Kocherlakota Lakshmipathi Rao

```
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
```

```
messages = pd.read_csv("messages.csv")
employees = pd.read_csv("employees.csv")
```

```
messages.head(10)
```

	sender	receiver	timestamp	message_length
0	79	48	2021-06-02 05:41:34	88
1	79	63	2021-06-02 05:42:15	72
2	79	58	2021-06-02 05:44:24	86
3	79	70	2021-06-02 05:49:07	26
4	79	109	2021-06-02 19:51:47	73
5	79	58	2021-06-03 01:12:11	37
6	144	99	2021-06-03 09:54:41	33
7	144	105	2021-06-03 09:57:02	80
8	144	121	2021-06-03 09:59:16	13
9	177	32	2021-06-04 07:50:33	32

Double-click (or enter) to edit

```
employees.head(10)
```

	id	department	location	age
0	3	Operations	US	33
1	6	Sales	UK	50
2	8	IT	Brasil	54
3	9	Admin	UK	32
4	12	Operations	Brasil	51
5	19	Marketing	US	50
6	23	Sales	Brasil	39
7	26	Operations	France	32
8	27	Sales	France	58
9	29	Admin	France	33

Next steps: Generate code with employees View recommended plots

```
send_act = messages.groupby('sender').size()
rec_act = messages.groupby('receiver').size()
dep_act = (send_act.add(rec_act, fill_value=0)).astype(int)
most_active_dep = dep_act.idxmax()
least_active_dep = dep_act.idxmin()
```

```
print(send_act)
```

```
print(rec_act)
```

```
print(dep_act)
print(most_active_dep)
print(least_active_dep)
```

```
sender
79       13
128     266
144     221
162      11
173      10
       ...
1800      4
1802      2
1807     16
1879      4
```

```
1881     28
Length: 85, dtype: int64
receiver
3       11
6       10
8        1
9       22
12      12
        ..
1796     2
1801     4
1830     2
1839     8
1890     2
Length: 617, dtype: int64
3       11
6       10
8        1
9       22
12      12
        ..
1830     2
1839     8
1879     4
1881    28
1890     2
Length: 664, dtype: int64
605
8
```

```
emp_con = pd.concat([messages['sender'], messages['receiver']]).value_counts()
emp_high_con = emp_con.idxmax()
```

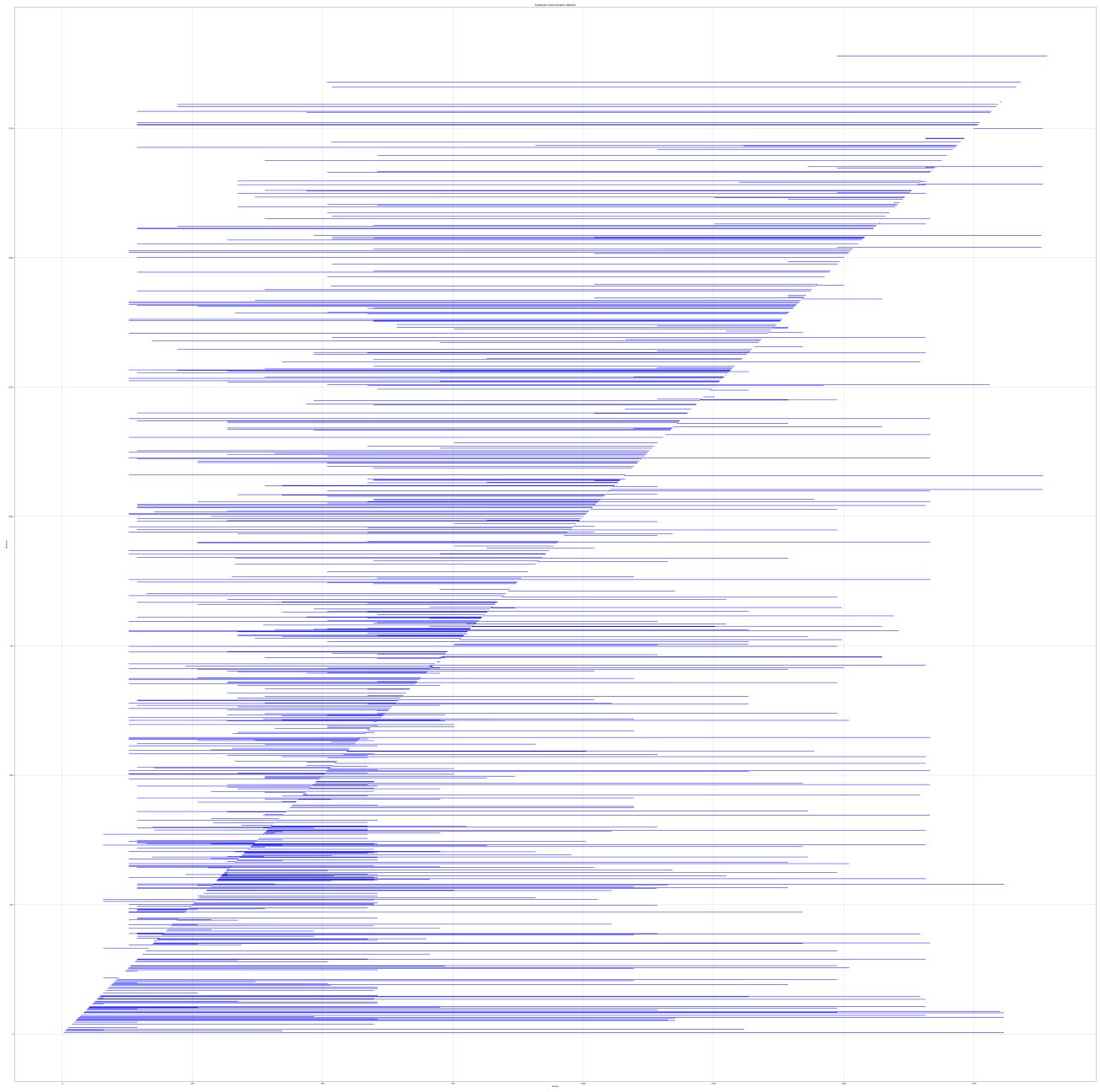
```
print(emp_high_con)
```

```
605
```

```
plt.figure(figsize=(100, 100))

for index, row in messages.iterrows():
    plt.plot([row['sender'], row['receiver']], [row['receiver'], row['receiver']], 'b-',

plt.xlabel('Sender')
plt.ylabel('Receiver')
plt.title('Employee Communication Network')
plt.grid(True)
plt.show()
```



```
L = nx.from_pandas_edgelist(messages, 'sender', 'receiver')

pos = nx.spring_layout(L)

plt.figure(figsize=(60, 60))
nx.draw(G, pos=pos, with_labels=True, node_size=1000, node_color='skyblue', font_size=10
plt.title('Employee Communication Network (Spring Layout)')
plt.show()
```

```
dep_influence = nx.degree_centrality(L)
emp_influence = nx.degree_centrality(L)
```

```
print(dep_influence)
print(emp_influence)
```

```
{79: 0.016591251885369532, 48: 0.0030165912518853697, 63: 0.006033182503770739, 58:
```

{79: 0.016591251885369532, 48: 0.0030165912518853697, 63: 0.006033182503770739, 58:

```
plt.figure(figsize=(70, 70))
nx.draw(L, with_labels=True, node_size=1000, node_color='orange', font_size=10, font_wei
plt.title('Employee Communication Network')
plt.show()
```

Employee Communication Network

```
print("Most Active Department:", most_active_dep)
print("Least Active Department:", least_active_dep)
print("Employee with Highest Connections:", emp_high_con)
print("Most Influential Departments:")
for department, influence in sorted(dep_influence.items(), key=lambda x: x[1], reverse=T
    print("- Department:", department, "| Influence:", influence)
print("Most Influential Employees:")
for employee, influence in sorted(emp_influence.items(), key=lambda x: x[1], reverse=Tru
```