

Data-Wrangling

September 6, 2019

0.1 Capstone #1

0.1.1 File Descriptions

- DATA_ELEMENT_DESCRIPTION.csv defines each data element and indicates where its description is found in Data Sources, Definitions, and Notes.
- DEFINED_DATA_VALUE.csv defines the meaning of specific values (such as missing or suppressed data).
- DEMOGRAPHICS.csv identifies the data elements and values in the Demographics indicator domain.
- HEALTHY_PEOPLE_2010.csv identifies the Healthy People 2010 Targets and the U.S. Percentages or Rates.
- LEADING_CAUSES_OF_DEATH.csv identifies the data elements and values in the Leading Causes of Death indicator domain.
- MEASURES_OF_BIRTH_AND_DEATH.csv identifies the data elements and values in the Measures of Birth and Death indicator domain.
- PREVENTIVE_SERVICES_USE.csv identifies the data elements and values in the Preventive Services indicator domain.
- RELATIVE_HEALTH_IMPORTANCE.csv identifies the data elements and values in the Relative Health Importance indicator domain.
- RISK_FACTORS_AND_ACCESS_TO_CARE.csv identifies the data elements and values in the Risk Factors and Access to Care indicator domain.
- SUMMARY_MEASURES_OF_HEALTH.csv identifies the data elements and values in the Summary Measures of Health indicator domain.
- VULNERABLE_POPS_AND_ENV_HEALTH.csv identifies the data elements and values in the Vulnerable Populations and Environmental Health indicator domain.

Data source can be found [here](#)

```
[1]: import glob
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import matplotlib.cm as cm
from matplotlib.pyplot import figaspect
from mpl_toolkits.axes_grid1 import make_axes_locatable
import geopandas as gpd
import seaborn as sns
```

```
mpl.rcParams.update({'font.size': 14})
```

```
[2]: #Reading all 11 csv files
files = sorted(glob.glob('../CHSI_dataset/*.csv'))
```

0.1.2 File Initialization

```
[3]: DATA_ELEMENT_DESCRIPTION = pd.read_csv(files[0])
DEFINED_DATA_VALUE = pd.read_csv(files[1])
DEMOGRAPHICS = pd.read_csv(files[2])
HEALTHY_PEOPLE_2010 = pd.read_csv(files[3])
LEADING_CAUSES_OF_DEATH = pd.read_csv(files[4])
MEASURES_OF_BIRTH_AND_DEATH = pd.read_csv(files[5])
PREVENTIVE_SERVICES_USE = pd.read_csv(files[6])
RELATIVE_HEALTH_IMPORTANCE = pd.read_csv(files[7])
RISK_FACTORS_AND_ACCESS_TO_CARE = pd.read_csv(files[8])
SUMMARY_MEASURES_OF_HEALTH = pd.read_csv(files[9])
VULNERABLE_POPS_AND_ENV_HEALTH = pd.read_csv(files[10])
```

```
[4]: #Auxiliary files
DATA_ELEMENT_DESCRIPTION
DEFINED_DATA_VALUE
HEALTHY_PEOPLE_2010
```

```
[4]:      Categories                                     Elements \
0      Birth Measures                Low Birth Wt. (<2500 g)
1      Birth Measures            Very Low Birth Wt. (<1500 g)
2      Birth Measures            Premature Births (<37 weeks)
3      Birth Measures            Births to Women under 18
4      Birth Measures            Births to Women over 40
5      Birth Measures            Births to Unmarried Women
6      Birth Measures            No Care in First Trimester
7      Infant Mortality                Infant Mortality
8      Infant Mortality  White non Hispanic Infant Mortality
9      Infant Mortality  Black non Hispanic Infant Mortality
10     Infant Mortality                Hispanic Infant Mortality
11     Infant Mortality                Neonatal Infant Mortality
12     Infant Mortality            Post-neonatal Infant Mortality
13     Death Measures                Breast Cancer (Female)
14     Death Measures                Colon Cancer
15     Death Measures            Coronary Heart Disease
16     Death Measures                Homicide
17     Death Measures                Lung Cancer
18     Death Measures            Motor Vehicle Injuries
19     Death Measures                Stroke
20     Death Measures                Suicide
21     Death Measures            Unintentional Injury
```

	US_Pct_or_Rate_2003	Healthy_People_2010_Target
0	7.9	5.0
1	1.4	0.9
2	12.3	7.6
3	3.4	-9998.9
4	2.6	-9998.9
5	34.6	-9998.9
6	16.0	10.0
7	6.8	4.5
8	5.7	4.5
9	13.6	4.5
10	5.6	4.5
11	4.6	2.9
12	2.2	1.2
13	25.3	21.3
14	19.1	13.7
15	172.0	162.0
16	6.0	2.8
17	54.1	43.3
18	14.8	8.0
19	53.0	50.0
20	10.8	4.8
21	37.3	17.1

```
[5]: LEADING_CAUSES_OF_DEATH.head()
```

```
[5]: State_FIPS_Code County_FIPS_Code CHSI_County_Name CHSI_State_Name \
0 1 1 Autauga Alabama
1 1 3 Baldwin Alabama
2 1 5 Barbour Alabama
3 1 7 Bibb Alabama
4 1 9 Blount Alabama
```

	CHSI_State_Abbr	Strata_ID_Number	A_Wh_Comp	CI_Min_A_Wh_Comp	\
0	AL	29	-1111	-1111	
1	AL	16	57	39	
2	AL	51	-1111	-1111	
3	AL	42	-1111	-1111	
4	AL	28	34	17	

	CI_Max_A_Wh_Comp	A_Bl_Comp	...	F_Bl_Cancer	CI_Min_F_Bl_Cancer	\
0	-1111	-1111	...	19	14	
1	75	-1111	...	20	15	
2	-1111	-1111	...	26	22	
3	-1111	-1111	...	20	14	
4	52	-1111	...	28	10	

	CI_Max_F_Bl_Cancer	F_Ot_Cancer	CI_Min_F_Ot_Cancer	CI_Max_F_Ot_Cancer	\
--	--------------------	-------------	--------------------	--------------------	---

0	23	-1111	-1111	-1111
1	25	-1111	-1111	-1111
2	31	-1111	-1111	-1111
3	25	-1111	-1111	-1111
4	46	-1111	-1111	-1111

	F_Hi_Cancer	CI_Min_F_Hi_Cancer	CI_Max_F_Hi_Cancer	LCD_Time_Span
0	-1111	-1111	-1111	1999-2003
1	-1111	-1111	-1111	2001-2003
2	-1111	-1111	-1111	1999-2003
3	-1111	-1111	-1111	1994-2003
4	-1111	-1111	-1111	1999-2003

[5 rows x 235 columns]

```
[6]: #Outlier column name
DATA_ELEMENT_DESCRIPTION.update(
    DATA_ELEMENT_DESCRIPTION.replace('C_Ot_homicide', 'C_Ot_Homicide'))
LEADING_CAUSES_OF_DEATH.rename(columns={'C_Ot_homicide': 'C_Ot_Homicide'},
    inplace=True)

#Functions to describe acronyms
def description(param):
    if param in list(DATA_ELEMENT_DESCRIPTION.COLUMN_NAME):
        description_value = DATA_ELEMENT_DESCRIPTION[DATA_ELEMENT_DESCRIPTION.
            COLUMN_NAME==param].DESCRIPTION.tolist()[0]
    else:
        description_value = ''
    return description_value

def short_description(param):
    list_of_removing = ['County data,', 'death measures,', 'birth measures,',
        '']
    result = description(param)
    for i in list_of_removing:
        if result.__contains__(i):
            result = result.replace(i, '')
    return result

def short_race_description(param):
    list_of_removing = ['Black', 'Hispanic', 'White', 'other',
        'County data,', 'death measures,', 'birth measures,',
        '']
    result = description(param)
    for i in list_of_removing:
        if result.__contains__(i):
            result = result.replace(i, '')
```

```
return result
```

```
[7]: #Common columns in all worksheets
county_info = ['State_FIPS_Code', 'County_FIPS_Code', 'CHSI_County_Name',
               'CHSI_State_Name', 'CHSI_State_Abbr', 'Strata_ID_Number']
```

```
[8]: #Joining files
worksheets = [DEMOGRAPHICS,
LEADING_CAUSES_OF_DEATH,
MEASURES_OF_BIRTH_AND_DEATH,
PREVENTIVE_SERVICES_USE,
RELATIVE_HEALTH_IMPORTANCE,
RISK_FACTORS_AND_ACCESS_TO_CARE,
SUMMARY_MEASURES_OF_HEALTH,
VULNERABLE_POPS_AND_ENV_HEALTH]
```

```
[9]: #Replacing not available/reported data by zeros
nan_values = [-1111, -1111.1, -1, -9999, -2222, -2222.2, -2]
for worksheet in worksheets:
    nda = []
    for i, nan_value in enumerate(nan_values):
        nda.append(worksheet.loc[:, :] != nan_value)
    worksheet.update(worksheet.where(nda[i]).fillna(0))
```

```
[10]: #Time dependent data should be identified and normalized
time_dependent_worksheets = [LEADING_CAUSES_OF_DEATH,
MEASURES_OF_BIRTH_AND_DEATH,
PREVENTIVE_SERVICES_USE,
VULNERABLE_POPS_AND_ENV_HEALTH]

time_spans = ['1994-2003', '1999-2003', '2001-2003']
time_span_convert = [7,4,2]
for worksheet in time_dependent_worksheets:
    for i in list(worksheet.columns):
        if i.__contains__('Time_Span'):
            worksheet[i].replace(time_spans, time_span_convert, inplace=True)
            worksheet.rename(columns={i: 'Time'}, inplace=True)

for worksheet in time_dependent_worksheets:
    for i in list(worksheet.columns):
        if worksheet.columns.get_loc(i) < worksheet.columns.get_loc('Time') and
        → i not in county_info:
            worksheet[i]=worksheet[i]/worksheet['Time']
```

```
[11]: #Merging all clean data files
df = worksheets[0]
for name in worksheets[1:]:
```

```
df = df.merge(name, on=county_info.append('Time'), how='outer', sort=True).
→fillna(0)
```

```
[12]: df.head()
```

```
[12]: State_FIPS_Code County_FIPS_Code CHSI_County_Name CHSI_State_Name \
0          1          1      Autauga      Alabama
1          1          3      Baldwin      Alabama
2          1          5      Barbour      Alabama
3          1          7        Bibb      Alabama
4          1          9      Blount      Alabama
```

```
CHSI_State_Abbr Strata_ID_Number \
0          AL          29
1          AL          16
2          AL          51
3          AL          42
4          AL          28
```

```
Strata_Determining_Factors Number_Counties \
0 frontier status, population size, poverty, age 37
1 frontier status, population size, poverty, age 27
2 frontier status, population size, poverty, age... 33
3 frontier status, population size, poverty, age 53
4 frontier status, population size, poverty, age 39
```

```
Population_Size Min_Population_Size ... Shig_Rpt Shig_Rpt_Ind \
0      48612      28447 ... 1.000000 0.750000
1     162586     118395 ... 20.500000 2.000000
2      28414      27269 ... 0.250000 0.750000
3      21516       8134 ... 0.428571 0.428571
4      55725      29009 ... 2.750000 1.000000
```

```
Shig_Exp Toxic_Chem Carbon_Monoxide_Ind Nitrogen_Dioxide_Ind \
0 2.750000 720799.250000 0.250000 0.250000
1 6.500000 17832.000000 0.500000 0.500000
2 4.000000 12468.500000 0.250000 0.250000
3 2.285714 1043.142857 0.142857 0.142857
4 1.250000 222255.000000 0.250000 0.250000
```

```
Sulfur_Dioxide_Ind Ozone_Ind Particulate_Matter_Ind Lead_Ind
0 0.250000 0.250000 0.250000 0.250000
1 0.500000 1.000000 0.500000 0.500000
2 0.250000 0.250000 0.250000 0.250000
3 0.142857 0.142857 0.142857 0.142857
4 0.250000 0.250000 0.250000 0.250000
```

```
[5 rows x 533 columns]
```

0.1.3 Redundant columns

```
[13]: observation = list(df.columns)
description_list = [description(i) for i in observation]
def observ_keyword(variable):
    results = []
    for i, j in enumerate(description_list):
        if variable in str(j):
            results.append(observation[i])
    return results

[14]: filter_percentile = list(description(i) for i in
    ↪list(observ_keyword('percentile')))
filter_CI = list(description(i) for i in list(observ_keyword('Confidence_
    ↪interval')))
filter_fav = list(description(i) for i in list(observ_keyword('Favorable_
    ↪indicator')))

filters = list(observ_keyword('Favorable indicator')
    + observ_keyword('Confidence interval')
    + observ_keyword('percentile'))
df = df.drop(columns=filters)

[15]: #Neutral columns
for i in list(df.columns):
    if len(np.unique(df[i]))==1 and i!='D_0t_HIV':
        df=df.drop(columns=i)
df.head()
```

```
[15]: State_FIPS_Code  County_FIPS_Code  CHSI_County_Name  CHSI_State_Name  \
0                1                1      Autauga      Alabama
1                1                3      Baldwin      Alabama
2                1                5      Barbour      Alabama
3                1                7      Bibb      Alabama
4                1                9      Blount      Alabama

CHSI_State_Abbr  Strata_ID_Number  \
0              AL              29
1              AL              16
2              AL              51
3              AL              42
4              AL              28

Strata_Determining_Factors  Number_Counties  \
0  frontier status, population size, poverty, age      37
1  frontier status, population size, poverty, age      27
2  frontier status, population size, poverty, age...     33
3  frontier status, population size, poverty, age      53
4  frontier status, population size, poverty, age      39
```

	Population_Size	Population_Density	...	Salm_Exp	Shig_Rpt	Shig_Exp	\
0	48612	82.0	...	7.750000	1.000000	2.750000	
1	162586	102.0	...	33.500000	20.500000	6.500000	
2	28414	32.0	...	7.250000	0.250000	4.000000	
3	21516	35.0	...	4.571429	0.428571	2.285714	
4	55725	86.0	...	7.750000	2.750000	1.250000	

	Toxic_Chem	Carbon_Monoxide_Ind	Nitrogen_Dioxide_Ind	\
0	720799.250000	0.250000	0.250000	
1	17832.000000	0.500000	0.500000	
2	12468.500000	0.250000	0.250000	
3	1043.142857	0.142857	0.142857	
4	222255.000000	0.250000	0.250000	

	Sulfur_Dioxide_Ind	Ozone_Ind	Particulate_Matter_Ind	Lead_Ind
0	0.250000	0.250000	0.250000	0.250000
1	0.500000	1.000000	0.500000	0.500000
2	0.250000	0.250000	0.250000	0.250000
3	0.142857	0.142857	0.142857	0.142857
4	0.250000	0.250000	0.250000	0.250000

[5 rows x 198 columns]

```
[16]: df.describe().transpose()
```

```
[16]:
```

	count	mean	std	min	\
State_FIPS_Code	3141.0	30.304680	15.134423	1.000000	
County_FIPS_Code	3141.0	103.716651	107.999484	1.000000	
Strata_ID_Number	3141.0	44.696275	25.118434	1.000000	
Number_Counties	3141.0	38.486151	10.290195	15.000000	
Population_Size	3141.0	94368.164279	306431.655763	62.000000	
...	
Nitrogen_Dioxide_Ind	3141.0	0.235275	0.125723	0.142857	
Sulfur_Dioxide_Ind	3141.0	0.235275	0.125723	0.142857	
Ozone_Ind	3141.0	0.263565	0.202474	0.142857	
Particulate_Matter_Ind	3141.0	0.237595	0.134062	0.142857	
Lead_Ind	3141.0	0.235435	0.126373	0.142857	

	25%	50%	75%	max
State_FIPS_Code	18.000000	29.000000	45.00	56.0
County_FIPS_Code	35.000000	79.000000	133.00	840.0
Strata_ID_Number	23.000000	44.000000	66.00	88.0
Number_Counties	32.000000	37.000000	45.00	62.0
Population_Size	11211.000000	25235.000000	64040.00	9935475.0
...
Nitrogen_Dioxide_Ind	0.142857	0.142857	0.25	0.5
Sulfur_Dioxide_Ind	0.142857	0.142857	0.25	0.5

Ozone_Ind	0.142857	0.142857	0.25	1.0
Particulate_Matter_Ind	0.142857	0.142857	0.25	1.0
Lead_Ind	0.142857	0.142857	0.25	1.0

[194 rows x 8 columns]

0.1.4 Demographics Map

To visualize the health indices throughout the country, I merge the map shapefile and data to `us_merge` dataframes. Because Alaska and Hawaii are further away and out of proportion I merged them separately. `US_plot` function creates the US map with any numeric variables in the data –default color is blue.

```
[17]: us_map = gpd.read_file('../US-map/UScounties.shp')
us_map_land = us_map[ (us_map['STATE_NAME'] != 'Alaska')
                      &(us_map['STATE_NAME'] != 'Hawaii')]
us_map_Alaska = us_map[us_map['STATE_NAME'] == 'Alaska']
us_map_Hawaii = us_map[us_map['STATE_NAME'] == 'Hawaii']

#Merging map and data:
df_land = us_map_land.set_index('NAME').join(df.set_index('CHSI_County_Name'))
df_Alaska = us_map_Alaska.set_index('NAME').join(df.
    ↳set_index('CHSI_County_Name'))
df_Hawaii = us_map_Hawaii.set_index('NAME').join(df.
    ↳set_index('CHSI_County_Name'))
```

```
[18]: def min_max_col(variable):
    minimum = min(df_land[variable].min(),
                  df_Alaska[variable].min(), df_Hawaii[variable].min())
    maximum = max(df_land[variable].max(),
                  df_Alaska[variable].max(), df_Hawaii[variable].max())
    return minimum, maximum

def US_plot(variable, color='Blues'):
    h, w = figaspect(1.)
    mn, mx = min_max_col(variable)
    fig, ax1 = plt.subplots(1, figsize=(w*10, h*10))
    ax1.axis('off')
    ax1.set_title(description(variable), fontsize=40)
    divider = make_axes_locatable(ax1)
    cax = divider.append_axes('right', size='2%', pad=0.1)
    cax.tick_params(labelsize=25)
    df_land.plot(column=variable, cmap=color,
                  linewidth=0.8, ax=ax1,
                  edgecolor='0.8', vmin=mn, vmax=mx,
                  legend=True, cax=cax)
```

```

left, bottom, width, height = [0.15, 0.22, 0.24, 0.3]
ax2 = fig.add_axes([left, bottom, width, height])
ax2.axis('off')
ax2.set_title('Alaska', fontsize=25)
df_Alaska.plot(column=variable, cmap=color, vmin=mn ,vmax=mx,
               linewidth=0.8, ax=ax2, edgecolor='0.6')

left, bottom, width, height = [0.35, 0.3, 0.2, 0.05]
ax3 = fig.add_axes([left, bottom, width, height])
ax3.axis('off')
ax3.set_title('Hawaii', fontsize=25)
df_Hawaii.plot(column =variable, cmap=color, vmin=mn ,vmax=mx,
               linewidth=0.8, ax=ax3, edgecolor='0.6')

```

```
[19]: list(description(i) for i in df.columns)
```

```
[19]: ['Two-digit state identifier, developed by the National Bureau of Standards',
      'Three-digit county identifier, developed by the National Bureau of Standards',
      'Name of county',
      'Name of State or District of Columbia',
      'Two-character postal abbreviation for state name',
      'CHSI Peer County Stratum Number',
      'Listing of strata factors',
      'Number of peer counties',
      'County data, population size',
      'County data, population density (people per square mile)',
      'County data, individuals living below poverty level',
      'County data, population under age 19',
      'County data, population age 19-64',
      'County data, population age 65-84',
      'County data, population age 85+',
      'County data, White',
      'County data, Black',
      'County data, American Indian',
      'County data, Asian/Pacific Islander',
      'County data, Hispanic origin',
      'County data, under age 1, complications of pregnancy/birth, White',
      'County data, under age 1, complications of pregnancy/birth, Black',
      'County data, under age 1, complications of pregnancy/birth, other',
      'County data, under age 1, complications of pregnancy/birth, Hispanic',
      'County data, under age 1, birth defects, White',
      'County data, under age 1, birth defects, Black',
      'County data, under age 1, birth defects, other',
      'County data, under age 1, birth defects, Hispanic',
      'County data, ages 1-14, injuries, White',
      'County data, ages 1-14, injuries, Black',
      'County data, ages 1-14, injuries, other',
      'County data, ages 1-14, injuries, Hispanic',
```

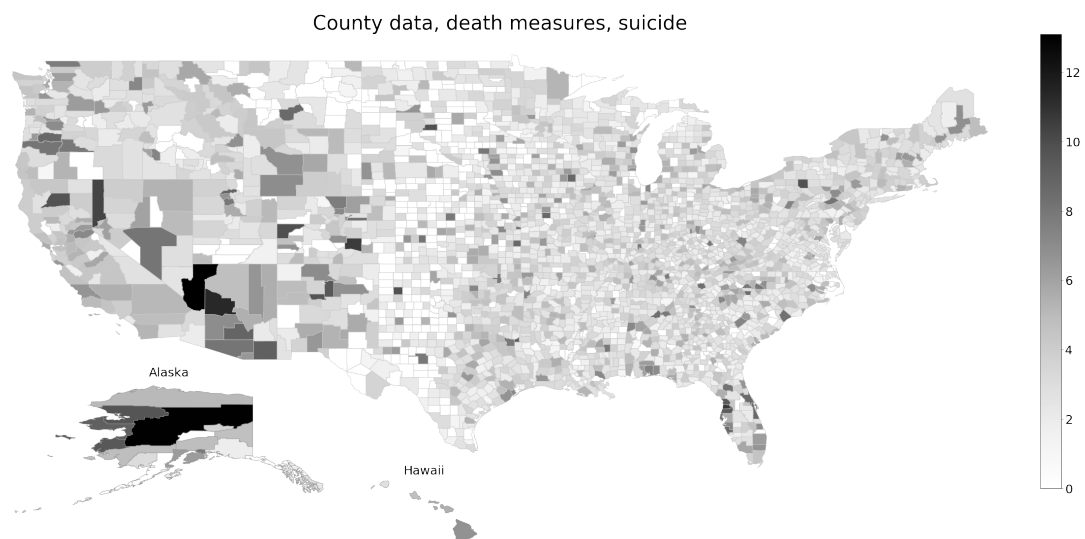
'County data, ages 1-14, cancer, White',
 'County data, ages 1-14, cancer, Black',
 'County data, ages 1-14, cancer, other',
 'County data, ages 1-14, cancer, Hispanic',
 'County data, ages 1-14, homicide, White',
 'County data, ages 1-14, homicide, Black',
 'County data, ages 1-14, homicide, other',
 'County data, ages 1-14, homicide, Hispanic',
 'County data, ages 15-24, injuries, White',
 'County data, ages 15-24, injuries, Black',
 'County data, ages 15-24, injuries, other',
 'County data, ages 15-24, injuries, Hispanic',
 'County data, ages 15-24, homicide, White',
 'County data, ages 15-24, homicide, Black',
 'County data, ages 15-24, homicide, other',
 'County data, ages 15-24, homicide, Hispanic',
 'County data, ages 15-24, suicide, White',
 'County data, ages 15-24, suicide, Black',
 'County data, ages 15-24, suicide, other',
 'County data, ages 15-24, suicide, Hispanic',
 'County data, ages 15-24, cancer, White',
 'County data, ages 15-24, cancer, Black',
 'County data, ages 15-24, cancer, other',
 'County data, ages 15-24, cancer, Hispanic',
 'County data, ages 25-44, injuries, White',
 'County data, ages 25-44, injuries, Black',
 'County data, ages 25-44, injuries, other',
 'County data, ages 25-44, injuries, Hispanic',
 'County data, ages 25-44, cancer, White',
 'County data, ages 25-44, cancer, Black',
 'County data, ages 25-44, cancer, other',
 'County data, ages 25-44, cancer, Hispanic',
 'County data, ages 25-44, heart disease, White',
 'County data, ages 25-44, heart disease, Black',
 'County data, ages 25-44, heart disease, other',
 'County data, ages 25-44, heart disease, Hispanic',
 'County data, ages 25-44, suicide, White',
 'County data, ages 25-44, suicide, Black',
 'County data, ages 25-44, suicide, other',
 'County data, ages 25-44, suicide, Hispanic',
 'County data, ages 25-44, hiv/aids, White',
 'County data, ages 25-44, hiv/aids, Black',
 'County data, ages 25-44, hiv/aids, other',
 'County data, ages 25-44, hiv/aids, Hispanic',
 'County data, ages 25-44, homicide, White',
 'County data, ages 25-44, homicide, Black',
 'County data, ages 25-44, homicide, other',

'County data, ages 25-44, homicide, Hispanic',
 'County data, ages 45-64, cancer, White',
 'County data, ages 45-64, cancer, Black',
 'County data, ages 45-64, cancer, other',
 'County data, ages 45-64, cancer, Hispanic',
 'County data, ages 45-64, heart disease, White',
 'County data, ages 45-64, heart disease, Black',
 'County data, ages 45-64, heart disease, other',
 'County data, ages 45-64, heart disease, Hispanic',
 'County data, ages 65+, heart disease, White',
 'County data, ages 65+, heart disease, Black',
 'County data, ages 65+, heart disease, other',
 'County data, ages 65+, heart disease, Hispanic',
 'County data, ages 65+, cancer, White',
 'County data, ages 65+, cancer, Black',
 'County data, ages 65+, cancer, other',
 'County data, ages 65+, cancer, Hispanic',
 '',
 'County data, birth measures, low birth wt. (<2500 g)',
 'County data, birth measures, very low birth wt. (<1500 g)',
 'County data, birth measures, premature births (<37 weeks)',
 'County data, birth measures, births to women under 18',
 'County data, birth measures, births to women over 40',
 'County data, birth measures, births to unmarried women',
 'County data, birth measures, no care in first trimester',
 'County data, infant mortality',
 'County data, infant mortality, White non Hispanic',
 'County data, infant mortality, Black non Hispanic',
 'County data, infant mortality, Hispanic',
 'County data, infant mortality, neonatal',
 'County data, infant mortality, post-neonatal',
 'County data, death measures, breast cancer (female)',
 'County data, death measures, colon cancer',
 'County data, death measures, coronary heart disease',
 'County data, death measures, homicide',
 'County data, death measures, lung cancer',
 'County data, death measures, motor vehicle injuries',
 'County data, death measures, stroke',
 'County data, death measures, suicide',
 'County data, death measures, unintentional injury',
 'County data, total number of births',
 'County data, total number of deaths',
 'County data, Haemophilus Influenzae B reported cases',
 'County data, Haemophilus Influenzae B expected cases',
 'County data, Hepatitis A reported cases',
 'County data, Hepatitis A expected cases',
 'County data, Hepatitis B reported cases',

'County data, Hepatitis B expected cases',
 'County data, Measles reported cases',
 'County data, Measles expected cases',
 'County data, Pertussis reported cases',
 'County data, Pertussis expected cases',
 'County data, Congenital Rubella Syndrome reported cases',
 'County data, Syphilis reported cases',
 'County data, Syphilis expected cases',
 'County data, pap smears (18+)',
 'County data, mammography (50+)',
 'County data, sigmoidoscopy (50+)',
 'County data, pneumonia vaccine (65+)',
 'County data, flu vaccine (65+)',
 'Relative health indicator, low birth wt. (<2500 g)',
 'Relative health indicator, very low birth wt. (<1500 g)',
 'Relative health indicator, premature births (<37 weeks)',
 'Relative health indicator, births to women under 18',
 'Relative health indicator, births to women over 40',
 'Relative health indicator, births to unmarried women',
 'Relative health indicator, no care in first trimester',
 'Relative health indicator, infant mortality',
 'Relative health indicator, White non Hispanic infant mortality',
 'Relative health indicator, Black non Hispanic infant mortality',
 'Relative health indicator, Hispanic infant mortality',
 'Relative health indicator, neonatal infant mortality',
 'Relative health indicator, post-neonatal infant mortality',
 'Relative health indicator, breast cancer (female)',
 'Relative health indicator, colon cancer',
 'Relative health indicator, coronary heart disease',
 'Relative health indicator, homicide',
 'Relative health indicator, lung cancer',
 'Relative health indicator, motor vehicle injuries',
 'Relative health indicator, stroke',
 'Relative health indicator, suicide',
 'Relative health indicator, unintentional injury',
 'County data, no exercise',
 'County data, few fruits/vegetables',
 'County data, obesity',
 'County data, high blood pressure',
 'County data, smoker',
 'County data, diabetes',
 'County data, uninsured individuals',
 'County data, medicare beneficiaries, elderly (age 65+)',
 'County data, medicare beneficiaries, disabled',
 'County data, primary care physicians per 100,000 pop.',
 'County data, dentists per 100,000 pop.',
 'Indicator for any Community/Migrant Health Centers located in the county',

```
'Indicator for single county designated Health Professional Shortage Area',
'County data, average life expectancy',
'Medium for all U.S. counties, average life expectancy',
'County data, all causes of death',
'Medium for all U.S. counties, all causes of death',
'County data, self-rated health status',
'County data, average number of unhealthy days in past month',
'County data, no high school diploma (among adults age 25 and older)',
'County data, unemployed',
'County data, severely work disabled',
'County data, major depression',
'County data, recent drug users (within past month)',
'County data, E.coli reported cases',
'County data, E.coli expected cases',
'County data, Salmonella reported cases',
'County data, Salmonella expected cases',
'County data, Shigella reported cases',
'County data, Shigella expected cases',
'County data, toxic chemicals released annually',
'Air quality standard indicator, carbon monoxide',
'Air quality standard indicator, nitrogen dioxide',
'Air quality standard indicator, sulfur dioxide',
'Air quality standard indicator, ozone',
'Air quality standard indicator, particulate matter',
'Air quality standard indicator, lead']
```

```
[20]: US_plot('Suicide', 'Greys')
```



```
[21]: US_plot('CHD', 'Reds')
```

County data, death measures, coronary heart disease

