PROJECT 3

Fate is inevitable

Author: Rick Olsen, Cole Geiger, Ashby Maryo

INTRODUCTION

The primary goal of this project is to conduct a comprehensive exploration, analysis, and visualization of essential demographic and economic indicators.

By leveraging a diverse set of datasets, we aim to uncover meaningful insights and trends that contribute to a holistic understanding of the socio-economic landscape.

Our project aims to analyze mortality data from mortality.org.

Area of Focus:

- Birth Rates
- Death Rates
- Income
- Population Statistics

PLATFORMS AND LIBRARIES

Python

Python is the primary programming language used for data analysis and processing.

Pandas

Pandas is used for data analysis, manipulation, and transformation.

SQLAlchemy

SQLAlchemy is used for database interaction and query generation in Python.

SQL

SQL is used for interacting with the database and storing the transformed data.

Pandera

Pandera is used for data processing and validation to ensure data quality and consistency.

Matplotlib

Matplotlib is used for creating basic visualizations of the analyzed data.

Seaborn

Seaborn is used for advanced data visualization and enhancing Matplotlib functions.

DATASET OVERVIEW

THE ANALYSIS OF MORTALITY DATA INVOLVES THE USE OF SEVERAL DATASETS. THESE DATASETS PROVIDE VALUABLE INFORMATION ON BIRTH RATES, DEATH RATES, AND POPULATION STATISTICS, WHICH ARE CRUCIAL FOR UNDERSTANDING AND ANALYZING MORTALITY TRENDS. THE FOLLOWING DATASETS WERE USED IN THIS ANALYSIS:

Datasets Used

Dataset	Description
Births_df	Provides information on the number of births per year.
Income_df	Information showing Size of Household and Median income
Bltper_df	Provides life expectancy and mortality rates by year and age.
Deaths _df	Includes the number of deaths by year, age, and gender.
Population_df	Contains population statistics by year, age, and gender.
Per_capita_df	Provides population and per capita income by year.

Year	Female	Male	Total
1933	1122180	1184820	2307000
1934	1166072	1229928	2396000
1935	1158000	1219000	2377000
1936	1148000	1207000	2355000
1937	1175000	1238000	2413000
1938	1217000	1280000	2497000
1939	1201000	1265000	2466000
1940	1246000	1313000	2559000
1941	1316000	1387000	2703000
1942	1452000	1537000	2989000
1943	1510000	1593000	3103000
1944	1430000	1509000	2939000
1945	1391000	1467000	2858000
1946	1657000	1754000	3411000
1947	1857000	1960000	3817000
1948	1771000	1866000	3637000
1949	1777000	1872000	3649000
1950	1768000	1863000	3631000
1951	1863000	1960000	3823000
1952	1908000	2005000	3913000
1953	1931000	2034000	3965000
1954	1988000	2090000	4078000
1955	2001000	2103000	4104000
1956	2056000	2162000	4218000

Storing Data in SQL (SQLAlchemy)

TO STORE THE ANALYZED MORTALITY DATA, WE WILL BE USING SQLALCHEMY, A PYTHON LIBRARY THAT PROVIDES A SQL TOOLKIT AND OBJECT-RELATIONAL MAPPING (ORM) TOOLS.

SQLALCHEMY ALLOWS US TO CONNECT TO A SQL DATABASE AND STORE OUR PANDAS DATAFRAMES AS SQL TABLES.

Connect to SQL Database ⇨ Create SQL Tables ⇨ Store Pandas DataFrames

Connect to SQL Database

The first step is to establish a connection to the SQL database using SQLAlchemy. This involves specifying the necessary connection details, such as the database type, host, port, username, and password.

Create SQL Tables

Once the connection is established, we can create SQL tables to store our data. We define the table schema, including the column names, data types, and any constraints.

Store Pandas DataFrames

With the SQL tables created, we can now store our Pandas DataFrames in the SQL database. SQLAlchemy provides methods to easily insert data into the tables, either row by row or in bulk.

TRANSFORMATION

Data Loading, Cleaning and Formatting

- Create primary keys to uniquely identify each record.
- Importing CSV's as DataFrames
- Limit the data to the past 50 years to focus on recent trends.
- Format numerical data to ensure consistency and accuracy.
- Round values to the appropriate decimal places for better readability.

Variable Renaming and Dropping

- Rename variables to provide clearer and more descriptive names.
- Drop unneeded variables that are not relevant to the analysis.

Incorporating Additional Health Indicators

- Include additional health indicators to provide a more nuanced analysis.
- Explore correlations between birth rates, death rates, population statistics, and income
- Gain insights into the health impact and median household income.

```python
# define income_df_schema
income_df_schema = pa.DataFrameSchema({
    "Year": pa.Column(int, checks=pa.Check(lambda s: s > 1972)),
    "Number (thousands)": pa.Column(int),
    "Median Income Current dollars": pa.Column(int),
    "Median Income 2022 dollars": pa.Column(int),
    "Mean Income Current dollars": pa.Column(int),
    "Mean Income 2022 dollars": pa.Column(int),
    "Average size of household": pa.Column(float),
})

income_dfv = income_df_schema(income_df)
```

IMAGES

Proj_3/postgres@PostgreSQL 16

No limit

Query Query History

```
SELECT *
FROM "annual_x_age_df"
```

Data Output Messages Notifications

	Annual_Death_Rate double precision	Probability_of_Death double precision	Number_of_Deaths integer	Life_Expectancy integer	ID [PK] character varying (255)	Female_Deaths double precision	Male_Deaths double precision	Total_Deaths double precision	Year integer	Age integer	Female_population double precision	Male_population double precision	Total_population double precision
1	1.755	1.728	1728	71	1973_0	23600.09	31996.44	55596.53	1973	0	1573380.82	1643375.34	3216756.16
2	1.709	1.6840000000000002	1684	71	1974_0	22475.64	30314.51	52790.15	1974	0	1512953.42	1583460.27	3096413.69
3	1.609	1.5859999999999999	1586	72	1975_0	21717.43	28819.38	50536.81	1975	0	1517558.9	1591060.27	3108619.17
4	1.532	1.512	1512	72	1976_0	20949.08	27328	48277.08	1976	0	1530978.08	1602972.6	3133950.68
5	1.434	1.4160000000000001	1416	73	1977_0	20104.73	26884.46	46989.19	1977	0	1561104.11	1636619.18	3197723.29
6	1.38	1.3639999999999999	1364	73	1978_0	19862.63	26100.81	45963.44	1978	0	1611534.25	1691030.14	3302564.39
7	1.327	1.312	1312	73	1979_0	19689.4	25991.42	45680.82	1979	0	1648567.12	1727569.86	3376136.98
8	1.284	1.27	1270	73	1980_0	19725.75	25813.47	45539.22	1980	0	1706540.64	1786571.22	3493111.86
9	1.199	1.187	1187	74	1981_0	18856.9	24461.78	43318.68	1981	0	1750850.42	1832863.26	3583713.68
10	1.164	1.1520000000000001	1152	74	1982_0	18333.01	24082.46	42415.47	1982	0	1770832.33	1854722.16	3625554.49
11	1.1159999999999999	1.105	1105	74	1983_0	17661.95	22977.83	40639.78	1983	0	1780609.01	1865775.68	3646384.69
12	1.099	1.089	1089	74	1984_0	17224.06	22366.35	39590.41	1984	0	1762780.79	1847655.85	3610436.64
13	1.091	1.08	1080	74	1985_0	17078.67	22968.28	40046.95	1985	0	1770423.82	1854993.4	3625417.22
14	1.05	1.0410000000000001	1041	74	1986_0	16674.76	22234.74	38909.5	1986	0	1801361.1	1887764.05	3689125.15
15	1.035	1.0250000000000001	1025	74	1987_0	16612.83	21806.05	38418.88	1987	0	1807388.95	1894031.16	3701420.11
16	1.0330000000000001	1.023	1023	74	1988_0	16905.76	22015.16	38920.92	1988	0	1821704.18	1908273.27	3729977.45
17	1.0370000000000001	1.027	1027	75	1989_0	17523.33	22597.7	40121.03	1989	0	1860348.6	1947269.48	3807618.08
18	0.976	0.968	968	75	1990_0	16719.35	22086.42	38805.77	1990	0	1916414.47	2006217.96	3922632.43
19	0.928	0.9209999999999999	921	75	1991_0	15913.05	21186.78	37099.83	1991	0	1951483.31	2043674.38	3995157.69
20	0.877	0.8710000000000001	871	75	1992_0	15169.69	19635.12	34804.81	1992	0	1949155.9	2039079.13	3988235.03
21	0.859	0.853	853	75	1993_0	14546.96	18992.45	33539.41	1993	0	1920502.71	2012872.64	3933375.35
22	0.827	0.8210000000000001	821	75	1994_0	13925.2	17844.43	31769.63	1994	0	1885457.71	1980116.1	3865573.81
23	0.782	0.777	777	75	1995_0	13006.34	16667.42	29673.76	1995	0	1861161.21	1953025.65	3814186.86
24	0.76	0.755	755	76	1996_0	12551.48	15987.97	28539.45	1996	0	1838712.29	1929416.67	3768128.96
25	0.747	0.742	742	76	1997_0	12270.86	15813.15	28084.01	1997	0	1829189.52	1918888.89	3748078.41
26	0.753	0.748	748	76	1998_0	12601.93	15804.4	28406.33	1998	0	1832735.7	1924255.29	3756990.99
27	0.735	0.73	730	76	1999_0	12298.78	15655.61	27954.39	1999	0	1843308.22	1936022.42	3779330.64
28	0.722	0.718	718	76	2000_0	12324.67	15729.83	28054.5	2000	0	1867013.18	1958928.27	3825941.45
29	0.694	0.69	690	76	2001_0	12095.88	15489.29	27585.17	2001	0	1922474.57	2012047.09	3934521.66
30	0.707	0.7020000000000001	702	77	2002_0	12320.72	15723.69	28044.41	2002	0	1947880.95	2034094.72	3981975.67
31	0.705	0.7000000000000001	700	77	2003_0	12123.69	15910.56	28034.25	2003	0	1937799.78	2025899.66	3963699.44
32	0.698	0.694	694	77	2004_0	12228.71	15726.62	27955.33	2004	0	1952451.82	2042665.26	3995117.08
33	0.7080000000000001	0.7040000000000001	704	77	2005_0	12431.52	16027.59	28459.11	2005	0	1959441.42	2049870.57	4009311.99
34	0.7040000000000001	0.7000000000000001	700	77	2006_0	12550.49	15981.28	28531.77	2006	0	1966326.85	2056789.81	4023116.66
35	0.705	0.701	701	78	2007_0	12852.46	16308.03	29160.49	2007	0	2001620.22	2093392.84	4095013.06
36	0.681	0.677	677	78	2008_0	12386.45	15669.3	28055.75	2008	0	2025029.12	2115315.97	4140345.09
37	0.657	0.653	653	78	2009_0	11589.74	14825.14	26414.88	2009	0	1990941.71	2077042.37	4067984.08
38	0.621	0.618	618	78	2010_0	10884.34	13702.95	24587.29	2010	0	1946038.12	2031431.52	3977469.64
39	0.606	0.603	603	78	2011_0	10658.28	13328.04	23986.32	2011	0	1933864.92	2023530.7	3957395.62
40	0.601	0.598	598	78	2012_0	10490.39	13140.03	23630.42	2012	0	1926803.8	2018143.65	3944947.45
41	0.5950000000000001	0.592	592	78	2013_0	10321.29	13119.96	23441.25	2013	0	1920266.72	2008810.69	3929077.41
42	0.586	0.583	583	78	2014_0	10329.42	12887.07	23216.49	2014	0	1926006.98	2017217.29	3943224.27

Total rows: 49 of 49 Query complete 00:00:00.078

Proj_3/postgres@PostgreSQL 16

No limit

Query Query History

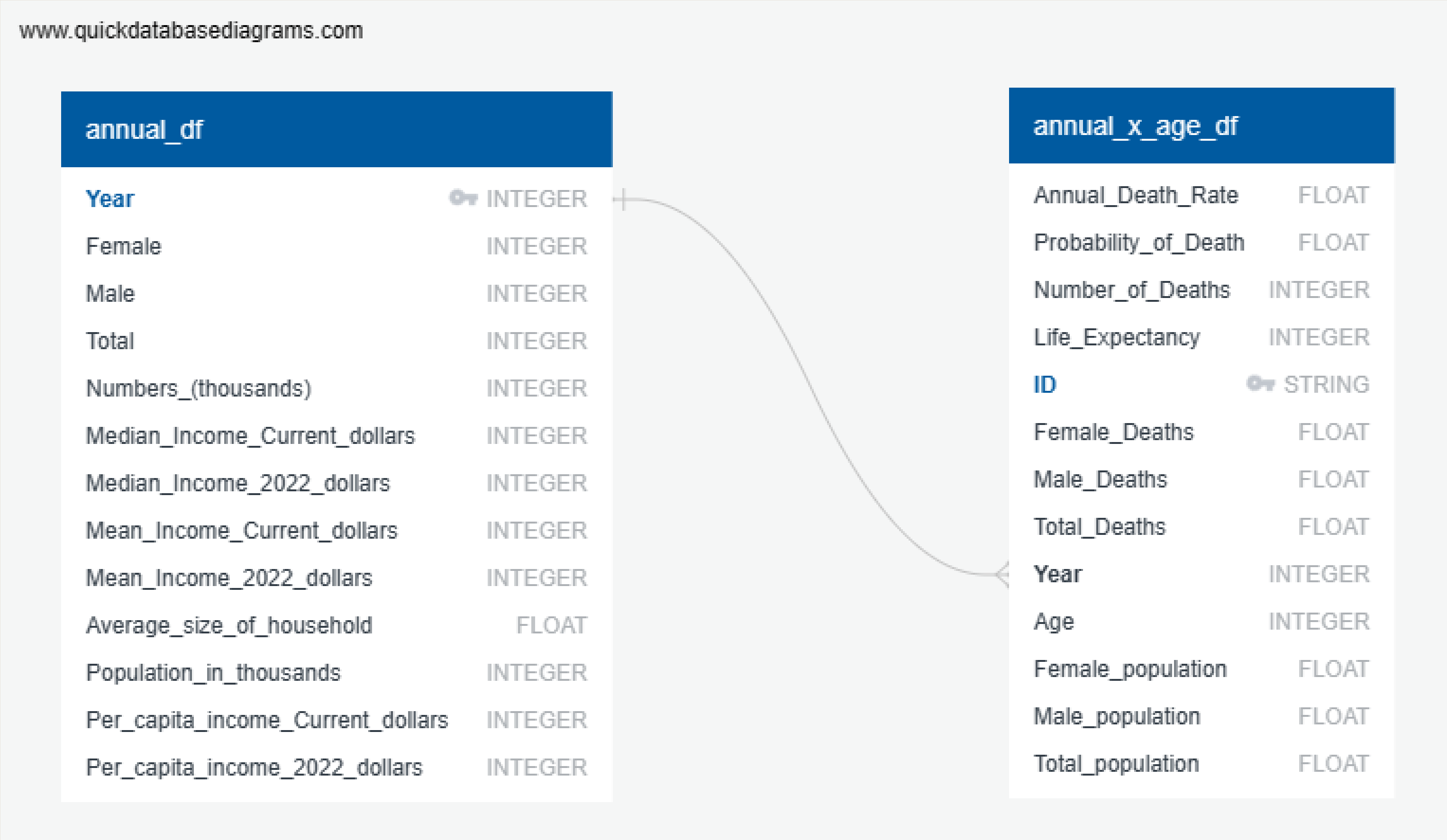
```
SELECT *
FROM "annual_df"
```

Data Output Messages Notifications

	Year [PK] integer	Female integer	Male integer	Total integer	Numbers_(thousands) integer	Median_Income_Current_dollars integer	Median_Income_2022_dollars integer	Mean_Income_Current_dollars integer	Mean_Income_2022_dollars integer	Average_size_of_household double precision	Population_in_thousands integer	Per_capita_income_Current_dollars integer	Per_c integ
1	1975	1531063	1613135	3144198	72870	11800	55100	13780	64340	2.89	211100	4818	
2	1976	1543352	1624436	3167788	74140	12690	56040	14920	65920	2.86	212600	5271	
3	1977	1620716	1705916	3326632	76030	13570	56320	16100	66810	2.81	214200	5785	
4	1978	1623885	1709394	3333279	77330	15060	58510	17730	68860	2.78	215900	6455	
5	1979	1703131	1791267	3494398	80780	16460	58400	19550	69370	2.76	223200	7168	
6	1980	1759642	1852616	3612258	82370	17710	56580	21060	67300	2.73	225200	7787	
7	1981	1768966	1860272	3629238	83530	19070	55630	22790	66460	2.72	227400	8476	
8	1982	1794861	1885676	3680537	83920	20170	55470	24310	66850	2.73	229600	8980	
9	1983	1773380	1865553	3638933	85410	20890	55120	25400	67040	2.71	231900	9494	
10	1984	1789651	1879490	3669141	86790	22420	56780	27460	69570	2.69	234100	10330	
11	1985	1832578	1927983	3760561	88460	23620	57860	29070	71210	2.67	236700	11010	
12	1986	1831679	1924868	3756547	89480	24900	60010	30760	74140	2.66	238800	11670	
13	1987	1858241	1951153	3809394	91120	26060	60760	32410	75560	2.64	241200	12390	
14	1988	1907086	2002424	3909510	92830	27230	61210	34020	76480	2.62	243700	13120	
15	1989	1971468	2069490	4040958	93350	28910	62260	36520	78660	2.63	246200	14060	
16	1990	2028717	2129495	4158212	94310	29940	61500	37400	76820	2.63	248900	14390	
17	1991	2009389	2101518	4110907	95670	30130	59710	37920	75160	2.62	251400	14620	
18	1992	1982917	2082097	4065014	96430	30640	59210	38840	75070	2.66	256800	14850	
19	1993	1951379	2048861	4000240	97110	31240	58920	41430	78140	2.67	259800	15780	
20	1994	1930178	2022589	3952767	98990	32260	59550	43130	79610	2.65	262100	16560	
21	1995	1903234	1996355	3899589	99630	34080	61440	44940	81030	2.65	264300	17230	
22	1996	1901014	1990480	3891494	101000	35490	62350	47120	82780	2.64	266800	18140	
23	1997	1895298	1985596	3880894	102500	37010	63640	49690	85460	2.62	269100	19240	
24	1998	1925348	2016205	3941553	103900	38890	65980	51860	87980	2.61	271700	20120	
25	1999	1932563	2026854	3959417	106400	40700	67650	54740	90990	2.6	276800	21240	
26	2000	1981845	2076969	4058814	108200	41990	67470	57140	91810	2.58	279500	22350	
27	2001	1968011	2057922	4025933	109300	42230	66360	58210	91470	2.58	282100	22850	
28	2002	1963747	2057979	4021726	111300	42410	65820	57850	89790	2.57	285900	22790	
29	2003	1996415	2093535	4089950	112000	43320	65860	59070	89810	2.57	288300	23280	
30	2004	2007391	2104661	4112052	113300	44330	65760	60470	89690	2.57	291200	23860	
31	2005	2019367	2118982	4138349	114400	46330	66780	63340	91310	2.57	293800	25040	
32	2006	2081318	2184237	4265555	116000	48200	67520	66570	93260	2.56	296800	26350	
33	2007	2108162	2208071	4316233	116800	50230	68610	67610	92340	2.56	299100	26800	
34	2008	2074305	2173389	4247694	117200	50300	66280	68420	90150	2.57	301500	26960	
35	2009	2016809	2113856	4130665	117500	49780	65850	67980	89920	2.59	304300	26530	
36	2010	1952451	2046935	3999386	119900	49280	64300	67390	87940	2.56	306600	26560	
37	2011	1929538	2024052	3953590	121100	50050	63350	69680	88190	2.55	308800	27550	
38	2012	1931407	2021434	3952841	122500	51020	63350	71270	88500	2.54	311100	28280	
39	2013	1919227	2012954	3932181	123000	51940	63720	72640	89120	2.55	313400	28830	
40	2014	1947375	2040701	3988076	124600	53660	64900	75740	91610	2.54	316200	30180	
41	2015	1942336	2036161	3978497	125800	56520	68410	79260	95950	2.53	318900	31650	
42	2016	1927692	2018183	3945875	126200	59040	70840	83140	99760	2.54	320400	33210	

Total rows: 47 of 47 Query complete 00:00:00.215 Ln 2, Col 14

DATA DESIGN -DIAGRAM



DATA PROCESSING AND VALIDATION (PANDERA)

IN THIS PROJECT, WE WILL BE USING PANDERA, A PYTHON LIBRARY, FOR DATA PROCESSING AND VALIDATION. PANDERA ALLOWS US TO DEFINE AND VALIDATE DATA SCHEMAS, ENSURING DATA QUALITY AND CONSISTENCY THROUGHOUT OUR ANALYSIS.

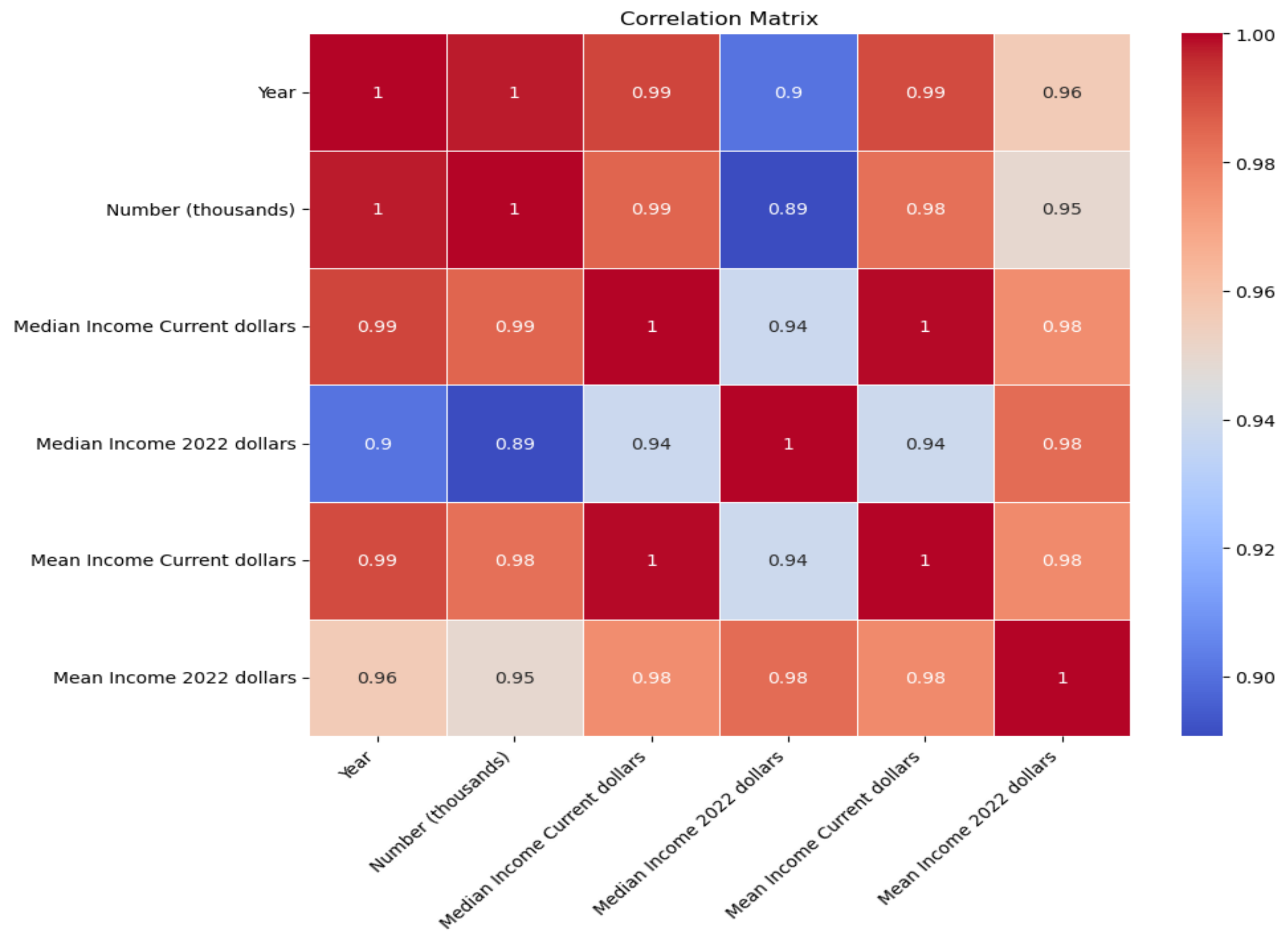
Data Processing and Validation Steps

Step	Description
1. Data Cleaning	Remove any duplicate or irrelevant data, handle missing values, and standardize data formats.
2. Data Transformation	Apply necessary transformations to the data, such as aggregating or disaggregating data, creating new variables, or merging datasets.
3. Data Validation	Define and validate data schemas using Pandera to ensure data quality and consistency. Validate data types, ranges, and relationships between variables.
4. Data Quality Assurance	Perform quality checks on the processed data to ensure accuracy, completeness, and integrity. Identify and resolve any data quality issues.

CORRELATION HEATMAP AND DISTRIBUTION ANALYSIS

Seaborn Heatmap

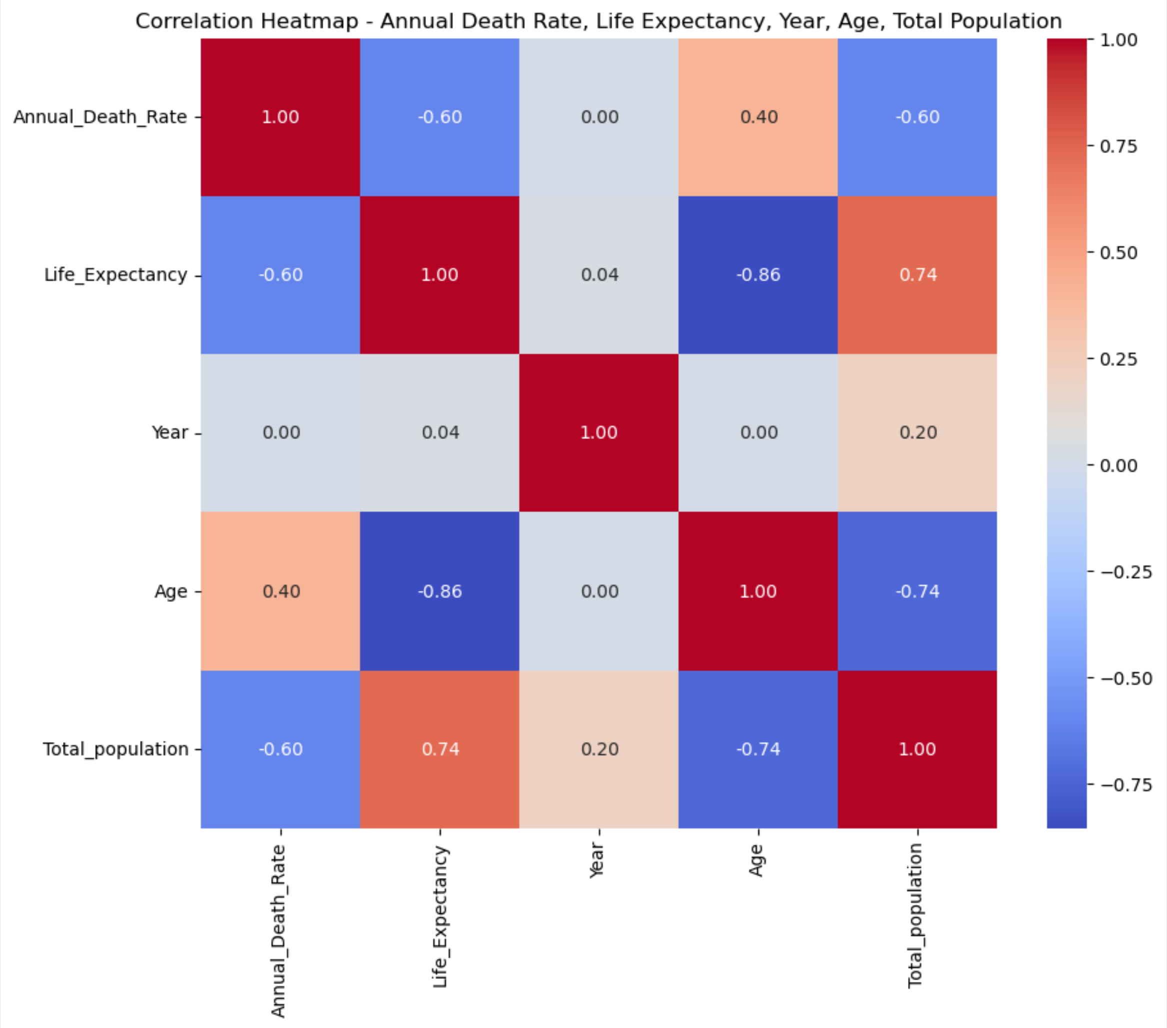
- To unveil relationships between income-related variables, we've utilized Seaborn to create a heatmap. This graphical representation allows for a quick and intuitive assessment of correlations, aiding in identifying patterns and potential areas of interest.



DATA VISUALIZATION (MATPLOTLIB AND SEABORN)

- Positive correlation between "Annual Death Rate" and "Total Population" would suggest that as the total population increases, the annual death rate tends to increase.

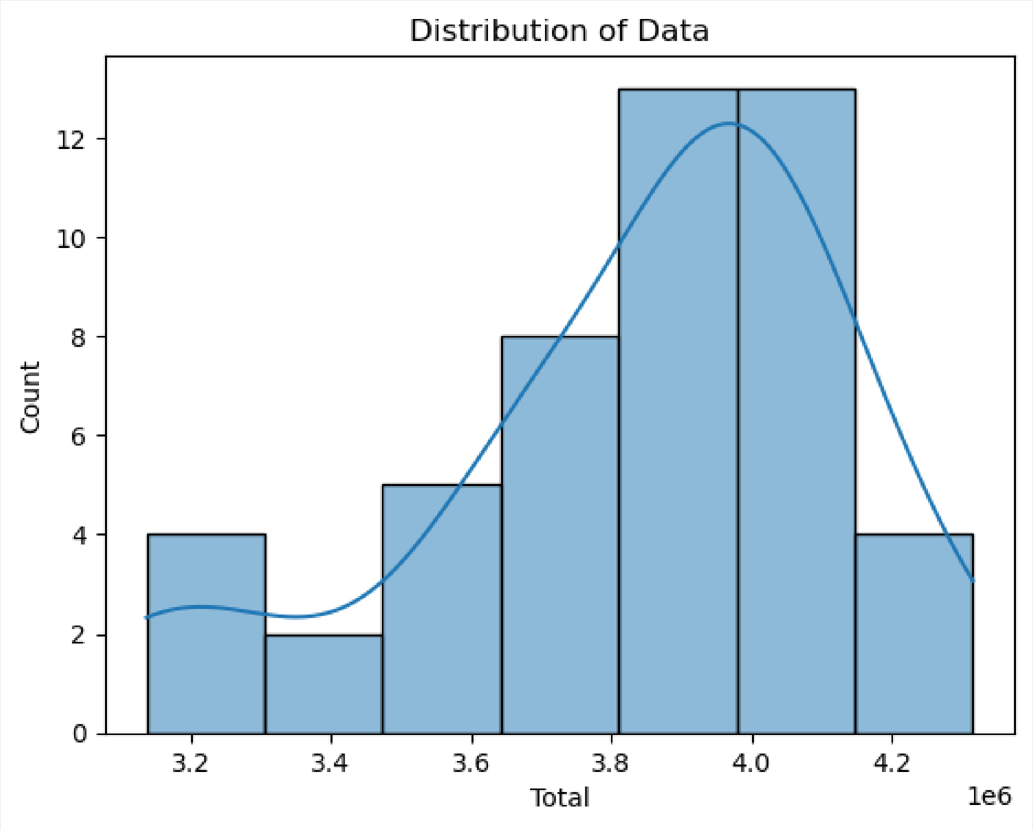
- Negative correlation between "Life Expectancy" and "Annual Death Rate" would suggest that as life expectancy increases, the annual death rate tends to decrease.

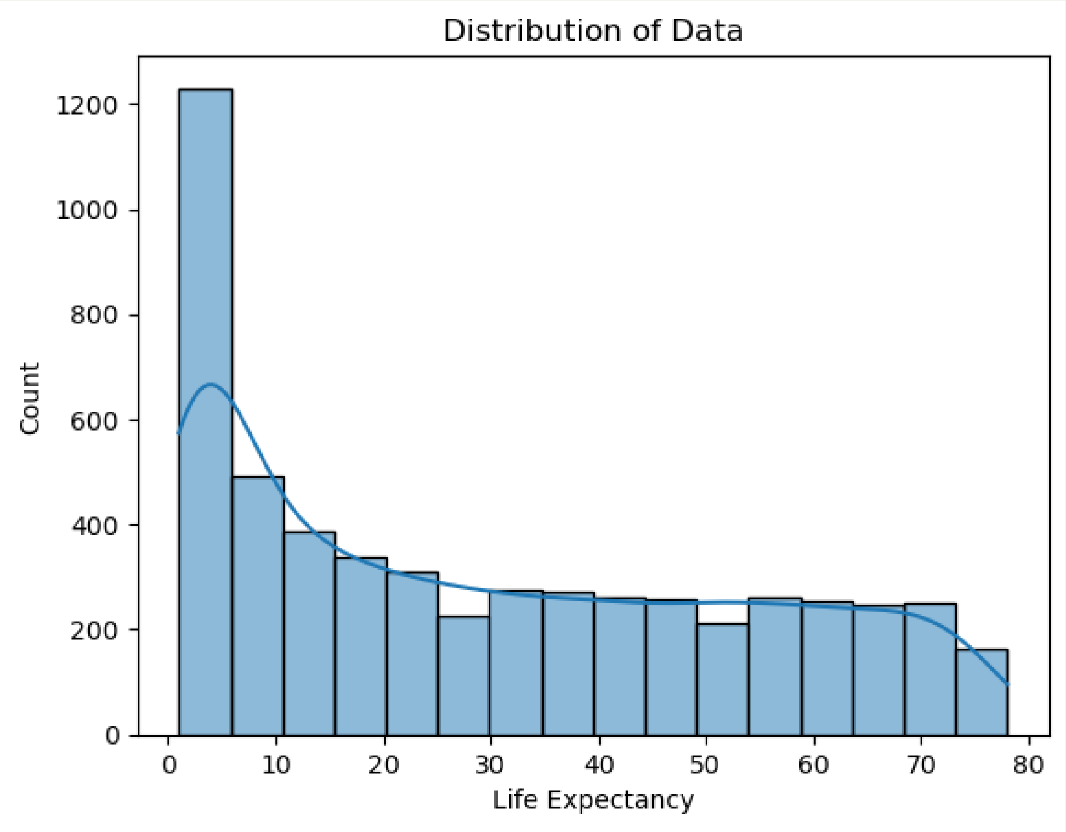


CORRELATION HEATMAP AND DISTRIBUTION ANALYSIS

Histograms with Seaborn:

- The distribution analysis involves using Seaborn to construct histograms for both births and life expectancy. These visualizations provide insights into the frequency and distribution of these events, offering a deeper understanding of the underlying trends.





RESULTS

Key Findings

- Birth rates have been steadily declining over the past decade, indicating a decrease in fertility rates.
- Death rates have been relatively stable, suggesting improvements in healthcare and medical advancements.
- The population has been aging, with a significant increase in the elderly population.

Insights

- The decrease in birth rates may have implications for future population growth and workforce dynamics.
- The stable death rates indicate the effectiveness of healthcare systems in maintaining overall health and well-being.
- The aging population presents challenges and opportunities for healthcare, retirement planning, and social support systems.

Ethics and Privacy

Data Confidentiality

- The project places a high priority on data confidentiality, ensuring that sensitive information is protected and only accessible to authorized individuals.

Responsible Data Handling

- Data handling practices adhere to ethical standards, including obtaining informed consent, anonymizing data, and securely storing and transmitting data.

Fairness in Analysis

- The analysis process is conducted with fairness in mind, avoiding bias and ensuring that all relevant factors are considered.

Accurate and Responsible Sharing of Findings

- The project is committed to sharing findings accurately and responsibly, avoiding misinterpretation and providing clear context for the data.

Privacy, Transparency, and Ethical Standards

- Privacy, transparency, and ethical standards are respected throughout the project, ensuring that the rights and well-being of individuals are protected.

SOURCES

Mortality.org and US Census.gov

Mortality.org is a comprehensive database that provides access to mortality data from various sources. It includes data on birth rates, death rates, and population statistics from different countries and regions.

Accessing Data

To access the data from mortality.org, we utilized web scraping techniques to extract the relevant information. Python libraries such as BeautifulSoup and Requests were used to automate the process and retrieve the data in a structured format.

Data Analysis

The extracted data was then transformed and analyzed using Python and Pandas.