

# EDA of GIS Final Project

## US Map at State Level

### Data Setup

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

```
In [ ]: # County Level Data  
# Link: https://data.cdc.gov/500-Cities-Places/PLACES-County-Data-GIS-Friendly-Format-2023-release/i46a-9kgh/about_data  
df = pd.read_csv("C:/Users/user/Desktop/GIS/Final Project/PLACES_County_Data_GIS_Friendly_Format__2023_release_2024050
```

```
In [ ]: df.head(5)
```

```
Out[ ]:   StateAbbr StateDesc CountyName CountyFIPS TotalPopulation ACCESS2_CrudePrev ACCESS2_Crude95CI ACCESS2_AdjPrev ACCESS2_Adj95CI  
  
0       AL    Alabama     Autauga      1001        59095          10.0      ( 7.7, 12.6)         10.4      ( 8.0, 13.0  
  
1       AL    Alabama     Bullock      1011        10320          18.7      (15.3, 23.0)         19.2      (15.7, 23.0  
  
2       AL    Alabama     Chilton      1021        45274          13.5      (10.8, 16.9)         14.1      (11.2, 17.0  
  
3       AL    Alabama     Cleburne      1029        15103          11.8      ( 8.9, 15.1)         12.5      ( 9.4, 16.0  
  
4       AL    Alabama     DeKalb      1049        71813          15.9      (12.5, 20.1)         16.7      (13.2, 21.0
```

5 rows × 154 columns

```
In [ ]: df[df['CountyName']=='Cook']['ACCESS2_CrudePrev']
```

```
Out[ ]: 629      5.3  
1033     18.9  
1722     10.7  
Name: ACCESS2_CrudePrev, dtype: float64
```

### Import USA Shapefile

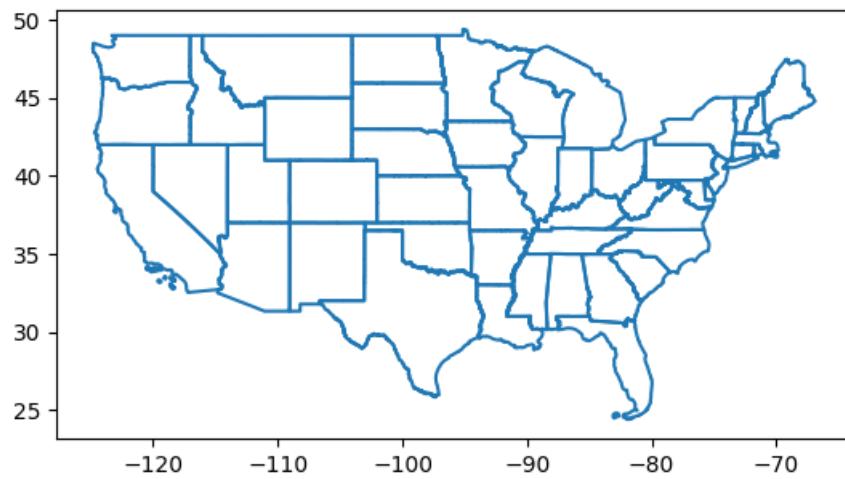
```
In [ ]: import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
import contextily as ctx  
import geopandas as gpd  
import os  
from mpl_toolkits.axes_grid1 import make_axes_locatable  
  
path = "C:/Users/user/Desktop/GIS/Final Project/tl_2023_us_state/tl_2023_us_state.shp"  
usa = gpd.read_file(path)  
usa = usa.to_crs("EPSG:4326")
```

```
In [ ]: usa.iloc[0,:]
```

```
Out[ ]: REGION          3
DIVISION        5
STATEFP         54
STATENS      01779805
GEOID           54
GEOIDFQ      0400000US54
STUSPS          WV
NAME          West Virginia
LSAD            00
MTFCC          G4000
FUNCSTAT          A
ALAND        62266499712
AWATER        489003081
INTPTLAT      +38.6472854
INTPTLON      -080.6183274
geometry    POLYGON ((-77.754376 39.333461, -77.754219 39...
Name: 0, dtype: object
```

```
In [ ]: # Exclude non-continental states
non_continental = ['HI', 'VI', 'MP', 'GU', 'AK', 'AS', 'PR']
for n in non_continental:
    usa = usa[usa.STUSPS != n]
```

```
In [ ]: usa.boundary.plot()
plt.show()
```



## Define Plotting Function

```
In [ ]: # Define function to plot states
def StatesPlot(df, data, cmap):
    f, ax = plt.subplots(1, 1, figsize=(15, 10), sharex=True, sharey=True, dpi=300)
    f.tight_layout()
    plt.title('United States Map - Variable = ' + data)
    ax.set_axis_off()
    divider = make_axes_locatable(ax)
    cax = divider.append_axes("right", size="3%", pad=0.5, alpha=0.5)
    df.plot(data, ax=ax, alpha=0.5, cmap=cmap, edgecolor='k', legend=True, cax=cax, linewidth=0.5)
    plt.ylabel('Prevalence Rate', fontsize=12)
    plt.show()
```

## Aggregate to State-Level

```
In [ ]: # Filter columns that end with '_CrudePrev'
columns_to_group = [col for col in df.columns if col.endswith('_CrudePrev')] + [col for col in df.columns if col.endswith('_CrudeRate')]

# Group by StateAbbr and aggregate the columns using mean()
state_df = df.groupby('StateAbbr')[columns_to_group].mean().reset_index()
```

```
In [ ]: state_df.head(5)
```

Out[ ]:

	StateAbbr	ACCESS2_CrudePrev	ARTHRITIS_CrudePrev	BINGE_CrudePrev	BPHIGH_CrudePrev	BPMED_CrudePrev	CANCER_CrudePrev	CASTH1
0	AK	12.523333	24.233333	17.650000	32.600000	70.293333	6.65667	
1	AL	12.614925	33.885075	13.043284	44.368657	82.298507	7.634328	
2	AR	11.332000	32.617333	12.688000	43.657333	80.389333	7.937333	
3	AZ	15.380000	27.446667	15.166667	33.853333	76.600000	7.920000	
4	CA	10.551724	23.967241	15.237931	31.218966	74.450000	7.081034	

5 rows × 76 columns

In [ ]:

```
print("Columns in usa GeoDataFrame:", usa.columns)
print("Columns in state_df DataFrame:", state_df.columns)
```

Columns in usa GeoDataFrame: Index(['REGION', 'DIVISION', 'STATEFP', 'STATENS', 'GEOID', 'GEOIDFQ', 'STUSPS', 'NAME', 'LSAD', 'MTFCC', 'FUNCSTAT', 'ALAND', 'AWATER', 'INTPTLAT', 'INTPTLON', 'geometry'],  
dtype='object')

Columns in state\_df DataFrame: Index(['StateAbbr', 'ACCESS2\_CrudePrev', 'ARTHRITIS\_CrudePrev', 'BINGE\_CrudePrev', 'BPHIGH\_CrudePrev', 'BPMED\_CrudePrev', 'CANCER\_CrudePrev', 'CASTHMA\_CrudePrev', 'CERVICAL\_CrudePrev', 'CHD\_CrudePrev', 'CHECKUP\_CrudePrev', 'CHOLSCREEN\_CrudePrev', 'COLON\_SCREEN\_CrudePrev', 'COPD\_CrudePrev', 'COREM\_CrudePrev', 'COREW\_CrudePrev', 'CSMOKING\_CrudePrev', 'DENTAL\_CrudePrev', 'DEPRESSION\_CrudePrev', 'DIABETES\_CrudePrev', 'GHLTH\_CrudePrev', 'HIGHCHOL\_CrudePrev', 'KIDNEY\_CrudePrev', 'LPA\_CrudePrev', 'MAMMOUSE\_CrudePrev', 'MHLTH\_CrudePrev', 'OBESITY\_CrudePrev', 'PHLTH\_CrudePrev', 'SLEEP\_CrudePrev', 'STROKE\_CrudePrev', 'TEETHLOST\_CrudePrev', 'HEARING\_CrudePrev', 'VISION\_CrudePrev', 'COGNITION\_CrudePrev', 'MOBILITY\_CrudePrev', 'SELCARE\_CrudePrev', 'INDEPLIVE\_CrudePrev', 'DISABILITY\_CrudePrev', 'ACCESS2\_AdjPrev', 'ARTHRITIS\_AdjPrev', 'BINGE\_AdjPrev', 'BPHIGH\_AdjPrev', 'BPMED\_AdjPrev', 'CANCER\_AdjPrev', 'CASTHMA\_AdjPrev', 'CERVICAL\_AdjPrev', 'CHD\_AdjPrev', 'CHECKUP\_AdjPrev', 'CHOLSCREEN\_AdjPrev', 'COLON\_SCREEN\_AdjPrev', 'COPD\_AdjPrev', 'COREM\_AdjPrev', 'COREW\_AdjPrev', 'CSMOKING\_AdjPrev', 'DENTAL\_AdjPrev', 'DEPRESSION\_AdjPrev', 'DIABETES\_AdjPrev', 'GHLTH\_AdjPrev', 'HIGHCHOL\_AdjPrev', 'KIDNEY\_AdjPrev', 'LPA\_AdjPrev', 'MAMMOUSE\_AdjPrev', 'MHLTH\_AdjPrev', 'OBESITY\_AdjPrev', 'PHLTH\_AdjPrev', 'SLEEP\_AdjPrev', 'STROKE\_AdjPrev', 'TEETHLOST\_AdjPrev', 'HEARING\_AdjPrev', 'VISION\_AdjPrev', 'COGNITION\_AdjPrev', 'MOBILITY\_AdjPrev', 'SELCARE\_AdjPrev', 'INDEPLIVE\_AdjPrev', 'DISABILITY\_AdjPrev', 'TotalPopulation'],  
dtype='object')

## Merge Shapefile with Selected Dataframe

In [ ]:

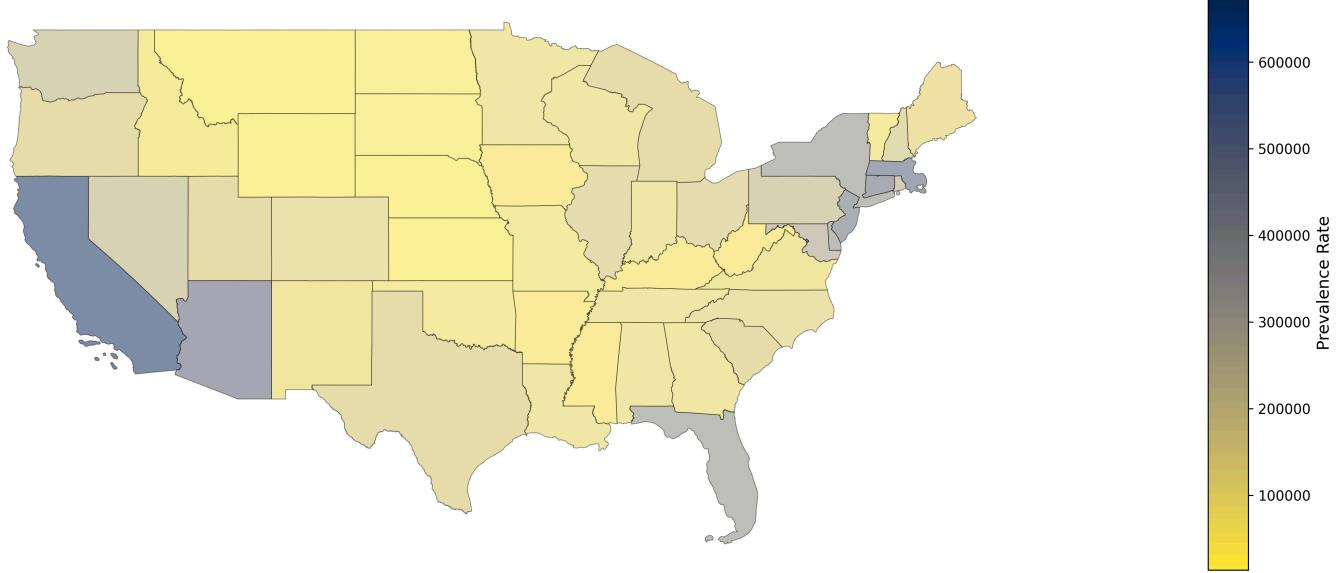
```
# Merge aggregated DataFrame with the GeoDataFrame
merged_gdf = usa.merge(state_df, left_on='STUSPS', right_on='StateAbbr', how='left')
```

## Population by State

In [ ]:

```
StatesPlot(merged_gdf, 'TotalPopulation', 'cividis_r')
```

United States Map - Variable = TotalPopulation



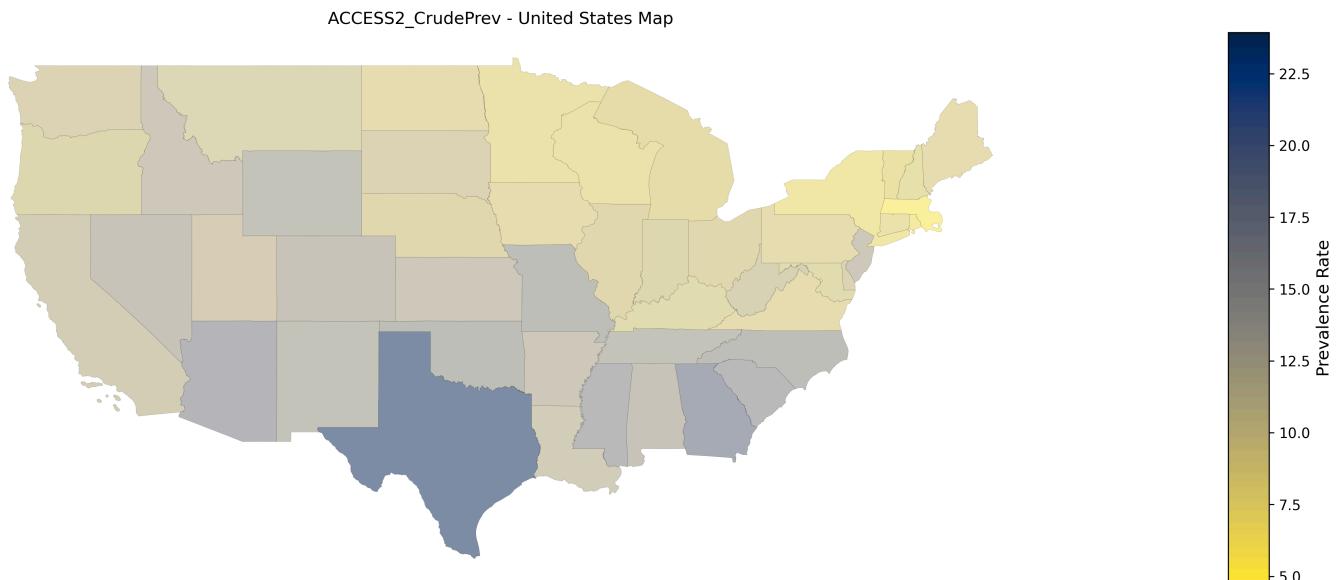
## Function to Plot Multiple Columns

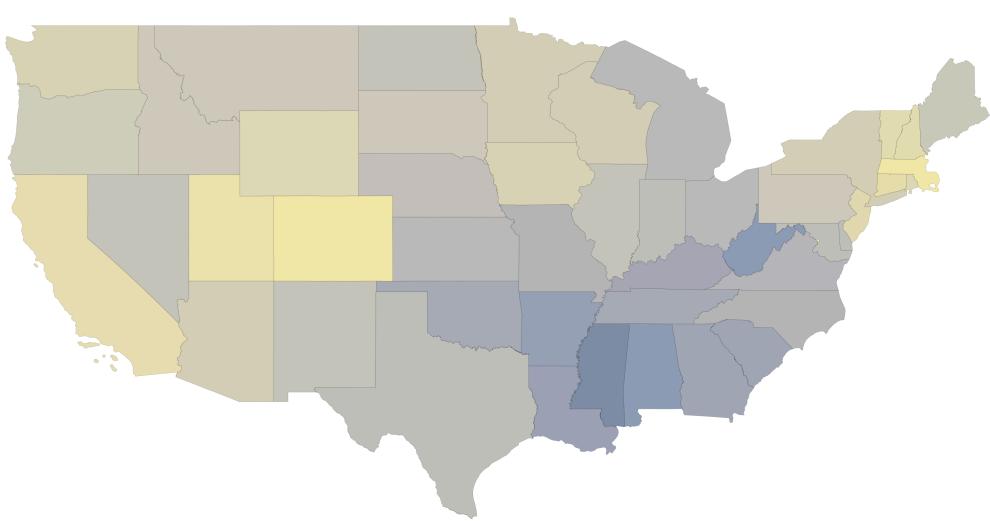
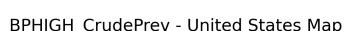
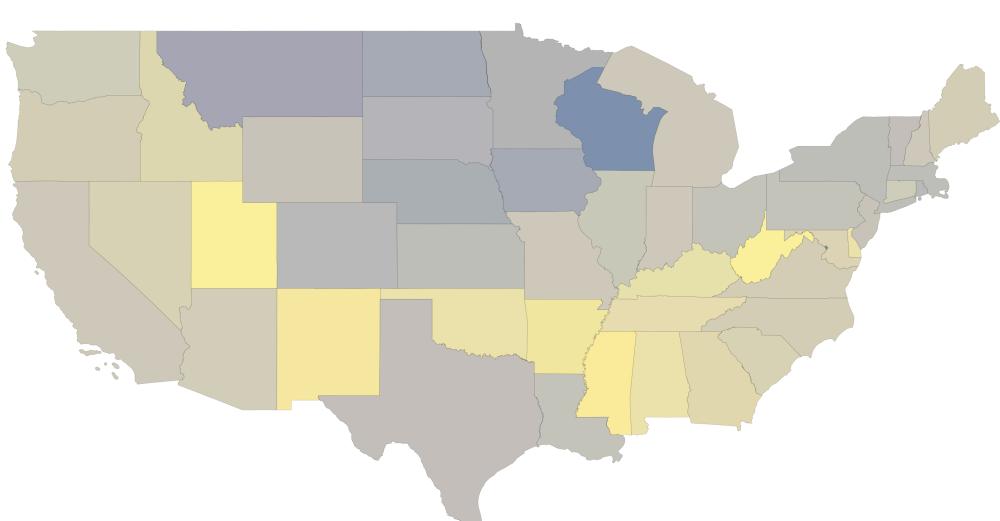
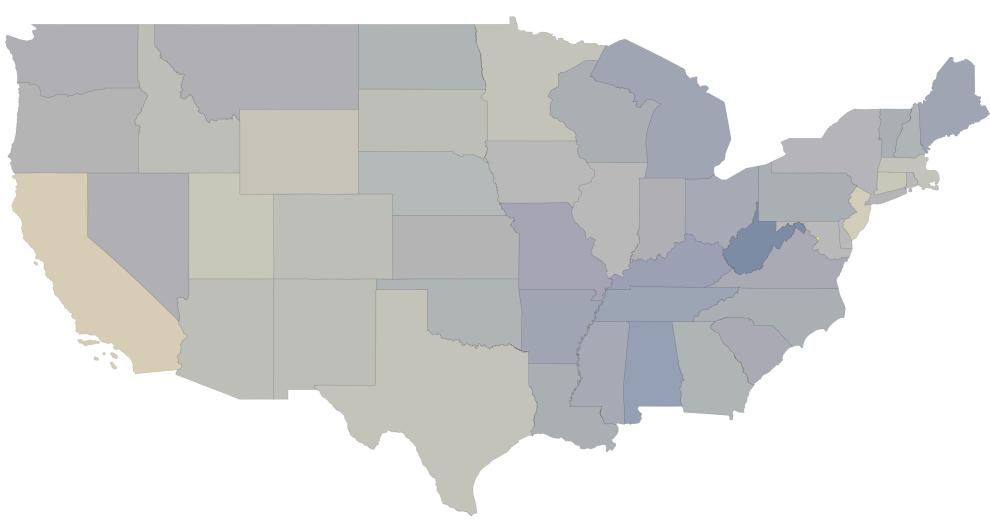
```
In [ ]: def plot_data_multiple_columns(geo_df, data_columns, colormap):
    for column in data_columns:
        # Plotting
        f, ax = plt.subplots(1, 1, figsize=(15, 10), sharex=True, sharey=True, dpi=300)
        f.tight_layout()
        plt.title(f'{column} - United States Map')
        ax.set_axis_off()
        divider = make_axes_locatable(ax)
        cax = divider.append_axes("right", size="3%", pad=0.5, alpha=0.5)
        geo_df.plot(column, ax=ax, alpha=0.5, cmap=colormap, edgecolor='k', legend=True, cax=cax, linewidth=0.1)
        plt.ylabel('Prevalence Rate', fontsize=12)
    plt.show()
```

## Use Crude Data

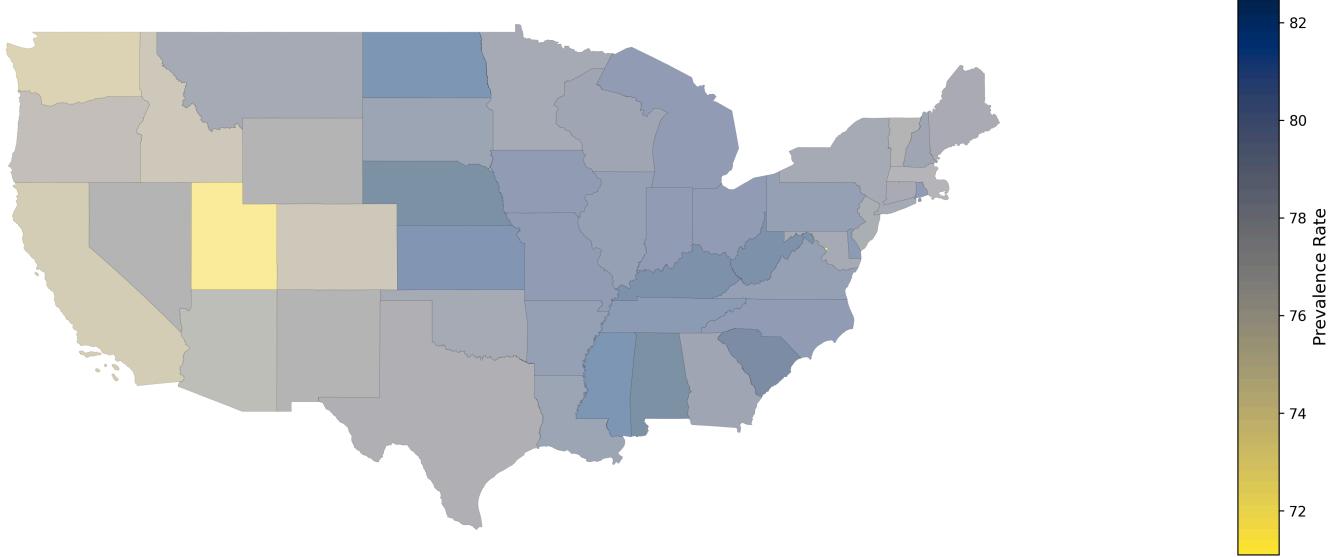
```
In [ ]: # Get columns ending with '_CrudePrev'
columns_to_plot = [col for col in df.columns if col.endswith('_CrudePrev')]

plot_data_multiple_columns(merged_gdf, columns_to_plot, 'cividis_r')
```

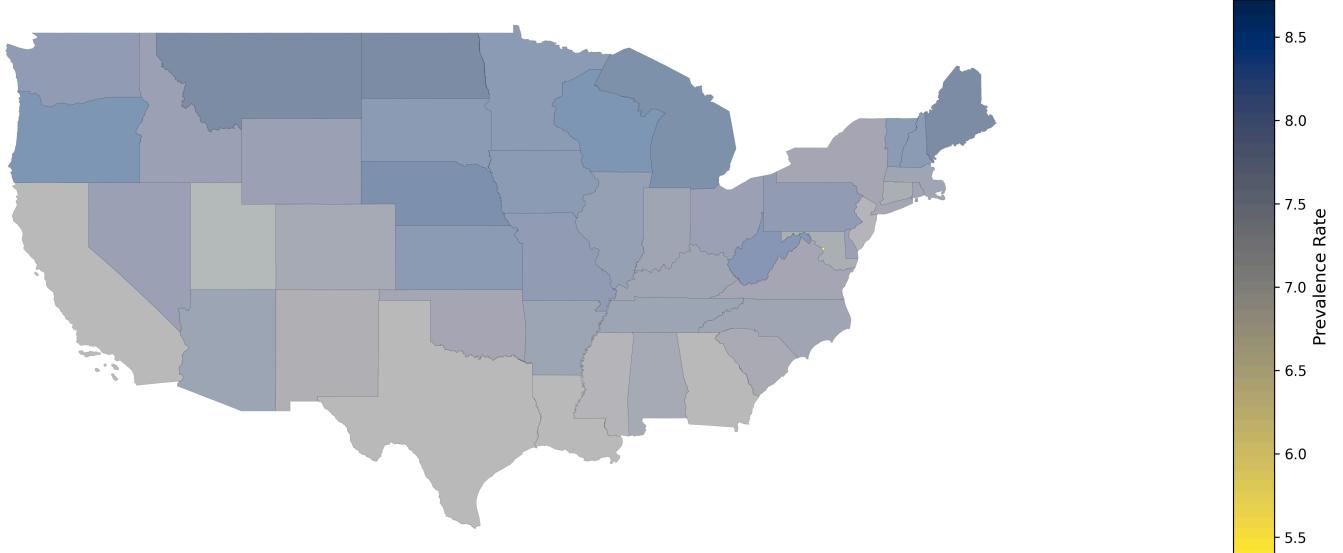




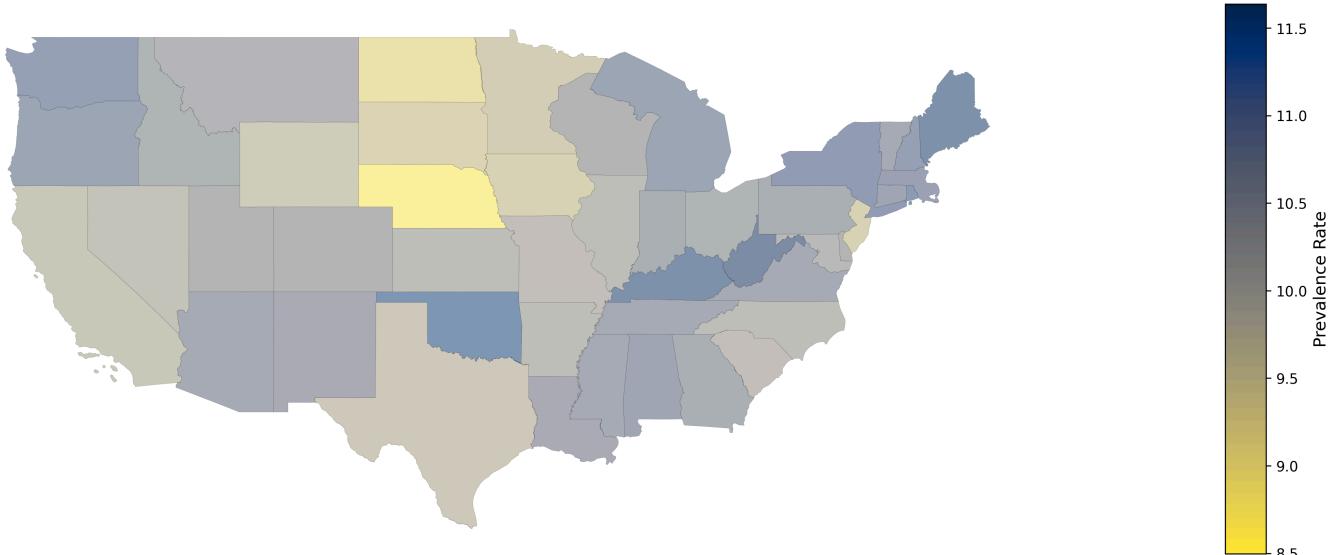
BPMED\_CrudePrev - United States Map



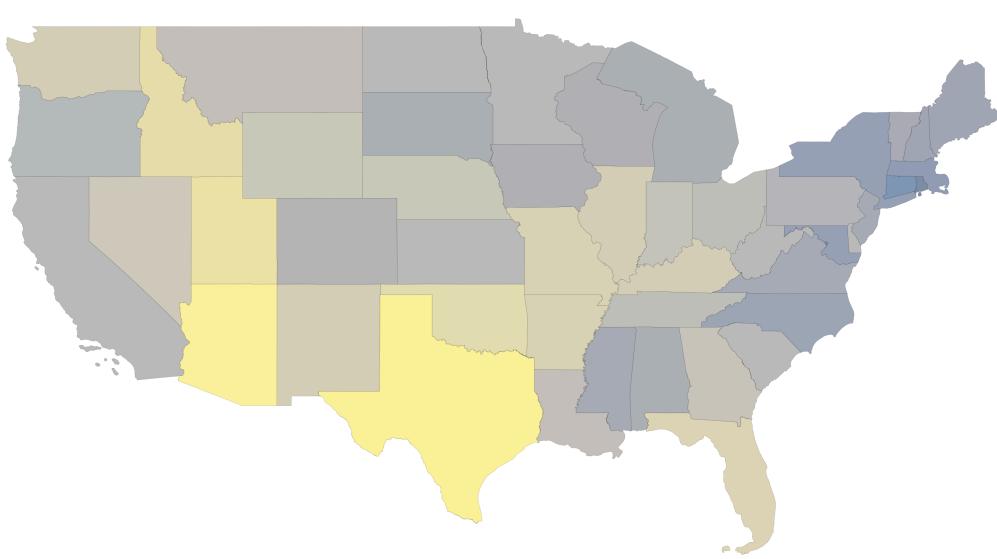
CANCER\_CrudePrev - United States Map



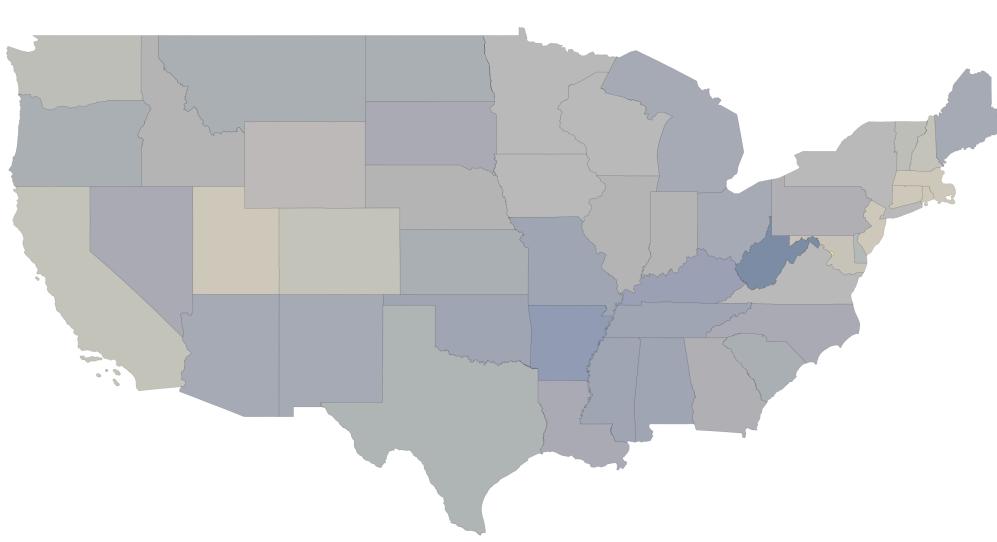
CASTHMA\_CrudePrev - United States Map



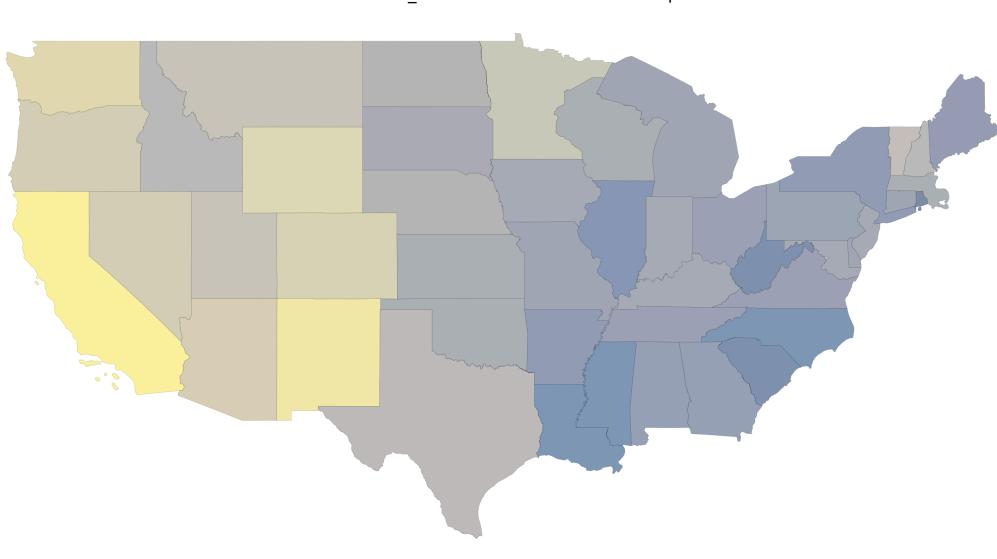
CERVICAL\_CrudePrev - United States Map



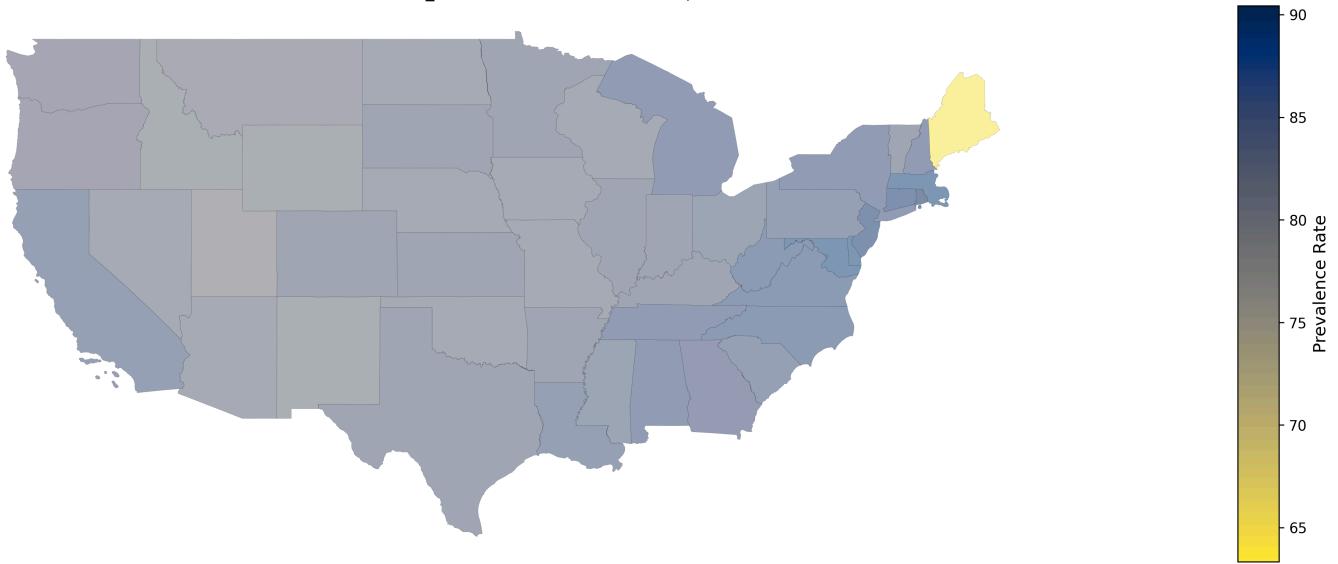
CHD\_CrudePrev - United States Map



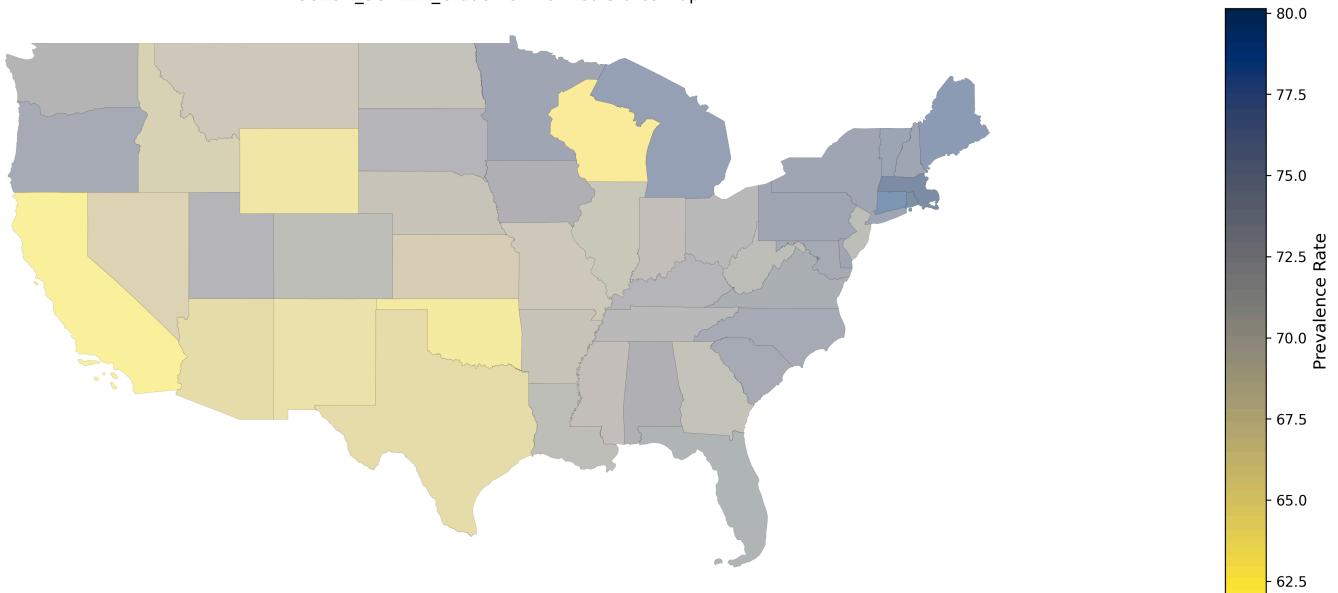
CHECKUP\_CrudePrev - United States Map



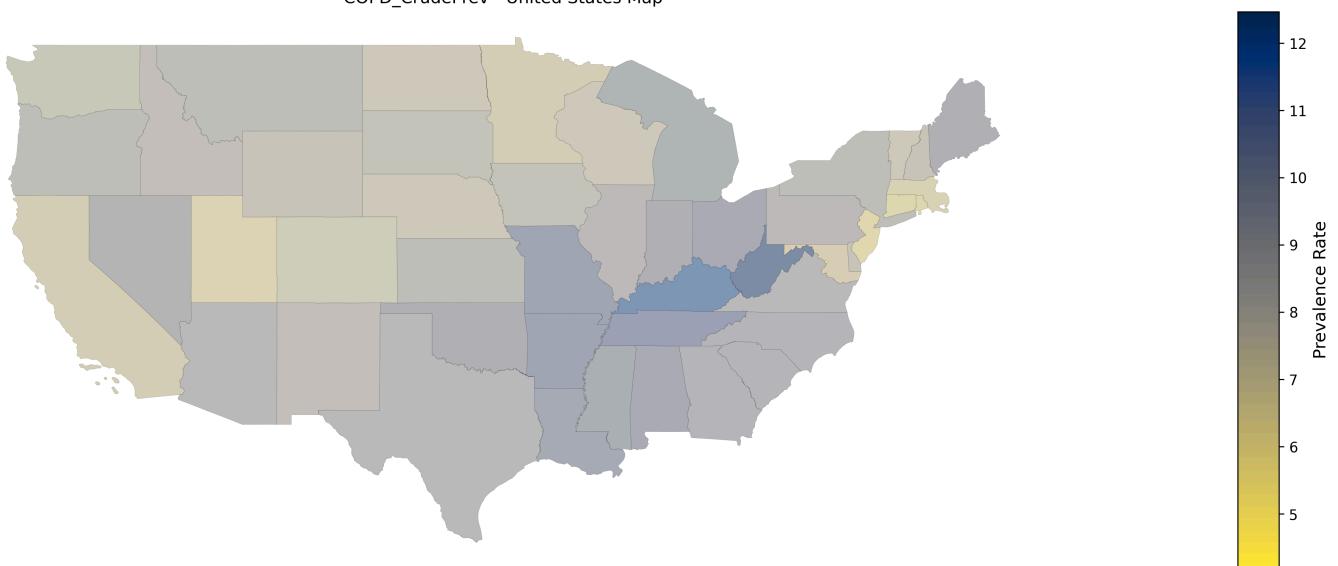
CHOLSCREEN\_CrudePrev - United States Map



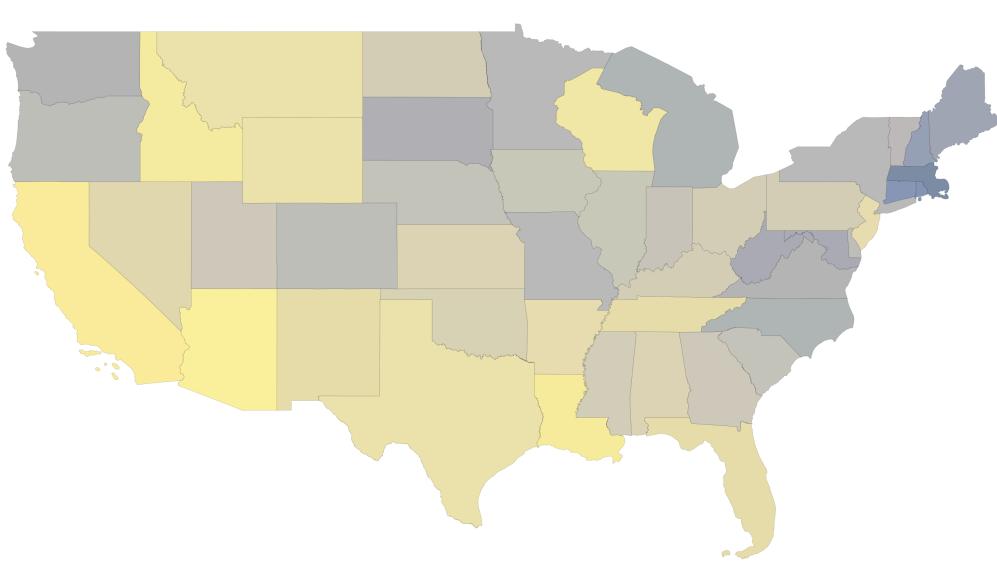
COLON\_SCREEN\_CrudePrev - United States Map



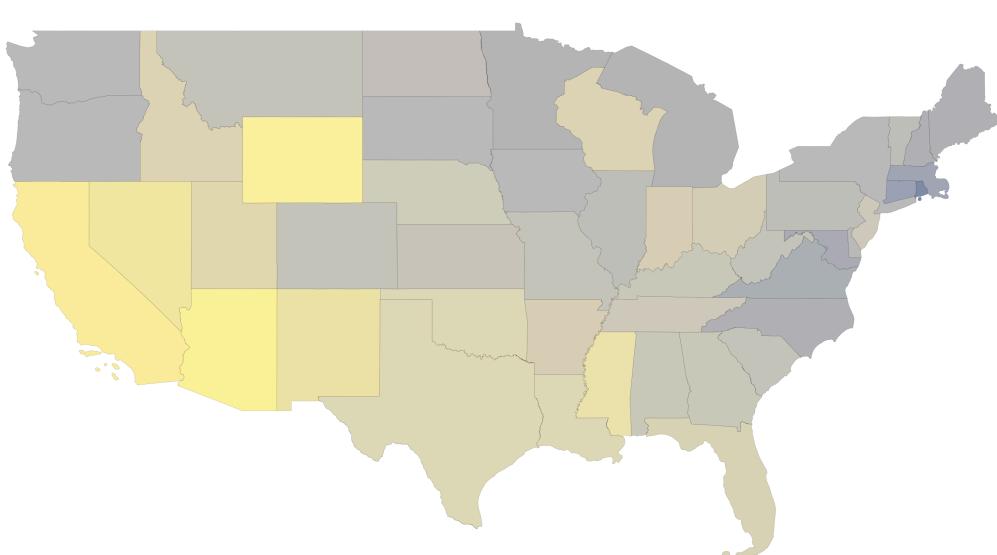
COPD\_CrudePrev - United States Map



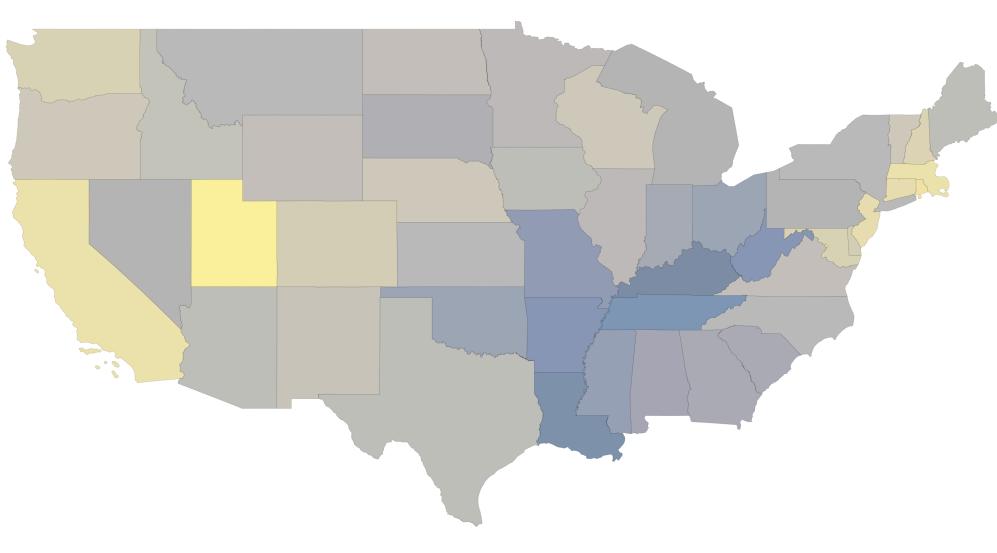
COREM\_CrudePrev - United States Map



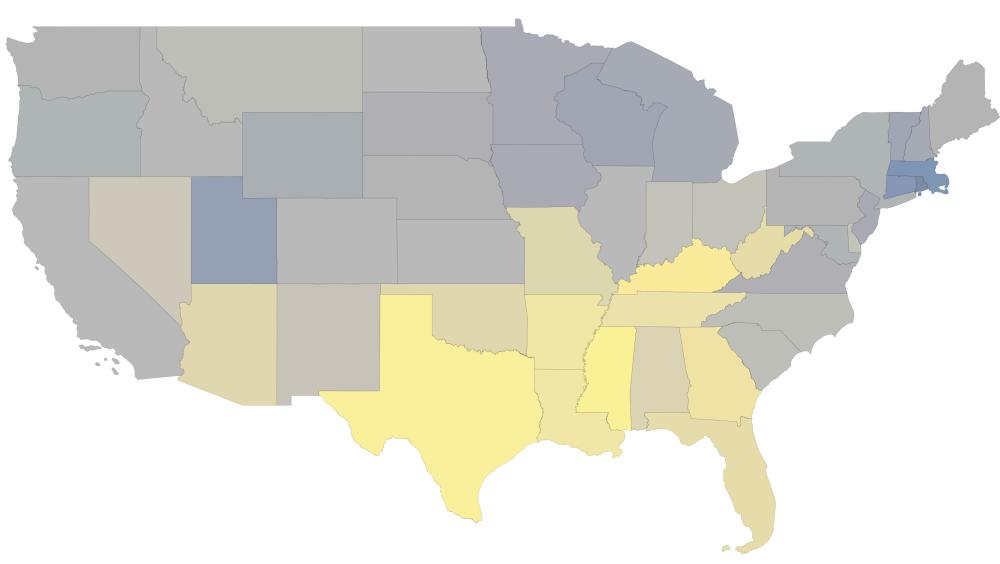
COREW\_CrudePrev - United States Map



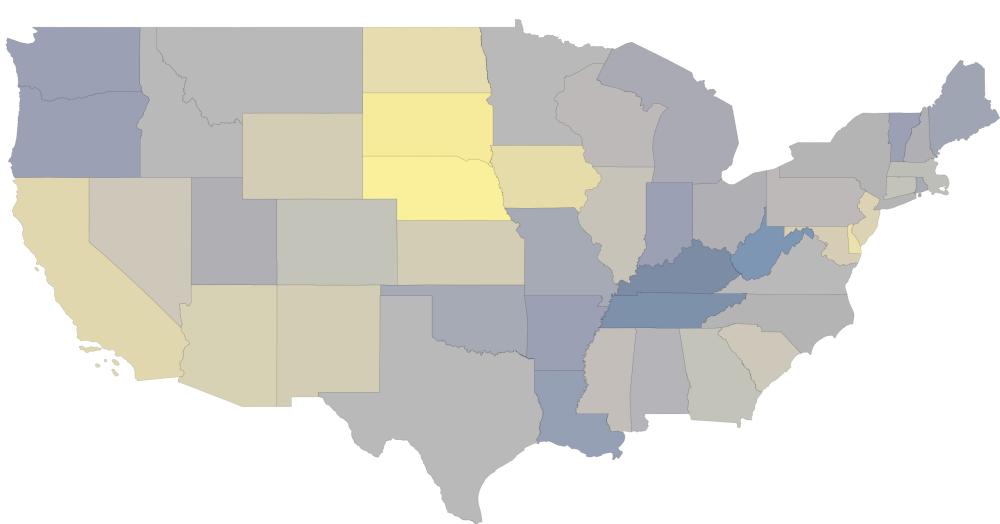
CSMOKING\_CrudePrev - United States Map



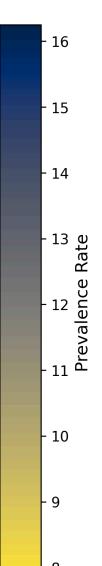
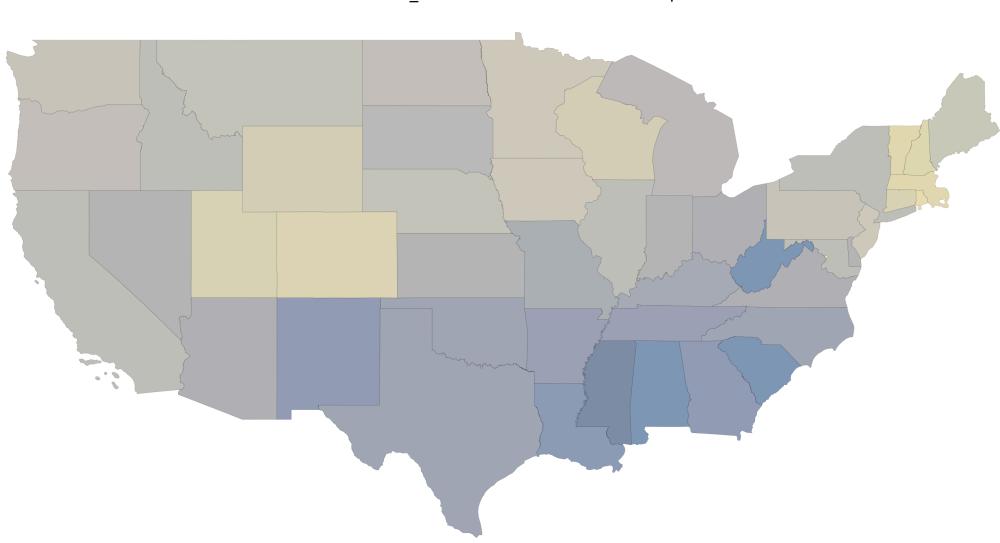
DENTAL\_CrudePrev - United States Map



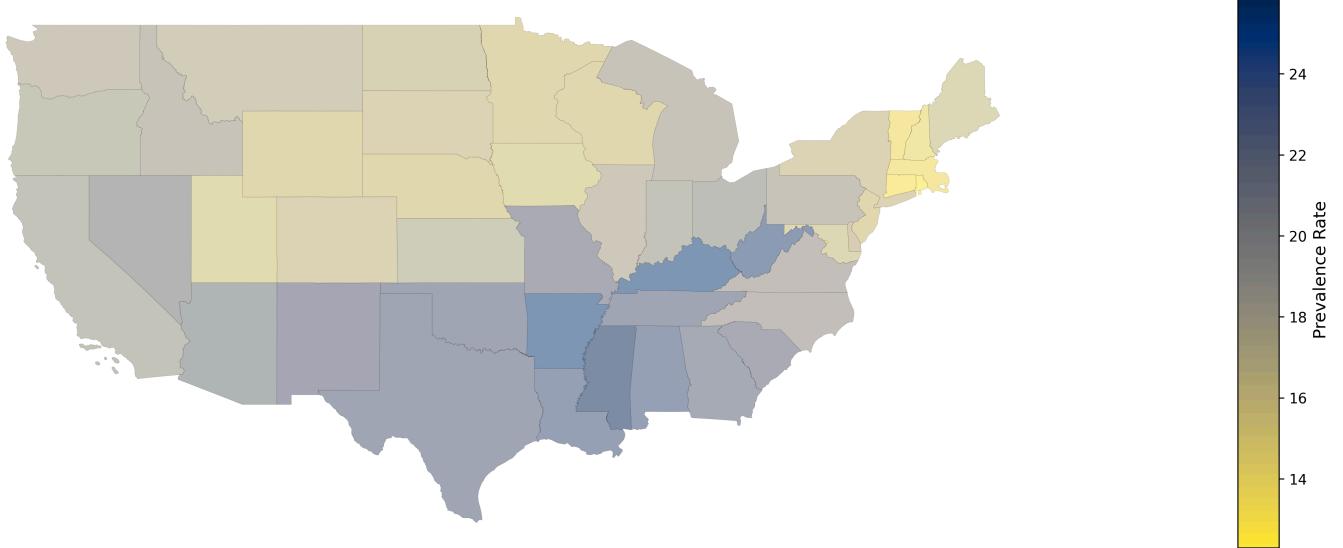
DEPRESSION\_CrudePrev - United States Map



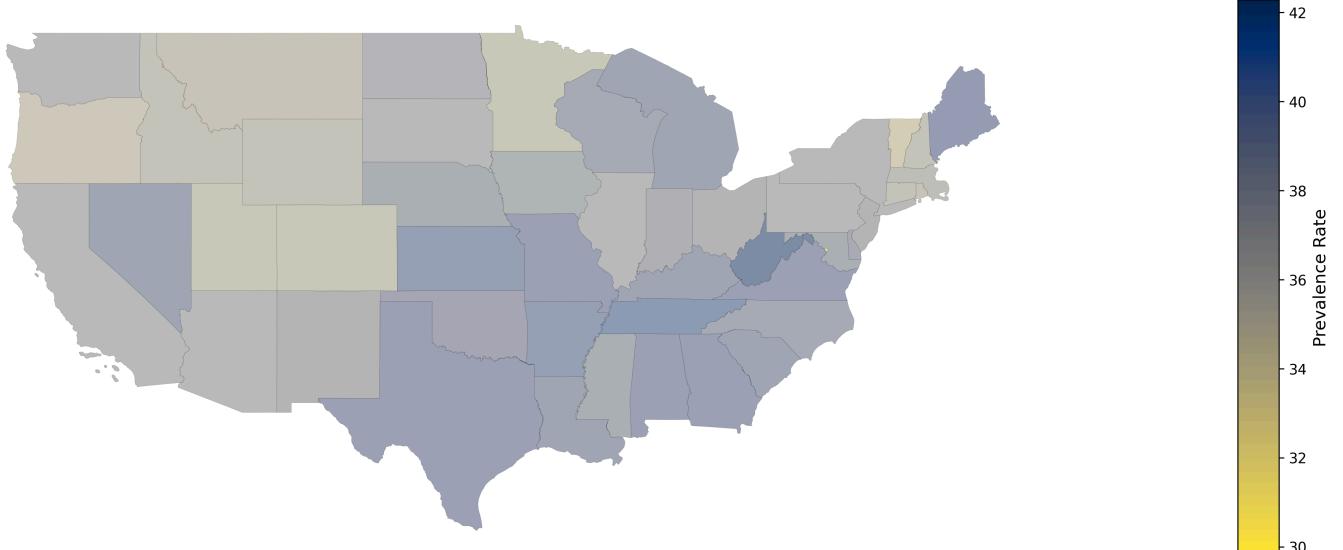
DIABETES\_CrudePrev - United States Map



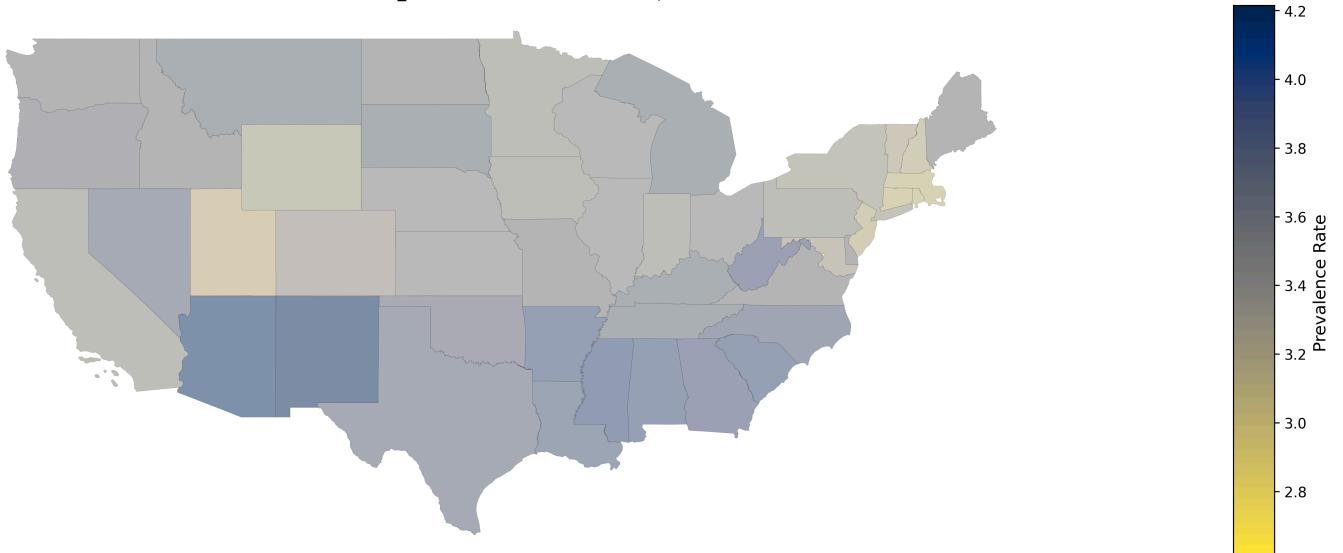
GHLTH\_CrudePrev - United States Map



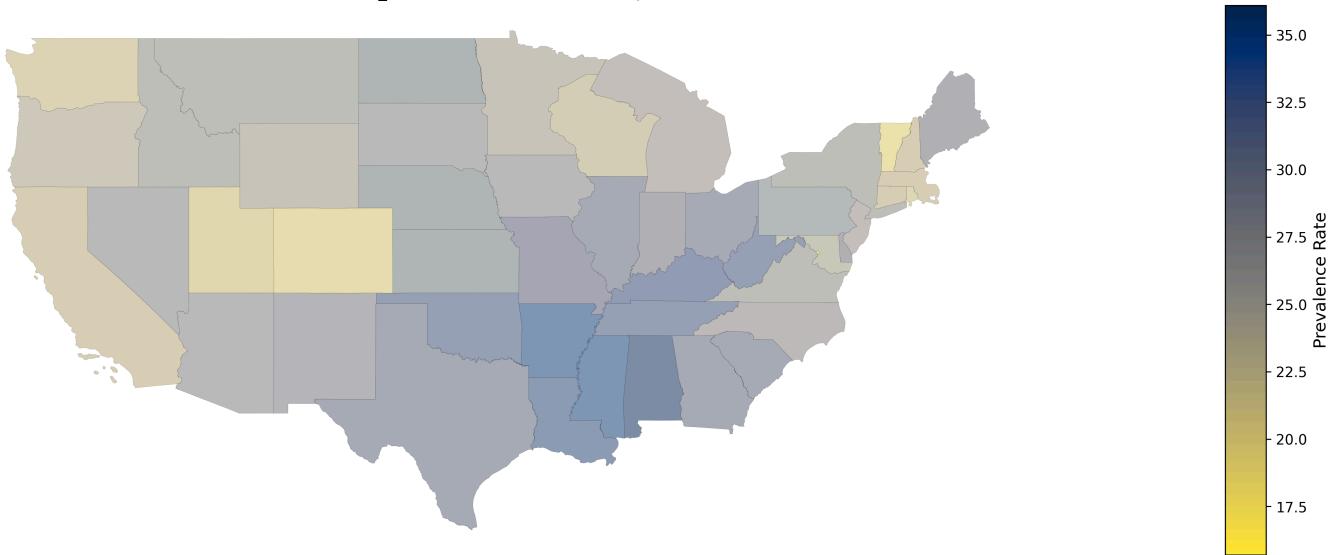
HIGHCHOL\_CrudePrev - United States Map



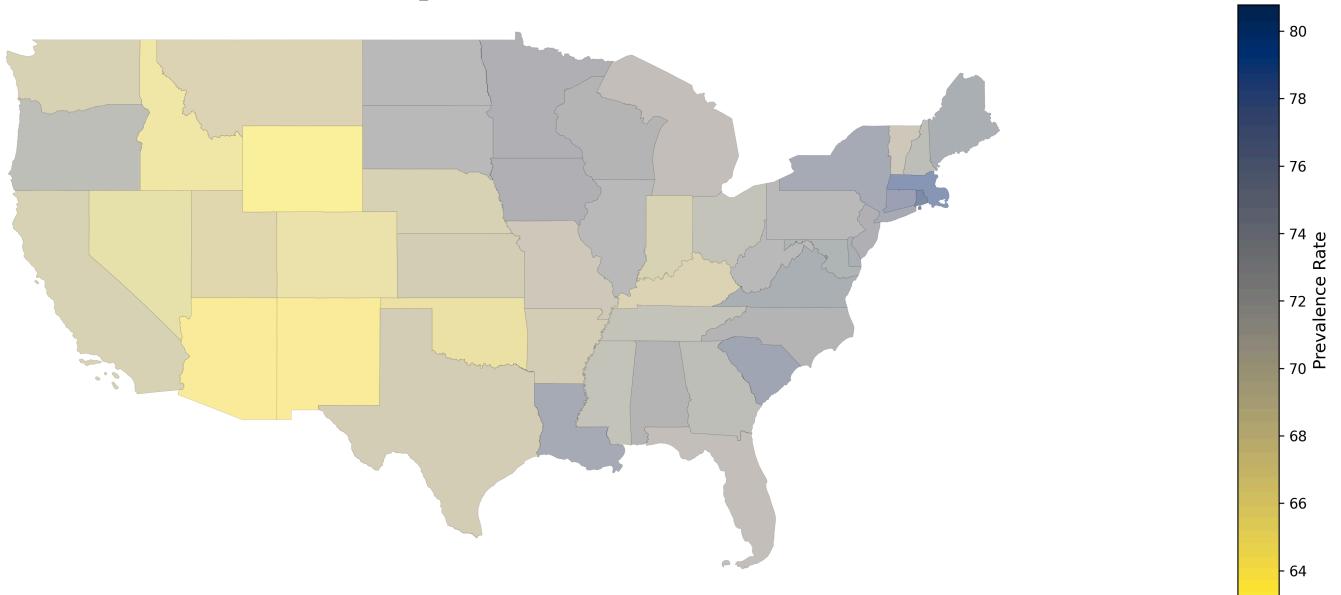
KIDNEY\_CrudePrev - United States Map



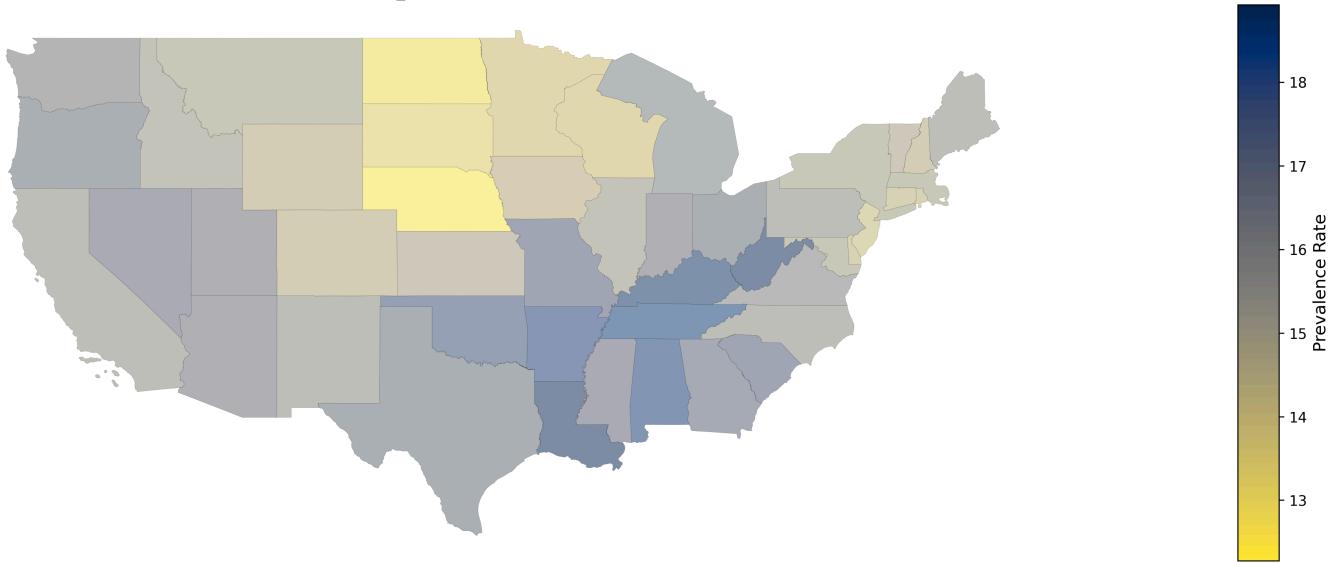
LPA\_CrudePrev - United States Map



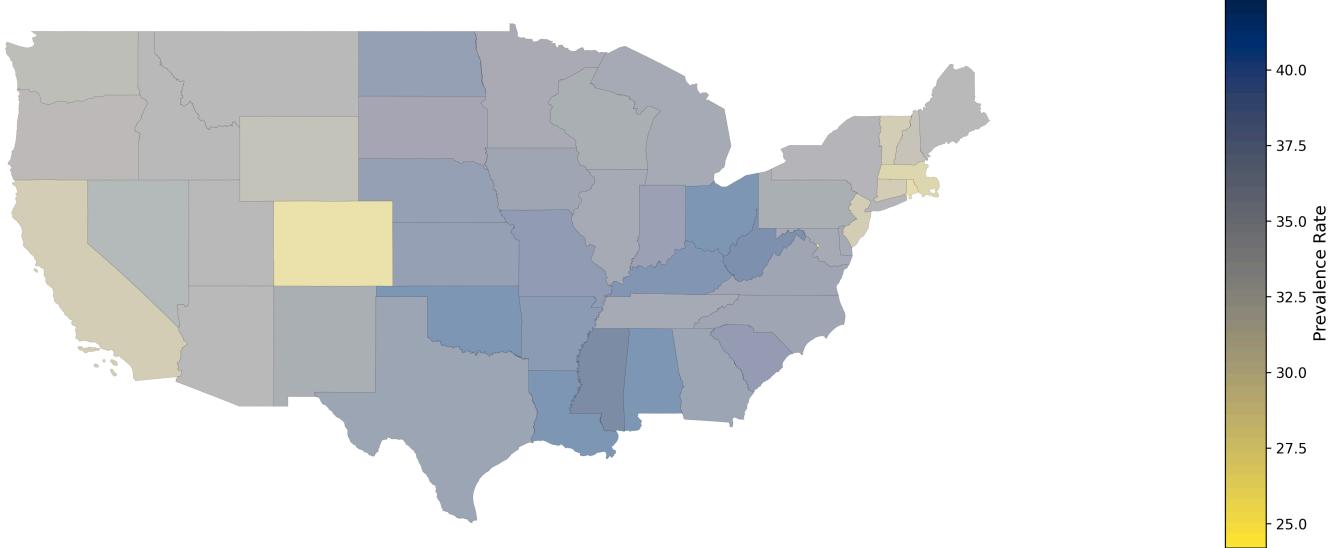
MAMMOUSE\_CrudePrev - United States Map



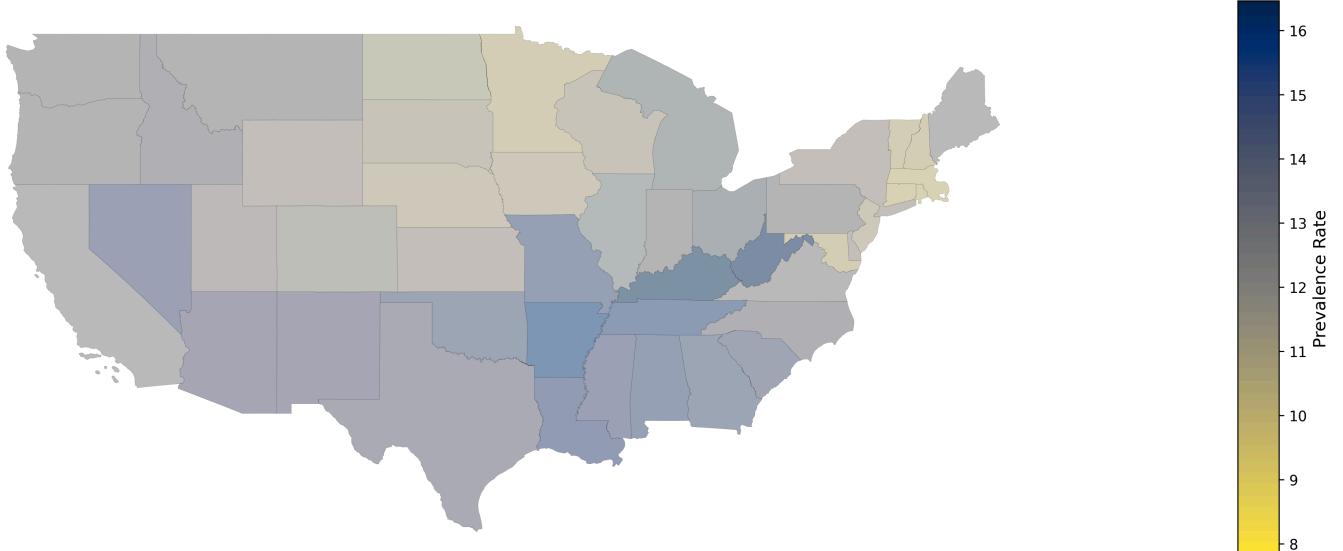
MHLTH\_CrudePrev - United States Map



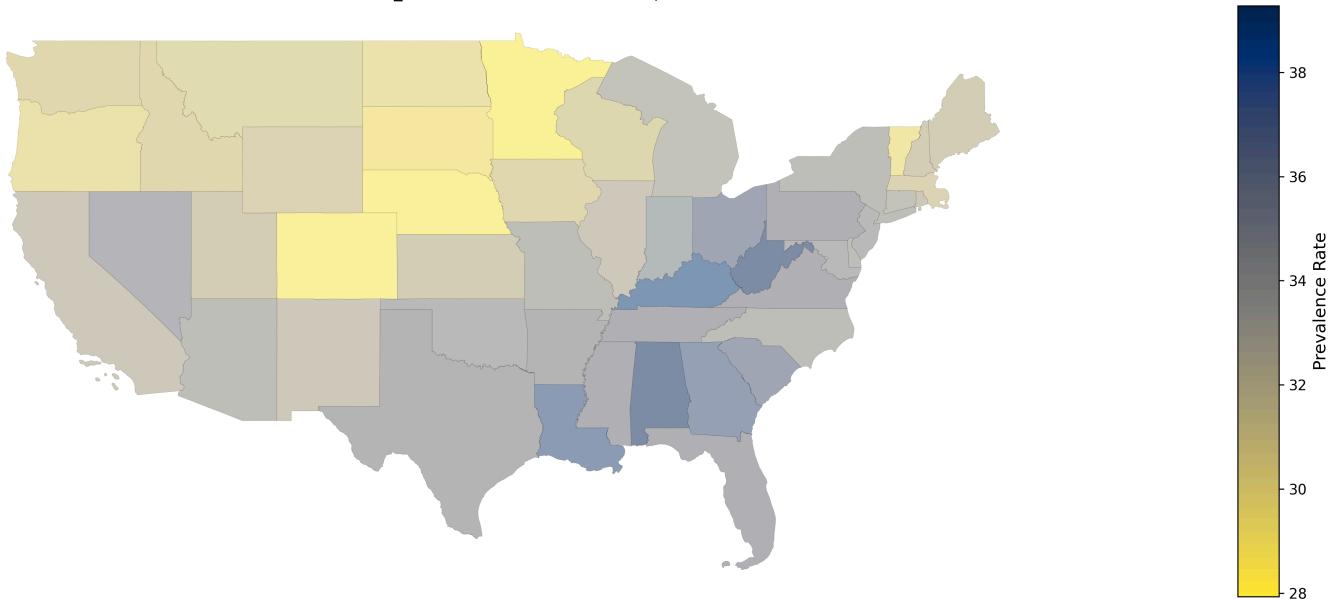
OBESITY\_CrudePrev - United States Map



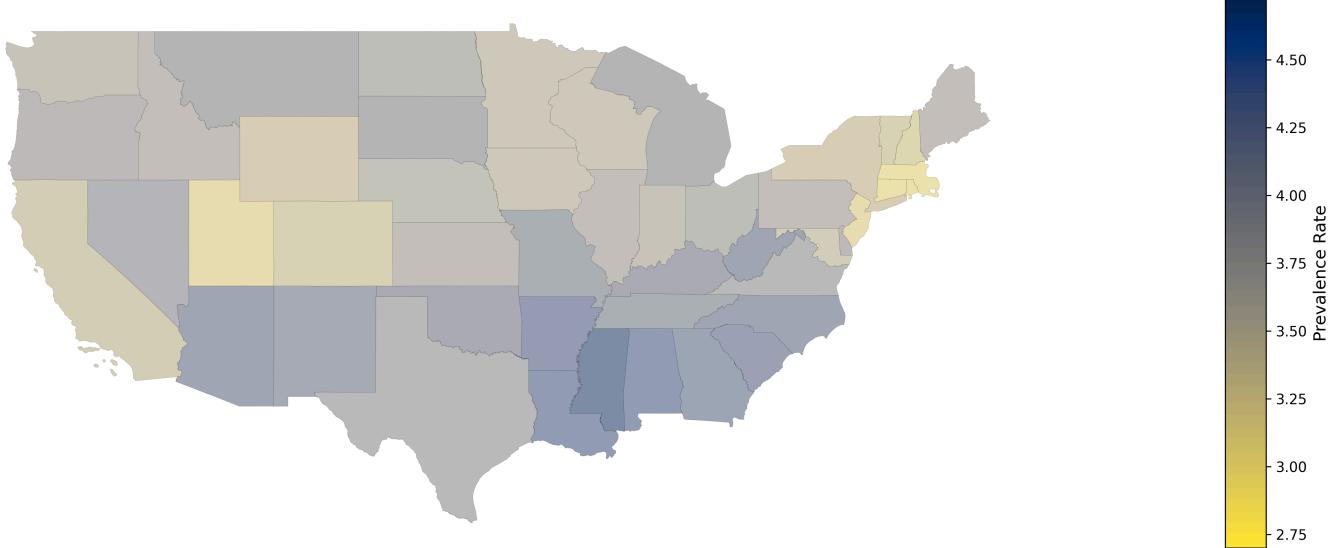
PHLTH\_CrudePrev - United States Map



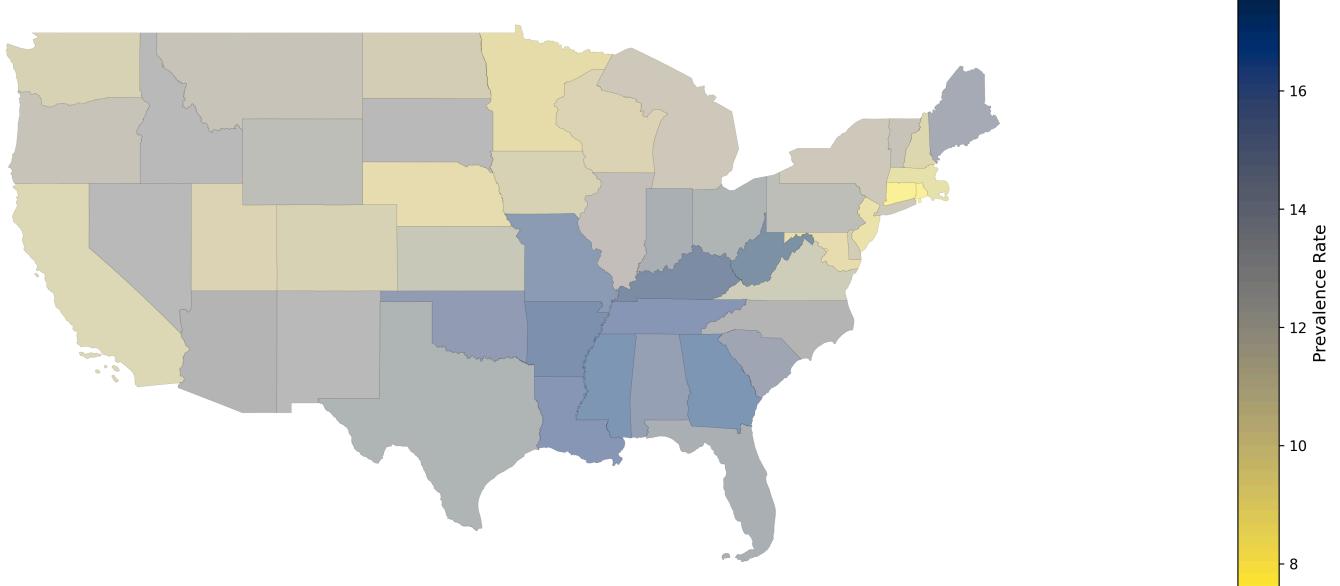
SLEEP\_CrudePrev - United States Map



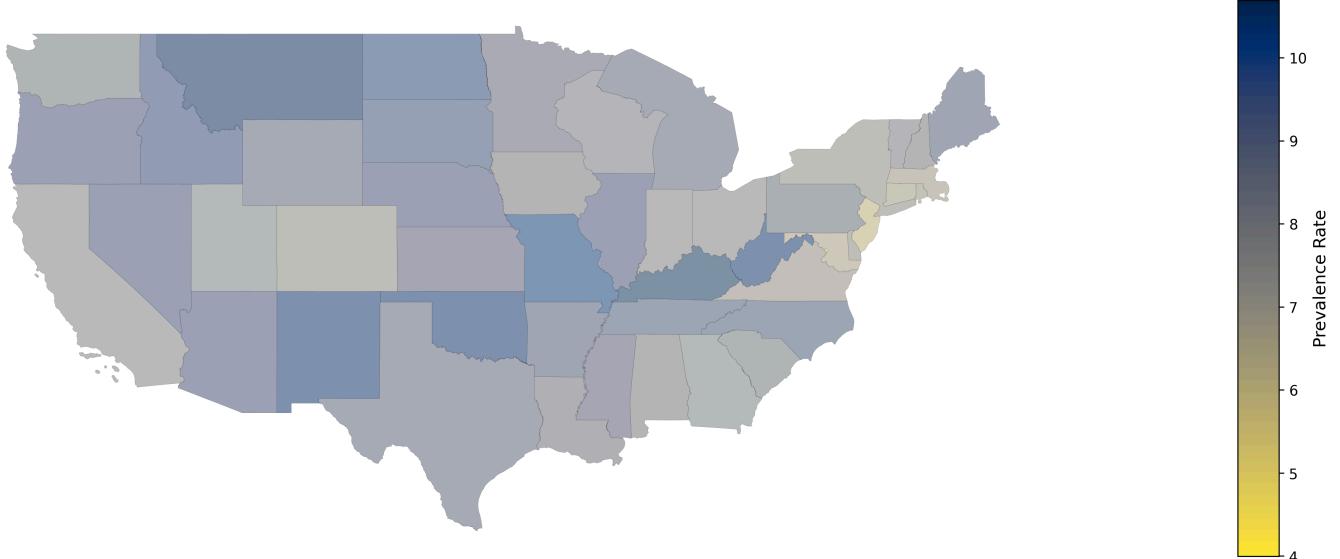
STROKE\_CrudePrev - United States Map



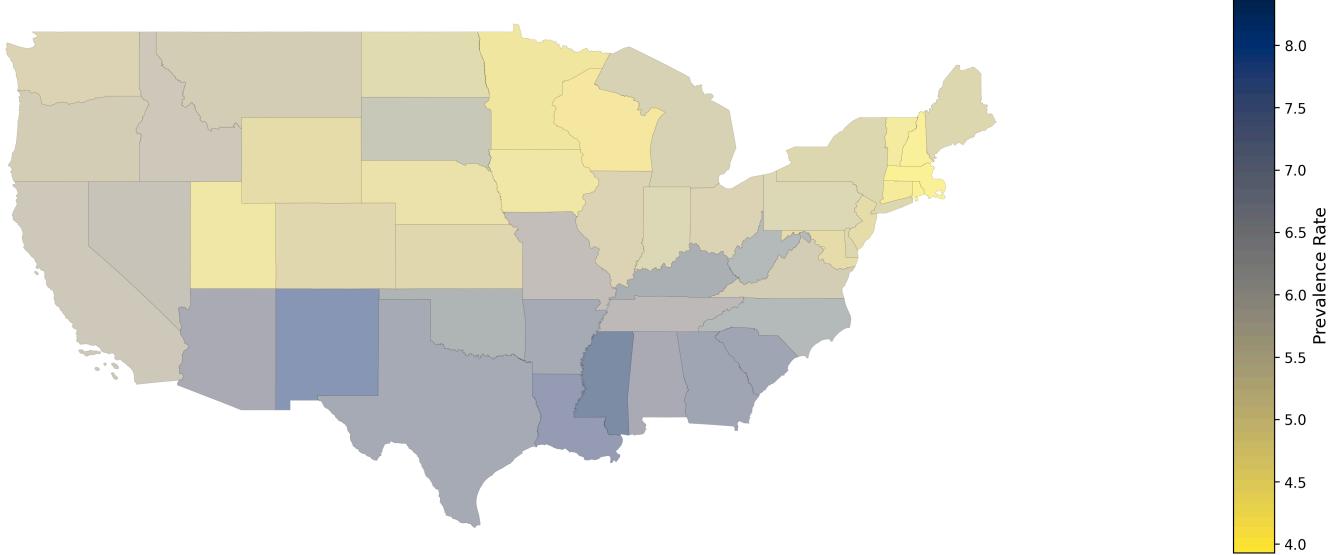
TEETHLOST\_CrudePrev - United States Map



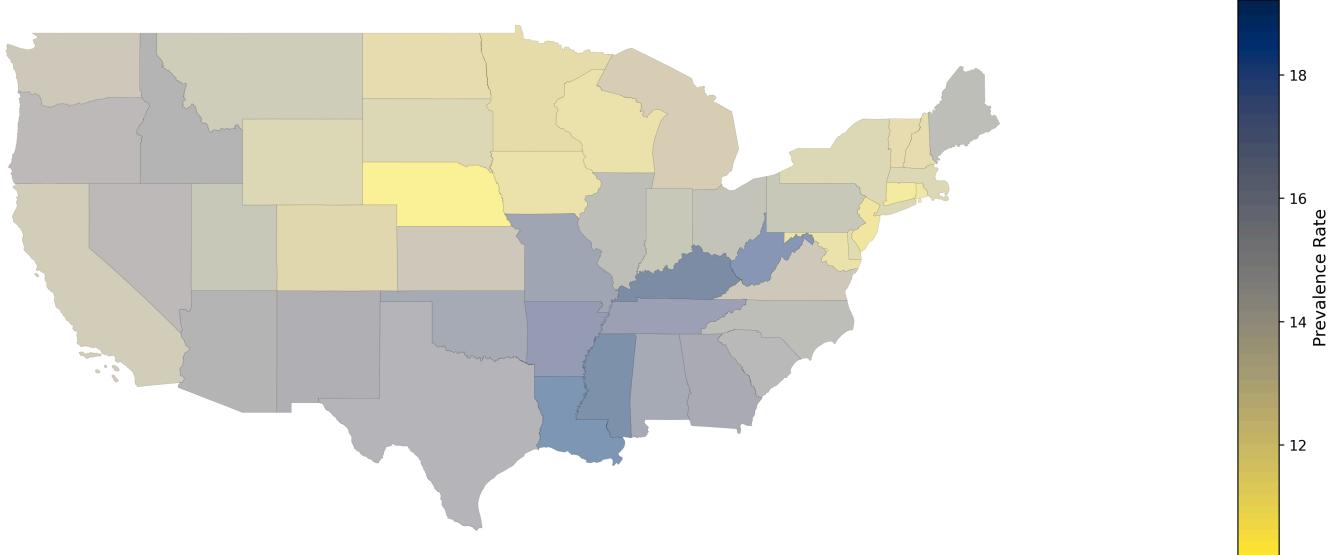
HEARING\_CrudePrev - United States Map



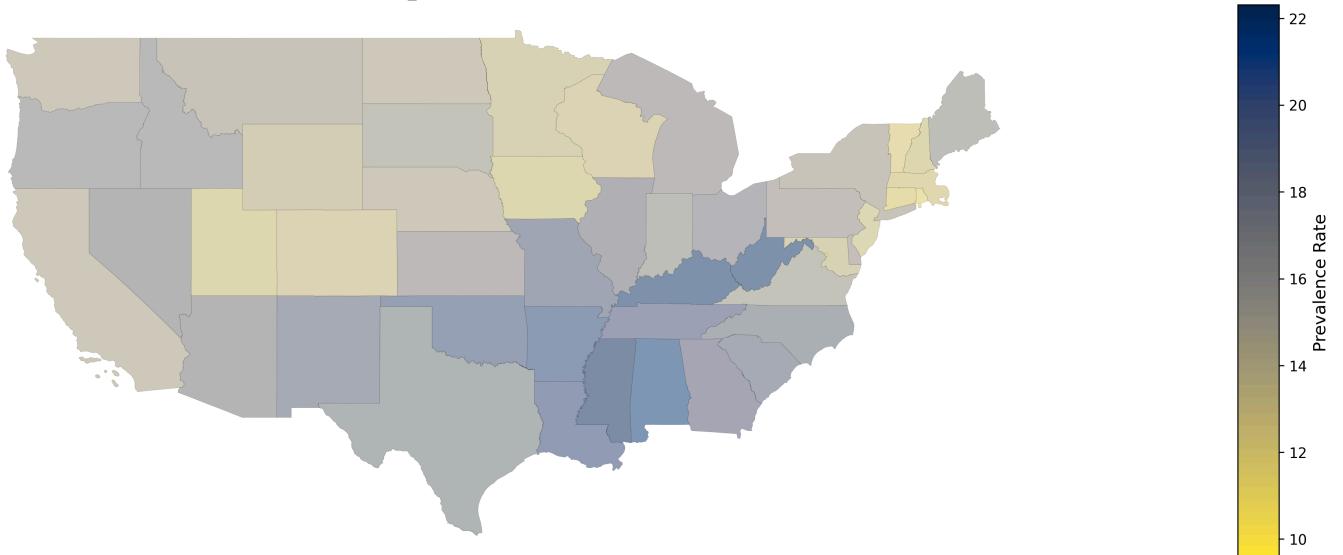
VISION\_CrudePrev - United States Map



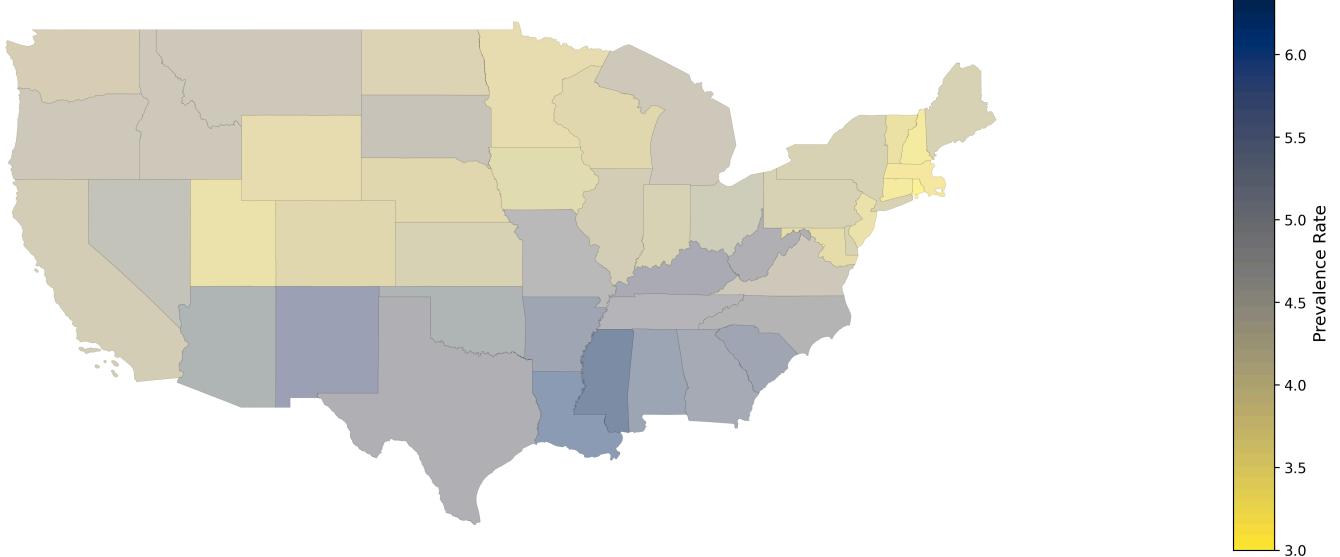
COGNITION\_CrudePrev - United States Map



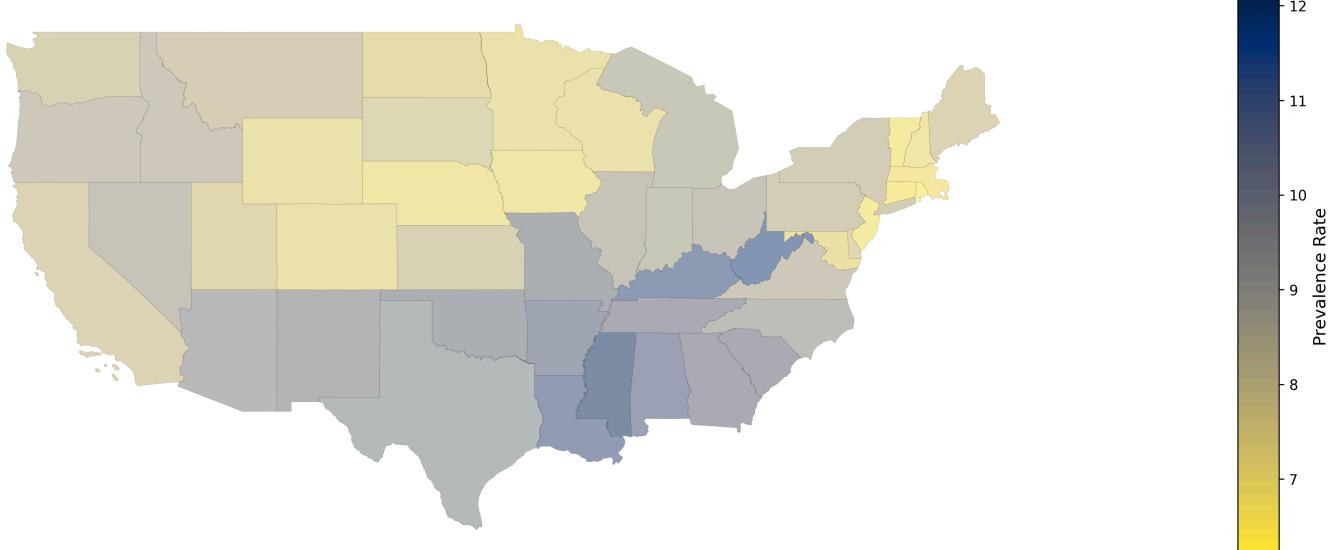
MOBILITY\_CrudePrev - United States Map



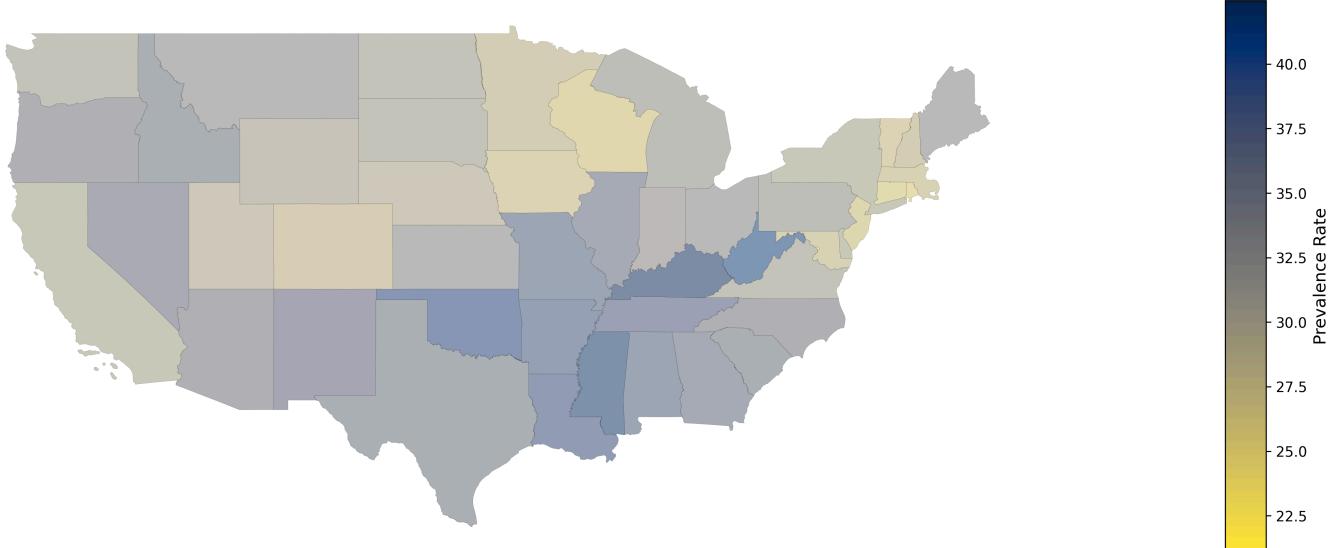
SELF CARE\_CrudePrev - United States Map



INDEPLIVE\_CrudePrev - United States Map



DISABILITY\_CrudePrev - United States Map

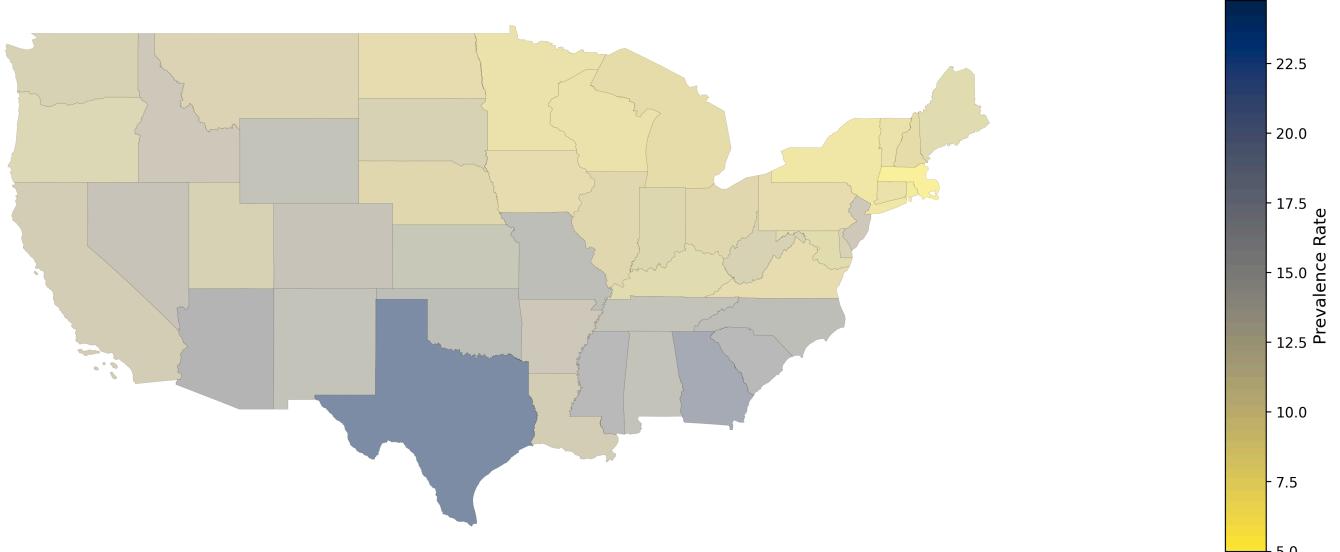


## Use Age-Adjusted Data

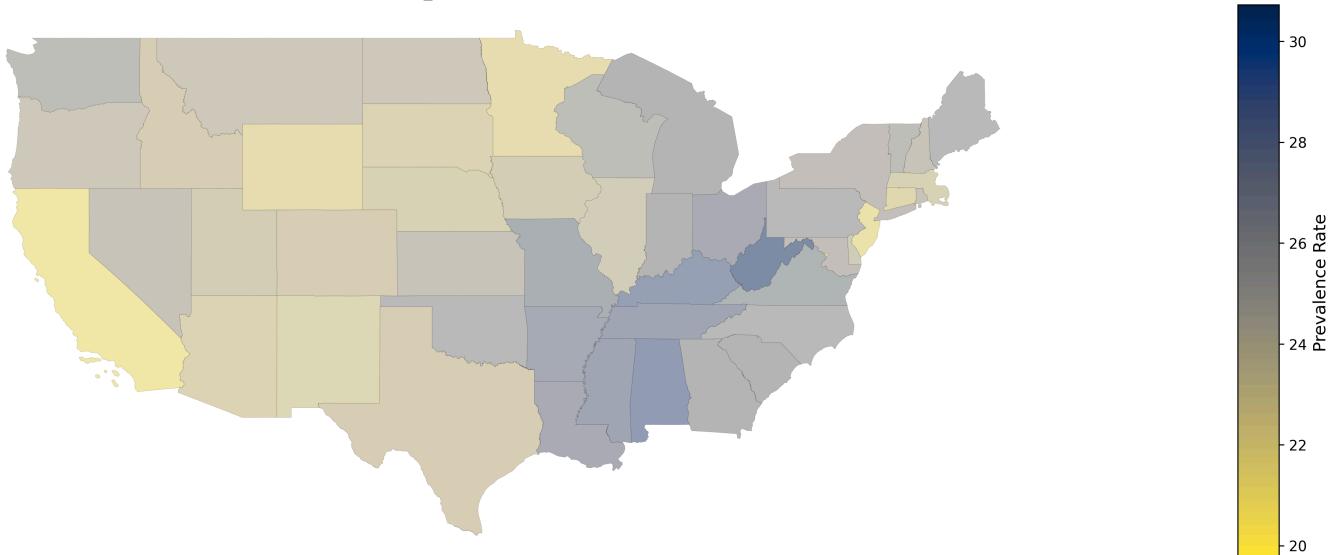
```
In [ ]: # Get columns ending with '_CrudePrev'
columns_to_plot = [col for col in merged_gdf.columns if col.endswith('_AdjPrev')]

plot_data_multiple_columns(merged_gdf, columns_to_plot, 'cividis_r')
```

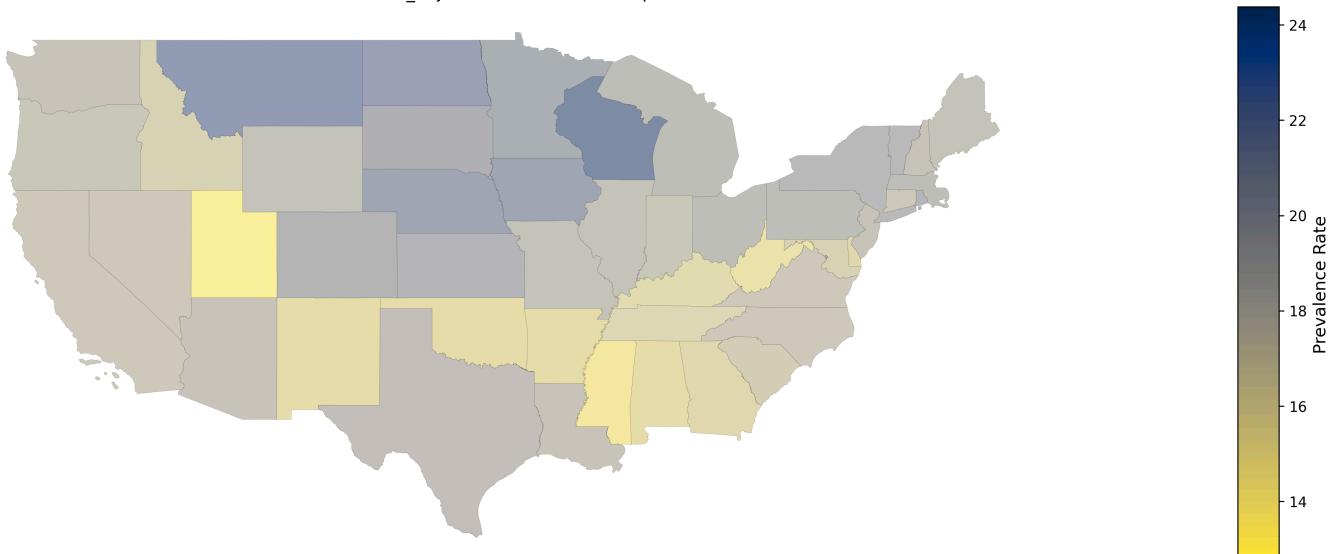
ACCESS2\_AdjPrev - United States Map



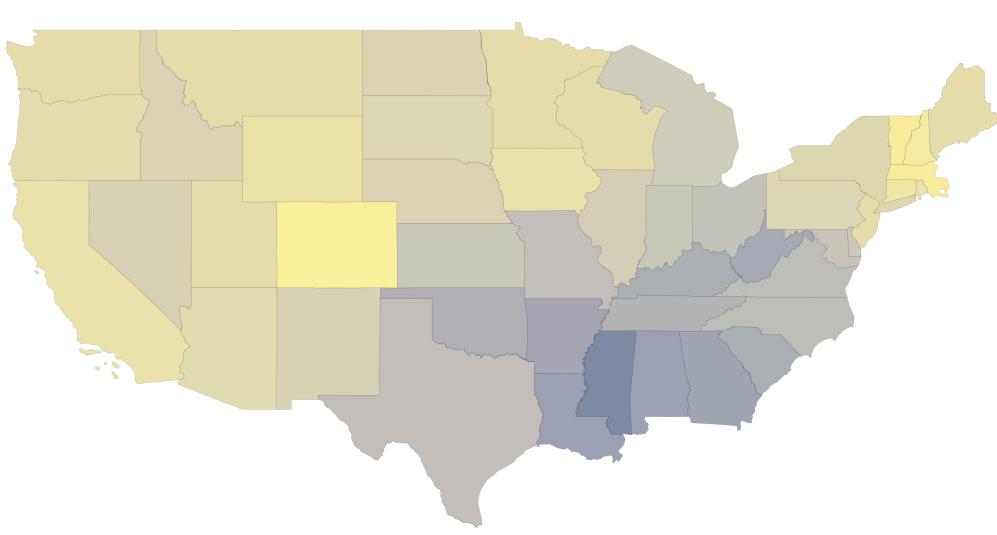
ARTHRITIS\_AdjPrev - United States Map



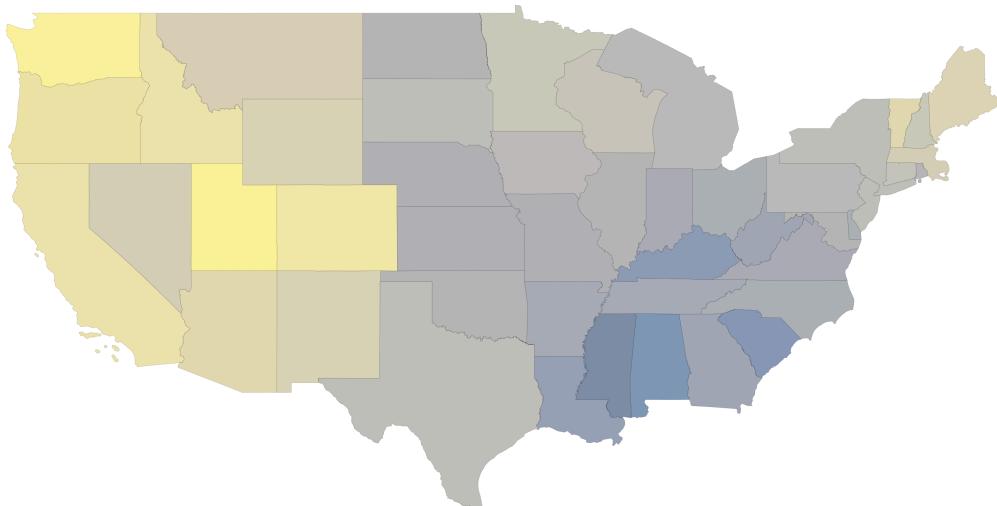
BINGE\_AdjPrev - United States Map



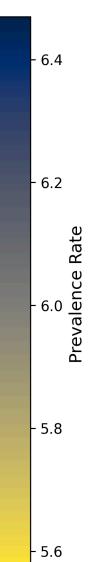
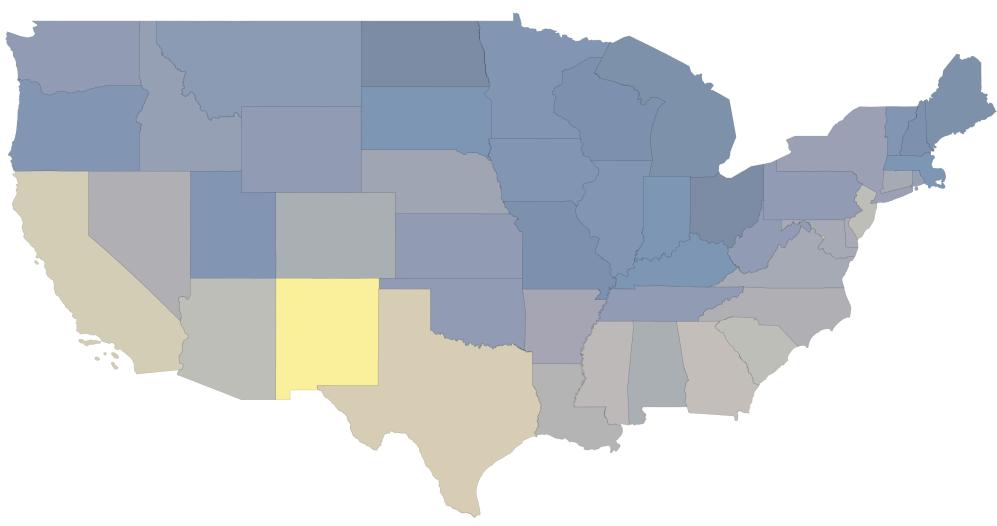
BPHIGH\_AdjPrev - United States Map



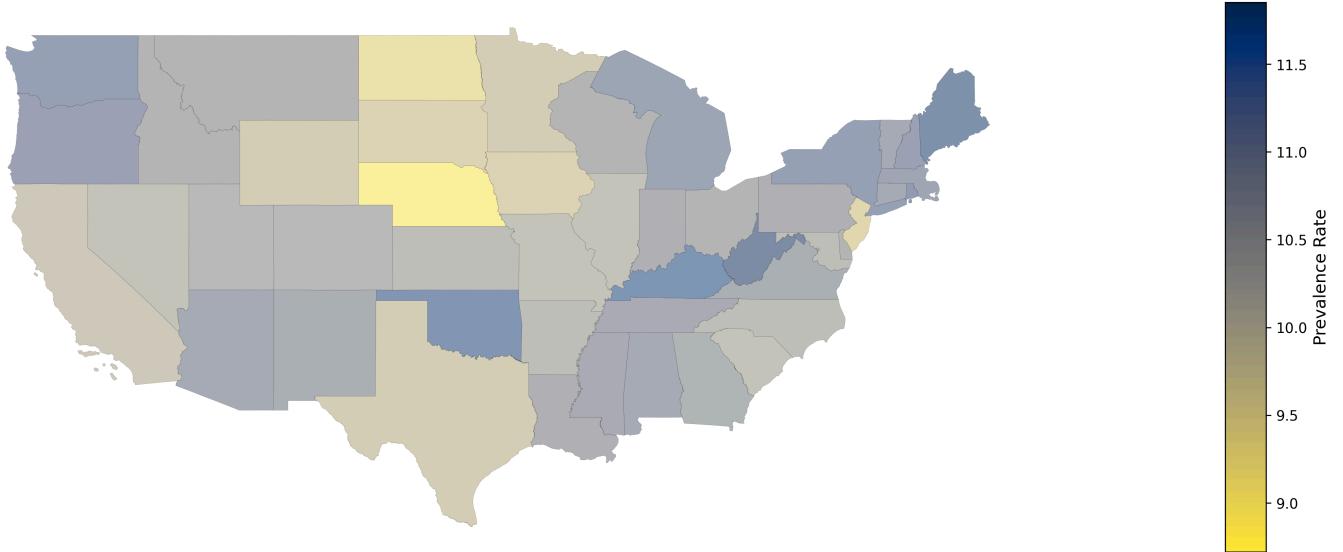
BPMED\_AdjPrev - United States Map



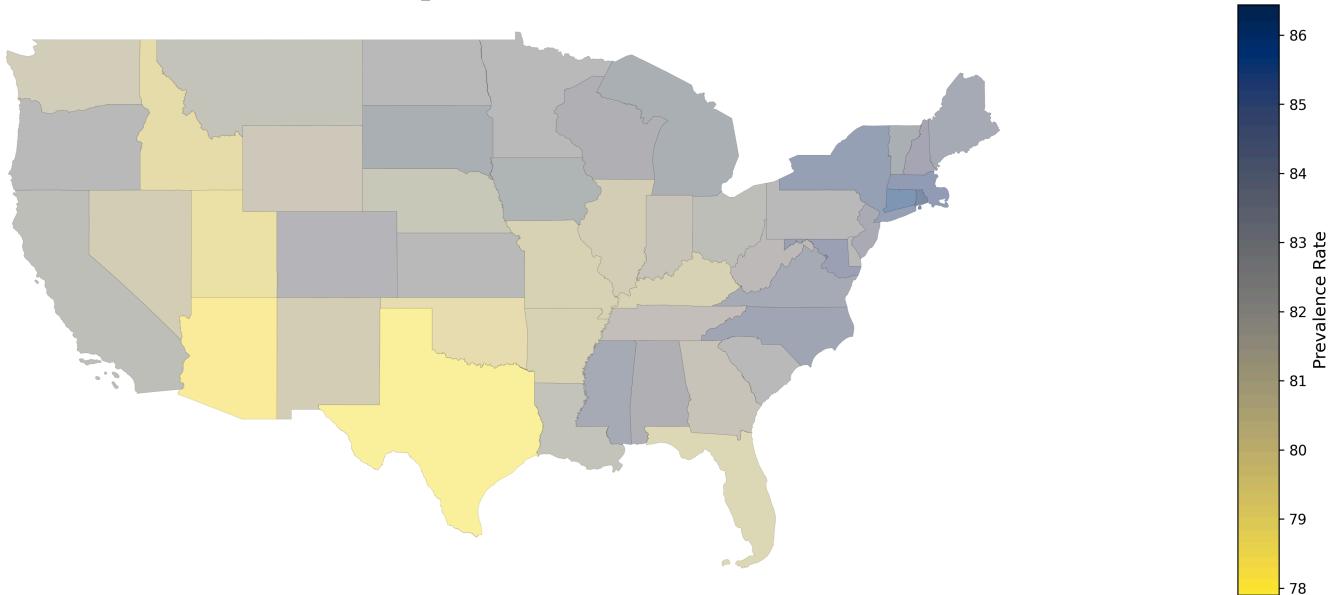
CANCER\_AdjPrev - United States Map



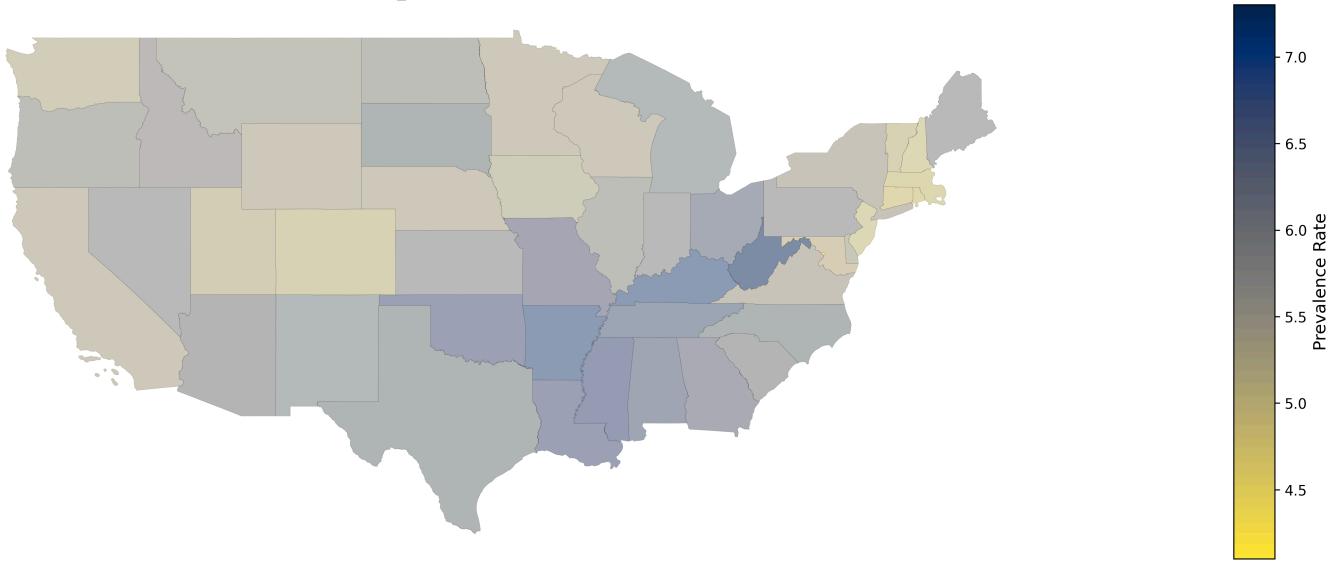
CASTHMA\_AdjPrev - United States Map



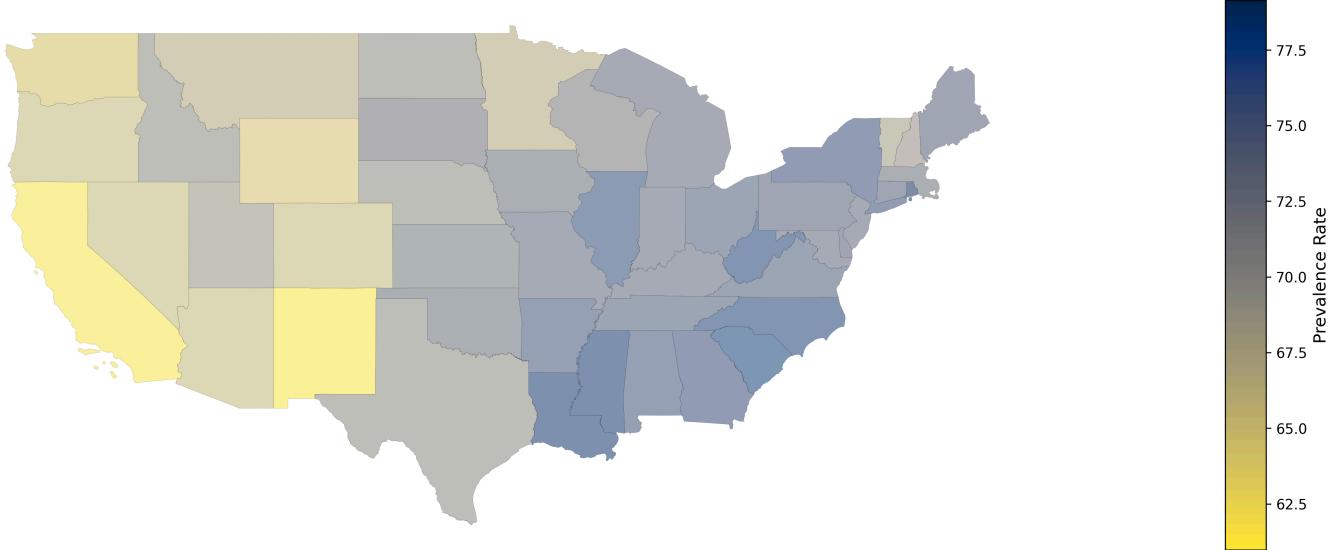
CERVICAL\_AdjPrev - United States Map



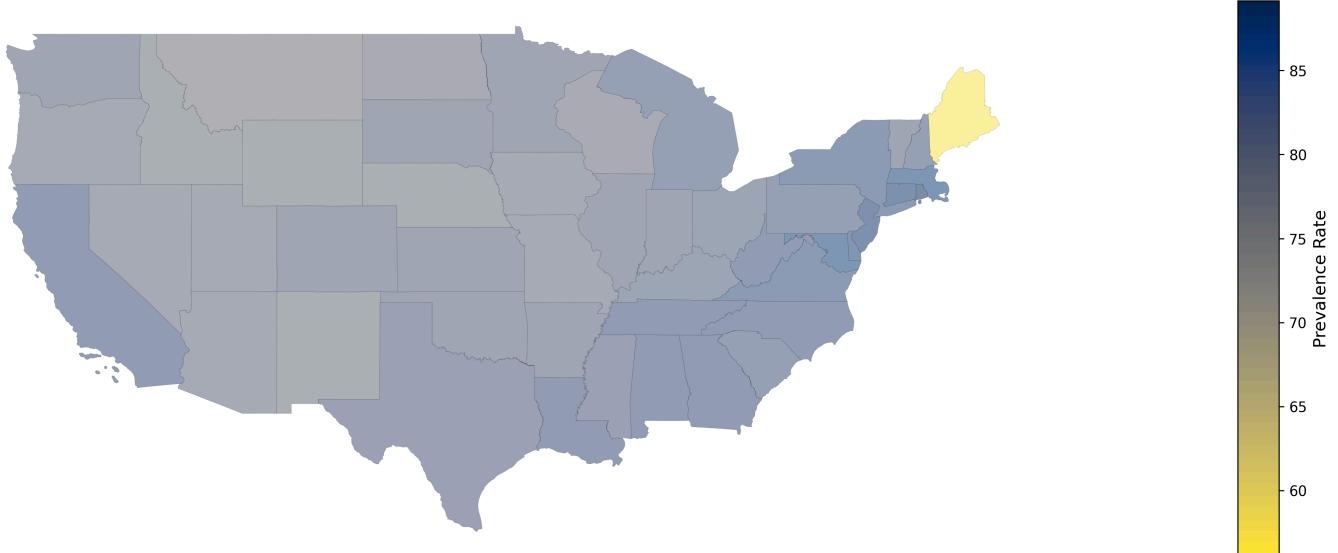
CHD\_AdjPrev - United States Map



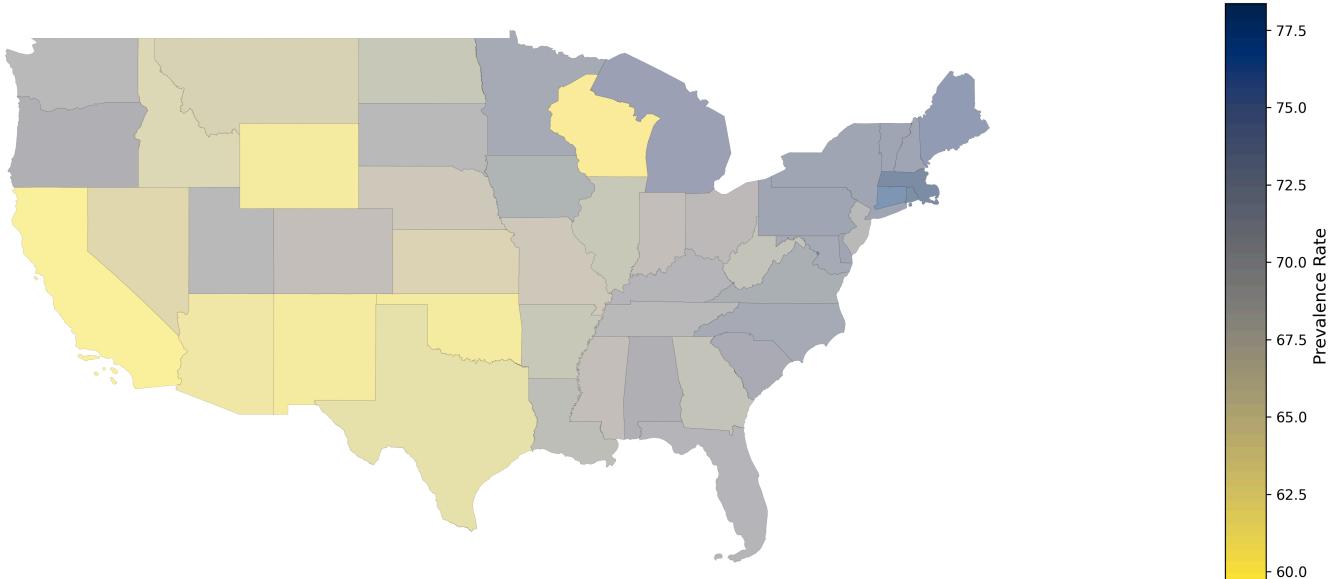
CHECKUP\_AdjPrev - United States Map



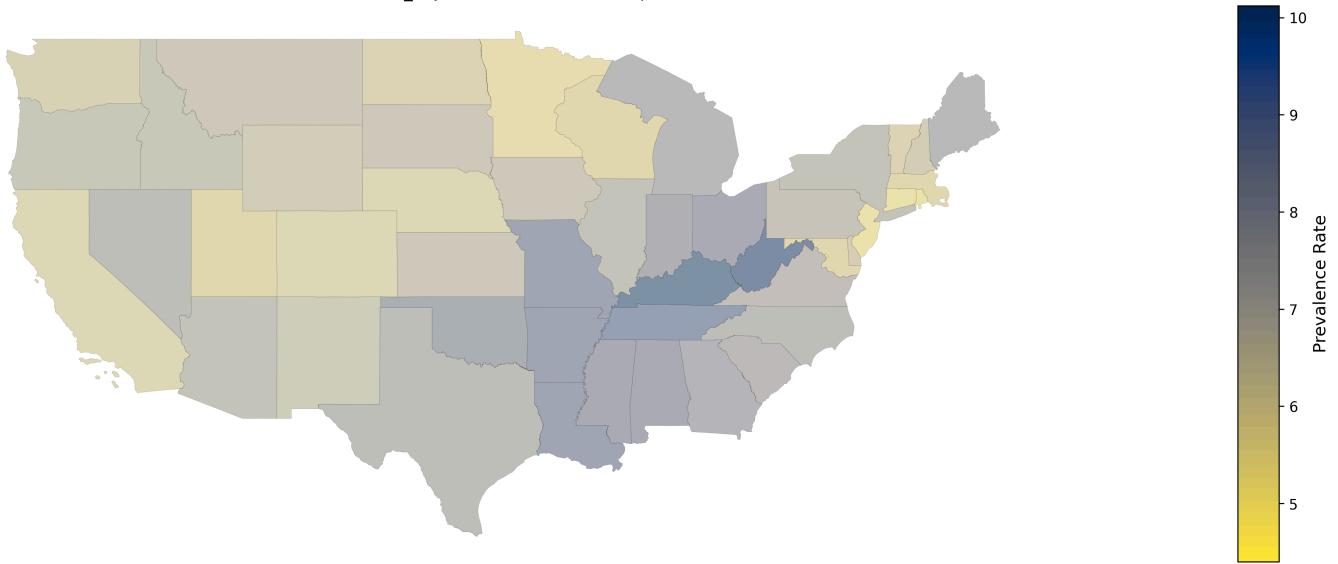
CHOLSCREEN\_AdjPrev - United States Map



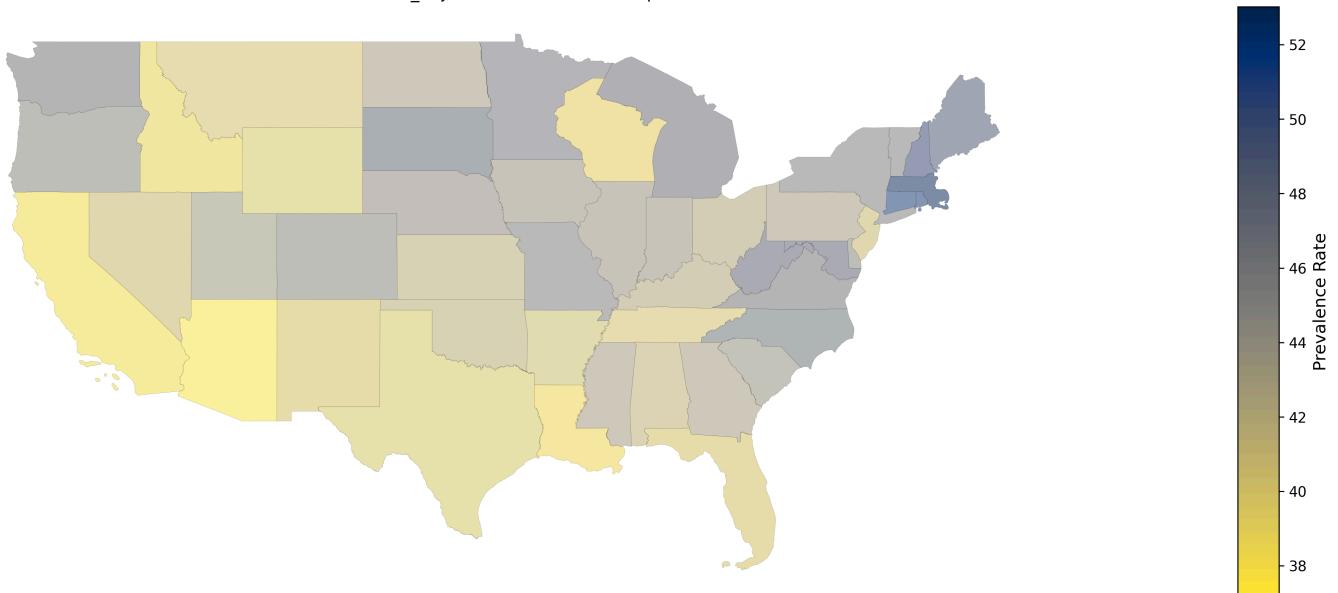
COLON\_SCREEN\_AdjPrev - United States Map



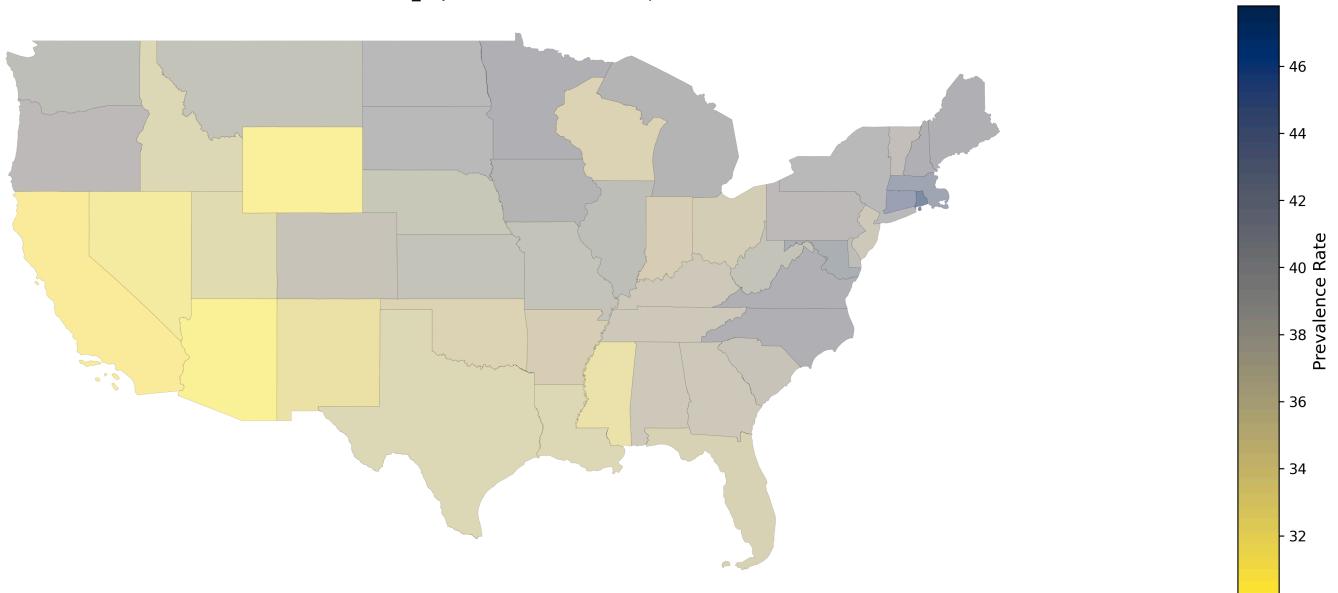
COPD\_AdjPrev - United States Map



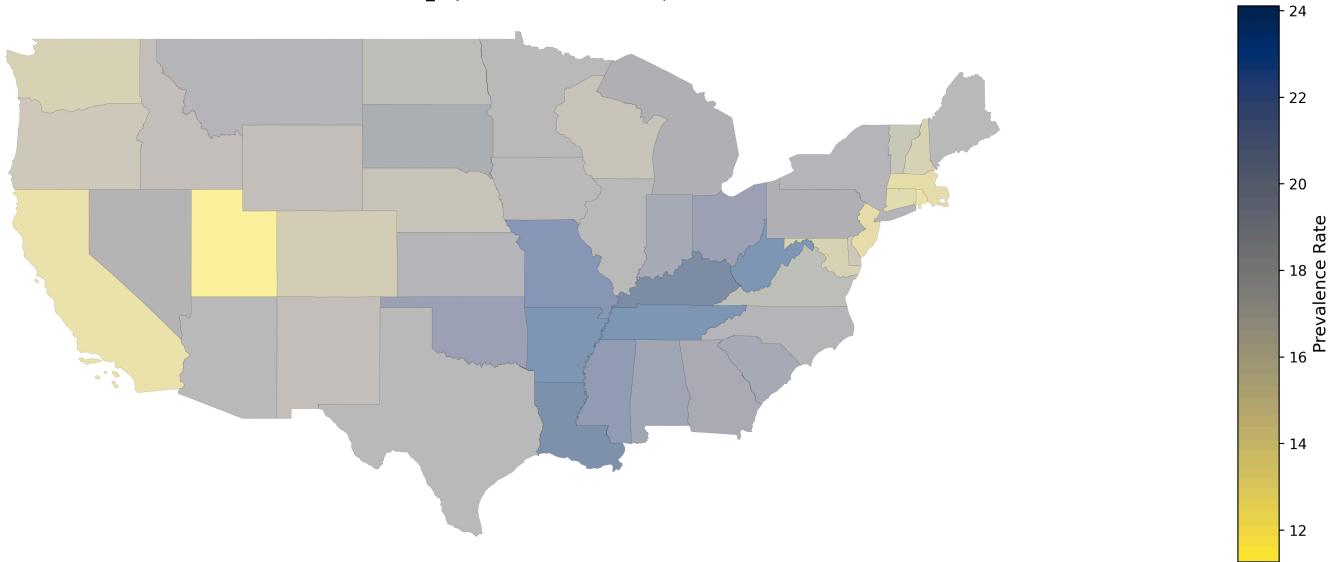
COREM\_AdjPrev - United States Map



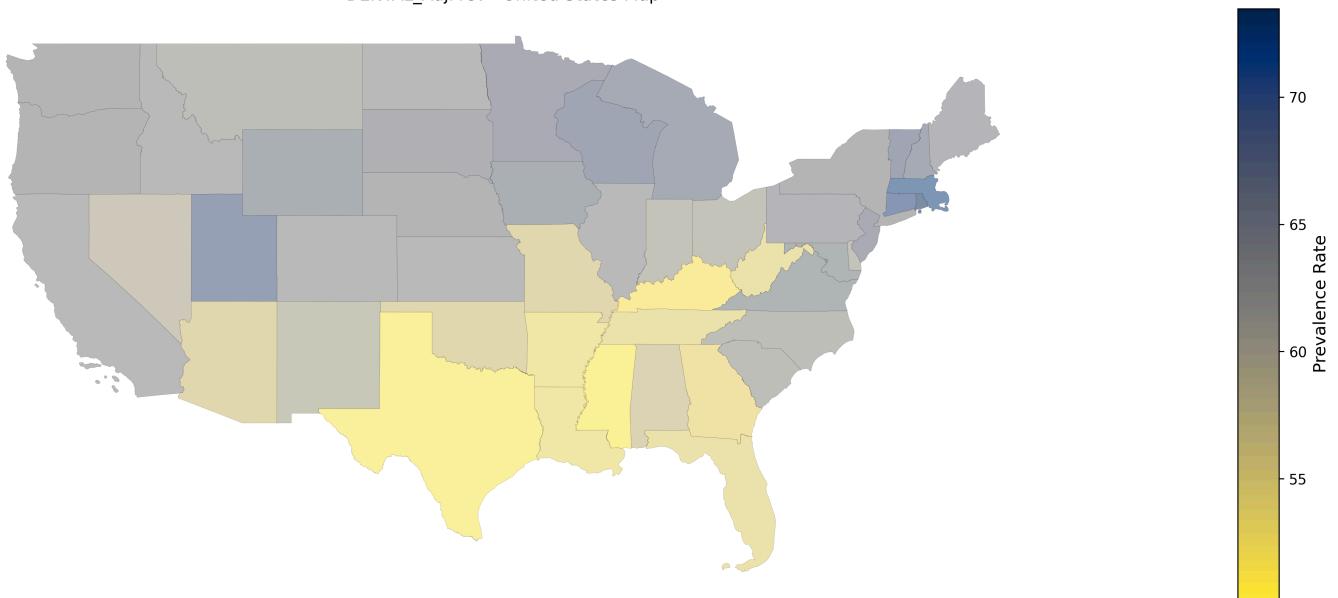
COREW\_AdjPrev - United States Map



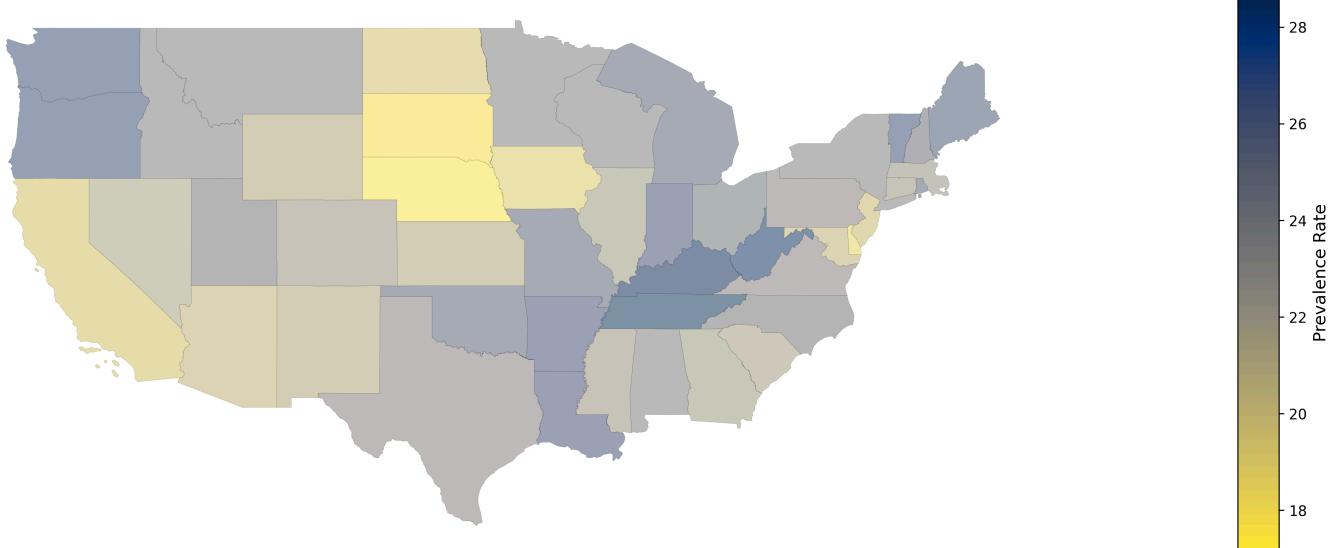
CSMOKING\_AdjPrev - United States Map



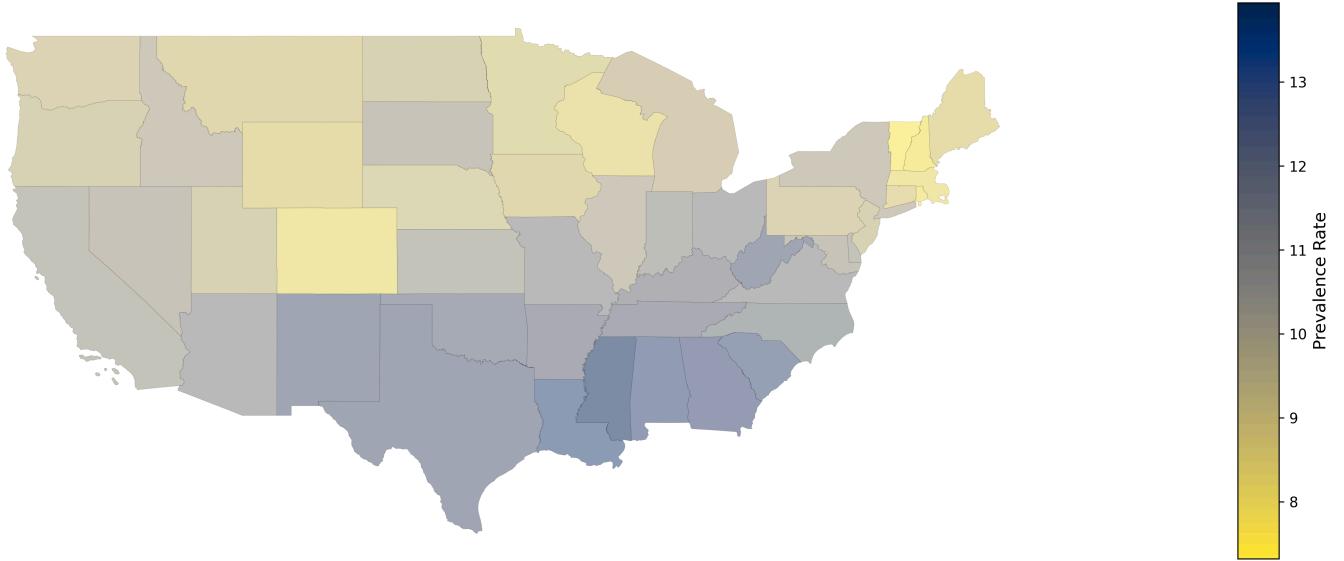
DENTAL\_AdjPrev - United States Map



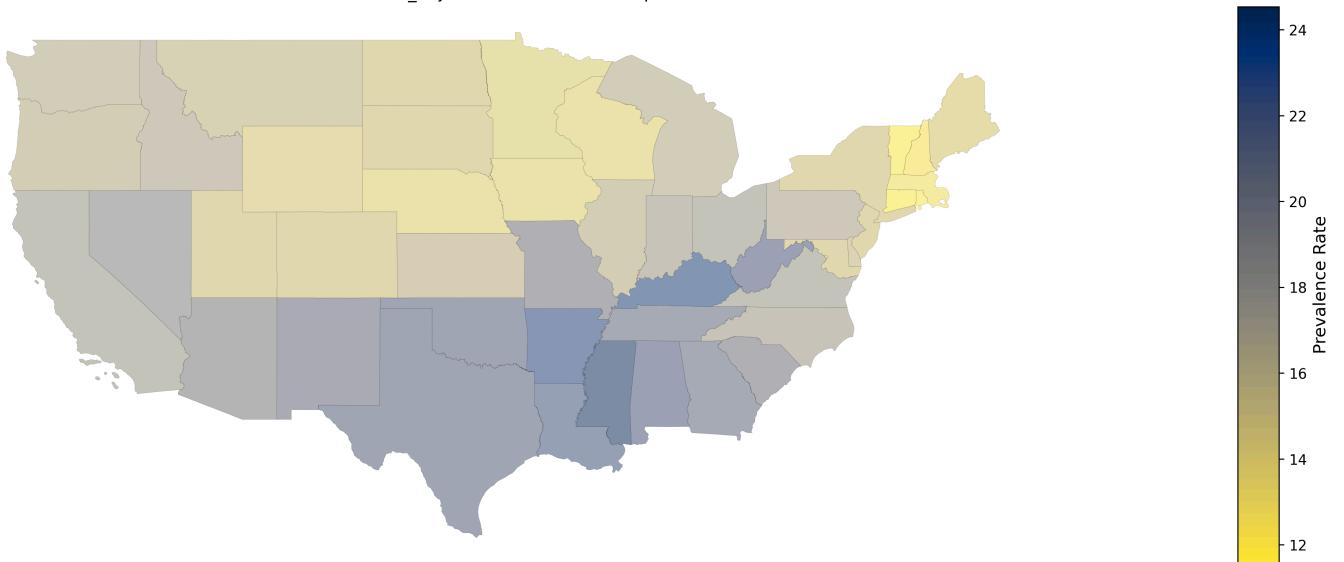
DEPRESSION\_AdjPrev - United States Map



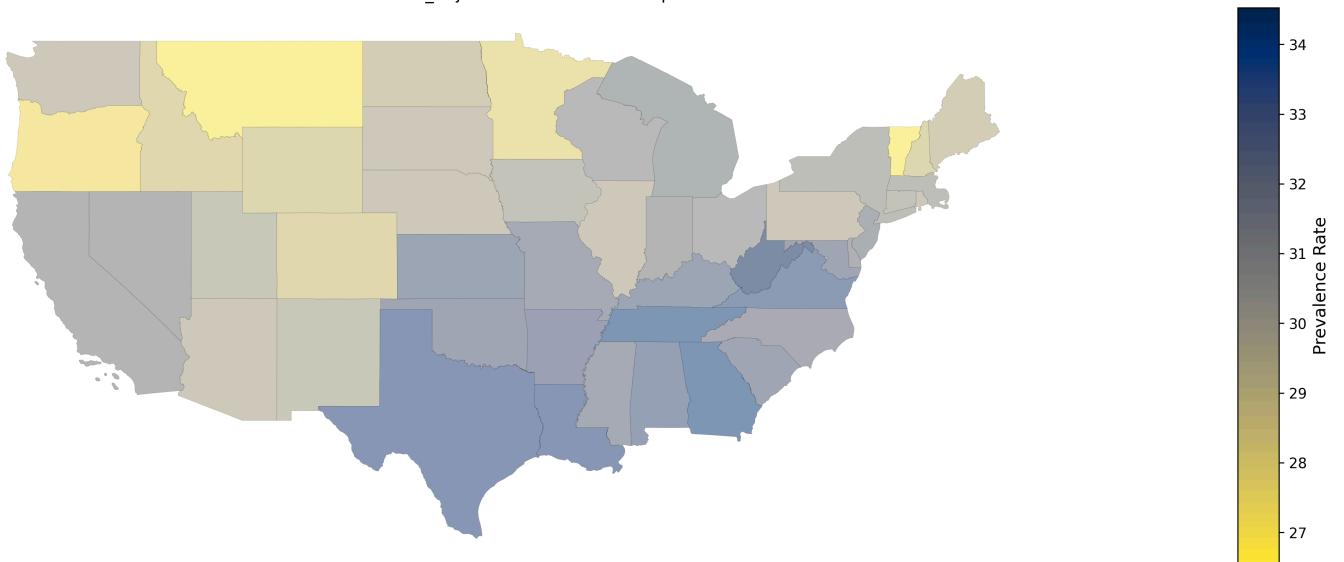
DIABETES\_AdjPrev - United States Map



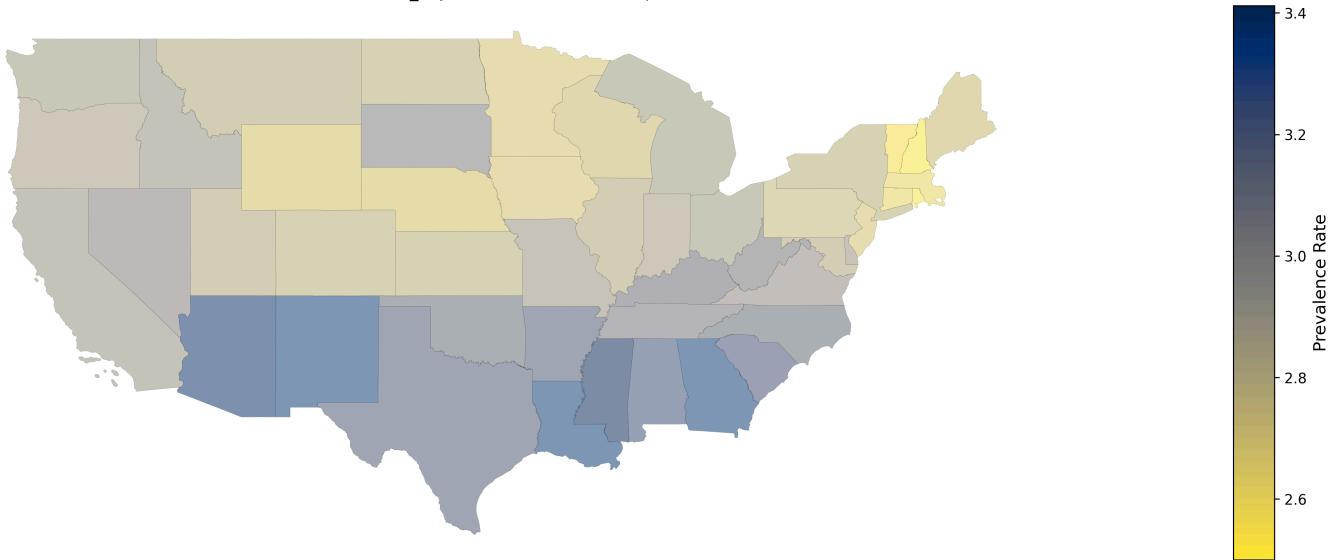
GHLTH\_AdjPrev - United States Map



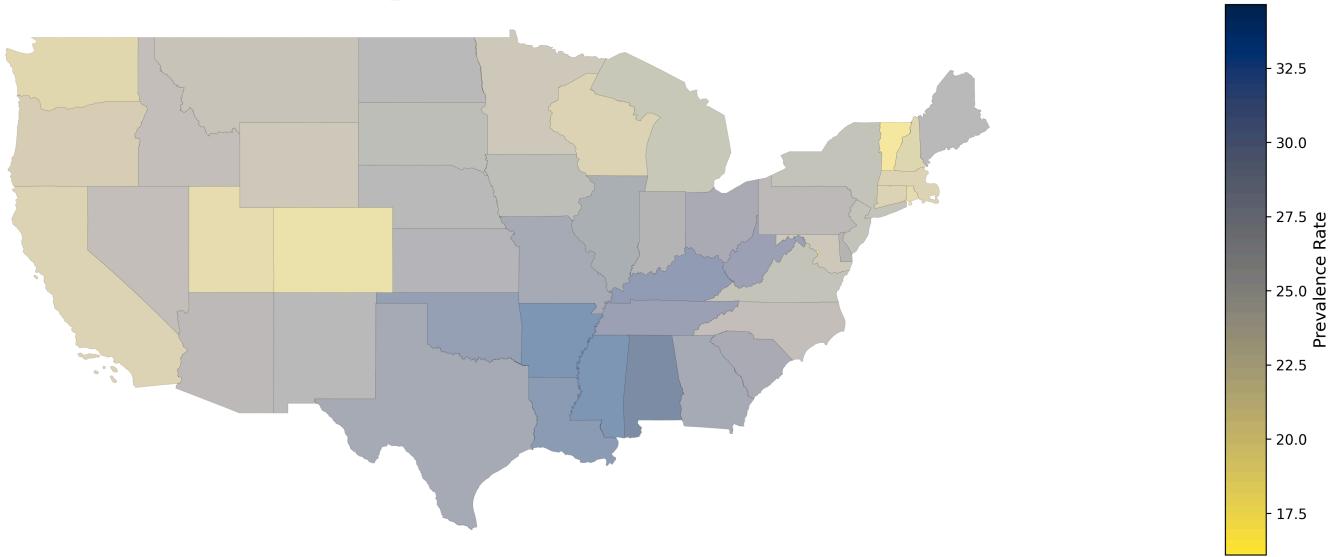
HIGHCHOL\_AdjPrev - United States Map



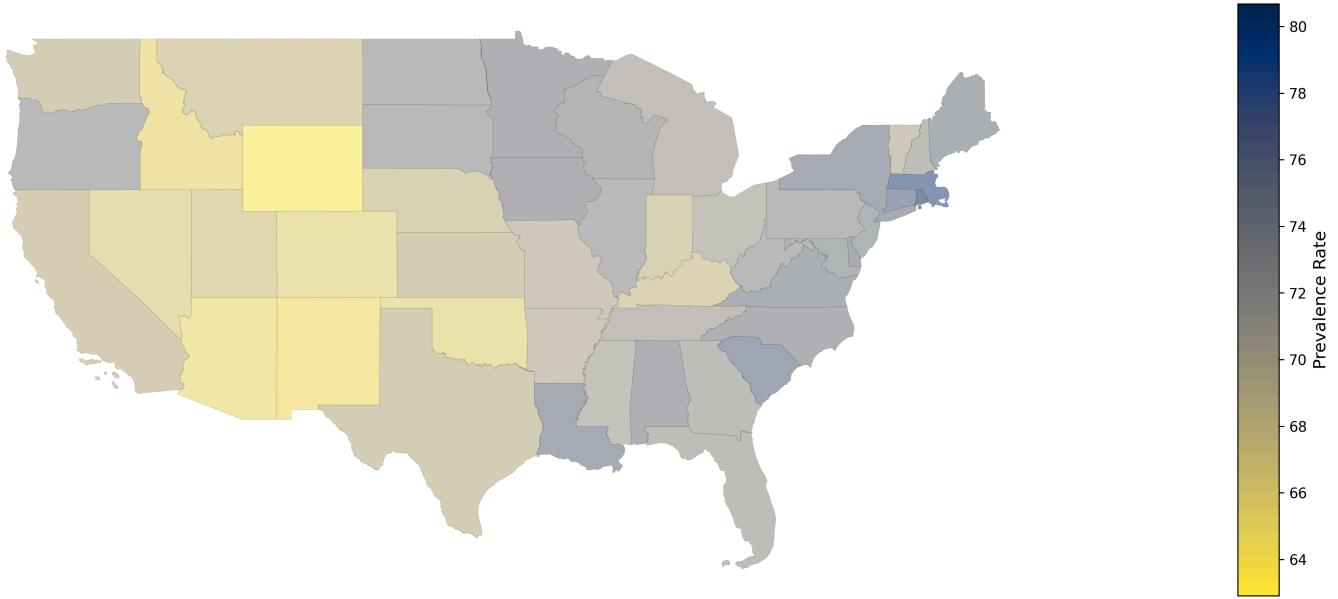
KIDNEY\_AdjPrev - United States Map



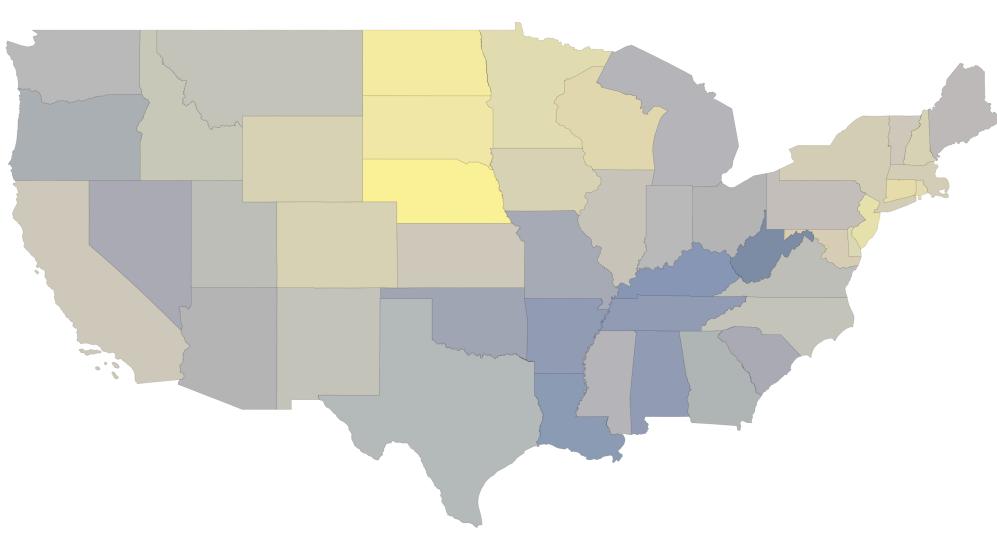
LPA\_AdjPrev - United States Map



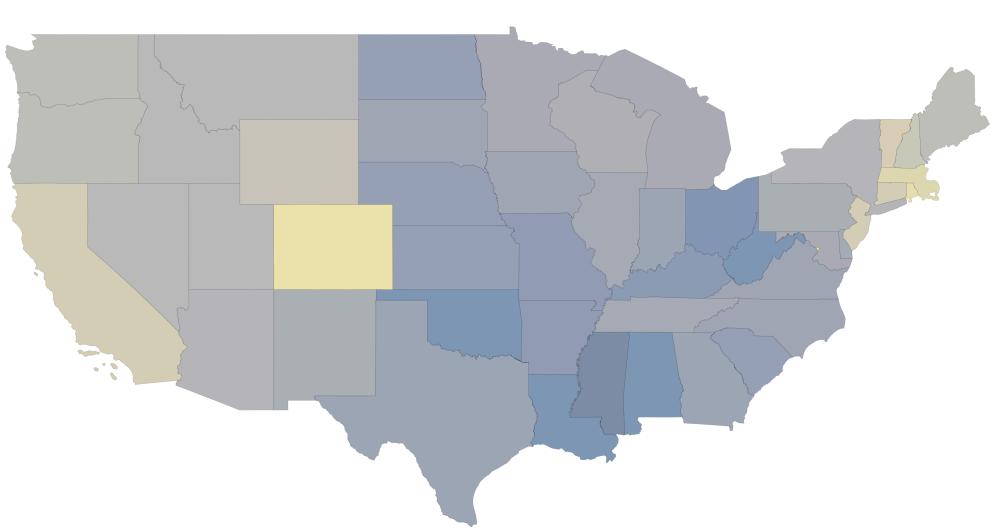
MAMMOUSE\_AdjPrev - United States Map



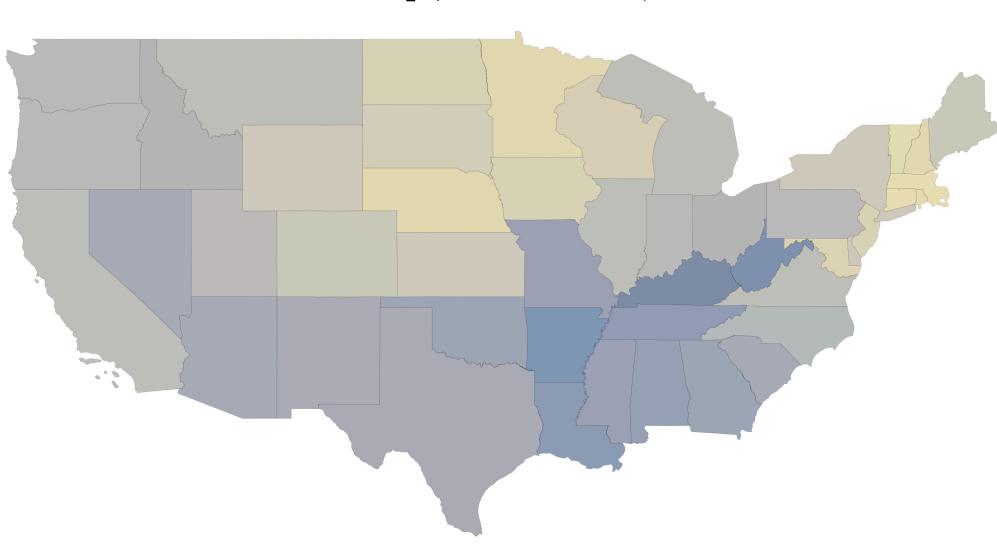
MHLTH\_AdjPrev - United States Map



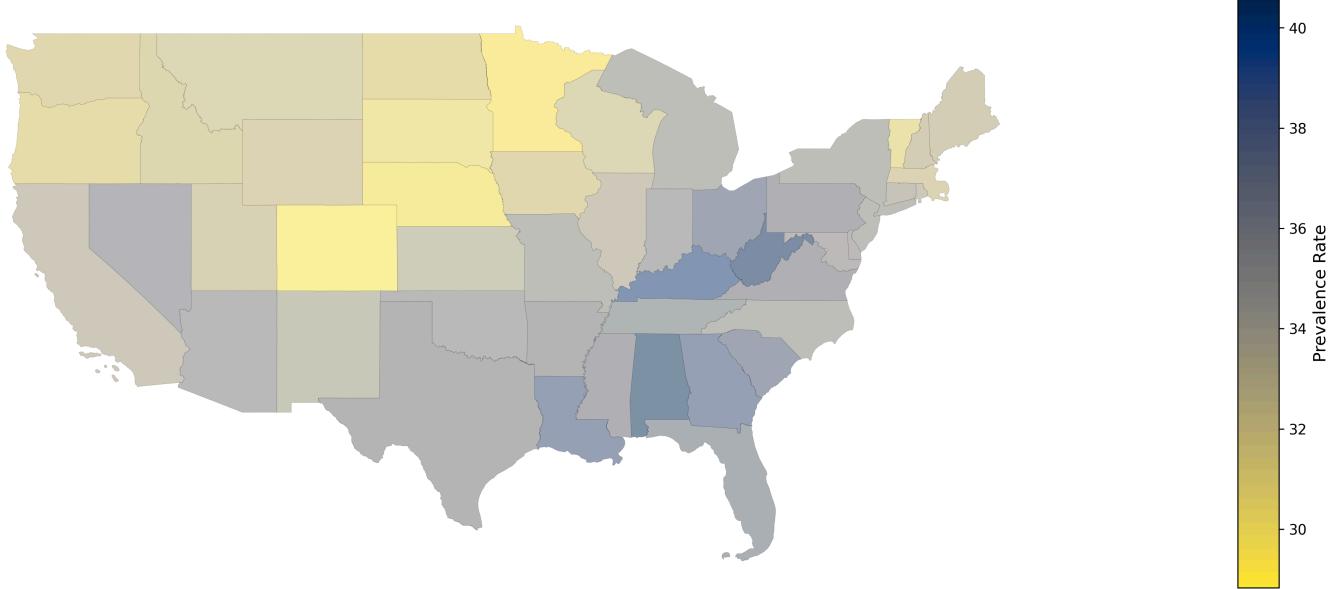
OBESITY\_AdjPrev - United States Map



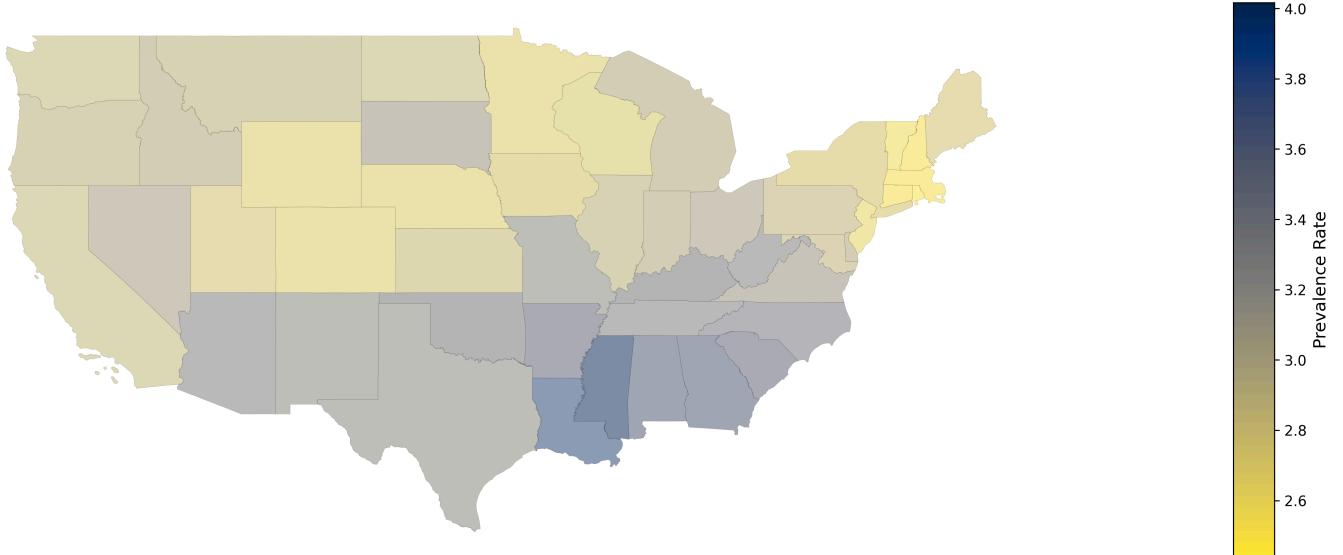
PHLTH\_AdjPrev - United States Map



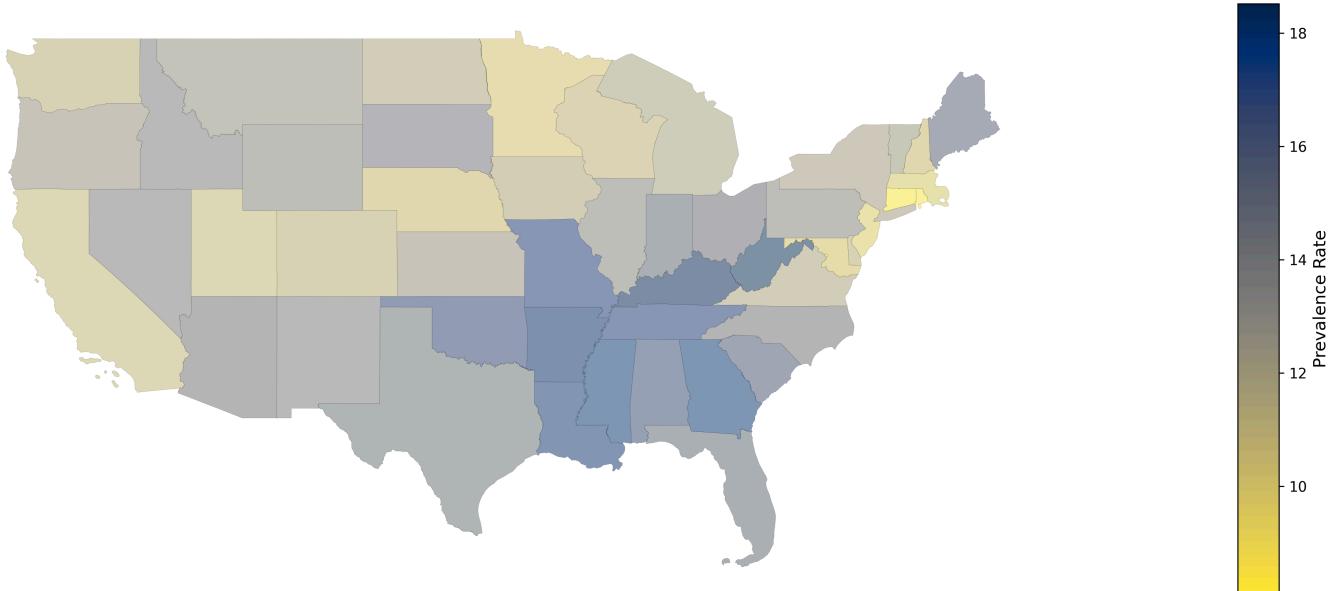
SLEEP\_AdjPrev - United States Map



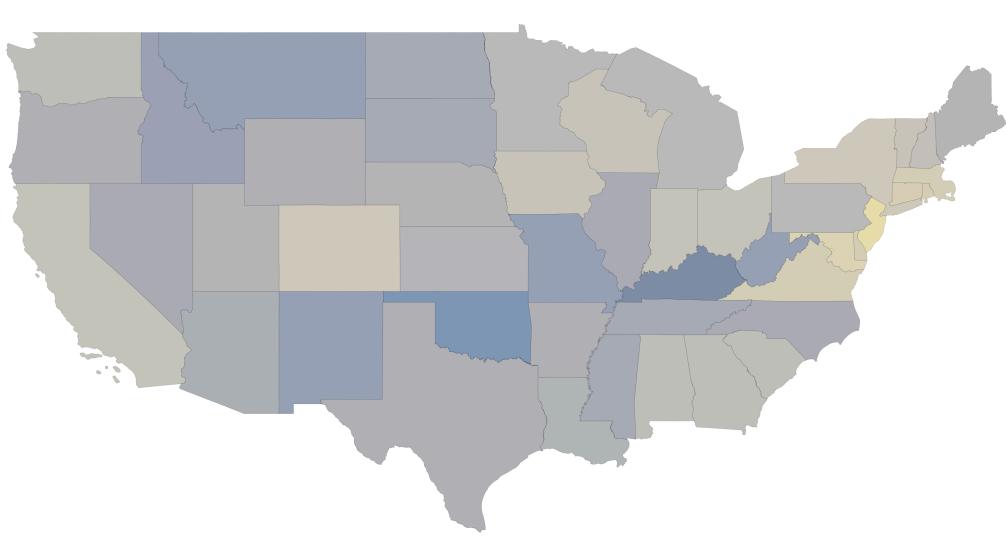
STROKE\_AdjPrev - United States Map



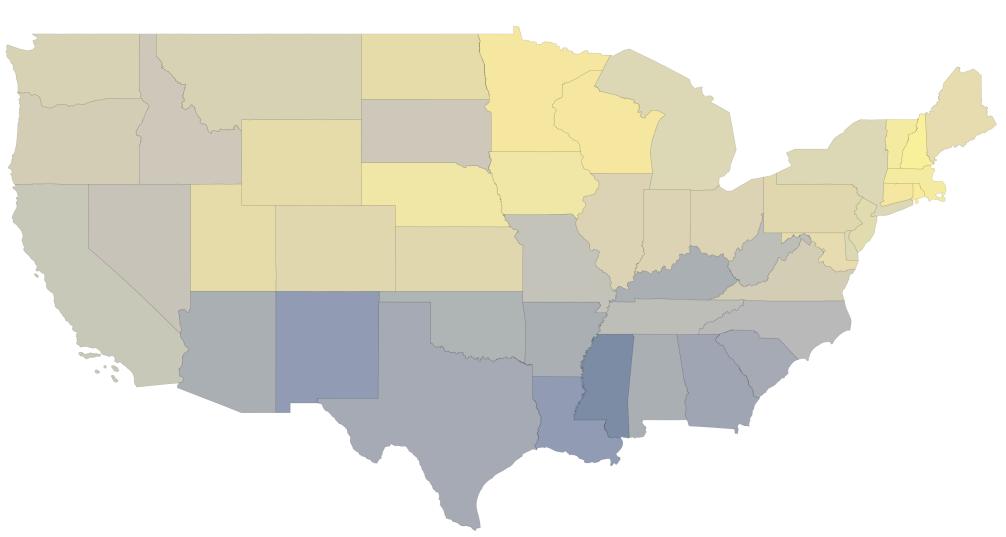
TEETHLOST\_AdjPrev - United States Map



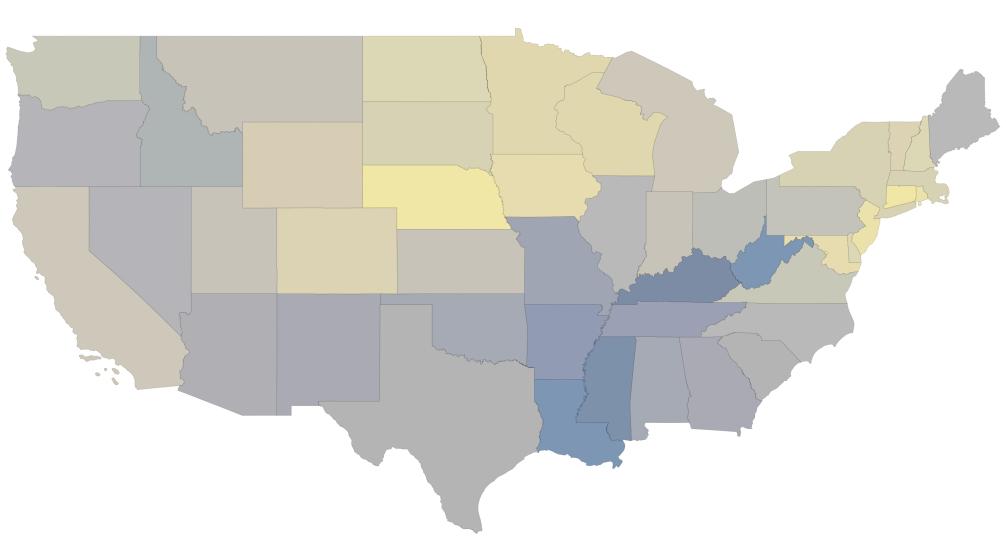
HEARING\_AdjPrev - United States Map



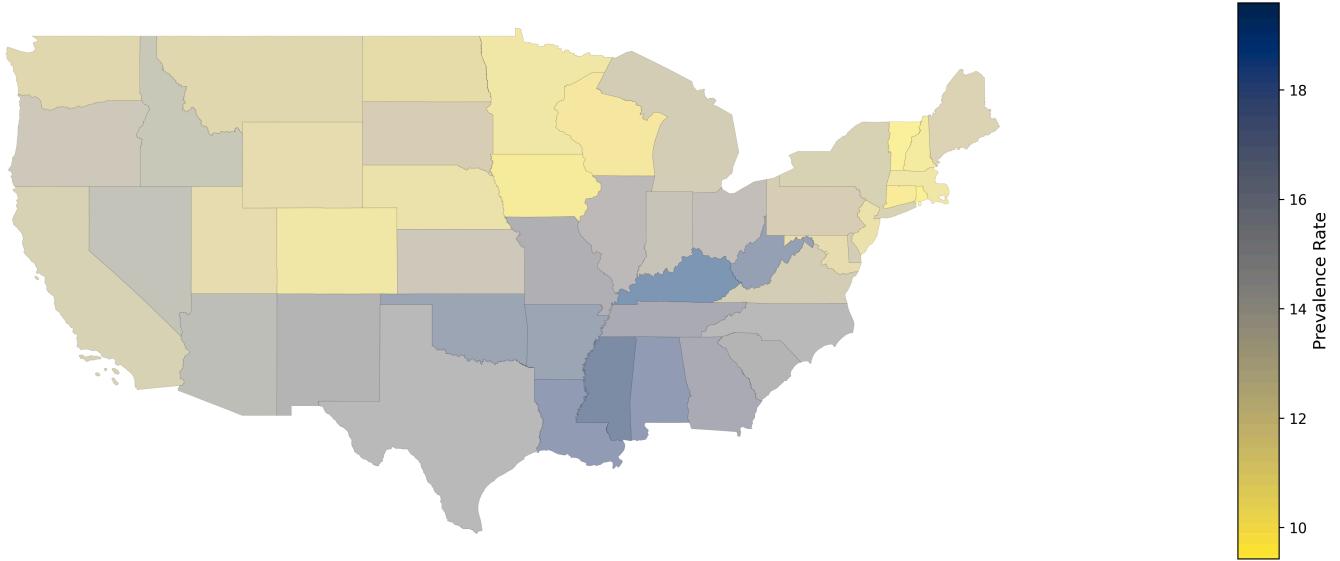
VISION\_AdjPrev - United States Map



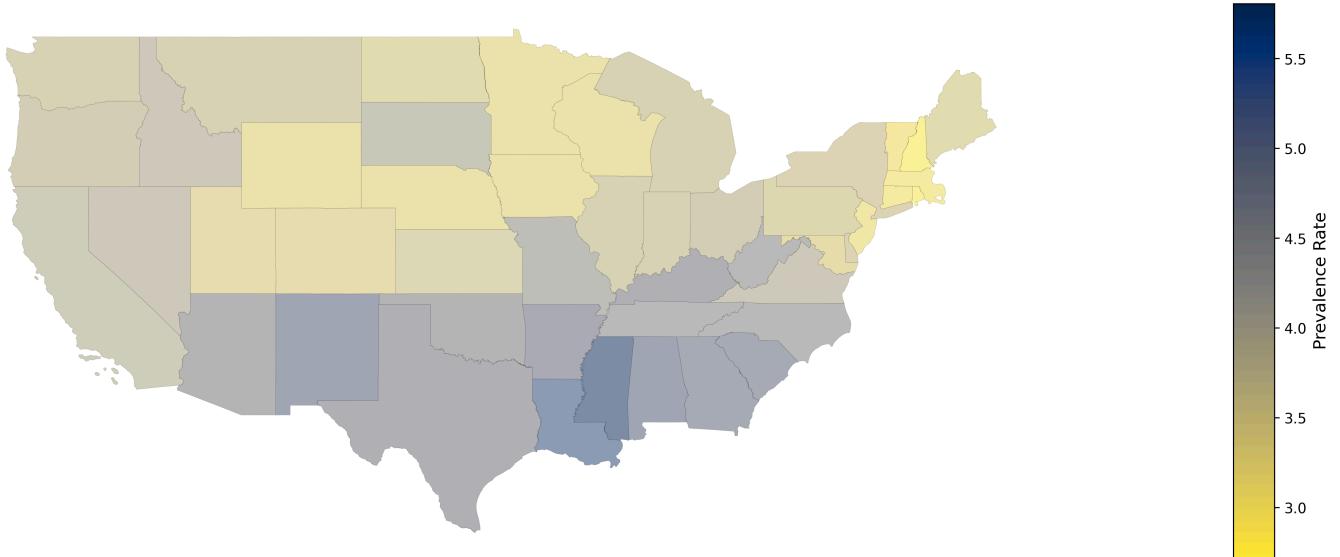
COGNITION\_AdjPrev - United States Map



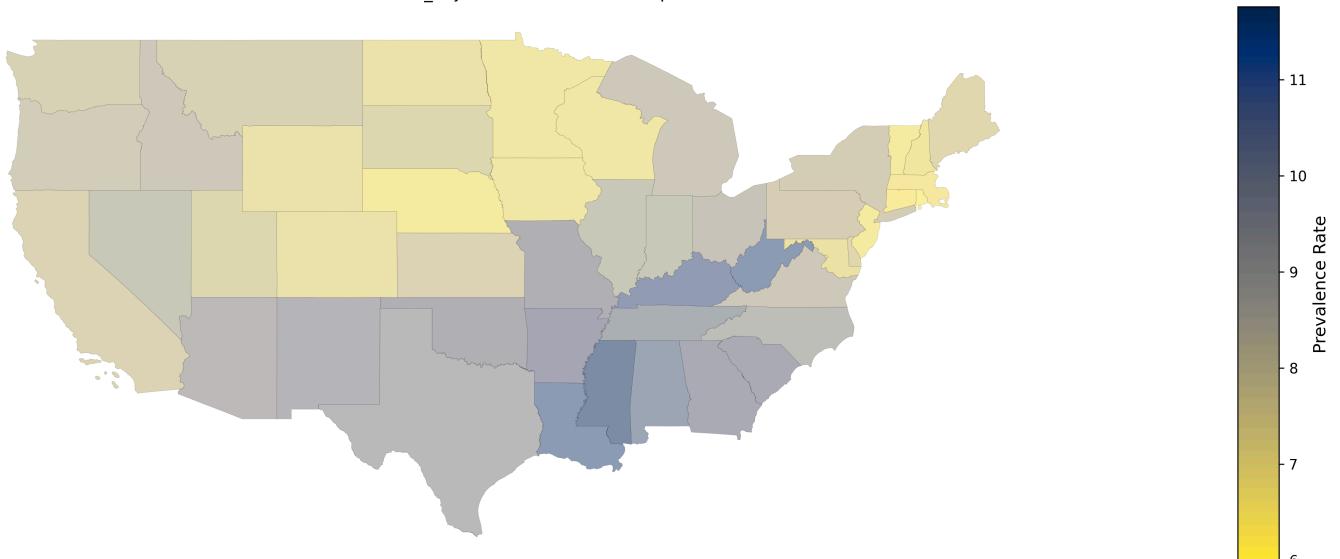
MOBILITY\_AdjPrev - United States Map

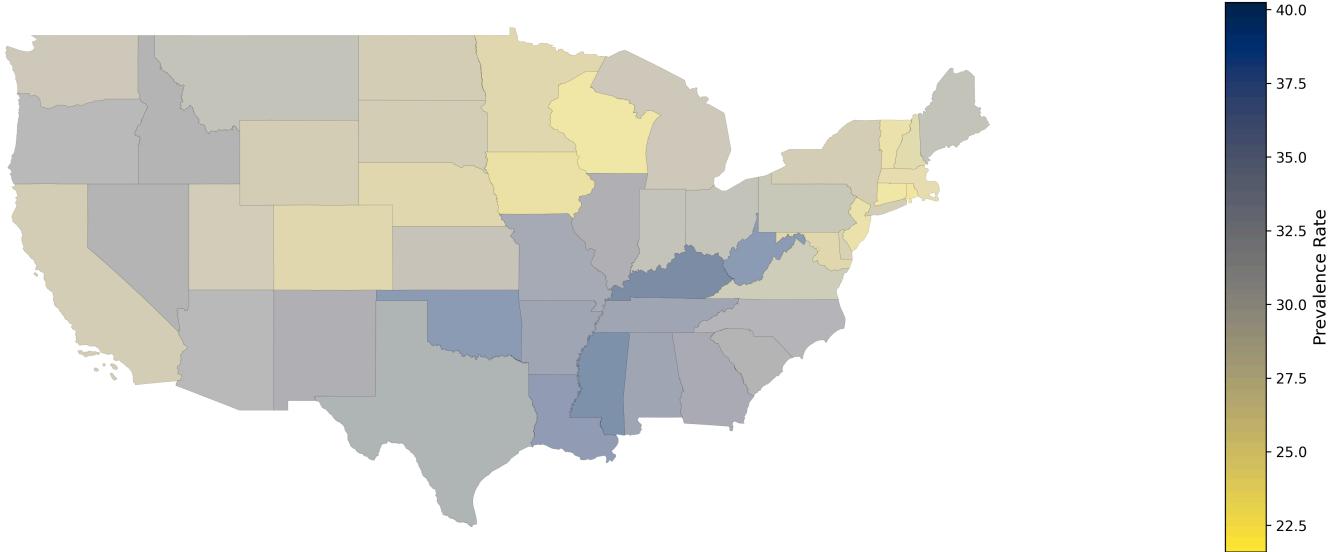


SELCARE\_AdjPrev - United States Map



INDEPLIVE\_AdjPrev - United States Map





## A Focus on Illinois State

### Filter by the State of Illinois

```
In [ ]: df_il = df[df['StateAbbr']=='IL']
```

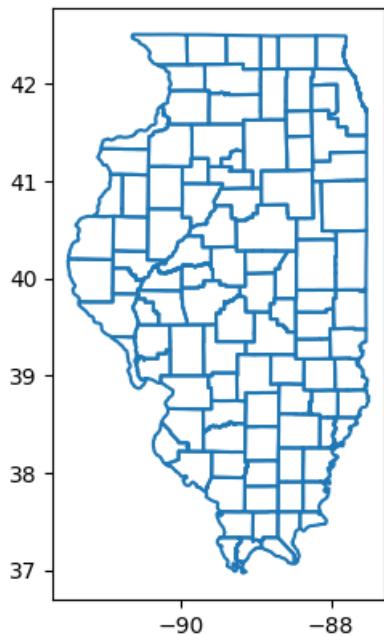
### Import Illinois State Shapefile

```
In [ ]: path = "C:/Users/user/Desktop/GIS/Final Project/IL_BNDY_County/IL_BNDY_County_Py.shp"
county = gpd.read_file(path)
county = county.to_crs("EPSG:4326")
county
```

	COUNTY_NAM	CO_FIPS	geometry
0	MCHENRY	111	POLYGON ((-88.70742 42.49351, -88.68809 42.493...
1	BOONE	7	POLYGON ((-88.70742 42.49351, -88.70740 42.493...
2	OGLE	141	POLYGON ((-89.68808 42.19949, -89.66846 42.200...
3	WILL	197	POLYGON ((-88.26146 41.72439, -88.26146 41.724...
4	LASALLE	99	POLYGON ((-88.93885 41.62836, -88.93871 41.628...
...	...	...	...
97	JEFFERSON	81	POLYGON ((-89.14445 38.47386, -89.12638 38.473...
98	LAWRENCE	101	POLYGON ((-87.90805 38.85012, -87.88944 38.849...
99	MARION	121	POLYGON ((-89.13843 38.73632, -89.13827 38.750...
100	UNION	181	POLYGON ((-89.04143 37.59649, -89.04143 37.596...
101	POPE	151	POLYGON ((-88.70860 37.59925, -88.69093 37.599...

102 rows × 3 columns

```
In [ ]: county.boundary.plot()
plt.show()
```



## Preprocess and Merge

```
In [ ]: df_il['CountyName'] = df_il['CountyName'].str.upper()
df_il.head(2)
```

C:\Users\user\AppData\Local\Temp\ipykernel\_17396\1893931094.py:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df_il['CountyName'] = df_il['CountyName'].str.upper()
```

```
Out[ ]:   StateAbbr StateDesc CountyName CountyFIPS TotalPopulation ACCESS2_CrudePrev ACCESS2_Crude95CI ACCESS2_AdjPrev ACCESS2_Adj95CI
          62       IL      Illinois      BOND        17005           16596            9.0    ( 6.9, 11.7)           9.4    ( 7.3, 11.7)
          63       IL      Illinois     CALHOUN        17013           4369             7.0    ( 5.1, 9.1)           7.7    ( 5.7, 10.7)
```

2 rows × 154 columns

```
In [ ]: # Assuming df2 is your DataFrame
df_il.loc[df_il['CountyName'] == 'DE WITT', 'CountyName'] = 'DEWITT'
```

```
In [ ]: # Merge aggregated DataFrame with the GeoDataFrame
merged_gdf = county.merge(df_il, left_on='COUNTY_NAM', right_on='CountyName', how='left')
merged_gdf.head(2)
```

```
Out[ ]:   COUNTY_NAM CO_FIPS geometry StateAbbr StateDesc CountyName CountyFIPS TotalPopulation ACCESS2_CrudePrev ACCESS2_Crude95CI
          0      MCHENRY    111  POLYGON((-88.70742, 42.49351, -88.68809, 42.493...  IL      Illinois      MCHENRY        17111         311122            7.4    ( 5.7, 10.7)
          1      BOONE       7  POLYGON((-88.70742, 42.49351, -88.70740, 42.493...  IL      Illinois      BOONE         17007          53159            10.6   ( 8.1, 12.1)
```

2 rows × 157 columns

## Plotting

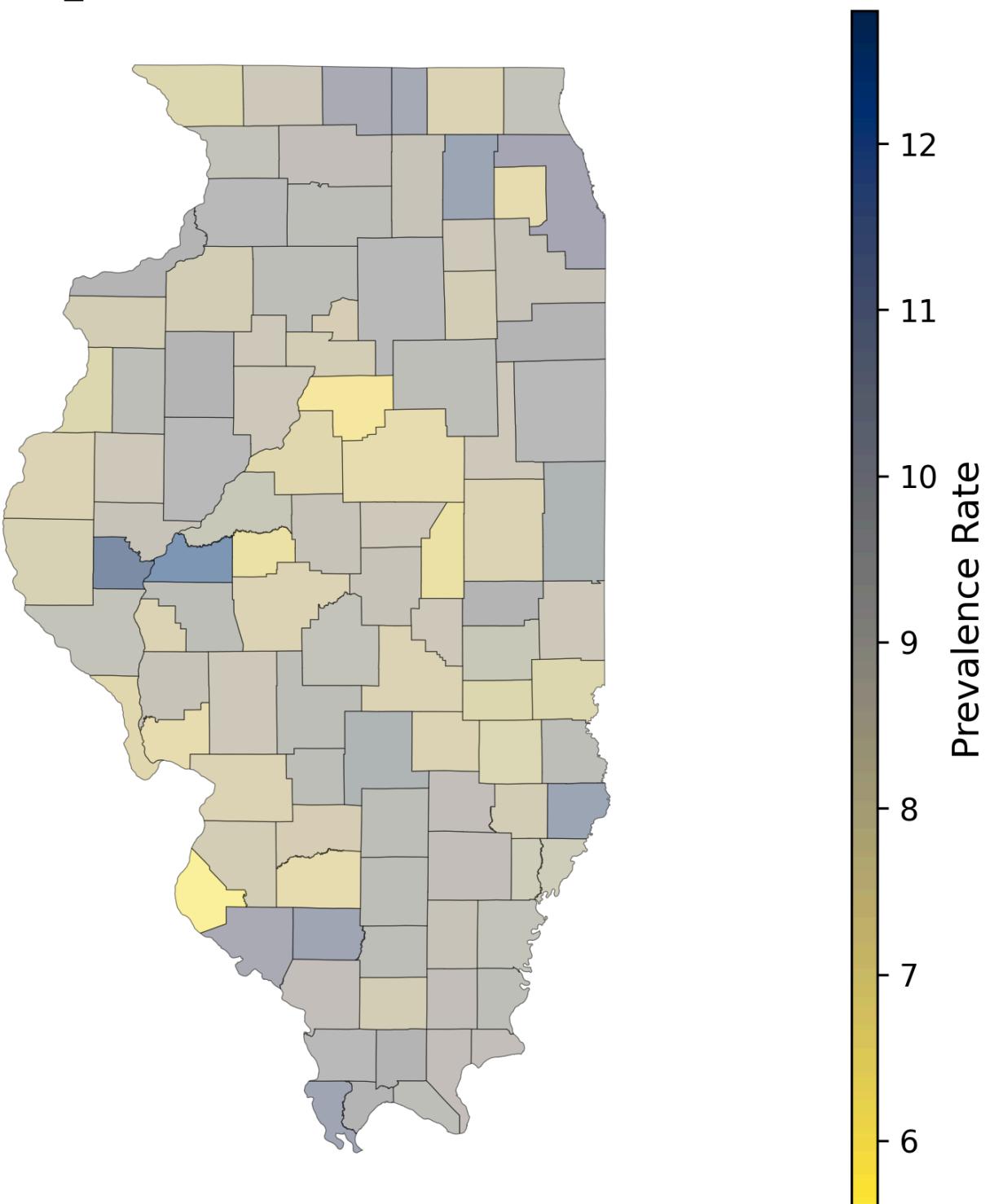
```
In [ ]: import matplotlib.pyplot as plt
from mpl_toolkits.axes_grid1 import make_axes_locatable

def plot_normalized_county_data(geo_df, data_columns, colormap):
    for column in data_columns:
        # Plotting
        f, ax = plt.subplots(1, 1, figsize=(10, 6), sharex=True, sharey=True, dpi=300)
        f.tight_layout()
        plt.title(f'{column} - Illinois Map by County')
        ax.set_axis_off()
        divider = make_axes_locatable(ax)
        cax = divider.append_axes("right", size="3%", pad=0.5, alpha=0.5)
        geo_df.plot(column, ax=ax, alpha=0.5, cmap=colormap, edgecolor='k', legend=True, cax=cax, linewidth=0.3)
        plt.ylabel('Prevalence Rate', fontsize=12)
    plt.show()
```

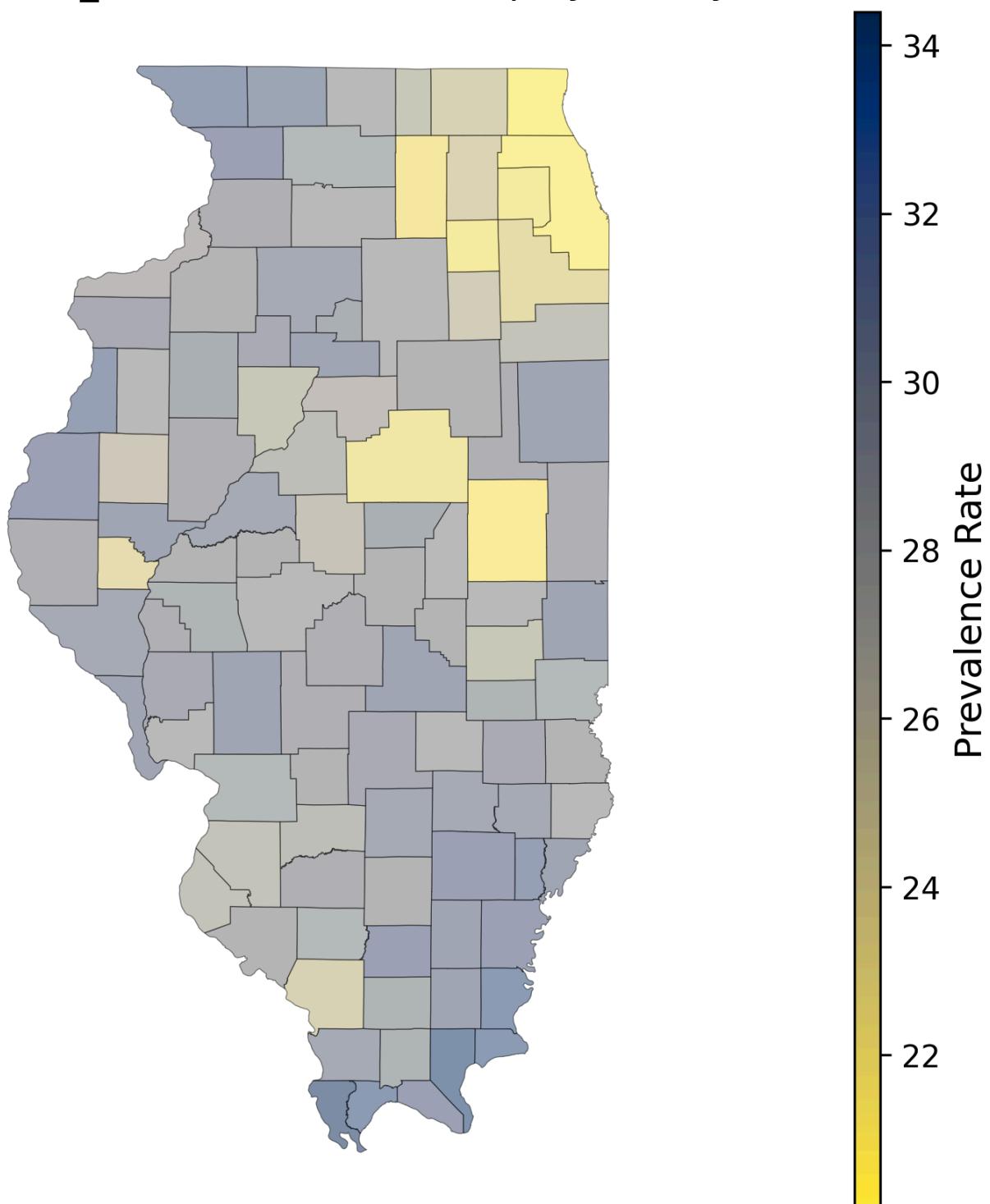
```
In [ ]: # Get columns ending with '_CrudePrev'
columns_to_plot = [col for col in df.columns if col.endswith('_CrudePrev')]

plot_normalized_county_data(merged_gdf, columns_to_plot, 'cividis_r')
```

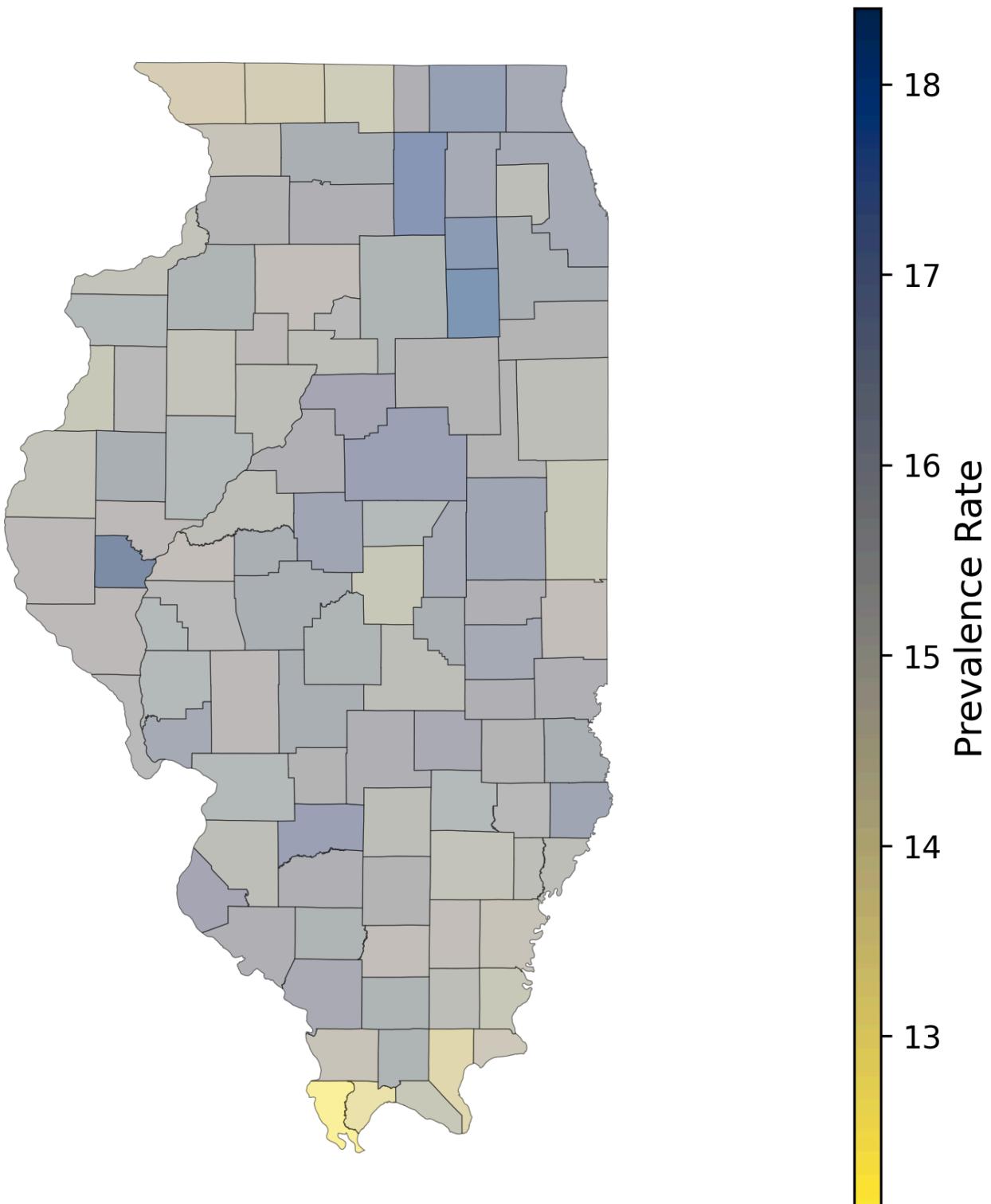
# ACCESS2\_CrudePrev - Illinois Map by County



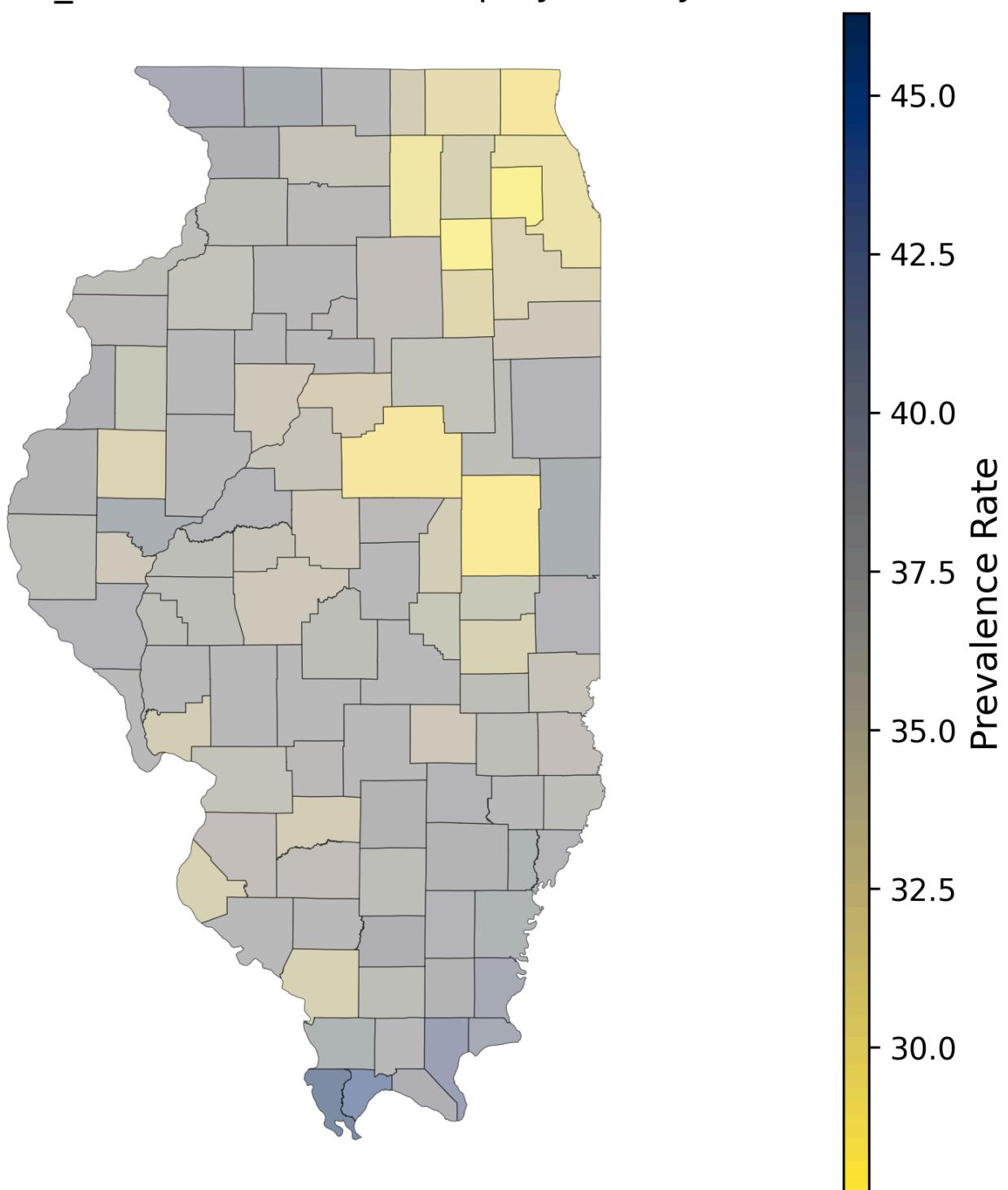
# ARTHRITIS\_CrudePrev - Illinois Map by County



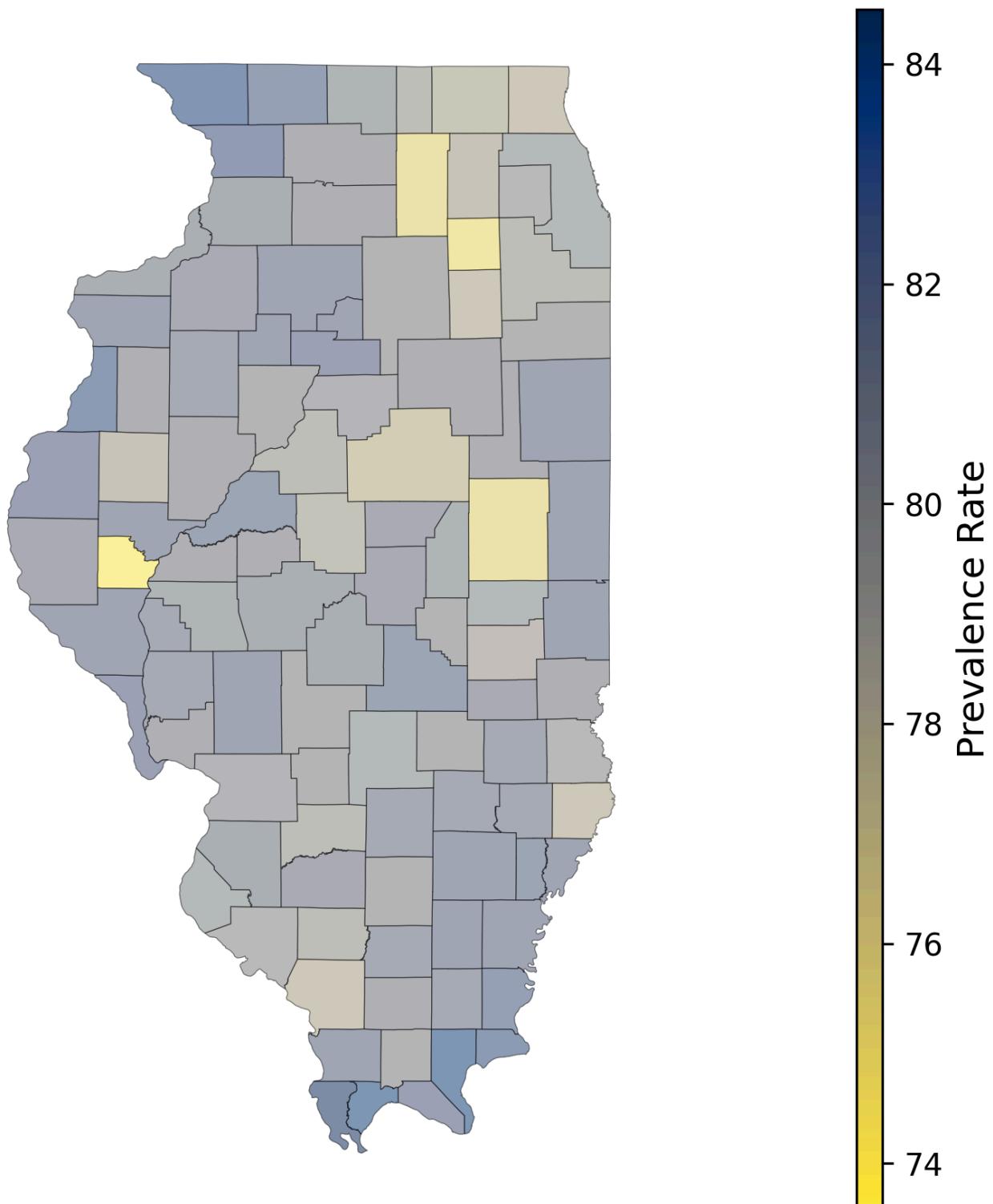
# BINGE\_CrudePrev - Illinois Map by County



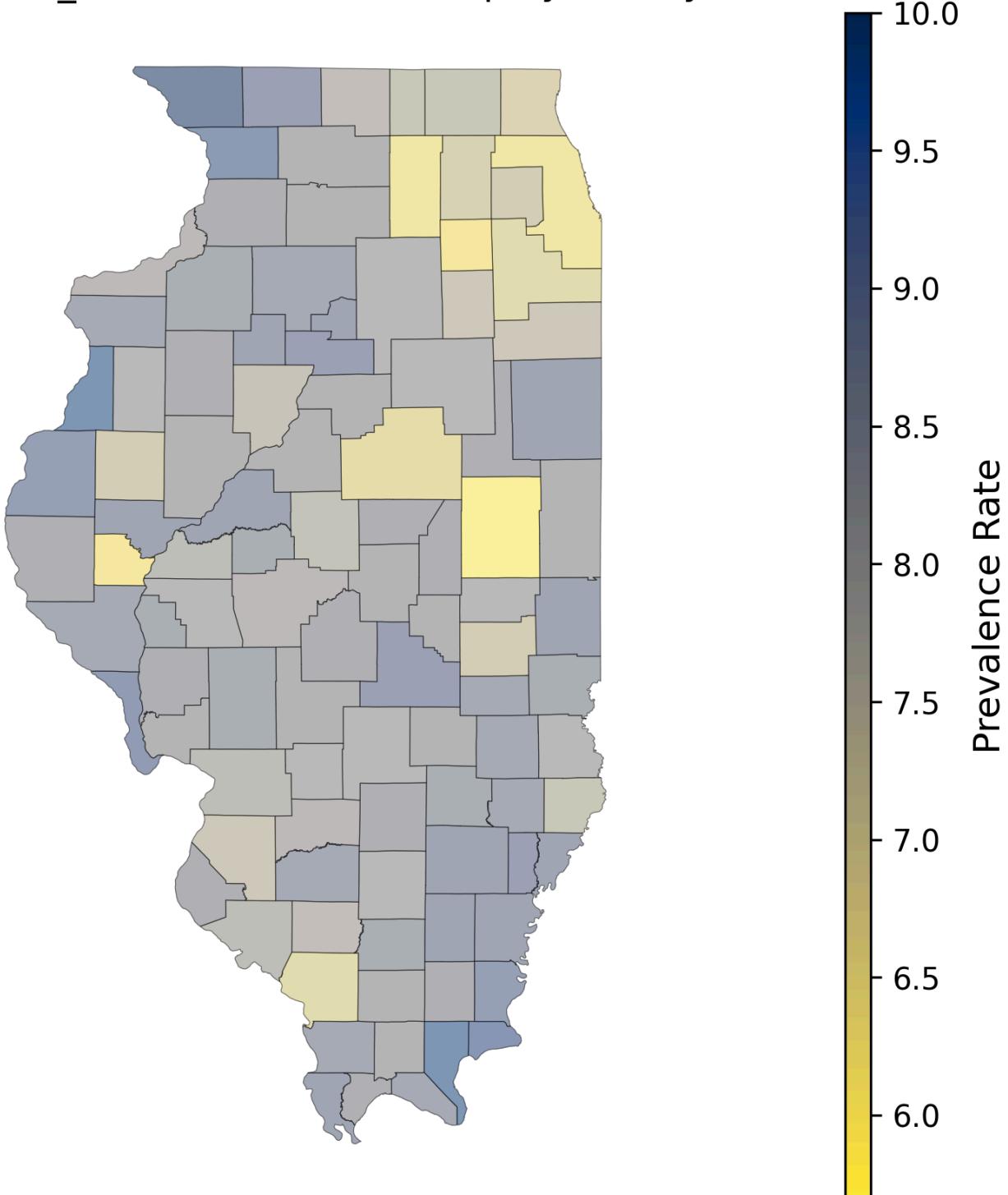
# BPHIGH\_CrudePrev - Illinois Map by County



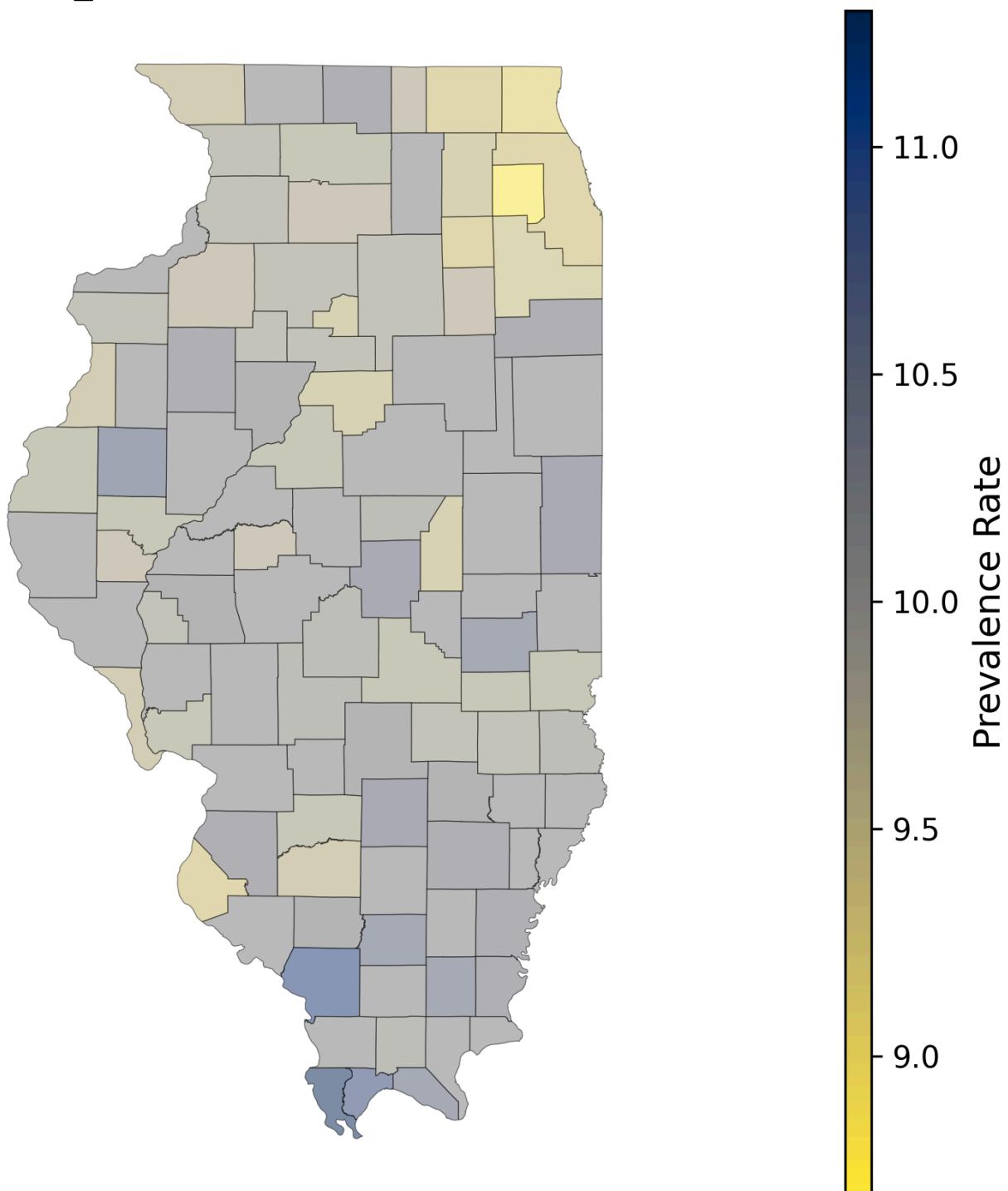
# BPMED\_CrudePrev - Illinois Map by County



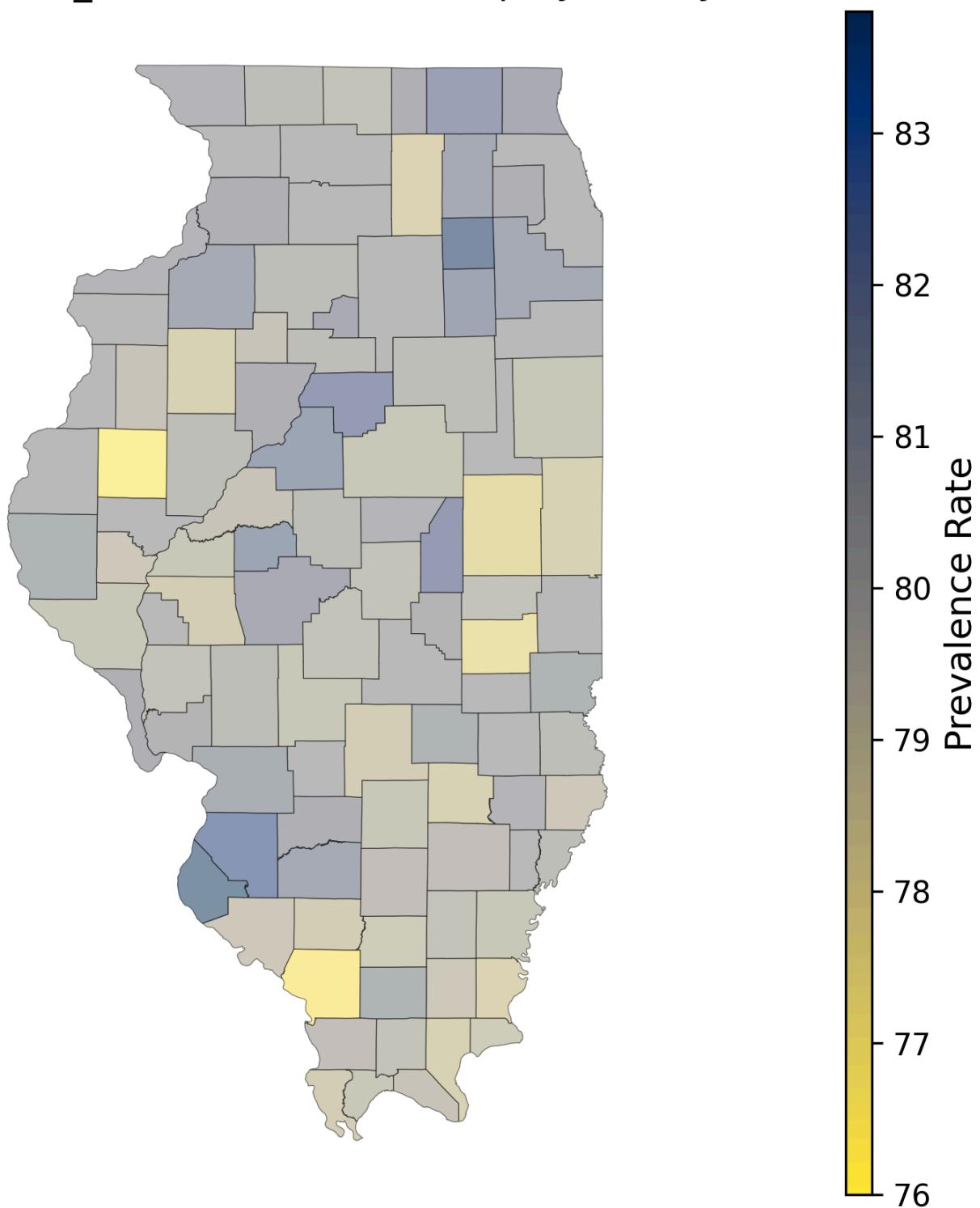
# CANCER\_CrudePrev - Illinois Map by County



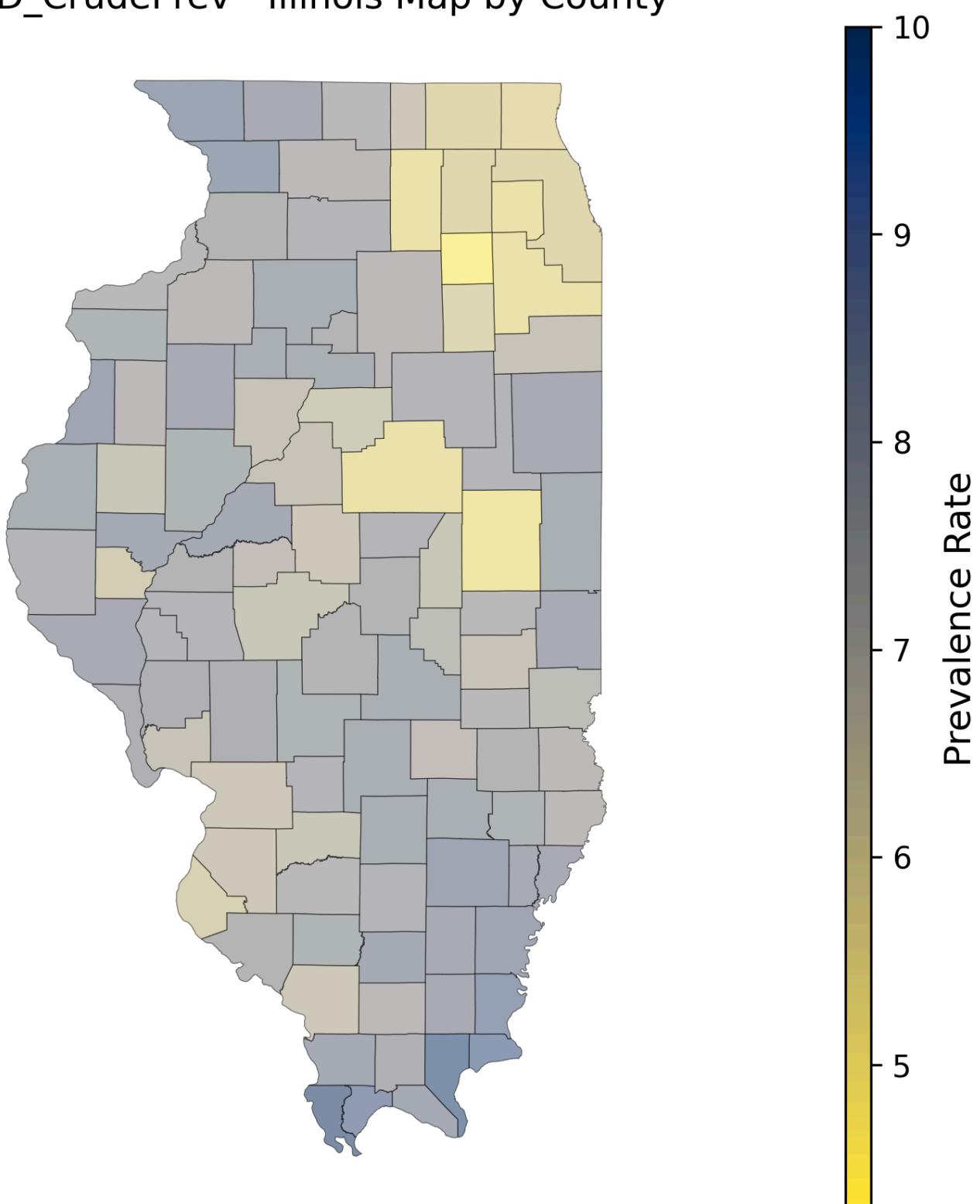
# CASTHMA\_CrudePrev - Illinois Map by County



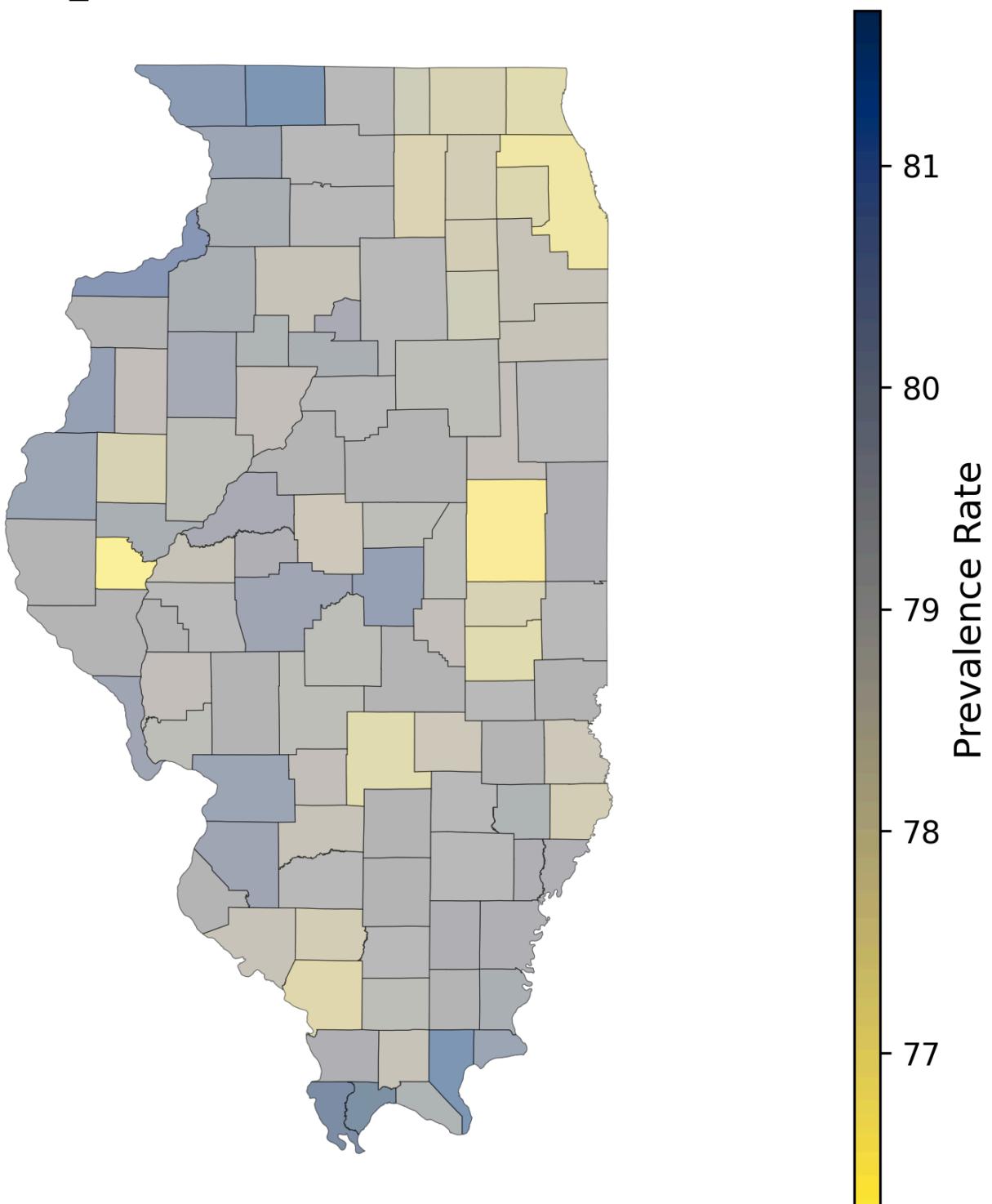
# CERVICAL\_CrudePrev - Illinois Map by County



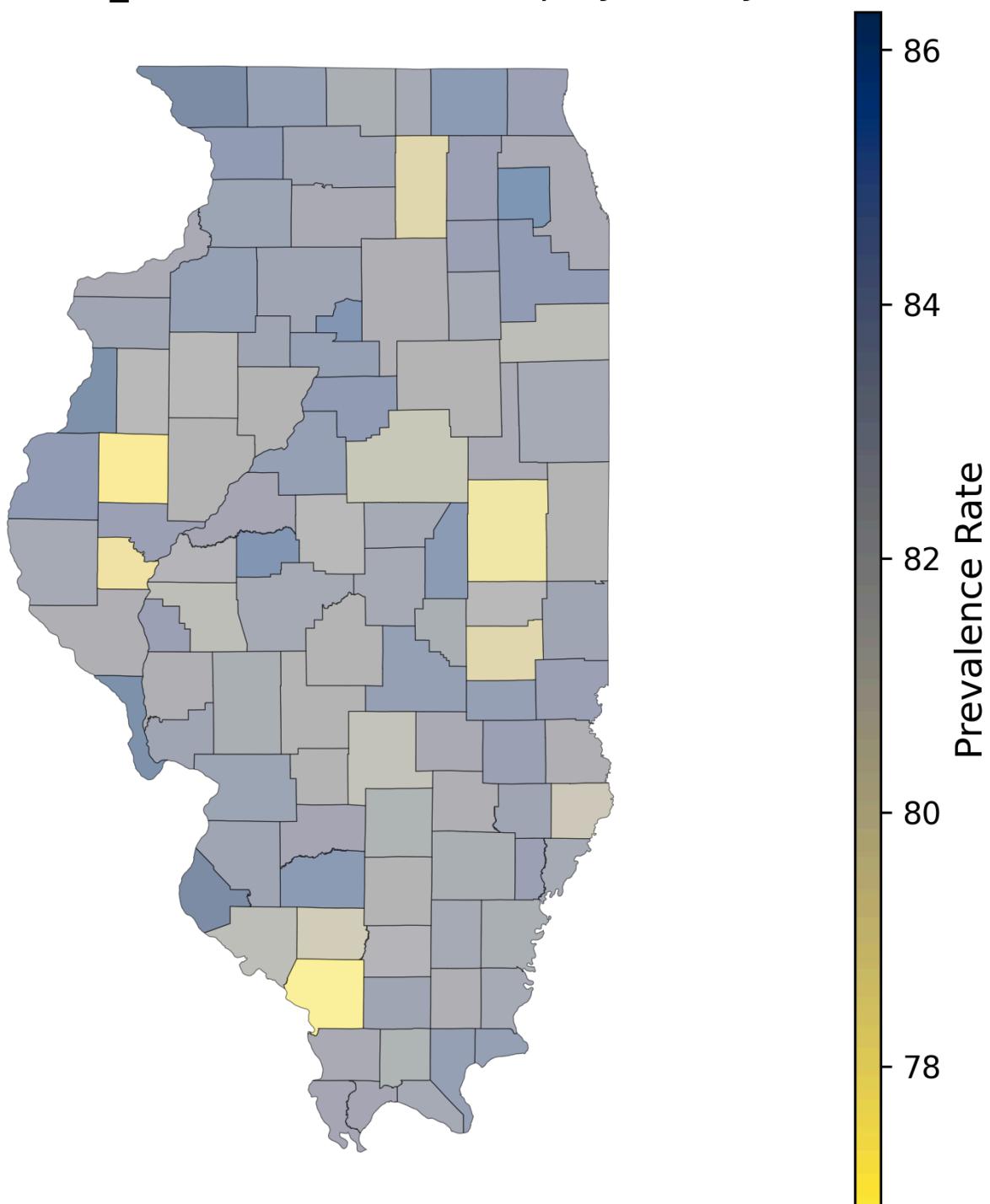
# CHD\_CrudePrev - Illinois Map by County



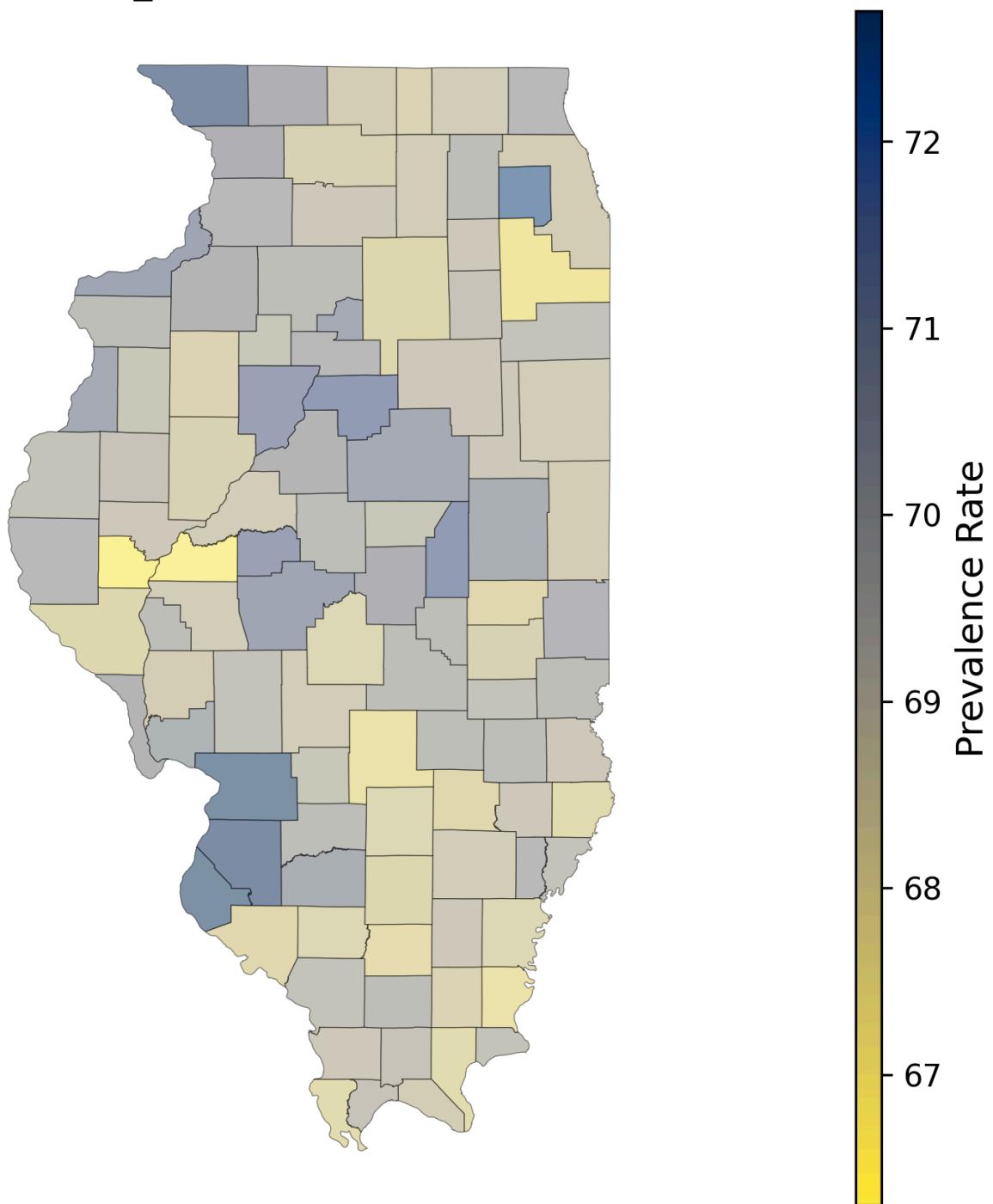
# CHECKUP\_CrudePrev - Illinois Map by County



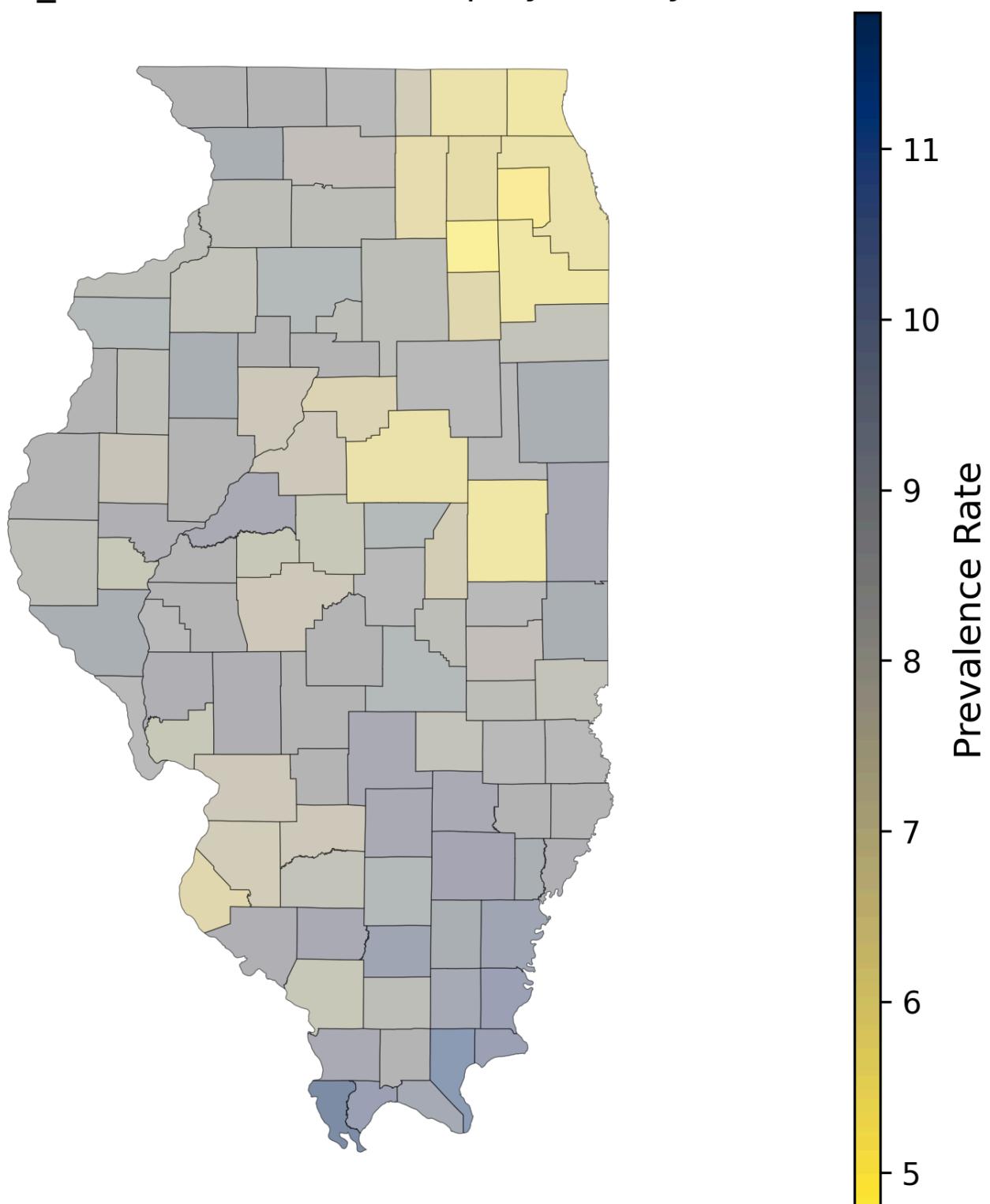
# CHOLSCREEN\_CrudePrev - Illinois Map by County



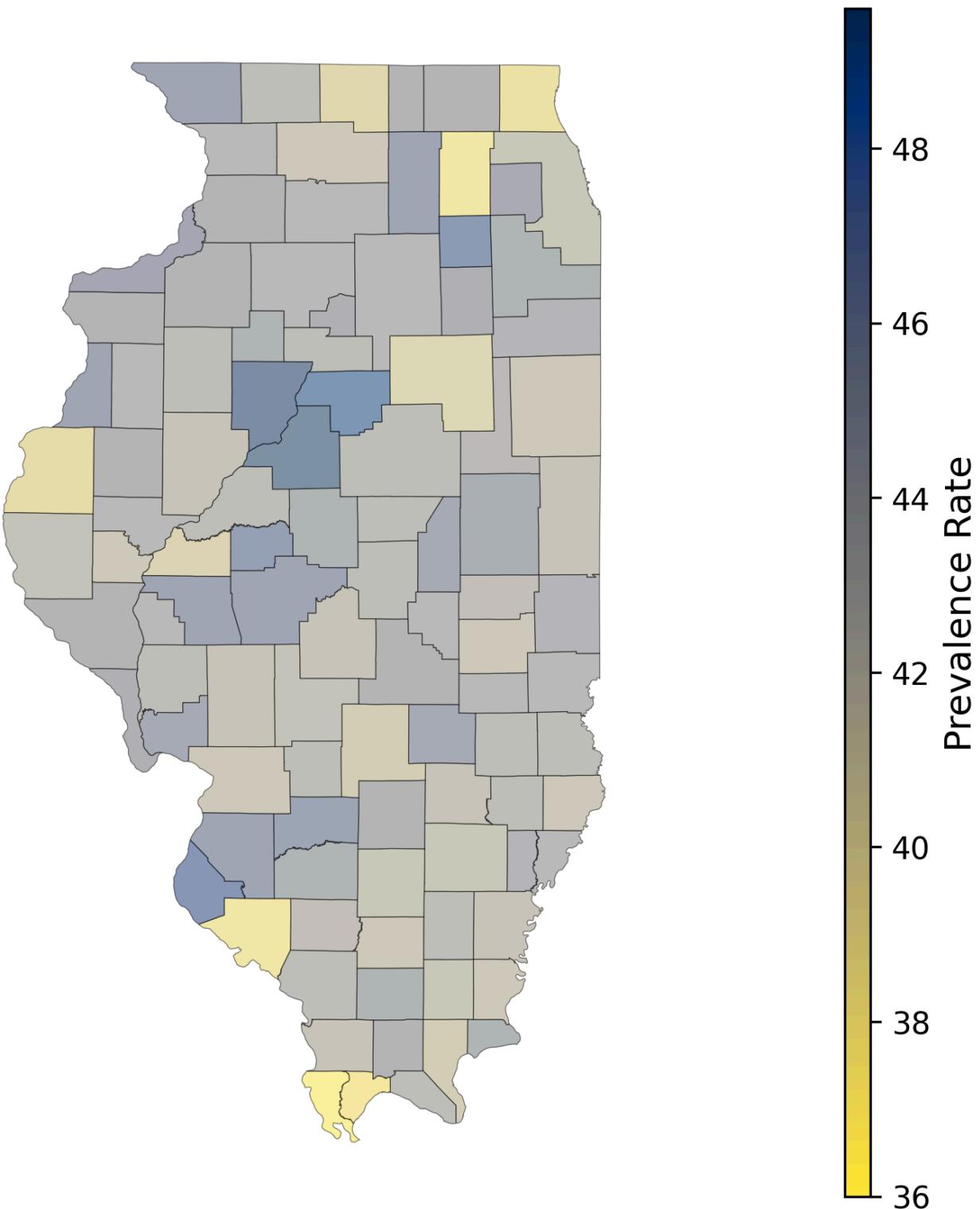
# COLON\_SCREEN\_CrudePrev - Illinois Map by County



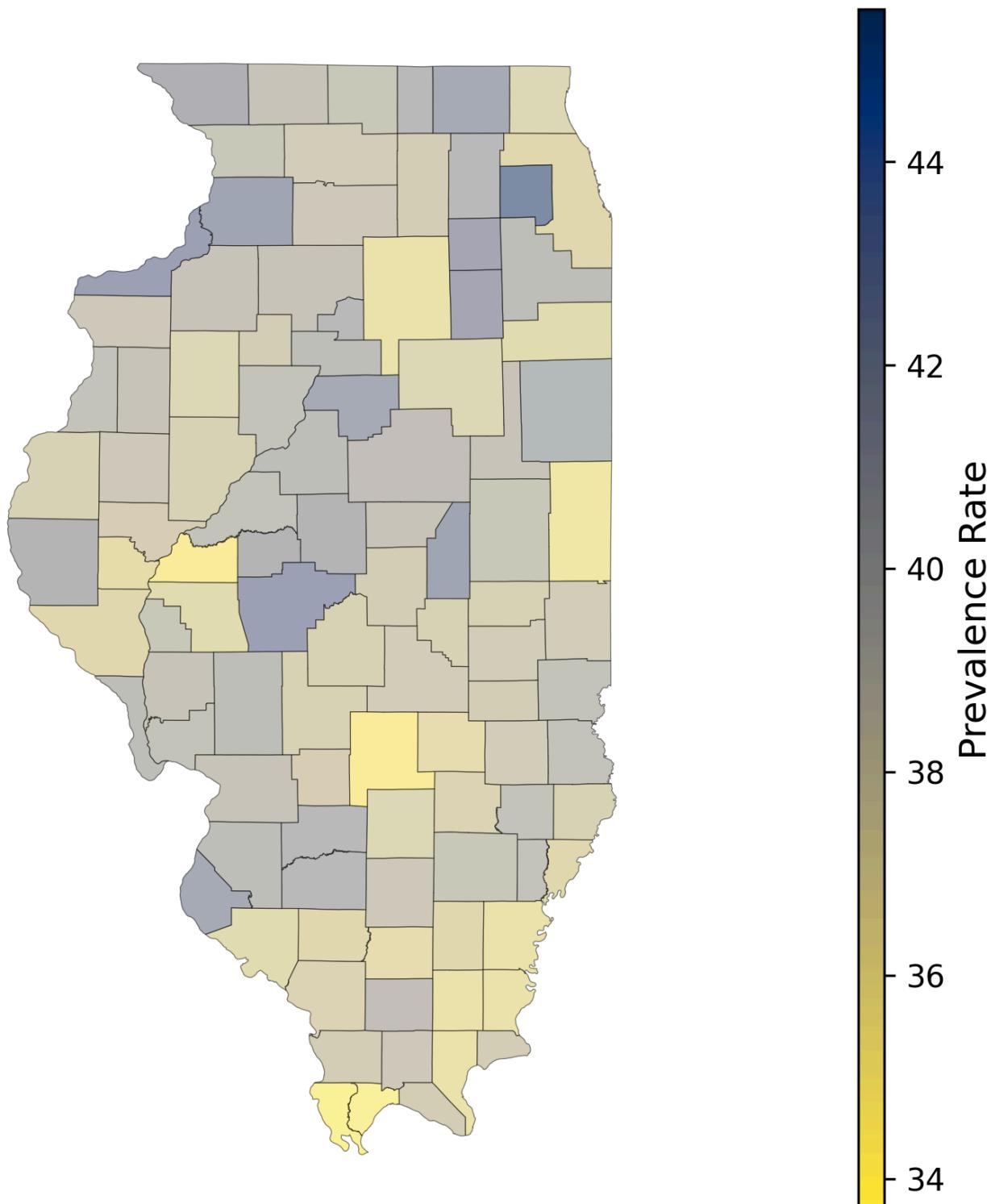
# COPD\_CrudePrev - Illinois Map by County



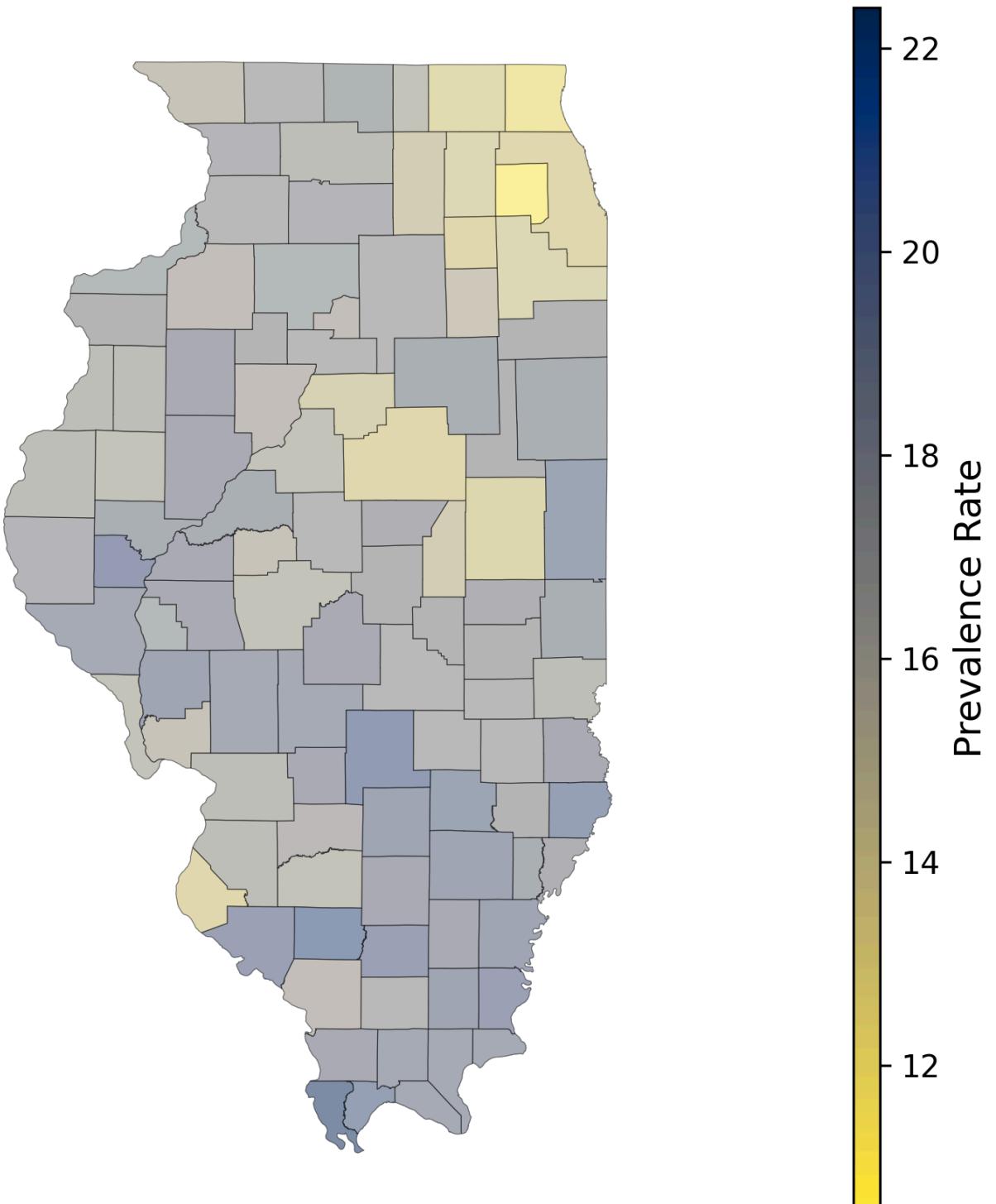
# COREM\_CrudePrev - Illinois Map by County



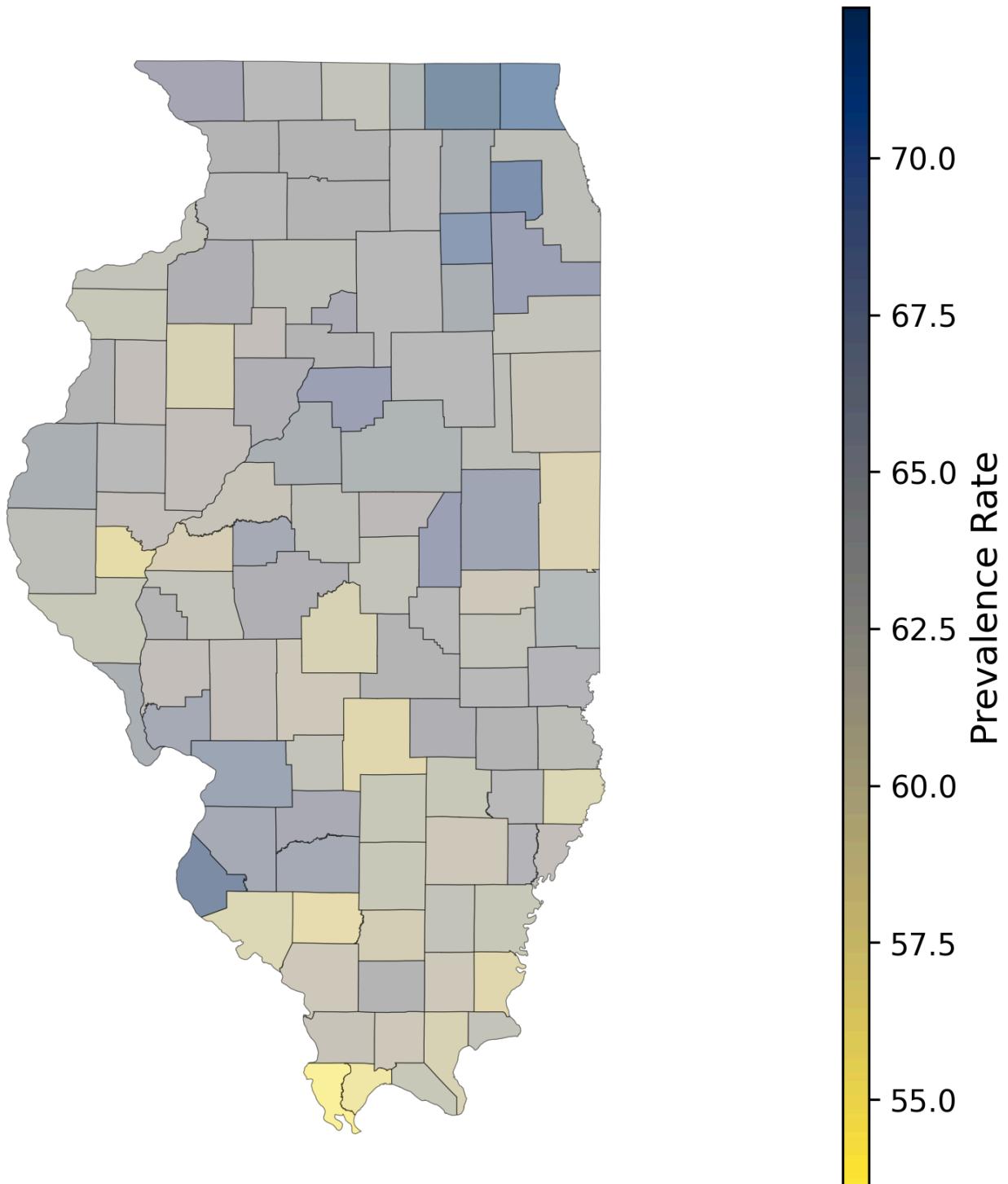
# COREW\_CrudePrev - Illinois Map by County



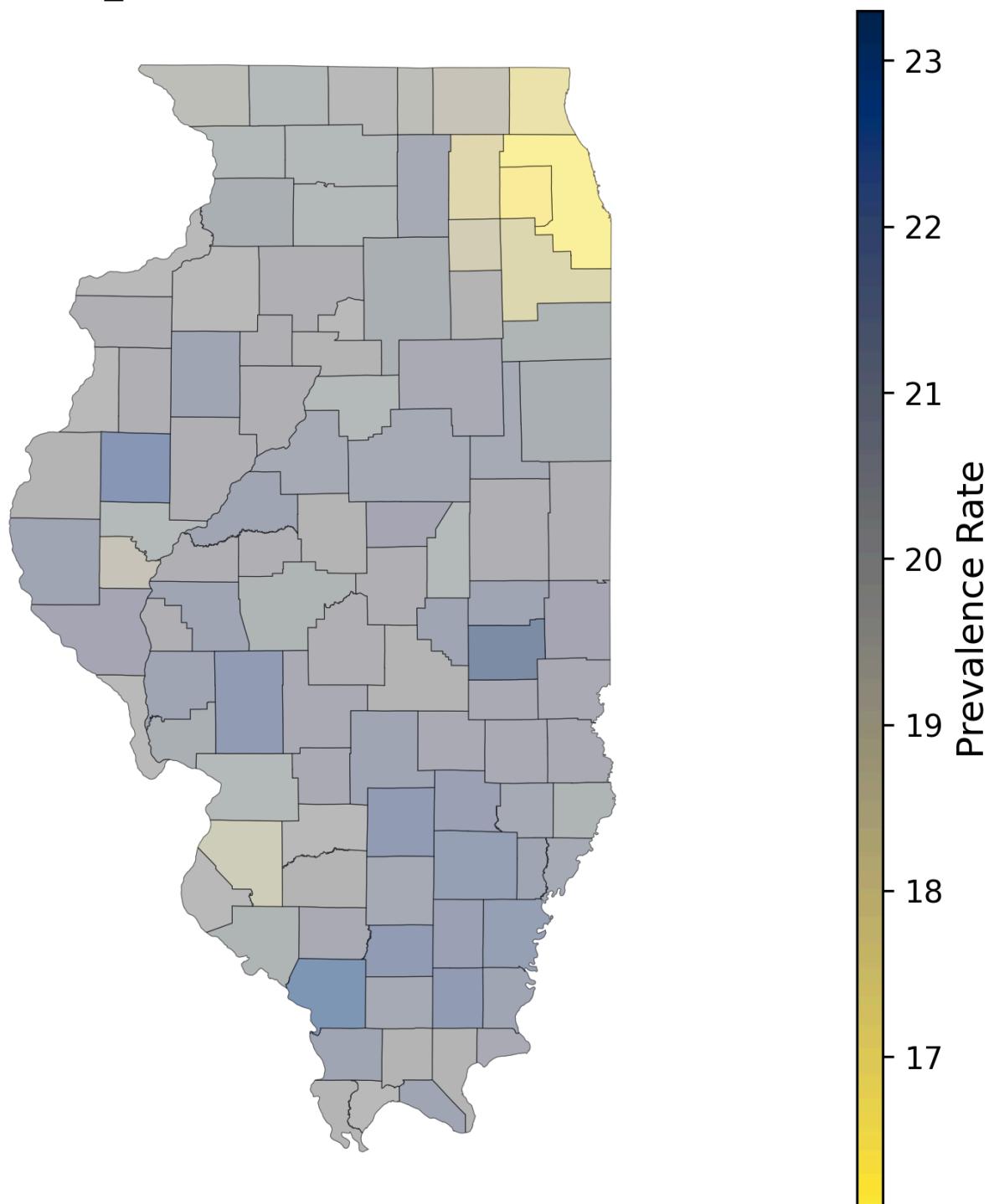
# CSMOKING\_CrudePrev - Illinois Map by County



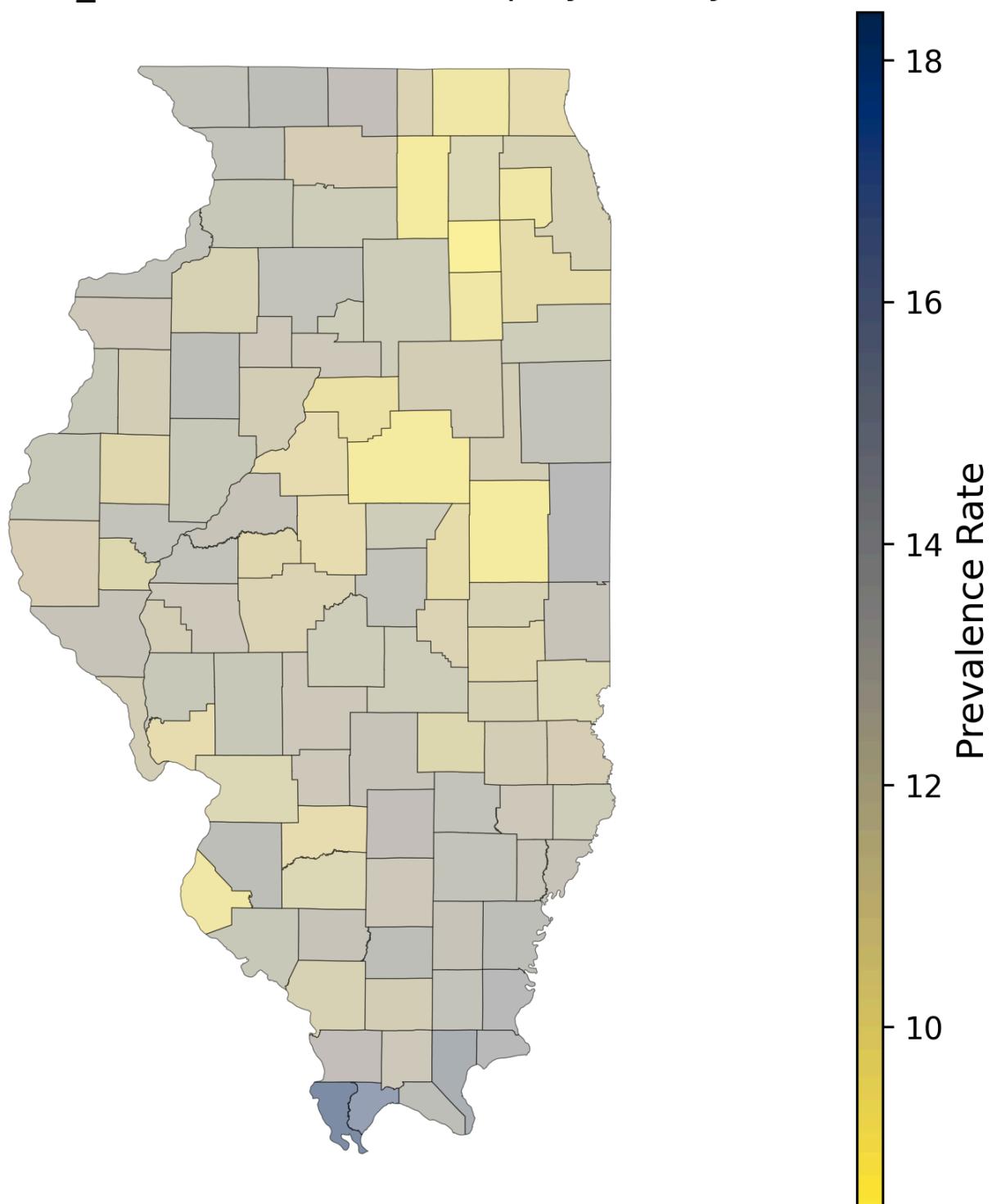
DENTAL\_CrudePrev - Illinois Map by County



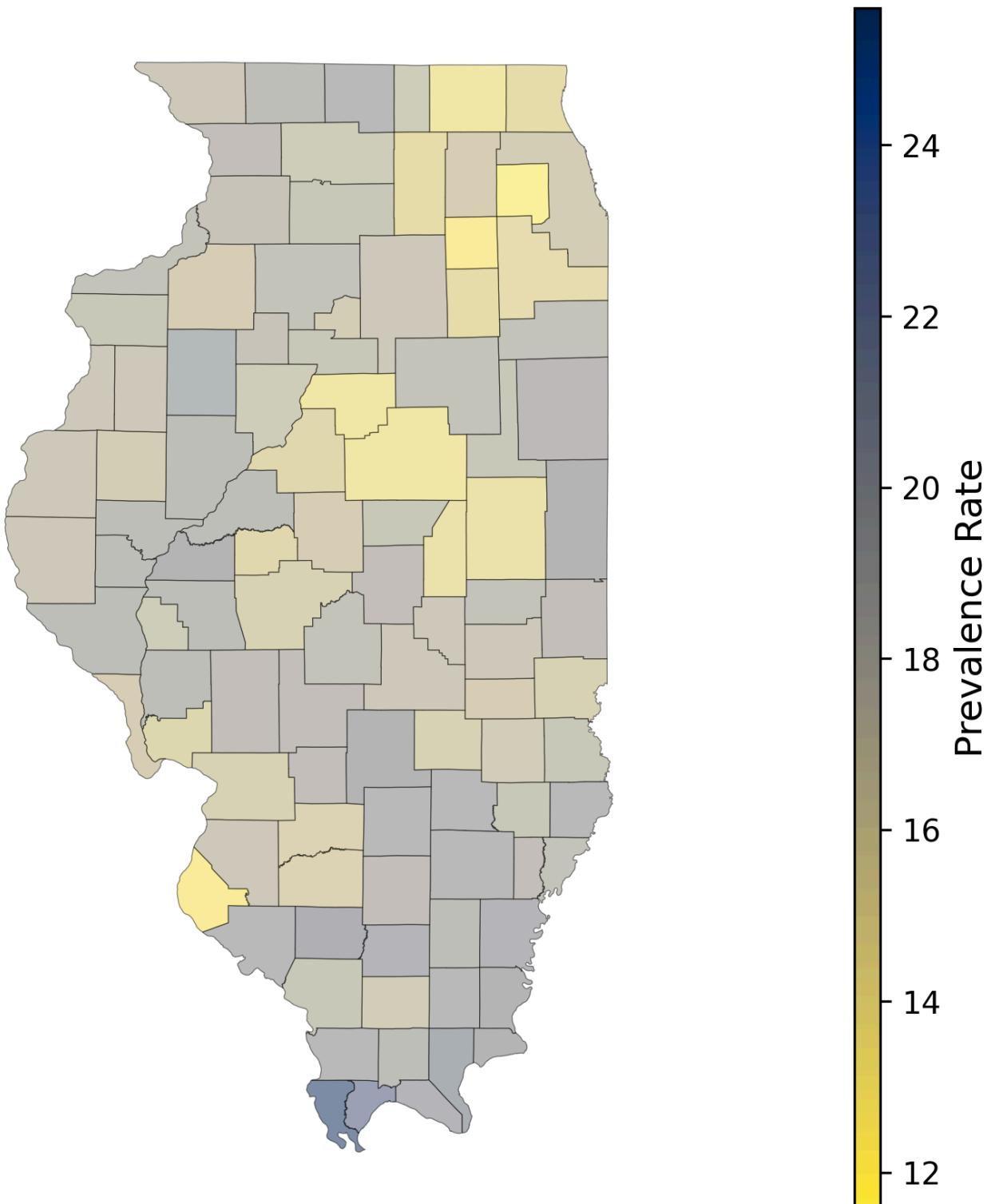
# DEPRESSION\_CrudePrev - Illinois Map by County



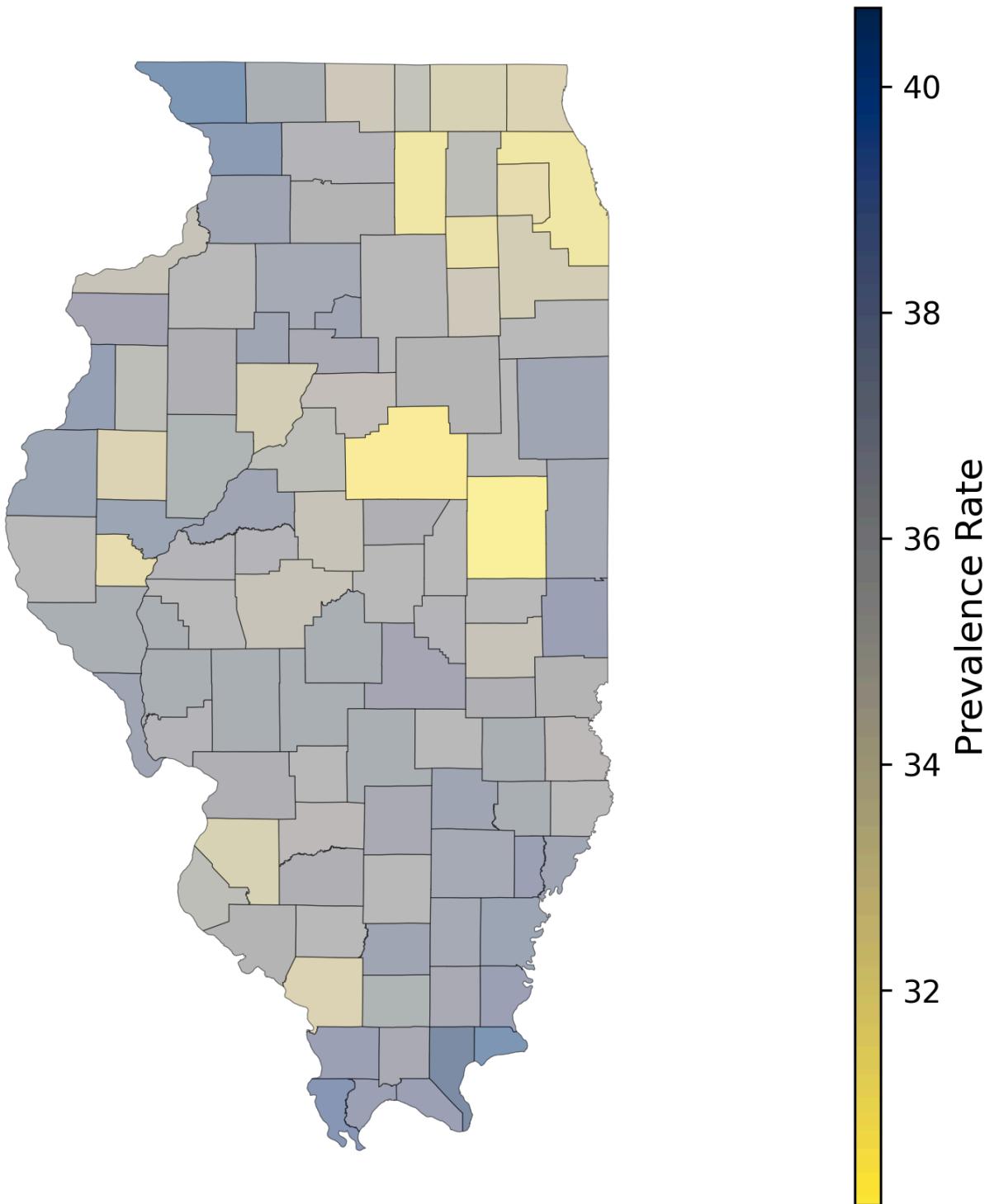
# DIABETES\_CrudePrev - Illinois Map by County



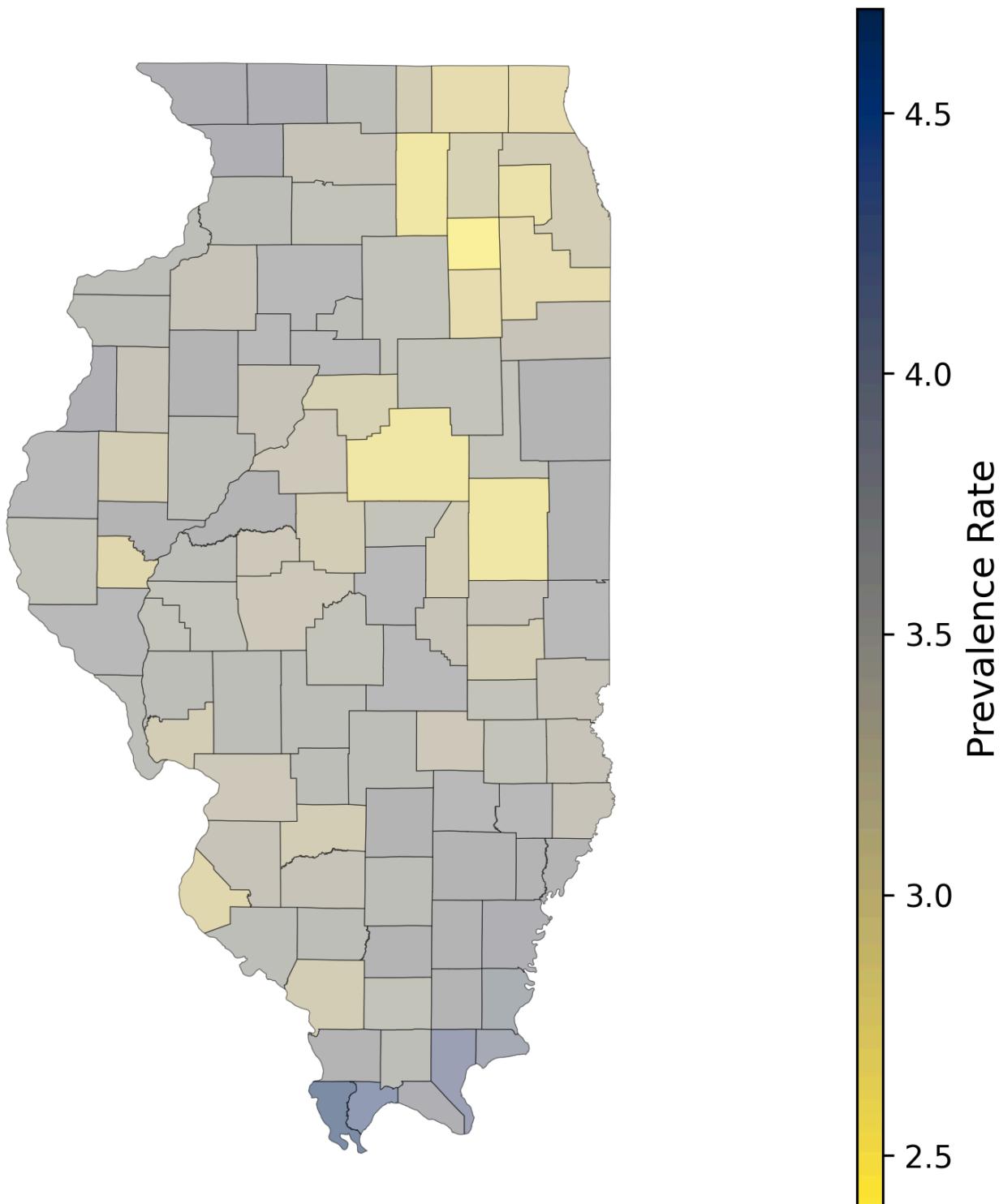
# GHLTH\_CrudePrev - Illinois Map by County



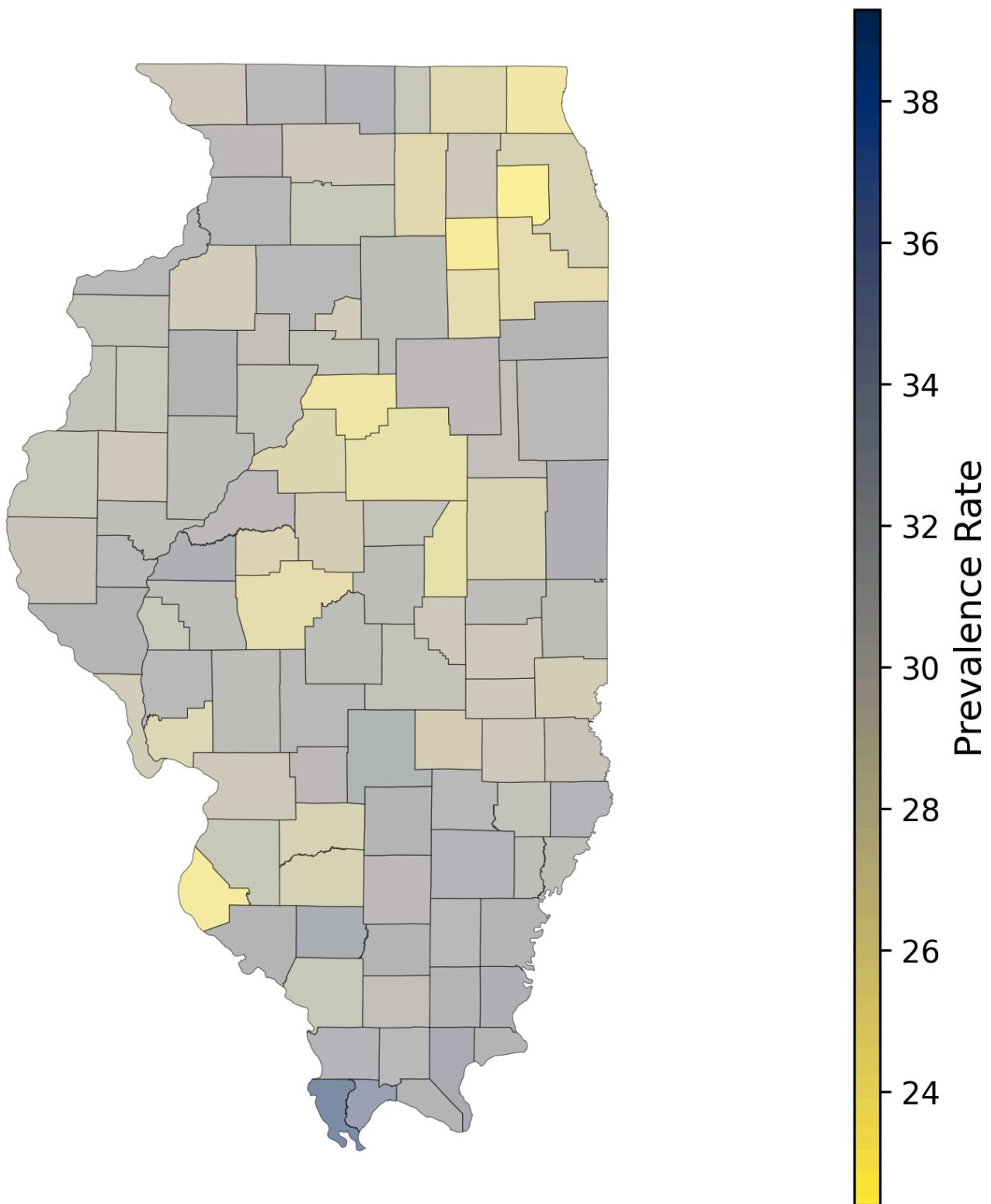
# HIGHCHOL\_CrudePrev - Illinois Map by County



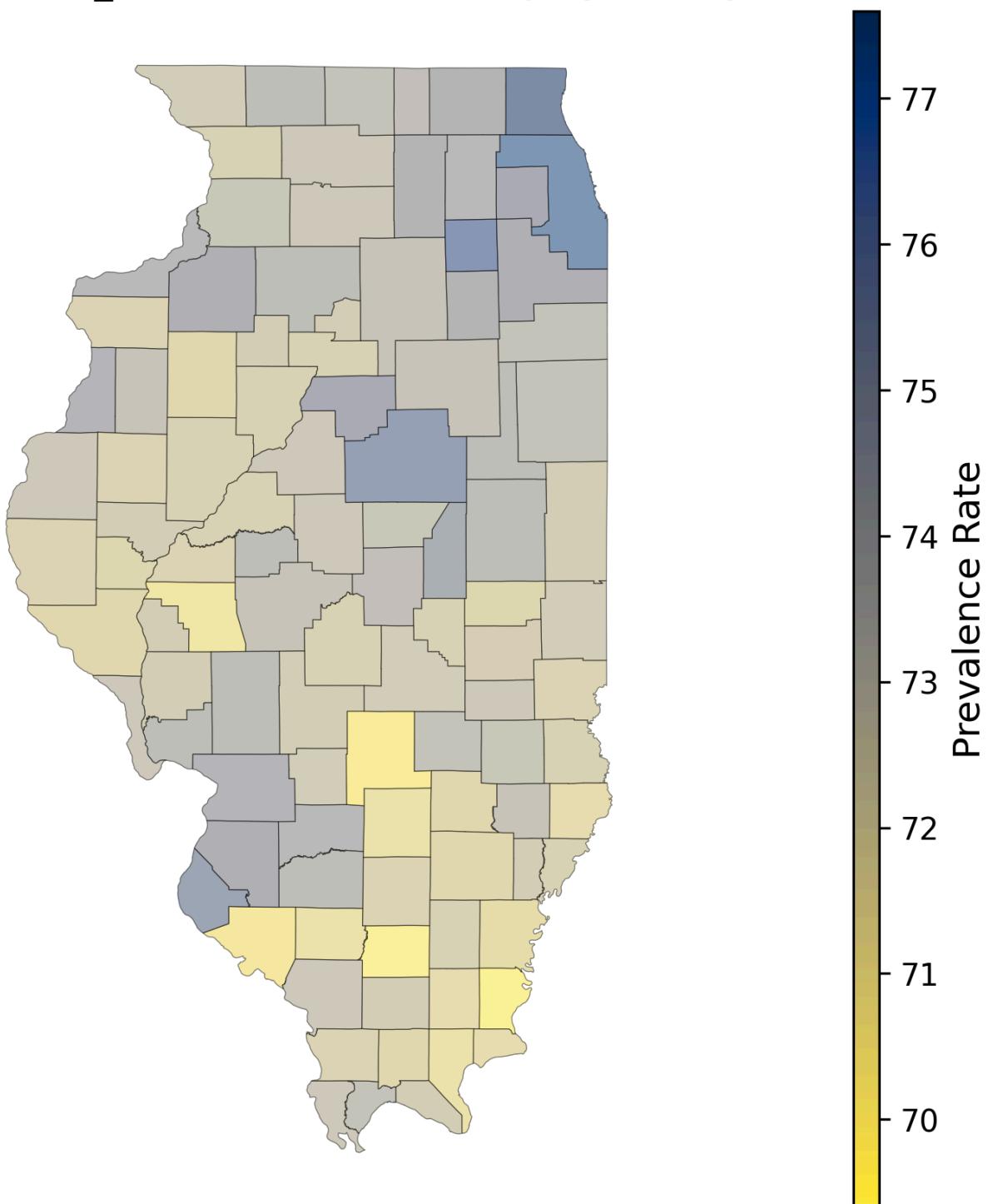
# KIDNEY\_CrudePrev - Illinois Map by County



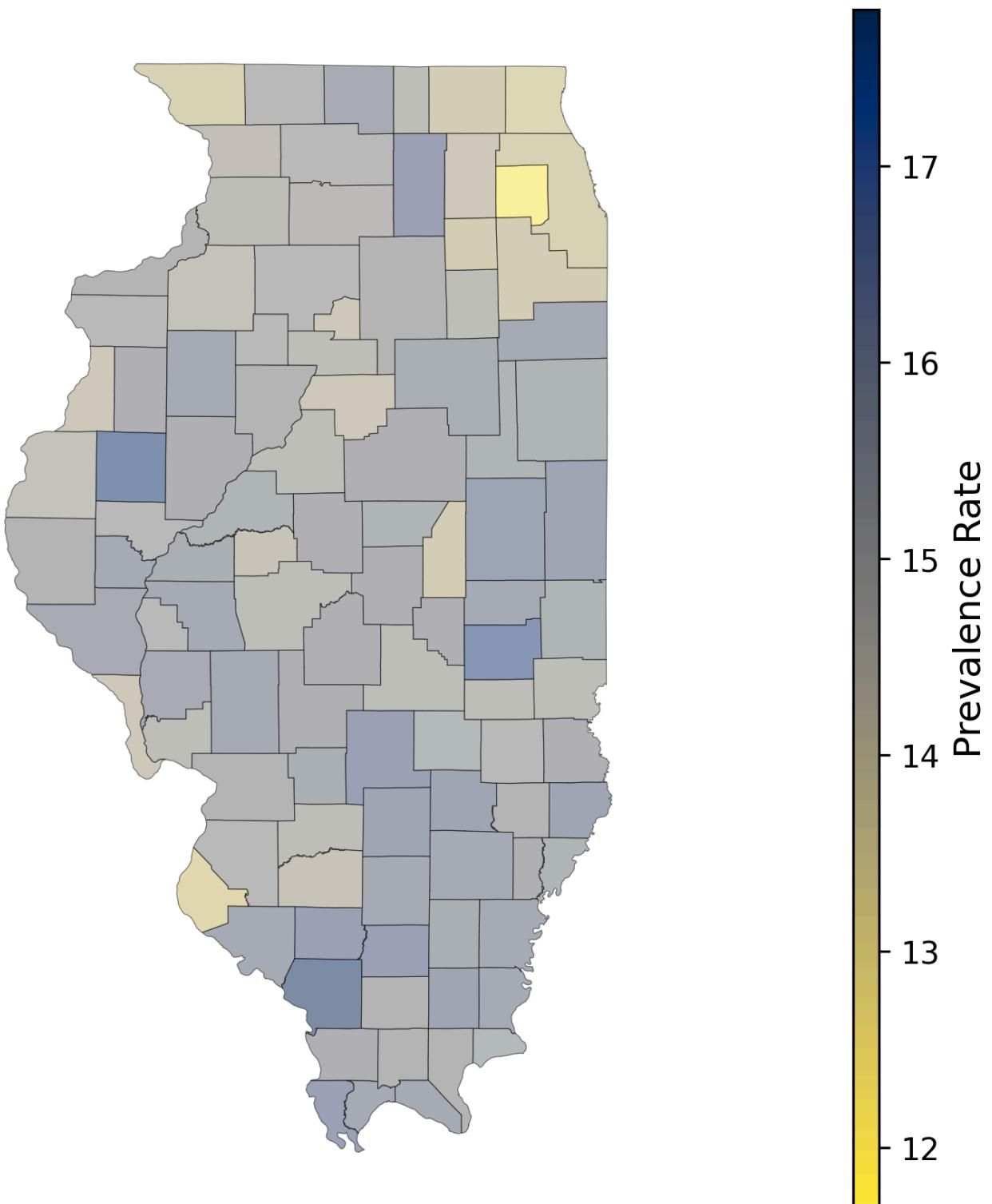
# LPA\_CrudePrev - Illinois Map by County



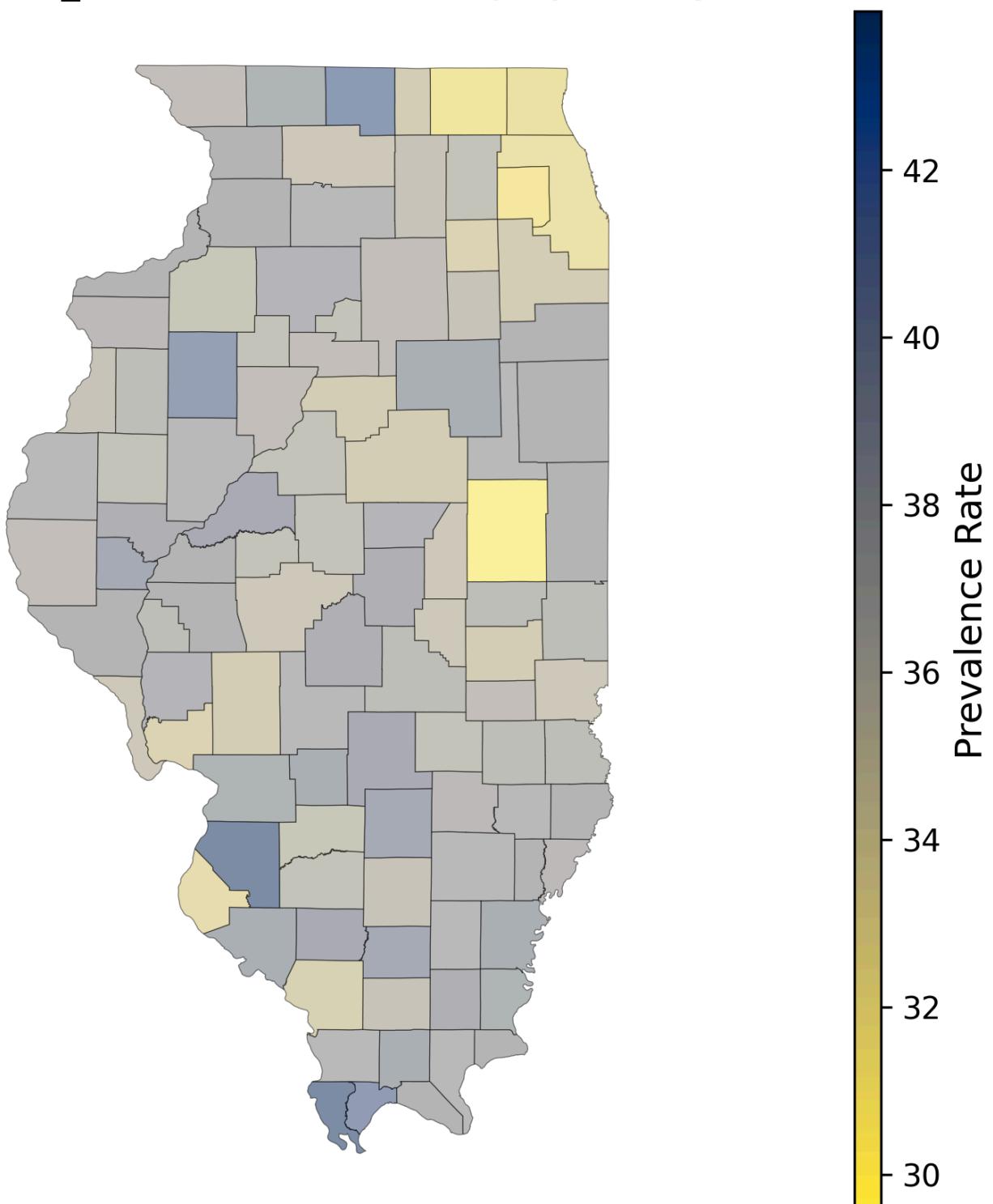
# MAMMOUSE\_CrudePrev - Illinois Map by County



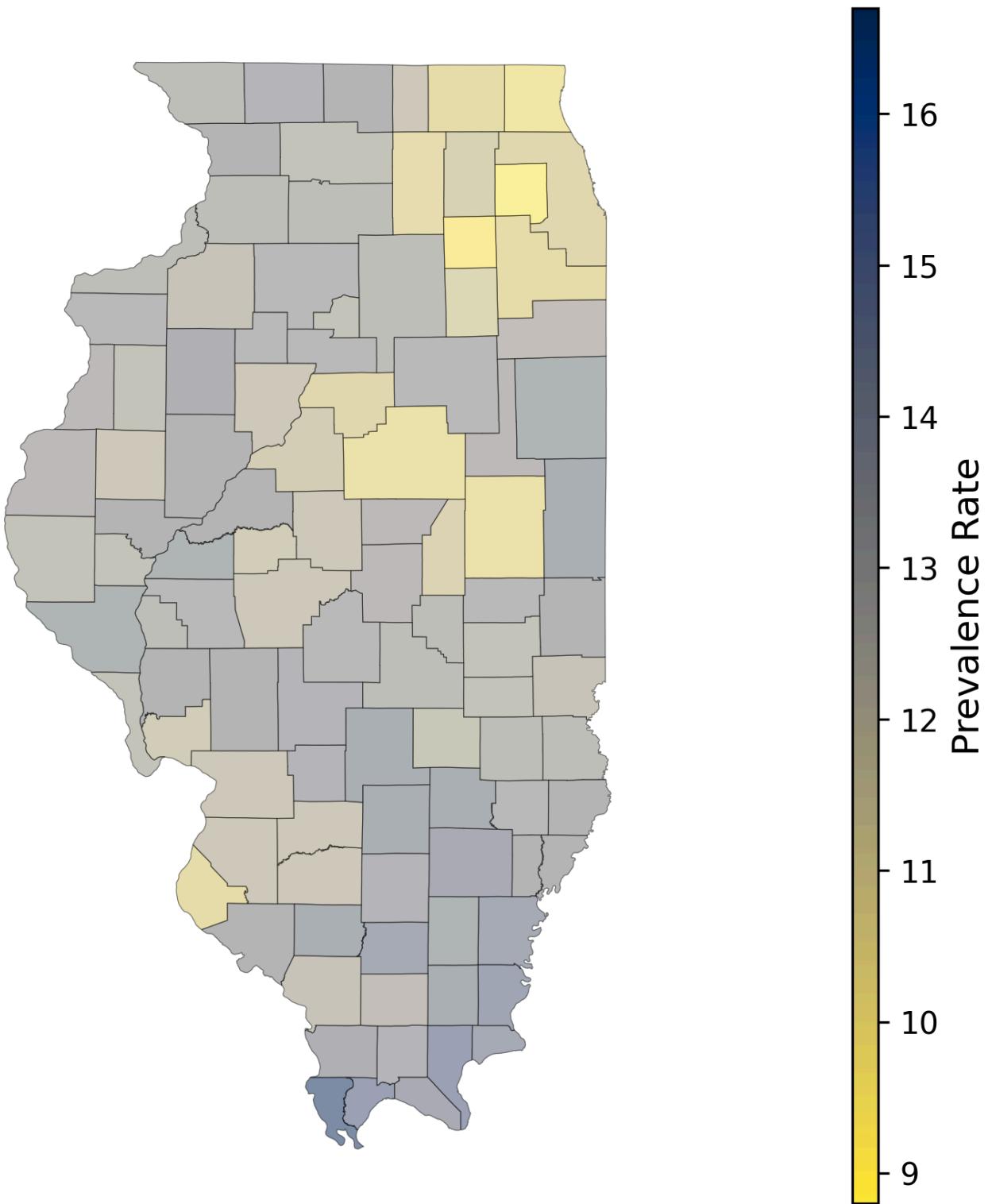
# MHLTH\_CrudePrev - Illinois Map by County



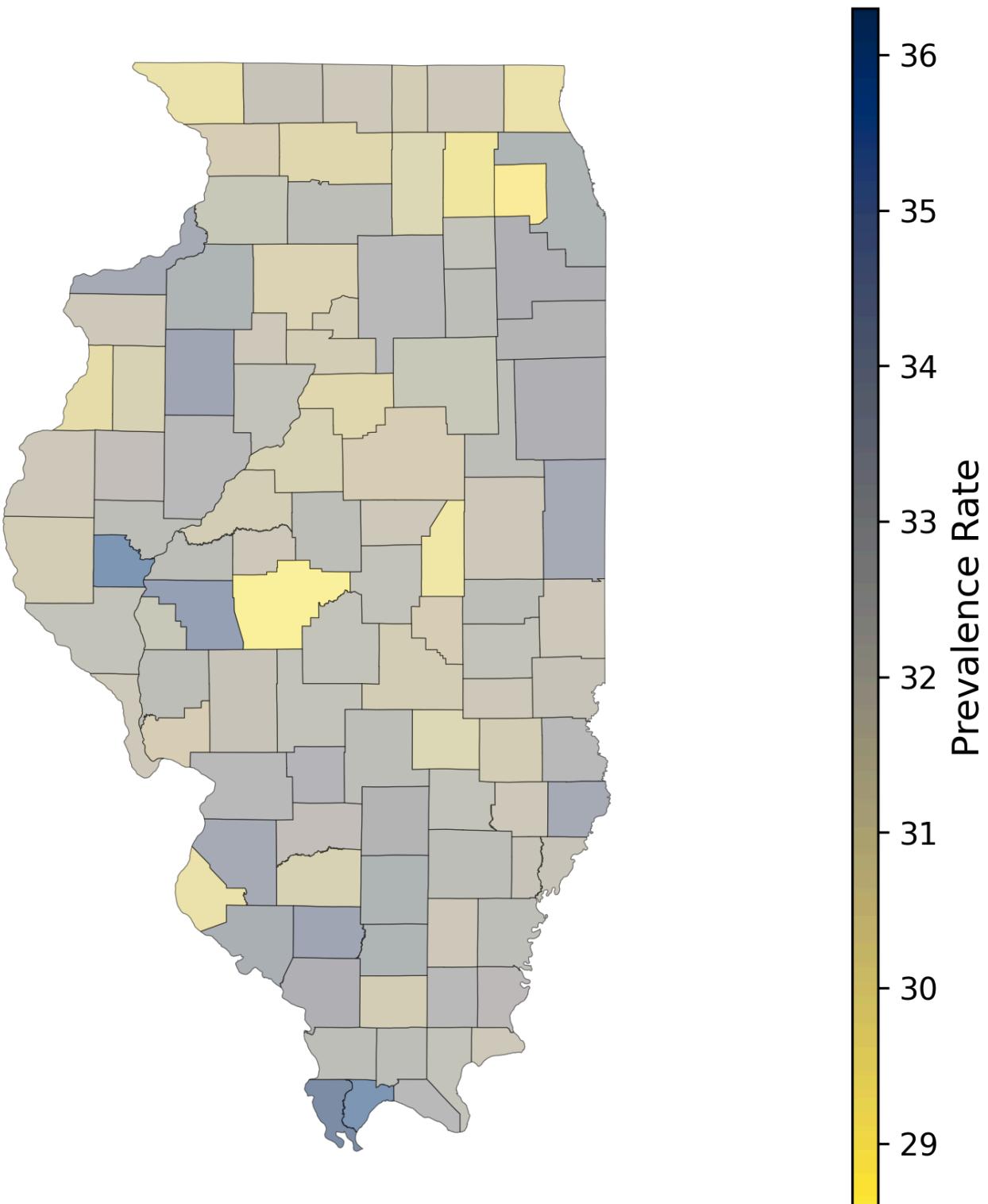
# OBESITY\_CrudePrev - Illinois Map by County



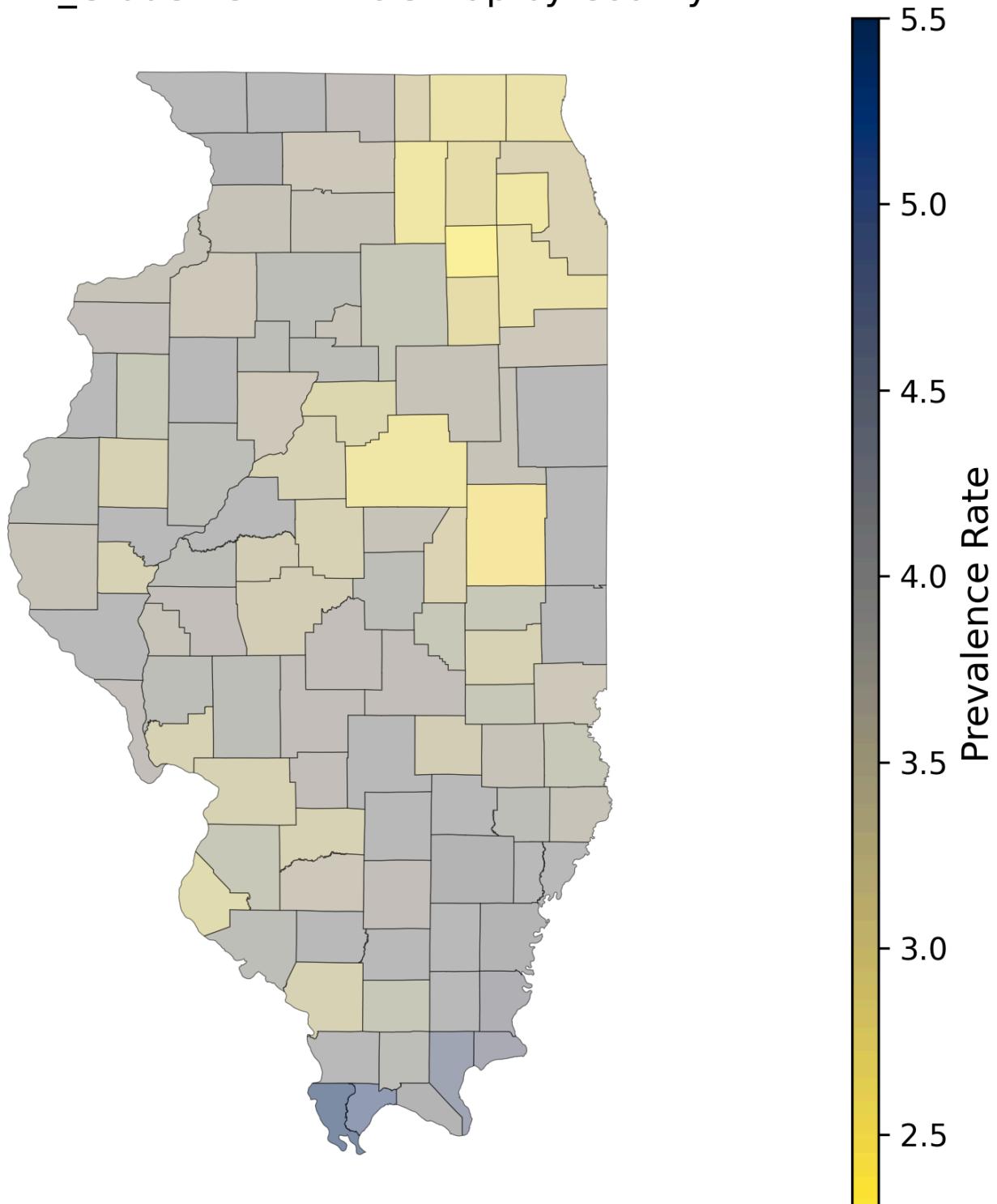
# PHLTH\_CrudePrev - Illinois Map by County



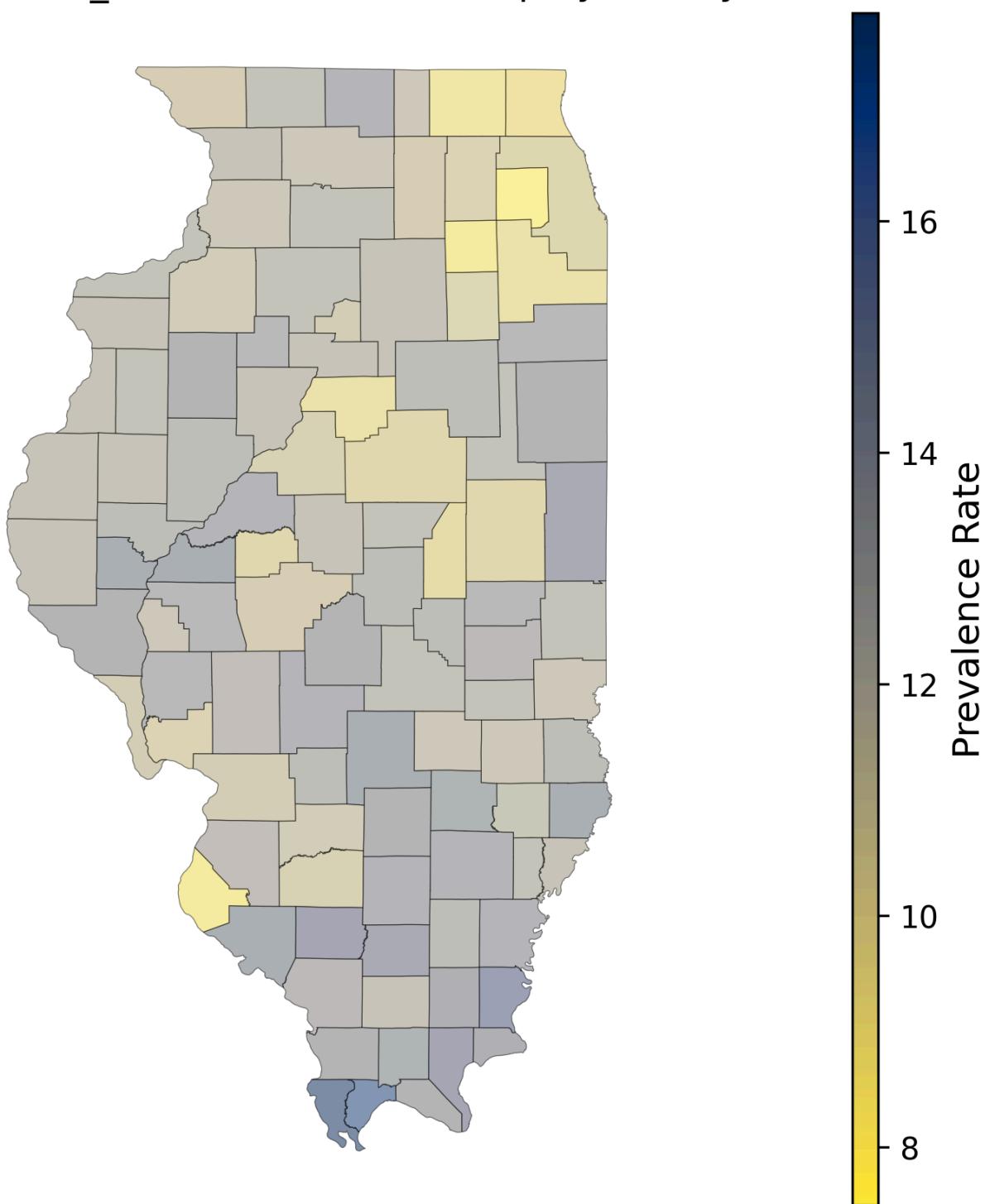
# SLEEP\_CrudePrev - Illinois Map by County



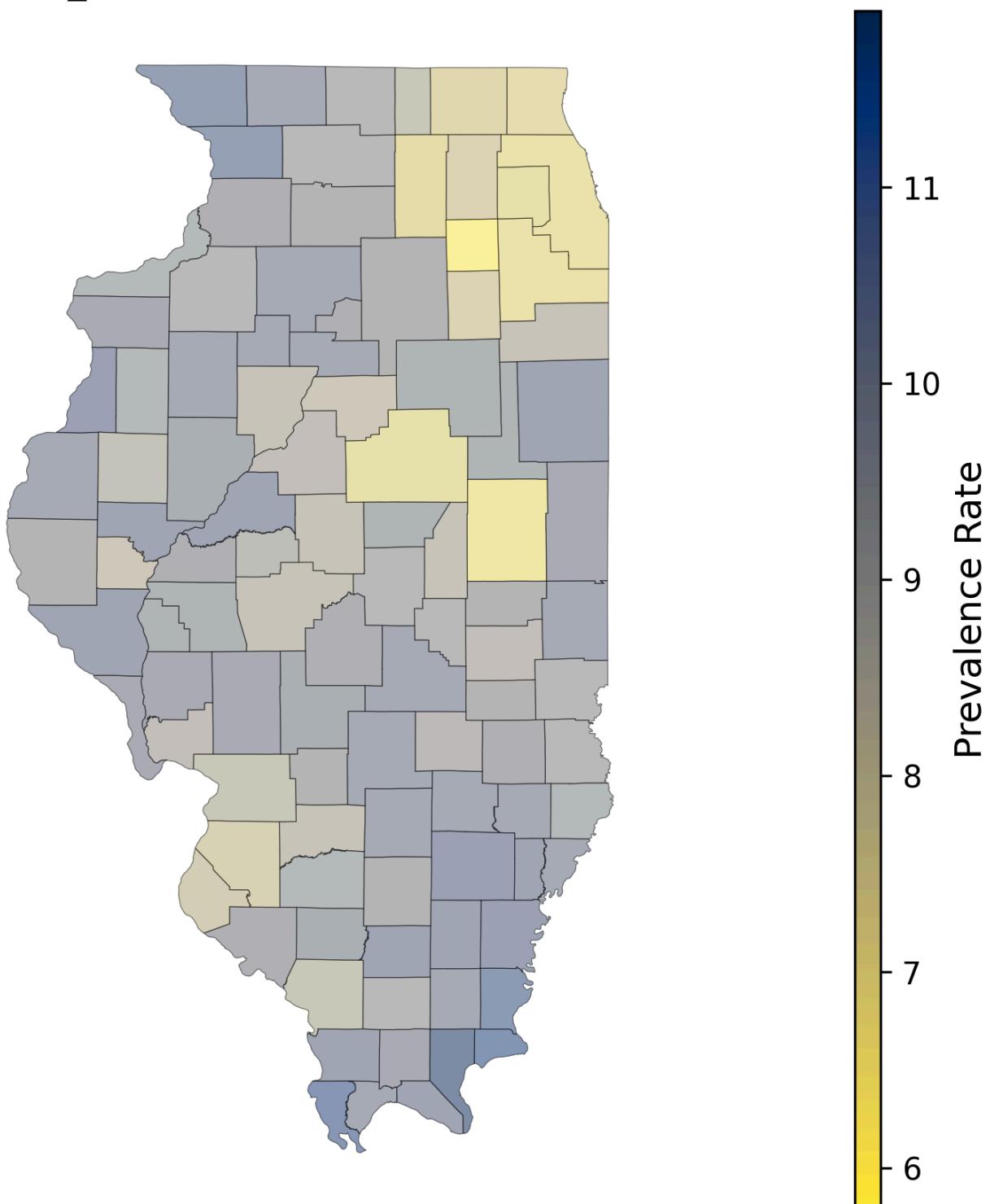
# STROKE\_CrudePrev - Illinois Map by County



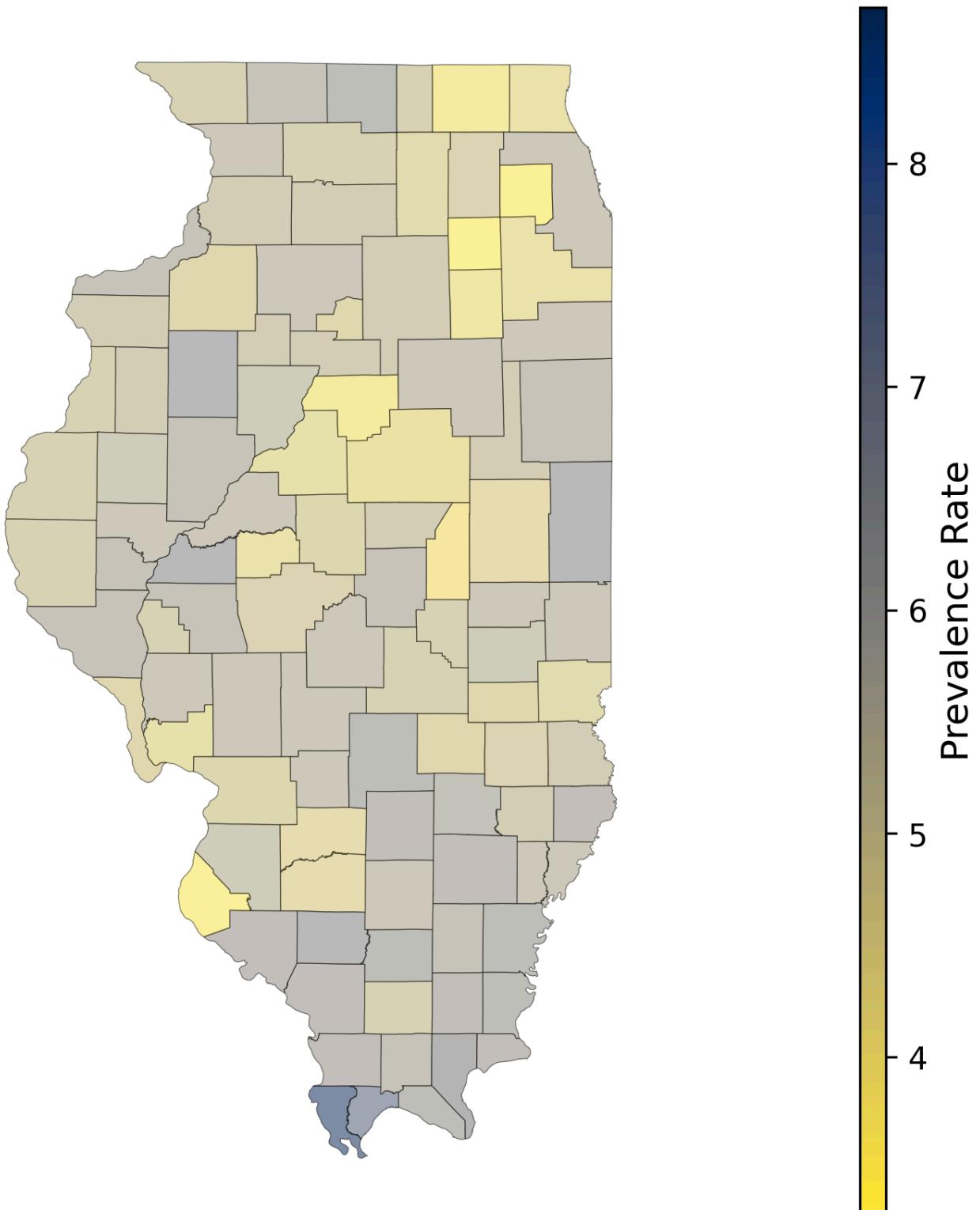
# TEETHLOST\_CrudePrev - Illinois Map by County



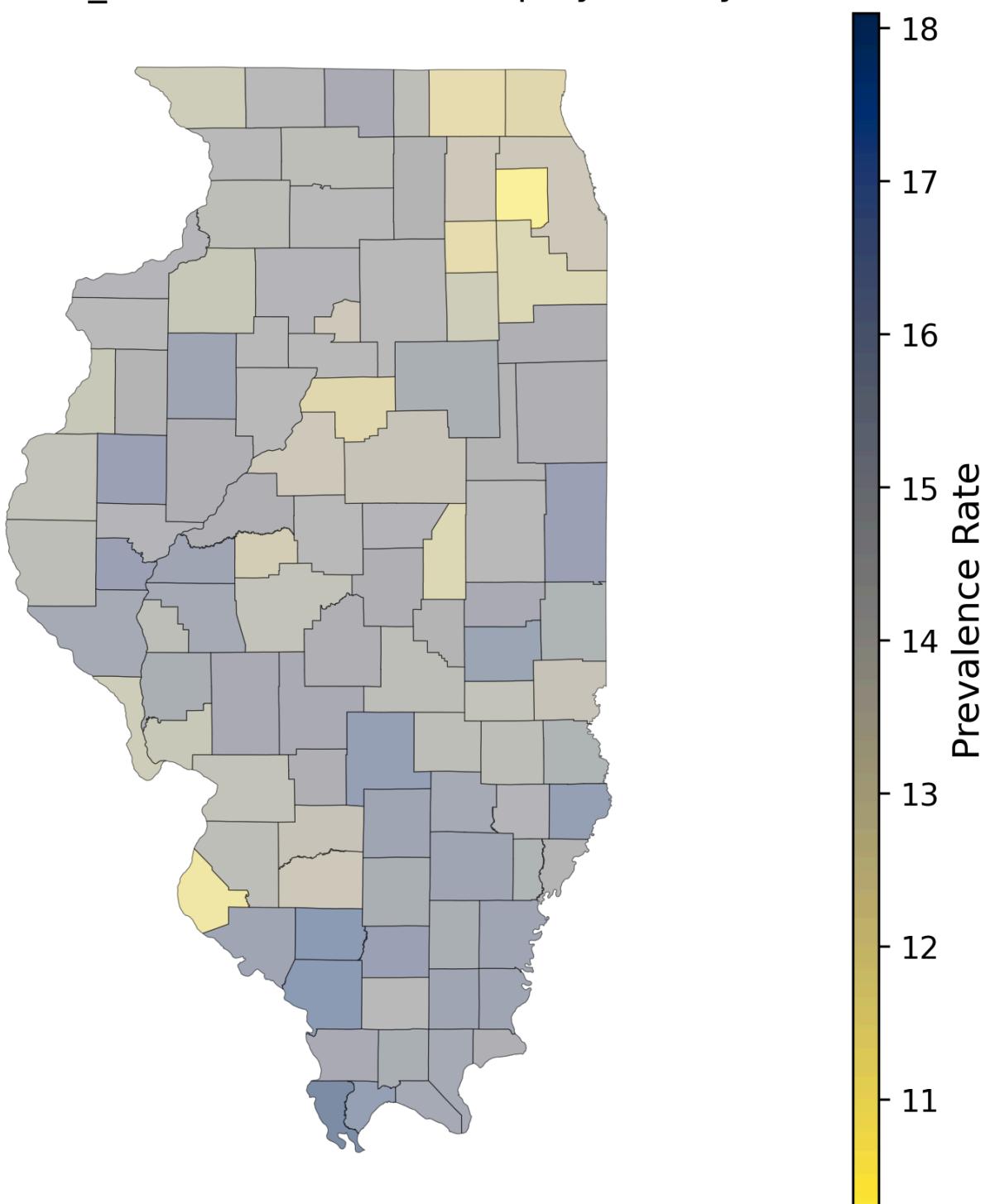
# HEARING\_CrudePrev - Illinois Map by County



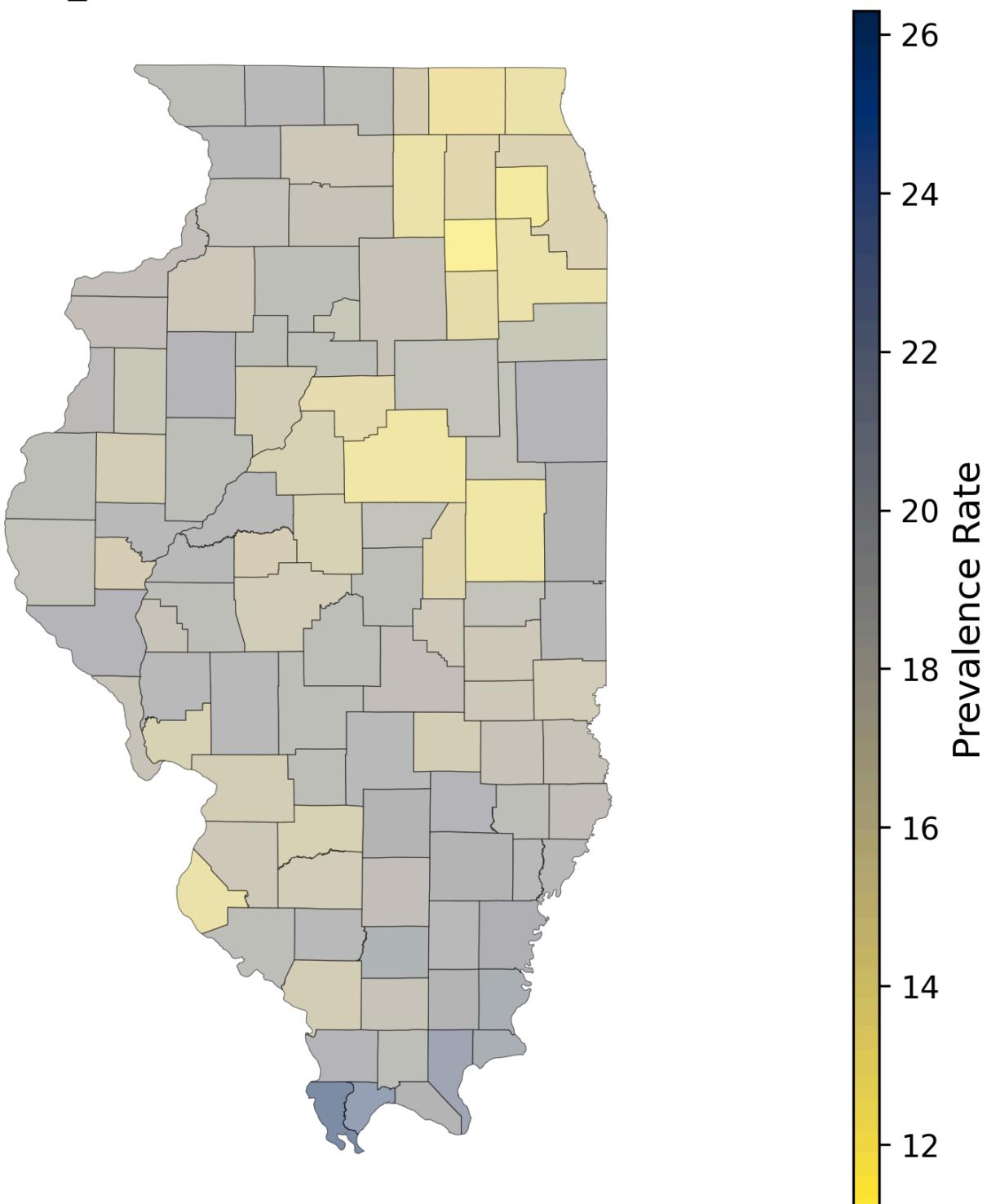
# VISION\_CrudePrev - Illinois Map by County



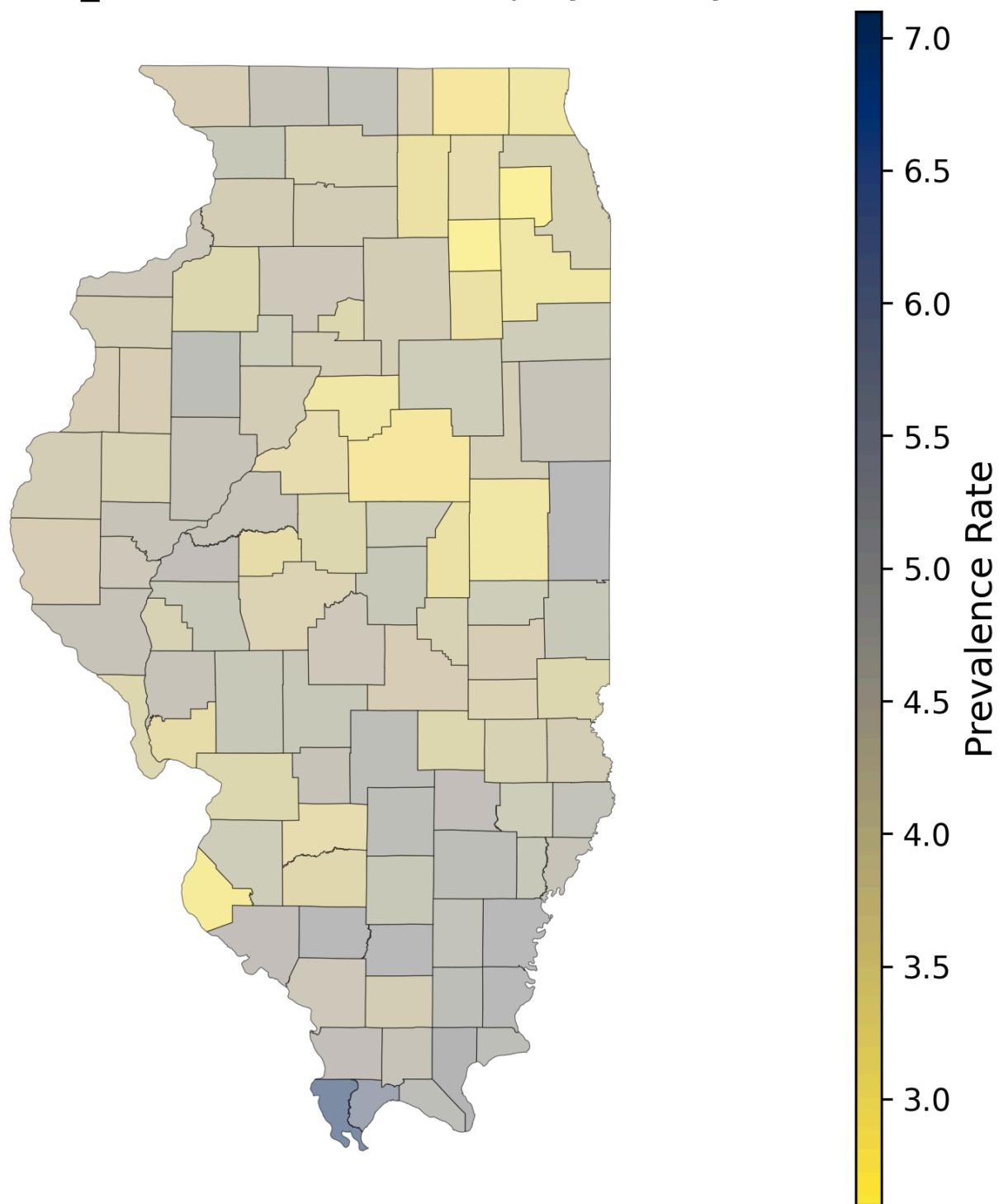
# COGNITION\_CrudePrev - Illinois Map by County



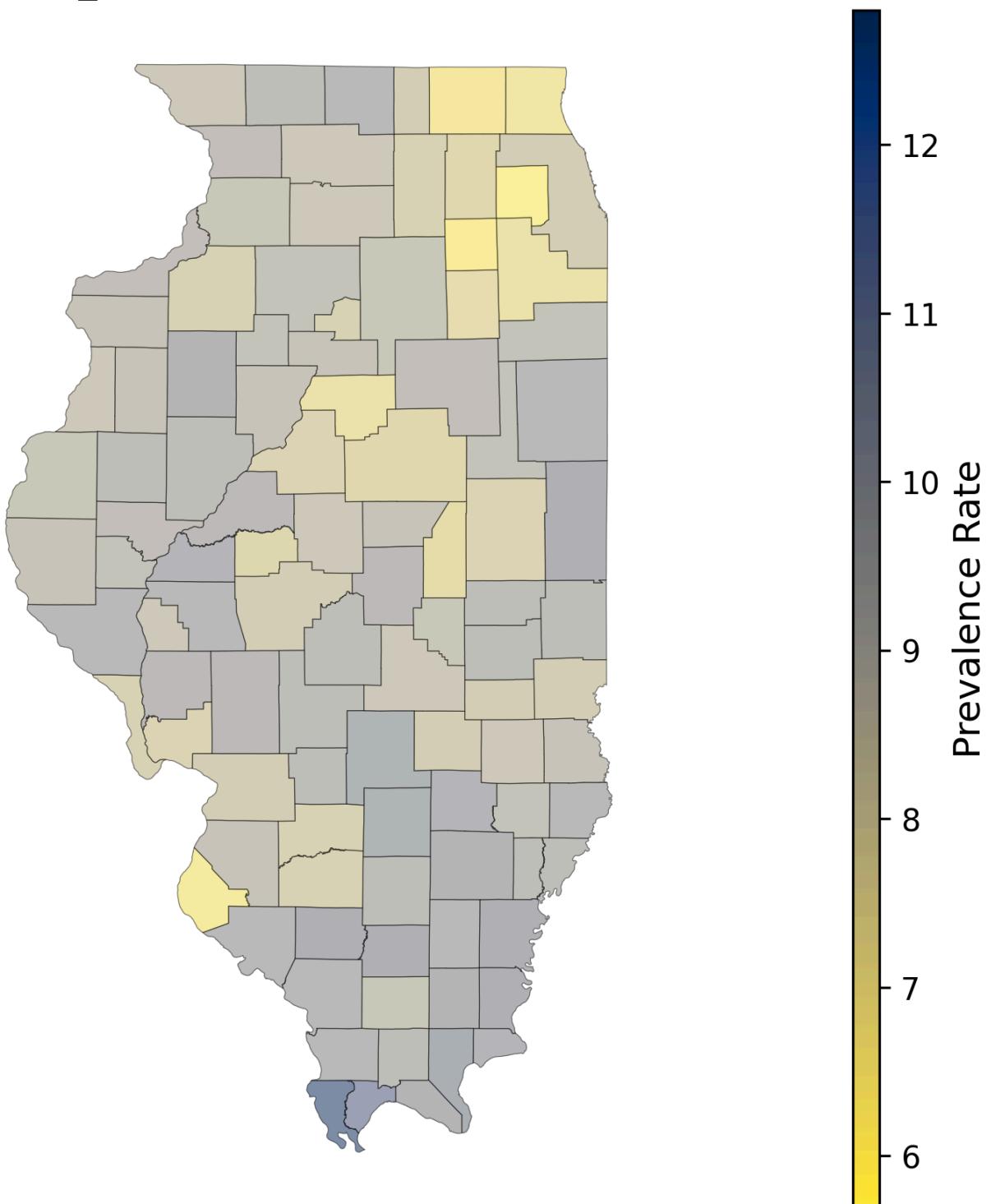
# MOBILITY\_CrudePrev - Illinois Map by County



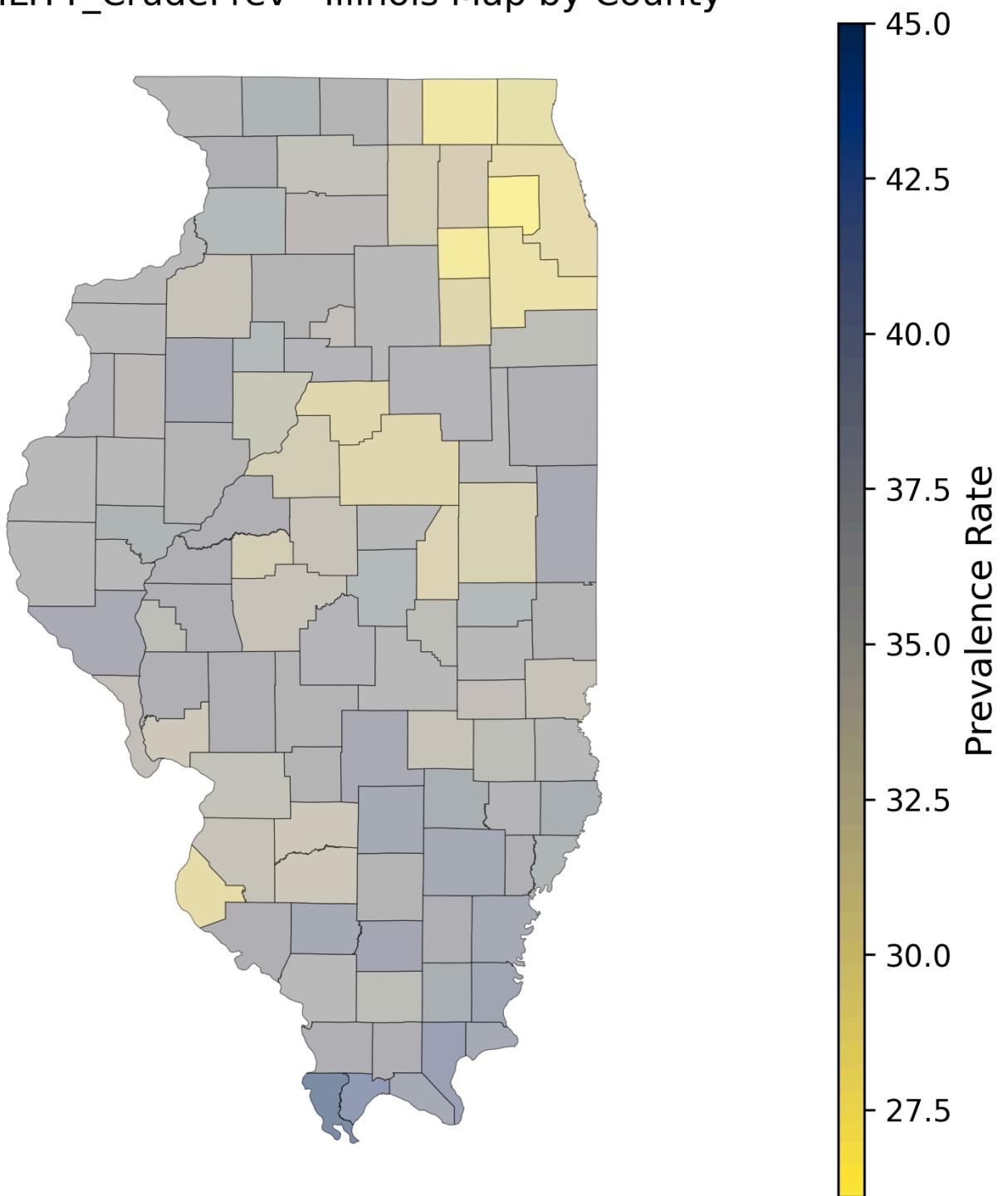
# SELCARE\_CrudePrev - Illinois Map by County



# INDEPLIVE\_CrudePrev - Illinois Map by County

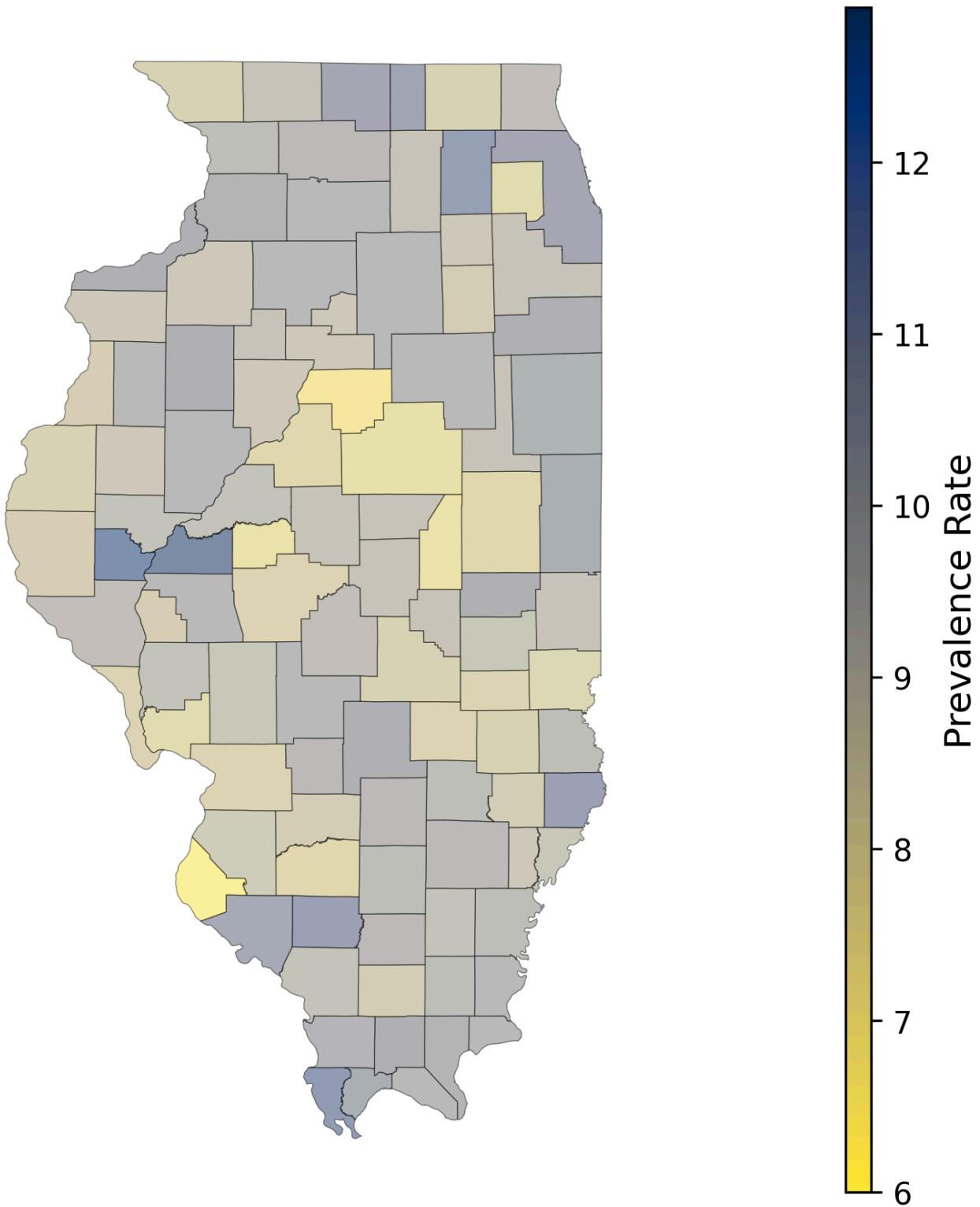


# DISABILITY\_CrudePrev - Illinois Map by County

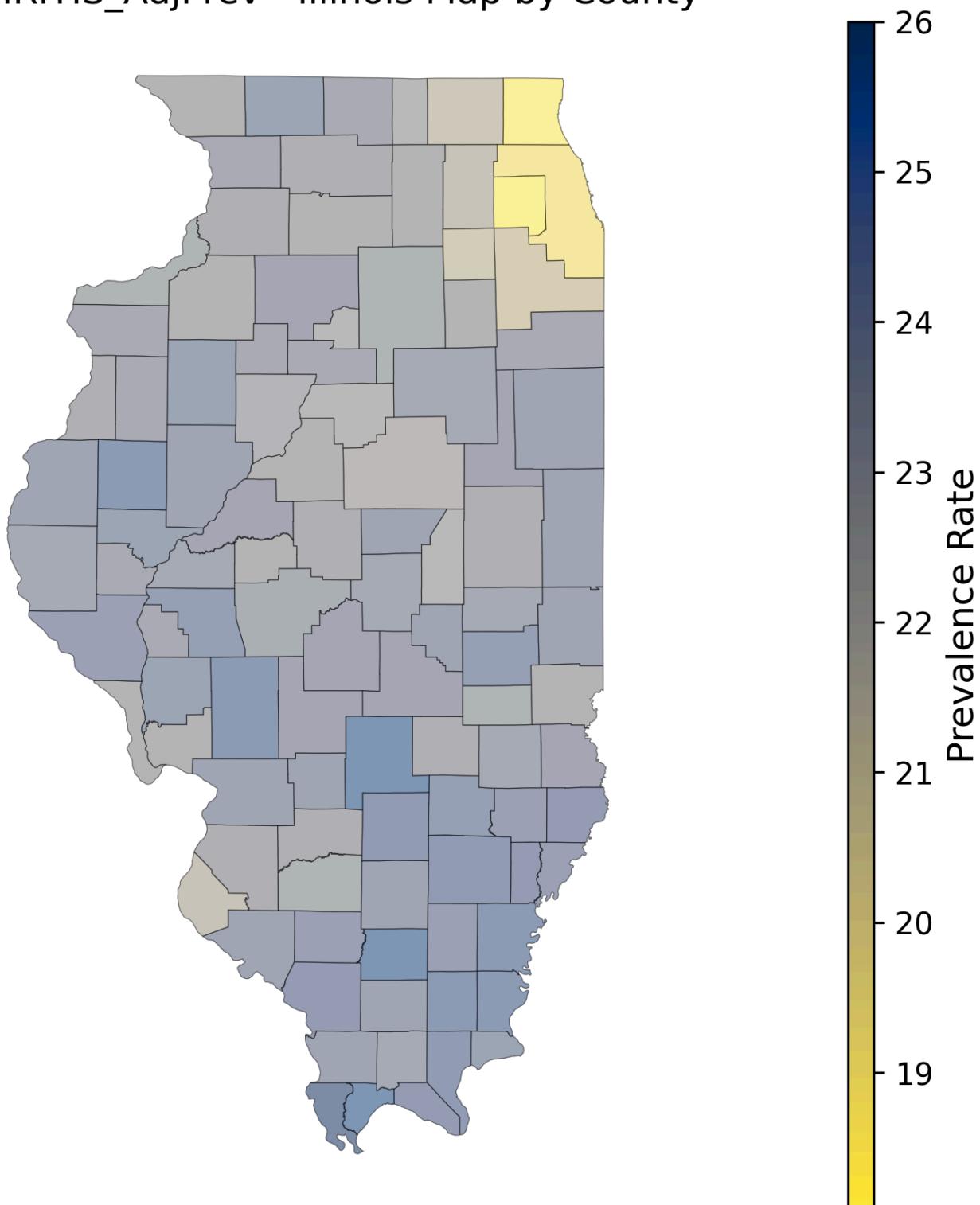


```
In [ ]: # Get columns ending with '_AdjPrev'  
columns_to_plot = [col for col in df.columns if col.endswith('_AdjPrev')]  
  
plot_normalized_county_data(merged_gdf, columns_to_plot, 'cividis_r')
```

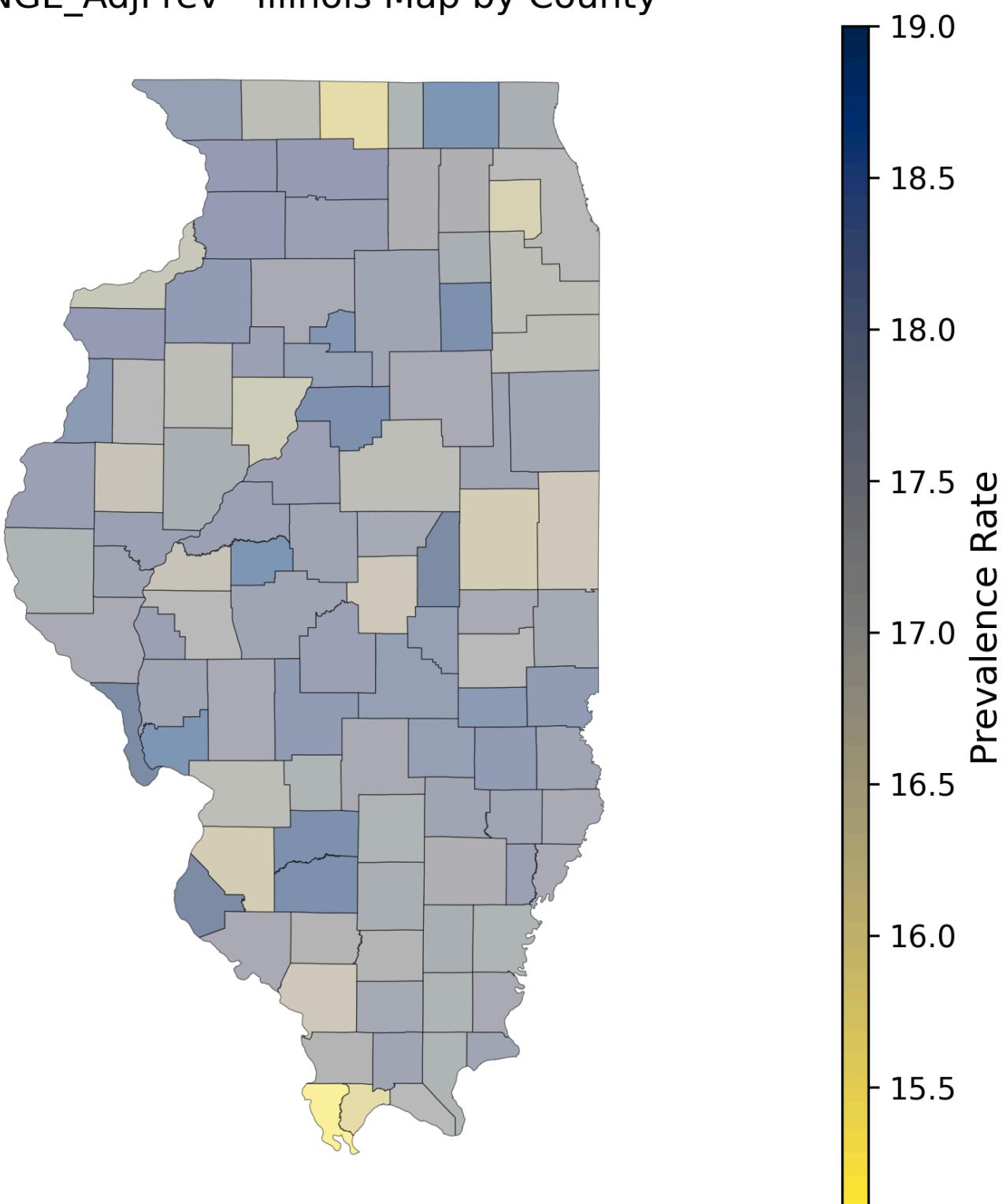
## ACCESS2\_AdjPrev - Illinois Map by County



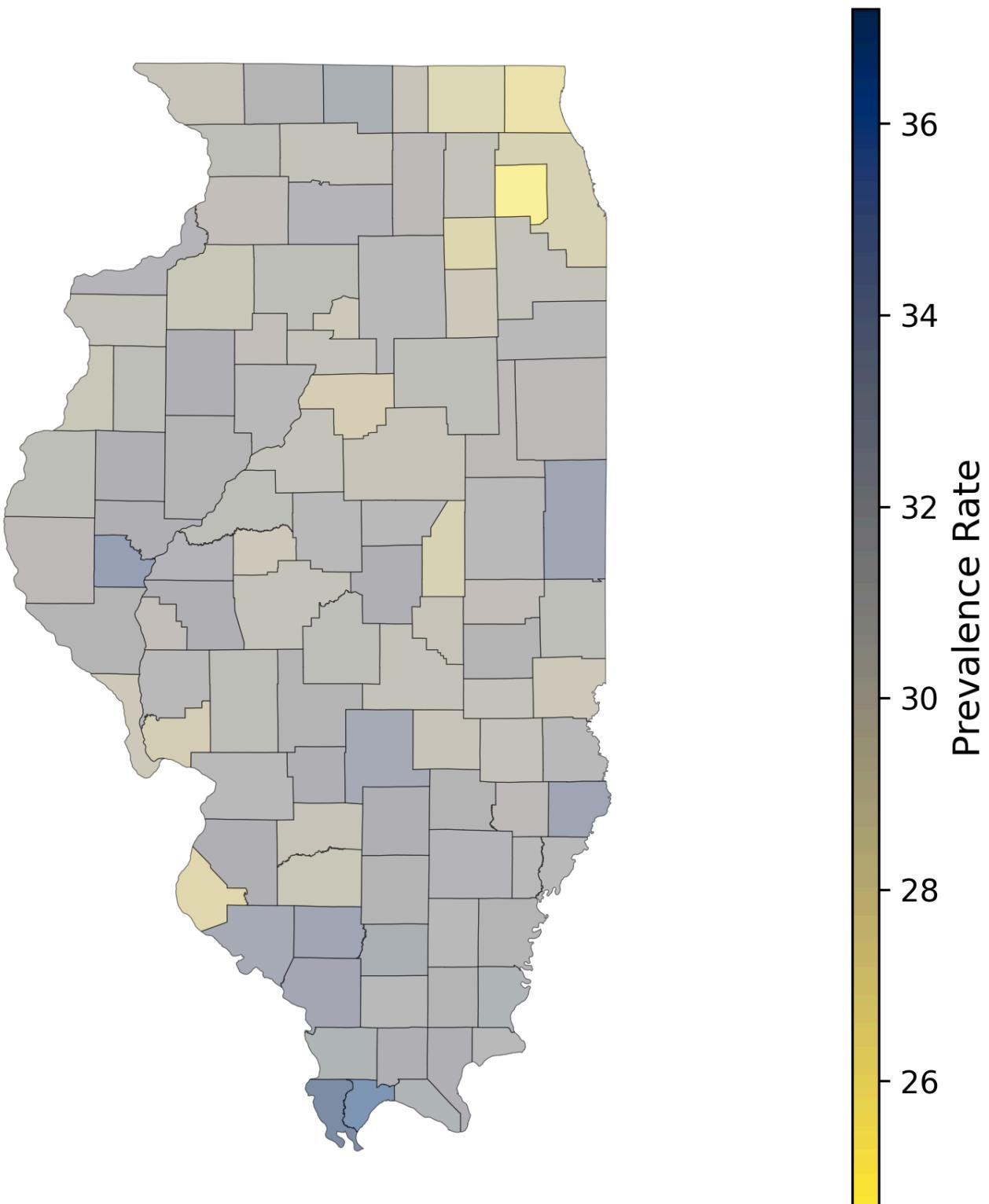
# ARTHRITIS\_AdjPrev - Illinois Map by County



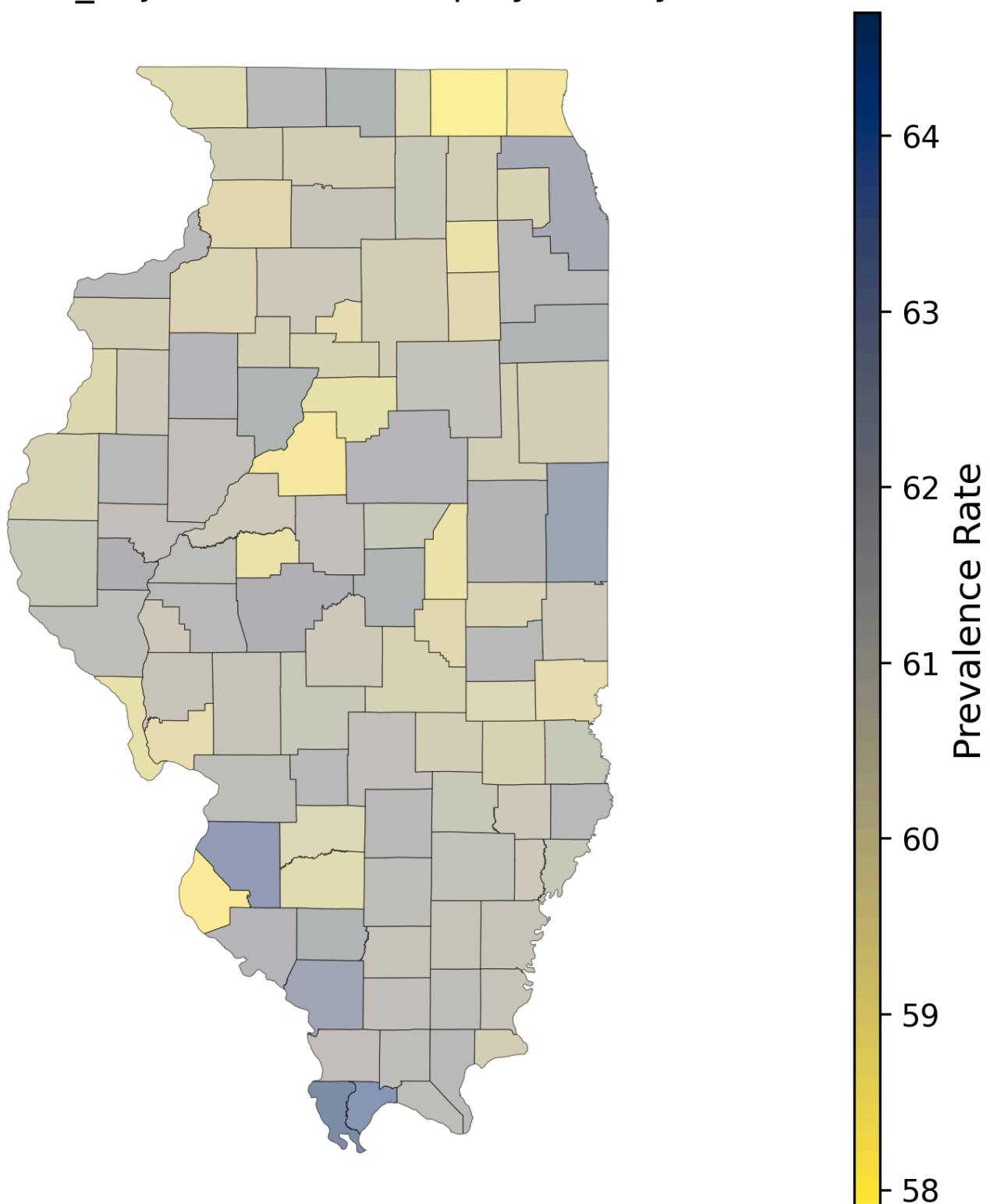
# BINGE\_AdjPrev - Illinois Map by County



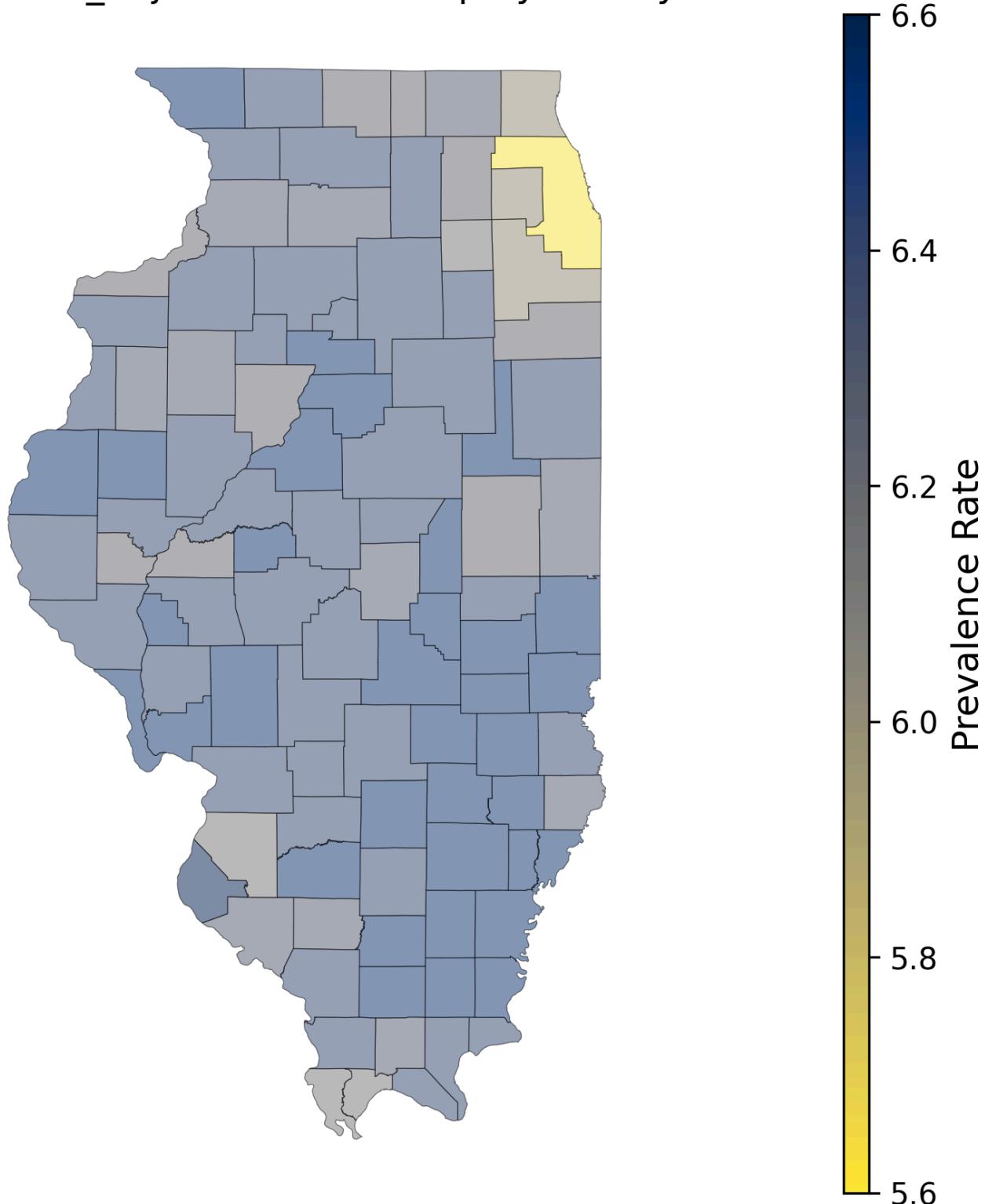
# BPHIGH\_AdjPrev - Illinois Map by County



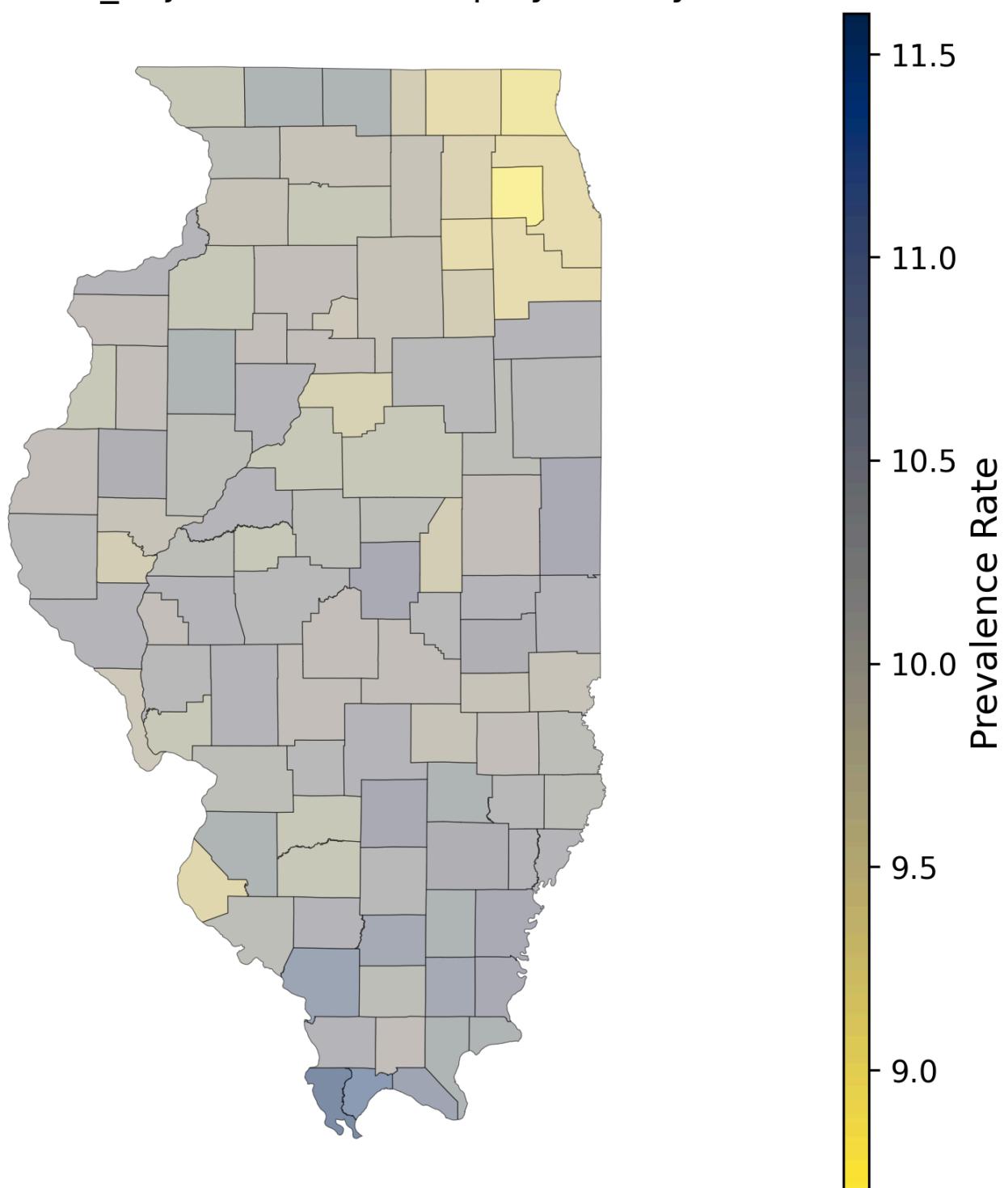
# BPMED\_AdjPrev - Illinois Map by County



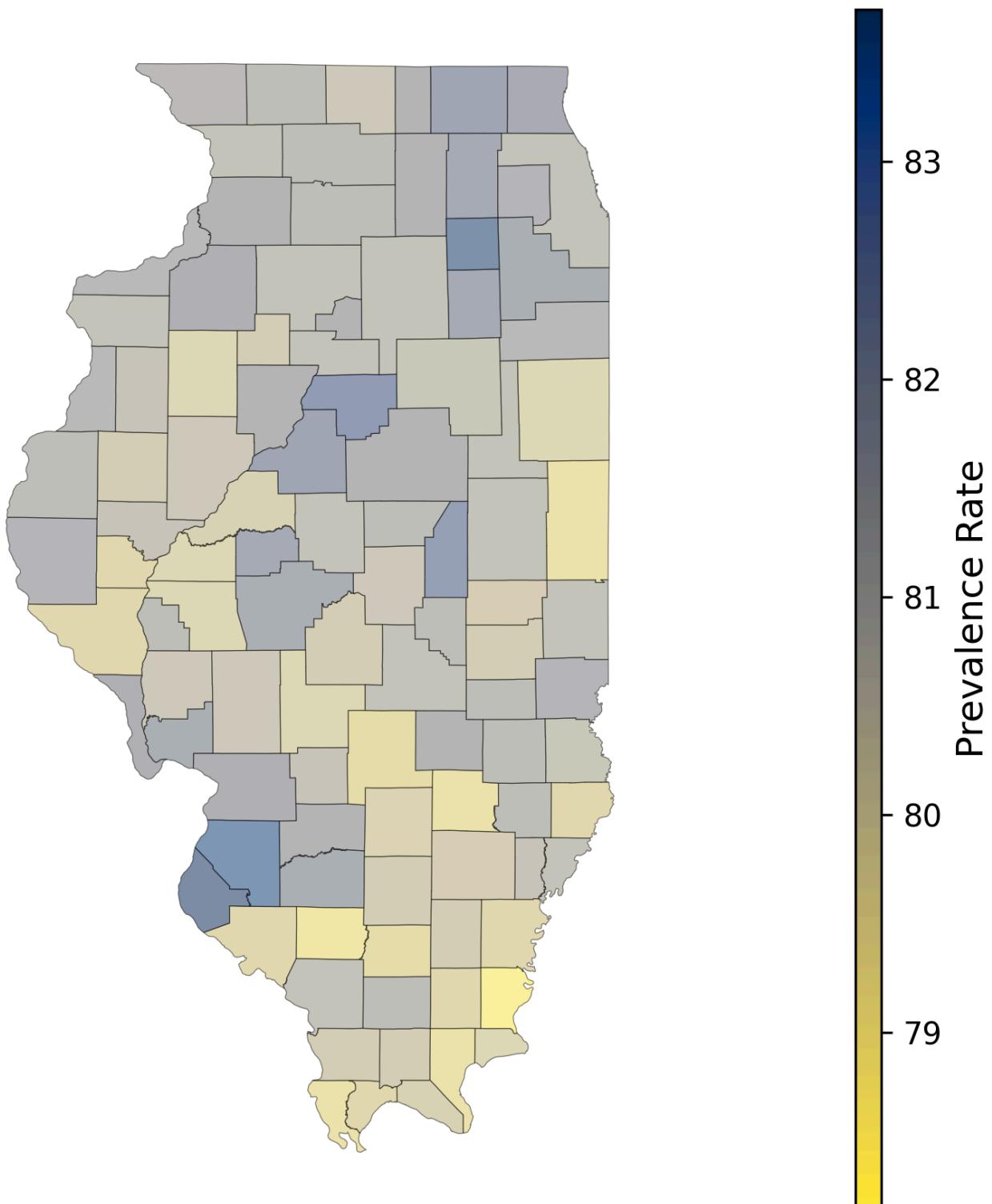
# CANCER\_AdjPrev - Illinois Map by County



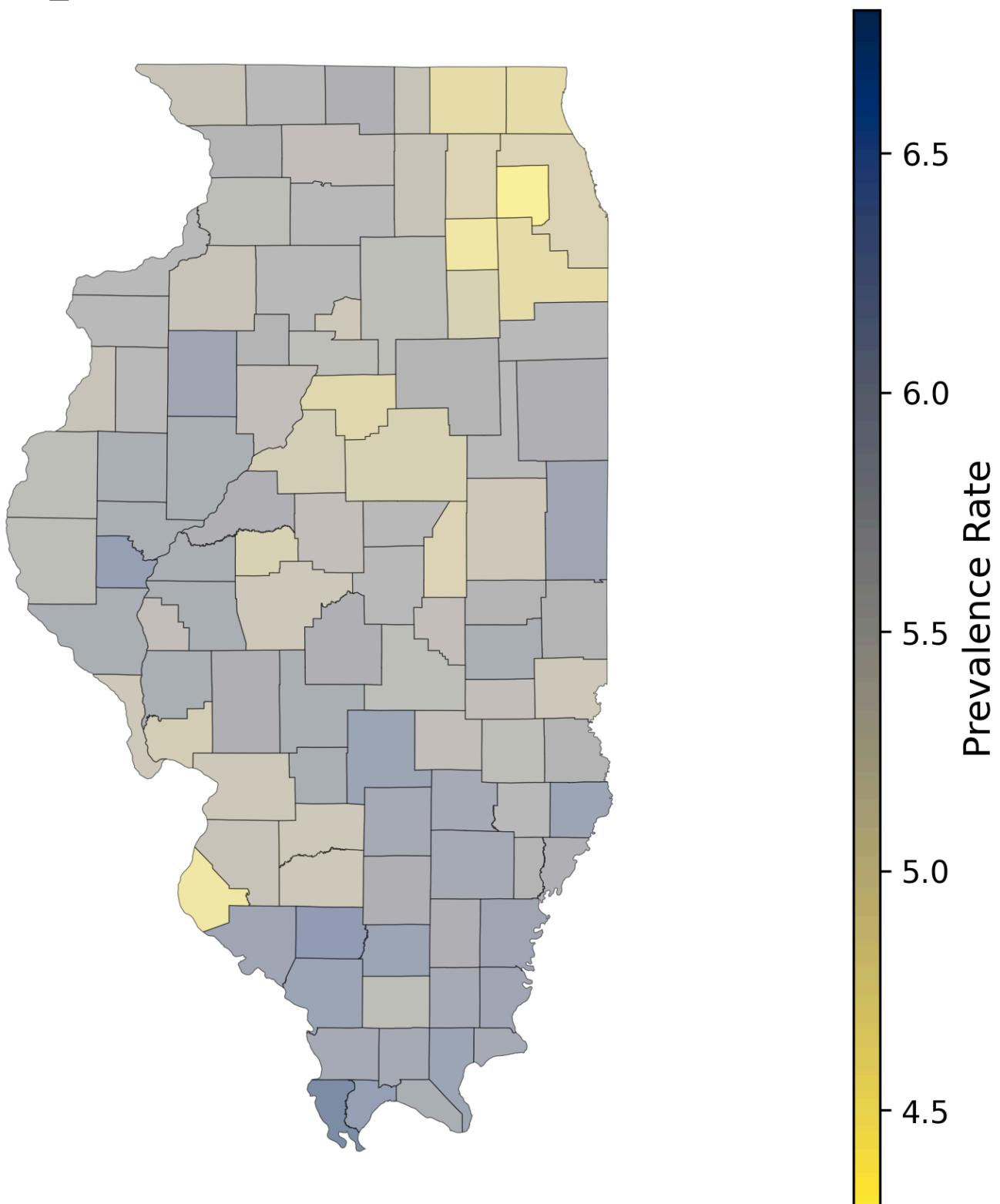
# CASTHMA\_AdjPrev - Illinois Map by County



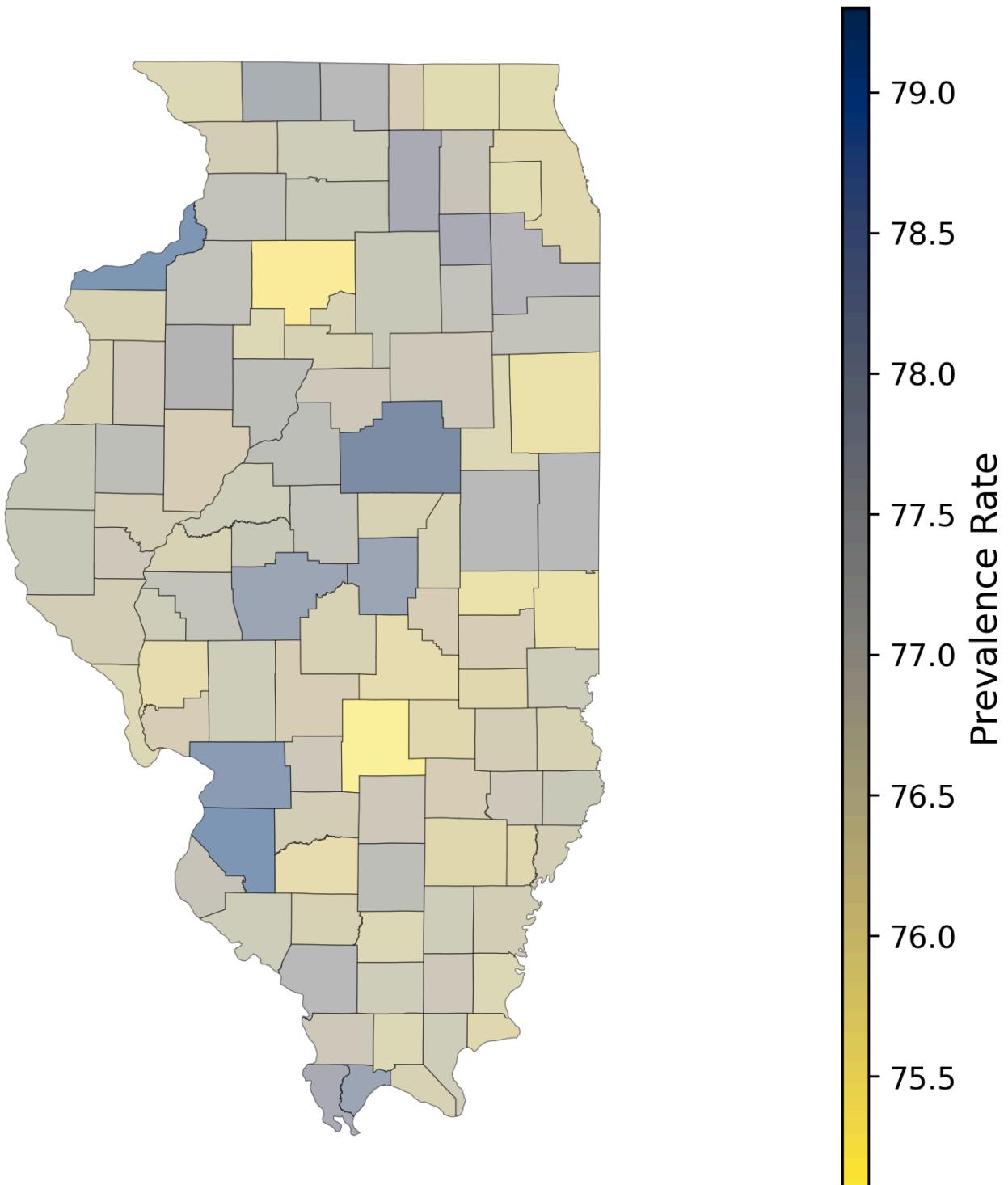
# CERVICAL\_AdjPrev - Illinois Map by County



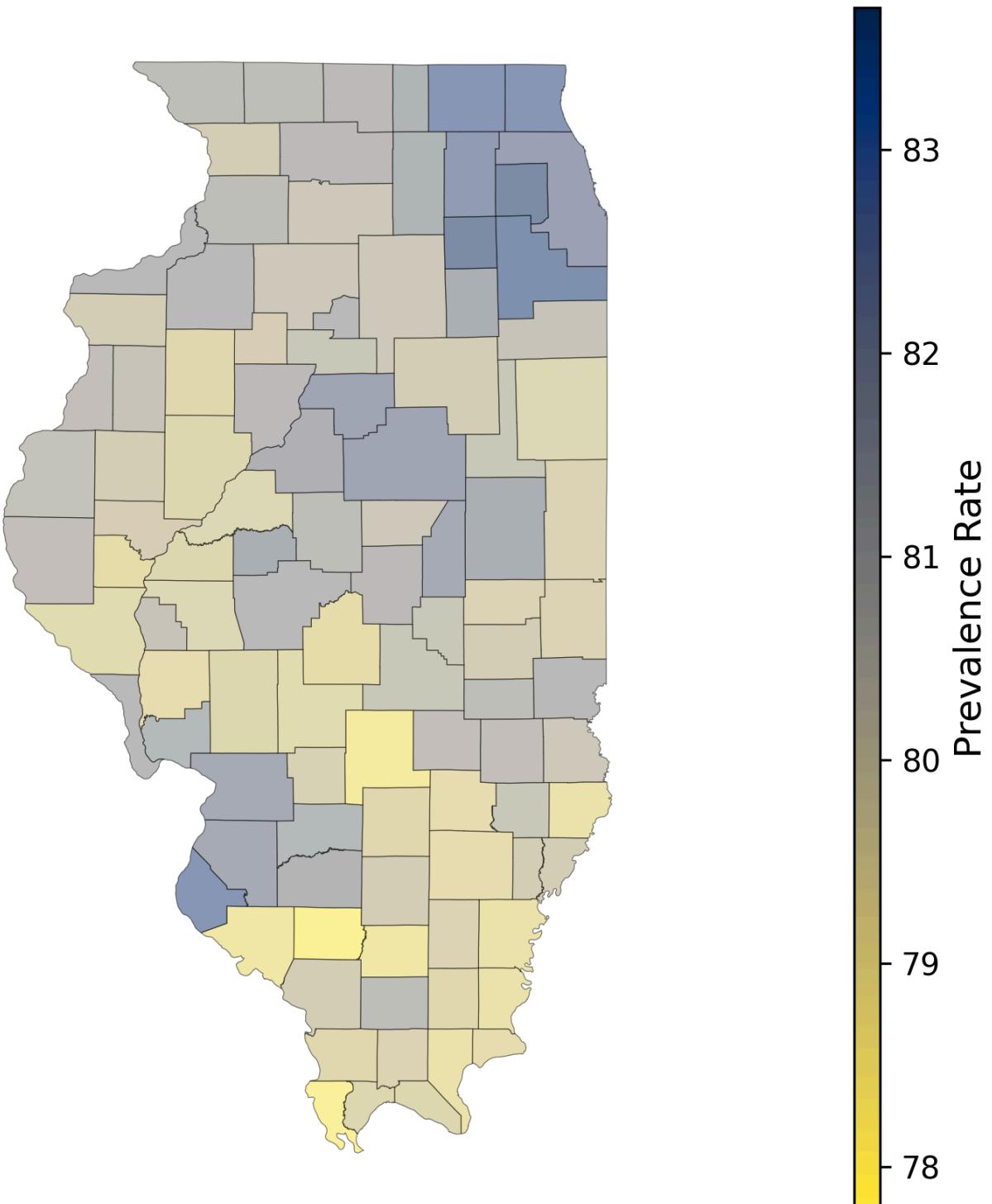
# CHD\_AdjPrev - Illinois Map by County



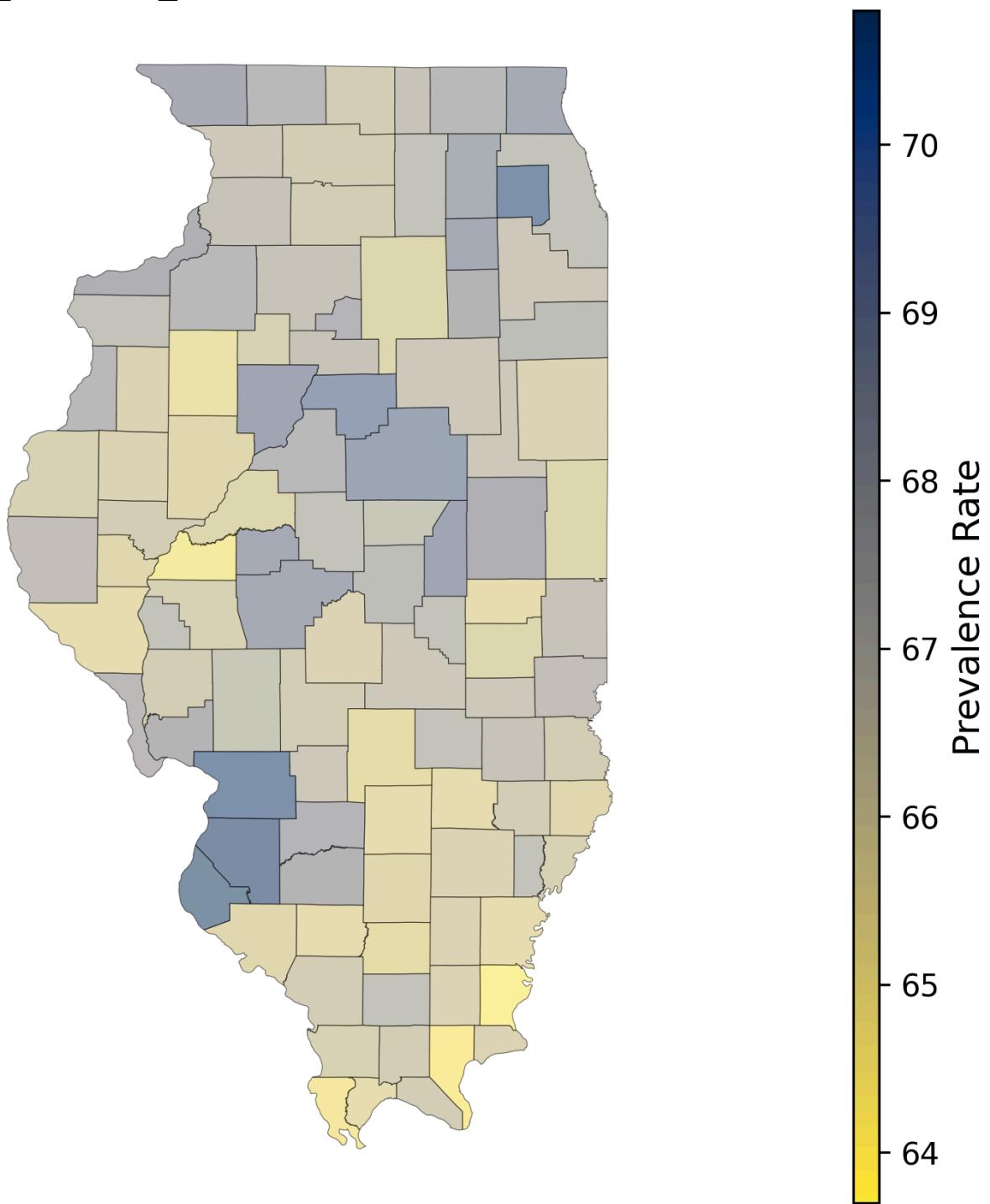
# CHECKUP\_AdjPrev - Illinois Map by County



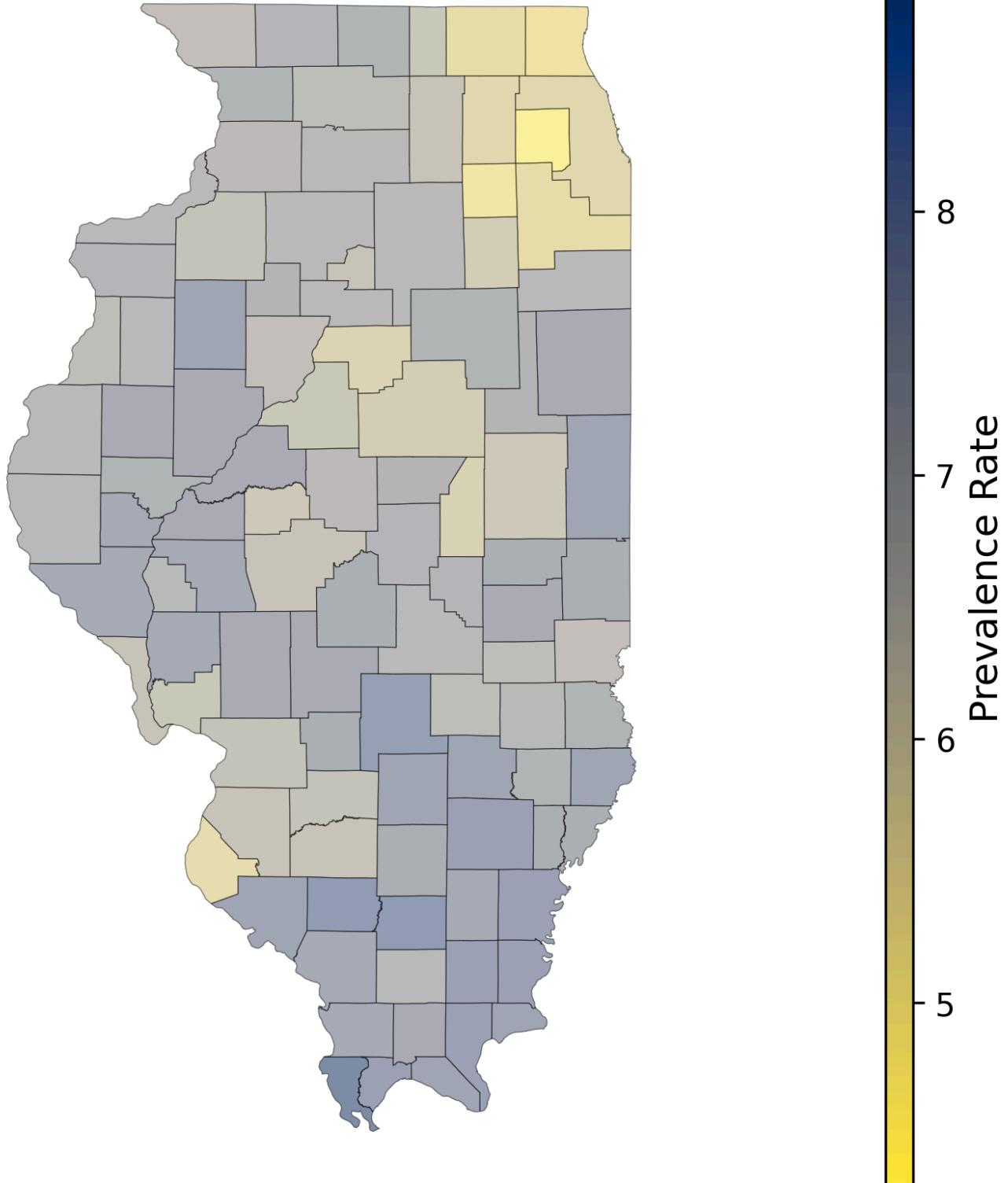
# CHOLSCREEN\_AdjPrev - Illinois Map by County



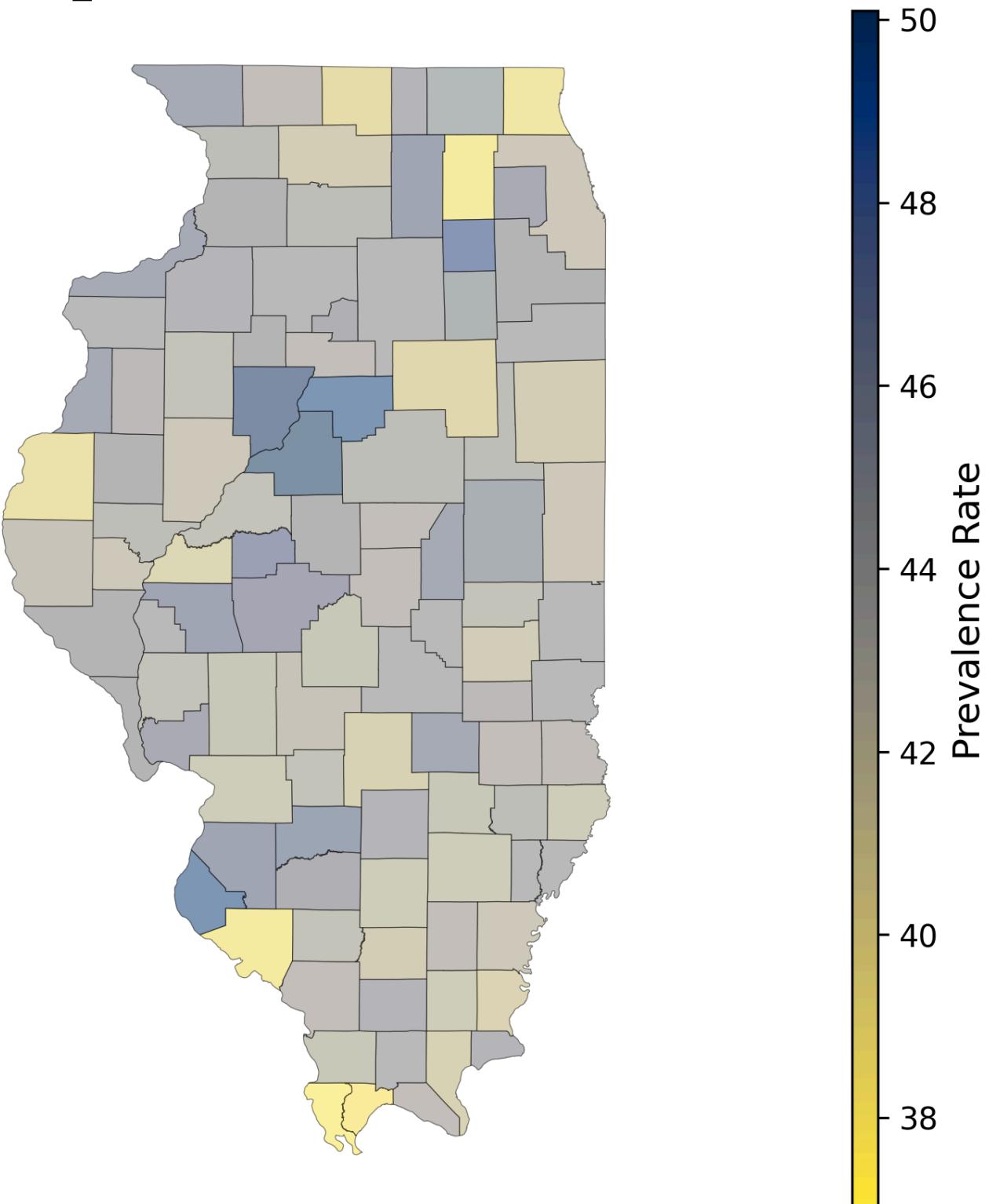
# COLON\_SCREEN\_AdjPrev - Illinois Map by County



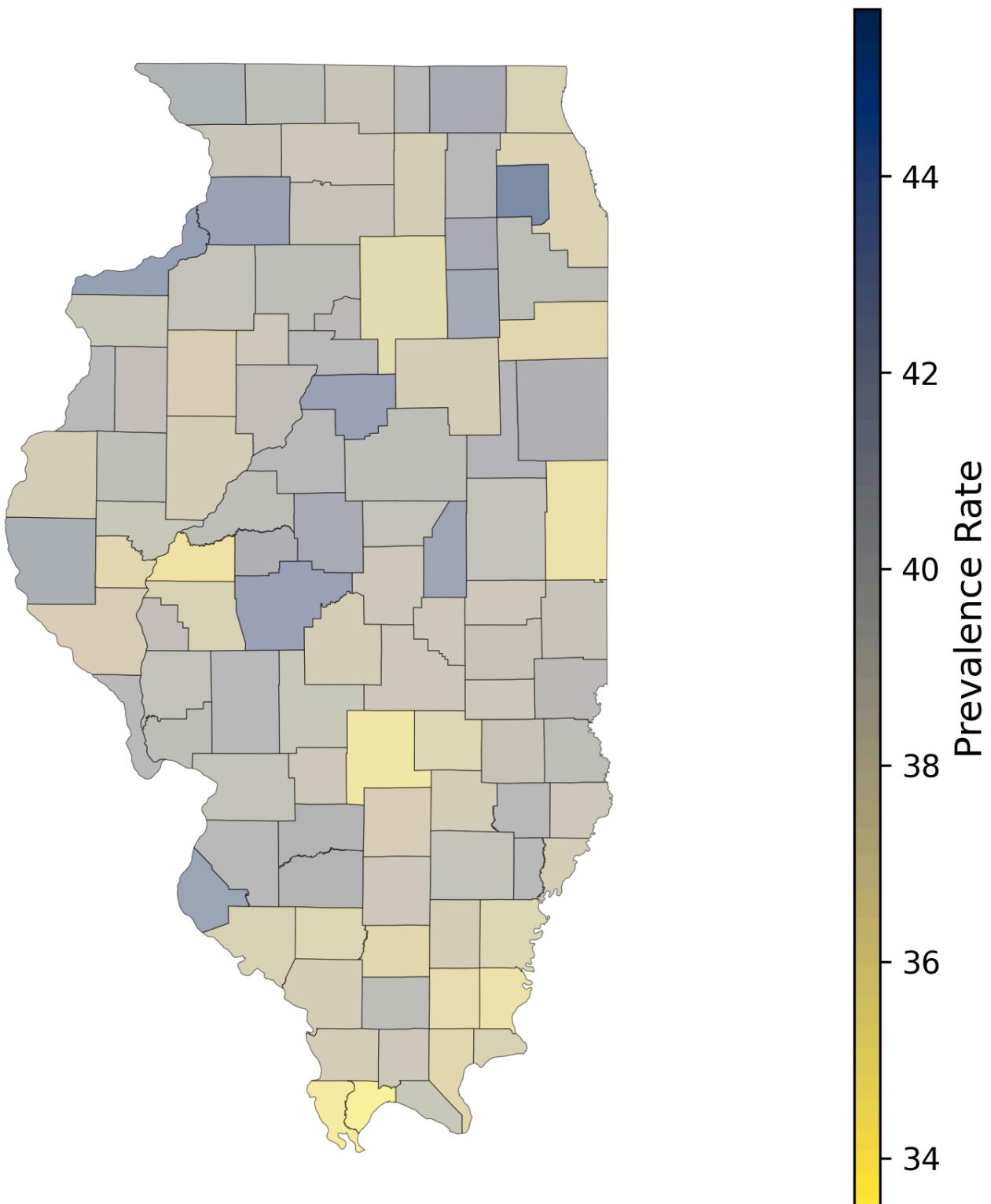
# COPD\_AdjPrev - Illinois Map by County



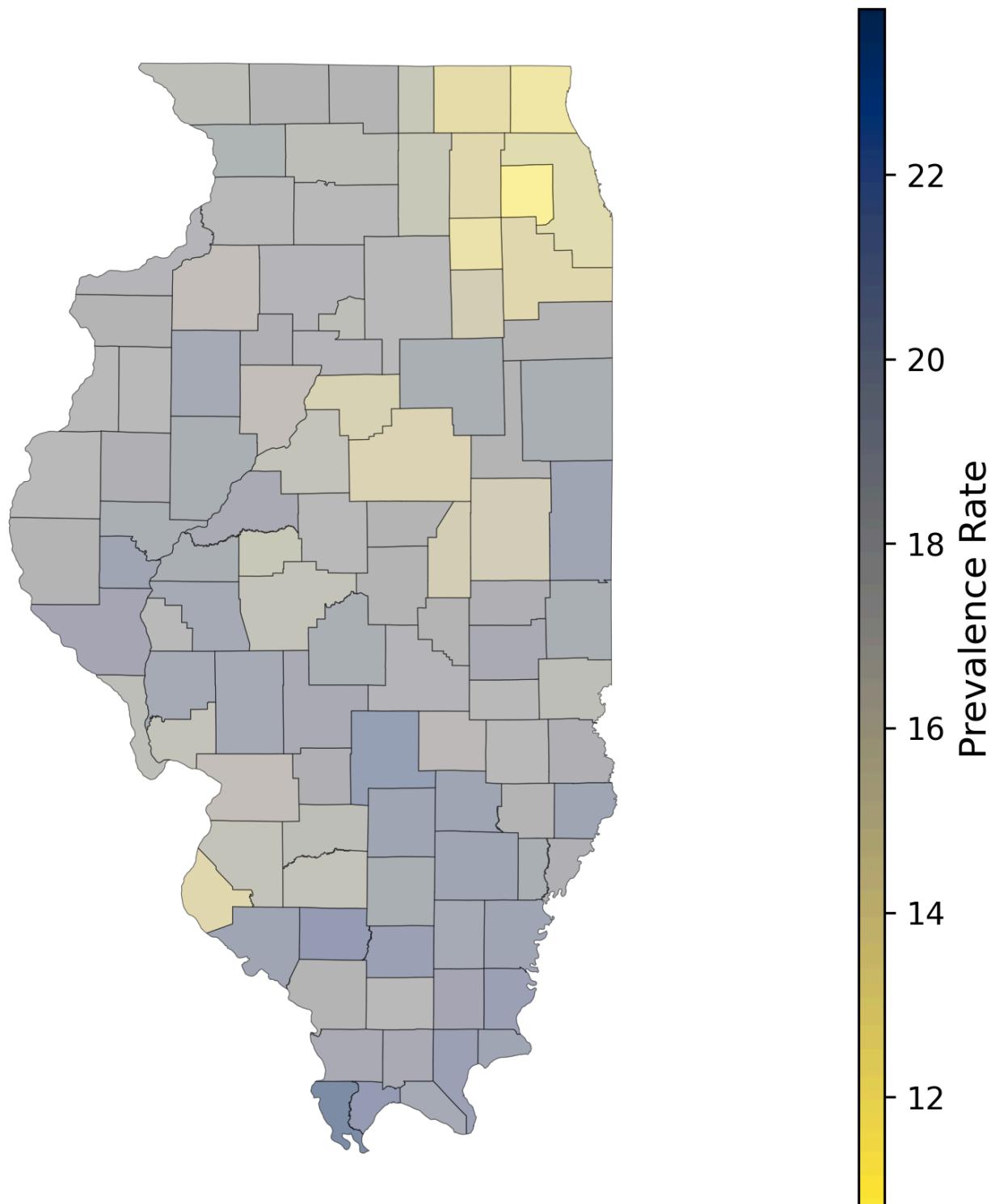
# COREM\_AdjPrev - Illinois Map by County



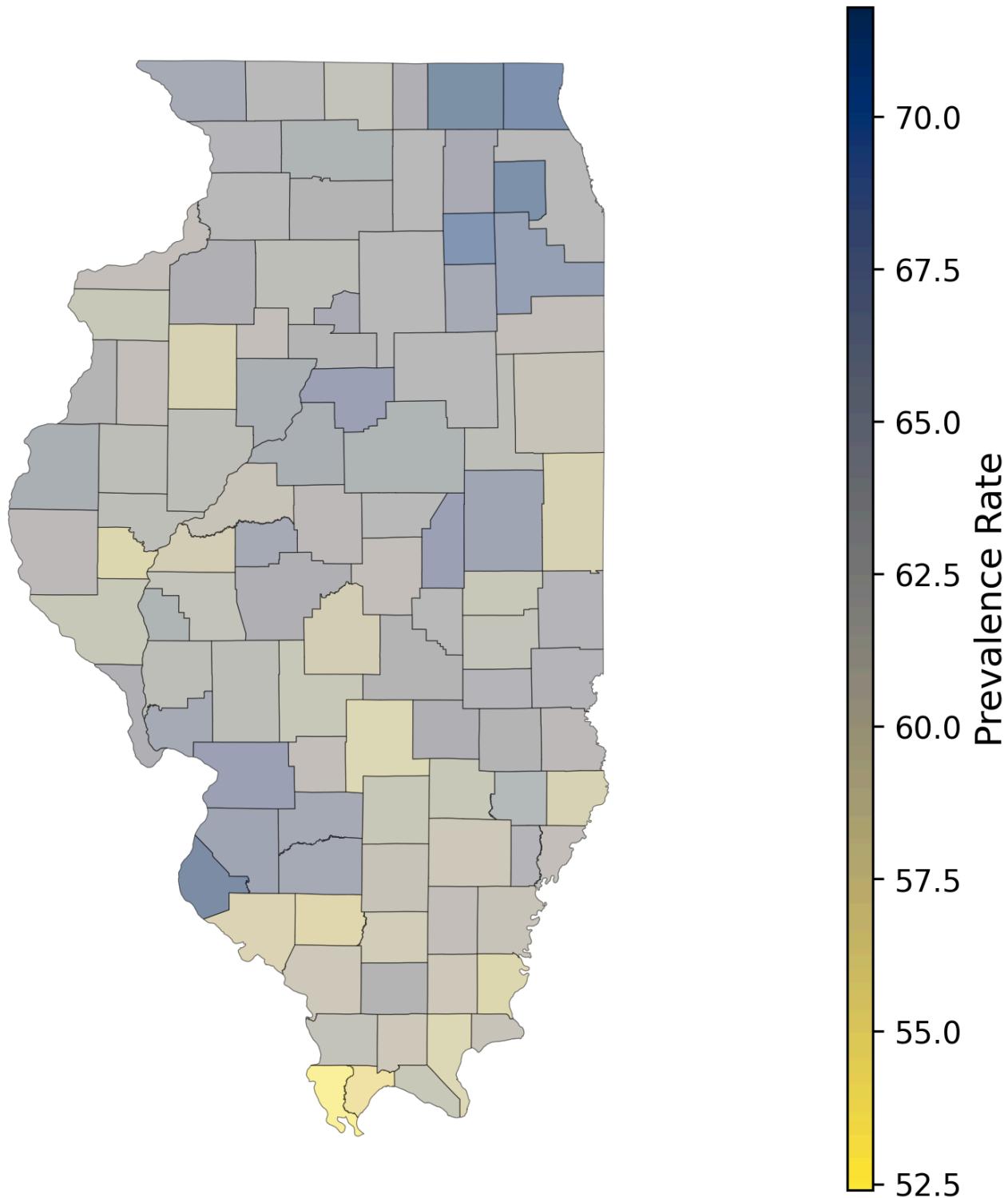
# COREW\_AdjPrev - Illinois Map by County



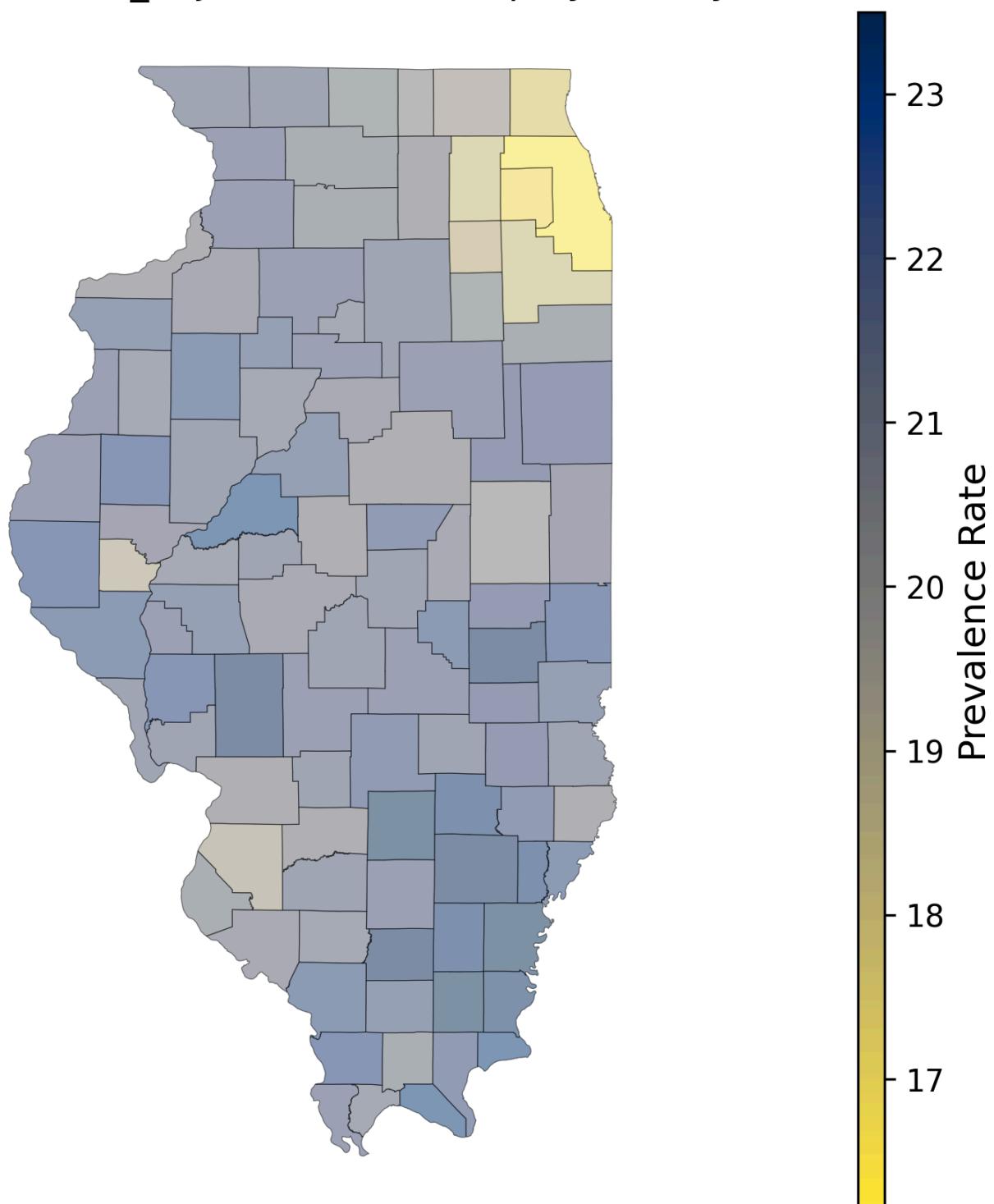
# CSMOKING\_AdjPrev - Illinois Map by County



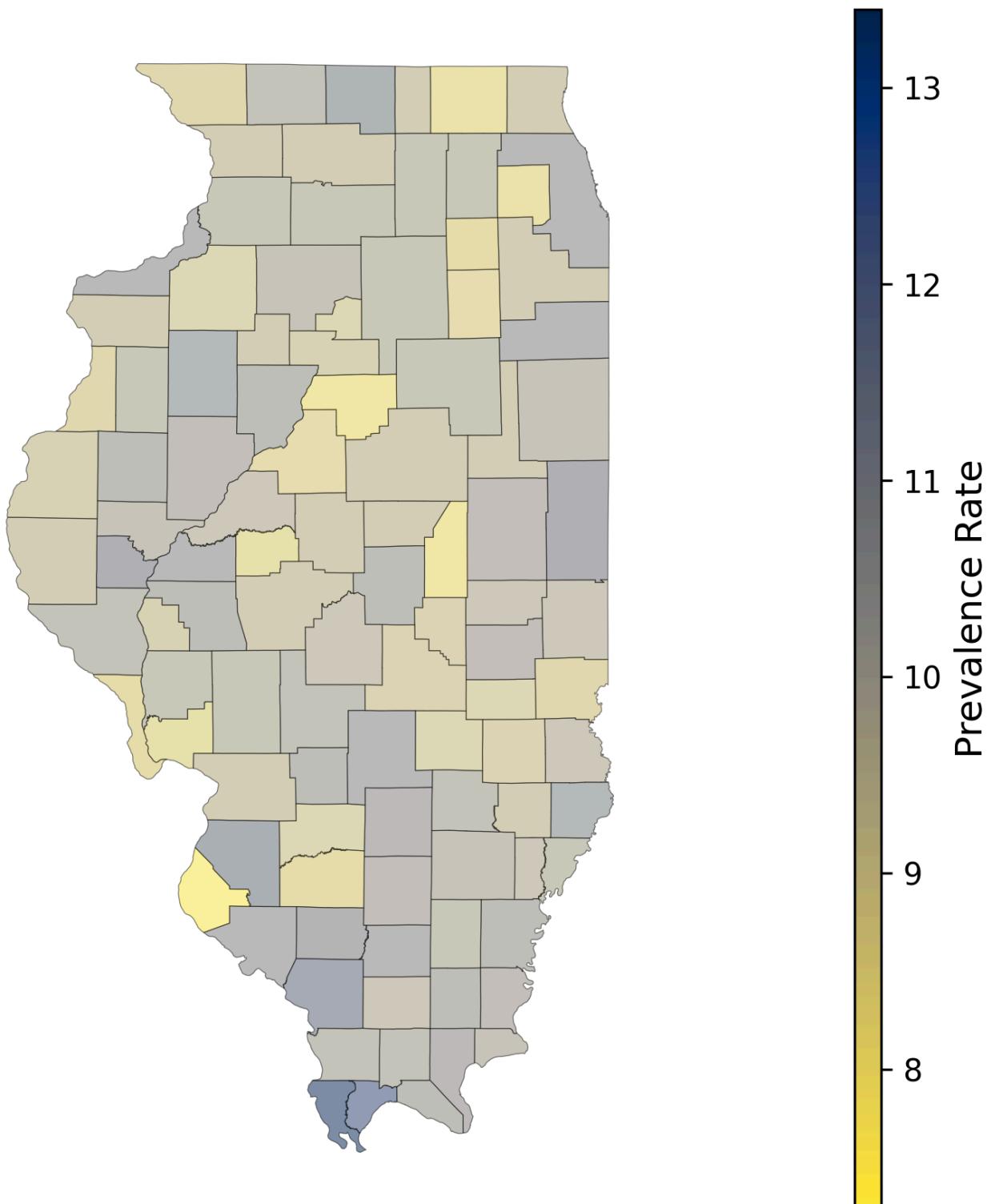
# DENTAL\_AdjPrev - Illinois Map by County



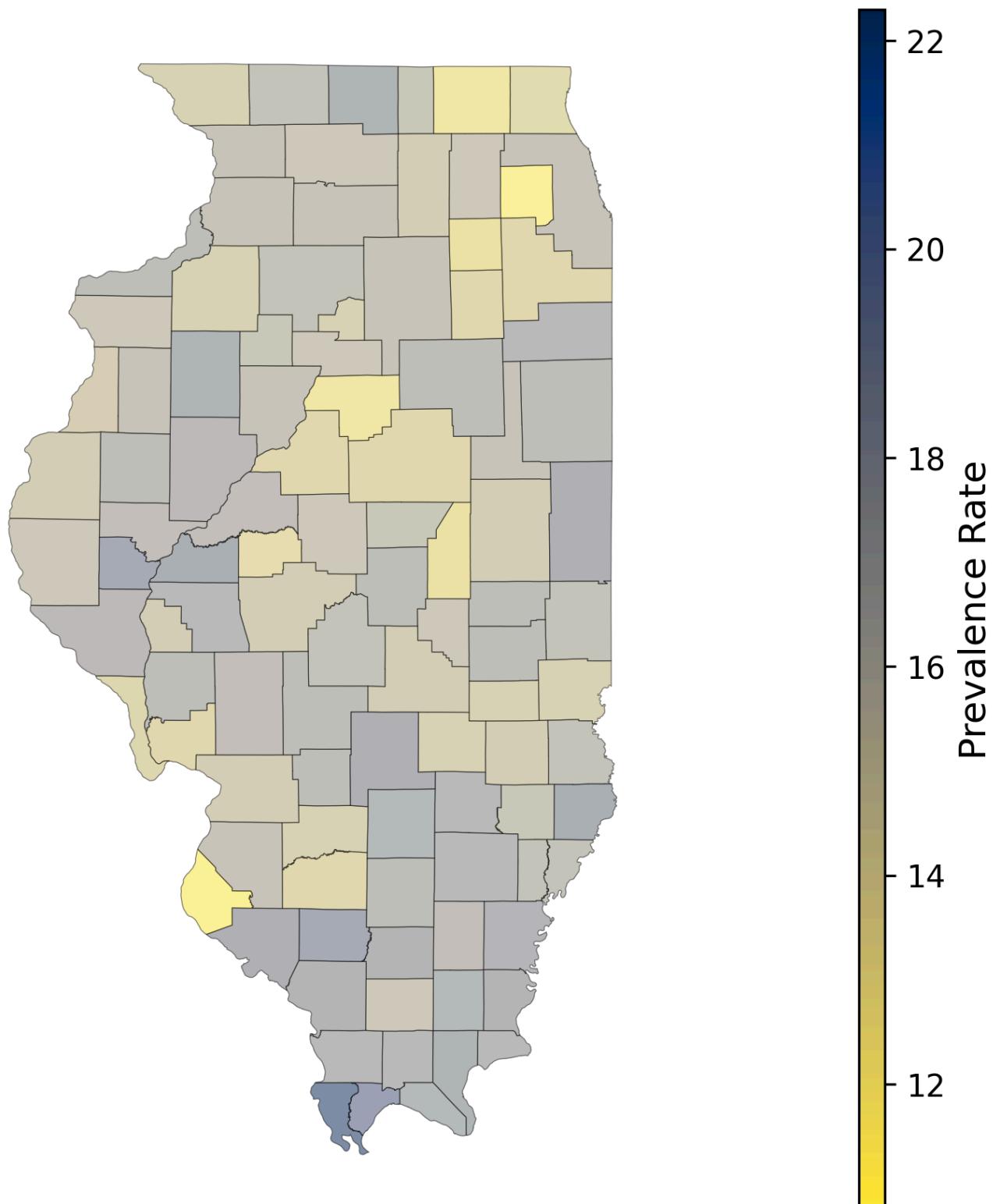
# DEPRESSION\_AdjPrev - Illinois Map by County



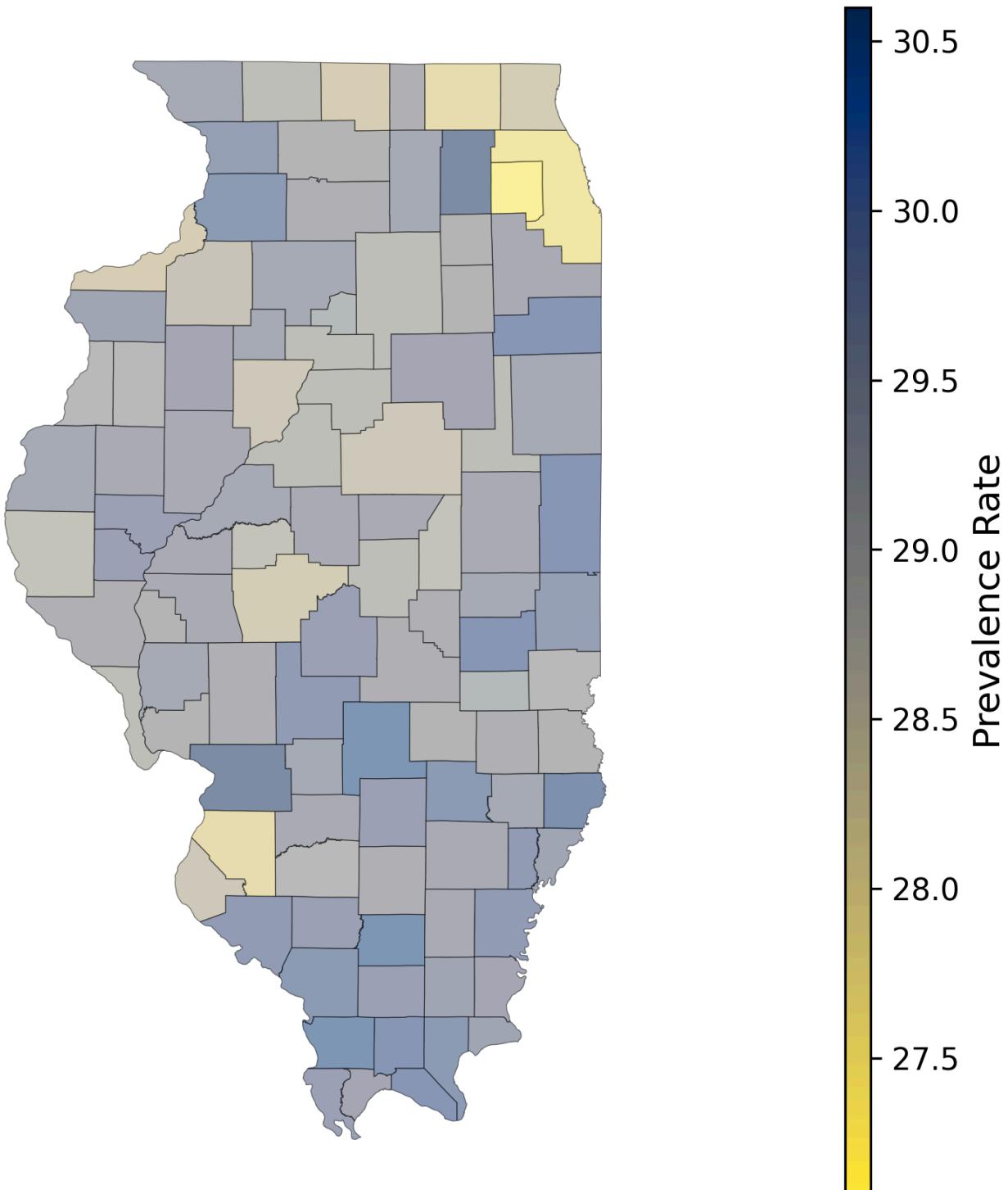
# DIABETES\_AdjPrev - Illinois Map by County



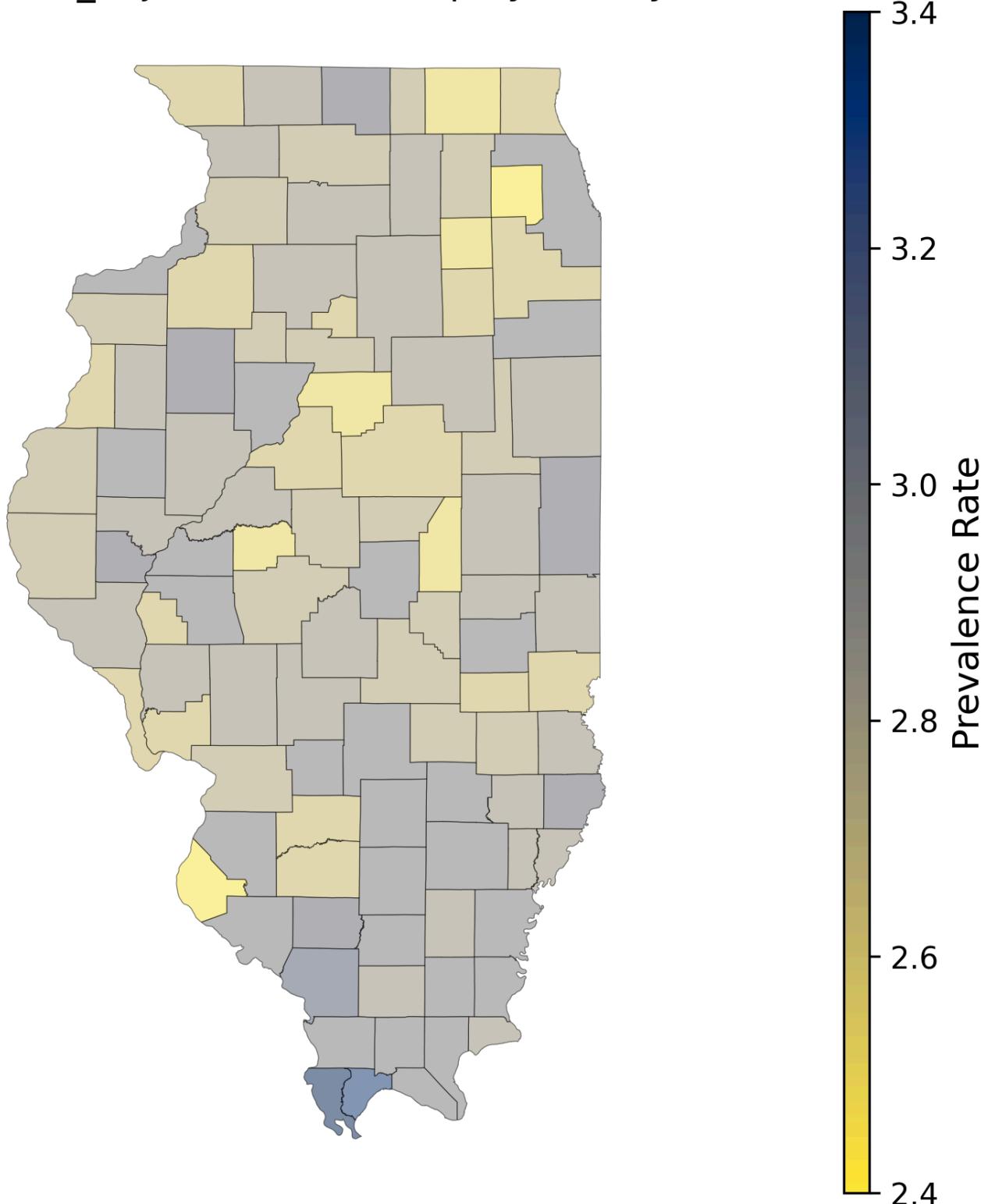
# GHLTH\_AdjPrev - Illinois Map by County



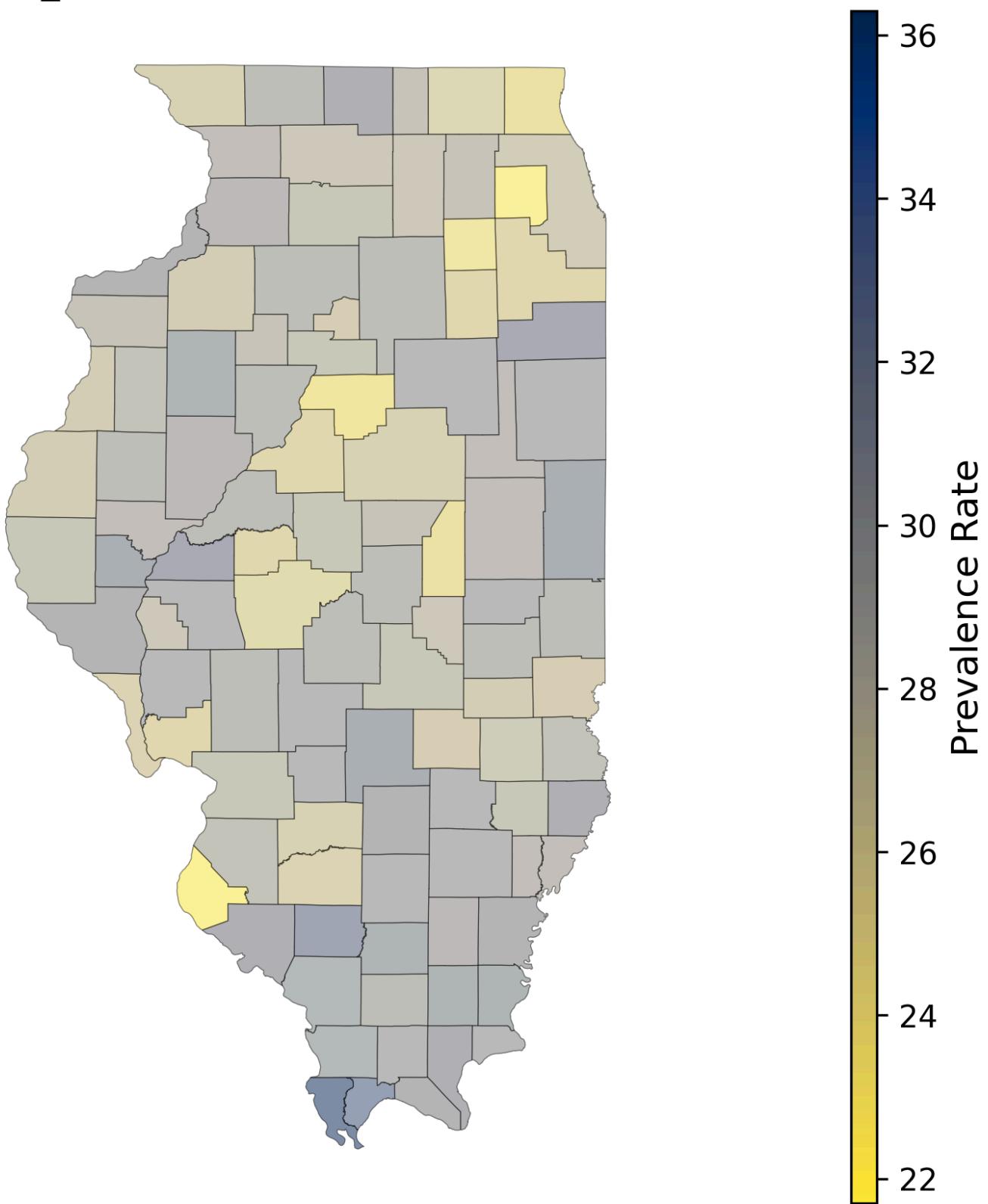
# HIGHCHOL\_AdjPrev - Illinois Map by County



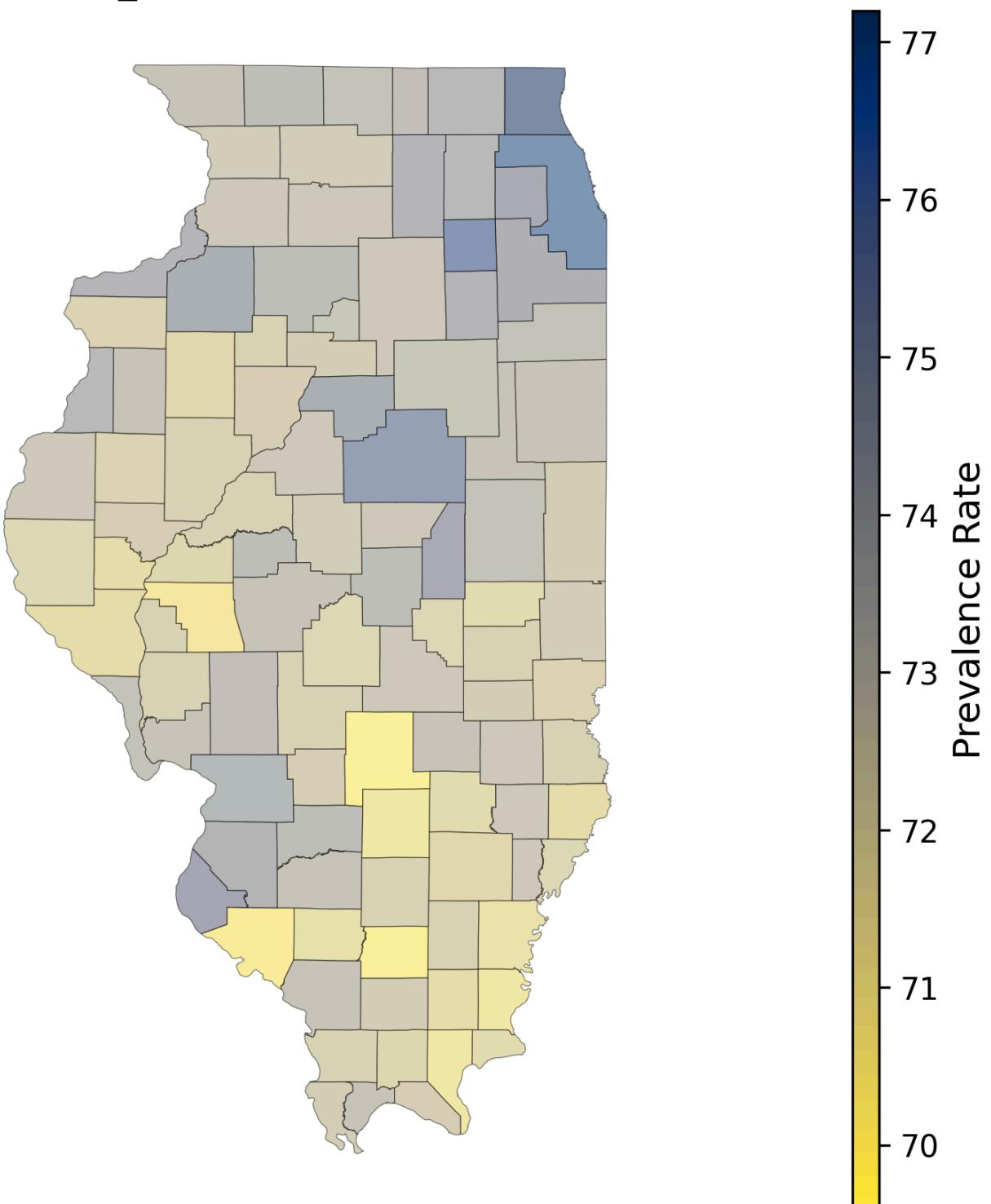
# KIDNEY\_AdjPrev - Illinois Map by County



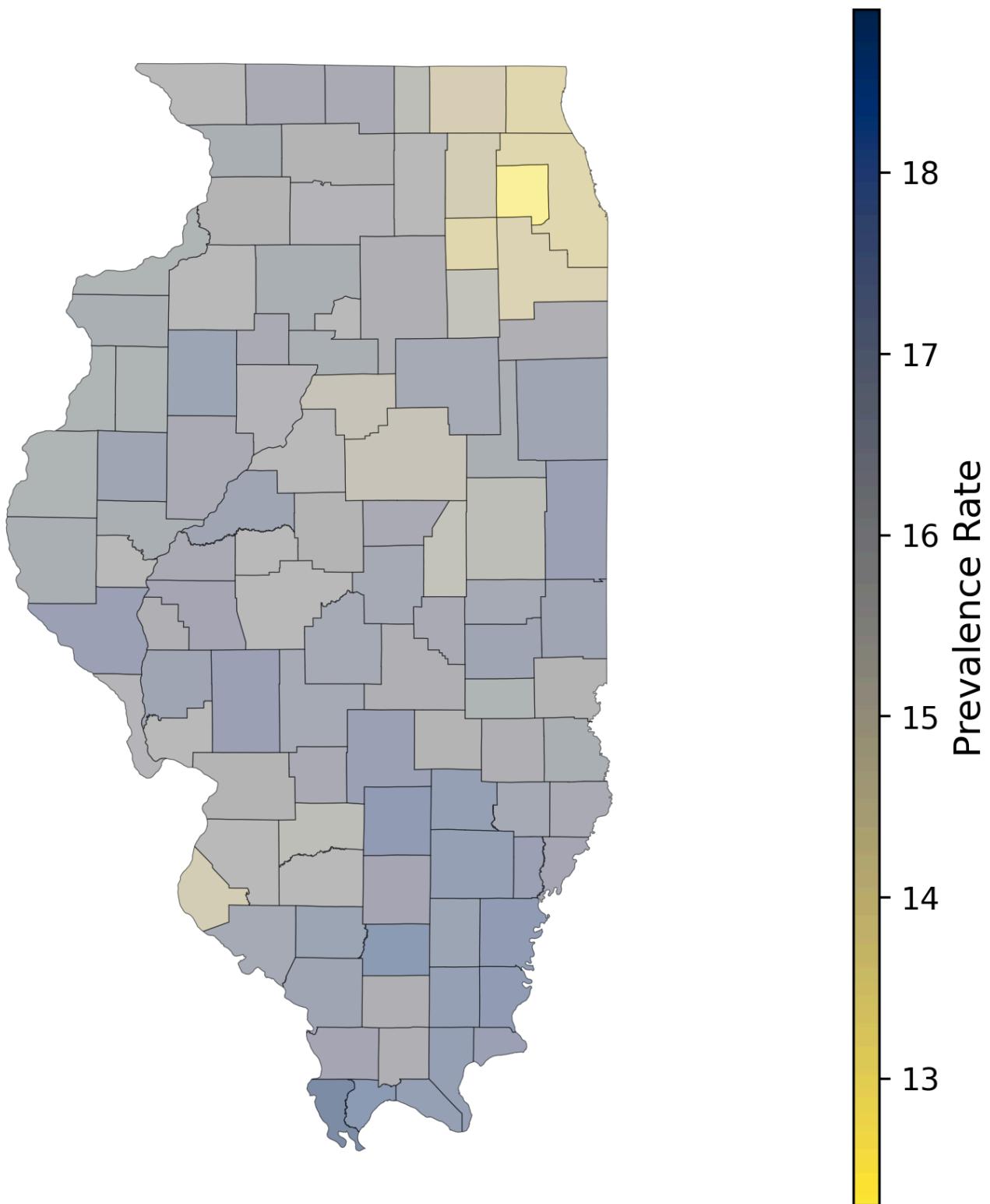
# LPA\_AdjPrev - Illinois Map by County



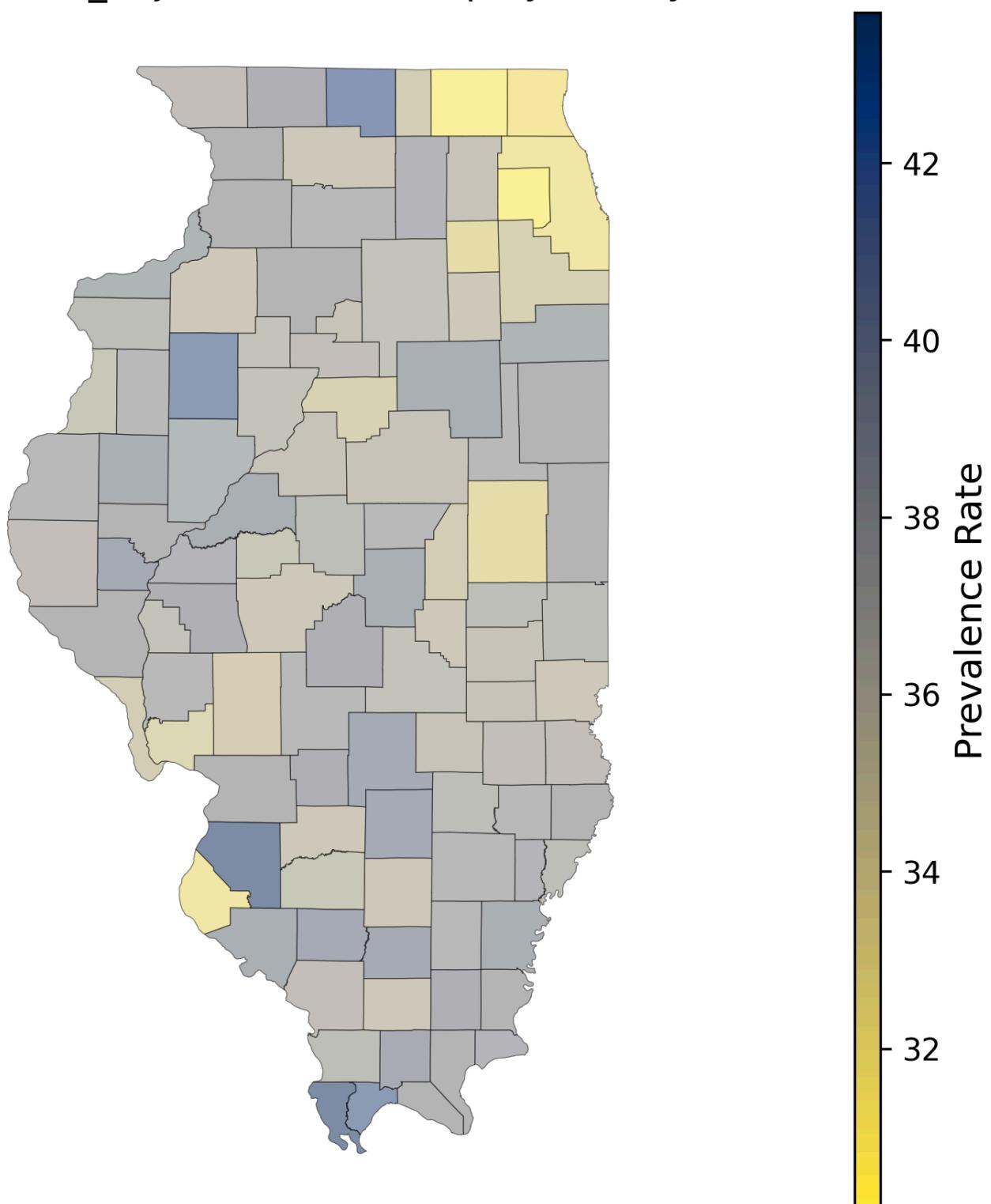
# MAMMOUSE\_AdjPrev - Illinois Map by County



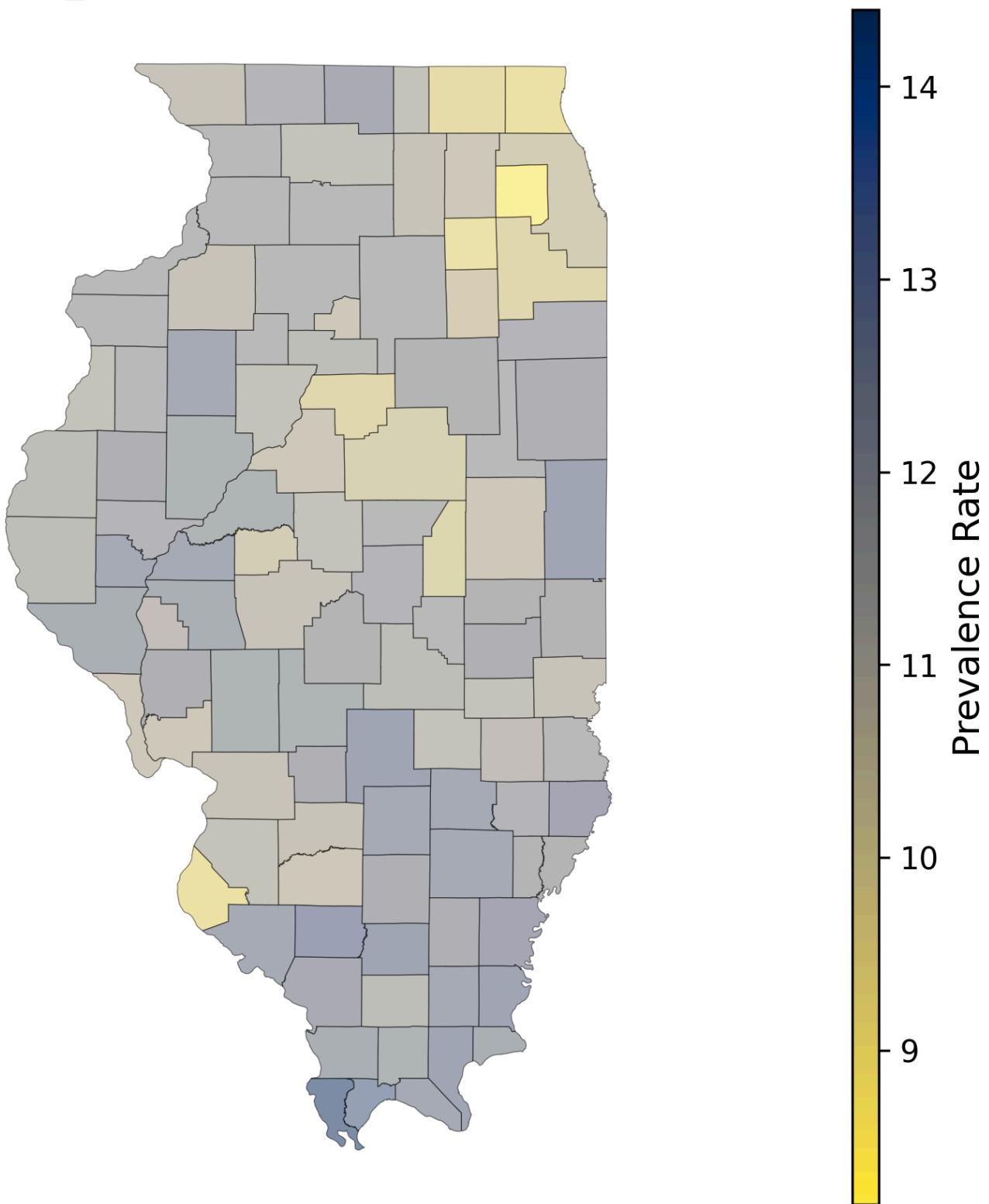
# MHLTH\_AdjPrev - Illinois Map by County



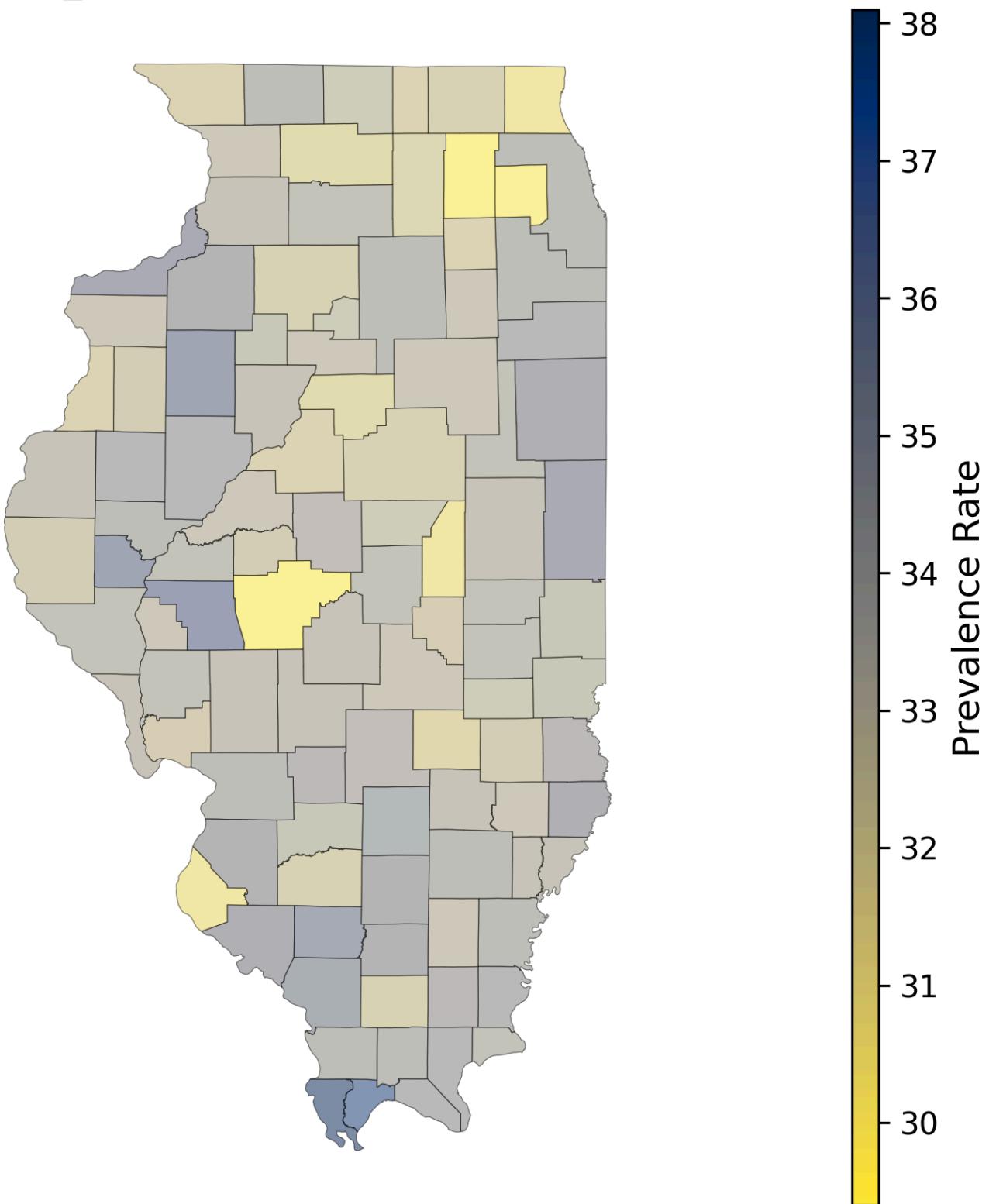
# OBESITY\_AdjPrev - Illinois Map by County



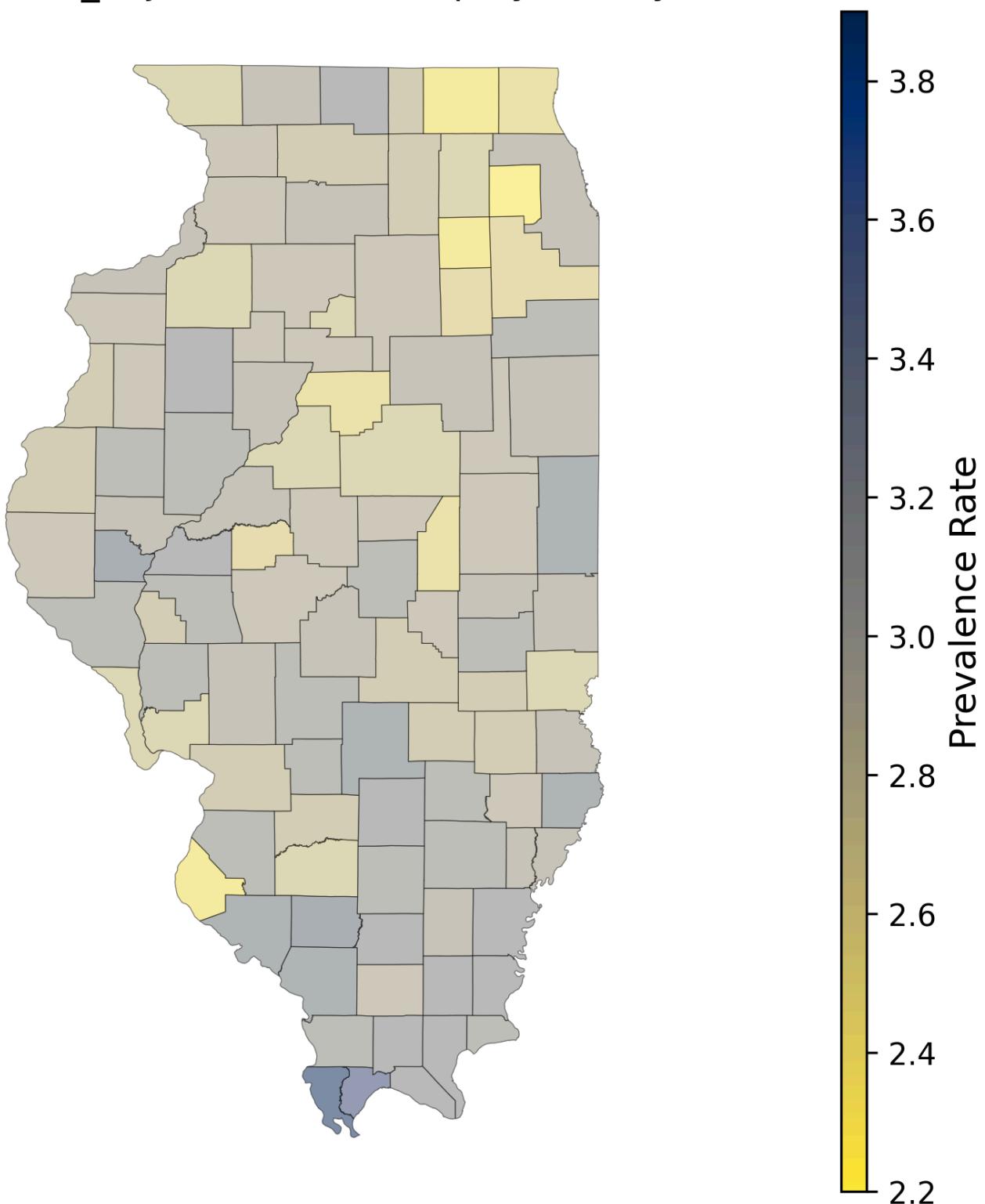
# PHLTH\_AdjPrev - Illinois Map by County



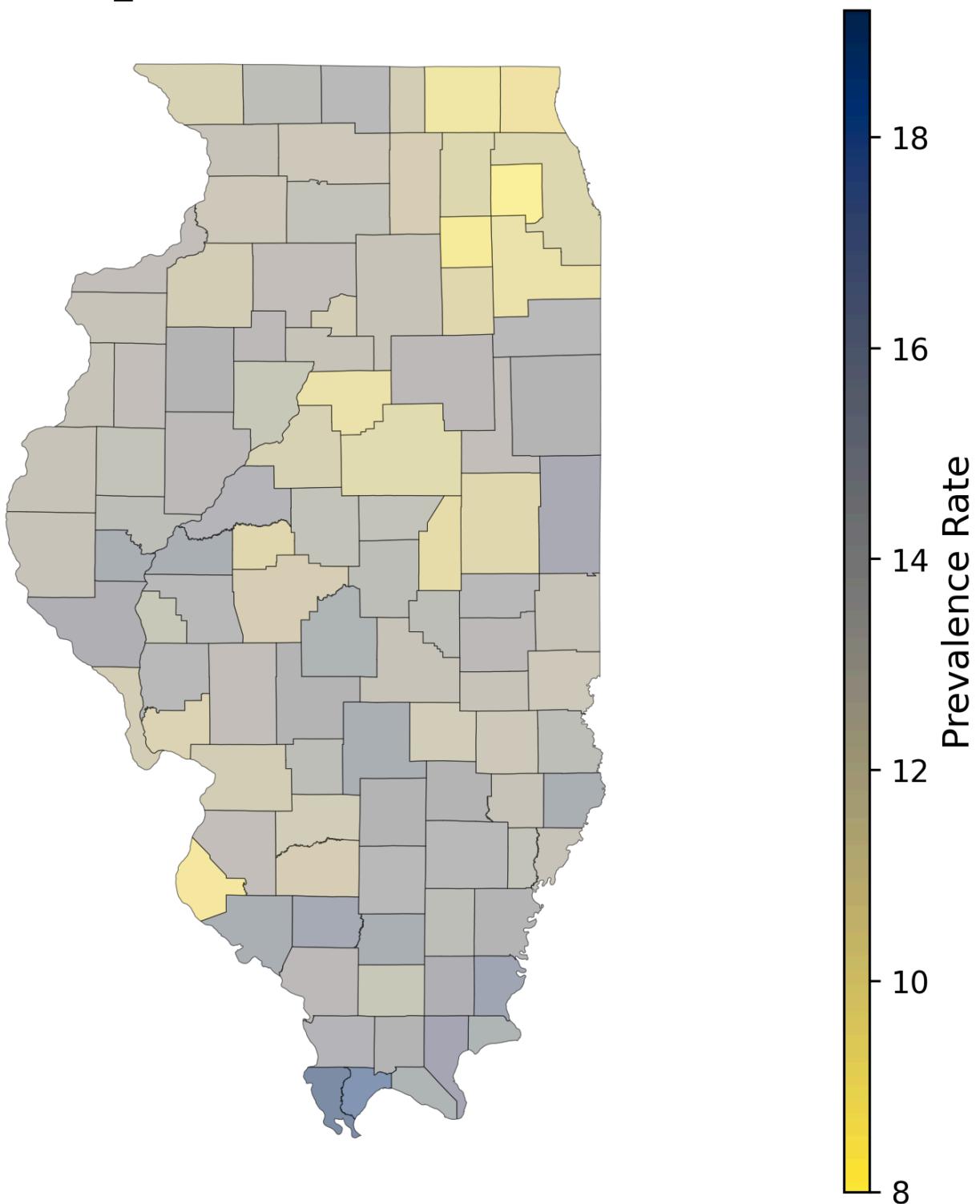
# SLEEP\_AdjPrev - Illinois Map by County



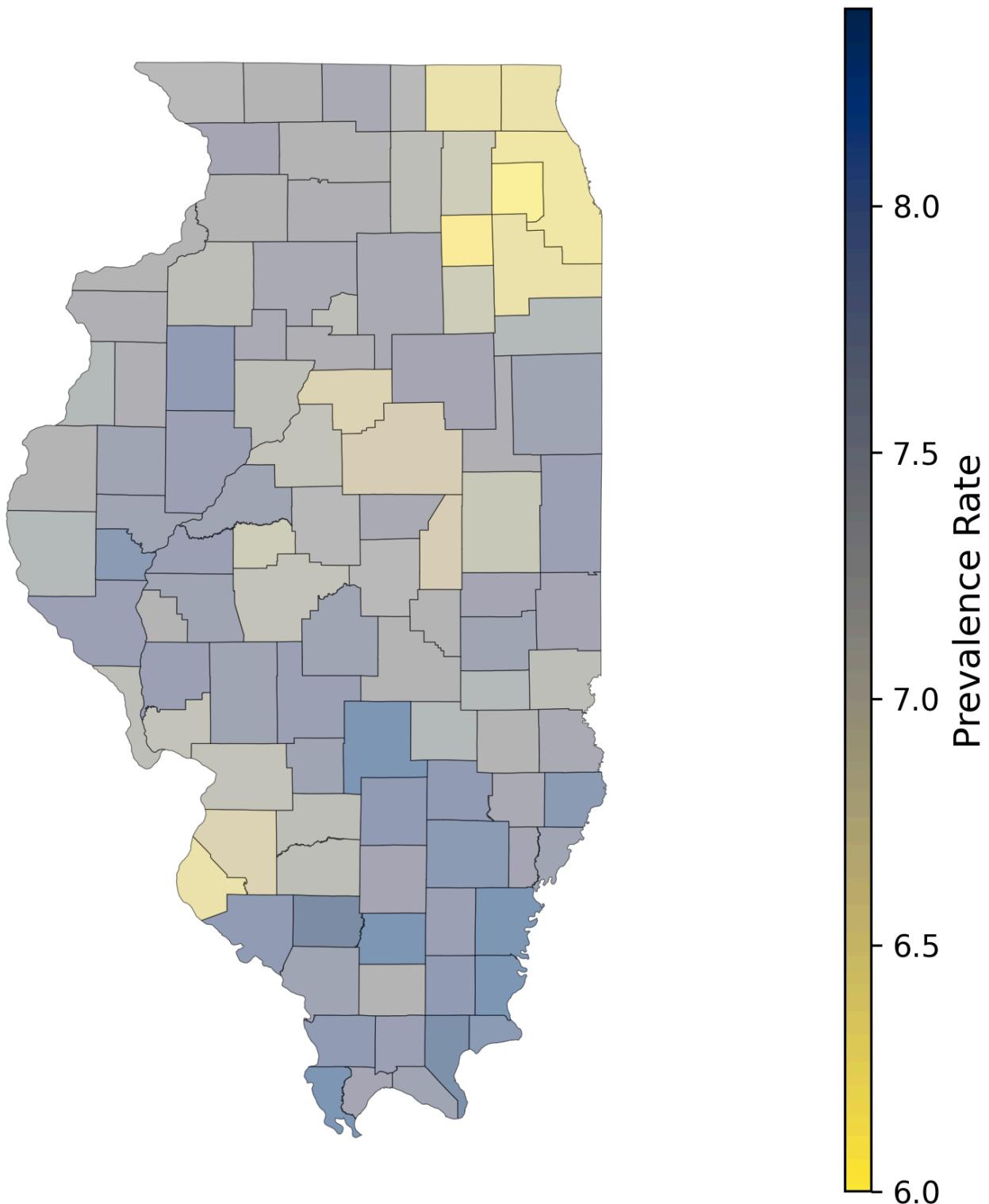
# STROKE\_AdjPrev - Illinois Map by County



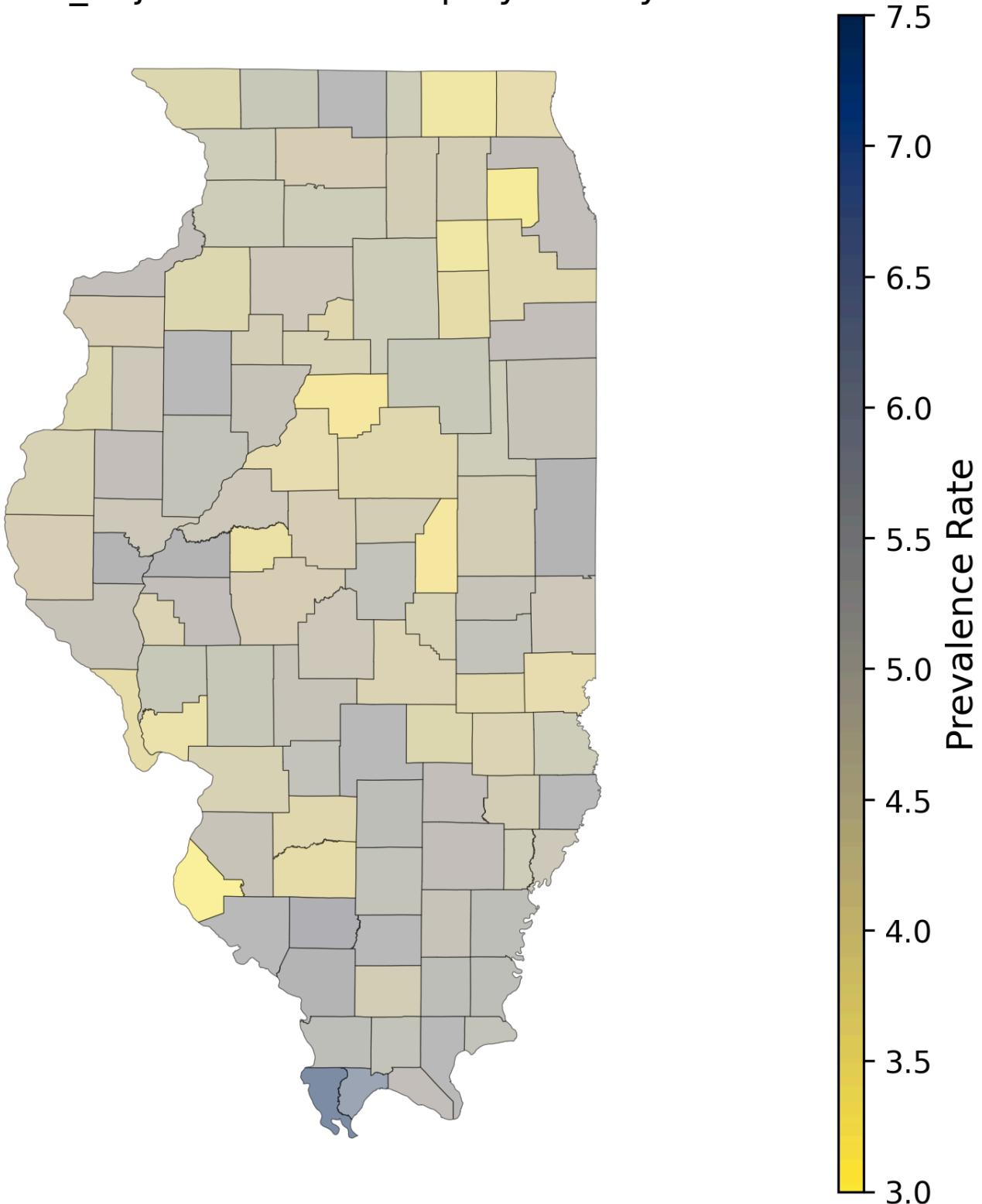
# TEETHLOST\_AdjPrev - Illinois Map by County



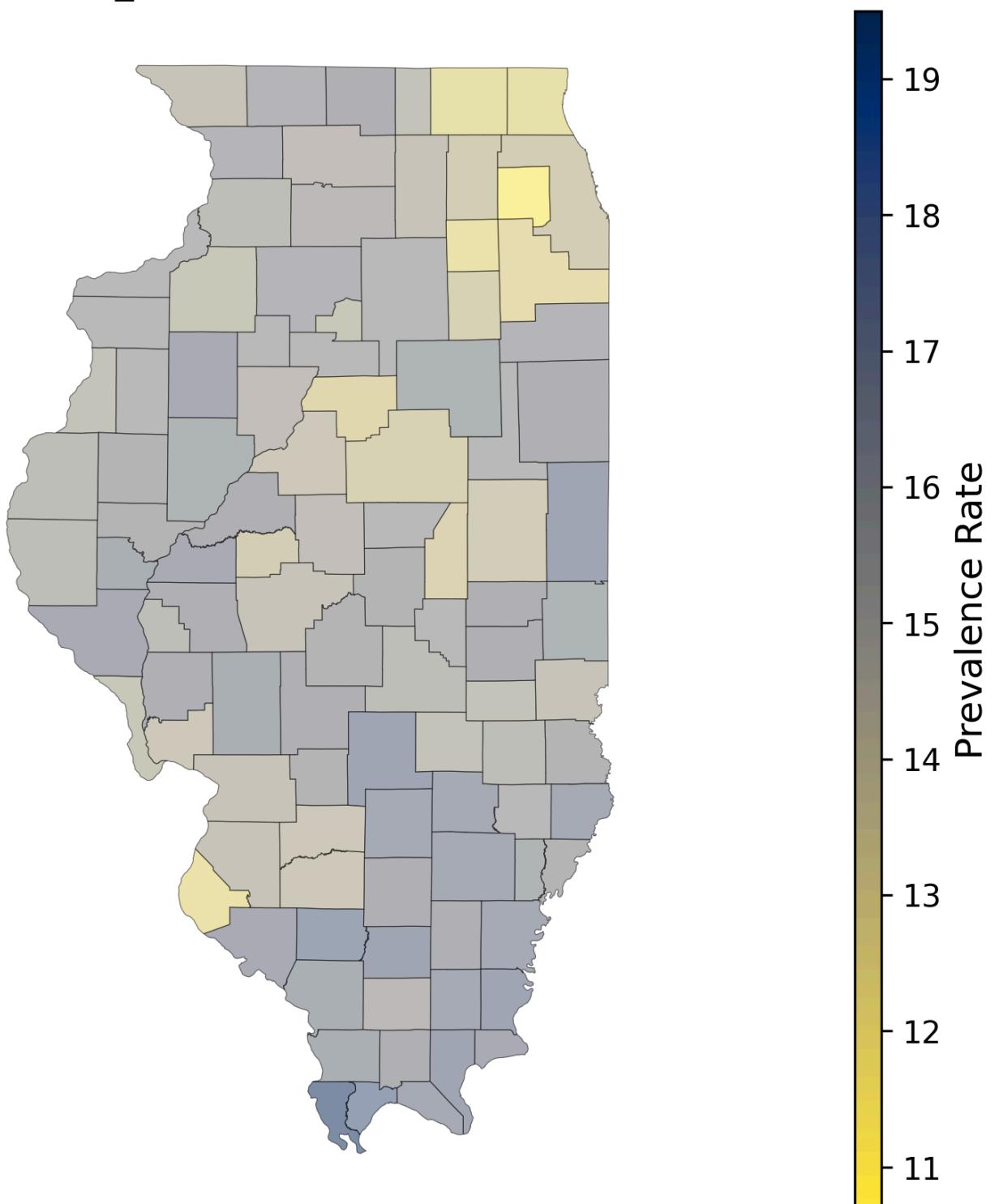
# HEARING\_AdjPrev - Illinois Map by County



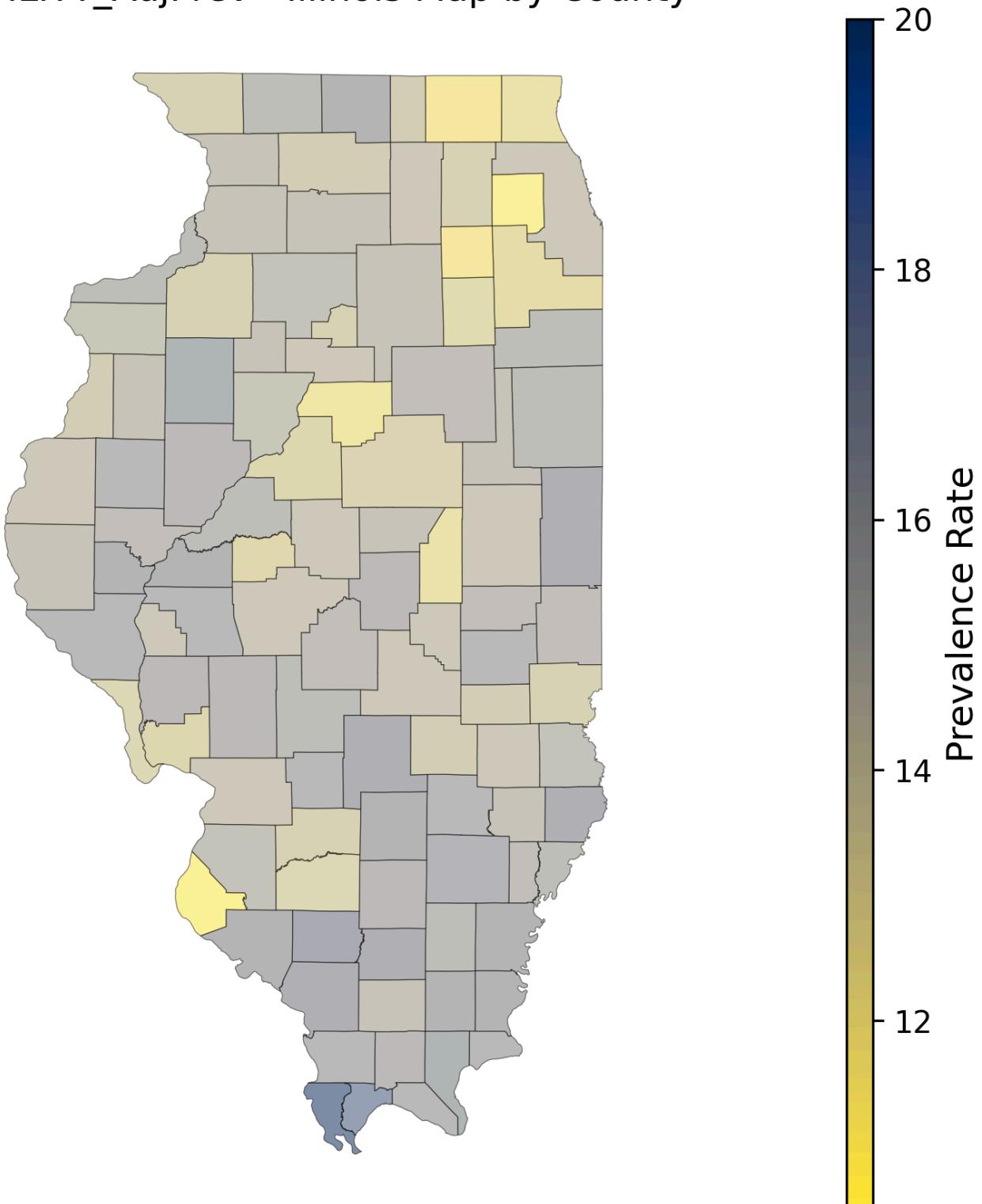
# VISION\_AdjPrev - Illinois Map by County



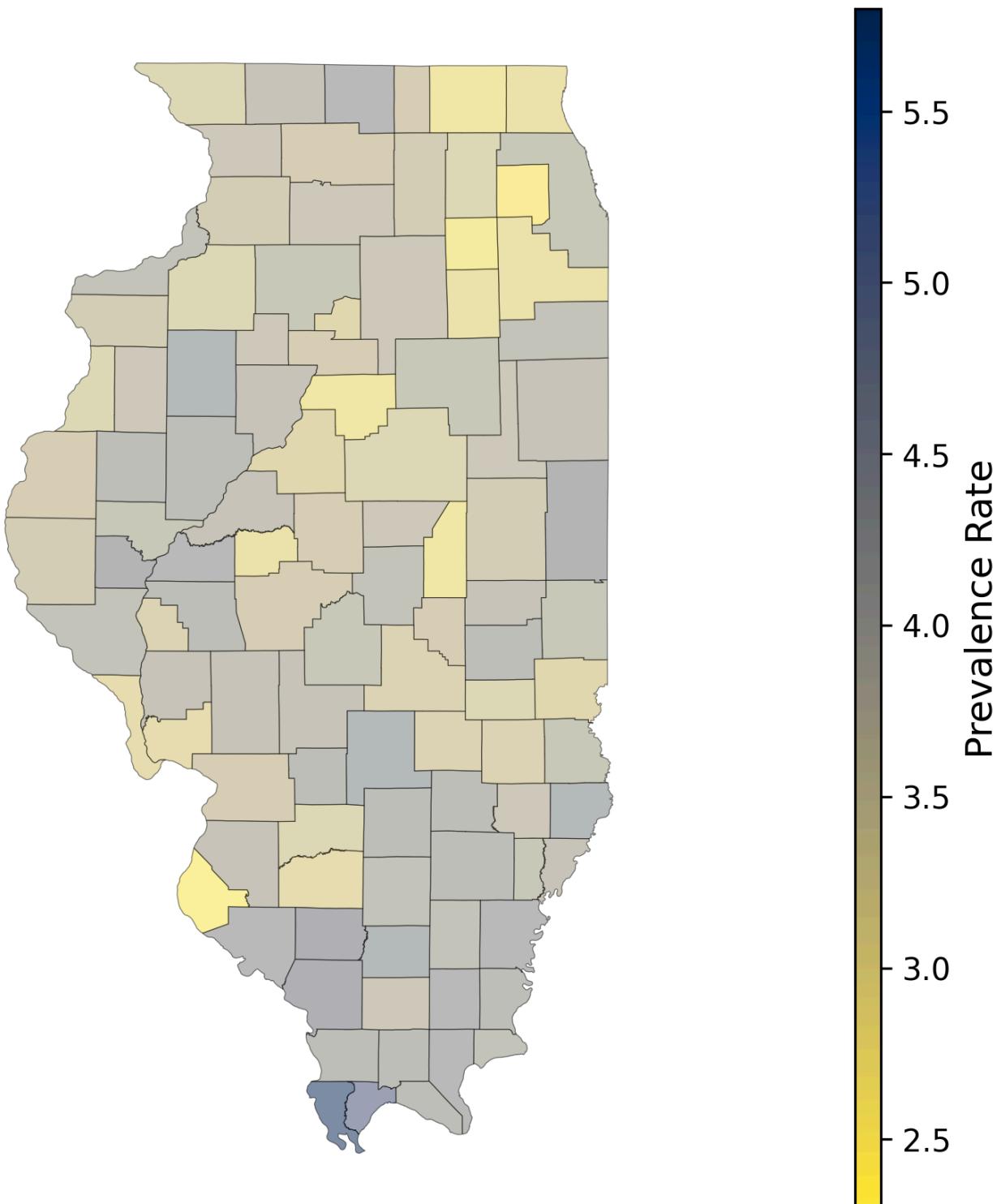
## COGNITION\_AdjPrev - Illinois Map by County



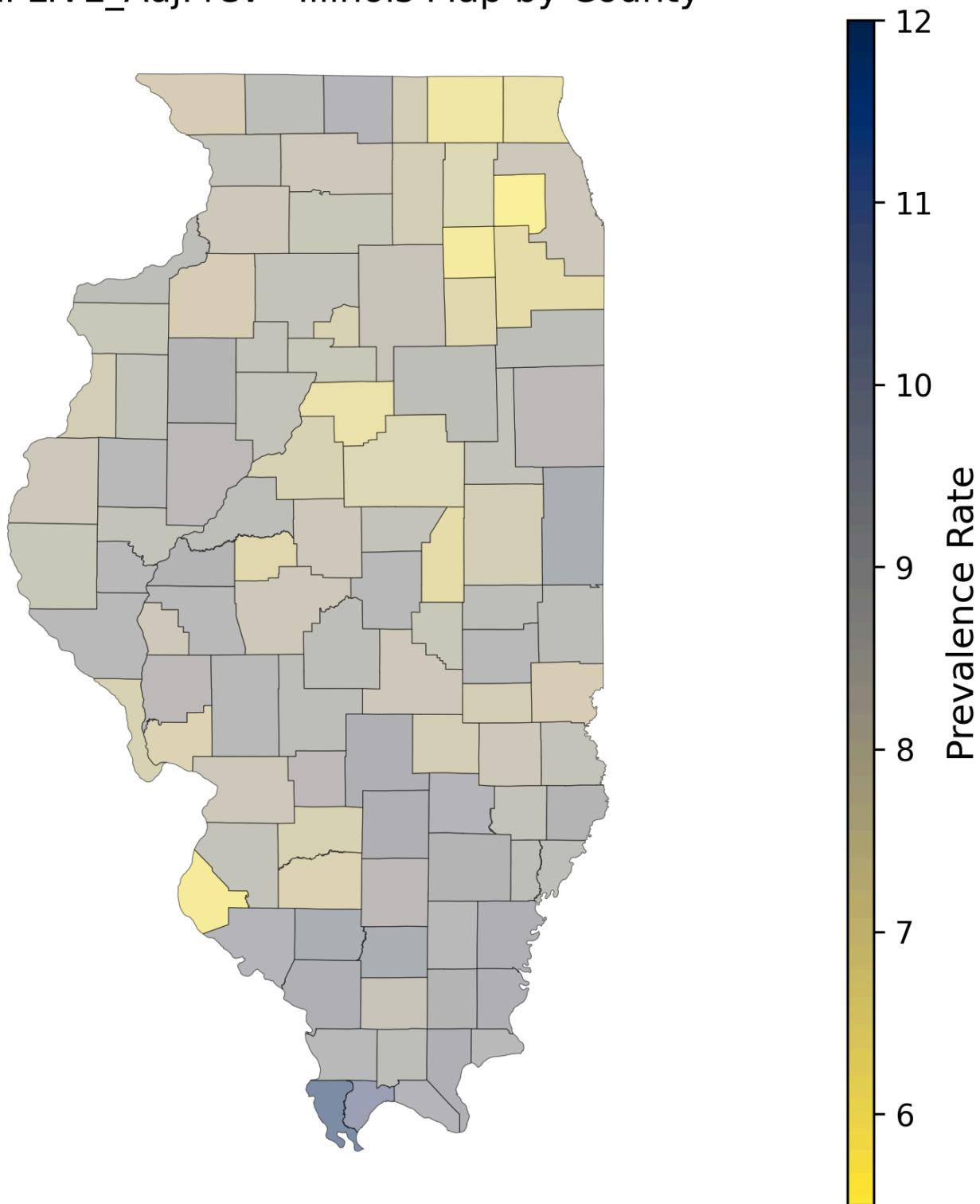
# MOBILITY\_AdjPrev - Illinois Map by County



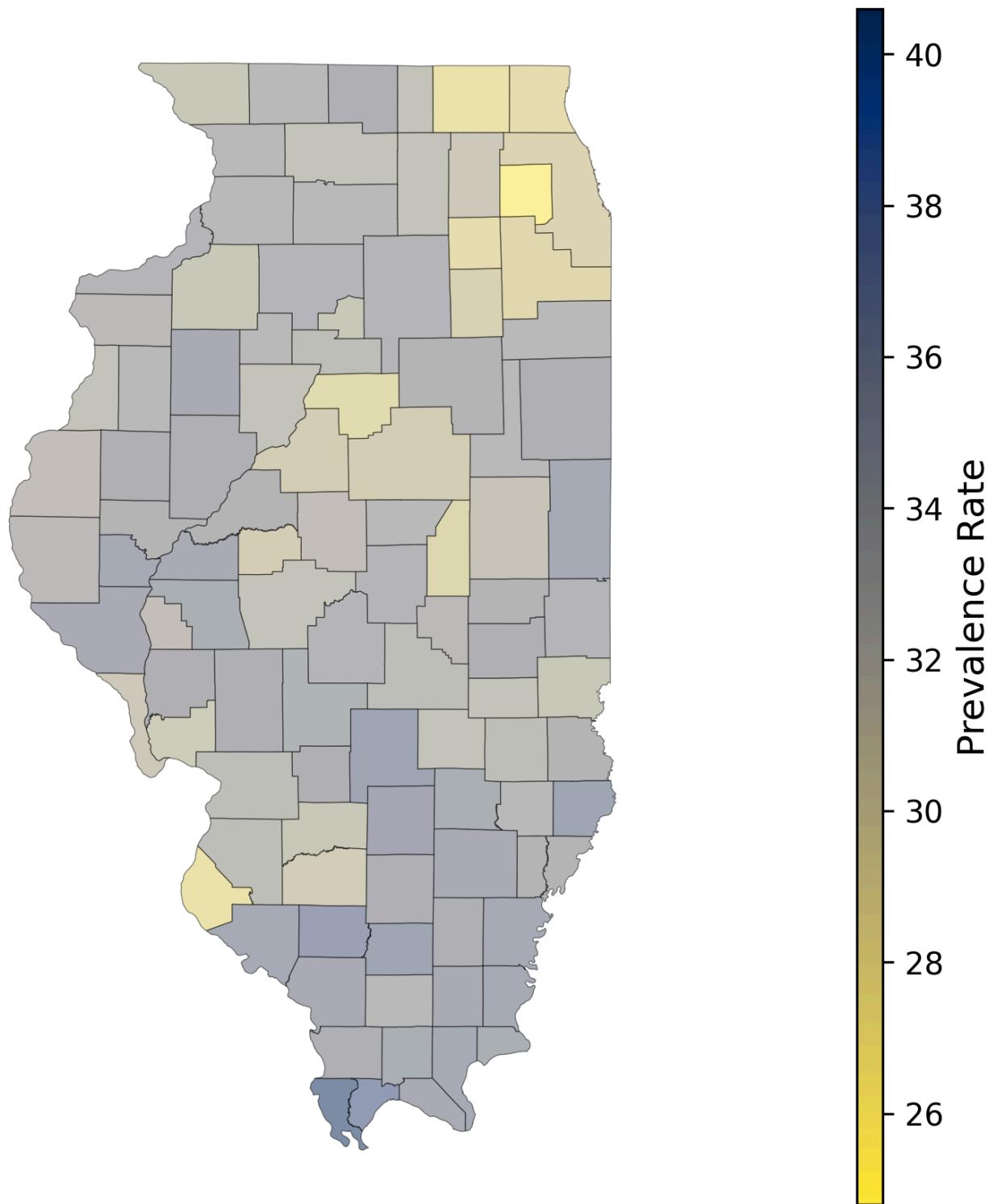
# SELF CARE\_AdjPrev - Illinois Map by County



# INDEPLIVE\_AdjPrev - Illinois Map by County



# DISABILITY\_AdjPrev - Illinois Map by County



## US Map at County Level

### Import Shapefile

```
In [ ]: path = "C:/Users/user/Desktop/GIS/Final Project/tl_2023_us_county/tl_2023_us_county.shp"
county_us = gpd.read_file(path)
county_us = county_us.to_crs("EPSG:4326")
```

```
In [ ]: # Define STATEFP values for non-continental states
non_continental_statefp = ['15', '78', '69', '66', '02', '60', '72']

# Filter out non-continental states from county_us
county_us = county_us[~county_us['STATEFP'].isin(non_continental_statefp)]
```

### Preprocess and Merge

```
In [ ]: # Add Leading zero to CountyFIPS if it has only four digits  
df['CountyFIPS'] = df['CountyFIPS'].astype(str)  
df['CountyFIPS'] = df['CountyFIPS'].apply(lambda x: x.zfill(5))
```

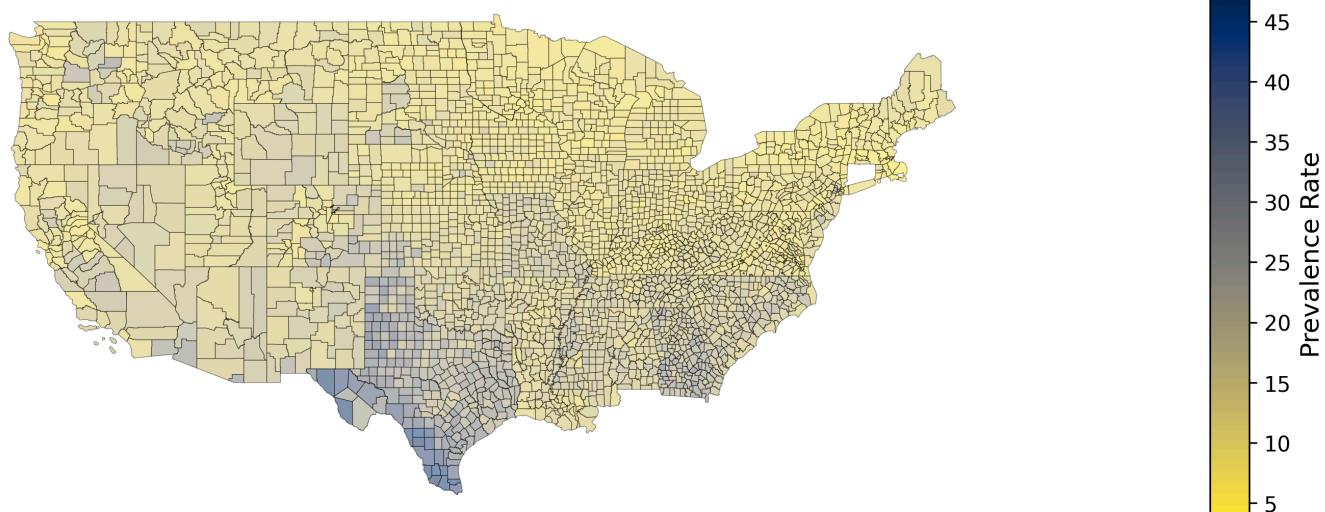
```
In [ ]: # Merge aggregated DataFrame with the GeoDataFrame  
merged_gdf = county_us.merge(df, left_on=['GEOID'], right_on=['CountyFIPS'], how='left')
```

## Plotting

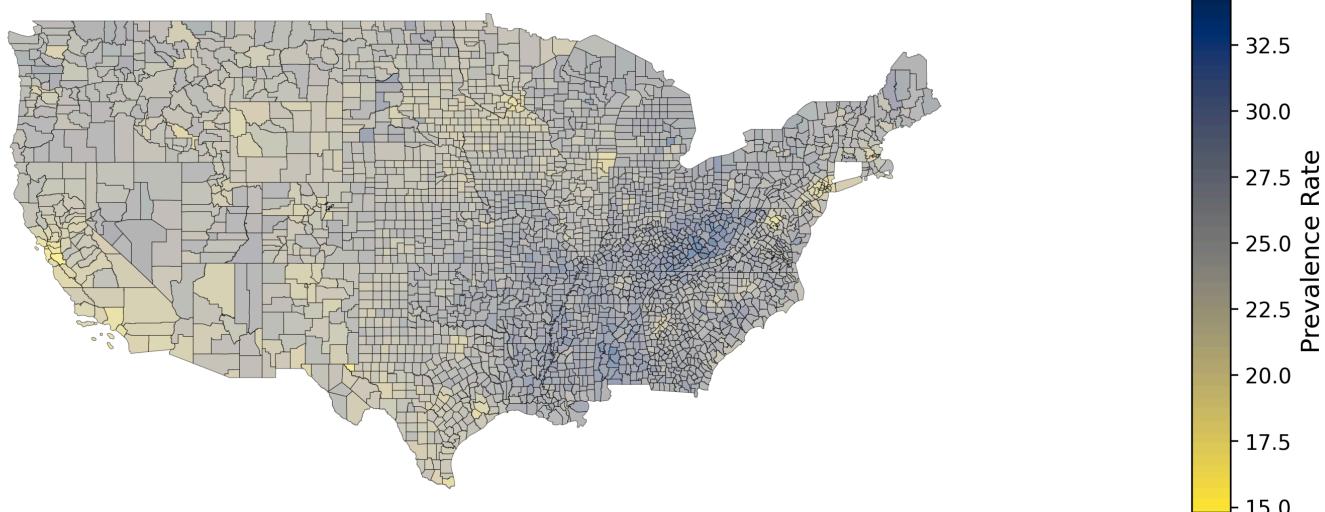
```
In [ ]: import matplotlib.pyplot as plt  
from mpl_toolkits.axes_grid1 import make_axes_locatable  
  
def plot_normalized_county_data(geo_df, data_columns, colormap):  
    for column in data_columns:  
        # Plotting  
        f, ax = plt.subplots(1, 1, figsize=(10, 6), sharex=True, sharey=True, dpi=300)  
        f.tight_layout()  
        plt.title(f'{column} - US Map by County')  
        ax.set_axis_off()  
        divider = make_axes_locatable(ax)  
        cax = divider.append_axes("right", size="3%", pad=0.5, alpha=0.5)  
        geo_df.plot(column, ax=ax, alpha=0.5, cmap=colormap, edgecolor='k', legend=True, cax=cax, linewidth=0.3)  
        plt.ylabel('Prevalence Rate', fontsize=12)  
        plt.show()
```

```
In [ ]: # Get columns ending with '_CrudePrev'  
columns_to_plot = [col for col in df.columns if col.endswith('_AdjPrev')]  
  
plot_normalized_county_data(merged_gdf, columns_to_plot, 'cividis_r')
```

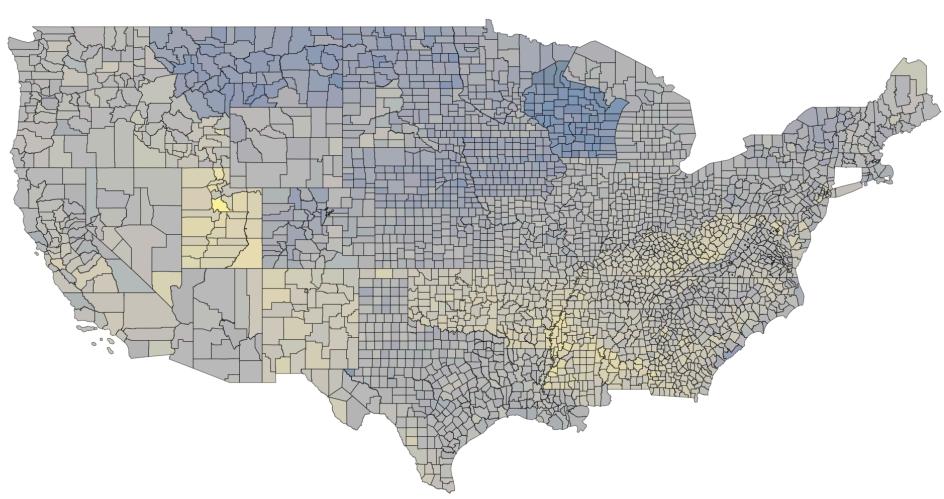
ACCESS2\_AdjPrev - US Map by County



ARTHRITIS\_AdjPrev - US Map by County



BINGE\_AdjPrev - US Map by County

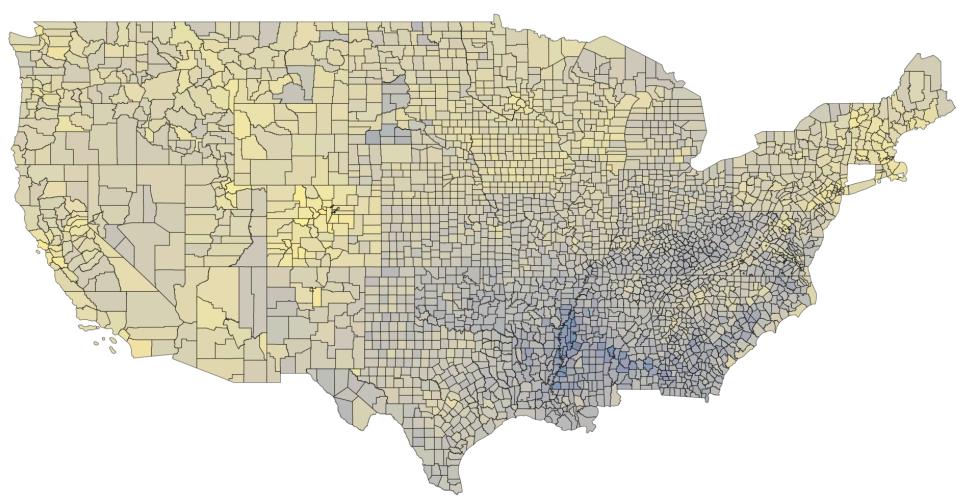


Prevalence Rate

Prevalence Rate

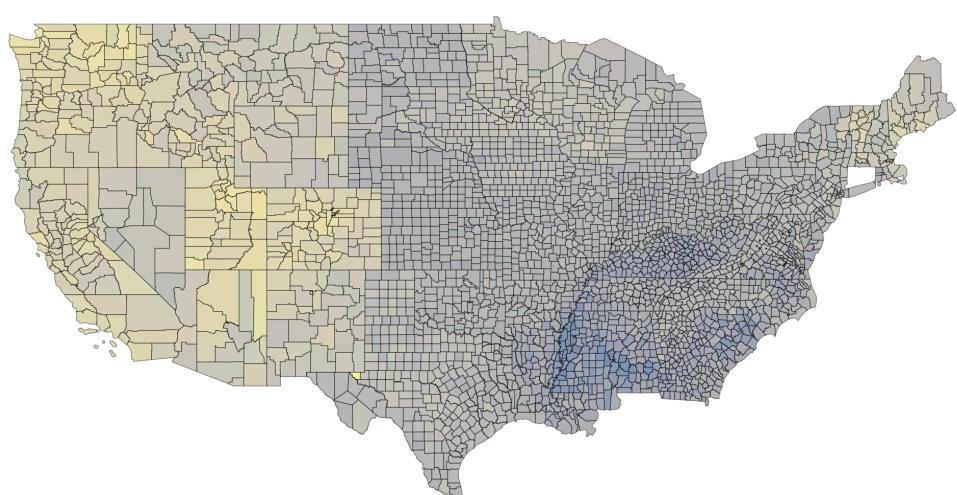
Prevalence Rate

BPHIGH\_AdjPrev - US Map by County



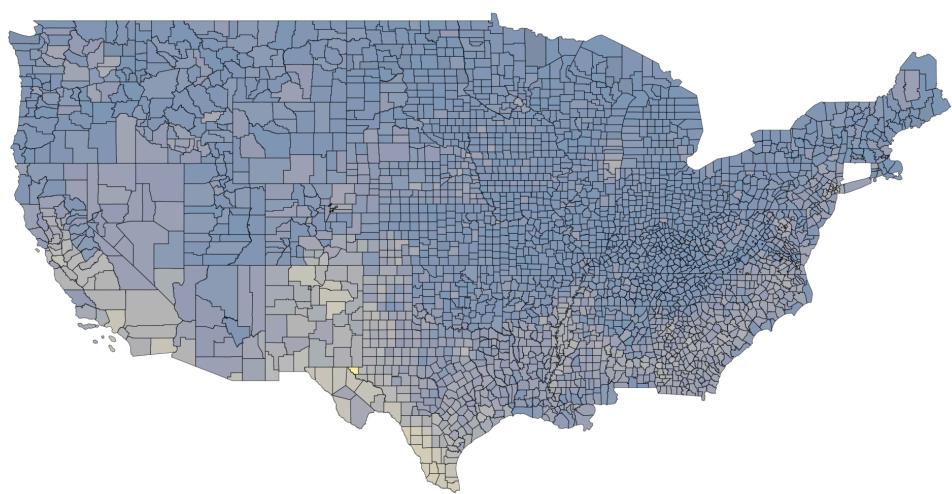
Prevalence Rate

BPMED\_AdjPrev - US Map by County

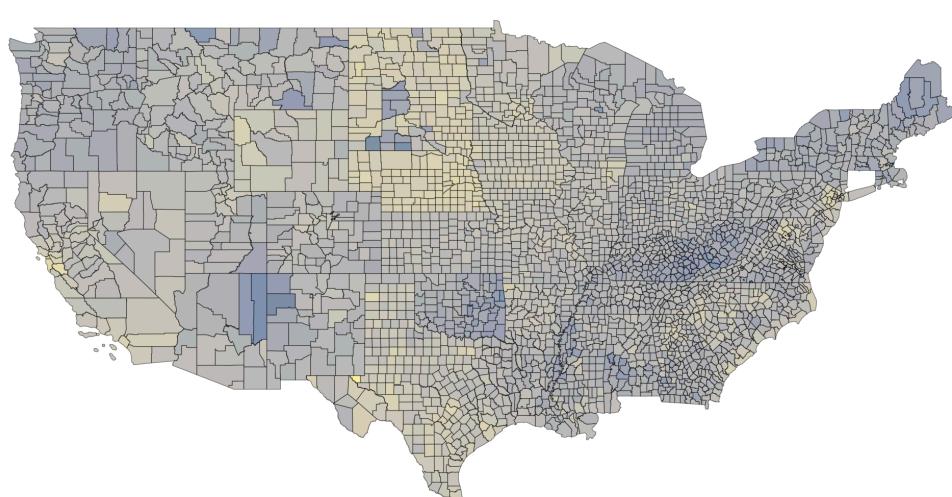


Prevalence Rate

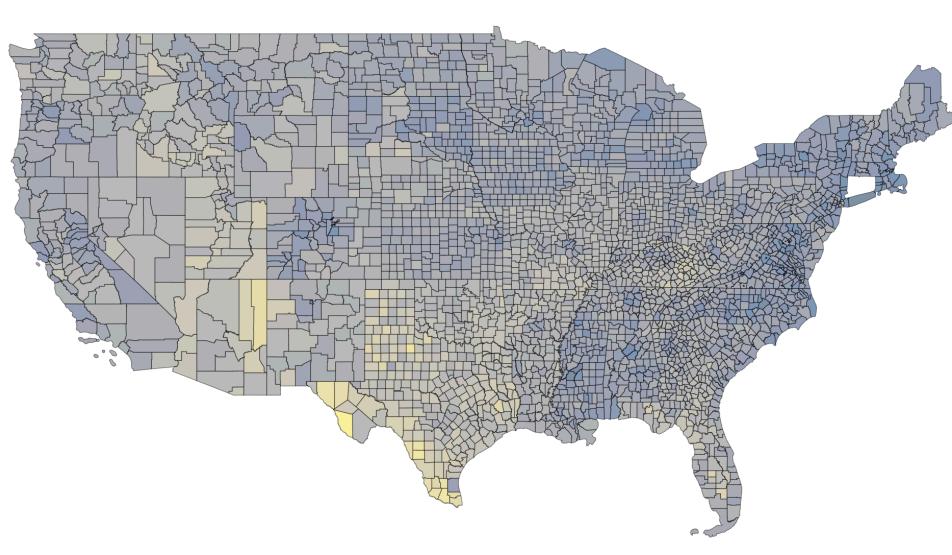
CANCER\_AdjPrev - US Map by County



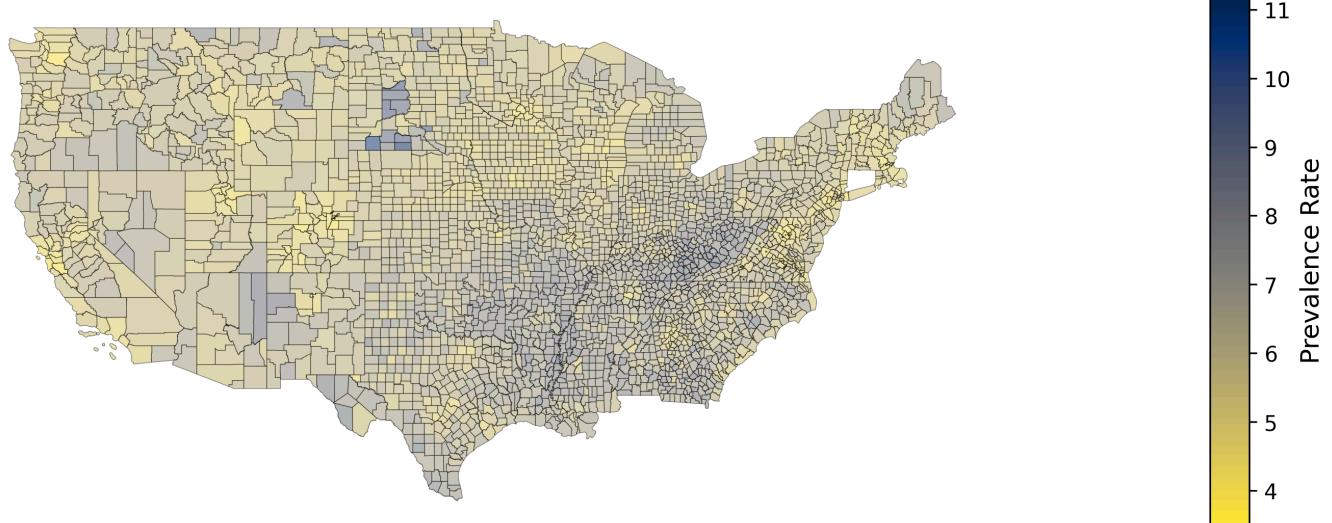
CASTHMA\_AdjPrev - US Map by County



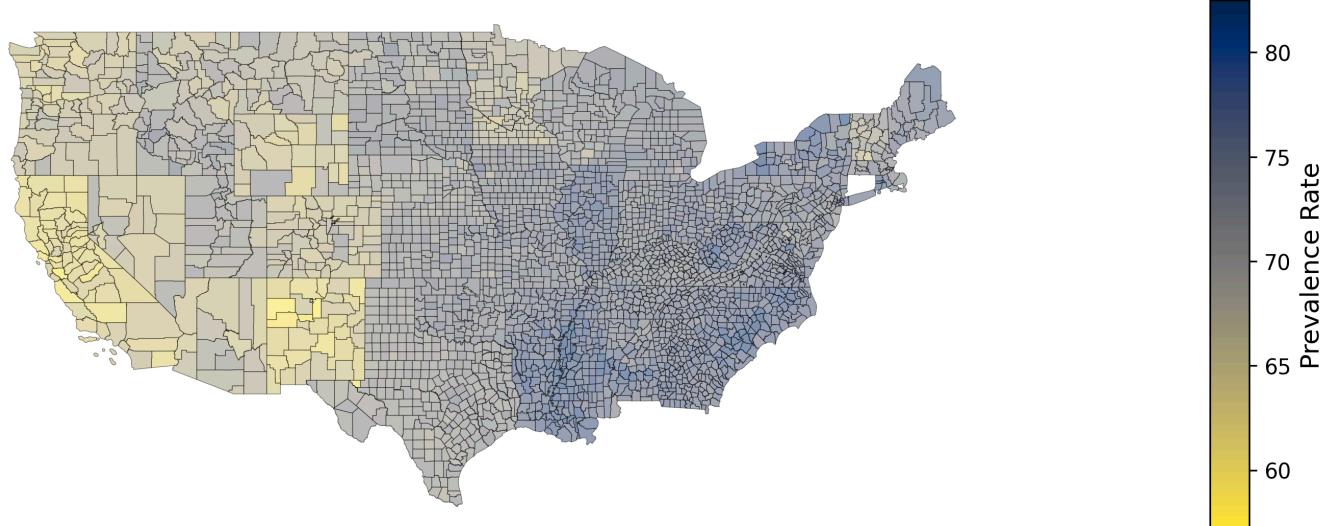
CERVICAL\_AdjPrev - US Map by County



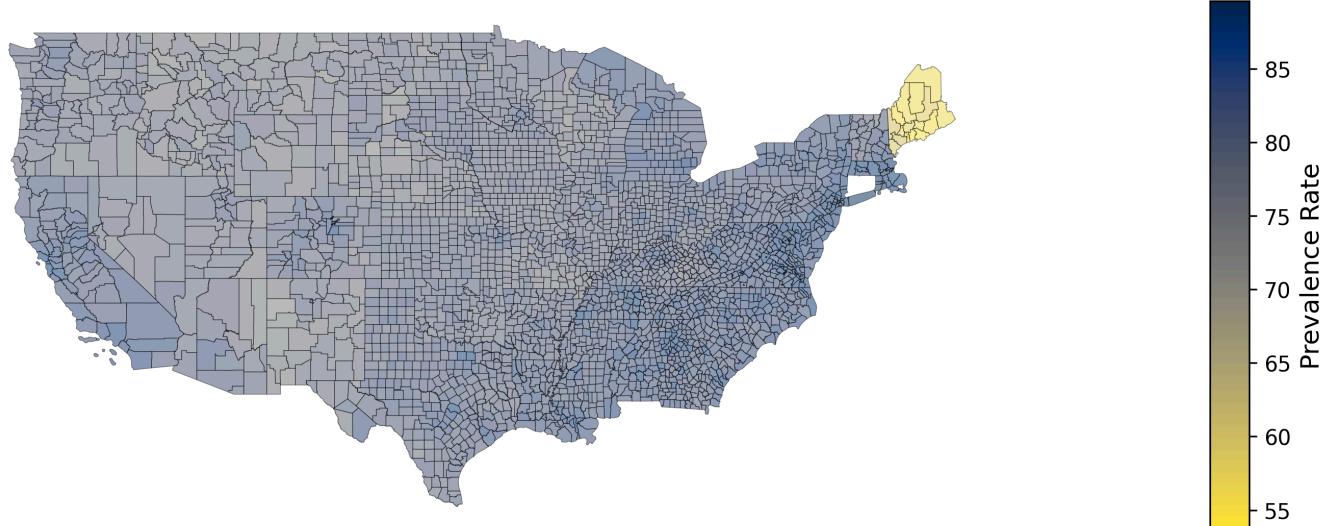
CHD\_AdjPrev - US Map by County



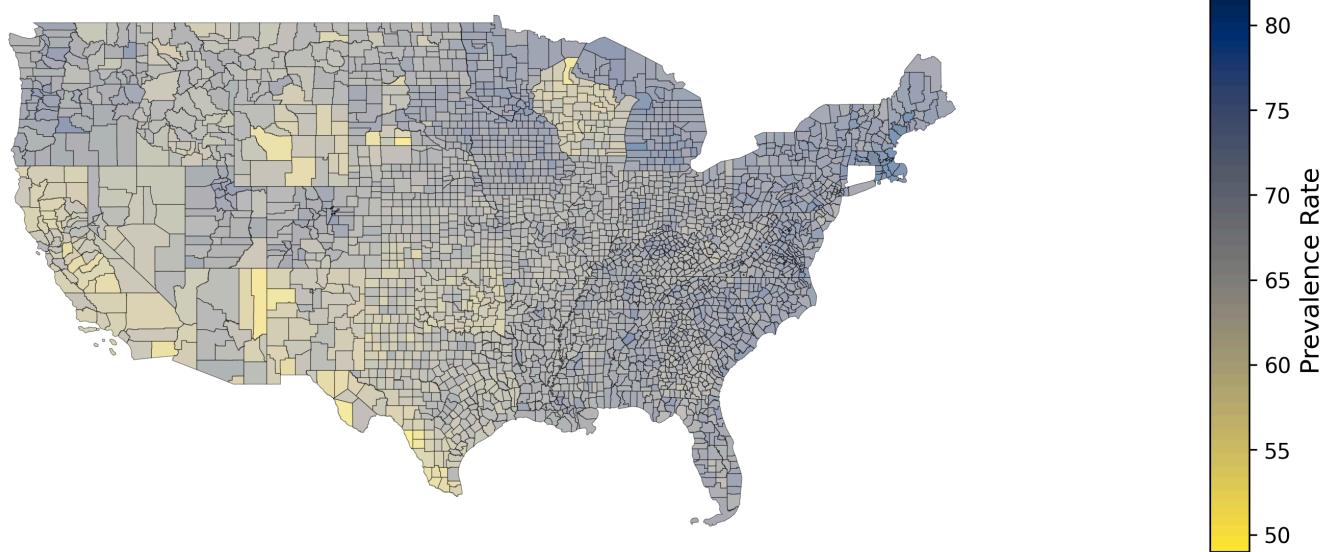
CHECKUP\_AdjPrev - US Map by County



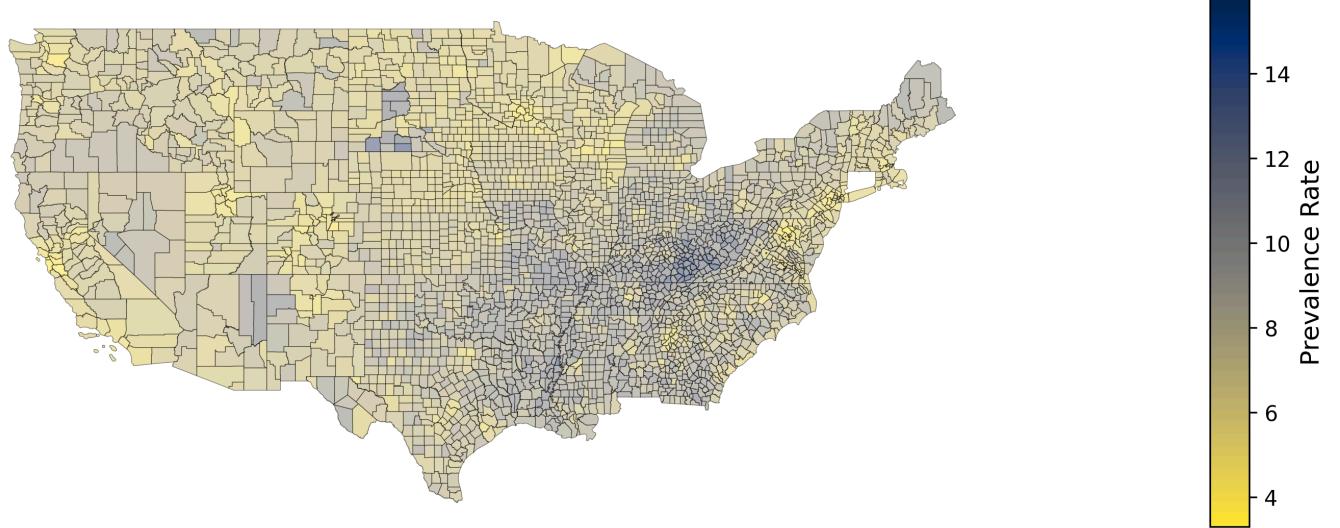
CHOLSCREEN\_AdjPrev - US Map by County



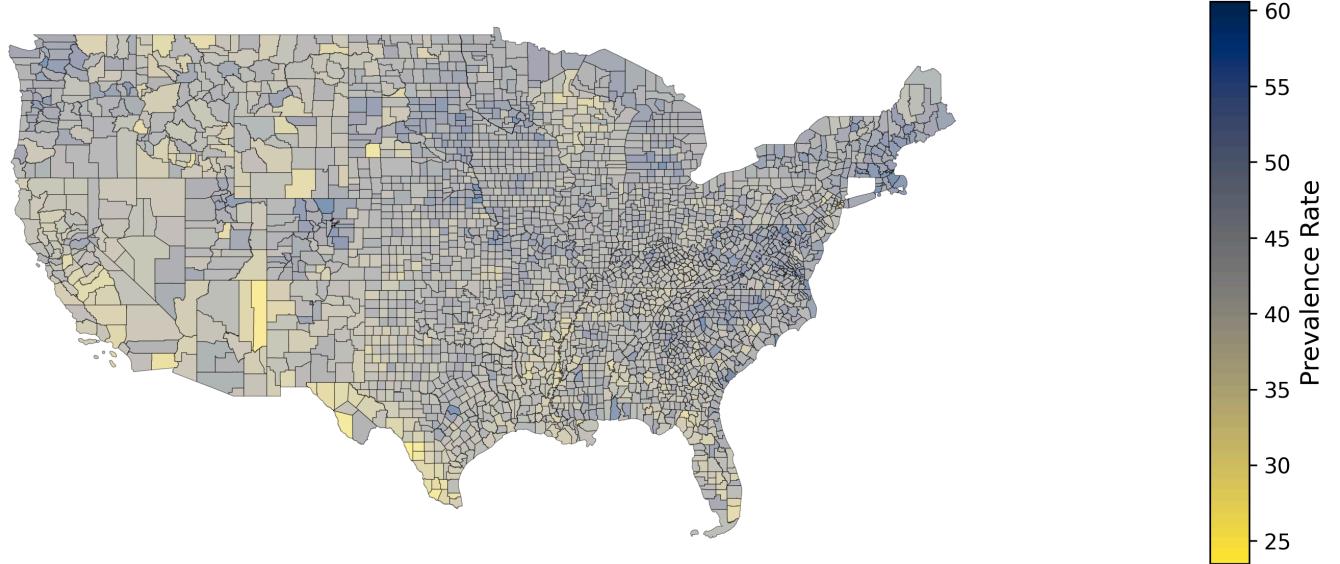
COLON\_SCREEN\_AdjPrev - US Map by County



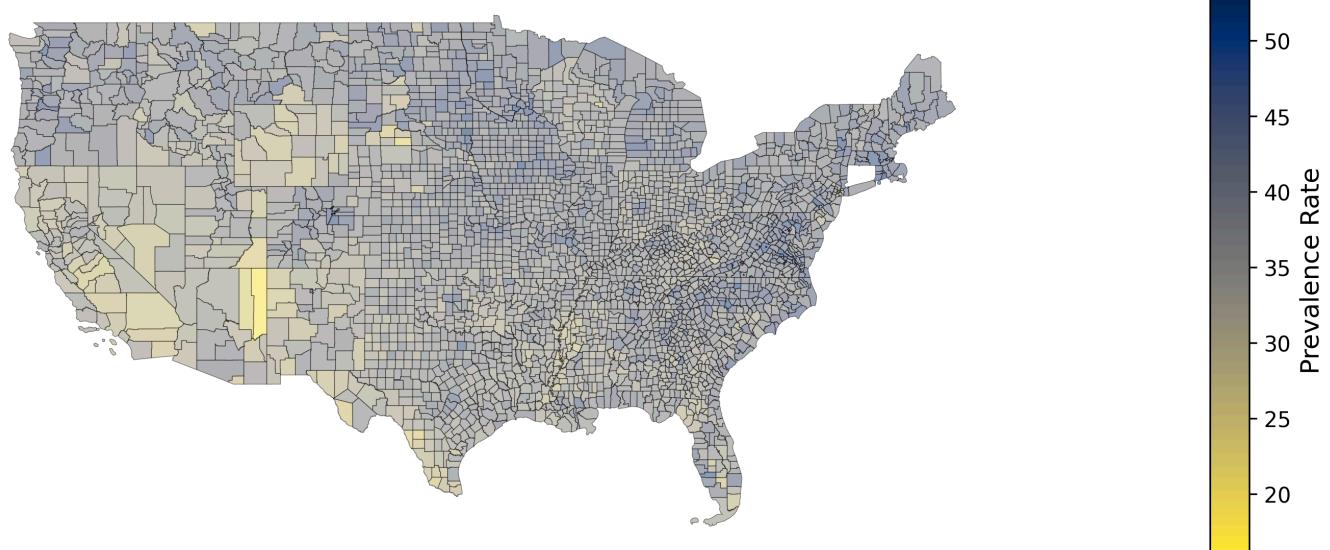
COPD\_AdjPrev - US Map by County



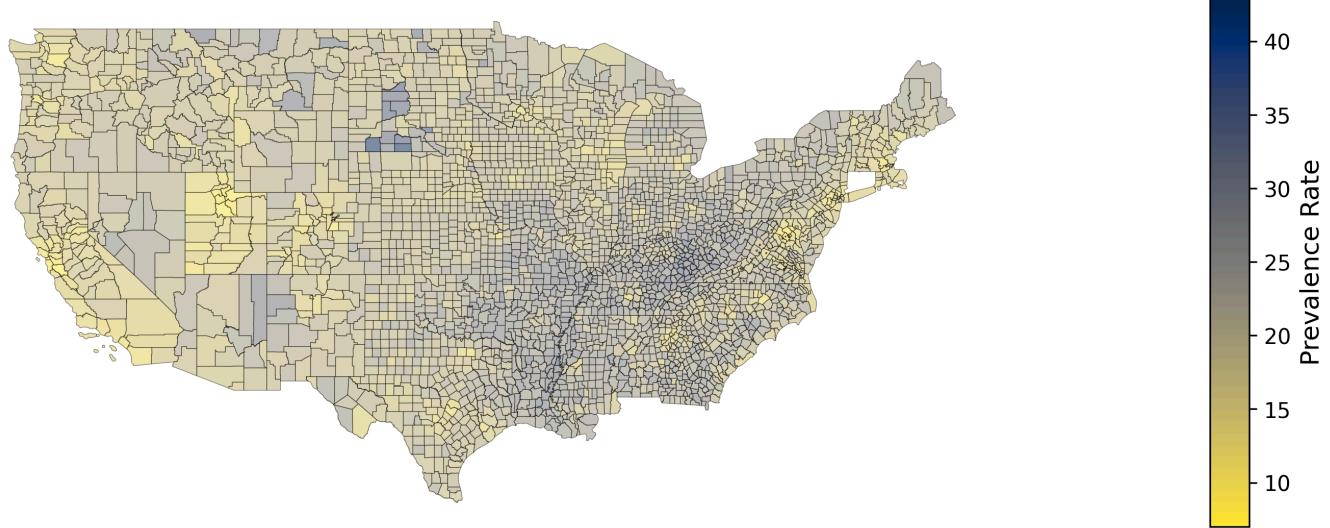
COREM\_AdjPrev - US Map by County



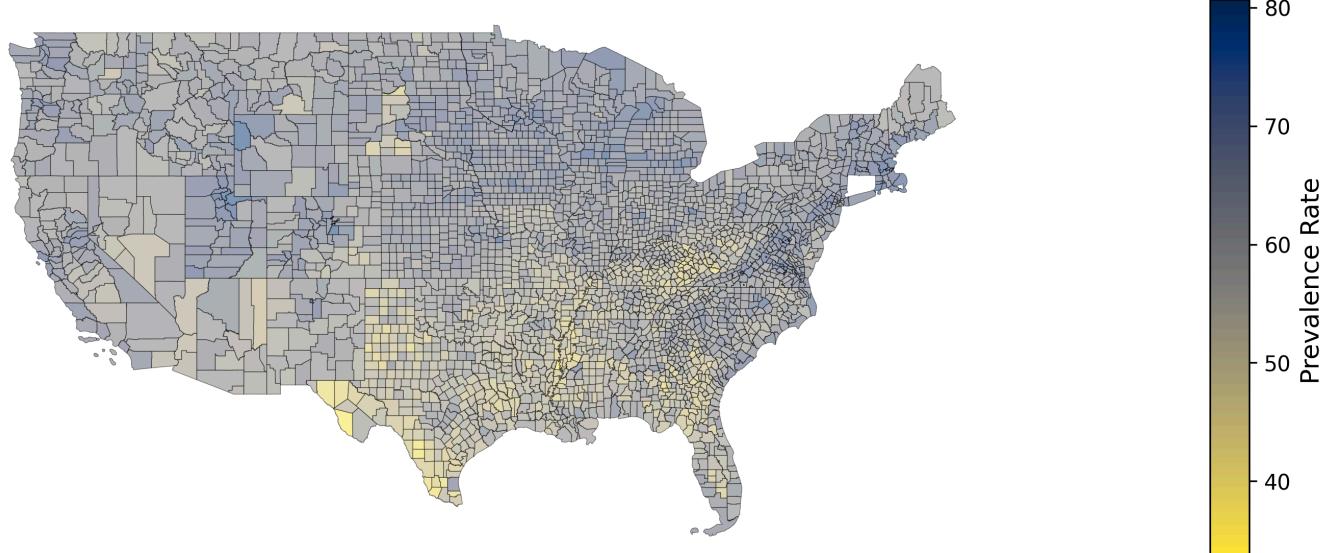
COREW\_AdjPrev - US Map by County



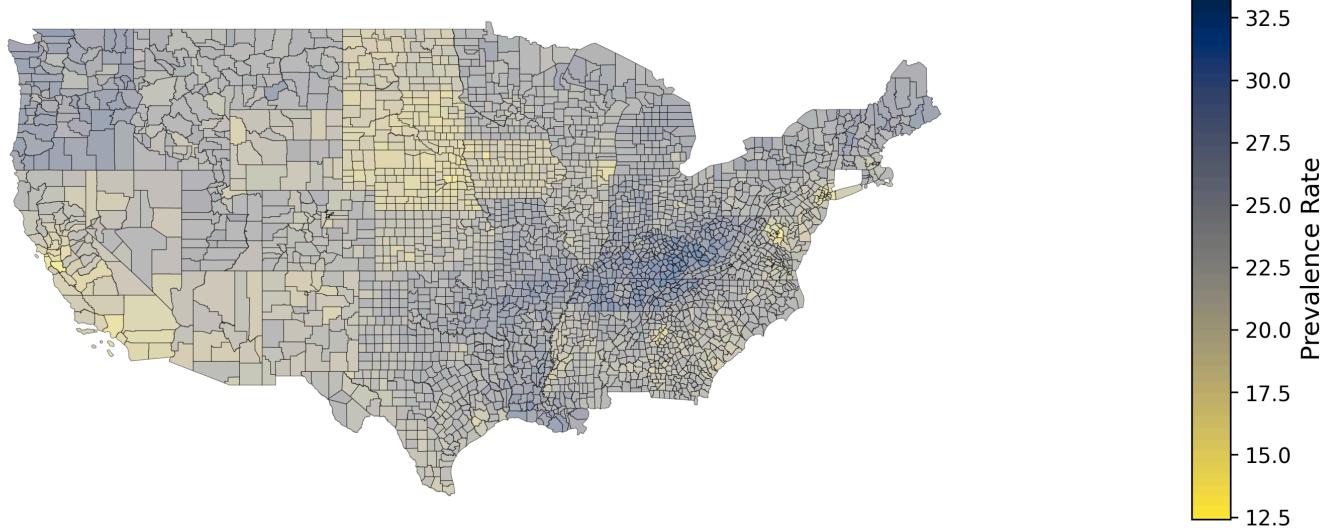
CSMOKING\_AdjPrev - US Map by County



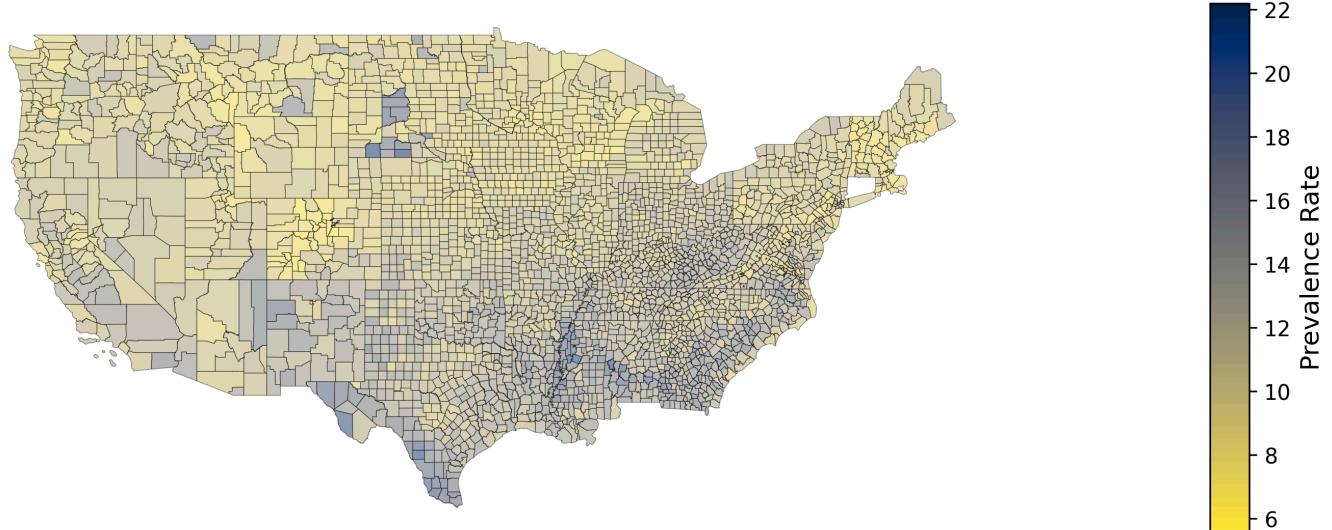
DENTAL\_AdjPrev - US Map by County



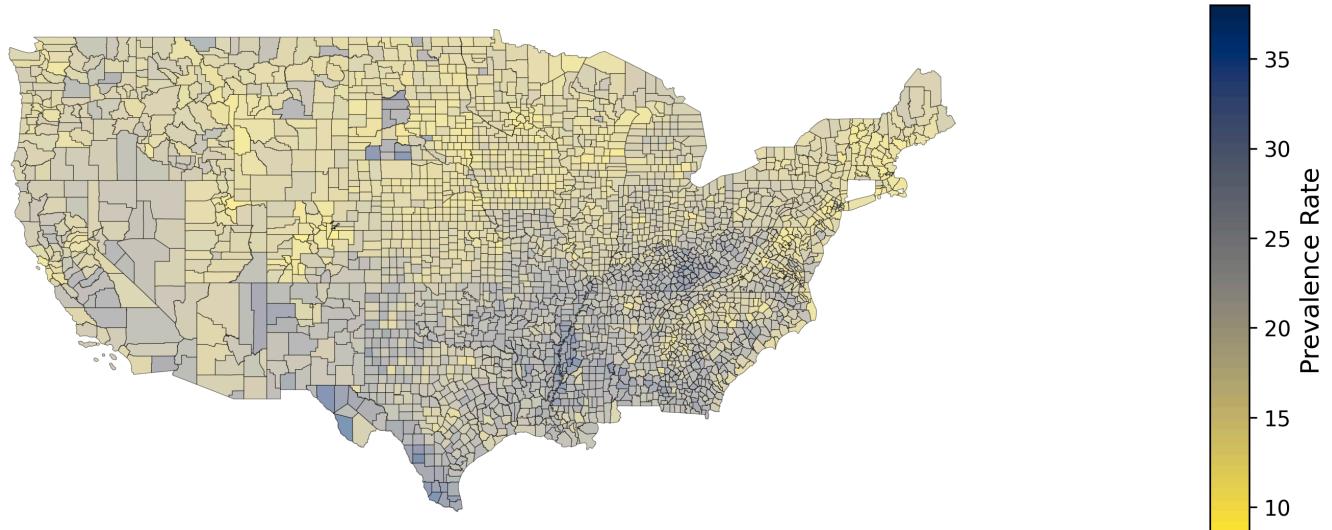
DEPRESSION\_AdjPrev - US Map by County



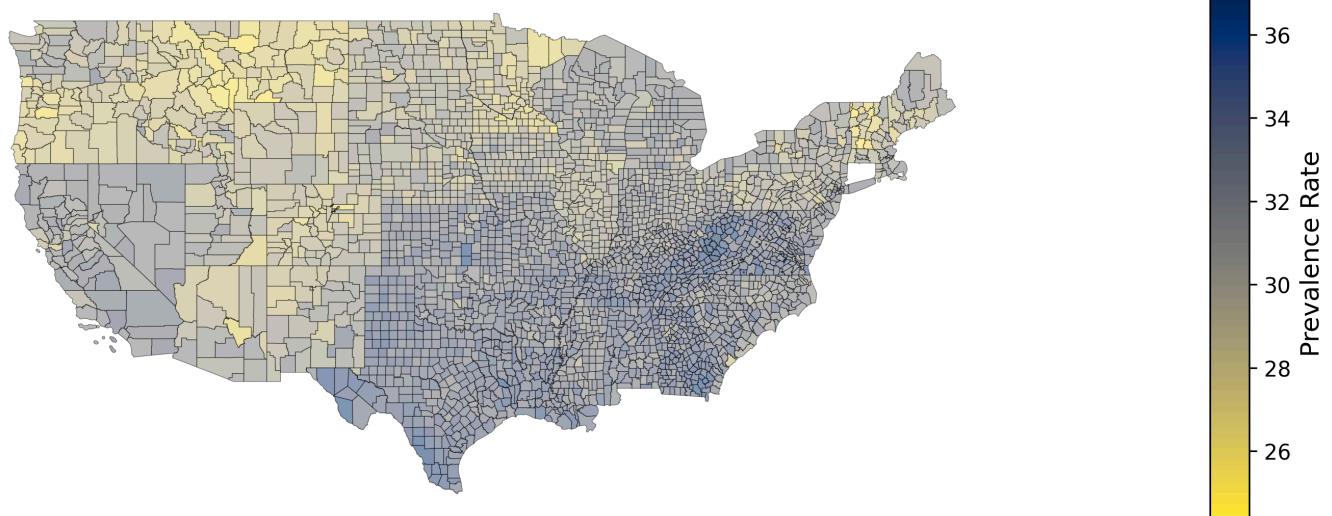
DIABETES\_AdjPrev - US Map by County



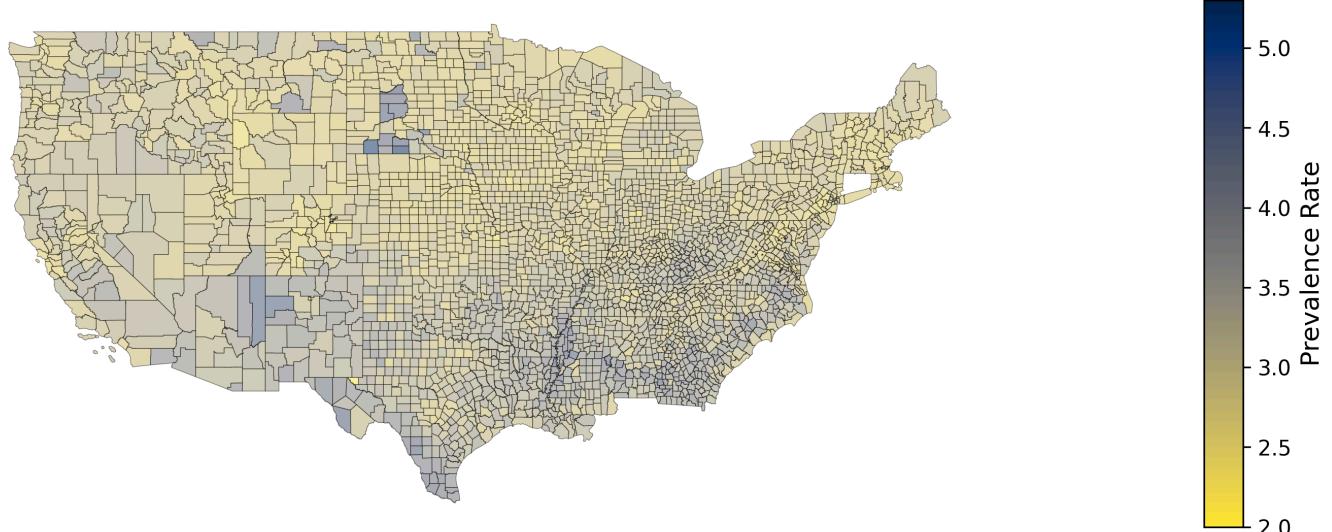
GHLTH\_AdjPrev - US Map by County



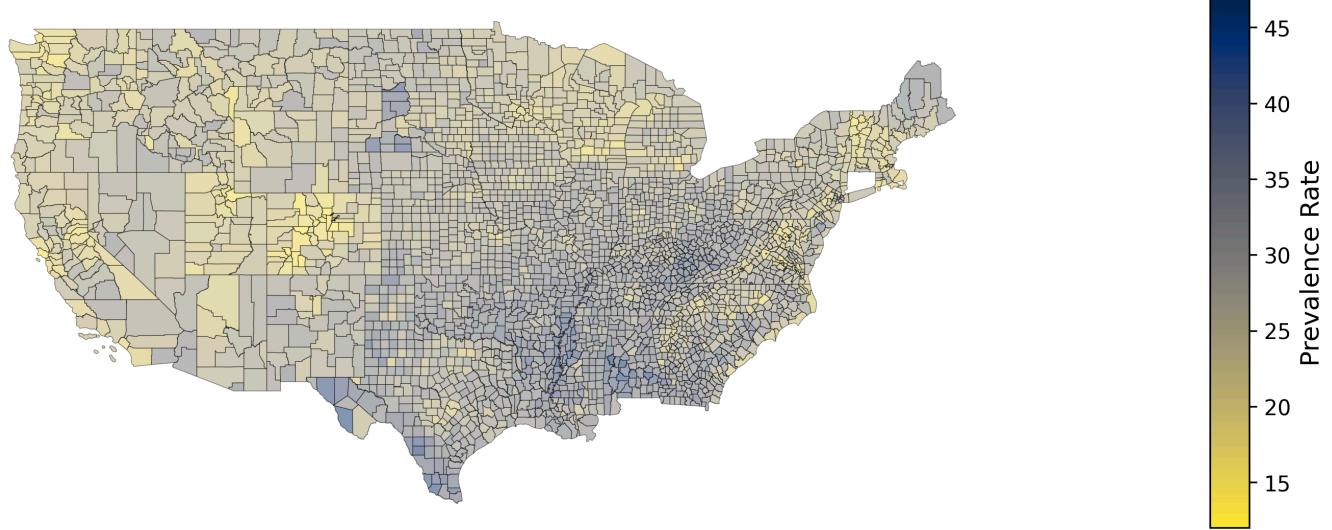
HIGHCHOL\_AdjPrev - US Map by County



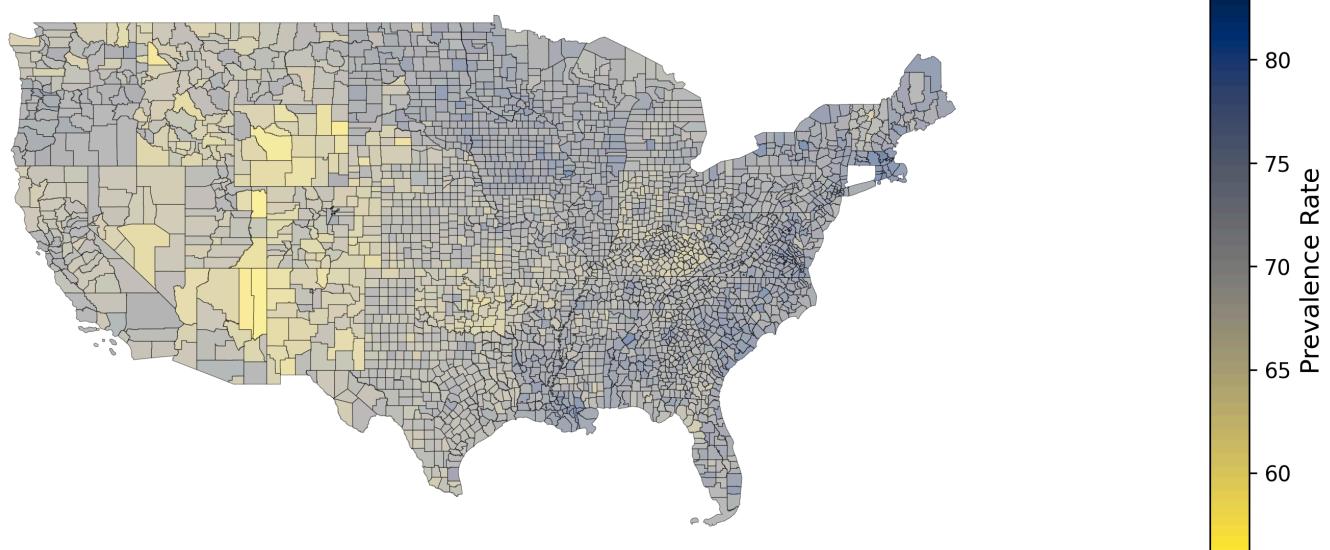
KIDNEY\_AdjPrev - US Map by County



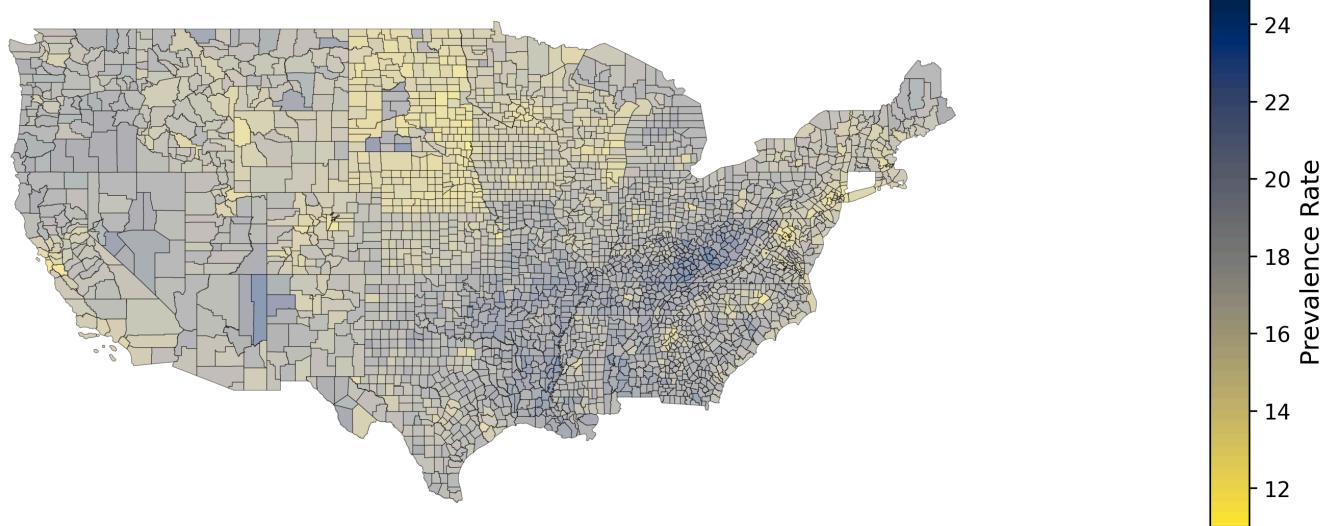
LPA\_AdjPrev - US Map by County



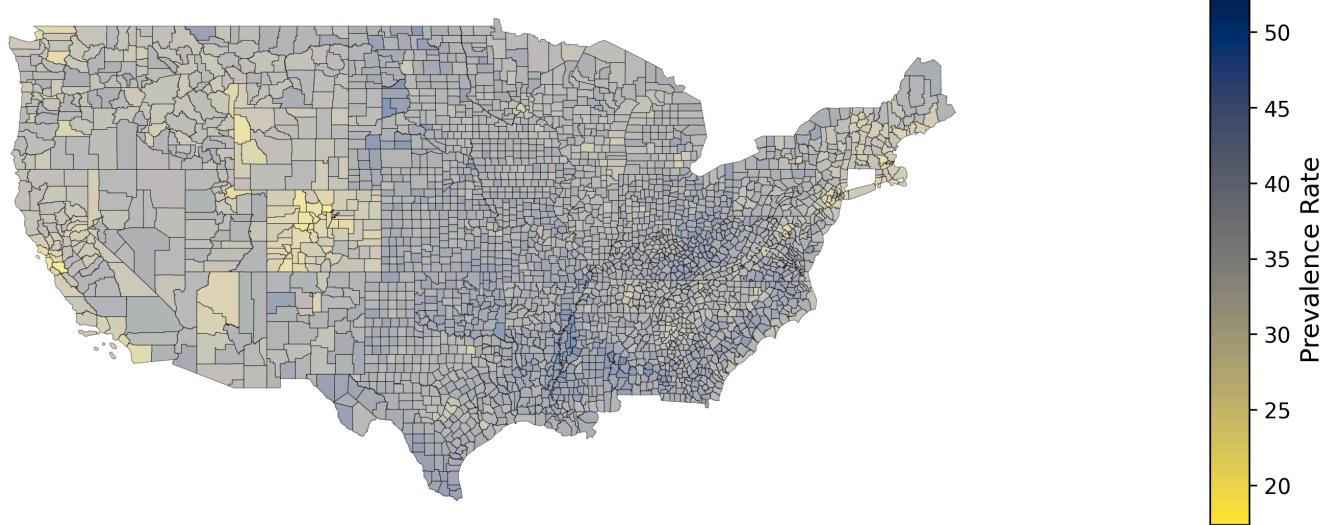
MAMMOUSE\_AdjPrev - US Map by County



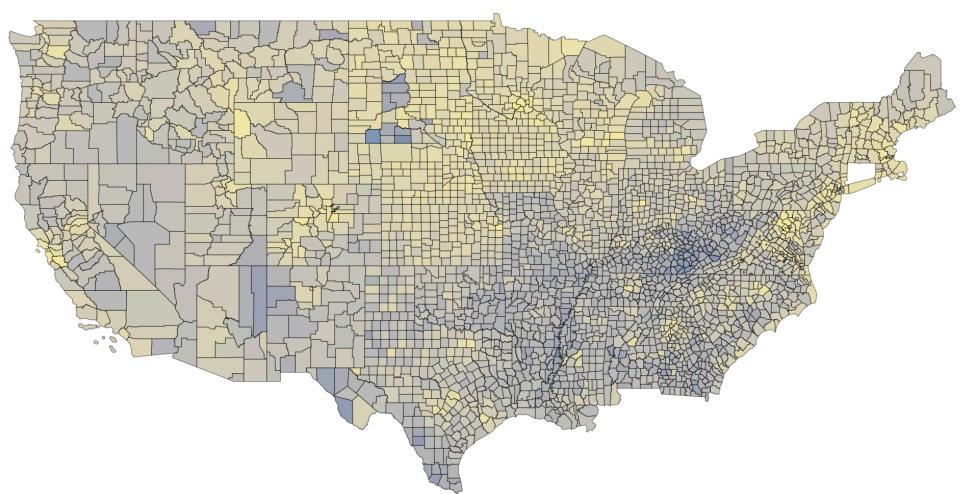
MHLTH\_AdjPrev - US Map by County



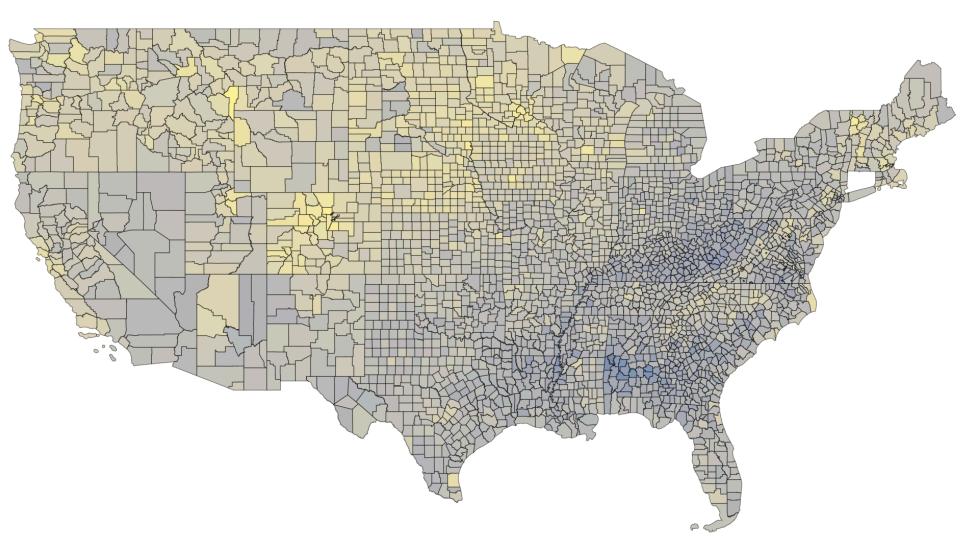
OBESITY\_AdjPrev - US Map by County



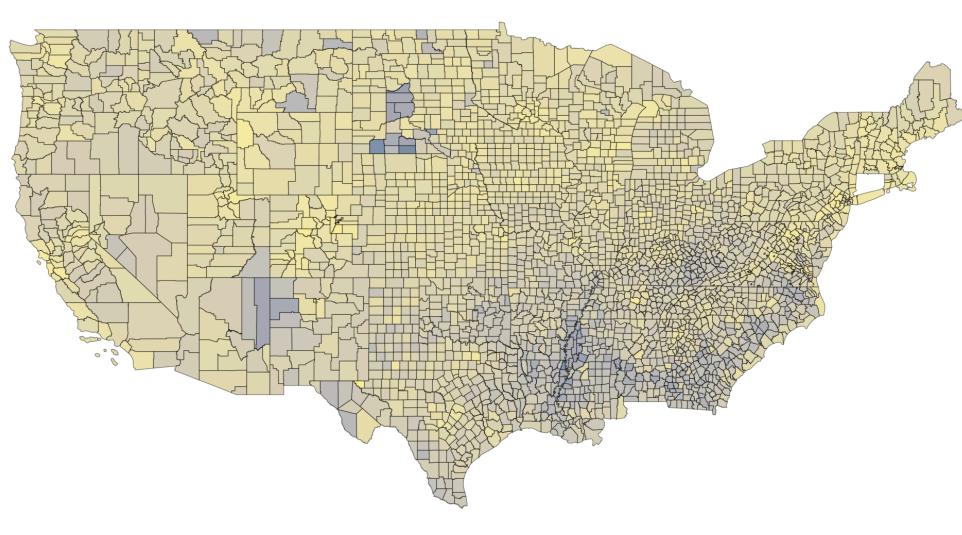
PHLTH\_AdjPrev - US Map by County



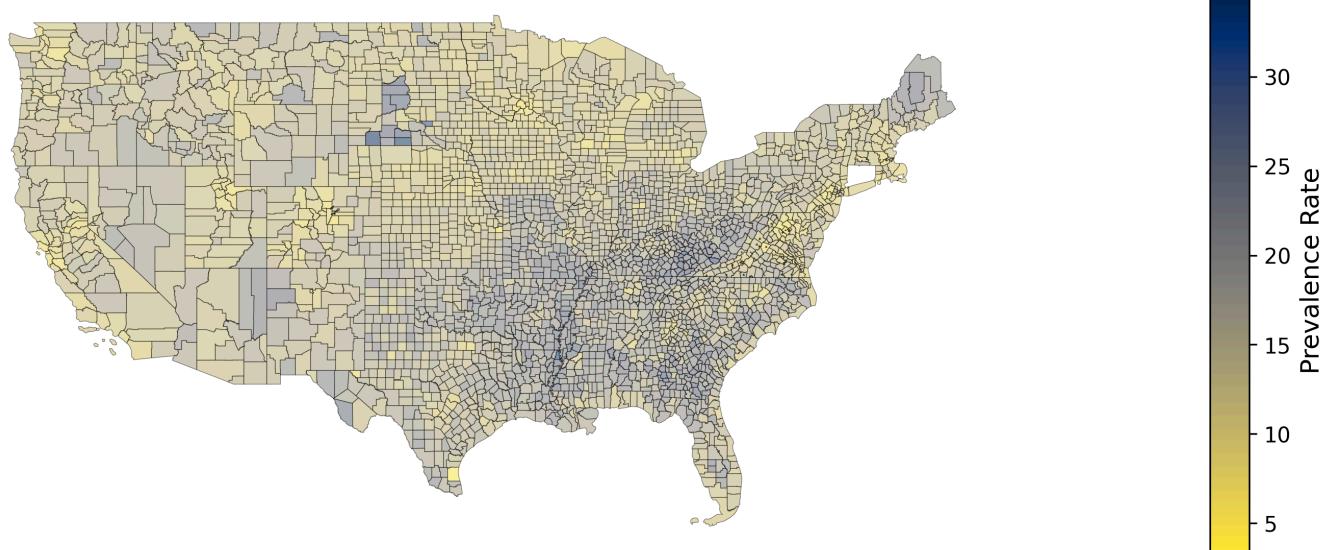
SLEEP\_AdjPrev - US Map by County



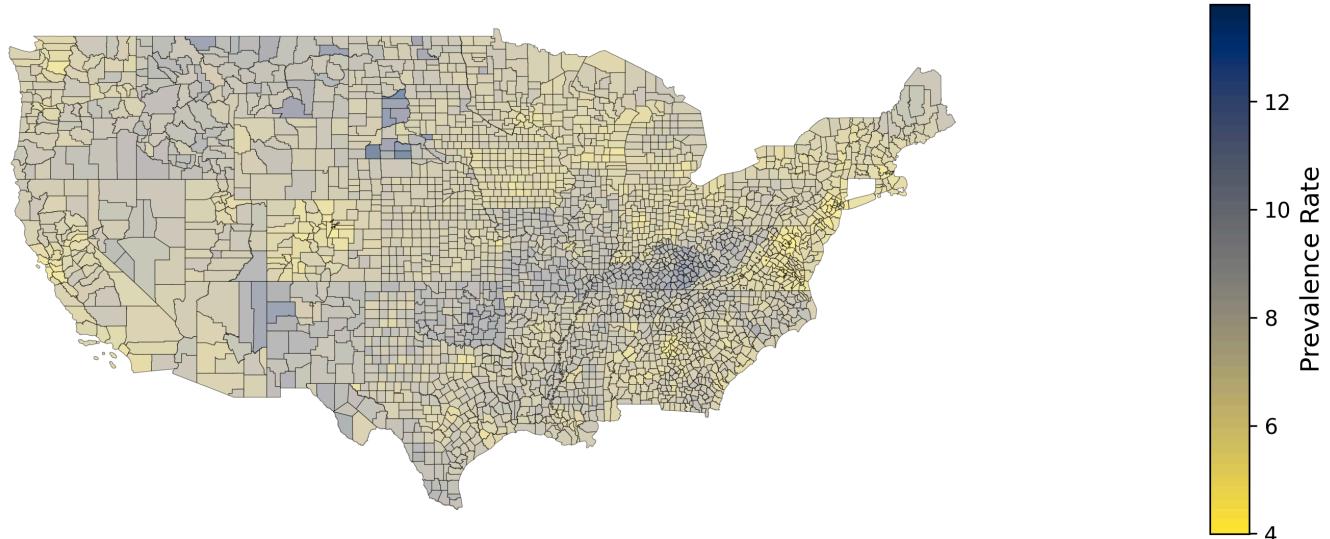
STROKE\_AdjPrev - US Map by County



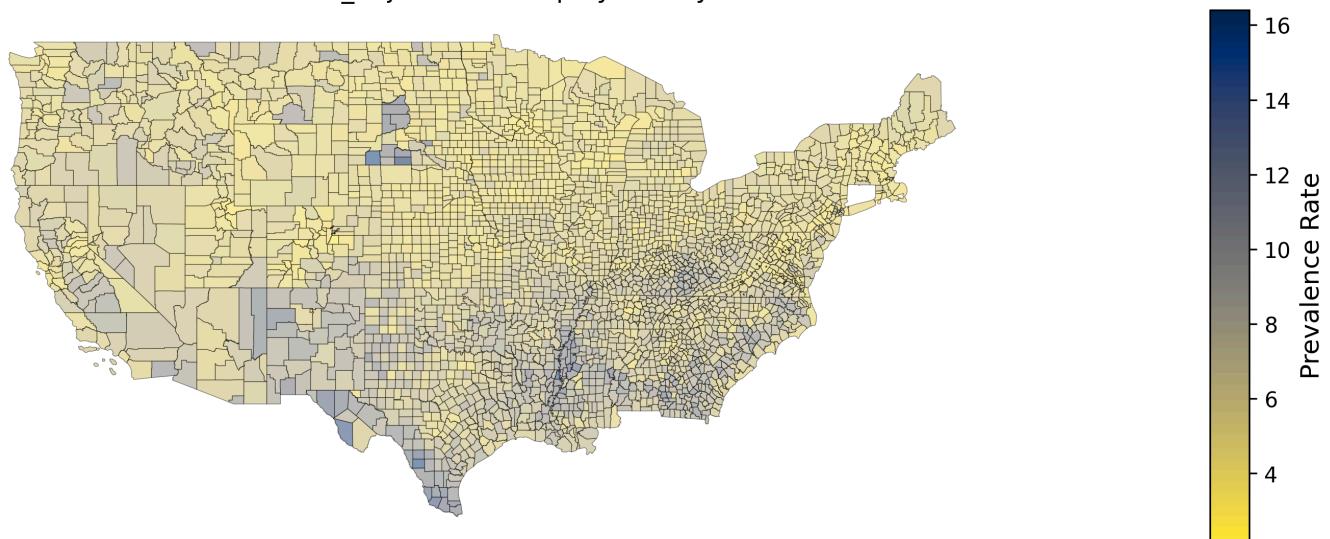
TEETHLOST\_AdjPrev - US Map by County



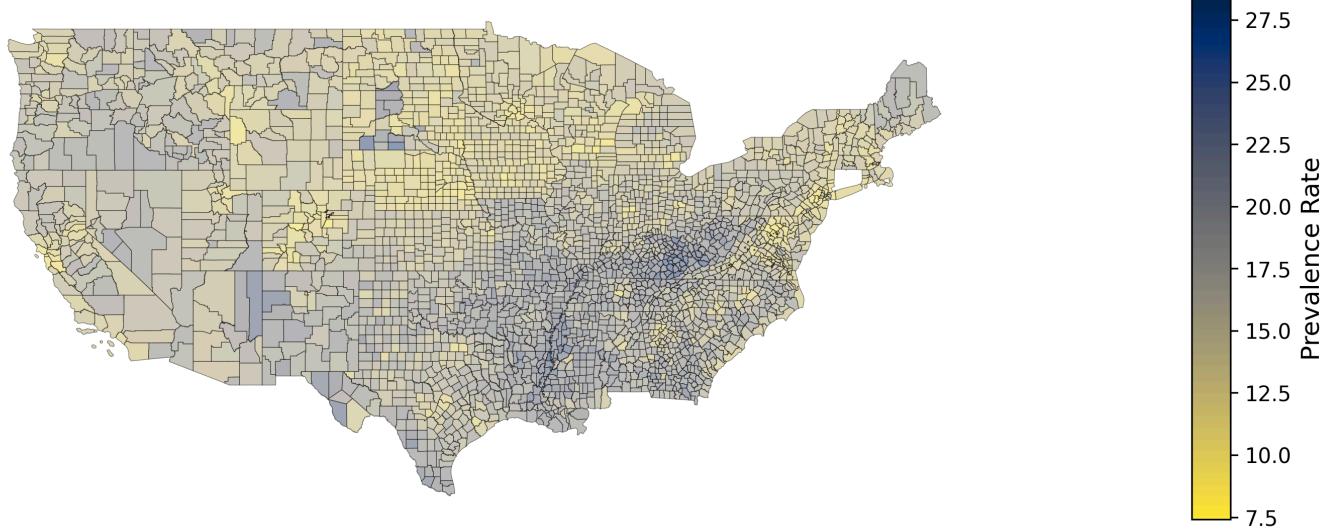
HEARING\_AdjPrev - US Map by County



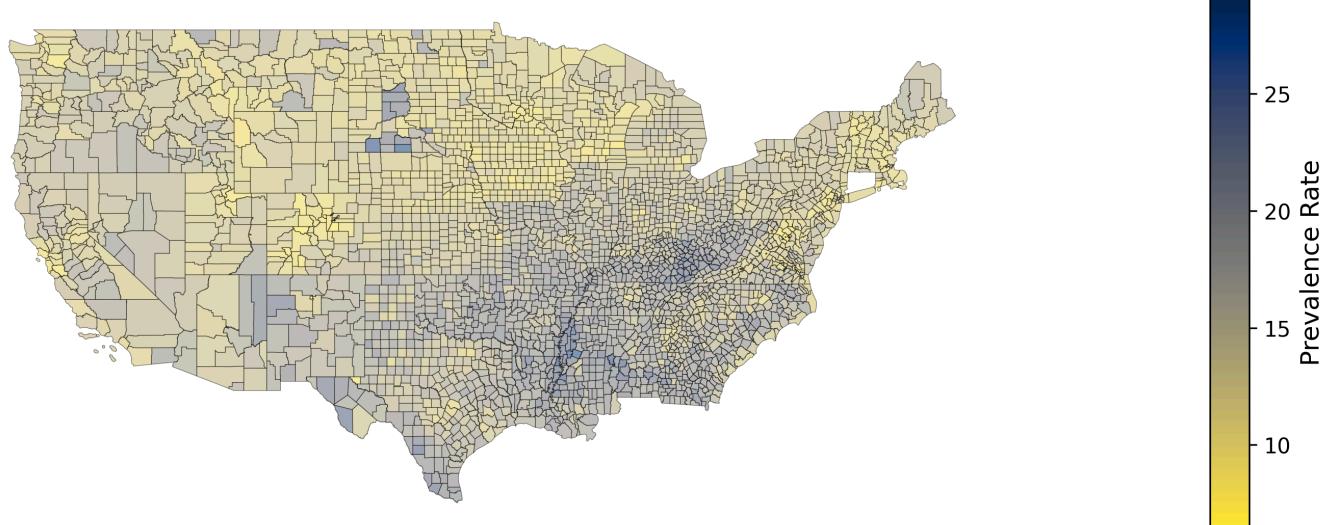
VISION\_AdjPrev - US Map by County



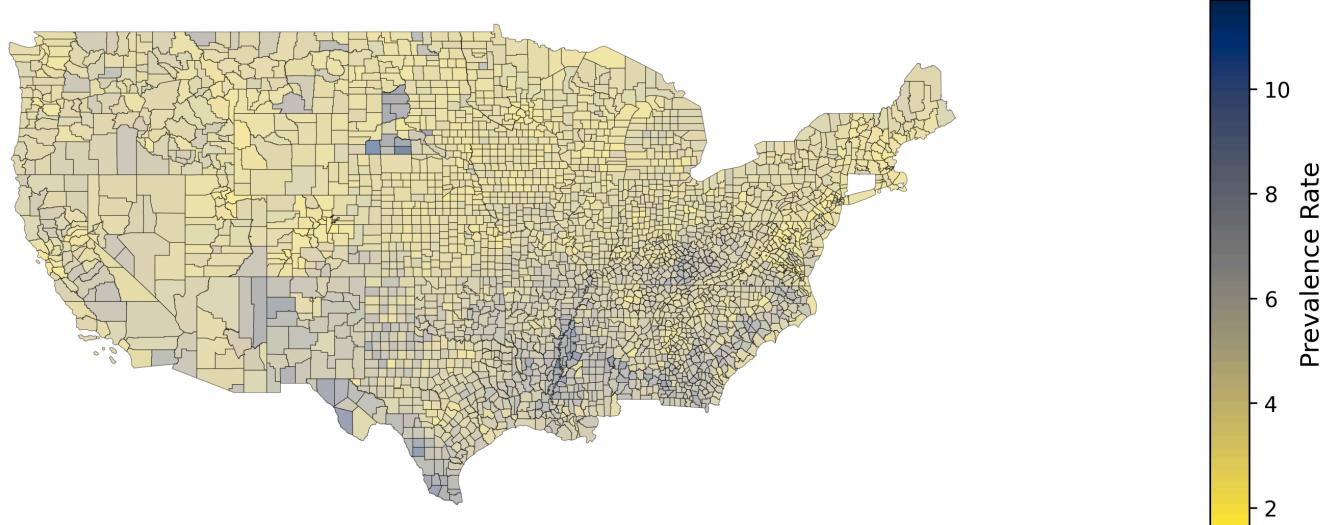
COGNITION\_AdjPrev - US Map by County



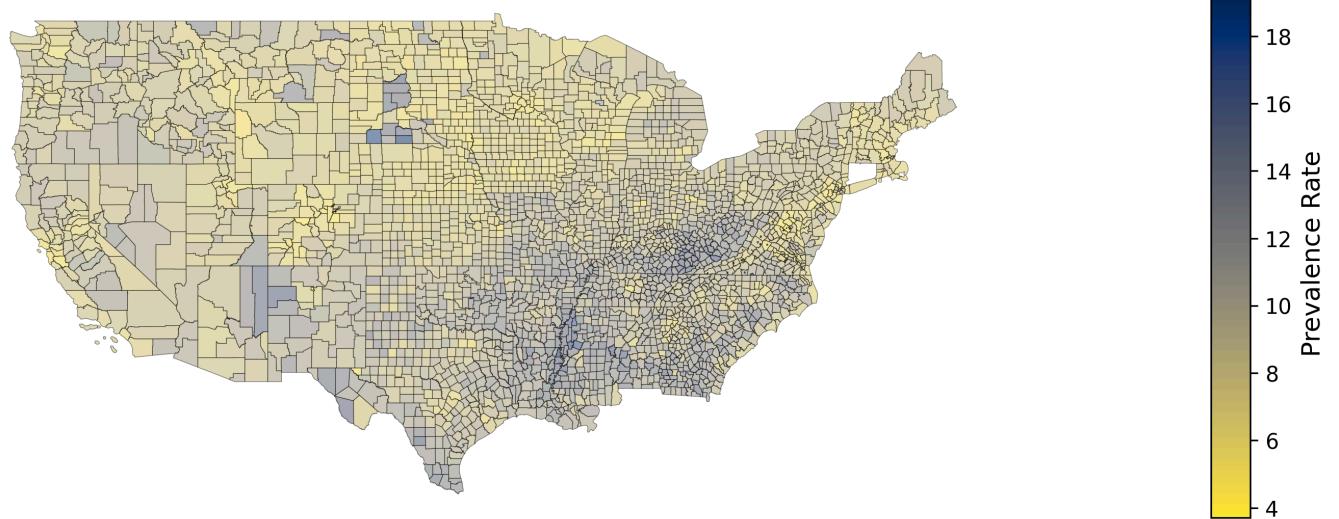
MOBILITY\_AdjPrev - US Map by County



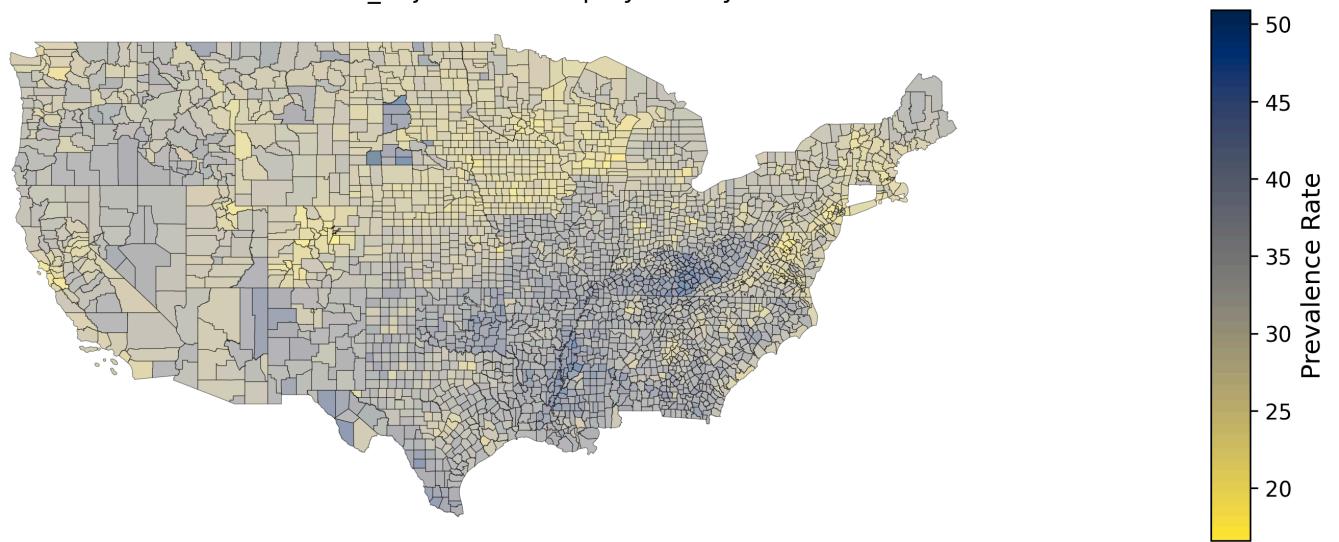
SELCARE\_AdjPrev - US Map by County



INDEPLIVE\_AdjPrev - US Map by County



DISABILITY\_AdjPrev - US Map by County



```
In [ ]: merged_gdf.to_csv("merged_data.csv", index=False)
```