Areal Data Overview

Areal Data

Defining features:

Election results

Map	Percent	Candidate	Party	Votes	Winner
●	**50.3%**	**Jon Tester***	**Dem**	**253,876**	✓
●	46.8%	Matt Rosendale	GOP	235,963	
●	2.9%	Other		14,545	

100% of precincts reporting (669/669) *Incumbent

504,384 total votes

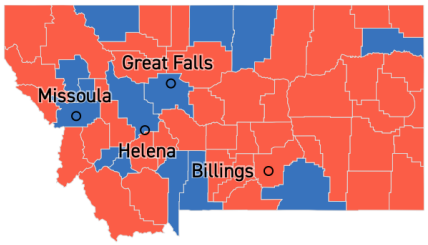


Figure 1: source: https://www.politico.com/election-results/2018/montana/

How can spatial information be incorporated with this data structure?

Choropleth Tutorial What are the objects `urbnmapr::states` and `urbnmapr::counties`?

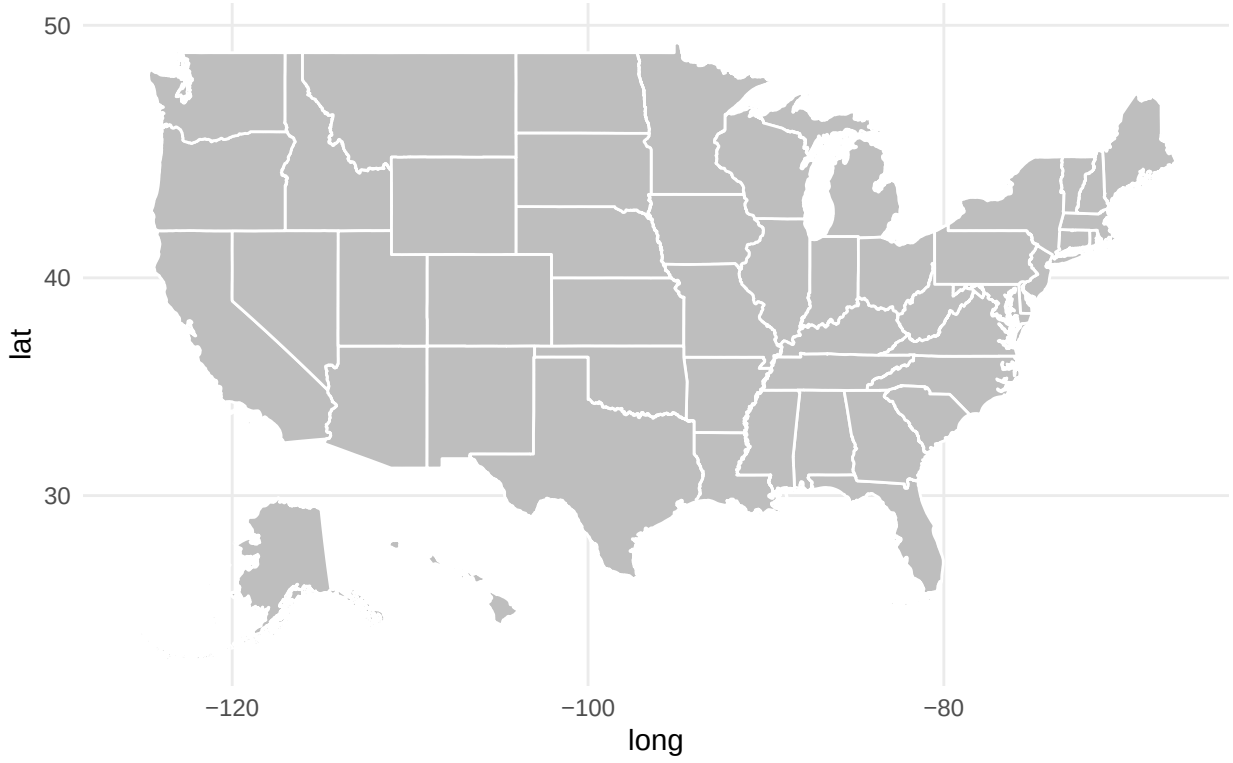
```
urbnmapr::states
```

```
## # A tibble: 83,933 x 9
##     long   lat order hole  piece group state_fips state_abbv state_name
##    <dbl> <dbl> <int> <lgl> <fct> <fct> <chr>      <chr>      <chr>
##  1 -88.5  31.9     1 FALSE 1     01.1  01         AL         Alabama
##  2 -88.5  31.9     2 FALSE 1     01.1  01         AL         Alabama
##  3 -88.5  31.9     3 FALSE 1     01.1  01         AL         Alabama
##  4 -88.5  32.0     4 FALSE 1     01.1  01         AL         Alabama
##  5 -88.5  32.0     5 FALSE 1     01.1  01         AL         Alabama
##  6 -88.5  32.1     6 FALSE 1     01.1  01         AL         Alabama
##  7 -88.4  32.2     7 FALSE 1     01.1  01         AL         Alabama
##  8 -88.4  32.2     8 FALSE 1     01.1  01         AL         Alabama
##  9 -88.4  32.2     9 FALSE 1     01.1  01         AL         Alabama
## 10 -88.4  32.3    10 FALSE 1     01.1  01         AL         Alabama
## # ... with 83,923 more rows
```

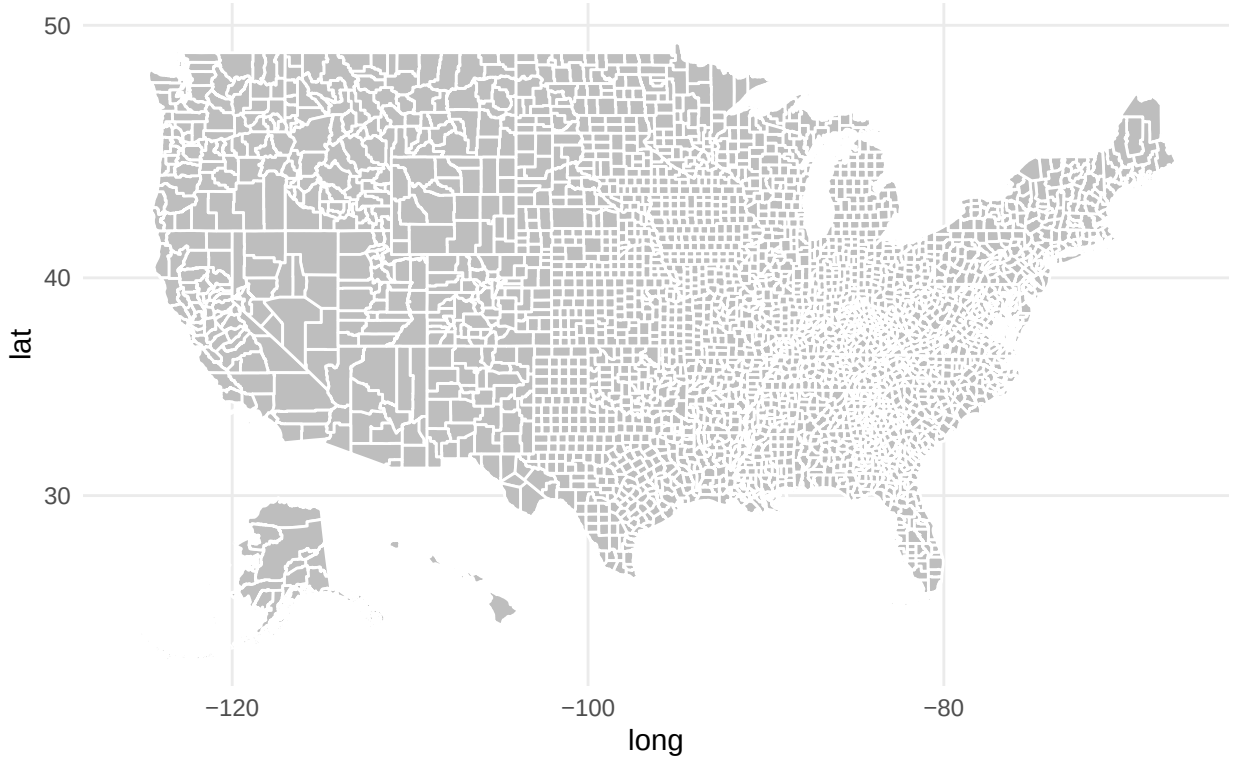
```
urbnmapr::counties
```

```
## # A tibble: 208,874 x 12
##     long   lat order hole  piece group   county_fips state_abbv state_fips
##    <dbl> <dbl> <int> <lgl> <fct> <fct>   <chr>       <chr>      <chr>
##  1 -86.9  32.7     1 FALSE 1     01001.1 01001       AL         01
##  2 -86.8  32.7     2 FALSE 1     01001.1 01001       AL         01
##  3 -86.7  32.7     3 FALSE 1     01001.1 01001       AL         01
##  4 -86.7  32.7     4 FALSE 1     01001.1 01001       AL         01
##  5 -86.4  32.7     5 FALSE 1     01001.1 01001       AL         01
##  6 -86.4  32.4     6 FALSE 1     01001.1 01001       AL         01
##  7 -86.4  32.4     7 FALSE 1     01001.1 01001       AL         01
##  8 -86.5  32.4     8 FALSE 1     01001.1 01001       AL         01
##  9 -86.5  32.4     9 FALSE 1     01001.1 01001       AL         01
## 10 -86.5  32.4    10 FALSE 1     01001.1 01001       AL         01
## # ... with 208,864 more rows, and 3 more variables: county_name <chr>,
## #   fips_class <chr>, state_name <chr>
```

```
ggplot() +
  geom_polygon(data = urbnmapr::states,
               mapping = aes(x = long, y = lat, group = group), fill = "grey", color = "white") +
  coord_map(projection = "mercator") +
  theme_minimal()
```



```
ggplot() +
  geom_polygon(data = urbnmapr::counties,
               mapping = aes(x = long, y = lat, group = group), fill = "grey", color = "white") +
  coord_map(projection = "mercator") +
  theme_minimal()
```



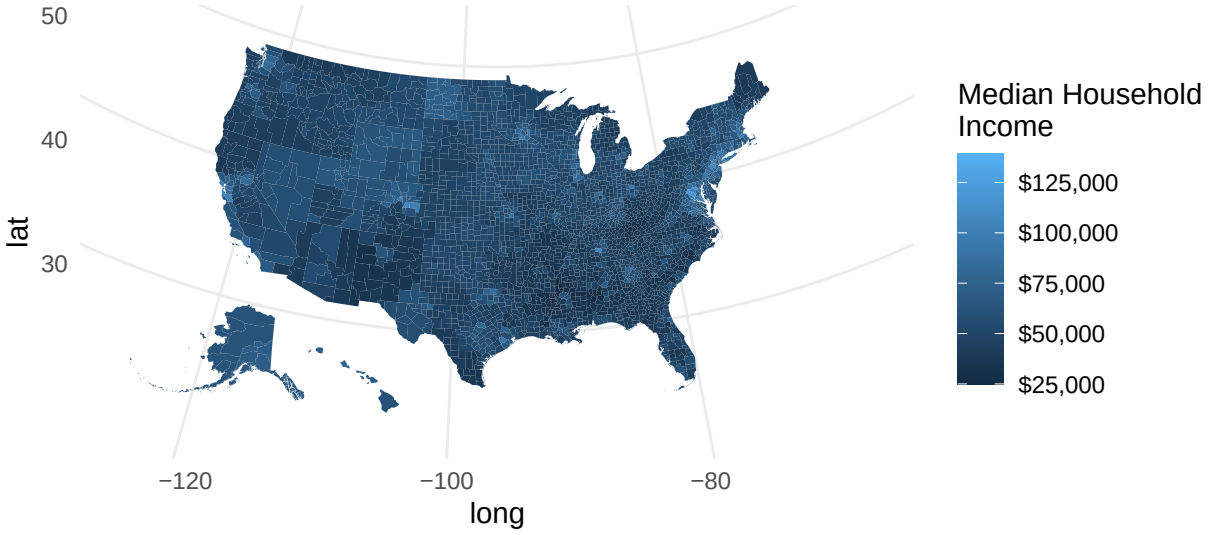
What is `urbnmapr::countydata`?

```
urbnmapr::countydata
```

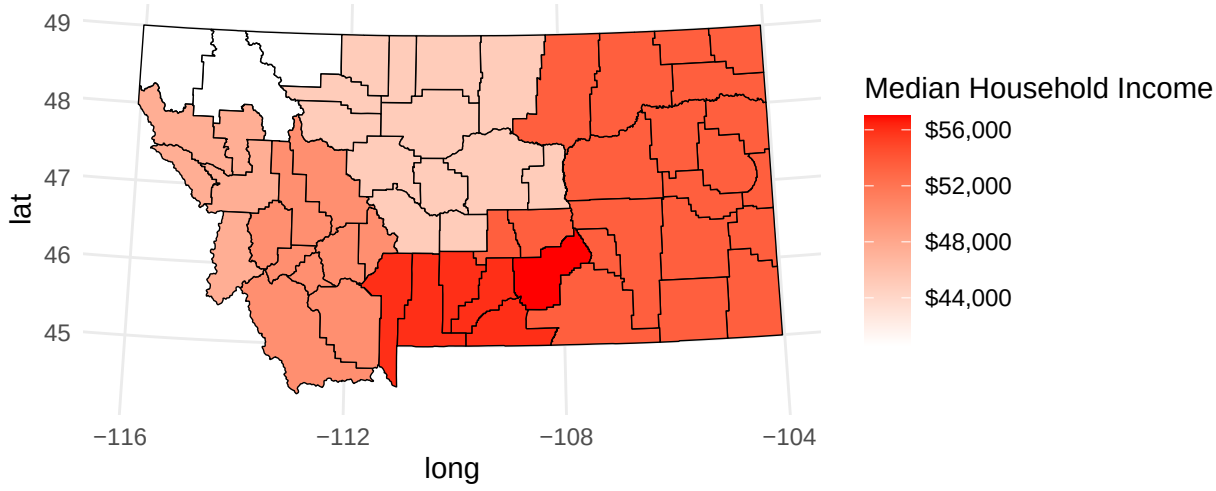
```
## # A tibble: 3,142 x 5
##     year county_fips  hhpop horate medhhincome
##    <int> <chr>        <dbl>  <dbl>       <int>
##  1  2015 01001       20237.  0.746       52200
##  2  2015 01003       72269   0.733       53600
##  3  2015 01005       10287.  0.587       32400
##  4  2015 01007        8198.  0.687       26000
##  5  2015 01009       21094.  0.832       53000
##  6  2015 01011        4104.  0.587       32400
##  7  2015 01013        7859.  0.686       37900
##  8  2015 01015       44323   0.696       42880
##  9  2015 01017       12987.  0.728       37300
## 10  2015 01019       10181.  0.713       37800
## # ... with 3,132 more rows
```

```
household_data <- left_join(urbnmapr::countydata, urbnmapr::counties, by = "county_fips")
```

```
household_data %>%
  ggplot(aes(long, lat, group = group, fill = medhhincome)) +
  geom_polygon(color = NA) +
  coord_map(projection = "albers", lat0 = 39, lat1 = 45) +
  labs(fill = "Median Household \nIncome") +
  theme_minimal() +
  scale_fill_gradient(labels = scales::dollar,
                      guide = guide_colorbar(title.position = "top"))
```



```r
countydata %>%
  left_join(counties, by = "county_fips") %>%
  filter(state_name =="Montana") %>%
  ggplot(mapping = aes(long, lat, group = group, fill = medhhincome)) +
  geom_polygon(color = "black", size = .25) +
  coord_map(projection = "albers", lat0 = 39, lat1 = 45) +
  theme(legend.title = element_text(),
        legend.key.width = unit(.5, "in")) +
  theme_minimal() +
  labs(fill = "Median Household Income") +
  scale_fill_gradient(labels = scales::dollar,
                      guide = guide_colorbar(title.position = "top"),
                      low = 'white', high = 'red')
```



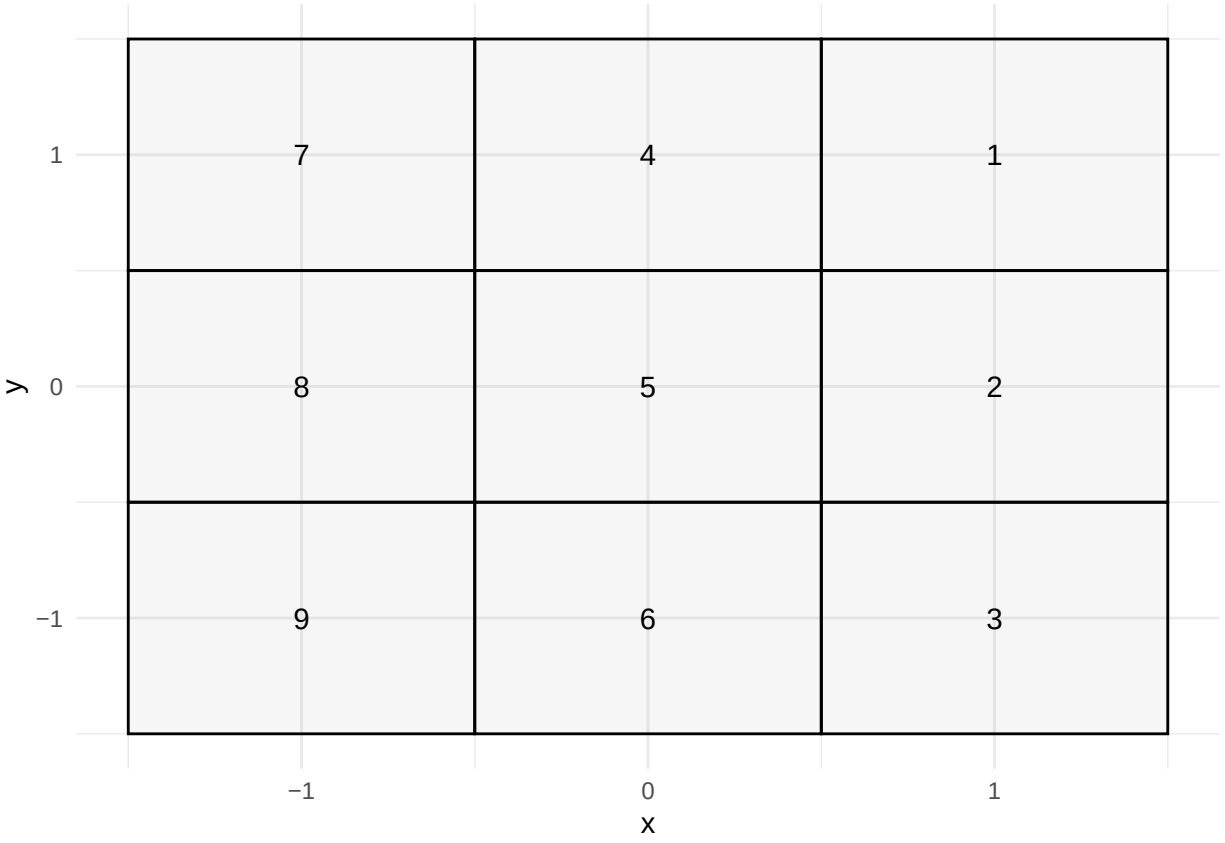
Additional choropleth resources

- Poverty in Nepal with ggplot
- Plotly
- Crime in Philly
- State and County Population
- Leaflet tutorial for creating choropleths.

Proximity Matrix Similar to the distance matrix with point-reference data, a proximity matrix W is used to model areal data.

Grid Example Create an adjacency matrix with diagonal neigbors

Create an adjacency matrix without diagonal neigbors



Spatial Association

There are two common statistics used for assessing spatial association: Moran's I and Geary's C.

Moran's I

$$I = \frac{n \sum_i \sum_j w_{ij}(Y_i - \bar{Y})(Y_j - \bar{Y})}{(\sum_{i \neq j} w_{ij}) \sum_i (Y_i - \bar{Y})^2}$$

Geary's C

$$C = \frac{(n-1) \sum_i \sum_j w_{ij}(Y_i - Y_j)^2}{2(\sum_{i \neq j} w_{ij}) \sum_i (Y_i - \bar{Y})^2}$$