

# 1-4

November 2, 2021

Problem 1-4

Assignment 1

Shuhan Xiao, Klaus Kades, Lucas-Raphael Müller, Melanie Schellenberg

```
[ ]: # Install a pip package in the current Jupyter kernel
! pip install numpy pandas python-igraph matplotlib pycairo cairocffi
```

```
Requirement already satisfied: numpy in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (1.21.3)
Requirement already satisfied: pandas in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (1.3.4)
Requirement already satisfied: python-igraph in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (0.9.7)
Requirement already satisfied: matplotlib in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (3.4.3)
Requirement already satisfied: pycairo in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (1.20.1)
Requirement already satisfied: cairocffi in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (1.3.0)
Requirement already satisfied: python-dateutil>=2.7.3 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
pandas) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
pandas) (2021.3)
Requirement already satisfied: texttable>=1.6.2 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
python-igraph) (1.6.4)
Requirement already satisfied: pyparsing>=2.2.1 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
matplotlib) (3.0.3)
Requirement already satisfied: pillow>=6.2.0 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
matplotlib) (8.3.2)
Requirement already satisfied: cycler>=0.10 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
matplotlib) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
```

```

/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
matplotlib) (1.3.2)
Requirement already satisfied: cffi>=1.1.0 in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
cairocffi) (1.14.6)
Requirement already satisfied: pycparser in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
cffi>=1.1.0->cairocffi) (2.20)
Requirement already satisfied: six in
/workplace/anaconda3/envs/complex_network/lib/python3.9/site-packages (from
cyclar>=0.10->matplotlib) (1.16.0)

```

```

[ ]: import pandas as pd
import igraph as ig
from igraph import *
import numpy as np

path = '/workplace/CNA/Complex-Network-Analysis-Exercises/assignment-1/
↳FAOSTAT_data_10-26-2021.csv'
data = pd.read_csv(path)
others_values=data[data['Partner Countries']== 'Others (adjustment)']
FAO_values = data[data['Partner Countries']== 'Total FAO']
Unspecified = data[data['Partner Countries']== 'Unspecified Area']
data = data.drop(others_values.index, axis=0)
data = data.drop(FAO_values.index, axis=0)
data = data.drop(Unspecified.index, axis=0)
df = data.fillna('NULL')
NULL_values = df[df['Flag']!= 'NULL']
df = df.drop(NULL_values.index, axis=0)
df=df.reset_index()

print(df.columns)
print(df.head())

#compare these features with the ones of the exercise session
print(df.shape)
print(df[df['Reporter Countries']=='United States of America'].Value.sum())
print(df[df['Partner Countries']=='United States of America'].Value.sum())
df_1 = df[['Reporter Countries', 'Partner Countries', 'Value']]

```

```

Index(['index', 'Domain Code', 'Domain', 'Reporter Country Code (FAO)',
      'Reporter Countries', 'Partner Country Code (FAO)', 'Partner Countries',
      'Element Code', 'Element', 'Item Code', 'Item', 'Year Code', 'Year',
      'Unit', 'Value', 'Flag', 'Flag Description'],
      dtype='object')

```

	index	Domain Code	Domain	Reporter Country Code (FAO)	\
0	0	FT	Forestry Trade Flows	2	
1	3	FT	Forestry Trade Flows	2	

2	4	FT	Forestry Trade Flows	3
3	5	FT	Forestry Trade Flows	3
4	6	FT	Forestry Trade Flows	3

	Reporter Countries	Partner Country Code (FAO)	Partner Countries \
0	Afghanistan	68	France
1	Afghanistan	165	Pakistan
2	Albania	11	Austria
3	Albania	33	Canada
4	Albania	68	France

	Element Code	Element	Item Code	Item \
0	5922	Export Value	1633	Sawnwood, non-coniferous all
1	5922	Export Value	1671	Newsprint
2	5922	Export Value	1633	Sawnwood, non-coniferous all
3	5922	Export Value	1619	Wood chips and particles
4	5922	Export Value	1632	Sawnwood, coniferous

	Year Code	Year	Unit	Value	Flag	Flag Description
0	2017	2017	1000 US\$	37	NULL	Official data
1	2017	2017	1000 US\$	2	NULL	Official data
2	2017	2017	1000 US\$	29	NULL	Official data
3	2017	2017	1000 US\$	0	NULL	Official data
4	2017	2017	1000 US\$	13	NULL	Official data

(15402, 17)

5047564

4949057

```
[ ]: def task1(df):
    g = ig.Graph.TupleList(df[["Reporter Countries", "Partner Countries"]].
        ↳itertuples(index=False), directed=True)
    g.es['weight'] = list(abs(df['Value']))
    print(g.summary())

    #test by comparing with pd dataframe
    for index in range(len(df)):
        v1, v2 = g.get_edgelist()[index]

        assert (g.vs['name'][v1]==df["Reporter Countries"][index]), "Oh no!_
        ↳'Reporter failed!"
        assert (g.vs['name'][v2]==df["Partner Countries"][index]), "Oh no!_
        ↳Partner failed!"
        assert (g.es['weight'][index]==df["Value"][index]), "Oh no! Value_
        ↳failed!"
    return g
```

```

def plot_task(g, save_name, margin):
    visual_style = {"vertex_size":15,
                    "vertex_label":g.vs()["name"],
                    "edge_width": [(np.log(value+1)+1) for value in g.
→es()['weight']],
                    "edge_label":g.es()['weight'],
                    "margin": margin}
    g.simplify(multiple=True, loops=True, combine_edges=dict(weight="sum"))

    layout = g.layout("kk")
    plot(g, save_name, **visual_style, layout=layout)

g = task1(df_1.iloc[:100,:])
plot_task(g, 'task1_first100.pdf', 20)

```

```

IGRAPH DNW- 46 100 --
+ attr: name (v), weight (e)

```

```

[ ]: def task2(g, search_name, df):
    vertex = g.vs.find(name=search_name)
    print("Vertex index: ", vertex.index)

    #print(g.es.select(_source=vertex.index)['weight'])
    weights = g.es.select()[0]['weight']

    dtype = [('partner', float), ('weight', float)]
    v2_array=np.zeros((1, len(g.es.select()[0]['weight']))), dtype=dtype)

    #find the partners according to the vertex.index
    i=0
    for (v1,v2), weight in zip(g.get_edgelist(), np.asarray(weights)):
        if v1==vertex.index:
            v2_array[0,i]=(v2, weight)
            i+=1

    #order the partners according to the weights and choose the 3 best ones
    ordered_v2=np.sort(v2_array, axis=1, order='weight')
    #print(ordered_v2)
    #print(ordered_v2[0,-3:]['weight'])

    #find the corresponding partner names
    seq = g.vs.select(ordered_v2[0,-3:]['partner'])
    partners=[v['name'] for v in seq]

    print(partners)
    g_new = Graph()

```

```

g_new.add_vertices(4)
g_new.add_edges([(0,1), (0,2), (0,3)])
g_new.es['weight'] = list(ordered_v2[0,-3:]['weight'])
g_new.vs['name'] = [search_name, *partners]

#control
sorted_df = df[df["Reporter Countries"] == search_name]
sorted_df=sorted_df.sort_values(by=['Value'])
values=list(sorted_df[-3:]["Value"])
values=[float(i) for i in values]

#control
if (g_new.vs['name'][1:]==list(sorted_df[-3:]["Partner Countries"])):
    print('Partners are right')
if (g_new.es['weight']==values):
    print('Values are right')
return g_new

```

```

[ ]: g = task1(df_1.iloc[:,:])
g_new=task2(g, 'Uruguay', df_1)
plot_task(g_new, 'task2.pdf', 100)

```

```

IGRAPH DNW- 168 15402 --
+ attr: name (v), weight (e)
Vertex index: 36
['United States of America', 'Mexico', 'China']
Partners are right
Values are right

```

```

[ ]: def task3(g, search_name, df):
    vertex = g.vs.find(name=search_name)
    print("Vertex index: ", vertex.index)

    export_value=np.sum(g.es.select(_source=vertex.index)['weight'])
    import_value=np.sum(g.es.select(_target=vertex.index)['weight'])
    print('Yearly net export' , export_value)
    print('Yearly net import', import_value)

    if export_value <= import_value:
        print('This country is an net importer.')
    else:
        print('This country is an net exporter.')

#control
df_exporter = df[df["Reporter Countries"] == search_name]

```

```

df_importer = df[df["Partner Countries"] == search_name]

export_value_control=np.sum(df_exporter["Value"])
import_value_control=np.sum(df_importer["Value"])
#print(import_value_control)
#print(export_value_control)

#control
if (import_value_control==import_value):
    print('Import value is right')
if (export_value_control==export_value):
    print('Export value is right')
return g_new

```

```

[ ]: g = task1(df_1.iloc[:,:])
g_new=task3(g, 'Germany', df_1)

```

```

IGRAPH DNW- 168 15402 --
+ attr: name (v), weight (e)
Vertex index: 39
Yearly net export 3593196
Yearly net import 3785835
This country is an net importer.
Import value is right
Export value is right

```