Introduction to Web Science

Assignment 10

Prof. Dr. Steffen Staab
staab@uni-koblenz.de

René Pickhardt
rpickhardt@uni-koblenz.de

Korok Sengupta
koroksengupta@uni-koblenz.de

Olga Zagovora
zagovora@uni-koblenz.de

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Submission until: January 25, 2016, 10:00 a.m.
Tutorial on: January 27, 2016, 12:00 p.m.

For all the assignment questions that require you to write code, **make sure to include the code in the answer sheet, along with a separate python file. Where screen shots are required, please add them in the answers directly and not as separate files.**

Team Name: mike Team members: Slobodan Kocevski, Shohel Ahamad, Anish Girijashivaraj

1 Modeling Twitter data (10 points)

In the meme paper[1] by Weng et al., in Figure 2[2] you find a plot, comparing the system entropy with the average user entropy. Your task is to reproduce the plot and corresponding calculations.

1. We provide you with the file 'onlyhashtag.data', containing a collection of hashtags from tweets. Use this data to reproduce the plot from the paper. Once you have the values for average user entropy and system entropy calculated per day create a scatter plot to display the values.
2. Interpret the scatter plot and compare it with the authors interpretation from the graph showed in the paper. Will the interpretations be compatible to each other or will they contradict each other? Do not write more than 5 sentences.

1.1 Hints

1. Use formulas from the lecture to calculate the entropy for one user and the system entropy.
2. Do not forget to give proper names of plot axes.

Answer:

```
1:
2: import math
3:
4: f1=open("onlyhash.data", "r", encoding="utf -8")
5: m = f1.read()
6: splittedContent = m.split('\n')
7:
8: print(len(splittedContent))
9:
10: i = 0
11: counter = 0
12: users = []
13: NuTweets = []
14: for i in range(len(splittedContent) - 1):
15:     j = 0
16:     rowSpletted = splittedContent[i].split()
17:     userN = rowSpletted[0]
18:     if (userN in users):
19:         continue
20:     else:
21:         users.append(userN)
```

[1] http://www.nature.com/articles/srep00335
[2] Slide 27, Lecture Meme spreading on the Web

```
22:         counter = 0
23:         while (j < 10000):
24:             rwSpl = splittedContent[j].split()
25:             usrN = rwSpl[0]
26:             if(userN == usrN):
27:                 counter = counter + 1
28:             j = j + 1
29:         NuTweets.append(counter)
30:     print(str(i))
31:
32: userEntropy = []
33: for i in NuTweets:
34:     userEntropy.append((1/NuTweets[i]) +  math.log(1/NuTweets[i], 10))
35:
36: memes = []
37: dates = []
38: for i in range(len(splittedContent) - 1):
39:     rowSpletted = splittedContent[i].split()
40:     meme = rowSpletted[2]
41:     date = rowSpletted[1]
42:     memes.append(meme)
43:     dates.append(date)
44:
45: date_meme_list = []
46: if not date in dates:
47:     dates.append(date)
48:     date_meme_list.append({})
49: else:
50:     for one in memes:
51:         ind = dates.index(date)
52:     if one in date_meme_list[ind]:
53:         date_meme_list[ind][one] += 1
54:     else:
55:         date_meme_list[ind][one] = 1
56:
57: sys_entro = []
58: for index in range(len(dates)):
59:     entro = 0
60:     n = sum(date_meme_list[index][one] for one in date_meme_list[index])
61:     for meme in date_meme_list[index]:
62:         fu = date_meme_list[index][meme] / n
63:         entro -= fu * (math.log10(fu))
64:     sys_entro.append(entro)
65:
66:
67: from matplotlib import pyplot as plt
68: import numpy as np
69:
70: plt.plot(userEntropy, color="red")
```

```
71: plt.plot(sys_entro, color="blue")
72: x = np.arange(0, len(dates), 1)
73: # print (x,len(order_sys_entropy))
74: plt.xlim(1, len(dates) + 1)
75: plt.ylim(0, sys_entro[len(dates) - 1])
76: # set the caption
77: plt.title('Daily system entropy', fontsize=10, fontweight='bold')
78: # the label of x-axis
79: plt.xlabel('rank')
80: # the label of y-axis
81: plt.ylabel('entropy')
82: plt.show()
```

2 Measuring inequality (10 points)

We provide you with a sample implementation of the Chinese Restaurant Process[3].

Assume there is a restaurant with an infinite number of tables. When a new customer enters a restaurant he chooses an occupied table or the next empty table with some probabilities.

According to the process first customer always sits at the first table. Probability of the next customer to sit down at an occupied table i equals ratio of guests sitting at the table (c_i/n), where n is the number of guests in the restaurant and c_i is the number of guests sitting at table i.
Probability of customer to choose an empty table equals : $1 - \sum_{i=1}^{S} p_i$, where S is the number of occupied tables and $p_i = c_i/n$.

Provided script simulates the process and returns number of people sitting at each table. We will study restaurants for 1000 customers. Now you should modify the code and evaluate how unequal were the customers' choices of tables.

Calculate the Gini- coefficient measuring the inequality between the tables, until the coefficient stabilizes. Do five different runs and plot your results in a similar way that plots in the lecture slides are done, cf. Slide 32 and Slide 33.

Answer:

```
 1: import random
 2: import json
 3: import matplotlib.pyplot as plt
 4: def generateChineseRestaurant(customers):
 5:     # First customer always sits at the first table
 6:         tables = [1]
 7:         #for all other customers do
 8:         for cust in range(2, customers+1):
 9:                 # rand between 0 and 1
10:                 rand = random.random()
11:                 # Total probability to sit at a table
12:                 prob = 0
13:                 # No table found yet
14:                 table_found = False
15:                 # Iterate over tables
16:
17:                 for table, guests in enumerate(tables):
18:                     # calc probability for actual table an add it to total probabi
19:                     prob += float(guests) / float(cust)
20:                     # If rand is smaller than the current total prob., customer wi
21:                     if rand < prob:
22:                         # incr. #customers for that table
```

[3]File "chinese_restaurant.py"; Additional information can be found here: https://en.wikipedia.org/wiki/Chinese_restaurant_process

```
                            tables[table] += 1
                            # customer has found table
                            table_found = True
                            # no more tables need to be iterated, break out for loop
                            break
                    # If table iteration is over and no table was found, open new tabl
                    if not table_found:
                        tables.append(1)
            GPlot(tables)
            giniVal = gini(tables)
            print ("gini coefficient of %s Guest is = %s " % (customers, giniVal))

def GPlot(table_list):
    table_list = sorted(table_list)
    var_list = []
    plotList = []
    for x in table_list:
        var_list.append(x)
        g = gini(var_list)
        plotList.append(g)
    plt.plot(var_list,plotList)
    plt.ylabel('Gini coefficient')
    plt.xlabel('Guest')
    plt.show()

def gini(list_of_values):
    sorted_list = sorted(list_of_values)
    height, area = 0, 0
    for value in sorted_list:
        height += value
        area += height - value / 2.
    fair_area = height * len(list_of_values) / 2.
    return (fair_area - area) / fair_area
for i in range(0,5):
    allRestaurants = [1000,10000,100000,1000000,10000000]
    restaurants = allRestaurants[i]
    network = generateChineseRestaurant(restaurants)
    with open('network_' + str(restaurants) + '.json', 'w') as out:
        json.dump(network, out)
```

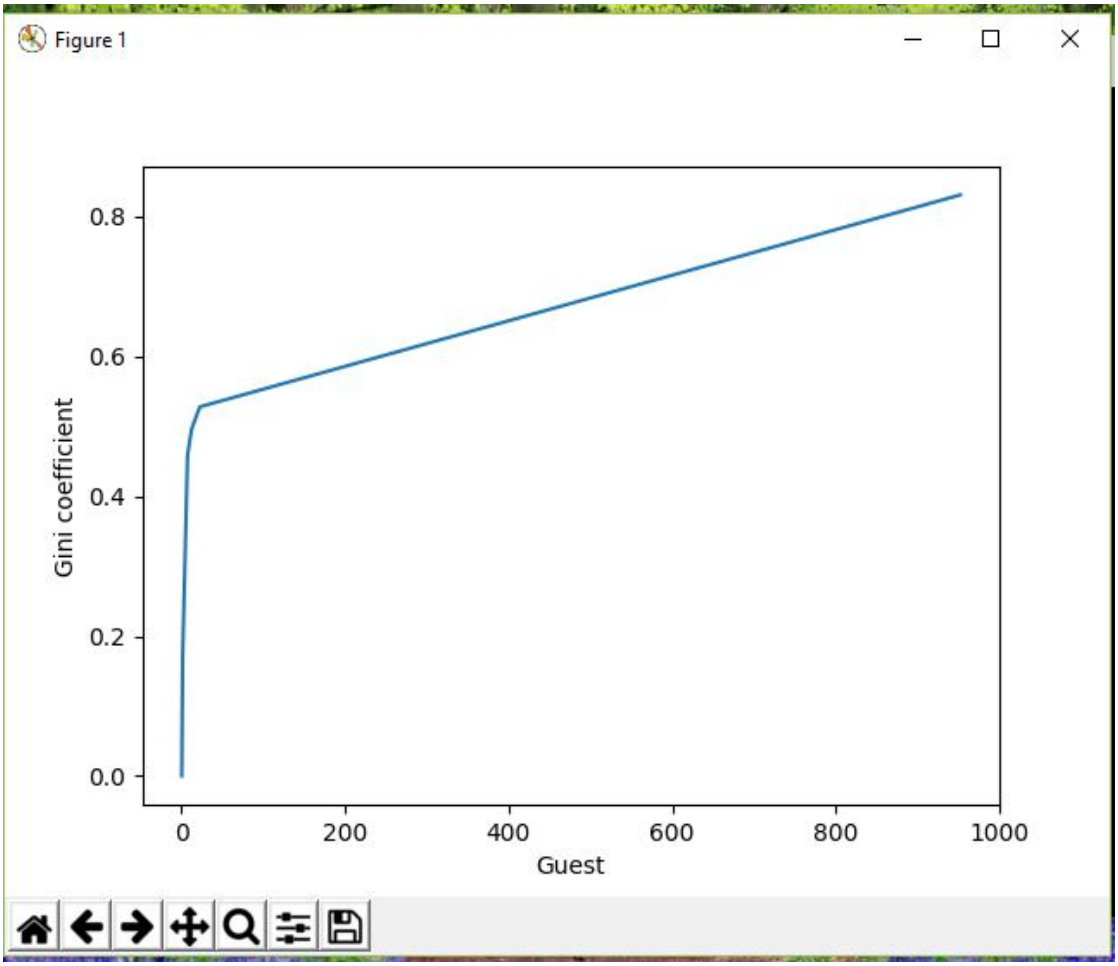


Figure 1
0.8
0.6
0.4
0.2
0.0
Gini coefficient
0
200
400
600
800
1000
Guest

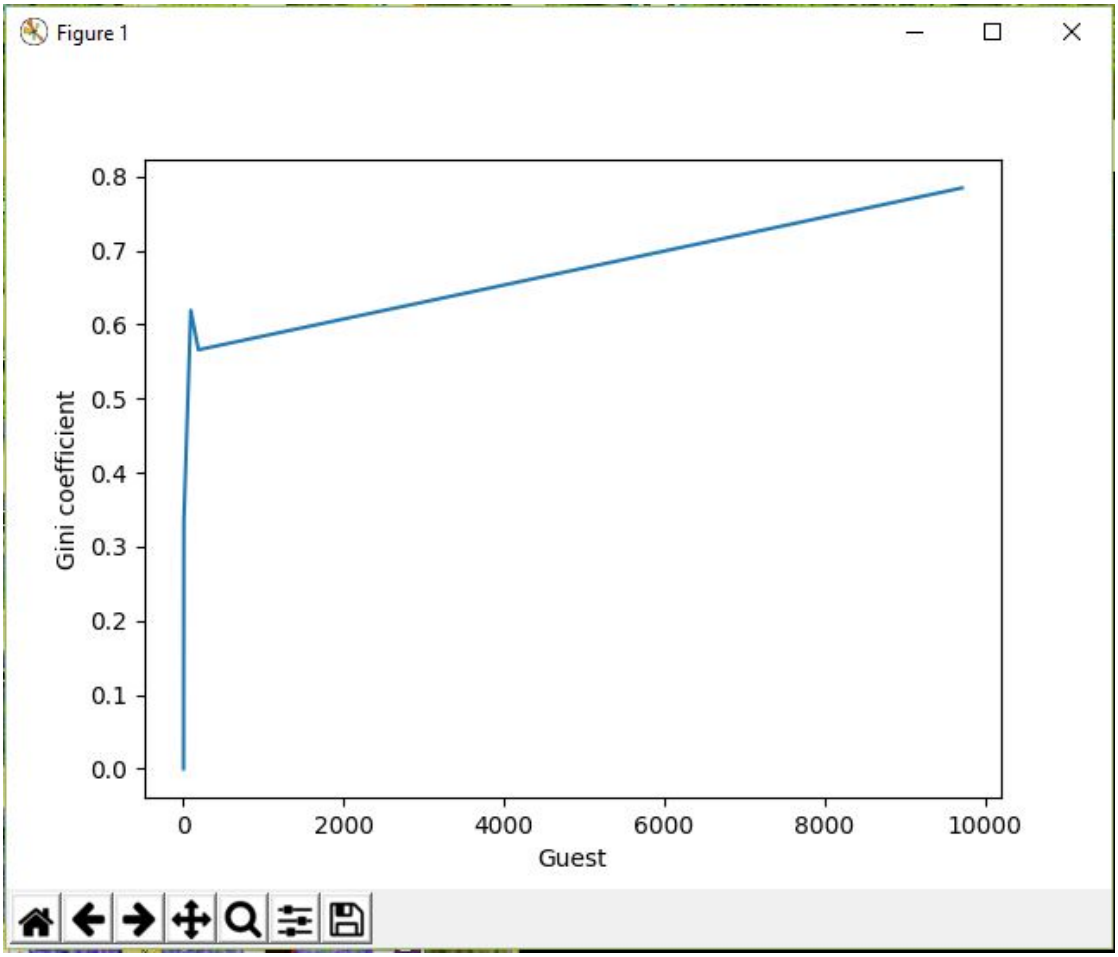


Figure 1
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Gini coefficient
0
2000
4000
6000
8000
10000
Guest

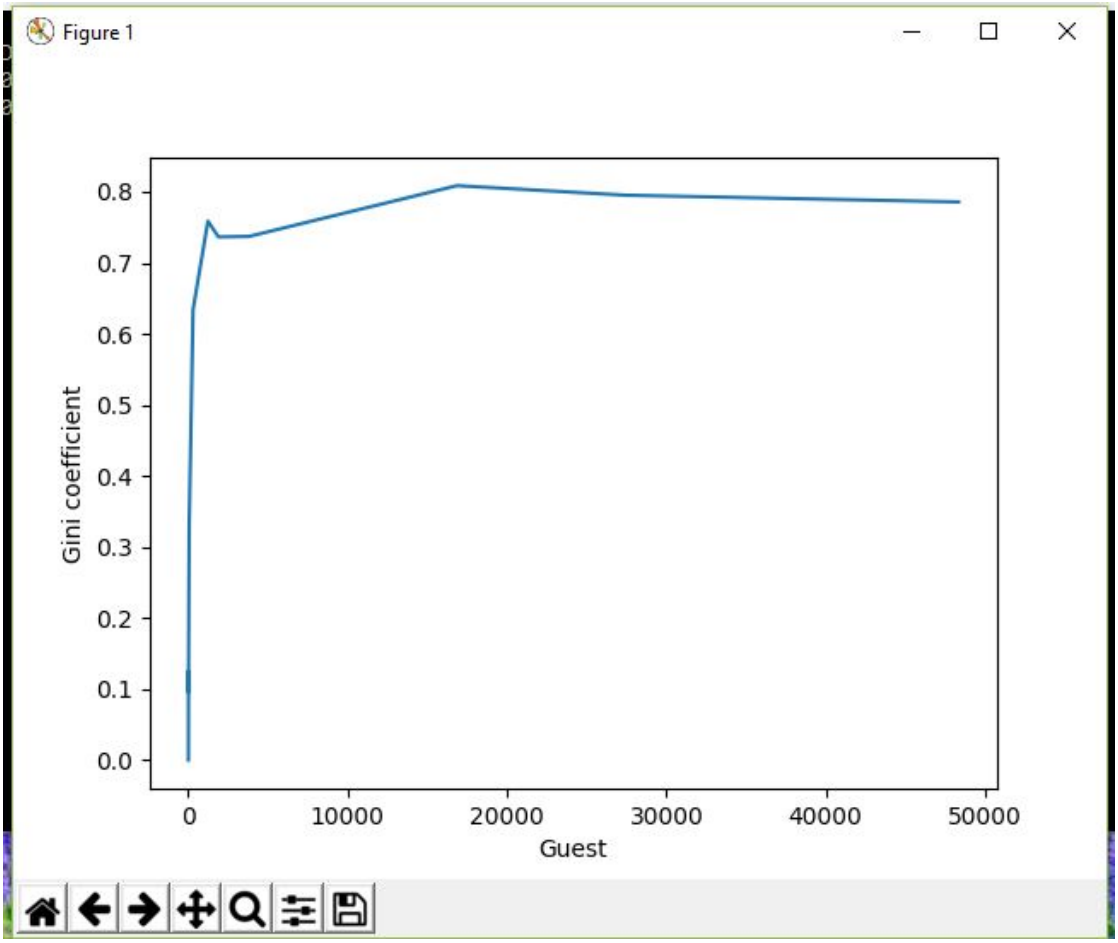


Figure 1
Gini coefficient
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
10000
20000
30000
40000
50000
Guest

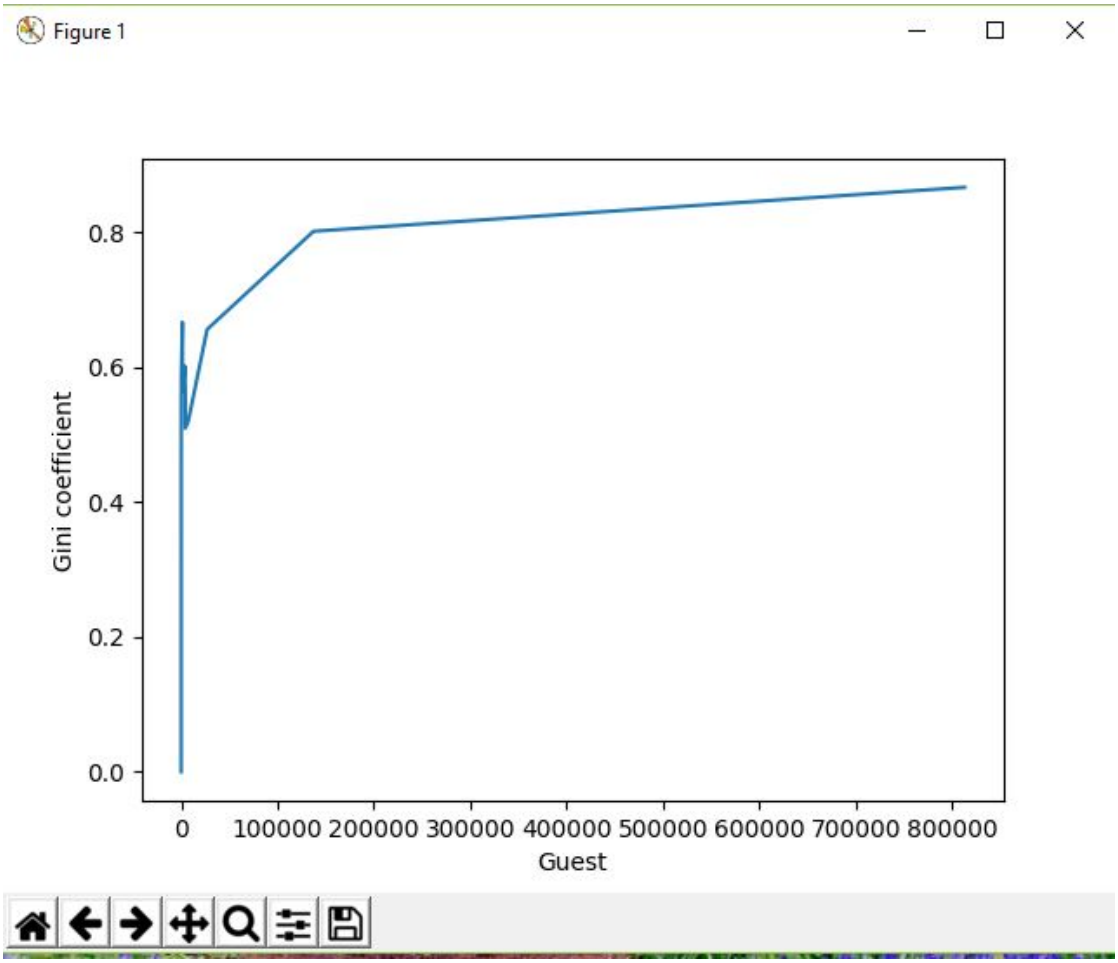


Figure 1
Gini coefficient
0.0
0.2
0.4
0.6
0.8
0
100000
200000
300000
400000
500000
600000
700000
800000
Guest

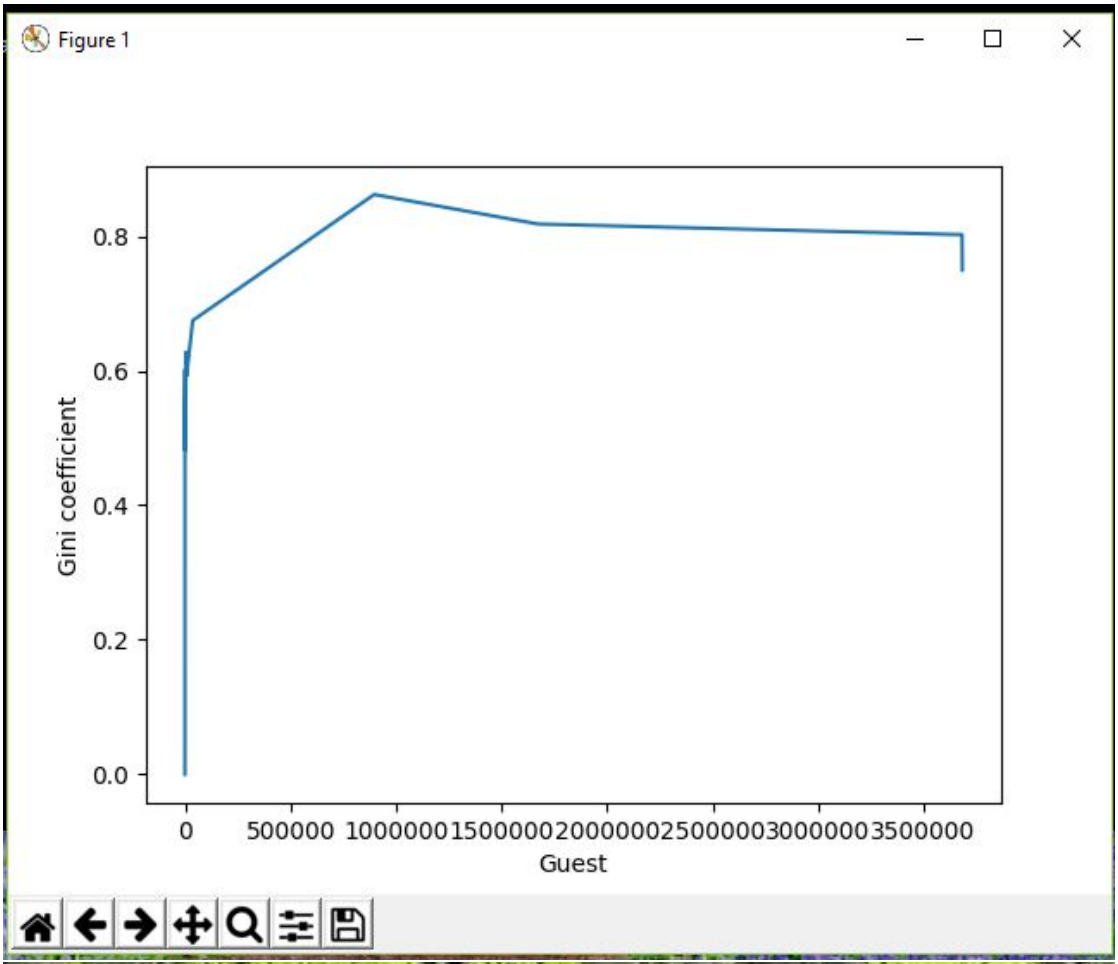
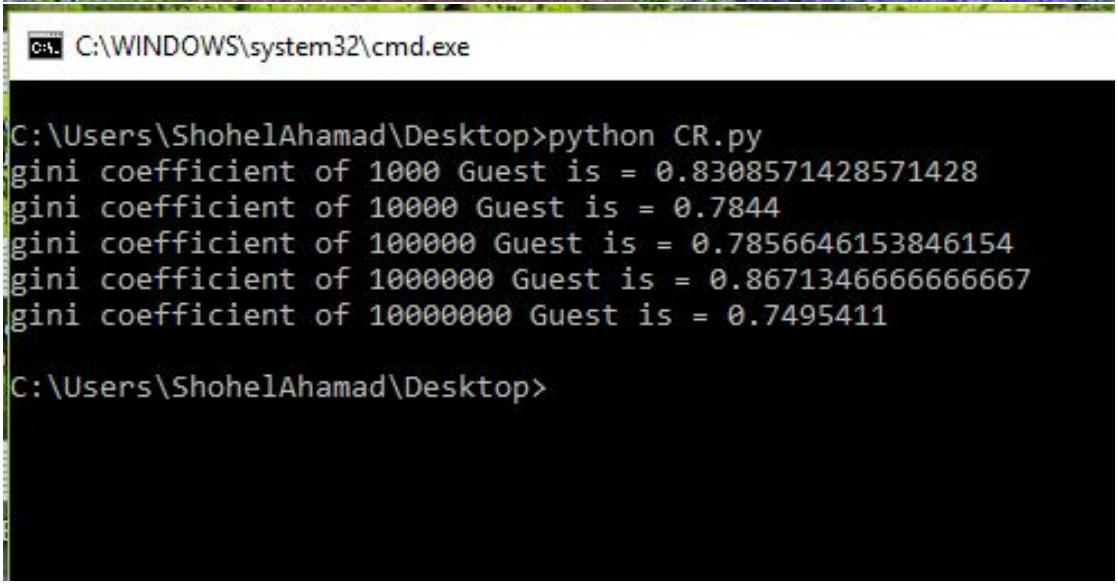


Figure 1
Gini coefficient
0.0
0.2
0.4
0.6
0.8
0
500000
1000000
1500000
2000000
2500000
3000000
3500000
Guest


C:\WINDOWS\system32\cmd.exe
C:\Users\ShohelAhamad\Desktop>python CR.py
gini coefficient of 1000 Guest is = 0.8308571428571428
gini coefficient of 10000 Guest is = 0.7844
gini coefficient of 100000 Guest is = 0.7856646153846154
gini coefficient of 1000000 Guest is = 0.8671346666666667
gini coefficient of 10000000 Guest is = 0.7495411
C:\Users\ShohelAhamad\Desktop>

3 Herding (10 points)

Let us consider the altitude of Koblenz to be 74 m above sea level. You are asked to figure out the height of the Ehrenbreitstein Fortress and the Fernmeldeturm Koblenz without googling.
The exercise is split in two parts:

Part 1 : The Secret
In *complete secrecy*, each member of the team will write down their estimated height of the Ehrenbreitstein Fortress without any form of discussion. Please keep in mind that you need to have reasons for your assumption. Once you are done, then openly discuss in the group and present you values in a tabulated format with the reasons each one assumed to arrive at that value.

Part II : The Discussion
Discuss amongst yourself with valid reasoning what could be the height of the Fernmeldeturm Koblenz. Only after discussing, each member of the group is asked to arrive at a value and present this value in a tabulated format as was done in Part I.

Calculate the Mean, Standard Deviation and Variance of your noted results for both the cases and explain briefly what you infer from it.

Note: This exercise is for you to understand the concepts of herding and not to get the perfect height by googling information. There is in fact no point associated with the height but with the complete reasoning that you provide for your answers.

Answer:

Part 01
Shohel:
Ehrenbreitstein : According to me the height would be like 60 meters from the ground and 134 meters from the sea level . This is because, I went there and it was not that high for me
Fernmeldeturm: About 400 m from the sea level. That is because, it is the highs tower in Koblenz and we can see it from anywhere in Koblenz.

Slobodan
Ehrenbreitstein : Maybe 80 meters , I have visited that place and I went at the top place and that is why I feel like it won't be more than 80 meters long
Fernmeldeturm: This is the tallest tower in the city and I think it would be about 450 meters.

Anish
Ehrenbreitstein : I think It would be around 100 meters high , I went to the top several times and what I can remember that it won't be more higher than 100 meters

Fernmeldeturm: That tower is the highest building in Koblenz and it seems like it would be like 350 meters.

Name	Ehrenbreitstein (Meters) as X	Fernmeldeturm (Meters) as Y
Shohel	60	300
Slobodan	80	320
Anish	100	350
	∑ = 240	∑=970

Part 2
Mean :
We know mean =

$$\sum X/n \tag{1}$$

(where n= number of examples)
Mean of Heights of Ehrenbreitstein =

$$\sum X/3 = 240/3 = 80. \tag{2}$$

Mean of Heights of Fernmeldeturm =

$$\sum Y/3 = 970/3 = 323.33 \tag{3}$$

Variance:
We know Variance is =

$$\sum (Xi - mean)^2 \tag{4}$$

Variance of Heights of Ehrenbreitstein =

$$((80-60)^2 + (80-80)^2 + (80-100)^2) = 800 \tag{5}$$

Variance of Heights of Fernmeldeturm =

$$((323.33-300)^2+(323.33-320)^2+(323.33-350)^2)/3 = (544.2889+11.0889+711.2889)/3 = 1266.67/3 = 422. \tag{6}$$

Standard Deviation: We know, Standard Deviation = square root of Variance
So, Standard Deviation of Heights of Ehrenbreitstein = 20.55
Standard Deviation of Heights of Fernmeldeturm = 35.59

Conclusion : According to our estimation about the heights of Ehrenbreitstein and heights of Fernmeldeturm all estimations were similar to each other's estimations. Moreover, the mean was also closed for all estimations for both cases. However, we also calculated variance and standard deviation for both cases according to our estimated heights and understand similarities and differences between all estimated values.

Important Notes

Submission

- Solutions have to be checked into the github repository. Use the directory name `groupname/assignment10/` in your group's repository.
- The name of the group and the names of all participating students must be listed on each submission.
- Solution format: all solutions as *one* PDF document. Programming code has to be submitted as Python code to the github repository. Upload *all* `.py` files of your program! Use `UTF-8` as the file encoding. *Other encodings will not be taken into account!*
- Check that your code compiles without errors.
- Make sure your code is formatted to be easy to read.
  - Make sure you code has consistent indentation.
  - Make sure you comment and document your code adequately in English.
  - Choose consistent and intuitive names for your identifiers.
- Do *not* use any accents, spaces or special characters in your filenames.

Acknowledgment

This latex template was created by Lukas Schmelzeisen for the tutorials of "Web Information Retrieval".

LaTeX

Currently the code can only be build using LuaLaTeX, so make sure you have that installed. If on Overleaf, there's an error, go to settings and change the LaTeXengine to `LuaLaTeX`.