African Food

Project Documentation

Studente

Cosentino Andrea
Dell'Agostino Luca
Somaruga Tomaso

Professore

Profeta Giovanni

Corso di laurea

Data Science and Artificial Intelligence

Corso

Data Visualization

Introduction

The research question of our project consists of the analysis of some food parameters in African countries to discover a part of the cost of life that people who live there have to face off.
The project is for informational purposes therefore the reference target is very vast even if there is a will to reach those people that may venture into deeper analyses on this topic or those people that could have the money and will to really make a change.
The final goal of the project is to show an overview of the food's price in the various countries to highlighting the poorest nations.
We did some researchers and according to an article in an Italian newspaper, there is no global shortage of food, it is just not distributed fairly (Repubblica, 2022). With our visualization you can identify the countries where there is a lack or high price of food and afterwards, this can be the starting point for new researches to analyze the reasons.

Data Sources

We got the database from Kaggle.com and we started working on it in order to delete, change or add some data. All these steps will be explained in the data pre-processing.
We did some researches about African food prices and we discovered important information for our project on important websites like the above cited Repubblica.

Interface Design

Regarding the interface we decided to use a simply but effective web page that is structured in three main steps. At the first we put an introduction of the problem and, by going ahead, you can find four visualizations that show, at first, an overview of the problem and later, the last visualization goes deeper to display a classification of the poorest countries and the richest ones.
In the main page we put a link that takes you to our visualization protocol web page in which provides us fundamental information about the origin and characteristics of our data. In the list on this page, you can find the links to the original dataset and the metadata file which explains you the meaning of all used columns in the dataset so you can understand which ones we picked for our Data visualizations.
Other important information can be found in the last step's link that takes you to another web page that shows all the steps we performed to pass from raw data to the final dataset we used for our visualizations. You can find a flowchart that explains you all the steps we have done to modify the original dataset to get the one from which we created our views. We made many changes because the original dataset included incorrect or incomplete data.

Data Pre-processing

The first step in this data visualisation process was to import the necessary libraries, including pandas and plotly.express. Next, the data was read in from a CSV file called 'global_food_prices.csv' and stored in a dataframe called 'df'.

	adm0_id	adm0_name	adm1_id	adm1_name	mkt_id	mkt_name	cm_id	cm_name	cur_name	pt_id	pt_name	um_id	um_name	mp_month	mp_year	mp_price	mp_commoditysource
0	1.0	Afghanistan	272	Badakhshan	266	Fayzabad	55	Bread - Retail	AFN	15	Retail	5	KG	1	2014	50.0000	NaN
1	1.0	Afghanistan	272	Badakhshan	266	Fayzabad	55	Bread - Retail	AFN	15	Retail	5	KG	2	2014	50.0000	NaN
2	1.0	Afghanistan	272	Badakhshan	266	Fayzabad	55	Bread - Retail	AFN	15	Retail	5	KG	3	2014	50.0000	NaN
3	1.0	Afghanistan	272	Badakhshan	266	Fayzabad	55	Bread - Retail	AFN	15	Retail	5	KG	4	2014	50.0000	NaN
4	1.0	Afghanistan	272	Badakhshan	266	Fayzabad	55	Bread - Retail	AFN	15	Retail	5	KG	5	2014	50.0000	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2050633	271.0	Zimbabwe	3444	Midlands	5594	Mbilashaba	432	Beans (sugar) - Retail	ZWL	15	Retail	5	KG	6	2021	233.3333	NaN
2050634	271.0	Zimbabwe	3444	Midlands	5594	Mbilashaba	539	Toothpaste - Retail	ZWL	15	Retail	116	100 ML	6	2021	112.5000	NaN
2050635	271.0	Zimbabwe	3444	Midlands	5594	Mbilashaba	540	Laundry soap - Retail	ZWL	15	Retail	5	KG	6	2021	114.0000	NaN
2050636	271.0	Zimbabwe	3444	Midlands	5594	Mbilashaba	541	Handwash soap - Retail	ZWL	15	Retail	66	250 G	6	2021	59.5000	NaN
2050637	271.0	Zimbabwe	3444	Midlands	5594	Mbilashaba	887	Fish (kapenta) - Retail	ZWL	15	Retail	5	KG	6	2021	1200.0000	NaN

2050638 rows × 17 columns

The next step was to drop the 'cur_id' and 'mp_commoditysource' columns from the dataframe. The length of the 'adm1_name' column was also checked for null values. The unique values in the 'adm0_name' column were also identified. The dataframe was then filtered to only include relevant columns and the average price for each country in the 'adm0_name' column was calculated and stored in a new dataframe called 'df_price'.

Next, separate dataframes were created for each African country in the dataset by filtering the original dataframe 'df_new' based on the country name in the 'adm0_name' column. These separate dataframes were named with the country's abbreviation, such as 'df_alg' for Algeria and 'df_ang' for Angola.

To conduct an analysis specifically focused on African countries, only data belonging to those countries was selected and included in a new DataFrame df_africa.

After that, the prices in "df_africa" are converted from various currencies to US dollars. The first step is to create a list of unique currencies in the DataFrame by using the ".unique()" method on the "cur_name" column. The data types of the DataFrame are also checked using the ".dtypes" attribute.

A dictionary is then created with keys as the unique currencies and values as "pd.NA" (a placeholder for missing data). The exchange rates for each currency are then manually assigned to their respective keys in the dictionary.

A new column named "mp_price" is then created in the DataFrame by applying a lambda function that converts the price in each row to US dollars using the exchange rate from the dictionary. The original "cur_name" column is then dropped and the modified DataFrame is exported to a CSV file.

```
currencies = df_africa.cur_name.unique().tolist()
currencies = dict.fromkeys(currencies, pd.NA)
currencies['DZD'] = 0.00712
currencies['AOA'] = 0.00206
currencies['XOF'] = 0.00152
currencies['BIF'] = 0.00048
currencies['XAF'] = 0.00152
currencies['CVE'] = 0.00906
currencies['CDF'] = 0.00048
currencies['DJF'] = 0.00562
currencies['EGP'] = 0.04107
currencies['ERN'] = 0.06666
currencies['ETB'] = 0.01874
currencies['GMD'] = 0.01638
currencies['GHS'] = 0.07019
currencies['GNF'] = 0.00011
currencies['KES'] = 0.00821
currencies['LSL'] = 0.05607
currencies['LRD'] = 0.00650
currencies['LYD'] = 0.19999
currencies['MGA'] = 0.00023
currencies['MWK'] = 0.00098
currencies['MRO'] = 0.00264
currencies['MZN'] = 0.01565
currencies['NAD'] = 0.05608
currencies['NGN'] = 0.00227
currencies['RWF'] = 0.00094
currencies['SLL'] = 0.00006
currencies['SOS'] = 0.0000176
currencies['ZAR'] = 0.05609
currencies['SSP'] = 0.00767
currencies['SDG'] = 0.00176
currencies['UGX'] = 0.00026
currencies['TZS'] = 0.00043
currencies['ZMW'] = 0.06128
currencies['USD'] = 1
currencies['ZWL'] = 0.00310
```

The new file is then reimported for other changes in another notebook in python.
The notebook reads from two csv files, 'coordinates.csv' and 'df_africa.csv', and performs several operations on them.

coordinates.csv

	place	lat	lon
0	Alger	36.775361	3.060188
1	Tindouf	27.671840	-8.139730
2	Luanda	-8.827270	13.243951
3	Lunda Norte	-7.381086	20.833799
4	Alibori	11.131103	2.932232
...	...	...	...
345	Mashonaland West	-17.361531	30.192935
346	Masvingo	-20.072014	30.834194
347	Matabeleland North	-18.934672	27.772832
348	Matabeleland South	-20.941429	29.003685
349	Midlands	-19.461632	29.820595

350 rows × 3 columns

df_africa.csv

	Unnamed: 0	adm0_name	adm1_name	mkt_name	cm_name	um_name	mp_month	mp_year	mp_price
0	0	Algeria	Alger	Algiers	Rice - Retail	KG	4	2015	0.640800
1	1	Algeria	Alger	Algiers	Rice - Retail	KG	5	2015	0.683520
2	2	Algeria	Alger	Algiers	Rice - Retail	KG	6	2015	0.683520
3	3	Algeria	Alger	Algiers	Rice - Retail	KG	7	2015	0.590960
4	4	Algeria	Alger	Algiers	Rice - Retail	KG	8	2015	0.569600
...	...	...	...	...	...	...	...	...	...
1088957	1088957	Zimbabwe	Midlands	Mbilashaba	Beans (sugar) - Retail	KG	6	2021	0.723333
1088958	1088958	Zimbabwe	Midlands	Mbilashaba	Toothpaste - Retail	100 ML	6	2021	0.348750
1088959	1088959	Zimbabwe	Midlands	Mbilashaba	Laundry soap - Retail	KG	6	2021	0.353400
1088960	1088960	Zimbabwe	Midlands	Mbilashaba	Handwash soap - Retail	250 G	6	2021	0.184450
1088961	1088961	Zimbabwe	Midlands	Mbilashaba	Fish (kapenta) - Retail	KG	6	2021	3.720000

1088962 rows × 9 columns

First, it reads in the 'coordinates.csv' file and drops certain columns, then it reads in 'df_africa.csv' file. It then removes certain substrings from the 'cm_name' column that were not needed.

It then creates a new column 'name' which is a combination of the 'adm1_name' and 'adm0_name' columns. It concatenates the 'coord' dataframe with a new dataframe which contains the data of cities not present in the 'coord' dataframe and that were added manually.

It then performs a left merge of the original dataframe and the coordinates dataframe on 'adm1_name' and 'Name' columns. It then drops certain columns, fills null values and checks how many unique names were contained in the 'cm_name' column.

```
dff = pd.merge(
    left=df_africa,
    right=coordinates,
    left_on='adm1_name',
    right_on='Name',
    how='left'
)
```

```
dff
```

	adm0_name	adm1_name	mkt_name	cm_name	um_name	mp_month	mp_year	mp_price	Latitude	Longitude
0	Algeria	Alger	Algiers	Rice	KG	4	2015	0.640800	36.775361	3.060188
1	Algeria	Alger	Algiers	Rice	KG	5	2015	0.683520	36.775361	3.060188
2	Algeria	Alger	Algiers	Rice	KG	6	2015	0.683520	36.775361	3.060188
3	Algeria	Alger	Algiers	Rice	KG	7	2015	0.590960	36.775361	3.060188
4	Algeria	Alger	Algiers	Rice	KG	8	2015	0.569600	36.775361	3.060188
...	...	...	...	...	...	...	...	...	...	...
1205567	Zimbabwe	Midlands	Mbilashaba	Beans (sugar)	KG	6	2021	0.723333	-19.461632	29.820595
1205568	Zimbabwe	Midlands	Mbilashaba	Toothpaste	100 ML	6	2021	0.348750	-19.461632	29.820595
1205569	Zimbabwe	Midlands	Mbilashaba	Laundry soap	KG	6	2021	0.353400	-19.461632	29.820595
1205570	Zimbabwe	Midlands	Mbilashaba	Handwash soap	250 G	6	2021	0.184450	-19.461632	29.820595
1205571	Zimbabwe	Midlands	Mbilashaba	Fish (kapenta)	KG	6	2021	3.720000	-19.461632	29.820595

1205572 rows × 10 columns

It then reads in a new csv file 'coordinates_updated.csv' and performs a left merge with the filtered dataframe 'g' containing only the rows with null latitude and longitude.

The final output is the merged dataframe containing the filtered data of missing latitude values and the updated coordinates of the missing values.

A new dataframe, the final one, is created by concatenating all the data that we processed before.

It then changes the unit of measures to convert them to KG and the mp_price column changes accordingly, this is done to convert the prices to a common unit of measurement.
This is achieved by creating a dictionary 'unit' containing keys as unique values of the 'um_name' (unit of measure) column and values as NaN. The dictionary is then updated with keys as specific unit of measurements and values as a factor to convert them to the base unit.

```
final
✓ 0.1s
```

	adm0_name	adm1_name	mkt_name	um_name	mp_month	mp_year	mp_price	Latitude	Longitude	cm_type
0	Algeria	Alger	Algiers	KG	4	2015	0.64080	36.775361	3.060188	Rice
1	Algeria	Alger	Algiers	KG	5	2015	0.68352	36.775361	3.060188	Rice
2	Algeria	Alger	Algiers	KG	6	2015	0.68352	36.775361	3.060188	Rice
3	Algeria	Alger	Algiers	KG	7	2015	0.59096	36.775361	3.060188	Rice
4	Algeria	Alger	Algiers	KG	8	2015	0.56960	36.775361	3.060188	Rice
...	...	...	...	...	...	...	...	...	...	...
128347	Sudan	Melut	Melut	KG	1	2021	1.91750	10.441975	32.202474	Milk
128348	Sudan	Melut	Melut	KG	2	2021	3.83500	10.441975	32.202474	Milk
128349	Sudan	Melut	Melut	KG	5	2021	4.44860	10.441975	32.202474	Milk
128350	Sudan	Melut	Melut	KG	6	2021	4.60200	10.441975	32.202474	Milk
128351	Sudan	Melut	Melut	KG	8	2021	3.06800	10.441975	32.202474	Milk

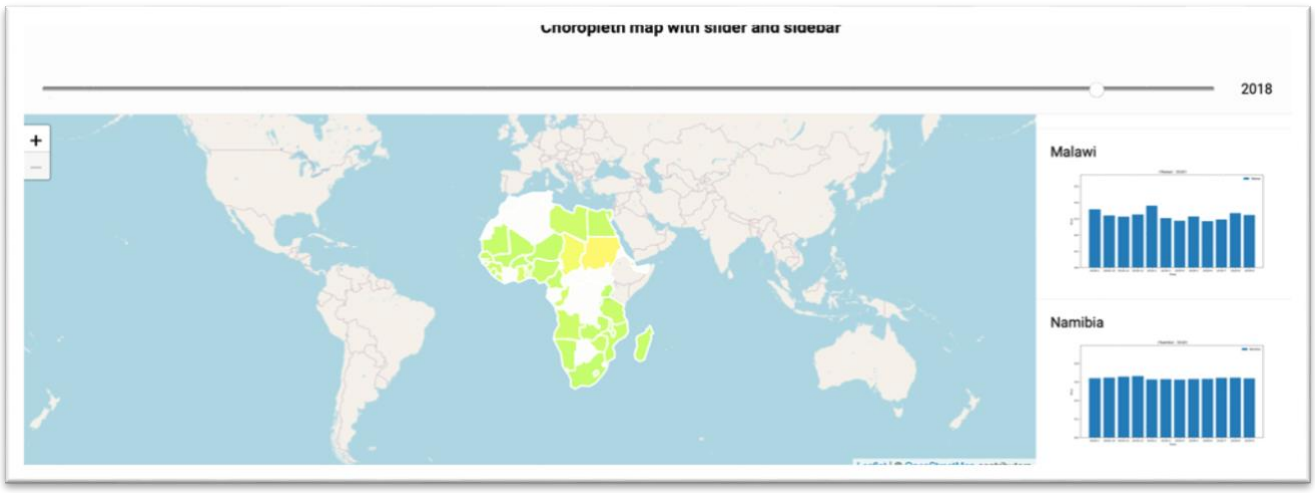
1205572 rows × 11 columns

It then applies this dictionary to the 'mp_price' column of the final DataFrame using the apply() function and the lambda function to multiply the 'mp_price' values by the corresponding conversion factor in the dictionary. The final output is the modified dataframe with prices in the common unit of measurement.

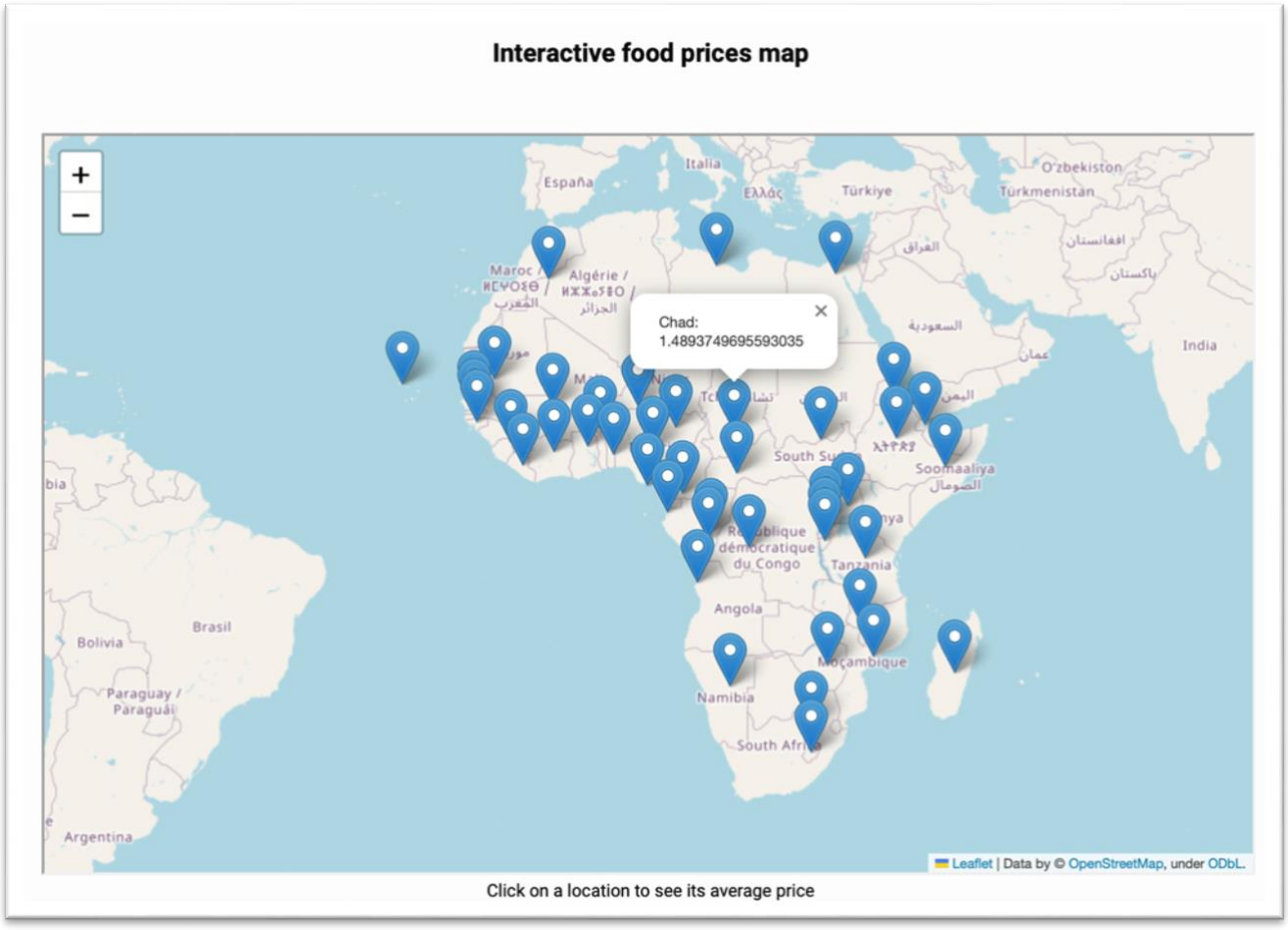
Last but not least we used the new cleaned data to obtain data visualization, the line plots and bar plots were made in python and then exported to the html file.

Data visualizations
(design, development, findings)

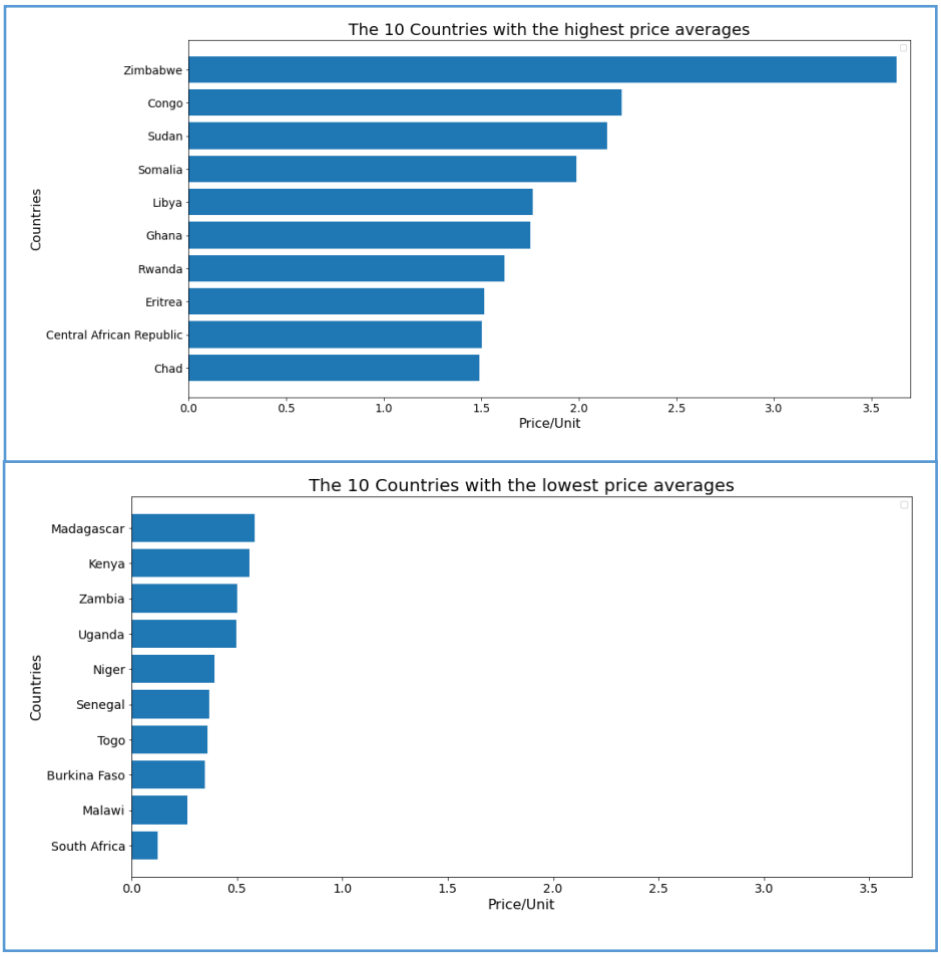
From the first interactive visualization we can visualize the price change over the years. With the sidebar above the graph, we can change year and see the data for this specific period.



With the second map we can see the average price over the years of a specific place.



In the last part of the report, we want to highlight the 10 best and worst places in Africa based on the overall average of the prices.
In the first graph we can see the countries where market prices are the highest. In these countries the cost of food is really high for the people who live there, this is because the level of poverty is certainly very high and furthermore, as there is a shortage of food, the products will be even more expensive, it could also mean an export/ import of valuable goods.
On the other hand, in the second graph, we can see the countries with the lowest price averages. We interpret this as a result of a lack of goods in the shown countries, another possibility is that they can afford to sell at lower prices since they have a higher welfare level.



From these graphs we can understand the places that need more help and based on this show the result to the people who can do something to improve this situation.
Obviously to do a complete research analysis we would need a greater amount of sources, possibly regarding the ratio between cost and revenue of the single countries.
We should also consider that these countries have a local currency that may differ greatly from USD in the same way as the old lire in Italy granted a well-suited life, yet compared to the Euros, its ratio is much smaller.
This might be a problem because more developed nations can invest and export at much lower costs than the rest of the world. This results in an excessively high purchasing power, which uses up a lot of resources for a lower-than-normal return on investment.

Next Step

One possible next step can be to seek further sources and verify the veracity of the data used and therefore of the results obtained, for example we could compare the obtained results with other datasets regarding GDP or wealth based on other commodities.
Further research could lead to the discovery of the reasons for the inefficient distribution of food in Africa and therefore to the formulation of possible solutions to the problem.