AFFORDABLE HOUSING DEVELOPMENT

Group 4

Elio Aybar, Martin Copello,

Matthew Fligiel, Matt Norgren

Executive Summary

CMHDC
CHICAGO METROPOLITAN
Housing Development Corporation

CHA
CHICAGO HOUSING AUTHORITY™

CITY OF CHICAGO
INCORPORATED 4th MARCH 1837

Affordable housing developments aid cities in promoting diversity and inclusion, improving quality of life, enriching neighborhoods, and growing the local economy. In this study, we investigate whether the locations of affordable housing developments provide positive features for its population, relating to accessibility to public transportation, amenities, active commercial scene, walkability, and general neighborhood activity.

Through our investigations, we discovered stark discrepancies between neighborhoods with a multitude of affordable housing developments and those without. Generally, areas with scarce access to transit and worse socio economic performance contain the majority of affordable housing developments.

Project Objective

This project seeks to develop an end-to-end data pipeline for affordable housing data to:

- Examine data between the location, price, and size of affordable housing developments and standard-of-life features available for tenants.
- Foster a better understanding of:
  - Factors that affect location, prices, amenities, and quality of affordable housing developments.
  - The distribution of socio-economic indicators, economic activity, access to transit and parks, in each neighborhood.
- Potentially identify correlations that may confirm or deny the team's initial hypotheses related to affordable housing

Business Use Cases

Understanding affordable housing development's characteristics' relationship to socio-economic features could be used to develop more conscious policies and drive significant investments that provide inhabitants of affordable housing a better quality of life (i.e. development of additional public transportation in isolated neighborhoods, investment in local businesses, etc.)

Data

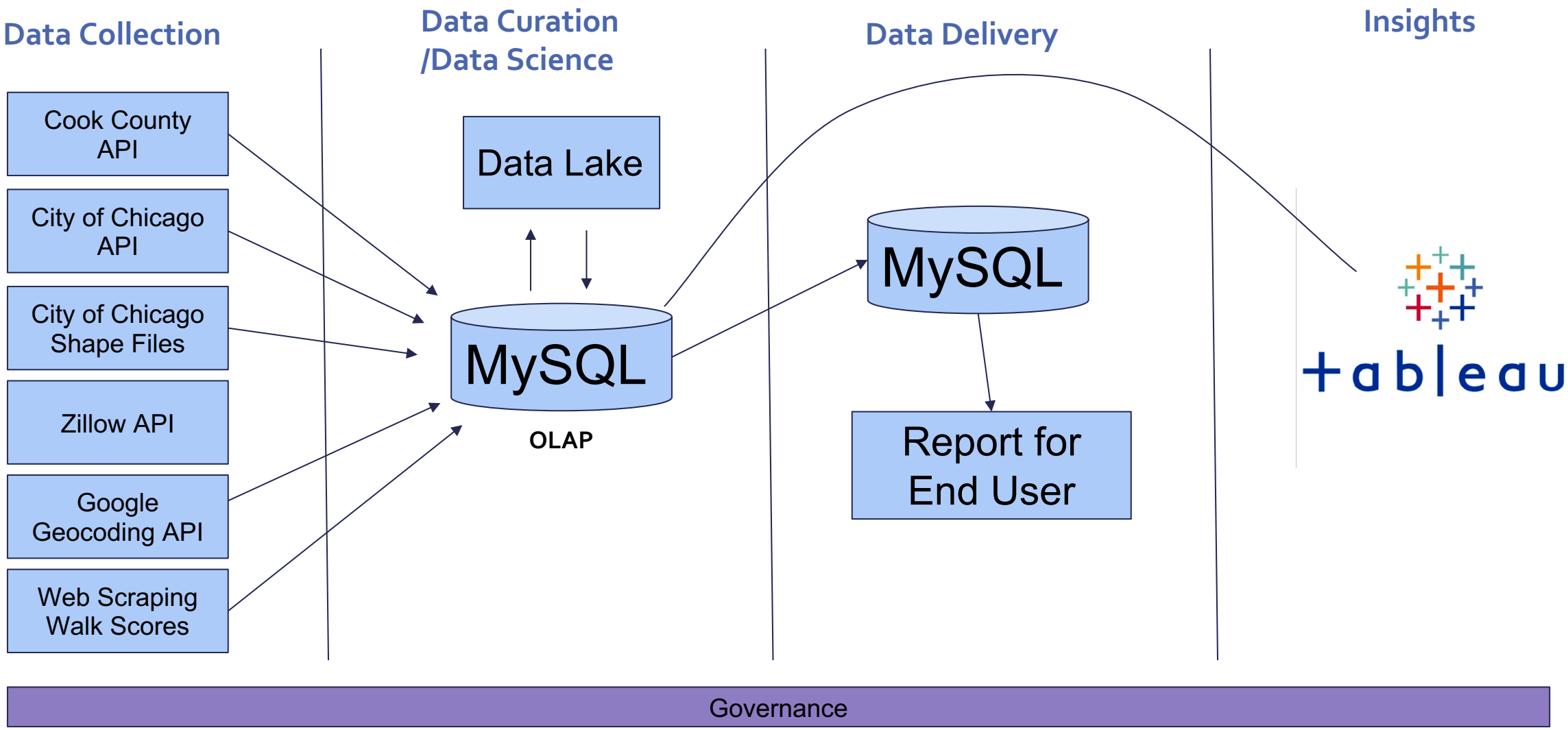
Data sources included:

- City of Chicago: Affordable Housing Development Dataset, Business Licenses Dataset, Park Dataset, Abandoned Property Datasets
- Zillow and Cook County Records API
- Neighborhood Shapefiles and Boundary files for GIS-type applications including generation of Table Keys (Neighborhoods), Public Transit and City Park GIS data points.
- Web-Scraping Walk-Scores of Chicago neighborhoods

Data Profile

- Reviewed data for 428 household developments, 24K units (processed data size)
- Raw data size: 88 features, 1.9M rows
- Data types used: csv, xls, Shapefiles, Boundary files. Additionally, we used pickle for data preservation. Developed .sql files for DDL/DML, and twbx or Tableau file
- Datasets included (the ones in bold were ultimately used): **Affordable housing developments, neighborhood shapefiles, abandoned buildings, business licenses, walkscores**, neighborhood crime, concentration of rideshares, **park location, CTA bus and train stop locations, per capita income, average household size, tax assessments of properties**.

Data Platform



Tools

Ingestion and Cleanup

pandas

Storage

MySQL

Delivery and Insights

+ a b | e a u

- Data was scraped via Python (Walk-Scores), downloaded from online sources (City of Chicago for Housing Developments, Commerce, Park) and gathered using APIs (Google Geocoding, Zillow, Cook County).
- Pandas was used for data cleanup.
- R was used to build an additional Geocoding API (Google's) to obtain coordinates from address.
- Our data database was built using MySQL. DDL and DML tables were created for each table.
- Tableau was used to generate tables and charts to draw insights from our data.

Data Cleanup

Out-of-scope Geographically

	A	B	C	D	E	F	G	H	I	J
	ID	LICENSE	ACCOU	SITE NU	LEGAL N	DOING	ADDRES	CITY	STATE	ZIP COD
1	1888878-2	2570948	21545	1	PSF MECH	PSF MECH	11621 E	SEATTLE	WA	98168
2	1888878-2	2705036	21545	1	PSF MECH	PSF MECH	11621 E	SEATTLE	WA	98168

Overwrote Assessed Value with Sale Value (where available)

```
df.loc[df['most_recent_sale_price'] == ('nan' or 'NaN')] = 0
df.most_recent_sale_price

#printing header names not values
def f(row):
    if 'most_recent_sale_price' != ('nan' or 'NaN'):
        'most_recent_sale_price'
    else:
        ('pri_est_land'+'pri_est_bldg')

df['price'] = df.apply(f,axis=1)

df.price
df['pri_est'] = (df.pri_est_land + df.pri_est_bldg)
df.pri_est
del df['pri_est_land']
del df['pri_est_bldg']
```

GIS Conversions

```
from geopy.geocoders import Nominatim
locator = Nominatim(user_agent="myGeocoder")
ln = locator.geocode("455 Cityfront Plaza Dr, Chicago, IL 60611")
print(ln.latitude, ln.longitude)
```

```
41.8903132 -87.62168978061763
```

```
In [10]: from shapely.geometry import Point
         def ca_assign4(pt):
             for i, v in enumerate(CA_boundaries):
                 if pt[0]>v[2]:
                     continue
                 if v[1]>pt[1]:
                     continue
                 if pt[1]>v[3]:
                     continue
                 if v[0]>pt[0]:
                     continue
                 elif CA_shapes[i].contains(Point(pt[0], pt[1])):
                     return i
                     break
```

```
In [11]: from shapely.geometry import Point
         import pickle
         CA_names = pickle.load(open("C:/Users/Martin/Desktop/CA_names.pkl", "rb"))
         CA_boundaries = pickle.load(open("C:/Users/Martin/Desktop/CA_boundaries.pkl", "rb"))
         CA_shapes = pickle.load(open("C:/Users/Martin/Desktop/CA_shapes.pkl", "rb"))
         nbc_tow = (ln.longitude, ln.latitude)
         CA_names[ca_assign4(nbc_tow)]

Out[11]: 'NEAR NORTH SIDE'
```

```
Source on Save | Run | Source
getwd()

# Geocoding a csv column of "addresses" in R

#load ggmap
library(ggmap)

# Select the file from the file chooser
fileToLoad <- file.choose(new = TRUE)

# Read in the CSV data and store it in a variable
origAddress <- read.csv(fileToLoad, stringsAsFactors = FALSE)

# Initialize the data frame
geocoded <- data.frame(stringsAsFactors = FALSE)

# Loop through the addresses to get the latitude and longitude of each address and add it to the
# origAddress data frame in new columns lat and lon
for(i in 1:nrow(origAddress))
{
  # Print("Working...")
  result <- geocode(origAddress$addresses[i], output = "latlona", source = register_google(key = "AIzaSyDe9YqYsf7_0pqUD21
  origAddress$lon[i] <- as.numeric(result[1])
  origAddress$lat[i] <- as.numeric(result[2])
  origAddress$geoAddress[i] <- as.character(result[3])
}
# Write a CSV file containing origAddress to the working directory
write.csv(origAddress, "geocoded.csv", row.names=FALSE)
```

Design Considerations I: Data

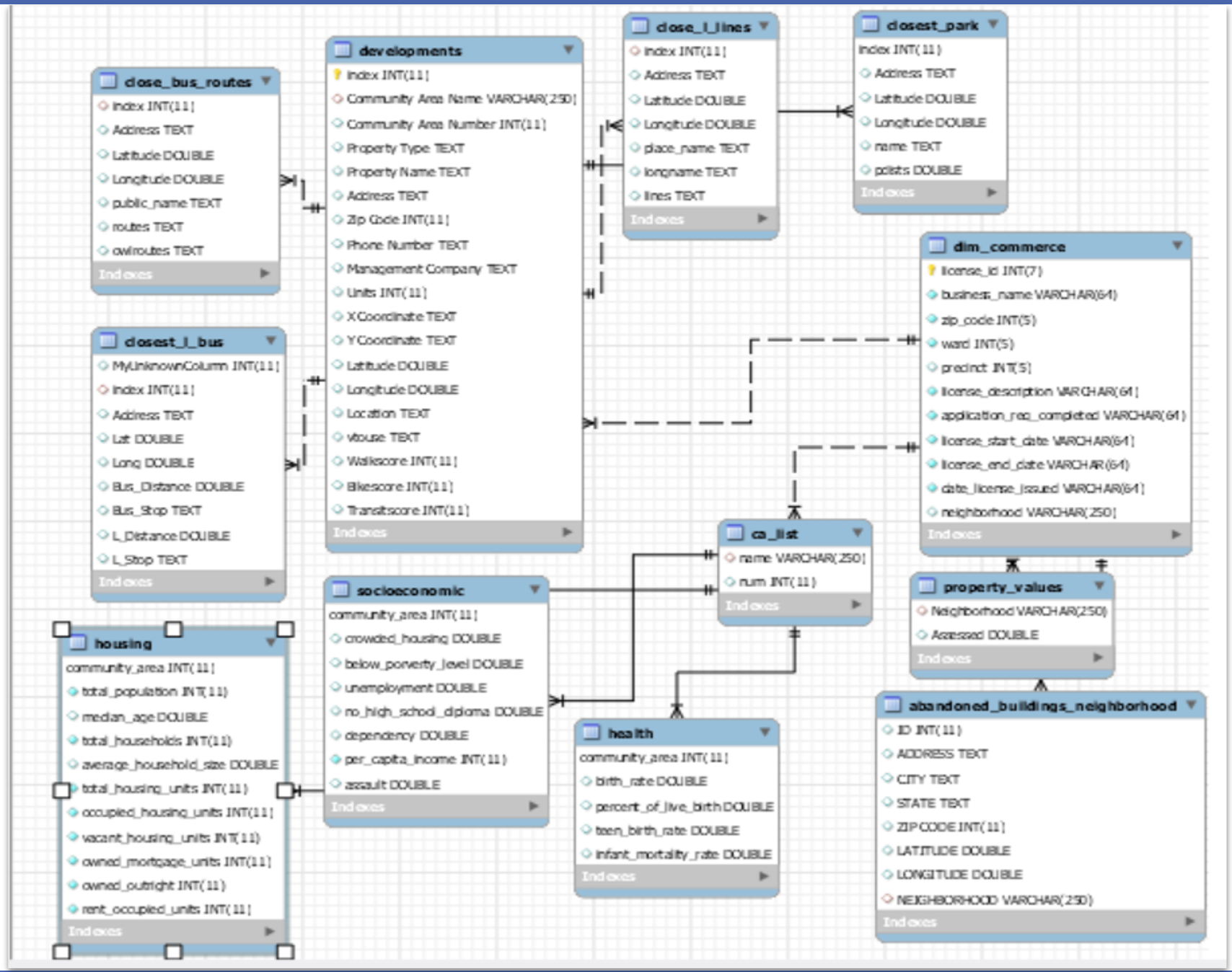
- ETL Scripts written in Python and R

- Keys were generated for tables (neighborhood from latitude and longitude, or from address):
  - Chicago neighborhood shapefiles
  - Geocodes via GeoPy and Google's Geocoding API via GCP to obtain Latitude and Longitude from address
  - Web-scraping Walk-Scores to calculate "walkability" of area

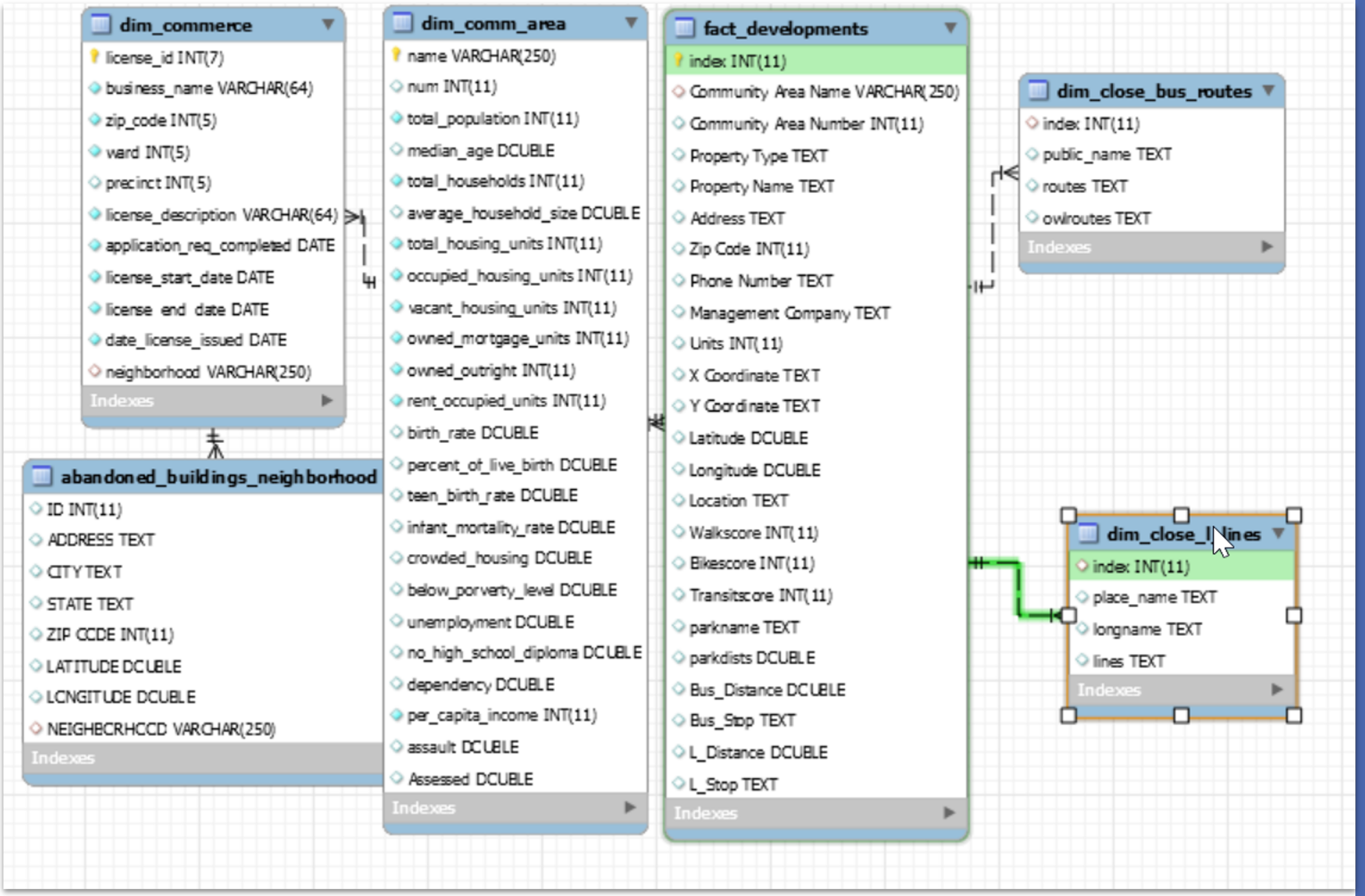
Design Considerations II: Implementation

- Number of tables and joins are not extensive, so this pipeline can be implemented locally given enough disk space
- Due to the archival nature of our data, our application can be used as an OLAP system.
- Snowflake data model
- One to many relationship between individual address and neighborhood requires definition of granularity at neighborhood level

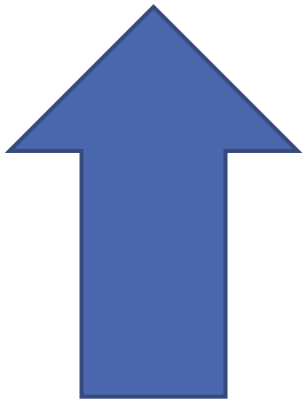
EER Diagram


close_bus_routes
Index INT(11)
Address TEXT
Latitude DOUBLE
Longitude DOUBLE
public_name TEXT
routes TEXT
owlroutes TEXT
Indexes
developments
Index INT(11)
Community Area Name VARCHAR(250)
Community Area Number INT(11)
Property Type TEXT
Property Name TEXT
Address TEXT
Zip Code INT(11)
Phone Number TEXT
Management Company TEXT
Units INT(11)
X Coordinate TEXT
Y Coordinate TEXT
Latitude DOUBLE
Longitude DOUBLE
Location TEXT
Wards TEXT
Walkscore INT(11)
Bikescore INT(11)
Transitscore INT(11)
Indexes
close_l_lines
Index INT(11)
Address TEXT
Latitude DOUBLE
Longitude DOUBLE
place_name TEXT
longname TEXT
lines TEXT
Indexes
closest_park
Index INT(11)
Address TEXT
Latitude DOUBLE
Longitude DOUBLE
name TEXT
points DOUBLE
Indexes
closest_l_bus
MyUnknownColumn INT(11)
Index INT(11)
Address TEXT
Lat DOUBLE
Long DOUBLE
Bus_Distance DOUBLE
Bus_Stop TEXT
L_Distance DOUBLE
L_Stop TEXT
Indexes
dim_commerce
license_id INT(7)
business_name VARCHAR(64)
zip_code INT(5)
ward INT(5)
precinct INT(5)
license_description VARCHAR(64)
application_req_completed VARCHAR(64)
license_start_date VARCHAR(64)
license_end_date VARCHAR(64)
date_license_issued VARCHAR(64)
neighborhood VARCHAR(250)
Indexes
ca_list
name VARCHAR(250)
num INT(11)
Indexes
socioeconomic
community_area INT(11)
crowded_housing DOUBLE
below_poverty_level DOUBLE
unemployment DOUBLE
no_high_school_diploma DOUBLE
dependency DOUBLE
per_capita_income INT(11)
assault DOUBLE
Indexes
property_values
Neighborhood VARCHAR(250)
Assessed DOUBLE
Indexes
housing
community_area INT(11)
total_population INT(11)
median_age DOUBLE
total_households INT(11)
average_household_size DOUBLE
total_housing_units INT(11)
occupied_housing_units INT(11)
vacant_housing_units INT(11)
owned_mortgage_units INT(11)
owned_outright INT(11)
rent_occupied_units INT(11)
Indexes
health
community_area INT(11)
birth_rate DOUBLE
percent_of_live_birth DOUBLE
teen_birth_rate DOUBLE
infant_mortality_rate DOUBLE
Indexes
abandoned_buildings_neighborhood
ID INT(11)
ADDRESS TEXT
CITY TEXT
STATE TEXT
ZIPCODE INT(11)
LATITUDE DOUBLE
LONGITUDE DOUBLE
NEIGHBORHOOD VARCHAR(250)
Indexes

Dimensional Model

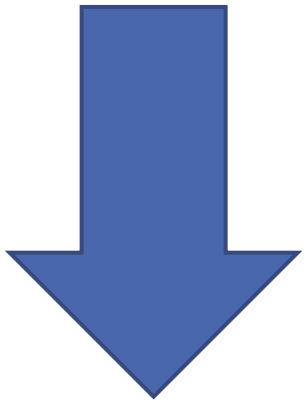


Preliminary Expectations



Greater number of affordable housing developments

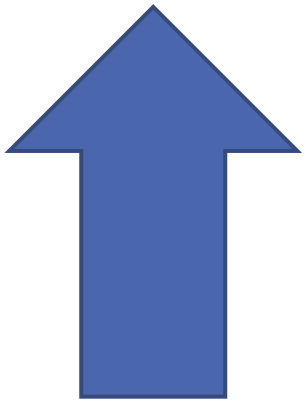
- In less affluent areas
- Near abandoned buildings
- In highly-commercialized areas



Fewer number of affordable housing developments

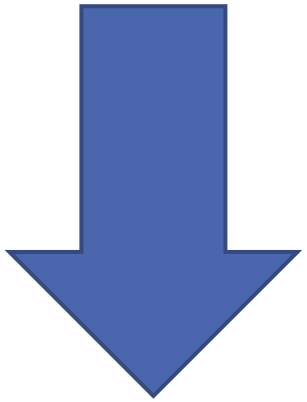
- Near public transit
- Near parks

Actual Results



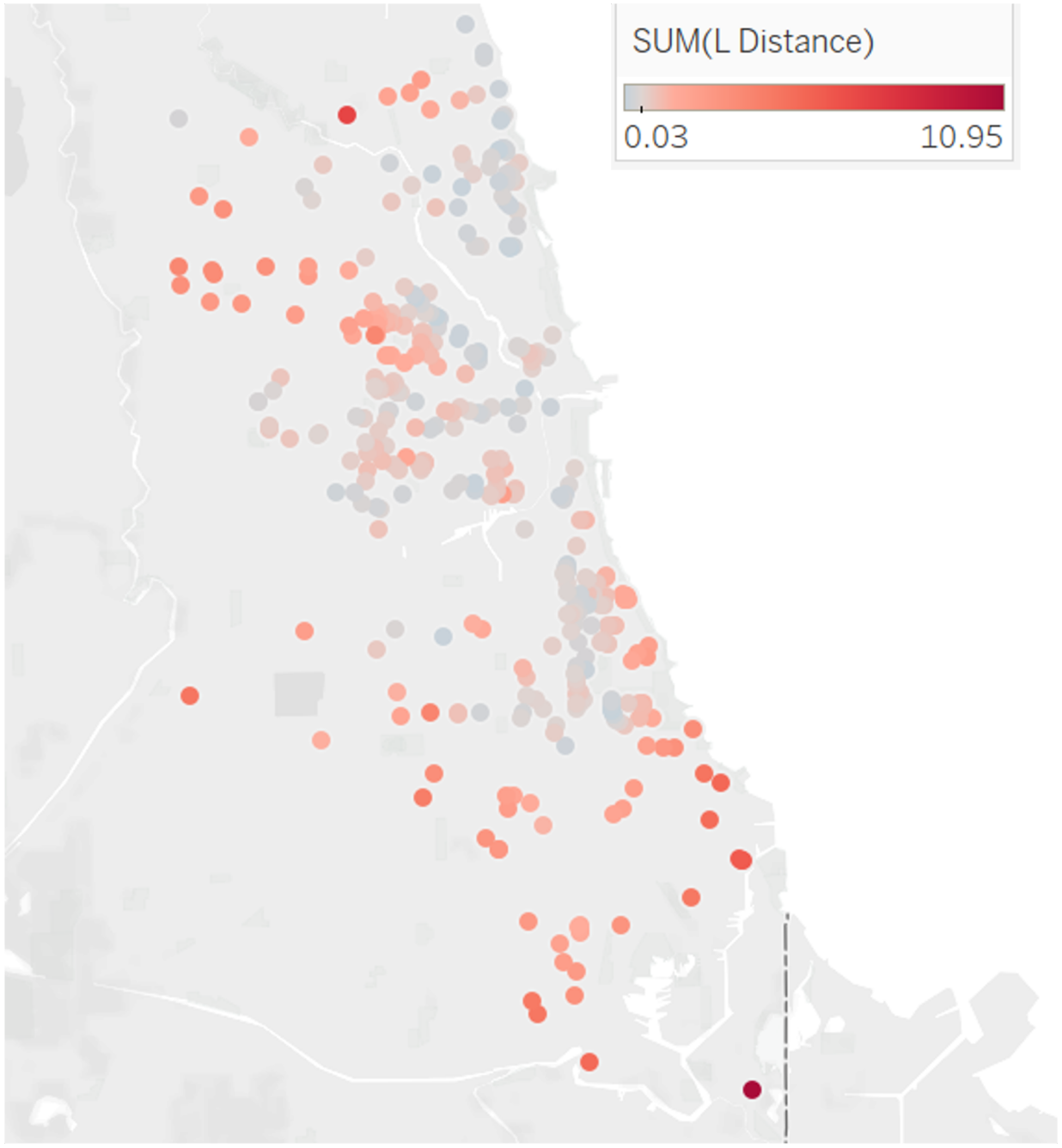
Greater number of affordable housing developments

- In less affluent areas ✓
- Near abandoned buildings ~
- In highly-commercialized ~ areas



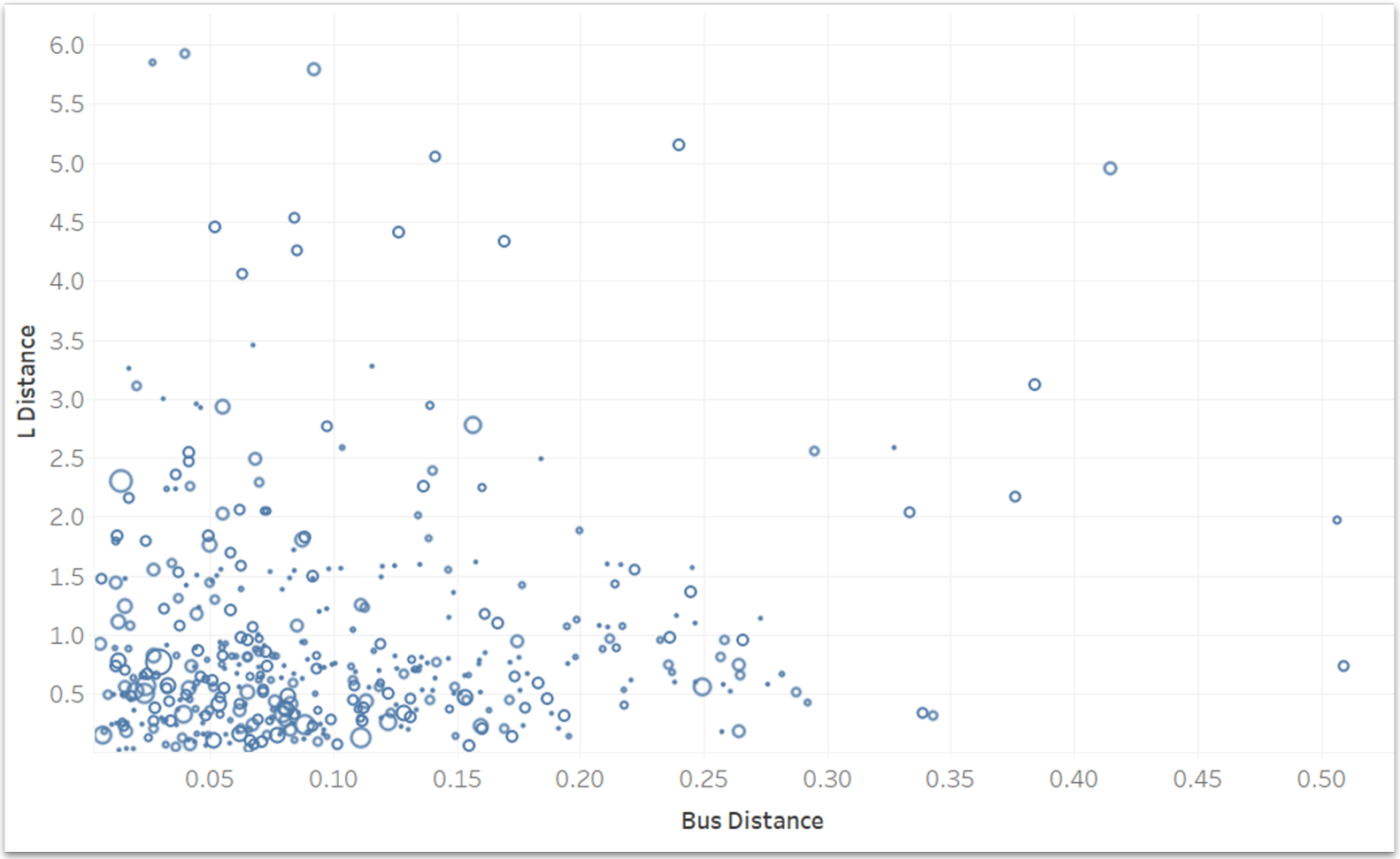
Fewer number of affordable housing developments

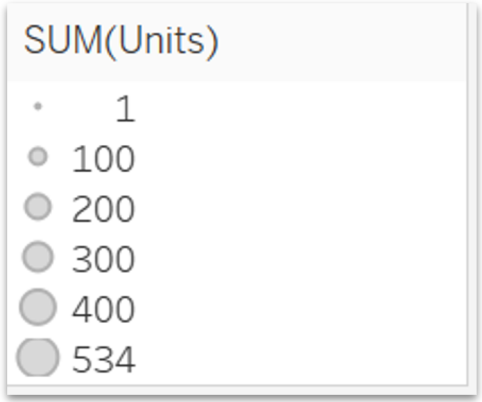
- Near public transit ~
- Near parks X



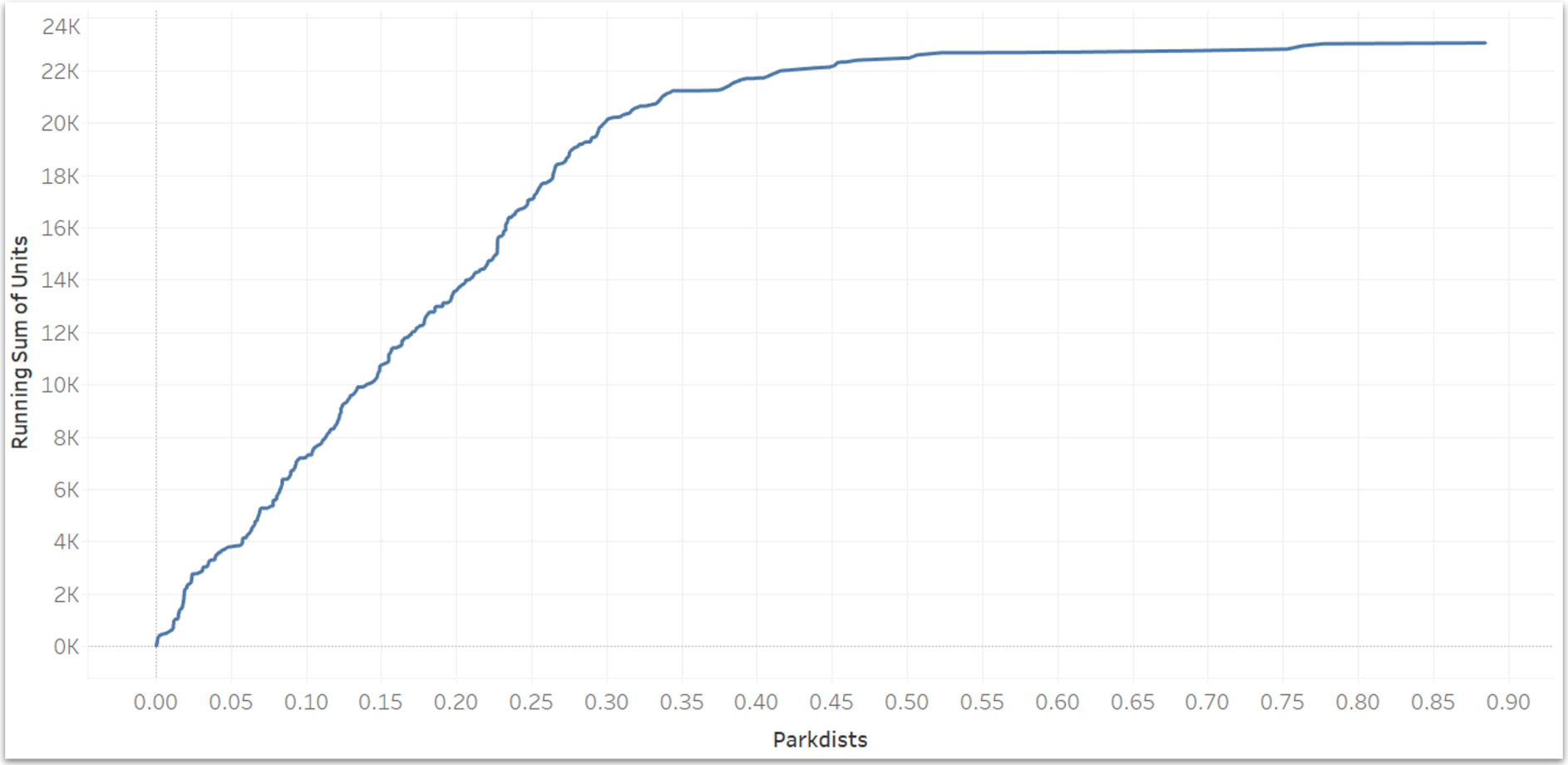
Distance from Housing Developments to One Instance of Public Transportation

Number of Units by Bus and L Distance

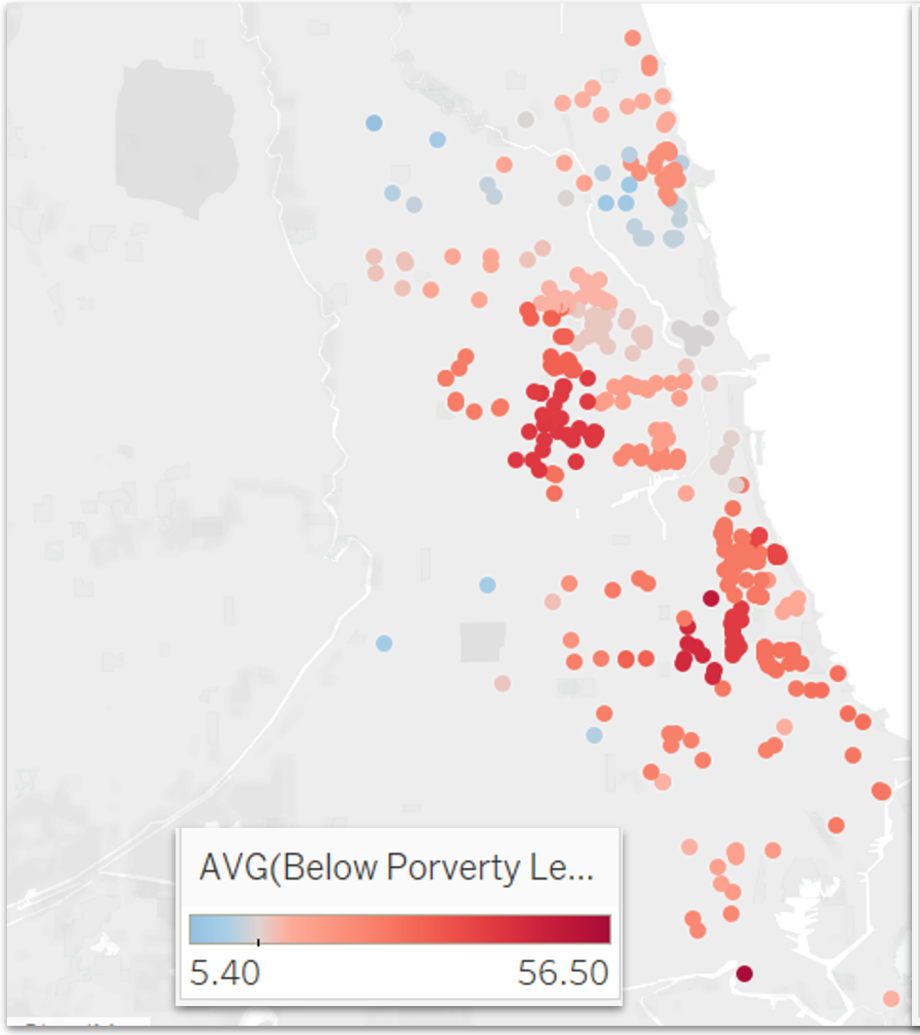


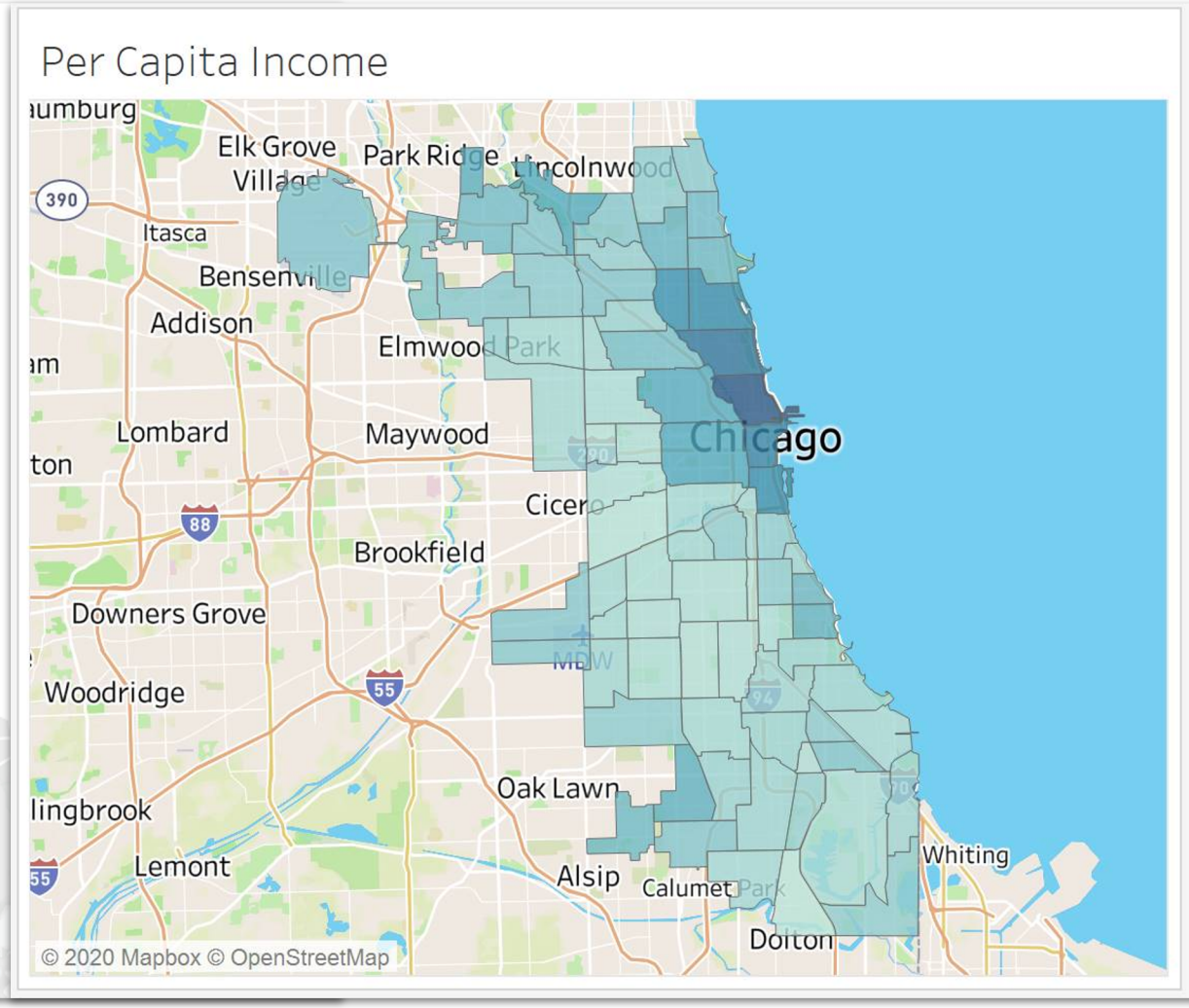


Most units did have access to nearby parks

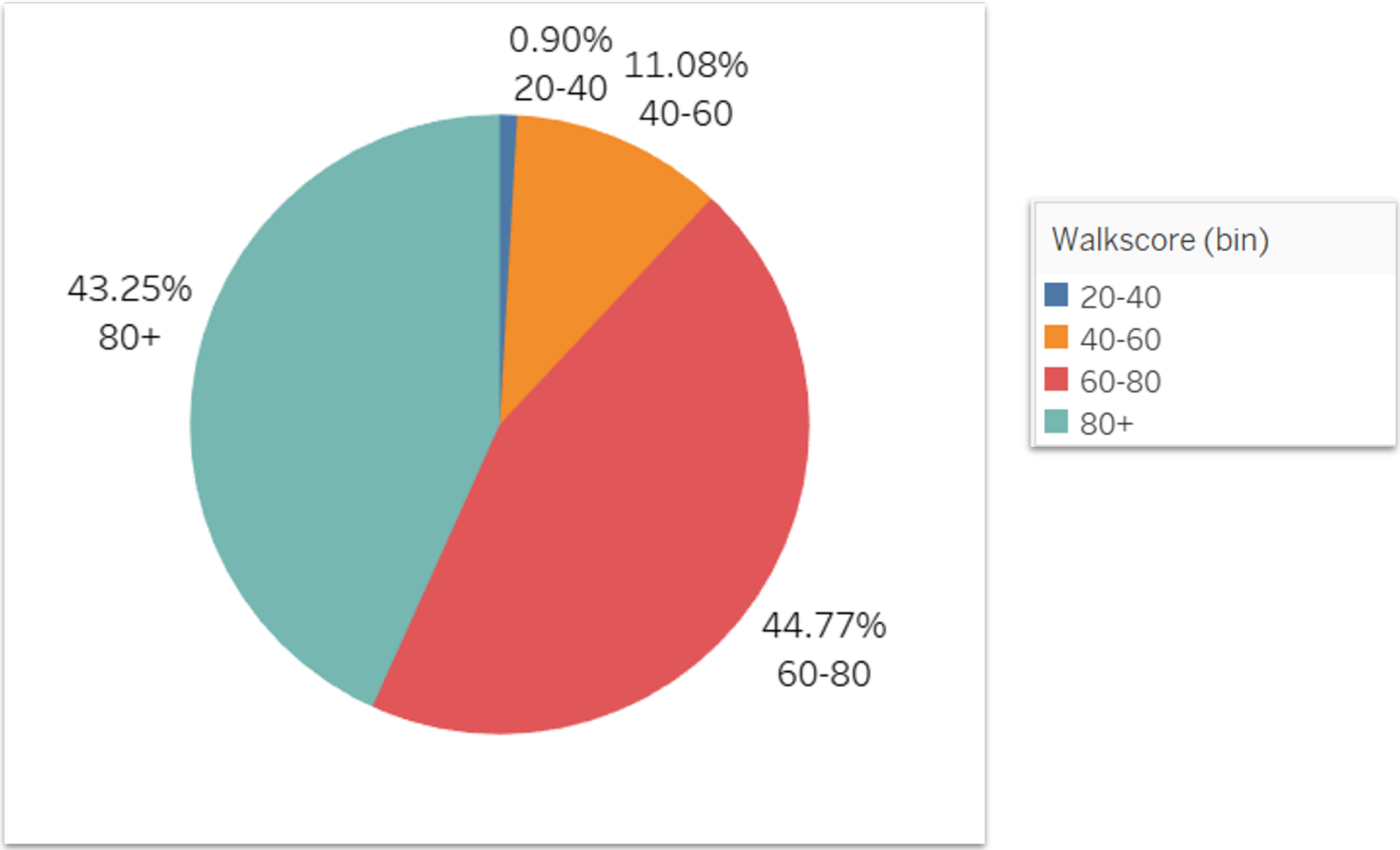


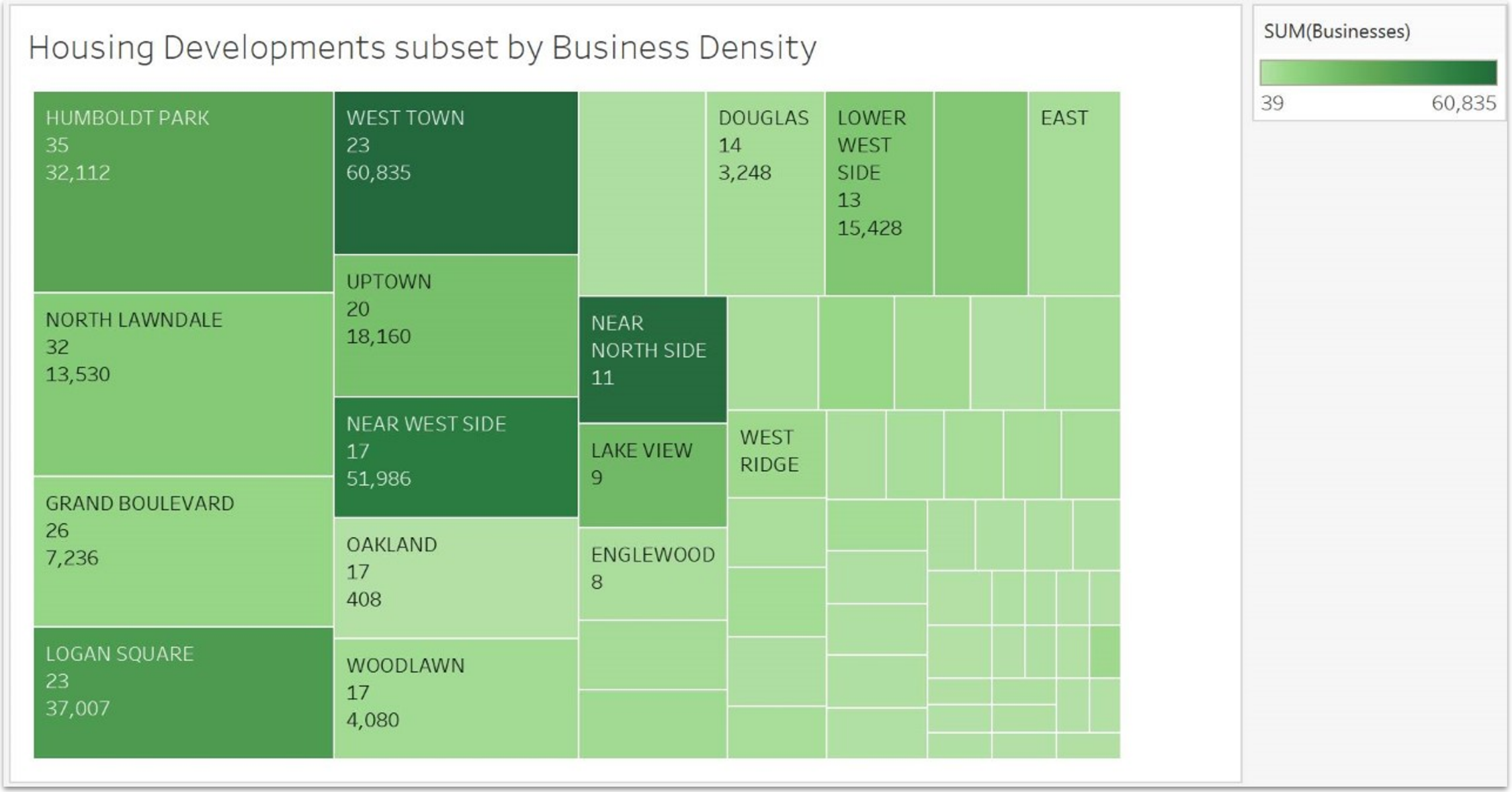
Affordable Housing Developments and Percentage of Neighborhood Below Poverty Line (Centered at 13%)



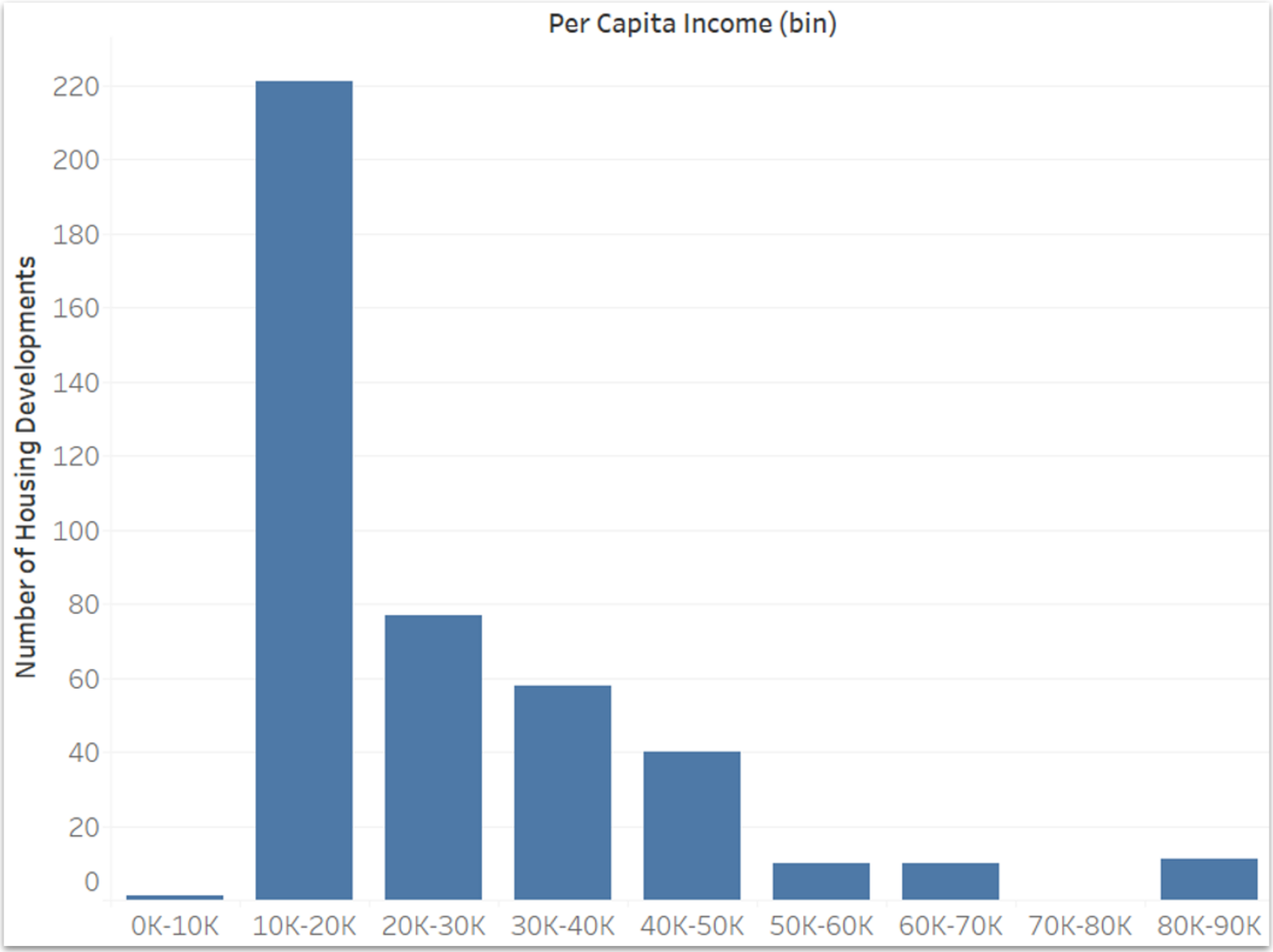


Walkscore (Binned by # of Units)



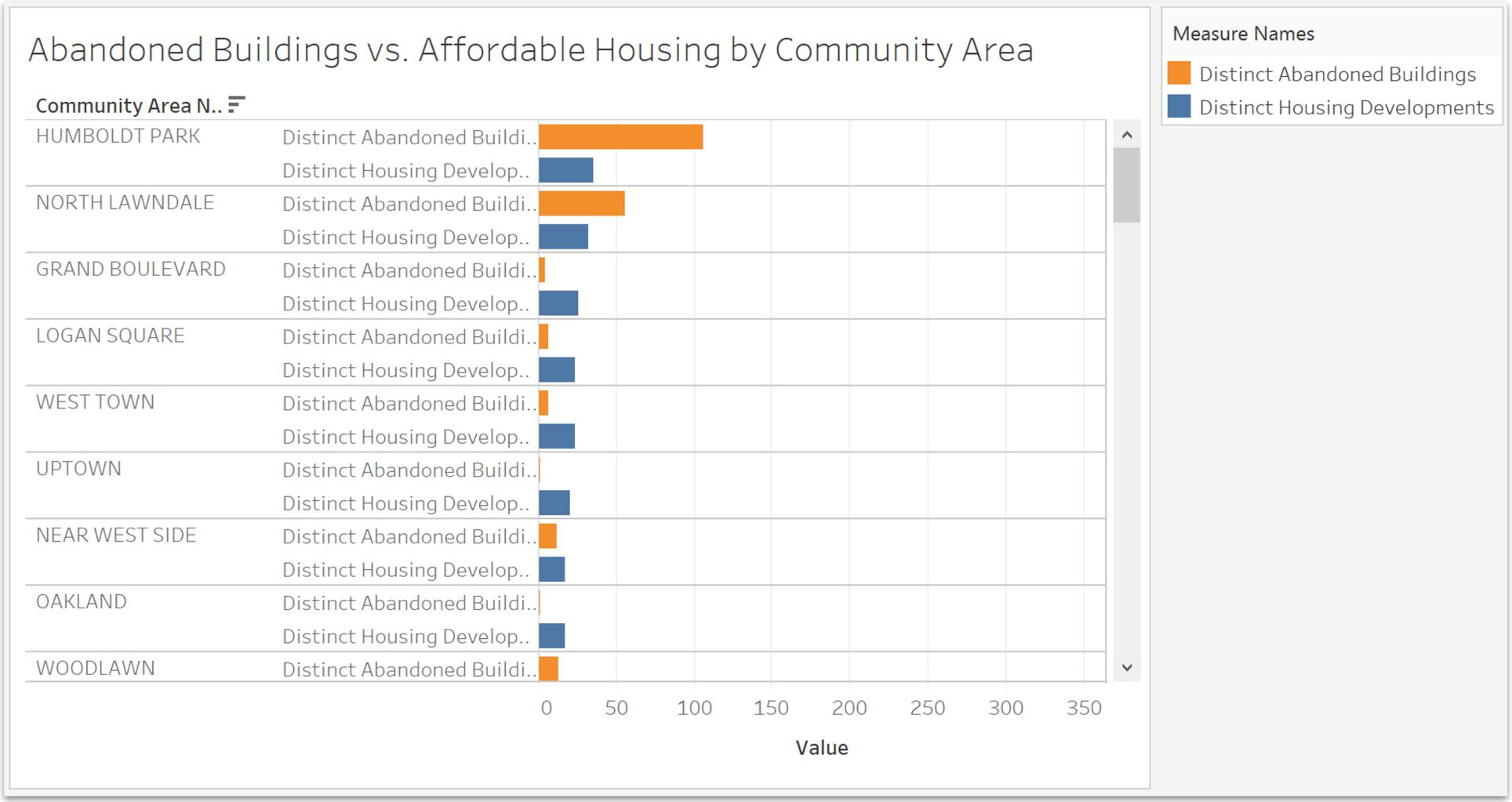

Housing Developments subset by Business Density
HUMBOLDT PARK
35
32,112
WEST TOWN
23
60,835
DOUGLAS
14
3,248
LOWER WEST SIDE
13
15,428
EAST
NORTH LAWNDALE
32
13,530
UPTOWN
20
18,160
NEAR NORTH SIDE
11
NEAR WEST SIDE
17
51,986
LAKE VIEW
9
WEST RIDGE
GRAND BOULEVARD
26
7,236
OAKLAND
17
408
ENGLEWOOD
8
LOGAN SQUARE
23
37,007
WOODLAWN
17
4,080
SUM(Businesses)
39
60,835

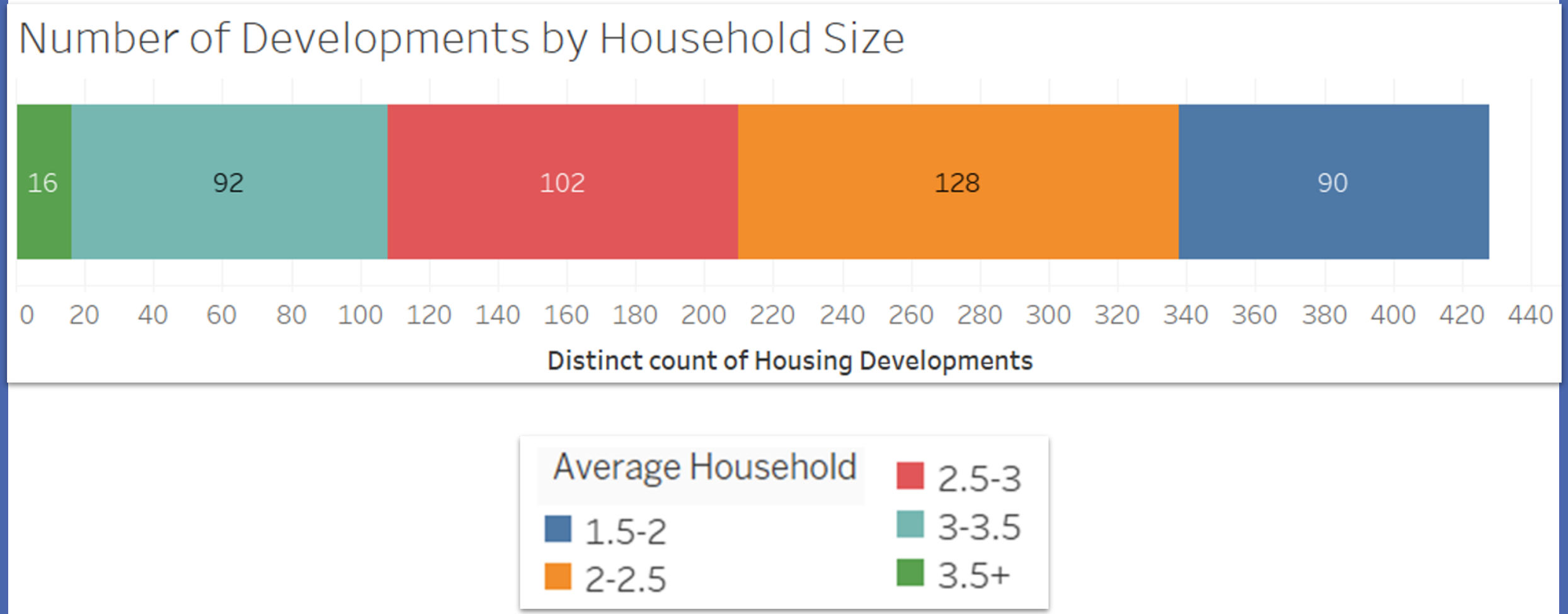
Developments by Per Capita Income



Count of Affordable Housing Developments

Count of Abandoned Buildings




Number of Developments by Household Size
16
92
102
128
90
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380 400 420 440
Distinct count of Housing Developments
Average Household
1.5-2
2-2.5
2.5-3
3-3.5
3.5+

Conclusion

Through the course of the project we discovered strong correlations between the count of affordable housing developments per neighborhood and:

- Socioeconomic status of neighborhood, which could perhaps be explained by higher land costs for more affluent neighborhoods
- Transit scores of neighborhood for L trains, which could perhaps further isolate neighborhoods with lower household income per capita.

Additionally, weaker correlations were identified between the count of affordable housing developments per neighborhood and:

- Business density of each neighborhood
- Count of abandoned buildings of each neighborhood

It is difficult to draw conclusions from these weaker correlations.

Areas of Improvement for Project

- With enough computing resources and time, further granularity could be utilized to map by-block impacts, driven by the by address level development, assessment, and transportation data, rather than by the neighborhood or area level. This could yield further insight and inform policy by visualizing ripple effects related to affordable housing developments and the blocks in which they are built.
- Obtaining data from the 2020 U.S. Census and re-evaluate our findings
- Study additional datasets (type of housing, family size, health background, etc.)

MongoDB

A document database could have served our project well

- We would have had one collection, with each document being a development
- Many of the attributes, including arrays of nearby L and Bus routes, could have been contained in the document
- Within each document, a community area subdocument would be included, with demographic information, and abandoned buildings/commerce for that CA
- This would have duplicated some data, but would be efficient

- Development Information
  - Transit Arrays
- Community Area
  - CA demographics
  - CA abandoned buildings
  - CA commerce

Neo4J

In a graph database, we would arrange it in the following way:

- Each development is a node
- Each CA would likely not be included, but just used as a description for the development
  - This format would not work well for the abandoned buildings and commerce, which would add many nodes and make the database harder to use
- Nodes could be connected by distance, but the use case for edges would be limited
- Thus, Neo4J would not be an optimal way of representing this data

References

- City of Chicago Data Portal: https://data.cityofchicago.org/
- Institute for Housing Studies at DePaul University: https://www.housingstudies.org/
- Google Maps Platform: https://developers.google.com/maps/documentation
- Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython 1st Edition by Wes McKinney


ANY QUESTIONS?