(/)

Categories: Covid-19 (/Categories/Covid-19) | Feature Construction (/Categories/Feature%20construction) | Line Plot (/Categories/Line%20plot)

Data Mining COVID-19 Epidemics: Part 1

By: Janez Demšar, Apr 2, 2020

These days we are all following the statistics of COVID-19, looking at how our own country is faring and how it's comparing with other countries. Luckily, only a few have a statistically meaningful number of deaths (which solemnly reminds us of the difference between statistical and practical significance!), so we concentrate on the number of confirmed cases.

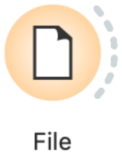
You're reading the first and most basic blog post from a series in which we will investigate this data using Orange. **Most** people are capable of doing **something** in Excel(-like programs), and **some** can do **everything** in Python with pandas and jupyter. I'll show you how **many** people can do **many** things in Orange.

Today, we will see how to get this data into Orange, draw some basic curves, and relate it to other data sources. Don't expect anything dramatic; this will be more about showing some creative ways of connecting a few widgets, and starting to explore the data about COVID-19 epidemics.

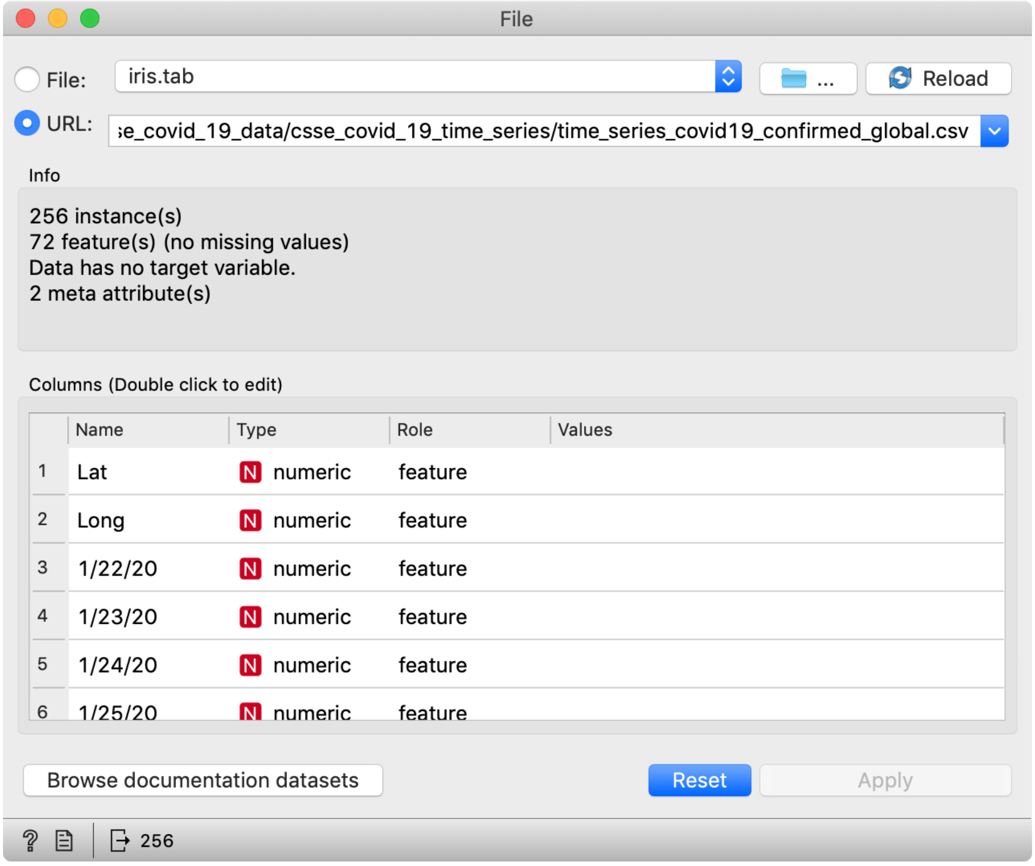
Getting the data

John Hopkins University collated some COVID-19 information in a machine-readable format and published it **on Github (https://github.com/CSSEGISandData/COVID-19)**. We will examine the table with **confirmed cases by regions and countries (https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_globa**

To get the numbers behind the above table, click "Raw". Or this link: **https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv (https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv)** Save the page and you'll have a file to play with.



(/blog_img/2020/2020-04-02-workflow-01-01.png) Easier still, Orange's File widget can load the data directly from the web, and handles a basic .csv file (for more fine-grained options, use CSV File Import).



○ File: iris.tab ... Reload

● URL: ;e_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv

Info

256 instance(s)
72 feature(s) (no missing values)
Data has no target variable.
2 meta attribute(s)

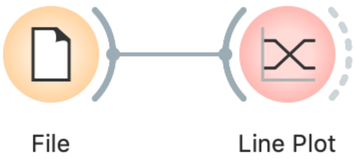
Columns (Double click to edit)

	Name	Type	Role	Values
1	Lat	N numeric	feature	
2	Long	N numeric	feature	
3	1/22/20	N numeric	feature	
4	1/23/20	N numeric	feature	
5	1/24/20	N numeric	feature	
6	1/25/20	N numeric	feature	

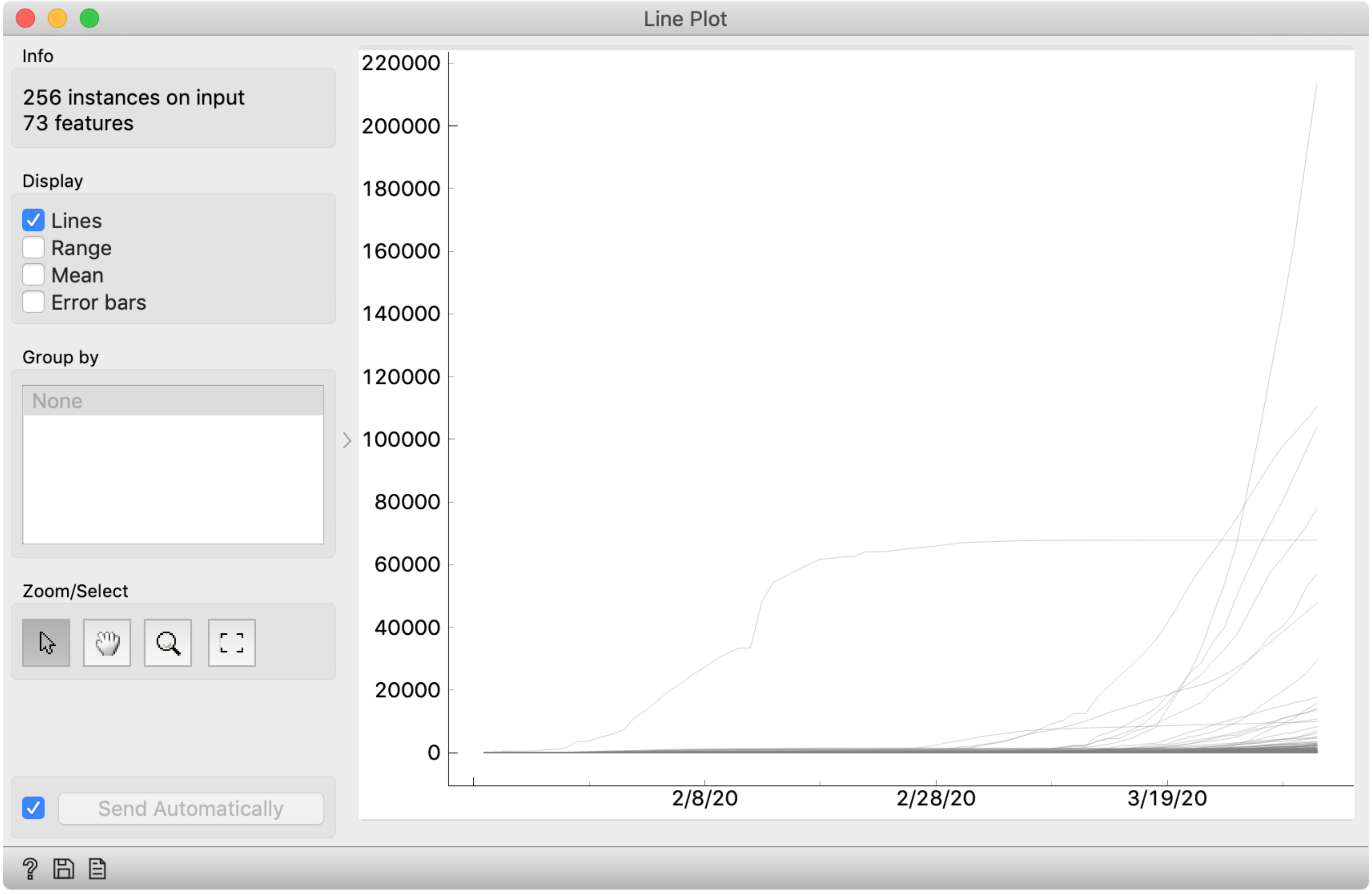
(/blog_img/2020/2020-04-02-file-widget.png) So, for starters, add the File widget to the canvas and copy the above link to the URL field. You can then connect it to a Data Table widget and check that everything's loaded OK. You should see a table with rows corresponding to regions and countries, and columns corresponding to dates, with two additional columns detailing region locations (latitude, longitude).

Note that this is live data, so your results will differ from those we show here.

Plot the usual plots



(/blog_img/2020/2020-04-02-workflow-01-02.png) Connect the File to a Line Plot widget to see the graph. By default, Line Plot shows means and ranges, while we are interested in raw lines. Let's click the checkboxes accordingly.



Info

256 instances on input
73 features

Display

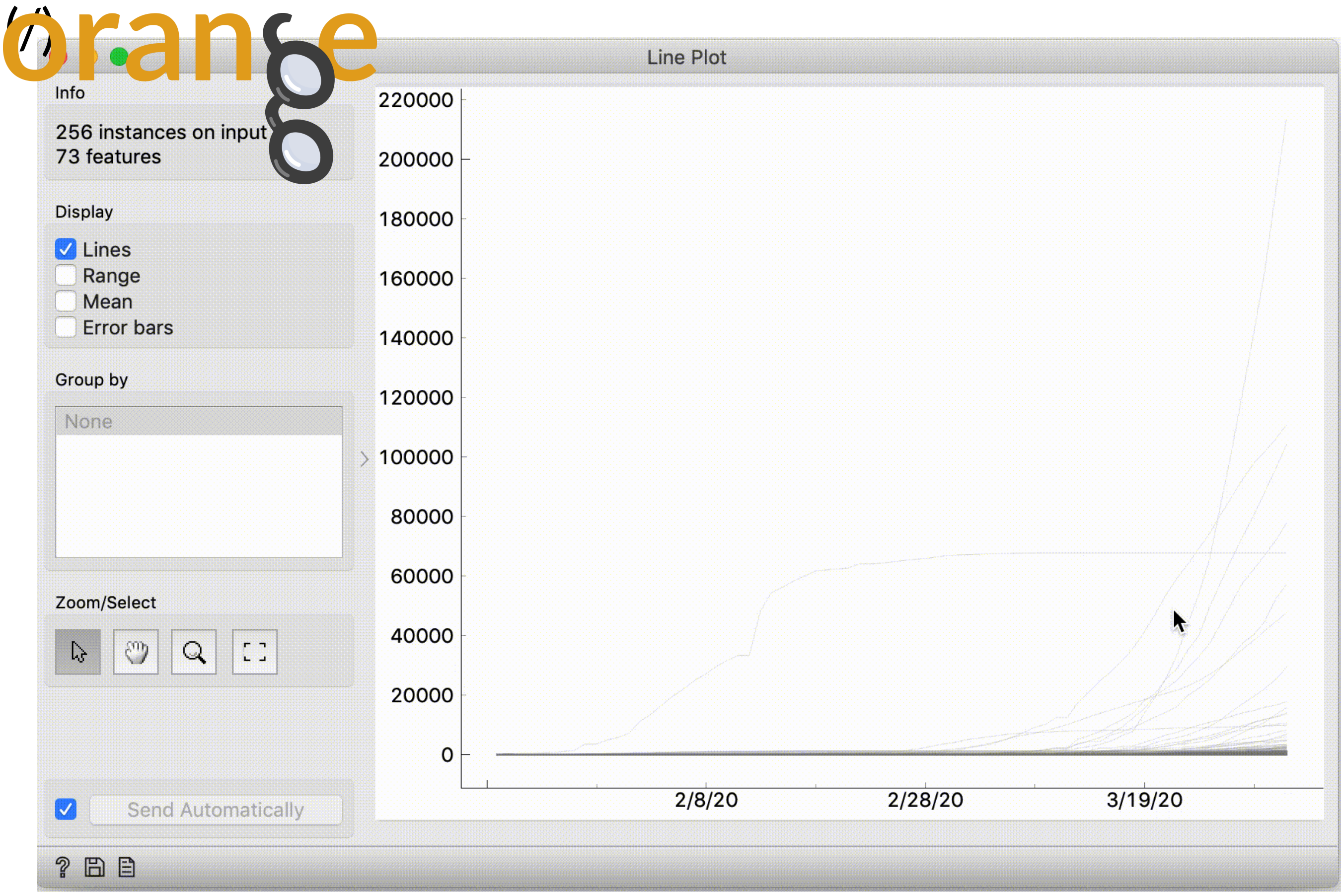
☑ Lines
□ Range
□ Mean
□ Error bars

Group by

None

☑ Send Automatically

(/blog_img/2020/2020-04-02-lineplot.png) Pretty boring. Anybody can do this in Excel. Orange allows us to play with these curves, though.



Info

256 instances on input
73 features

Display

☑ Lines
□ Range
□ Mean
□ Error bars

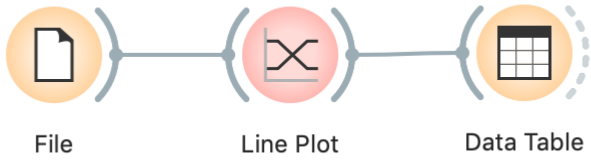
Group by

None

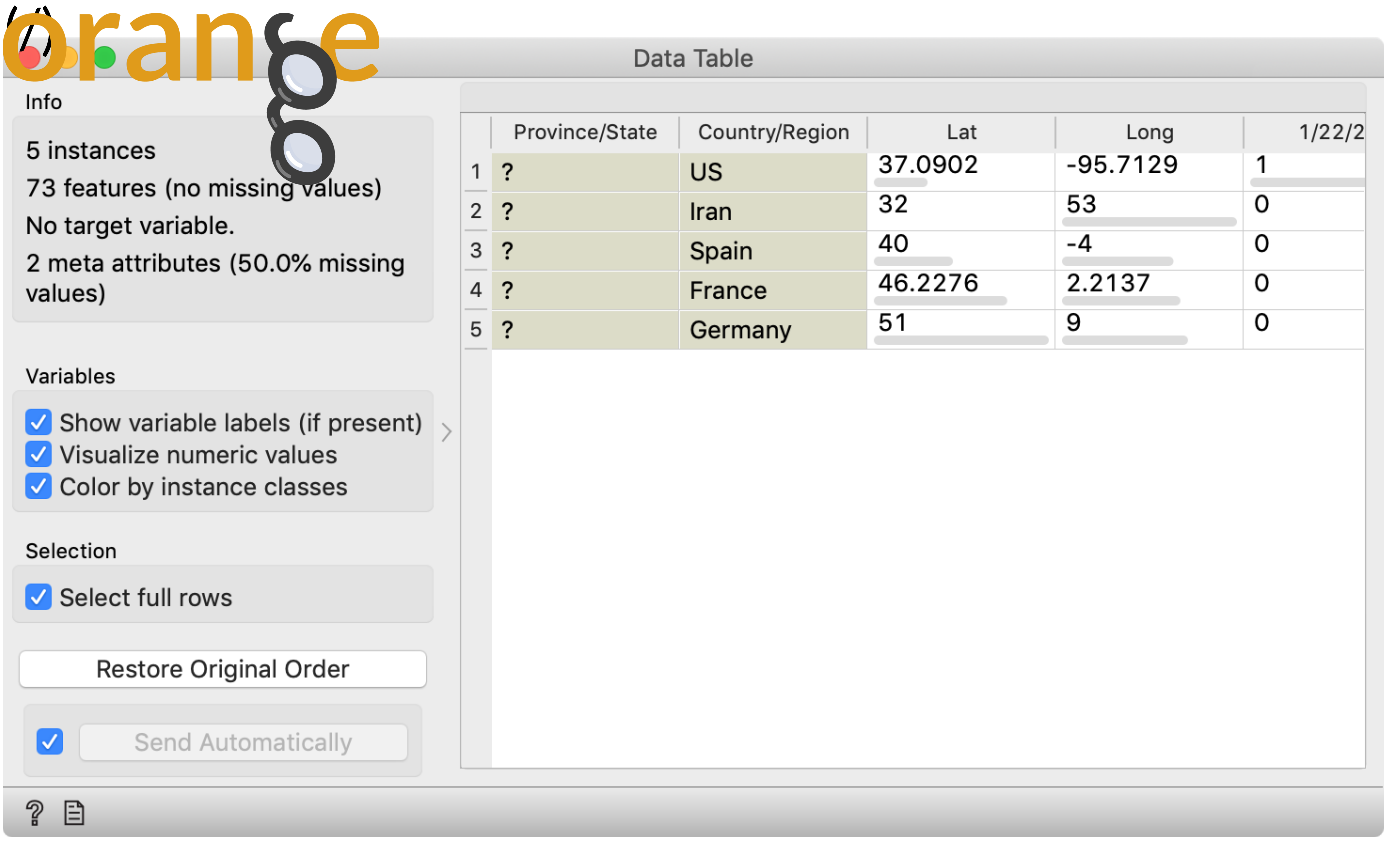
Zoom/Select

☑ Send Automatically

(/blog_img/2020/2020-04-02-select-line.gif) The curve that grows rapidly but then flattens out – is probably China, right? To check this, select it. It may be hard to click the curve, so select it by dragging a line across it.



(/blog_img/2020/2020-04-02-workflow-01-03.png) Then, connect the Line Plot widget to Data Table, and it will show the data for this curve. You'll learn that this is, of course, the Hubei province – where it all started.



Info

5 instances
73 features (no missing values)
No target variable.
2 meta attributes (50.0% missing values)

Variables

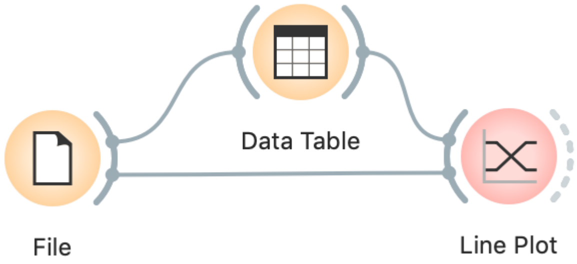
☑ Show variable labels (if present)
☑ Visualize numeric values
☑ Color by instance classes

Selection

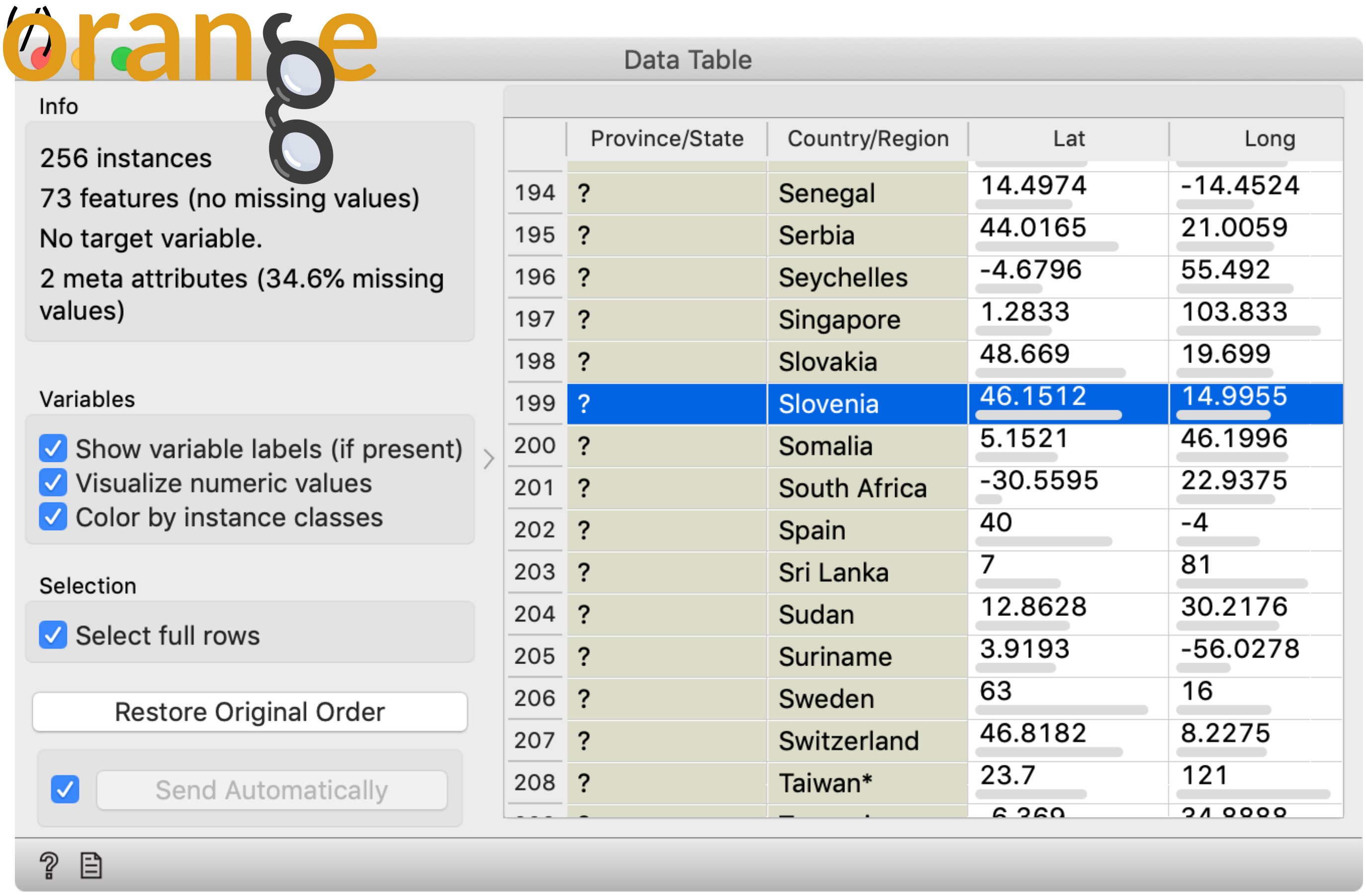
☑ Select full rows

	Province/State	Country/Region	Lat	Long	1/22/2
1	?	US	37.0902	-95.7129	1
2	?	Iran	32	53	0
3	?	Spain	40	-4	0
4	?	France	46.2276	2.2137	0
5	?	Germany	51	9	0

(/blog_img/2020/2020-04-02-steep-curves.png) What are the fast-rising curves? Select them by dragging a line across them, and check the Data Table widget: these are the US, Iran, Italy, Spain, Germany and France.



(/blog_img/2020/2020-04-02-workflow-01-04.png) We can do it the other way around: connect File to Data Table and Data Table to Line Plot, such that the Data Table sends its selected subset of data.



Data Table

Info

256 instances
73 features (no missing values)
No target variable.
2 meta attributes (34.6% missing values)

Variables

☑ Show variable labels (if present)
☑ Visualize numeric values
☑ Color by instance classes

Selection

☑ Select full rows

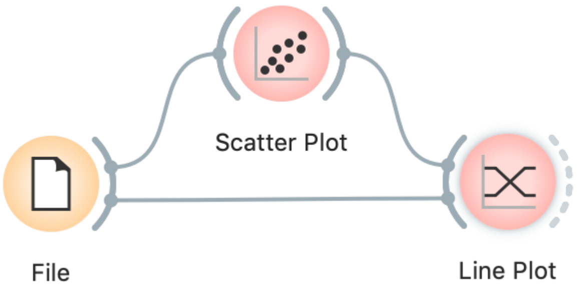
Restore Original Order

☑ Send Automatically

	Province/State	Country/Region	Lat	Long
194	?	Senegal	14.4974	-14.4524
195	?	Serbia	44.0165	21.0059
196	?	Seychelles	-4.6796	55.492
197	?	Singapore	1.2833	103.833
198	?	Slovakia	48.669	19.699
199	?	Slovenia	46.1512	14.9955
200	?	Somalia	5.1521	46.1996
201	?	South Africa	-30.5595	22.9375
202	?	Spain	40	-4
203	?	Sri Lanka	7	81
204	?	Sudan	12.8628	30.2176
205	?	Suriname	3.9193	-56.0278
206	?	Sweden	63	16
207	?	Switzerland	46.8182	8.2275
208	?	Taiwan*	23.7	121

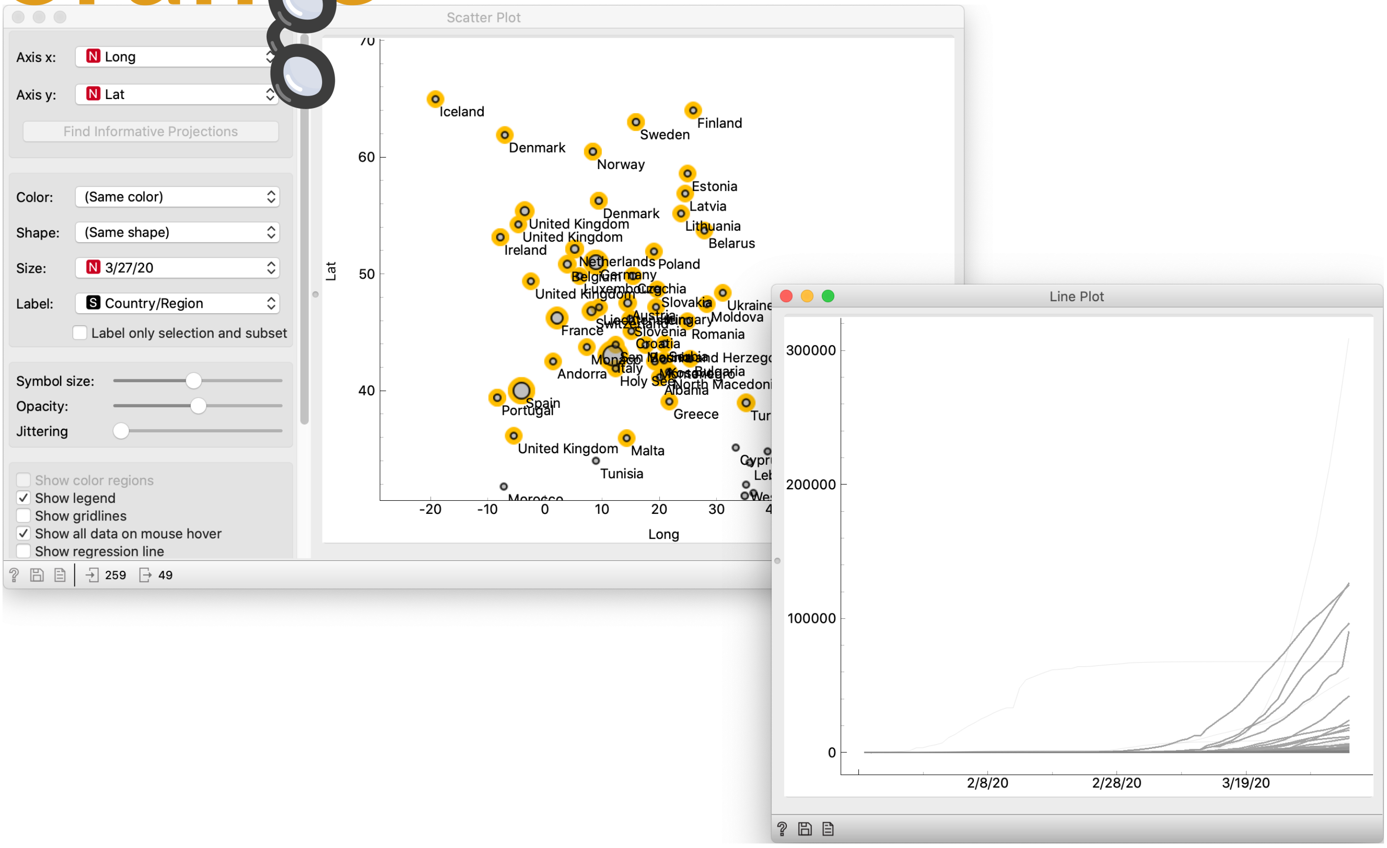
(/blog_img/2020/2020-04-02-select-country.png) In the Data Table, you can now find and click your favourite country. Line Plot will now highlight the curve for Slovenia (or whichever country you chose). You can also select multiple countries, like all the Chinese provinces. Or all European countries.

Well, the latter is a bit difficult; there is no data about the continent. There's a trick, though.



(/blog_img/2020/2020-04-02-workflow-01-05.png) Replace the Data Table widget with a Scatter Plot widget, and choose Long and Lat for x and y respectively. Zoom in on Europe and select the points by dragging across them. Line Plot will highlight the curves belonging to the countries chosen in the

Scatter Plot.



Axis x: Long
Axis y: Lat
Color: (Same color)
Shape: (Same shape)
Size: 3/27/20
Label: Country/Region
☐ Label only selection and subset
☐ Show color regions
☑ Show legend
☐ Show gridlines
☑ Show all data on mouse hover
☐ Show regression line

(/blog_img/2020/2020-04-02-europe-scatter-line-plot.png)

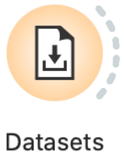
In the case of Slovenia (and possibly your choice as well), we couldn't really see its curve because it's negligent in comparison with others. The only thing we've learned is that the 327-million-man US have more cases of COVID-19 than a 2-million-man Slovenia. The same goes for Chinese provinces: the only one that sticks out is Hubei. Also, the "map" we improvised in Scatter Plot was a poor-man's map. Orange has proper geographical support (Orange3-Geo add-on), we'll look at it in-depth in a follow-up post.

The message here regards the essential difference between plotting curves and real data mining: static curves are dead; data mining is **interactive**.

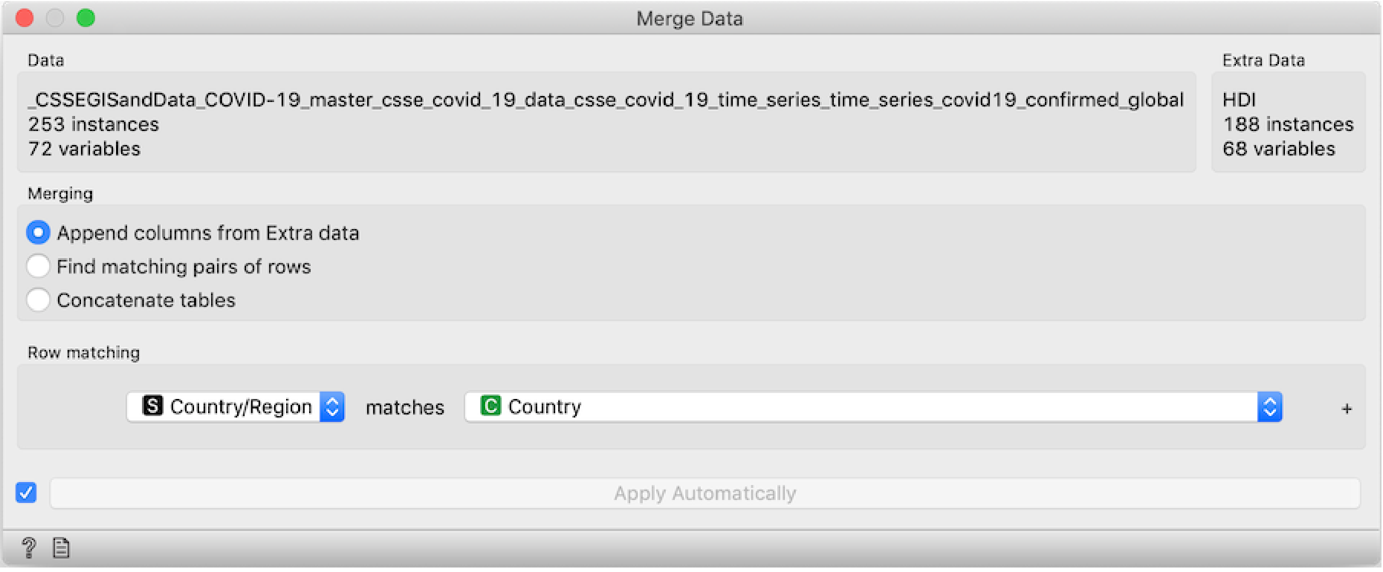
Connecting data sets

Let's solve the problem of negligible curves for small countries. This is not just about populations: the biggest problem is that confirmed cases are the result of testing, and testing strategies vary by country. Icelanders and Koreans test like crazy, while the number of tests in the US was (initially) so small it was comparable to that of Slovenia (and with it, the number of confirmed cases). The number of tests could thus **normalize** the number of confirmed cases, but this wouldn't do either; the testing isn't equivalent – some countries test more at random, while others save tests for groups at greater risk. And regardless, exact data about the number of tests is not readily available.

So, let's fallback to the number of cases per million inhabitants. Neither this nor the population of countries are present in the data provided by John Hopkins University. We do, however, have another data set at arm's reach: the population of countries (for year 2015) appears in the Worldbank's Human development index (HDI) data.



(/blog_img/2020/2020-04-02-workflow-02-01.png) To load it, use the Datasets widget, find and double-click the HDI data set.


Merge Data

Data

_CSSEGISandData_COVID-19_master_csse_covid_19_data_csse_covid_19_time_series_time_series_covid19_confirmed_global
253 instances
72 variables

Extra Data

HDI
188 instances
68 variables

Merging

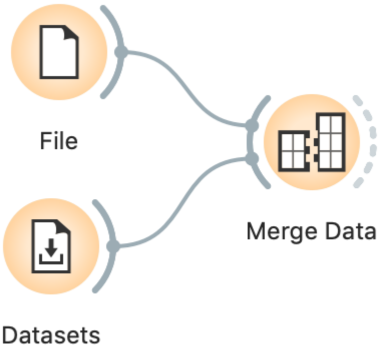
● Append columns from Extra data
○ Find matching pairs of rows
○ Concatenate tables

Row matching

S Country/Region matches C Country +

☑ Apply Automatically

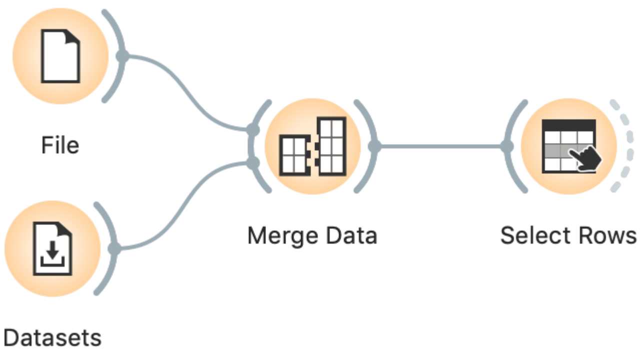
(/blog_img/2020/2020-04-02-merge-data.png) Now we're working with two data sets.



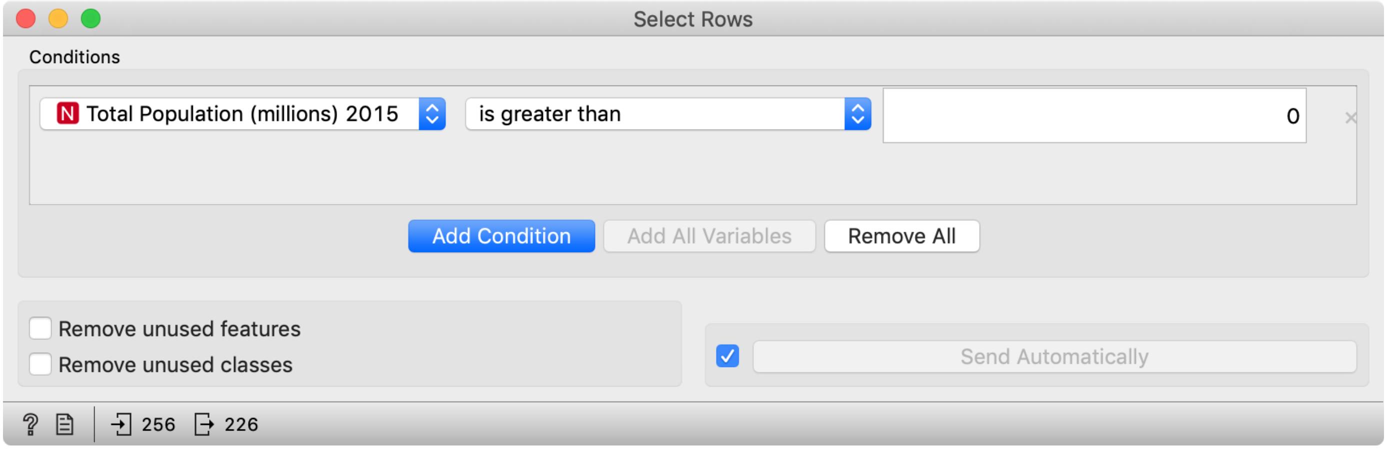
(/blog_img/2020/2020-04-02-workflow-02-02.png) Let's merge them using the Merge Data widget: connect both the File widget (with the John Hopkins data), and the Datasets widget (with the HDI data) to Merge Data. Make sure to match the "Country/Region" feature in COVID-19 data with the "Country" feature in HDI. It is also important to connect the widgets in this order, so File provides the main data and the Datasets widget augments it with additional annotation columns. If you do it incorrectly, double-click the connections to fix it.

The match is imperfect: some countries appear with different names, for instance, "*Russia*" from John Hopkins doesn't match the "*Russian Federation*" from HDI, and the John Hopkin's "*US*" and doesn't match HDI's "*United States*".

The population of Liechtenstein in millions with one decimal is 0.0; similar for Palau.



(/blog_img/2020/2020-04-02-workflow-02-03.png) To remove countries with unknown or zero population, we continue with Select Rows, where we set the condition "Total Population (millions) 2015 is greater than 0".



Conditions

N Total Population (millions) 2015 | is greater than | 0

☐ Remove unused features

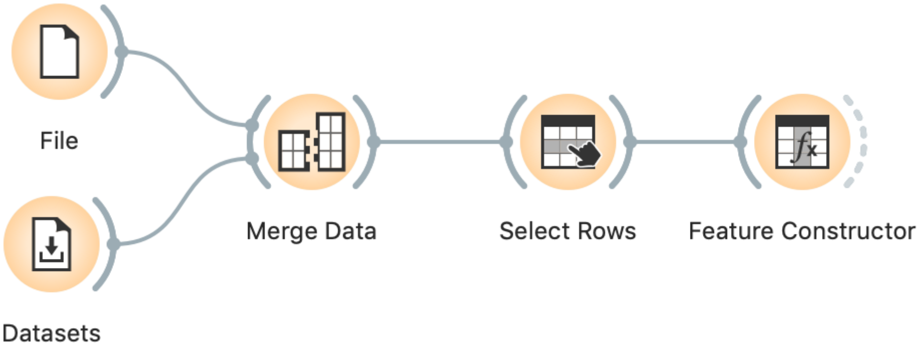
☐ Remove unused classes

☑ Send Automatically

256 226

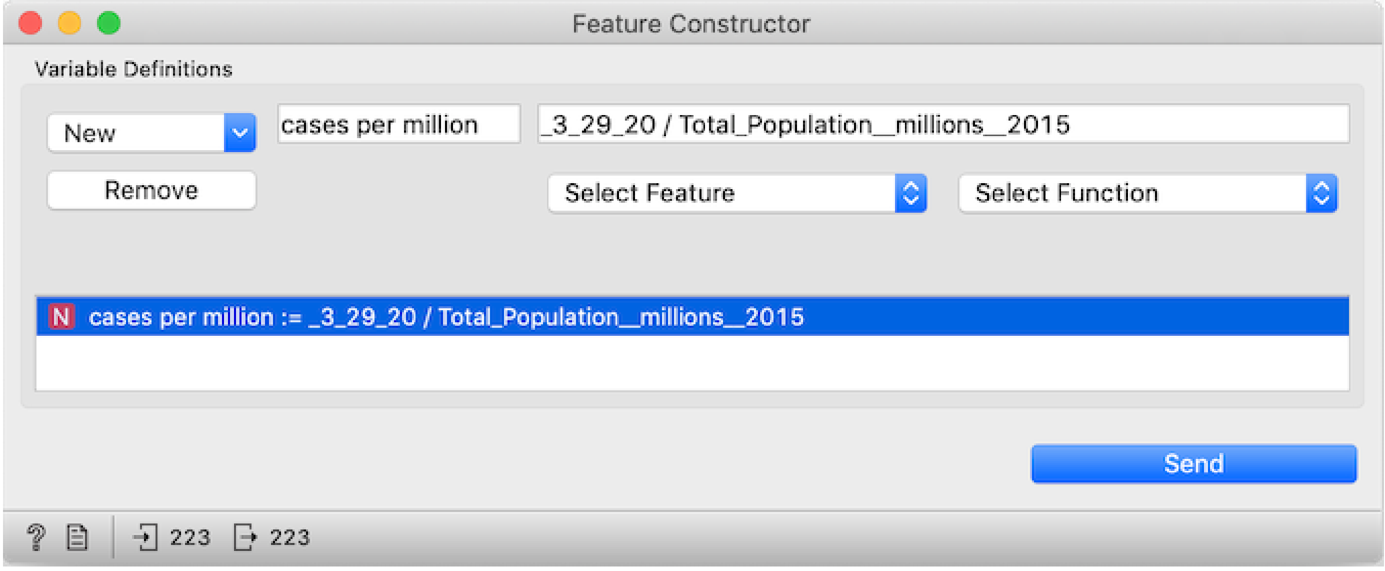
(/blog_img/2020/2020-04-02-select-rows.png)

Now for some constructive business.



(/blog_img/2020/2020-04-02-workflow-02-04.png) To get the number of cases on some particular day, we use the Feature Constructor widget, where we can input a formula to compute new data columns. In "New" we select Numeric, type `cases per million` as the name of the new column, Select Feature "3_29_20" (or whichever date we'd like), add `/` , and then select "Total_Population__millions__2015". As a shortcut, you can just copy

`_3_29_20 / Total_Population__millions__2015` into the “Expression…” line. (Note that an underscore precedes `3_29_20` : this way Orange knows that this is not a number but a name of a column.)


Feature Constructor

Variable Definitions

New | cases per million | _3_29_20 / Total_Population__millions__2015

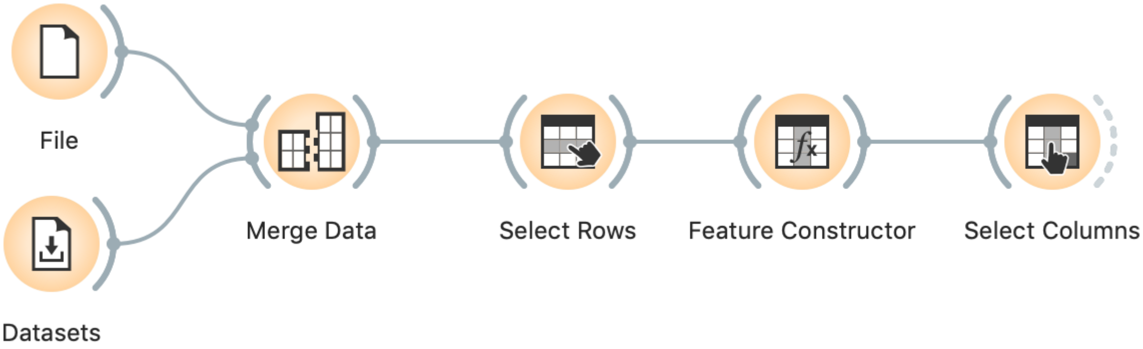
Remove | Select Feature | Select Function

N cases per million := _3_29_20 / Total_Population__millions__2015

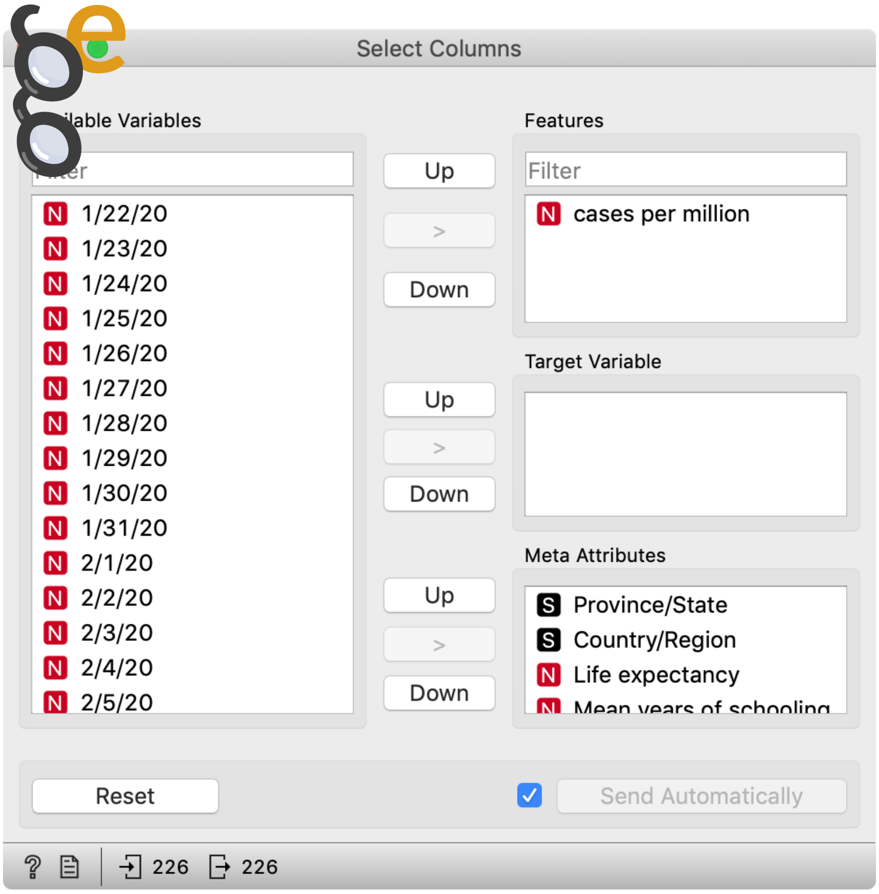
Send

223 223

(/blog_img/2020/2020-04-02-construct-features.png)
Things got crowded, so perhaps we can remove some columns.



(/blog_img/2020/2020-04-02-workflow-02-05.png) We add Select Columns, open it, and remove all columns we don’t need right now. We can just select all features (Ctrl-A or Cmd-A) and drag them to the left, and then drag “cases per million” back to features.



Select Columns

Available Variables

Filter

- N 1/22/20
- N 1/23/20
- N 1/24/20
- N 1/25/20
- N 1/26/20
- N 1/27/20
- N 1/28/20
- N 1/29/20
- N 1/30/20
- N 1/31/20
- N 2/1/20
- N 2/2/20
- N 2/3/20
- N 2/4/20
- N 2/5/20

Features

Filter

- N cases per million

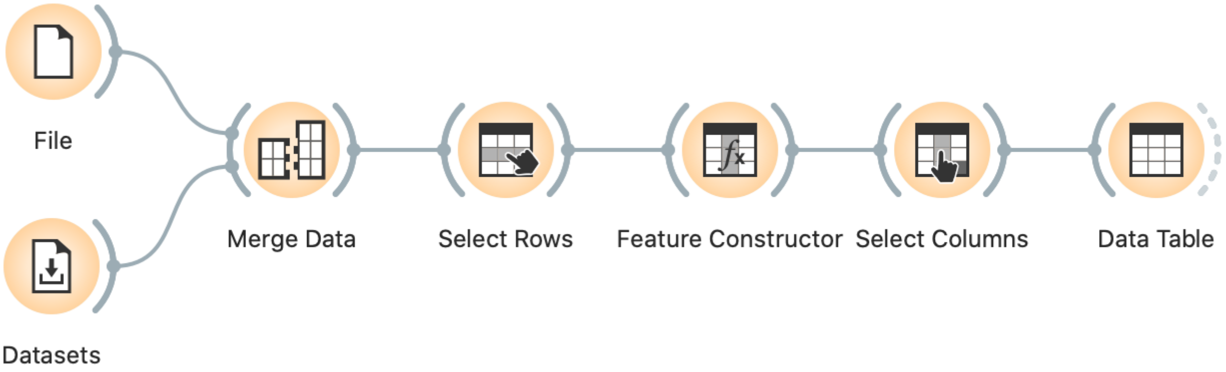
Target Variable

Meta Attributes

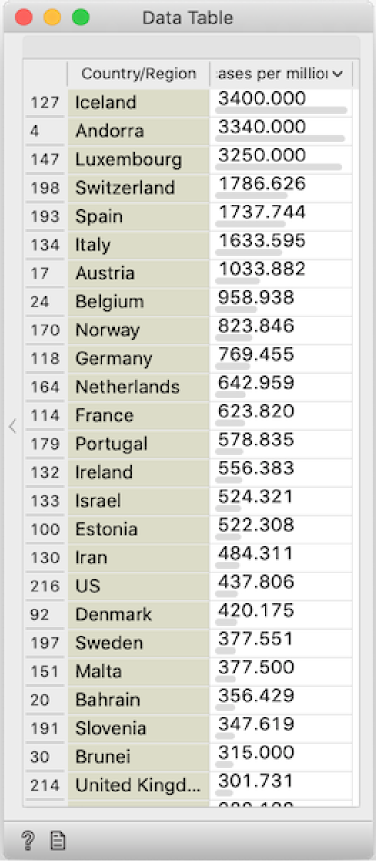
- S Province/State
- S Country/Region
- N Life expectancy
- N Mean years of schooling

(/blog_img/2020/2020-04-02-select-columns.png)

Finally, add and connect a Data Table.



(/blog_img/2020/2020-04-02-workflow-1-day.png)
Sort by the last column and, voilá, you've ranked countries the number of cases per a million inhabitants.



Data Table

	Country/Region	ases per millio
127	Iceland	3400.000
4	Andorra	3340.000
147	Luxembourg	3250.000
198	Switzerland	1786.626
193	Spain	1737.744
134	Italy	1633.595
17	Austria	1033.882
24	Belgium	958.938
170	Norway	823.846
118	Germany	769.455
164	Netherlands	642.959
114	France	623.820
179	Portugal	578.835
132	Ireland	556.383
133	Israel	524.321
100	Estonia	522.308
130	Iran	484.311
216	US	437.806
92	Denmark	420.175
197	Sweden	377.551
151	Malta	377.500
20	Bahrain	356.429
191	Slovenia	347.619
30	Brunei	315.000
214	United Kingd...	301.731

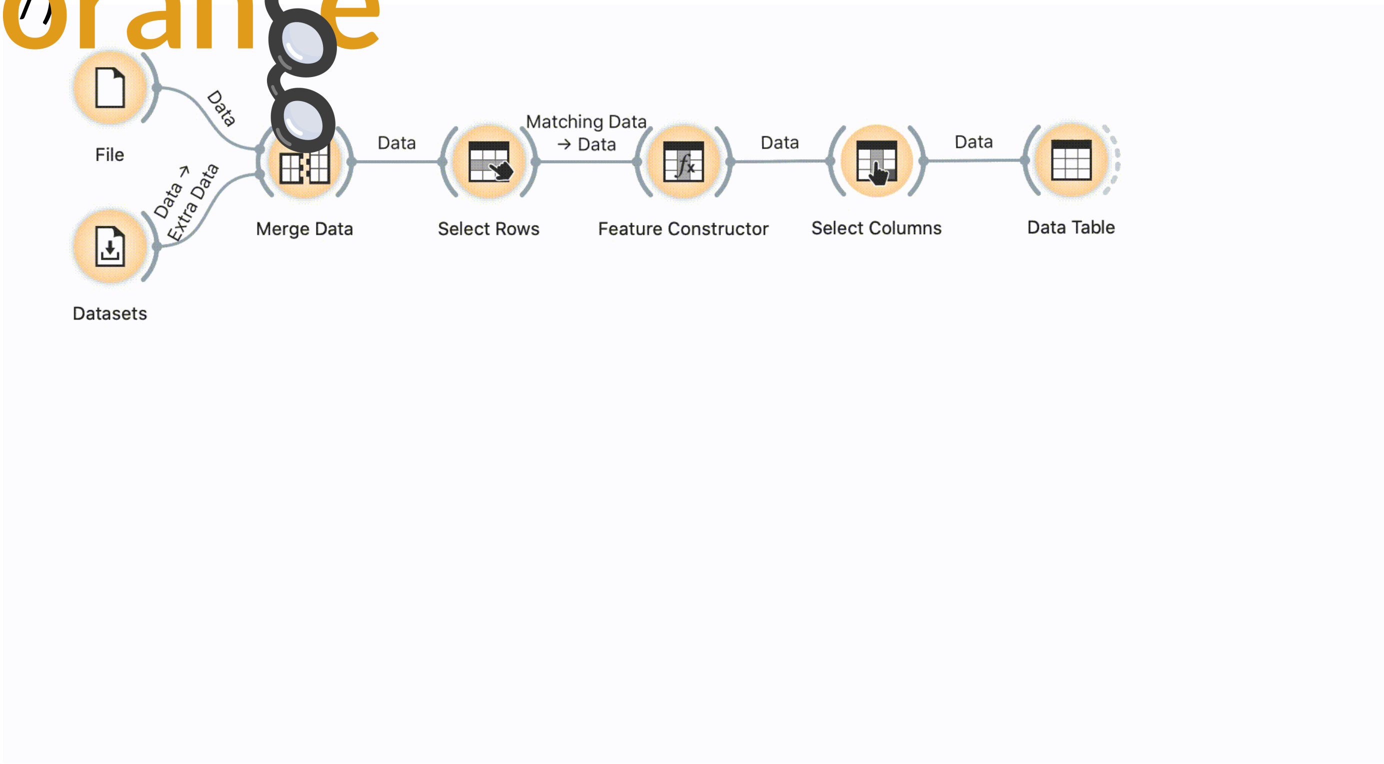
(/blog_img/2020/2020-04-02-1-day-table.png)
Missing in action: China. The data for China is split into provinces, yet every row is divided by the country's entire population. France seems to have a similar problem: it is given as a single country, but followed by external territories whose numbers are divided by the entire population. Also missing in action: Russia / Russian Federation, and US / United States, as we've previously explained.

Fixing these glitches will require some manual work.

Editing the data

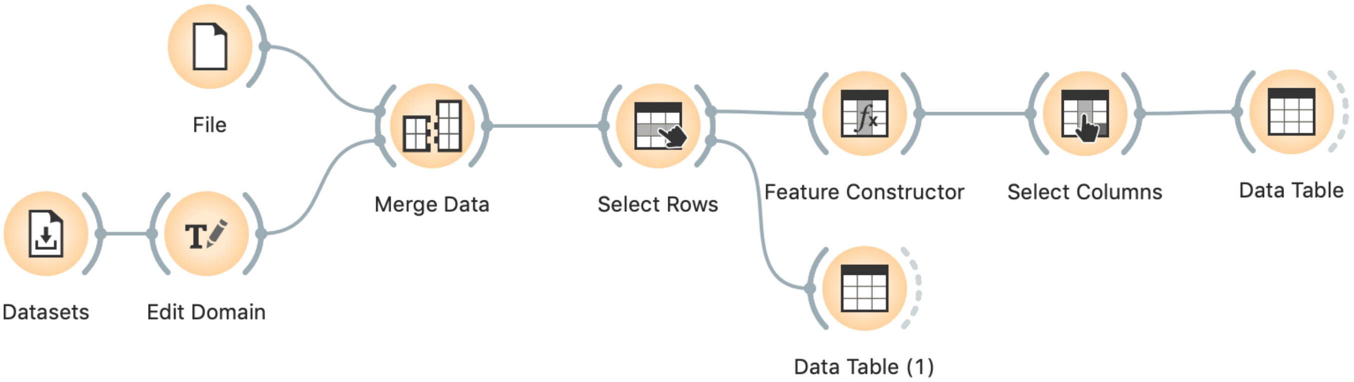
To fix the non-unique country names, we could open one of the two files in Excel and fix discrepancies, but there's no need for it: this is easier done in Orange, which will help us identify the missing countries. They are removed in the Select Rows widget. We hence connect a Data Table to Select Rows, double click the connection and do some editing: click the line between Matching Data and Data to remove it, and drag a line from Unmatched Data to Data.

(/)

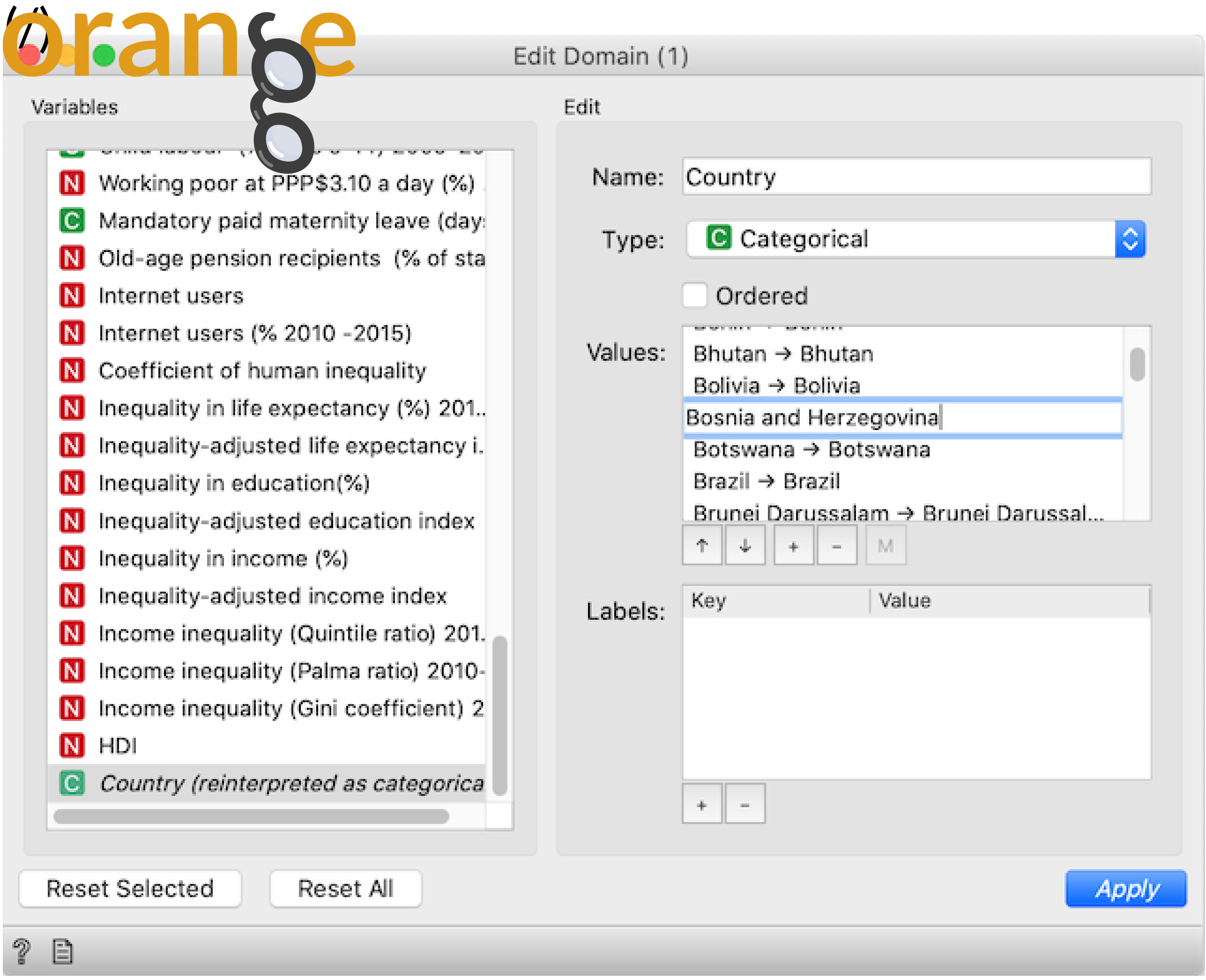


(/blog_img/2020/2020-04-02-select-rows-link-edit.gif)
This way, we can see which countries were filtered out. They seem to be absent due to either not being present in HDI, or having too small of a population.

Let's insert an Edit Domain widget between the Datasets widget and Merge Data.



(/blog_img/2020/2020-04-02-workflow-02-06.png)
We find and select column `Country`. Now change its type from Text to Categorical. Note, this is conceptually wrong: "*Country*" *is* a text variable, it contains a name and not a "category" of country, like the continent or the direction of writing. However, Edit Domain only lets us map its values if we change it to Categorical. And having it as categorical won't hurt.



Variables:
- N Working poor at PPP$3.10 a day (%)
- C Mandatory paid maternity leave (day:
- N Old-age pension recipients (% of sta
- N Internet users
- N Internet users (% 2010 -2015)
- N Coefficient of human inequality
- N Inequality in life expectancy (%) 201..
- N Inequality-adjusted life expectancy i..
- N Inequality in education(%)
- N Inequality-adjusted education index
- N Inequality in income (%)
- N Inequality-adjusted income index
- N Income inequality (Quintile ratio) 201..
- N Income inequality (Palma ratio) 2010-
- N Income inequality (Gini coefficient) 2
- N HDI
- C *Country (reinterpreted as categorica*

Name: Country

Type: C Categorical

□ Ordered

Values:
- Bhutan → Bhutan
- Bolivia → Bolivia
- Bosnia and Herzegovina
- Botswana → Botswana
- Brazil → Brazil
- Brunei Darussalam → Brunei Darussal...

Labels:

(/blog_img/2020/2020-04-02-edit-domain-hdi.png)

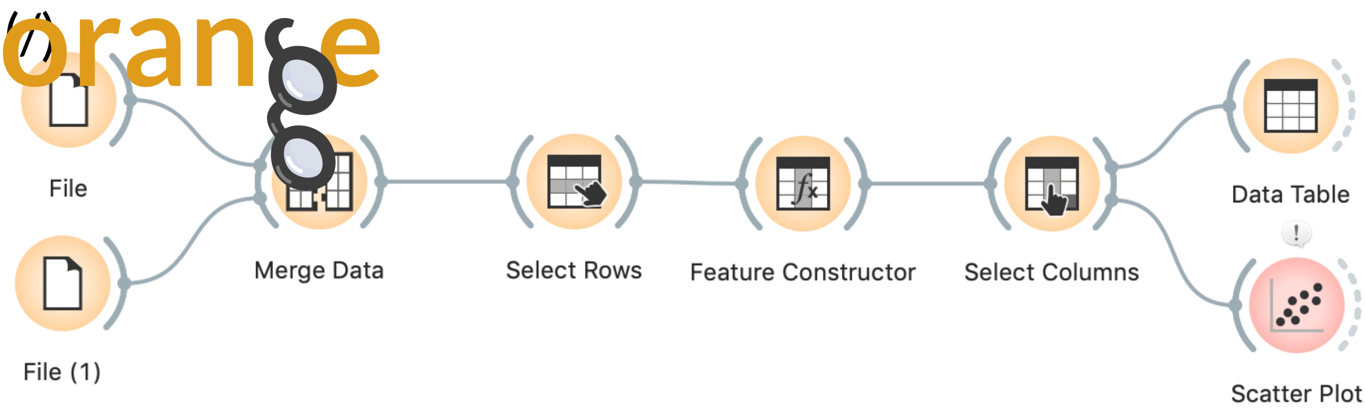
After changing the variable type, the widget will show another box, Values, with a mapping of values. Let's look at the countries in the new excluded rows Data Table widget. We see, for instance, "Bosnia and Herzegovina". We need to match to how John Hopkins data refers to countries, so go back to Edit Domain, and find "Bosnia and Herz.". Double click it and change it to "Bosnia and Herzegovina".

Similarly, Czech Rep. should become Czechia, Macedonia is now officially North Macedonia, Russian Federation is Russia, United States is US and Viet Nam is Vietnam. John Hopkins has "Korea, South" and HDI includes just Korea. Based on its high human development index and GDP, we can safely assume that this is the northern one. (Kidding, map this Korea to Korea, South, of course. There's no virus in the other Korea because Kim anticipated it before it even appeared and already invented a cure long ago.)

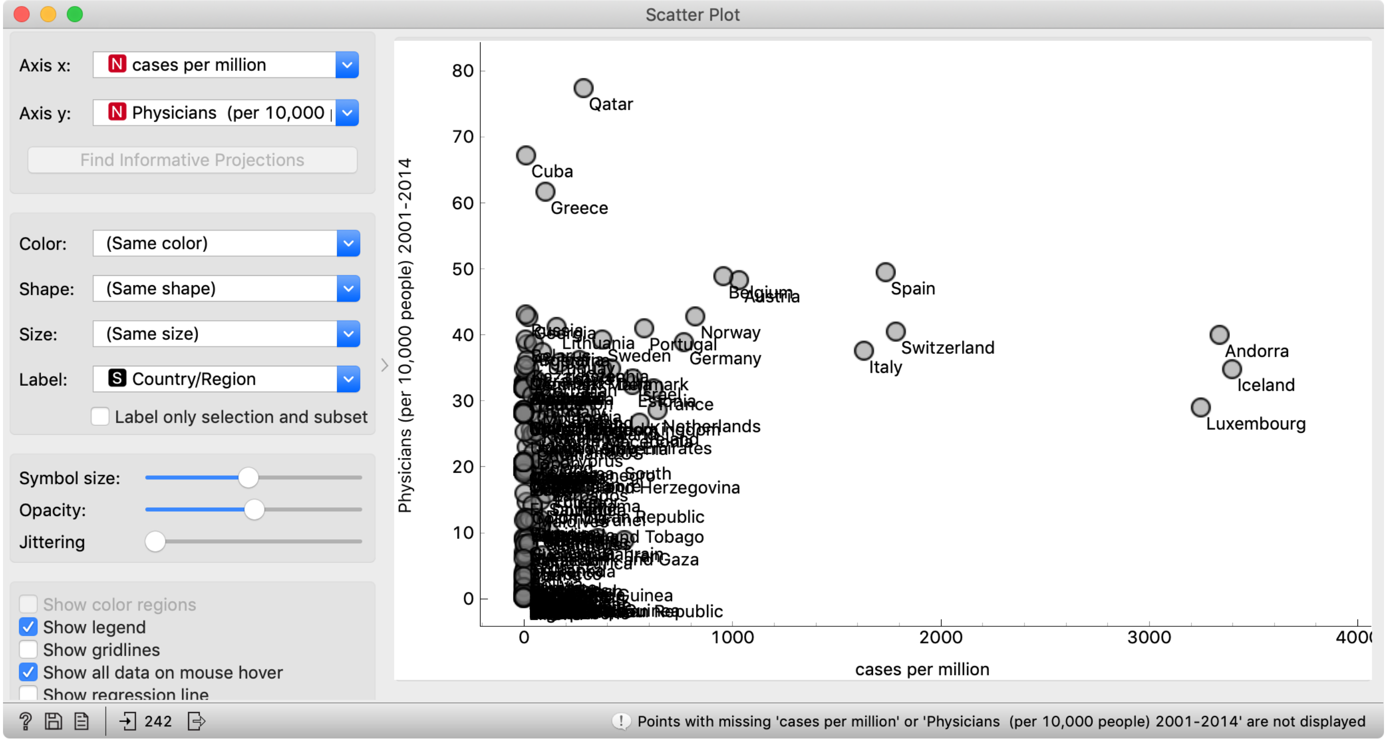
Now that I've taught you how to fish, I give you a fish for today: after editing these and all the other countries whose matches I found, I saved it to a **file you can download and use (http://file.biolab.si/datasets/HDI-fixed-for-jhu-covid.csv)**. Replace the Datasets widget with a File, and load it.

Analysis beyond population

Comparing this data with the World Bank's data is great because it allows us to relate the epidemiological data to data about particular countries.



(/blog_img/2020/2020-04-02-workflow-02-07.png) In Select Columns, we removed most of the features. Bring back all those that come from HDI. Now connect, say, a Scatter Plot and observe the relation between the number of cases per million and the number of physicians per ten thousand, which is the closest approximation to the capabilities of health systems.


Scatter Plot

Axis x: N cases per million

Axis y: N Physicians (per 10,000 |

Find Informative Projections

Color: (Same color)

Shape: (Same shape)

Size: (Same size)

Label: S Country/Region

□ Label only selection and subset

Symbol size:

Opacity:

Jittering

□ Show color regions

☑ Show legend

□ Show gridlines

☑ Show all data on mouse hover

□ Show regression line

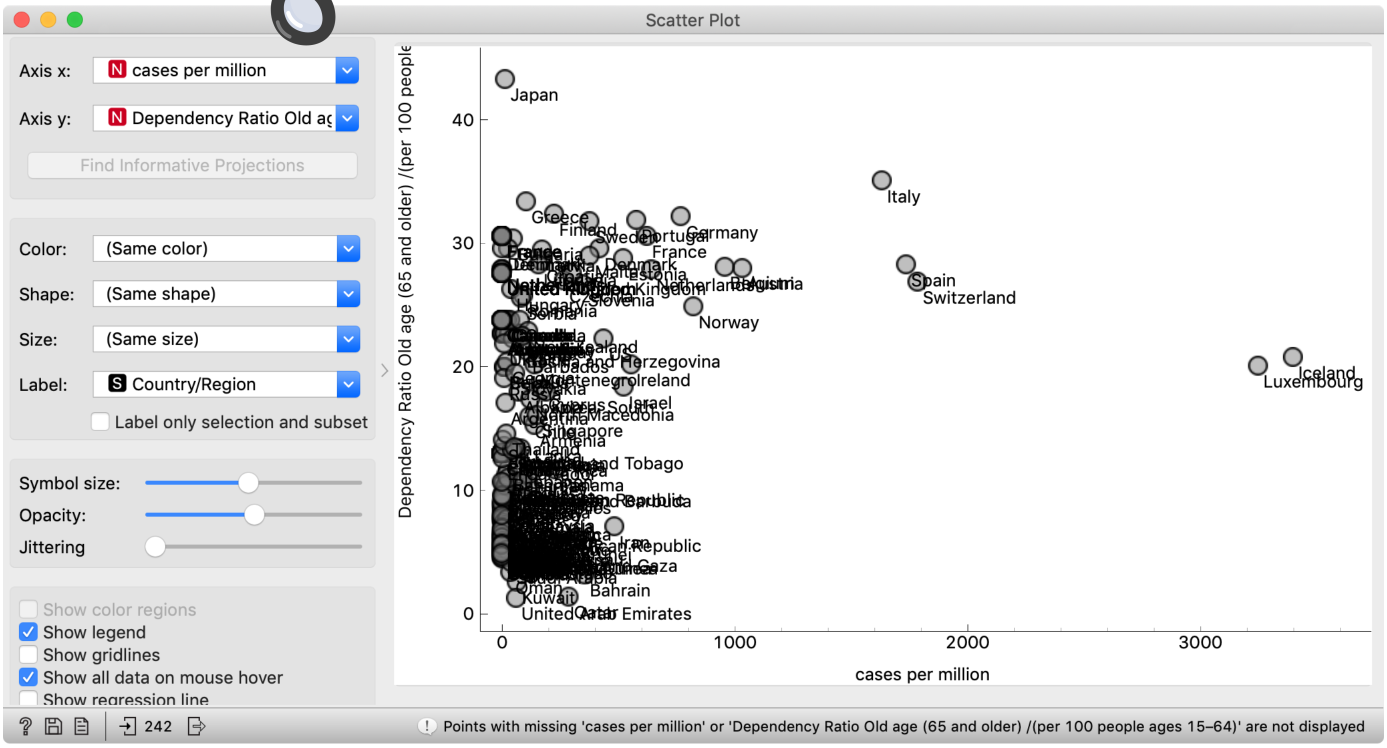
242

Points with missing 'cases per million' or 'Physicians (per 10,000 people) 2001-2014' are not displayed

(/blog_img/2020/2020-04-02-scatter-physicians.png) The conclusion here would be that you may want to find yourself in Qatar, Cuba or Greece right now. People are often surprised to see Cuba in such contexts, though Cuba has an excellent public health system and pretty good (though perhaps not entirely unbiased) free education. Seeing this graph, it shouldn't come as a surprise that **Cuban doctors are heading to Italy and other countries to help fight the epidemics (https://www.reuters.com/article/us-health-coronavirus-cuba/cuban-doctors-head-to-italy-to-battle-coronavirus-idUSKBN219051)**.

A seemingly related measure, Public health expenditure doesn't work here. It is expressed as a percentage of GDP, so it is adjusted for physicians' salaries, but not necessarily for the prices of (imported) equipment. Furthermore, higher spending may reflect a more expensive but not necessarily a more efficient health system.

Another potentially interesting factor to follow will be the age of the population. HDI tells us the number of people older than 65 years per 100 people between 15 and 64.



Scatter Plot

Axis x: N cases per million

Axis y: N Dependency Ratio Old ag

Find Informative Projections

Color: (Same color)

Shape: (Same shape)

Size: (Same size)

Label: S Country/Region

□ Label only selection and subset

Symbol size:

Opacity:

Jittering

□ Show color regions
☑ Show legend
□ Show gridlines
☑ Show all data on mouse hover
□ Show regression line

242

Points with missing 'cases per million' or 'Dependency Ratio Old age (65 and older) /(per 100 people ages 15–64)' are not displayed

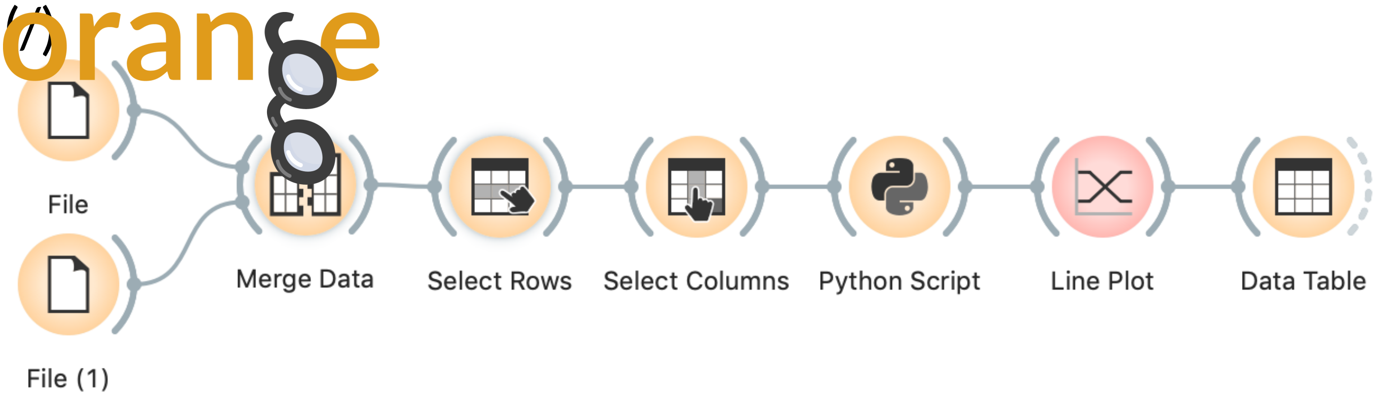
(/blog_img/2020/2020-04-02-scatter-dependency.png)

I should have used future tense; the population age **will** be interesting to observe with respect to mortality rate, but not enough countries have a sufficient number of deaths. Let's hope it stays this way.

We will eventually be able to (or – let's hope we won't) explore the relations between epidemics and other sociodemographic data. Generally, the problem with analyzing this data per country is currently that the disease has mostly spread (or been detected?) in developed countries, resulting in many correlations like the speed of spreading and percentage of urban populations, which may or may not reflect actual causation.

Back to curves

What about curves showing the number of confirmed cases per million? We computed the cases per million for a single day, not for all days. While Orange is a general-purpose data mining tool, some operations are too specific to be implemented in widgets. For such cases, we can program some processing in Python, using the Python Script widget.

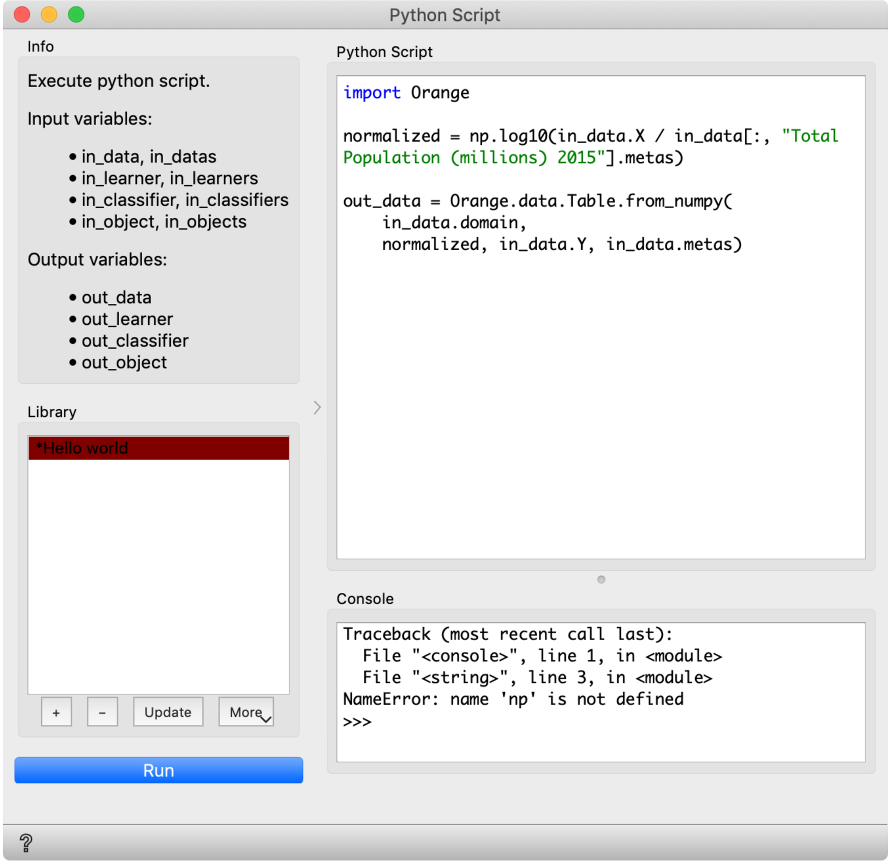


(/blog_img/2020/2020-04-02-workflow-03-01.png) The start of the sequence is the same as before: we merge the two data sources and remove countries with unknown or zero population.

In Select Columns, we move all variables except daily data from features to metas.

In Python Script, we type the following program and run it:

```
import numpy as np
import Orange

normalized = in_data.X / in_data[:, "Total Population (millions) 2015"].metas

out_data = Orange.data.Table.from_numpy(
    in_data.domain,
    normalized, in_data.Y, in_data.metas)
```



Execute python script.

Input variables:

- in_data, in_datas
- in_learner, in_learners
- in_classifier, in_classifiers
- in_object, in_objects

Output variables:

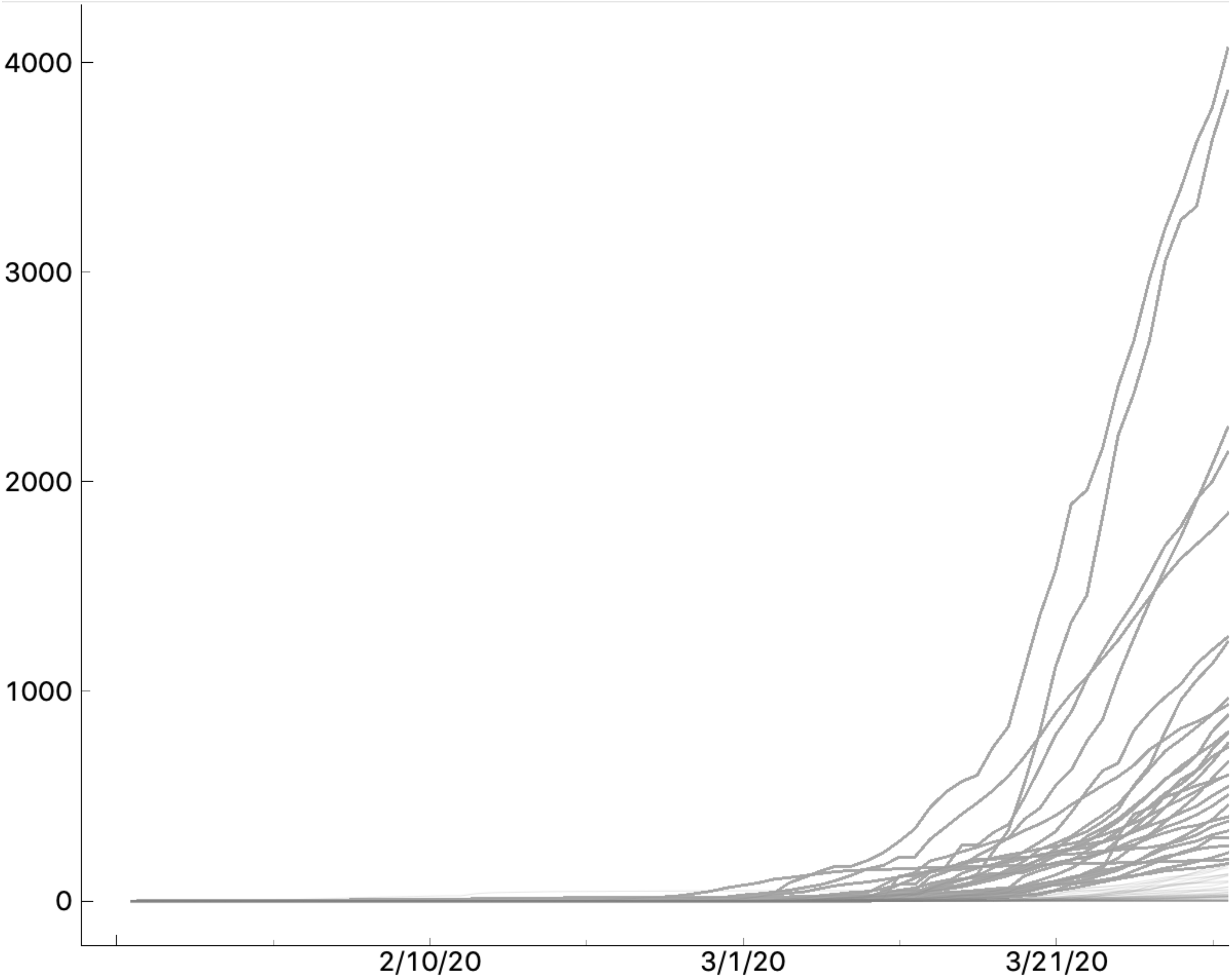
- out_data
- out_learner
- out_classifier
- out_object

```
import Orange

normalized = np.log10(in_data.X / in_data[:, "Total Population (millions) 2015"].metas)

out_data = Orange.data.Table.from_numpy(
    in_data.domain,
    normalized, in_data.Y, in_data.metas)
```

```
Traceback (most recent call last):
  File "<console>", line 1, in <module>
  File "<string>", line 3, in <module>
NameError: name 'np' is not defined
>>>
```

(/blog_img/2020/2020-04-02-python-script.png) With `in_data[:, "Total Population (millions) 2015"]` we take the table corresponding to the column with the population. Since population is now a meta attribute, we take `.metas`. We divide `in_data.X` with this column. Finally, we construct a new table named `out_data`, with the same domain (that is, the same variables), the normalized data, and the original data for target variable(s) and metas.

We get a similar line plot as before, just with proper scale. Observe it. There are two countries with 3000-4000 confirmed cases per million?! Which? Check yourself!



(/blog_img/2020/2020-04-02-lineplot-3.png) Would you like to see a plot with a logarithmic axis? And Line Plot doesn't support it? (Yet?) The beauty of using the Python script widget is that we can transform the data any way we want. Change the line that computes the normalized data to

```
normalized = np.log10(in_data.X / in_data[:, "Total Population (millions) 2015"].meta
s)
```

And we have a logarithmic axis (just with wrong labels).

Where from here?

This data is related to countries. Hence it would be nice to put it on a map. It also deals with time, and core Orange is not well-equipped for it. In the two follow-up posts, we will explore the add-ons for geographical data and for time series.

Orange is a multi-platform open-source machine learning and data visualization tool for beginners and experts alike. **Download Orange (https://orange.biolab.si/download)**, and load and explore your own data sets!

In addition to a variety of learning materials posted online in the form of **blog posts (https://orange.biolab.si/blog/)**, **tutorial videos (https://www.youtube.com/playlist?list=PLmNPvQr9Tf-ZSDLwOzxpvY-HrE0yv-8Fy)**, we've created a **Discord server. Join the community, tell us what you think! (https://discord.gg/FWrfeXV)**

6 Comments **orange** **Disqus' Privacy Policy** 1 **Login**

♡ **Recommend** 5 Tweet Share **Sort by Best**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Path Geek • 5 days ago

I, too, am having an error with the Python script. I typed it in, and then just copied and pasted your script.

Below the error message, in case that is of any benefit/value to you:

Python 3.7.6 (v3.7.6:43364a7ae0, Dec 18 2019, 14:18:50)

[Clang 6.0 (clang-600.0.57)] on darwin

Running script:

Traceback (most recent call last):

File "<console>", line 1, in <module>

File "<string>", line 4, in <module>

File "/Applications/Orange3.app/Contents/Framew...", line 787, in __getitem__

attributes, col_indices = self.domain._compute_col_indices(col_idx) see more

⌃ | ⌄ · Reply · Share ›

Drishti D days ago

View — uploads.disquscdn.com I couldn't change the HDI dataset country variable type in edit domain. That feature seems to be lacking in the orange data mining software provided in Anaconda Navigator.

⌃ | ⌄ · Reply · Share ›

Kach Wahed · 22 days ago

I didn't get any result after python script!

⌃ | ⌄ 1 · Reply · Share ›

Janez Demšar ➦ Kach Wahed · 21 days ago

What happenned, if anything? Did it show some error? Most importantly - did you click Run?

1 ⌃ | ⌄ · Reply · Share ›

Path Geek ➦ Janez Demšar · 5 days ago

I didn't see this before posting the error message separately. See my message from 4/27/2020

⌃ | ⌄ · Reply · Share ›

Ophir Shurany ➦ Kach Wahed · 21 days ago

Me too! Can't relate between Python script and Orange

⌃ | ⌄ · Reply · Share ›

✉ Subscribe Ⓓ Add Disqus to your siteAdd DisqusAdd ⚠ Do Not Sell My Data

Orange

FAQ (/faq/)
License (/license/)
Privacy (/privacy/)
Citation (/citation/)
Contact (/contact/)

Download

Windows (/download/#windows)

(/)

macOS (/download/#macos)

Community

Facebook (https://www.facebook.com/orangedm)

YouTube (https://www.youtube.com/channel/UClKKWBe2SCAEyv7ZNGhIe4g)

Twitter (https://twitter.com/orangedataminer)

Stack Exchange (https://datascience.stackexchange.com/questions/tagged/orange)

Discord (https://discord.gg/FWrfeXV)

Documentation

Get started (/getting-started/)

Widgets (/widget-catalog/)

Scripting (http://docs.orange.biolab.si/3/data-mining-library/)

Developers

GitHub (https://github.com/biolab/orange3)

Latest blog posts (/blog/)

20 Apr

Data Mining COVID-19 Epidemics: Part 3 (/blog/2020/2020-4-015-covid-19-part-3/)

13 Apr

Data Mining COVID-19 Epidemics: Part 2 (/blog/2020/2020-04-09-covid-19-part-2/)

02 Apr

Data Mining COVID-19 Epidemics: Part 1 (/blog/2020/2020-04-02-covid-19-basic/)