SUSTAINABLE DEVELOPMENT

A data analysis by Angela Selgado and Evan Strait

OUR AGILE APPROACH

1. Defining our topic/ planning in Jira
2. Webscrapping useful data
3. Compiling information in Python
4. Analysis of our findings in SQL

DEFINITION

What is Sustanable Development,
and how can we calculate it?

The UNESCO Definition of Sustainable Development:

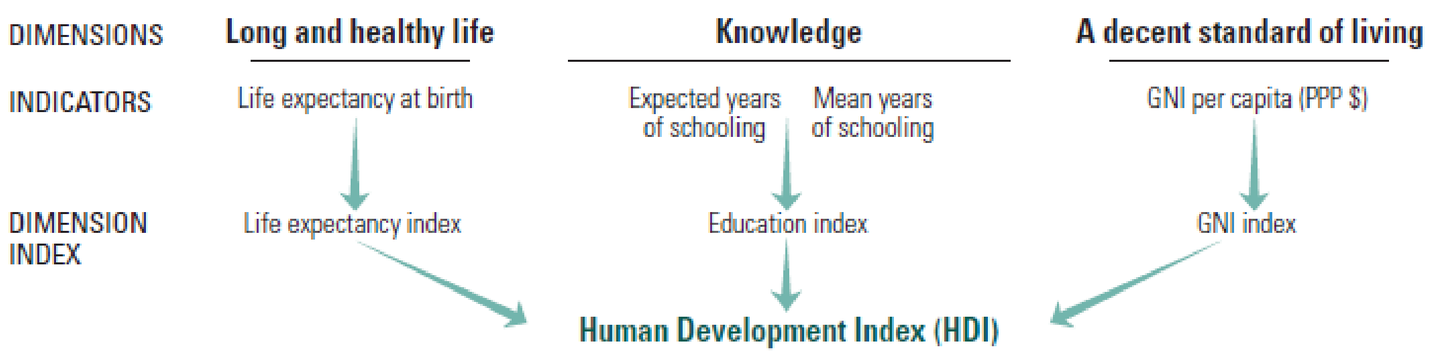
Environmentally sound development, which meets the needs of the present without compromising the ability of future generations to meet their own needs.

CALCULATION

How have NGOs calculated sustainable development, and what are the criticisms?

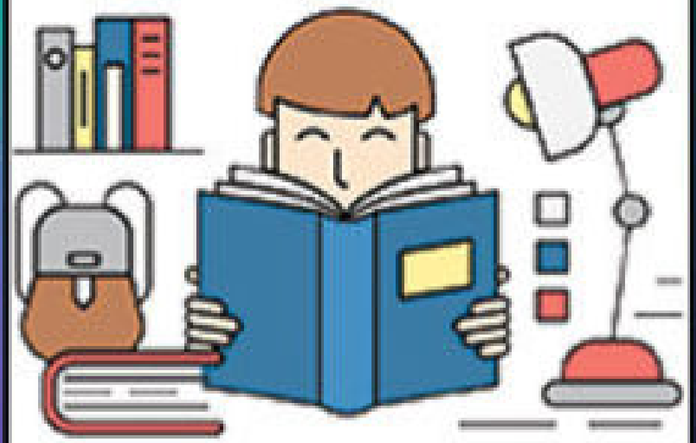
First, we calculate a country's Human Development Index:

Human Development Index (HDI)



Then, we adjust our calculations for planetary pressures...

LIFE EXPECTANCY
AT BIRTH



AVERAGE EDUCATION
LEVEL

STANDARD OF LIVING
(GNP PER CAPITA)

(HDI)
HUMAN DEVELOPMENT INDEX

CALCULATION

How have NGOs calculated sustainable development, and what are the criticisms?

The United Nations Development Programme

The UN's planetary pressure-adjusted HDI:

PHDI is the level of human development adjusted by carbon dioxide emissions per person (production-based) and material footprint per capita to account for the excessive human pressure on the planet.

However, their algorithm is experimental and we've found opportunities for development.

THE FORMULA

1. Combining HDI calculations
2. With new adjustments for Planetary Impact

LIFE EXPECTANCY INDEX

$$= \frac{\mathrm{LE} - 20}{85 - 20}$$

LEI is equal to 1 when life expectancy at birth is 85 years, and 0 when life expectancy at birth is 20 years.

EDUCATION INDEX

$$= \frac{\mathrm{MYSI} + \mathrm{EYSI}}{2}$$

Mean Years of Schooling Index (MYSI)

$$= \frac{\mathrm{MYS}}{15}$$

Expected Years of Schooling Index (EYSI)

$$= \frac{\mathrm{EYS}}{18}$$

INCOME INDEX

$$= \frac{\ln(\mathrm{GNIpc}) - \ln(100)}{\ln(75,000) - \ln(100)}$$

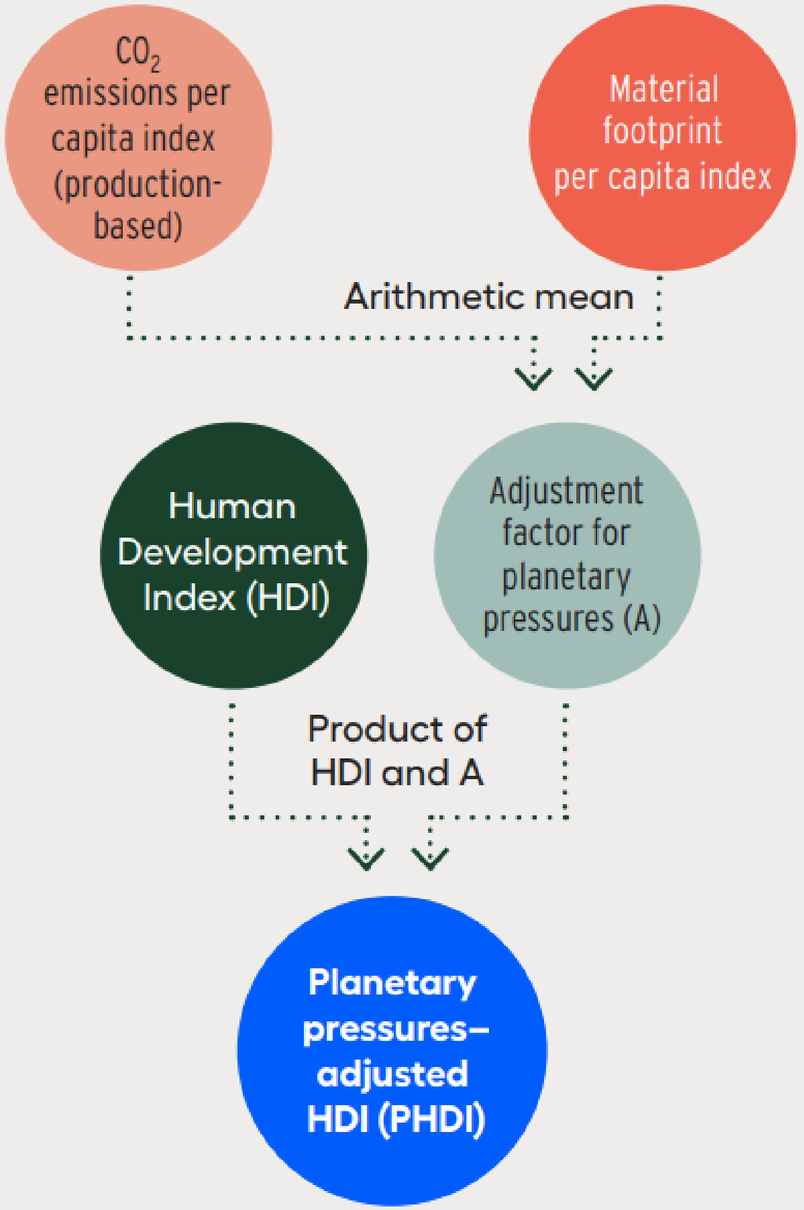
Income index is 1 when GNI per capita is $75,000 and 0 when GNI per capita is $100.

PLANETARY PRESSURE

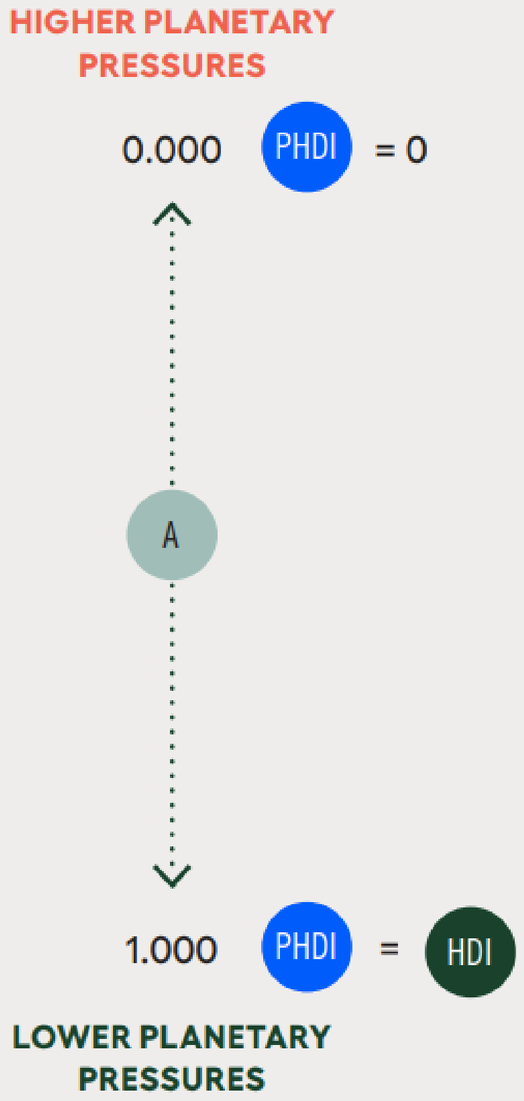
$$A\ index = \frac{maximum - observed\ value}{maximum - minimum}$$

The “observed value” here is a nation’s resource use (tons of Material Footprint per capita) or emissions (tons of CO2 per capita), the “maximum” is the highest value recorded for any nation in any year, and the “minimum” is 0.

PHDI is created by multiplying the HDI by an adjustment factor



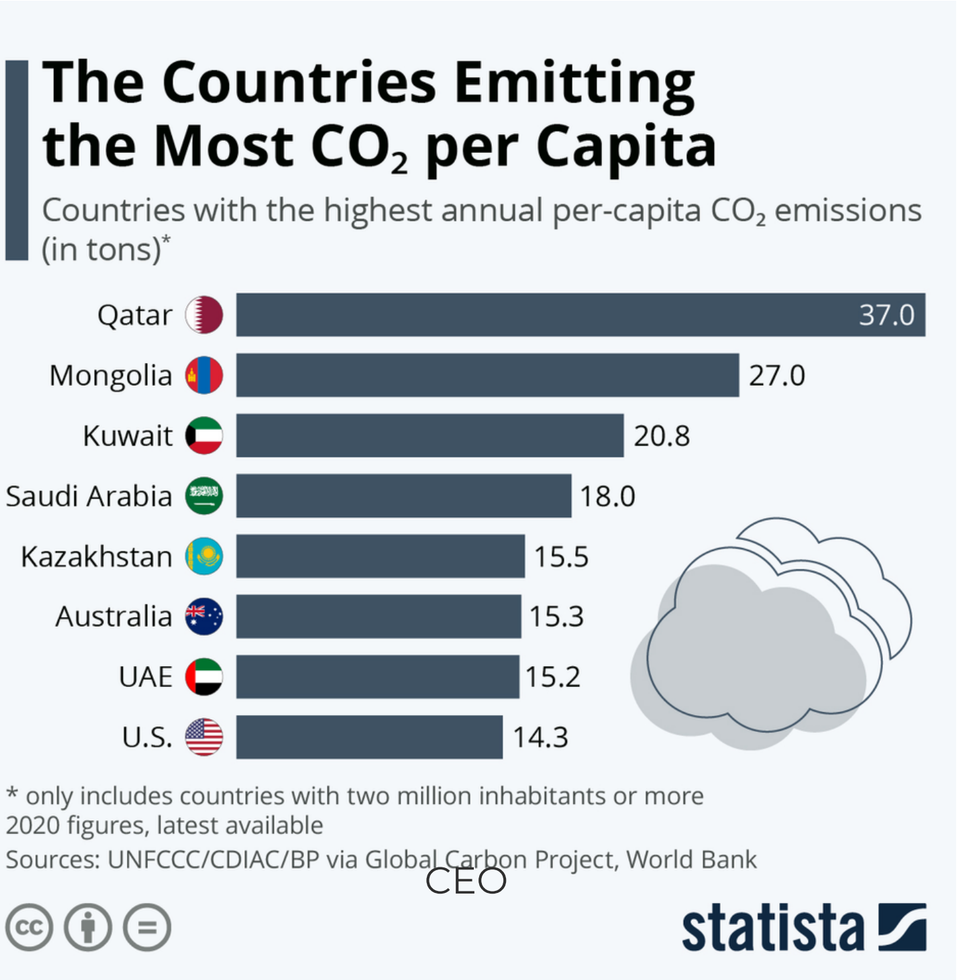
Relationship among HDI, A and PHDI



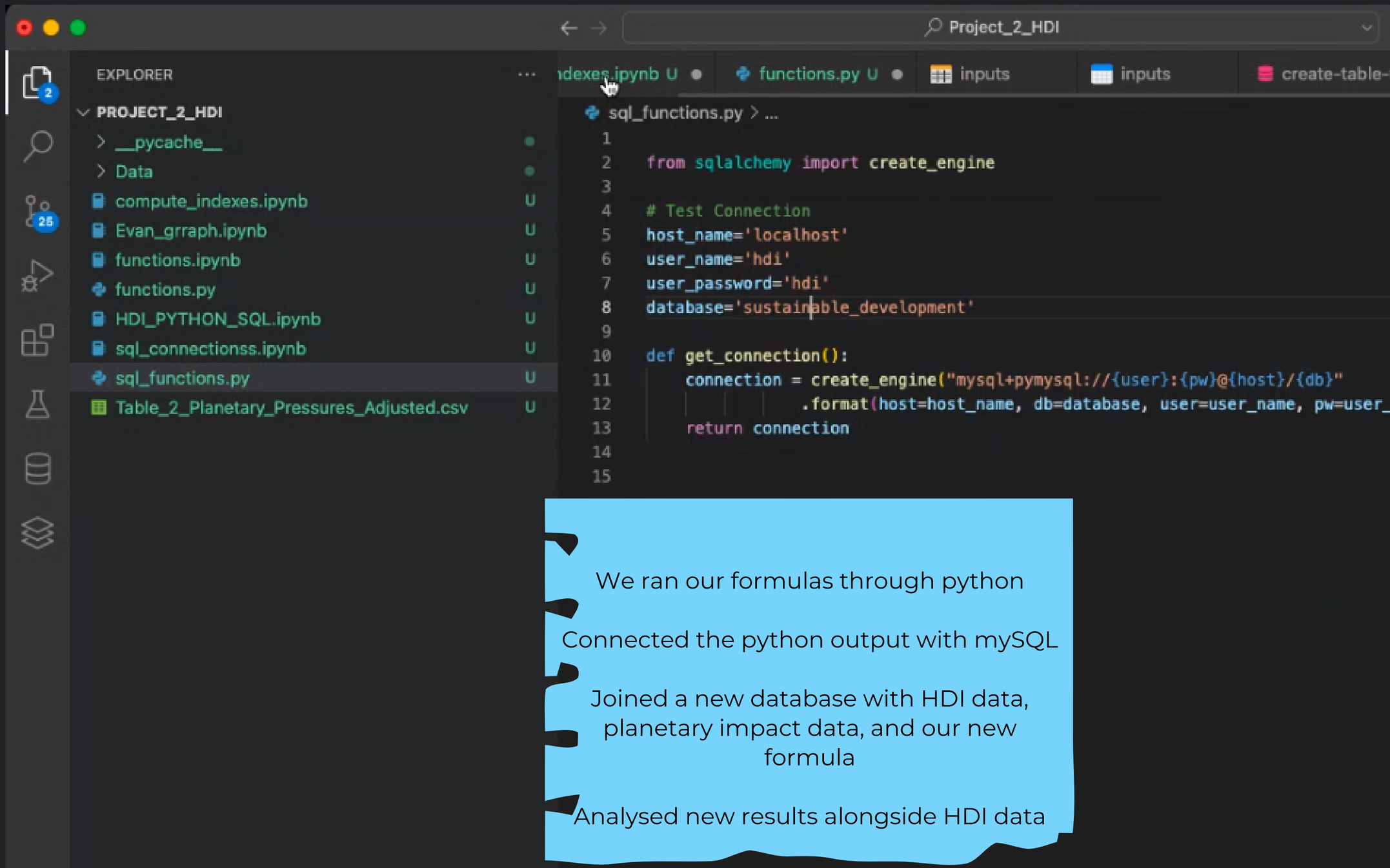
THE FLAW

Our experimental planerary-impact metrics need scaling

The CO2 and material waste levels that have been used to calculate the impacts that countries make on a planet are based upon the historical worst countries for polution. We think that countries should be evaluated by the impact of their modern-day peers.



CEO

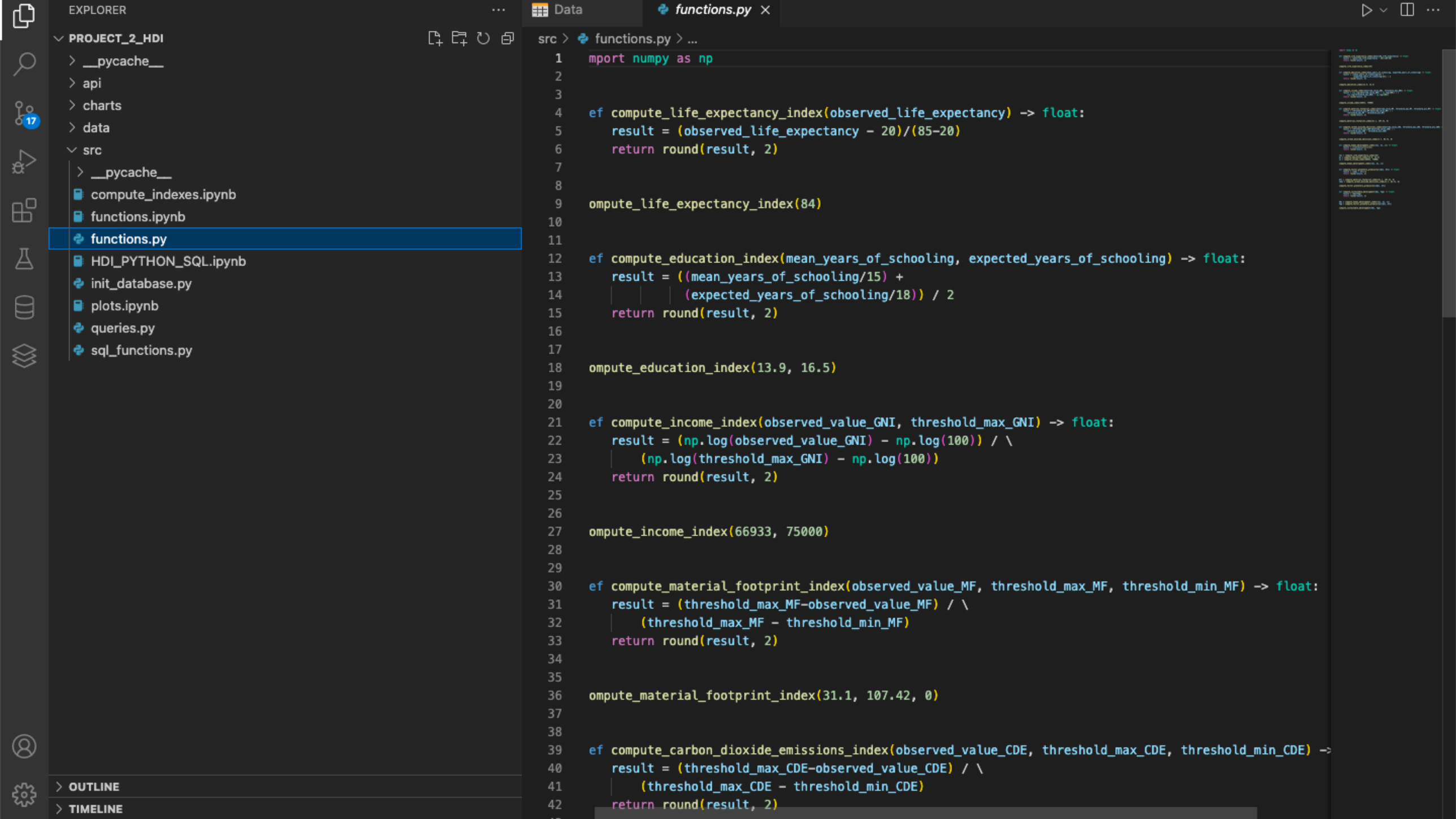


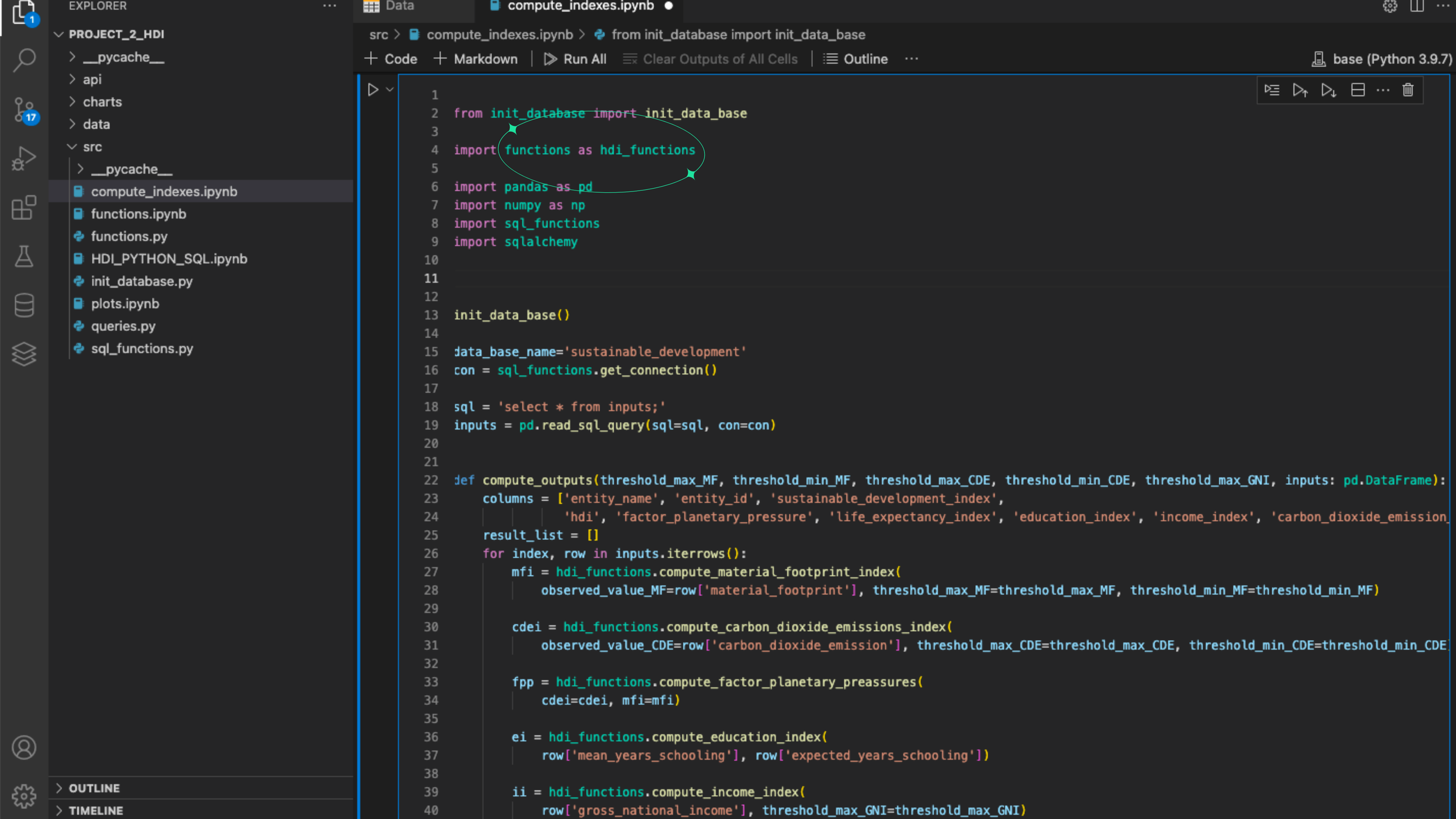
We ran our formulas through python

Connected the python output with mySQL

Joined a new database with HDI data, planetary impact data, and our new formula

Analysed new results alongside HDI data


EXPLORER
PROJECT_2_HDI
__pycache__
api
charts
data
src
__pycache__
compute_indexes.ipynb
functions.ipynb
functions.py
HDI_PYTHON_SQL.ipynb
init_database.py
plots.ipynb
queries.py
sql_functions.py
OUTLINE
TIMELINE
Data
functions.py
src > functions.py > ...
mport numpy as np
ef compute_life_expectancy_index(observed_life_expectancy) -> float:
result = (observed_life_expectancy - 20)/(85-20)
return round(result, 2)
ompute_life_expectancy_index(84)
ef compute_education_index(mean_years_of_schooling, expected_years_of_schooling) -> float:
result = ((mean_years_of_schooling/15) +
(expected_years_of_schooling/18)) / 2
return round(result, 2)
ompute_education_index(13.9, 16.5)
ef compute_income_index(observed_value_GNI, threshold_max_GNI) -> float:
result = (np.log(observed_value_GNI) - np.log(100)) / \
(np.log(threshold_max_GNI) - np.log(100))
return round(result, 2)
ompute_income_index(66933, 75000)
ef compute_material_footprint_index(observed_value_MF, threshold_max_MF, threshold_min_MF) -> float:
result = (threshold_max_MF-observed_value_MF) / \
(threshold_max_MF - threshold_min_MF)
return round(result, 2)
ompute_material_footprint_index(31.1, 107.42, 0)
ef compute_carbon_dioxide_emissions_index(observed_value_CDE, threshold_max_CDE, threshold_min_CDE) ->
result = (threshold_max_CDE-observed_value_CDE) / \
(threshold_max_CDE - threshold_min_CDE)
return round(result, 2)


EXPLORER
PROJECT_2_HDI
__pycache__
api
charts
data
src
__pycache__
compute_indexes.ipynb
functions.ipynb
functions.py
HDI_PYTHON_SQL.ipynb
init_database.py
plots.ipynb
queries.py
sql_functions.py
OUTLINE
TIMELINE
Data
compute_indexes.ipynb
src > compute_indexes.ipynb > from init_database import init_data_base
Code
Markdown
Run All
Clear Outputs of All Cells
Outline
base (Python 3.9.7)
from init_database import init_data_base
import functions as hdi_functions
import pandas as pd
import numpy as np
import sql_functions
import sqlalchemy
init_data_base()
data_base_name='sustainable_development'
con = sql_functions.get_connection()
sql = 'select * from inputs;'
inputs = pd.read_sql_query(sql=sql, con=con)
def compute_outputs(threshold_max_MF, threshold_min_MF, threshold_max_CDE, threshold_min_CDE, threshold_max_GNI, inputs: pd.DataFrame):
columns = ['entity_name', 'entity_id', 'sustainable_development_index',
'hdi', 'factor_planetary_pressure', 'life_expectancy_index', 'education_index', 'income_index', 'carbon_dioxide_emission_
result_list = []
for index, row in inputs.iterrows():
mfi = hdi_functions.compute_material_footprint_index(
observed_value_MF=row['material_footprint'], threshold_max_MF=threshold_max_MF, threshold_min_MF=threshold_min_MF)
cdei = hdi_functions.compute_carbon_dioxide_emissions_index(
observed_value_CDE=row['carbon_dioxide_emission'], threshold_max_CDE=threshold_max_CDE, threshold_min_CDE=threshold_min_CDE
fpp = hdi_functions.compute_factor_planetary_preassures(
cdei=cdei, mfi=mfi)
ei = hdi_functions.compute_education_index(
row['mean_years_schooling'], row['expected_years_schooling'])
ii = hdi_functions.compute_income_index(
row['gross_national_income'], threshold_max_GNI=threshold_max_GNI)

+ Code + Markdown | Run All | Clear Outputs of All Cells | Outline ··· base (Python 3.9.7)

```
37                row['mean_years_schooling'], row['expected_years_schooling'])
38
39            ii = hdi_functions.compute_income_index(
40                row['gross_national_income'], threshold_max_GNI=threshold_max_GNI)
41
42            lei = hdi_functions.compute_life_expectancy_index(
43                row['life_expectancy'])
44
45            hdi = hdi_functions.compute_human_development_index(
46                lei=lei, ei=ei, ii=ii)
47
48            sustainable_development = hdi_functions.compute_sustainable_development(
49                hdi=hdi, fpp=fpp)
50
51            new_row = [row['entity_name'], row['entity_id'], sustainable_development,  hdi,
52                       fpp, lei, ei, ii, cdei, mfi]
53            result_list.append(new_row)
54
55        np_arr = np.array(result_list)
56        result = pd.DataFrame(np_arr, columns=columns)
57        return result
58
59
60    onu_df = compute_outputs(107.42, 0, 68.72, 0, 75000, inputs)
61
62    onu_countries_data = onu_df[onu_df["entity_id"] == '1']
63    onu_countries_df = pd.DataFrame(onu_countries_data)
64    onu_countries_df['ranking'] = onu_countries_df['sustainable_development_in
65        ascending=False, method='max')
66    onu_countries_df.to_sql(con=con, name='outputs_countries_{tag}'.format(
67        tag='onu'), if_exists='replace')
68
69
70    onu_others_data = onu_df[onu_df["entity_id"] != '1']
71    onu_others_df = pd.DataFrame(onu_others_data)
72    onu_others_df['ranking'] = onu_others_df['sustainable_development_index'].rank(
73        ascending=False, method='max')
74
75    onu_others_df.to_sql(con=con, name='outputs_others_{tag}'.format(
76        tag='onu'), if_exists='replace')
```

threshold_max_MF: material footprint maximum threshold established by ONU 107,42

threshold_min_MF: material footprint minimun threshold established by ONU 0

threshold_max_CDE: Carbon dioxide emissions maximum threshold established by ONU 68,72

threshold_min_CDE: Carbon dioxide emissions minimun threshold established by ONU 0

```python
onu_others_df.to_sql(con=con, name='outputs_others_{tag}'.format(
    tag='onu'), if_exists='replace')


revised_df = compute_outputs(50, 0, 38, 0, 75000, inputs)
revised_countries_data = revised_df[revised_df["entity_id"] == '1']
revised_countries_df = pd.DataFrame(revised_countries_data)
revised_countries_df['ranking'] = revised_countries_df['sustainable_development_index'].rank(
    ascending=False, method='max')

outputs_data_types = {
    'entity_id': sqlalchemy.Integer,
    'sustainable_development_index': sqlalchemy.Float,
    'hdi': sqlalchemy.Float,
    'factor_planetary_pressure': sqlalchemy.Float,
    'life_expectancy_index': sqlalchemy.Float,
    'education_index': sqlalchemy.Float,
    'carbon_dioxide_emission_index': sqlalchemy.Float,
    'material_footprint_index': sqlalchemy.Float,
    'income_index': sqlalchemy.Float,
}
revised_countries_df.to_sql(con=con, name='outputs_countries_{tag}'.format(tag='revised'), if_exists='replace',
                            dtype=outputs_data_types)


revised_others_data = revised_df[revised_df["entity_id"] != '1']
revised_others_df = pd.DataFrame(revised_others_data)
revised_others_df['ranking'] = revised_others_df['sustainable_development_index'].rank(
    ascending=False, method='max')

revised_others_df.to_sql(con=con, name='outputs_others_{tag}'.format(
    tag='revised'), if_exists='replace', dtype=outputs_data_types)

```

[3]

Python

```python
onu_others_df.to_sql(con=con, name='outputs_others_{tag}'.format(
    tag='onu'), if_exists='replace')


revised_df = compute_outputs(50, 0, 38, 0, 75000, inputs)
revised_countries_data = revised_df[revised_df["entity_id"] == '1']
revised_countries_df = pd.DataFrame(revised_countries_data)
revised_countries_df['ranking'] = revised_countries_df['sustainable_development_index'].rank(
    ascending=False, method='max')

outputs_data_types = {
    'entity_id': sqlalchemy.Integer,
    'sustainable_development_index': sqlalchemy.Float,
    'hdi': sqlalchemy.Float,
    'factor_planetary_pressure': sqlalchemy.Float,
    'life_expectancy_index': sqlalchemy.Float,
    'education_index': sqlalchemy.Float,
    'carbon_dioxide_emission_index': sqlalchemy.Float,
    'material_footprint_index': sqlalchemy.Float,
    'income_index': sqlalchemy.Float,
}
revised_countries_df.to_sql(con=con, name='outputs_countries_{tag}'.format(tag='revised'), if_exists='replace',
                            dtype=outputs_data_types)


revised_others_data = revised_df[revised_df["entity_id"] != '1']
revised_others_df = pd.DataFrame(revised_others_data)
revised_others_df['ranking'] = revised_others_df['sustainable_development_index'].rank(
    ascending=False, method='max')

revised_others_df.to_sql(con=con, name='outputs_others_{tag}'.format(
    tag='revised'), if_exists='replace', dtype=outputs_data_types)

```

[3]

Python

```
ache__
te_indexes.ipynb
ons.ipynb
ons.py
YTHON_SQL.ipynb
tabase.py
pynb
s.py
nctions.py
```

```python
from sql_functions import get_connection
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

import queries as hdi_queries

import dataframe_image as dfi
```

[1] Python

```python
con = get_connection()
```

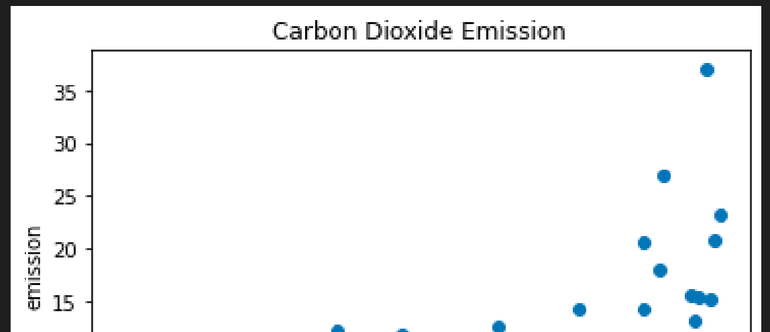
[2] Python

```python
result_set= pd.read_sql_query(sql=hdi_queries.ranking_per_cdei_query,con=con)

x_list = result_set[['revised_ranking']].values
y_list = result_set[['carbon_dioxide_emission']].values


plt.scatter(x_list, y_list)
plt.title("Carbon Dioxide Emission")
plt.xlabel("country ranking revised")
plt.ylabel("emission")
plt.show()
```

[3] Python



Project_2_HDI

EXPLORER

PROJECT_2_HDI

- __pycache__
- Data
- compute_indexes.ipynb U
- Evan_grraph.ipynb U
- functions.ipynb U
- functions.py U
- HDI_PYTHON_SQL.ipynb U
- sql_connectionss.ipynb U
- sql_functions.py U
- Table_2_Planetary_Pressures_Adjusted.csv U

compute_indexes.ipynb U | HDI_PYTHON_SQL.ipynb U | functions.py U | outputs_onu | inputs | create-tab

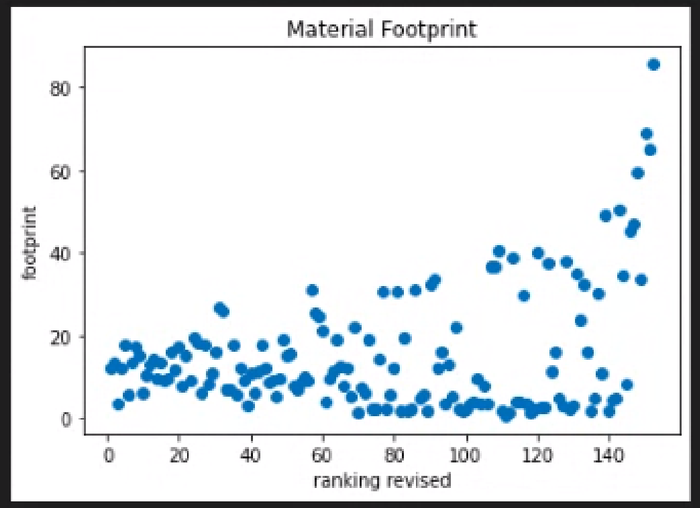
HDI_PYTHON_SQL.ipynb > # Carbon Dioxide Emission per Capital Distribution

+ Code | + Markdown | Run All | Clear Outputs of All Cells | Restart | Interrupt | Variables | Outline | base (Python 3.9.7)

ranking

```
1  sql='select hdi_rank_revised, material_footprint from hdi_revised order by hdi_rank_revised'
2  result_set= pd.read_sql_query(sql=sql,con=connection)
3
4  x_list = result_set[['hdi_rank_revised']].values
5  y_list = result_set[['material_footprint']].values
6
7
8  plt.scatter(x_list, y_list)
9  plt.title("Material Footprint")
10 plt.xlabel("ranking revised")
11 plt.ylabel("footprint")
12 plt.show()
```

[ ] Python



		Human Development Index (HDI)
HDI rank	**Country**	Value
		2021
	Very high human development	
1	Switzerland	0.962
2	Norway	0.961
3	Iceland	0.959
4	Hong Kong, China (SAR)	0.952
5	Australia	0.951
6	Denmark	0.948
7	Sweden	0.947
8	Ireland	0.945
9	Germany	0.942
10	Netherlands	0.941
11	Finland	0.940
12	Singapore	0.939
13	Belgium	0.937
13	New Zealand	0.937
15	Canada	0.936

OUR FINDINGS

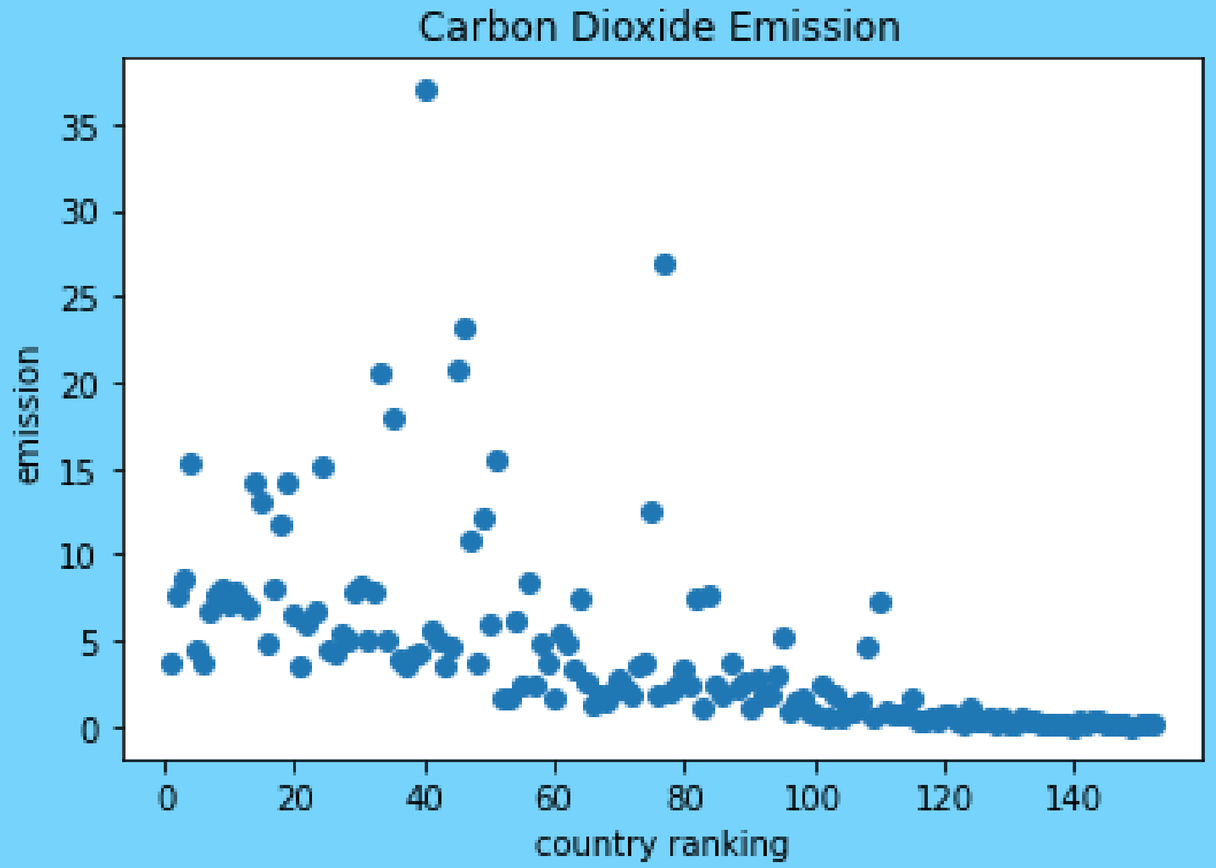
1. Top and bottom tier countries were more likely to have higher GDI. This is more equitable.

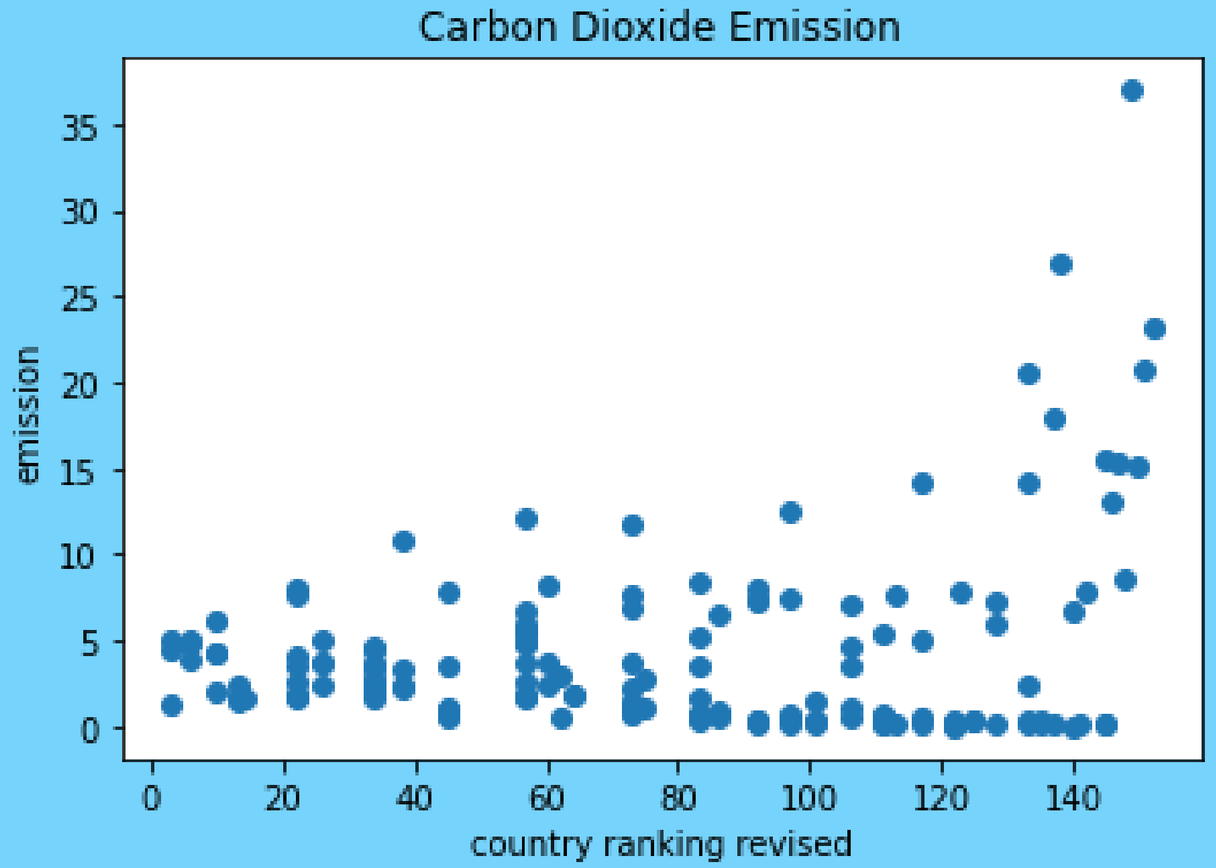
Brunei Darussalam	Brunei Darussalam
Chad	Kuwait
South Sudan	United Arab Emirates
Niger	Qatar
Central African Republic	Iceland
Burundi	Australia
Mali	Luxembourg
Burkina Faso	Chad
Mozambique	Kazakhstan
Guinea	Singapore
Yemen	South Sudan

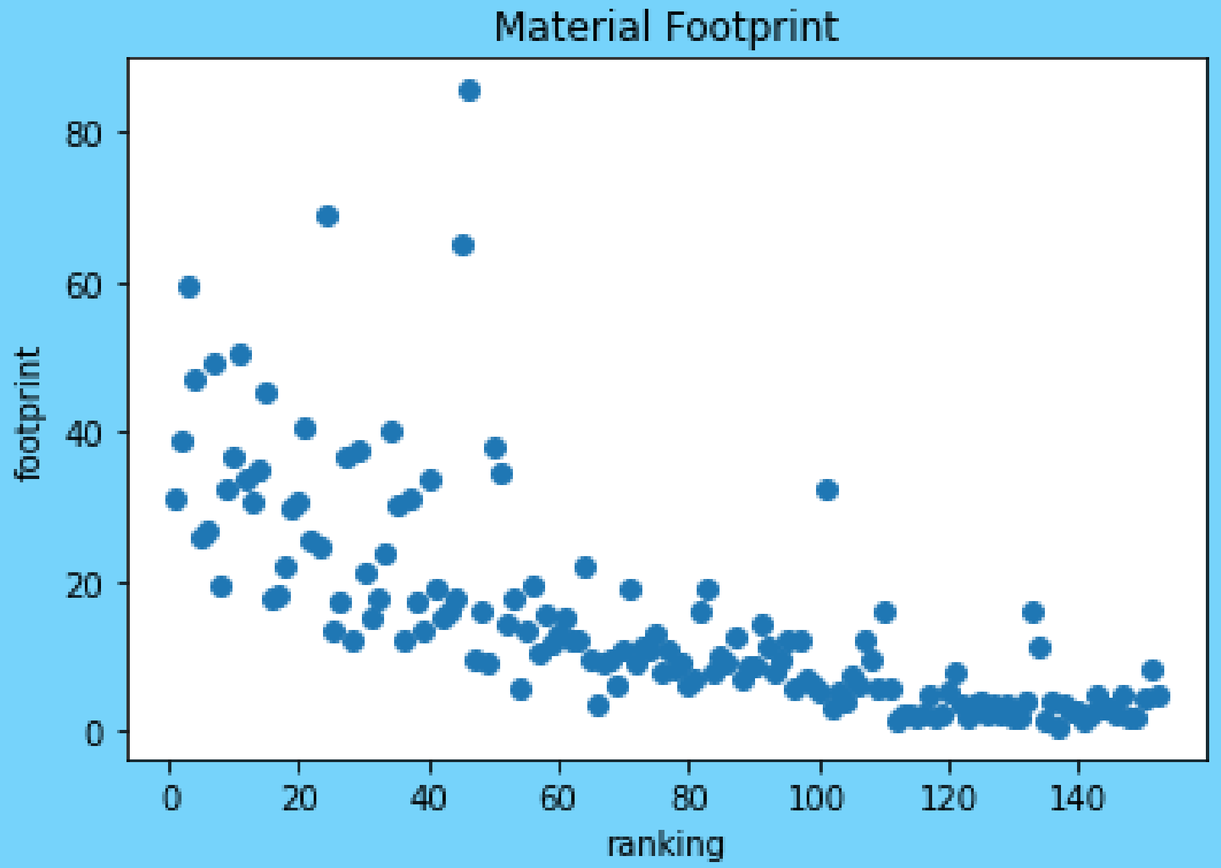
2. A number of countries that were high on the HDI list were remarkably low when planetary impact was reevaluated.

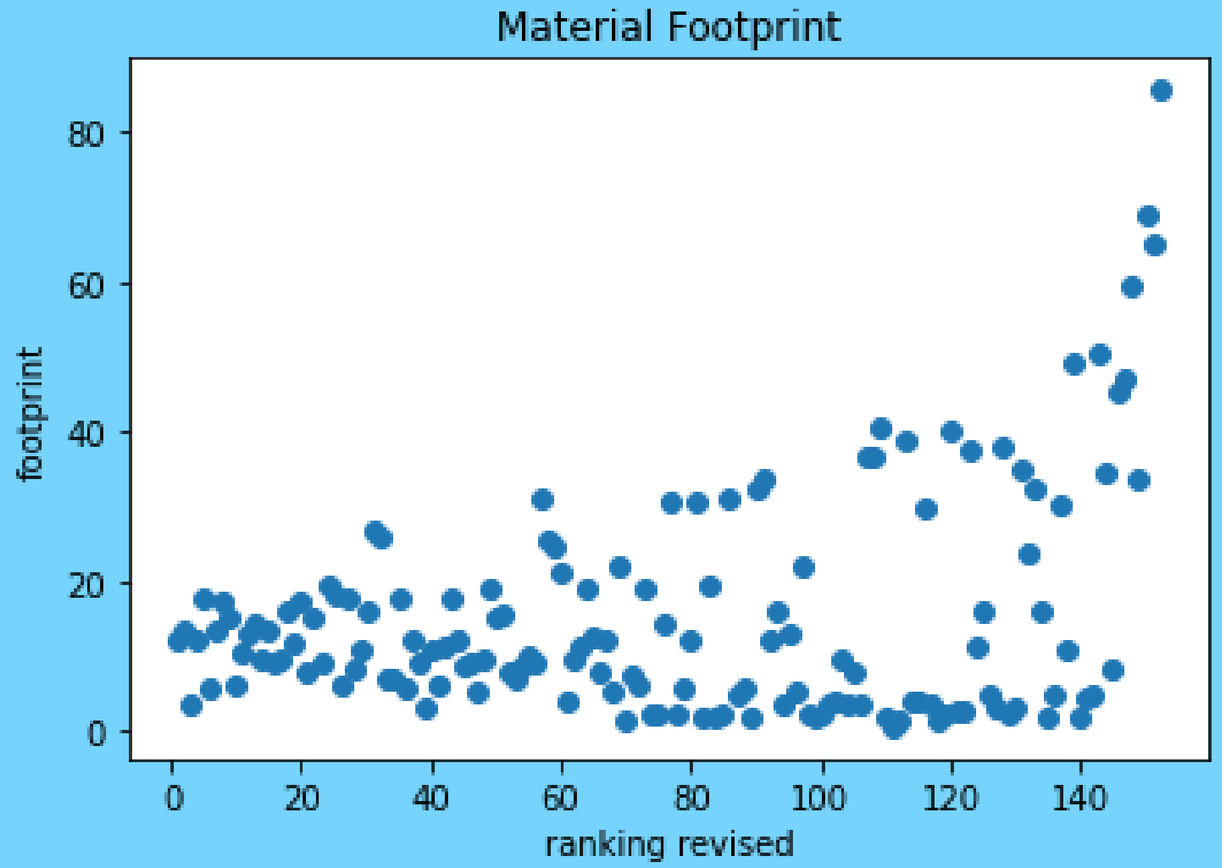
3. Our formula shows a lower standard deviation between developed and developing counries.

	entity_name	onu_sustainable_development_index	revised_sustainable_development_index
0	United Kingdom	0.82	0.71
1	Greece	0.82	0.71
2	Spain	0.82	0.73
3	Italy	0.82	0.73
4	Sweden	0.81	0.66
5	Switzerland	0.8	0.61
6	Denmark	0.8	0.65
7	Germany	0.8	0.67
8	Portugal	0.8	0.71
9	France	0.8	0.70
10	Japan	0.79	0.67
11	Chile	0.78	0.70
12	New Zealand	0.77	0.58
13	Croatia	0.77	0.67
14	Argentina	0.76	0.67
15	Hungary	0.76	0.66
16	Poland	0.76	0.63
17	Slovenia	0.76	0.62
18	Austria	0.76	0.62
19	Moldova (Republic of)	0.75	0.73
20	Belarus	0.75	0.70


Carbon Dioxide Emission
emission
country ranking
0
5
10
15
20
25
30
35
0
20
40
60
80
100
120
140


Carbon Dioxide Emission
emission
country ranking revised
0
5
10
15
20
25
30
35
0
20
40
60
80
100
120
140


Material Footprint
footprint
ranking
0
20
40
60
80
0
20
40
60
80
100
120
140


Material Footprint
footprint
ranking revised
0
20
40
60
80
0
20
40
60
80
100
120
140