Creating a Social Vulnerability Index with ACS Data

Patrick Bixler and Ethan Tenison

May 16, 2022

Constructing the Social Vulnerability index

For reference, the following (Medium Post)[https://medium.com/analytics-vidhya/the-factor-analysis-for-constructing-a-composite-index-2496686fc54c] was used to guide the svi's consturction. While it was conducted in Python using a variety of packages, I was able to recreate it in R using primarily the `psych` and `caret` packages which are used extensively in statistics.

ACS Variables

In order to construct the Social Vulnerability Index, 18 variables were pulled from the 2020 ACS 5-year estimates using the `tidycensus` package. The variables are as follows:

- Wealth
  - QRICH = Percent Households Earning over $200,000 annually
  - MDHSEVAL = Median Housing Value
  - PERCAP = Per Capita Income
  - MDGRENT = Median Gross Rent
- Language & Education
  - QESL = Percent Speaking English as a Second Language with Limited Proficiency
  - QSPANISH = Percent Hispanic
  - QED12LES = Percent Less than high school education for population over 25 years and older
- Elderly
  - QSSBEN = Percent Households Receiving Social Security Benefits
  - QAGEDEP = Percent Population under 5 years or 65 and over
  - MEDAGE = Median age
- Housing Status
  - PPUNIT = People per Unit (Average household size)
  - QFAM = Percent Children Living with both parents
- Social Status
  - QCVLUN = Percent Unemployment for Civilian in Labor Force 16 Years and Over
  - QBLACK = Percent Black or African American Alone
  - QNOAUTO = Percent Housing Units with No Car
  - QPOVTY = Percent Poverty
- Gender
  - QFEMALE = Percent Female
  - QFEMLBR = Percent Female Participation in Labor Force

Tidycensus

All the data was pulled for the state of Texas. The results data frame is in long format and must be pivoted wider for analysis. You can find documentation about the package on the Tidycensus website.

```
#acs variables
vars <- c(
"QRICH" = "B19001_017", #need to divide households
"households" = "B09019_002",
"MDGRENT" = "B25064_001",
"MDHSEVAL" =    "B25077_001",
"PERCAP" = "B19301_001",
"QESL-Spanish" = "B06007_005", #need to be added together and divided by pop
"QESL-Other" = "B06007_008",
"QSPANISH" = "B03001_003", # need to divide by pop
"POP" = "B03001_001",
"QED12LES" = "B16010_002", # divide by pop
"QSSBEN" = "B19055_002", #divide by pop
"QAGEDEP-under5" = "B06001_002",#add together and divide by pop
"QAGEDEP-over65" = "B18135_024",
"MEDAGE"= "B07002_001",
"PPUNIT" = "B25010_001",
"QFAM_under6" = "B05009_003", #add together and divide by children
"QFAM_to17" = "B05009_021",
"children" = "B05009_001",
"QCVLUN" = "B23025_005", # divide by pop
"QBLACK" = "B18101B_001", #divide by pop,
"QNOAUTO" = "B08203_002", #divide by households
"QPOVTY" = "B17020_002", #divide by pop
"QFEMALE" = "B01001_026", # divide by pop
"wlab1" = "C23002A_004", #can't really pull female labor participation except by race
"wlab2" = "C23002B_004", #Then divide by over 16
"wlab3" = "C23002C_004",
"wlab4" = "C23002D_004",
"wlab5" = "C23002E_004",
"wlab6" = "C23002F_004",
"wlab7" = "C23002G_004",
"f1" = "B01001_030", #pulling pop by age is actually the worst!
"f2" = "B01001_031",
"f3" = "B01001_032",
"f4" = "B01001_033",
"f5" = "B01001_034",
"f6" = "B01001_035",
"f7" = "B01001_036",
"f8" = "B01001_037",
"f9" = "B01001_038",
"f10" = "B01001_039",
"f11" = "B01001_040",
"f12" = "B01001_041",
"f13" = "B01001_042",
"f14" = "B01001_043",
"female_over65" = "B15001_076"
)
```

```r
#counties <- c("Travis", "Bastrop","Blanco", "Burnet", "Caldwell",
  #             "Fayette", "Hays", "Lee", "Llano", "Williamson")

#census_api_key("cbef6ecf15f8b2c9e1cb135b09f5bee3ead9c5a1", install=TRUE)
acs <-get_acs(state="TX", geography="tract",year = 2020,
              variables=vars, geometry= F)
```

```
## Getting data from the 2016-2020 5-year ACS
```

```r
df_raw <- acs |>
  select(GEOID, variable, estimate) |>
  pivot_wider(id_cols = GEOID,
              names_from = variable,
              values_from = estimate)
```

Data cleaning

Many of the variables total estimates, and need to be divided by the population. Additionally, some variables have much narrower populations, such as women in the labor force, and require more variables to construct the numerator. `NA` values and infinite values also have to be removed from the resulting data frame. In most cases `NA` were substituted with 0, but infinite values were removed completely.

```r
df <- df_raw |>
  mutate(
    QRICH = QRICH / households,
    QESL = (`QESL-Spanish` + `QESL-Other`) / POP,
    QSPANISH = QSPANISH / POP,
    QED12LES = QED12LES / POP,
    QSSBEN = QSSBEN / POP,
    QAGEDEP = (`QAGEDEP-under5` + `QAGEDEP-over65`) / POP,
    QFAM = (QFAM_under6 + QFAM_to17) / children,
    QCVLUN = QCVLUN / POP,
    QBLACK = QBLACK / POP,
    QNOAUTO = QNOAUTO / households,
    QPOVTY = QPOVTY / POP,
    QFEMALE = QFEMALE / POP,
    QFEMLBR = (wlab1 + wlab2 + wlab3 + wlab4 + wlab5 + wlab6 + wlab7) /
      (
        f1 + f2 + f3 + f4 + f5 + f6 + f7 + f8 + f9 + f10 + f11 + f12 + f13 + f14 + female_over65
      )
  ) |>
  select(
    -c(
      #Variables to remove
      POP,
      children,
      QFAM_under6,
      QFAM_to17,
      households,
      female_over65,
      `QAGEDEP-under5`,
```

```
      `QAGEDEP-over65`,
      `QESL-Spanish`,
      `QESL-Other`,
      starts_with("wlab"),
      "f1","f2","f3" ,"f4","f5" ,"f6" ,"f7" ,"f8" ,"f9" ,"f10" ,"f11" ,"f12","f13" ,
      "f14" ,
    )
  ) |>
  mutate(across(2:19,~replace_na(.x,0))) |>
  filter(across(everything(), ~!is.infinite(.)))

head(df)
```

```
## # A tibble: 6 x 19
##   GEOID QFEMALE QSPANISH MEDAGE QNOAUTO QED12LES  QPOVTY  QBLACK   QRICH  QSSBEN
##   <chr>   <dbl>    <dbl>  <dbl>   <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 4800~  0.483    0.0825   44    0.00794   0.0597 0.174   0.0526  0.0128  0.143
## 2 4800~  0.0314   0.285    35.2 0          0.230   0       0.00288 0       0.00144
## 3 4800~  0        0.280    41.1 0          0.281   0.00150 0       0       0
## 4 4800~  0.523    0.423    33.3 0.0189     0.0876 0.149   0.151   0.00300 0.132
## 5 4800~  0.523    0.0859   35.7 0.0283     0.179  0.195   0.270   0.00346 0.169
## 6 4800~  0.479    0.329    30.4 0.0291     0.132  0.117   0.420   0.00260 0.0993
## # ... with 9 more variables: PERCAP <dbl>, QCVLUN <dbl>, PPUNIT <dbl>,
## #   MDGRENT <dbl>, MDHSEVAL <dbl>, QESL <dbl>, QAGEDEP <dbl>, QFAM <dbl>,
## #   QFEMLBR <dbl>
```

Minmax scaling

Because each variable is in different units, it was important to transform the data to the same scale. Min-max scaling was used to convert the units from 0-1 using the `caret` function `preProcess`.

```
minmax <- preProcess(df[,-1], method = "range")

transformed <- predict(minmax, df[,-1])
```

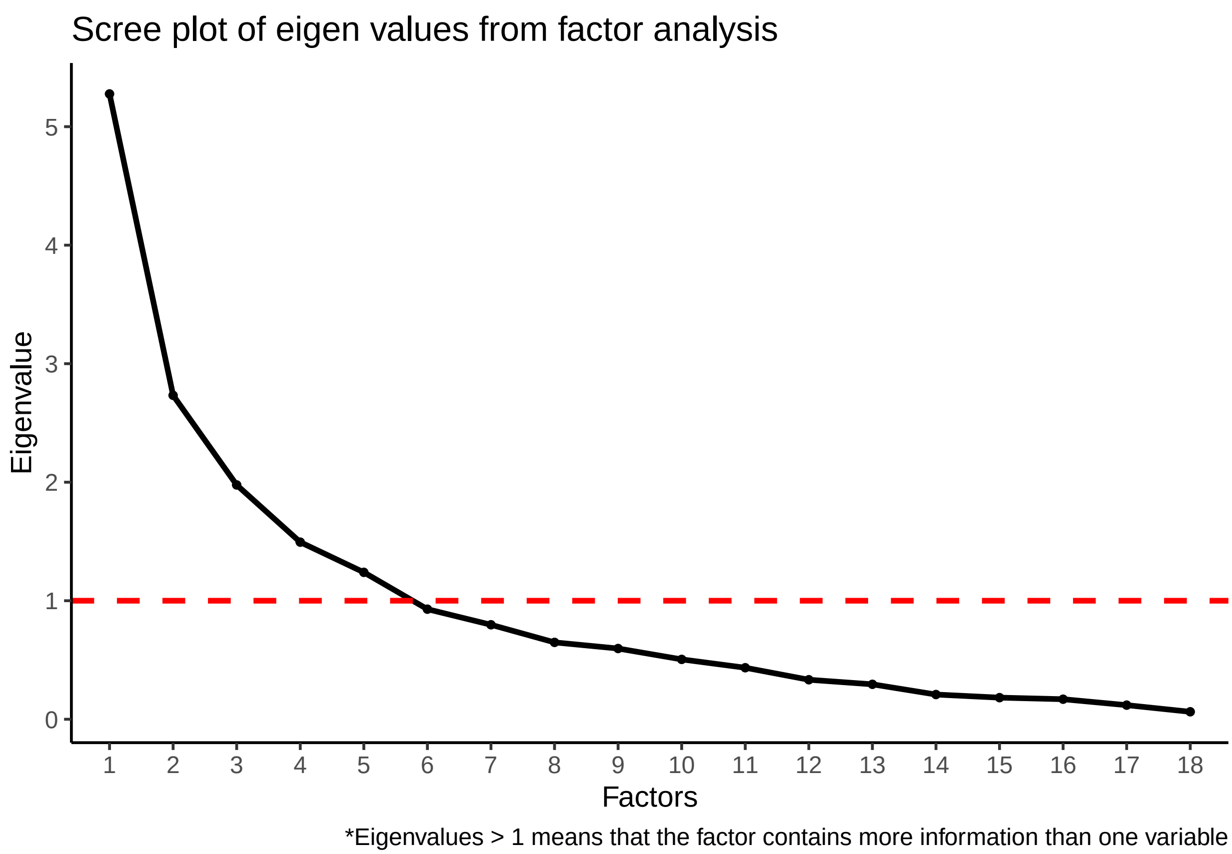
Factor Analysis

With the transformed data, a factor analysis with varimax rotation was performed to reduce dimensionality. According to the Kaiser criterion, there were 5 relevant factors, where eigenvalues are greater than 1.You can view these results in the following scree plot.

```
my_fa <-
  fa(
    r = transformed,
    nfactors = 18,
    rotate = "varimax",
    fm = "minres"
  )

print(my_fa$e.values)
```

```
##  [1] 5.27617031 2.73309043 1.97686753 1.49423853 1.23957082 0.92861566
##  [7] 0.79717087 0.64841646 0.59668977 0.50493119 0.43454446 0.33329760
## [13] 0.29468580 0.20836633 0.18223112 0.16910026 0.11901037 0.06300248
```

```
e <- as.data.frame(my_fa$e.values) |>
  rename(e_values = `my_fa$e.values`) |>
  rownames_to_column("Factor")

e$Factor <- factor(e$Factor, levels = unique(e$Factor))

theme_set(theme_classic())
scree_plot <- ggplot(data = e, aes(x = Factor , y = e_values, group = 1)) +
  geom_line(size = 1) +
  geom_point(size = 1) +
  geom_hline(yintercept = 1, linetype='dashed', col = 'red', size = 1) +
  labs(
    title = "Scree plot of eigen values from factor analysis",
    y = "Eigenvalue",
    x = "Factors",
    caption = "*Eigenvalues > 1 means that the factor contains more information than one variable"
  )

scree_plot
```



Factor Analysis Results

Determining the dominant indicators in a factor is based on the their loading scores. A loading score of > 0.5 signifies importance. Still, each factor includes all indicators to some degree, unless the loading score is 0. Please note that the factors were created using the minimum residual method, and as such each factor column is labeled MR. After computing each factor, they were reordered based on variance explained.

Most important variables in each factor: * **Factor 1**: QRICH, PERCAP, MDHSEVAL * **Factor 2**: QSPANISH, QED12LES, QESL * **Factor 3**: MEDAGE, QSSBEN, QAGEDEP * **Factor 4**: QNOAUTO, QPOVTY(at .45) * **Factor 5**: QFEMALE

Additionally, from the summary table below we can see that the first 5 factors account for approximately 72% of variance explained. Generally speaking, > 60% explained variance is considered a good factor analysis.

```
print(my_fa)
```

```
## Factor Analysis using method =  minres
## Call: fa(r = transformed, nfactors = 18, rotate = "varimax", fm = "minres")
## Standardized loadings (pattern matrix) based upon correlation matrix
##            MR1  MR10   MR2   MR4   MR5   MR3   MR7   MR6   MR9  MR12  MR14
## QFEMALE   0.08  0.02  0.20  0.05  0.76  0.14  0.10 -0.18  0.05  0.12  0.07
## QSPANISH -0.28  0.77 -0.15  0.07  0.09  0.21 -0.32  0.03 -0.19  0.12 -0.04
## MEDAGE    0.27 -0.12  0.77 -0.08  0.12 -0.08 -0.03  0.01  0.14 -0.03  0.00
## QNOAUTO  -0.10  0.14  0.07  0.73  0.03 -0.18  0.16 -0.01 -0.09  0.13 -0.03
## QED12LES -0.33  0.82  0.05  0.16 -0.10  0.14  0.01 -0.06 -0.06  0.06 -0.13
## QPOVTY   -0.30  0.41 -0.05  0.45  0.09  0.12  0.05 -0.07 -0.25  0.23 -0.20
## QBLACK   -0.12 -0.11 -0.10  0.17  0.10 -0.01  0.69 -0.03 -0.11  0.23  0.04
## QRICH     0.92 -0.18  0.06 -0.06 -0.05  0.00 -0.07 -0.01  0.09 -0.09  0.00
## QSSBEN   -0.05 -0.10  0.93  0.12  0.02 -0.10 -0.06 -0.17 -0.02  0.01 -0.05
## PERCAP    0.90 -0.26  0.13 -0.11  0.11 -0.15 -0.05  0.14  0.07 -0.12  0.02
## QCVLUN   -0.09  0.07 -0.04  0.09  0.07  0.03  0.13  0.03 -0.05  0.40  0.00
## PPUNIT   -0.09  0.36 -0.11 -0.25  0.22  0.76 -0.02  0.08  0.11  0.09  0.03
## MDGRENT   0.33 -0.15 -0.14 -0.12  0.19  0.05  0.09  0.15  0.07  0.00  0.36
## MDHSEVAL  0.84 -0.17  0.03 -0.05  0.03  0.02 -0.07  0.00  0.09 -0.05  0.10
## QESL     -0.17  0.86 -0.15  0.07  0.02  0.06  0.00  0.07  0.00  0.04  0.02
## QAGEDEP   0.03 -0.02  0.87  0.01  0.11  0.05 -0.03 -0.14  0.02 -0.10 -0.02
## QFAM      0.34 -0.17  0.12 -0.21  0.11  0.15 -0.20  0.16  0.60 -0.14  0.05
## QFEMLBR   0.07  0.04 -0.22 -0.01 -0.14  0.04 -0.02  0.68  0.07  0.05  0.05
##            MR8  MR15  MR13  MR11  MR17  MR16 MR18   h2    u2 com
## QFEMALE   0.01  0.00  0.00  0.00  0.00  0.00    0 0.71 0.286 1.5
## QSPANISH -0.12 -0.13  0.03 -0.06  0.22  0.00    0 0.98 0.015 2.6
## MEDAGE   -0.12  0.30  0.01  0.02 -0.03  0.00    0 0.84 0.160 1.9
## QNOAUTO   0.00  0.00  0.00  0.00  0.00  0.00    0 0.66 0.344 1.5
## QED12LES  0.00  0.12  0.01  0.20 -0.08  0.00    0 0.92 0.076 1.9
## QPOVTY    0.36 -0.05  0.01  0.00 -0.02  0.00    0 0.77 0.228 5.9
## QBLACK    0.00  0.00  0.00  0.00  0.00  0.00    0 0.63 0.374 1.7
## QRICH     0.02  0.04  0.17 -0.03  0.00 -0.02    0 0.95 0.054 1.2
## QSSBEN   -0.02 -0.02  0.02  0.02 -0.03  0.07    0 0.94 0.064 1.2
## PERCAP   -0.06  0.00  0.01  0.07  0.02  0.04    0 1.00 0.005 1.5
## QCVLUN    0.01  0.00  0.00  0.00  0.00  0.00    0 0.21 0.788 1.7
## PPUNIT    0.02 -0.01  0.00  0.00  0.00  0.00    0 0.86 0.138 2.1
## MDGRENT  -0.04  0.00  0.00  0.00  0.00  0.00    0 0.37 0.632 4.2
## MDHSEVAL -0.02  0.00 -0.18 -0.03 -0.03 -0.01    0 0.81 0.191 1.3
## QESL      0.08 -0.03 -0.02 -0.09 -0.04  0.00    0 0.82 0.179 1.2
```

```
## QAGEDEP   0.08 -0.13 -0.02 -0.03  0.03 -0.07    0 0.84 0.161 1.2
## QFAM     -0.03  0.01  0.00  0.00 -0.01  0.00    0 0.68 0.320 3.2
## QFEMLBR  -0.01  0.00  0.00  0.00  0.00  0.00    0 0.55 0.452 1.4
##
##                        MR1 MR10  MR2  MR4  MR5  MR3  MR7  MR6  MR9 MR12 MR14
## SS loadings           3.03 2.53 2.43 0.98 0.78 0.78 0.70 0.63 0.54 0.38 0.22
## Proportion Var        0.17 0.14 0.14 0.05 0.04 0.04 0.04 0.03 0.03 0.02 0.01
## Cumulative Var        0.17 0.31 0.44 0.50 0.54 0.59 0.62 0.66 0.69 0.71 0.72
## Proportion Explained  0.22 0.19 0.18 0.07 0.06 0.06 0.05 0.05 0.04 0.03 0.02
## Cumulative Proportion 0.22 0.41 0.59 0.66 0.72 0.78 0.83 0.88 0.92 0.95 0.96
##                        MR8 MR15 MR13 MR11 MR17 MR16 MR18
## SS loadings           0.18 0.14 0.07 0.06 0.06 0.01 0.00
## Proportion Var        0.01 0.01 0.00 0.00 0.00 0.00 0.00
## Cumulative Var        0.73 0.74 0.74 0.75 0.75 0.75 0.75
## Proportion Explained  0.01 0.01 0.00 0.00 0.00 0.00 0.00
## Cumulative Proportion 0.97 0.99 0.99 0.99 1.00 1.00 1.00
##
## Mean item complexity =  2.1
## Test of the hypothesis that 18 factors are sufficient.
##
## The degrees of freedom for the null model are  153  and the objective function was  11.06 with Chi S
## The degrees of freedom for the model are -18  and the objective function was  0
##
## The root mean square of the residuals (RMSR) is  0
## The df corrected root mean square of the residuals is  NA
##
## The harmonic number of observations is  6893 with the empirical chi square  0  with prob <  NA
## The total number of observations was  6893  with Likelihood Chi Square =  0  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.002
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                    MR1 MR10  MR2  MR4  MR5  MR3
## Correlation of (regression) scores with factors   0.98 0.95 0.97 0.79 0.84 0.85
## Multiple R square of scores with factors          0.96 0.90 0.93 0.62 0.71 0.72
## Minimum correlation of possible factor scores     0.92 0.80 0.87 0.23 0.42 0.44
##                                                    MR7  MR6  MR9  MR12  MR14
## Correlation of (regression) scores with factors   0.80 0.77 0.71  0.53  0.49
## Multiple R square of scores with factors          0.64 0.60 0.51  0.29  0.24
## Minimum correlation of possible factor scores     0.28 0.20 0.02 -0.43 -0.53
##                                                     MR8  MR15  MR13  MR11  MR17
## Correlation of (regression) scores with factors    0.65  0.65  0.60  0.61  0.51
## Multiple R square of scores with factors           0.42  0.43  0.36  0.38  0.26
## Minimum correlation of possible factor scores     -0.16 -0.15 -0.29 -0.25 -0.47
##                                                    MR16 MR18
## Correlation of (regression) scores with factors    0.31    0
## Multiple R square of scores with factors           0.10    0
## Minimum correlation of possible factor scores     -0.80   -1
```

Computing SVI

The factor scores are based on their z-scores, making comparison difficult. For that reason, minmax-scaling was used again to convert the scale from 0-1.Then, the direction of the components were adjusted to correspond theoretically to higher social vulnerability. Since component one has to do with wealth, where lower values increase vulnerability, the direction is changed to negative. Finally, all the scores are summed together to get the final numerical composite score.

```
#NOTE factor numbers will change based on subsequent years! Change accordingly
scores <- as.data.frame(my_fa$scores) |>
  select(MR1, MR10, MR2, MR4, MR5)


minmax <- preProcess(scores, method = "range")

trans_scores <- predict(minmax, scores)

svi <- trans_scores |>
  mutate(MR1 = MR1*-1,
    SVI = MR1 + MR10 + MR2 + MR4 + MR5) |>
  select(SVI)

minmax <- preProcess(svi, method = "range")

svi <- predict(minmax, svi)
```

Plotting

```
#enter your key census_api_key("Your key here")
acs2 <-get_acs(state="TX", geography="tract",year = 2020,
              variables="B03001_001", geometry= T)
```

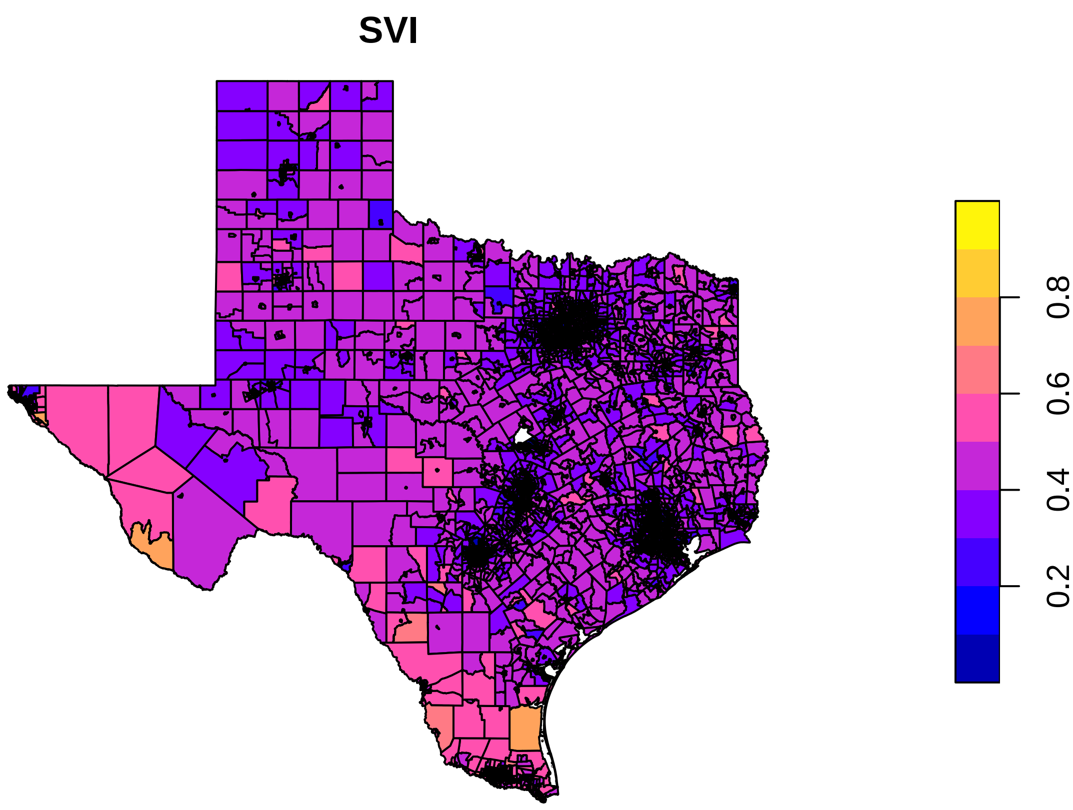
```
## Getting data from the 2016-2020 5-year ACS

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

##   |                                                                              |
```

```
tracts <- acs2 |>
  distinct(GEOID, .keep_all = TRUE) |>
  select(GEOID, geometry)

df2 <- df |>
  bind_cols(svi)

library(leaflet)
library(sf)
```

```
## Linking to GEOS 3.8.1, GDAL 3.2.1, PROJ 7.2.1
```

```
df2 <- df2 |>
  right_join(tracts, by = "GEOID") |>
  st_as_sf() |>
```

```
  filter(!is.na(GEOID)) #/>
  #mutate(index = scale(index))

df2 <- na.omit(df2)

plot(df2["SVI"])
```



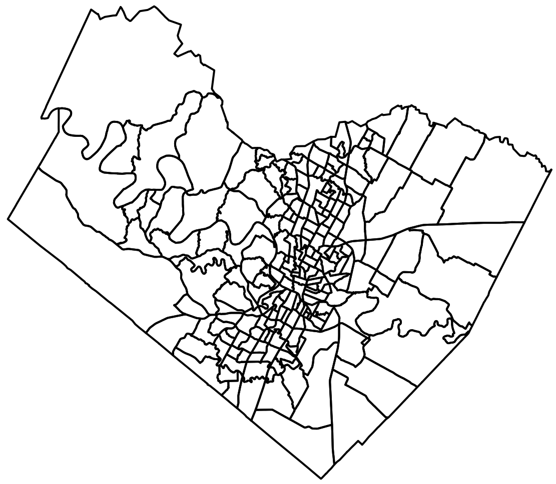

```
df2 <- as.data.frame(df2) |>
  select(GEOID, SVI)

write.csv(df2, "data/processed/svi_tracts.csv")
```

SVI at Major Metro Counties

```
austin_tracts <- tracts("TX", "Travis")
```

```
##   |                                                                              |
```

```
plot(austin_tracts$geometry)
```



```r
df2 <- df2 |>
  right_join(tracts, by = "GEOID") |>
  st_as_sf() |>
  filter(!is.na(GEOID)) #|>

#travis_svi <- st_join(austin_tracts, df2)
```

Census Block Groups

The following code was used to compute the SVI scores for census block groups. Although it is virtually identical, female labor force participation was not available and had to be taken out. The most important indicators also changed slightly from the census tracts.

```r
#acs variables
vars <- c(
"QRICH" = "B19001_017", #need to divide households
"households" = "B09019_002",
"MDHSEVAL" =    "B25077_001",
"MDGRENT" = "B25064_001",
"PERCAP" = "B19301_001",
"QESL-Spanish" = "C16002_004", #need to be added together and divided by household
"QESL-Other" = "C16002_013",
"QSPANISH" = "B03002_012", # need to divide by pop
"POP" = "B01003_001",
```

```r
"QED12LES" = "B28006_002", # divide by pop
"QSSBEN" = "B19055_002", #divide by pop
"QAGEDEP-under5-male" = "B01001_003",#add together and divide by pop
"QAGEDEP-under5-female" = "B01001_027",
"QAGEDEP-over65" = "B09021_022",
"MEDAGE"= "B01002_001",
"PPUNIT" = "B25010_001",
"QFAM" = "B09002_002", #divide by children
"children" = "B09002_001",
"QCVLUN" = "B23025_005", # divide by pop
"QBLACK" = "B02009_001", #divide by pop,
"QNOAUTO-owner" = "B25044_003", #add and divide by households
"QNOAUTO-renter" = "B25044_010",
"QPOVTY" = "B17021_002", #divide by pop
"QFEMALE" = "B01001_026"#, # divide by pop
# "wlab1" = #"C23002A_004", #add together COULD NOT FIND
# "wlab2" = #"C23002B_004", #Then divide by over 16
# "wlab3" = #"C23002C_004",
# "wlab4" = #"C23002D_004",
# "wlab5" = #"C23002E_004",
# "wlab6" = #"C23002F_004",
# "wlab7" = #"C23002G_004",
# "f1" = "B01001_030", #pulling pop by age is actually the worst!
# "f2" = "B01001_031",
# "f3" = "B01001_032",
# "f4" = "B01001_033",
# "f5" = "B01001_034",
# "f6" = "B01001_035",
# "f7" = "B01001_036",
# "f8" = "B01001_037",
# "f9" = "B01001_038",
# "f10" = "B01001_039",
# "f11" = "B01001_040",
# "f12" = "B01001_041",
# "f13" = "B01001_042",
# "f14" = "B01001_043",
# "female_over65" = "B15011_034"
)

counties <- c("Travis", "Bastrop","Blanco", "Burnet", "Caldwell",
              "Fayette", "Hays", "Lee", "Llano", "Williamson")

#enter your key census_api_key("Your key here")
acs_cbg <-get_acs(state="TX", geography="block group",year = 2020,
              variables=vars, geometry= F)

cbg_raw <- acs_cbg |>
  select(GEOID, variable, estimate) |>
  pivot_wider(id_cols = GEOID,
              names_from = variable,
              values_from = estimate)
```

```r
cbg <- cbg_raw |>
  mutate(
    QRICH = QRICH / households,
    QESL = (`QESL-Spanish` + `QESL-Other`) / POP,
    QSPANISH = QSPANISH / POP,
    QED12LES = QED12LES / POP,
    QSSBEN = QSSBEN / POP,
    QAGEDEP = (`QAGEDEP-under5-male` + `QAGEDEP-under5-female` +
                 `QAGEDEP-over65`) / POP,
    QFAM = (QFAM) / children,
    QCVLUN = QCVLUN / POP,
    QBLACK = QBLACK / POP,
    QNOAUTO = (`QNOAUTO-owner` + `QNOAUTO-renter`) / households,
    QPOVTY = QPOVTY / POP,
    QFEMALE = QFEMALE / POP
  ) |>
  select(
    -c(
      #Variables to remove
      POP,
      children,
      households,
      `QAGEDEP-under5-male`,
      `QAGEDEP-under5-female`,
      `QAGEDEP-over65`,
      `QESL-Spanish`,
      `QESL-Other`,
      `QNOAUTO-owner`,
      `QNOAUTO-renter`
  )) |>
  mutate(across(2:18,~replace_na(.x,0))) |>
  filter(across(everything(), ~!is.infinite(.)))

head(cbg)
```

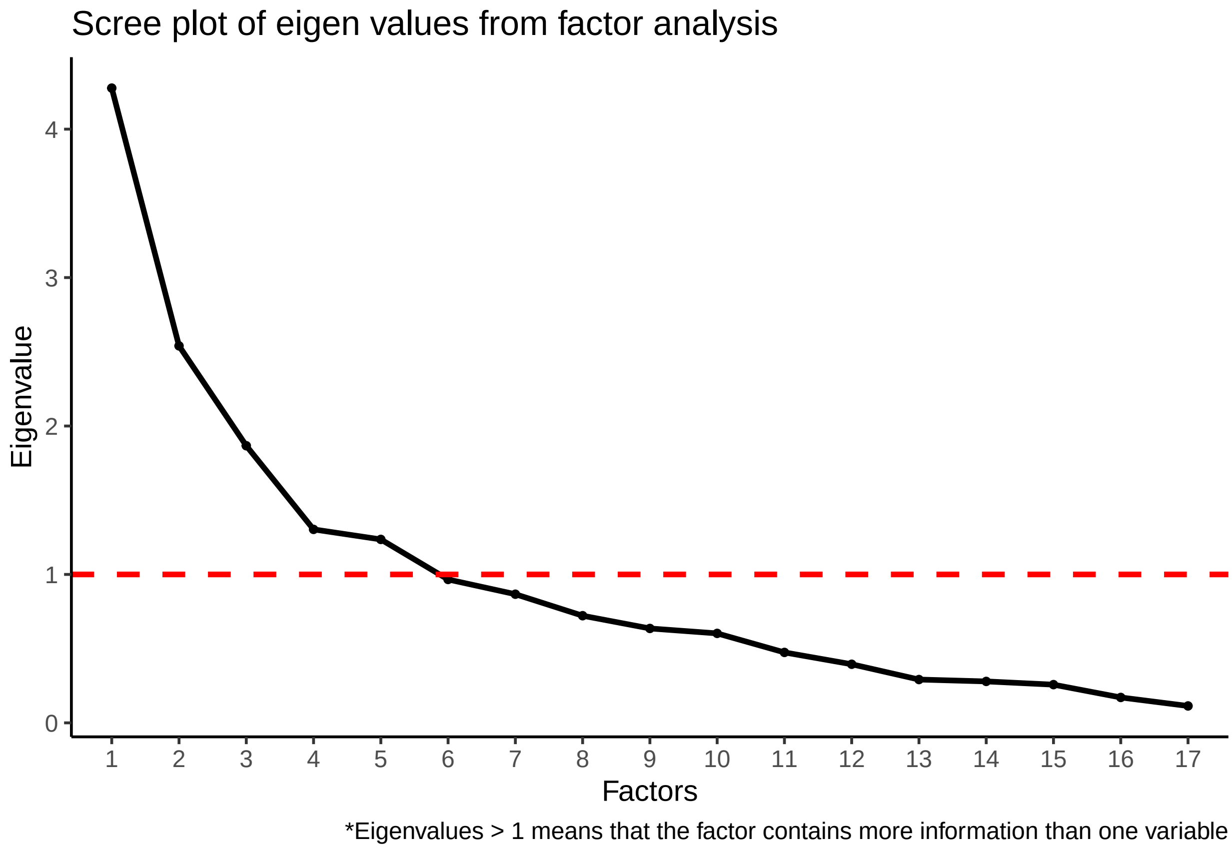
```
## # A tibble: 6 x 18
##   GEOID     QRICH MDHSEVAL MDGRENT PERCAP QSPANISH QED12LES QSSBEN MEDAGE PPUNIT
##   <chr>     <dbl>    <dbl>   <dbl>  <dbl>    <dbl>    <dbl>  <dbl>  <dbl>  <dbl>
## 1 4800500~ 0.0437   216300       0  42184   0.0332   0.0105 0.117    40.8   2.77
## 2 4800395~ 0         57700     855  23052   0.604    0.139  0.124    34.3   2.58
## 3 4800395~ 0         94800    1270  28357   0.499    0.190  0.0668   41.1   2.9
## 4 4800395~ 0.0319   115100       0  24933   0.505    0.127  0.0596   35.1   3.96
## 5 4800500~ 0         60500       0  31135   0        0.0462 0.223    48.7   1.87
## 6 4800500~ 0        239100     941  36568   0.212    0.234  0.129    46.2   2.07
## # ... with 8 more variables: QFAM <dbl>, QCVLUN <dbl>, QBLACK <dbl>,
## #   QPOVTY <dbl>, QFEMALE <dbl>, QESL <dbl>, QAGEDEP <dbl>, QNOAUTO <dbl>
```

```r
minmax <- preProcess(cbg[,-1], method = "range")

transformed <- predict(minmax, cbg[,-1])
```

```r
my_fa <-
  fa(
    r = transformed,
    nfactors = 17,
    rotate = "varimax",
    fm = "minres"
  )

print(my_fa$e.values)
```

```
##  [1] 4.2766708 2.5399075 1.8670550 1.3034875 1.2360075 0.9660814 0.8667149
##  [8] 0.7219387 0.6360499 0.6028129 0.4746236 0.3948893 0.2917618 0.2792771
## [15] 0.2575964 0.1711946 0.1139309
```

```r
e <- as.data.frame(my_fa$e.values) |>
  rename(e_values = `my_fa$e.values`) |>
  rownames_to_column("Factor")

e$Factor <- factor(e$Factor, levels = unique(e$Factor))

theme_set(theme_classic())
scree_plot <- ggplot(data = e, aes(x = Factor , y = e_values, group = 1)) +
  geom_line(size = 1) +
  geom_point(size = 1) +
  geom_hline(yintercept = 1, linetype='dashed', col = 'red', size = 1) +
  labs(
    title = "Scree plot of eigen values from factor analysis",
    y = "Eigenvalue",
    x = "Factors",
    caption = "*Eigenvalues > 1 means that the factor contains more information than one variable"
  )

scree_plot
```



Most important variables in each factor (for cbg): * **Factor 1**: QRICH, PERCAP, MDHSEVAL * **Factor 2**: MEDAGE, QSSBEN, QAGEDEP * **Factor 3**: QSPANISH, QED12LES, QESL * **Factor 4**: PPUNIT * **Factor 5**: QFAM

```
print(my_fa)
```

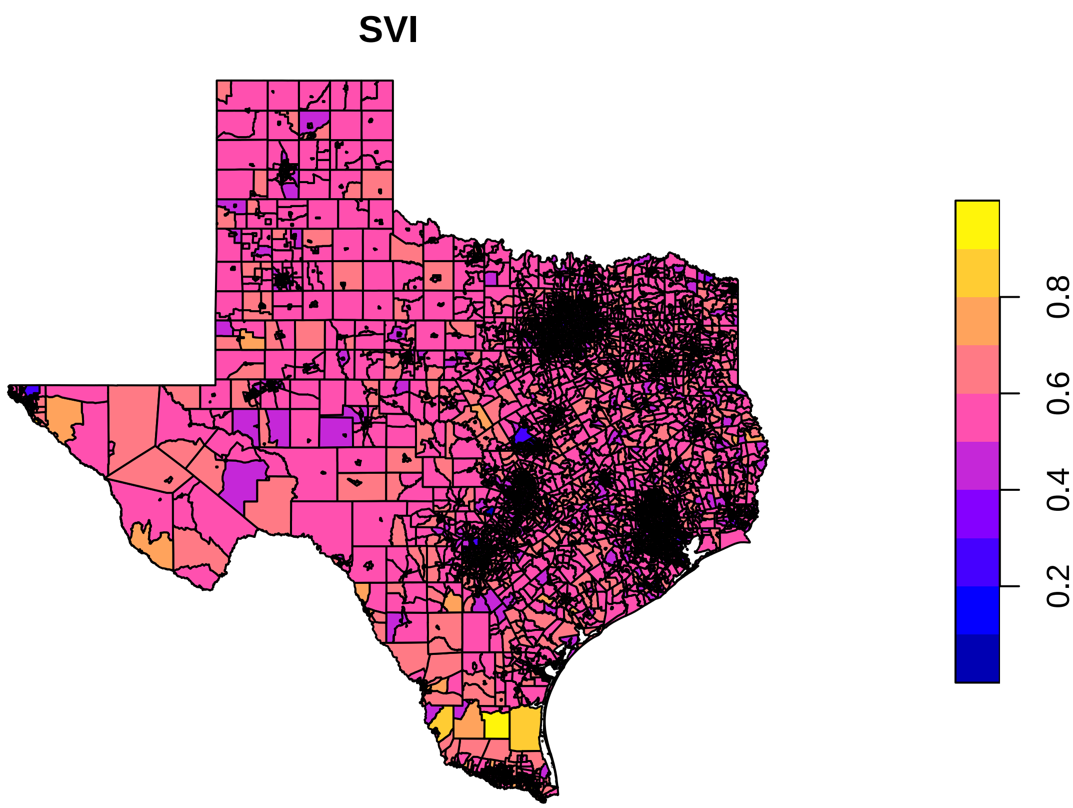
```
## Factor Analysis using method =  minres
## Call: fa(r = transformed, nfactors = 17, rotate = "varimax", fm = "minres")
## Standardized loadings (pattern matrix) based upon correlation matrix
##            MR1   MR2   MR4  MR11   MR8   MR3  MR10   MR5   MR7  MR12   MR6
## QRICH     0.89  0.02 -0.15 -0.02  0.07 -0.04 -0.02 -0.01 -0.06 -0.08  0.02
## MDHSEVAL  0.77  0.04 -0.19  0.06  0.15 -0.08 -0.04  0.02  0.09 -0.07 -0.01
## MDGRENT   0.05 -0.13 -0.05 -0.04  0.00  0.05  0.01  0.09  0.44  0.02  0.00
## PERCAP    0.90  0.11 -0.22 -0.15  0.11 -0.05 -0.09  0.06  0.10 -0.09 -0.10
## QSPANISH -0.28 -0.16  0.70  0.29 -0.14 -0.37  0.02  0.05  0.00  0.10 -0.14
## QED12LES -0.27  0.08  0.76  0.17 -0.04 -0.01  0.08 -0.03 -0.10  0.10  0.04
## QSSBEN   -0.02  0.93 -0.02 -0.13 -0.05 -0.04  0.15  0.04 -0.10 -0.02 -0.05
## MEDAGE    0.26  0.69 -0.04 -0.24  0.18 -0.01 -0.07  0.16 -0.17  0.03 -0.14
## PPUNIT   -0.07 -0.20  0.18  0.82  0.18 -0.04 -0.19  0.21 -0.09  0.09  0.03
## QFAM      0.21  0.04 -0.10  0.14  0.62 -0.14 -0.11  0.04 -0.01 -0.08 -0.03
## QCVLUN   -0.07 -0.04  0.05  0.03 -0.04  0.07  0.04  0.04  0.01  0.34  0.01
## QBLACK   -0.14 -0.08 -0.13 -0.03 -0.17  0.63  0.13  0.09  0.10  0.20  0.00
## QPOVTY   -0.26 -0.06  0.37  0.10 -0.33  0.02  0.30  0.15 -0.04  0.20  0.39
## QFEMALE   0.03  0.19  0.01  0.12  0.02  0.06  0.04  0.53  0.13  0.07  0.02
## QESL     -0.13 -0.03  0.76 -0.07 -0.04 -0.03  0.10  0.02 -0.02  0.01  0.06
## QAGEDEP   0.04  0.83  0.00  0.04  0.01 -0.03  0.02  0.12 -0.08 -0.11  0.10
```

```
## QNOAUTO  -0.09  0.14  0.20 -0.20 -0.15  0.16  0.54  0.05  0.02  0.12  0.04
##           MR14   MR9  MR13  MR15  MR16 MR17   h2    u2 com
## QRICH    -0.02  0.01  0.07 -0.05  0.00    0 0.84 0.155 1.1
## MDHSEVAL  0.02 -0.02 -0.15  0.00  0.00    0 0.70 0.304 1.4
## MDGRENT   0.00  0.00  0.00  0.00  0.00    0 0.23 0.771 1.4
## PERCAP    0.02  0.00  0.08  0.07  0.00    0 0.95 0.049 1.4
## QSPANISH -0.06  0.19  0.03  0.00  0.01    0 0.91 0.092 3.0
## QED12LES  0.14 -0.12  0.08  0.00  0.03    0 0.76 0.235 1.7
## QSSBEN   -0.01 -0.07  0.05 -0.03 -0.03    0 0.93 0.072 1.2
## MEDAGE    0.26 -0.01 -0.02  0.00  0.00    0 0.79 0.213 2.6
## PPUNIT   -0.01  0.00  0.00  0.00  0.00    0 0.89 0.110 1.7
## QFAM      0.01  0.00  0.00  0.00  0.00    0 0.50 0.499 1.7
## QCVLUN    0.00  0.00  0.00  0.00  0.00    0 0.13 0.866 1.4
## QBLACK    0.00  0.01  0.00  0.00  0.00    0 0.55 0.454 1.8
## QPOVTY   -0.02 -0.01  0.00  0.00  0.00    0 0.64 0.364 5.9
## QFEMALE   0.00  0.00  0.00  0.00  0.00    0 0.36 0.641 1.6
## QESL     -0.07  0.02 -0.06  0.00 -0.03    0 0.62 0.378 1.2
## QAGEDEP  -0.08  0.06 -0.04  0.03  0.03    0 0.74 0.257 1.2
## QNOAUTO   0.00  0.00  0.00  0.00  0.00    0 0.46 0.538 2.4
##
##                        MR1  MR2  MR4 MR11  MR8  MR3 MR10  MR5  MR7 MR12  MR6
## SS loadings           2.58 2.19 1.99 0.99 0.68 0.60 0.51 0.42 0.31 0.28 0.23
## Proportion Var        0.15 0.13 0.12 0.06 0.04 0.04 0.03 0.02 0.02 0.02 0.01
## Cumulative Var        0.15 0.28 0.40 0.46 0.50 0.53 0.56 0.59 0.60 0.62 0.63
## Proportion Explained  0.23 0.20 0.18 0.09 0.06 0.05 0.05 0.04 0.03 0.03 0.02
## Cumulative Proportion 0.23 0.43 0.61 0.70 0.77 0.82 0.87 0.91 0.93 0.96 0.98
##                       MR14  MR9 MR13 MR15 MR16 MR17
## SS loadings           0.11 0.06 0.05 0.01 0.00 0.00
## Proportion Var        0.01 0.00 0.00 0.00 0.00 0.00
## Cumulative Var        0.64 0.64 0.65 0.65 0.65 0.65
## Proportion Explained  0.01 0.01 0.00 0.00 0.00 0.00
## Cumulative Proportion 0.99 0.99 1.00 1.00 1.00 1.00
##
## Mean item complexity =  1.9
## Test of the hypothesis that 17 factors are sufficient.
##
## The degrees of freedom for the null model are  136  and the objective function was  7.45 with Chi Sq
## The degrees of freedom for the model are -17  and the objective function was  0
##
## The root mean square of the residuals (RMSR) is  0
## The df corrected root mean square of the residuals is  NA
##
## The harmonic number of observations is  18638 with the empirical chi square  0  with prob <  NA
## The total number of observations was  18638  with Likelihood Chi Square =  0  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.001
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                    MR1  MR2  MR4 MR11   MR8
## Correlation of (regression) scores with factors   0.96 0.95 0.90 0.88  0.70
## Multiple R square of scores with factors          0.92 0.90 0.81 0.77  0.49
## Minimum correlation of possible factor scores     0.84 0.81 0.62 0.53 -0.03
##                                                    MR3  MR10   MR5   MR7  MR12
## Correlation of (regression) scores with factors   0.74  0.62  0.62  0.57  0.47
```

```
## Multiple R square of scores with factors          0.55  0.38  0.39  0.32  0.22
## Minimum correlation of possible factor scores     0.10 -0.24 -0.23 -0.36 -0.57
##                                                     MR6  MR14   MR9  MR13  MR15
## Correlation of (regression) scores with factors    0.62  0.50  0.44  0.42  0.27
## Multiple R square of scores with factors           0.39  0.25  0.19  0.18  0.07
## Minimum correlation of possible factor scores     -0.22 -0.50 -0.61 -0.65 -0.86
##                                                    MR16 MR17
## Correlation of (regression) scores with factors    0.13    0
## Multiple R square of scores with factors           0.02    0
## Minimum correlation of possible factor scores     -0.97   -1
```

```
scores <- as.data.frame(my_fa$scores) |>
  select(MR1, MR2, MR4, MR11, MR8)


minmax <- preProcess(scores, method = "range")

trans_scores <- predict(minmax, scores)

svi <- trans_scores |>
  mutate(MR1 = MR1*-1,
         SVI = MR1 + MR2 + MR4 + MR11 + MR8) |>
  select(SVI)

minmax <- preProcess(svi, method = "range")

svi <- predict(minmax, svi)
```

```
block_group <- acs2 |>
  distinct(GEOID, .keep_all = TRUE) |>
  select(GEOID, geometry)

df2 <- cbg |>
  bind_cols(svi)

library(leaflet)
library(sf)
df2 <- df2 |>
  right_join(block_group, by = "GEOID") |>
  st_as_sf() |>
  filter(!is.na(GEOID))

df2 <- na.omit(df2)

plot(df2["SVI"])
```




```r
df2 <- as.data.frame(df2) |>
  select(GEOID, SVI)

write.csv(df2, "data/processed/svi_cbg.csv")
```