

IRV, Why the Counting Method Matters?

Some of you may have drugged through some of the scenarios I presented saying that depending on the voting method chosen I different candidate could be elected. I never imagined how great a case study Aspen's own city council race would be in demonstrating this. If you use the different methods discussed for tabulating the ballots you end up with not 1 outcome but with **4** possible outcomes. I'll give a very brief description of each tabulation procedure and the outcome at each step. The data used to run the simulations comes from the election itself and it spoils ballots in the same manor. I believe there are some errors in the procedure and data used in the election but for consistency I followed the election procedures and data. I will separately address the data and spoiling errors in another paper.

For those not interested in the details I will give a table that summarizes the results here.

	Method 1	Method 2	Method 3	Method 4	Method 5
Seat 1	D. Johnson	D. Johnson	D. Johnson	D. Johnson	D. Johnson
Seat 2	Torre	Behrendt	Frisch	J. Johnson	Torre

Method 1 – Official Election Procedure.

Method 2 – Two Vote Sequential Elimination.

Method 3 – One Vote Sequential Elimination (Restart after first seat).

Method 4 – One Vote Sequential Elimination (No Restart).

Method 5 – IRV version of traditional runoff.

The first seat was always filled by D. Johnson the second seat depended upon the method chosen. The details of each method and the elimination procedures are all included below. For those interested I'll also include the text of the program files used to generate this data.

Method 1: Tabulation Used in the Election.

Process:

1. Count the top two votes on everyone's ballot equally.
 - a. Check for winners (50+%).
 - i. If there are two winners the election is over.
 - ii. If there is exactly one winner, hold a runoff between the second and third candidates.
 - iii. If there are no winners take the top four candidates and go to 2.
2. Count only the top vote on everyone's ballot.
 - a. Check for winners (50+%).
 - i. If there is one winner that candidate is elected remove the winning candidate and restart two with the remaining three candidates.
 - ii. If there are no winners eliminate the last place candidate and restart at 2.

Below I will give tables showing the vote totals at each round of the election, showing who gets eliminated and elected.

Round 1 – Threshold 1244

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
538	717	695	777	735	102	268	884	138

Round 2 – Threshold 1201

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	569	0	533	538	0	0	760	0

Round 3 – Threshold 1154

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	737	0	0	678	0	0	891	0

Round 4 – Threshold 1069

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	906	0	0	0	0	0	1230	0

So D. Johnson won the first city council seat. Now we have a runoff between the three other candidates to see who wins the second seat.

Round 5 – Threshold 1114

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	666	0	736	825	0	0	0	0

Round 5 – Threshold 1048

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	0	0	1063	1032	0	0	0	0

So Torre wins the second seat by a mere 31 votes.

Result 1: D. Johnson and Torre.

Method 2: Two Vote Sequential Elimination.

Process:

1. Count the top two votes on everyone's ballot equally.
 - a. Check for winners (50+%).
 - i. If there are two winners the election is over.
 - ii. If there is exactly one winner, hold a single vote SE runoff between the remaining candidates.
 - iii. If there are no winners eliminate the candidate with fewest votes and restart.

Round 1 – Threshold 1244 (Write Ins already eliminated)

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
538	717	696	777	735	102	268	884	138

Round 2 – Threshold 1243

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
547	726	710	792	748	0	276	890	147

Round 3 – Threshold 1242

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
569	741	732	811	762	0	296	901	0

Round 4 – Threshold 1237

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
600	771	766	852	804	0	0	951	0

Round 5 – Threshold 1232

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	853	863	953	898	0	0	1054	0

Round 6 – Threshold 1201

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	0	966	1133	1059	0	0	1183	0

Round 7 – Threshold 1161

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	0	0	1317	1343	0	0	1368	0

Result 2: D. Johnson and Behrendt.

Method 3: One Vote Sequential Elimination (Mayor's Race twice).

Process:

1. Count the top votes on everyone's ballot.
 - a. Check for winners (50+%).
 - i. For the first winner eliminate that candidate and start over with all other candidates.
 - ii. If there are no winners eliminate the bottom candidate and repeat.

Round 1 – Threshold 1244 (Write Ins already eliminated)

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
249	456	418	385	354	27	94	465	38

Round 2 – Threshold 1243

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
249	460	419	394	357	0	96	468	41

Round 3 – Threshold 1242

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
255	464	425	396	367	0	100	476	0

Round 4 – Threshold 1237

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
261	478	436	416	385	0	0	496	0

Round 5 – Threshold 1237

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	536	470	458	448	0	0	549	0

Round 6 – Threshold 1204

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
----------	------------	--------	-------	----------	--------	---------	------------	-------

0	621	560	552	0	0	0	674	0
---	-----	-----	-----	---	---	---	-----	---

Round 7 – Threshold 1145

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	798	674	0	0	0	0	817	0

Round 8 – Threshold 1069

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	906	0	0	0	0	0	1230	0

D. Johnson is elected for the first seat.

Now the election starts over without D. Johnson.

Round 9 – Threshold 1232 (Write Ins already eliminated)

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
332	492	540	456	428	39	121	0	54

Round 10 – Threshold 1230

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
332	497	544	470	433	0	124	0	59

Round 11 – Threshold 1230

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
341	504	553	478	445	0	137	0	0

Round 13 – Threshold 1198

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	600	655	568	572	0	0	0	0

Round 14 – Threshold 1135

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	784	779	0	706	0	0	0	0

Round 15 – Threshold 1032

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	974	1099	0	0	0	0	0	0

So Frisch is elected to the second seat.

Result 3: D. Johnson and Frisch.

Method 4: One Vote Sequential Elimination (No Restart).

Process:

1. Count the top votes on everyone's ballot.
 - a. Check for winners (50+%).
 - i. For the first winner eliminate that candidate and continue with remaining candidate.
 - ii. If there are no winners eliminate the bottom candidate and repeat.

Round 1 – Threshold 1244 (Write Ins already eliminated)

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
249	456	418	385	354	27	94	465	38

Round 2 – Threshold 1243

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
249	460	419	394	357	0	96	468	41

Round 3 – Threshold 1242

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
255	464	425	396	367	0	100	476	0

Round 4 – Threshold 1237

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
261	478	436	416	385	0	0	496	0

Round 5 – Threshold 1237

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	536	470	458	448	0	0	549	0

Round 6 – Threshold 1204

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	621	560	552	0	0	0	674	0

Round 7 – Threshold 1145

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	798	674	0	0	0	0	817	0

Round 8 – Threshold 1069

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	906	0	0	0	0	0	1230	0

D. Johnson wins the first seat.

Round 9 – Threshold 776

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	1550	0	0	0	0	0	0	0

J. Johnson wins the second seat.

Result 3: D. Johnson and J. Johnson.

Method 5: Traditional Aspen Runoff.

Process:

1. Count the top two votes on everyone's ballot equally.
 - a. Check for winners (50+%).
 - i. If there are two winners the election is over.
 - ii. If there is exactly one winner, hold a single vote SE runoff between the second and third place candidates.
 - iii. If there are no winners eliminate take the top four candidates and run Method 2.

Round 1 – Threshold 1244 (Write Ins already eliminated)

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
538	717	696	777	735	102	268	884	138

Round 2 – Threshold 1201

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	930	0	1117	1117	0	0	1211	0

D. Johnson wins the first seat.

Round 3 – Threshold 1048

Kasabach	J. Johnson	Frisch	Torre	Behrendt	Lasser	Wampler	D. Johnson	Speck
0	0	0	1063	1032	0	0	0	0

Torre wins the second seat.

Result 5: D. Johnson and Torre.

For those really interested I simulated the elections in Maple and the text of the code can be seen below.

```
MinPos := proc(C)
  local m, pos, i;
  if nops(C)=0 then
    return empty;
  fi;
  m := 0;
  for i from 1 to nops(C) do
    if m = 0 then
      m := C[i];
      pos := i;
    fi;
    if C[i]>0 and C[i]<m then
      m := C[i];
      pos := i;
    fi;
  od;
  return pos;
end:

MaxPos := proc(C)
  local m, pos, i;
  if nops(C)=0 then
    return empty;
  fi;
  m := C[1];
  pos := 1;
  for i from 1 to nops(C) do
    if C[i]>m then
      m := C[i];
      pos := i;
    fi;
  od;
  return pos;
end:

SimCCElectionAlt1 := proc(V,m,L)
# 2 vote SE no reset
  local C, n, CR, i, j, v, k, p, S, q, b, Out;
  Out:= "";
  C := [0$m];
  CR := L;
  n := 0;
  S := 2;
  for i from 1 to nops(V) do
    b := 0;
    v := V[i];
    for j from 1 to nops(v) do
      if evalb(v[j] in CR) then
        C[v[j]] := C[v[j]]+1;
        n := n+1;
      fi;
    od;
  od;
end;
```

```

        b := b+1;
        if b = 2 then
            n := n-1;
            break;
        fi;
    fi;
od;
od;
print(n);
if max(C)>n/2 then
    p := MaxPos(C);
    print(C);
    print( "Candidate ", p, " is elected");
    C[p] := 0;
    if max(C)>n/2 then
        p := MaxPos(C);
        print("Candidate ", p, " is elected");
        return;
    fi;
    CR[p] := 0;
    SimMayElection(V,m,CR);
    return;
fi;
p := MinPos(C);
print("_____");
print(C);
print(cat("Candidate ", p, " is eliminated"));
CR[p] := 0;
SimCCElectionAlt1(V,m,CR);
end:

```

```

SimCCElectionAlt2 := proc(V,m,L)
# 1 vote SE, no reset
local C, n, CR, i, j, v, k, p, S, q, b, Out;
C := [0$m];
CR := L;
n := 0;
S := 2;
for i from 1 to nops(V) do
    v := V[i];
    for j from 1 to nops(v) do
        if evalb(v[j] in CR) then
            C[v[j]] := C[v[j]]+1;
            n := n+1;
            break;
        fi;
    od;
od;
print(n);
if max(C)>n/2 then
    p := MaxPos(C);
    print( "Candidate ", p, " is elected");
    C[p] := 0;
    if max(C)>n/2 then

```

```

    p := MaxPos(C);
    print( "Candidate ", p, " is elected");
    return;
fi;
CR[p] := 0;
SimMayElection(V,m,CR);
return;
fi;
p := MinPos(C);
print("_____");
print(C);
print("Candidate ", p, " is eliminated");
CR[p] := 0;
SimCCElectionAlt2(V,m,CR);
end:

```

```

SimCCElectionAlt3 := proc(V,m,L)
# 1 vote SE reset
local p, CR;
CR := L;
p := SimMayElection(V,m,CR);
CR[p] := 0;
SimMayElection(V,m,CR);
end:

```

```

SimCCElectionAlt4 := proc(V,m,L)
# Aspen Traditional
local C, n, CR, i, j, v, k, p, S, q, b;
C := [0$m];
CR := L;
n := 0;
S := 2;
for i from 1 to nops(V) do
b := 0;
v := V[i];
for j from 1 to nops(v) do
if evalb(v[j] in CR) then
C[v[j]] := C[v[j]]+1;
b := b+1;
n := n+1;
if b = 2 then
n := n-1;
break;
fi;
fi;
od;
od;
print("_____");
print("Initial Count");
print(C);
print(n);
while max(C)n/2 do
p := MaxPos(C);
print(cat("Candidate ", p, " wins"));

```

```

    C[p] := 0;
    CR[p] := 0;
    S := S-1;
od;
if S = 0 then return; fi;
if S = 1 then
    p := [0,0];
    p[1] := MaxPos(C);
    C[p[1]] := 0;
    p[2] := MaxPos(C);
    CR := [CR[p[1]], CR[p[2]], 0$(m-2)];
    SimMayElection(V,m,CR);
    return;
fi;
p := [0,0,0,0];
for i from 1 to 4 do
    p[i] := MaxPos(C);
    C[p[i]] := 0;
od;
for i from 1 to m do
    if not evalb(i in p) then
        CR[i] := 0;
    fi;
od;
SimCCElectionAlt4(V,m,CR);
end:

```

```

SimCCElection := proc(V,m,L)
    local C, n, CR, i, j, v, k, p, S, q, b;
    C := [0$m];
    CR := L;
    n := 0;
    S := 2;
    for i from 1 to nops(V) do
        b := 0;
        v := V[i];
        for j from 1 to nops(v) do
            if evalb(v[j] in CR) then
                C[v[j]] := C[v[j]]+1;
                b := b+1;
                n := n+1;
                if b = 2 then
                    n := n-1;
                    break;
                fi;
            fi;
        od;
    od;
    while max(C)n/2 do
        p := MaxPos(C);
        print("_____");
        print("winner");
        print(cat("Candidate", p, "wins"));
    end:

```

```

    C[p] := 0;
    CR[p] := 0;
    S := S-1;
od;
print("_____");
print("Initial Count");
print(C);
print(n);
if S = 0 then return; fi;
if S = 1 then
    p := [0,0];
    p[1] := MaxPos(C);
    C[p[1]] := 0;
    p[2] := MaxPos(C);
    CR := [CR[p[1]], CR[p[2], 0$(m-2)]];
    SimMayElection(V,m,CR);
    return;
fi;
p := [0,0,0,0];
for i from 1 to 4 do
    p[i] := MaxPos(C);
    C[p[i]] := 0;
od;
for i from 1 to m do
    if not evalb(i in p) then
        CR[i] := 0;
    fi;
od;
q := SimMayElection(V,m,CR);
CR := [0$m];
for i from 1 to m do
    for j from 1 to 4 do
        if p[j] = i and p[j] < q then
            CR[i] := i;
        fi;
    od;
od;
SimMayElection(V,m,CR);
end:

```

```

SimMayElection := proc(V,m,L)
local C, n, CR, i, j, v, k, p, c;
C := [0$m];
CR:= L;
while true do
    n := 0;
    for i from 1 to nops(V) do
        v := V[i];
        for j from 1 to nops(v) do
            if evalb(v[j] in CR) then
                C[v[j]] := C[v[j]]+1;
                n := n+1;
                break;
            fi;
        od;
    od;
end:

```

```

    od;
od;
if max(C)>n/2 then
  print("_____");
  print("winner");
  print(cat("Candidate ", MaxPos(C), " elected"));
  print(C);
  print(n);
  return MaxPos(C);
fi;
print("_____");
print("no winner");
print(C);
print(n);
p := MinPos(C);
print(cat("Candidate ", p, " eliminated"));
print(CR);
for i from 1 to nops(CR) do
  if CR[i]=p then
    CR[i]:=0;
  fi;
od;
C := [0$m];
od;
end:
SimCCElection(CCV3, 10, [seq(i,i=1..10)]);
SimCCElectionAlt1(CCV3, 9, [seq(i,i=1..9)]);
SimCCElectionAlt2(CCV3, 9, [seq(i,i=1..9)]);
SimCCElectionAlt3(CCV3, 9, [seq(i,i=1..9)]);
SimCCElectionAlt4(CCV3, 9, [seq(i,i=1..9)]);

```

Summary of how things work

CCV3 - All the votes for city council stored as a list ex. [3,4,1,2,8,9,5,6,7]

Each procedure is called with the same information, the votes, the number of candidates in the race, and a list [1, 2, .., 9] of the candidates left in the race

CityCouncilRace

```

SimCCElection(CCV3, 9, [1,2,3,4,5,6,7,8,9]) - Official Election Method
SimCCElectionAlt1(.....) - 2 vote SE no reset
SimCCElectionAlt2(.....) - 1 vote SE no reset
SimCCElectionAlt3(.....) - 1 vote SE reset
SimCCElectionAlt4(.....) - Traditional Apsen Method

```