NDI and SVI

Jiayi Zhou, Stephen P. Uong，Jeanette A. Stingone

10/27/2021

## Aim

In this RMD, we have documented the process of constructing an Neighborhood Deprivation Index (NDI) following a developed method. We have also imported the Social Vulnerability Index (SVI) data from the CDC and tested the correlation between NDI and the SVI on the census-tract level in the New York City.

## Setup

We set the working directory to one folder up from the RMarkdown file.

knitr::opts\_knit$set(root.dir = '..')

knitr::opts\_chunk$set(echo = TRUE)  
options(tigris\_class = "sf")

library(devtools)

## Loading required package: usethis

library(ggbiplot)

## Loading required package: ggplot2

## Loading required package: plyr

## Loading required package: scales

## Loading required package: grid

library(stats)  
library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(cluster)  
  
library(tidycensus)  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ tibble 3.1.2 ✓ dplyr 1.0.2  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ forcats 0.5.0  
## ✓ purrr 0.3.4

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::arrange() masks plyr::arrange()  
## x readr::col\_factor() masks scales::col\_factor()  
## x purrr::compact() masks plyr::compact()  
## x dplyr::count() masks plyr::count()  
## x purrr::discard() masks scales::discard()  
## x dplyr::failwith() masks plyr::failwith()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::id() masks plyr::id()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::mutate() masks plyr::mutate()  
## x dplyr::rename() masks plyr::rename()  
## x dplyr::summarise() masks plyr::summarise()  
## x dplyr::summarize() masks plyr::summarize()

library(psych)

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:scales':  
##   
## alpha, rescale

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(tigris)

## To enable   
## caching of data, set `options(tigris\_use\_cache = TRUE)` in your R script or .Rprofile.

library(sf)

## Linking to GEOS 3.8.1, GDAL 3.1.1, PROJ 6.3.1

library(Amelia)

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.7.6, built: 2019-11-24)  
## ## Copyright (C) 2005-2021 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

library(patchwork)  
library(rgdal)

## Loading required package: sp

## rgdal: version: 1.5-18, (SVN revision 1082)  
## Geospatial Data Abstraction Library extensions to R successfully loaded  
## Loaded GDAL runtime: GDAL 3.1.1, released 2020/06/22  
## Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/4.0/Resources/library/rgdal/gdal  
## GDAL binary built with GEOS: TRUE   
## Loaded PROJ runtime: Rel. 6.3.1, February 10th, 2020, [PJ\_VERSION: 631]  
## Path to PROJ shared files: /Library/Frameworks/R.framework/Versions/4.0/Resources/library/rgdal/proj  
## Linking to sp version:1.4-4  
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,  
## use options("rgdal\_show\_exportToProj4\_warnings"="none") before loading rgdal.

library(viridis)

## Loading required package: viridisLite

##   
## Attaching package: 'viridis'

## The following object is masked from 'package:viridisLite':  
##   
## viridis.map

## The following object is masked from 'package:scales':  
##   
## viridis\_pal

library(wesanderson)

## NDI Associated Variables:

Based on a review of literature, 23 census variables that have been used consistently to approximate neighborhood-level environments for possible inclusion in the deprivation index.

## Candidate SEP variables (n = 23)

Source: US Census American Communities Survey (2015-2019) **Education (among adults aged > 25)** % < High School  
% BA or more  
**Employment (among adult labor force, aged 20-64)** % unemployed % males in labor force  
% females in labor force  
**Housing** % renter occupied (among occupied units)  
% vacant housing units (among total housing units)  
% crowded (> 1 occupant per room, among occupied housing units) **Occupation (among full-time, year-round civilian employed population)** % adults in management or professional occupations **Income** % households in poverty (< 200% Federal Poverty Line)  
% Families w/ annual income < $35,000 (2009 inflation-adjusted)  
% female householders with children aged < 18  
% households w/ public assistance income  
% households w/ Food Stamp benefits (in past 12 months)  
Median household income (in the past 12 months)  
% renter or owner housing costs in excess of 30% household income (in past 12 months) **Racial composition** % African American (non-Hispanic) % non-White (calculated as 1 - % non-Hispanic White population) % Hispanic **Residential Stability** % living in the same house one year ago  
% Foreign-born  
% not a U.S. citizen **Language** % speak English less than “very well” (among pop > 5 years old who speak a language other than English at home)

## Check the ACS variables

acs\_19 <- load\_variables(2019, "acs5", cache = TRUE)  
  
#View(acs\_19)

## Write out All the Variables Names that will be needed from the ACS:

edu\_belowHS\_male = sprintf("B15002\_%0.3d",seq(3, 10, by = 1))  
edu\_belowHS\_female = sprintf("B15002\_%0.3d",seq(20, 27, by = 1))  
edu\_BA\_more = c("B15002\_015","B15002\_016","B15002\_017","B15002\_018","B15002\_032","B15002\_033","B15002\_034","B15002\_035")  
edu\_var = c(edu\_belowHS\_female, edu\_belowHS\_male, edu\_BA\_more,"B15002\_002", "B15002\_019")  
  
unempl\_male = sprintf("B23001\_%0.3d",seq(15, 71, by = 7))  
unempl\_female = sprintf("B23001\_%0.3d",seq(101, 157, by = 7))  
labor\_force\_male = sprintf("B23001\_%0.3d",seq(11, 67, by = 7))  
labor\_force\_female = sprintf("B23001\_%0.3d",seq(97, 153, by = 7))  
unempl\_var = c("B23001\_002", "B23001\_088","B23001\_001","B23001\_003","B23001\_089",  
 labor\_force\_female,labor\_force\_male, unempl\_male, unempl\_female)  
  
  
housing\_var = c("B25003\_001","B25002\_001","B25002\_003", "B25003\_003",  
 "B25014\_001", "B25014\_005", "B25014\_006", "B25014\_007",  
 "B25014\_011", "B25014\_012", "B25014\_013", "B25070\_007",  
 "B25070\_008","B25070\_009","B25070\_010","B25070\_011","B25070\_001")  
  
occup\_var = c("C24010\_001", "C24010\_003","C24010\_040")  
  
poverty\_var = c("B17010\_001", "B17010\_002",  
 "B11005\_007","B11005\_010","B11005\_001",  
 "B19057\_001","B19057\_002",  
 "B99221\_002","B99221\_001",  
 "B19013\_001",  
 "B19001\_001", "B19001\_002",  
 "B19001\_003", "B19001\_004",   
 "B19001\_005", "B19001\_006", "B19001\_007")  
  
racial\_var = c("B03002\_004", "B03002\_003", "B03002\_012", "B03002\_001")  
  
stability\_var = c("B07007\_001", "B07007\_006", "B07007\_003", "B07007\_005")  
  
language\_var = c("B16005\_001","B16005\_007", "B16005\_008","B16005\_012","B16005\_013",  
 "B16005\_017", "B16005\_018", "B16005\_022", "B16005\_023",  
 "B16005\_029", "B16005\_030", "B16005\_034", "B16005\_035",  
 "B16005\_039", "B16005\_040", "B16005\_044", "B16005\_045")  
  
c\_var = c(edu\_var, unempl\_var, housing\_var, occup\_var, poverty\_var, racial\_var,stability\_var,language\_var)

## Pulling Census Data

We used an exclusion list to filter out the census tracts that does not pass the inclusion criteria. Of the original 2,167 census tracts in New York City (NYC), we excluded 51 tracts that had a population of less than twenty people and 30 tracts that had a population of at least twenty people but had at least one missing feature in the calculation of the NVI or the Neighborhood Deprivation Index (described later). The majority of the 30 census tracts were non-residential areas, such construction sites, parks, and areas with institutions. As a result, we included 2,086 census tracts in our development of the NVI.

exclusion\_table = read.csv("data/processed/EXCLUSION\_LIST\_20210628.csv")  
  
view(exclusion\_table)

## Data Transformation

Considering the final NDI should be an index with the higher value indicating more deprivation, the value of the candidate variables should show the same pattern. Thus, we reverse coded measures that originally had smaller value associated with deprivation, such as: \* percent population with a college degree or higher \* percent male and female in labor force \* percent population in mangement \* percent non-White population \* percent living in the same house for the past year By doing this reverse coding, we can ensure that they were pointing in the same direction as the other candidate variables.

We also reverse coded and log transformed median household income to ensure normal distribution.

#pulling data  
nyc\_data =   
 get\_acs(geography = "tract", variables =c\_var,  
 state = "NY",   
 county = c('Bronx County', 'Kings County',   
 'New York County', 'Queens County', 'Richmond County'),  
 year = 2019,  
 output = "wide")

## Getting data from the 2015-2019 5-year ACS

total\_left = exclusion\_table %>% filter(flag\_exclude\_FINAL == 0) %>% select(GEOID)  
  
total\_left = as.vector(total\_left$GEOID)  
  
nyc\_acs\_data = nyc\_data %>%  
 filter(GEOID %in% total\_left) %>%   
 mutate(pct\_noHS =   
 (B15002\_003E+B15002\_004E+B15002\_005E+B15002\_006E+B15002\_007E+B15002\_008E+B15002\_009E+B15002\_010E+B15002\_020E+B15002\_021E+B15002\_022E+B15002\_023E+B15002\_024E+B15002\_025E+B15002\_026E+B15002\_027E)/(B15002\_002E+B15002\_019E),  
 pct\_BAmore =   
 1-((B15002\_015E+B15002\_016E+B15002\_017E+B15002\_018E+B15002\_032E+B15002\_033E+B15002\_034E+B15002\_035E)/(B15002\_002E+B15002\_019E)),  
 pct\_unempl =   
 (B23001\_015E+B23001\_022E+B23001\_029E+B23001\_036E+  
 B23001\_043E+B23001\_050E+B23001\_057E+B23001\_064E+  
 B23001\_071E+B23001\_101E+B23001\_108E+B23001\_115E+B23001\_122E+  
 B23001\_129E+B23001\_136E+B23001\_143E+B23001\_150E+B23001\_157E)/  
 (B23001\_001E-B23001\_003E-B23001\_089E),  
 pct\_male\_labor\_force =   
 1-((B23001\_011E+B23001\_018E+B23001\_025E+B23001\_032E+B23001\_039E+B23001\_046E+B23001\_053E+B23001\_060E+B23001\_067E)/(B23001\_002E-B23001\_003E)),  
 pct\_female\_labor\_force =   
 1-((B23001\_097E+B23001\_104E+B23001\_111E+B23001\_118E+B23001\_125E+B23001\_132E+B23001\_139E+B23001\_146E+B23001\_153E)/(B23001\_088E-B23001\_089E)),  
 pct\_rented =   
 B25003\_003E/B25003\_001E,  
 pct\_vacant = B25002\_003E/B25002\_001E,  
 pct\_crowded = (B25014\_005E+B25014\_006E+B25014\_007E+B25014\_011E+B25014\_012E+B25014\_013E)/B25014\_001E,  
 pct\_mgmt = 1-((C24010\_003E+C24010\_040E)/C24010\_001E),  
 pct\_poverty = B17010\_002E/B17010\_001E,  
 pct\_FHH = (B11005\_007E+B11005\_010E)/B11005\_001E,  
 pct\_under35K = (B19001\_002E+B19001\_003E+B19001\_004E+B19001\_005E+B19001\_006E+B19001\_007E)/B19001\_001E,  
 pct\_pubassist = B19057\_002E/B19057\_001E,  
 pct\_foodstamp = B99221\_002E/B99221\_001E,  
 median\_HH\_income = -log(B19013\_001E),  
 pct\_30cost =   
 (B25070\_007E+B25070\_008E+B25070\_009E+B25070\_010E+B25070\_011E)/B25070\_001E,  
 pct\_Black = B03002\_004E/B03002\_001E,  
 pct\_nonWhite = 1-(B03002\_003E/B03002\_001E),  
 pct\_Hispanic = B03002\_012E/B03002\_001E,  
 pct\_E\_notwell = (B16005\_007E+B16005\_008E+B16005\_012E+B16005\_013E+B16005\_017E+B16005\_018E+B16005\_022E+B16005\_023E+B16005\_029E+B16005\_030E+B16005\_034E+B16005\_035E+B16005\_039E+B16005\_040E+B16005\_044E+B16005\_045E)/B16005\_001E,  
 pct\_samehouse = -B07007\_006E/B07007\_001E,  
 pct\_foreignborn = B07007\_003E/B07007\_001E,  
 pct\_notcitizen = B07007\_005E/B07007\_001E)%>%   
 mutate(NAME = gsub(" County, New York", "", NAME)) %>%  
 select(GEOID, NAME,  
 pct\_noHS,pct\_BAmore,  
 pct\_unempl, pct\_male\_labor\_force,pct\_female\_labor\_force,  
 pct\_rented, pct\_vacant,pct\_crowded,  
 pct\_mgmt, pct\_poverty, pct\_under35K, pct\_FHH, pct\_pubassist, pct\_foodstamp,pct\_30cost,  
 median\_HH\_income,  
 pct\_Black,pct\_Hispanic,pct\_nonWhite,pct\_E\_notwell,  
 pct\_samehouse, pct\_foreignborn, pct\_notcitizen)  
  
missmap(nyc\_acs\_data)

## Warning: Unknown or uninitialised column: `arguments`.  
  
## Warning: Unknown or uninitialised column: `arguments`.

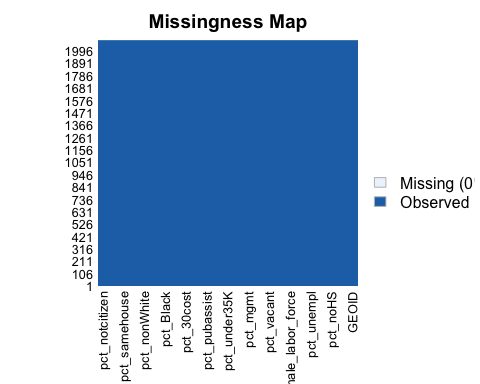
## Warning: Unknown or uninitialised column: `imputations`.

nyc\_acs\_data = nyc\_acs\_data %>% drop\_na()  
  
missmap(nyc\_acs\_data)

## Warning: Unknown or uninitialised column: `arguments`.

## Warning: Unknown or uninitialised column: `arguments`.

## Warning: Unknown or uninitialised column: `imputations`.



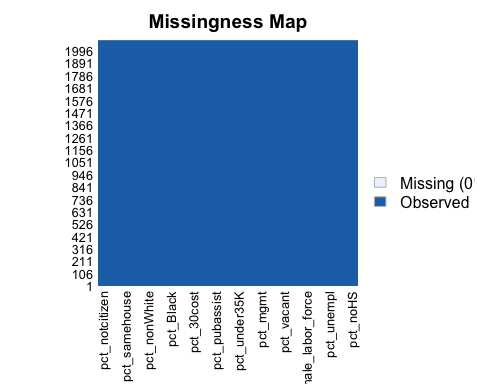
## Step 1: City Wide: Dimensionality Reduction

First city-wide PCA (starts with 24 candidate variables): a. Variable manipulation: Z-standardize the percentages b. Initial component extraction and eigenvalue calculation with all 24 variables included c. Selected the number of components based on eigenvalues > 1 d. Varimax rotate the loading in the selected components (I used psych::principal(…,rotate=“varimax”) in R) e. Inclusion criteria (according to Messer et al.): if a variable loaded above 0.25 in the first component, then keep f. Exclusion criteria (according to Shmool et al.): if a variable loaded strongly (greater than 0.4 or smaller than -0.4) on more than one component, then delete g. Tally all variables that are retained by accessing the initial city-wide PCA solution

values =  
 nyc\_acs\_data %>%   
 select(pct\_noHS,pct\_BAmore,  
 pct\_unempl, pct\_male\_labor\_force,pct\_female\_labor\_force,  
 pct\_rented, pct\_vacant,pct\_crowded,  
 pct\_mgmt, pct\_poverty, pct\_under35K, pct\_FHH, pct\_pubassist, pct\_foodstamp,pct\_30cost,  
 median\_HH\_income,  
 pct\_Black,pct\_Hispanic,pct\_nonWhite,pct\_E\_notwell,  
 pct\_samehouse, pct\_foreignborn, pct\_notcitizen)  
  
missmap(values)

## Warning: Unknown or uninitialised column: `arguments`.  
  
## Warning: Unknown or uninitialised column: `arguments`.

## Warning: Unknown or uninitialised column: `imputations`.



#values[values == 0] <- NA

# compute variance of each variable  
# will see standarization is needed  
apply(values, 2, var, na.rm = TRUE)

## pct\_noHS pct\_BAmore pct\_unempl   
## 0.0124689839 0.0423459510 0.0005708777   
## pct\_male\_labor\_force pct\_female\_labor\_force pct\_rented   
## 0.0100878240 0.0094271791 0.0644051824   
## pct\_vacant pct\_crowded pct\_mgmt   
## 0.0038407928 0.0054021429 0.0190837267   
## pct\_poverty pct\_under35K pct\_FHH   
## 0.0131407729 0.0233332561 0.0061293516   
## pct\_pubassist pct\_foodstamp pct\_30cost   
## 0.0018832185 0.0004750119 0.0170706887   
## median\_HH\_income pct\_Black pct\_Hispanic   
## 0.2488333248 0.0804108599 0.0498139656   
## pct\_nonWhite pct\_E\_notwell pct\_samehouse   
## 0.0842399560 0.0114500387 0.0038626993   
## pct\_foreignborn pct\_notcitizen   
## 0.0229380830 0.0083207011

# create new data frame with centered variables  
scaled\_df = apply(values, 2, scale)  
#scaled\_df[is.na(scaled\_df)] = 0 #assign 0 to NA values,.  
head(scaled\_df)

## pct\_noHS pct\_BAmore pct\_unempl pct\_male\_labor\_force  
## [1,] 1.0782972 0.8985422 0.5281842 -0.7419128  
## [2,] 0.5402499 1.0544374 -0.0109769 -0.2287875  
## [3,] 0.6353121 0.6240864 -0.9241071 0.1148664  
## [4,] 2.2085674 1.4786889 -0.9503552 -1.5266710  
## [5,] 2.1979945 1.3951581 -0.6350518 -0.9833780  
## [6,] 3.1713313 1.5321195 -0.8059469 -0.7709334  
## pct\_female\_labor\_force pct\_rented pct\_vacant pct\_crowded pct\_mgmt  
## [1,] 0.8630653 0.10273291 2.5277316 2.0805667 1.16877692  
## [2,] -0.6821145 -0.06607182 3.7627546 0.8201019 0.26335038  
## [3,] 1.3870467 -1.49004026 3.7797425 0.0659844 -0.01645666  
## [4,] -0.9888253 0.29529758 0.8697970 2.9782024 1.55823598  
## [5,] -0.9949257 0.48844883 -0.9638729 4.3984021 1.42186280  
## [6,] 0.1223694 0.22029157 -1.0242716 2.8985270 1.39666889  
## pct\_poverty pct\_under35K pct\_FHH pct\_pubassist pct\_foodstamp  
## [1,] -0.3099203 0.311129695 0.35284337 0.46050588 1.775816478  
## [2,] 0.9018043 -0.483375544 0.51403097 -0.05731521 1.575642463  
## [3,] -0.6722409 -0.613949995 -0.46494171 -0.11383015 2.800822040  
## [4,] -0.6050456 -0.663519264 0.45477130 -0.61471608 0.345356964  
## [5,] -0.2580814 -0.317747703 0.23690375 -0.84770477 1.028870642  
## [6,] 0.7728494 0.003072059 0.09330268 0.32722513 -0.002174215  
## pct\_30cost median\_HH\_income pct\_Black pct\_Hispanic pct\_nonWhite  
## [1,] 0.72995305 0.5737791 -0.2200154069 1.685171 0.9933440  
## [2,] 1.34248534 0.1768510 -0.2919456945 2.047861 0.9773274  
## [3,] 0.07223564 -0.1316767 0.0006013436 1.214134 1.0566920  
## [4,] -0.16868258 -0.2558755 -0.6295762099 2.819178 1.0895477  
## [5,] -0.55566123 -0.2855869 -0.6761476063 2.683354 1.0740447  
## [6,] 1.39741498 0.2747882 -0.8156423908 2.266756 1.0543694  
## pct\_E\_notwell pct\_samehouse pct\_foreignborn pct\_notcitizen  
## [1,] 1.1680790 -0.34692609 1.4970210 1.6952300  
## [2,] 1.1431636 -1.07494597 0.9092686 1.6253348  
## [3,] -0.1696911 -0.67051959 0.8032700 0.0822103  
## [4,] 2.6030645 -0.67467909 1.4375217 2.9973695  
## [5,] 2.6075658 -0.07961257 1.8924671 3.4135972  
## [6,] 2.6239092 -0.27303997 1.7076625 3.6178131

# Calculate eigenvalues & eigenvectors  
ndi\_var.cov = cov(scaled\_df)  
ndi\_var.eigen = eigen(ndi\_var.cov) #PC1-5  
str(ndi\_var.eigen)

## List of 2  
## $ values : num [1:23] 8.48 2.99 2.28 1.96 1.06 ...  
## $ vectors: num [1:23, 1:23] -0.295 -0.294 -0.171 -0.075 -0.106 ...  
## - attr(\*, "class")= chr "eigen"

pca\_firstcitywide\_rotated <- psych::principal(scaled\_df, rotate="varimax", nfactors=5, scores=TRUE)  
print(pca\_firstcitywide\_rotated$loadings[,1:5])

## RC1 RC2 RC4 RC3  
## pct\_noHS 0.61527462 0.566948698 0.169841978 0.248296461  
## pct\_BAmore 0.39698999 0.408096651 0.578594365 0.436887560  
## pct\_unempl 0.53171410 -0.027728048 0.391431508 -0.137653957  
## pct\_male\_labor\_force 0.19137393 -0.314902706 0.032263282 0.736745517  
## pct\_female\_labor\_force 0.11638628 0.091848798 -0.171990798 0.846381979  
## pct\_rented 0.79058819 0.203223761 -0.114246884 -0.330677996  
## pct\_vacant -0.11436198 -0.089424603 -0.299169219 -0.143838659  
## pct\_crowded 0.31501510 0.680383548 0.021387783 0.004907243  
## pct\_mgmt 0.33306427 0.427121797 0.631791081 0.375081609  
## pct\_poverty 0.86949117 0.141926373 0.105081241 0.247138641  
## pct\_under35K 0.86960827 0.133898606 0.125763522 0.330936555  
## pct\_FHH 0.61647565 -0.007187623 0.621223378 0.051709766  
## pct\_pubassist 0.73835676 -0.035376720 0.281006109 0.160333319  
## pct\_foodstamp 0.04685448 0.126264843 0.218159586 0.131223251  
## pct\_30cost 0.21056643 0.346392601 0.237465674 0.455299825  
## median\_HH\_income 0.79779282 0.244699015 0.257295104 0.372109945  
## pct\_Black 0.07317824 -0.294548192 0.849932648 -0.030132368  
## pct\_Hispanic 0.60119599 0.424952652 0.083817267 -0.069471060  
## pct\_nonWhite 0.34539632 0.358163171 0.762503137 -0.022905420  
## pct\_E\_notwell 0.28273597 0.835243778 -0.173693433 0.196254083  
## pct\_samehouse 0.12598784 -0.119121002 -0.430285308 -0.580619342  
## pct\_foreignborn -0.19750215 0.831839526 0.214079024 0.072786428  
## pct\_notcitizen 0.12342721 0.893039623 0.007714563 -0.187011853  
## RC5  
## pct\_noHS -0.134454860  
## pct\_BAmore -0.129329461  
## pct\_unempl -0.119233956  
## pct\_male\_labor\_force 0.097417891  
## pct\_female\_labor\_force 0.003081202  
## pct\_rented 0.133706424  
## pct\_vacant 0.653074257  
## pct\_crowded 0.095338653  
## pct\_mgmt -0.131917260  
## pct\_poverty 0.021897476  
## pct\_under35K 0.043552766  
## pct\_FHH -0.076010800  
## pct\_pubassist -0.023621163  
## pct\_foodstamp 0.659642767  
## pct\_30cost 0.091233422  
## median\_HH\_income 0.006000058  
## pct\_Black 0.143870814  
## pct\_Hispanic -0.212066216  
## pct\_nonWhite -0.011484074  
## pct\_E\_notwell -0.102250419  
## pct\_samehouse 0.211044483  
## pct\_foreignborn 0.052332959  
## pct\_notcitizen 0.041593836

**Retained in the initial city-wide** pct\_unempl, pct\_rented,pct\_crowded, pct\_poverty,pct\_under35K,pct\_pubassist,median\_HH\_income, pct\_nonWhite,pct\_E\_notwell

## Step 2: Stratified PCA (5 boroughs)

Borough-stratified PCAs (starts with 24 candidate variables): a. Repeat steps b-c described in the first city-wide PCA process b. Inclusion criteria: i. Based on Messer et al.: If a variable not only loaded above 0.25 in at least one borough but also never loaded below 0.16 in any borough in the first component, then keep for the next inclusion assessment ii. Based on Shmool et al.: If a variable passed previous inclusion criteria (> 0.25 and never <0.16 in the first component), then they are included in this second selection step-> If a variable loaded greater than 0.4 or smaller than -0.4 in any component in two or more borough-level PCA solution, then keep for second city-wide PCA process \*\* c. Exclusion criteria: Same as before. If a variable loaded strongly (greater than 0.4 or smaller than -0.4) on more than one component, then delete

#### Bronx County

values\_bronx=  
 nyc\_acs\_data %>%  
 filter(str\_detect(GEOID, "^36005" )) %>%   
 select(pct\_noHS,pct\_BAmore,  
pct\_unempl,pct\_male\_labor\_force, pct\_female\_labor\_force,   
pct\_rented,pct\_vacant,pct\_crowded,  
pct\_mgmt,  
pct\_poverty,pct\_under35K,pct\_FHH,pct\_pubassist,pct\_foodstamp,pct\_30cost,median\_HH\_income,  
pct\_Black, pct\_Hispanic,pct\_nonWhite,pct\_E\_notwell,  
pct\_samehouse,pct\_foreignborn, pct\_notcitizen)  
  
#values\_bronx[values\_bronx == 0] <- NA

# compute variance of each variable  
# will see standarization is needed  
apply(values\_bronx, 2, var, na.rm = TRUE)

# create new data frame with centered variables  
scaled\_df\_b = apply(values\_bronx, 2, scale)  
#scaled\_df\_b[is.na(scaled\_df\_b)] = 0 #assign 0 to NA values, no variance? not sure if ok.  
head(scaled\_df\_b)

# Calculate eigenvalues & eigenvectors  
ndi\_var\_b.cov = cov(scaled\_df\_b)  
ndi\_var\_b.eigen = eigen(ndi\_var\_b.cov) #PC5  
str(ndi\_var\_b.eigen)

pca\_bronx\_rotated <- psych::principal(scaled\_df\_b, rotate="varimax", nfactors=5, scores=TRUE)  
print(pca\_bronx\_rotated$loadings[,1:5])

**Retain: Bronx** pct\_noHS,pct\_BAmore, pct\_rented, pct\_mgmt, pct\_poverty,pct\_under35K,pct\_FHH,pct\_pubassist,pct\_30cost,median\_HH\_income, pct\_notcitizen

#### Kings County

values\_kings=  
 nyc\_acs\_data %>%  
 filter(str\_detect(GEOID, "^36047" )) %>%   
 select(pct\_noHS,pct\_BAmore,  
pct\_unempl,pct\_male\_labor\_force, pct\_female\_labor\_force,   
pct\_rented,pct\_vacant,pct\_crowded,  
pct\_mgmt,  
pct\_poverty,pct\_under35K,pct\_FHH,pct\_pubassist,pct\_foodstamp,pct\_30cost,median\_HH\_income,  
pct\_Black, pct\_Hispanic,pct\_nonWhite,pct\_E\_notwell,  
pct\_samehouse,pct\_foreignborn, pct\_notcitizen)  
   
#values\_kings[values\_kings == 0] <- NA

# compute variance of each variable  
# will see standarization is needed  
apply(values\_kings, 2, var, na.rm = TRUE)

# create new data frame with centered variables  
scaled\_df\_k = apply(values\_kings, 2, scale)  
#scaled\_df\_k[is.na(scaled\_df\_k)] = 0 #assign 0 to NA values, no variance? not sure if ok.  
head(scaled\_df\_k)

# Calculate eigenvalues & eigenvectors  
ndi\_var\_k.cov = cov(scaled\_df\_k)  
ndi\_var\_k.eigen = eigen(ndi\_var\_k.cov) #PC5  
str(ndi\_var\_k.eigen)

pca\_kings\_rotated <- psych::principal(scaled\_df\_k, rotate="varimax", nfactors=5, scores=TRUE)  
print(pca\_kings\_rotated$loadings[,1:5])

**Retain: Kings** pct\_unempl, pct\_rented, pct\_poverty,pct\_pubassist, pct\_Hispanic

#### New York County

values\_NY=  
 nyc\_acs\_data %>%  
 filter(str\_detect(GEOID, "^36061" )) %>%   
 select(pct\_noHS,pct\_BAmore,  
pct\_unempl,pct\_male\_labor\_force, pct\_female\_labor\_force,   
pct\_rented,pct\_vacant,pct\_crowded,  
pct\_mgmt,  
pct\_poverty,pct\_under35K,pct\_FHH,pct\_pubassist,pct\_foodstamp,pct\_30cost,median\_HH\_income,  
pct\_Black, pct\_Hispanic,pct\_nonWhite,pct\_E\_notwell,  
pct\_samehouse,pct\_foreignborn, pct\_notcitizen)  
  
 missmap(values\_NY)  
#values\_NY[values\_NY == 0] <- NA

# compute variance of each variable  
# will see standarization is needed  
apply(values\_NY, 2, var, na.rm = TRUE)

# create new data frame with centered variables  
scaled\_df\_ny = apply(values\_NY, 2, scale)  
#scaled\_df\_ny[is.na(scaled\_df\_ny)] = 0 #assign 0 to NA values, no variance? not sure if ok.  
head(scaled\_df\_ny)

# Calculate eigenvalues & eigenvectors  
ndi\_var\_ny.cov = cov(scaled\_df\_ny)  
ndi\_var\_ny.eigen = eigen(ndi\_var\_ny.cov) #PC4  
str(ndi\_var\_ny.eigen)

pca\_ny\_rotated <- psych::principal(scaled\_df\_ny, rotate="varimax", nfactors=4, scores=TRUE)  
print(pca\_ny\_rotated$loadings[,1:4])

**Retain: New York** pct\_BAmore, pct\_unempl, pct\_crowded, pct\_mgmt, pct\_poverty,pct\_FHH,pct\_pubassist,median\_HH\_income, pct\_Black,pct\_nonWhite

#### Queens County

values\_queens=  
 nyc\_acs\_data %>%  
 filter(str\_detect(GEOID, "^36081" )) %>%   
 select(pct\_noHS,pct\_BAmore,  
pct\_unempl,pct\_male\_labor\_force, pct\_female\_labor\_force,   
pct\_rented,pct\_vacant,pct\_crowded,  
pct\_mgmt,  
pct\_poverty,pct\_under35K,pct\_FHH,pct\_pubassist,pct\_foodstamp,pct\_30cost,median\_HH\_income,  
pct\_Black, pct\_Hispanic,pct\_nonWhite,pct\_E\_notwell,  
pct\_samehouse,pct\_foreignborn, pct\_notcitizen)  
  
 missmap(values\_queens)  
#values\_queens[values\_queens == 0] <- NA

# compute variance of each variable  
# will see standarization is needed  
apply(values\_queens, 2, var, na.rm = TRUE)

# create new data frame with centered variables  
scaled\_df\_q = apply(values\_queens, 2, scale)  
#scaled\_df\_q[is.na(scaled\_df\_q)] = 0 #assign 0 to NA values, no variance? not sure if ok.  
head(scaled\_df\_q)

# Calculate eigenvalues & eigenvectors  
ndi\_var\_q.cov = cov(scaled\_df\_q)  
ndi\_var\_q.eigen = eigen(ndi\_var\_q.cov) #PC6  
str(ndi\_var\_q.eigen)

pca\_queens\_rotated <- psych::principal(scaled\_df\_q, rotate="varimax", nfactors=6, scores=TRUE)  
print(pca\_queens\_rotated$loadings[,1:6])

**Retain: Queens** pct\_noHS, pct\_crowded, pct\_poverty,median\_HH\_income, pct\_Hispanic,pct\_E\_notwell, pct\_foreignborn, pct\_notcitizen

#### Richmond County

values\_Richmond=  
 nyc\_acs\_data %>%  
 filter(str\_detect(GEOID, "^36085" )) %>%   
 select(pct\_noHS,pct\_BAmore,  
pct\_unempl,pct\_male\_labor\_force, pct\_female\_labor\_force,   
pct\_rented,pct\_vacant,pct\_crowded,  
pct\_mgmt,  
pct\_poverty,pct\_under35K,pct\_FHH,pct\_pubassist,pct\_foodstamp,pct\_30cost,median\_HH\_income,  
pct\_Black, pct\_Hispanic,pct\_nonWhite,pct\_E\_notwell,  
pct\_samehouse,pct\_foreignborn, pct\_notcitizen)  
#values\_Richmond[values\_Richmond == 0] <- NA

# compute variance of each variable  
# will see standarization is needed  
apply(values\_Richmond, 2, var, na.rm = TRUE)

# create new data frame with centered variables  
scaled\_df\_r = apply(values\_Richmond, 2, scale)  
#scaled\_df\_r[is.na(scaled\_df\_r)] = 0 #assign 0 to NA values, no variance? not sure if ok.  
head(scaled\_df\_r)

# Calculate eigenvalues & eigenvectors  
ndi\_var\_r.cov = cov(scaled\_df\_r)  
ndi\_var\_r.eigen = eigen(ndi\_var\_r.cov) #PC6  
str(ndi\_var\_r.eigen)

pca\_richmond\_rotated <- psych::principal(scaled\_df\_r, rotate="varimax", nfactors=6, scores=TRUE)  
print(pca\_richmond\_rotated$loadings[,1:6])

**Retain: Richmond** pct\_male\_labor\_force, pct\_rented,pct\_vacant,pct\_crowded, pct\_poverty,pct\_under35K,pct\_FHH,pct\_pubassist,median\_HH\_income, pct\_Black,pct\_E\_notwell, pct\_notcitizen

## Summary:

I have tallied all variables that retained from each step here. To see a more detailed table, please refer to the supplemental Table 1 of the paper.

**Retained in the initial city-wide** (only > 0.4 or < -0.4 in the first component, and cannot be > 0.4 or < -0.4 in more then one component):

pct\_unempl, pct\_rented,pct\_crowded, pct\_poverty,pct\_under35K,pct\_pubassist,median\_HH\_income, pct\_nonWhite,pct\_E\_notwell

\*After doing the stratified PAC, we have the following variables loaded strongly borough-level PCA solutions: **Bronx** pct\_noHS,pct\_BAmore, pct\_rented, pct\_mgmt, pct\_poverty,pct\_under35K,pct\_FHH,pct\_pubassist,pct\_30cost,median\_HH\_income, pct\_notcitizen

**Kings** pct\_unempl, pct\_rented, pct\_poverty,pct\_pubassist, pct\_Hispanic

**New York** pct\_BAmore, pct\_unempl, pct\_crowded, pct\_mgmt, pct\_poverty,pct\_FHH,pct\_pubassist,median\_HH\_income, pct\_Black,pct\_nonWhite

**Queens** pct\_noHS, pct\_crowded, pct\_poverty,median\_HH\_income, pct\_Hispanic,pct\_E\_notwell, pct\_foreignborn, pct\_notcitizen

**Richmond** pct\_male\_labor\_force, pct\_rented,pct\_vacant,pct\_crowded, pct\_poverty,pct\_under35K,pct\_FHH,pct\_pubassist,median\_HH\_income, pct\_Black,pct\_E\_notwell, pct\_notcitizen

**more then 2 boroughs**: pct\_noHS,pct\_BAmore, pct\_unempl, pct\_rented,pct\_crowded, pct\_mgmt, pct\_poverty,pct\_under35K,pct\_FHH,pct\_pubassist,median\_HH\_income, pct\_Hispanic, pct\_Black, pct\_E\_notwell pct\_notcitizen

**less then 0.16:** bronx:(pct\_Black) kings:(pct\_notcitizen) queens:(pct\_unempl,pct\_FHH,pct\_pubassist)

## Included in the second city-wide:(only > 0.4 or < -0.4 in one component)

*from stratified* pct\_noHS,pct\_BAmore, pct\_rented,pct\_crowded, pct\_mgmt, pct\_poverty,pct\_under35K,median\_HH\_income, pct\_Hispanic, pct\_E\_notwell + *from first city-wide* pct\_unempl, pct\_rented,pct\_crowded, pct\_poverty,pct\_under35K,pct\_pubassist,median\_HH\_income, pct\_nonWhite,pct\_E\_notwell

**NEW TOTAL** pct\_noHS,pct\_BAmore, pct\_unempl， pct\_rented,pct\_crowded, pct\_mgmt, pct\_poverty,pct\_under35K,pct\_pubassist, median\_HH\_income, pct\_nonWhite,pct\_Hispanic, pct\_E\_notwell

## Second City-wide:

Second city-wide PCA (starts with 12 variables selected in the initial city-wide PCA process and the borough-level PCA process) : a. Repeat steps b-f described in the first city-wide PCA process (used the same inclusion and exclusion criteria as the first city wide PAC since cannot find description on what Messer or Shmool did):

values\_s =  
 nyc\_acs\_data %>%   
 select(pct\_noHS,pct\_BAmore,  
pct\_unempl,  
pct\_rented,pct\_crowded,  
pct\_mgmt,  
pct\_poverty,pct\_under35K,pct\_pubassist, median\_HH\_income,  
pct\_nonWhite,pct\_Hispanic, pct\_E\_notwell)  
  
missmap(values\_s)  
#values[values == 0] <- NA(0% are still meaningful)

# compute variance of each variable  
# will see standarization is needed  
apply(values\_s, 2, var, na.rm = TRUE)

# create new data frame with centered variables  
scaled\_df\_s = apply(values\_s, 2, scale)  
#scaled\_df[is.na(scaled\_df)] = 0 #assign 0 to NA values,.  
head(scaled\_df\_s)

# Calculate eigenvalues & eigenvectors  
ndi\_var\_s.cov = cov(scaled\_df\_s)  
ndi\_var\_s.eigen = eigen(ndi\_var\_s.cov) #PC3  
str(ndi\_var\_s.eigen)

pca\_secondcitywide\_rotated <- psych::principal(scaled\_df\_s, rotate="varimax", nfactors=3, scores=TRUE)  
print(pca\_secondcitywide\_rotated$loadings[,1:3])

**after omition:** pct\_BAmore, pct\_unempl， pct\_mgmt, pct\_poverty,pct\_under35K,pct\_pubassist, pct\_nonWhite

The final 7 features of deprivation were used to re-ran the final PCA and create the final NDI.

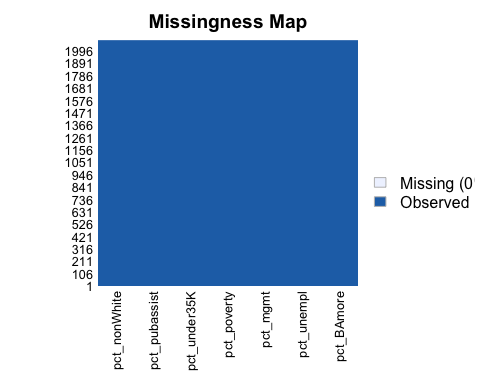
## Step 3: Final City-Wide with retained 7 vars

Final Index construction (starts with 7 variables: % BA or more,% unemployed, % adults in management or professional occupations ,% households in poverty (< 200% Federal Poverty Line),% Families w/ annual income < $35,000 (2019 inflation-adjusted),% households w/ public assistance income,% non-White) a. Re-run the PCA process with the 7 variables b. Extract the un-rotated loading in the first component as the NDI

# create new data frame with centered variables  
NDI\_df = nyc\_acs\_data %>%   
 select(pct\_BAmore,  
pct\_unempl,  
pct\_mgmt,  
pct\_poverty,pct\_under35K,pct\_pubassist,  
pct\_nonWhite)  
  
missmap(NDI\_df)

## Warning: Unknown or uninitialised column: `arguments`.  
  
## Warning: Unknown or uninitialised column: `arguments`.

## Warning: Unknown or uninitialised column: `imputations`.



scaled\_ndi\_df = apply(NDI\_df, 2, scale)  
#scaled\_ndi\_df[is.nan(scaled\_ndi\_df)] = 0 #assign 0 to NA values, no variance? not sure if ok.  
head(scaled\_ndi\_df)

## pct\_BAmore pct\_unempl pct\_mgmt pct\_poverty pct\_under35K pct\_pubassist  
## [1,] 0.8985422 0.5281842 1.16877692 -0.3099203 0.311129695 0.46050588  
## [2,] 1.0544374 -0.0109769 0.26335038 0.9018043 -0.483375544 -0.05731521  
## [3,] 0.6240864 -0.9241071 -0.01645666 -0.6722409 -0.613949995 -0.11383015  
## [4,] 1.4786889 -0.9503552 1.55823598 -0.6050456 -0.663519264 -0.61471608  
## [5,] 1.3951581 -0.6350518 1.42186280 -0.2580814 -0.317747703 -0.84770477  
## [6,] 1.5321195 -0.8059469 1.39666889 0.7728494 0.003072059 0.32722513  
## pct\_nonWhite  
## [1,] 0.9933440  
## [2,] 0.9773274  
## [3,] 1.0566920  
## [4,] 1.0895477  
## [5,] 1.0740447  
## [6,] 1.0543694

summary(NDI\_df)

## pct\_BAmore pct\_unempl pct\_mgmt pct\_poverty   
## Min. :0.0301 Min. :0.00000 Min. :0.2459 Min. :0.00000   
## 1st Qu.:0.5469 1st Qu.:0.02100 1st Qu.:0.6781 1st Qu.:0.05105   
## Median :0.6926 Median :0.03211 Median :0.7802 Median :0.10912   
## Mean :0.6419 Mean :0.03761 Mean :0.7489 Mean :0.13956   
## 3rd Qu.:0.7925 3rd Qu.:0.04861 3rd Qu.:0.8503 3rd Qu.:0.20113   
## Max. :0.9761 Max. :0.18698 Max. :0.9895 Max. :0.63501   
## pct\_under35K pct\_pubassist pct\_nonWhite   
## Min. :0.0000 Min. :0.00000 Min. :0.01082   
## 1st Qu.:0.1917 1st Qu.:0.01354 1st Qu.:0.42183   
## Median :0.2706 Median :0.03177 Median :0.76696   
## Mean :0.3052 Mean :0.04474 Mean :0.67907   
## 3rd Qu.:0.3931 3rd Qu.:0.06269 3rd Qu.:0.95805   
## Max. :0.8773 Max. :0.37303 Max. :1.00000

summary(scaled\_ndi\_df)

## pct\_BAmore pct\_unempl pct\_mgmt pct\_poverty   
## Min. :-2.9733 Min. :-1.5743 Min. :-3.6409 Min. :-1.2175   
## 1st Qu.:-0.4620 1st Qu.:-0.6953 1st Qu.:-0.5126 1st Qu.:-0.7721   
## Median : 0.2460 Median :-0.2304 Median : 0.2270 Median :-0.2656   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.7318 3rd Qu.: 0.4603 3rd Qu.: 0.7344 3rd Qu.: 0.5371   
## Max. : 1.6240 Max. : 6.2515 Max. : 1.7420 Max. : 4.3220   
## pct\_under35K pct\_pubassist pct\_nonWhite   
## Min. :-1.9982 Min. :-1.0310 Min. :-2.3024   
## 1st Qu.:-0.7432 1st Qu.:-0.7191 1st Qu.:-0.8863   
## Median :-0.2268 Median :-0.2988 Median : 0.3028   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.5753 3rd Qu.: 0.4137 3rd Qu.: 0.9612   
## Max. : 3.7452 Max. : 7.5649 Max. : 1.1057

# Calculate eigenvalues & eigenvectors  
ndi.cov = cov(scaled\_ndi\_df)  
ndi.eigen = eigen(ndi.cov)  
str(ndi.eigen)

## List of 2  
## $ values : num [1:7] 4.288 1.063 0.719 0.393 0.331 ...  
## $ vectors: num [1:7, 1:7] -0.412 -0.292 -0.396 -0.395 -0.406 ...  
## - attr(\*, "class")= chr "eigen"

PVE = ndi.eigen$values / sum(ndi.eigen$values) #61%  
PVE

## [1] 0.61252847 0.15186888 0.10269296 0.05614863 0.04734530 0.01810608 0.01130968

PVE\_initial = ndi\_var.eigen$values /sum(ndi\_var.eigen$values) #53%  
PVE\_initial

## [1] 0.368750047 0.129801111 0.099253341 0.085164857 0.046290210 0.041052666  
## [7] 0.032962997 0.030570184 0.028222988 0.024660729 0.018484302 0.017177512  
## [13] 0.014173725 0.013351496 0.010937798 0.010215778 0.007009239 0.005865544  
## [19] 0.005122984 0.003899766 0.002814627 0.002560552 0.001657548

pca\_ndi\_rotated = psych::principal(scaled\_ndi\_df, rotate="none", nfactors=1, scores=TRUE)  
NDI\_score = pca\_ndi\_rotated$scores[,1]  
  
NDI\_score = as.matrix(NDI\_score)

summary(NDI\_score)

## V1   
## Min. :-2.15673   
## 1st Qu.:-0.66705   
## Median :-0.01473   
## Mean : 0.00000   
## 3rd Qu.: 0.57845   
## Max. : 3.61134

NDI\_2086=  
 data.frame(GEOID = nyc\_acs\_data[,1], NAME = nyc\_acs\_data[,2], NDI\_score) %>%  
 separate(NAME,   
 into = c("Tract", "County"),   
 sep = ",") %>%   
 mutate(County = str\_trim(County), FIPS = GEOID) %>%   
 select(GEOID, FIPS, Tract, County, NDI\_score)  
  
NDI\_2086$FIPS = substr(NDI\_2086$FIPS, 0, 5)  
  
head(NDI\_2086)

## GEOID FIPS Tract County NDI\_score  
## 1 36081036100 36081 Census Tract 361 Queens 0.73369822  
## 2 36081036300 36081 Census Tract 363 Queens 0.49541853  
## 3 36081037100 36081 Census Tract 371 Queens -0.09517738  
## 4 36081037700 36081 Census Tract 377 Queens 0.29135733  
## 5 36081037900 36081 Census Tract 379 Queens 0.38270354  
## 6 36081041300 36081 Census Tract 413 Queens 0.84794727

## save the data output

#write.csv(NDI\_2086, "data/raw/NDI.NYC.2086t.csv")

## Scale the NDI:

Here we transformed the continues NDI into quartiles. Having NDI as a categorical variable will produce a map that have better color contract. Moreover, since we are aiming to compare our NDI to CDC’s SVI, it is necessary to transform these two indices on to the same scale.

NDI\_with\_scaled\_score = NDI\_2086 %>%   
 dplyr::mutate(NDI\_scaled= dplyr::ntile(NDI\_score, 4))

Check summary statistics

summary(NDI\_with\_scaled\_score)

## GEOID FIPS Tract County   
## Length:2086 Length:2086 Length:2086 Length:2086   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## NDI\_score NDI\_scaled   
## Min. :-2.15673 Min. :1.000   
## 1st Qu.:-0.66705 1st Qu.:1.250   
## Median :-0.01473 Median :2.000   
## Mean : 0.00000 Mean :2.499   
## 3rd Qu.: 0.57845 3rd Qu.:3.000   
## Max. : 3.61134 Max. :4.000

## Visualization: Check the NDI distribution

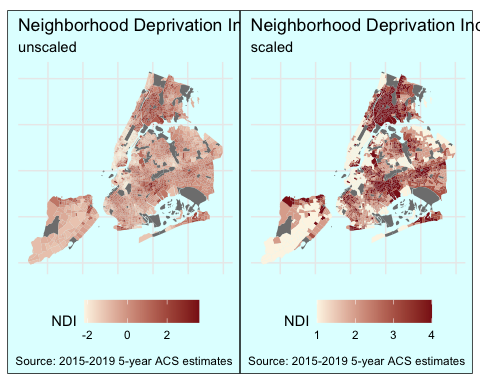
In order to check the validity of the NDI that we have calculated, we mapped the NDI scores across NYC. The maps produced here were not included in our manuscript. These maps were made for the quality control purposes.

library(nycgeo)

map\_ndi\_with\_score = nyc\_boundaries(geography = "tract") %>%   
 left\_join(NDI\_2086, by = c("geoid" = "GEOID")) %>%   
 ggplot() +  
 geom\_sf(aes(fill = NDI\_score),color = NA) +  
 theme\_minimal() +   
 theme(axis.text = element\_blank(),legend.position = "bottom") +  
 theme(plot.background = element\_rect(fill = "lightcyan")) +  
 scale\_fill\_gradient(low = "OldLace", high = "firebrick4")+  
 labs(fill = "NDI",  
 title = "Neighborhood Deprivation Index",  
 subtitle = "unscaled",  
 caption = "Source: 2015-2019 5-year ACS estimates")

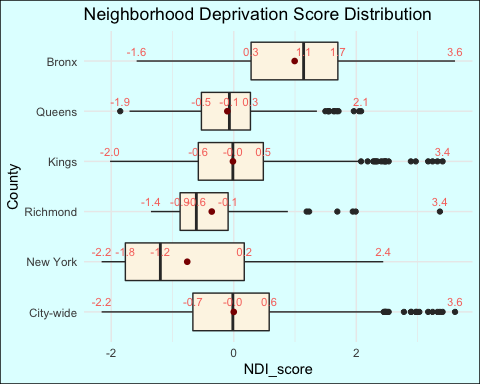
map\_ndi\_with\_scaled\_score = nyc\_boundaries(geography = "tract") %>%   
 left\_join(NDI\_with\_scaled\_score, by = c("geoid" = "GEOID")) %>%   
 ggplot() +  
 geom\_sf(aes(fill = NDI\_scaled),color = NA) +  
 theme\_minimal() +   
 theme(axis.text = element\_blank(),legend.position = "bottom") +  
 theme(plot.background = element\_rect(fill = "lightcyan")) +  
 scale\_fill\_gradient(low = "OldLace", high = "firebrick4")+  
 labs(fill = "NDI",  
 title = "Neighborhood Deprivation Index",  
 subtitle = "scaled",  
 caption = "Source: 2015-2019 5-year ACS estimates")

map\_ndi\_with\_score + map\_ndi\_with\_scaled\_score



#### Barplot

City\_score = NDI\_2086 %>%   
 select(NDI\_score) %>%   
 mutate(County = "City-wide")  
  
Stratified = NDI\_2086 %>%   
 select(NDI\_score, County)  
  
  
NDI\_box = rbind(City\_score,Stratified)  
  
NDI\_box$County <- factor(NDI\_box$County, levels=c("City-wide","New York", "Richmond","Kings", "Queens","Bronx"))  
  
  
  
ggplot(NDI\_box, aes(x = County, y = NDI\_score))+  
 geom\_boxplot(fill = "OldLace") +  
 stat\_summary(  
 aes(label=sprintf("%1.1f", ..y..), color = "red"),  
 geom="text",   
 fun = quantile,  
 #fun = function(y) boxplot.stats(y)$stats,  
 position=position\_nudge(x=0.2),   
 size=3.0, show.legend=FALSE)+  
 stat\_summary(fun=mean, colour="darkred", geom="point",   
 shape=16, size=2, show.legend=FALSE)+  
 coord\_flip() +  
 theme\_minimal()+  
 theme(plot.background = element\_rect(fill = "lightcyan")) +  
 labs(title = "Neighborhood Deprivation Score Distribution")



##SVI Import the CDC SVI data file. The variable “RPL\_THEMES” is the SVI.

SDI\_df = read\_csv("data/raw/svi.csv") %>%   
 mutate(FIPS = as.character(FIPS)) %>%   
 filter(FIPS %in% NDI\_2086$GEOID) %>%   
 select(FIPS, LOCATION, COUNTY, RPL\_THEMES) %>%   
 rename(GEOID = FIPS) %>%   
 filter(RPL\_THEMES != -999.0000)

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## STATE = col\_character(),  
## ST\_ABBR = col\_character(),  
## COUNTY = col\_character(),  
## LOCATION = col\_character()  
## )

## See spec(...) for full column specifications.

1. Scale the SVI in the same way as scaling the NDI. This scaled score will be used for future mapping
2. Conducted a spearman correlation test testing the association between the continouse NDI and SVI.

SDI = scale(SDI\_df$RPL\_THEMES)  
  
summary(SDI)

## V1   
## Min. :-2.3883   
## 1st Qu.:-0.7615   
## Median : 0.1601   
## Mean : 0.0000   
## 3rd Qu.: 0.8513   
## Max. : 1.4278

SDI\_df =  
 data.frame(GEOID = SDI\_df[,1], SDI)  
  
SDI\_scaled = SDI\_df %>%   
 dplyr::mutate(SDI\_s = dplyr::ntile(SDI, 4))  
  
summary(SDI\_df)

## GEOID SDI   
## Length:2086 Min. :-2.3883   
## Class :character 1st Qu.:-0.7615   
## Mode :character Median : 0.1601   
## Mean : 0.0000   
## 3rd Qu.: 0.8513   
## Max. : 1.4278

NDI\_c = join(NDI\_2086, SDI\_df, by = "GEOID") %>%   
 pull(NDI\_score) %>% as.numeric()  
  
SDI\_c = join(NDI\_2086, SDI\_df, by = "GEOID") %>%   
 pull(SDI) %>% as.numeric()  
  
cor.test(NDI\_c, SDI\_c, alternative = "two.sided", method = "spearman", conf.level = 0.95)

## Warning in cor.test.default(NDI\_c, SDI\_c, alternative = "two.sided", method =  
## "spearman", : Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: NDI\_c and SDI\_c  
## S = 203714718, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.8653424

The correlation coefficient between NDI and SVI is 0.86. They are correlated.

## Visualization: Check the SVI distribution

map\_sdi = nyc\_boundaries(geography = "tract") %>%   
 left\_join(SDI\_df, by = c("geoid" = "GEOID")) %>%   
 ggplot() +  
 geom\_sf(aes(fill = SDI),color = NA) +  
 theme\_minimal() +   
 theme(axis.text = element\_blank(),legend.position = "bottom") +  
 scico::scale\_fill\_scico(palette = "bilbao") +  
 theme(plot.background = element\_rect(fill = "lightcyan")) +  
 scale\_fill\_gradient(low = "White", high = "firebrick4")+  
 labs(fill = "SDI",  
 title = "Social Vulnerability Index",  
 subtitle = "unscaled",  
 caption = "Source: 2018 CDC/ATSDR SVI Data")

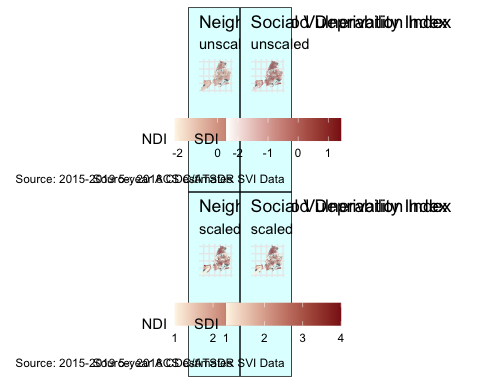
## Scale for 'fill' is already present. Adding another scale for 'fill', which  
## will replace the existing scale.

map\_scaled\_sdi = nyc\_boundaries(geography = "tract") %>%   
 left\_join(SDI\_scaled, by = c("geoid" = "GEOID")) %>%   
 ggplot() +  
 geom\_sf(aes(fill = SDI\_s),color = NA, ) +  
 theme\_minimal() +   
 theme(axis.text = element\_blank(),legend.position = "bottom") +  
 theme(plot.background = element\_rect(fill = "lightcyan")) +  
 scico::scale\_fill\_scico(palette = "bilbao") +  
 scale\_fill\_gradient(low = "OldLace", high = "firebrick4")+  
 labs(fill = "SDI",  
 title = "Social Vulnerability Index",  
 subtitle = "scaled",  
 caption = "Source: 2018 CDC/ATSDR SVI Data")

## Scale for 'fill' is already present. Adding another scale for 'fill', which  
## will replace the existing scale.

## NDI vs SVI Maps

(map\_ndi\_with\_score + map\_sdi)/(map\_ndi\_with\_scaled\_score + map\_scaled\_sdi)



## Result analysis:First Component Loadings

In this section, we will check the county-specific and city-wide feature loadings on the first principal component.

c\_city\_wide = pca\_ndi\_rotated$loadings[,1]  
  
#pca\_ndi\_income$loadings[,1]

#### Bronx：First Component Loading

Bronx=  
 nyc\_acs\_data %>%  
 filter(str\_detect(GEOID, "^36005" )) %>%   
 select(pct\_BAmore,pct\_unempl,pct\_mgmt,pct\_poverty,pct\_under35K,pct\_pubassist,pct\_nonWhite)

pca\_Bronx\_rotated <- psych::principal(Bronx, rotate="none", nfactors=1, scores=TRUE)  
c\_bronx = pca\_Bronx\_rotated$loadings[,1]  
print(c\_bronx)

## pct\_BAmore pct\_unempl pct\_mgmt pct\_poverty pct\_under35K   
## 0.8912247 0.6230822 0.8497903 0.8880392 0.8936478   
## pct\_pubassist pct\_nonWhite   
## 0.7693168 0.7925676

#### Kings

Kings=  
 nyc\_acs\_data %>%  
 filter(str\_detect(GEOID, "^36047" )) %>%   
 select(pct\_BAmore,pct\_unempl,pct\_mgmt,pct\_poverty,pct\_under35K,pct\_pubassist,pct\_nonWhite)

pca\_kings\_rotated <- psych::principal(Kings, rotate="none", nfactors=1, scores=TRUE)  
c\_kings = pca\_kings\_rotated$loadings[,1]  
print(c\_kings)

## pct\_BAmore pct\_unempl pct\_mgmt pct\_poverty pct\_under35K   
## 0.8635977 0.5271729 0.8145213 0.7906005 0.8594067   
## pct\_pubassist pct\_nonWhite   
## 0.7414879 0.6265817

#### NY

NY=  
 nyc\_acs\_data %>%  
 filter(str\_detect(GEOID, "^36061" )) %>%   
 select(pct\_BAmore,pct\_unempl,pct\_mgmt,pct\_poverty,pct\_under35K,pct\_pubassist,pct\_nonWhite)

pca\_ny\_rotated <- psych::principal(NY, rotate="none", nfactors=1, scores=TRUE)  
c\_ny = pca\_ny\_rotated$loadings[,1]  
print(c\_ny)

## pct\_BAmore pct\_unempl pct\_mgmt pct\_poverty pct\_under35K   
## 0.9678196 0.5928540 0.9205011 0.9136955 0.9455561   
## pct\_pubassist pct\_nonWhite   
## 0.8447637 0.9364677

#### Queens

Queens=  
 nyc\_acs\_data %>%  
 filter(str\_detect(GEOID, "^36081" )) %>%   
 select(pct\_BAmore,pct\_unempl,pct\_mgmt,pct\_poverty,pct\_under35K,pct\_pubassist,pct\_nonWhite)

pca\_queens\_rotated <- psych::principal(Queens, rotate="none", nfactors=1, scores=TRUE)  
c\_queens = pca\_queens\_rotated$loadings[,1]  
print(c\_queens)

## pct\_BAmore pct\_unempl pct\_mgmt pct\_poverty pct\_under35K   
## 0.8427162 0.4396688 0.8222197 0.6476706 0.5762765   
## pct\_pubassist pct\_nonWhite   
## 0.6032032 0.7472671

#### Richmond

Richmond=  
 nyc\_acs\_data %>%  
 filter(str\_detect(GEOID, "^36085" )) %>%   
 select(pct\_BAmore,pct\_unempl,pct\_mgmt,pct\_poverty,pct\_under35K,pct\_pubassist,pct\_nonWhite)  
#values\_Richmond[values\_Richmond == 0] <- NA

pca\_richmond\_rotated <- psych::principal(Richmond, rotate="none", nfactors=1, scores=TRUE)  
c\_richmond = pca\_richmond\_rotated$loadings[,1]  
print(c\_richmond)

## pct\_BAmore pct\_unempl pct\_mgmt pct\_poverty pct\_under35K   
## 0.7674728 0.2346507 0.7086721 0.9257778 0.9043481   
## pct\_pubassist pct\_nonWhite   
## 0.8953824 0.8581273

r\_prop\_var = c(67.4,55.8,79.1,48.9,64.2,61.4)

## Loading comparison

library(knitr)

df = data.frame(c\_bronx,c\_kings,c\_ny,c\_queens,c\_richmond, c\_city\_wide) %>% rbind(r\_prop\_var)  
new\_df = as.data.frame(lapply(df,round, 3))

df1 = round(max(df[1,]) - min(df[1,]), digits = 3)  
df2 = round(max(df[2,]) - min(df[2,]), digits = 3)  
df3 = round(max(df[3,]) - min(df[3,]), digits = 3)  
df4 = round(max(df[4,]) - min(df[4,]), digits = 3)  
df5 = round(max(df[5,]) - min(df[5,]), digits = 3)  
df6 = round(max(df[6,]) - min(df[6,]), digits = 3)  
df7 = round(max(df[7,]) - min(df[7,]), digits = 3)  
  
  
difference = c(df1,df2,df3,df4,df5,df6,df7," ")  
  
loading\_comparison = new\_df %>% cbind(difference)  
row.names(loading\_comparison) = c("% B.S./B.A. or higher","% Unempolyment", "% Management Occupation",  
 "% Households in Poverty","% Households with annual income < $35,000",   
 "% Households receving public assistance",  
 "% Non-Hispanic non-White", "% Variance Explained")  
colnames(loading\_comparison) = c("Bronx","Kings","New York", "Queens", "Richmond","City-wide Index","Loading Difference")  
  
loading\_comparison %>% kable(caption = "Supplemental Table 3. Comparison of Borough-specific and City-wide first principal component deprivation score loadings")

Supplemental Table 3. Comparison of Borough-specific and City-wide first principal component deprivation score loadings

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Bronx | Kings | New York | Queens | Richmond | City-wide Index | Loading Difference |
| % B.S./B.A. or higher | 0.891 | 0.864 | 0.968 | 0.843 | 0.767 | 0.853 | 0.2 |
| % Unempolyment | 0.623 | 0.527 | 0.593 | 0.440 | 0.235 | 0.606 | 0.388 |
| % Management Occupation | 0.850 | 0.815 | 0.921 | 0.822 | 0.709 | 0.820 | 0.212 |
| % Households in Poverty | 0.888 | 0.791 | 0.914 | 0.648 | 0.926 | 0.819 | 0.278 |
| % Households with annual income < $35,000 | 0.894 | 0.859 | 0.946 | 0.576 | 0.904 | 0.840 | 0.369 |
| % Households receving public assistance | 0.769 | 0.741 | 0.845 | 0.603 | 0.895 | 0.769 | 0.292 |
| % Non-Hispanic non-White | 0.793 | 0.627 | 0.936 | 0.747 | 0.858 | 0.744 | 0.31 |
| % Variance Explained | 67.400 | 55.800 | 79.100 | 48.900 | 64.200 | 61.400 |  |

This result is save as the Supplemental table 3 in the manualscript.