**Methodological Note**

This file contains supplementary information for the following paper:

**Banks, alternative institutions, and the spatial-temporal ecology of racial inequality**

Mario L. Small, Armin Akhavan, Mo Torres, and Qi Wang

Corresponding Author: Mario L. Small (mariosmall@fas.harvard.edu)

**1. Using Google Maps and Google Places API to find AFIs**

In this section, we describe in greater detail the process through which we acquired data on conventional banks and alternative financial institutions (AFIs).

We used Google Maps and Google Places API. Google Maps collects large amounts of data on many kinds of commercial and non-commercial establishments, including conventional banks and alternative financial institutions, such as check cashing places, payday lenders, etc. Each of these establishments in Google Maps’ database is stored as an “object” with a set of attributes. Google’s Places API “is a service that returns information about places using HTTP requests. Places are defined within this API as establishments, geographic locations, or prominent points of interest.”[[1]](#footnote-1) Through Google Places API, one can access the Google Maps data directly. The data returned from the Google APIs are stored as json “objects,” which we collected and then parsed into our own database for original analysis.

The current paper was part of a larger study for which we collected data on a large number of establishments. We collected establishment data using the Google Places API from December 2016 to January 2017 in the top 20 cities in the U.S. We split the area of each city into grids and used the “nearby” function of the Places API to collect any places in every grid. If a search returned more than 60 (i.e., the maximum numbers of places that can be returned) places, we split the grid into four equal smaller grids and searched again in each of the sub-grids. The algorithm stopped whenever a search returned less than 60 places and moved on to the next grid. The codes and algorithms can be found here: [https://github.com/urbaninformaticsandresiliencelab/gmaps\_scraper](about:blank).[[2]](#footnote-2) All place data collected from a city were stored in a combined .json file.

Of particular importance to our analysis was the place attribute “type,” which describes the function of the establishment (see [https://developers.google.com/places/supported\_types](about:blank)). On Google Places API, conventional banks are assigned the “type” *bank*. By accessing the Google Maps database through the Places API, we were able to simply and efficiently retrieve data that identified conventional banks in the twenty major cities in our analysis. However, not all Google Maps data are available through the Places API; some must be retrieved using a web browser. For example, searching for an establishment on Google Maps using a web browser or mobile phone returns data such as hours of operation that are not retrievable from the Places API. Similarly, while banks are easily accessible through the Places API, alternative financial institutions are not, because there is currently no Places API “type” for AFIs.

Still, AFIs are indeed readily identifiable through a simple Google Maps search using a web browser. Doing so requires deciding on search terms. For example, while there is no Places API “type” for “check cashing establishment,” a browser search on Google Maps for the term “check cashing” retrieves the data from the same database used by the Places API, and lists the full results, including the “name,” “rating,” and “vicinity” attributes from each place “object.” In fact, the results also include a line indicating that the establishment is of the “Check cashing service” category (see below). This establishment category as indicated on the web browser is not one of the Places API listed “types.”

A screenshot of a cell phone

Description automatically generated

On a browser, it is possible to find AFIs under different searches, and a particular AFI may be listed under different categories in different searches. For example, since many payday lenders cash checks, the same establishment may be listed as a “Loan agency” in one search and as a “Check cashing service” in another. However, some categories are broader than others. While payday lenders are rather consistently categorized under check cashing services, the reverse is not necessarily the case, as many places where one could cash a check for a fee do not offer loans.

We used the term “check cashing” to identify check cashing service categories, which is appropriate given the ubiquity of cashing checks as a basic service across a large swath of different AFIs, including not only check cashing establishments but also payday lenders and currency exchanges. Indeed, in some cities where payday lenders are not legally allowed, alternative financial institutions are simply known as “check cashing stores.” While our approach is appropriately comprehensive, it may nonetheless exclude some alternative institutions, such as pawn shops that do not cash checks or are otherwise not classified as doing so. Thus, our process may undercount alternative financial services, particularly more obscure ones. If so, then unless institutions such as pawnshops are especially likely in predominantly white neighborhoods, our results would likely understate the extent of racial inequality.

To create an intermediate dataset of AFIs, we first used Google Maps to identify all available establishments marked “Check cashing service.” We then matched these establishments to the “name” for each location in the Places API, and extracted the place “objects” and all related attributes. This process provided AFI data that are as rich as the data we had collected on conventional banks. To scrape the AFI data, we programmed a Python script to open a Firefox browser, navigate to the Google Maps URL, click on the search box, type in “check cashing,” and click on the search button. From there, we processed through the results, storing the names of all places categorized as “Check cashing service.” The script then matched the names to the place “object” from our dataset built from the Places API.

We experimented with four ways to collect these data with Google Maps on a web browser. Using http://www.google.com/maps/ as the base URL, we tried each of the following:

1. Using coordinates, in latitude/longitude format, and passing the coordinates to the search end point with a search query. For example, to access Chicago: http://www.google.com/maps/search/<search query, e.g. check cashing>/@40.7274826,-74.0902511,11z
2. Using the place endpoint and navigating to city, e.g., Chicago, using the URI of `https://www.google.com/maps/place/Chicago,+IL/` and then clicking on the element for NEARBY search `section-action-button-icon maps-sprite-pane-action-ic-searchnearby` and search for `check cashing`
3. Searching for the place with the search endpoint, meaning first using, e.g., Chicago to search for Chicago, which returns the place endpoint, and continuing with the steps above.
4. Combining the two searches and searching, e.g., for `Chicago,+IL+check+cashing`

The script we settled on uses the fourth method to search. It goes through the list of results, paginating and matching names between the browser-based Google Maps and the Places API. This process, for Chicago alone, returned 254 places we then added to our database. We then made additional rounds of browser scraping with different zooming scales, and looked at the rate of finding additional places as a determining factor to stop the operations. We were able to scrape the rest of the places that the first round did not return by making a loop/iterator for a zoom/drag operation with randomized values for zoom levels and drag directions, ensuring the completeness of our data.

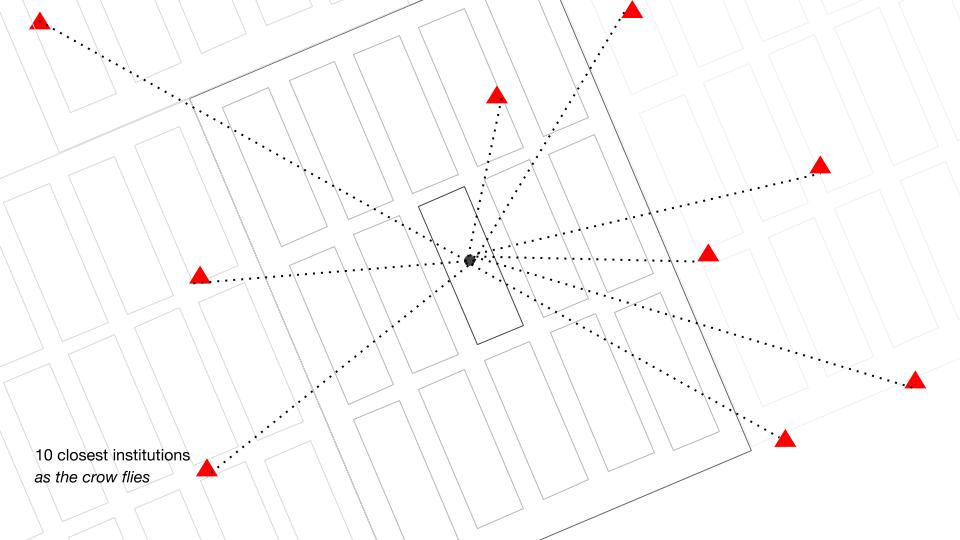
**2. Calculating Access**

An important contribution of this paper is our improvement in the measurement of access in the literature on the spatial distribution of conventional banks versus alternative financial institutions. In this section, we describe the process we used to calculate access in our analysis.

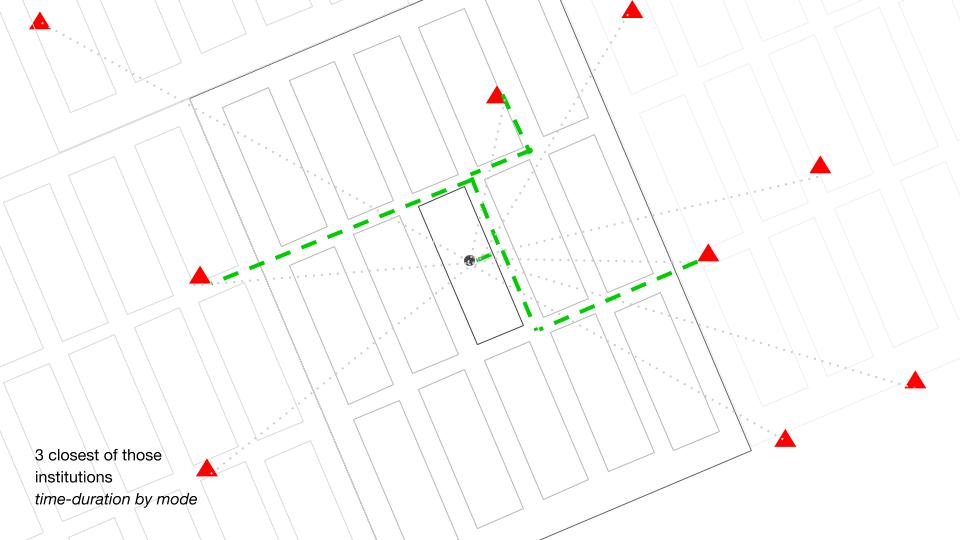
To minimize spatial aggregation error associated with accessibility measure calculations, we calculated our measures at the most granular census geography available: census blocks. We then summarized the block-level data for block groups when performing our analysis in relation to other socio-demographic attributes from census data.



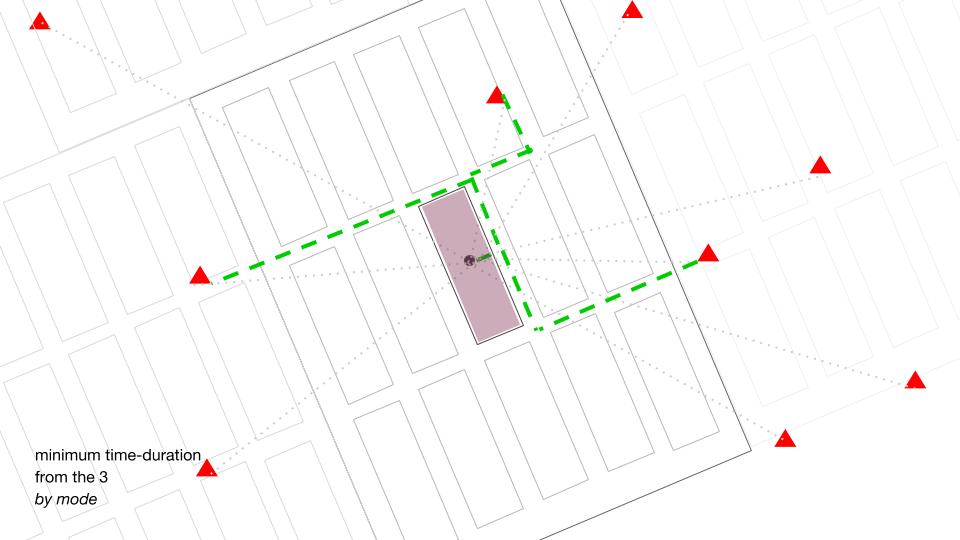
Calculating travel times for all possible combinations of blocks and amenities is not computationally feasible. We instead limited our calculations to the ten closest amenities (conventional banks and AFIs) based on their linear (“as the crow flies”) distance to each individual block’s centroid. We calculated linear distances using Postgres/PostGIS database’s ST\_Distance\_Sphere function.



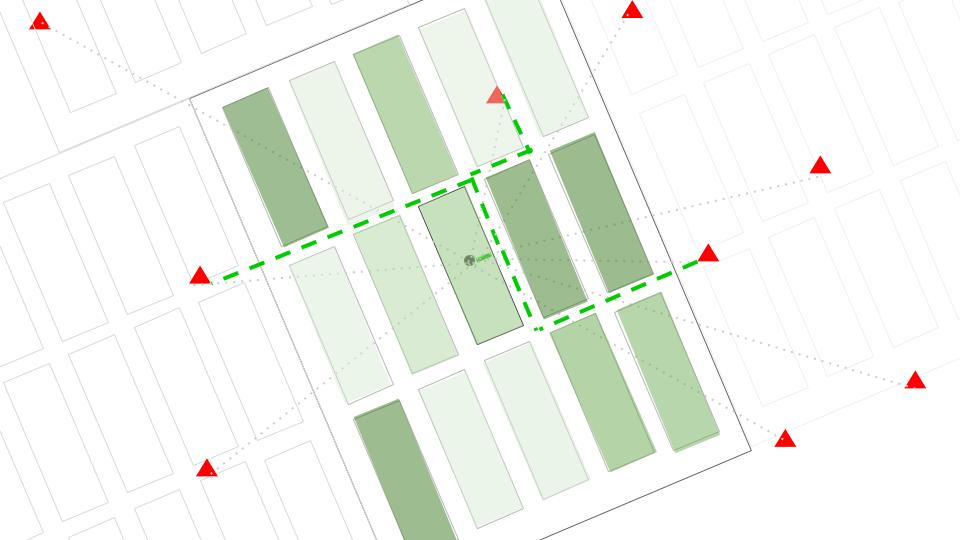
We then used open-source commercial routing engine Graphhopper to estimate the driving and walking times from each block centroid to the ten closest amenities. We repeated this process for all 20 cities in our analysis. We used OTP (Open Trip Planner) to calculate travel times for public transportation. We repeated this process for 19 of the 20 cities in our analysis. Public transportation data were not available for Memphis at the time of our analysis, and thus this city was excluded from public transit calculations.



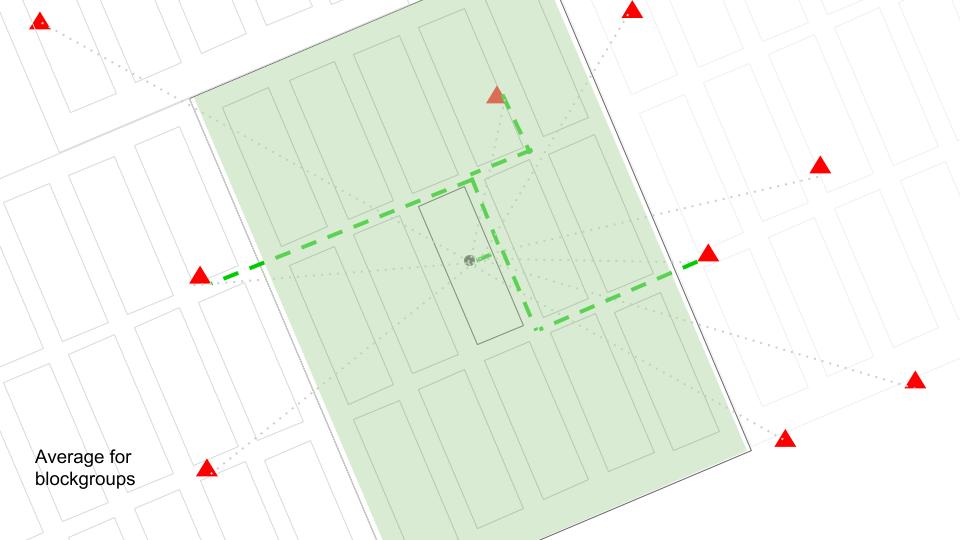
We then selected the three closest amenities using the duration-time measure.



We assigned the minimum of the three distances (vs. average of the three) to each block.



Finally, we calculated the average values of the blocks for assigning the time-duration to each block group.



**3 Data and Replication Code**

Dataset S1 (external link): Data and Replication Code for Calculating Access.

[https://github.com/urbaninformaticsandresiliencelab/bnk\_afi\_si/tree/master/scripts/python](about:blank)

Dataset S2 (external link): Data and Replication Code for the Random Effects Analysis of Time-Based Proximities to Banks and AFIs

[https://github.com/urbaninformaticsandresiliencelab/bnk\_afi\_si/tree/master/scripts/stata](about:blank)

1. Google Maps Platform (2019). Overview: Places API. Available online at [https://developers.google.com/places/web-service/intro](about:blank). [Accessed June 26, 2020] [↑](#footnote-ref-1)
2. Updated June 29, 2020. The data were collected using a previous version of Google Places APIs, and at the time collecting the data was free of charge. Google has updated its Place API since, and one can now incur a high cost by using the “nearby” function. [↑](#footnote-ref-2)