SDG 2 - Zero Hunger

Project Title

Food Deficiency & Agricultural Productivity

ES1101: Computational Data Analysis

Submitted To

Dr. Amit Sinhal

Dr. Umesh Gupta

Dr. Utsav Upadhyay

Submitted By – Team 15B

Swastik Kulshreshtha (2022BTech105)

Sheetal Kanwar (2022BTech095)

Riya Gupta (2022BTech086)

Puneet (2022BTech077)

Institute of Engineering and Technology (IET)

JK Lakshmipat University, Jaipur

Table of Contents

List of Figures

List of Tables

Abstract

In this Report, our main goal is to investigate the current state of stunting and malnutrition in children across the world. Also, we try to analyze the prevalence of food deficiency diseases in females and children with the help of the available data. In addition, we also examined the rate of agriculture production and improvement in hunger index throughout the world. We also learnt to apply various Statistical methods from both Descriptive and Inferential Statistics throughout the making of this Report. As the outcome, we concluded that the state of stunting and malnutrition (wasting/overweightness) among children is improving but at a very slow pace. As for the prevalence of anaemia in women, the growth rate is also significantly stale. Along with this, the growth in the Food Consumption per Capita in countries with high no. of cases of food deficiency diseases is not really dependent on the disease cases. For the analysis on agricultural economy, we accessed the data of agricultural sales and animal feed expenditure, both of which have a very different situation in many countries. Last but not the least, the growth rate in Hunger Score across the countries is improving quiet well but not as quickly as the GDP of a country.

Introduction

Figure 1: Sustainable Development Goals

The 2030 Agenda for Sustainable Development, which was adopted by all UN Member States in 2015, is a shared plan for peace and prosperity for people and the planet. At its core are the 17 Sustainable Development Goals (SDGs), which urge for a call to action by all governments in a global partnership. The UN understands that eliminating poverty and other deprivations must be linked with programmes to improve health and education, reduce inequality, and encourage economic growth – along with tackling the climate change and safeguarding the oceans and forests.

The 17 Sustainable Development Goals are-

- **No Poverty**
- **Zero Hunger**
- **Good Health and Well-being**
- **Quality Education**
- **Gender Equality**
- **Clean Water and Sanitation**
- **Affordable and Clean Energy**
- **Decent Work and Economic Growth**
- **Industry, Innovation and Infrastructure**
- **Reduced Inequalities**
- **Sustainable Cities and Communities**
- **Responsible Consumption and Production**
- **Climate Action**
- **Life below Water**
- **Life on Land**
- **Peace, Justice and Strong Institutions**
- **Partnerships for the Goals**

SDG 2 : Zero Hunger

Among the 17 Goals, SDG 2 seeks to eliminate hunger and malnutrition by 2030. It also promises to ensure everyone an access to healthy and adequate food throughout the year. It will necessitate sustainable food production systems and resilient agricultural practises, like fair access to land, technology, and markets, as well as infrastructural and technological investments, all over the world, for enhancing agricultural yield.

According to the following infographic from the Sustainable Development Goals Report of 2022 by the United Nations, the results of recent global events such as COVID-19 and the Ukraine-Russia War have resulted in Food Insecurity in many regions.

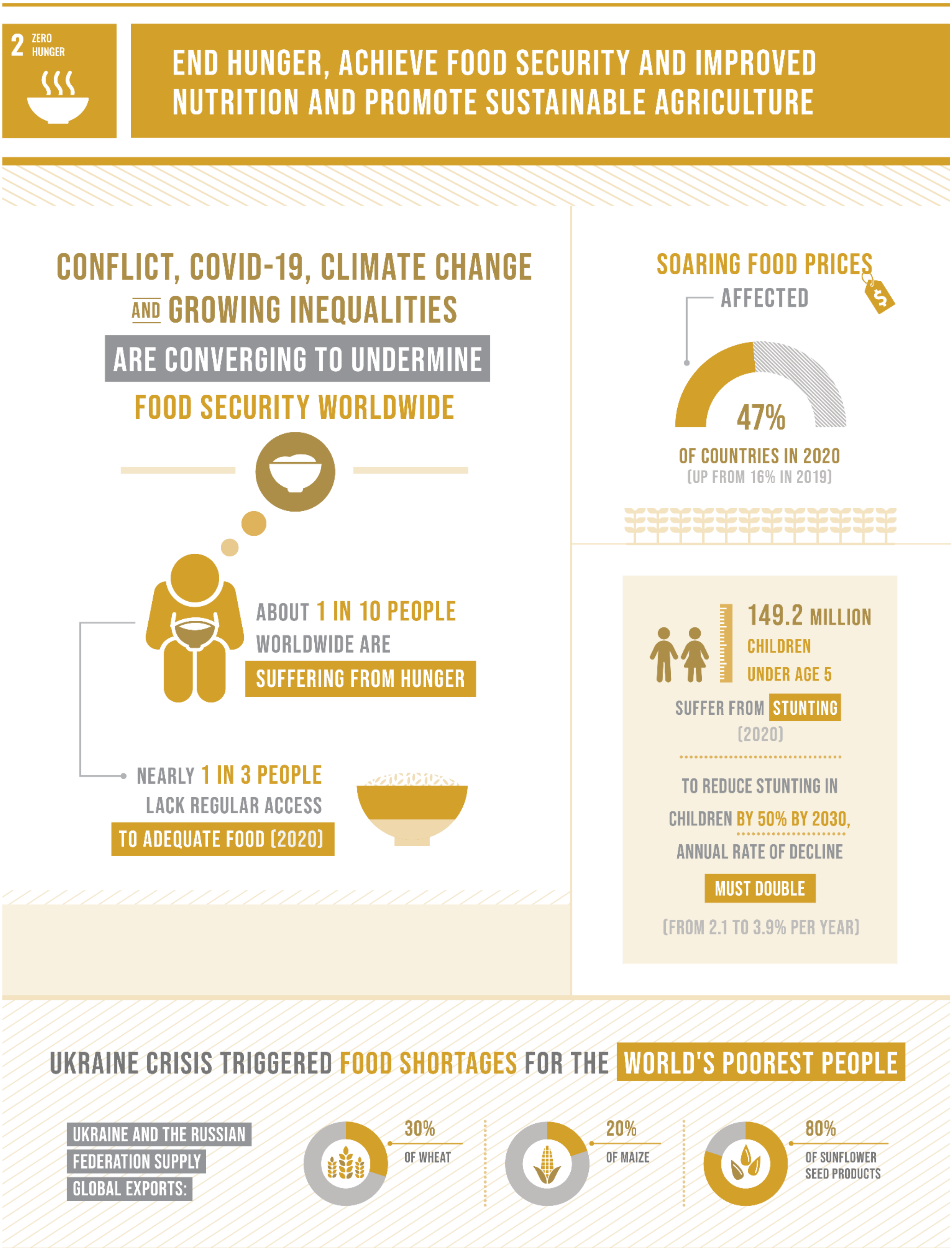


Figure 2: SDG 2 – Zero Hunger

Literature Survey

Tyson Brown from National Geographic Society[1] states that Hunger is not caused by food shortage alone, but also due to the combination of natural, social, and political forces. Many natural resources that are essential for human life, like water, forests, soil, and others are currently under decline. Climate change is going to speed up the deterioration of valuable resources because of the increased frequency of extreme weather events like droughts and their impact on harvests. Hunger is also driven by poverty and inequality, which ultimately affects who has access to food and also, the type and quantity of food that are available to them.

As stated by Farrell D. in his publication "*Poultry Development Review*"[2], The goal of several sanitary and phytosanitary (SPS) regulations and technological trade barriers (TBT) is to guarantee consumer food safety, for instance by establishing criteria for labelling and product quality. For the purpose of preserving the life of plants and animals by preventing the importation of pests and diseases, other SPS measures including inspections, quarantines, or temporary import bans. When it comes to the consumption and accessibility of wholesome food, these actions can have a direct influence on food security.

The evaluation of the literature and gaps analysis by FAO, Rome[3] was to offer operational definitions and measurement techniques for agricultural productivity across its various aspects, including farm-level or aggregated-level, physical-based or value-based, partial-input or multi-input. It has also aimed to clarify how productivity may be divided into primary influences, technical efficiency, and advancements, as well as how the former can be calculated.

The structure of farming, which in many developing countries is dominated by very small and frequently subsistence farms. This itself explains why different measurement objectives are appropriate. An example of this can be the focus on indicators like output quantities per labour unit labour productivity, data-demanding instead of highly aggregated value-weighted and data-demanding total factor productivity indices.

Mogues T and Others in their FAO report of 2012[4] say that, in the countries for which data are available, the average annual revenue of large-scale food producers is more than double that of small-scale food producers. Also, the income per production unit, led by females is consistently lower than the income of those led by males among small-scale food producers.

According to the most recent data from 44 nations, the incomes of small-scale food producers continue to fall short of those of larger-scale producers. The average annual income from agriculture for small-scale food producers is less than USD 2000 in the majority of nations, and less than USD 4500 globally. Additionally, small-scale food producers make less money on average than large-scale food producers in three-quarters of the nations for which data are available.

Derek Headey and Others in their paper *"Impacts of COVID-19 on childhood malnutrition and nutrition-related mortality"*[5] quote "The unprecedented global social and economic crisis triggered by the COVID-19 pandemic poses grave risks to the nutritional status and survival of young children in low-income and middle-income countries (LMICs)". As stated in the paper, sharp declines in household incomes, changes to the availability and cost of nutrient-dense foods, and disruptions to health, and social protection services are all expected to lead to an increase in malnutrition, particularly wasting, among children. One in ten infant deaths in LMICs can be related to severe wasting because severely wasted children are more prone to die from infectious diseases. Before the pandemic of COVID-19, an estimated number of 47 million children were wasted, with most of these infants living in Africa and South Asia.

The paper by FAO "*The State of Food Insecurity in the World. Meeting the 2015 International Hunger Targets: Taking Stock of Uneven Progress*"[6] quotes that "the number of undernourished persons has risen sharply over the past two years, with up to 828 million people in the world facing hunger in 2021."

The Food and Agriculture Organisation of the United States estimates that the prevalence of undernourishment in the world climbed from around 8 % in 2019 to around 9.3 % in 2020, and then further to about 9.8 % in 2021, after being almost stable for five years.

In their NLM report, DL Romana[7] says, "The global response to the growing threat of climate change needs to be accelerated to adequately preserve crop and crop-associated diversity." In 2021, there were 1.1 % more accumulations of plant genetic resources for agriculture that were conserved offsite either under medium-term or under long-term conditions. This is equivalent to nearly one-thirds of the average annual growth rate of germplasm accumulations during the previous three decades. Gene bank activities, including the gathering and procurement of new germplasm, gradually returned to normal after the first year of the COVID-19 pandemic, and the trend of a steady rise in the number of global germplasm holdings began after the slowdown recorded in 2020. Landrace and farmers' varieties, research materials, and wild samples made up the majority of the recently added materials to the offsite collections.

A report from WHO[8] said that Anaemia is a highly prevalent disease, which affects children, and women of reproductive age, globally. Insufficient iron during pregnancy means insufficient iron reserves for the developing foetus and other poor reproductive outcomes including pre-term delivery and low birthweight are all results of anaemia. Although anaemia is thought to be most frequently caused by iron deficiency, there are other nutritional and non-nutritional reasons as well. Anaemia is diagnosed by measuring the blood haemoglobin content, which is influenced by a variety of variables including age, sex, smoking, altitude, and pregnancy trimester. Blood haemoglobin concentrations can be used to determine the degree of iron shortage when combined with other markers of iron status. The prevalence of anaemia in a population can be used to categorise the issue's importance for public health.

The WHO estimates that the prevalence of anaemia among women of reproductive age worldwide in 2019 was 29.9%, or in numbers, about 500 million women aged 15 to 49. In non-pregnant women of reproductive age, the prevalence was 29.6%, whereas in pregnant women, it was 36.5%. While the prevalence of anaemia among pregnant women has somewhat decreased internationally since 2000, but it has remained steady among women of reproductive age. This happened because the women were not consuming enough calories.

Women who are active require roughly 40 calories per kilogram body weight, says the Harvard Medical School. For a 55 kg woman, this translates to around 2160 calories per day, whereas moderately active women require about 35 calories per kg of body weight per day, or roughly 1920 calories per day for a 55 kg woman. According to the United States Department of Agriculture, sedentary women need about 1600 to 2000 calories per day for good weight control. Researchers discovered that women had a lower age-standardized death rate than males due to this calorie deficiency. They found that women were more likely than men to suffer BMD (Becker Muscular Dystrophy) based on the 2019 Global Burden of Disease Study. The age-standardized mortality rate was much greater for males than for women. They also noted a higher age-standardized disability adjusted life year rate in males while evaluating the burden.

Objectives

To investigate the current state and estimate the future state of stunting and malnutrition in children, as well as the prevalence of food deficiency diseases in females and children. In addition, examining the rate of agriculture production and improvement in hunger index throughout the world.

1) To analyze the prevalence of stunting and malnutrition (wasting and overweightness) in children.
    - To access the prevalence of stunting in children across the world.
    - To access the prevalence of malnutrition in children across the world.
    - To correlate the prevalence of stunting and malnutrition along with the population growth across the world.
    - To illustrate the world regions requiring more work on removing stunting & malnutrition.
    - To estimate the future trend of prevalence of stunting and malnutrition in 2025 and 2030 for various regions of the world.

2) To analyze the prevalence of anaemia in women, on the basis of pregnancy. Additionally, check the regions with lowest per capita food consumption and the people with food deficiency disabilities.
    - To access the prevalence of anaemia in pregnant and non-pregnant women across the world.
    - To check the world regions which have the lowest per capita food consumption in the year 2030.
    - To access the difference in disability-adjusted life years (SEV) and annual % change (EAPC) between females and males in different regions and sub-regions across the world in 1990 and 2019.

3) To access the world regions based on farming and pastoral economy of the countries.
    - To access the growth rate of farming sales and animal feed expenditure across the world.
    - To rank the world regions based on the analysis before.
    - To check and determine the degree of relation between rates of farming sales and animal feed expenditure.

4) To analyze countries on the basis of global hunger index and GDP per capita.
    - To comprehend the rate of improvement in hunger index and GDP across various regions.
    - To check whether *"Hunger Index is a crucial indicator for food security and one that historically has been neglected in the context of developing countries."*
    - To illustrate the relationship between food deficiency diseases, and agricultural economy.

Data Collection

Table 1: Percentage of People suffering from Shunting & Malnutrition per population across various regions from 2000 to 2020.

GeoAreaName	TimePeriod	Value	...	Units
World	2000	33.1	...	PER_POP_U5
World	2001	32.5	...	PER_POP_U5
World	2002	31.9	...	PER_POP_U5
World	2003	31.5	...	PER_POP_U5
World	2004	31.1	...	PER_POP_U5
World	2005	30.7	...	PER_POP_U5
World	2006	30.2	...	PER_POP_U5
World	2007	29.6	...	PER_POP_U5
World	2008	29	...	PER_POP_U5
World	2009	28.4	...	PER_POP_U5
World	2010	27.7	...	PER_POP_U5
World	2011	26.9	...	PER_POP_U5
World	2012	26.2	...	PER_POP_U5
World	2013	25.5	...	PER_POP_U5
World	2014	24.9	...	PER_POP_U5
World	2015	24.4	...	PER_POP_U5
World	2016	23.8	...	PER_POP_U5
World	2017	23.4	...	PER_POP_U5
World	2018	22.9	...	PER_POP_U5
World	2019	22.4	...	PER_POP_U5
World	2020	22	...	PER_POP_U5
Africa	2000	41.5	...	PER_POP_U5
Africa	2001	41.1	...	PER_POP_U5
Africa	2002	40.6	...	PER_POP_U5
Africa	2003	40.1	...	PER_POP_U5
Africa	2004	39.6	...	PER_POP_U5
Africa	2005	39.1	...	PER_POP_U5
Africa	2006	38.5	...	PER_POP_U5
Africa	2007	37.9	...	PER_POP_U5
Africa	2008	37.2	...	PER_POP_U5
Africa	2009	36.6	...	PER_POP_U5
Africa	2010	35.9	...	PER_POP_U5
Africa	2011	35.2	...	PER_POP_U5
Africa	2012	34.5	...	PER_POP_U5
Africa	2013	33.9	...	PER_POP_U5
Africa	2014	33.3	...	PER_POP_U5
Africa	2015	32.8	...	PER_POP_U5
...	...	...	...	...

Table 2.1: Percentage of Anaemia-suffering pregnant women per Population across various regions from 2009 to 2019.

Location	2019	2018	…	2009
(WHO) Global	36.5	36.5	…	36.6
Africa	45.8	45.9	…	46.3
Americas	18.9	18.9	…	19
South-East Asia	47.8	47.8	…	47.9
Europe	23.5	23.4	…	23.3
Eastern Mediterranean	36.8	36.9	…	37.3
Western Pacific	21.3	21.4	…	21.8

Table 2.2: Percentage of Anaemia-suffering non-pregnant women per Population across various regions from 2009 to 2019.

Location	2019	2018	…	2009
(WHO) Global	29.6	29.2	…	28.4
Africa	39.8	39.7	…	41.3
Americas	15.3	15.1	…	16.1
South-East Asia	46.5	46.2	…	45.8
Europe	18.6	18.3	…	17.4
Eastern Mediterranean	34.8	34.6	…	34.8
Western Pacific	16.2	16	…	16.5

Table 3: Global & Region per Capita Food Consumption

(Kcal per Capita per Day).

Region	1964	1974	1984	1997	2015	2030
World	2358	2435	2655	2803	2940	3050
Developing countries	2054	2152	2450	2681	2850	2980
Near East and North Africa	2290	2591	2953	3006	3090	3170
Sub-Saharan Africa	2058	2079	2057	2195	2360	2540
Latin America and the Caribbean	2393	2546	2689	2824	2980	3140
East Asia	1957	2105	2559	2921	3060	3190
South Asia	2017	1986	2205	2403	2700	2900
Industrialized countries	2947	3065	3206	3380	3440	3500
Transition countries	3222	3385	3379	2906	3060	3180

Table 4: Age-Standardized SEV (Smallest Extreme Value Distribution) and EAPC (Estimated Annual Percentage Change) for both Sexes in 1990 & 2019.

Region	Age standardized SEV in 1990 for both sexes	Age standardized SEV in 2019 for both sexes	…	Age standardized SEV in 2019 for males	EAPC from 1990 to 2019 for males
Global	17.1	16.3	…	11.3	-0.34
High-middle SDI	15.8	15.2	…	10.3	-0.27
High SDI	13.9	13.4	…	9.9	-0.19
Low-middle SDI	19.1	17.9	…	12.6	-0.37
Low SDI	21.8	20.7	…	15	-0.25
Middle SDI	19.2	17.2	…	11.7	-0.56
East Asia	20.7	18	…	12.4	-0.74
Southeast asia	20.2	19.2	…	11.2	-0.32
Oceania	15	14.7	…	9.3	0.01
Central Asia	12.7	11.7	…	7.9	-0.23
Central Europe	13.3	11.8	…	6.3	-0.43
Eastern Europe	12.8	12.1	…	8.2	-0.03
High-income Asia Pacific	17	14.7	…	10.3	-0.36
Australasia	14.2	12.3	…	10.3	-0.54
Western Europe	12.2	11.1	…	8.3	-0.32
Southern Latin America	15.6	13.8	…	9.6	-0.39
High-income North America	14	14.8	…	11.2	-0.04
Caribbean	13.4	12	…	9.5	-0.48
Andean Latin America	14.3	12.6	…	9	-0.55
Central Latin America	16	14.7	…	8.9	-0.52
Tropical Latin America	17.3	15.5	…	10	-0.57
North Africa and Middle East	16.1	14.9	…	11.9	-0.22
South Asia	18.2	17.1	…	12.1	-0.36
Central Sub-Saharan Africa	23.8	23.2	…	17	-0.13
Eastern Sub-Saharan Africa	24.9	23.6	…	18.1	-0.29
Southern Sub-Saharan Africa	21.3	19.8	…	15.8	-0.38
Western Sub-Saharan Africa	25.9	25.1	…	16.8	-0.19

Table 5: Total Animal Feeding Expenditure per Country from 2004 to 2016.

Country or Area	Year	Unit	Value
Albania	2016	Mil. USD	3.769911
Albania	2016	Thousand metric tons	13
Albania	2015	Mil. USD	3.810708
Albania	2015	Thousand metric tons	13.11
Albania	2014	Mil. USD	4.275723
Albania	2014	Thousand metric tons	12.5
Albania	2013	Mil. USD	4.982555
Albania	2013	Thousand metric tons	11.9
Albania	2012	Mil. USD	2.159271
Albania	2012	Thousand metric tons	16.5
Albania	2011	Mil. USD	6.005255
Albania	2011	Thousand metric tons	26.9
Albania	2008	Mil. USD	4.809606
Albania	2008	Thousand metric tons	27.4
Albania	2007	Mil. USD	4.390238
Albania	2007	Thousand metric tons	27.2
Albania	2006	Mil. USD	2.086574
Albania	2006	Thousand metric tons	21.7
Albania	2005	Mil. USD	1.252625
Albania	2005	Thousand metric tons	14.6
Albania	2004	Mil. USD	2.59097
Albania	2004	Thousand metric tons	31.78
Algeria	2015	Thousand metric tons	378.45
Algeria	2014	Thousand metric tons	359.88
Algeria	2013	Thousand metric tons	343.8
Algeria	2012	Thousand metric tons	355.81
Algeria	2011	Thousand metric tons	319.9
Algeria	2010	Thousand metric tons	264.97
Algeria	2009	Thousand metric tons	210.59
Algeria	2008	Thousand metric tons	242.04
Algeria	2007	Thousand metric tons	357.64
Algeria	2006	Thousand metric tons	444.6
Algeria	2005	Thousand metric tons	622.96
Algeria	2004	Thousand metric tons	823.14
Algeria	2003	Thousand metric tons	809.73
Algeria	2002	Thousand metric tons	980
Algeria	2001	Thousand metric tons	898
Algeria	2000	Thousand metric tons	864
Algeria	1999	Thousand metric tons	1008
Algeria	1998	Thousand metric tons	1274
Algeria	1997	Thousand metric tons	1265
Algeria	1996	Thousand metric tons	1127
Algeria	1995	Thousand metric tons	1291
Armenia	2016	Thousand metric tons	87
Armenia	2015	Thousand metric tons	88
…	…	…	…

Table 6: Total Farming Sales per Country from 2000 to 2021.

Country or Area	Year	Currency	...	Value
Afghanistan	2021	Afghani	...	4.13E+11
Afghanistan	2020	Afghani	...	4.15E+11
Afghanistan	2019	Afghani	...	3.79E+11
Afghanistan	2018	Afghani	...	2.93E+11
Afghanistan	2017	Afghani	...	3.40E+11
Afghanistan	2016	Afghani	...	3.15E+11
Albania	2021	lek	...	3.34E+11
Albania	2020	lek	...	3.17E+11
Albania	2019	lek	...	3.11E+11
Albania	2018	lek	...	3.02E+11
Albania	2017	lek	...	2.95E+11
Albania	2016	lek	...	2.92E+11
Albania	2015	lek	...	2.84E+11
Albania	2014	lek	...	2.79E+11
Albania	2013	lek	...	2.64E+11
Albania	2012	lek	...	2.50E+11
Albania	2011	lek	...	2.37E+11
Albania	2010	lek	...	2.23E+11
Albania	2009	lek	...	1.92E+11
Albania	2008	lek	...	1.82E+11
Albania	2007	lek	...	1.66E+11
Albania	2006	lek	...	1.55E+11
Albania	2005	lek	...	1.52E+11
Albania	2004	lek	...	1.51E+11
Albania	2003	lek	...	1.49E+11
Albania	2002	lek	...	1.34E+11
Albania	2001	lek	...	1.28E+11
Albania	2000	lek	...	1.23E+11
Andorra	2021	Euro	...	14380000
Andorra	2020	Euro	...	14160197
Andorra	2019	Euro	...	14732032
Andorra	2018	Euro	...	14894511
Andorra	2017	Euro	...	14864561
Andorra	2017	Euro	...	12928655
Andorra	2016	Euro	...	13656737
Andorra	2016	Euro	...	12799675
Andorra	2015	Euro	...	12912032
Andorra	2015	Euro	...	11918579
Andorra	2014	Euro	...	13244486
Andorra	2014	Euro	...	12530107
Andorra	2013	Euro	...	14081698
Andorra	2013	Euro	...	14268801
Andorra	2012	Euro	...	14921437
Andorra	2012	Euro	...	15079009
Andorra	2011	Euro	...	12792911
...	...	...	...	...

Table 7: The Global Hunger Index along with Country Ranks.

Rank	Country	2000	2007	2014	2022
1	Belarus	< 5	< 5	< 5	< 5
1	Bosnia & Herzegovina	9.3	6.6	< 5	< 5
1	Chile	< 5	< 5	< 5	< 5
1	China	13.3	7.8	< 5	< 5
1	Croatia	< 5	< 5	< 5	< 5
1	Estonia	< 5	< 5	< 5	< 5
1	Hungary	5.5	< 5	< 5	< 5
1	Kuwait	< 5	< 5	< 5	< 5
1	Latvia	5.6	< 5	< 5	< 5
1	Lithuania	5.4	< 5	< 5	< 5
1	North Macedonia	7.5	7.2	< 5	< 5
1	Romania	7.9	5.8	5.1	< 5
1	Serbia	NaN	6.1	5.8	< 5
1	Slovak Republic	7	5.9	5.7	< 5
1	Turkey	10.1	5.8	< 5	< 5
1	Uruguay	7.4	6.5	< 5	< 5
18	Costa Rica	7	< 5	< 5	5.3
18	UAE	6.2	6.5	5.9	5.3
20	Brazil	11.4	7.1	5	5.4
21	Uzbekistan	24.2	15.4	8.3	5.6
22	Georgia	12.3	7.8	6.1	5.7
22	Mongolia	30	21.8	9.2	5.7
24	Bulgaria	8.6	7.9	7.4	5.9
24	Kazakhstan	11.2	11.6	5.8	5.9
26	Tunisia	10.3	7.6	6.7	6.1
27	Albania	20.7	15.8	9.2	6.2
28	Russia	10.1	7.1	6.7	6.4
29	Iran	13.7	8.8	7.4	6.5
30	Saudi Arabia	11	12.2	7.4	6.7
31	Argentina	6.6	5.5	5	6.8
32	Algeria	14.5	11.4	8.7	6.9
32	Armenia	19.3	12.1	7.3	6.9
32	Moldova	18.7	20.3	6.8	6.9
35	Jamaica	8.6	8.1	8.8	7
36	Azerbaijan	24.9	15.3	9.3	7.5
36	Ukraine	13	7.2	7.2	7.5
38	Colombia	10.9	11.2	8.6	7.6
38	Peru	20.6	15	7.6	7.6
40	Kyrgyz Republic	18	13.6	9.4	7.8
41	Paraguay	11.6	11.4	8.1	8
42	Mexico	10.2	8.5	7	8.1
42	Panama	18.6	14	9.4	8.1
44	El Salvador	14.7	12.1	10.4	8.4
45	Dominican Republic	15	13.9	9.8	8.8
…	…	…	…	…	…

Table 8: The Gross Domestic Product per Capita for each Country.

Country Name	Country Code	1960	1961	...	2021
Aruba	ABW	NaN	NaN	...	29342.1
Afghanistan	AFG	62.36937	62.4437	...	368.7546
Angola	AGO	NaN	NaN	...	1953.534
Albania	ALB	NaN	NaN	...	6492.872
Andorra	AND	NaN	NaN	...	42137.33
United Arab Emirates	ARE	NaN	NaN	...	44315.55
Argentina	ARG	NaN	1163.187	...	10636.12
Armenia	ARM	NaN	NaN	...	4966.513
American Samoa	ASM	NaN	NaN	...	15743.31
Antigua and Barbuda	ATG	NaN	NaN	...	15781.4
Australia	AUS	1810.51	1877.51	...	60443.11
Austria	AUT	935.4604	1031.815	...	53637.71
Azerbaijan	AZE	NaN	NaN	...	5387.998
Burundi	BDI	71.36022	72.08878	...	221.4777
Belgium	BEL	1273.692	1350.198	...	51247.01
Benin	BEN	90.03583	92.37486	...	1319.155
Burkina Faso	BFA	69.0832	72.17377	...	893.0772
Bangladesh	BGD	84.82533	92.85511	...	2457.925
Bulgaria	BGR	NaN	NaN	...	12221.5
Bahrain	BHR	NaN	NaN	...	26562.97
Bahamas, The	BHS	1483.004	1581.304	...	27478.39
Bosnia and Herzegovina	BIH	NaN	NaN	...	7143.311
Belarus	BLR	NaN	NaN	...	7302.258
Belize	BLZ	307.1222	319.5961	...	6228.267
Bermuda	BMU	1902.402	1961.538	...	114090.3
Bolivia	BOL	100.8437	107.4537	...	3345.197
Brazil	BRA	232.9988	229.3368	...	7507.161
Barbados	BRB	NaN	NaN	...	17225.46
Brunei Darussalam	BRN	NaN	NaN	...	31449.08
Bhutan	BTN	NaN	NaN	...	3266.365
Botswana	BWA	59.29886	62.74246	...	6805.221
Central African Republic	CAF	66.7701	71.9932	...	461.1375
Canada	CAN	2259.251	2240.433	...	51987.94
Switzerland	CHE	1787.36	1971.316	...	91991.6
Channel Islands	CHI	NaN	NaN	...	NaN
Chile	CHL	504.8011	554.4846	...	16265.1
China	CHN	89.52054	75.80584	...	12556.33
Cote d'Ivoire	CIV	147.2778	160.6527	...	2549.041
Cameroon	CMR	120.0182	125.5256	...	1666.933
Congo, Dem. Rep.	COD	219.9058	196.9432	...	577.2092
...	...	...	...	...	...

Methodology

Statistics-

Statistics is a discipline which concerns with the collection, organization, analysis, and representation of data.

Descriptive Statistics-

Descriptive statistics is the quantitative description and analysis of properties of a set of information or data.

- Mean:

  It is the average value of the data.

  Sample Mean ($\overline{x}$) -

$$\overline{x} = \frac{\sum x}{n}$$

  Population Mean (μ) –

$$\mu = \frac{\sum x}{N}$$

- Median:

  It is the middle value of the set of data when the data is in either ascending or descending order.

  When n or N is odd, Median =

$$\left(\frac{n+1}{2}\right)^{th} Observation$$

  When n or N is even, Median =

$$\frac{\left(\frac{n}{2}\right)^{th} Observation + \left(\frac{n+1}{2}\right)^{th} Observation}{2}$$

❖ Mode:

It is the data value that occurs with the greatest frequency in the data.

❖ Standard Deviation:

It is the value which tells that how much the data is scattered from the average value.

Sample Standard Deviation (s) –

$$s = \sqrt{\frac{\sum (x - \overline{x})^2}{n - 1}}$$

Population Standard Deviation (σ) –

$$\sigma = \sqrt{\frac{\sum (x - \mu)^2}{N}}$$

❖ Variance:

It is the square of the standard deviation and gives a statistical inference about the spread between the numbers in the data set.

Sample Variance (s^2) –

$$s^2 = \frac{\sum (x - \overline{x})^2}{n - 1}$$

Population Variance (σ^2) –

$$\sigma^2 = \frac{\sum (x - \mu)^2}{N}$$

Where, x is the i[th] observation,

n is the total no. of observations in sample data,

N is the total no. of observations in a population.

Inferential Statistics-

Inferential Statistics is the field of utilising data analysis to determine attributes of an underlying probability distribution. It derives the characteristics of a population from sampled data.

Correlation Analysis

Correlation Analysis is a statistical technique that is used to determine the strength of any potential relationships between two variables or datasets. There are many types of Correlation in Statistics.

- ❖ Pearson Correlation:

  Pearson Correlation Coefficient, or simply Pearson's r, is a measure of linear correlation between two sets of data. It was developed by Karl Pearson.

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 \ \sum(y_i - \overline{y})^2}}$$

Regression Analysis

Regression Analysis is a group of statistical procedures which are used to determine the association between a dependent variable and one or more independent variables.

- ❖ Linear Regression:

  Linear regression is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables in statistics.

$$Y_i = f(X_i, \beta) + e_i$$

Testing of Hypothesis

Hypothesis Testing is a method of Statistical Inference which is used to decide whether the data supports a particular hypothesis sufficiently or not.

- ❖ Null Hypothesis:

  The Hypothesis which is aimed to be rejected.

$$H_0: a \geq / \leq / = \ b$$

- ❖ Alternative Hypothesis:

  The Hypothesis which is accepted on the rejection of Null Hypothesis.

$$H_a = a > / < / \neq \ b$$

Statistical Tests

- T-Test:

  T-Test is a Statistical Method which is used to compare the average values of two data sets and determine if they belong to the same population or are they different.

$$t = \frac{(m-\mu)}{\left(\frac{s}{\sqrt{n}}\right)}$$

- Z-Test:

  Z-test is a Statistical Method which is used to determine whether two population means are different. It should be used when the variances are known, and the sample size is large.

$$z = \frac{(\overline{x}-\mu_0)}{s}$$

- Chi-Square Test:

  Chi-Square Test is a Statistical Method which is used to compare two variables in a contingency table to see if they are related to each other or not.

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

- ANOVA:

  ANOVA or Analysis of Variance is a Statistical Method which is used to compare the variances across the means of different groups.

$$F = \frac{(k-1)}{\sum_{i=1}^{k}\sum_{j=1}^{t}(X-\overline{X_j})^2} \times \frac{\sum_{j=1}^{k}(\overline{X_j}-\overline{X})^2}{(n-k)}$$

- Kolmogorov-Smirnov Test:
  Kolmogorov–Smirnov test is a nonparametric test in Statistics that can be used to check if a sample comes from a specific distribution or not.

$$D = Maximum|F_o(X) - F_r(X)|$$

Where, $F_o(X)$ = Observed Frequency Distribution

$F_r(X)$ = Theoretical Frequency Distribution

Python Modules-

Scientific Computing

- NumPy

  NumPy is a Python library which adds support for arrays and matrices, and a large collection of high-level mathematical functions.

- SciPy

  SciPy is a Python library which is used for scientific computing. It contains modules for optimization, linear algebra, integration, interpolation, and other common tasks in science.

- scikit-learn

  scikit-learn, or sklearn is a Python library which is used for machine learning. It features various classification, regression and clustering algorithms, along with many more scientific methods.

Data Handling

- pandas

  pandas is a Python library which is used for data manipulation and data analysis. Specifically, it offers data structures and mathematical operations for manipulating numerical tables.

- GeoPandas

  GeoPandas is a Python library which is used to add support for geographic data to pandas objects. It integrates the libraries of pandas and shapely in order to perform geometric operations.

Graphs and Plotting

- Matplotlib

  Matplotlib is a Python library which is used for creating static and animated visualizations.

- Plotly for Python

  Plotly is a set of graphing libraries, for languages like Python, R, Julia, and many more, which are used for creating interactive charts and maps.

Analysis & Interpretation

1) To analyze the prevalence of stunting and malnutrition (wasting and overweightness) in children.

a) To access the prevalence of stunting in children across the world.

Steps-

- Firstly, we categorise the Total Data [Table 1] based on Regions, Sub-Regions, and Countries.
- Then, we access the data for Stunting, calculate the Compound Annual Growth Rate (CAGR), and convert it to a Dictionary.
- After which, we calculate the Descriptive Statistical Information and plot the following graph.

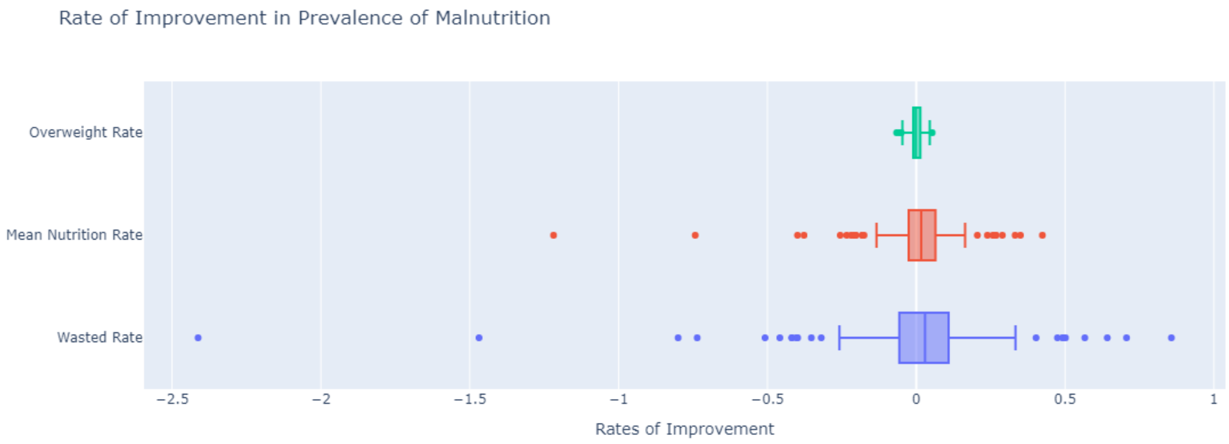
- Along with this, we perform the Kolmogorov–Smirnov Test in order to check the normality of the data.
- From the Output, we interpret that-
    - The Mean Rate of Improvement for the Prevalence of Stunting all across the world is **0.021 % per year**.
    - The Median Rate of Improvement for the Prevalence of Stunting all across the world is **0.020 % per year**.
    - The Worst Rate of Improvement for the Prevalence of Stunting all across the world is **-0.038 % per year**, which is in **Libya**.
    - The Best Rate of Improvement for the Prevalence of Stunting all across the world is **0.076 % per year**, which is in **Tonga**.
    - The KS Test gives the P-Value as **3.88e-35**, which proves that the data is not Normal.

With this, we know that the Rate of Improvement for the Prevalence of Stunting is very slow and is even slower in African countries like Libya.

b) To access the prevalence of malnutrition in children across the world.

Steps-

- First, we access the data for Wasting and Overweightness, and convert it to a Dictionary.
- Then, we find the CAGR from the Mean Prevalence of Malnutrition which was taken out from the Mean of the values in Prevalence of Wasting and Prevalence of Overweightness for each country.
- After which, we calculate the Descriptive Statistical Information and plot the following graph.



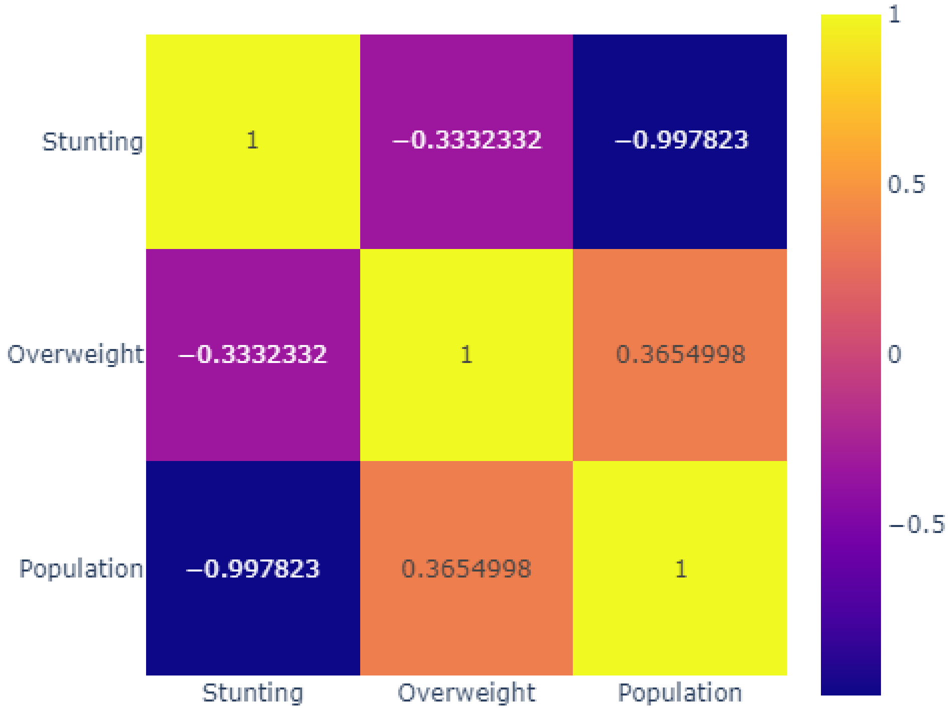
- Along with this, we perform the Kolmogorov–Smirnov Test in order to check the normality of the data.
- From the Output, we interpret that-
    - The Mean Rate of Improvement for the Prevalence of Malnutrition all across the world is **0.003 % per year**.
    - The Median Rate of Improvement for the Prevalence of Malnutrition all across the world is **0.017 % per year**.
    - The Worst Rate of Improvement for the Prevalence of Malnutrition all across the world is **-1.218 % per year**, which is in **Papua New Guinea**.
    - The Best Rate of Improvement for the Prevalence of Malnutrition all across the world is **0.423 % per year**, which is in **Tonga**.
    - The KS Test gives the P-Value as **6.09e-16**, which proves that the data is not Normal.

With this, we know that the Rate of Improvement for the Prevalence of Malnutrition (Wasting and Overweightness) is slow. It is also very bad in many countries, as they are going towards deterioration from the current condition.

c) To correlate the prevalence of stunting and malnutrition along with the population growth across the world.

Steps-

- First, we access the data for Stunting, Wasting and Overweight in world regions and convert it to a Pandas DataFrame.
- Then, we find the Pearson Correlation out of the DataFrame.
- After which, we plot the following graph.



- From the Output, we interpret that-
  - The Correlation coefficient between the Prevalence of Stunting and the Prevalence of Overweightness is **-0.333**, which shows that Stunting and Overweightness are not very related to each other.
  - The Correlation coefficient between the Prevalence of Stunting and the Region's Population is **-0.997**, which shows that Stunting is dependent on a place's population.
  - The Correlation coefficient between the Prevalence of Overweightness and the Region's Population is **0.365**, which shows that Overweightness is not really dependent on a place's population.

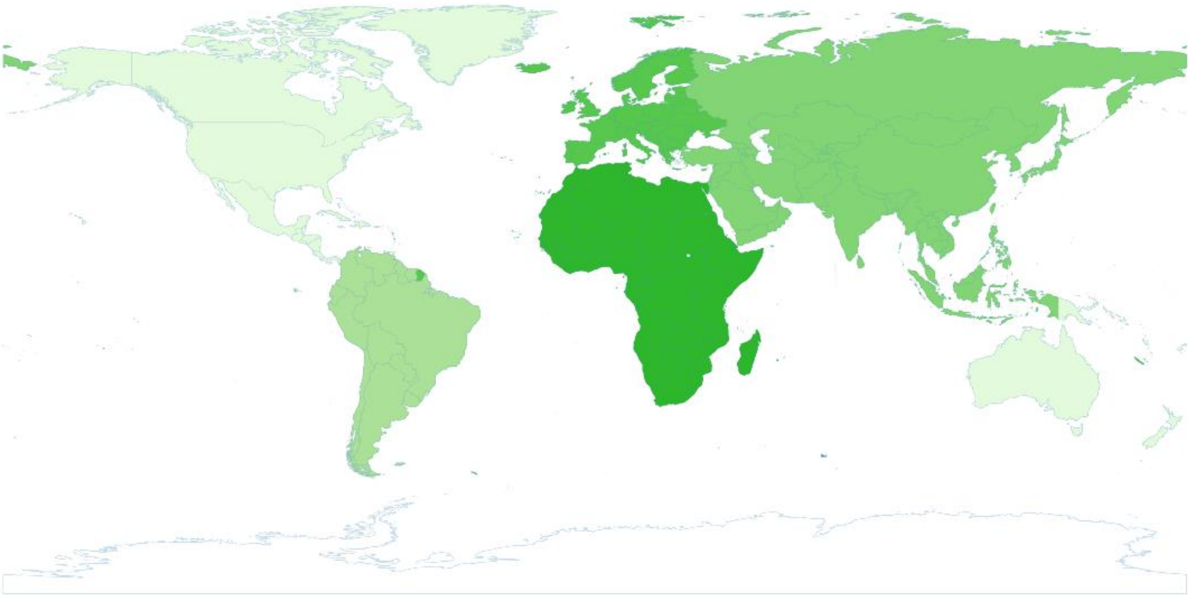
Here, the data for Wasting is not shown because of the absence of information on Wasting in many developed regions of the world.

d) To illustrate the world regions requiring more work on removing stunting & malnutrition.

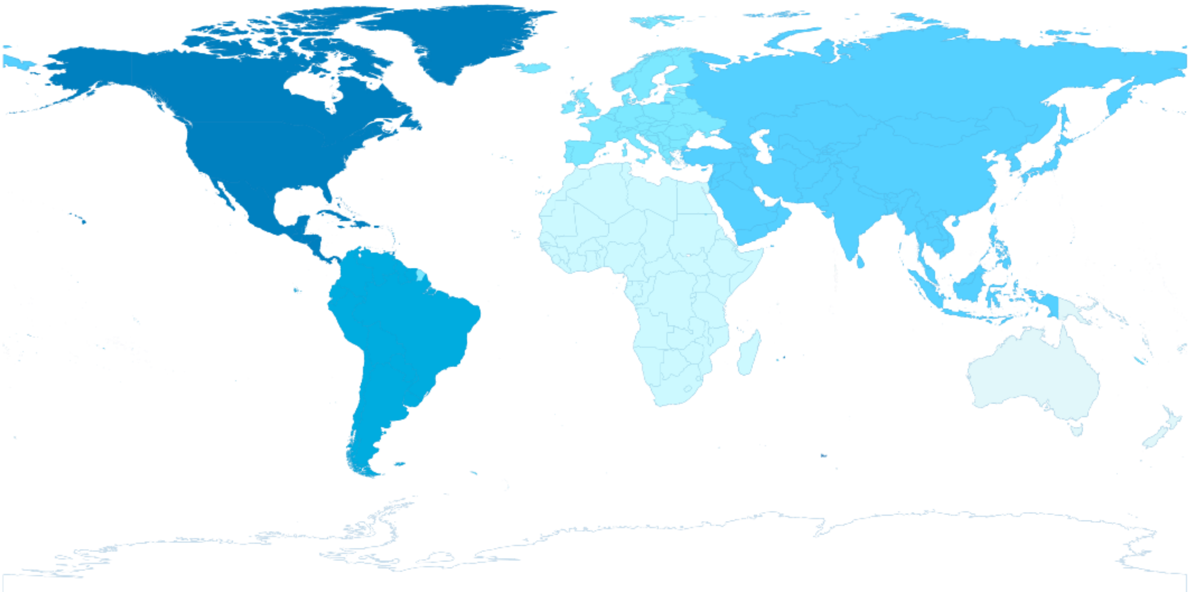
Steps-

- First, we access the data for Stunting, Wasting and Overweight in world regions and calculate the CAGR from it.
- Then, according to the CAGR of each region, we plot the following maps.

Requirement of Improvement in Stunting across Continents



Requirement of Improvement in Overweight across Continents



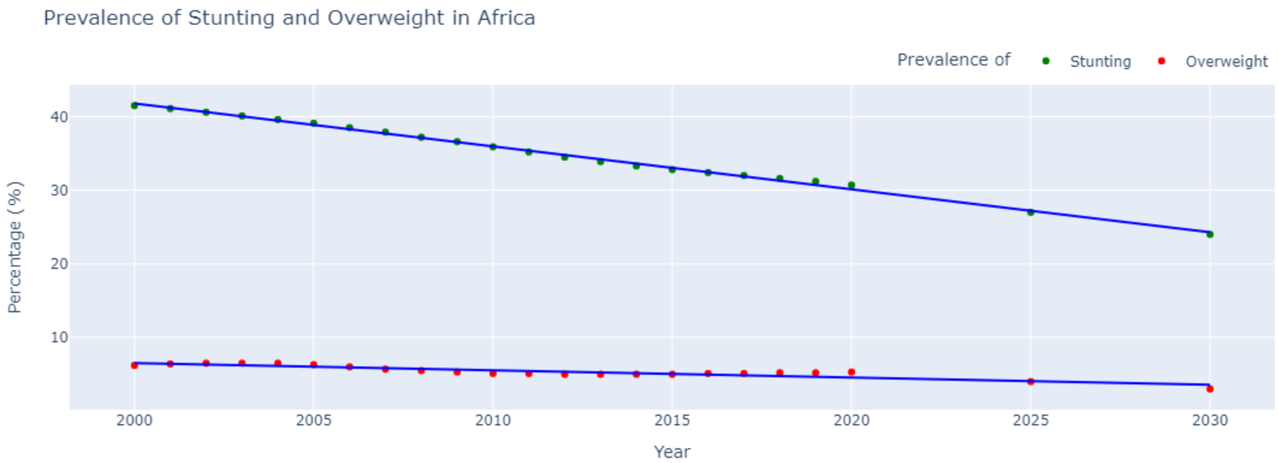
- From the Output, we interpret that-
    - The regions most affected with Stunting and Wasting are Africa and Asia, while that for Overweightness are North America and South America.

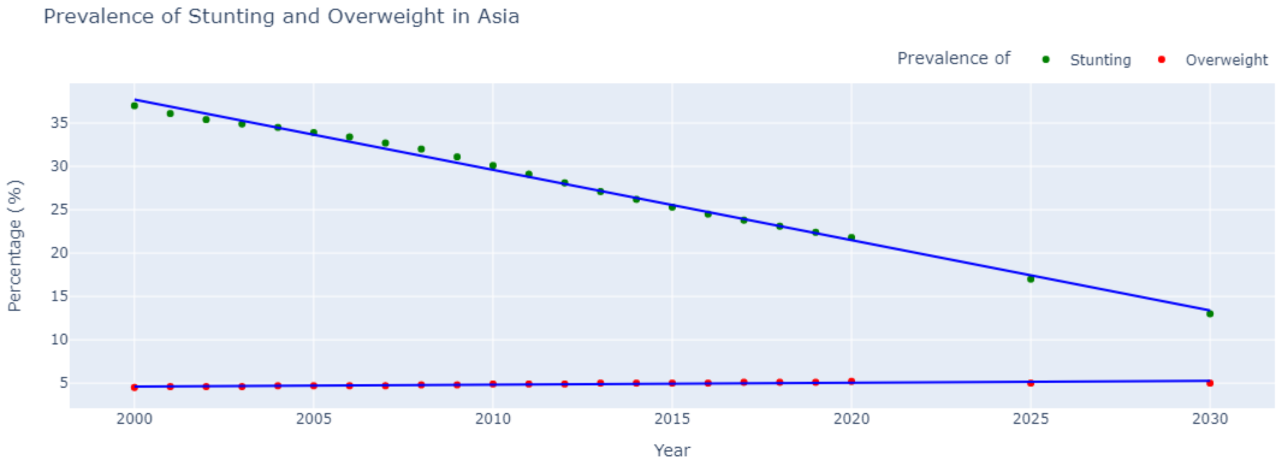
With this, we also interpret that malnutrition is dependent on both High Intake of food, mostly in Developed or Developing countries and Low Intake of food, mostly in Underdeveloped countries.

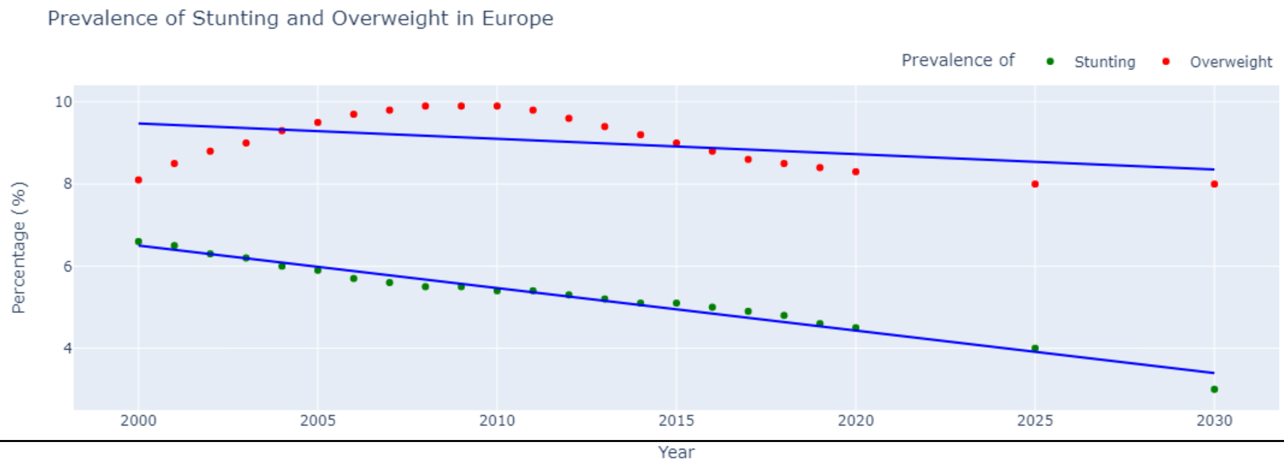
e) To estimate the future trend of prevalence of stunting and malnutrition in 2025 and 2030 for various regions of the world.

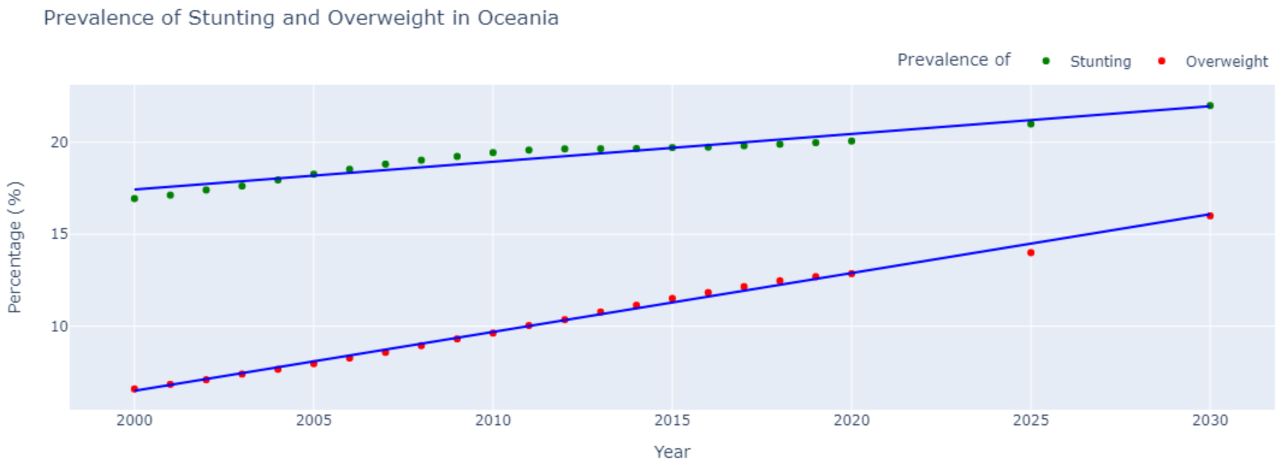
Steps-

- Firstly, we access the data for Stunting and Overweightness in world regions and create a Linear Regression model from it.
- With the help of the Regression model, we estimate the Prevalence of Stunting and Overweightness for the years 2025 and 2030 (in integer form).
- From the estimates, we create the following graphs for each world region.

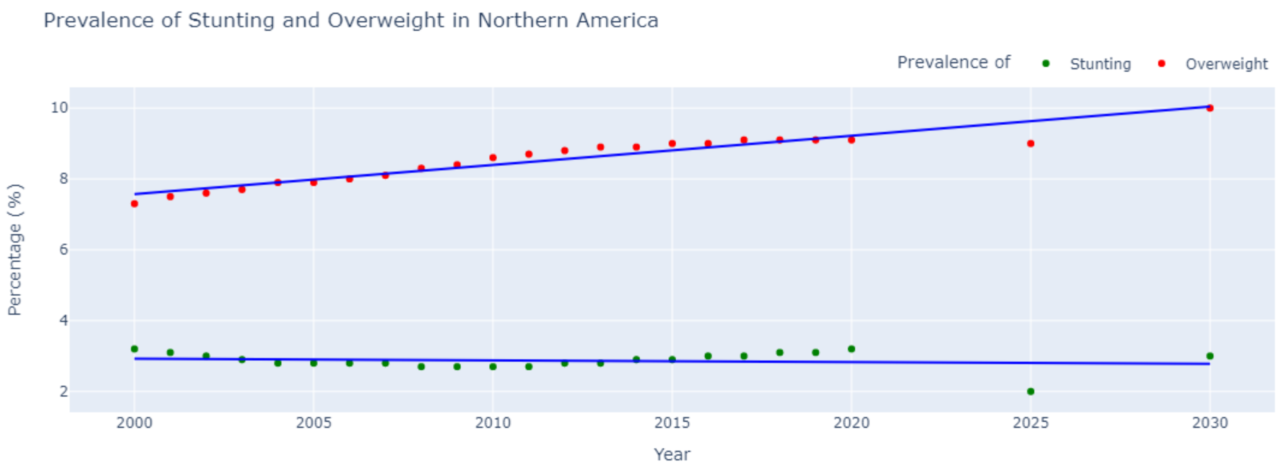




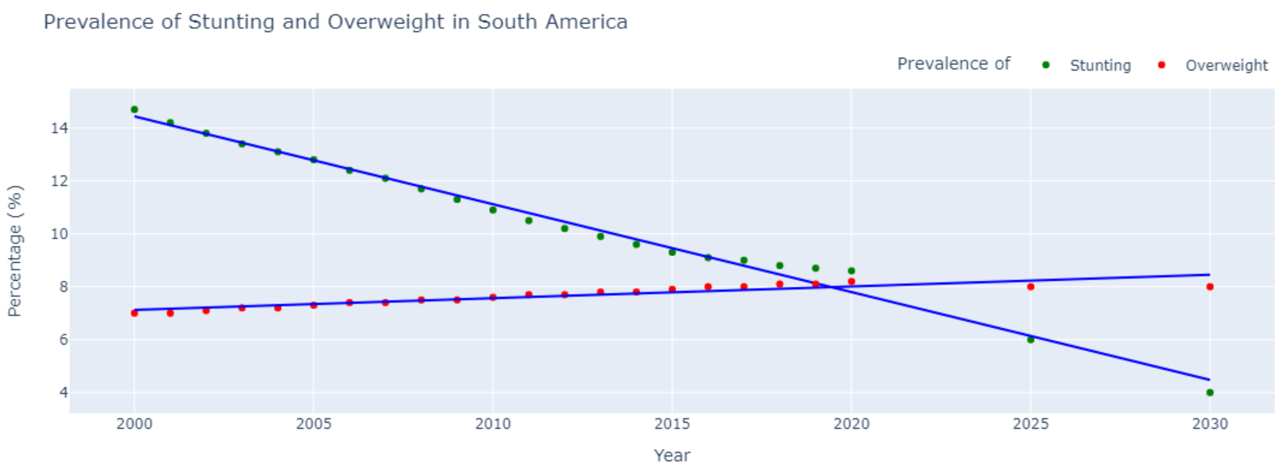




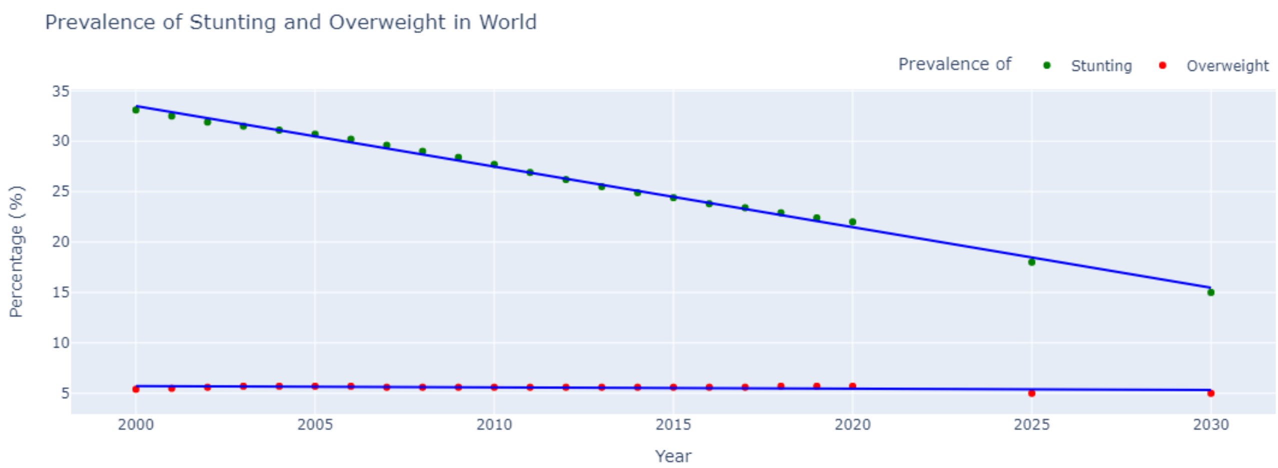
Prevalence of Stunting and Overweight in Oceania
Prevalence of
Stunting
Overweight
Percentage (%)
Year



Prevalence of Stunting and Overweight in Northern America
Prevalence of
Stunting
Overweight
Percentage (%)
Year



Prevalence of Stunting and Overweight in South America
Prevalence of
Stunting
Overweight
Percentage (%)
Year



Prevalence of Stunting and Overweight in World
Prevalence of
Stunting
Overweight
Percentage (%)
Year

- From the Output, we interpret that-
    - The estimated rate of Prevalence of Stunting-
        - in 2025 for-
            - Africa = 27%
            - Asia = 17%
            - Europe = 4%
            - Oceania = 21%
            - North America = 2%
            - South America = 6%
            - World = 18%
        - in 2030 for-
            - Africa = 24%
            - Asia = 13%
            - Europe = 3%
            - Oceania = 22%
            - North America = 3%
            - South America = 4%
            - World = 15%
    - The estimated rate of Prevalence of Overweightness-
        - in 2025 for-
            - Africa = 4%
            - Asia = 5%
            - Europe = 8%
            - Oceania = 14%
            - North America = 9%
            - South America = 8%
            - World = 5%
        - in 2030 for-
            - Africa = 3%
            - Asia = 5%
            - Europe = 8%
            - Oceania = 16%
            - North America = 10%
            - South America = 8%
            - World = 5%

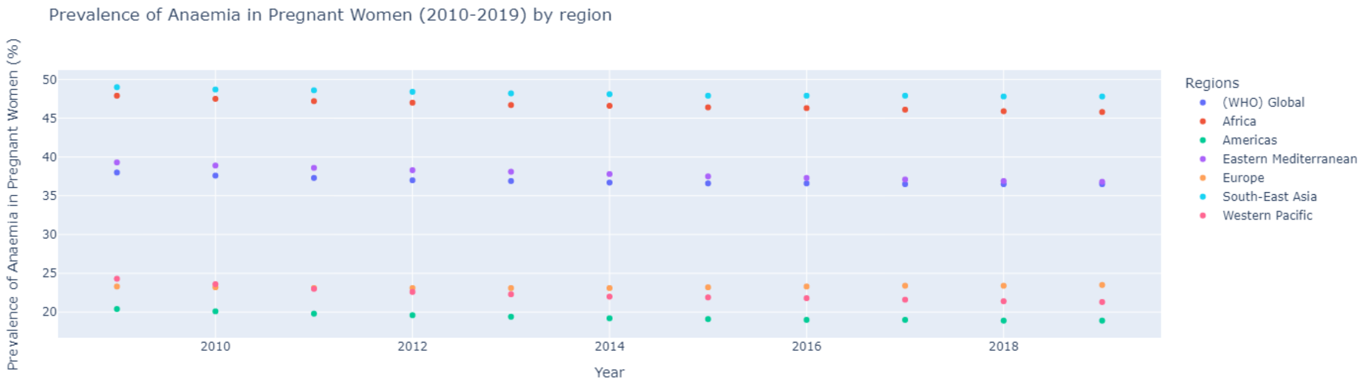
  With this, we know that Stunting is going to significantly decrease in the future, but Overweightness will start to become a serious problem as the countries develop over time.

2) To analyze the prevalence of anaemia in women, on the basis of pregnancy. Additionally, check the regions with lowest per capita food consumption and the people with food deficiency disabilities.

a) To access the prevalence of anaemia in pregnant and non-pregnant women across the world.

Steps-

- Firstly, we categorize the total data from [Table 2.1] and [Table 2.2] based on % of pregnant women having anaemia in 2009, pregnant women having anaemia in 2019, non-pregnant women having anaemia in 2009, non-pregnant women having anaemia in 2019.
- Then, the Pearson correlation function returns the correlation coefficient rounded to 5 decimal places
- The correlation coefficient for both years is quite high near 0.98 which indicates a strong positive correlation between the two variables.



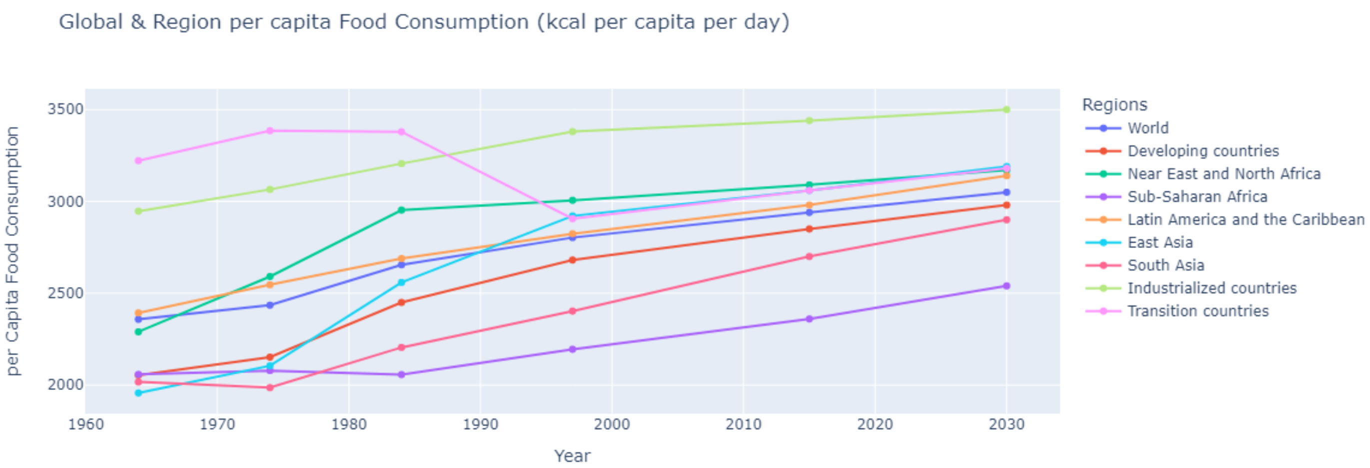
- This suggests that there is a strong association between the prevalence of anaemia in pregnant and non-pregnant women in both years. It can be concluded that both in 2009 and 2019 the correlation between anaemia in pregnant and non-pregnant women is strong, meaning that the prevalence of anaemia in pregnant women is strongly associated with the prevalence of anaemia in non-pregnant women.

b) To check the world regions which have the lowest per capita food consumption in the year 2030.

Steps-

- Firstly, we access the data from table from [Table3] based on region and kcal per capita per day for years 1964, 1974, 1984, 1997, 2015 and 2030.
- Then we create a line plot that visualizes the global and regional per capita food consumption (in kcal per capita per day) over time, using the Plotly library.

- It generates a line plot that shows the per capita food consumption over time for different regions, with different line for each region. The plot should display the title, axis titles, and legend with the regions. With this plot, one can easily compare the food consumption of different regions over the years.
- Then, we a threshold or benchmark value of 3000 for the per capita food consumption, and call the ttest() function and pass in the data for the year 2030 and the threshold value as arguments.



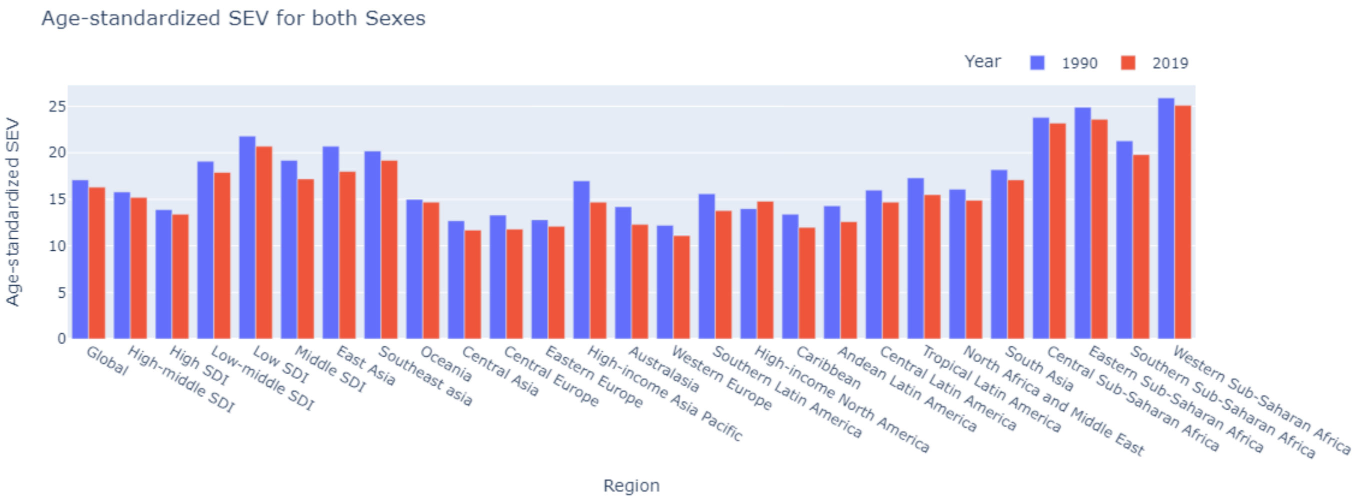
- The P-value is greater than or equal to 0.05, so the null hypothesis is not rejected, which means that the mean of the per capita food consumption in 2030 for the region of interest is equal to the threshold of 3000.

c) To access the difference in disability-adjusted life years (SEV) and annual % change (EAPC) between females and males in different regions and sub-regions across the world in 1990 and 2019.

Steps-

- By [table 4] creating a bar chart that visualizes the age standardized SEV for both sexes in different regions in the years 1990 and 2019, using the Plotly library.
- It starts by defining two lists, one for the regions and one for the SEV of both sexes in 1990 and 2019. It then creates a new figure object using the go.Figure() function from the Plotly library.
- It then creates two traces for the bar chart using the go.Bar() function, passing in the regions list as the x-values, and the SEV of both sexes in 1990 and 2019 as the y-values. It also sets the name of the trace to be the year 1990 and 2019 respectively.
- calculates the difference between SEV for females and males in 1990 and 2019 by subtracting the SEV of males from the SEV of females. It then prints the difference between SEV for females and males in 1990 and 2019. This will indicate the

difference in the SEV of females and males in both years. It can be used to investigate gender inequality in the socioeconomic vulnerability.



- In conclusion, it generates a bar chart that visualizes the age standardized SEV for both sexes in different regions in the years 1990 and 2019 and also calculates the difference between SEV for females and males in 1990 and 2019, which can be used to investigate gender inequality in the socioeconomic vulnerability.

3) To rank world regions based on farming and pastoral economy of the countries.

a) To access the growth rate of farming sales and animal feed expenditure across the world.

Steps-

- After reading the csv data. We specify the data for that The first step is to get the unique country names from both datasets using the np.unique() function and the to_numpy() method.
- These unique country names are stored in the variables countriesInFarmingData and countriesInAnimalsData.
- Then, a for loop is used to iterate through the smaller list of unique countries, countriesInAnimalsData.
- Within the for loop, the dataFarming and dataAnimals datasets are filtered by the current country name, and the unique years are extracted. If the length of the years lists is not zero, the farming and animals data values are extracted, and a nested dictionary is created with these values, and the year as key, this nested dictionary is finally added to the allData dictionary with the current country name as key.
- The final allData object is a dictionary containing one entry for each country, where each entry contains another dictionary with 'Farming' and 'Animals' keys, which in turn contain a dictionary for each year and the respective value.
- To define the growth rate, we will takes a dataset as input and calculates the growth rate of that dataset. The growth rate is calculated by taking the last value of the dataset (dataset[-1]) and dividing it by the first value of the dataset (dataset[0]), raising it to the power of 1 divided by the number of elements in the dataset minus 1, subtracting 1 and then multiplying by -1.
- Then, the code iterates through the countries in the allData dictionary and for each country, it extracts the farming and animals' data for that country. Then it calculates the growth rate for each of them using the growthRate function and storing the results in a new dictionary called growthRatesData. The growth rates for farming sales and animals feed spendings are printed for each country.
- It also includes a try-except block, this block will help to pass the iteration if any exception occurred while calculating the growth rate, this can happen if the dataset is empty or if the dataset has only one value, in both cases the growth rate cannot be calculated.
- After this convert the data to list.

- From the Output, we interpret that-
    - Basically, we calculated some basic statistical measures of a dataset, specifically the mean, median, mode, standard deviation, and variance of the dataset of total farming sales it is:
        - 0.04,
        - 0.04,
        - 0.019,
        - 0.038,
        - 0.001, respectively.
    - Same statistical measures are done for animal feeding expenditure
        - Mean - 0.094
        - Median - 0.094
        - Mode - 0.26
        - Standard Deviation - 0.15
        - Variance - 0.024

b) To rank the world regions based on the analysis before.

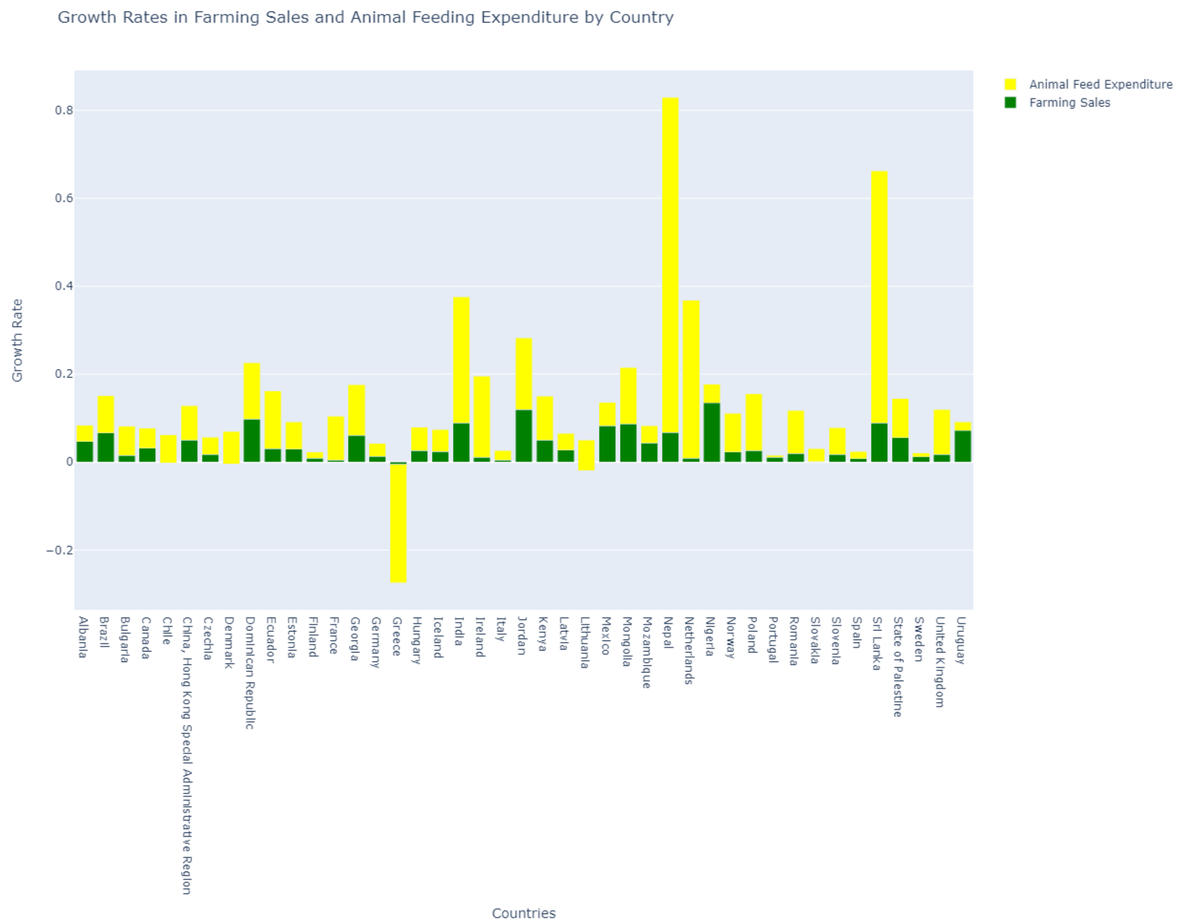
Steps-

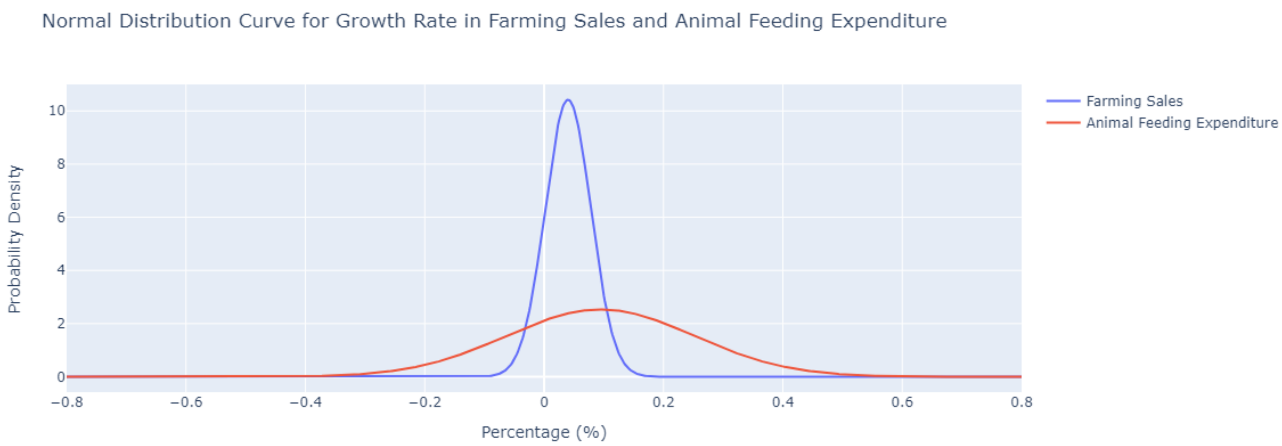
- After specifying the data, we will rank the world on the basis farming sales and animal feeding expenditure for this we performed several operations to generate a list of ranks of data in the input dataset.
- First, it creates an empty list called dataarray. Then, it iterates through the input dataset dat in nested loops, comparing each element in the dataset with every other element, and appending a value of 0, 1, 2, 3, or 4 to the dataarray list depending on the comparison of the elements' first and second values.
- Next, it converts dataarray into a 2D numpy array called matrix, with dimensions of 43x43, by using the reshape function.
- Then, it creates an array called eigVec of shape (43,1) and fills it with a value of 1. Then it performs matrix multiplication on matrix and eigVec for 7 iterations, in each iteration eigVec is updated as eigVec = np.matmul(matrix, eigVec)
- After that, it finds the maximum value of eigVec and then normalize the vector by dividing it by the max value and stores the result in the same variable eigVec.
- Then, it creates a new list dat2 by zipping the dataset cData and the first element of the eigenvector eigVec, and sorts dat2 in descending order by the second element of each tuple using the sorted function.
- After all these steps we get world ranked on the basis of farming sales and animal feeding expenditure.

c) To check and determine the degree of relation between rates of farming sales and animal feed expenditure.

Steps-

- We plotted one simple bar graph on the growth rate of farming sales and animal feeding expenditure country by country.
- At last, we made a normal distribution curve graph for the growth rate of farming sales and animal feeding expenditure to their percentage density.





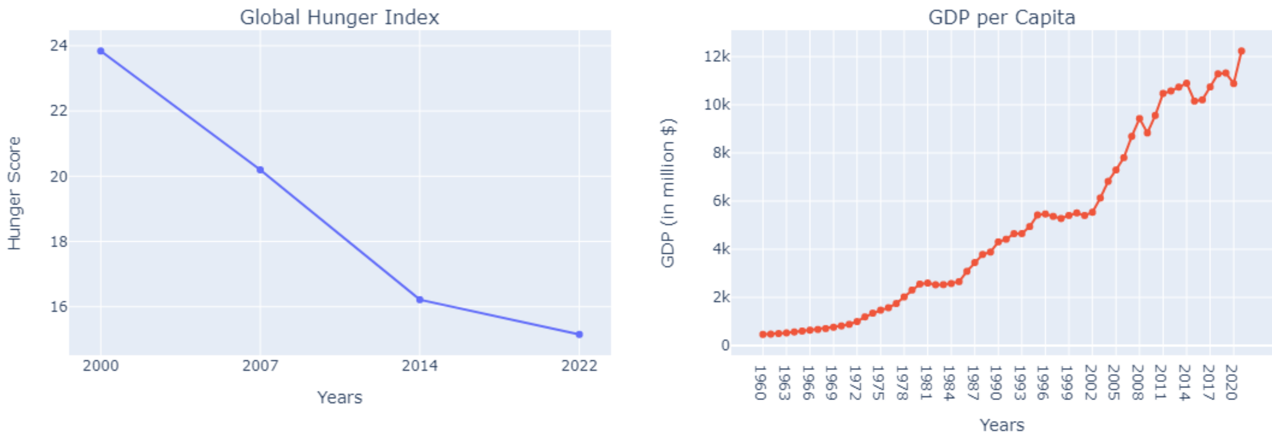
After observing all problem statements, ranking of word, data tables and graphs we can conclude that the Farming sales are much lesser than animal feeding expenditures in every country except Greece which is not a positive outcome. This is making agriculture growth difficult in many countries and becomes one of the major issues for the goal of Zero Hunger.

4) To analyze countries on the basis of global hunger index and GDP per capita.

a) To comprehend the rate of improvement in hunger index and GDP across various regions.

Steps-

- After reading the csv data, we first define a function called 'converter' that handles non-float values in the GHI data. It tries to convert the array elements into float values. If the conversion is successful, the float value is added to a new array, if not, the value 2.5 is added. This function is used to create four arrays for the GHI values in 2000, 2007, 2014, and 2022.
- Next, the code configures the data for the graph. Two arrays are created, 'yearsGHI' and 'valueGHI', which are used to store the years and mean values of the GHI, respectively. Similarly, 'yearsGDP' and 'valueGDP' arrays are created, which store the years and GDP per capita values, respectively.
- Then the code creates the graph with 1 row and 2 columns where the first subplot is for the GHI and the second one is for GDP per capita.



- From the output, we interpret that-
  - By comparing the values of hunger index and GDP per capita for different years, we will be able to comprehend the rate of improvement in hunger index and GDP across various regions.
  - We can also analyze the correlation between hunger index and GDP and identify the regions where food security is a major concern.
  - Additionally, by tracking the progress made in addressing hunger and food insecurity over time, we can get a better understanding of the relationship between food security and economic development.

As we can see in the result, with the increase of GDP per capita the hunger index is getting decrease which is a good indicator. A higher GDP per capita often indicates a stronger economy and higher standards of living, which can lead to improved food security and reduced hunger. This is why the decrease in hunger index as GDP per capita increases is often considered a good indicator of progress towards food security.

b) To check whether *“Hunger Index is a crucial indicator for food security and one that historically has been neglected in the context of developing countries.”*

Steps-

- We access the data on Hunger Index and GDP per capita for different countries over time. Then, we perform a T-Test to determine the relationship between Hunger Index and GDP per capita. A null hypothesis is established stating that the mean percent increase in Hunger score in 2000 is greater than or equal to that of 2022.

$$\alpha = 0.05\ (5\%)$$

$$H_0: \mu_{2000} \geq \mu_{2022}$$

$$H_a: \mu_{2000} < \mu_{2022}$$

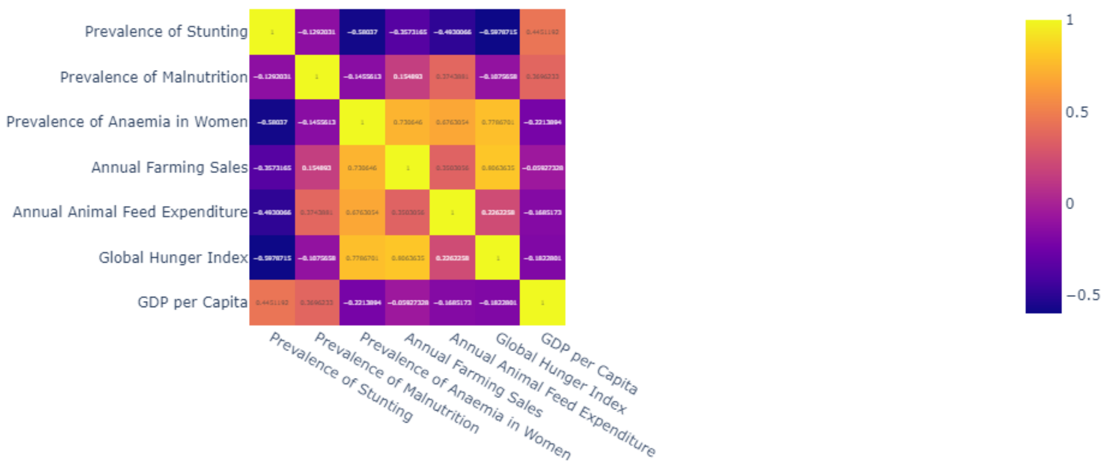
- With the help of a Two-Tailed T-Test, we calculate the P-value, which is a measure of the likelihood of obtaining a certain result by chance, given that the null hypothesis is true. Then, we compare the calculated P-value to the significance level (α) which is a pre-determined threshold (often 0.05) that determines whether a result is statistically significant.
- Based on the results of the hypothesis test, we draw a conclusion about the relationship between Hunger Index and GDP per capita in developing countries.
- Finally, we interpret that-
  - As the P-Value comes out to be **5.36e−07**, which is lesser than α, therefore, we reject the Null Hypothesis.
  - Therefore, by rejecting the null hypothesis, we can conclude that even though developing countries do not account for Hunger Index, but they still do some work to ensure food security and food production.

c) To illustrate the relationship between food deficiency diseases, and agricultural economy.

Steps-

- First, we define the "corrPearson()" function which takes in two lists of data, "x" and "y", and calculates the Pearson correlation coefficient between them.

- Then, we create a dictionary called "alldata" containing various data points.
- Then, we create an empty dictionary called "correlation" which will be used to store the correlation coefficients between each pair of variables in the "alldata" dictionary.
- After which, we loop through each variable in the "alldata" dictionary, using the "corrPearson()" function to calculate the correlation coefficient between it and every other variable.
- Then, it stores the calculated correlation coefficient in the "correlation" dictionary.
- Then, convert the "correlation" dictionary to a DataFrame and set the keys of the dictionary as the indexes of the DataFrame.
- After which, we create a variable "fig" which is a graph of the correlation matrix using the "px.imshow()" function from the Plotly library. We also set the attribute "text_auto" to True to automatically display the correlation coefficient values on the graph.



- From the Output, we interpret that-
    - The output of this code block would be a graph that shows the correlation matrix of the data in the "alldata" DataFrame. The graph would be a heatmap where each cell represents the correlation coefficient between the two variables represented by the rows and columns of the cell.
    - The colour of the cell represents the strength of the correlation, with darker colours indicating a stronger correlation. The correlation coefficients are displayed on each cell.
    - The diagonal of the matrix will be dark as it represents the correlation of the variable with itself which is always 1.

Conclusion

Upon accessing the data available for the prevalence of stunting and malnutrition, we found that the data is not actually normal. Therefore, after normalisation, we analyzed that the median rate of improvement in the prevalence of stunting is 0.020%, whereas that of in the prevalence of malnutrition is 0.017%. Also, this rate of improvement is mostly concentrated in Developing regions like Asia and Oceania, whereas underdeveloped regions like Africa are still lacking behind. As we know, this pace is not really perfect for reaching the target of Zero Hunger in 2030. After this, we accessed the correlation between the prevalence of stunting and malnutrition with the population of the world, which results into the conclusion that stunting is related with the population, but malnutrition is not. We also did the analysis of the future trend in the domains of stunting and malnutrition which gives us a positive response, i.e., the world is improving.

After analyzing the prevalence of anaemia in pregnant and non-pregnant women, we found that there is a strong correlation between the prevalence of anaemia in women on the basis of pregnancy, which implies that anaemia is not a problem related to pregnancy. Also, upon analyzing the food consumption per capita, we found that the food consumption is gradually increasing across all world regions, and transition countries are slowing joining the developed regions in this case. Also, the data on disability adjusted life years (DALYs) suggested that with the help of sufficient policies and implementation, the problem of disability raised due to deficiency of food is decreasing at a good pace.

With the help of Agricultural Sales and Animal Feed Expenditure Data Analysis, we found that the agricultural and food protection economy is not only dependent on these two disciplines but also conditions like economic crisis as, we passed upon the fact that countries like Nepal and Netherlands with a stable economy, are suffering from high expenditure and less income in the case of agricultural economy, and have a similar situation to countries like Greece and Sri Lanka who are under severe economic crisis at present. Also, upon ranking the countries on the basis of the two data available, we found that countries like Jordan and India are being profited in the case of agricultural economy, where other food-deficient countries in Africa and Asia are not able to compete and have to import food in order to eradicate Hunger.

For the analysis of Hunger Index with GDP per Capita for each country, we took the help of visualisation like line plot and correlation heatmap, which imply that even though the Hunger Score for each country is getting better but it is not as frequent as the growth of GDP in those same countries. This means that even though countries under development are earning great, but they are not really trying to reduce Hunger for fulfilling the UN SDG Goal 2.

References

Information References-

1. United Nations – *"Department of Economic and Social Affairs : Sustainable Development"* Website. Available at https://sdgs.un.org/ .
2. FAO – *"Food and Agriculture Organisation of the United Nations"* Website. Available at https://www.fao.org/ .
3. Wikipedia. Available at https://www.wikipedia.org/ .

Literature References-

1. Tyson Brown and Others (2022-05-19), *"Sustainable Development Goal: Zero Hunger"*, National Geographic Society. Available at https://education.nationalgeographic.org/resource/sustainable-development-goal-zero-hunger .

2. Farrell D. (2013), The role of poultry in human nutrition - In: "*Poultry Development Review*", FAO, Rome. Available at http://www.fao.org/docrep/019/i3531e/i3531e.pdf .

3. FAO, Literature Review and Gaps Analysis (February 2017) on *"Productivity and Efficiency Measurement in Agriculture"*, FAO, Rome. Available at https://www.fao.org/3/ca6428en/ca6428en.pdf .

4. Mogues T, Yu B, Fan S and McBride L (2012), "*The impacts of public investment in and for agriculture. Synthesis of the existing evidence."*, Agricultural Development Economics working paper No. 12-07, October, Agricultural Development Economics Division, FAO, Rome. Available at http://www.fao.org/docrep/016/ap108e/ap108e.pdf .
5. Derek Headey, Rebecca Heidkamp and Others (2022-08-22): "Impacts of COVID-19 on childhood malnutrition and nutrition-related mortality", The Lancet. Available at https://doi.org/10.1016/S0140-6736(20)31647-0 .
6. FAO, International Fund for Agricultural Development and WFP (2015), "*The State of Food Insecurity in the World. Meeting the 2015 International Hunger Targets: Taking Stock of Uneven Progress*", FAO, Rome. Available at http://www.fao.org/3/a-i4646e/index.html .

7. Daniel Lopez de Romana, Alison Greig, Andrew Thompson and Mandana Arabi (2021-03-21), *“Successful delivery of nutrition programs and the sustainable development goals”*, National Library of Medicine, United States. Available at https://pubmed.ncbi.nlm.nih.gov/33812278/ .
8. WHO (2017) Report on *“Anaemia in Children under 5 years”*. Available at https://www.who.int/data/gho/data/indicators/indicator-details/GHO/prevalence-of-anaemia-in-children-under-5-years-(-) .
9. HealthyEating, SFGate article on *“Daily recommended Caloric Intake for Women”*. Available at https://healthyeating.sfgate.com/daily-recommended-caloric-intake-women-6675.html .

Data References-

1. UNdata. Available at https://data.un.org/ .
2. FAO. Available at https://www.fao.org/ .
3. World Bank DataBank. Available at https://data.worldbank.org/ .
4. National Center for Biotechnology Information. Available at https://www.ncbi.nlm.nih.gov/ .
5. Global Hunger Index. Available at https://www.globalhungerindex.org/ .

Appendix

code.ipynb

Importing the Necessary Libraries

In [ ]:

```
import numpy as np
import pandas as pd
from scipy import stats
from sklearn.linear_model import LinearRegression

import geopandas as gpd
from geopandas import datasets
import matplotlib.pyplot as plt

import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

Objective 1 - Swastik Kulshreshtha (2022BTech105)

To analyze the prevalence of stunting and malnutrition (wasting and overweightness) in children.

- To access the prevalence of stunting in children across the world.
- To access the prevalence of malnutrition in children across the world.
- To correlate the prevalence of stunting and malnutrition along with the population growth across the world.
- To illustrate the world regions requiring more work on removing stunting & malnutrition.
- To estimate the future trend of prevalence of stunting and malnutrition in 2025 and 2030 for various regions of the world.

Reading the Dataset

In [ ]:

```
dataFrame = pd.read_csv("1 Data 1.csv")
dataPop = pd.read_csv("1 Data 2.csv")
```

Splitting the data based on Regions, Sub-regions and Countries

In [ ]:

```
# Specify the UN M49 Codes
regions, subregions = (1,2,5,9,10,21,142,150),
(11,13,14,15,17,18,19,29,30,34,35,39,53,54,57,61,143,145,151,154,155,199,202,419,432,72
2,830)
```

```
regionsIndex, subregionsIndex, countriesIndex = list(), list(),
list(range(len(dataFrame)))

for i in regions:
    x = dataFrame.index[dataFrame['GeoAreaCode'] == i].tolist()
    for j in x:
        regionsIndex.append(j)
for i in subregions:
    x = dataFrame.index[dataFrame['GeoAreaCode'] == i].tolist()
    for j in x:
        subregionsIndex.append(j)
for i in regionsIndex+subregionsIndex:
    countriesIndex.remove(i)

# Segregate the Data according to the Codes
regionsData = dataFrame.loc[regionsIndex]
subregionsData = dataFrame.loc[subregionsIndex]
countriesData = dataFrame.loc[countriesIndex]
```

Problem Statement 1

Segregating the Data

In [ ]:

```
# Specify the Data
stuntingData =
countriesData.loc[(countriesData['Indicator']=='2.2.1')&(countriesData['SeriesCode']=='
SH_STA_STNT')]
countries = np.unique(countriesData['GeoAreaName'].to_numpy())

# Convert the Data into Dictionary
stuntingRates = dict()
for i in countries:
    countryData = stuntingData.loc[stuntingData['GeoAreaName']==i]
    data = {k:v for k,v in countryData[['TimePeriod',
'Value']].to_dict('split')['data']}
    if data != {}:
        for j in data:
            try: data[j] = float(data[j])
            except: data[j] = 0
        rates = list(data.values())
        stuntingRates[i] = - ((rates[-1] / rates[0]) ** (1 / (len(rates) - 1)) - 1)
```

Calculating the Descriptive Statistical Information

In [ ]:

```
# Calculate the Statistical Information
dataValues = np.array(tuple(stuntingRates.values()), dtype=float)
table = [np.mean(dataValues), np.median(dataValues), np.min(dataValues),
np.max(dataValues)]
```

```
vals = {k:v for v,k in stuntingRates.items()}
table[2] = [table[2], vals[table[2]]]
table[3] = [table[3], vals[table[3]]]

# Print the Information
print(f'''The Statistical Info from the data available for the {len(vals)} out of
{len(countries)} countries:
\tThe Mean Rate of Improvement in Prevalence of Stunting is:\t{table[0]}
\tThe Median Rate of Improvement in Prevalence of Stunting is:\t{table[1]}
\tThe Minimum Rate of Improvement in Prevalence of Stunting is:\t{table[2][0]} which is
in {table[2][1]}
\tThe Maximum Rate of Improvement in Prevalence of Stunting is:\t{table[3][0]} which is
in {table[3][1]}''')
The Statistical Info from the data available for the 160 out of 162 countries:
        The Mean Rate of Improvement in Prevalence of Stunting is:
        0.02118318654660719
        The Median Rate of Improvement in Prevalence of Stunting is:
        0.02025723301898652
        The Minimum Rate of Improvement in Prevalence of Stunting is:-
0.03833327002221365 which is in Libya
        The Maximum Rate of Improvement in Prevalence of Stunting is:
        0.07696284742950776 which is in Tonga
```

Plotting the Graph

In [ ]:

```
# Configure the Data
plotDF = pd.DataFrame({'Country': list(stuntingRates.keys()), 'Rate':
list(stuntingRates.values())})

# Plot the Data
fig = px.box(x=dataValues, points='all')

# Customise the Data
fig.update_layout(title_text="Rate of Improvement in the Prevalence of Stunting",
xaxis_title="Rate of Improvement")

# Show the Data
fig.show()
```

In [ ]:

```
def kstest(data):
    data = sorted(data)
    cval = stats.norm.cdf(data)
    n = len(cval)
    Dplus = np.arange(1.0, n + 1) / n - cval
    Dplus = Dplus[Dplus.argmax()]
    Dminus = cval - np.arange(0.0, n)/n
    Dminus = Dminus[Dminus.argmax()]
    if Dplus > Dminus:
        D = Dplus
```

```
    else:
        D = Dminus
    probA = stats.ksone.sf(D, len(data)) * 2
    probB = np.clip(probA, 0, 1)
    return (D, probB)

kstest(list(stuntingRates.values()))
```

Out[ ]:

```
(0.48471098231507587, 3.887130866942939e-35)
```

Problem Statement 2

Segregating the Data

In [ ]:

```
# Specify the Data
countries = np.unique(countriesData['GeoAreaName'].to_numpy())
wasteData = countriesData.loc[countriesData['SeriesCode']=='SH_STA_WASTN']
ovrwtData = countriesData.loc[countriesData['SeriesCode']=='SN_STA_OVWGT']

# Convert the Data into Dictionaries
wasteRates, ovrwtRates = dict(), dict()
for i in countries:
    countryWasteData = wasteData.loc[wasteData['GeoAreaName']==i]
    countryOvrWtData = ovrwtData.loc[ovrwtData['GeoAreaName']==i]
    dataW = {k:v for k,v in countryWasteData[['TimePeriod',
'Value']].to_dict('split')['data']}
    dataO = {k:v for k,v in countryOvrWtData[['TimePeriod',
'Value']].to_dict('split')['data']}
    if dataW != {} and dataO != {}:
        for j in dataW:
            try: dataW[j] = float(dataW[j])
            except: dataW[j] = 0
        for j in dataO:
            try: dataO[j] = float(dataO[j])
            except: dataO[j] = 0
        wRates, oRates = list(dataW.values()), list(dataO.values())
        try:
            wasteRates[i] = - ((wRates[-1] / wRates[0]) ** (1 / (len(wRates) - 1)) - 1)
            ovrwtRates[i] = - ((oRates[-1] / oRates[0]) ** (1 / (len(oRates) - 1)) - 1)
        except ZeroDivisionError: pass
```

Taking out Mean of Wasting & Overweight Data to find out the Mean Increase/Decrease in Malnutrition

In [ ]:

```
meanMalnutRates = dict()
for i in wasteRates:
    meanMalnutRates[i] = np.mean((wasteRates[i], ovrwtRates[i]))
```

Calculating the Descriptive Statistical Information for the Rate of Improvement in Prevalence of Malnutrition

In [ ]:

```
# Calculate the Statistical Information
table = [np.mean(list(meanMalnutRates.values())),
np.median(list(meanMalnutRates.values())), np.min(list(meanMalnutRates.values())),
np.max(list(meanMalnutRates.values()))]
vals = {k:v for v,k in meanMalnutRates.items()}
table[2] = [table[2], vals[table[2]]]
table[3] = [table[3], vals[table[3]]]

# Print the Information
print(f'''The Statistical Info from the data available for the {len(vals)} out of
{len(countries)} countries:
\tThe Mean Rate of Improvement in Prevalence of Malnutrition is:\t{table[0]}
\tThe Median Rate of Improvement in Prevalence of Malnutrition is:\t{table[1]}
\tThe Minimum Rate of Improvement in Prevalence of Malnutrition is:\t{table[2][0]}
which is in {table[2][1]}
\tThe Maximum Rate of Improvement in Prevalence of Malnutrition is:\t{table[3][0]}
which is in {table[3][1]}''')
```

```
The Statistical Info from the data available for the 127 out of 162 countries:
        The Mean Rate of Improvement in Prevalence of Malnutrition is:
        0.003389371219568598
        The Median Rate of Improvement in Prevalence of Malnutrition is:
        0.017066560294318966
        The Minimum Rate of Improvement in Prevalence of Malnutrition is:    -
1.2184135430045917 which is in Papua New Guinea
        The Maximum Rate of Improvement in Prevalence of Malnutrition is:
        0.4237510657474796 which is in Tonga
```

Plotting the Graph

In [ ]:

```
# Plot the Data
fig = go.Figure()
fig.add_trace(go.Box(x=list(wasteRates.values()), name="Wasted Rate"))
fig.add_trace(go.Box(x=list(meanMalnutRates.values()), name="Mean Nutrition Rate"))
fig.add_trace(go.Box(x=list(ovrwtRates.values()), name="Overweight Rate"))

# Customise the Graph
fig.update_layout(title_text="Rate of Improvement in Prevalence of Malnutrition",
xaxis_title="Rates of Improvement", showlegend=False)

# # Show the Graph
fig.show()
```

In [ ]:

```
kstest(list(meanMalnutRates.values()))
```

Out[ ]:

```
(0.3681067758995739, 6.093422607807081e-16)
```

Problem Statement 3

Segregating the Data

In [ ]:

```
# Specify the Data
stntData = regionsData.loc[regionsData['SeriesCode']=='SH_STA_STNT']
wstData = regionsData.loc[regionsData['SeriesCode']=='SH_STA_WASTN']
ovwtData = regionsData.loc[regionsData['SeriesCode']=='SN_STA_OVWGT']
popData = dataPop.loc[dataPop['Year'].isin(range(2000, 2021))]['Total Population, as of 
1 July (thousands)'].to_list()
regions = np.unique(regionsData['GeoAreaName'].to_numpy())

# Convert the Data into Dictionary
regData = dict()
for i in regions:
    myData1 = stntData.loc[regionsData['GeoAreaName']==i]
    myData1 = {k:float(v) for k,v in myData1[['TimePeriod', 
'Value']].to_dict('split')['data']}
    myData2 = wstData.loc[regionsData['GeoAreaName']==i]
    myData2 = {k:float(v) for k,v in myData2[['TimePeriod', 
'Value']].to_dict('split')['data']}
    myData3 = ovwtData.loc[regionsData['GeoAreaName']==i]
    myData3 = {k:float(v) for k,v in myData3[['TimePeriod', 
'Value']].to_dict('split')['data']}
    regData[i] = {"Stunting": myData1, "Wasted": myData2, "Overweight": myData3}
```

Making the Correlation Table

In [ ]:

```
corrDF = pd.DataFrame({
    'Stunting': regData['World']['Stunting'].values(),
    'Overweight': regData['World']['Overweight'].values(),
    'Population': popData
    })
corrDF2 = pd.DataFrame({
    'stnt - mean(stnt)': (corrDF['Stunting'] - 
pd.Series(list(np.mean(corrDF['Stunting']) for i in range(21)))),
    'owt - mean(owt)': (corrDF['Overweight'] - 
pd.Series(list(np.mean(corrDF['Overweight']) for i in range(21)))),
    'pop - mean(pop)': (corrDF['Population'] - 
pd.Series(list(np.mean(corrDF['Population']) for i in range(21))))
})
corrDF2['sqr(stnt - mean(stnt))'] = corrDF2['stnt - mean(stnt)'] * corrDF2['stnt - 
mean(stnt)']
corrDF2['sqr(owt - mean(owt))'] = corrDF2['owt - mean(owt)'] * corrDF2['owt - 
mean(owt)']
corrDF2['sqr(pop - mean(pop))'] = corrDF2['pop - mean(pop)'] * corrDF2['pop - 
mean(pop)']
```

```
stntXowt, stntXpop, owtXpop = list(), list(), list()
for i in range(len(corrDF2)):
    stntXowt.append(corrDF2['stnt - mean(stnt)'][i] * corrDF2['owt - mean(owt)'][i])
    stntXpop.append(corrDF2['stnt - mean(stnt)'][i] * corrDF2['pop - mean(pop)'][i])
    owtXpop.append(corrDF2['owt - mean(owt)'][i] * corrDF2['pop - mean(pop)'][i])
corrDF2['(stnt - mean(stnt)) X (owt - mean(owt))'] = stntXowt
corrDF2['(stnt - mean(stnt)) X (pop - mean(pop))'] = stntXpop
corrDF2['(owt - mean(owt)) X (pop - mean(pop))'] = owtXpop

corrStntOwt = corrDF2['(stnt - mean(stnt)) X (owt - mean(owt))'].sum() /
np.sqrt(corrDF2['sqr(stnt - mean(stnt))'].sum() * corrDF2['sqr(owt -
mean(owt))'].sum())
corrStntPop = corrDF2['(stnt - mean(stnt)) X (pop - mean(pop))'].sum() /
np.sqrt(corrDF2['sqr(stnt - mean(stnt))'].sum() * corrDF2['sqr(pop -
mean(pop))'].sum())
corrOwtPop = corrDF2['(owt - mean(owt)) X (pop - mean(pop))'].sum() /
np.sqrt(corrDF2['sqr(owt - mean(owt))'].sum() * corrDF2['sqr(pop - mean(pop))'].sum())
```

In [ ]:

```
corrDF3 = pd.DataFrame({
        'Stunting': [1, corrStntOwt, corrStntPop],
        'Overweight': [corrStntOwt, 1, corrOwtPop],
        'Population': [corrStntPop, corrOwtPop, 1]
    },
    index=['Stunting', 'Overweight', 'Population']
)
corrDF3
```

Out[ ]:

	Stunting	**Overweight**	**Population**
Stunting	1.000000	-0.333233	-0.998465
Overweight	-0.333233	1.000000	0.359125
Population	-0.998465	0.359125	1.000000

In [ ]:

```
fig = px.imshow(corrDF3, text_auto=True)
fig.update_layout(width=500, height=500)
fig.show()
```

Problem Statement 4

Calculating the Compound Annual Growth Rate (CAGR) from the Data

In [ ]:

```
for i in regData:
    if i != 'World':
        stntRates = list(regData[i]['Stunting'].values())
```

```
        ovwtRates = list(regData[i]['Overweight'].values())
        try:
            gr_Stunting = ((stntRates[-1] / stntRates[0]) ** (1 / (len(stntRates) - 1))
- 1)
            gr_Overweight = ((ovwtRates[-1] / ovwtRates[0]) ** (1 / (len(ovwtRates) -
1)) - 1)
            print(f"For the region {i},\nRate of Improvement in Stunting:
{gr_Stunting}\nRate of Improvement in Overweight: {gr_Overweight}\n")
        except: pass
```

```
For the region Africa,
Rate of Improvement in Stunting: -0.014958531436904954
Rate of Improvement in Overweight: -0.007811454346558944

For the region Asia,
Rate of Improvement in Stunting: -0.026103649320611266
Rate of Improvement in Overweight: 0.0072552541834860484

For the region Europe,
Rate of Improvement in Stunting: -0.018967423584430376
Rate of Improvement in Overweight: 0.0012203166373554453

For the region Northern America,
Rate of Improvement in Stunting: 0.0
Rate of Improvement in Overweight: 0.011080947166022037

For the region Oceania,
Rate of Improvement in Stunting: 0.008511260352928263
Rate of Improvement in Overweight: 0.034018707807229376

For the region South America,
Rate of Improvement in Stunting: -0.02644821850482726
Rate of Improvement in Overweight: 0.007942576492167008
```

Making the Choropleth Maps for-

Requirement of Improvement in Stunting

In [ ]:

```
# Plot the Data
worldMap = gpd.read_file(datasets.get_path('naturalearth_lowres'))
world = worldMap.plot()
worldMap[(worldMap.continent=='Antarctica')].plot(color='#FFFFFF', ax=world)
worldMap[(worldMap.continent=='Africa')].plot(color='#2EB62C', ax=world)
worldMap[(worldMap.continent=='Asia')|(worldMap.name=='Russia')].plot(color='#83D475',
ax=world)
worldMap[(worldMap.continent=='Europe')&(worldMap.name!='Russia')].plot(color='#57C84D'
, ax=world)
worldMap[(worldMap.continent=='North America')].plot(color='#E4FADC', ax=world)
worldMap[(worldMap.continent=='South America')].plot(color='#ABE098', ax=world)
worldMap[(worldMap.continent=='Oceania')].plot(color='#E4FADC', ax=world)
```

```
# Customise the Graph
plt.title("Requirement of Improvement in Stunting across Continents")
world.set_axis_off()

# Show the Graph
plt.show()
```

Requirement of Improvement in Overweight Controlling

In [ ]:

```
# Plot the Data
worldMap = gpd.read_file(datasets.get_path('naturalearth_lowres'))
world = worldMap.plot()
worldMap[(worldMap.continent=='Antarctica')].plot(color='#FFFFFF', ax=world)
worldMap[(worldMap.continent=='Africa')].plot(color='#CCF9FF', ax=world)
worldMap[(worldMap.continent=='Asia')|(worldMap.name=='Russia')].plot(color='#55D0FF',
ax=world)
worldMap[(worldMap.continent=='Europe')&(worldMap.name!='Russia')].plot(color='#7CE8FF'
, ax=world)
worldMap[(worldMap.continent=='North America')].plot(color='#0080BF', ax=world)
worldMap[(worldMap.continent=='South America')].plot(color='#00ACDF', ax=world)
worldMap[(worldMap.continent=='Oceania')].plot(color='#E1F7FA', ax=world)

# Customise the Graph
plt.title("Requirement of Improvement in Overweight across Continents")
world.set_axis_off()

# Show the Graph
plt.show()
```

Problem Statement 5

Predicting Future Prevalence of Stunting and Malnutrition

In [ ]:

```
# Function for Estimation
def lrEstimate(x,y):
    n = len(x)
    df = pd.DataFrame({'x': x, 'y': y})
    df['x2'] = df['x'] * df['x']
    df['y2'] = df['y'] * df['y']
    df['xy'] = df['x'] * df['y']
    slope = ((n*(df['xy'].sum())) - ((df['x'].sum()) * (df['y'].sum()))) /
((n*(df['x2'].sum()) - ((df['x'].sum())**2)))
    intercept = (((df['y'].sum()) * (df['x2'].sum())) - ((df['x'].sum()) *
(df['xy'].sum()))) / ((n*(df['x2'].sum()) - ((df['x'].sum())**2)))
```

```
    return lambda x : slope * x + intercept


for i in regData:
    x = list(regData[i]['Stunting'].keys())
    y1 = list(regData[i]['Stunting'].values())
    y2 = list(regData[i]['Overweight'].values())
    if [] not in [x, y1, y2]:
        estimator = lrEstimate(x, y1)
        mymodel = list(map(estimator, x))
        data1 = list(int(estimator(j)) for j in (2025, 2030))
        print(f'''{i}:\n\tFor 2025, Estimated Prevalence of Stunting is {data1[-2]}%
        For 2030, Estimated Prevalence of Stunting is {data1[-1]}%''')

        estimator = lrEstimate(x, y2)
        mymodel = list(map(estimator, x))
        data2 = list(int(estimator(j)) for j in (2025, 2030))
        print(f'''\tFor 2025, Estimated Prevalence of Overweight is {data2[-2]}%
        For 2030, Estimated Prevalence of Overweight is {data2[-1]}%\n''')

        fig = px.scatter(pd.DataFrame({"Year": x+[2025,2030], "Stunting": y1+data1,
"Overweight": y2+data2}),
        x="Year", y=["Stunting", "Overweight"], trendline='ols',
trendline_color_override = 'blue',
        color_discrete_sequence=["green", "red"], title=f"Prevalence of Stunting and
Overweight in {i}"
        )
        fig.update_layout(
            legend_title="Prevalence of",
            yaxis_title="Percentage (%)",
            legend=dict(orientation="h", x=1, y=1.02, xanchor="right",
yanchor="bottom")
            )
        fig.show()
Africa:
        For 2025, Estimated Prevalence of Stunting is 27%
        For 2030, Estimated Prevalence of Stunting is 24%
        For 2025, Estimated Prevalence of Overweight is 4%
        For 2030, Estimated Prevalence of Overweight is 3%

Asia:
        For 2025, Estimated Prevalence of Stunting is 17%
        For 2030, Estimated Prevalence of Stunting is 13%
        For 2025, Estimated Prevalence of Overweight is 5%
        For 2030, Estimated Prevalence of Overweight is 5%

Europe:
        For 2025, Estimated Prevalence of Stunting is 4%
        For 2030, Estimated Prevalence of Stunting is 3%
        For 2025, Estimated Prevalence of Overweight is 8%
        For 2030, Estimated Prevalence of Overweight is 8%
```

```
Northern America:
        For 2025, Estimated Prevalence of Stunting is 2%
         For 2030, Estimated Prevalence of Stunting is 3%
        For 2025, Estimated Prevalence of Overweight is 9%
         For 2030, Estimated Prevalence of Overweight is 10%

Oceania:
        For 2025, Estimated Prevalence of Stunting is 21%
         For 2030, Estimated Prevalence of Stunting is 22%
        For 2025, Estimated Prevalence of Overweight is 14%
         For 2030, Estimated Prevalence of Overweight is 16%

South America:
        For 2025, Estimated Prevalence of Stunting is 6%
         For 2030, Estimated Prevalence of Stunting is 4%
        For 2025, Estimated Prevalence of Overweight is 8%
         For 2030, Estimated Prevalence of Overweight is 8%

World:
        For 2025, Estimated Prevalence of Stunting is 18%
         For 2030, Estimated Prevalence of Stunting is 15%
        For 2025, Estimated Prevalence of Overweight is 5%
         For 2030, Estimated Prevalence of Overweight is 5%
```

Objective 2 - Sheetal Kanwar (2022BTech095)

To analyze the prevalence of anaemia in women, on the basis of pregnancy. Additionally, check the regions with lowest per capita food consumption and the people with food deficiency disabilities.

- To access the prevalence of anaemia in pregnant and non-pregnant women across the world.
- To check the world regions which have the lowest per capita food consumption in the year 2030.
- To access the difference in disability-adjusted life years (SEV) and annual % change (EAPC) between females and males in different regions and sub-regions across the world in 1990 and 2019.

Reading the Datasets

In [ ]:

```
dataPregnant = pd.read_csv("2 Data 1.csv")
dataNonPregnant = pd.read_csv("2 Data 2.csv")
dataFood = pd.read_csv("2 Data 3.csv")
dataSEV = pd.read_csv("2 Data 4.csv")
```

Problem Statement 1

In [ ]:

```
years = list(map(int, dataPregnant.to_dict('split')['columns'][1:][::-1]))
```

```
regionsPreg = np.unique(dataPregnant['Location'].to_numpy())

fig = go.Figure()
for i in regionsPreg:
    data =
dataPregnant.loc[dataPregnant['Location']==i].to_dict('split')['data'][0][1:][::-1]
    fig.add_trace(go.Scatter(x=years, y=data, name=i, mode='markers'))

fig.update_layout(
    title_text='Prevalence of Anaemia in Pregnant Women (2010-2019) by region',
    xaxis_title='Year', yaxis_title='Prevalence of Anaemia in Pregnant Women (%)',
legend_title='Regions'
    )
fig.show()
```

In [ ]:

```
def PearsonCorrelation(x, y):
    df = pd.DataFrame({'x': x, 'y': y})
    df['x - u(x)'] = (df['x'] - pd.Series(list(np.mean(df['x']) for i in
range(len(x)))))
    df['y - u(y)'] = (df['y'] - pd.Series(list(np.mean(df['y']) for i in
range(len(y)))))
    df['(x - u(x))^2'] = df['x - u(x)'] ** 2
    df['(y - u(y))^2'] = df['y - u(y)'] ** 2

    xXy = []
    for i in range(len(df)):
        a = df['x - u(x)'][i] * df['y - u(y)'][i]
        xXy.append(a)
    df['(x - u(x)) X (y - u(y))'] = xXy
    corrN = sum(df['(x - u(x)) X (y - u(y))'])
    corrD = (sum(df['(x - u(x))^2']) * sum(df['(y - u(y))^2'])) ** 0.5
    corr = round(corrN / corrD, 5)
    return corr
```

In [ ]:

```
# Prevalence of anaemia in pregnant women (aged 15-49) in 2019
pregnant_2019 = dataPregnant['2019']
# Prevalence of anaemia in non-pregnant women (aged 15-49) in 2019
non_pregnant_2019 = dataNonPregnant['2019']

# Calculate the correlation between the prevalence of anaemia in pregnant and non-
pregnant women in 2019
correlation= PearsonCorrelation(pregnant_2019, non_pregnant_2019)
print(f"Correlation between pregnant and non-pregnant women in 2019: {correlation}")

# Prevalence of anaemia in pregnant women (aged 15-49) in 2009
pregnant_2009 = dataPregnant['2009']
# Prevalence of anaemia in non-pregnant women (aged 15-49) in 2009
non_pregnant_2009 = dataNonPregnant['2009']
```

```
#Calculate the correlation between the prevelence of anaemia in pregnant and non-
pregnant women in 2009
correlation= PearsonCorrelation(pregnant_2009, non_pregnant_2009)
print(f"Correlation between pregnant and non-pregnant women in 2009: {correlation}")
print("\nThis indicates a strong correlation between the two variables, which means
that there is a strong association between the prevalence of anaemia in pregnant and
non-pregnant women in 2019.")
Correlation between pregnant and non-pregnant women in 2019: 0.98653
Correlation between pregnant and non-pregnant women in 2009: 0.98414

This indicates a strong correlation between the two variables, which means that there
is a strong association between the prevalence of anaemia in pregnant and non-pregnant
women in 2019.
```

Problem Statement 2

In [ ]:

```
regions = dataFood['Region']
years = [1964, 1974, 1984, 1997, 2015, 2030]

fig = go.Figure()
for i in regions:
    data = dataFood.loc[dataFood['Region']==i].to_dict('split')['data'][0][1:]
    fig.add_trace(go.Scatter(x=years, y=data, name=i))

fig.update_layout(
    title_text='Global & Region per capita Food Consumption (kcal per capita per day)',
    xaxis_title='Year', yaxis_title='per Capita Food Consumption',
legend_title='Regions'
    )
fig.show()
```

In [ ]:

```
# Find the regions with the lowest numbers in the latest year (2030)
dataFood.sort_values("2030").head(2)
```

Out[ ]:

	Region	1964	1974	1984	1997	2015	2030
3	Sub-Saharan Africa	2058	2079	2057	2195	2360	2540
6	South Asia	2017	1986	2205	2403	2700	2900

In [ ]:

```
# Define the Function
def ttest(x, popmean):
    n = len(x)
    var_x = np.var(x, ddof = 1)
    std = np.sqrt((var_x) / 2)
    tval = (np.mean(x) - popmean) / (std * np.sqrt(2 / n))
    dof = 2 * n - 2
```

```
    pval = 1 - stats.t.cdf(-tval, df = dof)*2
    return tval, pval

# Define the threshold or benchmark value
threshold = 3000

# Perform the t-test
t, p = ttest(dataFood['2030'], threshold)
# Check the p-value
if p < 0.05:
    print("Reject the null hypothesis, the mean of the food deficiency diseases for the region of interest is not equal to the threshold.")
else:
    print("Fail to reject the null hypothesis, the mean of the food deficiency diseases for the region of interest is equal to the threshold.")
```

```
Fail to reject the null hypothesis, the mean of the food deficiency diseases for the region of interest is equal to the threshold.
```

Problem Statement 3

In [ ]:

```
regions = dataSEV['region']
sev_1990 = dataSEV['Age standardized SEV in 1990 for both sexes']
sev_2019 = dataSEV['Age standardized SEV in 2019 for both sexes']

fig = go.Figure()
fig.add_trace(go.Bar(x=regions, y=sev_1990, name='1990'))
fig.add_trace(go.Bar(x=regions, y=sev_2019, name='2019'))

fig.update_layout(
    xaxis_title='Region', yaxis_title='Age-standardized SEV', legend_title='Year',
    legend=dict(orientation="h", x=1, y=1.02, xanchor="right", yanchor="bottom"),
    title_text='Age-standardized SEV for both Sexes'
    )
fig.show()
```

In [ ]:

```
females_1990 = dataSEV['Age standardized SEV in 1990 for females']
females_2019 = dataSEV['Age standardized SEV in 2019 for  females']
males_1990 = dataSEV['Age standardized SEV in 1990 for males']
males_2019 = dataSEV['Age standardized SEV in 2019 for males']

# Calculate the difference between SEV for females and males in 1990 and 2019
diff_1990 = females_1990 - males_1990
diff_2019 = females_2019 - males_2019

# Print the difference between SEV for females and males in 1990 and 2019
print("Difference in SEV between females and males in 1990:", diff_1990)
print("Difference in SEV between females and males in 2019:", diff_2019)
```

```
Difference in SEV between females and males in 1990: 0      8.8
1      8.8
```

```
2      5.8
3     10.6
4     11.5
5     11.2
6     11.6
7     14.8
8     11.3
9      7.3
10    11.1
11     7.1
12     9.0
13     3.8
14     5.5
15     8.9
16     4.4
17     5.1
18     7.5
19    11.1
20    10.8
21     6.7
22    10.2
23    11.0
24    10.4
25     6.7
26    16.7
dtype: float64
Difference in SEV between females and males in 2019: 0      9.4
1      9.0
2      6.5
3     10.2
4     11.2
5     10.5
6     10.5
7     14.8
8     10.9
9      6.9
10    10.3
11     6.6
12     8.4
13     3.7
14     5.3
15     7.6
16     6.6
17     4.8
18     7.0
19    10.9
20    10.3
21     6.1
22     9.9
23    11.5
24    10.5
```

```
25     7.1
26    16.0
dtype: float64
```

Objective 3 - Puneet (2022BTech077)

To rank world regions based on farming and pastoral economy of the countries.

- To analyze the rate of farming sales and animal feed expenditure across the world.
- To rank the world regions based on the analysis before.
- To correlate the rates of farming sales and animal feed expenditure.

Reading the Datasets

In [ ]:

```
dataFarming = pd.read_csv("3 Data 1.csv")
dataAnimals = pd.read_csv("3 Data 2.csv")
```

Splitting the Dataset

In [ ]:

```
# Specify the Data
countriesInFarmingData = np.unique(dataFarming['Country or Area'].to_numpy())
countriesInAnimalData = np.unique(dataAnimals['Country or Area'].to_numpy())

# Convert the Dataset to Dictionary
allData = dict()
for i in countriesInAnimalData:       # Lesser Size
  fData = dataFarming.loc[dataFarming['Country or Area']==i]
  aData = dataAnimals.loc[(dataAnimals['Country or
Area']==i)&(dataAnimals['Unit']=='Mil. USD')]

  y1 = np.unique(fData['Year'].to_numpy())
  y2 = np.unique(aData['Year'].to_numpy())

  if len(y1) != 0 and len(y2) != 0:
    fCountryData = fData['Value'].to_list()
    aCountryData = aData['Value'].to_list()
    allData[i] = {
      'Farming': {k:v for k,v in zip(y1, fCountryData)},
      'Animals': {k:v for k,v in zip(y2, aCountryData)}
      }
```

Problem Statement 1

Calculating the Growth Rates

In [ ]:

```
# Define the Growth Rate Function
def growthRate(dataset):
  gr = -((dataset[-1] / dataset[0]) ** (1 / (len(dataset) - 1)) - 1)
  return gr

# Calculate and Print the Growth Rates
print("The Growth Rate in the sector of Farming Sales and Animal Feed Spendings in-")
growthRatesData = dict()
for i in allData:
  fData = allData[i]['Farming']
  aData = allData[i]['Animals']
  try:
    fGrowth = growthRate(list(fData.values()))
    aGrowth = growthRate(list(aData.values()))
    growthRatesData[i] = {
      'Farming Sales':fGrowth,
      'Animal Feed Spendings':aGrowth
      }
    print(f"\t{i} are {fGrowth} and {aGrowth}, respectively.")
  except:
    pass
```

```
The Growth Rate in the sector of Farming Sales and Animal Feed Spendings in-
        Albania are 0.046539054149960735 and 0.03680742214566357, respectively.
        Brazil are 0.0663700492936562 and 0.0846220713615713, respectively.
        Bulgaria are 0.015375657351417216 and 0.06650091789326595, respectively.
        Canada are 0.031505580629358265 and 0.04527139056921026, respectively.
        Chile are 0.06186134096717577 and -0.06430100470212419, respectively.
        China, Hong Kong Special Administrative Region are 0.04981374021152096 and
0.07807033065752833, respectively.
        Czechia are 0.017435605870597337 and 0.03952857906914142, respectively.
        Denmark are -0.004091227143505183 and 0.07341455906226202, respectively.
        Dominican Republic are 0.09783166675290811 and 0.12771195721597894,
respectively.
        Ecuador are 0.030018065324903143 and 0.13203975608901397, respectively.
        Estonia are 0.02969161354511718 and 0.06200677576657743, respectively.
        Finland are 0.008895535569544255 and 0.013529363276342798, respectively.
        France are 0.004446343482097825 and 0.09967827004325724, respectively.
        Georgia are 0.060961618377351035 and 0.11524674682605007, respectively.
        Germany are 0.013314737224209505 and 0.02900690647831705, respectively.
        Greece are -0.005801552245092267 and -0.26870440620516023, respectively.
        Hungary are 0.026273188202625364 and 0.05278947336024775, respectively.
        Iceland are 0.024438438298020282 and 0.04954442894933653, respectively.
        India are 0.08918221717632202 and 0.2868591441152062, respectively.
        Ireland are 0.010607752666591419 and 0.1849212128801756, respectively.
        Italy are 0.004507483292505832 and 0.02191232802145604, respectively.
        Jordan are 0.11912842079600283 and 0.16332311302469082, respectively.
        Kenya are 0.050117945401395025 and 0.10000243141096521, respectively.
        Latvia are 0.027156679411287854 and 0.03815402984513161, respectively.
        Lithuania are -0.01990232048802132 and 0.07016991924483085, respectively.
        Mexico are 0.08278736270768106 and 0.05344418028703346, respectively.
        Mongolia are 0.08717825677121271 and 0.1280016293419095, respectively.
```

```
        Mozambique are 0.04340239373575017 and 0.03915842308275086, respectively.
        Nepal are 0.06733305839039427 and 0.7619637967381618, respectively.
        Netherlands are 0.009311943039991388 and 0.3589971869478529, respectively.
        Nigeria are 0.17692093275253074 and -0.04241816255316344, respectively.
        Norway are 0.022853685220247 and 0.08840603691360027, respectively.
        Poland are 0.026286551241449296 and 0.12884495751478553, respectively.
        Portugal are 0.010565037847231173 and 0.0046775373292349265, respectively.
        Romania are 0.019159460428051944 and 0.09848487851338394, respectively.
        Slovakia are 0.03049416315443021 and -0.029041186200817082, respectively.
        Slovenia are 0.017987429182188097 and 0.06016416703875116, respectively.
        Spain are 0.0077050391864089685 and 0.015749665999867934, respectively.
        Sri Lanka are 0.08874455819916482 and 0.5737486754489549, respectively.
        State of Palestine are 0.05501400492516473 and 0.08894862814493865,
respectively.
        Sweden are 0.011916575852935707 and 0.009218689988038764, respectively.
        United Kingdom are 0.01780346864207094 and 0.10203086856568155, respectively.
        Uruguay are 0.09171142200733873 and -0.02005579517235656, respectively.
```

Converting the Data to List

In [ ]:

```
cData = list()
fData = list()
aData = list()
for i in growthRatesData:
  cData.append(i)
  fData.append(growthRatesData[i]['Farming Sales'])
  aData.append(growthRatesData[i]['Animal Feed Spendings'])
```

Other Statistical Information regarding Total Farming Sales

In [ ]:

```
# Calculate the Statistical Information
fMean = np.mean(fData)
fMedian = np.median(fData)
fMode = stats.mode(fData, keepdims=False)
fStd = np.std(fData)
fVar = np.var(fData)

# Print the Information
print("The Mean of Total Farming Sales is", fMean)
print("The Median of Total Farming Sales is", fMedian)
print("The Mode of Total Farming Sales is", fMode)
print("The Standard Deviation of Total Farming Sales is", fStd)
print("The Variance of Total Farming Sales is", fVar)
The Mean of Total Farming Sales is 0.040066348311678865
The Median of Total Farming Sales is 0.027156679411287854
The Mode of Total Farming Sales is ModeResult(mode=-0.01990232048802132, count=1)
The Standard Deviation of Total Farming Sales is 0.03828504206186734
The Variance of Total Farming Sales is 0.0014657444456789518
```

Other Statistical Information regarding Total Animal Feeding Expenditure

In [ ]:

```
# Calculate the Statistical Information
aMean = np.mean(aData)
aMedian = np.median(aData)
aMode = stats.mode(aData, keepdims=False)
aStd = np.std(aData)
aVar = np.var(aData)

# Print the Information
print("The Mean of Total Animal Feeding Expenditure is", aMean)
print("The Median of Total Animal Feeding Expenditure is", aMedian)
print("The Mode of Total Animal Feeding Expenditure is", aMode)
print("The Standard Deviation of Total Animal Feeding Expenditure is", aStd)
print("The Variance of Total Animal Feeding Expenditure is", aVar)
```

```
The Mean of Total Animal Feeding Expenditure is 0.09438209056575687
The Median of Total Animal Feeding Expenditure is 0.06650091789326595
The Mode of Total Animal Feeding Expenditure is ModeResult(mode=-0.26870440620516023,
count=1)
The Standard Deviation of Total Animal Feeding Expenditure is 0.15754712196007697
The Variance of Total Animal Feeding Expenditure is 0.024821095637903367
```

Problem Statement 2

Ranking the Countries

In [ ]:

```
# Specify the Data
dat = list(zip(fData, aData))

# Calculate the Rank from Eigenvalue
dataarray = list()
for x in dat:
    for y in dat:
        if (x[0] == y[0]) and (x[1] == y[1]):
            dataarray.append(0)
        elif (x[0] <= y[0]) and (x[1] <= y[1]):
            dataarray.append(1)
        elif (x[0] <= y[0]) and (x[1] >= y[1]):
            dataarray.append(2)
        elif (x[0] >= y[0]) and (x[1] <= y[1]):
            dataarray.append(3)
        elif (x[0] >= y[0]) and (x[1] >= y[1]):
            dataarray.append(4)
matrix = np.array(dataarray).reshape(43,43)
eigVec = np.array([[1] for _ in range(43)])
for _ in range(7):
    eigVec = np.matmul(matrix, eigVec)
    eigVal = np.max(eigVec)
```

```
    eigVec = eigVec / eigVal

dat2 = list(zip(cData, [i[0] for i in eigVec]))
dat3 = sorted(dat2, key=lambda x: x[1], reverse=True)

# Print the Ranks
for i in range(len(dat3)):
    print(i+1, dat3[i][0])
1 Jordan
2 India
3 Sri Lanka
4 Dominican Republic
5 Nepal
6 Mongolia
7 Nigeria
8 Georgia
9 Brazil
10 Uruguay
11 Mexico
12 Kenya
13 State of Palestine
14 Ecuador
15 China, Hong Kong Special Administrative Region
16 Poland
17 Chile
18 Albania
19 Mozambique
20 Canada
21 Estonia
22 Ireland
23 Netherlands
24 Romania
25 Norway
26 United Kingdom
27 Hungary
28 Latvia
29 Slovakia
30 Iceland
31 Slovenia
32 Bulgaria
33 Czechia
34 France
35 Germany
36 Denmark
37 Sweden
38 Lithuania
39 Portugal
40 Finland
41 Spain
42 Italy
43 Greece
```

Problem Statement 3

Plotting a Bar Graph

In [ ]:

```
# Plot the Data
fig = go.Figure()
fig.add_trace(go.Bar(x=cData, y=fData, name="Farming Sales", marker_color='green'))
fig.add_trace(go.Bar(x=cData, y=aData, name="Animal Feed Expenditure", 
marker_color='yellow'))

# Customise the Graph
fig.update_layout(
    width=1250, height=1000,
    barmode='stack',
    title_text="Growth Rates in Farming Sales and Animal Feeding Expenditure by 
Country",
    xaxis_title="Countries",
    yaxis_title="Growth Rate"
)

# # Show the Graph
fig.show()
```

Making a Normal Distribution Curve

In [ ]:

```
x = np.linspace(-0.8, 0.8, 1000)
y1 = stats.norm.pdf(x, fMean, fStd)
y2 = stats.norm.pdf(x, aMean, aStd)
```

Making the Graph

In [ ]:

```
# Plot the Data
fig = go.Figure()
fig.add_trace(go.Scatter(x=x, y=y1, name="Farming Sales", mode='lines'))
fig.add_trace(go.Scatter(x=x, y=y2, name="Animal Feeding Expenditure", mode='lines'))

# Customise the Graph
fig.update_layout(
    title_text="Normal Distribution Curve for Growth Rate in Farming Sales and Animal 
Feeding Expenditure",
    xaxis_title="Percentage (%)",
    yaxis_title="Probability Density",
    showlegend=True)

# # Show the Graph
```

```
fig.show()
```

Objective 4 - Riya Gupta (2022BTech086)

To analyze countries on the basis of global hunger index and GDP per capita.

- To comprehend the rate of improvement in hunger index and GDP across various regions.
- To check whether *"Hunger Index is a crucial indicator for food security and one that historically has been neglected in the context of developing countries."*
- To illustrate the relationship between food deficiency diseases, and agricultural economy.

Reading the Datasets

In [ ]:

```
dataGHI = pd.read_csv("4 Data 1.csv")
dataGDP = pd.read_csv("4 Data 2.csv")
```

Handling the Null Values

In [ ]:

```
dataGHI = dataGHI.fillna(0)
dataGDP = dataGDP.fillna(0)
```

Problem Statement 1

Calculating the World Data on the basis of All Countries

In [ ]:

```
# Function for Handling Non-Float Values
def converter(array):
    narray = list()
    for i in range(len(array)):
        fNum = 2.5
        try:
            fNum = float(array[i])
        except:
            pass
        narray.append(fNum)
    return narray

ghi2000 = converter(dataGHI['2000'].to_list())
ghi2007 = converter(dataGHI['2007'].to_list())
ghi2014 = converter(dataGHI['2014'].to_list())
ghi2022 = converter(dataGHI['2022'].to_list())

# Configure the Data
yearsGHI = ['2000', '2007', '2014', '2022']
```

```
valueGHI = [np.mean(ghi2000), np.mean(ghi2007), np.mean(ghi2014), np.mean(ghi2022)]

gdpData = dataGDP.loc[dataGDP['Country Name']=='World'].to_dict('split')
yearsGDP = gdpData['columns'][4:]
valueGDP = gdpData['data'][0][4:]
```

Making the Graph

In [ ]:

```
# Plot the Data
fig = make_subplots(rows=1, cols=2, subplot_titles=("Global Hunger Index", "GDP per
Capita"))
fig.add_trace(go.Scatter(
    x=yearsGHI,
    y=valueGHI,
    mode='lines+markers'),
    row=1, col=1
    )
fig.add_trace(go.Scatter(
    x=yearsGDP,
    y=valueGDP,
    mode='lines+markers'),
    row=1, col=2
    )

# Customise the Graph
fig.update_xaxes(title_text="Years", row=1, col=1)
fig.update_xaxes(title_text="Years", row=1, col=2)
fig.update_yaxes(title_text="Hunger Score", row=1, col=1)
fig.update_yaxes(title_text="GDP (in million $)", row=1, col=2)
fig.update_layout(showlegend=False)

# Show the Graph
fig.show()
```

Problem Statement 2

State the Hypothesis

$$\alpha = 0.05(5\%)$$

$$H_0: \mu_{2000} \geq \mu_{2022}$$

$$H_a: \mu_{2000} < \mu_{2022}$$

In [ ]:

```
def ttest(data1, data2, tail=2):
    mean1 = np.mean(data1)
    mean2 = np.mean(data2)
    std1 = np.std(data1)
```

```
    std2 = np.std(data2)
    n1 = len(data1)
    n2 = len(data2)
    tN = mean1 - mean2
    tD = np.sqrt(((std1**2)/n1) + ((std2**2)/n2))
    t = tN / tD
    if tail == 2:
        p = stats.t.sf(abs(t), df=n1-1)*2
    elif tail in (-1,1):
        p = stats.t.sf(abs(tail*t), df=n1-1)
    else:
        return None
    return (t, p)

x = ttest(ghi2000, ghi2022)
x
```

Out[ ]:

```
(5.174011353005595, 9.306383193067887e-07)
```

As the P-Value comes out to be $9.306383193067887e-07$, which is lesser than α, therefore, we reject the Null Hypothesis.

Hence, we can conclude that even though developing countries do not account for Hunger Index, but they still do some work to ensure food security and food production.

Problem Statement 3

Make the Correlation Heatmap

In [ ]:

```
# Define the Pearson Correlation Function
def corrPearson(x,y):
    df = pd.DataFrame({'x': x, 'y': y})
    df2 = pd.DataFrame({
        'x - mean(x)': (df['x'] - pd.Series(list(np.mean(df['x']) for i in
range(len(x))))),
        'y - mean(y)': (df['y'] - pd.Series(list(np.mean(df['y']) for i in
range(len(y)))))
    })
    df2['sqr(x - mean(x))'] = df2['x - mean(x)'] * df2['x - mean(x)']
    df2['sqr(y - mean(y))'] = df2['y - mean(y)'] * df2['y - mean(y)']

    xXy = list()
    for i in range(len(df2)):
        xXy.append(df2['x - mean(x)'][i] * df2['y - mean(y)'][i])
    df2['(x - mean(x)) X (y - mean(y))'] = xXy

    return df2['(x - mean(x)) X (y - mean(y))'].sum() / np.sqrt(df2['sqr(x -
mean(x))'].sum() * df2['sqr(y - mean(y))'].sum())
```

```
# Add the Data
alldata = {
    "Prevalence of Stunting": [-0.014, 0, -0.026, -0.018, 0.0, 0.0, -0.026],
    "Prevalence of Malnutrition": [-0.007, 0, 0.007, 0.001, 0.0, 0.011, 0.007],
    "Prevalence of Anaemia in Women": [42.8,0,47.15,21.05,18.75,17.1,17.1],
    "Annual Farming Sales": [0.09,0,0.08,0.01,0.05,0.07,0.07],
    "Annual Animal Feed Expenditure": [0.03,0,0.33,0.06,0.00,0.07,0.00],
    "Global Hunger Index": [10.12,0,6.36,4.91,2.81,5.4,6.978],
    "GDP per Capita": [1.809,0,5.635,25.851,35.087,37.477,8.520]
}

# Create Correlation Table
correlation = dict()
for i in alldata:
    correlation[i] = list()
    for j in alldata:
        correlation[i].append(corrPearson(alldata[i], alldata[j]))
correlation = pd.DataFrame(correlation, index=list(correlation.keys()))
```

In [ ]:

```
# Plot the Data
fig = px.imshow(correlation, text_auto=True)

# Show the Graph
fig.show()
```