Chapter 5 - The Small Scale: Nodes and Centrality

In [ ]:

```
# Configure plotting in Jupyter
from matplotlib import pyplot as plt
%matplotlib inline
plt.rcParams.update({
    'figure.figsize': (7.5, 7.5),
    'axes.spines.right': False,
    'axes.spines.left': False,
    'axes.spines.top': False,
    'axes.spines.bottom': False})
# Seed random number generator
from numpy import random as nprand
seed = hash("Network Science in Python") % 2**32
nprand.seed(seed)
```

In [ ]:

```
import networkx as nx
```

In [ ]:

```
# Create empty affiliation network and list of people
B = nx.Graph()
people = set()
# Load data file into network
from pathlib import Path
data_dir = Path('.') / 'data'
with open(data_dir / 'crossley2012' / '50_ALL_2M.csv') as f:
    # Parse header
    events = next(f).strip().split(",")[1:]
    # Parse rows
    for row in f:
        parts = row.strip().split(",")
        person = parts[0]
        people.add(person)
        for j, value in enumerate(parts[1:]):
            if value != "0":
                B.add_edge(person, events[j], weight=int(value))
# Project into person-person co-affilation network
from networkx import bipartite
G = bipartite.projected_graph(B, people)
```

In [ ]:

```
betweenness = nx.betweenness_centrality(G, normalized=False)
sorted(betweenness.items(), key=lambda x:x[1], reverse=True)[0:10]
```

Out[ ]:

```
[('Maud Joachim', 52896.53324605232),
 ('Ada Wright', 26344.263264276895),
 ('Patricia Woodlock', 24774.92342232244),
 ('Emily Duval', 19517.906214119474),
 ('Mary Leigh', 19404.225833772107),
 ('Mabel Capper', 18221.362811581766),
 ('Sylvia Pankhurst', 18127.596886368978),
 ('Elsie Evans', 15674.80629870348),
 ('Winifred Mayo', 15600.98968032138),
 ('Vera Wentworth', 13233.504078942542)]
```

In [ ]:

```
eigenvector = nx.eigenvector_centrality(G)
sorted(eigenvector.items(), key=lambda x:x[1], reverse=True)[0:10]
```

Out[ ]:

```
[('Maud Joachim', 0.11587964174472974),
 ('Caroline A Downing', 0.11437066100686205),
 ('Kitty Marion', 0.11344996012448638),
 ('Mabel Capper', 0.10991776240126293),
 ('Annie Bell', 0.10834705221110302),
 ('Grace Chappelow', 0.10818185244249964),
 ('Winifred Mayo', 0.10803831965810351),
 ('Ellen Pitfield', 0.10518714292397996),
 ('Dorothy Agnes Bowker', 0.10493919222545885),
 ('Mrs Maud Fussell', 0.10490326319130665)]
```

In [ ]:

```
closeness = nx.closeness_centrality(G)
sorted(closeness.items(), key=lambda x:x[1], reverse=True)[0:10]
```

Out[ ]:

```
[('Maud Joachim', 0.5357241748956739),
 ('Caroline A Downing', 0.5009438937877011),
 ('Winifred Mayo', 0.5009438937877011),
 ('Mabel Capper', 0.5006919099377073),
 ('Kitty Marion', 0.49793672684150186),
 ('Ada Wright', 0.4898501559823633),
 ('Patricia Woodlock', 0.4886477746471095),
 ('Vera Wentworth', 0.48769011119851163),
 ('Evelyn Whurry', 0.4874512815652116),
 ('Annie Bell', 0.4869743233640714)]
```

In [ ]:

```
triangles = nx.triangles(G)
sorted(triangles.items(), key=lambda x:x[1], reverse=True)[0:10]
```

Out[ ]:

```
[('Maud Joachim', 19687),
 ('Caroline A Downing', 18201),
 ('Kitty Marion', 17696),
 ('Mabel Capper', 16811),
 ('Winifred Mayo', 16455),
 ('Annie Bell', 16065),
 ('Grace Chappelow', 16018),
 ('Ellen Pitfield', 14910),
 ('Mrs Maud Fussell', 14841),
 ('Dorothy Agnes Bowker', 14750)]
```

In [ ]:

```
clustering = nx.clustering(G)
[(x, clustering[x]) for x in sorted(people, key=lambda x:eigenvector[x], reverse=True)[0:10]]
```

Out[ ]:

```
[('Maud Joachim', 0.23595330552759),
 ('Caroline A Downing', 0.34999903851700864),
 ('Kitty Marion', 0.3670988486671507),
 ('Mabel Capper', 0.33992518451117176),
 ('Annie Bell', 0.4233201581027668),
 ('Grace Chappelow', 0.43461037551551984),
 ('Winifred Mayo', 0.3480477177545582),
 ('Ellen Pitfield', 0.4828993392926545),
 ('Dorothy Agnes Bowker', 0.5058125578683859),
 ('Mrs Maud Fussell', 0.5006071645415908)]
```

In [ ]: