Analyzing Public Data with Wolfram: UN Sustainable Development Goals

Maureen Baehr, Ben Kickert, Benji Bernstein

Join the Conversation #WolframTechConf

Overview

The focus and investment that has occurred in the last 20 years around multi-lateral international development has brought with it an astounding amount of publicly available data. In 2015, the United Nations officially outlined the Sustainable Development Goals (SDGs) as a successor to the well-received Millennium Development Goals. As most recently outlined, the SDGs comprise 17 major goals, 169 targets and 244 indicators. When factoring in data segregation, this framework outlines thousands of data points for every country in the UN to report on. This multitude of data presents unprecedented opportunities as well as new challenges.

The structure and functionality of the Wolfram Language and its related tools provides its users with powerful resources for analyzing and utilizing this data to be able to gather insights and consolidate information.

In this presentation, we will explore:
1.) Accessing and structuring publicly available datasets for analysis within the Wolfram technology stack.
2.) Utilizing integrated Wolfram features for data analysis by creating relational connections between otherwise "flat" datasets.
3.) Connecting public data with built-in Wolfram resources to attain an accurate and resilient understanding of

trends in international development.
4.) Automating the connections.
5.) The challenges facing data analysis such as size, formatting data gaps and conflicting data.

At this end of this session, participants will have a firm grasp on how tools from Wolfram can be used for structuring and analyzing publicly available data as well as an understanding of the possibilities and limitations around aggregating data points from a variety of international sources.

Background

- The Sustainable Development Goals -- 15 Year Global Strategy
- The Wolfram Tech Stack -- Growing Together
- The First 3 Years -- Waiting, Planning, Learning
- The Next 12 Years...
- Today

Accessing and structuring publicly available datasets

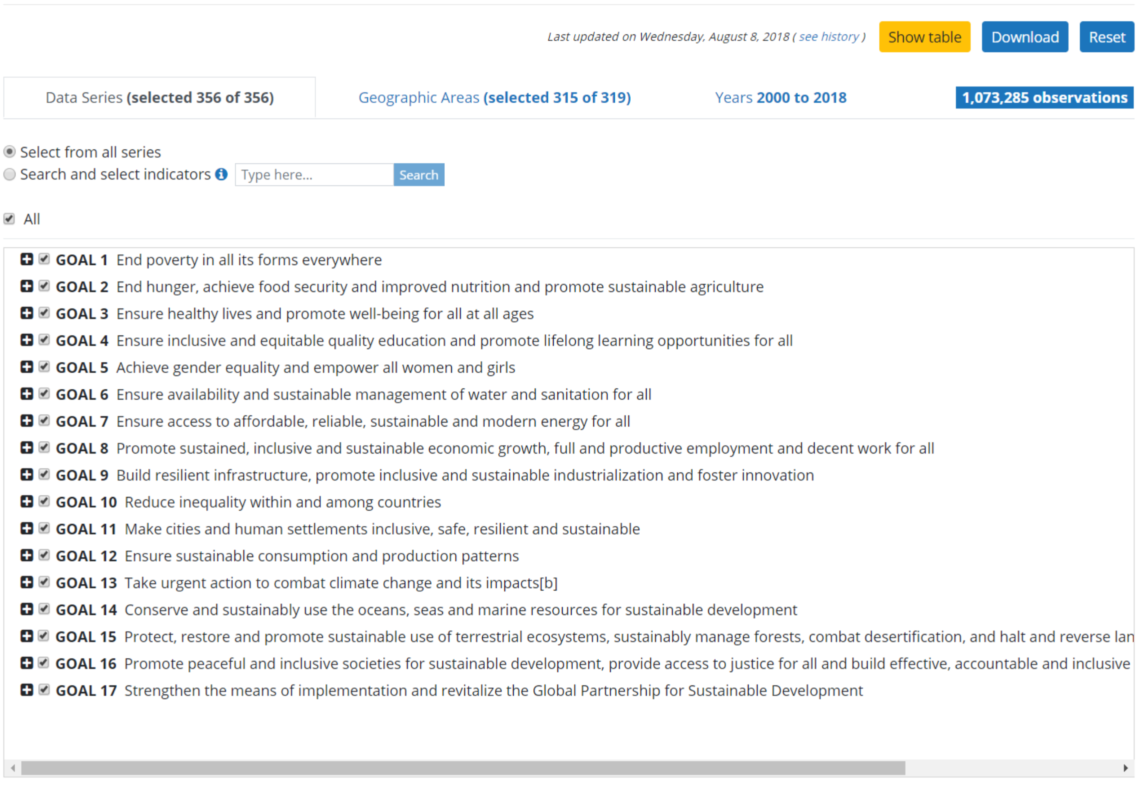
Getting a Grasp on the Data Points

- SDGs have 244 Indicators across 169 Targets and 17 Goals reported across 255 geographic areas with numerous groupings and sub groupings. A majority of indicators are then disaggregated by gender, age, population or measurement type.
- While SDG Goals, Targets and Indicators have been firmed up in recent years, data collection methods are still dynamic. As of October 2018, 93 Indicators are clearly defined and methodology is uniform, 77 are clearly defined but are not being uniformly collected by countries and ***57 Indicators still lack agreed upon collection methodology.***
- Public Data sources fall into three broad categories: ***Official UN Results, Derivative Analysis, Original Primary Data Sources***

SDG Data Sources

- Official Results: UN SDG Database

  - Comprehensive and customizable database of all official indicator results across multiple decades.
  - Robust but cumbersome (Full data includes over 1,000,000 observations and a file size of 300+MB)



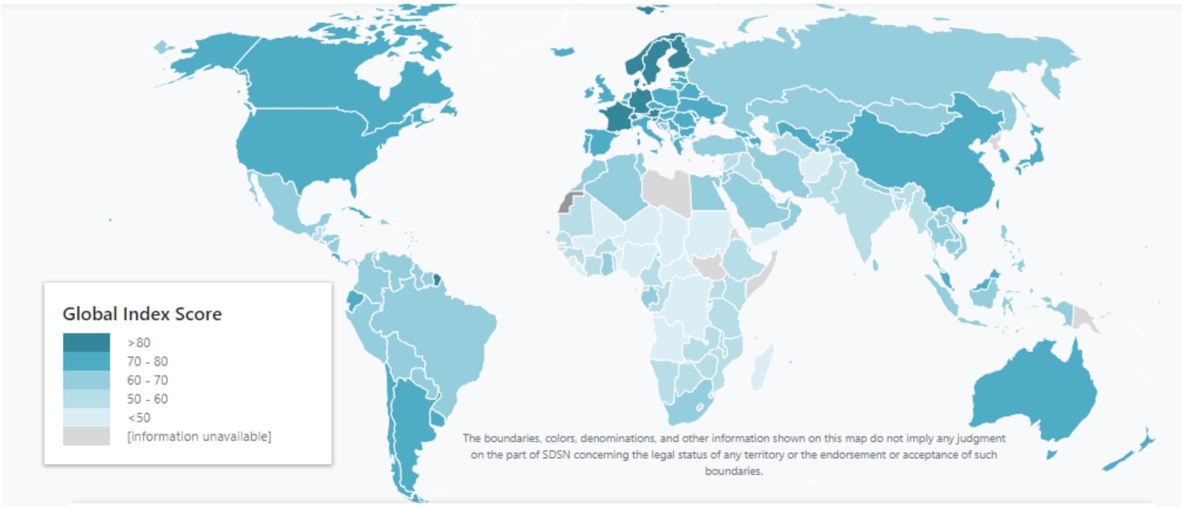
■ Derivative Analysis: SDG Index

- Provides simplified Indicator results for all countries along with analysis
- Designed to provide visual representations (through dashboards) for easier understanding along with simplified downloadable data.

SDG Index and Dashboards Report 2018

Global Responsibilities: Implementing the goals

SUSTAINABLE DEVELOPMENT SOLUTIONS NETWORK | BertelsmannStiftung



Sustainable Development Goals

Select one of the 17 SDGs to see it on the map

The SDG Index

Select a country to see its full profile

Search country...

Country	Region	SDGs	Rank ▲	Score
Sweden	OECD members		1	85.0
Denmark	OECD members		2	84.6
Finland	OECD members		3	83.0
Germany	OECD members		4	82.3
France	OECD members		5	81.2

- Original Primary Data Source: WHO TB Data
  - Considered the definitive source for Tuberculosis information and “supplier” of data for UN SDG

SDGs only provide one indicator on the disease, but using WHO data we get access to significant more

Accessing SDG Data

Global emphasis on the role of data in international development, has resulted in significant sources of raw data are being made available that can easily be downloaded or imported.

Official SDG Sample Data

Customizable datasets can be downloaded here: https://unstats.un.org/sdgs/indicators/database/

In[ ]:=
```
sdgdatabaserawdata = Import["https://www.wolframcloud.com/objects/ben.kickert/sdgsampledatabase.csv"][[1 ;; -2]];
sdgdatabaseassoc = AssociationThread[sdgdatabaserawdata[[1]] → #] & /@Rest[sdgdatabaserawdata];
Dataset[sdgdatabaseassoc]
```

Out[ ]=

Goal	Target	Indicator	SeriesCode	SeriesDescription
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000
3	3.3	3.3.1	SH_HIV_INCD	Number of new HIV infections per 1,000

showing 1–20 of 664

SDG Index Data

In[ ]:=
```
sdgindexrawdata = Import["http://www.sdgindex.org/assets/files/2018/Time_Series_Database.xlsx", {"Data", 1}];
sdgindexassoc = AssociationThread[sdgindexrawdata[[1]] → #] &/@Rest[sdgindexrawdata];
sdgindexdataset = Dataset[sdgindexassoc]
```

Out[ ]=

coun	id	year	UN re	UN s	Regi	Incoı	Pove	Cere	Preva
Afghanistan	AFG	2000.	Asia	Southern Asia	E. Europe & C. Asia	LIC		0.8063	2.3
Afghanistan	AFG	2001.	Asia	Southern Asia	E. Europe & C. Asia	LIC		1.0067	2.4
Afghanistan	AFG	2002.	Asia	Southern Asia	E. Europe & C. Asia	LIC		1.6698	2.6
Afghanistan	AFG	2003.	Asia	Southern Asia	E. Europe & C. Asia	LIC		1.458	2.7
Afghanistan	AFG	2004.	Asia	Southern Asia	E. Europe & C. Asia	LIC		1.3348	2.9
Afghanistan	AFG	2005.	Asia	Southern Asia	E. Europe & C. Asia	LIC		1.7904	3.
Afghanistan	AFG	2006.	Asia	Southern Asia	E. Europe & C. Asia	LIC		1.5517	3.2
Afghanistan	AFG	2007.	Asia	Southern Asia	E. Europe & C. Asia	LIC		1.9153	3.4
Afghanistan	AFG	2008.	Asia	Southern Asia	E. Europe & C. Asia	LIC		1.4554	3.6
Afghanistan	AFG	2009.	Asia	Southern Asia	E. Europe & C. Asia	LIC		2.0407	3.8
Afghanistan	AFG	2010.	Asia	Southern Asia	E. Europe & C. Asia	LIC	41.2375	2.0111	4.
Afghanistan	AFG	2011.	Asia	Southern Asia	E. Europe & C. Asia	LIC	40.1766	1.6599	4.2
Afghanistan	AFG	2012.	Asia	Southern Asia	E. Europe & C. Asia	LIC	37.0343	2.0296	4.4
Afghanistan	AFG	2013.	Asia	Southern Asia	E. Europe & C. Asia	LIC	36.302	2.0485	4.7
Afghanistan	AFG	2014.	Asia	Southern Asia	E. Europe & C. Asia	LIC	36.4169	2.0175	4.9
Afghanistan	AFG	2015.	Asia	Southern Asia	E. Europe & C. Asia	LIC	36.8821	2.1332	5.2
Afghanistan	AFG	2016.	Asia	Southern Asia	E. Europe & C. Asia	LIC	36.2571	1.9817	5.5
Afghanistan	AFG	2017.	Asia	Southern Asia	E. Europe & C. Asia	LIC	35.4439		
Afghanistan	AFG	2018.	Asia	Southern Asia	E. Europe & C. Asia	LIC	34.5652		
Albania	ALB	2000.	Europe	Southern Europe	E. Europe & C. Asia	UMIC		3.1751	13.2

showing 1–20 of 3667

WHO TB Data

https : // extranet.who.int/tme/generateCSV.asp?ds = estimates

In[ ]:=
```
tbrawdata = Import["https://www.wolframcloud.com/objects/ben.kickert/tbburden.csv"];
tbdataassoc = AssociationThread[tbrawdata[[1]] → #] & /@Rest[tbrawdata];
tbindexdataset = Dataset[tbdataassoc]
```

Out[ ]=

coun	iso2	iso3	iso_n	g_wh	year	e_po	e_inc	e_inc	e_inc	e_inc	e_inc	e_inc	e_inc	e_i
Afghanistan	AF	AFG	4	EMR	2000	20 093 7	190	123	271	38 000	25 000	54 000		
Afghanistan	AF	AFG	4	EMR	2001	20 966 4	189	123	271	40 000	26 000	57 000		
Afghanistan	AF	AFG	4	EMR	2002	21 979 9	189	122	270	42 000	27 000	59 000		
Afghanistan	AF	AFG	4	EMR	2003	23 064 8	189	122	270	44 000	28 000	62 000		
Afghanistan	AF	AFG	4	EMR	2004	24 118 9	189	122	270	46 000	29 000	65 000		
Afghanistan	AF	AFG	4	EMR	2005	25 070 7	189	122	270	47 000	31 000	68 000		
Afghanistan	AF	AFG	4	EMR	2006	25 893 4	189	122	270	49 000	32 000	70 000		
Afghanistan	AF	AFG	4	EMR	2007	26 616 7	189	122	270	50 000	33 000	72 000		
Afghanistan	AF	AFG	4	EMR	2008	27 294 0	189	122	270	52 000	33 000	74 000		
Afghanistan	AF	AFG	4	EMR	2009	28 004 3	189	123	270	53 000	34 000	76 000		
Afghanistan	AF	AFG	4	EMR	2010	28 803 1	189	123	270	55 000	35 000	78 000		
Afghanistan	AF	AFG	4	EMR	2011	29 708 5	189	123	270	56 000	36 000	80 000		
Afghanistan	AF	AFG	4	EMR	2012	30 696 9	189	122	270	58 000	38 000	83 000		
Afghanistan	AF	AFG	4	EMR	2013	31 731 6	189	122	270	60 000	39 000	86 000		
Afghanistan	AF	AFG	4	EMR	2014	32 758 0	189	122	270	62 000	40 000	88 000		
Afghanistan	AF	AFG	4	EMR	2015	33 736 4	189	122	270	64 000	41 000	91 000		
Afghanistan	AF	AFG	4	EMR	2016	34 656 0	189	122	270	65 000	42 000	93 000	4100	25
Albania	AL	ALB	8	EUR	2000	3 121 97	22	19	26	690	590	800		
Albania	AL	ALB	8	EUR	2001	3 122 40	20	17	24	640	550	740		
Albania	AL	ALB	8	EUR	2002	3 119 02	22	19	25	680	580	790		

showing 1–20 of 3634

Creating Relational Connections through EntityStores

- Analyzing large and complex data often requires understanding relational data.

- Consider the Goals, Indicators and Targets. While each have their own descriptions, it is also important to understand how they are nested.

- Relational EntityStores can provide these connections through Entities as Property Values

Relational EntityStore

Beginning with a simple document listing the Goals, Indicators and Targets, we created an EntityStore in the Wolfram Cloud (public, but not published in the Repository)

```
In[•]:= EntityRegister[ResourceData["https://www.wolframcloud.com/objects/d837db6f-955c-\
4893-a96c-d9674a0b61a4"]]

Out[•]= {Goals, Targets, Indicators}
```

We can look at the Properties as well as individual data points:

```
In[•]:= indicatorproperties = EntityProperties["Indicators"]

Out[•]= {UNSD Indicator Codes, Description, Parent Goal, Parent Target}

In[•]:= Entity["Indicators", "2.1.1"][#] & /@ indicatorproperties

Out[•]= {C020101, Prevalence of undernourishment, 2, 2.1}
```

By using Entities as Property Values, we can easily look at nested descriptions

```
In[•]:= Entity["Indicators", "2.1.1"]["Description"]
        Entity["Indicators", "2.1.1"]["ParentTarget"]["Description"]
        Entity["Indicators", "2.1.1"]["ParentGoal"]["Description"]

Out[•]= Prevalence of undernourishment

Out[•]= By 2030, end hunger and ensure access by all people, in particular the poor and people
          in vulnerable situations, including infants, to safe, nutritious and sufficient food all year round

Out[•]= End hunger, achieve food security and improved nutrition and promote sustainable agriculture
```

We can also find all the targets under goal 10 and then list their descriptions:

```
In[•]:= goal10targets = EntityList[EntityClass["Targets", EntityProperty["Targets", "ParentGoal"] → Entity["Goals", "10"]]]
        EntityValue[# & /@ goal10targets, "Description"]

Out[•]= {10.1, 10.2, 10.3, 10.4, 10.5, 10.6, 10.7, 10.a, 10.b, 10.}

Out[•]= {By 2030, progressively achieve and sustain income growth
           of the bottom 40 per cent of the population at a rate higher than the national average,
         By 2030, empower and promote the social, economic and political inclusion of all, irrespective of
           age, sex, disability, race, ethnicity, origin, religion or economic or other status,
         Ensure equal opportunity and reduce inequalities of outcome, including by eliminating discriminatory laws,
           policies and practices and promoting appropriate legislation, policies and action in this regard,
         Adopt policies, especially fiscal, wage and social protection policies, and progressively achieve greater equality,
         Improve the regulation and monitoring of global financial
           markets and institutions and strengthen the implementation of such regulations,
         Ensure enhanced representation and voice for developing countries in decision-making in global international economic
           and financial institutions in order to deliver more effective, credible, accountable and legitimate institutions,
         Facilitate orderly, safe, regular and responsible migration and mobility of people, including
           through the implementation of planned and well-managed migration policies,
         Implement the principle of special and differential treatment for developing countries, in particular
           least developed countries, in accordance with World Trade Organization agreements,
         Encourage official development assistance and financial flows, including foreign direct investment, to States
           where the need is greatest, in particular least developed countries, African countries, small island
           developing States and landlocked developing countries, in accordance with their national plans and programmes,
         By 2030, reduce to less than 3 per cent the transaction costs of migrant remittances and
           eliminate remittance corridors with costs higher than 5 per cent}
```

Incorporating SDG Descriptions into Public Data

Taking our SDG Database, we can now move take flat data from the UN and make it interactive.

In[∘]:=
```
sdgrelationaldata = sdgdatabaseassoc;
goallist = ToString /@DeleteDuplicates[sdgrelationaldata[[All, "Goal"]]];
goalreplace = Thread[goallist → (Entity["Goals", #] & /@goallist)];
goalreplacelist = ToString /@sdgrelationaldata[[All, "Goal"]] /. goalreplace;
sdgrelationaldata[[All, "Goal"]] = goalreplacelist;
sdgrelationaldata[[1]];

indicatorlist = ToString /@DeleteDuplicates[sdgrelationaldata[[All, "Indicator"]]];
indicatorreplace = Thread[indicatorlist → (Entity["Indicators", #] & /@indicatorlist)];
indicatorreplacelist = ToString /@sdgrelationaldata[[All, "Indicator"]] /. indicatorreplace;
sdgrelationaldata[[All, "Indicator"]] = indicatorreplacelist;

targetlist = ToString /@DeleteDuplicates[sdgrelationaldata[[All, "Target"]]];
targetreplace = Thread[targetlist → (Entity["Targets", #] & /@targetlist)];
targetreplacelist = ToString /@sdgrelationaldata[[All, "Target"]] /. targetreplace;
sdgrelationaldata[[All, "Target"]] = targetreplacelist;

sdgrelationaldata[[1]]
```

Out[∘]=
```
⟨|Goal → 3, Target → 3.3, Indicator → 3.3.1, SeriesCode → SH_HIV_INCD,
 SeriesDescription → Number of new HIV infections per 1,000 uninfected population, by sex and age (per 1,000 uninfected population),
 GeoAreaCode → 710, GeoAreaName → South Africa, TimePeriod → 2000, Value → 13.46, Time_Detail → 2000, Source → UNAIDS,
 FootNote → , Nature → E, [Age] → ALLAGE, [Bounds] → , [Sex] → BOTHSEX, [Units] → PER_1000_UNINFECTED_POP|⟩
```

Importing Wolfram Entities

- The same technique can be used to integrate curated Wolfram Entities
- The depth of information from sources such as CountryData is now available to users

In[∘]:=
```
countries = DeleteDuplicates[sdgrelationaldata[[All, "GeoAreaName"]]];
countryentities = Interpreter["Country"][countries];
countryreplace = Thread[countries → countryentities];
countryreplacelist = sdgrelationaldata[[All, "GeoAreaName"]] /. countryreplace;
sdgrelationaldata[[All, "GeoAreaName"]] = countryreplacelist;
```

In[∘]:=
```
Dataset[sdgrelationaldata[[1]]]
```

Out[∘]=

Goal	3
Target	3.3
Indicator	3.3.1
SeriesCode	SH_HIV_INCD
SeriesDescription	Number of new HIV infections per 1,000 uninfected population, by sex and age (pe
GeoAreaCode	710
GeoAreaName	South Africa
TimePeriod	2000
Value	13.46
Time_Detail	2000
Source	UNAIDS
FootNote	
Nature	E
[Age]	ALLAGE
[Bounds]	
[Sex]	BOTHSEX
[Units]	PER_1000_UNINFECTED_POP

Significance of linking relational data

- In the example above we can easily explore the definitions of the SDG's without having to exit the Wolfram Environment.
- External values from other sources can be accessed and incorporated
- Multi-dimensional analysis can then be done on an otherwise flat table

Using Wolfram to Analyze Public Data - A Real Life Example

Using our the sample SDG data with embedded entities created above, we can now drill deeper into the information

Utilizing CountryData Analysis

Let’s take a broad look at the information we have and how we can use our embedded Entities

Start with raw counts of country data:

In[∘]:=
```
sdgrelationaldataset = Dataset[sdgrelationaldata];
```

In[∘]:=
```
counts = Counts[sdgrelationaldataset[[All, "GeoAreaName"]]]
```

Out[∘]=

South Africa	206
Swaziland	198
United States	99
Canada	79
Switzerland	82

Now let’s look at something as simple as Life Expectancy

In[∘]:=
```
sdgcountries = DeleteDuplicates[sdgrelationaldata[[All, "GeoAreaName"]]]
```

Out[∘]= {South Africa, Swaziland, United States, Canada, Switzerland}

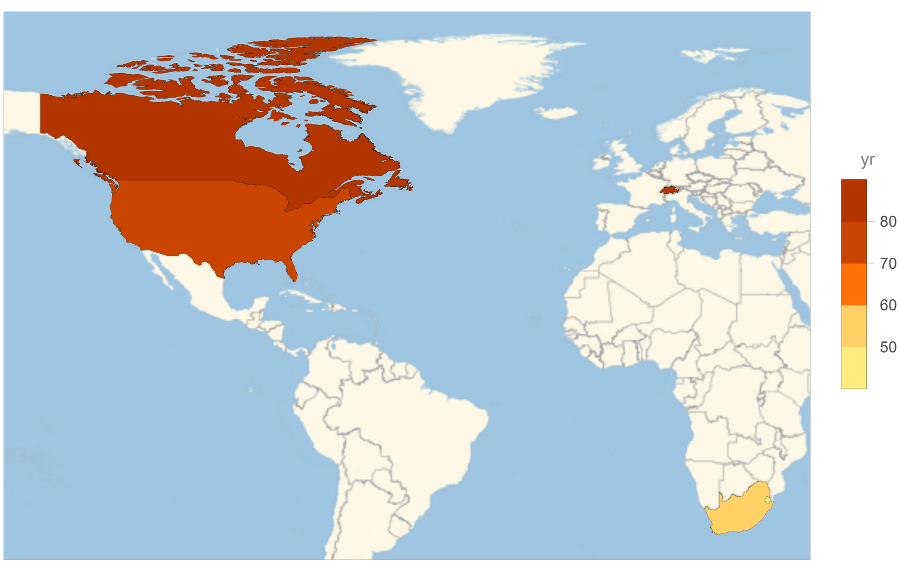
In[∘]:=
```
lifeexpectancy = Thread[# → CountryData[#, "LifeExpectancy"]] & /@ sdgcountries
```

Out[∘]= {South Africa → 56.916 yr, Swaziland → 49. yr, United States → 78.941 yr, Canada → 81.482 yr, Switzerland → 82.604 yr}

If we want to visualize that data, we can use a simple GeoChart:

In[∘]:=
```
GeoRegionValuePlot[lifeexpectancy]
```

Out[∘]=



Utilizing SDG EntityStore Descriptions

Using basic dataset commands, let’s look at how much information we have for each indicator:

In[∘]:=
```
sdgrelationaldataset[Counts, "Indicator"]
```

Out[∘]=

3.3.1	286
3.3.2	255
3.7.1	9
3.7.2	59
16.1.1	55

Seeing as we have the most information on 3.3.1, let’s remind ourselves of what that data point is representing:

In[∘]:=
```
Entity["Indicators", "3.3.1"]["Description"]
Entity["Indicators", "3.3.1"]["ParentTarget"]["Description"]
Entity["Indicators", "3.3.1"]["ParentGoal"]["Description"]
```

Out[∘]= Number of new HIV infections per 1,000 uninfected population, by sex, age and key populations

Out[∘]= By 2030, end the epidemics of AIDS, tuberculosis, malaria and neglected tropical diseases and combat hepatitis, water-borne diseases and other communicable diseases

Out[∘]= Ensure healthy lives and promote well-being for all at all ages

Manipulating the UN Dataset in Wolfram

Taking the Indicator from above, let’s look at country distribution:

In[∘]:=
```
maxhiv = sdgrelationaldataset[Select[#Indicator == Entity["Indicators", "3.3.1"] &], "GeoAreaName"][Counts]
```

Out[∘]=

South Africa	136
Swaziland	136
United States	14

HIV data is often disaggregated by age and gender, so let's pull only "ALLAGE" and "BOTHSEX" data points

In[ ]:=
```
totalhiv = sdgrelationaldataset[Select[#Indicator == 3.3.1 SDG INDICATOR &&#"[Age]" == "ALLAGE"&&#"[Sex]" == "BOTHSEX" &]]
```

Out[ ]=

Goal	Target	Indicator	SeriesCode	SeriesDescri
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n
3	3.3	3.3.1	SH_HIV_INCD	Number of n

showing 1–20 of 34

We can then Visualize this by year

In[ ]:=
```
totalhivbyyearandcountry = totalhiv[GroupBy["GeoAreaName"], Sort[GroupBy["Time_Detail"]], Last /* "Value"]
```

Out[ ]=

South Africa	2000	13.46
	2001	13.07
	2002	12.7
	2003	12.44
	2004	12.16
	2005	11.78
	2006	11.31
	2007	10.76
	2008	10.28
	2009	9.29
	17 total ›	
Swaziland	2000	21.47
	2001	19.61
	2002	18.03
	2003	17.12
	2004	16.37
	2005	16.26
	2006	15.42
	2007	15.5
	2008	15.35
	2009	15.05
	17 total ›	

Or pull only the most recent data:

In[ ]:=
```
mostrecenttotalhiv = totalhiv[GroupBy[#GeoAreaName &] /* Values, MaximalBy[#TimePeriod &] /* First][All, {"GeoAreaName", "Value"}]
maxhivrates = Query[GroupBy[#[["GeoAreaName"]] &], Max, #[["Value"]] &][totalhiv]
```

Out[ ]=

GeoAreaName	Value
South Africa	5.58
Swaziland	9.37

Out[ ]=

South Africa	13.46
Swaziland	21.47

We can also look at the average (mean) results across all indicators and countries:

In[•]:=
```
averageacrossindicators =
 Query[GroupBy[#[["GeoAreaName"]] &], GroupBy[#[["Indicator"]] &], Mean, #[["Value"]] &][sdgrelationaldataset]
```

Out[•]=

South Africa	3.3.1	11.8326
	3.3.2	872.549
	3.7.1	79.5
	3.7.2	63.3294
	5 total ›	
Swaziland	3.3.1	19.6121
	3.3.2	977.235
	3.7.1	74.8667
	3.7.2	89.6
	5 total ›	
United States	3.3.1	0.217143
	3.3.2	4.75294
	3.7.1	83.275
	3.7.2	35.6176
	5 total ›	
Canada	3.3.2	5.49804
	3.7.2	13.9714
	16.1.1	283.301
Switzerland	3.3.2	7.66863
	3.7.2	4.45141
	16.1.1	24.8021

Or quickly analyze how many observations each country has for each indicator:

In[ ]:=
```
Query[GroupBy[#[["Indicator"]] &], GroupBy[#[["GeoAreaName"]] &], Length, #[["Value"]] &][sdgrelationaldataset]
```

Out[ ]=

3.3.1	South Africa	136
	Swaziland	136
	United States	14
3.3.2	Canada	51
	Swaziland	51
	South Africa	51
	Switzerland	51
	5 total ›	
3.7.1	South Africa	2
	Swaziland	3
	United States	4
3.7.2	Canada	14
	Switzerland	17
	South Africa	5
	United States	17
	5 total ›	
16.1.1	Canada	14
	South Africa	12
	Swaziland	2
	Switzerland	14
	5 total ›	

Utilizing Wolfram for Visualization

In the chart above, we saw that Indicator 3.3.2 had the most consistent data reporting. Let’s look at the descriptions:

In[ ]:=
```
Entity["Indicators", "3.3.2"]["Description"]
Entity["Indicators", "3.3.2"]["ParentTarget"]["Description"]
Entity["Indicators", "3.3.2"]["ParentGoal"]["Description"]
```

Out[ ]= Tuberculosis incidence per 100,000 population

Out[ ]= By 2030, end the epidemics of AIDS, tuberculosis, malaria and neglected tropical diseases and combat hepatitis, water-borne diseases and other communicable diseases

Out[ ]= Ensure healthy lives and promote well-being for all at all ages

TB data is reported with high, low, and mid bounds. Let’s pull only the mid bound for analysis

In[ ]:=
```
tbdata = sdgrelationaldataset[Select[#Indicator == Entity["Indicators", "3.3.2"] && #"[Bounds]" == "MP" &]][[
  All, {"GeoAreaName", "TimePeriod", "Value"}]]
```

Out[ ]=

GeoAreaName	TimePeriod	Value
Canada	2000	6.4
Canada	2001	6.6
Canada	2002	6.1
Canada	2003	5.9
Canada	2004	5.6
Canada	2005	5.5
Canada	2006	5.4
Canada	2007	5.3
Canada	2008	5.4
Canada	2009	5.3
Canada	2010	4.6
Canada	2011	4.8
Canada	2012	5.4
Canada	2013	5.3
Canada	2014	5.1
Swaziland	2000	831
Swaziland	2001	855
Swaziland	2002	936
Canada	2015	5.2
Canada	2016	5.2

showing 1–20 of 85

Again we can pull the most recent data, then create an AssociationThread that can be used in a map for visualization

In[∘]:=
```
mostrecenttbdata = tbdata[GroupBy[#GeoAreaName &] /* Values, MaximalBy[#TimePeriod &] /* First][All, {"GeoAreaName", "Value"}]
tbmapdata = Rule@@@mostrecenttbdata
```

Out[∘]=

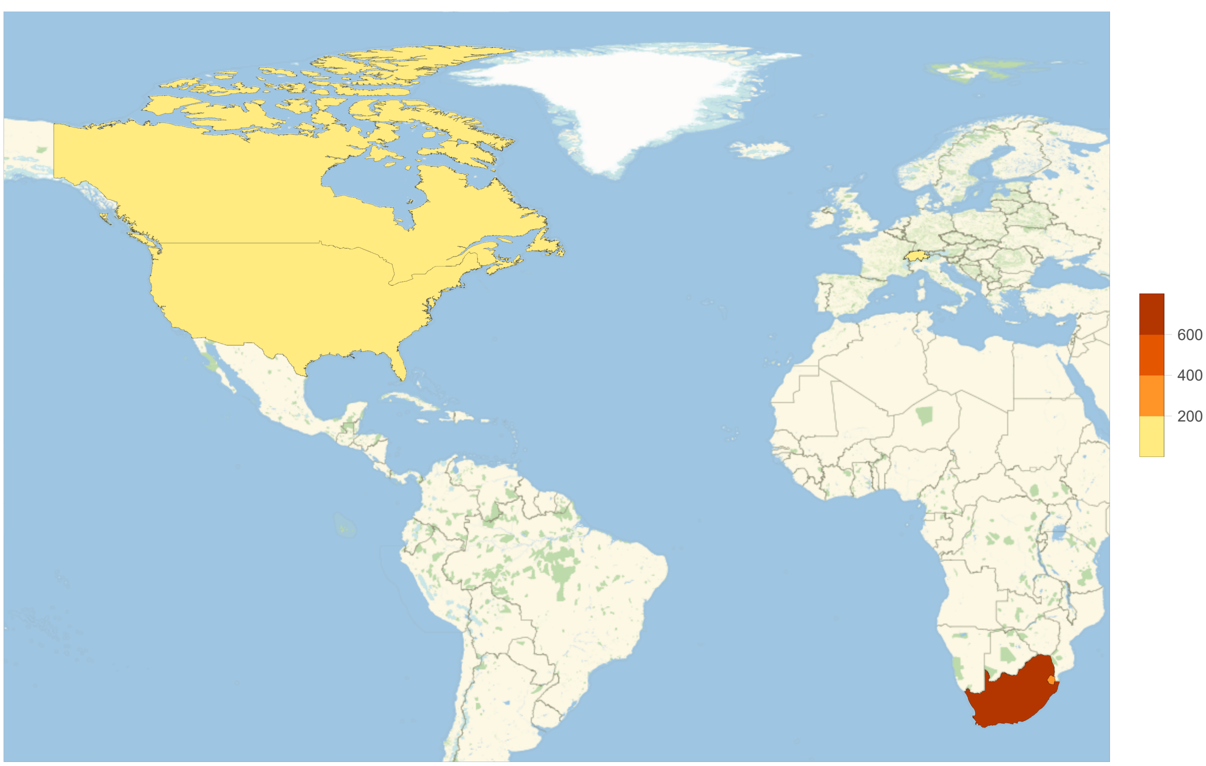
GeoAreaName	Value
Canada	5.2
Swaziland	398
South Africa	781
Switzerland	7.8
United States	3.1

Out[∘]=

Canada	→ 5.2
Swaziland	→ 398
South Africa	→ 781
Switzerland	→ 7.8
United States	→ 3.1

In[∘]:=
```
GeoRegionValuePlot[tbmapdata, ImageSize → Large]
```

Out[∘]=



Multi-Dimentional Analysis

Having looked at CountryData and UN Data separately, we can now bring those two values together.

In[∘]:=
```
tbandlife = Dataset[Join[mostrecenttbdata,
    AssociationThread["Life" → #] & /@ Normal[CountryData[#, "LifeExpectancy"] & /@ mostrecenttbdata[[All, "GeoAreaName"]]], 2]];
tbandlife = tbandlife[All, <|"Country" → "GeoAreaName", "TB Rate" → "Value", "Life Expectancy" → "Life"|>]
```

Out[∘]=

Country	TB Rate	Life Expectancy
Canada	5.2	81.482 yr
Swaziland	398	49. yr
South Africa	781	56.916 yr
Switzerland	7.8	82.604 yr
United States	3.1	78.941 yr

Using the natural language features of Wolfram, we can give this some context:

In[∘]:=
```
globallife = World COUNTRY [ life expectancy ]
```

Out[∘]= 72.0353 yr

Now we can easily see how TB rate relates to Life Expectancy of each country compared to the global rate.

In[∘]:=
```
tbandlife2 = Dataset[Join[tbandlife, AssociationThread["Life Percentage" → #] & /@
    UnitConvert[Round[N[# / globallife], .001] & /@ tbandlife[[All, "Life Expectancy"]], "Percent"], 2]]
```

Out[∘]=

Country	TB Rate	Life Expectancy	Life Percentage
Canada	5.2	81.482 yr	113.1%
Swaziland	398	49. yr	68.%
South Africa	781	56.916 yr	79.%
Switzerland	7.8	82.604 yr	114.7%
United States	3.1	78.941 yr	109.6%

It should come as no surprise that the countries with highest TB rates have the lower life expectancies.

The above examples are just a sample of how the Wolfram Technology Stack can be used for multi-dimensional analysis on publicly available data.

Automating the Process

For these examples, we have focused exclusively on Wolfram's country data. However, the same principles can apply to any of the Entity Types that Wolfram maintains.

In order to facilitate and simplify this process, the following defined functions have been developed.

convertHeader

Simple defined function for turning first row into headers.

```
In[•]:= convertHeader::usage =
  "convertHeader[list] Takes the first row of a list of lists and converts it into a header through AssocationThread.";
convertHeader[list_] := AssociationThread[First[list] → #] & /@ Rest[list]
```

listToEntities

Converts a list into Wolfram Entities so you can pull in more data if needed.It includes an error checker that returns the original input if a corresponding entity cannot be found.You must identify the Interpreter to use such as "County".

```
In[•]:= listToEntities::usage = "countryToEntityList[list,interpreter] Converts a list into Wolfram
    Entities by deduplicating and then replacing all instances according to the interpreter defined.";
listToEntities[list_, interpreter_] := list /. Thread[(DeleteDuplicates[list]) →
     (If[FailureQ[Interpreter[interpreter][#]], #, Interpreter[interpreter][#]] & /@ DeleteDuplicates[list])];
```

datasetEntityReplace

Takes values in a column of a dataset and runs them through the Entity Conversion process and replaces the original values.Useful for adding Relational data on countries, companies, or any of the other Entity classes that Wolfram Manages.

```
In[•]:= datasetEntityReplace::usage =
  "datasetEntityReplace[data,col,interpreter] Converts values in a dataset to their respective Wolfram Entities
    using three variables: datset, column data is in (name or number), and interpreter to use.";
datasetEntityReplace[data_, col_, interpreter_] := data[All,
  {col → Replace[(Thread[(DeleteDuplicates[Normal[data[[All, col]]]]) → (If[FailureQ[Interpreter[interpreter][#]],
          #, Interpreter[interpreter][#]] & /@ DeleteDuplicates[Normal[data[[All, col]]]])])]}]
```

addEntityToDataset

Similar to the datasetEntityReplace, except this adds a new column with the Entity so all the original data is preserved.

```
In[•]:= addEntityToDataset::usage =
  "addEntityToDataset[data,col,interpreter] Converts values in a dataset to their respective Wolfram Entities
    using three variables: dataset, column data is in (name or number), and interpreter to use.";
addEntityToDataset[data_, col_, interpreter_] := Join[AssociationThread["Entity" → #] & /@
    (data[[All, col]] /. Thread[(DeleteDuplicates[Normal[data[[All, col]]]]) → (If[FailureQ[Interpreter[interpreter][#]],
          #, Interpreter[interpreter][#]] & /@ DeleteDuplicates[Normal[data[[All, col]]]])]), data, 2];
```

Examples of automated functions

Adding a Relational Column to the SDG Index Dataset

In[ ]:=
```
relationalsdgindex = addEntityToDataset[sdgindexdataset, "id", "Country"]
```

Out[ ]=

Entit	coun	id	year	UN r	UN s	Regi	Incor	Pove
Afghanistan	Afghanistan	AFG	2000.	Asia	Southern Asia	E. Europe & C. Asia	LIC	
Afghanistan	Afghanistan	AFG	2001.	Asia	Southern Asia	E. Europe & C. Asia	LIC	
Afghanistan	Afghanistan	AFG	2002.	Asia	Southern Asia	E. Europe & C. Asia	LIC	
Afghanistan	Afghanistan	AFG	2003.	Asia	Southern Asia	E. Europe & C. Asia	LIC	
Afghanistan	Afghanistan	AFG	2004.	Asia	Southern Asia	E. Europe & C. Asia	LIC	
Afghanistan	Afghanistan	AFG	2005.	Asia	Southern Asia	E. Europe & C. Asia	LIC	
Afghanistan	Afghanistan	AFG	2006.	Asia	Southern Asia	E. Europe & C. Asia	LIC	
Afghanistan	Afghanistan	AFG	2007.	Asia	Southern Asia	E. Europe & C. Asia	LIC	
Afghanistan	Afghanistan	AFG	2008.	Asia	Southern Asia	E. Europe & C. Asia	LIC	
Afghanistan	Afghanistan	AFG	2009.	Asia	Southern Asia	E. Europe & C. Asia	LIC	
Afghanistan	Afghanistan	AFG	2010.	Asia	Southern Asia	E. Europe & C. Asia	LIC	41.2375
Afghanistan	Afghanistan	AFG	2011.	Asia	Southern Asia	E. Europe & C. Asia	LIC	40.1766
Afghanistan	Afghanistan	AFG	2012.	Asia	Southern Asia	E. Europe & C. Asia	LIC	37.0343
Afghanistan	Afghanistan	AFG	2013.	Asia	Southern Asia	E. Europe & C. Asia	LIC	36.302
Afghanistan	Afghanistan	AFG	2014.	Asia	Southern Asia	E. Europe & C. Asia	LIC	36.4169
Afghanistan	Afghanistan	AFG	2015.	Asia	Southern Asia	E. Europe & C. Asia	LIC	36.8821
Afghanistan	Afghanistan	AFG	2016.	Asia	Southern Asia	E. Europe & C. Asia	LIC	36.2571
Afghanistan	Afghanistan	AFG	2017.	Asia	Southern Asia	E. Europe & C. Asia	LIC	35.4439
Afghanistan	Afghanistan	AFG	2018.	Asia	Southern Asia	E. Europe & C. Asia	LIC	34.5652
Albania	Albania	ALB	2000.	Europe	Southern Europe	E. Europe & C. Asia	UMIC	

showing 1–20 of 3667

Just as we saw with the examples for TB, we can now utilize the Wolfram Repository gathering additional data for analysis

Challenges & Opportunities

Conclusions:

- By using Entities as Properties, otherwise flat data can be transformed into relational content
- The Wolfram Environment provides powerful tools for parsing and analyzing complex data around the UN SDGs
- Users can combine public data with curated Wolfram data to create multi-dimensional analysis

Challenges:

- Automation of the process requires relatively clean data to begin with. Variations in structure and content can often cause issues in linking Entities
- Data storage, management, and access can be an issue when attempting to use cloud based solutions on large processed data
- The Wolfram Interpreter function is not flawless as has been found in extended testing

Opportunities / Next Steps

- Explore integrating multiple data sources into a single repository to provide deeper analysis on SDG indicators by comparing published results
- Utilize “gold standard” sources to produce public Wolfram accessible data either in the published repositories or through Wolfram’s built in Entities
- Explore the experimental EntityExtend function which as discussed in the Live CEOing Twitch sessions to find similar opportunities to expand analysis of public data.
- Utilize database connection features introduced in Stephen Wolfram's Keynote

THANK YOU

We encourage you to follow our work in interact with us:

http://www.twitter.com/sdgcounting

https://medium.com/sdg-counting