

Exploring Google Transparency Reports Political Ad Data for US Election Cycle Prepared by: Mehr Qayyum

```
#Upload Google Political Ads file via Priorities USA and mount Google Drive to access  
from google.colab import drive  
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import numpy as np  
import pandas as pd  
import seaborn as sns  
import scipy as sp  
import matplotlib.pyplot as plt
```










Load Data

```
#problem reading at line at 37505  
maga = pd.read_csv('/content/google-political-ads-creative-stats.csv', engine= 'python')  
maga
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

Ad ID

0	CR104193501419274240	https://transparency
1	CR109957691127889920	https://transparency
2	CR111129083328331776	https://transparency

**1B) Preliminary Exploration *

```
#Identify columns
maga.info()
```

```
>>> class 'pandas.core.frame.DataFrame'>
RangeIndex: 310273 entries, 0 to 310272
Data columns (total 50 columns):
#      Column                                Non-Null Count
---  -
0      Ad_ID                                310273 non-null
1      Ad_URL                               310273 non-null
2      Ad_Type                              310272 non-null
3      Regions                             310272 non-null
4      Advertiser_ID                       310272 non-null
5      Advertiser_Name                     310272 non-null
6      Ad_Campaigns_List                   0 non-null
7      Date_Range_Start                   310272 non-null
8      Date_Range_End                     310272 non-null
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

11	Spend_USD	244291	non-null
12	First_Served_Timestamp	310264	non-null
13	Last_Served_Timestamp	310264	non-null
14	Age_Targeting	310272	non-null
15	Gender_Targeting	310272	non-null
16	Geo_Targeting_Included	310272	non-null
17	Geo_Targeting_Excluded	310272	non-null
18	Spend_Range_Min_USD	310272	non-null
19	Spend_Range_Max_USD	309821	non-null
20	Spend_Range_Min_EUR	310272	non-null
21	Spend_Range_Max_EUR	309579	non-null
22	Spend_Range_Min_INR	310272	non-null
23	Spend_Range_Max_INR	293449	non-null
24	Spend_Range_Min_BGN	310272	non-null
25	Spend_Range_Max_BGN	306264	non-null

```

26 Spend_Range_Min_HRK 310272 non-null
27 Spend_Range_Max_HRK 307867 non-null
28 Spend_Range_Min_CZK 310272 non-null
29 Spend_Range_Max_CZK 309098 non-null
30 Spend_Range_Min_DKK 310272 non-null
31 Spend_Range_Max_DKK 309654 non-null
32 Spend_Range_Min_HUF 310272 non-null
33 Spend_Range_Max_HUF 308029 non-null
34 Spend_Range_Min_PLN 310272 non-null
35 Spend_Range_Max_PLN 308141 non-null
36 Spend_Range_Min RON 310272 non-null
37 Spend_Range_Max RON 307071 non-null
38 Spend_Range_Min_SEK 310272 non-null
39 Spend_Range_Max_SEK 309708 non-null
40 Spend_Range_Min_GBP 310272 non-null
41 Spend_Range_Max_GBP 309525 non-null
42 Spend_Range_Min_ILS 310272 non-null
43 Spend_Range_Max_ILS 309608 non-null
44 Spend_Range_Min_NZD 310272 non-null
45 Spend_Range_Max_NZD 309818 non-null
46 Spend_Range_Min_TWD 310272 non-null
47 Spend_Range_Max_TWD 309843 non-null
48 Spend_Range_Min_AUD 310272 non-null
49 Spend_Range_Max_AUD 309590 non-null
dtypes: float64(34), object(16)
memory usage: 118.4+ MB

```

maga.columns

```

Index(['Ad_ID', 'Ad_URL', 'Ad_Type', 'Region',
      'Advertiser_Name', 'Ad_Campaigns_List',
      'Date_Range_End', 'Num_of_Days', 'Impressions',
      'First_Served_Timestamp', 'Last_Served_Timestamp',
      'Gender_Targeting', 'Geo_Targeting_Region',
      'Spend_Range_Min_USD', 'Spend_Range_Max_USD',
      'Spend_Range_Min_EUR', 'Spend_Range_Max_EUR',
      'Spend_Range_Min_BGN', 'Spend_Range_Max_BGN',
      'Spend_Range_Min_HRK', 'Spend_Range_Max_HRK',
      'Spend_Range_Min_DKK', 'Spend_Range_Max_DKK',
      'Spend_Range_Min_ILS', 'Spend_Range_Max_ILS',
      'Spend_Range_Min_NZD', 'Spend_Range_Max_NZD',
      'Spend_Range_Min_AUD', 'Spend_Range_Max_AUD'],
      dtype='object')

```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```

      'Spend_Range_Min_ILS', 'Spend_Range_Max_ILS',
      'Spend_Range_Min_NZD', 'Spend_Range_Max_NZD',
      'Spend_Range_Min_AUD', 'Spend_Range_Max_AUD'],
      dtype='object')

```

#What's the breakdown per column
maga.value_counts

```

<bound method DataFrame.value_counts of
0      CR104193501419274240 ...
1      CR109957691127889920 ...
2      CR111129083328331776 ...
3      CR112145169511350272 ...

```

```

4          CR112255773509156864 ...
...          ...
310268    CR498892749584465920 ...
310269    CR499011153242882048 ...
310270    CR49915422719344640 ...
310271    CR4993981813358592 ...
310272    CR499452950758817792 ...

```

```
[310273 rows x 50 columns]>
```

Part 1: Demonstration

```
maga.describe()
```

	Ad_Campaigns_List	Num_of_Days	Spend
count	0.0	310272.000000	
mean	NaN	13.093718	
std	NaN	27.107426	
min	NaN	1.000000	
25%	NaN	3.000000	
50%	NaN	6.000000	
75%	NaN	14.000000	
max	NaN	780.000000	

```
maga.value_counts('Advertiser_Name').head(50)
```

```
Advertiser_Name
BIDEN FOR PRESIDENT
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```

JEXAN LLC
DNC SERVICES CORP / DEMOCRATIC NATIONAL COMM
CONSERVATIVE BUZZ LLC
BEACHSIDE MEDIA INC
TULSI NOW
NRCC
INDEPENDENCE USA PAC
NRSC
PETE FOR AMERICA, INC.
NATURAL RESOURCES DEFENSE COUNCIL, INC.
FUSE WASHINGTON
SENATE LEADERSHIP FUND
Dravida Munnetra Kazhagam
Progresívne Slovensko

```

Atelierul de Internet SRL
 Need to Impeach
 MCSALLY FOR SENATE INC
 PARTIDUL SOCIAL DEMOCRAT - P.S.D.
 DCCC
 IMAGEWORKS, LLC
 SWALWELL FOR AMERICA
 NATIONAL HORIZON
 JAIME HARRISON FOR US SENATE
 DSCC
 Most-Híd
 EPOCH USA INC.
 Sloboda a Solidarita
 FRIENDS OF SPENCER COX
 THE LINCOLN PROJECT
 AMERICANS FOR TAX REFORM
 DFL HOUSE CAUCUS
 Komitet Wyborczy Prawo i Sprawiedliwość
 Sloboda a Solidarita (SaS)
 Magnet Media
 GALE PARTNERS LLC
 THE COMMITTEE TO DEFEND THE PRESIDENT
 Bündnis90/ Die Grünen
 PUBLIC ADVOCATE OF THE UNITED STATES
 CLEARPATH ACTION FUND, INC.
 Kansallinen Kokoomus r.p.
 CAL FOR NC
 ΣΥ.ΠΙΖ.Α.
 Bündnis 90/Die Grünen Baden-Württemberg
 PARTIDUL SOCIAL DEMOCRAT
 MOST nezavisnih lista
 HOMEOWNERS AND TENANTS UNITED
 dtype: int64

Review top 50, Flag MAGA/GOP, and group under
MAGA: Threshold:

1. DONALD J. TRUMP FOR PRESIDENT, INC.

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

4. NRSC (Source: <https://www.nrsc.org/about-us/>)
5. Conservative Buzz, LCC. (Source: <https://www.engadget.com/2019-11-11-google-political-ads-polls-email-collection.html>)
6. Beachside Media (Source: <https://transparencyreport.google.com/political-ads/advertiser/AR24814465610416128>)
7. Senate Leadership Fund (Source: <https://www.senateleadershipfund.org/about/>)



Mehr Qayyum

3:40 PM Today

[Resolve](#)



NON US groups buying ads, and to be removed b/c not in US: Kansallinen Kokoomus r.p.

8. AMERICANS FOR TAX REFORM (Source: <https://www.atr.org/>)
9. Committee To Defend the President (Source: <https://www.factcheck.org/2020/02/committee-to-defend-the-president-2/>)
10. PUBLIC ADVOCATE OF THE UNITED STATES (Source: <https://www.publicadvocateusa.org/>)
11. CLEARPATH ACTION FUND, INC. (Elects GOP candidates supporting "clean energy" like nuclear ~Source: <https://clearpathactionfund.org/>)

Notes: Reviewed those not intuitively associated with MAGA movement


- Confirmed Progressive: <https://fusewashington.org/news/fuses-2020-endorsements>
- Top 23 spenders spent more than \$2K

maga.corr()

Automatic saving failed. This file was updated remotely or in another tab.
[diff](#)

[Show](#)

Ad_Campaigns_List	
Ad_Campaigns_List	NaN
Num_of_Days	NaN
Spend_Range_Min_USD	NaN
Spend_Range_Max_USD	NaN
Spend_Range_Min_EUR	NaN
Spend_Range_Max_EUR	NaN
Spend_Range_Min_INR	NaN
Spend_Range_Max_INR	NaN
Spend_Range_Min_BGN	NaN
Spend_Range_Max_BGN	NaN
Spend_Range_Min_HRK	NaN
Spend_Range_Max_HRK	NaN
Spend_Range_Min_CZK	NaN
Spend_Range_Max_CZK	NaN
Spend_Range_Min_DKK	NaN
Spend_Range_Max_DKK	NaN
Spend_Range_Min_HUF	NaN
Spend_Range_Max_HUF	NaN
Spend_Range_Min_PLN	NaN
Spend_Range_Max_PLN	NaN

 Mehr Qayyum

2:58 PM Today

Resolve

'Spend_Range_Min_EUR',
--- 'Spend_Range_Max_EUR',
'Spend_Range_Min_INR',
'Spend_Range_Max_INR',
'Spend_Range_Min_BGN',
'Spend_Range_Max_BGN',
'Spend_Range_Min_HRK',
'Spend_Range_Max_HRK',
'Spend_Range_Min_CZK',
'Spend_Range_Max_CZK',
'Spend_Range_Min_DKK',
'Spend_Range_Max_DKK',
'Spend_Range_Min_HUF',
'Spend_Range_Max_HUF',
'Spend_Range_Min_PLN',
'Spend_Range_Max_PLN',
'Spend_Range_Min RON',
'Spend_Range_Max RON',
'Spend_Range_Min SEK',
'Spend_Range_Max SEK',

2. Cleaning

Automatic saving failed. This file was updated remotely or in another tab. [Show](#)

diff

columns:

```
.drop(columns=['Spend_Range_Min_EUR', 'Spend_Range_Min_BGN', 'Spend_Range_Max_BGN', 'Spend_Range_Max_HRK', 'Spend_Range_Min_CZK', 'Spend_Range_Min_DKK', 'Spend_Range_Max_DKK', 'Spend_Range_Max_HUF', 'Spend_Range_Min_PLN', 'Spend_Range_Min RON', 'Spend_Range_Max RON', 'Spend_Range_Max SEK', 'Spend_Range_Min GBP', 'Spend_Range_Max GBP', 'Spend_Range_Min ILS', 'Spend_Range_Max ILS', 'Spend_Range_Min NZD', 'Spend_Range_Max NZD', 'Spend_Range_Min TWD', 'Spend_Range_Max TWD', 'Spend_Range_Min AUD', 'Spend_Range_Max AUD'])
```

'Spend_Range_Max_ILS',
'Spend_Range_Min_NZD',
'Spend_Range_Max_NZD',
'Spend_Range_Min_TWD',
'Spend_Range_Max_TWD',
'Spend_Range_Min_AUD',
'Spend_Range_Max_AUD']

Ad_ID

0	CR104193501419274240	https://transparency
1	CR109957691127889920	https://transparency
2	CR111129083328331776	https://transparency
3	CR112145169511350272	https://transparency
4	CR112255773509156864	https://transparency
...
310268	CR498892749584465920	https://transparency
310269	CR499011153242882048	https://transparency
310270	CR49915422719344640	https://transparency
310271	CR4993981813358592	https://transparency
310272	CR499452950758817792	https://transparency

310273 rows × 20 columns

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

data set

maga_only.columns

```
Index(['Ad_ID', 'Ad_URL', 'Ad_Type', 'Region', 'Advertiser_Name', 'Ad_Campaigns_List', 'Date_Range_End', 'Num_of_Days', 'Impressions', 'First_Served_Timestamp', 'Last_Served_Timestamp', 'Gender_Targeting', 'Geo_Targeting_Indicator', 'Spend_Range_Min_USD', 'Spend_Range_Max_USD'],
      dtype='object')
```

Top Spender Filter maga_only further by selecting only
Ads affiliated with MAGA movement & GOP as identified

in value counts. And also area ads by removing rows
that aren't associated with US viewing

```
#Splicing rows for only DONALD J. TRUMP FOR PRESIDENT, INC.  
drumpf = maga_only.loc[maga_only['Advertiser_Name'] == 'DONALD J. TRUMP FOR PRESID  
drumpf
```

Ad_ID		
6191	CR100017006501167104	https://transparencyr
6192	CR100021679425585152	https://transparencyr
6193	CR100022778937212928	https://transparencyr
6194	CR100063735745347584	https://transparencyr
6195	CR100064148062208000	https://transparencyr
...	...	
33616	CR99898259245367296	https://transparencyr
33617	CR99899908512808960	https://transparencyr

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

33619	CR99923822890713088	https://transparencyr
33620	CR9992567831658496	https://transparencyr

27430 rows x 20 columns

Second Top Spender

```
#Splicing rows for only REPUBLICAN NATIONAL COMMITTEE
```

```
rnc = maga_only.loc[maga_only['Advertiser_Name'] == 'REPUBLICAN NATIONAL COMMITTEE']
rnc
```

Ad_ID		
274852	CR100070195376160768	https://transparency
274853	CR100095071826739200	https://transparency
274854	CR100109434197377024	https://transparency
274855	CR100173205871788032	https://transparency
274856	CR100313943360143360	https://transparency
...	...	
290099	CR99944507453210624	https://transparency
290100	CR99956464642162688	https://transparency
290101	CR99961481163964416	https://transparency
290102	CR99962099639255040	https://transparency
290103	CR99981890848555008	https://transparency

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

15252 rows × 20 columns

Third Top Spender

```
#Splicing rows for only NRCC
```

```
nrcc = maga_only.loc[maga_only['Advertiser_Name'] == 'NRCC']
nrcc
```

Ad_ID		
145348	CR100158912220626944	https://transparency
145349	CR100248659857244160	https://transparency
145350	CR100404103313620992	https://transparency
145351	CR100589783339761664	https://transparency
145352	CR100603802113015808	https://transparency
...
150712	CR99172650290511872	https://transparency
150713	CR99252777200386048	https://transparency
150714	CR9948106330210304	https://transparency
150715	CR99498586768670720	https://transparency
150716	CR99569436549185536	https://transparency

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

Fourth Top Spender

```
#Filter maga_only further by selecting only Ads affiliated with MAGA movement & GO
# And also area ads by removing rows that aren't associated with US viewing
#Selecting NRCC as example of 1 of top 8

nrsc = maga_only.loc[maga_only['Advertiser_Name'] == 'NRSC']
nrsc
```

Ad_ID

177807 CR100131355710455808 <https://transparency>

177808 CR100202136771493888 <https://transparency>

177809 CR100312225373224960 <https://transparency>

177810 CR100863080698740736 <https://transparency>

177811 CR10096196802576384 <https://transparency>

...

...

182377 CR99645783887839232 <https://transparency>

182378 CR99926021913968640 <https://transparency>

182379 CR99951997876174848 <https://transparency>

182380 CR99973438352916480 <https://transparency>

182381 CR99988831515705344 <https://transparency>

1575 rows × 30 columns

Automatic saving failed. This file was updated remotely or in another tab.
[diff](#)

[Show](#)

Fifth Top Spender

```
conbuzz = maga_only.loc[maga_only['Advertiser_Name'] == 'CONSERVATIVE BUZZ LLC']
conbuzz
```

Ad_ID

112499 CR100031162713374720 https://transparency

112500 CR100034117650874368 https://transparency

112501 CR100070538973544448 https://transparency

112502 CR100089505549123584 https://transparency

112503 CR100204473233702912 https://transparency

...

...

118927 CR9980404484276224 https://transparency

118928 CR99814009166888960 https://transparency

118929 CR99842252871827456 https://transparency

118930 CR99904512717750272 https://transparency

Sixth Top Spender:

118931 CR99915507834028032 https://transparency

```
beach = maga_only.loc[maga_only['Advertiser_Name'] == 'Beachside Media ']
beach
```

Automatic saving failed. This file was updated remotely or in another tab.

[Show diff](#)

Seventh Top Spender

```
slf = maga_only.loc[maga_only['Advertiser_Name'] == 'Senate Leadership Fund']
slf
```

Ad_ID Ad_URL Ad_Type Regions Advertise

Eighth Top Spender

Subset of Americans for Tax Reform Group

```
tax = maga_only.loc[maga_only['Advertiser_Name'] == 'AMERICANS FOR TAX REFORM']
tax
```

Ad_ID		
206313	CR100136647110164480	https://transparency
206314	CR100303772877586432	https://transparency
206315	CR100588408950226944	https://transparency
206316	CR101198088147828736	https://transparency
206317	CR101217398320791552	https://transparency
...	...	
207400	CR97859421090086912	https://transparency
207401	CR98385812281884672	https://transparency
207402	CR99275866944569344	https://transparency

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

207403	CR99397775296299008	https://transparency
207404	CR99911659543330816	https://transparency

1092 rows x 20 columns

Ninth Top Spender

```
defend = maga_only.loc[maga_only['Advertiser_Name'] == 'Committee To Defend the Pre]
```

```
defend
```

Ad_ID	Ad_URL	Ad_Type	Regions	Advertise
-------	--------	---------	---------	-----------

Tenth Top Spender

```
pres = maga_only.loc[maga_only['Advertiser_Name'] == 'PUBLIC ADVOCATE OF THE UNITED STATES']
pres
```

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

Ad_ID

290817 CR100076311409590272 <https://transparency>

290818 CR101090542166736896 <https://transparency>

Elevnth Top Spender

```
#Subset of CLEARPATH ACTION FUND, INC. (Elects GOP candidats supporting "clean ene:  
clear = maga_only.loc[maga_only['Advertiser_Name'] == 'CLEARPATH ACTION FUND, INC.'  
clear
```

Automatic saving failed. This file was updated remotely or in another tab.
[diff](#)

[Show](#)

Ad_ID

185809 CR102835123522633728 <https://transparency>

185810 CR103589938255101952 <https://transparency>

3. Exploring Filtered Data

185811 CR103603420787066368 <https://transparency>

```
#Concatenate the 11 subsets to create the 'maga_gop_drumpf_file' for focused explo:
# ([fb, fb_likes], ignore_index=True, sort =True)
maga_gop_drumpf_file = pd.concat([drumpf, rnc, nrcc, nrsc, conbuzz, beach, slf, ta:
maga_gop_drumpf_file
```

Automatic saving failed. This file was updated remotely or in another tab.
[diff](#)

[Show](#)

l_Timestamp	Num_of_Days	Regions	Spend_Range
-------------	-------------	---------	-------------

03T23:15:00Z	2.0	US	
--------------	-----	----	--

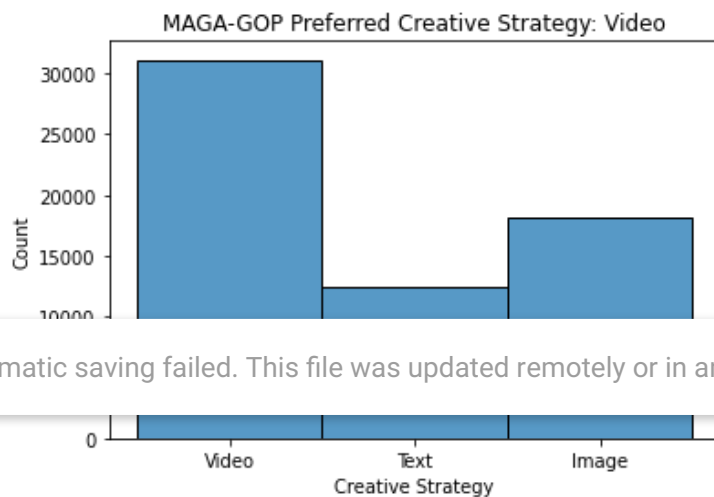
Our MAGA-GOP focused data subset includes 61735 records

```
#Review Creative Strategy by exploring 'Ad_Type'
maga_gop_drumpf_file.value_counts('Ad_Type')
```

```
Ad_Type
Video      31118
Image      18136
Text       12481
dtype: int64
```

Visualization #1: Most Popular MAGA-GOP Ad Type

```
ax = sns.histplot(data=maga_gop_drumpf_file, x="Ad_Type")
ax.set_title('MAGA-GOP Preferred Creative Strategy: Video')
ax.set_xlabel('Creative Strategy')
plt.show()
```



Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

estimate spending, we need to show the number of times for each ad category applied for political ad (Advertiser_Name associated with Spend_USD)

```
maga_gop_drumpf_file.value_counts('Spend_USD')
```

```
Spend_USD
≤ 100      35786
```

```

100-1k      16036
1k-50k      9536
> 100k      198
50k-100k    179
dtype: int64

```

Estimate on Spending: For an estimate on spending, I propose calculating a threshold where we calculate two sums: one at a minimum and one at a maximum.

1. Minimum Threshold: add up each minimum range at the lowest 'Spend_USD' column multiplied by the count for that category, and repeat for the remaining 4 categories (100 *from the* 100-1k group; 1k *from the* 1k – 50k group; 50k from the 50k-100k group, and 100,001 from the '>100k' group) Minimum Threshold = (
 $1 * 35,786) + (100 * 16,036) + (1000 * 9536) + (50,000 * 179) + (100,000 * 198)$
2. Maximum Threshold: Apply same process, but select the upper range: 100 *from group* 1; 1k from group 2; 50k *from group* 3; 100k from group 4; and use the

Maximum Threshold: $(100 * 35,786) + (1000 * 16,036) + (50000 * 9536) + (100,000 * 179) + (101,000 * 198)$

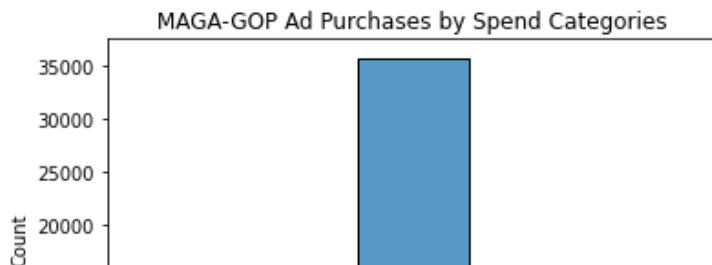
Visualization 2: US Dollars Spent by Groups Supporting MAGA Candidate

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```

ax = sns.histplot(data=maga_gop_drumpf_file, x="Spend_USD")
ax.set_title('MAGA-GOP Ad Purchases by Spend Categories')
ax.set_xlabel('US Dollars Spent')
plt.show()

```



Duration: How Long are they running ads?



```
maga_gop_drumpf_file.value_counts('Num_of_Days')
```

```
Num_of_Days
2.0      9419
1.0      6737
7.0      5601
3.0      4387
5.0      4065
...
284.0      1
279.0      1
278.0      1
274.0      1
442.0      1
Length: 318, dtype: int64
```

Visualization 3 & 4: Highlighting Ad Types by Advertiser across Number of Days

*Note: Unable to finish debugging my code on swarmplot visualization. Goal was to provide context for the top 8 MAGA-GOP Ad buy groups organized by Ad_Type versus the count of Num_of_Days(Number of Days that the purchased ad ran from start to end)as

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

```
sns.swarmplot(x='Ad_Type',y='Num_of_Days', hue='Advertiser_Name',data=maga_gop_drumpf_file)
```

```
-----
-----
KeyboardInterrupt
Traceback (most recent call last)
<ipython-input-62-9e5716b2a60c> in <module>
()
----> 1
sns.swarmplot(x='Ad_Type',y='Num_of_Days',
hue='Advertiser_Name',data=maga_gop_drumpf_1
```

```
----- 7 frames -----
<__array_function__ internals> in
all(*args, **kwargs)

/usr/local/lib/python3.7/dist-
packages/numpy/core/fromnumeric.py in
_all_dispatcher(a, axis, out, keepdims)
2331
2332
-> 2333 def _all_dispatcher(a, axis=None,
out=None, keepdims=None):
2334     return (a, out)
2335
```

```
swarmplot(x='Advertiser_Name',y='Num_of_Days', hue='Ad_Type',data=maga_gop_drumpf_
```

Automatic saving failed. This file was updated remotely or in another tab.
[diff](#)

[Show](#)

```
-----
KeyboardInterrupt
Traceback (most recent call last)
<ipython-input-63-3352d8cf2af6> in <module>
()
----> 1
sns.swarmplot(x='Advertiser_Name',y='Num_of_
hue='Ad_Type',data=maga_gop_drumpf_file)
```

```
----- 6 frames -----
/usr/local/lib/python3.7/dist-
packages/seaborn/categorical.py in
first_non_overlapping_candidate(self,
candidates, neighbors, d)
    1227         dy = neighbors_y - y_i
    1228
-> 1229         sq_distances =
np.power(dx, 2.0) + np.power(dy, 2.0)
    1230
    1231         # good candidate does
not overlap any of neighbors
```

KeyboardInterrupt:

SEARCH STACK OVERFLOW

Error in callback <function install_repl_dis

KeyboardInterrupt

'Let's pinpoint the top 15 states receiving Geo-Targeted ads by MAGA-GOP and
'And we will use to create a category for crosstab analysis by Advertiser Name
maga_gop_drumpf_file.value_counts('Geo_Targeting_Included').head(15)
'Skip first

Geo_Targeting_Included	
United States	13875
Georgia	2819

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

Wisconsin	1824
North Carolina	1699
Pennsylvania	1624
Michigan	1446
Nevada	1259
ME-2,Maine,United States	1225
Ohio	1176
Not targeted	1019
FL-26,Florida,United States	805
NE-2,Nebraska,United States	658

dtype: int64

1800 if DEBUG.

e Ad spent by Targeting area using Crosstab

+ of geo targeted areas we want to review

```
t of geo-targeted areas we want to review
```

```
orgia', 'Florida','Arizona','Iowa','Wisconsin','North Carolina',  
, 'Michigan', 'Nevada', 'ME-2,Maine,United States','Ohio',  
, 'FL-26,Florida,United States', 'NE-2,Nebraska,United States']
```

```
et_by_maga list to select top target 'geo' from Geo_Targeting_Included  
data
```

```
df_raw.make.isin(models)].copy()
```

```
a = maga_gop_drumpf_file[maga_gop_drumpf_file.Geo_Targeting_Included.isin(target)]  
a
```

Automatic saving failed. This file was updated remotely or in another tab.
[diff](#)

[Show](#)

```

ample pd.crosstab(df.make, df.body_style)
crosstab(target_by_maga.Advertiser_Name, target_by_maga.Geo_Targeting_Included)

```

Geo_Targeting_Included Arizona 26,Florida

Advertiser_Name	
CLEARPATH ACTION FUND, INC.	67
DONALD J. TRUMP FOR PRESIDENT, INC.	1442
NRCC	0
NRSC	84
REPUBLICAN NATIONAL COMMITTEE	654

2018-07-03T07:00:00Z ... male, female,

We see that the most targeted geographics areas are most closely coordinated with these 5 Advertisers promoting MAGA candidates. In fact, we could use heatmap to show how the the NRCC, NRSC, and RNC execute a complimentary ad buy and coordinate a division of labor by splitting up who spends the most in each target area, hence leveraging the superpac opportunity to coordinate because they do not have to follow the traditional PAC rules regarding coordination. Specifically NRCC doesn't invest in states like Arizona

Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)

NRCC will invest in those targeted areas instead.

```

#Visualize sns.heatmap(pd.crosstab([df.make, df.num_doors], [df.body_style, df.dri
sns.heatmap(pd.crosstab([target_by_maga.Advertiser_Name, target_by_maga.Geo_Targeti

```


File "<ipython-input-86-684e8e488d7c>",
line 2

```
sns.heatmap(pd.crosstab([target, by], maga, Adve
```

Part 2: Automation Automation Exercise

- *Extract & Store*
 - Report issuance yearly, then leverage Python.
 - import module CSV
 - Create function "SearchByAdName"
Ad_Name = input()
 - Write for loop regarding columns made to extract "Ad_Name" & 'Spend_USA' b/c we need to measure consistently and will be cleaned for standardized feature
 - Create 2nd function to search by 'Geo_Target'
 - Insert for loop row in CSV
 - (Source support:
<https://www.youtube.com/watch?v=7TOFPrOt2HE>)
- *Load*
 - Upload into Python as CSV, but also store as Excel as backup

Automatic saving failed. This file was updated remotely or in another tab.
[diff](#)

[Show](#)

- Organize and categorize age groups consistently that represent by "propensity to vote score" where we split college age (>22) and cap at Age where score significantly increases for next grouping
- Join creative_stats data to geo_spend data set on Ad_Id column to capture 'Data Range' of when ad dropped
- Join creative_stats data to weekly_spend on Ad_Id column to capture Election_Cycle

column, which would highlight how midterm differs from presidential

- *Tools for Weekly Shareable:*
 - Tableau: For Director and Exec staff
 - If Slack Group: Share KPI determined by Director/requested by Chelsea for broader PAC org consumption.
 - Propose specific age group as KPI to track nationwide
 - Estimated Time =
 - Extract: Released once a year, download, extract, and update data set= 1 week
 - Check if Uploads properly, if errors, add another week
 - Standardize Process: 1 week to report back to Director on initial data exploration to propose standardized columns needed for requested features for visualizing data points
 - Data cleaning: 2 weeks
 - Tools: If non Tableau tool requested, need 3-5 days to learn how to input features and visualize in preferred tool

Guesstimated Time: 4-6 weeks after each Google Report

Automatic saving failed. This file was updated remotely or in another tab.
[diff](#)

[Show](#)

✓ 0s completed at 6:26 PM



Automatic saving failed. This file was updated remotely or in another tab. [Show diff](#)