Homework/Inclass March 8 2023

Mike Kozlowski, HDS

2023-03-08

Homework for March 8, 2023

#Question 1

Get the 311 service requests data file from the city of Buffalo

It includes the lat and long so we can plot it over the city of buffalo map

Pick one type of response, and plot the instances over the city of buffalo map with neighborhoods shown

```
infile="C:\\Users\\Mike\\Documents\\DAT511\\3-8 class\\311_Service_Requests.csv"
threedata=read.csv(infile,stringsAsFactors=TRUE)
```

```
require("geojsonio")
```

```
## Loading required package: geojsonio
```

```
## Registered S3 method overwritten by 'geojsonsf':
##   method        from
##   print.geojson geojson
```

```
##
## Attaching package: 'geojsonio'
```

```
## The following object is masked from 'package:base':
##
##     pretty
```

```
require("ggplot2")
```

```
## Loading required package: ggplot2
```

```
require("broom")
```

```
## Loading required package: broom
```

```
nbounds_infile="C:\\Users\\Mike\\Documents\\DAT511\\3-8 class\\Neighborhoods.geojson"
nbounds_df <- geojson_read(nbounds_infile,  what = "sp")
streets_infile="C:\\Users\\Mike\\Documents\\DAT511\\3-8 class\\Highways and Roadways.geojson"
streets_df <- geojson_read(streets_infile,  what = "sp")
nbounds_df_fortified<-tidy(nbounds_df)
```

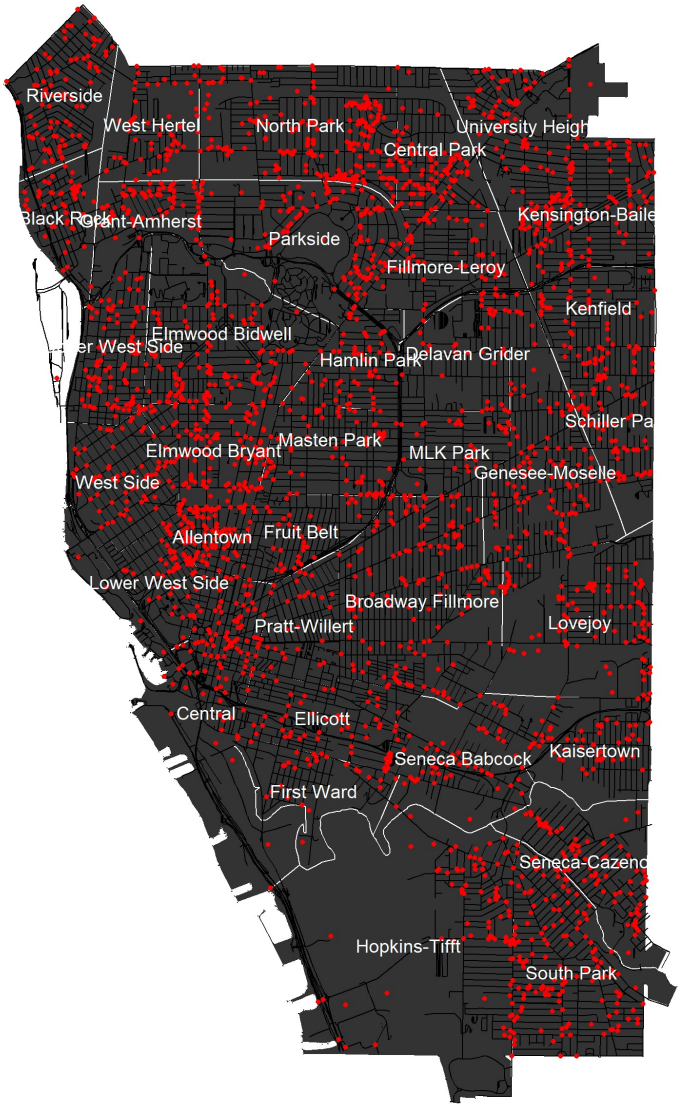
```
## Regions defined for each Polygons
```

```
streets_df_fortified<-tidy(streets_df)
```

```
sneakersdf <- threedata[threedata$Type=="Pot Hole (Req_Serv)",]
infile="C:\\Users\\Mike\\Documents\\DAT511\\3-8 class\\Neighborhood_Metrics.csv"
ndata=read.csv(infile,stringsAsFactors=TRUE)
```

```
ggplot()+geom_polygon(data=nbounds_df_fortified,aes(x = long, y = lat,group=group),color="white")+geom_
line(data=streets_df_fortified,aes(x = long, y = lat,group=group))+theme_void()+coord_map()+geom_point
(data=sneakersdf,aes(x=Longitude,y=Latitude),color="red")+geom_text(data=ndata,aes(x=Longitude,y=Latitu
de,label=Neighborhood),color="White",size=5)
```

```
## Warning: Removed 5417 rows containing missing values (`geom_point()`).
```



#Question 2

Get the tree inventory for the city of buffalo

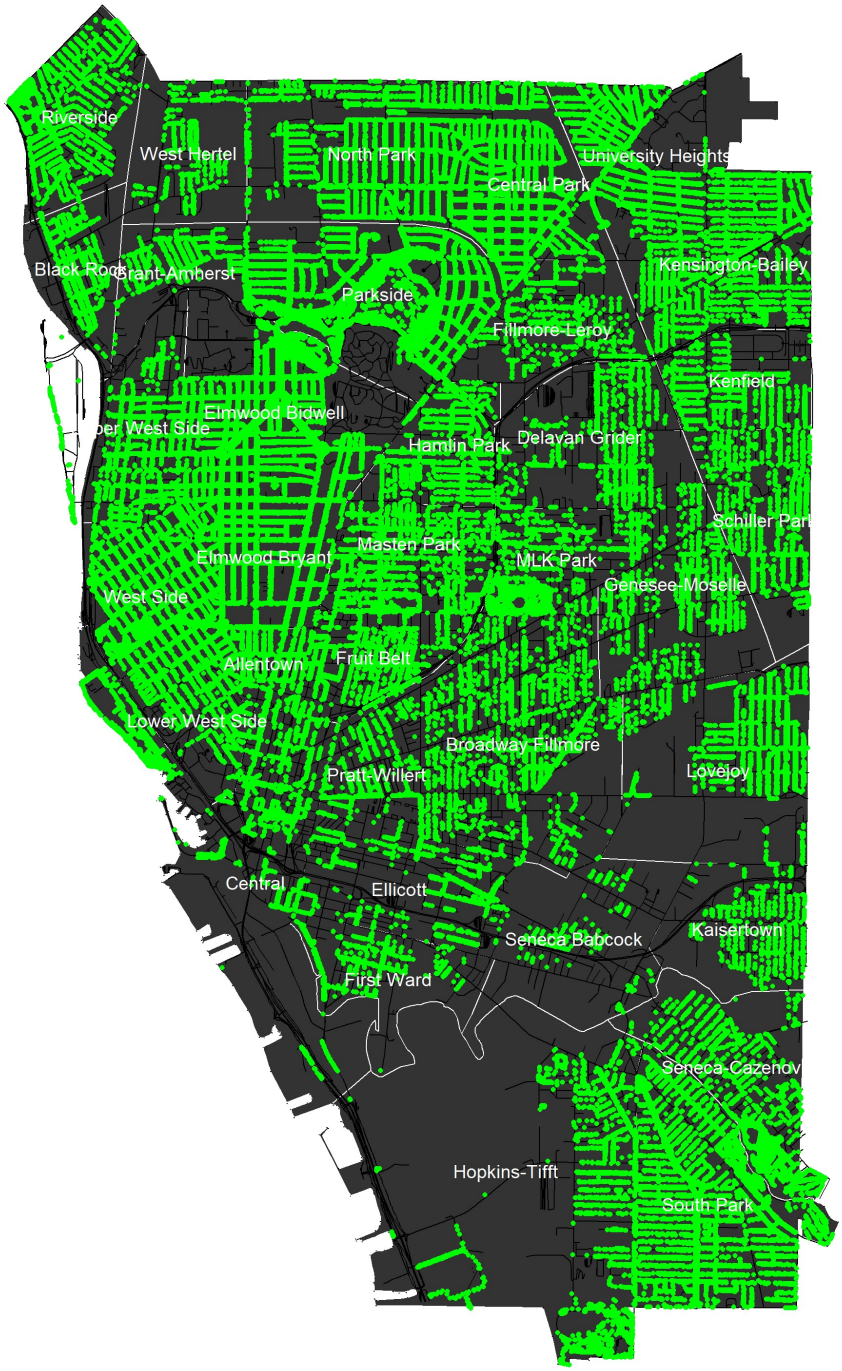
Remove the stumps and Na's and other junk from the tree type

Plot a small green dot for each tree over the map of Buffalo

```
treeinfile="C:\\Users\\Mike\\Documents\\DAT511\\3-8 class\\Tree_Inventory.csv"
treedata=read.csv(treeinfile,stringsAsFactors=TRUE)
```

```
cleantree <- na.omit(treedata)
cleantree <- cleantree[!(cleantree$Botanical.Name=="STUMP" | cleantree$Botanical.Name=="VACANT" | clean
tree$Botanical.Name=="0"),]
```

```
ggplot()+geom_polygon(data=nbounds_df_fortified,aes(x = long, y = lat,group=group),color="white")+geom_
line(data=streets_df_fortified,aes(x = long, y = lat,group=group))+theme_void()+coord_map()+geom_point
(data=cleantree,aes(x=Longitude,y=Latitude),color="green")+geom_text(data=ndata,aes(x=Longitude,y=Latit
ude,label=Neighborhood),color="White",size=5)
```



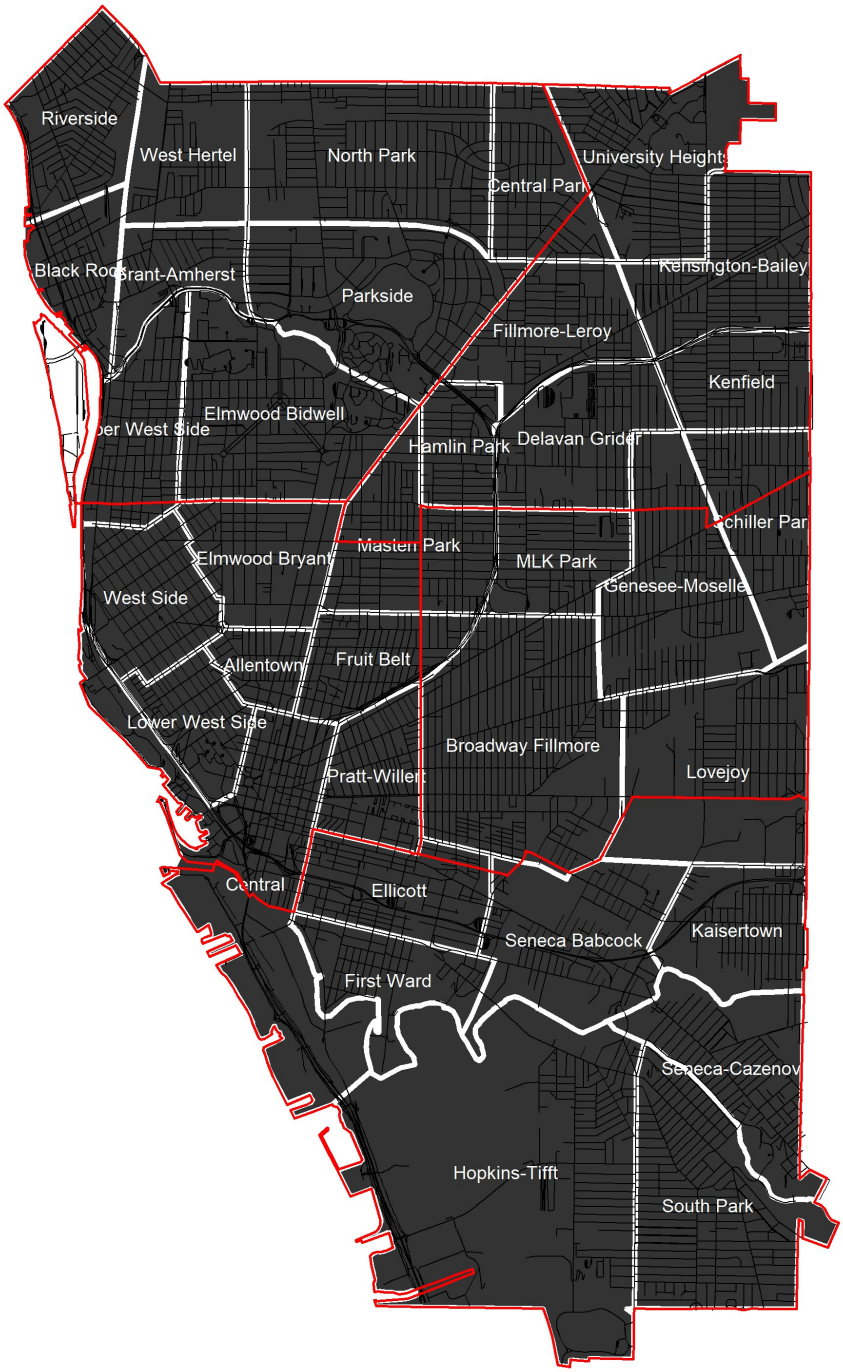
#Question 3

Get the information about Police Districts from open data Buffalo, find the GeoJSON file and plot the Districts over the neighborhoods and streets in the city

```
police_infile="C:\\Users\\Mike\\Documents\\DAT511\\3-8 class\\Police Districts.geojson"
police_df <- geojson_read(police_infile,  what = "sp")
police_df_fortified<-tidy(police_df)
```

```
## Regions defined for each Polygons
```

```
ggplot()+geom_polygon(data=nbounds_df_fortified,aes(x = long, y = lat,group=group),linewidth=2,color="white")+geom_line(data=streets_df_fortified,aes(x = long, y = lat,group=group))+theme_void()+coord_map()+geom_text(data=ndata,aes(x=Longitude,y=Latitude,label=Neighborhood),color="White",size=5)+geom_polygon(data=police_df_fortified,aes(x = long, y = lat,group=group),fill=NA,color="red",linewidth=1)
```



Question 4

Look through the various pre-built themes available for ggplot.

Pick a favorite and then customize it. Change at least 5 parameters to suit your preferences. Use this customized theme for

the questions below.

```
require("ggthemes")
```

```
## Loading required package: ggthemes
```

```
theme_my_gdocs<-function (base_size = 12, base_family = "sans")
{
    ltgray <- "#cccccc"                    #these look like base 16 or hex codes for colors in rgb not
ation, for different grey                                                                  # tones
    dkgray <- "#757575"
    dkgray2 <- "#666666"
    theme_foundation(base_size = base_size, base_family = base_family) +
        theme(rect = element_rect(colour = "pink", fill = "black"),
            line = element_line(colour = "pink"), text = element_text(colour = dkgray2),
            plot.title = element_text(face = "plain", size = rel(20/12),
                hjust = 0.5, colour = dkgray), plot.subtitle = element_text(hjust = 0,                #
set hjust=0.5, for centering
                size = rel(1), face = "plain", colour = dkgray),
            plot.caption = element_text(hjust = 0, size = rel(1),
                face = "plain", colour = dkgray), panel.background = element_rect(fill = NA,
                colour = NA), panel.border = element_rect(fill = NA,
                colour = NA), strip.text = element_text(hjust = 0,
                size = rel(1), colour = dkgray2, face = "plain"),
            strip.background = element_rect(colour = NA, fill = NA),
            axis.title = element_text(face = "plain", colour = "pink",                              # se
t axis title to black
                size = rel(1)), axis.text = element_text(face = "plain",
                colour = dkgray, size = rel(1)), axis.line = element_line(colour = "pink"),       # rem
oved the line below,
             axis.ticks = element_blank(),                                                          #axi
s.line.y = element_blank(),
            panel.grid.major = element_line(colour = dkgray),                                      # alt
ered major and minor grid
            panel.grid.minor = element_line(color=dkgray2,linetype="dotted"), legend.background = eleme
nt_rect(colour = NA),
            legend.text = element_text(size = rel(1), colour = dkgray),
            legend.title = element_text(size = rel(1), colour = dkgray2,
                face = "plain"), legend.key = element_rect(colour = NA),
            legend.position = "right", legend.direction = "vertical")
}
theme_set(theme_my_gdocs())
```

#Question 5

The census data shows more higher levels of resolution than the neighborhood level aggregations, since census blocks are much smaller

Get the data set “ACS 2017 Data Profile 5 Year Estimates, Erie County-Census Tract

a.) Use ggplot to plot a histogram and a boxplot of HC01_VC03, total households per census tract do this for total population as well HC01_VC48.

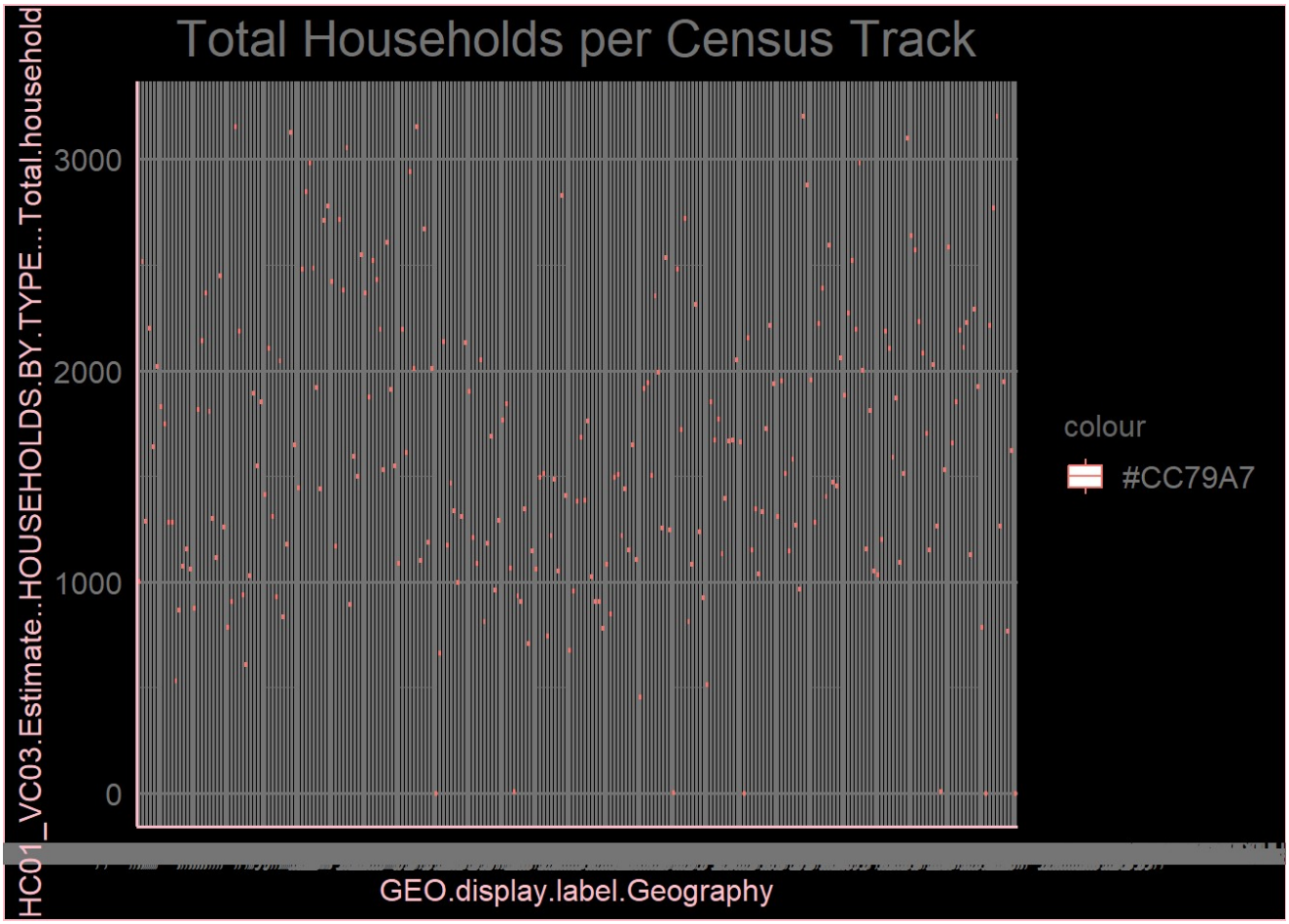
b.) Pick 2 variables that look interesting, create a biplot of the two

c.) Add a third variable to b and do a bubble plot

d.) Select 8 to 10 interesting looking variables and show a plot of the correlation structure

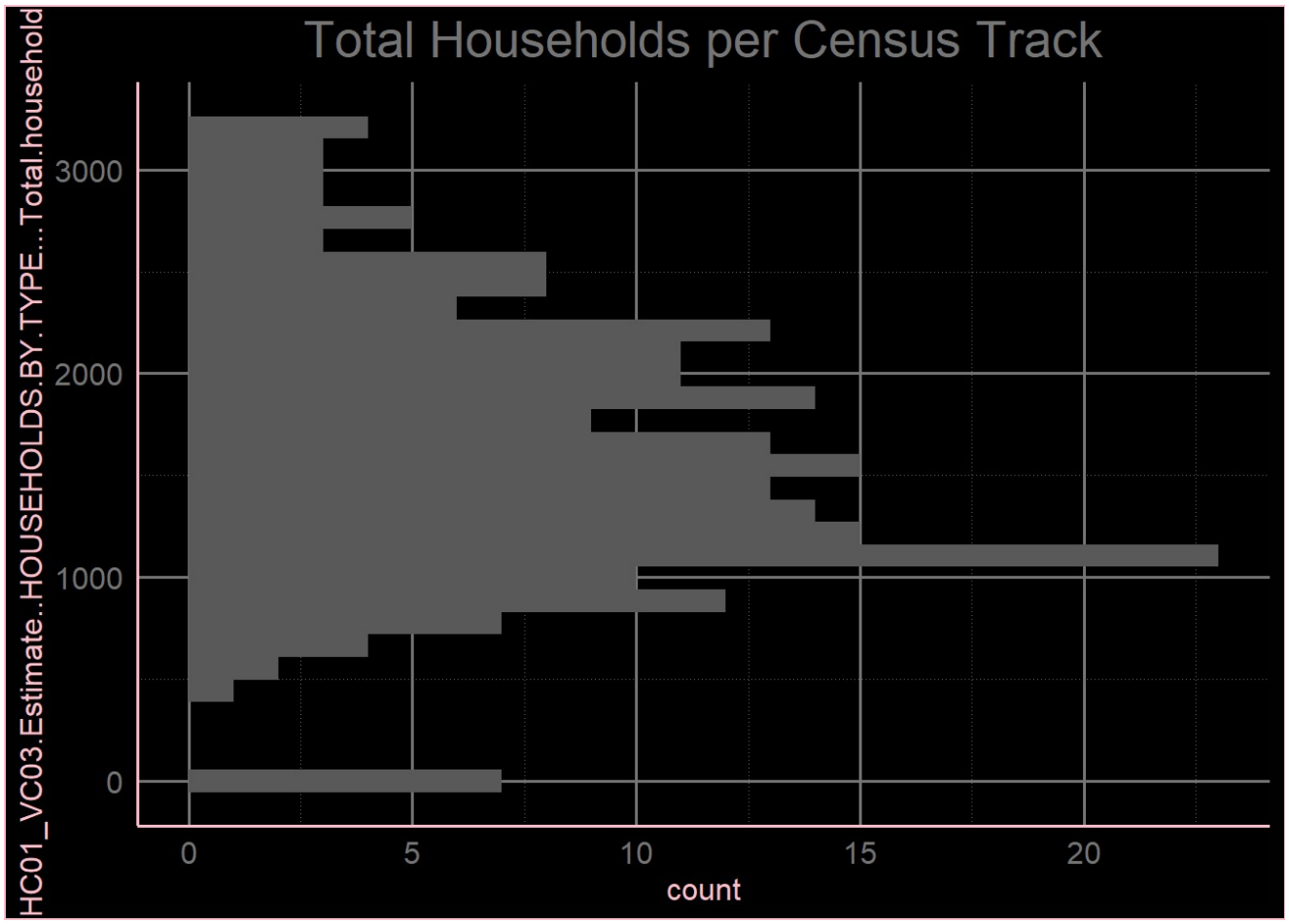
```
census_infile="C:\\Users\\Mike\\Documents\\DAT511\\3-8 class\\ACS_2017_Data_Profile_5_Year_Estimates__Erie_County_-_Census_Tract.csv"
censusdata=read.csv(census_infile,stringsAsFactors=TRUE)
```

```
require(ggplot2)
ggplot(censusdata,aes(x=GEO.display.label.Geography,y=HC01_VC03.Estimate..HOUSEHOLDS.BY.TYPE...Total.households,col="#CC79A7"))+geom_boxplot()+ggtitle("Total Households per Census Track")
```

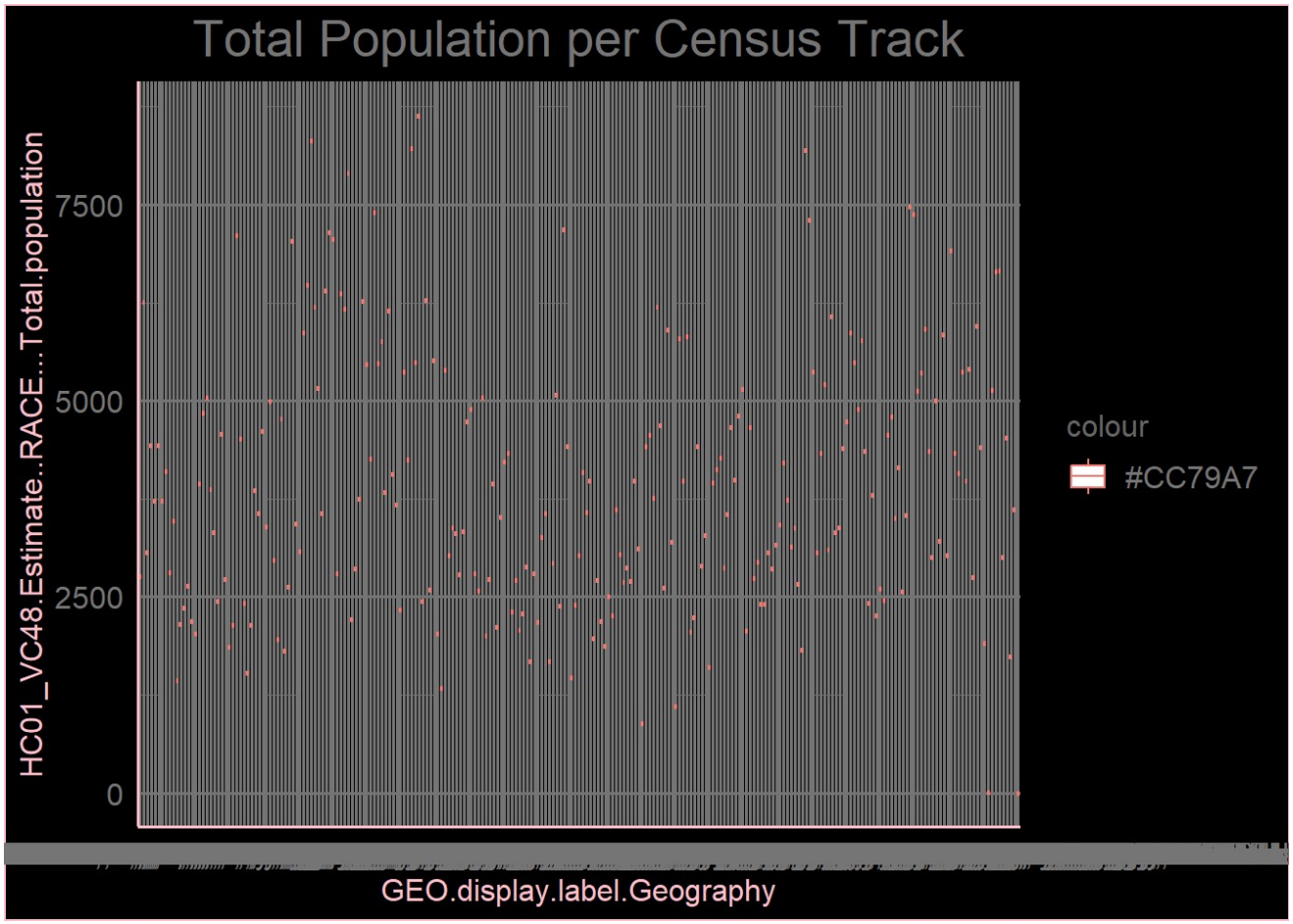



```
ggplot(censusdata,aes(y=HC01_VC03.Estimate..HOUSEHOLDS.BY.TYPE...Total.households))+geom_histogram()+ggtitle("Total Households per Census Track")
```

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




```
ggplot(censusdata,aes(x=GEO.display.label.Geography,y=HC01_VC48.Estimate..RACE...Total.population,col="
#CC79A7"))+geom_boxplot()+ggtitle("Total Population per Census Track")
```



```
ggplot(censusdata,aes(y=HC01_VC48.Estimate..RACE...Total.population))+geom_histogram()+ggtitle("Total P
opulation per Census Track")
```

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