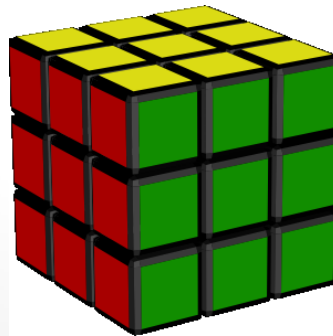# 2019 UK General Election

Data Analysis in Power BI
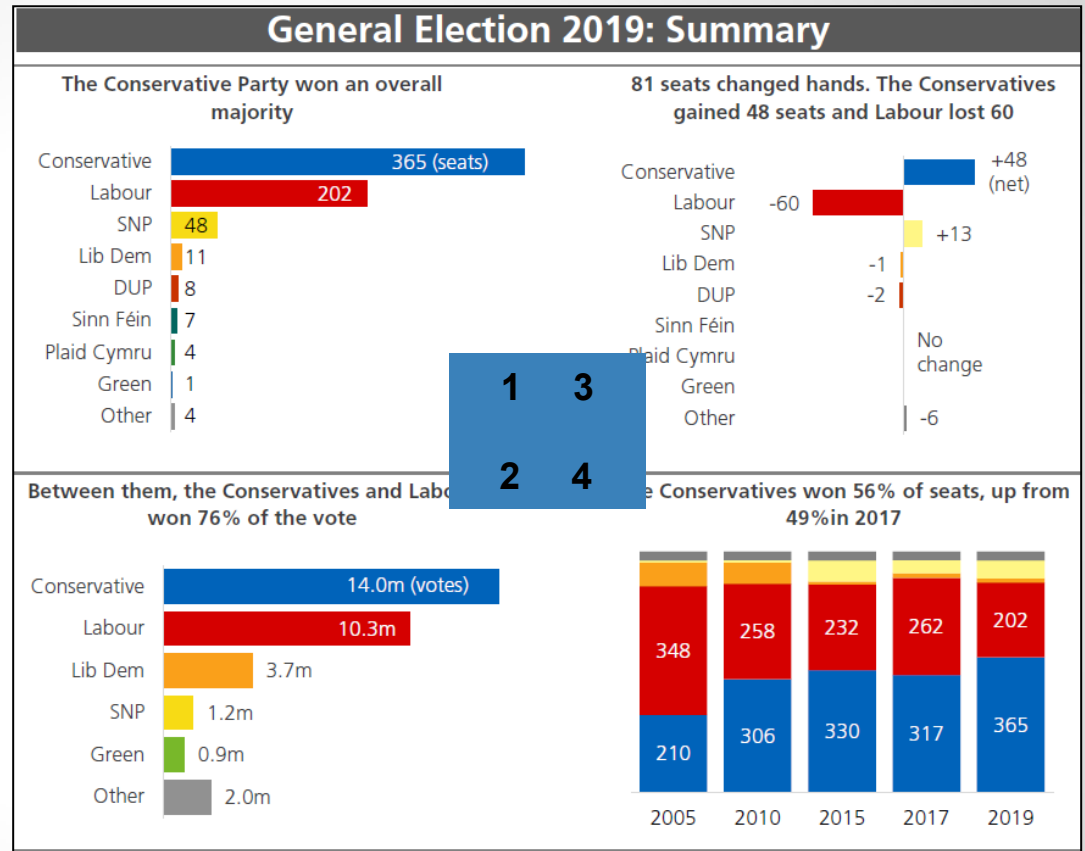
Overview

- The story: 5 whys?
- Data Model
- Easy DAX
- Hard DAX
- Visualisation
- Summary

Data sources

- https://researchbriefings.parliament.uk/ResearchBriefing/Summary/CBP-8749
- https://researchbriefings.parliament.uk/ResearchBriefing/Summary/CBP-7529

BeyondPowerBI.com

Story: Why did the Conservatives win?

General Election 2019: Summary

The Conservative Party won an overall majority

Party	Seats
Conservative	365 (seats)
Labour	202
SNP	48
Lib Dem	11
DUP	8
Sinn Féin	7
Plaid Cymru	4
Green	1
Other	4

81 seats changed hands. The Conservatives gained 48 seats and Labour lost 60

Party	Net change
Conservative	+48 (net)
Labour	-60
SNP	+13
Lib Dem	-1
DUP	-2
Sinn Féin	No change
Plaid Cymru	
Green	
Other	-6

1   3

2   4

Between them, the Conservatives and Labour won 76% of the vote

Party	Votes
Conservative	14.0m (votes)
Labour	10.3m
Lib Dem	3.7m
SNP	1.2m
Green	0.9m
Other	2.0m

The Conservatives won 56% of seats, up from 49% in 2017

Year	Labour	Conservative
2005	348	210
2010	258	306
2015	232	330
2017	262	317
2019	202	365

5 levels of Why?

1. They won most seats
2. They got most votes
3. They gained seats and Labour lost seats
4. Long term trend
5. Voter sentiment? – not recorded in the data presented

N.B. nothing specific about policies or personalities

Shaping the Data Model

- Star schema
  - **Fact**: VotesByCandidate
  - **Dimensions**: Constituency, Party names & 'Others'
  - **Disconnected**: selection of Measure & TopN Parties
- Calculated columns
  - Constituency candidates ranked by votes, hence winner.
- Measures
  - Usually preferred to calculated columns
  - Some checks on the data supplied
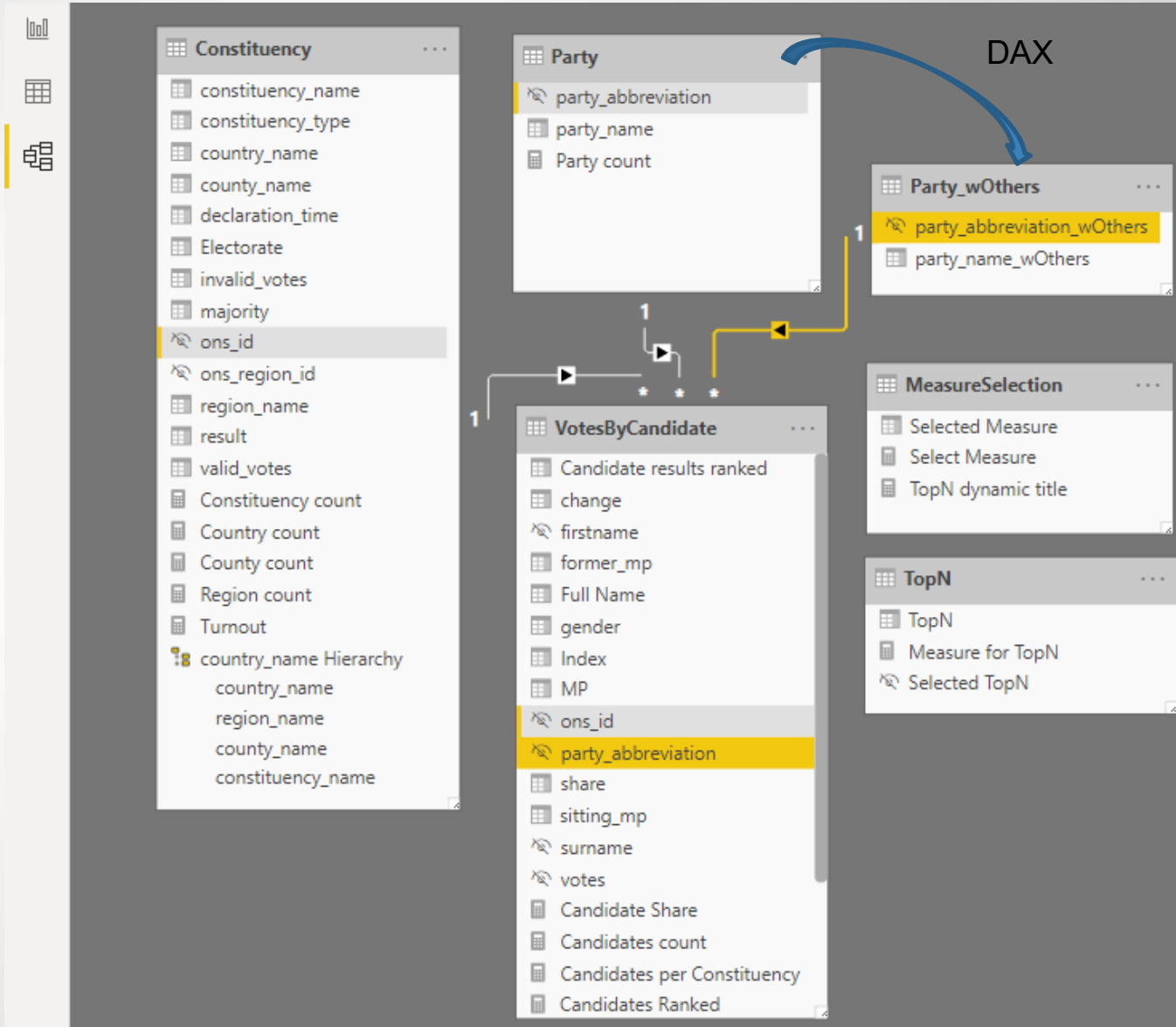- Other tables for data from previous Elections

Data Model

VotesByCandidate	Constituency
ons_id	**ons_id**
ons_region_id	ons_region_id
constituency_name	constituency_name
county_name	county_name
region_name	region_name
country_name	country_name
constituency_type	constituency_type
party_name	declaration_time
party_abbreviation	mp_firstname
index	mp_surname
firstname	mp_gender
surname	result
gender	first_party
sitting_mp	second_party
former_mp	electorate
votes	valid_votes
share	invalid_votes
change	majority
	con
Measures	lab
Candidates count	ld
Former MP count	brexit
Sitting MP count	green
Elected MP	snp
first_party	pc
second_party	dup
result in Const.	sf
Change in Votes share	sdlp
	uup
	alliance
	other
	other_winner

Red fields were
not imported

Data Model

Red fields were not imported

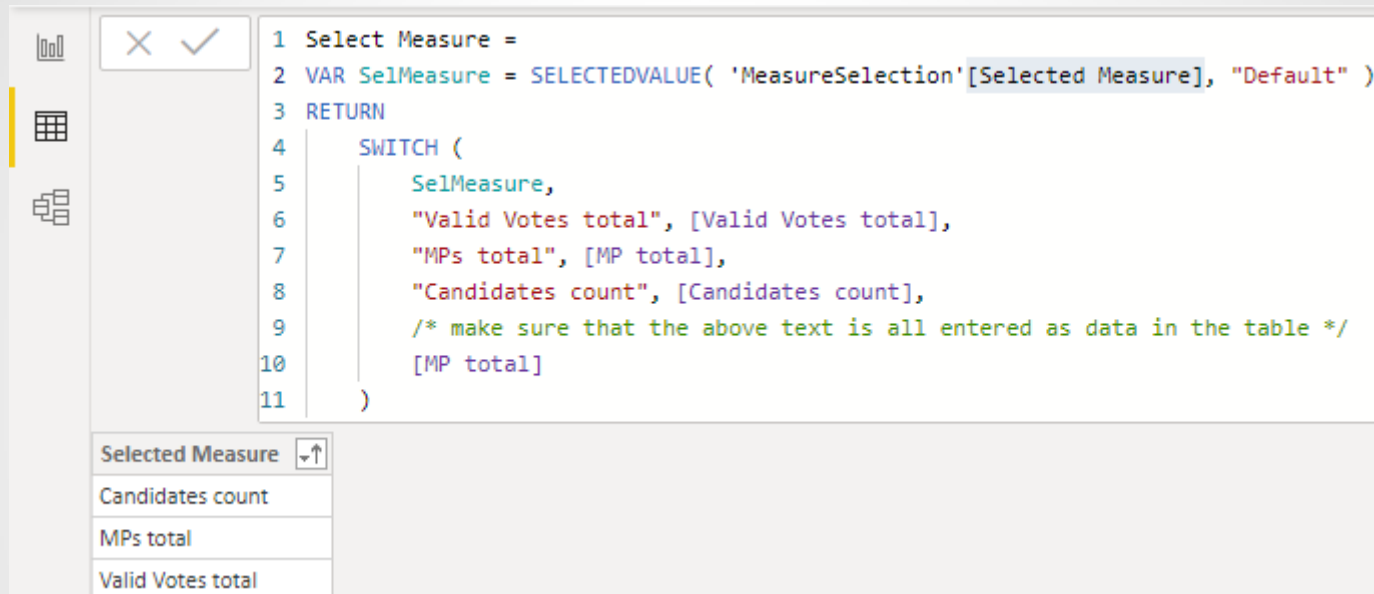# DAX for which candidate won

- Candidate results ranked

```
1  Candidate results ranked = RANKX (
2      FILTER (
3          'VotesByCandidate',
4          // [ons_id] identifies each Constituency
5          'VotesByCandidate'[ons_id] = EARLIER ( 'VotesByCandidate'[ons_id] )
6      ),
7      'VotesByCandidate'[votes], , 0
8  ) /* see https://www.red-gate.com/simple-talk/sql/bi/cracking-dax-the-earlier-and-rankx-functions/ */
```

- Winning MP (row context, in a calculated column)
  MP = IF('VotesByCandidate'[Candidate results ranked] = 1, 1, 0)

- Count MPs, depending on the filter context
  MP total = SUM('VotesByCandidate'[MP])

DAX for dynamic measures in slicers
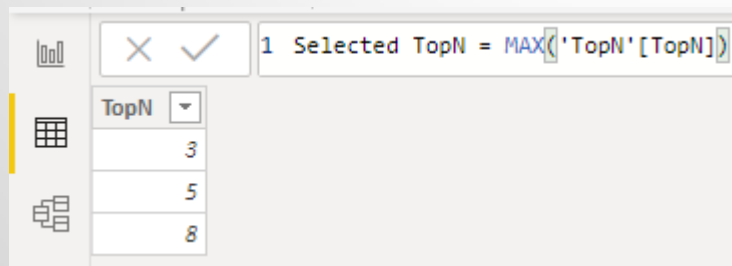
- ## Selected Measure

```
1  Select Measure =
2  VAR SelMeasure = SELECTEDVALUE( 'MeasureSelection'[Selected Measure], "Default" )
3  RETURN
4      SWITCH (
5          SelMeasure,
6          "Valid Votes total", [Valid Votes total],
7          "MPs total", [MP total],
8          "Candidates count", [Candidates count],
9          /* make sure that the above text is all entered as data in the table */
10         [MP total]
11     )
```

Selected Measure
Candidates count
MPs total
Valid Votes total

- **TopN dynamic title = SELECTEDVALUE('MeasureSelection' [Selected Measure],)**

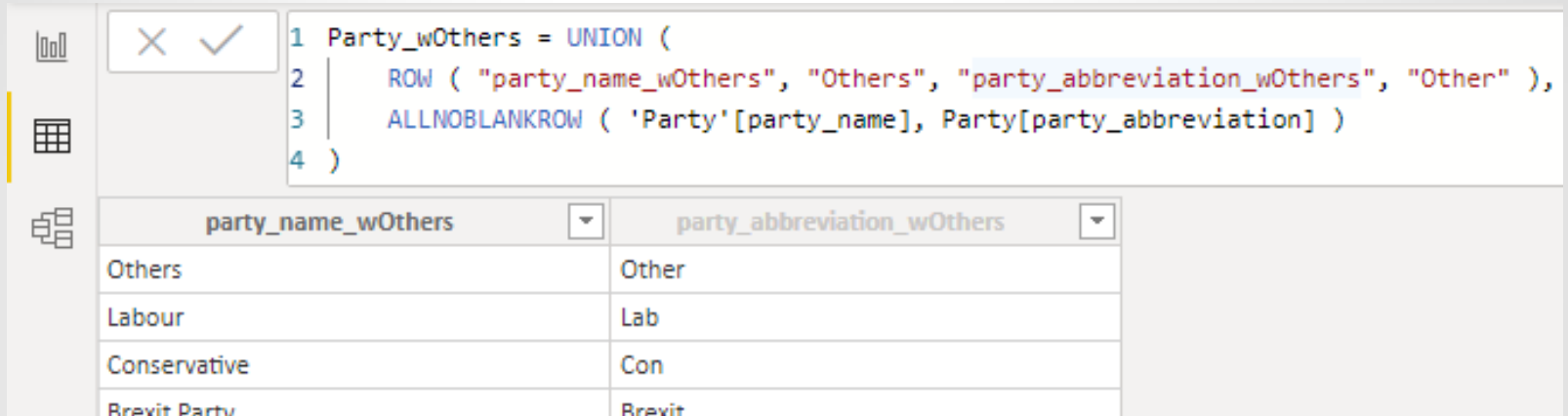- ## TopN (political parties) for the selected measure

```
1  Selected TopN = MAX('TopN'[TopN])
```

TopN
3
5
8

DAX for table 'Party with Others'

- Copy all of table 'Party', add a row for Others


```
1  Party_wOthers = UNION (
2      ROW ( "party_name_wOthers", "Others", "party_abbreviation_wOthers", "Other" ),
3      ALLNOBLANKROW ( 'Party'[party_name], Party[party_abbreviation] )
4  )
```

party_name_wOthers	party_abbreviation_wOthers
Others	Other
Labour	Lab
Conservative	Con
Brexit Party	Brexit

- This *new* dimension table filters the fact table 'VotesByCandidate' in the TopN measure(s)

DAX for TopN+Others

Thanks to Gerhard Brueckl

```dax
1  Measure for TopN =
2  /* get the items for which we want to calculate TopN + Others */
3  VAR Items =
4      SELECTCOLUMNS (
5          ALL (Party_wOthers ),
6          "RankItem", Party_wOthers[party_name_wOthers]
7      )
8  /* add a measure that we use for ranking */
9  VAR ItemsWithValue =
10     ADDCOLUMNS (
11         Items,
12         "RankMeasure", CALCULATE ( [Select Measure], ALL ( Party ) )
13     )
14 /* add a column with the rank of the measure within the items */
15 VAR ItemsWithRank =
16     ADDCOLUMNS (
17         ItemsWithValue,
18         "Rank", RANKX ( ItemsWithValue, [RankMeasure], [RankMeasure], DESC, DENSE )
19     )
20 /* calculate whether the item is a Top-item or belongs to Others */
21 VAR ItemsWithTop =
22     ADDCOLUMNS (
23         ItemsWithRank,
24         "TopOrOthers", IF ( [Rank] <= [Selected T
25     )

26 /* select the final items for which the value is calculated */
27 VAR ItemsFinal =
28     SELECTCOLUMNS (
29     /* we only select a single column to be used with TREATAS() in the final filter */
30         FILTER (
31             ItemsWithTop,
32             CONTAINSROW (
33                 VALUES ( Party_wOthers[party_name_wOthers] ), [TopOrOthers] )
34             /* need to obey current filters on _wOthers table. e.g. after Drill-Down */
35                 && CONTAINSROW ( VALUES ( Party[party_name] ), [RankItem] )
36         ),
37         /* need to obey current filters on base table */
38         "TopN_Others", [RankItem]
39     )
40 RETURN
41     CALCULATE (
42         [Selected Measure],
43         TREATAS ( ItemsFinal, Party_wOthers[party_name_wOthers] )
44     )
```

Visualisation

Barcharts for TopN with dynamic measures and titles
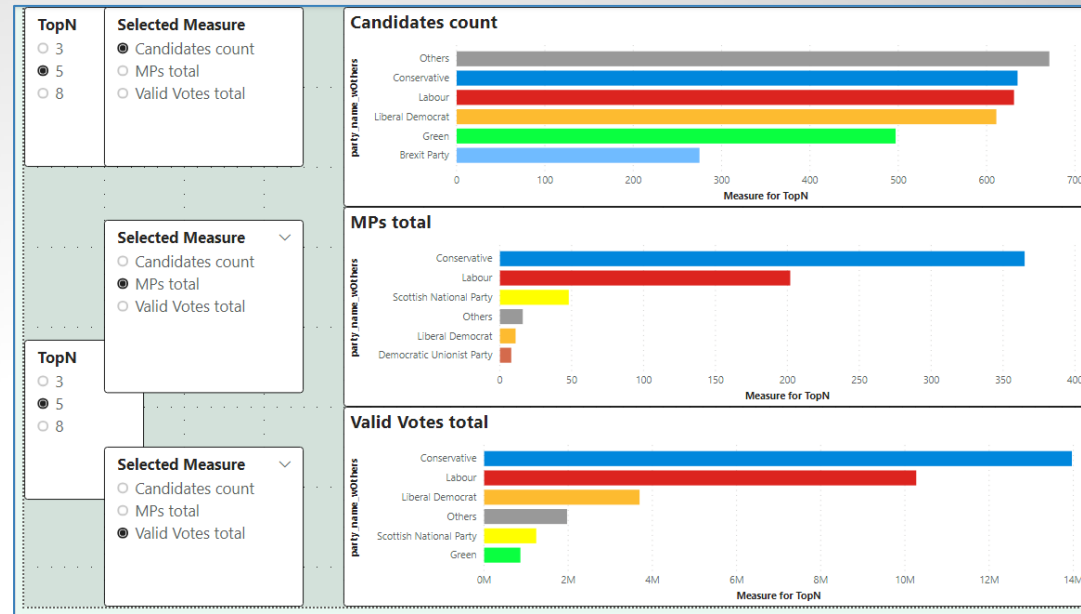
Matrices for details and for quality checks

Hierarchy: Country, Region, County, Constituency

country_name	Region count	County count	Constituency count	MP total	Valid Votes total	Electorate	Turnout	Candidates count
⊟ **England**	**9**	**46**	**533**	**533**	**26,911,657**	**39,901,035**	**67.7%**	**2,710**
⊟ **East**	**1**	**6**	**58**	**58**	**3,068,269**	**4,495,896**	**68.5%**	**277**
⊟ **Bedfordshire**	**1**	**1**	**6**	**6**	**314,996**	**467,503**	**67.6%**	**34**
Bedford	1	1	1	1	47,301	71,581	66.3%	5
Luton North	1	1	1	1	42,589	68,185	62.7%	7
Luton South	1	1	1	1	42,064	69,338	60.9%	7
Mid Bedfordshire	1	1	1	1	64,717	87,795	74.0%	6
North East Bedfordshire	1	1	1	1	65,018	90,678	71.9%	5
South West Bedfordshire	1	1	1	1	53,307	79,926	67.0%	4
⊟ **Cambridgeshire**	**1**	**1**	**7**	**7**	**409,488**	**589,833**	**69.7%**	**37**

Summary, Questions, Feedback

- Data Model
  - The usual guidance applied
- DAX
  - TopN+Others with dynamic measures
- Visualisation
  - Charts used the dynamic measures
  - Lots more potential for other measures and visualisations

P.S.
There's an error in the 2005 Election results in the official 2019 report.

Corrected: Labour 355, Conservative 198