Problem Set: Lesson 4.2

CDS 292. Fall, 2023

Instructions

In this document you will find a set of problems intended to be answered **by hand**, scanned and submitted to Blackboard. Make sure your document is well organized and contains readable responses to the questions. **Show your work! Full credit requires this. An answer without procedure is likely to elicit zero points.** There is one addendum to this handwritten rule: any required plot can be either scanned or sent as an additional file to your submission. If you choose to submit separate files with your plots, make sure they are labeled and can be easily identified.

Note: `Python` code questions will occur in problem sets. The answers have to be handwritten and must respect all `Python` syntax rules including proper indentation of code.

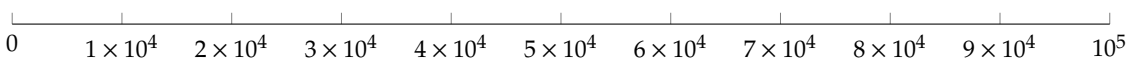
Total Points: 13

---

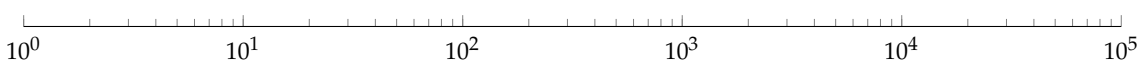
1. Consider the following set

$$S = \{32, 685, 801, 8, 3, 430, 97331, 1, 28, 4400\}$$

(a) *[1/2 point]* Place every element of S in a horizontal axis, using a linear scale. You can use the following axis as a guide.



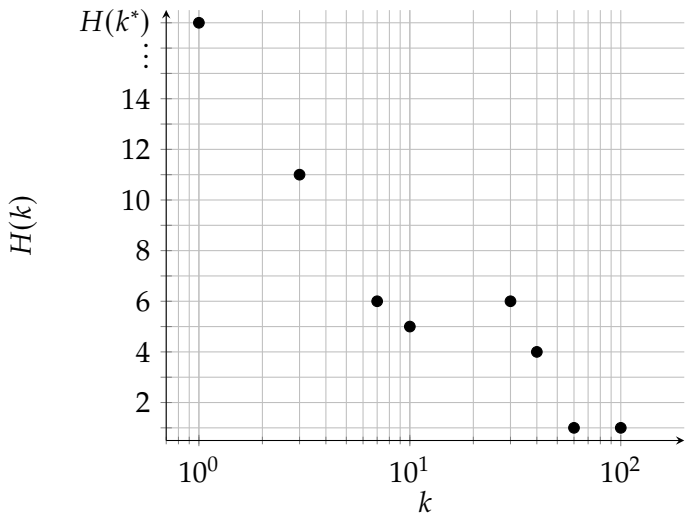
(b) *[1/2 point]* Using the set S again, place its elements in a horizontal axis, this time using a logarithmic scale. You can use the following guide.



(c) *[2 points]* In this last item, we want you to write a `Python` function to decide what scale to plot in. Note that some of the points in a are "invisible". Those same points *are* visible in b due to the difference in scale. We mentioned this effect in the slides: linear scales hide small numbers against the large ones, whereas logarithmic scales give those small numbers visibility. In the list of numbers above, the difference between 1 and 97331 is 4 orders of magnitude (if you write those two numbers using scientific notation, you will get $1 = 10^0$ and 9.7331×10^4; the difference in exponents between the numbers is the orders of magnitude separating them). It is recommended to plot using logarithmic scales when the minimum and maximum values are 2 or more orders of magnitude apart. Write `Python` code that takes a list of numbers as an input, and returns `True` if the plot should be in logarithmic scale, and `False` if the scale should be linear. Test the function on the list

of numbers for this problem. *HINT: figure out what mathematical function can give you the exponent associated with the value of a number in scientific notation. Then compare the exponents of your maximum and minimum numbers.*

2. Consider the degree histogram below. Note that $H(k^*)$ should be treated as a **variable** and the symbol $\vdots$ denotes that the values in the vertical axis less or equal than 14 are located **exactly** on visible markings of the plot. Any value above 14 on the vertical axis shows incomplete information, and thus, the markings on the plot should not be taken literally.



   (a) *[1 point]* Calculate n, m, ρ, and $\langle k \rangle$. *HINT: Your results should be a function of $H(k^*)$*

   (b) *[1 point]* What is the **minimum** value of $H(k^*)$?

   (c) *[1 point]* Using your result from part b, Calculate n, m, ρ, and $\langle k \rangle$.

3. In some degree distributions, $k_{\min}$ and $k_{\max}$ are more than 2 orders of magnitude apart. If that is the case, it is recommended to plot the horizontal axis (for k) using a log scale. The same applies for $H(k)$ where if $H(k^*)$ is 2 order of magnitude or more larger than the smallest $H(k)$ you should consider using log scale for the vertical ($H(k)$) axis. Thus, in principle, one can decide independently for each axis if it should be plotted in linear or log scale.

   (a) *[2 points]* Write a `Python` function to decide if the axes in a **degree histogram plot** should be scaled logarithmic or linearly. Try the function with with the degrees of the airports data (download the file `airline-net-und-2017.dat` and create the histogram using `Hdeg()`). *HINT: In order to test the function, create two separate lists with the same number of elements: one for all the k and another for all the $H(k)$.*

   (b) *[1 point]* Now, instead of testing each axis independently, write a single `Python` function that takes as input a degree histogram (in a dictionary), and outputs if *each* of the axes should be plotted using logarithmic or linear scale. Test the function with the airports network.

4. Network G has $n(G) = 1001$. Even though the degree histogram is unknown, there is a known approximate relationship between k and $H(k)$ given by

$$H(k) \approx 10^4 \times k^{-2}, \tag{2}$$

you also know that $H(k = 0) = 0$, and you should treat whatever value of $H(k) < 1$ as $H(k) = 0$.

(a) *[1 point]* What is the value of k^* *HINT: You don't have to calculate anything to get this value*

(b) *[1 point]* Provide an argument to justify that the relationship presented in Eq. 2 is *approximated* (as opposed to an *exact* relationship, i.e. with an equals symbol).

Use `Python` for the rest of this problem, assuming the relationship presented in Eq. 2 is **exact**, even if this will make the network impossible, because of the arguments you provided above

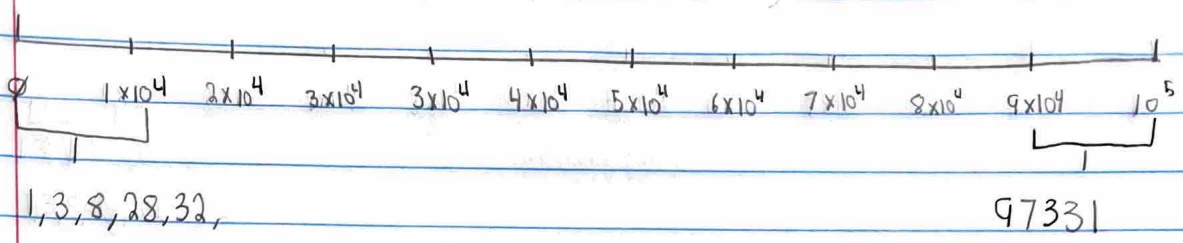
(c) *[1 point]* What is k_{min} and k_{max}?

(d) *[1 point]* Calculate m, $\langle k \rangle$, and ρ for network G. *HINT: You will not be able to obtain an integer for m. Assume it is the nearest integer to whatever value you obtain and use it for the other quantities.*

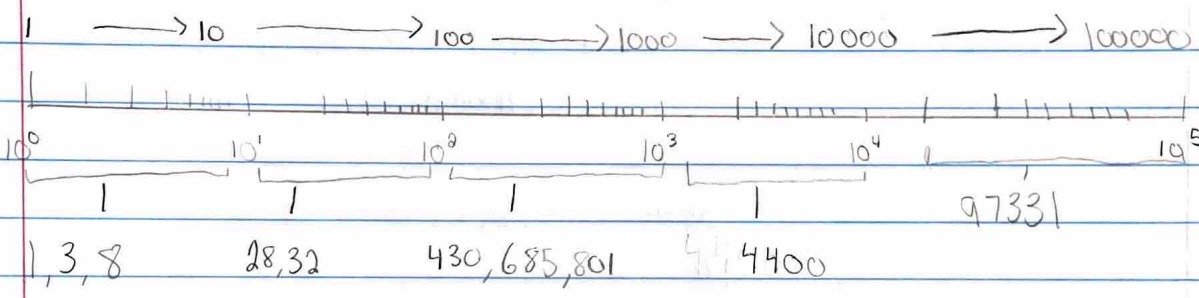
Problem Set, unit 4.2:

1.) $S = \{1, 3, 8, 28, 32, 430, 685, 801, 4400, 97331\}$

a.)



b.)



c.) → Write function, checking to use log or not. True for log, false for linear

→ Takes in a list, retrieve highest and lowest values from list.

→ Take the logarithm of the highest and lowest values to retrieve an exponent.

→ W

→

Problem Set, unit 4.2:

c.)
```
import math #Needed to use logarithm function.

def uselog(list_of_numbers):
#Retrieve lowest/highest values from list
    min_value = list_of_numbers[0]
    max_value = list_of_numbers[0]
#loop through elements
    for i in range(len(list_of_numbers)):
        if list_of_numbers[i] < min_value:
            min_value = list_of_numbers[i]
        if list_of_numbers[i] > max_value:
            max_value = list_of_numbers[i]
#Retrieve magnitudes of numbers, and find difference
    if min_value == 0:
        magnitude = math.log10(max_value)
    else:
        magnitude = math.log10(max_value) - math.log10(min_value)

#Check if magnitude difference is 2 or more
    return abs(magnitude) >= 2
```

2.)

a.) n, m, p, and $\langle k \rangle$ as function of $H(k^*)$.

n, the number of links in network. Sum of all frequencies of degrees.

In network, we have $k = 1, 3, 7, 10, 30, 40, 60, 100$

$H(k=1) = ?$, $H(k=3) = 11$, $H(k=7) = 6$, $H(k=10) = 5$, $H(k=30) = 6$, $H(k=40) = 4$, $H(k=60) = 1$, $H(k=100) = 1$

$n = H(k^*) + 11 + 6 + 5 + 6 + 4 + 1 + 1$

$\boxed{n = H(k^*) + 34}$

Problem Set, unit 4.2:

a.) m = number of links in network

K tells us the amount of links nodes have in a network.

$H(K)$ tells us how many times those number of links occur for a network.

want to find the total number of links in network.

$1 \times H(K^*) + (3 \times 11) + (7 \times 6) + (10 \times 5) + (30 \times 6) + (40 \times 4) + (60 \times 1) + (100 \times 1) \cdot \frac{1}{2}$

$[1 \cdot H(K^*)] + (33) + (42) + (50) + (180) + (160) + (60) + (100) \cdot \frac{1}{2}$

$[1 \cdot H(K^*)] + 75 + 230 + 320 = 625$

$[H(K^*)] + 625$, $\boxed{m = (625 + H(K^*)) \cdot \frac{1}{2}}$ $\times \frac{1}{2}$ because link connects 2 nodes.

$\langle K \rangle$, average number of links a node has in network

$\langle K \rangle = 2m/n$

$$\boxed{\langle K \rangle = \frac{(2) \cdot [(625 + H(K^*)) \cdot \frac{1}{2})]}{(H(K^*) + 34)}}$$

P, density, chance that 2 randomly selected nodes are linked together

$p = \langle K \rangle / (n-1)$

$$\boxed{p = \frac{2 \cdot [(625 + H(K^*)) \cdot \frac{1}{2}]}{(H(K^*) + 34)} \Big/ [H(K^*) + 34] - 1}$$

b.) minimum value of $H(k^*)$?, $H(k^*)$ must be at least what value?
What is the minimum frequency/number of nodes in network G, that have only 1 link?
In network, we know that the total number of nodes is:
$H(k^*) + 34$.
Max number of links a node can have is $n-1$.
Highest degree in our network is 100, therefore $\overbrace{H(k^*)}^{n} + 34 \geq 101$
or "$n \geq 101$"

$$-34 \quad -34$$

$$\boxed{H(k^*) \geq 67}$$

c.) $H(k=1) \geq 67$ nodes
$H(k=3) = 11$ nodes
$H(k=7) = 6$ nodes
$H(k=10) = 5$ nodes
$H(k=30) = 6$ nodes
$H(k=40) = 4$ nodes
$H(k=60) = 1$ node
$H(k=100) = 1$ node

Network G,

3 links x 11
7 links x 6
10 links x 5
30 links x 6
40 links x 4
60 links x 1
100 links x 1

34 nodes accounted for in network.

+

1 link $H(1)$

Network of 3 nodes $6/2 = 3$
4 nodes 6

$n-1$ max links for a node in any network
histogram tells us that highest number of links in this network is 100 links.
We know the number of nodes is at least 101, could be more
$101 \cdot 100 / 2 = 5{,}050$ max links in network if $n = 101$

→

Problem Set, unit 4.2

c.) num nodes.

Given: ① ② ③ ... ⑤ ⑥ ⑦ ⑧ > Already 34 nodes in network

+ more nodes from $\boxed{H(k=1)}$. So total number of nodes will be given by $\underline{34 + H(k=1)}$

$$\boxed{\underset{\text{Original}}{34} + \underset{\substack{H(k=1) \\ \text{Not given.}}}{?}}$$

→ N in network. We ALSO know the highest number of links that a node has in this network. Significant clue, because in ANY network, the max # of links for a single node is ALWAYS "total number of nodes - 1". We can extract the # from this rule. The network tells us that, our highest degree is 100 links. Therefore, there has to be minimum 101 nodes in our network.

$$\underset{\text{original nodes}}{34} + \underset{\substack{\text{67 nodes for "1" link} \\ \text{would give us 101 total nodes.}}}{67} = 101$$

At least 101 nodes in this network.

$\langle K \rangle / (n-1)$

$\boxed{n \geq 101}$

m, num links in network

$625 + 67 = 692/2 = 346$

$346 / 5{,}050 = 0.07$

$\boxed{m \geq 346}$

$\langle K \rangle = 2m/n =$

$100 \cdot 101 / 2 =$

$\boxed{\langle K \rangle \geq \approx 6.85}$

$\langle K \rangle = 346 \cdot 2 / 101 = 6.85$

$\boxed{p \geq 0.07}$

$6.85 / 100$

Problem Set, unit 4.2

3.) k_{min} / k_{max} sometimes more than 2 order of magnitude's apart.
Then use logarithmic scale to plot horizontal axis.
Same for $H(k)$ - y axis - when k^* is 2 or more orders of magnitude apart from smallest frequency.

a.)
```
#Write a function, to see if axes in a degree histogram
#plot should use linear (<2 orders of magnitude) or
#logarithmic (≥2 orders of magnitude apart) scale. Try
#with degrees of "airport data"
import math
from Readlist import readlist
import HKAnalysis as HK

def log-or-lin(x, y):
    x_min = x[0]
    x_max = x[0]
    for i in range(len(x)):
        if x[i] < x_min:
            x_min = x[i]
        if x[i] > x_max:
            x_max = x[i]

    y_min = y[0]
    y_max = y[0]
    for i in range(len(y)):
        if y[i] < y_min:
            y_min = y[i]
        if y[i] > y_max:
            y_max = y[i]
```
→

Problem Set, unit 4.2

a.) .....

```
mag_x = math.log10(x_max) - math.log10(x_min)
mag_y = math.log10(y_max) - math.log10(y_min)

if mag_x >= 2 and mag_y >= 2:
    print("Use log scale for both axes.")

elif mag_x >= 2 and mag_y < 2:
    print("Use log scale for x, and linear for y")

elif mag_x < 2 and mag_y >= 2:
    print("Use linear scale for x, and log for y")

else:
    print("Use linear scale for both axes")

# test function
G = readlist("Airports.txt", ∅)
GH = HK.Hdeg(G)
K = list(GH.keys())
hK = list(GH.values())
log_or_lin(K, hK)
```

↳ Output: Use log scale for both axes

Problem Set, unit 4.2

b.) # Write python function; takes in a degree histogram
Outputs if each of the axes should be plotted using
log or linear scale.

```
def log_or_lin_2 (histogram):
    k = list(histogram.keys())
    hk = list(histogram.values())
    k_max = max(k)
    k_min = min(k)
    hk_max = max(hk)
    hk_min = min(hk)
    mag_k = math.log10(k_max) - math.log10(k_min)
    mag_hk = math.log10(hk_max) - math.log10(hk_min)

    if mag_k >= 2:
        print("use log scale for x-axis")
    else:
        print("Use linear scale for x-axis")

    if mag_hk >= 2:
        print("Use log scale for y-axis")
    else:
        print("use linear scale for y-axis").
```

Problem Set, unit 4.2

4.) Network, G:

We know that there are 1,001 nodes. Max links for a node is 1,000.

Relationship exists between a node's degree and its frequency in the network:

$$H(k) \cong 10^4 \times k^{-2}$$

$$H(1) = 10{,}000 \times 1^{-2} = 10{,}000$$

$$H(2) = 10{,}000 \times 2^{-2} = 2{,}500$$

$\downarrow$

gets progressively smaller as the degree goes up

We also know that $H(k=0) = 0$, so zero nodes in the network have no links. Therefore, every node in network G has at least 1 link.

a.) $\boxed{k^* = 1}$

b.) Provide argument, saying why $10^4 \times k^{-2}$ is justified as an equation to give an "approximation" of a degrees frequency, rather than an "exact value".

$1{,}001 \cdot 1000 / 2 = 500{,}500$ max links

know that every node has at least 1 link.

First of all, we don't have the degree histogram, so there's no way of knowing the exact value for any $H(k)$ in the network. What we do know is "n", which is 1001. From this we can extract $m_{complete}$, which is 500,500. This means that we can not have all nodes have "n-1" links. It would violate $m_{complete}$, so as the degrees go up, the frequencies have to go down, since there are 1,001 nodes in G. Again, we can't give exact values, only approximations since the histogram is not given

Problem Set, unit 4.2

c.) Use python, assuming $H(k) = 10^4 \times k^{-2}$.

What is the lowest degree? And what is the highest degree?

No nodes have $k = \emptyset$
$H(1) = 10^4 \times 1^{-2} = 10{,}000$ nodes with 1 link
$H(2) = 10^4 \times 2^{-2} = 2{,}500$ nodes with 2 links
$H(3) = 10^4 \times 3^{-2} = 1{,}111$ nodes with 3 links

$\boxed{k_{min} = 1, \; k_{max} = 100}$

Doesn't say to include Python Code?

d.) $m = 25936$, approximately
$\langle k \rangle = 51.8$, approximately
$p = 0.03$, approximately

```
def calc(k):
    return 10**4 * k**(-2)

m = sum([k * calc(k) for k in range(1, k_max +1])/2
avg_k = (2*m)/n
p = m/n*(n-1)
```

n = 1001, k_max = 100, k_min = 1