

Les Misrebles Network Analysis using NetworkX ¶

More details at: <http://networkx.readthedocs.io/en/networkx-1.10/index.html> (<http://networkx.readthedocs.io/en/networkx-1.10/index.html>)

Source: <http://www-personal.umich.edu/~mejn/netdata/> (<http://www-personal.umich.edu/~mejn/netdata/>) and <http://networkdata.ics.uci.edu/data/lesmis/> (<http://networkdata.ics.uci.edu/data/lesmis/>) Code adopted from: <https://github.com/networkx/notebooks> (<https://github.com/networkx/notebooks>)

Citation: Les Miserables: coappearance network of characters in the novel Les Miserables. Please cite **D. E. Knuth, The Stanford GraphBase: A Platform for Combinatorial Computing, Addison-Wesley, Reading, MA (1993).**

```
In [1]: %matplotlib inline

from operator import itemgetter
import networkx as nx

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

import os
from io import StringIO
import pydotplus
from IPython.display import SVG, display
```

```
In [2]: sns.set_context("poster")
sns.set_style("ticks")
```

```
In [3]: DATA_DIR="../data"
INPUT_NETWORK=os.path.join(DATA_DIR, "lesmis", "lesmis.gml")
INPUT_NETWORK
```

```
Out[3]: '../data/lesmis/lesmis.gml'
```

Reading the GML format

Please read the following about the GML format for storing networks <http://networkx.readthedocs.io/en/networkx-1.10/reference/readwrite.gml.html#format> (<http://networkx.readthedocs.io/en/networkx-1.10/reference/readwrite.gml.html#format>)

```
In [4]: G = nx.read_gml(INPUT_NETWORK)
#nx.write_gml(G, "../data/lesmis/lesmis.paj.gml")
```

```
In [5]: df_node_degree = pd.DataFrame(list(dict(G.degree()).items()), columns=["node_name", "degree"])
```

```
In [6]: df_node_degree.sort_values("degree", ascending=False).head(10)
```

Out[6]:

	node_name	degree
10	Valjean	36
33	Gavroche	22
36	Marius	19
20	Javert	17
18	Thenardier	16
16	Fantine	15
37	Enjolras	15
38	Bossuet	13
62	Courfeyrac	13
69	Joly	12

```
In [7]: print("radius: {:d}\n".format(nx.radius(G)))
print("diameter: {:d}\n".format(nx.diameter(G)))
print("eccentricity: {}\n".format(nx.eccentricity(G)))
print("center: {}\n".format(nx.center(G)))
print("periphery: {}\n".format(nx.periphery(G)))
print("density: {:f}".format(nx.density(G)))
```

radius: 3

diameter: 5

eccentricity: {'Myriel': 4, 'Napoleon': 5, 'MlleBaptistine': 4, 'MmeMagloire': 4, 'CountessDeLo': 5, 'Geborand': 5, 'Champtercier': 5, 'Cravatte': 5, 'Count': 5, 'OldMan': 5, 'Valjean': 3, 'Labarre': 4, 'Marguerite': 4, 'MmeDeR': 4, 'Isabeau': 4, 'Gervais': 4, 'Fantine': 4, 'MmeThenardier': 4, 'Thenardier': 3, 'Cosette': 4, 'Javert': 3, 'Fauchevent': 4, 'Bamatabois': 4, 'Simplice': 4, 'Scaufflaire': 4, 'Woman1': 4, 'Judge': 4, 'Champfamille': 4, 'Brevet': 4, 'Chenildieu': 4, 'Cochepaille': 4, 'Woman2': 4, 'MotherInnocent': 4, 'Gavroche': 3, 'Gillenormand': 4, 'MlleGillenormand': 4, 'Marius': 3, 'Enjolras': 3, 'Bossuet': 3, 'Gueulemer': 3, 'Babet': 3, 'Claquesous': 4, 'Montparnasse': 3, 'Toussaint': 4, 'Tholomyes': 4, 'Listolier': 5, 'Fameuil': 5, 'Blacheville': 5, 'Favourite': 5, 'Dahlia': 5, 'Zephine': 5, 'Perpetue': 5, 'Eponine': 4, 'Anzelma': 4, 'Magnon': 5, 'Pontmercy': 4, 'Boulatruelle': 4, 'Brujon': 4, 'LtGillenormand': 4, 'Gribier': 5, 'MmePontmercy': 5, 'Mabeuf': 4, 'Courfeyrac': 4, 'Jondrette': 5, 'MmeBurton': 4, 'Combeferre': 4, 'Prouvaire': 4, 'Feuilly': 4, 'Bahorel': 4, 'Joly': 4, 'Grantaire': 4, 'Child1': 4, 'Child2': 4, 'MmeHucheloup': 4, 'BaronessT': 4, 'MlleVaubois': 5, 'MotherPlutarch': 5}

center: ['Valjean', 'Thenardier', 'Javert', 'Gavroche', 'Marius', 'Enjolras', 'Bossuet', 'Gueulemer', 'Babet', 'Montparnasse']

periphery: ['Napoleon', 'CountessDeLo', 'Geborand', 'Champtercier', 'Cravatte', 'Count', 'OldMan', 'Listolier', 'Fameuil', 'Blacheville', 'Favourite', 'Dahlia', 'Zephine', 'Perpetue', 'Magnon', 'Gribier', 'MmePontmercy', 'Jondrette', 'MlleVaubois', 'MotherPlutarch']

density: 0.086808

Connected components

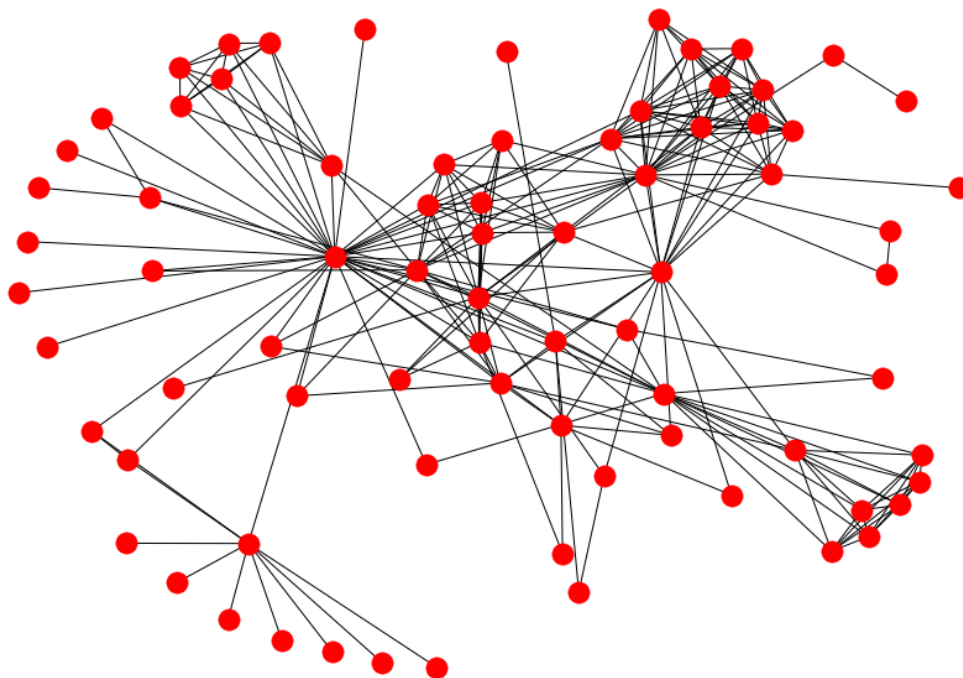
```
In [8]: connected_components = sorted(nx.connected_component_subgraphs(G), key = len,
reverse=True)
print("{} connected components found.".format(len(connected_components)))

1 connected components found.
```

Drawing the graph

```
In [9]: nx.draw(G)

/home/napsternxg/anaconda3/envs/get17_sna/lib/python3.6/site-packages/networkx/drawin
g/nx_pylab.py:126: MatplotlibDeprecationWarning: pyplot.hold is deprecated.
Future behavior will be consistent with the long-time default:
plot commands add elements without first clearing the
Axes and/or Figure.
b = plt.ishold()
/home/napsternxg/anaconda3/envs/get17_sna/lib/python3.6/site-packages/networkx/drawin
g/nx_pylab.py:138: MatplotlibDeprecationWarning: pyplot.hold is deprecated.
Future behavior will be consistent with the long-time default:
plot commands add elements without first clearing the
Axes and/or Figure.
plt.hold(b)
/home/napsternxg/anaconda3/envs/get17_sna/lib/python3.6/site-packages/matplotlib/_ini
t_.py:917: UserWarning: axes.hold is deprecated. Please remove it from your matplotli
brc and/or style files.
warnings.warn(self.msg_depr_set % key)
/home/napsternxg/anaconda3/envs/get17_sna/lib/python3.6/site-packages/matplotlib/rcset
up.py:152: UserWarning: axes.hold is deprecated, will be removed in 3.0
warnings.warn("axes.hold is deprecated, will be removed in 3.0")
/home/napsternxg/anaconda3/envs/get17_sna/lib/python3.6/site-packages/matplotlib/font_
manager.py:1297: UserWarning: findfont: Font family ['sans-serif'] not found. Falling
back to DejaVu Sans
(prop.get_family(), self.defaultFamily[fonttext]))
```




```
In [12]: show_graph(G, "../output/lesmis.svg")
```

