Gmail Explorer Imputer

By nickesc / N. Escobar

Now that we have data from the last notebook, we can start to analyze it! The first thing we need to do is load in our data:

In [1]:
```
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 IPython.display import clear_output, display
#from ipywidgets import *
#from tkinter import Tk, filedialog
#from math import floor
```

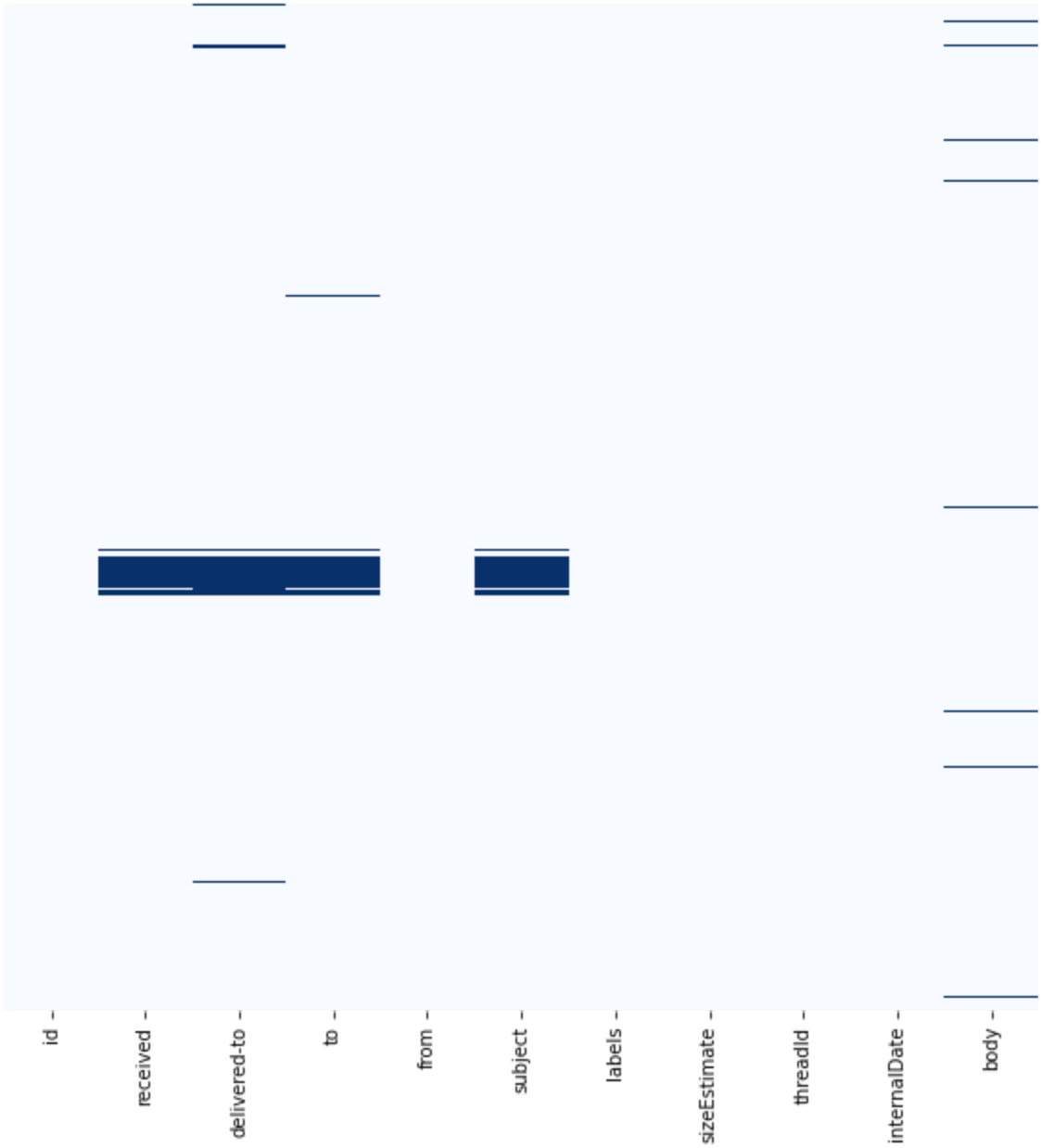
In [3]:
```
messages = pd.read_csv("../messages.csv")
messages.head()
```

Out[3]:

	id	received	delivered-to	to	
0	17f1070e03e35958	by 2002:a67:f25a:0:0:0:0:0 with SMTP id y26csp...	nickesc.media@gmail.com	nickesc.media@gmail.com	<noreply@redd
1	17f0b56c090c9862	by 2002:a67:f25a:0:0:0:0:0 with SMTP id y26csp...	nickesc.media@gmail.com	nickesc.media@gmail.com	<noreply@redd
2	17f062e5d45a24b2	by 2002:a67:f25a:0:0:0:0:0 with SMTP id y26csp...	nickesc.media@gmail.com	nickesc.media@gmail.com	<noreply@redd
3	17f0102f8d3f357a	by 2002:a67:f25a:0:0:0:0:0 with SMTP id y26csp...	nickesc.media@gmail.com	nickesc.media@gmail.com	<noreply@redd
4	17efbd7aa4817a39	by 2002:a67:f25a:0:0:0:0:0 with SMTP id y26csp...	nickesc.media@gmail.com	nickesc.media@gmail.com	<noreply@redd

In [4]:
```
fig, ax = plt.subplots(figsize=(10,10))
display(sns.heatmap(messages.isnull(),yticklabels=False, cbar=False,
            cmap="Blues",ax=ax))
```

```
<AxesSubplot:>
```



In [5]:

```
inboxes = pd.DataFrame()
chats = pd.DataFrame()
emails = pd.DataFrame()
```

Turning our data into something useful

Now, we can start to actually look at the data and come to some conclusions. First though, even though we made the data set, we still need to do some cleanup. It's still, for the most part, the same way we got it from google, just put together a little nicer. We haven't really had to fill in or change the data yet.

The `internalDate` is useful for tracking trends over time, but doesn't really tell us when it actually happened, so we need to convert that to `dateTime`. We also want to adjust our `from` column, which tells us the name and the address, and we really only want the address, because not all senders have a name. We also need to take care of all the `NaN`s in the data, which mostly come from weird or missed headers, but are mostly fixable too.

In [6]:

```
def convertTime(epochtime):
    thetime = datetime.datetime.fromtimestamp(epochtime/1000)
    return(thetime)
```

```
def convertYear(epochtime):
    thetime = datetime.datetime.fromtimestamp(epochtime/1000)
    return(thetime.year)
def convertMonth(epochtime):
    thetime = datetime.datetime.fromtimestamp(epochtime/1000)
    return(thetime.month)
def convertDay(epochtime):
    thetime = datetime.datetime.fromtimestamp(epochtime/1000)
    return(thetime.day)

nans=[]
def convertAddress(string):
    try:
        address = string.split('<')[-1].split('>')[0]
    except:
        nans.append(string)
        address = str(string)
    return address
def convertLabels(labels):
    string=""
    x=0
    for label in labels:
        if x==0:
            string=str(label)
            x+=1
        else:
            string=string+","+str(label)
    return labels

    #return labels.replace("'", "").strip(']['').split(', ')


inboxes['id'] = messages["id"]
inboxes['threadId'] = messages["threadId"]
inboxes["from"] = messages["from"].apply(convertAddress)
inboxes['delivered'] = messages["delivered-to"]
inboxes['to'] = messages["to"].apply(convertAddress)
inboxes['internalDate'] = messages["internalDate"]
inboxes["dateTime"] = messages["internalDate"].apply(convertTime)
inboxes["year"] = messages["internalDate"].apply(convertYear)
inboxes["month"] = messages["internalDate"].apply(convertMonth)
inboxes["day"] = messages["internalDate"].apply(convertDay)
inboxes["labels"] = messages["labels"].apply(convertLabels)
inboxes['sizeEstimate'] = messages["sizeEstimate"]
inboxes['subject'] = messages["subject"]
inboxes['body'] = messages['body']

inboxes["labels"].describe()
```

Out[6]:

```
count                                  92194
unique                                    81
top       ['CATEGORY_PROMOTIONS', 'INBOX']
freq                                   44386
Name: labels, dtype: object
```

Received vs. Chats vs. Drafts vs. Sent

Before we handle `NaN`s, we're going to split off the Google Hangouts chats. These, though not really emails, still appear as messages in your inbox. The problem is that they, in addition to drafts and sent emails, throw off a lot of the other metrics, especially because there are so many chats, including the onces I sent. The chats lack two of the addresses addresses or a subject for the most part, making them confusing blanks spots in the table (the big hole in the center on the heatmap at the top). Some of them you could figure out, but the ones I sent only have my name attached, not who I sent them to, so I decided

to skip them altogether, since they're already peripheral. Drafts and sent messages, however, we can still figure out the missing information in the same way we do received messages, so we'll leave those in for now.

In [7]:
```
chats = inboxes[(inboxes['labels'].str.contains('CHAT')) == True].copy(deep=True)
emails = inboxes[(inboxes['labels'].str.contains('CHAT')) == False].copy(deep=True)
display(chats,emails)
```

	id	threadId	from	delivered	
118	17ce8fff3791bda0	17ce8fff3791bda0	nickesc.media@gmail.com	NaN	
40479	15ecbd48cdcd6fd9	15ecbd48cdcd6fd9	+17134928662@gmail.com	josephgoodman85@gmail.com	jose
48257	1572f5d57668f9af	1572f5d21980ff4a	josephgoodman85@gmail.com	NaN	
48258	1572f5d3e7d52fac	1572f5d21980ff4a	josephgoodman85@gmail.com	NaN	
48259	1572f5d32bde0279	1572f5d21980ff4a	josephgoodman85@gmail.com	NaN	
...	...	...	...	...	
54518	13d2c4b7b47ae3d6	13d2c4b7b47ae3d6	stavem@gmail.com	josephgoodman85@gmail.com	jose
54520	13d2849215426742	13d2849215426742	stavem@gmail.com	josephgoodman85@gmail.com	jose
54522	13cde6bb4b7af503	13cde6bb4b7af503	mosheahron@gmail.com	josephgoodman85@gmail.com	jose
54530	13c87691aedc2062	13c8744e8652bdc3	stavem@gmail.com	josephgoodman85@gmail.com	jose
54558	13b01d23745678b5	13b01d23745678b5	stavem@gmail.com	josephgoodman85@gmail.com	jose

3851 rows × 14 columns

	id	threadId	from	delivered	
0	17f1070e03e35958	17f1070e03e35958	noreply@redditmail.com	nickesc.media@gmail.com	nickesc.me
1	17f0b56c090c9862	17f0b56c090c9862	noreply@redditmail.com	nickesc.media@gmail.com	nickesc.me
2	17f062e5d45a24b2	17f062e5d45a24b2	noreply@redditmail.com	nickesc.media@gmail.com	nickesc.me
3	17f0102f8d3f357a	17f0102f8d3f357a	noreply@redditmail.com	nickesc.media@gmail.com	nickesc.me

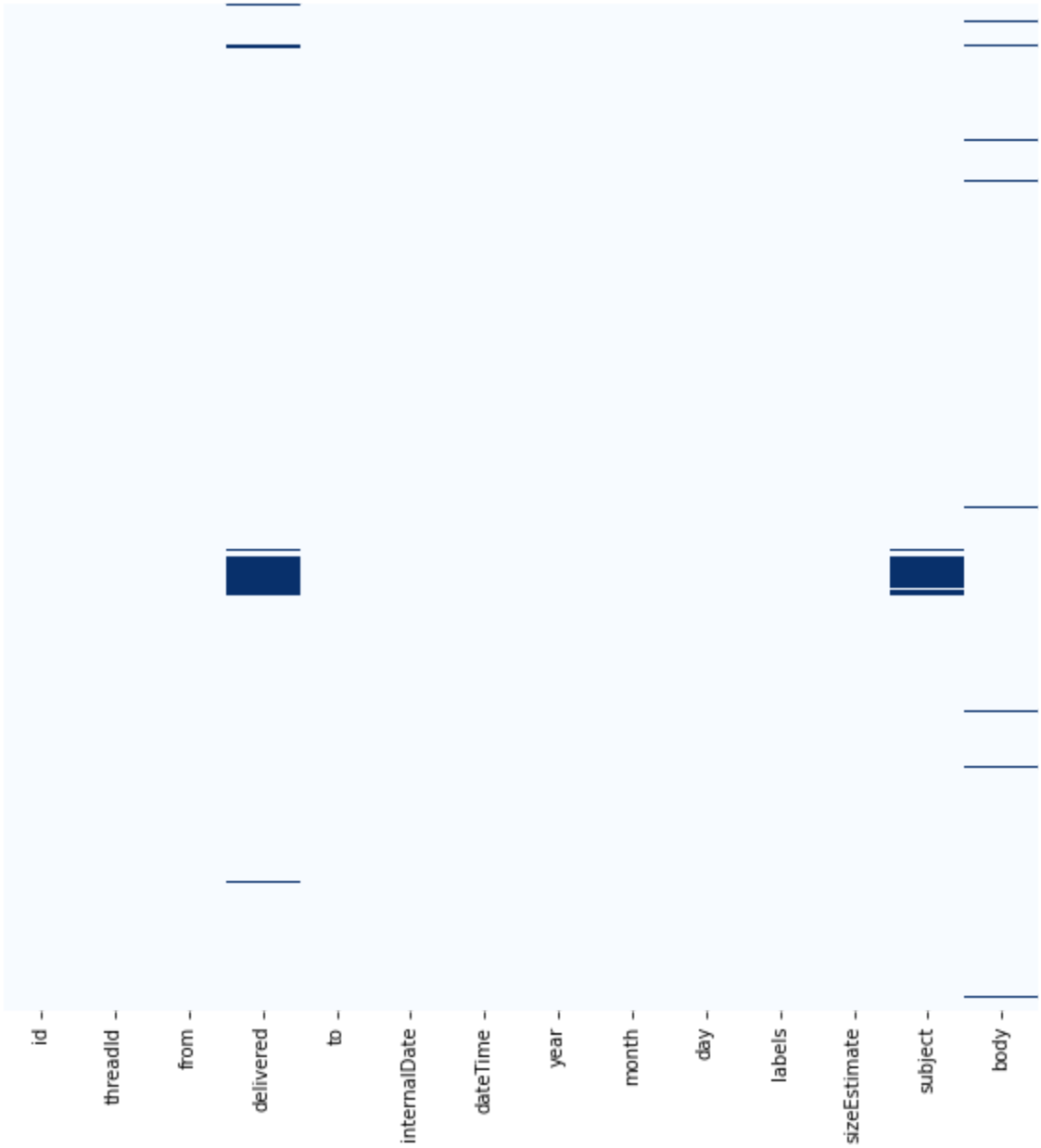
	id	threadId	from	delivered	
4	17efbd7aa4817a39	17efbd7aa4817a39	noreply@redditmail.com	nickesc.media@gmail.com	nickesc.me
...	...	...	...	...	
92189	15a1e546ba6f3c75	15a1e546ba6f3c75	noreply@creativemarket.com	nickesc.gd@gmail.com	nickesc
92190	15a1cdf13ff2d02e	15a1cdf13ff2d02e	newsletter@mightydeals.com	nickesc.gd@gmail.com	nickesc
92191	15a1bdd95e066d41	15a1bdd95e066d41	info@graphicpear.com	nickesc.gd@gmail.com	nickesc
92192	15a1bd9fae046422	15a1bd9fae046422	info@webdesignerdepot.com	nickesc.gd@gmail.com	nickesc
92193	15a1bbf6088cfc5b	15a1bbf6088cfc5b	noreply@selz.com	nickesc.gd@gmail.com	nickesc

88343 rows × 14 columns

In [8]:

```
fig, ax = plt.subplots(figsize=(10,10))
display(sns.heatmap(inboxes.isnull(),yticklabels=False, cbar=False,
           cmap="Blues",ax=ax))
```

```
<AxesSubplot:>
```



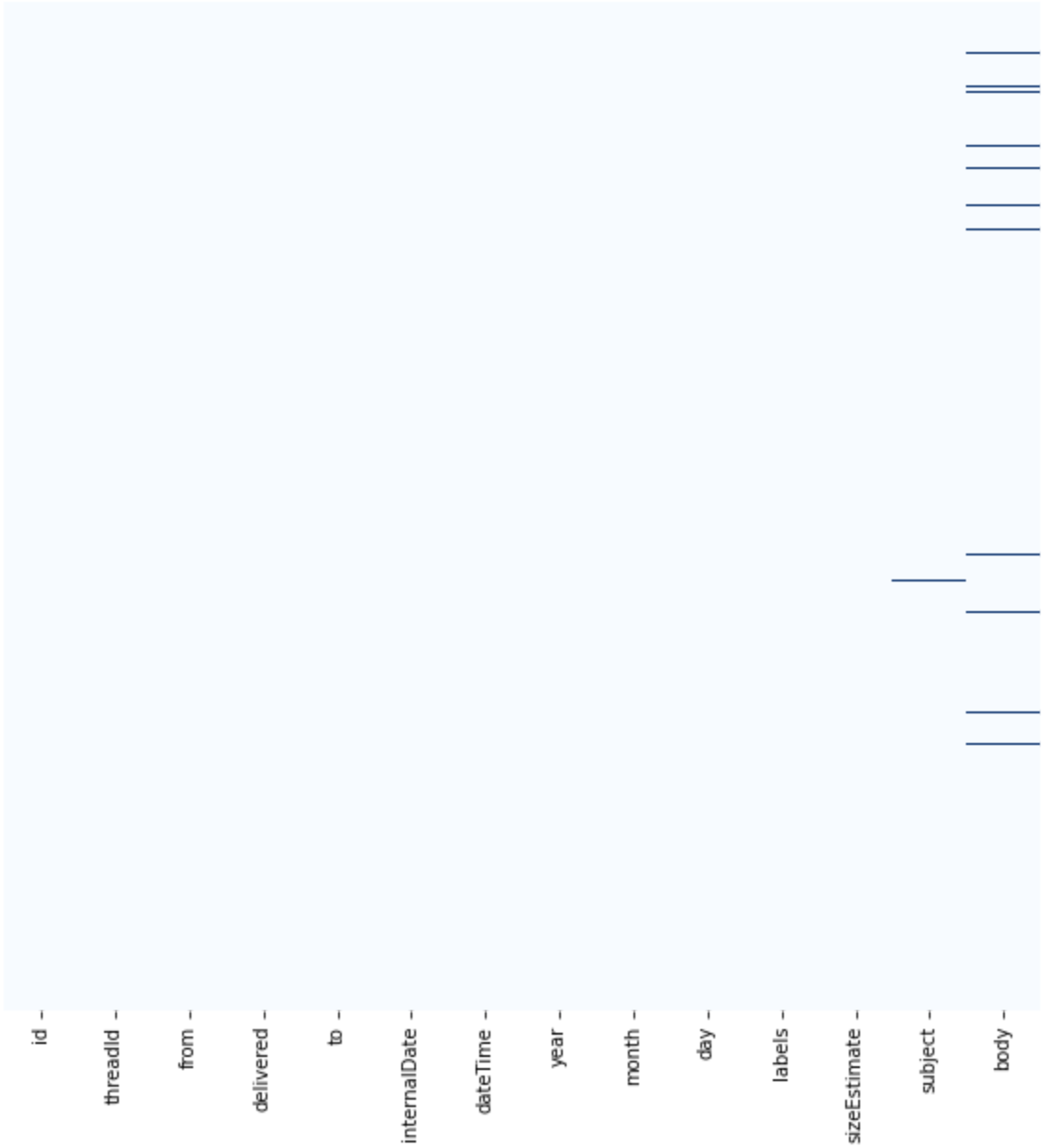
Missing Values

Then there are a lot of seemingly random missing values. Mostly, they're one of the three addresses -- `from`, `delivered` and `to`. The `from` address tells you the source, the `to` address is who it was addressed to, and the `delivered` address is the email it was delivered to. The strangest, by far, is that for some reason between August 2019 and April 2020, the World Wild Life Fund was emailing me from an account without any address attached to it. The email seemed like it was coming out of no where on the table, but I managed to find the subjects of the emails in the Gmail GUI and match them with the sender. Those are the only emails that I had track down the sender, otherwise I could fill pretty easily the `to` and `delivered` columns using each other, as they're almost always the same thing.

In [9]:

```
emails.loc[(emails['from'] == "nan") & (emails["subject"]!=""), 'from'] = 'ecomments@wwwfu
#emails.loc[(emails['labels'].str.contains('SENT')) == True, 'delivered'] = '???'
emails.loc[(emails['labels'].str.contains('DRAFT')) == True, 'delivered'] = '{{not delive
#emails.loc[(emails['labels'].str.contains('DRAFT') & emails['to']=="nan") == True, 'to']
emails['to']=np.where((emails['labels'].str.contains('DRAFT') & emails['to']==np.NaN) == T
emails["delivered"] = np.where((emails['labels'].str.contains('SENT')) == True, emails["to
fig, ax = plt.subplots(figsize=(10,10))
display(sns.heatmap(emails.isnull(),yticklabels=False, cbar=False,
           cmap="Blues",ax=ax))
```

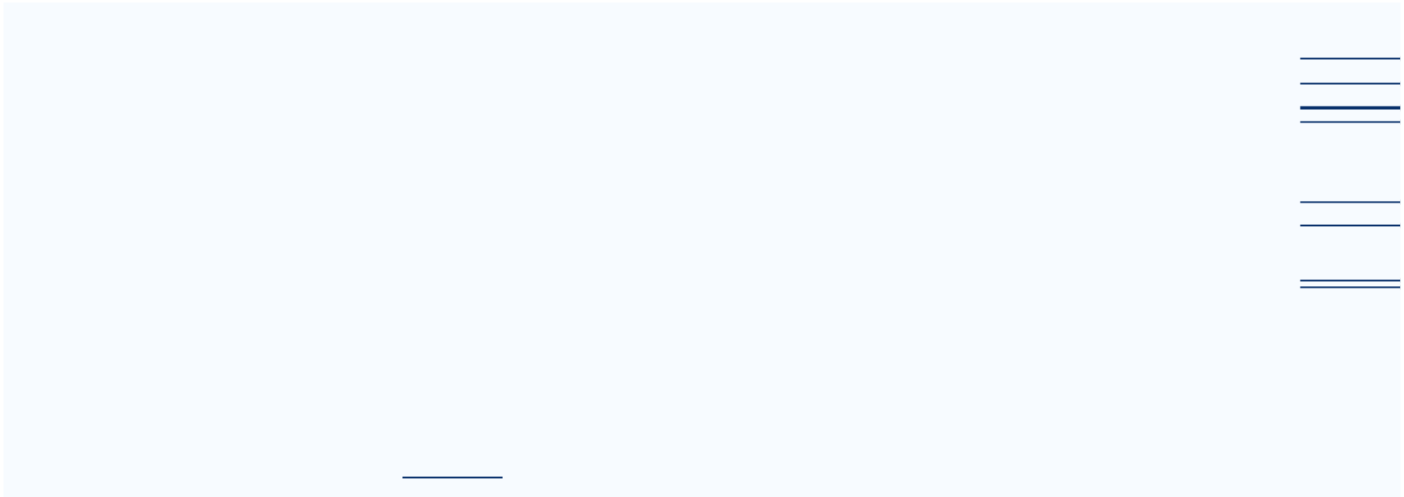
<AxesSubplot:>

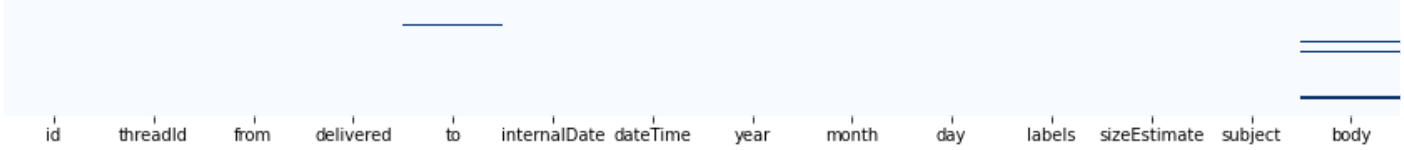


In [10]:

```
emails.replace("nan",np.NaN, inplace=True)
emails.replace("",np.NaN, inplace=True)
emails.sort_values("delivered")
fig, ax = plt.subplots(figsize=(15,30))
display(sns.heatmap(emails.isnull(),yticklabels=False, cbar=False,
            cmap="Blues",ax=ax))
```

<AxesSubplot:>





In [11]:
```
emails.sort_values("to")
```

Out[11]:

	id	threadId	from	delivered	
50537	14c709060c613284	14c708fdfc50635f	josephgoodman85@gmail.com	2018class.nines@blogger.com	2
3845	179edbf9afc73f27	179edbf9afc73f27	nescobar@oxy.edu	6577x25j@hpeprint.com	
45327	15a80e5210a8bfdc	15a807a3cb9f51dd	Elisabeth.Escobar@marriott.com	josephgoodman85@gmail.com	
51302	1481772b29dbb278	1481772b29dbb278	josephgoodman85@gmail.com	ALuther@cesjds.org	
50536	14c710087f14c5fc	14c70ff8084c97e0	josephgoodman85@gmail.com	Aaron.Liss@cesjds.org	
...	...	...	...	...	
91375	1606e805c9187519	1606e805c9187519	ecomments@wwwfus.org	nickesc.gd@gmail.com	
91388	1606429b0499730c	1606429b0499730c	ecomments@wwwfus.org	nickesc.gd@gmail.com	
91404	16056134a64bde29	16056134a64bde29	ecomments@wwwfus.org	nickesc.gd@gmail.com	
91416	1604bc1b07faaa57	1604bc1b07faaa57	ecomments@wwwfus.org	nickesc.gd@gmail.com	
91420	16046a53f8fc184b	16046a53f8fc184b	ecomments@wwwfus.org	nickesc.gd@gmail.com	

88343 rows × 14 columns

In [12]:
```
#emails["delivered"] = np.where((emails['delivered'] == np.NaN), emails["delivered"], emai
emails.delivered.fillna(emails.to, inplace=True)
emails.to.fillna(emails.delivered, inplace=True)
emails.sort_values("delivered")
```

Out[12]:

	id	threadId	from	delivered	
50537	14c709060c613284	14c708fdfc50635f	josephgoodman85@gmail.com	2018class.nines@blogger.com	2(

	id	threadId	from	delivered	
3845	179edbf9afc73f27	179edbf9afc73f27	nescobar@oxy.edu	6577x25j@hpeprint.com	
51302	1481772b29dbb278	1481772b29dbb278	josephgoodman85@gmail.com	ALuther@cesjds.org	
50536	14c710087f14c5fc	14c70ff8084c97e0	josephgoodman85@gmail.com	Aaron.Liss@cesjds.org	
50685	14c2df909bbe478d	14c2df8d62c9e0f8	josephgoodman85@gmail.com	Aaron.Liss@cesjds.org	
...	...	...	...	...	
1973	17ccdeca050b14f3	17cc93f8c563bc27	nescobar@oxy.edu	{{not delivered}}	
84842	16b6db70a82d4462	16b6db4967966902	nickesc.gd@gmail.com	{{not delivered}}	jenl
3279	17b9f29954d1ee97	17b9e9993030b370	nescobar@oxy.edu	{{not delivered}}	
4735	178843e10007605f	17882113ef762db0	nescobar@oxy.edu	{{not delivered}}	
2890	17bf8cd33b116ce7	17bf81c12e9338b5	nescobar@oxy.edu	{{not delivered}}	sar

88343 rows × 14 columns

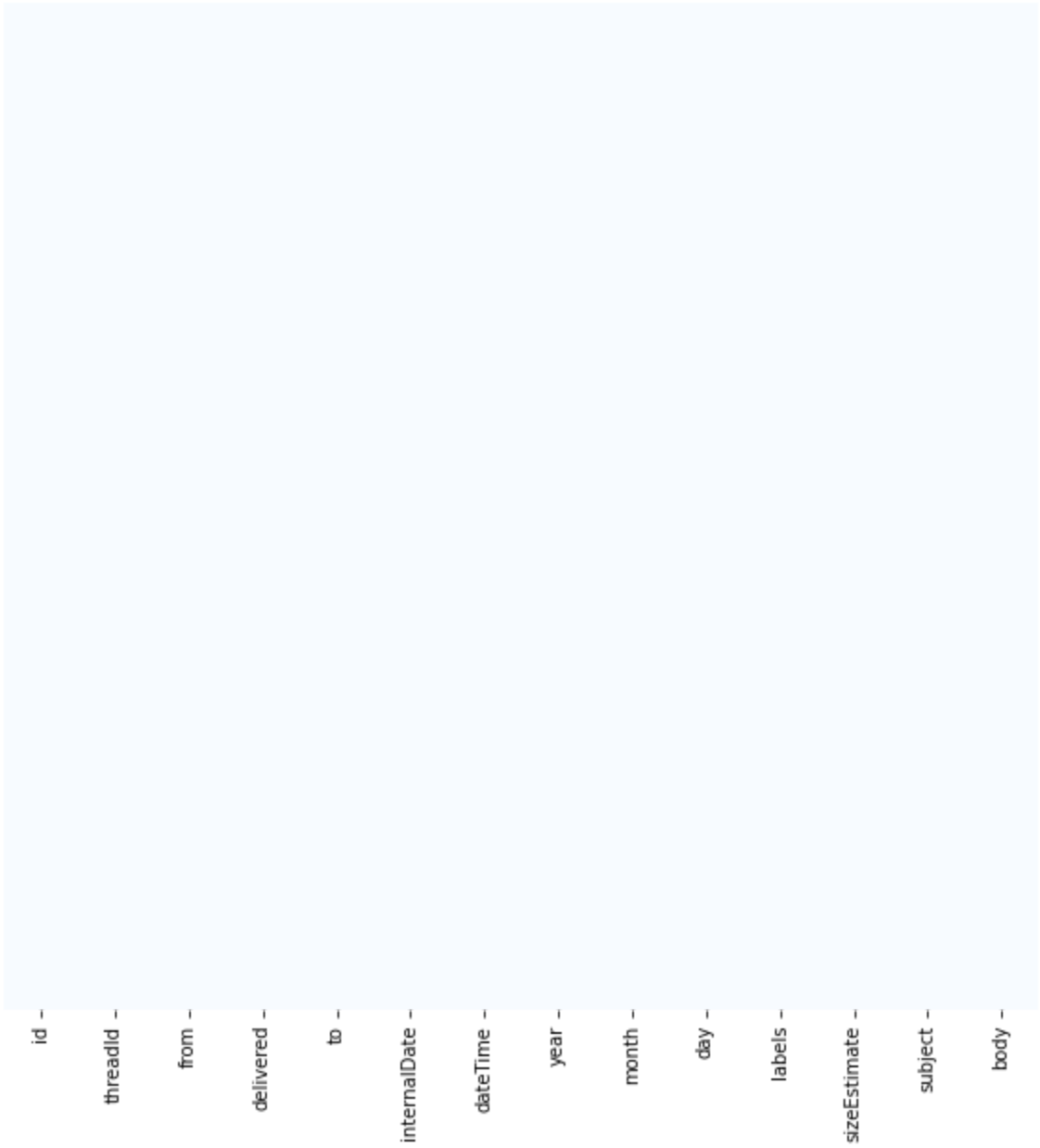
Next we fill the empty subject and body cells with a value to indicate they're empty, and lower all the addresses to be sure they're consistent when we're comparing them.

In [13]:
```
emptyBody=base64.urlsafe_b64encode("{{empty message body}}".encode('utf-8'))
emptySubject="{{no subject}}"

emails.replace("undisclosed-recipients:;","{{undisclosed-recipients}}")

def lowerIt(x):
    return x.lower()

emails["from"].apply(lowerIt)
emails["delivered"].apply(lowerIt)
emails["to"].apply(lowerIt)

emails["body"].fillna(emptyBody, inplace = True)
emails["subject"].fillna(emptySubject, inplace = True)
```

In [14]:
```
fig, ax = plt.subplots(figsize=(10,10))
display(sns.heatmap(emails.sort_values("delivered").isnull(),yticklabels=False, cbar=False
```

```
            cmap="Blues",ax=ax))
nans=emails[emails['delivered'].isnull()].index.tolist()
```

```
<AxesSubplot:>
```



Exporting the Data

In [15]:

```
sent = emails[(emails['labels'].str.contains('SENT')) == True].copy(deep=True)
drafts = emails[(emails['labels'].str.contains('DRAFT')) == True].copy(deep=True)
#chats = emails[(emails['labels'].str.contains('CHAT')) == True].copy(deep=True)

recieved = emails[(emails['labels'].str.contains('SENT') == False) &  (emails['labels'].st


display(recieved.head(),sent.head(), drafts.head(),chats.head())
```

	id	threadId	from	delivered	
0	17f1070e03e35958	17f1070e03e35958	noreply@redditmail.com	nickesc.media@gmail.com	nickesc.media@gmai

	id	threadId	from	delivered	
1	17f0b56c090c9862	17f0b56c090c9862	noreply@redditmail.com	nickesc.media@gmail.com	nickesc.media@gmai
2	17f062e5d45a24b2	17f062e5d45a24b2	noreply@redditmail.com	nickesc.media@gmail.com	nickesc.media@gmai
3	17f0102f8d3f357a	17f0102f8d3f357a	noreply@redditmail.com	nickesc.media@gmail.com	nickesc.media@gmai
4	17efbd7aa4817a39	17efbd7aa4817a39	noreply@redditmail.com	nickesc.media@gmail.com	nickesc.media@gmai

	id	threadId	from	delivered	
76	17dae8fa46d72a57	17dae8fa46d72a57	nickesc.media@gmail.com	steven@stevenrescobar.com	steven@steven
77	17dae79b7b93dede	17dae79b7b93dede	nickesc.media@gmail.com	steven@stevenrescobar.com	steven@steven
81	17da920352559430	17da920352559430	nickesc.media@gmail.com	steven@stevenrescobar.com	steven@steven
82	17da91bd48e8f9c9	17da91bd48e8f9c9	nickesc.media@gmail.com	steven@stevenrescobar.com	steven@steven
83	17da91a4996032a7	17da91a4996032a7	nickesc.media@gmail.com	steven@stevenrescobar.com	steven@steven

	id	threadId	from	delivered	to	internalDate	
361	17ee5e1f61224f87	17eace3b9e8e15f3	nescobar@oxy.edu	{{not delivered}}	alyford@oxy.edu	1644534298000	
371	17ee47ada2c5e9d4	17ee47ada2c5e9d4	nescobar@oxy.edu	{{not delivered}}	boscoe@oxy.edu	1644510763000	
520	17ebe546770d30e0	17ebe07a16a1b78a	nescobar@oxy.edu	{{not delivered}}	boscoe@oxy.edu	1643870709000	
990	17dec9b24f9afe09	17dec9b24f9afe09	nescobar@oxy.edu	{{not delivered}}	{{not delivered}}	1640352130000	
1155	17da0bf66d6a969b	17da0bf66d6a969b	nescobar@oxy.edu	{{not delivered}}	atasse@oxy.edu	1639079437000	

	id	threadId	from	delivered

	id	threadId	from	delivered	
118	17ce8fff3791bda0	17ce8fff3791bda0	nickesc.media@gmail.com	NaN	
40479	15ecbd48cdcd6fd9	15ecbd48cdcd6fd9	+17134928662@gmail.com	josephgoodman85@gmail.com	josep
48257	1572f5d57668f9af	1572f5d21980ff4a	josephgoodman85@gmail.com	NaN	
48258	1572f5d3e7d52fac	1572f5d21980ff4a	josephgoodman85@gmail.com	NaN	
48259	1572f5d32bde0279	1572f5d21980ff4a	josephgoodman85@gmail.com	NaN	

In [16]:

```
recieved.sort_values('internalDate').reset_index(drop=True).to_csv('imputed/recieved.csv')
sent.sort_values('internalDate').reset_index(drop=True).to_csv("imputed/sent.csv")
drafts.sort_values('internalDate').reset_index(drop=True).to_csv('imputed/drafts.csv')
chats.sort_values('internalDate').reset_index(drop=True).to_csv('imputed/chats.csv')
```

Finally, we pull all our various `DataFRames` together and output four `.csv` files, for recieved messages, for sent messages, for drafts, and for chats, each of which is a little different to look at.