

WISCONSIN NATIONAL
DATA CONSORTIUM

WiNDC Update

Adam Christensen
Martha Loewe
Thomas Rutherford
July 22, 2020



Quick Overview

- Data updates
- Data Interface updates

Why?

- Because data changes
- Need to to invent a methodology to ensure robustness





Data Philosophy

- Original source data changes asynchronously
 - ... format of data also changes
- WiNDC release versions offer a full (i.e., all sources) data snapshot in time
 - API connections are great, but not all sources have this capability,
 - Data changes might impact solver → WiNDC tries to isolate the user from these issues
 - We test all the builds with new data releases
 - We still need to automate these tests, but requires development
- Need to be able to track this data over time, easily, and enable comparisons
- ***WiNDC 2.01 and WiNDC 2.1 are available now!***

**BIG
SALE**



WiNDC 2.01 - Some Details

- A full dataset is available for years 1997-2016
- Data mostly approximates the data released in WiNDC 2.0
- New Methodology on Census Data for State Government Finances
 - Relies on Census “Datasets” as the original data format rather than “Summary Tables”
 - <https://www.census.gov/programs-surveys/state/data/datasets.html>
 - Could enable finer grained mappings to WiNDC categories
 - Pain to process, but the datasets are more machine readable



WiNDC 2.01 - Some Details

- A full dataset is available for years 1997-2016
- Data mostly approximates the data released in WiNDC 2.0
- New Methodology on Census Data for State Government Finances
 - Relies on Census “Datasets” as the original data format rather than “Summary Tables”
 - <https://www.census.gov/programs-surveys/state/data/datasets.html>
 - Could enable finer grained mappings to WiNDC categories
 - Pain to process, but the datasets are more machine readable

```
00state35.txt
0100000000000019H 1740552000024
0100000000000019T 1769177000024
0100000000000019X 957310000024
0100000000000021G 482950000024
0100000000000024G 497206000024
0100000000000024T 227245000024
0100000000000024X 197220000024
0100000000000031G 428350000024
0100000000000031X 319850000024
0100000000000034G 139666000024
0100000000000034T 206437000024
```



WiNDC 2.1 - Some Details

- A full dataset is available for years 1997-2017
- All data back to 1997 was reloaded from all the original sources
 - Data as far back as 1999 is still being updated (but most data that is updated is after 2014)
 - Most changes were to BEA, Census data
- Added in NASS Ag Census 2017 Data
- New Methodology on Census Data for State Government Finances



WiNDC 2.1 - Some Details

- A full dataset is available for years 1997-2017
- All data back to 1997 was reloaded from all the original sources
 - Data as far back as 1999 is still being updated (but most data that is updated is after 2014)
 - Most changes were to BEA, Census data
- Added in NASS Ag Census 2017 Data
- New Methodology on Census Data for State Government Finances

Both versions are available in the new data interface



Update to Data Interface

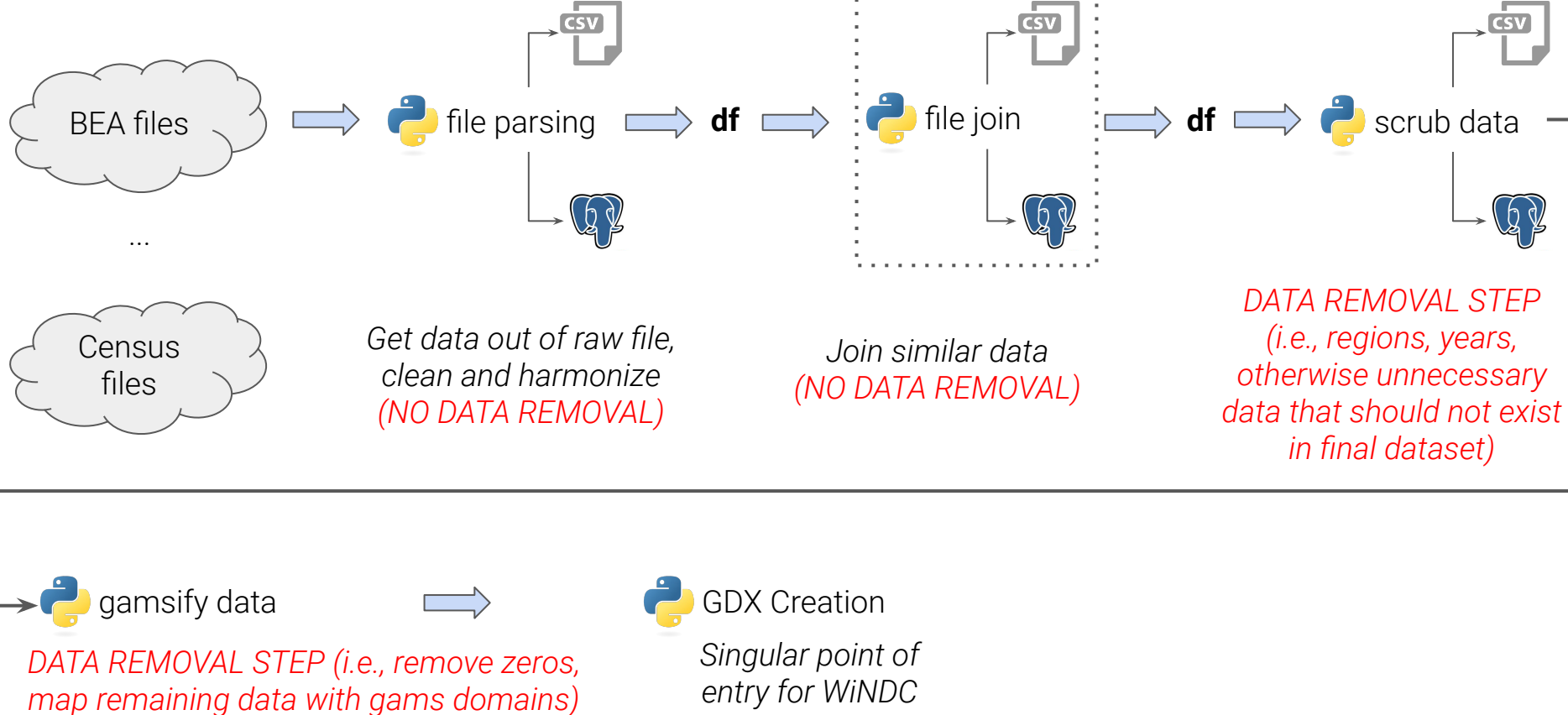
- Motivated by a need to:
 - Easily compare data over time
 - Relies on open source tools (python)
 - Provide a platform that can leverage powerful data analysis tools (Pandas)
 - Dropping direct support for SQL data, but Pandas has nice methods to export to SQL
 - New interface is called `windc_data`
 - `windc_data` is installable with `pip`



Update to Data Interface

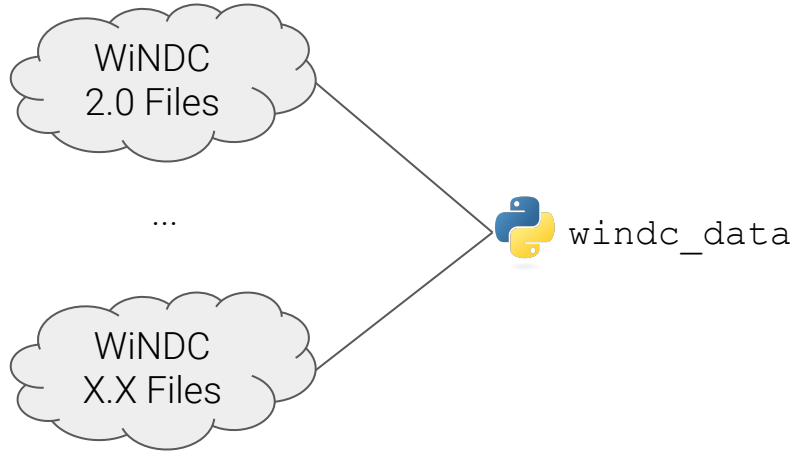
- Motivated by a need to:
 - Easily compare data over time
 - Relies on open source tools (python)
 - Provide a platform that can leverage powerful data analysis tools (Pandas)
 - Dropping direct support for SQL data, but Pandas has nice methods to export to SQL
 - New interface is called `windc_data`
 - `windc_data` is installable with `pip`

The Old Interface





Build Class Structure to Enable Data Processing



WINDC Data Loader Class (WindcEnvironment)

- Import different versions of windc with one package (`import windc_data`)
- Uniform label mapping
- Easier comparison operations once in memory



windc_data includes tools...

- Cleaning data is a problem
- `windc_data` attempts to help you with this...
 - We've heard from several groups that want to add in new data to WiNDC
 - Need a way to create a **global** WiNDC "notation" and apply that notation to a number of data sources
 - Some automation is helpful to get the data harmonized... still will always require some level of human interaction



windc_data includes tools...

- Cleaning data is a problem
- windc_data attempts to help you with this...
 - We've heard from several groups that want to add in new data to WiNDC
 - Need a way to create a **global** WiNDC “notation” and apply that notation to a number of data sources
 - Some automation is helpful to get the data harmonized... still will always require some level of human interaction



windc_data includes tools...

- Cleaning data is a problem
- windc_data attempts to help you with this...
 - We've heard from several groups that want to add in new data to WiNDC
 - Need a way to create a **global** WiNDC "notation" and apply that notation to a number of data sources
 - Some automation is helpful to get the data harmonized... still will always require some level of human interaction

		0	1	2	3
0	0	country	continent	GDP	population
1	0	USA	North America	19,390,604	322,179,605
2	1	China	Asia	12,237,700	1,403,500,365
3	2	Japan	Asia	4,872,137	127,748,513
4	3	Germany	Europe	3,677,439	81,914,672
5	4	UK	Europe	2,622,434	65,788,574
	5	India	Asia	2,597,491	1,324,171,354



windc_data includes tools...

- Cleaning data is a problem
- `windc_data` attempts to help you with this
 - We've heard from several groups that want to add in new data to WiNDC
 - Need a way to create a *global* WiNDC "notation" and apply that notation to a number of data sources
 - Some automation is helpful to get the data harmonized... still will always require some level of human interaction

	0	1	2	3		
0	country	continent	GDP	population		
1	0	USA	North America	19,390,604	322,179,605	
2	1	0	China	Asia	12,237,700	1,403,500,365
3	2	1	Japan	Asia	4,872,137	127,748,513
4	3	2	Germany	Europe	3,677,439	81,914,672
5	4	3	UK	Europe	2,622,434	65,788,574
5	5	4	India	Asia	2,597,491	1,324,171,354

**WiNDC Notation for
"windc.country":**
{ "United States",
"China",
"Japan",
"Germany",
"United Kingdom",
"India" }



windc_data includes tools...

- Cleaning data is a problem
- windc_data attempts to help you with this
 - We've heard from several groups that want to add in new data to WiNDC
 - Need a way to create a *global* WiNDC "notation" and apply that notation to a number of data sources
 - Some automation is helpful to get the data harmonized... still will always require some level of human interaction

	0	1	2	3		
0						
1	0					
2	1	0				
3	2	1	0			
4	3	2	1	0		
5	4	3	2	1	0	
	5	4	3	2	1	0
		country	continent	GDP	population	
		USA	North America	19,390,604	322,179,605	
		China	Asia	12,237,700	1,403,500,365	
		Japan	Asia	4,872,137	127,748,513	
		Germany	Europe	3,677,439	81,914,672	
		UK	Europe	2,622,434	65,788,574	
		India	Asia	2,597,491	1,324,171,354	



**WiNDC Notation for
"windc.country":**
{ "United States",
"China",
"Japan",
"Germany",
"United Kingdom",
"India" }



windc_data includes tools...

- Cleaning data is a problem
- `windc_data` attempts to help you with this
 - We've heard from several groups that want to add in new data to WiNDC
 - Need a way to create a *glob* of `windc_data` objects, with a way to connect a number of data sources
 - Some automation is helpful, but some level of human interaction

```
notation_link["gdp"] =  
[("country", "windc.country")]
```

	0	1	2	3
0				
1	0			
2	1	0		
3	2	1	0	
4	3	2	1	0
5	4	3	2	1
	5	4	3	2
		5	4	3
			5	4
				5

	country	continent	GDP	population
0	USA	North America	19,390,604	322,179,605
1	China	Asia	12,237,700	1,403,500,365
2	Japan	Asia	4,872,137	127,748,513
3	Germany	Europe	3,677,439	81,914,672
4	UK	Europe	2,622,434	65,788,574
5	India	Asia	2,597,491	1,324,171,354

Notation link

WiNDC Notation for "windc.country":
{ "United States",
"China",
"Japan",
"Germany",
"United Kingdom",
"India" }



.test_notation() output...

column name year linked to year

**** Drop detected... ({data} is a proper superset of {notation})

year :: eia_emissions

0 Notation elements not in Data

** 17 Drop Candidates **

□

['1980', '1981', '1982', '1983', '1984', '1985',
'1986', '1987', '1988', '1989', '1990', '1991',
'1992', '1993', '1994', '1995', '1996']

column name State linked to region.fullname

State :: eia_emissions

Valid dense data detected...

{data} == {notation}



.test_notation() output...

column name year linked to year

**** Drop detected... ({data} is a proper superset of {notation})

```
year :: eia_emissions
```

```
0 Notation elements not in Data
```

```
** 17 Drop Candidates **
```

```
[]
```

```
['1980', '1981', '1982', '1983', '1984', '1985',  
'1986', '1987', '1988', '1989', '1990', '1991',  
'1992', '1993', '1994', '1995', '1996']
```

column name State linked to region.fullname

```
State :: eia_emissions
```

```
Valid dense data detected...
```

```
{data} == {notation}
```

**User should arrive at valid data
before creating the GDX**



Good news!

- Users of WiNDC do not need to do all these cleaning operations
- All all users need to do is `.rebuild(gdxout=True)`



Good news!

- Users of WiNDC do not need to do all these cleaning operations
- All all users need to do is `.rebuild(gdxout=True)`

LIVE DEMO TIME

