Homework - Case Study 6

July 3, 2017

Network homophily occurs when nodes that share an edge share a characteristic more often than nodes that do not share an edge. In this case study, we will investigate homophily of several characteristics of individuals connected in social networks in rural India.

```
In [109]: import pandas as pd
```

`individual_characteristics.dta` contains several characteristics for each individual in the dataset such as age, religion, and caste. Use the pandas library to read in and store these characteristics as a dataframe called df

```
In [110]: data_filepath = 'https://s3.amazonaws.com/assets.datacamp.com/production/
          df  = pd.read_stata(data_filepath + "individual_characteristics.dta")
```

Store separate datasets for individuals belonging to Villages 1 and 2 as df1 and df2, respectively.

```
In [111]: df1 = df[df.village == 1]
          df2 = df[df.village == 2]

          # Enter code here!
          df1.head()

Out[111]:    village  adjmatrix_key     pid  hhid  resp_id  resp_gend  \
          0        1              5  100201  1002        1          1
          1        1              6  100202  1002        2          2
          2        1             23  100601  1006        1          1
          3        1             24  100602  1006        2          2
          4        1             27  100701  1007        1          1

                           resp_status  age  religion caste        ...          \
          0             Head of Household   38  HINDUISM   OBC        ...
          1   Spouse of Head of Household   27  HINDUISM   OBC        ...
          2             Head of Household   29  HINDUISM   OBC        ...
          3   Spouse of Head of Household   24  HINDUISM   OBC        ...
          4             Head of Household   58  HINDUISM   OBC        ...

                    privategovt work_outside work_outside_freq shgparticipate shg_no
```

```
0  PRIVATE BUSINESS          Yes                  0            No     NaN
1                            NaN                NaN            No     NaN
2        OTHER LAND           No                NaN            No     NaN
3  PRIVATE BUSINESS           No                NaN           Yes       1
4        OTHER LAND           No                NaN            No     NaN

  savings savings_no electioncard rationcard rationcard_colour
0      No        NaN          Yes        Yes             GREEN
1      No        NaN          Yes        Yes             GREEN
2      No        NaN          Yes        Yes             GREEN
3     Yes        1.0          Yes         No
4      No        NaN          Yes        Yes             GREEN

[5 rows x 48 columns]
```

In this dataset, each individual has a personal ID, or PID, stored in key_vilno_1.csv and key_vilno_2.csv for villages 1 and 2, respectively. Use pd.read_csv to read in and store key_vilno_1.csv and key_vilno_2.csv as pid1 and pid2 respectively.

```
In [112]: pid1 = pd.read_csv(data_filepath + "key_vilno_1.csv", dtype=int, header =
          pid2 = pd.read_csv(data_filepath + "key_vilno_2.csv", dtype=int, header =
```

```
In [113]: type(pid1)
```

```
Out[113]: pandas.core.frame.DataFrame
```

```
In [114]: sex1      = dict(zip(df1.pid, df1.resp_gend))
          caste1    = dict(zip(df1.pid, df1.caste))
          religion1 = dict(zip(df1.pid, df1.religion))

          # Continue for df2 as well.
          sex2      = dict(zip(df2.pid, df2.resp_gend))
          caste2    = dict(zip(df2.pid, df2.caste))
          religion2 = dict(zip(df2.pid, df2.religion))
```

Let's consider how much homophily exists in these networks. For a given characteristic, our measure of homophily will be the proportion of edges in the network whose constituent nodes share that characteristic. How much homophily do we expect by chance? If characteristics are distributed completely randomly, the probability that two nodes x and y share characteristic a is the probability both nodes have characteristic a, which is the frequency of a squared. The total probability that nodes x and y share their characteristic is therefore the sum of the frequency of each characteristic in the network. For example, in the dictionary favorite_colors provided, the frequency of red and blue is 1/3 and 2/3 respectively, so the chance homophily is $(1/3)^{2+(2/3)}2$ = 5/9. Create a function chance_homophily(chars) that takes a dictionary with personal IDs as keys and characteristics as values, and computes the chance homophily for that characteristic.

```
In [115]: from collections import Counter
          import numpy as np
```

```
def chance_homophily(chars):
    """
    Computes the chance homophily of a characteristic,
    specifie
    """
#     c = dict(Counter(chars.values()))
#     result = 0
#     for v in c.values():
#         result += (v / sum(c.values()))**2
#     return result

    # datacamp answer
    chars_counts_dict = Counter(chars.values())
    chars_counts = np.array(list(chars_counts_dict.values()))
    chars_props  = chars_counts / sum(chars_counts)
    return sum(chars_props**2)

favorite_colors = {
    "ankit":  "red",
    "xiaoyu": "blue",
    "mary":   "blue"
}

color_homophily = chance_homophily(favorite_colors)
print(color_homophily)
```

```
0.555555555556
```

```
In [116]: print("Village 1 chance of same sex:", chance_homophily(sex1))
          print("Village 1 chance of same caste:", chance_homophily(caste1))
          print("Village 1 chance of same religion:", chance_homophily(religion1))
          print()
          print("Village 2 chance of same sex:", chance_homophily(sex2))
          print("Village 2 chance of same caste:", chance_homophily(caste2))
          print("Village 2 chance of same religion:", chance_homophily(religion2))
```

```
Village 1 chance of same sex: 0.502729986168
Village 1 chance of same caste: 0.674148850979
Village 1 chance of same religion: 0.980489698852

Village 2 chance of same sex: 0.500594530321
Village 2 chance of same caste: 0.425368244801
Village 2 chance of same religion: 1.0
```

Now let's compute the observed homophily in our network. Recall that our measure of homophily is the proportion of edges whose nodes share a characteristic. homophily(G, chars, IDs) takes a network G, a dictionary of characteristics chars, and node IDs IDs. For each node pair, determine whether a tie exists between them, as well as whether they share a characteristic. The total count of these is num_same_ties and num_ties respectively, and their ratio is the homophily of chars in G. Complete the function by choosing where to increment num_same_ties and num_ties.

```
In [122]: def homophily(G, chars, IDs):
              """
              Given a network G, a dict of characteristics chars for node IDs,
              and dict of node IDs for each node in the network,
              find the homophily of the network.
              """
              num_same_ties, num_ties = 0, 0
              for n1 in G.nodes():
                  for n2 in G.nodes():
                      if n1 > n2:   # do not double-count edges!
                          if IDs[n1] in chars and IDs[n2] in chars:
                              if G.has_edge(n1, n2):
                                  num_ties += 1
                                  if chars[IDs[n1]] == chars[IDs[n2]]:
                                      num_same_ties += 1
              return (num_same_ties / num_ties)
```

Use your homophily function to compute the observed homophily for sex, caste, and religion in Villages 1 and 2.

```
In [118]: # load networkx graph objects G1 and G2
          import networkx as nx

          A1 = np.loadtxt("adj_allVillageRelationships_vilno_1.csv", delimiter=",")
          A2 = np.loadtxt("adj_allVillageRelationships_vilno_2.csv", delimiter=",")

          G1 = nx.to_networkx_graph(A1)
          G2 = nx.to_networkx_graph(A2)
```

```
In [119]: # convert dataframe to numpy array
          array_pid1 = np.array(pid1[0])
          array_pid2 = np.array(pid2[0])
```

```
In [123]: print("Village 1 observed proportion of same sex:", homophily(G1, sex1, a
          print("Village 1 observed proportion of same caste:", homophily(G1, caste
          print("Village 1 observed proportion of same religion:", homophily(G1, re
          print()
          print("Village 2 observed proportion of same sex:", homophily(G2, sex2, a
          print("Village 2 observed proportion of same caste:", homophily(G2, caste
          print("Village 2 observed proportion of same religion:", homophily(G2, re
```

```
Village 1 observed proportion of same sex: 0.5879345603271984
Village 1 observed proportion of same caste: 0.7944785276073619
Village 1 observed proportion of same religion: 0.99079754601227
Village 2 observed proportion of same sex: 0.5622435020519836
Village 2 observed proportion of same caste: 0.826265389876881
Village 2 observed proportion of same religion: 1.0
```