Employees communications adn network analysis

March 20, 2023

1 Data is like a piece of art needs a caring aye to meditate it so it can reveals it's secrets ,so let's take the first look at this painting

[1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import plotly
import plotly.graph_objects as go
import plotly.express as px
```

[2]:
```python
messages = pd.read_csv('data/messages.csv', parse_dates= ['timestamp'])
messages
```

[2]:
```
      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
…          …         …                   …               …
3507     469      1629 2021-11-24 05:04:57              75
3508    1487      1543 2021-11-26 00:39:43              25
3509     144      1713 2021-11-28 18:30:47              51
3510    1879      1520 2021-11-29 07:27:52              58
3511    1879      1543 2021-11-29 07:37:49              56

[3512 rows x 4 columns]
```

[3]:
```python
employees = pd.read_csv('data/employees.csv')
employees
```

[3]:
```
       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
..    …          …       …   …
659  1830        Admin       UK   42
660  1839        Admin   France   28
661  1879  Engineering       US   40
662  1881        Sales  Germany   57
663  1890        Admin       US   39

[664 rows x 4 columns]
```

1.0.1 First to get answers from our data we have to ask the right questions ,so lets starting with Messages data ,

1.0.2 We have (664) employees in our beautiful company ,

1.0.3 generally what is the percentage of employees are senders ,

1.0.4 Who is sending and did not receive response or interaction ,

1.0.5 Who is sender and receiver ,

1.0.6 Who is just receiver and don't make interaction with others (muted people)

1.0.7 Is there are any employee not sender and not receiver , i think this would be the worst case for any employee

[4]:
```
messages.shape
```

[4]: (3512, 4)

[5]:
```
messages.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3512 entries, 0 to 3511
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   sender          3512 non-null   int64
 1   receiver        3512 non-null   int64
 2   timestamp       3512 non-null   datetime64[ns]
 3   message_length  3512 non-null   int64
dtypes: datetime64[ns](1), int64(3)
memory usage: 109.9 KB
```

[6]:
```
# Ccounting total messages words

messages ['message_length'].sum()
```

[6]: 170159

[17]:
```
messages [messages.duplicated()]
```

[17]:

	sender	receiver	timestamp	message_length
3446	1807	32	2021-10-13 22:25:17	50
3478	1657	1675	2021-11-02 07:42:25	52
3490	1881	1676	2021-11-17 06:45:28	27

[26]:
```
messages[messages.timestamp=='2021-10-13 22:25:17']
```

[26]:

	sender	receiver	timestamp	message_length
3333	1807	32	2021-10-13 22:25:17	50
3446	1807	32	2021-10-13 22:25:17	50

[7]:
```
messages.describe()
```

[7]:

	sender	receiver	message_length
count	3512.000000	3512.000000	3512.000000
mean	591.953303	627.052677	48.450740
std	397.953749	460.981865	22.857461
min	79.000000	3.000000	10.000000
25%	332.000000	277.000000	29.000000
50%	509.000000	509.000000	49.000000
75%	605.000000	878.000000	68.000000
max	1881.000000	1890.000000	88.000000

[8]:
```
#counting our senders
senders_unique_ids=messages.sender.value_counts().rename_axis('unique_senders').
↪reset_index(name='senders_messages_counts')
senders_unique_ids
```

[8]:

	unique_senders	senders_messages_counts
0	605	459
1	128	266
2	144	221
3	509	216
4	389	196
..	…	…
80	977	1
81	1461	1
82	521	1
83	1605	1
84	280	1

[85 rows x 2 columns]

1.1 so let's go deeper and make our employees slices , i think we can slice into 5 groups,

1.2 all senders generally , most senders , only senders , senders and receivers and finally only receivers

[9]:
```python
all_senders=senders_unique_ids.unique_senders.tolist()
```

2 So we have only 85 employee trying to have communication with others ,

3 lees than 13 % from our people sending messages and it's a bad indicator

4

5 Here we just put our hands on the biggest , main and general problem witch is the most of our employees don't interact with others ,

6 so we will focus in it and we will ignore making date and time analysis because we have a general problem not a periodic problem

[11]:
```python
#check the most sender id messagees lengh
messages[(messages['sender']==605)]['message_length'].sum()
```

[11]: 21989

6.1 great now we have the most active id who is the best sender, and the lucky id is no (605) with the best score (21989) word

7 Now let's see employees whose have the most impact , according to to the role 20/80 I think that just 20% of senders employees making 80% of total impact

8 so let's explore

[12]:
```python
senders_unique_ids.senders_messages_counts.describe()
```

[12]:
```
count     85.000000
mean      41.317647
std       74.844476
```

```
min          1.000000
25%          4.000000
50%         11.000000
75%         41.000000
max        459.000000
Name: senders_messages_counts, dtype: float64
```

[13]:
```
#counting employees whose sendenig more than the average of all messages
mean=41
most_senders=senders_unique_ids[senders_unique_ids.senders_messages_counts>mean]
most_senders=most_senders.unique_senders.tolist()
len(most_senders)
```

[13]:
```
21
```

[14]:
```
#counting the most senders messages length by word
messages[messages['sender'].isin(most_senders)]['message_length'].sum()
```

[14]:
```
137198
```

8.0.1 great we have (21) of (85) employee -(23.5%) making (76.5%) of messages and (137198) word witch is (81%) of total words length ,As expected, and those are the employees whose making the most great impact , witch is the third targeted questions in the competition

[15]:
```
#counting lowest senders whose sent only one message
lowest_senders_ids=senders_unique_ids[senders_unique_ids.
 ↪senders_messages_counts==1]['unique_senders'].tolist()
```

[16]:
```
lowest_senders=messages[messages['sender'].isin(lowest_senders_ids)]['sender'].
 ↪tolist()
lowest_senders
```

[16]:
```
[186, 247, 521, 280, 977, 1140, 1461, 1569, 1605, 1670, 1780]
```

employee no (1605) is the lowest he is lazy in writing but still better than the only receivers

[17]:
```
# counting  Receivers
receiver_unique_ids=messages.receiver.value_counts().
 ↪rename_axis('unique_receivers').reset_index(name='receivers_messages_count')
receiver_unique_ids
```

[17]:

	unique_receivers	receivers_messages_count
0	281	60
1	704	54
2	308	51
3	32	47

```
4                236                   47
..               …                    …
612             1122                    1
613             1317                    1
614               94                    1
615              963                    1
616              872                    1

[617 rows x 2 columns]
```

8.0.2 good it means that (93%) of our employees are receiving messages

no (281) is the most receiver by (60) message

9 Now let's go deeper in our data

[18]:
```
#concat sent and received messages per employee id
sender_receiver_id=  pd.concat([receiver_unique_ids,senders_unique_ids],axis=1)
```

[19]:
```
#count common senders and rcivers


senders_receivers=sender_receiver_id[sender_receiver_id['unique_receivers'].
 ↪isin(sender_receiver_id['unique_senders'].tolist())]['unique_receivers'].
 ↪tolist()
```

[20]:
```
# counting senders-receivers employees
len(senders_receivers)
```

[20]:
```
38
```

9.0.1 Now we know that just 38 employee are communicating to gather , only (6%) of our people witch is a problem

9.0.2 only (45%) from people whose sending messages are receiving response

9.0.3 let's check if there are employees sending messages to others and received no response , and the same for receivers whose just receiving messages and don't make response

[21]:
```
only_senders=senders_unique_ids[~senders_unique_ids.unique_senders.
 ↪isin(senders_receivers)]['unique_senders'].tolist()
```

[22]:
```
only_receivers=receiver_unique_ids[~receiver_unique_ids.unique_receivers.
 ↪isin(senders_receivers)]['unique_receivers'].tolist()
```

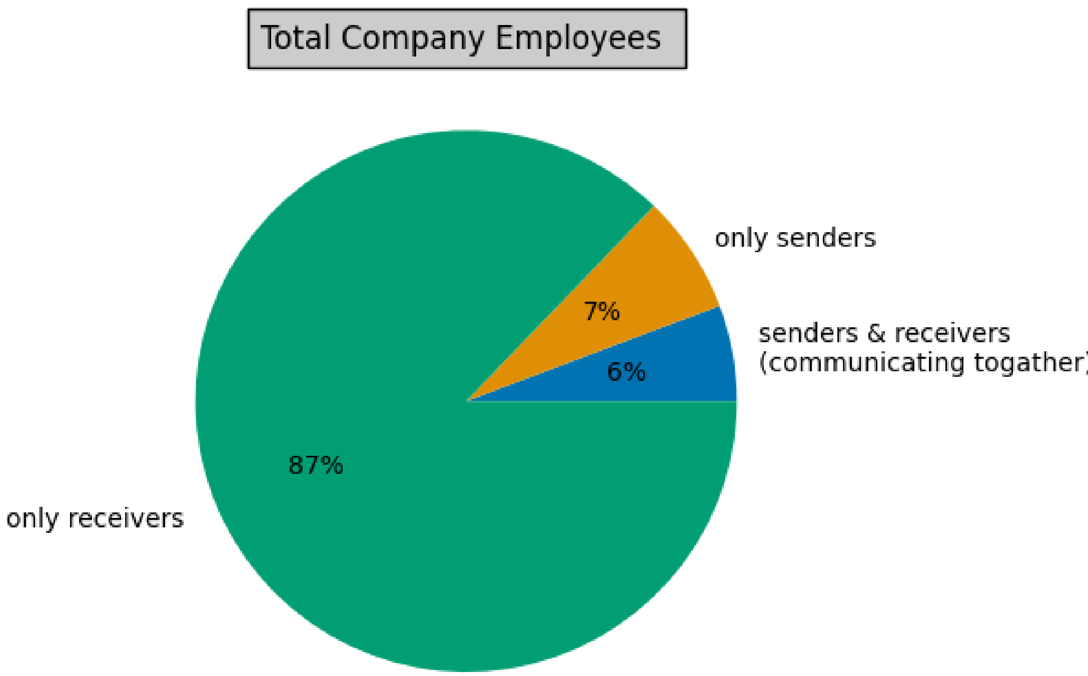
[23]:
```
# counting only senders employees
len(only_senders)
```

[23]: 47

[24]:
```
# counting only receivers employees
len(only_receivers)
```

[24]: 579

[25]:
```
y = np.array([38,47,579])
mylabels = ["senders & receivers\n(communicating togather)",
            'only senders','only receivers ']
colors = sns.color_palette('colorblind')

plt.pie(y, labels = mylabels , colors=colors , autopct='%.0f%%')
plt.savefig('')
plt.title("Total Company Employees " , bbox={'facecolor':'0.8', 'pad':5})

plt.show()
```



9.0.4 Now our data told us that (55%) of people whose sending messages didn't receive response witch is a big ratio

9.0.5 And only (6%) of people whose receiving messages responds to these messages and (94%) didn't make a response

[26]:
```
# check if we have an employee who don't sent or receive messages
employees.id.count() == len(senders_receivers) + len(only_senders) +␣
↪len(only_receivers)
```

[26]:
```
True
```

9.0.6 So fortunately we don't have any dead employee who didn't send or receve any message

10 now lets explore our employess distribution

[27]:
```
#counting departments and employees distribution
department_employees_count=employees.department.value_counts().
↪rename_axis('department').reset_index(name = 'employees_count')
department_employees_count
```

[27]:

	department	employees_count
0	Sales	161
1	Admin	140
2	Operations	134
3	Engineering	100
4	IT	77
5	Marketing	52

10.1 As we see sales department is the biggest one

[28]:
```
#counting locations  and employees distribution

employees.location.value_counts()
```

[28]:
```
US         277
France     157
Germany     99
UK          70
Brasil      61
Name: location, dtype: int64
```

10.2 So US is the most location contains employees

[29]:
```
#counting departments by how many senders inside (generally )

all_senders_per_department=employees[employees.id.
 ↪isin(all_senders)]['department'].value_counts().rename_axis('department').
 ↪reset_index(name='all_senders_count')
all_senders_per_department
```

[29]:

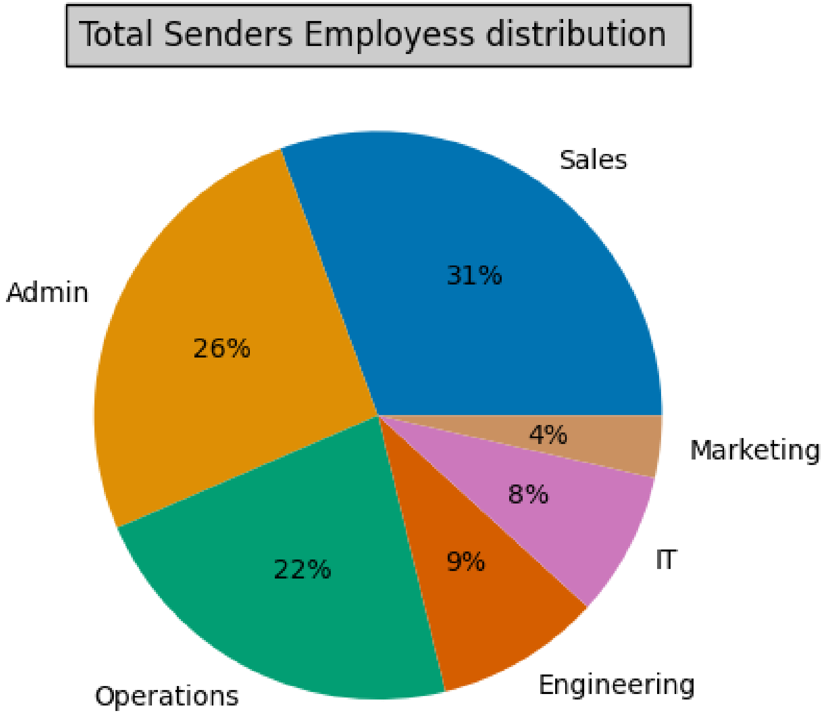
	department	all_senders_count
0	Sales	26
1	Admin	22
2	Operations	19
3	Engineering	8
4	IT	7
5	Marketing	3

[30]:
```
#counting departments by how many of the most senders inside

employees[employees.id.isin(most_senders)]['department'].value_counts()
```

[30]:
```
Sales         10
Operations     9
Admin          2
Name: department, dtype: int64
```

10.3 So the most influential departments are (Sales-Operations)

[31]:
```
y = np.array([26 , 22 , 19 , 8 , 7 , 3 ] )
mylabels = ["Sales",'Admin','Operations','Engineering',
            'IT','Marketing']
colors = sns.color_palette('colorblind')

plt.pie(y, labels = mylabels , colors=colors , autopct='%.0f%%')
plt.savefig('')
plt.title("Total Senders Employess distribution " , bbox={'facecolor':'0.8',␣
 ↪'pad':5})

plt.show()
```



10.3.1 Only by eye we see clearly that Salse is the most active and the most influential department , because it have more than (30%) of all senders in generally And (47%) of the most senders employees specifically

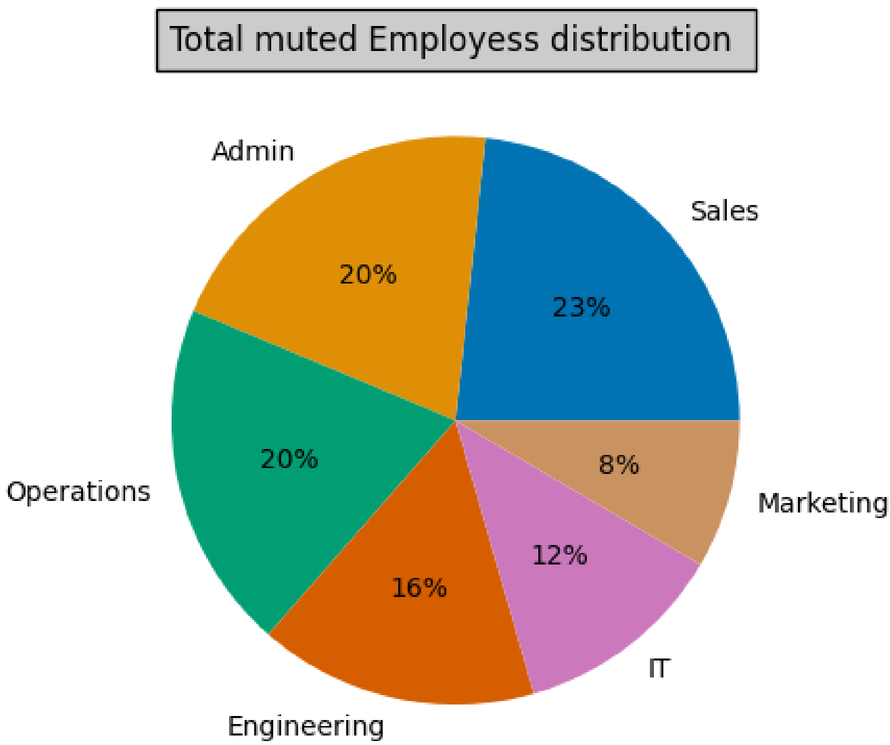
[32]:
```python
#counting departments by how many of only_receivers inside

only_receivers_per_department=employees[employees.id.
    ↪isin(only_receivers)]['department'].value_counts().rename_axis('department').
    ↪reset_index(name='only_receivers_count')
only_receivers_per_department
```

[32]:

	department	only_receivers_count
0	Sales	135
1	Admin	118
2	Operations	115
3	Engineering	92
4	IT	70
5	Marketing	49

10.4 So now clearly we can see that only_receivers employees are our target, let's describe them correctly and call them muted employees

[33]:
```
y = np.array([135 , 118 , 115 , 92 , 70 , 49 ] )
mylabels = ["Sales",'Admin','Operations','Engineering',
            'IT','Marketing']
colors = sns.color_palette('colorblind')

plt.pie(y, labels = mylabels , colors=colors , autopct='%1.0f%%')
plt.savefig('')
plt.title("Total muted Employess distribution " , bbox={'facecolor':'0.8',␣
 ↪'pad':5})

plt.show()
```



10.5 now let's go deeper and make network analysis across all departments

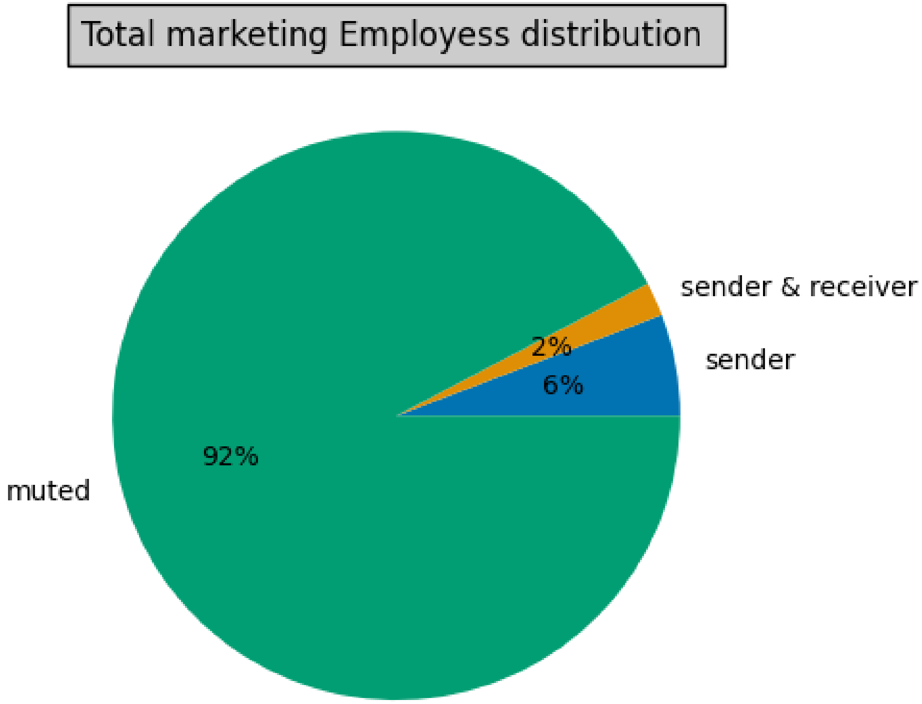
[34]:
```
#counting departments by how many of senders receivers  inside
all_senders_receivers_per_department=employees[employees.id.
 ↪isin(senders_receivers)]['department'].value_counts().
 ↪rename_axis('department').reset_index(name='senders_receivers_count')
all_senders_receivers_per_department
```

[34]:

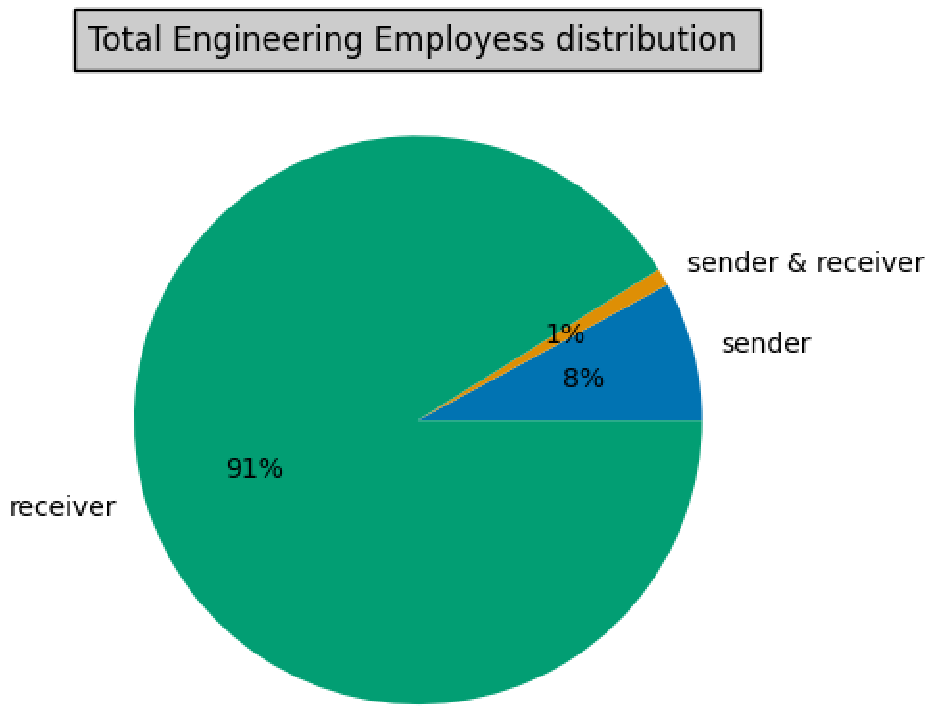
	department	senders_receivers_count
0	Sales	11
1	Operations	11
2	Admin	10
3	IT	3
4	Marketing	2
5	Engineering	1

[35]:
```
TOTAL=all_senders_per_department.
↪merge(all_senders_receivers_per_department,on='department').
↪merge(only_receivers_per_department,on='department').reset_index().
↪drop(columns='index',inplace=True)
TOTAL
```

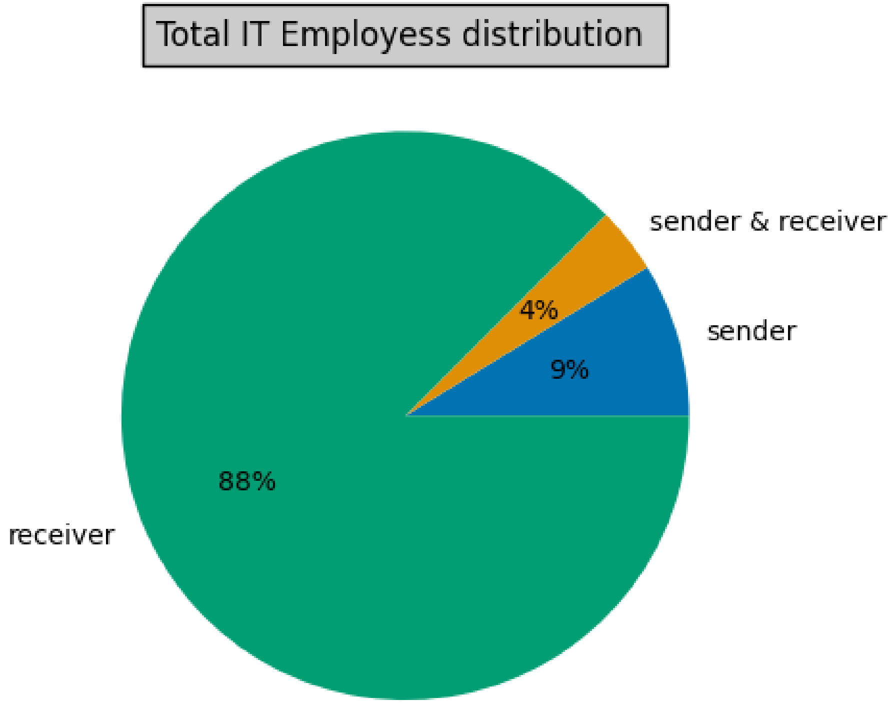
[36]:
```
y = np.array([3 , 1 , 48])
mylabels = ["sender",'sender & receiver','muted'
            ]
colors = sns.color_palette('colorblind')

plt.pie(y, labels = mylabels , colors=colors , autopct='%.0f%%')
plt.savefig('')
plt.title("Total marketing Employess distribution " , bbox={'facecolor':'0.8',␣
↪'pad':5})

plt.show()
```



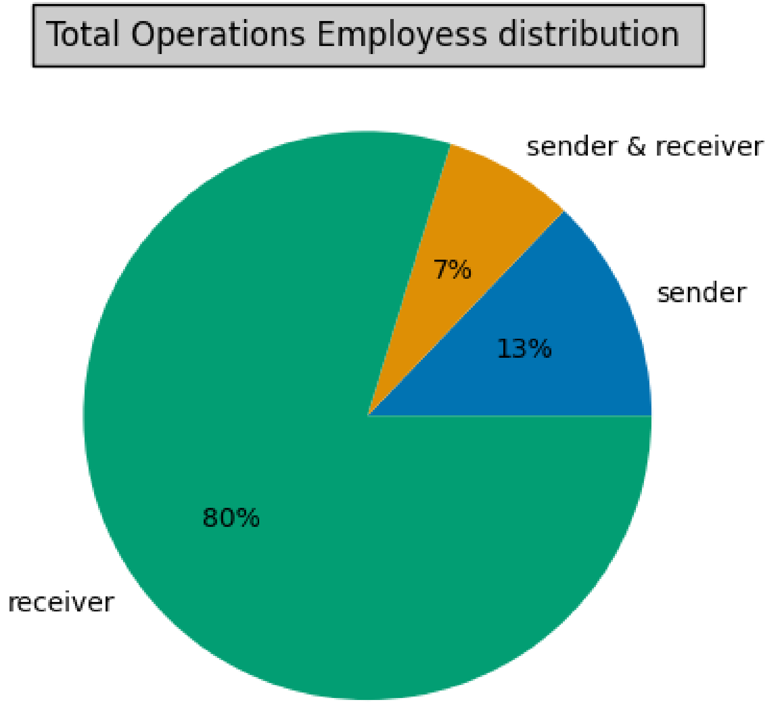
[37]:
```
y = np.array([8 , 1 , 92])
mylabels = ["sender",'sender & receiver','receiver'
            ]
colors = sns.color_palette('colorblind')

plt.pie(y, labels = mylabels , colors=colors , autopct='%.0f%%')
plt.savefig('')
plt.title("Total Engineering Employess distribution " , bbox={'facecolor':'0.
 ↪8', 'pad':5})

plt.show()
```



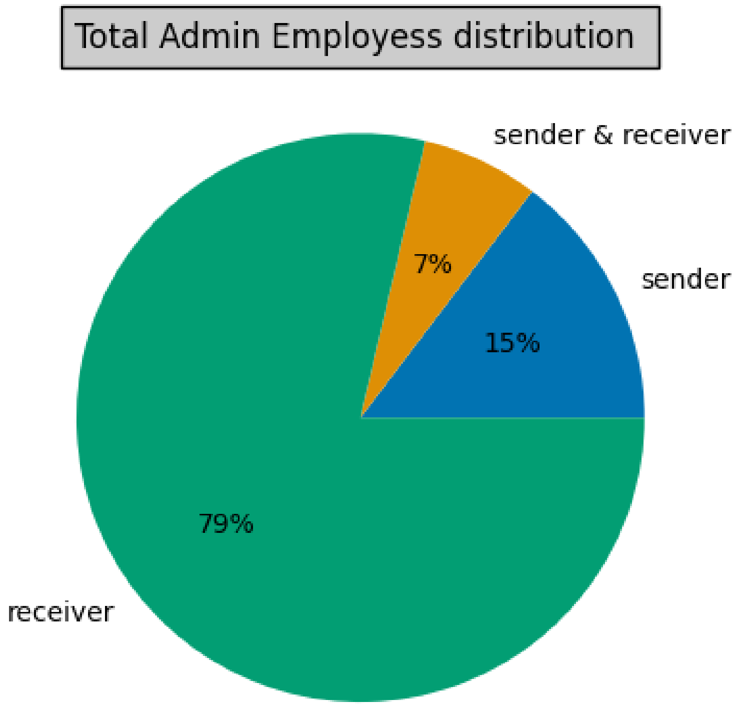
[38]:
```
y = np.array([7 , 3 , 70])
mylabels = ["sender",'sender & receiver','receiver'
            ]
colors = sns.color_palette('colorblind')

plt.pie(y, labels = mylabels , colors=colors , autopct='%.0f%%')
plt.savefig('')
plt.title("Total IT Employess distribution " , bbox={'facecolor':'0.8', 'pad':
 ↪5})

plt.show()
```



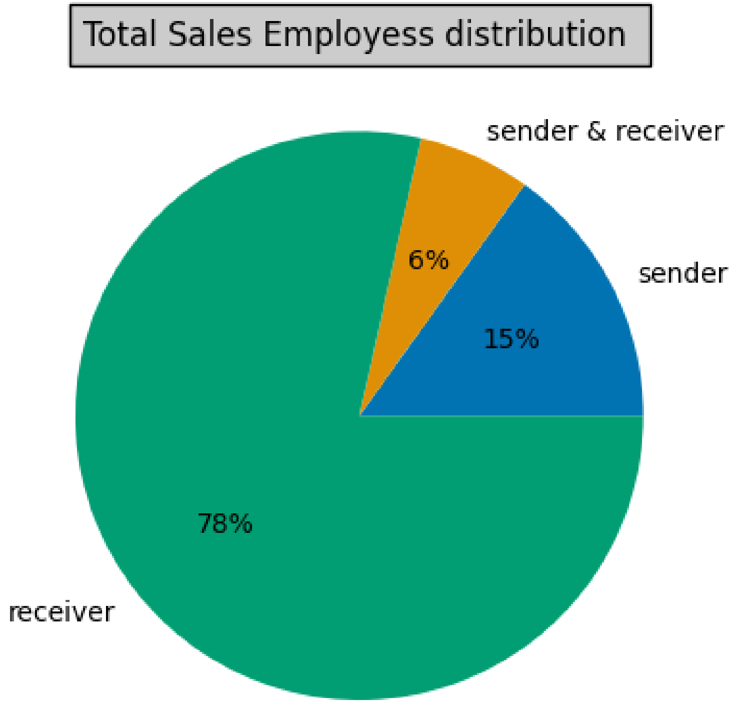
[39]:
```
y = np.array([19 , 11 , 118])
mylabels = ["sender",'sender & receiver','receiver'
            ]
colors = sns.color_palette('colorblind')

plt.pie(y, labels = mylabels , colors=colors , autopct='%.0f%%')
plt.savefig('')
plt.title("Total Operations Employess distribution " , bbox={'facecolor':'0.8',␣
 ↪'pad':5})

plt.show()
```



[40]:
```
y = np.array([22 , 10 , 118])
mylabels = ["sender",'sender & receiver','receiver'
            ]
colors = sns.color_palette('colorblind')

plt.pie(y, labels = mylabels , colors=colors , autopct='%.0f%%')
plt.savefig('')
plt.title("Total Admin Employess distribution " , bbox={'facecolor':'0.8',␣
  ↪'pad':5})

plt.show()
```



[41]:
```
y = np.array([26 , 11 , 135])
mylabels = ["sender",'sender & receiver','receiver'
            ]
colors = sns.color_palette('colorblind')

plt.pie(y, labels = mylabels , colors=colors , autopct='%.0f%%')
plt.savefig('')
plt.title("Total Sales Employess distribution " , bbox={'facecolor':'0.8',␣
 ↪'pad':5})

plt.show()
```



10.5.1 So Marketing department is the least active with only less than (5%) of the department employees sending messages

11

11.1 let's see how the geographical distribution impact on our target

11.1.1 lets grouping muted employees by department and location

[42]:
```
muted_employees=employees[employees.id.isin(only_receivers)]
```

[43]:
```
muted_employees=muted_employees.groupby(['location','department'])['id'].
 ↪count().reset_index()
```

[44]:
```
muted_employees.rename(columns={'id':'muted_employees_count'},inplace=True)
```

[45]:
```
muted_employees
```

[45]:

	location	department	muted_employees_count
0	Brasil	Admin	6
1	Brasil	Engineering	11
2	Brasil	IT	5

```
3    Brasil    Marketing                       3
4    Brasil   Operations                      14
5    Brasil        Sales                      17
6    France        Admin                      30
7    France  Engineering                      21
8    France           IT                      13
9    France    Marketing                      11
10   France   Operations                      28
11   France        Sales                      33
12  Germany        Admin                      15
13  Germany  Engineering                       9
14  Germany           IT                      12
15  Germany    Marketing                       7
16  Germany   Operations                      18
17  Germany        Sales                      24
18       UK        Admin                      14
19       UK  Engineering                      14
20       UK           IT                      10
21       UK    Marketing                       9
22       UK   Operations                       8
23       UK        Sales                      11
24       US        Admin                      53
25       US  Engineering                      37
26       US           IT                      30
27       US    Marketing                      19
28       US   Operations                      47
29       US        Sales                      50
```

[46]:

```
fig = px.treemap(muted_employees,
                path = ['department','location'],
                color_continuous_scale = 'deep',
                values='muted_employees_count' , color = 'muted_employees_count'
            )
fig.update_layout(width=1000 , height=550, title={
        'text':'Muted employees distribution by depatment and country ',
        'y':0.99,
        'x':0.4,
        'xanchor': 'center',
        'yanchor': 'top'})
plt.savefig('dep_loc_dist')

fig.show()
```

```
<Figure size 640x480 with 0 Axes>
```

11.2 So it's seems to that our target concentrated in sales , operations and admin departments specially at US

12 Story „„„ conclusions „„„ Recommendations

12.1 From the first look at our painting we can clearly see that senders employees ratio is very weak witch is means that most of our people don't trying to communicate or interact with others , and this is our problem in generally ,so there are two scenarios in this case

13

13.1 first one is related to human nature because people by nature tends to communicating together based on this we assuming that there are another communication channels like whats app groups for example , so in this case our data channel isn't the only way to communicate between employees, so we recommend in this case to make a single and unique system for communication across the entire company so we can collect the new data and reanalyze it to be aware of the real situation and in this case feel free to contact me if you want to make a data driven decision

13.2 The second scenario is this data reflecting the real situation so lets recap and answer the questions

13.3 since we have a problem witch is 87% of our employees didn't trying to communicate with others , and only 6% of our peoples in a unique case sharing messages to each others , so the only way to get through this problem is to deal with this 579 whose silent or muted employees because this factor will directly make the senders ratio increases and therefore the (senders-receivers) ratio as a unique case will increase too

14 And here we are answering the questions

14.1 Sales , Admin and Operations are the most active departments

14.2 Marketing-IT-Engineering are the least active department by ascending

14.3 employee who has the most connections is id no (605)

14.4 the most influential departments are (Sales-Operations)

14.5 and the most influential employees are this 21 employees by ascending [605,128,144,509,389,598,317,586,483,725,337,422,260,469,332,734,815,518,1142,1487,1

14.6 We agree that we have a general problem in all departments but if we have to choose We would recommend the HR team focus to boost collaboration in Marketing , Engineering and IT departments

15 Recommendation

15.1 Generally we need to encourage our employees to communicate with etch others , so for example we can honoring the ideal employees whose are the most influential , and also we can honoring the ideal departments and do this periodically and also periodically evaluating our employees

15.2 Sending a message and receiving a feedback is the normal case witch we have a problem with since we have many of senders didn't receive response