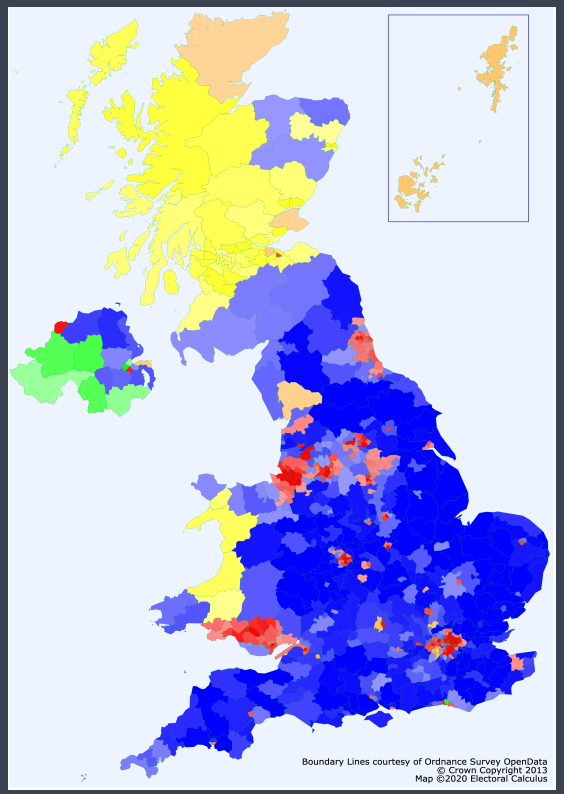
Using Algorithm X to re-analyse the last UK general election

Pydata Global 2020

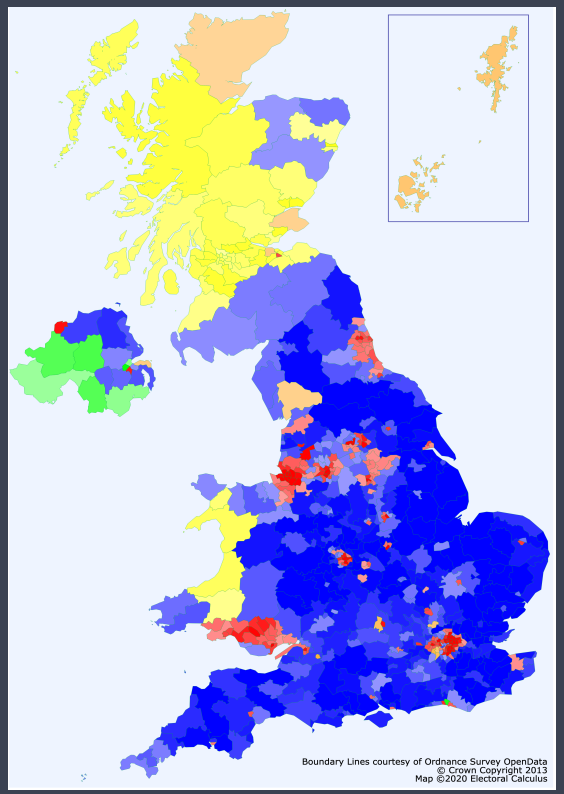
by Alex Glaser (AIGTech Ltd.)
on November 11th - 15th 2020

» Brief Intro to UK general elections



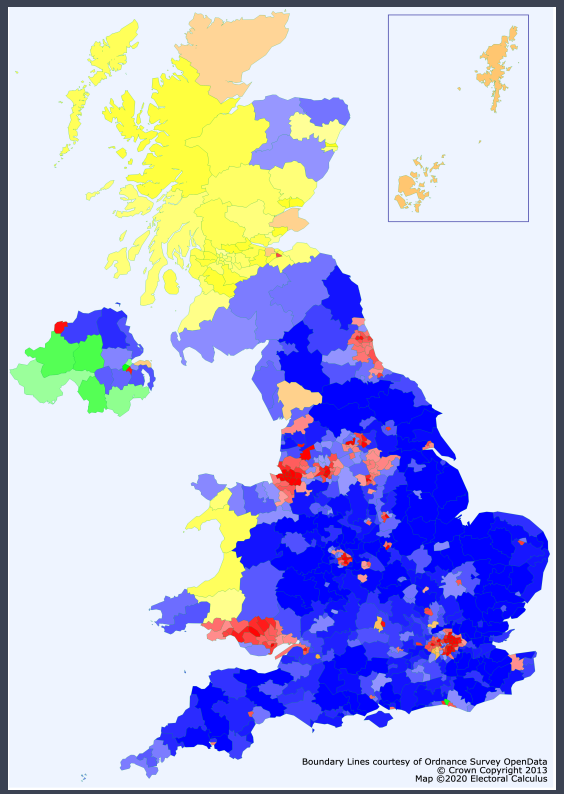
* House of Commons has 650 MPs

» Brief Intro to UK general elections



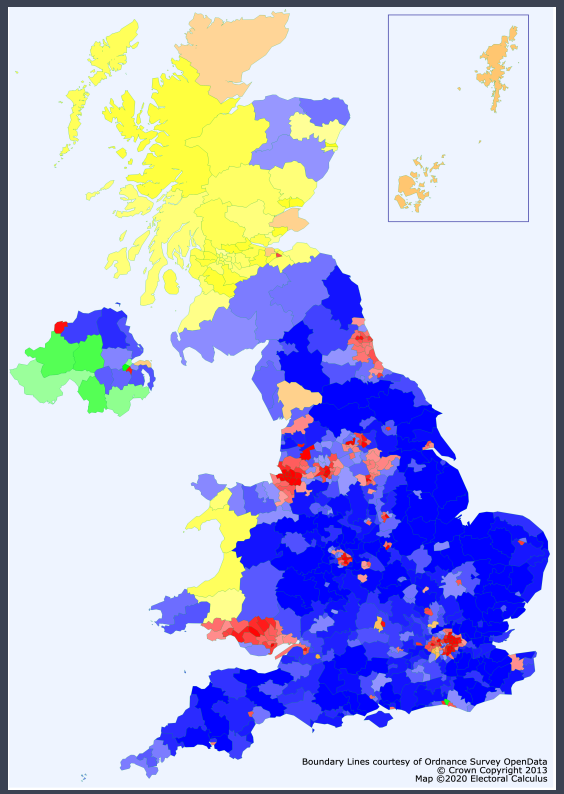
* House of Commons has 650 MPs
* UK is divided into 650 constituencies

» Brief Intro to UK general elections



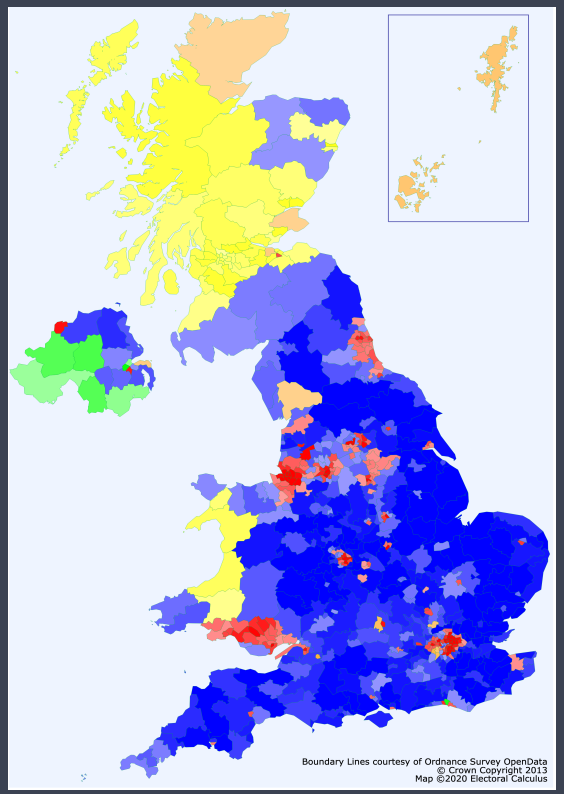
* House of Commons has 650 MPs
* UK is divided into 650 constituencies
* Each elector gets one vote

» Brief Intro to UK general elections



- House of Commons has 650 MPs
- UK is divided into 650 constituencies
- Each elector gets one vote
- Most votes wins the seat
  - First past the post method (FPTP)

» Brief Intro to UK general elections



- House of Commons has 650 MPs
- UK is divided into 650 constituencies
- Each elector gets one vote
- Most votes wins the seat
  - First past the post method (FPTP)
- Party with more than 325 seats forms government

First Past the Post (FPTP)

PROS

- Simple to understand
- Link to MP and their constituency

CONS

- Majority may disapprove of winner
- Not proportional

First Past the Post (FPTP)

PROS

- Simple to understand
- Link to MP and their constituency

CONS

- Majority may disapprove of winner
- Not proportional

	2019			
	Votes	Seats		
Conservatives	43.6%	365 (56%)		
Labour	32.6%	202 (31%)		

First Past the Post (FPTP)

PROS

- Simple to understand
- Link to MP and their constituency

CONS

- Majority may disapprove of winner
- Not proportional

	2019		2005	
	Votes	Seats	Votes	Seats
Conservatives	43.6%	365 (56%)	32.4%	
Labour	32.6%	202 (31%)	35.2%	

First Past the Post (FPTP)

PROS

- Simple to understand
- Link to MP and their constituency

CONS

- Majority may disapprove of winner
- Not proportional

	2019		2005	
	Votes	Seats	Votes	Seats
Conservatives	43.6%	365 (56%)	32.4%	196 (30%)
Labour	32.6%	202 (31%)	35.2%	

First Past the Post (FPTP)

PROS

- Simple to understand
- Link to MP and their constituency

CONS

- Majority may disapprove of winner
- Not proportional

	2019		2005	
	Votes	Seats	Votes	Seats
Conservatives	43.6%	365 (56%)	32.4%	196 (30%)
Labour	32.6%	202 (31%)	35.2%	355 (55%)

» Proportional Representation (PR)

Party	Total votes	Percentage of vote	Seats under PR	Seats now	difference under PR
Conservative	13905520	44.24%	288	363	-75
Labour	10282632	32.72%	213	203	10
Lib Dem	3,662,297	11.65%	76	11	65
SNP	1,242,380	3.95%	26	48	-22
Green Party	862,224	2.74%	18	1	17
Brexit Party	642,323	2.04%	13	0	13
DUP	244,127	0.78%	5	8	-3
Sinn Fein	181,853	0.58%	4	7	-3
Plaid Cymru	153,265	0.49%	3	4	-1
Alliance Party	134,115	0.43%	3	1	2
Social Democratic & Labour Party	118,737	0.38%	2	2	0
Total	31,429,473				

Wales Online 13th December 2019

Criticisms of calculation

- Too simplistic
- No link between MP and constituency

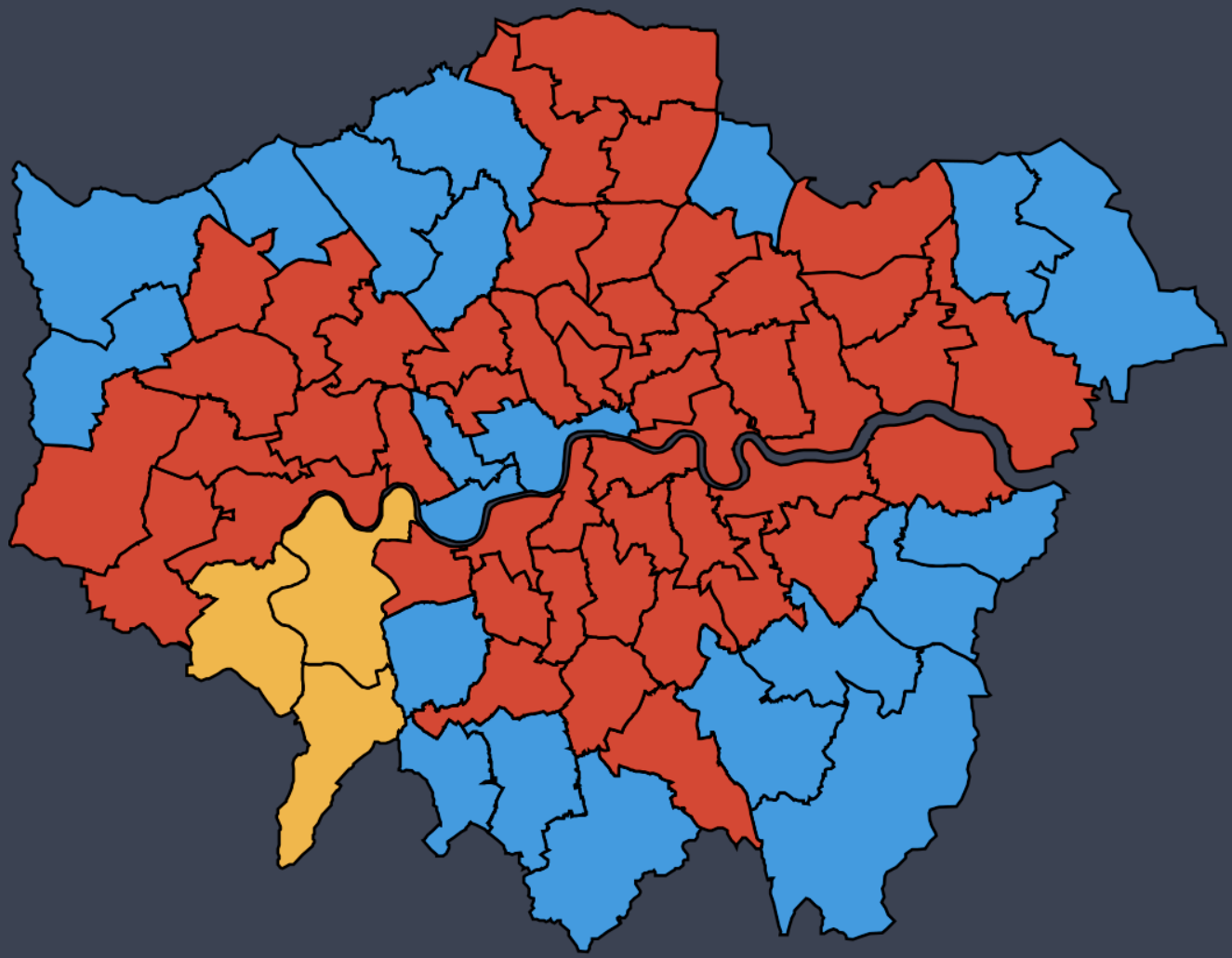
» Applying PR to the last UK general election

PR methods allocate multiple MPs to each political region.

- Keep the number of MPs the same
- Only use the results from 2019 general election, i.e. no inference or prediction
  - So cannot use e.g. Single Transferable Vote

For the UK this would require merging neighbouring constituencies but how?

» Constituency map of London



» Algorithm X

Algorithm used to solve 'Exact Cover' problems, for example:

We have the universe $U = 1, 2, 3, 4, 5, 6, 7$ and the collection of sets $S = A, B, C, D, E, F$, where

» Algorithm X

Algorithm used to solve 'Exact Cover' problems, for example:

We have the universe $U = 1, 2, 3, 4, 5, 6, 7$ and the collection of sets $S = A, B, C, D, E, F$, where

* A = {1,4,7}
* B = {1,4}
* C = {4,5,7}
* D = {3,5,6}
* E = {2,3,6,7}
* F = {2,7}

» Algorithm X

Algorithm used to solve 'Exact Cover' problems, for example:

We have the universe $U = 1, 2, 3, 4, 5, 6, 7$ and the collection of sets $S = A, B, C, D, E, F$, where

* A = {1,4,7}
* B = {1,4}
* C = {4,5,7}
* D = {3,5,6}
* E = {2,3,6,7}
* F = {2,7}

The answer is $\{B, D, F\}$

» Algorithm X

Another way of looking at the previous problem is to write is as a binary matrix:

	1	2	3	4	5	6	7
A	1	0	0	1	0	0	1
B	1	0	0	1	0	0	0
C	0	0	0	1	1	0	1
D	0	0	1	0	1	1	0
E	0	1	1	0	0	1	1
F	0	1	0	0	0	0	1

Sudoku

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Constraint matrix:

- 81×9 rows
- 81×4 columns

» Algorithm X

Problem is NP-complete.

» Algorithm X

Problem is NP-complete.

Algorithm X uses doubly linked lists and the dancing links method to efficiently remove and insert values

» Algorithm X

Problem is NP-complete.

Algorithm X uses doubly linked lists and the dancing links method to efficiently remove and insert values

$\longrightarrow$ speedy backtracking algorithm

» Applying Algorithm X to neighbouring constituencies

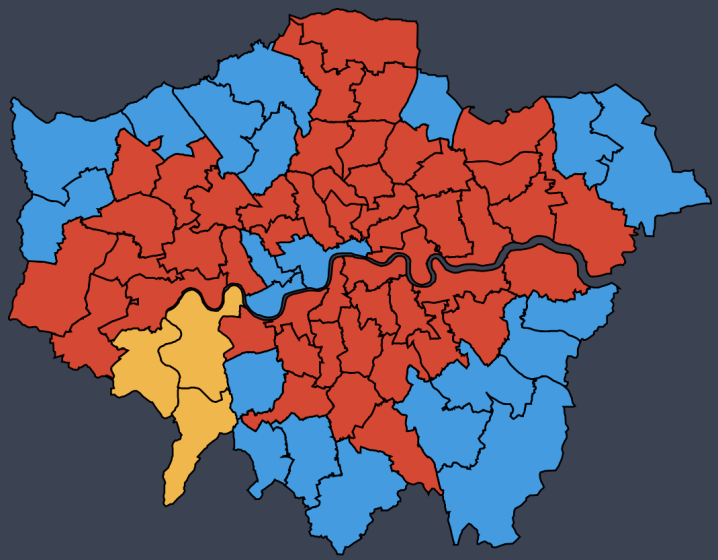
GeoPandas: use disjoint to see if constituencies are ‘neighbours‘.

Create a dataframe such as:

Set 1	This place	That place	The other
Set 2	This place	That place	Somewhere
Set 3	This place	The other	Somewhere
Set 4	That place	The other	Somewhere
Set 5	That place	Somewhere	Nowhere
Set 6	That place	The other	Nowhere

⋮ ⋮

» Solutions



For London with 70 constituencies we have about 540 million solutions taking roughly $3\frac{1}{2}$ hours.

Use of a PyPy kernel speeds this up by about 5-fold

» PR methods

Can be classified into two groups:

1. Highest averages (or Divisor) methods
   * After each round highest vote count allocated seat
   * Highest vote count then divided
   * Repeat until all seats allocated
2. Remainder methods
   * Divide votes by 'quota'
   * Allocate integer values to seats
   * Any remaining seats allocate on highest fractional remainder

» Highest Average method example

Electorate of 200,000 votes and 4 seats to allocate.

divisor = $\frac{votes}{seats+1}$ (D'Hondt method)

	Round 1	Round 2	Round 3	Round 4
Party A	70,000			
Seats	1			
Party B	50,000			
Seats	0			
Party C	46,000			
Seats	0			
Party D	34,000			
Seats	0			

» Highest Average method example

Electorate of 200,000 votes and 4 seats to allocate.

divisor = $\frac{votes}{seats+1}$ (D'Hondt method)

	Round 1	Round 2	Round 3	Round 4
Party A	70,000	35,000		
Seats	1	1		
Party B	50,000	50,000		
Seats	0	1		
Party C	46,000	46,000		
Seats	0	0		
Party D	34,000	34,000		
Seats	0	0		

» Highest Average method example

Electorate of 200,000 votes and 4 seats to allocate.

divisor = $\frac{votes}{seats+1}$ (D'Hondt method)

	Round 1	Round 2	Round 3	Round 4
Party A	70,000	35,000	35,000	35,000
Seats	1	1	1	2
Party B	50,000	50,000	25,000	25,000
Seats	0	1	1	1
Party C	46,000	46,000	46,000	23,000
Seats	0	0	1	1
Party D	34,000	34,000	34,000	34,000
Seats	0	0	0	0

» Highest Average method example

Other divisors:

$$D'Hondt = \frac{votes}{seats + 1}$$

$$Webster = \frac{votes}{2 \times seats + 1}$$

$$Imperiali = \frac{votes}{seats/2 + 1}$$

$$Huntington = \frac{votes}{\sqrt{seats(seats + 1)}}$$

Webster method is also known as the ”Sainte-Lague”

» Remainder methods

Electorate of 200,000 votes and 4 seats to allocate.

(Hare) quota = $\frac{\text{Total votes}}{\text{seats}}$ = 50,000

	Votes	Votes/ Quota	Quota Seats	Remainder	Remainder Seats	Total Seats
A	70,000					
B	50,000					
C	46,000					
D	34,000					

» Remainder methods

Electorate of 200,000 votes and 4 seats to allocate.

(Hare) quota = $\frac{\text{Total votes}}{\text{seats}}$ = 50,000

	Votes	Votes/ Quota	Quota Seats	Remainder	Remainder Seats	Total Seats
A	70,000	1.40	1	0.40		
B	50,000	1.00	1	0.00		
C	46,000	0.92	0	0.92		
D	34,000	0.68	0	0.68		

» Remainder methods

Electorate of 200,000 votes and 4 seats to allocate.

(Hare) quota = $\frac{\text{Total votes}}{\text{seats}}$ = 50,000

	Votes	Votes/ Quota	Quota Seats	Remainder	Remainder Seats	Total Seats
A	70,000	1.40	1	0.40	0	1
B	50,000	1.00	1	0.00	0	1
C	46,000	0.92	0	0.92	1	1
D	34,000	0.68	0	0.68	1	1

» Remainder methods

Electorate of 200,000 votes and 4 seats to allocate.

(Hare) quota = $\frac{\text{Total votes}}{\text{seats}}$ = 50,000

	Votes	Votes/ Quota	Quota Seats	Remainder	Remainder Seats	Total Seats
A	70,000	1.40	1	0.40	0	1
B	50,000	1.00	1	0.00	0	1
C	46,000	0.92	0	0.92	1	1
D	34,000	0.68	0	0.68	1	1

Can also have the Droop quota = $1 + \frac{\text{Total votes}}{1+\text{seats}}$ = 40,001

» Results

Gallagher index, measures relative disproportionality:

$$\sqrt{\frac{1}{2}\sum(\text{observed \%} - \text{expected \%})^2}$$

FPTP: 11.6

» Results

Gallagher index, measures relative disproportionality:

$$\sqrt{\tfrac{1}{2}\sum(\text{observed \%} - \text{expected \%})^2} \qquad \text{FPTP: 11.6}$$

	No. of merged seats		
Method	2	3	4
D'Hondt	9.6 (0.39)	8.0 (0.30)	7.7 (0.30)
Webster	6.1 (0.29)	5.3 (0.28)	4.3 (0.28)
Imperiali	12.3 (0.39)	11.4 (0.36)	10.5 (0.33)
Huntington	12.5 (0.38)	9.8 (0.33)	8.8 (0.29)
Hare	5.9 (0.34)	5.1 (0.30)	3.2 (0.30)
Droop	8.1 (0.34)	6.6 (0.30)	6.4 (0.32)

Table: Mean (and sd) Gallagher indexes for 10,000 simulations.

» Results

Gallagher index, measures relative disproportionality:

$$\sqrt{\tfrac{1}{2}\sum(\text{observed \%} - \text{expected \%})^2} \qquad \text{FPTP: 11.6}$$

	No. of merged seats		
Method	2	3	4
D'Hondt	9.6 (0.39)	8.0 (0.30)	7.7 (0.30)
Webster	6.1 (0.29)	5.3 (0.28)	4.3 (0.28)
Imperiali	12.3 (0.39)	11.4 (0.36)	10.5 (0.33)
Huntington	12.5 (0.38)	9.8 (0.33)	8.8 (0.29)
Hare	5.9 (0.34)	5.1 (0.30)	3.2 (0.30)
Droop	8.1 (0.34)	6.6 (0.30)	6.4 (0.32)

Table: Mean (and sd) Gallagher indexes for 10,000 simulations.

By comparison the metric for Ireland is 3.2 and Sweden has one of 0.63 using a modified Webster method.

» Results

Party	Total votes	Percentage of vote	Seats under PR	Seats now	difference under PR
Conservative	13905520	44.24%	288	363	-75
Labour	10282632	32.72%	213	203	10
Lib Dem	3,662,297	11.65%	76	11	65
SNP	1,242,380	3.95%	26	48	-22
Green Party	862,224	2.74%	18	1	17

» Results

Party	Total votes	Percentage of vote	Seats under PR	Seats now	difference under PR
Conservative	13905520	44.24%	288	363	-75
Labour	10282632	32.72%	213	203	10
Lib Dem	3,662,297	11.65%	76	11	65
SNP	1,242,380	3.95%	26	48	-22
Green Party	862,224	2.74%	18	1	17

Party	Simple	Hare	Actual
Conservatives	287	295	365
Labour	211	230	202
Liberal Democrat	76	65	11
SNP	26	28	48

Table: Comparison of Hare method with simple PR model shown earlier and actual results.

References, Thanks and Plugs

- Github github.com/alexiglaser/Constituency
- Algorithm X Wikipedia article
- Joel Lindop: FPTP and PR
- Bjarki Ágúst Guðmundsson: Algorithm X Python code
- Giovanni De Gasperis: Docker with PyPy kernel
- Too many members of the community to fit on this slide.

Final plug for meetups in London:

- Data Science Workshop
- Project Euler