

assignment1

October 5, 2021

```
[37]: import numpy as np
import matplotlib.pyplot as plt
import networkx as nx
import seaborn as sns
import pandas as pd
import csv
```

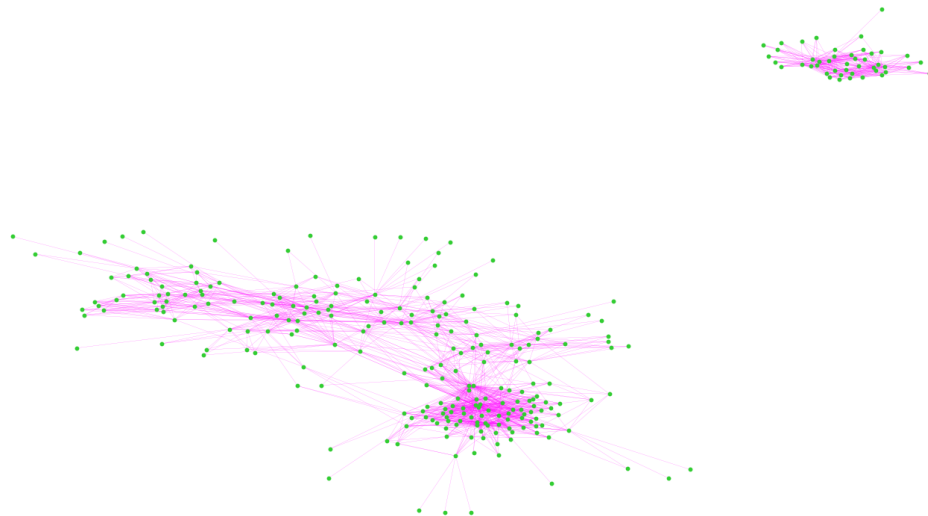
```
[38]: G = nx.Graph()
bipartite_cluster_of_bssid = []
with open('measurement.csv') as measurement_file:
    reader = csv.reader(measurement_file, delimiter=';')
    next(reader)
    for row in reader:
        bssid = row[2]
        bipartite_cluster_of_bssid.append(bssid)
        date = row[7]
        G.add_edge(date, bssid)
bipartite_cluster_of_bssid = list(set(bipartite_cluster_of_bssid))
```

```
[39]: # bipartite layout:
# one of the bipartite clusters needs to be provided
# graph_positioning = nx.bipartite_layout(G, bipartite_cluster_of_bssid)
# randoma layout:
# graph_positioning = nx.random_layout(G)
# spring layout:
# graph_positioning = nx.spring_layout(G)
# circular:
# graph_positioning = nx.circular_layout(G)
# graph_positioning = nx.fruchterman_reingold_layout(G)
# spectral: (bad for bipartite graphs)
# graph_positioning = nx.spectral_layout(G)

graph_positioning = nx.spring_layout(G)
plt.figure(figsize=(19.2, 10.8), dpi=100)
plt.title("Title")
nx.draw(G, pos=graph_positioning, node_size=10, width=0.1,
        node_color="limegreen", edge_color="magenta")
```

```
# plt.show()
# saving file:
# plt.savefig("filename.png")
# plt.clf()
# plt.close()
```

Title



```
[43]: paths = dict(nx.all_pairs_shortest_path(G))
```

```
[47]: list_of_distances = []
for node1 in G:
    for node2 in G:
        if node1 == node2:
            continue
        if node2 in paths[node1].keys():
            distance = len(paths[node1][node2]) - 1
            list_of_distances.append(distance)
```

```
[59]: a = np.min(list_of_distances)
b = np.max(list_of_distances)
b
```

```
[59]: 7
```

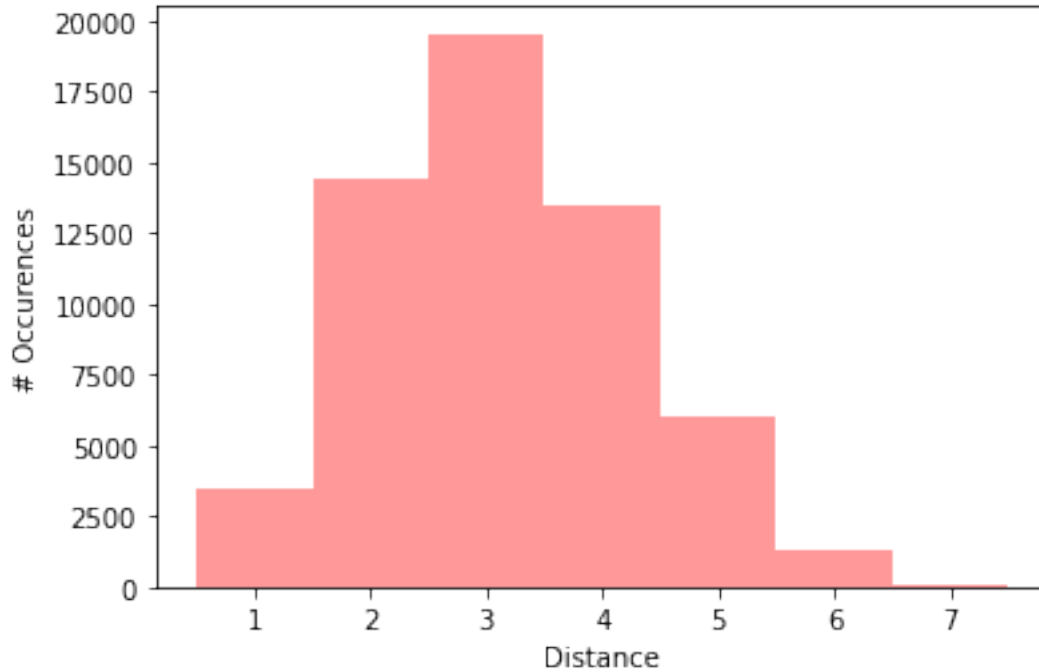
```
[69]: custom_range = np.arange(a-0.5, b+2-0.5, 1)
print(custom_range)
```

```
#sns.distplot(list_of_distances, kde=False, bins=range(a, b+1))

sns.distplot(list_of_distances, kde=False, bins=custom_range, color="red")
plt.xlabel("Distance")
plt.ylabel("# Occurrences")
```

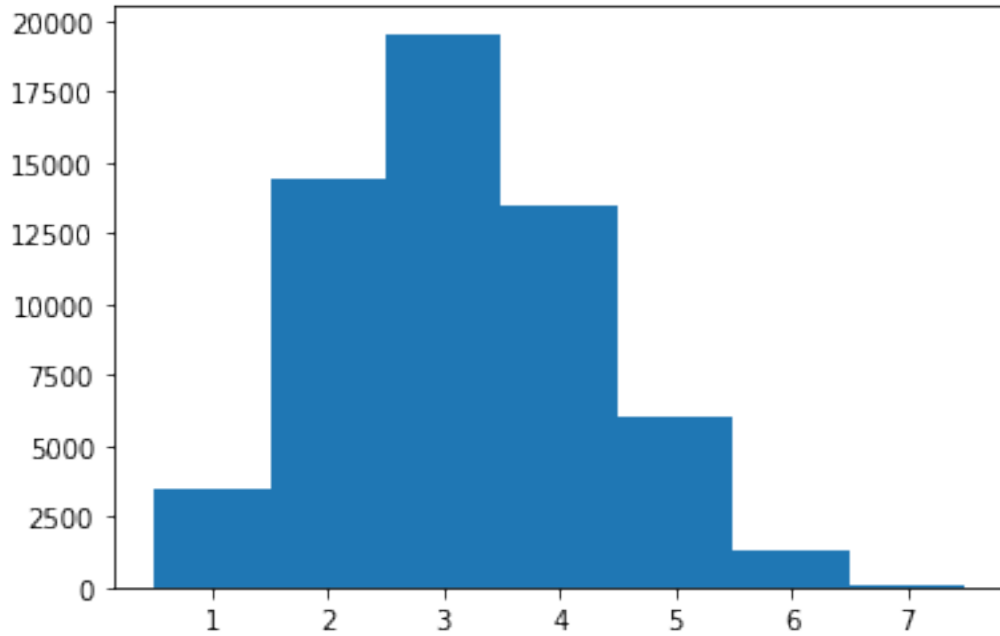
```
[0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5]
```

```
[69]: Text(0, 0.5, '# Occurrences')
```



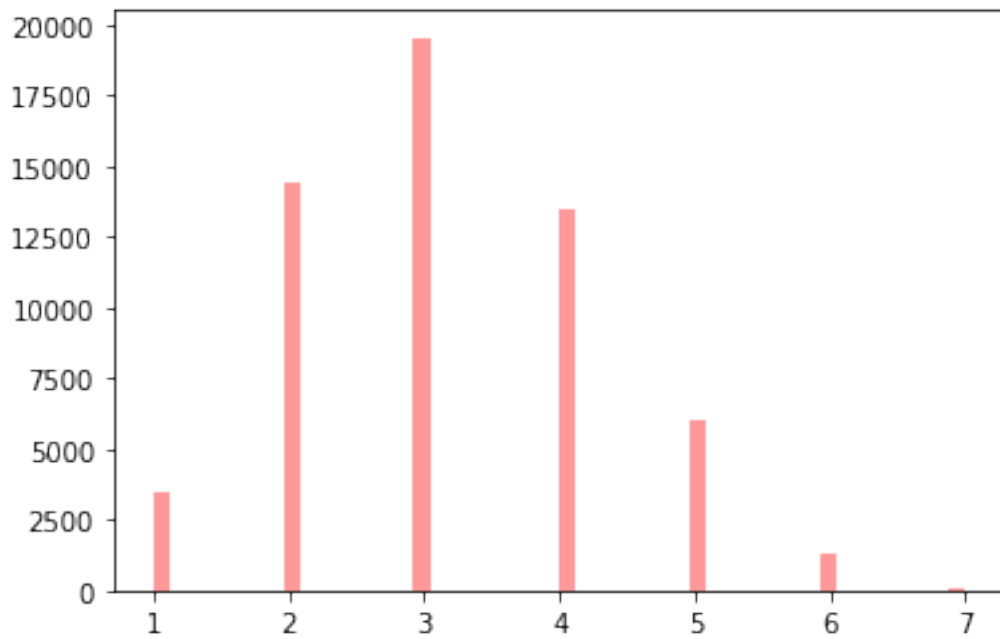
```
[63]: plt.hist(list_of_distances, bins=custom_range)
```

```
[63]: (array([ 3516., 14404., 19524., 13510.,  6042.,  1332.,    58.]),
array([0.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5]),
<BarContainer object of 7 artists>)
```



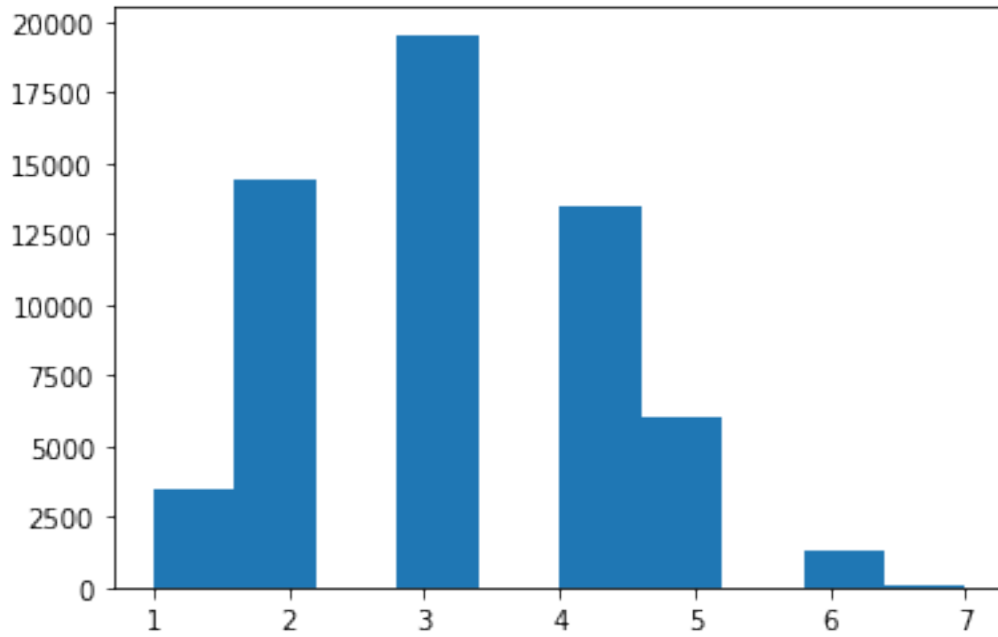
```
[68]: sns.distplot(list_of_distances, kde=False, color="red")
```

```
[68]: <AxesSubplot:>
```



```
[66]: plt.hist(list_of_distances)
```

```
[66]: (array([ 3516., 14404.,    0., 19524.,    0., 13510.,  6042.,    0.,  
          1332.,   58.]),  
array([1. , 1.6, 2.2, 2.8, 3.4, 4. , 4.6, 5.2, 5.8, 6.4, 7. ]),  
<BarContainer object of 10 artists>)
```



```
[ ]:
```