Employment Visual

Xiang XU

Contents

1 Intro

This project will help us explore employment data came from United States Department of Labor. We will use pay and benefits data in 2018.

1.1 Read data

```
ann2018 <- fread('data/2018.annual.singlefile.csv')
```

1.2 How to instore data into SQLite

```
# SQLite installed
sqldf('attach blsdb as new')
read.csv.sql('data/2018.annual.singlefile.csv',
             sql = 'create table main.ann2018 as select * from file',
             dbname = 'blsdb')
# now the data all in SQLite database but not in R
# then read data into R
ann2018 <- sqldf('select * from main.ann2018', dbname = 'blsdb')
```

1.3 View

```
dim(ann2018)
```

```
## [1] 3573215      38
```

```
head(ann2018,3)
```

```
##    area_fips own_code industry_code agglvl_code size_code year qtr
## 1:     01000        0            10          50         0 2018   A
## 2:     01000        1            10          51         0 2018   A
## 3:     01000        1           102          52         0 2018   A
##    disclosure_code annual_avg_estabs annual_avg_emplvl total_annual_wages
## 1:                            127038           1961448        92911268252
## 2:                              1196             53003         4421689834
## 3:                              1196             53003         4421689834
##    taxable_annual_wages annual_contributions annual_avg_wkly_wage
## 1:          15408152922            205567883                  911
## 2:                    0                    0                 1604
## 3:                    0                    0                 1604
##    avg_annual_pay lq_disclosure_code lq_annual_avg_estabs
## 1:          47369                                   1.0
## 2:          83424                                   1.6
## 3:          83424                                   1.6
##    lq_annual_avg_emplvl lq_total_annual_wages lq_taxable_annual_wages
## 1:                 1.00                 1.00                       1
## 2:                 1.41                 1.70                       0
## 3:                 1.44                 1.73                       0
##    lq_annual_contributions lq_annual_avg_wkly_wage lq_avg_annual_pay
## 1:                       1                    1.0              1.00
## 2:                       0                    1.2              1.21
## 3:                       0                    1.2              1.21
##    oty_disclosure_code oty_annual_avg_estabs_chg
## 1:                                          2157
## 2:                                           -12
## 3:                                           -12
##    oty_annual_avg_estabs_pct_chg oty_annual_avg_emplvl_chg
## 1:                           1.7                     24629
## 2:                          -1.0                      -128
## 3:                          -1.0                      -128
##    oty_annual_avg_emplvl_pct_chg oty_total_annual_wages_chg
## 1:                           1.3                 3822557436
## 2:                          -0.2                   82651203
## 3:                          -0.2                   82651203
```

```
##    oty_total_annual_wages_pct_chg oty_taxable_annual_wages_chg
## 1:                            4.3                     475040033
## 2:                            1.9                             0
## 3:                            1.9                             0
##    oty_taxable_annual_wages_pct_chg oty_annual_contributions_chg
## 1:                              3.2                    -13660378
## 2:                              0.0                            0
## 3:                              0.0                            0
##    oty_annual_contributions_pct_chg oty_annual_avg_wkly_wage_chg
## 1:                             -6.2                           26
## 2:                              0.0                           33
## 3:                              0.0                           33
##    oty_annual_avg_wkly_wage_pct_chg oty_avg_annual_pay_chg
## 1:                              2.9                  1372
## 2:                              2.1                  1756
## 3:                              2.1                  1756
##    oty_avg_annual_pay_pct_chg
## 1:                        3.0
## 2:                        2.2
## 3:                        2.2
```

1.4 Get, merge and add additional information

```
for (u in c('agglevel', 'area' ,'industry', 'ownership','size')){
  assign(u,  fread(paste0('data/',u,'_titles.csv'))   )
}
```

```
intersect(names(ann2018), names(agglevel))
```

```
## [1] "agglvl_code"
```

```
intersect(names(ann2018), names(area))
```

```
## [1] "area_fips"
```

```
intersect(names(ann2018), names(industry))
```

```
## [1] "industry_code"
```

```
intersect(names(ann2018), names(ownership))
```

```
## [1] "own_code"
```

```
intersect(names(ann2018), names(size))
```

```
## [1] "size_code"
```

1.4.1 Join data

```
title <- c('agglevel','industry', 'ownership','size')
ann2018_full <- ann2018

for (i in 1:length(title)){
  eval(parse(text = paste0('ann2018_full <- left_join(ann2018_full,',title[i],')')))
}
```

1.5 Add geographic information

1.5.1 view

```
head(area)
```

```
##    area_fips                                    area_title
## 1:     US000                                    U.S. TOTAL
## 2:     USCMS     U.S. Combined Statistical Areas (combined)
## 3:     USMSA U.S. Metropolitan Statistical Areas (combined)
## 4:     USNMS  U.S. Nonmetropolitan Area Counties (combined)
## 5:     01000                          Alabama -- Statewide
## 6:     01001                      Autauga County, Alabama
```

1.5.2 capitalize

```
# Capitalize the first letter
simpleCap <- function(x){
  if(!is.na(x)){
    s <- strsplit(x,' ')[[1]]
    # print( paste0("strsplit(x,' '):",strsplit(x,' ')))
    # print( paste0("strsplit(x,' ')[1]:",strsplit(x,' ')[1]))
    # print(paste0("s:",s))
    # print(substring(s,1))
    # print(substring(s,1,1))
    # print(toupper(substring(s,1,1)))
    # print(substring(s,2))
    paste(toupper(substring(s,1,1)), substring(s,2),sep= '', collapse = ' ' )

  }else {NA}
}
```

1.5.3 county fips & name

```
data("county.fips")
head(county.fips)
```

```
##   fips        polyname
## 1 1001 alabama,autauga
## 2 1003 alabama,baldwin
## 3 1005 alabama,barbour
## 4 1007    alabama,bibb
## 5 1009  alabama,blount
## 6 1011 alabama,bullock
```

```
# to acieve data matching, fill in 'fips'
county.fips$fips <- str_pad(county.fips$fips,width = 5, pad = '0', side = 'left')
head(county.fips)
```

```
##    fips        polyname
## 1 01001 alabama,autauga
## 2 01003 alabama,baldwin
## 3 01005 alabama,barbour
## 4 01007    alabama,bibb
## 5 01009  alabama,blount
```

```
## 6 01011 alabama,bullock
```

```
# extract city name from polyname
county.fips$county <- sapply(
  gsub('([a-z\ ]+),([a-z\ ]+)', '\\2' , as.character(county.fips$polyname) ),
  simpleCap )
county.fips <- unique(county.fips)
head(county.fips)
```

```
##    fips        polyname  county
## 1 01001 alabama,autauga Autauga
## 2 01003 alabama,baldwin Baldwin
## 3 01005 alabama,barbour Barbour
## 4 01007    alabama,bibb    Bibb
## 5 01009  alabama,blount  Blount
## 6 01011 alabama,bullock Bullock
```

1.5.4 state fips & name

```
data("state.fips")
head(state.fips)
```

```
##   fips ssa region division abb    polyname
## 1    1   1      3        6  AL     alabama
## 2    4   3      4        8  AZ     arizona
## 3    5   4      3        7  AR    arkansas
## 4    6   5      4        9  CA  california
## 5    8   6      4        8  CO    colorado
## 6    9   7      1        1  CT connecticut
```

```
state.fips$fips <- str_pad(state.fips$fips, width = 2, pad = '0', side = 'left')
# extract city name from polyname
state.fips$state <- sapply(
  gsub('([a-z\ ]+):([a-z\ \\]+)', '\\1' , as.character(state.fips$polyname) ),
  simpleCap )
head(state.fips)
```

```
##   fips ssa region division abb    polyname       state
## 1   01   1      3        6  AL     alabama     Alabama
## 2   04   3      4        8  AZ     arizona     Arizona
## 3   05   4      3        7  AR    arkansas    Arkansas
## 4   06   5      4        9  CA  california  California
## 5   08   6      4        8  CO    colorado    Colorado
## 6   09   7      1        1  CT connecticut Connecticut
```

```
mystate <- unique(state.fips[,c('fips', 'abb', 'state')])
lower48 <-  setdiff(unique(state.fips$state), c('Hawaii', 'Alaska'))
```

1.5.5 combine all geographic data

```
myarea <- merge(area, county.fips,
                by.x = 'area_fips', by.y = 'fips', all.x =TRUE)

myarea$state_fips <- substr(myarea$area_fips,1,2)
myarea <- merge(myarea, mystate, by.x = 'state_fips', by.y = 'fips', all.x = T)
```

1.5.6 combine all geographic + employment data and save to .rda

```
ann2018_full <- left_join(ann2018_full, myarea)
ann2018_full <- filter(ann2018_full, state %in% lower48)
save(ann2018_full, file ='data/ann2018full.rda', compress = T )
```

1.6 Get pay + employment data from county & state level

1.6.1 state level

1.6.1.1 focus on averge annual pay + averge annual employment

```
d.state <- filter(ann2018_full, agglvl_code == 50) # state summary
d.state <- select(d.state, state, avg_annual_pay, annual_avg_emplvl)
```

1.6.1.2 discretization, create var wage empquatile

```
d.state$wage <- cut(d.state$avg_annual_pay,
                    quantile(d.state$avg_annual_pay,
                             c(seq(0,.8,by=.2), .9, .95,.99,1)
                             )
                    )
d.state$empquatile <- cut(d.state$annual_avg_emplvl,
                          quantile(d.state$annual_avg_emplvl,
                                   c(seq(0,.8,by=.2), .9, .95,.99,1)
                                   )
                          )
```

```
x <- quantile(d.state$avg_annual_pay, c(seq(0,.8,by=.2), .9, .95,.99,1) )
xx <- paste0(round(x/1000),'K')
Labs <- paste(xx[-length(xx)],xx[-1],sep='-')
levels(d.state$wage) <- Labs

x <- quantile(d.state$annual_avg_emplvl, c(seq(0,.8,by=.2), .9, .95,.99,1) )
xx <- ifelse(x>1000, paste0(round(x/1000),'K'),round(x))
Labs <- paste(xx[-length(xx)],xx[-1],sep='-')
levels(d.state$empquatile) <- Labs
```

1.6.2 county level

1.6.2.1 focus on averge annual pay + averge annual employment

```
d.county <- filter(ann2018_full, agglvl_code == 70) # county summary
d.county <- select(d.county, county, avg_annual_pay, annual_avg_emplvl)
```

1.6.2.2 discretization, create var wage empquatile

```
d.county$wage <- cut(d.county$avg_annual_pay,
                    quantile(d.county$avg_annual_pay,
                             c(seq(0,.8,by=.2), .9, .95,.99,1)
                             )
                    )
d.county$empquatile <- cut(d.county$annual_avg_emplvl,
                          quantile(d.county$annual_avg_emplvl,
                                   c(seq(0,.8,by=.2), .9, .95,.99,1)
```

```
                                    )
                            )
```

```
x <- quantile(d.county$avg_annual_pay, c(seq(0,.8,by=.2), .9, .95,.99,1) )
xx <- ifelse(x>1000, paste0(round(x/1000),'K'),round(x))
Labs <- paste(xx[-length(xx)],xx[-1],sep='-')
levels(d.county$wage) <- Labs

x <- quantile(d.county$annual_avg_emplvl, c(seq(0,.8,by=.2), .9, .95,.99,1) )
xx <- ifelse(x>1000, paste0(round(x/1000),'K'),round(x))
Labs <- paste(xx[-length(xx)],xx[-1],sep='-')
levels(d.county$empquatile) <- Labs
```

1.6.2.3 use Discretize function

```
# create `Discretize` function
Discretize <- function(x, breaks = NULL){
  if(is.null(breaks)){

    breaks <- quantile(x, c(seq(0,.8,by=.2), .9, .95,.99,1) )

    if(sum(breaks==0) >1){
      tmp <- which(breaks ==0, arr.ind = TRUE)
      breaks <- breaks[max(tmp):length(breaks)]
    }

  }

  x.discrete <- cut(x, breaks, include.lowest = T)
  breaks.eng <- ifelse( breaks>1000, paste0(round(breaks/1000),'K'),round(breaks))
  Labs <- paste(breaks.eng[-length(breaks.eng)],breaks.eng[-1],sep='-')
  levels(x.discrete) <- Labs

  return(x.discrete)
}
```

```
#  apply
d.county <- ann2018_full %>%
  filter(agglvl_code == 70)%>%
  select(state, county , abb, avg_annual_pay, annual_avg_emplvl) %>%
  mutate( wage = Discretize(avg_annual_pay),
          empquatile = Discretize(annual_avg_emplvl))
```

Now data all prepared.

1.7 Visualize geographic features

1.7.1 extract map data

```
state_df <- map_data('state')
cty_df <- map_data('county')
head(state_df)
```

```
##        long      lat group order  region subregion
```

```
## 1 -87.46201 30.38968     1     1 alabama      <NA>
## 2 -87.48493 30.37249     1     2 alabama      <NA>
## 3 -87.52503 30.37249     1     3 alabama      <NA>
## 4 -87.53076 30.33239     1     4 alabama      <NA>
## 5 -87.57087 30.32665     1     5 alabama      <NA>
## 6 -87.58806 30.32665     1     6 alabama      <NA>
```

```
head(cty_df)
```

```
##        long      lat group order  region subregion
## 1 -86.50517 32.34920     1     1 alabama   autauga
## 2 -86.53382 32.35493     1     2 alabama   autauga
## 3 -86.54527 32.36639     1     3 alabama   autauga
## 4 -86.55673 32.37785     1     4 alabama   autauga
## 5 -86.57966 32.38357     1     5 alabama   autauga
## 6 -86.59111 32.37785     1     6 alabama   autauga
```

1.7.2 data transformation to match

```
transform_mapdata <- function(x){
  names(x)[5:6] <- c('state', 'county')

  for( u in c('state', 'county')){
    x[,u] <- sapply(x[,u], simpleCap)
  }
  return(x)
}
state_df <- transform_mapdata(state_df)
cty_df <- transform_mapdata(cty_df)
head(state_df)
```

```
##        long      lat group order   state county
## 1 -87.46201 30.38968     1     1 Alabama   <NA>
## 2 -87.48493 30.37249     1     2 Alabama   <NA>
## 3 -87.52503 30.37249     1     3 Alabama   <NA>
## 4 -87.53076 30.33239     1     4 Alabama   <NA>
## 5 -87.57087 30.32665     1     5 Alabama   <NA>
## 6 -87.58806 30.32665     1     6 Alabama   <NA>
```

```
head(cty_df)
```

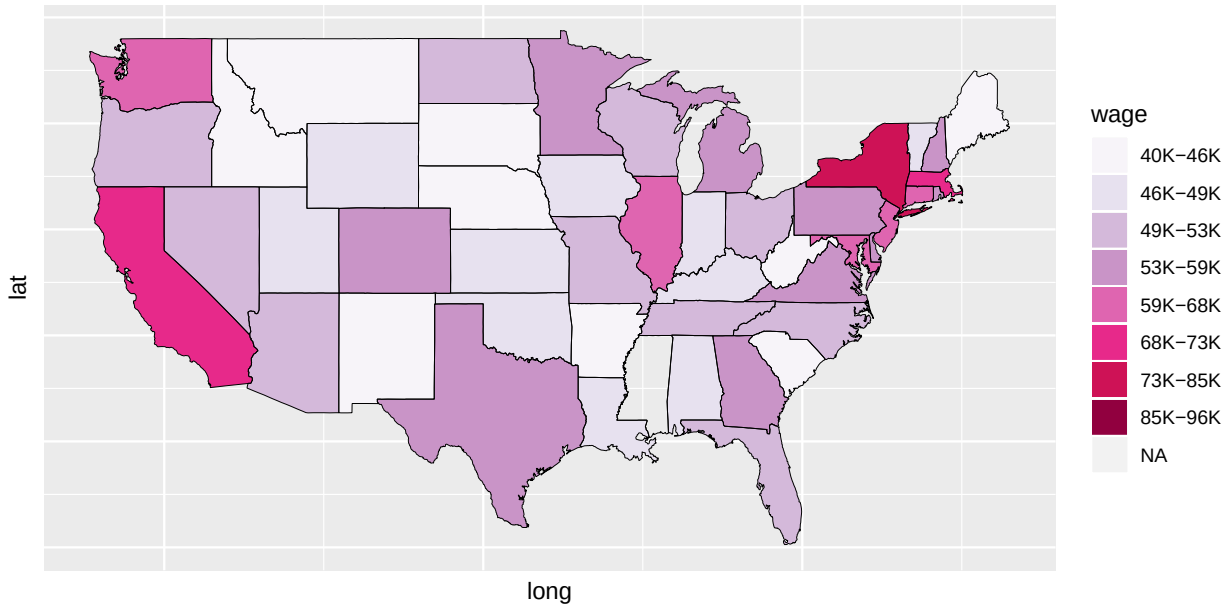
```
##        long      lat group order   state  county
## 1 -86.50517 32.34920     1     1 Alabama Autauga
## 2 -86.53382 32.35493     1     2 Alabama Autauga
## 3 -86.54527 32.36639     1     3 Alabama Autauga
## 4 -86.55673 32.37785     1     4 Alabama Autauga
## 5 -86.57966 32.38357     1     5 Alabama Autauga
## 6 -86.59111 32.37785     1     6 Alabama Autauga
```

1.7.3 connect map data above with `d.state`

```
chor_state <- left_join(state_df, d.state, by ='state')

ggplot(chor_state, aes(long, lat, group = group)) +
  geom_polygon(aes(fill =wage))+
```

```
  geom_path(color = 'black', size =.2)+
  scale_fill_brewer(palette = "PuRd")+
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank())
```

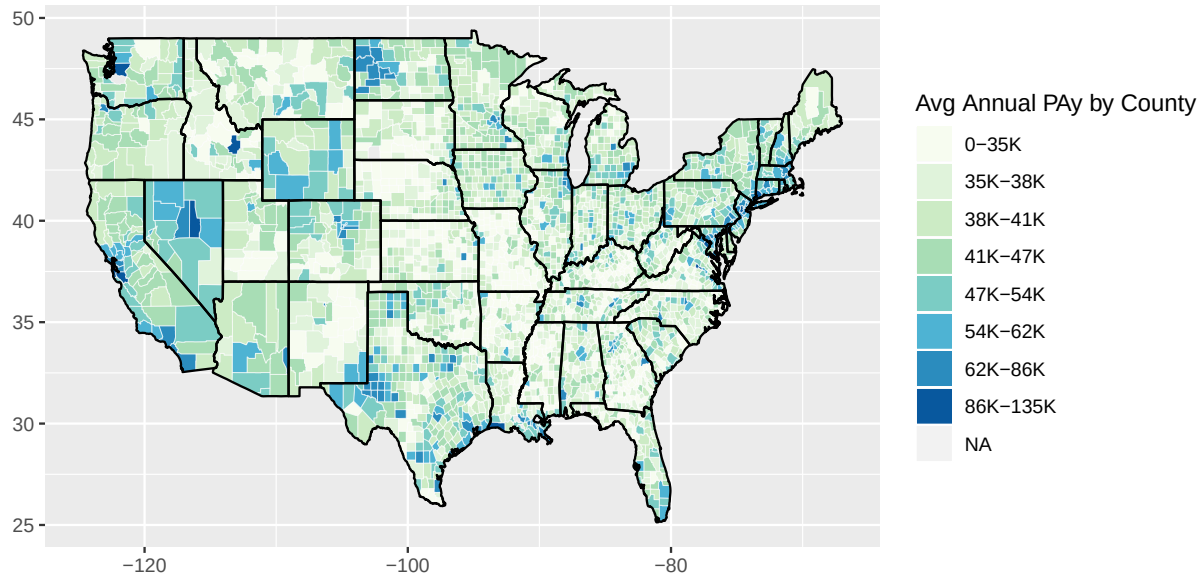


1.7.4 connect map data above with `d.state`

```
chor_cty <- left_join(cty_df, d.county,by = c("state", "county"))

ggplot(chor_cty, aes(long, lat, group = group)) +
  geom_polygon(aes(fill =wage))+
  geom_path(color = 'white', size =.2, alpha = .5)+
  geom_polygon(data = state_df, color  = "black", fill = NA)+

  scale_fill_brewer(palette = "GnBu")+
  labs(x= '', y='',fill = 'Avg Annual PAy by County')
```




```
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank())
```

```
## List of 4
##  $ axis.text.x : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.text.y : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x: list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.y: list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

1.8 Geography from industry level

According to North American Industry Classification System, we chose to focus on and visualize the geographical distribution of employment in the private sector across the 4 industries:
* **(NIACS 11)** Agriculture, forestry, fisheries, hunting;
* **(NIACS 21)** Mining, quarrying, oil, and natural gas extraction;
* **(NIACS 52)** Finance and insurance;
* **(NIACS 54)** Professional technical service.

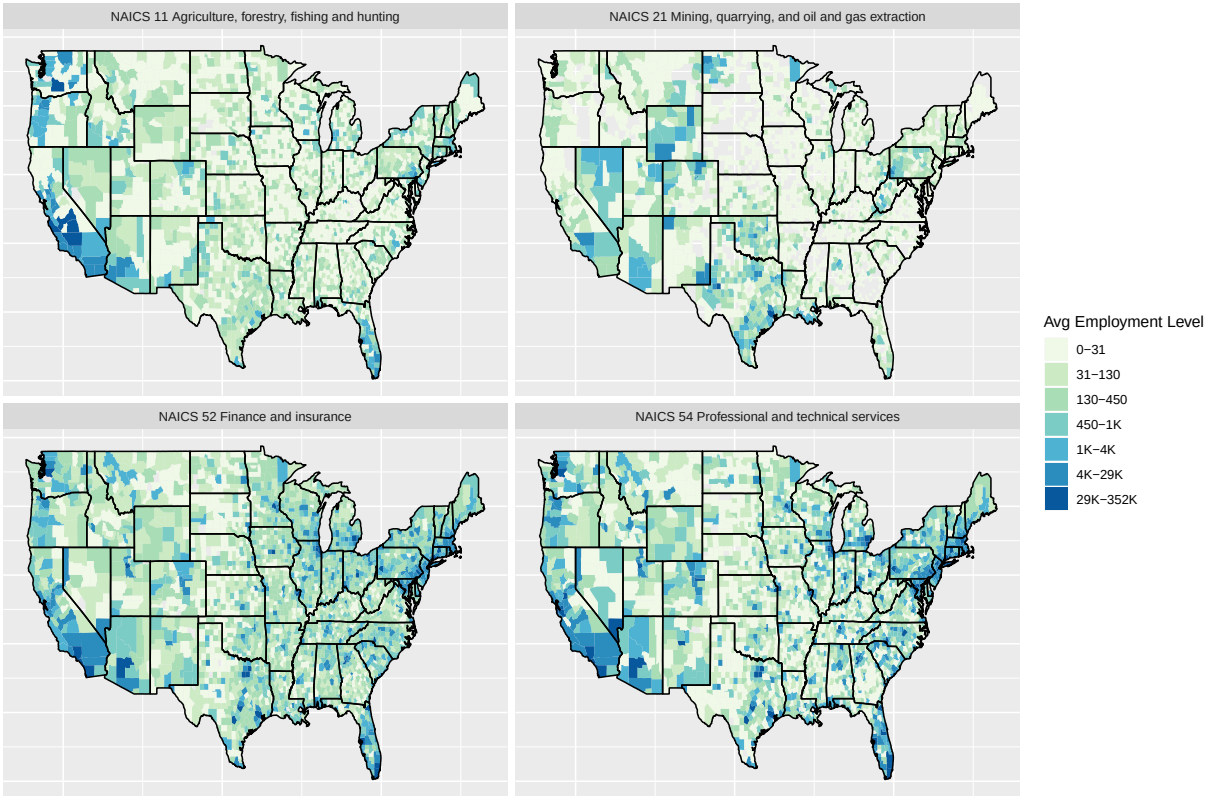
1.8.1 `filter` certain industry data

```
d.sectors <- filter(ann2018_full,
                    industry_code %in% c(11,21,52,54),
                    own_code == 5, # private sector
                    agglvl_code == 74 # county level
                    )%>%
```

```
  select(state, county, industry_code, own_code, agglvl_code,
         industry_title, own_title, avg_annual_pay, annual_avg_emplvl)%>%
  mutate(wage = Discretize(avg_annual_pay),
         emplevel = Discretize(annual_avg_emplvl ) )
d.sectors <- filter(d.sectors, !is.na(industry_code))
```

1.8.2 visualization

```
chor_ind_cty <- left_join(cty_df, d.sectors)
chor_ind_cty <- filter(chor_ind_cty, !is.na(industry_code))
ggplot(chor_ind_cty, aes(long, lat, group = group))+
  geom_polygon( aes(fill= emplevel)) +
  geom_polygon(data = state_df, color = 'black', fill =NA)+
  scale_fill_brewer(palette = 'GnBu')+
  facet_wrap(~industry_title, ncol =2, as.table = T)+
  labs(fill = 'Avg Employment Level', x= '', y= '')+
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank())
```



1.9 Time series map of geospatial

1.9.1 import data from 2008 to 2018

```
library(dplyr)
getdata <- function(zipfile){
  unzip(file.path('data/',zipfile), exdir = 'data')
  csvfile <- gsub('zip', 'csv',zipfile)
  csvfile <- gsub('_', '.',csvfile)
  dat <- fread(file.path('data/',csvfile))

  dat <- left_join(dat, myarea)
  dat %<>% filter(agglvl_code == 70) %>% # county level
    select(state , county , avg_annual_pay)

  return(dat)
}

# store 11 yrs data in list
files <- dir('data', pattern = 'annual_singlefile.zip') # find pattern
n = length(files)
dat_list <- vector('list',n)
for (i in 1:n){
 dat_list[[i]] <- getdata(files[i])
 names(dat_list)[i] <- substr(files[i],1,4)
}
```

1.9.2 Connect all the data together to create a unified discretization interval

```
annpay <- ldply(dat_list)
breaks <- quantile(annpay$avg_annual_pay,
                   c(seq(0,.8,by=.2), .9, .95,.99,1))
```

1.9.3 visualization

```
mychor_time <- function(d, fill_label = ''){

  # the dataset d will have a column named 'outcome'
  chor <- left_join(cty_df,d)
  plt <- ggplot(chor, aes(long, lat, group = group))+
  geom_polygon( aes(fill= outcome)) +
  geom_path(color = 'white', alpha = .5, size =.2)+
  geom_polygon(data = state_df, color = 'black', fill =NA)+
  scale_fill_brewer(palette = 'GnBu')+
  labs(fill = fill_label, x= '', y= '')+
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank())

  return(plt)
}
```

```
# Create a corresponding plot object for 2008-2018 data with a loop
plt_list <- vector('list',n)
for(i in 1:n){
  dat_list[[i]] <- mutate(dat_list[[i]],
```

```
                          outcome = Discretize(avg_annual_pay, breaks = breaks))
  plt_list[[i]] <- mychor_time(dat_list[[i]]) + ggtitle(names(dat_list)[i])
}
```

1.9.4 Generate an animated GIF web page file

```
library(choroplethr)
choroplethr_animate(plt_list)
```

According to the gif, we can see areaEmployment and wealth in North Dakota, Nevada, Wyoming and Northeastern coastal are growingobviously rapidly in last decade.