Myeloma_Mapping

February 17, 2019

```
In [11]: import pandas as pd
         import numpy as np
         import urllib2
         import webbrowser
         import os
         import bs4,  re
         from bs4 import BeautifulSoup
         import requests
         import math
         import nltk
         from nltk.tokenize import word_tokenize
         import matplotlib.pyplot as plt
         %matplotlib inline
         from datetime import datetime
```

0.0.1 The cell below opens (and converts to a dataframe) a csv file containing a URL that, when opened, will automatically download a csv file tuned to the prescribed search. For example, the first search performed looked (1) myeloma, (2) USA, (3) Recruiting AND Non yet recruiting AND Active, but no longer recruiting, (4) Car-T. The URL syntax can be found at ClinicalTrials.gov site.

```
In [12]: df = pd.read_csv('D:\Python_Database\Myeloma\MM Trials\keyword_search.csv')
         df1 = pd.read_csv('D:\Python_Database\uscities.csv')
```

```
In [13]: df.head()
```

```
Out[13]:   label                                                url
         0  CarT  https://clinicaltrials.gov/ct2/results/downloa...
         1    MM  https://clinicaltrials.gov/ct2/results/downloa...
```

0.0.2 The cells below open the url stored in the dataframe, and then moves the csv file from download to the desired file location

```
In [14]: chrome_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'
         url = df['url'].tolist()
         webbrowser.get(using=chrome_path).open(url[0])
```

```
Out[14]: True
```

```
In [16]: os.rename("C:/Users/robin/Downloads/SearchResults.csv", "D:/Python_Database/Myeloma/MM

In [17]: df = pd.read_csv('D:\Python_Database\Myeloma\MM Trials\CART_Results_' + str(datetime.
         df_ = df

In [18]: df = df[['NCT Number','Title','Locations','Phases', 'Status', 'Interventions', 'Last 

In [19]: color = ['#0000ff','#0e0dff','#1716ff','#1e1dff','#2323ff','#2828ff','#2c2dff','#2f32
                  '#383eff','#3b42ff','#3d46ff','#3f49ff','#414dff','#4351ff','#4554ff','#4757
                  '#4b61ff','#4d65ff','#4e68ff','#4f6bff','#506eff','#5172ff','#5275ff','#5378
                  '#5581ff','#5684ff','#5687ff','#568bff','#578eff','#5791ff','#5794ff','#5897
                  '#58a0ff','#58a3ff','#57a6ff','#57a9ff','#57acff','#56afff','#56b2ff','#55b5
                  '#53bfff','#52c2ff','#51c5ff','#50c8ff','#4ecbff','#4dceff','#4bd1ff','#4ad4
                  '#44ddff','#41e0ff','#3ee3ff','#3be6ff','#38e9ff','#34ecff','#30f0ff','#2bf3
                  '#13fcff','#00ffff','#08fdfb','#0efbf8','#13f9f4','#17f7f1','#1af6ed','#1df4
                  '#23eedf','#25ecdc','#27ead8','#28e8d5','#29e6d1','#2ae5ce','#2be3ca','#2ce1
                  '#2fdbbd','#2fdab9','#30d8b6','#30d6b3','#30d4af','#31d2ac','#31d0a8','#31cf
                  '#31c99b','#31c798','#31c594','#31c491','#31c28e','#31c08a','#31be87','#30bc
                  '#2fb77a','#2fb577','#2eb473','#2eb270','#2db06d','#2dae6a','#2cac66','#2bab
                  '#29a559','#28a456','#27a253','#26a04f','#259f4c','#249d49','#239b45','#2299
                  '#1e9438','#1c9334','#1a9131','#198f2d','#178d2a','#158c26','#138a22','#1088
                  '#07830e','#038207','#008000','#0c8200','#158400','#1c8500','#228700','#2789
                  '#358e00','#399000','#3d9200','#419300','#449500','#489700','#4b9900','#4f9a
                  '#59a000','#5da100','#60a300','#63a500','#66a700','#6aa900','#6daa00','#70ac
                  '#79b100','#7db300','#80b500','#83b700','#86b900','#89ba00','#8cbc00','#8fbe
                  '#99c300','#9cc500','#9fc700','#a2c900','#a5ca00','#a8cc00','#abce00','#aed0
                  '#b7d500','#bad700','#bed900','#c1db00','#c4dc00','#c7de00','#cae000','#cde2
                  '#d6e700','#d9e900','#ddeb00','#e0ed00','#e3ef00','#e6f000','#e9f200','#ecf4
                  '#f6fa00','#f9fb00','#fcfd00','#ffff00','#fffc00','#fffa00','#fff700','#fff5
                  '#ffed00','#ffeb00','#ffe800','#ffe500','#ffe300','#ffe000','#ffde00','#ffdb
                  '#ffd300','#ffd100','#ffce00','#ffcb00','#ffc900','#ffc600','#ffc300','#ffc1
                  '#ffb900','#ffb600','#ffb300','#ffb100','#ffae00','#ffab00','#ffa800','#ffa6
                  '#ff9d00','#ff9a00','#ff9800','#ff9500','#ff9200','#ff8f00','#ff8c00','#ff89
                  '#ff8000','#ff7d00','#ff7a00','#ff7600','#ff7300','#ff7000','#ff6c00','#ff69
                  '#ff5e00','#ff5a00','#ff5700','#ff5200','#ff4e00','#ff4a00','#ff4500','#ff40
                  '#ff2f00','#ff2800','#ff1f00','#ff1400','#ff0000']

In [20]: color = color[::len(color)/len(df_)]
         # len(color)

In [21]: mask = df['Status'].str.contains('Recruiting', case=True)

         colors = color[:len(df['NCT Number'])]
         colors = pd.DataFrame(colors)
         colors.columns = ['Color']

         df = pd.concat([df, colors], axis=1)

In [22]: pd.set_option('max_colwidth', 800)
         temp = df.reindex(index=df.index[::-1])
         temp[['NCT Number','Title', 'Status']]
```

```
Out[22]:      NCT Number  \
         21  NCT02529813
         20  NCT02658929
         19  NCT03430011
         18  NCT03361748
         17  NCT03274219
         16  NCT03601078
         15  NCT03318861
         14  NCT03548207
         13  NCT03651128
         12  NCT02215967
         11  NCT03288493
         10  NCT03602612
         9   NCT03448978
         8   NCT03672318
         7   NCT03338972
         6   NCT03710421
         5   NCT03070327
         4   NCT03502577
         3   NCT02546167
         2   NCT02794246
         1   NCT03464916
         0   NCT03549442


         21
         20
         19                                                                           Study Eva
         18
         17
         16                                            An Efficacy and Safety Study of bb2121 i
         15                                                                                A St
         14  A Study of JNJ-68284528, a Chimeric Antigen Receptor T Cell (CAR-T) Therapy Direct
         13                                                       Efficacy and Safety Study of
         12
         11
         10
         9
         8
         7                                                                          Immunothera
         6                                                                                    CS
         5
         4                                                               BCMA-Specific CAR T-Ce
         3
         2
         1                                                                            Study to
         0
```

```
                    Status
21              Recruiting
20  Active, not recruiting
19              Recruiting
18  Active, not recruiting
17              Recruiting
16              Recruiting
15  Active, not recruiting
14              Recruiting
13      Not yet recruiting
12  Active, not recruiting
11              Recruiting
10              Recruiting
9               Recruiting
8               Recruiting
7               Recruiting
6       Not yet recruiting
5               Recruiting
4               Recruiting
3   Active, not recruiting
2   Active, not recruiting
1               Recruiting
0               Recruiting
```

```
In [23]: test_data = df['Locations'].tolist()
```

```
In [24]: city, state, city_state = [], [], []

         for i in range(len(df1['city'])):
             city.append(df1['city'][i])
             state.append(df1['state_name'][i])
             city_state.append(city[i] + ', ' + state[i])
```

```
In [25]: results = []

         for i in range(len(test_data)):
             for j in range(len(city_state)):
                 if str(test_data[i]).find(city_state[j]) >=0:
                     results.append(city_state[j])
```

```
In [26]: temp = []
         temp1 = []

         for i in range(len(test_data)):
             for j in range(len(city_state)):
                 if str(test_data[i]).find(city_state[j]) >= 0:

                     temp1.append(city_state[j])
             temp.append(temp1)
             temp1=[]
```

```
In [27]: location_of_study=[]
         for i in range(len(temp)):
             names = set(temp[i])
             names = list(names)
             location_of_study.append(names)
```

```
In [28]: num=[]
         for i in range(len(location_of_study)):
             num.append(len(location_of_study[i]))
```

```
In [29]: ind, pos = [], []

         for i in range(len(num)):
             if num[i] > 1:
                 ind.append(num[i])
                 pos.append(i)
```

```
In [30]: df = df.reset_index(drop=True)
         index = df.index.tolist()

         temp = [x*1000 for x in index]

         df = df.set_index([temp])
```

```
In [31]: for i in range(len(pos)):
             k=0
             while k < ind[i]:
                 df.loc[(pos[i]*1000)+k]  = df.loc[pos[i]*1000]
                 k=k+1
```

```
In [32]: df = df.sort_index()
```

```
In [33]: new_column = []
         for i in range(len(location_of_study)):
             for j in range(len(location_of_study[i])):
                 new_column.append(location_of_study[i][j])

         City_State = pd.DataFrame(new_column, columns=['City, State'])
```

```
In [34]: df_new = pd.concat([df.reset_index(drop=True), City_State], axis=1)
         df_new.head()
```

```
Out[34]:     NCT Number  \
         0  NCT03549442
         1  NCT03464916
```

```
2  NCT03464916
3  NCT02794246
4  NCT02546167


0                                          Up-front CART-BCMA With or Without huCART19
1  Study to Evaluate the Safety and Efficacy of Anti-CD38 CAR-T in Relapsed or Refract
2  Study to Evaluate the Safety and Efficacy of Anti-CD38 CAR-T in Relapsed or Refract
3                                                                           CART-19 I
4                                                                             CART-BC


0
1  University of Pennsylvania, Abramson Cancer Center, Philadelphia, Pennsylvania, Uni
2  University of Pennsylvania, Abramson Cancer Center, Philadelphia, Pennsylvania, Uni
3                                                                  Abramson Cancer Ce
4                                                                  Abramson Cancer Ce

    Phases                  Status  \
0  Phase 1              Recruiting
1  Phase 1              Recruiting
2  Phase 1              Recruiting
3  Phase 2  Active, not recruiting
4  Phase 1  Active, not recruiting


0  Combination Product: BCMA CART + huCART19|Combination Product: CART BCMA or CART BC
1
2
3
4

  Last Update Posted                Sponsor/Collaborators    Color  \
0      June 20, 2018  University of Pennsylvania|Novartis  #0000ff
1      June 27, 2018          Sorrento Therapeutics, Inc.  #3d46ff
2      June 27, 2018          Sorrento Therapeutics, Inc.  #3d46ff
3  November 15, 2018           University of Pennsylvania  #506eff
4   October 15, 2018           University of Pennsylvania  #5794ff

                 City, State
0  Philadelphia, Pennsylvania
1    Providence, Rhode Island
2  Philadelphia, Pennsylvania
3  Philadelphia, Pennsylvania
4  Philadelphia, Pennsylvania
```

```
In [35]: cit = df1['city'].tolist()
         state = df1['state_name'].tolist()
```

```
        loc_db = []

        for i in range(len(cit)):
            loc_db.append(cit[i] + ', ' + state[i])
```

```
In [36]: lat_e, lng_e = [],[]

         citystate = df_new['City, State'].tolist()

         for i in range(len(citystate)):
             for j in range(len(loc_db)):
                 if citystate[i] == loc_db[j]:
                     lng_e.append(df1['lng'][j])
                     lat_e.append(df1['lat'][j])
```

```
In [37]: lyo = df_new['Last Update Posted'].tolist()

         lyo = [2000 + int(x[-2:]) for x in lyo]
```

```
In [38]: LYO = pd.DataFrame(lyo, columns=['Year'])
         Lat = pd.DataFrame(lat_e, columns=['Lat'])
         Lng = pd.DataFrame(lng_e, columns=['Lng'])
```

```
In [39]: Lat = Lat.reset_index(drop=True)
         Lng = Lng.reset_index(drop=True)
         LYO = LYO.reset_index(drop=True)
```

```
In [40]: df_new = pd.concat([df_new, Lat, Lng, LYO], axis=1)

         df_new.head()
```

```
Out[40]:     NCT Number  \
         0  NCT03549442
         1  NCT03464916
         2  NCT03464916
         3  NCT02794246
         4  NCT02546167


         0                                         Up-front CART-BCMA With or Without huCART19
         1  Study to Evaluate the Safety and Efficacy of Anti-CD38 CAR-T in Relapsed or Refrac
         2  Study to Evaluate the Safety and Efficacy of Anti-CD38 CAR-T in Relapsed or Refrac
         3                                                                            CART-19 I
         4                                                                              CART-BC


         0
         1  University of Pennsylvania, Abramson Cancer Center, Philadelphia, Pennsylvania, Uni
         2  University of Pennsylvania, Abramson Cancer Center, Philadelphia, Pennsylvania, Uni
         3                                                                   Abramson Cancer Ce
         4                                                                   Abramson Cancer Ce

             Phases                 Status  \
         0  Phase 1             Recruiting
         1  Phase 1             Recruiting
         2  Phase 1             Recruiting
         3  Phase 2  Active, not recruiting
         4  Phase 1  Active, not recruiting


         0  Combination Product: BCMA CART + huCART19|Combination Product: CART BCMA or CART BC
         1
         2
         3
         4

           Last Update Posted                Sponsor/Collaborators    Color  \
         0      June 20, 2018  University of Pennsylvania|Novartis  #0000ff
         1      June 27, 2018          Sorrento Therapeutics, Inc.  #3d46ff
         2      June 27, 2018          Sorrento Therapeutics, Inc.  #3d46ff
         3  November 15, 2018           University of Pennsylvania  #506eff
         4   October 15, 2018           University of Pennsylvania  #5794ff

                           City, State      Lat      Lng  Year
         0  Philadelphia, Pennsylvania  40.0076 -75.1340  2018
         1    Providence, Rhode Island  41.8229 -71.4186  2018
         2  Philadelphia, Pennsylvania  40.0076 -75.1340  2018
         3  Philadelphia, Pennsylvania  40.0076 -75.1340  2018
         4  Philadelphia, Pennsylvania  40.0076 -75.1340  2018
```

```
In [41]: df_new['Status'].unique().tolist()
```

```
Out[41]: ['Recruiting', 'Active, not recruiting', 'Not yet recruiting']
```

```
In [42]: df_new.head(1)
```

```
Out[42]:     NCT Number  \
         0  NCT03549442

                                                                                Title  \
         0  Up-front CART-BCMA With or Without huCART19 in High-risk Multiple Myeloma

                                                                   Locations   Phases  \
         0  Univ. of Pennsylvania, Philadelphia, Pennsylvania, United States  Phase 1

                Status  \
         0  Recruiting


         0  Combination Product: BCMA CART + huCART19|Combination Product: CART BCMA or CART BC

           Last Update Posted                Sponsor/Collaborators    Color  \
         0      June 20, 2018  University of Pennsylvania|Novartis  #0000ff

                          City, State      Lat     Lng  Year
         0  Philadelphia, Pennsylvania  40.0076 -75.134  2018
```

```
In [43]: NCT_no = df_new['NCT Number'].unique().tolist()
         NCT_no[1]
```

```
Out[43]: 'NCT03464916'
```

```
In [44]: df_new_ = df_new[['NCT Number','Title', 'Status', 'City, State', 'Lat', 'Lng', 'Color

         mask2 = []
         for i in range(len(NCT_no)):
             mask2.append(df_new_.mask(df_new_['NCT Number'] != NCT_no[i]).dropna(axis=0, inpla

         mask2[1]
```

```
Out[44]:     NCT Number  \
         0  NCT03464916
         1  NCT03464916


         0  Study to Evaluate the Safety and Efficacy of Anti-CD38 CAR-T in Relapsed or Refrac
         1  Study to Evaluate the Safety and Efficacy of Anti-CD38 CAR-T in Relapsed or Refrac

                Status                 City, State      Lat      Lng    Color
         0  Recruiting    Providence, Rhode Island  41.8229 -71.4186  #3d46ff
         1  Recruiting  Philadelphia, Pennsylvania  40.0076 -75.1340  #3d46ff
```

```
In [45]: city_state = df_new['City, State'].unique().tolist()
```

```
In [46]: mask3 = []
         for i in range(len(city_state)):
             mask3.append(df_new_.mask(df_new_['City, State'] != city_state[i]).dropna(axis=0,

         len(mask3[0])
```

```
Out[46]: 5
```

```
In [47]: city_site = []

         for i in range(len(mask3)):
```

```
    city_site.append(mask3[i].iloc[0])

city_site[0]
```

```
Out[47]: NCT Number                                                                   NCT035494
         Title          Up-front CART-BCMA With or Without huCART19 in High-risk Multiple Myelo
         Status                                                                        Recruiti
         City, State                                                  Philadelphia, Pennsylvan
         Lat                                                                              40.00
         Lng                                                                              -75.1
         Color                                                                            #0000
         Name: 0, dtype: object
```

```
In [48]: for i in range(len(mask3)):
             for j in range(len(mask3[i])):
                 if len(mask3[i]) > 1:
                     mask3[i]['Lat'][j]  = mask3[i]['Lat'][j] + 1* math.cos(j*math.pi/((7+1)/2
                     mask3[i]['Lng'][j]  = mask3[i]['Lng'][j] + 1* math.sin(j*math.pi/((7+1)/2
                 else:
                     mask3[i]['Lat'][j] = mask3[i]['Lat'][j]
                     mask3[i]['Lng'][j] = mask3[i]['Lng'][j]
```

```
C:\Users\robin\Anaconda2\lib\site-packages\ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  after removing the cwd from sys.path.
C:\Users\robin\Anaconda2\lib\site-packages\ipykernel_launcher.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  """
C:\Users\robin\Anaconda2\lib\site-packages\ipykernel_launcher.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  import sys
C:\Users\robin\Anaconda2\lib\site-packages\ipykernel_launcher.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
```

```
In [49]: mask3[0]
```

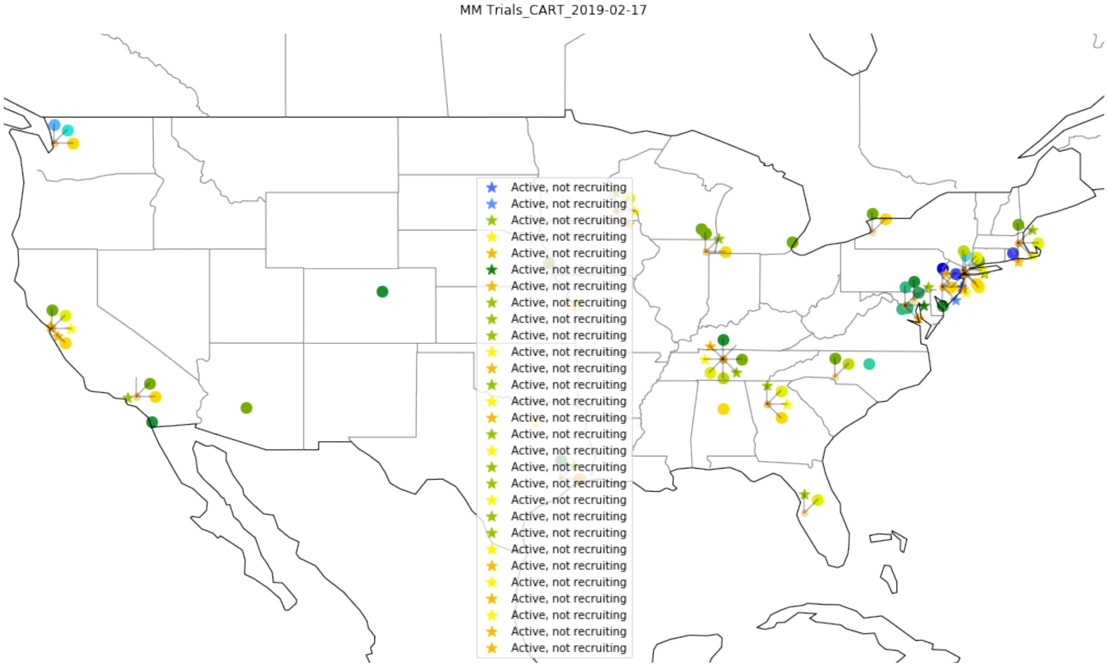
```
Out[49]:     NCT Number  \
         0   NCT03549442
         1   NCT03464916
         2  NCT02794246
         3  NCT02546167
         4  NCT03288493


         0                                         Up-front CART-BCMA With or Without huCART19
         1  Study to Evaluate the Safety and Efficacy of Anti-CD38 CAR-T in Relapsed or Refrac
         2                                                                             CART-19 I
         3                                                                               CART-BC
         4                               P-BCMA-101 Tscm CAR-T Cells in the Treatment of Patien

                              Status                City, State        Lat        Lng  \
         0              Recruiting  Philadelphia, Pennsylvania  41.007600 -75.134000
         1              Recruiting  Philadelphia, Pennsylvania  40.714707 -74.426893
         2  Active, not recruiting  Philadelphia, Pennsylvania  40.007600 -74.134000
         3  Active, not recruiting  Philadelphia, Pennsylvania  39.300493 -74.426893
         4              Recruiting  Philadelphia, Pennsylvania  39.007600 -75.134000

              Color
         0  #0000ff
         1  #3d46ff
         2  #506eff
         3  #5794ff
         4  #1a9131
```

```
In [50]: for i in range(len(mask3[0])):
             if mask3[0]['Status'][i] == 'Recruiting':
                 print 'yes'
             elif mask3[0]['Status'][i] == 'Active, not recruiting':
                 print 'no'
```

```
yes
yes
no
no
yes
```

```
In [79]: import cartopy.crs as ccrs
         import cartopy.feature as cfeature

         plt.figure(figsize=(18,16))

         states_provinces = cfeature.NaturalEarthFeature(
             category='cultural',
             name='admin_1_states_provinces_lines',
             scale='50m',
             facecolor='none')
```

```
ax = plt.axes(projection=ccrs.PlateCarree())
ax.set_extent([-125, -66.5, 20, 50], ccrs.Geodetic())
ax.coastlines()
ax.add_feature(cfeature.BORDERS)
ax.add_feature(states_provinces, edgecolor='gray')
ax.background_patch.set_visible(False)
ax.outline_patch.set_visible(False)


for i in range(len(mask3)):
    for j in range(len(mask3[i])):
        plt.plot([city_site[i]['Lng'],mask3[i].iloc[j]['Lng']], [city_site[i]['Lat'],
        plt.scatter(city_site[i]['Lng'], city_site[i]['Lat'], c='orange', alpha=0.2)
        if mask3[i]['Status'][j] == 'Recruiting':
            plt.scatter(mask3[i].iloc[j]['Lng'], mask3[i].iloc[j]['Lat'], c=mask3[i].
            plt.legend()
        elif mask3[i]['Status'][j] == 'Active, not recruiting':
            plt.scatter(mask3[i].iloc[j]['Lng'], mask3[i].iloc[j]['Lat'], c=mask3[i].
            plt.legend()
        elif mask3[i]['Status'][j] == 'Active, not recruiting':
            plt.scatter(mask3[i].iloc[j]['Lng'], mask3[i].iloc[j]['Lat'], c=mask3[i].
            plt.legend()

plt.savefig('MM Trials_CART_' + str(datetime.now())[:10] + '.png', format='png', dpi=
plt.title('MM Trials_CART_' + str(datetime.now())[:10] + '\n')
plt.show()
```



```
In [119]: lst = []
          for i in range(len(mask3)):
               for j in range(len(mask3[i])):
                  if mask3[i].iloc[j]['Status'] == 'Recruiting':
                      lst.append(mask3[i].iloc[j]['NCT Number'])

          list(set(lst))
```

```
Out[119]: ['NCT03288493',
           'NCT03430011',
           'NCT03464916',
           'NCT03602612',
           'NCT03548207',
           'NCT03274219',
           'NCT03549442',
           'NCT03070327',
           'NCT03448978',
           'NCT03502577',
           'NCT03338972',
           'NCT03672318',
           'NCT02529813',
           'NCT03601078']
```

```
In [91]: mask2[1].iloc[0]['NCT Number']
```

```
Out[91]: 'NCT03464916'
```

```
In [ ]: city_lng, city_lat, city_name = [],[],[]
        for i in range(len(city_site)):
            city_lng.append(city_site[i]['Lng'].tolist())
            city_lat.append(city_site[i]['Lat'].tolist())
            city_name.append(city_site[i]['City, State'])

        jit_lng, jit_lat, jit_trial_no, jit_trial_name = [],[],[],[]
        for i in range(len(mask3)):
            for j in range(len(mask3[i])):
                jit_lng.append(mask3[i].iloc[j]['Lng'])
                jit_lat.append(mask3[i].iloc[j]['Lat'])
                jit_trial_no.append(mask3[i].iloc[j]['NCT Number'])
                jit_trial_name.append(mask3[i].iloc[j]['Title'])
```

```
In [ ]: import folium
```

```
In [ ]: colors = []
        for i in range(len(mask3)):
            for j in range(len(mask3[i])):
                colors.append(mask3[i].iloc[j]['Color'])
```

```
In [ ]: map = folium.Map(location = [38.58, -99.09], zoom_start=3.5, prefer_canvas=True, tiles
        fg = folium.FeatureGroup(name = "My Map")


        for lat, lng, number, name, col in zip(jit_lat, jit_lng, jit_trial_no, jit_trial_name,
            fg.add_child(folium.CircleMarker(location = [lat, lng], popup = number + ', ' + na

        for lat, lng, city in zip(city_lat, city_lng, city_name):
            fg.add_child(folium.CircleMarker(location = [lat, lng], popup = city, radius = .2,

        segments = []
        for i in range(len(city_site)):
            for j in range(len(mask3[i])):
                segments.append(tuple([[city_site[i]['Lat'], city_site[i]['Lng']], [mask3[i].i

        for i in range(len(segments)):
            fg.add_child(folium.PolyLine(locations=segments[i], color="white", weight=.10, opa

        print 'CAR-T Trials in US as of ' + str(datetime.now())[:10]
        map.add_child(fg)
        map.save("Map1" + str(datetime.now())[:10] + ".html")
        map

In [ ]: NCT_No = []
        length = []

        for i in range(len(mask2)):
            NCT_No.append(mask2[i].iloc[0]['NCT Number'])
            length.append(len(mask2[i]))

In [ ]: plt.figure(figsize=(7,10))
        ax1 = plt.axes(frameon=False)
        barlist = plt.barh(df_['NCT Number'].tolist(), length, alpha = 0.8)
        for i in range(len(barlist)):
            barlist[i].set_color(color[i])
        plt.grid()
        plt.title('Legend and Number of Sites per Trials \n')
        plt.savefig('Legend_CART ' + str(datetime.now())[:10] + '.png', format='png', dpi=600,
        plt.show()

In [ ]: from matplotlib.pyplot import figure
        import mpld3

        fig = plt.figure(figsize=(7,10))
        plt.gca()
        plt.axes(frameon=False)
        barlist = plt.barh(df_['NCT Number'].tolist(), length, alpha = 0.8)
        for i in range(len(barlist)):
            barlist[i].set_color(color[i])




        plt.grid()
        plt.title('Legend and Number of Sites per Trials as of ' + str(datetime.now())[:10] +


        mpld3.display()

In [ ]: mpld3.save_html(fig,'Legend_' + str(datetime.now())[:10] +'.html')

In [ ]: df_['NCT Number'].tolist()

In [ ]: import pygal

        # from pygal.style import Style
        # custom_style = Style(
        #   background='transparent',
        #   plot_background='transparent',
        #   foreground='#53E89B',
        #   foreground_strong='#53A0E8',
        #   foreground_subtle='#630C0D',
        #   opacity='.6',
        #   opacity_hover='.9',
        #   transition='400ms ease-in',
        #   colors=colors)

In [ ]: # bar_chart = pygal.HorizontalBar()
        # for i in range(len(length)):

        #     bar_chart.add(df_['NCT Number'].tolist()[i], length[i])

        # bar_chart.render_to_file('bar_chart.svg')

In [ ]: import pygal

        from IPython.display import SVG, display
        from pygal.style import Style
        custom_style = Style(
          background='transparent',
          plot_background='transparent',
        #   foreground='#53E89B',
        #   foreground_strong='#53A0E8',
        #   foreground_subtle='#630C0D',

          opacity='.4',
          opacity_hover='.5',
          transition='400ms ease-in',
          colors=(color))

        # chart = pygal.StackedLine(fill=True, interpolate='cubic', )
        bar_chart = pygal.HorizontalBar(show_legend=False,  height = 1000, spacing = 1, style=

        for i in range(len(length)):
        #     bar_chart.add(df_['NCT Number'].tolist()[i], )
            bar_chart.add(df_['NCT Number'].tolist()[i], [{'value': length[i], 'label': df_['T

        # chart.add('A', [1, 3,  5, 16, 13, 3,  7])
        # chart.add('B', [5, 2,  3,  2,  5, 7, 17])
        # chart.add('C', [6, 10, 9,  7,  3, 1,  0])
        # chart.add('D', [2,  3, 5,  9, 12, 9,  5])
        # chart.add('E', [7,  4, 2,  1,  2, 10, 0])
        display({'image/svg+xml': bar_chart.render()}, raw=True)
        bar_chart.render_to_file('bar_chart1' + str(datetime.now())[:10] + '.svg')
```

In [ ]: