Matthew Ungaro
16 November 2020
GEOG 592

Environmental Justice and Unimpaired Streams in Texas

Environmental justice is a relatively young environmental concept, but it is very important. Up until the 1980s, there had been no environmental justice movement. In 1982, protests in Warren County, North Carolina erupted when it, a small, majority African American community, was chosen as the location for a site to dispose of hazardous PCB waste (DOE). The protesters expressed feelings of being targeted based on demographics, which other communities began to articulate as well. After the first National People of Color Environmental Justice Summit in 1991, the United States federal government began to consider environmental justice when impacting the environment. Additionally, researchers began to study the connection between living in a low income or minority community and environmental degradation.

Research shows a strong correlation between exposure to environmental inequities and being a member of a racial or ethnic minority (Ringquist 2005). Communities of color – communities in which a majority of individuals are part of a racial or ethnic minority - have had less access to green spaces and recreational areas (Arnold 2007). Studies suggest that low-income and communities of color are more likely to be located near industrial sites than wealthier, whiter communities (Perera and Lam 2013). Environmental restoration tends to occur away from communities of color (Moran 2009). Most studies examine low income communities and communities of color and air quality and land degradation. Fewer studies appear to examine water quality. I will study this using the state of Texas as an example. Texas is a diverse state that may mirror the future makeup of the United States. The percent of white and black or African American individuals mirrors present day America (78.7% and 12.9% in Texas; 76.3% and 13.4% in the USA), while the percent of Hispanic identifying individuals is much higher than in the rest of the USA (39.7% in Texas; 18.5% in USA) (US Census 2019). According to Passel and Cohn of the Pew Research Center, 29% of the United States' population will be Hispanic by 2050 if trends hold (2008). Thus, Texas is a good example of present-day American demographics and future American demographics as well. I will study if low income communities and communities of color have equitable access to unimpaired streams in the cities and large towns of Texas. I will determine correlations between percent of unimpaired streams within communities and the racial and economic makeup of those communities. I predict that unimpaired streams will be most common within majority white communities, and less common in communities of color.

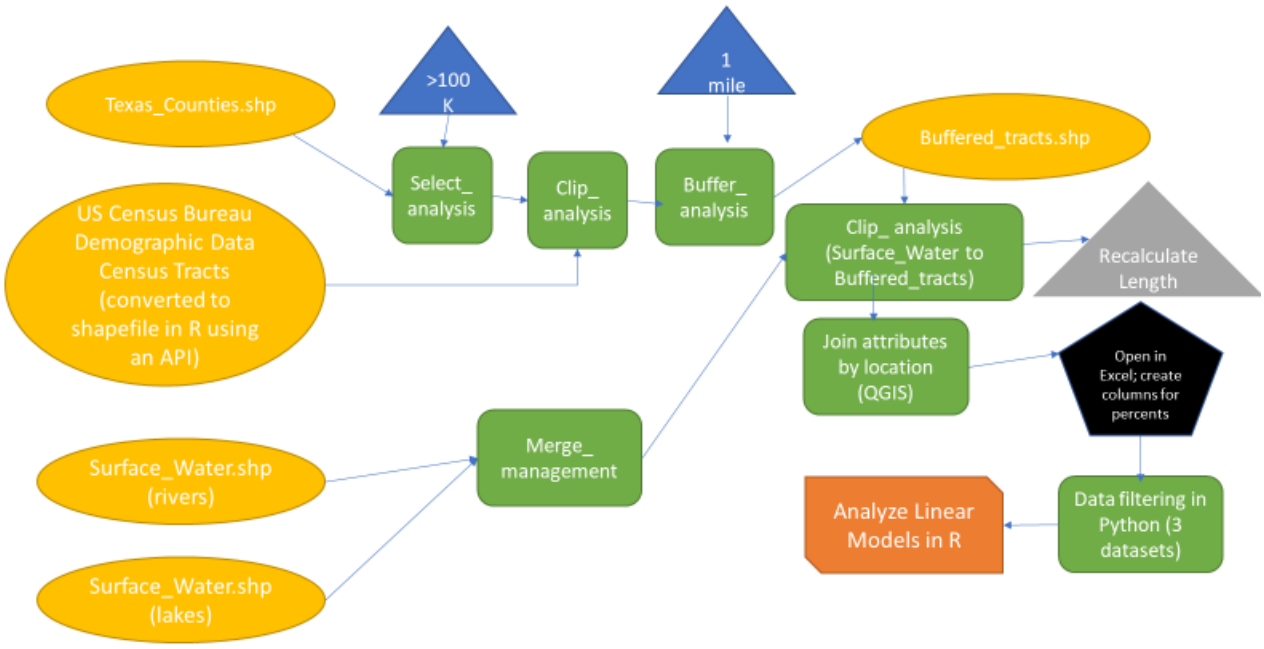
Method:

To begin this analysis, I first needed to gather data. I acquired a shapefile of Texas county boundaries, which included county population data for 2016, from the United States Census Bureau (2019). From the Texas Commission on Environmental Quality, I gathered two shapefiles of surface water – one of perennial, intermittent, and tidal streams and rivers and one of lakes and reservoirs (2020a, 2020b). Both shapefiles contained information regarding whether a stream, lake, reservoir was unimpaired or impaired. Impaired waters are polluted; types of

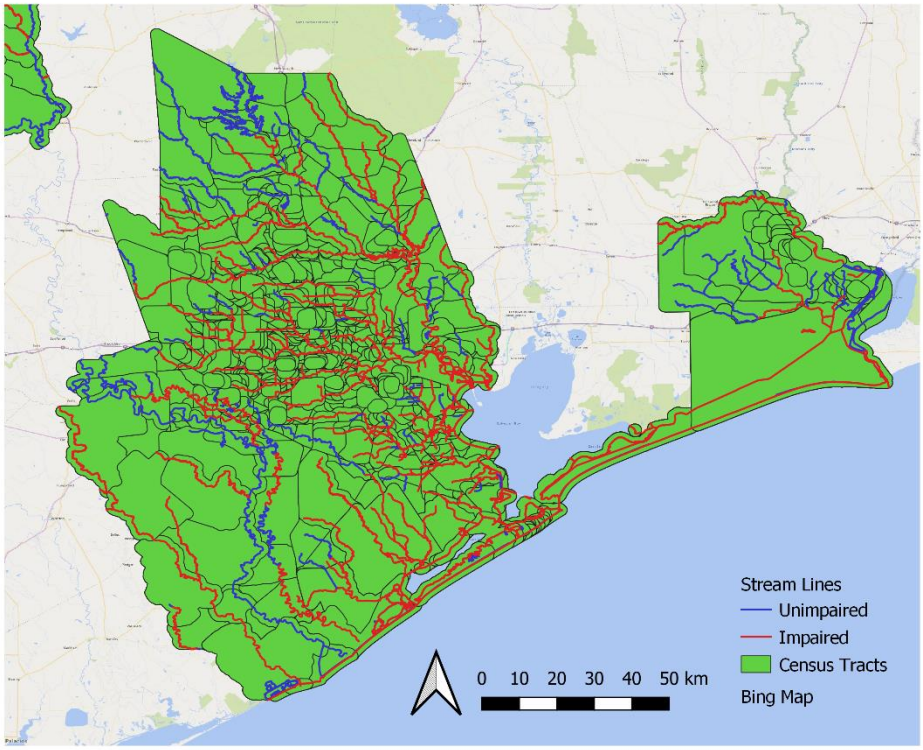
pollution vary, but some examples in Texas are high levels of bacteria, depressed dissolved oxygen, high levels of total suspended solids, chloride levels, mercury in fish tissue, PCBs in fish tissue, and others (TCEQ 2020c). I used the R statistical software platform [v3.6.2] to acquire US Census Bureau data using the package tidycensus from the American Community Survey (R Core Team 2019, Walker 2019). I obtained estimates for 2018 for white, black, Hispanic, Asian, and American Indian populations within census tracts in Texas, and I obtained estimated median income within those tracts. I converted these datasets into one shapefile based on census tract location using the package sf (Pebesma 2018).

After acquiring the data, I prepared it for geospatial analysis. In Python, I used the package arcpy to select counties with greater than 100,000 residents. I clipped my census tract shapefile to the selected counties. Assuming that within each census tract, individuals without vehicles would at most walk one mile to a water body for recreation and would be most likely to be negatively impacted by an impaired stream within that distance, I buffered all census tracts by 1 mile. The census tracts were then set aside. My shapefiles of surface water contained one polyline layer for streams/rivers and one polygon layer for lakes/reservoirs. Next, I converted my polygon layer for lakes/reservoirs into a polyline layer, with the outline of lakes/reservoirs representing the polylines and merged the two shapefiles. I clipped the polylines file to the buffered census tracts. I recalculated the length of each polyline in meters for the clipped file. With this, I finished my preparation in Python.

Chart 1: Project Process



Map 1: Map of Houston Area with Buffered Census Tracts and River and Lake Lines



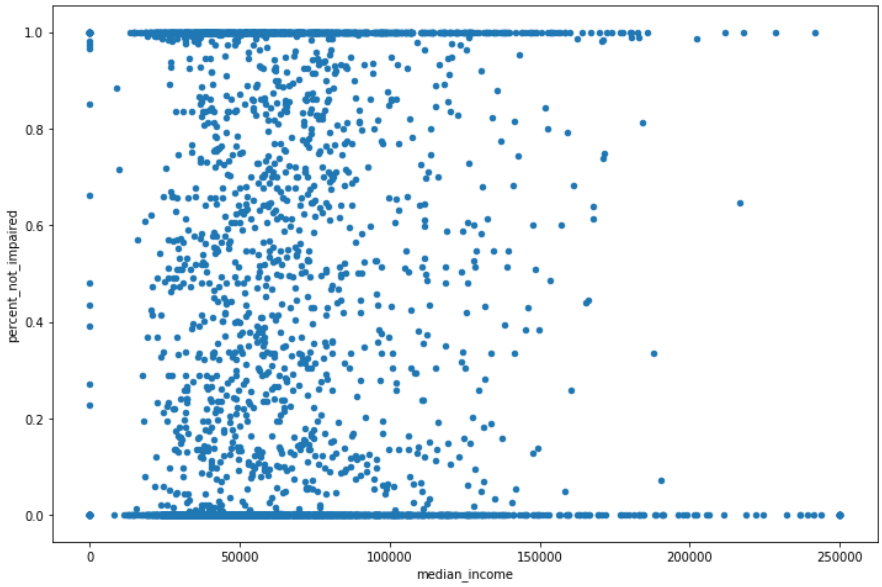
With my polyline layer of streams and lakes and my buffered census tracts, I used join attributes by location in QGIS [v3.10.6] that easily allowed me to join the attributes of the two shapefiles together. This created one shapefile combining attributes from both layers. I opened the .dbf file of this shapefile in Microsoft Excel and summed the total length of impaired and unimpaired streams and lakes by GEOID – the attribute representing each census tract. I combined this new dataset with the buffered census tract file in Excel, matching GEOID with GEOID. At this point, I had a dataset that included all census tracts, estimates for each tract for white, black, Hispanic, Asian, and American Indian populations and median income, and the lengths of impaired streams and lakes and unimpaired streams and lakes within the census tract buffer. Within Excel, I created new columns for percent impaired, percent unimpaired, percent white, percent black, percent Hispanic, percent Asian, and percent American Indian in these census tracts. I then imported this dataset into Python.

Within Python, I used the package pandas to try to visualize my data (the Pandas Development Team 2020). This proved more difficult than expected. The quantity of data made it very challenging to determine whether there was a correlation between various demographic percentages and unimpaired stream percentages, as seen in graph 1. Therefore, I instead used Python to manipulate the data in two ways – select the 1000 census tracts with the most stream and lake lines and remove any entries with unimpaired streamlines equaling 100% or 0%. I chose to remove 0% or 100% unimpaired as some census tracts may have no streams or lakes, so 0% unimpaired might just mean that the census tract is within an arid environment, and 100%

unimpaired might just mean that a small, short stream is making a huge impact on the data. I then exported all of these new datasets into R, along with the untouched dataset.

Within R, I ran ordinary least squares regression to determine a correlation between the percent of unimpaired streams within a census tract and the previously mentioned demographic categories (Wickham et al. 2018). I regressed each category three times – once with all the data, once with only the 1000 census tracts with the most meters of stream and lake lines, and once without 0% or 100% unimpaired. This data was placed into tables 1, 2, and 3. I also graphed significant data with regression lines. This data was placed into graph 2.

Graph 1: Median Income and Percentage of Streams and Lakes Unimpaired in Texas Census Tracts in Python



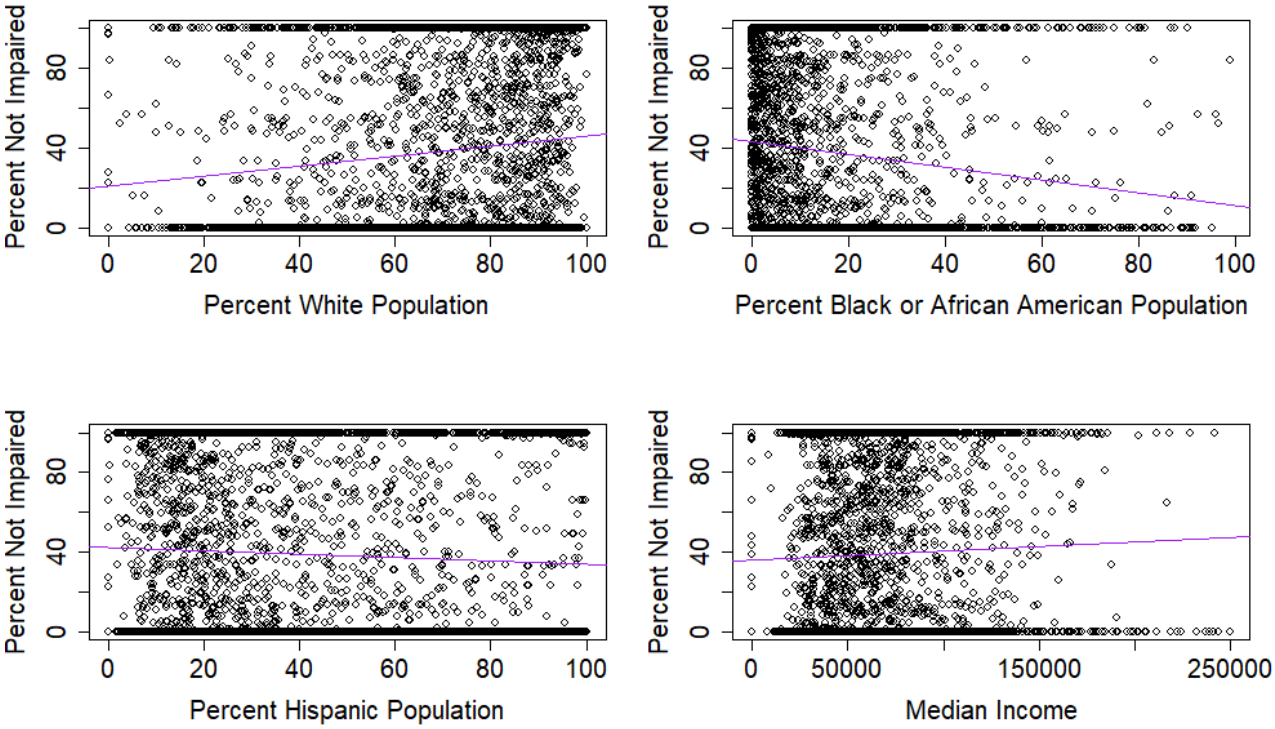
Results:

Most of the results of this analysis seem to indicate no statistically significant correlations between impaired streams and demographic makeup of census tracts. However, across all regressions, two categories seem to have a positive relationship and a negative relationship with impaired streams. The percentage of white individuals within a census tract is positively associated with unimpaired streams. As the percentage of white individuals rises within a census tract, unimpaired stream percentages also rise. Conversely, the percentage of black or African American individuals within a census tract is negatively associated with unimpaired streams. As the percentage of black or African American individuals increase, the percentage of high-quality streams appears to decrease. Across all the data, percentage Hispanic and percentage unimpaired seem to also have a negative association, but when the data is subset, the significance disappears. This is true also for median income and unimpaired streams, though the correlation is positive. Only the correlations between percent white and unimpaired and percent black or African American and unimpaired remain statistically significant across regressions. These relationships, based on all the data, can be seen in graph 2 and table 1 as well as tables within the appendices.

Table 1: Linear Relationship between Categories (Model 1)

Dataset	relationship	y-intercept	slope	p-value	significance
All Data (n = 4170)	% white - % not impaired	20.61997	0.25204	1.09e-13	**Statistically significant**
	% black - % not impaired	42.95479	-0.31661	9.48e-15	**Statistically significant**
	% Hispanic - % not impaired	42.17149	-0.08083	0.000743	**Statistically significant**
	median income - % not impaired	3.593e+01	4.534e-05	0.0183	**Statistically significant**

Graph 2: Correlation between Unimpaired Streams and Various Demographic Categories, All Data (n = 4170). Purple line represents linear regression.



Discussion:

What does this mean? This appears to indicate that in Texas, a correlation exists between exposure to impaired water quality and living within a predominantly black or African American community. This seems more apparent than being a member of any other minority group or low-income community. A correlation also seems to exist for members of the state's white community and better access to good water quality. However, caution seems necessary before making any claims. The data does not closely follow the regression line; many outliers exist. There is a very low R-squared value for both regressions. More analysis is needed before any claims are made when more time is available.

If I were to continue this analysis, I might go back to the original census tract data and select a large number of census tracts with approximately equal population sizes or the largest 200 census tracts. In my analysis, I assumed that the largest counties would all have census tracts

with large populations, but it is possible that a census tract might be in a large county but has a small population. These census tracts may have skewed the data results. I would also consider changing my method. I could take the center point of each census tract and find the closest unimpaired stream. Then I could see how access varies across census tracts. However, in Texas, where some census tracts are far from any water sources, this may skew the data as well. It might be best to repeat this analysis then in another state with more water sources and similar levels of diversity.

It is clear from the data that impaired streams affect different communities in Texas in different ways. An environmental scientist or city planner may view their community's waterways with satisfaction, forgetting that their experience may not be representative of the whole state or even the whole county. Planners and scientists need to recognize the inequitable environmental history of our nation and avoid repeating the mistakes of previous planners and scientists. Environmental work is needed to restore and protect water quality for all communities in Texas, no matter their demographics.

Works Cited

Arnold, C. A. (2007). *Fair and Healthy Land Use: Environmental Justice and Planning* (Issues 549–550). http://dx.doi.org/10.1016/j.jaci.2012.05.050

Department of Energy (DOE) – Office of Legacy Management. Environmental justice history. https://www.energy.gov/lm/services/environmental-justice/environmental-justice-history.

McKinney, W. (2010) Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*, 445.

Moran, S. (2010). Cities, creeks, and erasure: Stream restoration and environmental justice. *Environmental Justice*, *3*(2), 61–69. https://doi.org/10.1089/env.2009.0036

The Pandas Development Team (2020). pandas-dev/pandas: Pandas. Zenodo. 10.5281/zenodo.3509134

Passel, J., and Cohn, D. (2008). U.S. Population Projections: 2005-2050. Pew Research Center. https://www.pewresearch.org/hispanic/2008/02/11/us-population-projections-2005-2050/

Pebesma, E. (2018). Simple features for R: Standardized support for spatial vector data. *The R Journal*, *10*(1), 439-446, https://doi.org/10.32614/RJ-2018-009

Perera, P. K. P., & Lam, N. (2013). An environmental justice assessment of the Mississippi river industrial corridor in Louisiana, U.S. using a gis-based approach. *Applied Ecology and Environmental Research*, *11*(4), 681–697. https://doi.org/10.15666/aeer/1104_681697

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Ringquist, E. J. (2005). Assessing evidence of environmental inequities: A meta-analysis. *Journal of Policy Analysis and Management*, *24*(2), 223–247. https://doi.org/10.1002/pam.20088

Texas Commission on Environmental Quality (2020). Segments – line. Retrieved from https://gis-tceq.opendata.arcgis.com/datasets/segments-line?geometry=-114.527%2C24.526%2C-85.611%2C37.648

Texas Commission on Environmental Quality (2020). Segments – poly. Retrieved from https://gis-tceq.opendata.arcgis.com/datasets/segments-poly

Texas Commission on Environmental Quality (2020). 2020 Texas Integrated Report Index of Water Quality Impairments. Retrieved from https://www.tceq.texas.gov/waterquality/assessment

U.S. Census Bureau (2019). Quickfacts: TX. https://www.census.gov/quickfacts/TX

U.S. Census Bureau (2019). Quickfacts: United States. https://www.census.gov/quickfacts/fact/table/US/PST045219

U.S. Census Bureau (2018). Selected housing characteristics, 2014-2018 American Community Survey 5-year estimates.

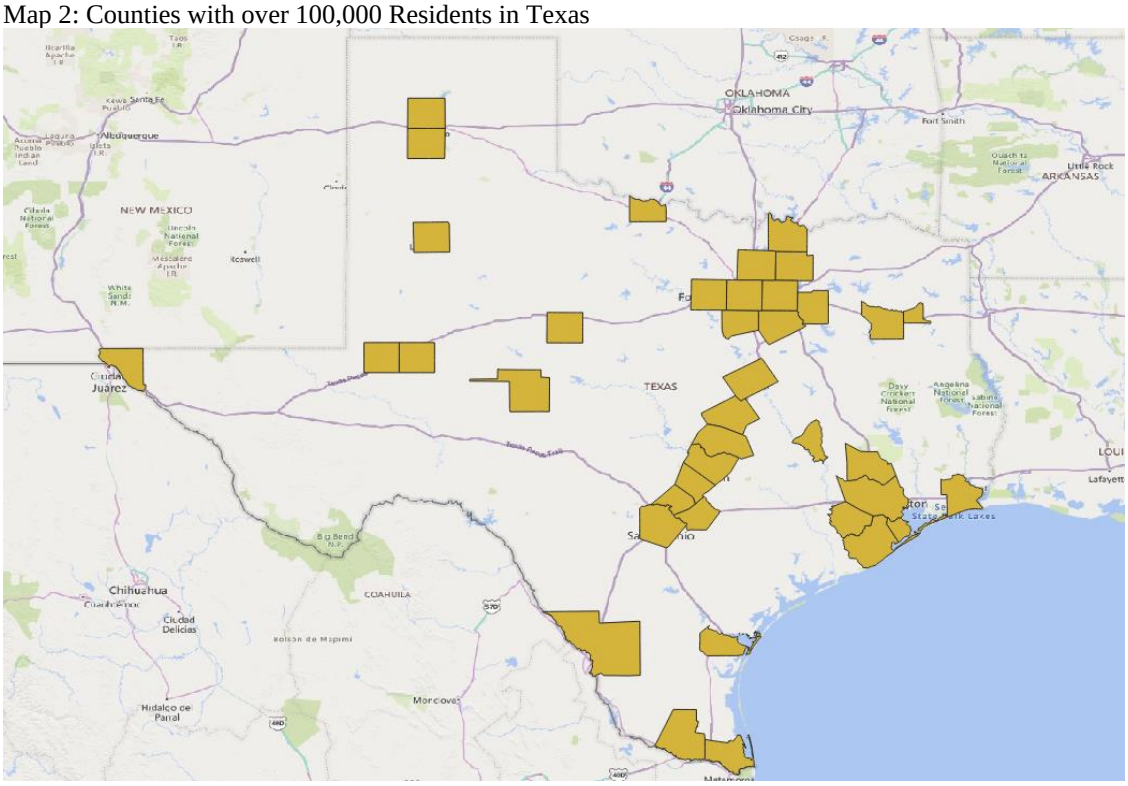
U.S. Census Bureau (2019). TIGER/Line shapefile, 2016, series information for the current county subdivision state-based shapefile. Retrieved from https://catalog.data.gov/dataset/tiger-line-shapefile-2016-series-information-for-the-current-county-subdivision-state-based-sha.

Walker, K. (2020). Tidycensus: Load US census boundary and attribute data as 'tidyverse' and 'sf'-ready data frames. R package version 0.9.6. https://CRAN.R project.org/package=tidycensus

Wickham, H., Hester, J., and Francois, R. (2018). readr: Read Rectangular Text Data. R package version 1.3.1. https://CRAN.R-project.org/package=readr

Appendix A: Tables and Maps

Map 2: Counties with over 100,000 Residents in Texas



Map 3: Census Tracts within Counties with over 100,000 Residents in Texas

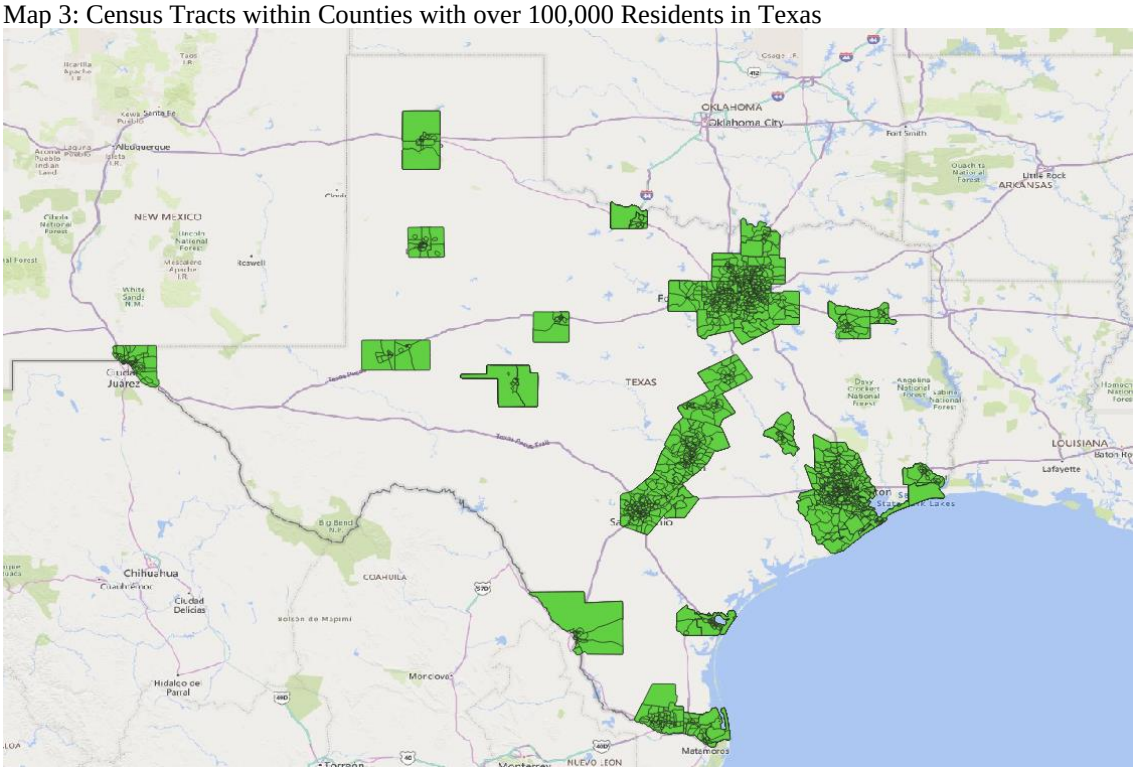


Table 2: Linear Relationship between Categories (Model 2)

Dataset	relationship	y-intercept	slope	p-value	significance
All Data, without 0% or 100% unimpaired (n = 1125)	% white - % not impaired	33.35629	0.22243	< 2e-16	**Statistically significant**
	% black - % not impaired	54.31539	-0.37453	3.14e-11	**Statistically significant**
	% Hispanic - % not impaired	47.44704	0.06010	0.0833	Statistically insignificant
	median income - % not impaired	4.758e+01	3.135e-05	0.253	Statistically insignificant

Table 3: Linear Relationship between Categories (Model 3)

Dataset	relationship	y-intercept	slope	p-value	significance
1000 census tracts with the most stream/lake lines (n = 1000)	% white - % not impaired	11.21033	0.47767	6.88e-11	**Statistically significant**
	% black - % not impaired	52.51527	-0.54619	8.77e-09	**Statistically significant**
	% Hispanic - % not impaired	43.07120	0.05889	0.249	Statistically insignificant
	median income - % not impaired	4.512e+01	4.154e-06	0.921	Statistically insignificant

Appendix B: Python Code

Merge_all.py

```
import arcpy

arcpy.env.workspace = "C:/Users/Owner/Documents/FALL2020/GIS
Programming/programming_project"

arcpy.env.overwriteOutput = True

arcpy.Merge_management(["clipped_all.shp", "clipped_am_indian.shp",
                        "clipped_asian.shp", "clipped_black.shp",
                        "clipped_hispanic.shp", "clipped_white.shp",
                        "clipped_med_income.shp"],
                       "C:\\Users\\Owner\\Documents\\FALL2020\\GIS
Programming\programming_project\\all_merged.shp",
                       "", "ADD_SOURCE_INFO")
```

Pop_county.py

```
import arcpy

arcpy.env.workspace = "C:/Users/Owner/Documents/FALL2020/GIS
Programming/programming_project"

arcpy.env.overwriteOutput = True

arcpy.Select_analysis("counties.shp", "C:/Users/Owner/Documents/FALL2020/GIS
Programming/programming_project/counties_selected.shp", '"estimate" >= 100000')

# This successfully created a new file that had only counties that have a population greater than
100,000 residents
# counties.shp was previously created in R through the use of tidycensus
```

Clipping_everything.py

```
# This is actually clipping all of the files toegether
import arcpy

arcpy.env.workspace = "C:/Users/Owner/Documents/FALL2020/GIS
Programming/programming_project"

arcpy.env.overwriteOutput = True
```

```
arcpy.Clip_analysis("everything.shp", "counties_selected.shp",
"C:/Users/Owner/Documents/FALL2020/GIS
Programming/programming_project/clipped_everything.shp")
```

Buffered_everything.py

```
import arcpy
arcpy.env.workspace = "C:/Users/Owner/Documents/FALL2020/GIS
Programming/programming_project"

arcpy.env.overwriteOutput = True

arcpy.Buffer_analysis("clipped_everything.shp", "buffered_everything.shp", "1 Mile", "FULL",
                      "ROUND")
```

Polygon to polyline.py

```
import arcpy
arcpy.env.overwriteOutput = True
arcpy.env.workspace = "C:/Users/Owner/Documents/FALL2020/GIS
Programming/programming_project/lakes"
polygon_fc = "Surface_Water.shp"
polyline_fc = "polyline_lakes.shp"

arcpy.PolygonToLine_management(polygon_fc,
                               polyline_fc,
                               "IGNORE_NEIGHBORS")

river_polyline = "Surface_Water_River.shp"
lake_polyline = polyline_fc
# probably could combine lakes and rivers using shapelength?

# Create FieldMappings object to manage merge output fields
fieldMappings = arcpy.FieldMappings()

fieldMappings.addTable(river_polyline)
fieldMappings.addTable(lake_polyline)

# Remove all output fields from the field mappings, except fields "Street_Class", "Street_Name",
& "Distance"
for field in fieldMappings.fields:
   if field.name not in ["OBJECTID","IMPAIRED","SEG_TYPE", "SHAPELEN"]:
      fieldMappings.removeFieldMap(fieldMappings.findFieldMapIndex(field.name))

# Use Merge tool to move features into single dataset
```

```
updated_poly = "C:/Users/Owner/Documents/FALL2020/GIS
Programming/programming_project/lakes/updated_polylines.shp"
arcpy.Merge_management([river_polyline, lake_polyline], updated_poly, fieldMappings)
```

clipped_polylines.py

```
import arcpy
arcpy.env.workspace = "C:/Users/Owner/Documents/FALL2020/GIS
Programming/programming_project"

arcpy.env.overwriteOutput = True

buff = "buffered_everything.shp"
poly = "updated_polylines.shp"

arcpy.Clip_analysis(poly, buff, "clipped_polylines.shp")


# now calculated the length of clipped polylines in QGIS....If I have time, I'll recalculate it here
as well
```

Pandas.py

```
import pandas as pd

x = r"C:\Users\Owner\Documents\FALL2020\GIS
Programming\programming_project\all_data.csv"

df = pd.read_csv(x)

pd.set_option("display.max.columns", None)

df.head()

df2 = df.sort_values(by = 'all', ascending = False)

df3= df2.iloc[0:50, ]

df3['percent_white'] = df3['percent_white'].astype(float)
df3['percent_am_indian'] = df3['percent_am_indian'].astype(float)
df3['percent_asian'] = df3['percent_asian'].astype(float)
df3['percent_black'] = df3['percent_black'].astype(float)
df3['percent_hispanic'] = df3['percent_hispanic'].astype(float)
df3['percent_not_impaired'] = df3['percent_not_impaired'].astype(float)
df3['percent_impaired'] = df3['percent_impaired'].astype(float)
```

```
df3.plot.scatter(x =  "percent_white", y = "percent_not_impaired")

df3.plot.scatter(x =  "percent_black", y = "percent_not_impaired")

df3.plot.scatter(x =  "percent_hispanic", y = "percent_not_impaired")
df3.plot.scatter(x =  "median_income", y = "percent_not_impaired")

df4= df2.iloc[0:100, ]

df4.plot.scatter(x =  "percent_white", y = "percent_not_impaired")

df4.plot.scatter(x =  "percent_black", y = "percent_not_impaired")

df4.plot.scatter(x =  "percent_hispanic", y = "percent_not_impaired")
df4.plot.scatter(x =  "median_income", y = "percent_not_impaired")

df4.to_csv(path_or_buf = r"C:\Users\Owner\Documents\FALL2020\GIS
Programming\programming_project\df4.csv", index=False)

# will run linear regression in R

df5 = df2.iloc[0:1000, ]
df6 = df2.iloc[3170:4170, ]
df5.to_csv(path_or_buf = r"C:\Users\Owner\Documents\FALL2020\GIS
Programming\programming_project\df5.csv", index=False)
df6.to_csv(path_or_buf = r"C:\Users\Owner\Documents\FALL2020\GIS
Programming\programming_project\df6.csv", index=False)



# top 1000 white
df_white = df.sort_values(by = 'percent_white', ascending = False)
df_white = df_white.query('percent_impaired != 1 & percent_impaired != 0')
df_white = df_white.iloc[0:100, ]
df_white.plot.scatter(x = "percent_white", y = "percent_impaired")
df_white.to_csv(path_or_buf = r"C:\Users\Owner\Documents\FALL2020\GIS
Programming\programming_project\top_100\df_white.csv",
            index = False)
# top 100 black
df_black = df.sort_values(by = 'percent_black', ascending = False)
df_black = df_black.query('percent_impaired != 1 & percent_impaired != 0')
df_black = df_black.iloc[0:100, ]
df_black.plot.scatter(x = "percent_black", y = "percent_impaired")
df_black.to_csv(path_or_buf = r"C:\Users\Owner\Documents\FALL2020\GIS
Programming\programming_project\top_100\df_black.csv",
            index = False)
```

```
# top 100 hispanic
df_hispanic = df.sort_values(by = 'percent_hispanic', ascending = False)
df_hispanic = df_hispanic.query('percent_impaired != 1 & percent_impaired != 0')
df_hispanic = df_hispanic.iloc[0:100, ]
df_hispanic.plot.scatter(x = "percent_hispanic", y = "percent_impaired")
df_hispanic.to_csv(path_or_buf = r"C:\Users\Owner\Documents\FALL2020\GIS
Programming\programming_project\top_100\df_hispanic.csv",
            index = False)
# top 100 median income
df_median = df.sort_values(by = 'median_income', ascending = False)
df_median = df_median.query('percent_impaired != 1 & percent_impaired != 0')
df_median = df_median.iloc[0:1000, ]
df_median.plot.scatter(x = "median_income", y = "percent_impaired", figsize=(12,8))
df_median.to_csv(path_or_buf = r"C:\Users\Owner\Documents\FALL2020\GIS
Programming\programming_project\top_100\df_median.csv",
            index = False)

df.plot.scatter(x = "median_income", y = "percent_not_impaired", figsize=(12,8), )
```

Appendix C: R Code

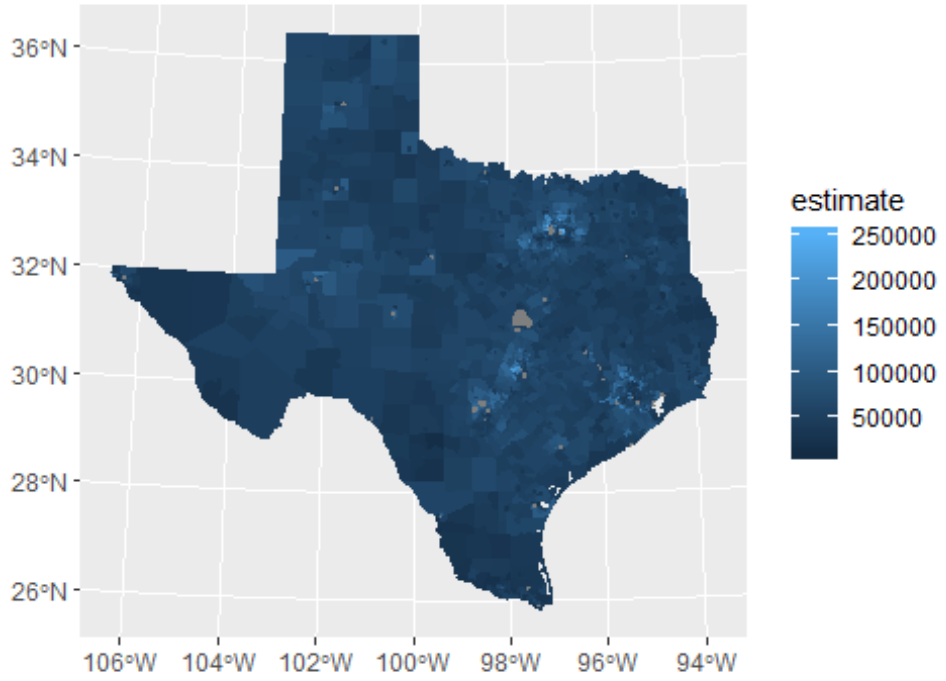
Matthew Ungaro

Api_attempt.r

10/6/2020

```
library(tidyverse)

## -- Attaching packages ---------------------------------------------------
------------------------------------------------------------------------ ti
dyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------
------------------------------------------------------------------ tidyvers
e_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(sf)

## Warning: package 'sf' was built under R version 3.6.3

## Linking to GEOS 3.6.1, GDAL 2.2.3, PROJ 4.9.3

library(tidycensus)

## Warning: package 'tidycensus' was built under R version 3.6.3

census_api_key('39e37ca2bcc68f7fb19aec0b37bea59d93ce7337', overwrite = TRUE)

## To install your API key for use in future sessions, run this function with
`install = TRUE`.

all <- get_acs(geography = "tract", variables = "B02001_001", state = "TX", g
eometry = TRUE)

## Getting data from the 2014-2018 5-year ACS

## Downloading feature geometry from the Census website.  To cache shapefiles
for use in future sessions, set `options(tigris_use_cache = TRUE)`.

##   |

white <- get_acs(geography = "tract", variables = "B02001_002", state = "TX",
geometry = TRUE)
```

```
## Getting data from the 2014-2018 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles
for use in future sessions, set `options(tigris_use_cache = TRUE)`.

black <- get_acs(geography = "tract", variables = "B02001_003", state = "TX",
geometry = TRUE)

## Getting data from the 2014-2018 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles
for use in future sessions, set `options(tigris_use_cache = TRUE)`.

hispanic1 <- get_acs(geography = "tract", variables = "B03001_003", state = "
TX", geometry = TRUE)

## Getting data from the 2014-2018 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles
for use in future sessions, set `options(tigris_use_cache = TRUE)`.

asian <- get_acs(geography = "tract", variables = "B02001_005", state = "TX",
geometry = TRUE)

## Getting data from the 2014-2018 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles
for use in future sessions, set `options(tigris_use_cache = TRUE)`.

am_indian <- get_acs(geography = "tract", variables = "B02001_004", state = "
TX", geometry = TRUE)

## Getting data from the 2014-2018 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles
for use in future sessions, set `options(tigris_use_cache = TRUE)`.

med_income <- get_acs(geography = "tract", variables = "B19013_001", state =
"TX", geometry = TRUE)

## Getting data from the 2014-2018 5-year ACS
## Downloading feature geometry from the Census website.  To cache shapefiles
for use in future sessions, set `options(tigris_use_cache = TRUE)`.

ggplot(med_income, aes(fill = estimate, color = estimate)) +
  geom_sf() +
  coord_sf(crs = 26914)
```




```
# some blank spots for some reason

all <- read_csv( "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\pro
gramming_project\\all.csv")

## Parsed with column specification:
## cols(
##   GEOID = col_double(),
##   NAME = col_character(),
##   variable = col_character(),
##   estimate = col_double(),
##   moe = col_double(),
##   geometry = col_character()
## )

#write_csv(white, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\pr
ogramming_project\\white.csv")
#write_csv(black, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\pr
ogramming_project\\black.csv")
#write_csv(hispanic1, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming
\\programming_project\\hispanic.csv")
#write_csv(asian, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\pr
ogramming_project\\asian.csv")
#write_csv(am_indian, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming
\\programming_project\\am_indian.csv")
#write_csv(med_income, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programmin
g\\programming_project\\med_income.csv")
```

```
#st_write(white, "white.shp")
#st_write(all, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\programming_project\\test.shp")
#st_write(black, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\programming_project\\black.shp")
#st_write(hispanic1, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\programming_project\\hispanic.shp")
#st_write(asian, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\programming_project\\asian.shp")
#st_write(am_indian, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\programming_project\\am_indian.shp")
#st_write(med_income, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\programming_project\\med_income.shp")

# tidycensus - county populations

counties <- get_acs(geography = "county", variables = "B02001_001", state = "TX", geometry = TRUE)

## Getting data from the 2014-2018 5-year ACS

## Downloading feature geometry from the Census website.  To cache shapefiles for use in future sessions, set `options(tigris_use_cache = TRUE)`.

##   |

# perfect!
#sf::st_write(counties, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\programming_project\\counties.shp")

# easier way to deal with different groups

# I manually copied and pasted columns in the csvs to create one complete csv

# Now I'll need to combine them

everything <- all

everything$estimate_am_indian <- am_indian$estimate
everything$estimate_asian <- asian$estimate
everything$estimate_black <- black$estimate
everything$estimate_hispanic <- hispanic1$estimate
everything$estimate_med_income <- med_income$estimate
everything$estimate_white <- white$estimate


#sf::st_write(everything, "C:\\Users\\Owner\\Documents\\FALL2020\\GIS Programming\\programming_project\\everything.shp")
```

Linear_reg.r

```
library(tidyverse)
read_csv("C:\\Users\\Owner\\Documents\\FALL2020\\GIS
Programming\\programming_project\\df4.csv") -> data1

summary(lm(percent_impaired ~ percent_white, data1))
summary(lm(percent_impaired ~ percent_black, data1))
summary(lm(percent_impaired ~ percent_asian, data1))
summary(lm(percent_impaired ~ percent_am_indian, data1))
summary(lm(percent_impaired ~ percent_hispanic, data1))
summary(lm(percent_impaired ~ median_income, data1))

# maybe if I grabbed everything? I doubt that would really help however. Maybe if I just grabbed
the top 20?

read_csv("C:\\Users\\Owner\\Documents\\FALL2020\\GIS
Programming\\programming_project\\df2.csv") -> data_all

data_all$percent_not_impaired*100 -> data_all$percent_not_impaired
data_all$percent_am_indian*100 -> data_all$percent_am_indian
data_all$percent_asian*100 -> data_all$percent_asian
data_all$percent_black*100 -> data_all$percent_black
data_all$percent_white*100 -> data_all$percent_white
data_all$percent_hispanic*100 -> data_all$percent_hispanic

q <- (lm(percent_not_impaired ~ percent_white, data_all))
r <- (lm(percent_not_impaired ~ percent_black, data_all))
s <- (lm(percent_not_impaired ~ percent_hispanic, data_all))
ss <- (lm(percent_not_impaired ~ median_income, data_all))

par(mfrow=c(2,2))
plot(data_all$percent_white, data_all$percent_not_impaired, xlab = "Percent White Population",
    ylab = "Percent Not Impaired", cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
abline(q, col = "purple")

plot(data_all$percent_black, data_all$percent_not_impaired, xlab = "Percent Black or African
American Population",
    ylab = "Percent Not Impaired", cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
abline(r, col = "purple")

plot(data_all$percent_hispanic, data_all$percent_not_impaired, xlab = "Percent Hispanic
Population",
    ylab = "Percent Not Impaired", cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
abline(s, col = "purple")

plot(data_all$median_income, data_all$percent_not_impaired, xlab = "Median Income",
```

```
    ylab = "Percent Not Impaired", cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1.5)
abline(ss, col = "purple")

summary(q) # positive
summary(r) # negataive
summary(s) # negative
summary(ss) # negative


data_all %>% filter(percent_impaired != 0) %>% filter(percent_impaired != 1) -> filtered_all

summary(lm(percent_impaired ~ percent_white, filtered_all)) # negative
summary(lm(percent_impaired ~ percent_black, filtered_all)) # positive
summary(lm(percent_impaired ~ percent_hispanic, filtered_all)) # not statistically signficant
summary(lm(percent_impaired ~ median_income, filtered_all)) # not statistically sign.

summary(lm(percent_not_impaired ~ percent_white, filtered_all)) # positive
summary(lm(percent_not_impaired ~ percent_black, filtered_all)) # negative
summary(lm(percent_not_impaired ~ percent_hispanic, filtered_all)) # not statistically signficant
summary(lm(percent_not_impaired ~ median_income, filtered_all)) # not statistically sign.




# so with all of the data being included, we see a statistical significance
# What about instead of the top 100, the bottom 100? Or the top 1000?

# TOP 1000
df5 <- read_csv("C:\\Users\\Owner\\Documents\\FALL2020\\GIS
Programming\\programming_project\\df5.csv")

df5$percent_not_impaired*100 -> df5$percent_not_impaired
df5$percent_am_indian*100 -> df5$percent_am_indian
df5$percent_asian*100 -> df5$percent_asian
df5$percent_black*100 -> df5$percent_black
df5$percent_white*100 -> df5$percent_white
df5$percent_hispanic*100 -> df5$percent_hispanic
t <- (lm(percent_not_impaired ~ percent_white, df5)) # positive
u <- (lm(percent_not_impaired ~ percent_black, df5)) # negative
v <- (lm(percent_not_impaired ~ percent_hispanic, df5))
w <- (lm(percent_not_impaired ~ median_income, df5))

# statistically significant
plot(df5$percent_white, df5$percent_impaired) #negative (meaning more white, less impaired)
abline(t)
plot(df5$percent_black, df5$percent_impaired) # positive
```

```
abline(u)
plot(df5$percent_hispanic, df5$percent_impaired) # negative
abline(v)



df5 %>% filter(percent_impaired != 0) %>% filter(percent_impaired != 1) ->x
x %>% ggplot(mapping = aes(percent_white, percent_impaired))+geom_point()
x1 <- (lm(percent_impaired ~ percent_white, x))
plot(x$percent_white, x$percent_impaired)
abline(x1)
x %>% ggplot(mapping = aes(percent_black, percent_impaired))+geom_point()
x1 <- (lm(percent_impaired ~ percent_black, x))
plot(x$percent_black, x$percent_impaired)
abline(x1)

# BOTTOM 1000
df_low <- read_csv("C:\\Users\\Owner\\Documents\\FALL2020\\GIS
Programming\\programming_project\\df6.csv")
#df_low %>% filter(percent_impaired != 0) %>% filter(percent_impaired != 1) ->df_low

w <- (lm(percent_impaired ~ percent_white, df_low)) # negative
x <- (lm(percent_impaired ~ percent_black, df_low)) # positive
y <- (lm(percent_impaired ~ percent_hispanic, df_low)) # positive
plot(df_low$percent_hispanic, df_low$percent_impaired)
abline(y)



# one more thing - I looked at each of them by top 100 of hispanic, black, white, and top 1000
median income
his <- read_csv("C:\\Users\\Owner\\Documents\\FALL2020\\GIS
Programming\\programming_project\\top_100\\df_hispanic.csv")
bla <- read_csv("C:\\Users\\Owner\\Documents\\FALL2020\\GIS
Programming\\programming_project\\top_100\\df_black.csv")
whi <- read_csv("C:\\Users\\Owner\\Documents\\FALL2020\\GIS
Programming\\programming_project\\top_100\\df_white.csv")
med <- read_csv("C:\\Users\\Owner\\Documents\\FALL2020\\GIS
Programming\\programming_project\\top_100\\df_median.csv")
summary(lm(percent_impaired ~ percent_hispanic, his)) # not significant
summary(lm(percent_impaired ~ percent_black, bla)) # not sig
summary(lm(percent_impaired ~ percent_white, whi)) # not sig
summary(lm(percent_impaired ~ median_income, med)) # not sig
```