



Methodology to Compare Districts and Schools: A Technical Report

Spring 2026

Table of Contents

Keywords	3
Contributors.	3
Introduction	4
Similar Peer Districts and Schools.	4
Design and Methods.....	4
Analytic Approach	5
Geographical Area	6
Design and Methods.....	6
Analytic Approach	6
Results	7
References.	8
Appendix	9

Keywords

Nebraska, compare, similar peers, Euclidean distance, Census data, Haversine distance, Python.

Contributors

This research effort was conducted by the following researchers at the office of Information Systems and Services at the Nebraska Department of Education:

- Shanshan Deng, Director & Psychometrician Lead
- Matthew Senseman, Statistical Research Specialist
- Ayotunde Akinleye, Ph.D., Statistical Research Specialist
- Jared Stevens, Statistical Research Specialist

We also acknowledge the pioneering contributions of the following previous NDE employees:

- Hongwook Suh, Ph.D.
- Shawn Gu
- Fisayo Adeniyan

Introduction

The Nebraska Education Profile (NEP) is a part of the Nebraska Department of Education’s initiative for providing high-quality information and data about Nebraska public schools and student performance to help inform educational policy and decision-making by stakeholders. The NEP website’s main utility is how it innovatively operationalizes similarity (i.e., of schools) in terms of Euclidean distances on 24 variables that characterize individual schools and districts, thus allowing for meaningful comparisons between schools and school districts. This technical report reiterates details of the study methodology and presents detailed citations for the sources of data to enhance the study’s reproducibility.

Similar Peer Districts and Schools

Design and Methods

To operationalize “similarity,” a combination of variables that uniquely describes each district or school was identified. These variables were selected due to their relevance, availability, and persistence. Table 1 presents the list of 24 variables that were selected to describe any given district or school. The reference section provides detailed information about the source(s) of data.

Table 1. Variables used to compare similarity between districts and schools.

VARIABLE	DESCRIPTION	SOURCE
Membership	Number of students enrolled	NDE
Attendance Rate	Average student attendance rate	NDE
FRL Rate	Percentage of free-and-reduced lunch students	NDE
Minority Rate	Percentage of non-White students	NDE
Homeless Rate	Percentage of homeless students	NDE
LEP Rate	Percentage of English language learners	NDE
Migrant Rate	Percentage of migrant students	NDE
Immigrant Rate	Percentage of immigrant students	NDE
Gifted Rate	Percentage of gifted students	NDE
SPED Rate	Percentage of Special Education students	NDE
Highly Mobile Rate	Percentage of students enrolled in two or more public schools during the school year	NDE
Mobility Rate	Percentage of mobile students	NDE
Teachers With Masters Percent	Percentage of teachers with at least a Master’s degree	NDE
Average Years Teaching Experience	Average number of years taught by teachers	NDE
Land Valuation	Annual land valuation sent out from the County Treasurer’s office of the district	NDE
Per Pupil Cost by Average Daily Membership	Total annual costs divided by the average daily membership for the district	NDE

Median Household Income	Median household income in the past 12 months (in 2024 inflation-adjusted dollars)	Census-ACS 2024 (5-Year)
Per Capita Income	Per capita income in the past 12 months (in 2024 inflation-adjusted dollars)	Census-ACS 2024 (5-Year)
Gini Index	Gini index of income inequality	Census-ACS 2024 (5-Year)
Percent Age 25+ With Bachelor's Degree or More	Percent of population 25 years and over with at least a Bachelor's degree	Census-ACS 2024 (5-Year)
Labor Force Participation Rate	Percent of population 16 years and over in the labor force	Census-ACS 2024 (5-Year)
Unemployment Rate	Percent of population 16 years and over who are unemployed	Census-ACS 2024 (5-Year)
Land Area	Area in square miles	Census 2020
Population Density	Population per square mile of land area	Census 2020

In creating the district and school data sets from various data sources, a number of challenges surfaced. First, the latest data from the Census was from 2020. Usually, the Census data was not updated as often as NDE's data on the districts and schools, but the Census data would still be informative since the variables described community characteristics (e.g., median household income, land area, etc.) that would likely not have changed as frequently as the school characteristics (e.g., membership, attendance rate, etc.).

Second, the Census data was only collected at the district-level, and not at the school-level. However, since the community characteristics of a given district would reflect that of the schools within the district, the same Census data was used at the school-level. This implied that all schools within the same district would, for example, have the same unemployment rate as that of the district.

Third, there were a number of districts that were consolidated after the Census data was collected. In these cases, the originating districts were first identified in the Census data, and the average values of the Census variables were then calculated to inform the Census variables for the new consolidated district.

Once the aforementioned decisions were made, a data split was performed on only the school data file. The school data file was split into three separate data files to reflect the differences among elementary, middle, and high schools. With three school data files, and one district data file, the analyses to identify similar districts and schools commenced.

Analytic Approach

Each district or school was compared to every other district or school by using a distance measure between each pair of districts or schools. This Euclidean distance measure was calculated as a summary index using the formula shown below:

$$d_{euc}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

In the formula above, d represents the Euclidean distance between any two districts or two schools x and y on each variable i (i.e., every variable shown in Table 1). Due to the wide differences in the ranges of values across the variables, each variable was scaled prior to computing the Euclidean distance.

Thus, for each district or school, the districts or schools with the shortest distances to it are grouped together. This is because the shorter the Euclidean distance between two districts or two schools, the more similar they are.

Geographical Distance

Design and Methods

The addresses for each district and school building were first converted into latitude and longitude information. Once this was done, the geographic distance (in miles) between every pair of districts and every pair of schools was calculated using the Haversine distance measure. Note that the school data file was split into three separate data files to ensure that similar school types were being compared to each other. For example, elementary schools were only compared with other elementary schools in terms of geographic distance. The same held true for middle schools and high schools as well.

Table 2. Variables used to describe geographic location for districts and schools.

VARIABLE	DESCRIPTION	SOURCE
Latitude	North-South geographic coordinate	Texas A&M Geocoding Services
Longitude	East-West geographic coordinate	Texas A&M Geocoding Services

Analytic Approach

Each district or school was compared to every other district or school by using a geographic distance measure between each pair of districts or schools. This Haversine distance represents the distance between two coordinates on a sphere and was calculated using the formula shown below:

$$d(x, y) = 2r \sin^{-1} \left(\sqrt{\sin^2 \left(\frac{\phi_x - \phi_y}{2} \right) + \cos(\phi_x) \cos(\phi_y) \sin^2 \left(\frac{\lambda_x - \lambda_y}{2} \right)} \right)$$

In the formula above, d represents the geographic distance in miles between any two districts or two schools x and y , with ϕ representing the latitude and λ representing the longitude, and r representing the Earth's radius in miles.

Results

The results of this work can be found as an interactive display in the Nebraska Education Profile website: <https://nep.education.ne.gov/#/> . Once a district or school is selected from the dropdown menu on the main page, the “Compare” feature can then be selected to show 12 other districts or schools that are most similar or geographically closest to the referent district or school. For questions or comments regarding the use of this feature, please reach out to NDE.Research@nebraska.gov.

References

U.S. Census Bureau, American Community Survey. (2026). Retrieved from <https://www.census.gov/data.html>

Appendix

All distance calculations were computed using Python, a programming language. The syntax is shown below.

Syntax for calculating Euclidean and Haversine distances for every pair of districts/schools.

Final Distance Calculations and Outputs

```
! Import necessary packages
from sklearn.metrics import DistanceMetric
from scipy.spatial.distance import cdist
from sklearn.preprocessing import MinMaxScaler

# Reading the geo data
df_geo = pd.read_csv("geo_2025_updated.csv")

# Convert Long and Lat to radians for calculation
df_geo['Longitude'] = np.radians(df_geo['Longitude'])
df_geo['Latitude'] = np.radians(df_geo['Latitude'])

# Split the data by school type
df_Dis_geo = df_geo[df_geo['SchoolTypeDistance'] == 'DISTRICT']
df_Sec_geo = df_geo[df_geo['SchoolTypeDistance'] == 'SECONDARY']
df_Ele_geo = df_geo[df_geo['SchoolTypeDistance'] == 'ELEMENTARY']
df_Mid_geo = df_geo[df_geo['SchoolTypeDistance'] == 'MIDDLE SCHOOL']

# Reading the full data
df_info = pd.read_csv("peers_2025_updated.csv")

# Scale the target variables except Membership (the three numbers below need represent the "AttendanceRate" column)
scaler = MinMaxScaler()
df_scaled = scaler.fit_transform(df_info.iloc[:, 17:].to_numpy())
df_info.iloc[:, 17:] = pd.DataFrame(df_scaled,
                                   columns=df_info.columns.to_list()[17:])

# Split the data by school type
df_Dis_info = df_info[df_info['SchoolTypeDistance'] == 'DISTRICT']
df_Sec_info = df_info[df_info['SchoolTypeDistance'] == 'SECONDARY']
df_Ele_info = df_info[df_info['SchoolTypeDistance'] == 'ELEMENTARY']
df_Mid_info = df_info[df_info['SchoolTypeDistance'] == 'MIDDLE SCHOOL']

# Define Haversine calculation
dist_h = DistanceMetric.get_metric('haversine')

# Create a function for repeated steps
def calculation(df_g, df_i):
    # Calculate the pairwise Haversine distance on Geo points
    column_tuples = [tuple(x) for x in df_g[['AgencyName', 'AGENCYID']].to_numpy()]
    res_h = pd.DataFrame(dist_h.pairwise(df_g[['Latitude', 'Longitude']].to_numpy()) * 3958.8,
                        columns=column_tuples, index=column_tuples)
```

```
# Create a function for repeated steps
def calculation(df_g, df_i):
    # Calculate the pairwise Haversine distance on Geo points
    column_tuples = [tuple(x) for x in df_g[['AgencyName', 'AGENCYID']].to_numpy()]
    res_h = pd.DataFrame(dist_h.pairwise(df_g[['Latitude', 'Longitude']].to_numpy()) * 3958.8,
                        columns=column_tuples, index=column_tuples)
    res_h.columns = pd.MultiIndex.from_tuples(res_h.columns)
    res_h.index = pd.MultiIndex.from_tuples(res_h.index)

    # Calculate the Euclidean distance/difference of all target variables
    column_tuples2 = [tuple(x) for x in df_i[['AgencyName', 'AGENCYID']].to_numpy()]
    mat_info = df_i.iloc[:, 16:].values #this number changes based on the "Membership" column location
    mat_res = cdist(mat_info, mat_info, lambda u, v: np.sqrt(np.nansum((u - v) ** 2)))
    res_e = pd.DataFrame(mat_res, columns=column_tuples2,
                        index=column_tuples2)
    res_e.columns = pd.MultiIndex.from_tuples(res_e.columns)
    res_e.index = pd.MultiIndex.from_tuples(res_e.index)
    return res_e, res_h

# Output the Datafiles
DistrictEuclideanDistance, DistrictGeographicDistance = calculation(df_Dis_geo, df_Dis_info)
SecondaryEuclideanDistance, SecondaryGeographicDistance = calculation(df_Sec_geo, df_Sec_info)
ElementaryEuclideanDistance, ElementaryGeographicDistance = calculation(df_Ele_geo, df_Ele_info)
MiddleEuclideanDistance, MiddleGeographicDistance = calculation(df_Mid_geo, df_Mid_info)

DistrictEuclideanDistance.to_csv("/content/drive/MyDrive/District Euclidean Distance 2025.csv")
DistrictGeographicDistance.to_csv("/content/drive/MyDrive/District Geographic Distance 2025.csv")
SecondaryEuclideanDistance.to_csv("/content/drive/MyDrive/Secondary Euclidean Distance 2025.csv")
SecondaryGeographicDistance.to_csv("/content/drive/MyDrive/Secondary Geographic Distance 2025.csv")
ElementaryEuclideanDistance.to_csv("/content/drive/MyDrive/Elementary Euclidean Distance 2025.csv")
ElementaryGeographicDistance.to_csv("/content/drive/MyDrive/Elementary Geographic Distance 2025.csv")
MiddleEuclideanDistance.to_csv("/content/drive/MyDrive/Middle Euclidean Distance 2025.csv")
MiddleGeographicDistance.to_csv("/content/drive/MyDrive/Middle Geographic Distance 2025.csv")
```