

2025 Swiss Stata Conference

Shapley value calculations: Implementation and illustrations

Philippe Van Kerm

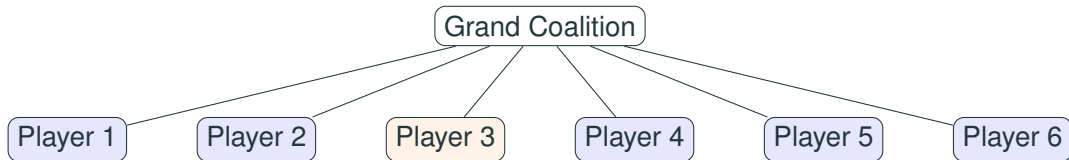
Bern – November 21, 2025

The Shapley and Owen values

Implementation with `shapowen`

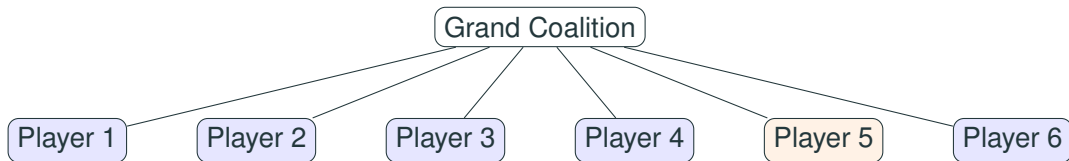
The Shapley value in a nutshell

What is the contribution of an individual element i to the aggregate “value” collectively created by a set of elements $N = \{1, 2, \dots, N\}$?



The Shapley value in a nutshell

What is the contribution of an individual element i to the aggregate “value” collectively created by a set of elements $N = \{1, 2, \dots, N\}$?



Diverse applications possible

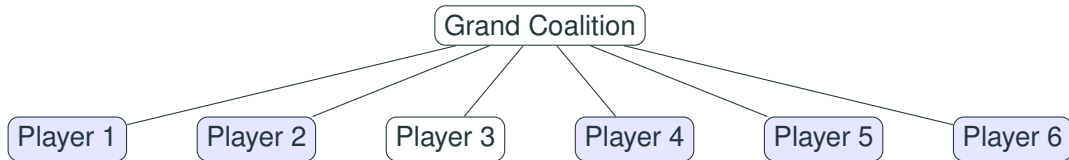
- Bargaining power of political parties in coalition formation
- Cost sharing: payment of individual users to total cost functions (cf. the “airport problem”)
- Contribution of individual covariates to a regression model's R^2
- Contribution of individual predictors to a machine learning predictive or classification model (“ShAP”)
- Contribution of different sources of income or groups of individuals to inequality in household income (many applications to income distribution analysis (Shorrocks, 2013))
- ...

The Shapley value formula

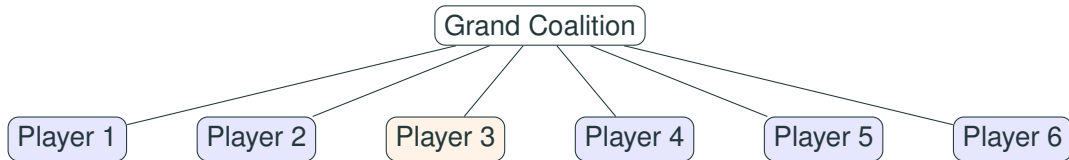
- A set of elements ('players') $N = \{1, 2, \dots, n\}$. N is the "grand coalition"
- A characteristic function $v : 2^N \rightarrow \mathbb{R}$ returns the collective output of any coalition formed by elements of N
- The Shapley value (Shapley, 1953) for element i is given by

$$\phi_i = \phi(i; v, N) = \underbrace{\sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!}}_{\text{weighted sum across coalitions}} \underbrace{(v(S \cup \{i\}) - v(S))}_{\text{marginal contribution to coalition } S}$$

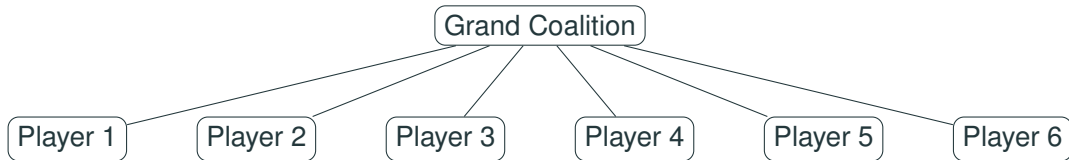
The Shapley value formula



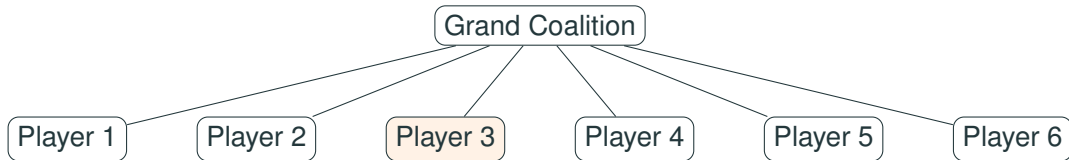
The Shapley value formula



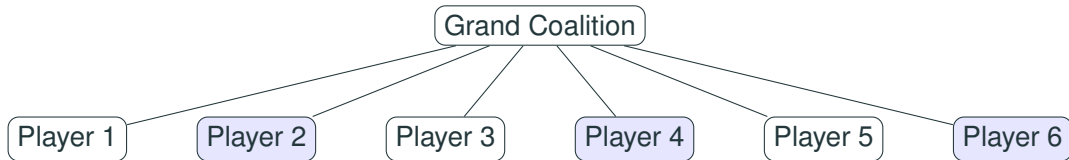
The Shapley value formula



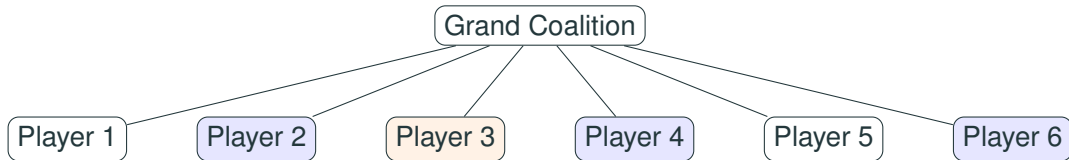
The Shapley value formula



The Shapley value formula



The Shapley value formula



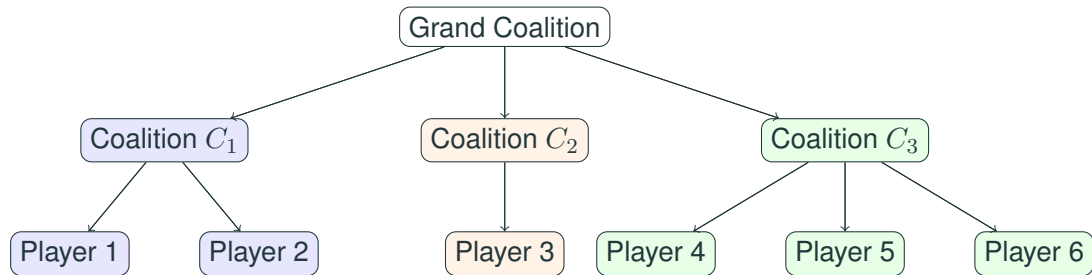
Nested structures: The Owen value (Owen, 1977)

Consider now preset (sub-)coalitions within the ‘grand coalition’: $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ is a partition of N into disjoint coalitions

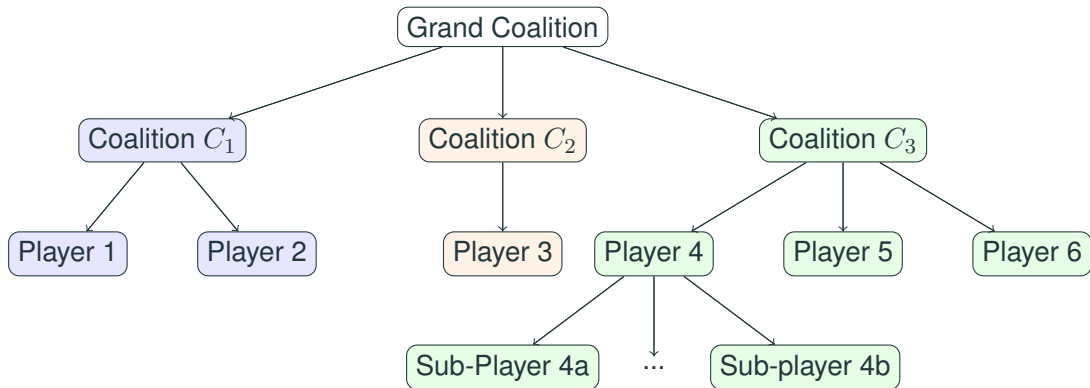
For example

- blocks of ‘similar’ covariates (e.g., demographic attributes, labour market characteristics and regional environment)
- blocks of ‘similar’ political parties (e.g., left, center, right?)
- blocks of ‘similar’ sources of income (public transfers vs market incomes)
- ...

Nested structures: The Owen value (Owen, 1977)



Nested structures: The Owen value (Owen, 1977)



Nested structures: The Owen value (Owen, 1977)

The Owen value is the composition of Shapley values for each preset sub-coalition

$$\psi_i(v, \mathcal{C}) = \phi_i(v_{C_j}) \cdot \phi_{C_j}(v^{\mathcal{C}})$$

- Recursive additivity: the contributions of each player in a sub-coalition add up to the contribution of the sub-coalition to the grand coalition
- Evaluate the Shapley value of a player in a sub-coalition for all possible coalitions of sub-coalitions...
- ... for any level of nested structures

The Shapley and Owen values

Implementation with `shapowen`

shapowen – Simplified syntax

shapowen evaluates Shapley and Owen values for **arbitrary** Stata instructions provided these

1. take input 'players' specification as (some form of) a *list*
2. return evaluation in `r()` or `e()` scalars or matrices (or functions thereof)

Simplified syntax diagram

```
shapowen list-of-items
```

```
[ , scalarexpressions(string) matrixexpressions(string) substitution(string) ... ]:  
cmd ... @ ...
```

(Syntax borrowed from the package `shapley` available on SSC (Kolenikov, 2000).)

Nested list of items

Nested structures are specified in *list-of-items* by grouping items within brackets, e.g.:

(a b) c (d e f)

or

a (b c (d e f) (g h) i) ((j k l) m) n "o p q"

NB: grouping by double-quotes forms an unbreakable item – here o p q are never evaluated separately.

- Simple (non-nested) Shapley value calculations are relatively straightforward to implement – `shapowen` mainly does ‘bookkeeping’ on behalf of the user
- Nested structures and Owen values *are* (very) significantly more difficult to deal with
 - » `shapowen` leverages “advanced” Mata features such as classes (objects), structures, pointers and recursivity
- Speed is potentially an issue: the number of evaluations increases exponentially with the number of ‘players’
 - » *cmd* needs to be fast or n needs to be relatively small

Addressing speed: approximation algorithms

- Approximation algorithms: evaluate marginal contributions $(v(S \cup \{i\}) - v(S))$ on a **random subset** of all potential coalitions
- `shapopen` implements a permutation-based algorithm:
 - » Shapley value is *also* the average marginal contribution of i across all possible *permutations* of elements in the grand coalition
 - » Monte Carlo approximation:
 1. take a random permutation of players/elements and evaluate marginal contributions in that sequence
 2. (optionally) evaluate marginal contributions on the inverse permutation (antithetic sampling)
 3. repeat for a (large) number of permutations
 4. approximate Shapley value of each element as average of their marginal contributions across permutations

Example 1: Ascribing covariate contributions to a regression's R^2

```
. svy : regress lnY c.age##c.age i.educ i.sex i.typehh2
```

(running regress on estimation sample)

Survey: Linear regression

Number of strata	=	1	Number of obs	=	6,715
Number of PSUs	=	6,715	Population size	=	11,448,293
			Design df	=	6,714
			F(9, 6706)	=	203.00
			Prob > F	=	0.0000
			R-squared	=	0.2679

lnY	Linearized					
	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
age	.0365523	.0028338	12.90	0.000	.0309972	.0421074
c.age#c.age	-.0003494	.0000255	-13.73	0.000	-.0003993	-.0002995
educ						
[2]medium	.2379809	.0209922	11.34	0.000	.1968296	.2791323
[3]high	.4566599	.0210029	21.74	0.000	.4154875	.4978322
sex						
[2]female	-.036143	.0142383	-2.54	0.011	-.0640546	-.0082315
typehh2						
Couple only	.3099867	.016799	18.45	0.000	.2770553	.3429181
Couple with child(ren) and possibly others	.236953	.019542	12.13	0.000	.1986444	.2752616
Single with child(ren) and possibly others	-.0790149	.0229858	-3.44	0.001	-.1240743	-.0339555
Other configuration	.2094785	.0422841	4.95	0.000	.1265883	.2923688
_cons	8.95386	.0766612	116.80	0.000	8.80358	9.10414

Example 1: Ascribing covariate contributions to a regression's R^2

```
. shapowen c.age##c.age i.educ i.sex i.typehh2 , scalar(e(r2)) : ///
```

```
>          svy : regress lnY @
```

```
.....
```

```
Instruction:  svy : regress lnY @
```

```
Items list:   c.age##c.age i.educ i.sex i.typehh2
```

```
--- e(r2) ---
```

```
Empty set value:      0
```

```
Full set value:  .2678879
```

	Nominal contribution				Relative contribution			
	Shapley -Owen	Banzhaf -Owen	First	Last	Shapley -Owen	Banzhaf -Owen	First	Last
FULL	.2678879				1			
c.age##c.age	.0430312	.0419648	.0613163	.0290116	.1606314	.1566507	.2288881	.1082975
i.educ	.1329032	.1316128	.1627722	.1081956	.496115	.4912983	.6076132	.4038838
i.sex	.0069608	.0068213	.0132664	.0012132	.025984	.0254633	.0495222	.0045286
i.typehh2	.0849927	.0836965	.1113913	.0637788	.3172696	.312431	.4158132	.2380803

Further examples

```
shapowen c.age##c.age i.educ i.typehh2 , scalar(_b[2.sex] e(N)) : ///
        svy : regress lnY i.sex @

shapowen i.educ i.typehh2 i.sex c.age##c.age , scalar(e(r2)) separator(##) : ///
        svy : regress lnY @

shapowen Lh Lo K PB T 0 , scalar(r(coef)) emptyvalue(0) : ///
        sgini @ [aw=hpwgt] , source

shapowen Lh Lo K PB T 0 , scalar(r(coef)) emptyvalue(0) approx(10 , anti) : ///
        sgini @ [aw=hpwgt] , source

shapowen ((Lh Lo) K) ((P B) T) 0 , emptyvalue(0) scalar(r(coef)) : ///
        sgini @ [aw=hpwgt] , source

shapowen Lh Lo K PB T 0 , scalar(r(coef)) ///
        substitution(mn_Lh mn_Lo mn_K mn_PB mn_T mn_0) : ///
        sgini @ [aw=hpwgt] , source

shapowen 1 2 3 4 5 , scalar(r(coef)) sep(,) emptyvalue(0) : ///
        sgini Yalt [aw=hpwgt] if inlist(typehh2,@)
```


Note: Leveraging A. Naqvi's treemap for visualising nested structure

```
shapowen (i.race i.collgrad c.age##c.age) (i.industry i.occupation) (i.south i.smsa) ///  
  , sca(e(r2_a)) treemap : regress lnw @
```



Using a wrapper program

```
. cap pr drop changini
. pr def changini , rclass
1.      tempvar p rw
2.      svy : logit period '0'
3.      qui predict 'p' , rules
4.      qui gen 'rw' = cond(period==1 , 1 , 'p'/(1-'p'))
5.      sgini Yalt [aw=hpwgt*'rw'] if period==0
6.      return scalar gini = r(coeff)
7. end
```

Using a wrapper program

```
. shapowen i.typehh2 c.shearn i.educ i.sex , scal(r(gini)) : ///  
> changini @
```

.....

Instruction: changini @

Items list: i.typehh2 c.shearn i.educ i.sex

--- r(gini) ---

Empty set value: .2714685

Full set value: .2521056

	Nominal contribution				Relative contribution			
	Shapley -Owen	Banzhaf -Owen	First	Last	Shapley -Owen	Banzhaf -Owen	First	Last
FULL	-.019363				1			
i.typehh2	.0008653	.0012075	.001221	-.000859	-.0446902	-.0623614	-.0630602	.0443643
c.shearn	-.0191425	-.0186721	-.0211986	-.0189679	.9886135	.9643199	1.094803	.9795988
i.educ	-.0032951	-.0028207	-.0050923	-.0033955	.1701762	.1456759	.2629924	.1753611
i.sex	.0022093	.0026497	.0024182	.0002386	-.1140995	-.1368462	-.12489	-.0123222

Bootstrap inference

```
bootstrap ///  
  diff=el(r(Shap0w),1,1) ///  
  a=el(r(Shap0w),2,1) ///  
  b=el(r(Shap0w),3,1) ///  
  c=el(r(Shap0w),4,1) ///  
  d=el(r(Shap0w),5,1) ///  
  , reps(499) nodots : ///  
  shapowen i.typehh2 c.shearn i.educ i.sex , scal(r(gini)) : ///  
    changini @
```

- The Shapley value and Shapley value decompositions and their Owen counterparts for nested structures have many potential applications
- `shapowen` (available on SSC) facilitates their calculation with a generic prefix-based syntax and flexible input processing

Comments, feedback and suggestions welcome!

References i

Kolenikov, S. (2000), 'Shapley: Stata module to perform additive decomposition of sample statistic'.

URL: <https://EconPapers.repec.org/RePEc:boc:bocode:s411401>

Owen, G. (1977), Values of games with a priori unions, *in* R. Henn and O. Moeschlin, eds, 'Mathematical Economics and Game Theory', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 76–88.

Shapley, L. S. (1953), A value for n-person games, *in* H. Kuhn and A. W. Tucker, eds, 'Contributions to the Theory of Games, Vol. II', Princeton: Princeton University Press, Princeton, NJ, pp. 307–317.

Shorrocks, A. F. (2013), 'Decomposition procedures for distributional analysis: a unified framework based on the Shapley value', *Journal of Economic Inequality* **11**(1), 99–126.

URL: <http://dx.doi.org/10.1007/s10888-011-9214-z>