A DATA ANALYSIS REPORT FOR THE LAUNCH OF A HEALTH TRACKER SMARTWATCH BUSINESS

Prepared by: Ralph Matthew Ong

Date Submitted: September 09, 2023

Project Information

I. Project Background:

The startup company is introducing health tracker smartwatches to the U.S. market, with its main headquarters situated in Los Angeles, California, within the United States of America.

II. Problem Statement:

The business is planning to sustain the startup company and expand in other states in U.S. but contemplating to select the most feasible state location for expansion. The factors to take into consideration are the economic, political, and social conditions of their target state such as market potential, business environment, business cost, availability of talent & workforce, and quality of life. The business is interested in evaluating the 50 states in terms of 5 key metrics: state income (US$ per person), crime level of corruption conviction (per capita), population size, property prices (US$ per square meter), and healthcare spendings (US$ per person). In addition to the key metrics, the business wants to implement competitor analysis in order to understand their potential competitors' performance and locations to help them in their decision.

III. Project Objectives:

The main goal of this study is to select 1 or 2 states with the best potential for business expansion in terms of high income and low corruption rates.

In order to optimize the best solution to the problem, the data analyst aims to further explore and analyze the data based on U.S.A.'s state income, corruption level, population, health spendings, and property prices, as well as competitor information to generate valuable results and data-driven insights on the following information:

1. Income:
a.) Average income per person for each state.
b.) State with the highest and lowest average income.
c.) Distribution of income for all states.
d.) Percentage of income by state and corruption convictions per capita in relation to the total amount of income.
e.) Map of U.S.A. with level of state's income.

2. Corruption Conviction Rate:
a.) Corruption conviction rates for each state.
b.) State with the highest and lowest corruption conviction rates.
c.) Distribution of level of corruption conviction rates for all states.
d.) Minimum, average, and maximum corruption levels for different regions of U.S.A.
e.) Map of U.S.A. with states colored by level of corruption conviction rates.

3. Population:
a.) Population size of each state.
b.) State with the highest and lowest population size.
c.) Distribution of the population for all states.
d.) States with population over 10 million people.
e.) Map visualizing population size for each state.

4. Healthcare Spendings:
a.) Average healthcare spendings per person for each state.
b.) State that spends the most and least amount of healthcare spendings per person.
c.) States with the highest maximum, lowest minimum healthcare spendings per person, and how does it compare to their averages.
d.) Map of U.S.A. with healthcare spendings across different states.

5. Property Prices:
a.) Average property price per square meter for each state.
b.) State that has the most and least expensive property prices on average.
c.) Map of U.S.A. with average property price per square meter for different states.

6. Competitors:
a.) Average profit of competitors by state.
b.) Top competitor information.
c.) Financial viability metrics of competitor companies.
d.) Total profit of competitors by state locations and compare with the state's population.
e.) Impact of the states' level of corruption rate to the profit of competitors in different states.
f.) Correlation between states' level of corruption and competitor companies' profits.

7. Correlation between average income per person, corruption convictions per capita, population size, average property price per square meter, and average healthcare spending per person, and how these factors impact one another.

8. Outliers found in the data that could skew the analysis results.

9. Test if the average profit of the competitors is greater than the median profit.

10. Test if the state with the highest population is over 12% of U.S.A.'s total population.

IV. Methodology:

These are the procedures in conducting data analysis:

1. Define the Problem
2. Set Objectives
3. PostgreSQL:
    - ⇒ Data Preparation in PostgreSQL
    - ⇒ Data Cleaning in PostgreSQL
    - ⇒ Data Analysis using SQL Queries
    - ⇒ Export SQL data to MS Excel
4. MS Excel:
    - ⇒ Data Visualization in MS Excel using Pivot Tables & Charts
5. Python:
    - ⇒ Data Preparation in Google Colab
    - ⇒ Data Transformation in Python
    - ⇒ Data Analysis using Python
    - ⇒ Data Visualization in Python using Charts
    - ⇒ Export Python Dataframe to Power BI
6. Power BI:
    - ⇒ Data Preparation in Power BI
    - ⇒ Data Analysis in Power BI
    - ⇒ Data Modeling in Power BI
    - ⇒ Data Visualization in Power BI using Dashboards
7. Generate Insights
8. Provide Recommendations

PostgreSQL

I. Data Preparation in PostgreSQL:

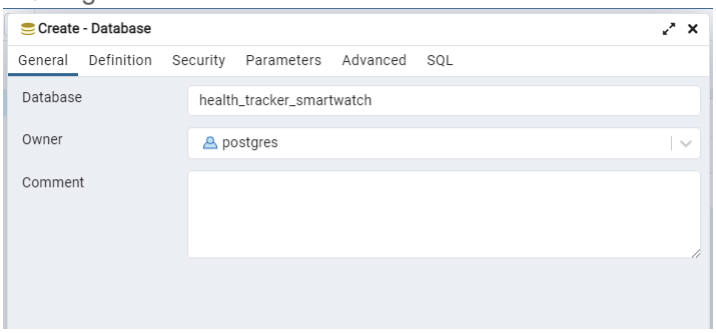
1. Started the process in PostgreSQL 15 server.

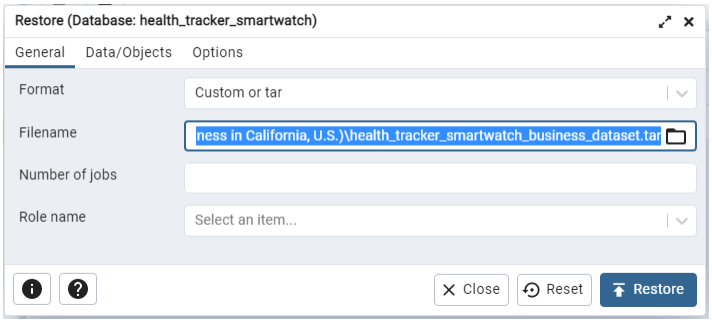
2. Created a database named “health_tracker_smartwatch”.

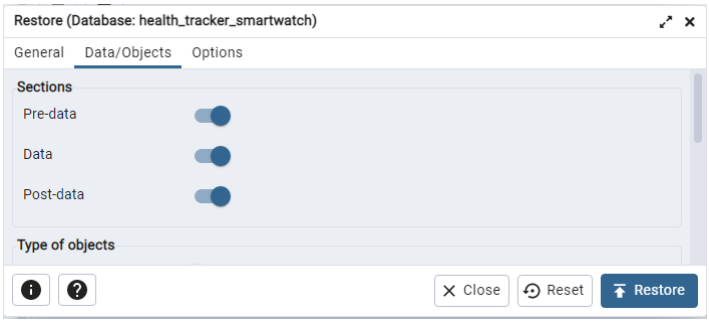
CREATE DATABASE health_tracker_smartwatch

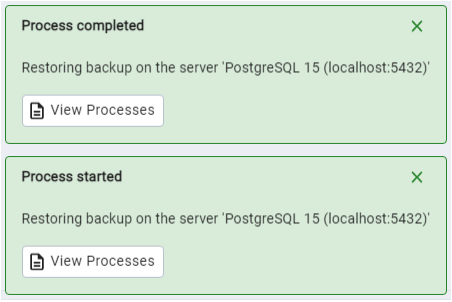
3. Loaded the .tar file dataset by restoring it in PostgreSQL.

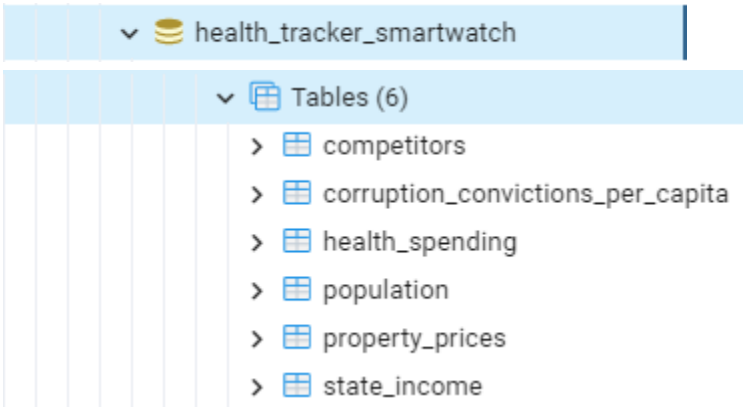
SQL Figures 1-12

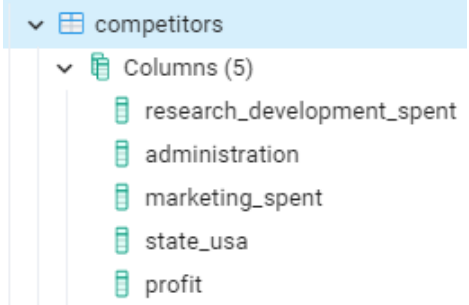


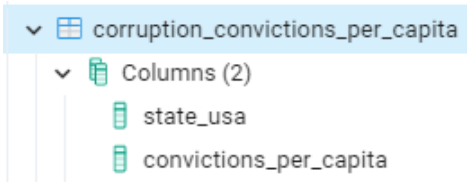


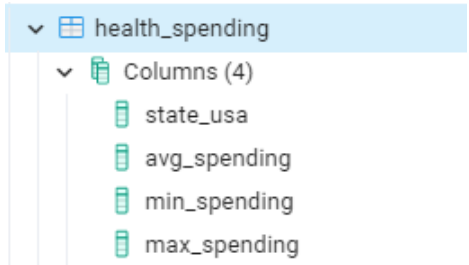












II. Data Cleaning Process in PostgreSQL:

1. Inspected all 6 tables.

SELECT * FROM competitors

	research_development_spent numeric	administration numeric	marketing_spent numeric	state_usa character varying	profit numeric
1	165349.2	136897.8	471784.1	New York	192261.83
2	162597.7	151377.59	443898.53	California	191792.06
3	153441.51	101145.55	407934.54	Florida	191050.39
4	144372.41	118671.85	383199.62	New York	182901.99
5	142107.34	91391.77	366168.42	Florida	166187.94
6	131876.9	99814.71	362861.36	New York	156991.12
7	134615.46	147198.87	127716.82	California	156122.51

Total rows: 271 of 271 | Query complete 00:00:00.059 | Ln 1, Col 1

SQL Output 1

SELECT * FROM corruption_convictions_per_capita

	state_usa character varying	convictions_per_capita numeric
1	Alabama	2.15
2	Alaska	1.06
3	Arizona	1.40
4	Arkansas	3.02
5	California	1.09
6	Colorado	0.80
7	Connecticut	2.01

Total rows: 50 of 50 | Query complete 00:00:00.109 | Ln 2, Col 1

SQL Output 2

SELECT * FROM health_spending

	state_usa character varying	avg_spending numeric	min_spending numeric	max_spending numeric
1	Alabama	200.50	50.00	500.00
2	Alaska	300.25	100.00	750.00
3	Arizona	150.00	25.00	300.00
4	Arkansas	175.00	75.00	400.00
5	California	250.75	50.00	600.00
6	Colorado	225.50	100.00	500.00
7	Connecticut	300.00	150.00	700.00

Total rows: 50 of 50 | Query complete 00:00:00.108 | Ln 3, Col 1

SQL Output 3

SELECT * FROM population

	state_usa character varying	estimate numeric
1	Alabama	4903185
2	Alaska	731545
3	Arizona	7278717
4	Arkansas	3017804
5	California	39512223
6	Colorado	5758736
7	Connecticut	3565287

Total rows: 51 of 51 | Query complete 00:00:00.050 | Ln 4, Col 1

SQL Output 4

SELECT * FROM property_prices

	state_usa character varying	avg_price numeric	min_price numeric	max_price numeric
1	Alabama	1797.50	1200.00	2500.00
2	Alaska	2684.00	2000.00	3500.00
3	Arizona	2356.75	1500.00	4000.00
4	Arkansas	1499.25	1000.00	2500.00
5	California	5832.50	3500.00	9000.00
6	Colorado	2987.25	2000.00	4500.00
7	Connecticut	3837.00	3000.00	5500.00

Total rows: 50 of 50 | Query complete 00:00:00.056 | Ln 5, Col 1

SQL Output 5

SELECT * FROM state_income

	state_usa character varying	average_income numeric	minimum_income numeric	maximum_income numeric
1	Alabama	51113	23999	96993
2	Alaska	76440	35219	134318
3	Arizona	62283	29466	113589
4	Arkansas	48829	23028	90052
5	California	80440	37698	149265
6	Colorado	76240	35636	130714
7	Connecticut	79287	37426	142596

Total rows: 50 of 50 | Query complete 00:00:00.052 | Ln 6, Col 1

SQL Output 6

2. Checked for distinct values in the “state_usa” column.

SELECT DISTINCT state_usa FROM competitors

	state_usa character varying
1	Oklahoma
2	Colorado
3	Mississippi
4	Florida
5	Delaware

Total rows: 41 of 41 | Query complete 00:00:00.052 | Ln 1, Col 1

SQL Output 7

SELECT DISTINCT state_usa FROM corruption_convictions_per_capita

	state_usa character varying
1	Nevada
2	West Virginia
3	South Carolina
4	New Mexico
5	Arkansas

Total rows: 50 of 50 | Query complete 00:00:00.084 | Ln 2, Col 1

SQL Output 8

SELECT DISTINCT state_usa FROM health_spending

	state_usa character varying
1	Nevada
2	West Virginia
3	South Carolina
4	New Mexico
5	Arkansas

Total rows: 50 of 50 | Query complete 00:00:00.052 | Ln 3, Col 1

SQL Output 9

SELECT DISTINCT state_usa FROM population

	state_usa character varying
1	Oklahoma
2	North Carolina
3	Colorado
4	Mississippi
5	Florida

Total rows: 51 of 51 | Query complete 00:00:00.135 | Ln 4, Col 1

SQL Output 10

SELECT DISTINCT state_usa FROM property_prices

	state_usa character varying
1	Nevada
2	West Virginia
3	South Carolina
4	New Mexico
5	Arkansas

Total rows: 50 of 50 | Query complete 00:00:00.061 | Ln 5, Col 1

SQL Output 11

SELECT DISTINCT state_usa FROM state_income

	state_usa character varying
1	Nevada
2	West Virginia
3	South Carolina
4	New Mexico
5	Arkansas

Total rows: 50 of 50 | Query complete 00:00:00.046 | Ln 6, Col 1

SQL Output 12

3. Check for duplicate row values in “competitors” table by sorting “state_usa” & “profit” columns in descending order.

SELECT * FROM competitors
ORDER BY state_usa, profit DESC

	research_development_spent numeric	administration numeric	marketing_spent numeric	state_usa character varying	profit numeric
1	1496277.99	104491.9	74515	Alabama	2430410.99
2	989534.99	73696.9	59152	Alabama	1397050.99
3	933266.99	107860.9	51672	Alabama	1346449.99
4	352053.99	87072.9	31951	Alabama	398021.99
5	901313.99	87164.9	87977	Alabama	120362.99

Total rows: 271 of 271 | Query complete 00:00:00.051 | Ln 1, Col 1

SQL Output 13

4. Selected “competitors” table without the 20 duplicated rows.

SELECT DISTINCT * FROM competitors
ORDER BY state_usa, profit DESC

	research_development_spent numeric	administration numeric	marketing_spent numeric	state_usa character varying	profit numeric
1	1496277.99	104491.9	74515	Alabama	2430410.99
2	989534.99	73696.9	59152	Alabama	1397050.99
3	933266.99	107860.9	51672	Alabama	1346449.99
4	352053.99	87072.9	31951	Alabama	398021.99
5	901313.99	87164.9	87977	Alabama	120362.99

Total rows: 251 of 251 | Query complete 00:00:00.052 | Ln 1, Col 1

SQL Output 14

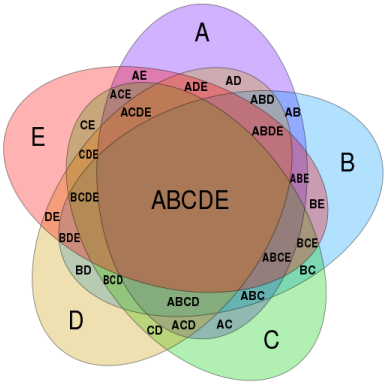
5. Selected the matching values between the 5 dimension tables (excluding the mismatched value “District of Columbia” under the “state_usa” column in the “population” table).

```
SELECT corruption_convictions_per_capita.state_usa,
convictions_per_capita,
avg_spending, min_spending, max_spending,
estimate,
avg_price, min_price, max_price,
average_income, minimum_income, maximum_income
FROM corruption_convictions_per_capita
INNER JOIN health_spending ON corruption_convictions_per_capita.state_usa = health_spending.state_usa
INNER JOIN population ON corruption_convictions_per_capita.state_usa = population.state_usa
INNER JOIN property_prices ON corruption_convictions_per_capita.state_usa = property_prices.state_usa
INNER JOIN state_income ON corruption_convictions_per_capita.state_usa = state_income.state_usa
```

	state_usa character varying	convictions_per_capita numeric	avg_spending numeric	min_spending numeric	max_spending numeric	estimate numeric	avg_price numeric	min_price numeric	max_price numeric	averag numeri
1	Alabama	2.15	200.50	50.00	500.00	4903185	1797.50	1200.00	2500.00	
2	Alaska	1.06	300.25	100.00	750.00	731545	2684.00	2000.00	3500.00	
3	Arizona	1.40	150.00	25.00	300.00	7278717	2356.75	1500.00	4000.00	
4	Arkansas	3.02	175.00	75.00	400.00	3017804	1499.25	1000.00	2500.00	
5	California	1.09	250.75	50.00	600.00	39512223	5832.50	3500.00	9000.00	

Total rows: 50 of 50 | Query complete 00:00:00.168 | Ln 1, Col 1

SQL Output 15



SQL Figure 13

6. Fact and Dimension Tables recognized.

Fact Table: competitors
Dimension Tables: corruption_convictions_per_capita, health_spending, population, property_prices, state_income

7. Created “competitors_revised” table (removed the 20 duplicated rows from the competitors table).

```
CREATE TABLE competitors_revised AS (
        SELECT DISTINCT * FROM competitors
        ORDER BY state_usa, profit ASC
);
```

```
SELECT 251

Query returned successfully in 54 msec.
```

Total rows: 251 of 251 | Query complete 00:00:00.054 | Ln 1, Col 1

SQL Output 16

8. Inspected “competitors_revised” table.

SELECT * FROM competitors_revised

	research_development_spent numeric	administration numeric	marketing_spent numeric	state_usa character varying	profit numeric
1	901313.99	87164.9	87977	Alabama	120362.99
2	352053.99	87072.9	31951	Alabama	398021.99
3	933266.99	107860.9	51672	Alabama	1346449.99
4	989534.99	73696.9	59152	Alabama	1397050.99
5	1496277.99	104491.9	74515	Alabama	2430410.99

Total rows: 251 of 251 | Query complete 00:00:00.204 | Ln 1, Col 1

SQL Output 17

9. Created “us_states” table containing inner join results of the 5 dimension tables.

```
CREATE TABLE us_states AS (
        SELECT corruption_convictions_per_capita.state_usa,
        convictions_per_capita,
        avg_spending, min_spending, max_spending,
        estimate,
        avg_price, min_price, max_price,
        average_income, minimum_income, maximum_income
        FROM corruption_convictions_per_capita
        INNER JOIN health_spending ON corruption_convictions_per_capita.state_usa = health_spending.state_usa
        INNER JOIN population ON corruption_convictions_per_capita.state_usa = population.state_usa
        INNER JOIN property_prices ON corruption_convictions_per_capita.state_usa = property_prices.state_usa
        INNER JOIN state_income ON corruption_convictions_per_capita.state_usa = state_income.state_usa
);
```

```
SELECT 50

Query returned successfully in 63 msec.
```

Total rows: 50 of 50 | Query complete 00:00:00.063 | Ln 1, Col 1

SQL Output 18

10. Inspected the “us_states” table (50 rows & 12 columns).

SELECT * FROM us_states

	state_usa character varying	convictions_per_capita numeric	avg_spending numeric	min_spending numeric	max_spending numeric	estimate numeric	avg_price numeric	min_price numeric	max_price numeric	averag numeri
1	Alabama	2.15	200.50	50.00	500.00	4903185	1797.50	1200.00	2500.00	
2	Alaska	1.06	300.25	100.00	750.00	731545	2684.00	2000.00	3500.00	
3	Arizona	1.40	150.00	25.00	300.00	7278717	2356.75	1500.00	4000.00	
4	Arkansas	3.02	175.00	75.00	400.00	3017804	1499.25	1000.00	2500.00	
5	California	1.09	250.75	50.00	600.00	39512223	5832.50	3500.00	9000.00	

Total rows: 50 of 50 | Query complete 00:00:00.113 | Ln 1, Col 1

SQL Output 19

III. Data Analysis using SQL Queries:

1. Total of the Average Income of all the 50 states.

SELECT SUM(average_income) FROM us_states

	sum numeric
1	3157304

Total rows: 1 of 1 | Query complete 00:00:00.047 | Ln 1, Col 1

SQL Output 20

2. Percentage of Income by State and Corruption Convictions per Capita in relation to Total Amount.

```
SELECT state_usa, convictions_per_capita, average_income, ROUND((average_income / (SELECT SUM(average_income) FROM us_states) * 100), 2) AS relative_percentage_average_income FROM us_states
GROUP BY state_usa, convictions_per_capita, average_income
ORDER BY average_income DESC
```

	state_usa character varying	convictions_per_capita numeric	average_income numeric	relative_percentage_average_income numeric
1	Maryland	1.38	89392	2.83
2	Massachusetts	2.27	82427	2.61
3	New Jersey	1.90	81740	2.59
4	California	1.09	80440	2.55
5	Connecticut	2.01	79287	2.51

Total rows: 50 of 50 | Query complete 00:00:00.061 | Ln 1, Col 1

SQL Output 21

3. State with the Lowest Average Income.

```
SELECT state_usa, average_income FROM us_states
ORDER BY average_income ASC
LIMIT 1
```

	state_usa character varying	average_income numeric
1	West Virginia	46254

Total rows: 1 of 1 | Query complete 00:00:00.087 | Ln 1, Col 1

SQL Output 22

4. State with the Highest Average Income.

```
SELECT state_usa, average_income FROM us_states
ORDER BY average_income DESC
LIMIT 1
```

	state_usa character varying	average_income numeric
1	Maryland	89392

Total rows: 1 of 1 | Query complete 00:00:00.103 | Ln 1, Col 1

SQL Output 23

5. State with the Lowest Corruption Conviction Rates.

```
SELECT state_usa, convictions_per_capita FROM us_states
ORDER BY convictions_per_capita ASC
LIMIT 1
```

	state_usa character varying	convictions_per_capita numeric
1	Hawaii	0.43

Total rows: 1 of 1 | Query complete 00:00:00.045 | Ln 1, Col 1

SQL Output 24

6. State with the Highest Corruption Conviction Rates.

```
SELECT state_usa, convictions_per_capita FROM us_states
ORDER BY convictions_per_capita DESC
LIMIT 1
```

	state_usa character varying	convictions_per_capita numeric
1	Rhode Island	8.35

Total rows: 1 of 1 | Query complete 00:00:00.107 | Ln 1, Col 1

SQL Output 25

7. Average Profit of Competitors in each state.

```
SELECT state_usa, SUM(profit), COUNT(profit), round(AVG(profit), 2) AS rounded_average_profit FROM competitors_revised
GROUP BY state_usa
ORDER BY AVG(profit) DESC
```

	state_usa character varying	sum numeric	count bigint	rounded_average_profit numeric
1	Arizona	2942282.99	1	2942282.99
2	Delaware	9060875.96	4	2265218.99
3	Oklahoma	15699222.93	7	2242746.13
4	Alaska	9845632.95	5	1969126.59
5	Minnesota	7859138.96	4	1964784.74

Total rows: 41 of 41 | Query complete 00:00:00.142 | Ln 1, Col 1

SQL Output 26

8. Created a new dataset named “megastate” with states having Population above 10,000,000.

```
CREATE TABLE megastate AS (
	SELECT state_usa, estimate FROM us_states
	WHERE estimate > 10000000
	ORDER BY estimate DESC
);
```

```
SELECT 9

Query returned successfully in 57 msec.
```

Total rows: 9 of 9 | Query complete 00:00:00.057 | Ln 5, Col 3

SQL Output 27

9. Inspected the “megastate” table.

```
SELECT * FROM megastate
```

	state_usa character varying	estimate numeric
1	California	39512223
2	Texas	28995881
3	Florida	21477737
4	New York	19453561
5	Pennsylvania	12801989

Total rows: 9 of 9 | Query complete 00:00:00.081 | Ln 1, Col 1

SQL Output 28

10. Average Healthcare Spending for each state in comparison with its corresponding population.

SELECT state_usa, avg_spending AS average_healthcare_spending, estimate AS population_size FROM us_states
ORDER BY avg_spending DESC

	state_usa character varying	average_healthcare_spending numeric	population_size numeric
1	Rhode Island	350.00	1059361
2	Massachusetts	350.00	6892503
3	New Jersey	350.00	8882190
4	Hawaii	350.00	1415872
5	Alaska	300.25	731545

Total rows: 50 of 50 | Query complete 00:00:00.068 | Ln 1, Col 1

SQL Output 29

11. Created “competitor_state” table with summarized total profit by state of competitors.

```
CREATE TABLE competitor_state AS (
        SELECT state_usa, SUM(profit) FROM competitors_revised
        GROUP BY state_usa
);
```

```
SELECT 41

Query returned successfully in 118 msec.
```

Total rows: 41 of 41 | Query complete 00:00:00.118 | Ln 1, Col 1

SQL Output 30

12. Total Profit of Competitors by state and its population.

SELECT competitor_state.state_usa, sum AS total_profit_competitor_state, estimate AS population_size FROM competitor_state
INNER JOIN us_states
ON competitor_state.state_usa = us_states.state_usa
ORDER BY sum DESC

	state_usa character varying	total_profit_competitor_state numeric	population_size numeric
1	Wyoming	16710539.90	578759
2	Oklahoma	15699222.93	3956971
3	Oregon	14574425.91	4217737
4	Montana	14333030.91	1068778
5	Mississippi	12863827.91	2976149
6	Washington	12466539.92	7614893

Total rows: 41 of 41 | Query complete 00:00:00.131 | Ln 1, Col 1

SQL Output 31

IV. PostgreSQL Data Exporting:

1. Exported “us_states” table (50 rows & 12 columns) to .csv file.

SELECT * FROM us_states

Save results to file F8

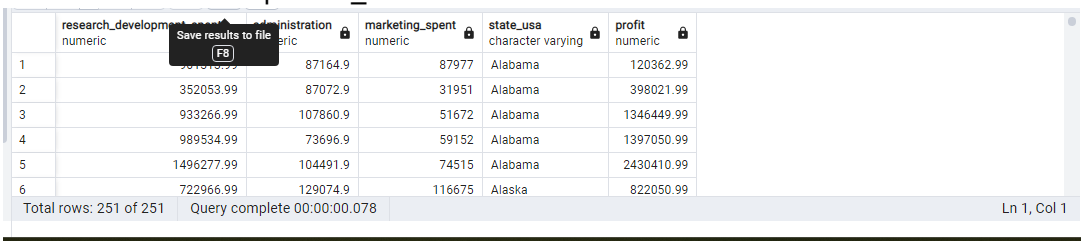
	state_usa character varying	capita	avg_spending numeric	min_spending numeric	max_spending numeric	estimate numeric	avg_price numeric	min_price numeric	max_price numeric	average numeric
1	Alabama	2.15	200.50	50.00	500.00	4903185	1797.50	1200.00	2500.00	
2	Alaska	1.06	300.25	100.00	750.00	731545	2684.00	2000.00	3500.00	
3	Arizona	1.40	150.00	25.00	300.00	7278717	2356.75	1500.00	4000.00	
4	Arkansas	3.02	175.00	75.00	400.00	3017804	1499.25	1000.00	2500.00	
5	California	1.09	250.75	50.00	600.00	39512223	5832.50	3500.00	9000.00	

Total rows: 50 of 50 | Query complete 00:00:00.055 | Ln 2, Col 1

SQL Output 32

2. Exported “competitors_revised” table (251 rows & 5 columns) to .csv file.

SELECT * FROM competitors_revised



	research_development_spent numeric	administration numeric	marketing_spent numeric	state_usa character varying	profit numeric
1		87164.9	87977	Alabama	120362.99
2	352053.99	87072.9	31951	Alabama	398021.99
3	933266.99	107860.9	51672	Alabama	1346449.99
4	989534.99	73696.9	59152	Alabama	1397050.99
5	1496277.99	104491.9	74515	Alabama	2430410.99
6	722966.99	129074.9	116675	Alaska	822050.99

Total rows: 251 of 251 | Query complete 00:00:00.078 | Ln 1, Col 1

SQL Output 33

3. Saved the 2 .csv files named “htsb_us_states” and “htsb_competitors_revised” under “Exported Data Output CSV File” folder.

SQL Figure 14

SQL Figure 15

V. **Saving PostgreSQL Codes:**

1. Saved the PostgreSQL Codes as .sql file named “SQL Queries”.

SQL Figure 16

MS Excel

I. MS Excel Tables and Pivot Tables Preparation:

1. Opened the htsb_us_states .csv file.

2. Modified the columns of the dataset in the original tab and autofit the column width.

3. Created “Sorted States”, “Pivot Tables”, and “Charts” tabs.

4. Sorted all the states by descending order according to corruption level, average healthcare spending, population size, average property price, and average income in the “Sorted States” tab.

5. Concatenated all the headers from the original tab for the “Sorted States” tab.

6. Applied conditional formatting to highlight top and bottom rows for the 1st to 4th sub-tables, as well as utilizing 3-color format style for the 5th sub-table in the “Sorted States” tab.

Note: Used LibreOffice Calc app to prepare and edit the tables and pivot tables.

Excel Figure 1

States by Corruption Rate	
Rhode Island	8.35
West Virginia	5.64
Louisiana	3.72
Tennessee	3.69
Oklahoma	3.23
Arkansas	3.02
Washington	2.53
Mississippi	2.43
Massachusetts	2.27
Alabama	2.15
New Mexico	2.14
South Carolina	2.04
Connecticut	2.01
New Jersey	1.90
Oregon	1.87
Texas	1.82
Florida	1.65
Nevada	1.63
Georgia	1.60
Kentucky	1.60
New York	1.59
Virginia	1.57
Missouri	1.50
Arizona	1.40
Maryland	1.38
Illinois	1.27
Pennsylvania	1.25
Utah	1.13
Idaho	1.12
California	1.09
Montana	1.09
Wisconsin	1.09
Delaware	1.08
Alaska	1.06
North Carolina	1.05
Wyoming	1.03
Kansas	1.02
Michigan	1.00
Indiana	0.90
Ohio	0.89
South Dakota	0.87
Colorado	0.80
Minnesota	0.68
Iowa	0.58
Nebraska	0.57
North Dakota	0.57
New Hampshire	0.51
Maine	0.48
Vermont	0.44
Hawaii	0.43

States by Average Healthcare Spending	
Hawaii	350.00
Massachusetts	350.00
New Jersey	350.00
Rhode Island	350.00
Alaska	300.25
Connecticut	300.00
Maryland	300.00
New Hampshire	300.00
Pennsylvania	300.00
Virginia	300.00
California	250.75
Nevada	250.75
Washington	250.75
Illinois	225.75
Maine	225.75
New York	225.75
Texas	225.75
Vermont	225.75
Colorado	225.50
Michigan	225.50
Oregon	225.50
Alabama	200.50
Indiana	200.25
Missouri	200.25
Ohio	200.25
Wisconsin	200.25
Georgia	200.00
Louisiana	200.00
North Carolina	200.00
South Carolina	200.00
Delaware	175.50
Kansas	175.50
Minnesota	175.50
Nebraska	175.50
Oklahoma	175.50
Utah	175.50
Florida	175.25
Tennessee	175.25
Arkansas	175.00
Kentucky	175.00
New Mexico	175.00
West Virginia	175.00
Arizona	150.00
Iowa	150.00
Mississippi	150.00
North Dakota	150.00
South Dakota	150.00
Idaho	125.50
Montana	125.50
Wyoming	125.50

States by Population Size	
California	39,512,223
Texas	28,995,881
Florida	21,477,737
New York	19,453,561
Pennsylvania	12,801,989
Illinois	12,671,821
Ohio	11,689,100
Georgia	10,617,423
North Carolina	10,488,084
Michigan	9,986,857
New Jersey	8,882,190
Virginia	8,535,519
Washington	7,614,893
Arizona	7,278,717
Massachusetts	6,892,503
Tennessee	6,829,174
Indiana	6,732,219
Missouri	6,137,428
Maryland	6,045,680
Wisconsin	5,822,434
Colorado	5,758,736
Minnesota	5,639,632
South Carolina	5,148,714
Alabama	4,903,185
Louisiana	4,648,794
Kentucky	4,467,673
Oregon	4,217,737
Oklahoma	3,956,971
Connecticut	3,565,287
Utah	3,205,958
Iowa	3,155,070
Nevada	3,080,156
Arkansas	3,017,804
Mississippi	2,976,149
Kansas	2,913,314
New Mexico	2,096,829
Nebraska	1,934,408
West Virginia	1,792,147
Idaho	1,787,065
Hawaii	1,415,872
New Hampshire	1,359,711
Maine	1,344,212
Montana	1,068,778
Rhode Island	1,059,361
Delaware	973,764
South Dakota	884,659
North Dakota	762,062
Alaska	731,545
Vermont	623,989
Wyoming	578,759

States by Average Property Price	
Hawaii	5,975.50
California	5,832.50
New York	4,968.75
New Jersey	4,672.25
Massachusetts	4,136.50
Connecticut	3,837.00
Rhode Island	3,458.25
Washington	3,458.25
Maryland	3,122.75
New Hampshire	3,122.75
Colorado	2,987.25
Florida	2,763.50
Alaska	2,684.00
Oregon	2,684.00
Nevada	2,503.00
Virginia	2,503.00
Arizona	2,356.75
Texas	2,356.75
Vermont	2,356.75
Delaware	2,289.75
Maine	2,236.00
Utah	2,236.00
Minnesota	2,195.50
Georgia	2,065.00
Illinois	2,056.50
North Carolina	2,056.50
Pennsylvania	2,056.50
Wisconsin	1,968.25
Wyoming	1,968.25
South Carolina	1,912.50
Michigan	1,812.75
Tennessee	1,812.75
Alabama	1,797.50
Ohio	1,797.50
Louisiana	1,718.50
Montana	1,705.00
Indiana	1,704.75
Missouri	1,677.75
Kentucky	1,605.00
North Dakota	1,505.50
Arkansas	1,499.25
Mississippi	1,499.25
New Mexico	1,499.25
Oklahoma	1,499.25
Kansas	1,479.50
Nebraska	1,459.25
Iowa	1,442.25
South Dakota	1,442.25
West Virginia	1,442.25
Idaho	1,382.00

States by Average Income	
Maryland	89,392.00
Massachusetts	82,427.00
New Jersey	81,740.00
California	80,440.00
Connecticut	79,287.00
New Hampshire	78,676.00
Hawaii	78,084.00
Washington	77,338.00
Alaska	76,440.00
Colorado	76,240.00
Virginia	75,417.00
Utah	72,558.00
Minnesota	72,027.00
Illinois	70,387.00
New York	70,137.00
Pennsylvania	65,135.00
Rhode Island	64,962.00
Delaware	64,040.00
Oregon	63,835.00
Wisconsin	63,795.00
North Dakota	63,715.00
Texas	63,656.00
Arizona	62,283.00
Iowa	62,075.00
Michigan	61,347.00
Wyoming	60,434.00
Nevada	60,106.00
Nebraska	59,929.00
Kansas	59,046.00
Georgia	58,932.00
Vermont	58,305.00
Florida	58,108.00
Indiana	57,881.00
Ohio	56,583.00
South Dakota	56,499.00
Missouri	55,685.00
Tennessee	55,107.00
Maine	54,927.00
Montana	54,875.00
North Carolina	54,560.00
Idaho	53,545.00
South Carolina	52,536.00
Oklahoma	51,424.00
Alabama	51,113.00
Louisiana	50,686.00
Kentucky	50,675.00
Arkansas	48,829.00
New Mexico	48,701.00
Mississippi	47,131.00
West Virginia	46,254.00

7. Created 4 Pivot Tables in the “Pivot Tables” tab.

Pivot Table # 1 @ Cell Range B2:C49:
Pivot Table Layout → Rows (**Corruption Rate**), Values (**Average of Average Income**)

Corruption Rate	AVERAGE of Average Income
0.43	78,084.00
0.44	58,305.00
0.48	54,927.00
0.51	78,676.00
0.57	61,822.00
0.58	62,075.00
0.68	72,027.00
0.80	76,240.00
0.87	56,499.00
0.89	56,583.00
0.90	57,881.00
1.00	61,347.00
1.02	59,046.00
1.03	60,434.00
1.05	54,560.00
1.06	76,440.00
1.08	64,040.00
1.09	66,370.00
1.12	53,545.00
1.13	72,558.00
1.25	65,135.00
1.27	70,387.00
1.38	89,392.00
1.40	62,283.00
1.50	55,685.00
1.57	75,417.00
1.59	70,137.00
1.60	54,803.50
1.63	60,106.00
1.65	58,108.00
1.82	63,656.00
1.87	63,835.00
1.90	81,740.00
2.01	79,287.00
2.04	52,536.00
2.14	48,701.00
2.15	51,113.00
2.27	82,427.00
2.43	47,131.00
2.53	77,338.00
3.02	48,829.00
3.23	51,424.00
3.69	55,107.00
3.72	50,686.00
5.64	46,254.00
8.35	64,962.00
Grand Total	**63,146.08**

Excel Figure 2

Pivot Table # 2 @ Cell Range E2:F53:
Pivot Table Layout → Rows (**States**), Values (**Sum of Average Income**)

States	SUM of Average Income
Maryland	89,392.00
Massachusetts	82,427.00
New Jersey	81,740.00
California	80,440.00
Connecticut	79,287.00
New Hampshire	78,676.00
Hawaii	78,084.00
Washington	77,338.00
Alaska	76,440.00
Colorado	76,240.00
Virginia	75,417.00
Utah	72,558.00
Minnesota	72,027.00
Illinois	70,387.00
New York	70,137.00
Pennsylvania	65,135.00
Rhode Island	64,962.00
Delaware	64,040.00
Oregon	63,835.00
Wisconsin	63,795.00
North Dakota	63,715.00
Texas	63,656.00
Arizona	62,283.00
Iowa	62,075.00
Michigan	61,347.00
Wyoming	60,434.00
Nevada	60,106.00
Nebraska	59,929.00
Kansas	59,046.00
Georgia	58,932.00
Vermont	58,305.00
Florida	58,108.00
Indiana	57,881.00
Ohio	56,583.00
South Dakota	56,499.00
Missouri	55,685.00
Tennessee	55,107.00
Maine	54,927.00
Montana	54,875.00
North Carolina	54,560.00
Idaho	53,545.00
South Carolina	52,536.00
Oklahoma	51,424.00
Alabama	51,113.00
Louisiana	50,686.00
Kentucky	50,675.00
Arkansas	48,829.00
New Mexico	48,701.00
Mississippi	47,131.00
West Virginia	46,254.00
Grand Total	**3,157,304.00**

Excel Figure 3

Pivot Table # 3 @ Cell Range H2:I17:
Pivot Table Layout → Rows(**Average Healthcare Spending**), Values (**Average of Average Income**)

Average Healthcare Spending	AVERAGE of Average Income
125.50	56,284.67
150.00	58,340.60
175.00	48,614.75
175.25	56,607.50
175.50	63,170.67
200.00	54,178.50
200.25	58,486.00
200.50	51,113.00
225.50	67,140.67
225.75	63,482.40
250.75	72,628.00
300.00	77,581.40
300.25	76,440.00
350.00	76,803.25
Grand Total	**63,146.08**

Excel Figure 4

Pivot Table # 4 @ Cell Range K2:L36:
Pivot Table Layout → Rows(**Average Property Price)**, Values(**Average of Average Income**)

Average Property Price	AVERAGE of Average Income
1,382.00	53,545.00
1,442.25	54,942.67
1,459.25	59,929.00
1,479.50	59,046.00
1,499.25	49,021.25
1,505.50	63,715.00
1,605.00	50,675.00
1,677.75	55,685.00
1,704.75	57,881.00
1,705.00	54,875.00
1,718.50	50,686.00
1,797.50	53,848.00
1,812.75	58,227.00
1,912.50	52,536.00
1,968.25	62,114.50
2,056.50	63,360.67
2,065.00	58,932.00
2,195.50	72,027.00
2,236.00	63,742.50
2,289.75	64,040.00
2,356.75	61,414.67
2,503.00	67,761.50
2,684.00	70,137.50
2,763.50	58,108.00
2,987.25	76,240.00
3,122.75	84,034.00
3,458.25	71,150.00
3,837.00	79,287.00
4,136.50	82,427.00
4,672.25	81,740.00
4,968.75	70,137.00
5,832.50	80,440.00
5,975.50	78,084.00
Grand Total	**63,146.08**

Excel Figure 5

8. Created 3 Tables in the “Pivot Tables” tab.

Table # 1 @ Cell Range B51:C53:
Table Name → Correlation between Average Income & Corruption Level
Used CORREL function to calculate the correlation value.
Used Nested IF function to show the remarks of the correlation value.

Correlation between Average Income & Corruption Level	
Correlation Value:	-0.2262823101
Remarks:	No Correlation

Excel Figure 6

Table # 2 @ Cell Range H19:I21:
Table Name → Correlation between Average Healthcare Spending & Average Income
Used CORREL function to calculate the correlation value.
Used Nested IF function to show the remarks of the correlation value.

Correlation between Average Healthcare Spending & Average Income	
Correlation Value:	0.7040445808
Remarks:	High Positive Correlation

Excel Figure 7

Table # 3 @ Cell Range K38:L40:
Table Name → Correlation between Average Property Price & Average Income
Used CORREL function to calculate the correlation value.
Used Nested IF function to show the remarks of the correlation value.

Correlation between Average Property Price & Average Income	
Correlation Value:	0.7394533981
Remarks:	High Positive Correlation

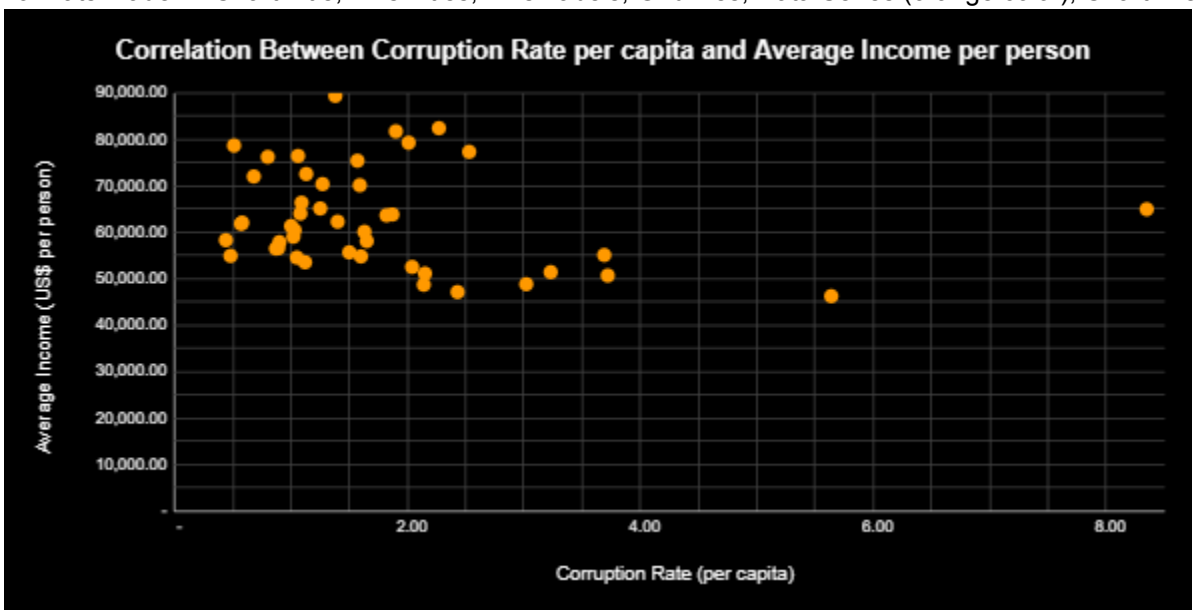
Excel Figure 8

II. MS Excel Charts:

1. Created 3 Scatterplots and 1 Bar Chart in the “Charts” tab.

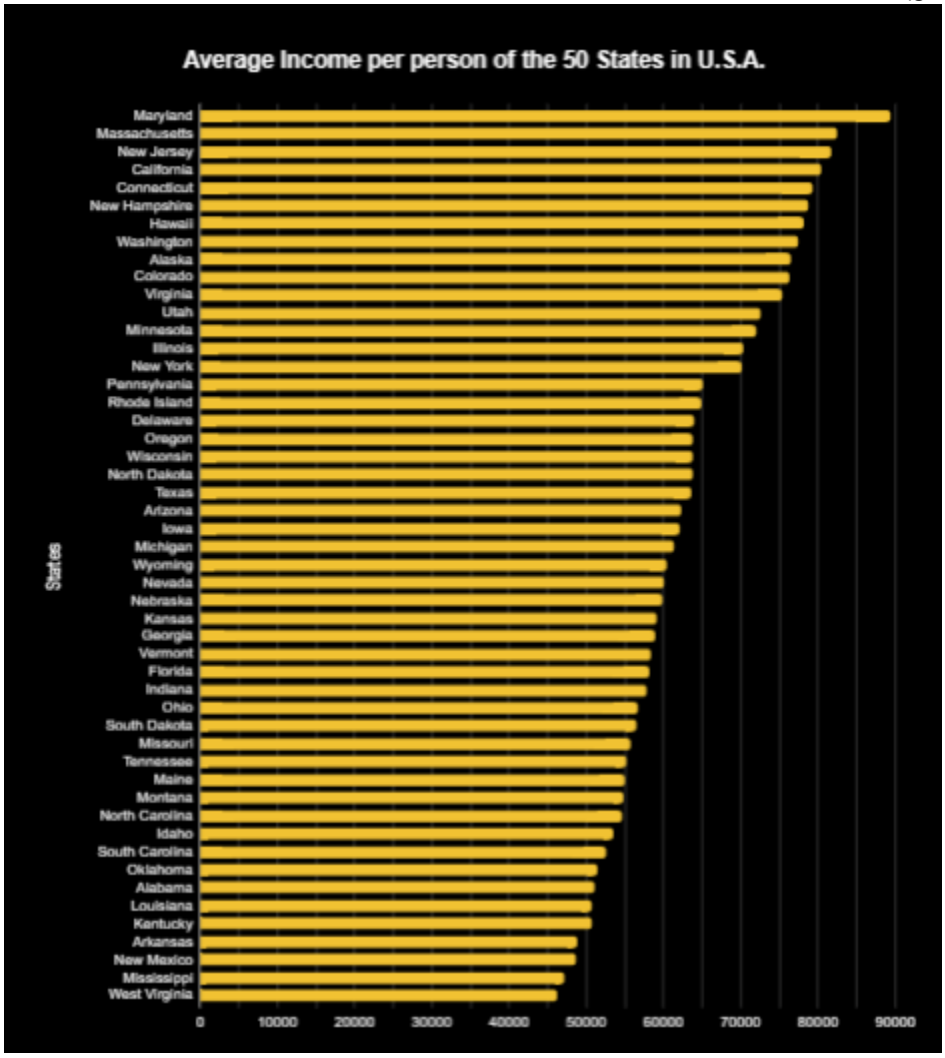
Note: Used Google Sheets to generate and modify the 4 charts.

Chart # 1 @ Cell A1:
Chart Name → Correlation Between Corruption Rate per capita and Average Income per person
Chart Type → Scatter Plot
Formats made → Chart Title, Axis Titles, Axis Labels, Gridlines, Data Series (orange color), Chart Area (dark background)



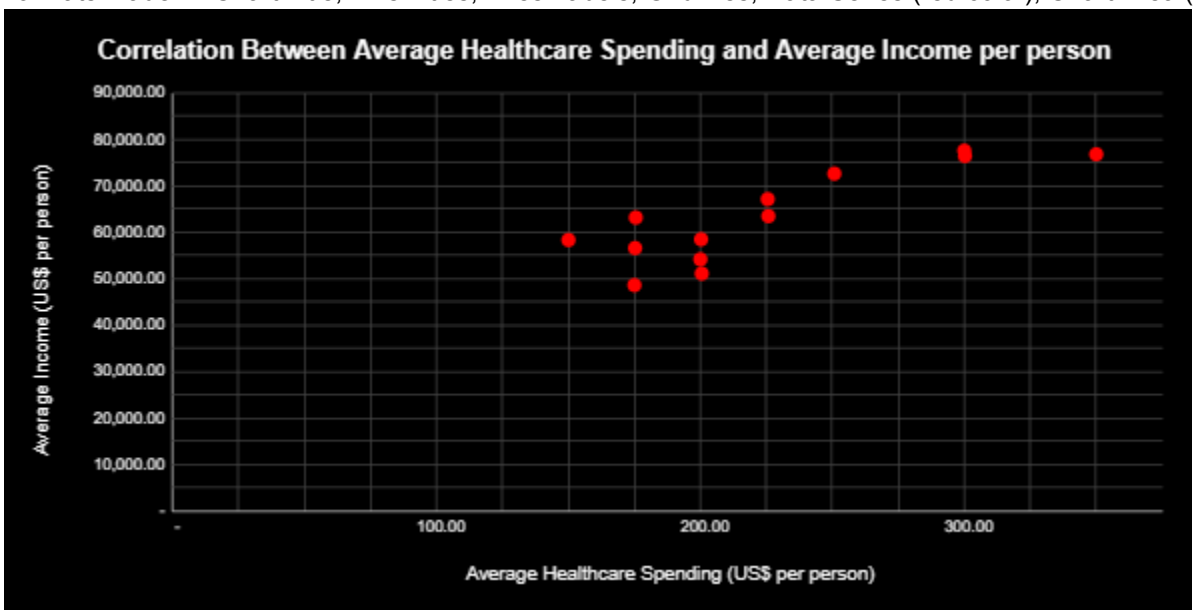
Excel Chart 1

Chart # 2 @ Cell I1:
Chart Name → Average Income per person of the 50 States in U.S.A.
Chart Type → Bar Chart
Formats made → Chart Title, Axis Titles, Axis Labels, Gridlines, Data Series (gold color), Chart Area (dark background)



Excel Chart 2

Chart # 3 @ Cell Q1:
Chart Name → Correlation Between Average Healthcare Spending and Average Income per person
Chart Type → Scatter Plot
Formats made → Chart Title, Axis Titles, Axes Labels, Gridlines, Data Series (red color), Chart Area (dark background)



Excel Chart 3

Chart # 4 @ Cell Q24:
Chart Name → Correlation Between Average Property Price and Average Income per person
Chart Type → Scatter Plot
Formats made → Chart Title, Axis Titles, Axes Labels, Gridlines, Data Series (green color), Chart Area (dark background)

Excel Chart 4

III. Data Saving Process in MS Excel:

1. Saved the analyzed “htsb_us_states” .csv file as .xlsx file named “Excel Pivot Tables & Charts”.

Excel Figure 9

Python

I. Data Loading Process in Google Colab:

1. Imported the necessary libraries, modules, functions, and magic commands.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as sp
import statsmodels as sm
import sklearn as sk
import datetime
from itertools import combinations
from scipy.stats import ttest_1samp, ttest_ind, ttest_rel, chisquare
from statsmodels.stats.weightstats import ztest
from statsmodels.stats.proportion import binom_test, proportions_ztest
%matplotlib inline
```

2. Loaded the 2 .csv files datasets “htsb_competitors_revised” & “htsb_us_states” under a new notebook in Google Colab.

```
us_states = pd.read_csv("https://drive.google.com/uc?export=download&id=15EvnChUJKeIT6DkukoxauieDFaH6s_kJ")
```

```
competitors_revised =
pd.read_csv("https://drive.google.com/uc?export=download&id=135zvjPO0fkoxETwbW4FZIuqpHfx2ek13")
```

3. Checked if the datasets were loaded properly, without any duplicates, missing values, or errors.

```
display(us_states.head(10))
print("-----------------------------------------------------")
display(us_states.info())
print("-----------------------------------------------------")
print("Dataframe Dimensions:", us_states.shape)
```

Python Output 1

	state_usa	convictions_per_capita	avg_spending	min_spending	max_spending	estimate	avg_price	min_price	max_price	average_income	minimum_income	maximum_income
0	Alabama	2.15	200.50	50	500	4903185	1797.50	1200	2500	51113	23999	96993
1	Alaska	1.06	300.25	100	750	731545	2684.00	2000	3500	76440	35219	134318
2	Arizona	1.40	150.00	25	300	7278717	2356.75	1500	4000	62283	29466	113589
3	Arkansas	3.02	175.00	75	400	3017804	1499.25	1000	2500	48829	23028	90052
4	California	1.09	250.75	50	600	39512223	5832.50	3500	9000	80440	37698	149265
5	Colorado	0.80	225.50	100	500	5758736	2987.25	2000	4500	76240	35636	130714
6	Connecticut	2.01	300.00	150	700	3565287	3837.00	3000	5500	79287	37426	142596
7	Delaware	1.08	175.50	75	350	973764	2289.75	1500	3500	64040	30544	120324
8	Florida	1.65	175.25	50	400	21477737	2763.50	1500	5000	58108	27064	105773
9	Georgia	1.60	200.00	75	450	10617423	2065.00	1200	3500	58932	27609	112609

```
--------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 12 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   state_usa              50 non-null     object
 1   convictions_per_capita  50 non-null     float64
 2   avg_spending           50 non-null     float64
 3   min_spending           50 non-null     int64
 4   max_spending           50 non-null     int64
 5   estimate               50 non-null     int64
 6   avg_price              50 non-null     float64
 7   min_price              50 non-null     int64
 8   max_price              50 non-null     int64
 9   average_income         50 non-null     int64
 10  minimum_income         50 non-null     int64
 11  maximum_income         50 non-null     int64
dtypes: float64(3), int64(8), object(1)
memory usage: 4.8+ KB
None
```

```
display(us_states[us_states.duplicated])
```

Python Output 2

state_usa	convictions_per_capita	avg_spending	min_spending	max_spending	estimate	avg_price	min_price	max_price	average_income	minimum_income	maximum_income

```
display(us_states.isna().sum())
```

```
state_usa              0
convictions_per_capita    0
avg_spending           0
min_spending           0
max_spending           0
estimate               0
avg_price              0
min_price              0
max_price              0
average_income         0
minimum_income         0
maximum_income         0
dtype: int64
```

Python Output 3

```
display(competitors_revised.head(10))
print("--------------------------------------------------")
display(competitors_revised.info())
print("--------------------------------------------------")
print("Dataframe Dimensions:", competitors_revised.shape)
```

	research_development_spent	administration	marketing_spent	state_usa	profit
0	901313.99	87164.9	87977.0	Alabama	120362.99
1	352053.99	87072.9	31951.0	Alabama	398021.99
2	933266.99	107860.9	51672.0	Alabama	1346449.99
3	989534.99	73696.9	59152.0	Alabama	1397050.99
4	1496277.99	104491.9	74515.0	Alabama	2430410.99
5	722966.99	129074.9	116675.0	Alaska	822050.99
6	155083.99	128398.9	63666.0	Alaska	1866877.99
7	1155299.99	51488.9	93013.0	Alaska	2102288.99
8	1172907.99	68409.9	101154.0	Alaska	2167641.99
9	1116726.99	127485.9	81283.0	Alaska	2886772.99

```
------------------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 251 entries, 0 to 250
Data columns (total 5 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   research_development_spent  251 non-null    float64
 1   administration              251 non-null    float64
 2   marketing_spent             251 non-null    float64
 3   state_usa                   251 non-null    object
 4   profit                      251 non-null    float64
dtypes: float64(4), object(1)
memory usage: 9.9+ KB
None
------------------------------------------------------------
Dataframe Dimensions: (251, 5)
```

Python Output 4

```
display(competitors_revised[competitors_revised.duplicated])
```

	research_development_spent	administration	marketing_spent	state_usa	profit

Python Output 5

```
display(competitors_revised.isna().sum())
```

```
research_development_spent    0
administration                0
marketing_spent               0
state_usa                     0
profit                        0
dtype: int64
```

Python Output 6

II. Data Transformation Using Python:

1. Renamed column names to make it more understandable.

```
print("Total Number of Columns:", len(us_states.columns))
for ind, column in enumerate(us_states.columns):
  print(ind + 1, column)
```

```
Total Number of Columns: 12
1 state_usa
2 convictions_per_capita
3 avg_spending
4 min_spending
5 max_spending
6 estimate
7 avg_price
8 min_price
9 max_price
10 average_income
11 minimum_income
12 maximum_income
```

Python Output 7

```
us_states = us_states.rename(columns=
 {
    "state_usa" : "States", "convictions_per_capita" : "Corruption Rate", "avg_spending" : "Average Healthcare Spending", "min_spending" : "Minimum Healthcare Spending",
     "max_spending" : "Maximum Healthcare Spending", "estimate" : "Population Size", "avg_price" : "Average Property Price", "min_price" : "Minimum Property Price",
      "max_price" : "Maximum Property Price", "average_income" : "Average Income", "minimum_income" : "Minimum Income", "maximum_income" : "Maximum Income"
 }
)
for i in us_states.columns:
 print(i)
```

```
States
Corruption Rate
Average Healthcare Spending
Minimum Healthcare Spending
Maximum Healthcare Spending
Population Size
Average Property Price
Minimum Property Price
Maximum Property Price
Average Income
Minimum Income
Maximum Income
```

Python Output 8

```
print("Total Number of Columns:", len(competitors_revised.columns))
for ind, column in enumerate(competitors_revised.columns):
 print(ind + 1, column)
```

```
Total Number of Columns: 5
1 research_development_spent
2 administration
3 marketing_spent
4 state_usa
5 profit
```

Python Output 9

```
competitors_revised = competitors_revised.rename(columns=
 {
    "research_development_spent" : "Research & Development Expenses", "administration" : "Salary & Wages Expenses", "marketing_spent" : "Marketing Expenses", "state_usa" : "States", "profit" : "Profit"
 }
)
for i in competitors_revised.columns:
 print(i)
```

```
Research & Development Expenses
Salary & Wages Expenses
Marketing Expenses
States
Profit
```

Python Output 10

2. Changed the data types of columns containing money values from integer to float.

```
us_states = us_states.astype(
 {
   "Minimum Healthcare Spending" : "float", "Maximum Healthcare Spending" : "float", "Minimum Property Price" : "float",
"Maximum Property Price" : "float",
   "Average Income" : "float", "Minimum Income" : "float", "Maximum Income" : "float"
 }
)
display(us_states.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 12 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   States                       50 non-null     object
 1   Corruption Rate              50 non-null     float64
 2   Average Healthcare Spending  50 non-null     float64
 3   Minimum Healthcare Spending  50 non-null     float64
 4   Maximum Healthcare Spending  50 non-null     float64
 5   Population Size              50 non-null     int64
 6   Average Property Price       50 non-null     float64
 7   Minimum Property Price       50 non-null     float64
 8   Maximum Property Price       50 non-null     float64
 9   Average Income               50 non-null     float64
 10  Minimum Income               50 non-null     float64
 11  Maximum Income               50 non-null     float64
dtypes: float64(10), int64(1), object(1)
memory usage: 4.8+ KB
None
```

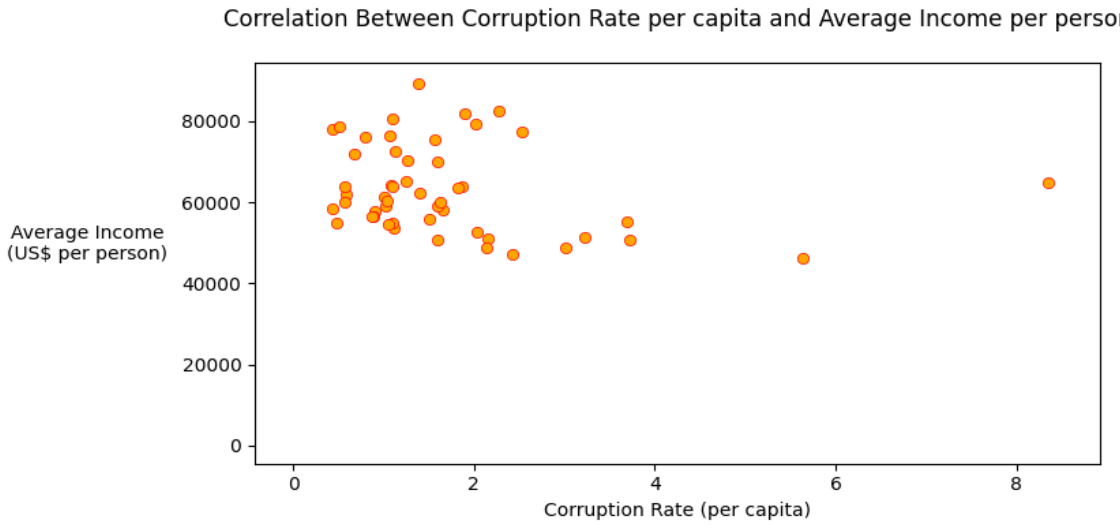
Python Output 11

III. Data Analysis & Visualization Process in Python:

1. Recreated the Excel Scatterplot in Python for checking.

```
aicr_sp_xscales = np.array([0,8.5])
aicr_sp_yscales = np.array([0,90000])
```

```
plt.figure(figsize=(8,4))
plt.plot(aicr_sp_xscales, aicr_sp_yscales, alpha=0)
sns.scatterplot(data=us_states, x="Corruption Rate", y="Average Income", color="orange",
edgecolor="red").set_title("Correlation Between Corruption Rate per capita and Average Income per person",
loc="center", pad=20)
plt.xlabel("Corruption Rate (per capita)")
plt.ylabel("Average Income\n(US$ per person)", rotation=0, labelpad=50)
plt.show()
```

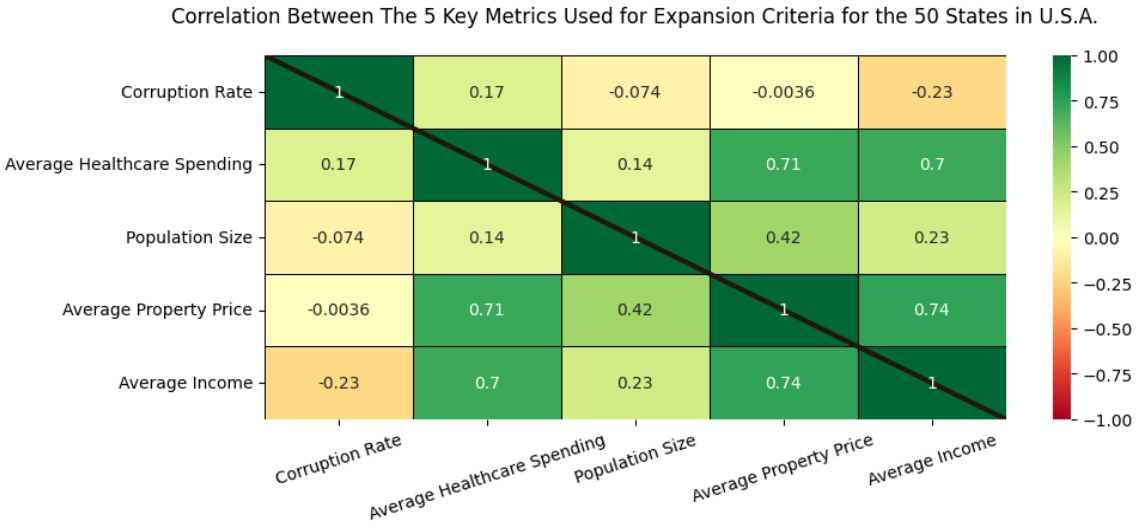


Python Chart 1

2. Created a Heatmap to summarize the correlation between the 5 key metrics used as a basis of criteria to consider expanding the business operations.

```
us_states_hm_xvalues = np.array([0,5])
us_states_hm_yvalues = np.array([0,5])
```

```
plt.figure(figsize=(10,4))
plt.plot(us_states_hm_xvalues, us_states_hm_yvalues, color="#231709", linewidth=3)
sns.heatmap(us_states[["Corruption Rate", "Average Healthcare Spending", "Population Size", "Average Property Price",
"Average Income"]].corr(), vmin=-1, vmax=1, cmap="RdYlGn", annot=True, linewidth=0.5, linecolor="black")
plt.title("Correlation Between The 5 Key Metrics Used for Expansion Criteria for the 50 States in U.S.A.", loc="center",
pad=20)
plt.xticks(rotation=18)
plt.show()
```



Python Chart 2

```
def correlation_remarks(r):
 if r >= 0.90:
   print("Very High Positive Correlation")
 elif r >= 0.70:
   print("High Positive Correlation")
 elif r >= 0.50:
   print("Moderate Positive Correlation")
 elif r >= 0.30:
   print("Low Positive Correlation")
 elif r > -0.30:
   print("No Correlation")
 elif r > -0.50:
   print("Low Negative Correlation")
 elif r > -0.70:
   print("Moderate Negative Correlation")
 elif r > -0.90:
   print("High Negative Correlation")
 else:
   print("Very High Negative Correlation")
```

```
metrics_list = ["Corruption Rate", "Average Healthcare Spending", "Population Size", "Average Property Price", "Average Income"]
metrics_combinations = list(combinations(metrics_list, 2))
print("Number of Combinations:", len(metrics_combinations))
for ind, combo in enumerate(metrics_combinations):
   print(ind + 1, combo)
```

```
Number of Combinations: 10
1 ('Corruption Rate', 'Average Healthcare Spending')
2 ('Corruption Rate', 'Population Size')
3 ('Corruption Rate', 'Average Property Price')
4 ('Corruption Rate', 'Average Income')
5 ('Average Healthcare Spending', 'Population Size')
6 ('Average Healthcare Spending', 'Average Property Price')
7 ('Average Healthcare Spending', 'Average Income')
8 ('Population Size', 'Average Property Price')
9 ('Population Size', 'Average Income')
10 ('Average Property Price', 'Average Income')
```

Python Output 12

```
print("Corruption Rate vs Average Healthcare Spending:")
CorruptionRate_vs_AverageHealthcareSpending = correlation_remarks(r = 0.17)
print("-------------------------------------------------")
print("Corruption Rate vs Population Size:")
CorruptionRate_vs_PopulationSize = correlation_remarks(r = -0.074)
print("-------------------------------------------------")
print("Corruption Rate vs Average Property Price:")
CorruptionRate_vs_AveragePropertyPrice = correlation_remarks(r = -0.0036)
print("-------------------------------------------------")
print("Corruption Rate vs Average Income:")
CorruptionRate_vs_AverageIncome = correlation_remarks(r = -0.23)
print("-------------------------------------------------")
print("Average Healthcare Spending vs Population Size:")
AverageHealthcareSpending_vs_PopulationSize = correlation_remarks(r = 0.14)
```

```
print("-------------------------------------------------")
print("Average Healthcare Spending vs Average Property Price:")
AverageHealthcareSpending_vs_AveragePropertyPrice = correlation_remarks(r = 0.71)
print("-------------------------------------------------")
print("Average Healthcare Spending vs Average Income:")
AverageHealthcareSpending_vs_AverageIncome = correlation_remarks(r = 0.70)
print("-------------------------------------------------")
print("Population Size vs Average Property Price:")
PopulationSize_vs_AveragePropertyPrice = correlation_remarks(r = 0.42)
print("-------------------------------------------------")
print("Population Size vs Average Income:")
PopulationSize_vs_AverageIncome = correlation_remarks(r = 0.23)
print("-------------------------------------------------")
print("Average Property Price vs Average Income:")
AveragePropertyPrice_vs_AverageIncome = correlation_remarks(r = 0.74)
```

```
Corruption Rate vs Average Healthcare Spending:
No Correlation
-------------------------------------------------
Corruption Rate vs Population Size:
No Correlation
-------------------------------------------------
Corruption Rate vs Average Property Price:
No Correlation
-------------------------------------------------
Corruption Rate vs Average Income:
No Correlation
-------------------------------------------------
Average Healthcare Spending vs Population Size:
No Correlation
-------------------------------------------------
Average Healthcare Spending vs Average Property Price:
High Positive Correlation
-------------------------------------------------
Average Healthcare Spending vs Average Income:
High Positive Correlation
-------------------------------------------------
Population Size vs Average Property Price:
Low Positive Correlation
-------------------------------------------------
Population Size vs Average Income:
No Correlation
-------------------------------------------------
Average Property Price vs Average Income:
High Positive Correlation
```

Python Output 13

3. Summarized the statistics of the Average Incomes (US$ per person) of the 50 states in U.S.A.

```
print("Frequencies:")
us_states["Average Income"].value_counts()
```

```
Frequencies:
51113.0    1
65135.0    1
60106.0    1
78676.0    1
81740.0    1
48701.0    1
70137.0    1
54560.0    1
63715.0    1
56583.0    1
51424.0    1
63835.0    1
64962.0    1
76440.0    1
52536.0    1
56499.0    1
55107.0    1
63656.0    1
72558.0    1
58305.0    1
75417.0    1
77338.0    1
46254.0    1
63795.0    1
59929.0    1
```

Python Output 14

```
def modal_value(mode):
 if len(mode) == 1:
   print("Mode:", mode[0])
```

```
print("Descriptive Statistics:")
print("Count:", us_states["Average Income"].count())
print("Distinct Count:", us_states["Average Income"].nunique())
print("Sum:", us_states["Average Income"].sum())
print("Minimum:", us_states["Average Income"].min())
print("Maximum:", us_states["Average Income"].max())
print("Average:", us_states["Average Income"].mean())
print("Median:", us_states["Average Income"].median())
print("Mode:", modal_value(us_states["Average Income"]))
print("Lower Quartile:", us_states["Average Income"].quantile(0.25))
print("Upper Quartile:", us_states["Average Income"].quantile(0.75))
print("Range:", us_states["Average Income"].max() - us_states["Average Income"].min())
print("Interquartile Range:", us_states["Average Income"].quantile(0.75) - us_states["Average Income"].quantile(0.25))
print("Standard Deviation:", us_states["Average Income"].std())
print("Variance:", us_states["Average Income"].std() ** 2)
print("Mean Absolute Deviation:", us_states["Average Income"].mad())
print("Skewness:", us_states["Average Income"].skew(axis=0))
print("Kurtosis:", us_states["Average Income"].kurtosis(axis=0))
print("----------------------------------------")
```

```
Descriptive Statistics:
Count: 50
Distinct Count: 50
Sum: 3157304.0
Minimum: 46254.0
Maximum: 89392.0
Average: 63146.08
Median: 60890.5
Mode: None
Lower Quartile: 54972.0
Upper Quartile: 71617.0
Range: 43138.0
Interquartile Range: 16645.0
Standard Deviation: 10806.597134441648
Variance: 116782541.62612244
Mean Absolute Deviation: 8820.569599999999
Skewness: 0.5328861882367197
Kurtosis: -0.6366284360693002
----------------------------------------
```

Python Output 15

```
print("Mean Absolute Deviation (Lower Threshold):", us_states["Average Income"].mean() - 2 * us_states["Average Income"].mad())
print("Mean Absolute Deviation (Upper Threshold):", us_states["Average Income"].mean() + 2 * us_states["Average Income"].mad())
print("Standard Deviation (Lower Threshold):", us_states["Average Income"].mean() - 2 * us_states["Average Income"].std())
print("Standard Deviation (Upper Threshold):", us_states["Average Income"].mean() + 2 * us_states["Average Income"].std())
print("Tukey's Fence Method (Lower Fence):", us_states["Average Income"].quantile(0.25) - 1.5 * (us_states["Average Income"].quantile(0.75) - us_states["Average Income"].quantile(0.25)))
print("Tukey's Fence Method (Upper Fence):", us_states["Average Income"].quantile(0.75) + 1.5 * (us_states["Average Income"].quantile(0.75) - us_states["Average Income"].quantile(0.25)))
```

```
Mean Absolute Deviation (Lower Threshold): 45504.940800000004
Mean Absolute Deviation (Upper Threshold): 80787.21919999999
Standard Deviation (Lower Threshold): 41532.88573111671
Standard Deviation (Upper Threshold): 84759.2742688833
Tukey's Fence Method (Lower Fence): 30004.5
Tukey's Fence Method (Upper Fence): 96584.5
```

Python Output 16

```
print("Extreme Values (Mean Absolute Deviation):", us_states["Average Income"][~((us_states["Average Income"] >= 45504.940800000004) & (us_states["Average Income"] <= 80787.21919999999))].count())
print("Extreme Values (Standard Deviation):", us_states["Average Income"][~((us_states["Average Income"] >= 41532.88573111671) & (us_states["Average Income"] <= 84759.2742688833))].count())
print("Extreme Values (Tukey's Fence Method):", us_states["Average Income"][~((us_states["Average Income"] >= 30004.5) & (us_states["Average Income"] <= 96584.5))].count())
```

```
Extreme Values (Mean Absolute Deviation): 3
Extreme Values (Standard Deviation): 1
Extreme Values (Tukey's Fence Method): 0
```

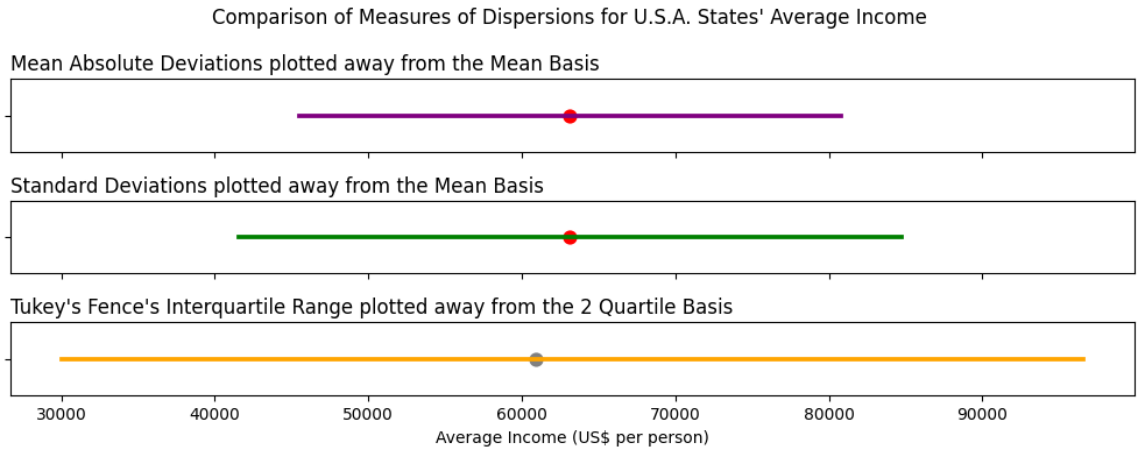
Python Output 17

```
us_ai_pp_thresholds = pd.DataFrame(
 {
  "class" : ["lower threshold", "basis", "upper threshold"],
  "mad" : ["45504.940800000004", "63146.08", "80787.21919999999"],
  "std" : ["41532.88573111671", "63146.08", "84759.2742688833"],
  "tfm" : ["30004.5", "60890.5", "96584.5"]
 }
)
us_ai_pp_thresholds = us_ai_pp_thresholds.set_index("class").astype("float")
display(us_ai_pp_thresholds)
```

class	mad	std	tfm
lower threshold	45504.9408	41532.885731	30004.5
basis	63146.0800	63146.080000	60890.5
upper threshold	80787.2192	84759.274269	96584.5

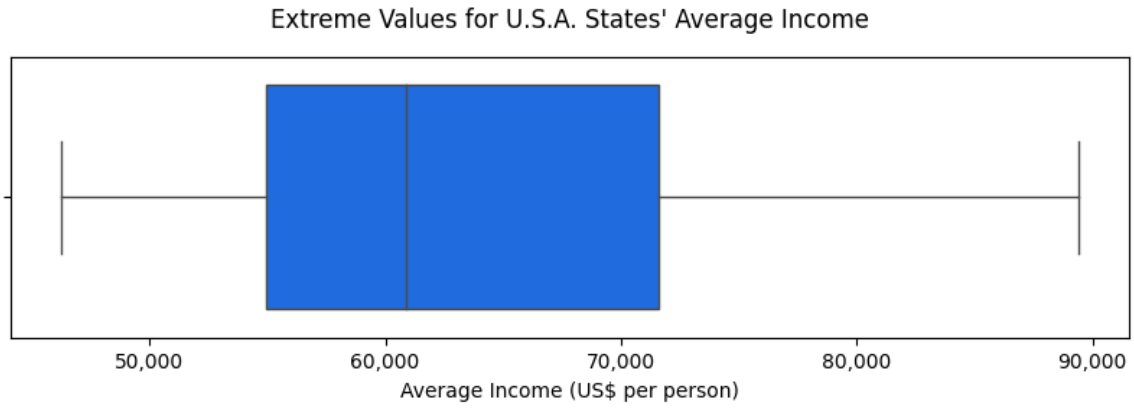
Python Output 18

```
fig, ax = plt.subplots(3,1, figsize=(10,4), sharex=True)
plt.suptitle("Comparison of Measures of Dispersions for U.S.A. States' Average Income")
plt.subplot(3,1,1)
sns.pointplot(data=us_ai_pp_thresholds, x="mad", estimator="mean", errorbar=("ci", 95), n_boot=1000, markers='|',
linestyles='-', scale=0.5, color="purple", ax=ax[0])
sns.pointplot(data=us_states, x="Average Income", estimator="mean", errorbar=None, n_boot=1000, markers='o',
linestyles='-', scale=1, color="red", ax=ax[0])
ax[0].set_title("Mean Absolute Deviations plotted away from the Mean Basis", loc="left")
plt.xlabel(None)
plt.subplot(3,1,2)
sns.pointplot(data=us_ai_pp_thresholds, x="std", estimator="mean", errorbar=("ci", 95), n_boot=1000, markers='|',
linestyles='-', scale=0.5, color="green", ax=ax[1])
sns.pointplot(data=us_states, x="Average Income", estimator="mean", errorbar=None, n_boot=1000, markers='o',
linestyles='-', scale=1, color="red", ax=ax[1])
ax[1].set_title("Standard Deviations plotted away from the Mean Basis", loc="left")
plt.xlabel(None)
plt.subplot(3,1,3)
sns.pointplot(data=us_ai_pp_thresholds, x="tfm", estimator="median", errorbar=("ci", 95), n_boot=1000, markers='|',
linestyles='-', scale=0.5, color="orange", ax=ax[2])
sns.pointplot(data=us_states, x="Average Income", estimator="median", errorbar=None, n_boot=1000, markers='o',
linestyles='-', scale=1, color="grey", ax=ax[2])
ax[2].set_title("Tukey's Fence's Interquartile Range plotted away from the 2 Quartile Basis", loc="left")
plt.xlabel("Average Income (US$ per person)")
plt.tight_layout()
plt.show()
```



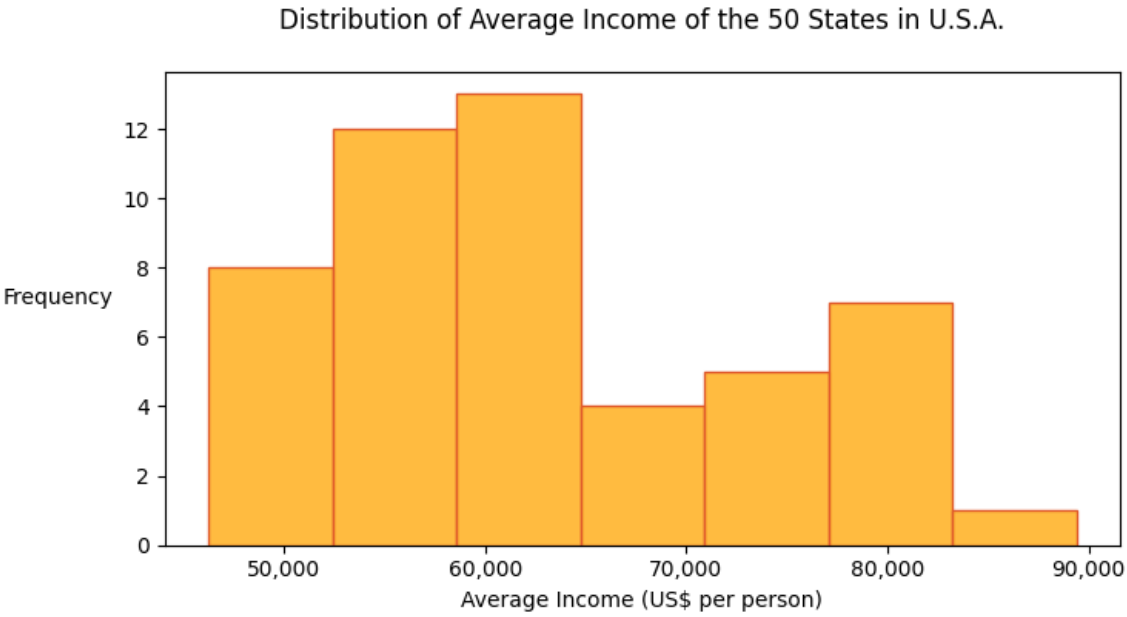
Python Chart 3

```
plt.figure(figsize=(10,2.5))
ax = sns.boxplot(data=us_states, x="Average Income", color="#0066FF", width=0.8, fliersize=5, linewidth=1, whis=1.5)
ax.set_title("Extreme Values for U.S.A. States' Average Income", pad=15)
ax.set_xlabel("Average Income (US$ per person)")
tick_labels = [f"{tick:,.0f}" for tick in ax.get_xticks()]
ax.set_xticklabels(tick_labels)
plt.show()
```



Python Chart 4

```
plt.figure(figsize=(8,4))
ax = sns.histplot(data=us_states, x="Average Income", bins="auto", shrink=1, color="#FFA500", edgecolor=(0.88,0.35,0.17))
ax.set_title("Distribution of Average Income of the 50 States in U.S.A.", pad=20)
ax.set_xlabel("Average Income (US$ per person)")
ax.set_ylabel("Frequency", rotation=0, labelpad=30)
tick_labels = [f"{tick:,.0f}" for tick in ax.get_xticks()]
ax.set_xticklabels(tick_labels)
plt.show()
```



Python Chart 5

4. Summarized the statistics of Corruption Rates (per capita) of the 50 states in U.S.A.

```
print("Frequencies:")
us_states["Corruption Rate"].value_counts()
```

```
Frequencies:
1.09    3
1.60    2
0.57    2
2.15    1
1.25    1
0.51    1
1.90    1
2.14    1
1.59    1
1.05    1
0.89    1
3.23    1
1.87    1
2.04    1
8.35    1
0.87    1
3.69    1
1.82    1
1.13    1
0.44    1
1.57    1
2.53    1
5.64    1
1.63    1
1.50    1
```

Python Output 19

```
print("Descriptive Statistics:")
print("Count:", us_states["Corruption Rate"].count())
print("Distinct Count:", us_states["Corruption Rate"].nunique())
print("Sum:", us_states["Corruption Rate"].sum())
print("Minimum:", us_states["Corruption Rate"].min())
print("Maximum:", us_states["Corruption Rate"].max())
print("Average:", us_states["Corruption Rate"].mean())
print("Median:", us_states["Corruption Rate"].median())
print("Mode:", list(us_states["Corruption Rate"].mode()))
print("Lower Quartile:", us_states["Corruption Rate"].quantile(0.25))
print("Upper Quartile:", us_states["Corruption Rate"].quantile(0.75))
print("Range:", us_states["Corruption Rate"].max() - us_states["Corruption Rate"].min())
print("Interquartile Range:", us_states["Corruption Rate"].quantile(0.75) - us_states["Corruption Rate"].quantile(0.25))
print("Standard Deviation:", us_states["Corruption Rate"].std())
print("Variance:", us_states["Corruption Rate"].std() ** 2)
print("Mean Absolute Deviation:", us_states["Corruption Rate"].mad())
print("Skewness:", us_states["Corruption Rate"].skew(axis=0))
print("Kurtosis:", us_states["Corruption Rate"].kurtosis(axis=0))
print("----------------------------------------")
```

```
Descriptive Statistics:
Count: 50
Distinct Count: 46
Sum: 84.73000000000002
Minimum: 0.43
Maximum: 8.35
Average: 1.6946000000000003
Median: 1.325
Mode: [1.09]
Lower Quartile: 1.005
Upper Quartile: 1.9825
Range: 7.92
Interquartile Range: 0.9775
Standard Deviation: 1.3771194574182735
Variance: 1.896458
Mean Absolute Deviation: 0.8678560000000002
Skewness: 2.945607896440458
Kurtosis: 11.336706778526779
----------------------------------------
```

Python Output 20

```
print("Mean Absolute Deviation (Lower Threshold):", us_states["Corruption Rate"].mean() - 2 * us_states["Corruption Rate"].mad())
print("Mean Absolute Deviation (Upper Threshold):", us_states["Corruption Rate"].mean() + 2 * us_states["Corruption Rate"].mad())
print("Standard Deviation (Lower Threshold):", us_states["Corruption Rate"].mean() - 2 * us_states["Corruption Rate"].std())
print("Standard Deviation (Upper Threshold):", us_states["Corruption Rate"].mean() + 2 * us_states["Corruption Rate"].std())
print("Tukey's Fence Method (Lower Fence):", us_states["Corruption Rate"].quantile(0.25) - 1.5 * (us_states["Corruption Rate"].quantile(0.75) - us_states["Corruption Rate"].quantile(0.25)))
print("Tukey's Fence Method (Upper Fence):", us_states["Corruption Rate"].quantile(0.75) + 1.5 * (us_states["Corruption Rate"].quantile(0.75) - us_states["Corruption Rate"].quantile(0.25)))
```

```
Mean Absolute Deviation (Lower Threshold): -0.04111200000000004
Mean Absolute Deviation (Upper Threshold): 3.4303120000000007
Standard Deviation (Lower Threshold): -1.0596389148365466
Standard Deviation (Upper Threshold): 4.448838914836547
Tukey's Fence Method (Lower Fence): -0.46125000000000016
Tukey's Fence Method (Upper Fence): 3.44875
```

Python Output 21

```
print("Extreme Values (Mean Absolute Deviation):", us_states["Corruption Rate"][~((us_states["Corruption Rate"] >= -0.04111200000000004) & (us_states["Corruption Rate"] <= 3.4303120000000007))].count())
print("Extreme Values (Standard Deviation):", us_states["Corruption Rate"][~((us_states["Corruption Rate"] >= -1.0596389148365466) & (us_states["Corruption Rate"] <= 4.448838914836547))].count())
print("Extreme Values (Tukey's Fence Method):", us_states["Corruption Rate"][~((us_states["Corruption Rate"] >= -0.46125000000000016) & (us_states["Corruption Rate"] <=  3.44875))].count())
```

```
Extreme Values (Mean Absolute Deviation): 4
Extreme Values (Standard Deviation): 2
Extreme Values (Tukey's Fence Method): 4
```

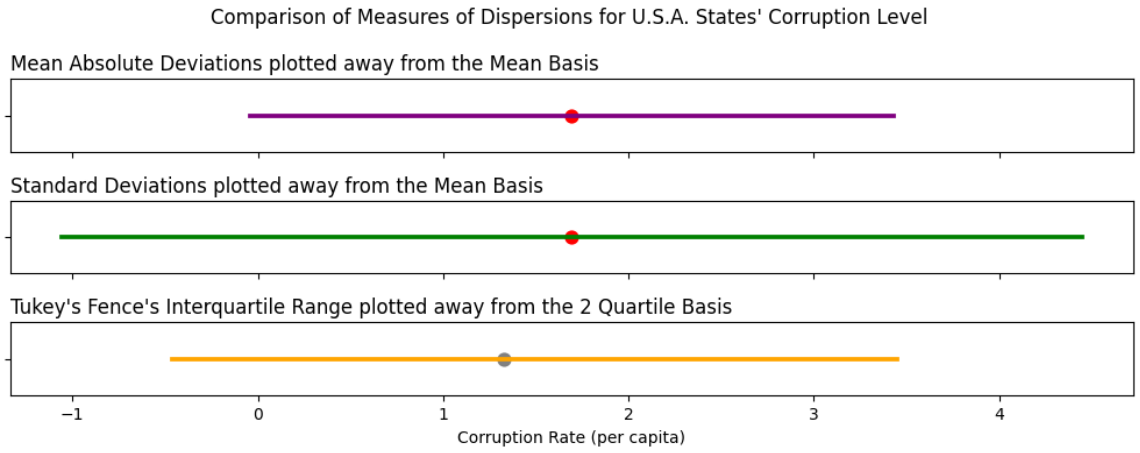
Python Output 22

```
us_cr_pp_thresholds = pd.DataFrame(
 {
   "class" : ["lower threshold", "basis", "upper threshold"],
   "mad" : ["-0.04111200000000004", "1.6946000000000003", "3.4303120000000007"],
   "std" : ["-1.0596389148365466", "1.6946000000000003", "4.448838914836547"],
   "tfm" : ["-0.46125000000000016", "1.325", " 3.44875"]
 }
)
us_cr_pp_thresholds = us_cr_pp_thresholds.set_index("class").astype("float")
display(us_cr_pp_thresholds)
```

class	mad	std	tfm
lower threshold	-0.041112	-1.059639	-0.46125
basis	1.694600	1.694600	1.32500
upper threshold	3.430312	4.448839	3.44875

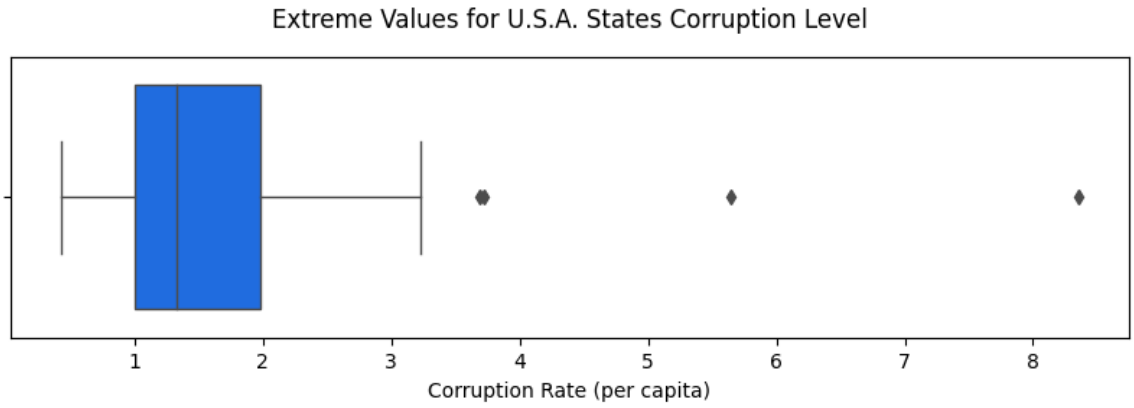
Python Output 23

```
fig, ax = plt.subplots(3,1, figsize=(10,4), sharex=True)
plt.suptitle("Comparison of Measures of Dispersions for U.S.A. States' Corruption Level")
plt.subplot(3,1,1)
sns.pointplot(data=us_cr_pp_thresholds, x="mad", estimator="mean", errorbar=("ci", 95), n_boot=1000, markers='|',
linestyles='-', scale=0.5, color="purple", ax=ax[0])
sns.pointplot(data=us_states, x="Corruption Rate", estimator="mean", errorbar=None, n_boot=1000, markers='o',
linestyles='-', scale=1, color="red", ax=ax[0])
ax[0].set_title("Mean Absolute Deviations plotted away from the Mean Basis", loc="left")
plt.xlabel(None)
plt.subplot(3,1,2)
sns.pointplot(data=us_cr_pp_thresholds, x="std", estimator="mean", errorbar=("ci", 95), n_boot=1000, markers='|',
linestyles='-', scale=0.5, color="green", ax=ax[1])
sns.pointplot(data=us_states, x="Corruption Rate", estimator="mean", errorbar=None, n_boot=1000, markers='o',
linestyles='-', scale=1, color="red", ax=ax[1])
ax[1].set_title("Standard Deviations plotted away from the Mean Basis", loc="left")
plt.xlabel(None)
plt.subplot(3,1,3)
sns.pointplot(data=us_cr_pp_thresholds, x="tfm", estimator="median", errorbar=("ci", 95), n_boot=1000, markers='|',
linestyles='-', scale=0.5, color="orange", ax=ax[2])
sns.pointplot(data=us_states, x="Corruption Rate", estimator="median", errorbar=None, n_boot=1000, markers='o',
linestyles='-', scale=1, color="grey", ax=ax[2])
ax[2].set_title("Tukey's Fence's Interquartile Range plotted away from the 2 Quartile Basis", loc="left")
plt.xlabel("Corruption Rate (per capita)")
plt.tight_layout()
plt.show()
```



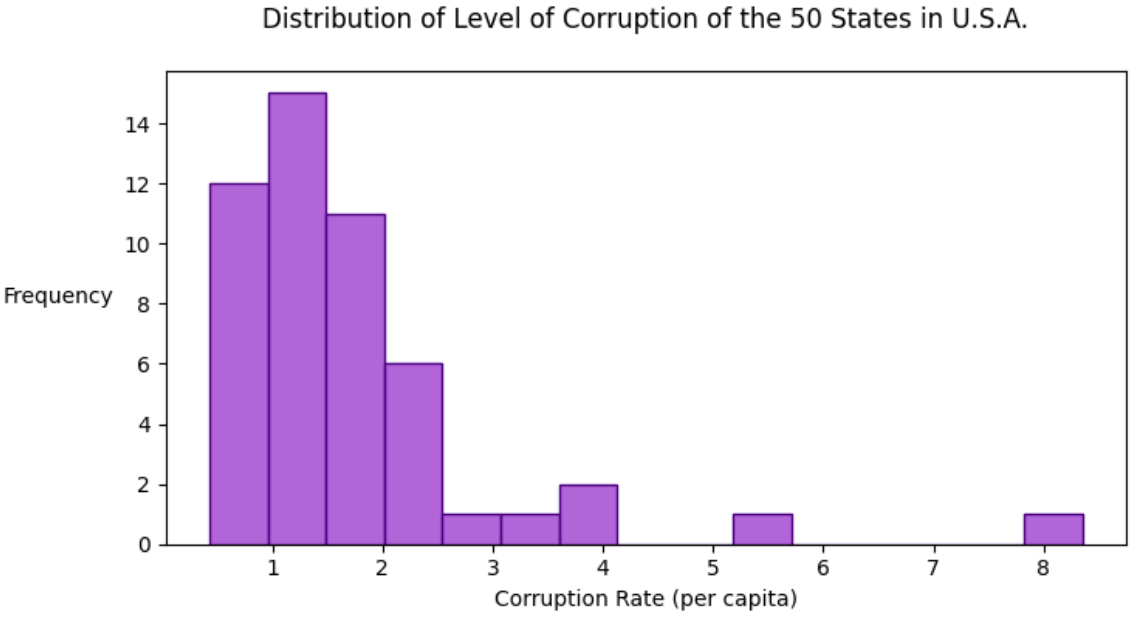
Python Chart 6

```
plt.figure(figsize=(10,2.5))
ax = sns.boxplot(data=us_states, x="Corruption Rate", color="#0066FF", width=0.8, fliersize=5, linewidth=1, whis=1.5)
ax.set_title("Extreme Values for U.S.A. States Corruption Level", pad=15)
ax.set_xlabel("Corruption Rate (per capita)")
plt.show()
```



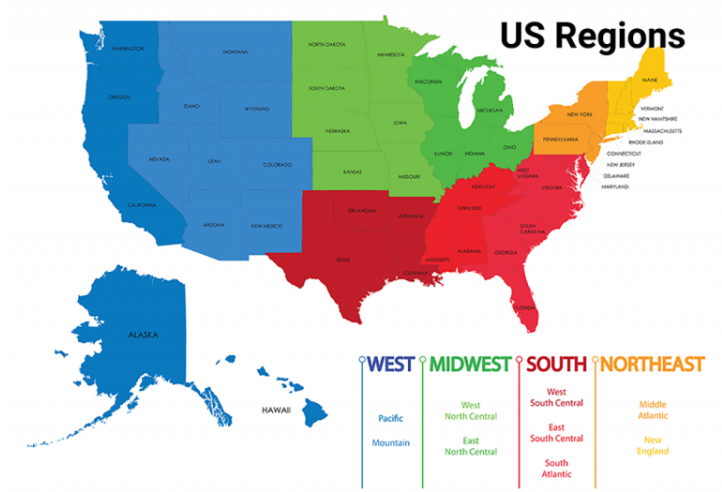
Python Chart 7

```
plt.figure(figsize=(8,4))
sns.histplot(data=us_states, x="Corruption Rate", bins="auto", shrink=1, color="darkorchid", edgecolor="indigo").set_title("Distribution of Level of Corruption of the 50 States in U.S.A.", pad=20)
plt.xlabel("Corruption Rate (per capita)")
plt.ylabel("Frequency", rotation=0, labelpad=30)
plt.show()
```



Python Chart 8

5. Created “region” column in the “us_states” dataframe to add the corresponding regions of the states for further analysis.



Python Figure 1

```
us_states["Region"] = us_states["States"].replace(
  to_replace=
    [
        "Connecticut", "Maine", "Massachusetts", "New Hampshire", "Rhode Island", "Vermont", "New Jersey", "New York",
"Pennsylvania", "Illinois", "Indiana", "Michigan", "Ohio", "Wisconsin", "Iowa",
        "Kansas", "Minnesota", "Missouri", "Nebraska", "North Dakota", "South Dakota", "Delaware", "Florida", "Georgia",
"Maryland", "North Carolina", "South Carolina", "Virginia", "West Virginia", "Alabama",
       "Kentucky", "Mississippi", "Tennessee", "Arkansas", "Louisiana", "Oklahoma", "Texas", "Arizona", "Colorado", "Idaho",
"Montana", "Nevada", "New Mexico", "Utah", "Wyoming",
      "Alaska", "California", "Hawaii", "Oregon", "Washington",
    ],
    value=
    [
        "Northeast", "Northeast", "Northeast", "Northeast", "Northeast", "Northeast", "Northeast", "Northeast", "Northeast",
"Midwest", "Midwest", "Midwest", "Midwest", "Midwest", "Midwest",
         "Midwest", "Midwest", "Midwest", "Midwest", "Midwest", "Midwest", "South", "South", "South", "South", "South",
"South", "South", "South", "South",
       "South", "South", "South", "South", "South", "South", "South", "West", "West", "West", "West", "West", "West", "West",
"West",
      "West", "West", "West", "West", "West",
    ]
)
display(us_states[["States", "Region"]].head(5))
display(us_states[["States", "Region"]].tail(5))
```

	States	Region
0	Alabama	South
1	Alaska	West
2	Arizona	West
3	Arkansas	South
4	California	West

	States	Region
45	Virginia	South
46	Washington	West
47	West Virginia	South
48	Wisconsin	Midwest
49	Wyoming	West

Python Output 24

```
print("Number of States in U.S.A. Regions:")
print("Northeast =", us_states["Region"].loc[(us_states["Region"] == "Northeast")].count())
print("Midwest =", us_states["Region"].loc[(us_states["Region"] == "Midwest")].count())
print("South =", us_states["Region"].loc[(us_states["Region"] == "South")].count())
print("West =", us_states["Region"].loc[(us_states["Region"] == "West")].count())
```

```
Number of States in U.S.A. Regions:
Northeast = 9
Midwest = 12
South = 16
West = 13
```

Python Output 25

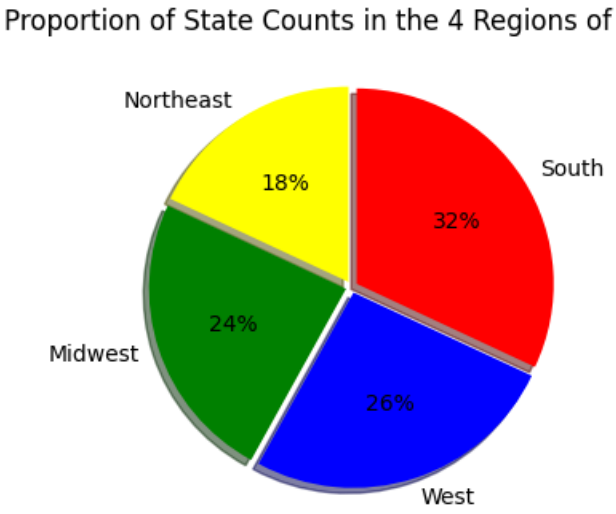
```
us_region_p_list = pd.DataFrame(
 {
   "South" : [16],
   "West" : [13],
   "Midwest" : [12],
   "Northeast" : [9]
 }
)
display(us_region_p_list)
us_region_p_labels = ["South", "West", "Midwest", "Northeast"]
us_region_p_colors = ["red", "blue", "green", "yellow"]
us_region_p_explode = [0.03, 0.03, 0.03, 0.03]
print("---------------------------------")
us_region_p_list_stacked_sorted = us_region_p_list.stack().sort_values(ascending=False)
display(us_region_p_list_stacked_sorted.to_frame())
```

	South	West	Midwest	Northeast
0	16	13	12	9

		0
0	South	16
	West	13
	Midwest	12
	Northeast	9

Python Output 26

```
plt.figure(figsize=(8,4))
plt.pie(us_region_p_list_stacked_sorted, labels=us_region_p_labels, colors=us_region_p_colors,
explode=us_region_p_explode, autopct="%1.0f%%", counterclock=False, startangle=90, shadow=True)
plt.title("Proportion of State Counts in the 4 Regions of U.S.A.")
plt.show()
```



Python Chart 9

```
us_states[["Region", "States", "Corruption Rate"]].sort_values(by=["Region", "States"], ascending=[True, True]).head(10)
```

	Region	States	Corruption Rate
12	Midwest	Illinois	1.27
13	Midwest	Indiana	0.90
14	Midwest	Iowa	0.58
15	Midwest	Kansas	1.02
21	Midwest	Michigan	1.00
22	Midwest	Minnesota	0.68
24	Midwest	Missouri	1.50
26	Midwest	Nebraska	0.57
33	Midwest	North Dakota	0.57
34	Midwest	Ohio	0.89

Python Output 27

```
display("Minimum Corruption Rate by Region:", us_states.groupby("Region")["Corruption Rate"].min().to_frame())
print("--------------------------------")
display("Average Corruption Rate by Region:", us_states.groupby("Region")["Corruption Rate"].mean().to_frame())
print("--------------------------------")
display("Maximum Corruption Rate by Region:", us_states.groupby("Region")["Corruption Rate"].max().to_frame())
```

'Minimum Corruption Rate by Region:'

Region	Corruption Rate
Midwest	0.57
Northeast	0.44
South	1.05
West	0.43

'Average Corruption Rate by Region:'

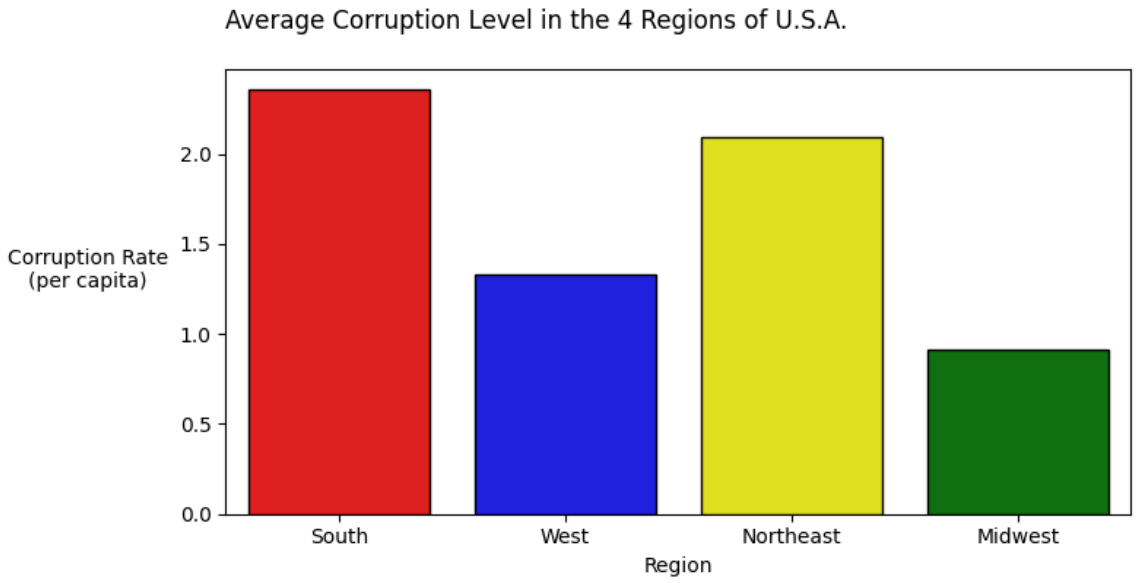
Region	Corruption Rate
Midwest	0.911667
Northeast	2.088889
South	2.354375
West	1.332308

'Maximum Corruption Rate by Region:'

Region	Corruption Rate
Midwest	1.50
Northeast	8.35
South	5.64
West	2.53

Python Output 28

```
plt.figure(figsize=(8,4))
sns.barplot(data=us_states, x="Region", y="Corruption Rate", palette={"Northeast" : "yellow", "Midwest" : "green", "South" : "red", "West" : "blue"}, edgecolor="black", errorbar=None)
plt.title("Average Corruption Level in the 4 Regions of U.S.A.", loc="left", pad=20)
plt.xlabel("Region")
plt.ylabel("Corruption Rate\n(per capita)", rotation=0, labelpad=45)
plt.show()
```



Python Chart 10

6. Summarized the statistics of Population Size of the 50 states in U.S.A.

```
print("Frequencies:")
us_states["Population Size"].value_counts()
```

```
Frequencies:
4903185     1
12801989    1
3080156     1
1359711     1
8882190     1
2096829     1
19453561    1
10488084    1
762062      1
11689100    1
3956971     1
4217737     1
1059361     1
731545      1
5148714     1
884659      1
6829174     1
28995881    1
3205958     1
623989      1
8535519     1
7614893     1
1792147     1
5822434     1
1934408     1
1068778     1
```

Python Output 29

```
print("Descriptive Statistics:")
print("Count:", us_states["Population Size"].count())
print("Distinct Count:", us_states["Population Size"].nunique())
print("Sum:", us_states["Population Size"].sum())
print("Minimum:", us_states["Population Size"].min())
print("Maximum:", us_states["Population Size"].max())
print("Average:", us_states["Population Size"].mean())
print("Median:", us_states["Population Size"].median())
print("Mode:", modal_value(us_states["Population Size"]))
print("Lower Quartile:", us_states["Population Size"].quantile(0.25))
print("Upper Quartile:", us_states["Population Size"].quantile(0.75))
print("Range:", us_states["Population Size"].max() - us_states["Population Size"].min())
print("Interquartile Range:", us_states["Population Size"].quantile(0.75) - us_states["Population Size"].quantile(0.25))
print("Standard Deviation:", us_states["Population Size"].std())
print("Variance:", us_states["Population Size"].std() ** 2)
print("Mean Absolute Deviation:", us_states["Population Size"].mad())
print("Skewness:", us_states["Population Size"].skew(axis=0))
print("Kurtosis:", us_states["Population Size"].kurtosis(axis=0))
print("----------------------------------------")
```

```
Descriptive Statistics:
Count: 50
Distinct Count: 50
Sum: 327533774
Minimum: 578759
Maximum: 39512223
Average: 6550675.48
Median: 4558233.5
Mode: None
Lower Quartile: 1827712.25
Upper Quartile: 7530849.0
Range: 38933464
Interquartile Range: 5703136.75
Standard Deviation: 7389281.84983264
Variance: 54601486256266.086
Mean Absolute Deviation: 4763936.3136
Skewness: 2.6734134592162357
Kurtosis: 8.642416911689871
----------------------------------------
```

Python Output 30

```
print("Mean Absolute Deviation (Lower Threshold):", us_states["Population Size"].mean() - 2 * us_states["Population Size"].mad())
print("Mean Absolute Deviation (Upper Threshold):", us_states["Population Size"].mean() + 2 * us_states["Population Size"].mad())
print("Standard Deviation (Lower Threshold):", us_states["Population Size"].mean() - 2 * us_states["Population Size"].std())
print("Standard Deviation (Upper Threshold):", us_states["Population Size"].mean() + 2 * us_states["Population Size"].std())
print("Tukey's Fence Method (Lower Fence):", us_states["Population Size"].quantile(0.25) - 1.5 * (us_states["Population Size"].quantile(0.75) - us_states["Population Size"].quantile(0.25)))
print("Tukey's Fence Method (Upper Fence):", us_states["Population Size"].quantile(0.75) + 1.5 * (us_states["Population Size"].quantile(0.75) - us_states["Population Size"].quantile(0.25)))
```

```
Mean Absolute Deviation (Lower Threshold): -2977197.1471999995
Mean Absolute Deviation (Upper Threshold): 16078548.1072
Standard Deviation (Lower Threshold): -8227888.21966528
Standard Deviation (Upper Threshold): 21329239.179665282
Tukey's Fence Method (Lower Fence): -6726992.875
Tukey's Fence Method (Upper Fence): 16085554.125
```

Python Output 31

```
print("Extreme Values (Mean Absolute Deviation):", us_states["Population Size"][~((us_states["Population Size"] >= -2977197.1471999995) & (us_states["Population Size"] <= 16078548.1072))].count())
print("Extreme Values (Standard Deviation):", us_states["Population Size"][~((us_states["Population Size"] >= -8227888.21966528) & (us_states["Population Size"] <= 21329239.179665282))].count())
print("Extreme Values (Tukey's Fence Method):", us_states["Population Size"][~((us_states["Population Size"] >= -6726992.875) & (us_states["Population Size"] <= 16085554.125))].count())
```

```
Extreme Values (Mean Absolute Deviation): 4
Extreme Values (Standard Deviation): 3
Extreme Values (Tukey's Fence Method): 4
```

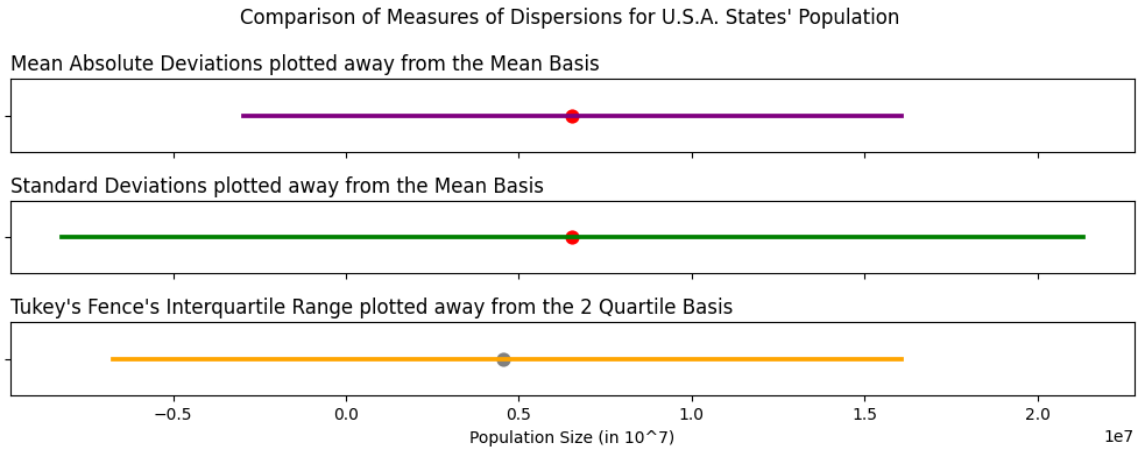
Python Output 32

```
us_population_pp_thresholds = pd.DataFrame(
 {
  "class" : ["lower threshold", "basis", "upper threshold"],
  "mad" : ["-2977197.1471999995", "6550675.48", "16078548.1072"],
  "std" : ["-8227888.21966528", "6550675.48", "21329239.179665282"],
  "tfm" : ["-6726992.875", "4558233.5", "16085554.125"]
 }
)
us_population_pp_thresholds = us_population_pp_thresholds.set_index("class").astype("float")
display(us_population_pp_thresholds)
```

class	mad	std	tfm
lower threshold	-2.977197e+06	-8.227888e+06	-6.726993e+06
basis	6.550675e+06	6.550675e+06	4.558234e+06
upper threshold	1.607855e+07	2.132924e+07	1.608555e+07

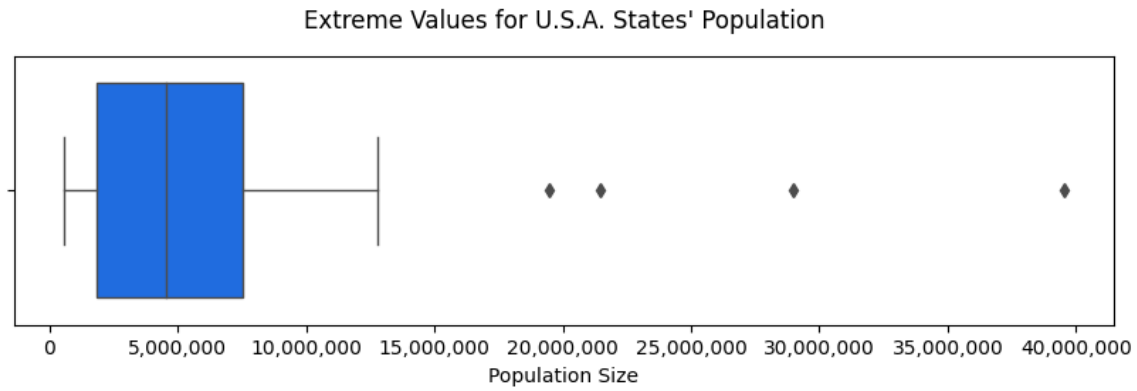
Python Output 33

```
fig, ax = plt.subplots(3,1, figsize=(10,4), sharex=True)
plt.suptitle("Comparison of Measures of Dispersions for U.S.A. States' Population")
plt.subplot(3,1,1)
sns.pointplot(data=us_population_pp_thresholds, x="mad", estimator="mean", errorbar=("ci", 95), n_boot=1000, markers='|', linestyles='-', scale=0.5, color="purple", ax=ax[0])
sns.pointplot(data=us_states, x="Population Size", estimator="mean", errorbar=None, n_boot=1000, markers='o', linestyles='-', scale=1, color="red", ax=ax[0])
ax[0].set_title("Mean Absolute Deviations plotted away from the Mean Basis", loc="left")
plt.xlabel(None)
plt.subplot(3,1,2)
sns.pointplot(data=us_population_pp_thresholds, x="std", estimator="mean", errorbar=("ci", 95), n_boot=1000, markers='|', linestyles='-', scale=0.5, color="green", ax=ax[1])
sns.pointplot(data=us_states, x="Population Size", estimator="mean", errorbar=None, n_boot=1000, markers='o', linestyles='-', scale=1, color="red", ax=ax[1])
ax[1].set_title("Standard Deviations plotted away from the Mean Basis", loc="left")
plt.xlabel(None)
plt.subplot(3,1,3)
sns.pointplot(data=us_population_pp_thresholds, x="tfm", estimator="median", errorbar=("ci", 95), n_boot=1000, markers='|', linestyles='-', scale=0.5, color="orange", ax=ax[2])
sns.pointplot(data=us_states, x="Population Size", estimator="median", errorbar=None, n_boot=1000, markers='o', linestyles='-', scale=1, color="grey", ax=ax[2])
ax[2].set_title("Tukey's Fence's Interquartile Range plotted away from the 2 Quartile Basis", loc="left")
plt.xlabel("Population Size (in 10^7)")
plt.tight_layout()
plt.show()
```



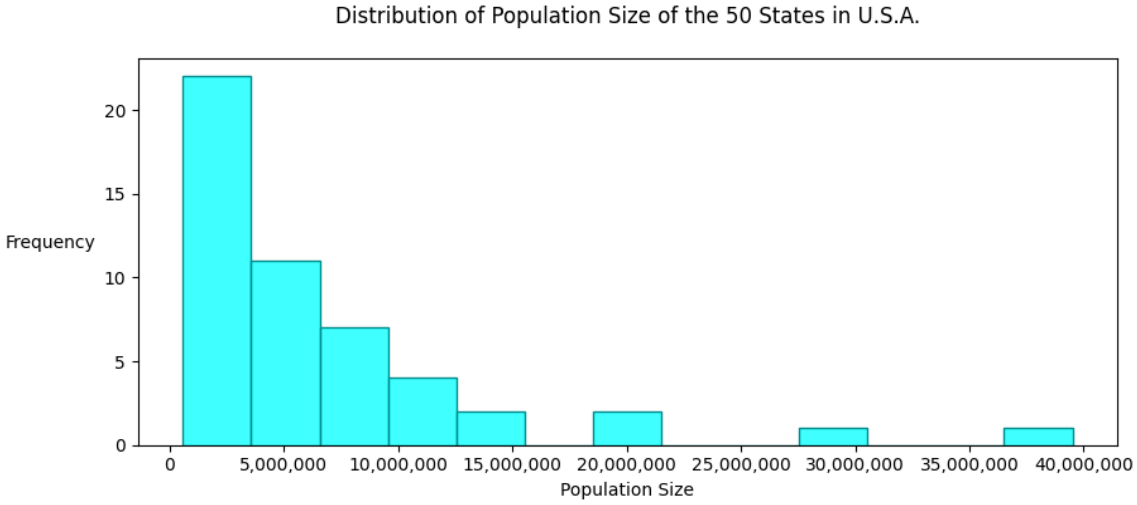
Python Chart 11

```
plt.figure(figsize=(10,2.5))
ax = sns.boxplot(data=us_states, x="Population Size", color="#0066FF", width=0.8, fliersize=5, linewidth=1, whis=1.5)
ax.set_title("Extreme Values for U.S.A. States' Population", pad=15)
ax.set_xlabel("Population Size")
tick_labels = [f"{tick:,.0f}" for tick in ax.get_xticks()]
ax.set_xticklabels(tick_labels)
plt.show()
```



Python Chart 12

```
plt.figure(figsize=(10,4))
ax = sns.histplot(data=us_states, x="Population Size", bins="auto", shrink=1, color="aqua", edgecolor="darkcyan")
ax.set_title("Distribution of Population Size of the 50 States in U.S.A.", pad=20)
ax.set_xlabel("Population Size")
ax.set_ylabel("Frequency", rotation=0, labelpad=30)
tick_labels = [f"{tick:,.0f}" for tick in ax.get_xticks()]
ax.set_xticklabels(tick_labels)
plt.show()
```



Python Chart 13

7. Checked the skewness levels of the 5 key metrics used as a basis of criteria to consider expanding the business operations.

```
def skewness_check(skewness):
  if skewness >= 1.5:
    print("High Positive Skewness")
  elif skewness >= 1.0:
    print("Moderate Positive Skewness")
  elif skewness >= 0.5:
    print("Low Positive Skewness")
  elif skewness > -0.5:
    print("Symmetric Skewness")
  elif skewness > -1.0:
    print("Low Negative Skewness")
  elif skewness > -1.5:
    print("Moderate Negative Skewness")
  else:
    print("High Negative Skewness")
```

```
print("Skewness of Corruption Rate:", us_states["Corruption Rate"].skew(axis=0))
Skewness_CorruptionRate = skewness_check(us_states["Corruption Rate"].skew(axis=0))
print("-------------------------------------------------")
print("Skewness of Average Healthcare Spending:", us_states["Average Healthcare Spending"].skew(axis=0))
Skewness_AverageHealthcareSpending = skewness_check(us_states["Average Healthcare Spending"].skew(axis=0))
print("-------------------------------------------------")
print("Skewness of Population Size:", us_states["Population Size"].skew(axis=0))
Skewness_PopulationSize = skewness_check(us_states["Population Size"].skew(axis=0))
print("-------------------------------------------------")
print("Skewness of Average Property Price:", us_states["Average Property Price"].skew(axis=0))
Skewness_AveragePropertyPrice = skewness_check(us_states["Average Property Price"].skew(axis=0))
print("-------------------------------------------------")
print("Skewness of Average Income:", us_states["Average Income"].skew(axis=0))
Skewness_AverageIncome = skewness_check(us_states["Average Income"].skew(axis=0))
```

```
Skewness of Corruption Rate: 2.945607896440458
High Positive Skewness
-------------------------------------------------
Skewness of Average Healthcare Spending: 0.8051107121512822
Low Positive Skewness
-------------------------------------------------
Skewness of Population Size: 2.6734134592162357
High Positive Skewness
-------------------------------------------------
Skewness of Average Property Price: 1.7556213325224739
High Positive Skewness
-------------------------------------------------
Skewness of Average Income: 0.5328861882367197
Low Positive Skewness
```

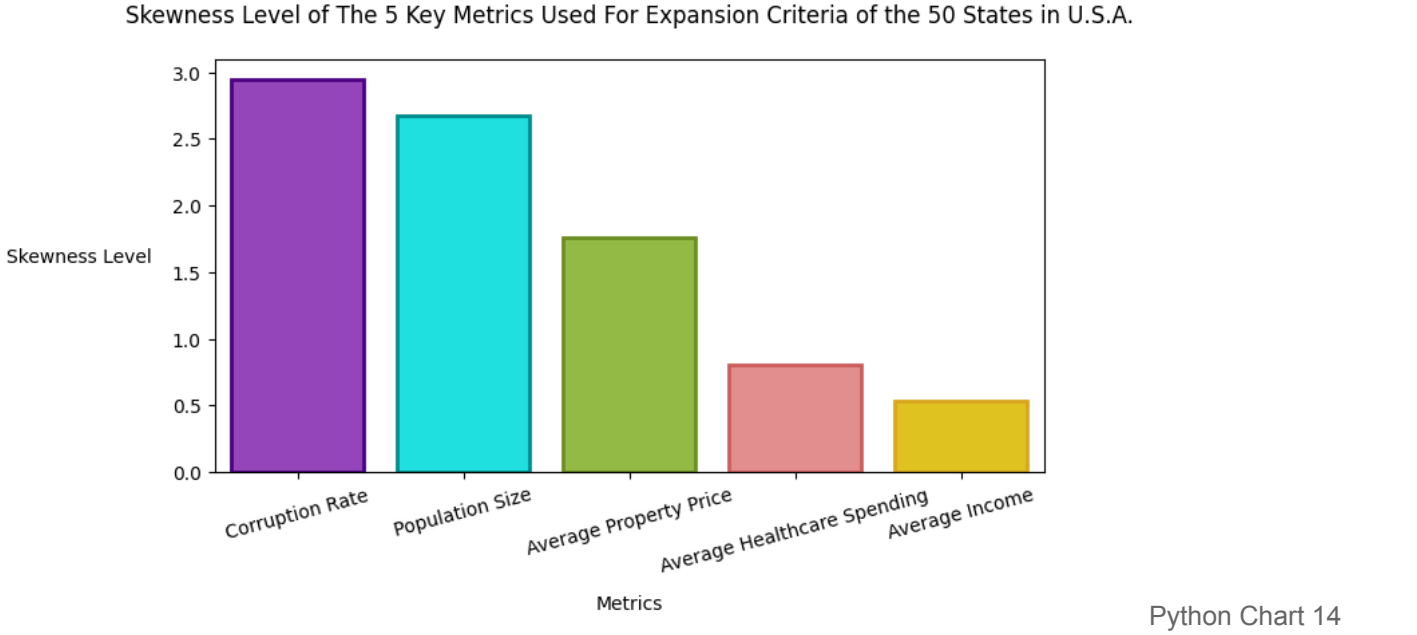
Python Output 34

```
Skewness_Metrics = pd.DataFrame(
 {
   "Corruption Rate" : [2.945607896440458],
   "Population Size" : [2.6734134592162357],
   "Average Property Price" : [1.7556213325224739],
   "Average Healthcare Spending" : [0.8051107121512822],
   "Average Income" : [0.5328861882367197]
 }
)
display(Skewness_Metrics)
us_sm_cc_colors = ["darkorchid", "aqua", "yellowgreen", "lightcoral", "gold"]
us_sm_cc_edgecolors = ["indigo", "darkcyan", "olivedrab", "indianred", "goldenrod"]
```

	Corruption Rate	Population Size	Average Property Price	Average Healthcare Spending	Average Income
0	2.945608	2.673413	1.755621	0.805111	0.532886

Python Output 35

```
plt.figure(figsize=(8,4))
sns.barplot(data=Skewness_Metrics, palette=us_sm_cc_colors, edgecolor=us_sm_cc_edgecolors, lw=2).set_title("Skewness Level of The 5 Key Metrics Used For Expansion Criteria of the 50 States in U.S.A.", pad=20)
plt.xlabel("Metrics", labelpad=10)
plt.ylabel("Skewness Level", rotation=0, labelpad=50)
plt.xticks(rotation=15)
plt.show()
```



Python Chart 14

8. Checked the lists of states having significantly high or low key metric values.

```
us_cr_lf = us_states["Corruption Rate"].quantile(0.25) - 1.5 * (us_states["Corruption Rate"].quantile(0.75) - us_states["Corruption Rate"].quantile(0.25))
us_cr_uf = us_states["Corruption Rate"].quantile(0.75) + 1.5 * (us_states["Corruption Rate"].quantile(0.75) - us_states["Corruption Rate"].quantile(0.25))
print("List of States with Corruption Rate outside the Tukey's Upper & Lower Fences:")
display(us_states[["States", "Corruption Rate"]][~((us_states["Corruption Rate"] >= us_cr_lf) & (us_states["Corruption Rate"] <= us_cr_uf))].sort_values(by="Corruption Rate", ascending=False))
print("-----------------------------------------------------------------------------------")
us_ahs_lf = us_states["Average Healthcare Spending"].quantile(0.25) - 1.5 * (us_states["Average Healthcare Spending"].quantile(0.75) - us_states["Average Healthcare Spending"].quantile(0.25))
us_ahs_uf = us_states["Average Healthcare Spending"].quantile(0.75) + 1.5 * (us_states["Average Healthcare Spending"].quantile(0.75) - us_states["Average Healthcare Spending"].quantile(0.25))
print("List of States with Average Healthcare Spending outside the Tukey's Upper & Lower Fences:")
display(us_states[["States", "Average Healthcare Spending"]][~((us_states["Average Healthcare Spending"] >= us_ahs_lf) & (us_states["Average Healthcare Spending"] <= us_ahs_uf))].sort_values(by="Average Healthcare Spending", ascending=False))
print("-----------------------------------------------------------------------------------")
us_population_lf = us_states["Population Size"].quantile(0.25) - 1.5 * (us_states["Population Size"].quantile(0.75) - us_states["Population Size"].quantile(0.25))
us_population_uf = us_states["Population Size"].quantile(0.75) + 1.5 * (us_states["Population Size"].quantile(0.75) - us_states["Population Size"].quantile(0.25))
print("List of States with Population Size outside the Tukey's Upper & Lower Fences:")
display(us_states[["States", "Population Size"]][~((us_states["Population Size"] >= us_population_lf) & (us_states["Population Size"] <= us_population_uf))].sort_values(by="Population Size", ascending=False))
print("-----------------------------------------------------------------------------------")
us_app_lf = us_states["Average Property Price"].quantile(0.25) - 1.5 * (us_states["Average Property Price"].quantile(0.75) - us_states["Average Property Price"].quantile(0.25))
us_app_uf = us_states["Average Property Price"].quantile(0.75) + 1.5 * (us_states["Average Property Price"].quantile(0.75) - us_states["Average Property Price"].quantile(0.25))
print("List of States with Average Property Price outside the Tukey's Upper & Lower Fences:")
display(us_states[["States", "Average Property Price"]][~((us_states["Average Property Price"] >= us_app_lf) & (us_states["Average Property Price"] <= us_app_uf))].sort_values(by="Average Property Price", ascending=False))
print("-----------------------------------------------------------------------------------")
us_ai_lf = us_states["Average Income"].quantile(0.25) - 1.5 * (us_states["Average Income"].quantile(0.75) - us_states["Average Income"].quantile(0.25))
us_ai_uf = us_states["Average Income"].quantile(0.75) + 1.5 * (us_states["Average Income"].quantile(0.75) - us_states["Average Income"].quantile(0.25))
print("List of States with Average Income outside the Tukey's Upper & Lower Fences:")
display(us_states[["States", "Average Income"]][~((us_states["Average Income"] >= us_ai_lf) & (us_states["Average Income"] <= us_ai_uf))].sort_values(by="Average Income", ascending=False))
print("-----------------------------------------------------------------------------------")
```

List of States with Corruption Rate outside the Tukey's Upper & Lower Fences:

	States	Corruption Rate
38	Rhode Island	8.35
47	West Virginia	5.64
17	Louisiana	3.72
41	Tennessee	3.69

List of States with Average Healthcare Spending outside the Tukey's Upper & Lower Fences:

	States	Average Healthcare Spending
10	Hawaii	350.0
20	Massachusetts	350.0
29	New Jersey	350.0
38	Rhode Island	350.0

List of States with Population Size outside the Tukey's Upper & Lower Fences:

	States	Population Size
4	California	39512223
42	Texas	28995881
8	Florida	21477737
31	New York	19453561

List of States with Average Property Price outside the Tukey's Upper & Lower Fences:

	States	Average Property Price
10	Hawaii	5975.50
4	California	5832.50
31	New York	4968.75
29	New Jersey	4672.25

Python Output 36

9. Searched the locations of the competitors across the states in U.S.A.

```
competitors_location = competitors_revised["States"].nunique()
print(f"Competitors are located in {competitors_location} states.")
del competitors_location
```

```
Competitors are located in 41 states.
```

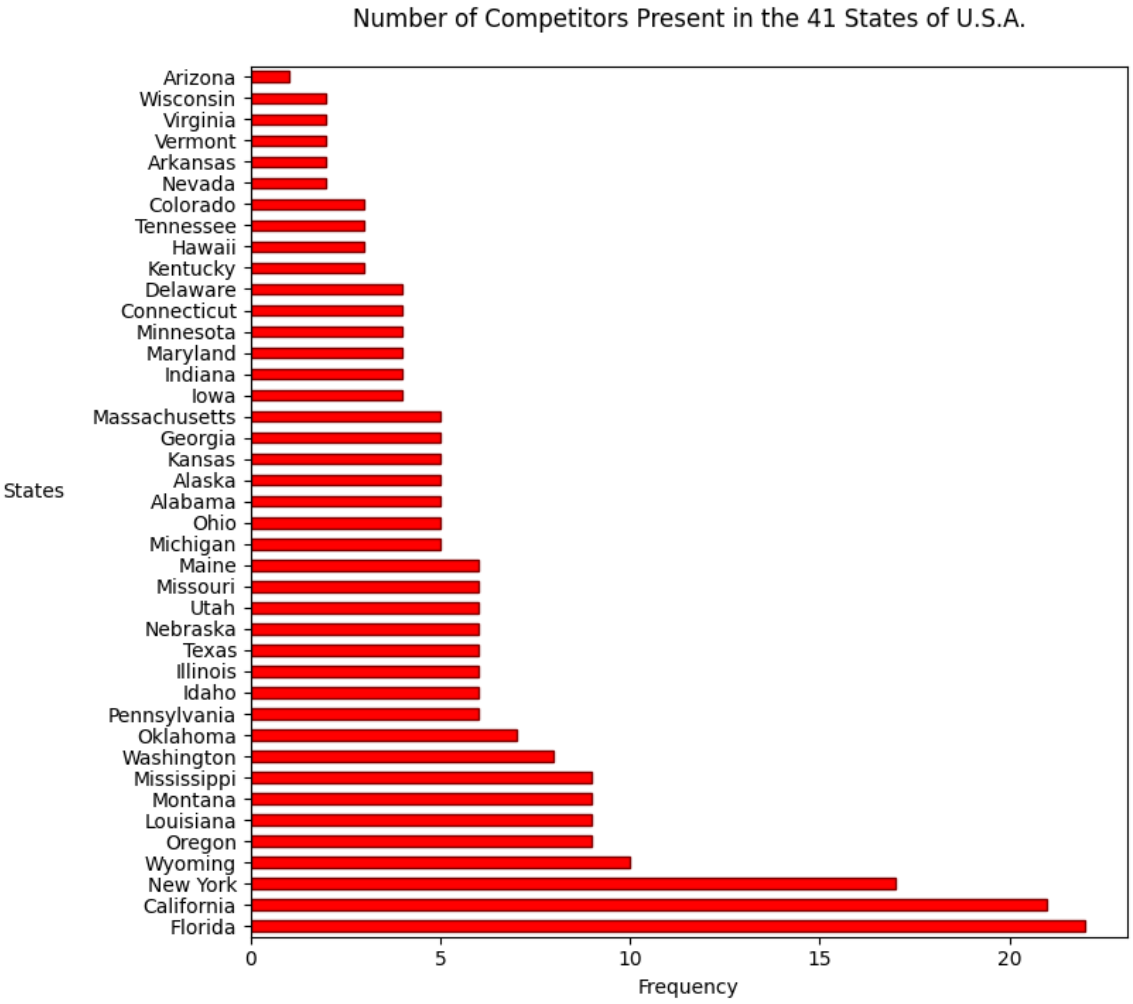
Python Output 37

```
comp_states_bc = competitors_revised["States"].value_counts().reset_index()
comp_states_bc.columns = ["States", "count"]
display(comp_states_bc.head(10))
```

	States	count
0	Florida	22
1	California	21
2	New York	17
3	Wyoming	10
4	Oregon	9
5	Louisiana	9
6	Montana	9
7	Mississippi	9
8	Washington	8
9	Oklahoma	7

Python Output 38

```
fig, ax = plt.subplots(figsize=(8,8))
comp_states_bc.plot(kind="barh", x="States", y="count", color="red", edgecolor="maroon", legend=None, ax=ax)
plt.title("Number of Competitors Present in the 41 States of U.S.A.", pad=20)
plt.xlabel("Frequency")
plt.ylabel("States", rotation=0, labelpad=30)
plt.show()
```



Python Chart 15

```
comp_states_list = list(comp_states_bc["States"])
us_states_list = list(us_states["States"])
i = 0
while i < len(us_states_list):
  if us_states_list[i] not in comp_states_list:
    print(f"No competitors in {us_states_list[i]}.")
  i = i + 1
```

```
No competitors in New Hampshire.
No competitors in New Jersey.
No competitors in New Mexico.
No competitors in North Carolina.
No competitors in North Dakota.
No competitors in Rhode Island.
No competitors in South Carolina.
No competitors in South Dakota.
No competitors in West Virginia.
```

Python Output 39

10. Summarized the statistics of the Competitors' Profits (US$) located in the 50 states in U.S.A.

```
print("Frequencies:")
competitors_revised["Profit"].value_counts()
```

```
Frequencies:
146121.95     2
120362.99     1
101004.64     1
1141042.99    1
1867711.99    1
             ..
2622572.99    1
87874.99      1
731798.99     1
1637569.99    1
2935405.99    1
Name: Profit, Length: 250, dtype: int64
```

Python Output 40

```
print("Descriptive Statistics:")
print("Count:", competitors_revised["Profit"].count())
print("Distinct Count:", competitors_revised["Profit"].nunique())
print("Sum:", competitors_revised["Profit"].sum())
print("Minimum:", competitors_revised["Profit"].min())
print("Maximum:", competitors_revised["Profit"].max())
print("Average:", competitors_revised["Profit"].mean())
print("Median:", competitors_revised["Profit"].median())
print("Mode:", list(competitors_revised["Profit"].mode()))
print("Lower Quartile:", competitors_revised["Profit"].quantile(0.25))
print("Upper Quartile:", competitors_revised["Profit"].quantile(0.75))
print("Range:", competitors_revised["Profit"].max() - competitors_revised["Profit"].min())
print("Interquartile Range:", competitors_revised["Profit"].quantile(0.75) - competitors_revised["Profit"].quantile(0.25))
print("Standard Deviation:", competitors_revised["Profit"].std())
print("Variance:", competitors_revised["Profit"].std() ** 2)
print("Mean Absolute Deviation:", competitors_revised["Profit"].mad())
print("Skewness:", competitors_revised["Profit"].skew(axis=0))
print("Kurtosis:", competitors_revised["Profit"].kurtosis(axis=0))
print("----------------------------------------")
```

```
Descriptive Statistics:
Count: 251
Distinct Count: 250
Sum: 314303300.90999997
Minimum: 14681.4
Maximum: 2946679.99
Average: 1252204.3860956174
Median: 1154271.99
Mode: [146121.95]
Lower Quartile: 236423.49
Upper Quartile: 2149321.49
Range: 2931998.5900000003
Interquartile Range: 1912898.0000000002
Standard Deviation: 949751.7389303034
Variance: 902028365601.1351
Mean Absolute Deviation: 845008.6172790909
Skewness: 0.2163514895410596
Kurtosis: -1.3771789986050575
----------------------------------------
```

Python Output 41

```
print("Mean Absolute Deviation (Lower Threshold):", competitors_revised["Profit"].mean() - 2 * competitors_revised["Profit"].mad())
print("Mean Absolute Deviation (Upper Threshold):", competitors_revised["Profit"].mean() + 2 * competitors_revised["Profit"].mad())
print("Standard Deviation (Lower Threshold):", competitors_revised["Profit"].mean() - 2 * competitors_revised["Profit"].std())
print("Standard Deviation (Upper Threshold):", competitors_revised["Profit"].mean() + 2 * competitors_revised["Profit"].std())
print("Tukey's Fence Method (Lower Fence):", competitors_revised["Profit"].quantile(0.25) - 1.5 * (competitors_revised["Profit"].quantile(0.75) - competitors_revised["Profit"].quantile(0.25)))
print("Tukey's Fence Method (Upper Fence):", competitors_revised["Profit"].quantile(0.75) + 1.5 * (competitors_revised["Profit"].quantile(0.75) - competitors_revised["Profit"].quantile(0.25)))
```

```
Mean Absolute Deviation (Lower Threshold): -437812.8484625644
Mean Absolute Deviation (Upper Threshold): 2942221.6206537993
Standard Deviation (Lower Threshold): -647299.0917649893
Standard Deviation (Upper Threshold): 3151707.863956224
Tukey's Fence Method (Lower Fence): -2632923.5100000007
Tukey's Fence Method (Upper Fence): 5018668.49
```

Python Output 42

```
print("Extreme Values (Mean Absolute Deviation):", competitors_revised["Profit"][~((competitors_revised["Profit"] >= -437812.8484625644) & (competitors_revised["Profit"] <= 2942221.6206537993))].count())
print("Extreme Values (Standard Deviation):", competitors_revised["Profit"][~((competitors_revised["Profit"] >= -647299.0917649893) & (competitors_revised["Profit"] <= 3151707.863956224))].count())
print("Extreme Values (Tukey's Fence Method):", competitors_revised["Profit"][~((competitors_revised["Profit"] >= -2632923.5100000007) & (competitors_revised["Profit"] <= 5018668.49))].count())
```

```
Extreme Values (Mean Absolute Deviation): 2
Extreme Values (Standard Deviation): 0
Extreme Values (Tukey's Fence Method): 0
```

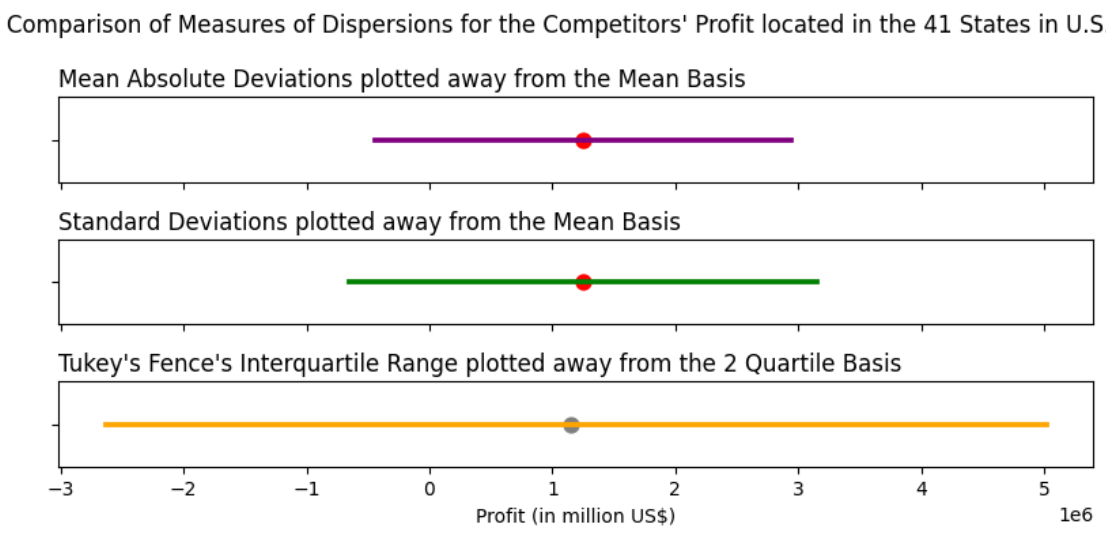
Python Output 43

```
comp_profit_pp_thresholds = pd.DataFrame(
 {
   "class" : ["lower threshold", "basis", "upper threshold"],
   "mad" : ["-437812.8484625644", "1252204.3860956174", "2942221.6206537993"],
   "std" : ["-647299.0917649893", "1252204.3860956174", "3151707.863956224"],
   "tfm" : ["-2632923.5100000007", "1154271.99", "5018668.49"]
 }
)
comp_profit_pp_thresholds = comp_profit_pp_thresholds.set_index("class").astype("float")
display(comp_profit_pp_thresholds)
```

class	mad	std	tfm
lower threshold	-4.378128e+05	-6.472991e+05	-2632923.51
basis	1.252204e+06	1.252204e+06	1154271.99
upper threshold	2.942222e+06	3.151708e+06	5018668.49

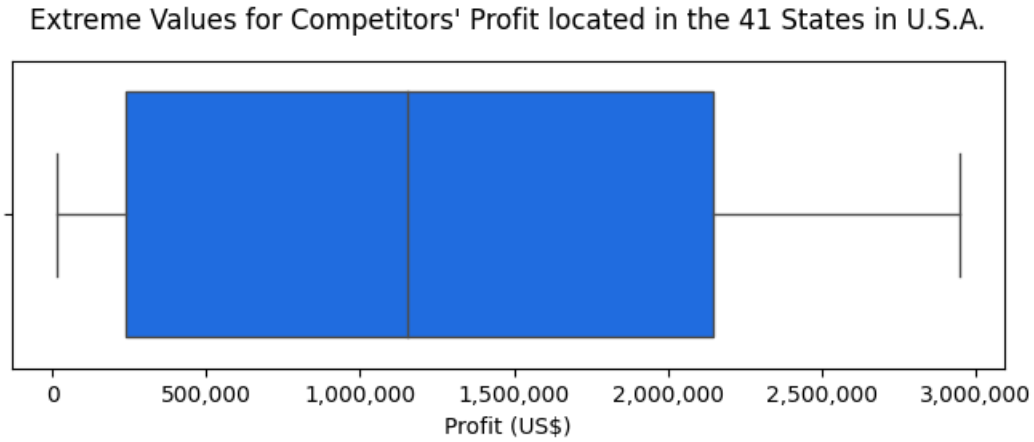
Python Output 44

```
fig, ax = plt.subplots(3,1, figsize=(8,4), sharex=True)
plt.suptitle("Comparison of Measures of Dispersions for the Competitors' Profit located in the 41 States in U.S.A.")
plt.subplot(3,1,1)
sns.pointplot(data=comp_profit_pp_thresholds, x="mad", estimator="mean", errorbar=("ci", 95), n_boot=1000, markers='|',
linestyles='-', scale=0.5, color="purple", ax=ax[0])
sns.pointplot(data=competitors_revised, x="Profit", estimator="mean", errorbar=None, n_boot=1000, markers='o',
linestyles='-', scale=1, color="red", ax=ax[0])
ax[0].set_title("Mean Absolute Deviations plotted away from the Mean Basis", loc="left")
plt.xlabel(None)
plt.subplot(3,1,2)
sns.pointplot(data=comp_profit_pp_thresholds, x="std", estimator="mean", errorbar=("ci", 95), n_boot=1000, markers='|',
linestyles='-', scale=0.5, color="green", ax=ax[1])
sns.pointplot(data=competitors_revised, x="Profit", estimator="mean", errorbar=None, n_boot=1000, markers='o',
linestyles='-', scale=1, color="red", ax=ax[1])
ax[1].set_title("Standard Deviations plotted away from the Mean Basis", loc="left")
plt.xlabel(None)
plt.subplot(3,1,3)
sns.pointplot(data=comp_profit_pp_thresholds, x="tfm", estimator="median", errorbar=("ci", 95), n_boot=1000,
markers='|', linestyles='-', scale=0.5, color="orange", ax=ax[2])
sns.pointplot(data=competitors_revised, x="Profit", estimator="median", errorbar=None, n_boot=1000, markers='o',
linestyles='-', scale=1, color="grey", ax=ax[2])
ax[2].set_title("Tukey's Fence's Interquartile Range plotted away from the 2 Quartile Basis", loc="left")
plt.xlabel("Profit (in million US$)")
plt.tight_layout()
plt.show()
```



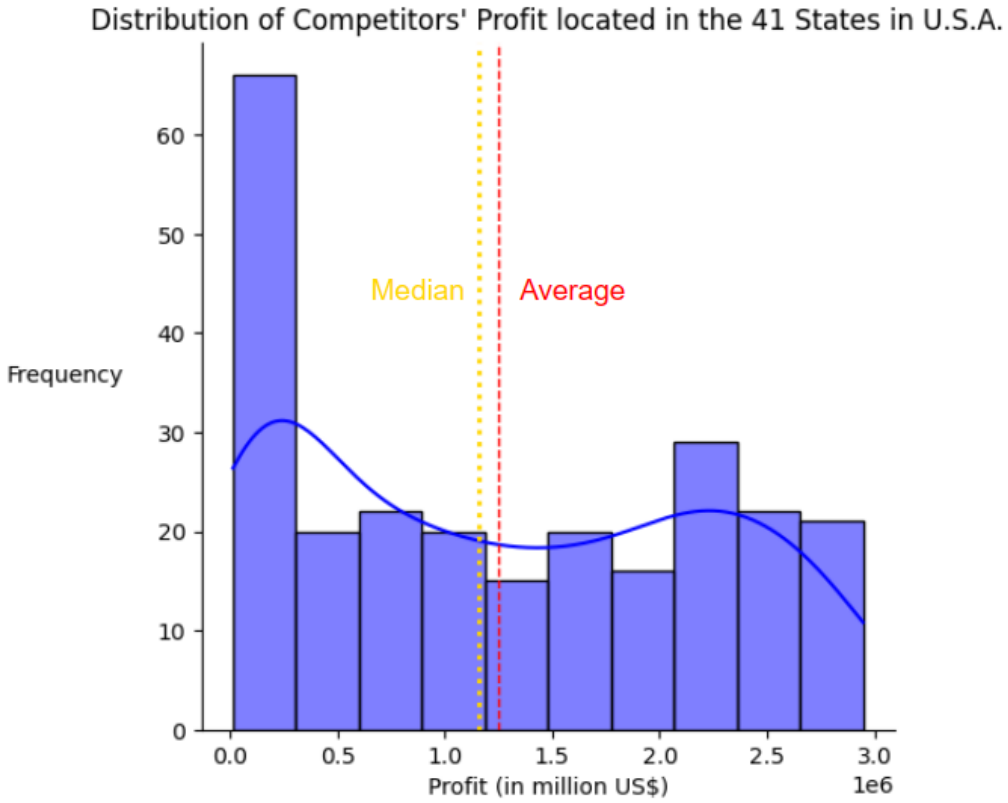
Python Chart 16

```
plt.figure(figsize=(8,2.5))
ax = sns.boxplot(data=competitors_revised, x="Profit", color="#0066FF", width=0.8, fliersize=5, linewidth=1, whis=1.5)
ax.set_title("Extreme Values for Competitors' Profit located in the 41 States in U.S.A.", pad=15)
ax.set_xlabel("Profit (US$)")
tick_labels = [f"{tick:,.0f}" for tick in ax.get_xticks()]
ax.set_xticklabels(tick_labels)
plt.show()
```



Python Chart 17

```
plt.figure(figsize=(8,4))
sns.displot(data=competitors_revised, x="Profit", kind="hist", kde=True, bins=10, color="blue").set(title="Distribution of Competitors' Profit located in the 41 States in U.S.A.")
plt.axvline(x=competitors_revised["Profit"].mean(), color="red", ls="--", lw=1)
plt.axvline(x=competitors_revised["Profit"].median(), color="gold", ls=":", lw=2)
plt.xlabel("Profit (in million US$)")
plt.ylabel("Frequency", rotation=0, labelpad=40)
plt.show()
```



Python Chart 18

11. Added new columns “Competitor Number” based on alphabetical order for easier tracking in case of multiple competitors in a state, “Operating Expenses” for combined values of the 3 expenses, and “Gross Income” under “competitors_revised” dataframe to analyze the competitors’ financial viability metrics.

```
competitors_revised["Competitor Number"] = competitors_revised.apply(lambda row: f"Competitor #{row.name + 1}", axis=1)
competitors_revised["Operating Expenses"] = competitors_revised["Research & Development Expenses"] + competitors_revised["Salary & Wages Expenses"] + competitors_revised["Marketing Expenses"]
competitors_revised["Gross Income"] = competitors_revised["Profit"] + competitors_revised["Operating Expenses"]
display(competitors_revised.head(5))
```

Python Output 45

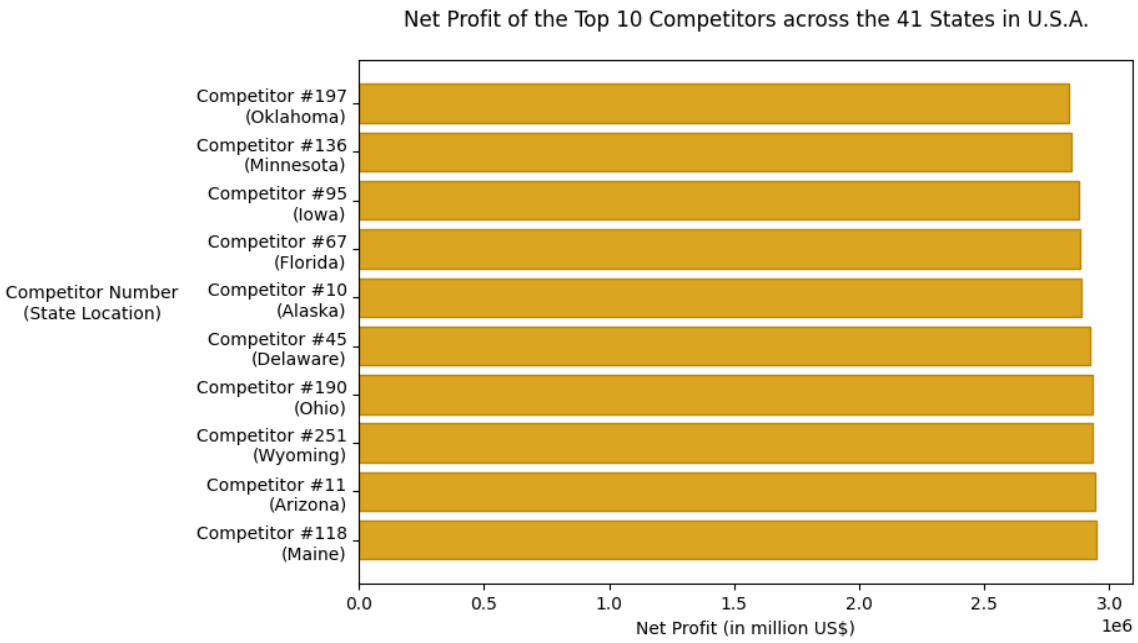
	Research & Development Expenses	Salary & Wages Expenses	Marketing Expenses	States	Profit	Competitor Number	Operating Expenses	Gross Income
0	901313.99	87164.9	87977.0	Alabama	120362.99	Competitor #1	1076455.89	1196818.88
1	352053.99	87072.9	31951.0	Alabama	398021.99	Competitor #2	471077.89	869099.88
2	933266.99	107860.9	51672.0	Alabama	1346449.99	Competitor #3	1092799.89	2439249.88
3	989534.99	73696.9	59152.0	Alabama	1397050.99	Competitor #4	1122383.89	2519434.88
4	1496277.99	104491.9	74515.0	Alabama	2430410.99	Competitor #5	1675284.89	4105695.88

```
comp_profit_bc = competitors_revised[["Competitor Number", "States", "Profit"]].sort_values(by="Profit", ascending=False).head(10)
display(comp_profit_bc)
```

	Competitor Number	States	Profit
117	Competitor #118	Maine	2946679.99
10	Competitor #11	Arizona	2942282.99
250	Competitor #251	Wyoming	2935405.99
189	Competitor #190	Ohio	2933777.99
44	Competitor #45	Delaware	2922433.99
9	Competitor #10	Alaska	2886772.99
66	Competitor #67	Florida	2882523.99
94	Competitor #95	Iowa	2879103.99
135	Competitor #136	Minnesota	2848832.99
196	Competitor #197	Oklahoma	2838170.99

Python Output 46

```
plt.figure(figsize=(8,5.5))
plt.barh(y=comp_profit_bc["Competitor Number"], width=comp_profit_bc["Profit"], color="goldenrod", edgecolor="darkgoldenrod")
plt.title("Net Profit of the Top 10 Competitors across the 41 States in U.S.A.", pad=20)
plt.xlabel("Net Profit (in million US$)")
plt.ylabel("Competitor Number\n(State Location)", rotation=0, labelpad=60)
plt.yticks(range(len(comp_profit_bc)), comp_profit_bc.apply(lambda row: f"{row['Competitor Number']}\n({row['States']})", axis=1))
plt.show()
```



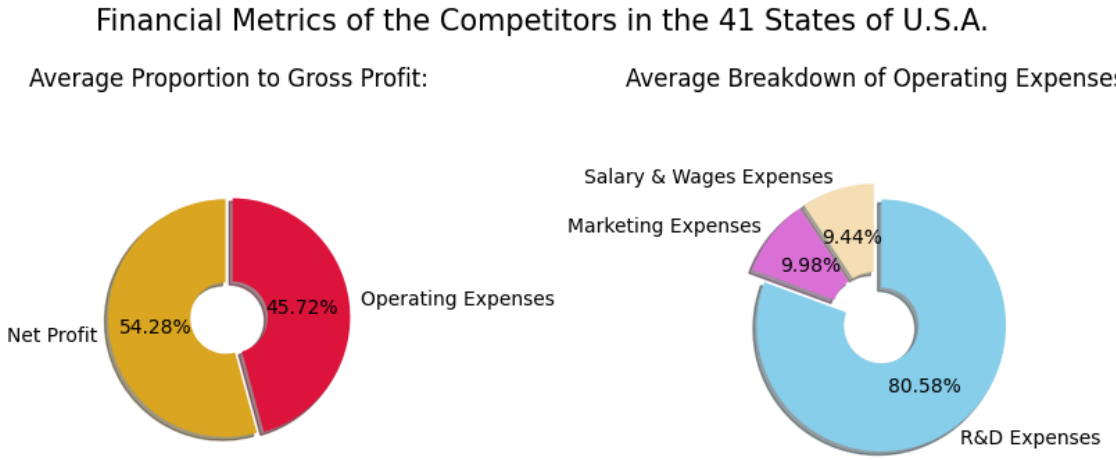
Python Chart 19

```
print("Average Gross Income:", competitors_revised["Gross Income"].mean())
print("Average Research & Development Expenses", competitors_revised["Research & Development Expenses"].mean())
print("Average Salary & Wages Expenses", competitors_revised["Salary & Wages Expenses"].mean())
print("Average Marketing Expenses", competitors_revised["Marketing Expenses"].mean())
print("Average Operating Expenses", competitors_revised["Operating Expenses"].mean())
print("Average Profit", competitors_revised["Profit"].mean())
comp_gi_dc_list = []
comp_oe_dc_list = []
comp_gi_dc_list.extend([competitors_revised["Operating Expenses"].mean(), competitors_revised["Profit"].mean()])
comp_oe_dc_list.extend([competitors_revised["Salary & Wages Expenses"].mean(), competitors_revised["Marketing Expenses"].mean(), competitors_revised["Research & Development Expenses"].mean()])
comp_gi_dc_labels = ["Operating Expenses", "Net Profit"]
comp_oe_dc_labels = ["Salary & Wages Expenses", "Marketing Expenses", "R&D Expenses"]
comp_gi_dc_explode = [0.03, 0.03]
comp_oe_dc_explode = [0.07, 0.07, 0.07]
```

```
Average Gross Income: 2306824.990478087
Average Research & Development Expenses 849775.2696015935
Average Salary & Wages Expenses 99557.2493625498
Average Marketing Expenses 105288.08541832668
Average Operating Expenses 1054620.6043824703
Average Profit 1252204.3860956174
```

Python Output 47

```
fig, (lf_ax, rt_ax) = plt.subplots(1,2, figsize=(8,4))
plt.suptitle("Financial Metrics of the Competitors in the 41 States of U.S.A.", fontsize=15)
lf_wedges, lf_texts, lf_autopct = lf_ax.pie(comp_gi_dc_list, labels=comp_gi_dc_labels, colors=["crimson", "goldenrod"],
explode=comp_gi_dc_explode, autopct="%1.2f%%", startangle=90, wedgeprops=dict(width=0.7), counterclock=False,
shadow=True)
lf_ax.set_title("Average Proportion to Gross Profit:", loc="center")
lf_ax.axis("equal")
rt_wedges, rt_texts, rt_autopct = rt_ax.pie(comp_oe_dc_list, labels=comp_oe_dc_labels, colors=["wheat", "orchid",
"skyblue"], explode=comp_oe_dc_explode, autopct="%1.2f%%", startangle=90, wedgeprops=dict(width=0.7),
shadow=True)
rt_ax.set_title("Average Breakdown of Operating Expenses:", loc="center")
rt_ax.axis("equal")
plt.tight_layout()
plt.show()
```



Python Chart 20

```
competitors_revised["R&D Expense Ratio"] = competitors_revised["Research & Development Expenses"] /
competitors_revised["Operating Expenses"]
competitors_revised["Salary & Wages Expense Ratio"] = competitors_revised["Salary & Wages Expenses"] /
competitors_revised["Operating Expenses"]
competitors_revised["Marketing Expense Ratio"] = competitors_revised["Marketing Expenses"] /
competitors_revised["Operating Expenses"]
competitors_revised["Operating Expense Ratio"] = competitors_revised["Operating Expenses"] /
competitors_revised["Gross Income"]
competitors_revised["Net Profit Coverage Ratio"] = competitors_revised["Profit"] / competitors_revised["Operating
Expenses"]
display(competitors_revised.head(5))
```

Python Output 48

	Research & Development Expenses	Salary & Wages Expenses	Marketing Expenses	States	Profit	Competitor Number	Operating Expenses	Gross Income	R&D Expense Ratio	Salary & Wages Expense Ratio	Marketing Expense Ratio	Operating Expense Ratio	Net Profit Coverage Ratio
0	901313.99	87164.9	87977.0	Alabama	120362.99	Competitor #1	1076455.89	1196818.88	0.837298	0.080974	0.081728	0.899431	0.111814
1	352053.99	87072.9	31951.0	Alabama	398021.99	Competitor #2	471077.89	869099.88	0.747337	0.184838	0.067825	0.542030	0.844918
2	933266.99	107860.9	51672.0	Alabama	1346449.99	Competitor #3	1092799.89	2439249.88	0.854015	0.098701	0.047284	0.448007	1.232110
3	989534.99	73696.9	59152.0	Alabama	1397050.99	Competitor #4	1122383.89	2519434.88	0.881637	0.065661	0.052702	0.445490	1.244718
4	1496277.99	104491.9	74515.0	Alabama	2430410.99	Competitor #5	1675284.89	4105695.88	0.893148	0.062373	0.044479	0.408039	1.450745

```
display(competitors_revised[["Competitor Number", "States", "Operating Expense Ratio", "Net Profit Coverage Ratio"]].sort_values(by="Net Profit Coverage Ratio", ascending=False))
```

	Competitor Number	States	Operating Expense Ratio	Net Profit Coverage Ratio
211	Competitor #212	Pennsylvania	0.071219	13.041153
86	Competitor #87	Illinois	0.080714	11.389474
249	Competitor #250	Wyoming	0.106825	8.361089
126	Competitor #127	Massachusetts	0.110295	8.066587
43	Competitor #44	Delaware	0.128926	6.756414
...	...	...	...	...
67	Competitor #68	Georgia	0.926975	0.078778
112	Competitor #113	Maine	0.932407	0.072493
241	Competitor #242	Wyoming	0.938328	0.065725
75	Competitor #76	Idaho	0.944644	0.058599
122	Competitor #123	Massachusetts	0.948602	0.054183

251 rows × 4 columns

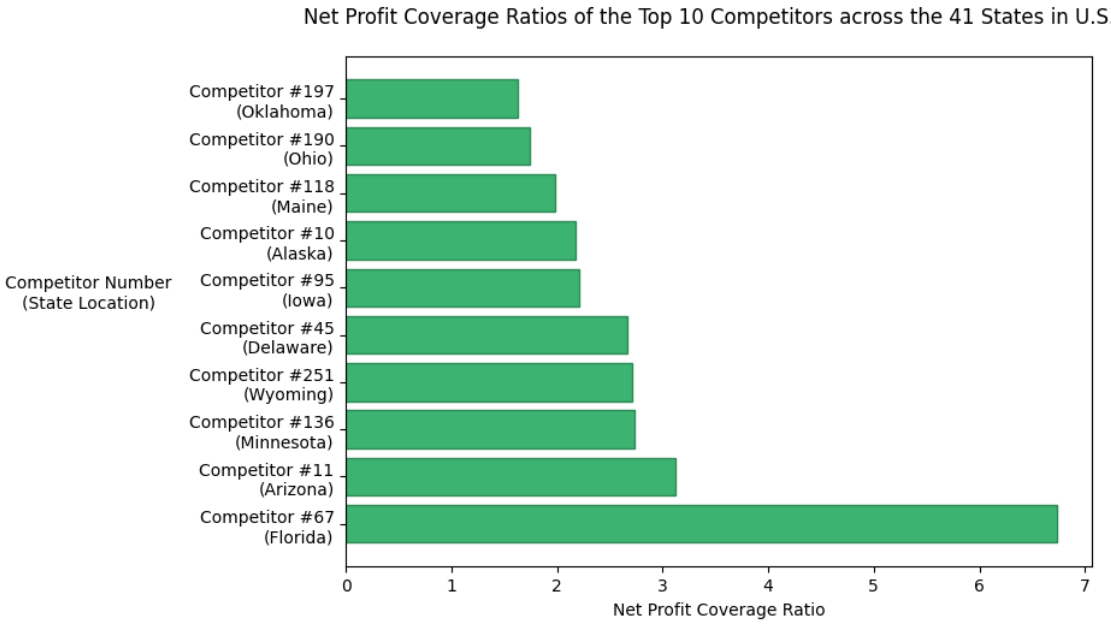
Python Output 49

```
comp_npcr_bc = competitors_revised[["Competitor Number", "States", "Profit", "Net Profit Coverage Ratio"]].sort_values(by="Profit", ascending=False).head(10)
comp_npcr_bc_sorted = comp_npcr_bc.sort_values(by="Net Profit Coverage Ratio", ascending=False)
display(comp_npcr_bc_sorted)
```

	Competitor Number	States	Profit	Net Profit Coverage Ratio
66	Competitor #67	Florida	2882523.99	6.732024
10	Competitor #11	Arizona	2942282.99	3.126480
135	Competitor #136	Minnesota	2848832.99	2.731413
250	Competitor #251	Wyoming	2935405.99	2.707827
44	Competitor #45	Delaware	2922433.99	2.668931
94	Competitor #95	Iowa	2879103.99	2.206827
9	Competitor #10	Alaska	2886772.99	2.177882
117	Competitor #118	Maine	2946679.99	1.981807
189	Competitor #190	Ohio	2933777.99	1.738344
196	Competitor #197	Oklahoma	2838170.99	1.630596

Python Output 50

```
plt.figure(figsize=(8,5.5))
plt.barh(y=comp_npcr_bc_sorted["Competitor Number"], width=comp_npcr_bc_sorted["Net Profit Coverage Ratio"], color="mediumseagreen", edgecolor="seagreen")
plt.title("Net Profit Coverage Ratios of the Top 10 Competitors across the 41 States in U.S.A.", pad=20)
plt.xlabel("Net Profit Coverage Ratio")
plt.ylabel("Competitor Number\n(State Location)", rotation=0, labelpad=60)
plt.yticks(range(len(comp_npcr_bc_sorted)), comp_npcr_bc_sorted.apply(lambda row: f"{row['Competitor Number']}\n({row['States']})", axis=1))
plt.show()
```



Python Chart 21

12. Explored hypothesis testing for the Average Competitors' Profit is significantly more than the Median Competitors' Profit (rounded in nearest ten-thousand), California's Total Population exceeds 12% of the total U.S.A. states population, and distribution of the 50 states across 4 regions are the same or different.

```
print("Skewness of Competitors' Profit:", competitors_revised.groupby("Competitor Number")["Profit"].mean().skew(axis=0), "(Symmetric Skewness)")
print("Average of Competitors' Profit: US$ {:,.2f}".format(competitors_revised["Profit"].mean()))
print("----------------------------------------------------------------------")
"""
Null Hypothesis: The Average of Competitors' Profit is not more than US$ 1,150,000.00. (Average Competitors' Profit <= US$ 1,150,000.00)
Alternative Hypothesis: The Average of Competitors' Profit is more than US$ 1,150,000.00. (Average Competitors' Profit > US$ 1,150,000.00)
"""
alpha = 0.05
ttest_critical_value = 1.651
print("Critical Value:", ttest_critical_value)
t_stat, p_val = ttest_1samp(competitors_revised.groupby("Competitor Number")["Profit"].mean(), 1150000.00, alternative="greater")
print("Test Statistics:", t_stat)
print("p-value:", p_val)
print("----------------------------------------------------------------------")
if p_val < alpha:
  print(f"Reject the Null Hypothesis (alpha = {alpha}).\nThe Average of Competitors' Profit is evidently more than US$ 1,150,000.00.")
else:
  print(f"Fail to Reject the Null Hypothesis (alpha = {alpha}).\nThe Average of Competitors' Profit is not evidently more than US$ 1,150,000.00.")
```

```
Skewness of Competitors' Profit: 0.21635148954105965 (Symmetric Skewness)
Average of Competitors' Profit: US$ 1,252,204.39
----------------------------------------------------------------------
Critical Value: 1.651
Test Statistics: 1.7048897403033882
p-value: 0.04472862224564088
----------------------------------------------------------------------
Reject the Null Hypothesis (alpha = 0.05).
The Average of Competitors' Profit is evidently more than US$ 1,150,000.00.
```

Python Output 51

```
print("California's Population Proportion", round((us_states["Population Size"][us_states["States"] == "California"].sum() / us_states["Population Size"].sum() * 100), 2), "%")
print("----------------------------------------")
"""
Null Hypothesis: The Proportion of California's Population is not more than 12% of the total population of the 50 states. (California's Population Proportion <= 0.12)
Alternative Hypothesis: The Proportion of California's Population is more than 12% of the total population of the 50 states. (California's Population Proportion > 0.12)
"""
alpha = 0.05
california_population = us_states["Population Size"][us_states["States"] == "California"].sum()
total_population = us_states["Population Size"].sum()
proportions_ztest_critical_value = 1.645
print("Critical Value:", proportions_ztest_critical_value)
z_stat, p_val = proportions_ztest(count=california_population, nobs=total_population, value=0.12, alternative="larger")
print("Test Statistics:", z_stat)
print("p-value:", p_val)
print("----------------------------------------")
if p_val < alpha:
  print(f"Reject the Null Hypothesis (alpha = {alpha}).\nThe Proportion of California's Population is evidently more than 12% of the total population of the 50 states.")
else:
  print(f"Fail to Reject the Null Hypothesis (alpha = {alpha}).\nThe Proportion of California's Population is not evidently more than 12% of the total population of the 50 states.")
```

```
California's Population Proportion 12.06 %
----------------------------------------
Critical Value: 1.645
Test Statistics: 35.31573202647204
p-value: 1.6840453505098866e-273
----------------------------------------
Reject the Null Hypothesis (alpha = 0.05).
The Proportion of California's Population is evidently more than 12% of the total population of the 50 states.
```

Python Output 52

```
display("Country Origins:", us_states["Region"].value_counts())
print("--------------------------------------------------")
"""
Null Hypothesis: The distribution of the states in 4 Regions are equal. (Northeast = Midwest = West = South)
Alternative Hypothesis: The distribution of the states in 4 Regions are different. (Northeast <> Midwest <> West <> South)
"""
alpha = 0.05
obs = us_states["Region"].value_counts()
n_obs = len(us_states)
f_obs = obs.values
chisquare_critical_value = 2
print("Critical Value (Lower & Upper):", chisquare_critical_value)
chi_sq, p_val = chisquare(f_obs=f_obs, f_exp=None)
print("Test Statistics:", chi_sq)
print("p-value:", p_val)
print("--------------------------------------------------")
if p_val < alpha:
  print(f"Reject the Null Hypothesis (alpha = {alpha}).\nThere is enough evidence to prove that the distribution of the states in 4 Regions are different.")
else:
  print(f"Fail to Reject the Null Hypothesis (alpha = {alpha}).\nThere is not enough evidence to prove that the distribution of the states in 4 Regions are different.")
```

```
'Country Origins:'
South        16
West         13
Midwest      12
Northeast     9
Name: Region, dtype: int64
--------------------------------------------------
Critical Value (Lower & Upper): 2
Test Statistics: 2.0
p-value: 0.5724067044708798
--------------------------------------------------
Fail to Reject the Null Hypothesis (alpha = 0.05).
There is not enough evidence to prove that the distribution of the states in 4 Regions are different.
```

Python Output 53

IV. Data Saving Process in Google Colab:

1. Rearranged the column orders of “us_states” and “competitors_revised” dataframes.

```
htsb_us_states_final = us_states[["States", "Region", "Minimum Income", "Average Income", "Maximum Income", "Corruption Rate", "Population Size", "Minimum Healthcare Spending", "Average Healthcare Spending", "Maximum Healthcare Spending", "Minimum Property Price", "Average Property Price", "Maximum Property Price"]]
display(htsb_us_states_final)
```

Python Output 54

	States	Region	Minimum Income	Average Income	Maximum Income	Corruption Rate	Population Size	Minimum Healthcare Spending	Average Healthcare Spending	Maximum Healthcare Spending	Minimum Property Price	Average Property Price	Maximum Property Price
0	Alabama	South	23999.0	51113.0	96993.0	2.15	4903185	50.0	200.50	500.0	1200.0	1797.50	2500.0
1	Alaska	West	35219.0	76440.0	134318.0	1.06	731545	100.0	300.25	750.0	2000.0	2684.00	3500.0
2	Arizona	West	29466.0	62283.0	113589.0	1.40	7278717	25.0	150.00	300.0	1500.0	2356.75	4000.0
3	Arkansas	South	23028.0	48829.0	90052.0	3.02	3017804	75.0	175.00	400.0	1000.0	1499.25	2500.0
4	California	West	37698.0	80440.0	149265.0	1.09	39512223	50.0	250.75	600.0	3500.0	5832.50	9000.0
5	Colorado	West	35636.0	76240.0	130714.0	0.80	5758736	100.0	225.50	500.0	2000.0	2987.25	4500.0
6	Connecticut	Northeast	37426.0	79287.0	142596.0	2.01	3565287	150.0	300.00	700.0	3000.0	3837.00	5500.0
7	Delaware	South	30544.0	64040.0	120324.0	1.08	973764	75.0	175.50	350.0	1500.0	2289.75	3500.0
8	Florida	South	27064.0	58108.0	105773.0	1.65	21477737	50.0	175.25	400.0	1500.0	2763.50	5000.0
9	Georgia	South	27609.0	58932.0	112609.0	1.60	10617423	75.0	200.00	450.0	1200.0	2065.00	3500.0
10	Hawaii	West	36987.0	78084.0	130299.0	0.43	1415872	200.0	350.00	800.0	4500.0	5975.50	7500.0
11	Idaho	West	25119.0	53545.0	92625.0	1.12	1787065	50.0	125.50	300.0	800.0	1382.00	2500.0
12	Illinois	Midwest	32737.0	70387.0	126359.0	1.27	12671821	100.0	225.75	500.0	1200.0	2056.50	3500.0
13	Indiana	Midwest	27050.0	57881.0	97462.0	0.90	6732219	75.0	200.25	400.0	1000.0	1704.75	3000.0
14	Iowa	Midwest	28709.0	62075.0	101572.0	0.58	3155070	50.0	150.00	250.0	1000.0	1442.25	2000.0
15	Kansas	Midwest	27487.0	59046.0	100276.0	1.02	2913314	50.0	175.50	350.0	1000.0	1479.50	2500.0
16	Kentucky	South	23915.0	50675.0	88278.0	1.60	4467673	75.0	175.00	400.0	1000.0	1605.00	3000.0
17	Louisiana	South	23855.0	50686.0	93827.0	3.72	4648794	75.0	200.00	450.0	1000.0	1718.50	3000.0
18	Maine	Northeast	25964.0	54927.0	98362.0	0.48	1344212	100.0	225.75	500.0	1500.0	2236.00	3500.0
19	Maryland	South	42343.0	89392.0	167535.0	1.38	6045680	150.0	300.00	700.0	2000.0	3122.75	5000.0
20	Massachusetts	Northeast	38980.0	82427.0	144960.0	2.27	6892503	200.0	350.00	800.0	3000.0	4136.50	6000.0
21	Michigan	Midwest	28357.0	61347.0	111915.0	1.00	9986857	100.0	225.50	500.0	1000.0	1812.75	3500.0

```
htsb_competitors_final = competitors_revised[["Competitor Number", "States", "Gross Income", "Research & Development Expenses", "Salary & Wages Expenses", "Marketing Expenses", "Operating Expenses", "Profit", "R&D Expense Ratio", "Salary & Wages Expense Ratio", "Marketing Expense Ratio", "Operating Expense Ratio", "Net Profit Coverage Ratio"]]
display(htsb_competitors_final)
```

Python Output 55

	Competitor Number	States	Gross Income	Research & Development Expenses	Salary & Wages Expenses	Marketing Expenses	Operating Expenses	Profit	R&D Expense Ratio	Salary & Wages Expense Ratio	Marketing Expense Ratio	Operating Expense Ratio	Net Profit Coverage Ratio
0	Competitor #1	Alabama	1196818.88	901313.99	87164.9	87977.0	1076455.89	120362.99	0.837298	0.080974	0.081728	0.899431	0.111814
1	Competitor #2	Alabama	869099.88	352053.99	87072.9	31951.0	471077.89	398021.99	0.747337	0.184838	0.067825	0.542030	0.844918
2	Competitor #3	Alabama	2439249.88	933266.99	107860.9	51672.0	1092799.89	1346449.99	0.854015	0.098701	0.047284	0.448007	1.232110
3	Competitor #4	Alabama	2519434.88	989534.99	73696.9	59152.0	1122383.89	1397050.99	0.881637	0.065661	0.052702	0.445490	1.244718
4	Competitor #5	Alabama	4105695.88	1496277.99	104491.9	74515.0	1675284.89	2430410.99	0.893148	0.062373	0.044479	0.408039	1.450745
...	...	...	...	...	...	...	...	...	...	...	...	...	...
246	Competitor #247	Wyoming	2402688.88	189440.99	134720.9	54263.0	378424.89	2024263.99	0.500604	0.356004	0.143392	0.157501	5.349183
247	Competitor #248	Wyoming	2997272.88	637793.99	75079.9	107567.0	820440.89	2176831.99	0.777380	0.091512	0.131109	0.273729	2.653247
248	Competitor #249	Wyoming	3195401.88	753705.99	135318.9	119221.0	1008245.89	2187155.99	0.747542	0.134212	0.118246	0.315530	2.169268
249	Competitor #250	Wyoming	3115097.88	131883.99	87447.9	113439.0	332770.89	2782326.99	0.396321	0.262787	0.340892	0.106825	8.361089
250	Competitor #251	Wyoming	4019450.88	861790.99	107318.9	114935.0	1084044.89	2935405.99	0.794977	0.098999	0.106024	0.269700	2.707827

251 rows × 13 columns

2. Saved the 2 dataframes as .csv file named “htsb_us_states_final” and “htsb_competitors_final”.

```
htsb_us_states_final.to_csv("/content/drive/MyDrive/Colab Notebooks/Refocus Final Project (Submitted Documents)/04- Exported Data Output CSV File/htsb_us_states_final.csv", index=False)
```

Python Figure 2


```
htsb_competitors_final.to_csv("/content/drive/MyDrive/Colab Notebooks/Refocus Final Project (Submitted Documents)/04- Exported Data Output CSV File/htsb_competitors_final.csv", index=True)
```

Python Figure 3

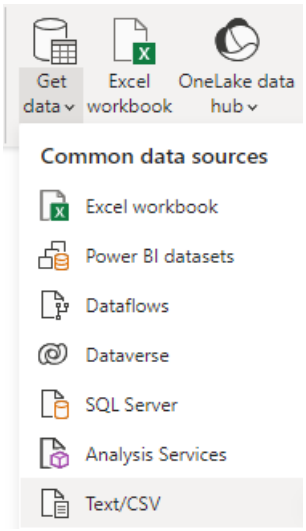
3. Saved the Google Colab notebook as .ipynb file named “Analysis using Python Codes”.

Python Figure 4

Power BI

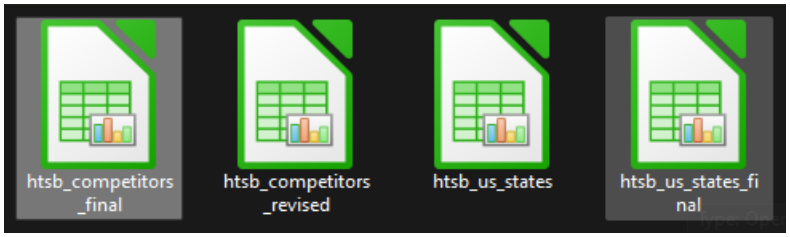
I. Data Preparation in Power BI:

1. Loaded the 2 updated .csv files “htsb_us_states_final” and “htsb_competitors_final” from the “Exported Data Output CSV File” folder in Power BI.



Power BI Figure 1

Power BI Figure 2



Power BI Figure 3

htsb_us_states_final.csv

File Origin: 1252: Western European (Windows) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

States	Region	Minimum Income	Average Income	Maximum Income	Corruption Rate	Population Size	Minimum Healthcare Spe
Alabama	South	23999	51113	96993	2.15	4903185	
Alaska	West	35219	76440	134318	1.06	731545	
Arizona	West	29466	62283	113589	1.4	7278717	
Arkansas	South	23028	48829	90052	3.02	3017804	
California	West	37698	80440	149265	1.09	39512223	
Colorado	West	35636	76240	130714	0.8	5758736	
Connecticut	Northeast	37426	79287	142596	2.01	3565287	
Delaware	South	30544	64040	120324	1.08	973764	
Florida	South	27064	58108	105773	1.65	21477737	
Georgia	South	27609	58932	112609	1.6	10617423	
Hawaii	West	36987	78084	130299	0.43	1415872	
Idaho	West	25119	53545	92625	1.12	1787065	
Illinois	Midwest	32737	70387	126359	1.27	12671821	
Indiana	Midwest	27050	57881	97462	0.9	6732219	
Iowa	Midwest	28709	62075	101572	0.58	3155070	
Kansas	Midwest	27487	59046	100276	1.02	2913314	
Kentucky	South	23915	50675	88278	1.6	4467673	
Louisiana	South	23855	50686	93827	3.72	4648794	
Maine	Northeast	25964	54927	98362	0.48	1344212	
Maryland	South	42343	89392	167535	1.38	6045680	

Extract Table Using Examples | Load | Transform Data | Cancel

Power BI Figure 4

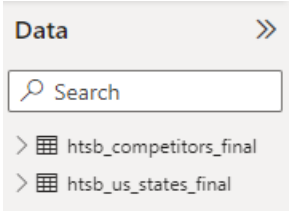
htsb_competitors_final.csv

File Origin: 1252: Western European (Windows) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

	Competitor Number	States	Gross Income	Research & Development Expenses	Salary & Wages Expenses	Marketing Expenses	Ope
0	Competitor #1	Alabama	1196818.88	901313.99	87164.9	87977	
1	Competitor #2	Alabama	869099.88	352053.99	87072.9	31951	
2	Competitor #3	Alabama	2439249.88	933266.99	107860.9	51672	
3	Competitor #4	Alabama	2519434.88	989534.99	73696.9	59152	
4	Competitor #5	Alabama	4105695.88	1496277.99	104491.9	74515	
5	Competitor #6	Alaska	1790767.88	722966.99	129074.9	116675	
6	Competitor #7	Alaska	2214026.88	155083.99	128398.9	63666	
7	Competitor #8	Alaska	3402090.88	1155299.99	51488.9	93013	
8	Competitor #9	Alaska	3510113.88	1172907.99	68409.9	101154	
9	Competitor #10	Alaska	4212268.88	1116726.99	127485.9	81283	
10	Competitor #11	Arizona	3883367.88	771423.99	129046.9	40614	
11	Competitor #12	Arkansas	1444374.88	1116497.99	110053.9	82921	
12	Competitor #13	Arkansas	1714050.88	207272.99	87555.9	66206	
13	Competitor #14	California	176838.26	0	116983.8	45173.06	
14	Competitor #15	California	177986.65	0	135426.92	0	
15	Competitor #16	California	270518.93	22177.74	154806.14	28334.72	
16	Competitor #17	California	339330.16	23640.93	96189.63	148001.11	
17	Competitor #18	California	398335.96	28754.33	118546.05	172795.67	
18	Competitor #19	California	377545.66	38558.51	82982.09	174999.3	
19	Competitor #20	California	382331.65	44069.95	51283.14	197029.42	

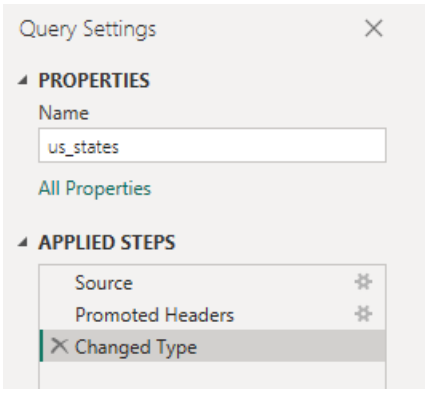
Extract Table Using Examples | Load | Transform Data | Cancel

Power BI Figure 5

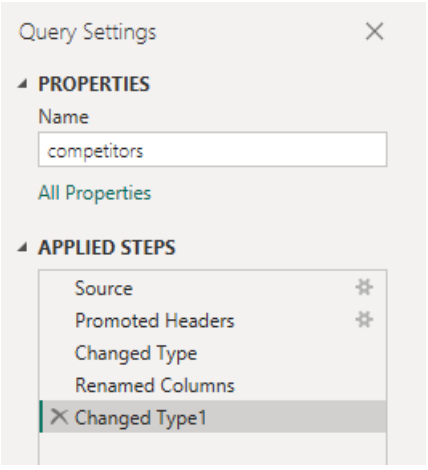


Power BI Figure 6

2. Went to Power Query Editor to change the data types and renamed the queries after importing the 2 files.



Power BI Figure 7



Power BI Figure 8

II. Data Analysis in Power BI:

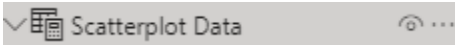
1. Added a new calculated column named “Region” in “competitors” table.

```
Region = LOOKUPVALUE(us_states[Region], us_states[States], competitors[States])
```

Power BI Figure 9

2. Created a new calculated table named “Scatterplot Data” that includes only the selected columns to be used for calculating the correlation between corruption conviction level vs average profits of the competitors by state locations then created a calculated measure (quick measure under calculations group) named “Correlation Coefficient Corruption Rate vs Average Profit” to measure the correlation value.

```
Scatterplot Data = SELECTCOLUMNS(competitors, "States", competitors[States], "Profit", competitors[Profit], "Corruption Rate", LOOKUPVALUE(us_states[Corruption Rate], us_states[States], competitors[States]), "Competitor Number", competitors[Competitor Number], "Region", competitors[Region])
```



Power BI Figure 10

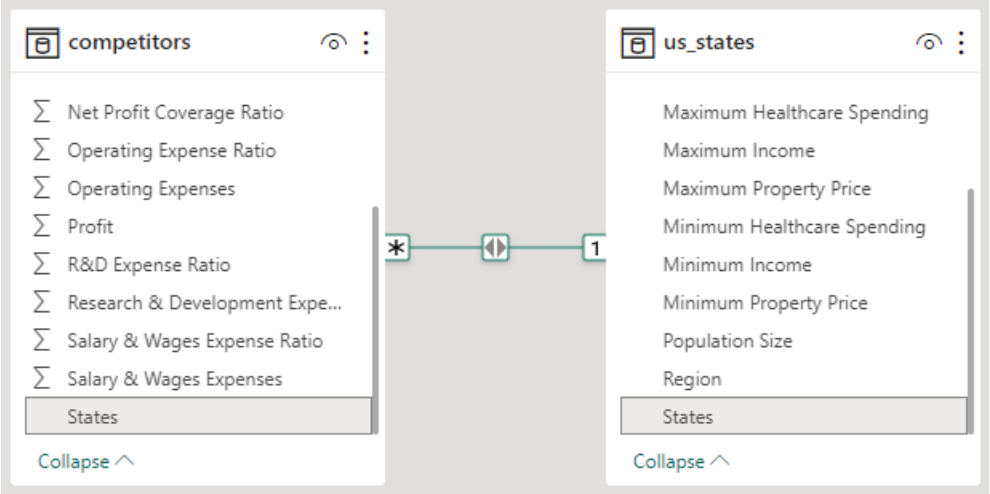
```
Correlation Coefficient Corruption Rate vs Average Profit =
VAR __CORRELATION_TABLE = VALUES('Scatterplot Data'[States])
VAR __COUNT =
    COUNTX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(
            AVERAGE('Scatterplot Data'[Corruption Rate])
                * AVERAGE('Scatterplot Data'[Profit])
        )
    )
VAR __SUM_X =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Scatterplot Data'[Corruption Rate]))
    )
VAR __SUM_Y =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Scatterplot Data'[Profit]))
    )
VAR __SUM_XY =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(
            AVERAGE('Scatterplot Data'[Corruption Rate])
                * AVERAGE('Scatterplot Data'[Profit]) * 1.
        )
    )
VAR __SUM_X2 =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Scatterplot Data'[Corruption Rate]) ^ 2)
    )
VAR __SUM_Y2 =
    SUMX(
        KEEPFILTERS(__CORRELATION_TABLE),
        CALCULATE(AVERAGE('Scatterplot Data'[Profit]) ^ 2)
    )
RETURN
    DIVIDE(
        __COUNT * __SUM_XY - __SUM_X * __SUM_Y * 1.,
        SQRT(
            (__COUNT * __SUM_X2 - __SUM_X ^ 2)
                * (__COUNT * __SUM_Y2 - __SUM_Y ^ 2)
        )
    )
```

Power BI Figure 11

III. Data Modeling in Power BI:

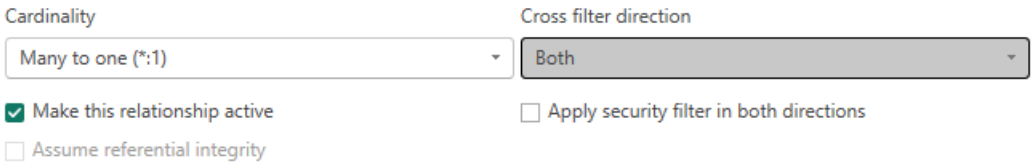
1. Created “Schema” page in the model view.

2. Created a relationship between “competitors” fact table and “us_states” dimension table on “States” columns.



Power BI Figure 12

3. Set cross-filter direction to both directions and the cardinality to many-to-one.



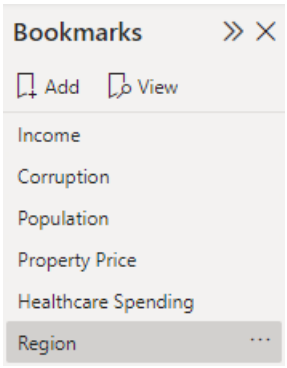
Power BI Figure 13

IV. Data Visualization in Power BI:

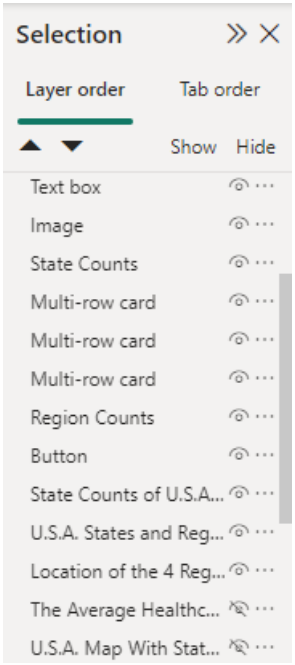
1. Created several maps, bar charts, filled map, pie chart, table, funnel charts, stacked bar chart, scatterplot, as well as various cards, multi-row cards, and gauge charts for the 3 pages named “State Locations and Statistics”, “Competitors”, and “Recommended State” in the report view.

2. Applied bookmarking buttons by creating “Region”, “Income”, “Corruption”, “Population”, “Property Price”, and “Healthcare Spending” for both buttons and their corresponding bookmarks in the “State Locations and Statistics” page.

Power BI Figure 14

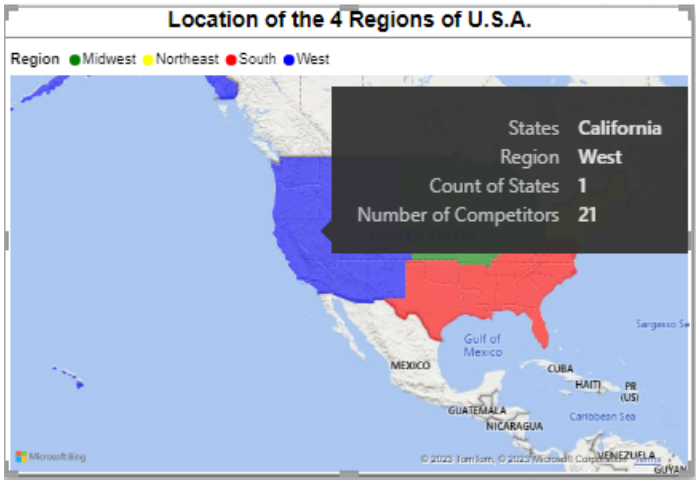


Power BI Figure 15



Power BI Figure 16

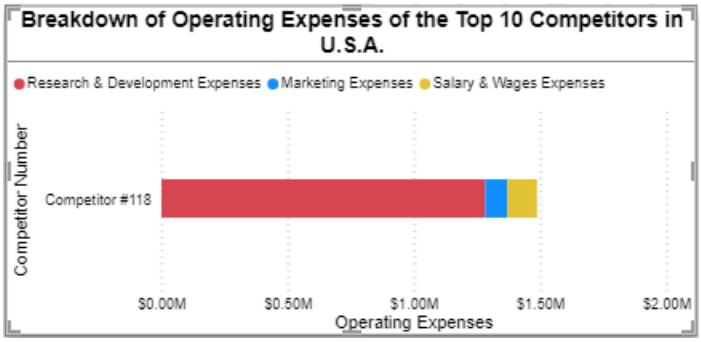
3. Added additional tooltips to provide more information on the data points for the “State Locations and Statistics” and “Competitors” pages in the report view.

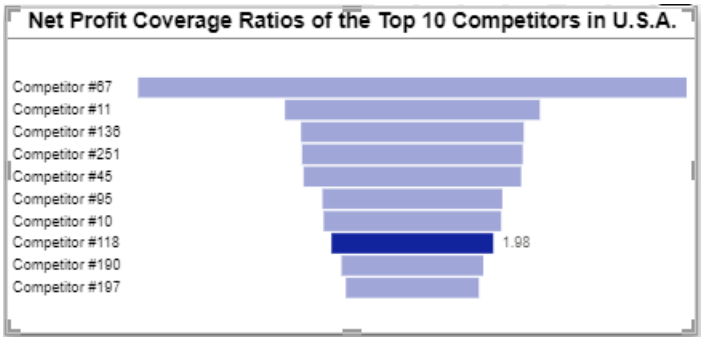


Power BI Chart 1

4. Edited the interactions between visuals by applying cross-filtering and cross-highlighting for “State Locations and Statistics” and “Competitors” pages in the report view.

Power BI Charts 2-3

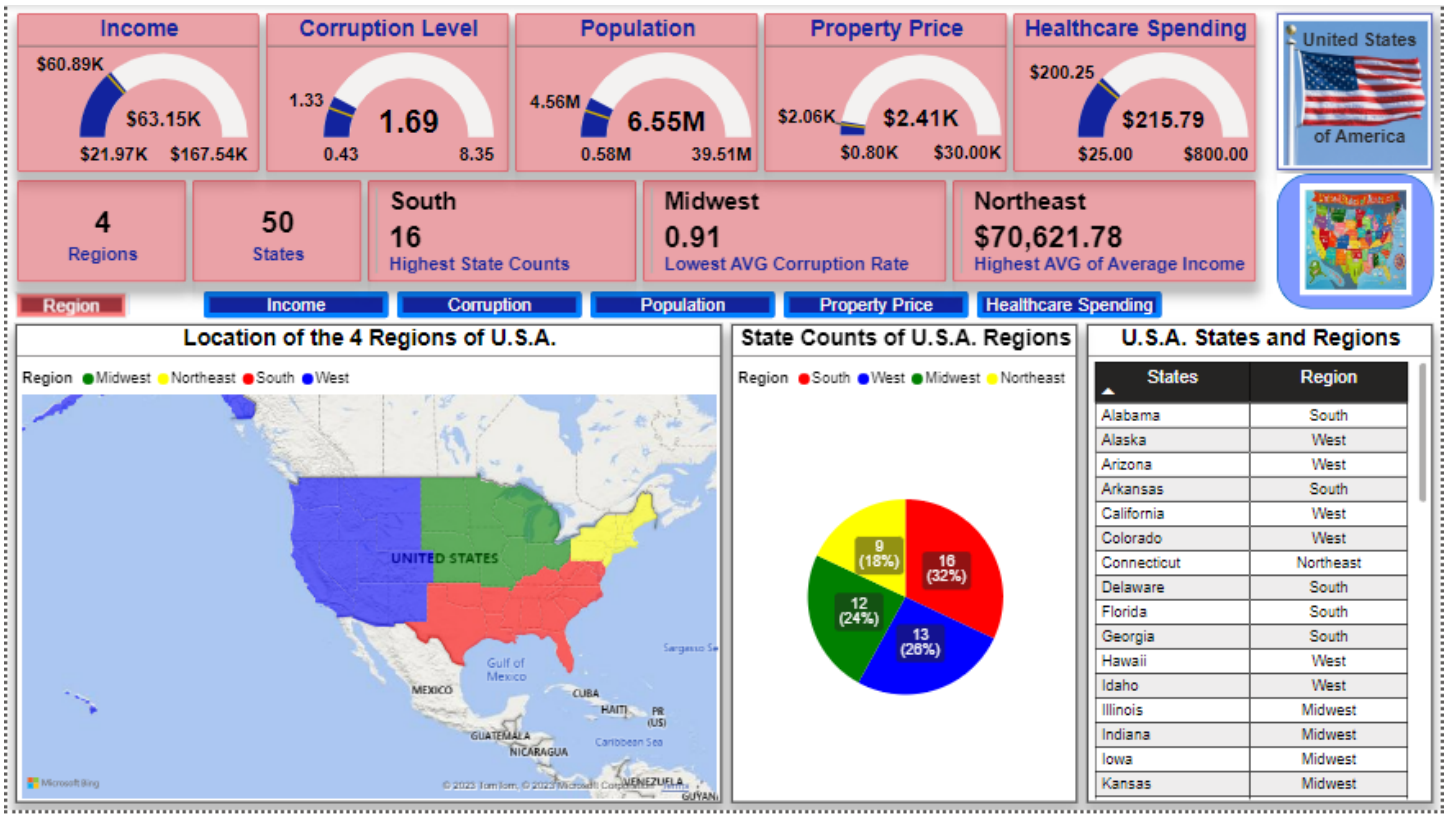




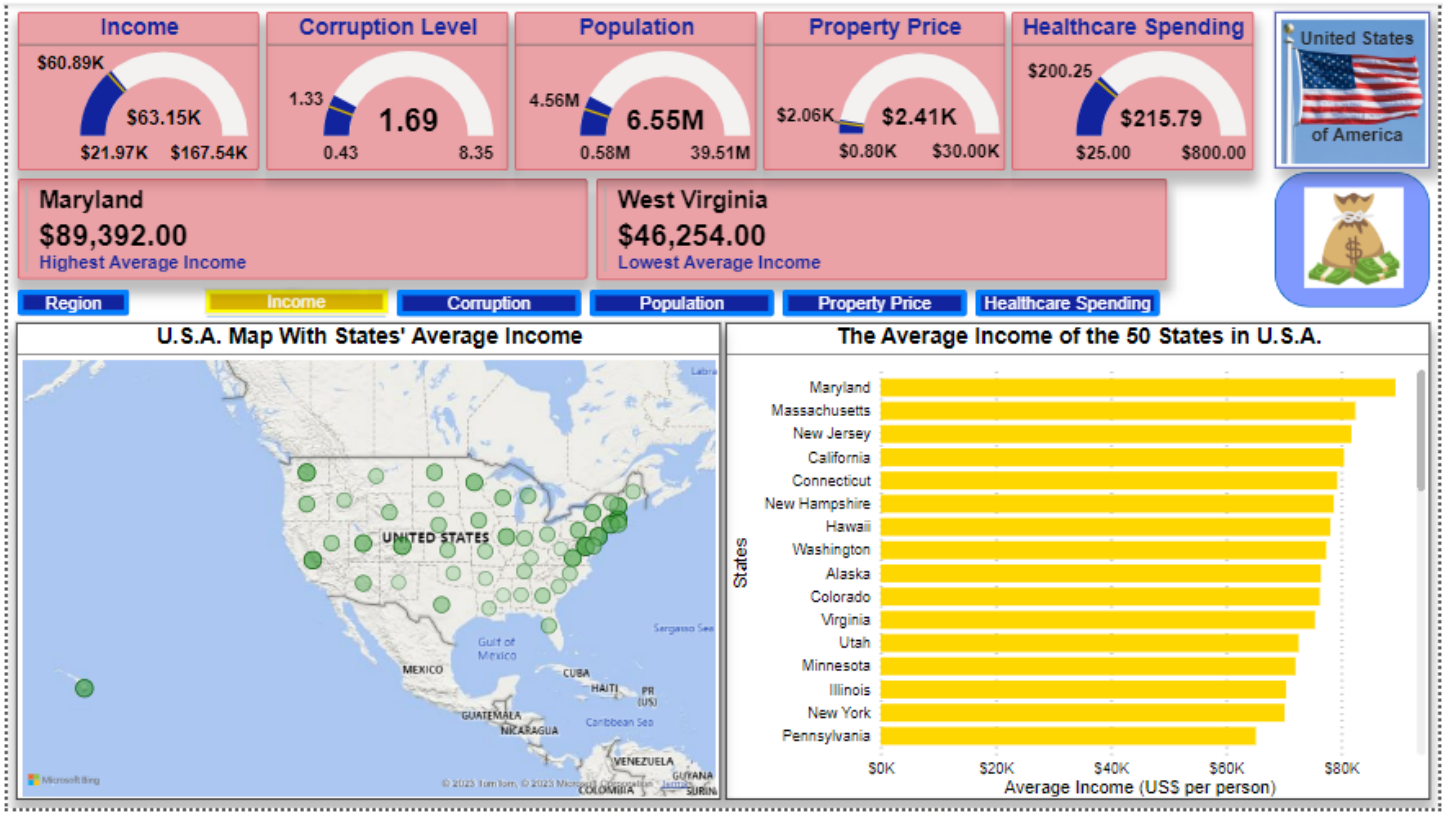
5. Used several images and shapes to make the dashboards more appealing.

6. Organized the charts into 3 dashboards (“State Locations and Statistics”, “Competitors”, and “Recommended State”).

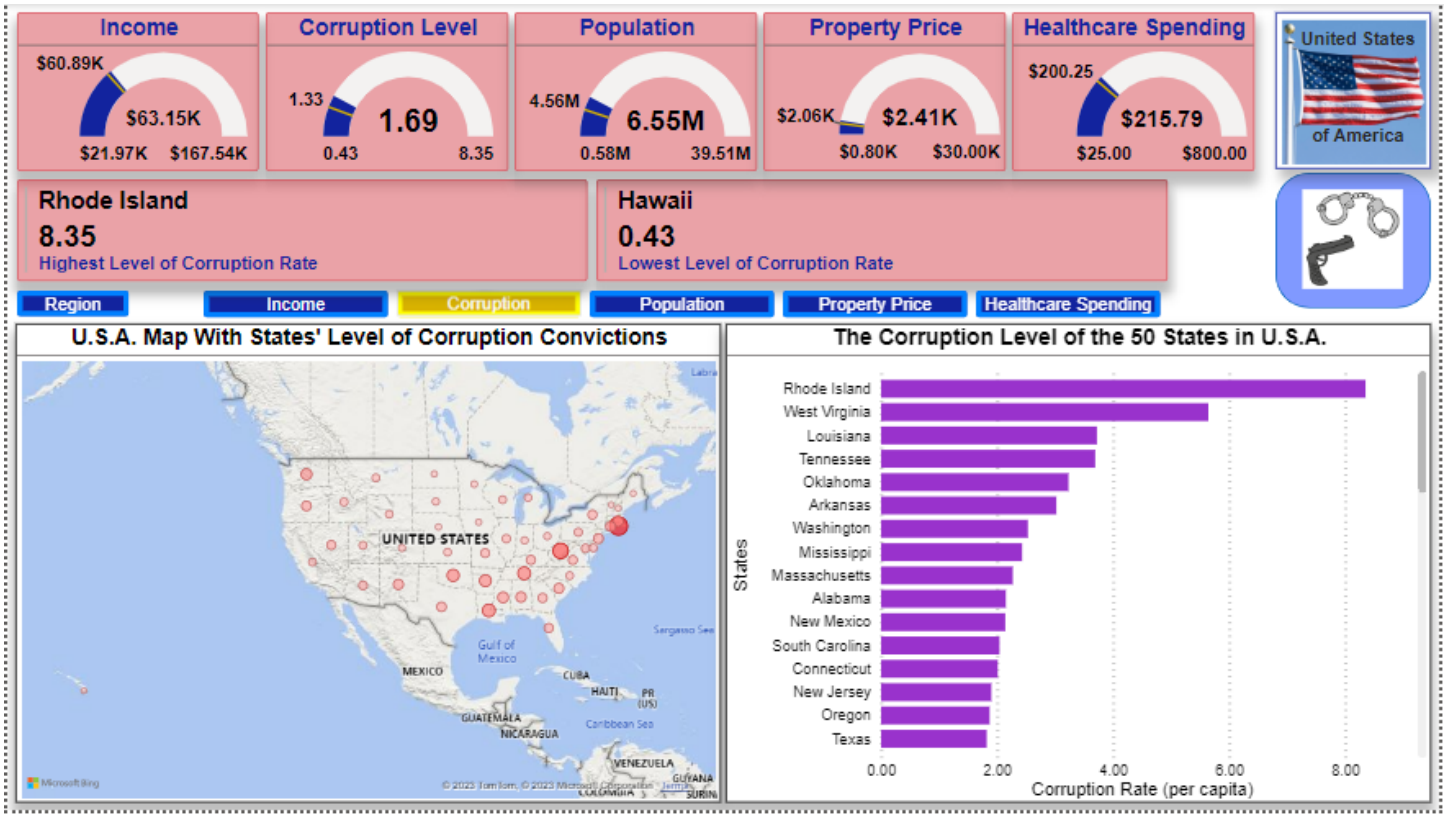
Power BI Dashboard 1.1



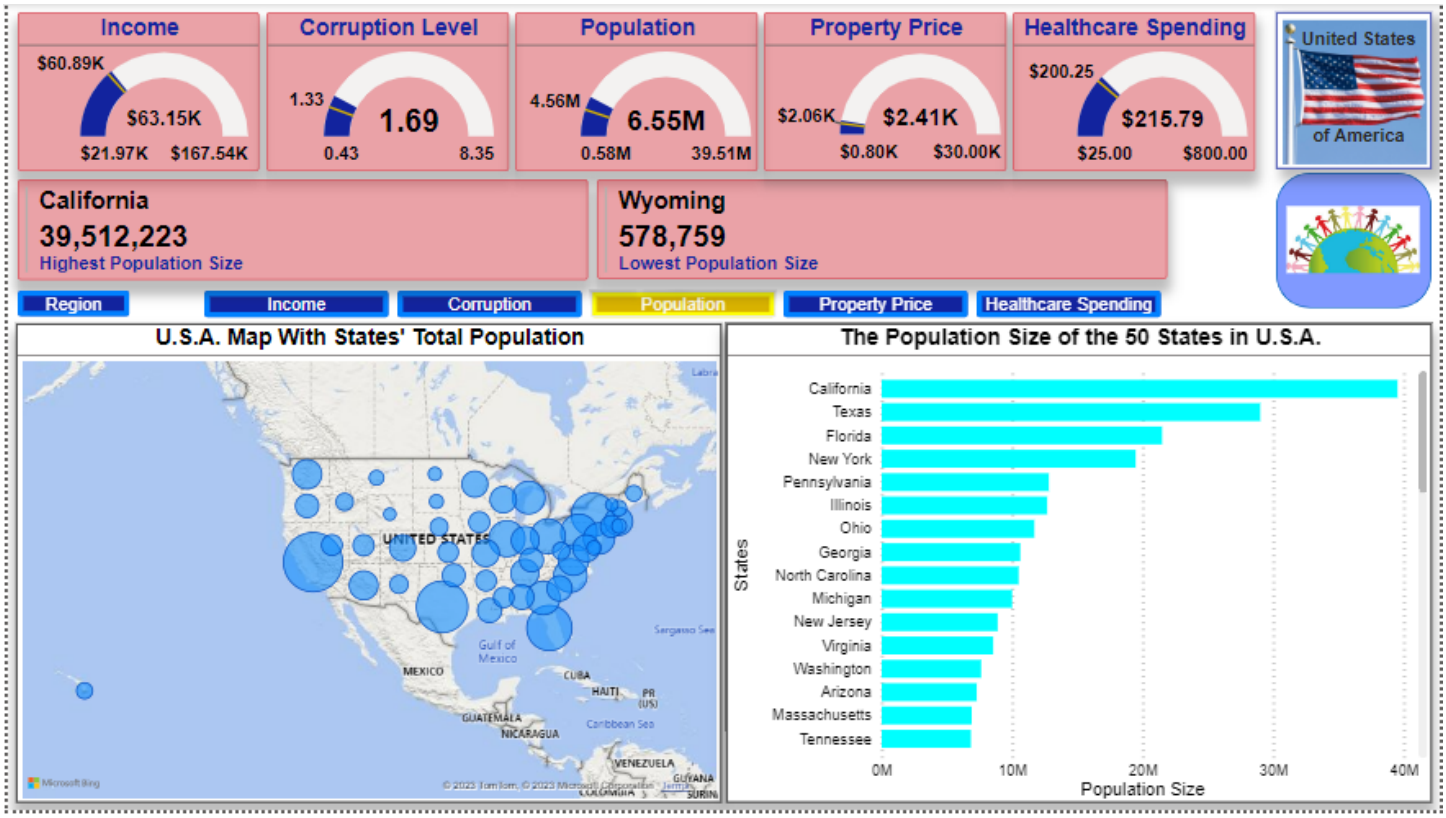
Power BI Dashboard 1.2



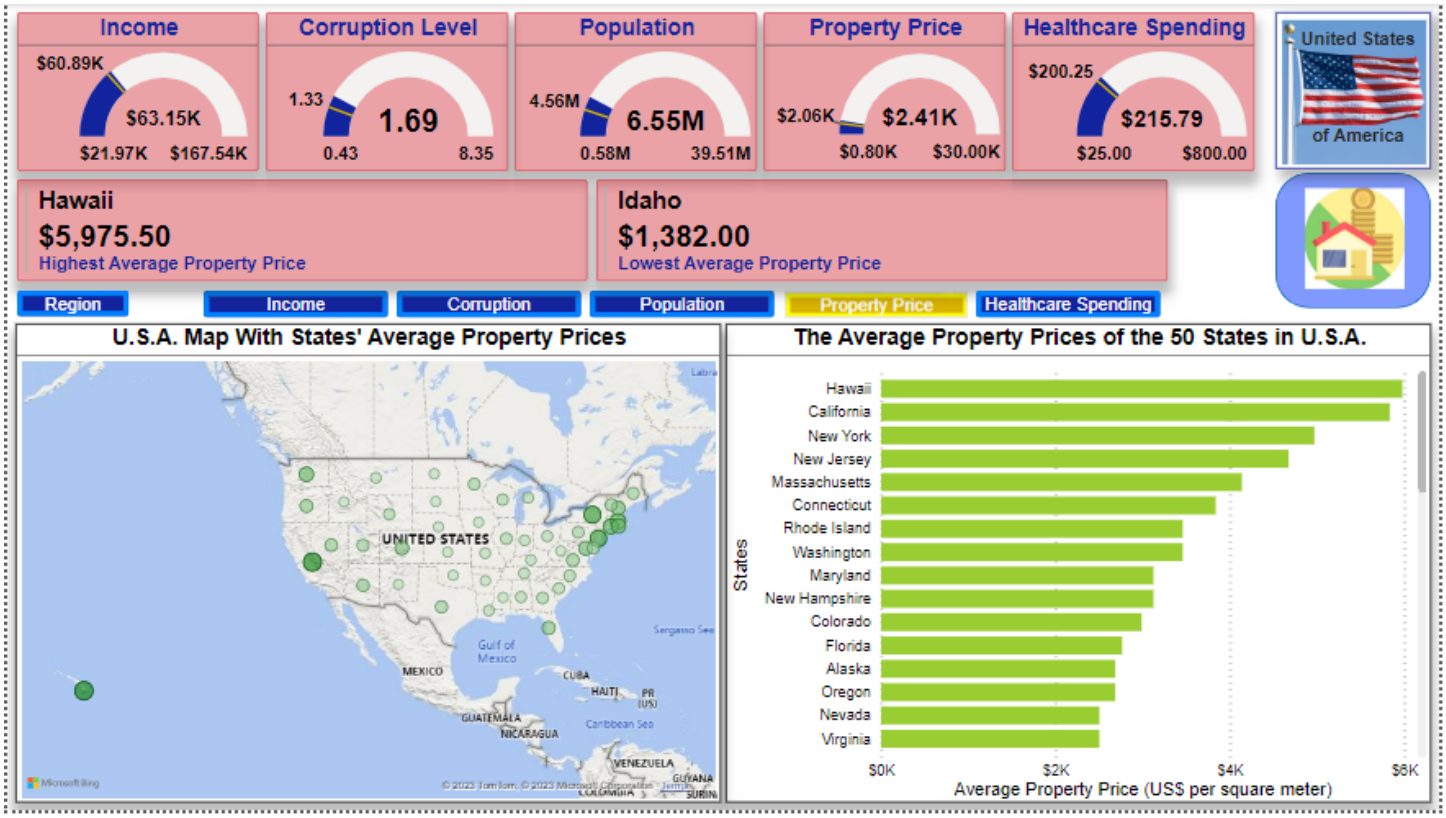
Power BI Dashboard 1.3



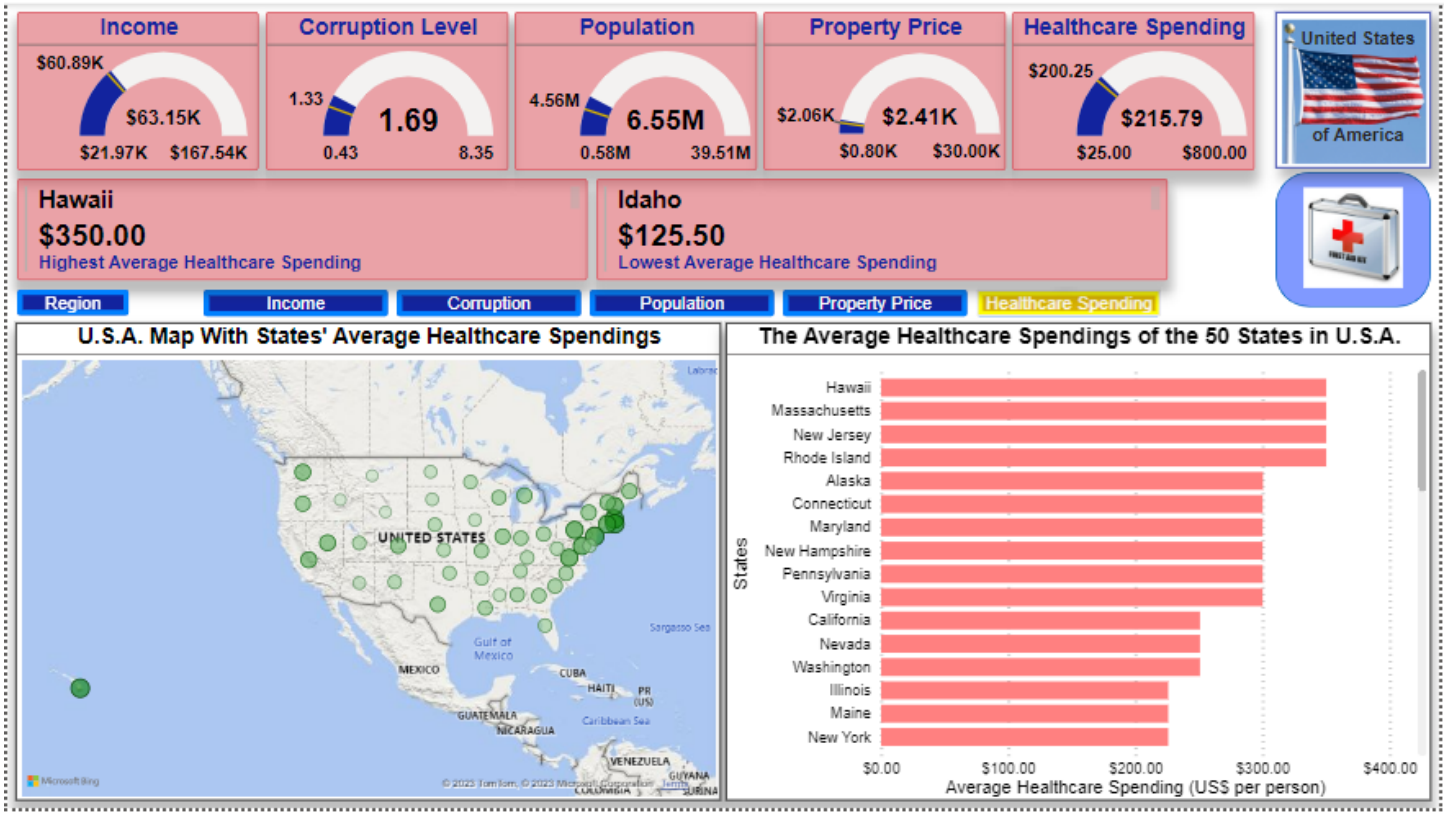
Power BI Dashboard 1.4



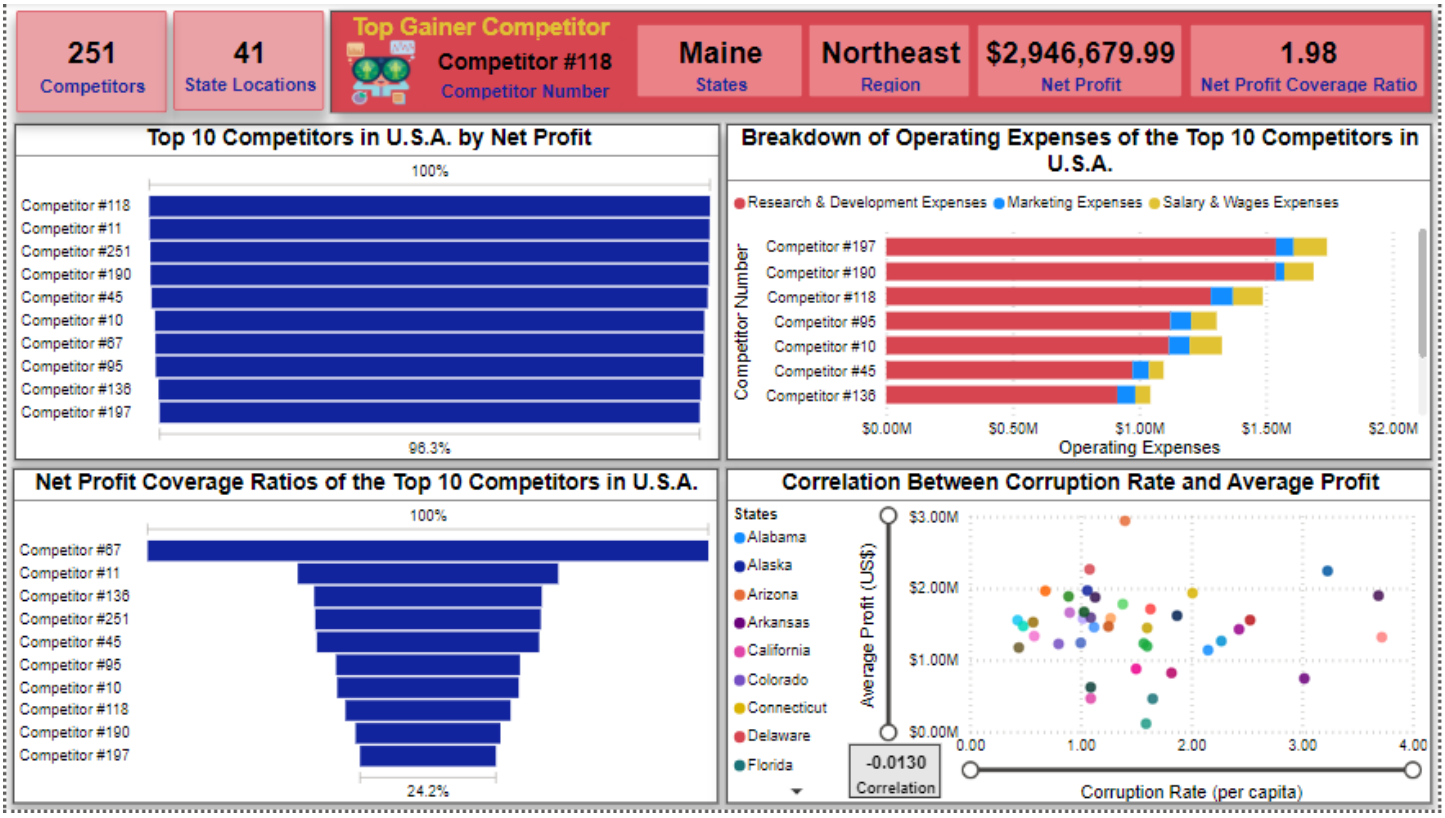
Power BI Dashboard 1.5



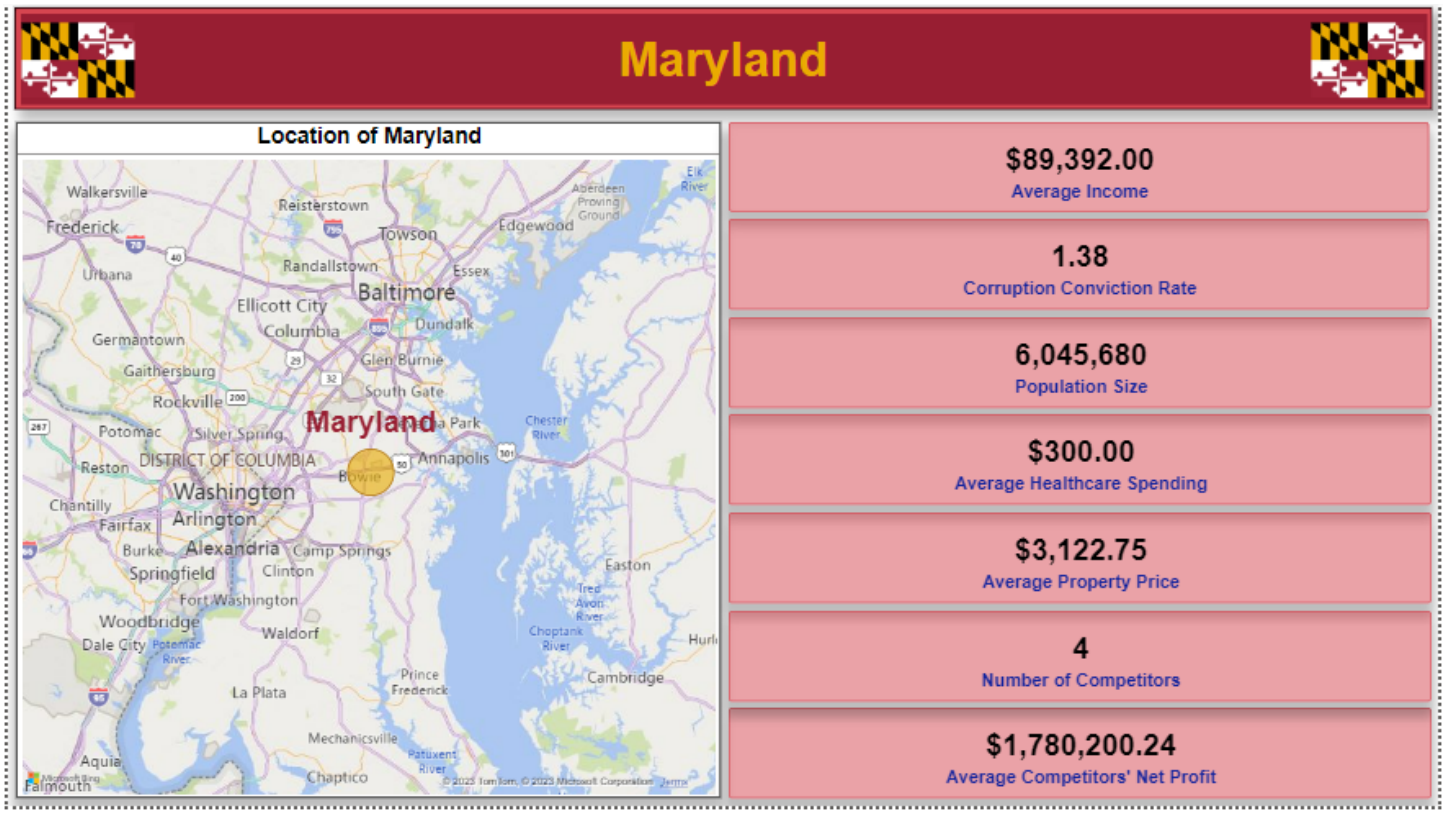
Power BI Dashboard 1.6



Power BI Dashboard 2

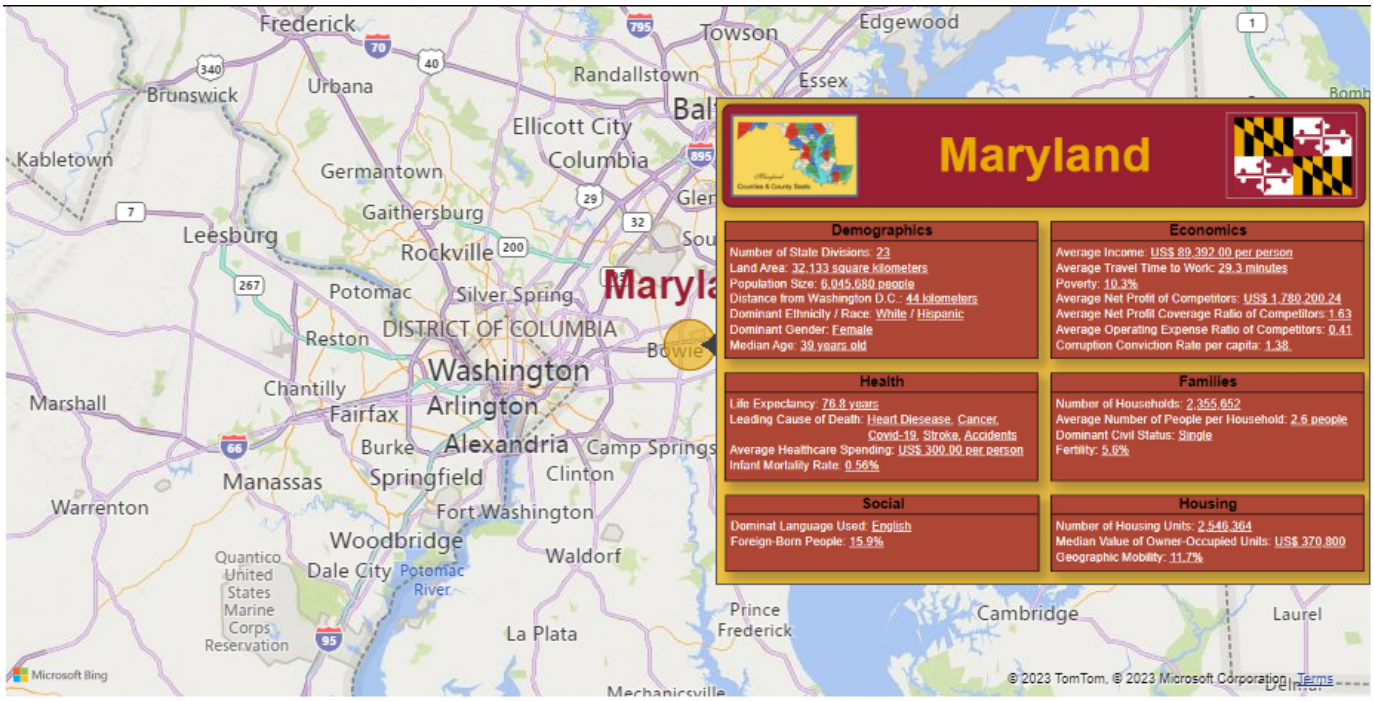


Power BI Dashboard 3



7. Created a custom tooltip for the recommended state with some informative demographic, economic, health, families, social, and housing information for the “Recommended State” page in the report view.

Power BI Chart 4



V. Data Saving Process in Power BI:

1. Saved the .pbix file named “Power BI Dashboards”.

Power BI Figure 17

Summary of Insights

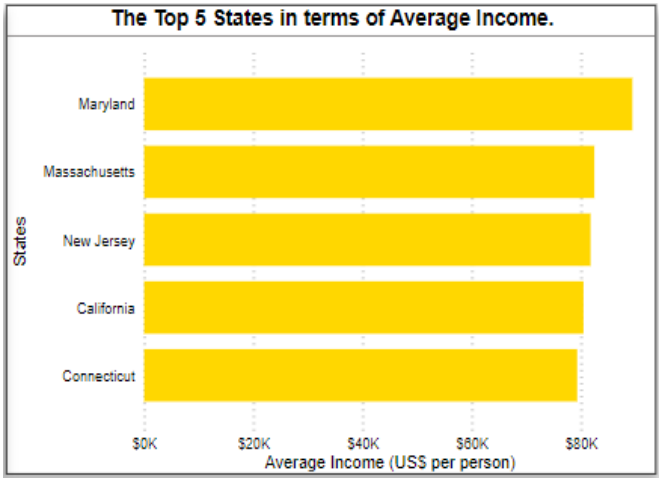
I. Answer the Information Questions:

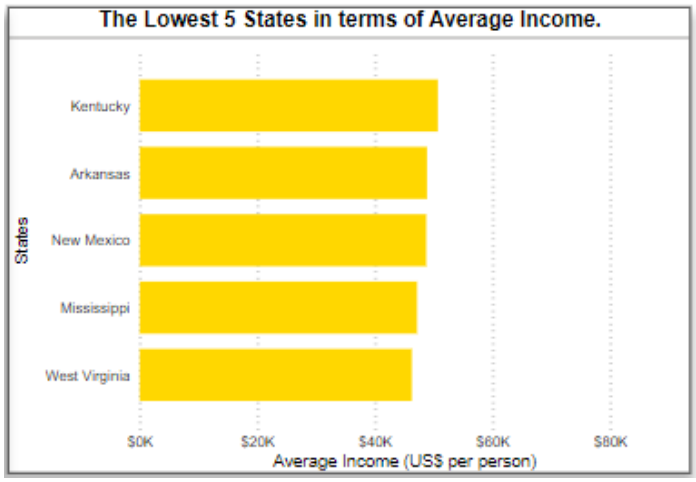
Based on the information generated from the data outputs, charts, and dashboards in PGAdmin, MS Excel, Google Colab, and Power BI, these are the findings:

1. Income.

a.) **The average income per person** of the states in U.S.A. ranges from US$ 46,254.00 to US$ 89,392.00.
Reference: Excel Chart 2, Python Output 15, Power BI Dashboard 1.2

b.) **The top 5 states with the highest average income** were Maryland, Massachusetts, New Jersey, California, and Connecticut. While the **lowest 5 states** were West Virginia, Mississippi, New Mexico, Arkansas, and Kentucky. Among the 50 states, Maryland had the **maximum average income per person** (US$ 89,392.00) and West Virginia had the **minimum** (US$ 46,254.00), about half of the former.



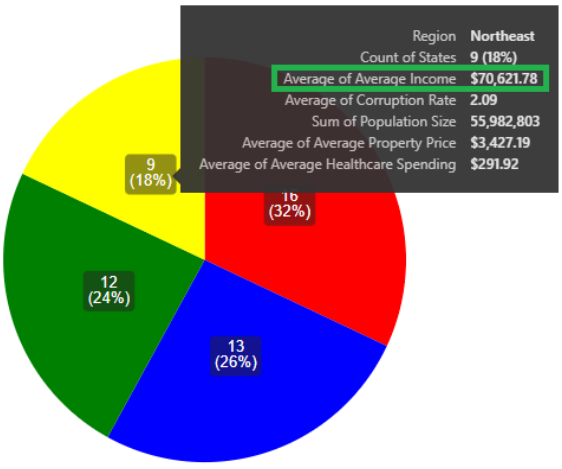


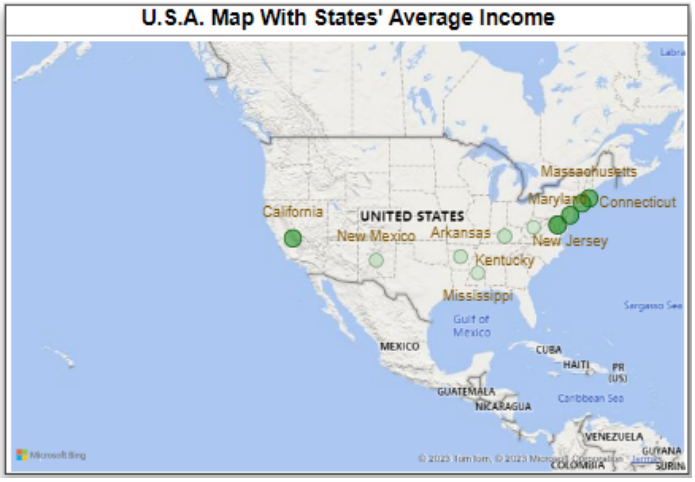
Additional Reference: SQL Output 22 & 23

c.) The **distribution of the average income** for all states were predominant within US$ 59,000 to 65,000 average income interval. The **mean of the average income** of the 50 states was US$ 63,146.08.
Reference: Python Chart 5, Python Output 15

d.) The **relative percentages of the top 5 states' average income per person** were 2.51% to 2.83% of the total average income per person for all the states in U.S.A.
Reference: SQL Output 21

e.) The northeastern part had the **highest average income by region** (US$ 70,621.78). It is indicated that majority of the top 5 states are located in the northeast region except Maryland and California but Maryland is located close to the Northeast, just below Pennsylvania.



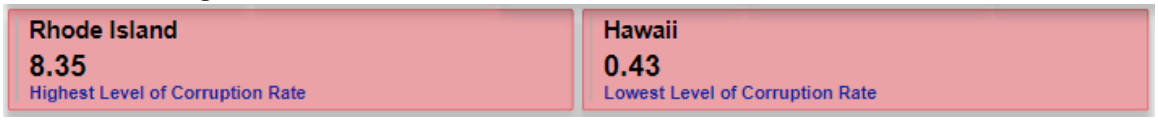


Additional Reference: Power BI Dashboards 1.1 & 1.2

2. Corruption Conviction Rate.

a.) The **level of corruption conviction per capita** of the states in U.S.A. ranges from 0.43 to 8.35.
Reference: Python Output 20, Power BI Dashboard 1.3

b.) Hawaii had the **least corruption conviction rating**, while Rhode Island on the contrary had the **highest corruption conviction rating** which was almost 20 times of Hawaii.



Additional Reference: SQL Outputs 24 & 25

c.) The **distribution of the level of corruption convictions** depicted highest frequencies in the range between 0.95 to 1.50 interval. The **average corruption convictions rate per capita** was approximately 1.69.
Reference: Python Chart 8, Python Output 20

d.) The **lowest minimum corruption conviction level by region** was in the western area where Hawaii is located, the **highest maximum corruption conviction level by region** was in the northeast area where Rhode Island sits. On average basis, southern region had the **highest rating** (2.35) and midwest region had the **lowest rating** (0.91).
Reference: Python Output 28, Python Chart 10, Power BI Dashboard Power BI Dashboard 1.1

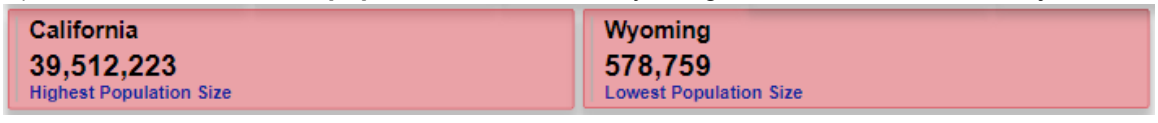
e.) It can be gleaned that Rhode Island and West Virginia had significantly much higher corruption rates compared to the other states as depicted by a **darker red circle**. Vermont, Maine, and New Hampshire were ranked 2nd, 3rd, and 4th lowest corruption rates respectively despite its location near Rhode Island. Hawaii, far from the other states can be seen with the **tiniest and lighter red circle** as compared to the other states.

Additional Reference: Power BI Dashboard 1.3

3. Population.

a.) The **population size** of the states in U.S.A. ranges from 578,759 to 39,512,223.
Reference: Python Output 30, Power BI Dashboard 1.4

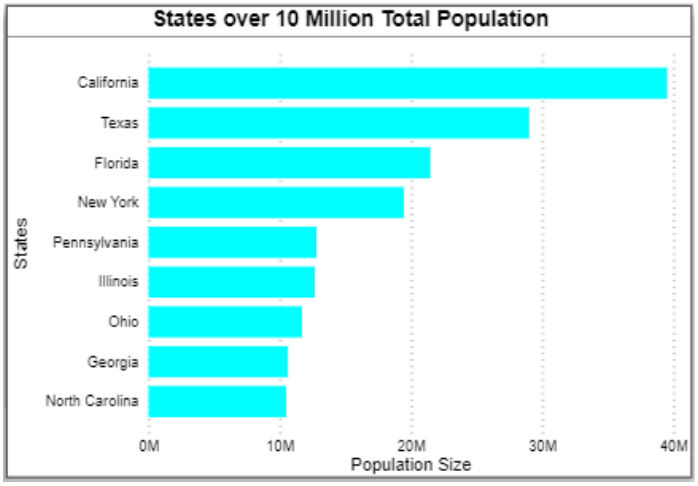
b.) California was the **most populated state**, while Wyoming had the **least inhabitants** just around 1.5% of California's.



Additional Reference: Power BI Dashboard 1.4

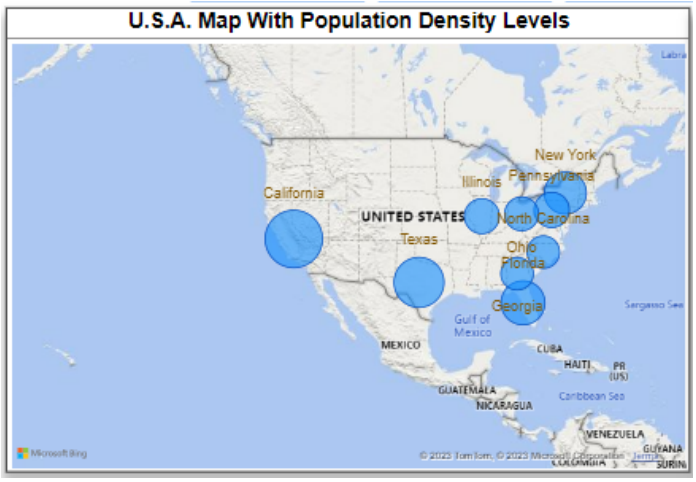
c.) The **distribution of the states' population** were prevalent in the 600,000 to 3,100,000 population range having a frequency of 22, which is almost half of the total number of states. The **average state population** was 6,550,675 people.
Reference: Python Chart 13, Python Output 30

d.) There were **9 states having a population over 10 million people**, these were California, Texas, Florida, New York, Pennsylvania, Illinois, Ohio, Georgia, and North Carolina. California and Texas (second highest state, 28,995,881) have a significant gap in terms of population size where California has more than 30% of the total population of Texas.



Additional Reference: SQL Output 28, Power BI Dashboard 1.4

e.) The map visualizes California, located in the western area having the highest population size as indicated with a **much larger circle** compared to the other states. It is also interesting to note that almost 60% of the total western region's population were from California. Surprisingly, southern region had the **highest combined state population** with a total of over 120 million people since 4 out of 9 states having over 10 million population size (Texas, Florida, Georgia, and North Carolina) were located in that region.



Additional Reference: Power BI Dashboard 1.4

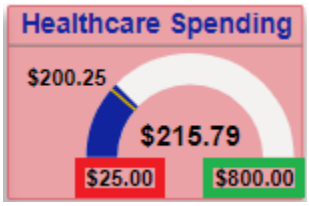
4. Healthcare Spendings.

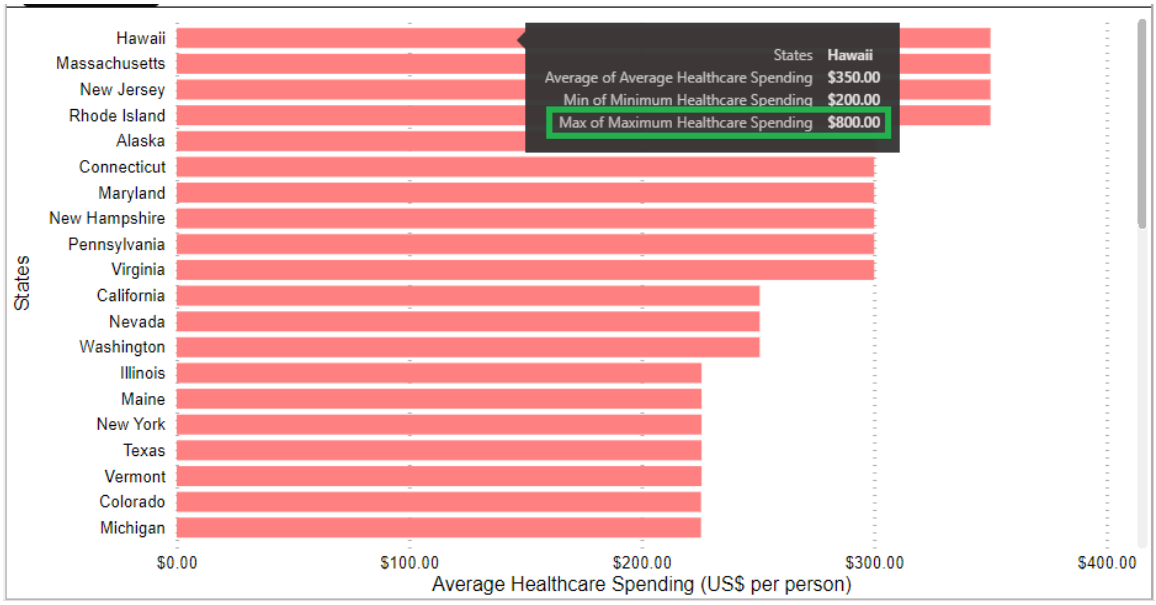
a.) The **average healthcare spendings per person** in U.S.A. were around US$ 125.50 to US$ 350.00.
Reference: Power BI Dashboard 1.6

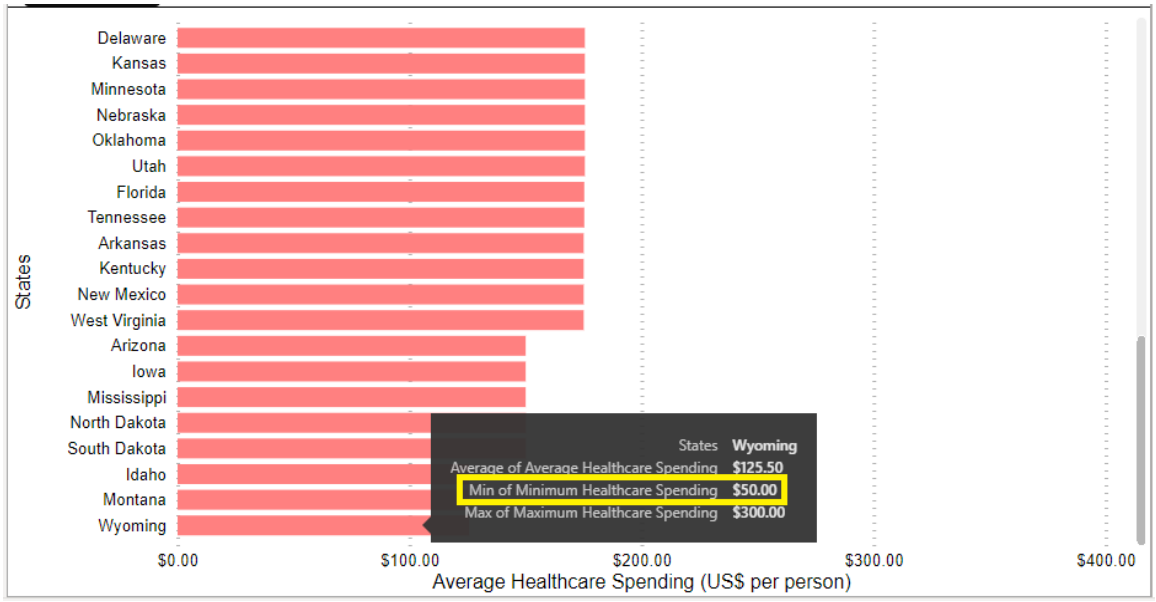
b.) The 4 states that **spent the most in healthcare spendings** were Hawaii, Massachusetts, New Jersey, and Rhode Island, while Wyoming, Montana, and Idaho **spent the least**, just about 36% of the former 4 states.

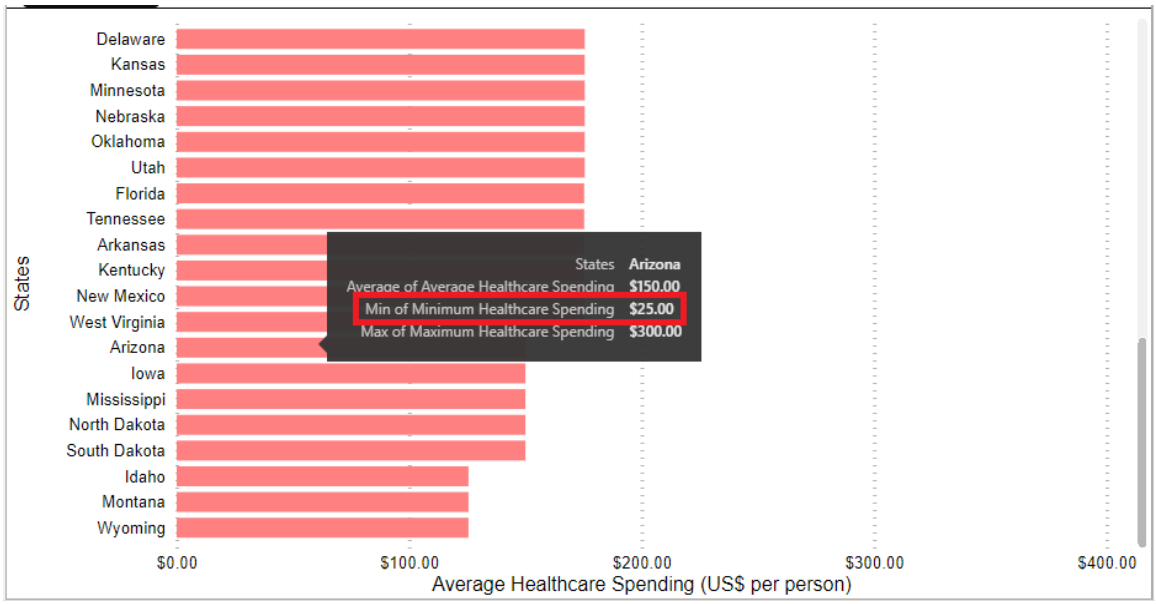
Additional Reference: Power BI Dashboard 1.6

c.) The states with the **highest maximum healthcare spendings** (US$ 800.00) were also those states with the highest average healthcare spendings but the **lowest minimum healthcare spendings** were Mississipi and Arizona (ranked 6th and 8th lowest average spendings) having a minimum spendings of US$ 25.00 which were even lower than of Wyoming, Montana, and Idaho's minimum (@ US$ 50.00 each).



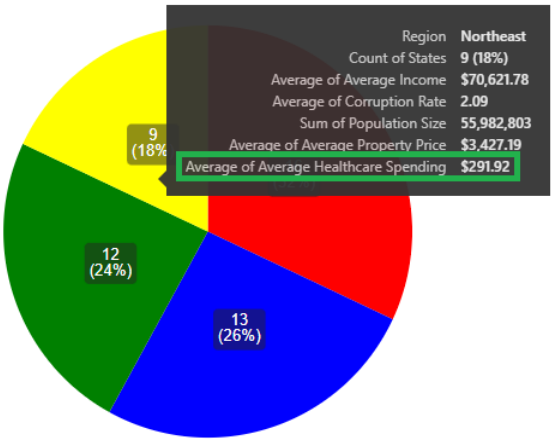


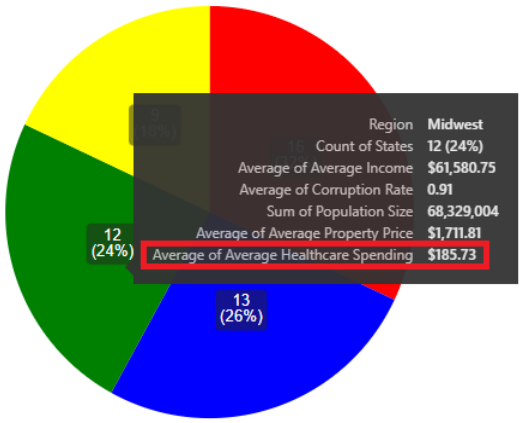


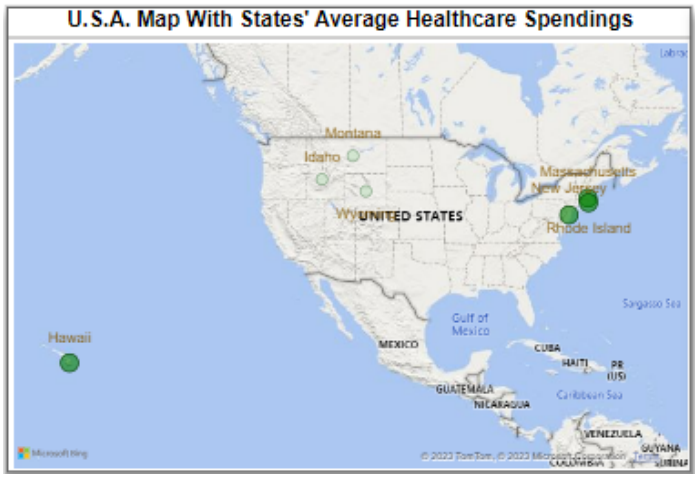


Additional Reference: Power BI Dashboard 1.6

d.) The map and pie chart reveal that <u>northeastern</u> region **spent the most on healthcare** (US$ 291.92) where Massachusetts, New Jersey, and Rhode Island are located. On the contrary, <u>midwest</u> region **spent the least on average** (US$ 185.73) but the 3 states with the lowest average spendings were located in the western area as well as Hawaii.





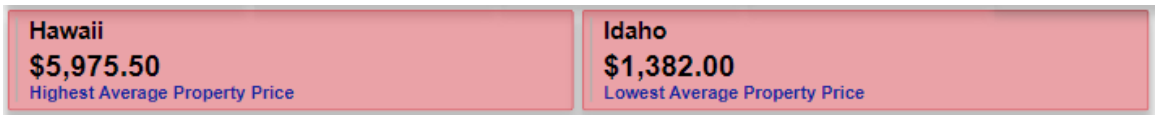


Additional Reference: Power BI Dashboard 1.1 & 1.6

5. Property Prices.

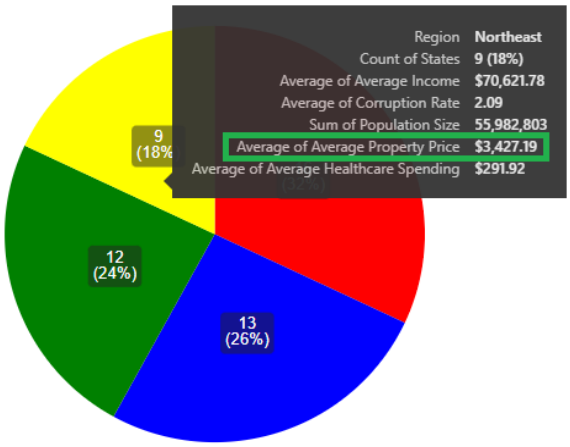
a.) The **average property price per square meter** in U.S.A. ranges from <u>US$ 1,382.00 to US$ 5,975.50</u>.
Reference: Power BI Dashboard 1.5

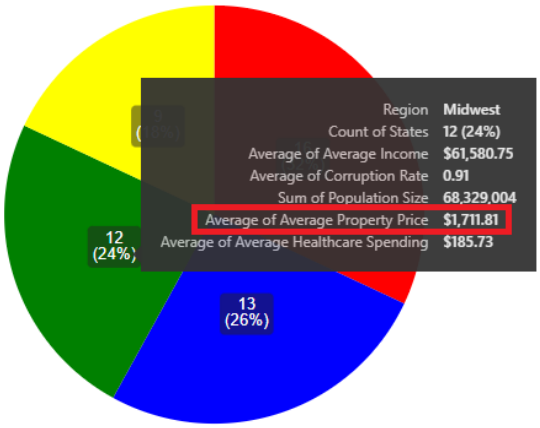
b.) Property prices in Hawaii are the **most expensive**, while Idaho has the **cheapest** just a little bit less than a quarter of Hawaii's.

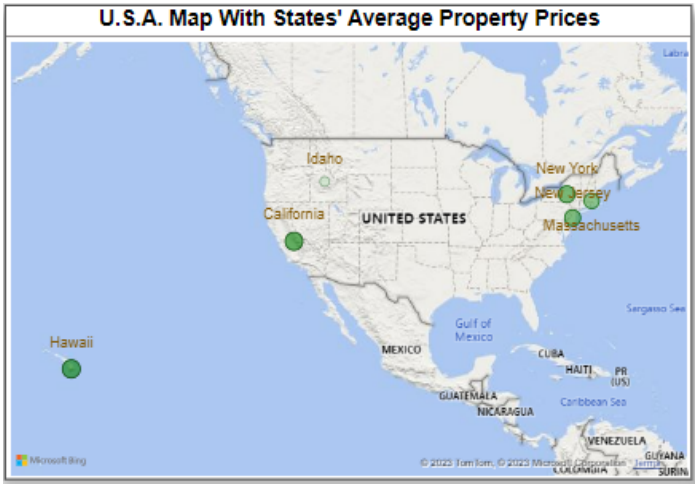


Additional Reference: Power BI Dashboard 1.5

c.) The pie chart illustrates the northeastern region had the **most expensive property price per square meter** on average (US$3,427.19). Based on the map visual, the top 2 states, Hawaii and California are located in the western region, but the 3rd to 5th highest states (New York, New Jersey, and Massachusetts) are located in the northeastern region. Properties located in the midwest region were the **cheapest** on average (US$1,711.81) but the state with the lowest average property price was in the west region.







Additional Reference: Power BI Dashboard 1.1 & 1.5

6. Competitors.

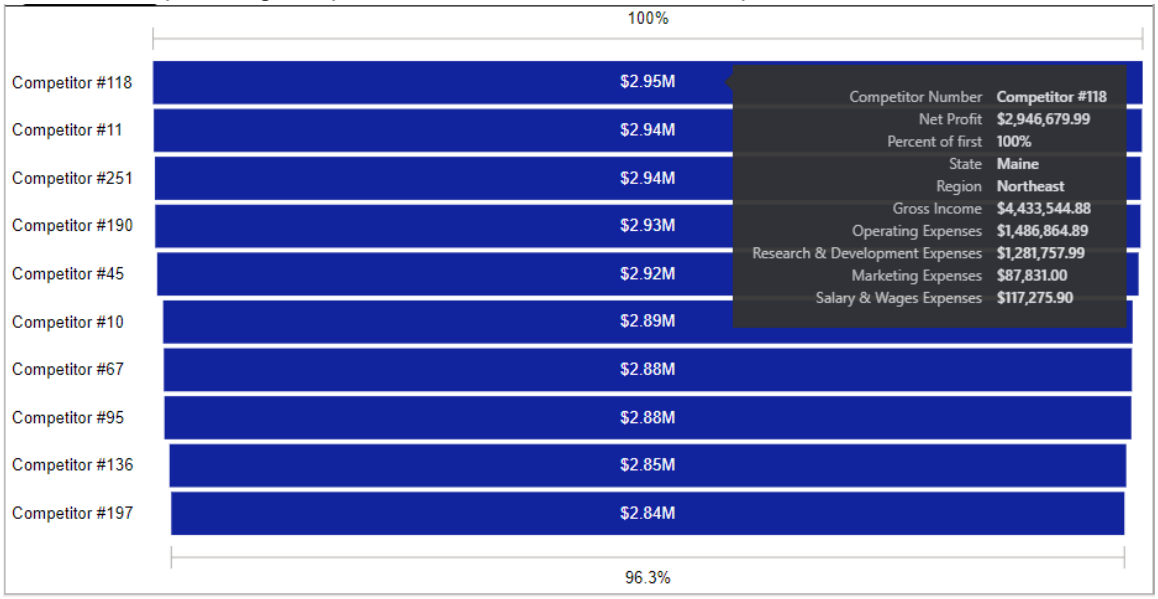
a.) Based on the data gathered, competitors were located in 41 states in U.S.A. Florida, California, and New York have the **highest number of competitor companies**, 22, 21, and 17 respectively. Arizona has the least number of competitors among the 41, but based on the average profit of the competitors by state in U.S.A., it achieved the **highest average profit** (US$ 2,942,282.99). New York on the other hand had the **lowest average profit by state** (US$ 113,756.45).

	state_usa character varying	avg numeric
1	Arizona	2942282.990000000000
2	Delaware	2265218.990000000000
3	Oklahoma	2242746.132857142857
4	Alaska	1969126.590000000000
5	Minnesota	1964784.740000000000

	state_usa character varying	avg numeric
36	Texas	821438.990000000000
37	Arkansas	743958.990000000000
38	Wisconsin	620797.490000000000
39	California	464722.616190476190
40	Florida	461162.922272727273
41	New York	113756.446470588235

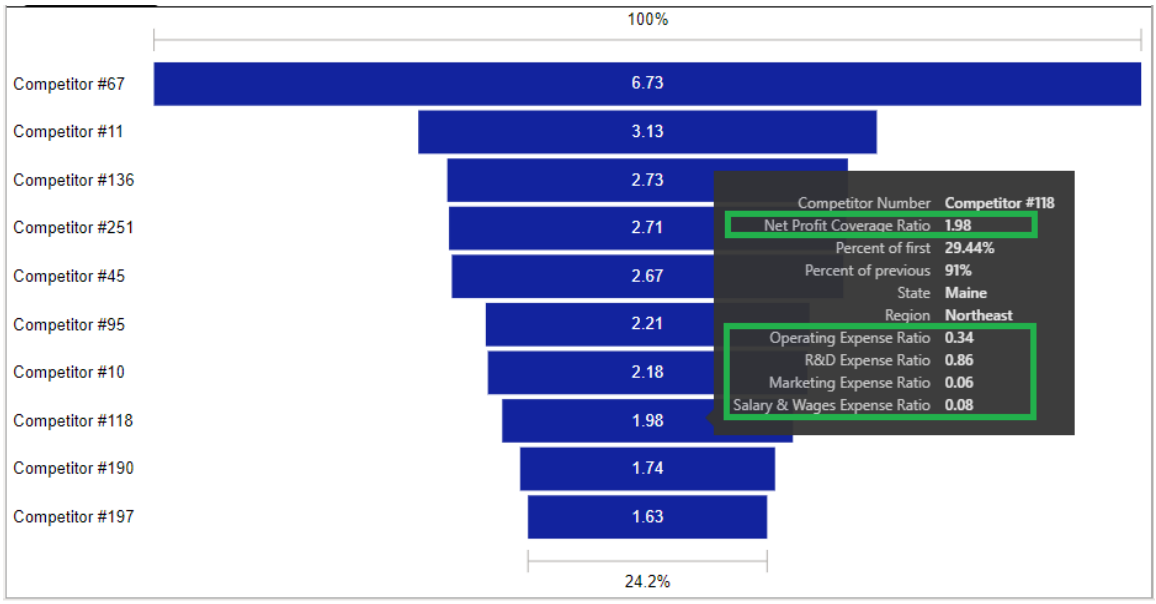
Additional Reference: Python Output 37, Python Chart 15, SQL Output 26

b.) Among the 251 competitors in U.S.A., competitor # 118 from Maine in the northeastern region was the **top leading competitor** in the health smartwatches industry in U.S.A. with a net profit amount of US$ 2,946,679.99 and a net profit coverage ratio of 1.98. The **net profit of the Top 10 Competitor's** were around US$ 2,838,170.99 to US$ 2,946,679.99 where the total amount of net profit earned by the rank 10th competitor, Competitor # 197 from Oklahoma was somewhat close to the top leading competitor's, about 96% of its total net profit.



Additional Reference: Python Output 46, Python Chart 19, Power BI Dashboard 2

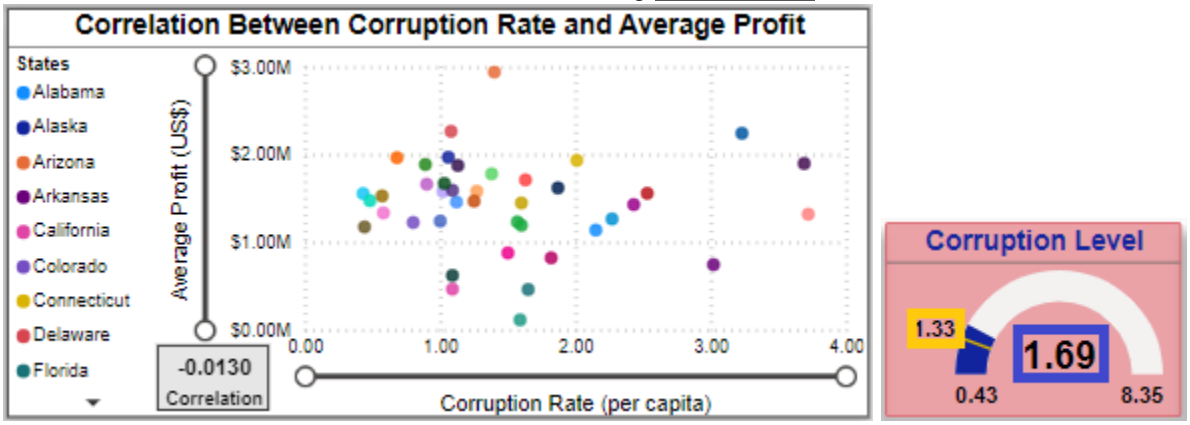
c.) The doughnut charts in Python Chart 20 present the **financial metrics breakdown** as a whole that the total operating expenses of all the competitors were over 45% of the total gross income. The operating expenses comprise of research & development expenses, marketing expenses, and salary & wages expenses, its majority composition came from research and development expenses, which cost over 80% of the total operating expenses. The **competitor with the highest net profit coverage ratio** was Competitor # 212 from Pennsylvania with a net profit coverage ratio of 13.04 meaning that their operating expenses cover less than 10% of their gross income however they're just ranked 11th in terms of total net profit but it's still kicking to be in the top 10. **Among the top 10 competitors** by net profit, Competitor # 67 from Florida, located in the southern region had the highest net profit coverage ratio, 6.73. It is also interesting to note that all of the top 10 competitors have a healthy **financial viability status**. Their net profit coverage ratios (NPCR) were all over 1.5 (the higher it is, the better) and their operating expense ratios (OER) were all below 0.60 (the lower it is, the better).



Additional Reference: Python Charts 20 & 21, Python Output 49, Power BI Dashboard 2

d.) **The highest total competitors' profit by state locations** was in Wyoming with US$ 16,710,539.00 (combined profit of the 10 competitors). However, it had the **lowest population size of all the states** in U.S.A.
Reference: SQL Output 31

e.) **The impact of the level of corruption rate to the average profit of competitors by state** doesn't show any signs of connection as evidenced by the scatterplot. States with competitors with low corruption conviction levels such as Hawaii, Vermont, and Maine have an average profit between US$ 1,174,427.99 and US$ 1,555,451.66 doesn't make any difference with those of high corruption conviction levels like of Arkansas, Oklahoma, Tennessee, and Louisiana with an average profit ranging from US$ 743,958.99 to US$ 2,242,746.13. Arizona and New York's corruption conviction level were between the median and mean levels but they have contrasting average profits, US$ 2,942,282.99 and US$ 113,756.45 respectively. And lastly, the **correlation coefficient level of the corruption rate and states' average profits** was -0.013 which fell between -0.30 to +0.30 indicating no correlation between the 2 variable measures.



Additional Reference: Power BI Dashboard 2

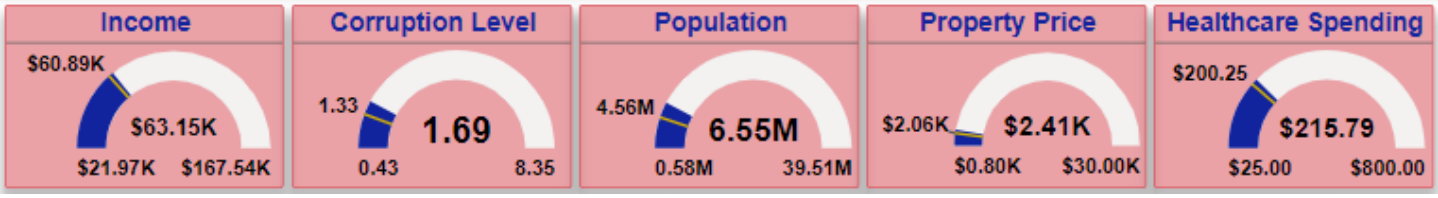
7. Metrics Correlation.

The heatmap in Python Chart 2 summarizes the **correlation between the combination of the 5 key metrics used as a basis of criteria for selecting a state for business expansion**. The 3 combinations: average income, average property price, and average healthcare spendings were highly-positively correlated to each other (r = 0.70 to 0.74). Population and property price have a low-positive correlation (r = 0.42). The other 6 combinations were not correlated to each other (-0.23 to 0.23). It emphasizes that the higher the average income per person in a state, the more spendings on healthcare per person the state have on average as well as higher average property price per square meter and vice versa. **Average income per person and corruption convictions per capita** had negligible correlation, -0.23 since it's within the range of -0.30 to +0.30. The scatterplot in Python Chart 1 also illustrates that the connection of the values for both metrics doesn't show any upward or downward trend. By the way, West Virginia can be seen as the second highest level of corruption conviction and had the lowest average income per person among all states. The **level of corruption convictions and**

healthcare spendings in different states don't seem to correlate, their correlation coefficient was 0.17, but it is interesting to mention that people in Hawaii had the one of the highest average spendings on healthcare and it has the lowest corruption rating. **Property prices and state population** were slightly positively correlated, around 0.42 correlation value. Both California and New York were top 5 in both metrics. While, South Dakota was included in the bottom 5 for both. **Income and property price** had the highest correlation value among all pairs, 0.74, California, New Jersey, and Massachusetts were included in the top 5 for both metrics, West Virginia on the other hand had the lowest average income and second to the lowest in terms of average property price per square meter. **Population size and corruption rate** don't correlate, their correlation coefficient value was -0.074. Though, it can be gleaned that Vermont was ranked second to the lowest for both metrics and North Dakota was found to be in the lowest 5 states in terms of corruption rate and population size. There was nearly 0 correlation between **property prices and corruption conviction levels**. It is worth mentioning that West Virginia which had the 2nd highest corruption rate and one of the 2nd lowest states in terms of property prices. Hawaii, in contrast had the lowest rating on corruption convictions and had the highest average property price per square meter. Then, Illinois was in the median class for both categories. **Population size** doesn't have a significant effect on **both average income and average healthcare spendings**, 0.23 and 0.14 correlation values respectively. California is included in the top 5 in terms of both population size and average income. Alaska was ranked 3rd to the lowest population size and top 5 average healthcare spending. **Average income and average healthcare spendings** have significantly high positive correlation, 0.70, Both Massachusetts and New Jersey were in top 5 for both metrics. Maryland and Connecticut, both in top 5 average income were one of the 6th in average healthcare spendings which were just a quarter dollar per square meter behind Alaska. On the other hand, Mississippi was included in the bottom 5 for both. **Property Price and healthcare spendings** have a high positive correlation, 0.71. Hawaii, Massachusetts, and New Jersey were one of the highest states in both property prices and spendings on healthcare. Idaho had the lowest average property price as well as being one of the 3 states with the lowest average healthcare spendings.
Reference: Python Charts 1-2, Python Output 13, Excel Figures 1, 6-8, Excel Charts 1, 3-4

8. Data Outliers and Skewness.

The **dispersion of the data** for average income, corruption rate, population, average healthcare spending, average property price, and competitors' profit were all high as indicated by the various measures of dispersions such as range, standard deviations, etc. but the average income was to a lesser extent among the others. Regarding **outliers**, there were significantly 4 huge values found on corruption rates, average healthcare spending, population, and average property prices using tukey's fence method for setting the upper and lower threshold, while there were none for the average income and competitors' profit. The data for the 5 key metrics show signs of **low to high positive skewness** (0.5329 to 2.9456) except for the competitors' profit with a **symmetric skewness** (0.2163) since it was within the range of -0.50 to +0.50. It is also interesting to note that all of the aforementioned variables' mean values were much higher than their median values proving that the deviations of the higher half of the data from the median values were more than of those of the lower half of the data.



Additional Reference:Python Charts 3-8, 11-14, 16-18, Python Output 36

9. Prove the average profit of the competitors is greater than the rounded median profit.

There is enough evidence to prove that **the average profit of the competitors US$ 1,252,204.38 is greater than the median profit**. The t-test statistics result shows a value of approximately 1.70 which is greater than the 1.65 critical value and the 0.0447 p-value is lower than the 0.05 significance level indicating that the average of the competitors' profit is significantly more than US$ 1,150,000.00.
Reference: Python Output 51

10. Prove that California's population size exceeds 12% of the total U.S.A. state population.

There is enough evidence to prove that **California's population size 39,512,223 exceeds 12% of total U.S.A. population**. The z-test statistics result shows a value of 35.31 which is way much higher than the 1.64 critical value and the 1.68×10^{-273} p-value is teeny-weeny that clearly stipulates a very strong evidence that California's population is significantly more than 12% of 327,533,774.
Reference: Python Output 52

II. Answer the Main Objective of the Project:

Pros and Cons Comparisons of Nominated States for Business Expansion:

Best State of Choice	Income	Corruption Rate	Population	Property Price	Healthcare Spending	Competitor
Maryland	✓ ✓	✓	●	✓	✓	✓
Massachusetts	✓ ✓	✗	●	✓	✓ ✓	✓
New Jersey	✓ ✓	✗	✓	✓	✓ ✓	✗
California	✓ ✓	✓ ✓	✓ ✓	✓	✓	✗ ✗
Hawaii	✓	✓ ✓	✗	✗	✓ ✓	✗ ✗
Washington	✓	✗	✓	✓	✓	●
New York	✓	●	✓ ✓	✗	●	✗ ✗
Vermont	✗	✓ ✓	✗ ✗	✗	●	✓
Maine	✗	✓ ✓	✗	✗	●	✗ ✗
Virginia	✓	●	✓	✓	✓	✓
Colorado	✓	✓	●	●	●	✓
New Hampshire	✓	✓	✗	✗	✓	✗
Alaska	✓	✓	✗ ✗	✗	✓ ✓	✗

Maryland has the highest average income among all states indicating more disposable income, low corruption conviction rate, and potential market for health-related products. High property prices attract a decent number of the populace having high average income earnings. Market demand and validation is already established in the state indicating there is potential for substantial growth. Competition and market saturation don't seem to pose much of a threat in the state.

Massachusetts has a very high average income and healthcare spending but a relatively high corruption conviction rate. Strong economy and potential demand for health-related products for a moderate population size. Market demand and validation is established and competition is not that flooded.

New Jersey has a very high average income and healthcare spending but a relatively high corruption conviction rate. High property prices attract a significantly large population with a high income. Consider conducting a research on the needs and wants of the target market in the vicinity.

California has a very high average income with a very low corruption conviction rating. Spend quite a lot on healthcare. High property prices can sustain a vast number of rich people. Strong competition, high number of competitors may likely lead to market saturation, and price wars.

Hawaii has high average income and the lowest corruption conviction rate of all states, and has a potential demand in healthcare spending. But, hefty property prices for a relatively small population and unique island market should be considered since it could impact distribution of logistics and marketing strategies. Competitors have earned profits, indicating demand but it is mainly influenced by tourism.

Washington has a high average income with decent healthcare spendings but a high corruption conviction rate. Tech-savvy population and potential demand for innovative products, but competition can be tough.

New York has a high average income with mediocre corruption conviction rate and healthcare spending. Property prices are too high despite a large state population. Presence of competitors suggests demand, but be aware of market saturation and conflicts in prices. This requires a good strategy in building strong brand loyalty and product differentiation in order to stand out in the market.

Vermont has a very low corruption conviction rate but has a low average state income, and a very small population size, and local market dynamics for a moderate property price. Has decent healthcare spendings. Market demand and validation is established and not many competitors are present.

Maine has a very low corruption conviction rate but not much disposable income and with the relatively small populace and moderately low healthcare spending, it doesn't have much market potential compared to the others. Market is established but the presence of the top competitor can be a problem since it is more likely to dominate the market.

Virginia has a decent average income, corruption conviction rate, population size, property price, and healthcare spending. Potential market, but consider competition and local preferences.

Colorado has a decent average income and low corruption conviction rate. Decent demand for health-related products, but assess competition.

New Hampshire has a high average income, low corruption conviction rate, and high healthcare spending. Consider the smaller population size and local market dynamics. Market demand and validation are not yet established, conduct further analysis on the market needs and wants first.

Alaska has a high average income and healthcare spending with a low corruption conviction rate. Competitors have earned profits, suggesting demand but one of the top 10 competitors can be a problem as well as unique market challenges due to geography and smaller population.

Recommendations

Based on the highlighted pros and cons summary, Maryland appears to be the most suitable state for the expansion of the health tracker smartwatch business. Here's a more detailed expansion of the reasons why Maryland stands out:

1. High Average Income: Maryland has the highest average income among all the listed states. This indicates that residents of Maryland have higher disposable income, making them more likely to afford premium health-related products like smartwatches. A higher average income also suggests a potential customer base that values health and wellness.

2. Low Corruption Conviction Rate: A low corruption conviction rate is a positive indicator of a stable business environment. Low corruption rates typically correlate with a well-regulated market, which can benefit your expansion plans. This means you'll likely face fewer bureaucratic hurdles and legal challenges.

3. Moderate Population: While Maryland's population is not as massive as some other states, it still provides ample opportunities for your business expansion.

4. Potential Market for Health-related Products: The combination of high average income and a low corruption conviction rate implies that Maryland's residents prioritize their health and well-being. This presents an opportunity for your health tracker smartwatches, as consumers with higher disposable incomes are more likely to invest in wellness and fitness products.

5. High Property Prices: The presence of high property prices can be seen as a positive aspect, especially in the context of your business. High property prices often indicate a wealthier population and a higher standard of living. This aligns well with the demographic that might be interested in purchasing premium health tracker smartwatches.

6. Market Demand and Validation: The presence of competitors who have earned profits in Maryland suggests that there is an existing demand for health-related products, including smartwatches. This is a strong indicator that the market is receptive to such products, reducing the risk of entering a completely untested market.

7. Potential for Growth: While competition does exist, the fact that market demand and validation are already established indicates that there is room for substantial growth. The presence of competitors can also be seen as a positive sign, as it demonstrates that consumers are already familiar with and interested in the product category.

8. Limited Threat from Market Saturation: The provided data doesn't indicate high levels of market saturation, which suggests that there is still potential for your business to capture a significant market share without facing extreme competition.

9. Business-Friendly Environment: Maryland's relatively low corruption conviction rate and its proximity to major cities on the East Coast, such as Washington D.C., create a business-friendly environment. This can facilitate logistics, partnerships, and overall business operations.

In conclusion, Maryland's combination of high average income, low corruption conviction rate, moderate population size, potential market for health related products, high property prices, existing market demand, potential for growth, and limited competitors make it a strong candidate for expanding the health tracker smartwatch business. However, it's essentially recommended to conduct further research into the local market dynamics, consumer preferences, competitive benchmarking, and regulatory factors to ensure a successful and sustainable expansion strategy.

Scope and Limitations

The scope of the study centered around assisting the health tracker smartwatch business in determining the ideal state for expanding its operations. The analysis encountered challenges primarily due to the data constraints. These constraints included the lack of specific time periods, competitor company information, detailed sales and purchase data, comprehensive lists and breakdowns of states' income, property prices, healthcare expenditures, and considerations related to other crime categories. These limitations may have impacted the precision of our findings.