# Final Web Application: Priority Neighborhoods for Detroit, MI KSI Pedestrian Crash Reducers

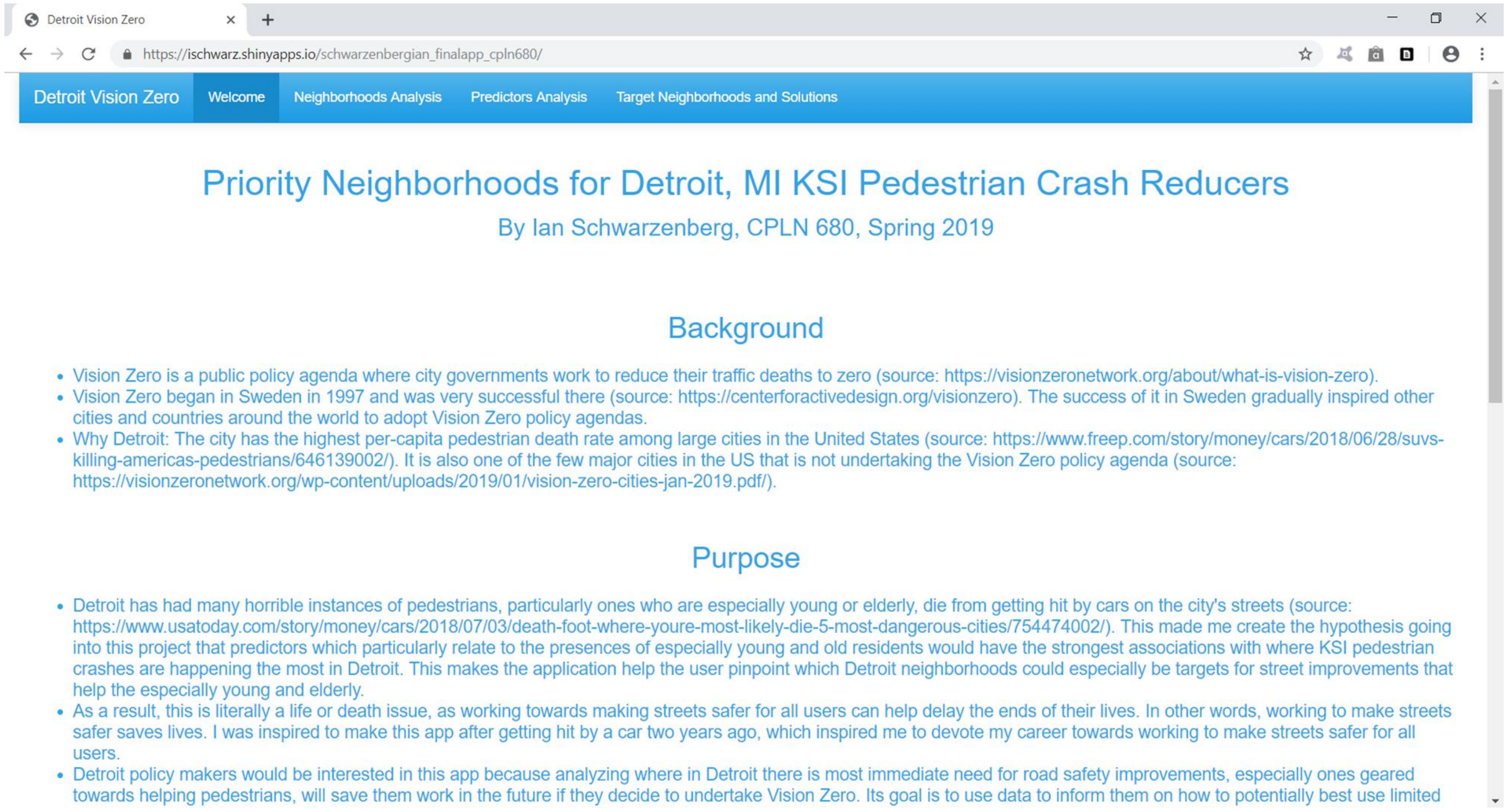
by Ian Schwarzenberg

CPLN 680, Spring 209

Link to final application:
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/

# Table of Contents

# Background


Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
Detroit Vision Zero
Welcome
Neighborhoods Analysis
Predictors Analysis
Target Neighborhoods and Solutions

## Priority Neighborhoods for Detroit, MI KSI Pedestrian Crash Reducers

By Ian Schwarzenberg, CPLN 680, Spring 2019

### Background

- Vision Zero is a public policy agenda where city governments work to reduce their traffic deaths to zero (source: https://visionzeronetwork.org/about/what-is-vision-zero).
- Vision Zero began in Sweden in 1997 and was very successful there (source: https://centerforactivedesign.org/visionzero). The success of it in Sweden gradually inspired other cities and countries around the world to adopt Vision Zero policy agendas.
- Why Detroit: The city has the highest per-capita pedestrian death rate among large cities in the United States (source: https://www.freep.com/story/money/cars/2018/06/28/suvs-killing-americas-pedestrians/646139002/). It is also one of the few major cities in the US that is not undertaking the Vision Zero policy agenda (source: https://visionzeronetwork.org/wp-content/uploads/2019/01/vision-zero-cities-jan-2019.pdf/).

### Purpose

- Detroit has had many horrible instances of pedestrians, particularly ones who are especially young or elderly, die from getting hit by cars on the city's streets (source: https://www.usatoday.com/story/money/cars/2018/07/03/death-foot-where-youre-most-likely-die-5-most-dangerous-cities/754474002/). This made me create the hypothesis going into this project that predictors which particularly relate to the presences of especially young and old residents would have the strongest associations with where KSI pedestrian crashes are happening the most in Detroit. This makes the application help the user pinpoint which Detroit neighborhoods could especially be targets for street improvements that help the especially young and elderly.
- As a result, this is literally a life or death issue, as working towards making streets safer for all users can help delay the ends of their lives. In other words, working to make streets safer saves lives. I was inspired to make this app after getting hit by a car two years ago, which inspired me to devote my career towards working to make streets safer for all users.
- Detroit policy makers would be interested in this app because analyzing where in Detroit there is most immediate need for road safety improvements, especially ones geared towards helping pedestrians, will save them work in the future if they decide to undertake Vision Zero. Its goal is to use data to inform them on how to potentially best use limited

## Background

Detroit, MI has the highest per-capita pedestrian death rate among large cities in the United States.[1] Adding on to the problem, Detroit's government is not undertaking Vision Zero, [2] a policy initiative undertaken by many cities across the world in which city governments work to ultimately prevent all road deaths and serious injuries. [3] Detroit has long been known for its fiscal problems, with the notable case of it undergoing bankruptcy in 2013. [3] This can naturally limit the city government from being table to target all necessary resources towards the problem of reducing pedestrian deaths in the city. However, the city can still have the ability to reduce its high rate of pedestrian deaths with solutions that have small costs but can have outsized impacts on reducing its pedestrian death rate.

[1] https://www.freep.com/story/money/cars/2018/06/28/suvs-killing-americas-pedestrians/646139002/
[2] https://visionzeronetwork.org/wp-content/uploads/2019/01/vision-zero-cities-jan-2019.pdf/
[3] https://www.citylab.com/transportation/2014/11/the-swedish-approach-to-road-safety-the-accident-is-not-the-major-problem/382995/

## Purpose

There are two main goals of the application: 1) show the user which neighborhoods have experienced the "most" KSI pedestrian crashes in recent years, and 2) show which predictors have the strongest associations with where KSI pedestrian crashes have been tending to happen. This way, Detroit city staff can identify neighborhoods where small or low-cost interventions can have outsized impacts on reducing these crashes, especially ones that involve naturally vulnerable groups like pedestrians under the ages of 18 and over the ages of 65. The hypothesis going into this analysis was that predictors that especially relate to pedestrians under the ages of 18 and over the ages of 65 would have the strongest associations with the locations of KSI pedestrian crashes in Detroit, more so than predictors that relate to diverse age groups.

This application could also give city officials ideas as to where to first implement projects if they decide to undertake Vision Zero in the future. The app only consider KSI pedestrian crashes as opposed to ones of any severity and victim type, and it ends up finding that predictors related to the presences of especially young and old residents have the strongest associations with KSI pedestrians crashes. These features further help city staff target its limited resources to neighborhoods where the chances of the most vulnerable pedestrians being seriously injured or killed could potentially be reduced the most. I became inspired to devote my career towards helping reduce the chances of pedestrians being hurt after being hit by a car myself two years ago.

## Data Sources

- KSI pedestrian crashes, 2011-16: https://data.detroitmi.gov/Transportation/Traffic-Crashes/9fph-m2jv
- Neighborhoods: https://data.detroitmi.gov/Government/Detroit-Neighborhoods/5mn6-ihjv
- Roads: http://gis-michigan.opendata.arcgis.com/datasets/all-roads-v17a
- Transit stops:
  - Woodward Avenue Light Rail stops: https://d3-d3.opendata.QGIS.com/datasets/ae0e1f2d9078426da08f51b318b7e314_0
  - City-owned bus stops: https://d3-d3.opendata.QGIS.com/datasets/9d24b9ce27294726aa8d769a0478b4d7_0
  - Suburb-owned bus stops: https://d3-d3.opendata.QGIS.com/datasets/93092f782bae4b97993954ba8c42e8ee_0?geometry=-83.163%2C42.321%2C-82.908%2C42.359 (many of these lines' stops are in Detroit)
- Parks: https://data.detroitmi.gov/Fun/Parks-2016/yu9n-k8rd
- Census tracts: US Census TIGER/Line Shapefiles
- Census tract age data 2012-16: US Census American Community Survey 2012-16 Table S0101

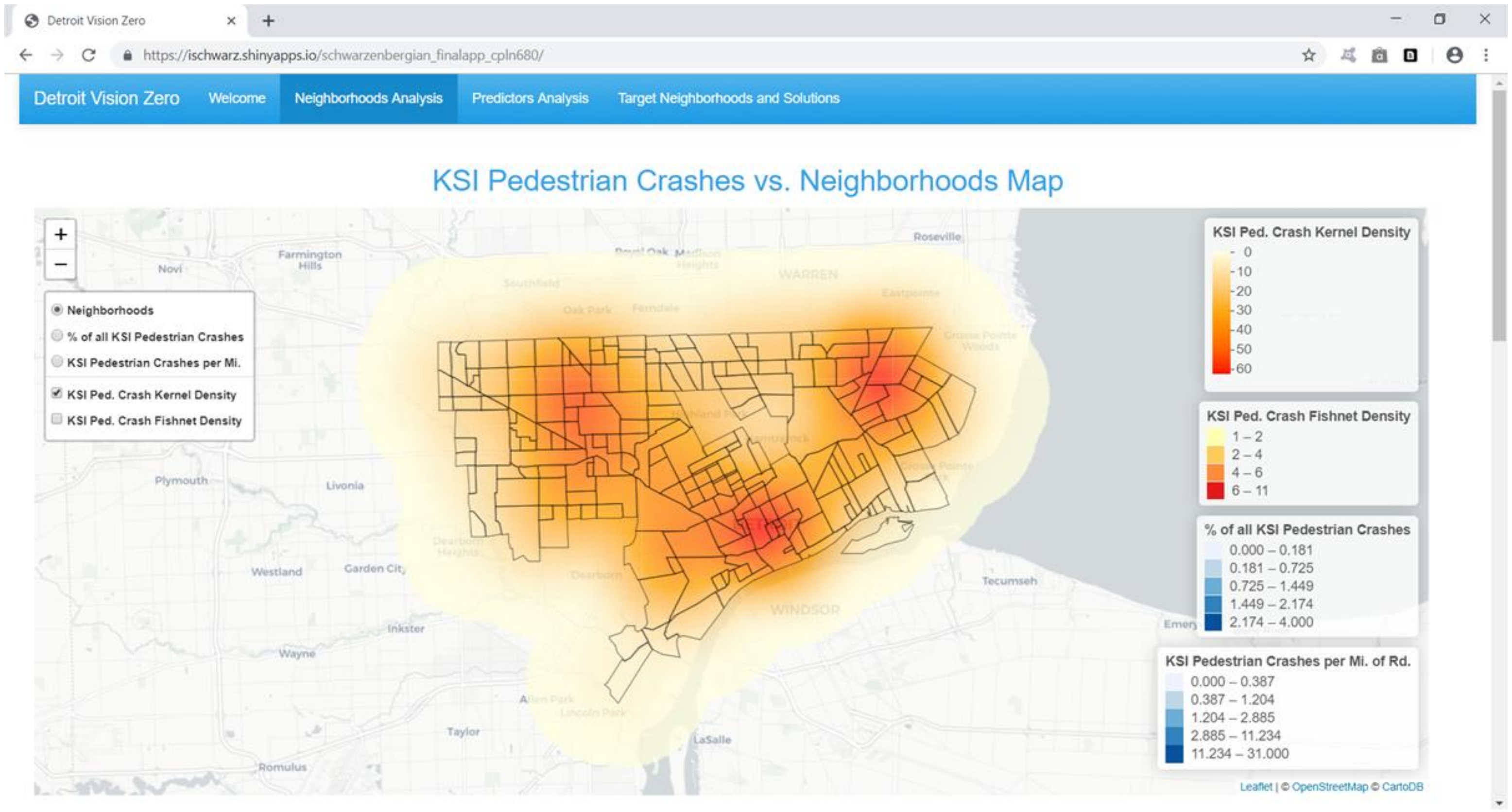
## Data Sources

- Youth-only trip generators:
  - Schools: https://data.detroitmi.gov/Education/All-Schools-2017-2018/wn8n-5a4d
  - Childcare providers: https://data.detroitmi.gov/Children-Families/Child-Care-Providers/9ssd-ypf9
- Older-adult-only trip generators:
  - Senior centers: https://www.seniorcitizensguide.com/detroit/listings/seniorcenters.htm
  - Nursing homes: https://www.dibbern.com/nursing-homes/michigan/detroit-nursing-homes-directory.htm
- General trip generators:
  - Factories, bars, arcades, pool halls, bowling alleys, restaurants, movie theaters and stores: https://data.detroitmi.gov/Business/Business-Licenses/pugj-2dh4
  - Office buildings and factory locations: https://data.detroitmi.gov/Property-Parcels/Parcel-Points-Ownership/dxgi-9s8s

## Data Sources

- Health clinics:
  - Hospitals: https://data.detroitmi.gov/Public-Health/Hospitals/9irz-u76s
  - Health centers: https://data.detroitmi.gov/Public-Health/Federally-Qualified-Health-Centers/uiy2-dk3s
- Recreation centers: https://d3-d3.opendata.QGIS.com/datasets/8ee240e505cb493eaa63cf0c5c390d34_0
- Places of worship: https://d3-d3.opendata.QGIS.com/datasets/f6c93d04857c4bd4b0783867def3ccf5_0

# First Tab after Welcome Page: Neighborhoods Analysis



[✓ Neighborhoods]
% of all KSI Pedestrian Crashes
KSI Pedestrian Crashes per Mi.
[✓ KSI Ped. Crash Kernel Density]
KSI Ped. Crash Fishnet Density

## Tab Purpose

The point of this tab is to accomplish the first goal of the app, to show the user which neighborhoods are experiencing the "most" KSI pedestrian crashes, and therefore should be considered for future prioritization of Vision Zero resources. This application defines a neighborhood having the "most" KSI pedestrian crashes in 2 different ways:

1) High percentage of all KSI pedestrian crashes happening within that neighborhood from 2011-16
2) High rate of KSI pedestrian crashes per mile of roadway in that neighborhood

Through the interactive map and pre-made graphs on the tab, the user can see which neighborhoods consistently come up as being notable in terms of both categories. This is how the neighborhoods for further consideration are selected. I originally tried using Detroit city council districts and groups of those known as design regions, [4] but ended up deciding to use neighborhoods for two reasons. Firstly, neighborhoods are more fine-grained than city council districts and design regions, allowing city staff to pinpoint on hotspots more. Secondly, neighborhoods could be more commonly known among Detroiters as opposed to their city council districts or design regions.

[4] https://detroitmi.gov/departments/planning-and-development-department

## Neighborhood Map Creation Process

The first part of this tab shows the KSI Pedestrian Crashes vs. Neighborhoods Map. The point of this map is to show the user which neighborhoods experienced the highest shares of concentrations of KSI pedestrian crashes from 2011-16 and highest KSI pedestrian crash rates per mile of road. This involved 2 different general workflows in the order specified below:

1) Spatially joining all KSI pedestrian crashes from 2011-16 to each neighborhood, and dividing each neighborhood's amount by the total number of crashes during those years
2) Clipping Michigan roads to Detroit, spatially joining those to each neighborhood to find each one's total miles of roads, and dividing each neighborhood's KSI pedestrian crash amount by its total road mileage to get its KSI pedestrian crash rate per mile

I wanted to also measure each neighborhood's KSI pedestrian crash rate because just analyzing each neighborhood's percentage of all KSI pedestrian crashes alone could ignore whether or not those neighborhoods have many roads. For example, a neighborhood with a high number or percentage of crashes could just be like that because it has a high amount of roads inside it. This map thereby allows the user to compare which neighborhoods just have high amounts of crashes versus neighborhoods which have both high amounts of crashes and less roadway.

# Neighborhood Map Creation Process

```
library(sf)

setwd("C:/Users/Ian/Documents/Documents/University_Of_
Pennsylvania/Courses/CPLN_680_AdvancedTopicsGIS/Data")

crashes <- read.csv("Traffic_Crashes.csv")

crashes$Lon <- gsub("location", "", crashes$location)
crashes$Lon <- gsub(".*,", "", crashes$Lon)
crashes$Lon <- gsub("\\)", "", crashes$Lon)
crashes$Lon <- as.numeric(crashes$Lon)
crashes$Lat <- gsub("location", "", crashes$location)
crashes$Lat <- gsub(",.*", "", crashes$Lat)
crashes$Lat <- gsub("\\(", "", crashes$Lat)
crashes$Lat <- as.numeric(crashes$Lat)
```

To make this map, I first created a shapefile out of my original Detroit crashes CSV in R. This involved first creating individual columns for each crash's longitude and latitude coordinates out of the original combined coordinates column that came with the CSV called "location". R basically cannot make a CSV into a shapefile without specifically defined individual latitude and longitude columns, which made me do this. I did this using R's native gsub() command which replaces specified text with new one, similar to using Ctrl + F to replace text in Windows. This series of lines goes through the "location" column, looks for specific text in it that I do not want, gets rid of it without modifying the original "location" column and puts the replacements in my new individual latitude and longitude columns.

## Neighborhood Map Creation Process

```
pedestrian_crashes <- crashes[
which(crashes$Pedestrian.=='1'), ]

fatal_pedestrian_crashes <- pedestrian_crashes[
which(pedestrian_crashes$K.Level.Fatal.Injuries > 0), ]

severe_pedestrian_crashes <- pedestrian_crashes[
which(pedestrian_crashes$A.Level.Injuries > 0), ]

KSI_pedestrian_crashes <- rbind(fatal_pedestrian_crashes,
severe_pedestrian_crashes)


KSI_pedestrian_crashes_SHP <-
st_as_sf(KSI_pedestrian_crashes, coords=c("Lon","Lat"),
crs=st_crs("+init=epsg:4326"))

st_write(KSI_pedestrian_crashes_SHP,
"KSI_pedestrian_crashes.shp", delete_layer=TRUE))
```

After individual latitude and longitude columns were made, I filtered out which crashes involved pedestrians in the CSV, and which were classified as resulting in severe injuries or deaths based on the CSV's metadata. I then created a new shapefile from these filtered results (crashes that were just KSI and involved a pedestrian) using the st_as_sf() command from R's sf package. That line takes my filtered results which are just a non-spatial data frame at this point, creates a simple features geometry column using the individual latitude and longitude columns I created in the previous step, gives it a coordinate system, and thereby makes it into a shapefile. I then exported this shapefile out of R using sf's st_write command for later analysis.

## Neighborhood Map Creation Process

```
neighborhoods <- st_read("geo_export_ec48e3a3-31d4-4ae4-a1fd-a72a3b8c39a8.shp")

neighborhoods <- st_transform(neighborhoods, 4326)


neighborhoods_copy <- st_read("geo_export_ec48e3a3-31d4-4ae4-a1fd-a72a3b8c39a8.shp")

neighborhoods_copy <- st_transform(neighborhoods_copy, 4326)


SpatialJoinOutput <- sapply(st_within(KSI_pedestrian_crashes_SHP, neighborhoods_copy), function(z) if (length(z)==0) NA_integer_ else z[1])

KSI_pedestrian_crashes_SHP$Nghbrhd <- SpatialJoinOutput
```

I then went about spatially joining all KSI pedestrian crashes to each neighborhood. This first involved making a copy of the neighborhoods shapefile to do all of the analysis on in case something goes wrong, and then joining the new column results from that back to the original neighborhoods shapefile. The spatial join is done through a series of lines which ask each KSI pedestrian crash what district it is in. If it is in a neighborhood, give it its neighborhood ID number, but if it is no neighborhood for any reason, give it a value of NA. The resulting set of neighborhood number values (“SpatialJoinOutput”) that gets created from the spatial join then gets added to the KSI pedestrian crashes shapefile as a new column showing each crash’s neighborhood.

## Neighborhood Map Creation Process

```
Nghbrhdricts_CrashCount <-
data.frame(table(KSI_pedestrian_crashes_SHP$Nghbrhd))

colnames(Nghbrhdricts_CrashCount) <- c("neighborho",
"KSIPdCrshs")


neighborhoods_copy <- merge(x=neighborhoods_copy,
y=Nghbrhdricts_CrashCount, by="neighborho", all.x=TRUE)

neighborhoods_copy$KSIPdCrshs[is.na(neighborhoods_copy$
KSIPdCrshs)] <- 0


neighborhoods$KSIPdCrshs <-
neighborhoods_copy$KSIPdCrshs

neighborhoods$KSIPCrshPc <-
(neighborhoods$KSIPdCrshs/sum(neighborhoods$KSIPdCrshs)
)*100
```

The table () command is then used to make a new data frame called "Nghbrhdricts_CrashCount" which summarizes each district by its sum of KSI pedestrian crashes from the spatial join in the previous step. This data frame is then tabular joined back to the neighborhoods shapefile copy to show each neighborhood's number of KSI pedestrian crashes. This column is then simply moved to the original neighborhoods shapefile.

I then created a new column in the original neighborhoods shapefile dividing each neighborhood's amount of KSI pedestrian crashes by the total number of crashes during those years to get each neighborhood's percent of all KSI pedestrian crashes from 2011-16.

## Neighborhood Map Creation Process

```
neighborhoods_DD <- st_transform(neighborhoods, 32610)
neighborhoods_buffer <- st_buffer(neighborhoods_DD, -5)

neighborhoods_buffer <-
st_transform(neighborhoods_buffer, 4326)


michigan_roads <- st_read("michigan_roads.shp")


roads_Detroit_SHP <- st_intersection(michigan_roads,
neighborhoods_buffer)


roads_Detroit_SHP <- st_transform(roads_Detroit_SHP, 4326)
```

At this point, I now have each neighborhood's number and share of KSI pedestrian crashes. To additionally find each neighborhood's total road mileage and KSI pedestrian crashes per mile of road, I first used R's sf package to create a version of the neighborhoods shapefile for clipping. This involved making a decimal degrees version of the original neighborhoods shapefile, making an internal buffer on that to select all neighborhoods, and then transforming that back to the same projection as my roads shapefile.

I then clipped the Michigan roads with the version of the neighborhoods shapefile for clipping to get my Detroit roads shapefile using sf's st_intersection command.

## Neighborhood Map Creation Process

```
SpatialJoinOutput <- sapply(st_within(roads_Detroit_SHP,
neighborhoods), function(z) if (length(z)==0) NA_integer_
else z[1])

SpatialJoinOutput[is.na(SpatialJoinOutput)] <- 0

roads_Detroit_SHP$Nghbrhd <- SpatialJoinOutput


roads_Detroit_SHP$length_m <-
st_length(roads_Detroit_SHP)

roads_Detroit_SHP$length_mi <-
as.numeric(roads_Detroit_SHP$length_m) * 0.00062137


Nghbrhds_byRdMi <- data.frame(aggregate(

  roads_Detroit_SHP$length_mi,

  by=list(roads_Detroit_SHP$Nghbrhd),

  FUN=sum))
```

I then spatially joined all Detroit roads to each neighborhood using the exact same spatial join process as before, except I gave any road not in a neighborhood (roads on borders between them) as a neighborhood ID number of 0. I now have each Detroit road and the neighborhood is in.

I then used the sf package's st_length command to find each Detroit road's length. This came out in meters because of its projection system, so I then made a new column converting that length in meters column to length in feet.

I then created a new data frame summarizing each neighborhood by its total (summed) miles of roads using R's native aggregate() command.

# Neighborhood Map Creation Process

```
colnames(Nghbrhds_byRdMi) = c("neighborho", "ttl_rd_mi")

neighborhoods <- merge(x=neighborhoods,
y=Nghbrhds_byRdMi, by="neighborho", keep.all=FALSE)

neighborhoods$KSIPCrshPc <-
(neighborhoods$KSIPdCrshs/sum(neighborhoods$KSIPdCrshs)
)*100

neighborhoods$KSIPCrPrMi <-
neighborhoods$KSIPdCrshs/neighborhoods$ttl_rd_mi
```

I then tabular joined the non-spatial data frame I made in the previous step detailing each neighborhood by its total road mileage with the original neighborhoods shapefile, to get a new neighborhoods shapefile with the “ttl_rd_mi” column detailing each neighborhood’s number of roads. I had to change the column names of the non-spatial data frame from the previous step to get the tabular join to work.

At this point, the neighborhoods shapefile now has a number of KSI pedestrian crashes column and total road mileage column. I used these to create a new column in the neighborhoods shapefile dividing each neighborhood’s number of crashes by its total road mileage to get the final KSI pedestrian crashes per mile column to map. I also created the share of KSI crashes column here.

## Neighborhood Map Creation Process

```
#Made kernel density of KSI pedestrian crashes in QGIS

#Fishnet creation:

fishnet_grid <- st_make_grid(neighborhoods_SHP, cellsize = 0.0125, square = TRUE)

fishnet_Detroit_SHP <- st_intersection(fishnet_grid, neighborhoods_buffer)

st_write(fishnet_Detroit_SHP, "fishnet_Detroit_Geometries.shp", delete_layer=TRUE)
#Exported as fishnet geometry, made this into regular polygon SHP in QGIS

#Then spatial joined KSI pedestrian crashes to fishnet polygon SHP in QGIS
```

For 2 of the application's maps, I also had to create a kernel density raster of the KSI pedestrian crashes and a fishnet density polygon shapefile of those crashes. I simply made the kernel density raster of them in QGIS using the Heatmap (Kernel Density Estimation) tool. However, I first made a fishnet grid geometry of Detroit in R using sf's st_make_grid command. That creates the grid inside the version of the neighborhoods shapefile used for clips. The cell size is the average area of a Detroit census tract, which I found by using the Basic Summary Statistics tool in QGIS on the Detroit census tracts shapefile. This shows density of a finer grain within the city. I then exported the fishnet geometry made in R to QGIS, which I then used to export it as a shapefile and joined the KSI pedestrian crashes to using the Join Attributes by Location tool there.

# Neighborhood Map Creation Process

```
val_ksi_ped_crash_KD = as.numeric(seq(0,62,1))

 pal_ksi_ped_crash_KD = colorNumeric(c("lightyellow",
"orange", "red"), val_ksi_ped_crash_KD, na.color =
"transparent")


bins_ksi_ped_crash_FN <- c(1,2,4,6,11) pal_ksi_ped_crash_FN
<- colorBin("YlOrRd", domain = ksi_ped_crash_FN$Crsh_ID_un,
bins = bins_ksi_ped_crash_FN, na.color = "transparent")


bins_KSIPCrshPc <- c(0, 0.1812, 0.7246, 1.4493, 2.1739, 4)
pal_KSIPCrshPc <- colorBin("Blues", domain =
neighborhoods$KSIPCrshPc, bins = bins_KSIPCrshPc, na.color =
"transparent")


bins_KSIPCrPrMi <- c(0, 0.3873, 1.2041, 2.8847, 11.2337, 31)

pal_KSIPCrPrMi <- colorBin("Blues", domain =
neighborhoods$KSIPCrPrMi, bins = bins_KSIPCrPrMi, na.color =
"transparent")
```

At this point, I now had all the data necessary to make my neighborhoods map. However, I had to first set the graphic design guidelines of how all the map's data would be displayed. This involved creating values and palette objects for each of the map's 4 layers in R. It involved using QGIS to see the 5 natural break values behind the neighborhood shapefile's KSI crash share and rate per mile columns and the KSI pedestrian crash fishnet density. To find the range of values for the raster, I briefly ran the "ksi_ped_crash_KD@data@min" and "ksi_ped_crash_KD@data@max" R commands.

## Neighborhood Map Creation Process

```
output$NeighborhoodsMap <- renderLeaflet({
   leaflet() %>% addProviderTiles("CartoDB.Positron") %>%
   setView(lng = -83.081659, lat = 42.360304, zoom = 10.5)
%>%
addRasterImage(x = ksi_ped_crash_KD, colors =
pal_ksi_ped_crash_KD,  opacity = 0.75, group = "KSI Ped.
Crash Kernel Density", project=FALSE) %>%
addLegend(pal = pal_ksi_ped_crash_KD, values =
val_ksi_ped_crash_KD,  title = "KSI Ped. Crash Kernel
Density", opacity = 1) %>%
addPolygons(data = ksi_ped_crash_FN, group = "KSI Ped.
Crash Fishnet Density", fillColor =
~pal_ksi_ped_crash_FN(as.numeric(ksi_ped_crash_FN$Crsh_
ID_un)),  weight = 0.5, opacity = 1.0, color = "#9B9B9C",
fillOpacity = 0.75) %>%
addLegend(position = "topright", pal =
pal_ksi_ped_crash_FN, values = bins_ksi_ped_crash_FN,  title
= "KSI Ped. Crash Fishnet Density", opacity = 1) %>%
```

To finally make the map, I then went to the app.R script and created the map using the R shiny package's renderLeaflet({}) command. It first adds the simple base map, then sets the center and zoom to Detroit. It then uses addRasterImage() from R's leaflet package to add the KSI pedestrian crash kernel density to the map using the specified guidelines and "project=FALSE" to make sure all values are mapped, and makes a legend for it . It then adds the KSI pedestrian crash fishnet density polygons to the map also under the specified graphic design guidelines from the previous step, and makes a legend for it. All legends were put on the screen and given appropriate titles. Except for the kernel density, blue was usually used for the color coding to keep with the app's theme.

## Neighborhood Map Creation Process

```
addPolygons(data = neighborhoods, group = "Neighborhoods", color = "#000000", weight = 0.5, opacity = 1.0, fillOpacity = 0, popup= paste0("<strong>Neighborhood: </strong>", neighborhoods$nhood_name)) %>%
```

```
addPolygons(data = neighborhoods, fillColor = ~pal_KSIPCrshPc(as.numeric(neighborhoods$KSIPCrshPc)), weight = 0.5, opacity = 1.0, color = "#000000", fillOpacity = 0.75, group = "% of all KSI Pedestrian Crashes", popup= paste0("<strong>Neighborhood: </strong>", neighborhoods$nhood_name, "<br>", "<strong>% of all KSI Ped. Crashes: </strong>", neighborhoods$KSIPCrshPc)) %>%
```

```
addPolygons(data = neighborhoods, fillColor = ~pal_KSIPCrPrMi(as.numeric(neighborhoods$KSIPCrPrMi)), weight = 0.5, opacity = 1.0, color = "#000000", fillOpacity = 0.75, group = "KSI Pedestrian Crashes per Mi.", popup= paste0("<strong>Neighborhood: </strong>", neighborhoods$nhood_name, "<br>", "<strong>KSI Ped. Crashes per Mi. of Rd.: </strong>", neighborhoods$KSIPCrPrMi)) %>%
```

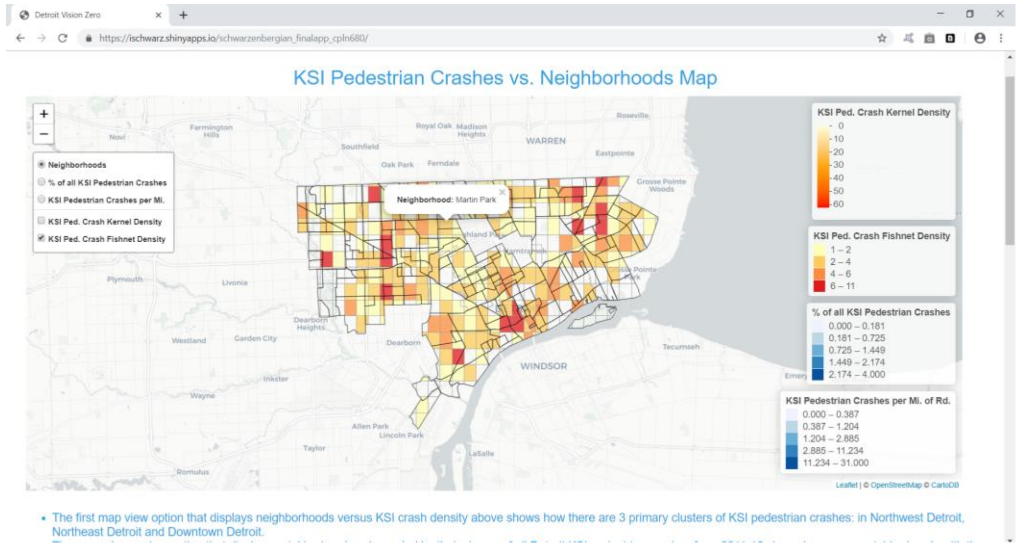
These 3 commands were especially important in making the map. They add the same neighborhood dataset 3 times, but each time they map the dataset in a different way. The first addPolygons() leaflet package command just maps the neighborhood boundaries without any fill. The point of this layer is to simply let the user compare the crash kernel and fishnet densities to just the districts to see which districts overlap with the crash concentrations. The other 2 addPolygons() commands map the crash share and rate per mile columns. All 3 commands utilize the popup= feature which allows the user view information about each district by clicking on them in addition to just viewing them on the map. All of the popup commands also make use of HTML elements such as <strong> (bolding) and <br> (making new lines) to organize the popups.

## Neighborhood Map Creation Process

```
hideGroup("KSI Ped. Crash Fishnet Density") %>%
   addLayersControl(
    baseGroups = c("Neighborhoods", "% of all KSI Pedestrian Crashes", "KSI Pedestrian Crashes per Mi."),
    overlayGroups = c("KSI Ped. Crash Kernel Density", "KSI Ped. Crash Fishnet Density"),
    position = c("topleft"),
    options = layersControlOptions(collapsed=FALSE))


 })
```

The hideGroup() command makes it so the fishnet density layer is unchecked by default, but the kernel density raster comes up by default. The addLayersControl() command is extremely important, since this organizes the map layers into radio buttons and check marks which the user can click through on the map. The baseGroups are the radio button layers which can be rotated through but at least one will always show in some way, and the overlayGroups are layers that can be checked on and off to compare to the baseGroups. Throughout the application, the overlayGroups are always the crash density layers and the baseGroups are what is to be compared with them. Each layer is identified by its group name, as shown in the previous page. Those make the layers referable so legends and layers controls can be made of them. The options line makes it so the radio buttons always show up and cannot be hidden.

# Final Neighborhoods Map Shots


Detroit Vision Zero

https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/

KSI Pedestrian Crashes vs. Neighborhoods Map

[✓ Neighborhoods]
% of all KSI Pedestrian Crashes
KSI Pedestrian Crashes per Mi.
KSI Ped. Crash Kernel Density
[✓ KSI Ped. Crash Fishnet Density]

Neighborhood: Martin Park

KSI Ped. Crash Kernel Density: 0, 10, 20, 30, 40, 50, 60

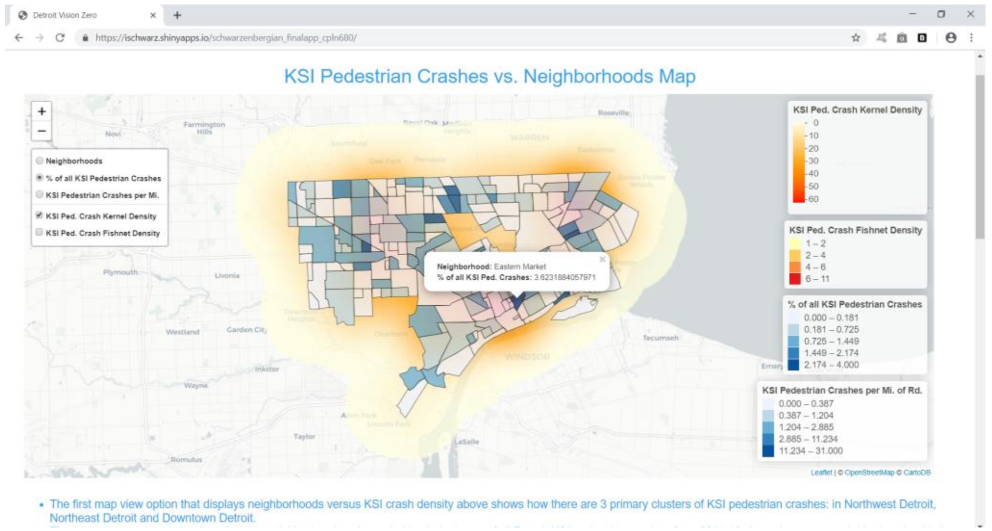
| KSI Ped. Crash Fishnet Density |
|---|
| 1 – 2 |
| 2 – 4 |
| 4 – 6 |
| 6 – 11 |

| % of all KSI Pedestrian Crashes |
|---|
| 0.000 – 0.181 |
| 0.181 – 0.725 |
| 0.725 – 1.449 |
| 1.449 – 2.174 |
| 2.174 – 4.000 |

| KSI Pedestrian Crashes per Mi. of Rd. |
|---|
| 0.000 – 0.387 |
| 0.387 – 1.204 |
| 1.204 – 2.885 |
| 2.885 – 11.234 |
| 11.234 – 31.000 |

Leaflet | © OpenStreetMap © CartoDB

- The first map view option that displays neighborhoods versus KSI crash density above shows how there are 3 primary clusters of KSI pedestrian crashes: in Northwest Detroit, Northeast Detroit and Downtown Detroit.


Detroit Vision Zero

https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/

KSI Pedestrian Crashes vs. Neighborhoods Map

Neighborhoods
[✓ % of all KSI Pedestrian Crashes]
KSI Pedestrian Crashes per Mi.
[✓ KSI Ped. Crash Kernel Density]
KSI Ped. Crash Fishnet Density

Neighborhood: Eastern Market
% of all KSI Ped. Crashes: 3.6231884057971

KSI Ped. Crash Kernel Density: 0, 10, 20, 30, 40, 50, 60

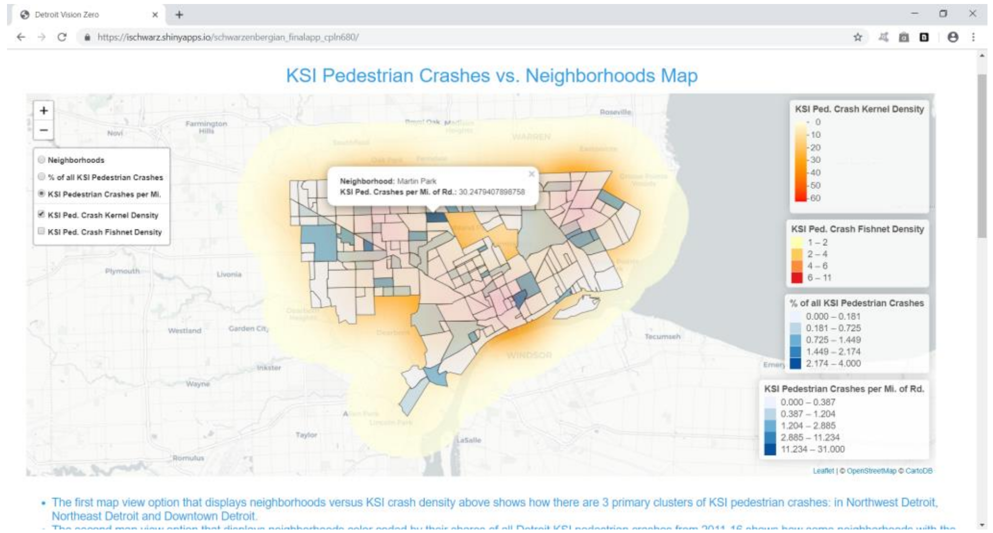
| KSI Ped. Crash Fishnet Density |
|---|
| 1 – 2 |
| 2 – 4 |
| 4 – 6 |
| 6 – 11 |

| % of all KSI Pedestrian Crashes |
|---|
| 0.000 – 0.181 |
| 0.181 – 0.725 |
| 0.725 – 1.449 |
| 1.449 – 2.174 |
| 2.174 – 4.000 |

| KSI Pedestrian Crashes per Mi. of Rd. |
|---|
| 0.000 – 0.387 |
| 0.387 – 1.204 |
| 1.204 – 2.885 |
| 2.885 – 11.234 |
| 11.234 – 31.000 |

Leaflet | © OpenStreetMap © CartoDB

- The first map view option that displays neighborhoods versus KSI crash density above shows how there are 3 primary clusters of KSI pedestrian crashes: in Northwest Detroit, Northeast Detroit and Downtown Detroit.


Detroit Vision Zero

https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/

KSI Pedestrian Crashes vs. Neighborhoods Map

Neighborhoods
% of all KSI Pedestrian Crashes
[✓ KSI Pedestrian Crashes per Mi.]
[✓ KSI Ped. Crash Kernel Density]
KSI Ped. Crash Fishnet Density

Neighborhood: Martin Park
KSI Ped. Crashes per Mi. of Rd.: 30.2479407898758

KSI Ped. Crash Kernel Density: 0, 10, 20, 30, 40, 50, 60

| KSI Ped. Crash Fishnet Density |
|---|
| 1 – 2 |
| 2 – 4 |
| 4 – 6 |
| 6 – 11 |

| % of all KSI Pedestrian Crashes |
|---|
| 0.000 – 0.181 |
| 0.181 – 0.725 |
| 0.725 – 1.449 |
| 1.449 – 2.174 |
| 2.174 – 4.000 |

| KSI Pedestrian Crashes per Mi. of Rd. |
|---|
| 0.000 – 0.387 |
| 0.387 – 1.204 |
| 1.204 – 2.885 |
| 2.885 – 11.234 |
| 11.234 – 31.000 |

Leaflet | © OpenStreetMap © CartoDB

- The first map view option that displays neighborhoods versus KSI crash density above shows how there are 3 primary clusters of KSI pedestrian crashes: in Northwest Detroit, Northeast Detroit and Downtown Detroit.

As the map's options are clicked through and analyzed, it is apparent that a handful of neighborhoods constantly show up as being notable across categories, such as in these shots.

# Pre-Made Graphs Creation Process

```
library(ggplot2)


topshare <- neighborhoods[order(-neighborhoods$KSIPCrshPc),]

topshare <- topshare[1:10,]

topshare <- topshare[c("nhood_name", "KSIPCrshPc")]

topshare$BarColor <- c("#E7B22F", "#E7B22F", "#E7B22F", "#E7B22F", "#2FA4E7", "#2FA4E7", "#2FA4E7", "#2FA4E7", "#2FA4E7", "#2FA4E7")

st_geometry(topshare) <- NULL


ggplot(topshare, aes(reorder(nhood_name, -KSIPCrshPc), (KSIPCrshPc))) + geom_bar(stat="identity", fill=topshare$BarColor) +theme_light()+labs(title="Top 10 Detroit Neighborhoods by Share of KSI Pedestrian Crashes, 2011-16", x="Neighborhood", y="% of Detroit KSI Pedestrian Crashes, 2011-16") + theme(axis.text.x = element_text(angle = 90))
```

The user can then view pre-made graphs further analyzing what neighborhoods come up as being notable across the categories in the map above it. To make the graph showing the top neighborhoods by share of crashes, I first made a new data frame called “topshare” which first reordered all neighborhoods by share of KSI pedestrian crashes, then takes the top 10 of them and only extracts the columns I want. It then makes a new column in it detailing what colors the bars will be. I use this to highlight how the top 4 neighborhoods for this category have notably higher shares than the rest of the neighborhoods. The “topshare” data frame then has its geometry removed from it so it can be graphed. The graph is then made using the ggplot package. The ggplot() command notably make it so the highest bars appear first and are color-coded by the bar color column created before.

## Pre-Made Graphs Creation Process

```
topshare <- neighborhoods[order(-neighborhoods$KSIPCrPrMi),]

topshare <- topshare[1:5,]

topshare <- topshare[c("nhood_name", "KSIPCrPrMi")]

topshare$BarColor <- c("#E7B22F", "#E7B22F", "#2FA4E7", "#2FA4E7", "#DCE72F")

st_geometry(topshare) <- NULL


ggplot(topshare, aes(reorder(nhood_name, -KSIPCrPrMi), (KSIPCrPrMi))) + geom_bar(stat="identity", fill=topshare$BarColor) +theme_light()+labs(title="Top 5 Detroit Neighborhoods by KSI Pedestrian Crash Rate per Mile of Road, 2011-16", x="Neighborhood", y="KSI Pedestrian Crash Rate per Mile") + theme(axis.text.x = element_text(angle = 90))
```

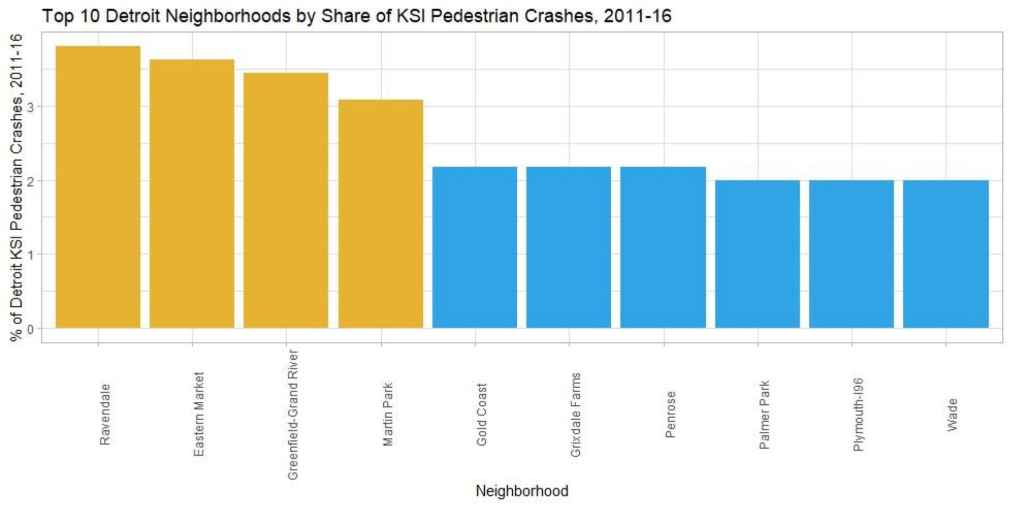
The code for the creation of the graph for neighborhood by crash rate per mile follows the exact same template as for the neighborhood shares graph, except this graph looks at the top 5 neighborhoods as opposed to the top 10 with the other due to the differing distributions of the crash rate data among the columns compared to the share column.

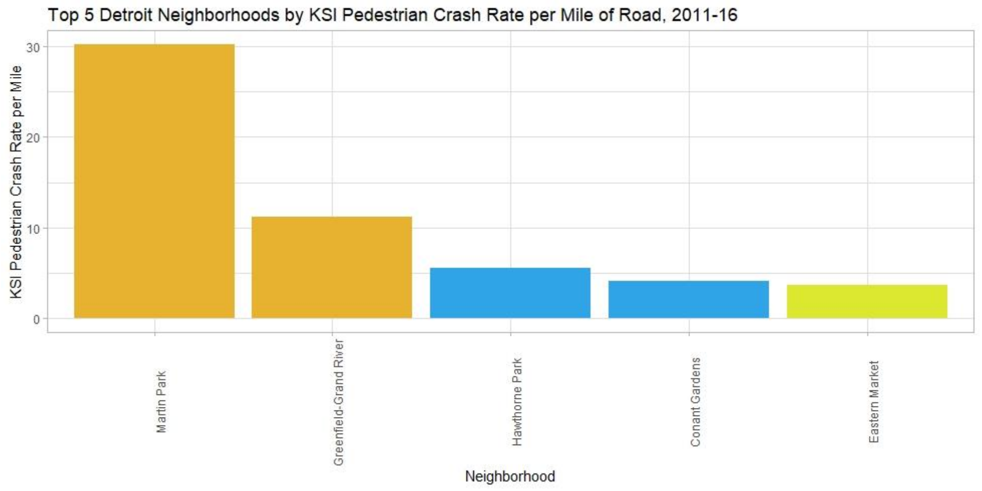
## Pre-Made Graphs Creation Process

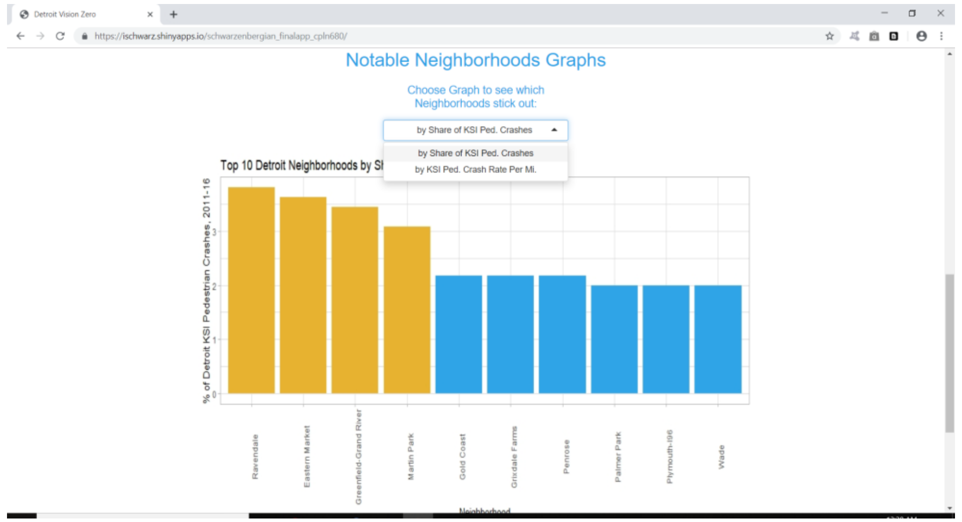
```
output$NeighborhoodsGraph <- renderUI({


  if(input$NeighborhoodsGraphChoice == "KSIPCrshPc_Graph"){
    img(height = 600, width = 900, src = "KSIPCrshPc_Graph.jpeg")
  }
  else if(input$NeighborhoodsGraphChoice == "KSIPCrPrMi_Graph"){
    img(height = 600, width = 900, src = "KSIPCrPrMi_Graph.jpeg")
  }


})
```

The code for the app is then created in the server part which uses a series of if statements to bring up the graph stored in the app's www folder that the user wants to see. This basically says "if the user chooses to see the crash shares graph in the user interface, pull up the corresponding graph image in the www folder and display it in the app using the specified height and width guidelines. If the user chooses to see the crashes per mile of road graph, replicate the process".

# Final Neighborhoods Graphs






Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/

Notable Neighborhoods Graphs

Choose Graph to see which Neighborhoods stick out:

by Share of KSI Ped. Crashes

by Share of KSI Ped. Crashes
by KSI Ped. Crash Rate Per Mi.
Top 10 Detroit Neighborhoods by Sh
% of Detroit KSI Pedestrian Crashes, 2011-16
0
1
2
3
Ravendale
Eastern Market
Greenfield-Grand River
Martin Park
Gold Coast
Grixdale Farms
Penrose
Palmer Park
Plymouth-I96
Wade
Neighborhood

As the graphs are viewed from the drop-down menu, it is apparent that many of the neighborhoods that are highlighted tend to be the same across both graphs. Eastern Market is highlighted in yellow in the crash rate graph because even though it does not have a notably higher crash rate than other neighborhoods, it still comes up as being in at least the top 5 for both.

## User Interface Creation Process

```
tabPanel("Neighborhoods Analysis", tags$head(tags$style("headerPanel {color: #2FA4E7; } h2 {color: #2FA4E7; } h4 {color: #2FA4E7}; }")),

fluidRow(column(12, align="center", h2("KSI Pedestrian Crashes vs. Neighborhoods Map"))),

leafletOutput("NeighborhoodsMap", width = 1425, height = 600), br(), h4(tags$ul(tags$li(“Description”), tags$li(“Description”))), br(), br(),

fluidRow(column(12, align="center", h2("Notable Neighborhoods Graphs"), selectInput("NeighborhoodsGraphChoice", h4("Choose Graph to see which Neighborhoods stick out:"), choices = c("by Share of KSI Ped. Crashes" = "KSIPCrshPc_Graph", "by KSI Ped. Crash Rate Per Mi." = "KSIPCrPrMi_Graph")), uiOutput("NeighborhoodsGraph"))), br(), h4(tags$ul(tags$li(“Description”), tags$li(“Description”))), br(), br(),))),
```

This code creates the tab in the UI panel with “Description” in place of what is actually written for the bullet points describing the maps and graphs for this tab. It first creates the tab using tabPanel by titling the tab and setting the graphic design guidelines for it. It then uses fluidRow() to guide the proportioning of all the page’s elements, uses leafletOutput() to output the map created in the server, and uses tags$ul and tags$li to make the bullet points. selectInput() is used to make the drop-down menu allowing users to choose which graph to see. The clean drop down menu name equals the file name in the app’s www folder.

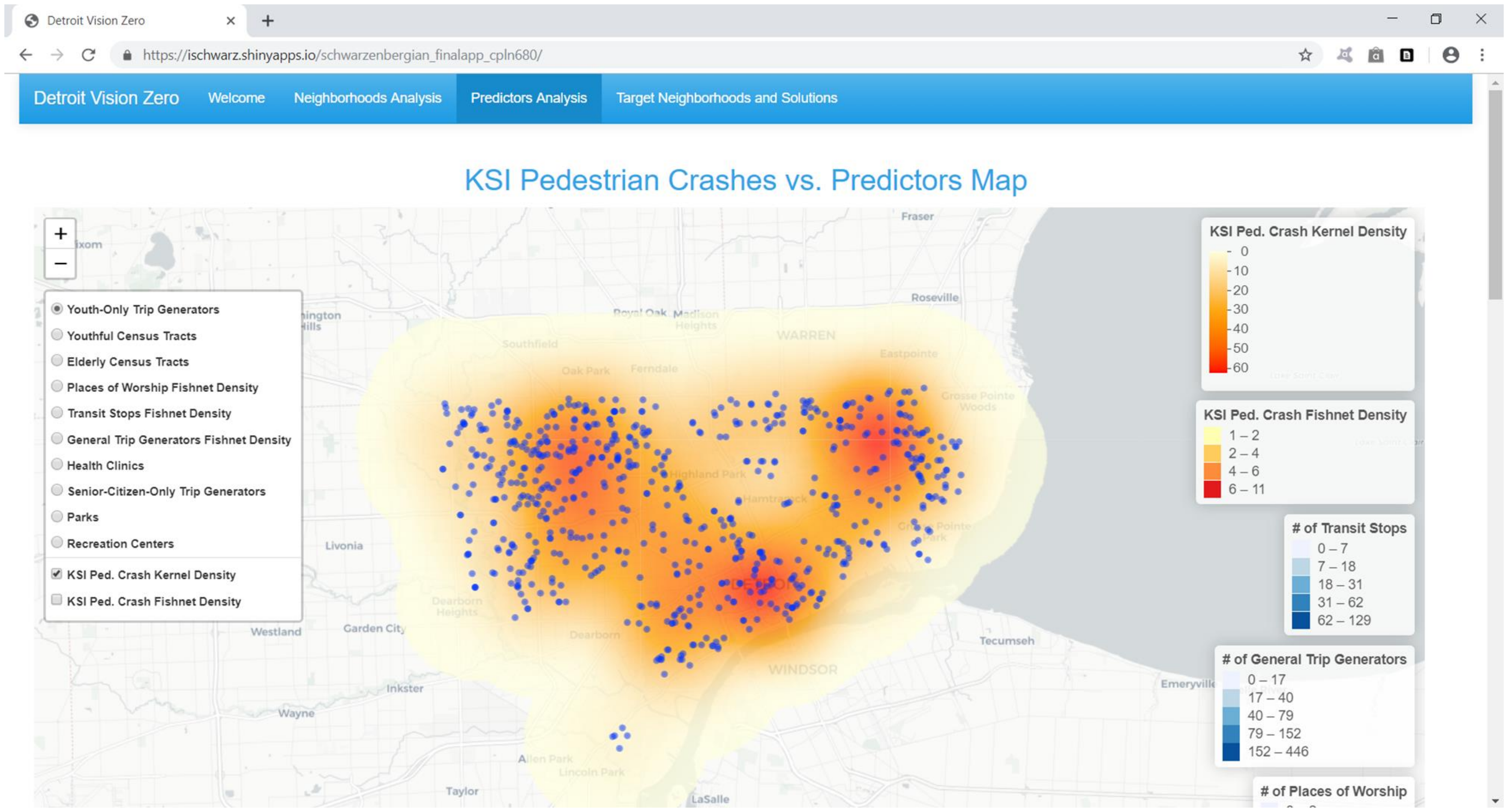
## Analysis Results

There are four neighborhoods which Detroit officials should especially consider prioritizing pedestrian safety solutions if it decides to undertake Vision Zero in the future:

1) Ravendale
2) Eastern Market
3) Greenfield-Grand River
4) Martin Park

These were selected because of how they either come up as notable neighborhoods across both measures used determining which neighborhoods have the 'most' KSI pedestrian crashes, or they come up as having the highest in either category. For example, those four neighborhoods show in both graphs, and come up across the layers in the map.

# Second Tab after Welcome Page: Predictors Analysis



# Tab Purpose

The point of this tab is to accomplish the second goal of the app, to show the user which predictors have the strongest associations with KSI crash clusters, and therefore should be considered when considering what types of solutions to implement in the most problematic neighborhoods determined in the previous tab. This application defines a predictor having the "strongest" associations with KSI pedestrian crashes in 3 different ways:

1) High visual correlation (strength of how much a predictor's clusters overlaps with KSI pedestrian crash clusters)
2) High share of KSI pedestrian crashes happening near the predictor's locations compared to other predictors
3) High correlation coefficient with KSI pedestrian crashes

Through the interactive map, graph and regression table on the tab, the user can see which predictors consistently come up as having strong associations with KSI pedestrian crashes. This is how the predictors for further consideration are selected.

## Predictors List

Many different types of predictors were used, but were grouped into 10 general categories for simplicity:

- Transit stops
- Parks
- Youthful Tracts
- Elderly Tracts
- Youth-only Trip Generators (Schools, Childcare Providers)
- Older-Adult-only Trip Generators (Senior Centers, Nursing Homes)
- General Trip Generators (factories, bars, arcades, pool halls, bowling alleys, restaurants, movie theaters, stores, office buildings)
- Health Clinics
- Recreation Centers
- Places of Worship

# Predictors Creation Process

```
census_tracts <- st_read("tl_2018_26_tract.shp")

census_tracts <- st_transform(census_tracts, 4326)

census_tracts_new <- st_as_sf(census_tracts %>% distinct)

census_tracts_Detroit_SHP <-st_intersection(census_tracts_new,
neighborhoods_buffer) st_write(census_tracts_Detroit_SHP,
"census_tracts_Detroit_SHP.shp", delete_layer=TRUE)

census_tracts_Detroit_SHP_with_ages <-
st_read("census_tracts_Detroit_SHP_with_ages.shp")

census_tracts_Detroit_SHP_with_ages$PCTUND18 <-
as.character(census_tracts_Detroit_SHP_with_ages$census_t_3)

census_tracts_Detroit_SHP_with_ages$PCTUND18 <-
as.numeric(census_tracts_Detroit_SHP_with_ages$PCTUND18)

young_tracts_Detroit_SHP <-
st_as_sf(census_tracts_Detroit_SHP_with_ages[
which(census_tracts_Detroit_SHP_with_ages$PCTUND18 >
mean(as.numeric(census_tracts_Detroit_SHP_with_ages$PCTUN
D18))), ])

st_write(young_tracts_Detroit_SHP,
"young_tracts_Detroit_SHP.shp")
```

To create my youthful and elderly tracts, I first modified the original ACS tables S0101 in Excel to only have the age columns I wanted in which the US Census Bureau aggregated the percents of people under 18 and 65. I then modified those column names to be the same as census tract shapefile column names. I then joined this new CSV to the census tracts shapefile I created in the code to the left where I clipped the Michigan ones to Detroit. I then brought that shapefile back into R, where I removed accidentally duplicated tracts there. I then created new columns in that called “PCTUND18” and “PCTOV65” which were numeric versions of the US Census Bureau’s original columns for those statistics. I then used those columns to select out which census tracts had above the average percents for both categories, and exported those to new shapefiles. This code just shows the process for youthful tracts, but the process for elderly tracts used this same code with only a few small changes.

# Predictors Creation Process

```
all_parcels <- read.csv("Parcel_Points_Ownership.csv")

vacant_parcel_rows <- grep("DETROIT LAND BANK
AUTHORITY", all_parcels$Owner)

occupied_parcels <- all_parcels[-c(vacant_parcel_rows), ]


office_building_rows <- grep("204",
occupied_parcels$Property.Class)

office_buildings <- occupied_parcels[office_building_rows,]

office_buildings <- na.omit(office_buildings)


factories_rows <- grep("301", occupied_parcels$Property.Class)

factories <- occupied_parcels[factories_rows,]

factories <- na.omit(factories)


#Repeated exact same template above for bars, arcades, pool
halls, bowling alleys, restaurants, movie theaters, stores
```

General trip generators are ones that attract people of all ages to visit them. They are important to analyze since they can attract pedestrians who get hit while walking to them. To create my general trip generators predictor, I first selected out all parcels that were occupied, which I defined as ones not owned by the Detroit Land Bank Authority which puts vacant land to reuse. [5] I then used the parcels' metadata to select out all codes that identified the general predictors I was looking for using grep() to find parcels matching that code, selecting out ones that were occupied and omitting blank rows.

[5] https://buildingdetroit.org/overview/

## Predictors Creation Process

```
general_trip_generators <- rbind(restaurants, arcades, bars, stores, billiards, bowling, theaters)

myvars <- c("Lon", "Lat")

general_trip_generators <- general_trip_generators[myvars]

general_trip_generators_SHP <- st_as_sf(general_trip_generators, coords=c("Lon","Lat"),
                                        crs=st_crs("+init=epsg:4326"))

st_write(general_trip_generators_SHP, "general_trip_generators_SHP.shp", delete_layer=TRUE)
```

After making data frames for all my individual general trip generators, I finally combined them into a general trip generators shapefile and exported that to my app's www folder to be used there. I only wanted the latitude and longitude columns for this so they could be mapped as easily as possible.

## Predictors Creation Process

```
#Parks:

parks <- st_read("parks.shp") #it's in 4326 already

parks_Detroit_SHP <- st_intersection(parks_SHP,
neighborhoods_buffer)

st_write(parks_Detroit_SHP, "parks_Detroit_SHP.shp",
delete_layer=TRUE)


#recreation centers:

rec_centers <- st_read("Detroit_Recreation_Centers.shp")

rec_centers_SHP <- st_transform(rec_centers, 4326)

st_write(rec_centers_SHP, "recreation_centers_SHP.shp")


#Places of Worship:

worship <- st_read("Detroit_Churches_2011.shp")

worship_SHP <- st_transform(worship, 4326)

st_write(worship_SHP, "places_of_worship_SHP.shp")
```

I created my parks, recreation centers and places of worship predictors with little modification from their original forms. I kept these predictors separate from the general trip generators because even though these could still attract large numbers of visitors, they more often than not are known to attract many visitors at once on a less frequent basis. For example, many people can go to office buildings every day to work at them, making them a general trip generator. However, people may often tend to visit churches, recreation centers and parks most on weekends as opposed to weekdays, which makes it not make sense to group those with the general trip generators.

# Predictors Creation Process

```
library(ggmap)

register_google(key = "AIzaSyAZtr7RNwOKj-cw3Q-trTN6AtzChUsdks0")

a <- scan("https://www.dibbern.com/nursing-homes/michigan/detroit-nursing-homes-directory.htm",what="",sep="\n")

a <- a[755:819]

a <- a[c(1:2, 21:45, 64:65)]

a <- gsub("<[^>]*>", "", a)

a <- gsub("\\([0-9]{3}\\) [0-9]{3}-[0-9]{4}", "", a)

a <- gsub("For.*", "", a)

names(a) <- c("Addr_toGC")

a$Addr_toGC <- as.character(a$Addr_toGC)

nursing_home_coordinates <- geocode(a$Addr_toGC, output = "more", messaging=TRUE, source="google")

nursing_home_coordinates_final <- nursing_home_coordinates[c(-3:-4,-6:-9)]

nursing_homes_final_SHP <- st_as_sf(nursing_home_coordinates_final, coords=c("lon","lat"), crs=st_crs("+init=epsg:4326"))

nursing_homes_Detroit_SHP <- st_intersection(nursing_homes_final_SHP, neighborhoods_buffer)

st_write(nursing_homes_Detroit_SHP,"nursing_homes_Detroit_SHP.shp", delete_layer=TRUE)
```

To create my older-adult-only trip generators, I had to web scrape the addresses of nursing homes and senior centers, geocode those addresses to get their coordinates, and make those into shapefiles which I then eventually merged with each other to get my older-adult-only trip generators predictor shapefile. This code shows the process for nursing homes, but it was essentially same for senior centers as well except for a different web address and slightly different character used in the gsub() commands. This code first scans the web site with the addresses, uses gsub() to extract only the address text I want from the site, geocodes the extracted text (the addresses) using the ggmap package geocode() command, and exports the new points as shapefiles for the app.

## Predictors Creation Process

```
schools <- read.csv("All_Schools_2017_2018.csv")
schools$Lon <- gsub("location", "", schools$location)
schools$Lon <- gsub(".*,", "", schools$Lon)
schools$Lon <- gsub("\\)", "", schools$Lon)
schools$Lon <- as.numeric(schools$Lon)
schools$Lat <- gsub("location", "", schools$location)
schools$Lat <- gsub(",.*", "", schools$Lat)
schools$Lat <- gsub("\\(", "", schools$Lat)
schools$Lat <- as.numeric(schools$Lat)

schools_SHP <- st_as_sf(schools, coords=c("Lon","Lat"),
                        crs=st_crs("+init=epsg:4326"))
st_write(schools_SHP, "schools_SHP.shp", delete_layer=TRUE)
#Same code template as above applied to childcare centers
essentially. Then merged schools and childcare centers shapefiles
in QGIS to get youth-only trip generators shapefile
```

To create my youth-only trip generators, I first had to create new latitude and longitude columns in the schools and childcare centers shapefiles. This code shows that process for schools, but the same template was used for childcare centers as well. After doing that for both shapefiles, I would export them from R to QGIS using st_write, where I then merged them to create my youth-only trip generators shapefile. I had problems merging them in R, which made me export them to QGIS to do the merging process there.

## Predictors Creation Process

```
#Transit Stops -- Merging of light rail stops, city bus stops and suburban bus stops DONE IN QGIS

transit_stops_SHP <- st_read("transit_stops_SHP.shp") #it's in 4326 already


transit_stops_Detroit_SHP <- st_intersection(transit_stops_SHP, neighborhoods_buffer)
st_crs(transit_stops_Detroit_SHP)

st_write(transit_stops_Detroit_SHP, "transit_stops_Detroit_SHP.shp", delete_layer=TRUE)


#Merged transit stops to Detroit fishnet in QGIS as well
```

Also due to problems merging the light rail stops, city bus stops and suburban bus stops shapefiles in R, I took them all to QGIS to merge them there. I then brought back that merged shapefile into R, where I clipped them using the neighborhoods to get the Detroit transit stops. I then brought that into QGIS to merge with the Detroit fishnet created earlier.

## Predictors Creation Process

```
hospitals <- st_read("hospitals.shp")

health_centers <- read.csv("Federally_Qualified_Health_Centers.csv")

health_centers$Location.1 <- gsub("MI [0-9]{5}.*", "",
health_centers$Location.1)

health_centers$Location.1 <- gsub(",,", ", MI",
health_centers$Location.1)


register_google(key = "AIzaSyAZtr7RNwOKj-cw3Q-
trTN6AtzChUsdks0")

health_center_coordinates <- geocode(health_centers$Location.1,
output = "more", messaging=TRUE, source="google")

health_center_coordinates <- health_center_coordinates[c(-3:-4,-6:-
9)]

health_centers_SHP <- st_as_sf(health_center_coordinates,
coords=c("lon","lat"), crs=st_crs("+init=epsg:4326"))

st_write(health_centers_SHP, "health_centers_SHP.shp",
delete_layer=TRUE)

#MERGED HOSPITALS AND HEALTH CENTERS SHAPEFILES IN QGIS TO
CREATE health_clinics_SHP
```

To create my health clinics predictor, I first geocoded the hospitals CSV by preparing their addresses for geocoding (ex. dropping Michigan and the zip codes using gsub) and then geocoding it to make a shapefile for it. Since health centers could already easily be made into a shapefile, I made that into one using sf and exported that to QGIS along with my new hospitals shapefile. In QGIS, I merged them to make my health clinics shapefile.

## Predictors Map Creation Process

*#Creates color scheme for transit stops fishnet:*

*bins_transit_stops_FN <- c(0,7,18,31,62,129) pal_transit_stops_FN <- colorBin("Blues", domain = transit_stops_FN$OBJECTID_c, bins = bins_transit_stops_FN, na.color = "transparent")*

*#Creates color scheme for places of worship fishnet:*

*bins_worship_FN <- c(0,2,5,9,17,27) pal_worship_FN <- colorBin("Blues", domain = worship_FN$ObjectID_cbins = bins_worship_FN, na.color = "transparent")*

*#Creates color scheme for general trip generators fishnet:*

*bins_general_trip_generators_FN <- c(0,17,40,79,152,446) pal_general_trip_generators_FN <- colorBin("Blues", domain = general_trip_generators_FN$FID_count, bins = bins_general_trip_generators_FN, na.color = "transparent")*

At this point, I now had all the data necessary to make my predictors map. However, I had to first set the graphic design guidelines of how all the map's data would be displayed. This involved creating values and palette objects for each of the map's layers in R. It involved using QGIS to see the 5 natural break values behind all of the predictor fishnets in QGIS. I had to display these predictors as fishnets because there were too many points for them to be displayed on the map in a way where the users would be able to view the concentrations of them effectively just through their points.

# Predictors Map Creation Process

```
output$PredictorsMap <- renderLeaflet({


  leaflet() %>%

    setView(lng = -83.081659, lat = 42.360304, zoom = 10.5)
%>%addProviderTiles("CartoDB.Positron") %>%


    addRasterImage(x = ksi_ped_crash_KD, colors =
pal_ksi_ped_crash_KD, opacity = 0.75, group = "KSI Ped. Crash
Kernel Density", project=FALSE) %>%addLegend(pal =
pal_ksi_ped_crash_KD, values = val_ksi_ped_crash_KD, title =
"KSI Ped. Crash Kernel Density", opacity = 1) %>%


    addPolygons(data = ksi_ped_crash_FN, group = "KSI Ped.
Crash Fishnet Density", fillColor =
~pal_ksi_ped_crash_FN(as.numeric(ksi_ped_crash_FN$Crsh_ID
_un)), weight = 0.5, opacity = 1.0, color = "#9B9B9C", fillOpacity
= 0.75) %>%addLegend(position = "topright", pal =
pal_ksi_ped_crash_FN, values = bins_ksi_ped_crash_FN, title =
"KSI Ped. Crash Fishnet Density", opacity = 1) %>%
```

This creates the KSI Pedestrian Crashes vs. Predictors Map by first adding the KSI pedestrian crash density raster and fishnet in the same exact way as the neighborhoods map.

## Predictors Map Creation Process

```
addPolygons(data = young_tracts_Detroit_SHP, group =
"Youthful Census Tracts", color = "#000000", weight = 0.375,
opacity = 1.0, fillOpacity = 0) %>%

    addPolygons(data = old_tracts_Detroit_SHP, group =
"Elderly Census Tracts", color = "#000000", weight = 0.375,
opacity = 1.0, fillOpacity = 0) %>%

    addPolygons(data = parks_Detroit_SHP, group = "Parks",
color = "#000000", weight = 0.5, opacity = 1.0, fillOpacity = 0)
%>%
```

The polygon predictors then get added to the map with no fills, just their thin outlines so they can be easily seen on the map when compared against the crash densities.

## Predictors Map Creation Process

```
addPolygons(data = general_trip_generators_FN, fillColor = ~pal_general_trip_generators_FN(as.numeric(general_trip_generators_FN$FID_count)), weight = 0.5, opacity = 1.0, color = "#9B9B9C", fillOpacity = 0.75, group = "General Trip Generators Fishnet Density") %>%

addLegend(position = "topright", pal = pal_general_trip_generators_FN, values = bins_general_trip_generators_FN, title = "# of General Trip Generators", opacity = 1) %>%


addPolygons(data = worship_FN, fillColor = ~pal_worship_FN(as.numeric(worship_FN$ObjectID_c)), weight = 0.5, opacity = 1.0, color = "#9B9B9C", fillOpacity = 0.75, group = "Places of Worship Fishnet Density") %>%

addLegend(position = "topright", pal = pal_worship_FN, values = bins_worship_FN, title = "# of Places of Worship", opacity = 1) %>%
```

The fishnets that need to be displayed in fishnet form are then added using the same template as the KSI pedestrian crash density fishnet roughly. Legends are added as well for these in the same manner as the KSI pedestrian crash density fishnet.

# Predictors Map Creation Process

```
addCircleMarkers(data = youth_only_trip_generators_SHP, group = "Youth-Only Trip Generators", lng = ~X, lat = ~Y, radius = 0.1 ) %>%

addCircleMarkers(data = older_adult_only_trip_generators_SHP, group = "Senior-Citizen-Only Trip Generators", lng = ~X, lat = ~Y, radius = 1) %>%

addCircleMarkers(data = health_clinics_SHP, group = "Health Clinics", lng = ~X, lat = ~Y, radius = 1) %>%

addCircleMarkers(data = rec_centers_SHP, group = "Recreation Centers", lng = ~X, lat = ~Y, radius = 1) %>%
```

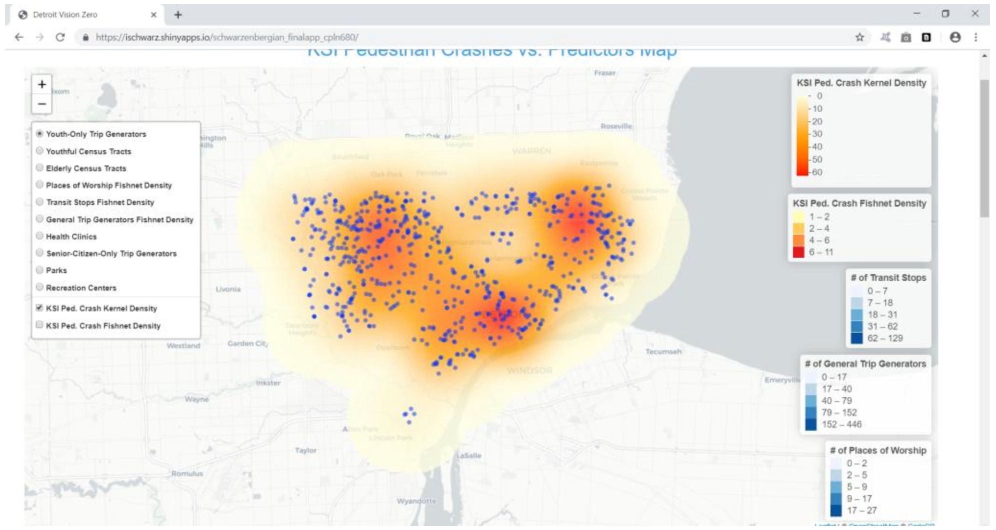
The point predictors are then added to the map using the leaflet package's addCircleMarkers() command.

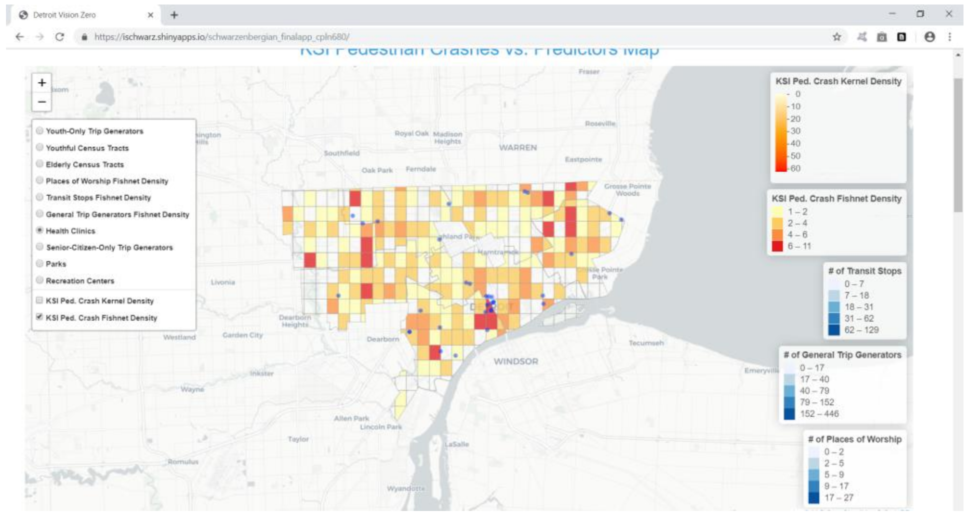
# Predictors Map Creation Process

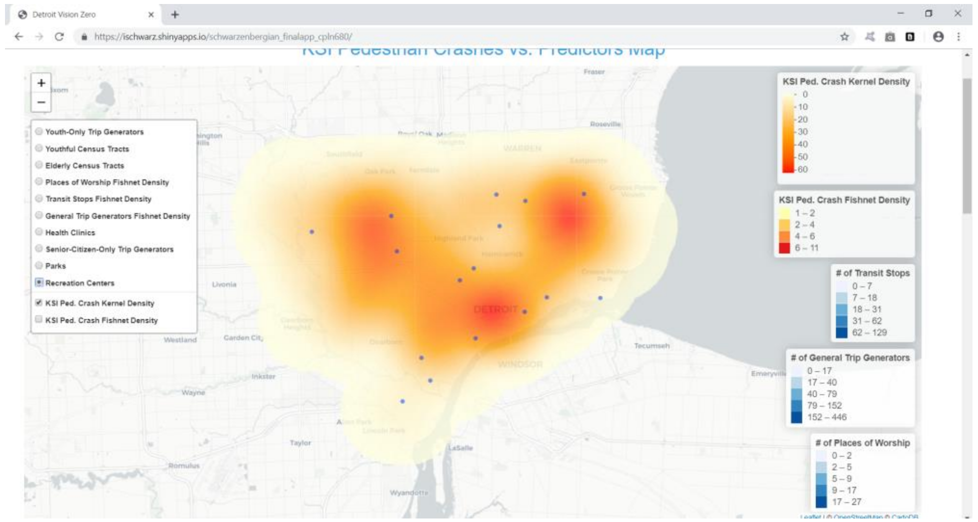
```
 hideGroup("KSI Ped. Crash Fishnet Density") %>%

    addLayersControl(

      baseGroups = c("Youth-Only Trip Generators", "Youthful
Census Tracts", "Elderly Census Tracts",

                     "Places of Worship Fishnet Density", "Transit
Stops Fishnet Density",

                     "General Trip Generators Fishnet Density",
"Health Clinics",

                     "Senior-Citizen-Only Trip Generators", "Parks",
"Recreation Centers"),

overlayGroups = c("KSI Ped. Crash Kernel Density", "KSI Ped.
Crash Fishnet Density"),

      position = c("topleft"),

      options = layersControlOptions(collapsed=FALSE))


  })
```

The legend is then created in the exact same way as the neighborhoods map, just altered for the predictors map.

# Final Predictors Map Shots


Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
KSI Pedestrian Crashes vs. Predictors Map
Youth-Only Trip Generators
Youthful Census Tracts
Elderly Census Tracts
Places of Worship Fishnet Density
Transit Stops Fishnet Density
General Trip Generators Fishnet Density
Health Clinics
Senior-Citizen-Only Trip Generators
Parks
Recreation Centers
KSI Ped. Crash Kernel Density
KSI Ped. Crash Fishnet Density
KSI Ped. Crash Kernel Density
0
10
20
30
40
50
60
KSI Ped. Crash Fishnet Density
1 – 2
2 – 4
4 – 6
6 – 11
# of Transit Stops
0 – 7
7 – 18
18 – 31
31 – 62
62 – 129
# of General Trip Generators
0 – 17
17 – 40
40 – 79
79 – 152
152 – 446
# of Places of Worship
0 – 2
2 – 5
5 – 9
9 – 17
17 – 27
WARREN
WINDSOR


Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
KSI Pedestrian Crashes vs. Predictors Map
Youth-Only Trip Generators
Youthful Census Tracts
Elderly Census Tracts
Places of Worship Fishnet Density
Transit Stops Fishnet Density
General Trip Generators Fishnet Density
Health Clinics
Senior-Citizen-Only Trip Generators
Parks
Recreation Centers
KSI Ped. Crash Kernel Density
KSI Ped. Crash Fishnet Density
WARREN
WINDSOR


Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
KSI Pedestrian Crashes vs. Predictors Map
Youth-Only Trip Generators
Youthful Census Tracts
Elderly Census Tracts
Places of Worship Fishnet Density
Transit Stops Fishnet Density
General Trip Generators Fishnet Density
Health Clinics
Senior-Citizen-Only Trip Generators
Parks
Recreation Centers
KSI Ped. Crash Kernel Density
KSI Ped. Crash Fishnet Density
DETROIT
WINDSOR

As the map's options are clicked through and analyzed, it is apparent that a handful of predictors relating to the presences of especially young and old residents show up as having the strongest visual correlations with KSI pedestrian crashes.

# Interactive Predictors Graph Creation Process

```
output$PredictorDistanceGraph <- renderPlot({
parks_DD <- st_transform(parks_Detroit_SHP, 32610)
   parks_buffer <- st_buffer(parks_DD,
as.numeric(input$selectDistance))
   parks_buffer <- st_transform(parks_buffer, 4326)


   SpatialJoinOutput <- sapply(st_within(ksi_ped_crash_SHP,
                              parks_buffer),
                         function(z) if (length(z)==0) NA_integer_
else z[1])
   parks_Pct <-
(sum(!is.na(SpatialJoinOutput))/nrow(ksi_ped_crash_SHP))*1
00
#Repeated this exact same template for other 9 predictors to
#get these objects: TSs_Pct, YTs_Pct, OTs_Pct, YTGs_Pct,
OTGs_Pct, GTGs_Pct, HCs_Pct, RCs_Pct, POWs_Pct
```

The point of this interactive graph is to help the user analyze which predictors have the highest shares of crashes happening "near" them. The definition of "near" is up to the user, as they can type in the buffer distance they would like to see.

The graph first starts out by making a buffer around each predictor, the buffer distance as determined by "input$selectDistance", which is connected to what the user types in for the buffer distance in the user interface ("selectDistance"). It then spatially joins the crashes to that user-created buffer. It then creates an R value object containing the predictor's share of all the crashes happening within the user-inputted buffer. This template is repeated for all predictors so that there will be 10 value objects, each containing each predictor's share of crashes within the user-inputted buffer.

# Interactive Predictors Graph Creation Process

*Pcts <- data.frame(c(TSs_Pct, parks_Pct, YTs_Pct, OTs_Pct, YTGs_Pct, OTGs_Pct, GTGs_Pct, HCs_Pct, RCs_Pct, POWs_Pct))*

*Pcts$PredName <- c("Transit Stops", "Parks", "Youthful Tracts", "Elderly Tracts", "Youth-only Trip Generators", "Older-Adult-Only Trip Generators", "General Trip Generators", "Health Clinics", "Recreation Centers", "Places of Worship")*

*names(Pcts) <- c("Percentage", "Predictor") Pcts$Percentage <- as.numeric(Pcts$Percentage)*

*Pcts <- data.frame(Pcts[order(-Pcts$Percentage),])*

*Pcts$BarColor <- c("#E7B22F", "#E7B22F", "#E7B22F", "#2FA4E7", "#2FA4E7", "#2FA4E7", "#2FA4E7", "#2FA4E7", "#2FA4E7", "#2FA4E7")*

*names(Pcts) <- c("Percentage", "Predictor", "BarColor") Pcts$Percentage <- as.numeric(Pcts$Percentage)*

After those 10 objects are created, a new data frame is created where:

- One column is the predictor names to be displayed on the x-axis of the interactive graph
- One column is each predictor's percent
- One column is each predictor's bar color, where the top 3 are always highlighted in orange

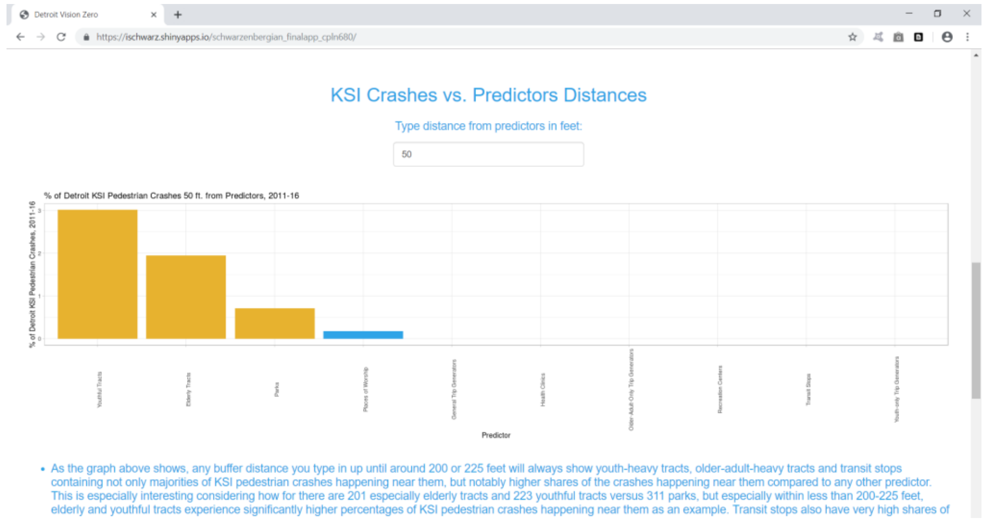
This data frame is then ordered from most to least by percent so the predictors with the biggest percents always show up on the left.

## Interactive Predictors Graph Creation Process

```
ggplot(Pcts, aes(reorder(Predictor, -as.numeric(Percentage)), as.numeric(Percentage))) +
    geom_bar(stat="identity", fill=Pcts$BarColor) +
    theme_light()+
    labs(title=paste("% of Detroit KSI Pedestrian Crashes",input$selectDistance,"ft. from Predictors, 2011-16"),
        x="Predictor",
        y="% of Detroit KSI Pedestrian Crashes, 2011-16") +
    theme(axis.text.x = element_text(angle = 90))
```

The reactive graph is then finally made. It plots the percents data frame made in the previous step where the predictors with the highest percents always show up on the left. The title is also reactive on top of the bars themselves, where the distance input is pasted into the title. The axis text is also rotated 90 degrees to accommodate the long predictor names.

# Final Interactive Predictors Graph Shots

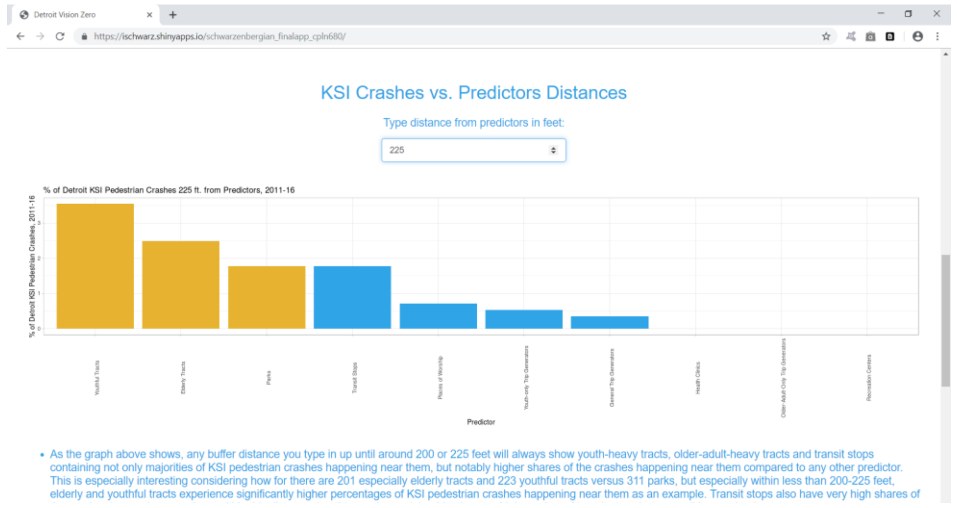

Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
% of Detroit KSI Pedestrian Crashes 50 ft. from Predictors, 2011-16
% of Detroit KSI Pedestrian Crashes, 2011-16
Youthful Tracts
Elderly Tracts
Parks
Places of Worship
General Trip Generators
Health Clinics
Older-Adult-Only Trip Generators
Recreation Centers
Transit Stops
Youth-only Trip Generators
Predictor

KSI Crashes vs. Predictors Distances

Type distance from predictors in feet:

50

- As the graph above shows, any buffer distance you type in up until around 200 or 225 feet will always show youth-heavy tracts, older-adult-heavy tracts and transit stops containing not only majorities of KSI pedestrian crashes happening near them, but notably higher shares of the crashes happening near them compared to any other predictor. This is especially interesting considering how for there are 201 especially elderly tracts and 223 youthful tracts versus 311 parks, but especially within less than 200-225 feet, elderly and youthful tracts experience significantly higher percentages of KSI pedestrian crashes happening near them as an example. Transit stops also have very high shares of

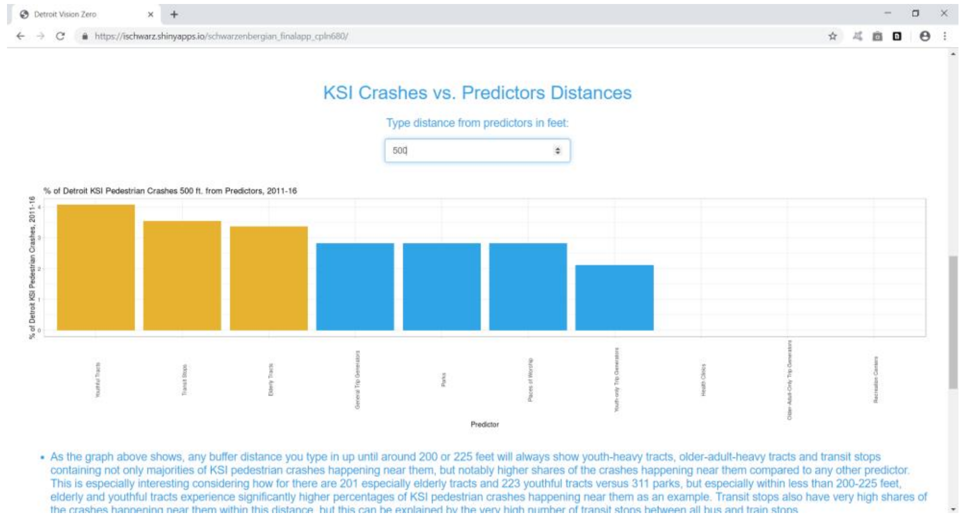

Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
% of Detroit KSI Pedestrian Crashes 225 ft. from Predictors, 2011-16
% of Detroit KSI Pedestrian Crashes, 2011-16
Youthful Tracts
Elderly Tracts
Parks
Transit Stops
Places of Worship
Youth-only Trip Generators
General Trip Generators
Health Clinics
Older-Adult-Only Trip Generators
Recreation Centers
Predictor

KSI Crashes vs. Predictors Distances

Type distance from predictors in feet:

225

- As the graph above shows, any buffer distance you type in up until around 200 or 225 feet will always show youth-heavy tracts, older-adult-heavy tracts and transit stops containing not only majorities of KSI pedestrian crashes happening near them, but notably higher shares of the crashes happening near them compared to any other predictor. This is especially interesting considering how for there are 201 especially elderly tracts and 223 youthful tracts versus 311 parks, but especially within less than 200-225 feet, elderly and youthful tracts experience significantly higher percentages of KSI pedestrian crashes happening near them as an example. Transit stops also have very high shares of


Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
% of Detroit KSI Pedestrian Crashes 500 ft. from Predictors, 2011-16
% of Detroit KSI Pedestrian Crashes, 2011-16
Youthful Tracts
Transit Stops
Elderly Tracts
General Trip Generators
Parks
Places of Worship
Youth-only Trip Generators
Health Clinics
Older-Adult-Only Trip Generators
Recreation Centers
Predictor

KSI Crashes vs. Predictors Distances

Type distance from predictors in feet:

500

- As the graph above shows, any buffer distance you type in up until around 200 or 225 feet will always show youth-heavy tracts, older-adult-heavy tracts and transit stops containing not only majorities of KSI pedestrian crashes happening near them, but notably higher shares of the crashes happening near them compared to any other predictor. This is especially interesting considering how for there are 201 especially elderly tracts and 223 youthful tracts versus 311 parks, but especially within less than 200-225 feet, elderly and youthful tracts experience significantly higher percentages of KSI pedestrian crashes happening near them as an example. Transit stops also have very high shares of the crashes happening near them within this distance, but this can be explained by the very high number of transit stops between all bus and train stops

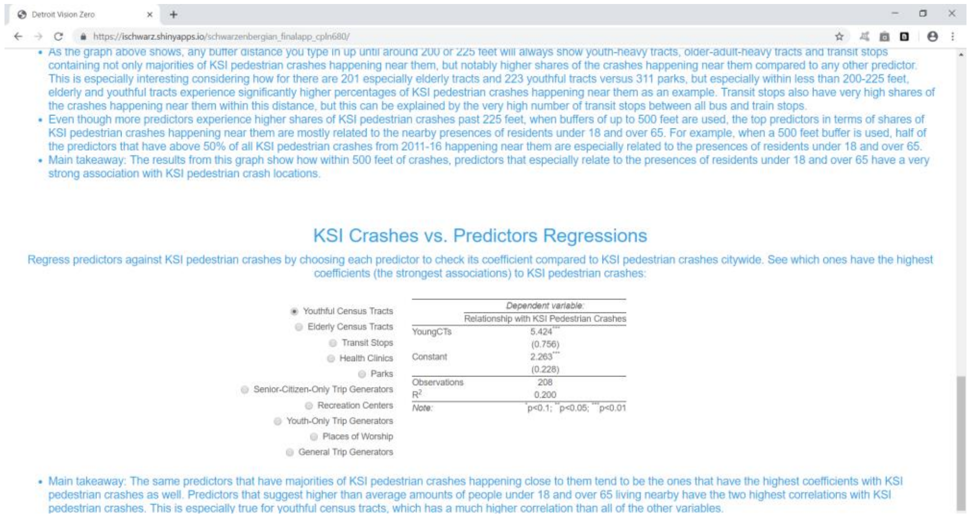
As the buffer is increased, it becomes apparent that even though the distribution of the shares of crashes among the predictors evens out, it is usually the predictors that relate most to the presences of especially young and older residents that always have the highest shares.

## Interactive Regression Table Creation Process

```
RegressionFormula<- reactive({
   as.formula(paste("KSIPdCrshs"," ~
",paste(input$PredictorsForRegression)))
 })
model <- reactive({
   lm(RegressionFormula(),neighborhoods)
 })


 output$RegressionTable <- renderText({
   stargazer(model(),type="html",dep.var.labels
="Relationship with KSI Pedestrian Crashes",
           omit.stat = c("f","ser","aic","adj.rsq"))
 })
```

This interactive regression table sets up a user-inputted regression where it regresses the KSI pedestrian crash amount column from the neighborhoods shapefile against the predictor of the user's choice. It creates a linear regression where the crashes amount column is always the dependent variable, and the user-inputted predictor (input$PredictorsForRegression) is always the dependent variable. This only regresses against the raw KSI crash amount column because if a user wanted to regress against KSI pedestrian crashes per mile of road for example, the predictors would also have to be in per mile of road. The output$RegressionTable creates the clean and organized regression table output using the stargazer package in R, and it only includes the statistics I want shown for each individual regression.

# Final Interactive Regression Table Shots



Detroit Vision Zero

https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/

- As the graph above shows, any buffer distance you type in up until around 200 or 225 feet will always show youth-heavy tracts, older-adult-heavy tracts and transit stops containing not only majorities of KSI pedestrian crashes happening near them, but notably higher shares of the crashes happening near them compared to any other predictor. This is especially interesting considering how for there are 201 especially elderly tracts and 223 youthful tracts versus 311 parks, but especially within less than 200-225 feet, elderly and youthful tracts experience significantly higher percentages of KSI pedestrian crashes happening near them as an example. Transit stops also have very high shares of the crashes happening near them within this distance, but this can be explained by the very high number of transit stops between all bus and train stops.
- Even though more predictors experience higher shares of KSI pedestrian crashes past 225 feet, when buffers of up to 500 feet are used, the top predictors in terms of shares of KSI pedestrian crashes happening near them are mostly related to the nearby presences of residents under 18 and over 65. For example, when a 500 feet buffer is used, half of the predictors that have above 50% of all KSI pedestrian crashes from 2011-16 happening near them are especially related to the presences of residents under 18 and over 65.
- Main takeaway: The results from this graph show how within 500 feet of crashes, predictors that especially relate to the presences of residents under 18 and over 65 have a very strong association with KSI pedestrian crash locations.

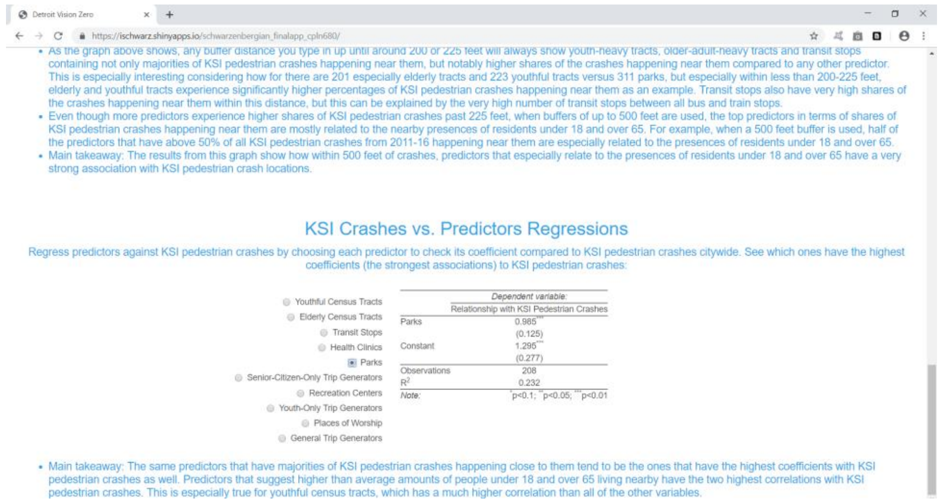
## KSI Crashes vs. Predictors Regressions

Regress predictors against KSI pedestrian crashes by choosing each predictor to check its coefficient compared to KSI pedestrian crashes citywide. See which ones have the highest coefficients (the strongest associations) to KSI pedestrian crashes:

[✓ Youthful Census Tracts]
Elderly Census Tracts
Transit Stops
Health Clinics
Parks
Senior-Citizen-Only Trip Generators
Recreation Centers
Youth-Only Trip Generators
Places of Worship
General Trip Generators

| | *Dependent variable:* |
|---|---|
| | Relationship with KSI Pedestrian Crashes |
| YoungCTs | 5.424*** |
| | (0.756) |
| Constant | 2.263*** |
| | (0.228) |
| Observations | 208 |
| $R^2$ | 0.200 |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

- Main takeaway: The same predictors that have majorities of KSI pedestrian crashes happening close to them tend to be the ones that have the highest coefficients with KSI pedestrian crashes as well. Predictors that suggest higher than average amounts of people under 18 and over 65 living nearby have the two highest correlations with KSI pedestrian crashes. This is especially true for youthful census tracts, which has a much higher correlation than all of the other variables.



Detroit Vision Zero

https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/

- As the graph above shows, any buffer distance you type in up until around 200 or 225 feet will always show youth-heavy tracts, older-adult-heavy tracts and transit stops containing not only majorities of KSI pedestrian crashes happening near them, but notably higher shares of the crashes happening near them compared to any other predictor. This is especially interesting considering how for there are 201 especially elderly tracts and 223 youthful tracts versus 311 parks, but especially within less than 200-225 feet, elderly and youthful tracts experience significantly higher percentages of KSI pedestrian crashes happening near them as an example. Transit stops also have very high shares of the crashes happening near them within this distance, but this can be explained by the very high number of transit stops between all bus and train stops.
- Even though more predictors experience higher shares of KSI pedestrian crashes past 225 feet, when buffers of up to 500 feet are used, the top predictors in terms of shares of KSI pedestrian crashes happening near them are mostly related to the nearby presences of residents under 18 and over 65. For example, when a 500 feet buffer is used, half of the predictors that have above 50% of all KSI pedestrian crashes from 2011-16 happening near them are especially related to the presences of residents under 18 and over 65.
- Main takeaway: The results from this graph show how within 500 feet of crashes, predictors that especially relate to the presences of residents under 18 and over 65 have a very strong association with KSI pedestrian crash locations.

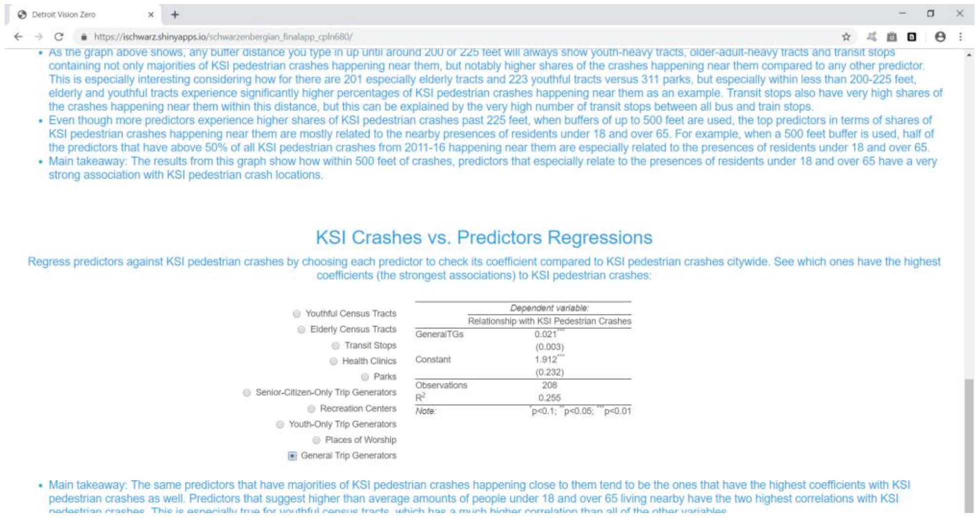
KSI Crashes vs. Predictors Regressions

Regress predictors against KSI pedestrian crashes by choosing each predictor to check its coefficient compared to KSI pedestrian crashes citywide. See which ones have the highest coefficients (the strongest associations) to KSI pedestrian crashes:

Youthful Census Tracts
Elderly Census Tracts
Transit Stops
Health Clinics
[✓ Parks]
Senior-Citizen-Only Trip Generators
Recreation Centers
Youth-Only Trip Generators
Places of Worship
General Trip Generators

| | *Dependent variable:* |
|---|---|
| | Relationship with KSI Pedestrian Crashes |
| Parks | 0.985*** |
| | (0.125) |
| Constant | 1.295*** |
| | (0.277) |
| Observations | 208 |
| $R^2$ | 0.232 |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

- Main takeaway: The same predictors that have majorities of KSI pedestrian crashes happening close to them tend to be the ones that have the highest coefficients with KSI pedestrian crashes as well. Predictors that suggest higher than average amounts of people under 18 and over 65 living nearby have the two highest correlations with KSI pedestrian crashes. This is especially true for youthful census tracts, which has a much higher correlation than all of the other variables.



Detroit Vision Zero

https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/

- As the graph above shows, any buffer distance you type in up until around 200 or 225 feet will always show youth-heavy tracts, older-adult-heavy tracts and transit stops containing not only majorities of KSI pedestrian crashes happening near them, but notably higher shares of the crashes happening near them compared to any other predictor. This is especially interesting considering how for there are 201 especially elderly tracts and 223 youthful tracts versus 311 parks, but especially within less than 200-225 feet, elderly and youthful tracts experience significantly higher percentages of KSI pedestrian crashes happening near them as an example. Transit stops also have very high shares of the crashes happening near them within this distance, but this can be explained by the very high number of transit stops between all bus and train stops.
- Even though more predictors experience higher shares of KSI pedestrian crashes past 225 feet, when buffers of up to 500 feet are used, the top predictors in terms of shares of KSI pedestrian crashes happening near them are mostly related to the nearby presences of residents under 18 and over 65. For example, when a 500 feet buffer is used, half of the predictors that have above 50% of all KSI pedestrian crashes from 2011-16 happening near them are especially related to the presences of residents under 18 and over 65.
- Main takeaway: The results from this graph show how within 500 feet of crashes, predictors that especially relate to the presences of residents under 18 and over 65 have a very strong association with KSI pedestrian crash locations.

KSI Crashes vs. Predictors Regressions

Regress predictors against KSI pedestrian crashes by choosing each predictor to check its coefficient compared to KSI pedestrian crashes citywide. See which ones have the highest coefficients (the strongest associations) to KSI pedestrian crashes:

Youthful Census Tracts
Elderly Census Tracts
Transit Stops
Health Clinics
Parks
Senior-Citizen-Only Trip Generators
Recreation Centers
Youth-Only Trip Generators
Places of Worship
[✓ General Trip Generators]

| | *Dependent variable:* |
|---|---|
| | Relationship with KSI Pedestrian Crashes |
| GeneralTGs | 0.021*** |
| | (0.003) |
| Constant | 1.912*** |
| | (0.232) |
| Observations | 208 |
| $R^2$ | 0.255 |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 |

- Main takeaway: The same predictors that have majorities of KSI pedestrian crashes happening close to them tend to be the ones that have the highest coefficients with KSI pedestrian crashes as well. Predictors that suggest higher than average amounts of people under 18 and over 65 living nearby have the two highest correlations with KSI pedestrian crashes. This is especially true for youthful census tracts, which has a much higher correlation than all of the other variables.

The predictors most associated with the crashes tend to be most associated with presences of young and elderly residents, as the interactive table shows. The predictors are listed in the radioButtons in order from highest coefficient at the top of the list to smallest coefficient at the bottom.

## User Interface Creation Process

```
tabPanel("Predictors Analysis",
tags$head(tags$style("headerPanel {color: #2FA4E7; }
radioButtons {color: #2FA4E7; } h2 {color: #2FA4E7; } h4
{color: #2FA4E7}; }")),

fluidRow(column(12, align="center", h2("KSI Pedestrian
Crashes vs. Predictors Map"))),
leafletOutput("PredictorsMap", width = 1425, height = 725),
br(), h4(tags$ul(tags$li(" Description."), tags$li("
Description."), tags$li(" Description."), tags$li("
Description."), tags$li(" Description."))), br(), br(),

#Interactive predictor distances analysis graph

fluidRow(column(12, align="center", h2("KSI Crashes vs.
Predictors Distances"), numericInput("selectDistance", label =
h4("Type distance from predictors in feet:"), value = 50))),
br(), plotOutput("PredictorDistanceGraph"), br(),
h4(tags$ul(tags$li(" Description."), tags$li(" Description."),
tags$li("Description") )), br(), br(),
```

This is the UI part which creates the predictors tab and sets the graphic design principles for the information of its layout. Like with the Neighborhoods Analysis tab, all text is in the exact same exact shade of blue as the presentation's theme. It prints the predictors map using leafletOutput() and plotOutput() for printing the interactive graph onto the app. numericInput() is used to let the user type in the buffer size they want, with a default value of 50.

## User Interface Creation Process

```
#Interactive predictor regression table

fluidRow(column(12, align="center", h2("KSI Crashes vs. Predictors Regressions"), h4(" Description "))), br(),


fluidRow(column(5, align="right", radioButtons("PredictorsForRegression", label = NA, list("Youthful Census Tracts" = "YoungCTs", "Elderly Census Tracts" = "OldCTs", "Transit Stops" = "TrnstStps", "Health Clinics" = "HlthClncs", "Parks" = "Parks", "Senior-Citizen-Only Trip Generators" = "OldOnlyTGs", "Recreation Centers" = "RecCenters", "Youth-Only Trip Generators" = "YthOnlyTGs",

"Places of Worship" = "WrshpPlcs", "General Trip Generators" = "GeneralTGs")), selected = "YoungCTs"),


column(7, align="left", tableOutput("RegressionTable")), br(),

column(12, align="left", h4(tags$ul(tags$li(“Description."))))),
```

This prints the interactive regression table onto the app, and it uses radioButtons() to let the user choose which predictors to regress against the crashes. radioButtons is specifically used over a checkbox input because the point of this is to highlight the importance of the associations between each individual predictor with the crashes, as opposed to comparing the relationships of the predictors with each other. I originally planned to make a predictive element of the project which involved this, but this was scrapped in favor of only analyzing existing associations between the predictors and the crashes. Like with the Neighborhoods Analysis tab pre-made graphs, a list is made in radioButtons() which translates the clean predictor names to their respective column names in the neighborhoods shapefile. Youthful census tracts is selected by default since this has the highest coefficient correlation.

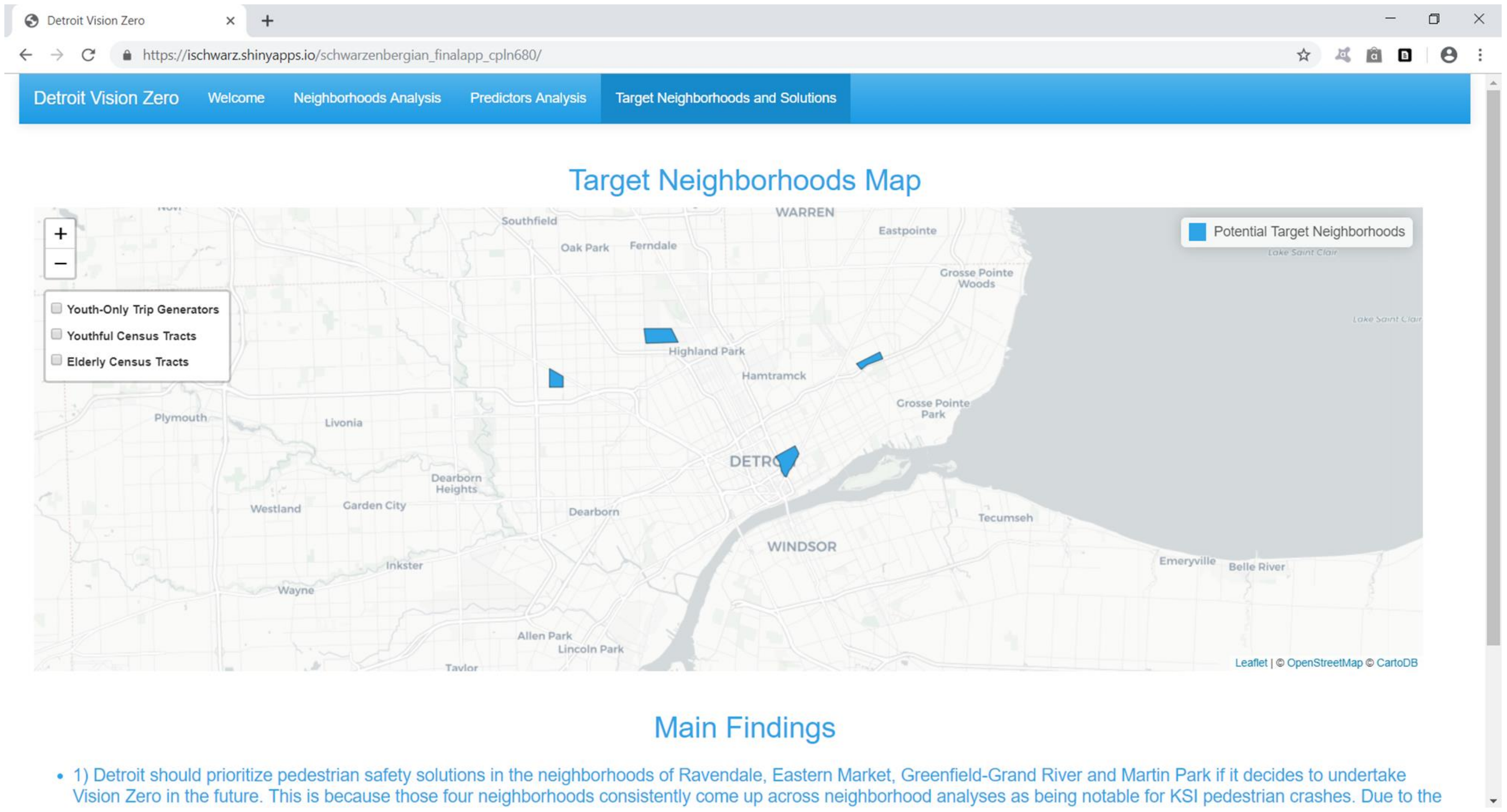
## Analysis Results

Across all three of the tab's analyses, the predictors that come up as having the strongest associations with the crashes tend to be related to the presences of residents under 18 and over 65 years old, falling in line with the original hypothesis of the project. There are three of these predictors in particular which Detroit officials should especially consider when figuring out how to prioritize pedestrian safety solutions if it decides to undertake Vision Zero in the future:

1) Youth-Only Trip Generators
2) Youthful Census Tracts
3) Elderly Census Tracts

These were selected because of how they usually come up as the top three predictors in terms of association with crashes in all the analyses. For example, those three predictors show up as the three most associated predictor with crashes in terms of visual correlation in the predictors map.

# Last Tab:
# Target Neighborhoods and Solutions



Target Neighborhoods Map

Main Findings

- 1) Detroit should prioritize pedestrian safety solutions in the neighborhoods of Ravendale, Eastern Market, Greenfield-Grand River and Martin Park if it decides to undertake Vision Zero in the future. This is because those four neighborhoods consistently come up across neighborhood analyses as being notable for KSI pedestrian crashes. Due to the

## Tab Purpose

The point of this tab is to combine the findings from the previous two tabs to simply show the user which neighborhoods should be prioritized the most and predictors the solutions should be placed near in those neighborhoods when considering where to prioritize implementations of solutions to reduce pedestrian deaths and serious injuries in Detroit.

Through the interactive map, the user can simultaneously see the locations of the neighborhoods that consistently came up as having the most crashes and which predictors consistently come up as having the strongest associations with the crashes.

## Target Neighborhoods Map Creation Process

```
target_neighborhoods <- neighborhoods[
which(neighborhoods$nhood_name=='Ravendale' |
neighborhoods$nhood_name=='Eastern Market'|
neighborhoods$nhood_name=='Greenfield-Grand River'|
neighborhoods$nhood_name=='Martin Park'), ]


output$TargetNeighborhoodsMap <- renderLeaflet({

  leaflet() %>%

    setView(lng = -83.081659, lat = 42.360304, zoom = 10.5)
%>%addProviderTiles("CartoDB.Positron") %>%


 addPolygons(data = target_neighborhoods, fillColor =
"#2FA4E7", weight = 0.5, opacity = 1.0, color = "#000000",
fillOpacity = 1, popup= paste0("<strong>Neighborhood:
</strong>", target_neighborhoods$nhood_name)) %>%


addLegend(position = "topright", colors = c("#2FA4E7"), labels
= c("Potential Target Neighborhoods"), opacity = 1) %>%
```

This creates a map that is similar in template to the application's other 2 maps. It first starts by creating a new shapefile of just the target neighborhoods, subsetted from the original neighborhoods shapefile. It then goes about creating the map by adding the target neighborhoods and a respective legend for them. This legend is different from the ones used in the other maps in the sense that it just has 1 item, the blue neighborhood symbol. Its fill color is in line with the exact shade of blue used throughout the entire app and this report.

# Target Neighborhoods Map Creation Process

*addCircleMarkers(data = youth_only_trip_generators_SHP, group = "Youth-Only Trip Generators", color = "#343434", lng = ~X, lat = ~Y,  radius = 0.1, popup= paste0("<strong>Name: </strong>", youth_only_trip_generators_SHP_forPredGraph$business_n) ) %>%*

*addPolygons(data = young_tracts_Detroit_SHP, group = "Youthful Census Tracts", color = "#000000", weight = 0.375, opacity = 1.0, fillOpacity = 0, popup= paste0("<strong>Census Tract Number: </strong>", young_tracts_Detroit_SHP$NAME, "<br>", "<strong>% of Pop. Under 18: </strong>", young_tracts_Detroit_SHP$PCTUND18, "<br>", "<strong>City Avg. for Tract % of Pop. Under 18: </strong>", 3.71)) %>%*

*addPolygons(data = old_tracts_Detroit_SHP, group = "Elderly Census Tracts", color = "#000000", weight = 0.375, opacity = 1.0, fillOpacity = 0, popup= paste0("<strong>Census Tract Number: </strong>", old_tracts_Detroit_SHP$NAME, "<br>", "<strong>% of Pop. Over 65: </strong>", old_tracts_Detroit_SHP$PCTOV65, "<br>", "<strong>City Avg. for Tract % of Pop. Over 65: </strong>", 3.47)) %>%*

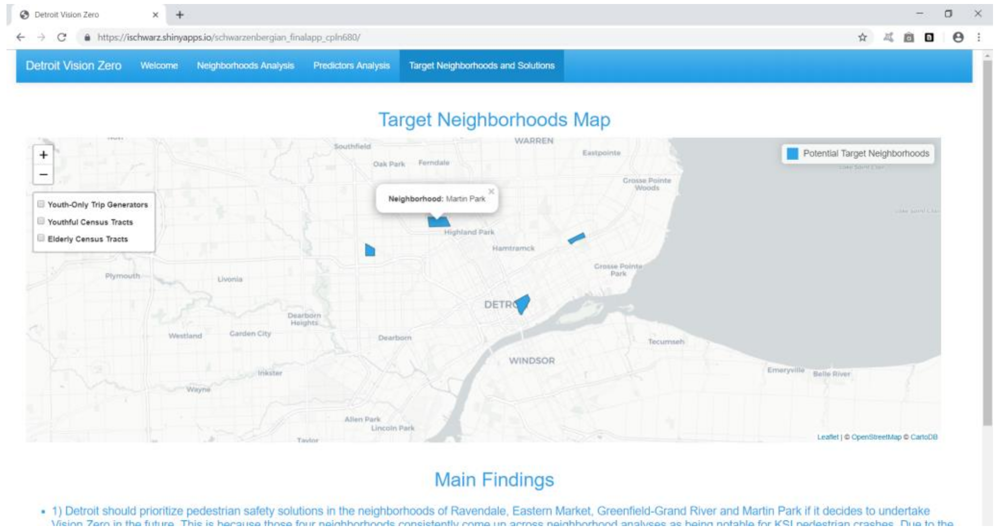
The map then adds the top 3 predictors, with popups detailing each predictor's name or census tract number, institution name, percent of young people compared to the city average or percent of old people compared to the city average depending on the predictor. The youth trip generators popup notably gets its information from a copy made of the shapefie that has all of its original data, as opposed to the version of it used throughout most of the app which has just each predictor's coordinates.

## Target Neighborhoods Map Creation Process

```
hideGroup("Youth-Only Trip Generators") %>%
    hideGroup("Youthful Census Tracts") %>%
    hideGroup("Elderly Census Tracts") %>%
    addLayersControl(
    overlayGroups = c("Youth-Only Trip Generators", "Youthful Census Tracts", "Elderly Census Tracts"),
    position = c("topleft"),
    options = layersControlOptions(collapsed=FALSE))


 })
```

The legend then finally gets added to the map, with the three predictors being unchecked by default using hideGroup() as to highlight the importance of the neighborhood locations. This also notably makes all the predictors check box items, unlike in the predictor tab when they were radio buttons. This is to further highlight the importance of the neighborhood locations in the map.

# Final Target Neighborhoods Map Shots

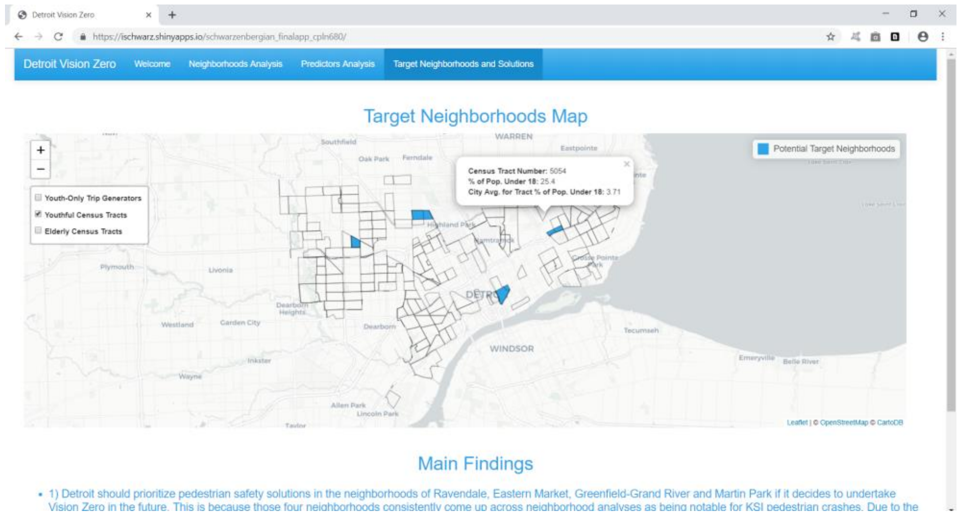

Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
Detroit Vision Zero
Welcome
Neighborhoods Analysis
Predictors Analysis
Target Neighborhoods and Solutions
Potential Target Neighborhoods
Youth-Only Trip Generators
Youthful Census Tracts
Elderly Census Tracts
Leaflet | © OpenStreetMap © CartoDB

Target Neighborhoods Map

**Neighborhood:** Martin Park

Main Findings

- 1) Detroit should prioritize pedestrian safety solutions in the neighborhoods of Ravendale, Eastern Market, Greenfield-Grand River and Martin Park if it decides to undertake

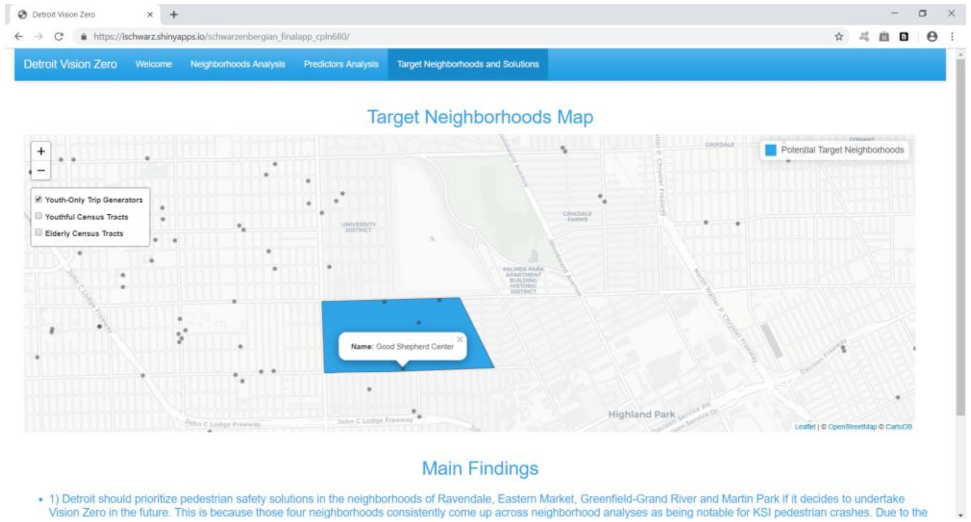

Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
Detroit Vision Zero
Welcome
Neighborhoods Analysis
Predictors Analysis
Target Neighborhoods and Solutions
Potential Target Neighborhoods
Youth-Only Trip Generators
Youthful Census Tracts
Elderly Census Tracts
Leaflet | © OpenStreetMap © CartoDB

Target Neighborhoods Map

**Census Tract Number:** 5054
**% of Pop. Under 18:** 25.4
**City Avg. for Tract % of Pop. Under 18:** 3.71

Main Findings

- 1) Detroit should prioritize pedestrian safety solutions in the neighborhoods of Ravendale, Eastern Market, Greenfield-Grand River and Martin Park if it decides to undertake Vision Zero in the future. This is because those four neighborhoods consistently come up across neighborhood analyses as being notable for KSI pedestrian crashes. Due to the


Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
Detroit Vision Zero
Welcome
Neighborhoods Analysis
Predictors Analysis
Target Neighborhoods and Solutions
Potential Target Neighborhoods
Youth-Only Trip Generators
Youthful Census Tracts
Elderly Census Tracts
Highland Park
Leaflet | © OpenStreetMap © CartoDB

Target Neighborhoods Map

**Name:** Good Shepherd Center

Main Findings

- 1) Detroit should prioritize pedestrian safety solutions in the neighborhoods of Ravendale, Eastern Market, Greenfield-Grand River and Martin Park if it decides to undertake Vision Zero in the future. This is because those four neighborhoods consistently come up across neighborhood analyses as being notable for KSI pedestrian crashes. Due to the

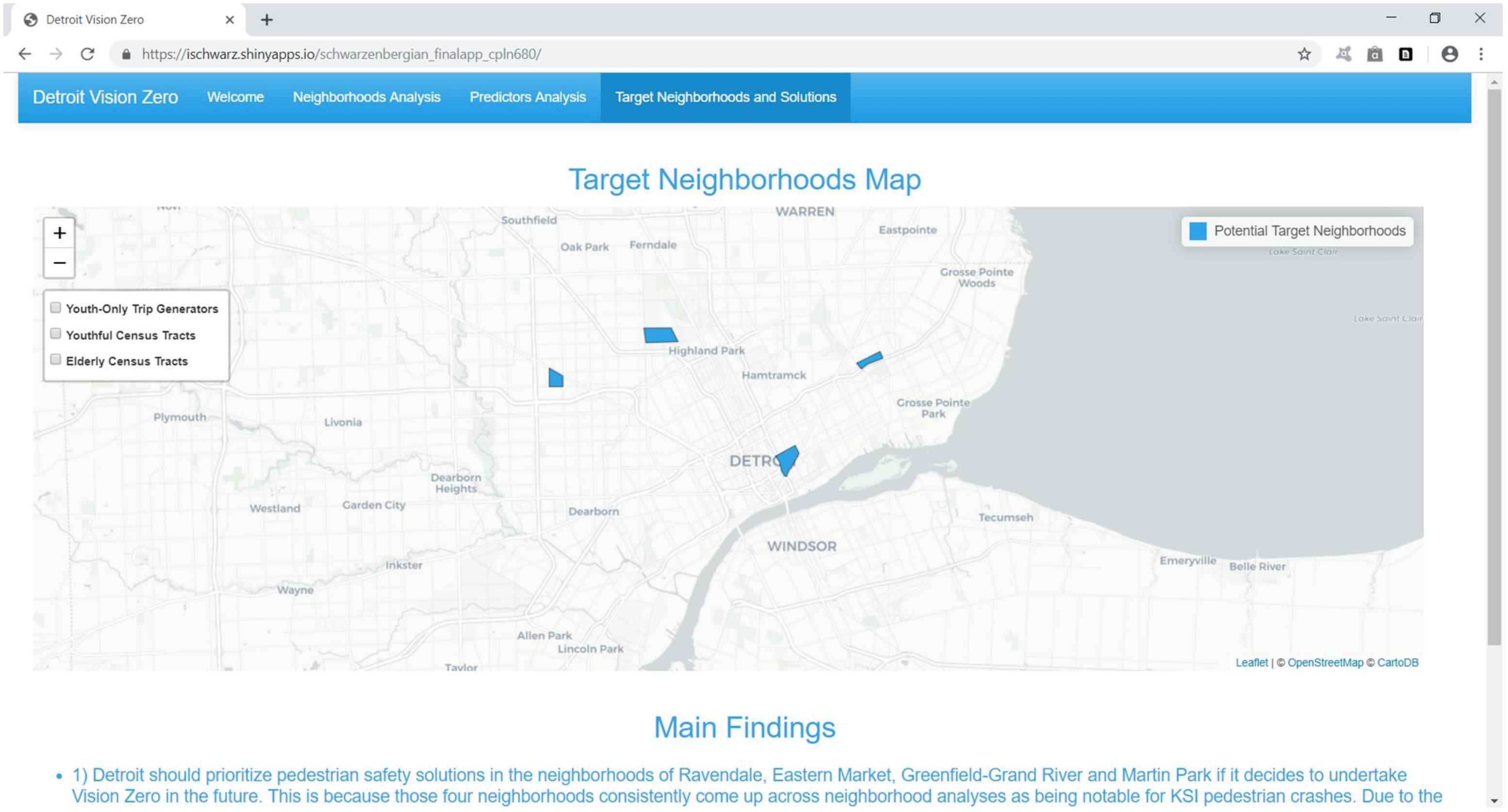
The map not only lets the user see where the potential target neighborhoods are in Detroit, but they can also zoom in to click on predictors located within them to further see which institutions to reach out to if they plan to install solutions by them in the future. All four of the neighborhoods notably line up with the crash densities.

# User Interface Creation Process

```
tabPanel("Target Neighborhoods and Solutions",
tags$head(tags$style("headerPanel {color: #2FA4E7; } h2
{color: #2FA4E7; } h4 {color: #2FA4E7}; }")),

fluidRow(column(12, align="center", h2(" Description "))),
leafletOutput("TargetNeighborhoodsMap", width = 1425,
height = 475), br(),

fluidRow(column(12, align="center", h2("Main Findings"))),

h4(tags$ul(tags$li(" Description "), tags$li("Description”),
tags$li(“Description")

        ))

       )

       )

)


shinyApp(ui = ui, server = server)
```

This prints the final tab printing the map and associated descriptions, which are labeled “Description” as a placeholder for this report as for the other tabs. Since this is the last tab, the line that combines the server and user interface objects to make the app comes right after this.

# Final Results


Detroit Vision Zero
https://ischwarz.shinyapps.io/schwarzenbergian_finalapp_cpln680/
Detroit Vision Zero
Welcome
Neighborhoods Analysis
Predictors Analysis
Target Neighborhoods and Solutions
Youth-Only Trip Generators
Youthful Census Tracts
Elderly Census Tracts
Potential Target Neighborhoods
Leaflet | © OpenStreetMap © CartoDB

## Target Neighborhoods Map

## Main Findings

- 1) Detroit should prioritize pedestrian safety solutions in the neighborhoods of Ravendale, Eastern Market, Greenfield-Grand River and Martin Park if it decides to undertake Vision Zero in the future. This is because those four neighborhoods consistently come up across neighborhood analyses as being notable for KSI pedestrian crashes. Due to the

## Final Results

Detroit should prioritize pedestrian safety solutions in these neighborhoods if it decides to undertake Vision Zero in the future:

1) Ravendale
2) Eastern Market
3) Greenfield-Grand River
4) Martin Park

This is because those four neighborhoods consistently come up across neighborhood analyses as being notable for KSI pedestrian crashes. Due to the city's especially high rate of pedestrian deaths compared to the rest of the country, the city should prioritize implementing pedestrian-oriented solutions first if it decides to undertake Vision Zero in the future.

## Final Results

Across all predictor analyses, the presences of especially youthful and elderly census tracts and other age-related predictors like youth-only trip generators came up as having the strongest associations with KSI pedestrian crash locations. Therefore, solutions Detroit takes if it decides to implement Vision Zero in the future should especially geared towards making streets safer for pedestrians under 18 and over 65 in the neighborhoods outlined above, and near those predictors.

## Final Results

Potential solutions geared towards especially young and elderly pedestrians that Detroit officials can consider prioritizing these 4 neighborhoods for in the future include:

1) Strategies that help pedestrians who walk slowly like curb extensions, refuge islands and increased crossing times

2) Strategies that help people who are hard of hearing and/or vision such as Accessible Pedestrian Signals

3) Designating at least some of those 4 neighborhoods as special senior or even youth safety zones to target both physical engineering improvements and community outreach programs geared towards helping older adult pedestrians travel more safely down streets

These particular solutions to help young and old pedestrians originate from The Vision Zero Toolbox, a report I helped create analyzing Philadelphia's Vision Zero agenda in my fall 2018 PennDesign urban planning transportation studio viewable here: https://www.design.upenn.edu/city-regional-planning/graduate/work/vision-zero-toolbox. These solutions can be applied to other cities besides Philadelphia, since they are designed to universally help young and old pedestrians.