CS 224W - Hwk 0
Ryan Foley

1.2
g = nx.read_adjlist('wiki-Vote.txt.gz', create_using=nx.DiGraph())

1.3
a) Number nodes: 7115
b) Number of nodes with a self-edge: 0
c) Number of directed edges in the network: 103689
d) Number of undirected edges in the network: 100762
e) Number of reciprocated edges in the network: 2927
f) Number of nodes of zero out-degree: 1005
g) Number of nodes of zero in-degree: 4734
h) Number of nodes with more than 10 outgoing edges: 1612
i) Number of nodes with less than 10 incoming edges: 5165

```
# 1.3
# a)
print 'Number nodes:', len(g)

# b)
print 'Number of nodes with a self-edge:', len(g.nodes_with_selfloops())

# c)
print 'Number of directed edges in the network:', len(g.edges())

# d)
g_undir = g.to_undirected()
print 'Number of undirected edges in the network:', len(g_undir.edges())

# e)
print 'Number of reciprocated edges in the network:', len(g.edges()) - len(g_undir.edges())

# f)
zero_out_nodes = Set()
for node in g.nodes():
    if g.out_degree(node) == 0:
        zero_out_nodes.add(node)

print 'Number of nodes of zero out-degree:', len(zero_out_nodes)

# g)
zero_in_nodes = Set()
for node in g.nodes():
    if g.in_degree(node) == 0:
        zero_in_nodes.add(node)

print 'Number of nodes of zero in-degree:', len(zero_in_nodes)

# h)
```

```
count = 0
for node in g.nodes():
    if g.out_degree(node) > 10:
        count += 1
print 'Number of nodes with more than 10 outgoing edges:', count

# i)
count = 0
for node in g.nodes():
    if g.in_degree(node) < 10:
        count += 1
print 'Number of nodes with less than 10 incoming edges:', count
```