hw8.R

joebrew

Mon Oct 27 09:23:07 2014

```r
# set local working directory
setwd("/home/joebrew/Documents/uf/phc6194/hw8/")

# load some packages for mapping and colrs
library(rgdal)
```

```
## Loading required package: sp
## rgdal: version: 0.9-1, (SVN revision 518)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 1.10.1, released 2013/08/26
## Path to GDAL shared files: /usr/share/gdal/1.10
## Loaded PROJ.4 runtime: Rel. 4.8.0, 6 March 2012, [PJ_VERSION: 480]
## Path to PROJ.4 shared files: (autodetected)
```

```r
library(RColorBrewer)
library(classInt)
library(sp)


############
# 1. MERGE THE DATA
############

# read in the florida shapefile (exported from arcgis)
fl <- readOGR(".", "hw8")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: ".", layer: "hw8"
## with 67 features and 4 fields
## Feature type: wkbPolygon with 2 dimensions
```

```r
# read in the std rate data
std <- read.csv("std.csv", stringsAsFactors = FALSE)

# compare to see if names are perfect matches
fl$NAME <- toupper(fl$NAME)
table(fl@data$NAME == std$NAME) # they're not all correct
```

```
##
## FALSE  TRUE
##    35    32
```

```r
# get the closest match for each county

fl$name <- NA # create empty vector where we'll put our names
```

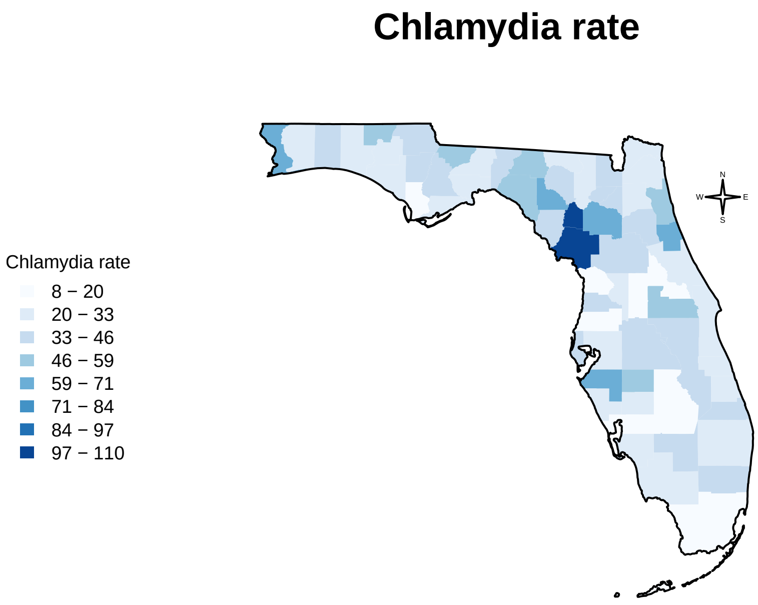
```
std$name <- std$NAME # we'll use lower cases for the matches
for (i in 1:nrow(fl)){
  # see how close (in character changes) each name in fl is to those in STD
  m <- adist(fl$NAME[i],
             std$NAME)
  # get the one with the least differences
  ind <- which.min(m)
  # get the name from std
  best <- std$NAME[ind]
  # assign to fl
  fl$name[i] <- best
}

# Merge the two datasets together
fl@data <- merge(x = fl@data,
                 y = std,
                 by = "name",
                 all.x = TRUE,
                 all.y = FALSE)

############
# 2. CHOROPLETH OF CHLAMYDIA
############

# Create boundary shapefile of just florida
library(maptools)
```

```
## Checking rgeos availability: TRUE
```

```
boundary <- unionSpatialPolygons(fl, rep(1, length(fl@polygons)))
```

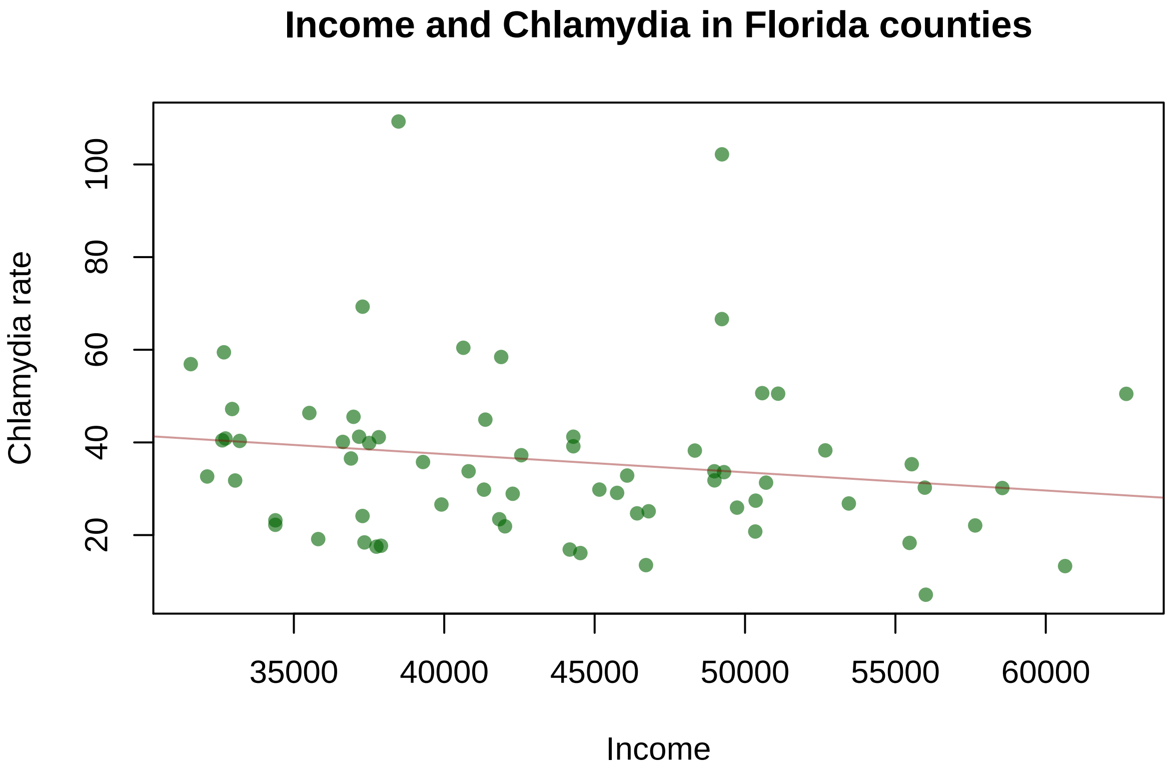
```
## Loading required package: rgeos
## rgeos version: 0.3-8, (SVN revision 460)
##  GEOS runtime version: 3.4.2-CAPI-1.8.2 r3921
##  Polygon checking: TRUE
```

```
# Code for compass rose
#(from http://r-sig-geo.2731867.n2.nabble.com/How-to-diplasy-a-compass-rose-on-a-map-td4509034.html)
compassRose<-function(x,y,rot=0,cex=1) {
  oldcex<-par(cex=cex)
  mheight<-strheight("M")
  xylim<-par("usr")
  plotdim<-par("pin")
  xmult<-(xylim[2]-xylim[1])/(xylim[4]-xylim[3])*plotdim[2]/plotdim[1]
  point.angles<-seq(0,7*pi/4,by=pi/4)+pi*rot/180
  crspans<-rep(c(mheight*3,mheight/2),4)
  xpoints<-cos(point.angles)*crspans*xmult+x
  ypoints<-sin(point.angles)*crspans+y
  polygon(xpoints,ypoints)
  txtxpoints<-cos(point.angles[c(1,3,5,7)])*1.33*crspans[1]*xmult+x
  txtypoints<-sin(point.angles[c(1,3,5,7)])*1.33*crspans[1]+y
  text(txtxpoints,txtypoints,c("E","N","W","S"))
  par(oldcex)
}

# Code for map (original)
TractFun <- function(var,
                     color = "Blues",
                     style = "equal", # or equal/quantile
                     nclr = 8,
                     title = NA,
                     dataPrecision = 0){
  plotclr <- brewer.pal(nclr, color)
  class <- classIntervals(var, nclr, style = style, dataPrecision=dataPrecision) #use "equal" instead
  #class <- classIntervals(0:100, nclr, style="equal")
  colcode <- findColours(class, plotclr)
  legcode <- paste0(gsub(",", " - ", gsub("[[]|[]]|[)]", "", names(attr(colcode, "table")))))
  plot(fl, border=NA, col=colcode)
  plot(boundary, add = TRUE)
  legend("left", # position
         legend = legcode, #names(attr(colcode, "table")),
         fill = attr(colcode, "palette"),
         cex = 0.6,
         border=NA,
         bty = "n",
         title = title)
  compassRose(x = -80.5 ,y = 30,rot=0,cex=0.25)
}

TractFun(fl$CHALMYDIA_, title = "Chlamydia rate")
title(main = "Chlamydia rate")
```

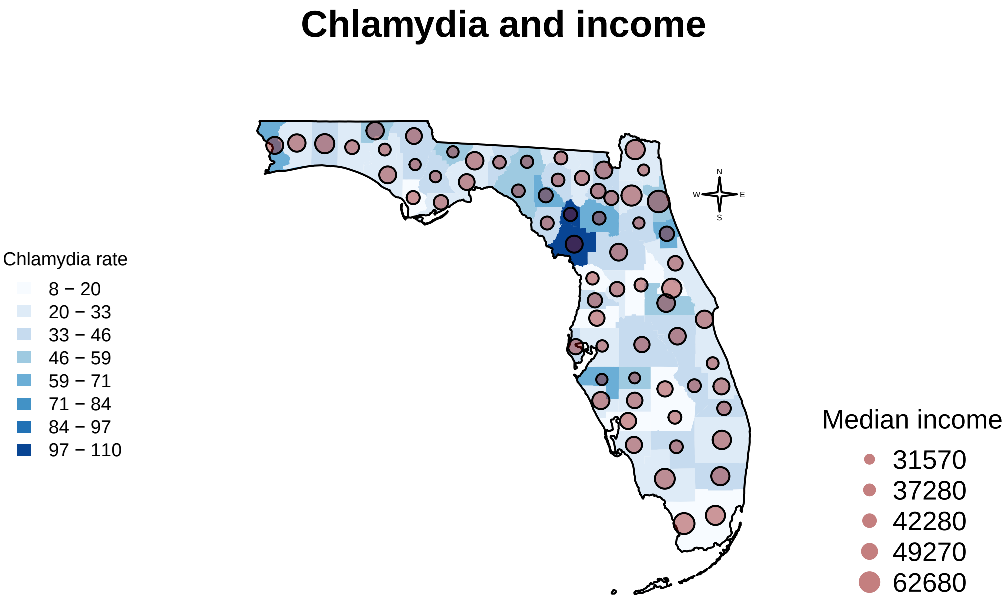


```
############
# 3. MAP INCOME (points) AND CHLAMYDIA (choropleth) TOGETHER
############
```

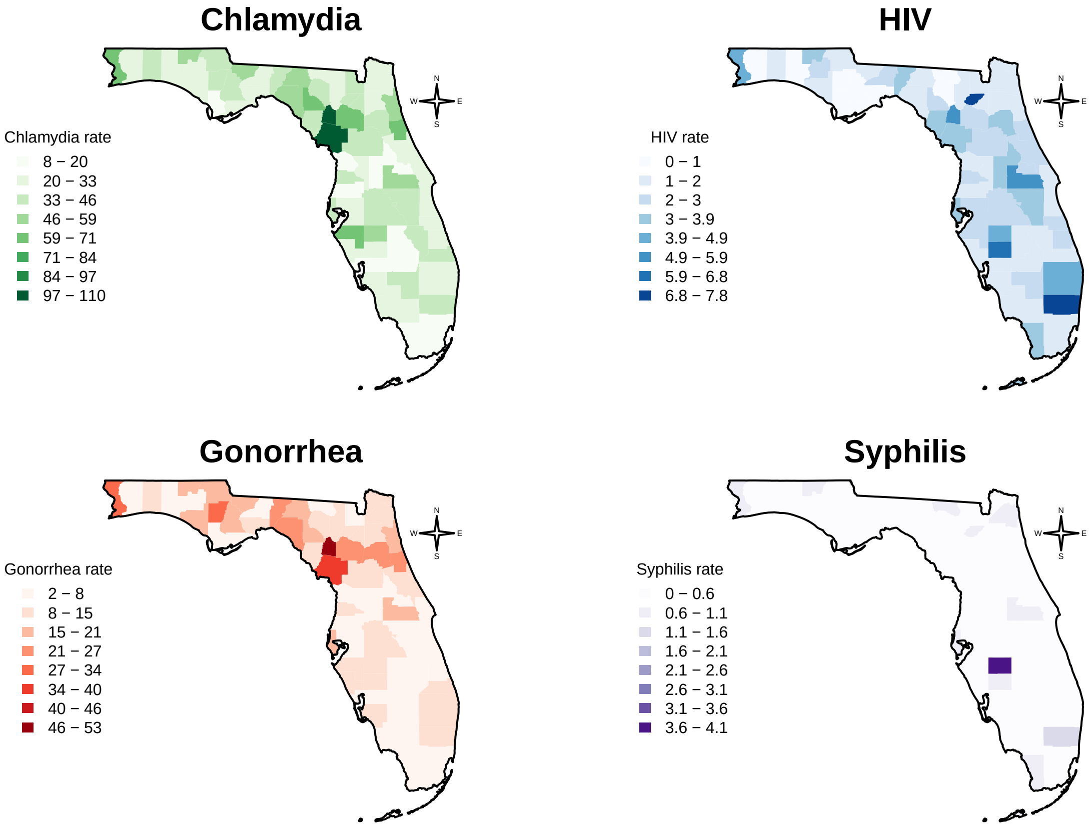
```r
# Before starting, let's just explore the linear relationship
plot(fl$INCOME, fl$CHALMYDIA_,
     xlab = "Income",
     ylab = "Chlamydia rate",
     pch = 16,
     col = adjustcolor("darkgreen", alpha.f = 0.6),
     main = "Income and Chlamydia in Florida counties")
abline(lm(CHALMYDIA_ ~ INCOME, data = fl),
       col = adjustcolor("darkred", alpha.f = 0.4))
```



```r
# First, write function to scale income to desired max
JoeScale <- function(my_vector, max_cex = 1.5){
    my_scale <- max(my_vector) / max_cex
    my_vector / my_scale
}

TractFun(fl$CHALMYDIA_,
         title = "Chlamydia rate")
points(coordinates(fl),
       cex = JoeScale(fl$INCOME),
       pch = 16,
       col = adjustcolor("darkred", alpha.f = 0.4))
points(coordinates(fl),
       cex = JoeScale(fl$INCOME))
legend("bottomright",
       pt.cex = JoeScale(quantile(fl$INCOME)),
       legend = round(quantile(fl$INCOME), digits = -1),
       pch = 16,
       col =  adjustcolor("darkred", alpha.f = 0.5),
       title = "Median income",
       bty= "n",
       cex = 0.8)
title(main = "Chlamydia and income")
```



```
############
# 4. MAP CHLAMYDIA, HIV, GONORRHEA AND SYPHILLIS
############
par(mfrow = c(2,2))
par(mar= c(1,1,1,1))
TractFun(fl$CHALMYDIA_,
         title = "Chlamydia rate",
         color = "Greens")
title(main = "Chlamydia")
TractFun(fl$HIV_RATES_,
         title = "HIV rate",
         color = "Blues",
         dataPrecision = 1)
title(main = "HIV")

TractFun(fl$GONORRHEA_,
         title = "Gonorrhea rate",
         color = "Reds")
title(main = "Gonorrhea")

TractFun(fl$SYPHILIS_R,
         title = "Syphilis rate",
         color = "Purples",
         dataPrecision = 1)
title(main = "Syphilis")
```



```
par(mfrow = c(1,1))
```