PopSim: An Individual-level Population Simulator for Equitable Allocation of City Resources[*]

Khanh Duy Nguyen, Nima Shahbazi, Abolfazl Asudeh[†]

Abstract

Historical systematic exclusionary tactics based on race have forced people of certain demographic groups to congregate in specific urban areas. Aside from the ethical aspects of such segregation, these policies have implications for the allocation of urban resources including public transportation, healthcare, and education within the cities. The initial step towards addressing these issues involves conducting an audit to assess the status of equitable resource allocation. However, due to privacy and confidentiality concerns, individual-level data containing demographic information cannot be made publicly available. By leveraging publicly available aggregated demographic statistics data, we introduce PopSim, a system for generating semi-synthetic individual-level population data with demographic information. We use PopSim to generate multiple benchmark datasets for the city of Chicago and conduct extensive statistical evaluations to validate those. We further use our datasets for several case studies that showcase the application of our system for auditing equitable allocation of city resources.

1 Introduction

Over the past several decades, the fast pace of urbanization has caused a sharp rise in city resource consumption. This not only affects residents' welfare levels but also plays a crucial role in shaping the sustainability of urban services and development. Unfortunately, a range of political and cultural forces motivated by racist attitudes towards people of color have created lines of separation between the citizens, effectively segregating people from different racial and ethnic groups into specific areas of the city. The impact of segregation remains particularly strong in cities like Chicago, which is home to some of the most deeply segregated areas that were once designated as "redlined" areas [14]. Such systematic exclusionary tactics have also been reflected in urban planning and resource allocation policies creating inequitable access levels to services such as public transport, healthcare, and education among different demographic groups.

Several organizations and research groups from a variety of disciplines, such as STEM (Science, Technology, Engineering, and Mathematics), social sciences, economics, and urban planning, are actively engaged in efforts to counteract the impact of the aforementioned tactics. More specifically, socially fair, just, and equitable resource allocation has gained significant attention in light of the growing concern for fairness in computational problems [3, 2, 10]. However, prior to any interventions, it is crucial to conduct a comprehensive audit of the current allocation of resources within the urban environment. Auditing may require access to individual-level data with demographic information. Unfortunately, obtaining such data is often challenging, as it is not publicly available due to concerns regarding privacy and confidentiality. Luckily, there is information readily available on aggregated demographic statistics for various neighborhoods (down to the block level in a city), providing a general overview of demographic distributions. This motivates us to build PopSim, a system for creating simulated individual-level data that is consistent with the available demographic statistics and can be utilized for auditing resource allocation within the city. In the development of our system, we use techniques such as Inverse-CDF [7] and Monte Carlo rejection sampling [16] to draw unbiased samples from different demographic groups and granularity levels in the geo-location hierarchies, ranging from an entire city down to a specific coordinate. PopSim enables its users to not only sample individuals but also generate large-size individual-level semi-synthetic datasets. It also enables designing a wide range of randomized and sampling-based algorithms for equitable allocation of resources.

To showcase an application of PopSim, we finetune it for Chicago and generate several semi-synthetic datasets with various sizes. We use statistical tests to validate that the simulated data follows the publicly available statistical information. Finally, we perform several case studies on the state of urban resource allocations such as public transport (Divvy bikes, CTA trains, and buses), schools, and hospitals in the city

[*] This project was supported in part by NSF 2107290.

[†] University of Illinois Chicago
{knguye71, nshahb3, asudeh}@uic.edu

of Chicago using the PopSim-generated data. Our experiments revealed a greater degree of equity in the allocation of schools, hospitals, and CTA bus stations, compared to the allocation of Divvy bikes and the CTA L train, which exhibited a high level of disparity. Furthermore, perhaps contrary to expectations, our results show a significant disadvantage for the Whites group. In addition to equity evaluation at the aggregate level, PopSim can be used for providing *visual explanations* for observed inequities. In particular, we use a set of samples generated by PopSim and generate a map that reveals the Divvy bikes inequities are due to a significantly lower access levels to the stations in the west and northwest neighborhoods of Chicago.

Summary of contributions. In summary, our contributions in this paper are as follows:

- We propose PopSim, a system for generating simulated individual-level population data that benefits from the publicly available aggregated demographic statistics.
- PopSim enables a wide range of applications requiring individual-level population data with demographic information. Two specific applications of PopSim include 1) creating semi-synthetic datasets that can be used for a variety of tasks, such as auditing equitable resource allocation, 2) Enabling the development of randomized algorithms for social applications for the urban population.
- We generated semi-synthetic benchmark datasets for Chicago and validated them through statistical tests, which serve as a tool to evaluate the equity of the allocation of city resources.
- We perform several interesting case studies investigating the equitable allocation of urban resources, such as *CTA trains, buses, Divvy bikes, schools*, and *hospitals* among different demographic groups.

2 System Overview

We aim to develop a system for simulating individual-level population data from publicly available city statistics that can be used for auditing the equitable allocation of resources. In particular, we would like to enable a variety of features that may be used to generate both semi-synthetic benchmarking datasets and individual sample generation for sampling-based approaches.

Figure 1 shows the high-level overview of our system. PopSim uses two types of publicly available data as input in its core: (i) *population statistics dataset* (§ 3.2.1) that provides the statistical information, and (ii) *geo databases* (§ 3.2.2) that is used for identifying the boundaries of geo-regions such as a block and zip code.

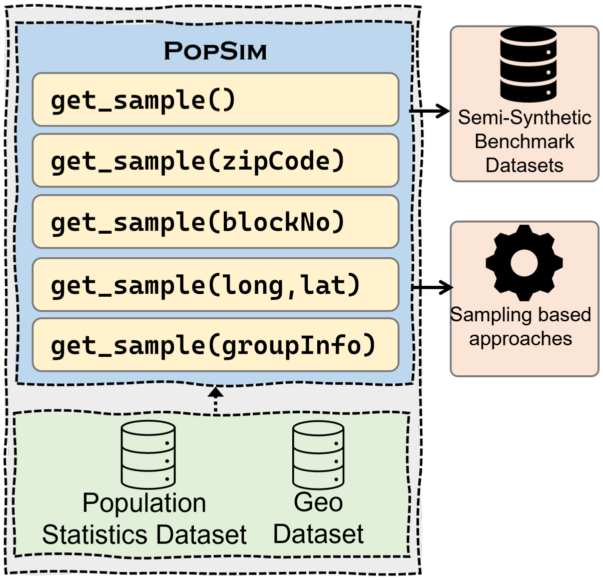


Figure 1: System Overview

PopSim provides a set of functionalities to sample from different granular levels of geo-location hierarchies and demographic groups. Each sample is a tuple in form of [⟨ `long, lat`⟩, ⟨ `groupInfo` ⟩], containing the sample location and its demographic information (`race`, `gender`, etc.).

- `get_sample()`: returns an unbiased sample based on the overall population distributions provided by population statistics dataset.
- `get_sample(zipCode/blockNo)`: returns an unbiased sample from the specified zip code or block[1].
- `get_sample(long,lat)`: returns an unbiased sample with the location as specified by the input longitude and latitude.
- `get_sample(groupInfo)`: returns an unbiased sample from the specified demographic group (e.g. `race`), or a set of demographic groups, based on the overall distribution of that group.

PopSim enables a wide range of applications that require individual-level population data with demographic information. Two specific applications of PopSim are:

[1] While being a standard notion as a fine-grained aggregate level in US cities, different countries may have other geographical units. Without loss of generality, in this paper, we used the US addresses system for explaining PopSim. To adapt PopSim for countries with different standards, it is enough to replace "block" with the most fine-grained geographical unit for which statistical aggregates are publicly available.

- *Semi-synthetic datasets:* Repeated sampling from PopSim provides benchmark datasets with individual-level demographic information that can be used for different tasks, such as auditing equitable allocation of city resources. For example, if one would like a dataset with n samples from the entire city (resp. a demographic group), it is enough to generate n samples using the `get_sample()` (resp. `get_sample(groupInfo)`). To demonstrate this, in this paper, as we shall further explain in § 5, we generated several datasets with various sizes from the city of Chicago and used them in § 6 for auditing equity in the allocation of various resources across the city.

- *Sampling-based approaches:* Unbiased sampling is a key requirement for Randomized Algorithms [13] and Monte-Carlo methods [9]. PopSim empowers developing sampling-based approaches for social applications for the city population. An example of such applications is fair allocation of resources, studied by [3, 2] (see § 7).

Implementation Details and Artifact availability. PopSim is an open source system, implemented `Python`, using `pandas`, `numpy`, and `geopandas` packages. In addition, we use `seaborns` and `matplotlib` libraries for the analytical step. The code is publicly available on GitHub.[2] As we shall explain in § 5, we use PopSim to generate several semi-synthetic datasets for the city of Chicago. The datasets are also publicly available[1].

3 Preliminaries

3.1 Sample Generation Techniques

We mainly use two sample generation techniques for simulating the individual-level data from the publicly available data:

- *Inverse-CDF Sampling* [7]. Also known as Inverse transform sampling, inverse-CDF is an approach for generating random samples from a given probability distribution. Let $f(x)$ be the reference probability density function (PDF). To draw samples from f, the inverse-CDF approach first computes the cumulative density function (CDF) of f, as $F(x) = P(X \leq x) = \int_{-\infty}^{x} f(x)dx$. It then computes the inverse CDF function $F^{-1}(x)$ and uses it for generating the samples. We will further explain this approach in § 4.1 for sampling from large regions. Indeed this approach is limited to the distributions for which the inverse CDF is commutable apriori.

- *Monte Carlo Rejection Sampling* [16]. This approach is useful for generating unbiased samples from a probability distribution with an odd-shaped probability density function ψ that is challenging to directly sample. The core idea behind this technique is to first, find a tight and "simple-to-sample" distribution (usually the bounding box) ξ that encloses ψ. Then, instead of sampling from ψ, it generates a sample from ξ. The sample is accepted if it falls under the curve of ψ. Otherwise, the sample is rejected (no sample is generated) and the process repeats. We use Monte Carlo rejection sampling in § 4.2 for generating unbiased samples within a block.

3.2 Input Datasets

PopSim takes two types of datasets as the input:

- *Population statistics datasets:* These datasets provide information about the population distribution across the city. Some of the well-known examples of such data include Decennial Census, American Community Survey (ACS), American Housing Survey (AHS), Population Projections, etc.

- *Geo databases:* These datasets provide spatial data such as geographic boundaries, with different levels of granularity, used to identify geo-regions varying from block to national level coverage.

Without loss of generality, in this paper, we fine-tune our system for the city of Chicago based on the following publicly available datasets.

3.2.1 Decennial Census of Population and Housing Data Database 2020

Decennial Census is a census of the population of the country that is conducted every 10 years, ending in a zero [4]. The census counts each resident of the country, recording where they reside on April 1st. The data contains fine-grained information at the **block level**, containing residency information such as the size of the household and the type of residence, as well as demographic characteristics of the population, including `gender`, `race and ethnicity`, and `age`. Census results are widely used for tasks such as determining the distribution of seats in the House of Representatives, shaping the boundaries of congressional districts, and annual allocation of federal funding. In this paper, we use the decennial data for the state of Illinois and more specifically the city of Chicago. We primarily focus on race-based statistics.

3.2.2 TIGER/Line Geodatabases 2022

The most detailed geospatial data for mapping census demographic statistics are the TIGER/Line data from the U.S. Census Bureau's Topologically Integrated Geographic Encoding and Referencing System [5].

[2] `https://github.com/UIC-InDeXLab/PopSim`

In this paper, we specifically, use the TIGER/Line geodatabase for identifying the boundaries of the blocks across the state of Illinois. While each block has several properties, we only use the Federal Information Processing Series (FIPS) code and the border geometry of Illinois blocks. FIPS codes are assigned to various geographic entities such as states, counties, metropolitan areas, cities, county subdivisions, consolidated cities, and indigenous areas, based on their alphabetical names.

4 System Development Details

After providing a high-level overview of our system and the preliminaries, in this section, we provide the development details of PopSim. In particular, in § 4.1 we first discuss sampling from the entire population, a specific zip code, or a demographic group. Next in § 4.2, we provide the details for sampling from the finest granularity levels, i.e., from a specific block or coordinate.

4.1 Sampling from A Large Region

Inverse-CDF is the core idea for sampling from regions larger than a block, i.e., a specific zip code, or the entire population. Recall from § 3.2.1 that the population statistics dataset contains fine-grained statistics at the *block* level. Therefore, in order to sample from a specific region R (e.g. a zip code), Algorithm 1 first samples a block within the specific region, with the probability density proportional to its size, and then returns a sample from the selected block.

Algorithm 1

```
1:  function GET_SAMPLE(zipCode=null)
        // find the set of blocks in given zipCode
2:      $B \leftarrow$ GET_BLOCKS(zipCode)
        // compute the cumulative function
3:      $F[0] \leftarrow 0$; $sum \leftarrow 0$
4:      for $i \leftarrow 1$ to $|B|$ do
5:          $F[i] \leftarrow F[i-1] + B[i]$.population
6:          $sum \leftarrow sum + B[i]$.population
7:      for $i \leftarrow 1$ to $|B|$ do $F[i] \leftarrow F[i]/sum$
        // find the block to sample next
8:      $u \leftarrow U[0,1]$ // random uniform in range [0,1]
9:      block ← BINARY_SEARCH(F,u)
10:     return GET_SAMPLE(block) //Algorithm 2
```

To further clarify how Algorithm 1 works, let us consider a toy example, where the selected region contains the following blocks with the specified populations:

blockNo	1	2	3	4	5	6
population	87	230	310	112	167	94

Following the lines 3 to 7 of Algorithm 1, the vector of the cumulative function F is computed as

blockNo	1	2	3	4	5	6
F	.087	.317	.627	.739	.906	1

Next, the algorithm draws a random uniform number in the range $[0, 1]$. Suppose the generated random number is 0.786. Since 0.786 is larger than 0.739 and smaller than 0.906, the binary search on F with 0.786 returns blockNo 5. Finally, the algorithm calls Algorithm 2 to draw an unbiased sample from block 5.

To draw a sample from a specific demographic group or a set of groups, `groupInfo`, one needs to first update the block populations to only include the counts for `groupInfo`. It should then limit the demographic groups of the selected block to `groupInfo` before calling Algorithm 2 to sample it.

4.2 Sampling from a specific block or location

Drawing an unbiased sample from a specific block requires (a) identifying the demographic information of the selected sample and (b) assigning a specific location (`long, lat`) to it.

Algorithm 2

```
1:  function GET_SAMPLE(blockNo)
        // (a) Find the groupNo of the selected sample
2:      g ← GET_GROUP(blockNo) //Algorithm 3
        // (b) specify the sample location
3:      $(x_{\triangleleft}, x_{\triangleright}) \leftarrow$ the minimum & maximum longitude of block[blockNo]
4:      $(y_{\bigtriangledown}, y_{\bigtriangleup}) \leftarrow$ the minimum & maximum latitude of block[blockNo]
5:      reject←true
6:      while reject do
7:          long $\leftarrow x_{\triangleleft} + U[0,1](x_{\triangleright} - x_{\triangleleft})$
8:          lat $\leftarrow y_{\bigtriangledown} + U[0,1](y_{\bigtriangleup} - y_{\bigtriangledown})$
9:          if IsInside((long,lat), block[blockNo]) then
10:             reject←false
11:     return $([long, lat], g)$
```

In Algorithm 2, we use Inverse-CDF for (a), similar to our approach in Algorithm 1. However, since the block boundaries do not form standard geometric shapes, we devise Monte-Carlo rejection sampling (§ 3.1) for (b). To do so, the algorithm first creates the tight bounding box around the specified block (lines 3 and 4). It then generates uniform random samples within the box and accepts them (line 10) if it falls inside the block.

In order to generate a sample from a specific location (`get_sample` in Algorithm 3), we first need to identify the corresponding block for the location. Then, it is enough to call `get_group` function to sample the demographic information of the selected sample.

Algorithm 3

```
function GET_GROUP(blockNo)
    groups$\leftarrow$ block[blockNo].groups
    $F[0] \leftarrow 0$; $sum \leftarrow 0$
    for $i \leftarrow 1$ to |groups| do
        $F[i] \leftarrow F[i-1] +$ groups[$i$].population
        $sum \leftarrow sum +$ groups[$i$].population
    for $i \leftarrow 1$ to |groups| do $F[i] \leftarrow F[i]/sum$
    $u \leftarrow U[0,1]$ // random uniform in range [0,1]
    return BINARY_SEARCH(F,u)

function GET_SAMPLE(long, lat)
    blockNo$\leftarrow$ BLOCK(LONG,LAT)
    return $([long, lat], get\_group(blockNo))$
```

4.3 System Extension While we demonstrate PopSim using the publicly available data sets for Chicago, its scope is indeed not limited to this city. First, to fine tune PopSim for a different city in the US, it is enough to use the populations statistics (Census Population Data) and geo-boundary databases of the target city. Tuning PopSim for other countries with different geographical units and aggregations statistics require replacing the notions such as zip-code and block to the standard notions in the target country.

It is easy to use heterogeneous statistical data sets to augment samples generated by PopSim with additional attributes. As an example, suppose one would like to add two columns `income` and `job title` to each sample. Note that `income` is ordinal continuous while `job title` is non-ordinal categorical. Data sets that provided these information at some aggregate level are publicly available (e.g., [8]). The first step to augment a sample with the additional data is to identify which unit it belongs to. For example, suppose the `income` and `job title` data are provided at the block level. Then given a sample, we should first identify which block it belongs to. Next, we should sample the attribute values according to the distribution of the given unit. For non-ordinal categorical attributes such as `job title`, one can use Inverse-CDF (similar to Algorithm 1 to draw an unbiased value (e.g., `job title: educator`). For ordinal continuous attributes, on the other hand, one can use Normal distribution (with the average and variance specified for the given unit) for specifying the attribute value (e.g., `income: $98,450`).

5 Benchmark Datasets

As previously mentioned in § 2, one of the applications of PopSim is to build semi-synthetic benchmark datasets. To demonstrate this, we generated six population datasets for the state of Illinois with PopSim. The largest dataset has an identical population size to that of the state of Illinois in the Decennial Census of Population dataset, containing $n = 12,812,508$ samples. The other datasets are of size $5M$, $1M$, $500K$, $200K$, and $50K$ samples, respectively. We shall later use these datasets for our case studies in § 6. But first, we use statistical tests in § 5.1 to confirm that the generated datasets indeed follow the underlying distribution by the input population statistics data.

5.1 Datasets Validation using Statistical Tests

In order to validate the generated datasets, we must ensure that they exhibit an identical distribution to that of the input population statistics dataset (Decennial Census of Population). Therefore, we establish the *null hypothesis* of "there is no substantial difference in population distribution between any of the PopSim-generated datasets and the Decennial Census of Population dataset", and proceed with a few statistical tests to reject it.

We start by normalizing both synthetic datasets and the Decennial Census of Population dataset, ensuring that our assessments concentrate on the underlying distributions of the data, rather than the size of the datasets. Next, we compute the mean (μ) and standard deviation (σ) of the datasets, enabling a comparison of the comparability of central tendency and dispersion across the various datasets.

We compare the distributions of each generated dataset with the Decennial Census of Population dataset based on two different properties.

- *Demographic Information:* We compare the distributions of each demographic group in the synthetic datasets with the total population of that group within the population statistics dataset.

- *Block:* We compare the population of each block in the synthetic datasets with the geo-distribution of that block according to the Decennial Census of Population dataset. Due to the significant variation in the block FIPS column, we combined every m adjacent block into one, prior to performing the statistical tests. The combined number of blocks m varies inversely with the size of the synthetic dataset. Specifically, for datasets of size 12.8M, 5M, 1M, 500K, 200K, and 50K, m takes on the values of 5, 10, 50, 100, 250, and 1000, respectively.

We used the **Kolmogorov–Smirnov (K-S) test** and

Datasets	t-test		K-S Test	
	Statistic	p-value	Statistic	p-value
50,000	0.50288	0.61611	0.40625	0.000312
200,000	0.21103	0.83323	0.21698	0.10847
500,000	0.09070	0.92787	0.11017	0.79395
1,000,000	0.05354	0.95738	0.10656	0.81513
5,000,000	0.0	1.0	0.06250	0.99972
12,854,526	0.01756	0.98601	0.05556	0.99971

Table 1: Statistical tests by race

Datasets	t-test		K-S Test	
	Statistic	p-value	Statistic	p-value
50,000	2.44894	0.999998	0.20317	4.21e-179
200,000	-6.7305	0.9999946	0.07142	2.10e-22
500,000	-4.9672	0.9999960	0.02846	0.00037
1,000,000	-3.4843	0.9999972	0.01115	0.39499
5,000,000	-3.1607	0.9999974	0.00681	0.65584
12,854,526	-3.5042	0.9999972	2.10259	1.0000

Table 2: Statistical tests by block (FIPS)

t-test to compare each synthetic dataset with the Decennial Census of Population dataset and test the null hypothesis. The results of the population comparison w.r.t *demographic group* (race) and *block* properties are presented in Tables 1 and 2 and confirm that the means values are not substantially different from those of the Decennial Census of Population dataset as evidenced by most of the p-values falling over the 0.05 threshold.

Table 2 shows a few exceptions when comparing based on the block property. For instance, according to the K-S test, the first three synthetic datasets have significantly different distributions from the Decennial Census of Population dataset, as their respective p-values are less than 0.05. However, the p-values of the three larger datasets suggest otherwise, indicating that they can be used with confidence for any related tasks.

For the race attribute, as demonstrated in Table 1, the K-S test results imply that with the exception of the first synthetic dataset (size 50K), the distributions of the remaining datasets are comparable to the Decennial Census of Population dataset.

Overall, the results indicate that, with the exception of the first three synthetic datasets, the null hypothesis can be rejected with high confidence. Therefore, the generated datasets are synthetic viable options for any task in need of individual data with demographic and geopositioning information.

6 Case Study

Having verified the validity of the synthetic datasets generated by PopSim, in this section, we perform several interesting case studies on the state of urban resource allocations in the city of Chicago. Specifically, we investigate the *equitable allocation* of the following urban resources across the city for difference *racial groups*: `(1) hospitals`[3], `(2) schools`[4], `(3) Divvy bikes stations`[5], `(4) CTA train stations`[6], and `(5) bus stops`[7]. Due to space limitations, we only focus on our largest dataset (with a population of 12,854,526 samples).

We define the "accessibility" of a resource as the *euclidean distance*[8] from an individual's geolocation to the closest resource of that particular type. We use the spatial KD-tree indexing [15] for locating the closest resource to each person. In order to evaluate racial equity in resource allocation, we compare the average distance-to-closest-resource for different groups. Formally, for each group $\mathbf{g}$ and the resource locations R, the average distance is computed using Equation 6.1.

$$\delta_R(\mathbf{g}) = \frac{1}{|\mathbf{g}|} \sum_{t \in \mathbf{g}} \min_{r \in R} \big(\text{dist}(t, r)\big) \tag{6.1}$$

6.1 Experiment Results Figure 2 shows our experiment results on the average distance-to-closest-resource (in meters) for different racial groups and urban resources in Chicago. Looking at the figure one can observe inequities in resource accessibility across all cases. On average, Asians had the highest distance to the closest school (449.50m), although they traveled the lowest distances to the nearest bus stop (142.90m) and CTA train station (1121.3m). Conversely, Black individuals walked the highest distance to the nearest bus stop (177.08m) and CTA train station (1992.6m), but the least distance to the nearest Divvy bike station (324.31m) and hospital (1636.35m). The maximum average distance traveled to the Divvy bike station was by Whites (601.44m), while the maximum for the nearest hospital was by Other Races (2170.20m).

[3] https://data.cityofchicago.org/Health-Human-Services/Hospitals-Chicago/ucpz-2r55

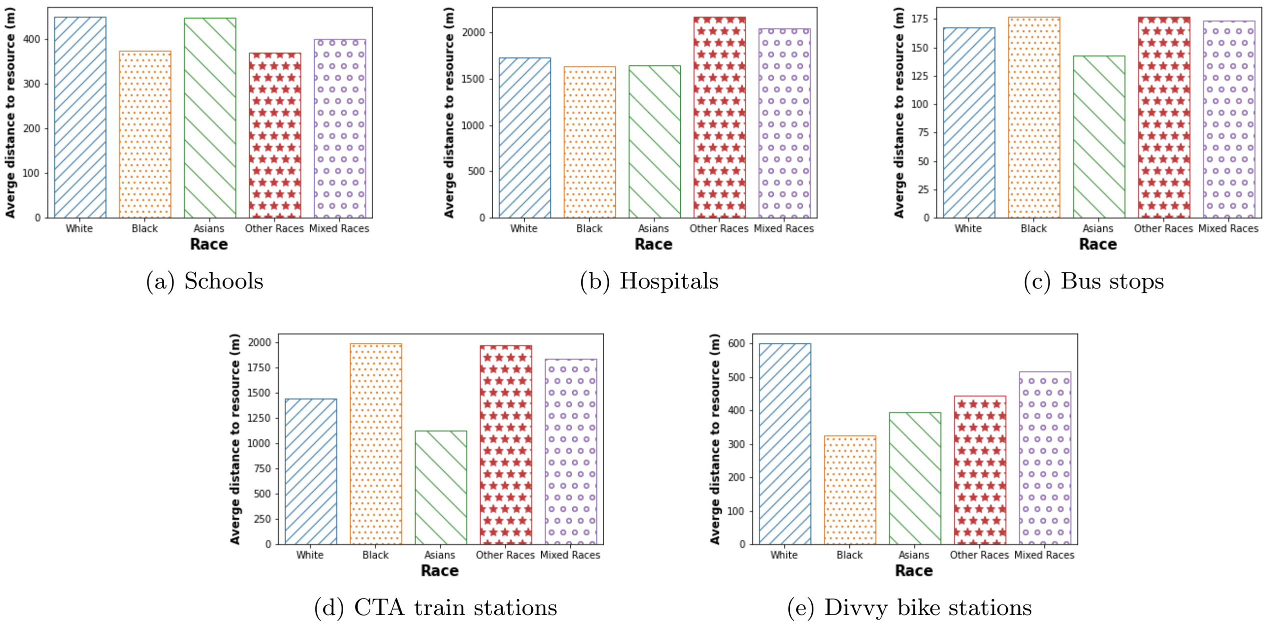
[4] https://data.cityofchicago.org/Education/Chicago-Public-Schools-School-Locations-SY2021/p83k-txqt/data

[5] https://data.cityofchicago.org/Transportation/Divvy-Bicycle-Stations-In-Service/67g3-8ig8

[6] https://data.cityofchicago.org/dataset/CTA-L-Rail-Stations-kml/4qtv-9w43

[7] https://data.cityofchicago.org/Transportation/CTA-Bus-Stops-kml/84eu-buny

[8] We transform the EPSG:4326 geographic coordinate system into the EPSG:26916 local projected coordinate system prior to calculating the metrics. This allows us to utilize the Euclidean distance formula to compute the distance with a 2.0 meters error in the region of Illinois [12]. We admit that actual route distance (walking distance) is more precise for computing the distance to resources. However, for simplicity, we use Euclidean distance.



(a) Schools

(b) Hospitals

(c) Bus stops

(d) CTA train stations

(e) Divvy bike stations

Figure 2: The Average distance-to-closest-resource for different racial groups and various urban resources in the city of Chicago. For presentation purposes, the populations of other races are combined as *"Other Races"*. *"Mixed Races"* includes all multi-racial population.

We measure the inequities in form of the *maximum disparity ratio*. For each resource R (e.g. bus stops) and the groups G, the maximum disparity ratio is computed using Equation 6.2 and reported in Table 3.

$$\text{disparity}(R) = \frac{\max_{\mathbf{g}\in G}(\delta_R(\mathbf{g}))}{\min_{\mathbf{g'}\in G}(\delta_R(\mathbf{g'}))} \tag{6.2}$$

	schools	hospitals	bus	train	Divvy
min	370.04	1636.35	142.90	1121.32	324.31
max	449.50	2170.20	177.08	1992.61	601.44
disparity	1.215	1.326	1.239	1.777	1.855

Table 3: Resource allocation disparities

Among the evaluated resources, allocation in `schools`, `hospitals`, and `bus stops` (Figures 2a, 2b, and 2c) were more equitable as the maximum disparity ratio was smaller. On the other hand, allocation of `CTA Train stations` and `Divvy bike stations` (Figures 2d and 2e) were more inequitable. In particular, *Divvy bike stations* had the maximum inequity, where the average distance of the White individuals to the closest station is 85% higher than that of Black individuals.

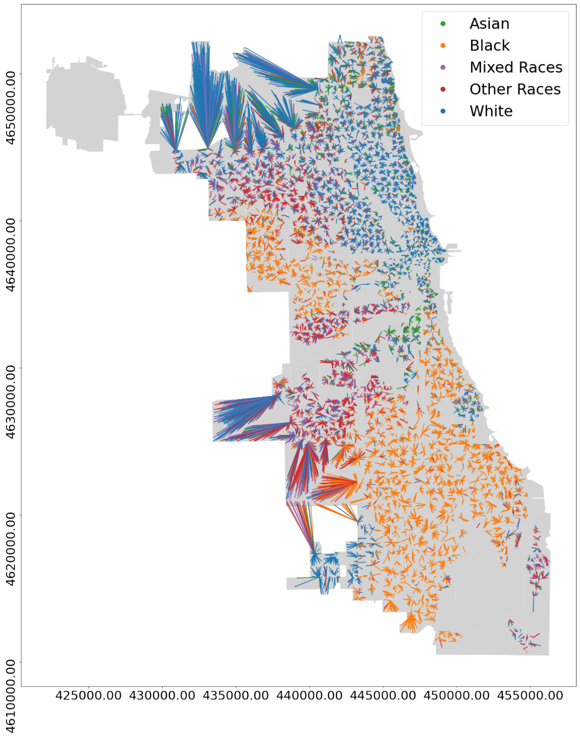


Figure 3: Visual Explanation of inequity in Divvy bike stations accessibility.

In addition to evaluating inequity, our datasets can be used for providing **visual explanations** for the inequities. In our experiments, we observed the maximum inequity for Divvy bike stations. Besides, perhaps counter-intuitively, we observed inequity against

White individuals. Therefore, to demonstrate using our datasets for visual explanations, we generated Figure 3 by generating 2000 random samples from Chicago using PopSim and connecting each sample to its closest Divvy Bike Station on the map. From the figure, it is evident that some of the mostly-White neighborhoods in the northwest (and west) of Chicago (with long blue lines) have caused the inequity. It turns out there is no Divvy bike station near those regions which caused an increase in the average travel distances for the White group.

Further investigating this issue, we realized that it happened due to the current phase of the Divvy expansion plan. The first phase of expansion focused on underrepresented communities in South Chicago, while the second phase, which began in 2021 and continues to the present, targets North West and South West communities[1]. The third phase, scheduled for the near future, will focus on the Northwest and Southwest areas. Furthermore, there has been significant interest expressed by users requesting new bike stations, as evidenced by the large number of requests on the proposed Divvy bike station map[6]. In particular, there is considerable demand for Divvy stations in North West communities like Jefferson Park, where there are currently no Divvy stations.

7 Related Works

Resource allocation problems have been extensively studied for decades in various disciplines namely, economics, management, healthcare, urban planning, computer science, etc. With the recent emergence of topics on fairness in computational problems, socially fair, just, and equitable resource allocation has drawn lots of attention. In [3], authors study the problem of fair resource allocation in the context of location problems where they try to determine the position of one or more facilities to satisfy the demand of a set of users while satisfying the fairness from the facilities' perspective. Similarly, in [2], they analyze a covering location problem with fairness constraints minimizing the pairwise deviations between the different covered sets. In [10], authors propose a set of 5 axioms for fairness measures based on which they construct a family of fairness measures for network resource allocation.

Many fairness-aware solutions for computational problems have been proposed in the past decade. These solutions usually fall into one of the categories of fairness-related interventions in the data, modifying the training process of the learning algorithms, and altering the outcomes of the models. Regardless of the level, at which the interventions are applied, they need to be evaluated empirically on benchmark datasets that represent realistic and diverse settings [11].

8 Final Remarks

In this paper, we introduced PopSim and used it to generate effective high-proximity benchmark datasets. The result datasets may be utilized for population statistical research and served as benchmarking datasets for recent fairness-aware solutions. As a disclaimer, we believe the inequalities found by our system and in our case study (§ 6) should not necessarily be seen as an indication of inequitable resource distributions, as other factors and criteria may also have been considered for resource allocation. For instance, different groups may have different demands on various resources and may prefer some resource types over the others. In such settings, an equitable allocation of resources that equally satisfies the demands of different groups may be different from an equal access to all resources.

References

[1] *Divvy expansion: Divvy bikes.* Accessed: March 20, 2023.

[2] A. Asudeh, T. Berger-Wolf, B. DasGupta, and A. Sidiropoulos, *Maximizing coverage while ensuring fairness: A tale of conflicting objectives*, Algorithmica, (2022), pp. 1–45.

[3] V. Blanco and R. Gázquez, *Fairness in maximal covering facility location problems*, arXiv preprint arXiv:2204.06446, (2022).

[4] U. C. Bureau, *Why we conduct the decennial census of population and housing*, Nov 2021.

[5] ——, *Tiger/line geodatabases*, Jan 2023.

[6] D. L. Chicago, *Divvy bike share locations*, n.d. Accessed: March 20, 2023.

[7] L. Devroye, *Sample-based non-uniform random variate generation*, in Proceedings of the 18th conference on Winter simulation, 1986, pp. 260–265.

[8] J. Diebel, J. Norda, and O. Kretchmer, *The demographic statistical atlas of the united states - statistical atlas.* statisticalatlas.com/place/Illinois/Chicago/Race-and-Ethnicity#data-map/tract, Accessed: March 20th, 2023.

[9] J. Hammersley, *Monte carlo methods*, Springer Science & Business Media, 2013.

[10] T. Lan, D. Kao, M. Chiang, and A. Sabharwal, *An axiomatic theory of fairness in network resource allocation*, IEEE, 2010.

[11] T. Le Quy, A. Roy, V. Iosifidis, W. Zhang, and E. Ntoutsi, *A survey on datasets for fairness-aware machine learning*, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 12 (2022), p. e1452.

[12] K. T. G. MapTiler team, *Epsg:26916*, 2020.

[13] R. Motwani and P. Raghavan, *Randomized algorithms*, Cambridge university press, 1995.

[14] A. Nardone, J. Chiang, and J. Corburn, *Historic redlining and urban health today in us cities*, Environmental Justice, 13 (2020), pp. 109–119.

[15] B. C. Ooi, *Spatial kd-tree: A data structure for geographic database*, in Datenbanksysteme in Büro, Technik und Wissenschaft: GI-Fachtagung Darmstadt, 1.–3. April 1987 Proceedings, Springer, 1987, pp. 247–258.

[16] D. Raeside, *Monte carlo principles and applications*, Physics in Medicine & Biology, 21 (1976), p. 181.