1 ArcGIS Pro Steps

The ArcGIS Pro steps could be described as follows:

```
Unify projection and units.
// Gun Shot
1. Add data: DC gunshot shapefiles
2. Run optimized hot-cold spot analysis
// Street Lights
1. Add data: DC streetlight shapefiles
2. Run optimized hot-cold spot analysis
// Ward 7 & 8 Hot Cold Spot Analysis
1. Add data: DC gunshot shapefiles
2. Add Ward shapefiles
3. Cut gunshot shapefiles in wards 7 & 8
2. Run hot-cold spot analysis in wards 7 & 8
// Visualizing park types
1. Add data: DC park and recreation shapefiles
2. Symbology: choose unique features
```

2 QGIS Steps

The QGIS steps could be described as follows:

```
Unify projection and units.
// Creat a grid vector layer
processing.run("native:creategrid", {'TYPE':2,'EXTENT
    ↪ ':'-77.119500000,-76.909500000,38.792500000,38.995500000 [EPSG:4326]','HSPACING':0.001,'
    ↪ VSPACING':0.001,'HOVERLAY':0,'VOVERLAY':0,'CRS':QgsCoordinateReferenceSystem('EPSG
    ↪ :4326'),'OUTPUT':'TEMPORARY_OUTPUT'})

Process population data
1. Add raster layer //add raster layer
2. Warp (Reproject) //align projection with the grid
processing.run("gdal:warpreproject", {'INPUT':'/Volumes/study/uw/23win/green space/process/
    ↪ population.tif','SOURCE_CRS':None,'TARGET_CRS':None,'RESAMPLING':5,'NODATA':None,'
    ↪ TARGET_RESOLUTION':0.001,'OPTIONS':'','DATA_TYPE':0,'TARGET_EXTENT
    ↪ ':'-77.119500000,-76.909500000,38.792500000,38.995500000 [EPSG:4326]','TARGET_EXTENT_CRS
    ↪ ':None,'MULTITHREADING':False,'EXTRA':'','OUTPUT':'TEMPORARY_OUTPUT'})

Process nighttime light data
1. Add raster layer //add raster layer
2. r.resample.interp //resample to finer grid
processing.run("grass7:r.resamp.interp", {'input':'/Volumes/study/uw/23win/green space/process/
    ↪ ward_raster.tif','method':0,'output':'TEMPORARY_OUTPUT','GRASS_REGION_PARAMETER
    ↪ ':'-77.119500000,-76.909500000,38.792500000,38.995500000 [EPSG:4326]','
    ↪ GRASS_REGION_CELLSIZE_PARAMETER':0.001,'GRASS_RASTER_FORMAT_OPT':'','
    ↪ GRASS_RASTER_FORMAT_META':''})

Calculate distance to park (distance to police is similar)
1. add vector layer //add vector layers of parks and waterways
2. polygons to lines //get frontlines of waterways and parks, separately for large parks and
    ↪ small parks
processing.run("native:polygonstolines", {'INPUT':'memory://memory?geometry=MultiPolygon&crs=
    ↪ EPSG:4326&field=osm_id:string(12,0)&field=code:integer(4,0)&field=fclass:string(28,0)&
    ↪ field=name:string(100,0)&uid={4787160c-eda5-467a-849a-cee15da6dc2f}','OUTPUT':'
    ↪ TEMPORARY_OUTPUT'})
```

```
3. Difference //remove water front lines from park front lines
processing.run("native:difference", {'INPUT':'memory://memory?geometry=MultiLineString&crs=EPSG
    ↪ :4326&field=osm_id:string(12,0)&field=code:integer(4,0)&field=fclass:string(28,0)&field=
    ↪ name:string(100,0)&field=WARD:integer(1,0)&field=layer:string(254,0)&field=path:string
    ↪ (254,0)&field=area:double(23,15)&field=small:integer(10,0)&uid={034fc0d2-0c62-4804-aa7a
    ↪ -77524288994a}','OVERLAY':'memory://memory?geometry=MultiLineString&crs=EPSG:4326&field=
    ↪ osm_id:string(12,0)&field=code:integer(4,0)&field=fclass:string(28,0)&field=name:string
    ↪ (100,0)&field=area:double(0,0)&uid={e94e5847-05c5-4fe6-a32e-76a0aadcd8a2}','OUTPUT':'
    ↪ TEMPORARY_OUTPUT','GRID_SIZE':None})
4. rasterize //change frontline vectors to rasters
processing.run("gdal:rasterize", {'INPUT':'memory://memory?geometry=MultiLineString&crs=EPSG
    ↪ :4326&field=osm_id:string(12,0)&field=code:integer(4,0)&field=fclass:string(28,0)&field=
    ↪ name:string(100,0)&field=WARD:integer(1,0)&field=layer:string(254,0)&field=path:string
    ↪ (254,0)&field=area:double(23,15)&uid={d1659823-9d75-4f7c-934f-d5ff025295c0}','FIELD
    ↪ ':'','BURN':1,'USE_Z':False,'UNITS':1,'WIDTH':0.001,'HEIGHT':0.001,'EXTENT':NONNE,'
    ↪ NODATA':0,'OPTIONS':'','DATA_TYPE':5,'INIT':None,'INVERT':False,'EXTRA':'','OUTPUT':'
    ↪ TEMPORARY_OUTPUT'})
5. r.grow.distance //generate distance raster
processing.run("grass7:r.grow.distance", {'input':'/private/var/folders/0z/
    ↪ ckyvszw94zl_wxs0cnljmg3c0000gn/T/processing_Cgnqzk/107204b50e404dd0803a6598c63c510f/
    ↪ OUTPUT.tif','metric':4,'-m':True,'-':False,'distance':'TEMPORARY_OUTPUT','value':'
    ↪ TEMPORARY_OUTPUT','GRASS_REGION_PARAMETER
    ↪ ':'-77.119500000,-76.909500000,38.792500000,38.995500000 [EPSG:4326]','
    ↪ GRASS_REGION_CELLSIZE_PARAMETER':0.001,'GRASS_RASTER_FORMAT_OPT':'','
    ↪ GRASS_RASTER_FORMAT_META':''})

Buffering area for parks
1. Fixed percentage buffer //for large and small parks
processing.run("bufferbypercentage:fixedpercentagebuffer", {'INPUT':'memory://memory?geometry=
    ↪ MultiPolygon&crs=EPSG:4326&field=osm_id:string(12,0)&field=code:integer(4,0)&field=
    ↪ fclass:string(28,0)&field=name:string(100,0)&uid={c7054a1b-f08f-4956-ba54-a2ddfd1940da
    ↪ }','PERCENTAGE':300,'SEGMENTS':5,'OUTPUT':'TEMPORARY_OUTPUT'})

Process all point data (streetlight and gunshot)
1. add vector layer //add vector layers
2. Count points in polygon
/processing.run("native:countpointsinpolygon", {'POLYGONS':'memory://memory?geometry=
    ↪ MultiPolygon&crs=EPSG:4326&field=osm_id:string(12,0)&field=code:integer(4,0)&field=
    ↪ fclass:string(28,0)&field=name:string(100,0)&field=WARD:integer(1,0)&field=layer:string
    ↪ (254,0)&field=path:string(254,0)&field=area:double(23,15)&field=small%20park:double(0,0)
    ↪ &field=pi%20rate:double(0,0)&field=ntlmean:double(0,0)&field=street_light:double(0,0)&
    ↪ field=gun_shot_num:double(0,0)&uid={d2fb193f-990f-4b12-9010-0c9e325ee2a9}','POINTS':'/
    ↪ Volumes/study/uw/23win/green space/process/processed_shp/gunshot_21_22_clipped.shp','
    ↪ WEIGHT':'','CLASSFIELD':'','FIELD':'NUMPOINTS','OUTPUT':'TEMPORARY_OUTPUT'})

Process other vector data
1. add vector layer //add vector layers
2. rasterize //change vectors to rasters
processing.run("gdal:rasterize", {'INPUT':'memory://memory?geometry=MultiLineString&crs=EPSG
    ↪ :4326&field=osm_id:string(12,0)&field=code:integer(4,0)&field=fclass:string(28,0)&field=
    ↪ name:string(100,0)&field=WARD:integer(1,0)&field=layer:string(254,0)&field=path:string
    ↪ (254,0)&field=area:double(23,15)&uid={d1659823-9d75-4f7c-934f-d5ff025295c0}','FIELD
    ↪ ':'','BURN':1,'USE_Z':False,'UNITS':1,'WIDTH':0.001,'HEIGHT':0.001,'EXTENT':NONNE,'
    ↪ NODATA':0,'OPTIONS':'','DATA_TYPE':5,'INIT':None,'INVERT':False,'EXTRA':'','OUTPUT':'
    ↪ TEMPORARY_OUTPUT'})
```

```
add rasters to the grid layer
1. Zonal statistics //add to the grid shapefile
qgis_process run native:zonalstatisticsfb --distance_units=mi --area_units=ac --ellipsoid=EPSG
    ↪ :7019 --INPUT='memory?geometry=MultiPolygon&crs=EPSG:4326&field=osm_id:string(12,0)&
    ↪ field=code:integer(4,0)&field=fclass:string(28,0)&field=name:string(100,0)&field=area:
    ↪ double(0,0)&uid={2c5c55f1-6c65-4b70-b4ed-910dc57dfab6}' --INPUT_RASTER='/Volumes/study/
    ↪ uw/23win/green space/process/ward_raster.tif' --RASTER_BAND=1 --COLUMN_PREFIX=ward --
    ↪ STATISTICS=9 --OUTPUT=TEMPORARY_OUTPUT
```

References

Optimized Hot Spot Analysis (Spatial Statistics). https://pro.arcgis.com/en/pro-app/3.0/tool-reference/spatial-statistics/optimized-hot-spot-analysis.htm

R code

March 6, 2023

Grid-level analysis

This is the R code for grid-level analysis, the shapefile is built by the previous QGIS section.

```
library(sp)
library(ggplot2)
library(rgdal)
library(spdep)
library(spaMM)
library(bigmemory)
library(RSpectra)

# load data
dc <- rgdal::readOGR(dsn = "/Volumes/study/uw/23win/green space/shp", layer = "update")

#drop area where population=0, and area out of DC
subset <- dc[!is.na(dc@data$wardmajori),]
subset <- subset(subset, population>0)

#replace null value with 0
subset@data$streetligh[is.na(subset@data$streetligh)] <- 0
subset@data$gunshotmaj[is.na(subset@data$gunshotmaj)] <- 0
subset@data$parkmajori[is.na(subset@data$parkmajori)] <- -1
subset@data$parkmajori <- subset@data$parkmajori+1
subset@data$buffer_sma[is.na(subset@data$buffer_sma)] <- 0
subset@data$buffer_lar[is.na(subset@data$buffer_lar)] <- 0

#continuity neighbors
#deal with unlinked polygons

adj = nb2mat(nb, style="B",zero.policy=F)
#transform adj to big matrix for edits
x <- as.big.matrix(adj)
options(bigmemory.allow.dimnames=TRUE)
rownames(x) <- NULL
adj_matrix <- as.matrix(x)

#regression

# check spatial autocorrelation robustness
poisson_d_total <- fitme(gunshotmaj ~ streetligh + ntlmajorit + dtopolicem + dtolargema +
    ↪  dtosmallma + adjacency(1|id) +offset(log(population)), adjMatrix = adj_matrix,
    ↪ data = try@data, family = 'poisson', control.HLfit=list(algebra="spprec"))
```

```
summary(poisson_d_total)

# distance to park as explanatory variable
poisson_d_total <- glm(gunshotmaj ~ streetligh + ntlmajorit + dtopolicem + dtolargema +
    ↪ dtosmallma + factor(wardmajori) + offset(log(population)), data = subset@data,
    ↪ family = 'poisson')
summary(poisson_d_total)
poisson_d_black <- glm(gunshotmaj ~ streetligh + ntlmajorit + dtopolicem + dtolargema +
    ↪ dtosmallma + factor(wardmajori) + offset(log(population)), data = black@data,
    ↪ family = 'poisson')
summary(poisson_d_black)
poisson_d_white <- glm(gunshotmaj ~ streetligh + ntlmajorit + dtopolicem + dtolargema +
    ↪ dtosmallma + factor(wardmajori) + offset(log(population)), data = black@data,
    ↪ family = 'poisson')
summary(poisson_d_white)

# buffer as explanatory variable
poisson_b_total <- glm(gunshotmaj ~ streetligh + ntlmajorit + dtopolicem + buffer_lar +
    ↪ buffer_sma + factor(wardmajori) + offset(log(population)), data = subset@data,
    ↪ family = 'poisson')
summary(poisson_d_total)
poisson_b_black <- glm(gunshotmaj ~ streetligh + ntlmajorit + dtopolicem + buffer_lar +
    ↪ buffer_sma + factor(wardmajori) + offset(log(population)), data = black@data,
    ↪ family = 'poisson')
summary(poisson_d_black)
poisson_b_white <- glm(gunshotmaj ~ streetligh + ntlmajorit + dtopolicem + buffer_lar +
    ↪ buffer_sma + factor(wardmajori) + offset(log(population)), data = black@data,
    ↪ family = 'poisson')
summary(poisson_d_white)

# buffer with interaction term with streetlight
#large park
poisson_inte_total_large <- glm(gunshotmaj ~ ntlmajorit + dtopolicem + factor(buffer_lar)
    ↪ *streetligh + factor(wardmajori) + offset(log(population)), data = subset@data,
    ↪ family = 'poisson')
summary(poisson_d_total)
poisson_inte_black_large <- glm(gunshotmaj ~ ntlmajorit + dtopolicem + factor(buffer_lar)
    ↪ *streetligh + factor(wardmajori) + offset(log(population)), data = black@data,
    ↪ family = 'poisson')
summary(poisson_d_black)
poisson_inte_white_large <- glm(gunshotmaj ~ ntlmajorit + dtopolicem + factor(buffer_lar)
    ↪ *streetligh + factor(wardmajori) + offset(log(population)), data = black@data,
    ↪ family = 'poisson')
summary(poisson_d_white)

#small park
poisson_inte_total_large <- glm(gunshotmaj ~ ntlmajorit + dtopolicem + factor(buffer_sma)
    ↪ *streetligh + factor(wardmajori) + offset(log(population)), data = subset@data,
    ↪ family = 'poisson')
summary(poisson_d_total)
poisson_inte_black_large <- glm(gunshotmaj ~ ntlmajorit + dtopolicem + factor(buffer_sma)
    ↪ *streetligh + factor(wardmajori) + offset(log(population)), data = black@data,
    ↪ family = 'poisson')
summary(poisson_d_black)
```

```
poisson_inte_white_large <- glm(gunshotmaj ~ ntlmajorit + dtopolicem + factor(buffer_sma)
    ↪ *streetligh + factor(wardmajori) + offset(log(population)), data = black@data,
    ↪ family = 'poisson')
summary(poisson_d_white)
```

Demographic characters in DC

```
library(knitr)
opts_chunk$set(message = FALSE)
```

Read "black" data for year 2013 and year 2021.

```
black1 <- get_acs(
  geography = "block group",
  state = "DC",
  variables = "B02001_003",
  year = 2013,
  geometry = TRUE
)
```

```
##   |                                                                              |
```

```
black2 <- get_acs(
  geography = "block group",
  state = "DC",
  variables = "B02001_003",
  year = 2021,
  geometry = TRUE
)
```

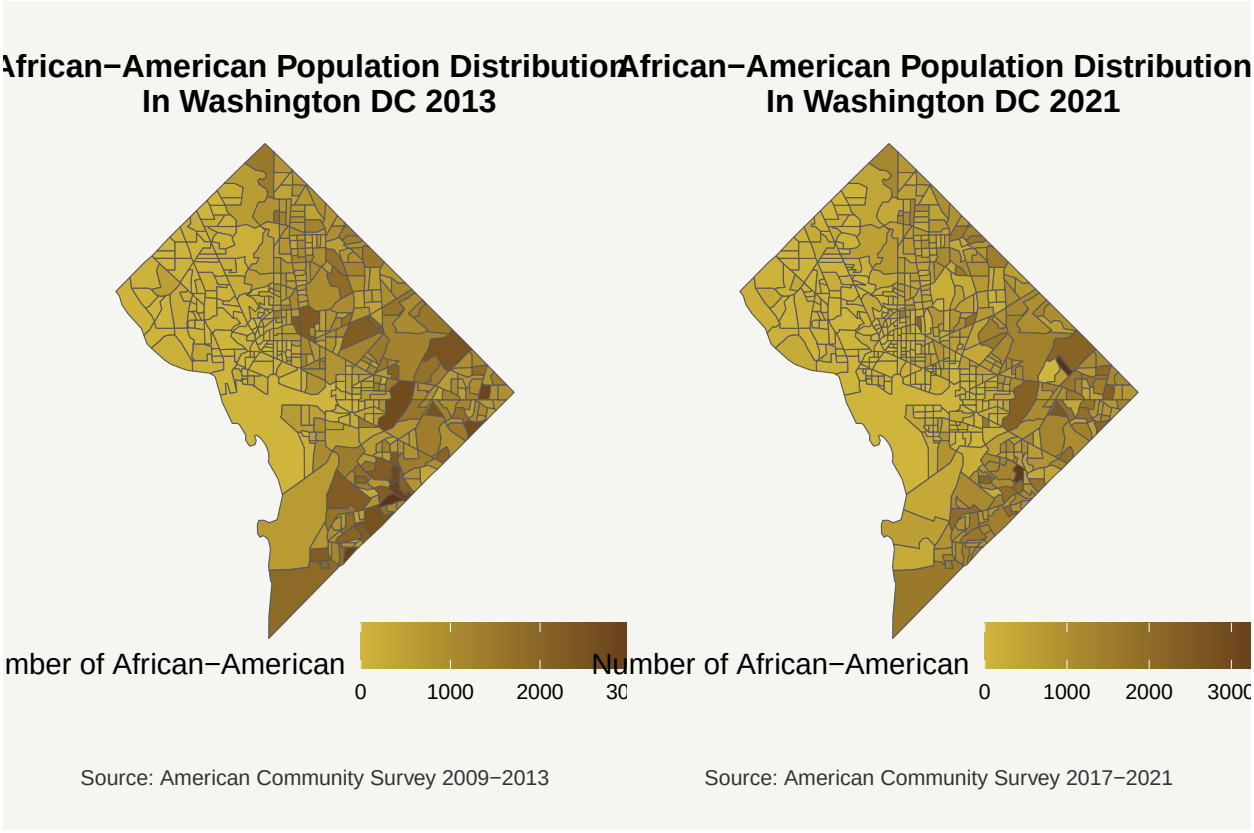
```
##   |                                                                              |
```

Plot the geographic distribution of African-Americans in 2013 and 2021.

```
p1 <- ggplot(black1) +
  geom_sf(aes(fill = estimate)) +
  scale_fill_gradient(low = "#CFB53B", high = "#5C3317")+
  theme_void() +
  labs(fill = "Number of African-American",
       title = paste("African-American Population Distribution \n In Washington DC 2013"),
       caption = "Source: American Community Survey 2009-2013")+
  theme(plot.title = element_text(color = "black", size = 12, face = "bold",
                                  margin = margin(t = 5),
                                  hjust = 0.5),
        plot.title.position = "plot",
        axis.text = element_text(size = 10, color = "black"),
        legend.position = c(0.5, 0),
```

```
        legend.direction = 'horizontal',
        legend.key.width = unit(0.8, "cm"),
        legend.text = element_text(color = "black",  size=8),
        plot.caption.position = "plot",
        plot.caption = element_text(color = "grey20", size = 8,
                                    lineheight = 0.3, hjust = 0.5,
                                    margin = margin(t = 40)),
        text = element_text(size = 11),
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        plot.margin = margin(15, 35, 15, 35),
        panel.background = element_rect(fill = "#f5f5f2",color = NA),
        plot.background = element_rect(fill = "#f5f5f2",color = NA),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_blank())

p2 <- ggplot(black2) +
  geom_sf(aes(fill = estimate)) +
  scale_fill_gradient(low = "#CFB53B", high = "#5C3317")+
  theme_void() +
  labs(fill = "Number of African-American",
       title = paste("African-American Population Distribution \n In Washington DC 2021"),
       caption = "Source: American Community Survey 2017-2021")+
  theme(plot.title = element_text(color = "black", size = 12, face = "bold",
                                  margin = margin(t = 5),
                                  hjust = 0.5),
        plot.title.position = "plot",
        axis.text = element_text(size = 10, color = "black"),
        legend.position = c(0.5, 0),
        legend.direction = 'horizontal',
        legend.key.width = unit(0.8, "cm"),
        legend.text = element_text(color = "black",  size=8),
        plot.caption.position = "plot",
        plot.caption = element_text(color = "grey20", size = 8,
                                    lineheight = 0.3, hjust = 0.5,
                                    margin = margin(t = 40)),
        text = element_text(size = 11),
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        plot.margin = margin(15, 35, 15, 35),
        panel.background = element_rect(fill = "#f5f5f2",color = NA),
        plot.background = element_rect(fill = "#f5f5f2",color = NA),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
```

```
        axis.line = element_blank())

library(gridExtra)

grid.arrange(p1, p2, ncol = 2)
```



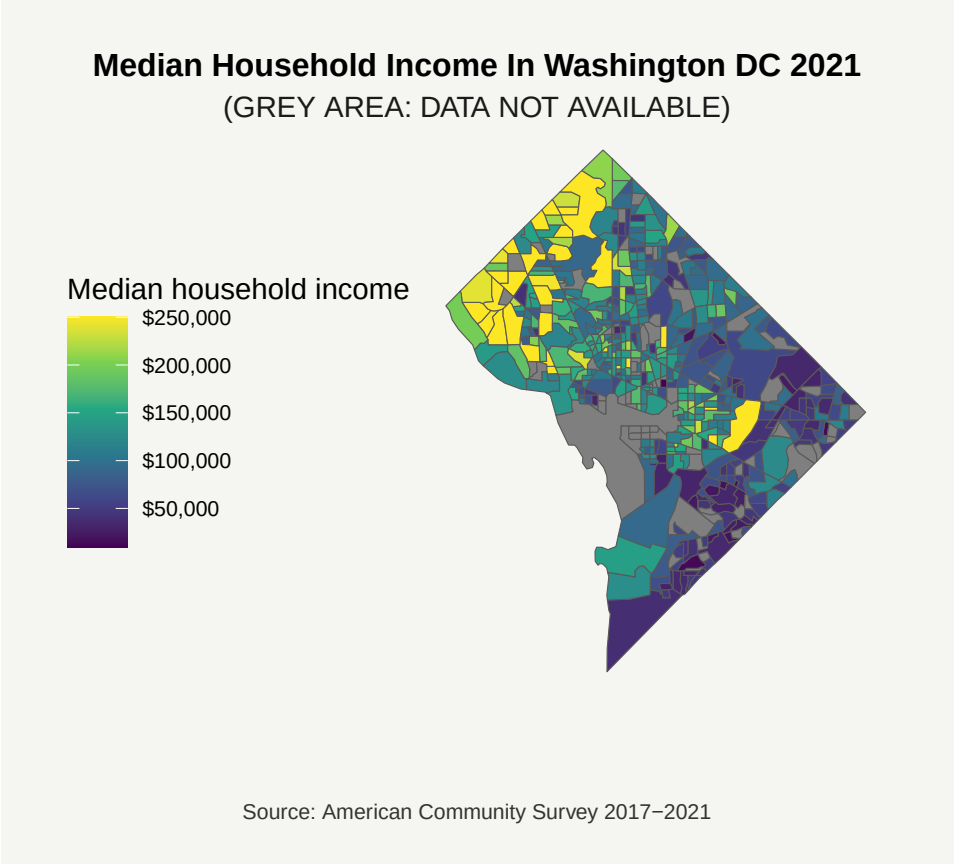
Read median household income in 2021.

```
income <- get_acs(
  geography = "block group",
  state = "DC",
  variables = "B19013_001",
  year = 2021,
  geometry = TRUE
)
```

Plot median household income in 2021.

```
ggplot(income) +
  geom_sf(aes(fill = estimate)) +
  scale_fill_viridis_c(labels = scales::label_dollar()) +
  theme_void() +
  labs(fill = "Median household income",
```

```
     title = paste("Median Household Income In Washington DC 2021"),
 subtitle = "(GREY AREA: DATA NOT AVAILABLE)",
 caption = "Source: American Community Survey 2017-2021")+
theme(plot.title = element_text(color = "black", size = 12, face = "bold",
                                margin = margin(t = 5),
                                hjust = 0.5),
      plot.title.position = "plot",
      axis.text = element_text(size = 10, color = "black"),
      plot.subtitle = element_text(color = "grey10", size = 11,
        lineheight = 3, hjust = 0.5,
        margin = margin(t = 5)),
      legend.position = "left",
      legend.key.width = unit(0.8, "cm"),
      legend.text = element_text(color = "black",  size=8),
      plot.caption.position = "plot",
      plot.caption = element_text(color = "grey20", size = 8,
                                  lineheight = 0.3, hjust = 0.5,
                                  margin = margin(t = 40)),
      text = element_text(size = 11),
      axis.title.x=element_blank(),
      axis.text.x=element_blank(),
      axis.ticks.x=element_blank(),
      axis.title.y = element_blank(),
      axis.text.y = element_blank(),
      axis.ticks.y = element_blank(),
      plot.margin = margin(15, 25, 15, 25),
      panel.background = element_rect(fill = "#f5f5f2",color = NA),
      plot.background = element_rect(fill = "#f5f5f2",color = NA),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.line = element_blank())
```



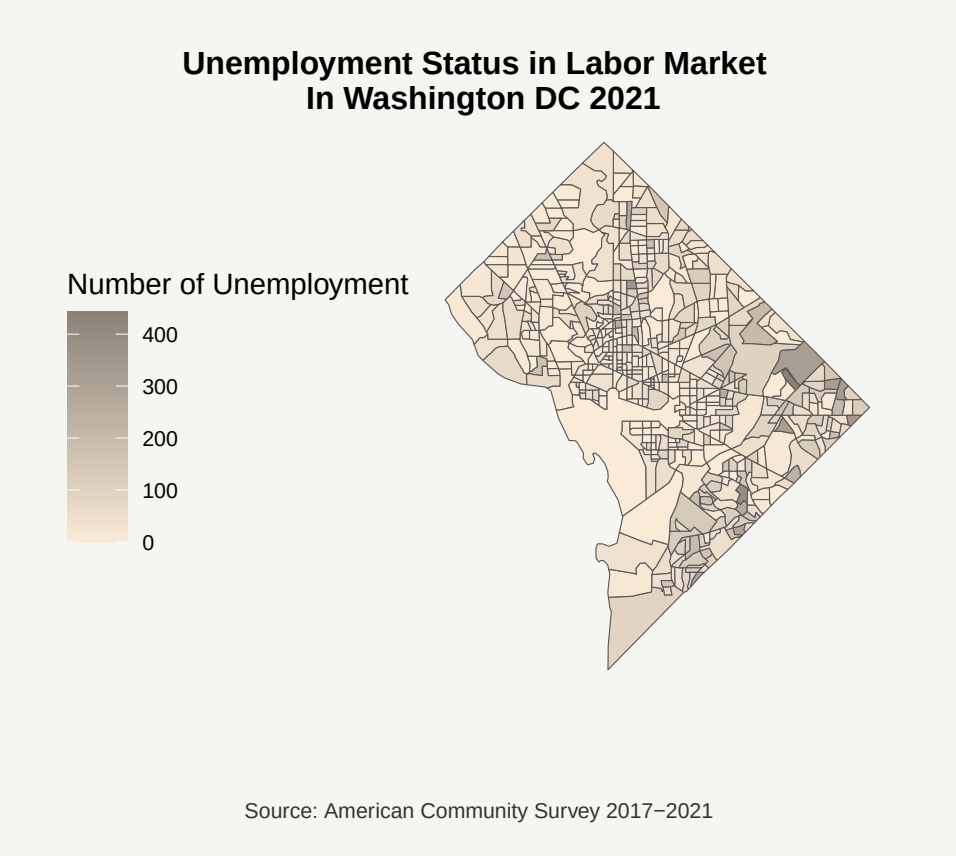
Read unemployment data in 2021

```
unemployment <- get_acs(
  geography = "block group",
  state = "DC",
  variables = "B23025_005",
  survey = "acs5",
  year = 2021,
  geometry = TRUE
)
```

Plot unemployment status for the population 16 and above in 2021.

```
ggplot(unemployment) +
  geom_sf(aes(fill = estimate)) +
  scale_fill_gradient(low = "antiquewhite", high = "antiquewhite4",na.value = "grey90")+
  theme_void() +
  labs(fill = "Number of Unemployment",
       title = paste("Unemployment Status in Labor Market \n In Washington DC 2021"),
       caption = "Source: American Community Survey 2017-2021")+
  theme(plot.title = element_text(color = "black", size = 12, face = "bold",
                                  margin = margin(t = 5),
                                  hjust = 0.5),
```

```
        plot.title.position = "plot",
        axis.text = element_text(size = 10, color = "black"),
        legend.position = "left",
        legend.key.width = unit(0.8, "cm"),
        legend.text = element_text(color = "black",  size=8),
        plot.caption.position = "plot",
        plot.caption = element_text(color = "grey20", size = 8,
                                    lineheight = 0.3, hjust = 0.5,
                                    margin = margin(t = 40)),
        text = element_text(size = 11),
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        plot.margin = margin(15, 25, 15, 25),
        panel.background = element_rect(fill = "#f5f5f2",color = NA),
        plot.background = element_rect(fill = "#f5f5f2",color = NA),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_blank())
```

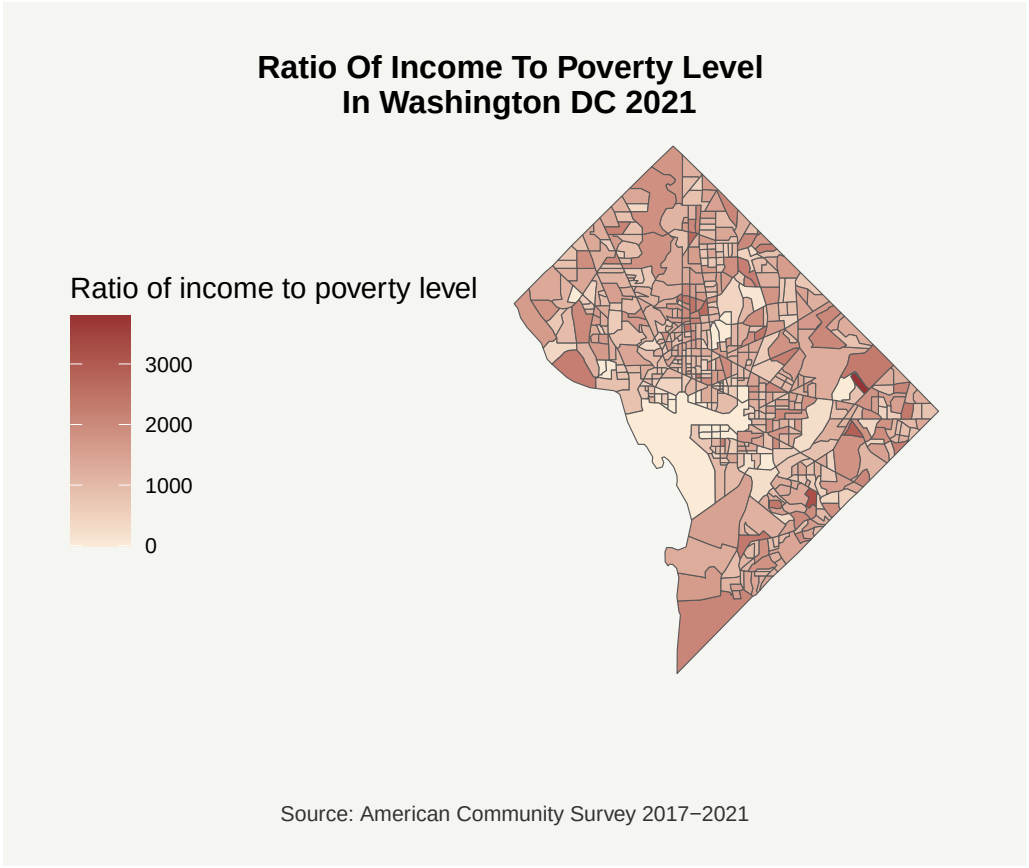


Read 'poverty' data

```
poverty <- get_acs(
  geography = "block group",
  state = "DC",
  variables = "C17002_001",
  survey = "acs5",
  year = 2021,
  geometry = TRUE
)
```

Mapping the ratio of income to poverty level in the past 12 months

```
ggplot(poverty) +
  geom_sf(aes(fill = estimate)) +
  scale_fill_gradient(low = "antiquewhite", high = "#993333",na.value = "grey90")+
  theme_void() +
  labs(fill = "Ratio of income to poverty level",
       title = paste("Ratio Of Income To Poverty Level \n In Washington DC 2021"),
       caption = "Source: American Community Survey 2017-2021")+
  theme(plot.title = element_text(color = "black", size = 12, face = "bold",
                                  margin = margin(t = 5),
                                  hjust = 0.5),
        plot.title.position = "plot",
        axis.text = element_text(size = 10, color = "black"),
        legend.position = "left",
        legend.key.width = unit(0.8, "cm"),
        legend.text = element_text(color = "black",  size=8),
        plot.caption.position = "plot",
        plot.caption = element_text(color = "grey20", size = 8,
                                    lineheight = 0.3, hjust = 0.5,
                                    margin = margin(t = 40)),
        text = element_text(size = 11),
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        plot.margin = margin(15, 25, 15, 25),
        panel.background = element_rect(fill = "#f5f5f2",color = NA),
        plot.background = element_rect(fill = "#f5f5f2",color = NA),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_blank())
```

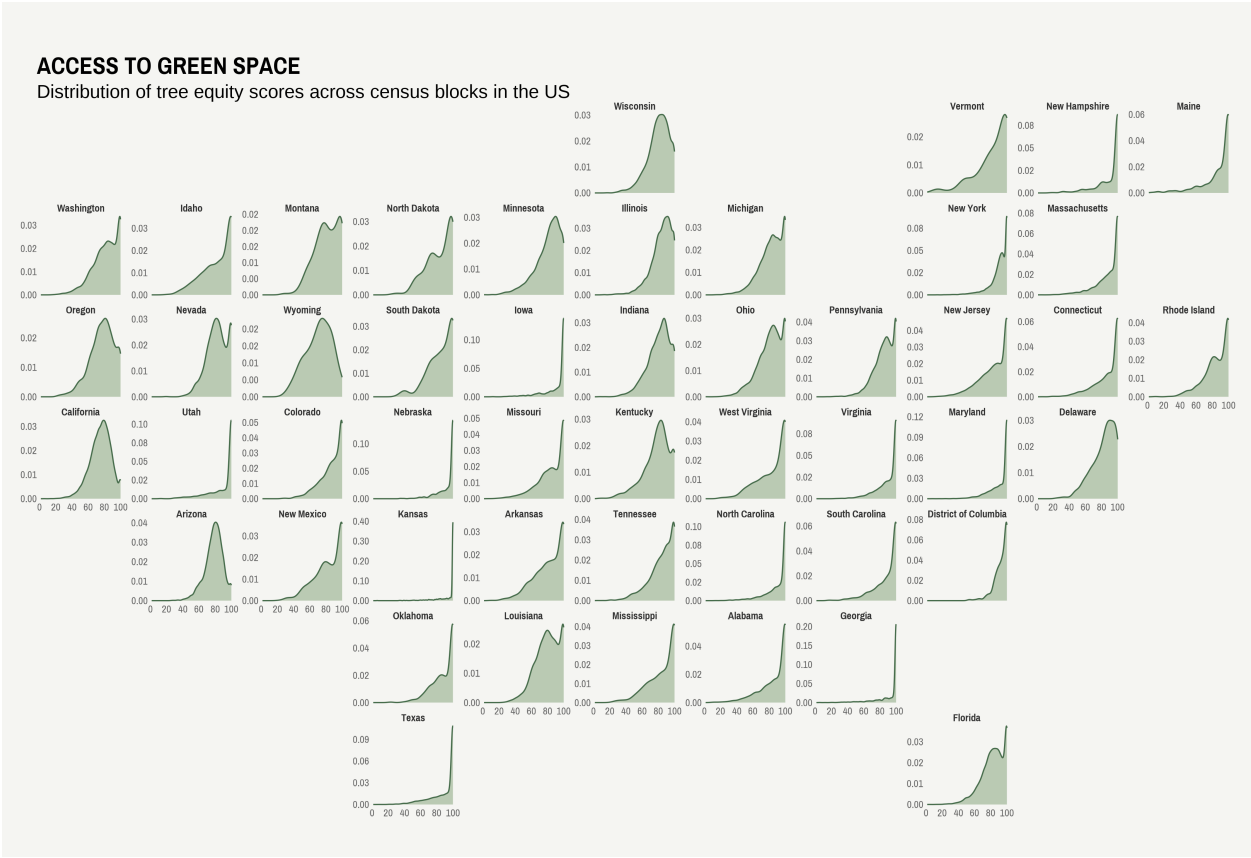

Ratio Of Income To Poverty Level
In Washington DC 2021
Ratio of income to poverty level
3000
2000
1000
0
Source: American Community Survey 2017−2021

Code Notebook for the Green Space Data Challenge

Green Space in the U.S.

```
library(sf)
library(geofacet)
library(showtext)
library(tidyverse)
states <- read.csv("state-names.csv") # downloaded https://worldpopulationreview.com/states/state-abbrev

font_add_google("Pragati Narrow")
showtext_auto()

# prepare a function to read zip urls with shapefiles
read_shape_URL <- function(URL){
  cur_tempfile <- tempfile()
  download.file(url = URL, destfile = cur_tempfile)
  out_directory <- tempfile()
  unzip(cur_tempfile, exdir = out_directory)

  read_sf(dsn = out_directory)
}

url <- paste0("https://national-tes-data-share.s3.amazonaws.com/national_tes_share/", "al", ".zip.zip")
map <- read_shape_URL(url) %>% select(tes, state, priority)

data_lists <- list()
for (i in 1:nrow(states)){
  state <- states$lower_code[i]
  print(state)
  URL <- paste0("https://national-tes-data-share.s3.amazonaws.com/national_tes_share/", state, ".zip.zip")
  if (RCurl::url.exists(URL) == T) {
    map <- read_shape_URL(paste0("https://national-tes-data-share.s3.amazonaws.com/national_tes_share/"
    data_lists[[i]] <- map %>% select(tes, state, priority)
  }
}
data <- do.call(rbind, data_lists)
saveRDS(data, "data.RDS")
data <- readRDS("data.RDS")

data <- na.omit(data)
summary(data$tes)

ggplot(data) +
  geom_density(aes(x = tes), color = "#466c4b", fill = "#7fa074", alpha = 0.5) +
  coord_cartesian(clip = "off") +
  facet_geo(vars(state), scales = "free_y", grid = us_state_grid1[c(-2, -11), ], label = "name") +
```

```
  scale_x_continuous(breaks = seq(0, 100, 20)) +
  scale_y_continuous(
    labels = scales::number_format(accuracy = 0.01)) +
  labs(x = "", y = "",
       title = "ACCESS TO GREEN SPACE",
       subtitle = "Distribution of tree equity scores across census blocks in the US",
       caption = str_wrap("
       Tree Equity Score (TES) computes how much tree canopy and surface temperature align with income,
       employment, race, age and health factors in the US, collected by American Forest | Visualization
  theme_void(base_family = "Pragati Narrow") +
  theme(strip.text = element_text(face = "bold", color = "grey20", size = 30),
        legend.position = "none",
        axis.text = element_text(color = "grey40", size = 30),
        strip.background = element_blank(),
        plot.background = element_rect(fill = "#f5f5f2", color = NA),
        plot.margin = margin(40, 15, 20, 15),
        plot.title = element_text(face = "bold", size = 70, margin = margin(l=0, t=5)),
        plot.subtitle = element_text(lineheight = 1, size = 50, margin(l=0, t=7)),
        plot.caption = element_text(margin = margin(t=35), color = "grey20", size = 30),
        plot.caption.position = "plot")
```

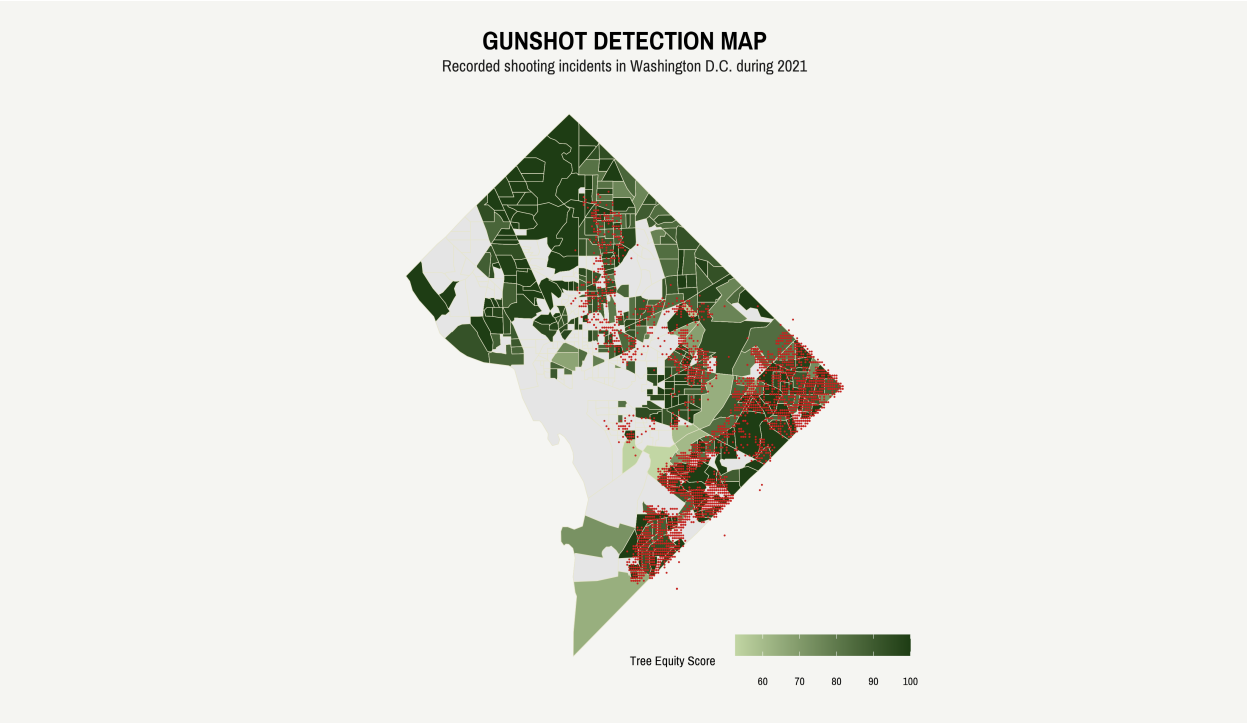


Green Space and Gunshots in Washington D.C.

```
library(sf)
library(showtext)
```

```
library(lubridate)
library(tidyverse)

# read tree equity score data
dc_tes <- read.csv("District_of_Columbia.csv") %>%
  mutate(GEOID = as.character(geoid))


# read gunshot data
dc_gunshot <- read.csv("Shot_Spotter_Gun_Shots.csv") %>%
  mutate(date = as_date(DATETIME),
         year = year(date)) %>%
  filter((year == 2021) & (TYPE %in% c("Single_Gunshot", "Multiple_Gunshots", "Multiple Gunshots", "Sing
  st_as_sf(., coords = c("LONGITUDE", "LATITUDE"), crs = 4326, agr = "constant")


#### plot tree equity score and gunshots
theme_update(

  # axis
  text = element_text(size = 35),
  axis.text.x = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks = element_blank(),

  # titles
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  plot.margin = margin(15, 25, 15, 25),
  plot.background = element_rect(fill = "#f5f5f2", color = NA),
  panel.background = element_rect(fill = "#f5f5f2", color = NA),

  legend.position = c(0.8, 0),
  legend.direction = 'horizontal',
  legend.key.width = unit(1, "cm"),
  legend.text = element_text(color = "black",  size=30),

  plot.title = element_text(
    color = "black",
    size = 70,
    face = "bold",
    margin = margin(t = 10),
    hjust = 0.5
  ),
  plot.subtitle = element_text(
    color = "grey10",
    size = 45,
    lineheight = 3,
    margin = margin(t = 5),
    hjust = 0.5
  ),
  plot.title.position = "plot",
  plot.caption.position = "plot",
```

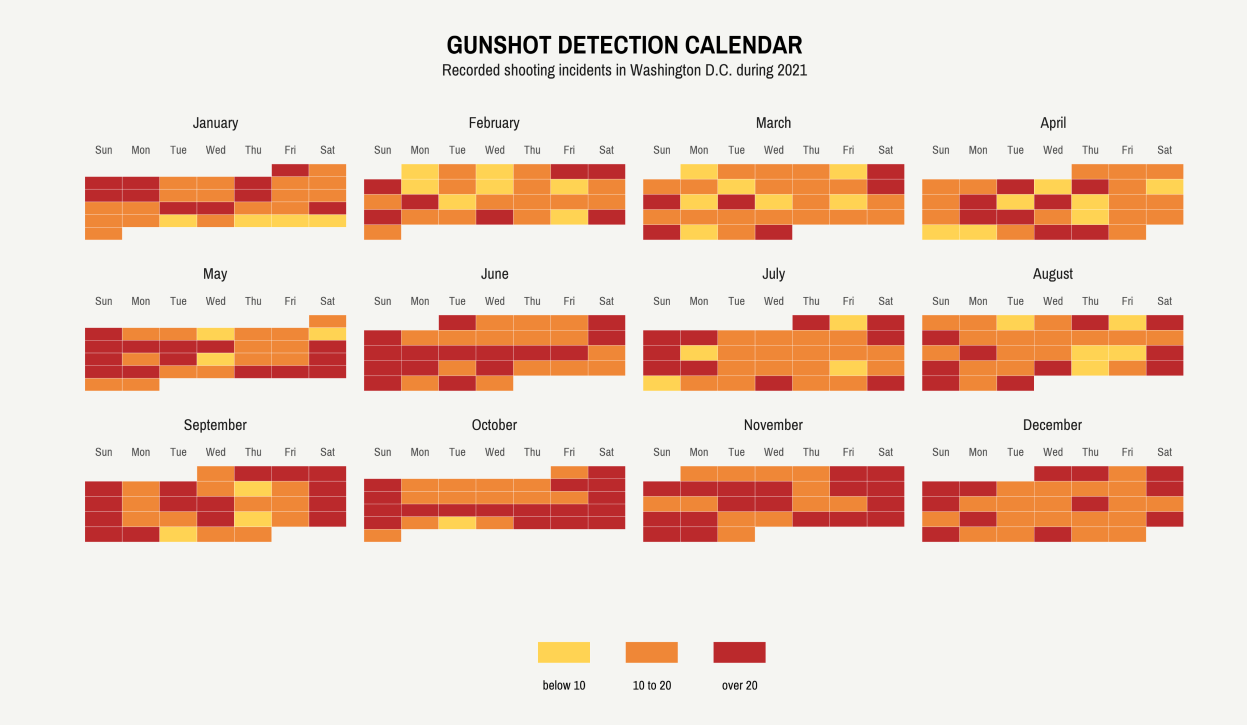
```
  plot.caption = element_text(
    color = "grey20",
    size = 35,
    lineheight = 0.3,
    hjust = 0.5,
    margin = margin(t = 40))
)


ggplot(dc_tes) +
  geom_sf(aes(fill = tes), color = "#e7e5cc") +
  scale_fill_continuous(low = "#c2d6a4", high = "#1e3d14", na.value = "grey90", name= "Tree Equity Score
  geom_sf(data = dc_gunshot, color = "#c62320", size = 0.1, inherit.aes = FALSE) +
  coord_sf(xlim = c(-77.12, -76.90), ylim = c(38.79, 39.01), expand = F) +
  labs(x = "", y = "", title = "GUNSHOT DETECTION MAP",
       subtitle = "Recorded shooting incidents in Washington D.C. during 2021",
       caption = str_wrap("Data comes from ShotSpotter gunshot detection system. Incidents of probable g
       employment, race, age and health factors at each block | Visualization by Zhaowen Guo", width = 1
```




```
library(showtext)
library(lubridate)
library(tidyverse)
gunshot <- read_csv("Shot_Spotter_Gun_Shots.csv")

unique(gunshot$TYPE)

gunshot_daily <- gunshot %>%
  mutate(date = as_date(DATETIME),
         year = year(date)) %>%
  filter((year == 2021) & (TYPE %in% c("Single_Gunshot", "Multiple_Gunshots", "Multiple Gunshots", "Sing
```

```
  group_by(date) %>%
  summarise(shots = n()) %>%
  ungroup() %>%
  mutate(week_day = str_sub(weekdays(date), 1, 3),
         month_day = day(date),
         month = month(date),
         week_start = ifelse(month_day == 1 | week_day == "Sun", 1, 0)) %>%
  group_by(month) %>%
  mutate(week = cumsum(week_start),
         month_name = months(date)) %>%
  ungroup() %>%
  mutate(shots_range = case_when(shots <= 10 ~ "1",
                                 shots >10 & shots <= 20 ~ "2",
                                 shots >20 ~ "3"))

week_day_code <- c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
gunshot_daily$week_day <- factor(gunshot_daily$week_day, levels = week_day_code)
month_code <- c("January", "February", "March", "April", "May", "June", "July", "August", "September",
gunshot_daily$month_name <- factor(gunshot_daily$month_name, levels = month_code)


# theme
font_add_google("Pragati Narrow")
showtext_auto()


# customize theme
theme_set(theme_minimal(base_family = "Pragati Narrow"))

theme_update(
  # legend
  legend.title = element_blank(),
  legend.position = 'bottom',
  legend.direction = 'horizontal',
  legend.key.width = unit(1.5, "cm"),
  legend.text = element_text(color = "black",  size=35),
  legend.box.margin = margin(t = 35),
  legend.spacing.x = unit(1, "cm"),
  legend.spacing.y = unit(0.5, "cm"),

  # axis
  axis.text.y = element_blank(),
  axis.text.x = element_text(vjust = 50),
  text = element_text(size = 40),
  strip.text.x = element_text(size = 43, margin = margin(b = 25)),

  # titles
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  plot.margin = margin(20, 50, 20, 50),
  plot.background = element_rect(fill = "#f5f5f2", color = NA),
  panel.background = element_rect(fill = "#f5f5f2", color = NA),
```

```
  #legend.title.align=1,
  plot.title = element_text(
    color = "black",
    size = 70,
    face = "bold",
    margin = margin(t = 10),
    hjust = 0.5
  ),
  plot.subtitle = element_text(
    color = "grey10",
    size = 45,
    lineheight = 3,
    margin = margin(t = 5, b = 30),
    hjust = 0.5
  ),
  plot.title.position = "plot",
  plot.caption.position = "plot",
  plot.caption = element_text(
    color = "grey20",
    size = 35,
    lineheight = 0.5,
    hjust = 0.5,
    margin = margin(t = 30))
)


gunshot_daily %>%
  ggplot(aes(x = week_day, y = week)) +
  geom_tile(aes(fill = shots_range), color = "white") +
  scale_fill_manual(values = MetBrewer::met.brewer("Tam", n=3),
                    labels = c("below 10", "10 to 20", "over 20"),
                    guide = guide_legend(label.position = "bottom", nrow = 1)) +
  facet_wrap(~month_name, scales = "free") +
  scale_y_reverse() +
  scale_x_discrete(position = "bottom") +
  labs(x = "", y = "", title = "GUNSHOT DETECTION CALENDAR",
       subtitle = "Recorded shooting incidents in Washington D.C. during 2021",
       caption = str_wrap("Data comes from ShotSpotter gunshot detection system. Incidents of probable g
```

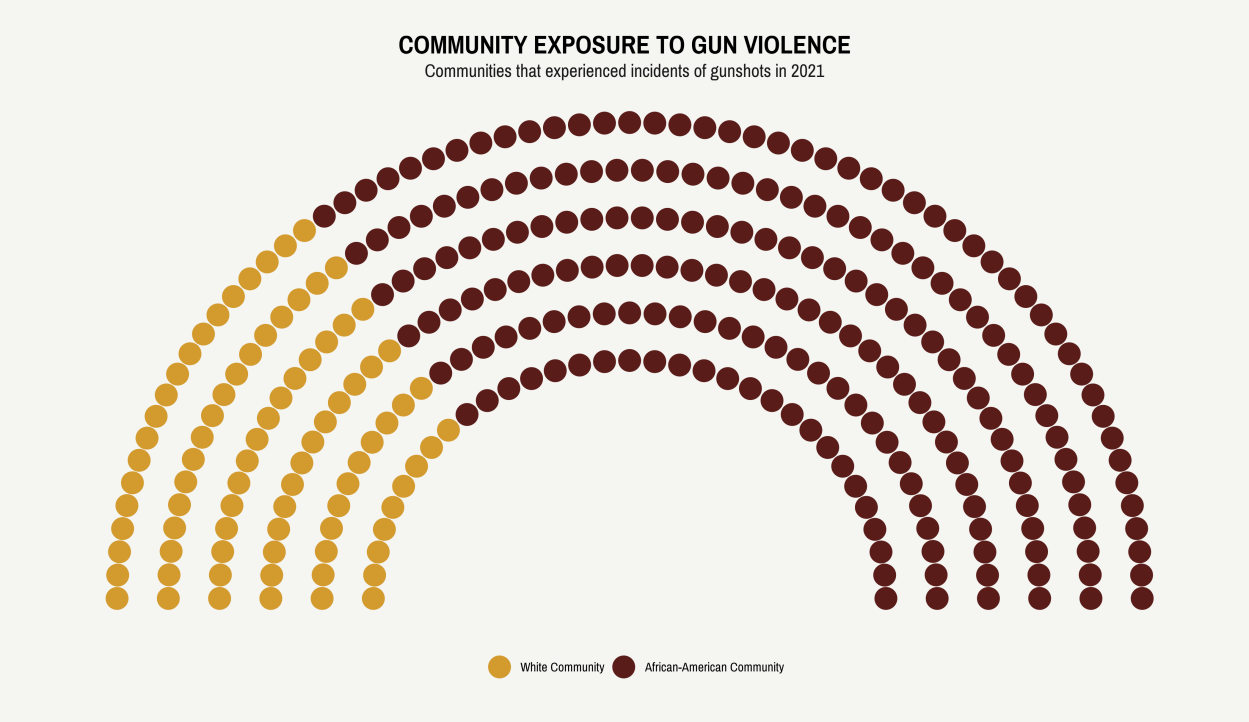


Racial and Ethnic Disparities of Gun Violence Exposure

```
library(sf)
library(ggparliament)
library(MetBrewer)
library(showtext)
library(lubridate)
library(tidycensus)
library(tidyverse)

# retrieve variables that may need
vars <- load_variables(year = 2020, dataset= "acs5") %>%
  filter(geography == "block group")

# B28012_002 total population (3 and older)
# B02001_002 white;
# B02001_003 black;
# B02001_004 American Indian and Alaska Native;
# B02001_005 Asian;
# B02001_005 Naive Hawaiian and Pacific Islander
# B03003_003 Hispanic or Latino
# B19013_001 inflation adjusted median household income
# B23025_002 employment status for the adult population


# read race data from 2020 census
dc_demographics <- get_acs(
  geography = "block group",
  variables = c("B28012_002", # population
```

```
                "B02001_002","B02001_003", "B02001_004", "B02001_005", "B02001_006", # race
                "B03003_002","B03003_003", # ethnicity
                "B19013_001", "B23025_002"), # socioeconomic
  state = "DC",
  year = 2020,
  geometry = TRUE
)

# regroup race variable
dc_race_community <- dc_demographics %>%
  filter(grepl("B02001", variable)) %>%
  filter(!estimate == 0) %>%
  group_by(GEOID) %>%
  top_n(1, estimate) %>%
  mutate(variable = case_when(variable == "B02001_002" ~ "White",
                              variable == "B02001_003" ~ "African American",
                              T ~ "Others")) %>%
  st_transform(4326)

# read gunshot data
dc_gunshot <- read.csv("Shot_Spotter_Gun_Shots.csv") %>%
  mutate(date = as_date(DATETIME),
         year = year(date)) %>%
  filter((year == 2021) & (TYPE %in% c("Single_Gunshot", "Multiple_Gunshots", "Multiple Gunshots", "Sing
  st_as_sf(., coords = c("LONGITUDE", "LATITUDE"), crs = 4326, agr = "constant")

# count gunshots within each racial community
dc_gunshot_counts <- st_join(dc_gunshot, dc_race_community, join = st_within) %>%
  as_tibble() %>%
  na.omit() %>%
  group_by(variable, GEOID) %>%
  summarise(total = n()) %>%
  ungroup()

table(dc_gunshot_counts$variable)
# 293 out of 555 communities experienced gunshot incidents; among these gun-exposure communities, 71.3%


# prepare gunshot per racial group data for parliament visuals
community_gunshot <- data.frame(groups = c("White Community", "African-American Community"),
                                count = c(84, 209),
                                colors = c("#d39a2d","#591c19"))


## theme
font_add_google("Pragati Narrow")
showtext_auto()

# customize theme
theme_set(theme_minimal(base_family = "Pragati Narrow"))

theme_update(legend.position = "bottom",
             legend.text = element_text(color = "black",  size=35),
```

```
            panel.grid.major = element_blank(),
            panel.grid.minor = element_blank(),
            plot.background = element_rect(fill = "#f5f5f2", color = NA),
            plot.margin = margin(20, 45, 20, 45),
            panel.background = element_rect(fill = "#f5f5f2", color = NA),

            plot.title = element_text(
              color = "black",
              size = 70,
              face = "bold",
              margin = margin(t = 10),
              hjust = 0.5
            ),
            plot.subtitle = element_text(
              color = "grey10",
              size = 50,
              lineheight = 3,
              margin = margin(t = 5, b=15),
              hjust = 0.5
            ),
            plot.title.position = "plot",
            plot.caption.position = "plot",
            plot.caption = element_text(
              color = "grey20",
              size = 30,
              lineheight = 0.3,
              hjust = 0.5,
              margin = margin(t = 40))
)


# plot racial groups and gunshots
community_gunshot_data <- parliament_data(election_data = community_gunshot,
                                          type = "semicircle",
                                          parl_rows = 6,
                                          party_seats = community_gunshot$count)

community_gunshot_data %>%
  ggplot(aes(x=x, y=y, color=groups)) +
  geom_parliament_seats(size=8) +
  theme(legend.title = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank()) +
  scale_color_manual(values = community_gunshot_data$colors, limits = community_gunshot_data$groups) +
  labs(x="", y="", title = "COMMUNITY EXPOSURE TO GUN VIOLENCE",
       subtitle = "Communities that experienced incidents of gunshots in 2021",
       caption = str_wrap("Data comes from ShotSpotter gunshot detection system. Incidents of probable g
```




```
4/5 # 80%
289/550 # 53%

icon_path = "path://M544 64h-16V56C528 42.74 517.3 32 504 32S480 42.74 480 56V64H43.17C19.33 64 0 83.33

hispanic = data.frame(ethnic = c("80% of Hispanic or Latino Communities", "53% of Non-Hispanic or \n No
                      ratio = c(40, 25),
                      path = c(icon_path,
                               icon_path))

hispanic %>%
  e_charts(ethnic) %>%
  e_x_axis(splitLine=list(show = FALSE),
           axisTick=list(show=FALSE),
           axisLine=list(show=FALSE),
           axisLabel = list(show=FALSE)) %>%
  e_y_axis(max=100,
           splitLine=list(show = FALSE),
           axisTick=list(show=FALSE),
           axisLine=list(show=FALSE),
           axisLabel=list(show=FALSE)) %>%
  e_color(color = c('#811e18','grey'), background = "#f5f5f2") %>%
  e_pictorial(ratio, symbol = path, z=10, name = "",
              symbolBoundingData= 50, symbolClip= TRUE) %>%
  e_pictorial(ratio, symbol = path, name= '',
              symbolBoundingData= 50) %>%
  e_legend(show = FALSE) %>%
  e_grid(bottom = "35%")
# save the graph by click on the "download" icon
```

```
library(magick)
library(ggpubr)
background <- image_read("ethnic-gun.png")
xaxis <- data.frame(xaxis = c(1, 2, 3),
                    labels = c("", "", ""))
yaxis <- data.frame(yaxis = c(1, 2, 3),
                    labels = c("", "", ""))

ggplot() +
  background_image(background) +
  geom_text(data = xaxis, aes(x = xaxis, y = 0, label = labels)) +
  geom_text(data = yaxis, aes(x = 0, y = yaxis, label = labels)) +
  labs(x="",y="") +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank(),
        plot.background = element_rect(fill = "#f5f5f2", color = NA)) +
  annotate(geom = "text", x = 0.8, y = 0.8, label = "80% of Hispanic or Latino Communities",
           size = 15,  family = "Pragati Narrow") +
  annotate(geom = "text", x = 2.1, y = 0.8, label = "53% of Other Communities",
           size = 15,  family = "Pragati Narrow") +
  annotate(geom = "text", x = 1.45, y = 2.95, label = "COMMUNITY EXPOSURE TO GUN VIOLENCE",
           size = 26,  family = "Pragati Narrow", fontface = "bold") +
  annotate(geom = "text", x = 1.45, y = 2.82, label = "Communities that experienced inidents of gunshots
           size = 19,  family = "Pragati Narrow") +
  annotate(geom = "text", x = 1.5, y = 0, label = "Data comes from ShotSpotter gunshot detection system
           size = 12,  family = "Pragati Narrow",color = "grey20")
```

Relationship between Green Space and Gun Violence Exposure

```
options(sciphen = 999)
library(sf)
library(MetBrewer)
library(showtext)
library(lubridate)
library(tidycensus)
library(tidyverse)

font_add_google("Pragati Narrow")
showtext_auto()

# retrieve variables that may need
vars <- load_variables(year = 2020, dataset= "acs5") %>%
  filter(geography == "block group")

# B28012_002 total population (3 and older)
# B02001_002 white;
# B02001_003 black;
# B02001_004 American Indian and Alaska Native;
# B02001_005 Asian;
# B02001_005 Naive Hawaiian and Pacific Islander
# B03003_003 Hispanic or Latino
# B19013_001 inflation adjusted median household income
# B23025_002 employment status for the adult population


# read demographics data from 2020 census
demographics <- get_acs(
  geography = "block group",
  variables = c("B28012_002", # population
                "B02001_002","B02001_003", "B02001_004", "B02001_005", "B02001_006", # race
                "B03003_002","B03003_003", # ethnicity
                "B19013_001", "B23025_002"), # socioeconomic
  state = "DC",
  year = 2020,
  geometry = TRUE
)

# widen demographics data
block_pop <- demographics %>%
  pivot_wider(names_from = "variable", values_from = "estimate") %>%
  select(-moe) %>%
  replace(is.na(.), 0) %>%
  group_by(GEOID) %>%
  summarise(across(B02001_002:B28012_002, sum)) %>%
  ungroup() %>%
  st_transform(4326)

# construct racial group variable
black_community <- demographics %>%
  filter(grepl("B02001", variable)) %>%
```

```
  filter(!estimate == 0) %>%
  group_by(GEOID) %>%
  top_n(1, estimate) %>%
  filter(variable == "B02001_003")

hispanic_community <- demographics %>%
  filter(grepl("B03003", variable)) %>%
  filter(!estimate == 0) %>%
  group_by(GEOID) %>%
  top_n(1, estimate) %>%
  filter(variable == "B03003_003")

# read green space data per block
green <- read.csv("green_space_block.csv") %>%
  select(GreenSpace_Percent, GreenSpace_PerCapita, bgrp, Outside_500m_population_percent) %>%
  rename(GEOID = bgrp) %>%
  mutate(GEOID = as.character(GEOID))

# read tree equity score
tree <- read.csv("tree_equity_score.csv") %>%
  select(geoid, tes) %>%
  rename(GEOID = geoid) %>%
  mutate(GEOID = as.character(GEOID))

# read park data
parks_locations <- readRDS("parks_locations.RDS")

# read gunshot data
gunshot <- read.csv("Shot_Spotter_Gun_Shots.csv") %>%
  mutate(date = as_date(DATETIME),
         year = year(date)) %>%
  filter((year == 2021) & (TYPE %in% c("Single_Gunshot", "Multiple_Gunshots", "Multiple Gunshots", "Sing
  st_as_sf(., coords = c("LONGITUDE", "LATITUDE"), crs = 4326, agr = "constant")

# count gunshots within each block
gunshot_counts <- st_join(gunshot, block_pop, join = st_within) %>%
  as_tibble() %>%
  drop_na(GEOID) %>%
  group_by(GEOID) %>%
  mutate(gunshots = n(),
         gunshots_percap = gunshots/B28012_002) %>%
  ungroup()

# count parks within each block
park_counts <- st_join(parks_locations, block_pop, join = st_within) %>%
  as_tibble() %>%
  drop_na(GEOID) %>%
  group_by(GEOID) %>%
  summarise(small_parks = sum(USE_TYPE == "SMALL PARK"),
         parks = sum(USE_TYPE == "PARK"),
         large_parks = sum(USE_TYPE == "LARGE PARK"),
         total_parks_recreation = n()) %>%
  ungroup()
```

```
# combine all data
data <- gunshot_counts %>%
  select(GEOID, gunshots, gunshots_percap) %>%
  full_join(green) %>%
  full_join(park_counts) %>%
  full_join(tree) %>%
  full_join(block_pop) %>%
  unique() %>%
  mutate(gunshots = ifelse(is.na(gunshots), 0, gunshots),
         gunshots_percap = ifelse(is.na(gunshots_percap), 0, gunshots_percap),
         is_gunshots = ifelse(gunshots > 0, 1, 0),
         B19013_001 = ifelse(B19013_001 == 0, 0.00001, B19013_001), # income levels
         B28012_002 = ifelse(B28012_002 == 0, 0.00001, B28012_002), # population
         is_black = ifelse(GEOID %in% black_community$GEOID, 1, 0),
         is_hispanic = ifelse(GEOID %in% hispanic_community$GEOID, 1, 0),
         total_parks_recreation = ifelse(is.na(total_parks_recreation), 0, total_parks_recreation),
         parks = ifelse(is.na(parks), 0, parks),
         small_parks = ifelse(is.na(small_parks), 0, small_parks),
         large_parks = ifelse(is.na(large_parks), 0, large_parks))

# green space and binary gunshot outcomes
m1 <- glm(is_gunshots ~ GreenSpace_Percent * is_black + log(B19013_001) + log(B28012_002),
          data = data, family = "binomial")
summary(m1)

m2 <- glm(is_gunshots ~ tes * is_black, data = data,
          family = "binomial")
summary(m2)

# build logit model
library(simcf)
model <- is_gunshots ~ GreenSpace_Percent * is_black + log(B19013_001) + log(B28012_002)
mdata <- extractdata(model, data, na.rm=TRUE)

# run logit
logit.result <- glm(model, family = "binomial", data = mdata)

# extract pe and vc
pe.glm <- logit.result$coefficients
vc.glm <- vcov(logit.result)

# simulate betas under multivariate normal distribution
sims <- 1000
sim.betas <- MASS::mvrnorm(sims, pe.glm, vc.glm)

# specify counterfactuals
GreenSpace.hyp <- seq(min(mdata$GreenSpace_Percent), max(mdata$GreenSpace_Percent))
nscen <- length(GreenSpace.hyp)
blackScen <- nonblackScen <- cfMake(model, mdata, nscen)

# research question 1: expected probabilities for voting by age and marital status
for (i in 1:nscen) {
  # Married (loop over each age)
```

```
  blackScen <- cfChange(blackScen, "GreenSpace_Percent", x = GreenSpace.hyp[i], scen = i)
  blackScen <- cfChange(blackScen, "is_black", x = 1, scen = i)

  # Not Married (loop over each age)
  nonblackScen <- cfChange(nonblackScen, "GreenSpace_Percent", x = GreenSpace.hyp[i], scen = i)
  nonblackScen <- cfChange(nonblackScen, "is_black", x = 0, scen = i)
}

# Simulate expected probabilities for all scenarios
blackSims <- logitsimev(blackScen, sim.betas, ci=0.95)
nonblackSims <- logitsimev(nonblackScen, sim.betas, ci=0.95)

# make tibble for visualization
blackSims.tbl <-
  blackSims %>%
  bind_rows() %>%
  mutate(GreenSpace_Percent = GreenSpace.hyp,
         is_black = "1")

nonblackSims.tbl <-
  nonblackSims %>%
  bind_rows() %>%
  mutate(GreenSpace_Percent = GreenSpace.hyp,
         is_black = "0")

allSims.tbl <- bind_rows(blackSims.tbl, nonblackSims.tbl)

# visualize expected difference
ggplot(allSims.tbl, aes(x = GreenSpace_Percent, y = pe, ymin = lower, ymax = upper, color = is_black, fi
  geom_line(linetype = "dashed") +
  geom_ribbon(alpha = 0.5) +
  scale_color_manual(values = c("#d39a2d","#591c19"), labels = c("Non-Black Community", "Black Community
  scale_fill_manual(values = c("#d39a2d","#591c19"), labels = c("Non-Black Community", "Black Community
  labs(x = "Percentage of Green Space", y = "Probability of Exposure to Gun Violence",
       title = "The Relationship between Green Space and Gun Violence Exposure",
       subtitle = "Expected probabilities of exposure to gun violence in relation to the percentage of g
  theme_minimal(base_family = "Pragati Narrow") +
  theme(plot.background = element_rect(fill = "#f5f5f2", color = NA),
        panel.background = element_rect(fill = "#f5f5f2", color = NA),
        axis.text = element_text(size = 35, color = "grey10"),
        axis.title = element_text(size = 35, color = "grey10"),
        plot.title = element_text(
          color = "black",
          size = 70,
          face = "bold",
          margin = margin(t = 10),
          hjust = 0.5
        ),
        plot.subtitle = element_text(
          color = "grey10",
          size = 50,
          lineheight = 3,
          margin = margin(t = 5, b=15),
```

```
    hjust = 0.5
  ),
  legend.position = "bottom",
  legend.text = element_text(color = "black",  size=35))
```