

# import\_export

---

This package is designed to support companies from no matter what countries to make informed decision pertaining to their cross boarder trade activities during and after the pandemic.

## Tutorial

An introduction of the package is given in this video

(<https://drive.google.com/drive/folders/1oBvMWMxe7AcZn7EdZcCIA3LyFb59gjV6?usp=sharing>). You can follow through to know more.

## Set up

It is better to create a virtual environment for this project since it will not affect your existing python environment.

### Step 0: Download ChromeDriver

Go to <https://chromedriver.chromium.org/> and download the suitable version chromedriver for your computer. And place it in the home directory of this repo after step 1.

### Step 1: Clone the repository and get the source code ready in your computer

```
# clone repo
git clone https://github.com/charleschan1108/import_export.git

# change directory
cd import_export
```

### Step 2: Create a virtual environment named env

```
python -m venv env
```

### Step 3: Activate the env virtual environment

```
source env/bin/activate
```

### Step 4: Install packages

```
pip install selenium
pip install pandas
pip install numpy
```

```
pip install beautifulsoup4
pip install requests
pip install matplotlib
```

## Manuals

Currently, the library support businesses to answer two questions (2 applications):

1. For a specific country, which country is the best partner for export/import during and after the pandemic?
2. In general, which country is the best partner for trade of goods and services during and after the pandemic?

The package will be based on different types of data:

- Covid data (death, case and vaccination)
- Macroeconomics data (gdp, unemployment rate, inflation rate, value of export/import/net trade in goods)
- Countries trade profile

to compute a rank for more than 230 countries, so that businesses can use the rank as a reference to choose the best country to look for trading partners.

## Application 1: For specified country

```
# With data being refreshed
# it takes around 3 mins to refresh the data
python main.py --refresh 1 --country China

# Without data being refreshed
python main.py --refresh 0 --country China
```

Sample output:

```
For China, the best export partners in short term (quarterly data):
      Top exported products
QuarterlyChange_people_fully_vaccinatedRank ...  GdpTradeCompRank
Overall Rank
Japan                                     NaN
113.0 ...                               43                      1
South Korea                             []
71.0 ...                               18                      2
United States                           []
189.0 ...                               47                      3

[3 rows x 18 columns]
Check 2021-12-01T00:36:13.910891_export.csv for more details.
For China, the best import partners in short term (quarterly data):
```

```

Top imported products
QuarterlyChange_people_fully_vaccinatedRank ... GdpTradeCompRank Overall Rank
Japan NaN
113.0 ... 43 1
South Korea ['Mineral fuels and oils', 'Electrical machine...
71.0 ... 18 2
United States ['Electrical machinery and equipment', 'Machin...
189.0 ... 47 3

```

[3 rows x 18 columns]

Check 2021-12-01T00:36:13.910891\_import.csv for more details.

For China, the best export partners in long term (yearly data):

```

Top exported products
YearlyChange_people_fully_vaccinatedRank ... GdpTradeCompRank Overall Rank
Japan NaN
46.0 ... 43 1
South Korea []
122.0 ... 18 2
United States []
227.0 ... 47 3

```

[3 rows x 18 columns]

Check 2021-12-01T00:36:13.939796\_export.csv for more details.

For China, the best import partners in long term (yearly data):

```

Top imported products
YearlyChange_people_fully_vaccinatedRank ... GdpTradeCompRank Overall Rank
Japan NaN
46.0 ... 43 1
South Korea ['Mineral fuels and oils', 'Electrical machine...
122.0 ... 18 2
United States ['Electrical machinery and equipment', 'Machin...
227.0 ... 47 3

```

[3 rows x 18 columns]

Check 2021-12-01T00:36:13.939796\_import.csv for more details.

## Application 2: For any country

```
python main.py --refresh 0
```

Sample output:

```

In general, the best trade partners in short term (quarterly data) are
Top exported products
Top imported products ... GdpTradeCompRank Overall Rank
Czech Republic NaN

```

NaN ...	19	1	
Japan			NaN
NaN ...	43	2	
Saudi Arabia			NaN
NaN ...	5	3	
Indonesia	['Mineral fuels & oils', 'Animal/vegetable fat...]		
	['Mineral fuels and oils', 'Machinery and mech...]		14
4			
Colombia	['Mineral oils and fuels', 'Coffee, tea, mate,...]		
	['Machinery & Mechanical Appliances', 'Electri...]		36
5			
South Korea			[]
	['Mineral fuels and oils', 'Electrical machine...]		18
6			
Chile	['Ores, slag and ash', 'Copper and articles th...]		
	['Mineral fuels and oils', 'Vehicles', 'Machin...]		20
7			
Sweden			NaN
NaN ...	21	8	
Luxembourg			NaN
NaN ...	33	9	
Italy			NaN
NaN ...	11	10	

[10 rows x 19 columns]

Check 2021-12-01T00:38:51.211457\_general\_short term (quarterly data).csv for more details.

In general, the best trade partners in long term (yearly data) are

			Top exported products	
Top imported products	...	GdpTradeCompRank	Overall Rank	
Japan				NaN
NaN ...	43	1		
Indonesia	['Mineral fuels & oils', 'Animal/vegetable fat...]			
	['Mineral fuels and oils', 'Machinery and mech...]			14
2				
Italy				NaN
NaN ...	11	3		
France				NaN
NaN ...	44	4		
Czech Republic				NaN
NaN ...	19	5		
Sweden				NaN
NaN ...	21	6		
Saudi Arabia				NaN
NaN ...	5	7		
Chile	['Ores, slag and ash', 'Copper and articles th...]			
	['Mineral fuels and oils', 'Vehicles', 'Machin...]			20
8				
Portugal				NaN
NaN ...	37	9		
Spain				NaN
NaN ...	38	10		

```
[10 rows x 19 columns]
Check 2021-12-01T00:38:51.232216_general_long term (yearly data).csv for
more details.
```

## Additional functions

### Weighting

The above example assume user put equal weight on factors:

- number of covid case
- number of death by covid
- vaccinated rate (people\_fully\_vacinated / population)
- GDP growth
- Inflation rate
- Value of export/import/net trade in goods
- Net trade / GDP
- Unemployment rate

Users can specify their weighting in config.py:

```
# config.py

weighting = {
    "vaccinated_rateRank": 0.2,
    "casesRank": 0.2,
    "deathsRank": 0.2,
    "GdpChangeRank": 0.1,
    "InflationRank": 0.1,
    "IMPRank": 0.3,
    "EXPRank": 0.3,
    "NTRADERank": 0.3,
    "UnemploymentRateChangeRank": 0.1
}
```

The module will normalize the weighting (i.e. such that the weightings sum to one) and plugged into the calculation of overall rank in the final output.

To enable weighting, the command becomes:

```
python main.py --refresh 0 --equal_weight 0
```

### TopN

Users can specify the argument topn to control the number of output to display in console.

Example snippet:

```
python main.py --refresh 0 --equal_weight 0 --topn 10
```

## Others

For more details, you can use the following command:

```
python main.py -h
```

## Package Logic

This package consists of 3 parts:

1. Data ingestion to crawl raw data
2. Data process to massage data
3. Application

### Data ingestion

All the crawlers are stored in the `data_ingestion` module. They will crawl the designated data from the specified websites and stored into the data directory.

### Data process

All the data processors are stored in the `data_process` module. They will process the raw data according to the logic specified and save the output at the insights directory.

### Application

Provide users with commands to interact with the insights uncovered in the previous part.

## Authors

For more details, please contact

- Charles Chan [chunyiuc@andrew.cmu.edu](mailto:chunyiuc@andrew.cmu.edu)
- Hsueh-i Lu [hsuehil@andrew.cmu.edu](mailto:hsuehil@andrew.cmu.edu)
- Jiaqi Song [jiaqis@andrew.cmu.edu](mailto:jiaqis@andrew.cmu.edu)
- Rui Pan [ruipan@andrew.cmu.edu](mailto:ruipan@andrew.cmu.edu)
- Yaheng Wang [yahengw@andrew.cmu.edu](mailto:yahengw@andrew.cmu.edu)
- Yigang Zhou [yigangz@andrew.cmu.edu](mailto:yigangz@andrew.cmu.edu)