Jenna LoBasso

Dr. Hachadoorian

Spatial Database Design

# The Effects of Polling Location Accessibility

# Dataset Overview

The primary dataset utilized for this analysis comprises three main components: Polling Place data, Voter Turnout data and Voter Registration data for the 2016 General Election. These datasets are sourced from reputable governmental agencies and were collected for the purpose of conducting an in-depth spatial analysis to discern patterns and correlations related to polling place accessibility and voter turnout.

## Source and Purpose:

*Polling Place Dataset*
This dataset is provided by the Philadelphia City Commissioners. It provides comprehensive information about polling places, including their geographical coordinates (point geometry), division, precinct, ward, and zip code. It also includes accessibility codes and parking facilities codes which denote the accessibility of both the polling places and the parking provided. The dataset aims to facilitate the efficient administration of elections by ensuring that polling places are equipped with adequate resources and accessibility features.
*Voter Turnout Dataset*
The Voter Turnout data was procured from the Philadelphia City Commissioners. It contains pertinent details about voter participation, including precinct information, political party affiliation, and the total count of voters per candidate. This dataset was gathered with the objective of evaluating the effectiveness of polling place allocation and accessibility in influencing voter turnout.
*Voter Registration Dataset*
The Voter Registration data was procured from the Philadelphia City Commissioners. It includes information on the Philadelphia voter registrants per precinct, categorizing by race, political party, and gender. This dataset allows for voter turnout percentage calculations, which will be used for further analysis of the effect of polling location accessibility.

These datasets will be joined and normalized to express correlations between the accessibility of each polling place and its subsequent voter turnout. Additionally, Philadelphia Neighborhoods data as well as Philadelphia Population data will be utilized to implement spatial analyses.

For a detailed breakdown of data sourcing and the upload process, please refer to Appendix A.

---

## Spatial Questions

The analysis will revolve around addressing the following key spatial questions:

> *Do polling places located in areas with higher voter turnout exhibit better accessibility features?*

This question seeks to establish a correlation between the accessibility features of polling places and the voter turnout rates in their respective areas. By examining the accessibility codes in conjunction with voter turnout data, we aim to discern whether areas with higher turnout rates tend to have polling places with superior accessibility provisions.

> *Are there neighborhoods without polling locations, and if so, what impact does the absence of polling places in those areas have on the voter turnout of surrounding polling locations?*

This question delves into the relationship between the presence or absence of polling places in specific neighborhoods and the potential impact on voter turnout, not only within those neighborhoods but also in the surrounding areas. The absence of a polling place in a particular neighborhood may impact wait times at nearby polling locations, potentially discouraging individuals from participating. The inconvenience of travel could also serve as a deterrent for some voters. It considers the interconnectedness of voting patterns and accessibility to polling locations in a broader spatial context.

> *Do variations exist in the quantity of polling place locations concerning population density, and how does this potential disproportionality influence voter turnout?*

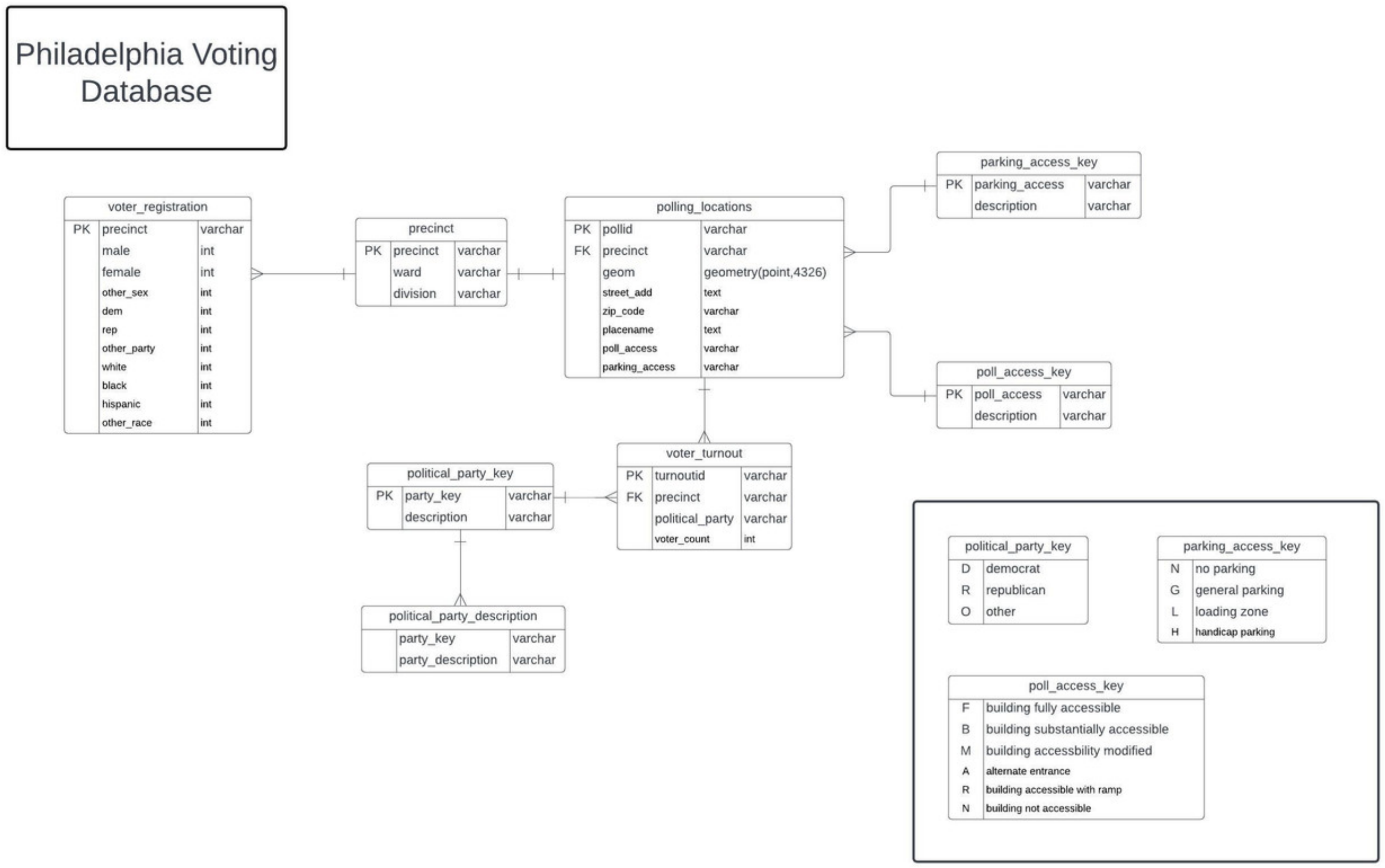
This question aims to investigate how the geographical placement of polling places relates to the density of the population. It seeks to determine whether certain areas exhibit a lack of convenient access to polling locations, potentially leading to a significan impact on voter turnout in those specific regions.

These combined datasets will afford a comprehensive view of the spatial dynamics influencing voter turnout and accessibility in the studied region, enabling us to draw meaningful conclusions and make informed recommendations for electoral process improvement.

---

## Normalization

The data used for this analysis required normalization. The image below is an entity relationship diagram that shows the normalization plan for the datasets discussed in the previous section.



The initial polling place dataset underwent normalization, resulting in the creation of two distinct tables. The first, now named "polling_locations," excluded the columns denoting the ward and division of each precinct. These excluded columns were relocated to a newly formed table termed "precinct." Concurrently, supplementary lookup tables were established for accessibility codes, comprising the "parking access key" and the "poll access key."

In the case of the original voter turnout dataset, the normalization process proved more intricate. Initial steps involved the formulation of two essential lookup tables. The first, the "political party description" table, encompassed the distinctive political party column from the original dataset, paired with a corresponding party key column. Given the multitude of parties in the original dataset, the party key column served to categorize them into

Democrat, Republican, and others. The second lookup table, the “political party key” table, elucidated the singular key variables D, R, and O. The finalized voter turnout table incorporated these new political party variables alongside their respective voter counts for each precinct. Additionally, a new primary key, “turnout id,” was introduced to address the absence of a unique constraint within the precinct column.

Conversely, the normalization of the original voter registration table proved comparatively straightforward. The sole modification involved the removal of the “total” column, given that it is a computable value.

For a detailed breakdown of the Extract-Transform-Load (ETL) processes employed, please refer to Appendix B.

---

## Optimization

To enhance the tables mentioned earlier for streamlined spatial analyses, three indices were generated specifically for the geometry of the Philadelphia Neighborhoods dataset, Philadelphia Population data, and the Polling Locations dataset. Additionally, a view table was generated (vt_percent), encompassing voter turnout percentage data for each precinct. Refer to Appendix C for a comprehensive overview of the index and view table construction process.

---

## Spatial Queries

> *Do polling places located in areas with higher voter turnout exhibit better accessibility features?*

```sql
select a.precinct, b.percent_turnout, a.poll_access, a.geom as p_geom, c.geom as n_geom
from polling_locations a
    join vt_percent b
        using (precinct)
    join philly_neighborhoods c
        on st_intersects(a.geom, c.geom)
where a.poll_access = 'F';
```

The aforementioned query addresses the correlation between voter turnout rates and accessibility features. It produces turnout percentages for precincts that are fully accessible. This query combines polling locations, percent voter turnout, and Philadelphia neighborhoods datasets through a spatial join using st_intersects. Incorporating both geometry columns facilitates a more thorough visual analysis when utilized in mapping software, such as QGIS.

> *Are there neighborhoods without polling locations, and if so, what impact does the absence of polling places in those areas have on the voter turnout of surrounding polling locations?*

```
select a.listname, b.precinct as closest_precinct, d.listname as nearest_neighborhood, di
stance, f.percent_turnout, b.geom
from
    philly_neighborhoods a
        cross join lateral
            (select c.geom, c.precinct, a.geom<->c.geom as distance
             from polling_locations c
             order by c.geom<->a.geom
             limit 3)
        as b
    join philly_neighborhoods d
        on st_intersects(b.geom,d.geom)
    left join polling_locations e
        on st_intersects(a.geom,e.geom)
    left join vt_percent f
        on b.precinct=f.precinct
where e.precinct is null
order by listname;
```

The mentioned query explores the potential consequences of lacking a polling place in neighborhoods. It outputs information on neighborhoods without polling locations, their nearest precinct, and the distance from said neighborhood to that precinct. Additionally, it includes the turnout rate of the closest precinct. This analysis employs spatial functions like st_intersects and K Nearest Neighbor (KNN) operations to obtain these results. The lateral join allows for the relationship to be refered to earlier entries in the FROM clause. Joining the Philadelphia Neighborhoods data twice allows for the result to include the column that represents the neighborhoods that do not have polling places and the column that shows in which neighborhood the nearest polling place resides.

> *Do variations exist in the quantity of polling place locations concerning population density, and how does this potential disproportionality influence voter turnout?*

```
select a.geography_, a.count_all_ as population_density, count(b.precinct) as precinct_count, avg(c.percent_turnout) as avg_percent_turnout, a.geom
from census_pop a
    join polling_locations b
        on st_intersects(a.geom, b.geom)
    join vt_percent c
        on b.precinct = c.precinct
group by a.geography_, a.geom, count_all_
order by count_all_ desc;
```

This query aims to investigate if variations exist in the number of polling places within each census tract and how this may impact voter turnout. By integrating census population data, polling location data, and voter turnout data, the analysis examines population density, associated precinct counts, and average voter turnout percentages within precinct clusters for each census tract. The spatial join operation, st_intersects, is employed to identify precincts that intersect with each census tract, ensuring a geographically relevant analysis.

---

# Appendices

## Appendix A:

### Acquiring the Data

The quantitative analysis relies on datasets from Open Data Philly, sourced from the Philadelphia City Commissioner.

### Loading the Data

To load the data, a schema was created for analysis. Various processes, including ogr2ogr commands for Excel files and QGIS for shapefiles were employed:

- For Excel files:
    - Voter Registration:

```
ogr2ogr -f PostgreSQL PG:"host=localhost port=5433 dbname=gis user=docker p
assword=docker" -lco SCHEMA=final_project -lco PRECISION=NO -oo EMPTY_STRIN
G_AS_NULL=YES -nlt PROMOTE_TO_MULTI qualified_voter_listing_2016_general_by
_precinct.csv
```

- Voter Turnout:

```
ogr2ogr -f PostgreSQL PG:"host=localhost port=5433 dbname=gis user=docker p
assword=docker" -lco SCHEMA=final_project -lco PRECISION=NO -oo EMPTY_STRIN
G_AS_NULL=YES -nlt PROMOTE_TO_MULTI voter_turnout_general_election_2016.csv
```

- For shapefiles:

1. Open QGIS's database manager.
2. Select PostGIS provider and the designated schema.
3. Click "Import Layer/File" and input necessary information.
4. Click "OK" and wait for "Import Successful" dialog.

Following these steps, the datasets were ready for normalization and analysis in DBeaver.

# Appendix B:

## Normalizing the data

The SQL scripts below show the normalization process of the data.

The SQL queries below show the creation of and addition to the polling locations table.

```
drop table if exists polling_locations cascade;
create table polling_locations
    (poll_id varchar primary key,
     precinct int,
     geom geometry(POINT, 4326),
     street_add text,
     zip_code varchar,
     placename text,
     poll_access varchar,
     parking_access varchar);

insert into polling_locations
select
    objectid::varchar,
    precinct::int,
    geom,
    street_add,
    zip_code,
    placename,
    accessibil,
    parking_co
from polling_places;
```

The SQL queries below show the creation of and addition to the precinct table.

```
drop table if exists precinct cascade;
create table precinct
    (precinct int primary key,
     ward int,
     division int);

insert into precinct
select
    precinct::int,
    ward,
    division
from polling_places ;
```

The SQL queries below show the creation of and addition to the voter registration table.

```
drop table if exists voter_registration cascade;
create table voter_registration
    (precinct int primary key,
     white int,
     black int,
     hispanic int,
     other_race int,
     democrat int,
     republican int,
     other_party int,
     male int,
     female int,
     other_sex int);

insert into voter_registration
select
    precinct::int,
    white:: int,
    black::int,
    hispanic::int,
    other_race::int ,
    dem::int ,
    rep::int ,
    other_party::int ,
    male::int ,
    female::int ,
    unknown_sex::int
from qualified_voter_listing_2016_general_by_precinct
where precinct not like 'Totals:';
```

The SQL queries below show the creation of and addition to the “poll access key” table.

```
drop table if exists poll_access_key cascade;
create table poll_access_key
    (access_key varchar primary key,
     description text);
```

```
insert into poll_access_key
values
    ('F', 'building fully accessible'),
    ('B', 'building substantially accessible'),
    ('M', 'building accessibility modified'),
    ('A', 'alternate entrance'),
    ('R', 'building accessble with ramp'),
    ('N', 'building not accessible');
```

The SQL queries below show the creation of and addition to the “parking access key” table.

```
drop table if exists parking_access_key cascade;
create table parking_access_key
    (parking_access varchar primary key,
     description text);

insert into parking_access_key
values
    ('N', 'no parking'),
    ('G', 'general parking'),
    ('L', 'loading zone'),
    ('H', 'handicap parking');
```

The queries below show the creation of and the addition to the “political party key” table.

```
drop table if exists political_party_key;
create table political_party_key
    (party_key varchar primary key,
     description varchar);

insert into political_party_key
values
    ('D', 'democrat'),
    ('R', 'republican'),
    ('O', 'other');
```

The SQL queries below show the creation of and addition to the “political party description” table.

```
drop table if exists political_party_description cascade;
create table political_party_description
    (party_key varchar,
     party_description varchar);

insert into political_party_description
    values
        ('D', '_DEMOCRACY'),
        ('R', 'CONSERVATIVE'),
        ('O', 'OTHER'),
        ('O', 'GOD'),
        ('D', 'HUMANITARIAN'),
        ('D', 'GREEN PARTY'),
        ('R', 'CONSTITUTION'),
        ('O', 'AMERICAN INDEPENDENT'),
        ('R', 'CONSTITUTIONAL'),
        ('R', 'PATRIOT'),
        ('R', 'NEW PATRIOT'),
        ('O', 'NOT SURE'),
        ('O', 'UNDETERMINED'),
        ('O', 'NO PARTY'),
        ('O', 'UNAFFILIATED'),
        ('R', 'FREEDOM'),
        ('O', 'CENTRIST'),
        ('R', 'LIBERTARIAN'),
        ('D', 'GREEN'),
        ('O', 'N/A'),
        ('O', 'INDEPENDANT'),
        ('D', 'GREEN PARTY OF THE US'),
        ('O', 'IND.'),
        ('O', 'PIRATE'),
        ('D', 'INDEPENDENT DEMOCRAT'),
        ('O', 'ANARCHIST'),
        ('O', 'NEUTRAL'),
        ('O', 'NONE'),
        ('O', 'I DONT KNOW'),
        ('O', 'INDEPENDENT'),
        ('R', 'MODERN WHIG'),
        ('R', 'REPUBLICAN'),
        ('R', 'TRUMP'),
        ('O', 'INDIVIDUAL'),
        ('D', 'MARXIST PARTY'),
        ('R', 'JUSTICE PARTY'),
        ('D', 'Libertarian Socialist'),
        ('D', 'WORKING FAMILIES PARTY'),
        ('R', 'FREE'),
        ('R', 'INDEPENDENT REPUBLICAN'),
        ('R', 'TEA PARTY'),
        ('O', 'EVERYDAY'),
        ('O', 'NO PARTY AFFILIATION'),
        ('O', 'Non Party'),
        ('D', 'TRANSHUMANIST PARTY'),
        ('R', 'CHRISTIAN'),
        ('O', 'NON-COMMITTAL'),
        ('D', 'NATIONAL SOCIALIST'),
        ('D', 'CITIZENS'),
        ('D', 'American Solidarity Party'),
        ('O', 'PA FOR PEROT'),
        ('D', 'CONSERVATIVE DEMOCRAT'),
        ('O', 'DEPENDS'),
        ('D', 'OBAMA'),
        ('R', 'WHIG'),
        ('D', 'COMMUNIST'),
        ('R', 'LIBERTARIAN NATIOAL SOCIALIST GREEN'),
        ('O', 'UNCERTAIN'),
        ('D', 'GREEN DEMOCRAT'),
        ('R', 'INDEPENDENT CONSERVATIVE'),
        ('R', 'PRO-LIFE'),
        ('O', 'U'),
        ('D', 'HILLARY'),
        ('O', 'BULL MOOSE'),
        ('O', 'UNDECLARED'),
        ('O', 'INDEPENT'),
        ('O', 'NON-PARTISAN'),
        ('O', 'CONSUMERS'),
        ('O', 'ANY'),
        ('D', 'HILLERY CLINTON'),
        ('O', 'NO LONGER IN USE'),
        ('R', 'LIBERTARIAN REPUBLICAN'),
        ('D', 'SOCIALIST DEMOCRAT'),
        ('D', 'Liberalist'),
        ('D', 'SOCIALIST'),
        ('D', 'SOCIAL DEMOCRAT'),
        ('O', 'Third Party'),
        ('O', 'RAINBOW'),
        ('O', 'Democratic Republican'),
        ('D', 'SOCIALIST WORKERS'),
        ('D', 'LABOR'),
        ('O', 'OPEN'),
        ('O', 'JEDI'),
        ('O', 'My Party'),
        ('O', 'NO AFFILIATION'),
        ('D', 'BLACK PANTHER'),
        ('D', 'SOCIALIST ALTERNATIVE'),
        ('D', 'DEMOCRATIC SOCIALIST'),
        ('O', 'UNDECIDED'),
        ('O', 'Old wagon Wheel'),
        ('R', 'AMERICAN'),
        ('O', 'ADARIAN'),
        ('O', 'MODERATE'),
        ('O', 'PERSONAL'),
        ('R', 'CONSERVATIVE INDEPENDENT'),
        ('O', 'REFORM'),
        ('O', 'UNKNOWN'),
        ('D', 'DEMOCRATIC'),
        ('O', 'INTERESTING'),
        ('D', 'PEACE AND FREEDOM'),
        ('O', 'INDEPENDENCE PARTY OF AMERICA'),
        ('D', 'PROGRESSIVE'),
        ('R', 'NEO-AGRARIAN'),
        ('O', 'INDEPENDENCE'),
        ('O', 'BIRTHDAY'),
        ('O', 'BI-PARTISAN'),
        ('D', 'LIBERAL'),
        ('D', 'AMERICAN LABOR');
```

The SQL queries below show the creation of and the addition to the voter turnout table.

```
drop table if exists voter_turnout cascade;
create table voter_turnout
    (turnout_id int generated always as identity primary key,
     precinct int,
     political_party varchar,
     voter_count int);

insert into voter_turnout (precinct, political_party, voter_count)
select
    a.precinct_code::int,
    b.party_key,
    sum(a.voter_count::int)
from voter_turnout_general_election_2016 a
    join political_party_description b
        on (a.political_party = b.party_description)
```

```
group by b.party_key, a.precinct_code
order by precinct_code asc;
```

# Appendix C:

## Creating the Indices

The SQL scripts below show the index queries used to optimize the data.

```
CREATE INDEX ON polling_locations USING gist(geom);


CREATE INDEX ON philly_neighborhoods USING gist(geom);


CREATE INDEX ON census_pop USING gist(geom);
```

## Creating the View Table

The SQL query below shows the creation of the percent voter turnout view table.

```
create view vt_percent as
select
    a.precinct,
    (1.0* (sum(a.voter_count))/(b.democrat + b.republican + b.other_party) * 100.00)
        AS percent_turnout
from voter_turnout a
    join voter_registration b
        using (precinct)
group by a.precinct, b.democrat, b.republican, b.other_party
order by precinct ;
```

# Appendix D:

## Data Dictionary

*Polling Locations*

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| poll_id | varchar | primary key | polling location id |
| precinct | int | foreign key | precinct id number |
| geom | geometry(POINT,4326) | | geometry points of polling locations |
| street_add | text | | polling location mailing address |
| zip_code | varchar | | polling location zip code |
| placename | text | | name of polling location |
| poll_access | varchar | | variable indication of polling place accessibility features |
| parking_access | varchar | | variable indication of polling place parking accessibility |

*Precincts*

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| precinct | varchar | primary key | precinct id number; concatenation of ward and division id's |
| ward | varchar | | electoral subdivision that houses the polling place |
| division | varchar | | electoral section that houses the polling place |

*Voter Registration*

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| precinct | varchar | primary key | precinct id number |
| male | int | | number of male voter registrants |
| female | int | | number of female voter registrants |
| other_sex | int | | number of unknown or other sex voter registrants |
| dem | int | | number of democratic voter registrants |
| rep | int | | number of republican voter registrants |
| other_party | int | | number of other party voter registrants |

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| white | int | | number of white voter registrants |
| black | int | | number of black voter registrants |
| hispanic | int | | number of hispanic voter registrants |
| other_race | int | | number of other race voter registrants |

*Voter Turnout*

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| turnout_id | varchar | primary key | unique constraint id |
| precinct | varchar | foreign key | precinct id number |
| political_party | varchar | | political party key per precinct |
| voter_count | int | | number of votes per political party per precinct |

*Parking Accessibility Key*

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| parking_access | varchar | primary key | variable indication of polling place parking accessibility |
| description | varchar | | description of what each parking access variable means |

*Polling Place Accessibility Key*

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| poll_access | varchar | primary key | variable indication of polling place accessibility features |
| description | varchar | | description of what each polling place access variable means |

*Political Party Key*

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| party_key | varchar | primary key | variable indication of political party |

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| description | varchar | | description of what each political party variable indicates |

*Political Party Description*

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| party_key | varchar | | variable indicator of political party |
| party_description | varchar | | the actual political party classified on voter ballots |

*Percent Voter Turnout - view*

| Field Name | Data Type | Constraint | Description |
|---|---|---|---|
| precinct | int | | precinct id number |
| percent_turnout | numeric | | percentage of registrants who voted |