Climate Change Tweets

By: Mitchell Leahy, Matthew Houde, and Michael Andrejco

Imports

In [78]:
```
from pymongo import MongoClient
import pandas as pd
import matplotlib.ticker as ticker
import matplotlib.cm as cm
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec
import seaborn as sns

import matplotlib.pyplot as plt
%matplotlib inline
```

Mongo DB Connection

In [19]:
```
# connect to local mongodb database

MONGO_HOST='mongodb://localhost:27017/climatedb'

client=MongoClient(MONGO_HOST)

#select tweets database

db=client['climatedb']
```

1. Top Words in Tweets

In [68]:
```
def top_words(n):
    pipeline=[
    {"$addFields": {
            "words": {"$split": ["$text", " "]}
    }},
    {"$unwind": "$words"},
    {"$group": {
            "_id": "$words",
            "count": {"$sum":1}
        }},
    {"$sort": {"count":-1}},
    {"$limit":n}
    ]

    result = db.tweets.aggregate(pipeline)
    result_list = []

    for res in result:
        result_list.append(res)

    return result_list
```

In [69]:
```
top_10_words = top_words(20)
top_10_words
```

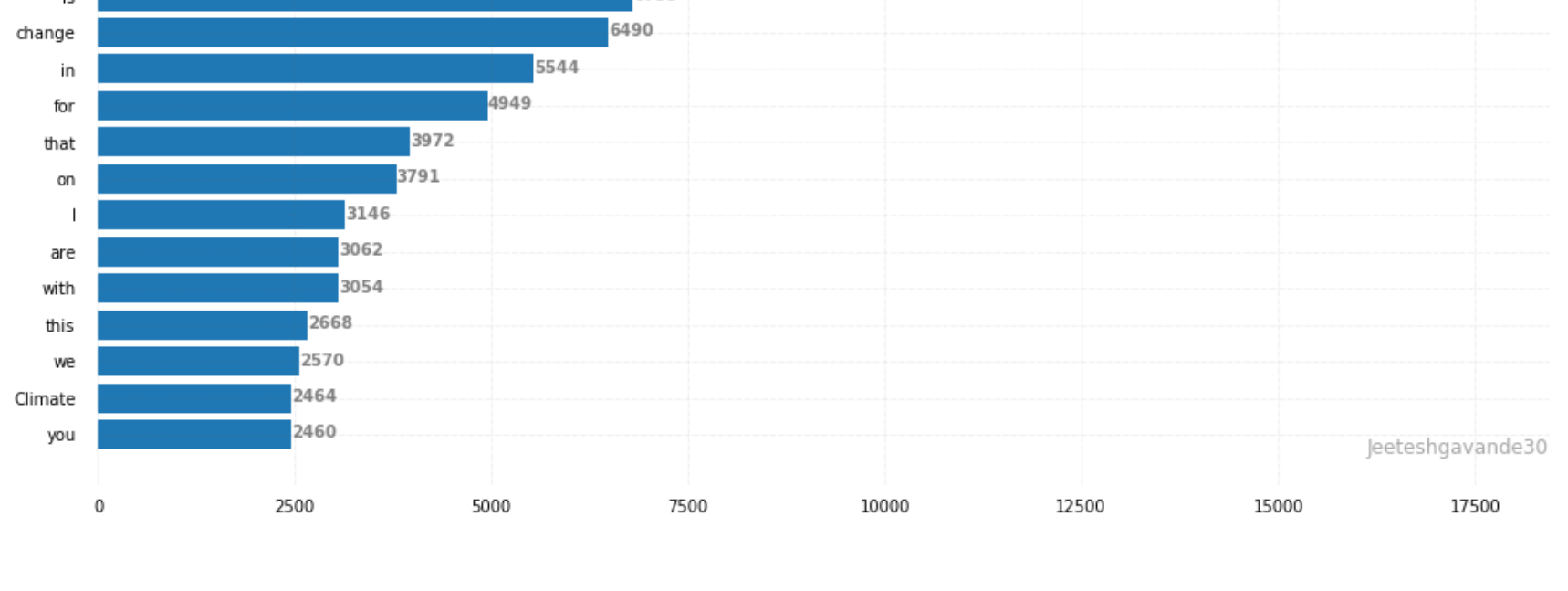
Out[69]:
```
[{'_id': 'RT', 'count': 17565},
 {'_id': 'the', 'count': 17407},
 {'_id': 'to', 'count': 12640},
 {'_id': 'climate', 'count': 10050},
 {'_id': 'of', 'count': 9559},
 {'_id': 'a', 'count': 7828},
 {'_id': 'and', 'count': 7603},
 {'_id': 'is', 'count': 6795},
 {'_id': 'change', 'count': 6490},
 {'_id': 'in', 'count': 5544},
 {'_id': 'for', 'count': 4949},
 {'_id': 'that', 'count': 3972},
 {'_id': 'on', 'count': 3791},
 {'_id': 'I', 'count': 3146},
 {'_id': 'are', 'count': 3062},
 {'_id': 'with', 'count': 3054},
 {'_id': 'this', 'count': 2668},
 {'_id': 'we', 'count': 2570},
 {'_id': 'Climate', 'count': 2464},
 {'_id': 'you', 'count': 2460}]
```

In [70]:
```
#covert the result to a python data frame
top_words_df=pd.DataFrame(top_10_words)
top_words_df
```

Out[70]:

	_id	count
0	RT	17565
1	the	17407
2	to	12640
3	climate	10050
4	of	9559
5	a	7828
6	and	7603
7	is	6795
8	change	6490
9	in	5544
10	for	4949
11	that	3972
12	on	3791
13	I	3146
14	are	3062
15	with	3054
16	this	2668
17	we	2570
18	Climate	2464
19	you	2460

In [71]:
```
# Figure Size
fig, ax = plt.subplots(figsize =(16, 9))

# Horizontal Bar Plot
ax.barh(top_words_df['_id'], top_words_df['count'])

# Remove axes splines
for s in ['top', 'bottom', 'left', 'right']:
    ax.spines[s].set_visible(False)

# Remove x, y Ticks
ax.xaxis.set_ticks_position('none')
ax.yaxis.set_ticks_position('none')

# Add padding between axes and labels
ax.xaxis.set_tick_params(pad = 5)
ax.yaxis.set_tick_params(pad = 10)

# Add x, y gridlines
ax.grid(b = True, color ='grey',
        linestyle ='-.', linewidth = 0.5,
        alpha = 0.2)

# Show top values
ax.invert_yaxis()

# Add annotation to bars
for i in ax.patches:
    plt.text(i.get_width()+0.2, i.get_y()+0.5,
             str(round((i.get_width()), 2)),
             fontsize = 10, fontweight ='bold',
             color ='grey')

# Add Plot Title
ax.set_title('Top 20 Words in Tweets',
             loc ='left', )

# Add Text watermark
fig.text(0.9, 0.15, 'Jeeteshgavande30', fontsize = 12,
         color ='grey', ha ='right', va ='bottom',
         alpha = 0.7)

# Show Plot
plt.show()
```



2. Number of Tweets per Hour

In [72]:
```
def tweets_by_hour():
    pipeline=[
    {"$addFields": {
  'hour': {'$hour': {'$toDate': '$created_at'}}
 }}, {'$group': {
  '_id': '$hour',
  'count': {'$sum':1}
 }},
 {'$sort':{'_id':1}}
    ]

    result = db.tweets.aggregate(pipeline)
    result_list = []

    for res in result:
        result_list.append(res)

    return result_list
```

In [73]:
```
tweets_by_hour = tweets_by_hour()
tweets_by_hour
```

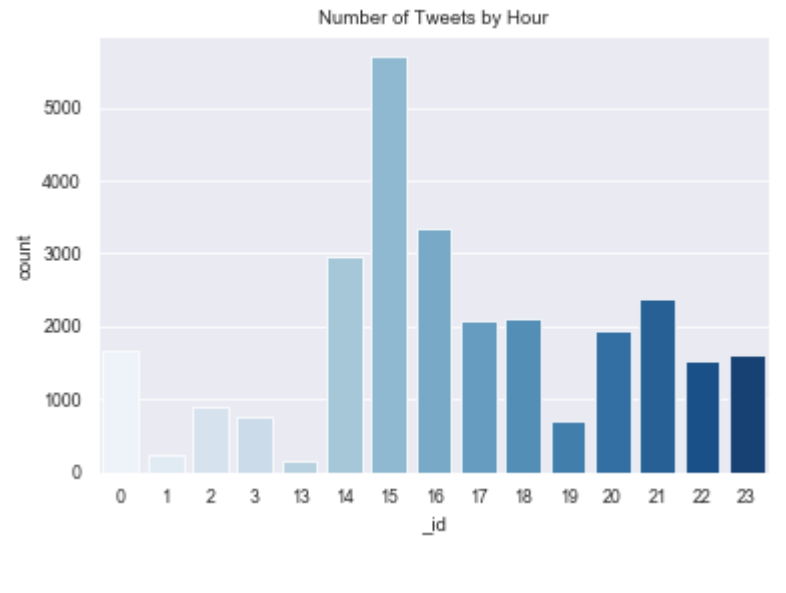
Out[73]:
```
[{'_id': 0, 'count': 1671},
 {'_id': 1, 'count': 229},
 {'_id': 2, 'count': 901},
 {'_id': 3, 'count': 763},
 {'_id': 13, 'count': 143},
 {'_id': 14, 'count': 2951},
 {'_id': 15, 'count': 5691},
 {'_id': 16, 'count': 3332},
 {'_id': 17, 'count': 2068},
 {'_id': 18, 'count': 2094},
 {'_id': 19, 'count': 698},
 {'_id': 20, 'count': 1937},
 {'_id': 21, 'count': 2388},
 {'_id': 22, 'count': 1514},
 {'_id': 23, 'count': 1607}]
```

In [74]:
```
hour_df=pd.DataFrame(tweets_by_hour)
hour_df
```

Out[74]:

	_id	count
0	0	1671
1	1	229
2	2	901
3	3	763
4	13	143
5	14	2951
6	15	5691
7	16	3332
8	17	2068
9	18	2094
10	19	698
11	20	1937
12	21	2388
13	22	1514
14	23	1607

In [79]:
```
sns.set_context('paper')

sns.barplot(x = '_id', y = 'count',  data = hour_df,
            palette = 'Blues').set_title("Number of Tweets by Hour")
sns.set(rc={'figure.figsize':(20,12)})
plt.show()
```



3. Top 5 Languages

In [80]:
```
def top_lang():
    pipeline = [
    {'$group': {
        '_id': '$lang',
        'count': {'$sum':1}
    }},
    {'$sort': {'count':-1}},
    {'$limit':6}
    ]

    result = db.tweets.aggregate(pipeline)
    result_list = []

    for res in result:
        result_list.append(res)

    return result_list
```

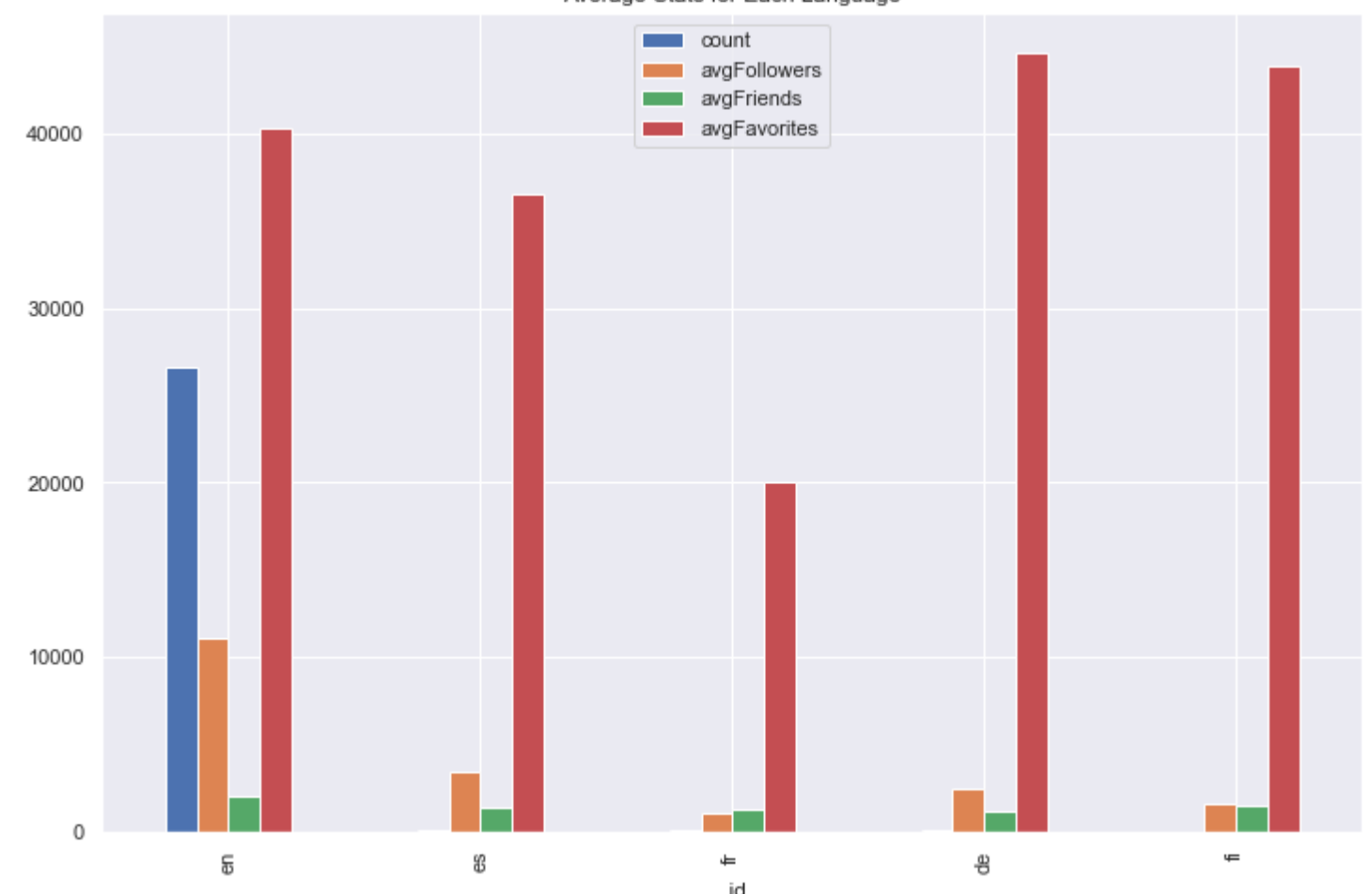
In [81]:
```
def top_lang():
    pipeline = [
    {'$group': {
        '_id': '$lang',
        'count': {'$sum':1},
        'avgFollowers':{'$avg':'$user.followers_count'},
        'avgFriends':{'$avg':'$user.friends_count'},
        'avgFavorites':{'$avg':'$user.favourites_count'}
    }},
    {'$sort': {'count':-1}},
    {'$limit':6},
    {'$project': {
        '_id': 1,
        'count': 1,
        'avgFollowers': {'$trunc': '$avgFollowers'},
        'avgFriends': {'$trunc': '$avgFriends'},
        'avgFavorites':{'$trunc':'$avgFavorites'}
    }}
    ]

    result = db.tweets.aggregate(pipeline)
    result_list = []

    for res in result:
        result_list.append(res)

    return result_list
```

In [82]:
```
lang_df=pd.DataFrame(top_lang())
lang_df.drop(1, inplace=True)
lang_df
```

Out[82]:

	_id	count	avgFollowers	avgFriends	avgFavorites
0	en	26655	11090.0	2003.0	40296.0
2	es	140	3508.0	1393.0	36516.0
3	fr	116	1107.0	1335.0	20074.0
4	de	94	2468.0	1239.0	44640.0
5	fi	37	1571.0	1510.0	43830.0

In [83]:
```
lang_df.plot(x="_id", y=["count", "avgFollowers", "avgFriends", 'avgFavorites'], kind="bar",figsize=(12
,8))
plt.title('Average Stats for Each Language')
plt.show()
```



4. Average Followers for Users That Tweeted About Climate Change

In [65]:
```
import plotly.express as px

def avg_followers(n):
    pipeline=[
    {"$group": {
        "_id": "null",
        "average_followers": {"$avg":"$user.followers_count"}}},
    {"$sort": {"count":-1}},
    {
        "$project": {
            "_id":1,
            "average_followers":1
        }
    }
]



    results = db.tweets.aggregate(pipeline)
    results_list=[]


    #for res in results:
        #print(res)
    results_list = list(results)
    return results_list
results_list = avg_followers(1)
results_pd = pd.DataFrame(list(results_list))
results_pd
```

Out[65]:

	_id	average_followers
0	null	10655.601136

5. Top 20 Hashtags

In [50]:
```
#Question 1: What are the top 20 hashtags used in the tweets?
def hashtags():
    pipeline=[
        {"$addFields": {"textArray": {"$split": ["$text", " "]}}},
        {"$unwind": "$textArray"},
        {"$addFields": {"textArray": {"$toLower": "$textArray"}}},
        {"$match": {"textArray": {"$regex":"^#"}}},
        {"$group": {
            "_id":  "$textArray",
            "frequency": {"$sum":1}
        }},
        {"$sort": {"frequency":-1}},
        {"$limit":20}
    ]

    result=db.tweets.aggregate(pipeline)
    result_list=[]

    for res in result:
        result_list.append(res)

    return result_list
```

In [51]:
```
result_list=hashtags()
result_list
```

Out[51]:
```
[{'_id': '#budget', 'frequency': 177},
 {'_id': '#solar', 'frequency': 170},
 {'_id': '#climate', 'frequency': 142},
 {'_id': '#climatechange', 'frequency': 140},
 {'_id': '#worldwildlifeday', 'frequency': 138},
 {'_id': '#impact', 'frequency': 116},
 {'_id': '#worldwildlifeday,', 'frequency': 98},
 {'_id': '#cancelcancelculture', 'frequency': 96},
 {'_id': '#cop26,', 'frequency': 89},
 {'_id': '#eriksensoreide', 'frequency': 85},
 {'_id': '#blackpink…', 'frequency': 82},
 {'_id': '#covid,', 'frequency': 82},
 {'_id': '#worldwithoutnature\n\nat', 'frequency': 65},
 {'_id': '#climateaction', 'frequency': 62},
 {'_id': '#timallen', 'frequency': 58},
 {'_id': '#lockdown', 'frequency': 56},
 {'_id': '#mustard', 'frequency': 55},
 {'_id': '#ev', 'frequency': 55},
 {'_id': '#market', 'frequency': 54},
 {'_id': '#farm', 'frequency': 54}]
```

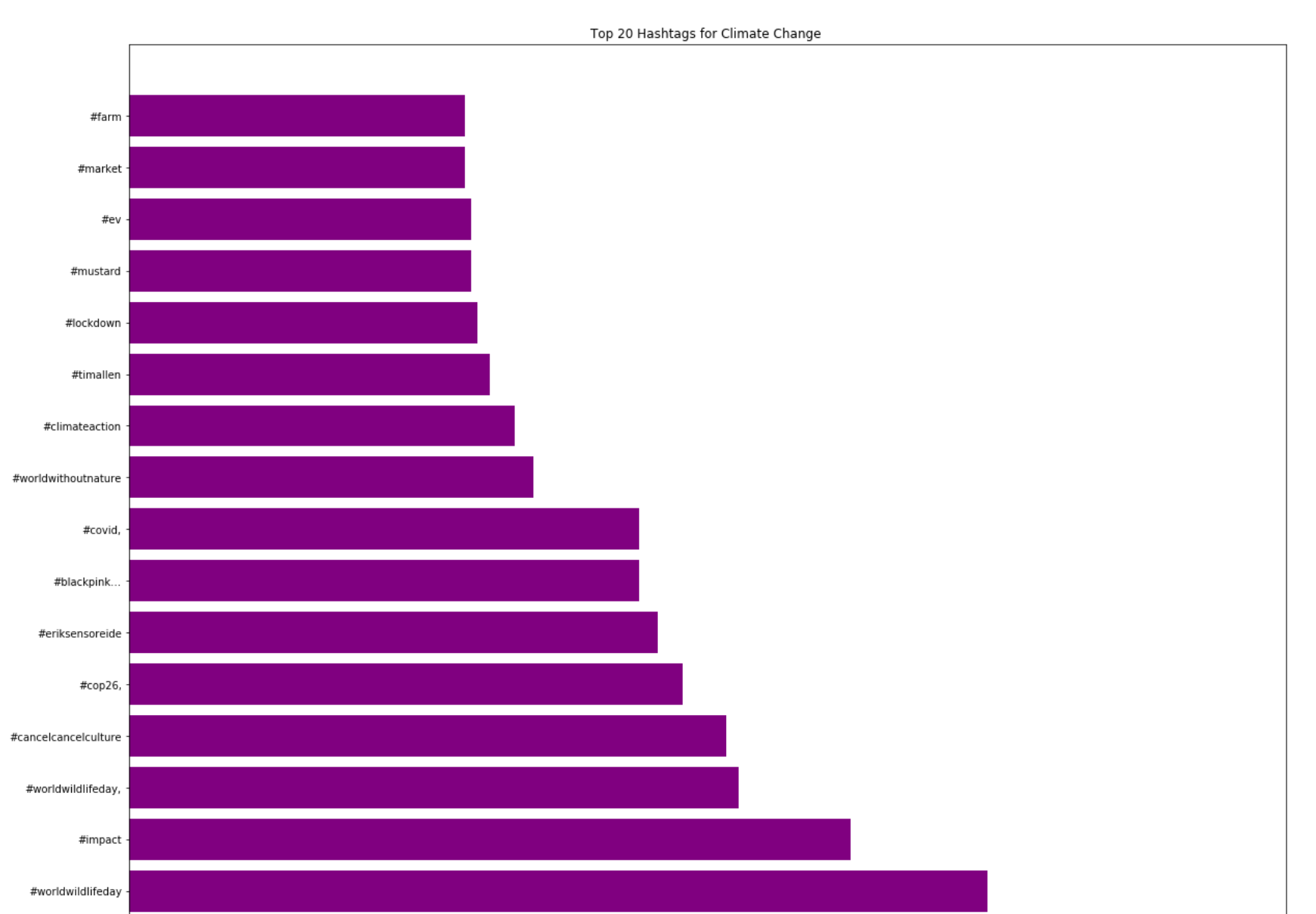
In [52]:
```
#covert the result to a python data frame
result_pd=pd.DataFrame(list(result_list))
result_pd.iloc[[12],[0]]='#worldwithoutnature'
result_pd
```

Out[52]:

	_id	frequency
0	#budget	177
1	#solar	170
2	#climate	142
3	#climatechange	140
4	#worldwildlifeday	138
5	#impact	116
6	#worldwildlifeday,	98
7	#cancelcancelculture	96
8	#cop26,	89
9	#eriksensoreide	85
10	#blackpink…	82
11	#covid,	82
12	#worldwithoutnature	65
13	#climateaction	62
14	#timallen	58
15	#lockdown	56
16	#mustard	55
17	#ev	55
18	#market	54
19	#farm	54

In [55]:
```
x=result_pd['_id']
y=result_pd['frequency']
plt.figure(figsize=(20,20))
plt.barh(x,y, color="purple")
plt.xlabel("Numer of Tweets")
plt.title("Top 20 Hashtags for Climate Change ")
```

Out[55]: `Text(0.5, 1.0, 'Top 20 Hashtags for Climate Change ')`



6. Top Users with most followers and hashtags used

In [56]:
```
#What are the top users 10 who have the most followers and what hashtags did they use?
def top_users_hashtags(amount_of_users):
    pipeline=[
        {"$addFields": {"textArray": {"$split": ["$text", " "]}}},
        {"$unwind": "$textArray"},
        {"$addFields": {"textArray": {"$toLower": "$textArray"}}},
        {"$match": {"textArray": {"$regex":"^#"}}},
        {"$group": {
            "_id":"$user.screen_name",
            "followers":{"$max":"$user.followers_count"},
            "hashtag":{"$push":"$textArray"}
        }},
        {"$sort": {"followers":-1}},
        {"$limit" :amount_of_users}
            ]

    result=db.tweets.aggregate(pipeline)
    result_list=[]

    for res in result:
        result_list.append(res)

    return result_list
```

In [57]:
```
result_list=top_users_hashtags(10)
result_list
```

Out[57]:
```
[{'_id': 'WWF', 'followers': 3990145, 'hashtag': ['#worldwildlifeday']},
 {'_id': 'World_Wildlife',
  'followers': 1480147,
  'hashtag': ['#worldwildlifeday']},
 {'_id': 'gulf_news', 'followers': 1198863, 'hashtag': ['#miyawaki']},
 {'_id': 'GhoshAmitav',
  'followers': 471508,
  'hashtag': ['#climateemergency', '#snow', '#himalayas']},
 {'_id': 'BrookingsInst',
  'followers': 423868,
  'hashtag': ['#brookingsblueprints']},
 {'_id': 'cabinetofficeuk',
  'followers': 416187,
  'hashtag': ['#cop26,', '#ppcasummit']},
 {'_id': 'mikeallen', 'followers': 410263, 'hashtag': ['#climate', '#energy']},
 {'_id': 'SierraClub', 'followers': 379833, 'hashtag': ['#zeroemissions']},
 {'_id': 'NRDC',
  'followers': 342537,
  'hashtag': ['#nepa',
   '#nepa',
   '#nepa',
   '#worldwildlifeday,',
   '#worldwithoutnature\n\nat',
   '#cop26,']},
 {'_id': 'SecBlinken', 'followers': 329817, 'hashtag': ['#eriksensoreide']}]
```

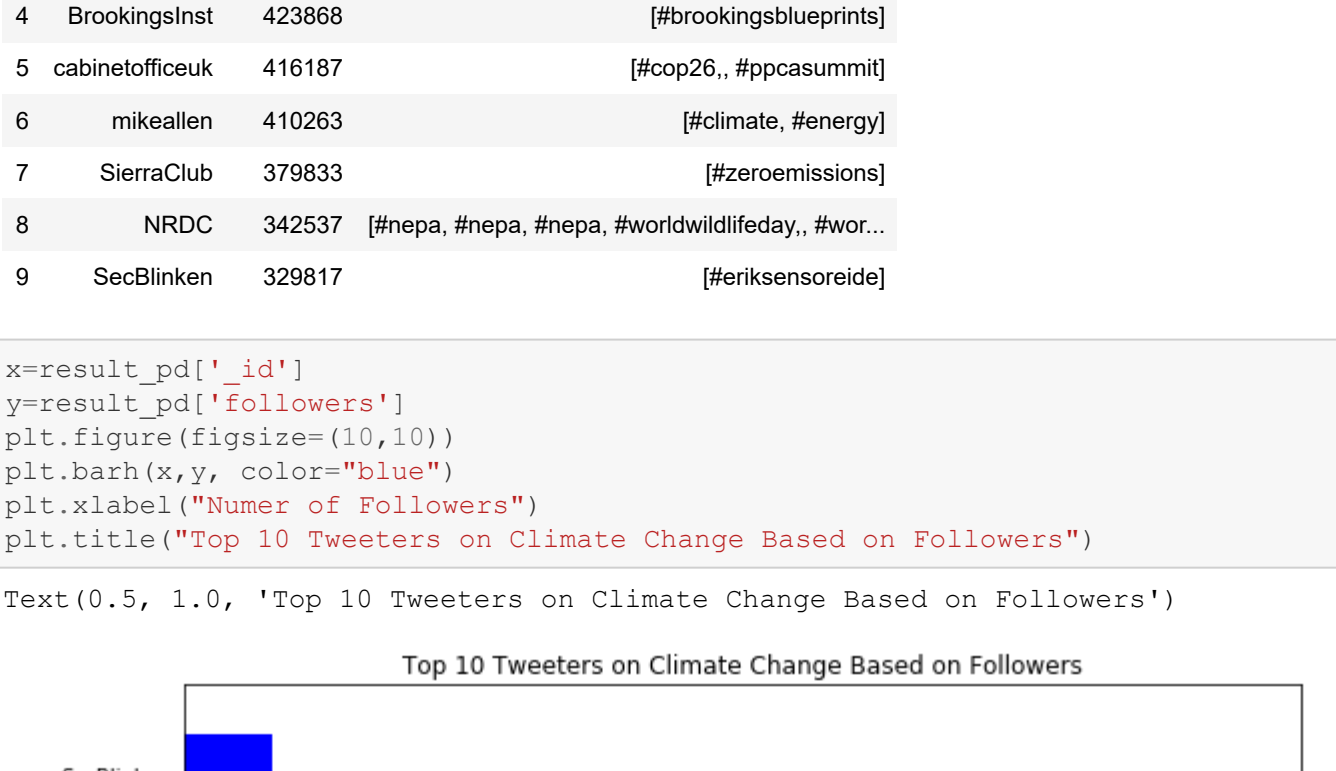
In [58]:
```
result_pd=pd.DataFrame(list(result_list))
result_pd
```

Out[58]:

	_id	followers	hashtag
0	WWF	3990145	[#worldwildlifeday]
1	World_Wildlife	1480147	[#worldwildlifeday]
2	gulf_news	1198863	[#miyawaki]
3	GhoshAmitav	471508	[#climateemergency, #snow, #himalayas]
4	BrookingsInst	423868	[#brookingsblueprints]
5	cabinetofficeuk	416187	[#cop26,, #ppcasummit]
6	mikeallen	410263	[#climate, #energy]
7	SierraClub	379833	[#zeroemissions]
8	NRDC	342537	[#nepa, #nepa, #nepa, #worldwildlifeday,, #wor...
9	SecBlinken	329817	[#eriksensoreide]

In [59]:
```
x=result_pd['_id']
y=result_pd['followers']
plt.figure(figsize=(10,10))
plt.barh(x,y, color="blue")
plt.xlabel("Numer of Followers")
plt.title("Top 10 Tweeters on Climate Change Based on Followers")
```

Out[59]: `Text(0.5, 1.0, 'Top 10 Tweeters on Climate Change Based on Followers')`



7. Countries and Climate Change

In [60]:
```
def cities_climate_talk():
    pipeline=[
        {"$group": {
            "_id":"$place.country",
            "count":{"$sum":1}
        }},
        {"$sort": {"count":-1}}
        ]

    result=db.tweets.aggregate(pipeline)
    result_list=[]

    for res in result:
        result_list.append(res)

    return result_list
```

In [61]:
```
result_list=cities_climate_talk()
result_list
```

Out[61]:
```
[{'_id': None, 'count': 27821},
 {'_id': 'United States', 'count': 83},
 {'_id': 'United Kingdom', 'count': 24},
 {'_id': 'Canada', 'count': 15},
 {'_id': 'Australia', 'count': 10},
 {'_id': 'Ireland', 'count': 6},
 {'_id': 'India', 'count': 4},
 {'_id': 'Nepal', 'count': 2},
 {'_id': 'Kenya', 'count': 2},
 {'_id': 'Sweden', 'count': 2},
 {'_id': 'Germany', 'count': 1},
 {'_id': 'Costa Rica', 'count': 1},
 {'_id': 'Virgin Islands, U.S.', 'count': 1},
 {'_id': 'Denmark', 'count': 1},
 {'_id': 'Jamaica', 'count': 1},
 {'_id': 'Sri Lanka', 'count': 1},
 {'_id': 'Republic of the Philippines', 'count': 1},
 {'_id': 'Liberia', 'count': 1},
 {'_id': 'Bangladesh', 'count': 1},
 {'_id': 'Brasil', 'count': 1},
 {'_id': 'België', 'count': 1},
 {'_id': 'New Zealand', 'count': 1},
 {'_id': 'Mexico', 'count': 1},
 {'_id': 'Belgium', 'count': 1},
 {'_id': 'Panamá', 'count': 1},
 {'_id': 'Honduras', 'count': 1},
 {'_id': 'Barbados', 'count': 1},
 {'_id': 'Singapore', 'count': 1}]
```

In [62]:
```
result_pd=pd.DataFrame(list(result_list))
result_pd
```

Out[62]:

	_id	count
0	None	27821
1	United States	83
2	United Kingdom	24
3	Canada	15
4	Australia	10
5	Ireland	6
6	India	4
7	Nepal	2
8	Kenya	2
9	Sweden	2
10	Germany	1
11	Costa Rica	1
12	Virgin Islands, U.S.	1
13	Denmark	1
14	Jamaica	1
15	Sri Lanka	1
16	Republic of the Philippines	1
17	Liberia	1
18	Bangladesh	1
19	Brasil	1
20	België	1
21	New Zealand	1
22	Mexico	1
23	Belgium	1
24	Panamá	1
25	Honduras	1
26	Barbados	1
27	Singapore	1

In [64]:
```python
import plotly.express as px
clist=['NA','USA','GBR','CAN','AUS','IRL','IND','SWE','EAK','NPL','LKA','HND','MEX','SGP','DEU','BEL',
'NZL','CRI','BGD','BRB','PAN','NA','BRA','PHL','LBR','VIR','DNK','JAM']
result_pd["iso_alpha"]=clist
fig = px.choropleth(result_pd[result_pd['iso_alpha']!='NA'], locations="iso_alpha",
                    color="count",
                    hover_name="_id",
                    color_continuous_scale=px.colors.sequential.Peach)

fig.update_layout(title_text='Count of Tweets by Country (Null Values: 27,821)')

fig.show()
```

8. Top Hashtags (Using Entities Object)

In [86]:
```python
def top_hashtags(n):
    pipeline=[
        {"$unwind": "$entities.hashtags"},
        {"$group": {
            "_id": "$entities.hashtags.text",
            "count": {"$sum":1}
        }
        },
        {"$sort": {"count":-1}},

        {"$limit":n},
        {
            "$project": {
                "_id":1,
                "count":1
            }
        }
    ]

    results = db.tweets.aggregate(pipeline)
    results_list=[]


    #for res in results:
        #print(res)
    results_list = list(results)
    return results_list
results_list = top_hashtags(10)
results_pd = pd.DataFrame(list(results_list))
results_pd.reset_index()
results_pd.columns=['Hashtag','Count']
results_pd.columns
```

Out[86]: Index(['Hashtag', 'Count'], dtype='object')

In [87]:
```python
import plotly.express as px



fig = px.bar(results_pd, y='Count', x='Hashtag')
fig.update_layout(uniformtext_minsize=8, uniformtext_mode='hide')
fig.update_layout(title='Popular Hashtags in Tweets About Climate')
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```

9. Top Hashtags in Spanish

In [22]:
```python
def top_hashtags(n):
    pipeline=[
    {"$match":{"lang":"es"}},
    {"$unwind": "$entities.hashtags"},
    {"$group": {
        "_id": "$entities.hashtags.text",
        "count": {"$sum":1}
    }
    },
    {"$sort": {"count":-1}},

    {"$limit":10},
    {
        "$project": {
            "_id":1,
            "count":1
        }
    }
]

    results = db.tweets.aggregate(pipeline)
    results_list=[]


    #for res in results:
        #print(res)
    results_list = list(results)
    return results_list
results_list = top_hashtags(10)
results_pd = pd.DataFrame(list(results_list))
results_pd.reset_index()
results_pd.columns=['Hashtag','Count']
results_pd.columns
```

Out[22]: Index(['Hashtag', 'Count'], dtype='object')

In [23]:
```python
import plotly.express as px



fig = px.bar(results_pd, y='Count', x='Hashtag')
fig.update_layout(uniformtext_minsize=8, uniformtext_mode='hide')
fig.update_layout(title='Most Popular Spanish Hashtags About Climate')
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```

10. People Who Tweeted the Most About Climate

In [26]:
```python
import plotly.express as px

def top_users(n):
    pipeline=[
    {"$group": {
        "_id": "$user.screen_name",
        "count": {"$sum":1}
    }
    },
    {"$sort": {"count":-1}},
    {"$limit":10}
]

    results = db.tweets.aggregate(pipeline)
    results_list=[]


    #for res in results:
        #print(res)
    results_list = list(results)
    return results_list
results_list = top_users(10)
results_pd = pd.DataFrame(list(results_list))
results_pd.reset_index()
results_pd.columns=['User','Count']
results_pd.columns


fig = px.bar(results_pd, y='Count', x='User')
fig.update_layout(uniformtext_minsize=8, uniformtext_mode='hide')
fig.update_layout(title='Users Who Tweeted The Most About Climate change')
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```

11. Who is Retweeted the most

In [88]:
```python
import plotly.express as px

def top_retweeters(n):
    pipeline=[
    {"$group": {
        "_id": "$retweeted_status.user.screen_name",
        "count": {"$sum":1}
    }
    },
    {"$sort": {"count":-1}},
    {"$limit":10}
]

    results = db.tweets.aggregate(pipeline)
    results_list=[]


    #for res in results:
        #print(res)
    results_list = list(results)
    return results_list
results_list = top_retweeters(10)
results_pd = pd.DataFrame(list(results_list))
results_pd.reset_index()
results_pd.columns=['User','Count']



fig = px.bar(results_pd, y='Count', x='User')
#fig.update_layout(uniformtext_minsize=8, uniformtext_mode='hide')
fig.update_layout(title='Top Retweeted Users')
fig.update_layout(yaxis_range=[0,600])
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```

In [42]:
```python
results_pd
```

Out[42]:

	User	Count
0	None	10447
1	VP	455
2	POTUS	289
3	Peston	252
4	JesseKellyDC	209
5	disclosetv	179
6	RLong_Bailey	175
7	kylenabecker	164
8	wtpBLUE	158
9	bobscartoons	158