Rochester Repair

Ming DeMers

Packages

```
library( geojsonio )   # read shapefiles
```

```
Warning: package 'geojsonio' was built under R version 4.2.3
```

```
Registered S3 method overwritten by 'geojsonsf':
  method        from
  print.geojson geojson
```

```
Attaching package: 'geojsonio'
```

```
The following object is masked from 'package:base':

    pretty
```

```
library( sp )          # work with shapefiles
```

```
Warning: package 'sp' was built under R version 4.2.3
```

```
library( sf )          # work with shapefiles - simple features format
```

```
Warning: package 'sf' was built under R version 4.2.3
```

```
Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```r
library( mclust )      # cluster analysis
```

```
Warning: package 'mclust' was built under R version 4.2.3
```

```
Package 'mclust' version 6.1
Type 'citation("mclust")' for citing this R package in publications.
```

```r
library( tmap )        # theme maps
```

```
Warning: package 'tmap' was built under R version 4.2.3
```

```
Breaking News: tmap 3.x is retiring. Please test v4, e.g. with
remotes::install_github('r-tmap/tmap')
```

```r
library( ggplot2 )     # graphing
```

```
Warning: package 'ggplot2' was built under R version 4.2.3
```

```r
library( ggthemes )    # nice formats for ggplots
library( dplyr )       # data wrangling
```

```
Warning: package 'dplyr' was built under R version 4.2.3
```

```
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
```

```
  library( pander )      # formatting RMD tables
```

```
Warning: package 'pander' was built under R version 4.2.3
```

```
  library( tidycensus )
```

```
Warning: package 'tidycensus' was built under R version 4.2.3
```

```
  library( cartogram )  # spatial maps w/ tract size bias reduction
```

```
Warning: package 'cartogram' was built under R version 4.2.3
```

```
  census_api_key('f8acece5d0048351481060b7ea04587e089d4c13', install = F)
```

```
To install your API key for use in future sessions, run this function with `install = TRUE`.
```

```
  set.seed(2010)
```

Get MSA

```
  crosswalk <- read.csv( "https://raw.githubusercontent.com/DS4PS/cpp-529-master/master/data

  grep( "^ROCHESTER, NY", crosswalk$msaname, value=TRUE )
```

```
[1] "ROCHESTER, NY" "ROCHESTER, NY" "ROCHESTER, NY" "ROCHESTER, NY"
[5] "ROCHESTER, NY" "ROCHESTER, NY"
```

```
  these_msp <- crosswalk$msaname == "ROCHESTER, NY"
  these_fips <- crosswalk$fipscounty[these_msp] |>
    na.omit()
```

Get Shapefile

```
state_fips <- substr(these_fips, 1, 2)
county_fips <- substr(these_fips, 3, 5)

msp_pop1 <- get_acs( geography = "tract",
                     variables = "B01003_001",
                     state = "New York", county = county_fips[state_fips=="36"],
                     geometry = TRUE ) |>
         select( GEOID, estimate ) |>
         rename( POP=estimate )
```

```
Getting data from the 2018-2022 5-year ACS

Downloading feature geometry from the Census website.  To cache shapefiles for use in future

  |
  |                                                                      |   0%
  |
  |=                                                                     |   1%
  |
  |=                                                                     |   2%
  |
  |==                                                                    |   3%
  |
  |====                                                                  |   6%
  |
  |=====                                                                 |   8%
  |
  |======                                                                |   9%
  |
  |=======                                                               |  10%
  |
  |========                                                              |  12%
  |
  |=========                                                             |  12%
  |
  |=========                                                             |  13%
  |
  |==========                                                            |  14%
  |
  |===========                                                           |  15%
```

```
|
|============                                                     |  17%
|
|=============                                                    |  18%
|
|=============                                                    |  19%
|
|===============                                                  |  21%
|
|================                                                 |  21%
|
|================                                                 |  22%
|
|=================                                                |  23%
|
|==================                                               |  24%
|
|==================                                               |  25%
|
|===================                                              |  26%
|
|====================                                             |  27%
|
|====================                                             |  28%
|
|=====================                                            |  28%
|
|=====================                                            |  29%
|
|======================                                           |  30%
|
|=======================                                          |  31%
|
|=======================                                          |  32%
|
|========================                                         |  33%
|
|=========================                                        |  35%
|
|=========================                                        |  36%
|
|==========================                                       |  37%
|
```

```
|==========================                                    |  39%
|
|===========================                                   |  40%
|
|============================                                  |  41%
|
|=============================                                 |  42%
|
|==============================                                |  44%
|
|===============================                               |  45%
|
|================================                              |  46%
|
|================================                              |  48%
|
|=================================                             |  49%
|
|==================================                            |  50%
|
|===================================                           |  51%
|
|====================================                          |  52%
|
|=====================================                         |  53%
|
|======================================                        |  54%
|
|=======================================                       |  55%
|
|=========================================                     |  57%
|
|==========================================                    |  58%
|
|===========================================                   |  59%
|
|============================================                  |  60%
|
|=============================================                 |  62%
|
|==============================================                |  63%
|
|===============================================               |  64%
```

```
|
|=============================================                         |  66%
|
|==============================================                        |  67%
|
|===============================================                       |  68%
|
|================================================                      |  69%
|
|================================================                      |  69%
|
|=================================================                     |  71%
|
|==================================================                    |  72%
|
|===================================================                   |  73%
|
|====================================================                  |  75%
|
|=====================================================                 |  76%
|
|======================================================                |  77%
|
|=======================================================               |  78%
|
|=========================================================             |  80%
|
|==========================================================            |  81%
|
|============================================================          |  82%
|
|==============================================================        |  84%
|
|===============================================================       |  85%
|
|================================================================      |  86%
|
|=================================================================     |  87%
|
|===================================================================   |  89%
|
|====================================================================  |  90%
|
```

```
  |=============================================================      |  91%
  |
  |==============================================================     |  93%
  |
  |================================================================   |  94%
  |
  |================================================================   |  95%
  |
  |================================================================== |  96%
  |
  |================================================================== |  98%
  |
  |==================================================================== |  99%
  |
  |======================================================================| 100%
```

Merge in Census Data

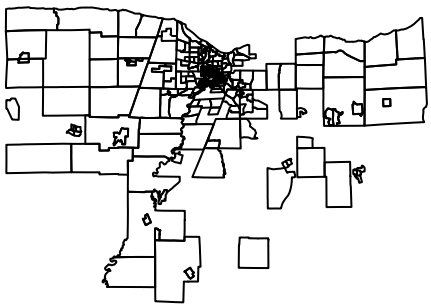
```
URL <- "https://github.com/DS4PS/cpp-529-master/raw/master/data/ltdb_std_2010_sample.rds"
census.dat <- readRDS(gzcon(url( URL )))

# can merge an sf object and data.frame
msp <- merge( msp_pop1, census.dat, by.x="GEOID", by.y="tractid" )

# make sure there are no empty polygons
msp <- msp[ ! st_is_empty( msp ) , ]
```

Transform into a Dorling Cartogram

```
msp.sp <- as_Spatial( msp )

class( msp.sp)
```

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

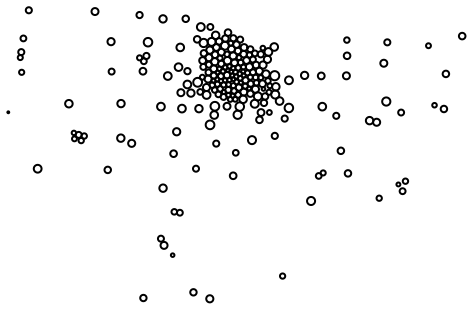
```
plot(msp.sp)
```



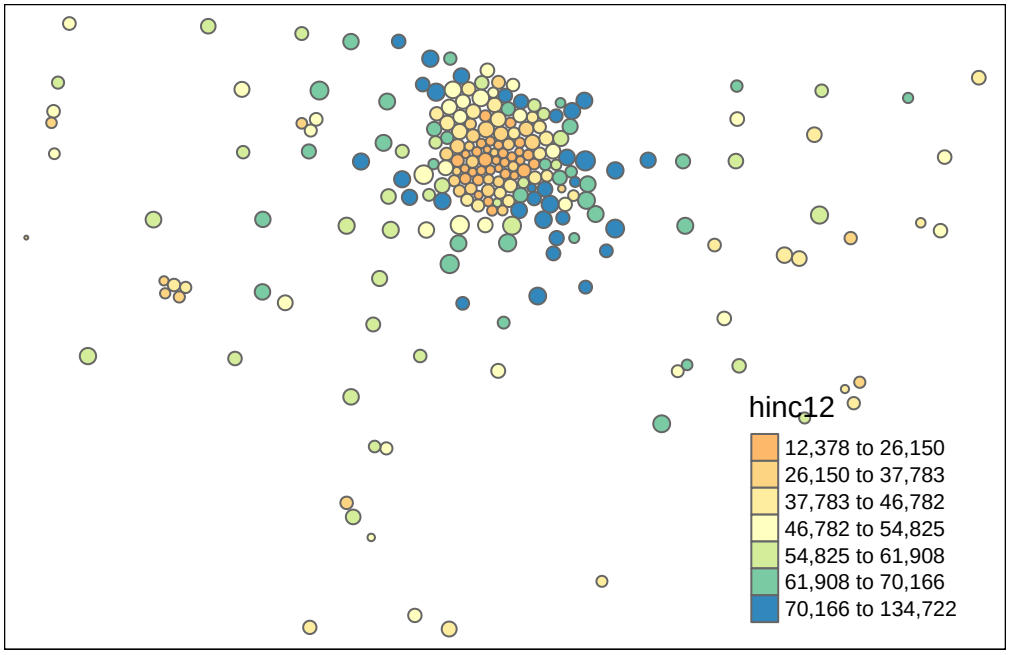
```
# project map and remove empty tracts
msp.sp <- spTransform( msp.sp, CRS("+init=epsg:3395"))
```

```
Warning in CPL_crs_from_input(x): GDAL Message 1: +init=epsg:XXXX syntax is
deprecated. It might return a CRS with a non-EPSG compliant axis order.
```

```
msp.sp <- msp.sp[ msp.sp$POP != 0 & (! is.na( msp.sp$POP )) , ]

# convert census tract polygons to dorling cartogram
# no idea why k=0.03 works, but it does - default is k=5
msp.sp$pop.w <- msp.sp$POP / 9000 # max(msp.sp$POP)   # standardizes it to max of 1.5
msp_dorling <- cartogram_dorling( x=msp.sp, weight="pop.w", k=0.05 )

plot( msp_dorling )
```



```
tm_shape( msp_dorling ) +
  tm_polygons( size="POP", col="hinc12", n=7, style="quantile", palette="Spectral" )
```



Clustering

```
keep.these <- c("pnhwht12", "pnhblk12", "phisp12", "pntv12", "pfb12", "polang12",
"phs12", "pcol12", "punemp12", "pflabf12", "pprof12", "pmanuf12",
"pvet12", "psemp12", "hinc12", "incpc12", "ppov12", "pown12",
"pvac12", "pmulti12", "mrent12", "mhmval12", "p30old12", "p10yrs12",
"p18und12", "p60up12", "p75up12", "pmar12", "pwds12", "pfhh12")

d1 <- msp_dorling@data
d2 <- select( d1, keep.these )
```

```
Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
i Please use `all_of()` or `any_of()` instead.
  # Was:
  data %>% select(keep.these)

  # Now:
  data %>% select(all_of(keep.these))

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

```
d3 <- apply( d2, 2, scale )
head( d3[,1:6] ) %>% pander()
```

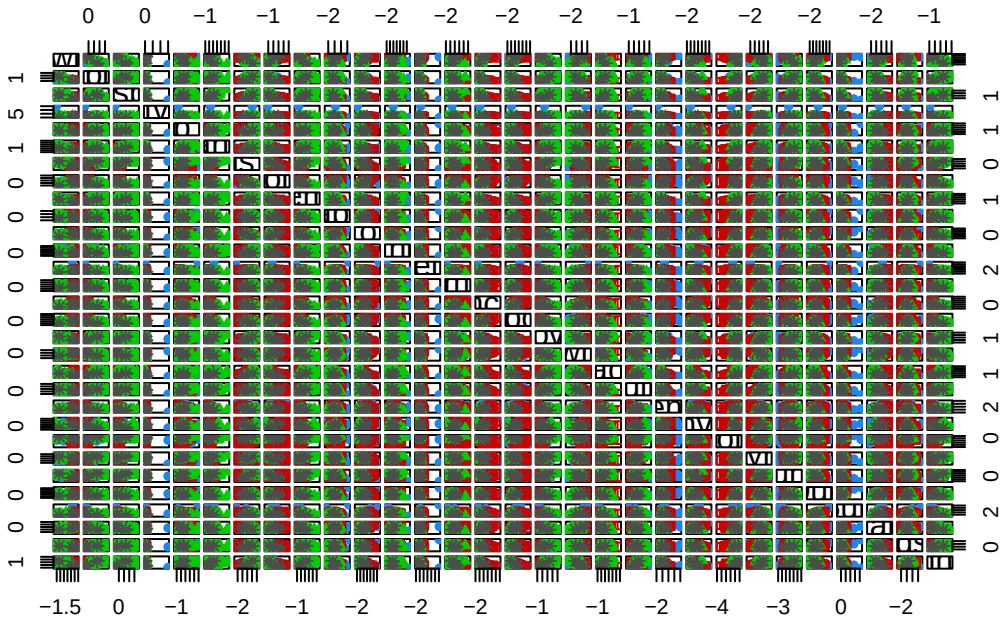
pnhwht12	pnhblk12	phisp12	pntv12	pfb12	polang12
-2.043	-0.7126	-0.7955	14.96	-0.671	0.3008
0.8138	-0.6941	-0.5024	-0.02926	-1.183	-0.9637
0.6548	-0.6526	-0.1411	-0.03836	-0.2845	-0.5429
0.4498	-0.5516	-0.5636	-0.1056	-1.178	-0.8524
0.3576	-0.2923	-0.4934	-0.04017	-1.217	-0.7707
0.4384	-0.5406	-0.09197	-0.1056	-0.2197	-1.039

```
fit <- Mclust( d3 )
msp_dorling$cluster <- as.factor( fit$classification )
summary( fit )
```

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
```

```
----------------------------------------------------

Mclust VVE (ellipsoidal, equal orientation) model with 3 components:

 log-likelihood   n  df       BIC       ICL
      -3686.183 228 617 -10722.27 -10722.32

Clustering table:
  1   2   3
 30 145  53
```

```r
plot( fit, what = "classification" )
```



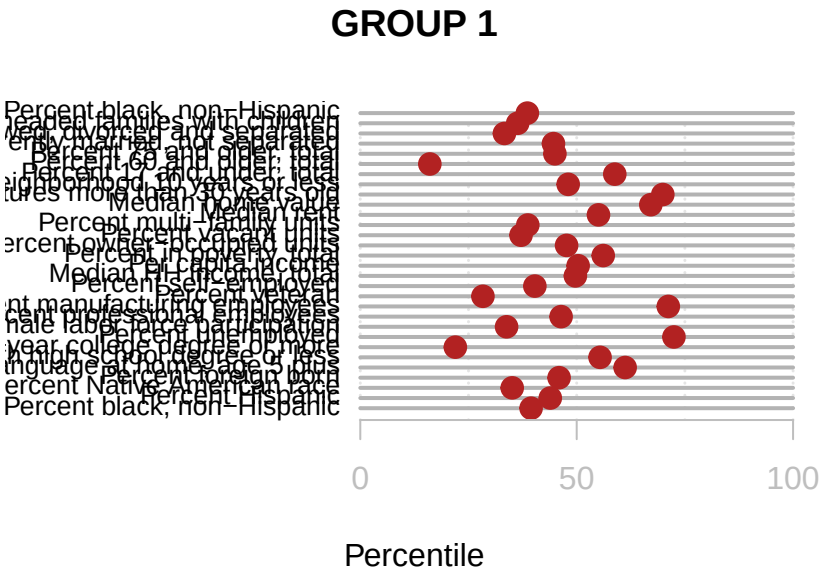
Identifying Neighborhood Clusters

```r
data.dictionary <-
structure(list(LABEL = c("pnhwht12", "pnhblk12", "phisp12",
"pntv12", "pfb12", "polang12", "phs12", "pcol12", "punemp12",
"pflabf12", "pprof12", "pmanuf12", "pvet12", "psemp12", "hinc12",
"incpc12", "ppov12", "pown12", "pvac12", "pmulti12", "mrent12",
```

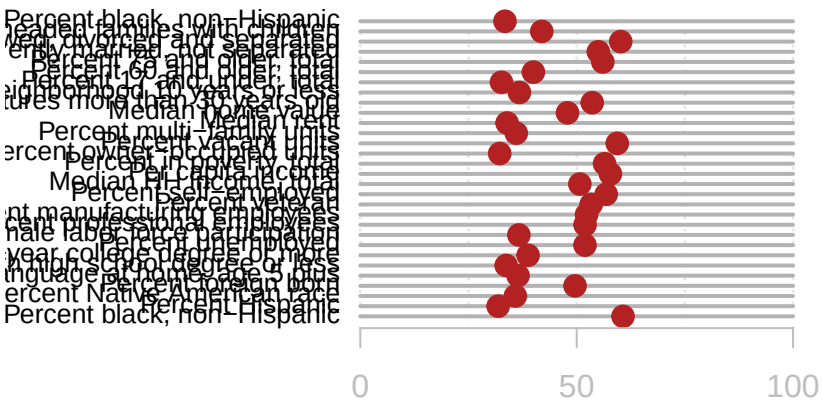
```
  "mhmval12", "p30old12", "p10yrs12", "p18und12", "p60up12", "p75up12",
  "pmar12", "pwds12", "pfhh12"), VARIABLE = c("Percent white, non-Hispanic",
  "Percent black, non-Hispanic", "Percent Hispanic", "Percent Native American race",
  "Percent foreign born", "Percent speaking other language at home, age 5 plus",
  "Percent with high school degree or less", "Percent with 4-year college degree or more",
  "Percent unemployed", "Percent female labor force participation",
  "Percent professional employees", "Percent manufacturing employees",
  "Percent veteran", "Percent self-employed", "Median HH income, total",
  "Per capita income", "Percent in poverty, total", "Percent owner-occupied units",
  "Percent vacant units", "Percent multi-family units", "Median rent",
  "Median home value", "Percent structures more than 30 years old",
  "Percent HH in neighborhood 10 years or less", "Percent 17 and under, total",
  "Percent 60 and older, total", "Percent 75 and older, total",
  "Percent currently married, not separated", "Percent widowed, divorced and separated",
  "Percent female-headed families with children")), class = "data.frame", row.names = c(NA,
  -30L))


  df.pct <- sapply( d2, ntile, 100 )
  d4 <- as.data.frame( df.pct )
  d4$cluster <- as.factor( paste0("GROUP-",fit$classification) )

  num.groups <- length( unique( fit$classification ) )

  stats <-
  d4 %>%
    group_by( cluster ) %>%
    summarise_each( funs(mean) )
```

```
Warning: `summarise_each_()` was deprecated in dplyr 0.7.0.
i Please use `across()` instead.
i The deprecated feature was likely used in the dplyr package.
  Please report the issue at <https://github.com/tidyverse/dplyr/issues>.

Warning: `funs()` was deprecated in dplyr 0.8.0.
i Please use a list of either functions or lambdas:

# Simple named list: list(mean = mean, median = median)

# Auto named with `tibble::lst()`: tibble::lst(mean, median)

# Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

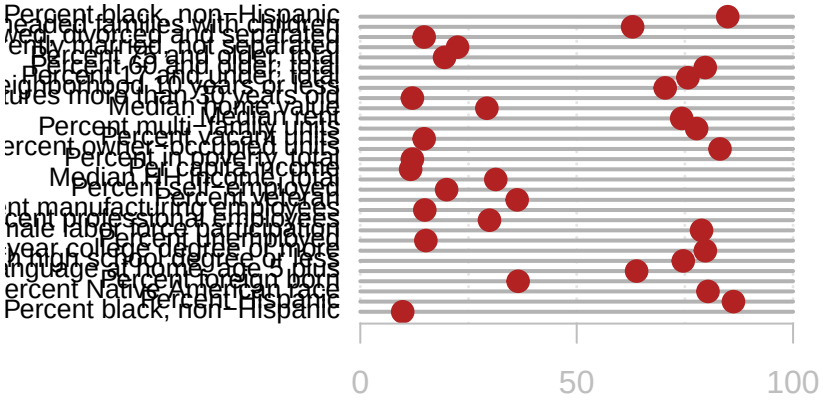
```r
t <- data.frame( t(stats), stringsAsFactors=F )
names(t) <- paste0( "GROUP.", 1:num.groups )
t <- t[-1,]



for( i in 1:num.groups )
{
  z <- t[,i]
  plot( rep(1,30), 1:30, bty="n", xlim=c(-75,100),
        type="n", xaxt="n", yaxt="n",
        xlab="Percentile", ylab="",
        main=paste("GROUP",i) )
  abline( v=seq(0,100,25), lty=3, lwd=1.5, col="gray90" )
  segments( y0=1:30, x0=0, x1=100, col="gray70", lwd=2 )
  text( -0.2, 1:30, data.dictionary$VARIABLE[-1], cex=0.85, pos=2 )
  points( z, 1:30, pch=19, col="firebrick", cex=1.5 )
  axis( side=1, at=c(0,50,100), col.axis="gray", col="gray" )
}
```



GROUP 2

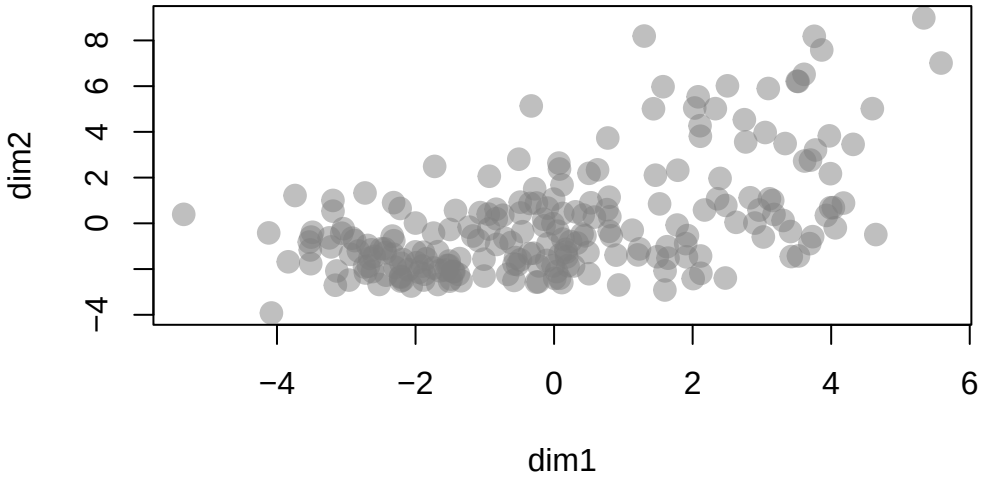


GROUP 3

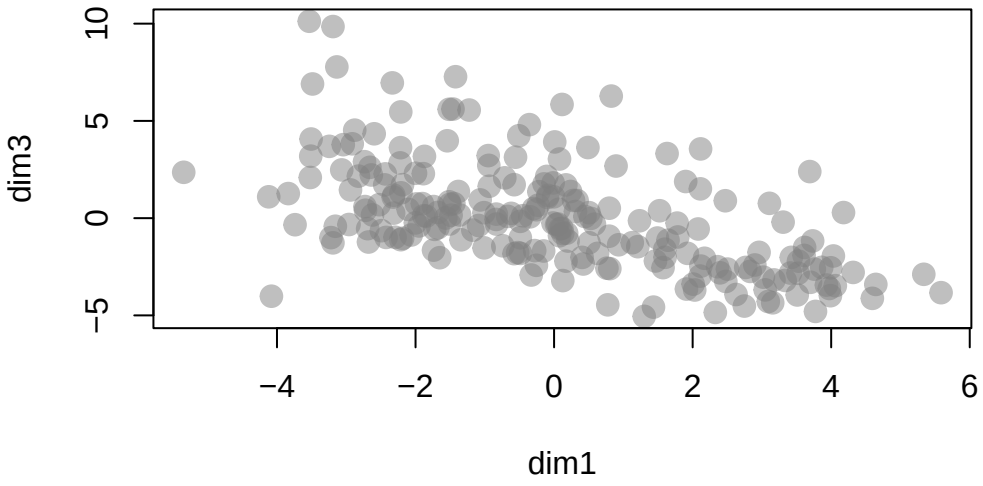


Variable Selection

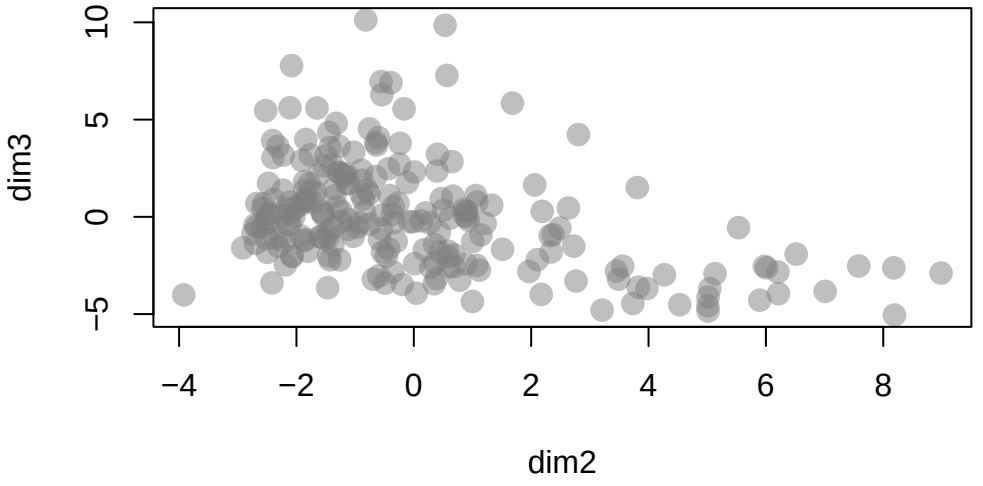
```
d3 <- as.data.frame(d3)
dim1 <- d3$pown12 + d3$pmulti12 + d3$p10yrs12 + d3$pwds12 + d3$pfhh12
dim2 <- d3$pnhwht12 + d3$pnhblk12 + d3$phisp12 + d3$pfb12 + d3$polang12
dim3 <- d3$pcol12 + d3$phs12 + d3$pprof12 + d3$hinc12 + d3$mhmval12


plot( dim1, dim2, pch=19, col=gray(0.5,0.5), cex=1.5 )
```




```
plot( dim1, dim3, pch=19, col=gray(0.5,0.5), cex=1.5 )
```




```
plot( dim2, dim3, pch=19, col=gray(0.5,0.5), cex=1.5 )
```



```r
# data set of three indices
d22 <- data.frame( dim1, dim2, dim3 )
fit2 <- Mclust( d22 )
summary( fit2 )
```

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust EVE (ellipsoidal, equal volume and orientation) model with 3 components:

 log-likelihood   n df       BIC       ICL
      -1407.188 228 21 -2928.393 -3000.713

Clustering table:
  1   2   3
106  67  55
```

```r
msp_dorling$cluster2 <- as.factor( fit2$classification )
```

```r
msp_dorling$cluster2 <- as.factor( fit2$classification )

# dataset of three census variables
d33 <- data.frame( d3$p18und12, d3$pflabf12, d3$hinc12 )
fit3 <- Mclust( d33 )
summary( fit3 )
```

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVE (ellipsoidal, equal orientation) model with 3 components:

 log-likelihood   n df       BIC       ICL
      -843.1147 228 23 -1811.104 -1861.522

Clustering table:
  1   2   3
 78 132  18
```

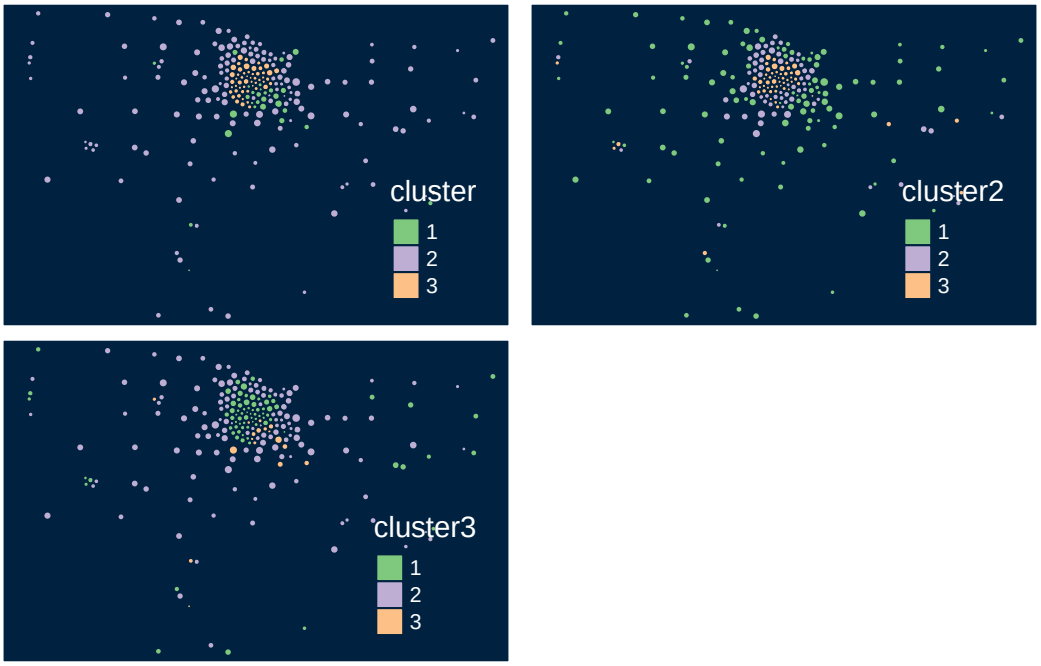
```r
msp_dorling$cluster3 <- as.factor( fit3$classification )


tmap_mode("plot")
```

```
tmap mode set to plotting
```

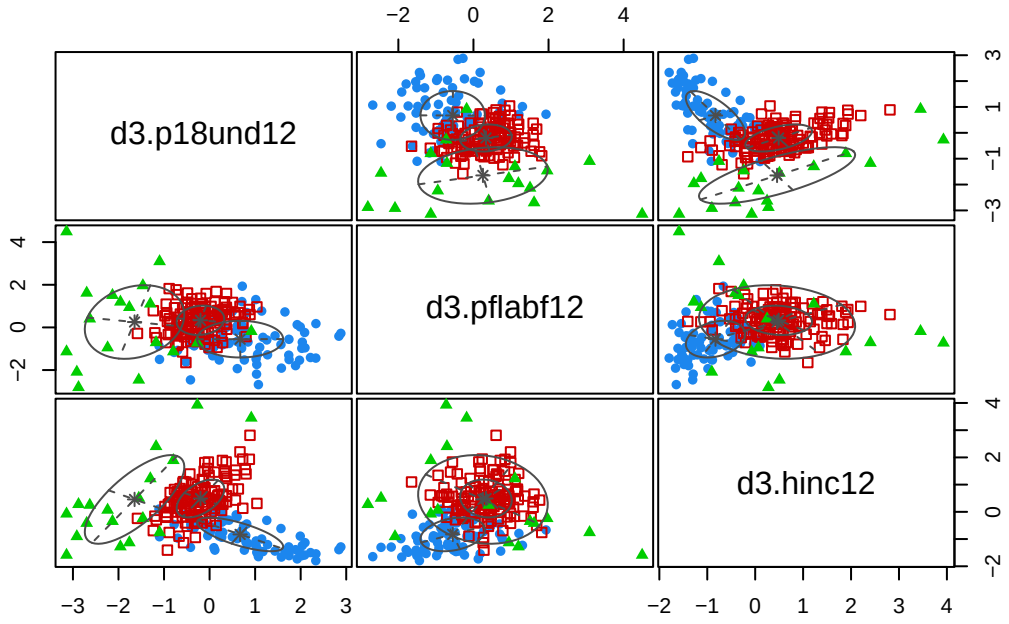
```r
tmap_style("cobalt")
```

```
tmap style set to "cobalt"

other available styles are: "white", "gray", "natural", "col_blind", "albatross", "beaver",
```

```r
tm1 <-
tm_shape( msp_dorling) +
  tm_polygons( col="cluster", palette="Accent"  )

tm2 <-
tm_shape( msp_dorling) +
  tm_polygons( col="cluster2", palette="Accent"  )

tm3 <-
tm_shape( msp_dorling) +
  tm_polygons( col="cluster3", palette="Accent"  )


tmap_arrange( tm1, tm2, tm3 )
```




```
plot( fit3, what = "classification" )
```



```
plot( fit2, what = "classification" )
```