

# Griff Volker Final Project

Griff Volker

5/21/23

##Griff Volker Final Project The primary objective of this project is to analyze the population dynamics in Baltimore City over time. We examining the population changes within the city's census tracts from 2010 to 2021. To accomplish this, we will utilize R for data analysis and export the results into QGIS, to create maps of the population change by taking the difference between 2010 and 2021 population in the data layer's the attribute table.

In addition to studying population changes, we will investigate the significance of transit and bus stops within Baltimore City. The importance of transit and bus stops allows residents mobility within the city. By understanding how population changes correlate with the presence of bus stops, we can gain insights the census tract we selected.

One census tract will be selected that experience a significant positive population change in addition to possess a high number of bus stops. This allows us to focus on areas where both population growth and transportation infrastructure coincide. Subsequently, we will examine whether the bus stops within these selected census tracts are clustered compared to all of Baltimore city. We will then prove that the number of bus stops in Baltimore is related to the total population of census tracts. We will then examine clustering of all bus stops in Baltimore city.

This project aims to comprehensively assess the population dynamics in Baltimore City, highlighting census tract population changes, transit infrastructure, and the spatial distribution of bus stops.

```
library(tidycensus)
library(tmap)
```

```
## Warning: package 'tmap' was built under R version 4.2.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.2      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr   1.5.0
## ✓ ggplot2    3.4.2      ✓ tibble     3.2.1
## ✓ lubridate  1.9.2      ✓ tidyr      1.3.0
## ✓ purrr      1.0.1
## — Conflicts — tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(dplyr)
library(ggplot2)
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.2.3
```

```
library(tidycensus)
library(tmap)
library(tigris)
```

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

```
library(tmap)
library(tidyverse)
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.2.3
```

```
## Linking to GEOS 3.9.3, GDAL 3.5.2, PROJ 8.2.1; sf_use_s2() is TRUE
```

```
library(tibble)
library(spatstat)
```

```
## Warning: package 'spatstat' was built under R version 4.2.3
```

```
## Loading required package: spatstat.data
```

```
## Warning: package 'spatstat.data' was built under R version 4.2.3
```

```
## Loading required package: spatstat.geom
```

```
## Warning: package 'spatstat.geom' was built under R version 4.2.3
```

```
## spatstat.geom 3.2-1  
## Loading required package: spatstat.random
```

```
## Warning: package 'spatstat.random' was built under R version 4.2.3
```

```
## spatstat.random 3.1-5  
## Loading required package: spatstat.explore
```

```
## Warning: package 'spatstat.explore' was built under R version 4.2.3
```

```
## Loading required package: nlme  
##  
## Attaching package: 'nlme'  
##  
## The following object is masked from 'package:dplyr':  
##  
##     collapse  
##  
## spatstat.explore 3.2-1  
## Loading required package: spatstat.model
```

```
## Warning: package 'spatstat.model' was built under R version 4.2.3
```

```
## Loading required package: rpart  
## spatstat.model 3.2-4  
## Loading required package: spatstat.linnet
```

```
## Warning: package 'spatstat.linnet' was built under R version 4.2.3
```

```
## spatstat.linnet 3.1-1  
##  
## spatstat 3.0-6  
## For an introduction to spatstat, type 'beginner'
```

```
library(spatstat.utils)
```

```
## Warning: package 'spatstat.utils' was built under R version 4.2.3
```

```
library(spatstat.explore)  
library(spatstat.geom)  
library(maptools)
```

```
## Warning: package 'maptools' was built under R version 4.2.3
```

```
## Loading required package: sp
## Checking rgeos availability: TRUE
## Please note that 'maptools' will be retired during 2023,
## plan transition at your earliest convenience;
## some functionality will be moved to 'sp'.
```

```
library(tmap)
library(stars)
```

```
## Warning: package 'stars' was built under R version 4.2.3
```

```
## Loading required package: abind
```

```
census_api_key("3a1aff440e9b2674e8f5b40779f00cf5f8d4fd59", overwrite = TRUE)
```

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

##Getting the variable for total population from the 5 year ACS First, we will get the variable for the total population from the 5 year ACS from 2017-2021

```
acs_2021 <- load_variables(year = 2021, #the year of our data from 2017-2021
                           dataset = "acs5",
                           cache = TRUE)
```

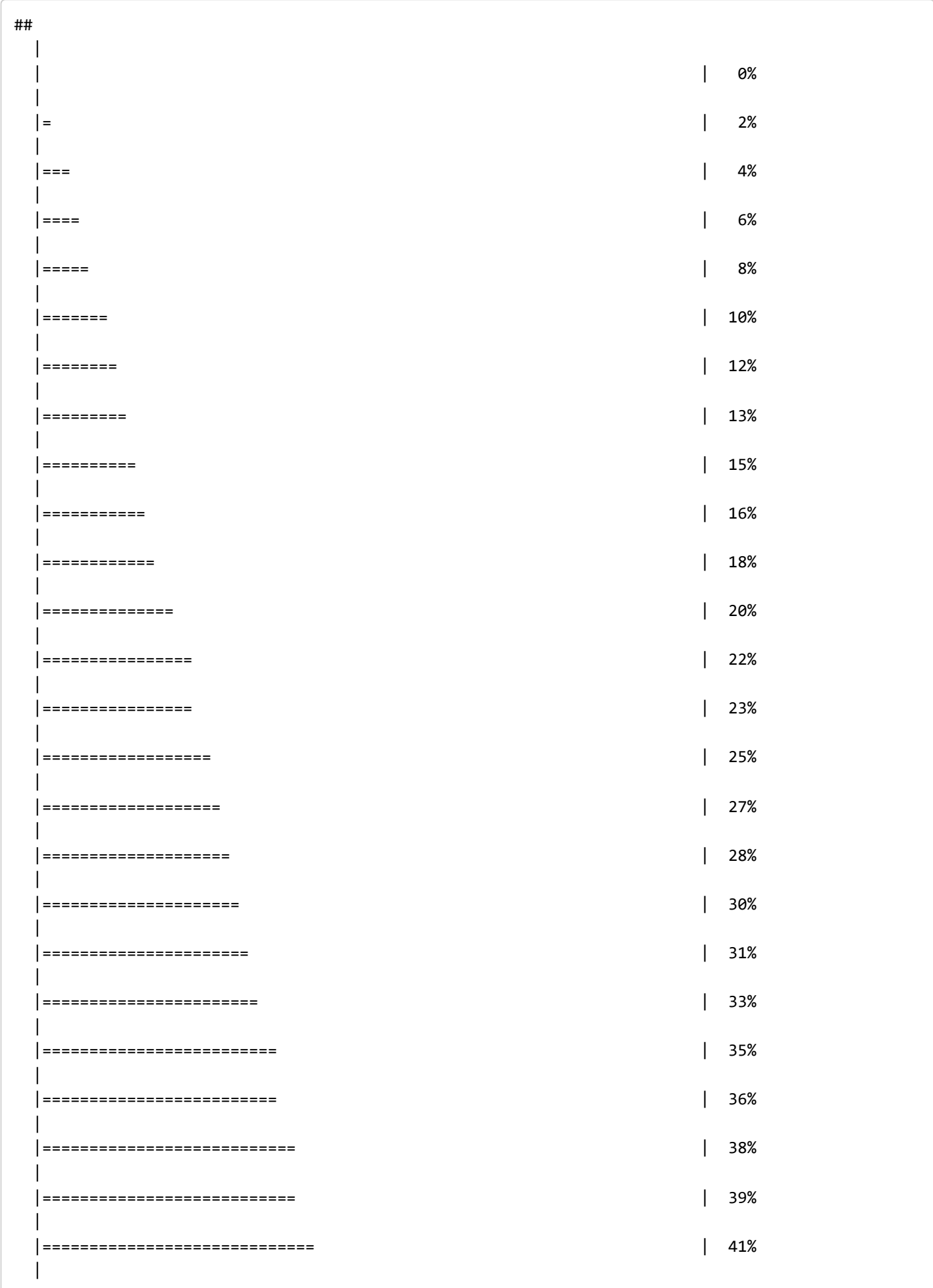
*#The variable for total population is "B01003\_001". We will use this as our starting point for the population of Baltimore City*

##Obtaining the 5 Year ACS of Baltimore City The total population obtained from the 5 year ACS will be our reference for population change in the city.

```
balt_population_5_Year_ACS <- get_acs(
  geography = "tract", #we will get the census tracts for Baltimore city
  state = "MD",
  county = "Baltimore City",
  variables = "B01003_001", #total population variable, estimate column is the total population
  year = 2021,
  geometry=TRUE) #year of our data
```

```
## Getting data from the 2017-2021 5-year ACS
```

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



=====	43%
=====	44%
=====	46%
=====	47%
=====	49%
=====	51%
=====	52%
=====	54%
=====	55%
=====	57%
=====	59%
=====	60%
=====	62%
=====	63%
=====	65%
=====	67%
=====	68%
=====	70%
=====	71%
=====	73%
=====	74%
=====	75%
=====	77%
=====	78%
=====	80%
=====	82%

=====	83%
=====	85%
=====	86%
=====	88%
=====	90%
=====	91%
=====	93%
=====	94%
=====	96%
=====	98%
=====	99%
=====	100%

##2010 Decennaial Data for Baltimore City We will obtain the 2010 decennial data to be compared to the 5 year ACS data.

```
balt_total_population_10 <- get_decennial(
  geography = "tract", #geography
  state = "MD",
  county = "Baltimore City",
  variables = "P001001", #our variable for getting the total population for Baltimore City from
the 2010 decennial
  year = 2010
)
```

## Getting data from the 2010 decennial Census

## Using Census Summary File 1

##Joining the data In order to complete our population change analysis in QGIS, we will have to join the data together and take out columns we do not need. The population change analysis is completed in QGIS because I am more familiar with calculating differences in QGIS than in R - the population changes per census tract will include negative numbers, which we did not review how to calculate in R this semester.

```

balt_population_acs_v_decennial_correct <- merge(balt_population_5_Year_ACS,balt_total_population_10, by.x = "GEOID", by.y ="GEOID") #joining the tables together

Balt_pop_acs_V_decennial_correct = subset(balt_population_acs_v_decennial_correct, select = -c(variable.x, moe, NAME.y, variable.y)) #deleting the columns we don't need

#renaming the columns
colnames(Balt_pop_acs_V_decennial_correct)[2]="Name"
colnames(Balt_pop_acs_V_decennial_correct)[3]="Total_Population_2017-2021_5_Year_ACS"
colnames(Balt_pop_acs_V_decennial_correct)[4]="Total_Population_2010_Decennial"

```

##Exporting joined tables as a Shapefile We will now export our joined columns as a shape file. Our column "geometry" will produce polygons inside QGIS that we can use to produce maps of our analysis.

```
st_write(Balt_pop_acs_V_decennial_correct, "Balt_Total_Population_Change5.shp")
```

```
## Warning in abbreviate_shapefile_names(obj): Field names abbreviated for ESRI
## Shapefile driver
```

```
## Writing layer `Balt_Total_Population_Change5' to data source
## `Balt_Total_Population_Change5.shp' using driver `ESRI Shapefile'
## Writing 198 features with 4 fields and geometry type Multi Polygon.
```

##Population Change Analysis Let's analyze the Baltimore Census tract population change. After completing the population change analysis in QGIS from taking the difference between the 5 year ACS and the 2010 decennial, you can see that the histogram is not normally distributed around 0. This made it difficult to set the correct symbology and legend in my map of population change.

Let's make a line graph showing overall population change in Baltimore City.

```

balt_pop_change_csv <- read.csv("C:\\Users\\rules\\Desktop\\UMBC Classes\\GEOG687\\Griff Volker Final Project\\src\\Balt_Total_Population_Change.csv") #reading the CSV of the SHP data we calculated in QGIS

balt_pop_change<-ggplot(balt_pop_change_csv, aes(x=Year, y=Population, group=1)) +
  geom_line()+ ggtitle("Population Change Baltimore City 2010-2021") + #plotting population change over time
  geom_point()

plot(balt_pop_change)

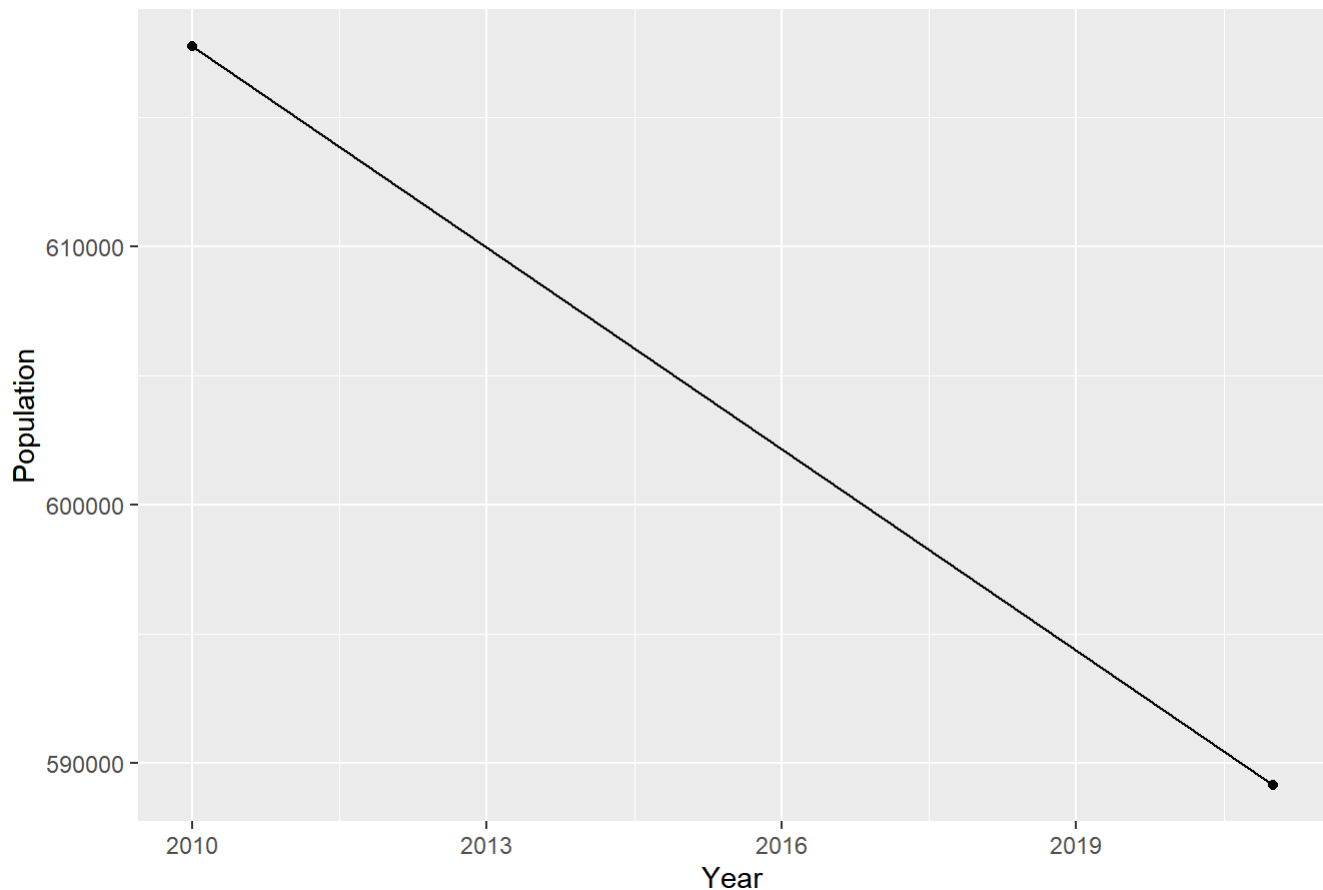
```

```
## Warning: Removed 197 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 197 rows containing missing values (`geom_point()`).
```



## Population Change Baltimore City 2010-2021



Lets make a histogram of the census tracts and how they experienced population change. You will notice that there are negative and positive changes in population and the data is not normally distributed around 0. As stated earlier, this made it difficult to map in QGIS.

```
balt_hist <-ggplot (balt_pop_change_csv, aes(x = Pop_Change)) +
  geom_histogram()

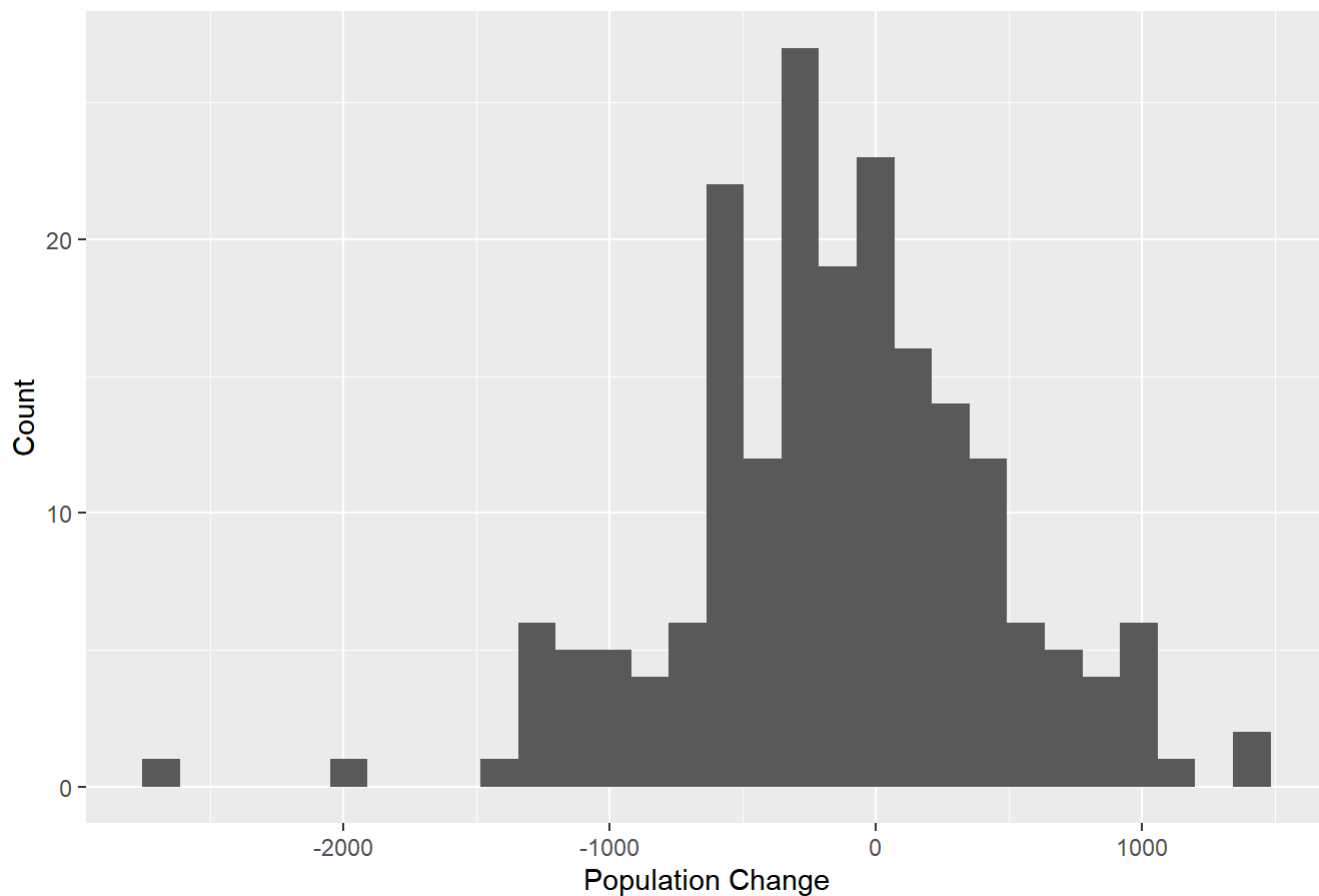
balt_hist_export<-balt_hist + ggtitle("Census Tract Population Change Baltimore City 2010-2021")
+ xlab("Population Change") + ylab("Count")

plot(balt_hist_export)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```

## Census Tract Population Change Baltimore City 2010-2021



#Further Statistical Analysis of Population Change Let's do some more analysis of changes in population in Baltimore city

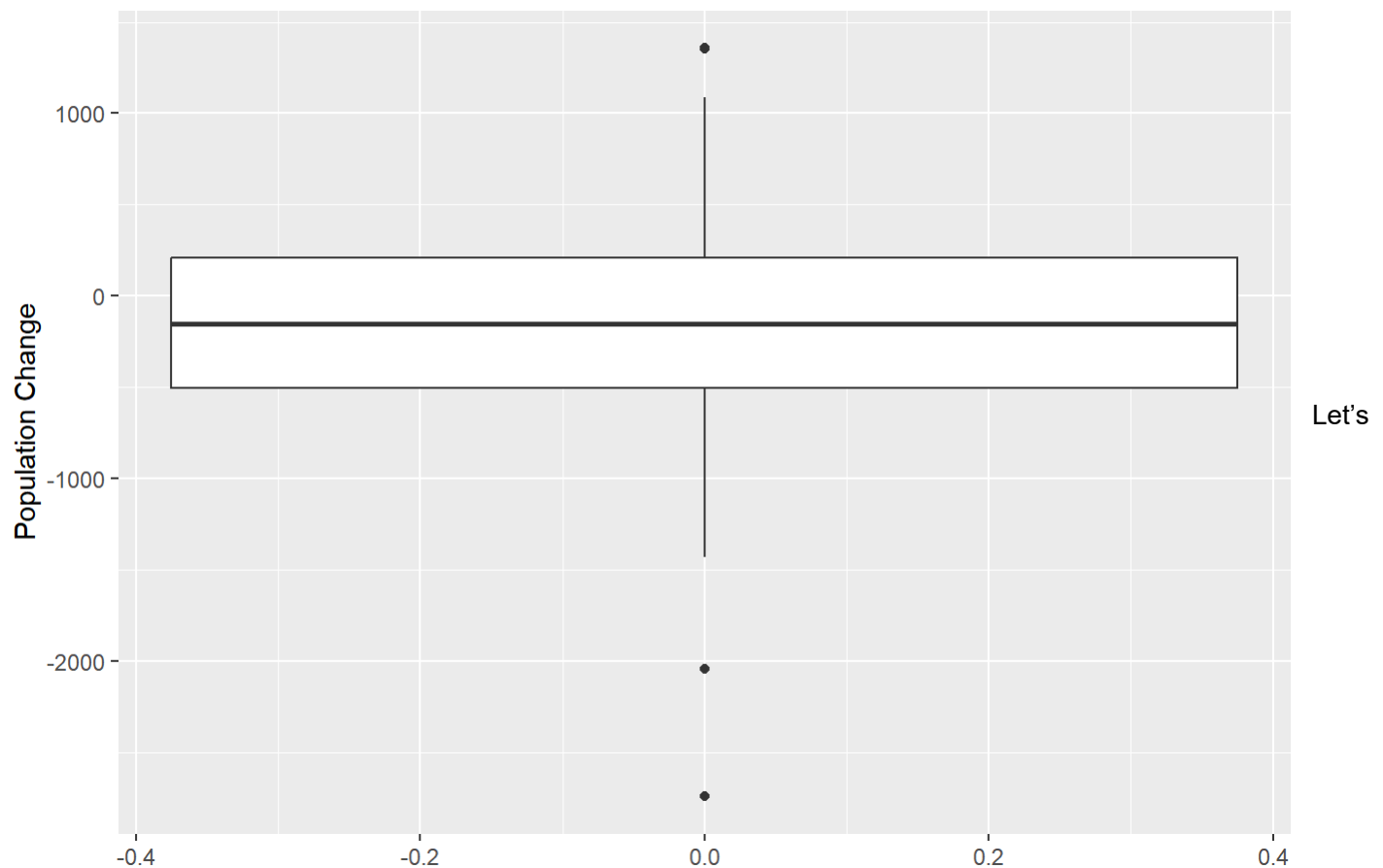
```
balt_box<-ggplot (balt_pop_change_csv, aes(y = Pop_Change)) +
  geom_boxplot() #making our box plot

balt_box_export<-balt_box + ggtitle("Census Tract Population Change Baltimore City 2010-2021") +
  ylab("Population Change")

plot(balt_box_export)
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
```

## Census Tract Population Change Baltimore City 2010-2021



continue our analysis with a scatter plot.

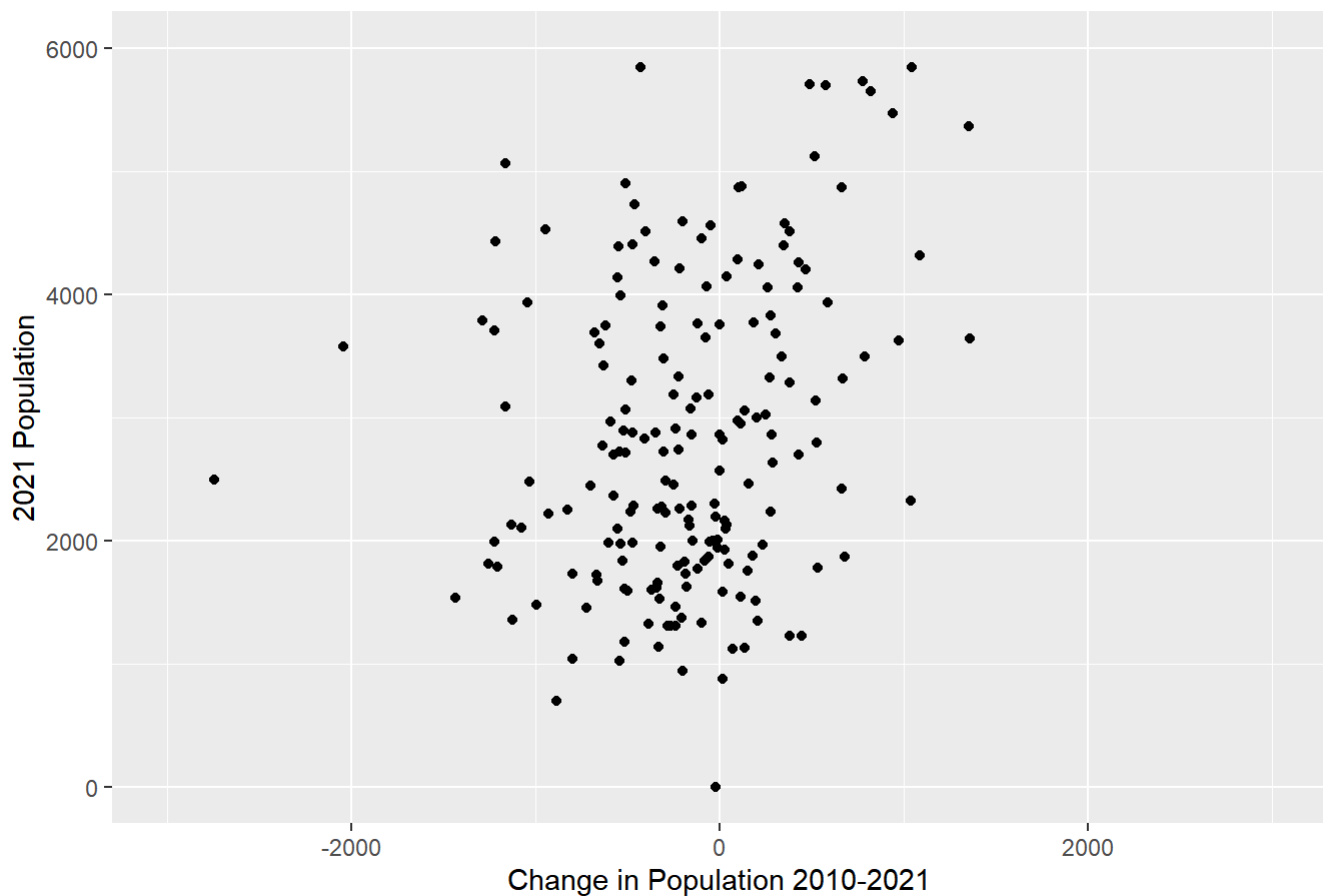
```
balt_scatter <-ggplot(balt_pop_change_csv, aes(x = Pop_Change, y = Current)) +
  geom_point()

balt_scatter_export<-balt_scatter + ggtitle("Census Tract Population Change Baltimore City 2010-
2021") + xlim(-3000, 3000)+ylim(0, 6000) + ylab("2021 Population") + xlab ("Change in Population
2010-2021")

plot(balt_scatter_export)
```

```
## Warning: Removed 9 rows containing missing values (`geom_point()`).
```

## Census Tract Population Change Baltimore City 2010-2021



##Using population data in QGIS In my QGIS project, I calculated the population change between 2010 and 2021 and mapped it out by census tract. The goal of this analysis was to find a census tract with a positive population change that we can analyze publicly available point data with. The publicly available point data we have selected for our project are bus stops, downloaded from Overpass Turbo.

Our project objective is to analyze population change by census tract and pick a census tract that we can do further analysis on. This may give insight into why people have moved there.

##Bus Stop Analysis for Census Tract 401 Census Tract 401 is our area we will be researching. This tract was chosen because it had the second largest positive population change from 2010-2021, was one of the largest population tracts, and had the highest number of bus stops in the city. After selecting Census Tract 401 as our census polygon to analyze bus stops using QGIS spatial analysis, we will now perform spatial analysis of the bus stops in that census tract.

```
balt_census_tract_401<-read_sf("C:\\Users\\rules\\Desktop\\UMBC Classes\\GEOG687\\Griff Volker Final Project\\src\\Balt_Census_401_Correct.gpkg")
```

```
bus_stops_401 <- read_csv("C:\\Users\\rules\\Desktop\\UMBC Classes\\GEOG687\\Griff Volker Final Project\\src\\Balt_Bus_Stops_401.csv")
```

```
## Rows: 89 Columns: 14
## — Column specification —————
## Delimiter: ","
## chr (6): stop_name, Routes_Ser, Distributi, Mode, Shelter, County
## dbl (8): X, Y, fid, Rider_On, Rider_Off, Rider_Tota, Stop_Rider, stop_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

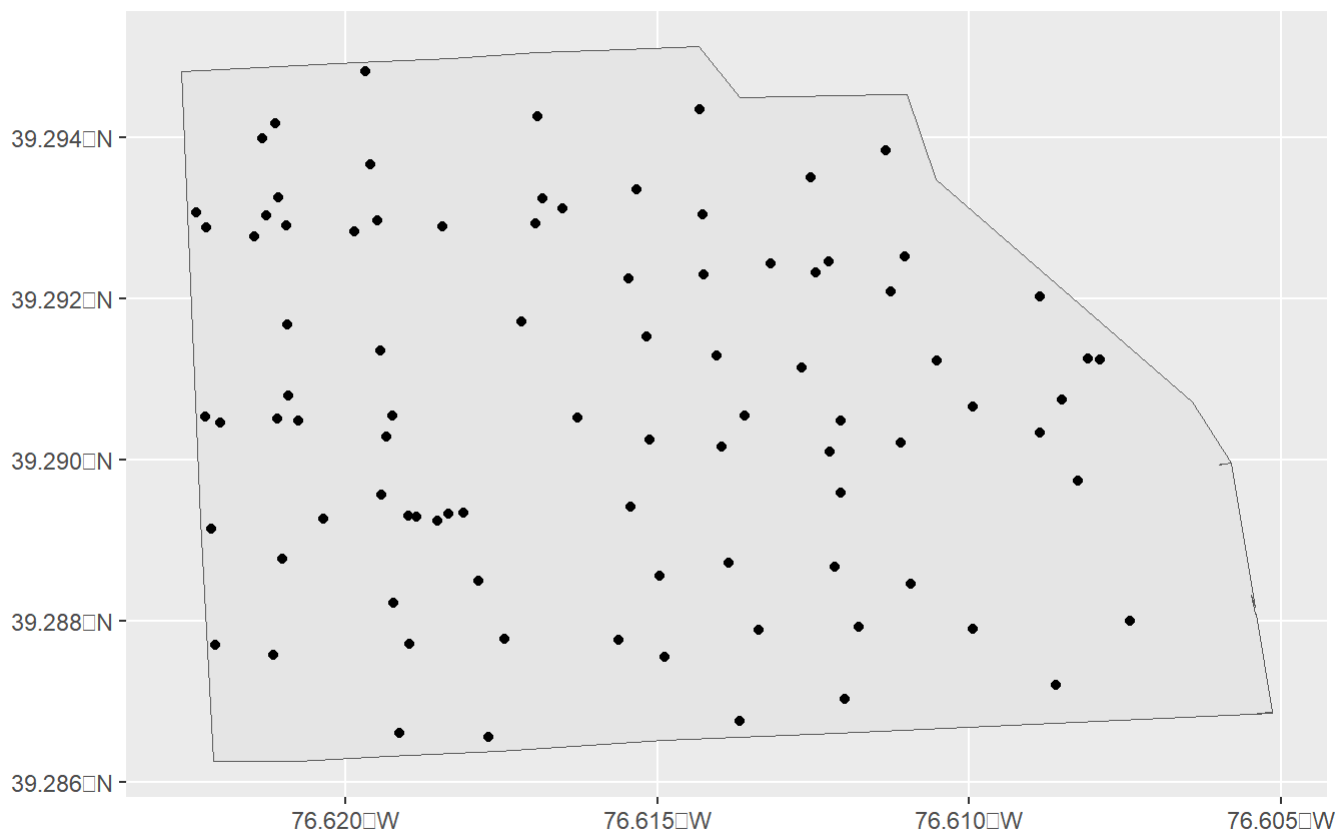
```
bus_stops_401_4326<-st_as_sf(bus_stops_401,coords=c(x="X",y="Y"), crs=4326)

bus_stops_401_correct<-st_transform(bus_stops_401_4326, crs=3857)

bus_stops_401_plot<-ggplot() + geom_sf(data=balt_census_tract_401) +
geom_sf(data=bus_stops_401_correct) + ggtitle("Bus Stops - Baltimore City Census Tract 401")

plot(bus_stops_401_plot)
```

Bus Stops - Baltimore City Census Tract 401



#Tests to Determine if bus stops are clustered After plotting the bus stop points for Census tract 401, let's run a KS test to determine if the bus stop data is clustered.

```
bus_stops_401_ppp <- as.ppp(bus_stops_401_correct)
```

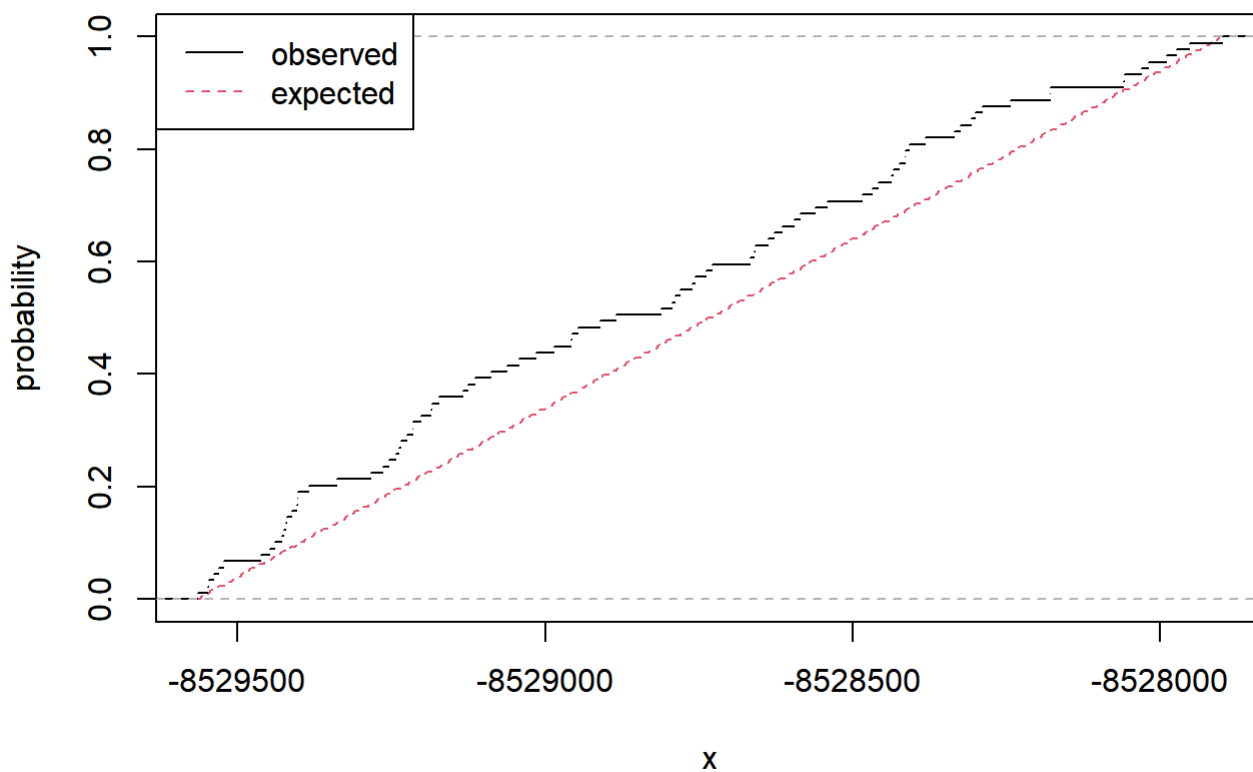
```
## Warning in as.ppp.sf(bus_stops_401_correct): only first attribute column is
## used for marks
```

```
ks_bus_401_stops <- cdf.test(bus_stops_401_ppp, "x")
```

```
## Warning in cdf.test.ppp(bus_stops_401_ppp, "x"): marks ignored
```

```
plot(ks_bus_401_stops)
```

### Spatial Kolmogorov-Smirnov test of CSR in two dimensions based on distribution of x coordinate p-value= 0.1028



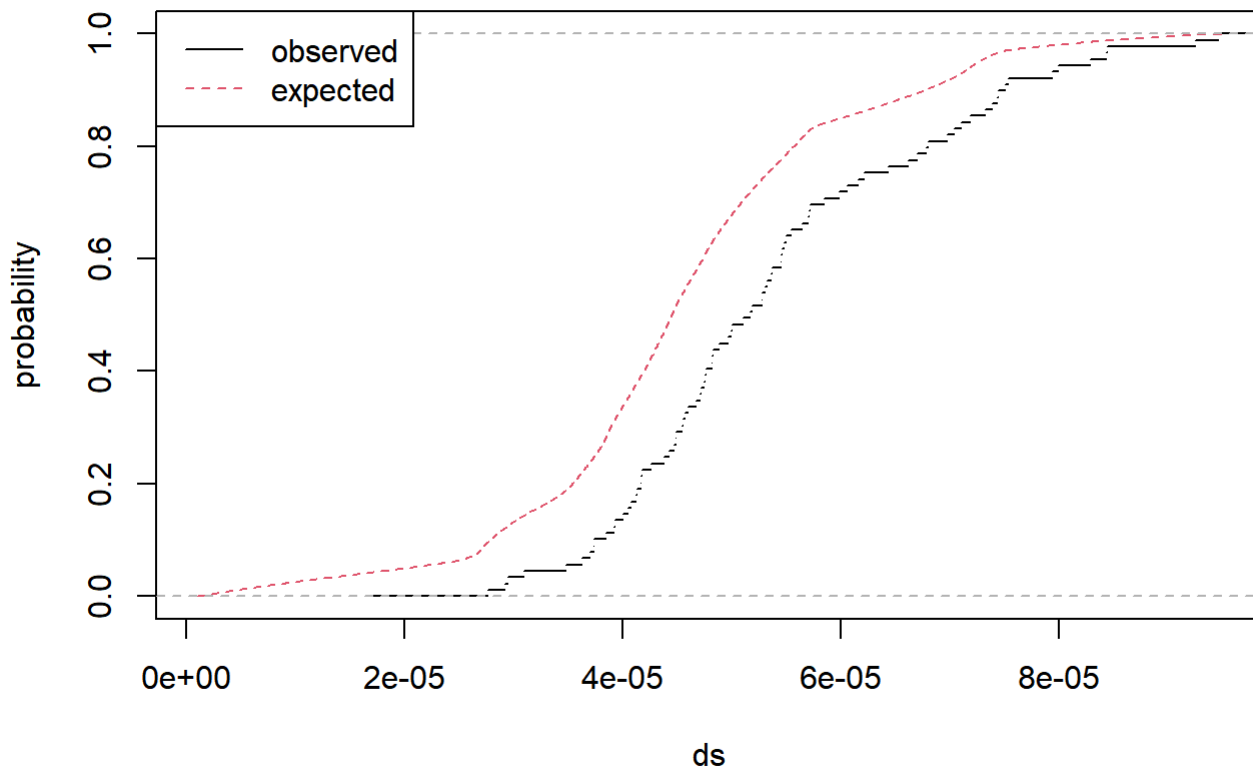
##Plotting the Density Covariate Plotting the density covariate of the bus stops in census tracts 401 will give us more information to determine if the data is clustered

```
ds <- density.ppp(bus_stops_401_ppp)
k_density_Covariate <- cdf.test(bus_stops_401_ppp, ds)
```

```
## Warning in cdf.test.ppp(bus_stops_401_ppp, ds): marks ignored
```

```
plot(k_density_Covariate)
```

## Spatial Kolmogorov-Smirnov test of CSR in two dimensions based on distribution of covariate "ds" p-value= 2.406e-05

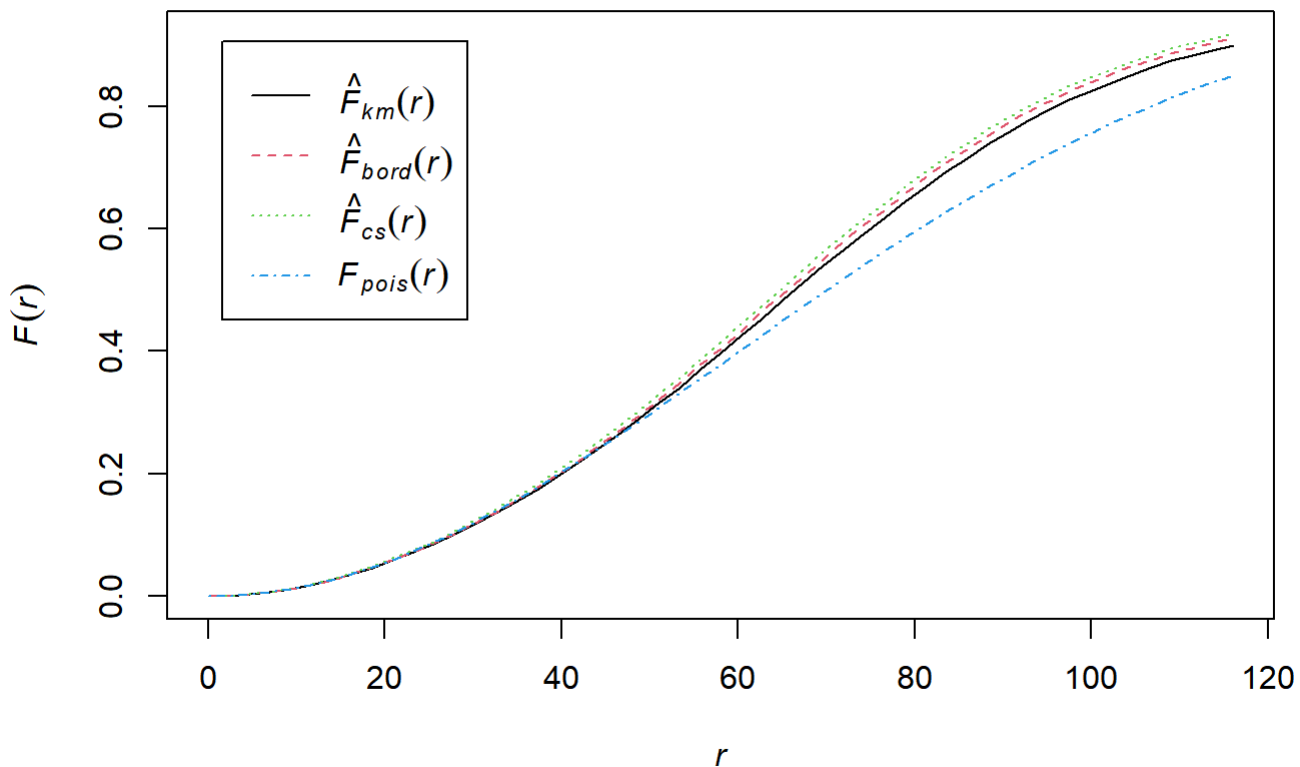


Based on the results of the KS test & density covariate test, bus stops census tracts 401 are not random. This makes sense since bus stops in this area are designed to reach as many people as possible and are intended to be spatially dispersed.

Let's run an additional F test to determine the distribution of all distances from an arbitrary point to its nearest point this is a measure of the average space left between bus stops.

```
ks_bus_401_stops_f_test <- Fest(bus_stops_401_ppp)
plot(ks_bus_401_stops_f_test)
```

## ks\_bus\_401\_stops\_f\_test



Based on our F test, bus stops in census tract 401 data appear to not be clustered because the observed distances between bus stops are shorter than the Poisson process.

##Analyzing Baltimore City Bus Stops Let's look at bus stops in Baltimore city as a whole and compare it to Census Tract 401.

```
balt_5_year_reprojected <- sf::st_transform(balt_population_5_Year_ACS, 3857) #reproject to UTM
12
balt_erase2 <- erase_water(balt_5_year_reprojected) #deleting the water
```

```
## Fetching area water data for your dataset's location...
```

```
## Erasing water area...
## If this is slow, try a larger area threshold value.
```



```

balt_5_year_pop <- stars::st_rasterize(balt_erase2["estimate"], dx = 10, dy = 10) #making our total pop as a raster
write_stars(balt_5_year_pop, "balt_5_year_pop.tif") #writing the tif
balt_5_year_pop_ras <- raster::raster("../src/balt_5_year_pop.tif") #fetching the tif
balt_5_year_pop_ras_im <- maptools::as.im.RasterLayer(balt_5_year_pop_ras) #using the map tools for the til

bus_stops_balt<-read_sf("../src/busstops2022.gpkg") #reading in bus stops
bus_stops_balt_4326<-st_as_sf(bus_stops_balt,coords=c(x="X",y="Y"), crs=4326) #setting the coordinates
bus_stops_balt_correct<-st_transform(bus_stops_balt, crs=3857) #reprojectiong
bus_stops_ppp<-as.ppp(bus_stops_balt_correct) #plotting as a ppp

```

```

## Warning in as.ppp.sf(bus_stops_balt_correct): only first attribute column is
## used for marks

```

```

ks <- cdf.test(bus_stops_ppp, balt_5_year_pop_ras_im) #running our KS test

```

```

## Warning in cdf.test.ppp(bus_stops_ppp, balt_5_year_pop_ras_im): marks ignored

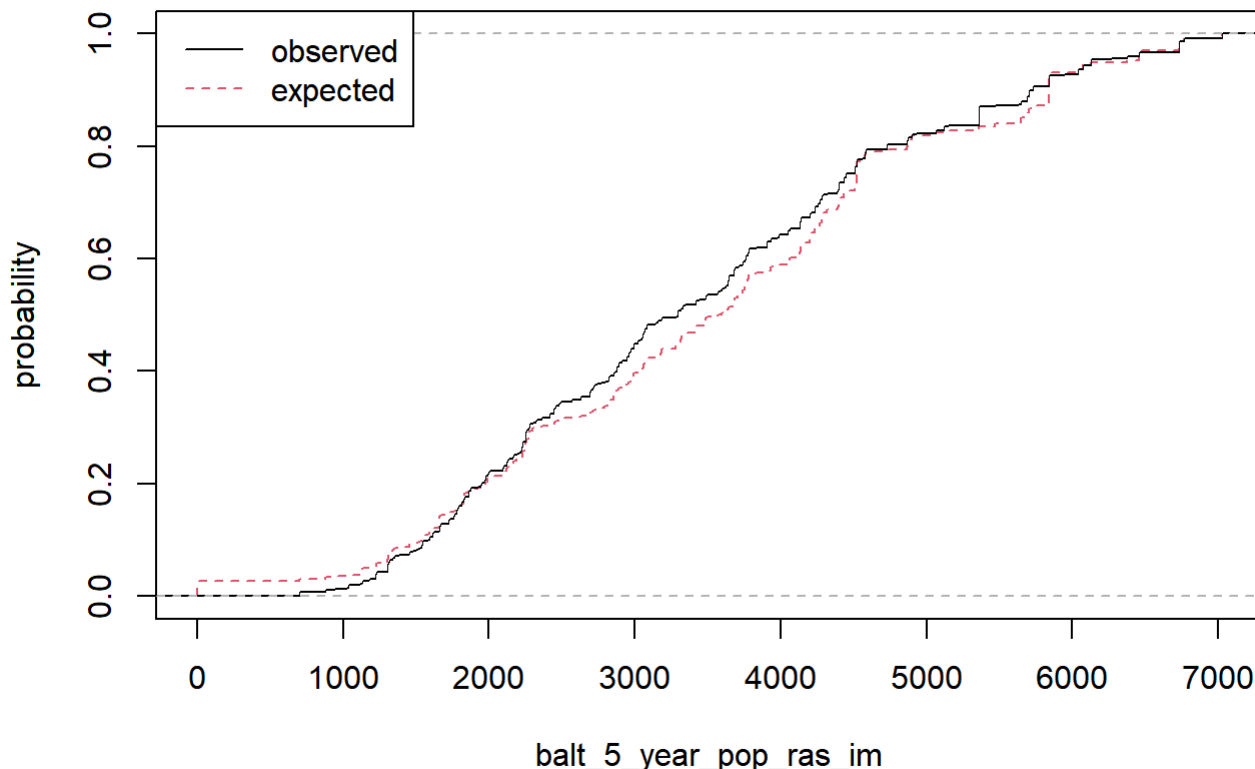
```

```

plot(ks)

```

### Spatial Kolmogorov-Smirnov test of CSR in two dimensions based on distribution of covariate "balt\_5\_year\_pop\_ras\_im" p-value= 2.11e-09



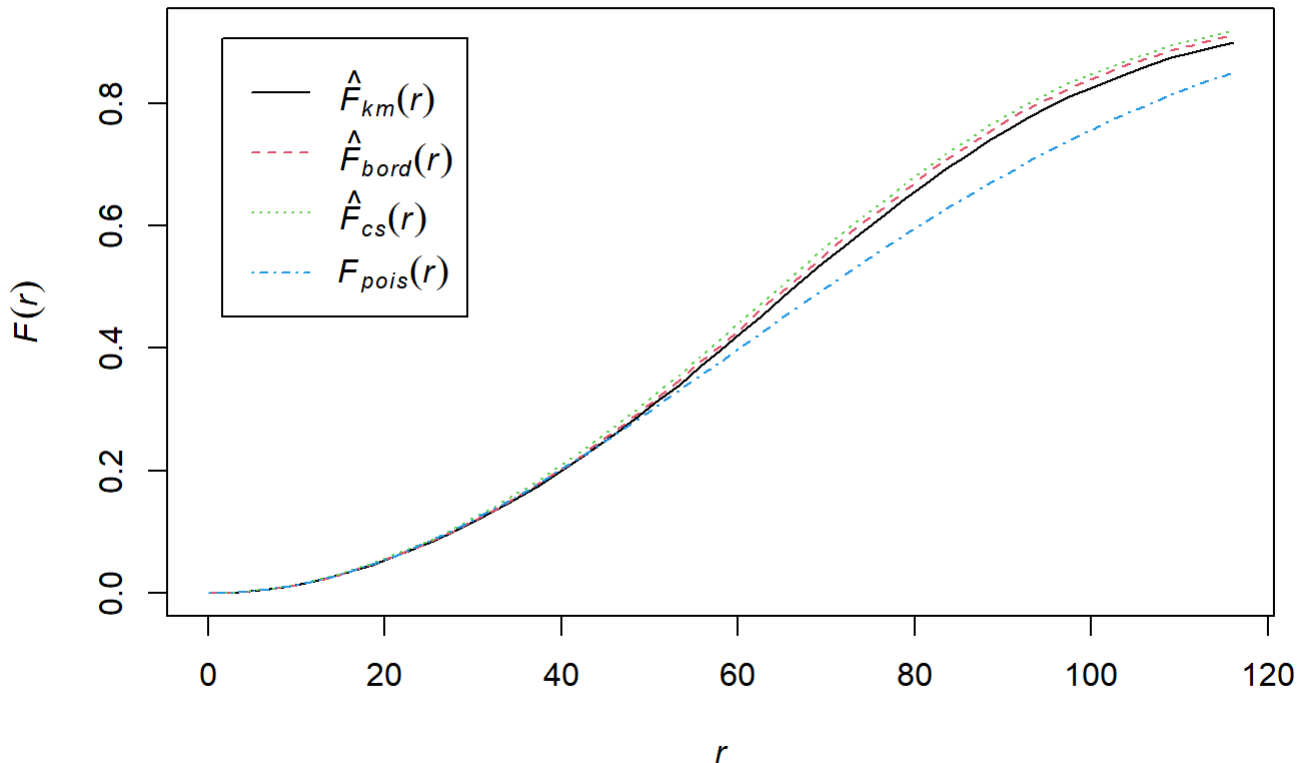
This

test shows that the relationship between total population and bus stops are not random.

Let's test for clustering.

```
ks_balt_bus_stops_f_test <- Ftest(bus_stops_ppp)
plot(ks_bus_401_stops_f_test)
```

### ks\_bus\_401\_stops\_f\_test



means that the relationship between bus stops and total population in Baltimore city is not random and are tied to total population.

##Conclusions of Bus Stop Data for Baltimore City Compared to Census Tract 401 Bus stops for all of Baltimore city appear to be clustered because the observed distances are shorter than the Poisson process. This is in contrast to Census tract 401, where the bus stops were not clustered. Census Tract 401 also had the largest number of bus stops per any census tract in the city. Let's see if this means that the clustering occurs at or around census tract 401.

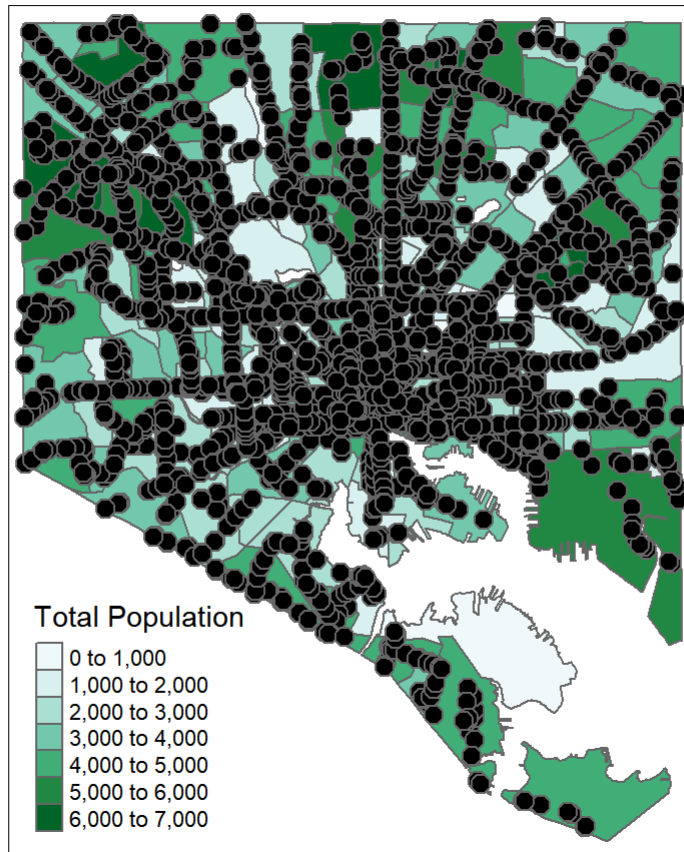
```
balt_pop_break <- classInt::classIntervals(balt_erase2$estimate, n = 5, style = "fixed", fixedBr
eaks = c(0, 1000, 2000, 3000, 4000, 5000, 6000, 7000)) #plotting an interactive map with cust breaks
```

```
## Warning in classInt::classIntervals(balt_erase2$estimate, n = 5, style = 
## "fixed", : variable range greater than fixedBreaks
```

```
tm_shape(balt_erase2) +
  tm_fill(col = "estimate", title = "Total Population", breaks = balt_pop_break$brks, palette =
"BuGn") +
  tm_borders()+tm_shape(bus_stops_balt_correct) +
  tm_symbols(col="black", shape = 21, size = 0.5) +
  tm_layout(main.title = "Bus Stops Baltimore City") #plotting the map
```

```
## Warning: Values have found that are higher than the highest break
```

## Bus Stops Baltimore City



Based on our map, you can see that a significant amount of bus stop points are clustered around census tract 401 in the center of the map.

In conclusion, population change in Baltimore city from 2010 to 2021 has resulted in a gain in population in some census tracts like 401, but mostly a loss in other tracts (see QGIS map in 'bin'). Census tract 401 has the largest amount of bus stops in the city. Bus stops are not clustered within the tract. However, bus stops in Baltimore city are statistically significant to the total population of census tracts and are clustered, and based on our analysis we can conclude that the clustering occurs around census tract 401.