TEXT AND WEB ANALYTICS ASSIGNMENT

CASE STUDY

IMPLEMENT HOMOPHILY FOR SOCIAL MEDIA ANALYSIS

Submitted By:	Mahima Munjal
Class:	VIII-B
Roll Number:	17CSU098
Taught By:	Ms. Vaishali Kalra

Department of Computer Science and Engineering

School of Engineering and Technology

The Northcap University, Gurugram- 122001, India

Session 2020-21

DATA CAMP CASE STUDY 6

Q1. 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 this exercise, we will calculate the chance homophily for an arbitrary characteristic. Homophily is 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.

100 XP

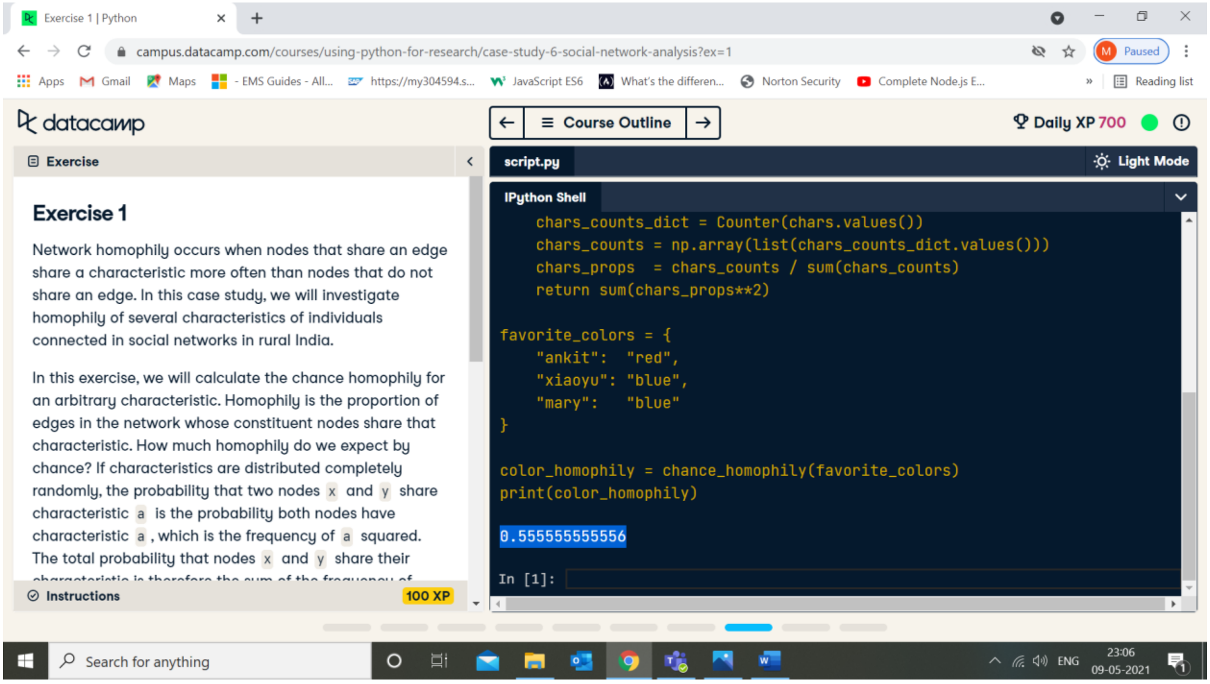
- Create a function that takes a dictionary `chars` with personal IDs as keys and characteristics as values, and returns a dictionary with characteristics as keys, and the frequency of their occurrence as values.
- Create a function `chance_homophily(chars)` that takes a dictionary `chars` defined as above and computes the chance homophily for that characteristic.
- A sample of three peoples' favorite colors is given in `favorite_colors`. Use your function to compute the chance homophily in this group, and store as `color_homophily`.
- Print `color_homophily`.

CODE:

```
from collections import Counter
def chance_homophily(chars):
    """
    Computes the chance homophily of a characteristic,
    specified as a dictionary, chars.
    """
    #enter your code here
    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)
```

OUTPUT:



Q2. 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 the remaining exercises, we will calculate and compare the actual homophily in these village to chance. In this exercise, we subset the data into individual villages and store them.

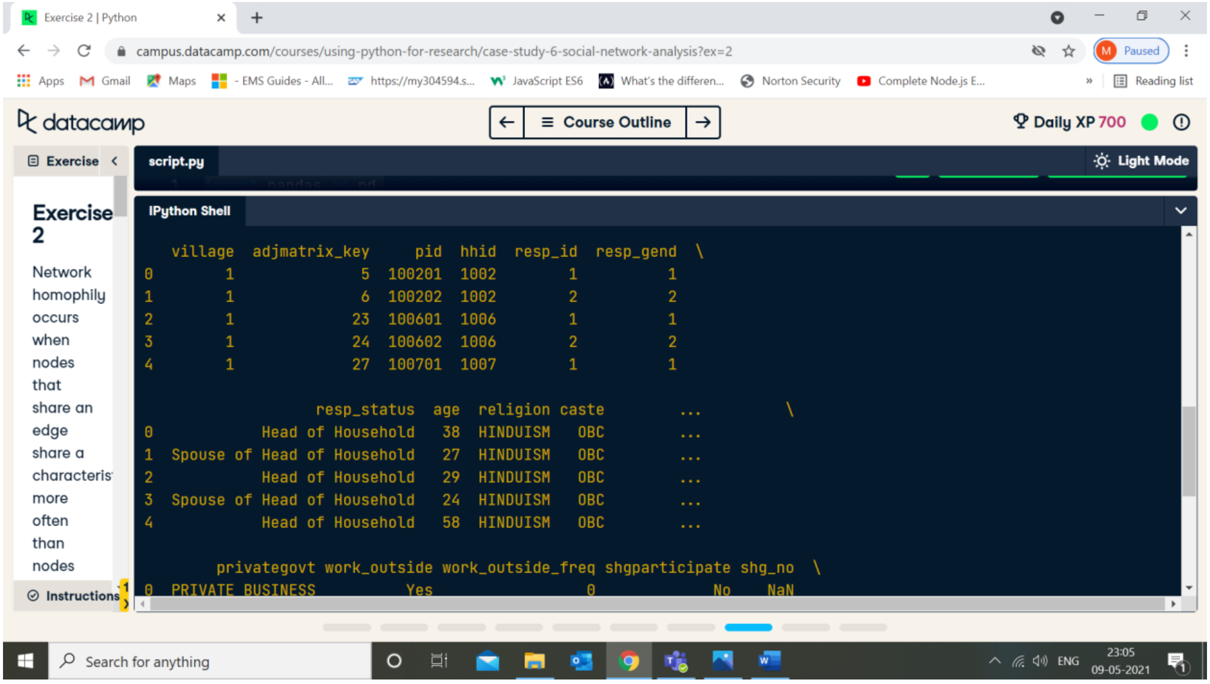
Instructions
100 XP

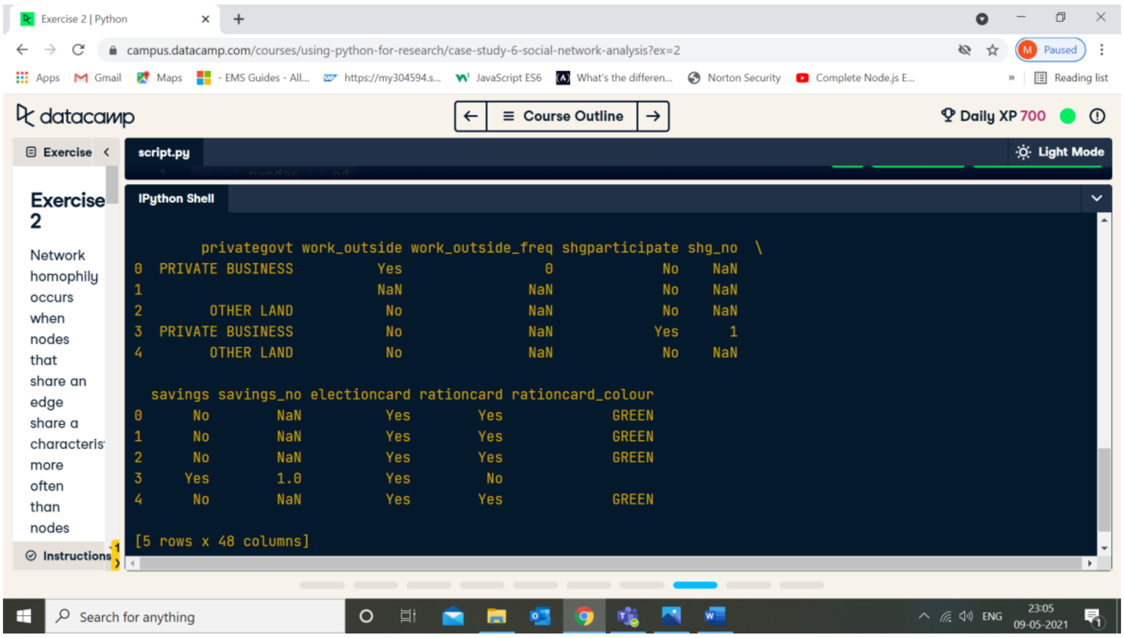
- `individual_characteristics.dta` contains several characteristics for each individual in the dataset such as age, religion, and caste. Use the `pandas` library to read in and store these characteristics as a dataframe called `df`.
- Store separate datasets for individuals belonging to Villages 1 and 2 as `df1` and `df2`, respectively.
    - Note that some attributes may be missing for some individuals. In this case study, we will ignore rows of data where some column information is missing.
- Use the `head` method to display the first few entries of `df1`.

CODE :

```python
import pandas as pd
df  = pd.read_stata(data_filepath + "individual_characteristics.dta"
)
df1 = df[df["village"]==1]# Enter code here!
df2 = df[df["village"]==2]# Enter code here!

# Enter code here!
df1.head()
```

OUTPUT:





Q3. 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 this exercise, we define a few dictionaries that enable us to look up the sex, caste, and religion of members of each village by personal ID. For Villages 1 and 2, their personal IDs are stored as `pid`.

Instructions
100 XP

- Define dictionaries with personal IDs as keys and a given covariate for that individual as values. Complete this for the sex, caste, and religion covariates, for Villages 1 and 2.
- For Village 1, store these dictionaries into variables named `sex1`, `caste1`, and `religion1`.
- For Village 2, store these dictionaries into variables named `sex2`, `caste2`, and `religion2`.

CODE:

```
sex1      = df1.set_index("pid")["resp_gend"].to_dict()
caste1    = df1.set_index("pid")["caste"].to_dict()
religion1 = df1.set_index("pid")["religion"].to_dict()

sex2      = df2.set_index("pid")["resp_gend"].to_dict()
caste2    = df2.set_index("pid")["caste"].to_dict()
religion2 = df2.set_index("pid")["religion"].to_dict()
```

Q4. 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 this exercise, we will print the chance homophily of several characteristics of Villages 1 and 2. The function `chance_homophily` is still defined from Exercise 1.

Instructions
100 XP

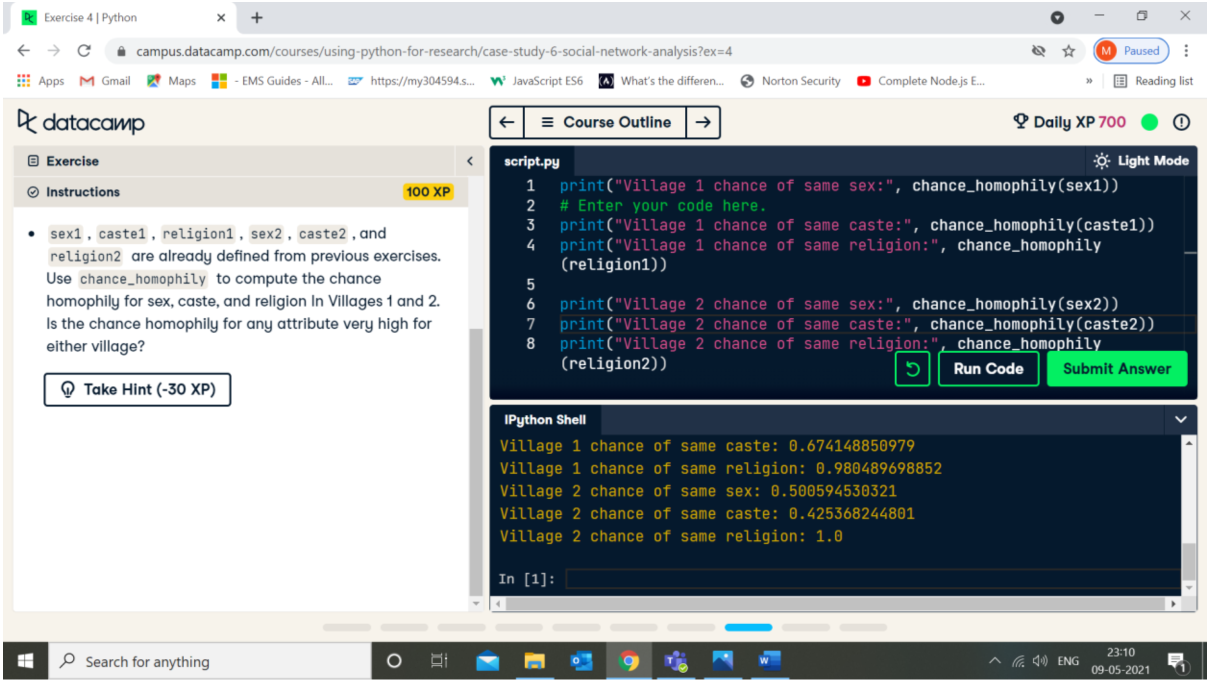
- `sex1`, `caste1`, `religion1`, `sex2`, `caste2`, and `religion2` are already defined from previous exercises. Use `chance_homophily` to compute the chance homophily for sex, caste, and religion In Villages 1 and 2. Is the chance homophily for any attribute very high for either village?

CODE :

```
print("Village 1 chance of same sex:", chance_homophily(sex1))
```

```python
# Enter your code here.
print("Village 1 chance of same caste:", chance_homophily(caste1))
print("Village 1 chance of same religion:", chance_homophily(religion1))

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))
```

OUTPUT:



Q5. 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 this exercise, we will create a function that computes the observed homophily given a village and characteristic.

Instructions
100 XP

- Complete the function `homophily()`, which 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`.

CODE:

```
def homophily(G, chars, IDs):
    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# Should `num_ties` be incremen
ted?  What about `num_same_ties`?
                        if chars[IDs[n1]] == chars[IDs[n2]]:
                            num_same_ties += 1# Should `num_ties` be
 incremented?  What about `num_same_ties`?
    return (num_same_ties / num_ties)
```

Q6. 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 this exercise, we will obtain the personal IDs for Villages 1 and 2. These will be used in the next exercise to calculate homophily for these villages.

Instructions
100 XP

- 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. `data_filepath` contains the base URL to the datasets used in this exercise. Use `pd.read_csv` to read in and store `key_vilno_1.csv` and `key_vilno_2.csv` as `pid1` and `pid2` respectively. The `csv` files have no headers, so make sure to include the parameter `header = None`.

CODE:

```
pid1 = pd.read_csv(data_filepath + "key_vilno_1.csv", dtype=int, hea
der = None)
pid2 = pd.read_csv(data_filepath + "key_vilno_2.csv", dtype=int, hea
der = None)
```

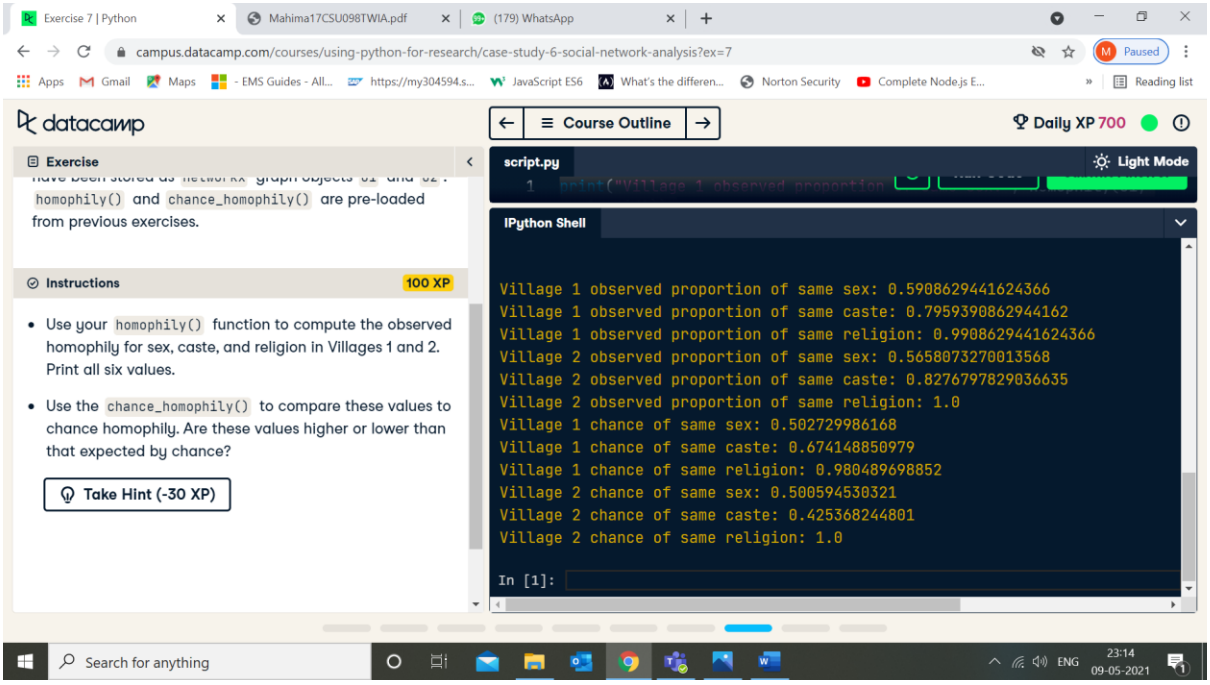
Q7. Network homophily occurs when nodes that share an edge share a characteristic more often than nodes that do not share an edge. In this case study, we will investigate homophily of several characteristics of individuals connected in social networks in rural India.

In this exercise, we will compute the homophily of several network characteristics for Villages 1 and 2, and compare this to chance homophily. The networks for these villages have been stored as `networkx` graph objects `G1` and `G2`. `homophily()` and `chance_homophily()` are pre-loaded from previous exercises.

CODE :

```
print("Village 1 observed proportion of same sex:", homophily(G1, se
x1, pid1))
# Enter your code here!

print("Village 1 observed proportion of same caste:", homophily(G1,
caste1, pid1))
print("Village 1 observed proportion of same religion:", homophily(G
1, religion1, pid1))

print("Village 2 observed proportion of same sex:", homophily(G2, se
x2, pid2))
print("Village 2 observed proportion of same caste:", homophily(G2,
caste2, pid2))
print("Village 2 observed proportion of same religion:", homophily(G
2, religion2, pid2))

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(religio
n1))

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(religio
n2))
```

OUTPUT:


Exercise 7 | Python
Mahima17CSU098TWIA.pdf
(179) WhatsApp
campus.datacamp.com/courses/using-python-for-research/case-study-6-social-network-analysis?ex=7
datacamp
Course Outline
Daily XP 700
Exercise
homophily() and chance_homophily() are pre-loaded from previous exercises.
Instructions
100 XP
Use your homophily() function to compute the observed homophily for sex, caste, and religion in Villages 1 and 2. Print all six values.
Use the chance_homophily() to compare these values to chance homophily. Are these values higher or lower than that expected by chance?
Take Hint (-30 XP)
script.py
Light Mode
IPython Shell
Village 1 observed proportion of same sex: 0.5908629441624366
Village 1 observed proportion of same caste: 0.7959390862944162
Village 1 observed proportion of same religion: 0.9908629441624366
Village 2 observed proportion of same sex: 0.5658073270013568
Village 2 observed proportion of same caste: 0.8276797829036635
Village 2 observed proportion of same religion: 1.0
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
In [1]:
Search for anything
ENG
23:14
09-05-2021