This dataset contains information about candidates, committees, PACs (political action committees), House & Senate campaigns & about transactions, contributions by individuals & committees & operating & independent expenditures for US elections from Jan 1, 2019 to May 2020. The original dataset was prodcured from: https://www.fec.gov/data/browse-data/?tab=bulk-data.

Analysis of North Carolina Senate race 2020: Thom Tillis & Cal Cunningham

Here I address the key question requested in prompt number three.

In [1]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import bs4
import urllib.request

# native python module
import os
import cufflinks


pd.set_option('display.max_columns', 1000)


base_dir = '.../data/data/'
dataset_name = '20192020-FEC/'
for dirname, _, filenames in os.walk(base_dir + '/' + dataset_name):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In [2]:

```
state = 'NC'
candidate1 = 'tillis, thom'
candidate2 = 'cunningham, cal'
```

In [3]:

```
import bs4
import urllib.request

def read_table(page_url):
    page = urllib.request.urlopen(page_url)
    soup = bs4.BeautifulSoup(page, 'lxml')
    table = soup.find(name='table')
    result = dict()
    for tr in table.findAll('tr'):
        tds = tr.findAll('td')

        if len(tds) >= 2:
            result[tds[0].text] = tds[1].text
    return result
```

In [4]:

```
def render_human_format(num):
    magnitude = 0
    while abs(num) >= 1000:
        magnitude += 1
        num /= 1000.0
    # add more suffixes if you need them
    return '%.2f%s' % (num, ['', 'K', 'M', 'G', 'T', 'P'][magnitude])
```

Candidate Campaign Finance summaries

In [5]:

```
all_candidates = pd.read_csv("../data/20192020-FEC/All candidates.csv")
```

```
all_candidates.head()
```

Out[5]:

	CAND_ID	CAND_NAME	CAND_ICI	PTY_CD	CAND_PTY_AFFILIATION	TTL_RECEIPTS	TRANS_FROM_AU
0	H8AK00132	SHEIN, DIMITRI	C	1	DEM	0.00	0
1	H6AK00045	YOUNG, DONALD E	I	2	REP	1012401.48	130245
2	H8AK01031	NELSON, THOMAS JOHN	C	2	REP	0.00	0
3	H8AK00140	GALVIN, ALYSE	C	3	IND	1358372.81	0
4	H0AL01097	AVERHART, JAMES	O	1	DEM	50126.74	0

In [6]:

```
def receipts_disbursements_clean_up_func(df):

    predicate = (df['TRANS_FROM_AUTH'] != 0.0) & (df['TRANS_TO_AUTH'] != 0.0)

    df.loc[predicate, 'TTL_RECEIPTS_CORRECTED'] = df['TTL_RECEIPTS'] - df['TRANS_FROM_AU
    df['TTL_RECEIPTS_CORRECTED'] = df['TTL_RECEIPTS_CORRECTED'].fillna(df['TTL_RECEIPTS'

    df.loc[predicate, 'TTL_DISB_CORRECTED'] = df['TTL_DISB'] - df['TRANS_TO_AUTH']
    df['TTL_DISB_CORRECTED'] = df['TTL_DISB_CORRECTED'].fillna(df['TTL_DISB'])

    return df

all_candidates = receipts_disbursements_clean_up_func(all_candidates)
all_candidates.head()
```

Out[6]:

	CAND_ID	CAND_NAME	CAND_ICI	PTY_CD	CAND_PTY_AFFILIATION	TTL_RECEIPTS	TRANS_FROM_AU
0	H8AK00132	SHEIN, DIMITRI	C	1	DEM	0.00	0
1	H6AK00045	YOUNG, DONALD E	I	2	REP	1012401.48	130245
2	H8AK01031	NELSON, THOMAS JOHN	C	2	REP	0.00	0
3	H8AK00140	GALVIN, ALYSE	C	3	IND	1358372.81	0
4	H0AL01097	AVERHART, JAMES	O	1	DEM	50126.74	0

In [7]:

```
def map_candidate_to_seat_type(df):

    ici_map = {'C': 'Challenger', 'I': 'Incumbent', 'O': 'Open Seat'}
    df['CAND_ICI_FULL'] = df['CAND_ICI'].map(ici_map)

    return df

all_candidates = map_candidate_to_seat_type(all_candidates)
all_candidates.head()
```

Out[7]:

	CAND_ID	CAND_NAME	CAND_ICI	PTY_CD	CAND_PTY_AFFILIATION	TTL_RECEIPTS	TRANS_FROM_AU
0	H8AK00132	SHEIN, DIMITRI	C	1	DEM	0.00	0

1	H6AK00045	YOUNG, DONALD E	I	2	REP	1012401.48	130245
2	H8AK01031	NELSON, THOMAS JOHN	C	2	REP	0.00	0
3	H8AK00140	GALVIN, ALYSE	C	3	IND	1358372.81	0
4	H0AL01097	AVERHART, JAMES	O	1	DEM	50126.74	0

In [8]:

```
party_codes = read_table("https://www.fec.gov/campaign-finance-data/party-code-descripti

def map_candidate_to_party(df):

    df['CAND_PTY_AFFILIATION_FULL'] = df['CAND_PTY_AFFILIATION'].map(party_codes)

    return df

all_candidates = map_candidate_to_party(all_candidates)
all_candidates.head()
```

Out[8]:

	CAND_ID	CAND_NAME	CAND_ICI	PTY_CD	CAND_PTY_AFFILIATION	TTL_RECEIPTS	TRANS_FROM_AU
0	H8AK00132	SHEIN, DIMITRI	C	1	DEM	0.00	0
1	H6AK00045	YOUNG, DONALD E	I	2	REP	1012401.48	130245
2	H8AK01031	NELSON, THOMAS JOHN	C	2	REP	0.00	0
3	H8AK00140	GALVIN, ALYSE	C	3	IND	1358372.81	0
4	H0AL01097	AVERHART, JAMES	O	1	DEM	50126.74	0

In [9]:

```
def correct_candidate_district_and_office(df):

    if df['CAND_OFFICE_DISTRICT'].dtype == 'int64':
        df['CAND_OFFICE_DISTRICT'] = df['CAND_OFFICE_DISTRICT'].map(lambda x: str(x).zfi
    elif df['CAND_OFFICE_DISTRICT'].dtype == 'float':
        df['CAND_OFFICE_DISTRICT'] = df['CAND_OFFICE_DISTRICT'].map(lambda x: str(int(x)
    return df

all_candidates = correct_candidate_district_and_office(all_candidates)
all_candidates.head()
```

Out[9]:

	CAND_ID	CAND_NAME	CAND_ICI	PTY_CD	CAND_PTY_AFFILIATION	TTL_RECEIPTS	TRANS_FROM_AU
0	H8AK00132	SHEIN, DIMITRI	C	1	DEM	0.00	0
1	H6AK00045	YOUNG, DONALD E	I	2	REP	1012401.48	130245
2	H8AK01031	NELSON, THOMAS JOHN	C	2	REP	0.00	0
3	H8AK00140	GALVIN, ALYSE	C	3	IND	1358372.81	0

4	H0AL01097	AVERHART, JAMES	O	1	DEM	50126.74	0

In [10]:
```
candidates = all_candidates[(all_candidates['CAND_NAME'].str.contains(f"{candidate1}", c
               (all_candidates['CAND_NAME'].str.contains(f"{candidate2}", case=False))]
candidates
```

Out[10]:

	CAND_ID	CAND_NAME	CAND_ICI	PTY_CD	CAND_PTY_AFFILIATION	TTL_RECEIPTS	TRANS_FRO
3512	S0NC00202	CUNNINGHAM, CAL	C	1	DEM	7716897.31	75
3523	S4NC00162	TILLIS, THOM R. SEN.	I	2	REP	8269271.66	115

In [11]:
```
from IPython.display import HTML

candidates_summary = candidates.groupby('CAND_NAME').agg({'TTL_RECEIPTS_CORRECTED': 'sum
                                                         'TTL_DISB_CORRECTED': 'sum', 'C
                                                         'CVG_END_DT': 'max'}).reset_ind

candidates_summary['Raised'] = '$' + candidates_summary['TTL_RECEIPTS_CORRECTED'].map(re
candidates_summary['Spent'] = '$' + candidates_summary['TTL_DISB_CORRECTED'].map(render_

candidates_summary['Cash on Hand'] = '$' + candidates_summary['COH_COP'].map(render_huma
candidates_summary['Last Report date'] = pd.to_datetime(candidates_summary['CVG_END_DT']

HTML(candidates_summary.rename(columns={'CAND_NAME': 'Candidate'})[['Candidate', 'Raised
```

Out[11]:

Candidate	Raised	Spent	Cash on Hand	Last Report date
CUNNINGHAM, CAL	$7.72M	$4.72M	$3.00M	Mar 31 2020
TILLIS, THOM R. SEN.	$7.12M	$3.80M	$6.48M	Jun 09 2020

In [12]:
```
republican = candidates[candidates['CAND_PTY_AFFILIATION'] == 'REP']['CAND_NAME'].values
democrat = candidates[candidates['CAND_PTY_AFFILIATION'] == 'DEM']['CAND_NAME'].values[0
print(republican)
democrat
```

```
TILLIS, THOM R. SEN.
```

Out[12]:
```
'CUNNINGHAM, CAL'
```

In [13]:
```
candidates['color'] = candidates['CAND_PTY_AFFILIATION'].map({'DEM': 'blue', 'REP': 'red
candidates['second_color'] = candidates['CAND_PTY_AFFILIATION'].map({'DEM': 'lightblue',
color_map = dict()
secondary_color_map = dict()
for row in candidates[['CAND_NAME', 'color', 'second_color']].values:
    color_map[row[0]] = row[1]
    secondary_color_map[f"For {row[0]}"] = row[1]
    secondary_color_map[f"Against {row[0]}"] = row[2]
print(color_map)
#candidates.head()
secondary_color_map
```

```
{'CUNNINGHAM, CAL': 'blue', 'TILLIS, THOM R. SEN.': 'red'}
```

Out[13]:
```
{'For CUNNINGHAM, CAL': 'blue',
 'Against CUNNINGHAM, CAL': 'lightblue',
 'For TILLIS, THOM R. SEN.': 'red',
 'Against TILLIS, THOM R. SEN.': 'pink'}
```

In [14]:
```
candidates_desc = read_table("https://www.fec.gov/campaign-finance-data/all-candidates-f
candidates_desc = {key.strip():val.strip() for key, val in candidates_desc.items()}
candidates_desc
```

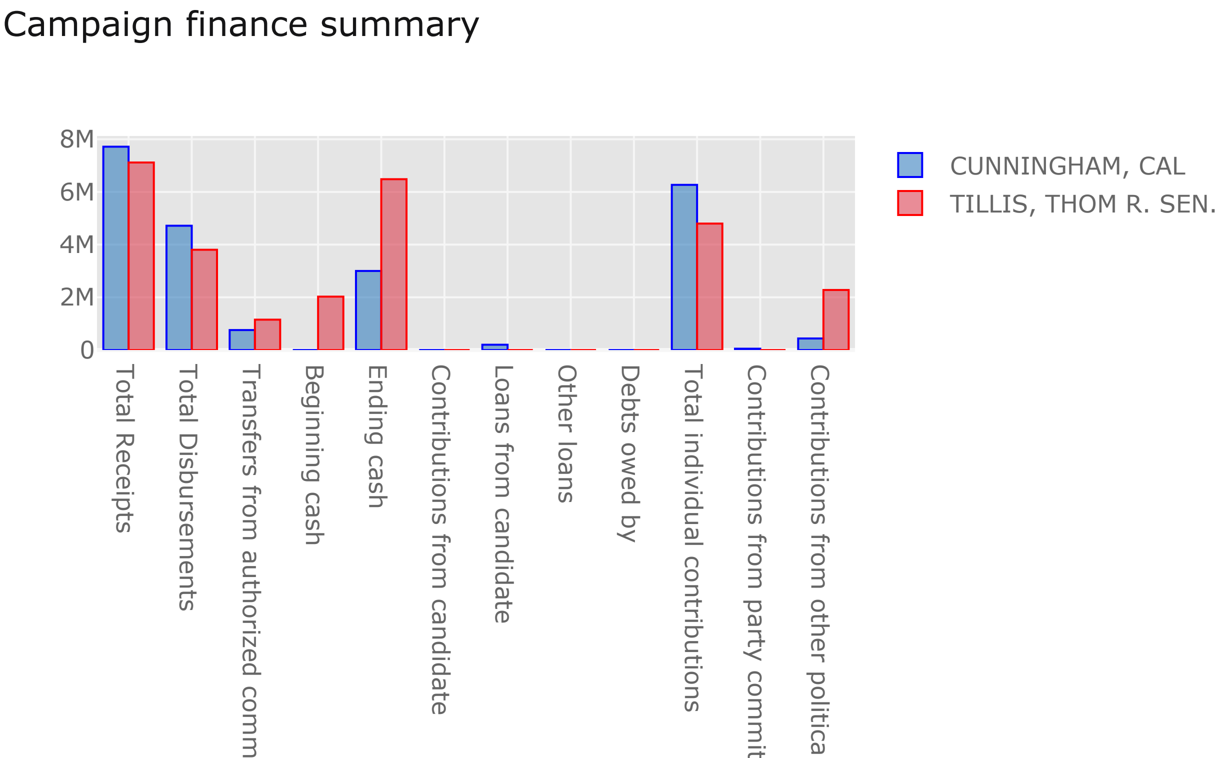
Out[14]:

```
{'Column name': 'Field name',
 'CAND_ID': 'Candidate identification',
 'CAND_NAME': 'Candidate name',
 'CAND_ICI': 'Incumbent challenger status',
 'PTY_CD': 'Party code',
 'CAND_PTY_AFFILIATION': 'Party affiliation',
 'TTL_RECEIPTS': 'Total receipts',
 'TRANS_FROM_AUTH': 'Transfers from authorized committees',
 'TTL_DISB': 'Total disbursements',
 'TRANS_TO_AUTH': 'Transfers to authorized committees',
 'COH_BOP': 'Beginning cash',
 'COH_COP': 'Ending cash',
 'CAND_CONTRIB': 'Contributions from candidate',
 'CAND_LOANS': 'Loans from candidate',
 'OTHER_LOANS': 'Other loans',
 'CAND_LOAN_REPAY': 'Candidate loan repayments',
 'OTHER_LOAN_REPAY': 'Other loan repayments',
 'DEBTS_OWED_BY': 'Debts owed by',
 'TTL_INDIV_CONTRIB': 'Total individual contributions',
 'CAND_OFFICE_ST': 'Candidate state',
 'CAND_OFFICE_DISTRICT': 'Candidate district',
 'SPEC_ELECTION': 'Special election status',
 'PRIM_ELECTION': 'Primary election status',
 'RUN_ELECTION': 'Runoff election status',
 'GEN_ELECTION': 'General election status',
 'GEN_ELECTION_PRECENT': 'General election percentage',
 'OTHER_POL_CMTE_CONTRIB': 'Contributions from other political committees',
 'POL_PTY_CONTRIB': 'Contributions from party committees',
 'CVG_END_DT': 'Coverage end date',
 'INDIV_REFUNDS': 'Refunds to individuals',
 'CMTE_REFUNDS': 'Refunds to committees'}
```

In [15]:

```
candidates_desc.update({'TTL_RECEIPTS_CORRECTED': 'Total Receipts', 'TTL_DISB_CORRECTED'
camp_finance_summary_cols = ['TTL_RECEIPTS_CORRECTED', 'TTL_DISB_CORRECTED', 'TRANS_FROM

focused_candidates = candidates.groupby('CAND_NAME').agg({x: 'sum' for x in camp_finance
total_finance = focused_candidates.set_index('CAND_NAME')[camp_finance_summary_cols].tra
total_finance = total_finance.rename(candidates_desc)
total_finance.iplot(kind='bar', title='Campaign finance summary', colors=color_map)
```



Export to plot.ly »

Contributions & expenditures from committees

In [16]:

```
contributions_from_committees = pd.read_csv(f"../data/20192020-FEC/Contributions from co
contributions_from_committees.head()
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (11,12) have mixed types.Specify dtype option on import or set low_memory=False.
```

Out[16]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
0	C00567180	T	TER	P2020	201901099143774199	24K	P
1	C00104885	A	TER	G2020	201901289144031511	24K	CC
2	C00104885	A	TER	P2022	201901289144031512	24K	CC
3	C00104885	A	TER	P2020	201901289144031511	24K	CC
4	C00688408	T	TER	G2018	201901319144305867	24E	O

In [17]:

```
def fix_zip_codes(col, df):
    if df[col].dtype == 'float':
        df[col] = df[col].map(lambda x: str(int(x)).zfill(5) if not pd.isna(x) else np.n
    elif df[col].dtype == 'int64':
        df[col] = df[col].map(lambda x: str(x).zfill(5) if not pd.isna(x) else np.nan)
    return df
contributions_from_committees = fix_zip_codes('ZIP_CODE', contributions_from_committees)
contributions_from_committees.head()
```

Out[17]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
0	C00567180	T	TER	P2020	201901099143774199	24K	P
1	C00104885	A	TER	G2020	201901289144031511	24K	CC
2	C00104885	A	TER	P2022	201901289144031512	24K	CC
3	C00104885	A	TER	P2020	201901289144031511	24K	CC

4	C00688408	T	TER	G2018	201901319144305867	24E	O

In [18]:

```python
def map_amndt_ind(df):
    df['AMNDT_IND_FULL'] = df['AMNDT_IND'].map({'N': 'New', 'A': 'Amendment', 'T': 'Term
    return df
contributions_from_committees = map_amndt_ind(contributions_from_committees)
contributions_from_committees.head()
```

Out[18]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
0	C00567180	T	TER	P2020	201901099143774199	24K	P
1	C00104885	A	TER	G2020	201901289144031511	24K	CC
2	C00104885	A	TER	P2022	201901289144031512	24K	CC
3	C00104885	A	TER	P2020	201901289144031511	24K	CC
4	C00688408	T	TER	G2018	201901319144305867	24E	O

In [19]:

```python
contributions_from_committees = contributions_from_committees[contributions_from_committ
```

In [20]:

```python
report_type_map = read_table("https://www.fec.gov/campaign-finance-data/report-type-code
def map_report_type(df):
    df['RPT_TP_FULL'] = df['RPT_TP'].map(report_type_map)
    return df
contributions_from_committees = map_report_type(contributions_from_committees)
contributions_from_committees.head()
```

Out[20]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
5	C00325324	N	M2	P2020	201902049145458880	24K	C
6	C00414425	N	M2	P	201902019145450791	24K	C
7	C00366013	N	M2	P2020	201902049145460163	24K	C
8	C00366013	N	M2	P2020	201902049145460162	24K	C
9	C00366013	N	M2	P2020	201902049145460162	24K	C

In [21]:

```python
election_type_map = {
    'P': 'Primary',
    'G': 'General',
    'O': 'Other',
    'C': 'Convention',
    'R': 'Runoff',
    'S': 'Special',
    'E': 'Recount'
}
def parse_transaction_pgi(df):
    df['ELECTION_TYPE'] = df['TRANSACTION_PGI'].astype('object').str[0].map(election_typ
    df['ELECTION_YEAR'] = df['TRANSACTION_PGI'].astype('object').str[1:].map(
        lambda x: int(x) if x and not pd.isnull(x) else np.nan).fillna(0).astype('int')
    return df
contributions_from_committees = parse_transaction_pgi(contributions_from_committees)
contributions_from_committees.head()
```

Out[21]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
5	C00325324	N	M2	P2020	201902049145458880	24K	C
6	C00414425	N	M2	P	201902019145450791	24K	C
7	C00366013	N	M2	P2020	201902049145460163	24K	C
8	C00366013	N	M2	P2020	201902049145460162	24K	C
9	C00366013	N	M2	P2020	201902049145460162	24K	C

In [22]:

```
transaction_type_map = read_table("https://www.fec.gov/campaign-finance-data/transaction

def map_transaction_type(df):

    df['TRANSACTION_TP_FULL'] = df['TRANSACTION_TP'].map(transaction_type_map)

    return df

contributions_from_committees = map_transaction_type(contributions_from_committees)
contributions_from_committees.head()
```

Out[22]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
5	C00325324	N	M2	P2020	201902049145458880	24K	C
6	C00414425	N	M2	P	201902019145450791	24K	C
7	C00366013	N	M2	P2020	201902049145460163	24K	C
8	C00366013	N	M2	P2020	201902049145460162	24K	C
9	C00366013	N	M2	P2020	201902049145460162	24K	C

In [23]:

```
entity_type_map = {
    'CAN': 'Candidate',
    'CCM': 'Candidate Committee',
    'COM': 'Committee',
    'IND': 'Individual (a person)',
    'ORG': 'Organization (not a committee and not a person)',
    'PAC': 'Political Action Committee',
    'PTY': 'Party Organization'
}
def map_entity_type(df):
    df['ENTITY_TP_FULL'] = df['ENTITY_TP'].map(entity_type_map)
    return df
contributions_from_committees = map_entity_type(contributions_from_committees)
contributions_from_committees.head()
```

Out[23]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
5	C00325324	N	M2	P2020	201902049145458880	24K	C
6	C00414425	N	M2	P	201902019145450791	24K	C
7	C00366013	N	M2	P2020	201902049145460163	24K	C
8	C00366013	N	M2	P2020	201902049145460162	24K	C
9	C00366013	N	M2	P2020	201902049145460162	24K	C

In [24]:

```
# We remove the original rows that have been amended later on
contributions_from_committees[['TRAN_ID_parent', 'TRAN_ID_child']] = contributions_from_

merged = contributions_from_committees.merge(contributions_from_committees, on=['CMTE_ID

amended_contribs = merged[(merged['AMNDT_IND_x'] == 'N') & (merged['AMNDT_IND_y'] == 'A'

# sanity check
contributions_from_committees = contributions_from_committees[~contributions_from_commit

contributions_from_committees.head()
```

Out[24]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
5	C00325324	N	M2	P2020	201902049145458880	24K	C
6	C00414425	N	M2	P	201902019145450791	24K	C
7	C00366013	N	M2	P2020	201902049145460163	24K	C
8	C00366013	N	M2	P2020	201902049145460162	24K	C
9	C00366013	N	M2	P2020	201902049145460162	24K	C

In [25]:
```
contributions_from_committees['TRANSACTION_DT'] = pd.to_datetime(
    contributions_from_committees['TRANSACTION_DT'].fillna(0).astype('int').astype('str'
    .replace('0', np.NaN).str.zfill(8), format='%m%d%Y')
contributions_from_committees.head()
```

Out[25]:

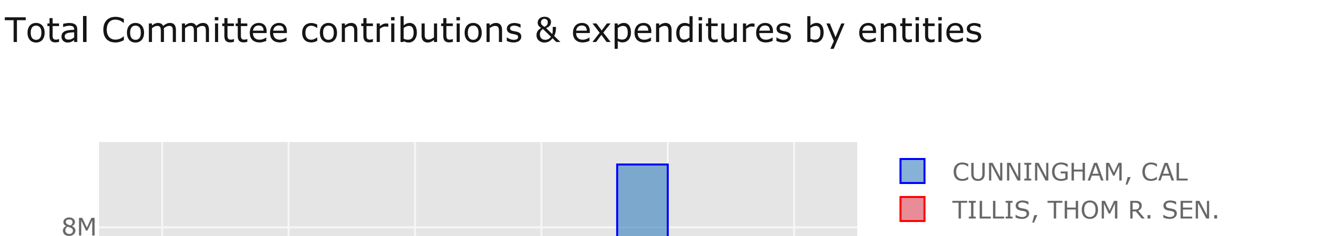
	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
5	C00325324	N	M2	P2020	201902049145458880	24K	C
6	C00414425	N	M2	P	201902019145450791	24K	C
7	C00366013	N	M2	P2020	201902049145460163	24K	C
8	C00366013	N	M2	P2020	201902049145460162	24K	C
9	C00366013	N	M2	P2020	201902049145460162	24K	C

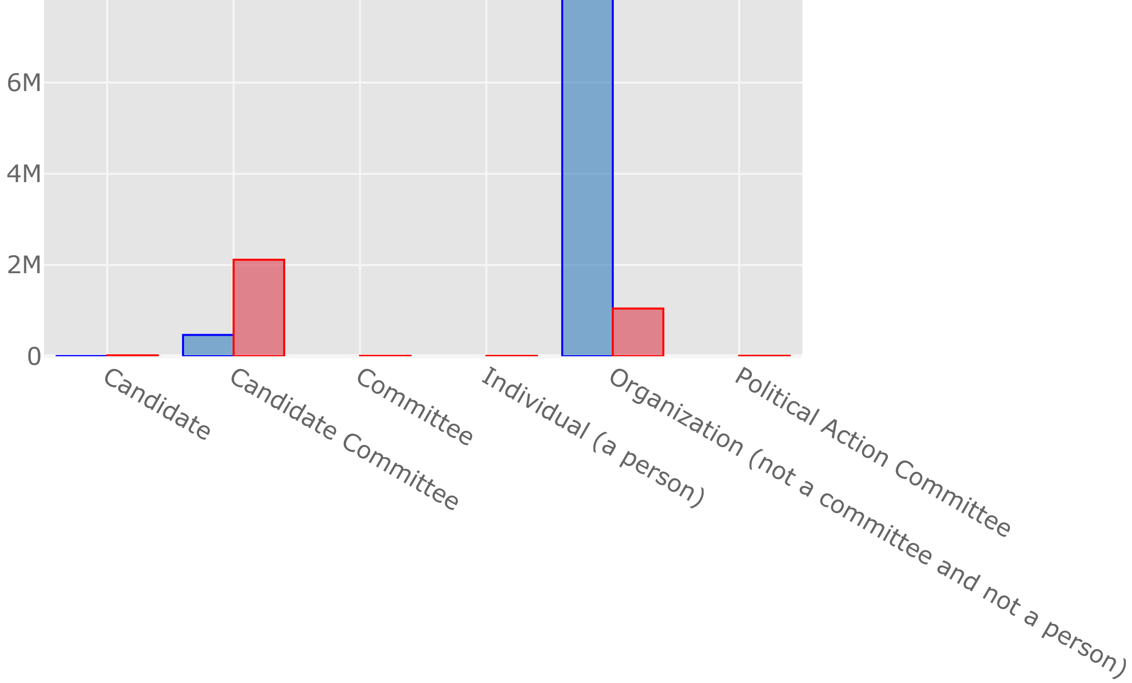
In [26]:
```
committee_contribs = contributions_from_committees.merge(candidates, on='CAND_ID')
committee_contribs.head()
```

Out[26]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
0	C00371385	N	M2	P	201902059145461441	24K	CC
1	C00551192	N	M2	P2020	201902119145496265	24K	CC
2	C00100107	N	M2	P2020	201902159145516138	24K	CC
3	C00214304	N	M2	G2020	201902199145530707	24K	CC
4	C00144774	N	M2	P2020	201902199145526887	24K	CC

In [27]:
```
by_entity = committee_contribs.groupby(['CAND_NAME', 'ENTITY_TP_FULL']).agg({'TRANSACTIO
by_entity_pivot = by_entity.pivot_table(values='TRANSACTION_AMT', index='ENTITY_TP_FULL'
by_entity_pivot.iplot(kind='bar', colors=color_map, title='Total Committee contributions
```





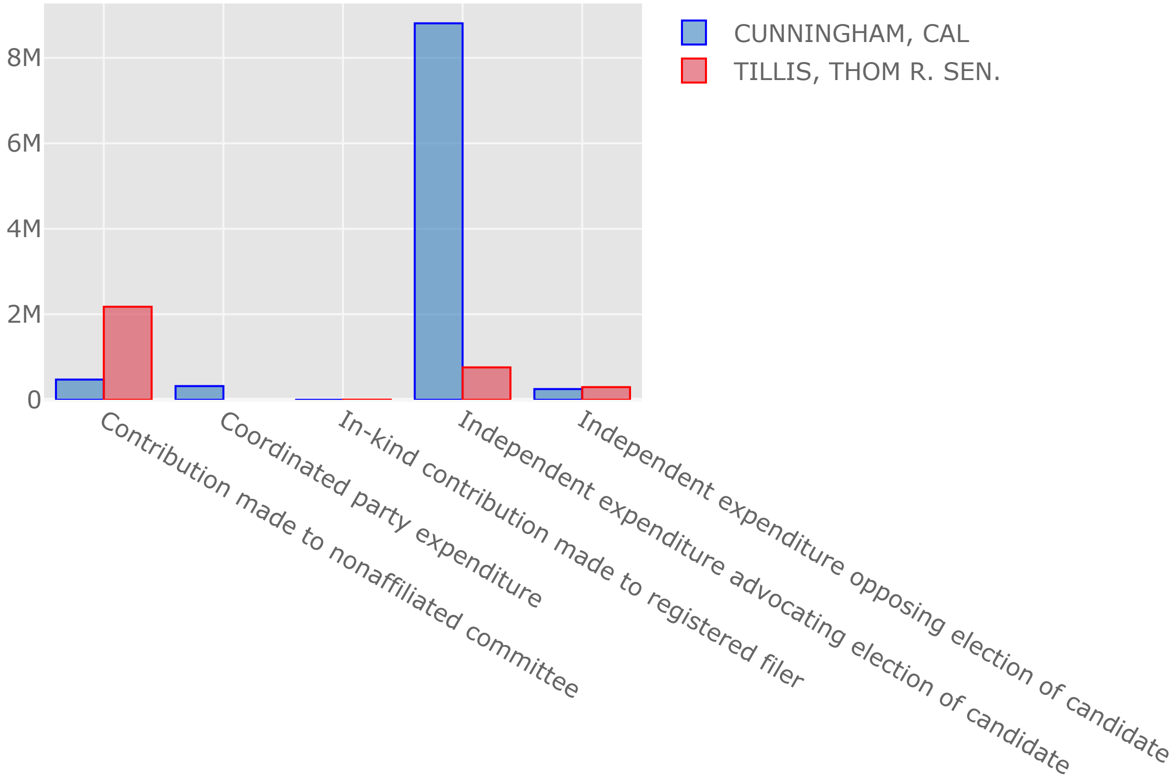
Export to plot.ly »

In [28]:
```
by_transaction = committee_contribs.groupby([
    'CAND_NAME', 'TRANSACTION_TP_FULL']).agg({'TRANSACTION_AMT': 'sum'}).reset_index()

by_transaction_pivot = by_transaction.pivot_table(
    values='TRANSACTION_AMT', index='TRANSACTION_TP_FULL', columns='CAND_NAME')

by_transaction_pivot.iplot(kind='bar',
                           colors=color_map, title='Total committee contributions & expe
```

Total committee contributions & expenditures by transaction type



Export to plot.ly »

```
In [29]: committee_master = pd.read_csv(f"../data/20192020-FEC/Committee master.csv")
         committee_master.head()
```

Out[29]:

	CMTE_ID	CMTE_NM	TRES_NM	CMTE_ST1	CMTE_ST2	CMTE_CITY	CMTE_ST	CMTE_
0	C00000059	HALLMARK CARDS PAC	SARAH MOE	2501 MCGEE	MD #500	KANSAS CITY	MO	64
1	C00000422	AMERICAN MEDICAL ASSOCIATION POLITICAL ACTION ...	WALKER, KEVIN MR.	25 MASSACHUSETTS AVE, NW	SUITE 600	WASHINGTON	DC	200017
2	C00000489	D R I V E POLITICAL FUND CHAPTER 886	JERRY SIMS JR	3528 W RENO	NaN	OKLAHOMA CITY	OK	73
3	C00000547	KANSAS MEDICAL SOCIETY POLITICAL ACTION COMMITTEE	JERRY SLAUGHTER	623 SW 10TH AVE	NaN	TOPEKA	KS	666121
4	C00000638	INDIANA STATE MEDICAL ASSOCIATION POLITICAL AC...	ACHENBACH, GRANT MR.	322 CANAL WALK, CANAL LEVEL	NaN	INDIANAPOLIS	IN	46

```
In [30]: committee_contribs = committee_contribs.merge(committee_master, on='CMTE_ID', how='left'
         committee_contribs.head()
```

Out[30]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
0	C00371385	N	M2	P	201902059145461441	24K	CC
1	C00551192	N	M2	P2020	201902119145496265	24K	CC
2	C00100107	N	M2	P2020	201902159145516138	24K	CC
3	C00214304	N	M2	G2020	201902199145530707	24K	CC
4	C00144774	N	M2	P2020	201902199145526887	24K	CC

Contributions from committees

In [31]:
```
contribs = committee_contribs[committee_contribs['TRANSACTION_TP'].isin(['24K', '24Z'])]
contribs.head()
```

Out[31]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
0	C00371385	N	M2	P	201902059145461441	24K	CC
1	C00551192	N	M2	P2020	201902119145496265	24K	CC
2	C00100107	N	M2	P2020	201902159145516138	24K	CC
3	C00214304	N	M2	G2020	201902199145530707	24K	CC
4	C00144774	N	M2	P2020	201902199145526887	24K	CC

Here are the committees with the largest contributions

In [32]:
```
largest_contribs = contribs.groupby('CMTE_NM').agg(
    {'TRANSACTION_AMT': 'sum'})['TRANSACTION_AMT'].nlargest(10).reset_index()

largest_contribs['TRANSACTION_AMT'] = largest_contribs['TRANSACTION_AMT'].map(lambda x:

largest_contribs.rename({'CMTE_NM': 'COMMITTEE_NAME'}, axis=1)
```

Out[32]:

	COMMITTEE_NAME	TRANSACTION_AMT
0	DSCC	49.60K
1	NRSC	44.60K
2	JSTREETPAC	34.93K
3	ABBOTT LABORATORIES EMPLOYEE POLITICAL ACTION ...	10.00K
4	ALL FOR OUR COUNTRY LEADERSHIP PAC	10.00K
5	ALLERGAN, INC. POLITICAL ACTION COMMITTEE	10.00K
6	AMERICAN BANKERS ASSOCIATION PAC (BANKPAC)	10.00K
7	ANADARKO PETROLEUM CORPORATION POLITICAL ACTIO...	10.00K
8	ARKANSAS FOR LEADERSHIP POLITICAL ACTION COMMI...	10.00K
9	ASSOCIATED BUILDERS AND CONTRACTORS, INC. POLI...	10.00K

In [33]:
```
from IPython.display import display
```

```
for cand_name in candidates['CAND_NAME'].unique():
    print("-"*60)
    print(f"Committees with most contributions for {cand_name}")
    print("-"*60)
    df = contribs[(contribs['CAND_NAME'] == cand_name)]
    result = pd.DataFrame(df[['CMTE_NM', 'TRANSACTION_AMT']].groupby(['CMTE_NM']).agg({'
    result['TRANSACTION_AMT'] = result['TRANSACTION_AMT'].map(lambda x: render_human_for
    display(result)
```

```
------------------------------------------------------------
Committees with most contributions for CUNNINGHAM, CAL
------------------------------------------------------------
```

	COMMITTEE_NAME	TRANSACTION_AMT
0	DSCC	49.60K
1	JSTREETPAC	34.93K
2	ALL FOR OUR COUNTRY LEADERSHIP PAC	10.00K
3	COMMON GROUND PAC	10.00K
4	COMMUNICATIONS WORKERS OF AMERICA-COPE POLITIC...	10.00K
5	FIRST STATE PAC	10.00K
6	FORWARD TOGETHER PAC	10.00K
7	GETTING STUFF DONE PAC (GSD-PAC)	10.00K
8	HAWAII PAC	10.00K
9	HOOPS PAC	10.00K

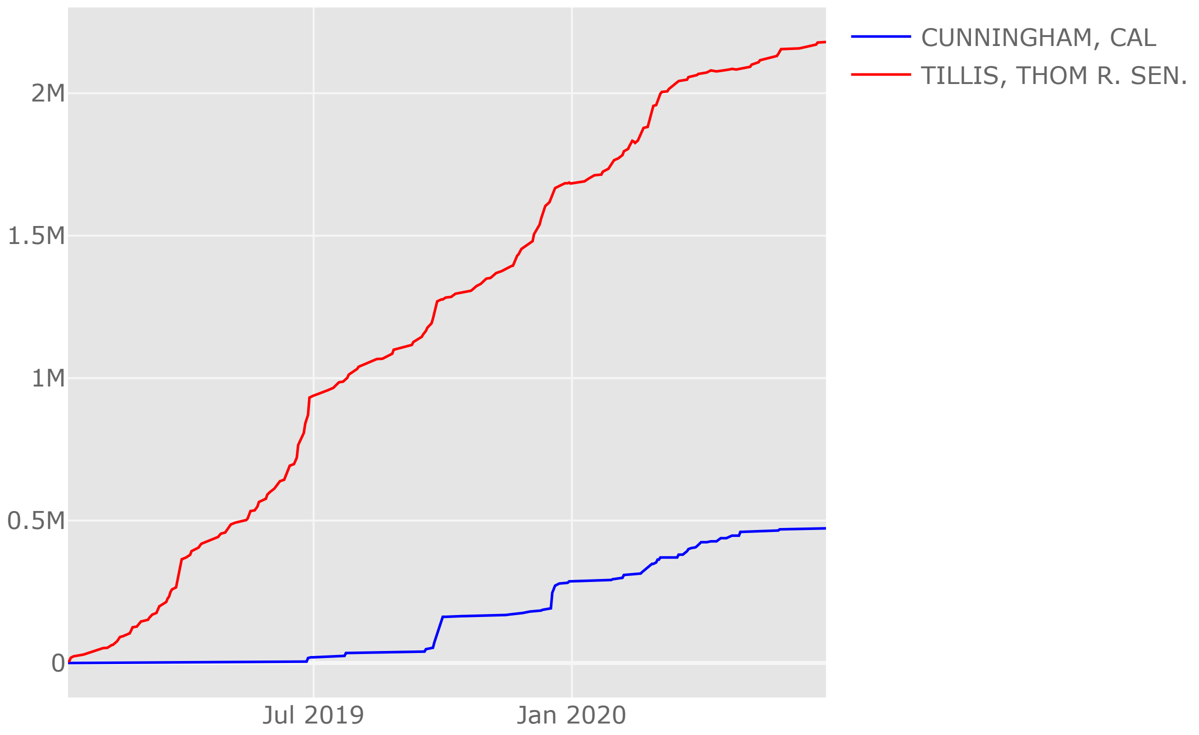
```
------------------------------------------------------------
Committees with most contributions for TILLIS, THOM R. SEN.
------------------------------------------------------------
```

	COMMITTEE_NAME	TRANSACTION_AMT
0	NRSC	44.60K
1	ABBOTT LABORATORIES EMPLOYEE POLITICAL ACTION ...	10.00K
2	ALLERGAN, INC. POLITICAL ACTION COMMITTEE	10.00K
3	AMERICAN BANKERS ASSOCIATION PAC (BANKPAC)	10.00K
4	ANADARKO PETROLEUM CORPORATION POLITICAL ACTIO...	10.00K
5	ARKANSAS FOR LEADERSHIP POLITICAL ACTION COMMI...	10.00K
6	ASSOCIATED BUILDERS AND CONTRACTORS, INC. POLI...	10.00K
7	ASSOCIATION OF KENTUCKY FRIED CHICKEN FRANCHIS...	10.00K
8	BANK POLICY INSTITUTE PAC	10.00K
9	BBVA USA BANCSHARES, INC. PAC	10.00K

In [34]:

```
ctrbs_by_dt = contribs.groupby(['TRANSACTION_DT',
                                'CAND_NAME']).agg({'TRANSACTION_AMT': 'sum'}).reset_inde
ctrbs_by_dt = ctrbs_by_dt.pivot_table(index='TRANSACTION_DT',
                                      columns=['CAND_NAME'], fill_value=0.0).droplevel(l
ctrbs_by_dt.cumsum().iplot(title='Total cumulative Committee Contributions for candidate
                           colors=color_map)
```

Total cumulative Committee Contributions for candidates over time



Export to plot.ly »

Committee expenditures for candidates

In [35]:

```
committee_contribs['for_against'] = committee_contribs['TRANSACTION_TP'].map({
    '24A': 'Against', '24E': 'For'})

expenditures = committee_contribs[~committee_contribs['for_against'].isnull()]
expenditures.head()
```

Out[35]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTIT
494	C00571703	N	YE	P2020	202001319185087439	24E	
495	C00571703	N	YE	P2020	202001319185087444	24E	
496	C00571703	N	YE	P2020	202001319185087444	24E	
497	C00571703	N	YE	P2020	202001319185087445	24E	
539	C00514125	N	YE	P2020	202001299182384741	24A	

Here are the committees with the largest expenditures

In [36]:

```
largest_expenditures = expenditures.groupby('CMTE_NM').agg({'TRANSACTION_AMT': 'sum'})['
largest_expenditures['TRANSACTION_AMT'] = largest_expenditures['TRANSACTION_AMT'].map(la
largest_expenditures.rename({'CMTE_NM': 'COMMITTEE_NAME'}, axis=1)
```

Out[36]:

	COMMITTEE_NAME	TRANSACTION_AMT
0	CAROLINA BLUE	4.52M
1	VOTEVETS	4.29M
2	AMERICANS FOR PROSPERITY ACTION, INC.(AFP ACTION)	675.66K
3	THE AMERICAN FOUNDATIONS COMMITTEE	254.00K
4	FAITH AND POWER PAC	250.01K
5	SENATE LEADERSHIP FUND	66.99K
6	INDIVISIBLE ACTION	22.50K
7	THE LINCOLN PROJECT	11.78K
8	NATIONAL RIFLE ASSOCIATION OF AMERICA POLITICA...	10.07K
9	HEALTH JUSTICE FOR ALL	6.52K

In [37]:

```
from IPython.display import display
for cand_name in candidates['CAND_NAME'].unique():
    for for_against in ['For', 'Against']:
        print("-"*60)
        print(f"Committees with most expenditures {for_against} {cand_name}")
        print("-"*60)
        df = expenditures[(expenditures['for_against'] == for_against) & (expenditures['
        result = pd.DataFrame(df[['CMTE_NM', 'TRANSACTION_AMT']].groupby(['CMTE_NM']).ag
        result['TRANSACTION_AMT'] = result['TRANSACTION_AMT'].map(lambda x: render_human
        display(result)
```

```
------------------------------------------------------------
Committees with most expenditures For CUNNINGHAM, CAL
------------------------------------------------------------
```

	COMMITTEE_NAME	TRANSACTION_AMT
0	CAROLINA BLUE	4.52M
1	VOTEVETS	4.29M
2	PAGE COMMUNICATIONS L.L.C.	3.26K
3	PLANNED PARENTHOOD VOTES	579.00
4	SIERRA CLUB INDEPENDENT ACTION	10.00

```
------------------------------------------------------------
Committees with most expenditures Against CUNNINGHAM, CAL
------------------------------------------------------------
```

	COMMITTEE_NAME	TRANSACTION_AMT
0	FAITH AND POWER PAC	250.01K

```
------------------------------------------------------------
Committees with most expenditures For TILLIS, THOM R. SEN.
------------------------------------------------------------
```

	COMMITTEE_NAME	TRANSACTION_AMT
0	AMERICANS FOR PROSPERITY ACTION, INC.(AFP ACTION)	675.66K
1	SENATE LEADERSHIP FUND	66.99K

2	NATIONAL RIFLE ASSOCIATION OF AMERICA POLITICA...	10.07K
3	WOMEN SPEAK OUT PAC	4.54K

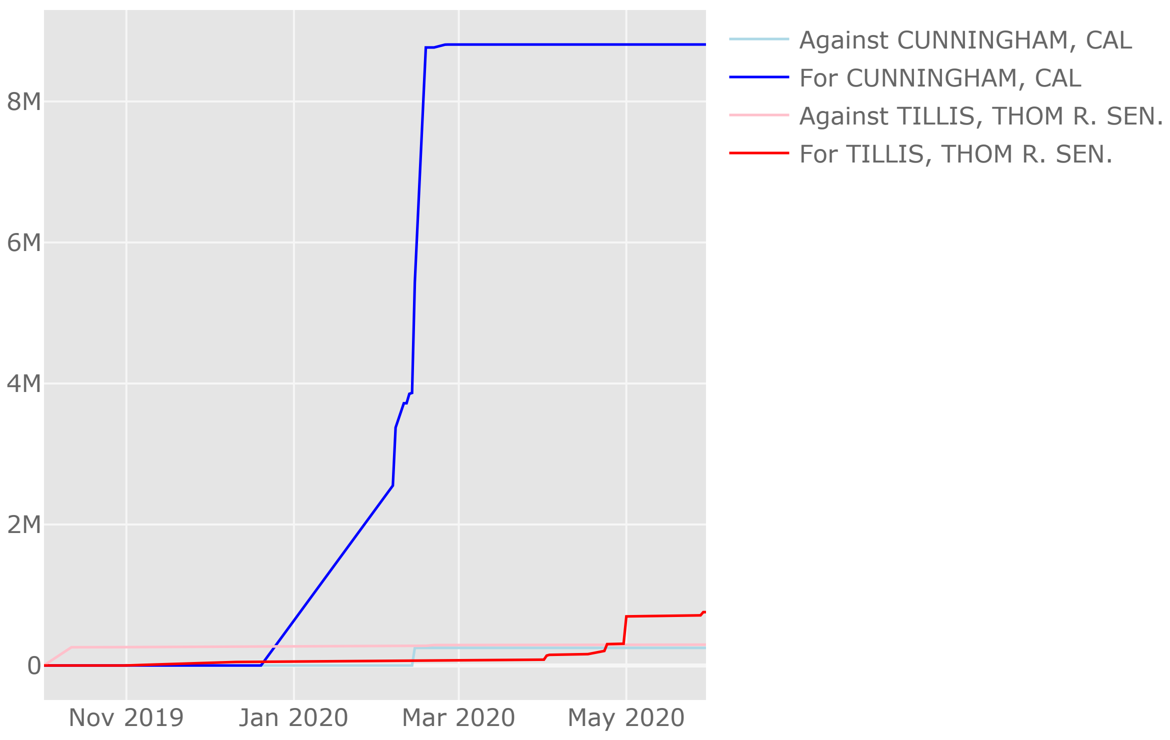
```
------------------------------------------------------------
Committees with most expenditures Against TILLIS, THOM R. SEN.
------------------------------------------------------------
```

	COMMITTEE_NAME	TRANSACTION_AMT
0	THE AMERICAN FOUNDATIONS COMMITTEE	254.00K
1	INDIVISIBLE ACTION	22.50K
2	THE LINCOLN PROJECT	11.78K
3	HEALTH JUSTICE FOR ALL	6.52K
4	PLANNED PARENTHOOD VOTES	866.00

In [38]:

```
exp_by_dt = expenditures.groupby(['TRANSACTION_DT', 'CAND_NAME', 'for_against']).agg({'T
exp_by_dt = exp_by_dt.pivot_table(index='TRANSACTION_DT', columns=['CAND_NAME', 'for_aga
exp_by_dt.columns = exp_by_dt.columns.map(lambda x: x[1] + " " + x[0])
exp_by_dt.cumsum().iplot(title='Total cumulative Expenditures for or against candidates
```

Total cumulative Expenditures for or against candidates over time



Export to plot.ly »

Contributions from individuals

In [39]:

```
# First, we determine which committees are for or against a candidate so that we can buc
committee_contribs['for_against'] = committee_contribs['for_against'].fillna('For')
committee_contribs['CAND_NAME'] = np.where(((committee_contribs['for_against'] == 'Again
                                            (committee_contribs['CAND_NAME'] == democrat
                                            ((committee_contribs['for_against'] == 'For'
                                            (committee_contribs['CAND_NAME'] == republic
                                           republican, democrat)
committee_contribs.head()
```

Out[39]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
0	C00371385	N	M2	P	201902059145461441	24K	CC
1	C00551192	N	M2	P2020	201902119145496265	24K	CC
2	C00100107	N	M2	P2020	201902159145516138	24K	CC
3	C00214304	N	M2	G2020	201902199145530707	24K	CC
4	C00144774	N	M2	P2020	201902199145526887	24K	CC

In [40]:

```
cmte_cand_mapping = committee_contribs.groupby(['CMTE_ID', 'CAND_NAME']).agg({'TRANSACTI
cmte_cand_mapping = cmte_cand_mapping.pivot_table(index='CMTE_ID', columns='CAND_NAME').
cmte_cand_mapping = cmte_cand_mapping.loc[~(cmte_cand_mapping[democrat] == cmte_cand_map
cmte_cand_mapping['CAND_NAME'] = np.where(cmte_cand_mapping[democrat] > cmte_cand_mappin
cmte_cand_mapping.head()
```

Out[40]:

CAND_NAME	CMTE_ID	CUNNINGHAM, CAL	TILLIS, THOM R. SEN.	CAND_NAME
0	C00000422	0.0	2500.0	TILLIS, THOM R. SEN.
1	C00000729	0.0	3000.0	TILLIS, THOM R. SEN.
2	C00000885	2500.0	0.0	CUNNINGHAM, CAL
3	C00000901	0.0	5000.0	TILLIS, THOM R. SEN.
4	C00001016	5000.0	0.0	CUNNINGHAM, CAL

In [41]:

```
s = f"../data/20192020-FEC/Contributions by individuals.csv"
cmd_result = ! wc -l "{s}"
contribs_cnt = int(cmd_result[0].split()[0])
contribs_cnt
```

Out[41]:

```
26170487
```

In [42]:

```
# Iteration 42 has some bad data that needs to be fixed manually as below
def handle_iteration_42(df):
    bad_row = df[df['TRANSACTION_DT'] == 'SAN DIEGO']
    bad_rows = bad_row['OCCUPATION'].str.split('\n').values[0]
    other_rows = []
    for i, row in enumerate(bad_rows):
        if i==0:
            first_row_last_part = row.split('|')
        elif i == len(bad_rows)-1:
            last_row_first_part = row.split('|')
        else:
            other_rows.append(row.split('|'))
    all_dfs = [df]
    first_row = list(df.loc[bad_row.index].iloc[:, 0:12].copy().values[0]) + first_row_l
    all_dfs.append(pd.DataFrame([first_row], columns=df.columns))
    all_dfs.append(pd.DataFrame(other_rows, columns=df.columns))
    all_dfs.append(pd.DataFrame([last_row_first_part + list(df.loc[bad_row.index].iloc[:
                                   + [np.NaN, np.NaN, np.NaN, np.NaN, np.NaN]], columns=
    df = pd.concat(all_dfs, axis=0)
    df = df.drop(index=bad_row.index).reset_index()
    return df
```

In [44]:

```
import gc
chunksize = 100000
contribs_by_indivs = []
cols = None
for i in range(contribs_cnt//chunksize):
    print(f"Processing chunk {i}")
    df = pd.read_csv(f"../data/20192020-FEC/Contributions by individuals.csv", skiprows=
    if i == 0:
        cols = df.columns
    else:
        df.columns = cols
    if i == 42:
        df = handle_iteration_42(df)
    interesting_ones = df[df['CMTE_ID'].isin(cmte_cand_mapping['CMTE_ID'])]
    if len(interesting_ones) > 0:
        contribs_by_indivs.append(interesting_ones)
#len(contribs_by_indivs)
```

```
Processing chunk 0
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (10,18) have mixed types.Specify dtype option on import or set low_memory=False.
```

```
Processing chunk 1
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (10) have mixed types.Specify dtype option on import or set low_memory=False.
```

```
Processing chunk 2
Processing chunk 3
Processing chunk 4
Processing chunk 5
Processing chunk 6
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (5,10,15,19) have mixed types.Specify dtype option on import or set low_memory=F
alse.
```

```
Processing chunk 7
Processing chunk 8
Processing chunk 9
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (5,10,15,18,19) have mixed types.Specify dtype option on import or set low_memor
```

```
y=False.
```

```
Processing chunk 10
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (10,18,19) have mixed types.Specify dtype option on import or set low_memory=Fal
se.
```

```
Processing chunk 11
Processing chunk 12
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (5,10,15,18) have mixed types.Specify dtype option on import or set low_memory=F
alse.
```

```
Processing chunk 13
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (16,18) have mixed types.Specify dtype option on import or set low_memory=False.
```

```
Processing chunk 14
Processing chunk 15
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (10,15,18) have mixed types.Specify dtype option on import or set low_memory=Fal
se.
```

```
Processing chunk 16
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (10,15) have mixed types.Specify dtype option on import or set low_memory=False.
```

```
Processing chunk 17
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (10,16,18) have mixed types.Specify dtype option on import or set low_memory=Fal
se.
```

```
Processing chunk 18
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (10,16,18,19) have mixed types.Specify dtype option on import or set low_memory=
False.
```

```
Processing chunk 19
Processing chunk 20
Processing chunk 21
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (3,10,18) have mixed types.Specify dtype option on import or set low_memory=Fals
e.
```

```
Processing chunk 22
Processing chunk 23
Processing chunk 24
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (3,5,10,15,18) have mixed types.Specify dtype option on import or set low_memory
=False.

Processing chunk 25
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (5,15,18,19) have mixed types.Specify dtype option on import or set low_memory=F
alse.

Processing chunk 26
Processing chunk 27
Processing chunk 28
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (16) have mixed types.Specify dtype option on import or set low_memory=False.

Processing chunk 29
Processing chunk 30
Processing chunk 31
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (18) have mixed types.Specify dtype option on import or set low_memory=False.

Processing chunk 32
Processing chunk 33
Processing chunk 34
Processing chunk 35
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (3,18) have mixed types.Specify dtype option on import or set low_memory=False.

Processing chunk 36
Processing chunk 37
Processing chunk 38
Processing chunk 39
Processing chunk 40
Processing chunk 41
Processing chunk 42
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (13,14,17,20) have mixed types.Specify dtype option on import or set low_memory=
False.

Processing chunk 43
Processing chunk 44
Processing chunk 45
Processing chunk 46
Processing chunk 47
Processing chunk 48
Processing chunk 49
Processing chunk 50
Processing chunk 51
Processing chunk 52
Processing chunk 53
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:
```

```
Columns (10,15,16) have mixed types.Specify dtype option on import or set low_memory=Fal
se.
```

```
Processing chunk 54
Processing chunk 55
Processing chunk 56
Processing chunk 57
Processing chunk 58
Processing chunk 59
Processing chunk 60
Processing chunk 61
Processing chunk 62
Processing chunk 63
Processing chunk 64
Processing chunk 65
Processing chunk 66
Processing chunk 67
Processing chunk 68
Processing chunk 69
Processing chunk 70
Processing chunk 71
Processing chunk 72
Processing chunk 73
Processing chunk 74
Processing chunk 75
Processing chunk 76
Processing chunk 77
Processing chunk 78
Processing chunk 79
Processing chunk 80
Processing chunk 81
Processing chunk 82
Processing chunk 83
Processing chunk 84
Processing chunk 85
Processing chunk 86
Processing chunk 87
Processing chunk 88
Processing chunk 89
Processing chunk 90
Processing chunk 91
Processing chunk 92
Processing chunk 93
Processing chunk 94
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (10,16) have mixed types.Specify dtype option on import or set low_memory=False.
```

```
Processing chunk 95
Processing chunk 96
Processing chunk 97
Processing chunk 98
Processing chunk 99
Processing chunk 100
Processing chunk 101
Processing chunk 102
Processing chunk 103
Processing chunk 104
Processing chunk 105
Processing chunk 106
Processing chunk 107
Processing chunk 108
Processing chunk 109
Processing chunk 110
Processing chunk 111
Processing chunk 112
Processing chunk 113
Processing chunk 114
Processing chunk 115
Processing chunk 116
Processing chunk 117
Processing chunk 118
Processing chunk 119
Processing chunk 120
Processing chunk 121
Processing chunk 122
Processing chunk 123
Processing chunk 124
Processing chunk 125
Processing chunk 126
Processing chunk 127
Processing chunk 128
Processing chunk 129
Processing chunk 130
Processing chunk 131
Processing chunk 132
Processing chunk 133
Processing chunk 134
Processing chunk 135
Processing chunk 136
Processing chunk 137
Processing chunk 138
Processing chunk 139
Processing chunk 140
Processing chunk 141
Processing chunk 142
Processing chunk 143
Processing chunk 144
Processing chunk 145
Processing chunk 146
Processing chunk 147
Processing chunk 148
Processing chunk 149
Processing chunk 150
Processing chunk 151
Processing chunk 152
Processing chunk 153
Processing chunk 154
Processing chunk 155
Processing chunk 156
Processing chunk 157
Processing chunk 158
Processing chunk 159
Processing chunk 160
Processing chunk 161
Processing chunk 162
Processing chunk 163
Processing chunk 164
Processing chunk 165
Processing chunk 166
Processing chunk 167
Processing chunk 168
Processing chunk 169
Processing chunk 170
Processing chunk 171
Processing chunk 172
Processing chunk 173
Processing chunk 174
Processing chunk 175
Processing chunk 176
Processing chunk 177
Processing chunk 178
Processing chunk 179
Processing chunk 180
Processing chunk 181
Processing chunk 182
Processing chunk 183
Processing chunk 184
Processing chunk 185
Processing chunk 186
Processing chunk 187
Processing chunk 188
Processing chunk 189
Processing chunk 190
Processing chunk 191
Processing chunk 192
Processing chunk 193
Processing chunk 194
Processing chunk 195
Processing chunk 196
Processing chunk 197
Processing chunk 198
Processing chunk 199
Processing chunk 200
Processing chunk 201
Processing chunk 202
Processing chunk 203
Processing chunk 204
Processing chunk 205
Processing chunk 206
Processing chunk 207
Processing chunk 208
Processing chunk 209
Processing chunk 210
Processing chunk 211
Processing chunk 212
Processing chunk 213
Processing chunk 214
Processing chunk 215
Processing chunk 216
Processing chunk 217
Processing chunk 218
Processing chunk 219
Processing chunk 220
Processing chunk 221
Processing chunk 222
Processing chunk 223
Processing chunk 224
Processing chunk 225
Processing chunk 226
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (18,19) have mixed types.Specify dtype option on import or set low_memory=False.
```

```
Processing chunk 227
Processing chunk 228
Processing chunk 229
Processing chunk 230
Processing chunk 231
Processing chunk 232
Processing chunk 233
Processing chunk 234
Processing chunk 235
Processing chunk 236
Processing chunk 237
Processing chunk 238
Processing chunk 239
Processing chunk 240
Processing chunk 241
Processing chunk 242
Processing chunk 243
Processing chunk 244
Processing chunk 245
Processing chunk 246
Processing chunk 247
Processing chunk 248
Processing chunk 249
Processing chunk 250
Processing chunk 251
Processing chunk 252
Processing chunk 253
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (3) have mixed types.Specify dtype option on import or set low_memory=False.
```

```
Processing chunk 254
Processing chunk 255
Processing chunk 256
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (11,12) have mixed types.Specify dtype option on import or set low_memory=False.
```

```
Processing chunk 257
```

```
/Users/sahluwalia/acre/lib/python3.7/site-packages/IPython/core/interactiveshell.py:325
7: DtypeWarning:

Columns (3,10,11,12,19) have mixed types.Specify dtype option on import or set low_memor
y=False.
```

```
Processing chunk 258
Processing chunk 259
Processing chunk 260
```

In [45]:

```
contribs_by_indivs = pd.concat(contribs_by_indivs)
contribs_by_indivs.head()
```

Out[45]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY
15	C00428110	N	M2	P	201902049145460549	15	
16	C00428110	N	M2	P	201902049145460548	15	
17	C00428110	N	M2	P	201902049145460548	15	
18	C00428110	N	M2	P	201902049145460548	15	
19	C00428110	N	M2	P	201902049145460549	15	

In [46]:

```
contribs_by_indivs = contribs_by_indivs[(~contribs_by_indivs['TRANSACTION_DT'].isnull())
contribs_by_indivs['TRANSACTION_DT'] = pd.to_datetime(
    contribs_by_indivs['TRANSACTION_DT'].fillna(0).astype('int').astype('str')
    .replace('0', np.NaN).str.zfill(8), format='%m%d%Y')
contribs_by_indivs.head()
```

Out[46]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY
15	C00428110	N	M2	P	201902049145460549	15	
16	C00428110	N	M2	P	201902049145460548	15	
17	C00428110	N	M2	P	201902049145460548	15	
18	C00428110	N	M2	P	201902049145460548	15	
19	C00428110	N	M2	P	201902049145460549	15	

In [47]:

```
contribs_by_indivs = contribs_by_indivs[(contribs_by_indivs['TRANSACTION_DT'] >= '2018-0
contribs_by_indivs.head()
```

Out[47]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY
15	C00428110	N	M2	P	201902049145460549	15	
16	C00428110	N	M2	P	201902049145460548	15	
17	C00428110	N	M2	P	201902049145460548	15	
18	C00428110	N	M2	P	201902049145460548	15	
19	C00428110	N	M2	P	201902049145460549	15	

In [48]:

```
contribs_by_indivs = contribs_by_indivs.merge(cmte_cand_mapping[['CMTE_ID', 'CAND_NAME']
#contribs_by_indivs = contribs_by_indivs.merge(candidates[['CAND_ID', 'CAND_NAME']], on=
contribs_by_indivs.head()
```

Out[48]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_
0	C00428110	N	M2	P	201902049145460549	15	I
1	C00428110	N	M2	P	201902049145460548	15	I
2	C00428110	N	M2	P	201902049145460548	15	I

3	C00428110	N	M2	P	201902049145460548	15	II
4	C00428110	N	M2	P	201902049145460549	15	II

In [49]:

```python
from IPython.display import display
for cand_name in candidates['CAND_NAME'].unique():
    print("-"*60)
    print(f"Most individual contributions for {cand_name}")
    print("-"*60)
    df = contribs_by_indivs[(contribs_by_indivs['CAND_NAME'] == cand_name)]
    result = pd.DataFrame(df[['NAME', 'TRANSACTION_AMT']].groupby(['NAME']).agg({'TRANSA
    result['TRANSACTION_AMT'] = result['TRANSACTION_AMT'].map(lambda x: render_human_for
    display(result)
```

```
------------------------------------------------------------
Most individual contributions for CUNNINGHAM, CAL
------------------------------------------------------------
```

	NAME	TRANSACTION_AMT
0	SMP	7.85M
1	ROSENTHAL, RICHARD	2.51M
2	MANDEL, SUSAN	1.58M
3	SIMON, DEBORAH	1.11M
4	DELANEY, MARY QUINN	1.01M
5	INDIVISIBLE PROJECT, INC.	1.00M
6	CHC BOLD PAC	803.64K
7	SIMON, DEBORAH J	600.00K
8	SOROS, GEORGE	512.00K
9	O'BRIEN, PAULA	505.00K
10	SIERRA CLUB	500.00K
11	SOROS COLOMBEL, ANDREA	500.00K
12	SIMON, DEBORAH J.	497.00K
13	SOSNICK, AARON	497.00K
14	EYCHANER, FRED	493.60K
15	SUSSMAN, S. DONALD	488.60K
16	KLARMAN, SETH	480.60K
17	CAPPELL, AUDREY	412.80K
18	CAPPELL, JACOB	412.80K
19	MCGRATH, KATHLEEN	405.50K

```
------------------------------------------------------------
Most individual contributions for TILLIS, THOM R. SEN.
------------------------------------------------------------
```

	NAME	TRANSACTION_AMT

0	SCHWARZMAN, STEPHEN A.	10.01M
1	MELLON, TIMOTHY	10.00M
2	KOCH INDUSTRIES INC.	7.00M
3	FREEDOM PARTNERS ACTION FUND INC.	6.48M
4	MARCUS, BERNARD	4.00M
5	SCHWAB, CHARLES R.	3.50M
6	DUNCAN, JAN	2.99M
7	SENATE LEADERSHIP FUND	2.95M
8	STEPHENS, WARREN A.	2.75M
9	SCHWAB, HELEN O'NEILL	2.50M
10	SINGER, PAUL ELLIOTT	2.00M
11	CHEVRON CORPORATION	1.62M
12	GOPAC INC	1.50M
13	DUCHOSSOIS, CRAIG J.	1.30M
14	REYES, J. CHRISTOPHER	1.25M
15	REYES, M. JUDE	1.25M
16	BERGAN, MARY ALICE	1.14M
17	CAMERON, RONALD	1.00M
18	CL MACHINERY COMPANY	1.00M
19	CONOCOPHILLIPS ANS MARKETING COMPANY	1.00M

In [50]:

```python
from IPython.display import display
indiv_contribs_cmte = contribs_by_indivs.merge(committee_master[['CMTE_ID', 'CMTE_NM']],
for cand_name in candidates['CAND_NAME'].unique():
    print("-"*80)
    print(f"Committees that received the most individual contributions for {cand_name}")
    print("-"*80)
    df = indiv_contribs_cmte[(indiv_contribs_cmte['CAND_NAME'] == cand_name)]
    result = pd.DataFrame(df[['CMTE_NM', 'TRANSACTION_AMT']].groupby(['CMTE_NM']).agg({
        'TRANSACTION_AMT': 'sum'})['TRANSACTION_AMT'].nlargest(10)).reset_index()
    result['TRANSACTION_AMT'] = result['TRANSACTION_AMT'].map(lambda x: render_human_for
    display(result)
```

```
--------------------------------------------------------------------------------
Committees that received the most individual contributions for CUNNINGHAM, CAL
--------------------------------------------------------------------------------
```

	CMTE_NM	TRANSACTION_AMT
0	DSCC	58.86M
1	VOTEVETS	12.93M
2	PLANNED PARENTHOOD VOTES	10.02M
3	END CITIZENS UNITED	8.26M
4	MCCREADY FOR CONGRESS	4.22M
5	JSTREETPAC	3.05M
6	INTERNATIONAL ASSOCIATION OF FIREFIGHTERS INTE...	2.21M
7	CHC BOLD PAC	2.18M

	CMTE_NM	TRANSACTION_AMT
8	INDIVISIBLE ACTION	2.17M
9	LEAGUE OF CONSERVATION VOTERS ACTION FUND	1.66M

```
--------------------------------------------------------------------------------
Committees that received the most individual contributions for TILLIS, THOM R. SEN.
--------------------------------------------------------------------------------
```

	CMTE_NM	TRANSACTION_AMT
0	SENATE LEADERSHIP FUND	63.36M
1	NRSC	54.02M
2	AMERICANS FOR PROSPERITY ACTION, INC.(AFP ACTION)	24.60M
3	HONEYWELL INTERNATIONAL POLITICAL ACTION COMMI...	4.52M
4	WOMEN SPEAK OUT PAC	4.42M
5	NATIONAL ASSOCIATION OF REALTORS POLITICAL ACT...	4.00M
6	THE BOEING COMPANY POLITICAL ACTION COMMITTEE	3.57M
7	GOPAC ELECTION FUND	3.34M
8	DELOITTE POLITICAL ACTION COMMITTEE	3.16M
9	PRICEWATERHOUSECOOPERS POLITICAL ACTION COMMIT...	3.05M

In [51]:

```
contribs_by_indivs.loc[contribs_by_indivs['EMPLOYER'] == 'SELF', 'EMPLOYER'] = 'SELF-EMP
contribs_by_indivs.loc[contribs_by_indivs['EMPLOYER'] == 'SELF EMPLOYED', 'EMPLOYER'] =
contribs_by_indivs.loc[contribs_by_indivs['EMPLOYER'] == 'NOT EMPLOYED', 'EMPLOYER'] = '
contribs_by_indivs
```

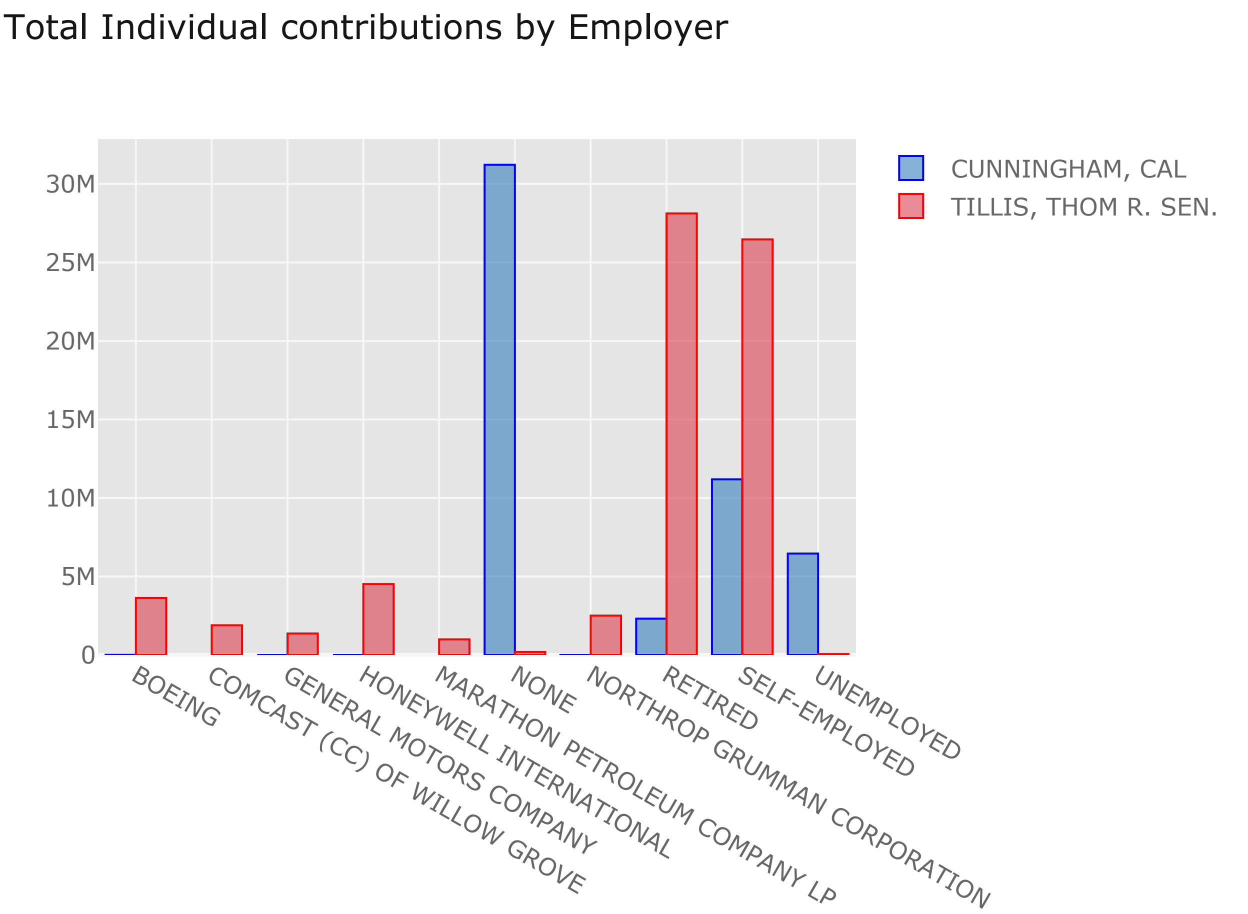
Out[51]:

	CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP
0	C00428110	N	M2	P	201902049145460549	15
1	C00428110	N	M2	P	201902049145460548	15
2	C00428110	N	M2	P	201902049145460548	15
3	C00428110	N	M2	P	201902049145460548	15
4	C00428110	N	M2	P	201902049145460549	15
...	...	...	...	...	...	...
2714376	C00659508	N	Q1	P	202004139216653073	15
2714377	C00659508	N	Q1	P	202004139216653073	15
2714378	C00659508	N	Q1	P	202004139216653074	15
2714379	C00736751	N	12P	P	202002209187234958	10

2714380	C00736751	N	12P	P	202002209187234958	10

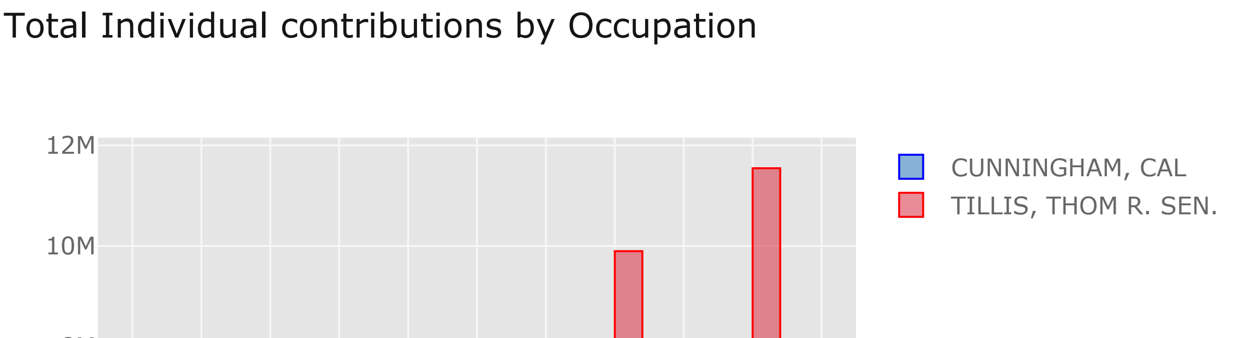
2714381 rows × 22 columns

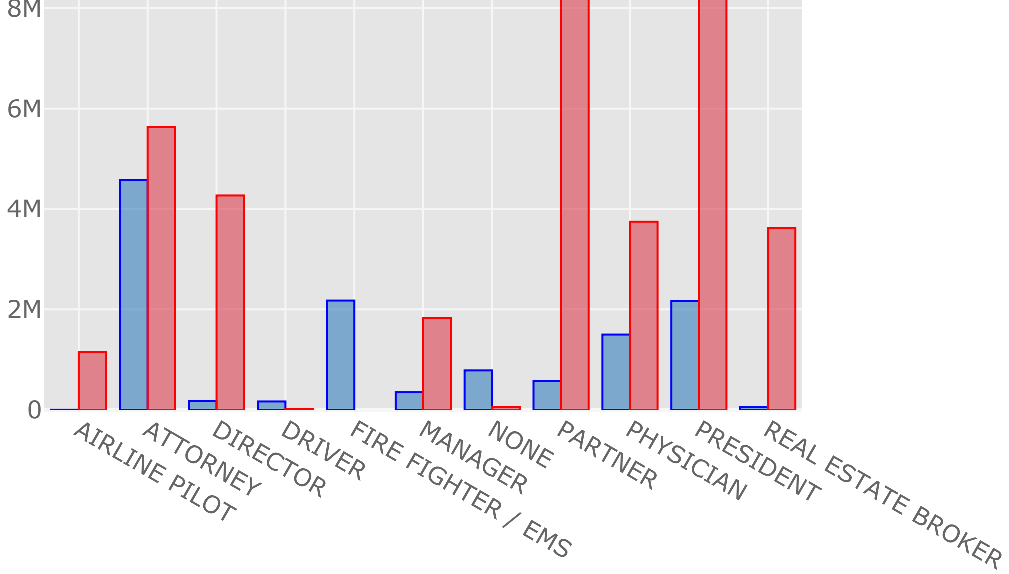
```
In [52]: top_employers = contribs_by_indivs['EMPLOYER'].value_counts()[:10].index
         by_employer = contribs_by_indivs[contribs_by_indivs['EMPLOYER'].isin(top_employers)].gro
         by_employer_pivot = by_employer.pivot_table(values='TRANSACTION_AMT', index='EMPLOYER',
         by_employer_pivot.iplot(kind='bar', colors=color_map, title='Total Individual contributi
```



Export to plot.ly »

```
In [53]: top_occupations = set(contribs_by_indivs['OCCUPATION'].value_counts()[:13].index) - set(
         by_occupation = contribs_by_indivs[contribs_by_indivs['OCCUPATION'].isin(top_occupations
         by_occupation_pivot = by_occupation.pivot_table(values='TRANSACTION_AMT', index='OCCUPAT
         by_occupation_pivot.iplot(kind='bar', colors=color_map, title='Total Individual contribu
```

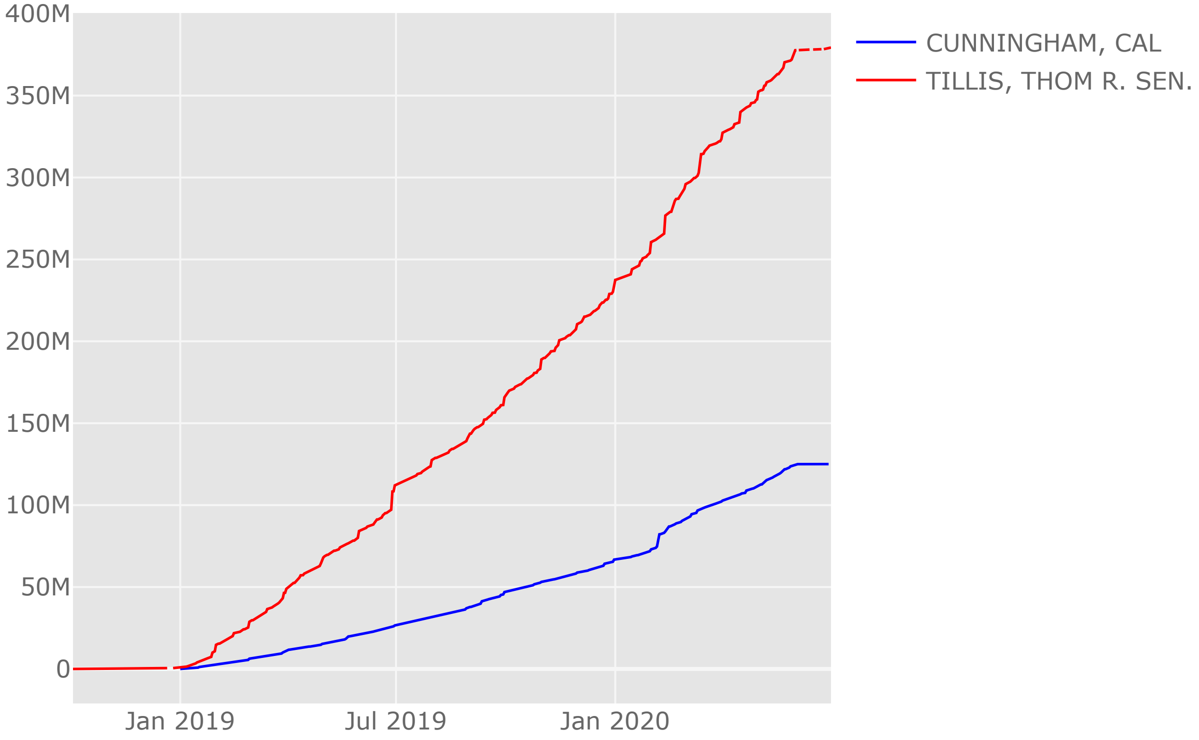




Export to plot.ly »

```
In [54]: by_date = contribs_by_indivs.groupby(['CAND_NAME', 'TRANSACTION_DT']).agg({'TRANSACTION_
         by_date_cumsum = by_date.groupby(level=0).cumsum().reset_index().pivot_table(values='TRA
         by_date_cumsum.iplot(kind='line', colors=color_map, title='Total individual contribution
```

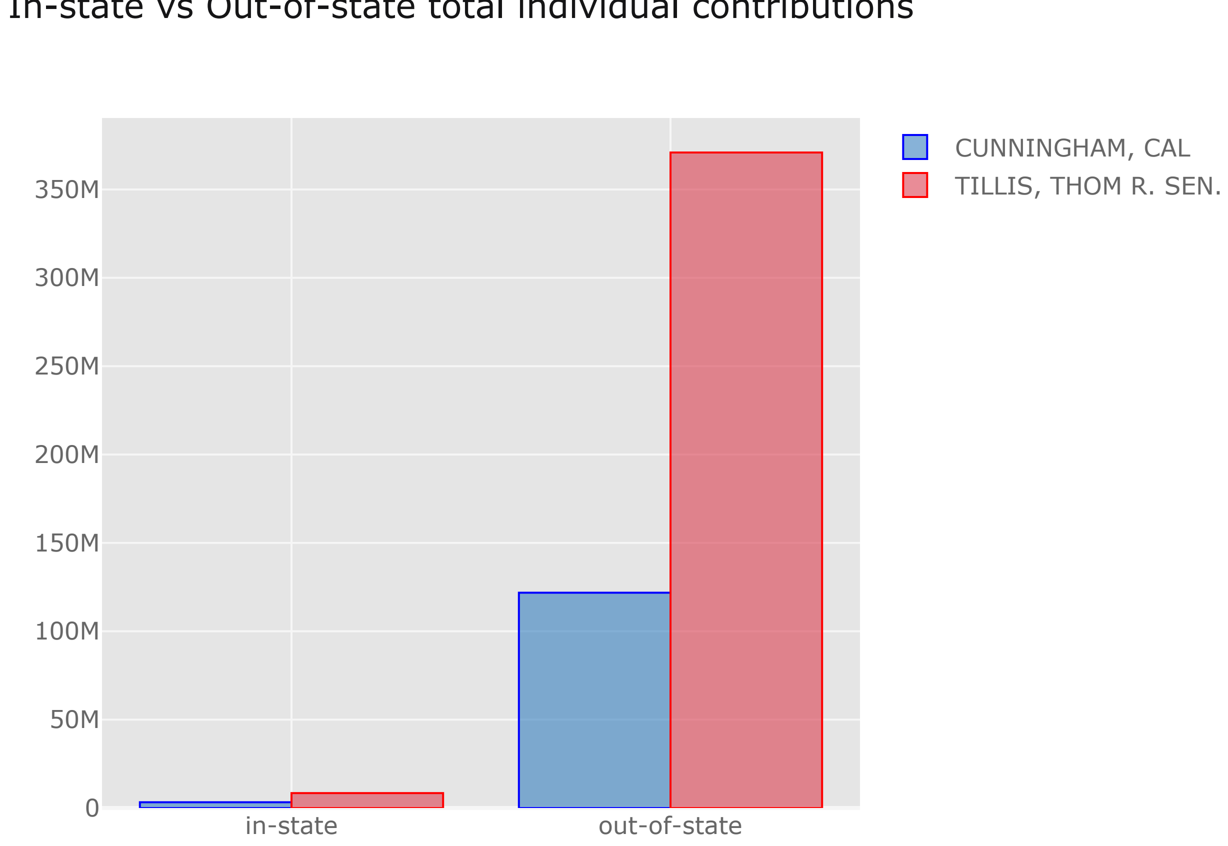
Total individual contributions over time



Export to plot.ly »

```
In [55]: contribs_by_indivs['in_state'] = np.where(contribs_by_indivs['STATE'] == state, 'in-stat
         by_in_state = contribs_by_indivs.groupby(['CAND_NAME', 'in_state']).agg({'TRANSACTION_AM
         by_in_state.pivot_table(values='TRANSACTION_AMT', index='in_state', columns='CAND_NAME')
```

In-state vs Out-of-state total individual contributions



Export to plot.ly »

In [56]:
```
by_state_and_cand = contribs_by_indivs.groupby(['STATE', 'CAND_NAME']).agg({'TRANSACTION
by_state_and_cand.head()
```

Out[56]:

	STATE	CAND_NAME	TRANSACTION_AMT
0	AA	CUNNINGHAM, CAL	11568
1	AA	TILLIS, THOM R. SEN.	40
2	AE	CUNNINGHAM, CAL	10412
3	AE	TILLIS, THOM R. SEN.	4139
4	AK	CUNNINGHAM, CAL	156673

In [57]:
```
by_state = by_state_and_cand.pivot_table(values='TRANSACTION_AMT', index='STATE', column
cand_cols = set(by_state.columns) - {'STATE'}
by_state.loc[:, cand_cols] = by_state[cand_cols].fillna(0.0)
by_state.head()
```

Out[57]:

CAND_NAME	STATE	CUNNINGHAM, CAL	TILLIS, THOM R. SEN.
0	AA	11568.0	40.0
1	AE	10412.0	4139.0
2	AK	156673.0	541903.0
3	AL	378135.0	4080413.0
4	AP	1244.0	2591.0

In [58]:
```
if democrat not in by_state.columns:
    by_state[democrat] = 0.0
```

```
if republican not in by_state.columns:
    by_state[republican] = 0.0
```

In [59]:

```
by_state['Total'] = by_state[democrat] + by_state[republican]
by_state['Republican_versus_Total_Ratio'] = by_state[republican]/by_state['Total']
by_state.head()
```

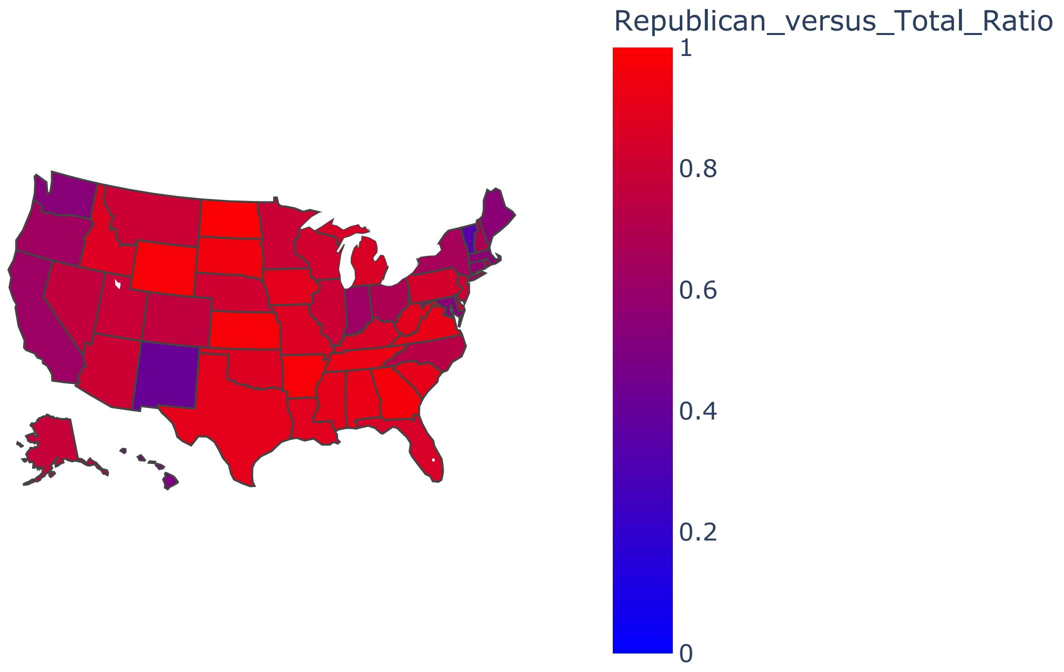
Out[59]:

CAND_NAME	STATE	CUNNINGHAM, CAL	TILLIS, THOM R. SEN.	Total	Republican_versus_Total_Ratio
0	AA	11568.0	40.0	11608.0	0.003446
1	AE	10412.0	4139.0	14551.0	0.284448
2	AK	156673.0	541903.0	698576.0	0.775725
3	AL	378135.0	4080413.0	4458548.0	0.915189
4	AP	1244.0	2591.0	3835.0	0.675619

In [60]:

```
import plotly.express as px
fig = px.choropleth(by_state, locations="STATE", color="Republican_versus_Total_Ratio",
                    hover_data=[democrat, republican], color_continuous_scale='bluered',
                    locationmode='USA-states', scope='usa',
                    title='Total individual contributions for Republican vs Democrat by
                    range_color=[0, 1])
fig.show()
```

Total individual contributions for Republican vs Democrat by state



In [61]:

```
contribs_by_indivs['YEAR_MONTH'] = contribs_by_indivs['TRANSACTION_DT'].dt.to_period('M'
contribs_by_indivs.head()
```

CMTE_ID	AMNDT_IND	RPT_TP	TRANSACTION_PGI	IMAGE_NUM	TRANSACTION_TP	ENTITY_

Out[61]:

0	C00428110	N	M2	P	201902049145460549	15	I
1	C00428110	N	M2	P	201902049145460548	15	I
2	C00428110	N	M2	P	201902049145460548	15	I
3	C00428110	N	M2	P	201902049145460548	15	I
4	C00428110	N	M2	P	201902049145460549	15	I

In [62]:

```
by_date_state_and_cand = contribs_by_indivs.groupby(['YEAR_MONTH', 'STATE', 'CAND_NAME']
by_date_state_and_cand.head()
```

Out[62]:

	YEAR_MONTH	STATE	CAND_NAME	TRANSACTION_AMT
0	2018-10	CA	TILLIS, THOM R. SEN.	800
1	2018-10	NV	TILLIS, THOM R. SEN.	450
2	2018-11	AR	TILLIS, THOM R. SEN.	156800
3	2018-11	AZ	TILLIS, THOM R. SEN.	7300
4	2018-11	FL	TILLIS, THOM R. SEN.	243400

In [63]:

```
dates = by_date_state_and_cand['YEAR_MONTH'].unique()
states = by_date_state_and_cand['STATE'].unique()
cands = by_date_state_and_cand['CAND_NAME'].unique()
cands
```

Out[63]:

```
array(['TILLIS, THOM R. SEN.', 'CUNNINGHAM, CAL'], dtype=object)
```

In [64]:

```
by_date_state_and_cand_skel = pd.DataFrame([{'YEAR_MONTH': date, 'STATE': state, 'CAND_N
by_date_state_and_cand_skel['TRANSACTION_AMT'] = 0.0
by_date_state_and_cand_skel.head()
```

Out[64]:

	YEAR_MONTH	STATE	CAND_NAME	TRANSACTION_AMT
0	2018-10	CA	TILLIS, THOM R. SEN.	0.0
1	2018-10	CA	CUNNINGHAM, CAL	0.0
2	2018-10	NV	TILLIS, THOM R. SEN.	0.0
3	2018-10	NV	CUNNINGHAM, CAL	0.0
4	2018-10	AR	TILLIS, THOM R. SEN.	0.0

In [65]:

```
by_date_state_and_cand = by_date_state_and_cand_skel.merge(by_date_state_and_cand, how='
by_date_state_and_cand['TRANSACTION_AMT'] = np.where(by_date_state_and_cand['TRANSACTION
by_date_state_and_cand = by_date_state_and_cand.drop('TRANSACTION_AMT_x', axis=1).drop('
by_date_state_and_cand.head()
```

Out[65]:

	YEAR_MONTH	STATE	CAND_NAME	TRANSACTION_AMT
0	2018-10	CA	TILLIS, THOM R. SEN.	800.0

1	2018-10	CA	CUNNINGHAM, CAL	0.0
2	2018-10	NV	TILLIS, THOM R. SEN.	450.0
3	2018-10	NV	CUNNINGHAM, CAL	0.0
4	2018-10	AR	TILLIS, THOM R. SEN.	0.0

In [66]:
```
by_date_state_and_cand = by_date_state_and_cand.sort_values('YEAR_MONTH').set_index(['YE
by_date_state_and_cand.head()
```

Out[66]:

	YEAR_MONTH	STATE	CAND_NAME	TRANSACTION_AMT
0	2018-10	CA	TILLIS, THOM R. SEN.	800.0
1	2018-10	OH	CUNNINGHAM, CAL	0.0
2	2018-10	OH	TILLIS, THOM R. SEN.	0.0
3	2018-10	NM	CUNNINGHAM, CAL	0.0
4	2018-10	NM	TILLIS, THOM R. SEN.	0.0

In [67]:
```
by_date_state = by_date_state_and_cand.pivot_table(values='TRANSACTION_AMT', index=['YEA
cand_cols = set(by_date_state.columns) - {'YEAR_MONTH', 'STATE'}
by_date_state.loc[:, cand_cols] = by_date_state[cand_cols].fillna(0.0)
by_date_state.head()
```

Out[67]:

CAND_NAME	YEAR_MONTH	STATE	CUNNINGHAM, CAL	TILLIS, THOM R. SEN.
0	2018-10	AA	0.0	0.0
1	2018-10	AE	0.0	0.0
2	2018-10	AK	0.0	0.0
3	2018-10	AL	0.0	0.0
4	2018-10	AP	0.0	0.0

In [68]:
```
if democrat not in by_date_state.columns:
    by_date_state[democrat] = 0.0
if republican not in by_date_state.columns:
    by_date_state[republican] = 0.0
```

In [69]:
```
by_date_state['Total'] = by_date_state[democrat] + by_date_state[republican]
by_date_state['RepublicanVsTotalRatio'] = by_date_state[republican]/by_date_state['Total
by_date_state = by_date_state.sort_values('YEAR_MONTH')
by_date_state['YEAR_MONTH'] = by_date_state['YEAR_MONTH'].astype('str')
by_date_state.head()
```

Out[69]:

CAND_NAME	YEAR_MONTH	STATE	CUNNINGHAM, CAL	TILLIS, THOM R. SEN.	Total	RepublicanVsTotalRatio
0	2018-10	AA	0.0	0.0	0.0	NaN
32	2018-10	MT	0.0	0.0	0.0	NaN
33	2018-10	NC	0.0	0.0	0.0	NaN
34	2018-10	ND	0.0	0.0	0.0	NaN
35	2018-10	NE	0.0	0.0	0.0	NaN