ARCADE

Arcade is an expression language created by ESRI. It can be used only in ESRI's products. It looks similar to JavaScript, but has small differences. This can cause some confusion for people who are using JavaScript often.

Arcade can be used in ArcGIS Pro and ArcGIS Online for

- Symbology
- Labeling
- Calculate field
- pop-ups

In order to learn Arcade more, as a test case, the walkability of Minneapolis by Census Tract is studied.

Minneapolis Walkability Analysis by Census Tracts

A Case Study for Making Beautiful Web Maps with ArcGIS Arcade

OBJECTIVES

The aim of this study is to learn how to implement Arcade, which is an expression language of ArcGIS, to ArcGIS Online and ArcGIS Pro, and discover its limitations. As a test case, the walkability of Minneapolis by Census Tract is studied. The main aims are to determine Census Tract areas in Minneapolis that are suitable for leisure walking by calculating a walkability score, to discover Arcade functions to make the analysis faster and easier, and to perform statistical analysis of the walkability score against median income per census tract to see if they are correlated.

METHODS

A quantitative measure is defined to assign a "Walkability Score" to each feature using data from 7 different datasets. In order to calculate scores without coding, one has to run many tools such as buffer, spatial join, select features by location, merge, etc., create new fields, and calculate the fields. Arcade enables users to use a single code, and calculate everything in one step.

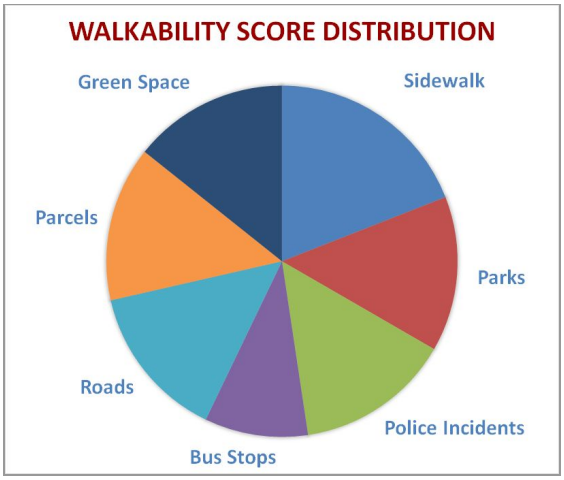
The calculation was performed in ArcGIS Pro to the Census Tract feature class. Within the feature class, I created a field "Score", and when calculating the field, Arcade was chosen as the expression language. With the FeatureSetByName function, I calculated scores from different datasets. My original intention was to use ArcGIS Online only, however, it was very slow when calculating anything from more than 2 different datasets while the code was using FeatureSetByName more than 10 times.

I shared the layer in ArcGIS Online. I created a web map, and added all the datasets that were mentioned in the "Dataset" section. Walkability by Census Tracts layer is symbolized to show scores. Instead of using Arcade to calculate the final score, I used it when calculating scores for each category. So, the layer draws total score information from the layer's "Score" field. But each score category are calculated dynamically within the map, and it can be seen in the pop-up windows. (The web map's link is given on the last page.)

Dataset for Score Calculation

The following data is used to calculate the walkability measure.

- **Police Incidents 2017 (Score 0 to 15)**
  Number of crime points by Census tract, normalized by Census Tract area
  Data source: Minneapolis Open Data
- **Road (Score 0 to 15)**
  Length of State, Interstate and US Highways, County Highways, and Local Streets; normalized by total road length. Roads Weighted as:
  State, Interstate and US Highways: 0
  County Highways: 0.8
  Local Streets: 1
  Data source: MN Geospatial Commons/Minnesota Department of Transportation (MnDOT)
- **Green Space (0 to 15)**
  Green space classification from 2015 NAIP Imagery. Green space area normalized by Census Tract's land area.
  Data Source: National Agriculture Imagery Program (NAIP) - 2015 (Data were taken from FNRM 5262—Remote Sensing & Geospatial Analysis class project by McGraw, Birol, Sjostrom)
- **Sidewalk (Score 0 to 20)**
  Sidewalk length divided by 2 and normalized by total street length within the census tracts.
  Data Source: Minneapolis Open Data (PW Planimetric Lines)
- **Parcel (Score 0 to 15)**
  Area of residential, commercial, apartment, and townhouse parcels divided by the total parcel area including the industrial parcels as well, per census tracts. (Other parcel types such as golf courses are not included.)
  Data Source: Hennepin County Open Data (County Parcels)
- **Regional Parks (Score 0 to 15)**
  Regional Park areas within Census Tracts, normalized by total Census Tract areas.
  Data Source: MN Geospatial Commons/Metropolitan Council
- **Bus Stops (Score 0 to 10)**
  Number of Bus stops by Census tract, normalized with Census Tract area.
  Data Source: MN Geospatial Commons/Metropolitan Council - Metro Transit



Additional Data:

- **Census Tract Shapefile**
  Clipped by Minneapolis neighborhoods
  Data source: Census Bureau
- **Median Household Income per Census Tracts**
  American Community Survey (ACS) 2017 - 5 year estimates
- **School Locations**
  Minneapolis Geospatial Commons/Minnesota Department of Education
- **Minneapolis Neighborhoods, 2019 Police Incidents (Service Link), Minneapolis Police Precincts (Service Link)**
  Minneapolis Open Data

Arcade - Walkability Score Calculation

Arcade is used when calculating the Walkability Score in ArcGIS Pro "Calculate Field" tool. We create a new field in Census Tracts feature class as "score", and we calculate the field with Arcade expression. The image below is an example of how the Parcel Score is calculated. After calculating each score per Census Tracts, the sum of the scores is assigned to the Score field.

Score =

```
var comm = area(intersects(filter(FeatureSetByName($datastore, "MinneapolisParcels_Tracts"),
"PR_TYP_NM1 LIKE '%COMMER%' "), $feature), 'square-meters' )
var res = area(intersects(filter(FeatureSetByName($datastore, "MinneapolisParcels_Tracts"),
"PR_TYP_NM1 LIKE '%RESID%' "), $feature))
var ind = area(intersects(filter(FeatureSetByName($datastore, "MinneapolisParcels_Tracts"),
"PR_TYP_NM1 LIKE '%INDUS%' "), $feature))
var apt = area(intersects(filter(FeatureSetByName($datastore, "MinneapolisParcels_Tracts"),
"PR_TYP_NM1 LIKE '%APT%'OR PR_TYP_NM1 LIKE '%TOWN%' "), $feature))
var totalParc = (res+ind+comm+apt)
var parcelScore = ((res+apt+comm)/(totalParc))*15
```

In the code below, the total area of commercial parcels which intersect with tracts are calculated and assigned to a new variable named “comm”.

```
var comm = area(
              intersects(
                     filter(
                            FeatureSetByName($datastore, "MinneapolisParcels_Tracts"),
                            "PR_TYP_NM1 LIKE '%COMMER%' "), $feature), 'square-meters' )
```

Explaining functions:

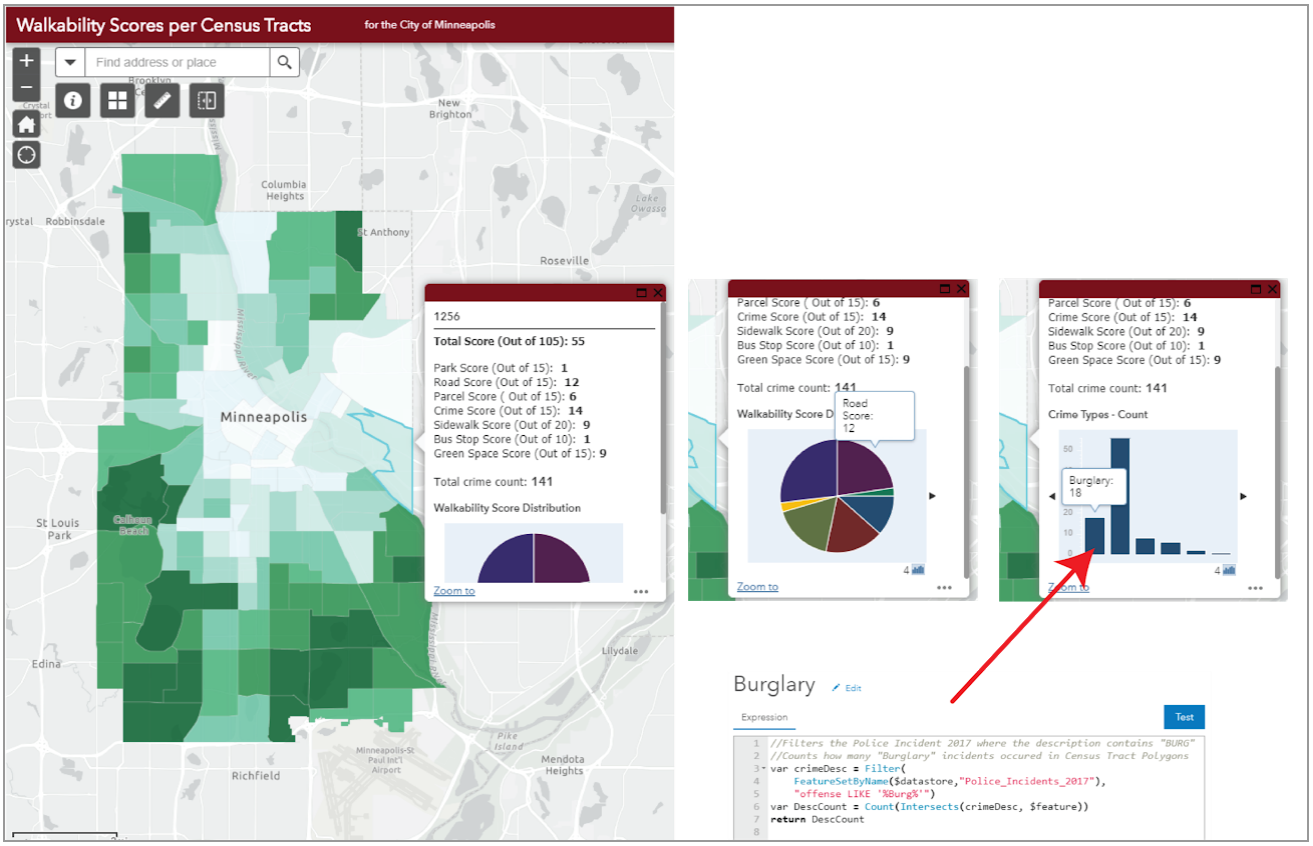
- The FeatureSetByName function gathers information from another layer as a FeatureSet.
- The Filter function filters the FeatureSet for parcel types which have “COMMER%” wildcard by using SQL92 expressions.
- The Intersects function gets the features from the FeatureSet which are intersecting with the features.
- The Area function calculates the area of intersected and filtered features in square meters.

The very same codes are also used in Web Map pop-up windows to calculate Parcel Scores of individual Census Tracts On-The-Fly.

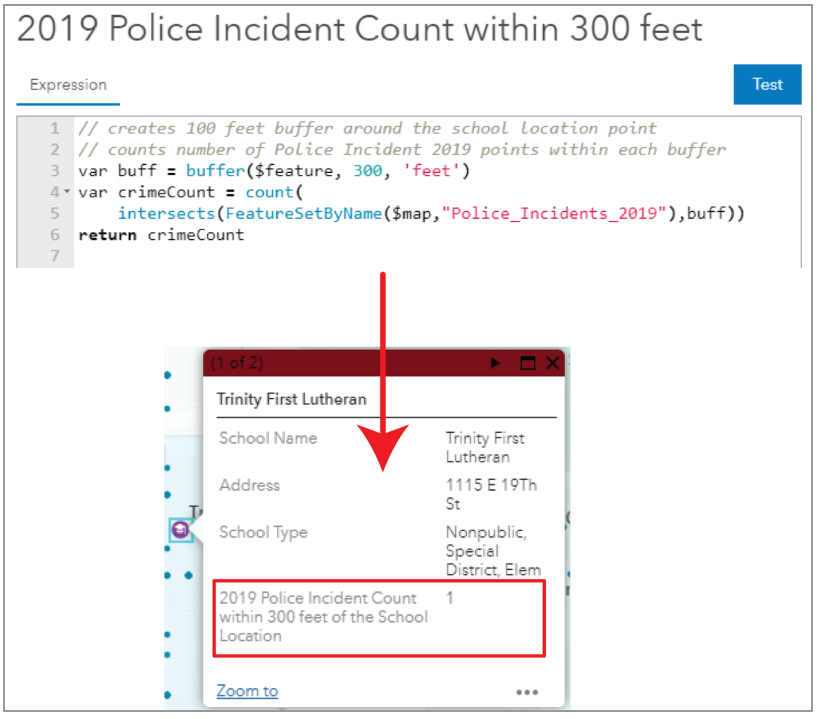
Using Arcade in Web Maps

Web maps are more useful when displaying multiple analyses. It allows users to interact with the map. With Arcade, users can create dynamic popups that derive information from different datasets without making any necessary changes to the data. In this example, various charts that display data from other layers can be seen.

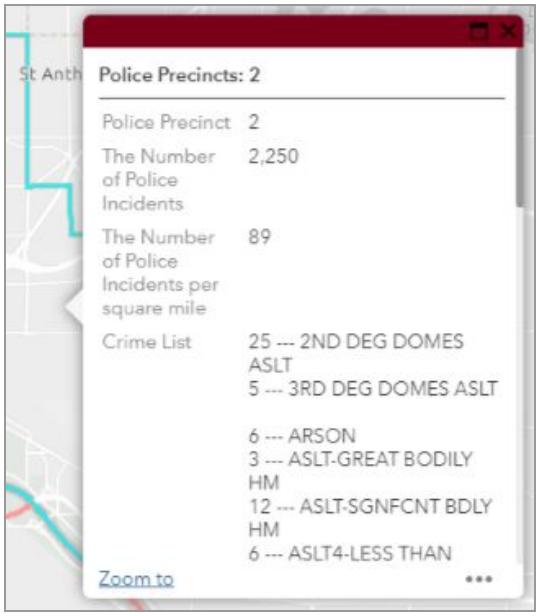
With Arcade, multiple variables/attributes are created within the “Walkability Scores per Census Tract” layer. The count information is derived from the 2017 Police incident data which is on the map. Different types of charts can also be plotted within popups.



Arcade helps users to create dynamic pop-ups. In the example on the right, the School location data has an expression in its pop-up to calculate how many Police Incident locations (in 2019) are in 300 feet around the point. The 2019 Police Incident data is added to this map as a feature service, and when Minneapolis updates the data, the service layer will automatically update itself as well.



In the example below, both Police Precinct and Police Incidents 2019 are service layers. The number of incidents that are within the polygons of each class can also be shown as a list in the Police Precinct pop-ups. The data is shown on-the-fly, and it calculates the results every time when the pop-up is opened. When the source data is updated, the information will also be updated.



Code Block:

```
var bla = {};
var count0 = 0;
var crimePoint = "";
var crimeList = "";
var crimeCount = Intersects(FeatureSetByName($map,"Police_Incidents_2019"),$feature)
for (var i in crimeCount) {
    if(haskey(bla, i.Description)==false){
         bla[i.Description]=1;
    }
    else{
        bla[i.Description]+=1
    }
}
for (var j in bla) {
    crimePoint = bla[j] + " --- " + j + TextFormatting.NewLine
    crimeList += crimePoint
}
return crimeList
```

RESULTS and DISCUSSIONS

About Arcade:

- Arcade's geometry functions are considerably slow when using feature sets in ArcGIS Online pop-ups. That's why I ended up using ArcGIS Pro to calculate the final score. It is more useful when running codes with geometry functions of feature sets. If the total score was being calculated within
- Feature sets are useful when calculating fields, but each layer has to be in the same data source which makes things a bit complicated when working in different environments.
- Arcade is the most useful when using service layers that are being updated regularly, such as the Police Incident 2019 and Police Precinct example.
- Arcade decreases the number of steps for spatial analysis.
- Arcade enriches data visualization with custom pop-ups, symbology, and labeling.
- In order to perform these analyses with Arcade, and use functions such as feature set and geometry, users need the version of ArcGIS Server 10.7 or ArcGIS Pro 2.4.

About analysis:

- Minneapolis walkability scores are correlated with Median Household Income with high R^2 value. (Shown in scatter plot.)
- The datasets come from different years: Police incident data (2017), Greenspace classification (2015), sidewalk (various years, including 1991), Average income (ACS 2017 estimates). Using consistent data from the same year would improve the reliability of the results.
- The sidewalk data is not up to date, hence it is not completely reliable. When better data is publicly available, the same analysis can be re-run to gain more reliable results.
- Walking time (the length of available walking paths between two points) was not considered in this study.
- Since there is not a unique definition of "the walkability score", it is subjective. In this study we used only a few parameters. In an ideal study, sidewalk pavement quality, intersection density, various commercial type (such as restaurants, stores , etc.) land uses, and pollution index would have been also considered at the "sidewalk" level, rather than large areas like Census Tracts.

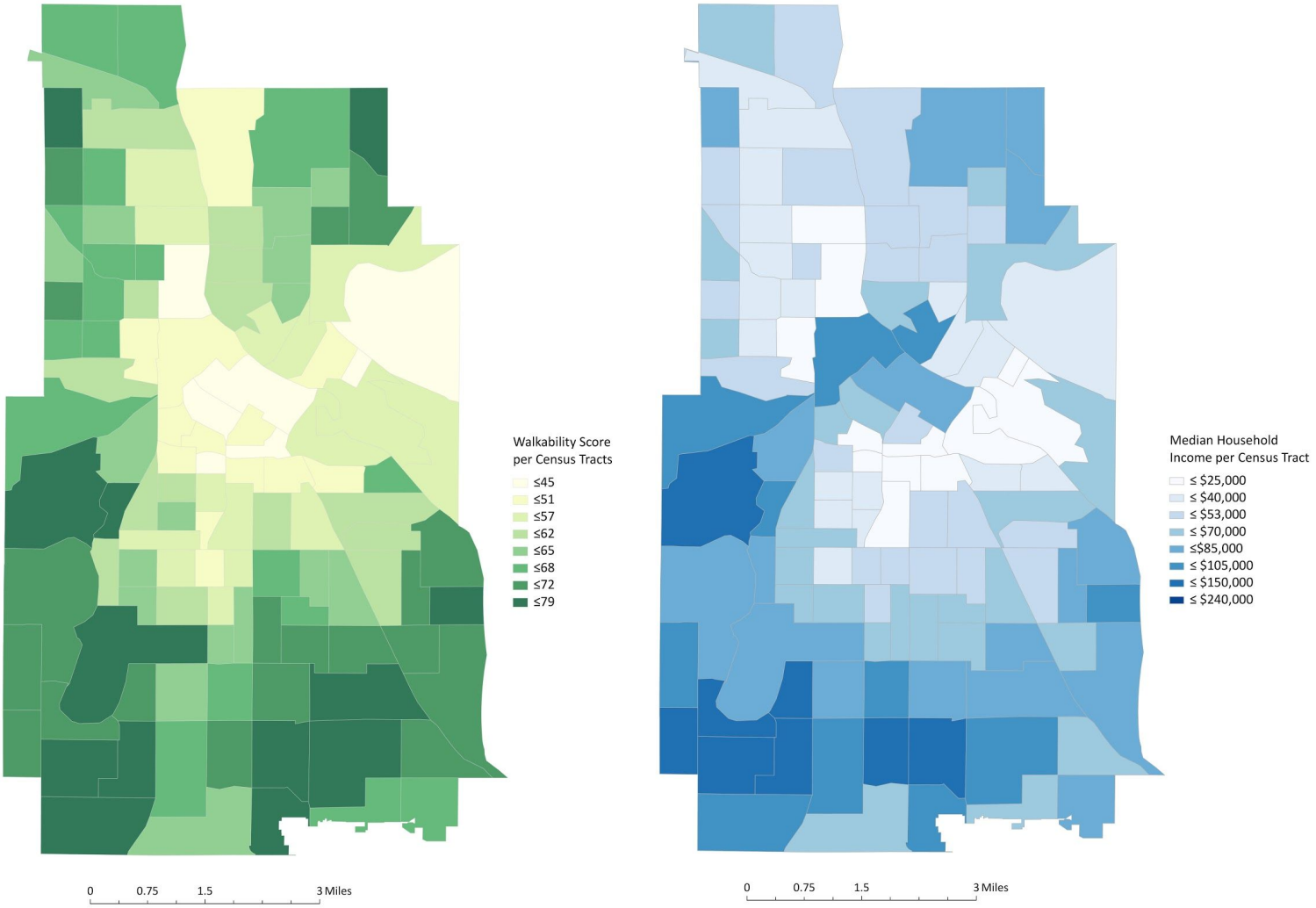


Figure: Walkability score per Census Tracts (Left), Median Household Income per Census Tracts (Right)

Link to the Web App:
https://umn.maps.arcgis.com/apps/webappviewer/index.html?id=87c89e205c93465795dc730657fd5e5e
Link to the Web Map:
https://umn.maps.arcgis.com/home/webmap/viewer.html?webmap=aa3ba9df165445d9bf8ded525f917772

Resources:
Youtube videos by Esri Events:
ArcGIS Online: Web Mapping with Arcade Expressions
Pump Up Your Pop-Ups with Arcade Expressions

Esri Blogs:
Pump up Your Pop-ups With Arcade FeatureSets and the Living Atlas
Discover patterns in your data with ArcGIS Arcade
Use Arcade Expressions to Map Your Ideas
What's New in Arcade 1.8

Story Map by me:
Informative Web Maps

The code when calculating score in ArcGIS Pro.

```
var comm = area(intersects(filter(FeatureSetByName($datastore, "MinneapolisParcels_Tracts"),
"PR_TYP_NM1 LIKE '%COMMER%' "), $feature), 'square-meters' )
var res = area(intersects(filter(FeatureSetByName($datastore, "MinneapolisParcels_Tracts"),
"PR_TYP_NM1 LIKE '%RESID%' "), $feature))
var ind = area(intersects(filter(FeatureSetByName($datastore, "MinneapolisParcels_Tracts"),
"PR_TYP_NM1 LIKE '%INDUS%' "), $feature))
var apt = area(intersects(filter(FeatureSetByName($datastore, "MinneapolisParcels_Tracts"),
"PR_TYP_NM1 LIKE '%APT%'OR PR_TYP_NM1 LIKE '%TOWN%' "), $feature))
var totalParc = (res+ind+comm+apt)
var a = ((res+apt+comm)/(totalParc))*15
var greenSpaceRatio = $feature.GreenSpace / $feature.AreaWOWater
var b = (($feature.GreenSpaceRatio)/0.72677286)*15
var sidewalk =
Length(intersects(FeatureSetByName($datastore,"Sidewalk"),$feature),'meters')/2
var street = Length(intersects(FeatureSetByName($datastore, "Streets"),$feature),'meters')
var ratio = sidewalk/street
var c = (ratio/2.81748439)*20
var hw = length(intersects(filter(FeatureSetByName($datastore, "Streets"), "ROUTE_SYSTEM =
'01' Or ROUTE_SYSTEM = '02' Or ROUTE_SYSTEM = '03' "), $feature), 'meters' )
var chw = length(intersects(filter(FeatureSetByName($datastore, "Streets"), "ROUTE_SYSTEM =
'04'  "), $feature), 'meters' )
var st = length(intersects(filter(FeatureSetByName($datastore, "Streets"), "ROUTE_SYSTEM NOT
IN ('01', '02', '03', '04')  "), $feature), 'meters' )
var totalRoad = (hw+chw+st)
var d = (((chw*0.2)+st)/totalRoad)*15
var pa = Area(intersects(FeatureSetByName($datastore, "ParksRegional"), $feature),
'square-meters' )
var e = (pa)/($feature.Shape_Area)*10
var crimeCount = count(intersects(FeatureSetByName($datastore,
"Police_Incidents_2017"),$feature))
var crimeDensity = crimeCount/$feature.AreaWOWater
var f = 15 - (((crimeDensity )/0.00109605)*15)
var busStops = count(intersects(FeatureSetByName($datastore, "TransitStops"),$feature))
var busStopDensity = busStops /$feature.AreaWOWater
var g = ((busStopDensity )/0.000181176209241154)*10
var score = a+b+c+d+e+f+g
return score
```