## Network Visualization in Python

* Helper notebook for article of same name published on Medium.

### Import

import pandas as pd  
import networkx as nx  
import matplotlib.pyplot as plt  
%matplotlib inline

### Load and process data

# load data  
df = pd.read\_csv("data/book1.csv")  
# pick only important weights (hard threshold)  
df = df.loc[df['weight']>10, :]  
df

Source

Target

Type

weight

book

8

Aemon-Targaryen-(Maester-Aemon)

Jeor-Mormont

Undirected

13

1

9

Aemon-Targaryen-(Maester-Aemon)

Jon-Snow

Undirected

34

1

16

Aerys-II-Targaryen

Robert-Baratheon

Undirected

12

1

17

Aggo

Daenerys-Targaryen

Undirected

11

1

30

Alliser-Thorne

Jon-Snow

Undirected

32

1

…

…

…

…

…

…

658

Sandor-Clegane

Sansa-Stark

Undirected

23

1

664

Shae

Tyrion-Lannister

Undirected

12

1

666

Shagga

Tyrion-Lannister

Undirected

17

1

676

Tyrion-Lannister

Tywin-Lannister

Undirected

40

1

683

Waymar-Royce

Will-(prologue)

Undirected

18

1

175 rows × 5 columns

# import  
import networkx as nx  
# load pandas df as networkx graph  
G = nx.from\_pandas\_edgelist(df,   
 source='Source',   
 target='Target',   
 edge\_attr='weight')  
print("No of unique characters:", len(G.nodes))  
print("No of connections:", len(G.edges))

No of unique characters: 80  
No of connections: 175

## Option 1: NetworkX

# all graph options  
graphs\_viz\_options = [nx.draw, nx.draw\_networkx, nx.draw\_circular, nx.draw\_kamada\_kawai, nx.draw\_random, nx.draw\_shell, nx.draw\_spring]  
  
# plot graph option  
selected\_graph\_option = 0  
  
# plot  
plt.figure(figsize=(8,6), dpi=100)   
graphs\_viz\_options[selected\_graph\_option](G)

png

## Option 2: PyVis

# import pyvis  
from pyvis.network import Network  
# create vis network  
net = Network(notebook=True, width=1000, height=600)  
# load the networkx graph  
net.from\_nx(G)  
# show  
net.show("example.html")

## Option 3: Visdcc in Dash

* See dash\_app.py file for the demo.

### Extra - get all draw functions in NetworkX

from inspect import getmembers  
for x in getmembers(nx):  
 if 'draw' in x[0]:  
 print(x)

('draw', <function draw at 0x0000023163903BF8>)  
('draw\_circular', <function draw\_circular at 0x0000023163903F28>)  
('draw\_kamada\_kawai', <function draw\_kamada\_kawai at 0x0000023163909048>)  
('draw\_networkx', <function draw\_networkx at 0x0000023163903C80>)  
('draw\_networkx\_edge\_labels', <function draw\_networkx\_edge\_labels at 0x0000023163903EA0>)  
('draw\_networkx\_edges', <function draw\_networkx\_edges at 0x0000023163903D90>)  
('draw\_networkx\_labels', <function draw\_networkx\_labels at 0x0000023163903E18>)  
('draw\_networkx\_nodes', <function draw\_networkx\_nodes at 0x0000023163903D08>)  
('draw\_planar', <function draw\_planar at 0x00000231639092F0>)  
('draw\_random', <function draw\_random at 0x00000231639090D0>)  
('draw\_shell', <function draw\_shell at 0x0000023163909268>)  
('draw\_spectral', <function draw\_spectral at 0x0000023163909158>)  
('draw\_spring', <function draw\_spring at 0x00000231639091E0>)  
('drawing', <module 'networkx.drawing' from 'C:\\Users\\Mohit\\Anaconda3\\lib\\site-packages\\networkx\\drawing\\\_\_init\_\_.py'>)  
('planar\_drawing', <module 'networkx.algorithms.planar\_drawing' from 'C:\\Users\\Mohit\\Anaconda3\\lib\\site-packages\\networkx\\algorithms\\planar\_drawing.py'>)