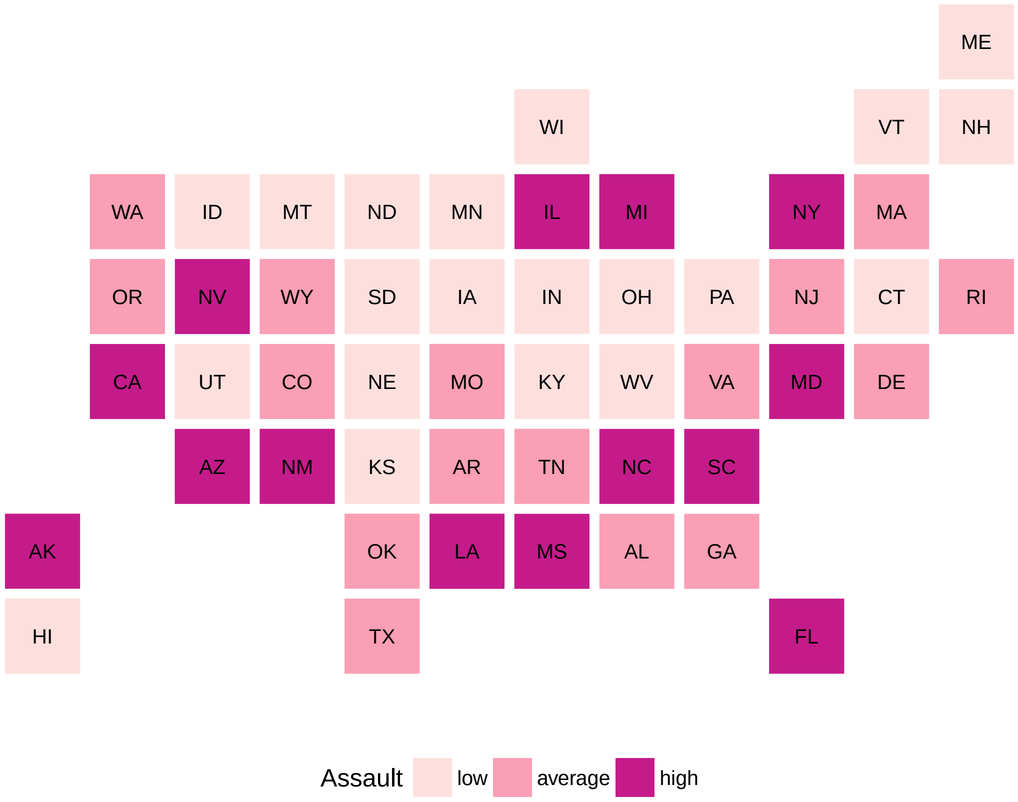
Alternatives to Choropleths

- We have discussed how traditional choropleths can be problematic.
- One problem is that the viewer will place importance to areas that are geographically larger than other regions
- Some solutions have been recently considered that sizes states based on a factor other than geographical area. These factors can include population, number of electoral votes, or equal area per state alternatives.
- Such plots are commonly called Statebins or Tilegrams.

Statebins

- A state bin can be used if our objective is to see how states compare and to retain some spatial information about the states.
- While statebins will help deter a viewer from giving importance to a geographically large area (like Alaska), some of the spatial information is lost.
- As always, keep in mind what the overall objective of the visualization is to determine whether a statebin or traditional choropleth is more effective.

```r
require(statebins)
require(tilegramsR)
require(leaflet)
require(leaflet.extras)
require(colormap)
require(dplyr)
require(RColorBrewer)

USArrests<-mutate(USArrests, state =rownames(USArrests))

statebins(state_data=USArrests,
          state_col="state",            #Variable in data set that corresponds to the state names
                                        #states must be in form "Connecticut" or "CT")
          value_col="Assault",          #Variable in data set that we wish to map
          text_color="black",
          font_size=3,
          legend_title = "Assault",
          legend_position="bottom",
          brewer_pal = "RdPu",
          breaks=3,
          labels=c("low","average","high"))
```

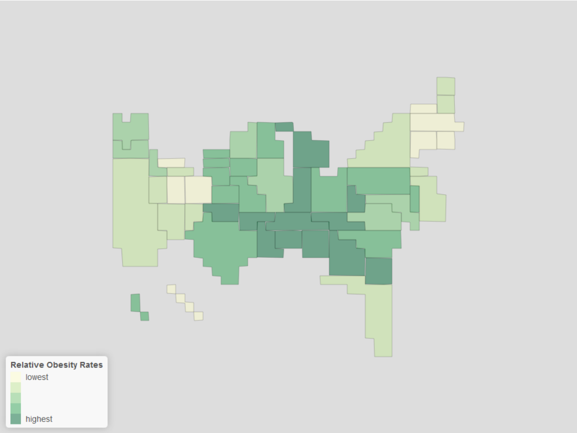


Tilegrams

- A "tilegram" is a map made of tiles where regions are sized proportionally to a dataset. The name is short for a tiled cartogram.
- Tilegrams can represent demographic data more truthfully than conventional, geographic maps, while still retaining a familiar appearance. However, as with statebins, some of the geographic information may be lost.
- The tilegramR package has a collection of tilegrams to select from. An updated list is kept here: https://cran.r-project.org/web/packages/tilegramsR/vignettes/UsingTilegramsInR.html
- Some examples include:
  - A sf object where each feature = 500K people
  - A sf object where each feature = 1 state of the U.S.A.
  - Regions of France and Germany (sized by population)
  - SF objects 'seen in the wild': Wall Street Journal and Washington Post

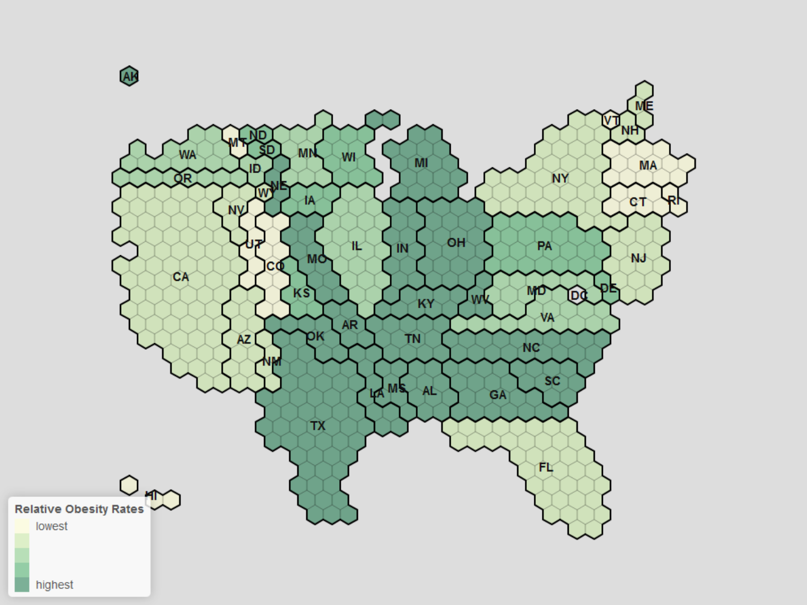
- Let's begin with a familiar example of State-level data - Obesity. Suppose we want to create a tilegram where each state is sized by the population. Suppose we want to use the tiles created by the Wall Street Journal.

```
Obese <- read.csv("Obese.csv")


##Leaflet Options to avoid zooming, panning, etc.

getLeafletOptions <- function(minZoom, maxZoom) {
  leafletOptions(
    crs = leafletCRS("L.CRS.Simple"),
    minZoom = minZoom, maxZoom = maxZoom,
    dragging = FALSE, zoomControl = FALSE,
    tap = TRUE,
    attributionControl = FALSE)
}


pal <- colorQuantile("YlGn",domain=NULL, n =5)


Obese$id<-state.abb[match(Obese$State,state.name)]
merged_data<-merge(sf_WSJ, Obese, by="id")

leaflet(
  options= getLeafletOptions(-20, 20)) %>%
  addPolygons(
    data=merged_data,
    weight=1,color='#000000', fillOpacity = 0.5, opacity=0.2,
    fillColor= ~pal(AdultsObese)) %>%
  addLegend("bottomleft",
            colors=brewer.pal(5,"YlGn"),
            labels=c("lowest","","","","highest"),
            title="Relative Obesity Rates")
```



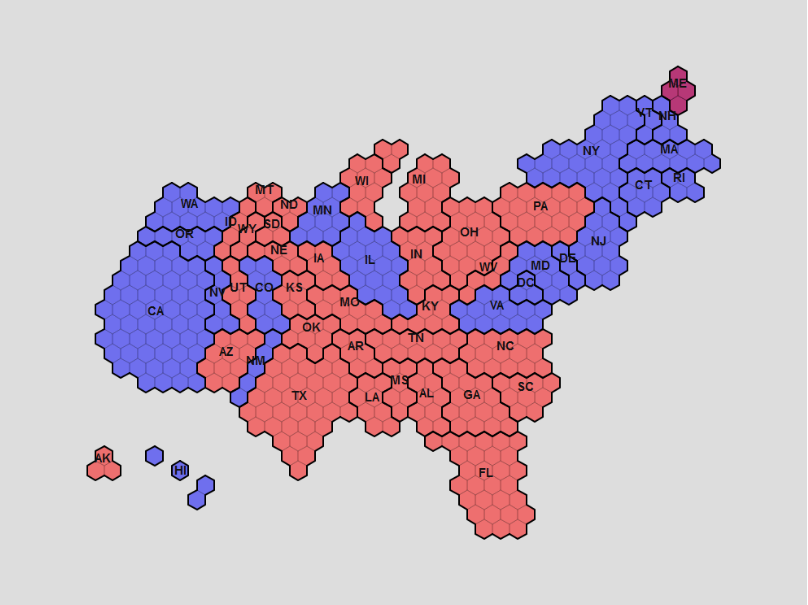
- Next suppose we want to create a tilegram where each hexagon represents 500,000 people.

```r
pal <- colorQuantile("YlGn",domain=NULL, n =5)


Obese$state<-state.abb[match(Obese$State,state.name)]
merged_data<-merge(sf_Pitch_US_Population_2016_v1, Obese, by="state")
merged_data_states<-merge(sf_Pitch_US_Population_2016_v1.states, Obese, by="state")

leaflet(
  options= getLeafletOptions(-1.5, -1.5)) %>%
  addPolygons(
    data=merged_data,
    weight=1,color='#000000', fillOpacity = 0.5, opacity=0.2,
    fillColor= ~pal(AdultsObese)) %>%
  addPolygons(
    data=merged_data_states, group = 'states',
    weight=2,color='#000000',
    fill = T, opacity = 1, fillOpacity = 0,
    highlightOptions = highlightOptions(weight = 4),
    popup=~paste(state, "'s Obesity Rate", AdultsObese)) %>%
  addLabelOnlyMarkers(
    data=sf_Pitch_US_Population_2016_v1.centers,
    label = ~as.character(state),
    labelOptions = labelOptions(
      noHide = 'T', textOnly = T,
      offset=c(-4,-10), textsize = '15px')) %>%
  addLegend("bottomleft",
            colors=brewer.pal(5,"YlGn"),
            labels=c("lowest","","","","highest"),
            title="Relative Obesity Rates")
```



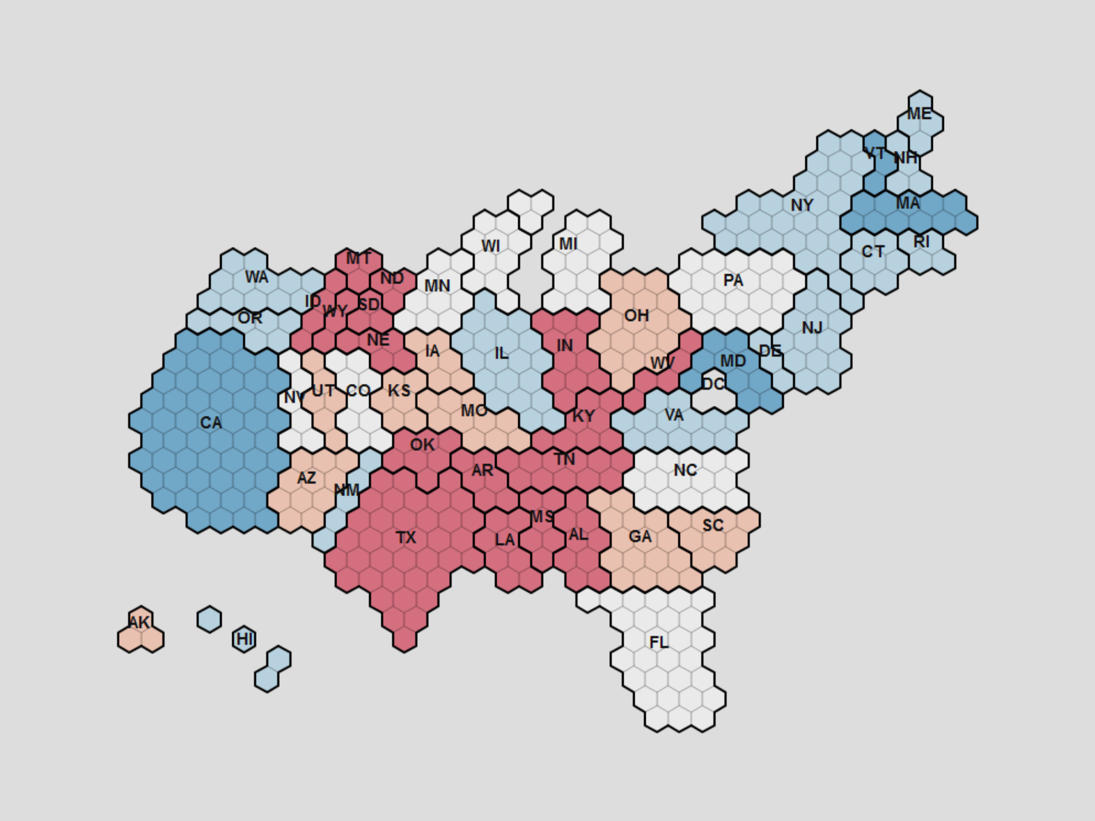
- Now let's make a Tilegram Plot where states are sized by number of electoral college votes, each hexagon=1 vote
- We will utilize post-election results to show how many electoral votes went to each candidate.

```
ElectoralVotes <- read.csv("2016 Electoral Votes.csv")

pal<-colorFactor(c("blue","red"), domain=NULL)

ElectoralVotes$state<-state.abb[match(ElectoralVotes$State,state.name)]

#Some additional data management required
ElectoralVotes$state[ElectoralVotes$State=="District of Columbia"]<-"DC"
ElectoralVotes$state[ElectoralVotes$State=="Lousiana"]<-"LA"

merged_data2<-merge(sf_FiveThirtyEightElectoralCollege, ElectoralVotes, by="state")

leaflet(
  options= getLeafletOptions(-1.5, -1.5)) %>%
  addPolygons(
    data=merged_data2,
    weight=1,color='#000000', fillOpacity = 0.5, opacity=0.2,
    fillColor= ~pal(Winning.Party)) %>%
  addPolygons(
    data=sf_FiveThirtyEightElectoralCollege.states, group = 'states',
    weight=2,color='#000000',
    fill = T, opacity = 1, fillOpacity = 0,
    highlightOptions = highlightOptions(weight = 4)) %>%
  addLabelOnlyMarkers(
    data=sf_FiveThirtyEightElectoralCollege.centers,
    label = ~as.character(state),
    labelOptions = labelOptions(
      noHide = 'T', textOnly = T,
      offset=c(-8,-20), textsize = '15px'))
```



- Now suppose we wanted to use pre-election polling results to determine how the elector votes would be decided.
    - Once again we will use the tilegrams where the states are sized by the number of electoral college votes.
    - Let us only examine data that provides state-level polling and have the expected outcome be the median across all polls.
    - To determine the difference in support between the candidates we will take the difference of Trump and Clinton (thus positive numbers will represent greater support for Trump and negative values will represent greater support of Clinton).

```
polls <- read.csv("president_general_polls_2016.csv")

polls2<-filter(polls, state%in%state.name)
polls2<-summarise(group_by(polls2, state), clinton=median(rawpoll_clinton), trump=median(rawpoll_tr
polls2<-mutate(polls2, difference=trump-clinton)
polls2<-mutate(polls2, state=state.abb[match(state,state.name)])
merged_data3<-merge(sf_FiveThirtyEightElectoralCollege, polls2, by="state")

pal<-colorQuantile(palette="RdBu",domain=NULL,n=5, reverse=TRUE)

leaflet(
  options= getLeafletOptions(-1.5, -1.5)) %>%
  addPolygons(
    data=merged_data3,
    weight=1,color='#000000',
    fillOpacity = 0.5,
    opacity=0.2,
    fillColor= ~pal(difference)) %>%
  addPolygons(
    data=sf_FiveThirtyEightElectoralCollege.states,
    group = 'states',
    weight=2,color='#000000',
    fill = T, opacity = 1, fillOpacity = 0,
    highlightOptions = highlightOptions(weight = 4)) %>%
  addLabelOnlyMarkers(
    data=sf_FiveThirtyEightElectoralCollege.centers,
    label = ~as.character(state),
    labelOptions = labelOptions(
      noHide = 'T', textOnly = T,
      offset=c(-8,-20), textsize = '15px'))
```


ME
VT
NH
NY
MA
RI
CT
WI
MI
MT
ND
WA
MN
PA
ID
SD
WY
OR
OH
NJ
IN
NE
IA
IL
DE
MD
WV
DC
NV
UT
CO
KS
KY
VA
MO
CA
OK
TN
AR
NC
AZ
NM
MS
SC
TX
LA
AL
GA
AK
HI
FL