

## Geo data生成手順と入カマスタ・ファイル (1/5)

### Geo data生成手順 (1)

#### 入カマスタ・ファイル

1. node\_city\_name.csv
2. city\_geo\_data.csv

処理: PySIのnoode\_nameを地図座標データとマージする

merge\_df\_node\_city\_geo.py

#### 出力ファイル

node\_city\_geo.csv

node\_name, capital\_en

root,Saitama City  
JPN,Yokohama  
YTO,Toronto  
NYC,New York City  
LAX,Los Angeles  
MEX,Mexico City  
SAO,Sao Paulo  
BUE,Buenos Aires  
KUL,Kuala Lumpur

name\_ens,capital\_jp, capital\_en,iscapital,lat,lon

Belgium,ブリュセル,Brussels,1,50.8503,4.3517  
Switzerland,チューリッヒ,Zurich,0,47.3842,8.53185  
Portugal,リスボン,Lisbon,1,38.7222,-9.1393  
Poland,ワルシャワ,Warsaw ,1,52.22977,21.01178  
Netherlands,アムステルダム,Amsterdam,1,52.3915,4.9035  
Swedn,ストックホルム,Stockholm,1,59.3293,18.0685  
Swedn,ヨーテボリ,Göteborg,0,57.70716,11.96679  
Norway,オスロ,Oslo,1,59.9138,10.7522  
Denmark,コペンハーゲン,Kopenhagen,1,55.676,12.5683  
Finland,ヘルシンキ,Helsinki,1,60.1698,24.9383

node\_name,capital\_en,name\_ens,capital\_jp,iscapital,lat,lon

root,Saitama City,Japan,さいたま,0,35.861665,139.6454616  
JPN,Yokohama,Japan,横浜,0,35.4503614,139.6342277  
YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901  
NYC,New York City,United States,ニューヨーク,0,40.7127744,-74.006059  
LAX,Los Angeles,United States,ロサンゼルス,0,34.0537136,-118.2426533  
MEX,Mexico City,Mexico,メキシコシティ,1,19.431793,-99.1339996  
SAO,Sao Paulo,Brazil,サンパウロ,0,-23.5474385,-46.637397

```
df_node_geo = pd.merge(df_node_city_name, df_city_geo, on='capital_en')
```

```
df_node_geo.to_csv( filename , encoding="utf8", header=True , index=False)
```

## Geo data生成手順と入カマスタ・ファイル (2/5)

### Geo data生成手順(2)

#### 入カマスタ・ファイル

1. node\_city\_geo.csv
2. common\_plan\_unit.csv

#### 処理:

前処理で生成されたPySIのnode\_nameと地図座標データのマスタを使って、CPUの出発地のノード名Dpt\_node(サプライチェーンの供給元の事業所)から地図座標を検索、付加する

merge\_df\_CPU\_geo.py

#### 出力ファイル

CPU\_geo\_df.csv

```
node_name,capital_en,name_ens,capital_jp,iscapital,lat,lon
root,Saitama City,Japan,さいたま,0,35.861665,139.6454616
JPN,Yokohama,Japan,横浜,0,35.4503614,139.6342277
YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901
NYC,New York City,United States,ニューヨーク,0,40.7127744,-74.006059
LAX,Los Angeles,United States,ロサンゼルス,0,34.0537136,-118.2426533
MEX,Mexico City,Mexico,メキシコシティ,1,19.431793,-99.1339996
SAO,Sao Paulo,Brazil,サンパウロ,0,-23.5474385,-46.637397
```

```
seq_no,control_flag,priority_no,modal,LT,Dpt_entity,Dpt_week,Dpt_step,Arv_entity,Arv_week,Arv_step
YTOLEAF202301009,F,001202301009,B,2,YTO,-1,0,YTOLEAF,1,0
YTOLEAF202301014,F,001202301014,B,2,YTO,-1,1,YTOLEAF,1,1
YTOLEAF202301016,F,001202301016,B,2,YTO,-1,2,YTOLEAF,1,2
YTOLEAF202301019,F,001202301019,B,2,YTO,-1,3,YTOLEAF,1,3
YTOLEAF202301022,F,001202301022,B,2,YTO,-1,4,YTOLEAF,1,4
```

```
seq_no,control_flag,priority_no,modal,LT,Dpt_entity,Dpt_week,Dpt_step,Arv_entity,Arv_week,Arv_step,node_name,capital_e
n,name_ens,capital_jp,iscapital,lat,lon
YTOLEAF202301009,F,1202301009,B,2,YTO,-1,0,YTOLEAF,1,0,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901
YTOLEAF202301014,F,1202301014,B,2,YTO,-1,1,YTOLEAF,1,1,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901
YTOLEAF202301016,F,1202301016,B,2,YTO,-1,2,YTOLEAF,1,2,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901
```

```
df_CPU_geo = pd.merge(df_CPU, df_city_geo, left_on='Dpt_entity', right_on='node_name')
```

```
df_CPU_geo.to_csv( filename , encoding="utf8", header=True, index=False)
```

## Geo data生成手順と入カマスタ・ファイル (3/5)

### Geo data生成手順 (3)

#### 入カマスタ・ファイル

1. node\_city\_geo. csv
2. CPU\_geo\_df. csv

#### 処理:

前処理で生成されたCPU(出荷地のノード名Dpt\_entity)と地図座標データのマスタを使って、CPUの着荷地のノード名Arv\_entity(サプライチェーンの着荷事業所)の頭3桁から地図座標を検索し、座標データを付加する  
着荷地nodeが最終消費、LEAF\_node(=4桁以上)の場合、  
(4桁以降の記号\_D/\_I/\_N/LEAF) 販売チャネル種別(直販、間接販売、WEB、総合)に対応する、  
地図座標(それぞれ東-西-南-北の方角に、距離500m)を仮に配置

#### merge\_df\_CPU\_Arv\_geo. py

#### 出力ファイル

#### CPU\_Arv\_geo\_df. csv

ヘッダーは、入力ファイルのCPUと同じものを使う

```
node_name,capital_en,name_ens,capital_jp,iscapital,lat,lon
root,Saitama City,Japan,さいたま,0,35.861665,139.6454616
JPN,Yokohama,Japan,横浜,0,35.4503614,139.6342277
YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901
NYC,New York City,United States,ニューヨーク,0,40.7127744,-74.006059
LAX,Los Angeles,United States,ロサンゼルス,0,34.0537136,-118.2426533
MEX,Mexico City,Mexico,メキシコシティ,1,19.431793,-99.1339996
SAO,Sao Paulo,Brazil,サンパウロ,0,-23.5474385,-46.637397
```

```
seq_no,control_flag,priority_no,modal,LT,Dpt_entity,Dpt_week,Dpt_step,Arv_entity,Arv_week,Arv_step,node_name,capital_e
n,name_ens,capital_jp,iscapital,lat,lon
YTOLEAF202301009,F,1202301009,B,2,YTO,-1,0,YTOLEAF,1,0,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901
YTOLEAF202301014,F,1202301014,B,2,YTO,-1,1,YTOLEAF,1,1,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901
YTOLEAF202301016,F,1202301016,B,2,YTO,-1,2,YTOLEAF,1,2,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901
```

```
seq_no,control_flag,priority_no,modal,LT,Dpt_entity,Dpt_week,Dpt_step,Arv_entity,Arv_week,Arv_step,node_name,capital_en,name_ens,ca
pital_jp,iscapital,lat,lon,node_name_Arv,capital_en_Arv,name_ens_Arv,capital_jp_Arv,iscapital_Arv,lat_Arv,lon_Arv
YTOLEAF202301009,F,1202301009,B,2,YTO,-1,0,YTOLEAF,1,0,YTO,Toronto,Canada,トロント,0,43.6534399,-
79.3840901,YTO,Toronto,Canada,トロント,0,43.65794658648729,-79.3840901
YTOLEAF202301014,F,1202301014,B,2,YTO,-1,1,YTOLEAF,1,1,YTO,Toronto,Canada,トロント,0,43.6534399,-
79.3840901,YTO,Toronto,Canada,トロント,0,43.66245327297458,-79.3840901
YTOLEAF202301016,F,1202301016,B,2,YTO,-1,2,YTOLEAF,1,2,YTO,Toronto,Canada,トロント,0,43.6534399,-
79.3840901,YTO,Toronto,Canada,トロント,0,43.66695995946187,-79.3840901
```

```
df_CPU_geo = pd.merge(df_CPU, df_city_geo, left_on='Dpt_entity', right_on='node_name')
```

```
df_CPU_geo.to_csv( filename , encoding="utf8", header=True , index=False)
```

## Geo data生成手順と入カマスタ・ファイル (4/5)

### Geo data生成手順(4)

### 入カマスタ・ファイル

CPU\_Arv\_geo\_df.csv

```
seq_no,control_flag,priority_no,modal,LT,Dpt_entity,Dpt_week,Dpt_step,Arv_entity,Arv_week,Arv_step,node_name,capital_en,name_ens,ca
pital_jp,iscapital,lat,lon,node_name_Arv,capital_en_Arv,name_ens_Arv,capital_jp_Arv,iscapital_Arv,lat_Arv,lon_Arv
YTOLEAF202301009,F,1202301009,B,2,YTO,-1,0,YTOLEAF,1,0,YTO,Toronto,Canada,トロント,0,43.6534399,-
79.3840901,YTO,Toronto,Canada,トロント,0,43.65794658648729,-79.3840901
YTOLEAF202301014,F,1202301014,B,2,YTO,-1,1,YTOLEAF,1,1,YTO,Toronto,Canada,トロント,0,43.6534399,-
79.3840901,YTO,Toronto,Canada,トロント,0,43.66245327297458,-79.3840901
YTOLEAF202301016,F,1202301016,B,2,YTO,-1,2,YTOLEAF,1,2,YTO,Toronto,Canada,トロント,0,43.6534399,-
79.3840901,YTO,Toronto,Canada,トロント,0,43.66695995946187,-79.3840901
```

処理:

1 週間を5分割して移動するアニメーション用のデータ生成

**Dpt\_day,lat\_day,lon\_dayを追加する**

make\_CPU\_days\_trans\_geo010\_all.py

```
import make_Arv_by_Dpt_and_Distance_Azimuth
```

```
import make_Distance_Azimuth_by_Dpt2Arv
```

### 出力ファイル

1 週間を5分割したアニメーション用のデータ

CPU\_days\_trans\_geo

```
seq_no,control_flag,priority_no,modal,LT,Dpt_entity,Dpt_week,Dpt_step,Arv_entity,Arv_week,Arv_step,node_name,capital_en,name_ens,capital
_jp,iscapital,lat,lon,node_name_Arv,capital_en_Arv,name_ens_Arv,capital_jp_Arv,iscapital_Arv,lat_Arv,lon_Arv,Dpt_day,lat_day,lon_day
YTOLEAF202301009,F,1202301009,B,2,YTO,-1,0,YTOLEAF,1,0,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901,YTO,Toronto,Canada,
トロント,0,43.65794658648729,-79.3840901,-1,0,43.6534399,-79.3840901
YTOLEAF202301009,F,1202301009,B,2,YTO,-1,0,YTOLEAF,1,0,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901,YTO,Toronto,Canada,
トロント,0,43.65794658648729,-79.3840901,-0.8,43.65389056902886,-79.3840901
YTOLEAF202301009,F,1202301009,B,2,YTO,-1,0,YTOLEAF,1,0,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901,YTO,Toronto,Canada,
トロント,0,43.65794658648729,-79.3840901,-0.6,43.654341238022035,-79.3840901
```

## Geo data生成手順と入カマスタ・ファイル (5/5)

### Geo data生成手順 (5)

#### 入カマスタ・ファイル

1 週間を5分割したアニメーション用のデータ

CPU\_days\_trans\_geo

```
seq_no,control_flag,priority_no,modal,LT,Dpt_entity,Dpt_week,Dpt_step,Arv_entity,Arv_week,Arv_step,node_name,capital_en,name_ens,capital
_jp,iscapital,lat,lon,node_name_Arv,capital_en_Arv,name_ens_Arv,capital_jp_Arv,iscapital_Arv,lat_Arv,lon_Arv,Dpt_day,lat_day,lon_day
YTOLEAF202301009,F,1202301009,B,2,YTO,-1,0,YTOLEAF,1,0,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901,YTO,Toronto,Canada,
トロント,0,43.65794658648729,-79.3840901,-1.0,43.6534399,-79.3840901
YTOLEAF202301009,F,1202301009,B,2,YTO,-1,0,YTOLEAF,1,0,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901,YTO,Toronto,Canada,
トロント,0,43.65794658648729,-79.3840901,-0.8,43.65389056902886,-79.3840901
YTOLEAF202301009,F,1202301009,B,2,YTO,-1,0,YTOLEAF,1,0,YTO,Toronto,Canada,トロント,0,43.6534399,-79.3840901,YTO,Toronto,Canada,
トロント,0,43.65794658648729,-79.3840901,-0.6,43.654341238022035,-79.3840901
```

処理:

1 週間を5分割して移動するアニメーションを表示

geo\_animation\_test4\_CPU\_days\_trans\_sort.py

表示前に、入力データをソートする

```
df_sort = df.sort_values(by=["Arv_entity", "Dpt_day", "Dpt_entity", ])
```

表示方法のオプション指定

```
projection="orthographic")
#projection="equiarectangular")
#projection="natural earth")
```

出力ファイル

xxxxx.Html 地図とサプライチェーンの移動アニメーション

