R-bloggers

R news and tutorials contributed by (552) R bloggers

- Home
- About
- add your blog!
- Contact us
- RSS
- R jobs���

Welcome!

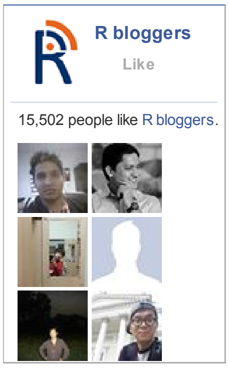
Here you will find daily **news and tutorials about R**, contributed by over 450 bloggers. You can subscribe for e-mail updates:

Your e-mail here

Subscribe

18077 readers BY FEEDBURNER

And get updates to your Facebook:



R bloggers

Like

15,502 people like R bloggers.

If you are an R blogger yourself you are invited to add your own R content feed to this site (**Non-English** R bloggers should add themselves- here)

Search & Hit Enter

Jobs for R-users

- SAS Analyst role who has experience with "R" Language
- Data Scientist
- Senior programmer / business analyst for customized solutions
- ATM Data Scientist
- Part-time R Instructor for Data Science Retreat

Popular Searches

- heatmap
- hadoop
- web scraping
- twitter

- maps
- ggplot
- finance
- shiny
- pca
- excel
- ggplot2
- googlevis
- RStudio
- trading
- animation
- boxplot
- eclipse
- LaTeX
- quantmod
- knitr
- time series
- tutorial
- sweave
- alt=
- market research
- how to import image file to R
- rattle
- GIS
- ecdf
- rcmdr

Recent Posts

- Gender gap and visualisation challenge @ useR!2014
- Jun 26-27, 2014 – Introduction to Data Science with R in NYC
- An infographic comparing R, SAS and SPSS
- Shiny 0.10
- Paper: The OpenCPU system – towards a universal interface for scientific computing through separation of concerns
- New R package hazus: Damage functions from FEMA’s HAZUS software for use in modeling financial losses from natural disasters
- Rcpp11 conference marathon
- Quickly export multiple R objects to an Excel Workbook
- Multilevel modeling of community composition with imperfect detection
- Interactive documents: An incredibly easy way to use Shiny
- Contest: Prizes for Best R User

Groups Plotting Code

- Keep calm and #include <Rcpp11>
- Conditional Distributions from some Elliptical Vectors
- Free Geometric Mophometrics – a Summary
- Where World Cup players actually come from

Other sites

- Jobs for R-users
- Statistics of Israel
- SAS blogs

ggplot2 Chloropleth of Supreme Court Decisions: A Tutorial

July 4, 2013
By tylerrinker

(This article was first published on **TRinker's R Blog » R**, and kindly contributed to R-bloggers)

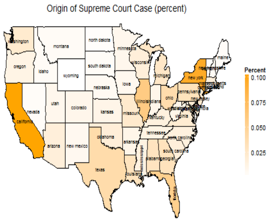
I don't do much GIS but I like to. It's rather enjoyable and involves a tremendous skill set. Often you will find your self grabbing data sets from some site, scraping, data cleaning and reshaping, and graphing. On the ride home from work yesterday I heard an NPR talk about the Supreme Court decisions being very close with this court. This got me wondering if there is a data base with this information and the journey began. This tutorial is purely exploratory but you will learn to:

1. Grab .zip files from a data base and read into R
2. Clean data
3. Reshape data with reshape2
4. Merge data sets
5. Plot a chloropleth map in ggplot2
6. Arrange several grid plots with gridExtra

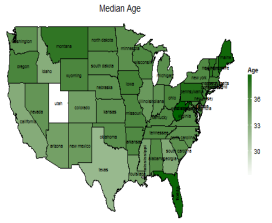
I'm lazy and like a good challenge. I challenged myself to not manually open a file so I downloaded Biobase from bioconductor to open the pdf files for the codebook. Also I used my own package qdap because it had some functions I like and I'm used to using them. This blog post was created in the dev. version of the reports package using the **wordpress_rmd** template.

Also note that this is designed to be instructional. I broke the code up into chunks with an explanation in between, This is extremely annoying if you just want to run the code so for this sort of person I have provided the code here. Thanks to Bryan Goodrich of talkstats.com who has many programming and statistics talents and shared much with me on GIS topics.

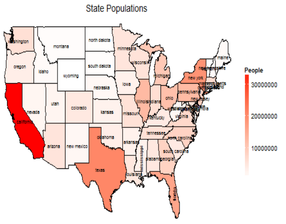
Enjoy!



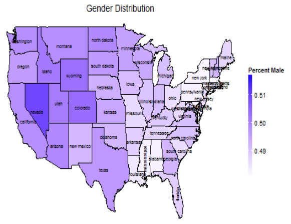
Origin of Supreme Court Case (percent)



Median Age



State Populations



Gender Distribution

Load Packages

```
## download Biobase so we don't have to manua
source("http://bioconductor.org/biocLite.R")
biocLite("Biobase", suppressUpdates = TRUE)
library(qdap)
## Load initial required packages
lapply(qcv(ggplot2, maps, ggthemes, Biobase),
```

Get Data

The Supreme Court Codebook and opened without clicking

```
## download the pdf code book and open it
url_dl(SCDB_2012_01_codebook.pdf, url = "http
openPDF(file.path(getwd(), "SCDB_2012_01_code
```

The Supreme Court Data; learn to download and open a zip file

```
temp <- tempfile()
download.file("http://scdb.wustl.edu/_brickFi
    temp)
dat <- read.csv(unz(temp, "SCDB_2012_01_caseC
unlink(temp)
htruncdf(dat, 6, 6)
```

```
##   caseId docket caseIs voteId dateDe decisi usCite sctCit ledCit lexisC
## 1 1946-0 1946-0 1946-0 1946-0 11/18/      1 329 U. 67 S.  91 L.  1946 U
## 2 1946-0 1946-0 1946-0 1946-0 11/18/      1 329 U. 67 S.  91 L.  1946 U
## 3 1946-0 1946-0 1946-0 1946-0 11/18/      1 329 U. 67 S.  91 L.  1946 U
## 4 1946-0 1946-0 1946-0 1946-0 11/25/      7 329 U. 67 S.  91 L.  1946 U
## 5 1946-0 1946-0 1946-0 1946-0 11/25/      1 329 U. 67 S.  91 L.  1946 U
## 6 1946-0 1946-0 1946-0 1946-0 11/25/      1 329 U. 67 S.  91 L.  1946 U
##   term natura  chief docket caseNa dateAr dateRe petiti petiti respon
## 1 1946   1301 Vinson     24 HALLIB 1/9/19 10/23/    198   <NA>    172
## 2 1946   1301 Vinson     12 CLEVEL 10/10/ 10/17/    100   <NA>     27
## 3 1946   1301 Vinson     21 CHAMPL 11/8/1 10/18/    209   <NA>     27
## 4 1946   1301 Vinson     26 UNITED 1/31/1 10/25/     27   <NA>    170
## 5 1946   1301 Vinson     50 UNITED 10/25/            27   <NA>    176
## 6 1946   1301 Vinson     46 RICHFI 10/24/           198   <NA>      4
##   respon jurisd adminA adminA threeJ caseOr caseOr caseSo caseSo lcDisa
## 1   <NA>      6   <NA>   <NA>      0     51      6     29   <NA>      0
## 2   <NA>      1   <NA>   <NA>      0    123     52     30   <NA>      0
## 3   <NA>      2     66   <NA>      1    107     42    107     42      0
## 4   <NA>      1   <NA>   <NA>      0      3   <NA>      3   <NA>      0
## 5   <NA>      1   <NA>   <NA>      0      3   <NA>      3   <NA>      0
## 6      6      2    117      6      0    302      6    300      6      1
##   certRe lcDisp lcDisp declar caseDi caseDi partyW preced voteUn issue
## 1     11      2      1      1      3      0      1      0      0 80180
## 2      4      2      1      1      2      0      0      0      0 10500
## 3      1   <NA>      2      1      2      0      0      0      0 80250
## 4     10   <NA>      2      1      2      0      0      0      0 20150
## 5      2   <NA>      2      1      3      0      1      0      0 80060
## 6      1      3      2      3      3      0      1      0      0 80100
##   issueA decisi decisi author author lawTyp lawSup lawMin majOpi majOpi
## 1      8      2      0      4   <NA>      6    600 35 U.S     78     78
## 2       1       1       0       4    <NA>       6     600 18 U.S      81      87
## 3       8       2       0       1    <NA>       2     207             84      78
## 4       2       2       0       4    <NA>       6     600 49 Sta      87      87
## 5       8       1       0       7    <NA>    <NA>    <NA>             78      78
## 6       8       1       0       2    <NA>       1     129             81      87
##    splitV majVot minVot
## 1       1      8      1
## 2       1      6      3
## 3       1      5      4
## 4       1      5      3
## 5       1      6      3
## 6       1      7      1
```

Source a Codebook for State Keys Used By Supreme Court Data

```
source("http://copy.com/zEtAXJC8tG7yv7Zz")
head(state.key)
```

```
##   code          state
## 1    1        alabama
## 2    2         alaska
## 3    3 american samoa
## 4    4        arizona
## 5    5       arkansas
## 6    6     california
```

Clean Data

Clean Supreme Court Data

```
dat$state <- lookup(dat$caseOriginState, stat
dat2 <- dat[!is.na(dat$state), ]
dat_state <- data.frame(with(dat2, prop.table
head(dat_state)
```

```
##          state     Freq
## 1      alabama 0.030063
## 2       alaska 0.005010
## 3      arizona 0.017954
## 4     arkansas 0.010438
## 5 california 0.103549
## 6     colorado 0.009603
```

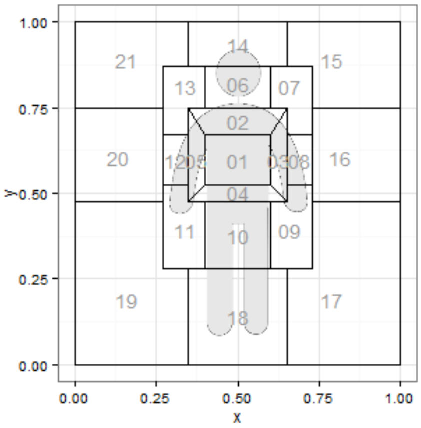
Before I get started with any sizable graphing project I start with the bare minimum and add to the code. Dr. Hadley Wickham has provided just such a minimal example for chloropleth mapping on pages 10-11 in the Changes and Additions guide

Minimal Chloropleth Example

```
crimes <- data.frame(state = tolower(rownames

states_map <- map_data("state")

ggplot(crimes, aes(map_id = state)) +
    geom_map(aes(fill = Murder), map = states
    expand_limits(x = states_map$long, y = st
```



Map the Data

Here I use the maps package to get the state longitude and latitudes for the shapes.

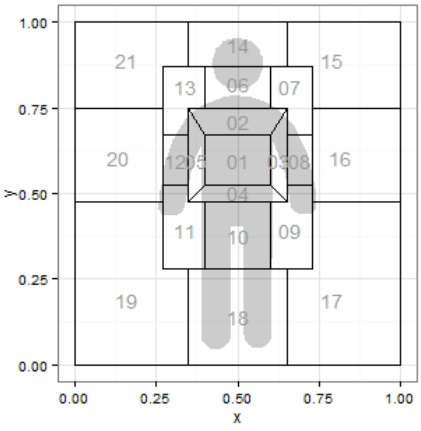
```
states_map <- map_data("state")
head(states_map)
```

```
##     long   lat group order  region subregion
## 1 -87.46 30.39     1     1 alabama      <NA>
## 2 -87.48 30.37     1     2 alabama      <NA>
## 3 -87.53 30.37     1     3 alabama      <NA>
## 4 -87.53 30.33     1     4 alabama      <NA>
## 5 -87.57 30.33     1     5 alabama      <NA>
## 6 -87.59 30.33     1     6 alabama      <NA>
```

Plot the Data

Use `map_id` to map the states. `expand_limits` is filled with the **states_map** data set's latitude and longitude.

```
ggplot(dat_state, aes(map_id = state)) +
    geom_map(aes(fill = Freq), map = states_
    expand_limits(x = states_map$long, y = s
    theme_few()+
    theme(legend.position = "bottom",
         axis.ticks = element_blank(),
         axis.title = element_blank(),
         axis.text =  element_blank()) +
    scale_fill_gradient(low="white", high="b
    guides(fill = guide_colorbar(barwidth =
    ggtitle("Chloropleth Supreme Court")
```



Notice the little trick of moving the legend to the bottom and making it narrow? This is done with:

```
theme(legend.position = "bottom")

## and

guides(fill = guide_colorbar(barwidth = 10, b
```

Generate labels

I said to myself, "Self I forgot all the state names; this needs labels". Here is a question on label centering I asked at stackoverflow. The trick to supplying text data to ggplot is it has to be in a data.frame format. Note that you need the state (region) name, latitude and longitude. I also added angle to be able to manually twist the angles of individual labels. The trick here was to take the mean of the range of the shape file lats/longs in an answer provided by Andrie from stackoverflow.com. Note that it is extremely important that you are now adding a new date set to ggplot and you need to unmap the `map_id` with `map_id = NULL` otherwise ggplot2 will become enraged and refuse to comply with what you consider to be a reasonable request.

```
cnames <- aggregate(cbind(long, lat) ~ region
cnames$angle <- 0
head(cnames)
```

```
##        region    long   lat angle
## 1     alabama  -86.69 32.63     0
## 2     arizona -111.92 34.17     0
## 3    arkansas  -92.14 34.75     0
## 4  california -119.26 37.28     0
## 5    colorado -105.55 39.00     0
## 6 connecticut  -72.75 41.53     0
```

Plot With Labels 1

```
ggplot(dat_state, aes(map_id = state)) +
    geom_map(aes(fill = Freq), map = states_
    expand_limits(x = states_map$long, y = s
    theme_few()+
    theme(legend.position = "bottom",
         axis.ticks = element_blank(),
         axis.title = element_blank(),
         axis.text =  element_blank()) +
    scale_fill_gradient(low="white", high="b
    guides(fill = guide_colorbar(barwidth =
    geom_text(data=cnames, aes(long, lat, la
        angle=angle, map_id =NULL), size=2.5
    ggtitle("Chloropleth Supreme Court (With
```

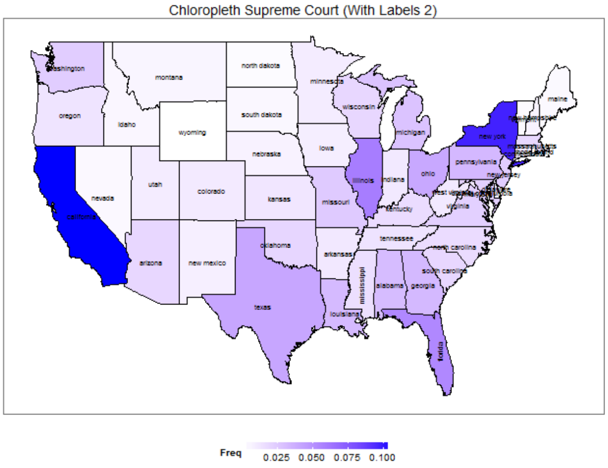
You can manually adjust the labels with indexing the dataframe cnames.

Manually Move State Locations and Change Angle

```
cnames[11, c(2:3)] <- c(-114.5, 43.5)  # alte
cnames[17, 3] <- 30.75  # alter louisiana's c
cnames[21, c(2:3)] <- c(-84.5, 43)  # alter m
cnames[23, 4] <- 90  # alter mississippi's an
cnames[9, c(2, 4)] <- c(-81.5, 90)  # alter f
```

Plot With Labels 2

```
ggplot(dat_state, aes(map_id = state)) +
    geom_map(aes(fill = Freq), map = states_
    expand_limits(x = states_map$long, y = s
    theme_few()+
    theme(legend.position = "bottom",
         axis.ticks = element_blank(),
         axis.title = element_blank(),
         axis.text =  element_blank()) +
    scale_fill_gradient(low="white", high="b
    guides(fill = guide_colorbar(barwidth =
    geom_text(data=cnames, aes(long, lat, la
        angle=angle, map_id =NULL), size=2.5
    ggtitle("Chloropleth Supreme Court (With
```



Further Exploring the Data

From there a new thought emerged. It seemed that a few states had the most percentage of Supreme Court cases originating in them. I wondered if this had something to do with population. I wanted to compare a population chloropleth. This meant grabbing more data and the US Census database is just the place.

Download and read in a zip file just like the Supreme Court Data.

```
>## Download the US census database
temp <- tempfile()
download.file("http://www2.census.gov/census_
    temp)
demo <- read.csv(unz(temp, "2khxx.csv"))
unlink(temp)
```

Clean Data

Clean Census Data and Merge With dat_state From Above

```
## browseURL("http://www2.census.gov/census_
vars <- data.frame(codes = qcv(X281421906, X
    var = qcv(pop, male, female, med_age, st

colnames(demo)[colnames(demo) %in% vars[, 1]

demo$state <- tolower(demo$state)
demo <- demo[, colnames(demo) %in% vars[, 2]
demo <- demo[demo$state %in% tolower(state.n

## Merge it
dat_state <- merge(demo, dat_state, by = "st
```

One thing led to another and before I knew it I decided to include male to female ratio and median age in the analysis. My first approach (which I did not like) was to combine all the data into a long format data set that I could pass to `ggplot2` and then facet by the chosen variables. This required me to use a common scale for the variables. I used the `apply` function to `scale` the variables.

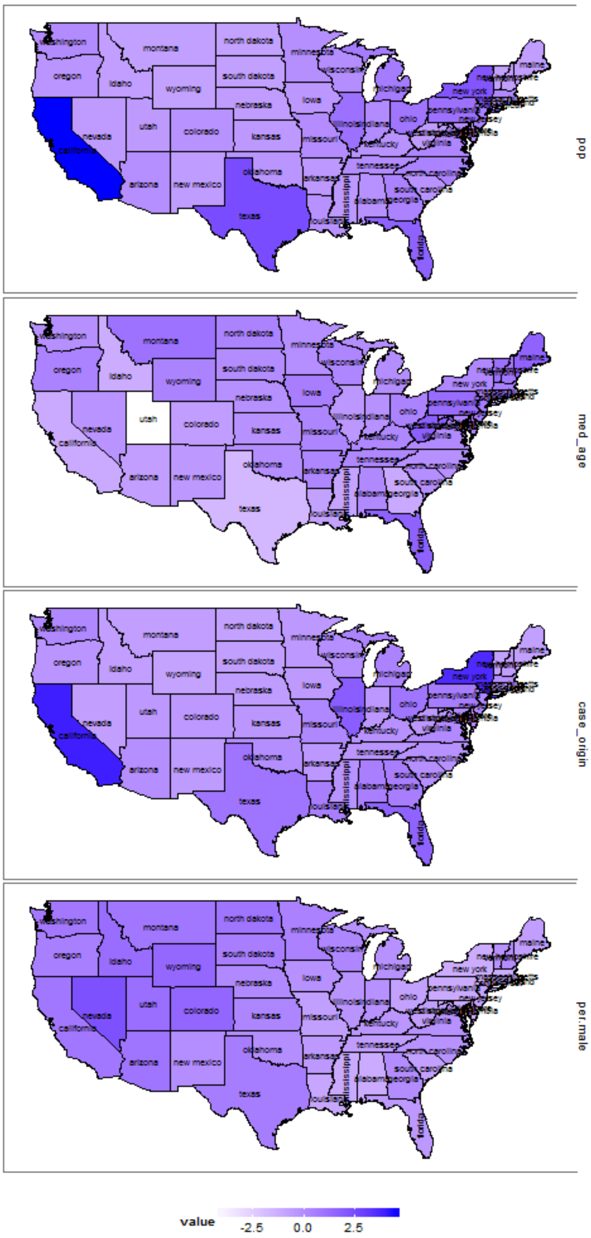
Clean Census Data and Reshape it using the `melt` function from Reshape2.

```
library(reshape2)
dat_state <- transform(dat_state, per.male =
colnames(dat_state)[6] <- "case_origin"
dat_state2 <- melt(data.frame(dat_state[, 1,
    -c(1, 3:4)], 2, scale)))
head(dat_state2)
```

```
##        state variable    value
## 1    alabama      pop -0.18913
## 2     alaska      pop -0.80673
## 3    arizona      pop -0.07863
## 4   arkansas      pop -0.47588
## 5 california      pop  4.56783
## 6   colorado      pop -0.21271
```

Faceted Plot Attempt 1

```
ggplot(dat_state2, aes(map_id = state)) +
    geom_map(aes(fill = value), map = states
    expand_limits(x = states_map$long, y = s
    theme_few()+
    theme(legend.position = "bottom",
         axis.ticks = element_blank(),
         axis.title = element_blank(),
         axis.text =  element_blank()) +
    scale_fill_gradient(low="white", high="b
    guides(fill = guide_colorbar(barwidth =
    geom_text(data=cnames, aes(long, lat, la
        angle=angle, map_id =NULL), size=2.5
    facet_grid(variable~.)
```



A Hunger for Better Display

This was unsatisfying in that I had to use a common scale and the meaning was lost. Also I couldn't control individual map colors easily (though I'm sure there's a way). I decided to instead create 4 separate plots and feed them to `grid.arange` of the `gridExtra` package (a compliment to ggplot2).

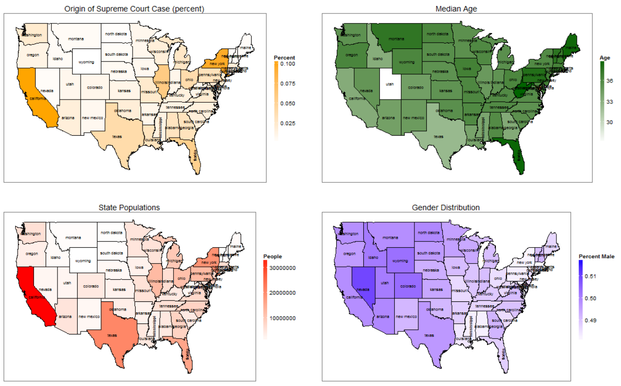
```
plot1 <- ggplot(dat_state, aes(map_id = stat
    geom_map(aes(fill = case_origin), map =
    expand_limits(x = states_map$long, y = s
    theme_few()+
    theme(axis.ticks = element_blank(),
         axis.title = element_blank(),
         axis.text =  element_blank()) +
    scale_fill_gradient(low="white", high="o
    guides(fill = guide_colorbar(barwidth =
    geom_text(data=cnames, aes(long, lat, la
        angle=angle, map_id =NULL), size=2.5
    ggtitle("Origin of Supreme Court Case (p

plot2 <- ggplot(dat_state, aes(map_id = stat
    geom_map(aes(fill = pop), map = states_m
    expand_limits(x = states_map$long, y = s
    theme_few()+
    theme(axis.ticks = element_blank(),
         axis.title = element_blank(),
         axis.text =  element_blank()) +
    scale_fill_gradient(low="white", high="r
    guides(fill = guide_colorbar(barwidth =
    geom_text(data=cnames, aes(long, lat, la
        angle=angle, map_id =NULL), size=2.5
    ggtitle("State Populations")

plot3 <- ggplot(dat_state, aes(map_id = stat
    geom_map(aes(fill = med_age), map = stat
    expand_limits(x = states_map$long, y = s
    theme_few()+
    theme(axis.ticks = element_blank(),
         axis.title = element_blank(),
         axis.text =  element_blank()) +
    scale_fill_gradient(low="white", high="d
    guides(fill = guide_colorbar(barwidth =
    geom_text(data=cnames, aes(long, lat, la
        angle=angle, map_id =NULL), size=2.5
    ggtitle("Median Age")

plot4 <- ggplot(dat_state, aes(map_id = stat
    geom_map(aes(fill = per.male), map = sta
    expand_limits(x = states_map$long, y = s
    theme_few()+
    theme(axis.ticks = element_blank(),
         axis.title = element_blank(),
         axis.text =  element_blank()) +
    scale_fill_gradient(low="white", high="b
    guides(fill = guide_colorbar(barwidth =
    geom_text(data=cnames, aes(long, lat, la
        angle=angle, map_id =NULL), size=2.5
    ggtitle("Gender Distribution")

library(gridExtra)
grid.arrange(plot1, plot3, plot2, plot4, nco
```



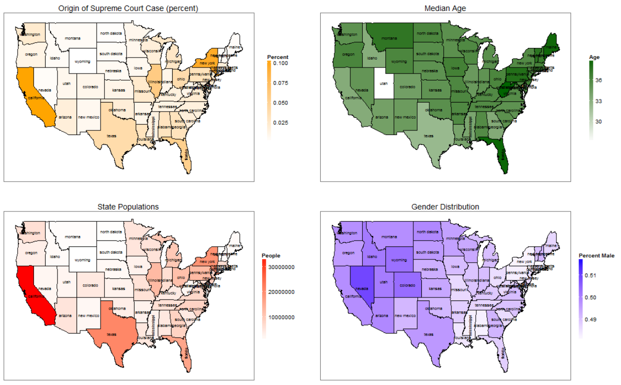
I did not like the alignment of the plot edges and didn't know how to solve the problem. I asked on stackoverflow.com and Kohske gave the following approaches in his response:

Using `grid.draw` and Aligning Plot Edges

```
library(gtable)

p1 <- ggplotGrob(plot1)
p2 <- ggplotGrob(plot2)
p3 <- ggplotGrob(plot3)
p4 <- ggplotGrob(plot4)

library(gtable)
grid.draw(cbind(rbind(p1, p2, size="last"), r
```



Eliminating the Plot Box and Aligning the Legends

This approach looks nice (though the plot box had to be removed as the legend covered it). It is not recomended by Kohske as it's a bit hacky. If someone has a better approach please share.

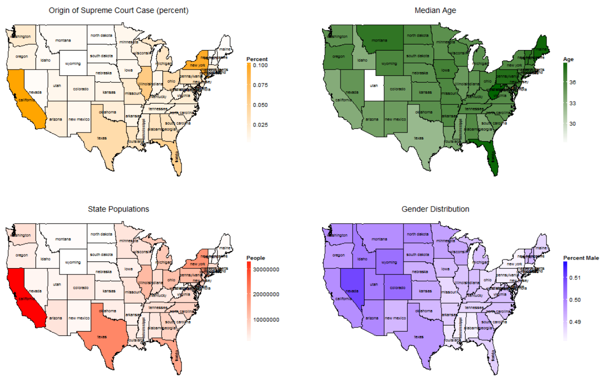
```
plot1b <- plot1 + theme(panel.border = eleme
plot2b <- plot2 + theme(panel.border = eleme
plot3b <- plot3 + theme(panel.border = eleme
plot4b <- plot4 + theme(panel.border = eleme

p1b <- ggplotGrob(plot1b)
p2b <- ggplotGrob(plot2b)
p3b <- ggplotGrob(plot3b)
p4b <- ggplotGrob(plot4b)

gt <- cbind(rbind(p1b, p2b, size="last"), rb

for (i in which(gt$layout$name == "guide-box
  gt$grobs[[i]] <- gt$grobs[[i]]$grobs[[1]]
}

grid.draw(gt)
```



The final out put was very satisfying. I did notice that yes indeed the states of high population also had a high number of Supreme Court cases originating in them. That was sensible. I noted that New York and Alabama seemed to have more Supreme Court cases originating in them in comparison to their populations (but only slightly). There is still a ton of data in the two data sets left to explore (particularly from a time series perspective). Feel free to experiment yourself with the data.

Please be sure to provide feedback in the comments below.

Blog created with the reports package utilizing knitr and RWordPress.

For the knitr html version click here
For the knitr Rmd of the post click here

Comments: 20

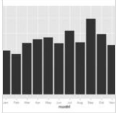
52 Like 10 Tweet 5 g+1

Share

Related aRticles

ggplot2 Tutorial
Here is a link to a wonderful ggplot2 tutorial: http://www.ceb-institute.org/bbs/wp-content/uploads/2011/09/handout_ggplot2.pdf by R.

Saccilotto and the Basel Institute for Clinical Epidemiology and Biostatistics. The tutorial ...



Web Analytics Visualization through ggplot2
During our last webinar , we covered some of the basic ideas behind ggplot2 , the R Visualization package by Dr. Hadley Wickham. In ...

R, knitr & markdown = HTML
Welcome to this demo of how R code and results can be combined into an HTML report. This entire blogpost was generated by using ...

If you got this far, why not **subscribe for updates** from the site?
Choose your flavor: e-mail, twitter, RSS, or facebook...

Like Share 52 Tweet 10 g+1 5

Comments are closed.

Top 3 Posts from the past 2 days

- An infographic comparing R, SAS and SPSS
- Prediction model for the FIFA World Cup 2014
- Quickly export multiple R objects to an Excel Workbook

Search & Hit Enter

Top 9 articles of the week

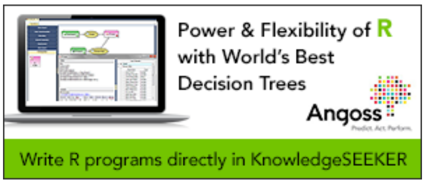
1. Prediction model for the FIFA World Cup 2014
2. Who will win the World Cup and which prediction model?
3. Using apply, sapply, lapply in R
4. Installing R packages
5. The Most Comprehensive Review of Comic Books Teaching Statistics
6. New Version of RStudio: R Markdown v2 and More
7. A suggestion to Windows-based users of R: It may be time to relocate
8. R Markdown v2
9. Basics of Histograms

Sponsors

plotly online R graphing

DataCamp
Learn R Interactively

"Data Scientists making $300,000 a year!"
Wall Street Journal
Come, learn DATA SCIENCE with us...
And become an Expert in Data Analysis!
Instructor-Led Online Classes
24x7 Training Support
Skill Certification
edureka!
LEARN MORE!



Search & Hit Enter

Full list of contributing R-bloggers

R-bloggers was founded by Tal Galili, with gratitude to the R community.

Is powered by WordPress using a bavotasan.com design.