

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

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

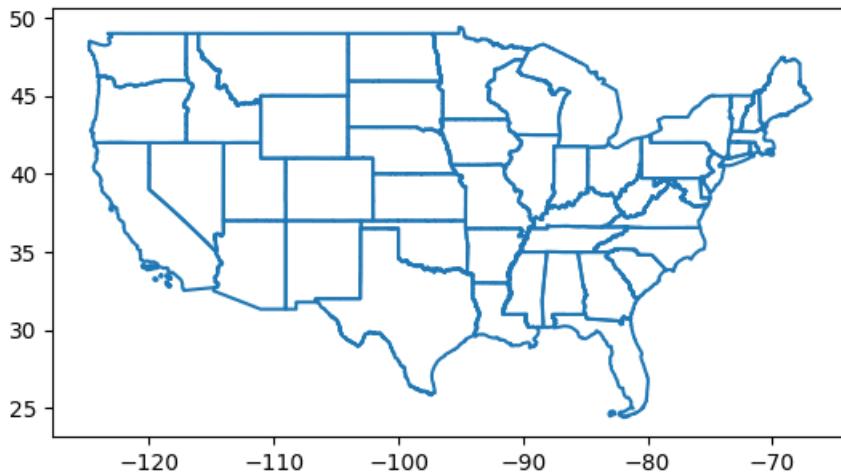
path = "C:/Users/user/Desktop/GIS/Final Project/tl_2023_us_state/tl_2023_us_state.shp"
usa = gpd.read_file(path)
usa = usa.to_crs("EPSG:4326")
```

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

```
Out[ ]: REGION                               3
DIVISION                             5
STATEFP                                54
STATENS      01779805
GEOID                                     54
GEOIDFQ      04000000US54
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('_AdjPrev')] + ['TotalPopulation']  
  
# 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_AdjPrev ARTHRITIS_AdjPrev BINGE_AdjPrev BPHIGH_AdjPrev BPMED_AdjPrev CANCER_AdjPrev CASTHMA_AdjPrev C  
0       AK      12.900000     22.083333    18.556667    30.243333    51.640000    6.076667    10.126667  
1       AL      13.220896     28.986567    14.617910    39.455224    66.020896    6.153731    10.883582  
2       AR      11.934667     27.272000    14.470667    38.052000    62.222667    6.208000    10.177333  
3       AZ      15.993333     22.473333    17.306667    28.706667    54.746667    6.000000    10.833333  
4       CA      10.905172     20.593103    16.727586    27.650000    53.794828    5.841379    9.774138
```

5 rows × 39 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_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',
       'SELF CARE_AdjPrev', 'INDEPLIVE_AdjPrev', 'DISABILITY_AdjPrev',
       'TotalPopulation'],
      dtype='object')

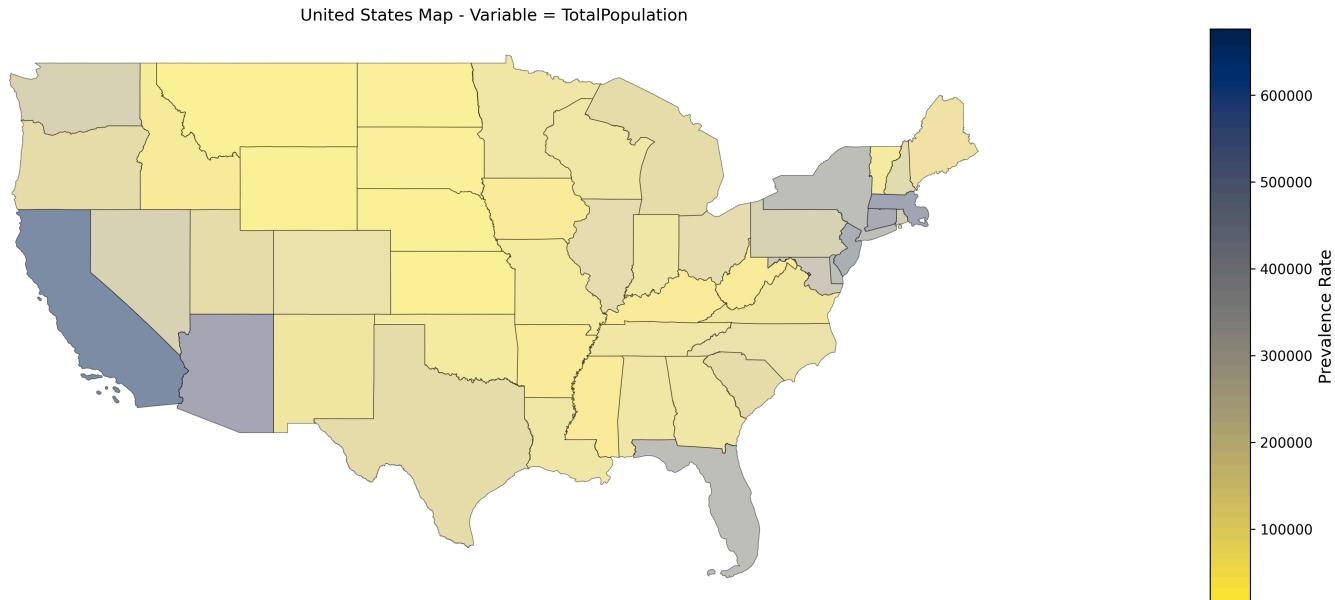
```

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')
```



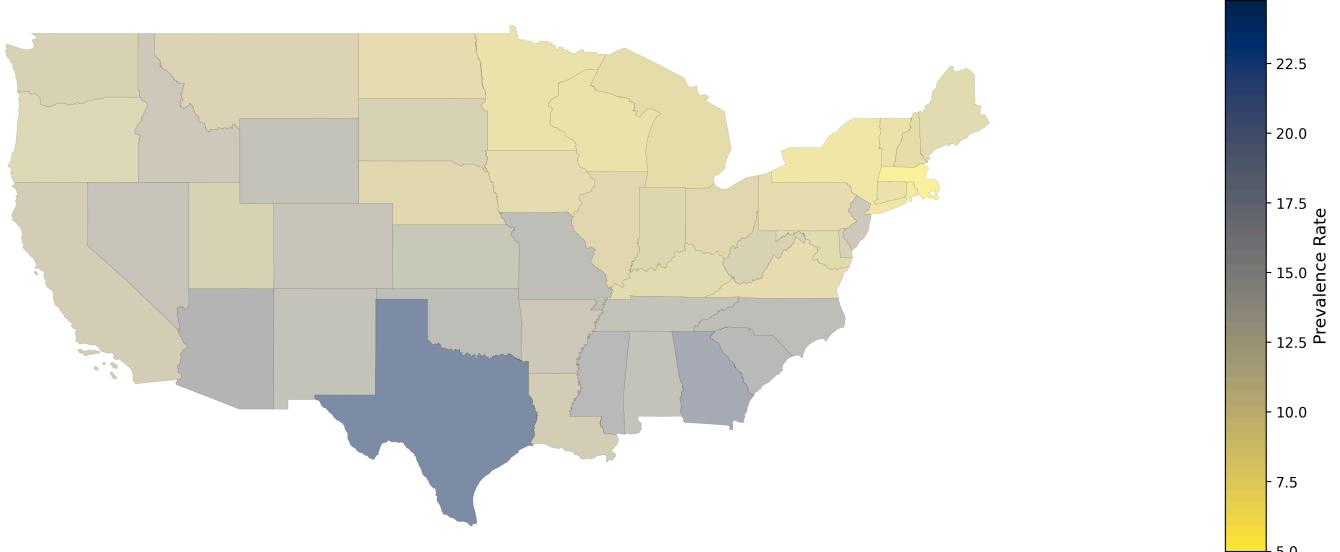
Function to Plot Multiple Columns

```
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()
```

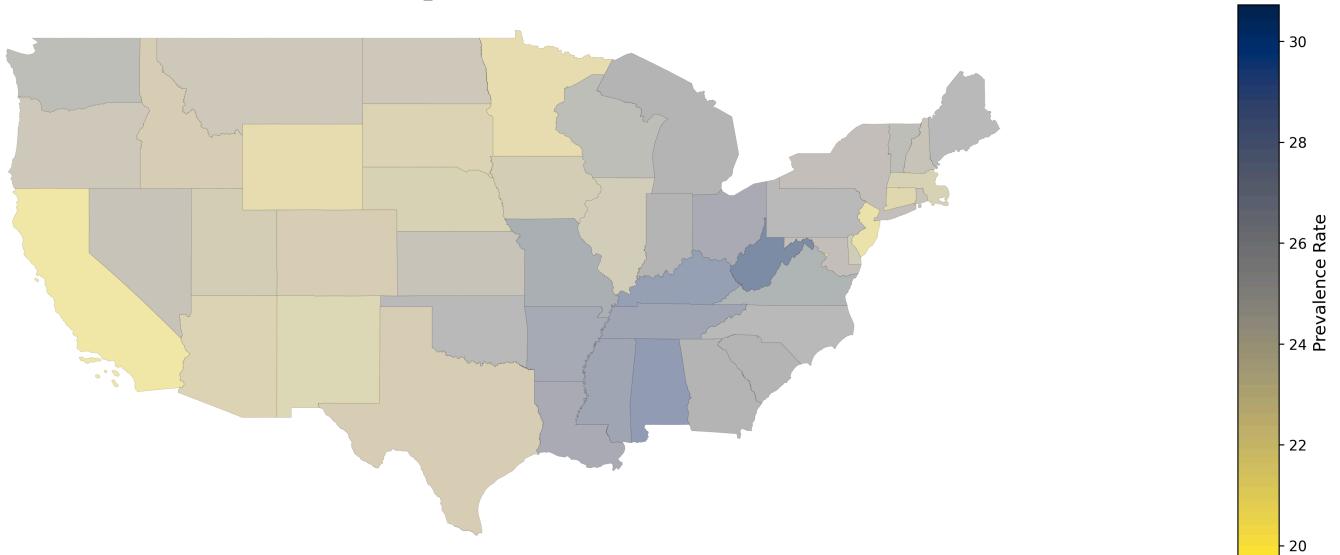
Use Age-Adjusted Data

```
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')
```

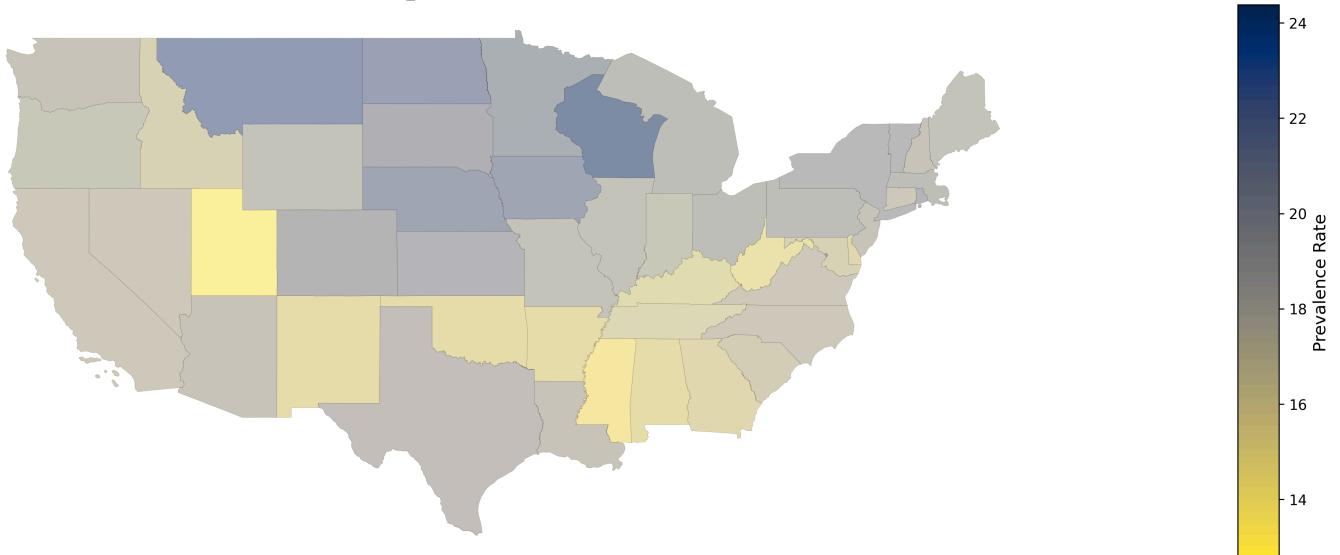
ACCESS2_AdjPrev - United States Map



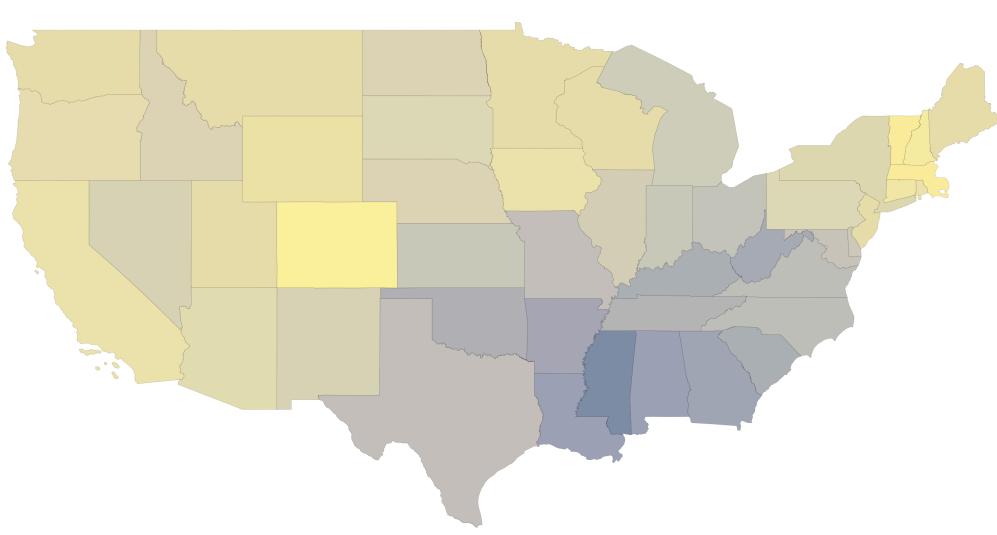
ARTHRITIS_AdjPrev - United States Map



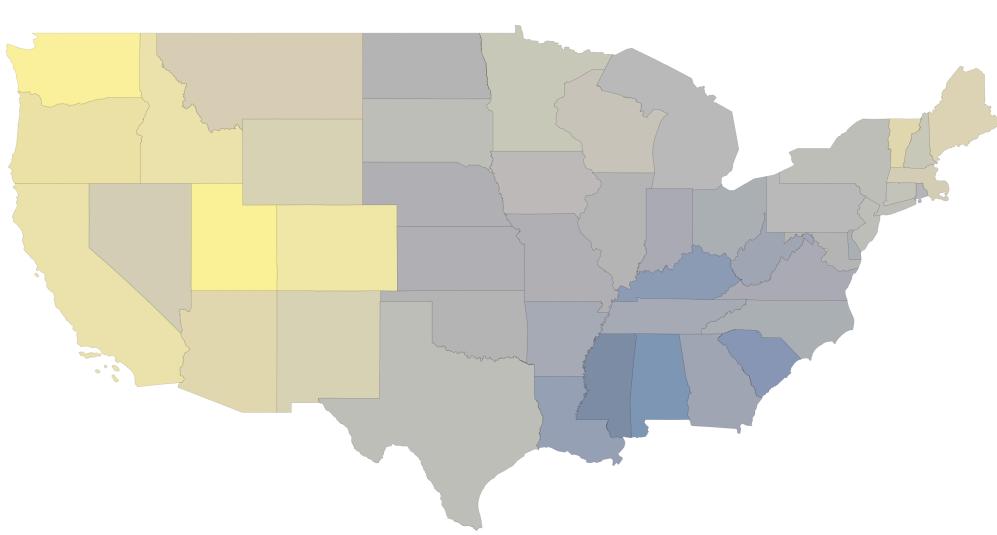
BINGE_AdjPrev - United States Map



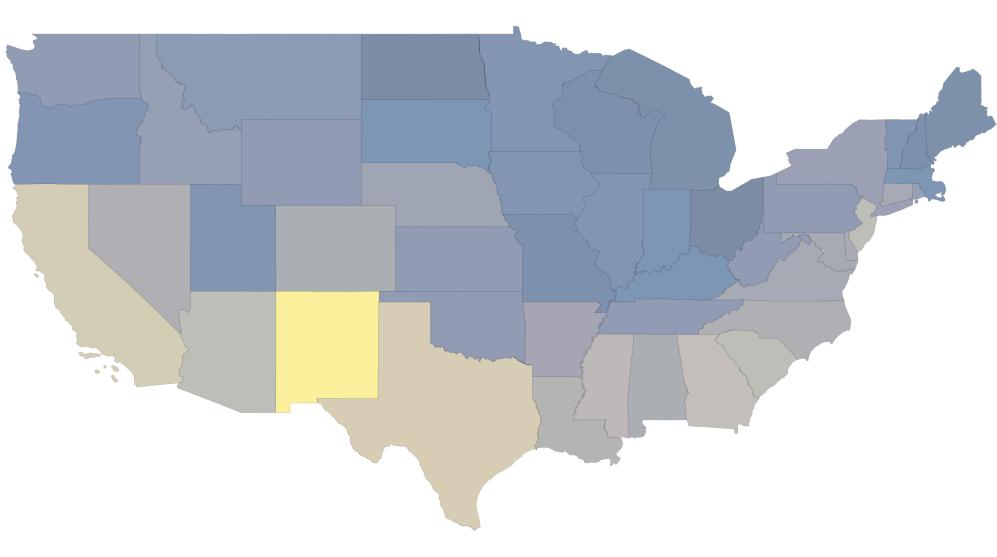
BPHIGH_AdjPrev - United States Map



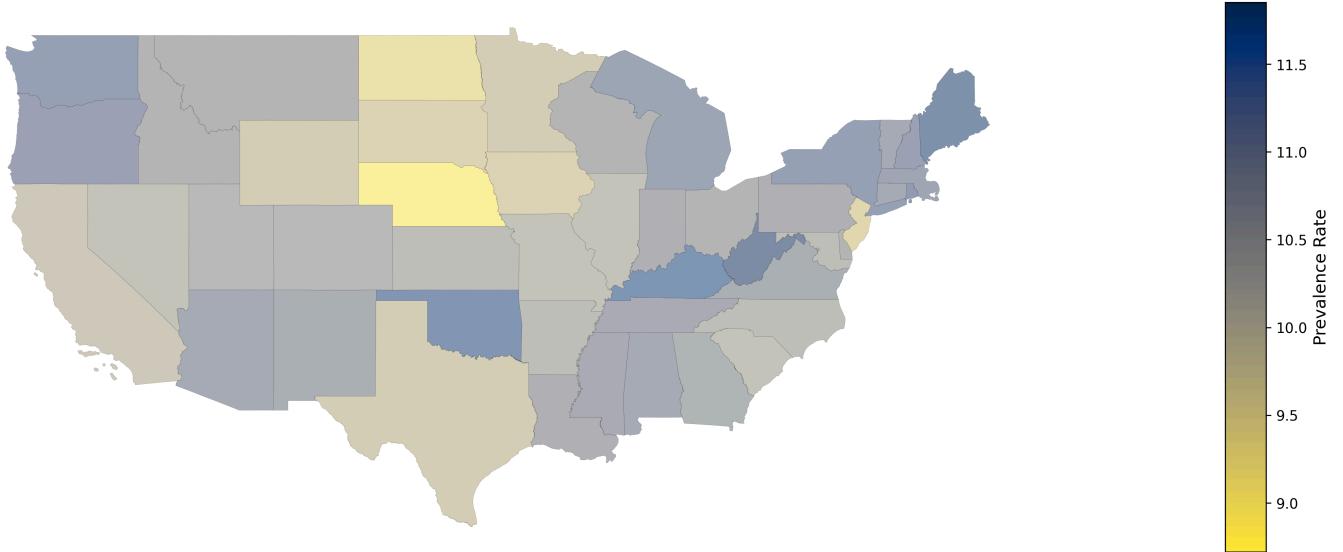
BPMED_AdjPrev - United States Map



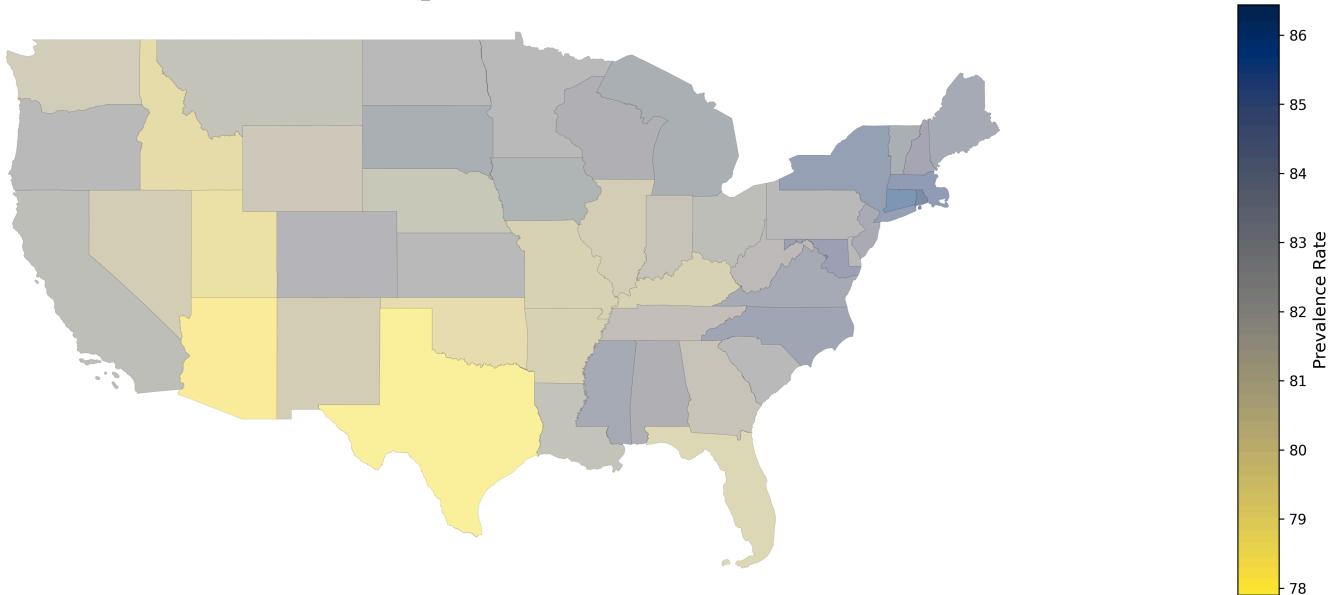
CANCER_AdjPrev - United States Map



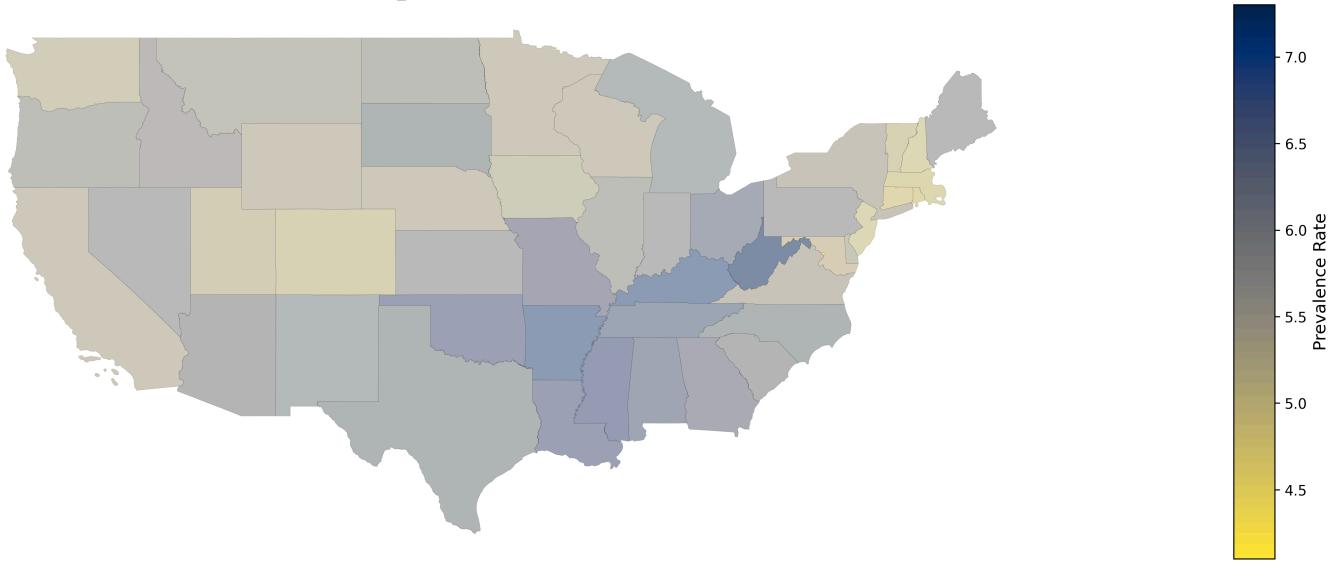
CASTHMA_AdjPrev - United States Map



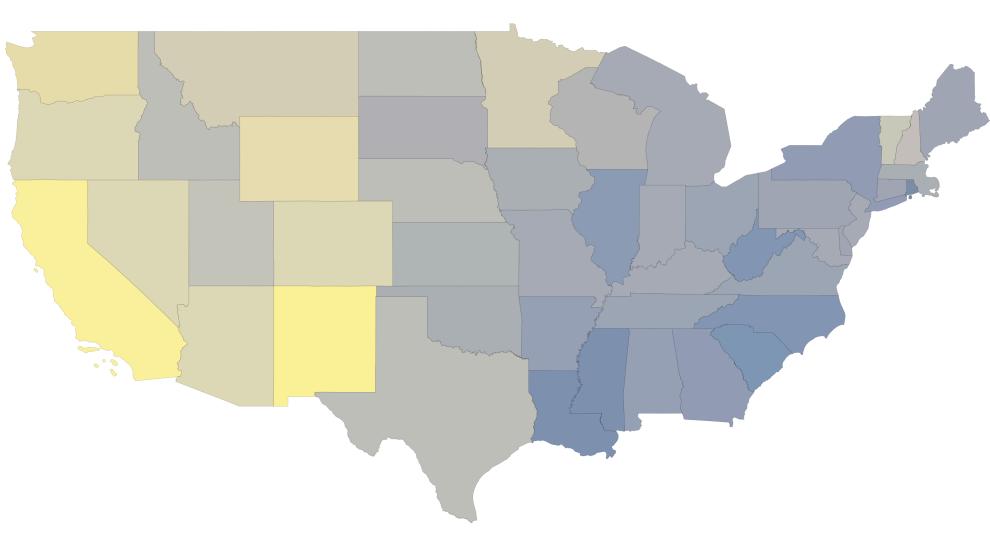
CERVICAL_AdjPrev - United States Map



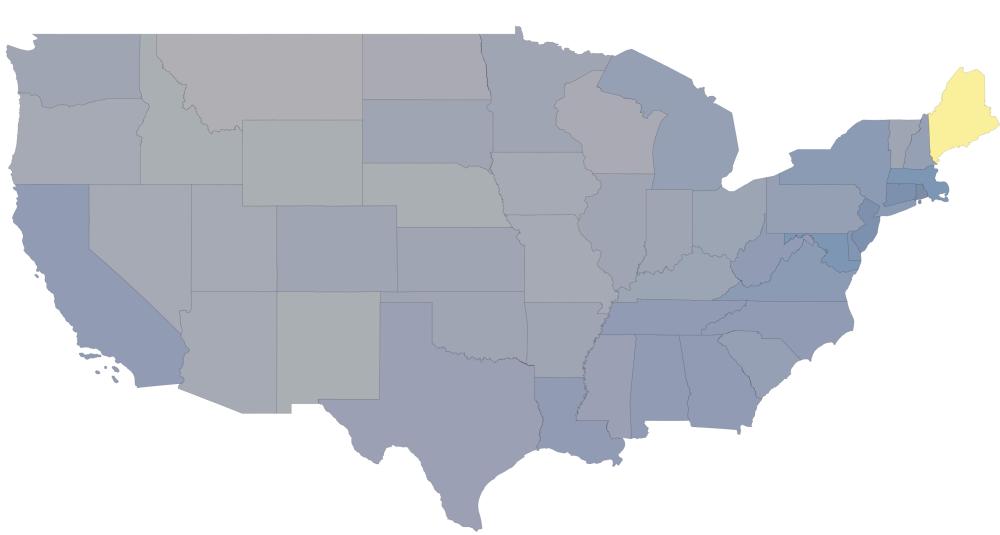
CHD_AdjPrev - United States Map



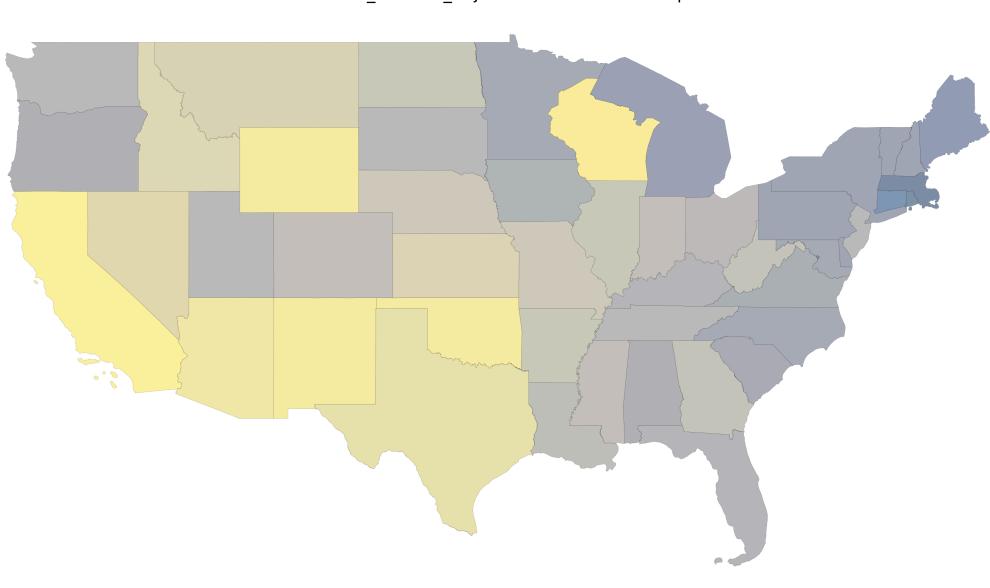
CHECKUP_AdjPrev - United States Map



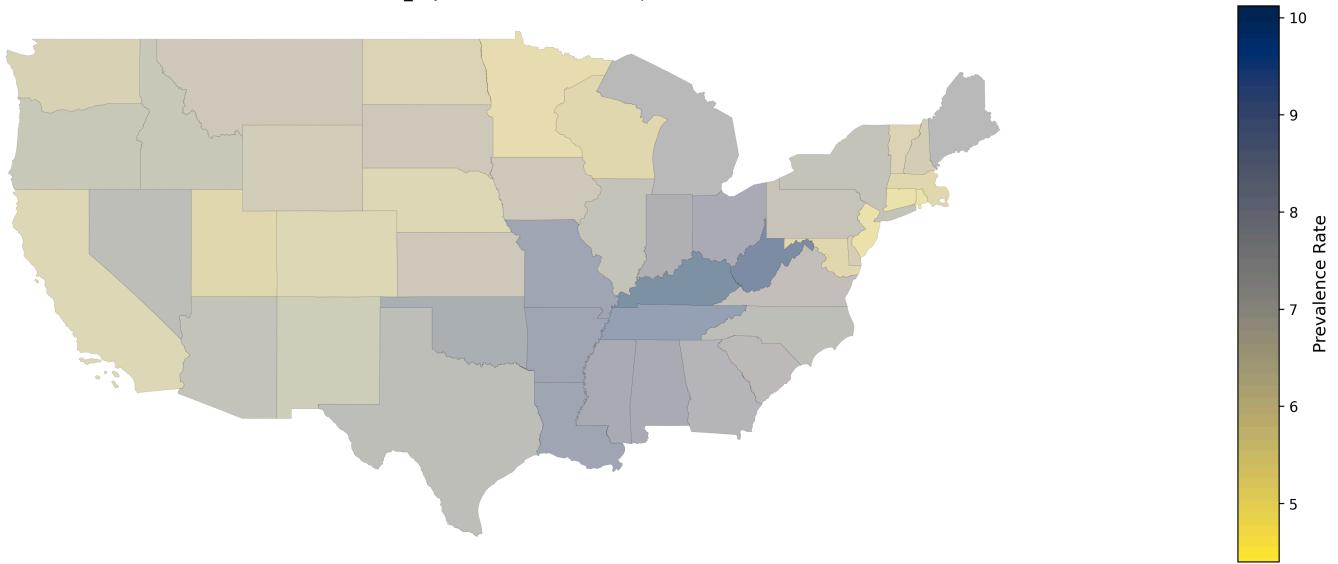
CHOLSCREEN_AdjPrev - United States Map



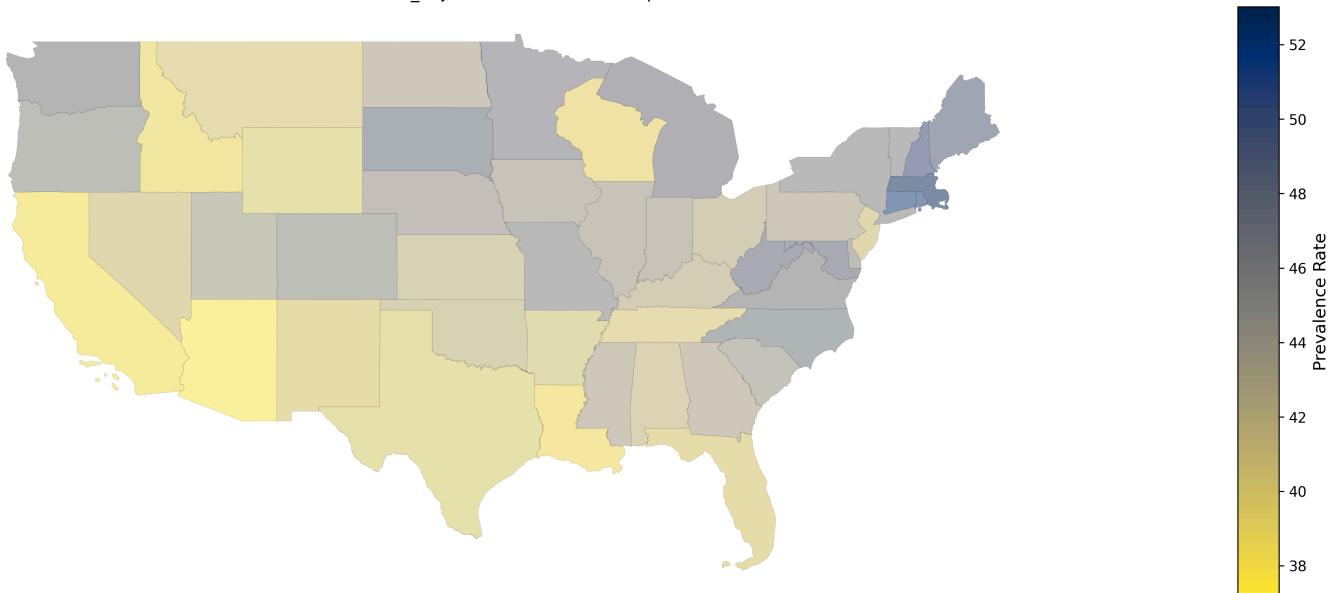
COLON_SCREEN_AdjPrev - United States Map



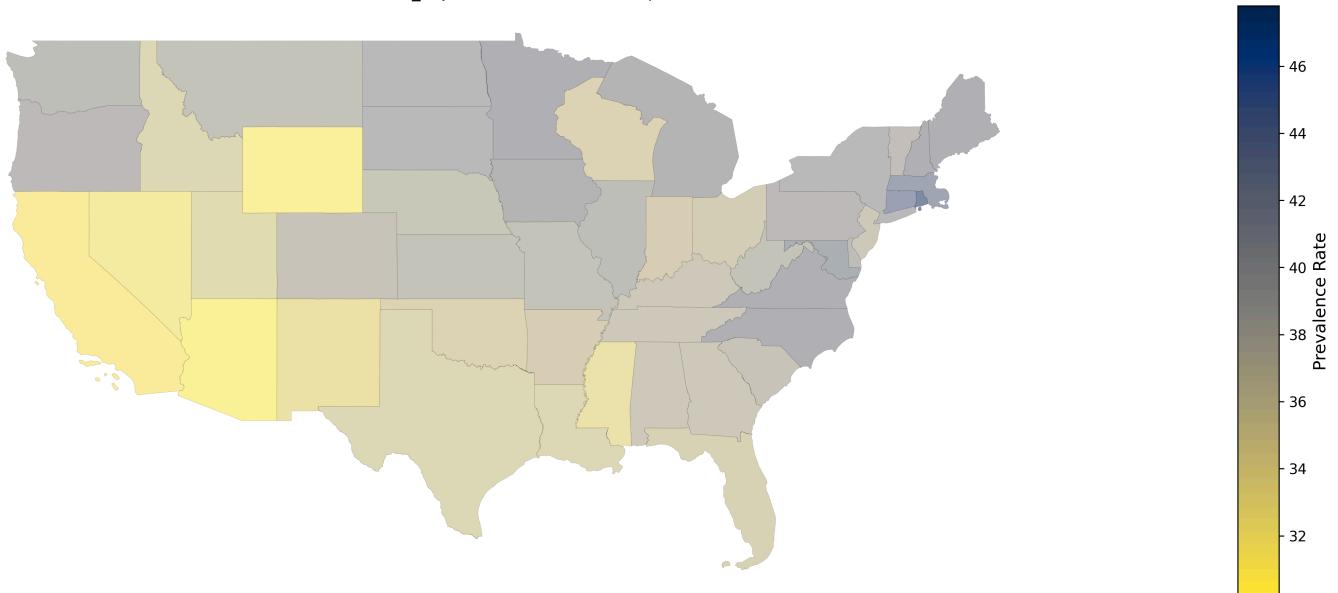
COPD_AdjPrev - United States Map



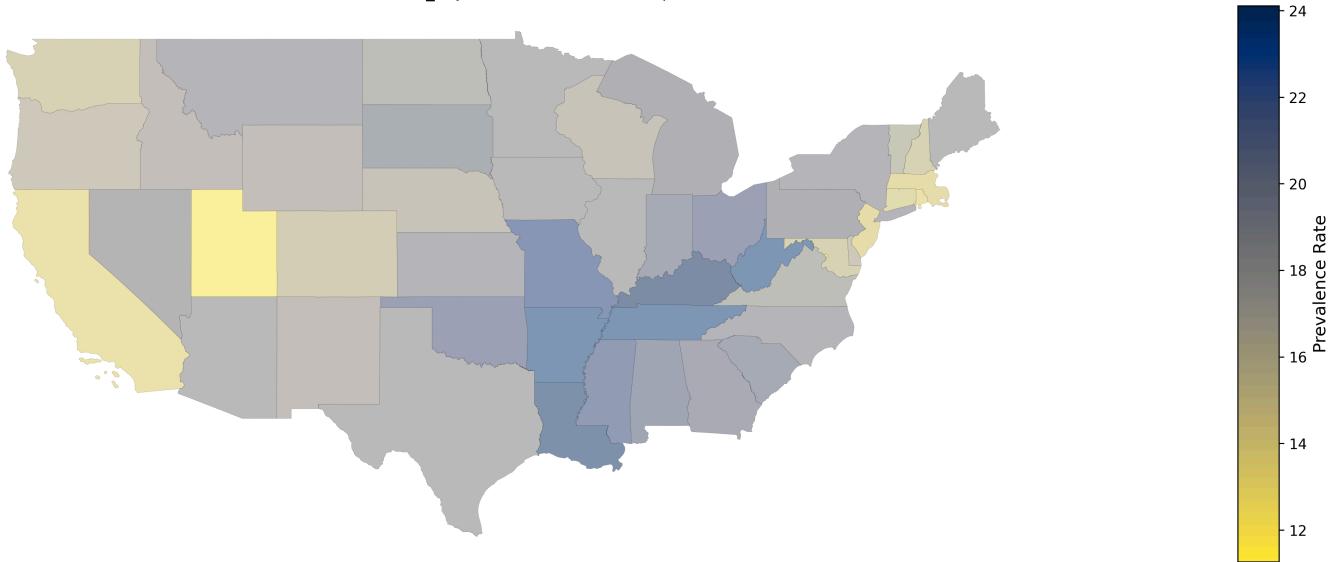
COREM_AdjPrev - United States Map



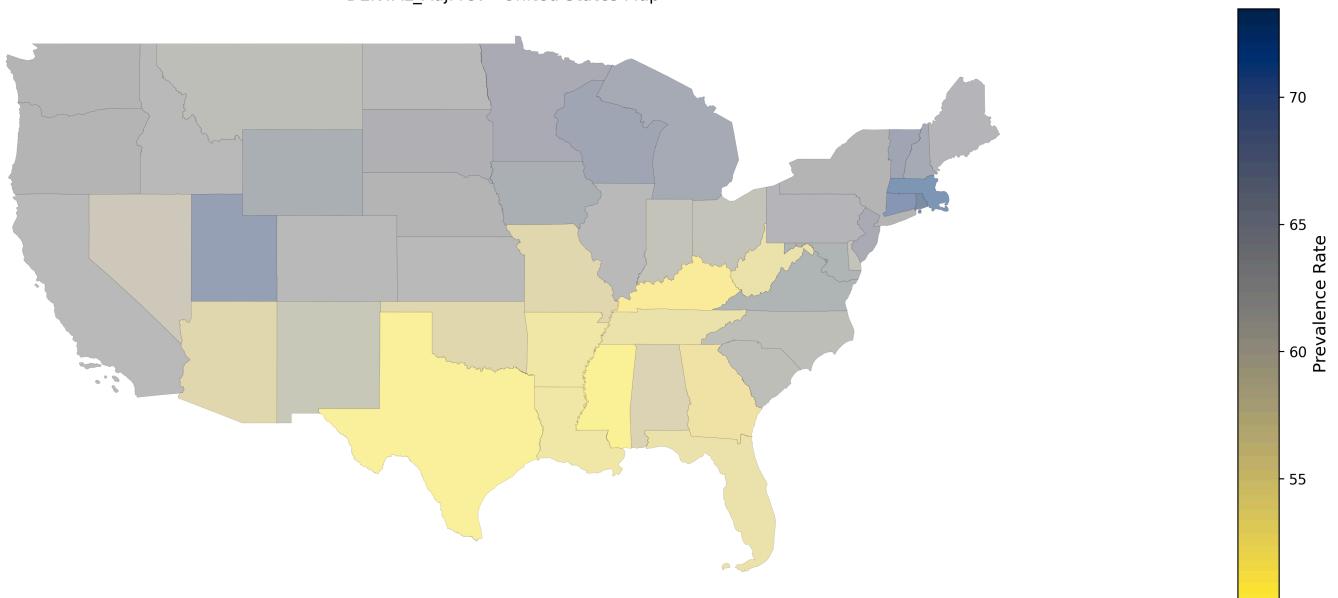
COREW_AdjPrev - United States Map



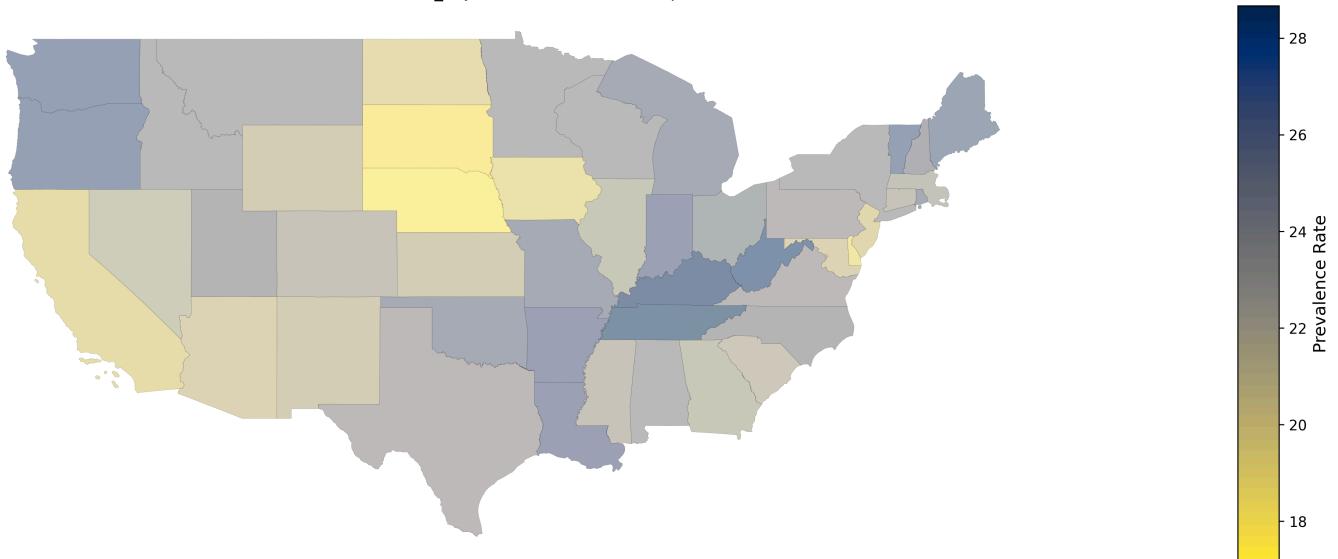
CSMOKING_AdjPrev - United States Map



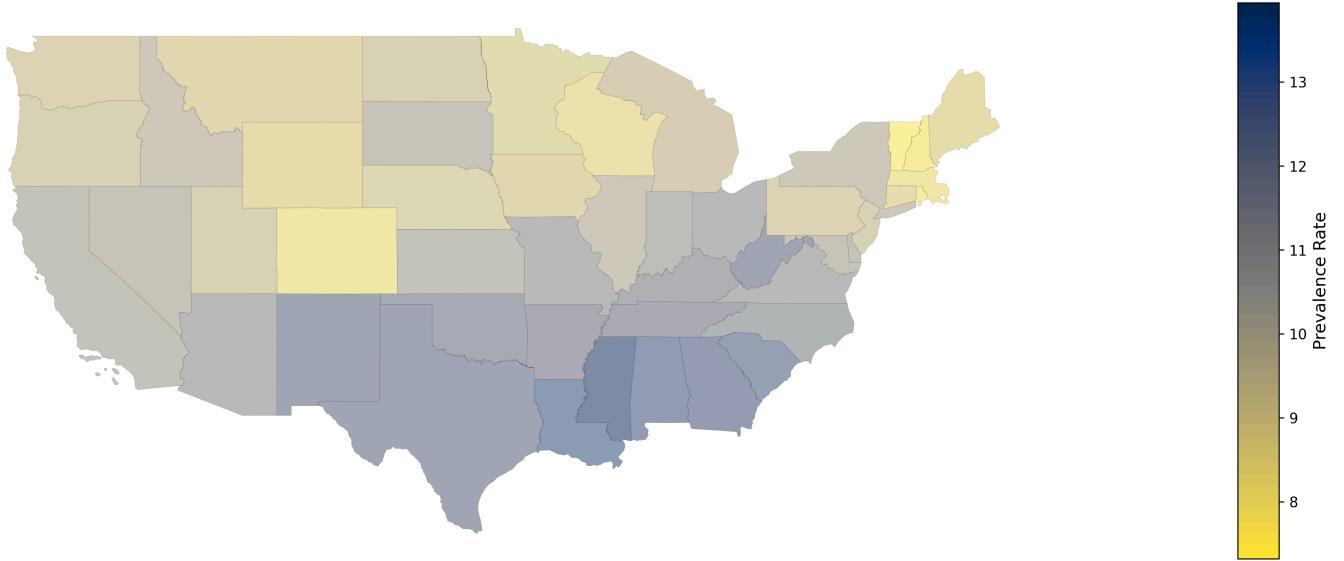
DENTAL_AdjPrev - United States Map



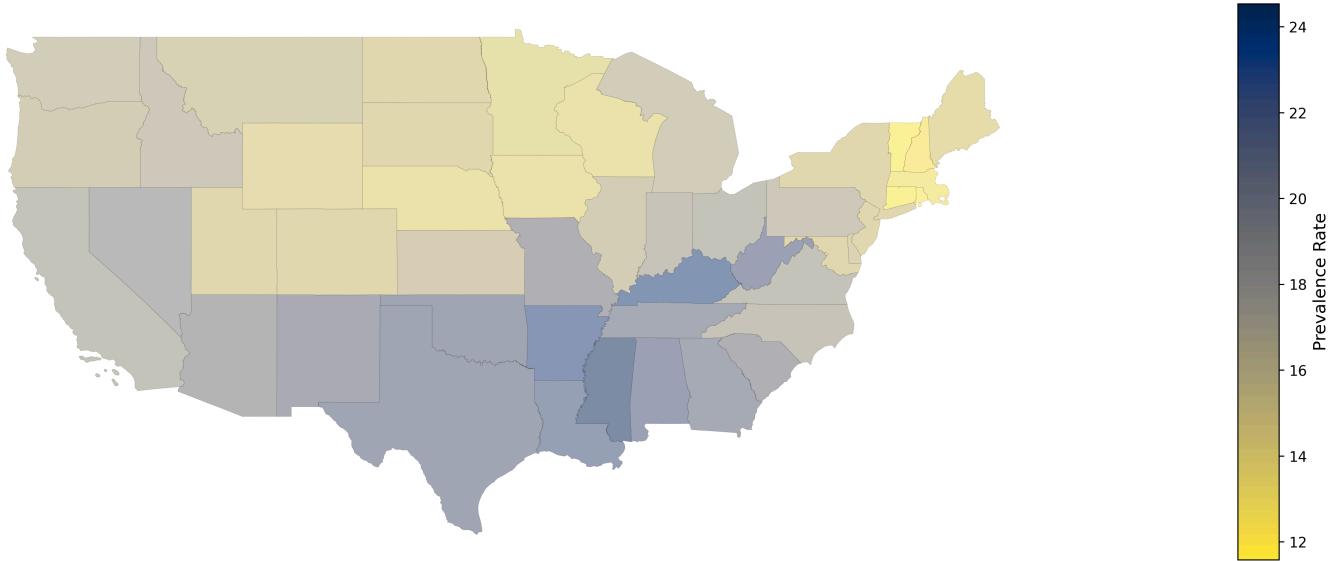
DEPRESSION_AdjPrev - United States Map



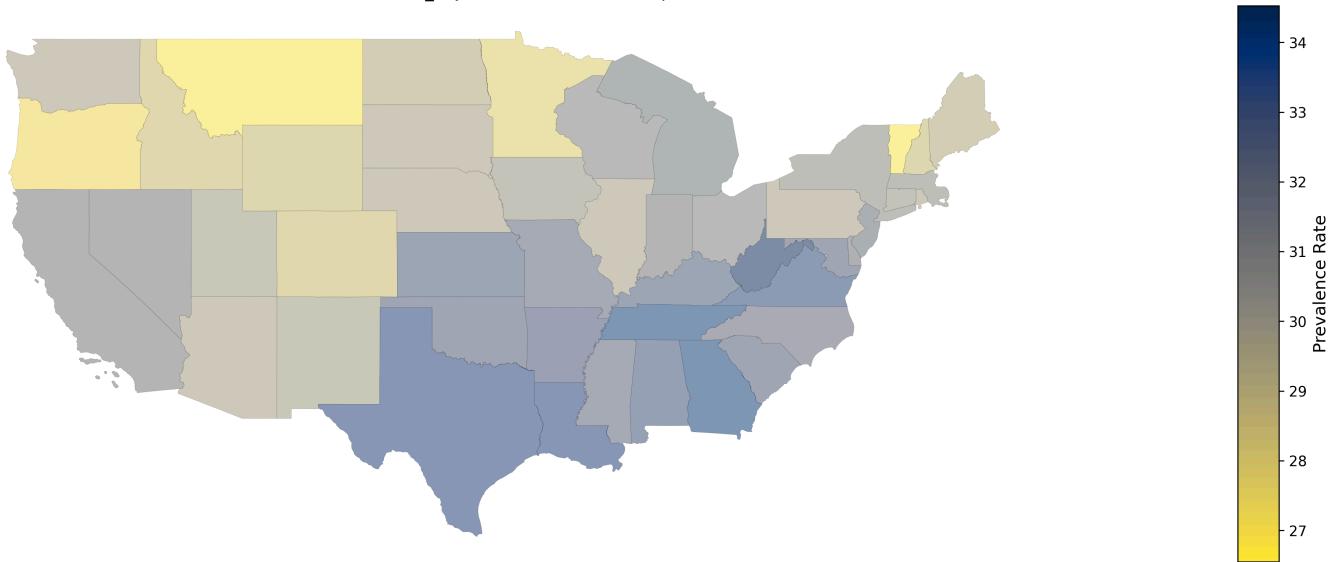
DIABETES_AdjPrev - United States Map



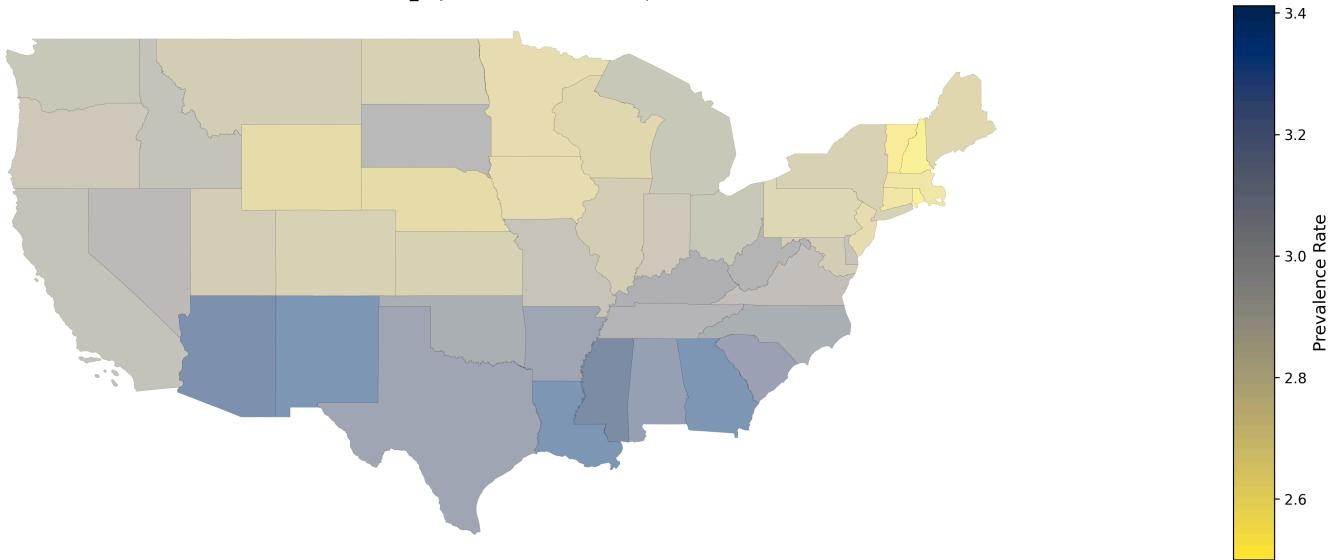
GHLTH_AdjPrev - United States Map



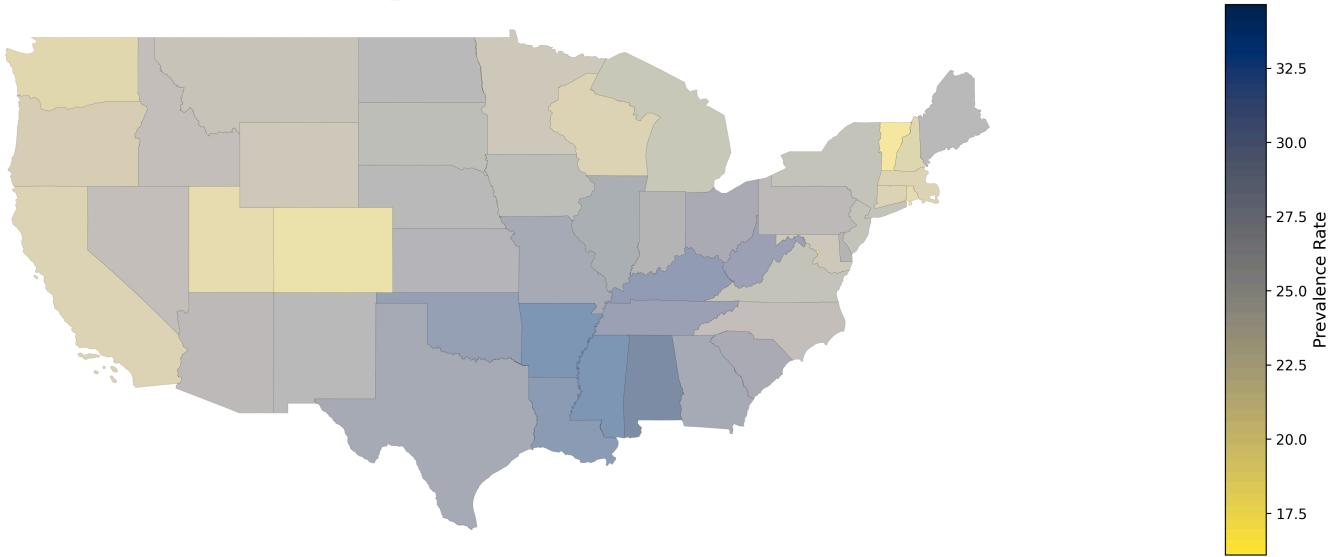
HIGHCHOL_AdjPrev - United States Map



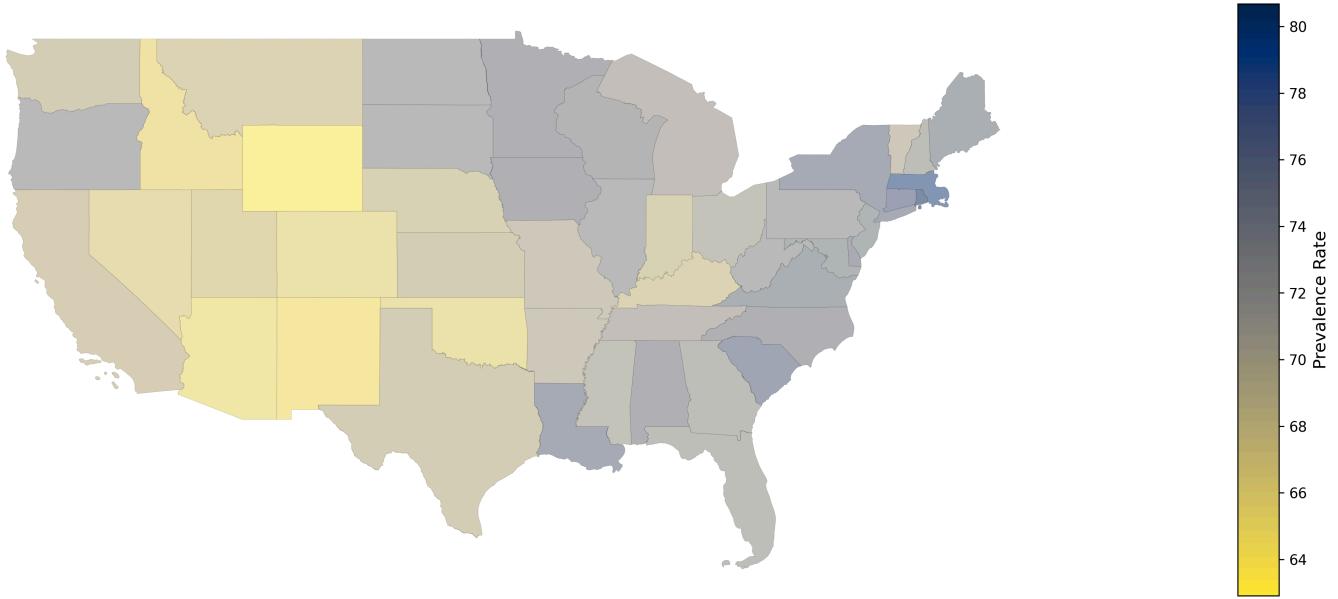
KIDNEY_AdjPrev - United States Map



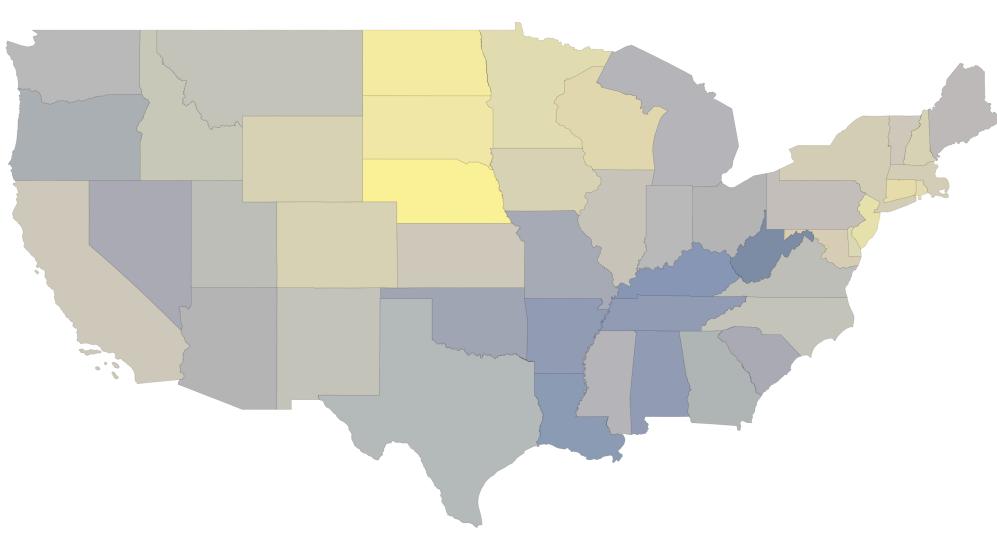
LPA_AdjPrev - United States Map



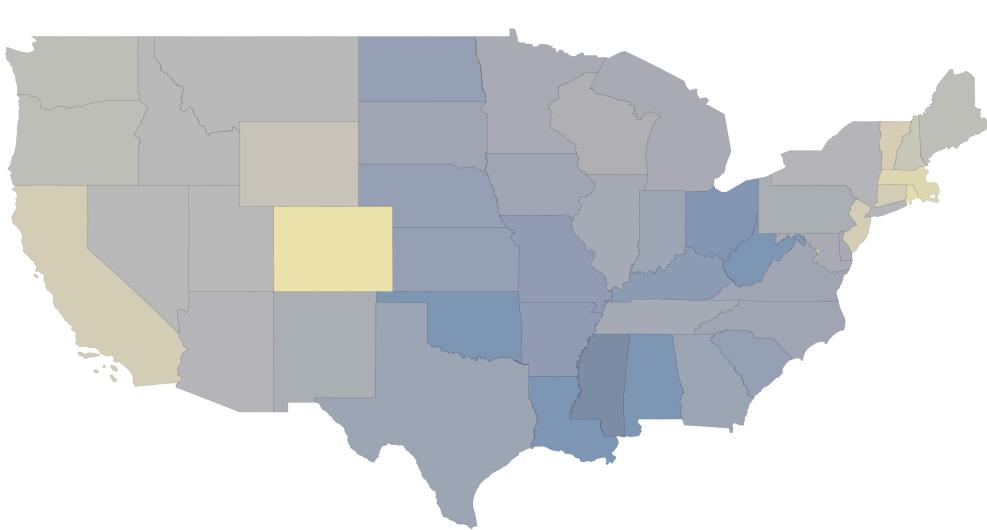
MAMMOUSE_AdjPrev - United States Map



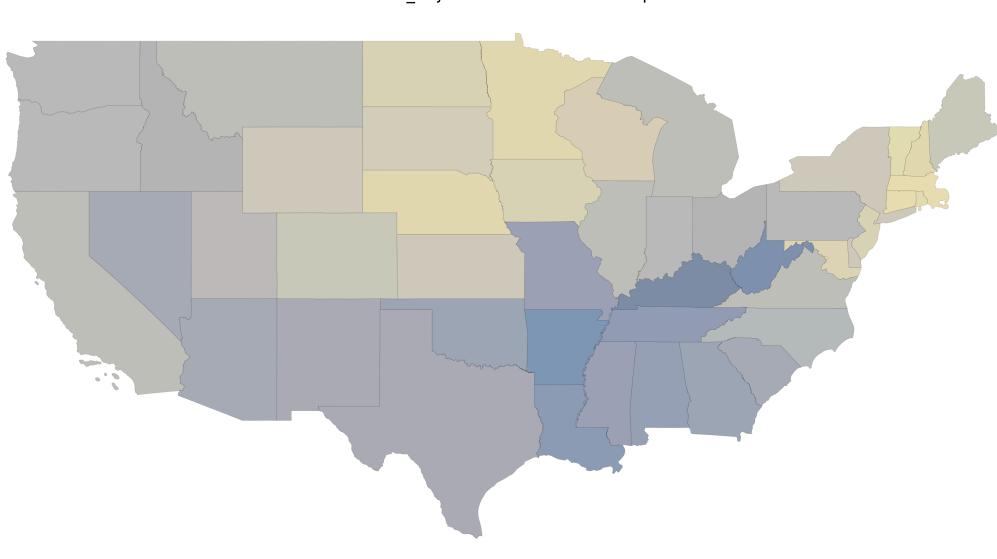
MHLTH_AdjPrev - United States Map



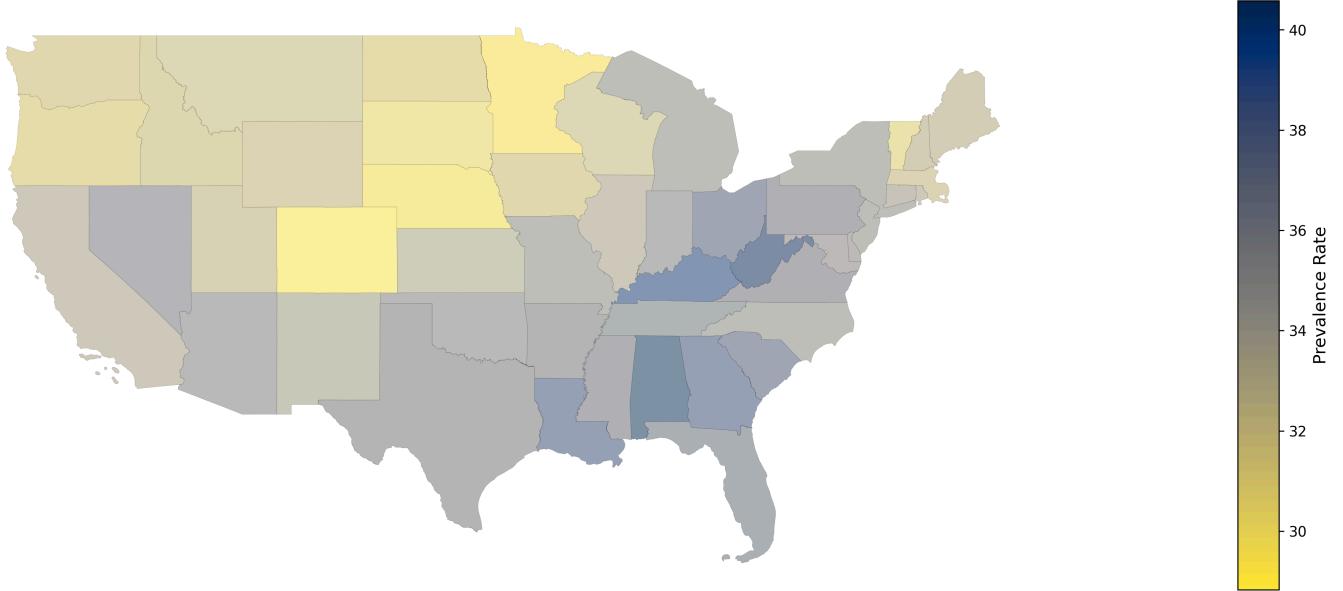
OBESITY_AdjPrev - United States Map



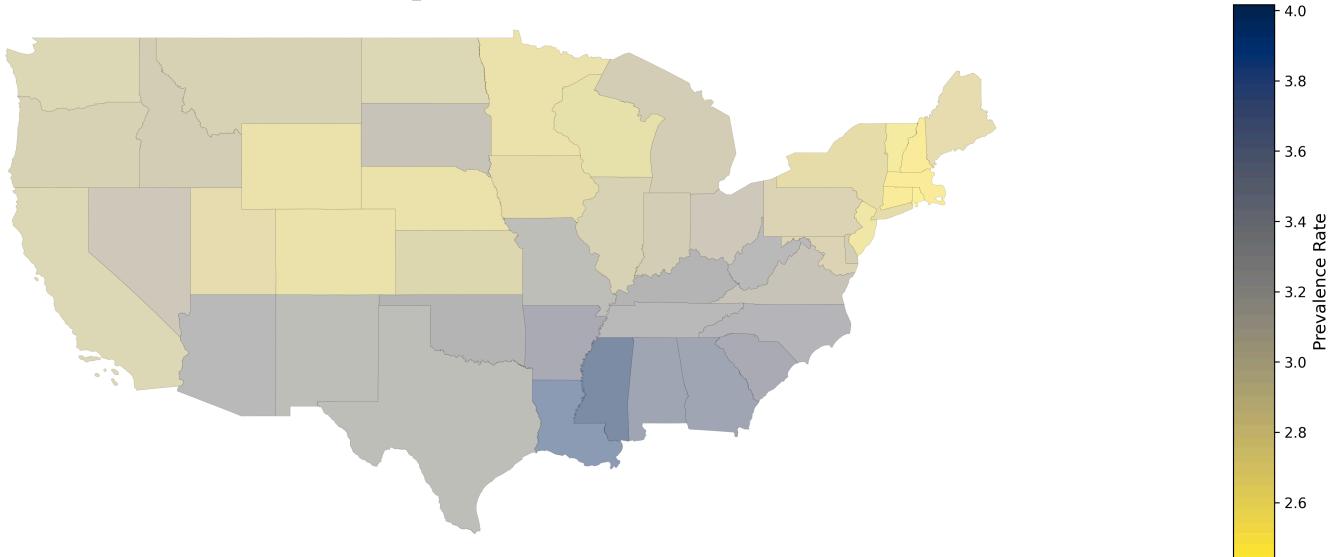
PHLTH_AdjPrev - United States Map



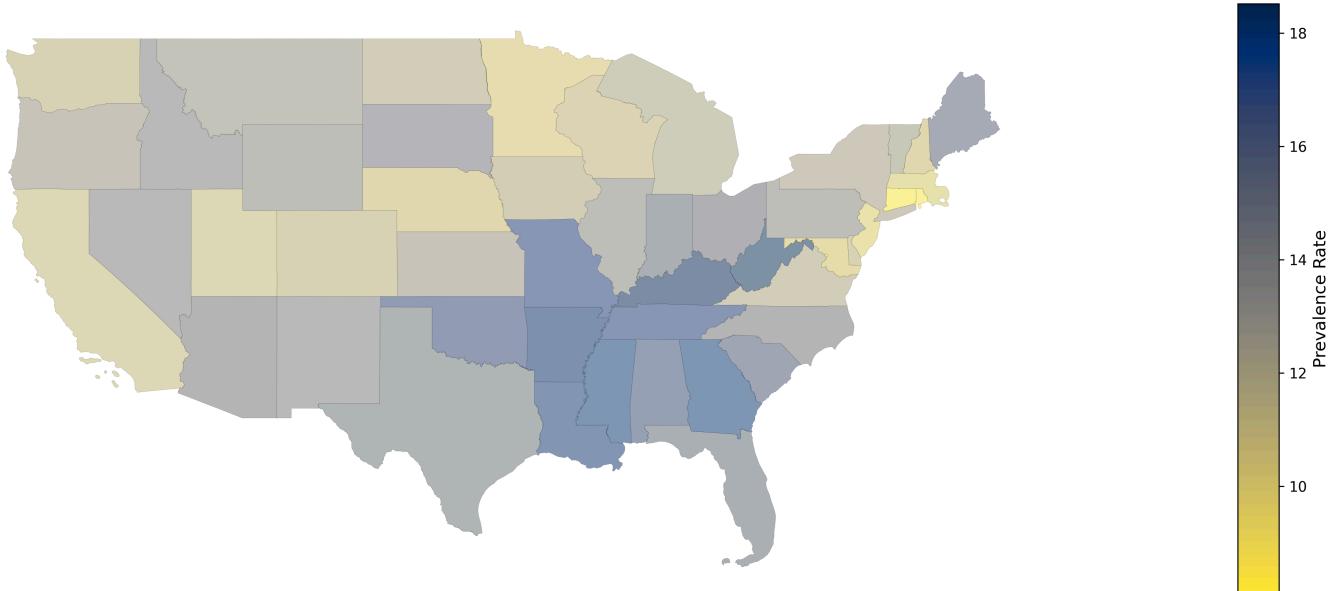
SLEEP_AdjPrev - United States Map



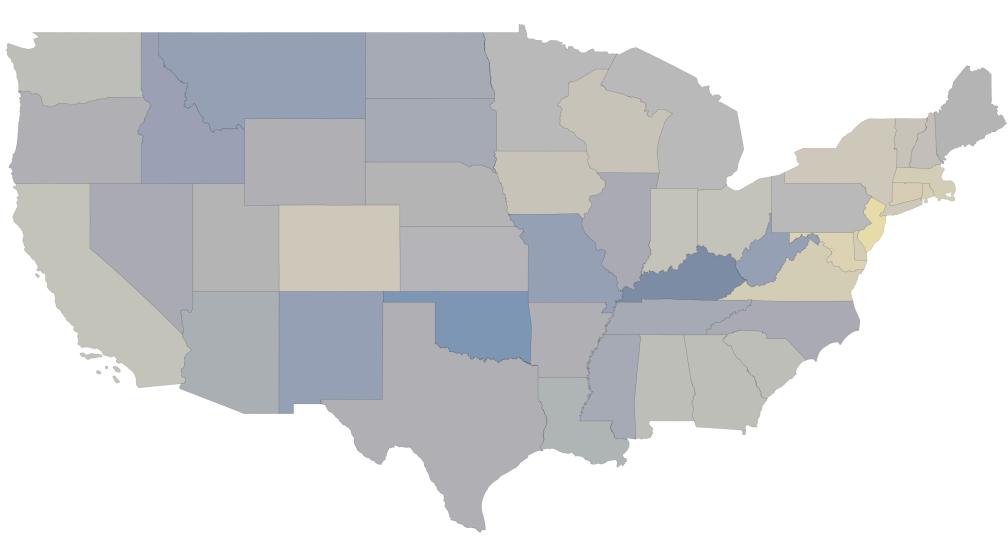
STROKE_AdjPrev - United States Map



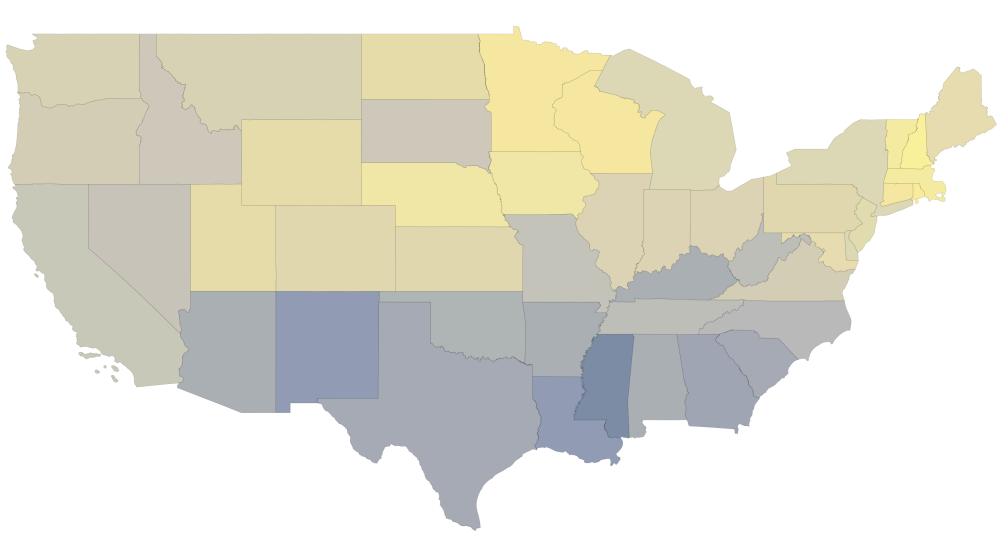
TEETHLOST_AdjPrev - United States Map



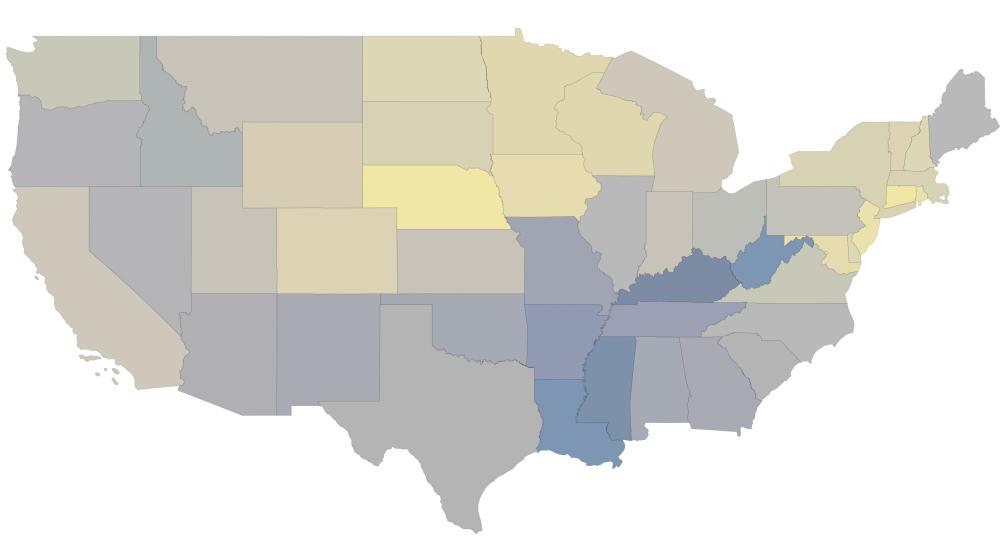
HEARING_AdjPrev - United States Map



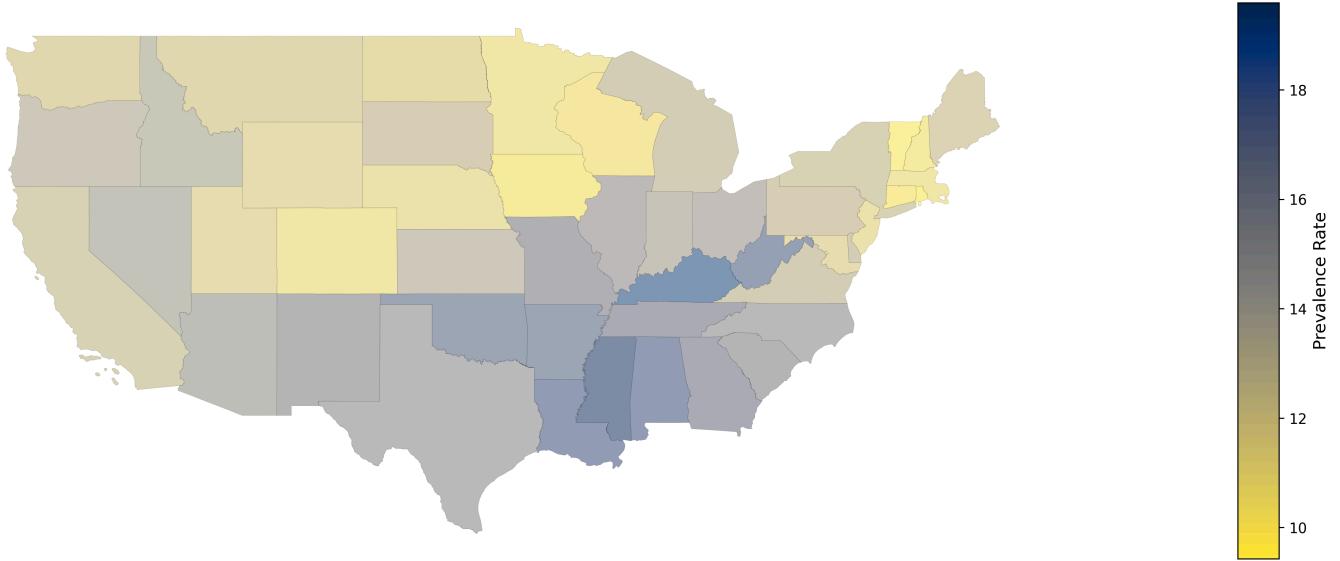
VISION_AdjPrev - United States Map



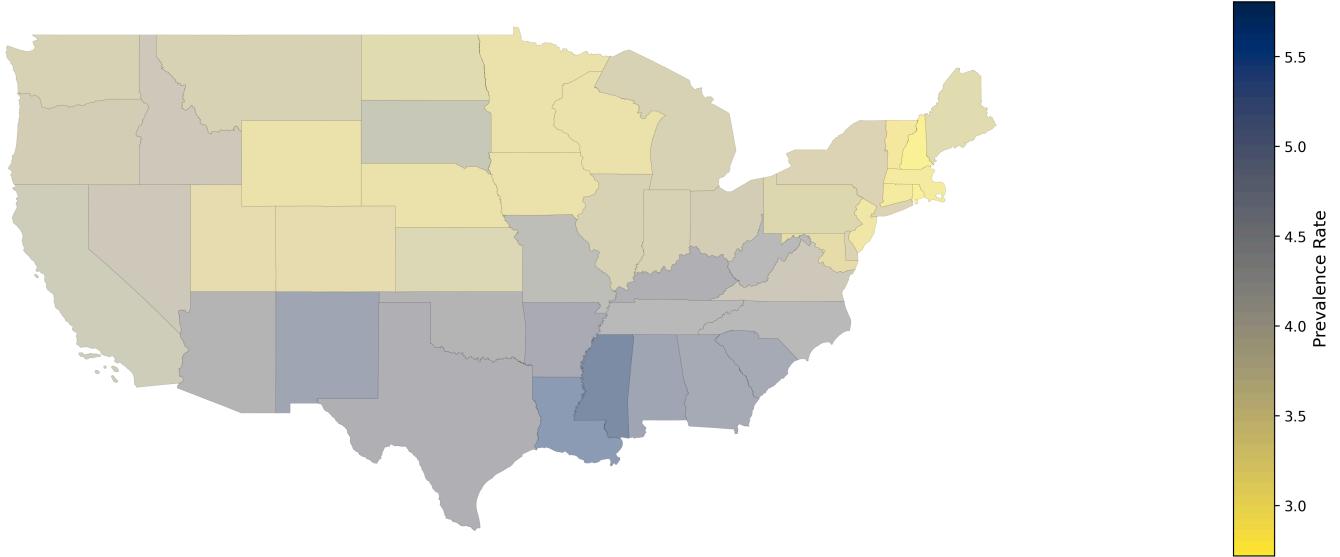
COGNITION_AdjPrev - United States Map



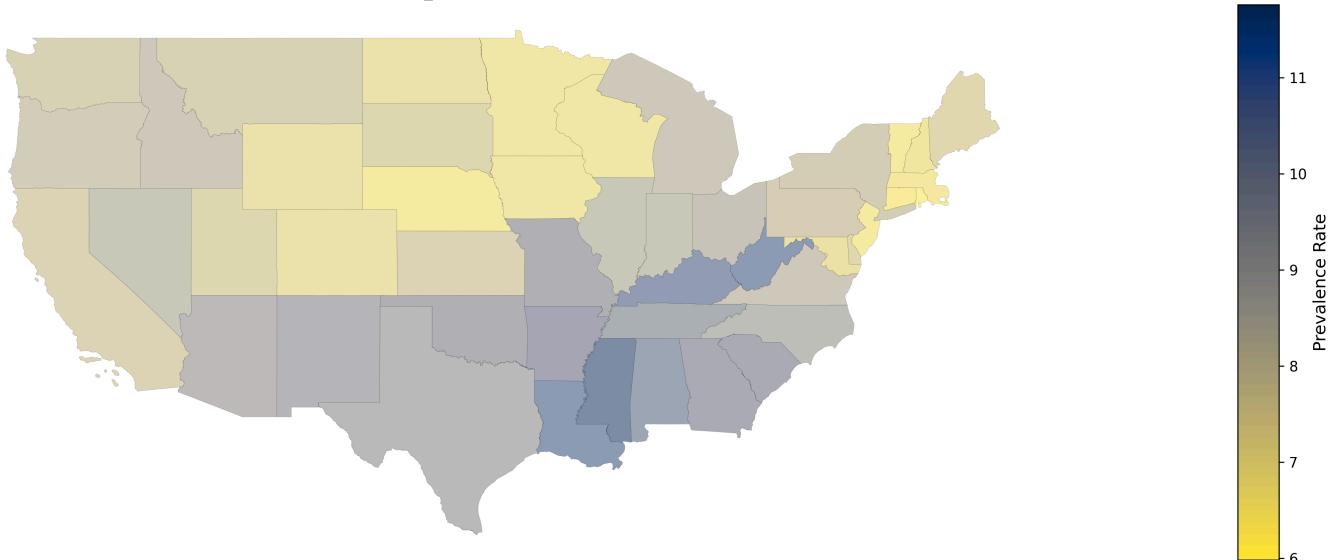
MOBILITY_AdjPrev - United States Map

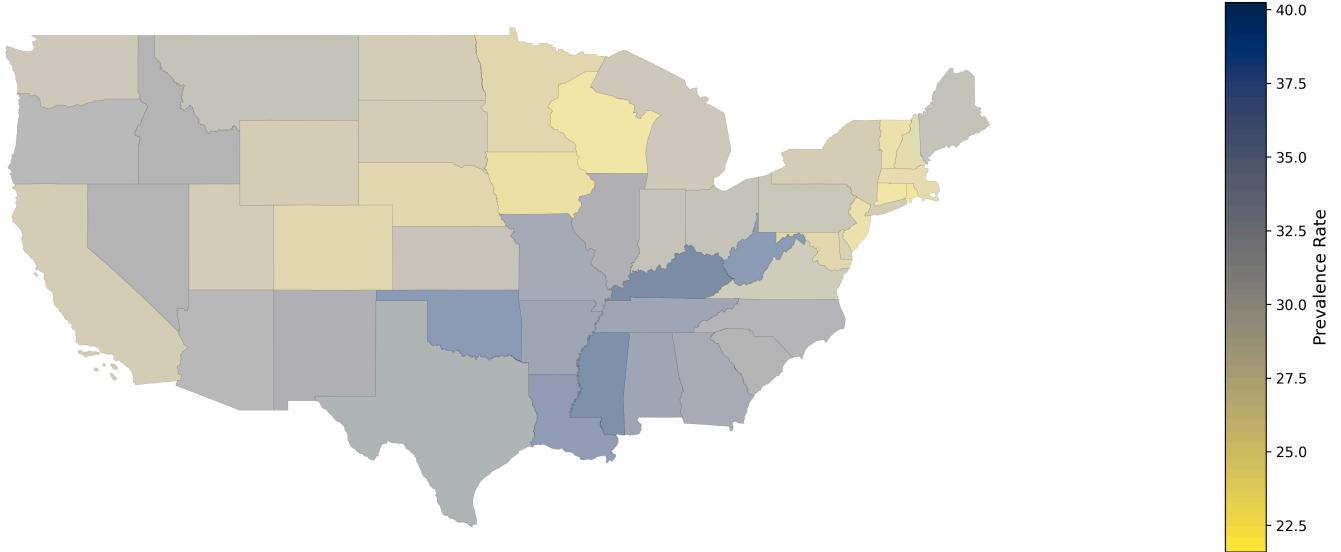


SELCARE_AdjPrev - United States Map



INDEPLIVE_AdjPrev - United States Map





A Focus on Illinois State

Filter by the State of Illinois

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

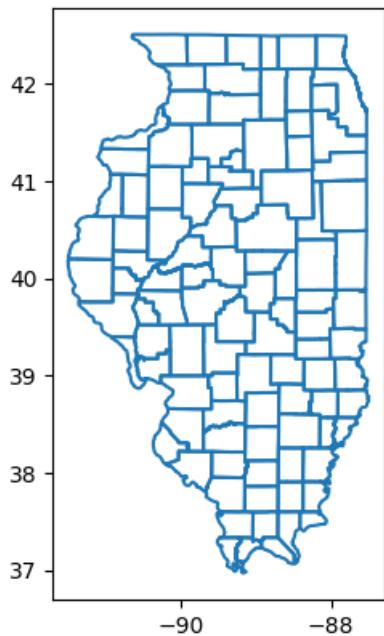
Import Illinois State Shapefile

```
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_23392\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

```
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, 11.7)
```

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, 11.7)
          1      BOONE        7 POLYGON((-88.70742, 42.49351, -88.70740, 42.493...  IL      Illinois      BOONE         17007          53159            10.6   ( 8.1, 11.7)
```

2 rows × 157 columns

Plotting

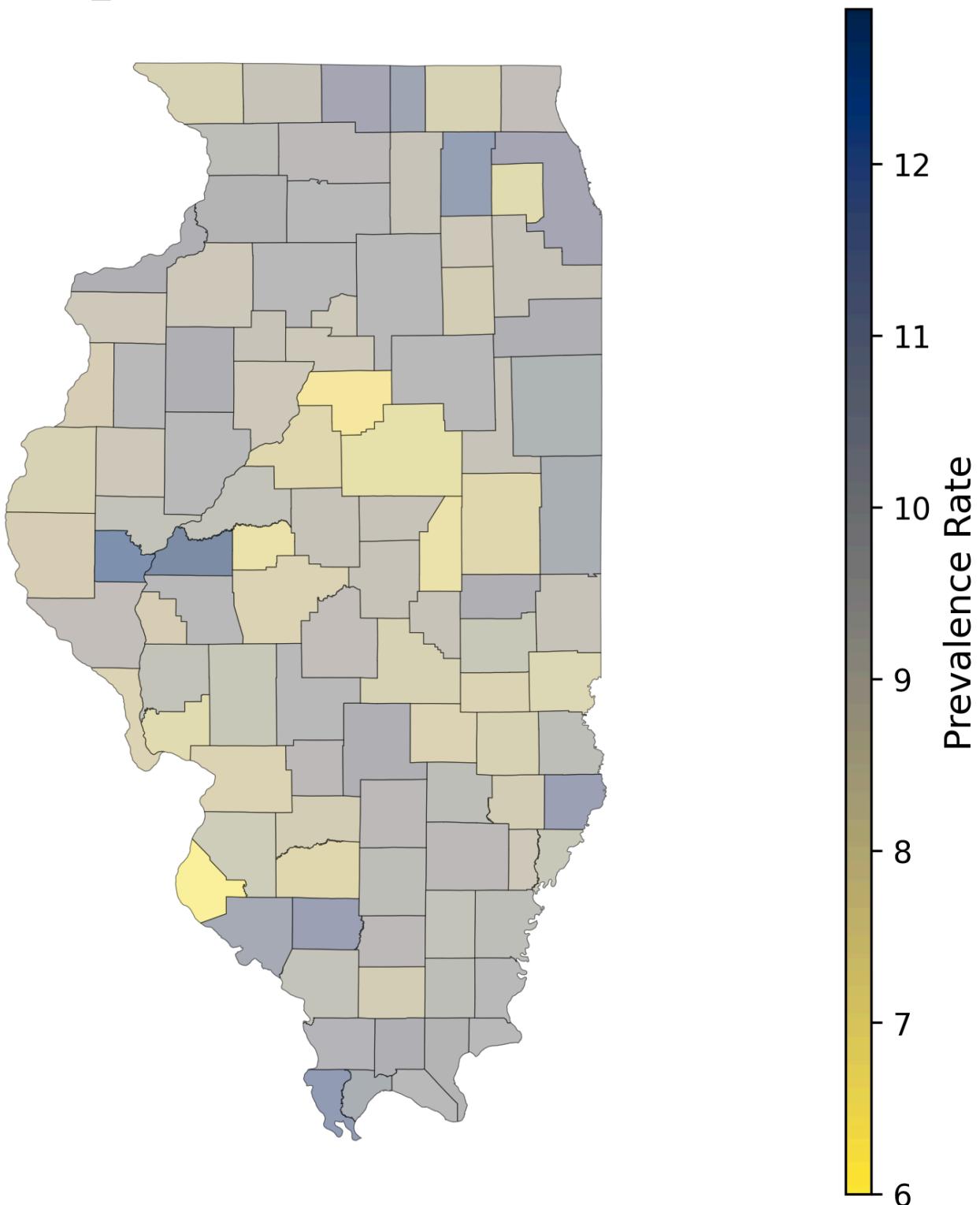
```
In [ ]: import matplotlib.pyplot as plt
from mpl_toolkits.axes_grid1 import make_axes_locatable

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()
```

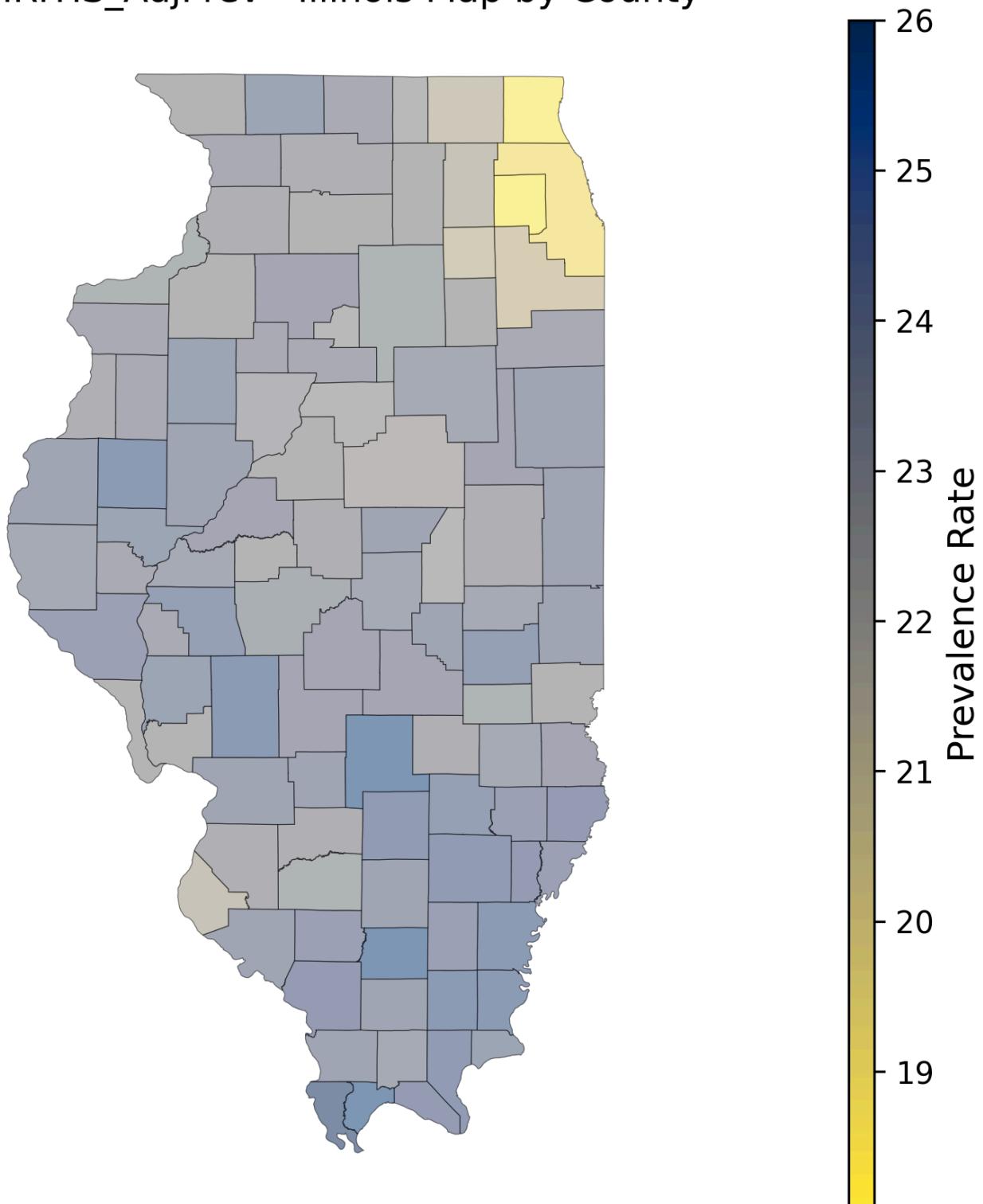
```
In [ ]: # Get columns ending with '_CrudePrev'
columns_to_plot = [col for col in df.columns if col.endswith('_AdjPrev')]

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

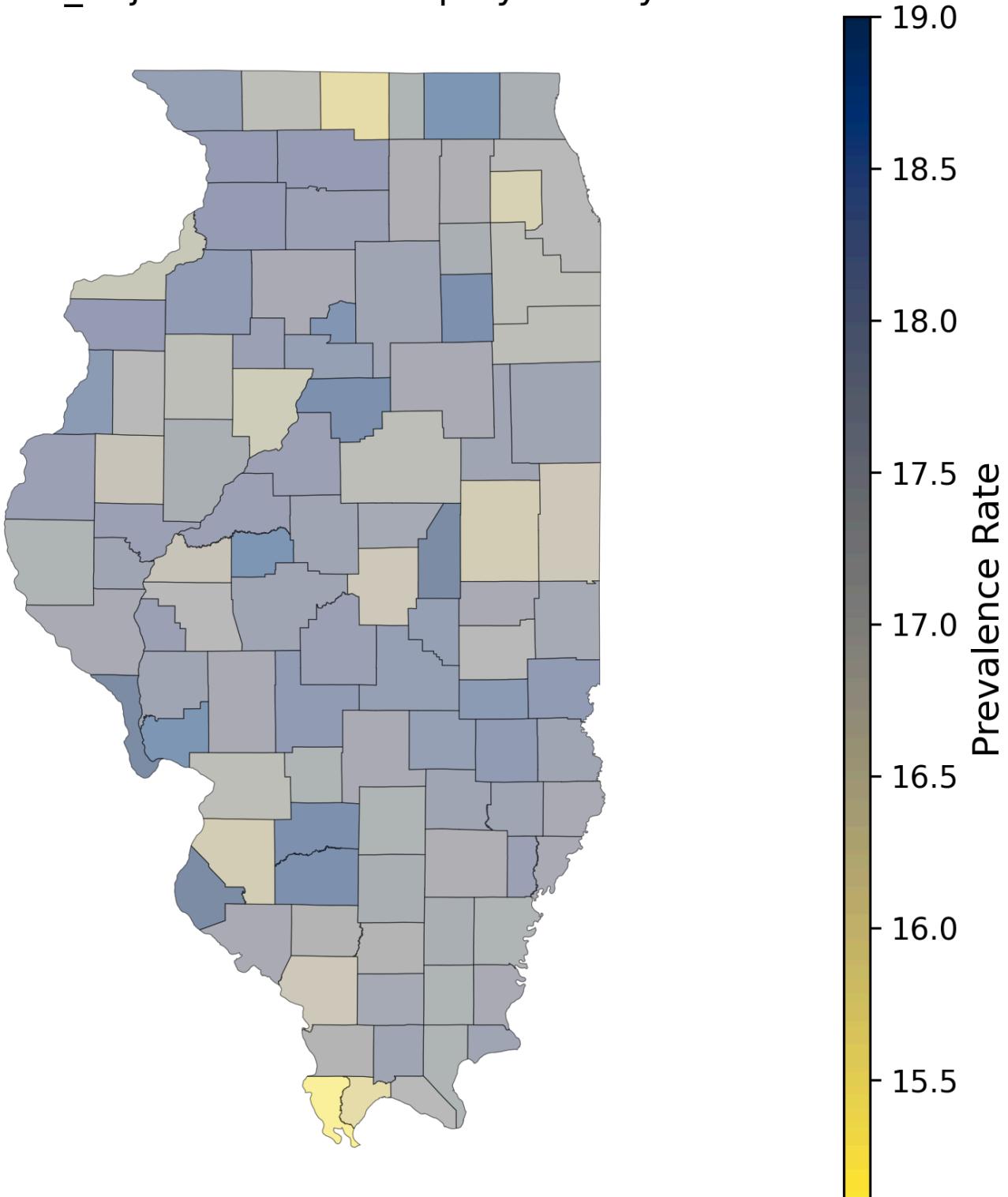
ACCESS2_AdjPrev - Illinois Map by County



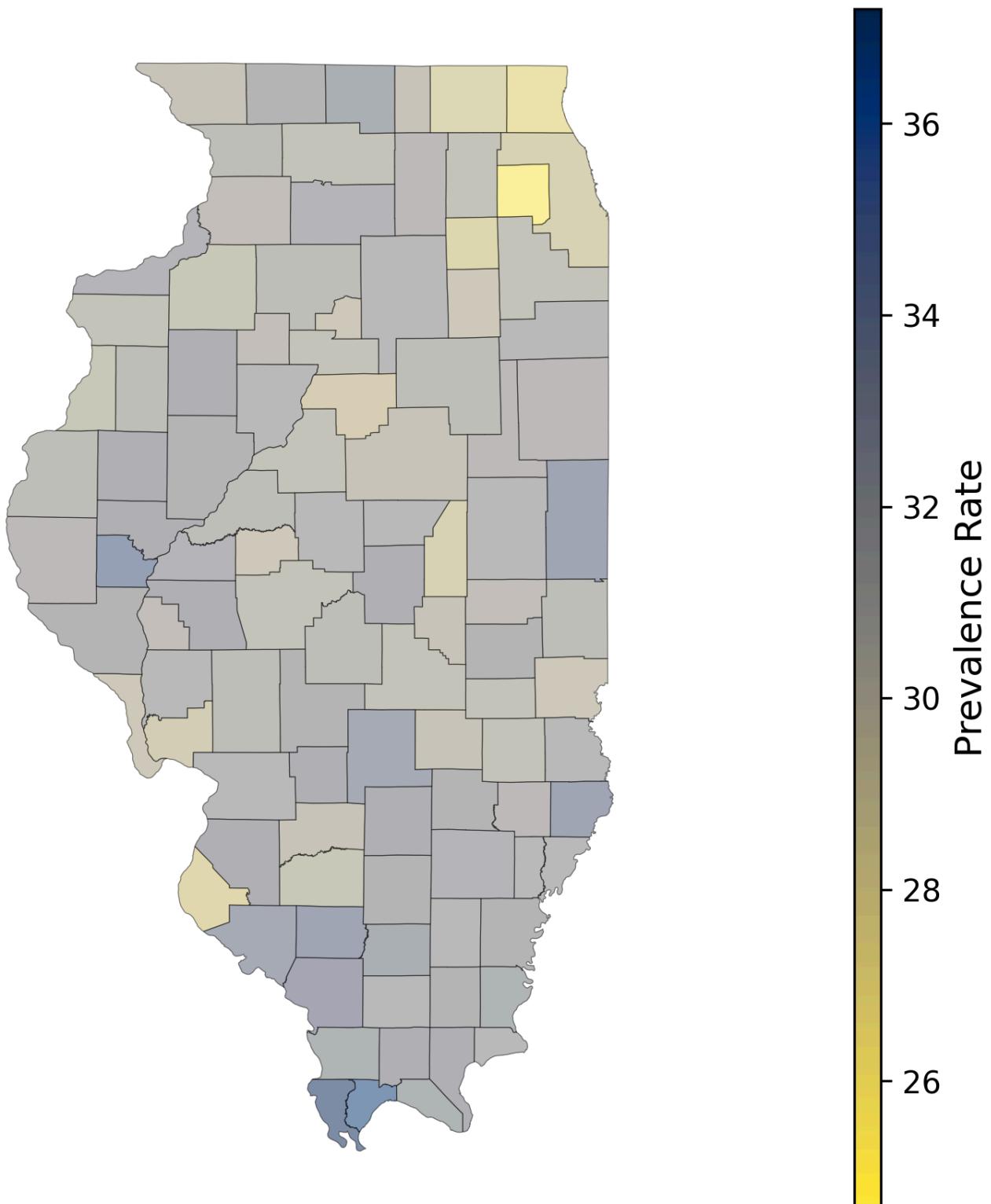
ARTHRITIS_AdjPrev - Illinois Map by County



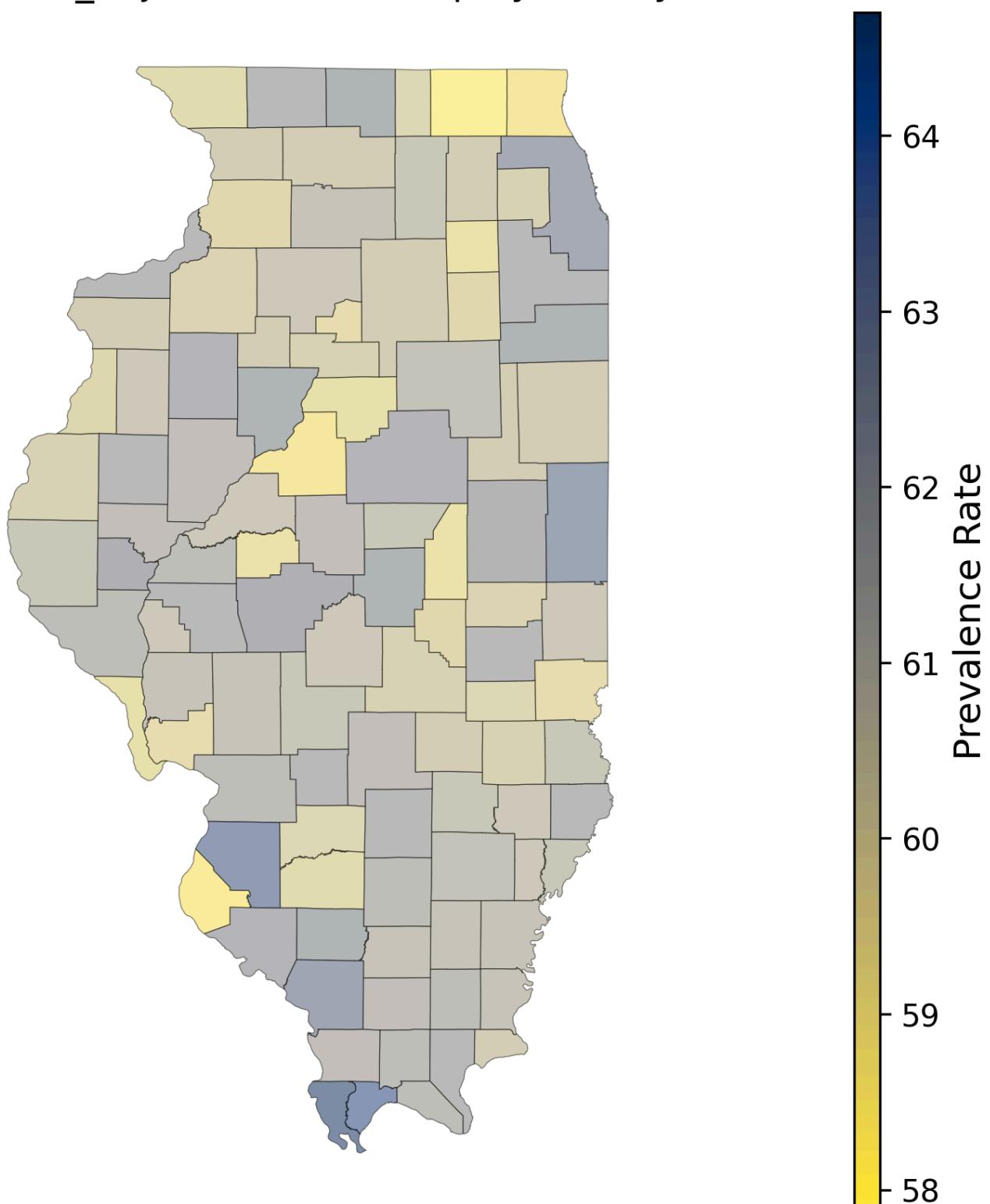
BINGE_AdjPrev - Illinois Map by County



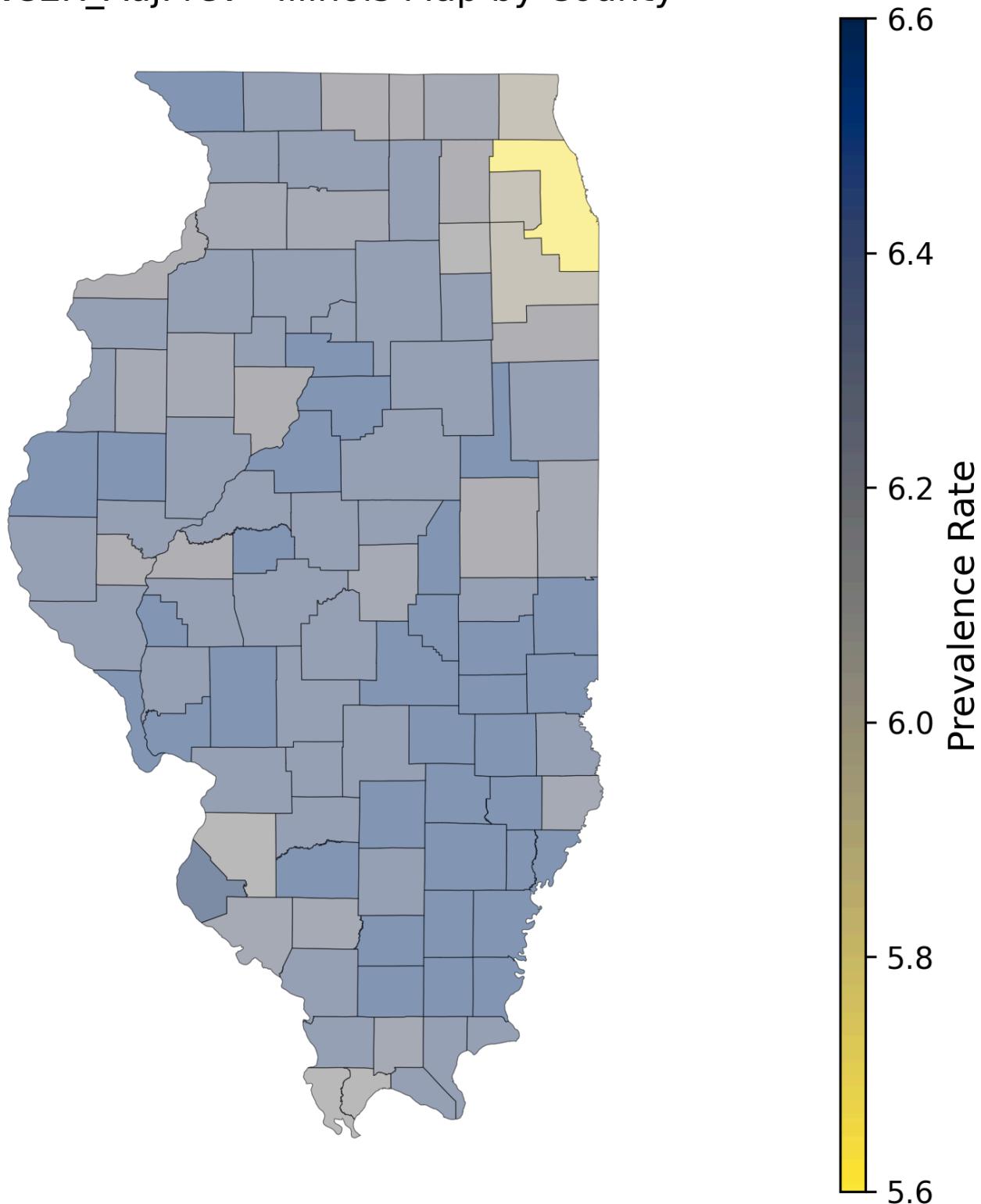
BPHIGH_AdjPrev - Illinois Map by County



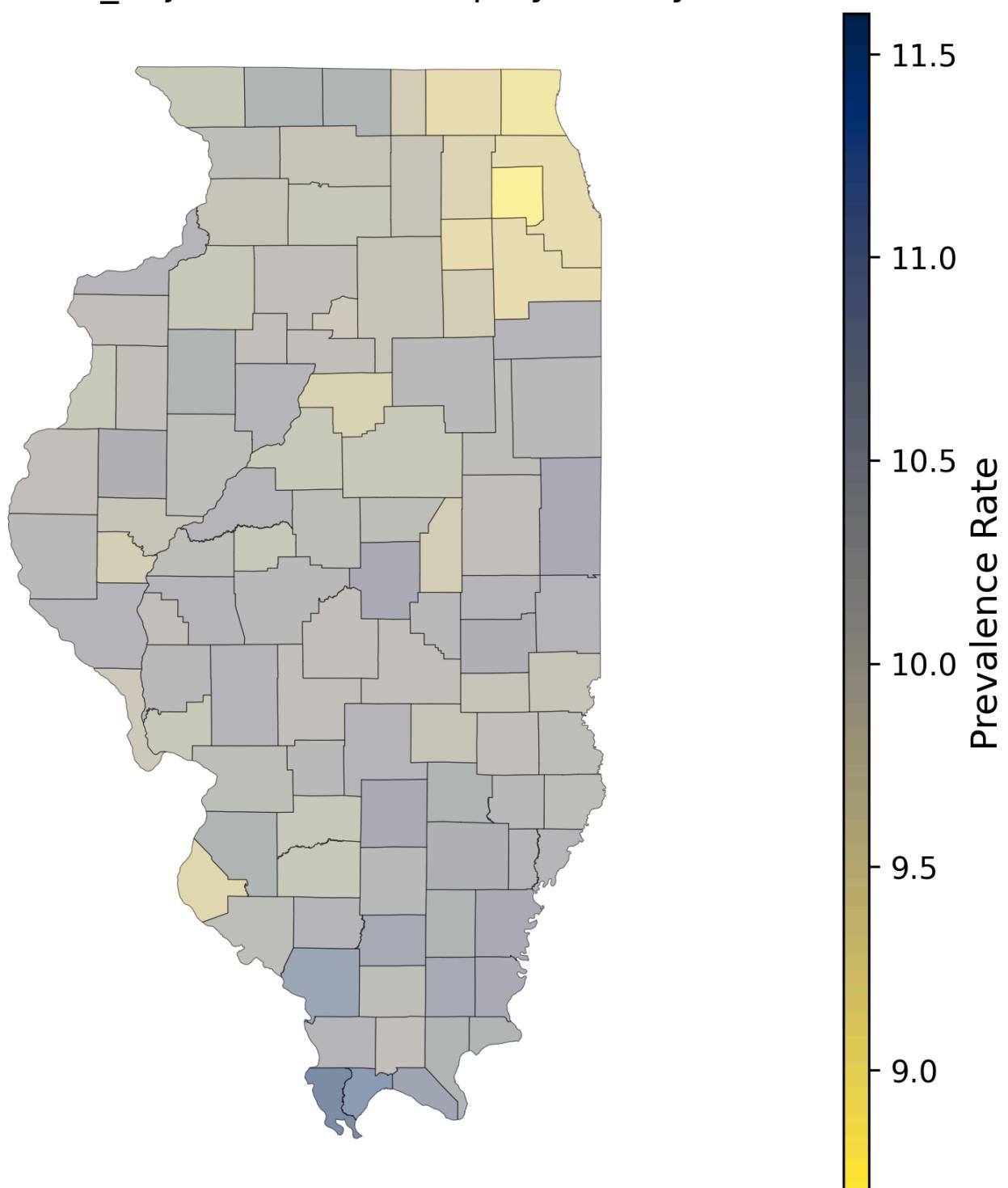
BPMED_AdjPrev - Illinois Map by County



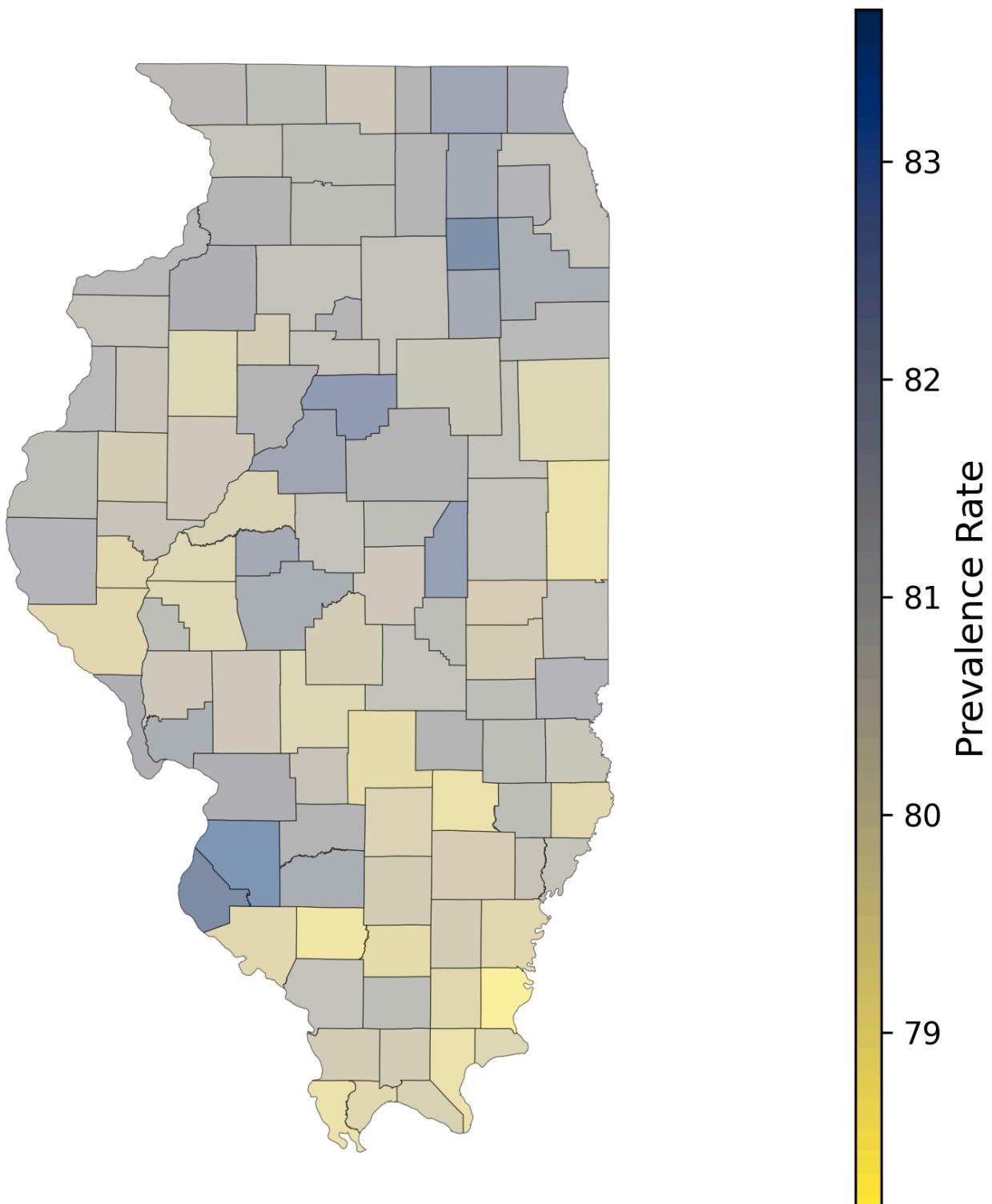
CANCER_AdjPrev - Illinois Map by County



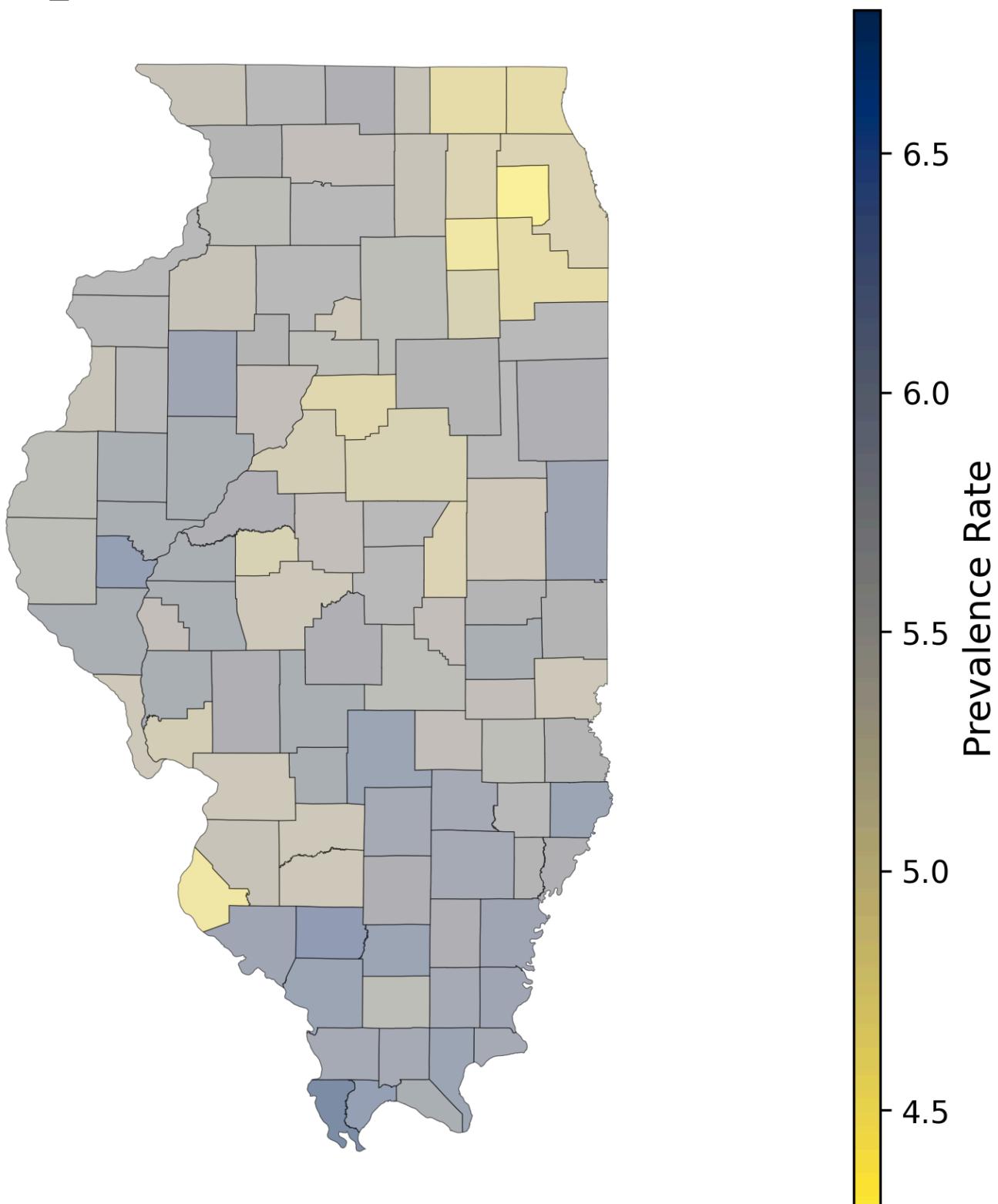
CASTHMA_AdjPrev - Illinois Map by County



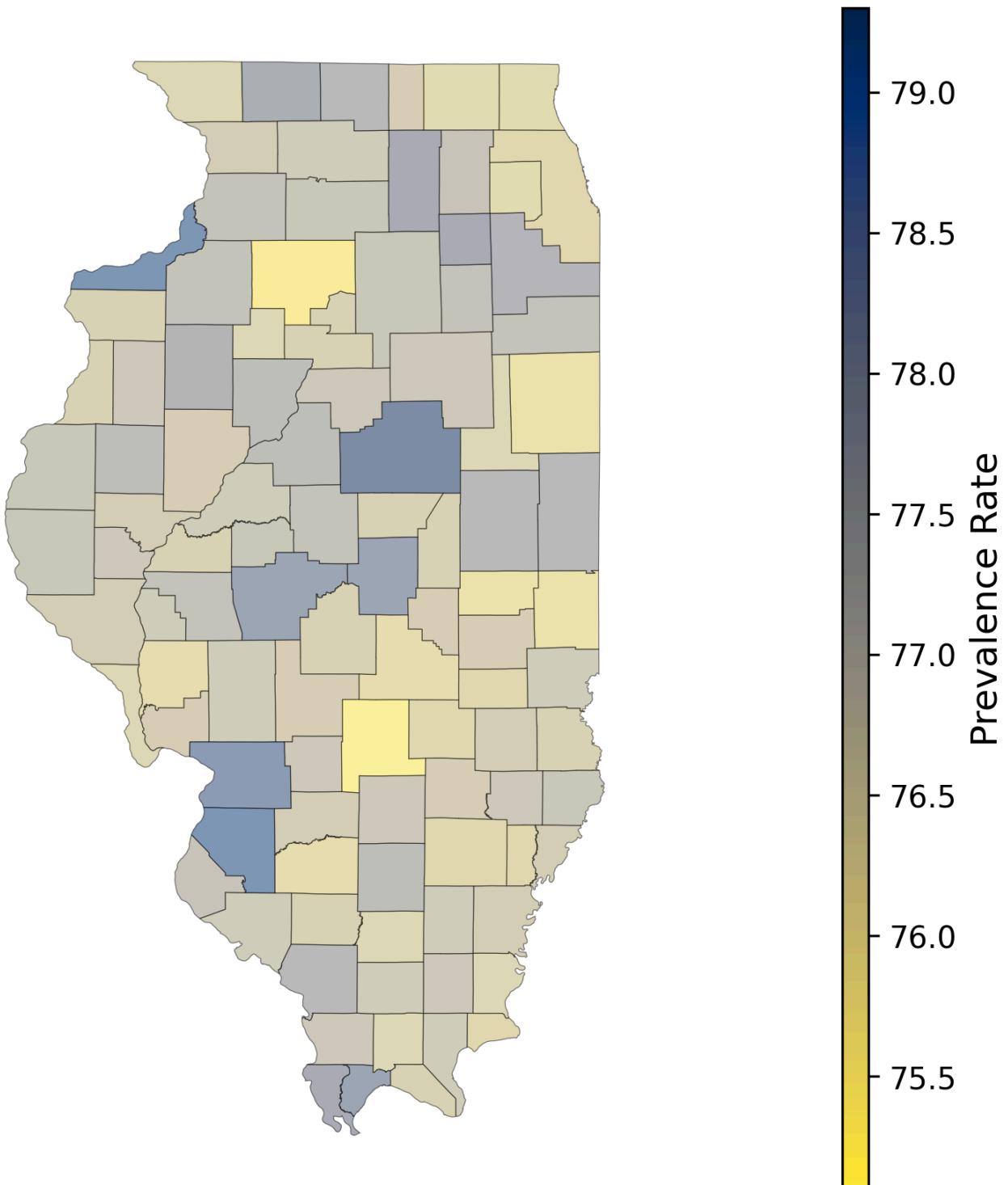
CERVICAL_AdjPrev - Illinois Map by County



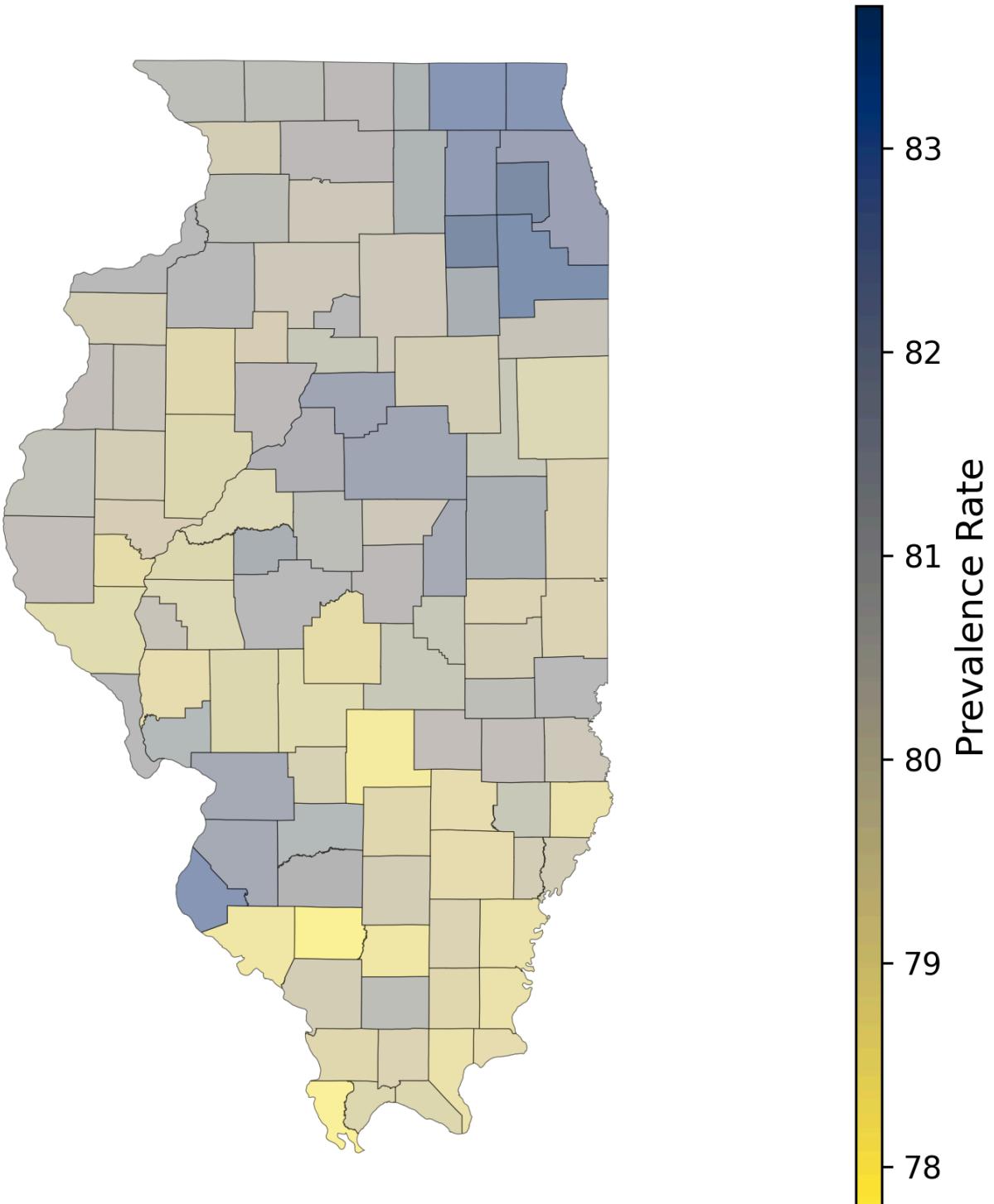
CHD_AdjPrev - Illinois Map by County



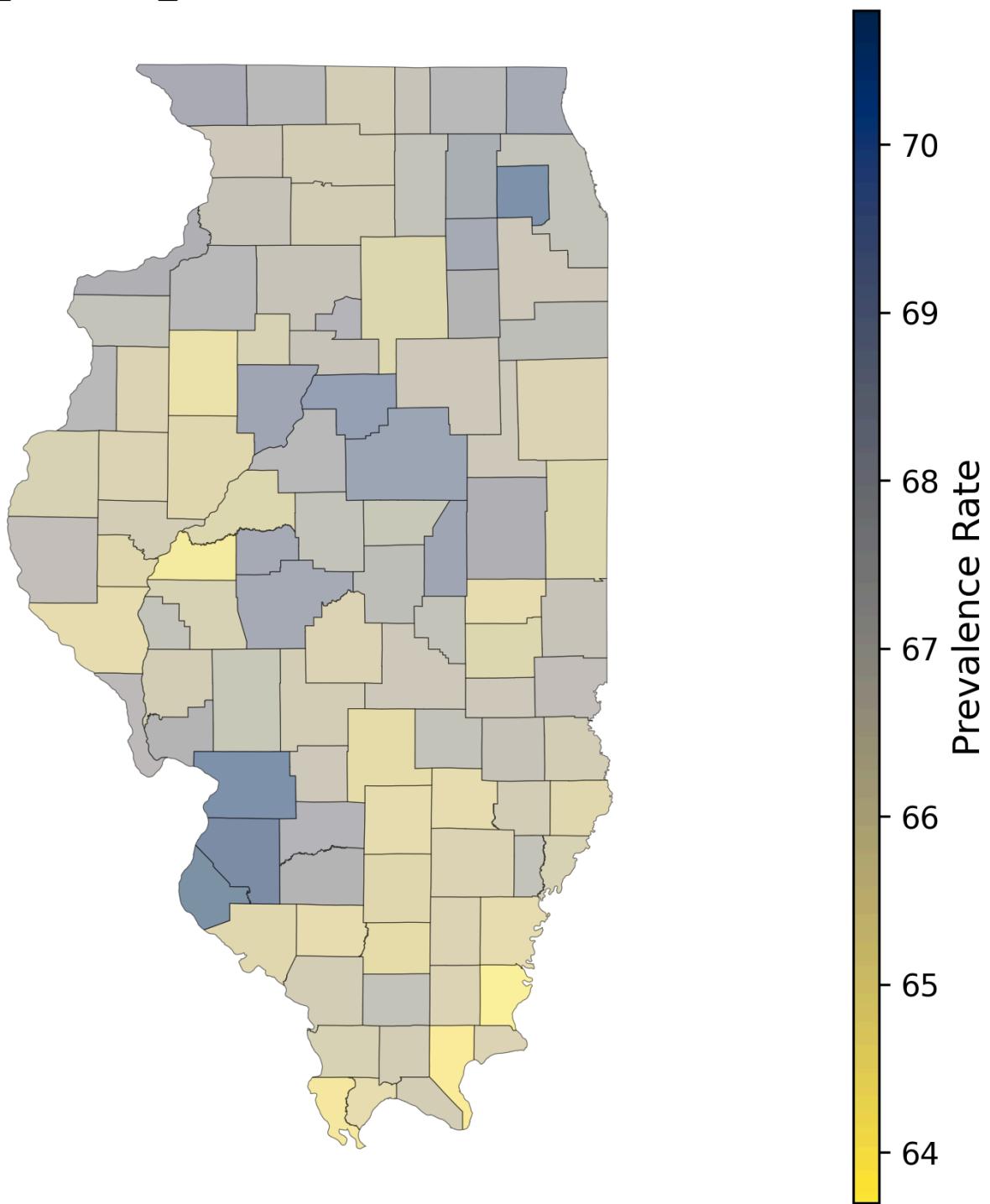
CHECKUP_AdjPrev - Illinois Map by County



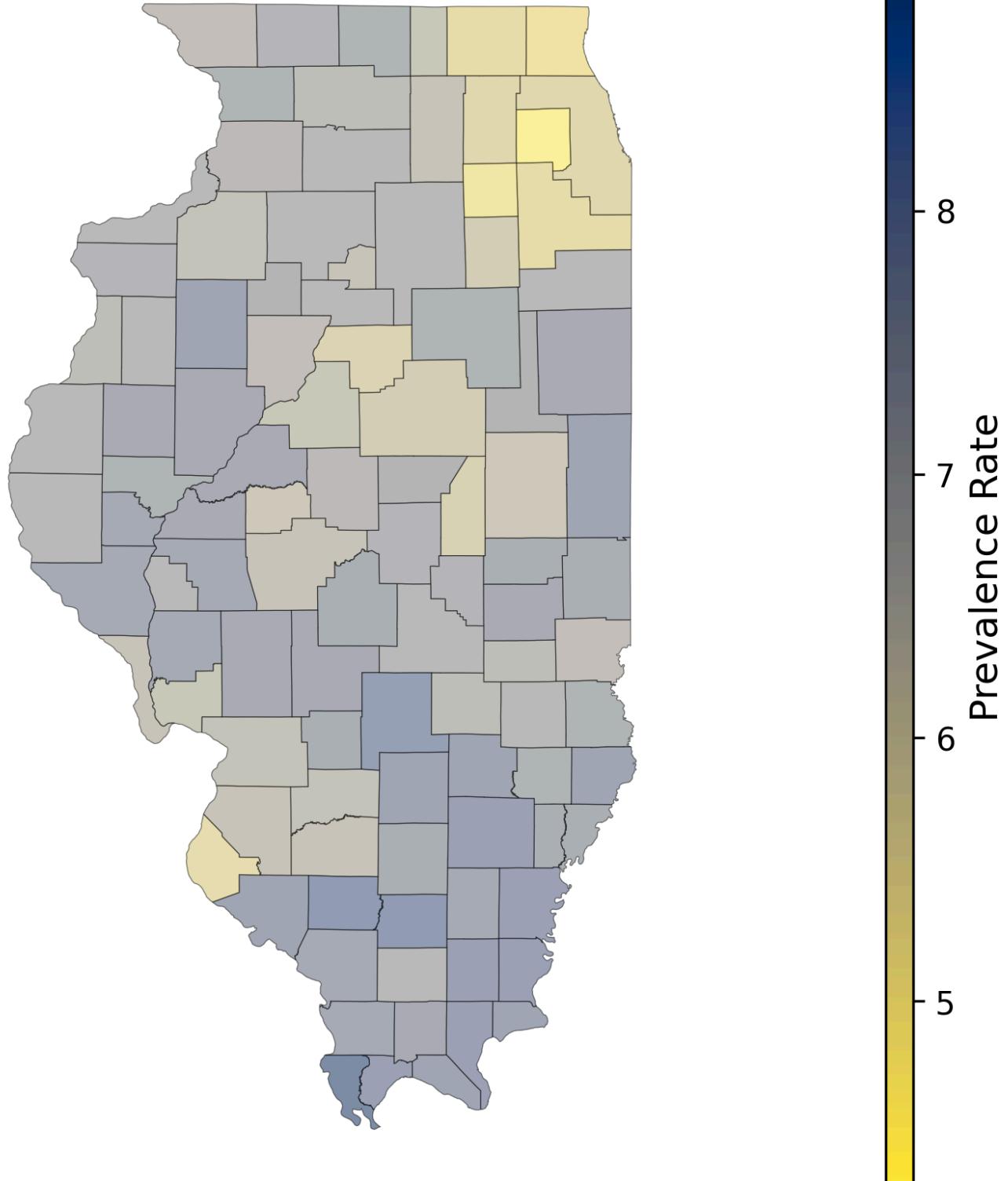
CHOLSCREEN_AdjPrev - Illinois Map by County



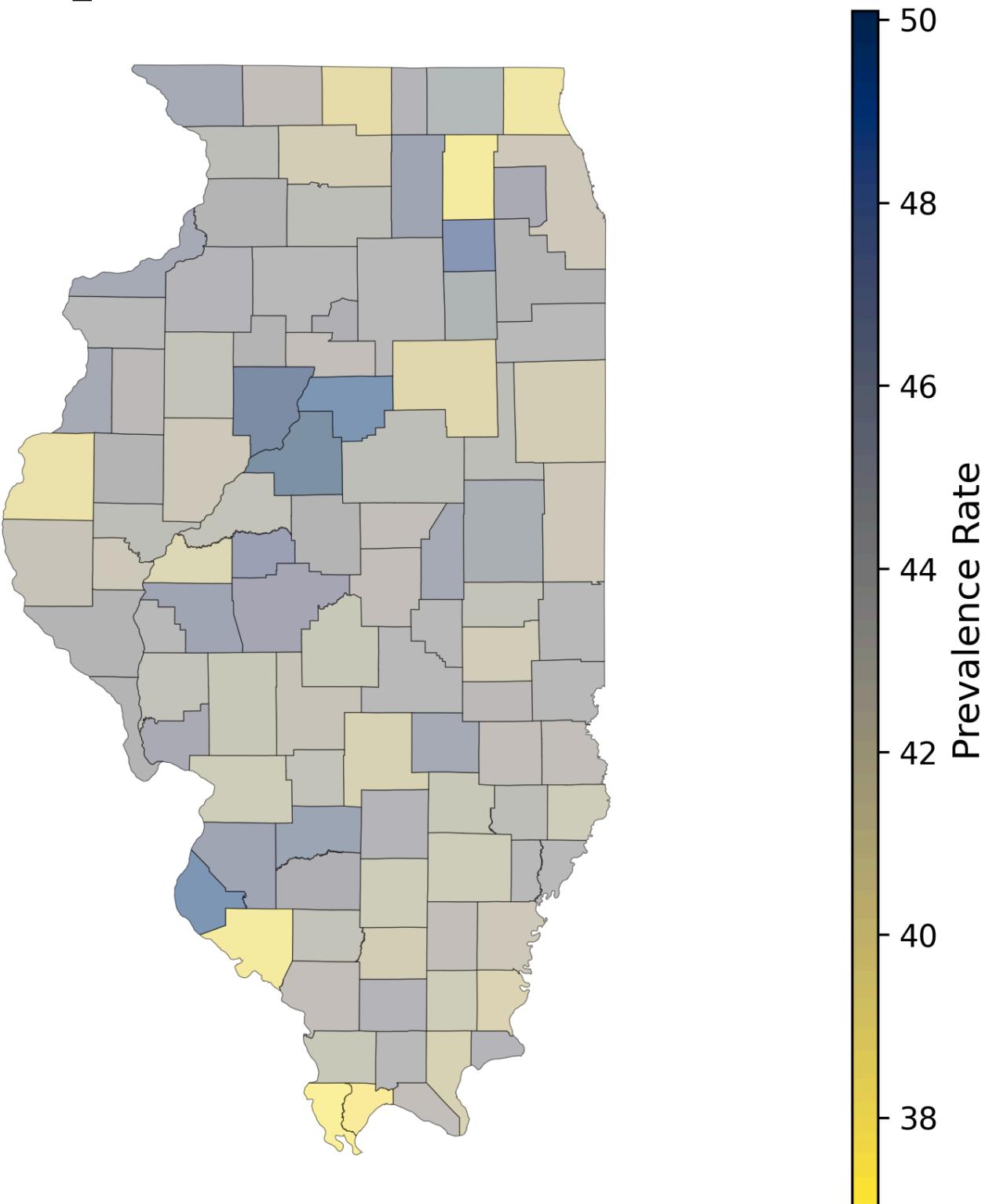
COLON_SCREEN_AdjPrev - Illinois Map by County



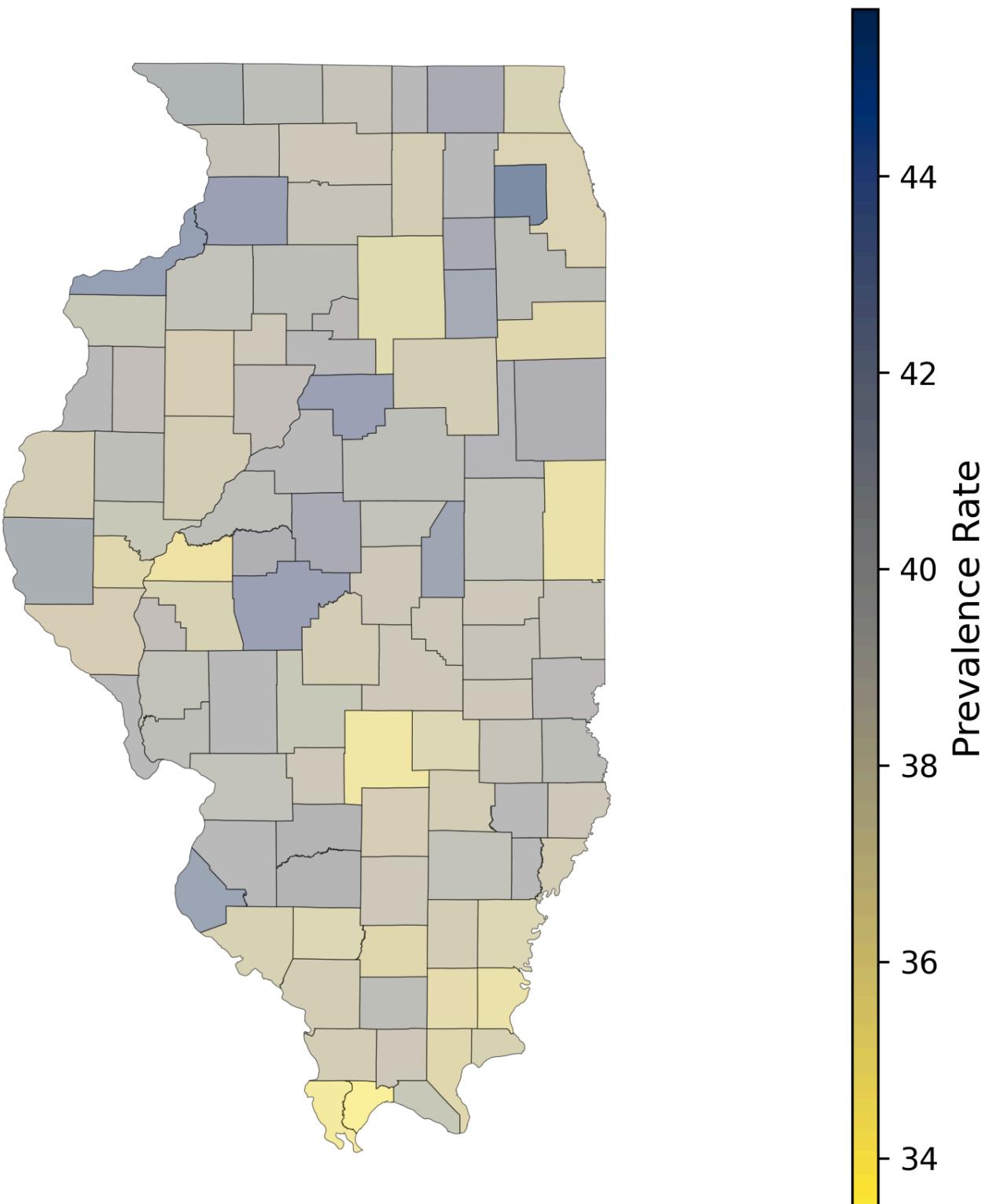
COPD_AdjPrev - Illinois Map by County



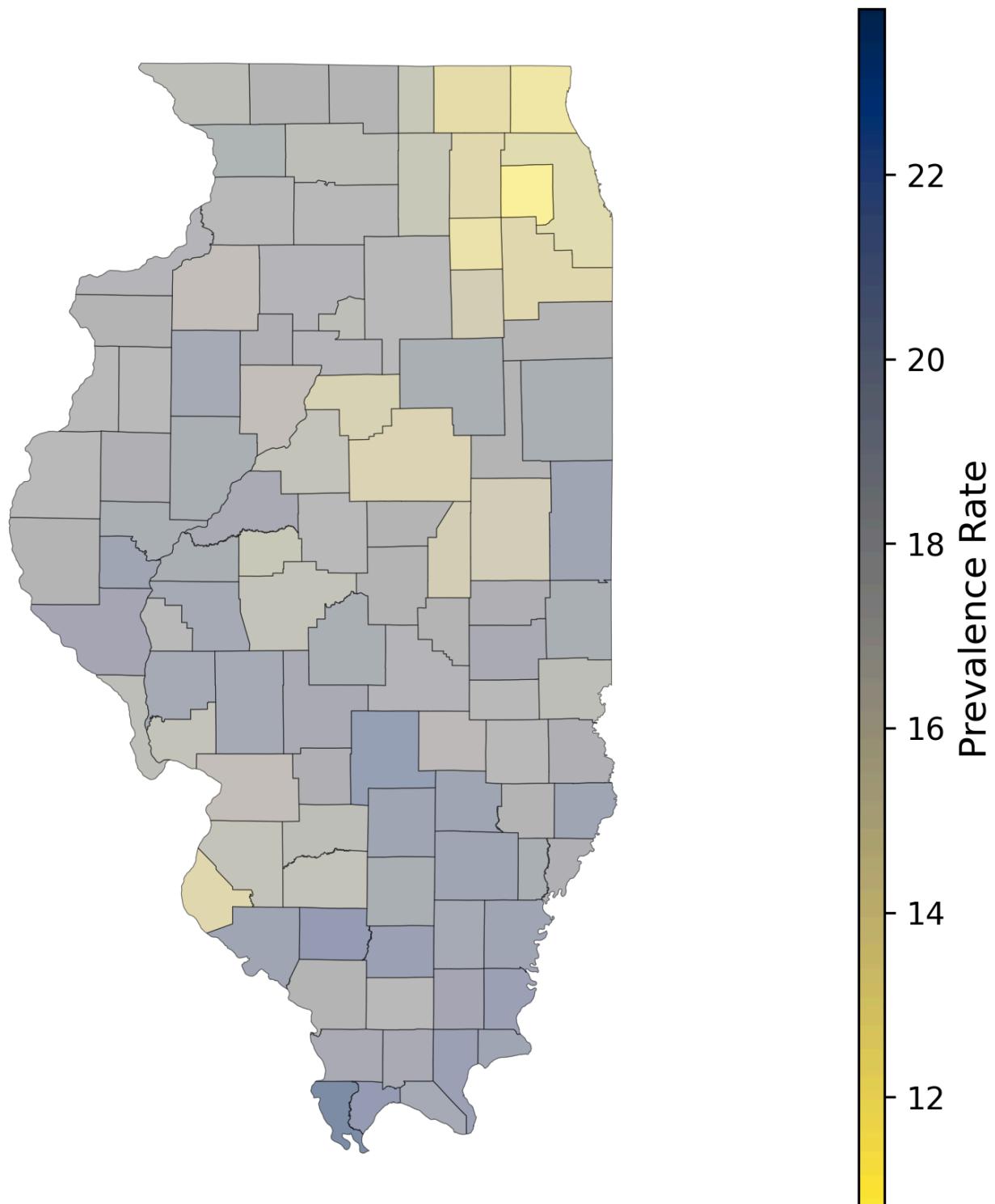
COREM_AdjPrev - Illinois Map by County



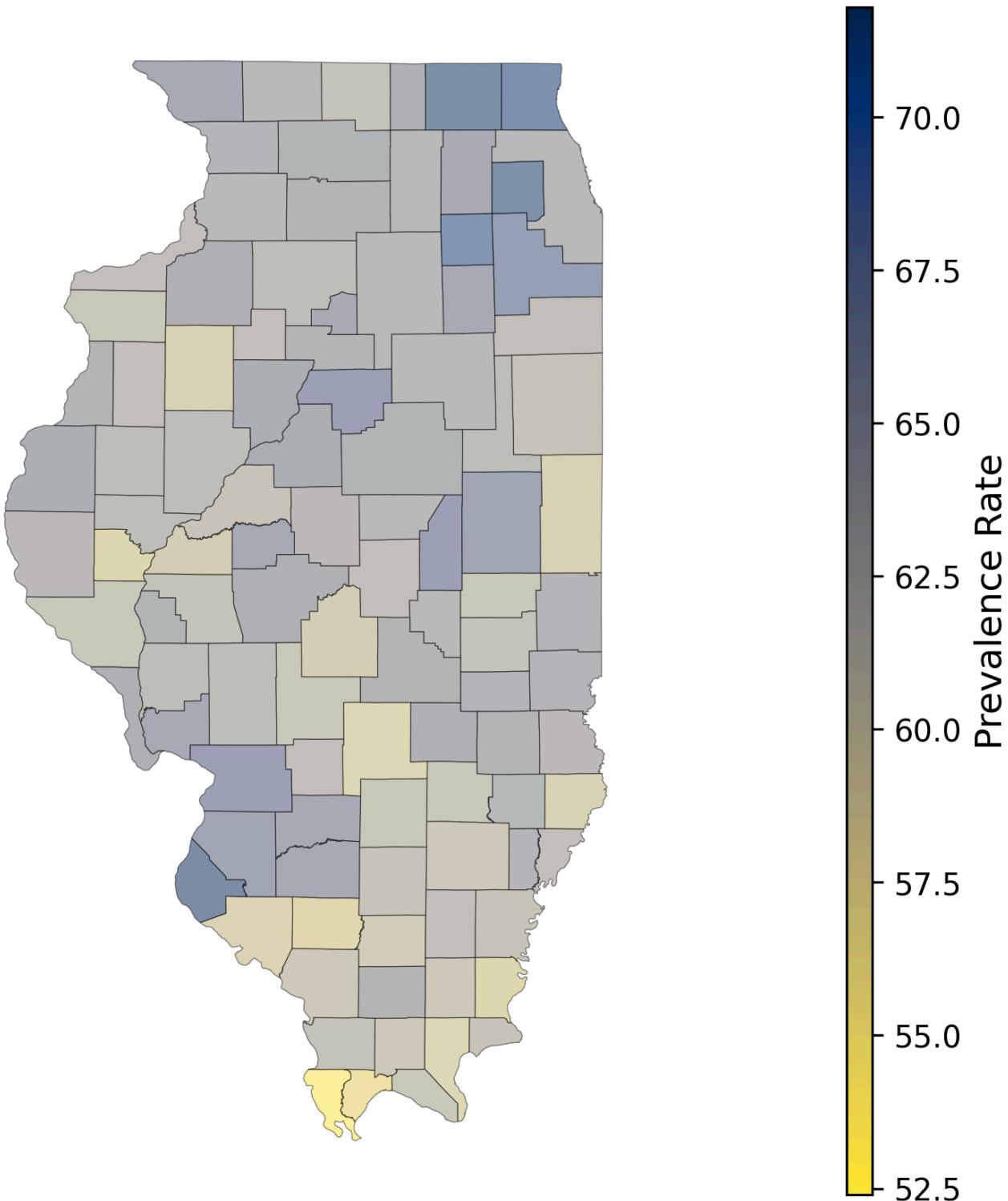
COREW_AdjPrev - Illinois Map by County



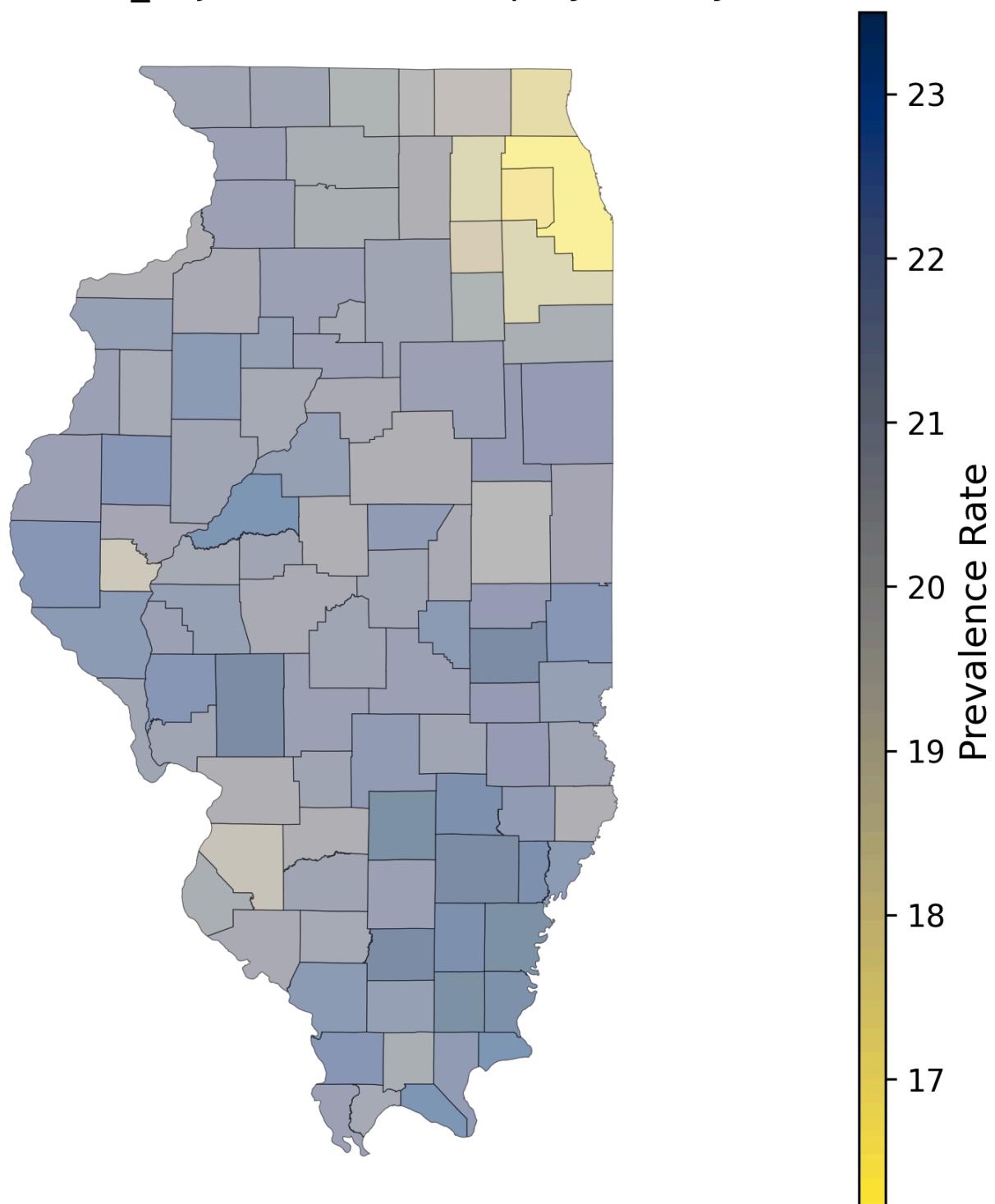
CSMOKING_AdjPrev - Illinois Map by County



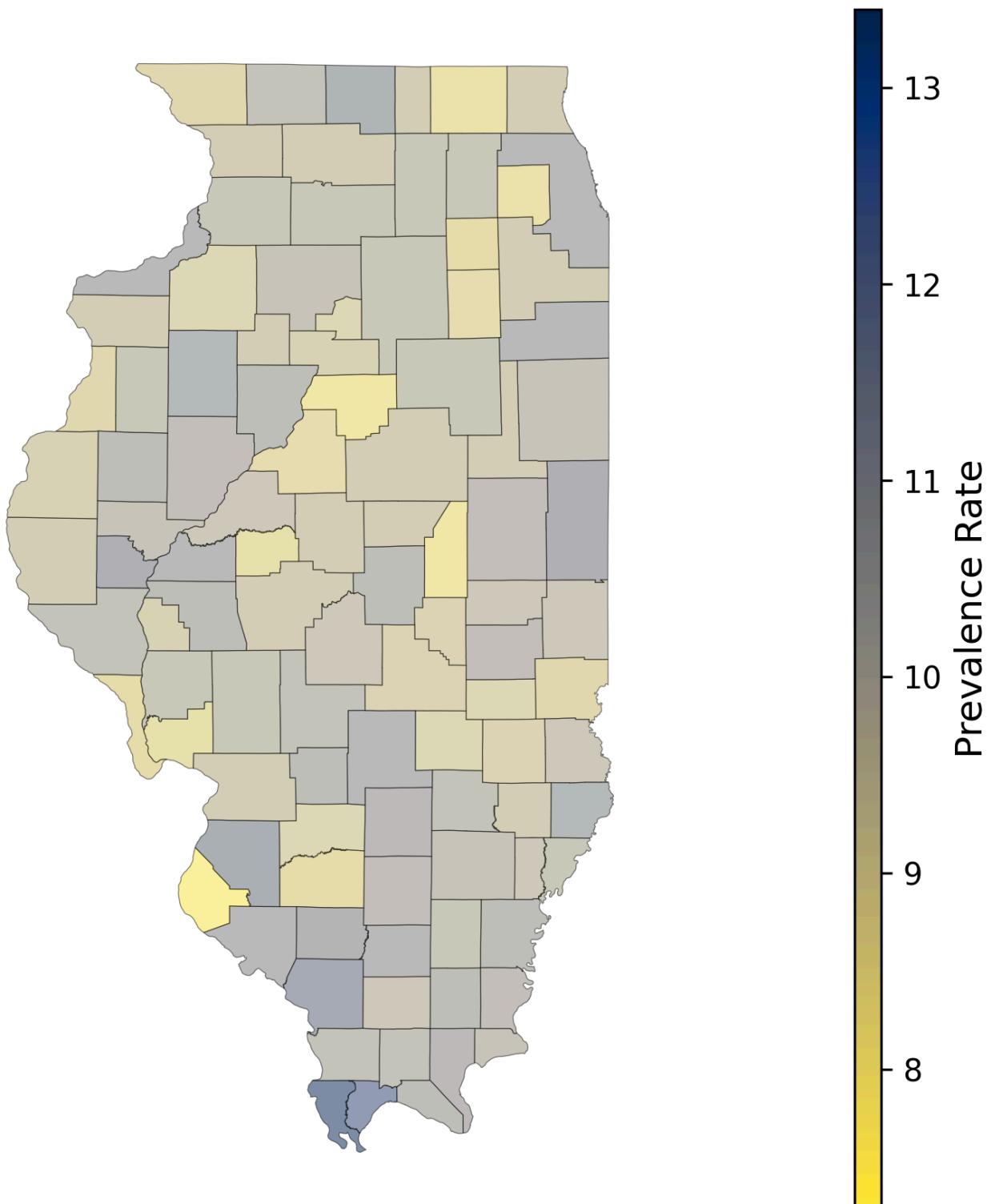
DENTAL_AdjPrev - Illinois Map by County



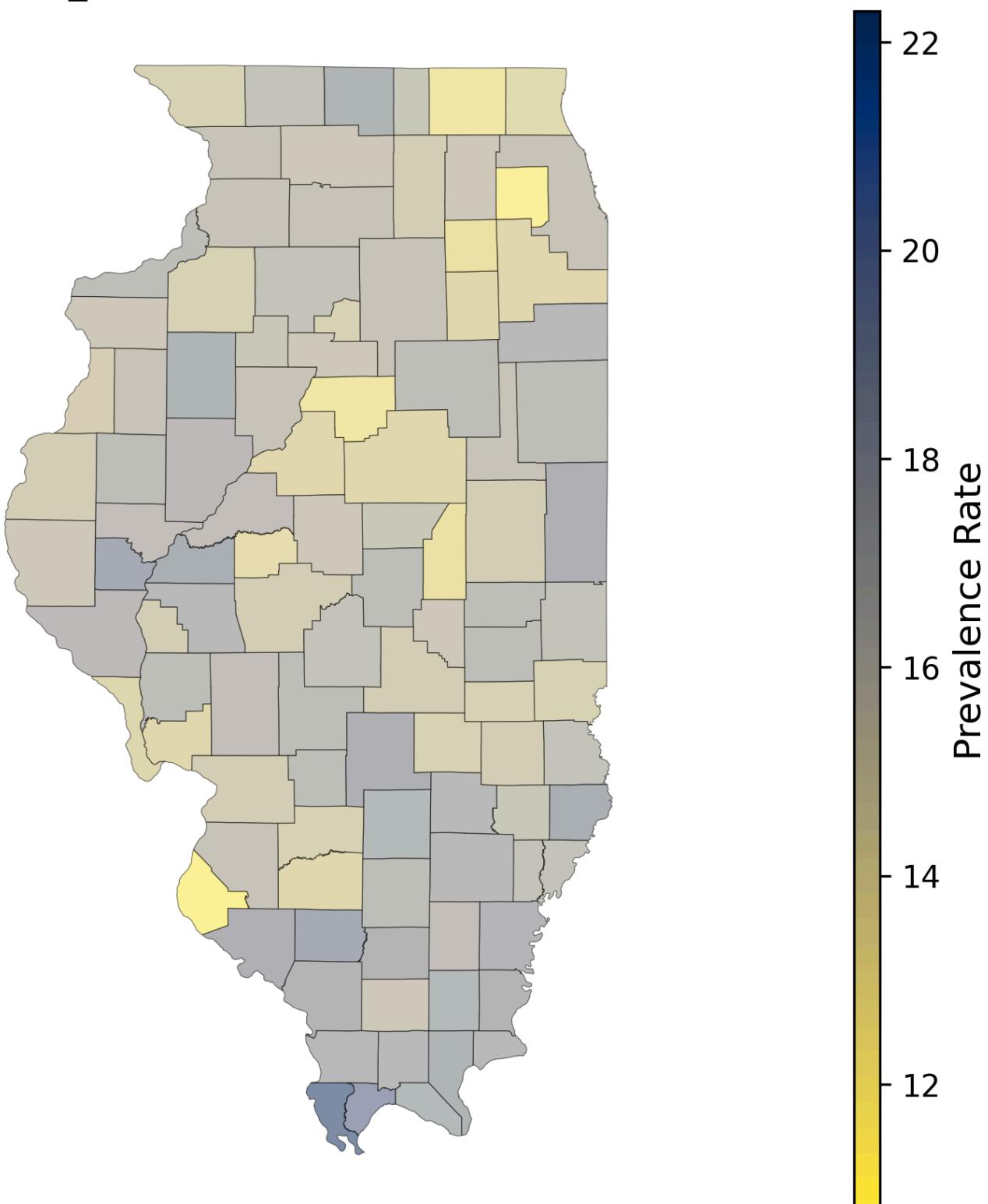
DEPRESSION_AdjPrev - Illinois Map by County



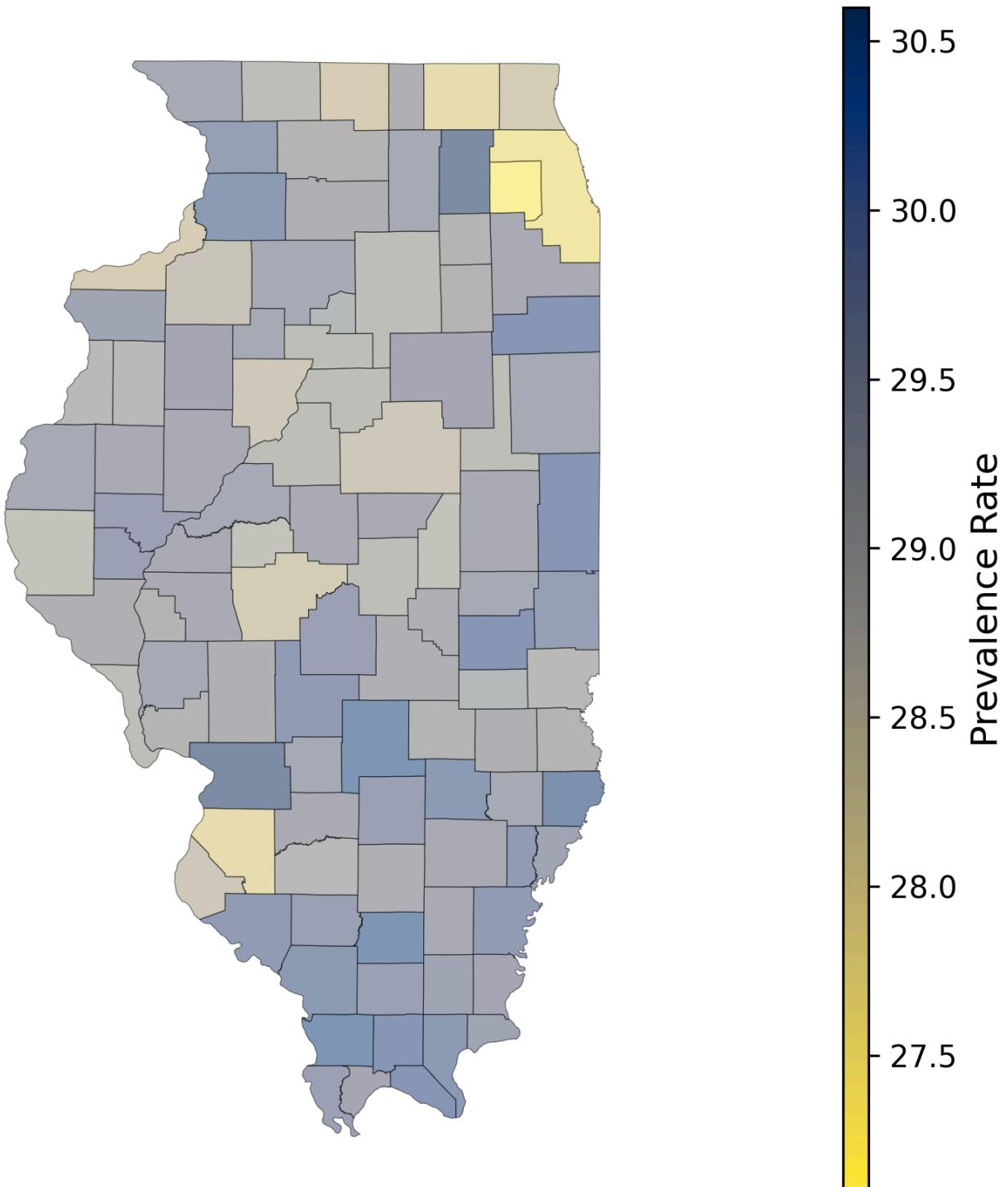
DIABETES_AdjPrev - Illinois Map by County



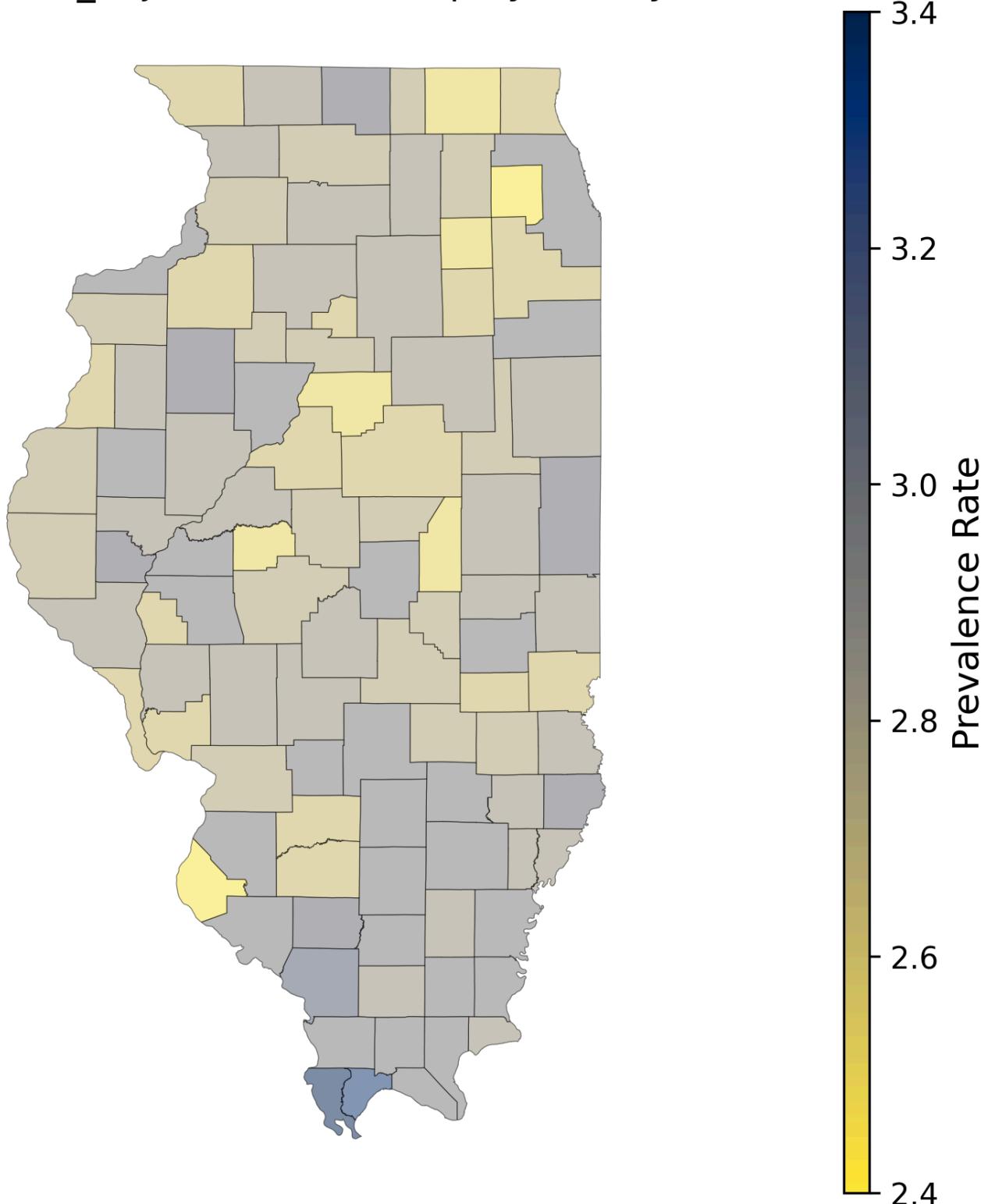
GHLTH_AdjPrev - Illinois Map by County



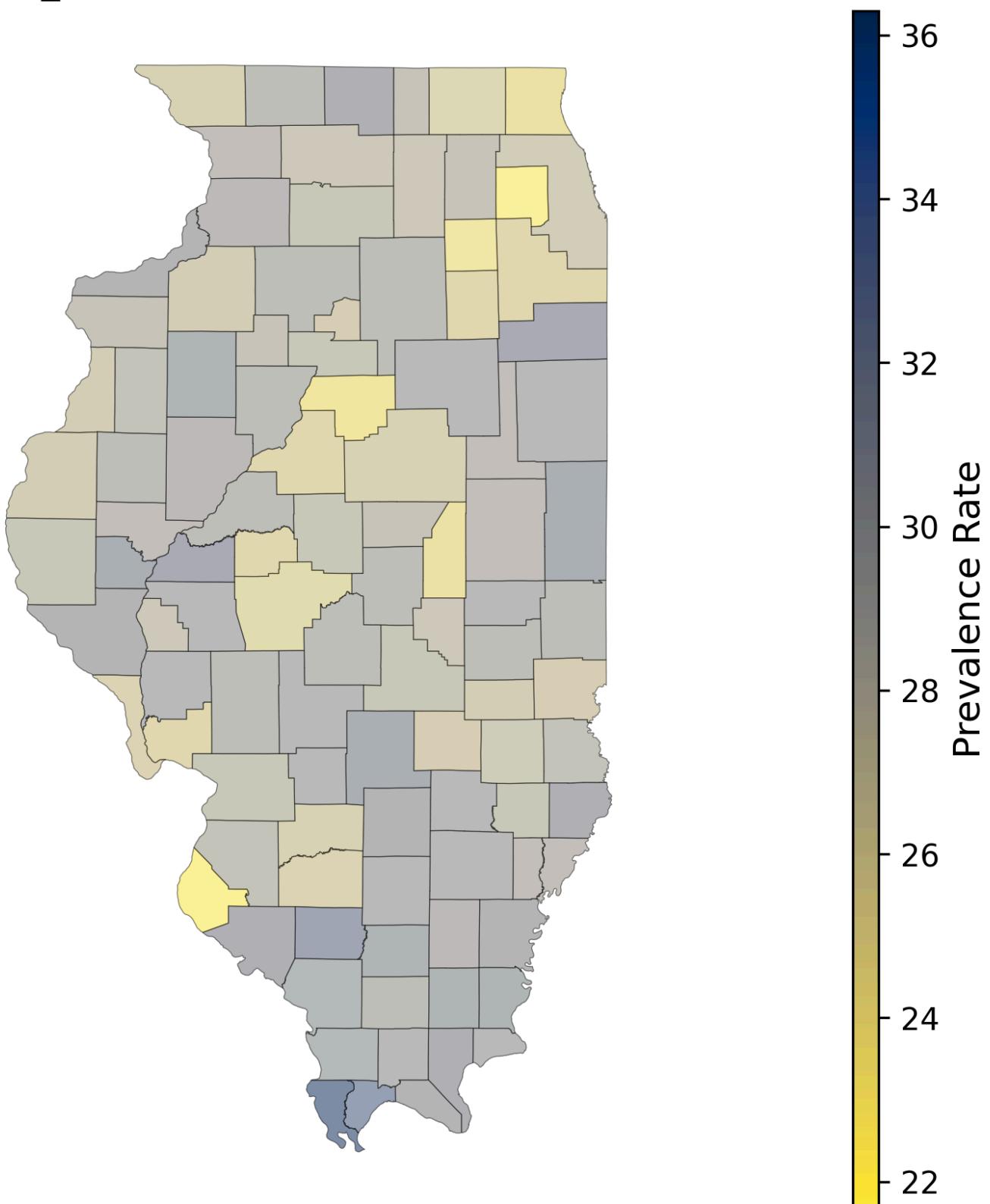
HIGHCHOL_AdjPrev - Illinois Map by County



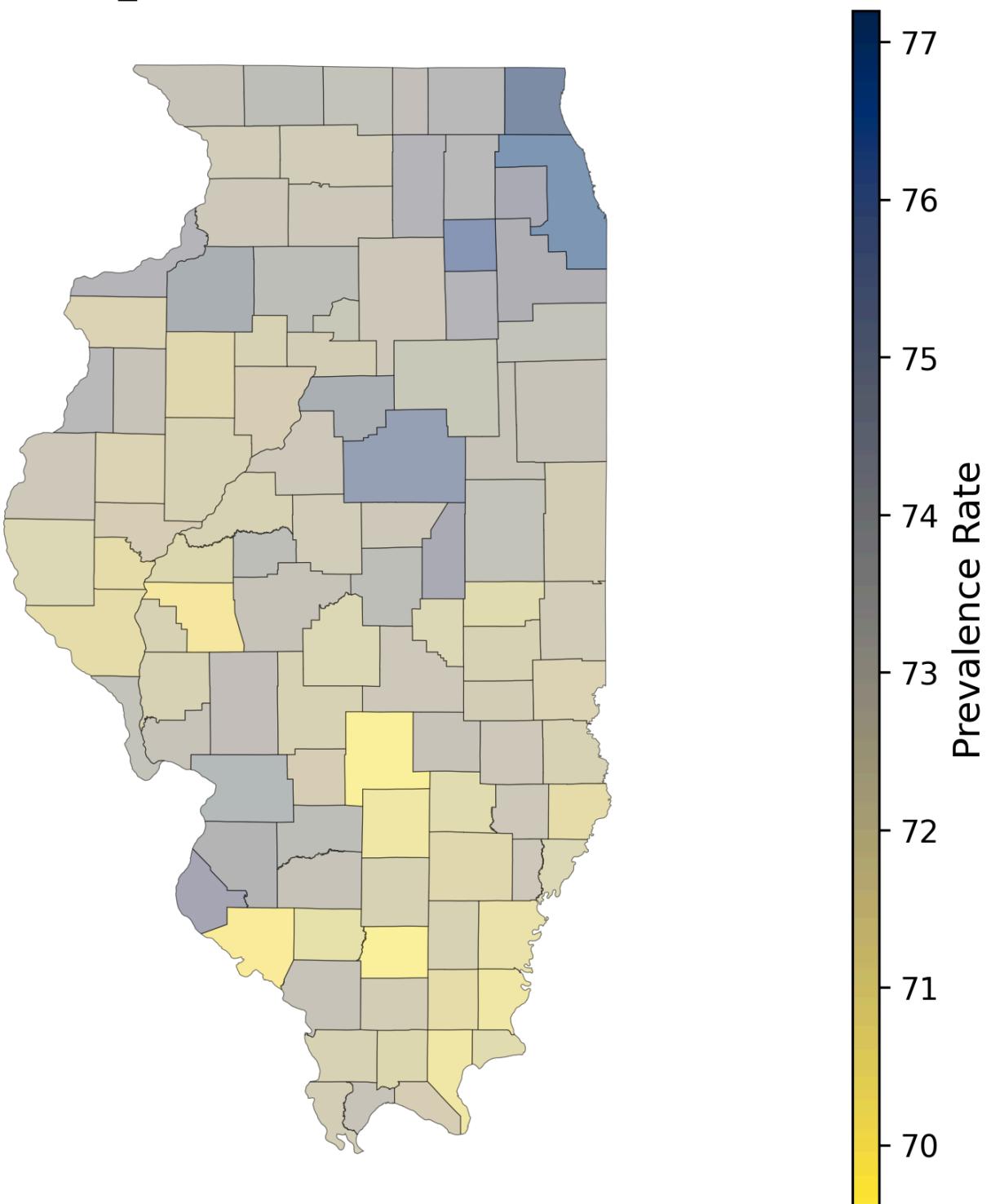
KIDNEY_AdjPrev - Illinois Map by County



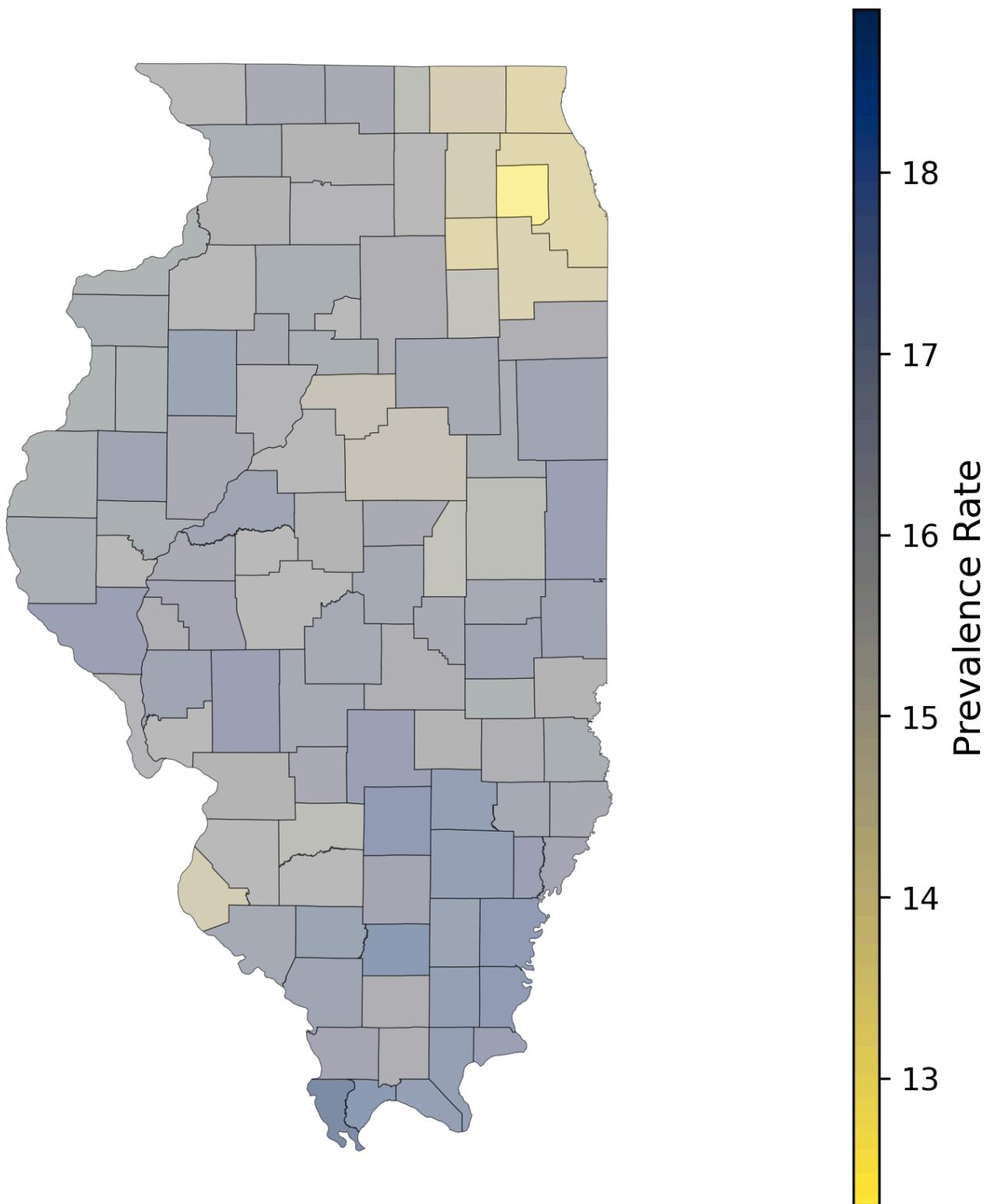
LPA_AdjPrev - Illinois Map by County



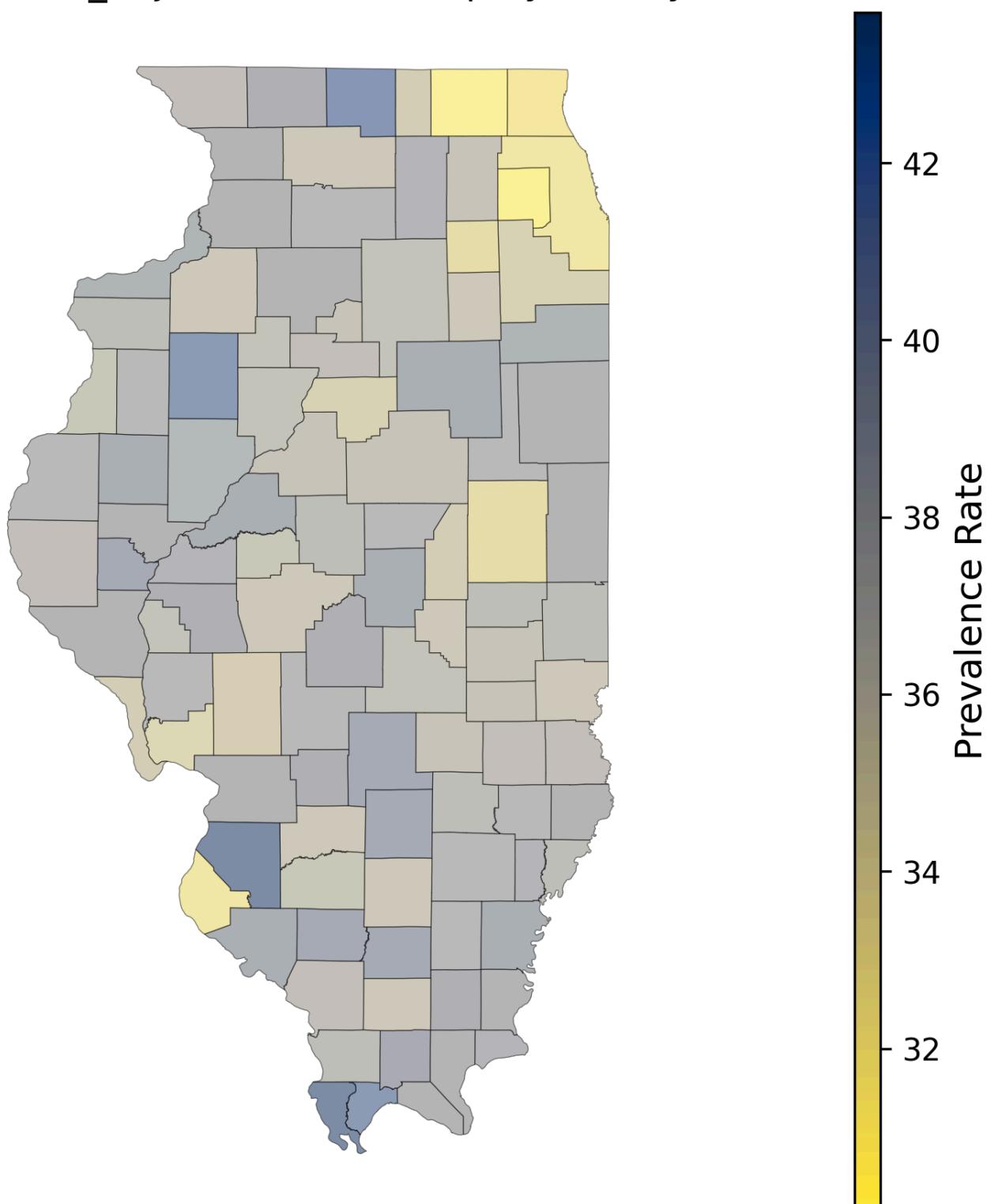
MAMMOUSE_AdjPrev - Illinois Map by County



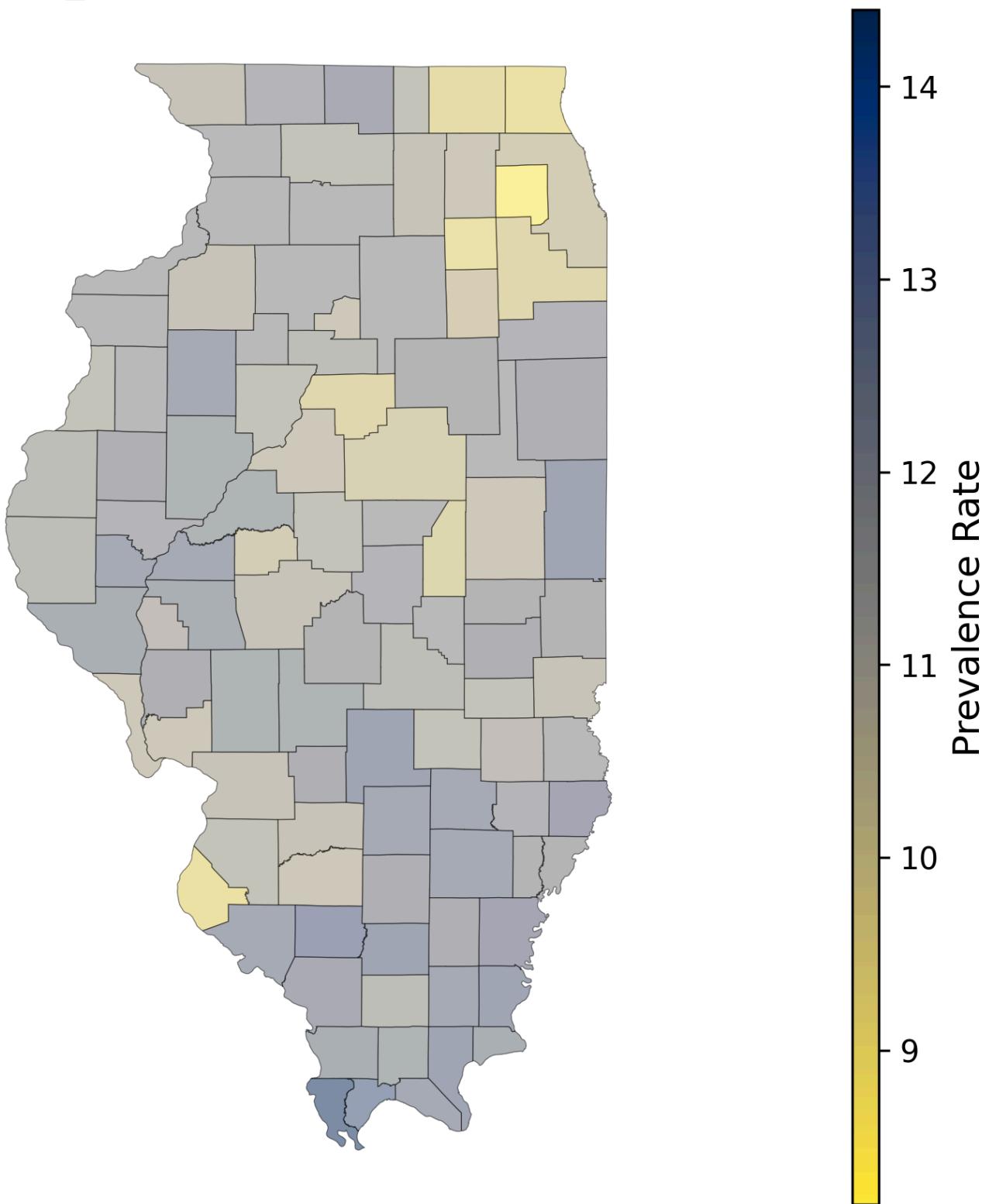
MHLTH_AdjPrev - Illinois Map by County



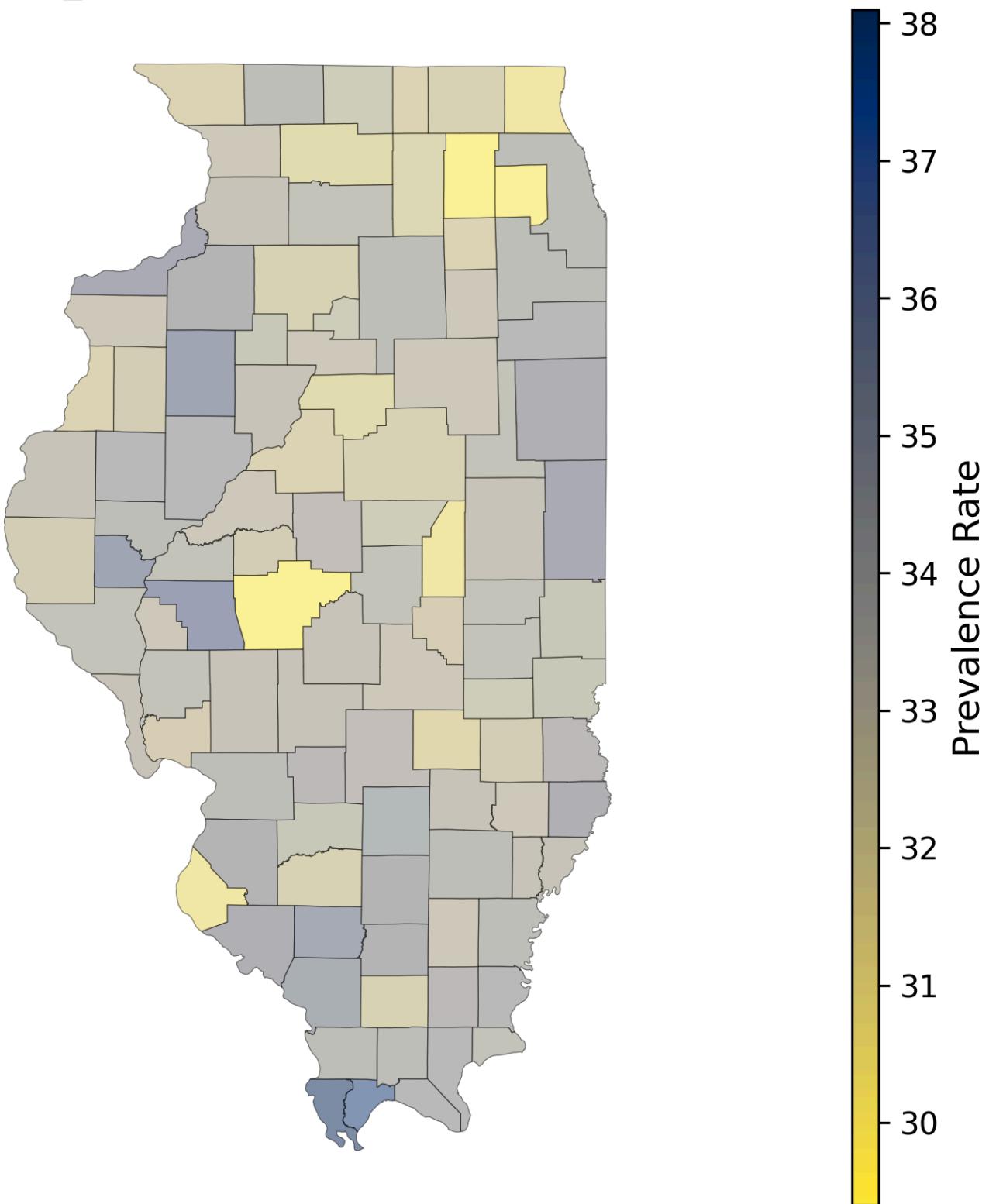
OBESITY_AdjPrev - Illinois Map by County



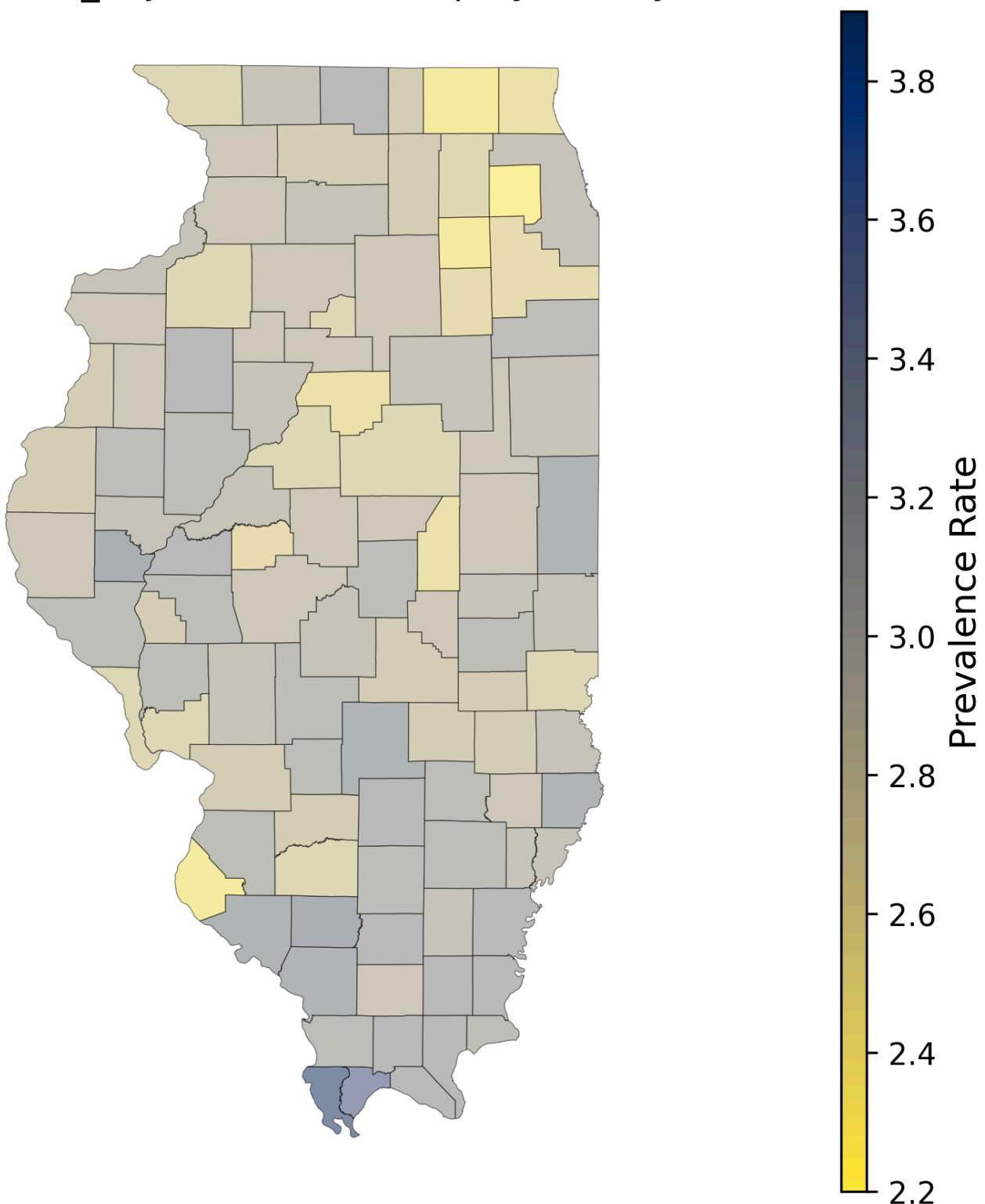
PHLTH_AdjPrev - Illinois Map by County



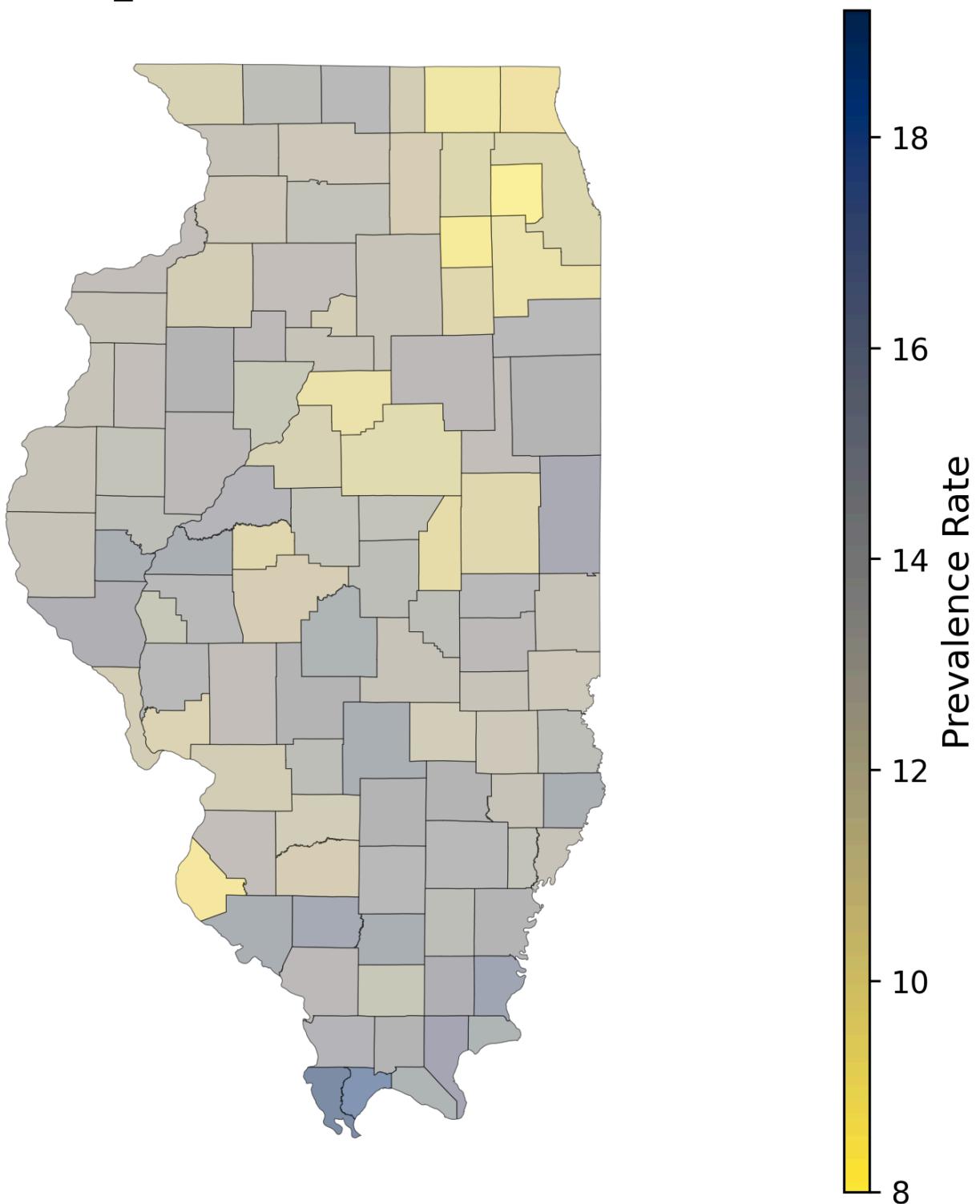
SLEEP_AdjPrev - Illinois Map by County



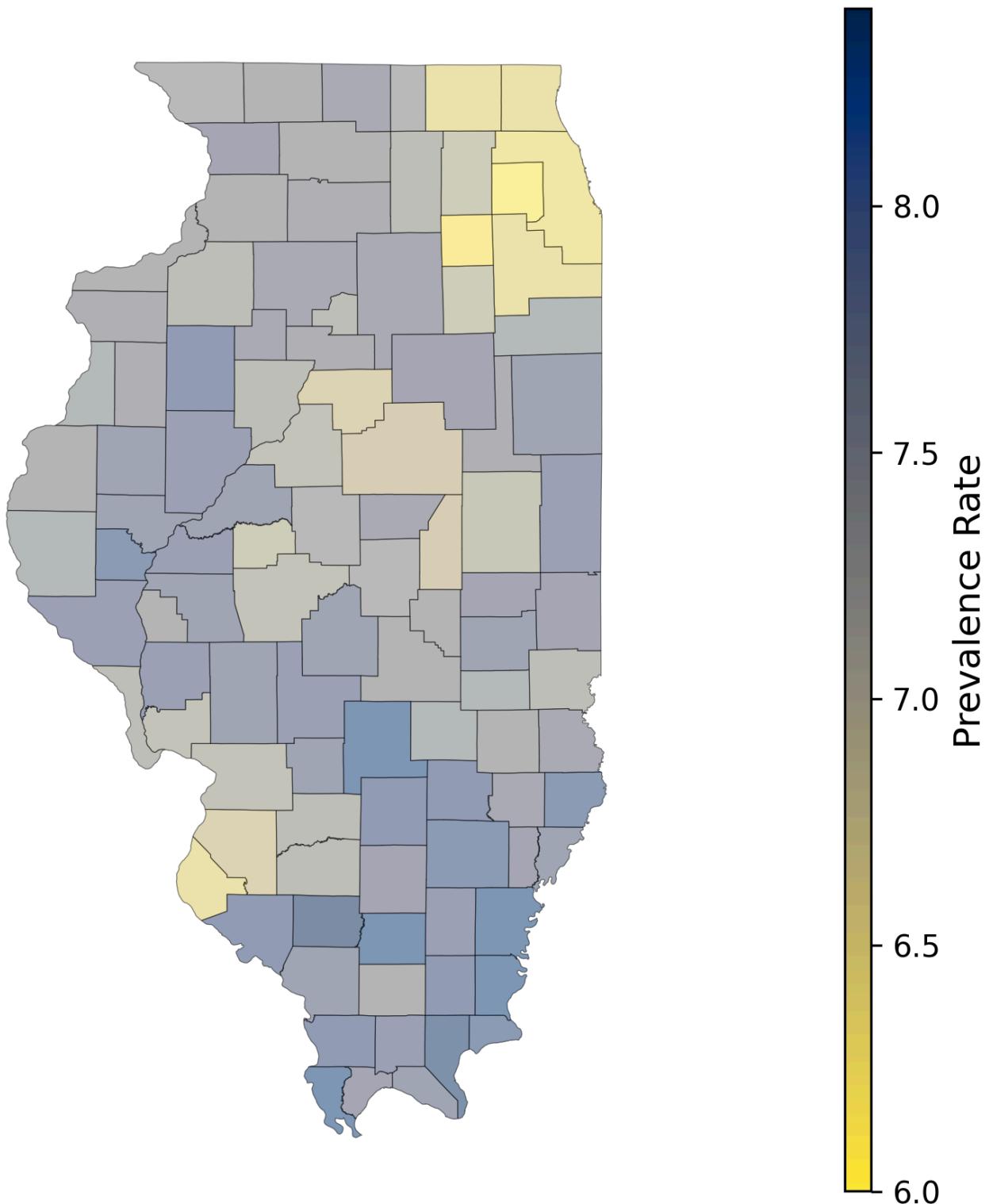
STROKE_AdjPrev - Illinois Map by County



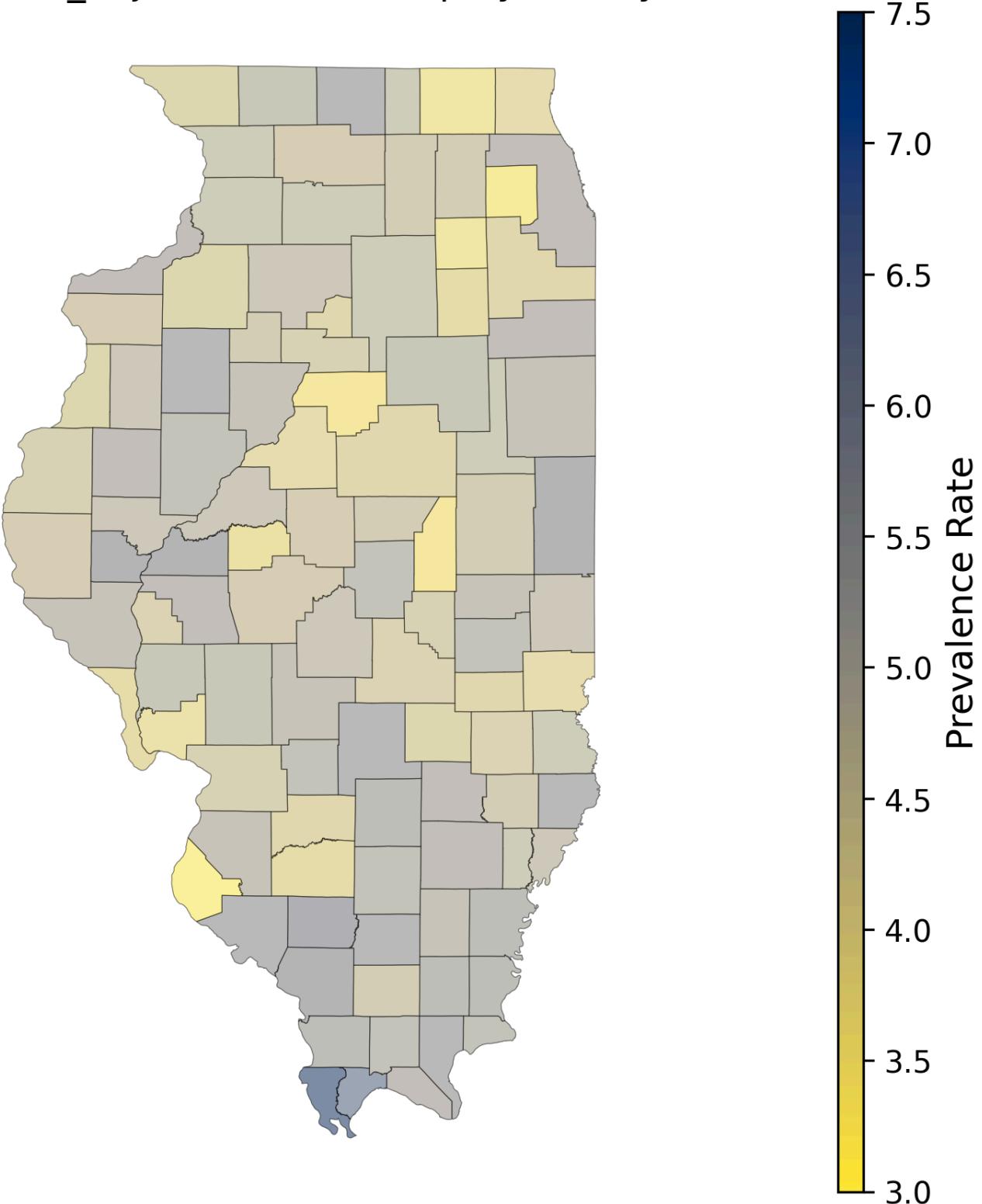
TEETHLOST_AdjPrev - Illinois Map by County



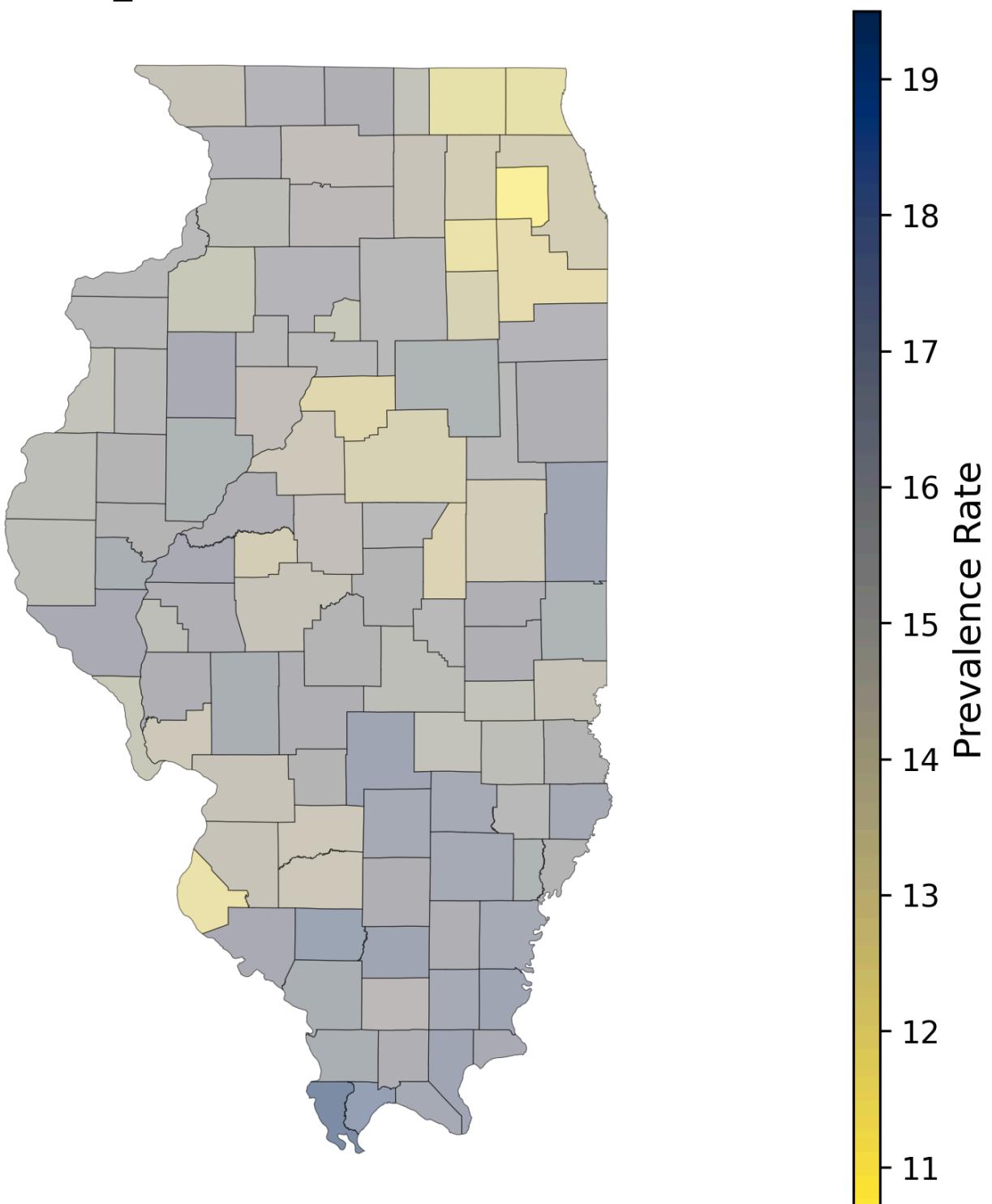
HEARING_AdjPrev - Illinois Map by County



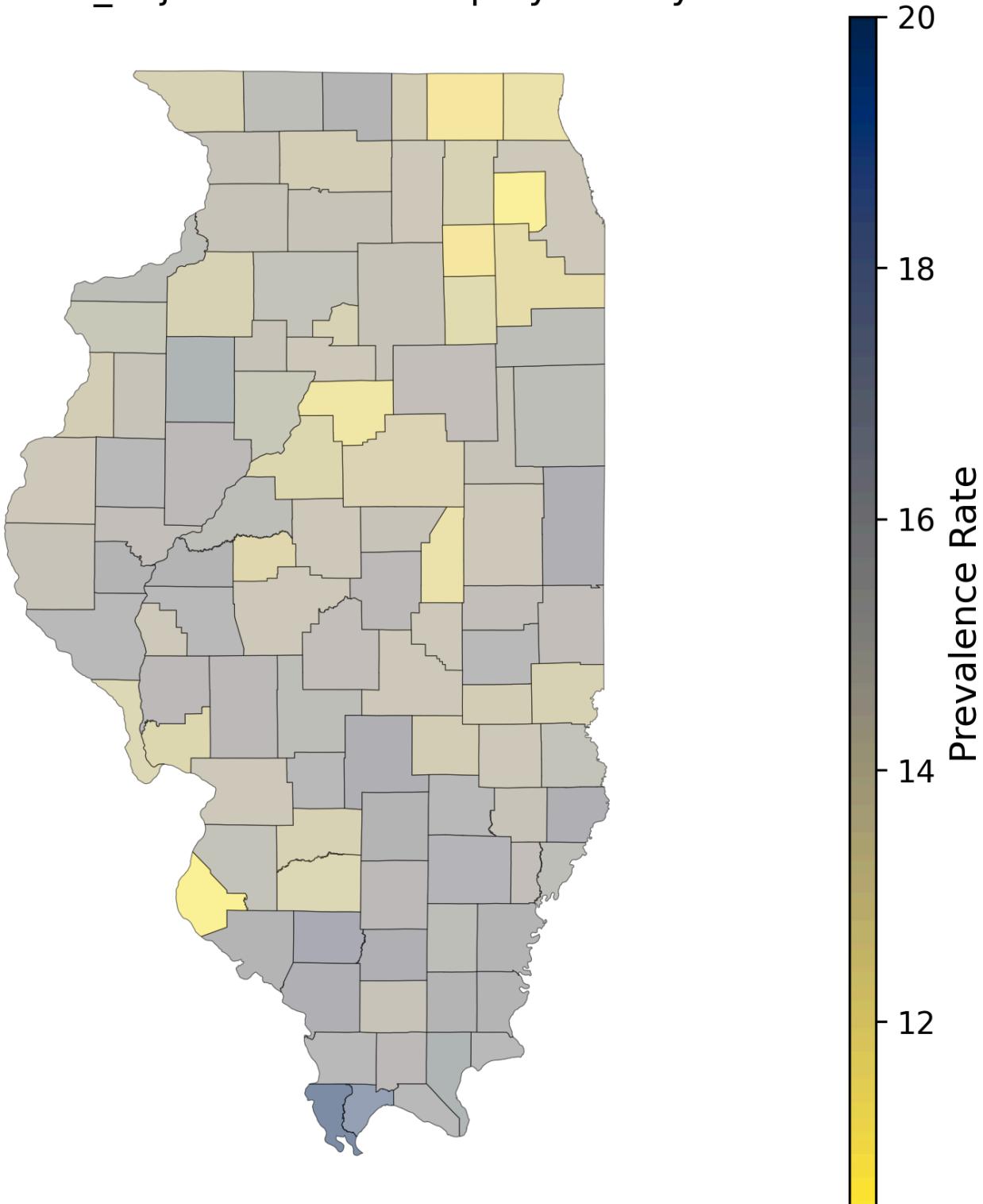
VISION_AdjPrev - Illinois Map by County



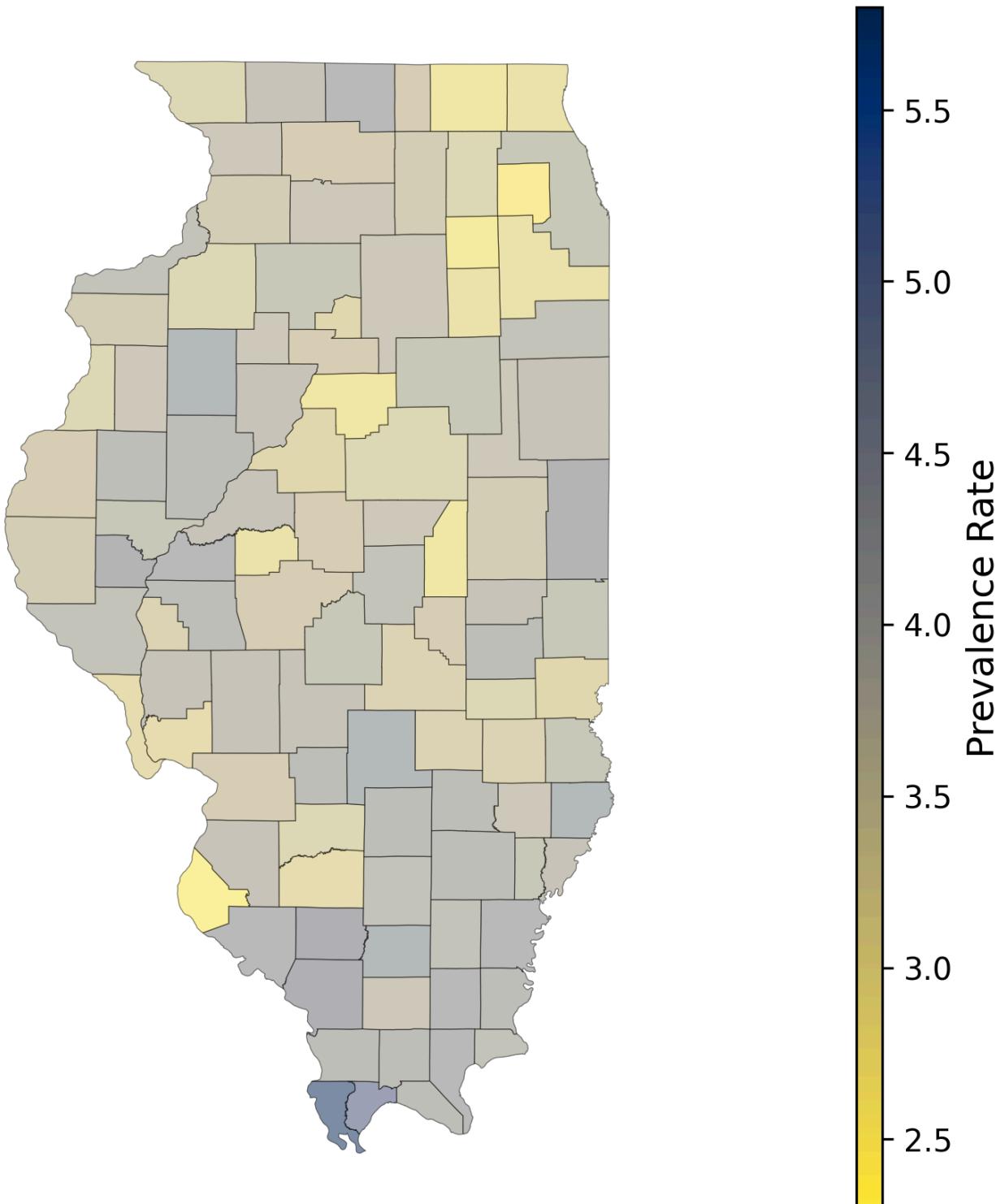
COGNITION_AdjPrev - Illinois Map by County



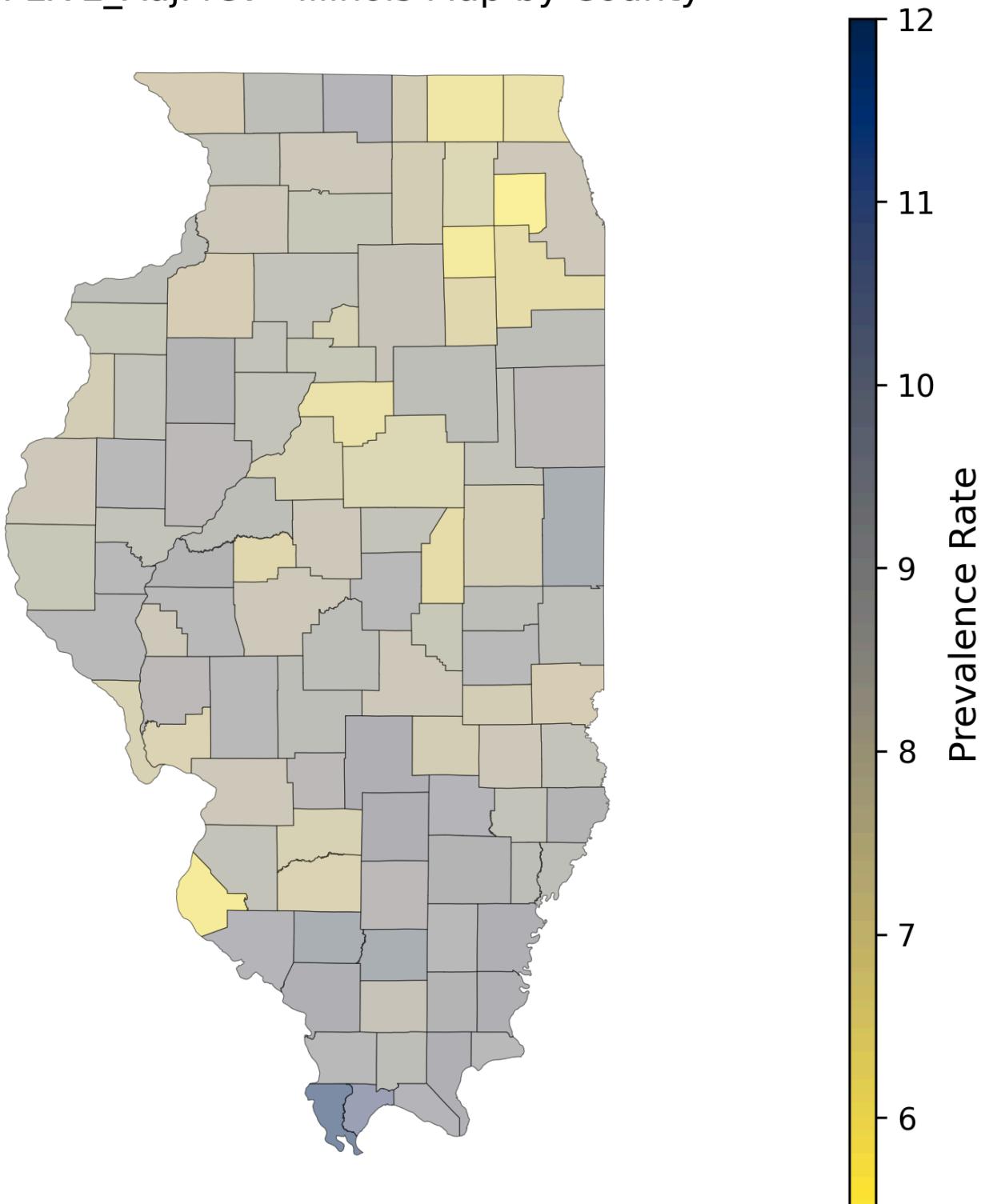
MOBILITY_AdjPrev - Illinois Map by County



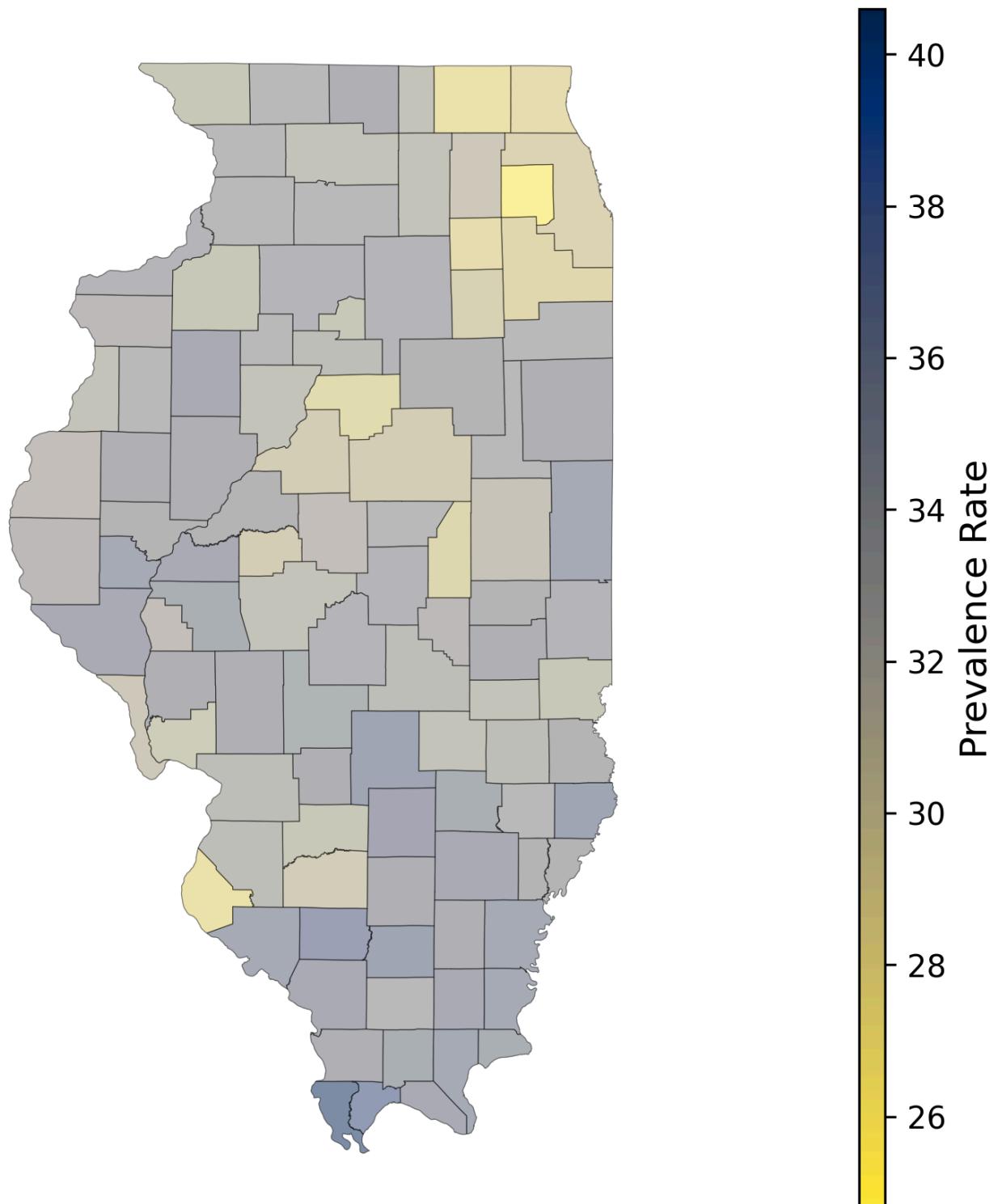
SELCARE_AdjPrev - Illinois Map by County



INDEPLIVE_AdjPrev - Illinois Map by County

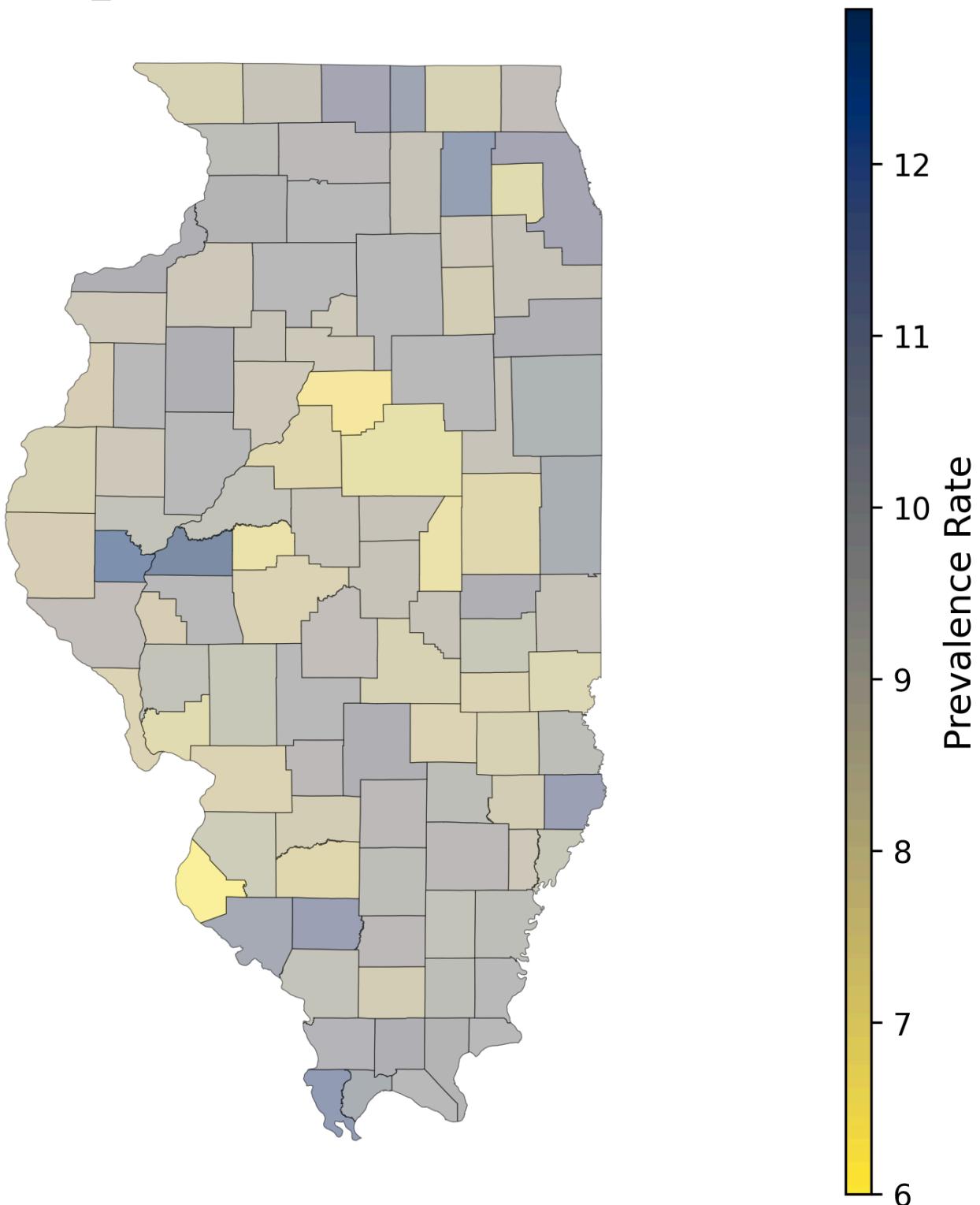


DISABILITY_AdjPrev - Illinois Map by County

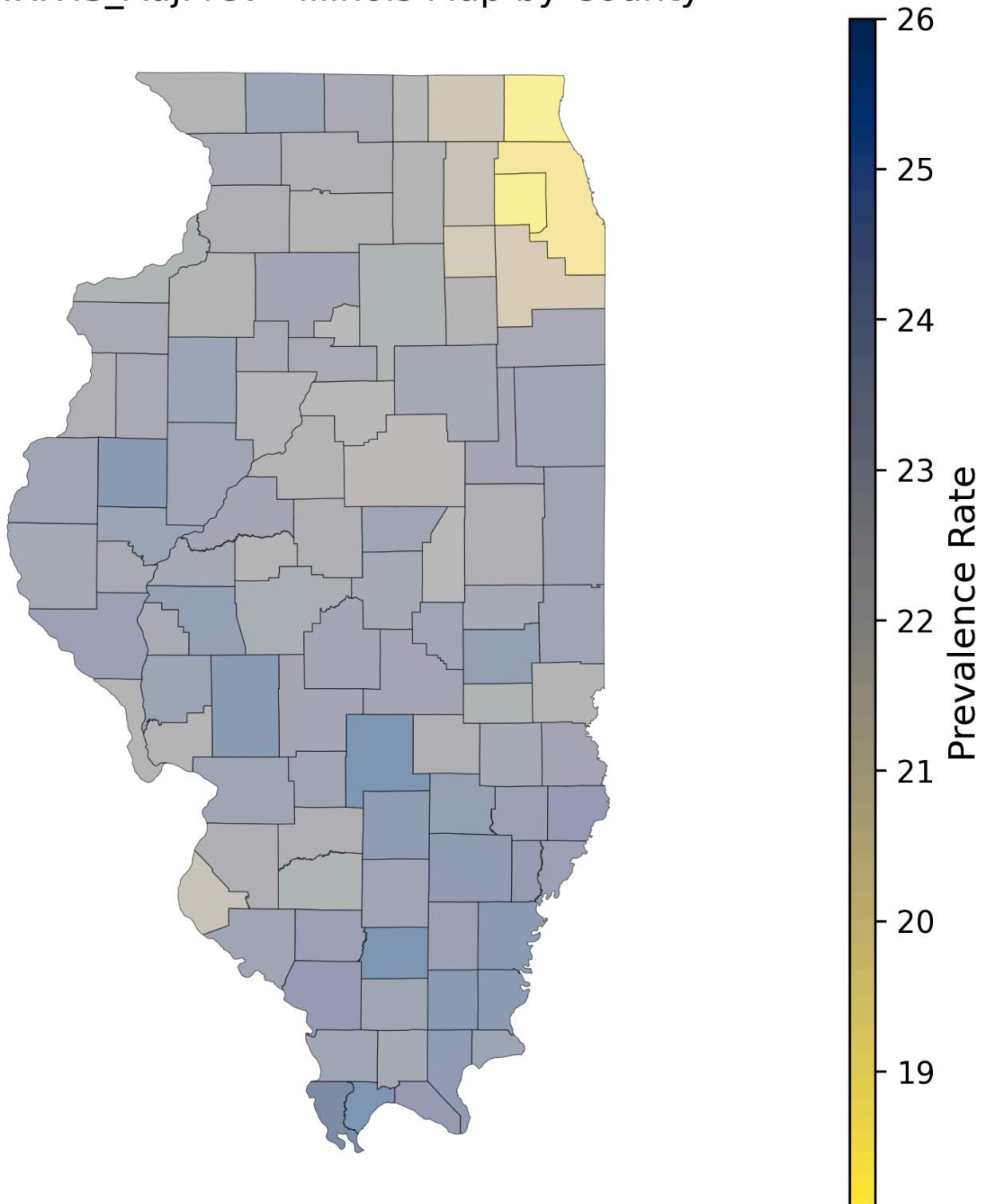


```
In [ ]: # Get columns ending with '_AdjPrev'  
columns_to_plot = [col for col in df.columns if col.endswith('_AdjPrev')]  
  
plot_normalized_county_data(merged_gdf, columns_to_plot, 'cividis_r')
```

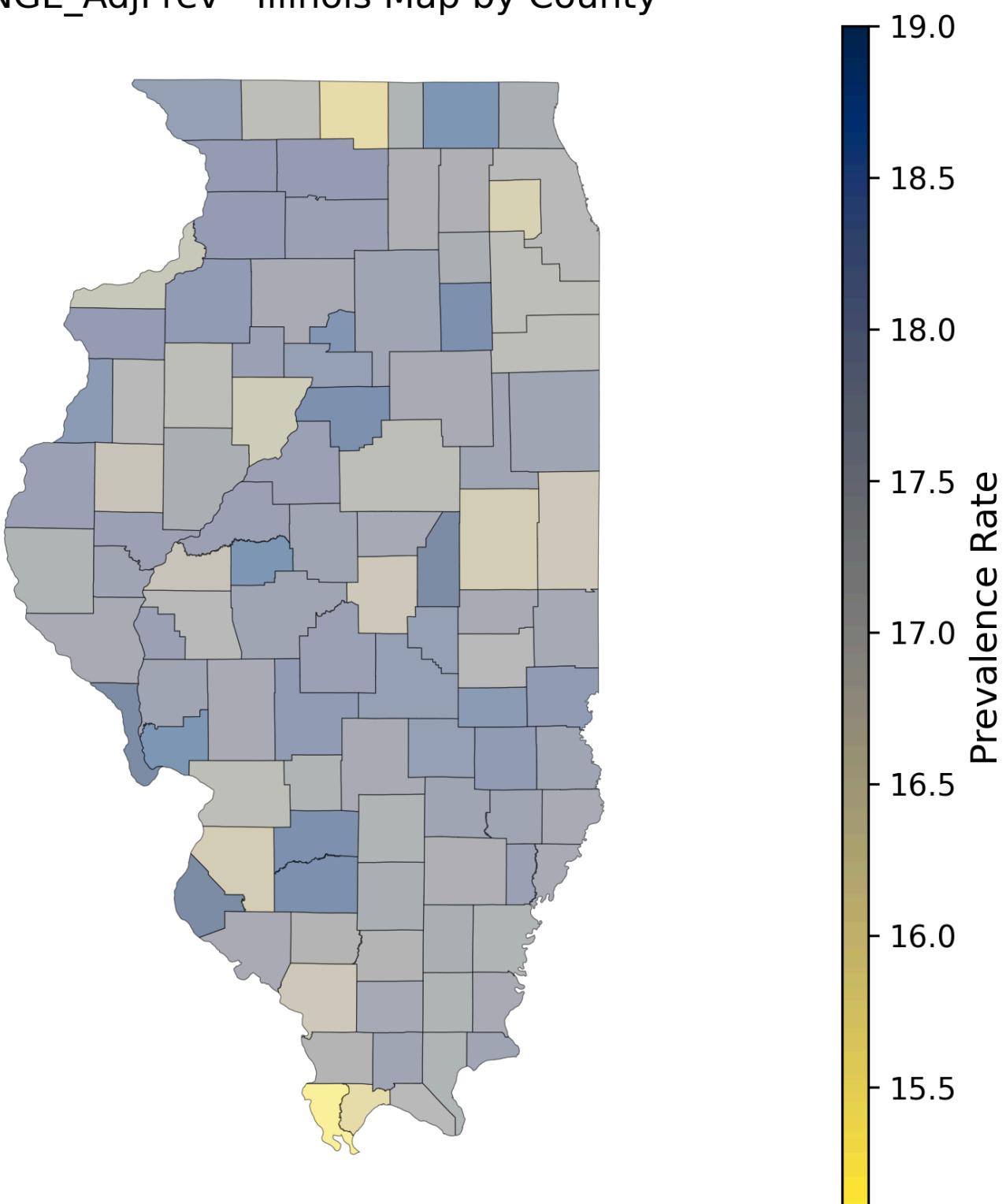
ACCESS2_AdjPrev - Illinois Map by County



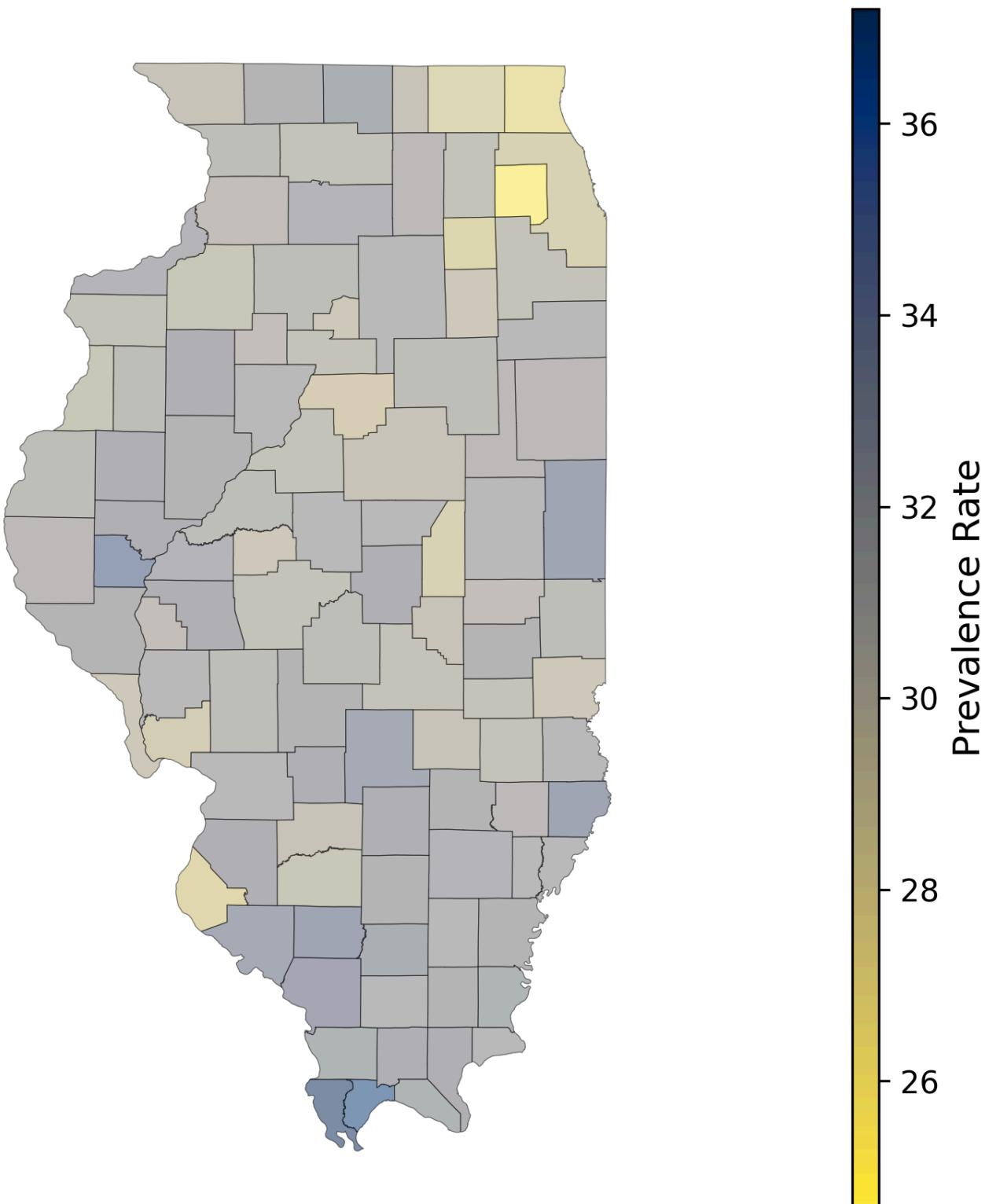
ARTHRITIS_AdjPrev - Illinois Map by County



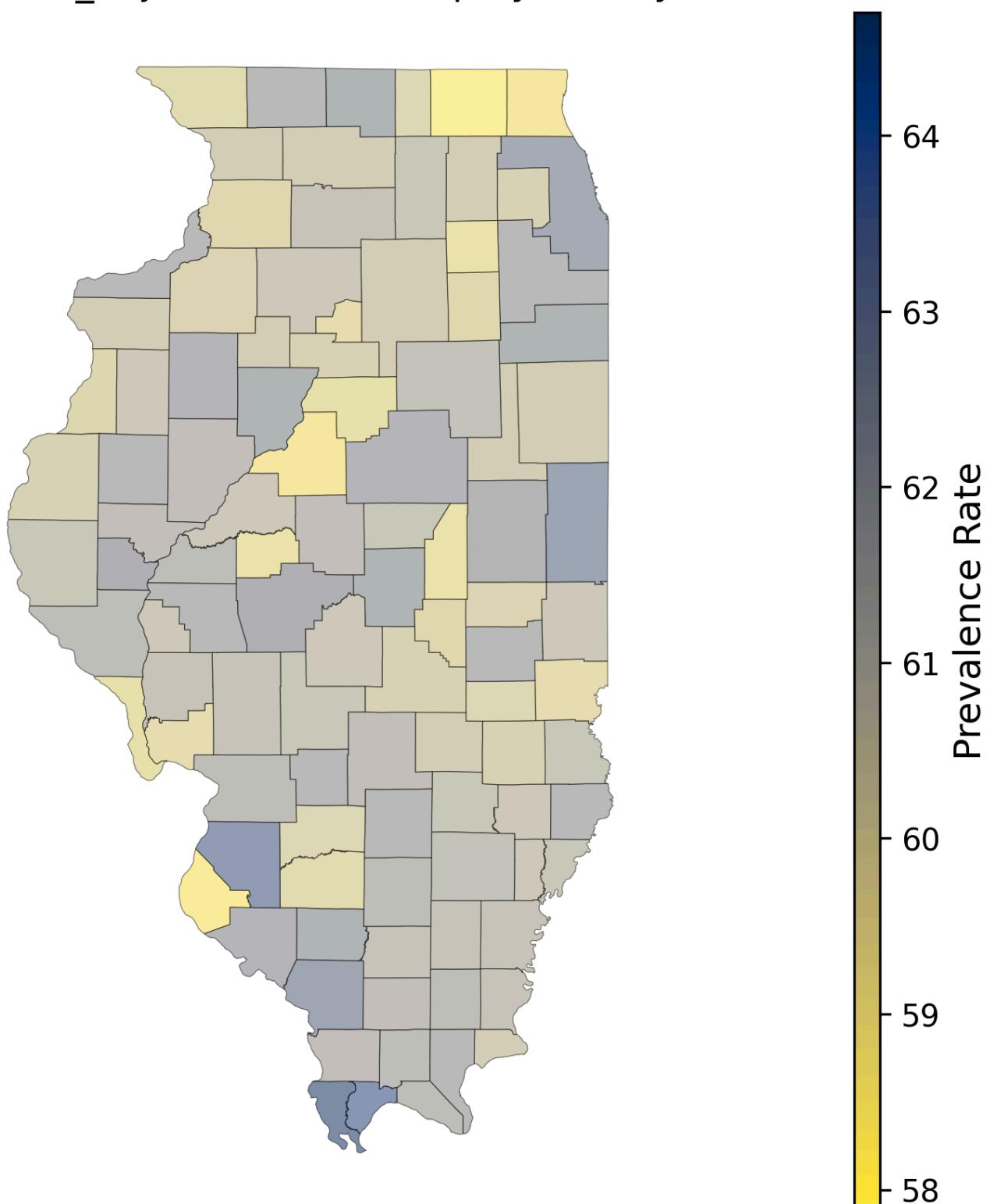
BINGE_AdjPrev - Illinois Map by County



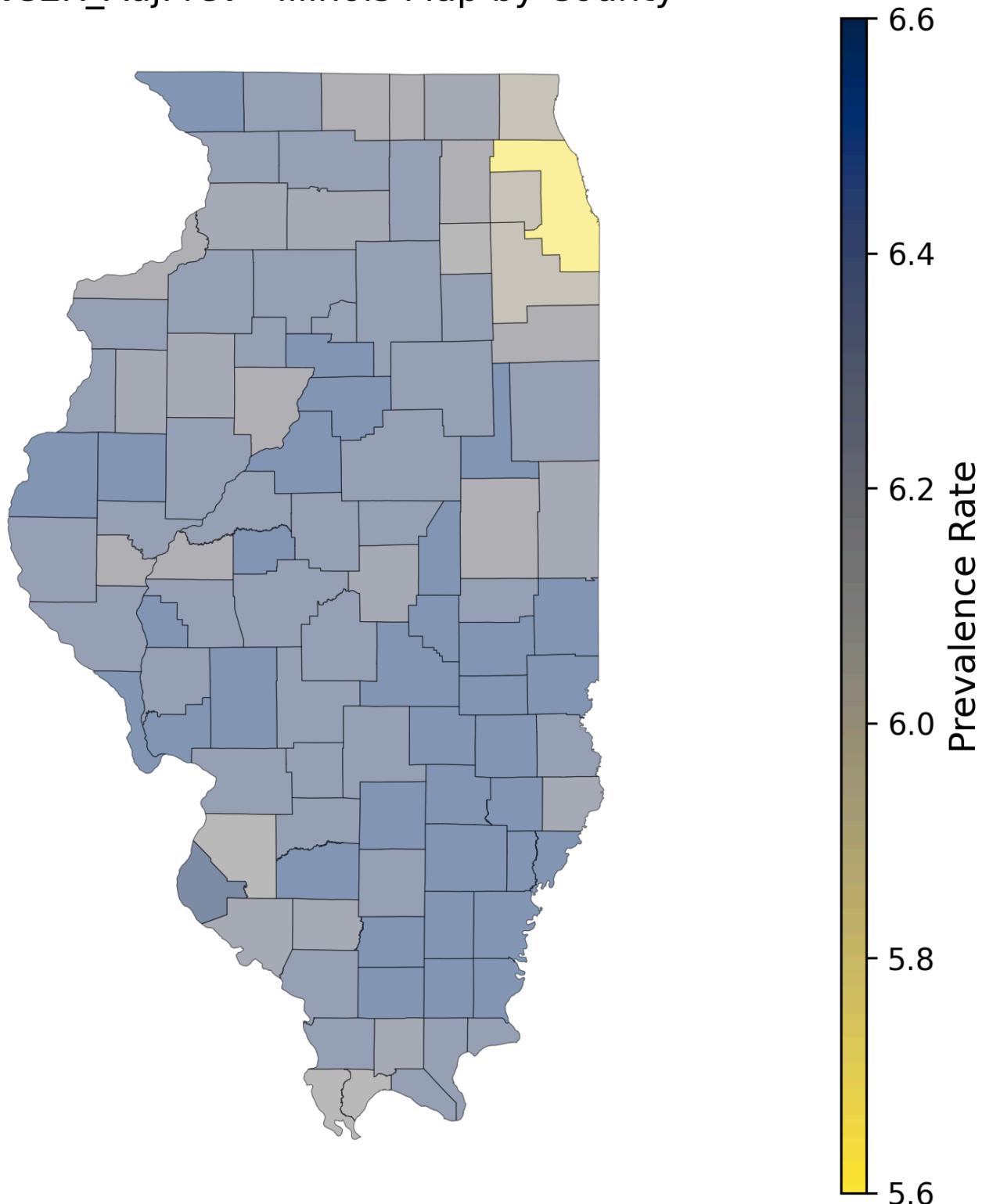
BPHIGH_AdjPrev - Illinois Map by County



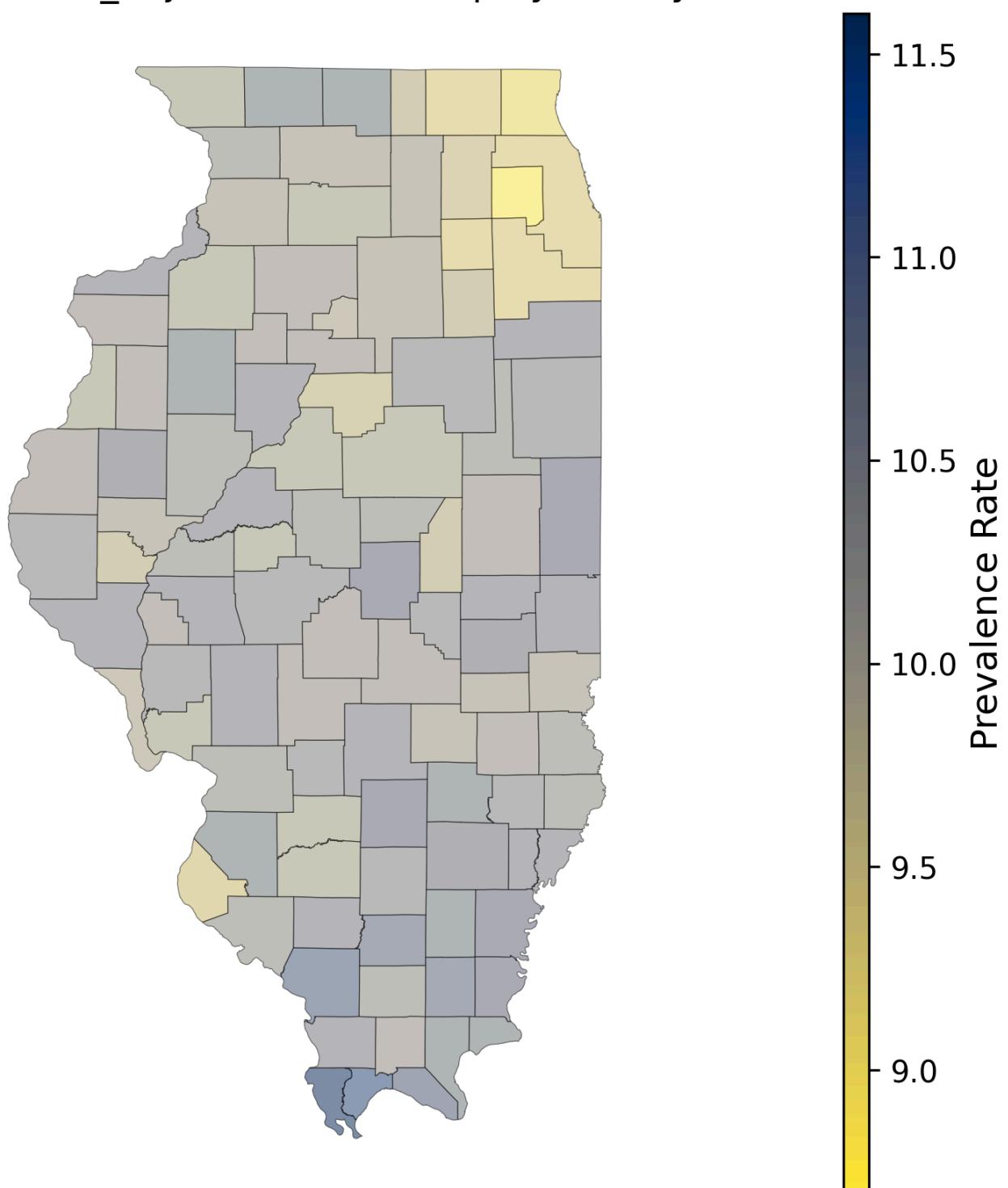
BPMED_AdjPrev - Illinois Map by County



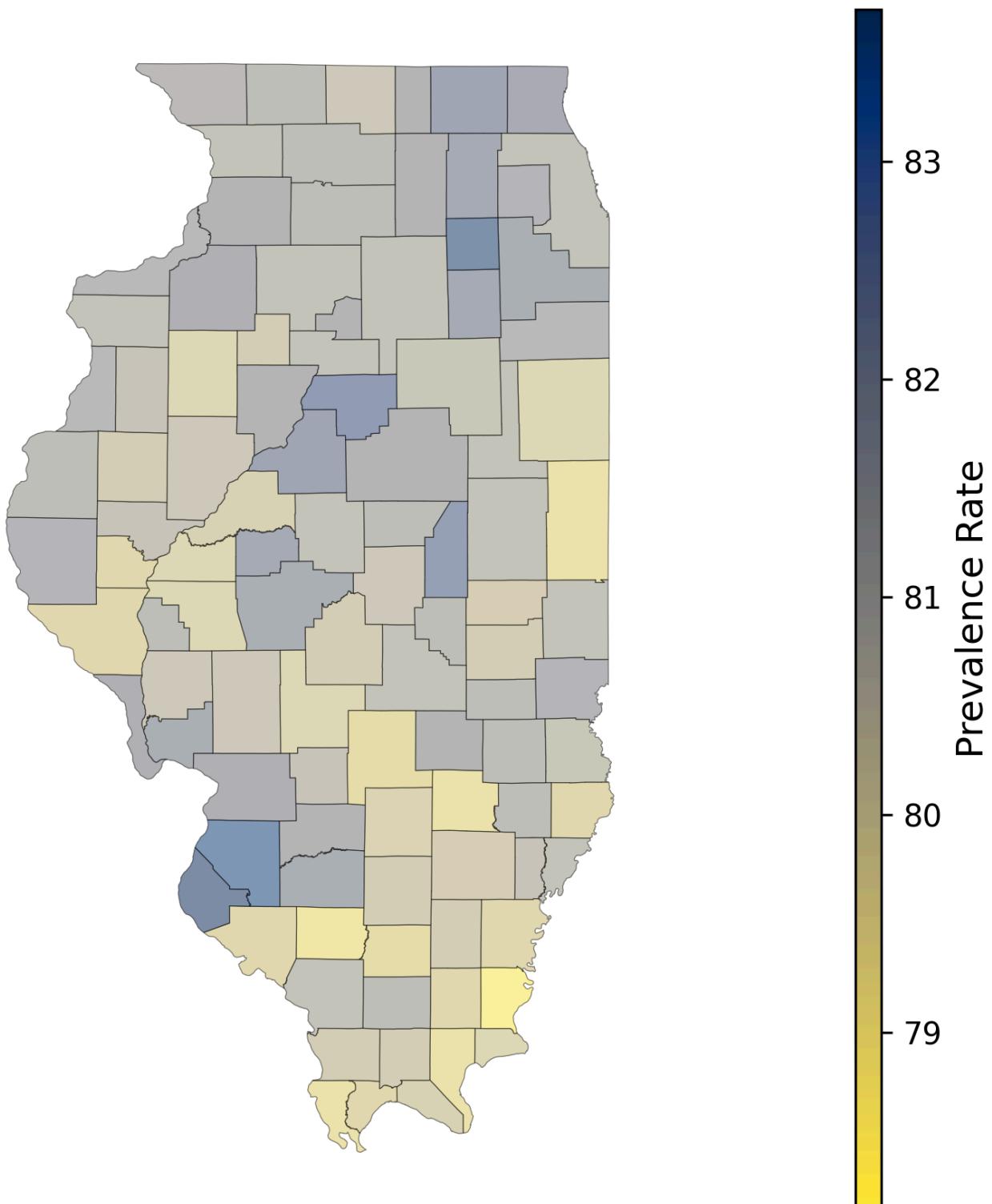
CANCER_AdjPrev - Illinois Map by County



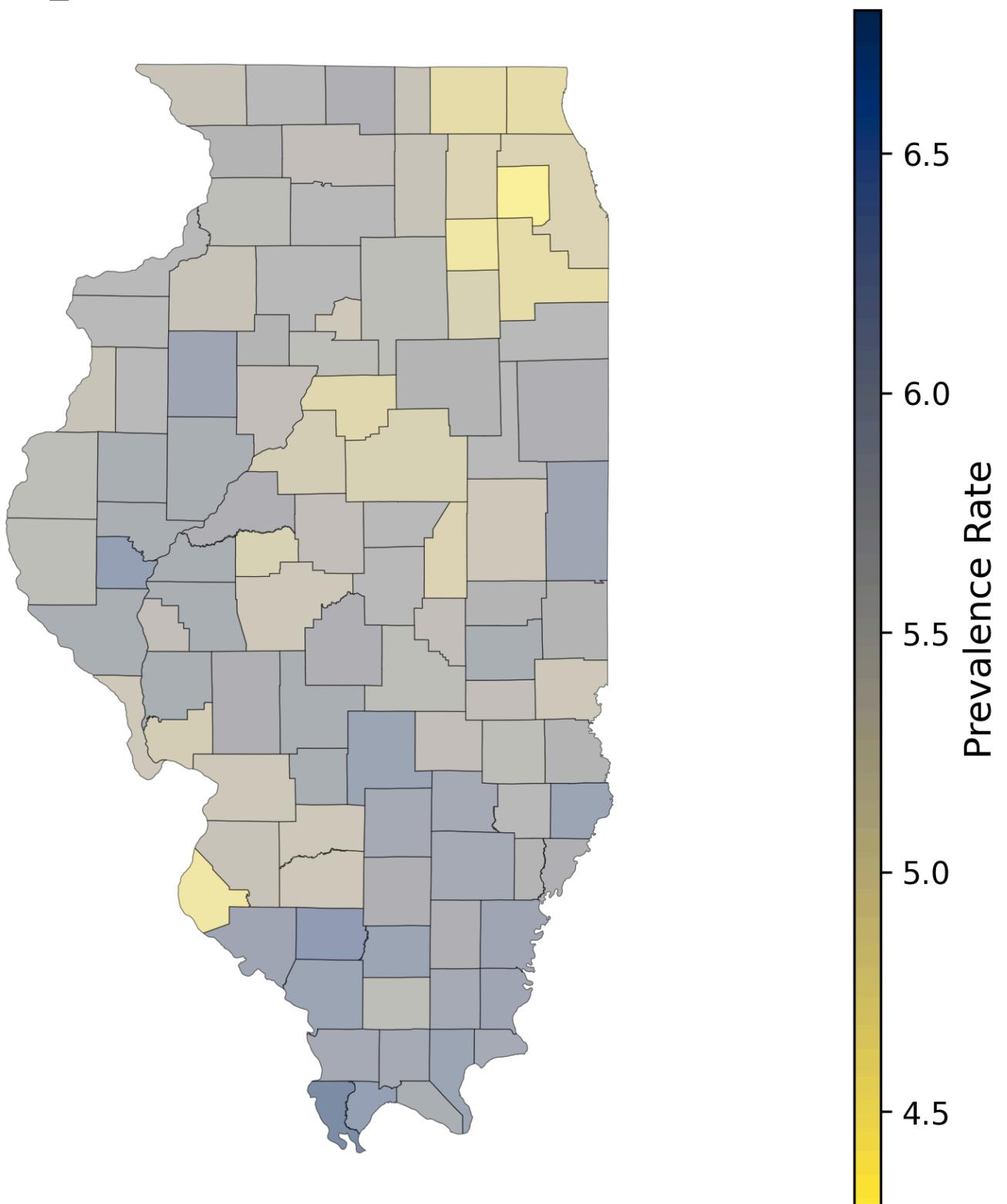
CASTHMA_AdjPrev - Illinois Map by County



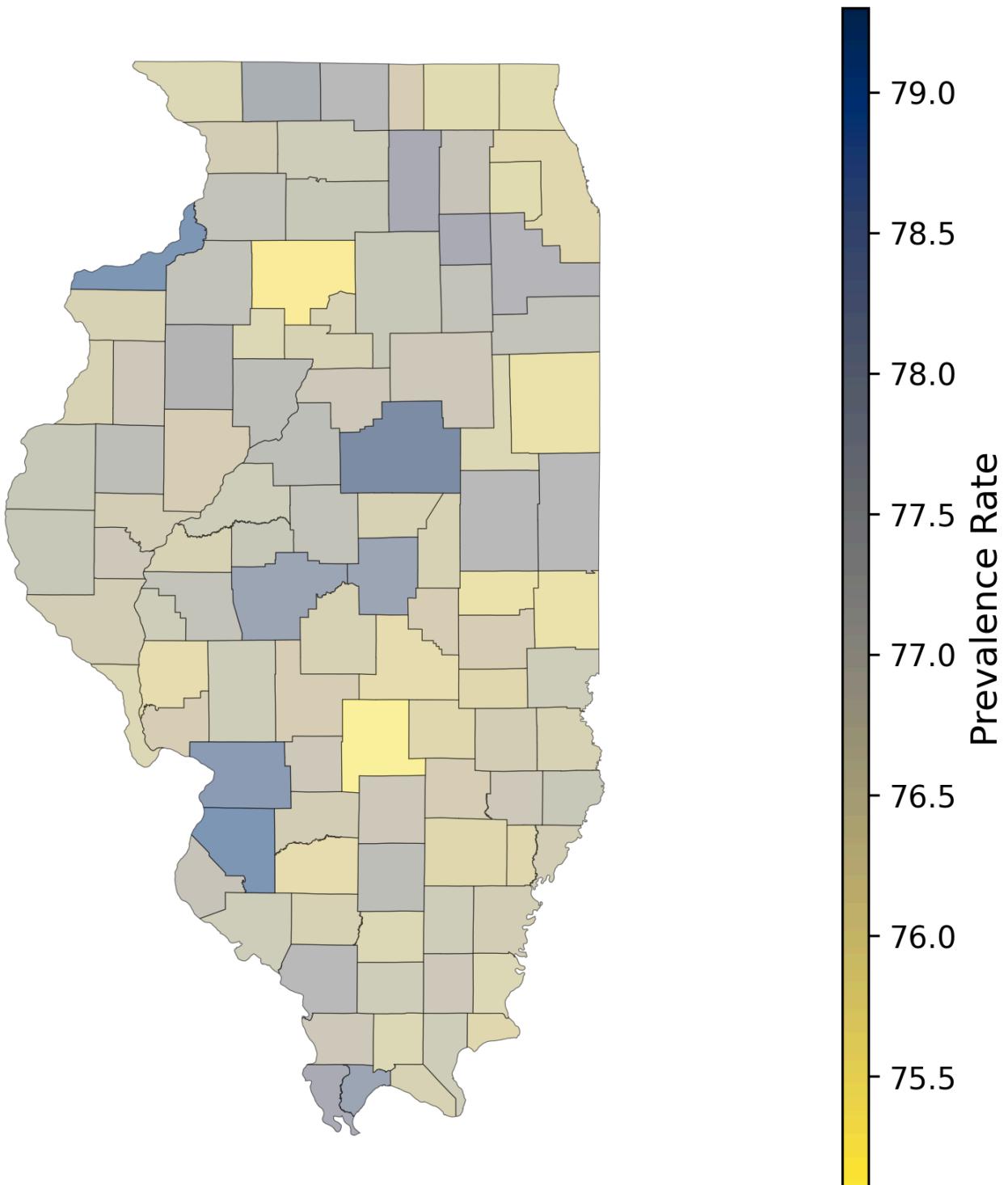
CERVICAL_AdjPrev - Illinois Map by County



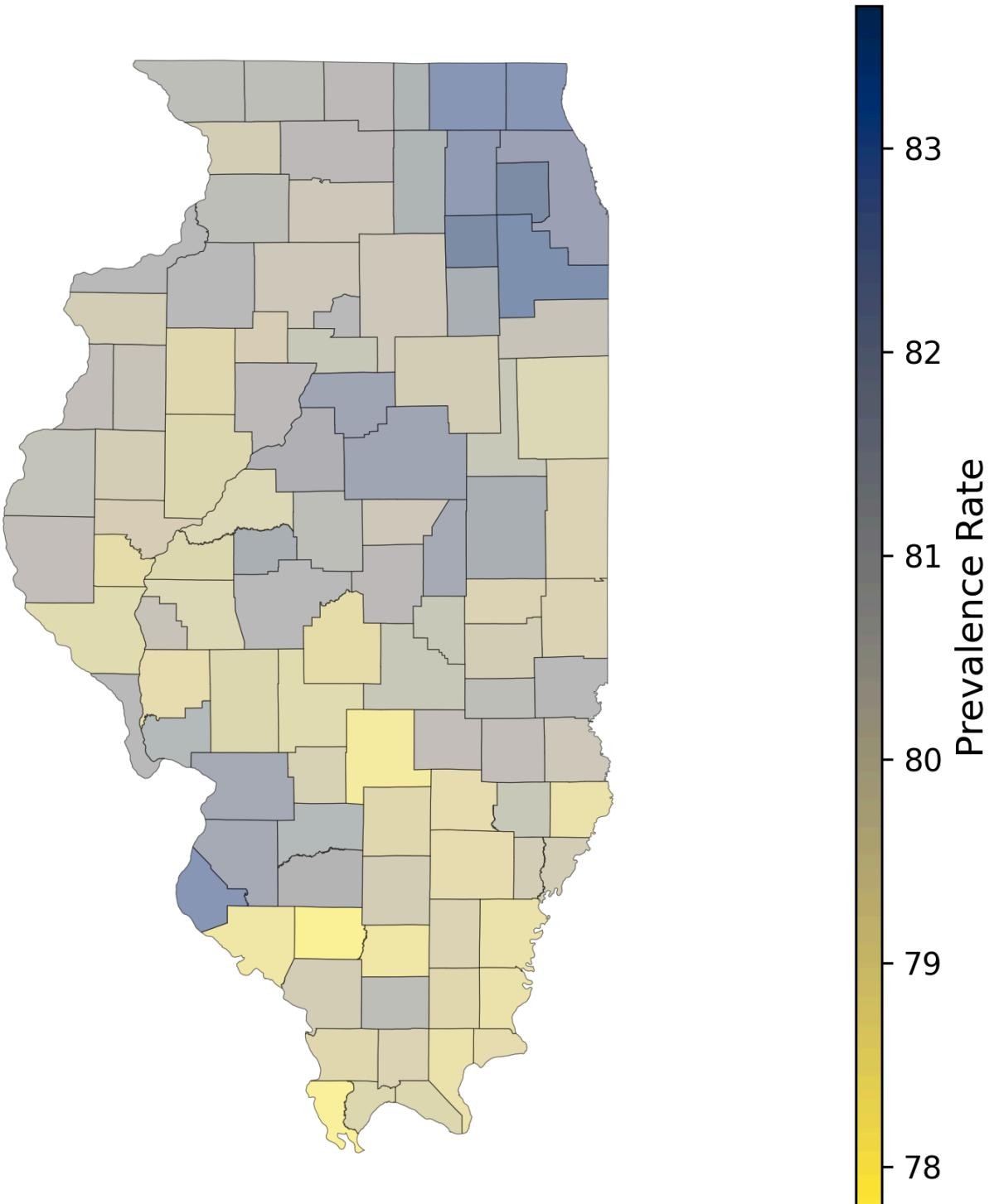
CHD_AdjPrev - Illinois Map by County



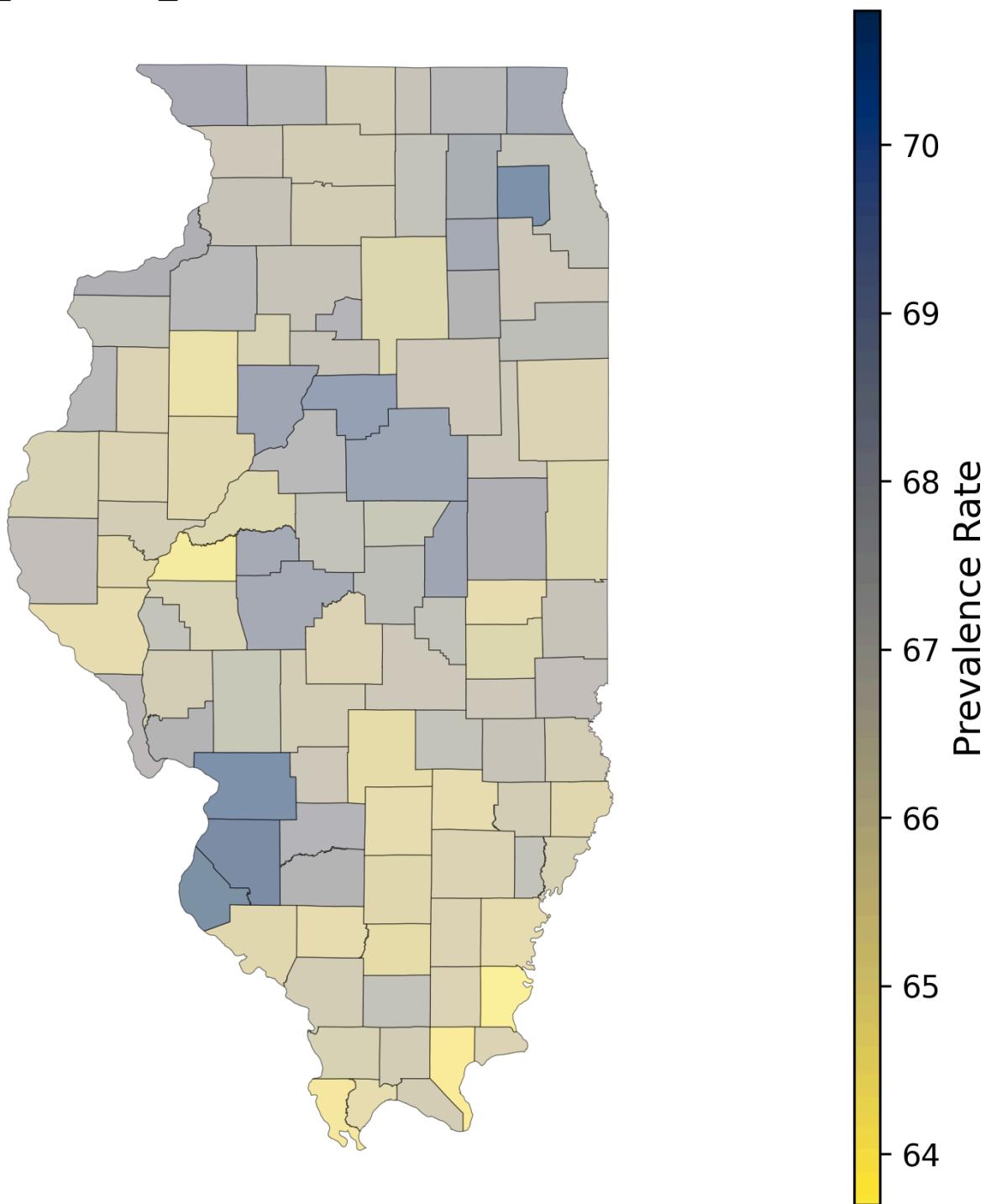
CHECKUP_AdjPrev - Illinois Map by County



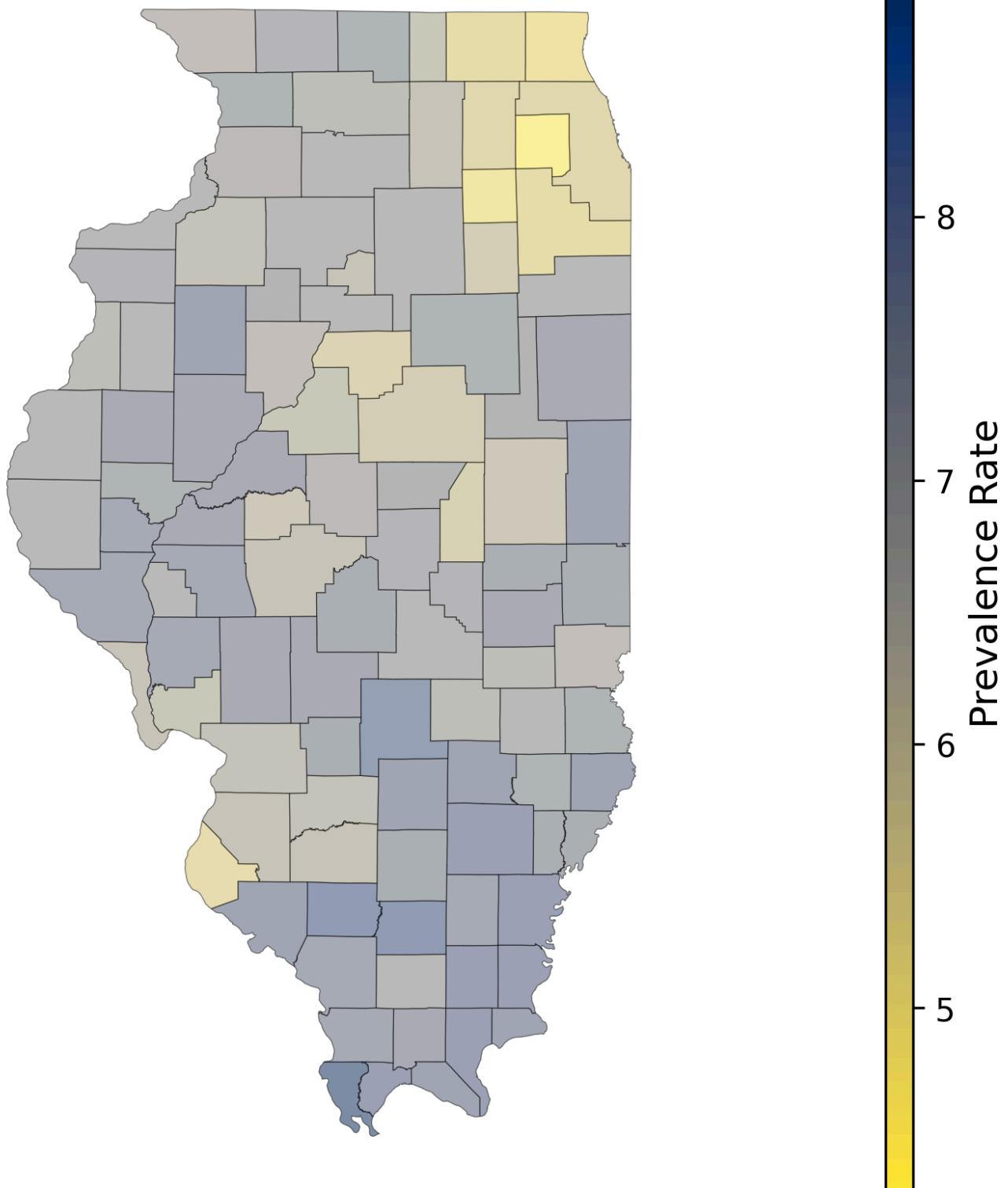
CHOLSCREEN_AdjPrev - Illinois Map by County



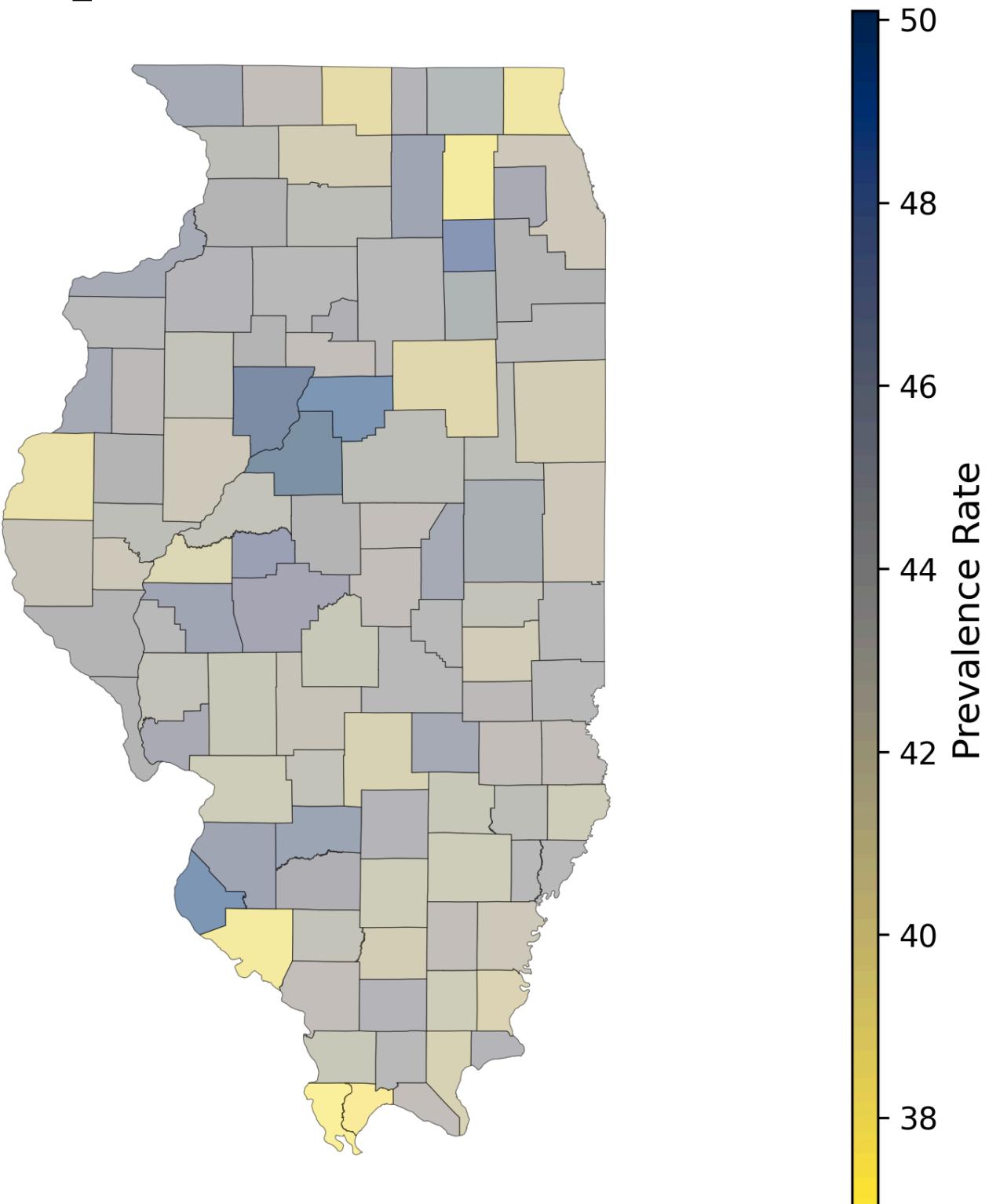
COLON_SCREEN_AdjPrev - Illinois Map by County



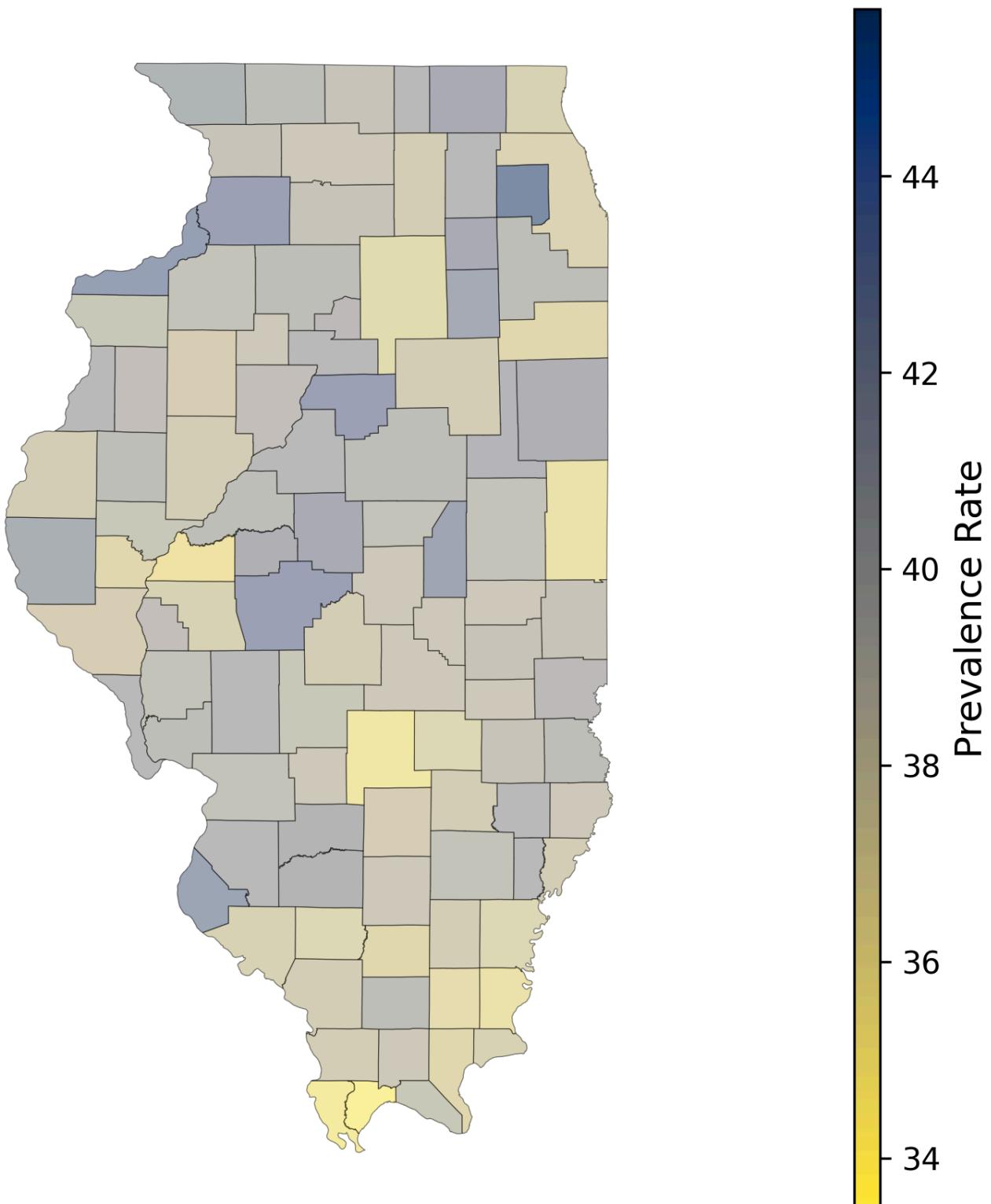
COPD_AdjPrev - Illinois Map by County



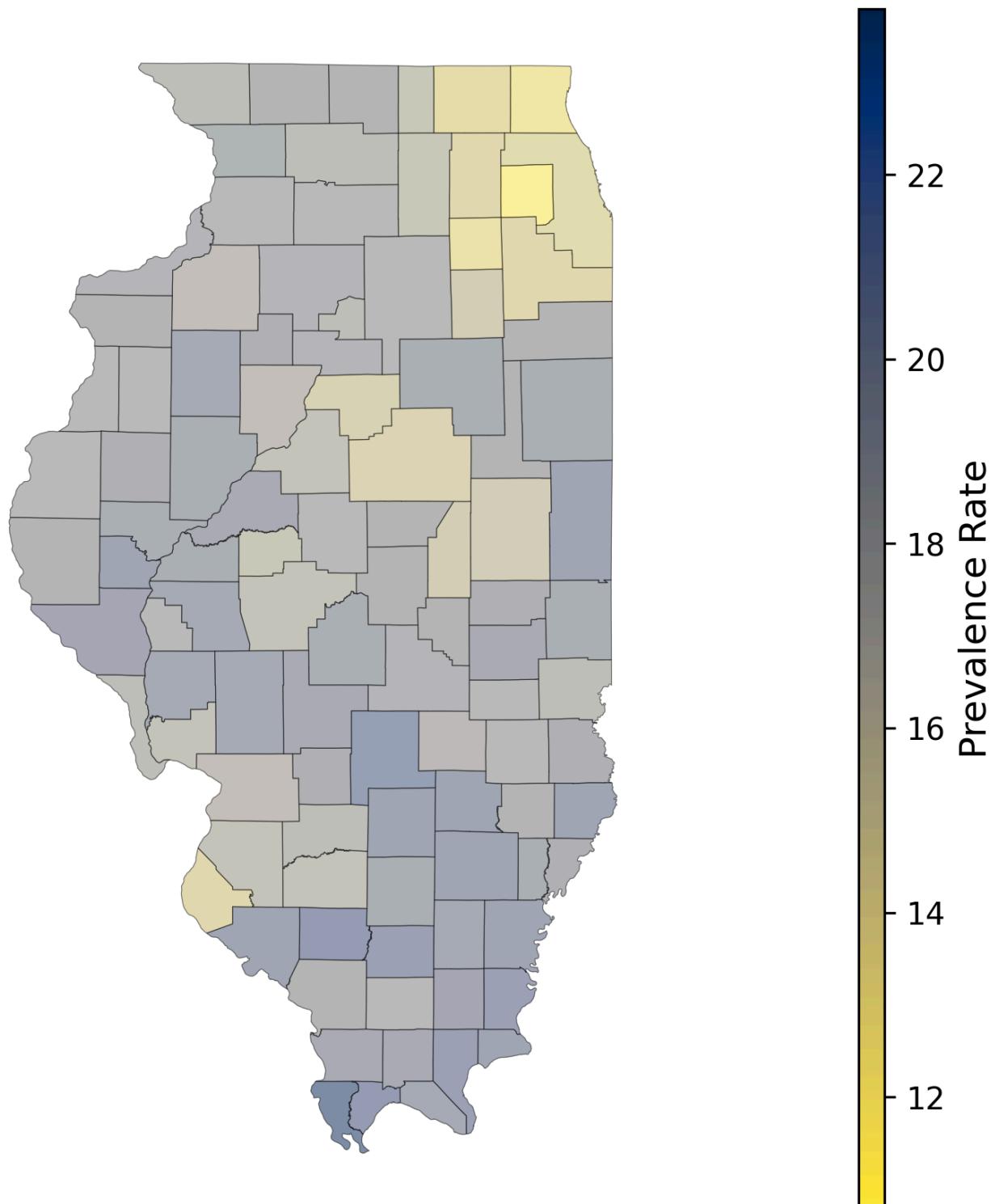
COREM_AdjPrev - Illinois Map by County



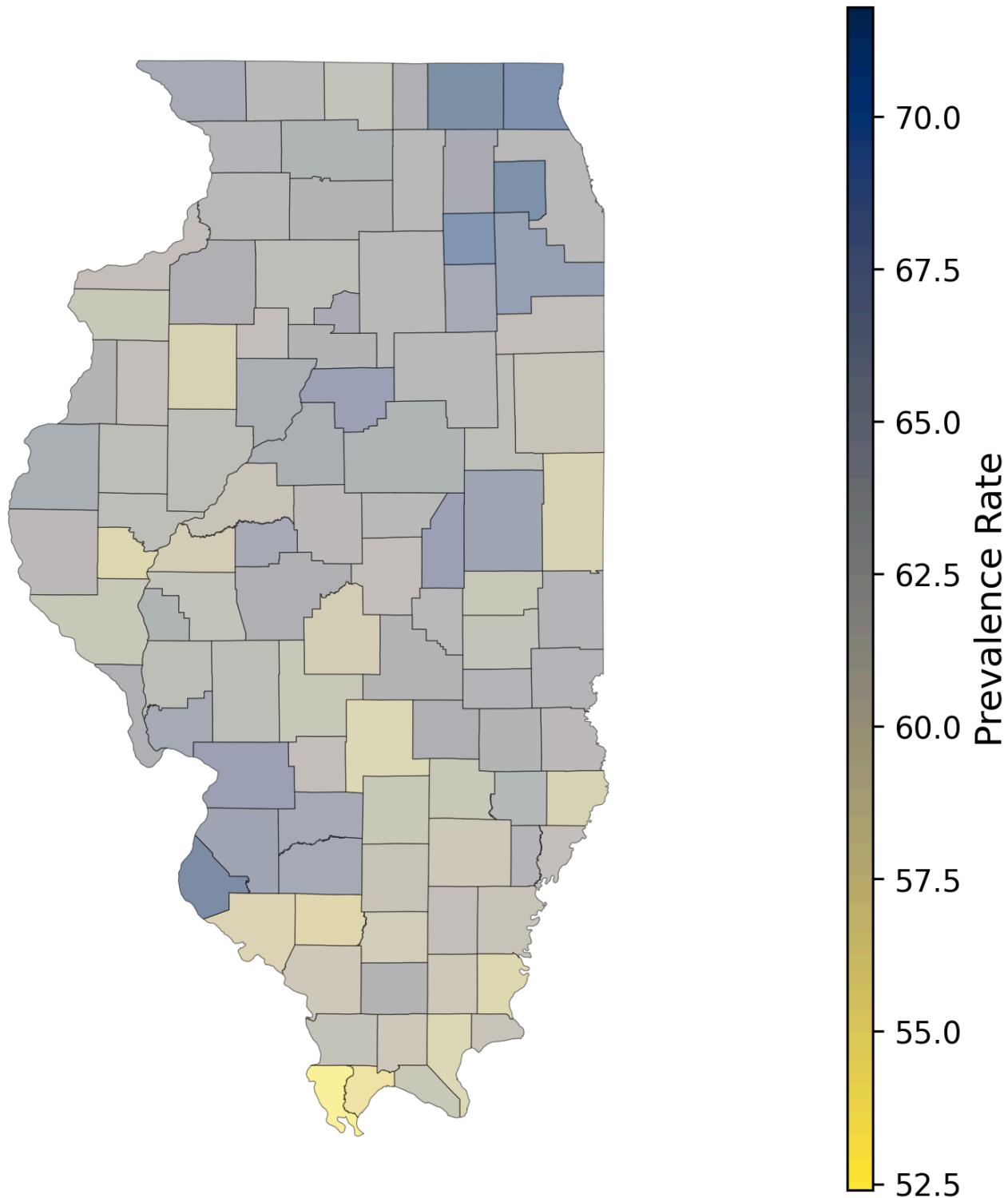
COREW_AdjPrev - Illinois Map by County



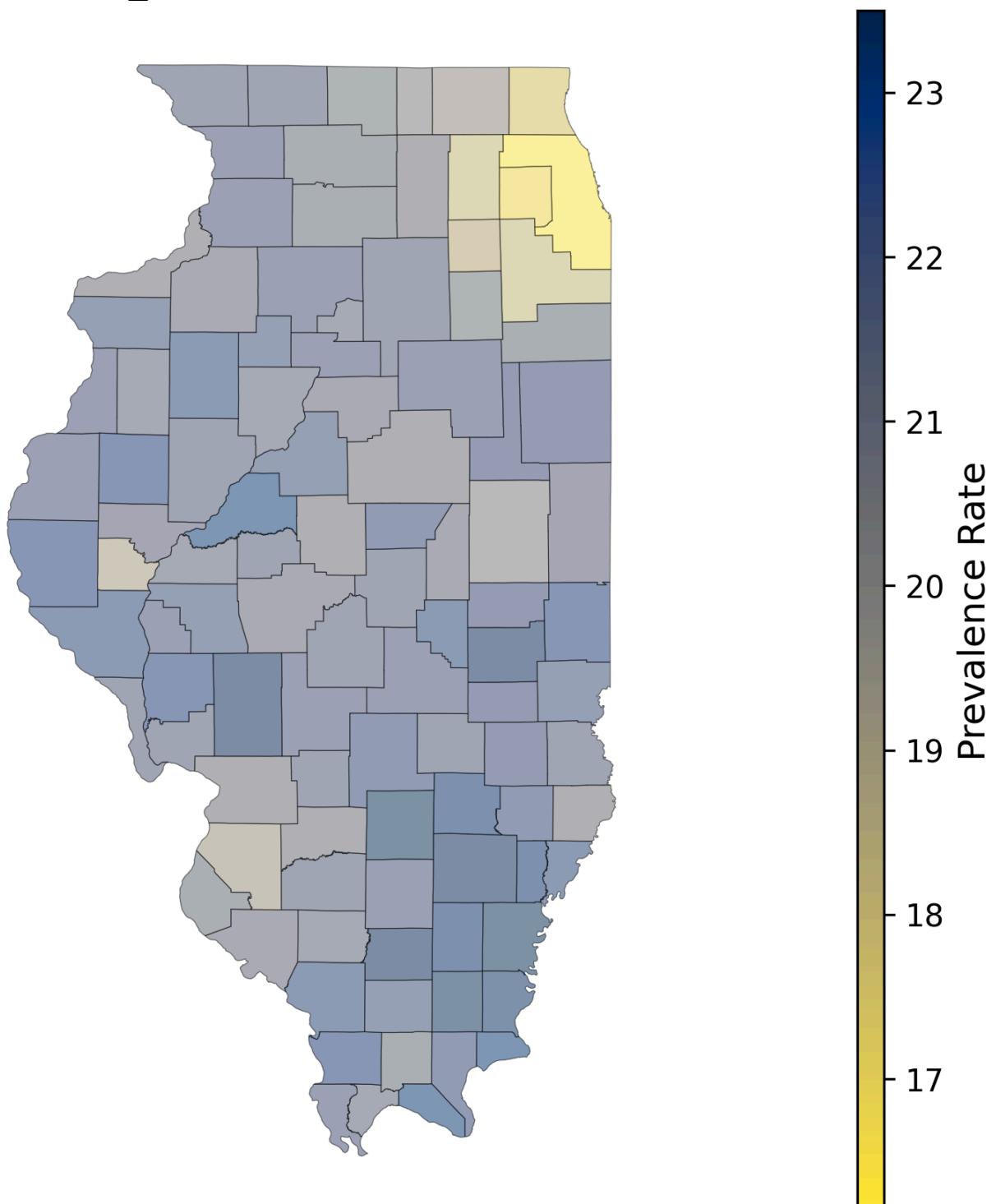
CSMOKING_AdjPrev - Illinois Map by County



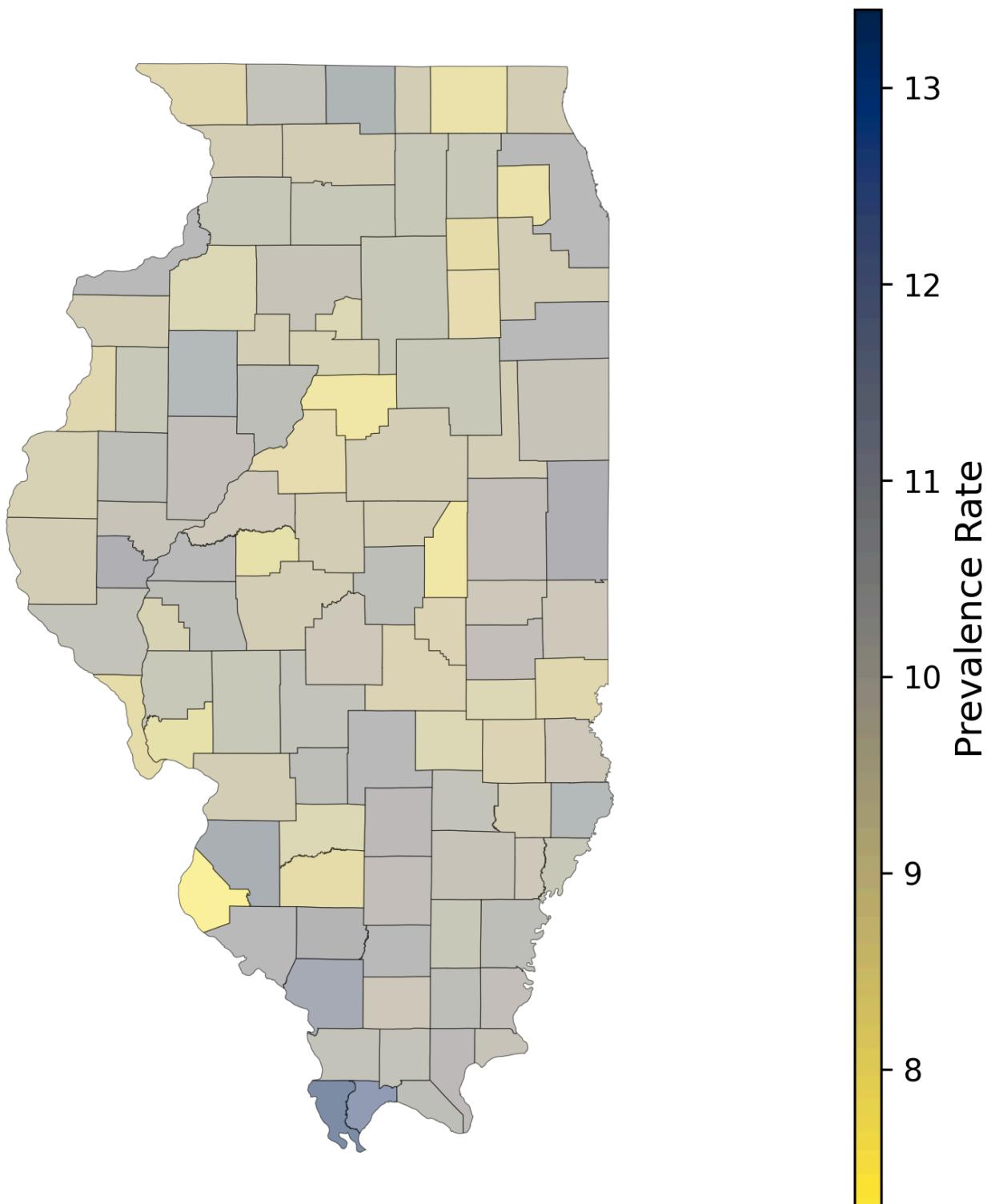
DENTAL_AdjPrev - Illinois Map by County



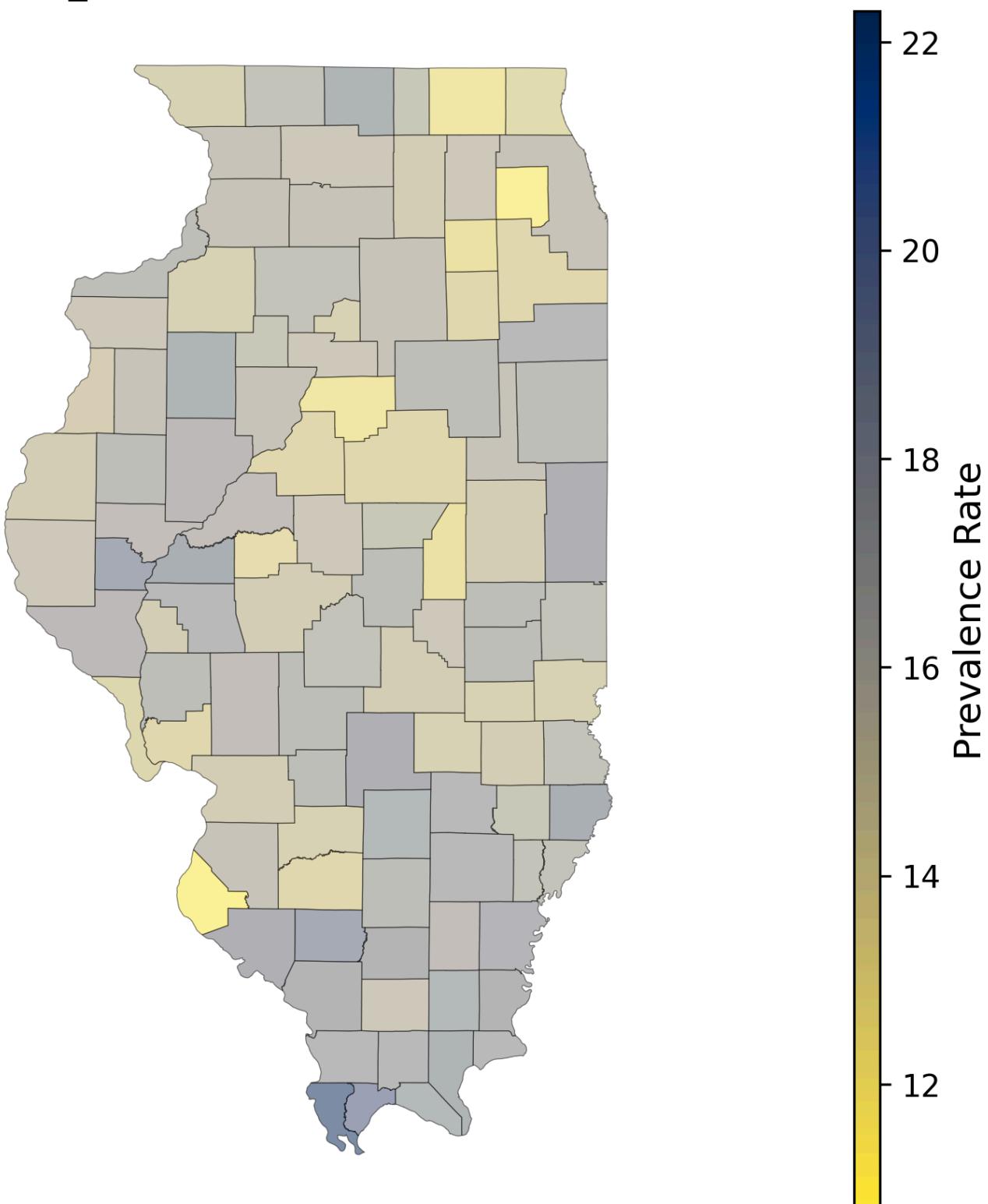
DEPRESSION_AdjPrev - Illinois Map by County



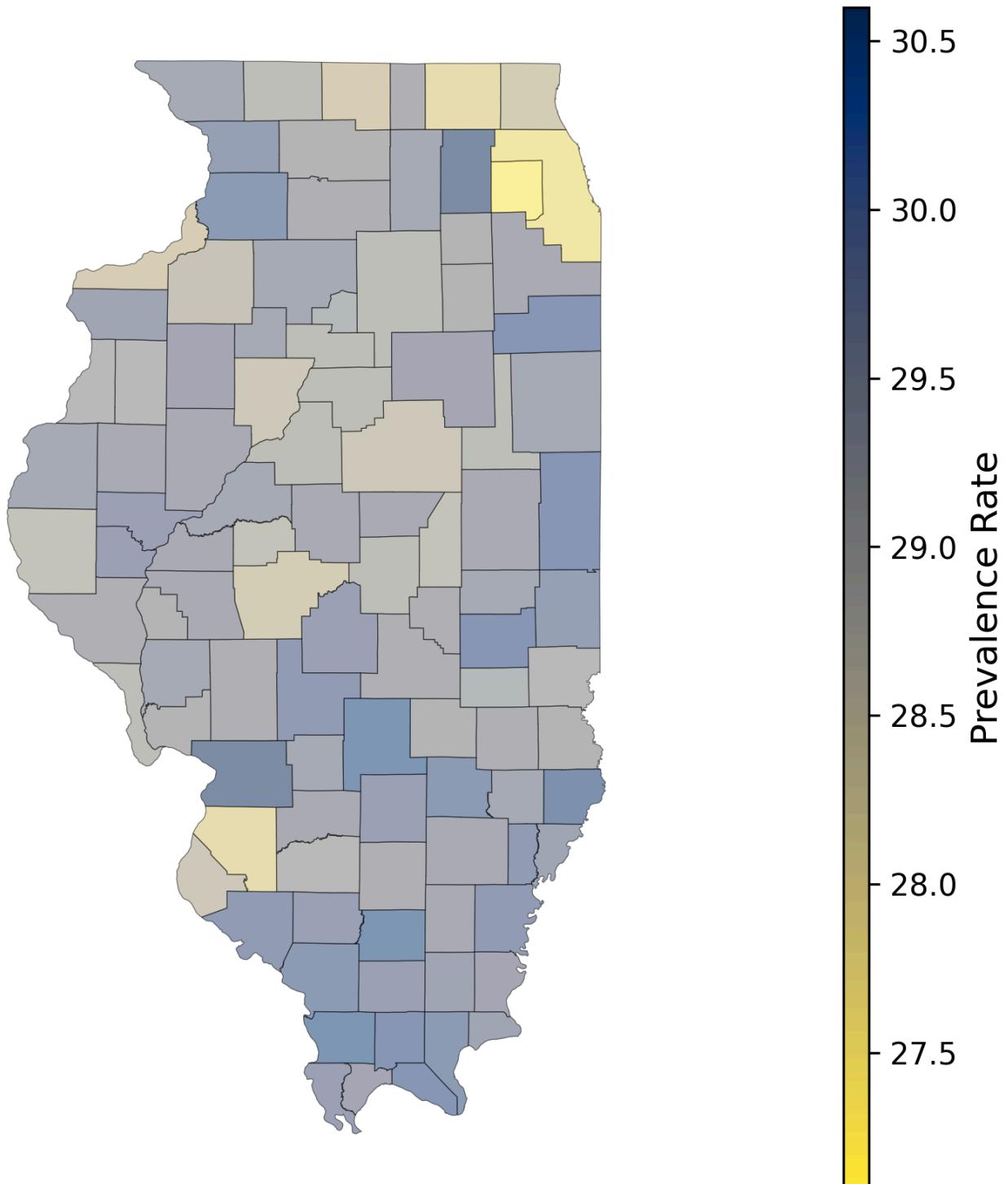
DIABETES_AdjPrev - Illinois Map by County



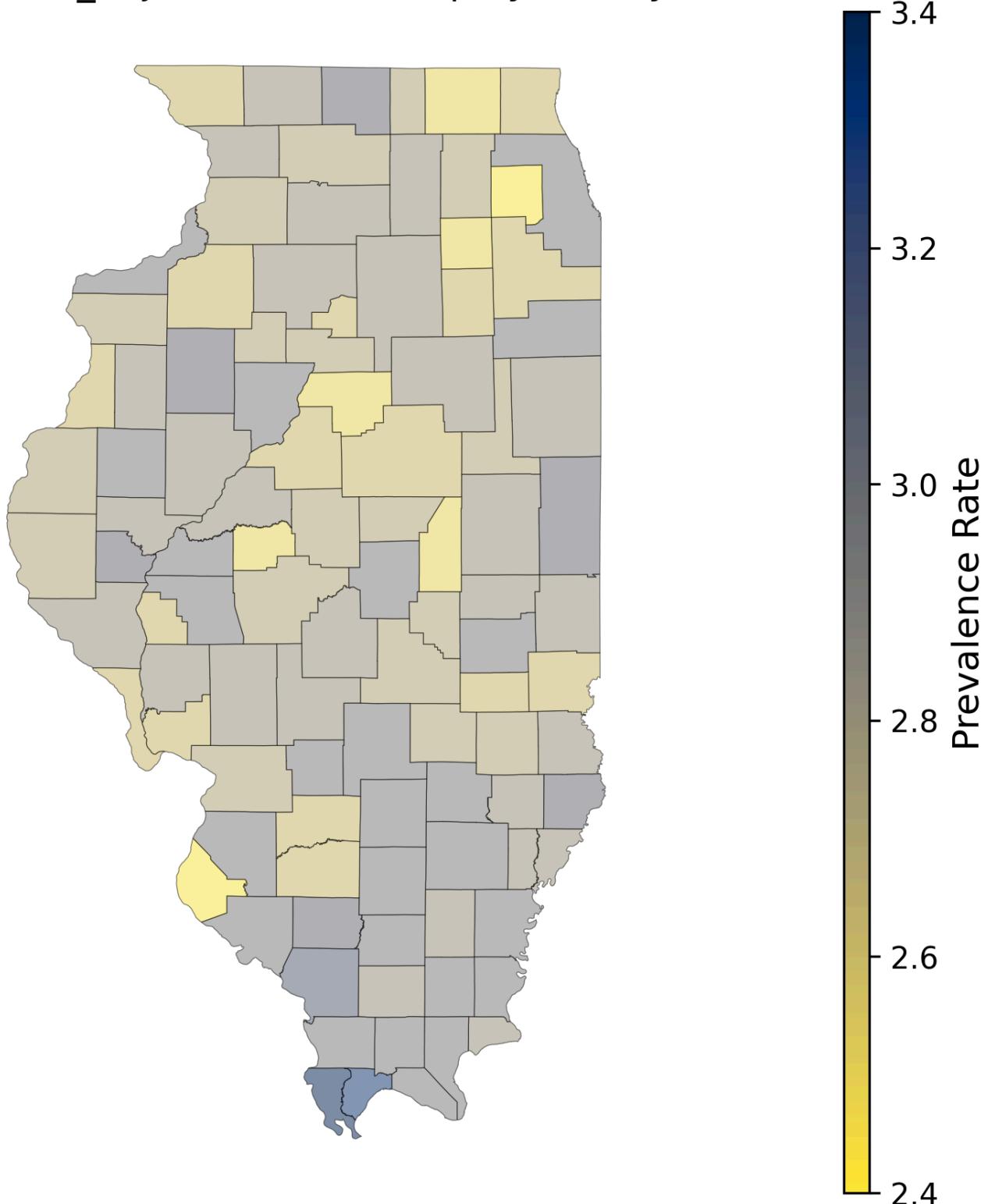
GHLTH_AdjPrev - Illinois Map by County



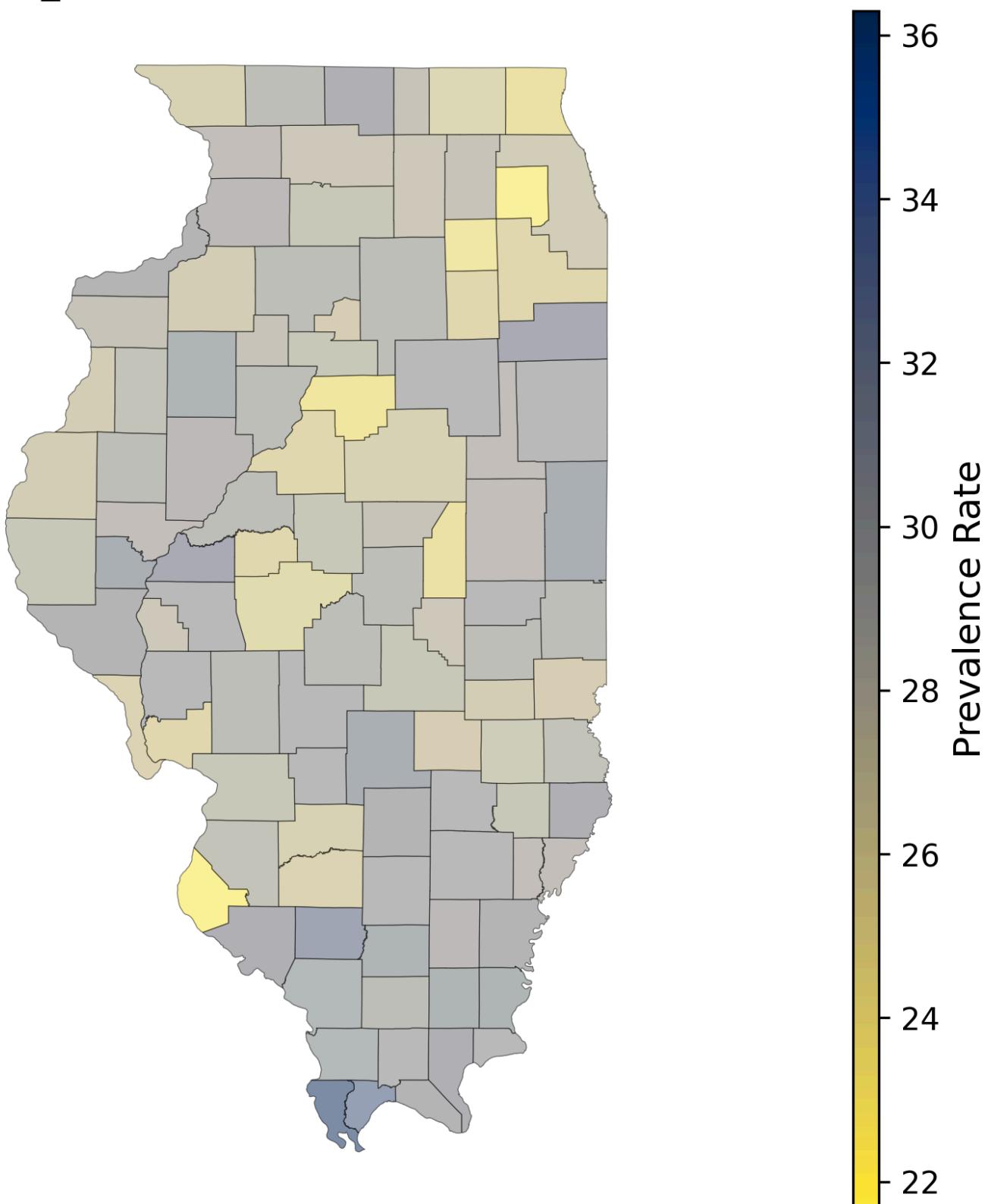
HIGHCHOL_AdjPrev - Illinois Map by County



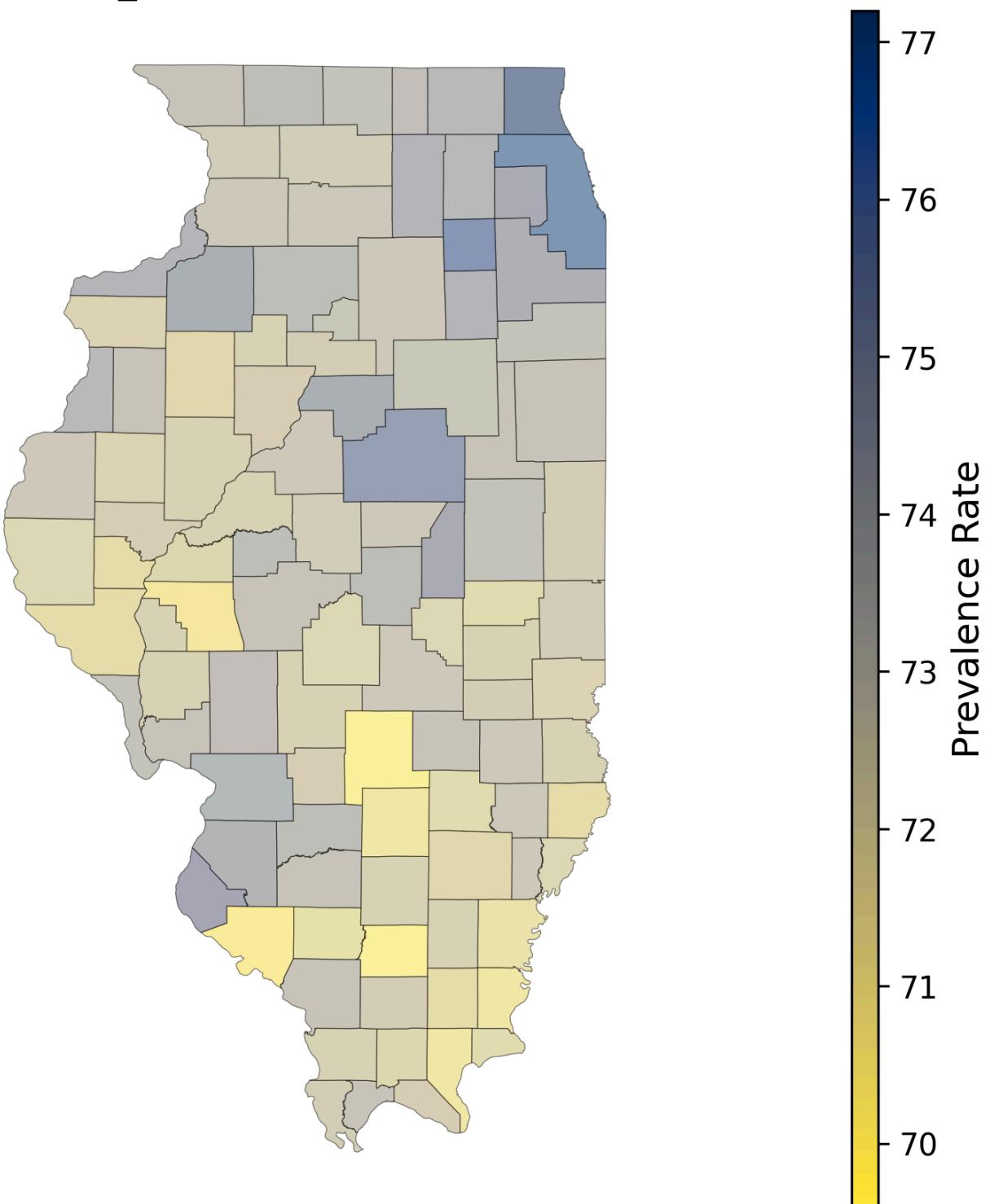
KIDNEY_AdjPrev - Illinois Map by County



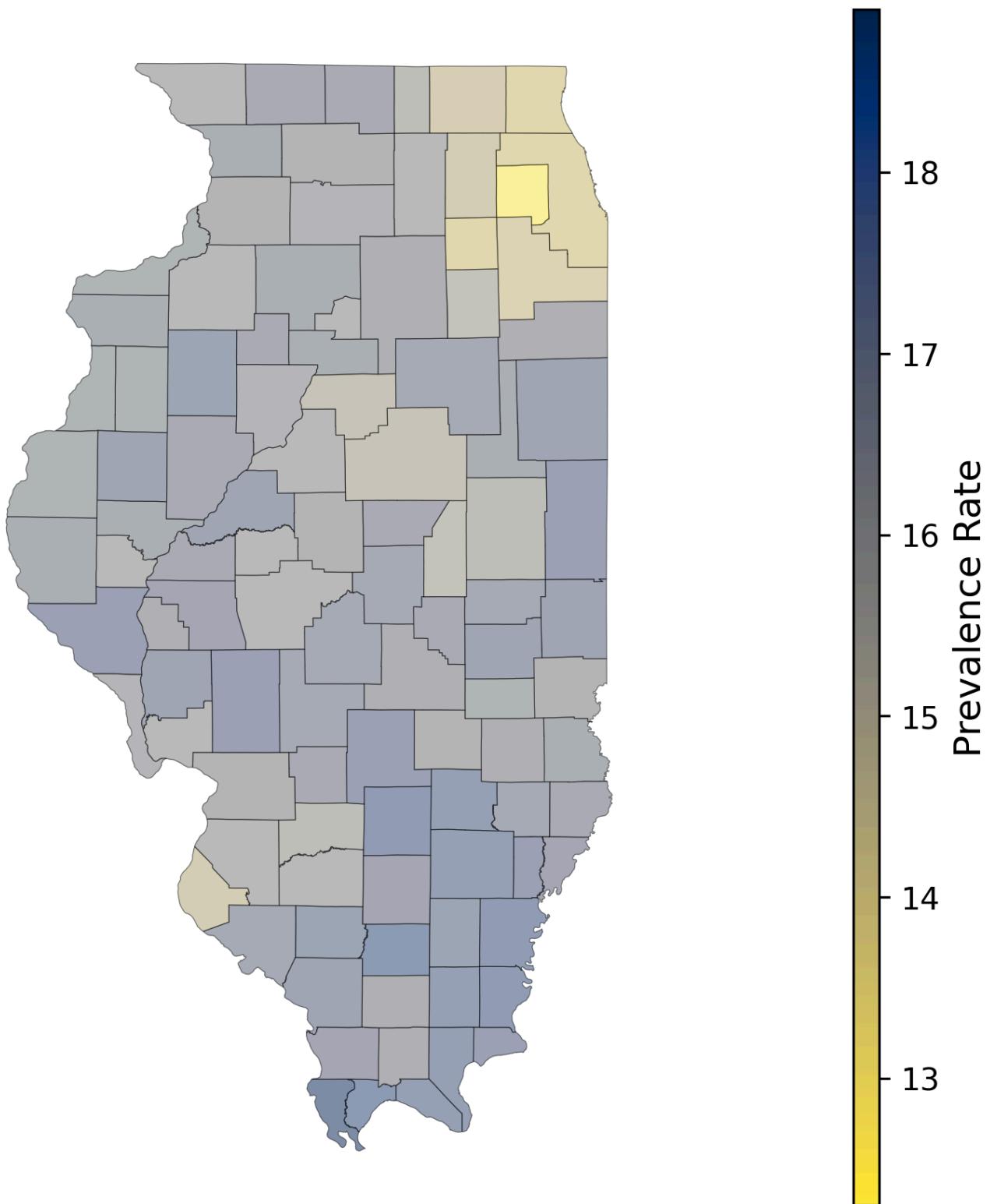
LPA_AdjPrev - Illinois Map by County



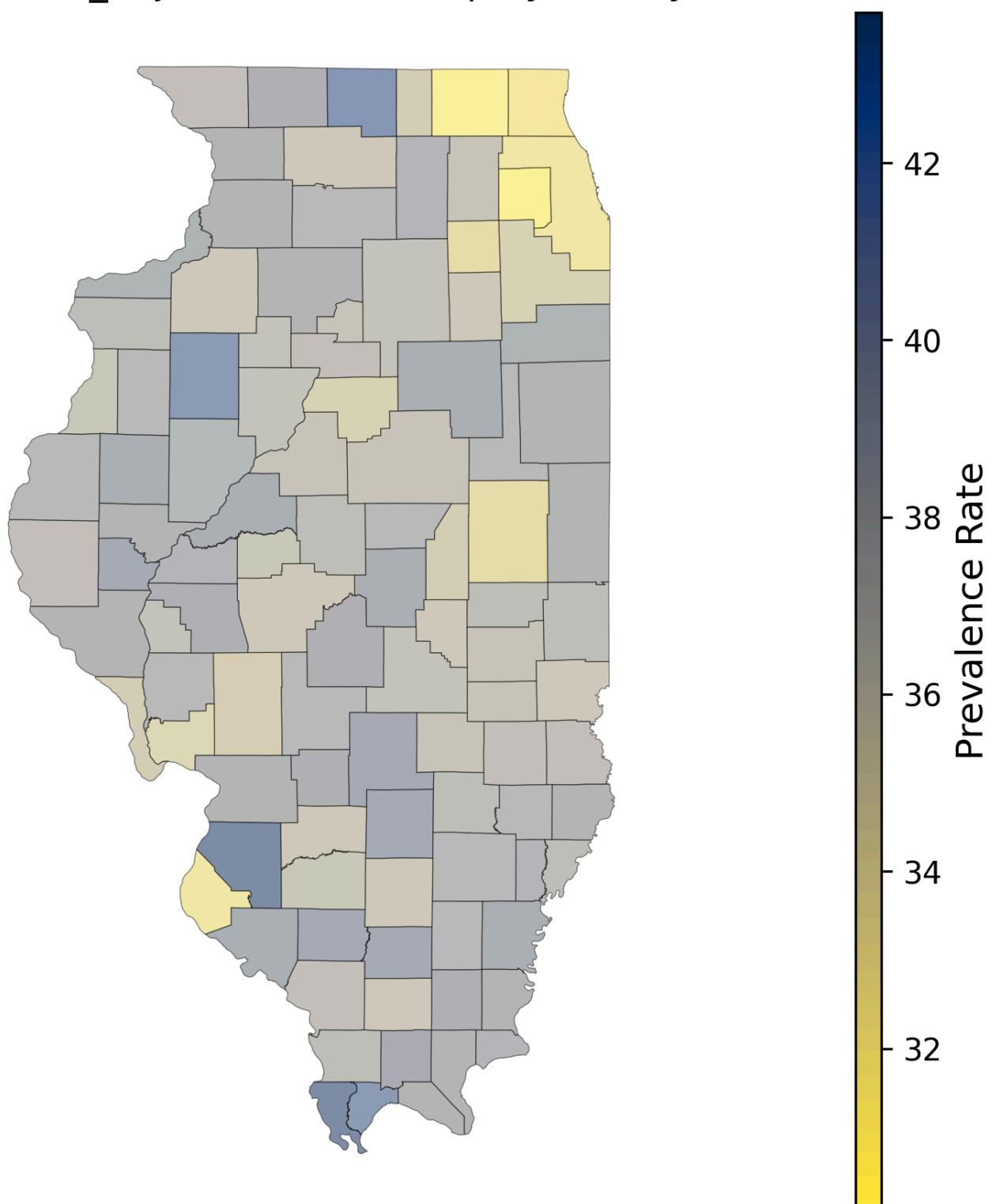
MAMMOUSE_AdjPrev - Illinois Map by County



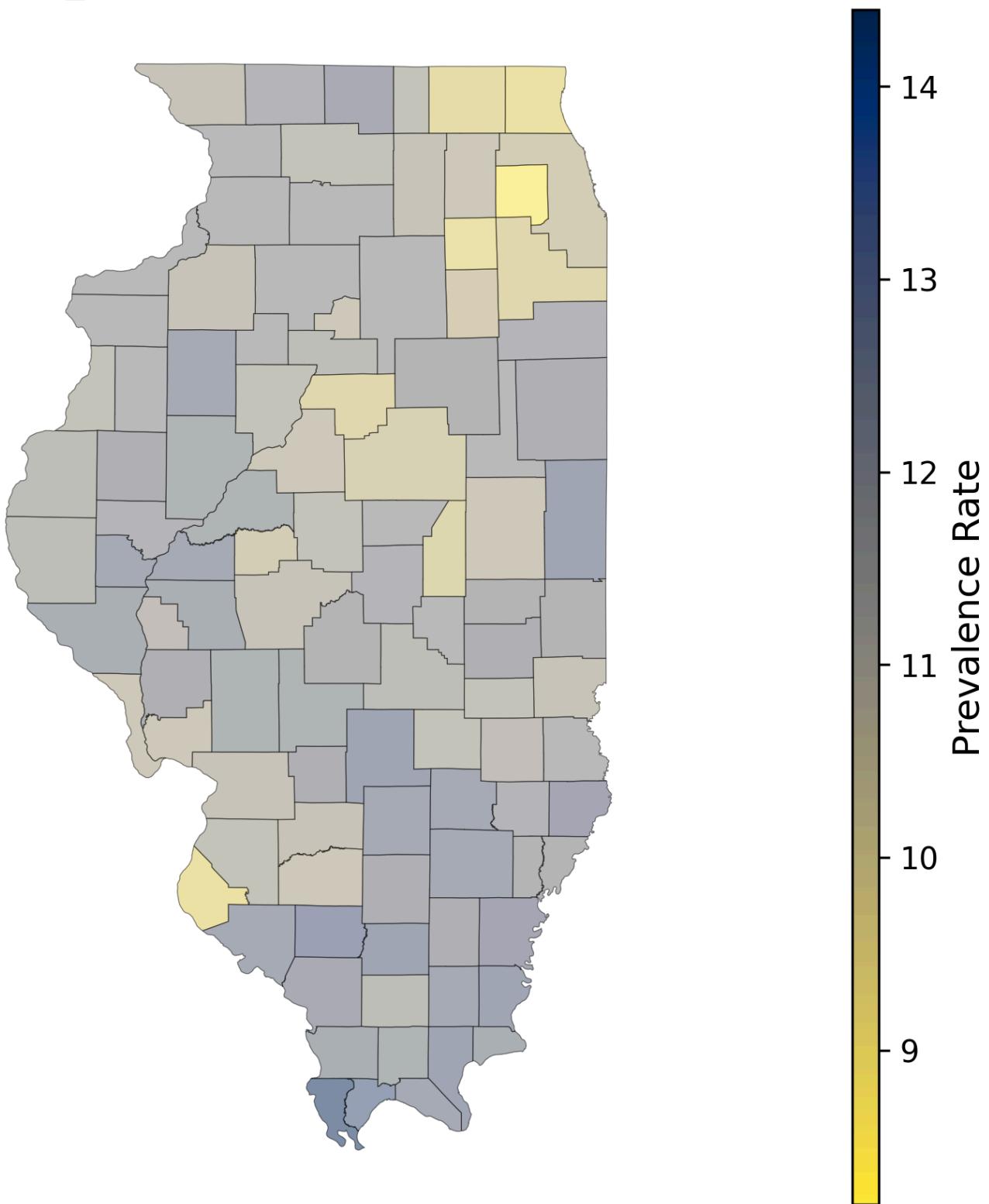
MHLTH_AdjPrev - Illinois Map by County



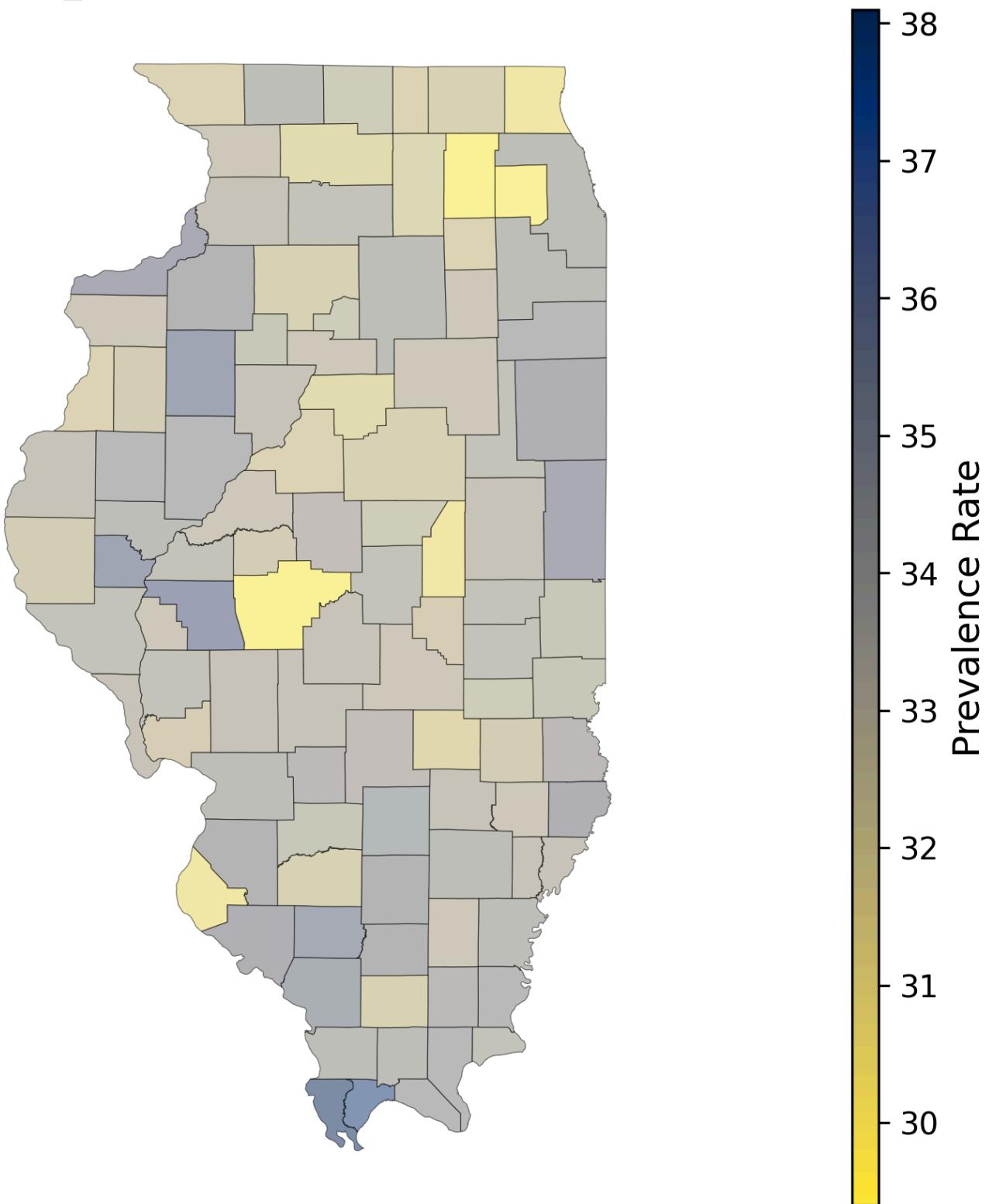
OBESITY_AdjPrev - Illinois Map by County



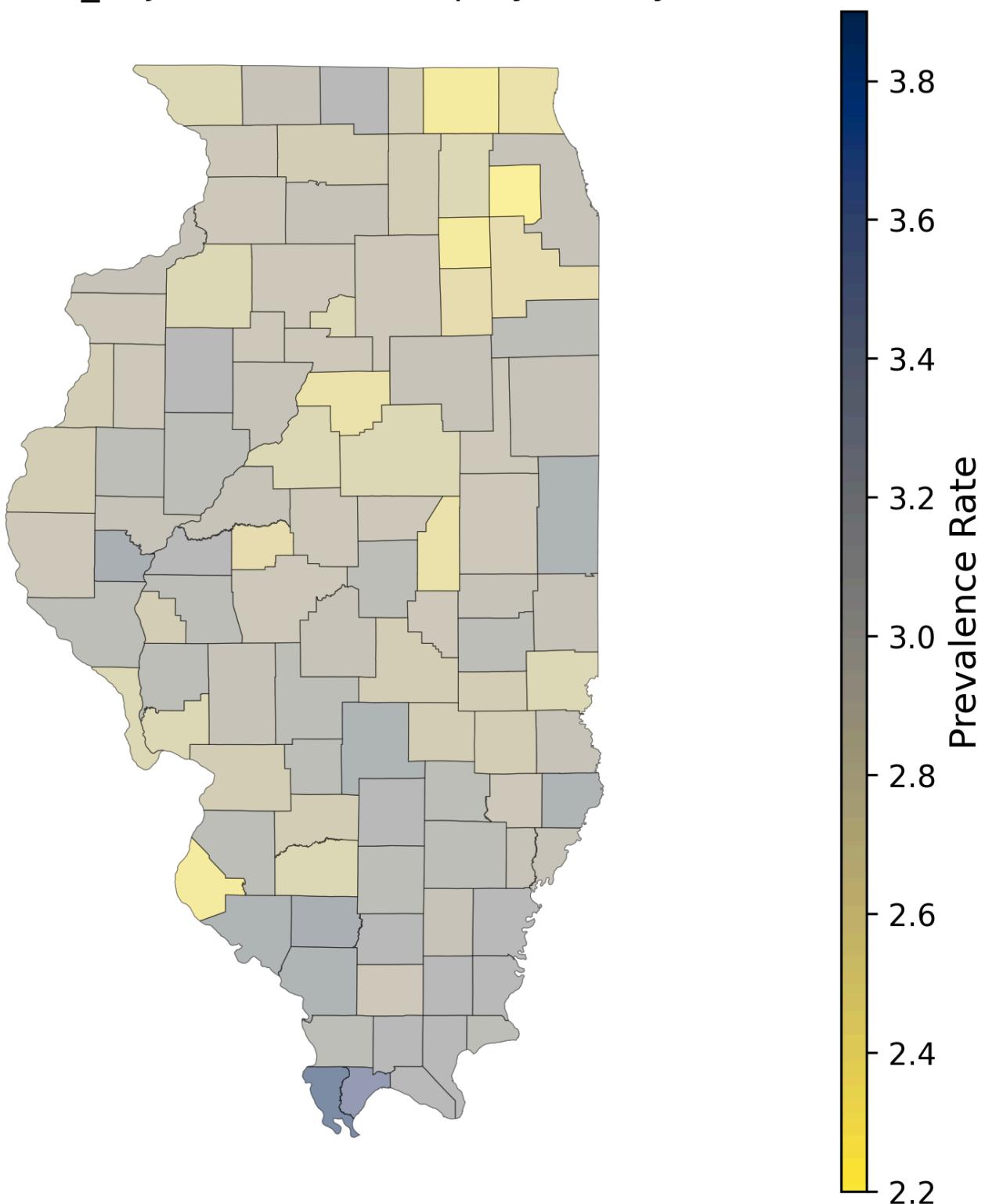
PHLTH_AdjPrev - Illinois Map by County



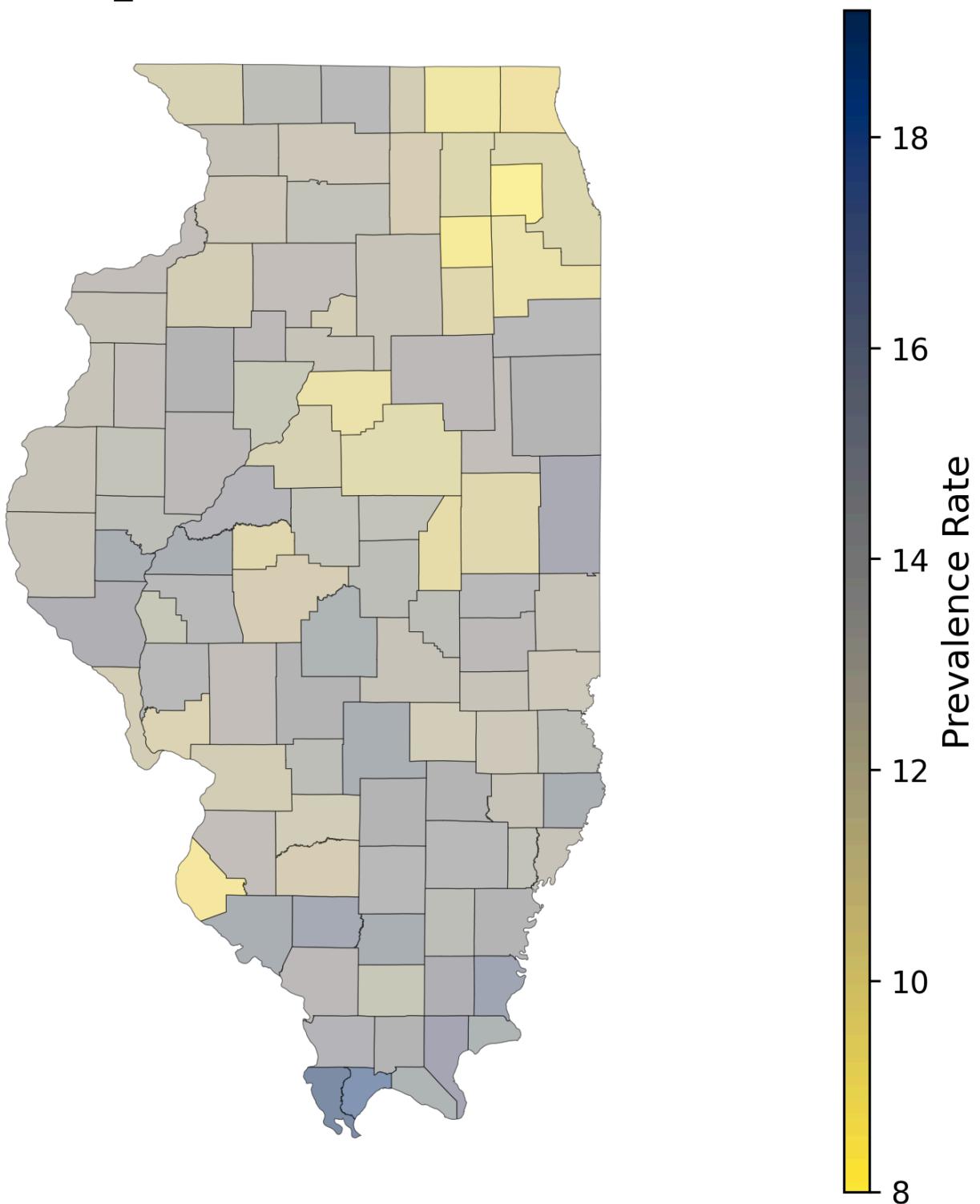
SLEEP_AdjPrev - Illinois Map by County



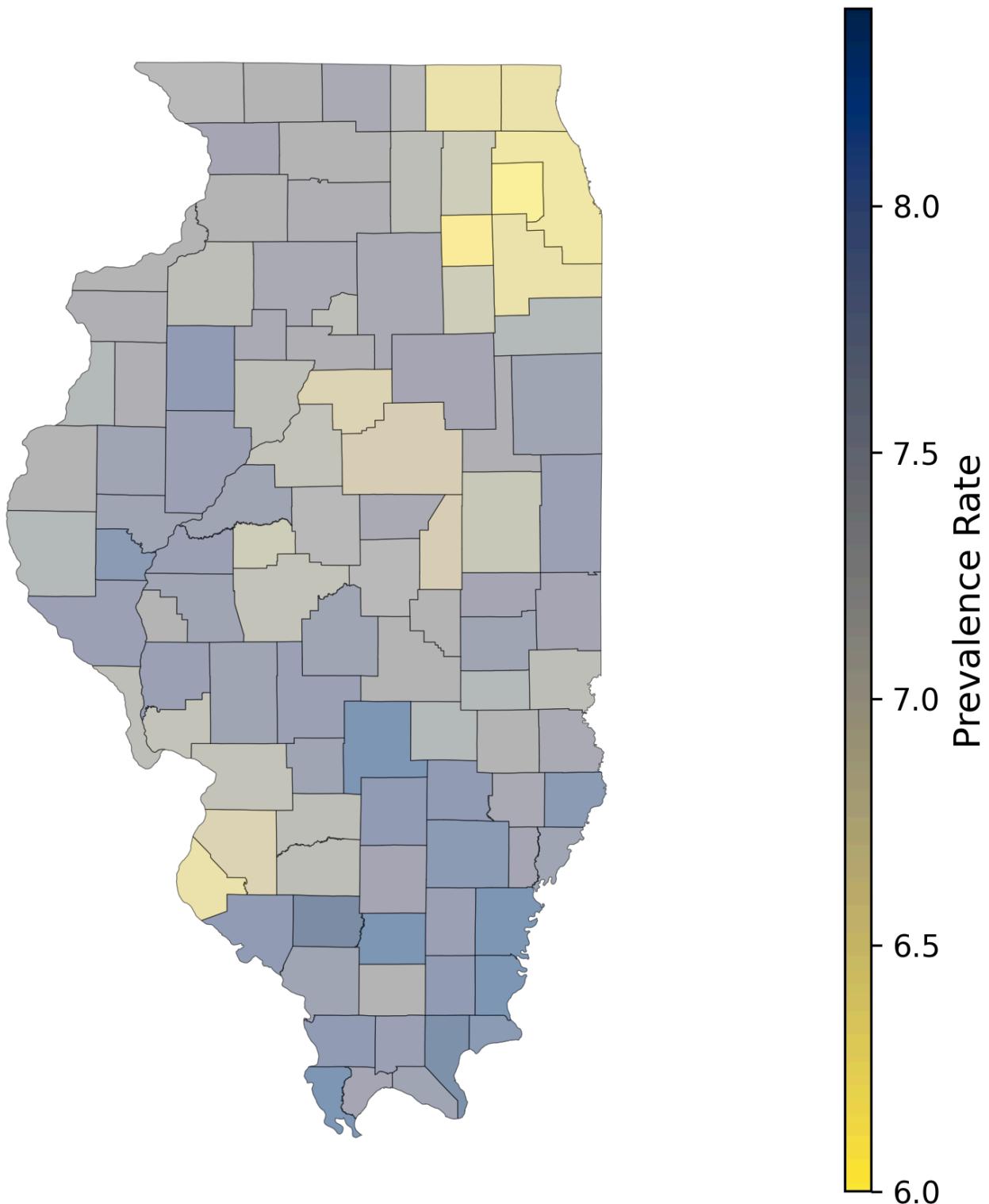
STROKE_AdjPrev - Illinois Map by County



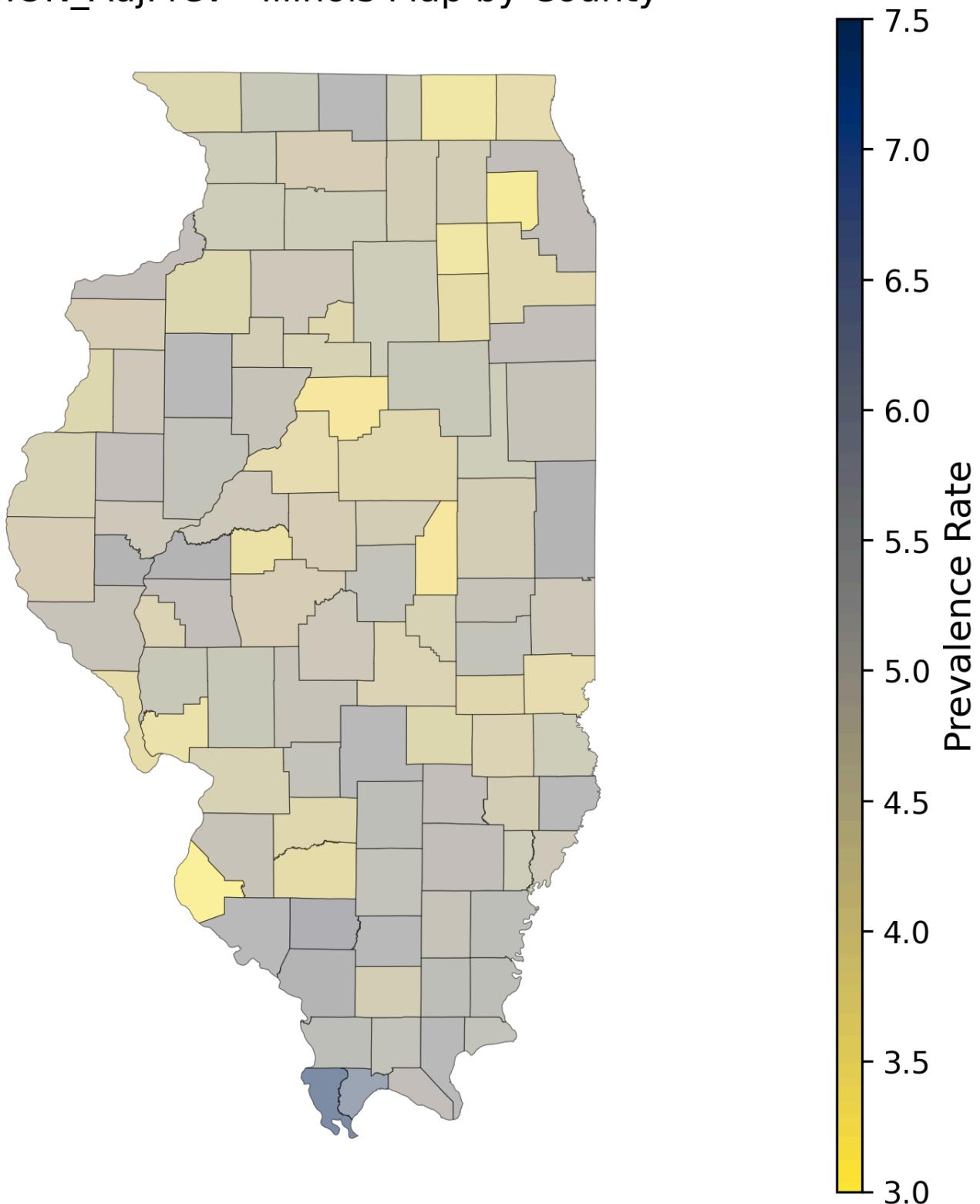
TEETHLOST_AdjPrev - Illinois Map by County



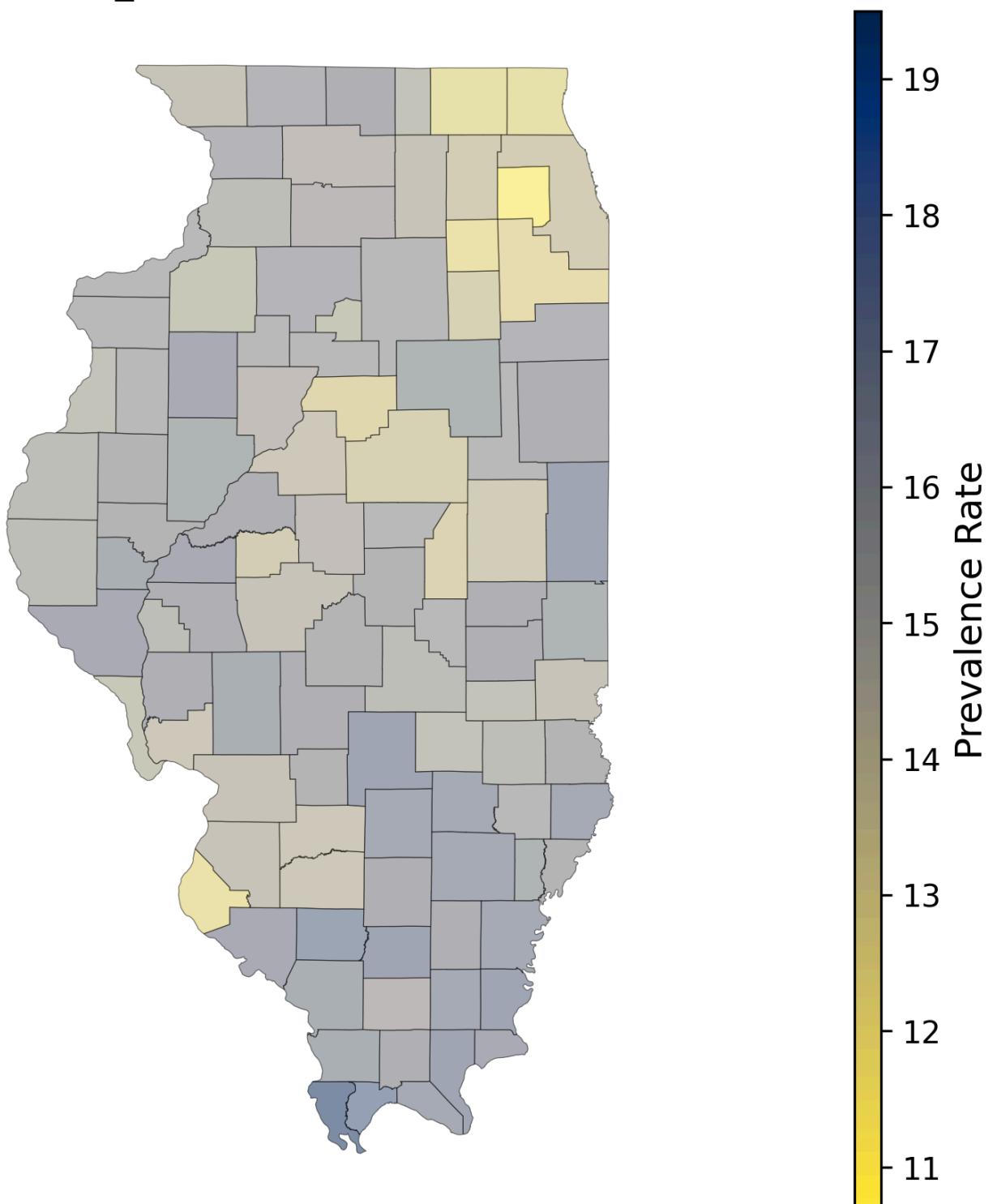
HEARING_AdjPrev - Illinois Map by County



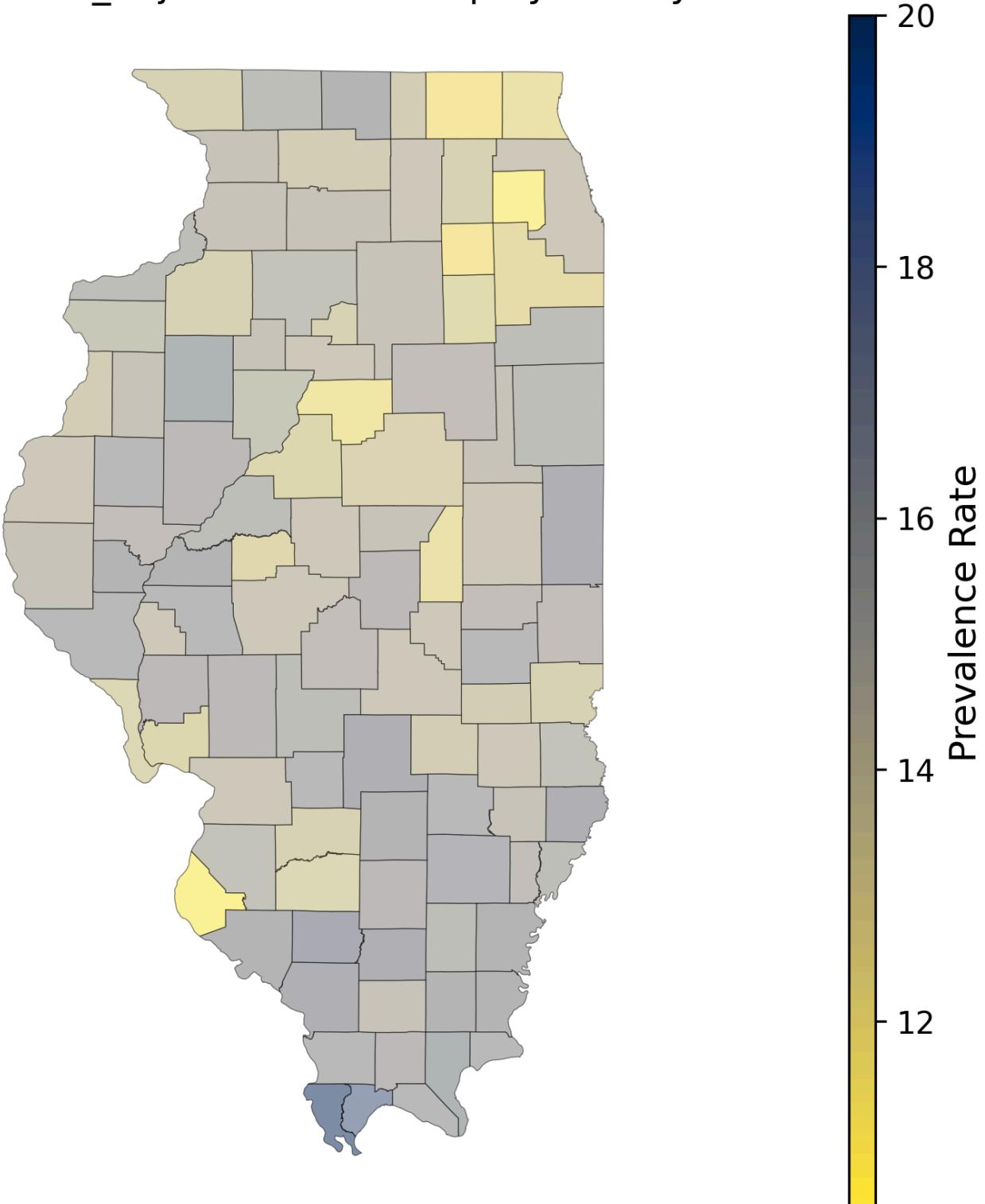
VISION_AdjPrev - Illinois Map by County



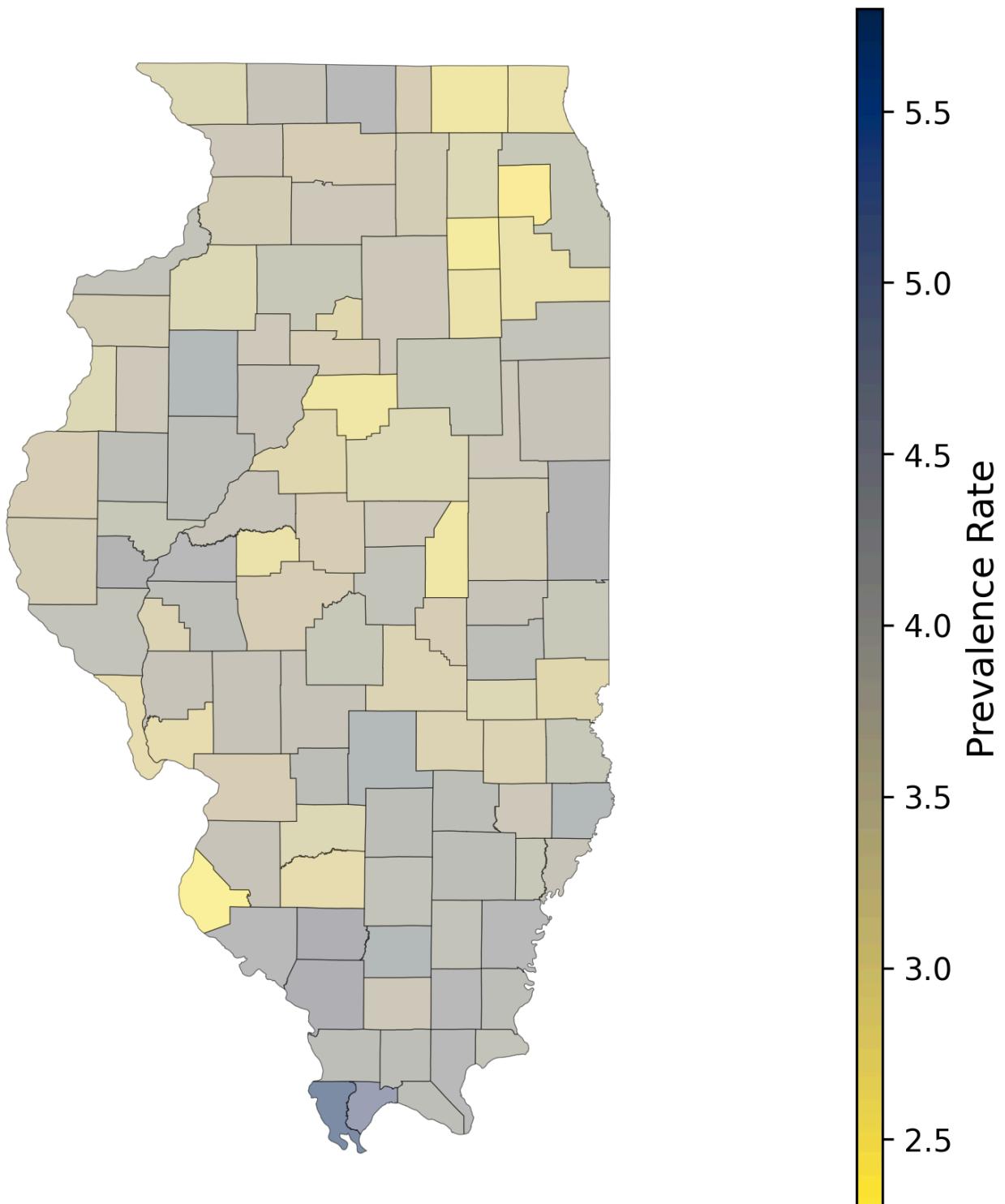
COGNITION_AdjPrev - Illinois Map by County



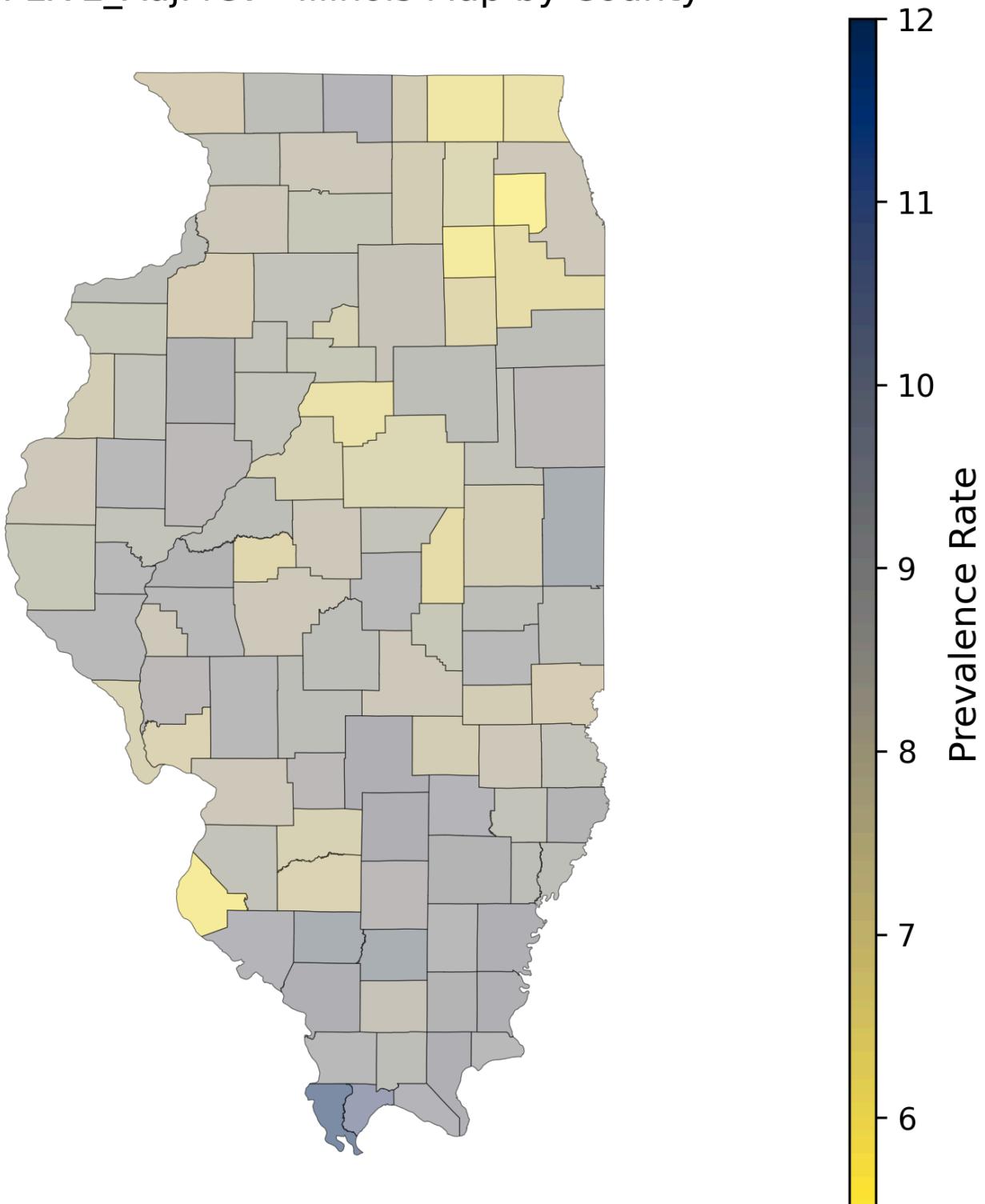
MOBILITY_AdjPrev - Illinois Map by County



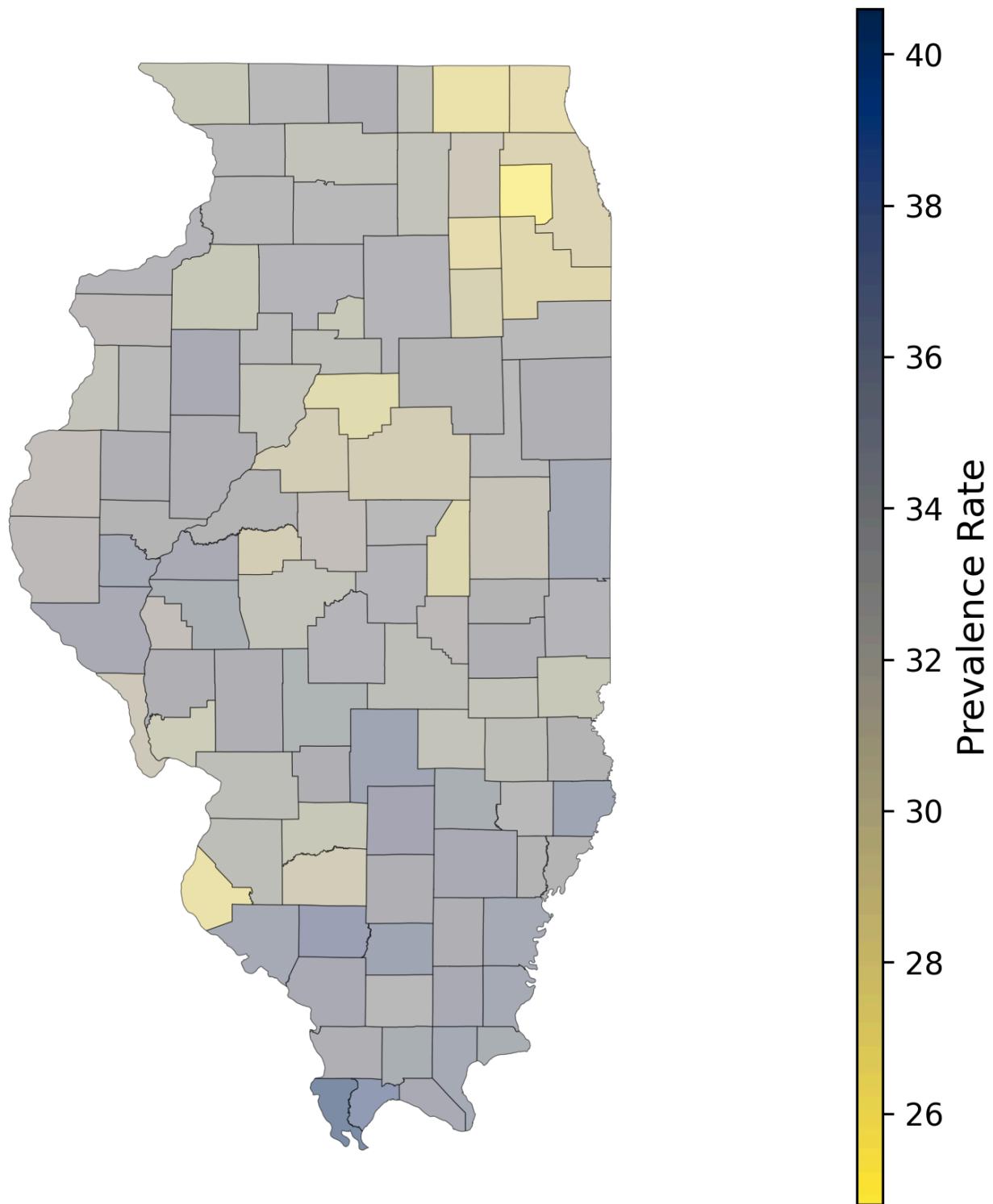
SELF CARE_AdjPrev - Illinois Map by County



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))
```

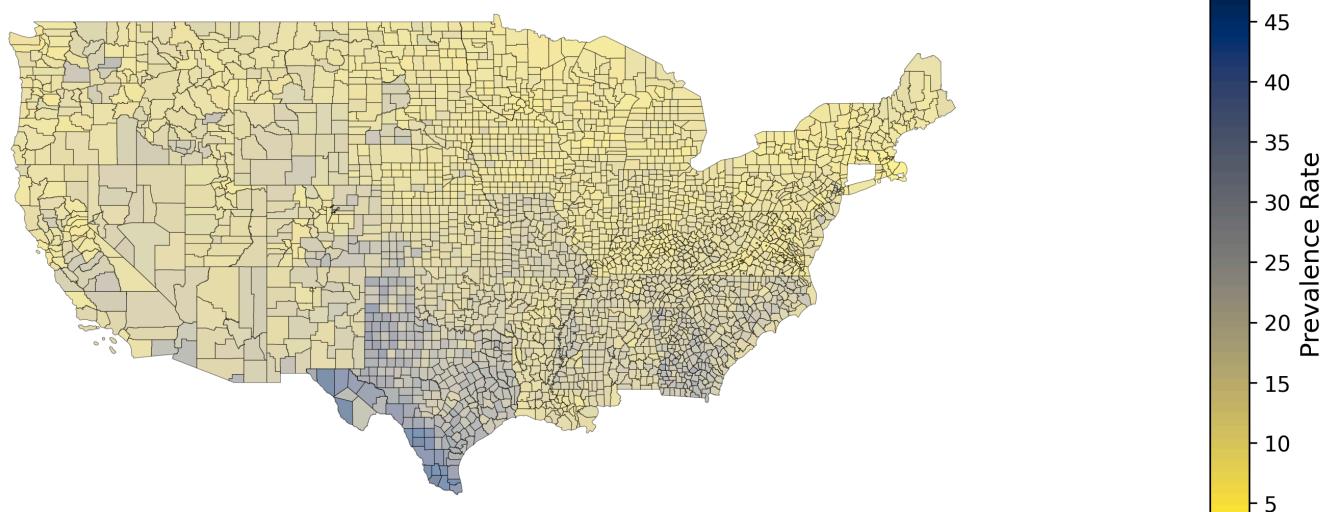
```
In [ ]: # Merge aggregated DataFrame with the GeoDataFrame  
merged_gdf = county_us.merge(df, left_on=['GEOID'], right_on=['CountyFIPS'], how='left')
```

Plotting

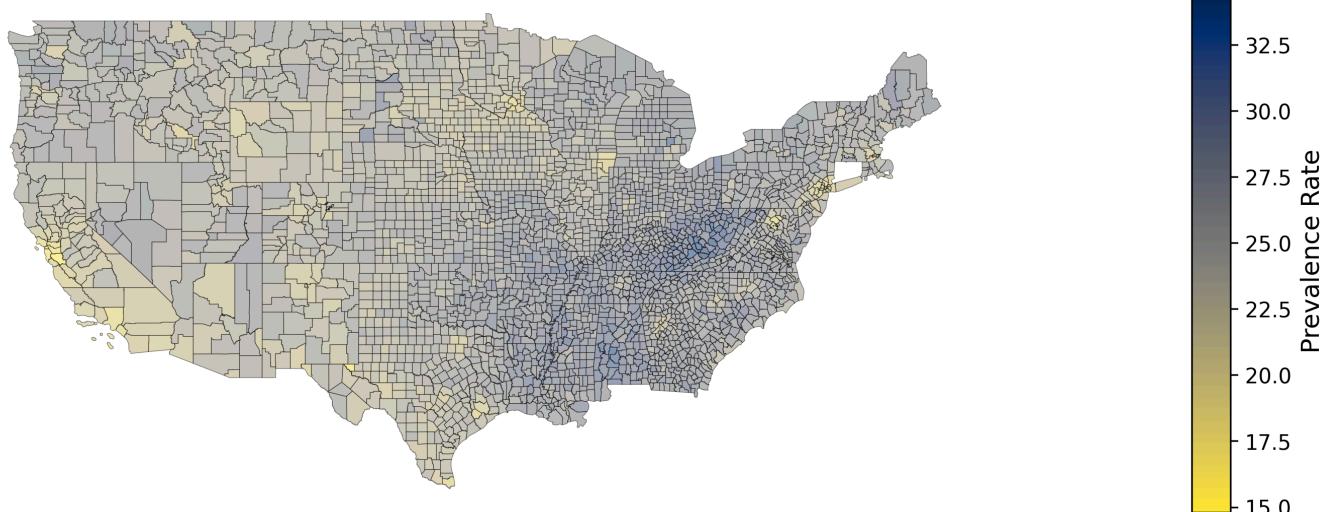
```
In [ ]: import matplotlib.pyplot as plt  
from mpl_toolkits.axes_grid1 import make_axes_locatable  
  
def plot_normalized_county_data(geo_df, data_columns, colormap):  
    for column in data_columns:  
        # Plotting  
        f, ax = plt.subplots(1, 1, figsize=(10, 6), sharex=True, sharey=True, 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')
```

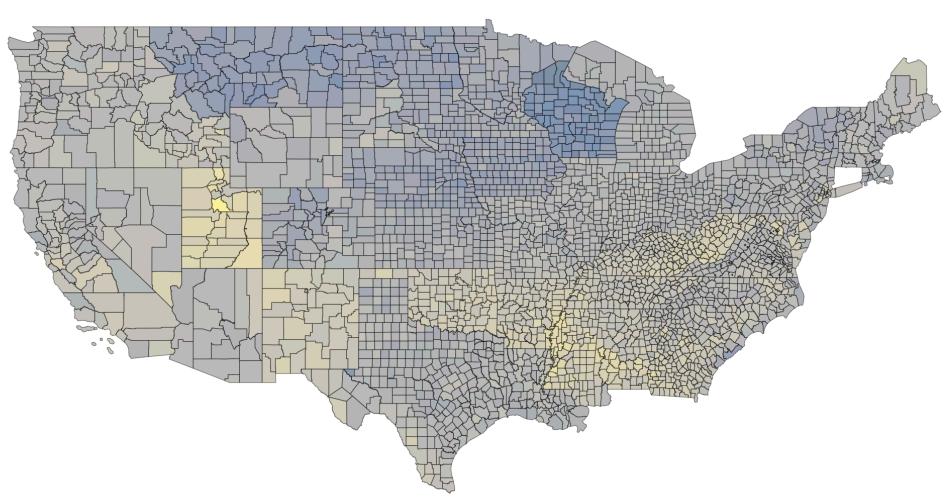
ACCESS2_AdjPrev - US Map by County



ARTHRITIS_AdjPrev - US Map by County



BINGE_AdjPrev - US Map by County

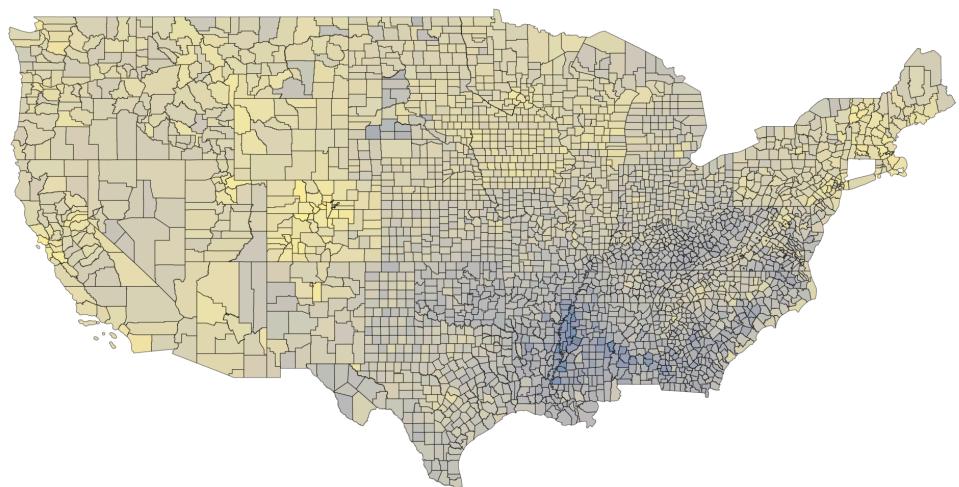


Prevalence Rate

Prevalence Rate

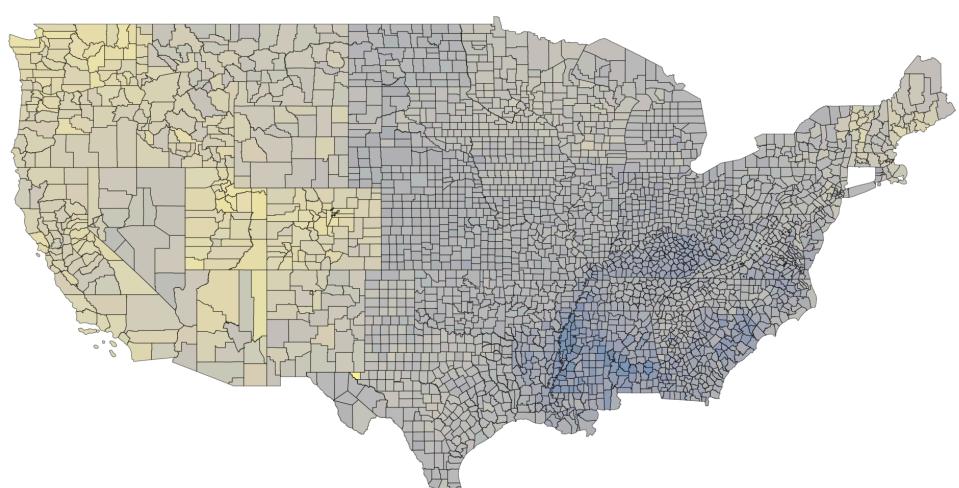
Prevalence Rate

BPHIGH_AdjPrev - US Map by County



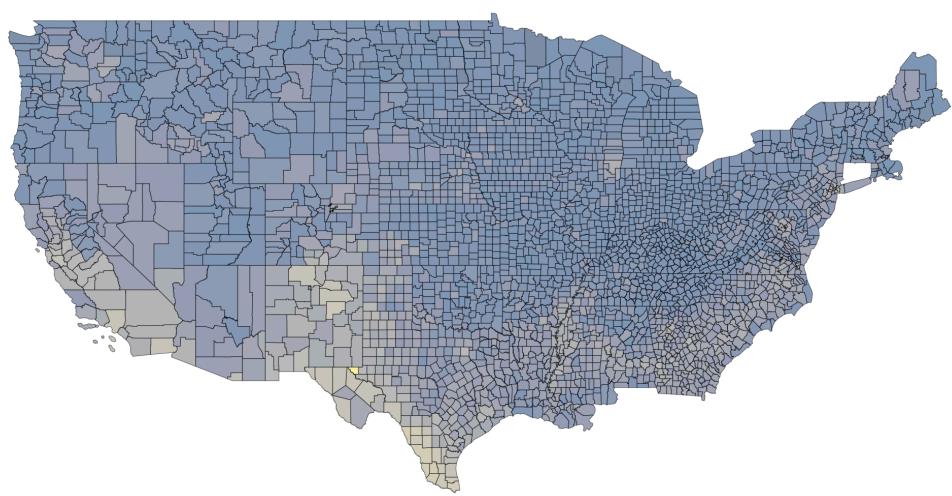
Prevalence Rate

BPMED_AdjPrev - US Map by County

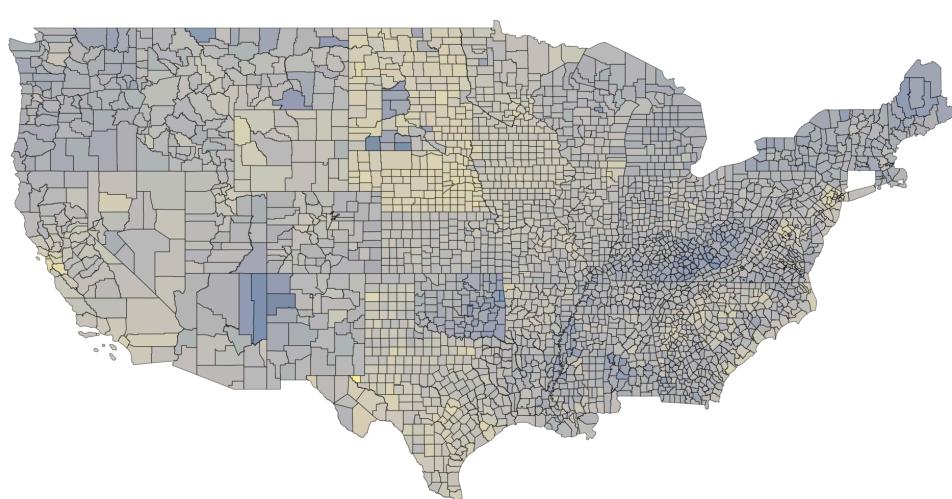


Prevalence Rate

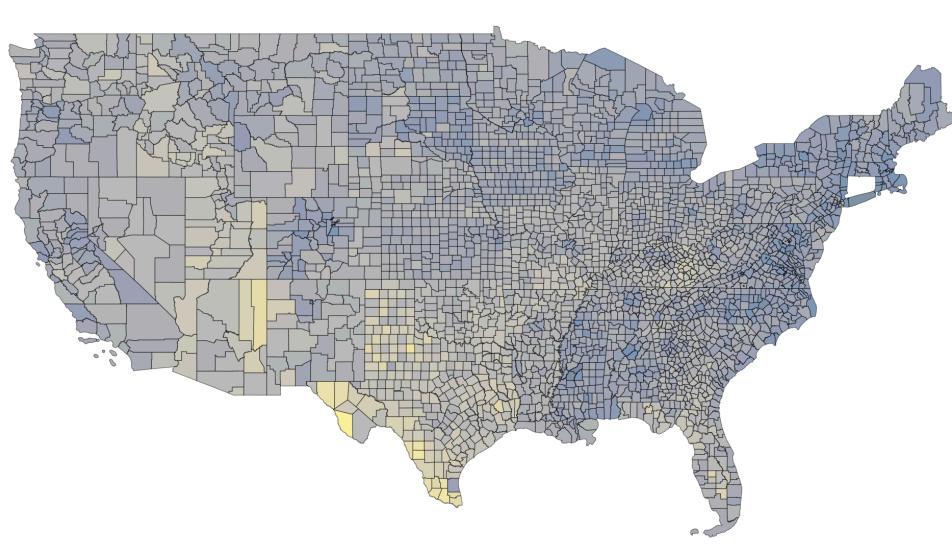
CANCER_AdjPrev - US Map by County



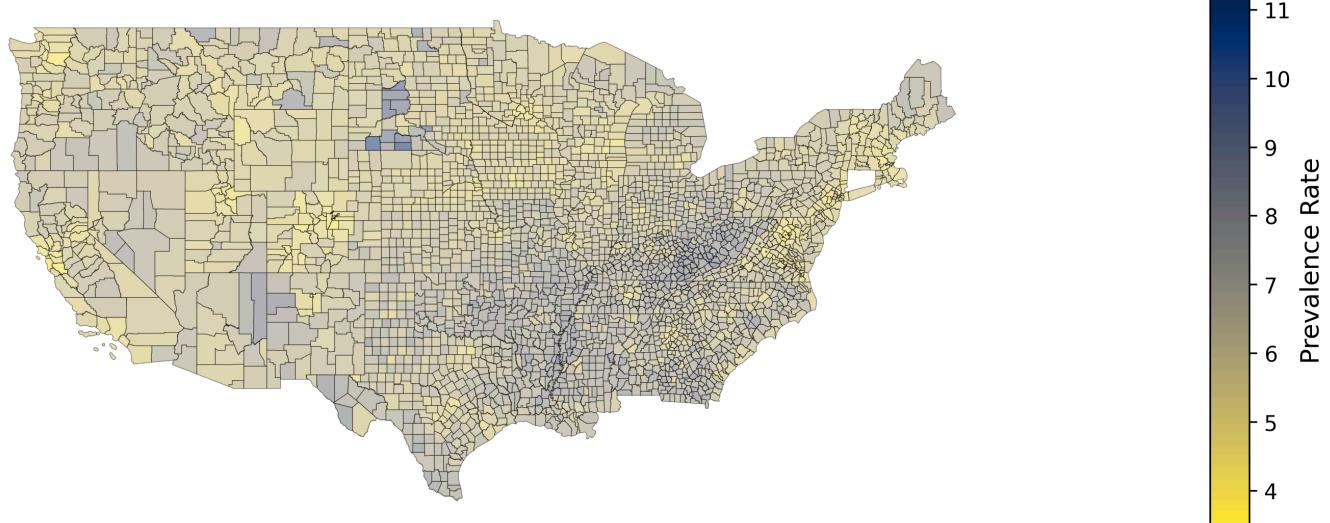
CASTHMA_AdjPrev - US Map by County



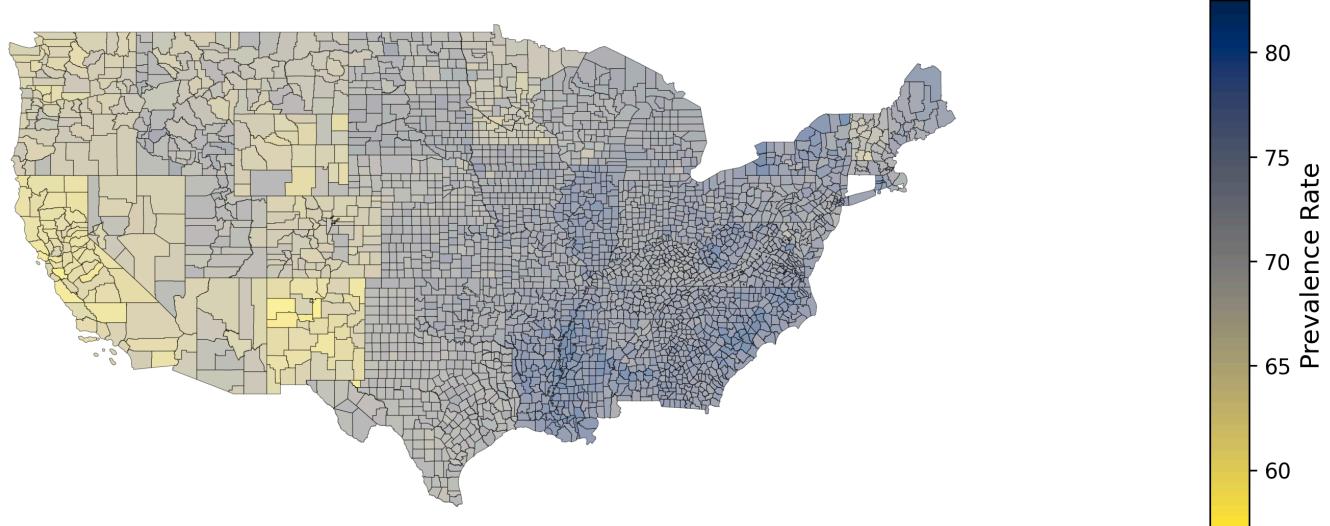
CERVICAL_AdjPrev - US Map by County



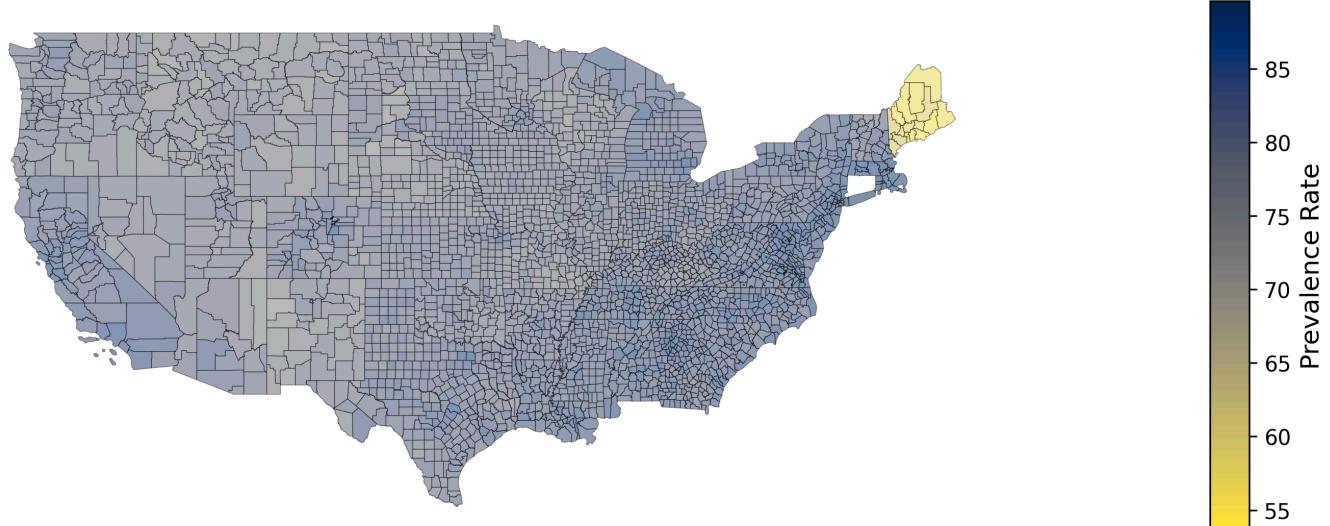
CHD_AdjPrev - US Map by County



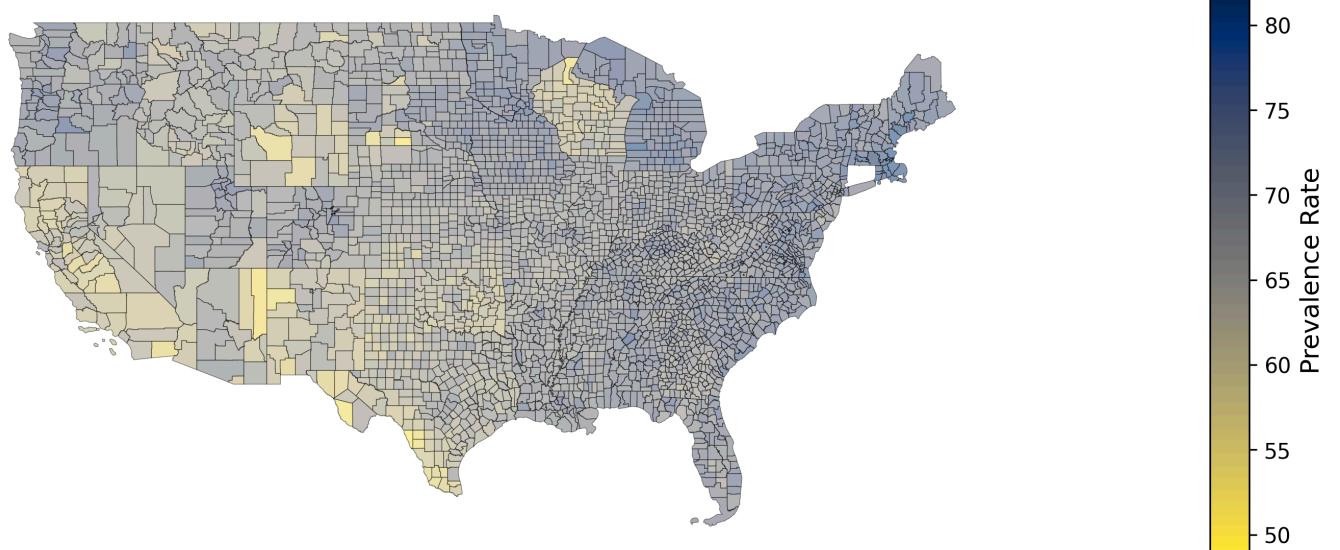
CHECKUP_AdjPrev - US Map by County



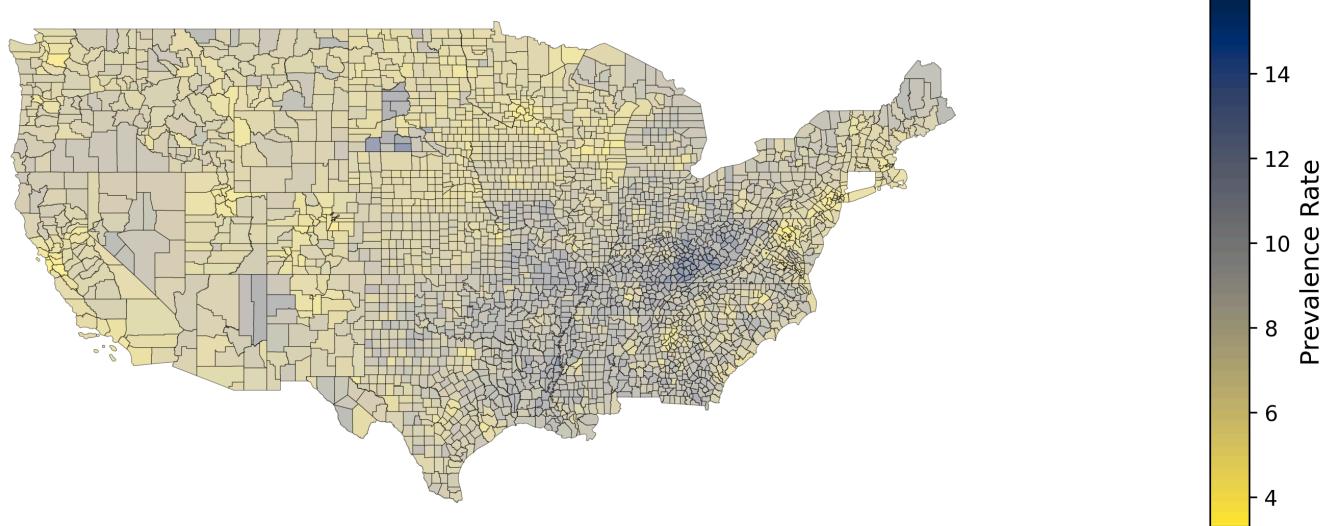
CHOLSCREEN_AdjPrev - US Map by County



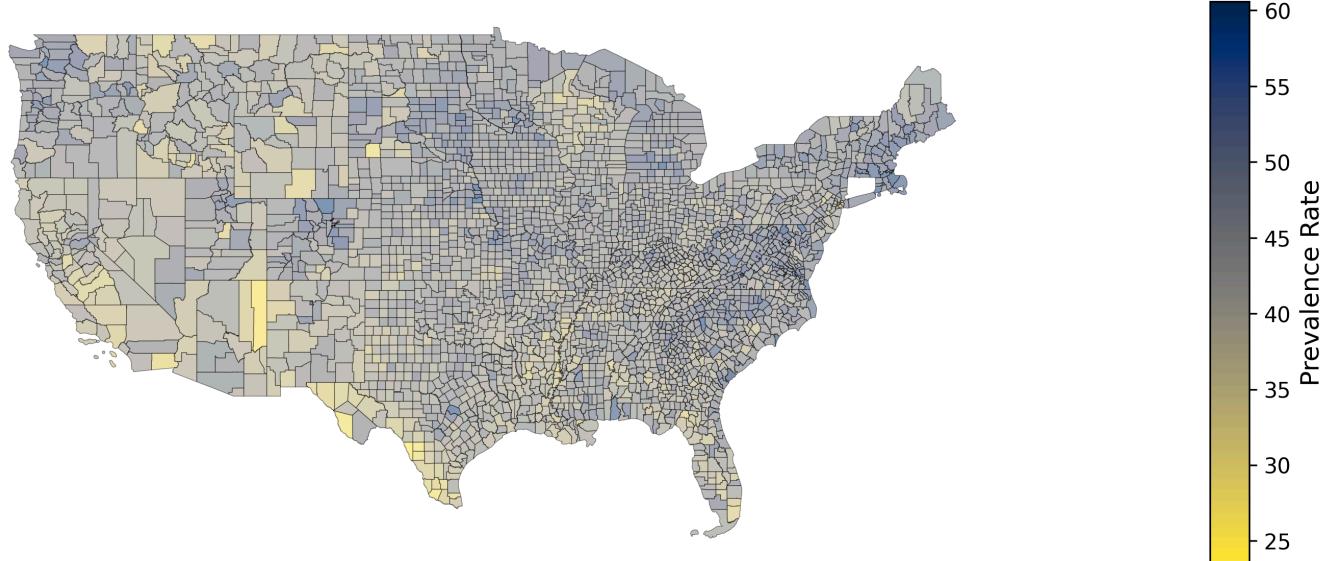
COLON_SCREEN_AdjPrev - US Map by County



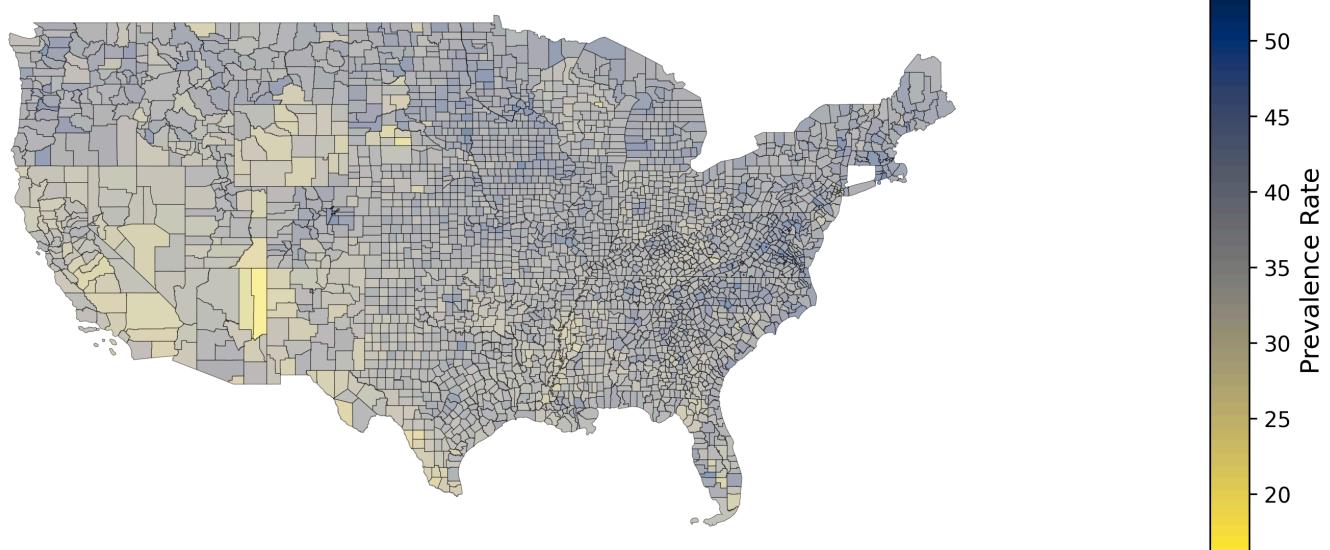
COPD_AdjPrev - US Map by County



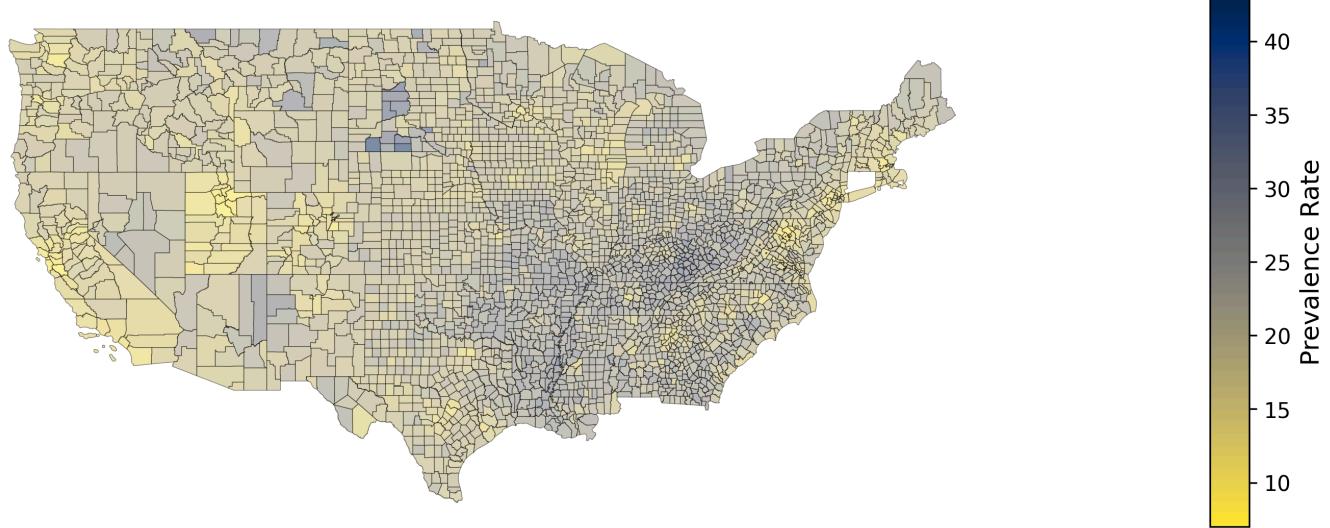
COREM_AdjPrev - US Map by County



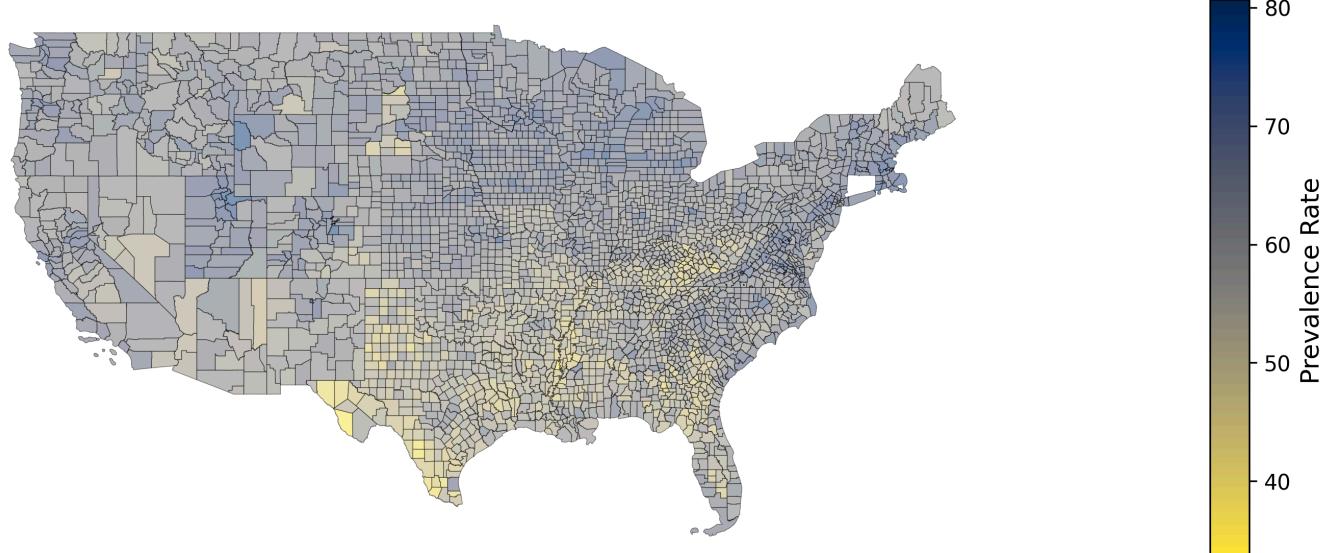
COREW_AdjPrev - US Map by County



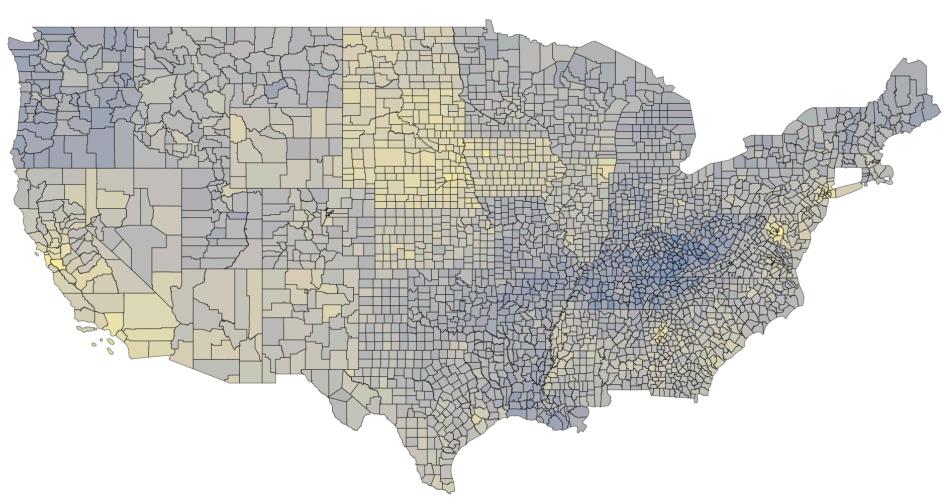
CSMOKING_AdjPrev - US Map by County



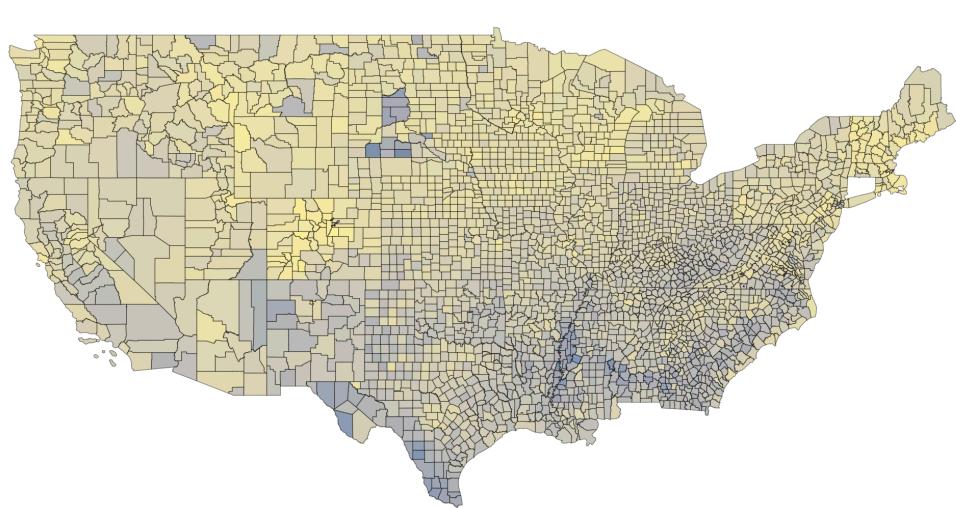
DENTAL_AdjPrev - US Map by County



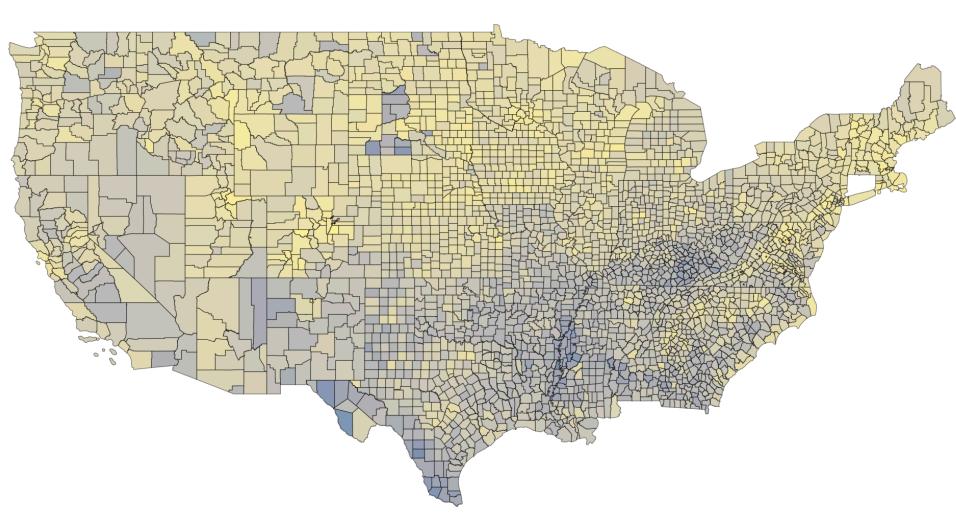
DEPRESSION_AdjPrev - US Map by County



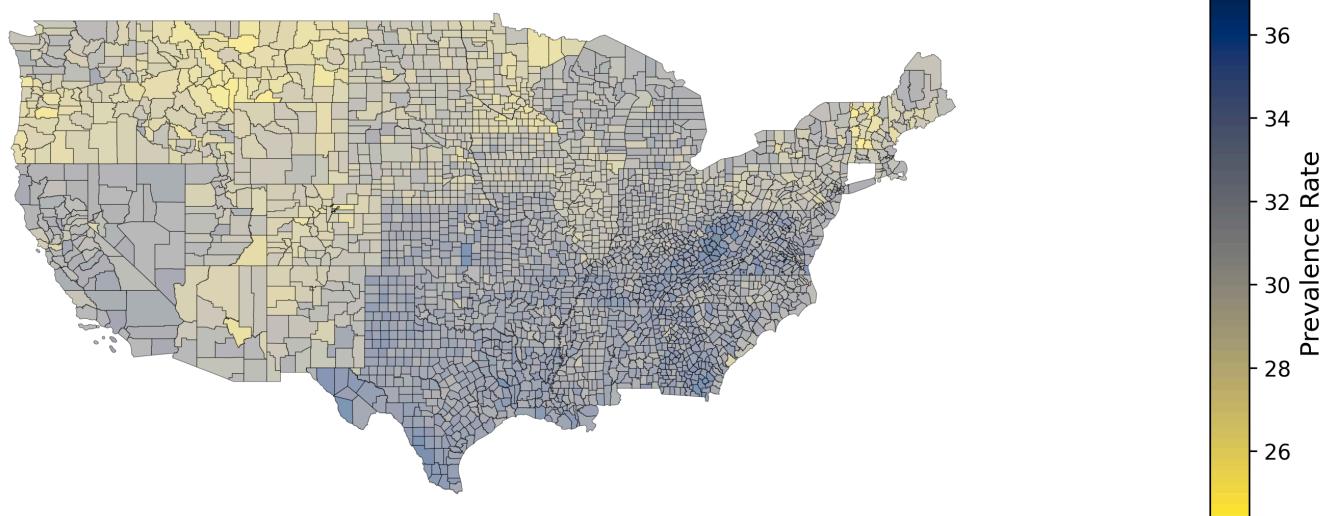
DIABETES_AdjPrev - US Map by County



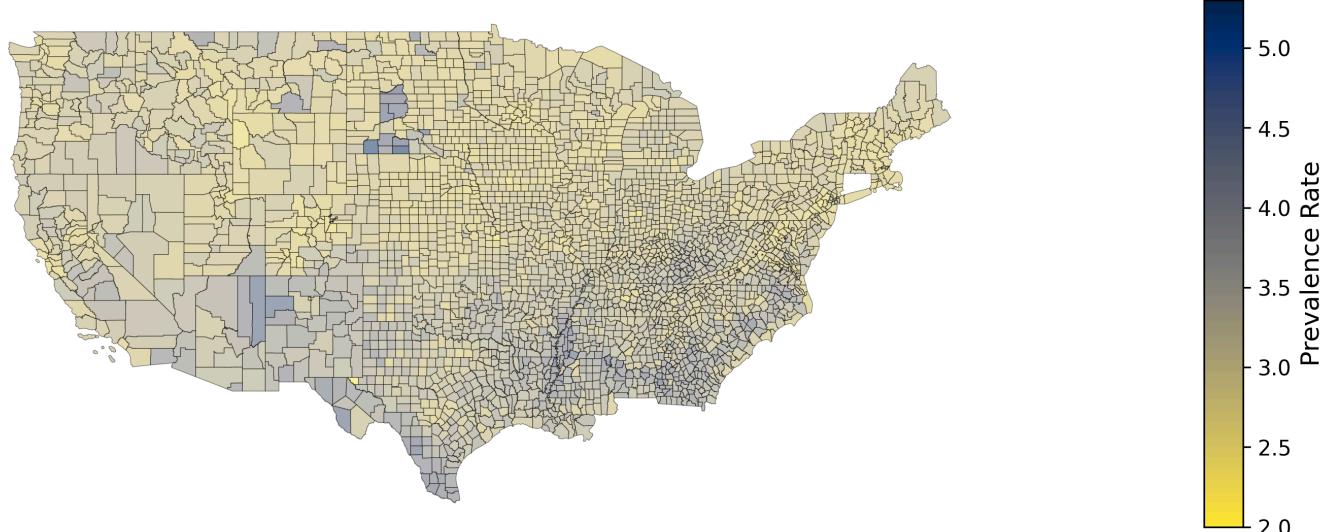
GHLTH_AdjPrev - US Map by County



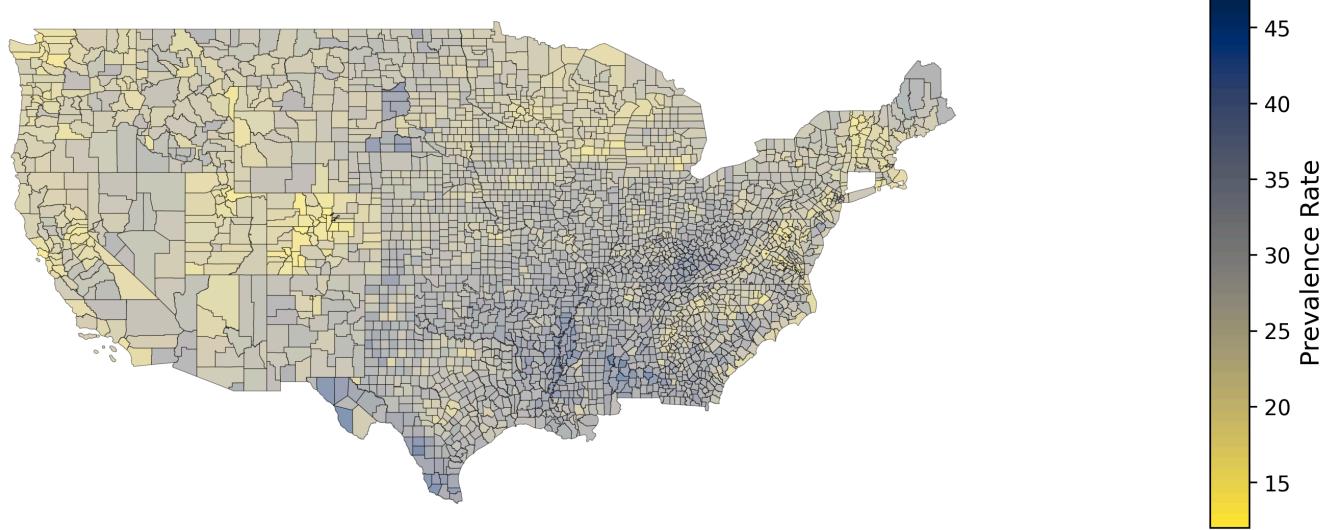
HIGHCHOL_AdjPrev - US Map by County



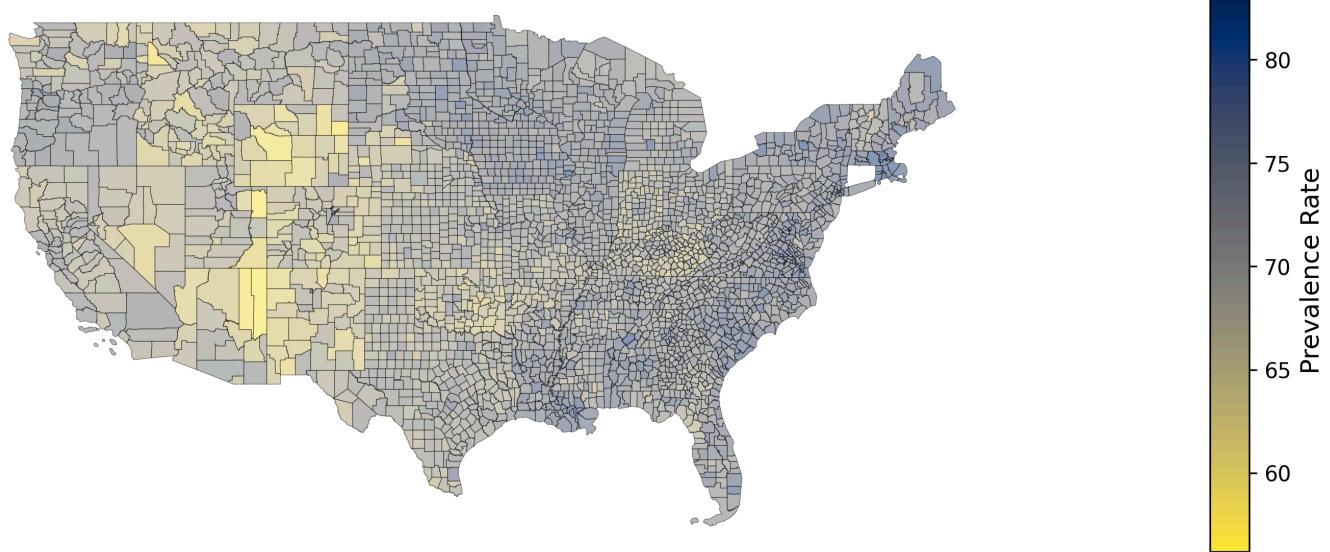
KIDNEY_AdjPrev - US Map by County



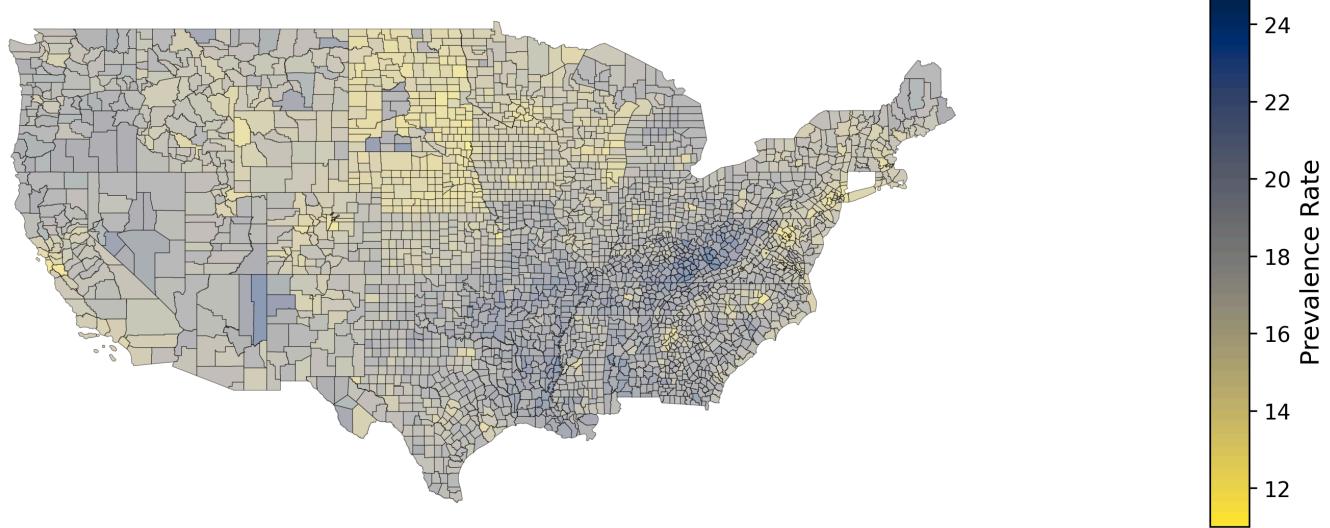
LPA_AdjPrev - US Map by County



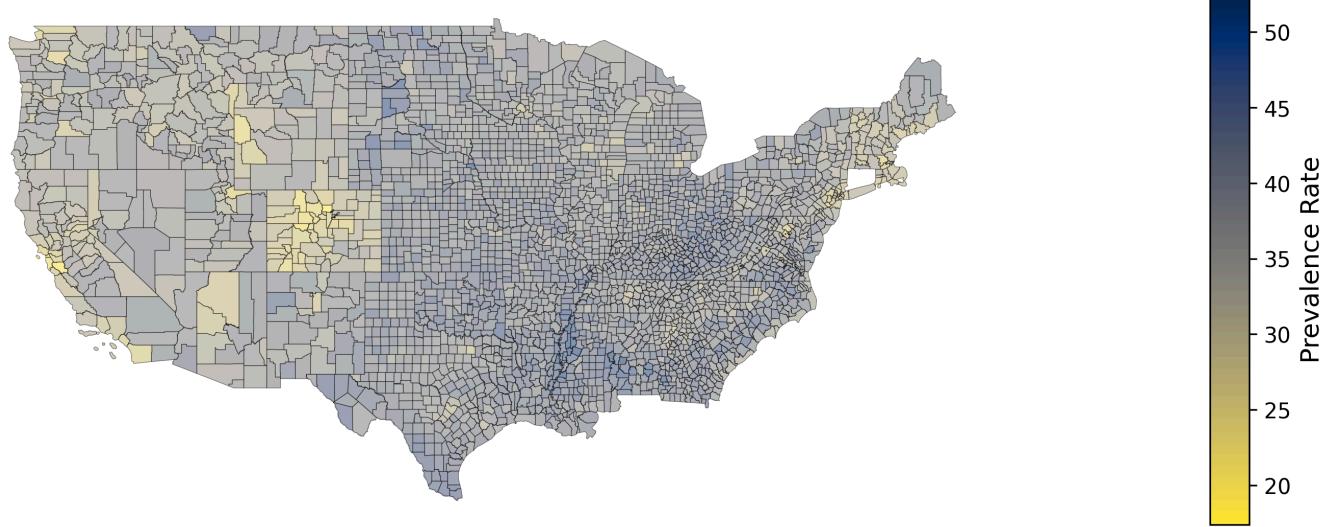
MAMMOUSE_AdjPrev - US Map by County



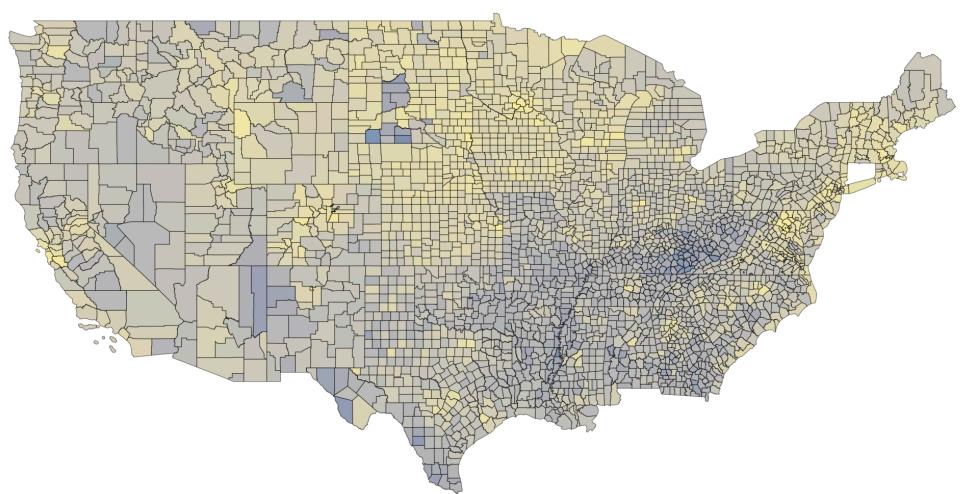
MHLTH_AdjPrev - US Map by County



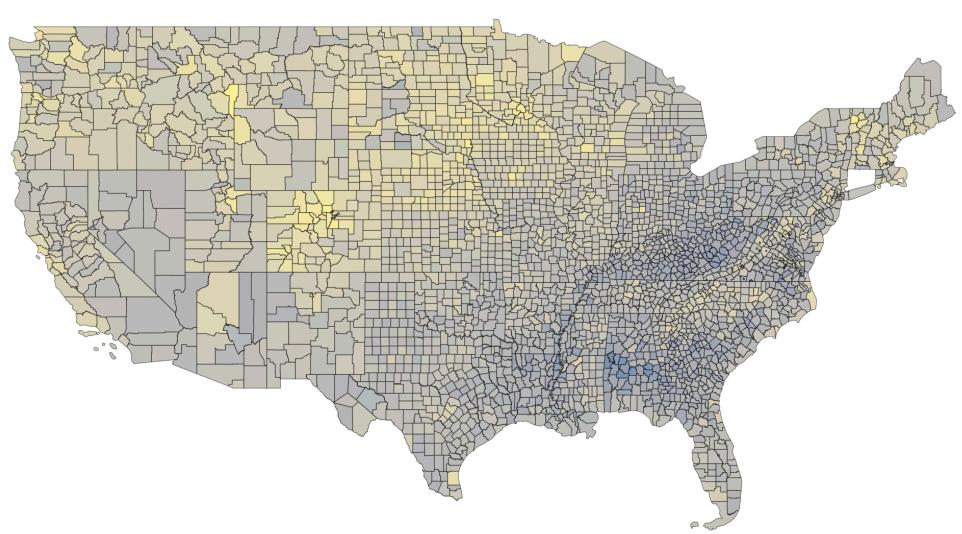
OBESITY_AdjPrev - US Map by County



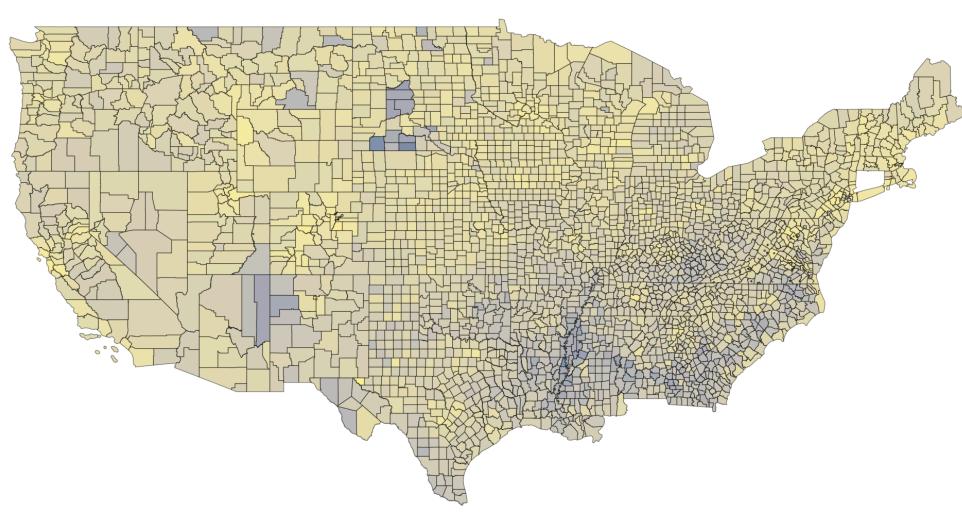
PHLTH_AdjPrev - US Map by County



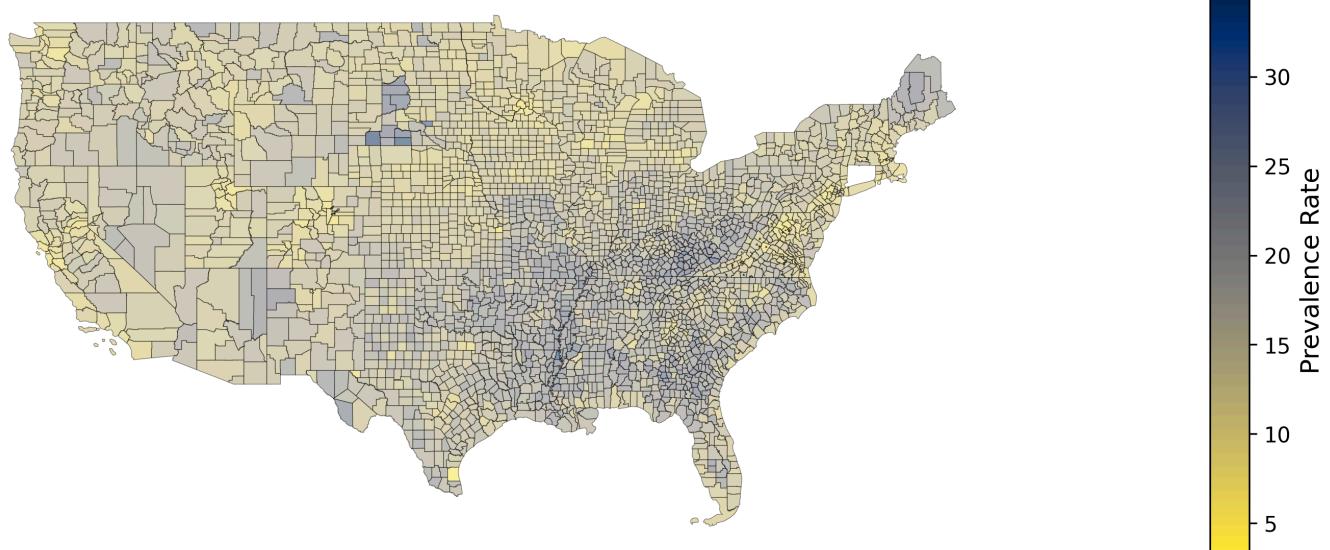
SLEEP_AdjPrev - US Map by County



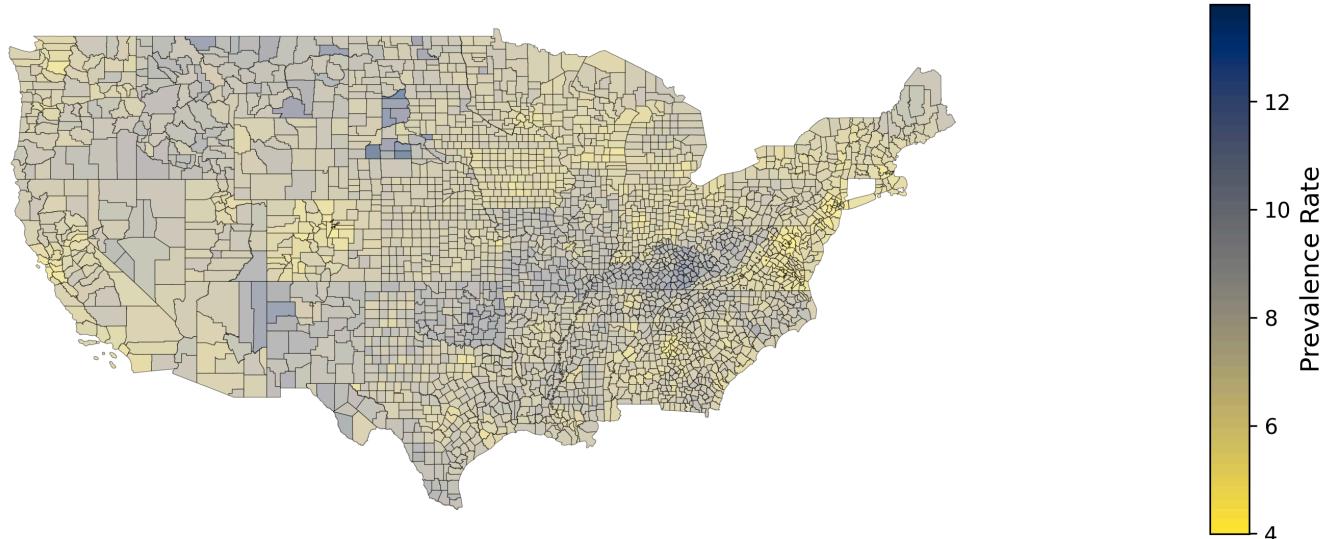
STROKE_AdjPrev - US Map by County



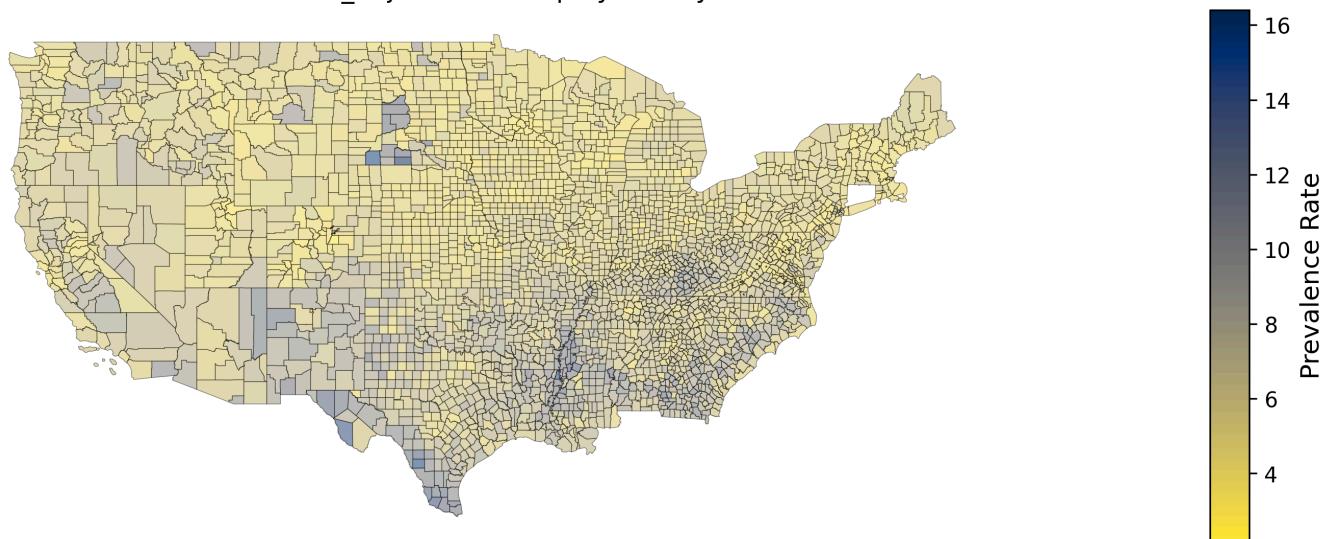
TEETHLOST_AdjPrev - US Map by County



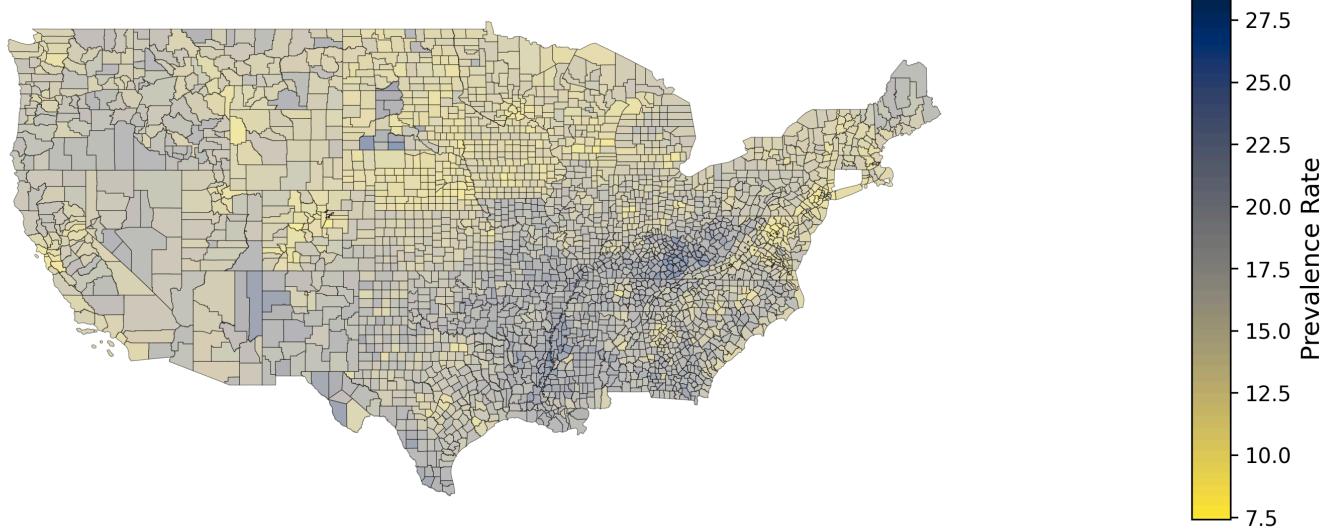
HEARING_AdjPrev - US Map by County



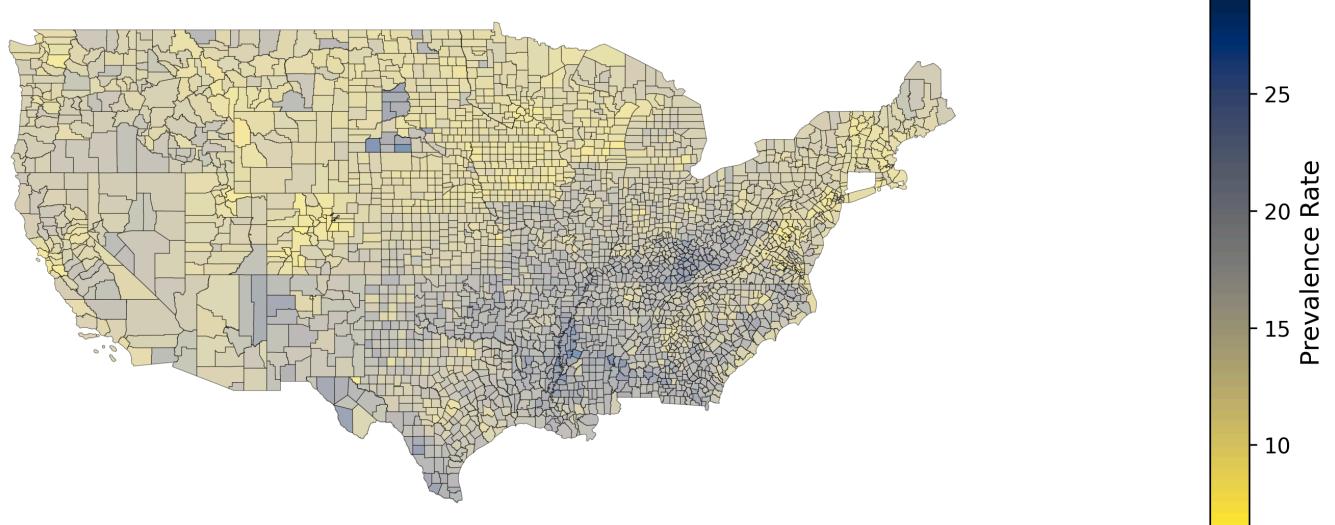
VISION_AdjPrev - US Map by County



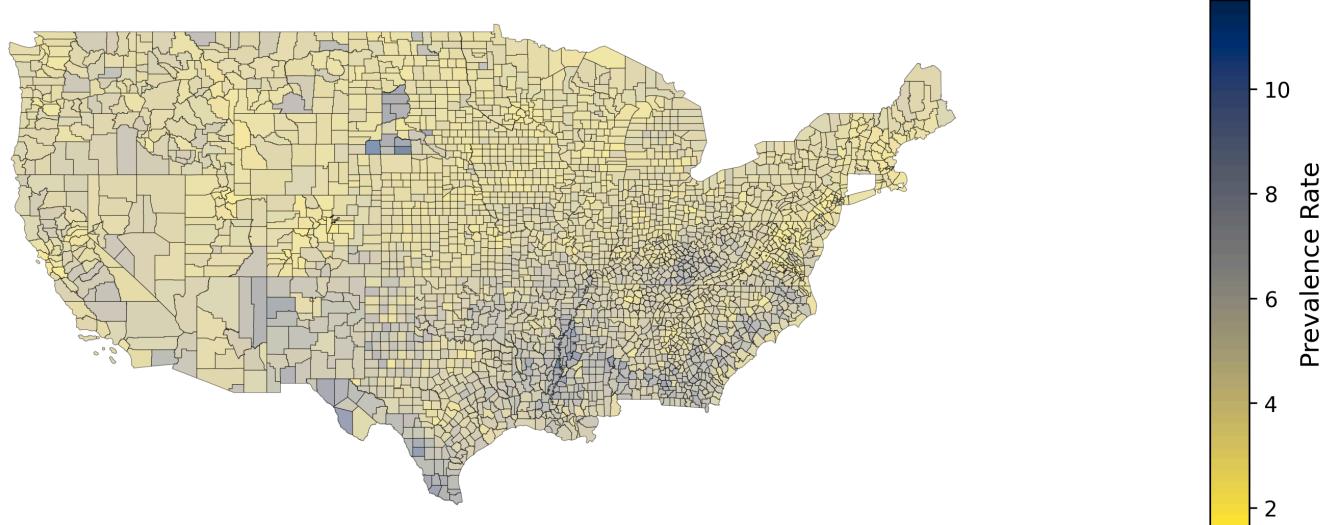
COGNITION_AdjPrev - US Map by County



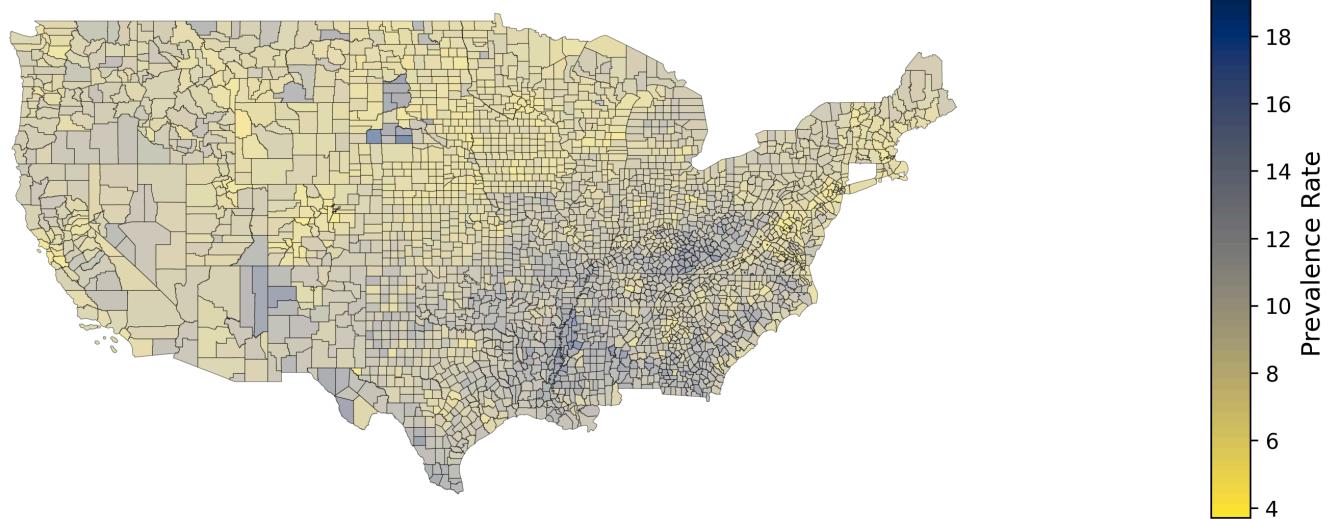
MOBILITY_AdjPrev - US Map by County



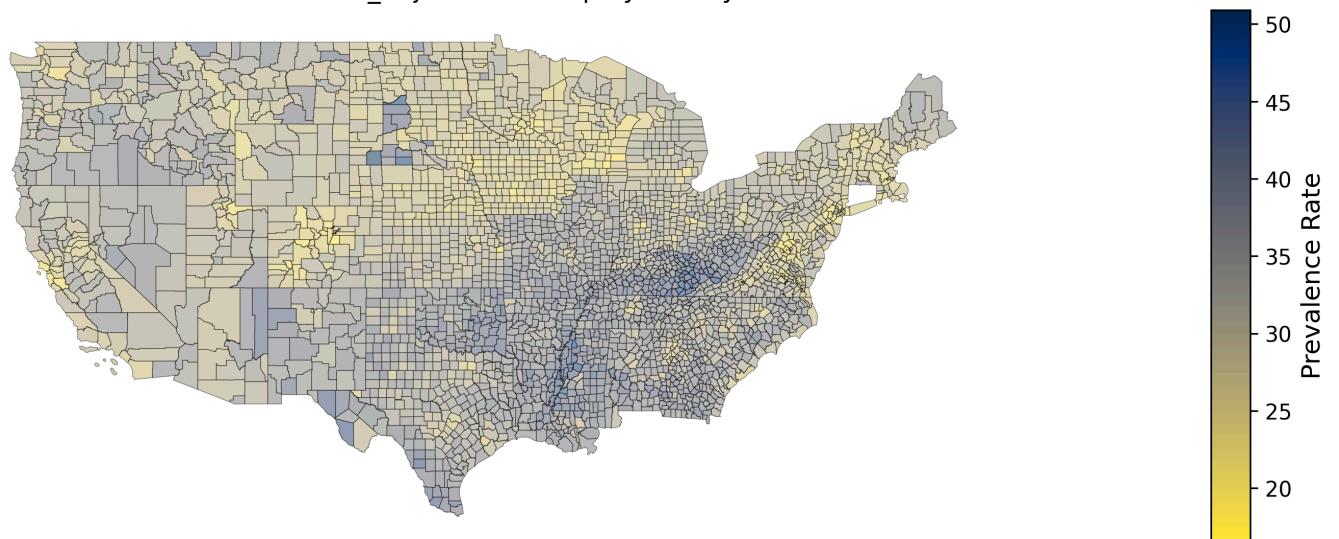
SELCARE_AdjPrev - US Map by County



INDEPLIVE_AdjPrev - US Map by County



DISABILITY_AdjPrev - US Map by County



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