Exploring the Predictive Power of Socioeconomic Data on Homelessness in Los Angeles, CA.

Table of Contents

Introduction

Motivation

Homelessness represents a persistent and growing challenge in the United States[1] with the magnitude of homelessness particularly evident in Los Angeles, California[2]. While multiple private and public efforts have been launched in an attempt to stem the growing size of these populations, the practical results of investments have been difficult to come by[3]. Motivated by a proximity to the epicenter of this issue, Skid Row, this study was undertaken to look for possible answers that may ultimately help this underserved community.

Hypothesis

The inability to design and measure the impact of programs intended to reduce homelessness may be due to underlying issues in the data collected.

Research question

Does existing publicly available data include predictive markers of homelessness in Los Angeles?

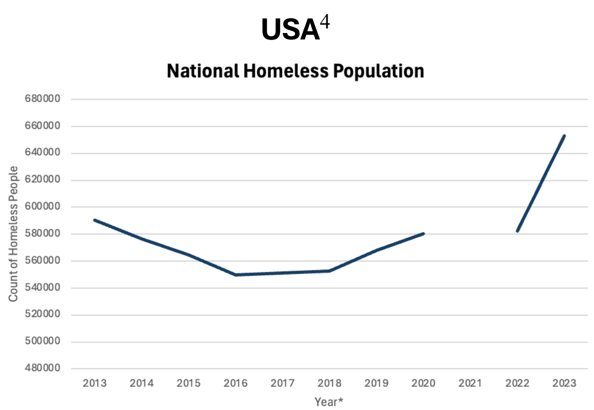


Figure 1 Rising homelessness in the U.S. No data for 2021 due to global pandemic.*

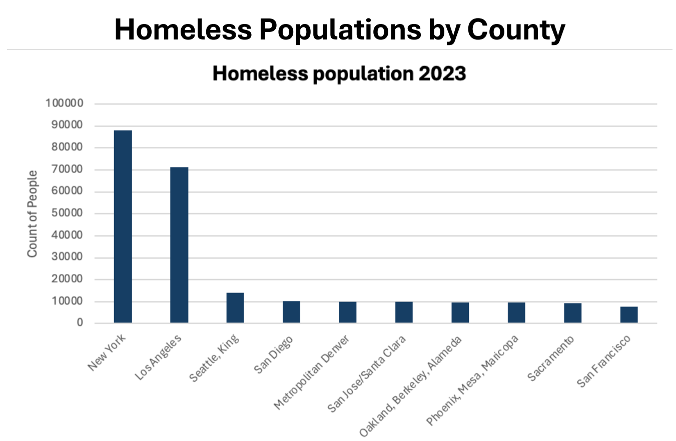


Figure 2 Comparison of homelessness in major U.S. counties.

---

[1] https://endhomelessness.org/homelessness-in-america/homelessness-statistics/state-of-homelessness/
[2] https://www.usnews.com/news/best-states/slideshows/cities-with-the-largest-homeless-populations-in-the-u-s
[3] https://www.npr.org/2024/04/10/1243825536/californias-effort-to-combat-homelessness-fails.../
[4] https://usafacts.org/articles/how-many-homeless-people-are-in-the-us-what-does-the-data-miss/

Methods

While multiple organizations report findings on homeless populations, there are just a few core data sources that are continuously re-used. And of these sources, on their own none lend themselves to statistical modeling methods without considerable data engineering efforts. This study attempted to use the following datasets and measure their efficacy in quantitative analysis:

U.S. Census 2020 (Census)
While rich in demographics, this national source-of-truth has no known attribute that can be directly used to measure populations of persons experiencing homelessness. This, despite literature that states that these populations are included in Census counts[5].

Housing and Urban Development (HUD)
HUD commissions an annual survey of homeless populations each year. Their Point-in-Time (PIT) surveys estimate populations of eight communities within Los Angeles. However, it is not possible to directly use the PIT data to measure the percent of a population experiencing homelessness or the percent that belong to a particular cohort (as an example, the percent of a community that are unwed mothers with children under 18 years that are experiencing homelessness).

Los Angeles Homeless Services Authority (LAHSA)
LAHSA manages the HUD PIT surveys each year for Los Angeles. However, published data cannot easily be used in conjunction with Census data to identify and measure deeper correlations. Each dataset uses non-standardized categories and datatypes that cannot easily be leveraged across other data to uncover answers on program impact. In addition, the data is pre-aggregated and not at the person or observation level.

Along with these syndicated public datasets, Kaggle and Github were investigated for additional resources. A scorecard of usability is shown in Table 1.

Data Source[6]	raw counts[7]	homeless flag[8]	non-homeless flag[9]	sub-county geos[10]
Census	No	No	No	Yes
HUD: PIT	No	Yes	No	Yes
Tom Byrne	No	Yes	No	Yes
Paul Beeman	No	Yes	No	Yes
Hiren Nisar	No	Yes	Yes	Yes
Adam Schroder	No	Yes	No	No

Table 1 Scorecard on data sources for this study.

[5] https://www.census.gov/content/dam/Census/library/factsheets/2020/dec/census-counts-homeless.pdf
[6] Source and location information can be found in the Appendix.
[7] Observation/individual vs. aggregated totals.
[8] Some sources use multiple categories of homelessness, some use none.
[9] Does the data allow us to analyze homelessness vs non-homelessness ?
[10] Does the data include the ability to see down to sub-county/city geographies ?

Data Models

As stated, the native source data does not lend itself to direct quantitative analysis of homelessness in part due to its non-normalized format, which typically presents absolute counts without contextual population ratios. To facilitate a more nuanced analysis, Census data was combined with HUD PIT survey data. This aggregation process was crucial to develop a dataset that allows for the identification of differential attributes across various populations, thereby enabling analysts to distinguish and analyze the factors that set one population apart from another.

```
[LAHSA Data]                                          [Census Data]
(ZIP Code, SPA) -------------------------------- (ZIP Code, Census Tract)
      |                                                      |
      ------------------> [Lookup Table] <--------------------
                    (ZIP Code, Census Tract, SPA)
                                 |
                    +----------------------------+
                    |   Calculate Population   |
                    |  per SPA from Census  |
                    +----------------------------+
                                 |
                        [Population Count]
              (SPA, Demographic, Population Count)
                                 |
                 +----------------------------------+
                 |  Calculate % Homelessness  |
                 |     for each Demographic      |
                 |             and SPA                 |
                 +----------------------------------+
                                 |
                     [Homelessness Percentage]
              (SPA, Demographic, Percent Homeless)
```

Figure 3 Data Aggregation Process

Exploratory Data Analysis

Dependent Variables

Counts of homeless populations from LAHSA use Service Planning Areas (SPA) to segment Los Angeles County into eight communities. This data allows for a more granular level of analysis than treating Los Angeles as a single homogenous community.

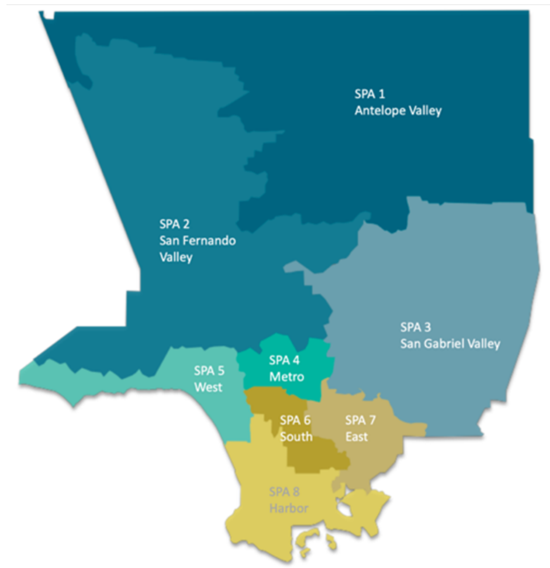


Figure 4 LAHSA data subdivides Los Angeles into 8 communities (referred to as SPAs)

LAHSA data shows significant variation of homelessness by Los Angeles community. This can be of value in understanding potential differences across the disparate communities in Los Angles necessary for measuring program impact.

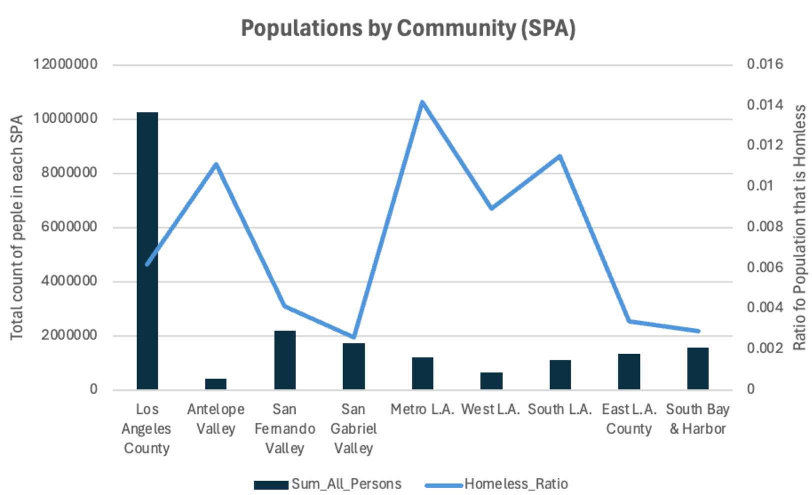


Figure 5 Variance of Homelessness Ratios by Community

Independent Variables

Census data is made available across multiple cohorts and aggregations. For this study the table, “Census P1 Total Population” was used. This table includes the following information per Census Tract (a geographic region):

- SEX: Count of populations across 2 different classifications (M/F)
- AGE: Count of populations across 17 different age groupings
- RACE: Count of populations across 8 different racial groups
- RELATIONSHIP: Count of populations across 8 different cohabitating cohorts
- HOUSEHOLDS BY TYPE: Count of populations across 14 different classifications
- HOUSING OCCUPANCY: Count of populations across 9 different classifications
- HOUSING TENURE: Count of populations across 3 different classifications

These data are available along multiple groupings to produce 160 total data points.

Variable	Count	Min	Mean	Max
HISPANIC.OR.LATINO.BY.RACE__Total.population	2561	0	4015.12	11373
HISPANIC.OR.LATINO.BY.RACE__Total.population__Hispanic.or.Latino	2561	0	1924.49	7579
HISPANIC.OR.LATINO.BY.RACE__Total.population__Hispanic.or.Latino__American.Indian.and.Alaska.Native.alone	2561	0	57.91	454
HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino__American.Indian.and.Alaska.Native.alone	2561	0	7.38	90
HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino__Asian.alone	2561	0	584.52	5135
HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino__Black.or.African.American.alone	2561	0	315.83	4914
HOUSEHOLDS.BY.TYPE__Total.households	2561	0	1372.46	6818
HOUSEHOLDS.BY.TYPE__Total.households__Cohabiting.couple.household	2561	0	102.37	826
HOUSEHOLDS.BY.TYPE__Total.households__Cohabiting.couple.household__With.own.children.under.18..3.	2561	0	34.45	153
HOUSEHOLDS.BY.TYPE__Total.households__Male.householder..no.spouse.or.partner.present.__Living.alone	2561	0	156.11	1703
HOUSEHOLDS.BY.TYPE__Total.households__Male.householder..no.spouse.or.partner.present.__Living.alone__65.years.and.over	2561	0	40.75	345
HOUSEHOLDS.BY.TYPE__Total.households__Male.householder..no.spouse.or.partner.present.__With.own.children.under.18..3.	2561	0	22.61	183

Table 2 Sample of data from Census table P1

Initial exploratory data analysis included the use of pairwise plots to visualize the relationships between all pairs of variables in the dataset. These plots were instrumental in identifying any obvious biases or anomalies such as outliers or clustering, which could influence a model's performance. This visual assessment helped confirm the data's suitability for further analysis, supporting the decision not perform transformations or more drastic data engineering steps before proceeding with statistical modeling.

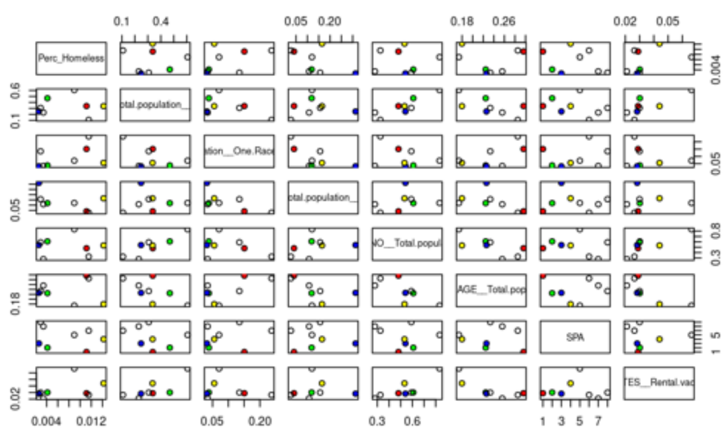


Figure 5 Pairwise Plots

Variable Selection

Census data was further analyzed to calculate the variance of values across all SPAs. The number of independent variables was reduced to those showing a high rate of variance across the communities (SPA) under the assumption that data with a higher variance may hold more predictive value. Multiple potential outliers are evident in the plots below. However, these data were left in until their impact was better understood; an outlier may be the marker of homelessness this study is looking for.

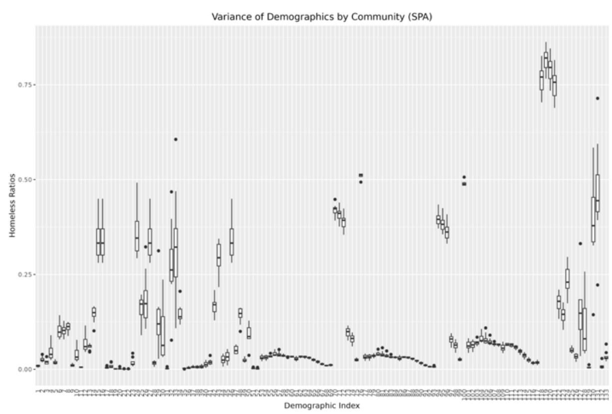


Figure 6 Variance of demographics within SPAs

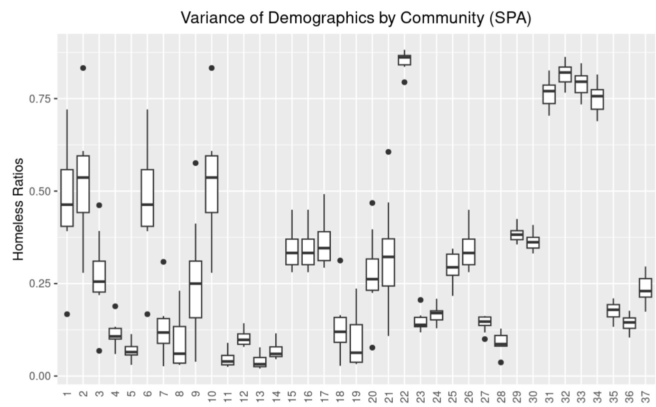


Figure 6 Variance of demographics within SPAs *after reduction*

Index	Name
1	HISPANIC.OR.LATINO__Total.population__Hispanic.or.Latino..of.any.race._sum
2	HISPANIC.OR.LATINO__Total.population__Not.Hispanic.or.Latino_sum
3	HISPANIC.OR.LATINO.BY.RACE__Total.population__Hispanic.or.Latino__Some.Other.Race.alone_sum
4	HISPANIC.OR.LATINO.BY.RACE__Total.population__Hispanic.or.Latino__Two.or.More.Races_sum
5	HISPANIC.OR.LATINO.BY.RACE__Total.population__Hispanic.or.Latino__White.alone_sum
6	HISPANIC.OR.LATINO.BY.RACE__Total.population__Hispanic.or.Latino_sum
7	HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino__Asian.alone_sum
8	HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino__Black.or.African.American.alone_sum
9	HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino__White.alone_sum
10	HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino_sum
11	HOUSEHOLDS.BY.TYPE__Total.households__Female.householder..no.spouse.or.partner.present.__Living.alone_sum
12	HOUSEHOLDS.BY.TYPE__Total.households__Female.householder..no.spouse.or.partner.present._sum
13	HOUSEHOLDS.BY.TYPE__Total.households__Male.householder..no.spouse.or.partner.present.__Living.alone_sum
14	HOUSEHOLDS.BY.TYPE__Total.households__Male.householder..no.spouse.or.partner.present._sum
15	HOUSEHOLDS.BY.TYPE__Total.households_sum
16	HOUSING.OCCUPANCY__Total.housing.units__Occupied.housing.units_sum
17	HOUSING.OCCUPANCY__Total.housing.units_sum
18	RACE__Total.population__One.Race__Asian_sum
19	RACE__Total.population__One.Race__Black.or.African.American_sum
20	RACE__Total.population__One.Race__Some.Other.Race_sum
21	RACE__Total.population__One.Race__White_sum
22	RACE__Total.population__One.Race_sum
23	RACE__Total.population__Two.or.More.Races_sum
24	RELATIONSHIP__Total.population__In.households__Child..2.__Under.18.years_sum
25	RELATIONSHIP__Total.population__In.households__Child..2._sum
26	RELATIONSHIP__Total.population__In.households__Householder_sum
27	RELATIONSHIP__Total.population__In.households__Opposite.sex.spouse_sum
28	RELATIONSHIP__Total.population__In.households__Other.relatives_sum
29	SEX.AND.AGE__Male.population__Selected.Age.Categories__18.years.and.over_sum
30	SEX.AND.AGE__Male.population__Selected.Age.Categories__21.years.and.over_sum
31	SEX.AND.AGE__Total.population__Over_19_sum
32	SEX.AND.AGE__Total.population__Selected.Age.Categories__16.years.and.over_sum
33	SEX.AND.AGE__Total.population__Selected.Age.Categories__18.years.and.over_sum
34	SEX.AND.AGE__Total.population__Selected.Age.Categories__21.years.and.over_sum
35	SEX.AND.AGE__Total.population__Selected.Age.Categories__62.years.and.over_sum
36	SEX.AND.AGE__Total.population__Selected.Age.Categories__65.years.and.over_sum
37	SEX.AND.AGE__Total.population__Under_20_sum

Table 3 Description of reduced data

To assess potential collinearity between variables in our dataset (which does not necessarily follow a normal distribution), Spearman's Rank Correlation was utilized. Unlike Pearson's correlation, Spearman's does not assume a linear relationship or normally distributed data. This non-parametric method allowed us to identify and subsequently focus on reducing variables with significant correlations. This approach was crucial given the potential non-linearity of our data as shown in subsequent sections.[11]

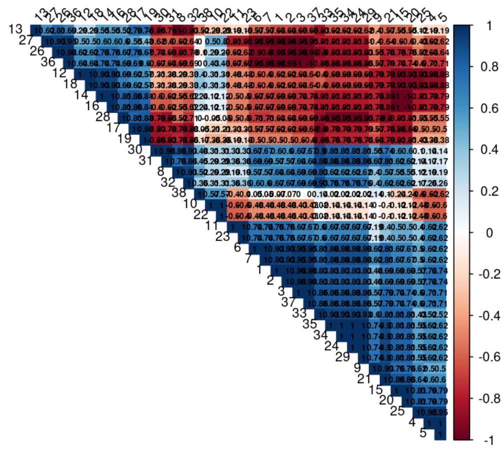
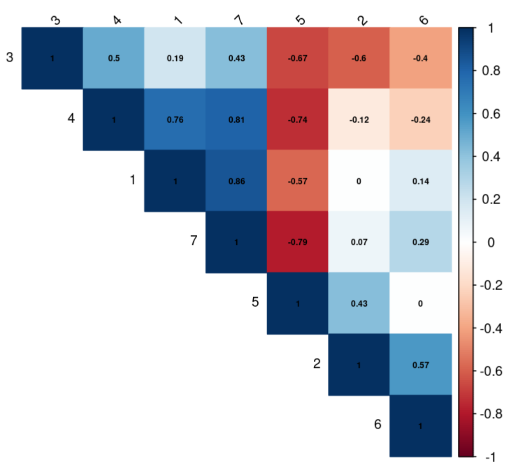


Figure 7 initial Spearman's Rank Correlation

.

Many values showed significant correlation. Variables were reduced through an iterative process until satisfactory results were produced with the following remaining values.



Index	Name
1	Race__White
2	Race__Black.or.African.American
3	Race__Asian
4	Race_Not.Hispanic.or.Latino
5	Age__Total.population__Under_20
6	Perc_Homeless
7	Rates__Rental.vacancy.rate

Figure 8 Final Spearman's Rank Correlation

[11] https://ademos.people.uic.edu/

Final Dataset

The hypothesis of this study is that progress against homelessness is due, in part, to underlying issues with the collection, pre-processing and distribution of data. Support of this hypothesis can be understood by examining the resulting dataset used in modeling:

```
$ SPA                                                                          : int [1:8]
$ Perc_Homeless                                                                : num [1:8]
$ Perc_Count__RACE__Total.population__One.Race__White_sum                      : num [1:8]
$ Perc_Count__RACE__Total.population__One.Race__Black.or.African.American_sum : num [1:8]
$ Perc_Count__RACE__Total.population__One.Race__Asian_sum                      : num [1:8]
$ Perc_Count__HISPANIC.OR.LATINO__Total.population__Not.Hispanic.or.Latino_sum: num [1:8]
$ Perc_Count__SEX.AND.AGE__Total.population__Under_20_sum                      : num [1:8]
$ Perc_Count__VACANCY.RATES__Rental.vacancy.rate..percent...5._mean            : num [1:8]
```

Table 4 Final dataset used in modeling.

The sparseness of the resulting dataset is due to nuanced root causes identified with this study:

- Homeless populations are not expressed as a proportion of total populations.
- Definitions of cohorts are not consistent across data sources.
- Geographic definitions are not aligned across data sources.
- Geographic regions are not stable over time.

Model Algorithm Selection

The data set's structure and inherent characteristics drive the selection of modeling techniques. Preliminary analysis indicated that the data do not conform perfectly to a normal distribution, a key assumption for many statistical methods. This was evident from skewness in the distributions and patterns observed in the diagnostic plots below.

Linear Regression[12]:

Initially, linear regression was considered for its simplicity and interpretability. However, diagnostic plots revealed certain inadequacies:

- Residuals vs. Fitted Plot: Displayed curvature, suggesting non-linear relationships between the predictors and the response variable.
- Q-Q Plot: Showed deviations from the expected line, suggesting heavy tails which could affect the robustness of the regression results.
- Scale-Location Plot: Although the spread of residuals was even, the presence of outliers as shown in the Residuals vs. Leverage plot suggested that a simple linear model might be insufficient.

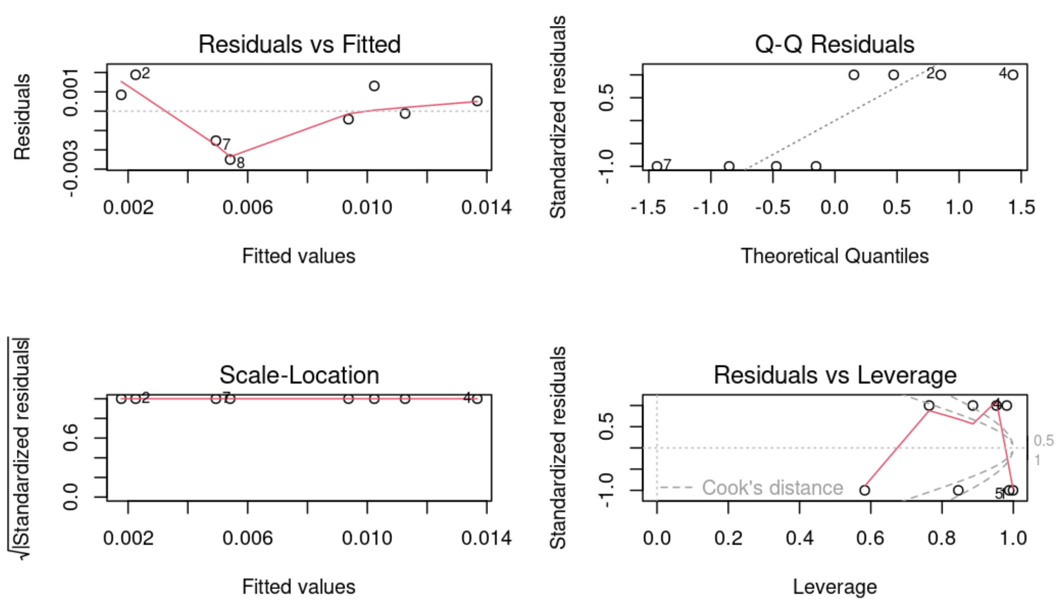


Figure 9 Diagnostic plots to confirm linearity

[12] Summary information is in the Appendix.

XGBoost[13]:

The decision to investigate the use of XGBoost was largely driven by its' success in multiple Kaggle competitions. Initial results showed an RMSE that exceeded our nominal homeless ratio (0.006) and a plot of learning rate between the training and test data shown that the model was overfitting (evident by the continued improvement of the training dataset after the testing dataset had ceased to improve). Despite k-fold cross-validation and a grid search for the best set of model parameters, the post-tuning RMSE increased to 0.036, which, while more reflective of the model's generalizability, remained above the acceptable threshold for predictive accuracy given the context of our homelessness data. This outcome led us to conclude that XGBoost, despite its advanced capabilities, was not suitable for providing reliable predictions in this particular study.

Figure 10 Model performance before grid search parameter tuning.

RMSE: 0.0079

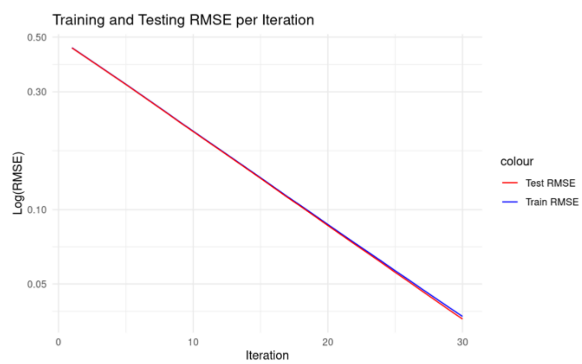


Figure 7 Model performance after grid search parameter tuning

RMSE: 0.036

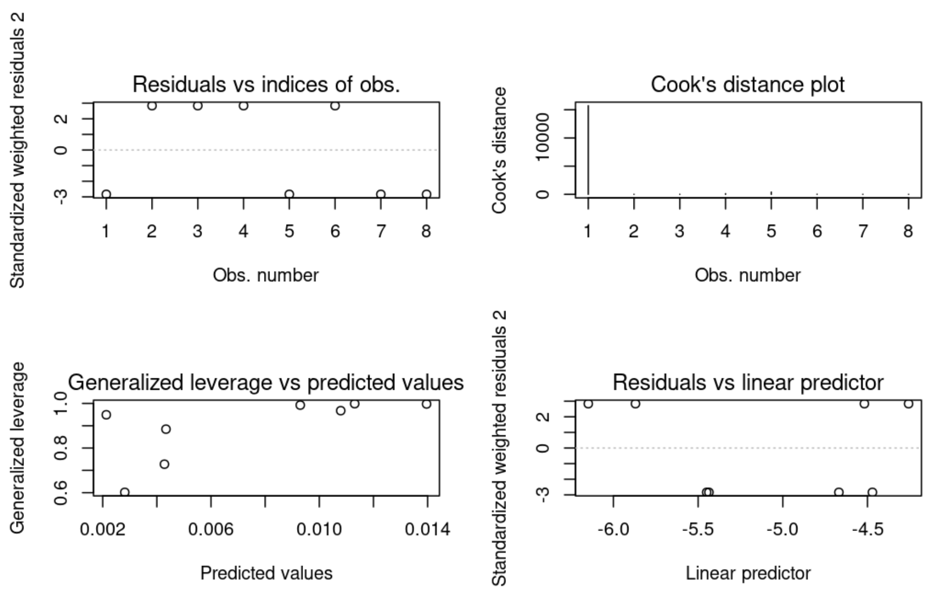
```
# grid of hyperparameters to search for my small dataset
grid <- expand.grid(
  nrounds = c(5, 10, 20),
  max_depth = c(2, 3),
  eta = c(0.01, 0.05),
  gamma = c(0, 0.1),
  colsample_bytree = c(0.5, 0.7),
  min_child_weight = c(1, 2),
  subsample = c(0.5, 0.8)
)
```

Table 5 Code snippet of the parameters used to tune the model

[13] Summary information is in the Appendix.

Beta Regression[14]:

The choice of Beta regression was driven by the bounded nature of the response variable—homelessness rates—which lie strictly between 0 and 1. This algorithm is particularly well-suited for handling proportional data, as it assumes a variable transformation that follows a beta distribution. Beta regression not only accommodates the skewness and heteroscedasticity inherent in the data but also provides more reliable estimates as confirmed by diagnostic plots below. These plots demonstrated the absence of patterned residuals, and no single observation exerted undue influence, validating the model fit. This robust methodological choice ensures that the analysis is well-grounded in statistical theory while tailored to the specific characteristics of this dataset.



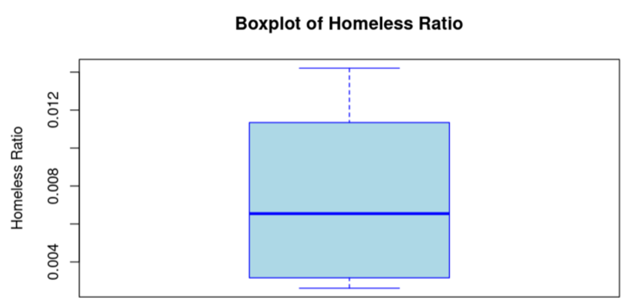


Table 6 Diagnostic plots from Beta Regression

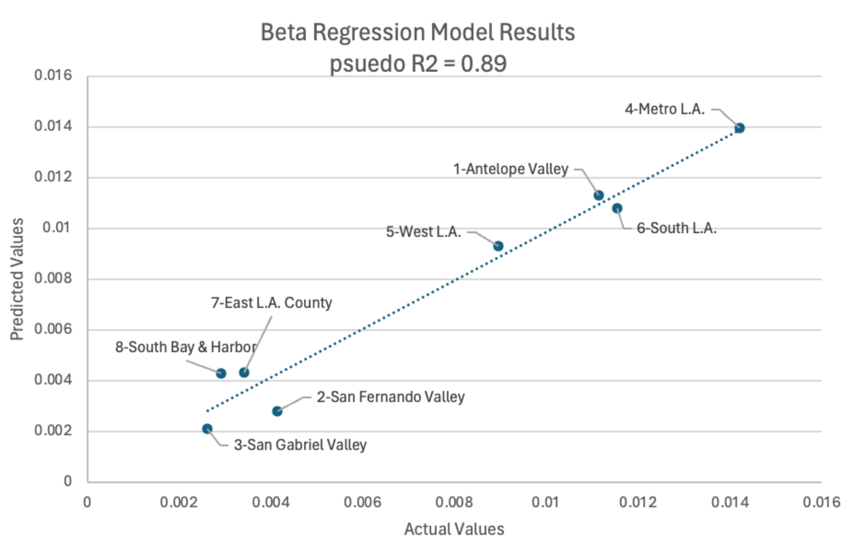
[14] Summary information is in the Appendix.

Results

The Beta regression model showed a reasonable R^2 of 0.89 with multiple significant independent variables. As shown below, race, ethnicity, age, and housing all show significant correlation with homelessness in Los Angeles. Unfortunately, the limited dataset did not allow us to leverage Bootstrapping or ANOVA to identify community level differences.

```
Call:
betareg(formula = Perc_Homeless ~ ., data = df_lr_b)

Standardized weighted residuals 2:
    Min      1Q  Median      3Q     Max 
-2.8396 -2.8396  0.0000  2.8396  2.8396 

Coefficients (mean model with logit link):
                                                                       Estimate Std. Error z value Pr(>|z|)    
(Intercept)                                                              -5.109      1.320  -3.870 0.000109 ***
Perc_Count__RACE__Total.population__One.Race__White_sum                 180.106     29.202   6.168 6.93e-10 ***
Perc_Count__RACE__Total.population__One.Race__Black.or.African.American_sum   225.979     36.389   6.210 5.29e-10 ***
Perc_Count__RACE__Total.population__One.Race__Asian_sum                 159.994     26.410   6.058 1.38e-09 ***
Perc_Count__HISPANIC.OR.LATINO__Total.population__Not.Hispanic.or.Latino_sum -147.002     23.324  -6.302 2.93e-10 ***
Perc_Count__SEX.AND.AGE__Total.population__Under_20_sum                 -94.537     15.612  -6.055 1.40e-09 ***
Perc_Count__VACANCY.RATES__Rental.vacancy.rate..percent...5._mean       -76.078     25.057  -3.036 0.002396 ** 

Phi coefficients (precision model with identity link):
      Estimate Std. Error z value Pr(>|z|)  
(phi)     5590       2803   1.994   0.0461 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Type of estimator: ML (maximum likelihood)
Log-likelihood: 43.59 on 8 Df
Pseudo R-squared: 0.8875
Number of iterations: 5000 (BFGS) + 15 (Fisher scoring)
```

Model fit here is very good, particularly when considering how little data was available for use. The model explains 88.75% of the variance in homelessness rates.



Challenges and Limitations

This study identified the following specific issues impacting efforts to use existing data to identify a significant number of predictive markers of homelessness in Los Angeles:

Geography: ZIP, Tract and SPA

Census

Defines a geography with a “tract” that is loosely defined as a collection of ZIP codes. However, specific ZIP code geographies are subject to change and translation tables between tract and ZIP code were observed to vary based on data source.

HUD

Defines a geography called a SPA that is also loosely defined as a collection of ZIP codes. However, any given SPA is not strictly defined by ZIP codes and a SPA can span more than one ZIP code or not fully encompass a given ZIP code.

Observations vs Aggregations

Census

Census reports that homelessness observations are included in their surveys. However, attempts to find these counts or classifications were unsuccessful.

HUD

Through PIT surveys, aggregated counts of homeless populations and demographics are available. However, analysis would be better served by making observation level data available. This can easily be cleaned of any Personably Identifiable Information to address privacy concerns.

Data Availability

LAHSA

Multiple efforts via email, LinkedIn messaging and in-person visits to LAHSA offices in Los Angeles to gain access to observation level data were unsuccessful.

Conclusions and Recommendations

A limited set of markers of homelessness were identified including younger populations and areas with higher rental housing vacancy rates showing lower correlations with homelessness, suggesting targeted interventions such as bolstering youth support services and managing rental stock, may effectively mitigate homelessness. While the findings of this study of Los Angeles generally align with findings in other national level research on the influence of housing[15], the study affirms the hypothesis: "The inability to design and measure the impact of programs intended to reduce homelessness may be due to underlying issues in the data collected on homelessness."

Despite the power of available data science tools, the lack of usable datasets on homelessness starkly contrasts with the publicly available data in other domains. To address this, the following specific changes would improve data utility for homelessness studies:

1. **Census as the Primary Data Source**: The Census should remain the authoritative source for all population data in the US, including detailed observational counts of homelessness, to ensure consistency and reliability across studies.
2. **Standardization of Geographic Definitions**: Geographic definitions should either remain constant or be easily translatable between different datasets to facilitate comparative and longitudinal studies.
3. **Accessibility of Observation-Level Data**: Observation level data should be made readily available through APIs, with appropriate privacy protections, to enhance the granularity and applicability of analyses.
4. **Leveraging Private and Public Tech Resources**: Public agencies should consider partnering with private technology firms like Google and Meta, or utilizing platforms like Kaggle or Stack, to leverage cutting-edge data science techniques.
5. **Advancing Future Research with Machine Learning**[16]: Future studies should explore machine learning techniques to measure and understand the complex interactions affecting homelessness more effectively. These techniques may offer more precise modeling capabilities than traditional regression models.

By implementing these recommendations, Los Angeles can better harness the potential of data science to understand and address the nuances of homelessness, ultimately leading to more effective interventions and policies.

---

[15] *"Market Predictors of Homelessness",* https://www.huduser.gov/portal/publications/Market-Predictors-of-Homelessness.html
[16] *"Los Angeles is using AI in a pilot program to try to predict homelessness and allocate aid"*
https://www.cnbc.com/2024/04/19/los-angeles-is-using-an-ai-pilot-program-to-try-to-predict-homelessness.html

Appendix: Dataset Sources

Data Source	Location
U.S. Census	https://www.census.gov/data.html
HUD: PIT	https://www.hudexchange.info/programs/hdx/pit-hic/
HUD: HMIS	https://www.lahsa.org/data-refresh
Los Angeles Open Data	https://data.lacounty.gov/datasets/
Tom Byrne	https://github.com/tomhbyrne
Paul Beeman	https://github.com/paulbeeman21
Hiren Nisar	https://www.huduser.gov/portal/publications/Market-Predictors-of-Homelessness.html
Adam Schroder	https://www.kaggle.com/adamschroeder
Capital Access Program	https://www.treasurer.ca.gov/cpcfa/calcap/evcs/disadvantaged.pdf
Crosswalk Files:	https://www.huduser.gov/apps/public/uspscrosswalk/home
LA County	http://publichealth.lacounty.gov/dhsp/Archived_Maps/ClusterAreasbyZipCode-SPA12-15.pdf
LA Almanac	https://www.laalmanac.com/health/he798.php

Comparison of Census data aggregation used in this study to other sources.

Populations: Total and per SPA

SPA	LA Almanac	LA County	Census Tables	Variance
1	413,966	418,046	426,612	2%
2	2,154,399	2,208,639	2,201,110	0%
3	1,720,779	1,753,582	1,741,054	1%
4	1,090,182	1,120,541	1,205,171	8%
5	648,902	664,790	71,049	1%
6	991,811	1,016,269	1,127,296	11%
7	1,258,726	1,281,049	1,343,614	5%
8	1,513,402	1,549,498	1,566,823	1%
Not in a SPA[17]			1,643,141	
Total Populations	9,792,167.00	10,012,414	11,925,870.00	19%

Populations: Total Homeless persons

Region		LAHSA	Census Tables	Variance
LA County		66,436	63,706	4%

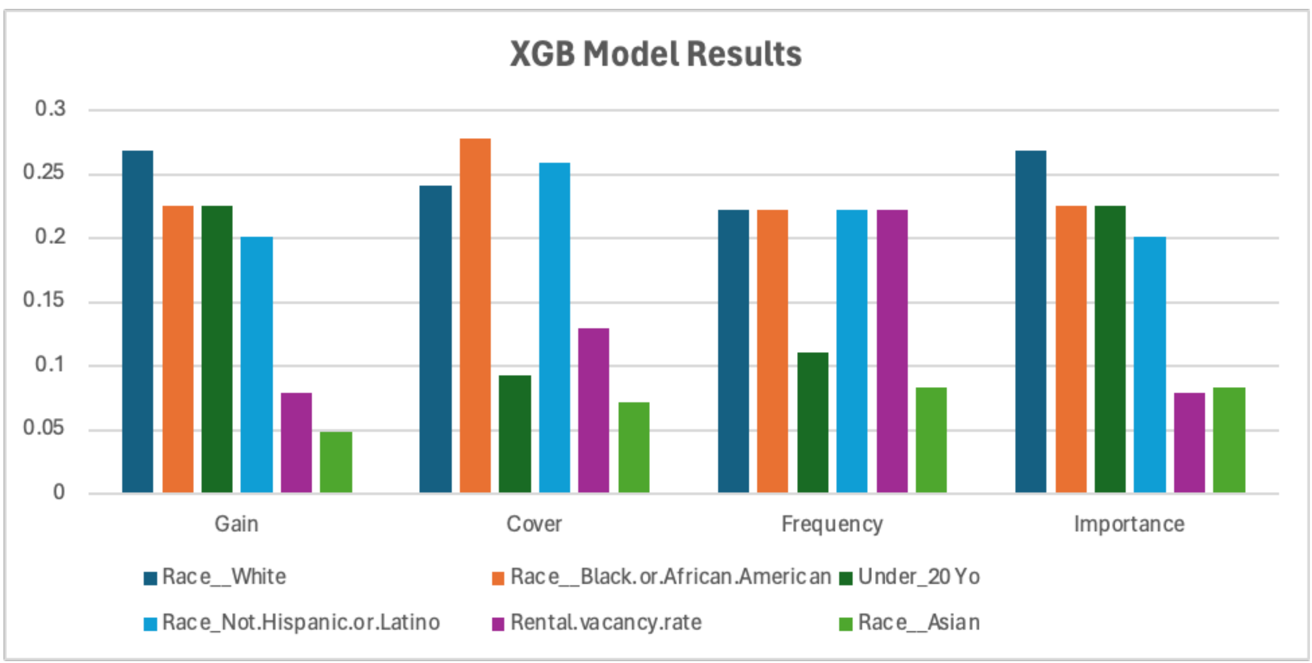
[17] Not all regions of LA are within a SPA

Appendix: LR Summary

```
Call:
betareg(formula = Perc_Homeless ~ ., data = df_lr_b)

Standardized weighted residuals 2:
    Min      1Q  Median      3Q     Max 
-2.8396 -2.8396  0.0000  2.8396  2.8396 

Coefficients (mean model with logit link):
                                                                      Estimate Std. Error z value Pr(>|z|)    
(Intercept)                                                             -5.109      1.320  -3.870 0.000109 ***
Perc_Count__RACE__Total.population__One.Race__White_sum                180.106     29.202   6.168 6.93e-10 ***
Perc_Count__RACE__Total.population__One.Race__Black.or.African.American_sum   225.979     36.389   6.210 5.29e-10 ***
Perc_Count__RACE__Total.population__One.Race__Asian_sum                159.994     26.410   6.058 1.38e-09 ***
Perc_Count__HISPANIC.OR.LATINO__Total.population__Not.Hispanic.or.Latino_sum -147.002     23.324  -6.302 2.93e-10 ***
Perc_Count__SEX.AND.AGE__Total.population__Under_20_sum                -94.537     15.612  -6.055 1.40e-09 ***
Perc_Count__VACANCY.RATES__Rental.vacancy.rate..percent...5._mean      -76.078     25.057  -3.036 0.002396 ** 

Phi coefficients (precision model with identity link):
      Estimate Std. Error z value Pr(>|z|)  
(phi)     5590       2803   1.994   0.0461 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Type of estimator: ML (maximum likelihood)
Log-likelihood: 43.59 on 8 Df
Pseudo R-squared: 0.8875
Number of iterations: 5000 (BFGS) + 15 (Fisher scoring)
```

Appendix: XGB Summary

```
> print(final_model$params)
$objective
[1] "reg:squarederror"

$eval_metric
[1] "rmse"

$max_depth
[1] 3

$eta
[1] 0.1

$colsample_bytree
[1] 0.5

$min_child_weight
[1] 1

$subsample
[1] 0.8

$validate_parameters
[1] TRUE
```



Appendix: Beta Summary

```
Call:
betareg(formula = Perc_Homeless ~ ., data = df_lr_b)

Standardized weighted residuals 2:
    Min      1Q  Median      3Q     Max 
-2.8396 -2.8396  0.0000  2.8396  2.8396 

Coefficients (mean model with logit link):
                                                                        Estimate Std. Error z value Pr(>|z|)    
(Intercept)                                                               -5.109      1.320  -3.870 0.000109 ***
Perc_Count__RACE__Total.population__One.Race__White_sum                  180.106     29.202   6.168 6.93e-10 ***
Perc_Count__RACE__Total.population__One.Race__Black.or.African.American_sum   225.979     36.389   6.210 5.29e-10 ***
Perc_Count__RACE__Total.population__One.Race__Asian_sum                  159.994     26.410   6.058 1.38e-09 ***
Perc_Count__HISPANIC.OR.LATINO__Total.population__Not.Hispanic.or.Latino_sum -147.002     23.324  -6.302 2.93e-10 ***
Perc_Count__SEX.AND.AGE__Total.population__Under_20_sum                  -94.537     15.612  -6.055 1.40e-09 ***
Perc_Count__VACANCY.RATES__Rental.vacancy.rate..percent...5._mean        -76.078     25.057  -3.036 0.002396 ** 

Phi coefficients (precision model with identity link):
      Estimate Std. Error z value Pr(>|z|)  
(phi)     5590       2803   1.994   0.0461 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Type of estimator: ML (maximum likelihood)
Log-likelihood: 43.59 on 8 Df
Pseudo R-squared: 0.8875
Number of iterations: 5000 (BFGS) + 15 (Fisher scoring)
```

Appendix: Source Code

Provision Enviornment

```
#install.packages("ggplot2")
#install.packages("dplyr")
#install.packages("stringr")
#install.packages("psych")
#install.packages("tidyr")
#install.packages("car")
#install.packages("corrplot")
#install.packages("betareg")
#install.packages("lmtest")
#install.packages("xgboost")
#install.packages("Hmisc")

# Load libraries
library(ggplot2)
library(dplyr)
library(stringr)
library(psych)
library(tidyr)
library(car)
library(corrplot)
library(betareg)
library(lmtest)
library(purrr)
library(xgboost)
library(Hmisc)
library(data.table)
library(caret)
library(randomForest)

print("DONE")
```

Build lookup table to translate between different datasets

```
# This file is a mapping of tract and zipcode for a lookup file
df_tab_20 <- read.csv("datasets/input_tab20_zcta520_tract20_natl.csv", sep = ",")

# Pre-processing
# Set feature names and data types to agree other datasets
df_tab_20$TRACT <- format(df_tab_20$TRACT, nsmall = 0)

# Remove "." character from Tract column
df_tab_20$TRACT <- gsub("\\.", "", df_tab_20$TRACT)

df_tab_20$TRACT <- trimws(df_tab_20$TRACT)

df_tab_20 <- df_tab_20 %>%
  rename(`ZIP` = GEOID_ZCTA5_20)

df_tab_20$ZIP <- as.character(df_tab_20$ZIP)

df_tab_20 <- df_tab_20 %>%
  rename(`Tract_Name` = NAMELSAD_TRACT_20)

df_tab_20 <- df_tab_20 %>%
  rename(`Tract` = TRACT)
```

```
df_tab_20$Tract <- as.character(df_tab_20$Tract)

df_tab_20 <- df_tab_20[, c("ZIP", "Tract", "Tract_Name")]

# Reduce to unique values
df_tab_20 <- df_tab_20 %>%
  select(ZIP, Tract) %>%
  distinct()

# Check for duplicates
duplicate_counts <- df_tab_20 %>%
  group_by(ZIP, Tract) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  filter(Count > 1)

# This file has zipcodes for  each SPA
df_spa_zip <- read.csv("datasets/input_SPA_ZIP_Calfund.csv", sep = ",")

df_spa_zip <- df_spa_zip %>%
  rename(`ZIP` = Zipcode)

df_spa_zip <- df_spa_zip[, c("SPA","ZIP")]

df_spa_zip <- df_spa_zip %>%
  mutate(ZIP = as.character(ZIP))

df_spa_zip <- df_spa_zip %>%
  mutate(SPA = as.character(SPA))

# Reduce to unique values
df_spa_zip <- df_spa_zip %>%
  select(SPA,ZIP) %>%
  distinct()

# This table merges the two prior to product SPA, ZIP and Tract
df_lookup <- df_tab_20 %>%
  left_join(df_spa_zip, by = "ZIP") %>%
  mutate(SPA = coalesce(SPA, "Not Available"))

# Reduce to unique values
df_lookup <- df_lookup %>%
  select(SPA,ZIP, Tract) %>%
  distinct()

# Check for duplicates
duplicate_counts <- df_lookup %>%
  group_by(Tract, SPA, ZIP) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  filter(Count > 1)

#print(names(df_lookup))
#print(df_lookup)

# Write data to a local file for offline use
write.csv(df_lookup, "datasets/output_df_lookup.csv", row.names = FALSE)

print("DONE")
```

Ingest P1 table from 2020 Census

```
# Ingest 2020 Census data
# Append SPA per Tract
# No aggregation at this point

df_census <- read.csv("datasets/input_2020_census_p1.csv", sep = ",")

df_census$Census.Tract <- format(df_census$Census.Tract, nsmall = 0)

df_census$Census.Tract <- trimws(df_census$Census.Tract)

# Remove "." character from Tract column
df_census$Census.Tract <- gsub("\\.", "", df_census$Census.Tract)

names(df_census) <- trimws(names(df_census))

df_census <- df_census %>%
  rename(Count_Tract = `Census.Tract`)

# Reduce dataset to just counts
df_census <- df_census %>%
  select(starts_with("Count"))

# Rename to a common value
df_census <- df_census %>%
  rename(Tract = `Count_Tract`)

# Append the SPA value to the Census data
# Do not include ZIP
df_census_spa <- df_census %>%
  left_join(df_lookup %>% select(-ZIP), by = "Tract") %>%
  distinct()

# Remove unusable data

unsuable_demos <- c(

  "Count__TOTAL.RACES.TALLIED..1.__Total.races.tallied",
  "Count__TOTAL.RACES.TALLIED..1.__Total.races.tallied__White.alone.or.in.combination.with.one.or.mor
e.other.races",
  "Count__TOTAL.RACES.TALLIED..1.__Total.races.tallied__American.Indian.and.Alaska.Native.alone.or.in
.combination.with.one.or.more.other.races",
  "Count__TOTAL.RACES.TALLIED..1.__Total.races.tallied__Asian.alone.or.in.combination.with.one.or.mor
e.other.races",
  "Count__TOTAL.RACES.TALLIED..1.__Total.races.tallied__Native.Hawaiian.and.Other.Pacific.Islander.al
one.or.in.combination.with.one.or.more.other.races",
  "Count__TOTAL.RACES.TALLIED..1.__Total.races.tallied__Some.Other.Race.alone.or.in.combination.with.
one.or.more.other.races",
  "Count__TOTAL.RACES.TALLIED..1.__Total.races.tallied__Black.or.African.American.alone.or.in.combina
tion.with.one.or.more.other.races",
  "Count__HOUSING.TENURE__Occupied.housing.units",
  "Count__HOUSING.TENURE__Occupied.housing.units__Owner.occupied.housing.units",
  "Count__HOUSING.TENURE__Occupied.housing.units__Renter.occupied.housing.units",
  "Count__MEDIAN.AGE.BY.SEX__Both.sexes",
  "Count__MEDIAN.AGE.BY.SEX__Male",
  "Count__MEDIAN.AGE.BY.SEX__Female"
)

# Remove the manually specified variables from the final reduced dataset
df_census_spa <- df_census_spa[, !(names(df_census_spa) %in% unsuable_demos)]
```

```r
# Move the SPA column to the first position
df_census_spa <- df_census_spa %>%
  select(SPA,Tract, everything()) %>%
  distinct()

# Order by SPA
df_census_spa <- df_census_spa[order(df_census_spa$SPA), ]

# Reduce to unique values
df_census_spa <- df_census_spa %>%
  distinct()

# Cast all values to Int in support of future manipulations
df_census_spa <- df_census_spa %>%
  mutate_all(as.integer)

# Print the ordered dataframe
#print(names(df_census_spa))

# Write data to a local file for offline use
write.csv(df_census_spa, "datasets/output_df_census_spa.csv", row.names = FALSE)

print("DONE")
```

Calculate pecent homeless by SPA

```r
# Calculate total populations per SPA based on Census data

df_census_spa_tot <- df_census_spa %>%
  select(SPA, "Count__SEX.AND.AGE__Total.population") %>%  # Keep only SPA and Count__SEX.AND.AGE__To
tal.population
  group_by(SPA) %>%
  summarise(Total_SPA_Census = sum(Count__SEX.AND.AGE__Total.population, na.rm = TRUE),
            .groups = "drop")

# Cast SPA to an int to join later
df_census_spa_tot <- df_census_spa_tot %>%
  mutate(SPA = as.integer(SPA))

# Eliminate entry rows
df_census_spa_tot <- df_census_spa_tot[!is.na(df_census_spa_tot$SPA), ]

#print(df_census_spa_tot)

rows_with_na <- df_census_spa_tot[!complete.cases(df_census_spa_tot), ]

# Cast SPA to an int to join later
df_census_spa_tot <- df_census_spa_tot %>%
  mutate(SPA = as.integer(SPA))

# Write data to a local file for offline use
write.csv(df_census_spa_tot, "datasets/output_df_census_spa_tot.csv", row.names = FALSE)

# Calculate homeless population per SPA

df_pit <- read.csv("datasets/input_2020_PIT.csv", sep = ",", check.names = FALSE)

# print(df_pit)

df_pit_spa_hmlss <- df_pit %>%
```

```r
  filter(Feature == "All Persons") %>%  # Filter to include only rows where Feature is "All Persons"
  group_by(SPA) %>%  # Group the data by SPA
  summarise(Total_Homeless = sum(Total, na.rm = TRUE))  # Sum the Total column, handling NA values

# Cast SPA to an int to join later
df_pit_spa_hmlss <- df_pit_spa_hmlss %>%
  mutate(SPA = as.integer(SPA))

# Write data to a local file for offline use
write.csv(df_pit_spa_hmlss, "datasets/output_df_pit_spa_hmlss.csv", row.names = FALSE)

#Join to get % homeless per SPA
df_pit_spa_hmlss_perc <- inner_join(df_pit_spa_hmlss, df_census_spa_tot, by = "SPA") %>%
          mutate(Perc_Homeless = Total_Homeless / Total_SPA_Census)

# Write data to a local file for offline use
write.csv(df_pit_spa_hmlss_perc, "datasets/output_df_pit_spa_hmlss_perc.csv", row.names = FALSE)

print("DONE")
```

Aggregate and normalize Census demographics by SPA

```r
#######AGGREGATE COUNT VALUES AS A SUM AND % VALUES AS A MEAN######

# Cast all values to Int in support of future manipulations
df_census_spa <- df_census_spa %>%
  mutate_all(as.integer)


# Note the vacancy information is difficult because it is on a different scale
# Age is too granular here. Need a better aggregations

# Define the names of the columns to sum
columns_to_sum <- c(
  "Count__SEX.AND.AGE__Total.population__Under.5.years",
  "Count__SEX.AND.AGE__Total.population__5.to.9.years",
  "Count__SEX.AND.AGE__Total.population__10.to.14.years",
  "Count__SEX.AND.AGE__Total.population__15.to.19.years"
)

df_census_spa <- df_census_spa %>%
  mutate(Count__SEX.AND.AGE__Total.population__Under_20 = rowSums(select(., all_of(columns_to_sum)),
na.rm = TRUE))

# Define the names of the columns to sum
columns_to_sum <- c(
"Count__SEX.AND.AGE__Total.population__20.to.24.years",
"Count__SEX.AND.AGE__Total.population__25.to.29.years",
"Count__SEX.AND.AGE__Total.population__30.to.34.years",
"Count__SEX.AND.AGE__Total.population__35.to.39.years",
"Count__SEX.AND.AGE__Total.population__40.to.44.years",
"Count__SEX.AND.AGE__Total.population__45.to.49.years",
"Count__SEX.AND.AGE__Total.population__50.to.54.years",
"Count__SEX.AND.AGE__Total.population__55.to.59.years",
"Count__SEX.AND.AGE__Total.population__60.to.64.years",
"Count__SEX.AND.AGE__Total.population__65.to.69.years",
"Count__SEX.AND.AGE__Total.population__70.to.74.years",
"Count__SEX.AND.AGE__Total.population__75.to.79.years",
"Count__SEX.AND.AGE__Total.population__80.to.84.years",
"Count__SEX.AND.AGE__Total.population__85.years.and.over"
)
```

```r
df_census_spa <- df_census_spa %>%
  mutate(Count__SEX.AND.AGE__Total.population__Over_19 = rowSums(select(., all_of(columns_to_sum)), n
a.rm = TRUE))

# Remove missing SPA values where tract did not map to a SPA
df_census_spa <- na.omit(df_census_spa)

df_census_perc <- df_census_spa %>%
  # Drop the "Tract" column
  select(-Tract) %>%

  # Group by SPA
  group_by(SPA) %>%

  summarise(

    # Calculate mean for columns that include "VACANCY.RATES" and divide by 100
    across(contains("VACANCY.RATES"), ~ mean(./100, na.rm = TRUE), .names = "Perc_{.col}_mean"),
    # Sum all other columns
    across(!contains("VACANCY.RATES"), sum, .names = "{.col}_sum"),
    .groups = "drop"
  )

# Join df_census_perc with df_census_spa_tot to append Total_SPA_Census per SPA
df_census_perc <- df_census_perc %>%
  left_join(df_census_spa_tot, by = "SPA")

# Divide columns that don't include "VACANCY.RATES" or "SPA" by Total_SPA_Census
exclude_columns <- c(
  "Perc_Count__VACANCY.RATES__Homeowner.vacancy.rate..percent...4._mean",
  "Perc_Count__VACANCY.RATES__Rental.vacancy.rate..percent...5._mean",
  "SPA"
)

# Calculate percentages for all columns except the ones explicitly named
df_census_perc <- df_census_perc %>%
  mutate(across(
    .cols = !all_of(exclude_columns),  # Exclude specified columns
    .fns = ~ . / Total_SPA_Census,  # Apply division function to remaining columns
    .names = "Perc_{.col}"  # Rename resulting columns
  ))

df_census_perc%>%
  select (SPA, Perc_Total_SPA_Census)

# Remove the 'Total_SPA_Census' after calculation
df_census_perc <- select(df_census_perc, -Perc_Total_SPA_Census)

#print(names(df_census_perc))

# Keep only SPA and columns with the prefix "Perc"
df_census_perc <- select(df_census_perc, SPA, starts_with("Perc"))

# Write data to a local file for offline use
write.csv(df_census_perc, "datasets/output_df_census_perc.csv", row.names = FALSE)

print("DONE")
```

Create master table of all data

```
# Append SPA homeless Rate to the census rates

merged_df <- left_join(df_census_perc, df_pit_spa_hmlss_perc %>% select(SPA, Perc_Homeless), by="SPA"
)

#print(names(merged_df))

# Create wide format df
df_lr_wide_all <- merged_df

# Remove rows where SPA is NA
df_lr_wide_all <- df_lr_wide_all[!is.na(df_lr_wide_all$SPA), ]

# Check for unusable values
null_count <- sum(sapply(df_lr_wide_all, function(x) any(is.null(x))))
#print(null_count)

na_count <- sum(sapply(df_lr_wide_all, function(x) any(is.na(x))))
#print(na_count)

na_indices <- which(is.na(df_lr_wide_all), arr.ind = TRUE)
#print(na_indices)

# Create long format df
df_lr_long_all <- df_lr_wide_all %>%
  pivot_longer(
    cols = -c(SPA, Perc_Homeless),  # Exclude these columns from pivoting
    names_to = "Demographic",            # The name of the new column for the labels
    values_to = "Perc_Demographic"      # The name of the new column for the values
  )

# Check for unusable values
null_count <- sum(sapply(df_lr_long_all, function(x) any(is.null(x))))
#print(null_count)

na_count <- sum(sapply(df_lr_long_all, function(x) any(is.na(x))))
#print(na_count)

print("DONE")
```

EDA Dependent Variable

```
df <- df_lr_wide_all

#print(names(df))

ggplot(df, aes(x = as.factor(SPA), y = Perc_Homeless, fill = as.factor(SPA))) +
  geom_col() +
  labs(
    title = "Homeless Ratio by SPA",
    x = "Service Planning Area (SPA)",
    y = "Homeless Ratio",
    fill = "SPA"  # This will set the legend title
  ) +
  scale_x_discrete(breaks = levels(as.factor(df$SPA))) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5),
    legend.title = element_text(size = 12) # This can further adjust the legend title appearance if n
```

```
eeded
  )

# Write data to a local file for offline use
write.csv(df_lr_wide_all, "datasets/output_df_lr_wide_all.csv", row.names = FALSE)

print("DONE")
```

EDA Independent Variables

```
# Summary Counts of Independent varaiables
df_census_spa_wide <- df_census_spa

# Convert to long format
df_census_spa_long <- df_census_spa_wide %>%
  pivot_longer(
    cols = -c(SPA, Tract),
    names_to = "Variable",
    values_to = "Value"
  )

df_summary <- df_census_spa_long %>%
  group_by(Variable) %>%
  summarise(
    Count = n(),
    Min = min(Value, na.rm = TRUE),
    Mean = mean(Value, na.rm = TRUE),
    Max = max(Value, na.rm = TRUE)
  )

# Print the summary table for the paper
#print(df_summary)

###########################################################################

#Plot of variance of independent variables

df_lr_long_all <- df_lr_wide_all %>%
  pivot_longer(
    cols = -c(SPA, Perc_Homeless),
    names_to = "Demographic",
    values_to = "Perc_Demographic"
  )

df <- df_lr_long_all

demographic_levels <- levels(factor(df$Demographic))
df$Demographic_Index <- as.integer(factor(df$Demographic))

# Generate the boxplot with Demographic indices as x-axis labels
p <- ggplot(df, aes(x = as.factor(Demographic_Index), y = Perc_Demographic)) +
  geom_boxplot() +
  labs(
    title = "Variance of Demographics by Community (SPA)",
    x = "Demographic Index", # Set the x-axis title
    y = "Homeless Ratios"
  ) +
  theme(
    axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
    plot.title = element_text(hjust = 0.5), # Center the plot title
    # Do not remove the x-axis title
```

```
    axis.ticks.x = element_blank()
  )

# Print the plot
#print(p)

# Save the plot
ggsave("images/demographics_boxplot.png", plot = p, width = 12, height = 8, dpi = 300)

print("DONE")
```

Independent variable reduction

```
# Remove "SPA" and "Perc_Homeless" from the dataset for variance calculation
df_eda <- df_lr_wide_all[, !(names(df_lr_wide_all) %in% c("SPA", "Perc_Homeless"))]

# Measure the IQR
iqr_values <- apply(df_eda, 2, IQR)

# Set an IQR threshold
#iqr_threshold <- median(iqr_values)
iqr_threshold <- quantile(iqr_values, 0.75)  # Use a higher percentile for a stricter threshold

# Select variables with an IQR above the threshold
high_iqr_vars <- names(iqr_values[iqr_values > iqr_threshold])

# Include "SPA" and "Perc_Homeless" back into the list of selected variables
final_vars <- c(high_iqr_vars, "SPA", "Perc_Homeless")

additional_vars <- c(
                     )

# Combine the lists
all_vars <- unique(c(final_vars, additional_vars))

df_lr_wide_reduced <- df_lr_wide_all[, all_vars]

#print(names(df_lr_wide_reduced))

#print(df_lr_wide_reduced)

# Define list of variables to remove from the final dataset
manual_removal_list <- c( )
df_lr_wide_reduced <- df_lr_wide_reduced[, !(names(df_lr_wide_reduced) %in% manual_removal_list)]

#print(df_lr_wide_reduced)
#print(names(df_lr_wide_reduced))

df_lr_long_reduced <- df_lr_wide_reduced %>%
  pivot_longer(
    cols = -c(SPA, Perc_Homeless),
    names_to = "Demographic",
    values_to = "Perc_Demographic"
  )

#print(df_lr_long_reduced)

print("DONE")
```

Replot the reduced set of variables

```
df <- df_lr_long_reduced %>%
  select(-SPA)

demographic_levels <- levels(factor(df$Demographic))
df$Demographic_Index <- as.integer(factor(df$Demographic))

# Generate the boxplot with Demographic indices as x-axis labels
p <- ggplot(df, aes(x = as.factor(Demographic_Index), y = Perc_Demographic)) +
  geom_boxplot() +
  labs(
    title = "Variance of Demographics by Community (SPA)",
    x = "Demographic Index",
    y = "Homeless Ratios"
  ) +
  theme(
    axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
    plot.title = element_text(hjust = 0.5),
    axis.title.x = element_blank(),
    axis.ticks.x = element_blank()
  )

# Print the plot to the screen
#print(p)

# Save the plot
ggsave("images/demographics_boxplot_reduced.png", plot = p, width = 12, height = 8, dpi = 300)

# Create a data frame for the legend
legend_table <- data.frame(
  Demographic_Index = 1:length(demographic_levels),
  Demographic_Name = demographic_levels
)

# Print the legend table
#print(legend_table)

# Save to a CSV
write.csv(legend_table, "datasets/Demographic_Index_Legend.csv", row.names = FALSE)

print("DONE")
```

Spearman's Correlation

```
library(ggplot2)
library(corrplot)
library(dplyr)

df_spearman <- df_lr_wide_reduced %>%
  select(-SPA)

# Create index numbers for the columns
index_numbers <- seq_along(df_spearman)

# Update column names to index numbers
original_names <- names(df_spearman)
names(df_spearman) <- index_numbers

# Calculate Spearman
spearman_cor <- cor(df_spearman, method = "spearman", use = "complete.obs")
```

```r
# Plot the matrix with index numbers for labels
corrplot(spearman_cor, method = "color", type = "upper", order = "hclust",
         addCoef.col = "black",
         tl.cex = 0.8,
         tl.srt = 45,
         tl.col = "black",
         number.cex = 0.5,
         cl.cex = 0.8
)

# Save the plot
ggsave("images/spearman_correlation_matrix.png", width = 12, height = 8, dpi = 300)


# Correct map
index_to_name_mapping <- setNames(as.character(index_numbers), original_names)

# Print the mapping
#print(index_to_name_mapping)


index_name_df <- data.frame(Index = names(index_to_name_mapping),
                            Name = index_to_name_mapping,
                            stringsAsFactors = FALSE)

# Write to a CSV
write.csv(index_name_df, "datasets/output_index_to_name_mapping.csv", row.names = FALSE)

###### REMOVE CORELATED FEATURES############

columns_to_remove <- c(
"Perc_Count__HOUSING.OCCUPANCY__Total.housing.units_sum",
"Perc_Count__HOUSING.OCCUPANCY__Total.housing.units__Occupied.housing.units_sum",

"Perc_Count__RELATIONSHIP__Total.population__In.households__Householder_sum",
"Perc_Count__RELATIONSHIP__Total.population__In.households__Child..2._sum",
"Perc_Count__RELATIONSHIP__Total.population__In.households__Opposite.sex.spouse_sum",
"Perc_Count__RELATIONSHIP__Total.population__In.households__Child..2.__Under.18.years_sum",
"Perc_Count__RELATIONSHIP__Total.population__In.households__Grandchild_sum",
"Perc_Count__RELATIONSHIP__Total.population__In.households__Other.relatives_sum",
"Perc_Count__RELATIONSHIP__Total.population__In.households__Nonrelatives_sum",
"Perc_Count__HOUSEHOLDS.BY.TYPE__Total.households_sum",

"Perc_Count__HOUSEHOLDS.BY.TYPE__Total.households__Married.couple.household_sum",
"Perc_Count__HOUSEHOLDS.BY.TYPE__Total.households__Male.householder..no.spouse.or.partner.present._su
m",
"Perc_Count__HOUSEHOLDS.BY.TYPE__Total.households__Male.householder..no.spouse.or.partner.present.__L
iving.alone_sum",
"Perc_Count__HOUSEHOLDS.BY.TYPE__Total.households__Female.householder..no.spouse.or.partner.present._
sum",
"Perc_Count__HOUSEHOLDS.BY.TYPE__Total.households__Female.householder..no.spouse.or.partner.present._
_Living.alone_sum",


"Perc_Count__RACE__Total.population__One.Race_sum",
"Perc_Count__RACE__Total.population__Two.or.More.Races_sum",
"Perc_Count__RACE__Total.population__One.Race__Some.Other.Race_sum",

"Perc_Count__HISPANIC.OR.LATINO__Total.population__Hispanic.or.Latino..of.any.race._sum",
"Perc_Count__HISPANIC.OR.LATINO.BY.RACE__Total.population__Hispanic.or.Latino__Some.Other.Race.alone_
```

```
sum",
"Perc_Count__HISPANIC.OR.LATINO.BY.RACE__Total.population__Hispanic.or.Latino__Two.or.More.Races_sum"
,
"Perc_Count__HISPANIC.OR.LATINO.BY.RACE__Total.population__Hispanic.or.Latino__White.alone_sum",
"Perc_Count__HISPANIC.OR.LATINO.BY.RACE__Total.population__Hispanic.or.Latino_sum",
"Perc_Count__HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino__Asian.alone_sum",
"Perc_Count__HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino__Black.or.African.A
merican.alone_sum",
"Perc_Count__HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino__White.alone_sum",
"Perc_Count__HISPANIC.OR.LATINO.BY.RACE__Total.population__Not.Hispanic.or.Latino_sum",

"Perc_Count__SEX.AND.AGE__Male.population__Selected.Age.Categories__18.years.and.over_sum",
"Perc_Count__SEX.AND.AGE__Total.population__Selected.Age.Categories__16.years.and.over_sum",
"Perc_Count__SEX.AND.AGE__Total.population__Selected.Age.Categories__18.years.and.over_sum",
"Perc_Count__SEX.AND.AGE__Total.population__Selected.Age.Categories__21.years.and.over_sum",
"Perc_Count__SEX.AND.AGE__Total.population__Selected.Age.Categories__62.years.and.over_sum",
"Perc_Count__SEX.AND.AGE__Total.population__Selected.Age.Categories__65.years.and.over_sum",
"Perc_Count__SEX.AND.AGE__Male.population__Selected.Age.Categories__16.years.and.over_sum",
"Perc_Count__SEX.AND.AGE__Male.population__Selected.Age.Categories__21.years.and.over_sum",
"Perc_Count__SEX.AND.AGE__Female.population__Selected.Age.Categories__21.years.and.over_sum",

"Perc_Count__SEX.AND.AGE__Total.population__Over_19_sum"
)

df_lr_wide_reduced_cln <- df_lr_wide_reduced %>%
  select(-any_of(columns_to_remove))

columns_to_add_back<- c(
#"Perc_Count__VACANCY.RATES__Homeowner.vacancy.rate..percent...4._mean",
"Perc_Count__VACANCY.RATES__Rental.vacancy.rate..percent...5._mean"
)

df_lr_wide_reduced_cln <- df_lr_wide_reduced_cln %>%
  bind_cols(df_lr_wide_all %>% select(all_of(columns_to_add_back)))

spearman_reduced_cln <- df_lr_wide_reduced_cln %>%
  select(-SPA)

# Create index
index_numbers <- seq_along(spearman_reduced_cln)

# Update column names
original_names <- names(spearman_reduced_cln)
names(spearman_reduced_cln) <- index_numbers

# Calculate Spearman
spearman_cor <- cor(spearman_reduced_cln, method = "spearman", use = "complete.obs")

# Plot the  matrix
corrplot(spearman_cor, method = "color", type = "upper", order = "hclust",
         addCoef.col = "black",
         tl.cex = 0.8,
         tl.srt = 45,
         tl.col = "black",
         number.cex = 0.5,
         cl.cex = 0.8
)

index_to_name_mapping <- setNames(original_names, index_numbers)

# Print the mapping
```

```r
#print(index_to_name_mapping)


index_name_df <- data.frame(Index = names(index_to_name_mapping),
                            Name = index_to_name_mapping,
                            stringsAsFactors = FALSE)

# Write this data frame to a CSV file
write.csv(index_name_df, "datasets/output_index_to_name_mapping_cln.csv", row.names = FALSE)

# Save the plot
ggsave("images/spearman_correlation_matrix_cln.png", width = 12, height = 8, dpi = 300)

# Write this data frame to a CSV file
write.csv(df_lr_wide_reduced_cln, "datasets/output_df_lr_wide_reduced_cln.csv", row.names = FALSE)


print("DONE")
```

Diagnostic Plots

```r
df_plots <- df_lr_wide_reduced_cln %>%
    select(-SPA)

lm_model <- lm(Perc_Homeless ~ ., data = df_plots)

#summary(lm_model)

# Create  diagnostic plots
par(mfrow=c(2,2))
plot(lm_model)

# Remove  outliers based on their row numbers

df_lr_b <- df_lr_wide_reduced_cln %>%
    select(-SPA)

beta_model <- betareg(Perc_Homeless ~ .,data = df_lr_b)

#summary(beta_model)

# Create the  plots
par(mfrow=c(2,2))
plot(beta_model)

print("DONE")
```

ANOVA

```r
df_anova <- df_lr_long_reduced_cln %>%
    select(-SPA)

str(df_anova)

anova_result <- aov(Perc_Homeless ~ Perc_Demographic, data = df_anova)
summary(anova_result)

print("DONE")
```

LR Model

```
df_lr <- df_lr_wide_reduced_cln %>%
    select(-SPA)

lm_model <- lm(Perc_Homeless ~ ., data = df_lr)
#summary(lm_model)

print("DONE")
```

Beta Model

```
#print(names(df_lr_wide_reduced_cln))

df_lr_b <- df_lr_wide_reduced_cln %>%
    select(-SPA)

beta_model <- betareg(Perc_Homeless ~ .,data = df_lr_b)

#summary(beta_model)

print("DONE")
```

XGB Model

```
# Simple model

library(xgboost)

# Prepare data
df <- df_lr_wide_reduced_cln[, -which(names(df_lr_wide_reduced_cln) == "SPA")]  # Remove 'SPA' column
labels <- df$Perc_Homeless  # Define labels
data <- df[, -which(names(df) == "Perc_Homeless")]  # Remove labels from data

# Convert to matrix
data_matrix <- xgb.DMatrix(data = as.matrix(data), label = labels)

# Train the model
model <- xgb.train(
  params = list(objective = "reg:squarederror"),
  data = data_matrix,
  nrounds = 10
)

# Output the model
#print(model)
#final_rmse <- cv_results$evaluation_log$test_rmse_mean[length(cv_results$evaluation_log$test_rmse_me
an)]
#final_mae <- cv_results$evaluation_log$test_mae_mean[length(cv_results$evaluation_log$test_mae_mean)
]

#cat("Final RMSE: ", final_rmse, "\n")
#cat("Final MAE: ", final_mae, "\n")





# Add train test splits and learning curves and feature importance

library(xgboost)
```

```r
library(ggplot2)

# Prepare data
df <- df_lr_wide_reduced_cln[, -which(names(df_lr_wide_reduced_cln) == "SPA")]
labels <- df$Perc_Homeless
data <- df[, -which(names(df) == "Perc_Homeless")]

# Train/test split
set.seed(123)
indices <- sample(1:nrow(df), size = 0.5 * nrow(df), replace = FALSE)
train_data <- data[indices, ]
train_labels <- labels[indices]
test_data <- data[-indices, ]
test_labels <- labels[-indices]

# Convert to DMatrix
dtrain <- xgb.DMatrix(data = as.matrix(train_data), label = train_labels, missing = NA)
dtest <- xgb.DMatrix(data = as.matrix(test_data), label = test_labels, missing = NA)

# Parameters for XGBoost
params <- list(objective = "reg:squarederror")

# Watchlist to monitor training and testing data
watchlist <- list(train = dtrain, test = dtest)

# Train the model
final_model <- xgb.train(
  params = params,
  data = dtrain,
  nrounds = 50,  # 10  is the point were I see overfitting
  watchlist = watchlist,
  verbose = 0,  # Change to 1 to see output in console
  print_every_n = 10,
  early_stopping_rounds = 10
)

# Extract RMSE log
evaluation_log <- final_model$evaluation_log

# Prepare the data for plotting
eval_df <- data.frame(
  Iteration = seq_len(nrow(evaluation_log)),
  Train_RMSE = evaluation_log$train_rmse,
  Test_RMSE = evaluation_log$test_rmse
)

# Plotting with ggplot2 using a log scale for RMSE
ggplot(eval_df, aes(x = Iteration)) +
  geom_line(aes(y = Train_RMSE, colour = "Train RMSE")) +
  geom_line(aes(y = Test_RMSE, colour = "Test RMSE")) +
  labs(title = "Training and Testing RMSE per Iteration", x = "Iteration", y = "Log(RMSE)") +
  scale_color_manual(values = c("Train RMSE" = "blue", "Test RMSE" = "red")) +
  scale_y_log10() +  # Applying log scale to y-axis
  theme_minimal()

# Calculate feature importance
importance <- xgb.importance(feature_names = colnames(data), model = final_model)

# Print feature importance
print(importance)
```

```r
# Plot feature importance
xgb.plot.importance(importance)

# Print the final RMSE and MAE from the last iteration of the cross-validation
#final_rmse <- cv_results$evaluation_log$test_rmse_mean[length(cv_results$evaluation_log$test_rmse_me
an)]
#final_mae <- cv_results$evaluation_log$test_mae_mean[length(cv_results$evaluation_log$test_mae_mean)
]

#cat("Final RMSE: ", final_rmse, "\n")
#cat("Final MAE: ", final_mae, "\n")


print("DONE")







# Add Cross Validation

# Prepare data
df <- df_lr_wide_reduced_cln[, -which(names(df_lr_wide_reduced_cln) == "SPA")]
labels <- df$Perc_Homeless
data <- df[, -which(names(df) == "Perc_Homeless")]

# Convert data to DMatrix
data_matrix <- xgb.DMatrix(data = as.matrix(data), label = labels, missing = NA)

# Parameters for XGBoost
params <- list(
  objective = "reg:squarederror",
  eval_metric = "rmse",
  max_depth = 6,
  eta = 0.1
)

# Perform cross-validation
cv_results <- xgb.cv(
  params = params,
  data = data_matrix,
  nrounds = 50,
  nfold = 5,
  showsd = TRUE,
  verbose = 0,
  print_every_n = 10,
  early_stopping_rounds = 10
)

# Train the final model on the full dataset
final_model <- xgb.train(
  params = params,
  data = data_matrix,
  nrounds = 50,
  verbose = 0
)

# Calculate and print feature importance
importance <- xgb.importance(feature_names = colnames(data), model = final_model)
print(importance)
```

```r
xgb.plot.importance(importance_matrix = importance)

# Plotting the learning curves using a log scale for RMSE
eval_df <- data.frame(
  Iteration = seq_len(nrow(cv_results$evaluation_log)),
  Train_RMSE = cv_results$evaluation_log$train_rmse_mean,
  Test_RMSE = cv_results$evaluation_log$test_rmse_mean
)

# Plotting with ggplot2 using a log scale for RMSE
ggplot(eval_df, aes(x = Iteration)) +
  geom_line(aes(y = Train_RMSE, colour = "Train RMSE")) +
  geom_line(aes(y = Test_RMSE, colour = "Test RMSE")) +
  labs(title = "Training and Testing RMSE per Iteration", x = "Iteration", y = "Log(RMSE)") +
  scale_color_manual(values = c("Train RMSE" = "blue", "Test RMSE" = "red")) +
  scale_y_log10() +  # Applying log scale to y-axis
  theme_minimal()

# Print the final RMSE and MAE from the last iteration of the cross-validation
final_rmse <- cv_results$evaluation_log$test_rmse_mean[length(cv_results$evaluation_log$test_rmse_mea
n)]
final_mae <- cv_results$evaluation_log$test_mae_mean[length(cv_results$evaluation_log$test_mae_mean)]

cat("Final RMSE: ", final_rmse, "\n")
cat("Final MAE: ", final_mae, "\n")

print("DONE")



# Add a hyper parameter grid serach
library(xgboost)
library(caret)
library(data.table)

# Prepare data
df <- df_lr_wide_reduced_cln[, -which(names(df_lr_wide_reduced_cln) == "SPA")]  # Remove 'SPA' column
labels <- df$Perc_Homeless
data <- df[, -which(names(df) == "Perc_Homeless")]

# Convert data to a format that caret can use (data frame instead of DMatrix)
train_data <- as.data.frame(data)
train_data$Perc_Homeless <- labels

# Set up training control
train_control <- trainControl(
  method = "cv",
  number = 5,
  verboseIter = TRUE,
  returnData = FALSE,
  returnResamp = "all",
  allowParallel = TRUE
)

# grid of hyperparameters to search for my small dataset
grid <- expand.grid(
  nrounds = c(5, 10, 20),
  max_depth = c(2, 3),
  eta = c(0.01, 0.05),
  gamma = c(0, 0.1),
```

```
  colsample_bytree = c(0.5, 0.7),
  min_child_weight = c(1, 2),
  subsample = c(0.5, 0.8)
)

# Run the model
model <- train(
  Perc_Homeless ~ .,
  data = train_data,
  method = "xgbTree",
  trControl = train_control,
  tuneGrid = grid,
  metric = "RMSE"
)

# Print the best tuning parameters
print(model$bestTune)

# Plot model performance
print(model)

#The final values used for the model were nrounds = 20, max_depth = 3, eta = 0.05, gamma = 0.1, colsa
mple_bytree = 0.5,
#min_child_weight = 1 and subsample = 0.8.

# Print the final RMSE and MAE from the last iteration of the cross-validation
final_rmse <- cv_results$evaluation_log$test_rmse_mean[length(cv_results$evaluation_log$test_rmse_mea
n)]
final_mae <- cv_results$evaluation_log$test_mae_mean[length(cv_results$evaluation_log$test_mae_mean)]

cat("Final RMSE: ", final_rmse, "\n")
cat("Final MAE: ", final_mae, "\n")

print("DONE")



# Build a new model with the parameters from the grid search

# Prepare data
df <- df_lr_wide_reduced_cln[, -which(names(df_lr_wide_reduced_cln) == "SPA")]
labels <- df$Perc_Homeless
data <- df[, -which(names(df) == "Perc_Homeless")]

# Convert data to DMatrix
data_matrix <- xgb.DMatrix(data = as.matrix(data), label = labels, missing = NA)

# Parameters for XGBoost # These plot
params <- list(
  objective = "reg:squarederror",
  eval_metric = "rmse",
  #max_depth = 6,  # Wont print
  max_depth = 3,
  #gamma = 0.1,# Wont print
  eta = 0.1,
  #eta = 0.05 # Wont print
  colsample_bytree = 0.5, # HELPED
  min_child_weight = 1,
  subsample = 0.8
)
```

```r
# Perform cross-validation
cv_results <- xgb.cv(
  params = params,
  data = data_matrix,
  nrounds = 30, # Model was seen overfitting slightly after 40
  nfold = 5,
  showsd = TRUE,
  verbose = 0,
  print_every_n = 10,
  early_stopping_rounds = 10
)

# Train the final model on the full dataset
final_model <- xgb.train(
  params = params,
  data = data_matrix,
  nrounds = 50,
  verbose = 0
)

# Calculate and print feature importance
importance <- xgb.importance(feature_names = colnames(data), model = final_model)
print(importance)

xgb.plot.importance(importance_matrix = importance)

# Plotting the learning curves using a log scale for RMSE
eval_df <- data.frame(
  Iteration = seq_len(nrow(cv_results$evaluation_log)),
  Train_RMSE = cv_results$evaluation_log$train_rmse_mean,
  Test_RMSE = cv_results$evaluation_log$test_rmse_mean
)

# Plotting with ggplot2 using a log scale for RMSE
ggplot(eval_df, aes(x = Iteration)) +
  geom_line(aes(y = Train_RMSE, colour = "Train RMSE")) +
  geom_line(aes(y = Test_RMSE, colour = "Test RMSE")) +
  labs(title = "Training and Testing RMSE per Iteration", x = "Iteration", y = "Log(RMSE)") +
  scale_color_manual(values = c("Train RMSE" = "blue", "Test RMSE" = "red")) +
  scale_y_log10() +
  theme_minimal()

# Print the final RMSE and MAE from the last iteration of the cross-validation
final_rmse <- cv_results$evaluation_log$test_rmse_mean[length(cv_results$evaluation_log$test_rmse_mea
n)]
final_mae <- cv_results$evaluation_log$test_mae_mean[length(cv_results$evaluation_log$test_mae_mean)]

cat("Final RMSE: ", final_rmse, "\n")
cat("Final MAE: ", final_mae, "\n")

# Write data to a local file for offline use
write.csv(importance, "datasets/output_xgb_importance.csv", row.names = FALSE)

print("DONE")

print(final_model$params)

ocv_model$results)
```