Mapping SAPA data

PMC lab
Northwestern University

Abstract

A guided tour in using the SAPA mapping functions.

To graphically display the SAPA data can be done using a number of external packages as well as a set of functions developed at the PMC lab. This is a short guide to using these sapa functions. Comments are welcome. Suggestions for improvement of the functions are particularly welcome.

Install other packages and then make them active

The most relevant package is the *zip code* and its associated `zip code` database. In addition, the *mvcv* package is used for finding those points within a map.

installing the relevant packages

```
install.packages(zipcode)
install.packages(mgcv}
```

Activate them

Then, once they are installed, they need to be activated every time you start R.

```
library(zipcode)
library{mvcv}
```

Get the SAPA mapping functions

The current working version of the SAPA mapping functions may be sourced from the personality-project.org site. Eventually these should be wrapped into the package that Jason and David are developing at ICAR-project.org

```
source("http://personality-project.org/r/src/sapa.r")
```

---

contact: William Revelle revelle@northwestern.edu
Draft version of October 7, 2013

Using the SAPA mapping function

Getting the zip codes attached to the summary statistics

The `sapa.zip` takes a data file (e.g., IRT.scores) and combines it with the zip codes from the zip code database. The output is a data.frame of the means and ns of the original variables by zip code.

This just needs to be done once for a particular data set. Maps for particular dependent variables are drawn using this data base.

```
s.zip <- sapa.zip(IRT.scores)
dim(s.zip)
colnames(s.zip)
```

```
 3208  100

 [1] "zip"                          "gender"                       "relstatus"
  [4] "age"                          "marstatus"                    "height"
  [7] "BMI"                          "ParentalEdu"                  "weight"
 [10] "country"                      "state.x"                      "ethnic"
 [13] "education"                    "discipline"                   "major"
 [16] "jobstatus"                    "jobfield"                     "occupation"
 [19] "p1occ"                        "p1edu"                        "p2occ"
 [22] "p2edu"                        "Assertiveness"                "Balance"
 [25] "Boldness"                     "Compassion"                   "Enthusiasm"
 [28] "Honesty"                      "Humility"                     "Industriousness"
 [31] "Intellect"                    "Openness"                     "Orderliness"
 [34] "Politeness"                   "Agreeableness"                "Conscientiousness"
 [37] "Extraversion"                 "Integrity"                    "OpennessNewExp"
 [40] "EmotionalStability"           "GunControl"                   "Environmental"
 [43] "RightWingAuthoritarianism"    "g"                            "ANiq"
 [46] "MXiq"                         "R3Diq"                        "VEdiq"
 [49] "gender.n"                     "relstatus.n"                  "age.n"
 [52] "marstatus.n"                  "height.n"                     "BMI.n"
 [55] "ParentalEdu.n"                "weight.n"                     "country.n"
 [58] "state.n"                      "ethnic.n"                     "zip.n"
 [61] "education.n"                  "discipline.n"                 "major.n"
 [64] "jobstatus.n"                  "jobfield.n"                   "occupation.n"
 [67] "p1occ.n"                      "p1edu.n"                      "p2occ.n"
 [70] "p2edu.n"                      "Assertiveness.n"              "Balance.n"
 [73] "Boldness.n"                   "Compassion.n"                 "Enthusiasm.n"
 [76] "Honesty.n"                    "Humility.n"                   "Industriousness.n"
 [79] "Intellect.n"                  "Openness.n"                   "Orderliness.n"
 [82] "Politeness.n"                 "Agreeableness.n"              "Conscientiousness.n"
 [85] "Extraversion.n"               "Integrity.n"                  "OpennessNewExp.n"
 [88] "EmotionalStability.n"         "GunControl.n"                 "Environmental.n"
 [91] "RightWingAuthoritarianism.n" "g.n"                          "ANiq.n"
 [94] "MXiq.n"                       "R3Diq.n"                      "VEdiq.n"
 [97] "city"                         "state.y"                      "latitude"
[100] "longitude"
```

Create a longitude by latitude grid of the data

This next step creates a grid of the means for each zip code for a particular Dependent Variable (e.g., 'g'). Data that are in the same grid are averaged. Note that the larger

the grid size, the more precise the results, but the time taken for this and subsequent calculations increase by the square of the grid dimensions. There is an 'average' option which subtracts the overall weighted average from every cell. This functionally centers the data around that average. The alternative is to have all empty cells have the average.

```
iq.grid <- sapa.grid(data=s.zip,DV='g',grid=300,average=TRUE)
names(iq.grid)
dim(iq.grid$av)
```

```
[1] "average" "total"   "count"
```

```
[1] 300 300
```

Smooth the data

Although not necessary for graphics, smoothing the data makes an overall more pleasing map. The smooth function takes a normal "kernel" and applies it to every cell. That is, nearby cells are weighted as a function of the density of the normal curve. The default value of 9 means that the kernel is a 9 x 9 grid. Each grid step is taken to be half a sd of the normal curve. The miss parameter specifies how much to weight cells with no observations. By increasing the weight, we are increasing the tendency to plot average values for missing cells.

The smoothing function operates on all cells of the grid.

It is useful to compare smoothed data with the raw plots.

```
iq.smooth <-  sapa.smooth(iq.grid,size=11,miss=.05)
dim(iq.smooth)
```

```
 300 300
```

Find the points inside the map.

The procedure for smoothing that I am using will create points outside the map region. This is rectified by using a function from the *mvcv* package, `inSide`. In addition, the `sapa.inside` function converts the grid based data to a list of x, y, and z coordinates that is used by the `sapa.plot` function.

For some reason the inSide function gives a warning message which I think can be ignored.

In the example below, I do this operation for both the smoothed data and the raw data. This will allow for a comparison of the two approaches. The `sapa.inside` returns a list of x, y, and z for plotting.

```
iq.smoothed.inside <- sapa.inside(iq.smooth)

iq.inside <- sapa.inside(iq.grid$average)

 names(iq.smoothed.inside)
```

```
 "x" "y" "z"
```

Now plot the results

We can either plot the raw data points (unsmoothed) or the smoothed data points. Even better, we can now draw image plots of the smoothed or unsmoothed data. These image plots make for much prettier pictures.

This one was done using a 400 x 400 grid and 51 colors.

```
sapa.plot(iq.inside)
 sapa.image(iq.smooth.inside,main="iq by state",n=51,database="state")
```

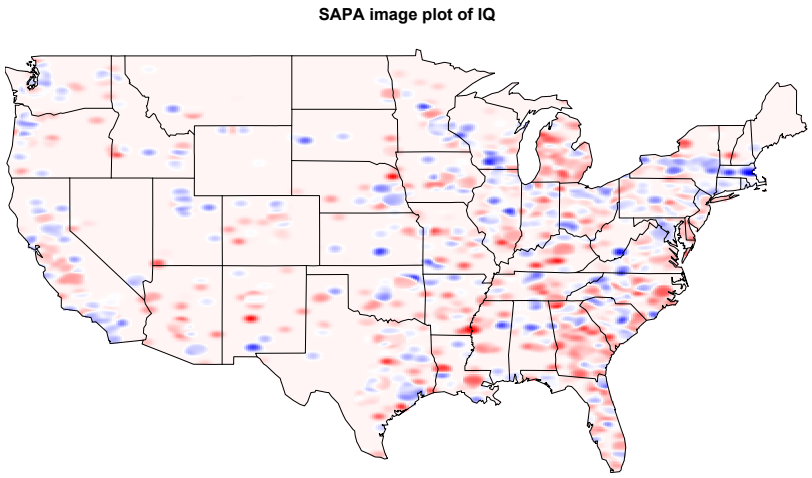

Figure 1. IQ by state, smoothed

And One function to Rule them All

It is obviously tedious to enter the four lines necessary to make pretty pictures, so, lets try the `sapa.combined` function. You must first run the `sapa.zip` function once to prepare the data base. Then a call to `sapa.combined` will do the rest. The function returns the object for remapping with different parameters.

```
my.zips <- sapa.zip(IRT.scores)
g <- sapa.combined(my.zips,'g',gridsize=400,main='ICAR IQ scores from the SAPA project')
```

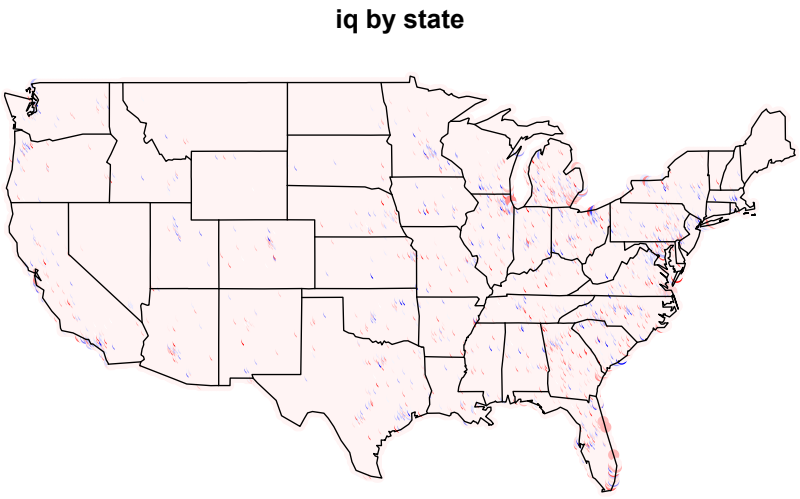

Figure 2. IQ by state, raw

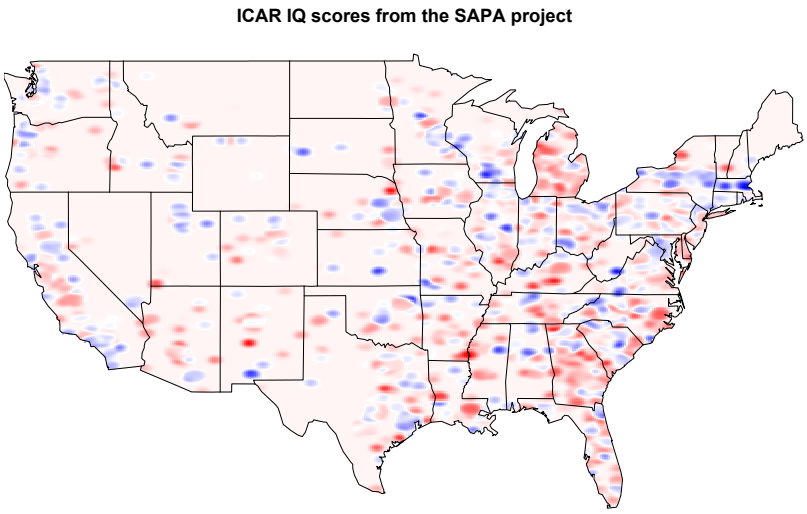

Figure 3. IQ from ICAR for the US. Smoothed size = 11 and with a grid size of 400. By default, the number of colors is the grid size (400).