In [2]:

```
import seaborn as sns
import pandas as pd
import numpy as np
import re
import names
import warnings
warnings.filterwarnings('ignore')
```

In [3]:

```
data = pd.read_csv("C:\\Users\\reonh\\Documents\\NUS\\1920 S2\\BT3103\\Midsem project\\emai
```

In [4]:

```
#Function to categorise dates into their year
def year_from_date(date):
    match = re.search('[2]\d{3}', date)
    year = match.group(0) if match else '2013'
    return year
```

In [5]:

```
#creating a "year" column for each email
data["year"] = list(map(lambda x: int(year_from_date(str(x))), data["date"]))
data = data[data["year"]>2016]
```

I will be generating placeholder names/strings for the following personal data:

1. Email senders (from field)
2. Email subjects

In [6]:

```
#replacing from field with a placeholder name
for n in data["from"].unique():
    #skip assigning a new name for myself
    try:
        if "reon" in n.lower():
            u = "Reon Ho"
        else:
            u = names.get_full_name()
    except:
        u = names.get_full_name()
    while u in data["from"]:
        u = names.get_full_name()
    data["from"] = data["from"].replace({n:u})
```

In [13]:

```
#replacing subject with dummy placeholder (i.e. the row number)
data["subject"] =  data.index
```

In [8]:

```
#Data after processing
data.head()
```

Out[8]:

	subject	from	date	labels	to	year
0	0	Thomas Whitt	04 Feb 2020 20:03:42 +0000 (GMT)	Inbox,Category Promotions,Unread	reonho@gmail.com	2020
1	1	Susan Giron	Mon, 3 Feb 2020 18:20:39 +0000 (UTC)	Inbox,Category Social,Unread	Reon Ho <reonho@gmail.com>	2020
2	2	Ronald Richardson	Sun, 02 Feb 2020 03:12:24 +0100	Spam,Category Promotions,Unread	reonho@gmail.com	2020
3	3	Howard Kamiya	Sat, 1 Feb 2020 03:26:39 +0000	Inbox,Category Promotions,Unread	=?utf-8?Q? Reon=20Ho=20Rui=20An?= <reonho@gmail...	2020
4	4	Vernon Jacobs	Sat, 01 Feb 2020 19:22:52 +0000	Inbox,Category Promotions	reonho@gmail.com	2020

Question 1

1. Plot a barchart of the number of emails sent and received in each year for the past 3 years

In [9]:

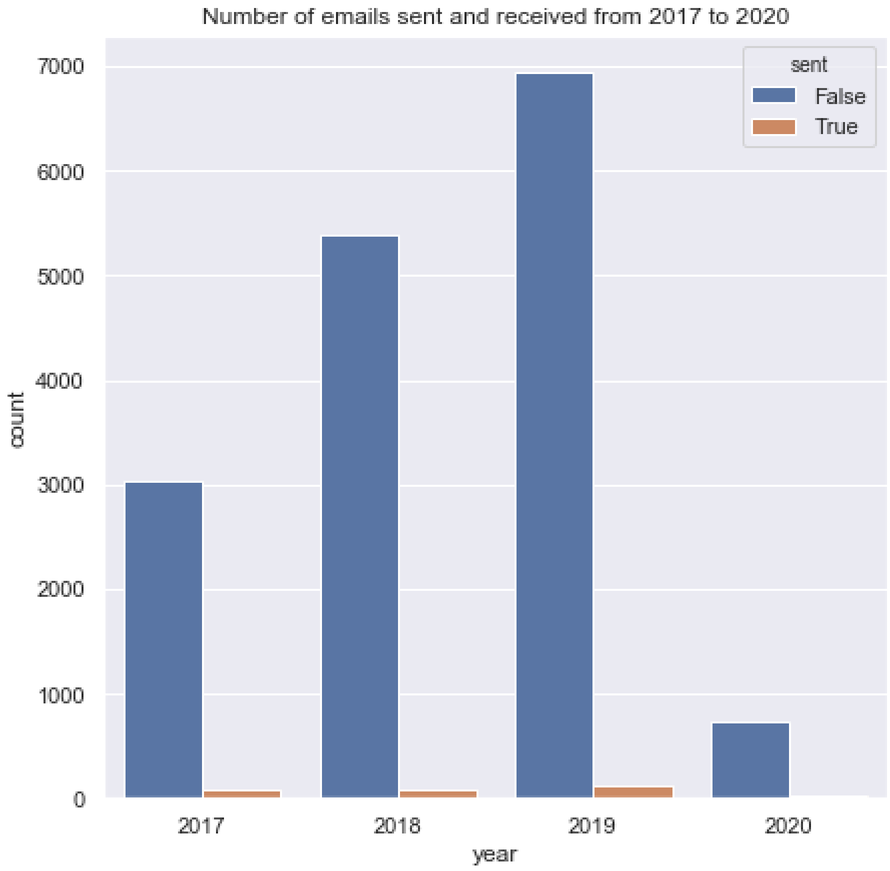
```
data["sent"] = data["from"]=="Reon Ho"
data.head()
```

Out[9]:

	subject	from	date	labels	to	year	sent
0	0	Thomas Whitt	04 Feb 2020 20:03:42 +0000 (GMT)	Inbox,Category Promotions,Unread	reonho@gmail.com	2020	False
1	1	Susan Giron	Mon, 3 Feb 2020 18:20:39 +0000 (UTC)	Inbox,Category Social,Unread	Reon Ho <reonho@gmail.com>	2020	False
2	2	Ronald Richardson	Sun, 02 Feb 2020 03:12:24 +0100	Spam,Category Promotions,Unread	reonho@gmail.com	2020	False
3	3	Howard Kamiya	Sat, 1 Feb 2020 03:26:39 +0000	Inbox,Category Promotions,Unread	=?utf-8?Q? Reon=20Ho=20Rui=20An?= <reonho@gmail...	2020	False
4	4	Vernon Jacobs	Sat, 01 Feb 2020 19:22:52 +0000	Inbox,Category Promotions	reonho@gmail.com	2020	False

In [10]:

```
sns.set(rc={'figure.figsize':(7,7)})

ax = sns.countplot(x="year", data=data, hue="sent")
ax = ax.set_title("Number of emails sent and received from 2017 to 2020")
```



2. Plot a breakdown of the number of emails received between June 2019 - August 2019 by From field (You can use fictitious names for the From sender to protect data privacy)

In [11]:

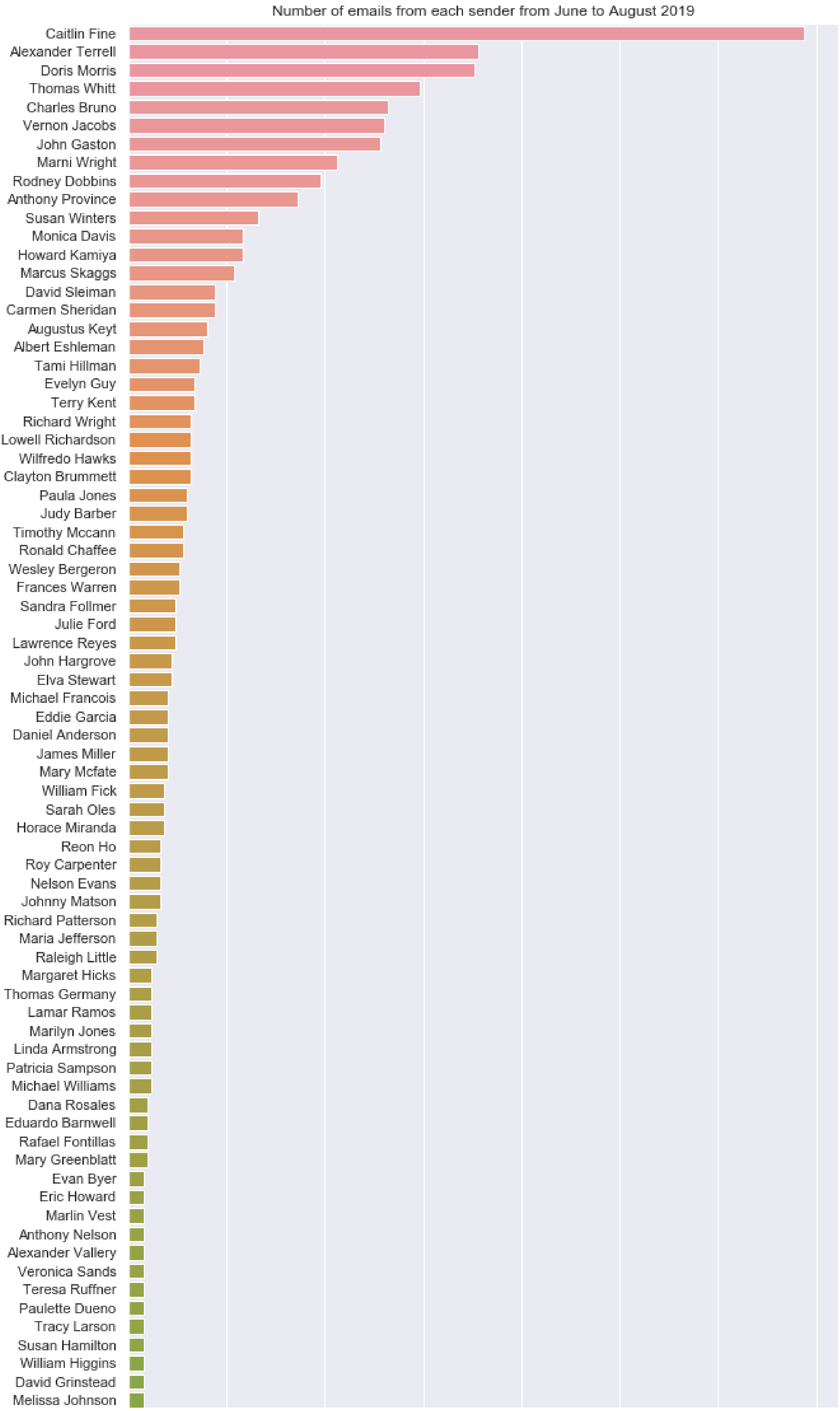
```
#creating a dataframe of only emails from Jun to Aug 2019
data_2019 = data[data["year"] == 2019]
data_2019_jun_to_aug = data_2019[data_2019["date"].str.contains('jun|jul|aug', case = False
data_2019_jun_to_aug.tail()
```

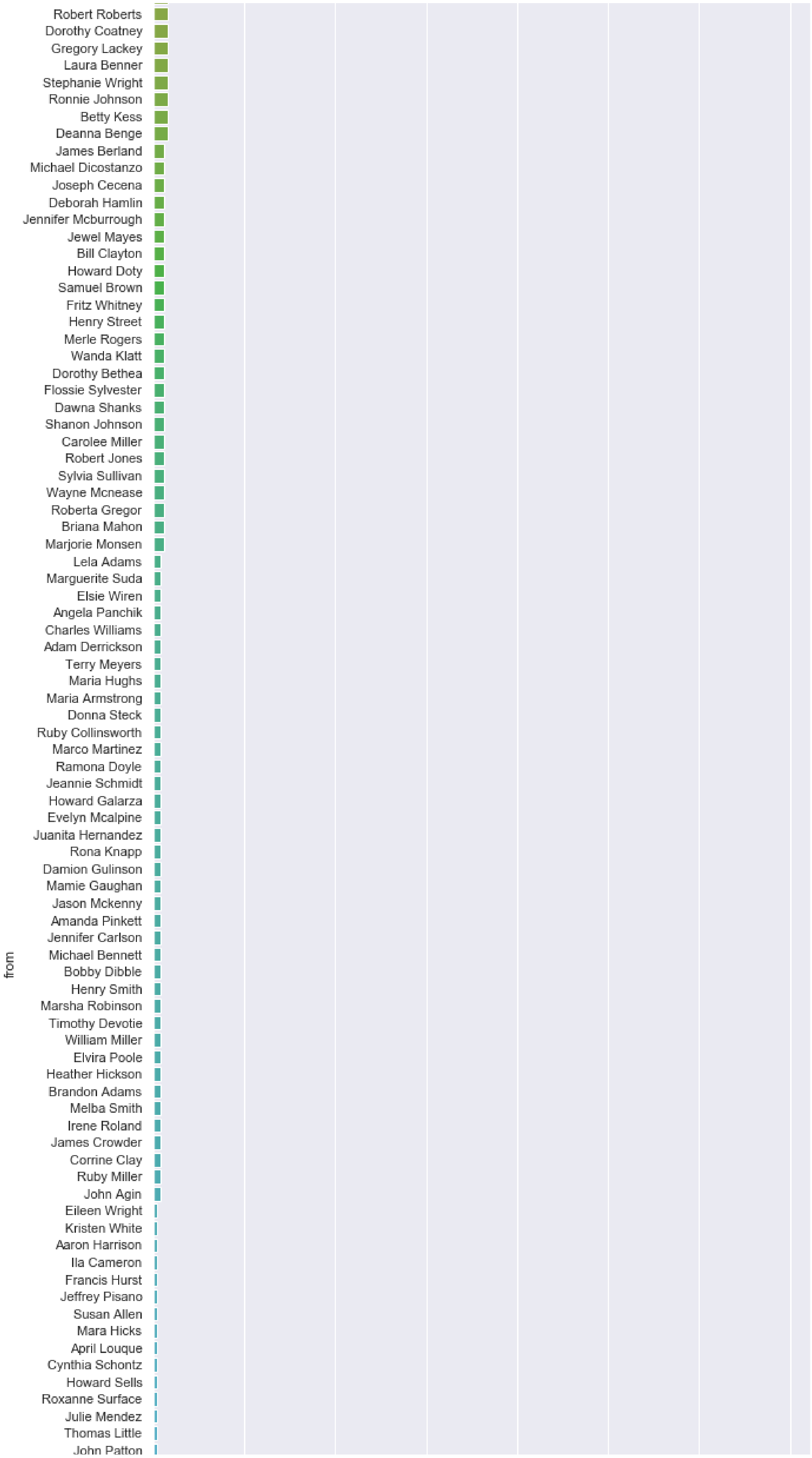
Out[11]:

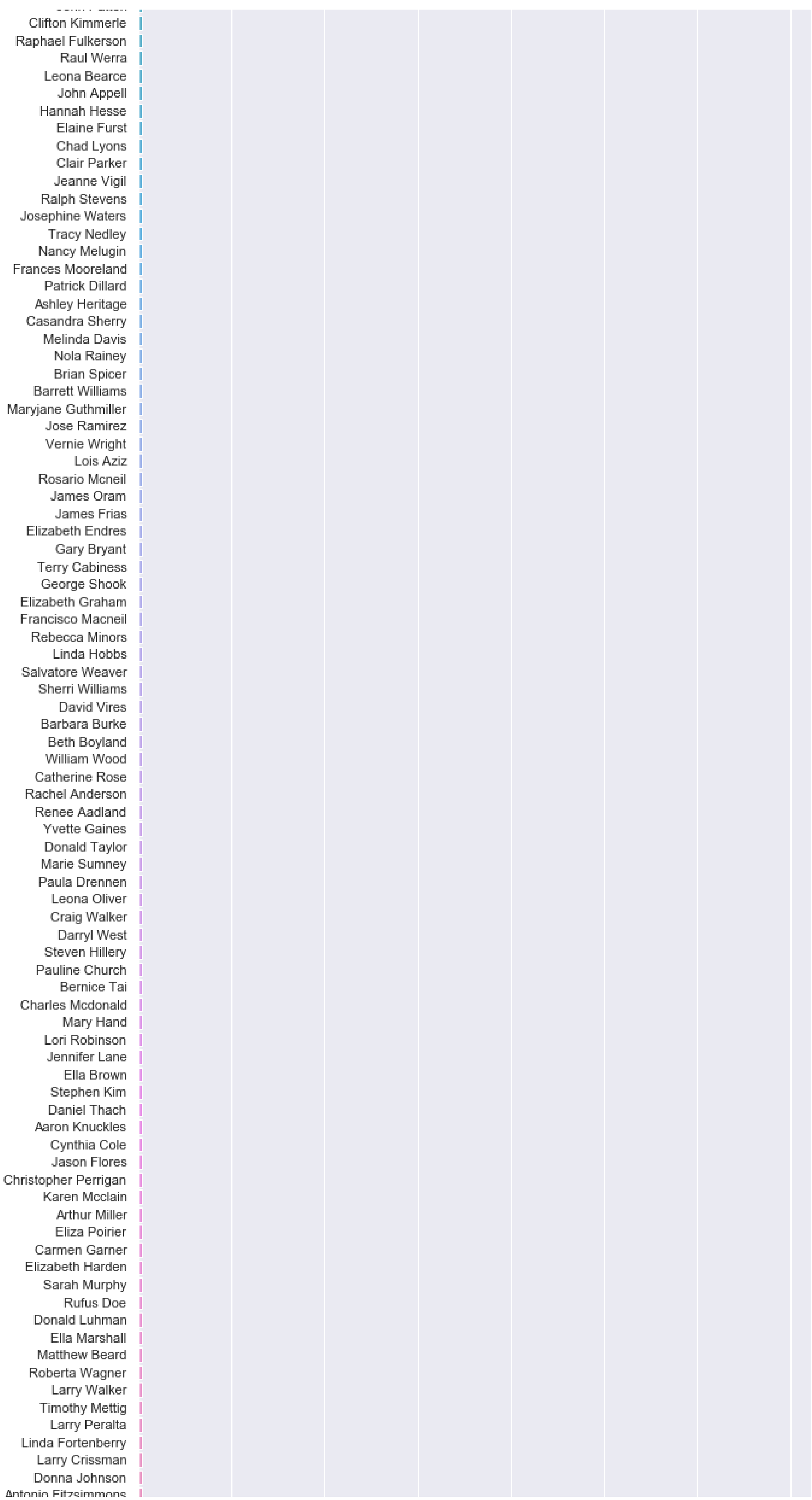
	subject	from	date	labels	to	year	sent
20590	20590	Linda Hobbs	Wed, 21 Aug 2019 11:30:40 +0200	Inbox,Category Updates,Unread	reonho@gmail.com	2019	False
20902	20902	Caitlin Fine	Sun, 9 Jun 2019 05:53:13 -0700	Inbox,Category Social,Unread	Reon Ho <reonho@gmail.com>	2019	False
22552	22552	Richard Wright	Thu, 25 Jul 2019 03:18:02 +0000	Inbox,Category Promotions,Unread	reonho@gmail.com	2019	False
23103	23103	Rodney Dobbins	Sat, 31 Aug 2019 07:01:22 -0700	Inbox,Opened,Category Promotions	reonho@gmail.com	2019	False
23233	23233	James Berland	Fri, 28 Jun 2019 18:39:27 -0700	Inbox,Category Promotions,Unread	reonho@gmail.com	2019	False

In [12]:

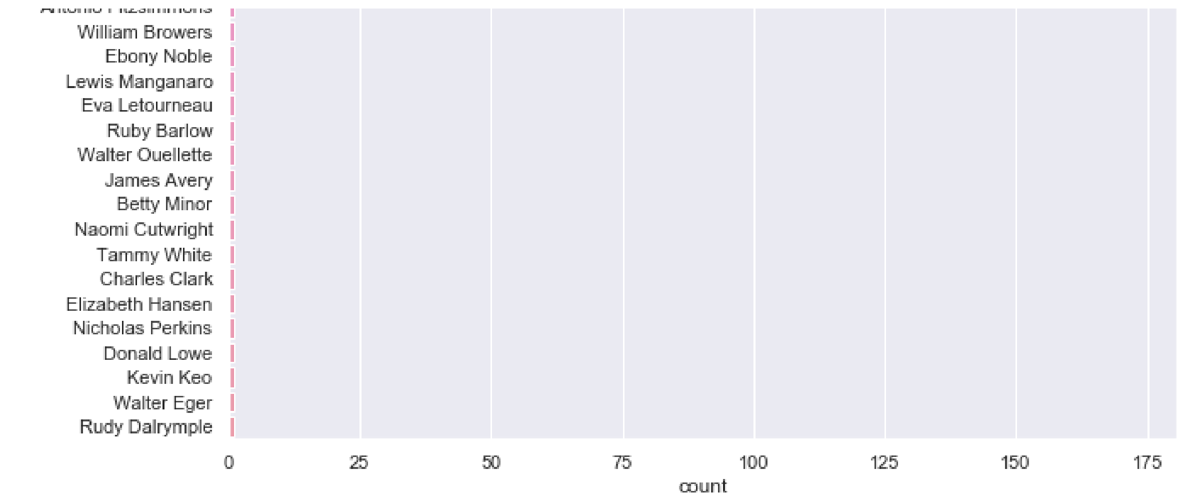
```
sns.set(rc={'figure.figsize':(10,70)})
ax = sns.countplot(y="from", data=data_2019_jun_to_aug, order=data_2019_jun_to_aug['from'].
ax = ax.set_title("Number of emails from each sender from June to August 2019")
```




Robert Roberts
Dorothy Coatney
Gregory Lackey
Laura Benner
Stephanie Wright
Ronnie Johnson
Betty Kess
Deanna Benge
James Berland
Michael Dicostanzo
Joseph Cecena
Deborah Hamlin
Jennifer Mcburrough
Jewel Mayes
Bill Clayton
Howard Doty
Samuel Brown
Fritz Whitney
Henry Street
Merle Rogers
Wanda Klatt
Dorothy Bethea
Flossie Sylvester
Dawna Shanks
Shanon Johnson
Carolee Miller
Robert Jones
Sylvia Sullivan
Wayne Mcnease
Roberta Gregor
Briana Mahon
Marjorie Monsen
Lela Adams
Marguerite Suda
Elsie Wiren
Angela Panchik
Charles Williams
Adam Derrickson
Terry Meyers
Maria Hughs
Maria Armstrong
Donna Steck
Ruby Collinsworth
Marco Martinez
Ramona Doyle
Jeannie Schmidt
Howard Galarza
Evelyn Mcalpine
Juanita Hernandez
Rona Knapp
Damion Gulinson
Mamie Gaughan
Jason Mckenny
Amanda Pinkett
Jennifer Carlson
Michael Bennett
Bobby Dibble
Henry Smith
Marsha Robinson
Timothy Devotie
William Miller
Elvira Poole
Heather Hickson
Brandon Adams
Melba Smith
Irene Roland
James Crowder
Corrine Clay
Ruby Miller
John Agin
Eileen Wright
Kristen White
Aaron Harrison
Ila Cameron
Francis Hurst
Jeffrey Pisano
Susan Allen
Mara Hicks
April Louque
Cynthia Schontz
Howard Sells
Roxanne Surface
Julie Mendez
Thomas Little
John Patton
from



Clifton Kimmerle
Raphael Fulkerson
Raul Werra
Leona Bearce
John Appell
Hannah Hesse
Elaine Furst
Chad Lyons
Clair Parker
Jeanne Vigil
Ralph Stevens
Josephine Waters
Tracy Nedley
Nancy Melugin
Frances Mooreland
Patrick Dillard
Ashley Heritage
Casandra Sherry
Melinda Davis
Nola Rainey
Brian Spicer
Barrett Williams
Maryjane Guthmiller
Jose Ramirez
Vernie Wright
Lois Aziz
Rosario Mcneil
James Oram
James Frias
Elizabeth Endres
Gary Bryant
Terry Cabiness
George Shook
Elizabeth Graham
Francisco Macneil
Rebecca Minors
Linda Hobbs
Salvatore Weaver
Sherri Williams
David Vires
Barbara Burke
Beth Boyland
William Wood
Catherine Rose
Rachel Anderson
Renee Aadland
Yvette Gaines
Donald Taylor
Marie Sumney
Paula Drennen
Leona Oliver
Craig Walker
Darryl West
Steven Hillery
Pauline Church
Bernice Tai
Charles Mcdonald
Mary Hand
Lori Robinson
Jennifer Lane
Ella Brown
Stephen Kim
Daniel Thach
Aaron Knuckles
Cynthia Cole
Jason Flores
Christopher Perrigan
Karen Mcclain
Arthur Miller
Eliza Poirier
Carmen Garner
Elizabeth Harden
Sarah Murphy
Rufus Doe
Donald Luhman
Ella Marshall
Matthew Beard
Roberta Wagner
Larry Walker
Timothy Mettig
Larry Peralta
Linda Fortenberry
Larry Crissman
Donna Johnson
Antonio Fitzsimmons



Question 2

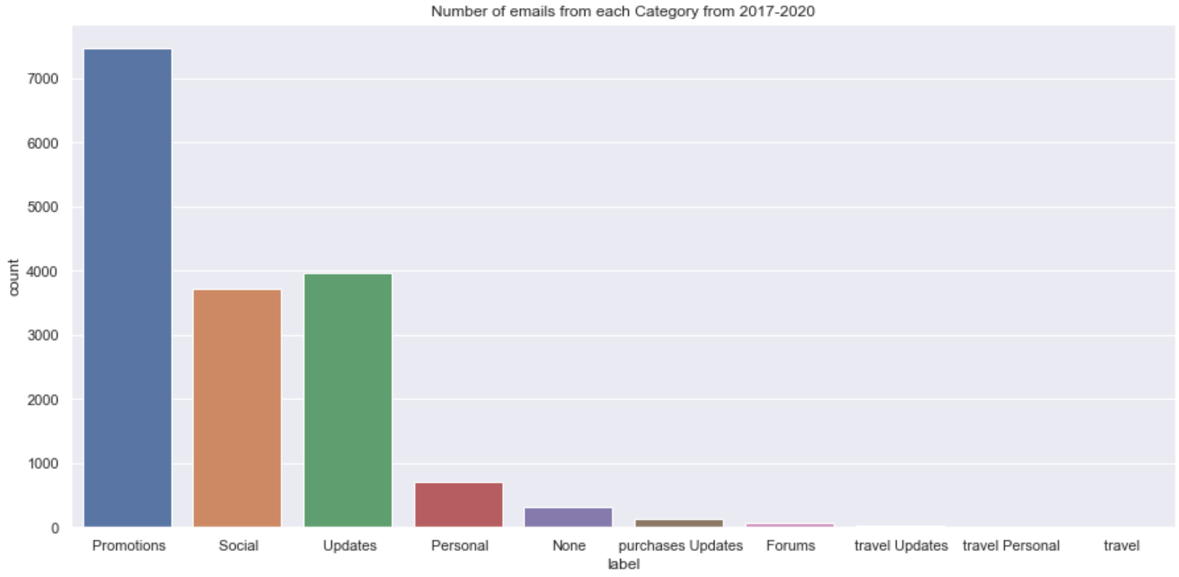
Categorize your emails based on labels and plot them

In [14]:

```
data["label"] = list(map(lambda x: "".join([ele.replace("Category", "").replace("\n","") fo
data["label"] = data["label"].replace({"":"None","\n":"" })
```

In [15]:

```
sns.set(rc={'figure.figsize':(15,7)})
ax = sns.countplot(x = data["label"])
ax = ax.set_title("Number of emails from each Category from 2017-2020")
```



Question 3

Explore the data and identify two other possible insights that you can get from the data.

Exploratory Analysis

My Sent Mail stats

In this section, I want to find out which month and what time I send the most emails, and how that has changed over 3 years.

In [16]:

```
#extract sent emails
sent = data[data["sent"]]
sent["date"].head()
```

Out[16]:

```
20     Mon, 20 Jan 2020 10:28:34 +0000
148    Wed, 15 Jan 2020 02:59:39 +0000
193     Mon, 9 Dec 2019 16:23:50 +0900
203    Mon, 20 Jan 2020 10:28:31 +0000
240    Thu, 16 Jan 2020 15:48:16 +0800
Name: date, dtype: object
```

In [17]:

```
#extract month, day and time information
import dateutil.parser as parser
sent["day"] = list(map(lambda x : parser.parse(x).weekday(), sent["date"]))
sent["month"] = list(map(lambda x : parser.parse(x).month, sent["date"]))
sent["time"] = list(map(lambda x: parser.parse(x).time(), sent["date"]))
```

In [18]:

```
st = sent.groupby(['year','month'],as_index=False).count()
```

In [19]:

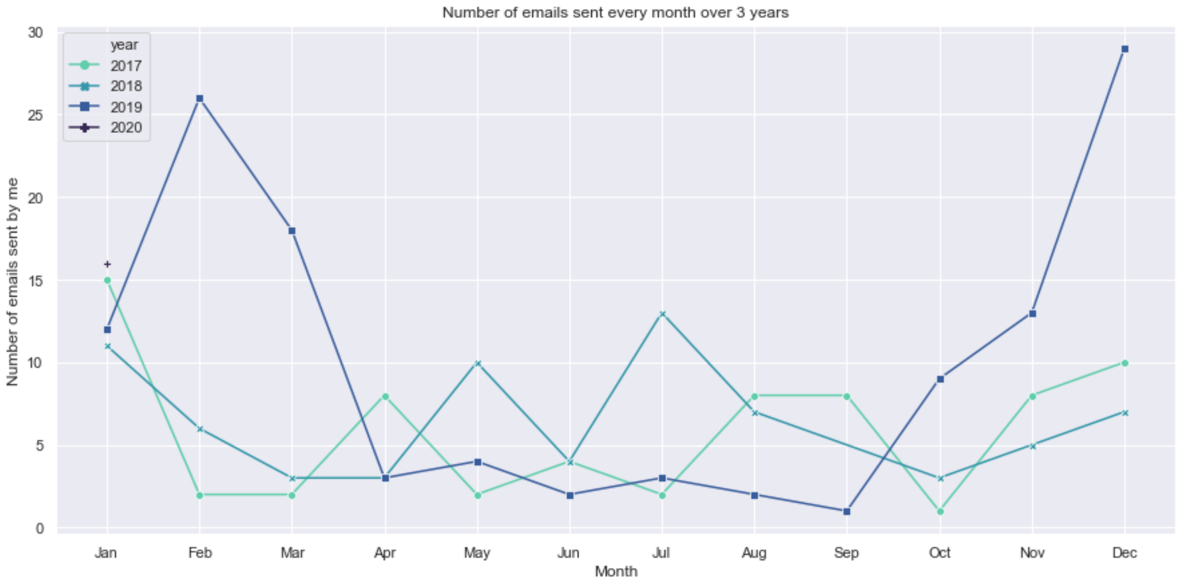
```
st["month"] = st["month"].replace({
    1:"Jan",
    2:"Feb",
    3:"Mar",
    4:"Apr",
    5:"May",
    6:"Jun",
    7:"Jul",
    8:"Aug",
    9:"Sep",
    10:"Oct",
    11:"Nov",
    12:"Dec"
})
st.head()
```

Out[19]:

	year	month	subject	from	date	labels	to	sent	label	day	time
0	2017	Jan	15	15	15	15	15	15	15	15	15
1	2017	Feb	2	2	2	2	2	2	2	2	2
2	2017	Mar	2	2	2	2	2	2	2	2	2
3	2017	Apr	9	9	9	9	8	9	9	9	9
4	2017	May	3	3	3	3	2	3	3	3	3

In [20]:

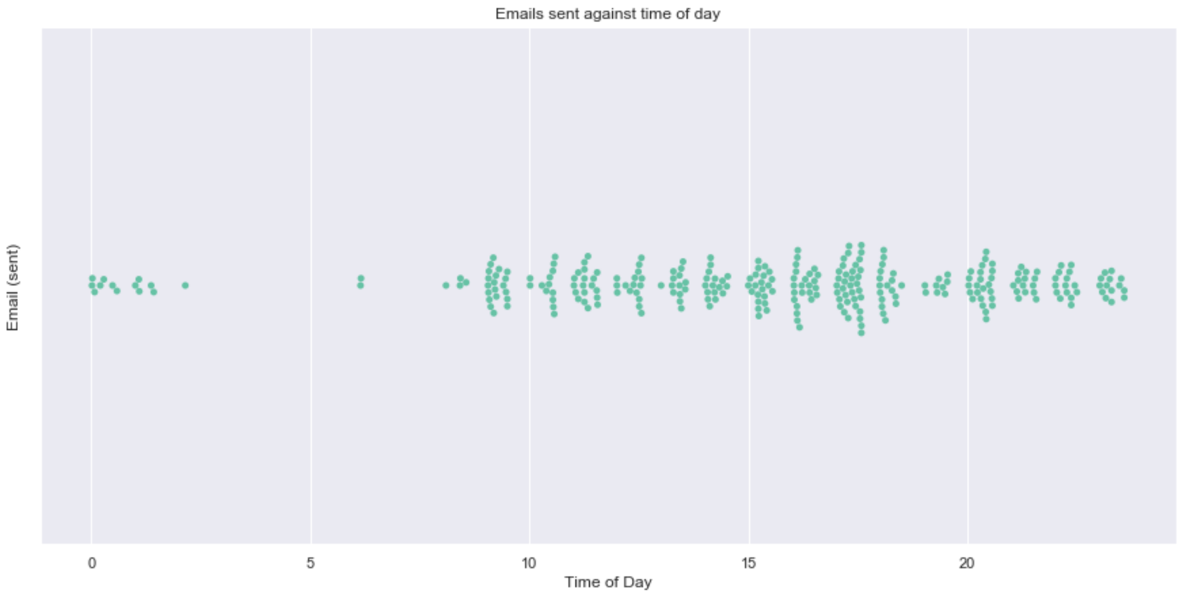
```
ax = sns.lineplot(y = st["to"],
                  x= st["month"],
                  hue=st["year"],
                  err_style=None,
                  palette=sns.color_palette("mako_r", 4),
                  style=st["year"],
                  markers=True,
                  dashes = False,
                  sort=False)
ax = ax.set(xlabel="Month",ylabel="Number of emails sent by me", title = "Number of emails
```



From the plot above (Number of Emails sent by me against Month and Year) - I sent the most emails in December 2019, followed by February 2019. This pattern seems consistent across the 3 years 2017, 2018 and 2019, where I send more emails at the start and end of the year compared to the other months.

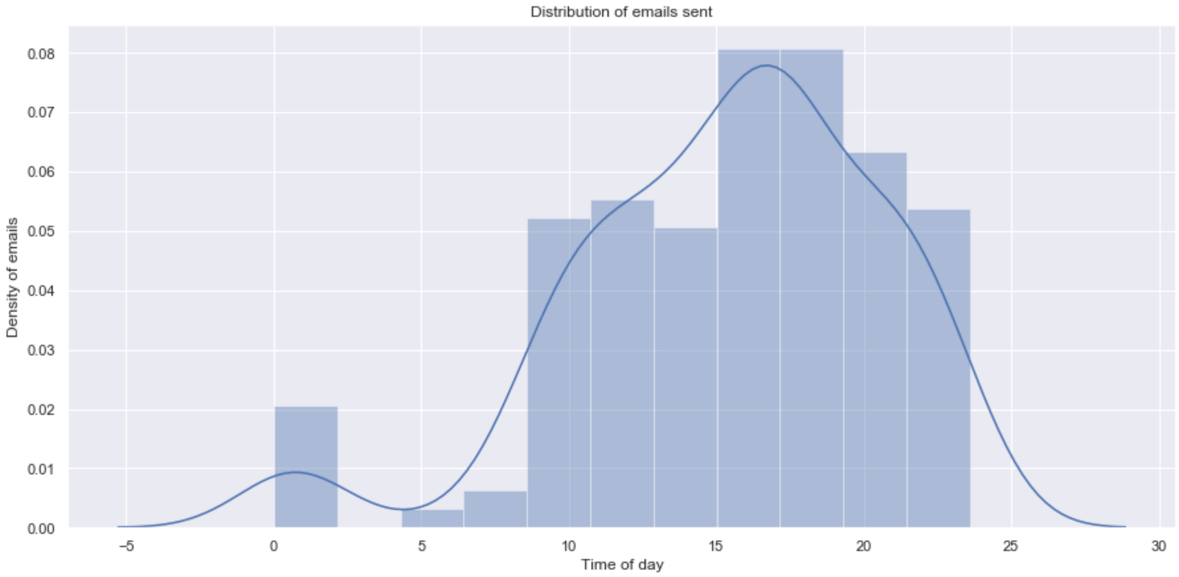
In [21]:

```
#extracting timestamp, correcting for timezone differences
sent["time"] = sent["date"].apply(lambda x: float(x[-14:-12]) + 0.01*float(x[-11:-9]) + 8 -
sent["time"] = sent["time"]%24
#sent
ax = sns.swarmplot(sent["time"],palette="Set2", dodge=True)
ax = ax.set(ylabel="Email (sent)", xlabel = "Time of Day", title = "Emails sent against tim
```



In [22]:

```
ax = sns.distplot(sent["time"])
ax = ax.set(title = "Distribution of emails sent", xlabel ="Time of day", ylabel = "Density
```



Do I read Promotional emails?

In this part I explore the promotional emails I received over the past 3 years, and if I read them. I first filter out all the promotional emails and split them into read or unread.

In [23]:

```
#filter promo emails
promos = data[data["label"].str.contains("Promotions")]
promos["read"] = promos["labels"].str.contains("Opened")
promos = promos[['from','subject','read','year']]
promos.head()
```

Out[23]:

	from	subject	read	year
0	Thomas Whitt	0	False	2020
2	Ronald Richardson	2	False	2020
3	Howard Kamiya	3	False	2020
4	Vernon Jacobs	4	False	2020
7	Vernon Jacobs	7	False	2020

Then I group the data by their promoter, read or not and year. The subject is aggregrated with count to acheive the following dataframe.

In [24]:

```
pt = promos.groupby(['from','read','year'],as_index=False).count()
#since there are too many senders to plot, just find the top 20 promotion senders by volume
counts = pt.groupby(["from"]).sum()["subject"].sort_values(ascending=False)
pt = pt[pt["from"].isin(counts[0:20].index.to_list())]
pt.head()
```

Out[24]:

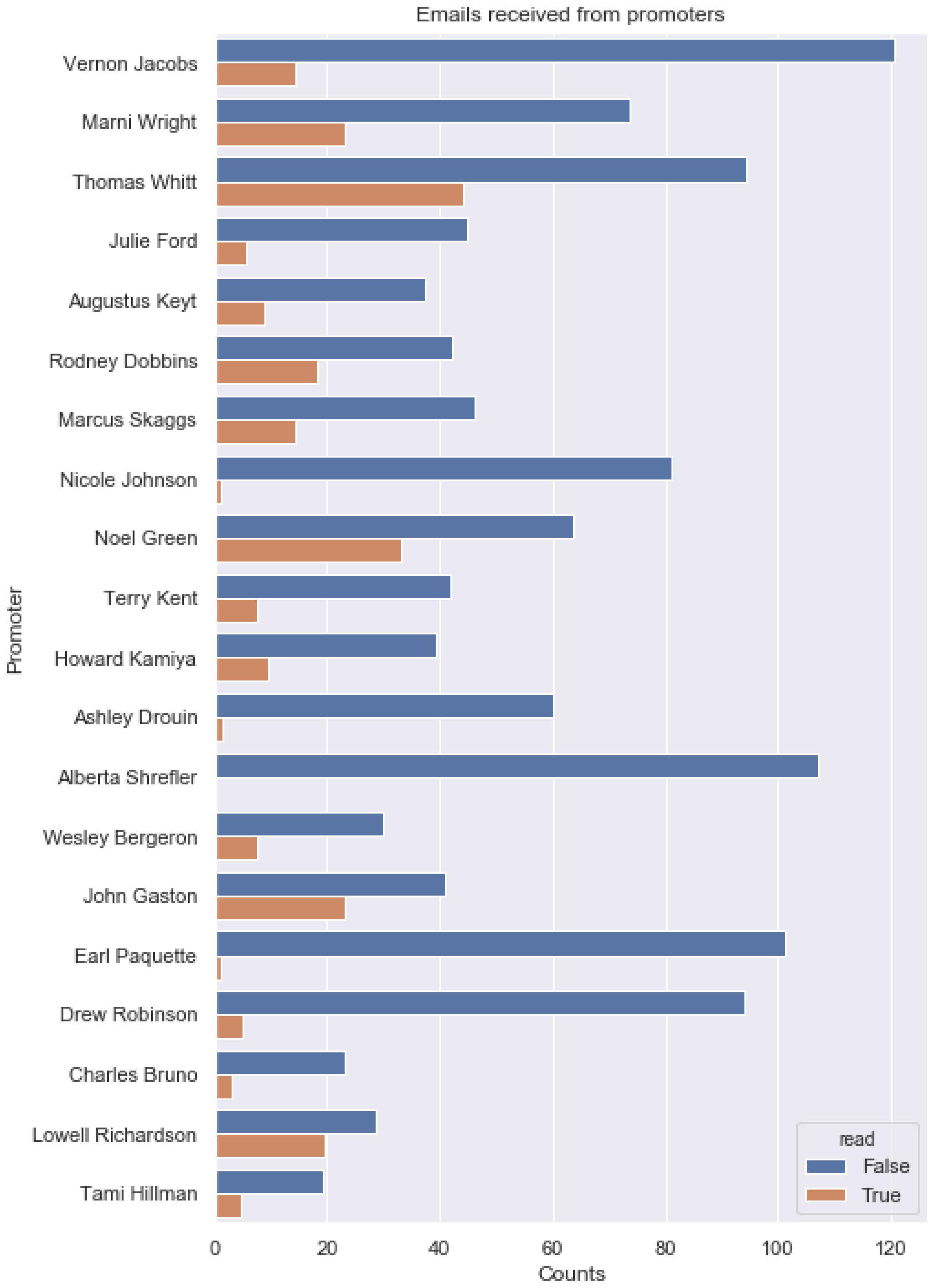
	from	read	year	subject
9	Alberta Shrefler	False	2017	107
38	Ashley Drouin	False	2017	38
39	Ashley Drouin	False	2018	82
40	Ashley Drouin	True	2017	2
41	Ashley Drouin	True	2018	1

In [25]:

```
order_most = counts[0:20].index.to_list()
```

In [26]:

```
sns.set(rc={'figure.figsize':(7,12)})
ax = sns.barplot(y = pt['from'], x=pt['subject'],hue=pt['read'],ci=None, order = order_most
ax = ax.set(xlabel = "Counts", ylabel = "Promoter", title = "Emails received from promoters
```



We observe that I received the most promotional emails from James Roy. However it can also be observed that I only read a small number of emails from James Roy. Therefore, I would like to find out whose promotions (among the top 20 promoters) I read the most. That is, whose promotions have the highest read rate by me.

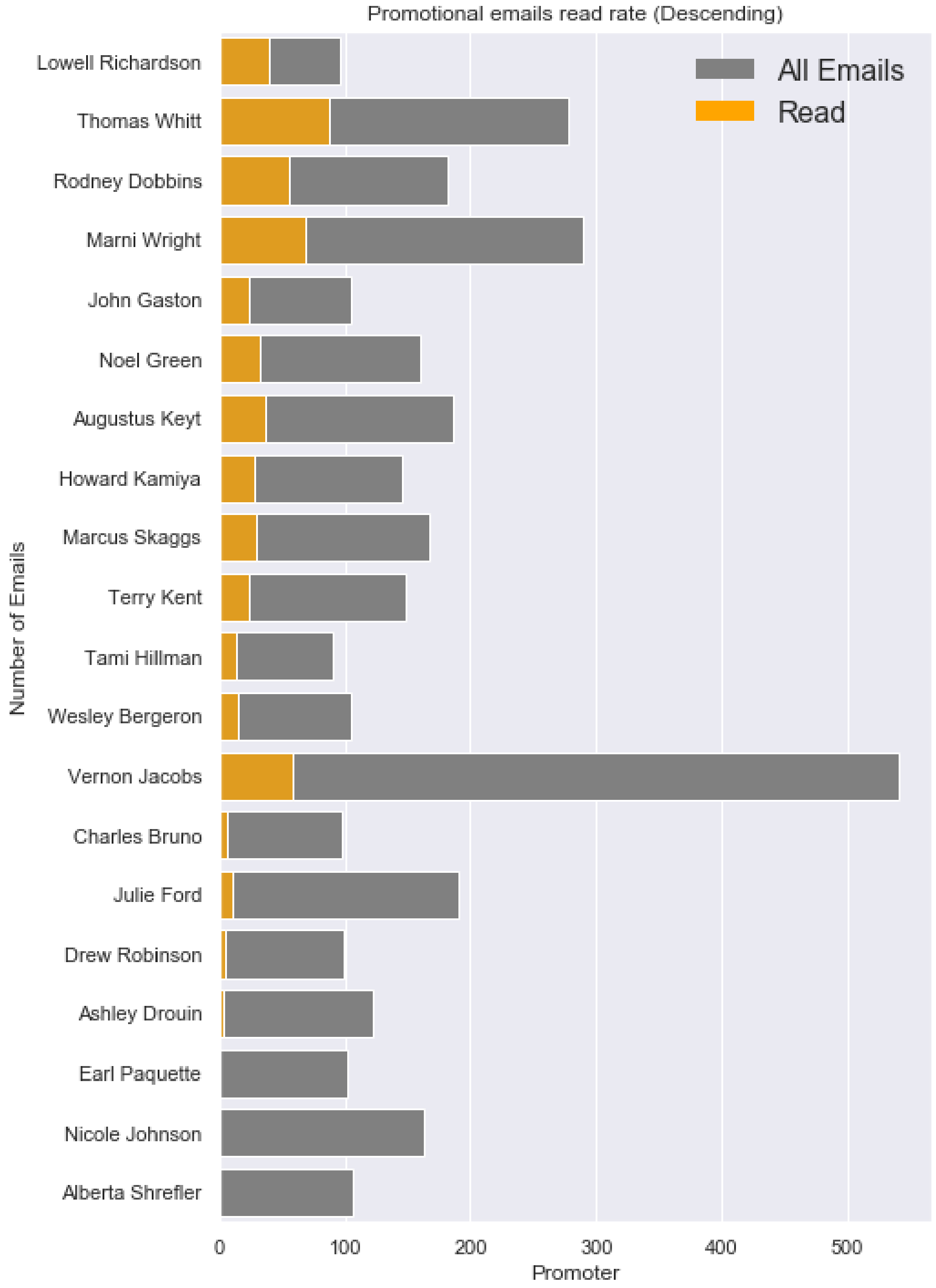
In [27]:

```
#create read counts for each promoter, and emails from each promoter
pt_read = pt[pt["read"] == True].groupby(["from"]).sum()[["subject"]]
pt_all = pt.groupby(["from"]).sum()[["subject"]]
rate = pt_read/pt_all

#order the bar plot by rate of reading emails
order = rate.sort_values(by = "subject", ascending = False).index
```

Plotting read against total emails,

```
In [28]:
```

```
#plot stacked bar chart ordered by read rate of promotional emails
bot = sns.barplot(y = pt_all.index, x=pt_all['subject'],ci=None, order = order, color = "gr
ax = sns.barplot(y = pt_read.index, x=pt_read['subject'],ci=None, order = order, color = "o
ax = ax.set(title = "Promotional emails read rate (Descending)", xlabel = "Promoter", ylabe

import matplotlib.pyplot as plt
topbar = plt.Rectangle((0,0),1,1,fc="orange", edgecolor = 'none')
bottombar = plt.Rectangle((0,0),1,1,fc='gray',  edgecolor = 'none')
l = plt.legend([bottombar, topbar], ['All Emails', 'Read'], loc=1, ncol = 1, prop={'size':1
l.draw_frame(False)
```



We observe that the I am most likely to read emails from Lowell Richardson