Sean Connelly

UPP 465

Professor Tilahun

5/5/20

Final Project – The 15-Minute City

**Introduction**

This report summarizes an effort to create a simple, open source, reproducible set of livability metrics for neighborhoods in American cities using Chicago, Illinois as a case study. The remainder of this paper is structured as follows. First, this document will offer some context around the notion of a “15-minute city” and highlight similar planning efforts underway throughout the world. Then, this report will walk through the methodology and data sources, before finally discussing the results of the analysis and lessons for future efforts.

**Context**

Paris Mayor Anne Hidalgo has long been considered a leader in urbanism. Since her election in 2014, Hidalgo has banned cars from along the Seine, planted thousands of trees, remodeling public spaces, and rolled out dozens of miles of bikes lanes. Hidalgo’s reelection campaign, however, is proposing to go even further by transforming Paris into a “15-minute city” by providing citizens “with grocery stores, parks, cafes, sports facilities, health centers, and workplaces easily accessible within a 15-minute walk or bike ride” (Cobbs, 2020).

Of course, the concept of a “15-minute city”, or some other variant emphasizing neighborhood livability, is not new. In Oregon, the 2012 Portland Plan aims to have 90 percent of all residents be within a 20-minute walk of all daily necessities, outside of work (City of Portland, 2012). Syndey, Australia is pursuing a similar concept, and Chicago planning commissioner Maurice Cox is an avowed proponent of the model (Moore, 2019).

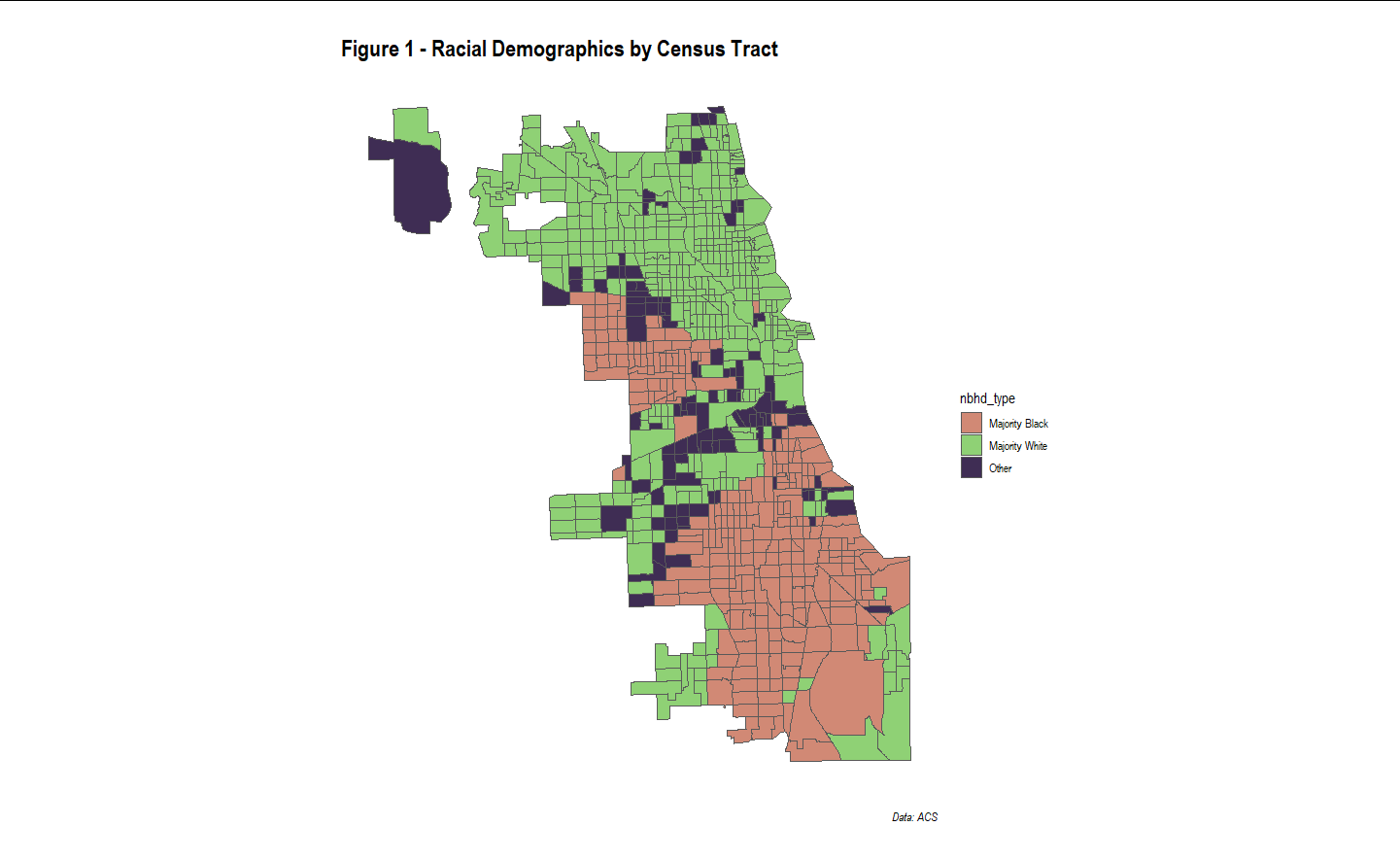
Reviewing the published methodologies behind Deloitte’s *ImagineSydney* plan (2018) and the Lane Council of Governments’ application of the Portland walkability index (2012), it is clear that there is no consistent definition, making comparisons across space and time impossible. For example, the Deloitte analysis uses a 30-minute travel-shed that includes trips taken by automobiles, with a focus on employment accessibility. The Lane COG study, however, uses a 20-minute walkshed that uses a small 33’ by 33’ raster grid as the unit of analysis Studies also rely on proprietary data sources like Google’s Places API, which might provide more nuanced insights, but make it difficult to replicate and are often abstracted from meaningful political and geographical units like wards. If a “15-minute city” is to be a mayor’s main policy goal, evidence that her or his policies have achieved this target is necessary.

**Methodology**

This report conducts an analysis of local accessibility in Chicago for the city’s 801 Census Tracts. After using the *dodgr* package to calculate walking isochrones from tract centroids, it uses the *osmdata* package to get OpenStreetMap feature data for information on stores, parks, health centers, and the like. Weighted by distance and population, this produces comparable metrics across tracts that also be cut by community area. By using the Census Tract as the essential unit of analysis, demographic data and other indicators can be used to track how equitable these livable neighborhoods are by race and income, and perhaps can serve as a baseline for a longitudinal analysis. The full code for this analysis can be found in the Appendix.

*Get Tract Data*

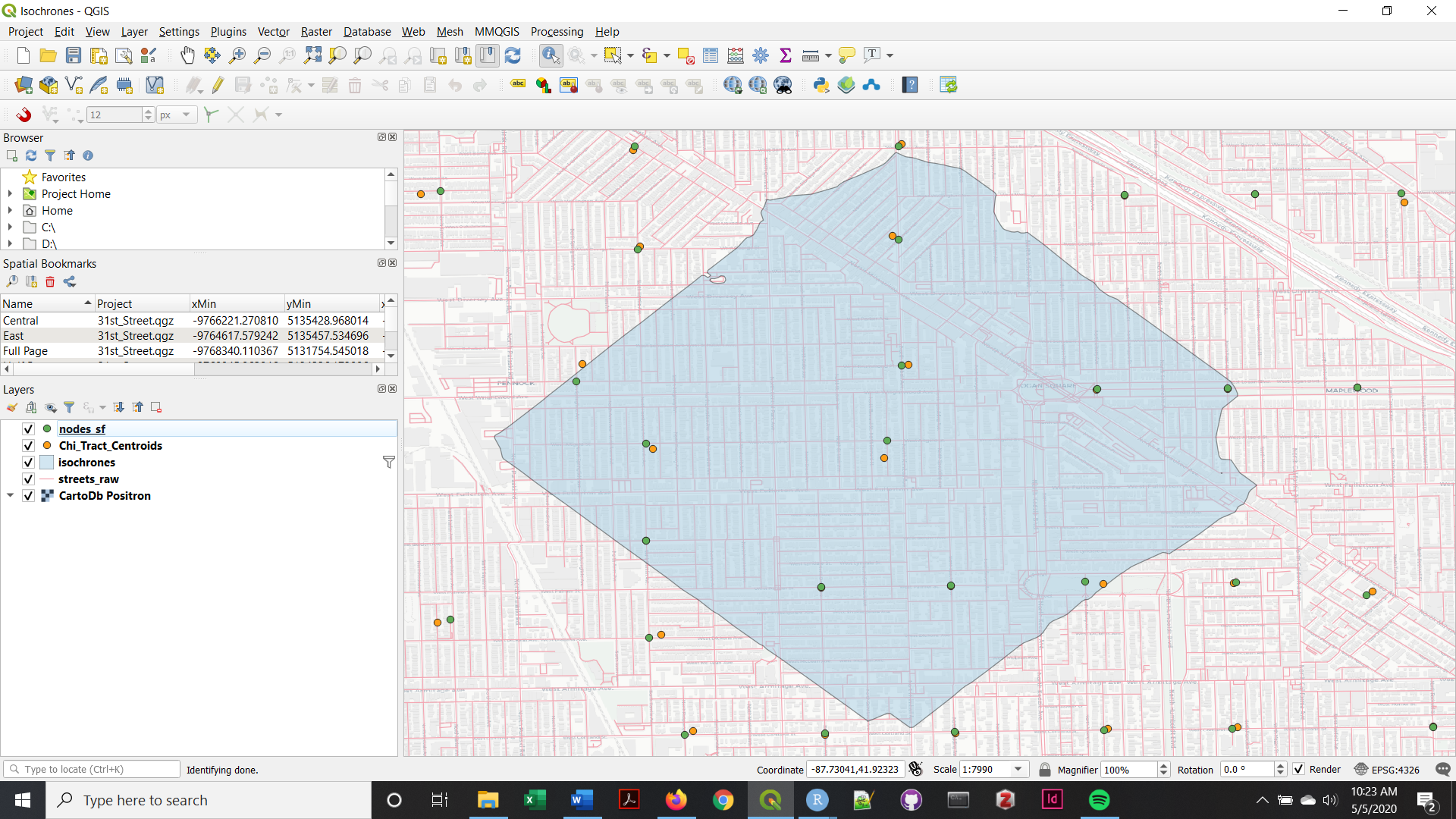
The first step requires downloading the relevant spatial and attribute data for the Census Tracts in question. This analysis uses the *tidycensus* package to get basic racial and income characteristics for the city’s 801 Census Tracts via an API call. This allows one to quickly construct a basic neighborhood typology to examine neighborhood amenity access from an equity standpoint, notably whether the majority of a tract’s population is black, white, or other (which encompasses situations where one race may be a plurality but not a majority, etc.). Figure 1 shows that 280 tracts (35%) are Majority Black, 412 tracts (51%) are Majority White, and 109 tracts (14%) are other.



*Build Isochrones*

The next step was to link the tracts to the underlying street network. While many analyses use uniform distance buffers of one-quarter or one-half miles, this is not always an accurate depiction of local accessibility. In Chicago, neighborhoods located beside highways or industrial corridors would have their access to amenities overstated by this kind of proxy measurement. Instead, this report uses the *dodgr* package to download a complete network of local residential street, and then constructs a 15-minute walkshed from the network node that is the closest to a respective tract centroid. Figure 2 demonstrates this process in action for the Census Tract with GEOID 17031220601. The tract centroid (orange) is approximated by the nearest node in the street network (green), from which an isochrone is built. The *dodgr* package does offer this capability, however, the package is still under development and the author faced memory constraints when attempting to process this calculation for each of the 801 Census Tracts. Instead, the nodes and street network were fed into QGIS’ QNEAT3 Network Analysis Tool, which created a 15-minute walkshed iteratively for each feature.

**Figure 2 – Isochrone for Tract with GEOID = 17031220601**

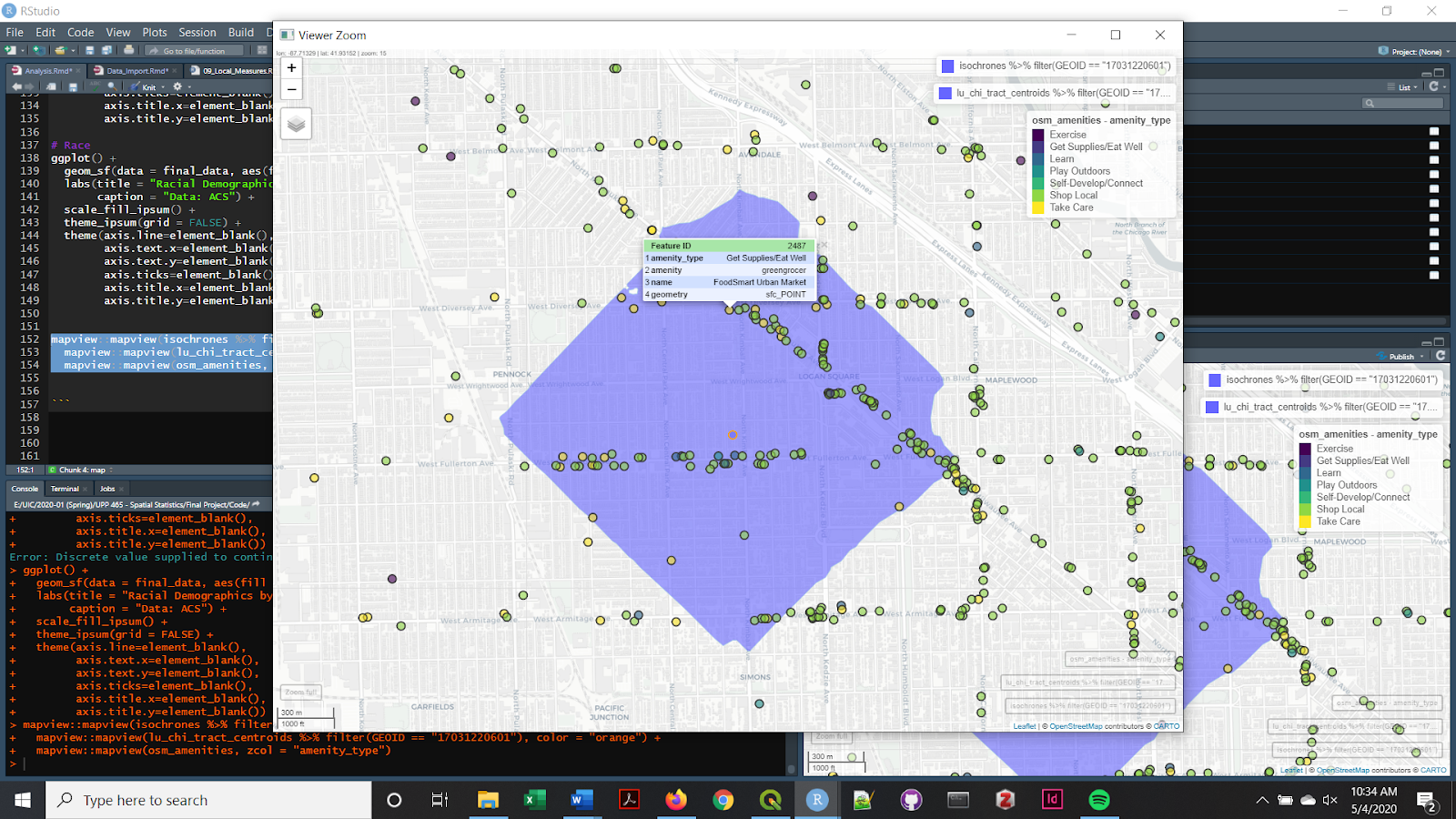


*Get Amenities*

The final step in the data collection and cleaning process required getting detailed data on a wide variety of neighborhood amenities throughout the city. Mayor Hildago’s manifesto calls for focusing on “Learn, Work, Share and Re-Use, Get Supplies, Take the Air, Self-Develop and Connect, Look After Yourself, Get Around, Spend, and Eat Well” (O’Sullivan, 2020). “Work” and “Share and Re-Use” fell outside the scope of this investigation, however, it required some personal judgement to decide which amenities to use from OpenStreetMap. In the end, information on a total of roughly 5,600 features were downloaded using the following logic to determine definitions. Figure 3 presents a spatial example of these data:

* "Learn" → school
* "Self-Develop/Connect" → library, community\_centre, theatre
* "Take Care" → hospital, clinic, pharmacy
* "Exercise" → fitness\_centre, sports\_centre
* "Play Outdoors" → park, playground
* "Shop Local" → marketplace, restaurant, bar, cafe, clothes, department\_store, variety\_store
* "Get Supplies/Eat Well" → supermarket, convenience, greengrocer, bakery, deli

**Figure 3 – OSM Amenities for Tract with GEOID = 17031220601**



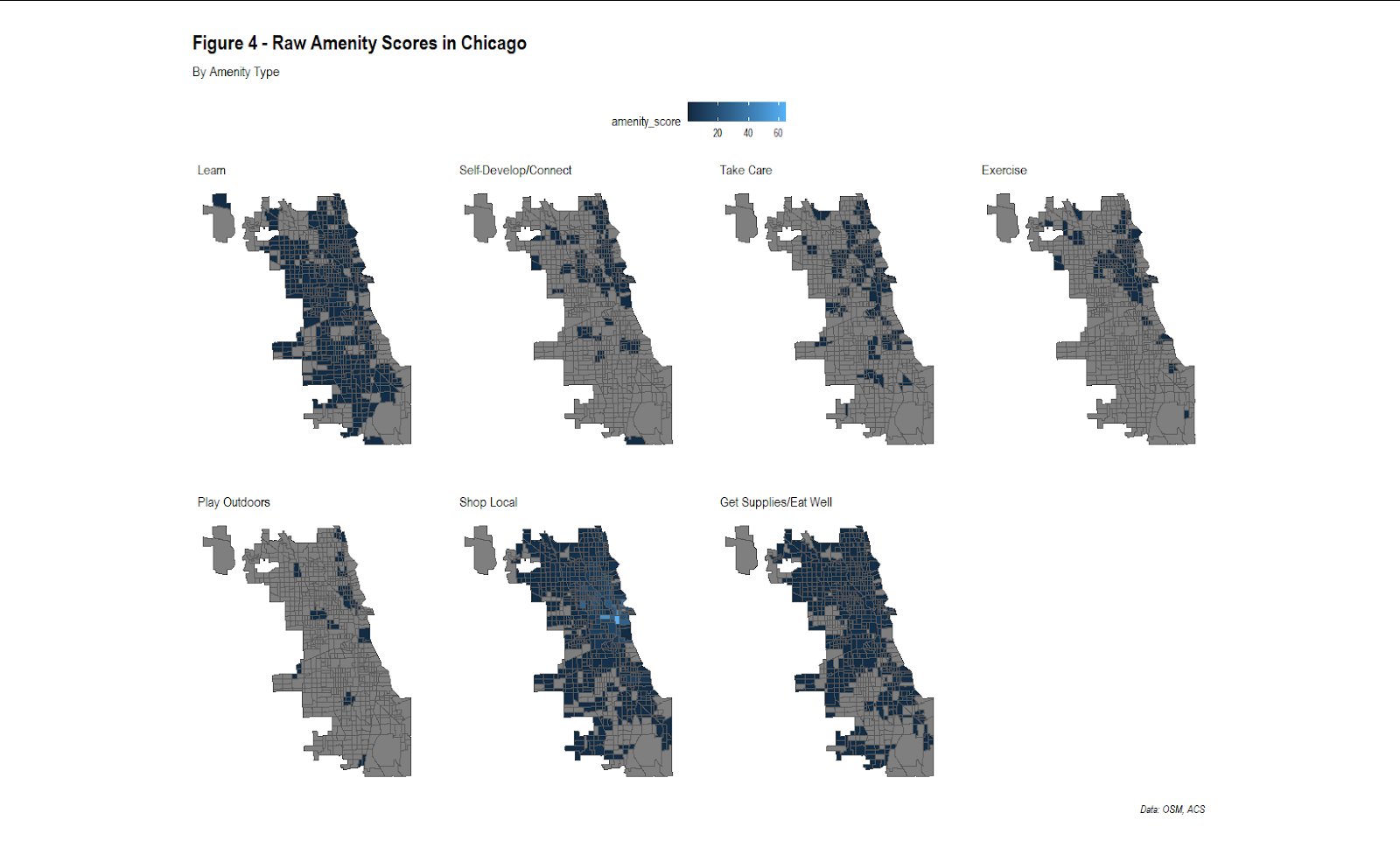
**Results**

Once the number of amenities per amenity type within the isochrone associated with a given tract is known, a simple amenity score is constructed using the following formula:

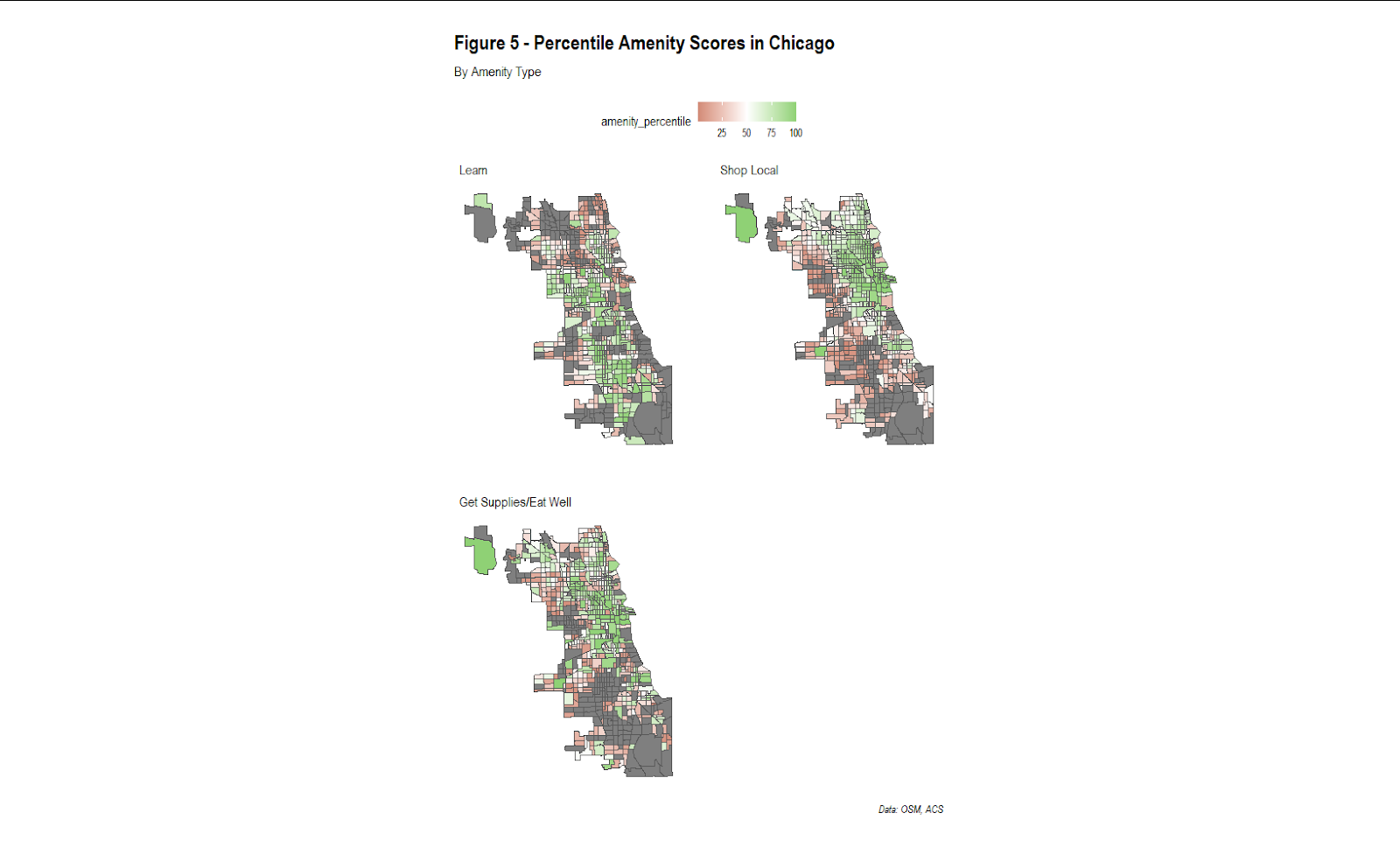
*(# of amenities within isochrone)*

*amenity\_score for tract =*

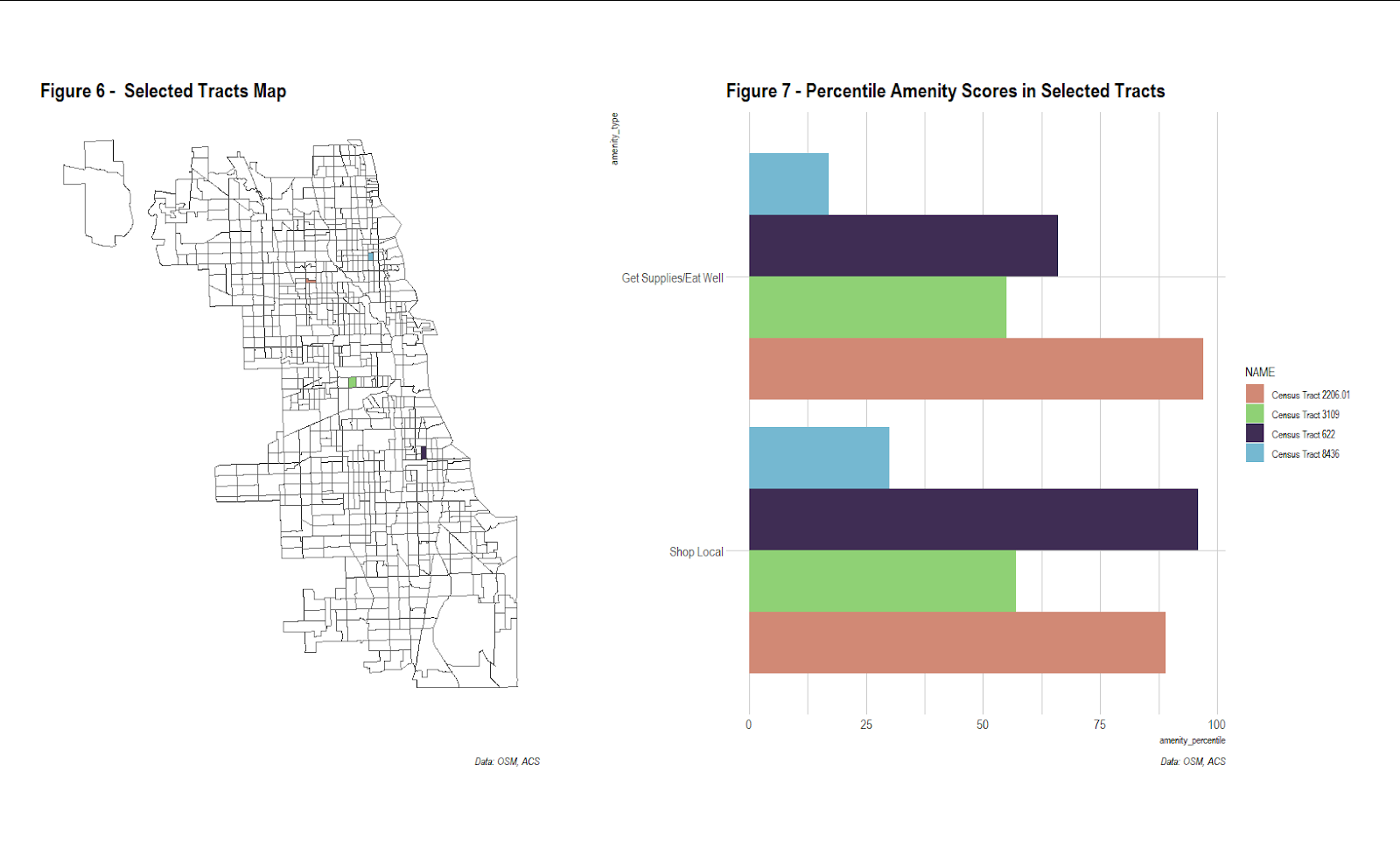
*(total population / area in km^2)*

 Figure 4 below presents the resulting amenity scores by amenity type across Chicago. It is apparent that underlying data for OSM amenities is only complete enough in “Learn”, “Shop Local”, and “Get Supplies/Eat Well” to conduct a citywide analysis, but other categories exhibit a distinct North Side bias, perhaps suggesting that platform users might be whiter and richer.

Of course, these scores are not reflective of any true value, are perhaps most useful when compared among themselves. In Figure 5, the scores are shown as percentiles within each given amenity type. The findings suggests that, at least in terms of raw numbers relative to population density, the near South and West Sides have many schools, but that access to local nightlife and retail is highly concentrated on the North Side and in the West Loop.



**Conclusions and Next Steps**

 Given the lack of data quality and consistency across the amenity types, formalized spatial analysis on neighborhood accessibility is impossible. However, for “Learn”, “Shop Local”, and “Get Supplies/Eat Well” amenities, preliminary Local Moran’s I analysis does suggest the presence of clustering, especially along the North Lakefront, tracking the racial and income divides known to all individuals familiar with the city. Future steps may involve deepening this analysis for a narrow geographic subset given the limitations of the data. Another consideration would be to create an interactive tool using *Shiny* and *Flexdashboard* that allows for a direct comparison between Census Tract percentile scores and a view of each areas underlying amenity data. Figures 6 and 7 demonstrate a theoretical layout for such a tool, with the map powering the bar chart.

Works Cited

City of Portland. (2012). “The Portland Plan: 20-Minute Neighborhoods”. Retrieved from: <https://www.portlandonline.com/portlandplan/index.cfm?c=52256&a=288547>

Cobbs, C. (2020, February 25). “Can Chicago Become a 15 Minute City”. *Streetsblog Chicago*. Retrieved from: <https://chi.streetsblog.org/2020/02/25/can-chicago-become-a-15-minute-city>

Deloitte. (2018). “ImagineSydney – Live”. *Shaping Future Cities*. Retrieved from: <https://www2.deloitte.com/content/dam/Deloitte/au/Documents/about-deloitte/deloitte-au-about-imagine-sydney-live-digital-appendix-230318.pdf>

Kamin, B. (2019, October 25). “In his first in-depth Chicago interview, Lightfoot’s planning chief talks about the Obama Center, the Thompson Center and reviving struggling neighborhoods”. *The Chicago Tribune*. Retrieved from: <https://www.chicagotribune.com/columns/blair-kamin/ct-biz-maurice-cox-interview-kamin-20191025-2gpeguqmzvhjdhxsvxcalu6ubi-story.html>

Lane Council of Governments. (2012). “20-Minute Neighborhood Walkability Analysis for the Eugene-Springfield Metropolitan Area”. Retrieved from: <https://thempo.org/DocumentCenter/View/4521/20MinHood_Slides_MPO_120918_dr?bidId=>

Moore, N. (2019, November 29). “Chicago’s New Planning Chief Has Fresh Eyes For INVEST South/West”. *WBEZ*. Retrieved from <https://www.wbez.org/shows/wbez-news/chicagos-new-planning-chief-has-fresh-eyes-for-invest-southwest/02b2dff1-e2ba-4b87-9a2a-53fdccda4daf>

O’Sullivan, F. (2020, February 18). “Paris Mayor: It's Time for a '15-Minute City'”. *CityLab*. Retrieved from: <https://www.citylab.com/environment/2020/02/paris-election-anne-hidalgo-city-planning-walks-stores-parks/606325/>

Zhang, L., & Pfoser, D. (2019). Using OpenStreetMap point-of-interest data to model urban change—A feasibility study. Retrieved from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6388917/>

Appendix

**Data Import**

---

title: "15-Minute City - Data Import"

author: "Sean Connelly"

date: "`r format(Sys.time(), '%d %B, %Y')`"

output: html\_document

editor\_options:

chunk\_output\_type: console

---

```{r setup, include=FALSE}

# Load libraries

pacman::p\_load(tidyverse, janitor, tidycensus, sf, osmdata, dodgr, here,

extrafont)

#Options, call stored Census API key, load fonts

options(scipen = 1000, stringsAsFactors = F, tigris\_use\_cache = TRUE)

invisible(Sys.getenv("CENSUS\_API\_KEY"))

# Set working directory

setwd(here::here())

# Increase memory limit

memory.limit(16000)

```

# ACS 5-Year Estimates from Census

### Import Data

```{r import ACS}

# List of variables from ACS 5-Year estimates

ref\_vars\_18 <- load\_variables(2018, "acs5", cache = TRUE)

ref\_tables\_18 <- ref\_vars\_18 %>%

mutate(table = str\_extract(name, "^.\*(?=\_)")) %>%

group\_by(table, concept) %>%

summarize(n())

# Grab variables in race, income tables

vars <- ref\_vars\_18 %>%

filter(str\_detect(name, pattern = "(^B02001\_\*)|(^B19001\_\*)"))

# ACS end years

years <- list(2010, 2018)

# ===============

# Chicago reference spatial data

# ===============

# Tracts, Community Areas, Citywide

lu\_chi\_tracts <- st\_read("../Data/Base/Chi\_Tracts\_2010.geojson")

lu\_chi\_commareas <- st\_read("../Data/Base/Chi\_CommAreas.geojson")

lu\_citywide <- st\_read("../Data/Base/Chi\_Boundary.geojson")

# Build link from comm areas to tracts

link\_commareas\_tract <- lu\_chi\_commareas %>%

left\_join(., lu\_chi\_tracts %>% st\_drop\_geometry(), by = "commarea\_n")

# ===============

# Chicago - Tracts

# ===============

# Grab census data for tracts located within Cook County

acs\_raw\_cook <- map(years,

~ get\_acs(

geography = "tract",

variables = vars %>% pull(name),

year = .x,

survey = "acs5",

state = "IL",

county = "Cook",

geometry = FALSE)) %>%

map2(years, ~mutate(.x, id = .y))

# Restrict to City of Chicago tracts

acs\_chi\_tracts <- reduce(acs\_raw\_cook, rbind) %>%

semi\_join(., lu\_chi\_tracts %>% st\_drop\_geometry(), by = "GEOID") %>%

rename("year" = id)

# ===============

# Chicago - Citywide

# ===============

# Grab census data for City of Chicago

acs\_raw\_citywide <- map(years,

~ get\_acs(

geography = "place",

variables = vars %>% pull(name),

year = .x,

survey = "acs5",

state = "IL",

geometry = FALSE)) %>%

map2(years, ~mutate(.x, id = .y))

# Restrict to City of Chicago

acs\_citywide <- reduce(acs\_raw\_citywide, rbind) %>%

filter(NAME == "Chicago city, Illinois") %>%

rename("year" = id)

```

### Clean and Tidy

```{r clean ACS}

# ===============

# Chicago - Tracts

# ===============

# Join variable labels, create table field, remove sub-tables

acs\_chi\_tracts <- left\_join(acs\_chi\_tracts, vars,

by = c("variable" = "name")) %>%

mutate(sheet\_name = gsub( "\_.\*$", "", variable),

label = gsub("!!", "; ", label)) %>%

filter(grepl("\\d$", sheet\_name)) %>%

pivot\_wider(names\_from = year, values\_from = c(estimate, moe)) %>%

mutate(change = `estimate\_2018` - `estimate\_2010`,

pct\_change = change/`estimate\_2010`) %>%

left\_join(.,

link\_commareas\_tract %>%

st\_drop\_geometry() %>%

select(community, GEOID),

by = "GEOID") %>%

select(sheet\_name, concept, community, GEOID, NAME,

variable, label, starts\_with("estimate"), starts\_with("moe"),

change, pct\_change)

# ===============

# Chicago - Citywide

# ===============

# Join variable labels, create table field, remove sub-tables

acs\_citywide <- left\_join(acs\_citywide, vars,

by = c("variable" = "name")) %>%

mutate(sheet\_name = gsub( "\_.\*$", "", variable),

label = gsub("!!", "; ", label)) %>%

filter(grepl("\\d$", sheet\_name)) %>%

pivot\_wider(names\_from = year, values\_from = c(estimate, moe)) %>%

mutate(change = `estimate\_2018` - `estimate\_2010`,

pct\_change = change/`estimate\_2010`) %>%

select(sheet\_name, concept, GEOID, NAME,

variable, label, starts\_with("estimate"), starts\_with("moe"),

change, pct\_change)

# ===============

# Summarize to Community Areas

# ===============

# Join comm area spatial to tract-level ACS data, summarize up to comm areas

acs\_chi\_commareas <- acs\_chi\_tracts %>%

select(-GEOID, -NAME, -change, -pct\_change) %>%

group\_by(community, sheet\_name, concept, variable, label) %>%

summarize(estimate\_2010 = sum(estimate\_2010),

estimate\_2018 = sum(estimate\_2018),

moe\_2010 = moe\_sum(moe\_2010, estimate\_2010),

moe\_2018 = moe\_sum(moe\_2018, estimate\_2018)) %>%

ungroup() %>%

mutate(change = `estimate\_2018` - `estimate\_2010`,

pct\_change = change/`estimate\_2010`) %>%

select(sheet\_name, concept, community,

variable, label, starts\_with("estimate"), starts\_with("moe"),

change, pct\_change)

```

### Export

```{r export clean ACS data}

# Illinois state plane

lu\_citywide <- lu\_citywide %>% st\_transform(crs = 26971)

lu\_chi\_commareas <- lu\_chi\_tracts %>% st\_transform(crs = 26971)

lu\_chi\_tracts <- lu\_chi\_tracts %>% st\_transform(crs = 26971)

# Tract centroids

lu\_chi\_tract\_centroids <- lu\_chi\_tracts %>%

st\_centroid()

# Write to shapefile

st\_write(lu\_chi\_tract\_centroids, "../Data/Census/Chi\_Tract\_Centroids.shp",

delete\_dsn = TRUE)

st\_write(lu\_chi\_tracts, "../Data/Census/Chi\_Tracts.shp", delete\_dsn = TRUE)

st\_write(lu\_chi\_commareas, "../Data/Census/Chi\_CommAreas.shp", delete\_dsn =

TRUE)

st\_write(lu\_citywide, "../Data/Census/Chi\_Citywide.shp", delete\_dsn = TRUE)

# Write to attributes to CSV

write\_csv(acs\_chi\_tracts, "../Data/Census/Chi\_Tracts\_Attributes.csv")

write\_csv(acs\_chi\_commareas, "../Data/Census/Chi\_CommAreas\_Attributes.csv")

write\_csv(acs\_citywide, "../Data/Census/Chi\_Citywide\_Attributes.csv")

# Note: worth removing data stored in memory at this step

rm(list = ls())

```

# Street Network

### Import Data

```{r import street network}

# Chicago street network

streets\_raw <- opq("Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "highway") %>%

osmdata\_sf() %>%

pluck("osm\_lines") %>%

select(osm\_id, highway, name, lanes, maxspeed, geometry) %>%

filter(is.na(name) | (!str\_detect(name, "Expressway"))) %>%

mutate(highway = "pedestrian")

# Transform to Illinois State Plane East 1201 Feet

streets\_raw <- st\_transform(streets\_raw, crs = 26971)

# Weight by mode, remove streets\_raw

graph <- weight\_streetnet(streets\_raw, wt\_profile = "foot")

rm(streets\_raw)

```

### Clean and Tidy

```{r clean street network}

# Find nodes closest to centroid of tracts, need temporary sf object

graph\_sf <- st\_as\_sf(graph, coords = c("from\_lon", "from\_lat"), crs = 26971)

lu\_chi\_tract\_centroids <- st\_read("../Data/Census/Chi\_Tract\_Centroids.shp",

crs = 26971)

nodes\_sf <- graph\_sf %>%

filter(row\_number() %in% st\_nearest\_feature(lu\_chi\_tract\_centroids,

graph\_sf))

# Join Census tract info

tract\_nums <- lu\_chi\_tract\_centroids %>%

st\_drop\_geometry() %>%

mutate(match\_node = st\_nearest\_feature(nodes\_sf, lu\_chi\_tract\_centroids))

%>%

select(match\_node, GEOID)

nodes\_sf <- nodes\_sf %>%

mutate(match\_node = row\_number()) %>%

left\_join(tract\_nums, by = "match\_node")

```

### Export

```{r export street network}

# Export for QGIS calculations

st\_write(streets\_raw, "../Data/Isochrones/streets\_raw.shp", delete\_dsn = TRUE)

st\_write(nodes\_sf, "../Data/Isochrones/nodes\_sf.shp", delete\_dsn = TRUE)

# Note: worth removing data stored in memory at this step

rm(list = ls())

```

# Isochrones

### Import Data

```{r import amenities}

# Import nodes

nodes\_sf <- st\_read("../Data/Isochrones/nodes\_sf.shp", crs = 26971) %>%

arrange(mtch\_nd)

# Import QGIS Files

isochrones <- list.files("../Data/Isochrones/Individual",

pattern = "\\.shp$",

full.names = TRUE) %>%

set\_names %>%

map\_df(~st\_read(.x, crs = 26971), .id = "file\_name") %>%

as\_tibble() %>%

st\_as\_sf(., crs = 26971)

# Rename and clean

isochrones <- isochrones %>%

select(-id) %>%

mutate(cost\_level = cost\_level / 60,

node\_id = as.numeric(str\_extract(file\_name, "\\d+")) + 1)

# Join node and tract info

isochrones <- isochrones %>%

left\_join(.,

nodes\_sf %>%

st\_drop\_geometry() %>%

select(mtch\_nd, GEOID,commr\_n),

by = c("node\_id" = "mtch\_nd")) %>%

select(GEOID, cost\_level, geometry)

```

### Export

```{r export isochrones}

# Write to shapefile

st\_write(isochrones, "../Data/Isochrones/isochrones.shp", delete\_dsn = TRUE)

# Note: worth removing data stored in memory at this step

rm(list = ls())

```

# OSM Amenities

### Import Data

```{r import amenities}

# From CityLab (https://www.citylab.com/environment/2020/02/paris-electionanne-

hidalgo-city-planning-walks-stores-parks/606325/)

# Paris en Commun’s 15-minute city concept. From the top, clockwise, the

headings read: Learn, Work, Share and Re-Use, Get Supplies, Take the Air,

Self-Develop and Connect, Look After Yourself, Get Around, Spend, and Eat

Well. (Paris en Commun)

# Grab data from OpenStreetMap (https://wiki.openstreetmap.org/wiki/

Map\_Features#Amenity)

# Work (N/A)

# Share and Re-use (N/A)

# Get Around (N/A)

# Learn

osm\_temp\_school <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "school") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "school")

# Self-Develop and Connect

osm\_temp\_library <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "library") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "library")

osm\_temp\_community\_centre <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "community\_centre") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "community\_centre")

osm\_temp\_theatre <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "theatre") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "theatre")

# Look After Yourself

osm\_temp\_hospital <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "hospital") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "hospital")

osm\_temp\_clinic <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "clinic") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "clinic")

osm\_temp\_pharmacy <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "pharmacy") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "pharmacy")

# Take the Air

osm\_temp\_park <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "leisure", value = "park") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = leisure, name) %>%

filter(amenity == "park")

osm\_temp\_playground <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "leisure", value = "playground") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = leisure, name) %>%

filter(amenity == "playground")

# Exercise (Look After Yourself Pt 2)

osm\_temp\_fitness\_centre <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "leisure", value = "fitness\_centre") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = leisure, name) %>%

filter(amenity == "fitness\_centre")

osm\_temp\_sports\_centre <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "leisure", value = "sports\_centre") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = leisure, name) %>%

filter(amenity == "sports\_centre")

# Spend

osm\_temp\_marketplace <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "marketplace") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "marketplace")

osm\_temp\_restaurant <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "restaurant") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "restaurant")

osm\_temp\_bar <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "bar") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "bar")

osm\_temp\_cafe <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "amenity", value = "cafe") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select(amenity, name) %>%

filter(amenity == "cafe")

osm\_temp\_clothes <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "shop", value = "clothes") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = shop, name) %>%

filter(amenity == "clothes")

osm\_temp\_department\_store <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "shop", value = "department\_store") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = shop, name) %>%

filter(amenity == "department\_store")

osm\_temp\_variety\_store <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "shop", value = "variety\_store") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = shop, name) %>%

filter(amenity == "variety\_store")

# get Supplies/Eat Well

osm\_temp\_supermarket <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "shop", value = "supermarket") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = shop, name) %>%

filter(amenity == "supermarket")

osm\_temp\_convenience <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "shop", value = "convenience") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = shop, name) %>%

filter(amenity == "convenience")

osm\_temp\_greengrocer <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "shop", value = "greengrocer") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = shop, name) %>%

filter(amenity == "greengrocer")

osm\_temp\_bakery <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "shop", value = "bakery") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = shop, name) %>%

filter(amenity == "bakery")

osm\_temp\_deli <- opq(bbox = "Chicago, Illinois, USA") %>%

add\_osm\_feature(key = "shop", value = "deli") %>%

osmdata\_sf() %>%

unique\_osmdata() %>%

pluck("osm\_points") %>%

select("amenity" = shop, name) %>%

filter(amenity == "deli")

```

### Clean and Tidy

````{r clean osm amenities}

# Bind together, fix projection

osm\_amenities <- mget(ls(pattern = "^osm\_temp\_")) %>%

bind\_rows() %>%

as\_tibble() %>%

st\_as\_sf(., crs = 4326) %>%

st\_transform(crs = 26971)

rm(list = (ls(pattern = "^osm\_temp\_")))

# Create groupings

osm\_amenities <- osm\_amenities %>%

mutate(amenity\_type = case\_when(amenity == "school" ~

"Learn",

amenity %in% c("library",

"community\_centre", "theatre") ~

"Self-Develop/Connect",

amenity %in% c("hospital", "clinic",

"pharmacy") ~

"Take Care",

amenity %in% c("fitness\_centre",

"sports\_centre") ~

"Exercise",

amenity %in% c("park", "playground") ~

"Play Outdoors",

amenity %in% c("marketplace", "restaurant",

"bar", "cafe",

"clothes",

"department\_store", "variety\_store") ~

"Shop Local",

amenity %in% c("supermarket", "convenience",

"greengrocer",

"bakery", "deli") ~

"Get Supplies/Eat Well") %>%

factor(., levels = c("Learn", "Self-Develop/Connect", "Take Care",

"Exercise",

"Play Outdoors", "Shop Local", "Get Supplies/

Eat Well"))) %>%

select(amenity\_type, everything())

```

### Export

```{r export osm amenities}

# Write to shapefile

st\_write(osm\_amenities, "../Data/OSM/amenities.shp", delete\_dsn = TRUE)

rm(list = ls())

```

**Analysis**

---

title: "15-Minute City - Analysis"

author: "Sean Connelly"

date: "`r format(Sys.time(), '%d %B, %Y')`"

output: html\_document

editor\_options:

chunk\_output\_type: console

---

```{r setup, include=FALSE}

# Load libraries

pacman::p\_load(tidyverse, janitor, tidycensus, sf, tmap, patchwork,

hrbrthemes, here, extrafont)

#Options, call stored Census API key, load fonts

options(scipen = 1000, stringsAsFactors = F, tigris\_use\_cache = TRUE)

# Set working directory

setwd(here::here())

```

# Import Data

```{r import data}

# ACS

lu\_chi\_tracts <- st\_read("../Data/Census/Chi\_Tracts.shp", crs = 26971)

lu\_chi\_tract\_centroids <- st\_read("../Data/Census/Chi\_Tract\_Centroids.shp",

crs = 26971)

lu\_chi\_commareas <- st\_read("../Data/Census/Chi\_CommAreas.shp", crs = 26971)

acs\_chi\_tracts <- read\_csv("../Data/Census/Chi\_Tracts\_Attributes.csv")

# Isochrones

isochrones <- st\_read("../Data/Isochrones/isochrones.shp", crs = 26971) %>%

filter(cost\_level == 15)

# OSM

osm\_amenities <- st\_read("../Data/OSM/amenities.shp", crs = 26971) %>%

rename("amenity\_type" = amnty\_t) %>%

mutate(amenity\_type = factor(amenity\_type,

levels = c("Learn", "Self-Develop/Connect",

"Take Care", "Exercise",

"Play Outdoors", "Shop Local", "Get

Supplies/Eat Well")))

```

### Create Measures

````{r clean osm amenities}

# Tract Area

lu\_chi\_tracts <- lu\_chi\_tracts %>%

mutate(area = st\_area(lu\_chi\_tracts))

# Race data

tract\_race <- acs\_chi\_tracts %>%

filter(str\_detect(variable, "^B02001\_00(1|2|3|4|5|6|7|8)")) %>%

select(GEOID, NAME, label, estimate\_2018) %>%

mutate(GEOID = as.character(GEOID),

label = stringi::stri\_extract\_last\_regex(label, "[^;]+") %>%

str\_trim()) %>%

mutate(label = case\_when(label == "White alone" ~

"White",

label == "Black or African American alone" ~

"Black",

label == "American Indian and Alaska Native alone"

~

"AIAN",

label == "Asian alone" ~

"Asian",

label == "Native Hawaiian and Other Pacific

Islander alone" ~

"NHOPI",

label == "Two or more races" ~

"Multiple",

label == "Some other race alone" ~

"Other",

TRUE ~ label)) %>%

pivot\_wider(names\_from = label, values\_from = estimate\_2018) %>%

mutate(nbhd\_type = case\_when(Black / Total > 0.5 ~ "Majority Black",

White / Total > 0.5 ~ "Majority White",

TRUE ~ "Other"))

# Amenities

osm\_by\_iso\_raw <- st\_join(isochrones, osm\_amenities) %>%

st\_drop\_geometry()

osm\_by\_iso\_counts <- osm\_by\_iso\_raw %>%

group\_by(GEOID, amenity\_type) %>%

summarize(amenity\_n = n())

# Final

final\_data <- lu\_chi\_tracts %>%

st\_drop\_geometry() %>%

left\_join(tract\_race, by = "GEOID") %>%

left\_join(osm\_by\_iso\_counts, by = "GEOID") %>%

complete(GEOID, amenity\_type) %>%

left\_join(lu\_chi\_tracts %>% dplyr::select(GEOID), by = "GEOID") %>%

st\_as\_sf(., crs = 26971) %>%

filter(!is.na(amenity\_type)) %>%

mutate(area = units::drop\_units(area) / 1000,

amenity\_score = amenity\_n / (Total / area)) %>%

group\_by(amenity\_type) %>%

mutate(amenity\_percentile = ntile(amenity\_score, 100)) %>%

ungroup()

```

### Maps

```{r maps}

# Figure 1- Race

lu\_chi\_tracts %>%

left\_join(tract\_race, by = "GEOID") %>%

ggplot() +

geom\_sf(aes(fill = nbhd\_type)) +

labs(title = "Figure 1 - Racial Demographics by Census Tract",

caption = "Data: ACS") +

scale\_fill\_ipsum() +

theme\_ipsum(grid = FALSE) +

theme(axis.line = element\_blank(),

axis.text.x = element\_blank(),

axis.text.y = element\_blank(),

axis.ticks = element\_blank(),

axis.title.x = element\_blank(),

axis.title.y = element\_blank())

tract\_race %>% tabyl(nbhd\_type) %>% adorn\_totals("row")

# Figure 2 - Build Isochrones (QGIS)

# Figure 3 - OSM Amenities (QGIS)

# Figure 4 - Amenity Scores

ggplot() +

geom\_sf(data = final\_data, aes(fill = amenity\_score)) +

facet\_wrap(~ amenity\_type, nrow = 2) +

labs(title = "Figure 4 - Raw Amenity Scores in Chicago",

subtitle = "By Amenity Type",

caption = "Data: OSM, ACS") +

theme\_ipsum(grid = FALSE) +

theme(legend.position = "top",

axis.line = element\_blank(),

axis.text.x = element\_blank(),

axis.text.y = element\_blank(),

axis.ticks = element\_blank(),

axis.title.x = element\_blank(),

axis.title.y = element\_blank())

# Percentiles

ggplot() +

geom\_sf(data = final\_data %>%

filter(amenity\_type %in% c("Learn", "Shop Local", "Get Supplies/

Eat Well")),

aes(fill = amenity\_percentile)) +

facet\_wrap(~ amenity\_type, nrow = 2) +

labs(title = "Figure 5 - Percentile Amenity Scores in Chicago",

subtitle = "By Amenity Type",

caption = "Data: OSM, ACS") +

scale\_fill\_gradient2(low = "#D18975",

high = "#8FD175",

midpoint = 50) +

theme\_ipsum(grid = FALSE) +

theme(legend.position = "top",

axis.line = element\_blank(),

axis.text.x = element\_blank(),

axis.text.y = element\_blank(),

axis.ticks=element\_blank(),

axis.title.x = element\_blank(),

axis.title.y = element\_blank())

# # Crosstab

# final\_data %>%

# st\_drop\_geometry() %>%

# mutate(amenity\_quant = ntile(amenity\_score, 5)) %>%

# tabyl(amenity\_quant, nbhd\_type) %>%

# adorn\_totals("row") %>%

# adorn\_percentages("col") %>%

# adorn\_pct\_formatting() %>%

# adorn\_ns() %>%

# adorn\_title("combined") %>% knitr::kable()

# Map

# Example Census Tracts

plot\_map <- ggplot() +

geom\_sf(data = final\_data, fill = "white", alpha = 0.2) +

geom\_sf(data = final\_data %>% filter(GEOID == "17031220601"),

fill = "#D18975") +

geom\_sf(data = final\_data %>% filter(GEOID == "17031310900"),

fill = "#8FD175") +

geom\_sf(data = final\_data %>% filter(GEOID == "17031062200"),

fill = "#75B8D1") +

geom\_sf(data = final\_data %>% filter(GEOID == "17031843600"),

fill = "#3F2D54") +

labs(title = "Figure 6 - Selected Tracts Map",

caption = "Data: OSM, ACS") +

theme\_ipsum(grid = FALSE) +

theme(legend.position = "top",

axis.line = element\_blank(),

axis.text.x = element\_blank(),

axis.text.y = element\_blank(),

axis.ticks = element\_blank(),

axis.title.x = element\_blank(),

axis.title.y = element\_blank())

# Plot of scores

plot\_scores <- ggplot(data = final\_data %>%

filter(GEOID %in% c("17031220601", "17031310900",

"17031843600", "17031062200"),

amenity\_type %in% c("Get Supplies/Eat Well",

"Shop Local")) %>%

mutate(NAME = str\_remove(NAME, ", Cook County, Illinois")),

aes(amenity\_type, amenity\_percentile, fill = NAME)) +

geom\_col(position = "dodge") +

coord\_flip() +

labs(title = "Figure 7 - Percentile Amenity Scores in Selected Tracts",

caption = "Data: OSM, ACS") +

scale\_fill\_ipsum() +

theme\_ipsum() +

theme(legend.position = "right")

# Together

dev.off()

plot\_map + plot\_scores

```