In this notebook the voting records from Alameda are

data source: https://www.rankedchoicevoting.org/data_clearinghouse (https://www.rankedchoicevoting.org/data_clearinghouse)

RCV definition: https://ballotpedia.org/Ranked-choice_voting_(RCV (https://ballotpedia.org/Ranked-choice_voting_(RCV))

Additional analysis:

- http://archive3.fairvote.org/press/san-leandro-facts/ (http://archive3.fairvote.org/press/san-leandro-facts/)
- https://laurendo.wordpress.com/2010/11/24/running-the-numbers/ (https://laurendo.wordpress.com/2010/11/24/running-the-numbers/)
- http://www.acgov.org/rov/rcv/results/index.htm (http://www.acgov.org/rov/rcv/results/index.htm)

Objective for this notebook: separate the elections into the following categories:

1. Leading candidate in the first round has greater than 50% first choice votes
2. Leading candidate in the first round has between 45-50% first choice votes
3. Leading candidate in the first round has less than 45% of first choice votes

In [1]:
```
import glob
import pandas
import time
import csv
import re
print('pandas',pandas.__version__)
```

```
pandas 0.23.4
```

data gathering: download all folders from drive manually

all the data: https://drive.google.com/drive/folders/1DJzIrTaDW3GSGJTkPTGAlpAMbozFG_pm (https://drive.google.com/drive/folders/1DJzIrTaDW3GSGJTkPTGAlpAMbozFG_pm)

download all content as a zip. Size is 1.5 GB. Of this, Sante Fe is 1.4GB

I started with just "Alameda County, CA (Berkeley, Oakland, San Leandro)" which is 18MB as a .zip

https://drive.google.com/drive/folders/1u_airJzoLC2PMYMHcF2KYJEKxxKBi5H7 (https://drive.google.com/drive/folders/1u_airJzoLC2PMYMHcF2KYJEKxxKBi5H7)

!mkdir voting_data !mkdir voting_data/Alameda !unzip voting_data/Alameda/drive-download-20190724T221439Z-001.zip

get raw data

In [2]:
```
#list_of_ballot_files = glob.glob('voting_data/Alameda/Alameda (Oakland, San Leandro, Berkeley) 2010/ballot_image_*')
list_of_ballot_files = glob.glob('voting_data/Alameda/**/ballot_image*.txt',recursive=True)
print('number of election results to parse:',len(list_of_ballot_files))
```

```
number of election results to parse: 48
```

In [3]:
```
list_of_lookup_files = glob.glob('voting_data/Alameda/**/master_lookup_*',recursive=True)
print('number of ballot lookup tables:',len(list_of_lookup_files))
```

```
number of ballot lookup tables: 51
```

pair ballot results with lookup table files

In [4]:
```
def create_ballot_lookup_tuples(list_of_ballot_files,list_of_lookup_files):
    list_of_ballots_and_lookups=[]
    for this_ballot in list_of_ballot_files:
        ballot_name = this_ballot.split('/')[-1].replace('ballot_image_','').strip().replace('.txt','').replace(' ','_')
        found_match=False
        for this_lookup in list_of_lookup_files:
            lookup_name = this_lookup.split('/')[-1].replace('master_lookup_','').strip().replace('.txt','').replace(' ','_')
            if ballot_name == lookup_name:
                list_of_ballots_and_lookups.append((this_ballot,this_lookup))
                found_match=True
        if not found_match:
            print('no lookup found for',this_ballot)
            list_of_ballots_and_lookups.append((this_ballot,None))
    return list_of_ballots_and_lookups
```

In [5]:
```
list_of_ballots_and_lookups = create_ballot_lookup_tuples(list_of_ballot
_files,list_of_lookup_files)
print('number of paired files found:',len(list_of_ballots_and_lookups))
```

```
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/BerkeleyCouncilD3/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/BerkeleyCouncilD2/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/BerkeleyCouncilD5/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/OaklandSchoolD1/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/OaklandCouncilD7/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/BerkeleyMayor/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/OaklandCouncilD1/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/OaklandSchoolD7/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/SanLeandroCouncilD2/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/SanLeandroCouncilD4/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/BerkeleyCouncilD6/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/OaklandSchoolD5/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/OaklandCouncilD3/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/OaklandCouncilD5/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/OaklandSchoolD3/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/OaklandCouncilAtLrg/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/OaklandAttorney/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2016/SanLeandroCouncilD6/ballot_image.txt
no lookup found for voting_data/Alameda/Alameda (Oakland, San Leandro,
Berkeley) 2017/BerkCouncilD4/ballot_image.txt
number of paired files found: 48
```

convert ballot to dataframe

In [6]:
```
def vote_table(ballot_file):
    fwf=False
    with open(ballot_file,'r') as fil:
        file_contents = fil.readlines()
        if len(file_contents[0].strip())==45:
            fwf=True

    if fwf:
        df = pandas.read_fwf(ballot_file,
                             header=None,
                             widths=[7,9,7,3,7,3,7,1,1])
        df.columns=['contest_id','pref_voter_id',
            'serial_number','tally_type_id',
            'precinct_id','Vote_Rank',
            'CandidateID','over_vote','under_vote']
        df_cand = df[df['CandidateID']!=0] # drop rows where no candidat
e is specified
    else:
        df_cand = pandas.read_csv(ballot_file, sep='\t', engine='python'
)#,delim_whitespace=True)

    return df_cand
```

let's see what that looks like for a ballot file

In [7]:
```
start_time=time.time()
df_votes = vote_table(list_of_ballots_and_lookups[0][0])
df_votes.shape
print('elapsed',round(time.time()-start_time,2),'seconds')
```

```
elapsed 14.95 seconds
```

In [8]:
```
df_votes.head()
```

Out[8]:

	contest_id	pref_voter_id	serial_number	tally_type_id	precinct_id	Vote_Rank	Candida
0	68	30773	1	3	101	1	394
1	68	30773	1	3	101	2	395
3	68	30774	1	3	101	1	395
6	68	30775	1	3	101	1	394
9	68	30776	1	3	101	1	394

when possible, decorate the ballot table with candidate names

In [9]:
```
def candidate_id_table(master_lookup):
    df = pandas.read_fwf(master_lookup,
                                header=None,
                                widths=[10,7,50,7,7,1,1])
    df.columns=['record_id','id','description','list_order','candidates_
contest_id','is_writein','is_provisional']
    return df[df['record_id']=='Candidate']
```

In [10]:
```
df_cand = candidate_id_table(list_of_ballots_and_lookups[0][1])
df_cand
```

Out[10]:

	record_id	id	description	list_order	candidates_contest_id	is_writein	is_provisiona
0	Candidate	394	JANE BRUNNER	1	68	0	0
1	Candidate	395	BARBARA PARKER	2	68	0	0
2	Candidate	92	Write-In	3	68	1	0

join the candidate names with candidate IDs

For example,

In [11]:
```
start_time=time.time()
df_cand_reduced = df_cand.drop(['record_id', 'list_order','candidates_co
ntest_id','is_writein','is_provisional'], axis=1)
cand_and_votes_df = pandas.merge(df_votes,df_cand_reduced,how='left',lef
t_on='CandidateID', right_on='id')

print('elapsed',round(time.time()-start_time,2),'seconds')
cand_and_votes_df.head()
```

```
elapsed 0.09 seconds
```

Out[11]:

	contest_id	pref_voter_id	serial_number	tally_type_id	precinct_id	Vote_Rank	Candida
0	68	30773	1	3	101	1	394
1	68	30773	1	3	101	2	395
2	68	30774	1	3	101	1	395
3	68	30775	1	3	101	1	394
4	68	30776	1	3	101	1	394

The following cell does all the computational work needed for the task associated with the objective for the notebook.

`df_votes[df_votes['Vote_Rank']==1]` down selects only the rows where vote_rank==1

`.groupby('CandidateID')` groups the rows by candidate ID

`['Vote_Rank']` selects the column "vote_rank"

`.count()` counts how many first rank votes each candidate got

In [12]:
```
series_of_candidates_and_first_choice_count = cand_and_votes_df[cand_and
_votes_df['Vote_Rank']==1].groupby('description')['Vote_Rank'].count()
series_of_candidates_and_first_choice_count
```

Out[12]:
```
description
BARBARA PARKER    89727
JANE BRUNNER      40356
Write-In           1051
Name: Vote_Rank, dtype: int64
```

In [13]:
```
number_of_first_choice_votes =  series_of_candidates_and_first_choice_co
unt.sum()
number_of_first_choice_votes
```

Out[13]:
```
131134
```

with the series describing candidates and votes, and with the scalar number of first round votes, we can now address the question of what percentage of votes the Leading candidate in the first round has

In [14]:
```
(series_of_candidates_and_first_choice_count/number_of_first_choice_votes)*100
```

Out[14]:
```
description
BARBARA PARKER    68.423902
JANE BRUNNER      30.774627
Write-In           0.801470
Name: Vote_Rank, dtype: float64
```

In [15]:
```
print('percentage votes for the leading candidate',series_of_candidates_and_first_choice_count.idxmax(),
      'in the first round:',max((series_of_candidates_and_first_choice_count/number_of_first_choice_votes)*100))
```

```
percentage votes for the leading candidate BARBARA PARKER in the first round: 68.42390226790917
```

Now that we know what to do, encapsulate the above lines in a function

In [16]:
```
def percentage_of_votes_for_leading_candidate_in_first_round(ballot_file
,candidate_names):
    df_votes = vote_table(ballot_file)
    if candidate_names is not None:
        df_cand = candidate_id_table(candidate_names)
        df_cand_reduced = df_cand.drop(['record_id', 'list_order','candi
dates_contest_id','is_writein','is_provisional'], axis=1)
        cand_and_votes_df = pandas.merge(df_votes,df_cand_reduced,how='l
eft',left_on='CandidateID', right_on='id')
        series_of_candidates_and_first_choice_count = cand_and_votes_df[
cand_and_votes_df['Vote_Rank']==1].groupby('description')['Vote_Rank'].c
ount()
    else:
        series_of_candidates_and_first_choice_count = df_votes[df_votes[
'Vote_Rank']==1].groupby('CandidateID')['Vote_Rank'].count()
    number_of_first_choice_votes =  series_of_candidates_and_first_choic
e_count.sum()
    if number_of_first_choice_votes==0:
        print('no first choice votes present in ballot file',ballot_file
)
    reslt_dict={}
    reslt_dict['number of candidates']=len(series_of_candidates_and_firs
t_choice_count)
    reslt_dict['number of first choice votes']=number_of_first_choice_vo
tes
    reslt_dict['name or ID of top-ranked candidate']=series_of_candidate
s_and_first_choice_count.idxmax()
    reslt_dict['percentage of votes in first round for leading candidat
e']=max((series_of_candidates_and_first_choice_count.values/number_of_fi
rst_choice_votes)*100)

    if reslt_dict['percentage of votes in first round for leading candid
ate']>50:
        reslt_dict['Leading candidate in the first round has greater tha
n 50% first choice votes']='Yes'
        reslt_dict['Leading candidate in the first round has between 45-
50% first choice votes']='No'
        reslt_dict['Leading candidate in the first round has less than 4
5% of first choice votes']='No'
    elif (reslt_dict['percentage of votes in first round for leading can
didate']<=50 and
          reslt_dict['percentage of votes in first round for leading can
didate']>=45):
        reslt_dict['Leading candidate in the first round has greater tha
n 50% first choice votes']='No'
        reslt_dict['Leading candidate in the first round has between 45-
50% first choice votes']='Yes'
        reslt_dict['Leading candidate in the first round has less than 4
5% of first choice votes']='No'
    elif (reslt_dict['percentage of votes in first round for leading can
didate']<45):
        reslt_dict['Leading candidate in the first round has greater tha
n 50% first choice votes']='No'
        reslt_dict['Leading candidate in the first round has between 45-
50% first choice votes']='No'
        reslt_dict['Leading candidate in the first round has less than 4
```

```
5% of first choice votes']='Yes'
    else:
        print("ERROR: this condition should never be reached")

    return reslt_dict
```

validate that the function does what we want for a single election

In [17]:
```
percentage_of_votes_for_leading_candidate_in_first_round(list_of_ballots
_and_lookups[0][0],list_of_ballots_and_lookups[0][1])
```

Out[17]:
```
{'number of candidates': 3,
 'number of first choice votes': 131134,
 'name or ID of top-ranked candidate': 'BARBARA PARKER',
 'percentage of votes in first round for leading candidate': 68.4239022
6790917,
 'Leading candidate in the first round has greater than 50% first choic
e votes': 'Yes',
 'Leading candidate in the first round has between 45-50% first choice
votes': 'No',
 'Leading candidate in the first round has less than 45% of first choic
e votes': 'No'}
```

loop over that function and write the results to file

In [18]:
```
def reslts_to_file(file_name,list_of_ballots_and_lookups):
    list_of_dicts=[]
    for ballot_and_lookup in list_of_ballots_and_lookups:
        reslt_dict = percentage_of_votes_for_leading_candidate_in_first_
round(
            ballot_and_lookup[0],ballot_and_lookup[1])
        reslt_dict['ballot file'] = ballot_and_lookup[0]
        if ballot_and_lookup[1] is not None:
            reslt_dict['lookup_table']=ballot_and_lookup[1]
        else:
            reslt_dict['lookup_table']="no lookup table"
        list_of_dicts.append(reslt_dict)
    pandas.DataFrame.from_dict(list_of_dicts).to_csv(file_name,index=Fal
se)
    return
```

In [19]:
```
file_name='alameda_percentage.csv'
start_time=time.time()
reslts_to_file(file_name,list_of_ballots_and_lookups)
print('elapsed',round(time.time()-start_time,2),'seconds')
```

```
elapsed 132.9 seconds
```

Pierce County data

https://www.rankedchoicevoting.org/data_clearinghouse (https://www.rankedchoicevoting.org/data_clearinghouse)

https://drive.google.com/drive/folders/1DJzIrTaDW3GSGJTkPTGAlpAMbozFG_pm (https://drive.google.com/drive/folders/1DJzIrTaDW3GSGJTkPTGAlpAMbozFG_pm)

In [20]:
```
list_of_files = glob.glob('voting_data/Pierce_County/Pierce County/*')
len(list_of_files)
```

Out[20]:
```
8
```

In [21]:
```
list_of_files
```

Out[21]:
```
['voting_data/Pierce_County/Pierce County/Pierce County Auditor 2009 Ba
llot Image.txt',
 'voting_data/Pierce_County/Pierce County/Pierce County Executive 2008
Master Lookup.txt',
 'voting_data/Pierce_County/Pierce County/Pierce County Assessor - Trea
surer 2008 Ballot Image.txt',
 'voting_data/Pierce_County/Pierce County/Pierce County Council, Distri
ct No. 2 2008 Master Lookup.txt',
 'voting_data/Pierce_County/Pierce County/Pierce County Council, Distri
ct No. 2 2008 Ballot Image.txt',
 'voting_data/Pierce_County/Pierce County/Pierce County Assessor - Trea
surer 2008 Master Lookup.txt',
 'voting_data/Pierce_County/Pierce County/Pierce County Executive 2008
Ballot Image Data.txt',
 'voting_data/Pierce_County/Pierce County/Pierce County Auditor 2009 Ma
ster Lookup.txt']
```

In [22]:
```
list_of_ballot_files=[]
for filename in list_of_files:
    if filename.endswith('.txt'):
        with open(filename,'r') as fil:
            file_contents = fil.readlines()
        if len(file_contents[0].strip())==45:
            print(filename)
            list_of_ballot_files.append(filename)
            print(file_contents[1])
```

```
voting_data/Pierce_County/Pierce County/Pierce County Auditor 2009 Ball
ot Image.txt
00007140000154360000001005000002002000044000

voting_data/Pierce_County/Pierce County/Pierce County Assessor - Treasu
rer 2008 Ballot Image.txt
00001920000631580000001005000002002000000001

voting_data/Pierce_County/Pierce County/Pierce County Council, District
No. 2 2008 Ballot Image.txt
00001930000769770000001005000063002000013100

voting_data/Pierce_County/Pierce County/Pierce County Executive 2008 Ba
llot Image Data.txt
00001970000631580000001005000002002000000001
```

In [23]:
```
list_of_lookup_files = glob.glob('voting_data/Pierce_County/Pierce Count
y/*ookup.txt')
list_of_lookup_files
```

Out[23]:
```
['voting_data/Pierce_County/Pierce County/Pierce County Executive 2008
Master Lookup.txt',
 'voting_data/Pierce_County/Pierce County/Pierce County Council, Distri
ct No. 2 2008 Master Lookup.txt',
 'voting_data/Pierce_County/Pierce County/Pierce County Assessor - Trea
surer 2008 Master Lookup.txt',
 'voting_data/Pierce_County/Pierce County/Pierce County Auditor 2009 Ma
ster Lookup.txt']
```

In [24]:
```
def create_ballot_lookup_tuples(list_of_ballot_files,list_of_lookup_file
s):
    list_of_ballots_and_lookups=[]
    for this_ballot in list_of_ballot_files:
        ballot_name = re.sub(r'ballot.*', '', this_ballot.split('/')[-1
],flags=re.IGNORECASE)
        #print(ballot_name)
        found_match=False
        for this_lookup in list_of_lookup_files:
            lookup_name = re.sub(r'master.*','',this_lookup.split('/')[-
1],flags=re.IGNORECASE)
            #print(lookup_name)
            if ballot_name == lookup_name:
                list_of_ballots_and_lookups.append((this_ballot,this_loo
kup))
                found_match=True
        if not found_match:
            print('no lookup found for',this_ballot)
            list_of_ballots_and_lookups.append((this_ballot,None))
    return list_of_ballots_and_lookups
```

In [25]:
```
list_of_ballots_and_lookups = create_ballot_lookup_tuples(list_of_ballot
_files,list_of_lookup_files)
print('number of paired files found:',len(list_of_ballots_and_lookups))
```

```
number of paired files found: 4
```

In [26]:
```
file_name='pierceCounty_percentage.csv'
start_time=time.time()
reslts_to_file(file_name,list_of_ballots_and_lookups)
print('elapsed',round(time.time()-start_time,2),'seconds')
```

```
elapsed 79.16 seconds
```

San Fransisco

In [27]:
```
list_of_files = glob.glob('voting_data/San_Fransisco/San Francisco/**/*'
)
len(list_of_files)
```

Out[27]: 46

In [28]:
```
list_of_ballot_files=[]
for filename in list_of_files:
    if filename.endswith('.txt'):
        with open(filename,'r') as fil:
            file_contents = fil.readlines()
        if len(file_contents[0].strip())==45:
            #print(filename)
            list_of_ballot_files.append(filename)
            #print(file_contents[1])
print('number of ballot files:',len(list_of_ballot_files))
```

```
number of ballot files: 14
```

In [29]:
```
list_of_lookup_files=[]
for filename in list_of_files:
    if filename.endswith('.txt'):
#        print(filename)
        if 'master' in filename.lower():
            list_of_lookup_files.append(filename)
print('number of lookup files:',len(list_of_lookup_files))
```

```
number of lookup files: 14
```

In [30]:
```
def create_ballot_lookup_tuples(list_of_ballot_files,list_of_lookup_file
s):
    list_of_ballots_and_lookups=[]
    for this_ballot in list_of_ballot_files:
        ballot_name = re.sub(r'ballot.*', '', ''.join(this_ballot.split(
'/')[-2:]),flags=re.IGNORECASE)
        #print('BALLOT NAME:',ballot_name)
        found_match=False
        for this_lookup in list_of_lookup_files:
            lookup_name = re.sub(r'master.*','',''.join(this_lookup.spli
t('/')[-2:]),flags=re.IGNORECASE)
            #print('LOOKUP NAME:',lookup_name)
            if ballot_name == lookup_name:
                list_of_ballots_and_lookups.append((this_ballot,this_loo
kup))
                found_match=True
        if not found_match:
            print('no lookup found for',this_ballot)
            list_of_ballots_and_lookups.append((this_ballot,None))
    return list_of_ballots_and_lookups
```

In [31]:
```
list_of_ballots_and_lookups = create_ballot_lookup_tuples(list_of_ballot
_files,list_of_lookup_files)
print('number of paired files found:',len(list_of_ballots_and_lookups))
```

```
number of paired files found: 14
```

In [32]:
```
file_name='sanFrancisco_percentage.csv'
start_time=time.time()
reslts_to_file(file_name,list_of_ballots_and_lookups)
print('elapsed',round(time.time()-start_time,2),'seconds')
```

```
elapsed 150.42 seconds
```