

TERM PROJECT DOCUMENTATION | CORRELATIONS WITH INCOME INEQUALITY ACROSS COUNTRIES

Data Engineering 1: Different Shapes of Data

MSc in Business Analytics 2021-2022 - Central European University (CEU)

PARIS TEAM

Ghazal Ayobi	-	1902457
Shah Ali Gardezi	-	2104924
Ádám József Kovács	-	2100773
Abigail Chen	-	2103124

Prepared for:

Mr. László Salló

DE1 Term Project 2 - Report

Overview of Data Flow

Data Sources:

- Kaggle dataset (from *World Income Inequality Database*)
<https://www.kaggle.com/mannmann2/world-income-inequality-database>
- World Bank API
- Eurostat Data

Indicators considered

- | | |
|---------------------|--|
| - Gini index | Kaggle data |
| - Unemployment rate | World Bank ID: SL.UEM.TOTL.ZS |
| - GDP per capita | World Bank ID: NY.GDP.PCAP.CD |
| - Life expectancy | Eurostat link https://appsso.eurostat.ec.europa.eu/nui/setupDownloads.do |
| - Education | Eurostat link: https://appsso.eurostat.ec.europa.eu/nui/setupDownloads.do |
| - Poverty | Eurostat link: https://appsso.eurostat.ec.europa.eu/nui/setupDownloads.do |

Analysis questions

- What is the association pattern between the extent of income inequality (measured by the GINI index) and the parameters below for each country in the year 2018.
 - Unemployment rate
 - GDP per capita
 - Life expectancy
 - Education
 - Poverty

Abstract

This report is a part of the second term project for the course, *Data Engineering 1: Different shapes of Data* at Central European University's *MSc Business Analytics program*. The project required us to choose the dataset of our choice and combine it with other (one or multiple) datasets and then perform in-depth analysis of the combined dataset to uncover hidden relationships and patterns. We were expected to apply concepts and tools learned in class about *SQL*, *APIs*, *KNIME* to answer our analytical question and present visualizations showing our findings.

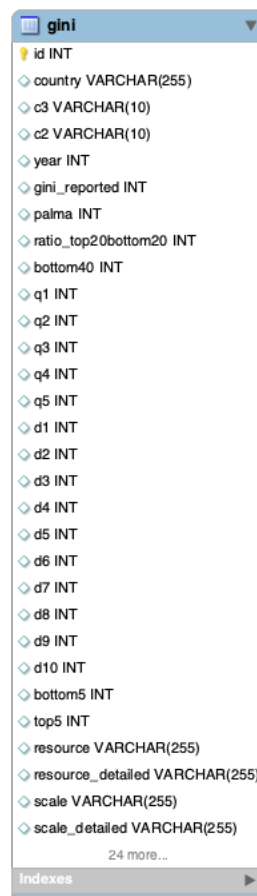
In order to answer our analysis question, we created a KNIME-based workflow, built an Extract-Transform-Load pipeline while utilizing data extracted from MySQL, the Eurostat API & the World Bank's World Development Indicators platforms. The choice of year 2018 is made due to the fact that not only is the data for this year readily available in all of our data sources but it is also available for all the countries we are looking to examine.

Our findings show that higher income countries and European countries tend to have the lowest income inequality. We also find not so surprisingly that there is a very strong correlation between the extent of poverty and income inequality across the countries, but other important socioeconomic variables, like unemployment rate and life expectancy are also significantly associated with income inequality in our sample of countries.

Data collection & Infrastructure

Kaggle:

As mentioned in the *Data Sources* section above, we collected data from 3 sources. Our main dataset is the one from *Kaggle* which is from *World Income Inequality Databases*, it contained 57 variables. The most important of all which we will be using for analysis of income inequality is the ***gini_reported***. In the data the income resource was calculated using either the *net income* of each country's household or *consumption* of the household. We choose *net income* because while most of the countries had income calculated on both parameters, majority of the countries had their source of income calculated on *net income*. We loaded this data set in *MySQL* using .csv file, here is EER diagram table.



id
INT
country VARCHAR(255)
c3 VARCHAR(10)
c2 VARCHAR(10)
year INT
gini_reported INT
palma INT
ratio_top20bottom20 INT
bottom40 INT
q1 INT
q2 INT
q3 INT
q4 INT
q5 INT
d1 INT
d2 INT
d3 INT
d4 INT
d5 INT
d6 INT
d7 INT
d8 INT
d9 INT
d10 INT
bottom5 INT
top5 INT
resource VARCHAR(255)
resource_detailed VARCHAR(255)
scale VARCHAR(255)
scale_detailed VARCHAR(255)

24 more...

Indexes

World Bank:

Since our analytical goal was to investigate the relationship of income inequality with other variables/metrics that we considered relevant in the global economics. To execute this, we looked at the world development indicators offered by the World Bank and chose ***Unemployment Rate & GDP per Capita*** for our analysis. This is our second dataset. We accessed these indicators via the [World Bank's public API service](#) and using the concepts learned from class we created the URL which will be used to access the data. Below figures

below illustrates the process of collecting data from World bank using *Postman* for unemployment and **GDP per capita**

GET

http://api.worldbank.org/v2/country/all/indicator/SL.UEM.TOTL.ZS?source=2&date=2018&format=json

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	source	2			
<input checked="" type="checkbox"/>	date	2018			
<input checked="" type="checkbox"/>	format	json			
	Key	Value	Description		

Body

Cookies (4)

Headers (14)

Test Results

Status: 200 OK

Time: 860 ms

Size: 14.45 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  [
2    {
3      "page": 1,
4      "pages": 6,
5      "per_page": 50,
6      "total": 266,
7      "sourceid": "2",
8      "sourcename": "World Development Indicators",
9      "lastupdated": "2021-10-28"
10   },
11   [
12     {
13       "indicator": {
14         "id": "SL.UEM.TOTL.ZS",
```

GET

http://api.worldbank.org/v2/country/all/indicator/NY.GDP.PCAP.CD?source=2&date=2018&format=json

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	source	2			
<input checked="" type="checkbox"/>	date	2018			
<input checked="" type="checkbox"/>	format	json			
	Key	Value	Description		

Body

Cookies (4)

Headers (15)

Test Results

Status: 200 OK

Time: 823 ms

Size: 12.62 KB

Save Response

Pretty

Raw

Preview

Visualize

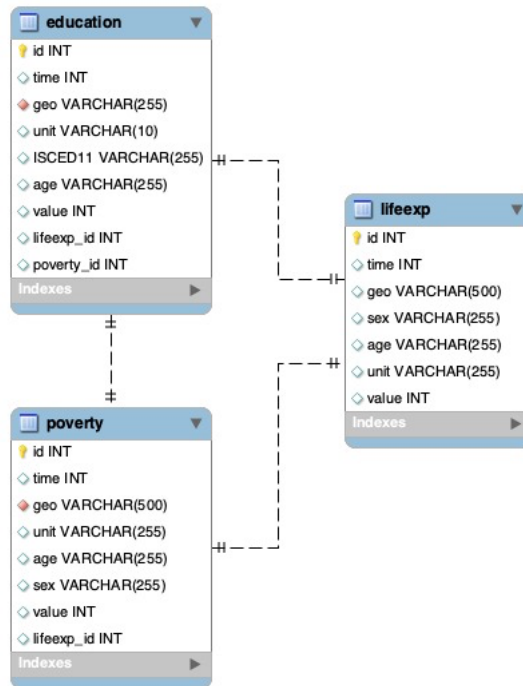
JSON

```
1  [
2    {
3      "page": 1,
4      "pages": 6,
5      "per_page": 50,
6      "total": 266,
7      "sourceid": "2",
8      "sourcename": "World Development Indicators",
9      "lastupdated": "2021-10-28"
10   },
11   [
12     {
13       "indicator": {
14         "id": "NY.GDP.PCAP.CD",
```

Eurostat:

Our third source of data was *Eurostat*. The metrics we wanted to analyze against the **Gini** included, **Life Expectancy, Education & Poverty**. We downloaded three different datasets for the countries as .csv file (The links for these datasets are mentioned above).

We loaded all of these datasets as separate tables in a new schema of *MySQL* named, [eurostat_data.sql](#) and created a relational database. The EER diagram below depicts the results.



We then created a stored procedure (**eurostat_data_warehouse**) which outputs a data warehouse combining all three datasets in a table called **eurostat_data_table** using inner join on country and year. We also created two *Views* in *MySQL*, one for *life expectancy* and other one for *poverty* in order to see year wise trend of both metrics.

A slight caveat while working with Eurostat datasets was that it lacked the country code needed to join the countries' information with those from Kaggle and World Bank datasets, correctly. We overcame this problem by downloading [ISO2](#) from Eurostat and then using Eurostat API link. It helped us by incorporating universal codes for each country. The figure below illustrates how we used the Eurostat API using Postman

GET http://ec.europa.eu/eurostat/wdds/rest/data/v2.1/json/en/demo_mlexpec?precision=1&sex=T&unit=YR&time=2018&age=Y_LT1 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> precision	1	
<input checked="" type="checkbox"/> sex	T	
<input checked="" type="checkbox"/> unit	YR	
<input checked="" type="checkbox"/> time	2018	
<input checked="" type="checkbox"/> age	Y_LT1	
Key	Value	Description

Body Cookies Headers (10) Test Results Status: 200 OK Time: 370 ms Size: 1.95 KB Save Response

Pretty Raw Preview Visualize JSON

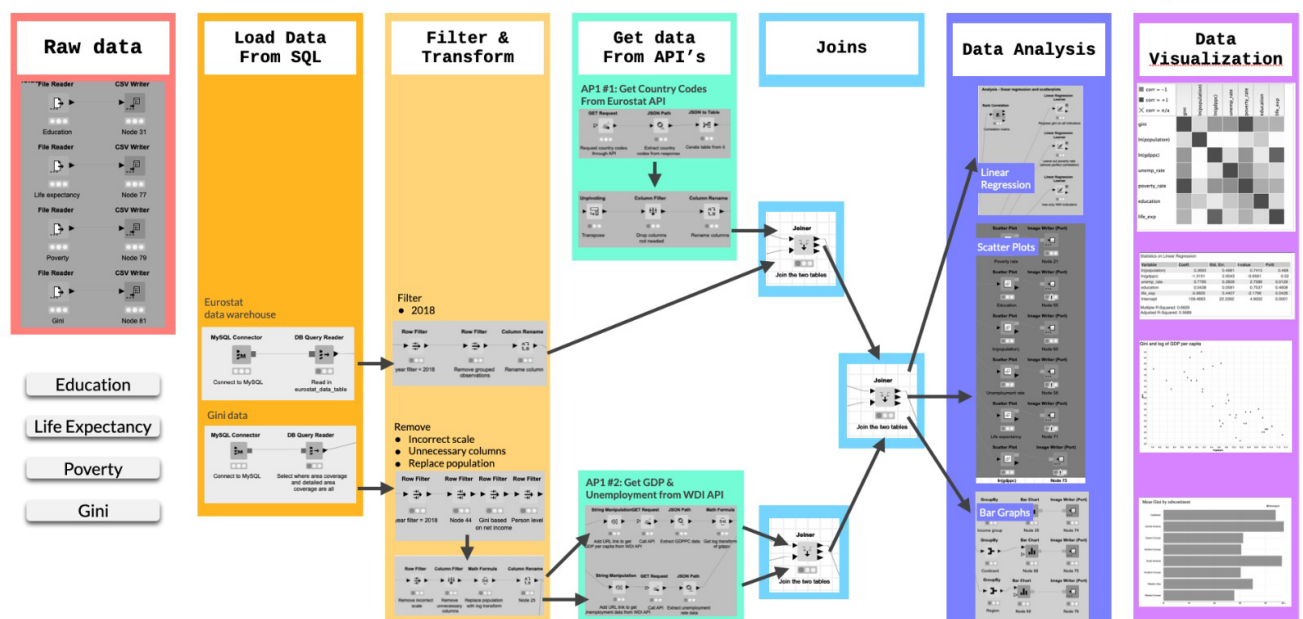
```

1  {
2    "version": "2.0",
3    "label": "Life expectancy by age and sex",
4    "href": "http://ec.europa.eu/eurostat/wdds/rest/data/v2.1/json/en/demo_mlexpec?precision=1&sex=T&unit=YR&time=2018&age=Y_LT1",
5    "source": "Eurostat",
6    "updated": "2021-04-28",
7    "status": {
8      "13": "ep",
9      "14": "ep",
10     "16": "ep",

```

KNIME WORKFLOW

KNIME WORK FLOW DIAGRAM



Preparatory phase:

users must ensure MySQL and KNIME are functioning

Read and write the files (for reproducibility)

To ensure that the data is readily available for reproducibility, we read the files using *File Reader node* and used the *CSV Writer node* to save data in the working directory.

Setting the SQL workflow:

User must select the MySQL Query connector node and put their MySQL credentials to proceed. Since we have two MySQL sources for the data, first we load *MySQL Connector node* to load Eurostat data from data warehouse. Second, we load *Gini* data from MySQL using Gini table.

Filter & transform:

We perform data filtering and transformation as a part of our data cleaning process. Since we have two datasets here, we perform the cleaning for each as under:

For Eurostat:

- Filter the data for the year 2018 using *Row Filter Node*
- Remove grouped observations such as *European Union* using Row Filter
- Renaming columns using *Column Filter Node* e.g **lifeExp** for *life_exp*

For Gini data:

- Use *Row Filter Node* to filter, year as 2018 , area coverage as *All Area Coverage*, Resource as *Net income*, reference unit as *Person level*, exclude square root measurement from scale detailed
- Use *Column Filter Node* to keep the target variables
- Use *Math formula Node* to calculate natural log to population (since the population values for each country is quite large)
- Use *Column Rename Node* to rename the columns

Get Data from API:

Eurostat API: To ensure Eurostat API is selected, we configured *GET Request* node in order to request for Eurostat country codes. As this request yields the data in JSON format, we used the *JSON Path* node to extract data which is country name and country codes. To read the JSON data into the table we used *JSON To Table* node. Since the JSON table is in wide format we use *Unpivoting* node to transpose the data. The *Column Filter* is then used to drop irrelevant column e.g RowIDs. Lastly, we used *Column Rename* to columns as country codes and country names.

World Bank API: Since we have to access two sources of data from World Bank API one for *Unemployment* and other for *GDP Per Capita*. The steps to load and transform each were quite similar. We first use *String Manipulation* node to connect API to our **Gini** table, then we use *Get Request* node to request for data in a JSON format. Then for **GDP Per Capita** we used natural log in order to standardize our values, while for **Unemployment** we used *Column Rename* adjust the column name for unemployment

Joins:

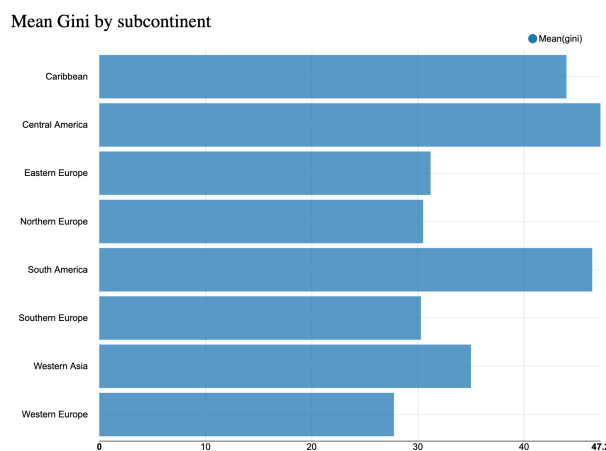
We used three *Joiner* nodes to combine our data

- Joiner 1: joins data from Eurostat data warehouse from MySQL to Eurostat API for country codes
- Joiner 2: joins the output data from **GDP Per Capita** and **Unemployment**
- Joiner 3: connects Joiner 1 (Eurostat data warehouse) and Joiner 2 (**GDP Per Capita and Unemployment**)

Visualization & Analysis:

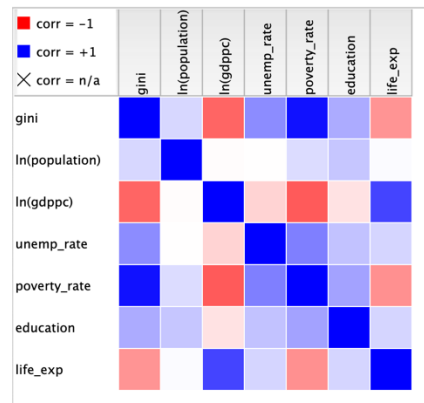
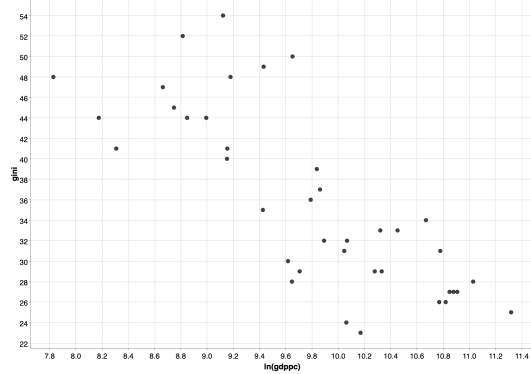
We end our pipeline with three types of visualization/analysis. First, we created bar charts on all the countries in our sample grouped by region(continent), subcontinent and the income groups. Based on these charts, displayed below, we see that among greater regions (continents), the countries of Europe are the least inequal, Asia comes second, while the Americas come third.

Regarding income level of countries, high income countries tend to be more inequal, while there seem to be not too big differences among lower or upper-middle income countries. Finally, among subcontinents, Central America is the least, while Western Europe is the most equal (lowest gini) in terms of income distribution. This is shown also on the graph below.



Next, as it was our main goal, we were interested in our selected variables correlation with the gini index. To this end, we visualized their joint distribution on scatter plots. All these graphs were saved and can be found in the 'graphs' folder on github. We also created a correlation heatmap that captures these relationships. What we can conclude from these is that the poverty indicator is extremely positively correlated with the gini index, but there is a less strong, but also positive correlation with the unemployment rate as well. Regarding life expectancy and even more so GDP per capita, on the other hand, we see negative correlations with income inequality. A scatter and the heatmap here serve as illustration:

Gini and log of GDP per capita



Finally, we also ran three regressions: In the first one, we included all our explanatory variables. Since the Eurostat data is only available for European countries, our sample of countries in this regression was limited to those, where we had all these information available. What we find is that the poverty indicator is the only variable with a significant coefficient. This is unsurprising given the extent of correlation uncovered earlier. But this way, it is possible that it takes away the significance of other variables that may also affect the Gini coefficient, so in the next regression we dropped poverty to see how the model changes. While this second model has a lower R-squared (66% compared to 82%), here there are two statistically significant coefficients as well, even at 5% significance level, namely unemployment rate and life expectancy. The output of this regression looks as follows:

Variable	Coeff.	Std. Err.	t-value	P> t
ln(population)	0.3693	0.4981	0.7415	0.468
ln(gdppc)	-1.3151	2.0043	-0.6561	0.52
unemp_rate	0.7795	0.2826	2.7586	0.0129
education	0.0438	0.0581	0.7537	0.4608
life_exp	-0.9605	0.4407	-2.1796	0.0428
Intercept	109.4663	22.3392	4.9002	0.0001

Multiple R-Squared: 0.6626
Adjusted R-Squared: 0.5689

Finally, we also wanted to run a regression on our entire sample of countries, so in the third regression, we included variables only from the Kaggle and the WDI data that are available for all countries (population, GDP per capita and unemployment rate). Here we found the latter two variables to be statistically significant.

Task Distribution:

- Ghazal Ayobi – Eurostat Data Warehouse on MySQL & Eurostat API-Knime workflow
- Shah Ali Gardezi – Income Inequality Index Schema on MySQL & World Bank API-Knime workflow
- Ádám József Kovács - Knime Visualizations, Regression models, Data loading, Debugging
- Abigail Chen – Presentation & Report writing