In [1]:
```
#### Maine Congressional District 2 (2018) Ranked Choice Voting Replication
# by Nathan Tefft, Ph.D., 11/29/2018
# This notebook uses an open source platform to replicate the Maine Congressio
nal District 2 ranked choice voting results certified to the Governor on 11/2
6/18 (obtained from https://www.maine.gov/sos/cec/elec/results/results18.html#
Nov6).
# The replication implements the rank-choiced voting rules as provided in 21-A
 M.R.S.A. Chapter 723-A, sub-chapter 1, described at https://www.maine.gov/so
s/cec/elec/upcoming/pdf/250rcvnew.pdf
# Although the rules allow for a "batch elimination" based on mathematical imp
ossibility (used by the Secretary of State's office), I simply run subsequent
 instant runoffs to demonstrate the equivalence of the two approaches
# The replication was specifically written for the Maine Congressional Distric
t 2 race (2018), so it would need to be modified to replicate other races
# The code may not be particularly efficient, and I hope to one day improve it
s efficiency, but it transparently implements the process for those familiar w
ith the Python language


# Import packages for mathematical operations (numpy) and data manipulation (p
andas)

import numpy
import pandas
```

In [2]:
```
# Import the cleaned ballot data from the Maine Secretary of State's web site
 (https://www.maine.gov/sos/cec/elec/results/results18.html#Nov6)

ballots = pandas.read_excel('Nov18CVRExportFINAL1.xlsx').rename(index=str, col
umns={"Rep. to Congress 1st Choice District 2": "choice1", "Rep. to Congress 2
nd Choice District 2": "choice2", "Rep. to Congress 3rd Choice District 2": "c
hoice3", "Rep. to Congress 4th Choice District 2": "choice4", "Rep. to Congres
s 5th Choice District 2": "choice5"})
ballots = ballots.append(pandas.read_excel('Nov18CVRExportFINAL2.xlsx').rename
(index=str, columns={"Rep. to Congress 1st Choice District 2": "choice1", "Re
p. to Congress 2nd Choice District 2": "choice2", "Rep. to Congress 3rd Choice
 District 2": "choice3", "Rep. to Congress 4th Choice District 2": "choice4",
"Rep. to Congress 5th Choice District 2": "choice5"}), sort=True)
ballots = ballots.append(pandas.read_excel('Nov18CVRExportFINAL3.xlsx').rename
(index=str, columns={"Rep. to Congress 1st Choice District 2": "choice1", "Re
p. to Congress 2nd Choice District 2": "choice2", "Rep. to Congress 3rd Choice
 District 2": "choice3", "Rep. to Congress 4th Choice District 2": "choice4",
"Rep. to Congress 5th Choice District 2": "choice5"}), sort=True)
ballots = ballots.append(pandas.read_excel('UOCAVA-FINALRepCD2.xlsx').rename(i
ndex=str, columns={"Rep. to Congress District 2 1st Choice": "choice1", "Rep.
 to Congress District 2 2nd Choice": "choice2", "Rep. to Congress District 2 3
rd Choice": "choice3", "Rep. to Congress District 2 4th Choice": "choice4", "R
ep. to Congress District 2 5th Choice": "choice5"}), sort=True)
ballots = ballots.append(pandas.read_excel('UOCAVA-AUX-CVRRepCD2.xlsx').rename
(index=str, columns={"Rep. to Congress District 2 1st Choice": "choice1", "Re
p. to Congress District 2 2nd Choice": "choice2", "Rep. to Congress District 2
 3rd Choice": "choice3", "Rep. to Congress District 2 4th Choice": "choice4",
"Rep. to Congress District 2 5th Choice": "choice5"}), sort=True)
ballots = ballots.append(pandas.read_excel('UOCAVA2CVRRepCD2.xlsx').rename(ind
ex=str, columns={"Rep. to Congress District 2 1st Choice": "choice1", "Rep. to
 Congress District 2 2nd Choice": "choice2", "Rep. to Congress District 2 3rd
 Choice": "choice3", "Rep. to Congress District 2 4th Choice": "choice4", "Re
p. to Congress District 2 5th Choice": "choice5"}), sort=True)
ballots = ballots.append(pandas.read_excel('AUXCVRProofedCVR95RepCD2.xlsx').re
name(index=str, columns={"Rep. to Congress District 2 1st Choice": "choice1",
"Rep. to Congress District 2 2nd Choice": "choice2", "Rep. to Congress Distric
t 2 3rd Choice": "choice3", "Rep. to Congress District 2 4th Choice": "choice
4", "Rep. to Congress District 2 5th Choice": "choice5"}), sort=True)
ballots = ballots.append(pandas.read_excel('RepCD2-8final.xlsx').rename(index=
str, columns={"Rep. to Congress 1st Choice District 2": "choice1", "Rep. to Co
ngress 2nd Choice District 2": "choice2", "Rep. to Congress 3rd Choice Distric
t 2": "choice3", "Rep. to Congress 4th Choice District 2": "choice4", "Rep. to
 Congress 5th Choice District 2": "choice5"}), sort=True)
```

In [3]:
```
# Remove extra characters and spaces from candidate names, and store ballots i
n choices dataframe for rules implementation

choices = pandas.DataFrame()
choices['1'] = ballots['choice1'].str.replace("\(5931\)","").str.replace("\(54
71\)","").str.strip()
choices['2'] = ballots['choice2'].str.strip()
choices['3'] = ballots['choice3'].str.strip()
choices['4'] = ballots['choice4'].str.strip()
choices['5'] = ballots['choice5'].str.strip()
```

In [4]:
```
# Count the total number of ballots to tabulate

choices['1'].count()
```

Out[4]:
```
296077
```

In [5]:
```
# Tabulate the raw counts of first round choices

choices['1'].value_counts()
```

Out[5]:
```
REP Poliquin, Bruce     133993
DEM Golden, Jared F.    131822
Bond, Tiffany L.         16415
Hoar, William R.S.        6782
undervote                 6641
overvote                   424
Name: 1, dtype: int64
```

In [6]:
```
# Tabulate the raw percents of first round choices

100*choices['1'].value_counts()/choices['1'].value_counts().sum()
```

Out[6]:
```
REP Poliquin, Bruce     45.256133
DEM Golden, Jared F.    44.522877
Bond, Tiffany L.         5.544166
Hoar, William R.S.       2.290620
undervote                2.242998
overvote                 0.143206
Name: 1, dtype: float64
```

In [7]:
```
# Before conducting the first instant runoff, implement rules regarding missin
g and multiple votes per rank (as provided in 21-A M.R.S.A. Chapter 723-A, sub
-chapter 1, described at https://www.maine.gov/sos/cec/elec/upcoming/pdf/250rc
vnew.pdf)
# Two consecutive skipped rankings: When a voter does not mark two or more con
secutive rankings, then the ballot is deemed exhausted for that contest, and n
o subsequent candidate rankings on that ballot are counted

choices['3'] = numpy.where(choices['1'].str.contains('undervote') & choices[
'2'].str.contains('undervote'),'invalidated',choices['3'])
choices['4'] = numpy.where(choices['3'].str.contains('invalidated') | (choices
['2'].str.contains('undervote') & choices['3'].str.contains('undervote')),'inv
alidated',choices['4'])
choices['5'] = numpy.where(choices['3'].str.contains('invalidated') | choices[
'4'].str.contains('invalidated') | (choices['3'].str.contains('undervote') & c
hoices['4'].str.contains('undervote')),'invalidated',choices['5'])
```

In [8]:
```
# An overvote (when a voter marks more than one candidate for the same rankin
g) invalidates the overvoted rankings and all subsequent rankings marked for t
hat contest on the ballot.

choices['2'] = numpy.where(choices['1'].str.contains('overvote|invalidated'),
'invalidated',choices['2'])
choices['3'] = numpy.where(choices['2'].str.contains('overvote|invalidated'),
'invalidated',choices['3'])
choices['4'] = numpy.where(choices['3'].str.contains('overvote|invalidated'),
'invalidated',choices['4'])
choices['5'] = numpy.where(choices['4'].str.contains('overvote|invalidated'),
'invalidated',choices['5'])
```

In [9]:
```
#### FIRST ROUND (ZERO RUNOFF)
# Being with the "zero" runoff, which tallies the first choice, moving to the
 second choice if there is a single undervote, according to the "single skippe
d ranking" rule
choices['runoff0'] = numpy.where(choices['1'].str.contains('undervote'),choice
s['2'],choices['1'])

# Tabulate all ballot statuses after this runoff
choices['runoff0'].value_counts()
```

Out[9]:
```
REP Poliquin, Bruce     134184
DEM Golden, Jared F.    132013
Bond, Tiffany L.         16552
Hoar, William R.S.        6875
undervote                 6018
overvote                   435
Name: runoff0, dtype: int64
```

In [10]:
```
# Mark ballots that are counted in this runoff as "continuing"
choices['continuing0'] = ~choices['runoff0'].isin(['exhausted','overvote','und
ervote','invalidated'])

# Total of ballots that are counted in this runoff
choices['continuing0'].value_counts()
```

Out[10]:
```
True     289624
False      6453
Name: continuing0, dtype: int64
```

In [11]:
```
# Calculate and display percents of vote for each candidate among ballots coun
ted in this runoff
counts0 = choices.loc[choices['continuing0']]['runoff0'].value_counts()
100*counts0/counts0.sum()
```

Out[11]:
```
REP Poliquin, Bruce     46.330415
DEM Golden, Jared F.    45.580822
Bond, Tiffany L.         5.714996
Hoar, William R.S.       2.373767
Name: runoff0, dtype: float64
```

In [12]:
```
#### FIRST AUTOMATIC RUNOFF
# Using the results from the first round, eliminate the last place vote-getter
 by assigning their voters' next-highest ranked choice, according to the rules
 referenced above

choices['runoff1'] = choices['runoff0']
choices['runoff1'] = numpy.where(choices['runoff1'].str.contains('undervote|Ho
ar, William R.S.'),choices['2'],choices['runoff1'])
choices['runoff1'] = numpy.where(choices['runoff1'].str.contains('undervote|Ho
ar, William R.S.'),choices['3'],choices['runoff1'])
choices['runoff1'] = numpy.where(choices['runoff1'].str.contains('undervote|Ho
ar, William R.S.'),choices['4'],choices['runoff1'])
choices['runoff1'] = numpy.where(choices['runoff1'].str.contains('undervote|Ho
ar, William R.S.'),choices['5'],choices['runoff1'])
choices['runoff1'] = numpy.where(choices['runoff1'].str.contains('Hoar, Willia
m R.S.'),'exhausted',choices['runoff1'])

# Tabulate all ballot statuses after this runoff
choices['runoff1'].value_counts()
```

Out[12]:
```
REP Poliquin, Bruce     135073
DEM Golden, Jared F.    133216
Bond, Tiffany L.         19173
invalidated               8056
overvote                   456
exhausted                   94
undervote                    9
Name: runoff1, dtype: int64
```

In [13]:
```
# Mark ballots that are counted in this runoff as "continuing"
choices['continuing1'] = ~choices['runoff1'].isin(['exhausted','overvote','und
ervote','invalidated'])

# Total of ballots that are counted in this runoff
choices['continuing1'].value_counts()
```

Out[13]:
```
True     287462
False      8615
Name: continuing1, dtype: int64
```

In [14]:
```
# Calculate and display percents of vote for each candidate among ballots coun
ted in this runoff
counts1 = choices.loc[choices['continuing1']]['runoff1'].value_counts()
100*counts1/counts1.sum()
```

Out[14]:
```
REP Poliquin, Bruce     46.988124
DEM Golden, Jared F.    46.342125
Bond, Tiffany L.         6.669751
Name: runoff1, dtype: float64
```

In [15]:
```
#### SECOND AUTOMATIC RUNOFF
# Using the results from the first runoff, eliminate the last place vote-gette
r by assigning their voters' next-highest ranked choice, according to the rule
s referenced above

choices['runoff2'] = choices['runoff1']
choices['runoff2'] = numpy.where(choices['runoff2'].str.contains('undervote|Ho
ar, William R.S.|Bond, Tiffany L.'),choices['2'],choices['runoff2'])
choices['runoff2'] = numpy.where(choices['runoff2'].str.contains('undervote|Ho
ar, William R.S.|Bond, Tiffany L.'),choices['3'],choices['runoff2'])
choices['runoff2'] = numpy.where(choices['runoff2'].str.contains('undervote|Ho
ar, William R.S.|Bond, Tiffany L.'),choices['4'],choices['runoff2'])
choices['runoff2'] = numpy.where(choices['runoff2'].str.contains('undervote|Ho
ar, William R.S.|Bond, Tiffany L.'),choices['5'],choices['runoff2'])
choices['runoff2'] = numpy.where(choices['runoff2'].str.contains('Hoar, Willia
m R.S.|Bond, Tiffany L.'),'exhausted',choices['runoff2'])

# Tabulate all ballot statuses after this runoff
choices['runoff2'].value_counts()
```

Out[15]:
```
DEM Golden, Jared F.    142440
REP Poliquin, Bruce     138931
invalidated              13745
overvote                   533
exhausted                  342
undervote                   86
Name: runoff2, dtype: int64
```

In [16]:
```
# Mark ballots that are counted in this runoff as "continuing"
choices['continuing2'] = ~choices['runoff2'].isin(['exhausted','overvote','und
ervote','invalidated'])

# Total of ballots that are counted in this runoff
choices['continuing2'].value_counts()
```

Out[16]:
```
True     281371
False     14706
Name: continuing2, dtype: int64
```

In [17]:
```
# Calculate and display percents of vote for each candidate among ballots coun
ted in this runoff
counts2 = choices.loc[choices['continuing2']]['runoff2'].value_counts()
100*counts2/counts2.sum()
```

Out[17]:
```
DEM Golden, Jared F.    50.623554
REP Poliquin, Bruce     49.376446
Name: runoff2, dtype: float64
```