In [15]:
```
# COVID-19 South Korea Propagation Scheme Visualization Code in Python
# By Kyunghyun Ryu and Seongwon Yoon

import sys
import os

import numpy as np
import numba as nb
import pandas as pd
import seaborn as sb
import datetime
```

In [16]:
```
df_region_info = pd.read_csv('/home/tonyyoon/week5/Region.csv')
```

In [17]:
```
def add_lat_long_to_df(df_target, df_region):
    # Assume that there is only city name in the df_target file but no latitude or longitude
    # This function adds latitude and longitude from df_region to df_target

    # Add latitude and longitude to df_target DataFrame
    df_target['latitude'] = 0.0
    df_target['longitude'] = 0.0

    for row in df_target.itertuples():
        # Extract latitude and longitude of specific city
        # df_target 'province' = df_region 'city'
        df_region_info_tmp = df_region[df_region['city'].isin([row.province])]
        df_region_info_tmp = df_region_info_tmp.loc[:, ['latitude']]
        df_target.loc[row.Index, ['latitude']] = df_region_info_tmp.iat[0,0]

        df_region_info_tmp = df_region[df_region['city'].isin([row.province])]
        df_region_info_tmp = df_region_info_tmp.loc[:, ['longitude']]
        df_target.loc[row.Index, ['longitude']] = df_region_info_tmp.iat[0,0]
```

In [18]:
```
df_simulation_results = pd.read_csv('/home/tonyyoon/week5/covidproject/odeData/simulationresults.csv')
```

In [19]:
```
%time _=add_lat_long_to_df(df_simulation_results, df_region_info)
```

```
CPU times: user 13.2 s, sys: 10.2 ms, total: 13.2 s
Wall time: 13.2 s
```

In [20]:
```
@nb.jit
def add_lat_long_to_df_2(df_target, df_region):
    # Assume that there is only city name in the df_target file but no latitude or longitude
    # This function adds latitude and longitude from df_region to df_target

    # Add latitude and longitude to df_target DataFrame
    df_target['latitude'] = 0.0
    df_target['longitude'] = 0.0

    for row in df_target.itertuples():
        # Extract latitude and longitude of specific city
        # df_target 'province' = df_region 'city'
        df_region_info_tmp = df_region[df_region['city'].isin([row.province])]
        df_region_info_tmp = df_region_info_tmp.loc[:, ['latitude']]
        df_target.loc[row.Index, ['latitude']] = df_region_info_tmp.iat[0,0]

        df_region_info_tmp = df_region[df_region['city'].isin([row.province])]
        df_region_info_tmp = df_region_info_tmp.loc[:, ['longitude']]
        df_target.loc[row.Index, ['longitude']] = df_region_info_tmp.iat[0,0]
```

In [21]:
```
%time _=add_lat_long_to_df_2(df_simulation_results, df_region_info)
```

```
<ipython-input-20-847431e6a81e>:1: NumbaWarning:
Compilation is falling back to object mode WITH looplifting enabled because Function "add_lat_long_to_df_2" failed type inference due to: non-precise type pyobject
[1] During: typing of argument at <ipython-input-20-847431e6a81e> (7)

File "<ipython-input-20-847431e6a81e>", line 7:
def add_lat_long_to_df_2(df_target, df_region):
    <source elided>
    # Add latitude and longitude to df_target DataFrame
    df_target['latitude'] = 0.0
    ^

  @nb.jit
<ipython-input-20-847431e6a81e>:1: NumbaWarning:
Compilation is falling back to object mode WITHOUT looplifting enabled because Function "add_lat_long_to_df_2" failed type inference due to: cannot determine Numba type of <class 'numba.dispatcher.LiftedLoop'>

File "<ipython-input-20-847431e6a81e>", line 10:
def add_lat_long_to_df_2(df_target, df_region):
    <source elided>

    for row in df_target.itertuples():
    ^

  @nb.jit
/opt/ohpc/pub/compiler/intel/intelpython3/lib/python3.7/site-packages/numba/object_mode_passes.py:178: NumbaWarning: Function "add_lat_long_to_df_2" was compiled in object mode without forceobj=True, but has lifted loops.

File "<ipython-input-20-847431e6a81e>", line 7:
def add_lat_long_to_df_2(df_target, df_region):
    <source elided>
    # Add latitude and longitude to df_target DataFrame
    df_target['latitude'] = 0.0
    ^

  state.func_ir.loc))
/opt/ohpc/pub/compiler/intel/intelpython3/lib/python3.7/site-packages/numba/object_mode_passes.py:188: NumbaDeprecationWarning:
Fall-back from the nopython compilation path to the object mode compilation path has been detected, this is deprecated behaviour.

For more information visit http://numba.pydata.org/numba-doc/latest/reference/deprecation.html#deprecation-of-object-mode-fall-back-behaviour-when-using-jit (http://numba.pydata.org/numba-doc/latest/reference/deprecation.html#deprecation-of-object-mode-fall-back-behaviour-when-using-jit)

File "<ipython-input-20-847431e6a81e>", line 7:
def add_lat_long_to_df_2(df_target, df_region):
    <source elided>
    # Add latitude and longitude to df_target DataFrame
    df_target['latitude'] = 0.0
    ^

  state.func_ir.loc))
<ipython-input-20-847431e6a81e>:1: NumbaWarning:
Compilation is falling back to object mode WITHOUT looplifting enabled because Function "add_lat_long_to_df_2" failed type inference due to: non-precise type pyobject
[1] During: typing of argument at <ipython-input-20-847431e6a81e> (10)

File "<ipython-input-20-847431e6a81e>", line 10:
def add_lat_long_to_df_2(df_target, df_region):
    <source elided>

    for row in df_target.itertuples():
    ^

  @nb.jit
/opt/ohpc/pub/compiler/intel/intelpython3/lib/python3.7/site-packages/numba/object_mode_passes.py:178: NumbaWarning: Function "add_lat_long_to_df_2" was compiled in object mode without forceobj=True.

File "<ipython-input-20-847431e6a81e>", line 10:
def add_lat_long_to_df_2(df_target, df_region):
    <source elided>

    for row in df_target.itertuples():
    ^

  state.func_ir.loc))
/opt/ohpc/pub/compiler/intel/intelpython3/lib/python3.7/site-packages/numba/object_mode_passes.py:188: NumbaDeprecationWarning:
Fall-back from the nopython compilation path to the object mode compilation path has been detected, this is deprecated behaviour.

For more information visit http://numba.pydata.org/numba-doc/latest/reference/deprecation.html#deprecation-of-object-mode-fall-back-behaviour-when-using-jit (http://numba.pydata.org/numba-doc/latest/reference/deprecation.html#deprecation-of-object-mode-fall-back-behaviour-when-using-jit)

File "<ipython-input-20-847431e6a81e>", line 10:
def add_lat_long_to_df_2(df_target, df_region):
    <source elided>

    for row in df_target.itertuples():
    ^

  state.func_ir.loc))

CPU times: user 14.1 s, sys: 35.5 ms, total: 14.1 s
Wall time: 14.1 s
```

In [ ]: