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.

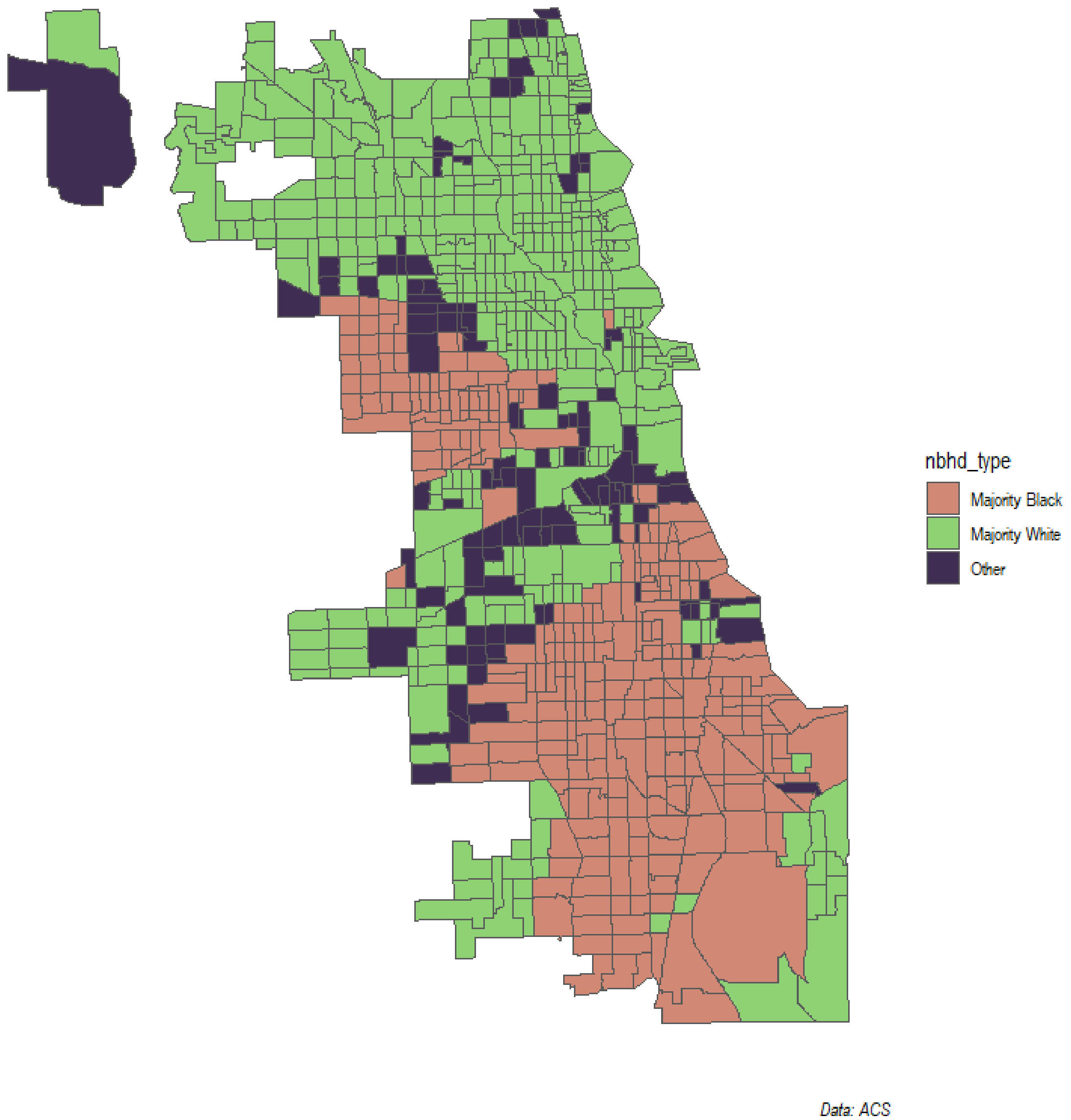
Methodology

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.

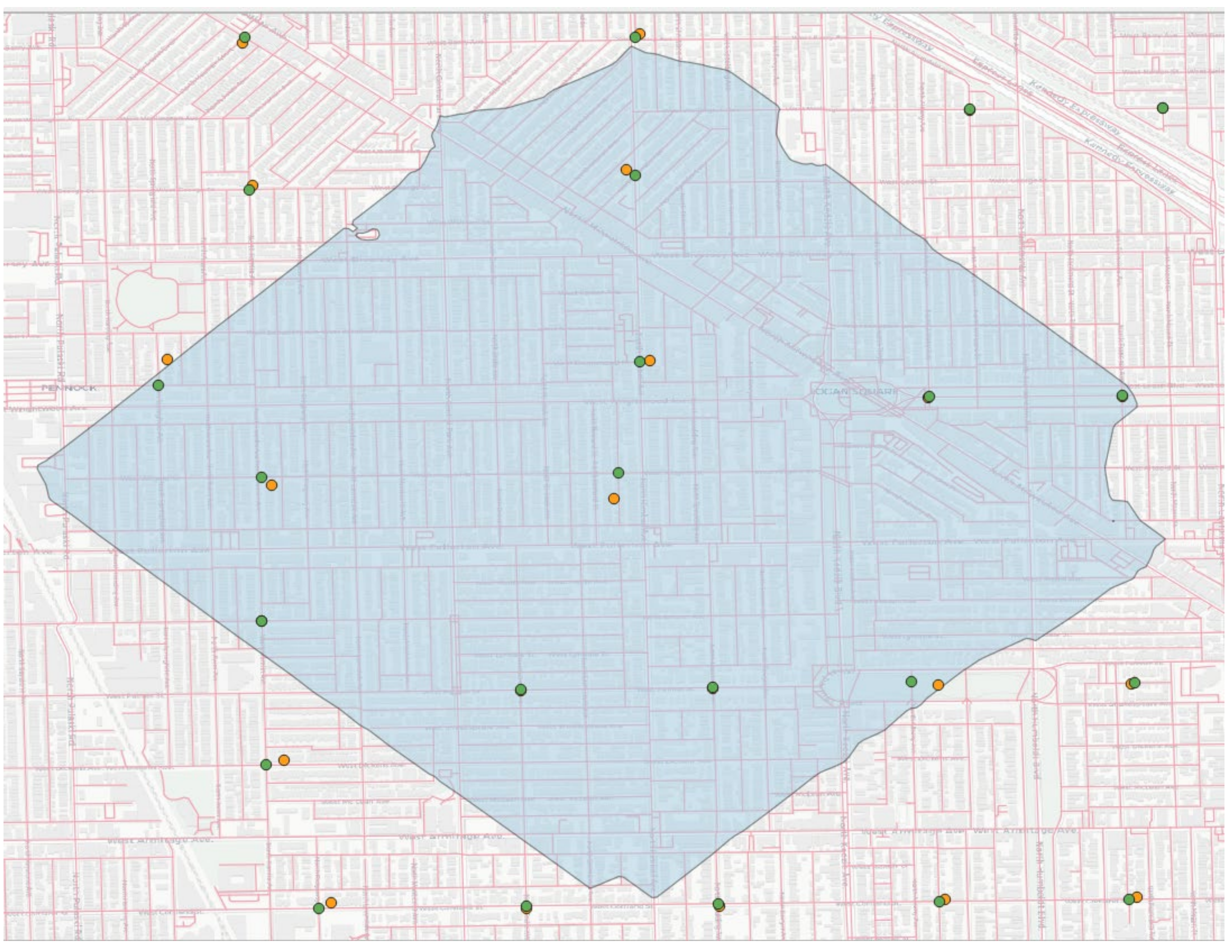
Figure 1 - Racial Demographics by Census Tract



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

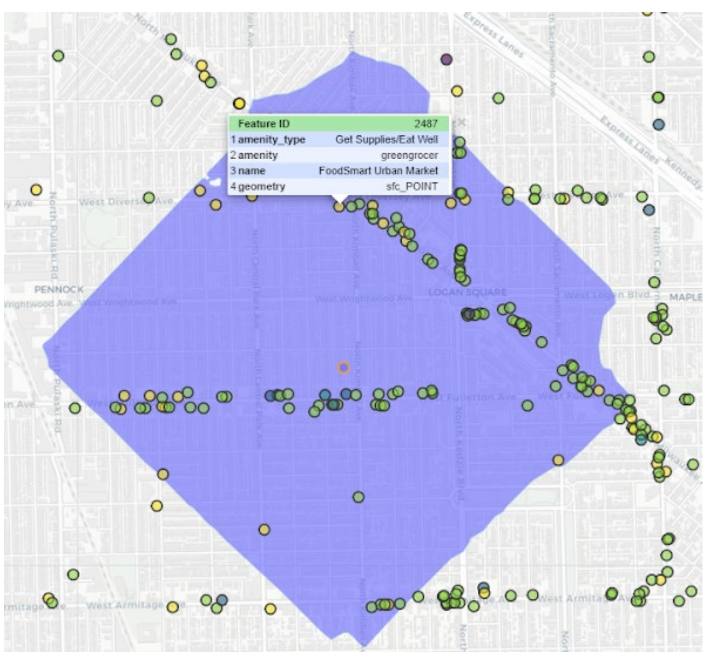


Get Amenities

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

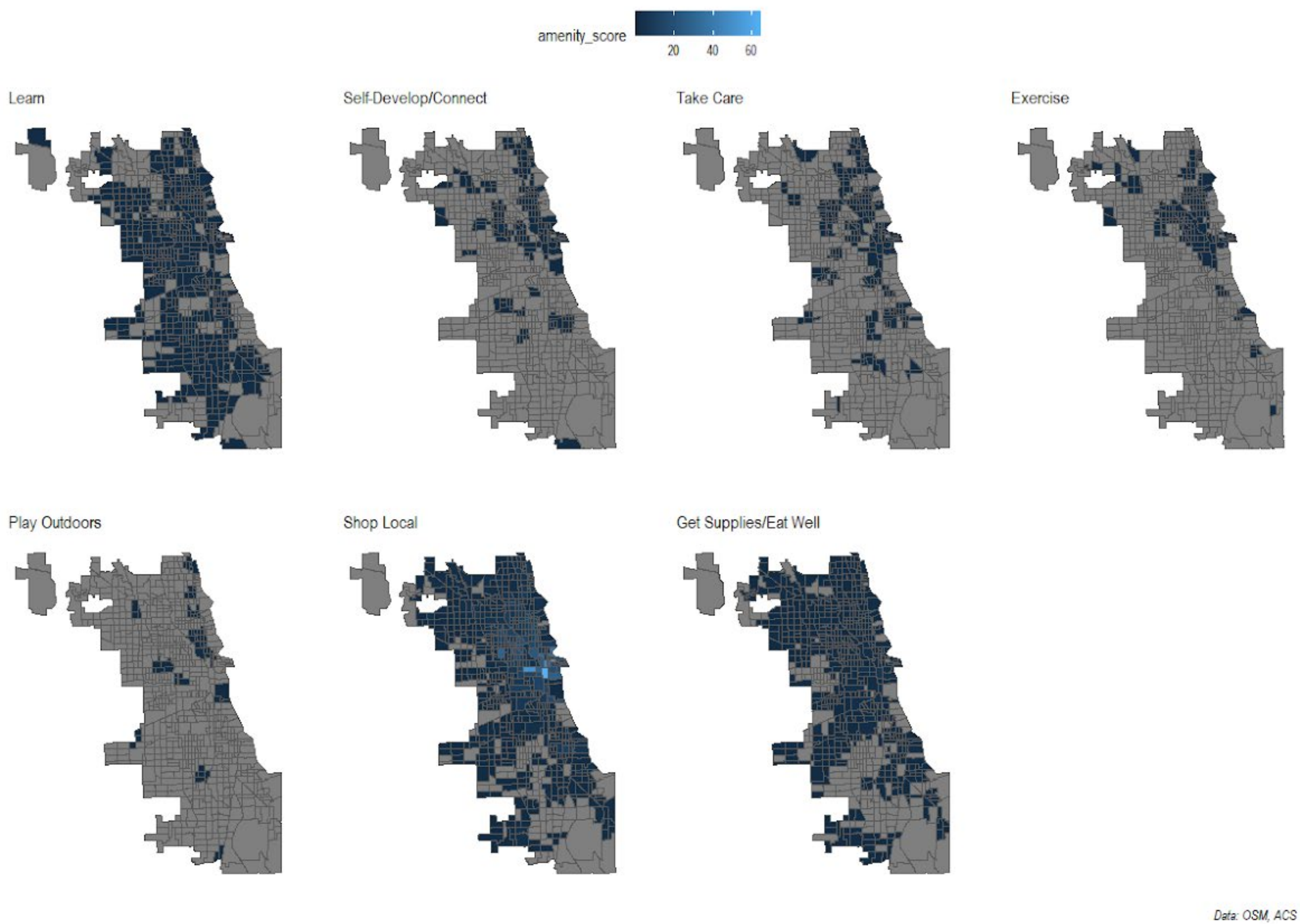
Once the number of amenities per amenity type within the isochrone associated with a given tract is known, a simple amenity score is constructed using the following formula:

$$amenity\_score \text{ for tract} = \frac{(\text{\# of amenities within isochrone})}{(\text{total population / area in km}^2)}$$

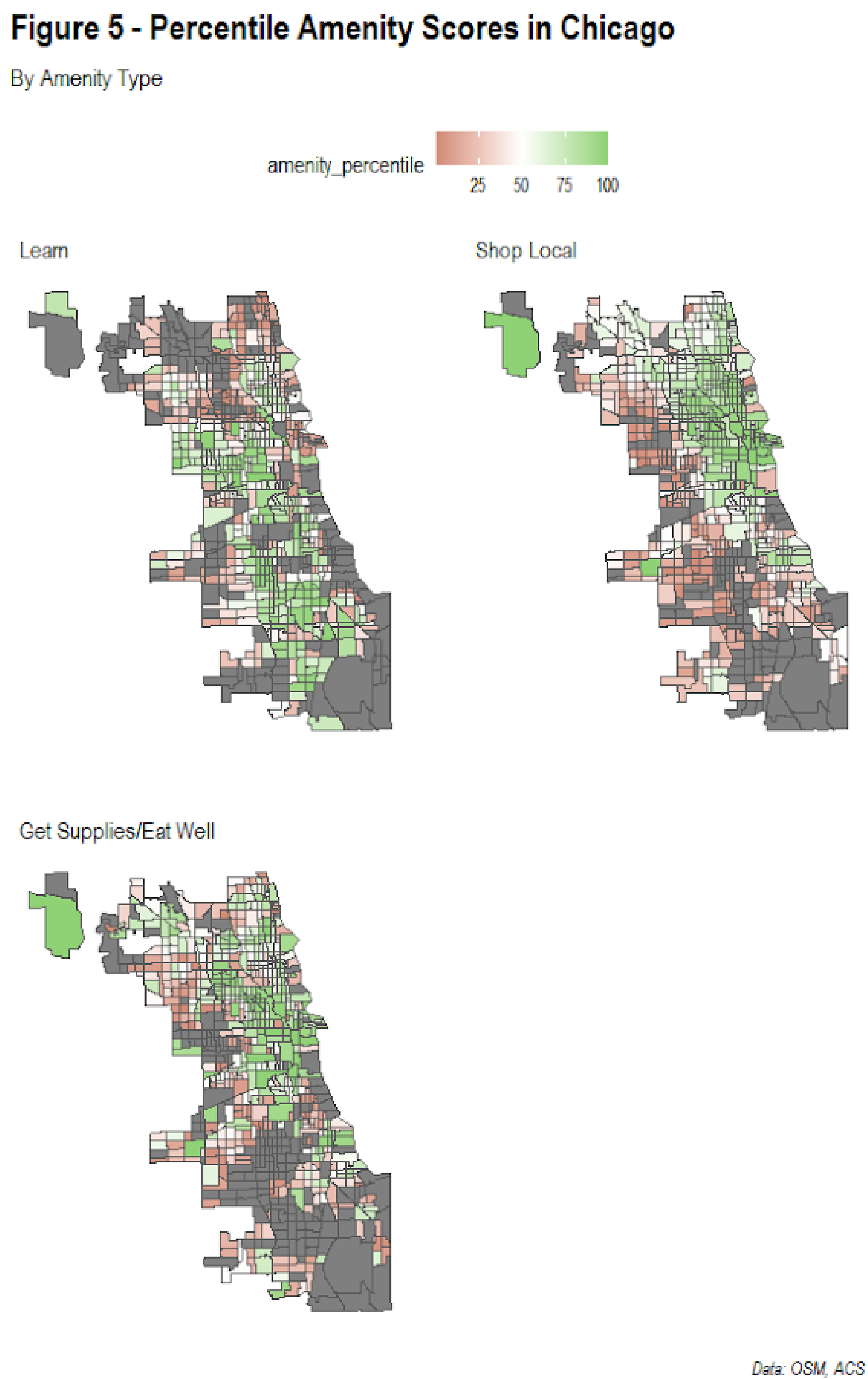
Figure 4 below presents the resulting amenity scores by amenity type across Chicago. It is apparent that underlying data for OSM amenities is only complete enough in "Learn", "Shop Local", and "Get Supplies/Eat Well" to conduct a citywide analysis, but other categories exhibit a distinct North Side bias,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

Figure 4 - Raw Amenity Scores in Chicago

By Amenity Type



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.

Figure 6 - Selected Tracts Map

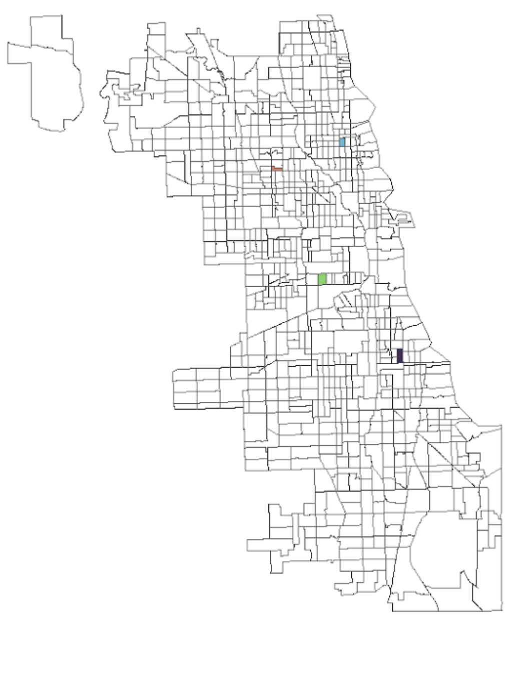
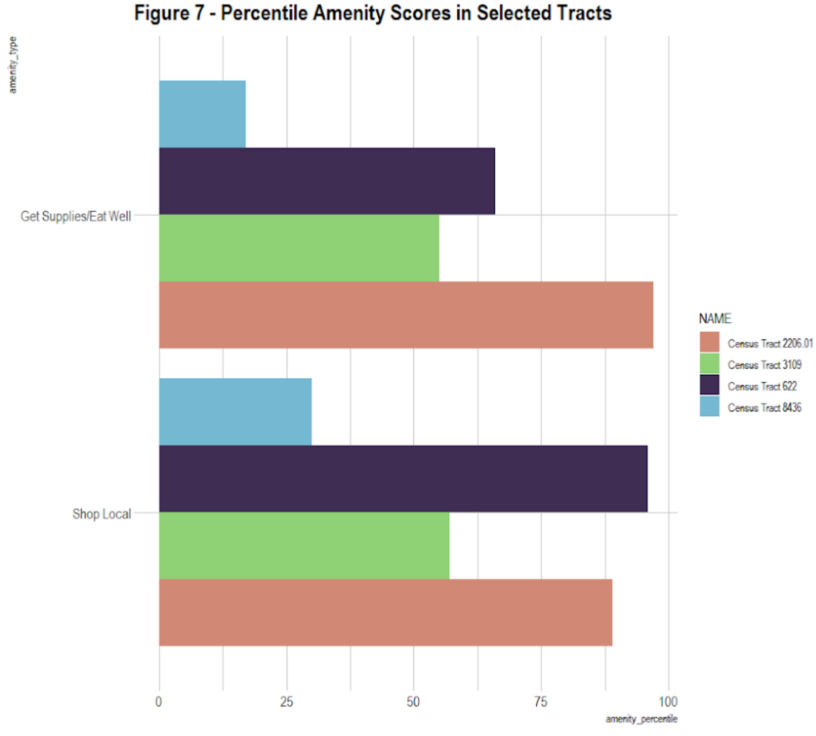


Figure 7 - Percentile Amenity Scores in Selected Tracts



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
```