DATA 607 - Final Project

Adam Douglas

12/9/2018

Goal

The goal of this project is to look at rates of opioid overdoses within New York state and compare to a variety of socioeconomic factors to see if there is a correlation between them.

Introduction

The opioid crisis is a well-documented public health crisis. It seems that one only needs to turn on the television or radio to hear something about how the rate of overdose deaths due to opioids (e.g. oxycodone, heroin, etc.) has been on the rise for years. Meanwhile local, state, and federal governments have struggled to find solutions to the crisis.

Drug addiction has traditionally been stigmatized as being a "social disease", one which causes embarassment to both individuals and their families. Addiction has also been associated with homelessness and poverty, portrayed as something that happens to the lower income levels. Only recently has that perception started to change.

Data Sources

Data for this analysis comes in three parts, each broken out by county: data on number of overdose deaths, unemployment data, and data on poverty and income.

Overdose Deaths

New York State tracks opioid overdose deaths by county on their website[1]. They offer data for years 2013, 2014, and 2015 by county. To get this data we need to scrape it from the site.

```
library(rvest)
rawHTML <- read_html("https://www.health.ny.gov/statistics/opioid/data/d2.htm")
```

Once we have the raw HTML, we can then parse it to get our raw data:

```
# Get the table from the HTML
rawTable <- html_table(rawHTML)[[1]]

### Fix up the data frame ###
# Fix column names
colnames(rawTable) <- c("county","deaths_2013","deaths_2014",
                        "deaths_2015","total_deaths","avgPop",
                        "rate","adjRate")

# Remove region titles and region totals
opioids <- rawTable[-grep("[[:space:]]?Reg",rawTable$county),]
```

Next we tidy the data:

[1] https://www.health.ny.gov/statistics/opioid/data/d2.htm

```r
# Put years into rows
opioids <- gather(opioids, key="year", value = "deaths",
                  deaths_2013, deaths_2014, deaths_2015)

opioids$year <- as.numeric(str_extract(opioids$year,"[0-9]+"))

# Standardize our counties
opioids$county <- tolower(opioids$county)
opioids$county <- str_replace(opioids$county,"\\.","")

# Fix column types
opioids$total_deaths <- as.numeric(str_replace_all(opioids$total_deaths,"(\\*|,)",""))
opioids$deaths <- as.numeric(str_replace_all(opioids$deaths,"(\\*|,)",""))
opioids$avgPop <- as.numeric(str_replace_all(opioids$avgPop,"(\\*|,)",""))
opioids$rate <- as.numeric(str_replace_all(opioids$rate,"(\\*|,)",""))
opioids$adjRate <- as.numeric(str_replace_all(opioids$adjRate,"(\\*|,)",""))

head(opioids)
```

```
##     county total_deaths  avgPop rate adjRate year deaths
## 1   nassau          446 1357374 11.0    11.5 2013    131
## 2  suffolk          618 1501431 13.7    14.3 2013    209
## 3    bronx          382 1437445  8.9     8.8 2013    105
## 4    kings          486 2616892  6.2     6.1 2013    126
## 5 new york          313 1635648  6.4     5.8 2013     94
## 6   queens          351 2318968  5.0     4.7 2013    120
```

Our variable of interest here is `rate` which is the number of deaths per 100,000 population. Normalizing to this rate allows us to compare counties of different sizes.

Unemployment Data

Next we turn to unemployment rates. We can also get these data from New York State[2]. This time, the data is in a much easier to retrieve CSV format:

```r
unemployment <- read_csv("NY_unemployment.csv")
unemployment
```

```
## # A tibble: 186 x 6
##     year county      meanRate meanLabor meanEmployed meanUnemployed
##    <int> <chr>          <dbl>     <dbl>        <dbl>          <dbl>
##  1  2013 albany          6.06   160808.       151075           9750
##  2  2013 allegany        7.5     23808.        22000           1783.
##  3  2013 bronx          11.8    603450        532425          71050
##  4  2013 broome          7.76    91925         84767.          7158.
##  5  2013 cattaraugus     8.51    38108.        34850           3250
##  6  2013 cayuga          7.37    38858.        35992.          2867.
##  7  2013 chautauqua      8.02    60233.        55400           4825
##  8  2013 chemung         7.88    39600         36475           3133.
##  9  2013 chenango        7.32    24100         22333.          1758.
## 10  2013 clinton         8.31    37167.        34083.          3083.
## # ... with 176 more rows
```

[2] https://data.ny.gov/Economic-Development/Local-Area-Unemployment-Statistics-Beginning-1976/5hyu-bdh8

Luckily these data are already in a tidy format, so we don't need to do anything to them.

Poverty and Income Data

Finally, we gather our poverty and income data from the US Census Bureau[3]. This data is also in a CSV format, which makes importing a bit easier:

```r
rawPoverty <- read_csv(url("https://raw.githubusercontent.com/lysanthus/Data607/master/Final/poverty.csv

rawPoverty$county <- tolower(rawPoverty$county)

rawPoverty
```

```
## # A tibble: 125 x 44
##     year state countyID county   `All Ages SAIPE Pove~ `All Ages in Pover~
##    <int> <int>    <int> <chr>                    <dbl>               <dbl>
##  1  2016    36    36001 albany                  293097               35585
##  2  2013    36    36001 albany                  291194               39857
##  3  2016    36    36003 allegany                 42697                7836
##  4  2013    36    36003 allegany                 43635                7296
##  5  2016    36    36005 bronx                  1418238              405516
##  6  2013    36    36005 bronx                  1381104              423904
##  7  2016    36    36007 broome                  184887               30417
##  8  2013    36    36007 broome                  187458               33205
##  9  2016    36    36009 cattara~                 75207               11014
## 10  2013    36    36009 cattara~                 76451               14442
## # ... with 115 more rows, and 38 more variables: `All Ages in Poverty
## #   Count LB 90%` <dbl>, `All Ages in Poverty Count UB 90%` <dbl>, `90%
## #   Confidence Interval (All Ages in Poverty Count)` <chr>, `All Ages in
## #   Poverty Percent` <dbl>, `All Ages in Poverty Percent LB 90%` <dbl>,
## #   `All Ages in Poverty Percent UB 90%` <dbl>, `90% Confidence Interval
## #   (All Ages in Poverty Percent)` <chr>, `Under Age 18 SAIPE Poverty
## #   Universe` <dbl>, `Under Age 18 in Poverty Count` <dbl>, `Under Age 18
## #   in Poverty Count LB 90%` <dbl>, `Under Age 18 in Poverty Count UB
## #   90%` <dbl>, `90% Confidence Interval (Under Age 18 in Poverty
## #   Count)` <chr>, `Under Age 18 in Poverty Percent` <dbl>, `Under Age 18
## #   in Poverty Percent LB 90%` <dbl>, `Under Age 18 in Poverty Percent UB
## #   90%` <dbl>, `90% Confidence Interval (Under Age 18 in Poverty
## #   Percent)` <chr>, `Ages 5 to 17 in Families SAIPE Poverty
## #   Universe` <dbl>, `Ages 5 to 17 in Families in Poverty Count` <dbl>,
## #   `Ages 5 to 17 in Families in Poverty Count LB 90%` <dbl>, `Ages 5 to
## #   17 in Families in Poverty Count UB 90%` <dbl>, `90% Confidence
## #   Interval (Ages 5 to 17 in Families in Poverty Count)` <chr>, `Ages 5
## #   to 17 in Families in Poverty Percent` <dbl>, `Ages 5 to 17 in Families
## #   in Poverty Percent LB 90%` <dbl>, `Ages 5 to 17 in Families in Poverty
## #   Percent UB 90%` <dbl>, `90% Confidence Interval (Ages 5 to 17 in
## #   Families in Poverty Percent)` <chr>, `Under Age 5 SAIPE Poverty
## #   Universe` <chr>, `Under Age 5 in Poverty Count` <chr>, `Under Age 5 in
## #   Poverty Count LB 90%` <chr>, `Under Age 5 in Poverty Count UB
## #   90%` <chr>, `90% Confidence Interval (Under Age 5 in Poverty
## #   Count)` <chr>, `Under Age 5 in Poverty Percent` <chr>, `Under Age 5 in
## #   Poverty Percent LB 90%` <chr>, `Under Age 5 in Poverty Percent UB
```

[3] https://www.census.gov/data-tools/demo/saipe/saipe.html?s_appName=saipe&map_yearSelector=2013&map_geoSelector=aa_c&s_state=36&s_year=2016,2013

```
## #   90%` <chr>, `90% Confidence Interval (Under Age 5 in Poverty
## #   Percent)` <chr>, `Median Household Income in Dollars` <chr>, `Median
## #   Household Income in Dollars LB 90%` <chr>, `Median Household Income in
## #   Dollars UB 90%` <chr>, `90% Confidence Interval (Median Household
## #   Income in Dollars)` <chr>
```

The CSV contains several variables, however for this analysis we will look at only poverty rate and median incomes for each county.

Also, we have values for 2013 and 2016 only. So we will linearly impute the middle values of 2014 and 2015 as equally distant from 2013 and 2016. We also transform the values to thousands of dollars, to make visualization easier:

```
# 2013 values
pov13 <- rawPoverty %>% filter(year == 2013) %>% select(county,pct = `All Ages in Poverty Percent`, inc

# 2016 values
pov16 <- rawPoverty %>% filter(year == 2016) %>% select(county,pct = `All Ages in Poverty Percent`, inc

# Fix income values by removing $ and ,
pov13$inc <- as.numeric(str_replace_all(pov13$inc,"\\$|,",""))
pov16$inc <- as.numeric(str_replace_all(pov16$inc,"\\$|,",""))

# Combine our data frames
poverty <- inner_join(pov13, pov16, by=c("county" = "county"),
                      suffix = c("_2013","_2016"))

# Compute changes and impute interim values
poverty <- poverty %>%
  mutate(povChg = pct_2016 - pct_2013, incChg = inc_2016 - inc_2013,
         povIncrement = povChg / 3, incIncrement = incChg / 3,
         pct_2014 = pct_2013 + povIncrement, pct_2015 = pct_2016 - povIncrement,
         inc_2014 = inc_2013 + incIncrement, inc_2015 = inc_2016 - incIncrement)

# Tidy the data
poverty <- poverty %>%
  gather(key="year",
         value="value",
         pct_2013,pct_2014,pct_2015,pct_2016,
         inc_2013,inc_2014,inc_2015,inc_2016)

poverty <- poverty %>% separate(year,c("measure","yr"),"_")

poverty <- poverty %>% spread(key="measure",value="value") %>% select(county, year = yr, income = inc, p

poverty$year <- as.numeric(poverty$year)

# Adjust income to 1,000's scale
poverty$income <- round(poverty$income/1000,2)

poverty
```

```
## # A tibble: 248 x 4
##    county       year income poverty
##    <chr>       <dbl>  <dbl>   <dbl>
##  1 albany       2013   55.8    13.7
```

```
## 2 allegany     2013   41.8    16.7
## 3 bronx        2013   33.1    30.7
## 4 broome       2013   45.1    17.7
## 5 cattaraugus  2013   40.9    18.9
## 6 cayuga       2013   49.0    14.2
## 7 chautauqua   2013   40.5    19.1
## 8 chemung      2013   45.3    17
## 9 chenango     2013   44.3    16.8
## 10 clinton     2013   45.9    15.7
## # ... with 238 more rows
```

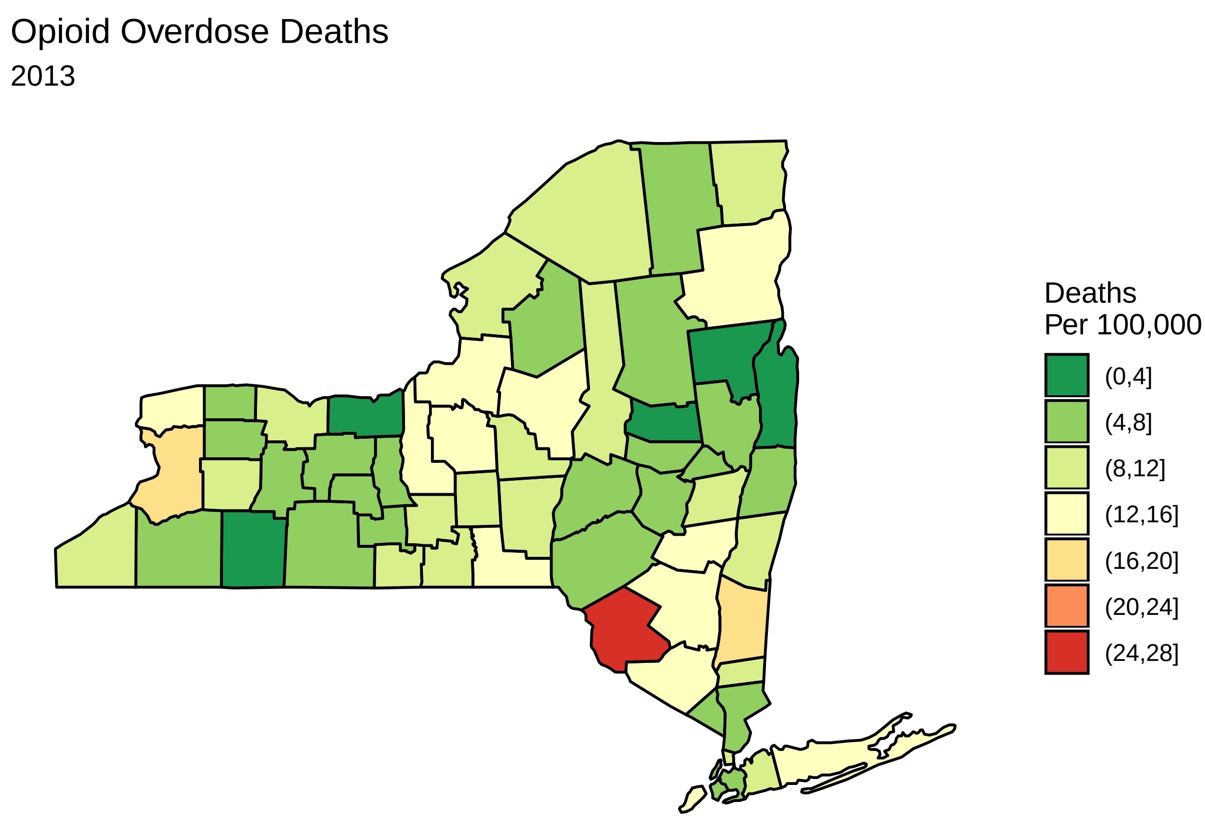
Now our data is tidy and ready to use.

Visualization

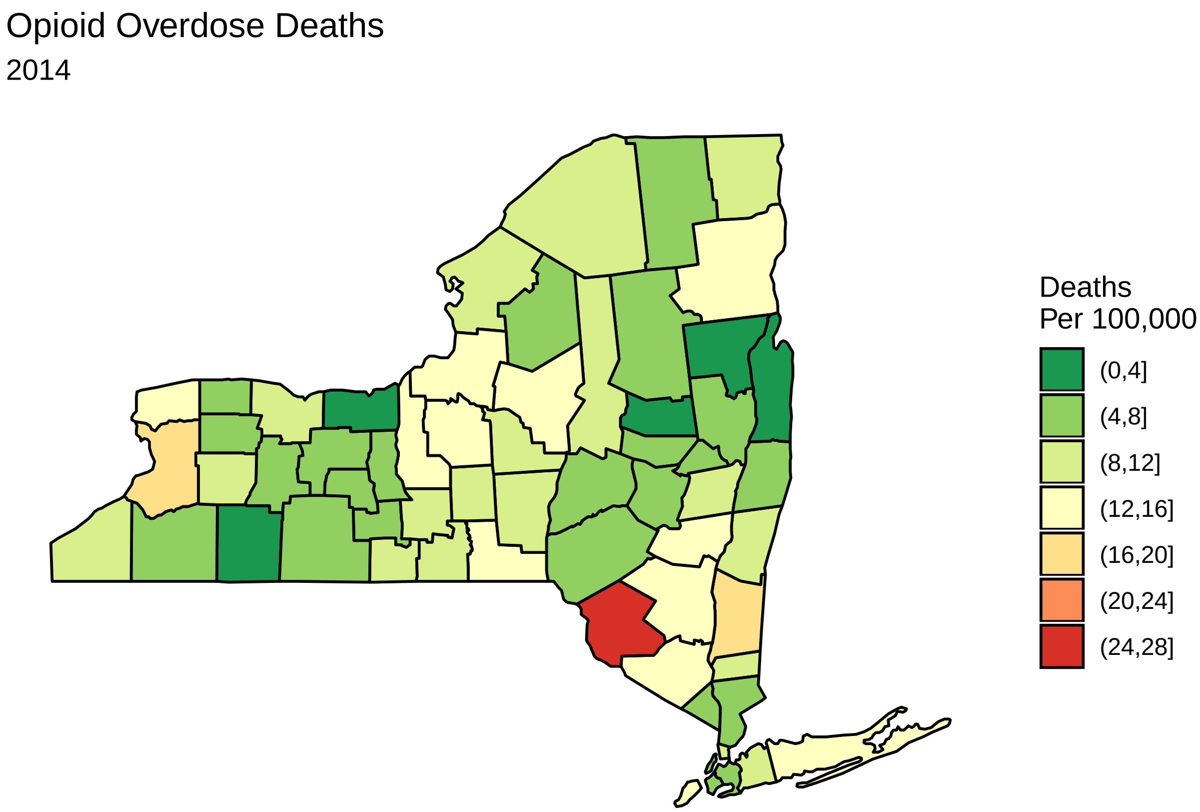
Now that we have all three data sets loaded, let's look at them and see what we patterns we can easily detect.

First, we look at our first variable of interest: overdose deaths, and plot it on a map of New York State:

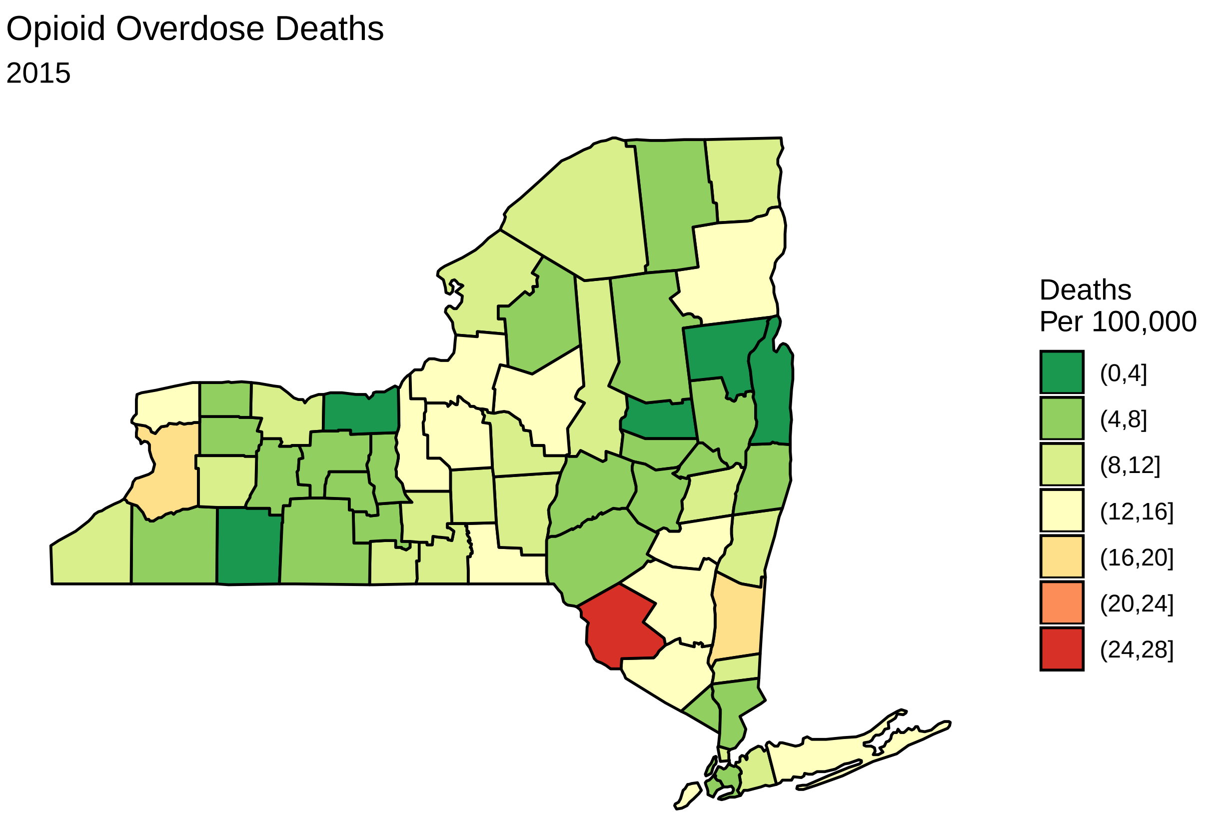
```
## Breaking out data into bins
breaks <- c(0, seq(4,28,by=4))

opioids %>% filter(year == "2013") %>%
  NYMap("rate","Opioid Overdose Deaths","2013",
        "Deaths\nPer 100,000", breaks)
```



```r
opioids %>% filter(year == "2014") %>%
  NYMap("rate","Opioid Overdose Deaths","2014",
        "Deaths\nPer 100,000", breaks)
```



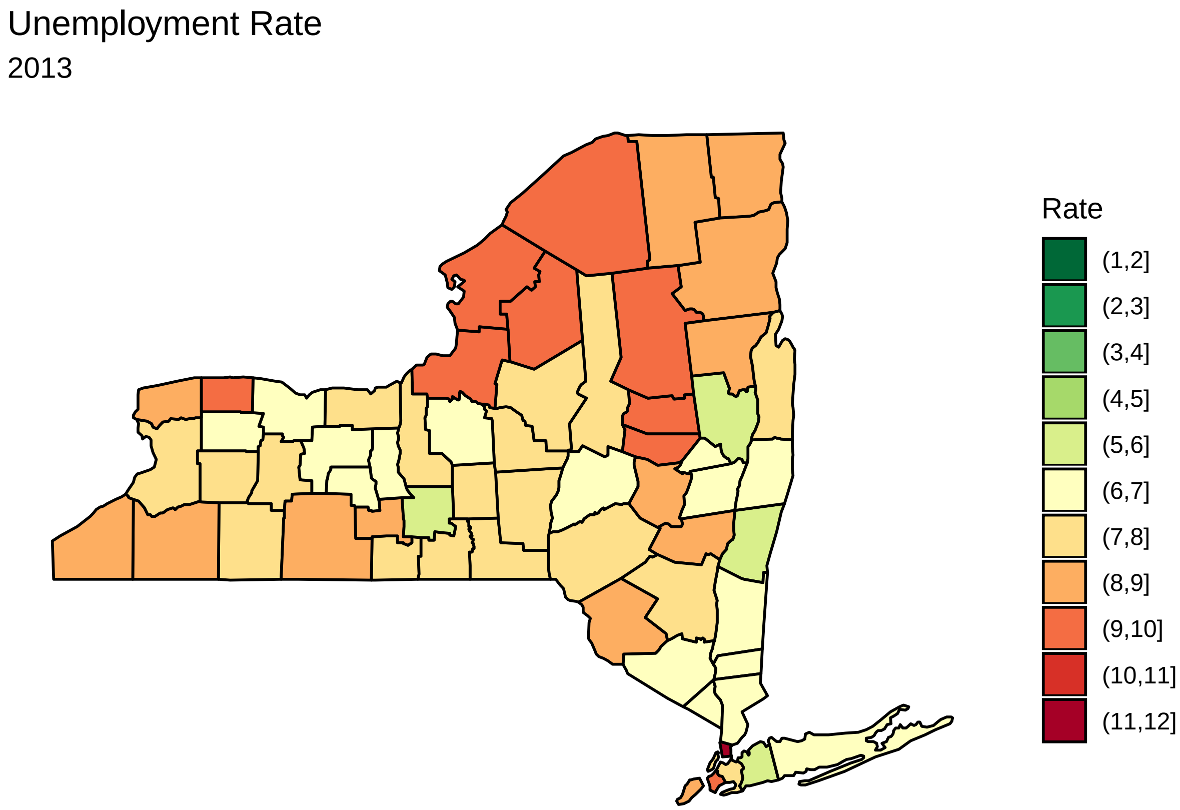

```r
opioids %>% filter(year == "2015") %>%
  NYMap("rate","Opioid Overdose Deaths","2015",
        "Deaths\nPer 100,000", breaks)
```



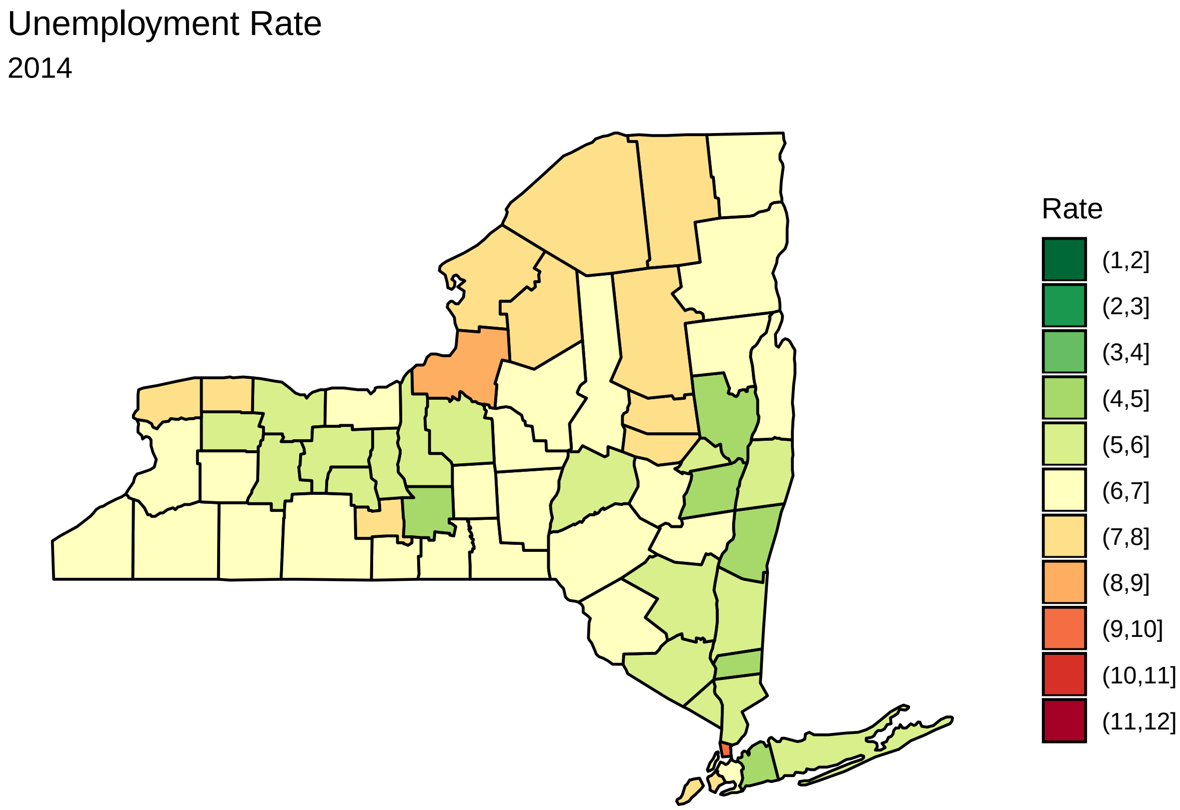
Looking at the maps, there are a few counties with a larger number of overdose deaths than others. Specifically, Sullivan, Erie, and Dutchess counties seem to be some of the worst areas.

Let's do the same for our unemployment data:

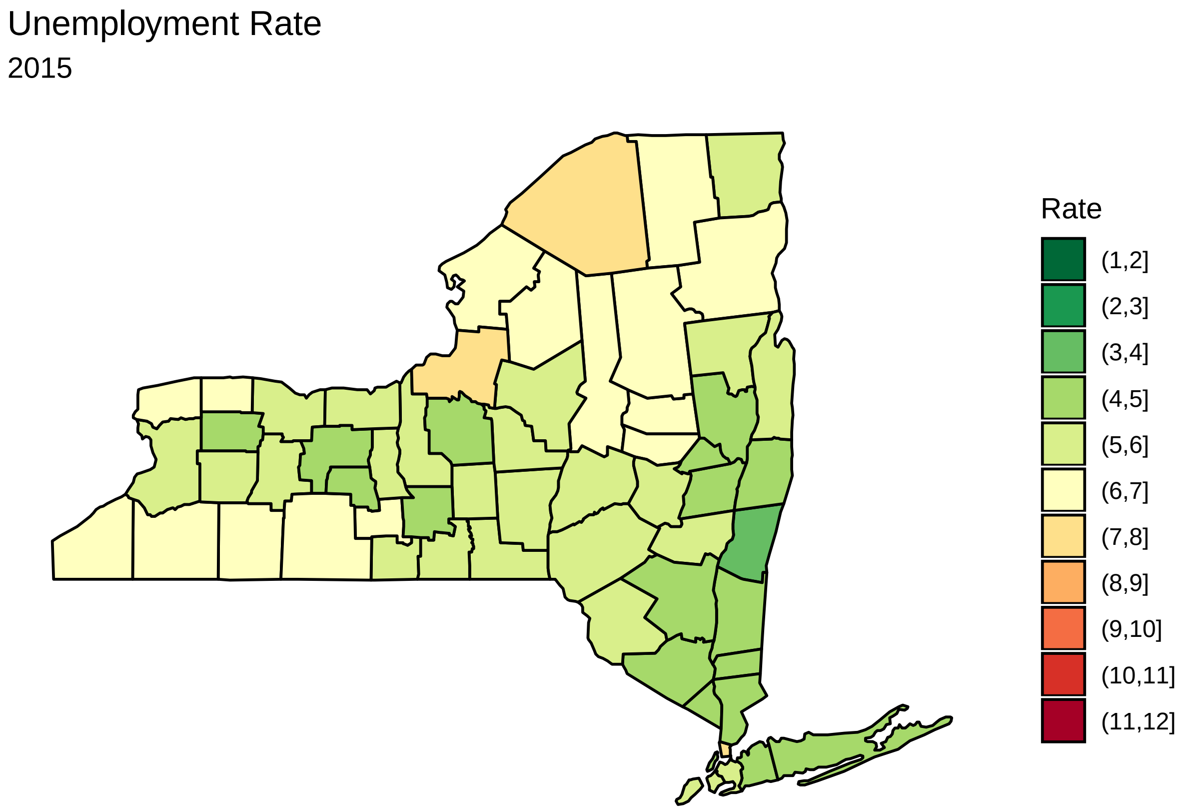
```
breaks <- c(1, seq(2,12,by=1))

unemployment %>% filter(year == "2013") %>%
  NYMap("meanRate","Unemployment Rate","2013",
        "Rate", breaks)
```




```
unemployment %>% filter(year == "2014") %>%
  NYMap("meanRate","Unemployment Rate","2014",
        "Rate", breaks)
```



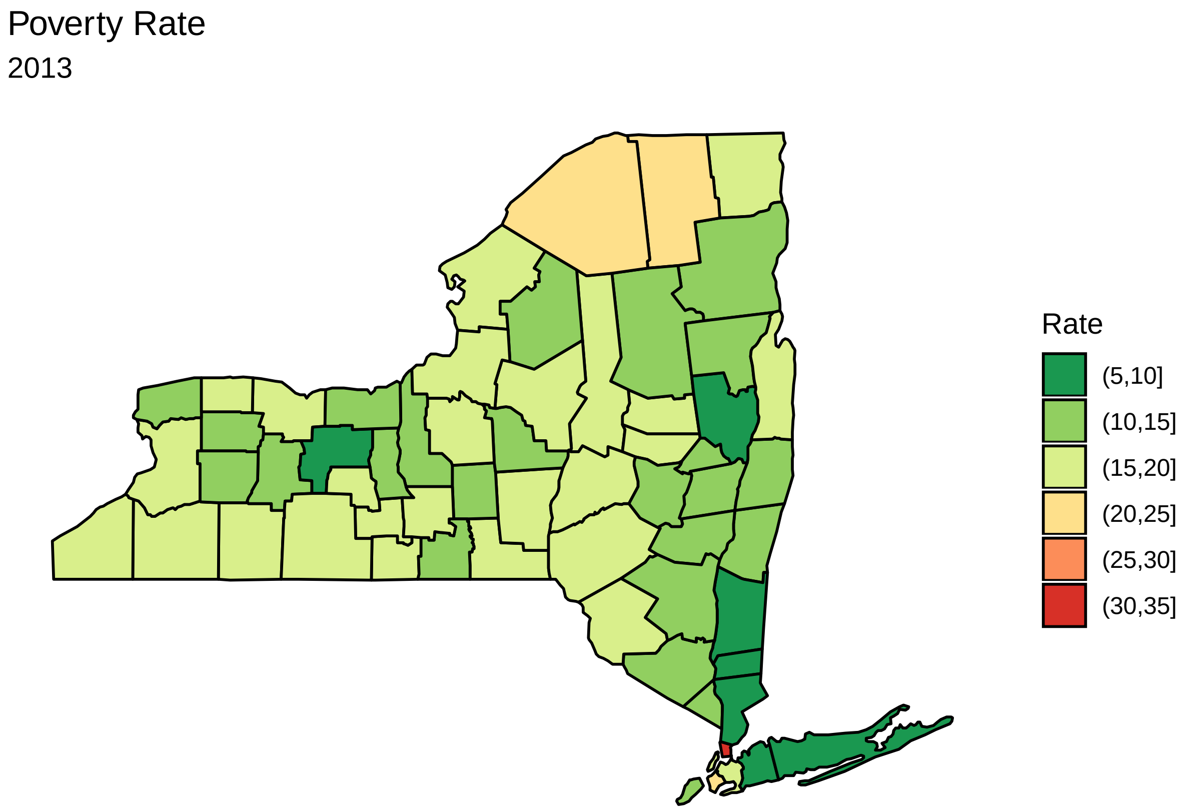

```
unemployment %>% filter(year == "2015") %>%
  NYMap("meanRate","Unemployment Rate","2015",
        "Rate", breaks)
```



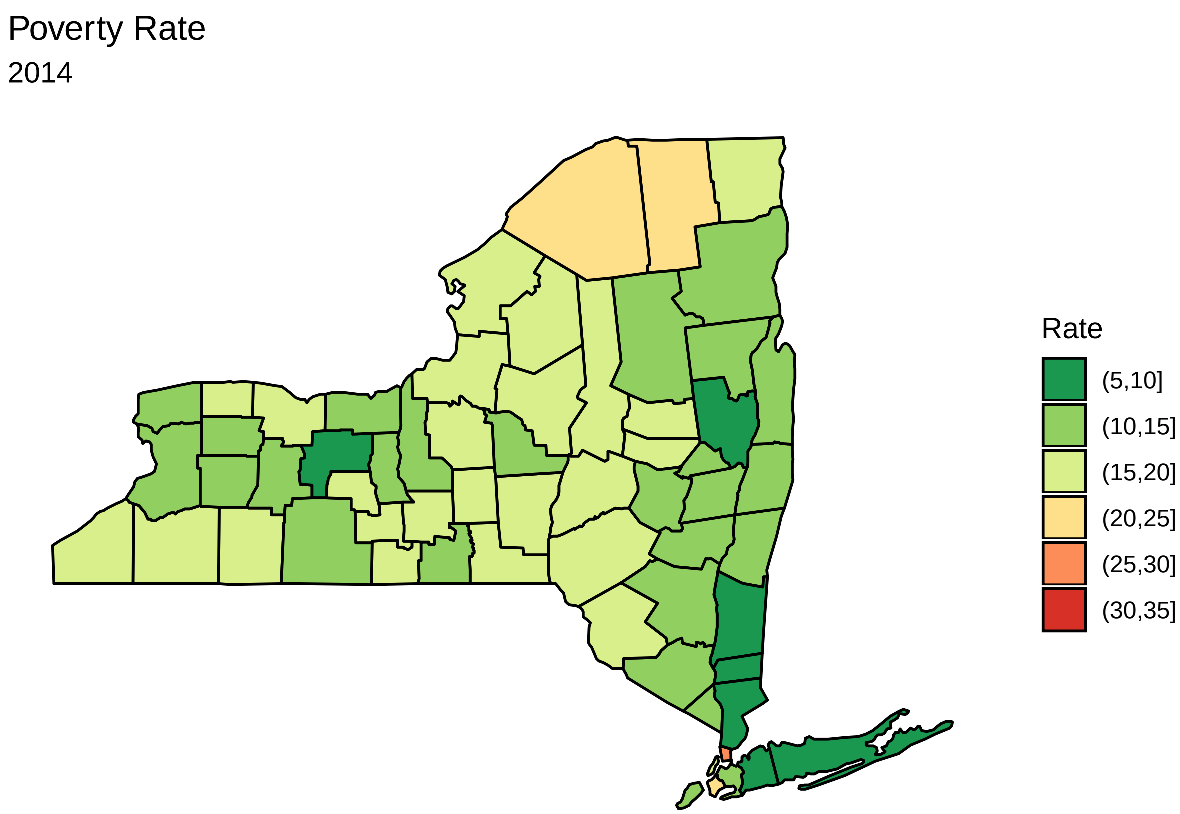
Surprisingly, the unemployment data in some of the worst counties for opioid deaths isn't very bad. In fact, it seems to get better from 2013 to 2015.

How about the poverty rate? Let's map those as well:

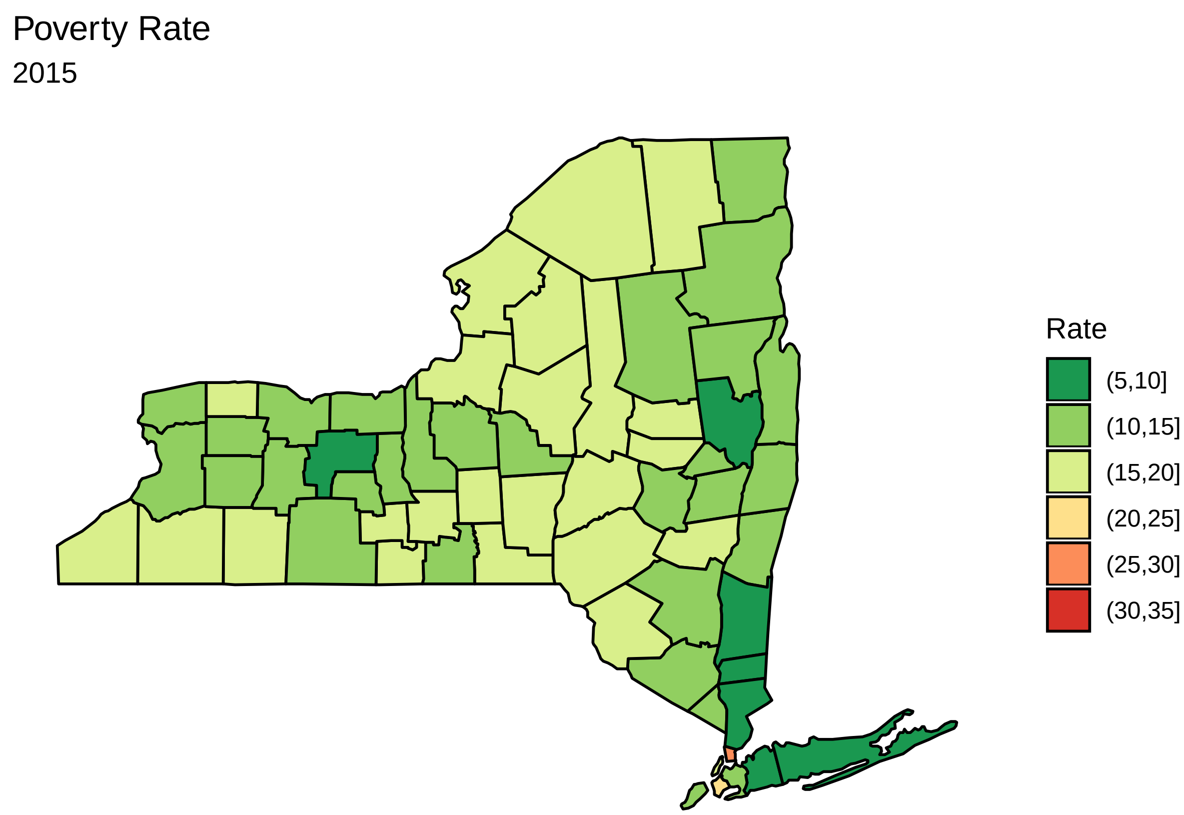
```r
breaks <- c(5, seq(10,35,by=5))

poverty %>% filter(year == "2013") %>%
  NYMap("poverty","Poverty Rate","2013",
        "Rate", breaks)
```




```
poverty %>% filter(year == "2014") %>%
  NYMap("poverty","Poverty Rate","2014",
        "Rate", breaks)
```

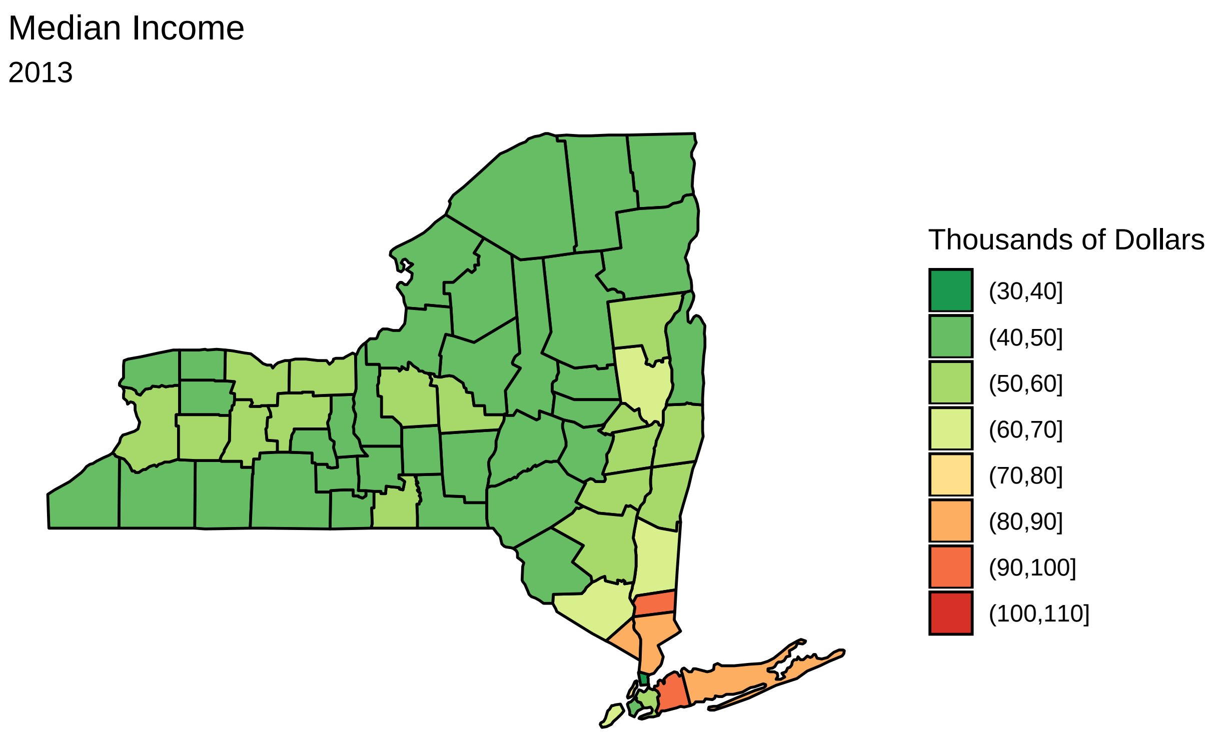



```r
poverty %>% filter(year == "2015") %>%
  NYMap("poverty","Poverty Rate","2015",
        "Rate", breaks)
```

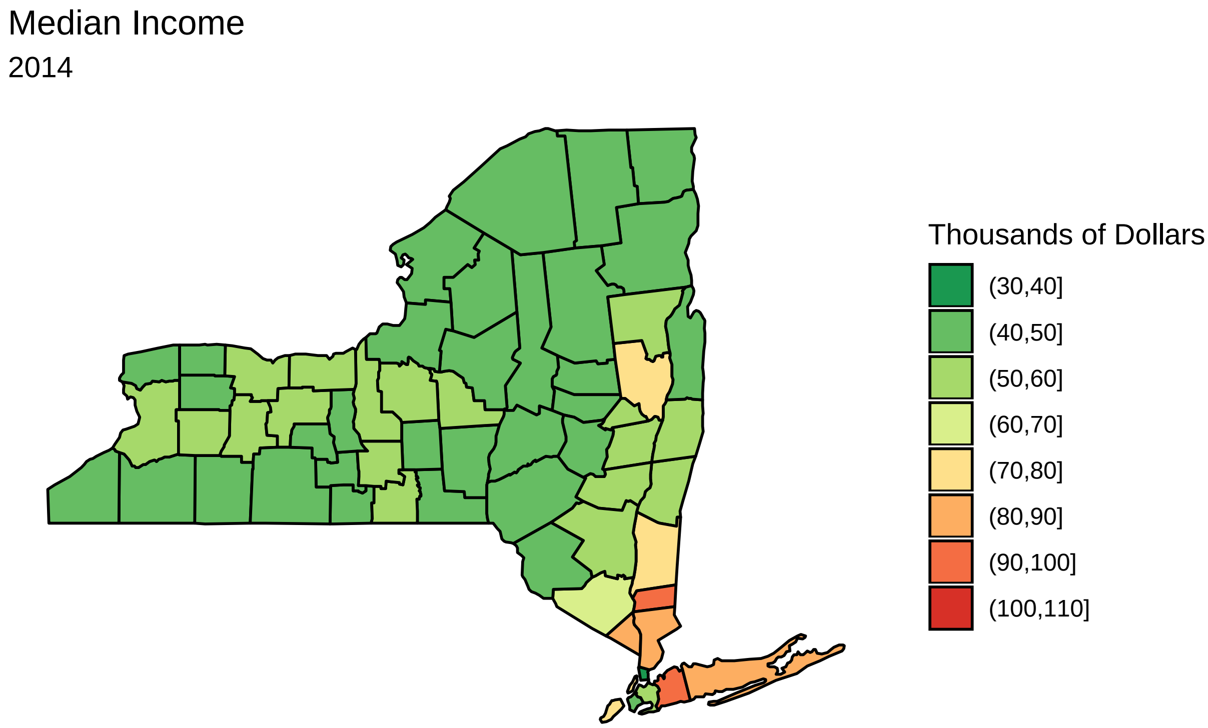


The poverty rate also appears to be low in counties where opioid deaths are high. We can also look at the median incomes of each county:

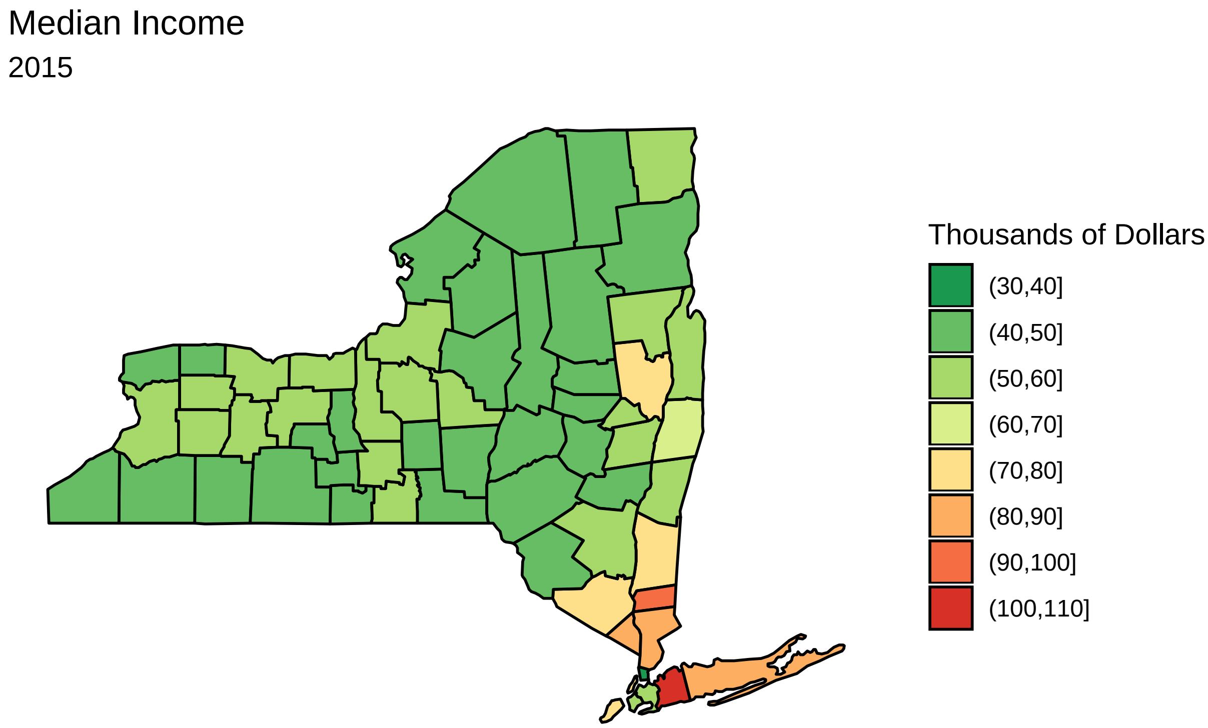
```
breaks <- c(30, seq(40,110,by=10))

poverty %>% filter(year == "2013") %>%
  NYMap("income","Median Income","2013",
        "Thousands of Dollars", breaks)
```




```
poverty %>% filter(year == "2014") %>%
  NYMap("income","Median Income","2014",
        "Thousands of Dollars", breaks)
```




```r
poverty %>% filter(year == "2015") %>%
  NYMap("income","Median Income","2015",
        "Thousands of Dollars", breaks)
```



We see rather high median incomes in some of the problematic counties, which we mostly expected from the poverty levels above.

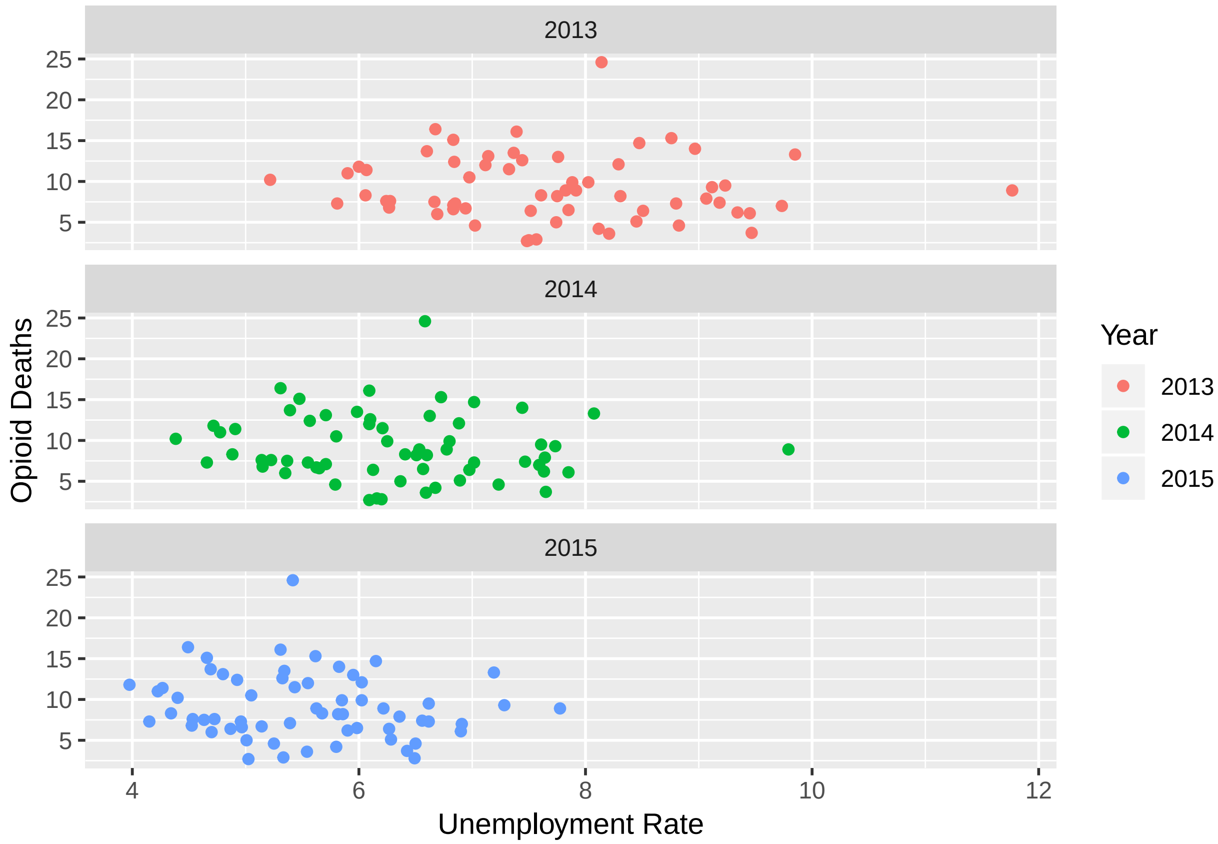
Analysis

To support our analysis, we will join our data frames so we have all the variables we will be using in a single data frame.

```
joint <- inner_join(opioids,unemployment,
                    by=c("county" = "county","year" = "year")) %>%
  inner_join(poverty, by=c("county" = "county","year" = "year"))
```

Let's start by plotting deaths versus unemployment rates.

```
joint %>%
  ggplot(aes(x=meanRate, y=rate, col=as.factor(year))) +
  geom_point() +
  facet_wrap(~ year, nrow = 3) + scale_color_discrete("Year") +
  ylab("Opioid Deaths") + xlab("Unemployment Rate")
```



There appears to be only a slight linear relationship here. In fact, the variables do not seem to be very correlated:

```
joint %>% filter(year==2013) %>%
  {cor(.$rate, .$meanRate)}
```

```
## [1] -0.08282978
```

```
joint %>% filter(year==2014) %>%
  {cor(.$rate, .$meanRate)}
```

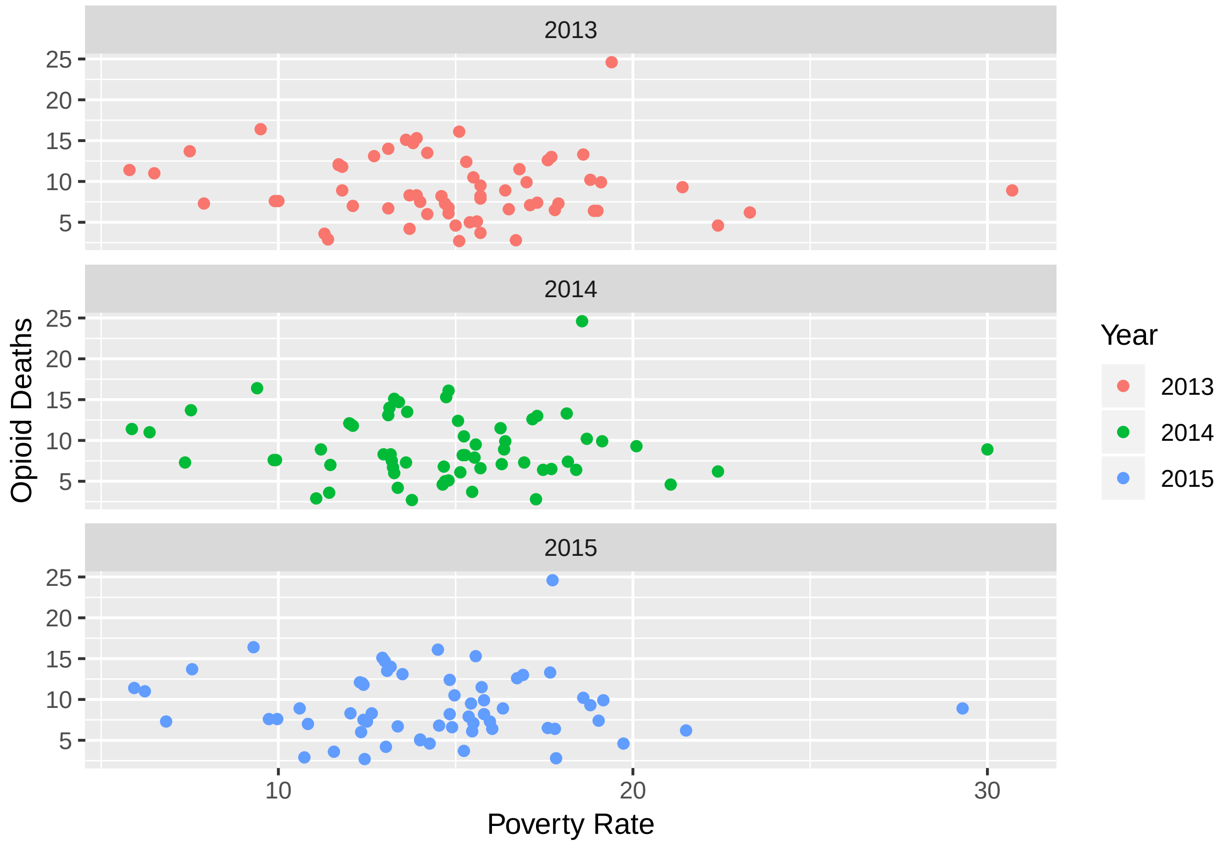
```
## [1] -0.08867533
```

```
joint %>% filter(year==2015) %>%
  {cor(.$rate, .$meanRate)}
```

```
## [1] -0.1518608
```

Next we look at poverty rates:

```
joint %>%
  ggplot(aes(x=poverty, y=rate, col=as.factor(year))) +
  geom_point() +
  facet_wrap(~ year, nrow = 3) + scale_color_discrete("Year") +
  ylab("Opioid Deaths") + xlab("Poverty Rate")
```



The `poverty` variable seems to have little in the way of a linear relationship with the number of opioid deaths as well.

```
joint %>% filter(year==2013) %>%
  {cor(.$rate, .$poverty)}
```

```
## [1] -0.0941887
```

```
joint %>% filter(year==2014) %>%
  {cor(.$rate, .$poverty)}
```

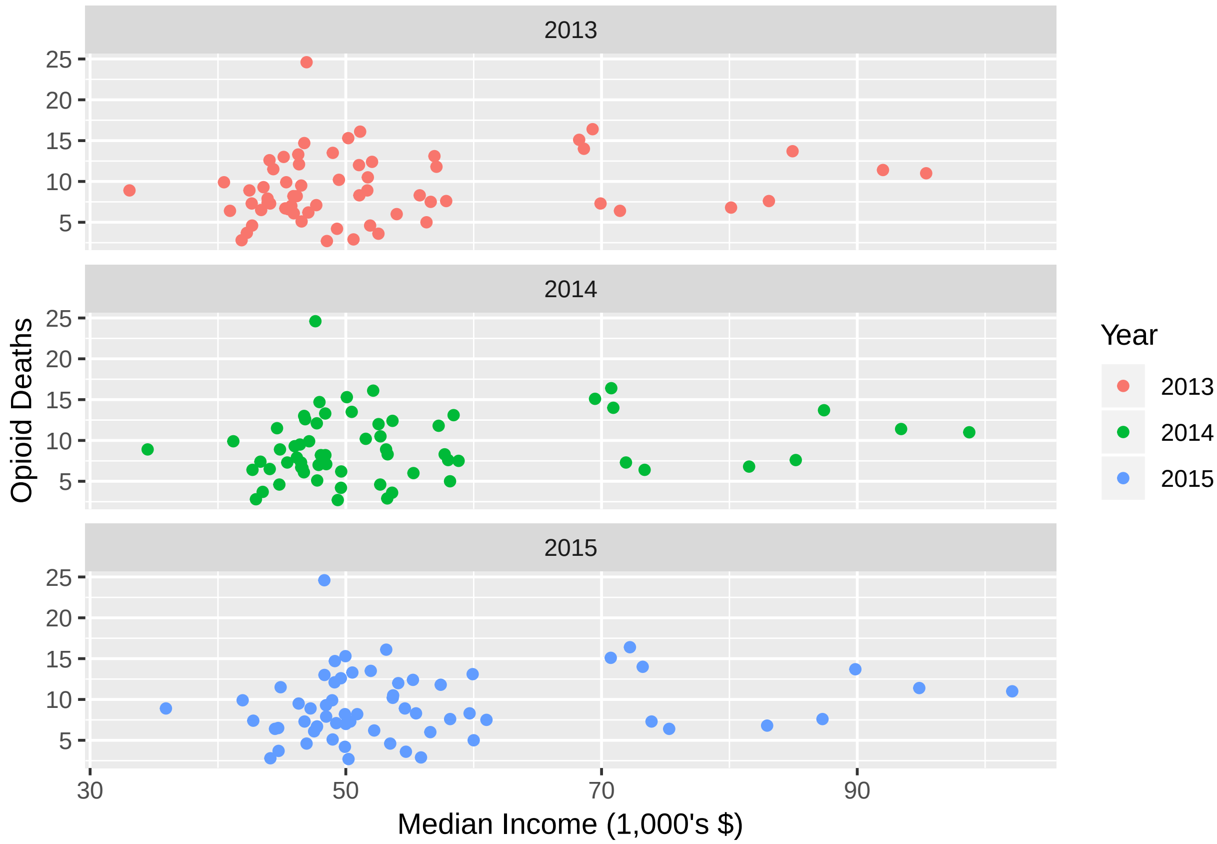
```
## [1] -0.07322007
```

```
joint %>% filter(year==2015) %>%
  {cor(.$rate, .$poverty)}
```

```
## [1] -0.04950194
```

Finally, we explore median income:

```
joint %>%
  ggplot(aes(x=income, y=rate, col=as.factor(year))) +
  geom_point() +
  facet_wrap(~ year, nrow = 3) + scale_color_discrete("Year") +
  ylab("Opioid Deaths") + xlab("Median Income (1,000's $)")
```



Strangely, income too appears to have somewhat of a relationship to the number of overdose deaths.

```
joint %>% filter(year==2013) %>%
  {cor(.$rate, .$income)}
```

```
## [1] 0.1819219
```

```
joint %>% filter(year==2014) %>%
  {cor(.$rate, .$income)}
```

```
## [1] 0.1756106
```

```
joint %>% filter(year==2015) %>%
  {cor(.$rate, .$income)}
```

```
## [1] 0.1688844
```

Analysis

Let's build a model to see how the variables relate to the overdose death rate. For brevity, we'll use 2013 specifically:

```
# Linear model for 2013
mod_income1 <- lm(rate ~ income + poverty + meanRate, data = joint[which(joint$year==2013),])

summary(mod_income1)
```

```
##
## Call:
```

```
## lm(formula = rate ~ income + poverty + meanRate, data = joint[which(joint$year ==
##     2013), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1581 -2.6726 -0.6752  2.8234 15.6239
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.34337    6.60449   0.658    0.513
## income       0.06990    0.05734   1.219    0.228
## poverty      0.04385    0.18473   0.237    0.813
## meanRate     0.06160    0.56122   0.110    0.913
##
## Residual standard error: 4.085 on 58 degrees of freedom
## Multiple R-squared:  0.03472,    Adjusted R-squared:  -0.01521
## F-statistic: 0.6955 on 3 and 58 DF,  p-value: 0.5586
```

Here we see that none of the variables are statistically significant. However, because they could definitely have some level of colinearity, we'll remove the worst one (`meanRate`) and run a new model.

```
# Linear model for 2013 (minus unemployment rate)
mod_income2 <- lm(rate ~ income + poverty, data = joint[which(joint$year==2013),])

summary(mod_income2)
```

```
##
## Call:
## lm(formula = rate ~ income + poverty, data = joint[which(joint$year ==
##     2013), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1799 -2.6500 -0.6622  2.7961 15.6098
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.81082    5.00544   0.961    0.340
## income       0.06805    0.05436   1.252    0.216
## poverty      0.05080    0.17207   0.295    0.769
##
## Residual standard error: 4.051 on 59 degrees of freedom
## Multiple R-squared:  0.03452,    Adjusted R-squared:  0.001794
## F-statistic: 1.055 on 2 and 59 DF,  p-value: 0.3547
```

Now it appears that the `income` variable's p-value has decreased, yet `poverty` remains statistically insignificant.

Now we will do a simple regression model with `income` only to see how it describes the opioid death rates.

```
# Linear model for 2013 (only income)
mod_income3 <- lm(rate ~ income, data = joint[which(joint$year==2013),])

summary(mod_income3)
```

```
##
## Call:
## lm(formula = rate ~ income, data = joint[which(joint$year ==
```

```
##     2013), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2202 -2.6632 -0.6796  2.7883 15.7708
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.14121    2.16290   2.839  0.00616 **
## income       0.05728    0.03997   1.433  0.15703
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.02 on 60 degrees of freedom
## Multiple R-squared:  0.0331, Adjusted R-squared:  0.01698
## F-statistic: 2.054 on 1 and 60 DF,  p-value: 0.157
```

Finally, even the `income` variable alone does not seem statistically significant enough in this model to demonstrate a linear relationship.

Conclusions

All of the regression models have shown that there is **no statistically significant linear relationship between the death rate of opioids and either median income, unemployment, or poverty rates**.

Is this what we expected to see? That depends on your point of view. As mentioned in the introduction, addiction has been stigmatized as a personal failing, a flaw in character that allows someone to become addicted. That characterization has led to the widespread association with the lower rungs of the socioeconomic ladder.

Our results, seem to run counter to that. They seem to support the more modern and enlightened view that addiction is not a problem common only to the disadvantaged.

Caveats and Assumptions

Some assumptions were taken in the course of this analysis.

First, we have assumed that the opioid death rate is a good surrogate for opioid *use*. It is possible that use and deaths are not as tightly correlated as assumed. If we were looking at more recent data, one could make an argument that with the widespread use of Naloxone, and an increase of education about overdose dangers, that this doesn't hold true. However, back in 2013 it seems a somewhat safe assumption.

Secondly, we are treating each county as a monolithic entity. There can be, however, significant differences in demographics *within* a county that may make summary statistics like we are using less accurate.

Works Cited