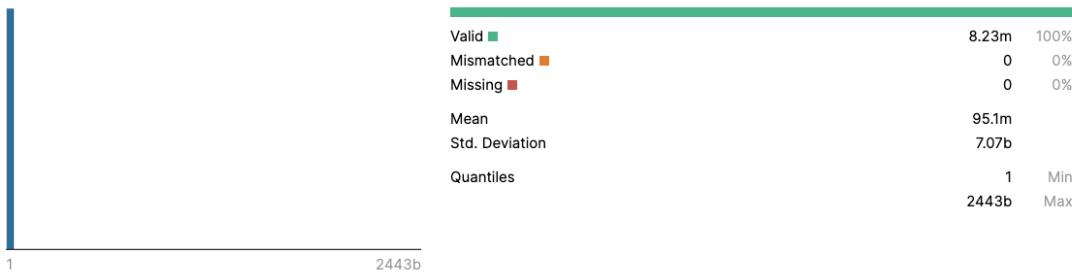


◦ trade_usd (int)

- 美元交易金額
- 下圖資料筆數對 USD 做圖中，明顯的藍色 bar 為 1.00 ~ 488866210482.26 USD 共計 8223674 筆，在圖表中分佈相當不平均，可能是做圖尺度設定不佳或資料特性造成。從平均 95.1m USD、最大 2443b USD、標準差 7.07b USD 來看，沒有藍色 bar 的部分應該要存在少量資料分佈。
- 無 Mismatched 和 Missing 資料。

trade_usd

Value of the trade in USD



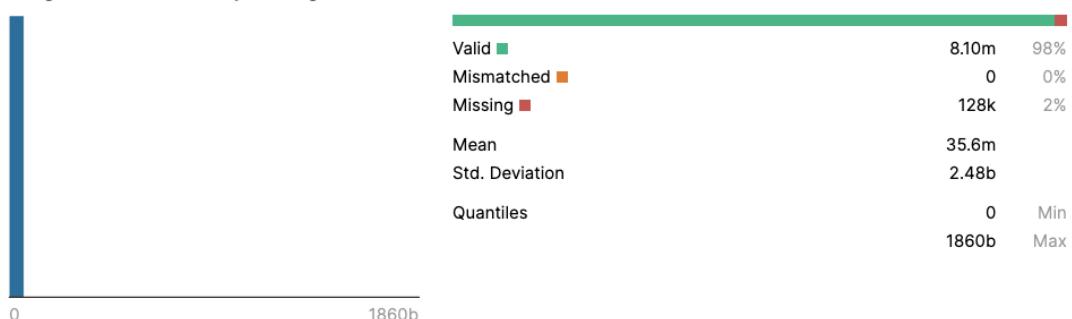
- **weight_kg** (int)

- 下圖為重量 (kg) 的 histogram，明顯的藍色 bar 為 0.00 ~ 37202664820.00 kg 共計 8096431 筆，在圖表中分佈相當不平均，可能是做圖尺度設定不佳或資料特性造成。從平均 35.6m kg、最大 1860b kg、標準差 2.48b kg來看，沒有藍色 bar 的部分應該要存在少量資料分佈。

- 無 Mismatched 資料， Missing 資料占總數的 2%應影響不大。

weight_kg

Weight of the commodity in Kilograms



- **quantity_name** (str)

- 描述給定項目類型的單位，例如 Number of Items, Weight in Kilograms。
- 最多使用的是Weight in Kilograms (80%)，次多為 Number of items (9%)，其他 10 種單位則佔 11%。
- 無 Mismatched 和 Missing 資料。

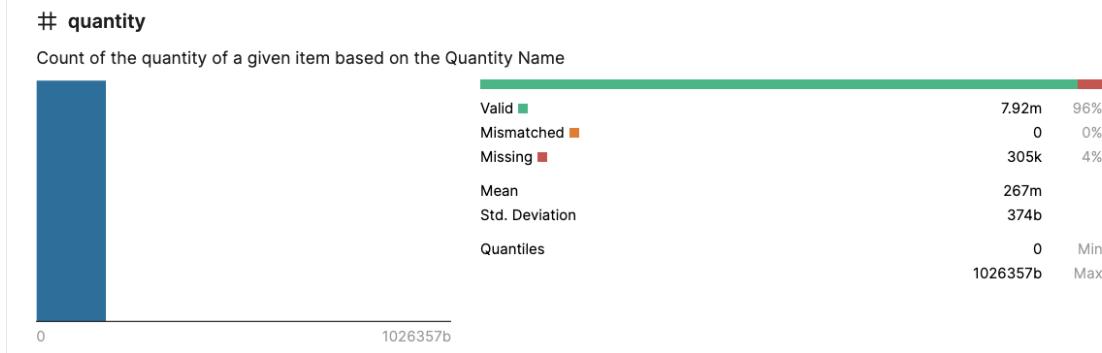
A quantity_name

A description of the quantity measurement type given the type of item (i.e. Number of Items, Weight in Kilograms, etc.)



- **quantity** (int)

- 根據 quantity_name 細分項目的數量
- 無 Mismatched 資料， Missing 資料占總數的 4%。
- 下圖為 quantity 的 histogram，由於使用的單位可能不一樣，我覺得應修正為針對不同的 quantity_name 將 quantity 做圖才有意義。



- category (str)

- 98種有編號的物品類別，例如占做多數的 02_meat_and_edible_meat_offal (1%)。
- 無 Mismatched 和 Missing 資料。

A category

Category to identify commodity



1.2 資料特性描述

以下列出一些此資料集的特性：

- 此資料依據想分析的對象不同，可以有多種不同的用法：
 - 例如希望分析的對象是進口、出口的金額，就可以分析各個國家的進出口金額、進出口金額隨年份變化的趨勢等議題。
 - 如果分析的對象是某商品的重量時，就可以分析各個國家的進出口需求量、不同國家每公斤的價格等議題。
- 此資料集的 comm_code 大約有標示 5,000 種商品種類，只要搭配世界海關組織的Harmonized System 編碼系統查詢，就能獲得非常詳細的進出口商品分類。也可以直接從 commodity 獲得不同 comm_code 的描述，省去查表麻煩，相當方便。
- United Nations Statistics Division 提供的進出口資料，具有一定程度的可信度，從欄位資料缺失值最多占總數的 5% 來看，此資料算是相當完整。
- 此資料集的時間範圍從 1988 ~ 2016 年，而在原始資料的網站上可以獲得直至 2019 年的數據更新，因此只此不會受限於此資料集只有到 2016 年的缺點，仍然可以持續深入研究相關議題。
- country_or_area、flow、category 等 str 類別的欄位資料都可以近一步做數值編碼，方便用於機器學習預測任務、或其他數值分析。
- 此資料集也適合與其他資料集搭配做分析，例如過去就有人拿此資料集搭配 2016 Global Ecological Footprint 與 Multidimensional Poverty Measures 兩個資料集，來做平窮與貿易對碳足跡的影響分析。（參考 <https://www.kaggle.com/jroachel/how-poverty-and-trade-impact-carbon-footprint>）

2. Data analysis 與產出

2.1 現有分析方法 Notebook 1 介紹—Deep Analysis of China Trade Market



選擇原因：此 Notebook 演示了如何在此全球貿易資料集中，分析單一國家的進出口變化狀況。（以中國至 2016 的 25 年資料為例）

分析目標：

- 中國每年的進口、出口金額變化趨勢與比較。
- 中國 1992 與 2016 年，前幾多進口、出口金額的項目。
- 一些商品項目在中國從 1992 至 2016 年的進口、出口金額變化趨勢。
- 中國的進口、出口金額佔世界總進口、出口金額比例的逐年變化趨勢。
- 中國與其他幾國的進口、出口金額比較。

2.1.1 Import Python modules and dataset

此步驟導入會用到的 python module

- pandas → 用於表格數據操作
- np → 用於矩陣（numpy array）計算
- plotly, matplotlib → 資料視覺化
- wordcloud → 文字雲視覺化
- os → 在此用於讀取 input 資料夾中的檔案

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
# plotly
import plotly.plotly as py
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go
# word cloud library
from wordcloud import WordCloud
# matplotlib
import matplotlib.pyplot as plt
# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory
import os
print(os.listdir("../input"))
```

2.1.2 Read dataset

此步驟以 `pandas` 讀取 csv 格式的 dataset，並進行簡單查看。

`cn_df = df[df.country_or_area == "China"]` 這行程式碼用來建立一個名為 `cn_df` 的新 dataframe，此 data dataframe 只包含原始資料中 `"country_or_area"` 值為 `"China"` 的 row data，目的是只提取出中國的貿易資料。

```
df=pd.read_csv('../input/commodity_trade_statistics_data.csv', low_memory=False)
cn_df = df[df.country_or_area == "China"]
df.head(3)
```

country_or_area	year	comm_code	commodity	flow	trade_usd	weight_kg	quantity_name	quantity	category
0	Afghanistan	2016	10410	Sheep, live	Export	6088	2339	Number of items	51
1	Afghanistan	2016	10420	Goats, live	Export	3958	984	Number of items	53
2	Afghanistan	2008	10210	Bovine animals, live pure-bred breeding	Import	1026804	272	Number of items	3769

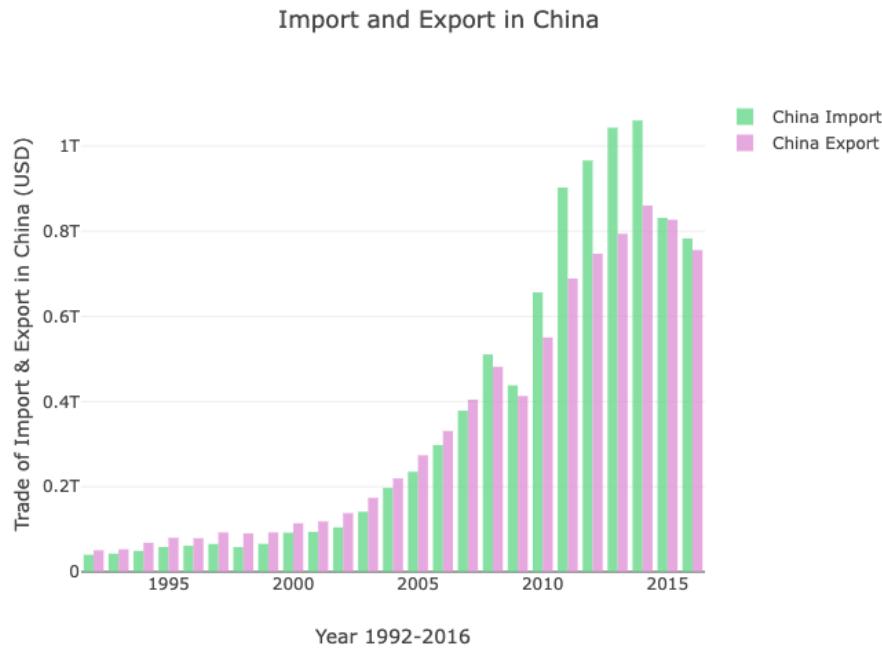
2.1.3 Analysis trade in China

此步驟以 `plotly` 視覺化分析中國每年的進口、出口美元金額，採用上一步得到的 `cn_df` 這個 data frame。

```

cn_i = cn_df[(cn_df.flow == 'Import') & (cn_df.comm_code!= 'TOTAL')].groupby(['year'],as_index=False)[['trade_usd']].agg('sum')
cn_e = cn_df[(cn_df.flow == 'Export') & (cn_df.comm_code!= 'TOTAL')].groupby(['year'],as_index=False)[['trade_usd']].agg('sum')
trace1 = go.Bar( x = cn_i.year,
                 y = cn_i.trade_usd,
                 name = "China Import",
                 marker = dict(color = 'rgba(102, 216, 137, 0.8)'),)
trace2 = go.Bar( x = cn_e.year,
                 y = cn_e.trade_usd,
                 name = "China Export",
                 marker = dict(color = 'rgba(224, 148, 215, 0.8)'),)
data = [trace1, trace2]
layout = { 'xaxis': { 'title': 'Year 1992-2016'},
           'yaxis': { 'title': 'Trade of Import & Export in China (USD)' },
           'barmode': 'group',
           'title': 'Import and Export in China'}
fig = go.Figure(data = data, layout = layout)
iplot(fig)

```



觀察上圖發現：

- 進入21世紀以後（2000年後），中國的進出口貿易呈急遽成長（金額逐年增加更快速），或許與2001中國加入世貿組織（WTO）有著密切關聯。
- 或許是受2008年金融危機影響，進出口受到了一些衝擊，進出口金額在逐年增長的趨勢中出現下降的反轉，但在2010年以後仍恢復持續性成長。
- 相較於出口，進口的波動會比較大。



接著繼續深入分析，找出什麼貨品分別在進口和出口中扮演著重要的角色？

2.1.4 China Import 1992 vs 2016

此步驟希望從圖表中比較中國 1992 與 2016 年進口金額前十多的項目差別，以美元金額作為單位。

```

temp = cn_df[(cn_df.year==1992) & (cn_df.flow=='Import')].sort_values(by="trade_usd", ascending=False).iloc[1:11, :]
cn_1992i = temp.sort_values(by="trade_usd", ascending=True)
trace1 = go.Bar(x = cn_1992i.trade_usd,
                 y = cn_1992i.commodity,
                 marker = dict(color = 'rgba(152, 213, 245, 0.8)'),
                 orientation = 'h')

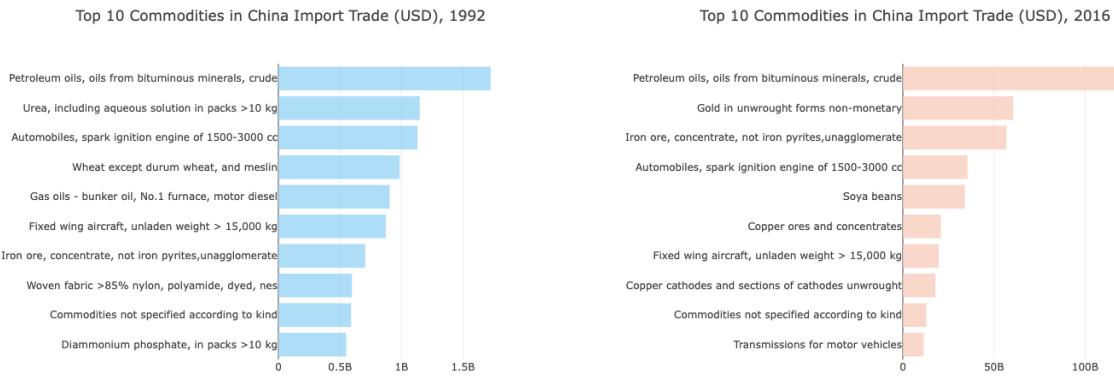
```

```

data = [trace1]
layout = {'xaxis': {'title': 'Trade in USD'},
          '#yaxis': {'automargin':True,},
          'title': "Top 10 Commodities in China Import Trade (USD), 1992"}
fig = go.Figure(data = data, layout = layout)
iplot(fig)

temp1 = cn_df[(cn_df.year==2016) & (cn_df.flow=='Import')].sort_values(by="trade_usd", ascending=False).iloc[1:11, :]
cn_2016i = temp1.sort_values(by="trade_usd", ascending=True)
trace1 = go.Bar(x = cn_2016i.trade_usd,
                 y = cn_2016i.commodity.tolist(),
                 marker = dict(color = 'rgba(249, 205, 190, 0.8)'),
                 orientation = 'h')
data = [trace1]
layout = {'xaxis': {'title': 'Trade in USD'},
          '#yaxis': {'automargin':True,},
          'title': "Top 10 Commodities in China Import Trade (USD), 2016"}
fig = go.Figure(data = data, layout = layout)
iplot(fig)

```



對比1992 與 2016 年的兩張圖表，25年間的進口貿易發生了很大的變化：

- 從金額數目上對比，進口量提升至25年前的幾十倍乃至幾百倍，比如石油的需求提升了整整116倍。
- 從項目上看，雖然石油一直處於進口貿易的榜首，但2016年幾乎其他所有主要進口項目都和25年前（1992）不一樣。比如，1992年肥料為進口貿易額的第二，而在2016年，肥料在圖中已不見踪影，取而代之的是鐵礦，黃金等一些其他資源。



所以接下來分析1992至2016年間，石油、鐵礦、肥料的變化趨勢，看看是否可以從中窺探出什麼信息？

這一步驟以 `comm_code` 選取石油、鐵礦、肥料，將 1992 至 2016 年的進口金額以 `plotly` 畫成折線圖。

```

petro = cn_df[(cn_df.comm_code == '270900') & (cn_df.flow == 'Import')]
urea = cn_df[(cn_df.comm_code == '310210') & (cn_df.flow == 'Import')]
iron = cn_df[(cn_df.comm_code == '260111') & (cn_df.flow == 'Import')]
trace1 = go.Scatter(x = petro.year,
                     y = petro.trade_usd,
                     mode = "lines+markers",
                     name = "Petroleum oils",
                     marker = dict(color = 'rgba(255, 196, 100, 0.8)'))
trace2 = go.Scatter(x = urea.year,
                     y = urea.trade_usd,
                     mode = "lines+markers",
                     name = "Urea",
                     marker = dict(color = 'rgba(241, 130, 133, 0.8)'))
trace3 = go.Scatter(x = iron.year,
                     y = iron.trade_usd,
                     mode = "lines+markers",
                     name = "Iron ore",
                     marker = dict(color = 'rgba(130, 241, 140, 0.8)'))
data = [trace1, trace2, trace3]
layout = dict(title = "Some Commodities' value in China Import Trade (USD)",
              xaxis= dict(title= 'Year 1992-2016', ticklen= 5,zeroline= False),

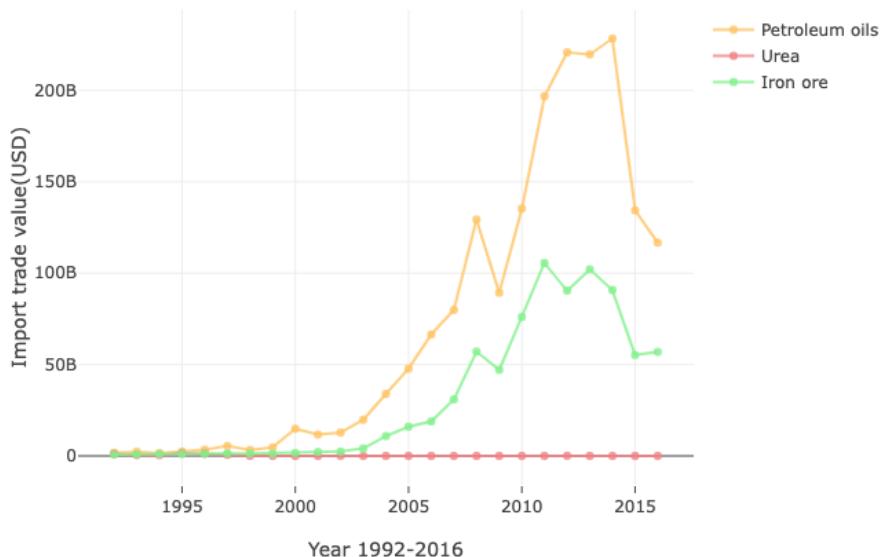
```

```

yaxis = {'title': 'Import trade value(USD)'})
fig = dict(data = data, layout = layout)
iplot(fig)

```

Some Commodities' value in China Import Trade (USD)



觀察上面圖表可以發現，石油與鐵礦增長趨勢與中國總進口貿易額在 1992 至 2016 的趨勢相同，都呈現出爆炸式增長模式，然後在 2008 年金融危機以後出現下降，回升後又在近幾年（近 2016 年）呈現下降。

如果只看肥料（Urea）進口金額變化，不難發現在 1995 年以後就驟降至低潮，並保持至 2016 年。從中可以映射出中國從農業時代到工業時代的轉型，但更大的原因應該是中國自主生產力的提升導致某些產品的進口額的減少。

2.1.5 China Export 1992 vs 2016

此步驟希望從圖表中比較中國 1992 與 2016 年出口金額前十多的項目差別，以美元金額作為單位。

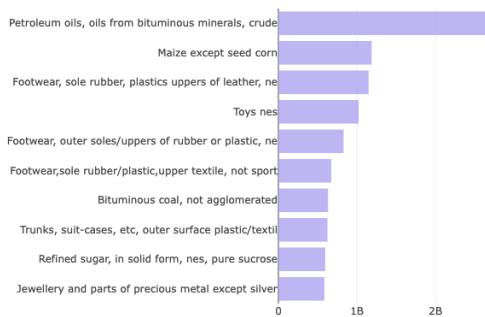
```

temp = cn_df[(cn_df.year==1992) & (cn_df.flow=='Export')].sort_values(by="trade_usd", ascending=False).iloc[1:11, :]
cn_1992e = temp.sort_values(by="trade_usd", ascending=True)
trace1 = go.Bar(x = cn_1992e.trade_usd,
                 y = cn_1992e.commodity,
                 marker = dict(color = 'rgba(173, 164, 239, 0.8)'),
                 orientation = 'h')
data = [trace1]
layout = {#      'xaxis': {'title': 'Trade in USD'},
          'yaxis': {'automargin':True, },
          'title': "Top 10 Commodities in China Export Trade (USD), 1992"}
fig = go.Figure(data = data, layout = layout)
iplot(fig)

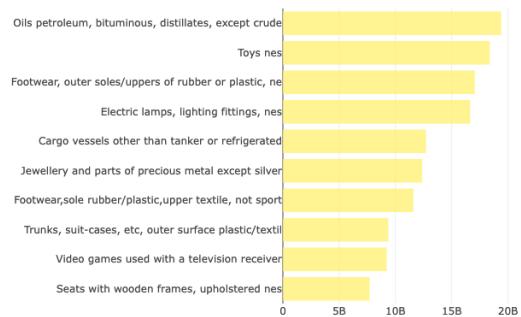
temp1 = cn_df[(cn_df.year==2016) & (cn_df.flow=='Export')].sort_values(by="trade_usd", ascending=False).iloc[1:11, :]
cn_2016e = temp1.sort_values(by="trade_usd", ascending=True)
trace1 = go.Bar(x = cn_2016e.trade_usd,
                 y = cn_2016e.commodity,
                 marker = dict(color = 'rgba(255, 241, 117, 0.8)'),
                 orientation = 'h')
data = [trace1]
layout = {#      'xaxis': {'title': 'Trade in USD'},
          'yaxis': {'automargin':True, },
          'title': "Top 10 Commodities in China Export Trade (USD), 2016"}
fig = go.Figure(data = data, layout = layout)
iplot(fig)

```

Top 10 Commodities in China Export Trade (USD), 1992



Top 10 Commodities in China Export Trade (USD), 2016



比較兩圖表可以發現：

- 中國出口貿易額中，石油、玩具、鞋類等，持續大比重佔據出口金額。
- 相較於1992年有大量煤和玉米輸出，2016年有更多的電燈泡或者塑料部件輸出。



接下來就分析一下玩具、玉米還有電燈25年間出口狀況變化。

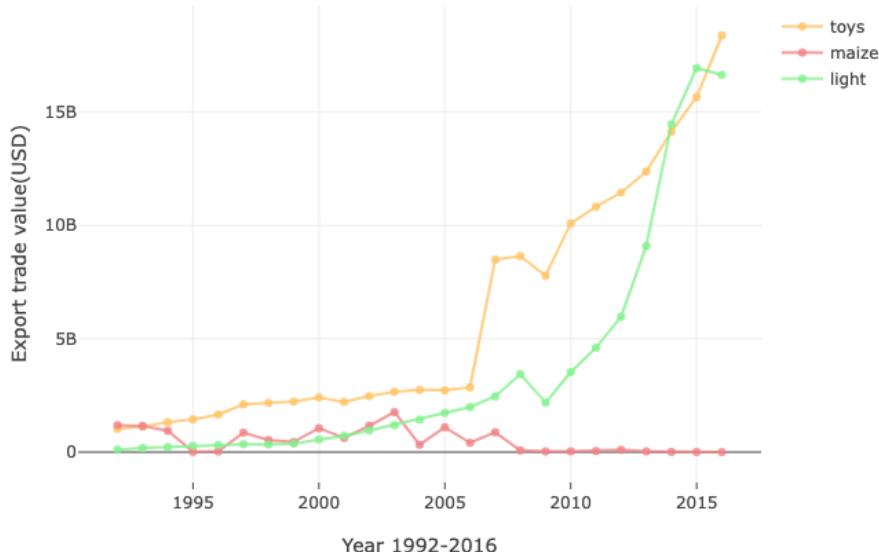
這一步驟以 `comm_code` 選取玩具、玉米還有電燈，將 1992 至 2016 年的出口金額以 `plotly` 畫成折線圖。

```

toys = cn_df[(cn_df.comm_code == '950390') & (cn_df.flow == 'Export')]
maize = cn_df[(cn_df.comm_code == '100590') & (cn_df.flow == 'Export')]
light = cn_df[(cn_df.comm_code == '940540') & (cn_df.flow == 'Export')]
trace1 = go.Scatter(x = toys.year,
                     y = toys.trade_usd,
                     mode = "lines+markers",
                     name = "toys",
                     marker = dict(color = 'rgba(255, 196, 100, 0.8)'))
trace2 = go.Scatter(x = maize.year,
                     y = maize.trade_usd,
                     mode = "lines+markers",
                     name = "maize",
                     marker = dict(color = 'rgba(241, 130, 133, 0.8)'))
trace3 = go.Scatter(x = light.year,
                     y = light.trade_usd,
                     mode = "lines+markers",
                     name = "light",
                     marker = dict(color = 'rgba(130, 241, 140, 0.8)'))
data = [trace1, trace2, trace3]
layout = dict(title = "Some Commodities' value in China Export Trade (USD)",
              xaxis= dict(title= 'Year 1992-2016', ticklen= 5, zeroline= False),
              yaxis = {'title': 'Export trade value(USD)'})
fig = dict(data = data, layout = layout)
iplot(fig)

```

Some Commodities' value in China Export Trade (USD)



上面的圖表告訴我們：

- 中國的玩具一直是出口貿易額的巨頭，並保持持續增長。
- 電燈泡的出口暴增以及玉米等農作物出口的下跌再一次印證了中國從傳統農業轉型成現代工業。



既然中國從農業到工業的轉型，使中國經濟獲得了巨大的飛躍。那麼中國在世界中有什麼樣的地位？為什麼是世界第二大經濟體？接下來就擴大到世界範圍來研究中國進出口的貿易數據。

2.1.6 China VS World

Value of the trade in USD

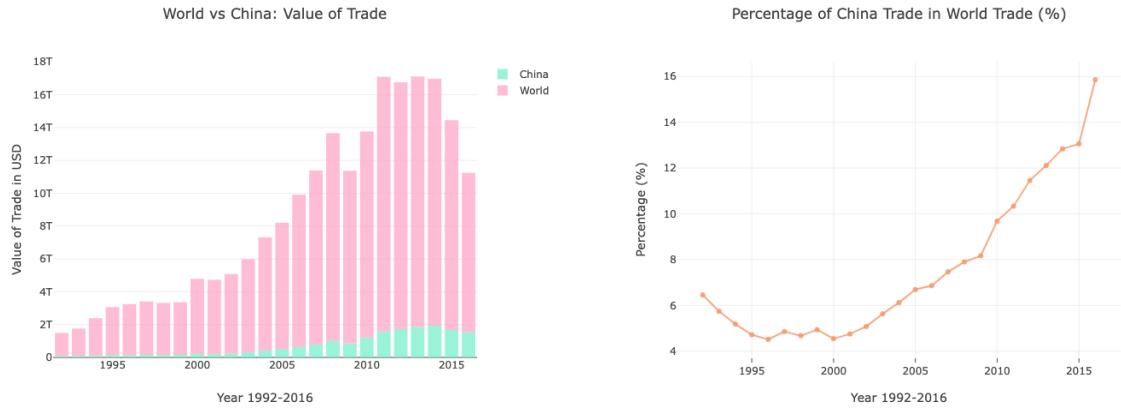
此步驟將中國與世界的貿易金額（進口加出口），做成長條圖與百分比折線圖。

```
cn_trade = cn_df[cn_df.comm_code!= 'TOTAL'].groupby(['year'],as_index=False)[['trade_usd']].agg('sum')
wd_trade = df[(df.year >1991) & (df.comm_code!= 'TOTAL')].groupby(['year'],as_index=False)[['trade_usd']].agg('sum')
# cn_trade.shape
trace0 = {'x': cn_trade.year,
          'y': cn_trade.trade_usd,
          'name': "China",
          'type': 'bar',
          'marker': {'color':'rgba(129, 239, 208, 0.8)'}}
trace1 = {'x': wd_trade.year,
          'y': wd_trade.trade_usd,
          'name': "World",
          'type': 'bar',
          'marker': {'color':'rgba(255, 171, 202, 0.8)'}}
data = [trace0, trace1]
layout = {'xaxis': {'title': 'Year 1992-2016'},
          'yaxis': {'title': 'Value of Trade in USD'},
          'barmode': 'relative',
          'title': 'World vs China: Value of Trade'};
fig = go.Figure(data = data, layout = layout)
iplot(fig)
# ratio
trace3 = go.Scatter(x = cn_trade.year,
                     y = cn_trade.trade_usd/wd_trade.trade_usd*100,
                     mode = "lines+markers",
                     name = "Ratio of China/World",
                     marker = dict(color = 'rgba(245, 150, 104, 0.8)'))
data2 = [trace3]
layout2 = dict(title = 'Percentage of China Trade in World Trade (%)',
               xaxis= dict(title= 'Year 1992-2016', ticklen= 5,zeroline= False),
```

```

yaxis = {'title': 'Percentage (%)'}
fig2 = dict(data = data2, layout = layout2)
iplot(fig2)

```



從上面的圖表中發現，中國在世界進出口貿易額的比重逐年增加，特別是2000年以後。這與2001年中國加入WTO可能有著密切的關係。

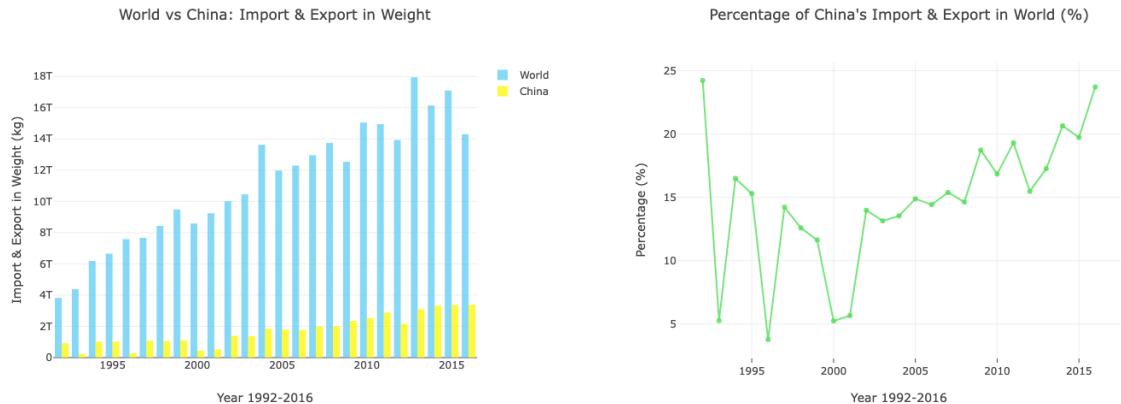
Import & Export in terms of weight

此步驟將上一步驟所在乎的單位從美元換成公斤重（進口加出口），並以相似方法製作圖表。

```

cn_ie = cn_df[cn_df.comm_code!= 'TOTAL'].groupby(['year'],as_index=False)[['weight_kg']].agg('sum')
wd_ie = df[(df.year >1991) & (df.comm_code!= 'TOTAL')].groupby(['year'],as_index=False)[['weight_kg']].agg('sum')
trace1 = go.Bar(x = wd_ie.year,
                 y = wd_ie.weight_kg,
                 name = "World",
                 marker = dict(color = 'rgba(104, 206, 245, 0.8)'),)
trace2 = go.Bar(x = cn_ie.year,
                 y = cn_ie.weight_kg,
                 name = "China",
                 marker = dict(color = 'rgba(255, 248, 12, 0.8)'),)
data = [trace1, trace2]
layout = {'xaxis': {'title': 'Year 1992-2016'},
          'yaxis': {'title': 'Import & Export in Weight (kg)'},
          'barmode': 'group',
          'title': 'World vs China: Import & Export in Weight'}
fig = go.Figure(data = data, layout = layout)
iplot(fig)
# ratio
trace3 = go.Scatter(x = cn_ie.year,
                     y = cn_ie.weight_kg/wd_ie.weight_kg*100,
                     mode = "lines+markers",
                     name = "Ratio of China/World",
                     marker = dict(color = 'rgba(84, 222, 90, 0.8)'))
data2 = [trace3]
layout2 = dict(title = 'Percentage of China\'s Import & Export in World (%)',
               xaxis= dict(title= 'Year 1992-2016', ticklen= 5, zeroline= False),
               yaxis = {'title': 'Percentage (%)'})
fig2 = dict(data = data2, layout = layout2)
iplot(fig2)

```



上面圖表顯示，在 2000 年以前，中國的進出口貿易重量還十分不穩定。但是2000年以後，特別是 2001 年，增長速率空前絕後。這個數據再一次烘托 2001 年加入WTO是中國在世界進出口貿易中起到了轉折性關鍵作用。

China in World 1992 vs 2016

此步驟以繪製圓餅圖的方式，比較美國，日本，中國三大經濟體在 1992 和 2016 年的貿易金額。

```

us_trade = df[(df.country_or_area == "USA") & (df.comm_code!= 'TOTAL')].groupby(['year'],as_index=False)[['trade_usd']].agg('sum')
jp_trade = df[(df.country_or_area == "Japan") & (df.comm_code!= 'TOTAL')].groupby(['year'],as_index=False)[['trade_usd']].agg('sum')

cn_1992 = int(cn_trade[cn_trade.year==1992].iloc[0][1])
us_1992 = int(us_trade[us_trade.year==1992].iloc[0][1])
jp_1992 = int(jp_trade[jp_trade.year==1992].iloc[0][1])
ot_1992 = int(wd_trade[wd_trade.year==1992].iloc[0][1]) - cn_1992 - us_1992 - jp_1992

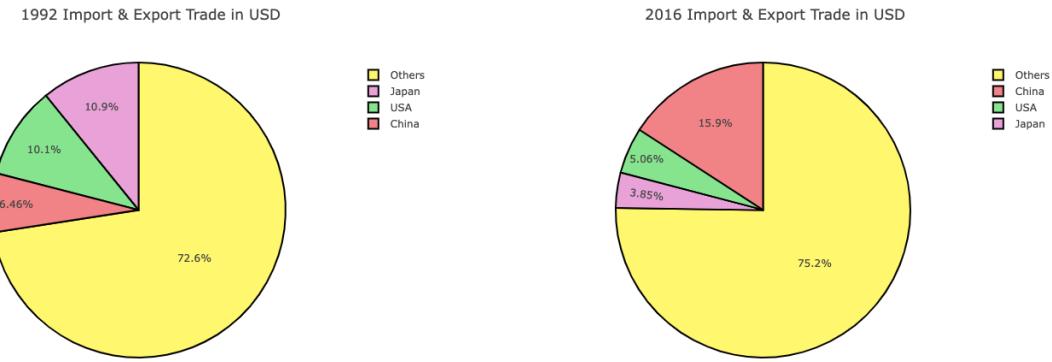
cn_2016 = int(cn_trade[cn_trade.year==2016].iloc[0][1])
us_2016 = int(us_trade[us_trade.year==2016].iloc[0][1])
jp_2016 = int(jp_trade[jp_trade.year==2016].iloc[0][1])
ot_2016 = int(wd_trade[wd_trade.year==2016].iloc[0][1]) - cn_2016 - us_2016 - jp_2016

labels = ['China', 'USA', 'Japan', 'Others']
colors = ['#f18285', '#86e48f', '#e8a2d8', '#ffff76e']

#####
trace = go.Pie(labels=labels, values=[cn_1992, us_1992, jp_1992, ot_1992],
                 marker=dict(colors=colors, line=dict(color='#000', width=2)) )
layout = go.Layout(
    title='1992 Import & Export Trade in USD',
)
fig = go.Figure(data=[trace], layout=layout)
iplot(fig, filename='basic_pie_chart')

#####
trace1 = go.Pie(labels=labels, values=[cn_2016, us_2016, jp_2016, ot_2016],
                 marker=dict(colors=colors, line=dict(color='#000', width=2)) )
layout1 = go.Layout(
    title='2016 Import & Export Trade in USD',
)
fig1 = go.Figure(data=[trace1], layout=layout1)
iplot(fig1, filename='basic_pie_chart1')

```



圖表告訴我們短短25年間，中國掌握世界第一大進出口貿易額。

推論中國經濟的高速發展離不開中國及時地展開工業革命，促進農業向工業化的轉型，這一切也必然離不開2001加入WTO的政策與加入後遇到機遇。

2.2 現有分析方法 Notebook 2 介紹—Sheeps vs Goats

- <https://www.kaggle.com/traker/sheeps-vs-goats>



選擇原因：此 Notebook 演示了如何在此全球貿易資料集中，只選取活羊以及活山羊來做分析的方法。

分析目標：

- 每年活羊、活山羊的全球進口公斤數變化趨勢。
- 進口與出口羊隻最多的是哪些國家。

2.2.1 Module import

此步驟導入會用到的 python module

- pandas → 用於表格數據操作
- np → 用於矩陣（numpy array）計算
- re → regular expression，用於文字處理
- seaborn, matplotlib → 用於視覺化做圖

```
# Processing
import pandas as pd
import numpy as np
np.set_printoptions(threshold=np.nan)
import re
# Visuals
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams['figure.figsize']=(20,10)
```

2.2.2 Read dataset

此步驟以 pandas 讀取 csv 格式的 dataset，並進行簡單查看。

```
df = pd.read_csv("../input/commodity_trade_statistics_data.csv", na_values=["No Quantity", 0.0, ''], sep=',')
```

	country_or_area	year	comm_code	commodity	flow	trade_usd	weight_kg	quantity_name	quantity	category
--	-----------------	------	-----------	-----------	------	-----------	-----------	---------------	----------	----------

	country_or_area	year	comm_code	commodity	flow	trade_usd	weight_kg	quantity_name	quantity	category
0	Afghanistan	2016	10410	Sheep, live	Export	6088	2339.0	Number of items	51.0	01_live_animals
1	Afghanistan	2016	10420	Goats, live	Export	3958	984.0	Number of items	53.0	01_live_animals
2	Afghanistan	2008	10210	Bovine animals, live pure-bred breeding	Import	1026804	272.0	Number of items	3769.0	01_live_animals

使用 `df.count()` 取得每個 column 的資料筆數。

```
country_or_area    8225871
year              8225871
comm_code         8225871
commodity         8225871
flow              8225871
trade_usd         8225871
weight_kg         7745831
quantity_name     7720695
quantity          7693161
category          8225871
dtype: int64
```

使用 `df['commodity'].unique()` 查看商品項目（此處以 `[:5]` 選取五樣商品項目）

```
df['commodity'].unique()[:5]

"""
output
array(['Sheep, live', 'Goats, live',
       'Bovine animals, live pure-bred breeding',
       'Bovine animals, live, except pure-bred breeding',
       'Swine, live except pure-bred breeding > 50 kg'], dtype=object)
"""
```

`commodity` 中被選取的前五個項目分別為：

- 'Sheep, live' → 活羊
- 'Goats, live' → 活山羊
- 'Bovine animals, live pure-bred breeding' → 活的牛類動物，純種繁殖
- 'Bovine animals, live, except pure-bred breeding' → 牛類動物，純種繁殖除外
- 'Swine, live except pure-bred breeding > 50 kg' → 活的豬，除了純種繁殖外

2.2.3 Deal with missing values

此步驟使用 `pandas` 的 `dropna()` method, 捨棄含有缺失值得 row data。

```
df.isnull().sum()
df = df.dropna(how='any').reset_index(drop=True)
df.isnull().sum()
```

執行 `dropna()` 前每個 column 的缺失值總數

```
country_or_area      0
year                0
comm_code           0
commodity           0
flow                0
trade_usd           0
weight_kg           480040
quantity_name       505176
quantity            532710
category            0
dtype: int64
```

執行 `dropna()` 後每個 column 都沒有缺失值

```
country_or_area      0
year                0
comm_code           0
commodity           0
flow                0
trade_usd           0
weight_kg           0
quantity_name       0
quantity            0
category            0
dtype: int64
```

2.2.4 Split dataframe

為了分別分析活羊、活山羊的數據，此步驟將活羊、活山羊的數據從原始的 dataframe 中提取，並建立兩個新的 dataframe，分別為 `dfSheeps`、`dfGoats`，且執行簡單查看。

```
dfSheeps = df[df['commodity']=='Sheep, live'].reset_index(drop=True)
dfGoats = df[df['commodity']=='Goats, live'].reset_index(drop=True)
dfSheeps.head(3)
#dfGoats.head(3)
```

	country_or_area	year	comm_code	commodity	flow	trade_usd	weight_kg	quantity_name	quantity	category
0	Afghanistan	2016	10410	Sheep, live	Export	6088	2339.0	Number of items	51.0	01_live_animals
1	Albania	2014	10410	Sheep, live	Import	21633	8125.0	Number of items	257.0	01_live_animals
2	Albania	2013	10410	Sheep, live	Import	188979	72560.0	Number of items	1740.0	01_live_animals

2.2.5 Plotting the weight in kgs of imported of Sheep & Goats

此步驟主要使用 `pandas` 的 `groupby()`，`sum()` 兩個 method，來統計每年不同 `flow` 的活羊、活山羊的公斤數，並為此數值創建名為 `weight_kg` 的新 column。

```
dfSheepsGrouped = pd.DataFrame({'weight_kg' : dfSheeps.groupby( ["year","flow","commodity"] )["weight_kg"].sum()}).reset_index()
dfGoatsGrouped = pd.DataFrame({'weight_kg' : dfGoats.groupby( ["year","flow","commodity"] )["weight_kg"].sum()}).reset_index()
dfSheepsGrouped.head(3)
# dfGoatsGrouped.head()
```

程式碼所產生的預覽表格

	year	flow	commodity	weight_kg
0	1988	Export	Sheep, live	420020534.0
1	1988	Import	Sheep, live	12930779.0
2	1989	Export	Sheep, live	469372796.0

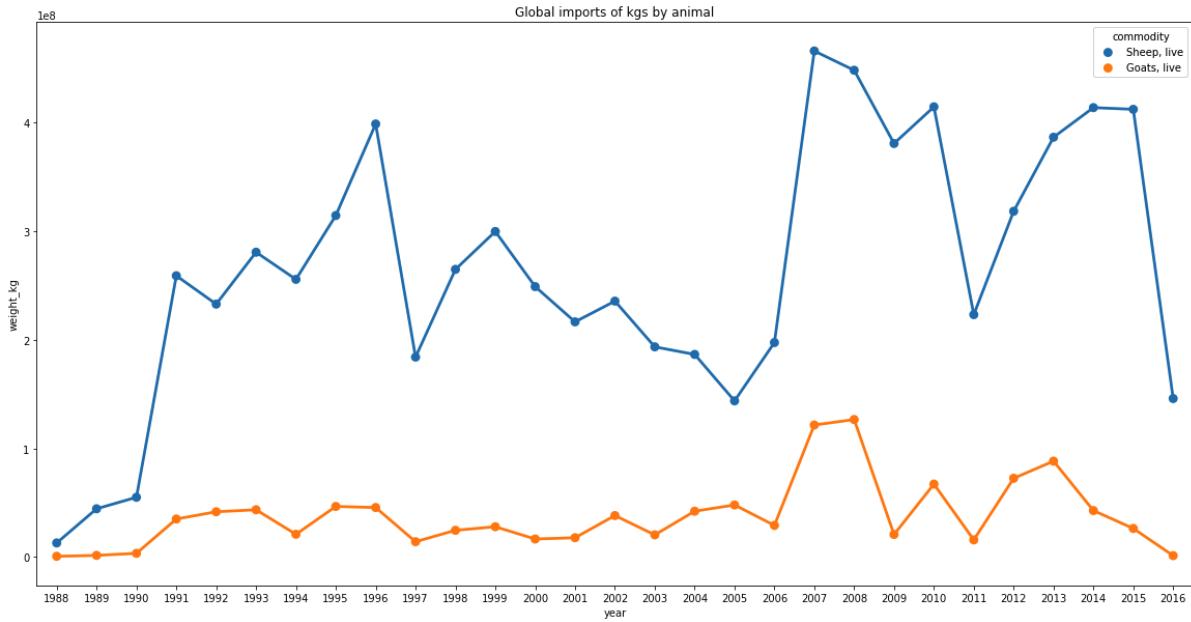
我的註解：

- `dfSheeps.groupby(["year","flow","commodity"])["weight_kg"].sum()` 當中的 `dfSheeps.groupby(["year","flow","commodity"])` 表示依照 `"commodity"`，`"flow"`，`"year"` 的順序將 `dfSheeps` 的 row data 進行分群，然後選取 `"weight_kg"` 這個 column，將其被分在同一群的數值進行加總（`.sum()`）。
- `groupby()` 中的 `"commodity"` 實際上是沒有必要的，因為 `dfSheeps` 在 `"commodity"` 只存在 `"Sheep, live"` 這個數值。
- 查看 `dfSheeps.groupby(["year","flow"])["weight_kg"].sum().head()` 的輸出能更好的理解分群邏輯。

```
year   flow
1988  Export      420020534.0
      Import      12930779.0
1989  Export      469372796.0
      Import      44413812.0
      Re-Export    1020500.0
Name: weight_kg, dtype: float64
```

接下來就可以使用上一步得到的結果，繪製出每年活羊、活山羊的全球進口公斤數折線圖。

```
f, ax = plt.subplots(1, 1)
dfgr = pd.concat([dfSheepsGrouped,dfGoatsGrouped])
ax = sns.pointplot(ax=ax,x="year",y="weight_kg",data=dfgr[dfgr['flow']=='Import'],hue='commodity')
_ = ax.set_title('Global imports of kgs by animal')
```



觀察上圖可以明顯看出：

- 每年全球對於活羊的進口需求遠大於活山羊。
- 活羊的進口需求重量浮動較大，而活山羊的進口需求重量較為平穩。

2.2.6 Analysis of sheep



由於活山羊的分析步驟與活羊相同，只在此演示活羊的數據分析

此步驟分析不同國家的活羊進口、出口重量，以得知最大生產國與消費國的前幾名排名。比起之前的資料選取，少選了 `year`，但多選了 `country_or_area`。

```
dfSheepsGrouped = pd.DataFrame({'weight_kg' : dfSheeps.groupby( ["country_or_area","flow","commodity"] )["weight_kg"].sum()}).reset_index()
dfSheepsGrouped.head(3)
```

	country_or_area	flow	commodity	weight_kg
0	Afghanistan	Export	Sheep, live	2339.0
1	Albania	Export	Sheep, live	5659.0
2	Albania	Import	Sheep, live	336981.0

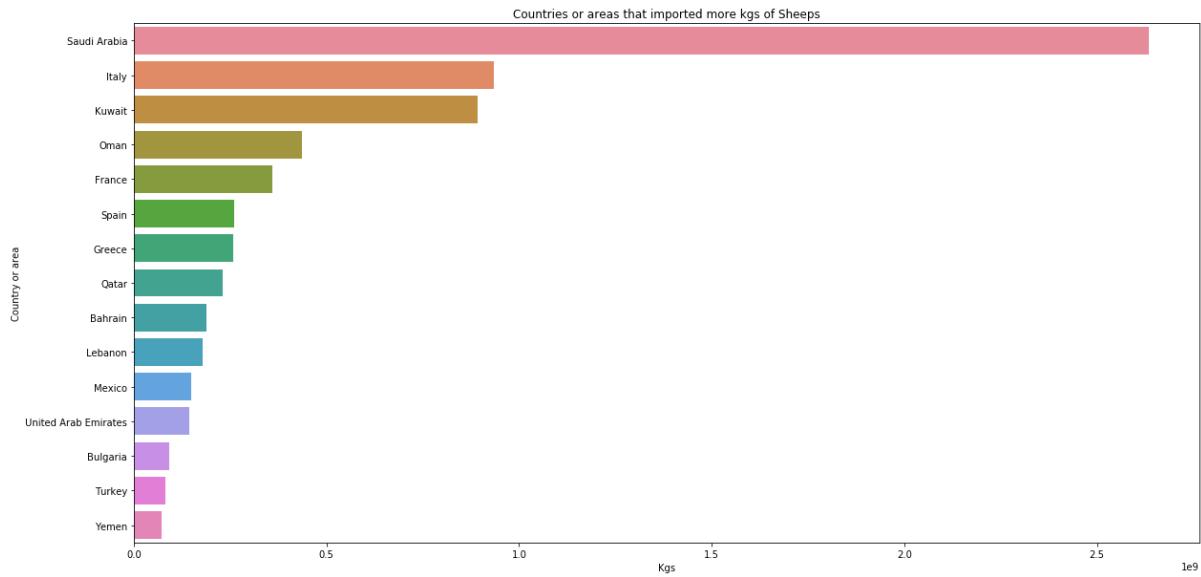
將上一步得到的資料選取，分別製作成進口與出口兩個表格。

```
sheepsImportsCountry = dfSheepsGrouped[dfSheepsGrouped['flow']=='Import']
sheepsExportsCountry = dfSheepsGrouped[dfSheepsGrouped['flow']=='Export']
sheepsImportsCountry.head(3)
```

	country_or_area	flow	commodity	weight_kg
2	Albania	Import	Sheep, live	336981.0
4	Algeria	Import	Sheep, live	9292358.0
6	Andorra	Import	Sheep, live	2067757.0

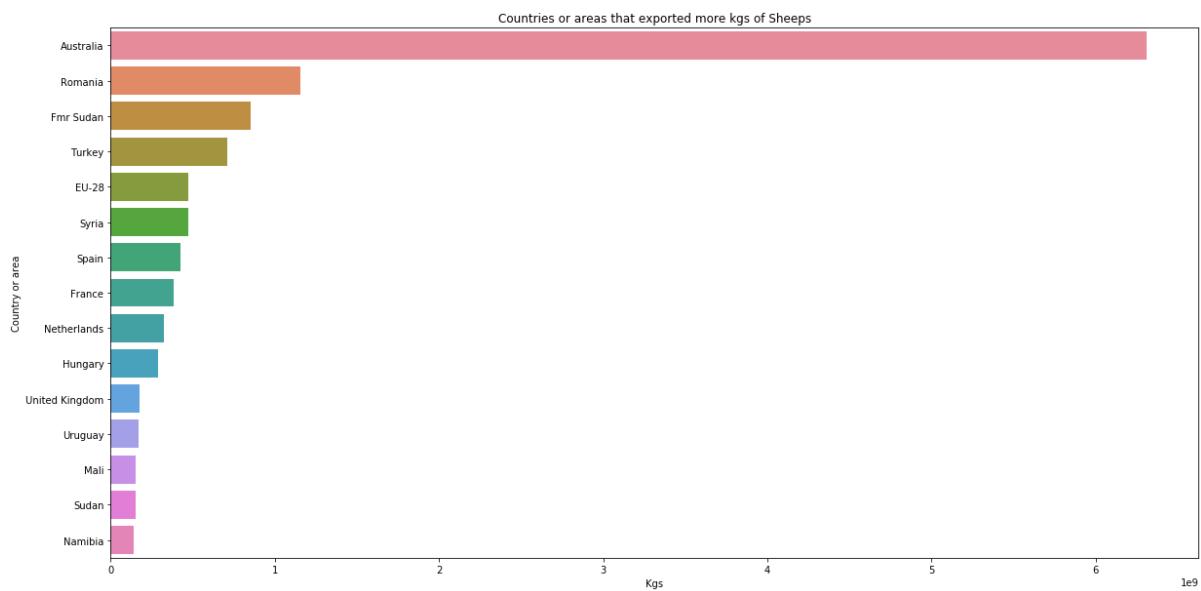
此步驟將剛剛得到的兩個表格，繪製成前十五名進口、出口重量最多國家的長條圖。

```
ax = sns.barplot(x="weight_kg", y="country_or_area", data=sheepsImportsCountry.sort_values('weight_kg', ascending=False)[:15])
_ = ax.set(xlabel='Kgs', ylabel='Country or area', title = "Countries or areas that imported more kgs of Sheeps")
```



從上圖中可以明顯看出，進口羊最多公斤的國家依序是沙特阿拉伯、意大利和科威特。

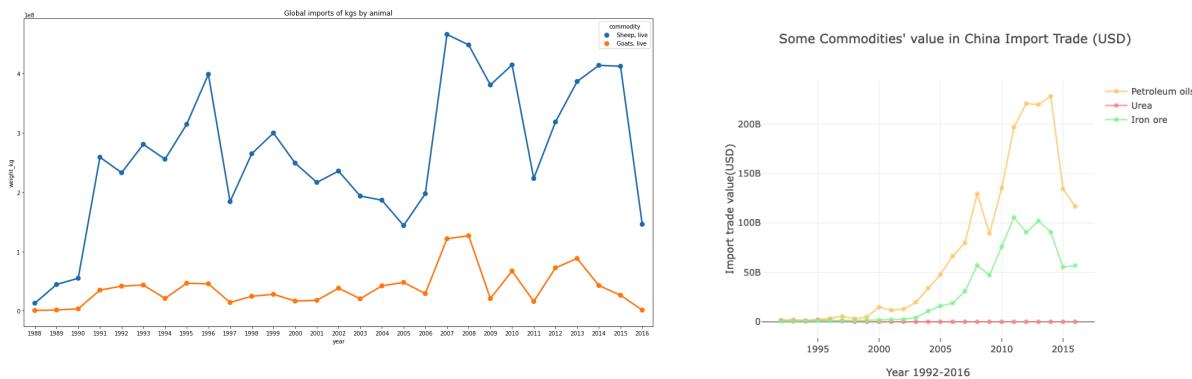
```
ax = sns.barplot(x="weight_kg", y="country_or_area", data=sheepsExportsCountry.sort_values('weight_kg', ascending=False)[:15])
_ = ax.set(xlabel='Kgs', ylabel='Country or area', title = "Countries or areas that exported more kgs of Sheeps")
```



從上圖中可以明顯看出，出口羊較多公斤的國家依序是澳大利亞、羅馬尼亞和蘇丹。

2.3 Notebook 1 與 Notebook 2 方法比較

- 分別採用 Kg 和 USD 作為分析商品項目在歷年進出口的單位



- 我認為 Notebook 2 採用 Kg 作為單位的可能理由：
 - 在某樣定物品在不同國家間價差很大時，用於比較多國家的進出口時使用 Kg 作為單位，對於進出口量的掌握會比金額來的準確。
 - 在長時間（橫跨多年）的分析上，由於會需要考量到物價的通貨膨脹，所以如過在乎的是每年對某品項的需求量變化時，就更適合使用 Kg 作為單位。（像是某些時候在乎有多少羊肉可以吃更勝其市場定價）
- 我認為 Notebook 1 採用 USD 作為單位的可能理由：
 - 原始資料欄位 weight_kg 有 480040 比缺失值，但是否造成影響其實需要近一步分析。（480040 占總數的比例少到不造成影響？缺失值集中的部分不在分析的對象中？）
 - 主要想分析的是進出口在金融方面上的議題。

2. Notebook 2 需要處理缺失值而 Notebook 1 則不需要

country_or_area	0
year	0
comm_code	0
commodity	0
flow	0
trade_usd	0
weight_kg	480040
quantity_name	505176
quantity	532710
category	0
dtype: int64	

- 從使用 `df.isnull().sum()` 顯示的缺失值來看，Notebook 2 採用 `weight_kg` 所以需要處理缺失值，而 Notebook 1 使用 `trade_usd` 則不需要處理缺失值。

- 我覺得 Notebook 2 比起使用 `df.dropna(how='any')` 把所有有缺失值的 row data 刪去，應該有更好的作法例如：
 - 改採用 `df.dropna(subset=['weight_kg'])` 達到只把缺少重量缺失值的 row data 刪去。
 - 如果是要針對特定商品分析，先檢查 `'weight_kg'` 在該品項的缺失值個數；如果是要針對每年的進出口分析，先檢查包含缺失值的年份 `'year'`，也許就能採用無缺失值的年份範圍做分析。
 - 比起直接把缺失值得 row data 刪去，可以先使用 `df.fillna()` 的方法以平均數、中位數、眾數等數值填補缺失值，接著做圖比對只刪去 row data 的趨勢，如果趨勢相近，也許就能採用填補數據的方式，減小缺失值對總數上的影響。

3. Notebook 1 有採用 dataset 以外的資訊來解釋數據趨勢

Notebook 1 分析進出口金額隨年份變化時，有採用「2001中國加入世貿組織」、「2008 年金融危機」、「農業向工業化的轉型」等外部資訊來解釋數據趨勢，比起 Notebook 1 就只有做出表格上的趨勢，不僅可以幫助評估數據的合理性，更能使圖表推論更具說服力。

4. Notebook 1 與 Notebook 2 的分析目標不同

兩個 Notebook 各自的分析目標在前面方法分析中有描述。

2.4 資料分析價值與可能產出

2.4.1 現有分析方法的產出

現有方法 Notebook 1 有五樣主要產出：

- 可以讓我們從做圖的方式了解一個國家（例子中採用中國）每年的進口、出口貿易金額的變化。還可以用來比對過往事件所帶來的影響（例子中採用2001中國加入世貿組織、2008 金融危機等）。

- 可以讓我們從圖表中了解到特定國家（例子中採用中國）進出口金額占比最多的商品項目。另外若是進一步將此圖表依不同年份分開繪製，就能更了解該國家主要進口、出口項目隨時代的變化。
- 將數樣特定商品每年的進出口金額繪製在同一圖表，可以透過金額變化趨勢相似與否來猜測兩樣商品間是否存在關聯性（例子中採用石油與鐵礦），也可以從金額變化看出該產品的需求量變化。
- 將定國家（例子中採用中國）與全球（所有國家）每年的進出口金額繪製成長條圖，可以看出全球進出口金額變化趨勢與特定國家每年金額的大概占比。若是更進一步將特定國家每年進口、出口金額占全球的百分比繪製成長條圖，就能更好的了解到該國家在全球貿易重要性的成長與衰退。
- 選定特定年份（例子中採用 1992 與 2016 年），將幾個主要國家的進口、出口金額繪製成圓餅圖，就能更好的比較這幾個國家該年在全球貿易中的重要性。

現有方法 Notebook 2 有兩樣主要產出：

- 可以讓我們從原始資料中提取出特定商品（例子中採用活羊、活山羊）的進出口公斤數，並依照不同的年份來做圖。最終透過圖表得知該特定商品歷年的進口、出口需求量（公斤重）。
- 可以讓我們做出不同國家針對特定商品的進出口量統計，繪圖後就可以明顯看出該產品出口量或進口量前幾名的國家是哪些。有助於了解該產品的供貨來源與主要消費國家。

2.4.1 其他分析可能的產出與價值

我想的鮭魚分析例子

現在受到俄烏戰爭的影響，導致台灣的鮭魚進口大量缺貨。但是常聽到的有「挪威鮭魚」、「智利鮭魚」、「阿拉斯加鮭魚」等，從來沒聽過有烏克蘭或俄羅斯進口的鮭魚啊！那為什麼俄烏戰爭會影響到台灣的鮭魚供應？這就值得我們來分析一下全球的鮭魚進口、出口國家以及數量與價格。

可能從此資料集分析的項目：

- 近年來出口鮭魚的主要國家與出口重量、金額（採 Notebook 1 類似方法），搭配像是右圖（示意圖）的貿易路線進行分析推論。可以用來找許替代的進口國家。
- 近年台灣對鮭魚需要的進口量變化。（目前數據中找不到 Taiwan，也許可以用其他鄰近島國代替作為預估值）
- 不同國家出口鮭魚每公斤的價格，用來找尋進口成本低的國家，但可能要加入不同的關稅計算。



上圖：trade routes from the Black Sea region to Central Asia and the South Caucasus

3. My insight

這一年來發生了不少對全球貿易影響巨大的事件，例如之前的長賜輪受困蘇伊士運河，還有尚未結束的烏俄戰爭以及對俄羅斯的經濟制裁。這些事件對全球貿易的影響進而使你我的民生用品價格上漲或是直接缺貨，如果能更了解全球的貿易數據，或許能得知某國家無法出口某品項時，應該改為從哪些其他供貨大國進口、了解從哪個國家供貨較為便宜，又或者是用在股

票期貨、供應鏈的判斷上。由此可見分析全球貿易數據能產出許多價值，我在上面也另外舉了一個自己覺得有趣的鮭魚例子。

對於此全球貿易的數據集，我覺得如果能在時間紀錄上擁有更細的顆粒度，例如以每月份為單位而不是年份，就能掌握一些像是季節性商品的資訊。還有如果能知道商品是出口到哪個國家或是從哪個國家進口，就能更準確的掌握商品的貿易流向。以上兩點單靠現在的資料集無法做到，實屬可惜。

透過分析 Notebook1 與 Notebook2，我基礎掌握了簡單的表格資料選取與製圖，也提出了一些改進方法，但我認為接下來更難的挑戰會是如何將國際貿易與商品交易的專業知識運用在分析上，如果無法達到就很難知道有什麼其他有用的資料可以加入分析當中。

從實際的數據做分析，絕對讓你比起課本中「某某國盛產某某項目，占GDP多少 %」的描述更理解全球的貿易，必定獲益良多。