Gmail Explorer Visualizer

By nickesc / N. Escobar

In [106…

```
import os
import csv
import base64
import pandas as pd
import seaborn as sns
import numpy as np
import sklearn
import matplotlib.pyplot as plt
import datetime
from scipy.stats import zscore
from matplotlib import rcParams

# figure size in inches
rcParams['figure.figsize'] = 10,10
sns.set(rc={'figure.figsize':(11.7,8.27)})
color = ['blue','green','orange','red']

from IPython.display import clear_output, display
#from ipywidgets import *
#from tkinter import Tk, filedialog
#from math import floor
```

Looking at the data!

Finally! Our data is in a format we can really work with! We're only going to worry about the received messages for now, but we still have the chats, sent and drafts if we want them later. This is a fairly preliminary look at the data, most of my time was spent on getting collection and imputing done -- I definitely want to continue to work on this project. I have a few places I'd like to go though, and some interesting thoughts based on the current visualizations. Lets load the emails into the notebook (dropping an extra, unnamed column for some reason which I'm sure is my fault):

In [3]:

```
messages = pd.read_csv("../imputer/imputed/recieved.csv")
messages.drop(messages.columns[messages.columns.str.contains('unnamed',case = False)],axis
messages.head()
```

Out[3]:

	id	threadId	from	delivered	
0	1340953ff42a62c8	1340953ff42a62c8	accounts@mochigames.com	josephgoodman85@gmail.com	joseph
1	136dc4cb8ea460d1	136dc4cb8ea460d1	pottermore@mail.pottermore.com	josephgoodman85@gmail.com	joseph
2	136dc4ddab520861	136dc4ddab520861	pottermore@mail.pottermore.com	josephgoodman85@gmail.com	joseph
3	136ebd147fcd9bba	136ebd147fcd9bba	pottermore@mail.pottermore.com	josephgoodman85@gmail.com	joseph

	id	threadId	from	delivered	
4	13750efdfd5fd883	13750efdfd5fd883	pottermore@mail.pottermore.com	josephgoodman85@gmail.com	joseph

In [5]:

```
msg_filtered = messages[(messages['delivered'] == "josephgoodman85@gmail.com") | (messages
msg_filtered
```

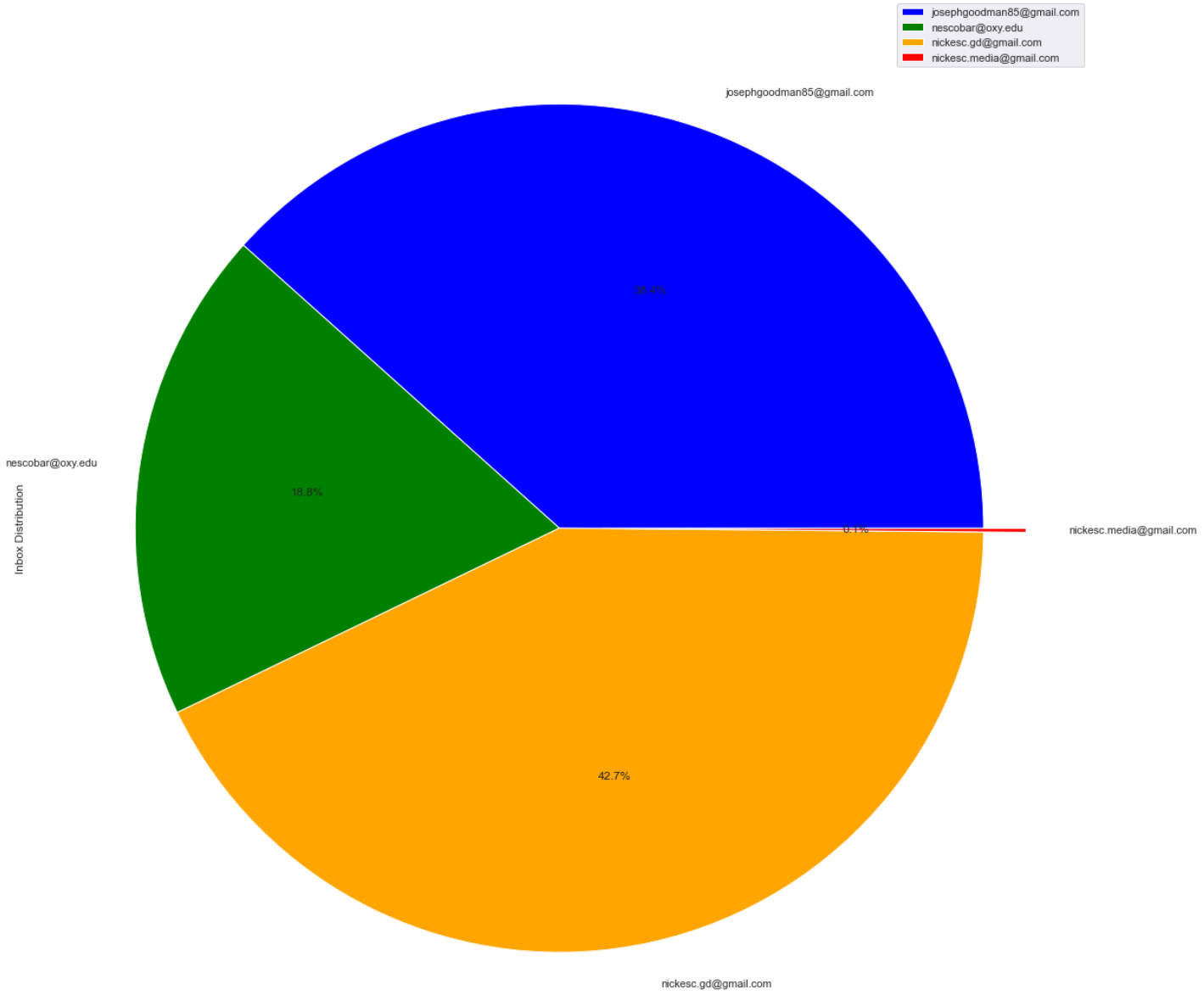
Out[5]:

	id	threadId	from	delivered	
0	1340953ff42a62c8	1340953ff42a62c8	accounts@mochigames.com	josephgoodman85@gmail.com	jo
1	136dc4cb8ea460d1	136dc4cb8ea460d1	pottermore@mail.pottermore.com	josephgoodman85@gmail.com	jo
2	136dc4ddab520861	136dc4ddab520861	pottermore@mail.pottermore.com	josephgoodman85@gmail.com	jo
3	136ebd147fcd9bba	136ebd147fcd9bba	pottermore@mail.pottermore.com	josephgoodman85@gmail.com	jo
4	13750efdfd5fd883	13750efdfd5fd883	pottermore@mail.pottermore.com	josephgoodman85@gmail.com	jo
...	...	...	...	...	
87430	17f14a27dd9a9b7d	17f14a27dd9a9b7d	no-reply@mail.instagram.com	josephgoodman85@gmail.com	jo
87431	17f14af10caa271c	17f14af10caa271c	noreply@uber.com	nickesc.gd@gmail.com	
87432	17f1575c1426497e	17f1575c1426497e	noreply@redditmail.com	josephgoodman85@gmail.com	jo
87433	17f15a705a5f7989	17f15a705a5f7989	noreply@uber.com	nickesc.gd@gmail.com	
87434	17f15ab170b8f99e	17f15a705a5f7989	noreply@uber.com	nickesc.gd@gmail.com	

87058 rows × 14 columns

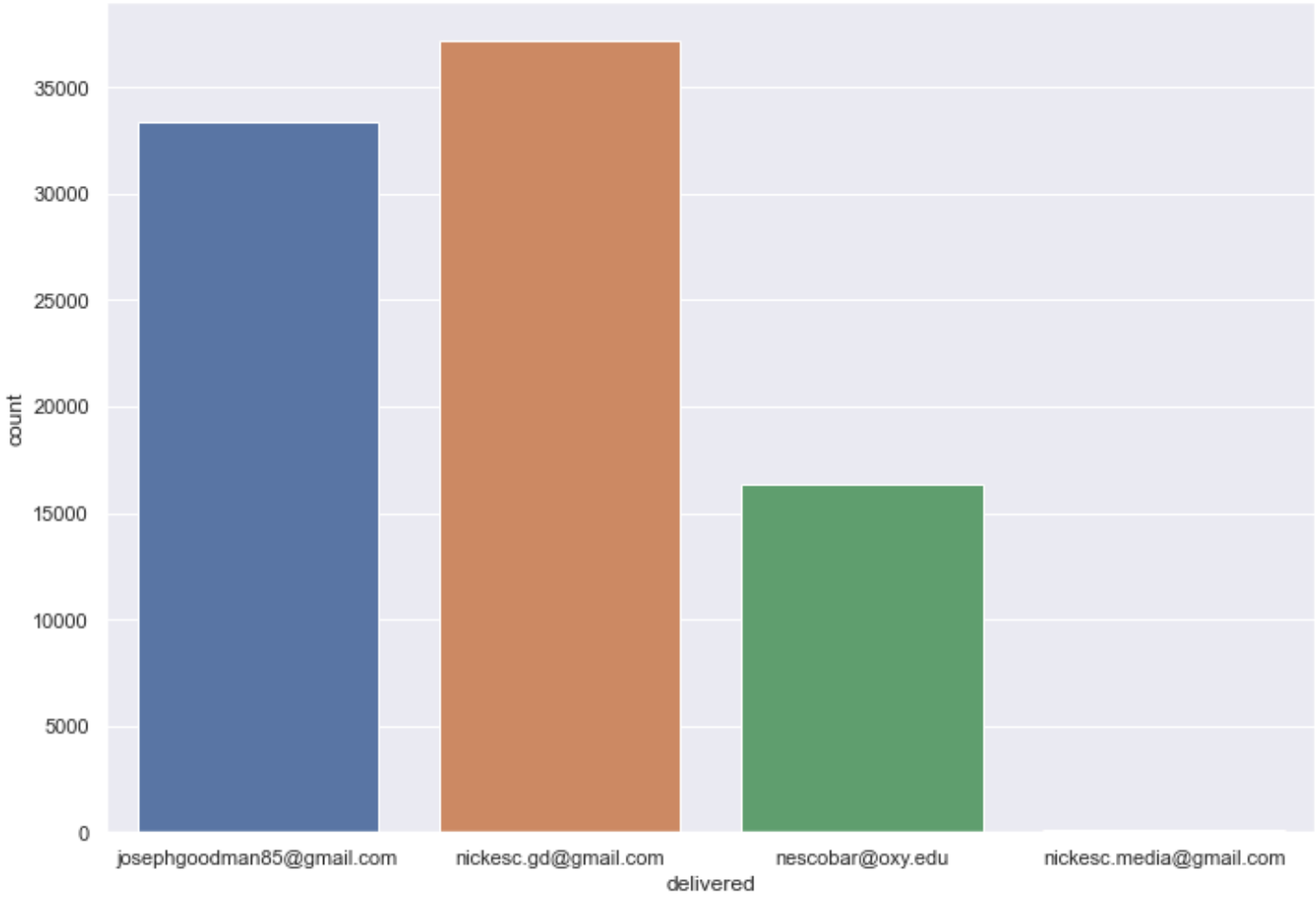
In [110…

```
#colors = sns.color_palette('tab10')
# plt.pie(.value_counts(), colors = sns.color_palette('tab10'),labels = msg_filtered.keys

pie1 = msg_filtered.groupby(['delivered']).count().plot(kind='pie', y='sizeEstimate', figs

#plt.show()
```



In [8]:
```
sns.countplot(x='delivered',data=msg_filtered)
```

Out[8]:
```
<AxesSubplot:xlabel='delivered', ylabel='count'>
```

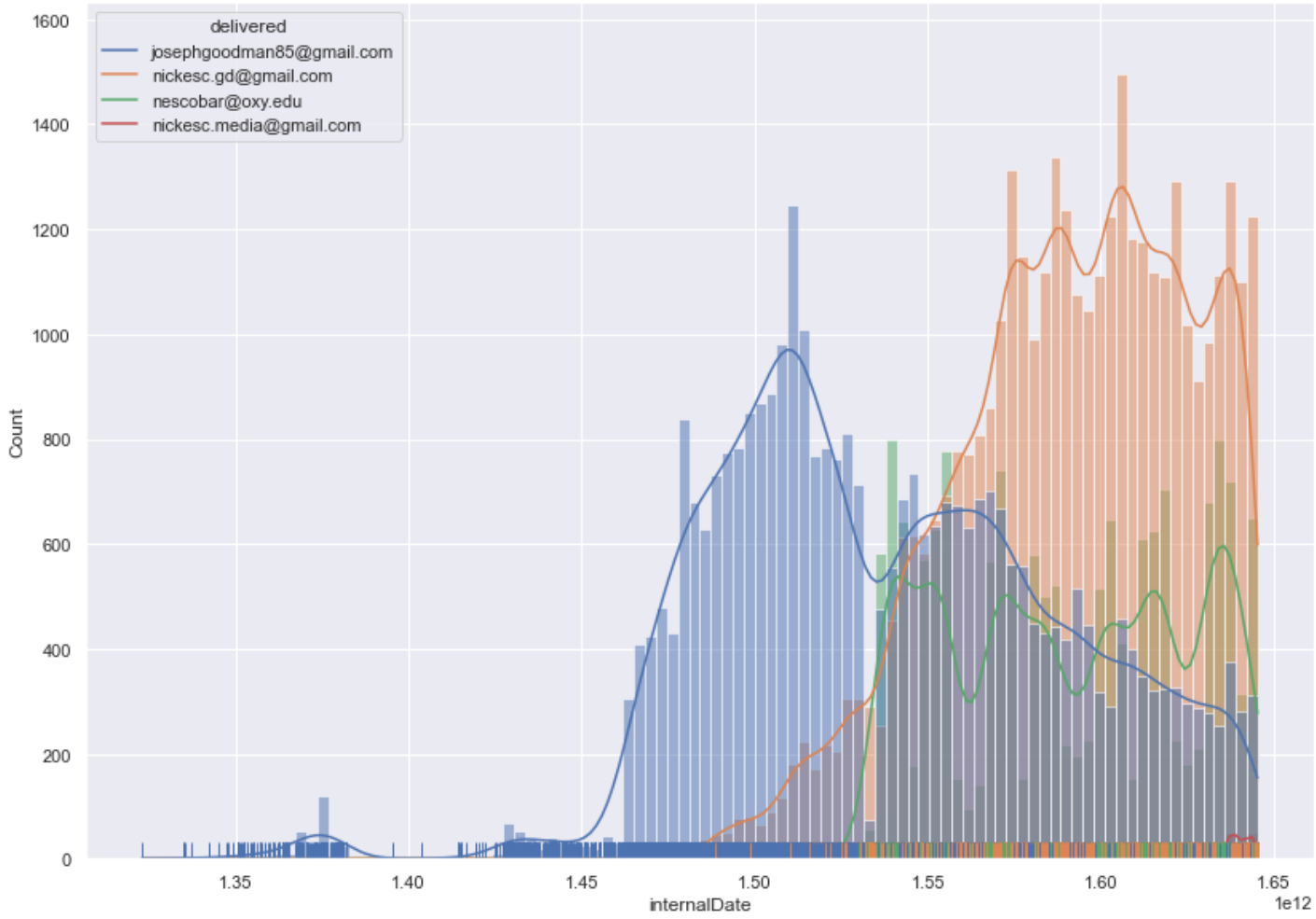


First, let's take a look at the overall distribution of emails in my inbox. We can see that the vast majority of emails have been sent to `josephgoodman85@gmail.com` (JG) and `nickesc.gd@gmail.com` (GD). A relatively smaller amount comes from my Oxy email, `nescobar@oxy.edu` and virtually no mail is sent to `nickesc.media@gmail.com` (NM). Let's take a slightly closer look though.

Tracking email account use over time

In [143…

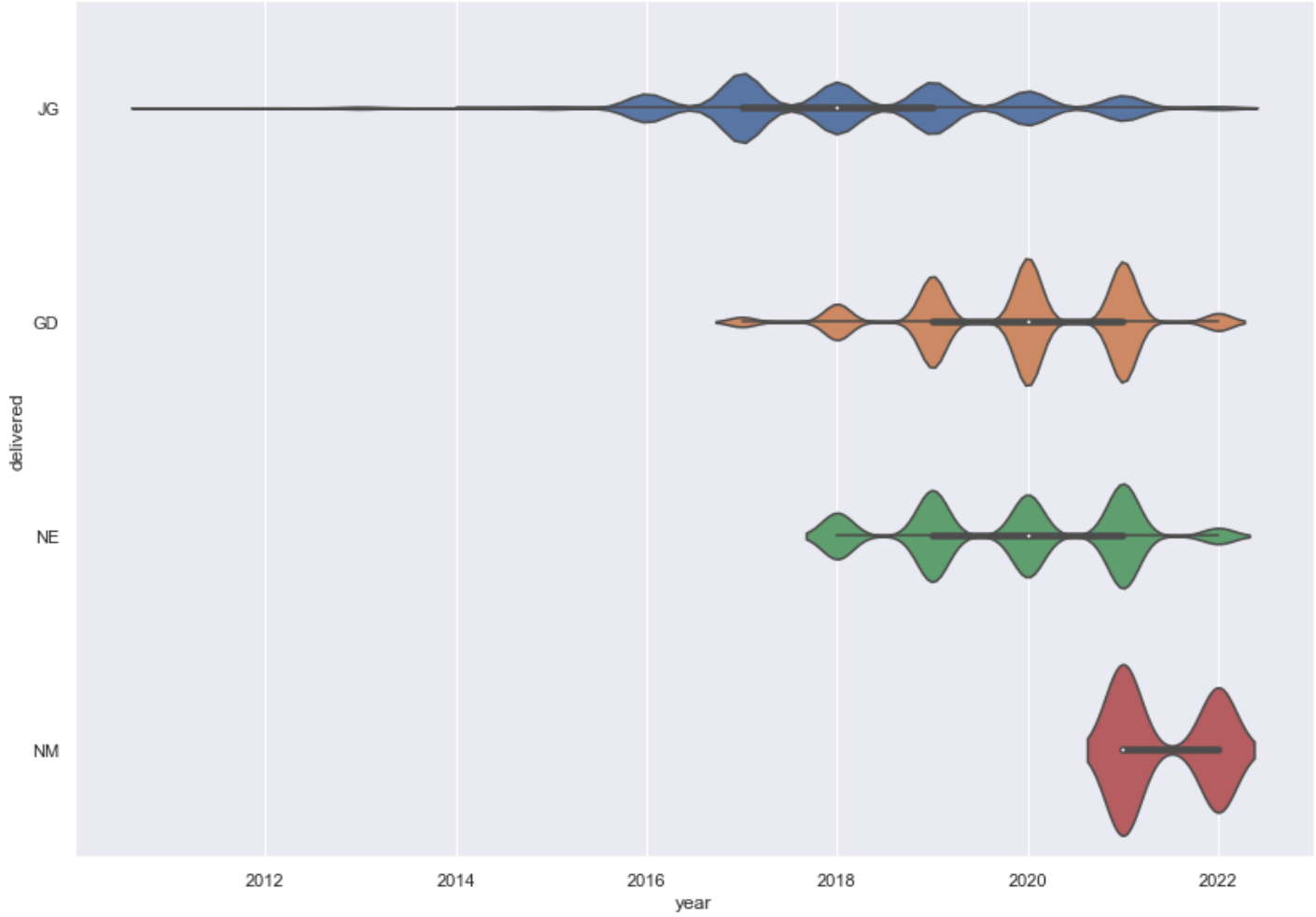
```
sns.histplot(data=msg_filtered.drop(columns=['body']), x="internalDate", hue = 'delivered'
sns.rugplot(data=msg_filtered.drop(columns=['body']), x="internalDate", hue = 'delivered',
plt.figure(figsize = (15,8))
plt.show()
```




```
<Figure size 1080x576 with 0 Axes>
```

In [144…

```
ax = sns.violinplot(x="year", y="delivered", data=msg_filtered.drop(columns=['body']))
#ax.tick_params(left=False, bottom=False, top=True, right=True)
#ax.set_tickLabels(left=False, bottom=False, top=True, right=True)

#ax.axes.xaxis.set_ticklabels([])
ax.axes.yaxis.set_ticklabels(['JG', 'GD', 'NE', 'NM'])
plt.figure(figsize = (15,8))
plt.show()
```




```
<Figure size 1080x576 with 0 Axes>
```

Looking at this, we get a little more information. We start to see some trends forming in my email use -- this graph is over time, starting in Dec 2011 going until Feb 2022. First off, we can see the distribution over time of the accounts, with JG starting with the most use, until GD is introduced in 2017, where we see a sharp drop off in JG's emails, and which continue to lower as GD goes up. You can see in the rugplot at the bottom exactly where the shift occurs.

> Those trends all track with my real life email use -- I transitioned my main email to one that actually had my name instead of *'Joseph Goodman.' I moved most things to the new email during the beginning and middle of 2018, which is when we see a sharp increase, and started using JG mostly for spam.

Next, we can see the introduction of NE, which starts at a high use already with hardly any build up. For the most part, that number stays consistent, but with a dip in 2020. The addition of this email address seems to have had a minor impact on GD, the main email at the time.

> Again, we can track this with real life: thsi was when I started at Oxy and got my Oxy email, which came with a ton of subscriptions to school newsletters already. Those emails continued pretty consistently while I was at school, but when we were sent home in 2020 to isolate, the emails decreased because nothing was happening on campus at the time. They pick back up when I come back to LA though, which explains why they rebound to exactly where they were before.

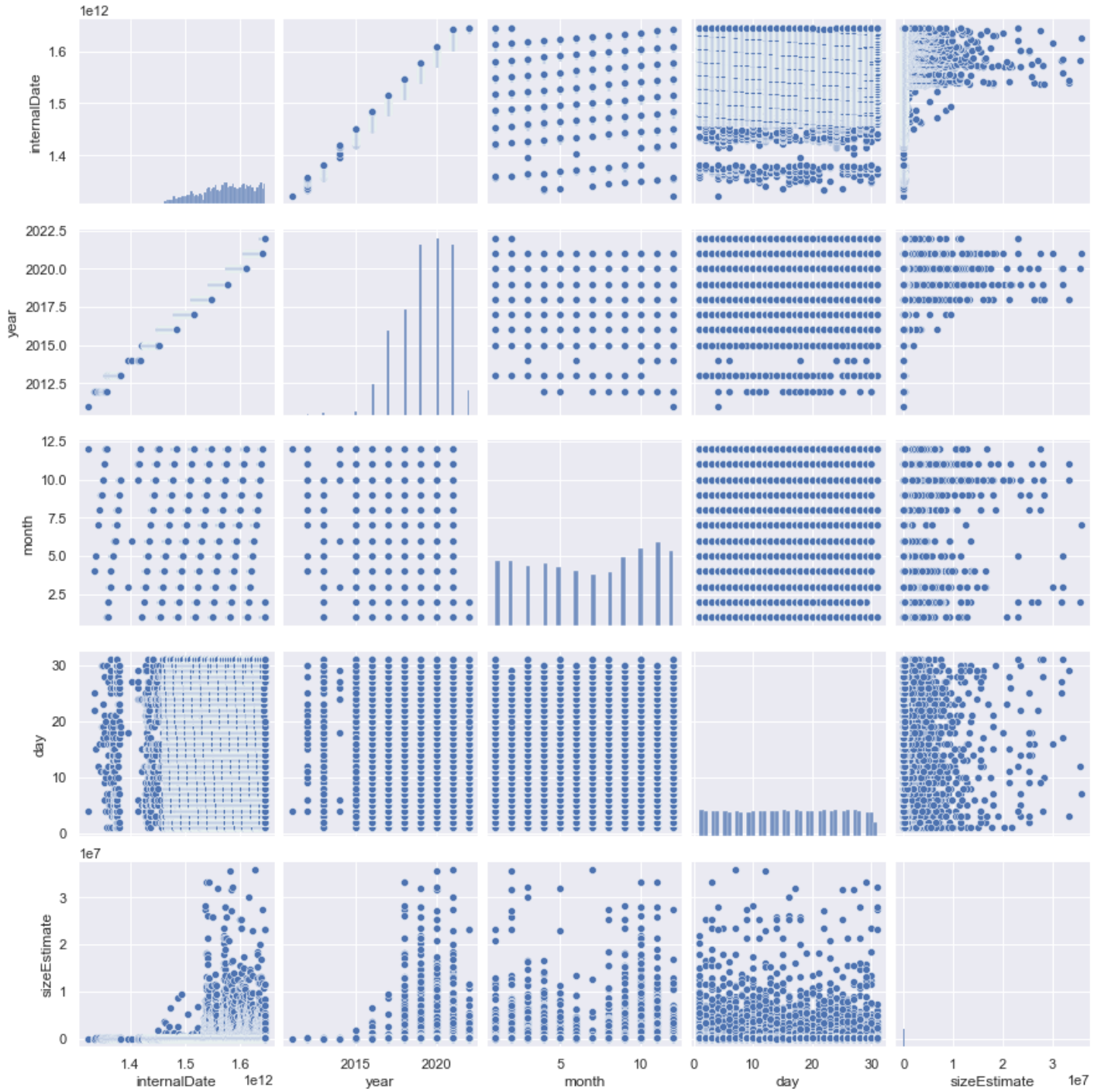
The last addition, NM, is hard to see on the first graph, but we can tell that it was introduced in mid 2020-2021 from the second. It's emails are only spread out over two years, but it has under 100 emails in its inbox.

> This email was used mostly as a bot for a website I set up that would send email notifications to people when I updated on the site. It isn't really an active email account, so it doesn't have very many emails, it's used for projects and a handful of other accounts.

Looking at frequency, size, and patterns

In [21]:
```
sns.pairplot(data=msg_filtered.drop(columns=['body']))
```

Out[21]:
```
<seaborn.axisgrid.PairGrid at 0x7fcdec1ddb80>
```

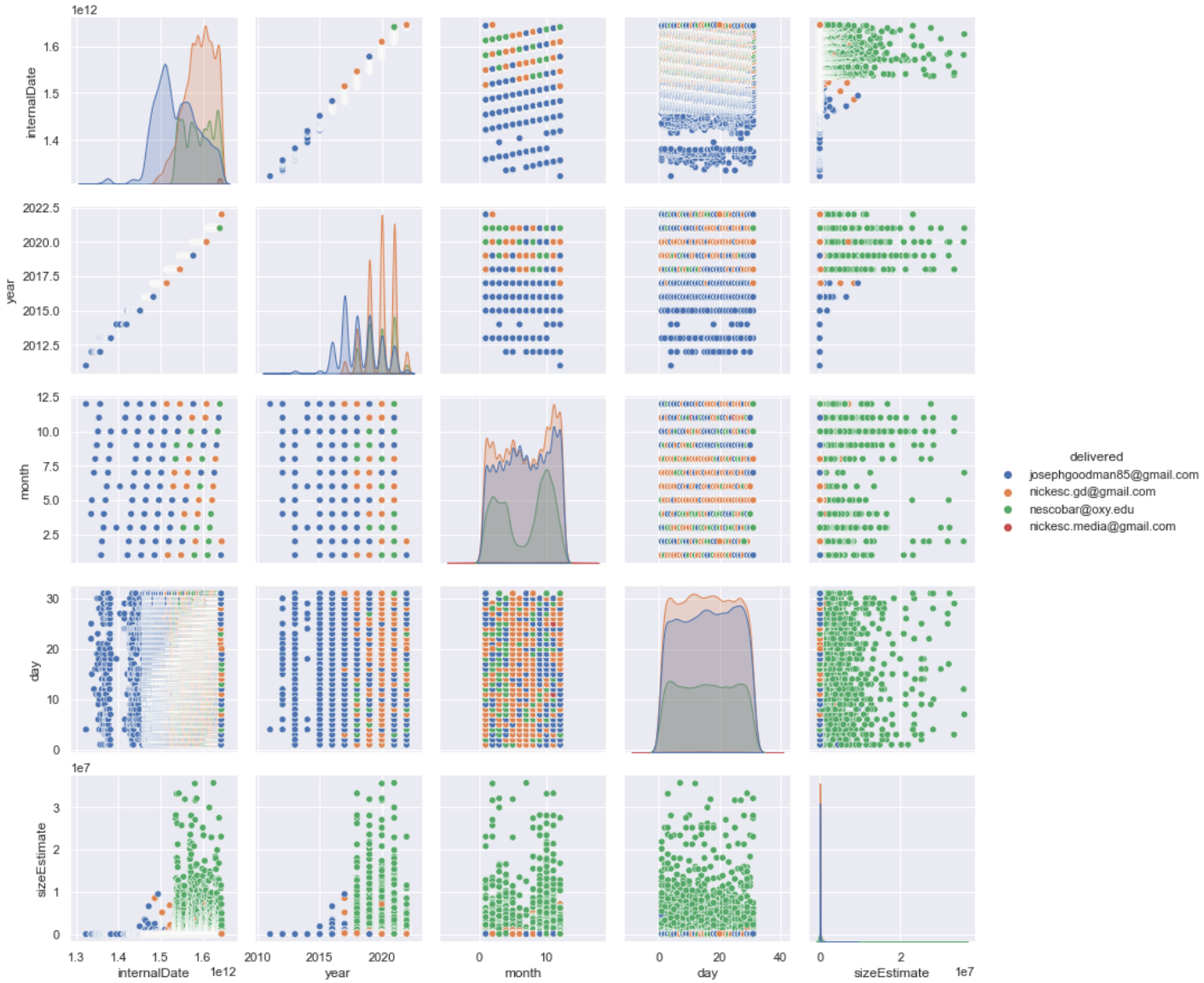


The two pairplots here were some of the most interesting parts of this. The first time I looked at them it was the completely blue version, without account differentiation. I looked at the year-over-year graph, where I saw the first part of the shape of the graph above. It also showed me how there are months I get more emails, at the end of the year and dipping around the summer, presumably because of things like holiday advertising. They also showed a relationship between time and size that confused me. In the top let, you

can see that at some point the size of emails skyrockets -- I had no idea why and wanted to get to the bottom of it. SO I looked deeper at the individual accounts, and found some answers.

In [9]:
```
sns.pairplot(msg_filtered,hue='delivered')
```

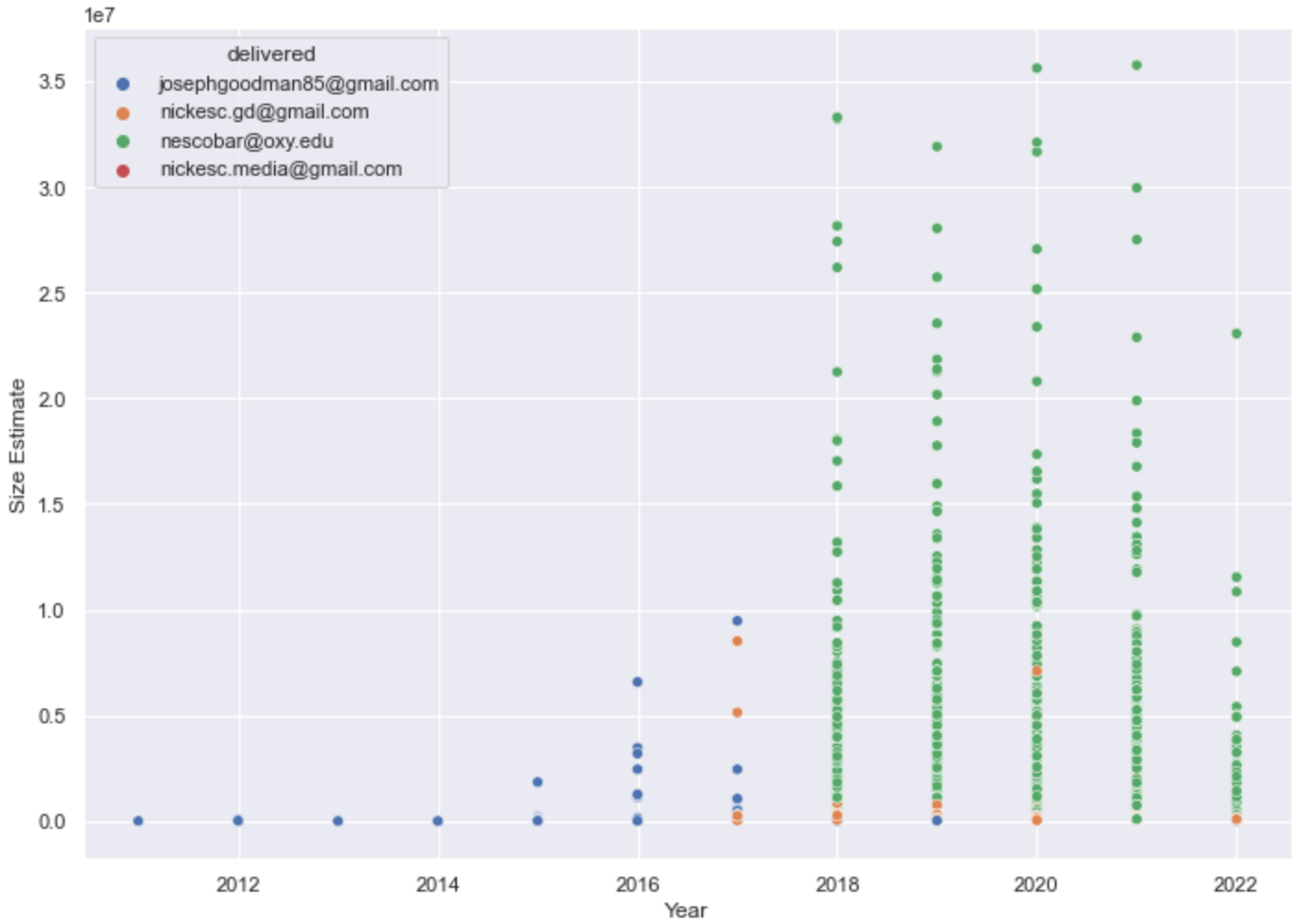
Out[9]:
```
<seaborn.axisgrid.PairGrid at 0x7fce0ca64f70>
```



In the second pairplot, we can see a lot more information -- we see the distribution for individual accounts over time, the basis for the graph above, and we see the dip in months explained a little on the graph in the middle, monthly distribution -- during the summer, NE doesn't really get any mail because I'm out of school. We can also see I get about 80 new emails daily, but that JG and GD sit about equal, while NE halves them and NM is no where to be seen. You can also see the transition to GD from JG in the year over month graph.

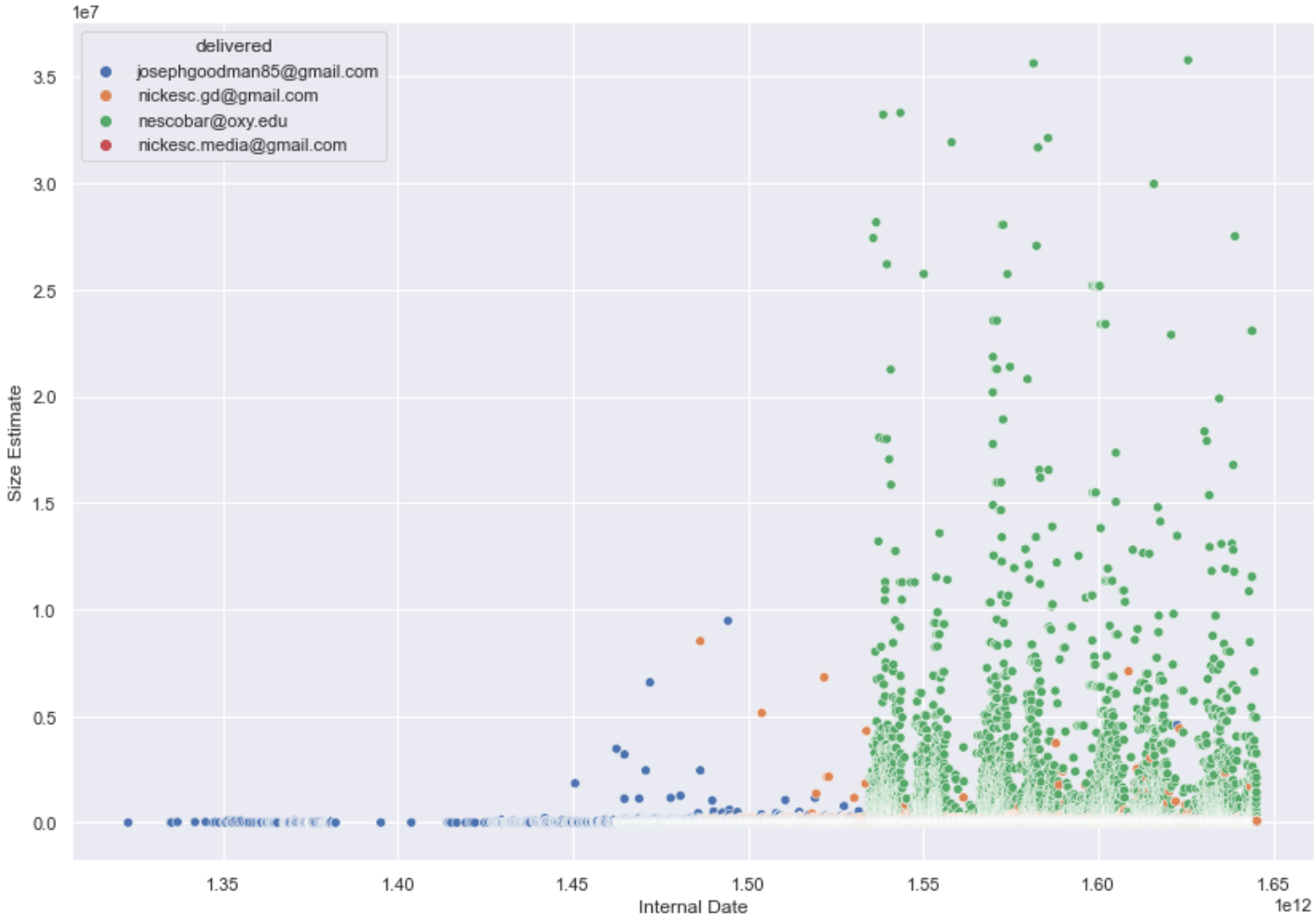
And we get some more clarity on the size question -- the even that increased sizes was the introduction on NE. Emails to NE, presumable from Oxy, as they are the unique emails on that account, and they take up significantly more space, by about 6 times on average. I don't know what it is about those emails, but they are not space efficient.

In [69]:
```
sns.scatterplot(x = msg_filtered['year'],y = msg_filtered['sizeEstimate'], hue=msg_filtere
plt.xlabel('Year')
plt.ylabel('Size Estimate')
plt.show()
```



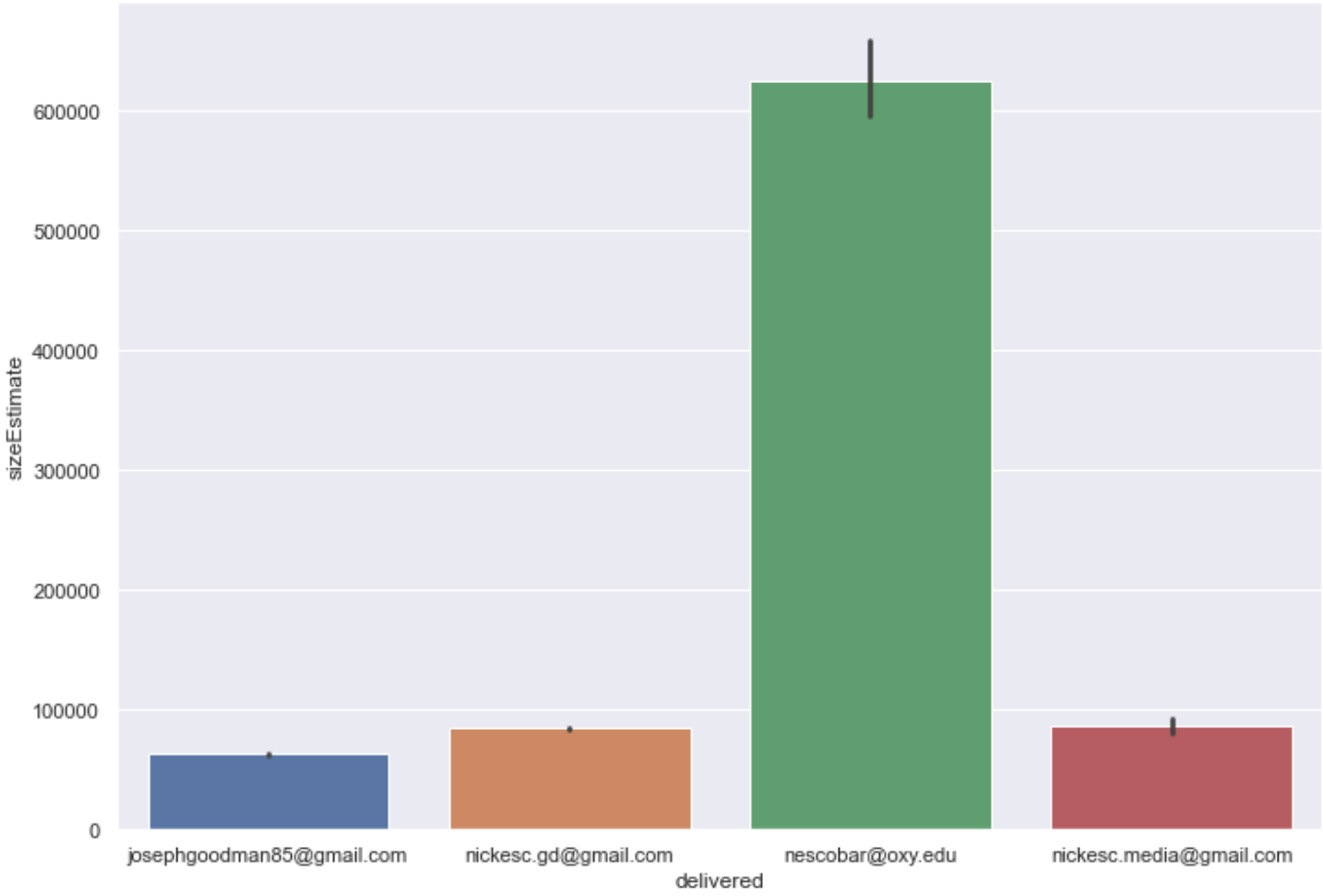
In [145…

```
sns.scatterplot(x = msg_filtered['internalDate'],y = msg_filtered['sizeEstimate'], hue=msg
plt.xlabel('Internal Date')
plt.ylabel('Size Estimate')
plt.show()
```



In [11]:
```
sns.barplot(x='delivered',y='sizeEstimate',data=msg_filtered)
```

Out[11]:
```
<AxesSubplot:xlabel='delivered', ylabel='sizeEstimate'>
```



Body and subject relationship

In [4]:

```
def getDecodeLen(string):
    try:

        decode = base64.urlsafe_b64decode(string).decode('utf-8')
    except:
        decode = ""
        #print("failed on", string)
    return len(decode)

def getLen(string):
    return len(string)
```

In [25]:

```
lengs = messages[['body', 'subject']].copy()

lengs['body']=lengs['body'].apply(getDecodeLen)
lengs['subject']=lengs['subject'].apply(getLen)

lengs
```

Out[25]:

	body	subject
0	787	35
1	8516	23
2	6671	50
3	6992	26
4	2318	52

	body	subject
...	...	...
87430	19858	58
87431	113185	34
87432	98530	49
87433	72543	42
87434	78840	42

87435 rows × 2 columns

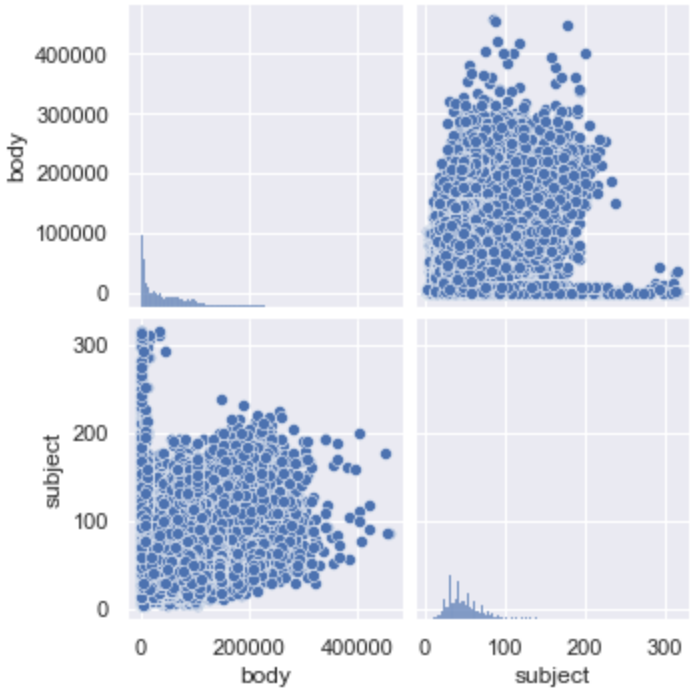
Finally, we come to the last part, a small exploration of the content of the messages. In the next few graphs, we compare the length of the message's subject with the length of its body and calculate the correlation coefficient. We see a really strange shape when we look at these graphs, made up of three distinct parts -- messages with a body > ~10500 characters have a smaller subject size, messages with a body < ~10500 characters have a larger subject size, and messages with a body < 300 characters have an even bigger subject size. I have absolutely no idea why it looks like that, but I imagine it has to do with the most frequent emails I get on each account.

In [152…

```
lengs.drop(lengs['body'].idxmax(), inplace=True)
lengs.drop(lengs['body'].idxmax(), inplace=True)
sns.pairplot(lengs)
```

Out[152…

```
<seaborn.axisgrid.PairGrid at 0x7fce0a14f7c0>
```



In [129…

```
corr = lengs.corr(method="pearson")["body"][1]
print("the correlation coefficient is", corr)
plt.scatter(lengs['body'], lengs['subject'])
plt.xlabel('body (chars)')
plt.ylabel('subject (chars)')
plt.show()
```

```
the correlation coefficient is 0.1579934970411838
```

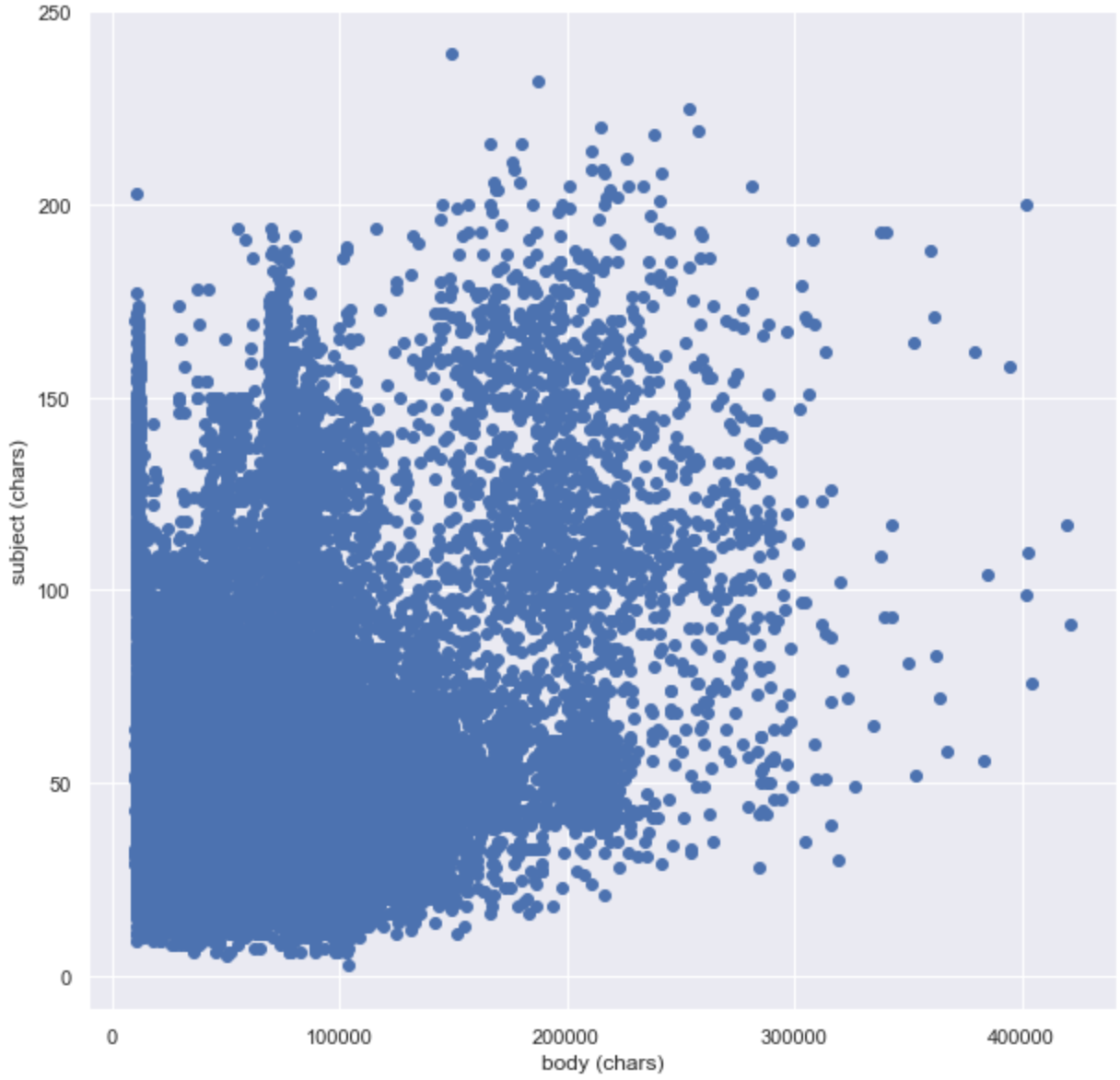
In [130…

```
lengsBig = lengs.loc[lengs['body']>10500]
lengsSmall = lengs.loc[lengs['body']<10500]
lengsSmaller = lengs.loc[lengs['body']<300]

z_scores = zscore(lengsBig)
abs_z_scores = np.abs(z_scores)
filtered_entries = (abs_z_scores < 7.5).all(axis=1)
lengsBig = lengsBig[filtered_entries]
```

In [131…

```
corr = lengsBig.corr(method="pearson")["body"][1]
print("the correlation coefficient is", corr)
plt.scatter(lengsBig['body'], lengsBig['subject'])
plt.xlabel('body (chars)')
plt.ylabel('subject (chars)')
plt.show()
```

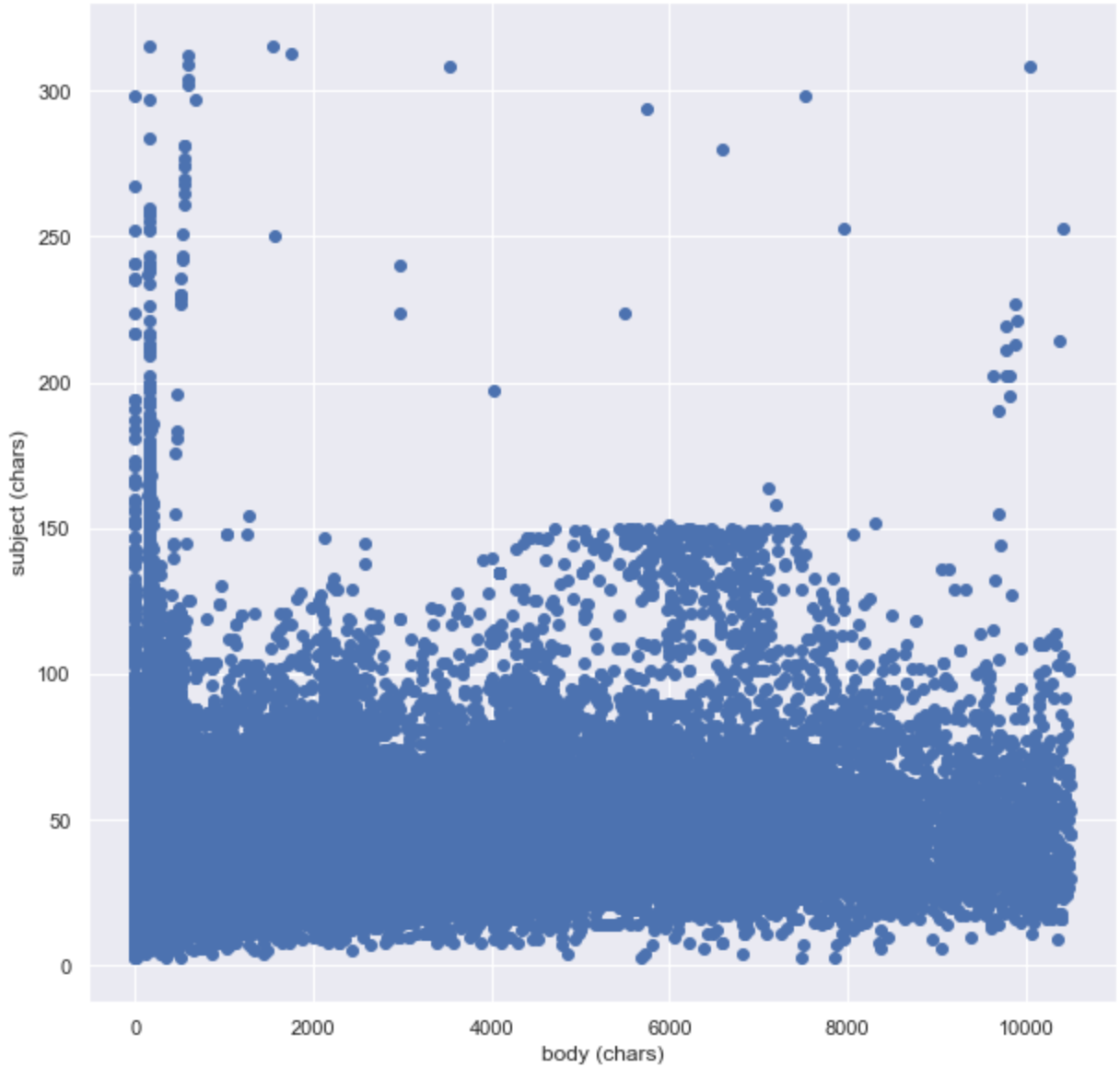
```
the correlation coefficient is 0.2698308286090695
```



In [132…

```
corr = lengsSmall.corr(method="pearson")["body"][1]
print("the correlation coefficient is", corr)
plt.scatter(lengsSmall['body'], lengsSmall['subject'])
plt.xlabel('body (chars)')
plt.ylabel('subject (chars)')
plt.show()
```

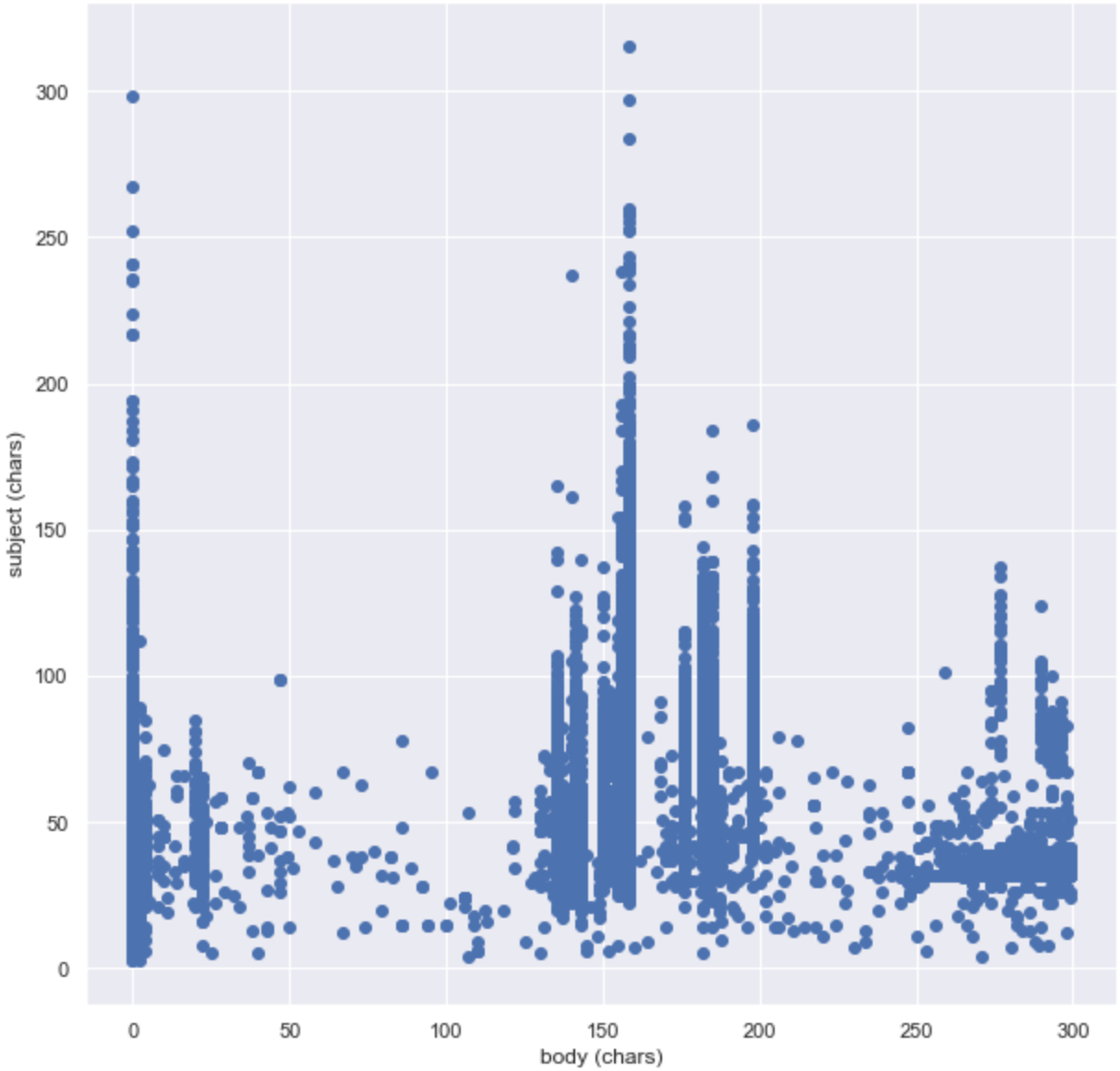
```
the correlation coefficient is -0.09134321058250898
```



In [133…

```
corr = lengsSmaller.corr(method="pearson")["body"][1]
print("the correlation coefficient is", corr)
plt.scatter(lengsSmaller['body'], lengsSmaller['subject'])
plt.xlabel('body (chars)')
plt.ylabel('subject (chars)')
plt.show()
```

```
the correlation coefficient is 0.0975900658864091
```


subject (chars)
body (chars)
0
50
100
150
200
250
300