hw2_zheng

Charles Zheng

07/24/2015

Problem 1

```
edat0 <- read.csv("PrezElection2012.csv", sep = ",")
sapply(edat0, class)
```

```
##      State       FIPS       Name TotalVotes      Obama     Romney
##   "factor"  "integer"   "factor"  "integer"  "integer"  "integer"
```

```
filt <- !(edat0$State %in% c("AK", "HI", "DC") | edat0$FIPS == 0)
edat <- edat0[filt, ]
rownames(edat) <- paste0("FIPS", edat$FIPS)

dim(edat) # 3108 6
```

```
## [1] 3108    6
```

```
length(unique(edat$State)) # 48
```

```
## [1] 48
```

```
setdiff(unique(edat0$State), unique(edat$State)) # AK DC HI
```

```
## [1] "HI" "DC" "AK"
```

```
names(edat)
```

```
## [1] "State"      "FIPS"       "Name"       "TotalVotes" "Obama"
## [6] "Romney"
```

Problem 2

```
library(maptools)
```

```
## Loading required package: sp
## Checking rgeos availability: TRUE
```

```
library(RColorBrewer)
library(classInt)

shapes <- readShapeSpatial("UScounties/UScounties.shp")
class(shapes)
```

```
## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"
```

```
class(shapes)
```

```
## [1] "SpatialPolygonsDataFrame"
## attr(,"package")
## [1] "sp"
```

```
sapply(shapes@data, class)
```

```
##       NAME STATE_NAME STATE_FIPS  CNTY_FIPS       FIPS
##   "factor"   "factor"   "factor"   "factor"   "factor"
```

```
sfips <- as.numeric(as.character(shapes@data$FIPS))
rownames(shapes@data) <- paste0("FIPS", sfips)
length(shapes@polygons)
```

```
## [1] 3141
```

```
shapes@polygons[[1]]
```

```
## An object of class "Polygons"
## Slot "Polygons":
## [[1]]
## An object of class "Polygon"
## Slot "labpt":
## [1] -94.90359  48.77171
##
## Slot "area":
## [1] 0.5654499
##
## Slot "hole":
## [1] FALSE
##
## Slot "ringDir":
## [1] 1
##
## Slot "coords":
##            [,1]     [,2]
##  [1,] -95.34283 48.54668
##  [2,] -95.34105 48.71517
##  [3,] -95.09436 48.71736
```

```
##  [4,] -95.09491 48.91176
##  [5,] -95.13382 48.89448
##  [6,] -95.21958 48.87945
##  [7,] -95.29026 48.90295
##  [8,] -95.31417 48.93207
##  [9,] -95.30376 48.94594
## [10,] -95.32092 48.96098
## [11,] -95.32324 48.97896
## [12,] -95.31012 48.99340
## [13,] -95.27666 48.99999
## [14,] -95.15775 49.00000
## [15,] -95.15187 49.37173
## [16,] -94.83204 49.33081
## [17,] -94.68125 48.87716
## [18,] -94.69443 48.77762
## [19,] -94.57031 48.71368
## [20,] -94.43063 48.71079
## [21,] -94.43169 48.36821
## [22,] -95.21179 48.36900
## [23,] -95.21984 48.54436
## [24,] -95.34283 48.54668
##
##
##
## Slot "plotOrder":
## [1] 1
##
## Slot "labpt":
## [1] -94.90359  48.77171
##
## Slot "ID":
## [1] "0"
##
## Slot "area":
## [1] 0.5654499
```

Joining edat and shapes

```
dim(edat)
```

```
## [1] 3108    6
```

```
length(unique(edat$FIPS)) #3108
```

```
## [1] 3108
```

```
length(unique(shapes@data$FIPS)) #3141
```

```
## [1] 3141
```

```
length(unique(intersect(edat$FIPS, sfips))) #3108
```

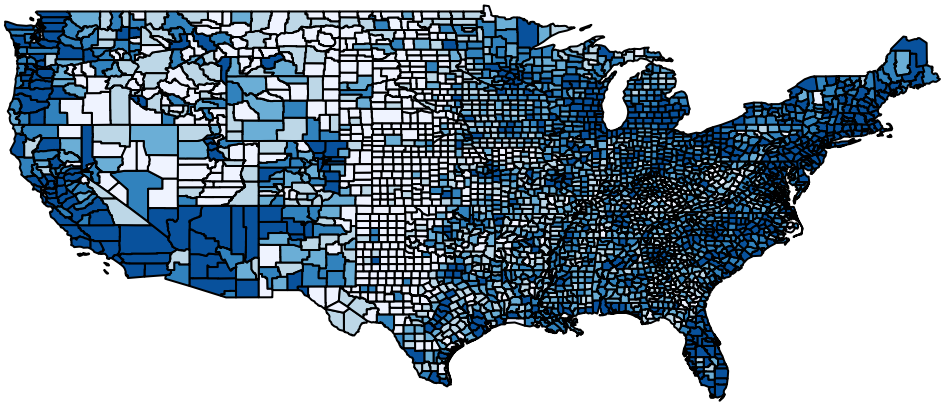
```
## [1] 3108
```

The shapes data contains all the counties we need. Now we make the spatial polygons data frame.

```
edat2 <- edat[rownames(edat) %in% rownames(shapes@data), ]
shapes2 <- shapes[rownames(shapes@data) %in% rownames(edat), ]
rownames(shapes2@data) <- rownames(shapes@data)[rownames(shapes@data) %in% rownames(edat)]
for (i in 1:length(shapes2)) {
  shapes2@polygons[[i]]@ID <- rownames(shapes2@data)[i]
}
edat3 <- SpatialPolygonsDataFrame(shapes2, edat2)
names(edat3)
```

```
## [1] "State"      "FIPS"       "Name"       "TotalVotes" "Obama"
## [6] "Romney"
```

```
pal <- brewer.pal(5, "Blues")
q5 <- classIntervals(edat3@data$Obama, n = 5, style = "quantile")
q5colors <- findColours(q5, pal)
plot(edat3, col = q5colors)
```



Problem 3

```
library(spdep)
```

```
## Loading required package: Matrix
```

```
adj <- read.table("county_adjacency.txt", sep = "", header = FALSE, fill = TRUE)
filler_inds <- which(!is.na(adj[, 4]))
filled_inds <- cumsum(!is.na(adj[, 4]))
adj[, 3:4] <- adj[filler_inds[filled_inds], 3:4]
## Found an misentered row
adj[9629, 1:2] <- adj[9629, 2:1]
adj[, 2] <- as.numeric(as.character(adj[, 2]))
adj[, 4] <- as.numeric(as.character(adj[, 4]))
```

Forming the nb object

```
temp0 <- c(as.character(adj[, 2]), as.character(adj[, 4]))
afips <- sort(unique(as.numeric(temp0)))
## Define the mapping from integers to FIPS
efips <- sort(unique(edat$FIPS))
fips2int <- numeric(max(efips))
fips2int[efips] <- 1:length(efips)
enames <- paste0("FIPS", efips)
names(fips2int)[efips] <- enames
length(afips) # 3234
```

```
## [1] 3234
```

```
length(unique(edat$FIPS)) # 3108
```

```
## [1] 3108
```

```
length(intersect(unique(edat$FIPS), afips)) # 3108
```

```
## [1] 3108
```

```
## Filter out guys in adj
filt <- (adj[, 2] %in% efips) & (adj[, 4] %in% efips) & (adj[, 2] != adj[, 4])
adjf <- adj[filt, ]
dim(adjf) # 21139 4
```

```
## [1] 21139     4
```

```
## Symmetrize neighbor list
adjfs <- adjf; names(adjfs) <- names(adjf)[c(3, 4, 1, 2)]
adj2 <- rbind(adjf, adjfs)
dim(adj2)
```

```
## [1] 42278     4
```

```
length(unique(adj2[, 2])) # 3107
```

```
## [1] 3107
```

```
sort(setdiff(efips, unique(adj2[, 2]))) # 51510
```

```
## [1] 51510
```

```
fips2int[51510] # 2881
```

```
## FIPS51510
##      2881
```

```
nb0 <- tapply(fips2int[adj2[, 4]], enames[fips2int[adj2[, 2]]], c)
length(nb0) # 3107
```

```
## [1] 3107
```

```
nb0 <- sapply(nb0, unique)
## These counties only had neighbors outside the census
empties <- fips2int[setdiff(enames, names(nb0))]
length(empties) # 1
```

```
## [1] 1
```

```
nb1 <- as.list(numeric(3108))
nb1[fips2int[names(nb0)]] <- nb0
nb1[[2881]] <- 2881
nb1 <- sapply(nb1, sort)
nb1 <- sapply(nb1, as.integer)
nb2 <- structure(nb1, class = "nb", region.id = enames,
                 GeoDa = list(shpfile = "unknown", ind = "unknown"),
                 gal = TRUE, call = TRUE, sym = TRUE)
```

Converting NB to listw. We have to manually handle row 2881 since it has no neighbors.

```
lw <- nb2listw(nb2)
lw$weights[[2881]] <- 0
```

Problem 4

```
res1 <- spautolm(Obama ~ 1, data = edat, listw=lw)
summary(res1)
```

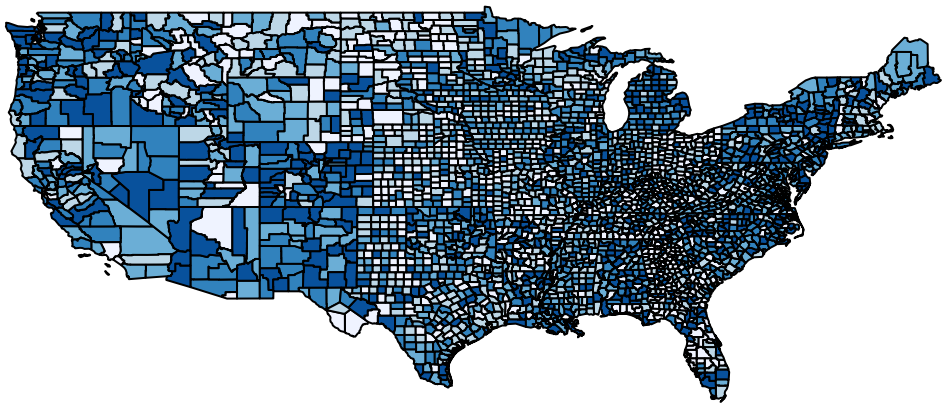
```
##
## Call: spautolm(formula = Obama ~ 1, data = edat, listw = lw)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
##  -20974.7  -18309.5  -15896.2   -8896.9 1652407.1
##
## Coefficients:
##             Estimate Std. Error z value  Pr(>|z|)
## (Intercept)  19858.7     1188.1  16.714 < 2.2e-16
##
## Lambda: 0.005985 LR test value: 0.030869 p-value: 0.86053
## Numerical Hessian standard error of lambda: 0.0095023
##
## Log likelihood: -38893.32
## ML residual variance (sigma squared): 4.335e+09, (sigma: 65841)
## Number of observations: 3108
## Number of parameters estimated: 3
## AIC: 77793
```

Code for plotting

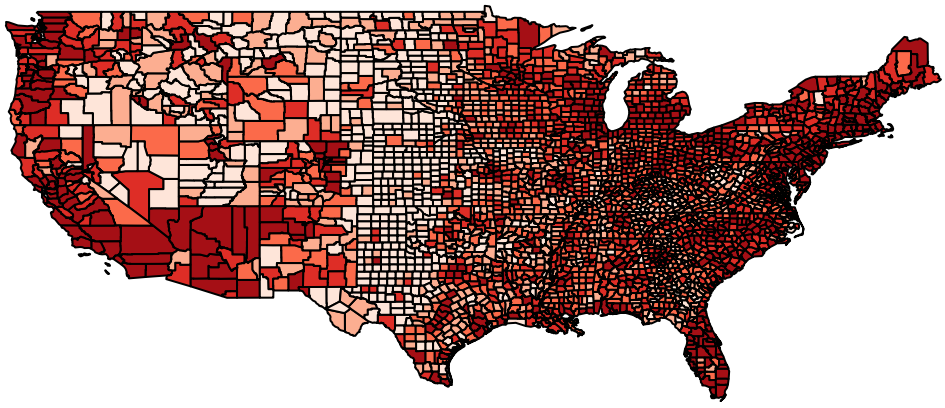
```
edat2 <- edat
edat2["fitted1"] <- res1$fit$fitted.values
edat2["resid1"] <- res1$fit$residuals

plot_edat <- function(edat, variable, colr = "Blues") {
  edat3 <- SpatialPolygonsDataFrame(shapes2, edat)
  pal <- brewer.pal(5, colr)
  q5 <- classIntervals(edat3@data[, variable], n = 5, style = "quantile")
  q5colors <- findColours(q5, pal)
  plot(edat3, col = q5colors)
}

plot_edat(edat2, "fitted1")
```



```
plot_edat(edat2, "resid1", "Reds")
```



Problem 5

```
census_dat <- read.table("census/DataSet.txt", header = TRUE, sep = ",")
census_dict <- read.fwf("census/DataDict.txt", header = FALSE,
                        widths=c(10, 87, 5, 7, 12, 8, 9, 8))
```

```
census_dat$fips <- as.numeric(as.character(census_dat$fips))

## Choose some variables that make sense
census_dat2 <- with(census_dat,
                    data.frame(FIPS = fips, pop = PST045214,
                               age18 = AGE295213, age65 = AGE775213,
                               sex = SEX255213, white = RHI125213 ,
                               black = RHI225213, hisp = RHI725213,
                               college = EDU685213, homes = HSG445213,
                               density =  POP060210))

length(unique(census_dat2$FIPS)) # 3195
```

```
## [1] 3195
```

```
length(unique(intersect(census_dat2$FIPS, efips))) # 3108
```

```
## [1] 3108
```

```
edat2 <- merge(edat, census_dat2, by = "FIPS")
rownames(edat2) <- paste0("FIPS", edat2$FIPS)

## Divide Obaama by Number of votes

edat2["Obama_Rate"] <- edat2$Obama/edat2$TotalVotes
```

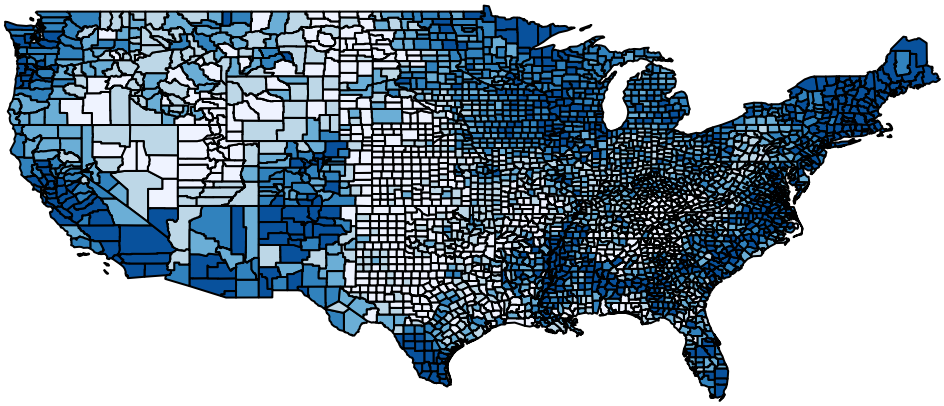
Problem 6

```
res2 <- spautolm(Obama_Rate ~ log(pop + 1) + age18 + age65 + sex + black + hisp + density,
                 data = edat2, listw=lw)
summary(res2)
```

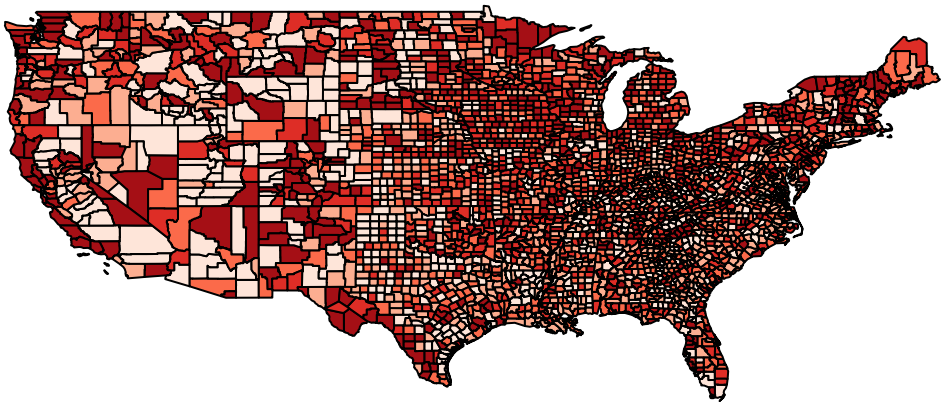
```
##
## Call: spautolm(formula = Obama_Rate ~ log(pop + 1) + age18 + age65 +
##     sex + black + hisp + density, data = edat2, listw = lw)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.3620830 -0.0415024 -0.0053612  0.0359281  0.6905973
##
## Coefficients:
##                 Estimate  Std. Error  z value  Pr(>|z|)
## (Intercept)   6.0292e-02  3.3701e-02   1.7890  0.073612
## log(pop + 1)  2.0721e-03  1.4650e-03   1.4144  0.157252
## age18        -8.0906e-03  6.3196e-04 -12.8023 < 2.2e-16
## age65        -6.9375e-03  5.2397e-04 -13.2403 < 2.2e-16
## sex           9.7121e-03  7.1070e-04  13.6656 < 2.2e-16
## black         7.7683e-03  1.7404e-04  44.6362 < 2.2e-16
## hisp          3.0561e-03  2.1333e-04  14.3252 < 2.2e-16
```

```
## density       2.3913e-06  8.5042e-07   2.8119  0.004925
##
## Lambda: 0.88447 LR test value: 2663.9 p-value: < 2.22e-16
## Numerical Hessian standard error of lambda: 0.0095435
##
## Log likelihood: 3565.033
## ML residual variance (sigma squared): 0.0051998, (sigma: 0.072109)
## Number of observations: 3108
## Number of parameters estimated: 10
## AIC: -7110.1
```

```
edat2["fitted2"] <- res2$fit$fitted.values
edat2["resid2"] <- res2$fit$residuals
plot_edat(edat2, "fitted2")
```



```
plot_edat(edat2, "resid2", "Reds")
```



Compared with the residuals in problem 4, the residuals of the new model are much less spatially correlated.