Import needed packages/libraries

In [ ]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from babel.numbers import format_currency
from matplotlib.colors import LinearSegmentedColormap
import matplotlib.ticker as mtick
import plotly.express as px
import plotly.graph_objects as go
```

Import csv file as a pandas dataframe

In [ ]:

```
df = pd.read_csv('us_county_sociohealth_data.csv')
```

find out the percentage of each column that's null or missing

In [ ]:

```
for col in df.columns:
    pct_missing = np.mean(df[col].isnull())
    print('{} - {}%'.format(col, round(pct_missing*100)))
```

```
fips - 0%
state - 0%
county - 0%
lat - 0%
lon - 0%
total_population - 0%
area_sqmi - 0%
population_density_per_sqmi - 0%
num_deaths - 9%
years_of_potential_life_lost_rate - 9%
percent_fair_or_poor_health - 0%
average_number_of_physically_unhealthy_days - 0%
average_number_of_mentally_unhealthy_days - 0%
percent_low_birthweight - 3%
percent_smokers - 0%
percent_adults_with_obesity - 0%
food_environment_index - 1%
percent_physically_inactive - 0%
percent_with_access_to_exercise_opportunities - 0%
percent_excessive_drinking - 0%
num_alcohol_impaired_driving_deaths - 1%
num_driving_deaths - 1%
percent_driving_deaths_with_alcohol_involvement - 1%
num_chlamydia_cases - 5%
chlamydia_rate - 5%
teen_birth_rate - 5%
num_uninsured - 0%
percent_uninsured - 0%
num_primary_care_physicians - 5%
primary_care_physicians_rate - 5%
num_dentists - 3%
dentist_rate - 3%
num_mental_health_providers - 7%
mental_health_provider_rate - 7%
preventable_hospitalization_rate - 1%
percent_with_annual_mammogram - 1%
percent_vaccinated - 1%
high_school_graduation_rate - 3%
num_some_college - 0%
population - 0%
percent_some_college - 0%
num_unemployed_CHR - 0%
labor_force - 0%
percent_unemployed_CHR - 0%
percent_children_in_poverty - 0%
eightieth_percentile_income - 0%
twentieth_percentile_income - 0%
income_ratio - 0%
num_single_parent_households_CHR - 0%
num_households_CHR - 0%
percent_single_parent_households_CHR - 0%
num_associations - 0%
social_association_rate - 0%
annual_average_violent_crimes - 6%
violent_crime_rate - 6%
num_injury_deaths - 3%
```

```
injury_death_rate - 3%
average_daily_pm2_5 - 1%
presence_of_water_violation - 1%
percent_severe_housing_problems - 0%
severe_housing_cost_burden - 0%
overcrowding - 0%
inadequate_facilities - 0%
percent_drive_alone_to_work - 0%
num_workers_who_drive_alone - 0%
percent_long_commute_drives_alone - 0%
life_expectancy - 2%
num_deaths_2 - 2%
age_adjusted_death_rate - 2%
num_deaths_3 - 39%
child_mortality_rate - 39%
num_deaths_4 - 60%
infant_mortality_rate - 60%
percent_frequent_physical_distress - 0%
percent_frequent_mental_distress - 0%
percent_adults_with_diabetes - 0%
num_hiv_cases - 27%
hiv_prevalence_rate - 27%
num_food_insecure - 0%
percent_food_insecure - 0%
num_limited_access - 1%
percent_limited_access_to_healthy_foods - 1%
num_drug_overdose_deaths - 45%
drug_overdose_mortality_rate - 45%
num_motor_vehicle_deaths - 14%
motor_vehicle_mortality_rate - 14%
percent_insufficient_sleep - 0%
num_uninsured_2 - 0%
percent_uninsured_2 - 0%
num_uninsured_3 - 0%
percent_uninsured_3 - 0%
other_primary_care_provider_rate - 1%
percent_disconnected_youth - 56%
average_grade_performance - 20%
average_grade_performance_2 - 21%
median_household_income - 0%
percent_enrolled_in_free_or_reduced_lunch - 4%
segregation_index - 34%
segregation_index_2 - 11%
homicide_rate - 59%
num_deaths_5 - 24%
suicide_rate_age_adjusted - 24%
num_firearm_fatalities - 29%
firearm_fatalities_rate - 29%
juvenile_arrest_rate - 32%
average_traffic_volume_per_meter_of_major_roadways - 0%
num_homeowners - 0%
percent_homeowners - 0%
num_households_with_severe_cost_burden - 0%
percent_severe_housing_cost_burden - 0%
population_2 - 0%
percent_less_than_18_years_of_age - 0%
```

```
percent_65_and_over - 0%
num_black - 0%
percent_black - 0%
num_american_indian_alaska_native - 0%
percent_american_indian_alaska_native - 0%
num_asian - 0%
percent_asian - 0%
num_native_hawaiian_other_pacific_islander - 0%
percent_native_hawaiian_other_pacific_islander - 0%
num_hispanic - 0%
percent_hispanic - 0%
num_non_hispanic_white - 0%
percent_non_hispanic_white - 0%
num_not_proficient_in_english - 0%
percent_not_proficient_in_english - 0%
percent_female - 0%
num_rural - 0%
percent_rural - 0%
num_housing_units - 0%
num_households_CDC - 0%
num_below_poverty - 0%
num_unemployed_CDC - 0%
per_capita_income - 0%
num_no_highschool_diploma - 0%
num_age_65_and_older - 0%
num_age_17_and_younger - 0%
num_disabled - 0%
num_single_parent_households_CDC - 0%
num_minorities - 0%
num_limited_english_abilities - 0%
num_multi_unit_housing - 0%
num_mobile_homes - 0%
num_overcrowding - 0%
num_households_with_no_vehicle - 0%
num_institutionalized_in_group_quarters - 0%
percent_below_poverty - 0%
percent_unemployed_CDC - 0%
percent_no_highschool_diploma - 0%
percent_age_65_and_older - 0%
percent_age_17_and_younger - 0%
percent_disabled - 0%
percent_single_parent_households_CDC - 0%
percent_minorities - 0%
percent_limited_english_abilities - 0%
percent_multi_unit_housing - 0%
percent_mobile_homes - 0%
percent_overcrowding - 0%
percent_no_vehicle - 0%
percent_institutionalized_in_group_quarters - 0%
percentile_rank_below_poverty - 0%
percentile_rank_unemployed - 0%
percentile_rank_per_capita_income - 0%
percentile_rank_no_highschool_diploma - 0%
percentile_rank_socioeconomic_theme - 0%
percentile_rank_age_65_and_older - 0%
percentile_rank_age_17_and_younger - 0%
```

```
percentile_rank_disabled - 0%
percentile_rank_single_parent_households - 0%
percentile_rank_household_comp_disability_theme - 0%
percentile_rank_minorities - 0%
percentile_rank_limited_english_abilities - 0%
percentile_rank_minority_status_and_language_theme - 0%
percentile_rank_multi_unit_housing - 0%
percentile_rank_mobile_homes - 0%
percentile_rank_overcrowding - 0%
percentile_rank_no_vehicle - 0%
percentile_rank_institutionalized_in_group_quarters - 0%
percentile_rank_housing_and_transportation - 0%
percentile_rank_social_vulnerability - 0%
```

Drop columns that are unrelated to objective

In [ ]:
```
df1 = df.drop(columns=['num_deaths','years_of_potential_life_lost_rate','num
```

Create new dataframe with related columns that have less than 5% missing values

In [ ]:
```
avg= df1[['state','county','area_sqmi','total_population','population_densit
```

Fill missing values with the average for each column

In [ ]:
```
avg.fillna((avg.mean()), inplace=True)
```

Drop off all records of counties whose food environment index is 6 or greater

In [ ]:
```
avg = avg[avg['food_environment_index'] < 6]
```

Find the correlation between the food environment index and other related columns

In [ ]:
```
avg.corr('pearson')['food_environment_index'].sort_values()
```

Combine the two columns counting unhealthy days into one and drop the original two

In [ ]:
```
avg['total_num_of_unhealthy_days'] = avg['average_number_of_mentally_unhealt

avg = avg.drop('average_number_of_physically_unhealthy_days', axis =1)
avg = avg.drop('average_number_of_mentally_unhealthy_days', axis =1)
```

Group each state by averaging each attribute together from the individual counties

In [ ]:
```
pop = avg.groupby('state')['total_population'].sum()
count = avg.groupby('state')['county'].count()
life = avg.groupby('state')['life_expectancy'].mean()
fei = avg.groupby('state')['food_environment_index'].mean()
days = avg.groupby('state')['total_num_of_unhealthy_days'].mean()
death = avg.groupby('state')['age_adjusted_death_rate'].mean()
food = avg.groupby('state')['percent_limited_access_to_healthy_foods'].mean(
insecure = avg.groupby('state')['percent_food_insecure'].mean()
```

```
pov = avg.groupby('state')['percent_below_poverty'].mean()
health = avg.groupby('state')['percent_fair_or_poor_health'].mean()
```

Concatenate each series together to make a dataframe

In [ ]:
```
total = pd.concat([pop,count,life,fei,days,death,food,insecure,pov,health],
```

Reset the index of new dataframe

In [ ]:
```
total = total.reset_index(0)
```

Round the age adjusted death rate column to the nearest whole number for cleaning purposes

In [ ]:
```
total['age_adjusted_death_rate'] = total['age_adjusted_death_rate'].round(0)

total = total.round(1)
```

Plot a bar chart showing the food environment index by each state

In [ ]:
```
total = total.sort_values(by='food_environment_index')

cmap = plt.get_cmap('YlGnBu')
norm = plt.Normalize(total['food_environment_index'].min(), total['food_envi
colors = cmap(norm(total['food_environment_index']))

plt.figure(figsize=(10,5))
plt.bar(total['state'], total['food_environment_index'], color = colors)
plt.xlabel("State")
plt.xticks(rotation=85)
plt.ylabel("Food Environment Index")
plt.title("Food Environment Index by State")
plt.show()
```

Plot a bar chart of each state's percentage of people with limited access to healthy food

In [ ]:
```
total = total.sort_values(by='percent_limited_access_to_healthy_foods', asce

cmap = plt.get_cmap('plasma')
norm = plt.Normalize(total['percent_limited_access_to_healthy_foods'].min(),
colors1 = cmap(norm(total['percent_limited_access_to_healthy_foods']))

plt.figure(figsize=(10,5))
plt.bar(total['state'], total['percent_limited_access_to_healthy_foods'], co
plt.xlabel("State")
plt.xticks(rotation=85)
plt.ylabel("Limited Access Percentage")
plt.gca().yaxis.set_major_formatter(plt.FuncFormatter('{:.0f}%'.format))
plt.title("Percent with Limited Access to Healthy Food by State")
plt.show()
```

Plot a bar chart of each state's percentage of citizens below the poverty line

In [ ]:

```
total = total.sort_values(by='percent_below_poverty', ascending=False)

cmap = plt.get_cmap('viridis')
norm = plt.Normalize(total['percent_below_poverty'].min(), total['percent_be
colors1 = cmap(norm(total['percent_below_poverty']))

plt.figure(figsize=(10,5))
plt.bar(total['state'], total['percent_below_poverty'], color=colors1)
plt.xlabel("State")
plt.xticks(rotation=85)
plt.ylabel("Percent Below Poverty")
plt.gca().yaxis.set_major_formatter(plt.FuncFormatter('{:.0f}%'.format))
plt.title("Percent Below Poverty by State")
plt.show()
```

Plot a bar chart of each state's percentage of citizens having continually fair or poor health

In [ ]:

```
total = total.sort_values(by='percent_fair_or_poor_health', ascending=False)

cmap = plt.get_cmap('cividis')
norm = plt.Normalize(total['percent_fair_or_poor_health'].min(), total['perc
colors1 = cmap(norm(total['percent_fair_or_poor_health']))

plt.figure(figsize=(10,5))
plt.bar(total['state'], total['percent_fair_or_poor_health'], color=colors1)
plt.xlabel("State")
plt.xticks(rotation=85)
plt.ylabel("Percent Fair/Poor Health")
plt.gca().yaxis.set_major_formatter(plt.FuncFormatter('{:.0f}%'.format))
plt.title("Percent Fair/Poor Health by State")
plt.show()
```

Plot a bar chart of each state's percentage of citizens who are considered food insecure

In [ ]:

```
total = total.sort_values(by='percent_food_insecure', ascending=False)

cmap = plt.get_cmap('inferno')
norm = plt.Normalize(total['percent_food_insecure'].min(), total['percent_fo
colors1 = cmap(norm(total['percent_food_insecure']))

plt.figure(figsize=(10,5))
plt.bar(total['state'], total['percent_food_insecure'], color=colors1)
plt.xlabel('State')
plt.xticks(rotation=85)
plt.ylabel("Percent of Food Insecure")
plt.gca().yaxis.set_major_formatter(plt.FuncFormatter('{:.0f}%'.format))
plt.title("Percent Food Insecure by State")
plt.show()
```

Sort dataframe by state alphabetically

In [ ]:

```
total = total.sort_values(by='state')
```

Select rows containing the states of each region represented

In [ ]:

```
south = total.iloc[np.r_[0:1, 3:4, 6:8, 11:14, 19:20, 22:23, 24:25, 26:28, 2
midwest = total.iloc[np.r_[9:11, 14:15, 16:17, 20:22, 25:26,]]
west = total.iloc[np.r_[1:3, 4:6, 8:9, 15:16, 17:19, 23:24, 28:29, 30:31]]
```

Create a pivot table heatmap with each region's dataframe based on their food environment index

In [ ]:

```
result = west.pivot(index='state', columns='food_environment_index', values=
sns.heatmap(result, annot=True, fmt="g", cmap='RdYlGn')
plt.show()

result1 = south.pivot(index='state', columns='food_environment_index', value
sns.heatmap(result1, annot=True, fmt="g", cmap='RdYlGn')
plt.show()

result2 = midwest.pivot(index='state', columns='food_environment_index', val
sns.heatmap(result2, annot=True, fmt="g", cmap='RdYlGn')
plt.show()
```

Seperate the original dataframe for each state below a 4.5 food environment index and concatenate them together

In [ ]:

```
sd = avg[avg['state'] == 'South Dakota']
idaho = avg[avg['state'] == 'Idaho']
alaska = avg[avg['state'] == 'Alaska']
miss = avg[avg['state'] == 'Mississippi']
lst = [sd, idaho, alaska, miss]
food5 = pd.concat(lst)
```

Find the most correlated attributes to the food environment index in Mississippi

In [ ]:

```
miss.corr('pearson')['food_environment_index'].sort_values()
```

Find the most correlated attributes to the food environment index in South Dakota

In [ ]:

```
sd.corr('pearson')['food_environment_index'].sort_values()
```

Find the most correlated attributes to the food environment index in Alaska

In [ ]:
```
alaska.corr('pearson')['food_environment_index'].sort_values()
```

Find the most correlated attributes to the food environment index in Idaho

In [ ]:
```
idaho.corr('pearson')['food_environment_index'].sort_values()
```

Create subset of original dataframe wit those attributes that are closely correlated to the food environment index in those four states.

In [ ]:
```
avg1 = avg[['state','life_expectancy','food_environment_index','total_num_of
```

Rename columns to more readable formats for plotting purposes and drop the old columns

In [ ]:
```
avg1['% limited access to healthy foods'] = avg1['percent_limited_access_to_
avg1['total num of unhealthy days'] = avg1['total_num_of_unhealthy_days']
avg1['life expectancy'] = avg1['life_expectancy']
avg1['food environment index'] = avg1['food_environment_index']
avg1['age adjusted death rate'] = avg1['age_adjusted_death_rate']
avg1['% food insecure'] = avg1['percent_food_insecure']
avg1['% below poverty'] = avg1['percent_below_poverty']
avg1['% fair/poor health'] = avg1['percent_fair_or_poor_health']

avg1 = avg1.drop(columns= ['life_expectancy','food_environment_index','total
```

Create new subset dataframes for each specific state

In [ ]:
```
sd1 = avg1[avg1['state'] == 'South Dakota']
idaho1 = avg1[avg1['state'] == 'Idaho']
alaska1 = avg1[avg1['state'] == 'Alaska']
miss1 = avg1[avg1['state'] == 'Mississippi']
```

Create food environment index correlation heatmap for South Dakota

In [ ]:
```
f = plt.figure(figsize=(10, 10))
plt.matshow(sd1.corr(), fignum=f.number, cmap = 'RdYlGn')
plt.xticks(range(sd1.select_dtypes(['float','int']).shape[1]), sd1.select_dt
plt.yticks(range(sd1.select_dtypes(['float','int']).shape[1]), sd1.select_dt
cb = plt.colorbar()
cb.ax.tick_params(labelsize=16)
plt.title('South Dakota Correlation', fontsize=16);
```

Create food environment index correlation heatmap for Mississippi

In [ ]:
```
t = plt.figure(figsize=(10, 10))
plt.matshow(miss1.corr(), fignum=t.number, cmap = 'RdYlGn')
plt.xticks(range(miss1.select_dtypes(['float','int']).shape[1]), miss1.selec
plt.yticks(range(miss1.select_dtypes(['float','int']).shape[1]), miss1.selec
cb = plt.colorbar()
cb.ax.tick_params(labelsize=10)
plt.title('Mississippi Correlation', fontsize=16);
```

Create food environment index correlation heatmap for Alaska

In [ ]:

```
j = plt.figure(figsize=(10, 10))
plt.matshow(alaska1.corr(), fignum=j.number, cmap='RdYlGn')
plt.xticks(range(alaska1.select_dtypes(['float','int']).shape[1]), alaska1.s
plt.yticks(range(alaska1.select_dtypes(['float','int']).shape[1]), alaska1.s
cb = plt.colorbar()
cb.ax.tick_params(labelsize=10)
plt.title('Alaska Correlation', fontsize=16);
```

Create food environment index correlation heatmap for Idaho

In [ ]:

```
g = plt.figure(figsize=(10, 10))
plt.matshow(idaho1.corr(), fignum=g.number, cmap = 'RdYlGn')
plt.xticks(range(idaho1.select_dtypes(['float','int']).shape[1]), idaho1.sel
plt.yticks(range(idaho1.select_dtypes(['float','int']).shape[1]), idaho1.sel
cb = plt.colorbar()
cb.ax.tick_params(labelsize=10)
plt.title('Idaho Correlation', fontsize=16);
```

There's a multitude of socioeconomic factors determining a state's food environment index. The above highlighted some of the most important components.