Cluster Modeling

```
here() starts at /Users/clipo/PycharmProjects/sustainable_communities

Package 'mclust' version 5.4.10
Type 'citation("mclust")' for citing this R package in publications.

Loading required package: sp

Please note that rgdal will be retired by the end of 2023,
plan transition to sf/stars/terra functions using GDAL and PROJ
at your earliest convenience.

rgdal: version: 1.5-32, (SVN revision 1176)
Geospatial Data Abstraction Library extensions to R successfully loaded
Loaded GDAL runtime: GDAL 3.5.1, released 2022/06/30
Path to GDAL shared files: /usr/local/Cellar/gdal/3.5.1_2/share/gdal
 GDAL does not use iconv for recoding strings.
GDAL binary built with GEOS: TRUE
Loaded PROJ runtime: Rel. 9.0.1, June 15th, 2022, [PJ_VERSION: 901]
Path to PROJ shared files: /Users/clipo/Library/Application Support/proj:/usr/local/Cellar/p
PROJ CDN enabled: FALSE
Linking to sp version:1.5-0
To mute warnings of possible GDAL/OSR exportToProj4() degradation,
use options("rgdal_show_exportToProj4_warnings"="none") before loading sp or rgdal.

Loading required package: ggplot2

Loading required package: ggrepel


Attaching package: 'PCAtools'
```

```
The following objects are masked from 'package:stats':

    biplot, screeplot

Linking to GEOS 3.11.0, GDAL 3.5.1, PROJ 9.0.1; sf_use_s2() is TRUE

terra 1.6.7


Attaching package: 'terra'

The following object is masked from 'package:rgdal':

    project

rgeos version: 0.5-9, (SVN revision 684)
 GEOS runtime version: 3.8.1-CAPI-1.13.3
 Please note that rgeos will be retired by the end of 2023,
plan transition to sf functions using GEOS at your earliest convenience.
 Linking to sp version: 1.4-6
 Polygon checking: TRUE

To access larger datasets in this package, install the spDataLarge
package with: `install.packages('spDataLarge',
repos='https://nowosad.github.io/drat/', type='source')`


Attaching package: 'knitr'

The following object is masked from 'package:terra':

    spin
```

You can add options to executable code like this

The `echo: false` option disables the printing of code (only output is displayed).

```
dat_comps <- prcomp(dat, center = F, scale=F) #set center and scale to FALSE because done
#get summary
s <- summary(dat_comps)
```

```r
kable(s$importance)
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	
Standard deviation	1.952378	1.514911	1.264653	1.165375	1.075166	1.01657	0.9178323	0.895
Proportion of Variance	0.224220	0.135000	0.094080	0.079890	0.068000	0.06079	0.0495500	0.047
Cumulative Proportion	0.224220	0.359220	0.453300	0.533190	0.601190	0.66198	0.7115300	0.758

```r
#get eigenvalues
#### change this # of columns to match
ev <- data.frame(Component=paste("PC",1:17, sep=""),eigenvalue=dat_comps$sdev^2)
kable(ev)
```

Component	eigenvalue
PC1	3.8117804
PC2	2.2949543
PC3	1.5993464
PC4	1.3580994
PC5	1.1559810
PC6	1.0334144
PC7	0.8424162
PC8	0.8010683
PC9	0.7496125
PC10	0.7020398
PC11	0.5922121
PC12	0.5558430
PC13	0.4547213
PC14	0.4016146
PC15	0.3038364
PC16	0.2103073
PC17	0.1327525

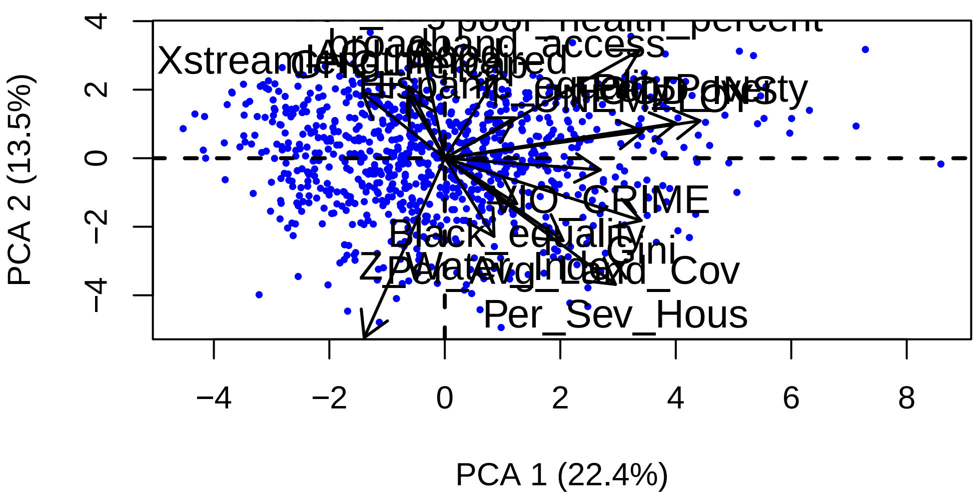
```r
#write to CSV
write.csv(dat_comps$x,file=here("results/community_loadings.csv"))
```

Extract components with eigenvalues > 1

```r
dat_pcs <- dat_comps$x[,1:5]
```

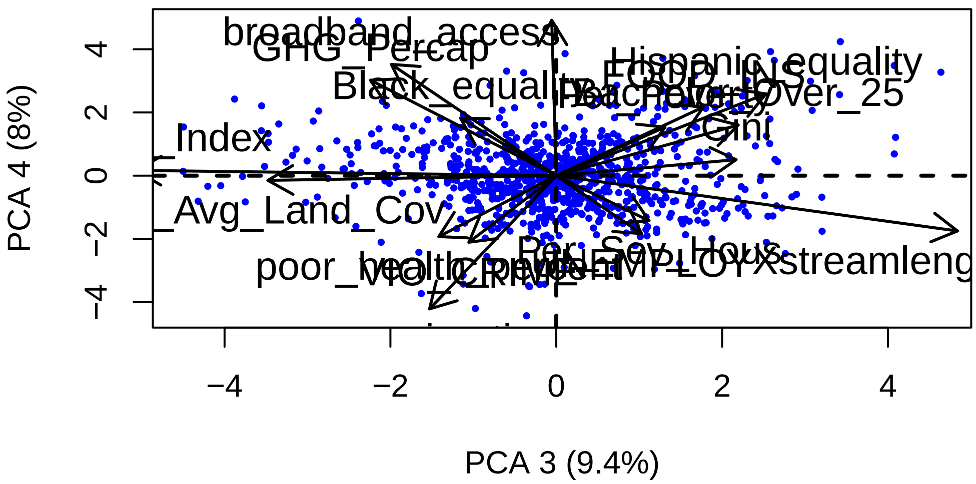
Plot loadings

```
#plot loadings of pc1 and 2
plot(dat_comps$x[,1],dat_comps$x[,2],
     xlab=paste("PCA 1 (", round(s$importance[2,1]*100, 1), "%)", sep = ""),
     ylab=paste("PCA 2 (", round(s$importance[2,2]*100, 1), "%)", sep = ""),
     pch=16, col="blue", cex=0.5)
abline(v=0, lwd=2, lty=2)
abline(h=0, lwd=2, lty=2)
#get loadings
l.x <- dat_comps$rotation[,1]*10
l.y <- dat_comps$rotation[,2]*10
arrows(x0=mean(dat_comps$x[,1]), x1=l.x, y0=mean(dat_comps$x[,2]), y1=l.y, col="black", le
# Label position
l.pos <- l.y # Create a vector of y axis coordinates
lo <- which(l.y < 0) # Get the variables on the bottom half of the plot
hi <- which(l.y > 0) # Get variables on the top half
# Replace values in the vector
l.pos <- replace(l.pos, lo, "1")
l.pos <- replace(l.pos, hi, "3")
text(l.x, l.y, labels=row.names(dat_comps$rotation), col="black", pos=l.pos, cex=1.25)
```




```
#plot loadings of pc 3 and 4
plot(dat_comps$x[,3],dat_comps$x[,4],
     xlab=paste("PCA 3 (", round(s$importance[2,3]*100, 1), "%)", sep = ""),
     ylab=paste("PCA 4 (", round(s$importance[2,4]*100, 1), "%)", sep = ""),
     pch=16, col="blue", cex=0.5)
abline(v=0, lwd=2, lty=2)
```

```
abline(h=0, lwd=2, lty=2)

#get loadings
l.x <- dat_comps$rotation[,3]*10
l.y <- dat_comps$rotation[,4]*10
arrows(x0=mean(dat_comps$x[,3]), x1=l.x, y0=mean(dat_comps$x[,4]), y1=l.y, col="black", le
# Label position
l.pos <- l.y # Create a vector of y axis coordinates
lo <- which(l.y < 0) # Get the variables on the bottom half of the plot
hi <- which(l.y > 0) # Get variables on the top half
# Replace values in the vector
l.pos <- replace(l.pos, lo, "1")
l.pos <- replace(l.pos, hi, "3")
text(l.x, l.y, labels=row.names(dat_comps$rotation), col="black", pos=l.pos, cex=1.25)
```



Summary of PC Loadings

```
print(dat_comps)
```

```
Standard deviations (1, .., p=17):
 [1] 1.9523781 1.5149107 1.2646527 1.1653752 1.0751656 1.0165699 0.9178323
 [8] 0.8950242 0.8658016 0.8378781 0.7695532 0.7455488 0.6743303 0.6337307
[15] 0.5512136 0.4585928 0.3643522

Rotation (n x k) = (17 x 17):
                                    PC1             PC2             PC3             PC4
```

```
AQI_Good              -0.06295202  0.20965371 -0.005432617  0.49163969
Bachelor_Over_25      -0.13968804 -0.52379010  0.219346822  0.16013665
Per_Poverty            0.44232701  0.10915175  0.131118078  0.15353597
Gini                   0.34007067 -0.18190264  0.216850424  0.05068366
non_migration         -0.04219990  0.34266599 -0.152782501 -0.42139852
Per_Sev_Hous           0.29560691 -0.36825603  0.111885245 -0.14263533
Xstreamlengthimpaired -0.14269057  0.19068065  0.483798649 -0.17501278
Per_Avg_Land_Cov       0.20498968 -0.24061925 -0.347750130 -0.01493466
poor_health_percent    0.33693731  0.31436589 -0.141628751 -0.19274740
Z_Water_Index          0.08518167 -0.22839053 -0.506690724  0.01676122
GHG_Percap            -0.06184721  0.17812359 -0.223924718  0.30201735
UNEMPLOY               0.34346073  0.07875371  0.102777616 -0.18331874
FOOD_INS               0.40034521  0.10233172  0.177464027  0.21728649
VIO_CRIME              0.26943276 -0.03400812 -0.105354267 -0.21055356
broadband_access       0.08955274  0.23922383 -0.198765892  0.35203655
Black_equality         0.12465909 -0.13294244 -0.115427628  0.18191510
Hispanic_equality      0.12113838  0.11810579  0.252715621  0.25866712
                              PC5         PC6          PC7          PC8
AQI_Good              -0.06860641  0.503779071 -0.198420163 -0.154641861
Bachelor_Over_25      -0.03151075 -0.127648345  0.073469915 -0.236716539
Per_Poverty           -0.06698781  0.051457984 -0.061029453 -0.014175963
Gini                  -0.17246943 -0.148603745 -0.003264975 -0.491638728
non_migration          0.18717529 -0.057892103  0.024476913 -0.609012060
Per_Sev_Hous           0.03296150  0.072066033 -0.062428713  0.027107536
Xstreamlengthimpaired  0.20766303 -0.106812888  0.058971727  0.165583527
Per_Avg_Land_Cov       0.24810692  0.290996372  0.011450266 -0.257866970
poor_health_percent    0.07455852  0.052483438 -0.042252204  0.012400111
Z_Water_Index          0.06874951 -0.009694891 -0.051691389  0.137713489
GHG_Percap            -0.10454123 -0.510126665 -0.590760260 -0.032583098
UNEMPLOY               0.06473546  0.222448615 -0.152837871  0.324599608
FOOD_INS              -0.16979422 -0.008944185  0.006946807  0.036049129
VIO_CRIME             -0.26135372 -0.429636802  0.056761926  0.091667961
broadband_access      -0.09586330 -0.155544011  0.747232124  0.009391392
Black_equality         0.63291532 -0.224341196  0.053137854  0.209747952
Hispanic_equality      0.53567966 -0.169696392 -0.046120693 -0.182393815
                               PC9         PC10        PC11         PC12
AQI_Good               0.328962683 -0.315136060 -0.09519628 -0.276212690
Bachelor_Over_25      -0.046501630  0.009614631 -0.06389346 -0.139586191
Per_Poverty            0.003559301 -0.117332854  0.17843890  0.181338803
Gini                   0.082509265 -0.182208805  0.18148151  0.052555023
non_migration         -0.020129550 -0.149508308  0.04581246 -0.100932086
Per_Sev_Hous          -0.150570861 -0.119624903  0.28568987  0.080276003
Xstreamlengthimpaired  0.169868052 -0.140043906  0.48041376 -0.470012473
```

```
Per_Avg_Land_Cov      -0.245207996  0.219363752 -0.03827679 -0.468911748
poor_health_percent    0.017417675 -0.098603366 -0.02312504  0.286477821
Z_Water_Index          0.555290226  0.105384363  0.49747290  0.033505963
GHG_Percap            -0.330432181  0.042513514  0.21633611 -0.168895123
UNEMPLOY              -0.221229018  0.218291436 -0.01491920 -0.319928058
FOOD_INS              -0.030385304  0.047723743 -0.09737632 -0.084906858
VIO_CRIME              0.417569418 -0.040791749 -0.44181808 -0.391674821
broadband_access      -0.203831635  0.067156709  0.22436585 -0.132177312
Black_equality        -0.088786385 -0.581925093 -0.20570819  0.006719023
Hispanic_equality      0.281506033  0.576071849 -0.11713298  0.137655132
                              PC13        PC14         PC15         PC16
AQI_Good               0.188966606  0.19680823  0.099814588  0.097563908
Bachelor_Over_25       0.056755409 -0.11736988  0.242266575  0.468221291
Per_Poverty           -0.042350713  0.04886193 -0.263448543 -0.327914300
Gini                   0.007167778 -0.24704326  0.352431296 -0.343866706
non_migration          0.303768042 -0.12501549 -0.270583051  0.200840559
Per_Sev_Hous           0.280391258  0.61188848 -0.209248754  0.237598309
Xstreamlengthimpaired -0.294621738  0.05931406  0.044977681  0.014014053
Per_Avg_Land_Cov      -0.439300475  0.11004133 -0.026368903 -0.174231995
poor_health_percent   -0.337934455  0.17288226  0.557365023  0.405847618
Z_Water_Index          0.066906200 -0.25708347 -0.034050989  0.128685949
GHG_Percap             0.015061283  0.09671103  0.058386754  0.050635813
UNEMPLOY               0.516246258 -0.31810061  0.281662237 -0.009851343
FOOD_INS              -0.262297705 -0.36347102 -0.458655874  0.474951723
VIO_CRIME              0.051033277  0.26352053 -0.013102229 -0.070536424
broadband_access       0.189800609  0.12427080  0.111202159  0.039978275
Black_equality         0.049448023 -0.16155858 -0.022052615 -0.044714251
Hispanic_equality      0.115140318  0.16967458  0.008091776 -0.007027731
                               PC17
AQI_Good              -0.0418559639
Bachelor_Over_25       0.4826097533
Per_Poverty            0.6935335308
Gini                  -0.3590028968
non_migration          0.1161855233
Per_Sev_Hous          -0.2343487966
Xstreamlengthimpaired  0.0210978527
Per_Avg_Land_Cov       0.0181418723
poor_health_percent    0.0899902108
Z_Water_Index          0.0276587565
GHG_Percap            -0.0003994658
UNEMPLOY               0.0544642952
FOOD_INS              -0.2520702031
VIO_CRIME              0.0668097852
```

```
broadband_access      -0.0136736230
Black_equality        -0.0522337755
Hispanic_equality     -0.0593578214
```

```r
summary(dat_comps)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4    PC5     PC6     PC7
Standard deviation     1.9524 1.5149 1.26465 1.16538 1.0752 1.01657 0.91783
Proportion of Variance 0.2242 0.1350 0.09408 0.07989 0.0680 0.06079 0.04955
Cumulative Proportion  0.2242 0.3592 0.45330 0.53319 0.6012 0.66198 0.71153
                           PC8     PC9   PC10    PC11   PC12    PC13    PC14
Standard deviation     0.89502 0.86580 0.8379 0.76955 0.7455 0.67433 0.63373
Proportion of Variance 0.04712 0.04409 0.0413 0.03484 0.0327 0.02675 0.02362
Cumulative Proportion  0.75865 0.80275 0.8440 0.87888 0.9116 0.93832 0.96195
                          PC15    PC16    PC17
Standard deviation     0.55121 0.45859 0.36435
Proportion of Variance 0.01787 0.01237 0.00781
Cumulative Proportion  0.97982 0.99219 1.00000
```

```r
# information about PCs
tidy(dat_comps, "pcs")
```

```
# A tibble: 17 x 4
      PC std.dev percent cumulative
   <dbl>   <dbl>   <dbl>      <dbl>
 1     1   1.95   0.224       0.224
 2     2   1.51   0.135       0.359
 3     3   1.26   0.0941      0.453
 4     4   1.17   0.0799      0.533
 5     5   1.08   0.068       0.601
 6     6   1.02   0.0608      0.662
 7     7   0.918  0.0495      0.712
 8     8   0.895  0.0471      0.759
 9     9   0.866  0.0441      0.803
10    10   0.838  0.0413      0.844
11    11   0.770  0.0348      0.879
12    12   0.746  0.0327      0.912
13    13   0.674  0.0268      0.938
14    14   0.634  0.0236      0.962
```

```
15     15    0.551 0.0179       0.980
16     16    0.459 0.0124       0.992
17     17    0.364 0.00781      1
```

Run Cluster GMM

Assumptions: - MSAs form clusters characterized by a multivariate distribution

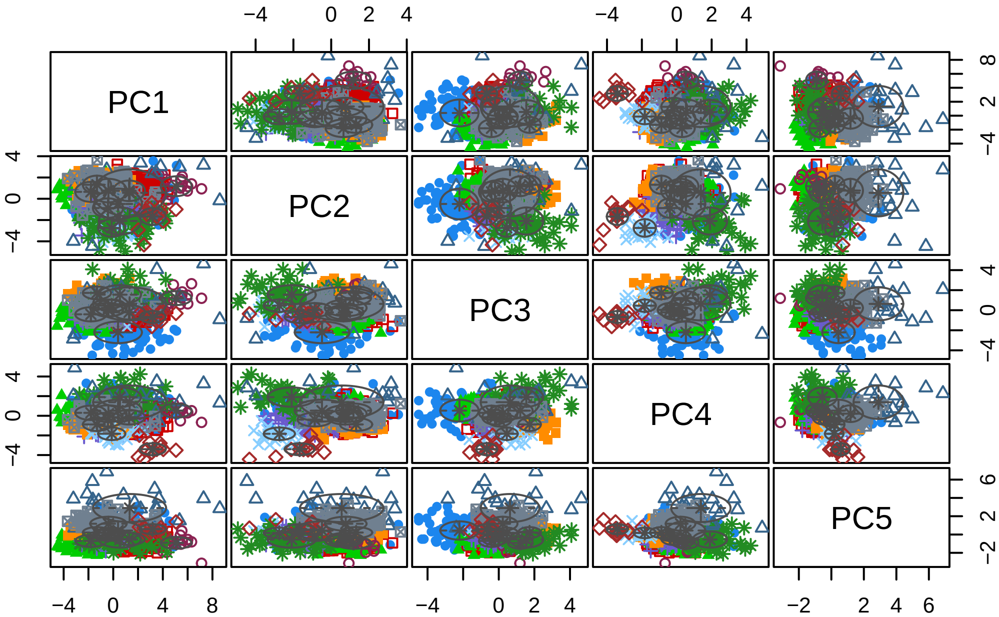
- Model forms: shape, volume, orientation
- GMM fits a series of models with different forms and numbers of clusters
- Models with highest probability and fewest parameters selected as most optimal
- Based on Bayesian Information Criterion (BIC)

```
#run GMM on 1-20 clusters
dat_pcs_mc <- Mclust(dat_pcs, G=c(1:20))
#print summary of model fit
summary(dat_pcs_mc)
```

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

Mclust VEI (diagonal, equal shape) model with 11 components:

 log-likelihood   n df       BIC       ICL
      -7022.462 869 80 -14586.31 -15086.77

Clustering table:
  1   2   3   4   5   6   7   8   9  10  11
 73 187 133 198  71  24  14  57  22  78  12
```

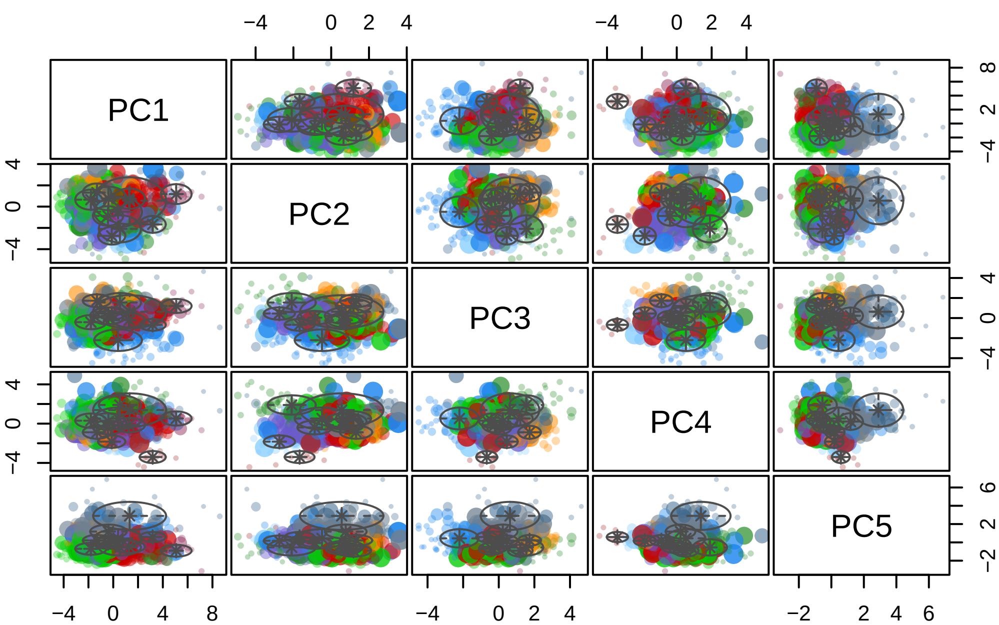
Cluster Assignments

```
plot(dat_pcs_mc, what = c("classification"))
```

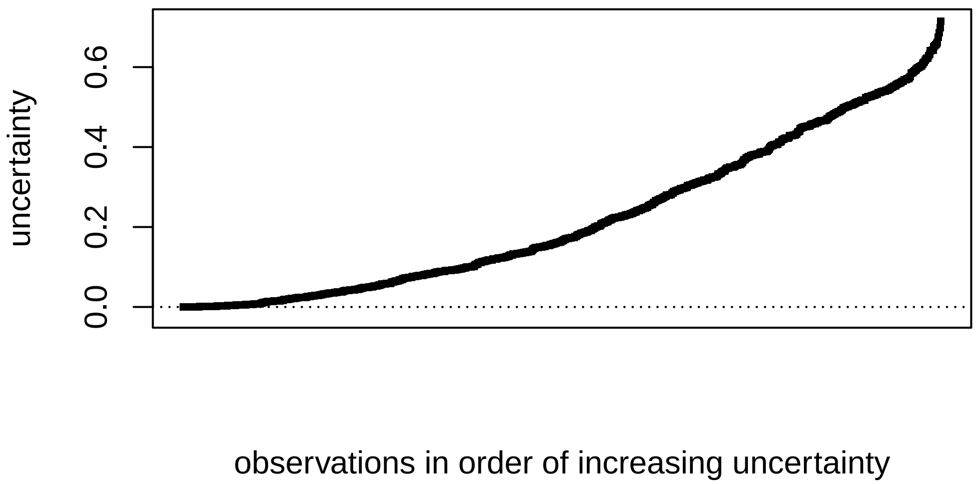


Uncertainty in assignments

```
write.csv(dat_pcs_mc$uncertainty,file=here("results/community_group_uncertainty.csv"))
plot(dat_pcs_mc, what = "uncertainty")
```

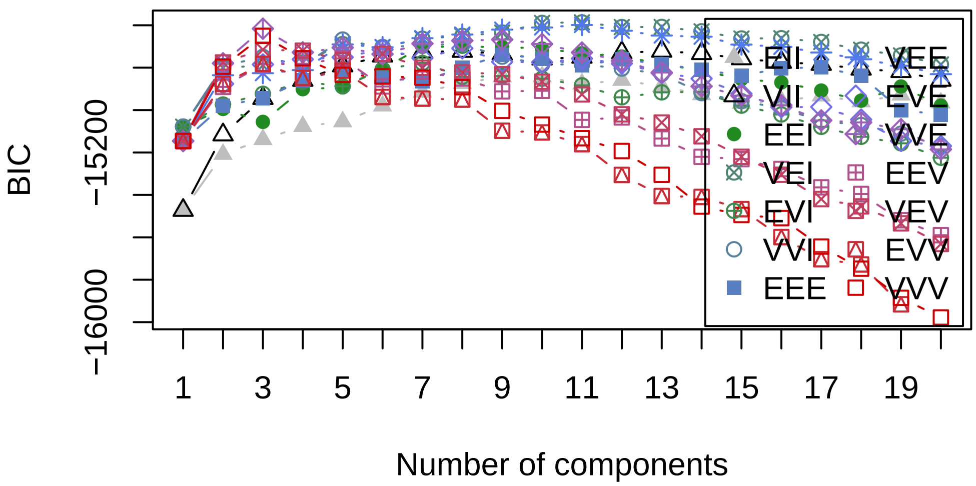


```r
uncerPlot(z = dat_pcs_mc$z)
```

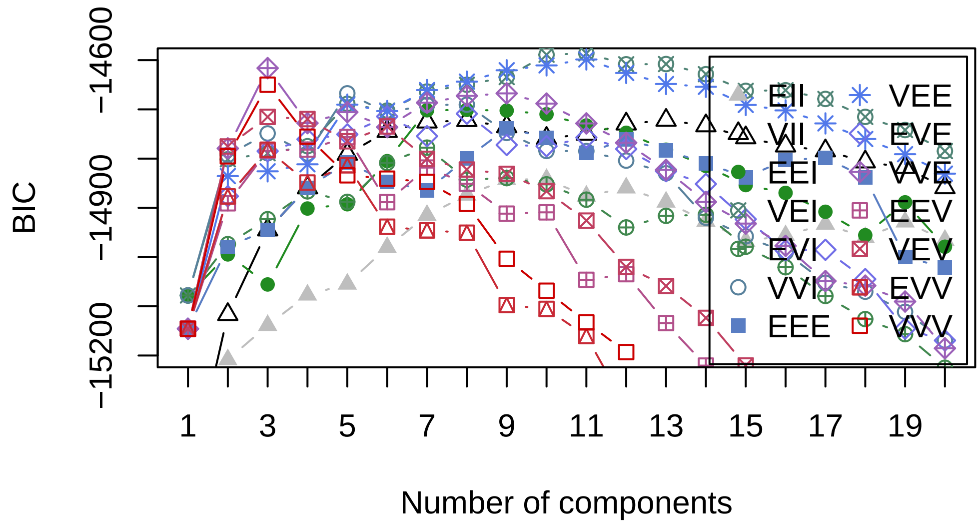


Model Comparisons

```r
#plot BIC scores of different models
plot(dat_pcs_mc, what='BIC')
```




```r
#zoom in on best-fitting models
plot(dat_pcs_mc, what='BIC', ylim=c(-15200,-14600))
```

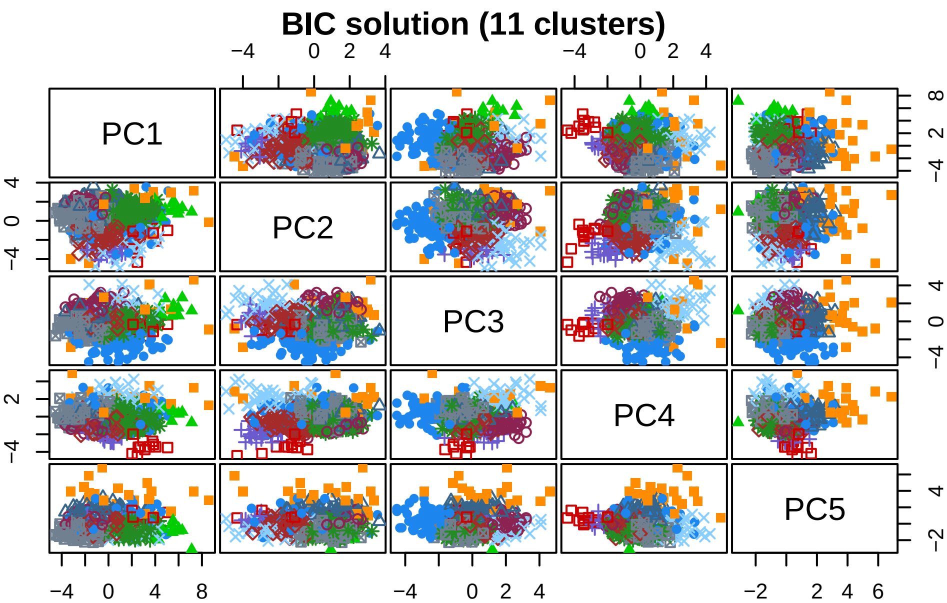



```
#model type VEI and VEE have similar BIC scores for 8-9 clusters
#do additional model comparisons
m_VEI_8 <- Mclust(dat_pcs, G=8, modelNames="VEI")
m_VEE_8 <- Mclust(dat_pcs, G=8, modelNames="VEE")
m_VEI_9 <- Mclust(dat_pcs, G=9, modelNames="VEI")
m_VEI_10 <- Mclust(dat_pcs, G=10, modelNames="VEI")
m_VEI_11 <- Mclust(dat_pcs, G=11, modelNames="VEI")
m_VEE_9 <- Mclust(dat_pcs, G=9, modelNames="VEE")
m_VEE_10 <- Mclust(dat_pcs, G=10, modelNames="VEE")
m_VEE_11 <- Mclust(dat_pcs, G=11, modelNames="VEE")
#extract BIC scores
BICs<-c(m_VEI_8$BIC[1],m_VEE_8$BIC[1],m_VEI_9$BIC[1],m_VEE_9$BIC[1],m_VEI_10$BIC[1],m_VEE_
#calculate change in BIC score, since in this method the goal to to maximize BIC
#we calculate change from highest scoring model
delta_BIC <- max(BICs) - BICs
#calculate BIC weights
w_BIC <- round(exp(-0.5*delta_BIC)/sum(exp(-0.5*delta_BIC)), digits=3)
#make table of results
results_table <- cbind.data.frame(clusters=c("VEI_8","VEE_8","VEI_9","VEE_9","VEI_10","VEE
#order by delta
results_table <- results_table[order(delta_BIC),]
rownames(results_table)<-NULL
#print table
kable(results_table[1:4], caption = "Model Comparison")
```

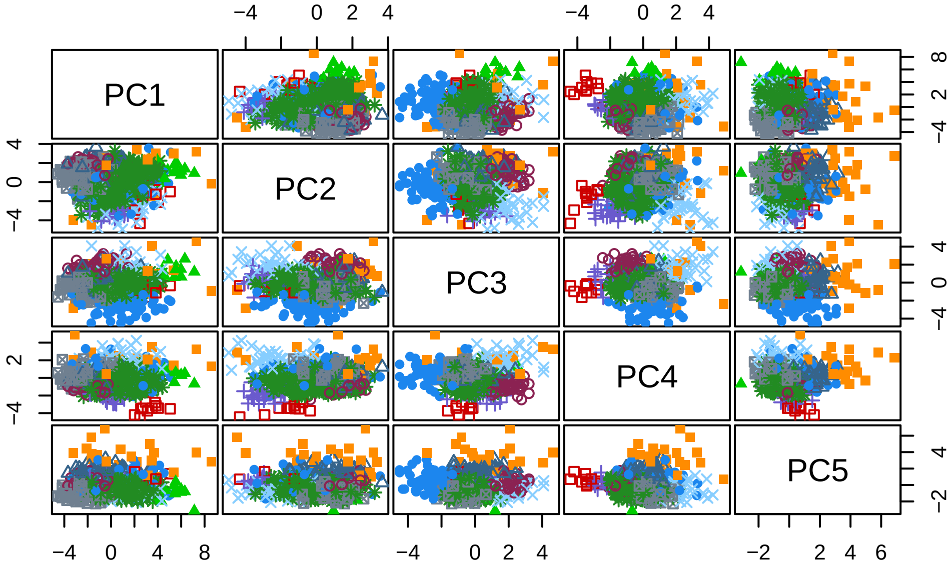
Table 1: Model Comparison

clusters	BIC	delta_BIC	weight
VEI_11	-14586.31	0.000000	0.792
VEI_10	-14589.00	2.690641	0.206
VEE_11	-14598.66	12.344357	0.002
VEE_10	-14610.68	24.368039	0.000
VEE_9	-14620.58	34.272145	0.000
VEI_9	-14634.38	48.065807	0.000
VEE_8	-14643.68	57.369224	0.000
VEI_8	-14650.08	63.763589	0.000

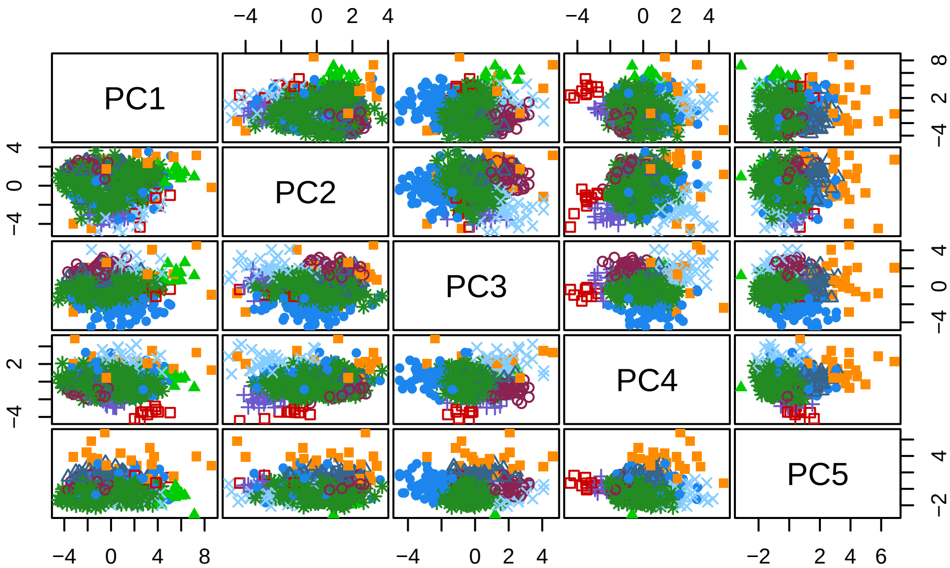
```
## providing some output about the model. It seems that VEI is the best model. But how man
## MODEL CHOSEN FOR REST OF ANALYSES

output <- clustCombi(data = dat_pcs, modelName = "VEI", G = 11)
# plot the hierarchy of combined solutions
plot(output, what = "classification")
```

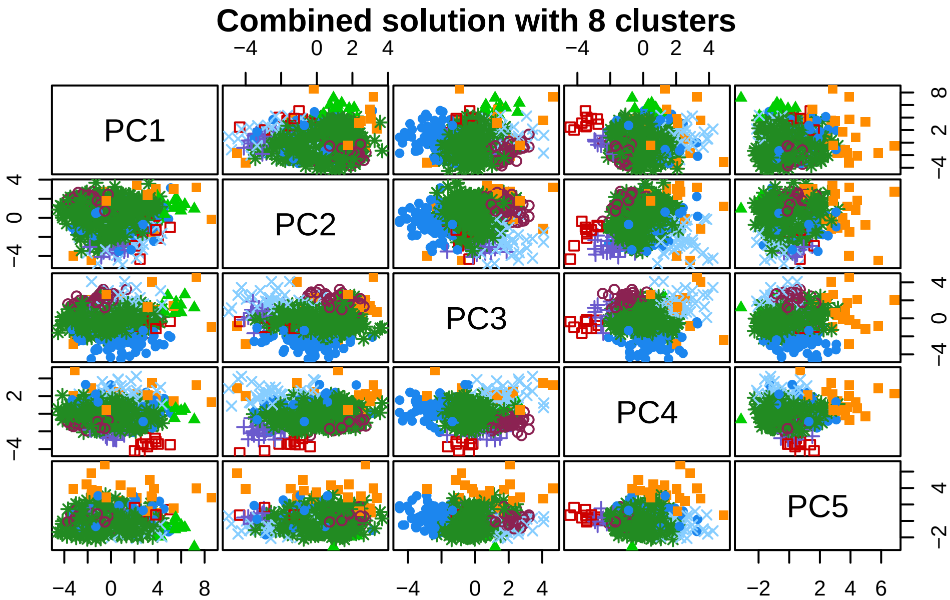


Combined solution with 10 clusters

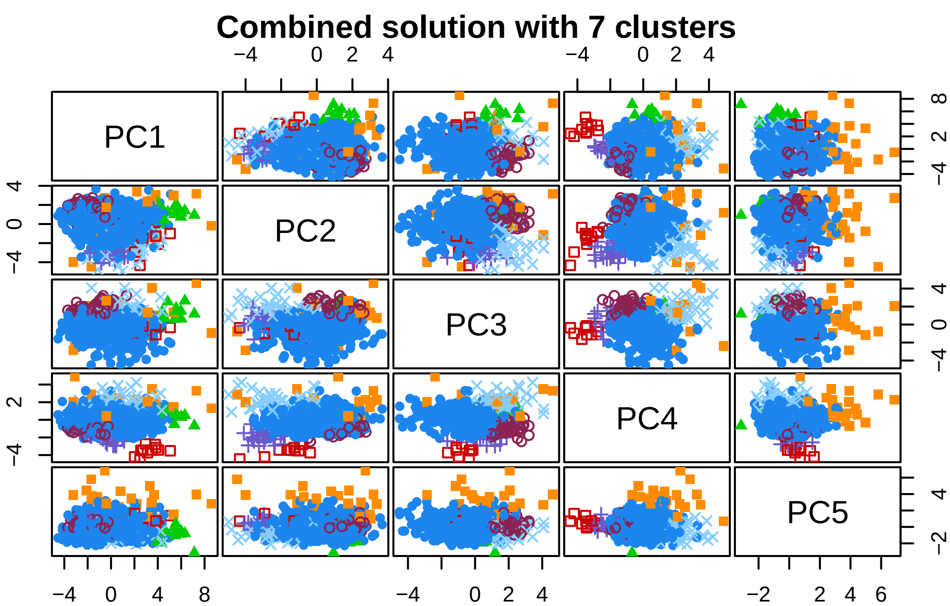


Combined solution with 9 clusters

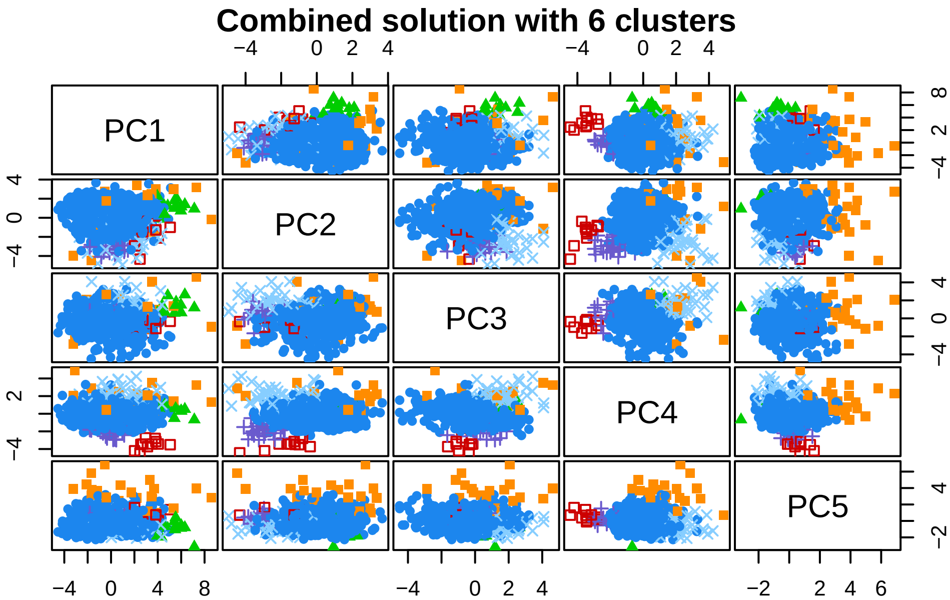


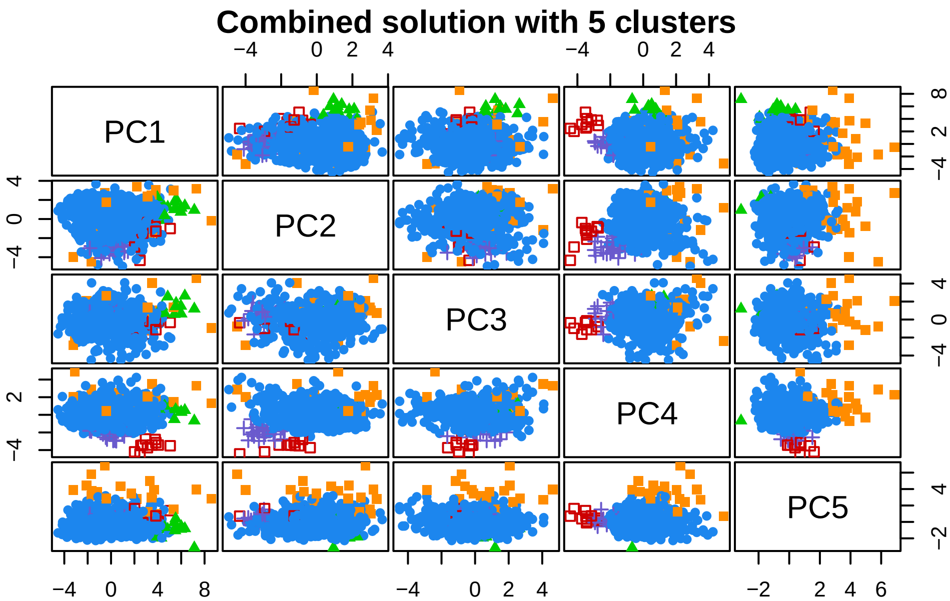


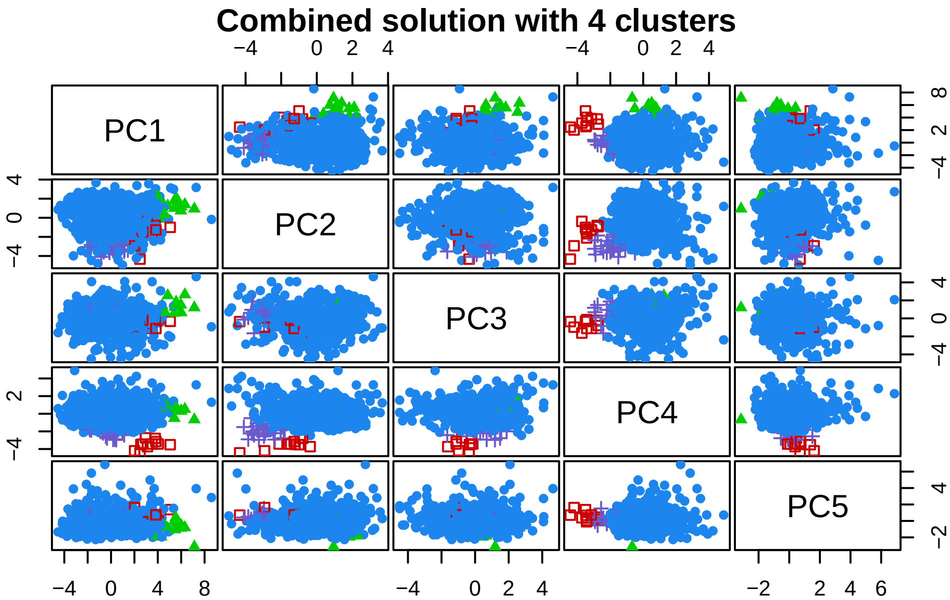
Combined solution with 8 clusters
PC1
PC2
PC3
PC4
PC5

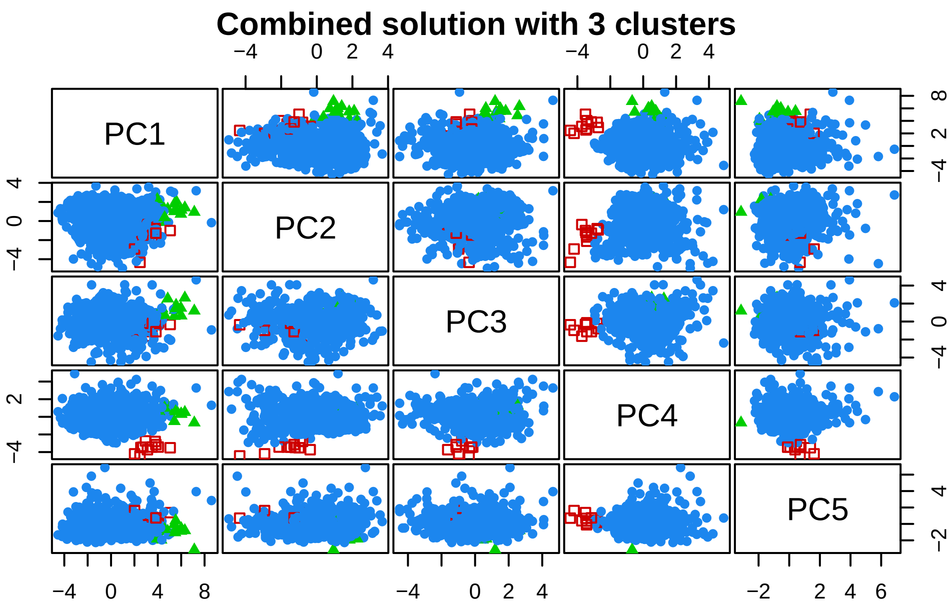


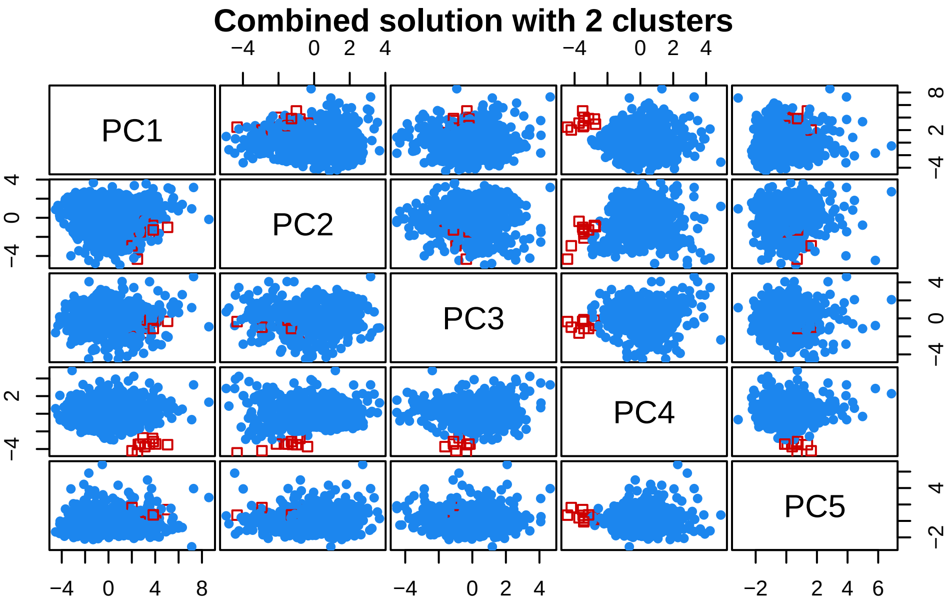
Combined solution with 7 clusters
PC1
PC2
PC3
PC4
PC5

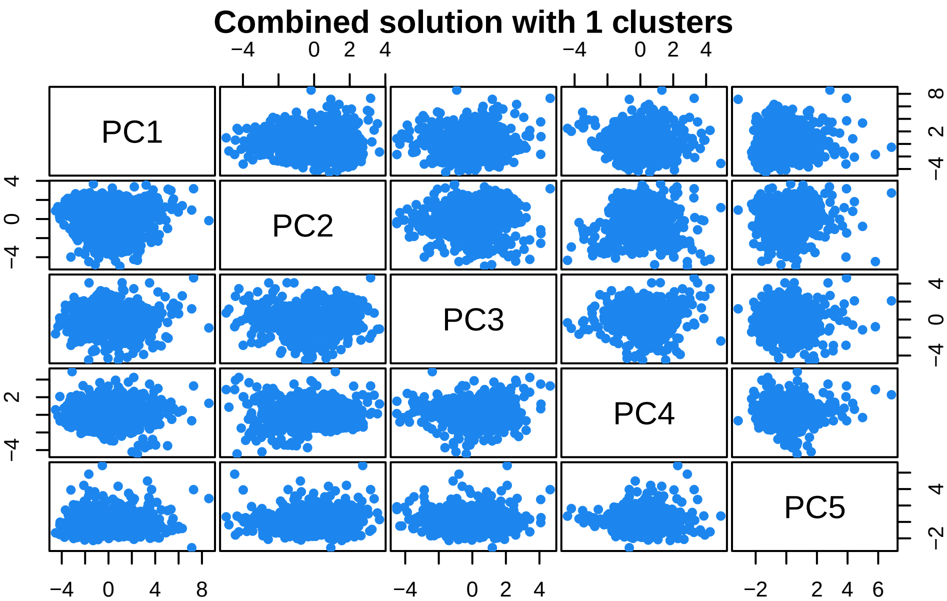

Combined solution with 6 clusters
PC1
PC2
PC3
PC4
PC5


Combined solution with 5 clusters
PC1
PC2
PC3
PC4
PC5

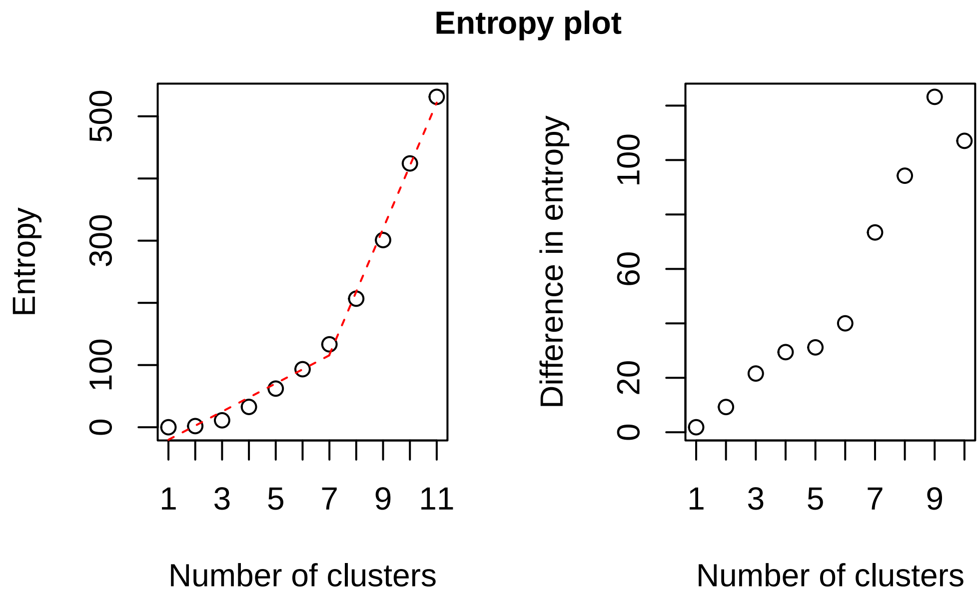

Combined solution with 4 clusters
PC1
PC2
PC3
PC4
PC5

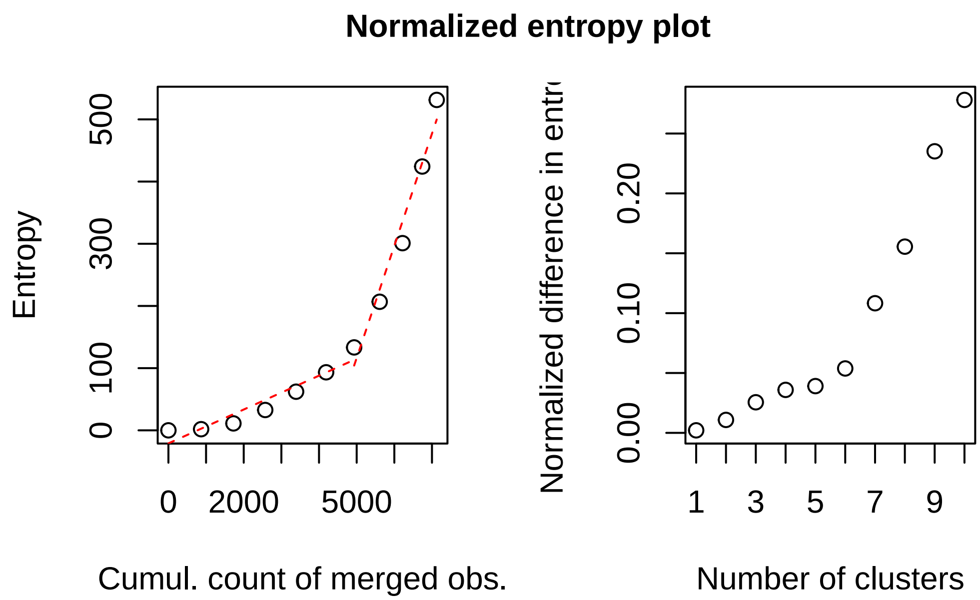

Combined solution with 3 clusters
PC1
PC2
PC3
PC4
PC5



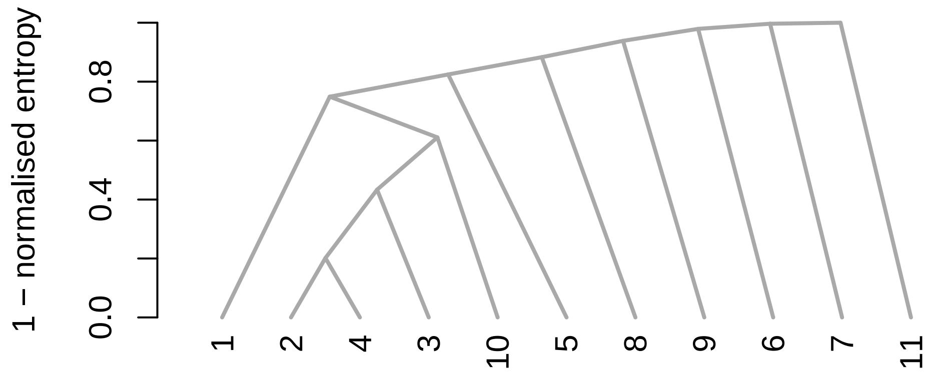



```
# plot some "entropy plots" which may help one to select the number of classes
plot(output, what = "entropy")
```






```
# plot the tree structure obtained from combining mixture components
plot(output, what = "tree")
```




```
head( output$combiz[[output$MclustOutput$G]] )
```

```
                      [,1]         [,2]         [,3]         [,4]         [,5]
Aberdeen, SD 6.432924e-01 3.974472e-04 0.0386093109 0.0162019774 1.404253e-08
Aberdeen, WA 9.427474e-03 7.759977e-01 0.0694798829 0.1397955273 2.266526e-05
Abilene, TX  9.644515e-01 5.484786e-03 0.0206430241 0.0003166873 2.540331e-14
Ada, OK      6.792666e-02 4.719357e-01 0.1388080998 0.2728284337 2.617152e-06
Adrian, MI   5.547996e-06 9.116646e-05 0.0006045694 0.0006490153 9.945213e-01
Akron, OH    8.026771e-06 1.003766e-04 0.0000153712 0.0016469574 9.853800e-01
                      [,6]         [,7]         [,8]         [,9]        [,10]
Aberdeen, SD 5.538256e-10 5.960728e-14 4.596205e-03 3.914632e-02 0.2577563279
Aberdeen, WA 1.565435e-06 9.426227e-07 2.838064e-05 8.469585e-05 0.0051611747
Abilene, TX  6.486613e-14 1.593203e-09 6.510071e-03 2.492665e-03 0.0001012194
Ada, OK      2.849144e-07 3.057900e-07 5.368111e-04 5.546647e-04 0.0474064483
Adrian, MI   1.097613e-08 2.492575e-11 1.957276e-05 1.757200e-04 0.0039330644
Akron, OH    2.538377e-06 2.915347e-09 3.722966e-04 1.528955e-03 0.0109454954
                    [,11]
Aberdeen, SD 1.024742e-21
Aberdeen, WA 8.005957e-10
Abilene, TX  1.918474e-23
Ada, OK      8.103400e-13
Adrian, MI   5.256702e-19
Akron, OH    1.984818e-15
```

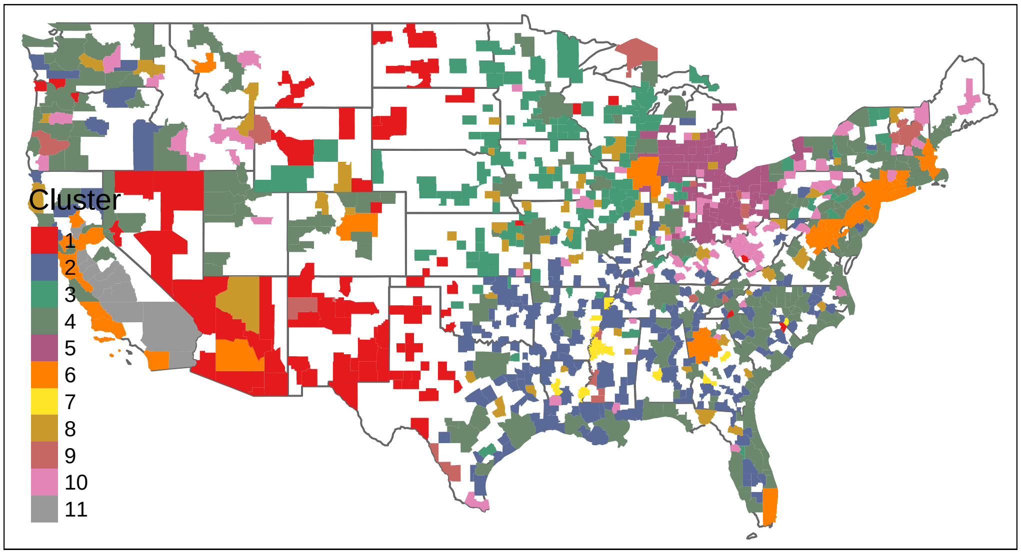
```
## output the probabilities by cluster.
write.csv(output$combiz[[output$MclustOutput$G]],file=here("results/cluster_probabilities.
```

Map Classification Output

```r
#extract classifications, e.g., which city belongs to which cluster, export to csv and rel

### Change this value to pick a different model
## currently the best solution is VEI with 11 clusters.

#####NOTE THIS OUTPUT IS FOR VEI with 10 clusters ####
write.csv(m_VEI_11$classification,file=here("results/cluster_assignment_pca.csv"))
data<- read.csv(here('results/cluster_assignment_pca.csv'))

#import shapefile
msa_Boundary <-readOGR(here("data/MSA"),"tl_2015_us_cbsa")
```

```
OGR data source with driver: ESRI Shapefile
Source: "/Users/clipo/PycharmProjects/sustainable_communities/data/MSA", layer: "tl_2015_us_
with 929 features
It has 12 fields
Integer64 fields read as strings:  ALAND AWATER
```

```r
#merge them
merged <- merge(msa_Boundary,data,by.x="NAME",by.y="X")
#remove any cases MSAs with no cluster assignments
merged_clean <- merged[!is.na(merged$x),]
#convert x to factor
merged_clean$Cluster <- as.factor(merged_clean$x)
#convert to sf object
cluster_sf <- st_as_sf(merged_clean)
#make us map
us_map <- tm_shape(us_states) + tm_borders()
#combine them
cluster_map <- tm_shape(cluster_sf) + tm_fill(col="Cluster", palette="Set1")
#plot them
us_map + cluster_map
```

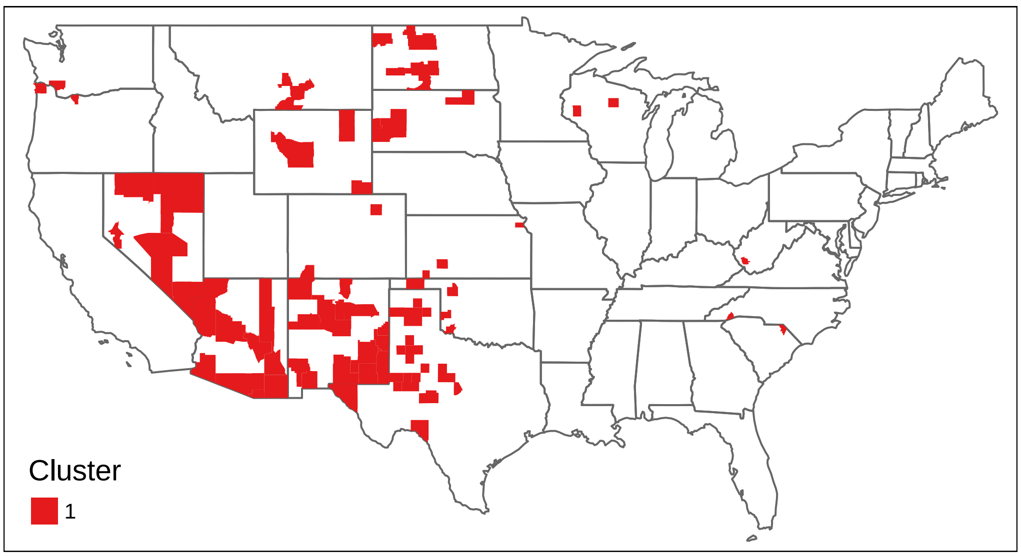



```r
## now show the distribution of each of the clusters one at a time
## dat_pcs_mc$G is the number of clusters
par(mfrow = c(6, 2))  # Set up a 2 x 2 plotting space
output_map <- us_map
library(svMisc)
```

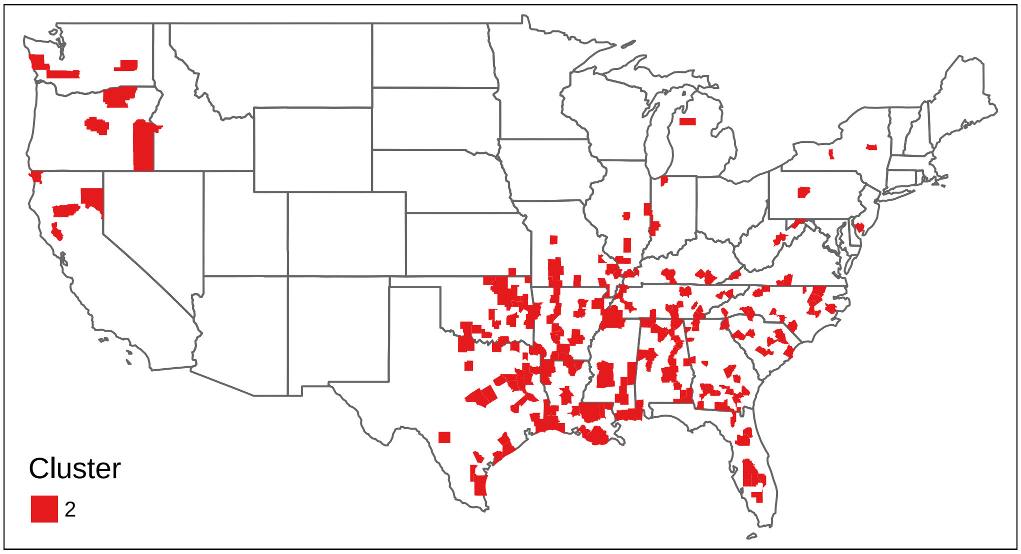
```
Attaching package: 'svMisc'

The following object is masked from 'package:utils':

    ?
```

```r
pb = txtProgressBar(min = 1, max = dat_pcs_mc$G, initial = 1)

for (g in 1:dat_pcs_mc$G) {
 setTxtProgressBar(pb,g)
 subset_dat <-subset(data,x==g)
  merged <- merge(msa_Boundary,subset_dat,by.x="NAME",by.y="X")
  #remove any cases MSAs with no cluster assignments
  merged_clean <- merged[!is.na(merged$x),]
  #convert x to factor
  merged_clean$Cluster <- as.factor(merged_clean$x)
  #convert to sf object
```

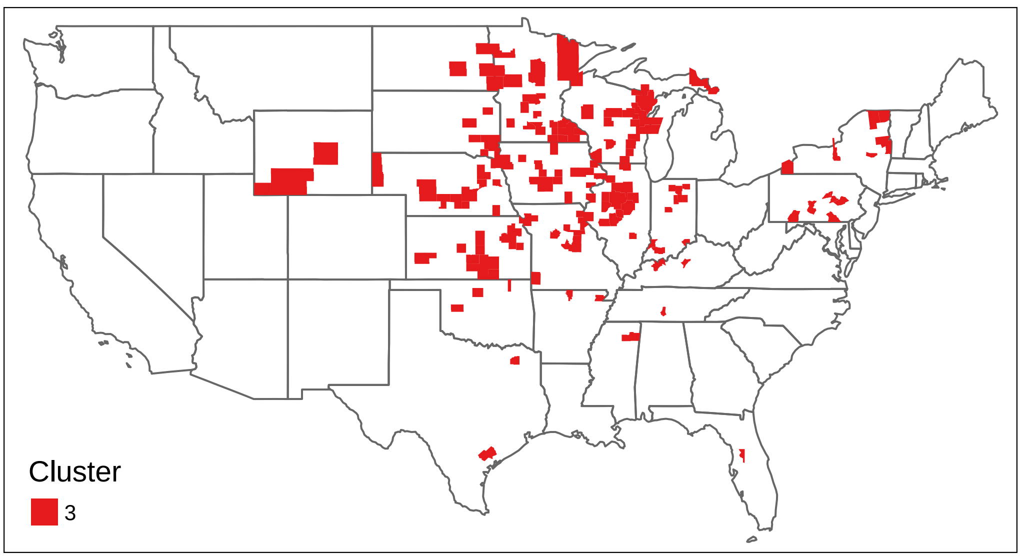
```
  cluster_sf <- st_as_sf(merged_clean)
  #make us map
  us_map <- tm_shape(us_states) + tm_borders()
  #combine them
  new_cluster_map <- tm_shape(cluster_sf)+tm_fill(col="Cluster", palette="Set1")
#plot them
  #plot them
  print (us_map + new_cluster_map)
  #dev.off()#
}
```



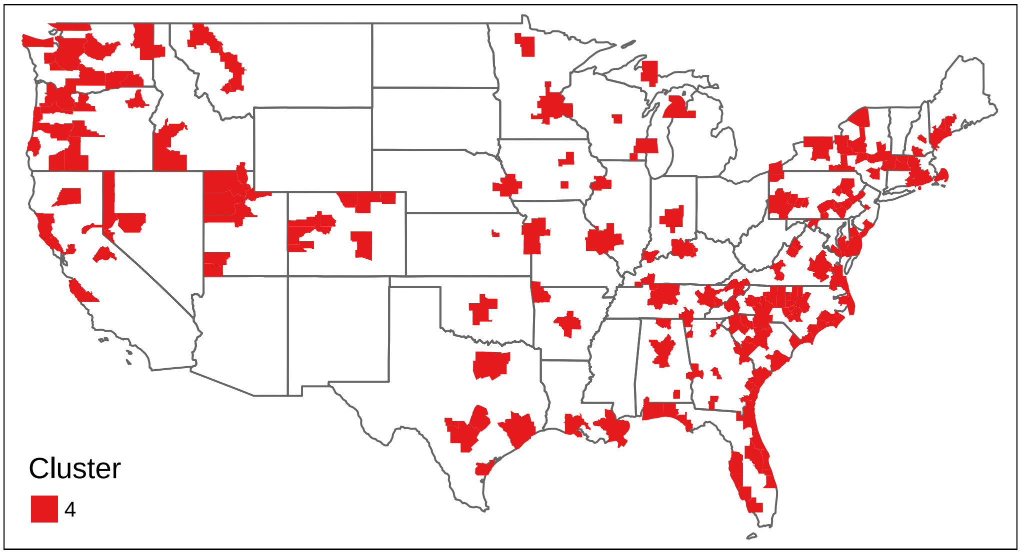
========


Cluster
2

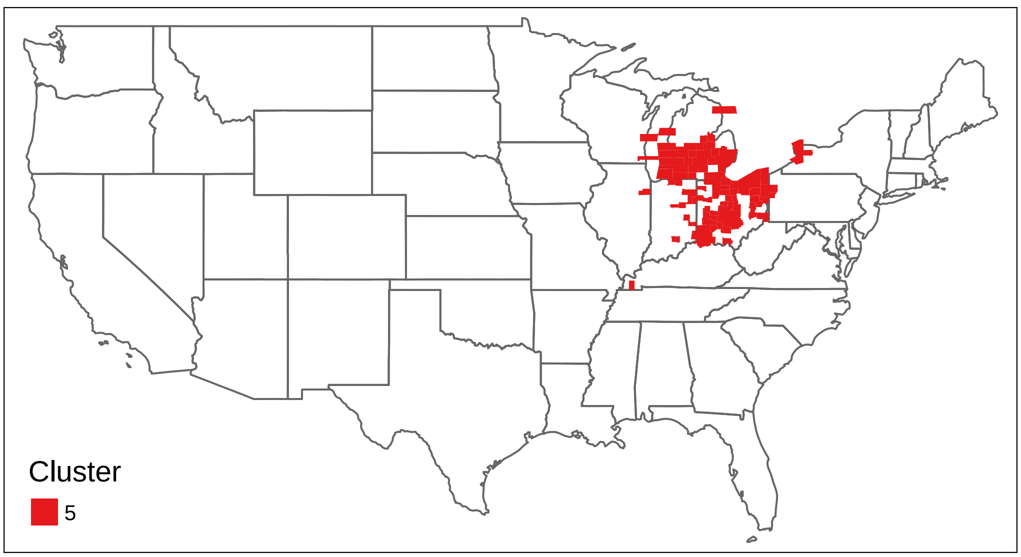
========


Cluster
3

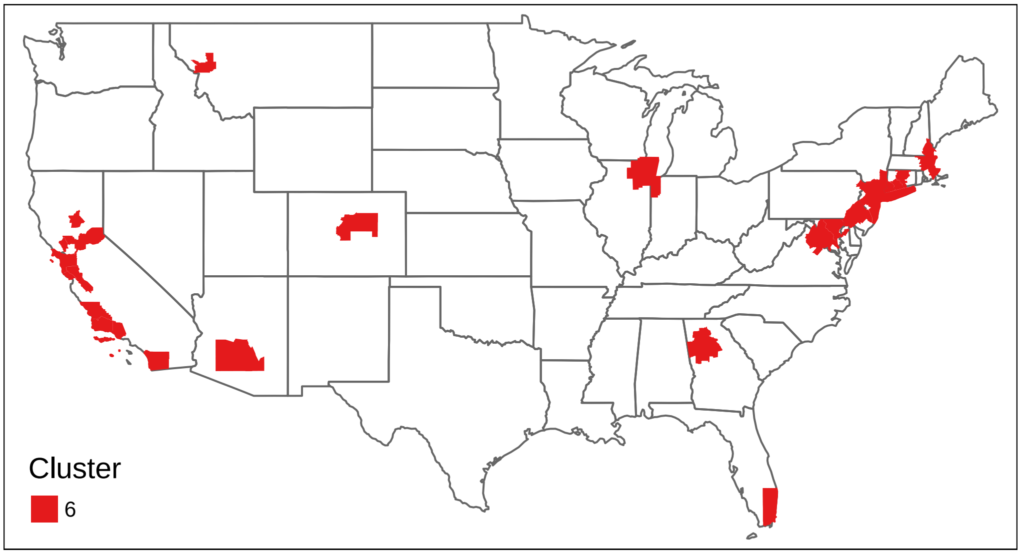
========


Cluster
4

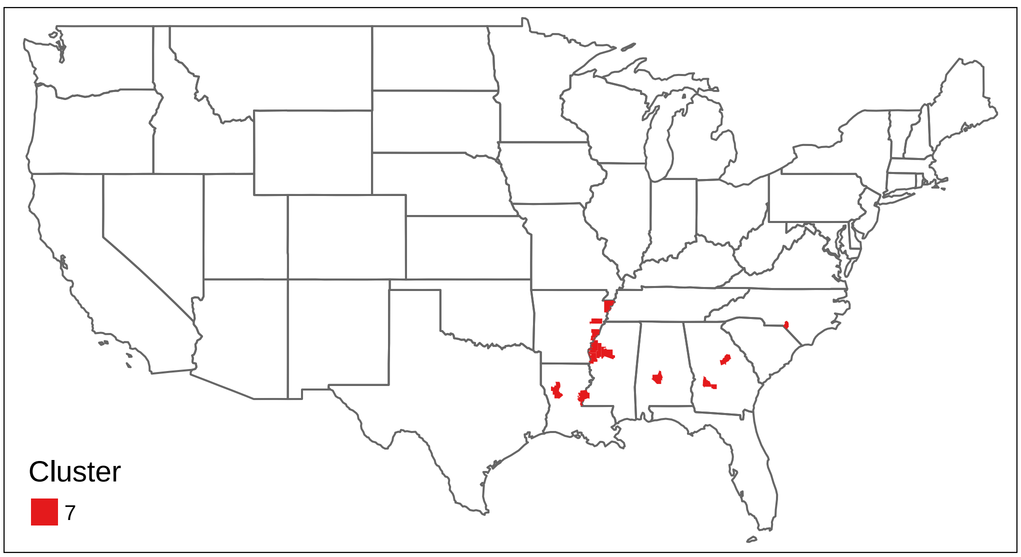
========


Cluster
5

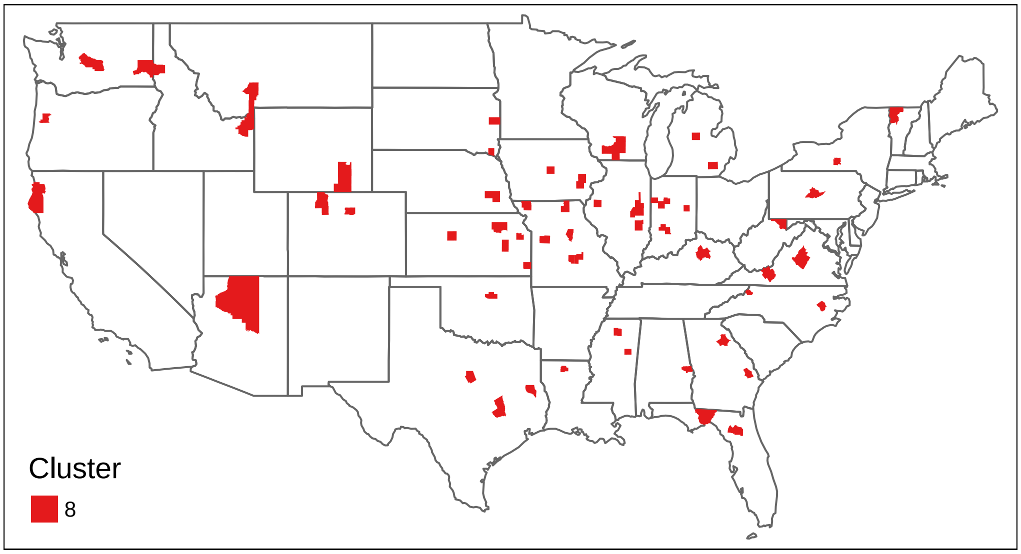
========


Cluster
6

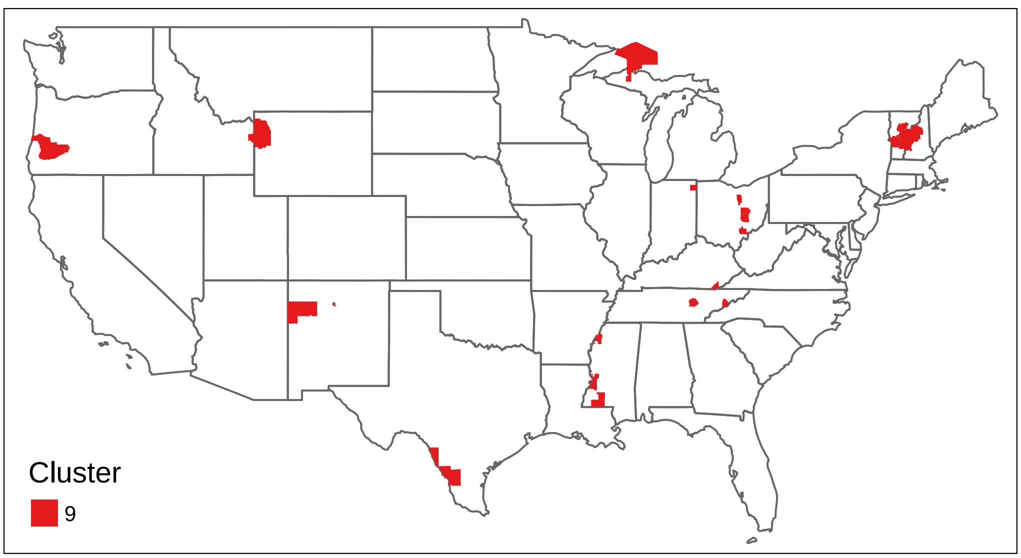
========


Cluster
7

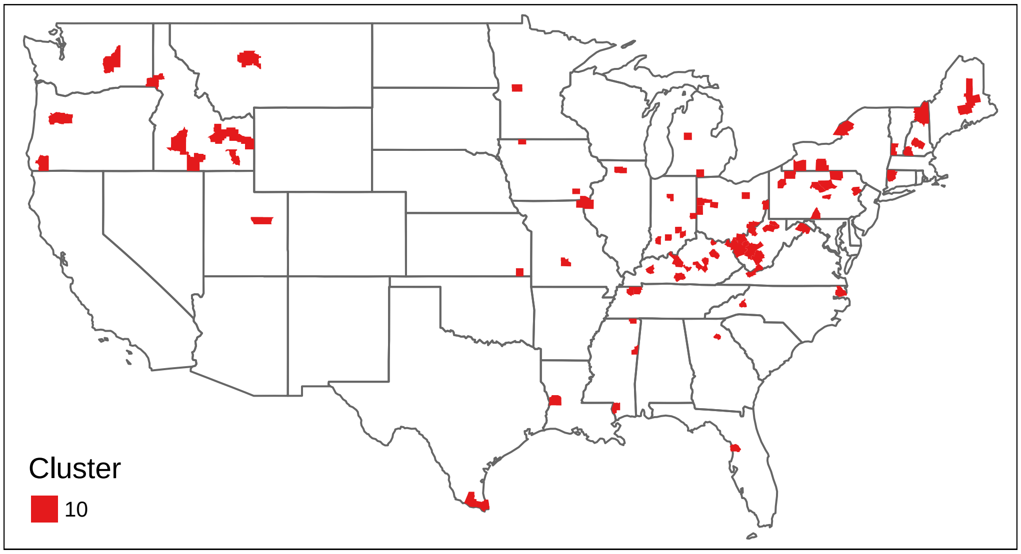
========


Cluster
8

========


Cluster
9

========



========

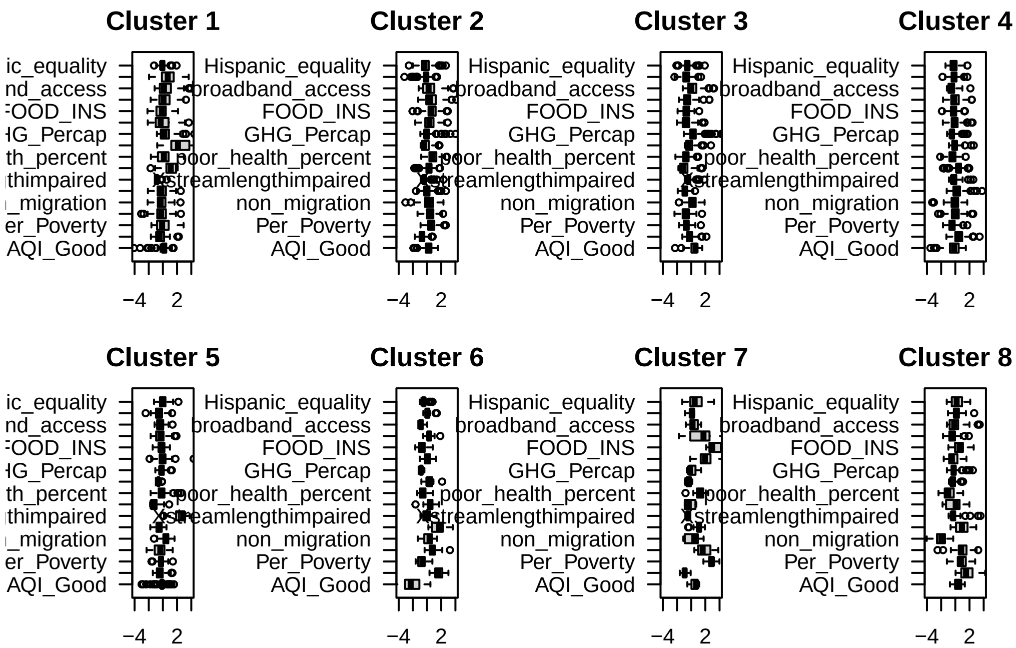
Metrics for clusters

```
#new data frame
dat_clust <- as.data.frame(dat)
#add names
dat_clust$ME <- rownames(dat)
rownames(dat_clust) <- NULL
```

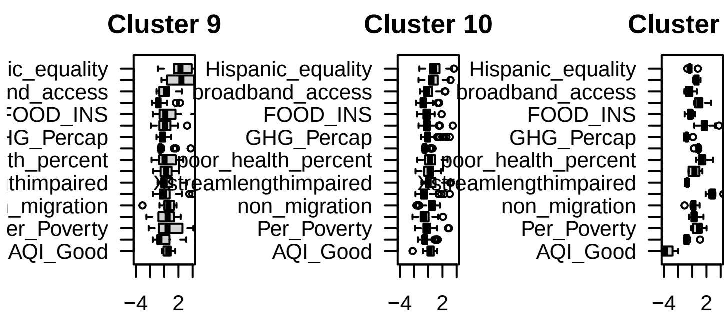
```
#merge in cluster assignments
dat_clust <- merge(dat_clust,data,by.x="ME",by.y="X")
#calculate cluster means for different variables
c1m <- data.frame(one=colMeans(subset(dat_clust, x=="1")[2:18]))
c2m <- data.frame(two=colMeans(subset(dat_clust, x=="2")[2:18]))
c3m <- data.frame(three=colMeans(subset(dat_clust, x=="3")[2:18]))
c4m <- data.frame(four=colMeans(subset(dat_clust, x=="4")[2:18]))
c5m <- data.frame(five=colMeans(subset(dat_clust, x=="5")[2:18]))
c6m <- data.frame(six=colMeans(subset(dat_clust, x=="6")[2:18]))
c7m <- data.frame(seven=colMeans(subset(dat_clust, x=="7")[2:18]))
c8m <- data.frame(eight=colMeans(subset(dat_clust, x=="8")[2:18]))
c9m <- data.frame(nine=colMeans(subset(dat_clust, x=="9")[2:18]))
c10m <- data.frame(ten=colMeans(subset(dat_clust, x=="10")[2:18]))
c11m <- data.frame(eleven=colMeans(subset(dat_clust, x=="11")[2:18]))
cluster_means <- cbind.data.frame(c1m,c2m,c3m,c4m,c5m,c6m,c7m,c8m,c9m,c10m,c11m)
cluster_means$variable <- row.names(cluster_means)
rownames(cluster_means) <- NULL
```

Box Plot of Scores

```
#boxplots of scores for clusters
par(mfrow=c(2,4), mar=c(2.5,5,2.5,3))
boxplot(subset(dat_clust, x=="1")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
boxplot(subset(dat_clust, x=="2")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
boxplot(subset(dat_clust, x=="3")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
boxplot(subset(dat_clust, x=="4")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
boxplot(subset(dat_clust, x=="5")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
boxplot(subset(dat_clust, x=="6")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
boxplot(subset(dat_clust, x=="7")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
boxplot(subset(dat_clust, x=="8")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
```



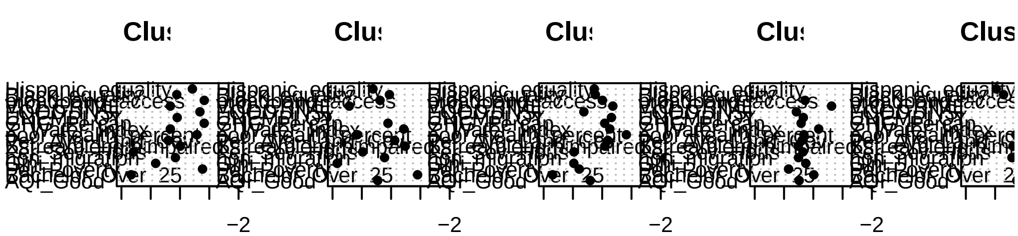

```
boxplot(subset(dat_clust, x=="9")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
boxplot(subset(dat_clust, x=="10")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
boxplot(subset(dat_clust, x=="11")[2:18], horizontal=T, las=1, ylim=c(-4,4), main="Cluster
par(mfrow=c(1,1))
```



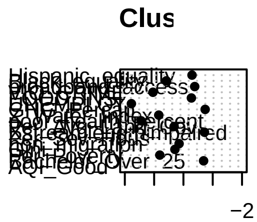
Dotplot of clusters

```
#dotplots of cluster mean values
par(mfrow=c(2,5))
dotchart(cluster_means$one, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 1',
abline(v=0, lwd=2)
dotchart(cluster_means$two, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 2',
abline(v=0, lwd=2)
```

```
dotchart(cluster_means$three, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 3'
abline(v=0, lwd=2)
dotchart(cluster_means$four, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 4',
abline(v=0, lwd=2)
dotchart(cluster_means$five, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 5',
abline(v=0, lwd=2)
dotchart(cluster_means$six, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 6',
abline(v=0, lwd=2)
dotchart(cluster_means$seven, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 7'
abline(v=0, lwd=2)
dotchart(cluster_means$eight, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 8'
abline(v=0, lwd=2)
dotchart(cluster_means$nine, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 9',
abline(v=0, lwd=2)
dotchart(cluster_means$ten, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 10',
abline(v=0, lwd=2)
```



```
dotchart(cluster_means$eleven, labels=cluster_means$variable,xlim=c(-2,2), main='Cluster 1
abline(v=0, lwd=2)
par(mfrow=c(1,1))
```


Clus
Hispanic_equality
AQI_Good
−2