county

Hassan

February 22, 2018

R Markdown

countytax contains data about the tax obtained from : https://www.irs.gov/statistics/soi-tax-stats-county-data-2015

```
  library(readr)
  countytax <- read_csv("https://www.irs.gov/pub/irs-soi/15incyallnoagi.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   STATEFIPS = col_character(),
##   STATE = col_character(),
##   COUNTYFIPS = col_character(),
##   COUNTYNAME = col_character(),
##   AGI_STUB = col_integer()
## )
```

```
## See spec(...) for full column specifications.
```

now we read in the spatial data. This was obtained from https://www.census.gov/geo/maps-data/data/tiger-line.html

although more recent data is available, to conform to out tax information data we use the one from 2015.

```
library(rgdal)
```

```
## Loading required package: sp
```

```
## rgdal: version: 1.2-16, (SVN revision 701)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.2.2, released 2017/09/15
##  Path to GDAL shared files: /usr/share/gdal/2.2
##  GDAL binary built with GEOS: TRUE
##  Loaded PROJ.4 runtime: Rel. 4.9.2, 08 September 2015, [PJ_VERSION: 492]
##  Path to PROJ.4 shared files: (autodetected)
##  Linking to sp version: 1.2-7
```

```
library(sf) #to add to data column
```

```
## Linking to GEOS 3.5.1, GDAL 2.2.2, proj.4 4.9.2
```

```
#also going to try read in as multipolygon using sf
shpdata <- st_read(dsn = "tl_2015_us_county", layer = "tl_2015_us_county", stringsAsFactors = FALSE)
```

```
## Reading layer `tl_2015_us_county' from data source `/home/hassan/Documents/R/Maps and county data/tl
## Simple feature collection with 3233 features and 17 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -179.2311 ymin: -14.60181 xmax: 179.8597 ymax: 71.44106
## epsg (SRID):    4269
## proj4string:    +proj=longlat +datum=NAD83 +no_defs
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
statenames <- read_csv("https://raw.githubusercontent.com/jasonong/List-of-US-States/master/states.csv")
```

```
## Parsed with column specification:
## cols(
##   State = col_character(),
##   Abbreviation = col_character()
## )
```

```
#merge(countytax, statenames, by.x="STATE", by.y = "Abbreviation")
countytax <- merge(countytax, statenames, by.x="STATE", by.y = "Abbreviation")
countytax3 <- select(countytax, State, STATE, STATEFIPS, COUNTYFIPS, COUNTYNAME, N1 , N19300, A19300, N
```

A19300 = Mortgage Interest paid A18425 = state and local income tax A07220 = child tax credit A18500 = Real Estate tax A59660 = Earned Income credit

A01000 = Net capital gain (less loss) A00300 = Taxable interest amount A00650 = Qualified Dividends amount A04470 = Total itemized deductions A10600 = Total Tax payments amount

Now we move on with plotting. Its possible to plot the maps with ggplot. An example that would plot states by their color is:

ggplot(shpdata) + geom_sf(aes(fill = STATEFP),size=0.1, color = "grey40")

To start merging data, we I'll create a unique id by combining the stateid and countyid. The only data we need from shp is the one about the geometry.

```
#first give the unique id to the tax data.
countytax3$uniqueID <- paste(countytax3$STATEFIPS,countytax3$COUNTYFIPS)

#then give the unique id to the location data.
shpdata$uniqueID <- paste(shpdata$STATEFP, shpdata$COUNTYFP)

#now we create a temporary version of shpdata with only the geometry and unique id
tmp <- select(shpdata, uniqueID, geometry)

#now we complete the merge
countytax4 <- merge(tmp, countytax3)

#delete all the old data files
remove(countytax)
remove(countytax3)
remove(shpdata)
remove(statenames)
remove(tmp)

#also need to rename the columns so they make more sense.
```

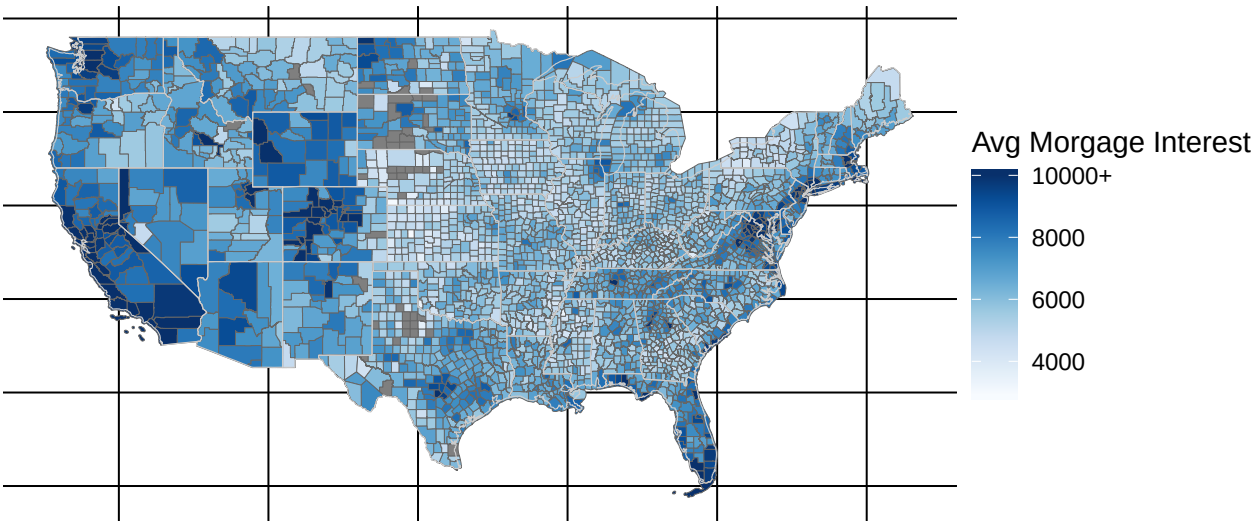
```
#A19300 = Mortgage Interest paid
#A18425 = state and local income tax
#A07220 = child tax credit
#A18500 = Real Estate tax
#A59660 = Earned Income credit
names(countytax4) = c("UniqueID", "State", "StateAbb", "StateFIPS", "CountyFIPS", "CountyName","NumRetu

#also create new columns for averages. This is done by amount divided by number of filings. Also, all t
countytax4$AvgMortgageI <- (countytax4$MortgageI)/(countytax4$NumMortgageI)*1000
countytax4$AvgChildTC <- (countytax4$ChildTC)/(countytax4$NumChildTC)*1000
countytax4$AvgStateIT <- (countytax4$StateIT)/(countytax4$NumStateIT)*1000
countytax4$AvgRealET <- (countytax4$RealET)/(countytax4$NumRealET)*1000
countytax4$AvgEITC <- (countytax4$EITC)/(countytax4$NumEITC)*1000

countytax4$AvgNetCapitalGain <- (countytax4$NetCapitalGain)/(countytax4$NumNetCapitalGain)*1000
countytax4$AvgTaxableInterest <- (countytax4$TaxableInterest)/(countytax4$NumTaxableInterest)*1000
countytax4$AvgQDividends <- (countytax4$QDividends)/(countytax4$NumQDividends)*1000
countytax4$AvgItemizedDeductions <- (countytax4$ItemizedDeductions)/(countytax4$NumItemizedDeductions)*
countytax4$AvgTaxPayments <- (countytax4$TaxPayments)/(countytax4$NumTaxPayments)*1000

#also removing hawaii and alaska so theres only mainland US.
countytax <- countytax4[countytax4$State!="Hawaii", ]
countytax <- countytax[countytax$State!="Alaska", ]

remove(countytax4)
```

Now we have a data set called countytax4 which has all the information we currently need. Time to work on the plots.

```
#import necessary libraries
library(ggplot2) #to make the plot
library(RColorBrewer) #color selection

#creating a copy so the original data set is not changed
tmpdata <- countytax
```

plotting using the same method as Sirui did earlier (NOTE: This takes some time as:

```
#for mortgage data

tmpdata$AvgMortgageI[tmpdata$AvgMortgageI > 10000] <- 10001

ggplot(tmpdata) + geom_sf(aes(fill = tmpdata$AvgMortgageI),size=0.05, color = "grey40")+  borders(databa
```




```
#for child care tax credit
tmpdata$AvgChildTC[tmpdata$AvgChildTC > 1600] <- 1601

ggplot(tmpdata) + geom_sf(aes(fill = tmpdata$AvgChildTC),size=0.05, color = "grey40")+  borders(databas
```