World Inequality Lab – Technical Note N° 2021/01

Building the World Political Cleavages and Inequality Database: A New Dataset on Electoral Behaviors in 50 Democracies, 1948-2020

Amory Gethin
Clara Martínez-Toledano
Thomas Piketty

March 2021

Building the World Political Cleavages and Inequality Database: A New Dataset on Electoral Behaviors in 50 Democracies, 1948-2020

Amory Gethin

Clara Martínez-Toledano

Thomas Piketty*

March 2021

Abstract

This note provides technical details on the construction of the World Political cleavages and Inequality Database (WPID: see http://wpid.world), exploited in our book *Political Cleavages and Social Inequalities: A Study of 50 Democracies, 1948-2020*.

* Amory Gethin, Thomas Piketty: Paris School of Economics – World Inequality Lab; Clara Martínez-Toledano: Imperial College London.

1. Introduction

The World Political Cleavages and Inequality Database (WPID) is a new dataset on political cleavages in contemporary democracies, providing detailed information on the determinants of the vote and the structure of support for political parties across time and space. It is the outcome of a collective data harmonization effort involving over 20 researchers worldwide, whose main results are exposed in our book *Political Cleavages and Social Inequalities: A Study of 50 Democracies, 1948-2020*, published in French at Seuil/EHESS and in English at Harvard University Press.

The dataset, downloadable from http://wpid.world/resources, compiles results from electoral surveys covering over 500 elections since 1948. It is available in two main formats: a "macro database", corresponding to summary statistics on the vote for specific parties by a number of variables such as income, education or gender; and a "micro database" containing harmonized microfiles with data on the vote and on the sociodemographic characteristics of voters at the individual level (1,500,000 observations).

In this technical note, we briefly outline the methodology used to build the "macro database" (section 1), and we explain how we exploit data on income brackets and educational categories to derive estimates of the vote for specific parties by income decile and education decile (section 2; see section 3 for the Stata program implementing this transformation). For other details on data sources and methodology, please refer to the material exposed in the book and in the corresponding working papers (see http://wpid.world/resources).

For further questions, please contact wpid.world@gmail.com.

2. Construction of the WPID – Indicators on the structure of the vote by party

The WPID database provides data on the vote shares received by specific parties depending on a number of variables such as income, education, age, gender, or ethnoreligious affiliation. This dataset is obtained by aggregating raw data from electoral surveys and performing a few corrections. More precisely, we proceed as follows.

First, we compute simple sample averages of vote shares received by party and by variable for each electoral survey (or groups of surveys when combining several surveys for a given year or when computing statistics over decades).

Secondly, we reweigh these aggregated figures to match official election results by party. We do that by multiplying each value by the ratio of the official vote share received by the considered party (as recorded in official election results) to the survey sample average (which may overestimate or underestimate the actual vote share).[1] In the vast majority of electoral surveys, we are able to match all major parties reported in the survey with official election results, so as to cover 90% to 95% of the vote (the remaining 5-10% correspond to independents and other small parties). We exclude parties or candidates reported in surveys but not in election results (these inconsistencies arise in various cases, for instance when the survey is organized not immediately before or after the election), as well as parties receiving less than 1% of the vote.

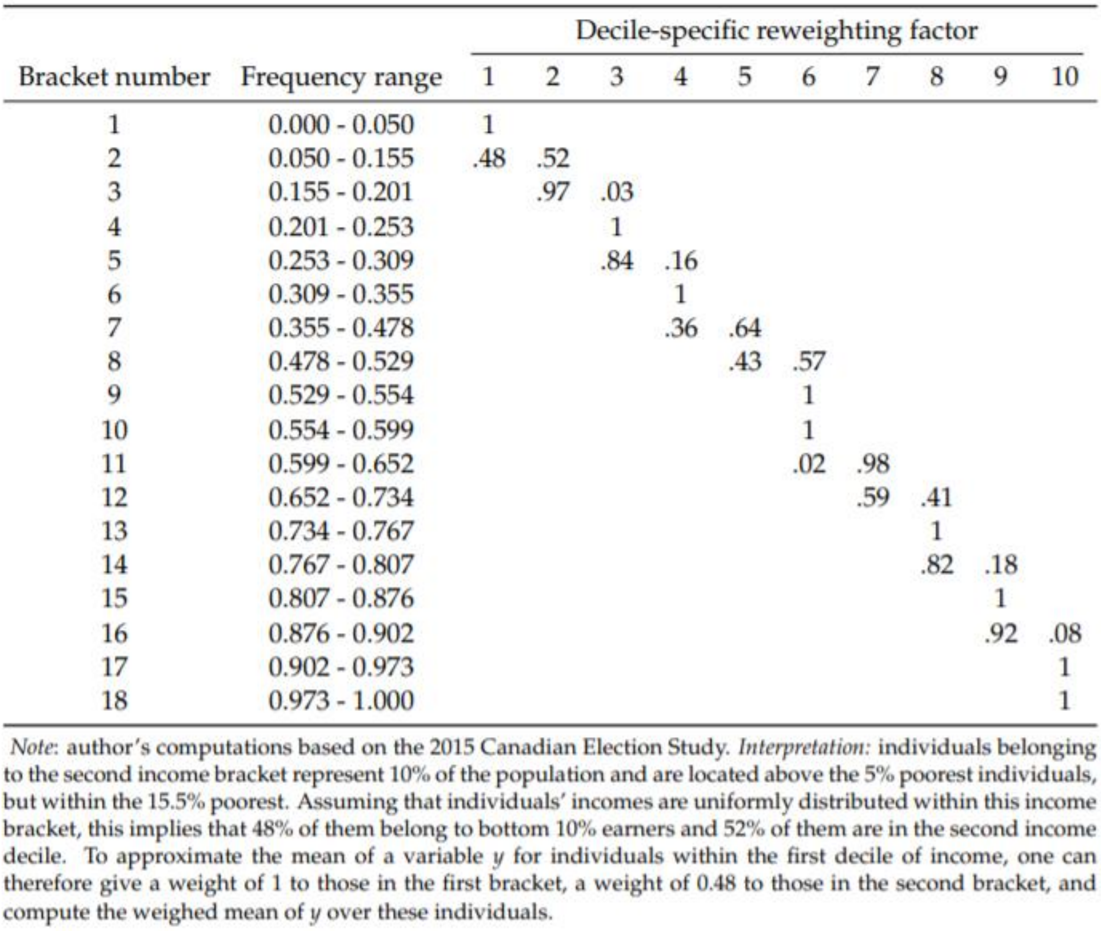
[1] As a rough approximation, we also reweigh the corresponding standard deviation and compute confidence intervals based on these "election-rescaled" sample averages and standard deviations.

3. Estimation of quantile groups from discrete categories

One of the contribution of the WPID is to provide data on the vote share received by specific parties and coalitions by income and education groups, decomposing for instance the population into its poorest or least educated half (the bottom 50%), the next 40% (the middle 40%), and the highest decile (the top 10%). Such groups are key to track political cleavages over time and compare them across countries. The problem is that existing surveys do not provide continuous values for income or education: these variables are most often coded in discrete categories (educational levels in the case of education, income brackets in the case of income).

To partially overcome this issue, we introduce a simple reweighing method, which exploits the distribution of individuals in each bracket or category to approximate quantiles. Consider for example the 2015 Canadian Election Study, which contains an income variable coded in eighteen brackets (see table 1). One is interested in computing the proportion of individuals belonging to the lowest income decile voting for the New Democratic Party $\bar{y}_{\{d=1\}}$, where y is a binary variable taking 1 is the respondent voted for the NDP and 0 otherwise, and where d refers to the income decile to which the respondents belong. Unfortunately, this is not directly possible with this income variable since only 5% of individuals belong to the first income bracket ($b = 1$), and 15.5% of them belong to the lowest two brackets ($b \in [1,2]$). If support for the NDP decreases linearly with income, then $\bar{y}_{\{b=1\}}$ will strongly overestimate $\bar{y}_{\{d=1\}}$, while $\bar{y}_{\{b=2\}}$ will strongly underestimate it since we are looking at individuals who are on average too poor in the first case and too rich in the second. However, it is easy to see that since individuals within the second bracket range from quantiles 0.05 to 0.155, this means that $\frac{0.05}{0.155-0.05} \approx 48\%$ of them belong to the bottom 10%, while 52% of them belong to the rest of the population, assuming for simplicity that individuals within brackets are uniformly distributed.

Table 1 - Reweighing categories to approximate quantiles: example for income brackets in Canada, 2015



		Decile-specific reweighting factor									
Bracket number	Frequency range	1	2	3	4	5	6	7	8	9	10
1	0.000 - 0.050	1									
2	0.050 - 0.155	.48	.52								
3	0.155 - 0.201		.97	.03							
4	0.201 - 0.253			1							
5	0.253 - 0.309			.84	.16						
6	0.309 - 0.355				1						
7	0.355 - 0.478				.36	.64					
8	0.478 - 0.529					.43	.57				
9	0.529 - 0.554						1				
10	0.554 - 0.599						1				
11	0.599 - 0.652						.02	.98			
12	0.652 - 0.734							.59	.41		
13	0.734 - 0.767								1		
14	0.767 - 0.807								.82	.18	
15	0.807 - 0.876									1	
16	0.876 - 0.902									.92	.08
17	0.902 - 0.973										1
18	0.973 - 1.000										1

Note: author's computations based on the 2015 Canadian Election Study. *Interpretation:* individuals belonging to the second income bracket represent 10% of the population and are located above the 5% poorest individuals, but within the 15.5% poorest. Assuming that individuals' incomes are uniformly distributed within this income bracket, this implies that 48% of them belong to bottom 10% earners and 52% of them are in the second income decile. To approximate the mean of a variable y for individuals within the first decile of income, one can therefore give a weight of 1 to those in the first bracket, a weight of 0.48 to those in the second bracket, and compute the weighed mean of y over these individuals.

Therefore, a reasonable approximation of the vote share received by the NDP among bottom 10% earners is a weighed average of vote shares in the two brackets:

$$\bar{y}_{\{d=1\}} = \frac{1 \times \bar{y}_{\{b=1\}} + 0.48 \times \bar{y}_{\{b=2\}}}{1 + 0.48}$$

This estimator is consistent, assuming that the average value taken by the dependent variable is constant within brackets. In practice, however, it does make sense to believe that the vote shares vary also within brackets in the same direction as observed between them. Therefore, this approximation should be considered as a lower bound of the true effect. Still, this method clearly does much better than computing deciles or quintiles directly from brackets – which could in fact not be quantile groups given that frequencies would necessarily be imbalanced.

Figure 1 - From brackets to deciles: vote for the New Democratic Party by income group in Canada, 2015

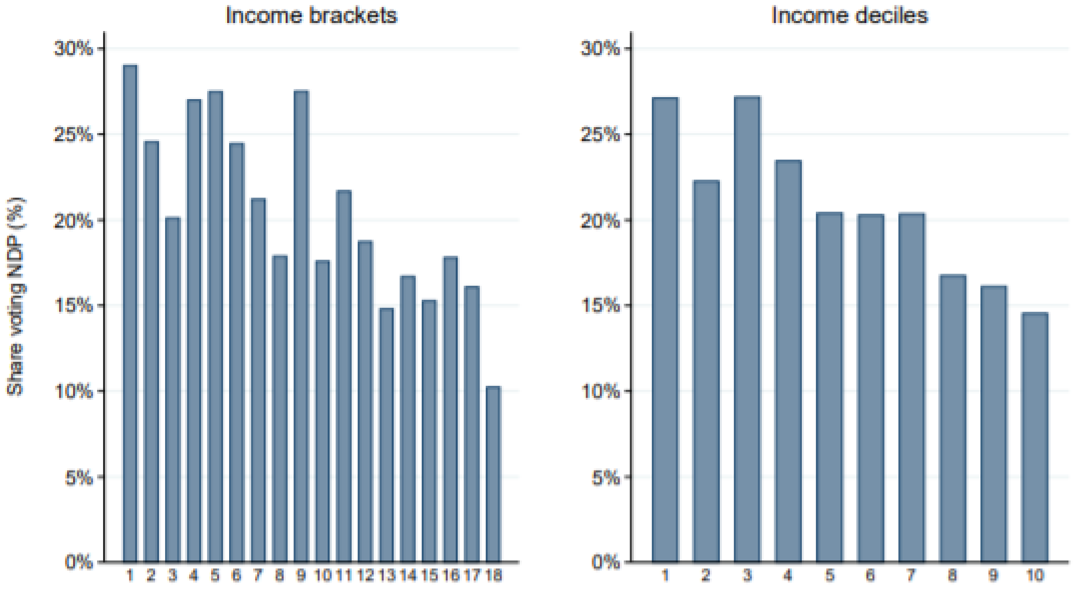


Figure 1 shows the results obtained when computing vote shares for the New Democratic Party in the 2015 Canadian national election. Unsurprisingly, the two pictures look very similar, since computing vote shares by decile amounts to computing weighed averages across income brackets.

Another interesting aspect of this method is that it enables us to control for structural changes not only in income, but also in other ordered variables such as education, wealth or even rural-urban scales. If university graduates were originally 5% in the 1960s and increased up to 30% in the 2010s, for instance, then one can exploit detailed educational categories to approximate "top 10% educated voters". In the 1960s, this category is composed of both university graduates and some secondary educated voters; in the 2010s, it gives more weight to individuals with masters or PhDs. This is what we do in the WPID.

Finally, one issue is that ‘splitting’ brackets into deciles implies that a single individual may belong to different quantile groups: in the example above, individuals in bracket 2 belong both to the first and the second deciles. While this is not problematic when computing averages, it makes regression models impossible to solve: without changing the dataset, one cannot compare the vote shares of the first and second decile with control variables.

To solve this problem, we expand the entire dataset as many times as the number of quantile groups required. In the case of deciles, for instance, the procedure consists in duplicating all observations ten times. Then, one simply needs to attribute the corresponding weights to duplicated individuals: individuals belonging to bracket 2 see their sample weight multiplied by 0.48 in their first observation, 0.52 in the second time they appear in the dataset, and 0 in all other instances. Since this process only reweighs individuals, it leaves the effect of other explanatory variables completely unchanged. Because we are increasing the number of observations in the dataset, however, normal standard errors will be downward-biased. We partially correct this issue by clustering standard errors by individual.

3. Stata code to generate quantiles groups from discrete categories

```
// ---------------------------------------------------------------------- //
// Program for expanding dataset
// ---------------------------------------------------------------------- //

// Wrapper program to expand the dataset and reweigh observations
cap program drop deciles
program define deciles, nclass
syntax, variable(namelist) by(namelist)

        // Generate identifier
        cap drop identifier
        bys `by': gen identifier=_n
        lab var identifier "Identifier (individual)"

        tempfile data
        save `data'

        // Get cumulated frequencies to create new decile weights
        gen x=1
        drop if mi(`variable')
        gcollapse (count) x [pw=weight], by(`by' `variable')
        sort `by' `variable' x
        bys `by': egen tot=sum(x)
        replace x=x/tot
        bys `by': replace x=sum(x)
        ren x freq
        drop tot
```

```
    bys `by': gen freq0=freq[_n-1] if _n>1
    bys `by': replace freq0=0 if _n==1
    order `by' y freq0 freq

    * First decile
    bys `by': gen d1=1 if freq<0.1 | _n==1
    bys `by': replace d1=(0.1-freq0)/(freq-freq0) if freq0<0.1 & freq>0.1 & _n!=1

    * Deciles 2 to 9
    forval d=2/9{
        local lower=(`d'-1)/10
        local upper=`d'/10
        bys `by': gen d`d'=1 if freq0>`lower' & freq<`upper' // decile in good bracket
        bys `by': replace d`d'=(freq-`lower')/(freq-freq0) if freq0<`lower' & freq>`lower' // & _n>1 // reweigh lower bracket
        bys `by': replace d`d'=(`upper'-freq0)/(freq-freq0) if freq0<`upper' & freq>`upper' // reweigh upper bracket

        bys `by': egen x=nvals(d`d') // when there is only one bracket for decile, fix value to one
        replace d`d'=1 if x==1
        drop x
    }

    * Upper decile
    bys `by': gen d10=1 if freq0>0.9 | _n==_N // decile in good bracket
    bys `by': replace d10=(freq-0.9)/(freq-freq0) if freq0<0.9 & freq>0.9 & _n!=_N // reweigh lower bracket
```

```
* Finally, distribute equally deciles with single bracket so that weights of brackets add up to 1
egen x=rowtotal(d*)
egen count=rcount(d*), cond(@==1)
forval d=1/10{
    replace d`d'=(1-(x-count))/count if d`d'==1
}
egen x2=rowtotal(d*)
assert inrange(x2,0.99,1.01)
drop x count x2

tempfile weights
save `weights'

// Duplicate dataset and merge with new weights by variable level
use `data', clear
gen id2=1
forval i=2/10{
    preserve
        use `data', clear
        gen id2=`i'
        tempfile temp
        save `temp'
    restore
    qui append using `temp'
}
merge m:1 `by' `variable' using `weights', nogen

// Reweigh and drop useless observations
forval d=1/10{
	replace weight=weight*d`d' if id2==`d' & !mi(`variable')
}
drop if mi(weight) & !mi(`variable')
drop if mi(`variable') & id2!=1

// Generate decile variable and decile dummies
forval d=1/10{
	gen d`variable'_`d'=(id2==`d') if !mi(`variable')
	lab var d`variable'_`d' "Decile `d' of `variable'"
}
cap drop d`variable'
gen d`variable'=.
forval d=1/10{
	replace d`variable'=`d' if d`variable'_`d'==1
}
lab var d`variable' "Decile of `variable'"

// Generate quintile variable and quintile dummies
gen q`variable'=1 if inlist(d`variable',1,2)
replace q`variable'=2 if inlist(d`variable',3,4)
replace q`variable'=3 if inlist(d`variable',5,6)
replace q`variable'=4 if inlist(d`variable',7,8)
replace q`variable'=5 if inlist(d`variable',9,10)
lab var q`variable' "Quintile of `variable'"
forval i=1/5{
	gen q`variable'_`i'=(q`variable'==`i') if !mi(`variable')
	lab var q`variable'_`i' "Quintile `i' of `variable'"
}

// Generate three broad groups
```

```
    gen g`variable'=1 if inrange(d`variable',1,5)
    replace g`variable'=2 if inrange(d`variable',6,9)
    replace g`variable'=3 if d`variable'==10
    lab var g`variable' "Groups of `variable'"
    label define g`variable' 1 "Bottom 50%" 2 "Middle 40%" 3 "Top 10%"
    label value g`variable' g`variable'
    forval i=1/3{
        gen g`variable'_`i'=(g`variable'==`i') if !mi(`variable')
    }
    lab var g`variable'_1 "Bottom 50% of `variable'"
    lab var g`variable'_2 "Middle 40% of `variable'"
    lab var g`variable'_3 "Top 10% of `variable'"

    // Generate Bottom 50% dummy
    gen b50=(inrange(d`variable',1,5)) if !mi(`variable')
    lab var b50 "Bottom 50% of `variable'"

    // Drop useless variable and add id2
    drop freq0 freq d1-d10
    lab var id2 "Secondary identifier (decile of `variable')"
end

// ------------------------------------------------------------------------ //
// Example of application
// ------------------------------------------------------------------------ //

sysuse auto, clear
drop weight
gen weight = 1
gen survey = "Auto"
```

```
graph bar price, over(headroom) name(g1, replace) t("Price by headroom space")
graph close g1

deciles, variable(headroom) by(survey)
graph bar price, over(dheadroom) name(g2, replace) t("Price by decile of headroom space")
graph close g2

graph combine g1 g2, ycommon
```