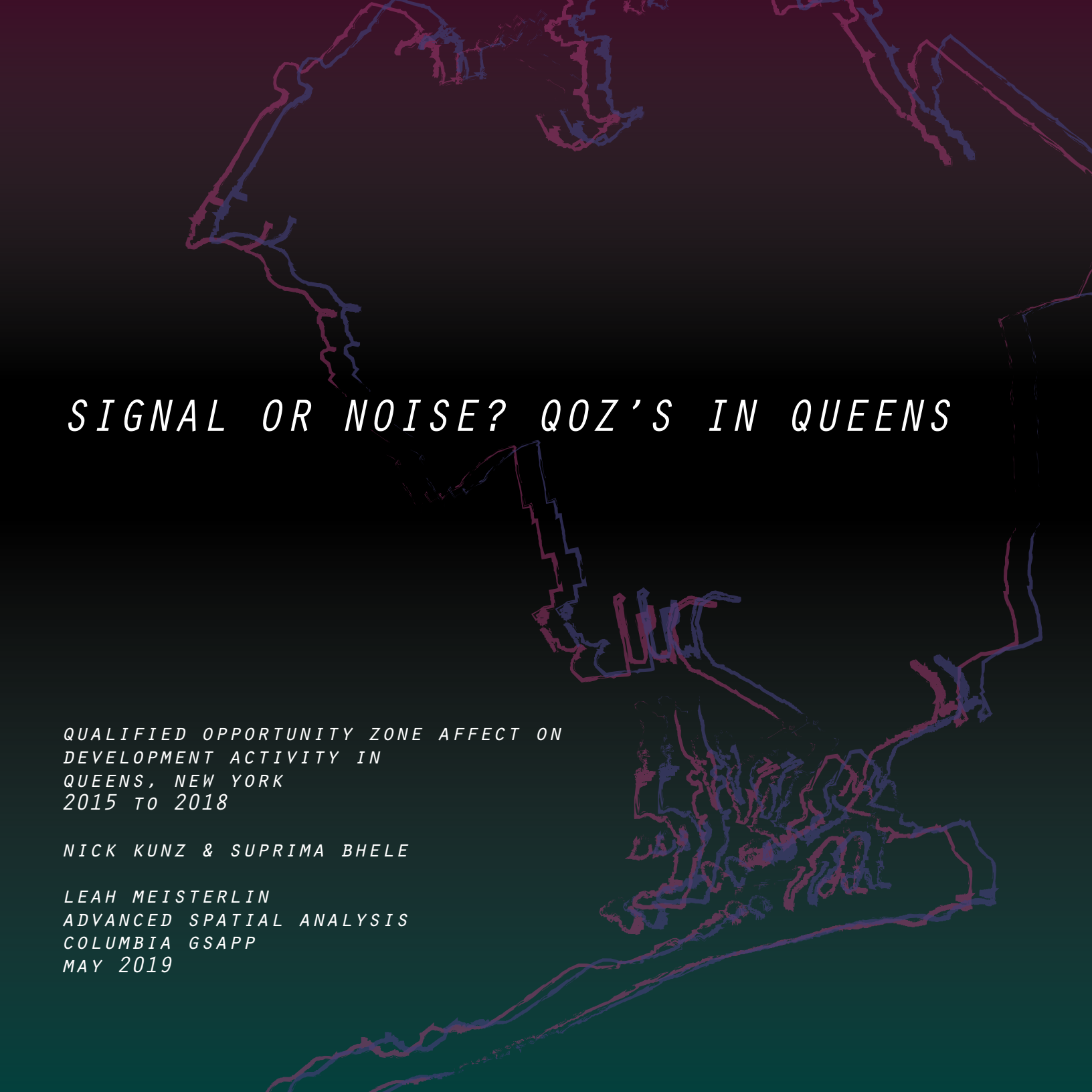


SIGNAL OR NOISE? QOZ'S IN QUEENS

QUALIFIED OPPORTUNITY ZONE AFFECT ON DEVELOPMENT ACTIVITY IN QUEENS, NEW YORK
2015 TO 2018

NICK KUNZ & SUPRIMA BHELE

LEAH MEISTERLIN
ADVANCED SPATIAL ANALYSIS
COLUMBIA GSAPP
MAY 2019

CONTENTS

1. INTRODUCTION

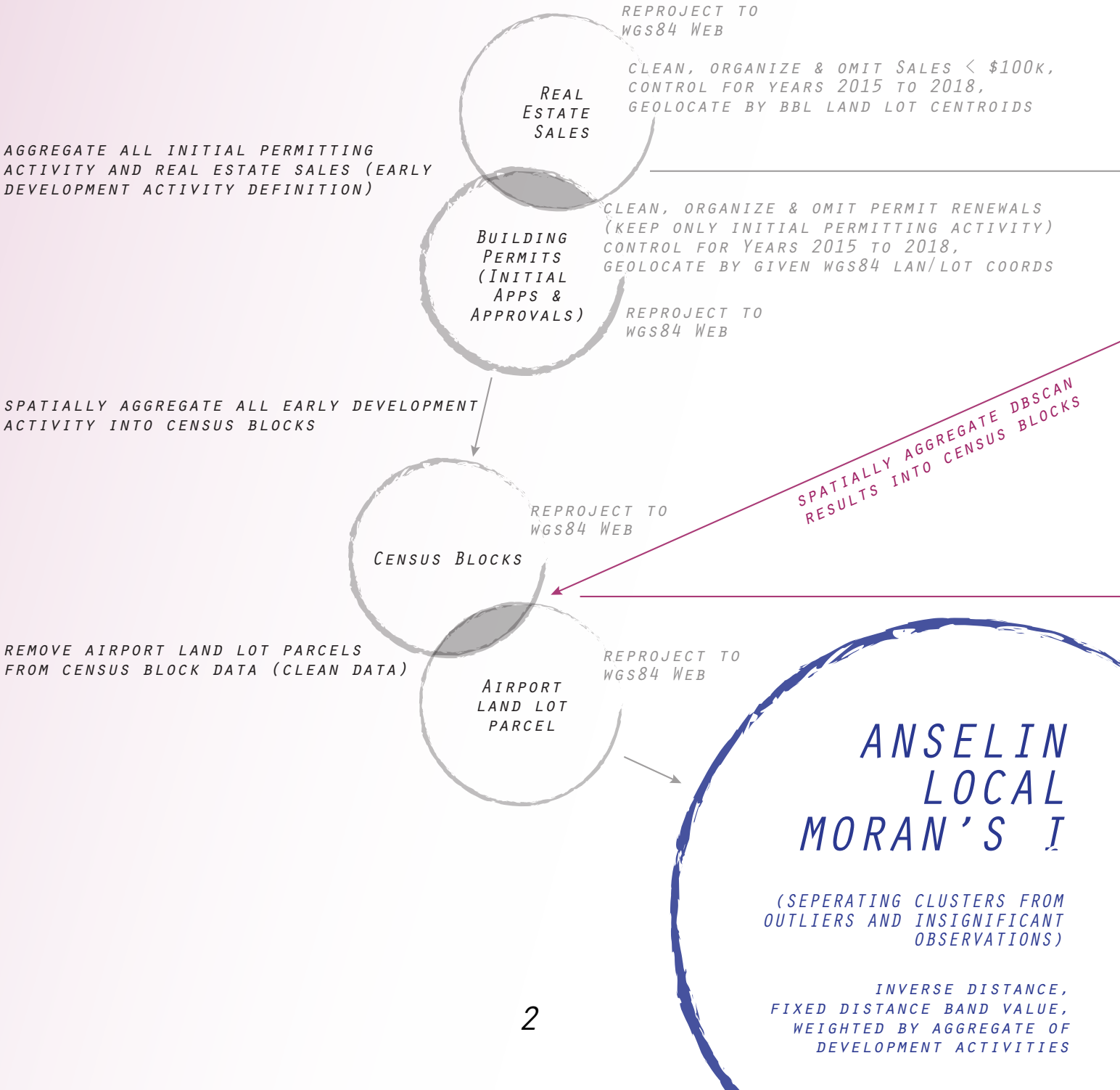
As a part of the Tax Cuts and Jobs Act of 2017, the Qualified Opportunity Zone (QOZ) program was introduced to encourage long-term private investment in low-income urban and rural communities nationwide.

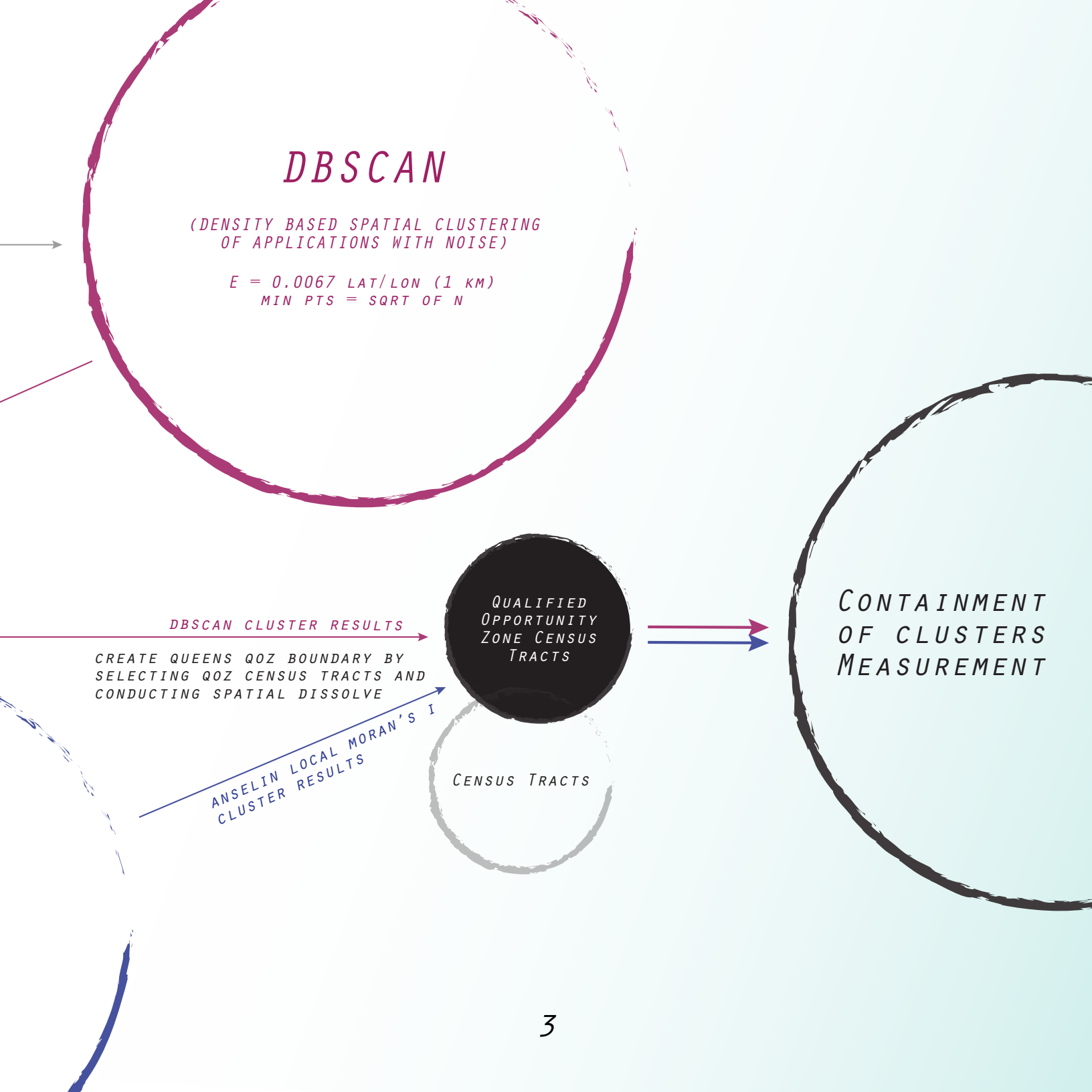
A QOZ lies within a census-tract boundary where the poverty rate is at least 20 percent and the median family income is no greater than 80 percent of the area median. Developments located within QOZ's qualify for deferred tax treatment on capital gains tax, where the total benefit is maximized over a 10-year time horizon. The financing development within QOZ's relies upon Opportunity Funds, which primarily are designed to provide investors with the opportunity for investing into projects located within QOZ's. The Opportunity Funds are available to a wide range of investors to pool their resources in QOZ's with the stated goal of increasing the scale of private investment in underserved communities.

Considering that the QOZ program has a limited time horizon, investors realize a greater benefit by investing quickly into developments located within QOZ's. Therefore, we hypothesized that the areas within QOZ's might likely exhibit an increase in development activity in 2018 (the year following their introduction). We also suspected that the introduction of QOZ's might lead to spillover effects around the census tract boundaries, resulting in clusters of developments around them.

Was there any evidence of spatial concentration and containment of early development activity within Qualified Opportunity Zones the year after their introduction within Queens, New York? To what extent did these concentrations affect the area around them with regard to spatial clustering?

2. METHODOLOGY




DBSCAN
(DENSITY BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE)
E = 0.0067 LAT/LON (1 KM)
MIN PTS = SQRT OF N
DBSCAN CLUSTER RESULTS
CREATE QUEENS QOZ BOUNDARY BY SELECTING QOZ CENSUS TRACTS AND CONDUCTING SPATIAL DISSOLVE
ANSELIN LOCAL MORAN'S I CLUSTER RESULTS
Qualified Opportunity Zone Census Tracts
Census Tracts
Containment of Clusters Measurement

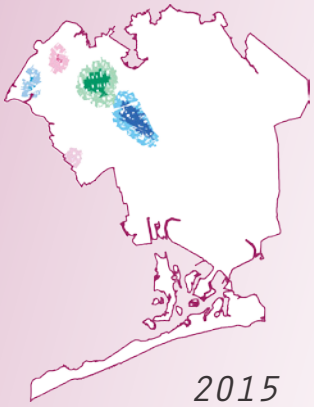
DBSCAN

(DENSITY BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE)

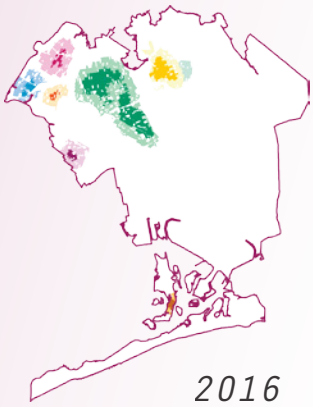
E = 0.0067 LAT/LON (1 KM)
MIN PTS = SQRT OF N

DBSCAN CORE CLUSTERS & MINOR OUTLIERS

QOZ'S BEGIN

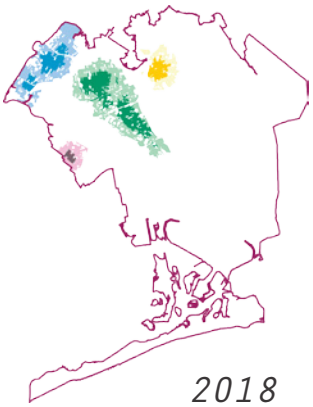


2015



2016

2017



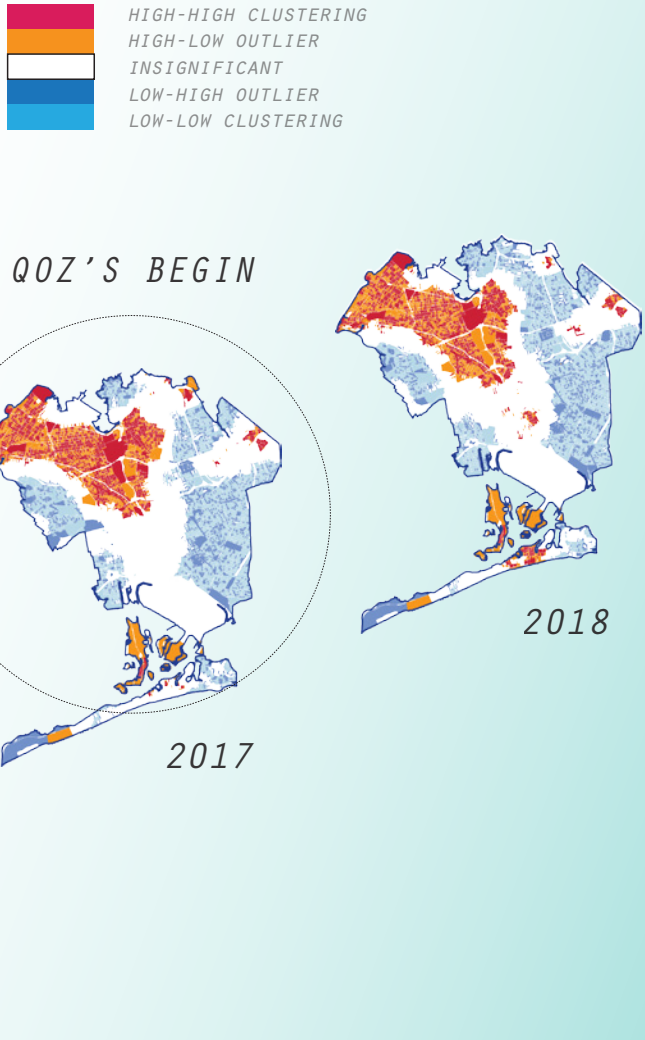
2018

DBSCAN GENERATED CLUSTERS BY TUNING THE DISTANCE BAND BY KNN DENSITY AND A MINIMUM CLUSTER THRESHOLD SET AT THE SQUARE ROOT OF THE TOTAL NUMBER OF SALES AND PERMITS ON AN ANNUAL BASIS. THIS METHOD DISREGARDED SPATIAL OUTLIERS. ACROSS ALL FOUR YEARS OF THE ANALYSES, DBSCAN SUGGESTED THAT THERE WERE EXISTING CLUSTERS OF DEVELOPMENT ACTIVITY AROUND QOZ'S PRIOR TO 2017. THE DARKER AREAS REFER TO THE CORE CLUSTER DENSITY AND REQUIRE A MINIMUM NUMBER OF POINTS WITHIN A MAXIMUM DISTANCE (EPSILON) AROUND THE MEMBERS OF THE CLUSTERS. THE LIGHTER AREAS AROUND THEM ARE THE POINTS THAT SATISFY THE SEED CLUSTERS CRITERIA (EPSILON), BUT DO NOT MEET THE MINIMUM NUMBER OF POINTS REQUIRED. OTHER POINTS THAT DO NOT MEET EITHER OF THE REQUIREMENTS ARE REGARDED AS NOISE.

ANSELIN LOCAL MORAN'S I

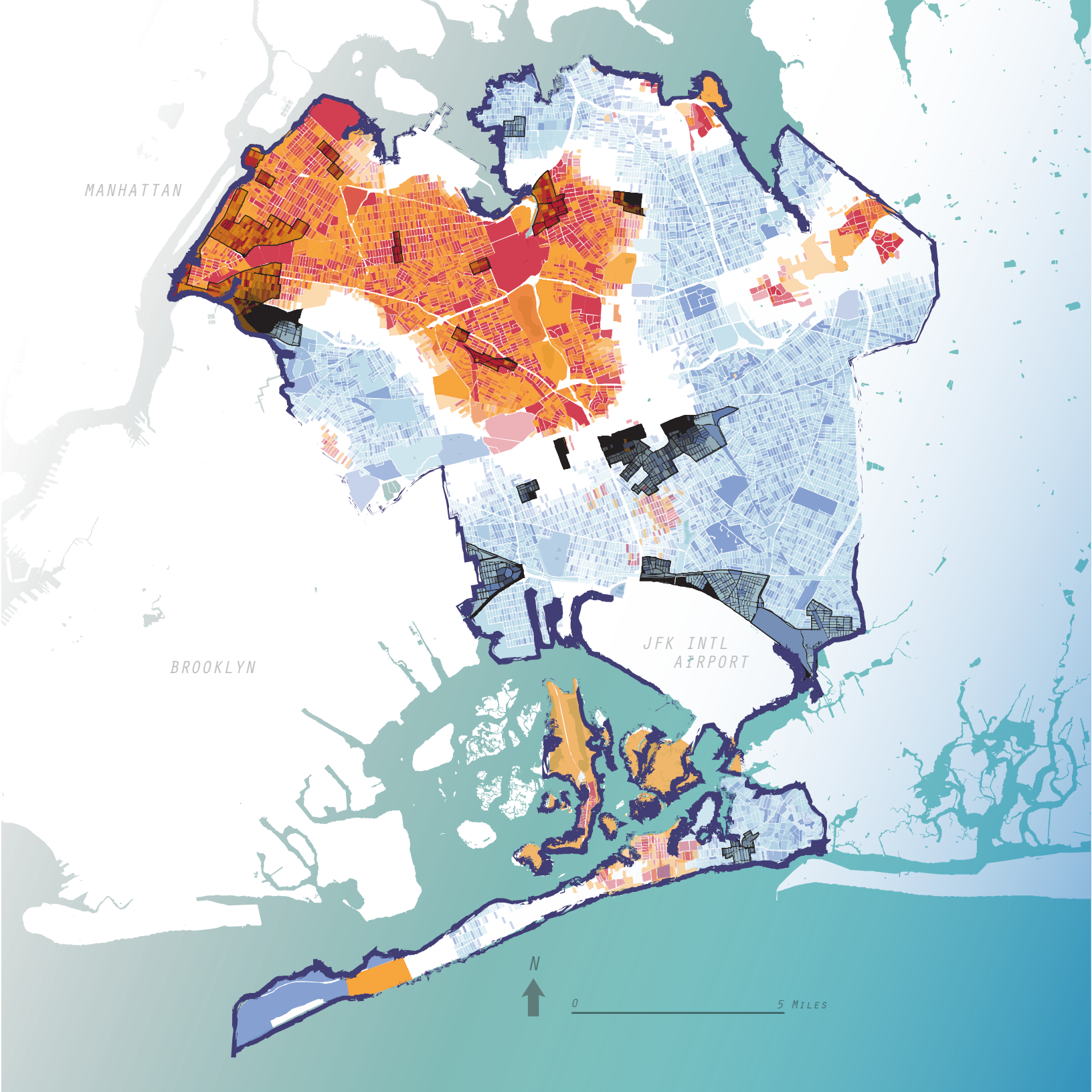
(SEPERATING CLUSTERS FROM OUTLIERS AND INSIGNIFICANT OBSERVATIONS)

INVERSE DISTANCE,
FIXED DISTANCE BAND VALUE,
WEIGHTED BY AGGREGATE OF
DEVELOPMENT ACTIVITIES

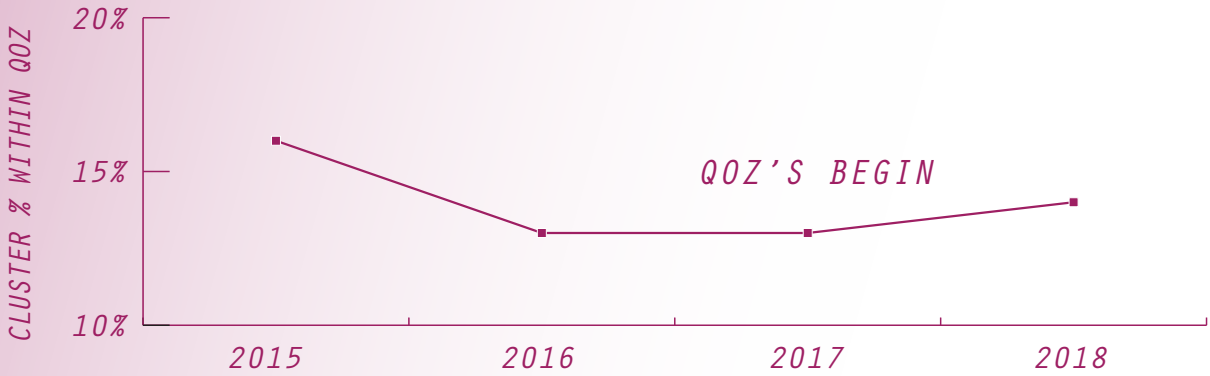


USING ANSELIN LOCAL MORAN'S I, THE LOCATIONS OF DEVELOPMENT ACTIVITIES WERE FIRST SPATIAL JOINED TO THE CENSUS BLOCKS. THE ANALYSIS USED THE NUMBER OF DEVELOPMENT ACTIVITIES WITHIN CENSUS BLOCKS ON AN ANNUAL BASIS. HIGH-DENSITY CLUSTERS CONSISTENTLY AGGREGATED IN SIMILAR AREAS OF QUEENS FOR ALL YEARS FROM 2015 TO 2018. THIS METHOD RECOGNIZED CLUSTERS OF CENSUS BLOCKS WITH BOTH HIGH AND LOW CONCENTRATIONS OF DEVELOPMENT ACTIVITY INCLUDING OUTLIERS, WHICH HAVE LOW VALUES AMID A HIGH-DENSITY CLUSTERS AND VICE VERSA. INSIGNIFICANT BLOCKS THAT WERE NEITHER PART OF THE CLUSTERS NOR THE OUTLIERS WERE SEPARATED. THE INSIGNIFICANT BLOCKS WERE REMOVED.

DBSCAN used point data to reveal clusters, while Anselin Local Moran's I used weighted census blocks. These two clustering analyses exhibited some similarities. A high concentration of development activity took place towards the eastern side of the borough near the waterfront and has progressed over towards central Queens around Queens Blvd leading onto Jackson Heights. Both of the time series analyses exhibited high density of clusters in similar vicinity. However, some clusters classified by DBSCAN had been categorized as low density clusters by Anselin Moran's I. This included the clusters of development that spills out to Queens from Bushwick (Brooklyn).


MANHATTAN
BROOKLYN
JFK INTL
AIRPORT
N
0
5 Miles

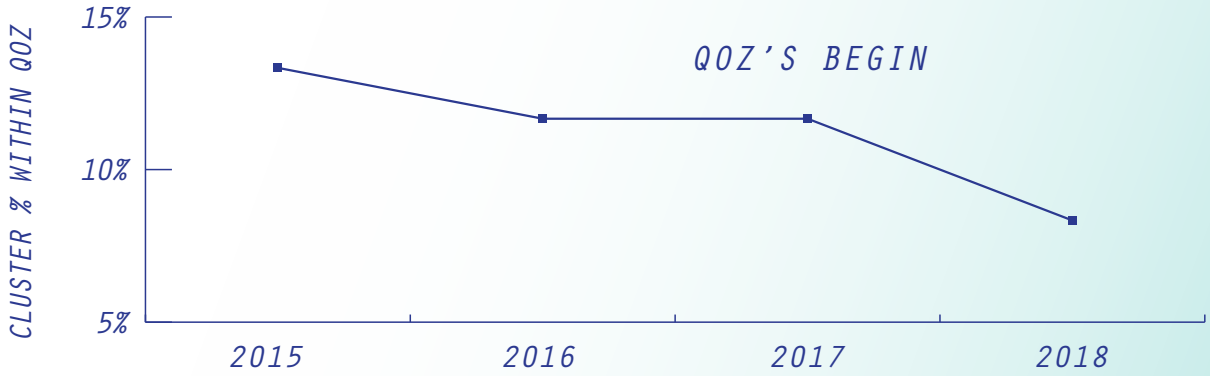
DBSCAN



Years	QOZ	Total	%
2015	1712	11035	16
2016	1525	12140	13
2017	1498	11612	13
2018	1026	7333	14

DBSCAN resulted in a slight increase in the percentage of clusters that were contained within QOZ's in 2018. This method revealed only high density clusters by discarding a proportion of development activity as noise. Therefore, the concentration of DBSCAN clusters suggested a geographic trend of development activity.

ANSELIN LOCAL MORAN'S I



Years	QOZ	Total	%
2015	1944	13927	14
2016	1750	13654	13
2017	1733	13333	13
2018	1398	12670	11

	AVG P-VALUE		
Years	QOZ	Clusters in QOZ's	Total
2015	0.030	0.006	0.072
2016	0.049	0.008	0.075
2017	0.090	0.008	0.090
2018	0.090	0.006	0.077

Anselin Local Moran's I resulted in a decrease in the percentage of clusters contained within QOZ's in 2018. However, this method suggested a high significance of clustering. The average p-value of the clusters in the whole borough in 2018 was 0.07, but the average p-value of clusters within QOZ's in 2018 was 0.006. This suggested a strong relationship between development activity around and within QOZ's.

4. CONCLUSION

This study asked if there was any early evidence of development activity within QOZ's in Queens, New York. The lack of significance in the proportion of clusters within QOZ's negated our hypothesis. There was evidence of spatial concentration and containment of early development activity within QOZ's, but those activities had been consistent in the area prior to the introduction of QOZ's in 2017. Therefore, it was postulated that the introduction of QOZ's could not be credited for the containment and concentration of development activity in Queens in 2018.

There was an increase in development activity in the borough as a whole. The amount of development activity within QOZ's in Queens increased as well. However, both of the cluster analyses conducted in this study suggested no notable increase in the spatial concentration of development activity clusters, defined by sales and permitting within QOZ's. The proportion of high-density clusters within QOZ's either declined or stayed relatively consistent. DBSCAN exhibited a slight increase in the proportion of clustered development activity. Anselin Local Moran's I exhibited a high significance of clusters despite a decrease in the proportion of clustered development activity.

Although the analyses suggested no immediately obvious or significant borough wide influence in the proportion of development within the incentivized areas in 2018, both analyses exhibited significance in the density of clusters, which may suggest evidence of spillover effects. One year of residual effects limits the scope of this study, but serves as the basis for future investigation.

5. BIBLIOGRAPHY

New York City Department of Finance.
“Detailed Annual Sales Reports by Borough: 2015 - 2016 New York City Rolling Sales Data”.
New York City Department of Finance. 2019.
https://www1.nyc.gov/site/finance/taxes/property-rolling-sales-data.page

New York City Department of City Planning.
"Administrative and Political Districts".
New York City Open Data. 2019.
https://www1.nyc.gov/site/planning/data-maps/open-data.page#district_political

New York City Department of City Planning.
"Facilities Database".
New York City Open Data. 2019.
https://www1.nyc.gov/site/planning/data-maps/open-data.page#city_facilities

New York City Department of Building.
"DOB Permit Issuance".
New York City Open Data. 2019.
https://data.cityofnewyork.us/Housing-Development/DOB-Permit-Issuance/ipu4-2q9a

New York State of Opportunity.
“The Opportunity Zone Program in New York State”.
Empire State Development. 2019.
https://esd.ny.gov/sites/default/files/FINAL-Recommendations-OZ-42618.pdf

Empire State Development. "Opportunity Zone Program”. February 28, 2018. Retrieved May 5, 2019.
https://esd.ny.gov/opportunity-zones

Economic Innovation Group. “Opportunity Zones”. (n.d.).
Retrieved May 5, 2019.
https://eig.org/opportunityzones

6. APPENDIX

R IMPLEMENTATION OF DBSCAN

```
## settings
set.seed(123)          # fix random number for reproduction
options(warn = -1)     # disable warning messages (0 = on, -1 = off)
options(scipen = 999)  # disable scientific notation

## load libraries

# data handling
library(lubridate)   # data cleaning, date / time
library(stringr)     # data cleaning, string whitespace removal
library(data.table)  # data cleaning, rename columns, matrix


## geographic info systems
library(sp)          # spatial join
library(rgdal)       # shapefile handling
library(rgeos)       # centroid
library(fpc)         # dbscan
library(spatstat)    # ripley's k
library(tmaptools)   # geolocation search
library(leaflet)     # interactive mapping

## load shapefile - land lots - wgs84 projection
qoz <- readOGR("queens_qoz/QN_OZ_trct.shp")
qoz <- spTransform(qoz, CRS("+proj=longlat +ellps=WGS84"))

## load shapefile - land lots - wgs84 projection
c_blocks <- readOGR("queens_blocks/queens_census_block.shp")
c_blocks <- spTransform(c_blocks, CRS("+proj=longlat +ellps=WGS84"))

## load data
sales <- read.csv("./queens_sales/all_queens_geo.csv",
                  sep  = ",",                 # separator values
                  header = TRUE,              # create headers
                  check.names = TRUE,         # auto change header names
                  strip.white = TRUE,         # remove leading/trailing whitespace
                  stringsAsFactors = TRUE,    # strings as factor data type
                  na.strings = c("", "NA"))  # assign blank cells na

## remove non-applicable features
sales <- sales[ , c("Sales_subset_real.SALEDATE",
                    "Queens_Lots_Point.Latitude",
                    "Queens_Lots_Point.Longitude")]

## rename headers
sales <- setnames(sales,
                  old = c("Sales_subset_real.SALEDATE",
                          "Queens_Lots_Point.Latitude",
                          "Queens_Lots_Point.Longitude"),
                  new = c("SALEDATE",
                          "LONGITUDE",
                          "LATITUDE"))

## convert to date type
sales$SALEDATE <- as.Date(ymd(sales$SALEDATE))

## subset data - application filings only, 2015
sales_2015 <- subset(sales,
```

```
                    SALEDATE <= "2016-01-01")

## subset data - application filings only, 2016
sales_2016 <- subset(sales,
                    SALEDATE >= "2016-01-01" &
                    SALEDATE < "2017-01-01" )

## subset data - application filings only, 2017
sales_2017 <- subset(sales,
                    SALEDATE >= "2017-01-01" &
                    SALEDATE < "2018-01-01" )

## subset data - application filings only, 2018
sales_2018 <- subset(sales,
                    SALEDATE >= "2018-01-01")

## create points
sales_pts <- sales[2:3]
sales_pts_2015 <- sales_2015[2:3]
sales_pts_2016 <- sales_2016[2:3]
sales_pts_2017 <- sales_2017[2:3]
sales_pts_2018 <- sales_2018[2:3]

## load data
permits <- read.csv("dob_permits.csv",
                    sep  = ",",  # separator values
                    header = TRUE,  # create headers
                    check.names = TRUE,  # auto change header names
                    strip.white = TRUE,  # remove leading/trailing whitespace
                    stringsAsFactors = TRUE,  # strings as factor data type
                    na.strings = c("", "NA"))  # assign blank cells na

## preview dimensions
dim(permits)

## preview variable names
names(permits)

## preview permit types
unique(permits$Permit.Status)

## preview filing types
unique(permits$Filing.Status)

## subset data - queens only
permits_queens <- subset(permits,
                         BOROUGH == "QUEENS")

## preview dimensions
dim(permits_queens)

## preview variable names
names(permits_queens)

## preview permit types
unique(permits_queens$Permit.Status)

## preview filing types
unique(permits_queens$Filing.Status)

## subset data - remove non-applicable features
permits_queens <- permits_queens[c("House..",
                                   "Street.Name",
                                   "Block",
                                   "Lot",
```

```
                                    "Permit.Status",
                                    "Filing.Status",
                                    "Filing.Date",
                                    "Issuance.Date",
                                    "LATITUDE",
                                    "LONGITUDE",
                                    "NTA_NAME")]

## clean dates - filing date
permits_queens$Filing.Date <- gsub(permits_queens$Filing.Date,
                                    pattern = "12:00:00 AM",
                                    replacement = "",
                                    fixed = TRUE);

## clean dates - issuance date
permits_queens$Issuance.Date <- gsub(permits_queens$Issuance.Date,
                                      pattern = "12:00:00 AM",
                                      replacement = "",
                                      fixed = TRUE);

## convert to date type
permits_queens$Filing.Date <- as.Date(mdy(permits_queens$Filing.Date))
permits_queens$Issuance.Date <- as.Date(mdy(permits_queens$Issuance.Date))

## subset data - date range scope
permits_queens <- subset(permits_queens,
                         Filing.Date >= "2015-01-01" &
                         Filing.Date < "2019-01-01")

## subset data - initial filings only
permits_queens <- subset(permits_queens,
                         Filing.Status == "INITIAL")

## remove duplicate observations
permits_queens <- permits_queens[!duplicated(permits_queens), ]

## remove rows with missing geocoords
permits_queens <- permits_queens[complete.cases(permits_queens$LATITUDE), ]

## annualize data

## subset data - application filings only, 2015
permits_queens_2015 <- subset(permits_queens,
                              Filing.Date <= "2016-01-01")

## subset data - application filings only, 2016
permits_queens_2016 <- subset(permits_queens,
                              Filing.Date >= "2016-01-01" &
                              Filing.Date < "2017-01-01" )

## subset data - application filings only, 2017
permits_queens_2017 <- subset(permits_queens,
                              Filing.Date >= "2017-01-01" &
                              Filing.Date < "2018-01-01" )

## subset data - application filings only, 2018
permits_queens_2018 <- subset(permits_queens,
                              Filing.Date >= "2018-01-01")

## create points
permits_queens_pts <- permits_queens[9:10]
permits_queens_pts_2015 <- permits_queens_2015[9:10]
permits_queens_pts_2016 <- permits_queens_2016[9:10]
permits_queens_pts_2017 <- permits_queens_2017[9:10]
```

```
permits_queens_pts_2018 <- permits_queens_2018[9:10]

## combine rows of sales and permit activity
sales_permits_pts <- rbind(sales_pts, permits_queens_pts)
sales_permits_pts_2015 <- rbind(sales_pts_2015, permits_queens_pts_2015)
sales_permits_pts_2016 <- rbind(sales_pts_2015, permits_queens_pts_2016)
sales_permits_pts_2017 <- rbind(sales_pts_2015, permits_queens_pts_2017)
sales_permits_pts_2018 <- rbind(sales_pts_2015, permits_queens_pts_2018)

## determine optimal epsilon
dbscan::kNNdistplot(sales_permits_pts,
                    k = sqrt(nrow(sales_permits_pts)))

## set epsilon
epsilon <- 0.0066

## draw epsilon line
abline(h = epsilon, lty = 2)

## conduct dbscan - 2015
dbscan_sales_permits_2015 <-
    fpc::dbscan(sales_permits_pts_2015,
                eps = epsilon,
                MinPts = sqrt(nrow(sales_permits_pts))
                );

## conduct dbscan - 2016
dbscan_sales_permits_2016 <-
    fpc::dbscan(sales_permits_pts_2016,
                eps = epsilon,
                MinPts = sqrt(nrow(sales_permits_pts))
                );

## conduct dbscan - 2017
dbscan_sales_permits_2017 <-
    fpc::dbscan(sales_permits_pts_2017,
                eps = epsilon,
                MinPts = sqrt(nrow(sales_permits_pts))
                );

## conduct dbscan - 2018
dbscan_sales_permits_2018 <-
    fpc::dbscan(sales_permits_pts_2018,
                eps = epsilon,
                MinPts = sqrt(nrow(sales_permits_pts))
                );

## duplicate points data frame
dbscan_results_2015 <- sales_permits_pts_2015
dbscan_results_2016 <- sales_permits_pts_2016
dbscan_results_2017 <- sales_permits_pts_2017
dbscan_results_2018 <- sales_permits_pts_2018

## cluster identification - 2015
dbscan_results_2015$CLUSTERID <- dbscan_sales_permits_2015$cluster
dbscan_results_2015$SEEDCLUSTER <- dbscan_sales_permits_2015$isseed

## cluster identification - 2016
dbscan_results_2016$CLUSTERID <- dbscan_sales_permits_2016$cluster
dbscan_results_2016$SEEDCLUSTER <- dbscan_sales_permits_2016$isseed

## cluster identification - 2017
dbscan_results_2017$CLUSTERID <- dbscan_sales_permits_2017$cluster
dbscan_results_2017$SEEDCLUSTER <- dbscan_sales_permits_2017$isseed
```

```
## cluster identification - 2018
dbscan_results_2018$CLUSTERID <- dbscan_sales_permits_2018$cluster
dbscan_results_2018$SEEDCLUSTER <- dbscan_sales_permits_2018$isseed

## subset clusters only - 2015
dbscan_clusters_2015 <- subset(dbscan_results_2015,
                               CLUSTERID >= 1)

## subset clusters only - 2016
dbscan_clusters_2016 <- subset(dbscan_results_2016,
                               CLUSTERID >= 1)

## subset clusters only - 2017
dbscan_clusters_2017 <- subset(dbscan_results_2017,
                               CLUSTERID >= 1)

## subset clusters only - 2018
dbscan_clusters_2018 <- subset(dbscan_results_2018,
                               CLUSTERID >= 1)

## export sales points to csv
write.csv(sales_pts, file = "sales_pts.csv")

## export permits points to csv
write.csv(permits_queens_pts, file = "permits_pts.csv")

## export dbscan results to csv - 2015
write.csv(dbscan_clusters_2015, file = "dbscan_clusters_2015.csv")

## export dbscan results to csv - 2016
write.csv(dbscan_clusters_2016, file = "dbscan_clusters_2016.csv")

## export dbscan results to csv - 2017
write.csv(dbscan_clusters_2017, file = "dbscan_clusters_2017.csv")

## export dbscan results to csv - 2018
write.csv(dbscan_clusters_2018, file = "dbscan_clusters_2018.csv")
```