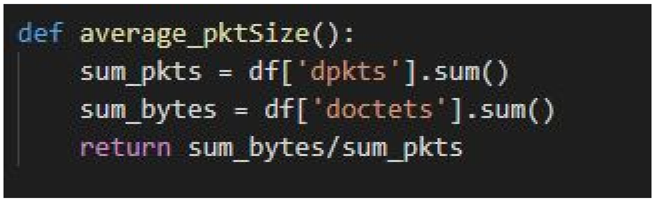
Assignment 5

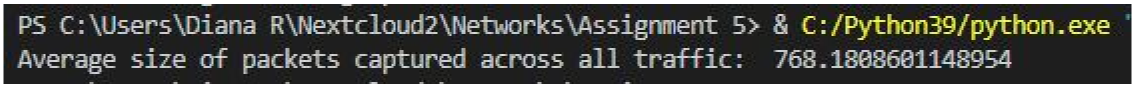
a) What is the average size of the packets across all the traffic captured in the dataset? Provide the description on how you calculated this number.

The average size of all the packets was calculated using the *average_pktSize()* method. This method sums all the packets in the table, and all the bytes in the table. Using these two values, I was able to calculate the average size of a packet by dividing the total sum of all the bytes by the packets.As shown below.



```
def average_pktSize():
    sum_pkts = df['dpkts'].sum()
    sum_bytes = df['doctets'].sum()
    return sum_bytes/sum_pkts
```

Here is the output:



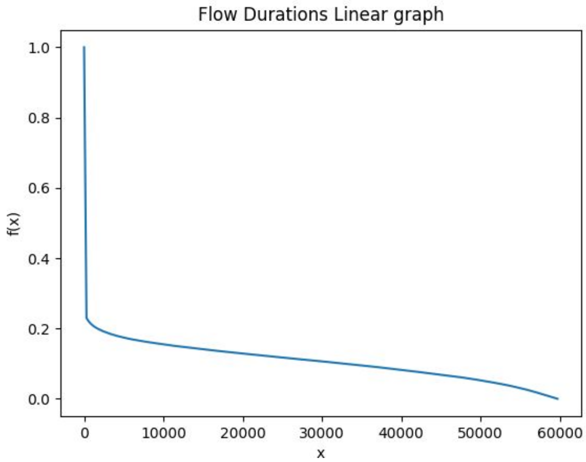
```
PS C:\Users\Diana R\Nextcloud2\Networks\Assignment 5> & C:/Python39/python.exe
Average size of packets captured across all traffic:  768.1808601148954
```

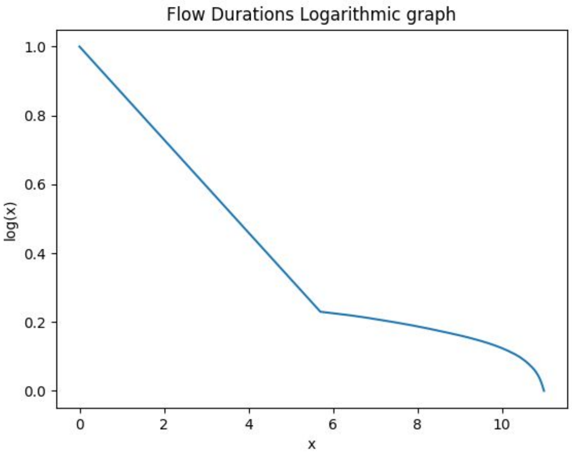
According to the calculation, the average size of a packet is about 768 bytes.

b) Plot the Complementary Cumulative Probability Distribution (CCDF) of flow durations (i.e., the finish time minus the start time) and of flow sizes (i.e., number of bytes, and number of packets).

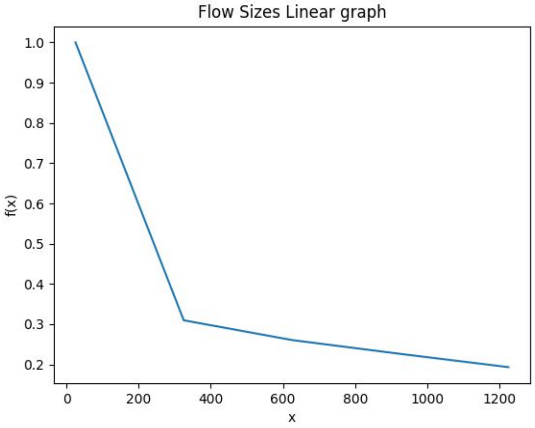
i. First plot each graph with a linear scale on each axis, and then a second time with a logarithmic scale on each axis.

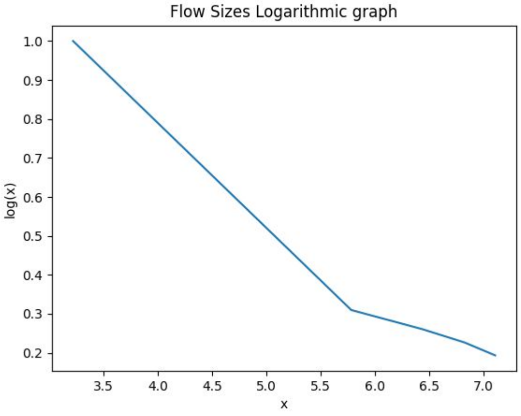
The flow duration graph was calculated using the difference between the time a flow started until its end from each row in the table.





The flow sizes duration graph was calculated using the average of bytes over the packets in the table.





ii. What are the main features of the graphs?

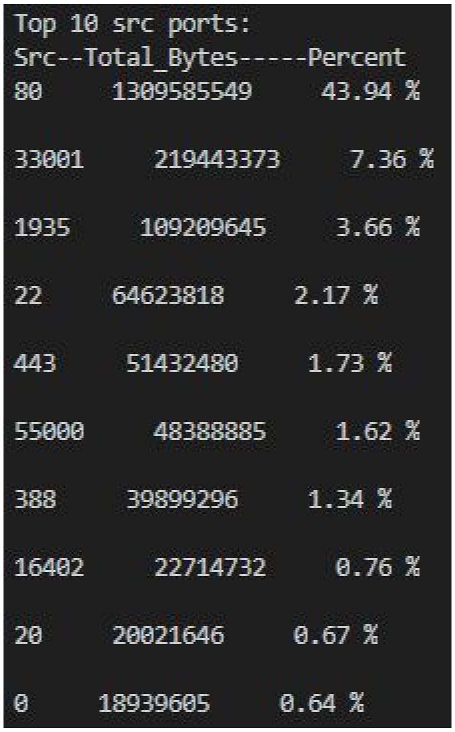
The linear graphs start with a steep linear drop, then do a drastic change and return to a more linear function drop. The logarithmic graphs do not show any steep drops, but instead follow a more reasonable drop in both flow sizes and flow durations.

iii. Why is it useful to plot on a logarithmic scale?

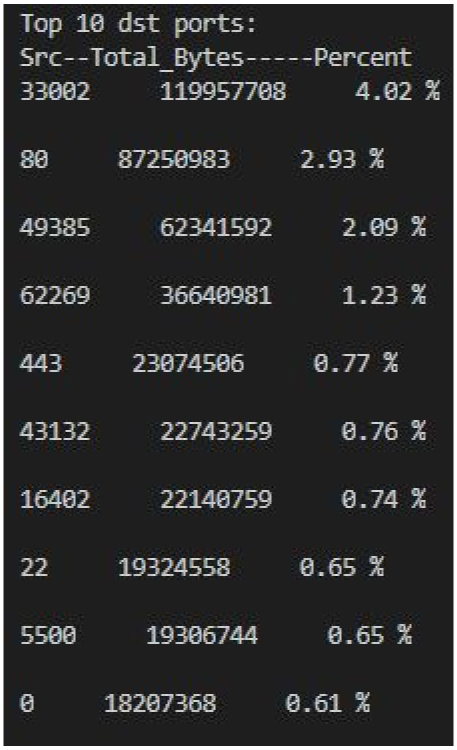
Like I mentioned before, the linear graphs show extreme steep drops, while the logarithmic graphs show a more reasonable relationship while not necessarily maintaining a linear relationship. Which could happen even if it is not shown in the linear graphs.

c) Summarize what kind of traffic is going through this router the most.

i. Create two tables, listing the top-ten port numbers by sender traffic volume and by receiver traffic volume including the percentage of traffic (by bytes) they contribute.



```
Top 10 src ports:
Src--Total_Bytes-----Percent
80     1309585549     43.94 %

33001     219443373     7.36 %

1935     109209645     3.66 %

22     64623818     2.17 %

443     51432480     1.73 %

55000     48388885     1.62 %

388     39899296     1.34 %

16402     22714732     0.76 %

20     20021646     0.67 %

0     18939605     0.64 %
```



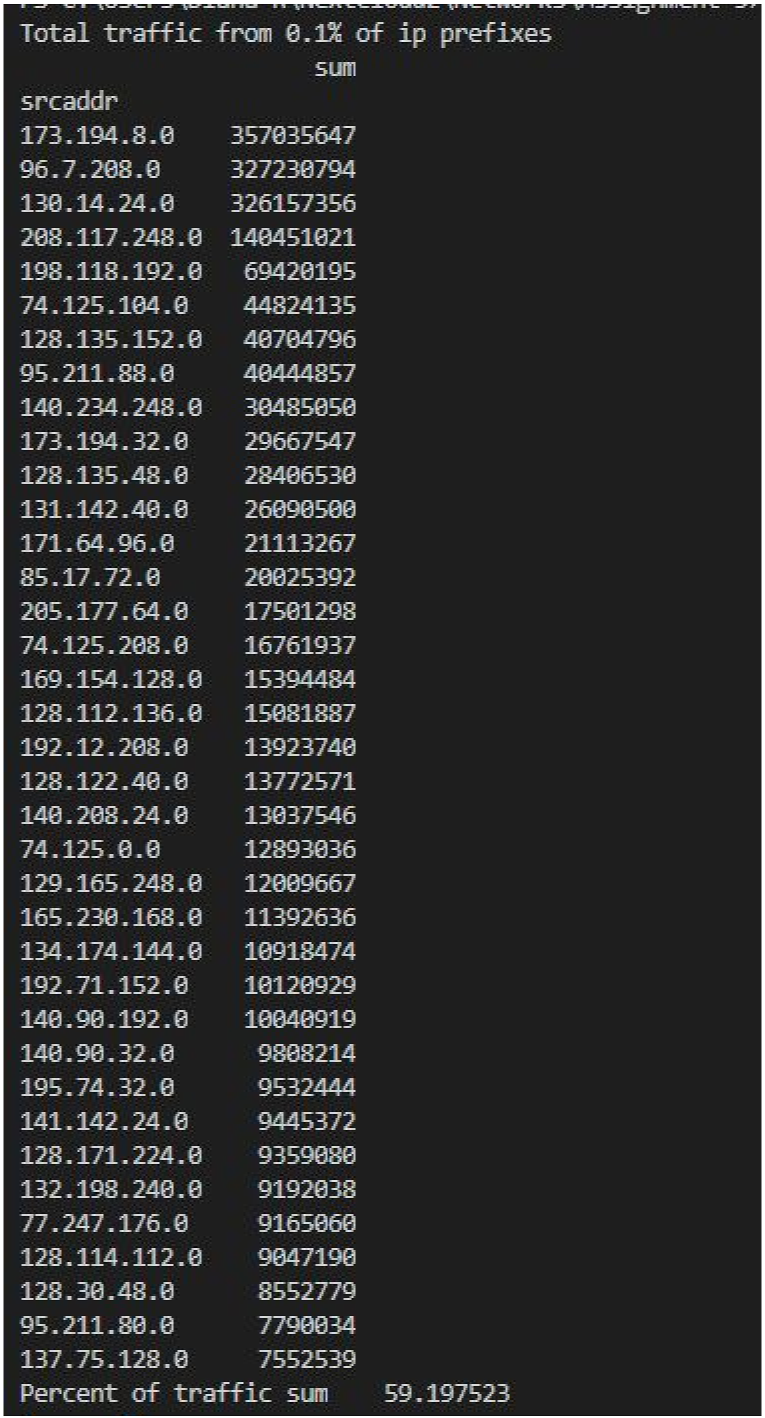
```
Top 10 dst ports:
Src--Total_Bytes-----Percent
33002     119957708     4.02 %

80     87250983     2.93 %

49385     62341592     2.09 %

62269     36640981     1.23 %

443     23074506     0.77 %

43132     22743259     0.76 %

16402     22140759     0.74 %

22     19324558     0.65 %

5500     19306744     0.65 %

0     18207368     0.61 %
```

Above are the top most used source and destination ports of the traffic.
ii. Explain what applications are likely be responsible for this traffic. (See the IANA port numbers reference for details.) Explain any significant differences between the results for sender vs. receiver port numbers.
Based on the tables, I concluded that the most traffic comes from visiting web pages using port 80(there is also traffic from 443 port, but that one is far less but it is usually regenerated from web pages as well). Based on the destination ports, there is some kind of service using TCP or UDP to communicate through port 33001 or 33002, it could be two separate services or the same channel. There also seems to be a ssh connection of some kind, using port 22.

d) Summarize the traffic volumes based on the source IP prefix.
i. What fraction of the total traffic comes from the most popular 0.1% of source IP prefixes? (count by number of bytes)

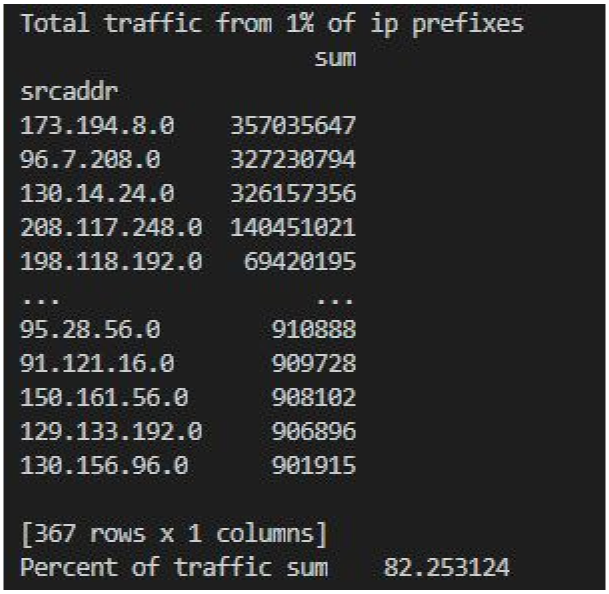
The following picture shows the traffic per IP. I defined the *traffic_fromIP* method in order to calculate the traffic from the top X ips. This method takes in the dataframe, the percentage desired, and the row which we are interested in. In this case since we want to find the traffic coming for X ips, I used the srcaddr row.
For this question, I decided to not only include the fraction, but also the src addresses and the number of bytes they contribute (sum column).

**For the rest of the tables, the rows got larger and I only included the head and tail of the list.



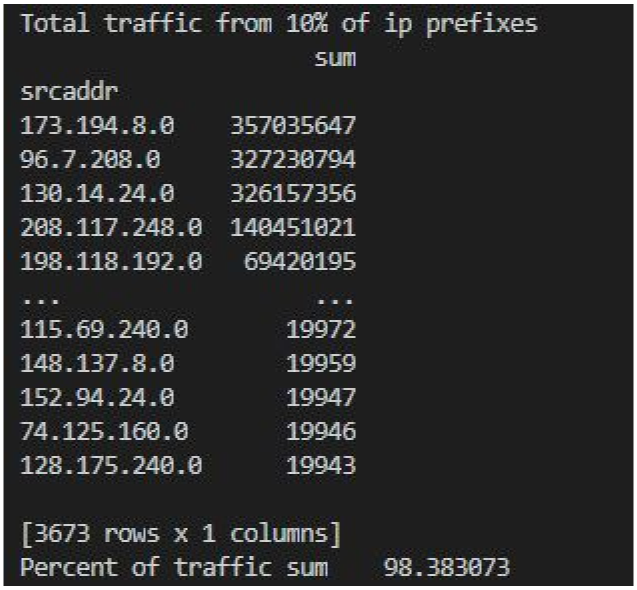
```
Total traffic from 0.1% of ip prefixes
                      sum
srcaddr
173.194.8.0     357035647
96.7.208.0      327230794
130.14.24.0     326157356
208.117.248.0   140451021
198.118.192.0    69420195
74.125.104.0     44824135
128.135.152.0    40704796
95.211.88.0      40444857
140.234.248.0    30485050
173.194.32.0     29667547
128.135.48.0     28406530
131.142.40.0     26090500
171.64.96.0      21113267
85.17.72.0       20025392
205.177.64.0     17501298
74.125.208.0     16761937
169.154.128.0    15394484
128.112.136.0    15081887
192.12.208.0     13923740
128.122.40.0     13772571
140.208.24.0     13037546
74.125.0.0       12893036
129.165.248.0    12009667
165.230.168.0    11392636
134.174.144.0    10918474
192.71.152.0     10120929
140.90.192.0     10040919
140.90.32.0       9808214
195.74.32.0       9532444
141.142.24.0      9445372
128.171.224.0     9359080
132.198.240.0     9192038
77.247.176.0      9165060
128.114.112.0     9047190
128.30.48.0       8552779
95.211.80.0       7790034
137.75.128.0      7552539
Percent of traffic sum    59.197523
```

ii. What fraction of the total traffic comes from the most popular 1% of source IP prefixes?



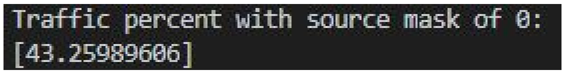
```
Total traffic from 1% of ip prefixes
                    sum
srcaddr
173.194.8.0    357035647
96.7.208.0     327230794
130.14.24.0    326157356
208.117.248.0  140451021
198.118.192.0   69420195
...                  ...
95.28.56.0        910888
91.121.16.0       909728
150.161.56.0      908102
129.133.192.0     906896
130.156.96.0      901915

[367 rows x 1 columns]
Percent of traffic sum    82.253124
```

iii. What fraction of the total traffic comes from the most popular 10% of source IP prefixes?



```
Total traffic from 10% of ip prefixes
                    sum
srcaddr
173.194.8.0    357035647
96.7.208.0     327230794
130.14.24.0    326157356
208.117.248.0  140451021
198.118.192.0   69420195
...                  ...
115.69.240.0       19972
148.137.8.0        19959
152.94.24.0        19947
74.125.160.0       19946
128.175.240.0      19943

[3673 rows x 1 columns]
Percent of traffic sum    98.383073
```

iv. Some flows will have a source mask length of 0. What fraction of traffic (by bytes) that has a source mask of 0? To calculate this, I defined the *maskZero_traffic()* method. This method returns the traffic percent of the traffic with mask length of zero by grouping the traffic by src_mask, and then using the top row to calculate the value.



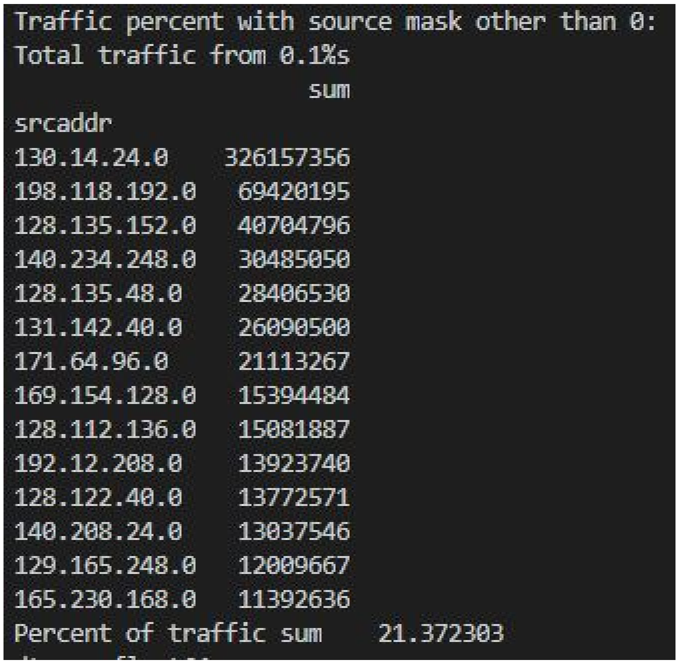
```
Traffic percent with source mask of 0:
[43.25989606]
```

Around 43%

v. Now, exclude this traffic (mask=0) from the rest of the analysis and answer d(i), d(ii), and d(iii).

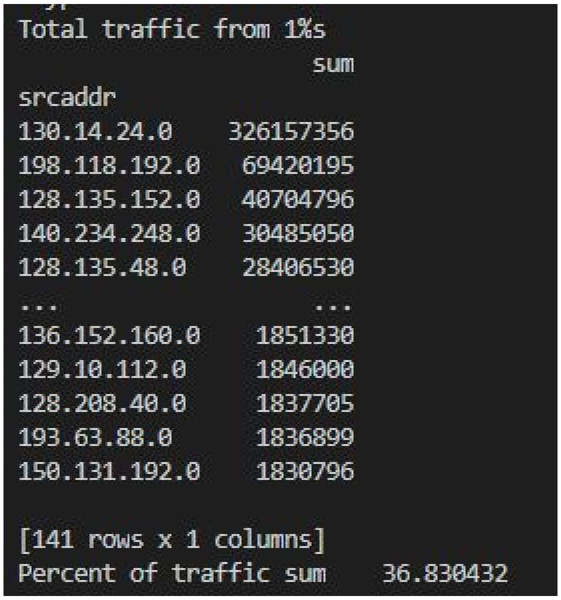
For this question, I used the *maskNonZero_traffic()* method to exclude the sources with source mask of zero, and then using the *traffic_fromIP()* method previously used to calculate the traffic percentage.

For 0.1%:



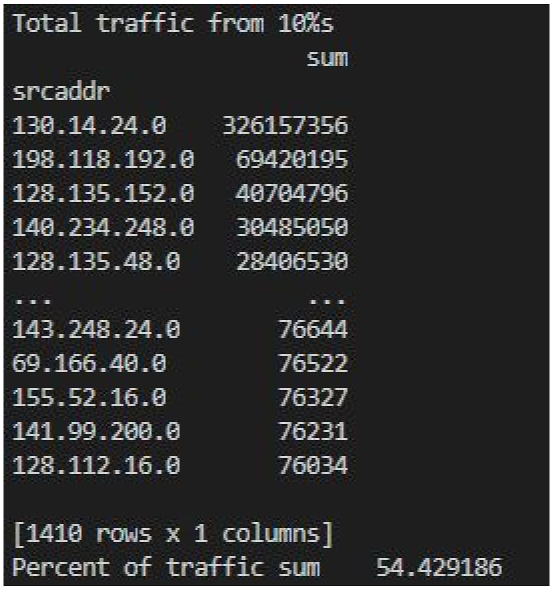
```
Traffic percent with source mask other than 0:
Total traffic from 0.1%s
                     sum
srcaddr
130.14.24.0    326157356
198.118.192.0   69420195
128.135.152.0   40704796
140.234.248.0   30485050
128.135.48.0    28406530
131.142.40.0    26090500
171.64.96.0     21113267
169.154.128.0   15394484
128.112.136.0   15081887
192.12.208.0    13923740
128.122.40.0    13772571
140.208.24.0    13037546
129.165.248.0   12009667
165.230.168.0   11392636
Percent of traffic sum    21.372303
```

For 1%:



```
Total traffic from 1%s
                     sum
srcaddr
130.14.24.0    326157356
198.118.192.0   69420195
128.135.152.0   40704796
140.234.248.0   30485050
128.135.48.0    28406530
...                  ...
136.152.160.0    1851330
129.10.112.0     1846000
128.208.40.0     1837705
193.63.88.0      1836899
150.131.192.0    1830796

[141 rows x 1 columns]
Percent of traffic sum    36.830432
```

For 10%:

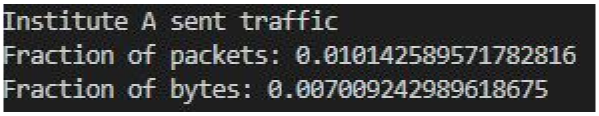


```
Total traffic from 10%s
                       sum
srcaddr
130.14.24.0    326157356
198.118.192.0   69420195
128.135.152.0   40704796
140.234.248.0   30485050
128.135.48.0    28406530
...                  ...
143.248.24.0       76644
69.166.40.0        76522
155.52.16.0        76327
141.99.200.0       76231
128.112.16.0       76034

[1410 rows x 1 columns]
Percent of traffic sum    54.429186
```

e) Assume an Institute-A has the 128.112.0.0/16 address block.

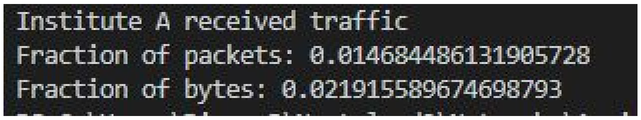
For this last question I used the *instituteA_addrBlock()* method to calculate the traffic sent and received by A. This method receives as parameters the dataframe and the column we are interested in. For the case of question i, we are interested in the 'srcaddr' column, since the ips between 128.112.0.0 and 128.112.255.255 should be the source of the traffic sent by A. For the second question,I used the 'dstaddr' column, since this indicated the traffic received/sent to A.

i. What fraction of the traffic in the dataset is sent by A? Measure both in terms of bytes and packets.



```
Institute A sent traffic
Fraction of packets: 0.010142589571782816
Fraction of bytes: 0.007009242989618675
```

ii. What fraction of the traffic in the dataset is sent to A? Measure both in terms of bytes and packets.



```
Institute A received traffic
Fraction of packets: 0.014684486131905728
Fraction of bytes: 0.021915589674698793
```

References:

https://pandas.pydata.org/docs/user_guide/index.html#user-guide

https://notebook.community/hanhanwu/Hanhan_Data_Science_Practice/Applied_Statistics/thinkstats_chapter4