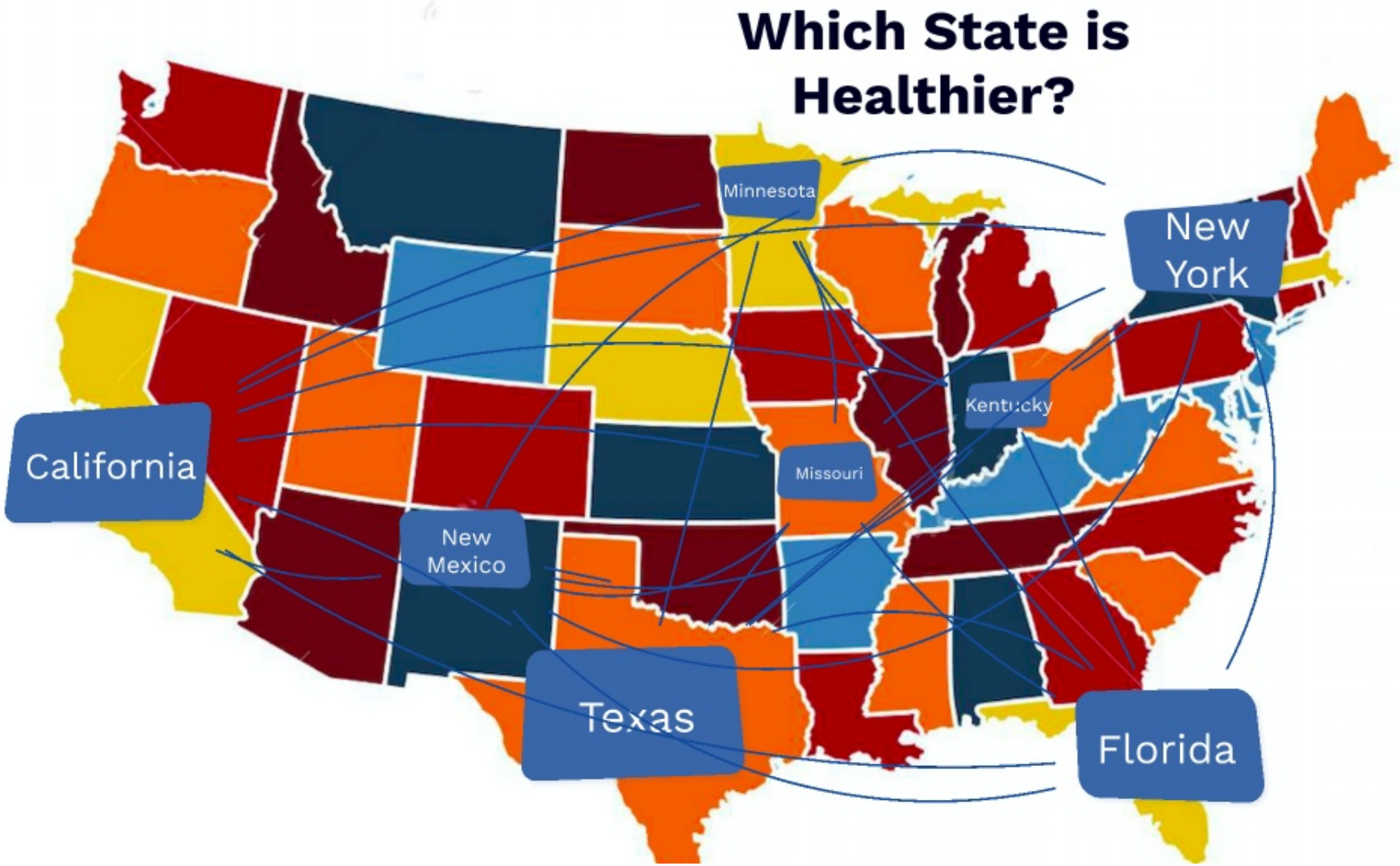
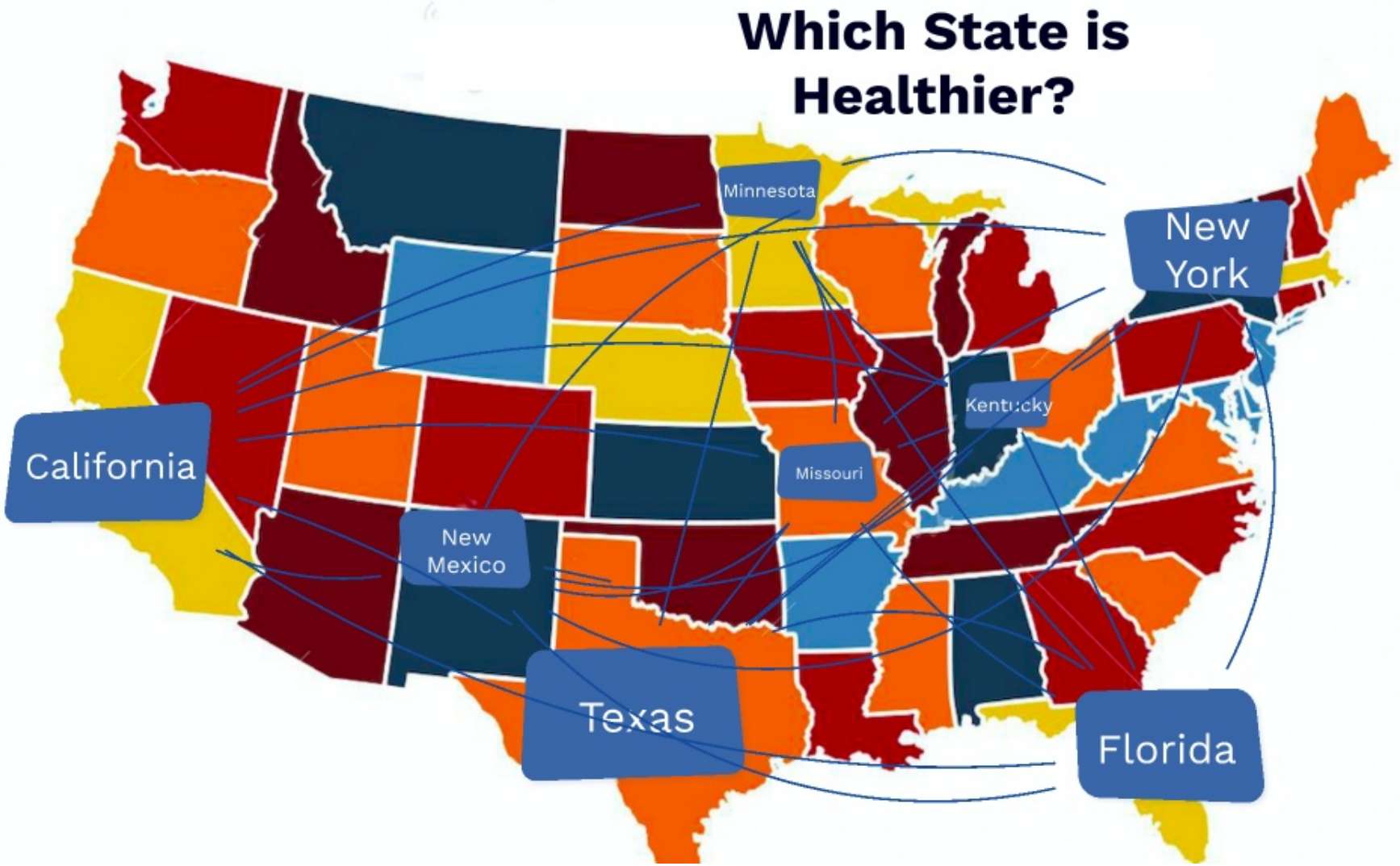

Which State is Healthier?
Minnesota
New York
Kentucky
California
Missouri
New Mexico
Texas
Florida

Chronic Diseases in the United States

- Six in ten Adults in the US have at least one chronic disease, and four in ten adults have more than two diseases
- 60% of the worldwide death rate is due to chronic disease

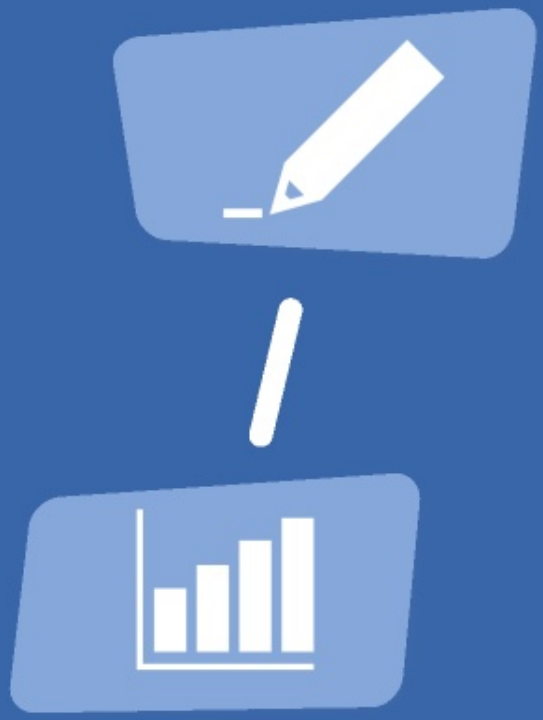
Study the possible relationship between GDP and Chronic Diseases


Which State is Healthier?
Minnesota
New York
Kentucky
California
Missouri
New Mexico
Texas
Florida

Data Cleaning

Prepare the data for chart making

- Using a list comprehension filtered out all of the rows not matching the criteria given in the conditional. So this made it easy to keep only the information we need.
- The challenge with cleaning the data was the use of a regular expression. Since the data had extra parentheses in the latitude and longitude coordinates. So it became necessary to filter these characters out using a simple regular expression, which are more convenient to be used during the G-map visualizing part.



States GDP Data Cleaning

Step 1

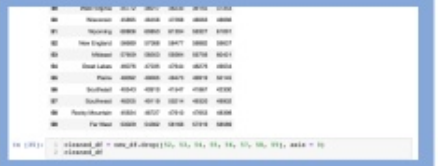
- Import the dirty CSV
- Remove the extra column by using the DEL function

Step 2

- Use DROPNA function to remove the incomplete data, and rename the column

Step 3

- Use DROP and .ILOC functions to remove unwanted rows



In [2]:
```
1 import pandas as pd
```

In [3]:
```
1 csv_path = ("GDP_Data/bea-gdp-by-state.csv")
2
3 csv_df = pd.read_csv(csv_path)
4 csv_df.head()
```

Out[3]:

	Fips	Area	2013	2014	2015	2016	2017
0	0	United States	48534	49329	50301	50660	51337
1	1000	Alabama	36674	36473	36818	37158	37508
2	2000	Alaska	69711	67179	65971	63304	63610
3	4000	Arizona	38352	38534	38787	38940	39583
4	5000	Arkansas	35888	36265	36295	36502	36714

In [4]:
```
1 del csv_df["Fips"]
2 csv_df.head()
```

States GDP Data Cleaning

Step 1

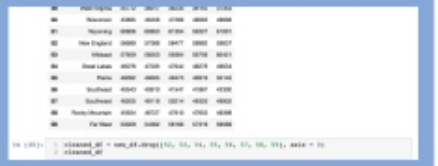
- Import the dirty CSV
- Remove the extra column by using the DEL function

Step 2

- Use DROPNA function to remove the incomplete data, and rename the column

Step 3

- Use DROP and .ILOC functions to remove unwanted rows



In [5]:
```
csv_df = csv_df.dropna(how="any")
```

In [6]:
```
renamed_df = csv_df.rename(columns={
    "Area": "State"
})
```

In [7]:
```
renamed_df.head()
```

Out[7]:

	State	2013	2014	2015	2016	2017
0	United States	48534	49329	50301	50660	51337
1	Alabama	36674	36473	36818	37158	37508
2	Alaska	69711	67179	65971	63304	63610
3	Arizona	38352	38534	38787	38940	39583
4	Arkansas	35888	36265	36295	36502	36714

States GDP Data Cleaning

Step 1

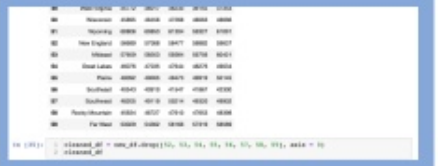
- Import the dirty CSV
- Remove the extra column by using the DEL function

Step 2

- Use DROPNA function to remove the incomplete data, and rename the column

Step 3

- Use DROP and .ILOC functions to remove unwanted rows



In [8]:
```
1 renamed_df.dtypes
```

Out[8]:
```
State    object
2013      int64
2014      int64
2015      int64
2016      int64
2017      int64
dtype: object
```

In [17]:
```
1 new_df = renamed_df.drop([0], axis = 0)
2 new_df.head()
```

Out[17]:

	State	2013	2014	2015	2016	2017
1	Alabama	36674	36473	36818	37158	37508
2	Alaska	69711	67179	65971	63304	63610
3	Arizona	38352	38534	38787	38940	39583
4	Arkansas	35888	36265	36295	36502	36714
5	California	53838	55571	57637	58974	60359

49	West Virginia	35772	36017	36233	36155	37353
50	Wisconsin	45895	46456	47268	48063	48666
51	Wyoming	60806	60853	61304	59327	61091
52	New England	56689	57068	58477	58882	59637
53	Mideast	57659	58353	59364	59756	60421
54	Great Lakes	46076	47035	47644	48276	49034
55	Plains	48092	49005	49473	49919	50145
56	Southeast	40543	40910	41547	41867	42300
57	Southwest	48205	49119	50214	49320	49902
58	Rocky Mountain	45634	46727	47610	47653	48398
59	Far West	53029	54362	56166	57319	58589

In [35]:
```
1 cleaned_df = new_df.drop([52, 53, 54, 55, 56, 57, 58, 59], axis = 0)
2 cleaned_df
```

States GDP Data Cleaning

Step 1

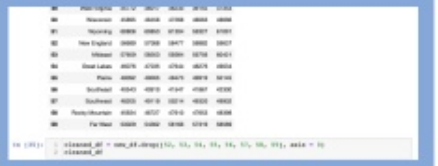
- Import the dirty CSV
- Remove the extra column by using the DEL function

Step 2

- Use DROPNA function to remove the incomplete data, and rename the column

Step 3

- Use DROP and .ILOC functions to remove unwanted rows



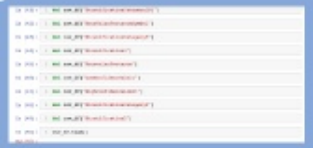
Chronical Diseases Data Cleaning

Step 2

- importing the CSV into a data frame
- Debugging the low memory error by using the LOW_MEMORY =FAlSE Boolean

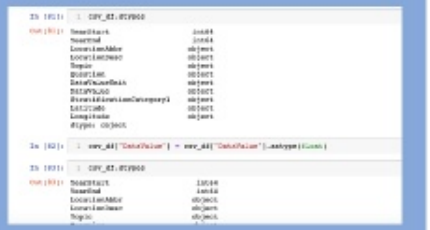
Step 2

- deleted unwanted columns using the DEL function
- followed that up with a DROPNA function that cleaned up the empty data and corresponding rows



Step 3

- then used the ASTYPE function to convert data in certain columns to floats



In [31]:
```
import pandas as pd
```

In [32]:
```
csv_path = ("../U.S._Chronic_Disease_Indicators__CDI_.csv")

csv_df = pd.read_csv(csv_path)
low_memory=False
csv_df
```

```
/Users/Pariah/anaconda3/envs/PythonData3/lib/python3.7/site-packages/IPython/core/interactive
shell.py:3049: DtypeWarning: Columns (7,10,18,19,20,21,23,30,31,32,33) have mixed types. Spec
ify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Out[32]:

	YearStart	YearEnd	LocationAbbr	LocationDesc	DataSource	Topic	Question	Response	DataValueUnit	DataVa
0	2016	2016	US	United States	BRFSS	Alcohol	Binge drinking prevalence among adults aged >=...	NaN	%	Pr
1	2016	2016	AL	Alabama	BRFSS	Alcohol	Binge drinking prevalence among	NaN	%	Pr

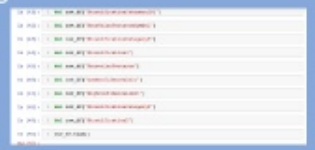
Chronical Diseases Data Cleaning

Step 2

- importing the CSV into a data frame
- Debugging the low memory error by using the LOW_MEMORY =FAlSE Boolean

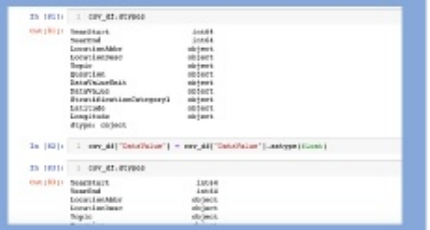
Step 2

- deleted unwanted columns using the DEL function
- followed that up with a DROPNA function that cleaned up the empty data and corresponding rows



Step 3

- then used the ASTYPE function to convert data in certain columns to floats



```
In [41]: 1 del csv_df["StratificationCategoryID1"]

In [42]: 1 del csv_df["DataValueFootnoteSymbol"]

In [43]: 1 del csv_df["StratificationCategory3"]

In [44]: 1 del csv_df["Stratification3"]

In [45]: 1 del csv_df["DatavalueFootnote"]

In [46]: 1 del csv_df["LowConfidenceLimit"]

In [47]: 1 del csv_df["HighConfidenceLimit"]

In [48]: 1 del csv_df["StratificationCategory2"]

In [49]: 1 del csv_df["Stratification2"]

In [50]: 1 csv_df.head()

Out[50]:
```

```
In [50]: 1 csv_df.head()
```

Out[50]:

						aged >=...				

```
In [60]: 1 csv_df = csv_df.dropna(how="any")
```

```
In [61]: 1 csv_df.head()
```

Out[61]:

	YearStart	YearEnd	LocationAbbr	LocationDesc	Topic	Question	DataValueUnit	DataValue	DataValueAlt	StratificationCa
1	2016	2016	AL	Alabama	Alcohol	Binge drinking prevalence among adults aged >=...	%	13	13.0	
2	2016	2016	AK	Alaska	Alcohol	Binge drinking prevalence among adults aged >=...	%	18.2	18.2	

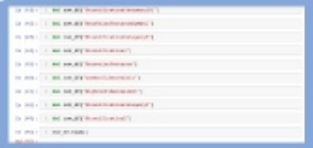
Chronical Diseases Data Cleaning

Step 2

- importing the CSV into a data frame
- Debugging the low memory error by using the LOW_MEMORY =FAlSE Boolean

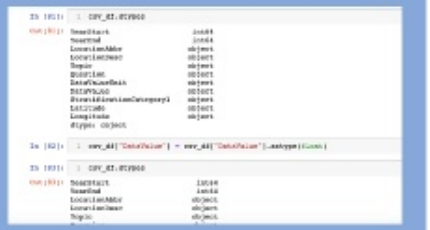
Step 2

- deleted unwanted columns using the DEL function
- followed that up with a DROPNA function that cleaned up the empty data and corresponding rows



Step 3

- then used the ASTYPE function to convert data in certain columns to floats



```
In [81]:  1  csv_df.dtypes
```

```
Out[81]: YearStart                  int64
         YearEnd                    int64
         LocationAbbr              object
         LocationDesc              object
         Topic                     object
         Question                  object
         DataValueUnit             object
         DataValue                 object
         StratificationCategory1   object
         Latitude                  object
         Longitude                 object
         dtype: object
```

```
In [82]:  1  csv_df["DataValue"] = csv_df["DataValue"].astype(float)
```

```
In [83]:  1  csv_df.dtypes
```

```
Out[83]: YearStart                  int64
         YearEnd                    int64
         LocationAbbr              object
         LocationDesc              object
         Topic                     object
```

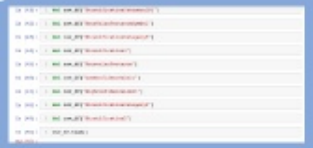
Chronical Diseases Data Cleaning

Step 2

- importing the CSV into a data frame
- Debugging the low memory error by using the LOW_MEMORY =FAlSE Boolean

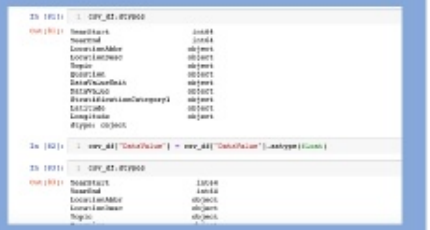
Step 2

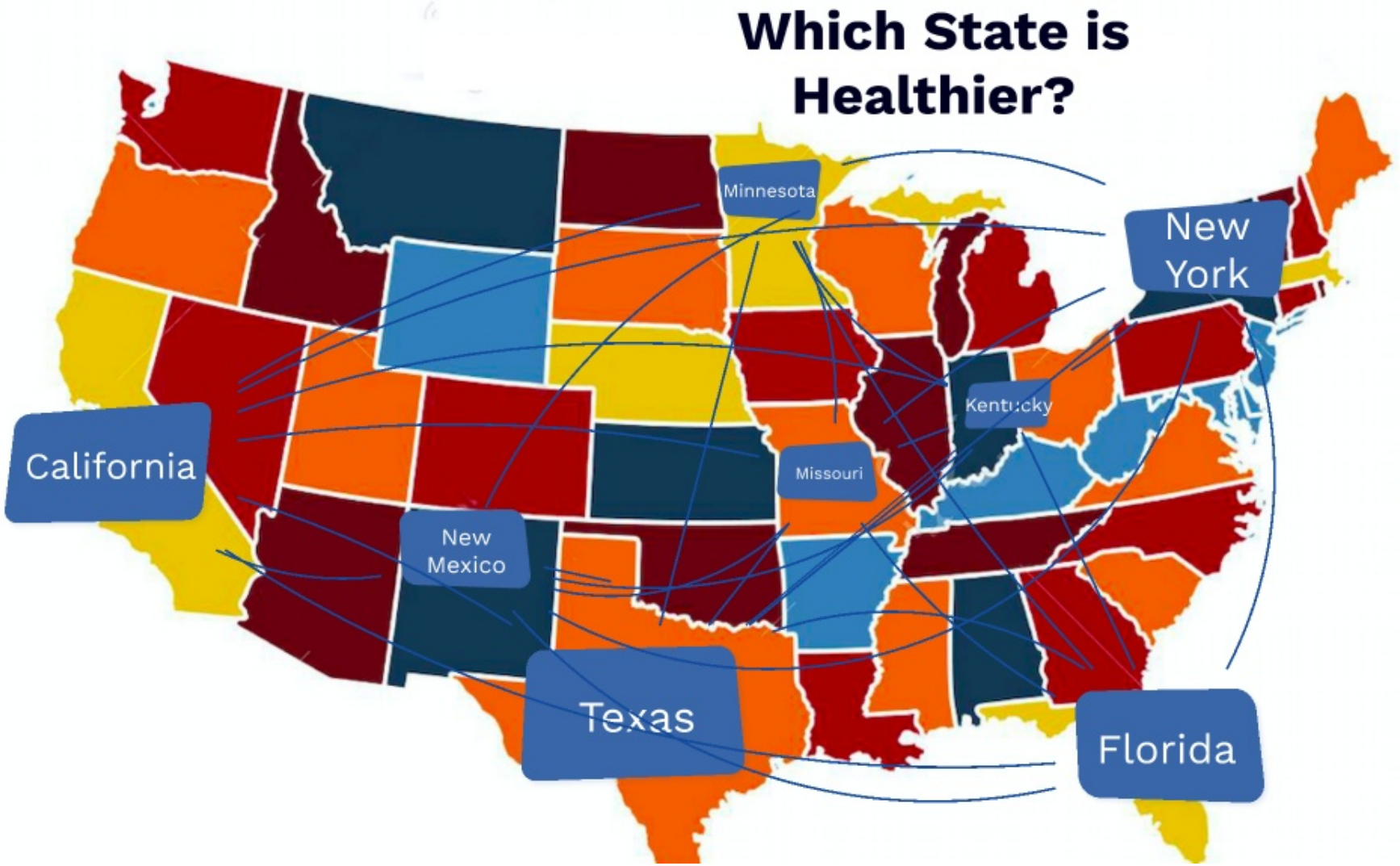
- deleted unwanted columns using the DEL function
- followed that up with a DROPNA function that cleaned up the empty data and corresponding rows



Step 3

- then used the ASTYPE function to convert data in certain columns to floats

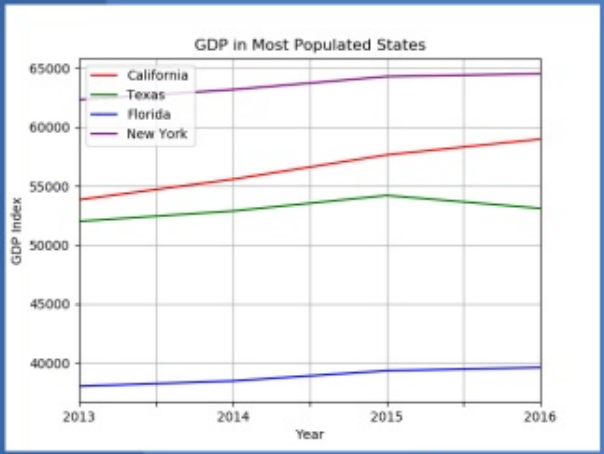


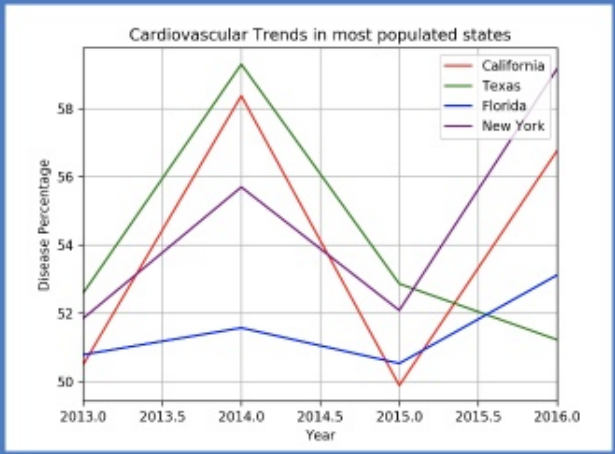

Which State is Healthier?
Minnesota
New York
Kentucky
California
Missouri
New Mexico
Texas
Florida

Visualization

Make the data easier to interpret

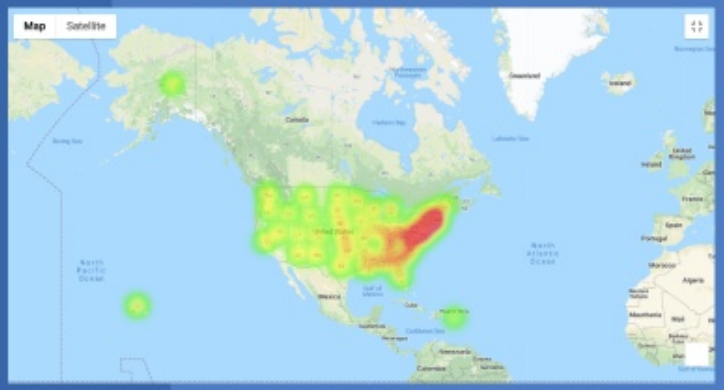
Line Chart for comparing the Trend between Cardiovascular and GDP

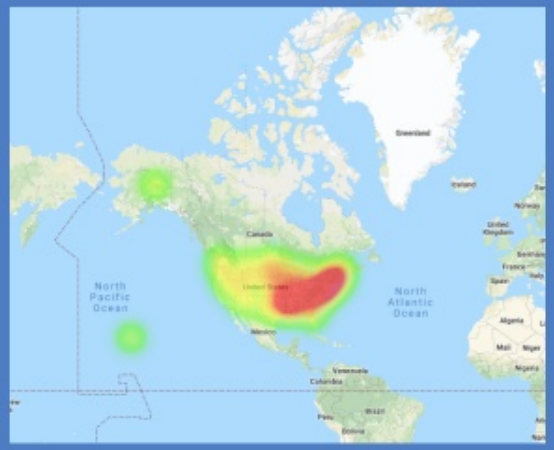




- we have chosen Cardiovascular Disease to further our analyzation. Here we have a trend chart of Cardiovascular Disease with the top 4 most populated states which are California, Texas, Florida and New York.
- I did the line chart by first by using the 'get group' function which has helped me group the dataframe according to 'Cardiovascular Diseases'. From there, I have managed to the loc function for each of the 4 states. Each state has been converted into a data frame.
- From this chart, it can be analyzed that Texas has the highest rate of Cardiovasculardisease and Florida is the lowest

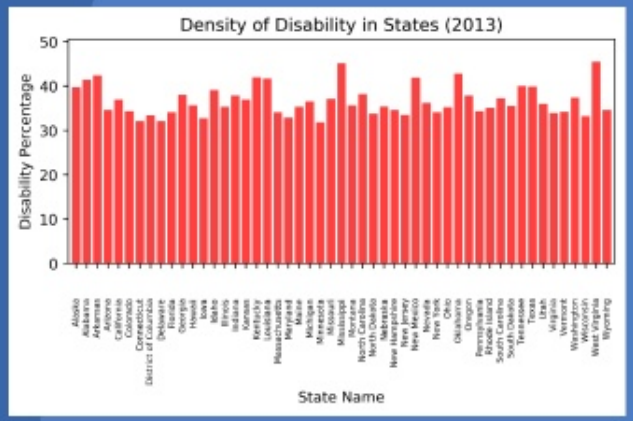
G-Map for Chronic Diseases and GDP Distribution

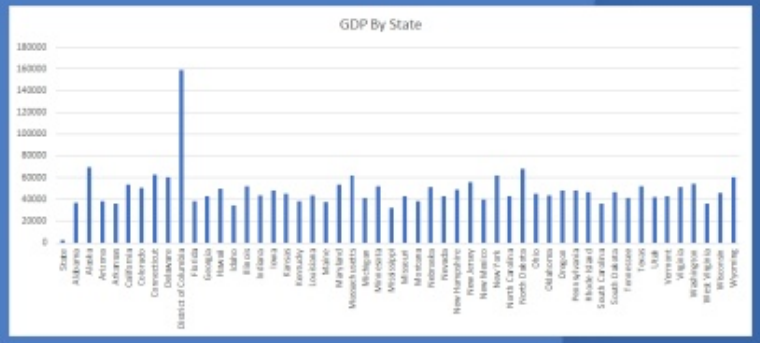




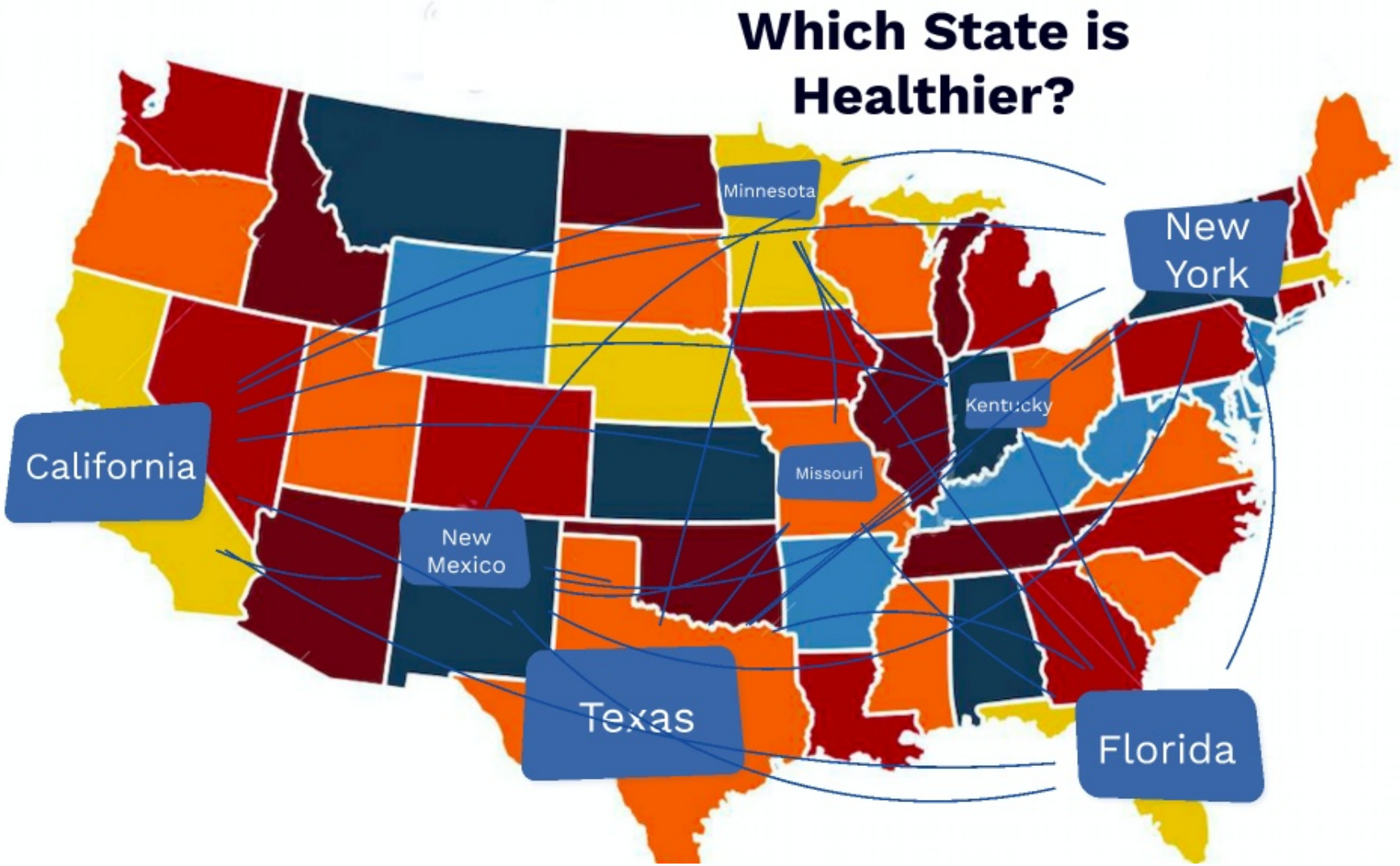
- Imported all my dependencies that I needed to use for the mapping and added my google API key.
- Created my locations variable which included the Latitude and Longitude columns from the data files.
- Next, reated the rating variable. For the Disease data rating I used the 2017 values from the DATAVALUE column and for the GDP Data rating I used the 2017 data column to compare the most recent year of general data we had. Then I plotted the figure and added a heat layer to the figure.
- Finally, saved the final figure as a PNG

Bar Chart for Comparing the density between Diseases and GDP of states

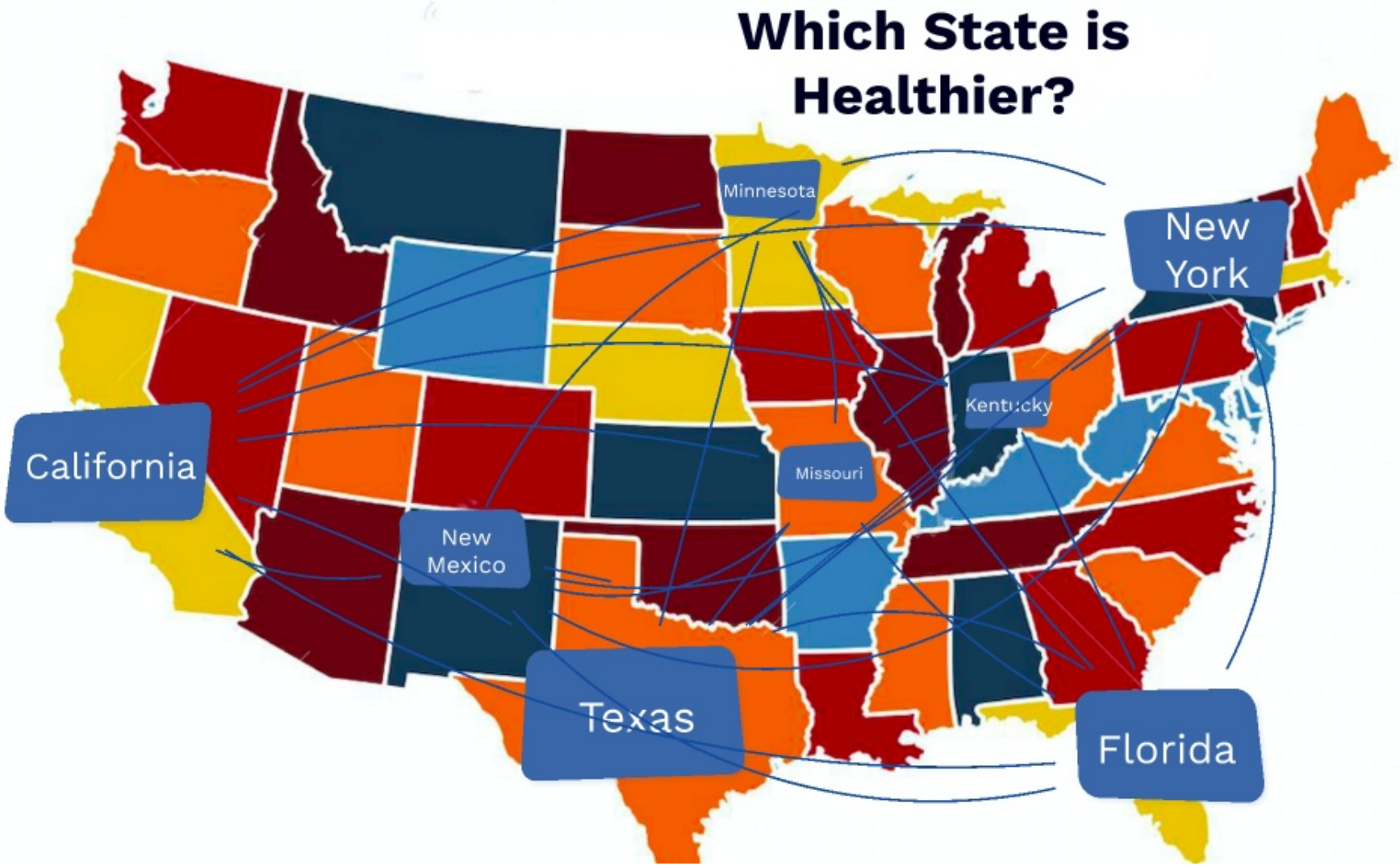


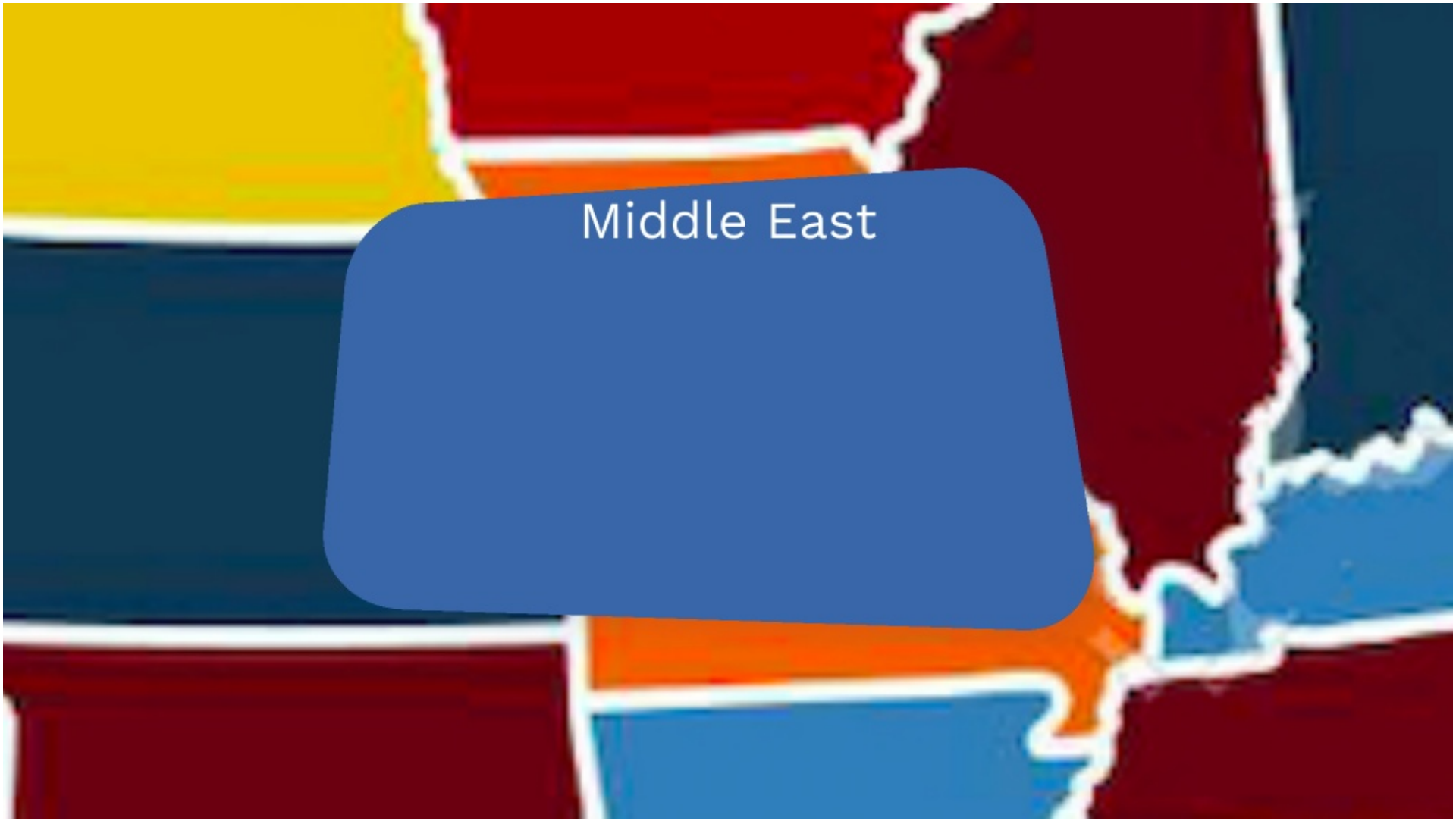


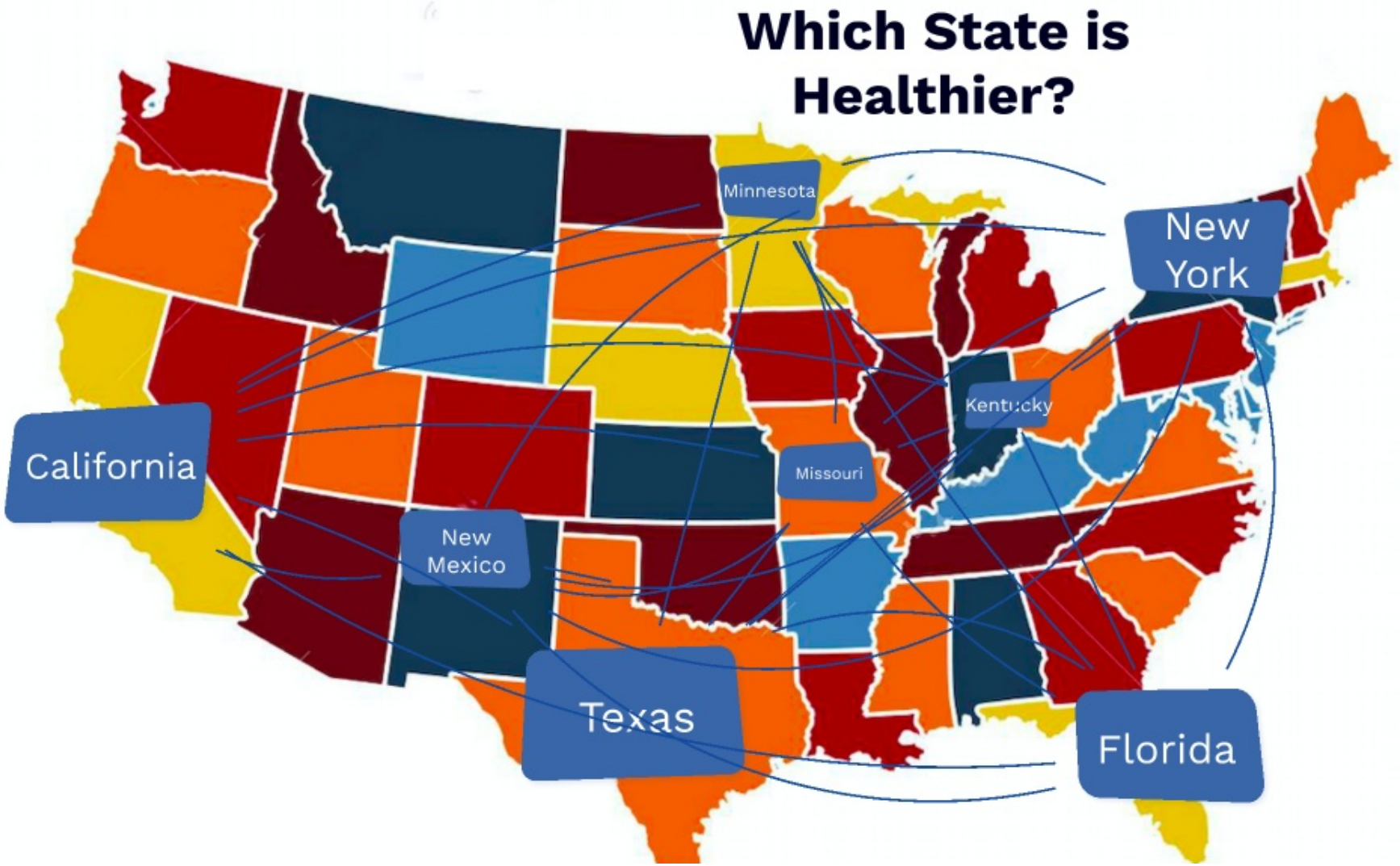
- According to our bar chart, the states with the highest disability rates were Arkansas, Mississippi, Oklahoma, and West Virginia. The states with the lowest disability rates were
- 3 of the four poorest states match up with 4 of the states with highest disability rates.
- There is a relationship between GDP and health of a state, though not for both ends of the spectrum.

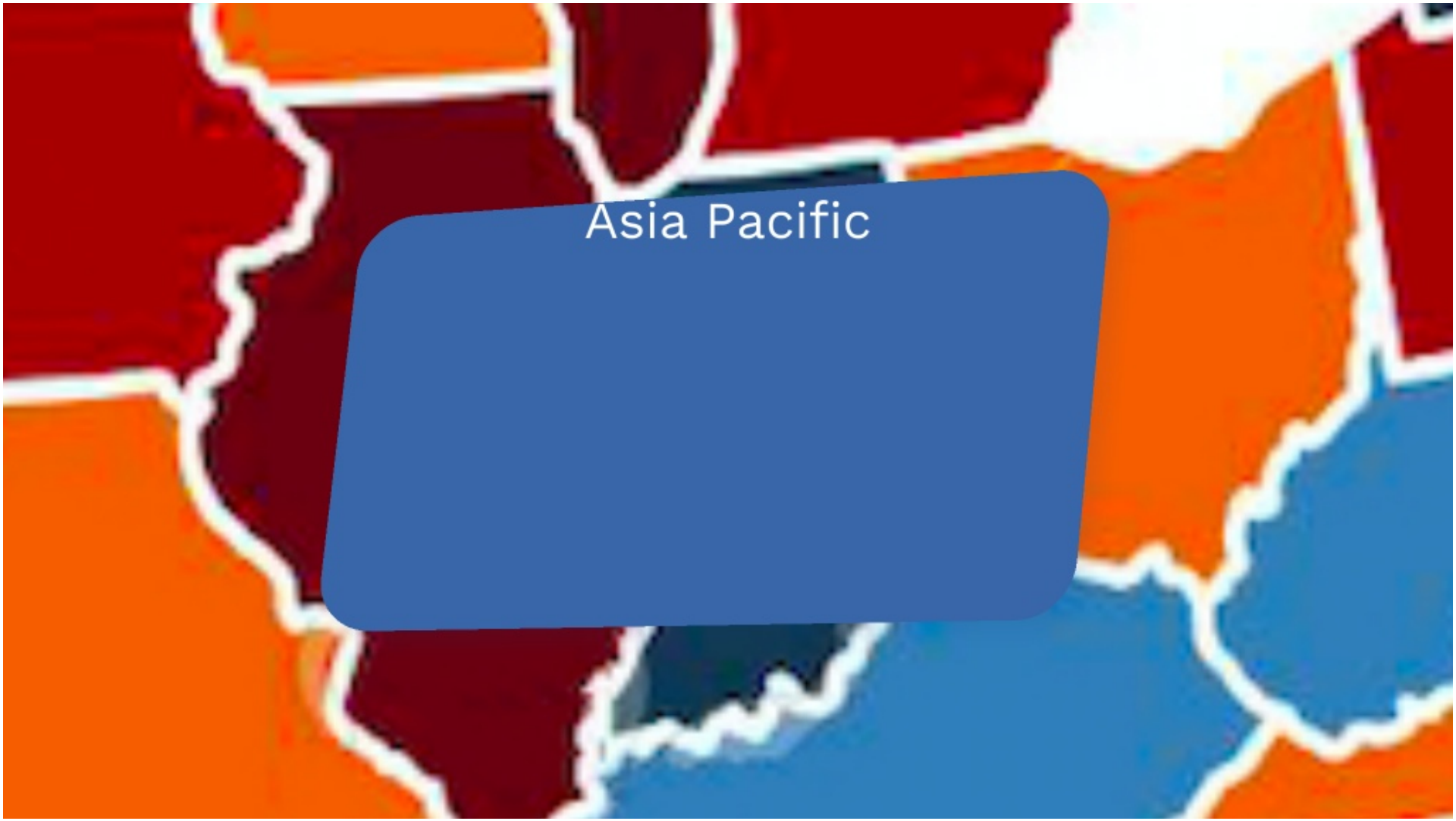

Which State is Healthier?
Minnesota
New York
Kentucky
California
Missouri
New Mexico
Texas
Florida

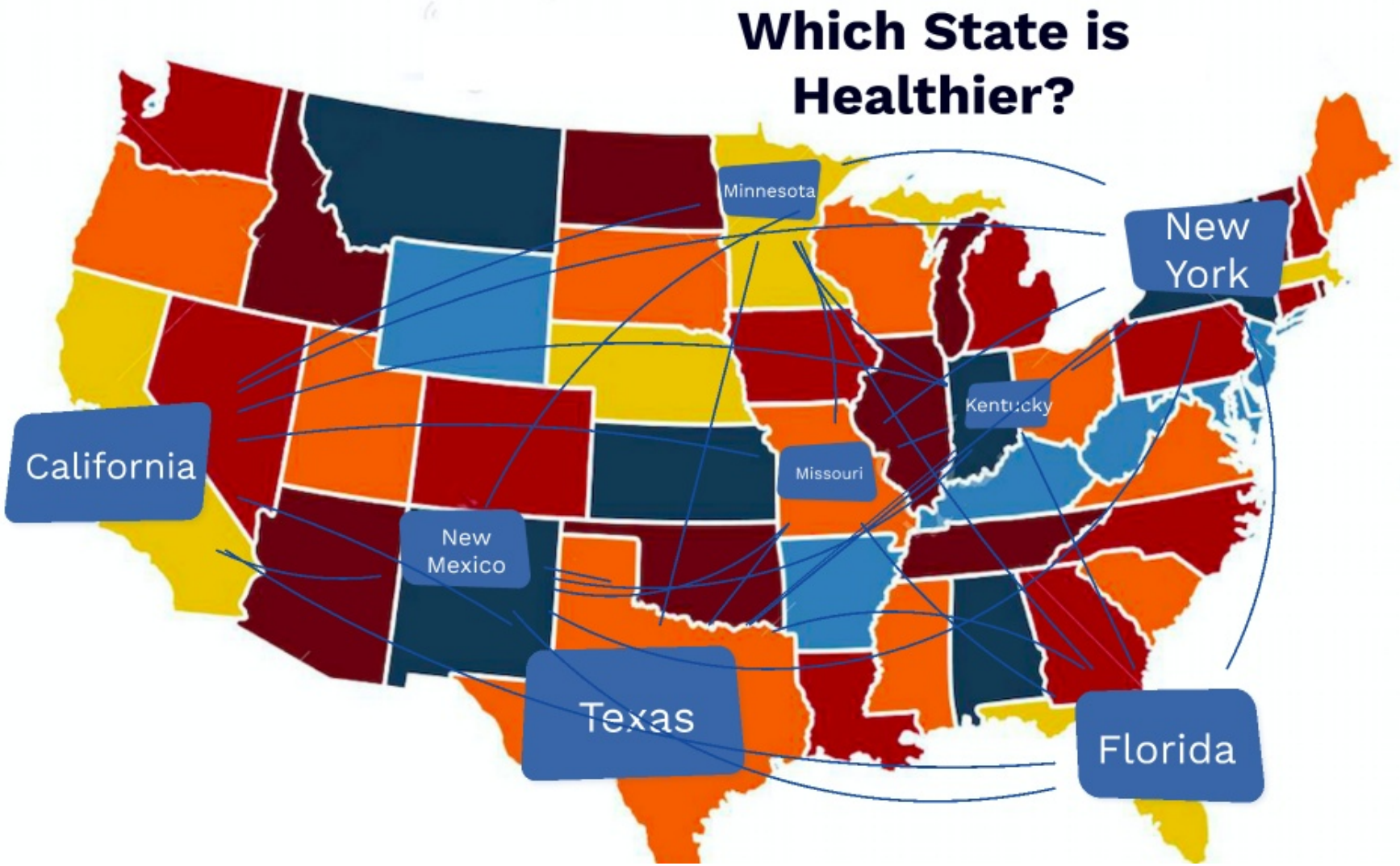
Data cleaning

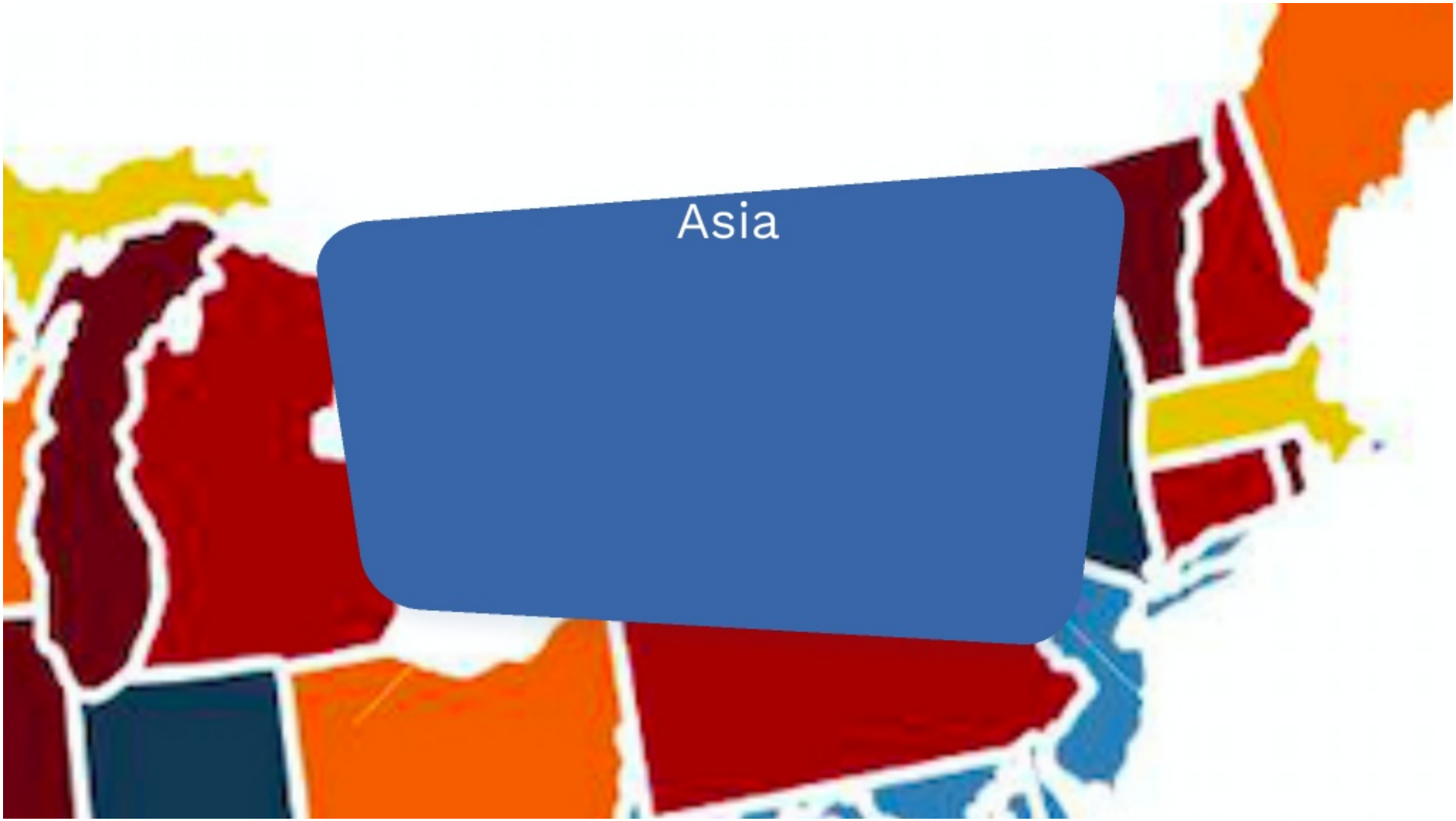

Which State is Healthier?
Minnesota
New York
Kentucky
California
Missouri
New Mexico
Texas
Florida

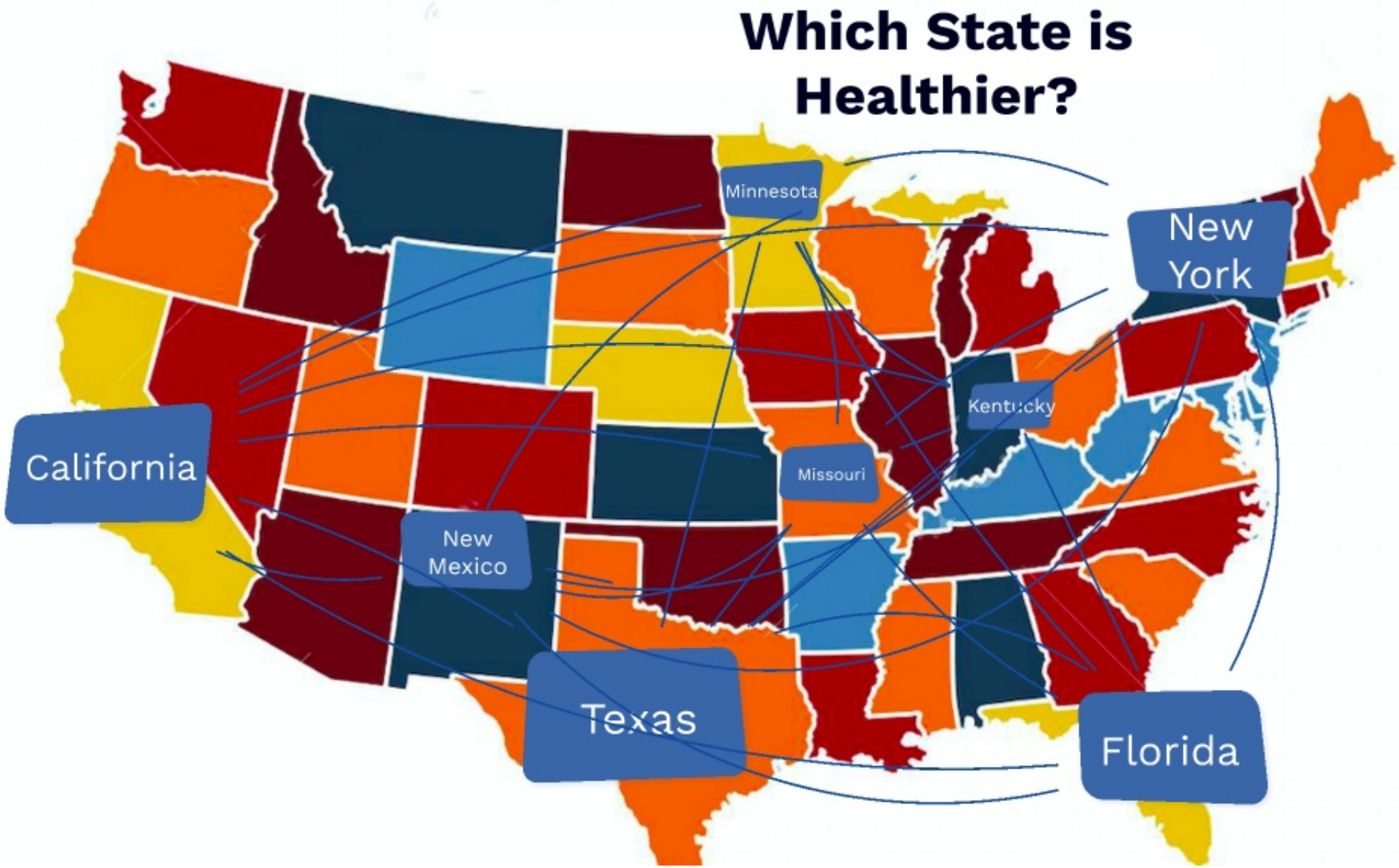

Middle East

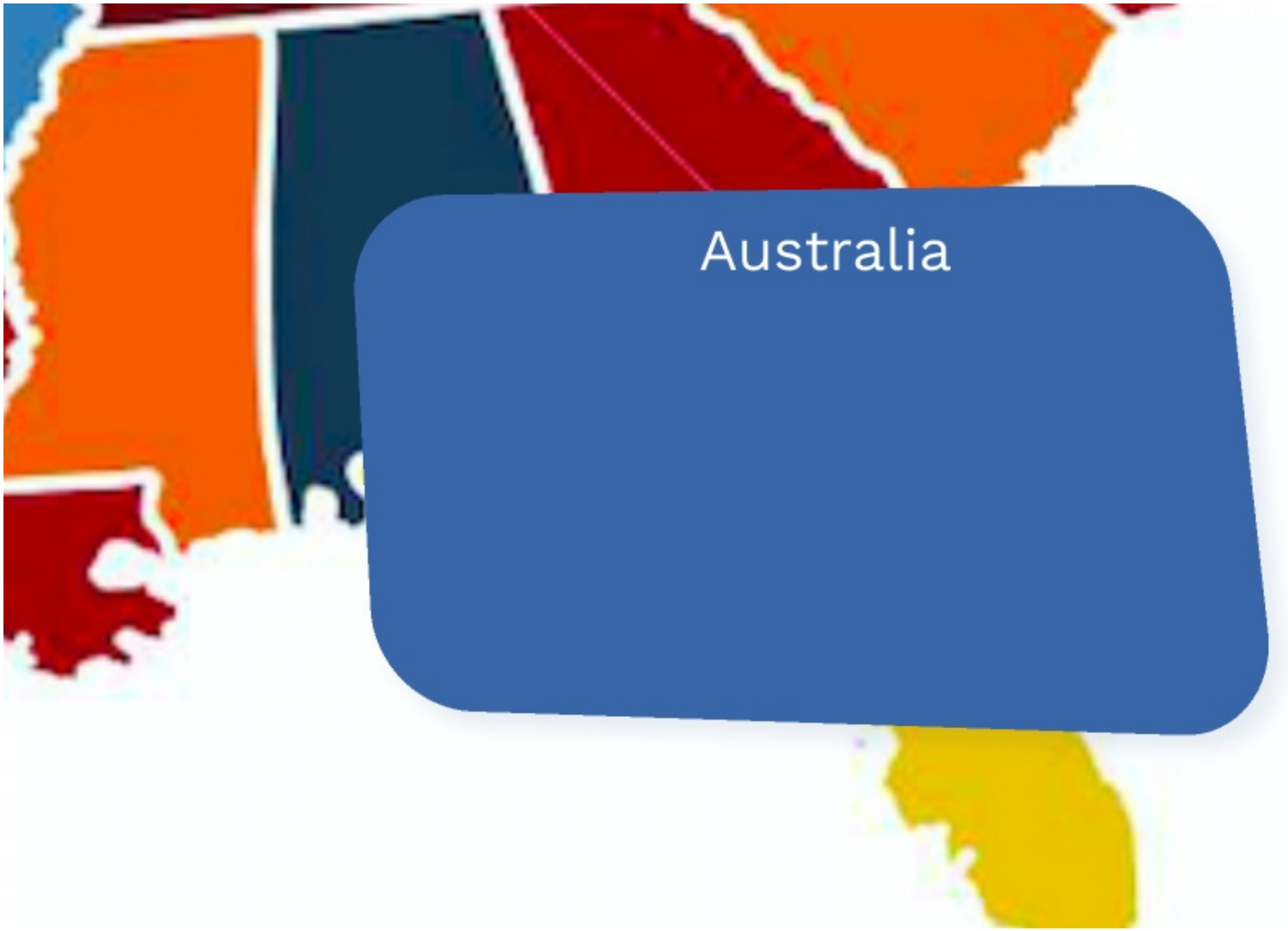

Which State is Healthier?
Minnesota
New York
Kentucky
California
Missouri
New Mexico
Texas
Florida

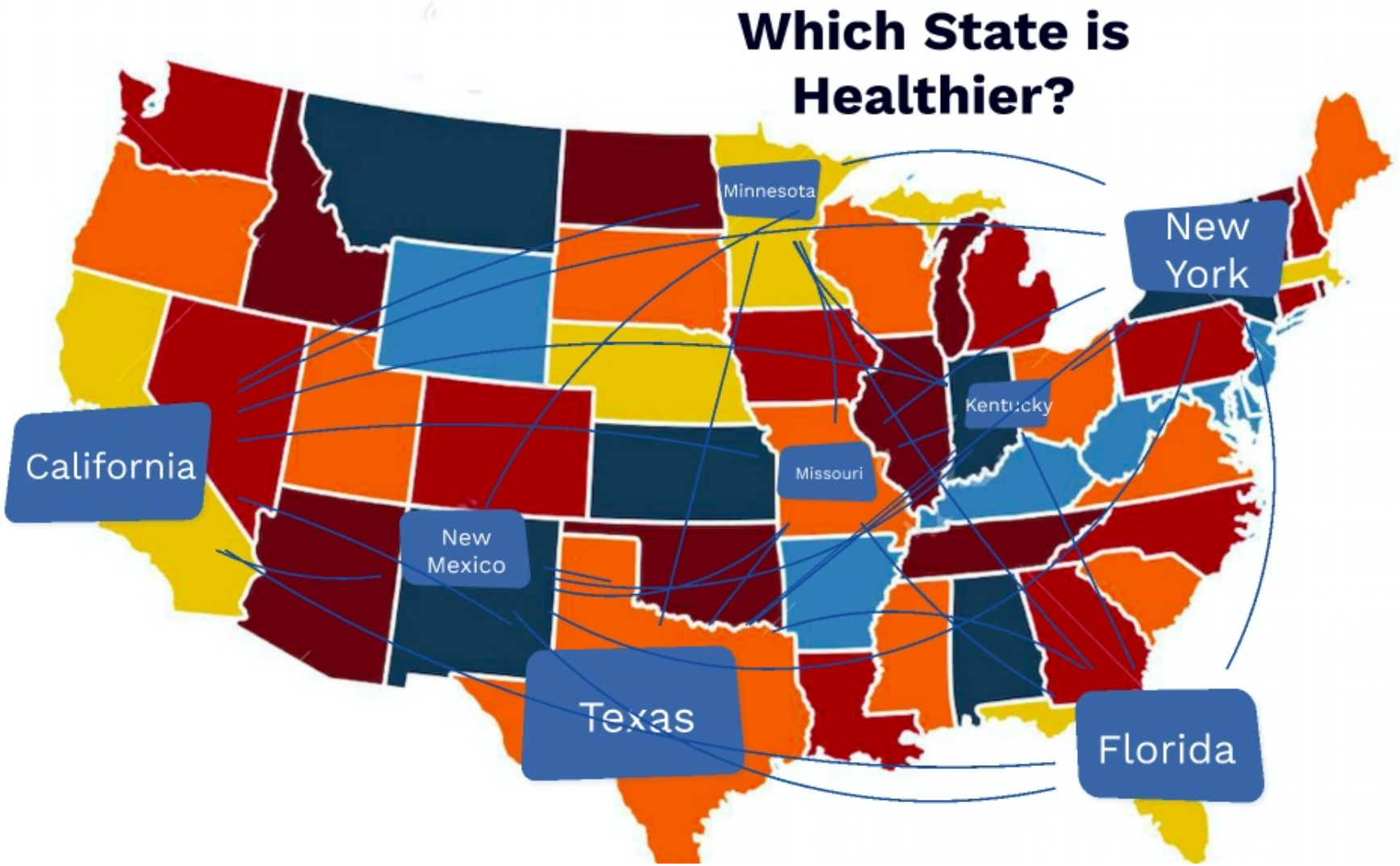

Asia Pacific


Which State is Healthier?
Minnesota
New York
Kentucky
California
Missouri
New Mexico
Texas
Florida


Asia


Which State is Healthier?
Minnesota
New York
Kentucky
California
Missouri
New Mexico
Texas
Florida


Australia


Which State is Healthier?
Minnesota
New York
Kentucky
California
Missouri
New Mexico
Texas
Florida