

S&DS 625: Housing Project

Matt Shu, Peter Yun, Eugene Han

2024-10-14

Contents

Introduction	1
Data Preprocessing and Exploration	1
Subset Construction	1
Base Features	9
Food Access Features	19
Modeling	30
Model Results	35
Limitations of the Model and Analysis	36
Conclusion	37
Sessioninfo	38

Introduction

In this study, we aim to explore how a home's proximity to food retailers, such as grocery stores and supermarkets, contributes to its value. This is a valuable question to answer from the perspective of someone trying to buy a home; for instance, an individual may be interested in moving with their family to an affordable home while being located within a convenient distance from a grocery store to shop for food. For this case study, we specifically focus on single-family homes in New Haven and consider their total assessed values.

In addition to the parcel dataset for Connecticut properties, we utilize the Connecticut Supplemental Nutrition Assistance Program (SNAP) Authorized Retailers dataset, which contains information such as store type (supermarket, convenience store, farmer's market, etc.) and latitude/longitude coordinates that we use together with the spatial format of the parcel data.

Data Preprocessing and Exploration

Subset Construction

In order to address our study's objective, we take the perspective of a potential homebuyer and restrict our analysis to single-family homes. This came with several challenges: - State use codes that were clearly labeled as single-family homes according to their descriptions were not always reliable. There exist cases where these "single-family" properties actually pointed to vacant lots or other land used for non-residential purposes. Despite this, these codes were generally effective as an initial filter. - State use codes and descriptions were inconsistent across different towns. - Data quality and completeness varied significantly across towns; for

some towns, initial filtering based on state use code would be more effective and lead to less case-by-case inspections of state use descriptions.

```
# Load libraries
library(sf)

## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
library(leaflet)
library(webshot)

# The below is necessary one time to allow knitting
# webshot::install_phantomjs()

# Load CT parcel data
d <- st_read("data/CT-parcel-data/4c5501b8-b68e-4888-bf6a-d92670d69c3b.gdb/")

## Reading layer `CT_Parcels' from data source
##   `/Users/mattshu/Code/2024-25/S&DS625/stat-625-housing-project/data/CT-parcel-data/4c5501b8-b68e-4888-bf6a-d92670d69c3b.gdb'
##   using driver `OpenFileGDB'

## Warning in CPL_read_ogr(dsn, layer, query, as.character(options), quiet, : GDAL
## Message 1: organizePolygons() received a polygon with more than 100 parts. The
## processing may be really slow. You can skip the processing by setting
## METHOD=SKIP, or only make it analyze counter-clock wise parts by setting
## METHOD=ONLY_CCW if you can assume that the outline of holes is counter-clock
## wise defined

## replacing null geometries with empty geometries
## Simple feature collection with 1247506 features and 46 fields (with 45 geometries empty)
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: -8207340 ymin: 5009497 xmax: -7991341 ymax: 5168563
## Projected CRS: WGS 84 / Pseudo-Mercator

# Observe below that state use descriptions are unreliable, but also that
# multiple state use codes can correspond to single family homes
d$State_Use_Description <- toupper(d$State_Use_Description)

# Filter for codes that contain the word FAMILY and could be single-family
d_sf <- d[which(grepl("FAMILY", d$State_Use_Description) &
  !grepl("[2-9]", d$State_Use_Description) &
  !grepl("TWO", d$State_Use_Description) &
  !grepl("THREE", d$State_Use_Description) &
  !grepl("FOUR", d$State_Use_Description) &
  !grepl("FIVE", d$State_Use_Description) &
  !grepl("SIX", d$State_Use_Description) &
  !grepl("MULTI", d$State_Use_Description)), ]

# Display table of mappings between State Use Description and State Use
table(d_sf$State_Use_Description, d_sf$State_Use, useNA = "always")

##
```

	0	101	1010	1011	1013	1070	110
1 FAMILY	0	0	0	0	0	0	0
1 FAMILY ON COM LN	0	0	0	0	0	0	0
1 FAMILY PLANNED COMM	0	0	0	0	0	0	0

##	1 FAMILY WF	0	0	0	0	0	0
##	1-FAMILY	0	0	0	0	0	0
##	HSNG AUTH 1 FAMILY	0	0	0	0	0	0
##	ONE FAMILY	0	18133	2316	0	0	0
##	ONE FAMILY - LAKE REGION	0	0	0	0	0	0
##	ONE FAMILY - LAKEFRONT	0	0	0	0	0	0
##	ONE FAMILY + ACCESSORY UNIT	0	0	0	51	0	0
##	RES SINGLE FAMILY	0	5996	0	0	0	0
##	SINGLE FAMILY OF	0	0	0	0	0	0
##	SINGLE FAMILY RF	0	0	0	0	0	0
##	SINGLE FAMILY WF	0	0	0	0	0	0
##	SINGLE FAMILY	0	220638	82547	1	0	0
##	SINGLE FAMILY - VACANT	0	0	0	0	0	0
##	SINGLE FAMILY GC	0	0	0	0	0	0
##	SINGLE FAMILY GCF	0	0	0	0	0	0
##	SINGLE FAMILY INLAW	0	0	0	0	0	0
##	SINGLE FAMILY OF	0	0	0	0	0	0
##	SINGLE FAMILY PUD	0	0	0	0	0	127
##	SINGLE FAMILY RES	0	19344	0	0	0	0
##	SINGLE FAMILY RF	0	0	0	0	0	0
##	SINGLE FAMILY UNIT	0	0	0	0	0	0
##	SINGLE FAMILY VAC	93	0	0	0	0	0
##	SINGLE FAMILY W/ IN-LAW	0	0	0	0	31	0
##	SINGLE FAMILY WA	0	0	0	0	0	0
##	SINGLE FAMILY WACC	0	0	0	0	0	0
##	SINGLE FAMILY WATER	0	0	0	0	316	0
##	SINGLE FAMILY WF	0	0	0	0	0	0
##	<NA>	0	0	0	0	0	0
##							
##		112	114	115	201	941	<NA>
##	1 FAMILY	0	0	0	0	0	646
##	1 FAMILY ON COM LN	0	0	0	0	0	88
##	1 FAMILY PLANNED COMM	0	0	0	0	0	95
##	1 FAMILY WF	0	0	0	0	0	49
##	1-FAMILY	0	0	0	0	0	2097
##	HSNG AUTH 1 FAMILY	0	0	0	0	61	0
##	ONE FAMILY	0	0	0	0	0	0
##	ONE FAMILY - LAKE REGION	0	0	0	0	0	312
##	ONE FAMILY - LAKEFRONT	0	0	0	0	0	222
##	ONE FAMILY + ACCESSORY UNIT	0	0	0	0	0	0
##	RES SINGLE FAMILY	0	0	0	0	0	0
##	SINGLE FAMILY OF	0	0	0	0	0	65
##	SINGLE FAMILY RF	0	0	0	0	0	163
##	SINGLE FAMILY WF	0	0	0	0	0	57
##	SINGLE FAMILY	0	0	0	0	0	859
##	SINGLE FAMILY - VACANT	0	0	0	0	0	45
##	SINGLE FAMILY GC	0	0	0	0	0	30
##	SINGLE FAMILY GCF	0	4	0	0	0	0
##	SINGLE FAMILY INLAW	0	0	0	0	0	62
##	SINGLE FAMILY OF	205	0	0	0	0	107
##	SINGLE FAMILY PUD	0	0	0	0	0	0
##	SINGLE FAMILY RES	0	0	0	2236	0	0
##	SINGLE FAMILY RF	0	0	0	0	0	100
##	SINGLE FAMILY UNIT	0	0	6	0	0	0

```

## SINGLE FAMILY VAC          0    0    0    0    0    0
## SINGLE FAMILY W/ IN-LAW   0    0    0    0    0    0
## SINGLE FAMILY WA          0    0    0    0    0    63
## SINGLE FAMILY WACC        0    0    0    0    0    172
## SINGLE FAMILY WATER       0    0    0    0    0    0
## SINGLE FAMILY WF          0    0    0    0    0    1286
## <NA>                      0    0    0    0    0    0

```

We observe that the large majority of properties in this subset are captured by state use codes “101” and “1010”, with these mapping to multiple descriptions (One Family, Res. Single Family, Single Family, Single Family Res.) which all have the same meaning. As a result, we can use these codes to form a heuristic to gauge town data quality with respect to how easily we can isolate single-family homes.

```

# View the distribution of single family homes with codes 101 and 1010 in each
# town, looking also at NAs
d$sh <- ifelse(is.na(d$State_Use), NA,
               ifelse(d$State_Use %in% c("101", "1010"), "Single", "Others"))

# Comment out for brevity
# table(d$sh, d$Town_Name, useNA = "always")

```

The distribution of NAs and how many single family homes are captured by the two codes depends a lot on the town, although it's important to note that the percentage of single family homes in different Connecticut towns also varies.

Going forward, let's zoom into a particular town/city: New Haven. Based on the table, 9168 properties have state use code “101” or “1010”, 2968 are missing a state use code, and 15242 fall in the “Others” bin. Clearly, many candidate single-family homes are captured by the two state use codes, but there are also more than 18000 properties that we need to investigate further to identify other single-family homes. In addition to requiring a closer look, New Haven is also interesting with respect to our study because parts of the city are considered food deserts.

```

# Subset the original dataset to only New Haven
dnh <- d[d$Town_Name == "NEW HAVEN", ]

# Some initial observations show the many codes and descriptions, we comment
# this out to reduce clutter in the output
# table(dnh$State_Use)
# table(dnh$State_Use_Description)
# table(dnh$State_Use, dnh$State_Use_Description, useNA = "always")

```

We repeat the same analysis above looking at the descriptions with FAMILY and observe that in New Haven, descriptions containing FAMILY that do not refer to multi-family homes are all captured by description SINGLE FAMILY and all of these codes are 1010.

```

nh_sf <- dnh[which(grepl("FAMILY", dnh$State_Use_Description) &
                     !grepl("[2-9]", dnh$State_Use_Description) &
                     !grepl("TWO", dnh$State_Use_Description) &
                     !grepl("THREE", dnh$State_Use_Description) &
                     !grepl("FOUR", dnh$State_Use_Description) &
                     !grepl("FIVE", dnh$State_Use_Description) &
                     !grepl("SIX", dnh$State_Use_Description) &
                     !grepl("MULTI", dnh$State_Use_Description)), ]
table(nh_sf$State_Use, nh_sf$State_Use_Description, useNA = "always")

##
##      SINGLE FAMILY <NA>

```

```

##   1010      9168      0
## <NA>          0      0

```

Moreover, we also observe that 101x codes correspond to single-family homes, with the last digit indicating some type of accessory/special feature. As a result, we can create an initial filter for candidate single-family homes and focus on the remaining state use codes/descriptions.

```

# Create state use code-based filter checking if state use code starts with 101
sfh_coded <- dnh[which(startsWith(dnh$State_Use, "101")), ]
table(sfh_coded$State_Use, sfh_coded$State_Use_Description, useNA = "always")

```

```

##
##           SFR IN-LAW SFR RIVERFRONT SFR WATER SINGLE FAMILY <NA>
##   1010          0          0          0      9168      0
##   1012          26         0          0          0      0
##   1013          0          0          39         0      0
##   1015          0          13         0          0      0
## <NA>          0          0          0          0      0

```

While we were able to confidently skip over properties with certain state use descriptions such as “APT 4-UNIT”, “PARK GAR MDL-96”, and “YACHT CLUB”, others weren’t as clear. For instance, some descriptions are prefixed by “HSNG AUTH” or “CITY”, indicating ownership by a housing authority or the city of New Haven, with “HSNG AUTH MDL-01” and “CITY MDL-01” corresponding to single-family homes based on manual inspection and cross-referencing with websites like Zillow and Redfin. On the other hand, “MDL” suffixes other than “MDL-01” did not correspond to single-family homes. As a result, we can construct another loose filter for additional candidate single-family homes based on the inclusion of “MDL-01” in the state use description column.

```

# Create state use description-based filter checking for inclusion of "MDL-01"
# Exclude apartments
mdl_01 <- dnh[which(grep("MDL-01", dnh$State_Use_Description) &
                     !grep("APT", dnh$State_Use_Description)), ]

# Combine with state use code-based candidates
sfh <- rbind(sfh_coded, mdl_01)
dim(sfh)

```

```

## [1] 9425    48

```

It’s important to note that our methods for isolating single-family homes, while motivated by meticulous exploration and cross-referencing properties, is based on heuristics. In particular, our “MDL-01” filter may have introduced some properties that shouldn’t be considered single-family either due to data quality issues or the fact that it wasn’t practical to spot-check every single property with state use description containing “MDL-01”.

To address this, we will run a few sanity checks on several variables and investigate extreme cases.

```

# Explore extremes of assessed total
head(st_drop_geometry(sfh[order(sfh$Assessed_Total, decreasing = T),
                           c("Link", "Owner", "Location", "Assessed_Total",
                             "State_Use_Description"))], 12)

##
##           Link                               Owner
## 777573 52070-392 1188 00101             CITY OF NEW HAVEN PARK
## 769380 52070-244 0365 00500             YALE UNIVERSITY
## 761570 52070-125 1039 01341             HOUSING AUTHORITY OF NEW HAVEN
## 768352 52070-251 1072 00100             CT CLINICAL SERVICES INC
## 758051 52070-094 0999 02600             NEW HAVEN MONTHLY MEETING OF

```

```

## 767998 52070-218 0494 00100          ELLIS CHARLES D & LORIMER LINDA
## 768103 52070-219 0458 01900 BROWN INVESTMENT ADVISORY & TRUST COMPANY TR
## 768067 52070-218 1075 00100          ALEXANDER MARTHA O
## 768367 52070-251 1073 00400          GREWAL ELENA
## 781176 52070-033 0868 00100          198 COVE STREET LMTD PARTNER
## 768364 52070-251 1073 00100          CARTER LOUISE B EST
## 758711 52070-102 1030 00100          CITY OF NEW HAVEN GOLF
##           Location Assessed_Total State_Used_Description
## 777573    358 SPRINGSIDE AV      15926750   REC FACIL MDL-01
## 769380     43 HILLHOUSE AV      3682000    PVT UNIV MDL-01
## 761570    1435 QUINNIPAC AV      2400230    HSNG AUTH MDL-01
## 768352     787 PROSPECT ST      1793960    SINGLE FAMILY
## 758051    223 EAST GRAND AV      1682030    RELIGIOUS MDL-01
## 767998     55 HIGHLAND ST      1549940    SINGLE FAMILY
## 768103    352 SAINT RONAN ST      1548820    SINGLE FAMILY
## 768067     123 EDGEHILL RD      1520890    SINGLE FAMILY
## 768367     130 EDGEHILL RD      1502970    SINGLE FAMILY
## 781176      198 COVE ST      1437380    SFR WATER
## 768364     100 EDGEHILL RD      1422540    SINGLE FAMILY
## 758711     35 EASTERN ST      1403780    REC FACIL MDL-01

tail(st_drop_geometry(sfh[order(sfh$Assessed_Total, decreasing = T),
                           c("Link", "Owner", "Location", "Assessed_Total",
                             "State_Used_Description")]), 12)

```

	Link	Owner	Location
## 775605	52070-325 0516 00900	SGH PROPERTY GROUP LLC	26 CHERRY ANN ST
## 771583	52070-304 0046 03700	BETTER HAVEN LLC	90 ADELINE ST
## 766915	52070-168 0781 03200	LANE LOSER J	117 FILLMORE ST
## 766544	52070-302 0072 00800	ZARAGOZA LUIS	31 FRANK ST
## 770929	52070-275 0033 03200	CHONA ARMANDO JR	73 MORRIS ST
## 759470	52070-126 1051 01200	SSG PROPERTIES LLC	242 BARNES AV
## 764114	52070-275 0030 00600	GANT MARITZA	124 ROSETTE ST
## 764533	52070-265 0058 00100	MEANY WILLIAM H	146 CARLISLE ST
## 777012	52070-342 0152 00500	NATIONAL CONSTRUCTION LLC	6 EVERGREEN CT
## 762032	52070-152 0844 00401	TYLER LARK	6 WELTON ST
## 757827	52070-090 0998 00800	MORENO JIMMY	47 REVERE ST
## 780897	52070-035 0854 00700	GOFF DAVID	287 LIGHTHOUSE RD
	Assessed_Total	State_Used_Description	
## 775605	51240	SINGLE FAMILY	
## 771583	51170	SINGLE FAMILY	
## 766915	49700	SINGLE FAMILY	
## 766544	48370	CITY MDL-01	
## 770929	48160	SINGLE FAMILY	
## 759470	47110	SINGLE FAMILY	
## 764114	44870	SINGLE FAMILY	
## 764533	42210	SINGLE FAMILY	
## 777012	42000	SINGLE FAMILY	
## 762032	41370	SINGLE FAMILY	
## 757827	40670	SINGLE FAMILY	
## 780897	40250	SINGLE FAMILY	

Examining the higher end of assessed total values, “358 SPRINGSIDE AV” stands out with an assessed total of almost \$16 million. This is unrealistic and is confirmed to be a park/school area with 3 buildings; it appears that the data regarding the entire plot of land has incorrectly been attributed to the first of the three

buildings, which indeed is a single-family home, based on cross-referencing the Vision Appraisal database. Similarly, “223 EAST GRAND AV” corresponds to a plot of land with a collection of 3 buildings belonging to a religious society. Therefore, these 2 properties shouldn’t be considered in our subset of single-family homes.

The lower end of assessed total values, on the other hand, were confirmed through Vision Appraisal and appears to be in part due to severe depreciation based on their “effective year built” measures.

```
# Occupancy may give us a hint, expectation is 1 if all data is accurate


```

```
##
##      0     1     2     3     4     5     6     7     8     9     10    12    14    15    16    24
##      1 9275   94   24     5     5     2     2     2     2     2     2     1     1     1     1
## <NA>
##      5
```

```
# Check 0 case - outbuilding at 115 CRANSTON ST, valid
st_drop_geometry(sfh[which(sfh$Occupancy == 0), c("Owner", "Location")])
```

```
##
##           Owner          Location
## 761786 SKOMRO BARBARA M 115 CRANSTON ST
# Check NA cases - all confirmed to be single-family
st_drop_geometry(sfh[is.na(sfh$Occupancy), c("Owner", "Location")])
```

```
##
##           Owner          Location
## 756973  BALLARD TANYA R 124 OAKLEY ST
## 761741    BAILEY ALVAN 603 MIDDLETOWN AV
## 765422 ASSIOBO FAROUZ 136 ROSETTE ST
## 769017 BRIONES STEPHANIE 70 WOOLSEY ST
## 773391 COLUMBUS HOUSE INC 324 WEST DIVISION ST
```

```
# Check >= 4 cases, commented out to avoid long output
# Properties with a lot of occupants are generally rooming houses
# Multiple tenants, but the home itself is a single-family model according to
# Vision Appraisal
# st_drop_geometry(sfh[which(sfh$Occupancy >= 4), ])
```

We also checked for any duplicate listings and found that there were two properties at each of “1 RESERVOIR ST” and “115 CRANSTON ST”. When cross-referenced with Vision Appraisal, we found that the two properties located at “1 RESERVOIR ST” correspond to two separate single-family homes. In contrast, the two properties at “115 CRANSTON ST” refer to the same home, based on the photos provided by Vision Appraisal, with one of the entries containing incorrect information (0s for all room-related columns) which should be removed.

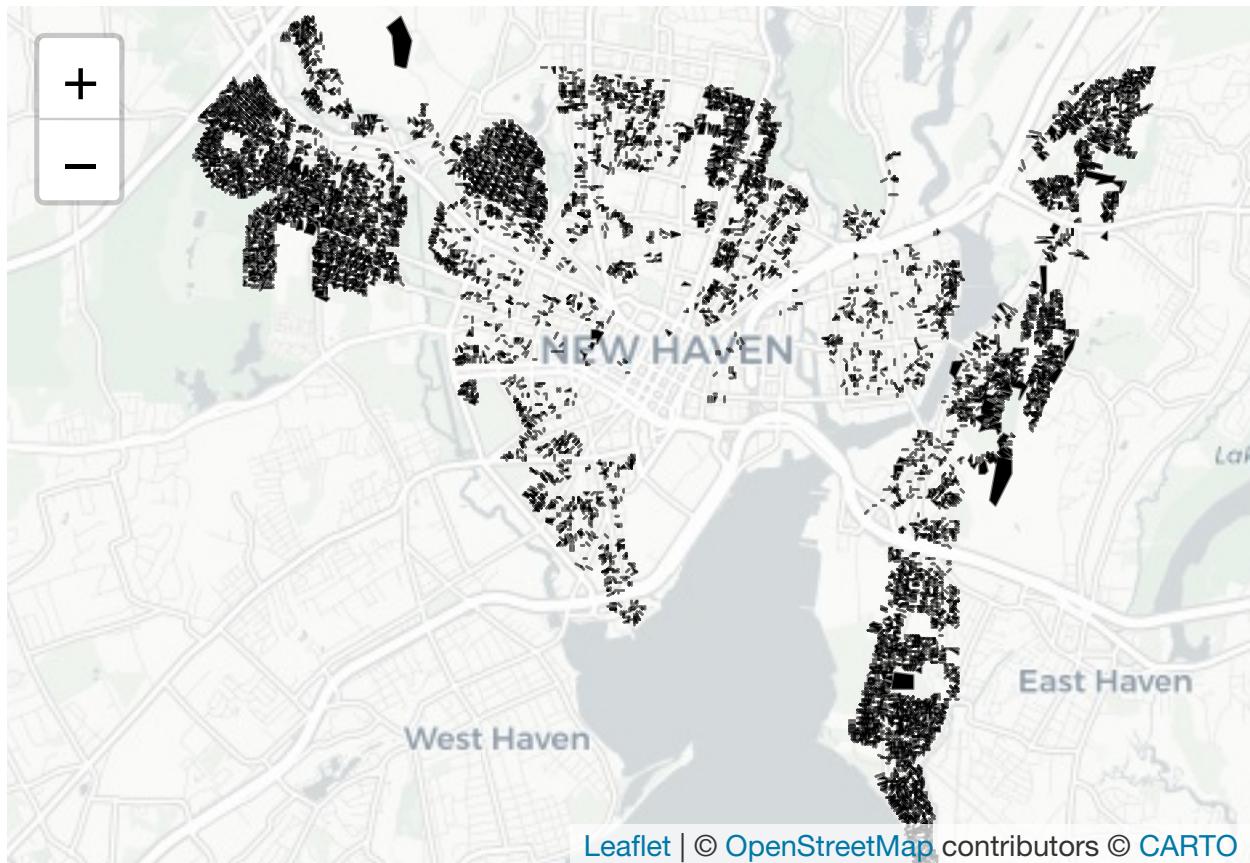
```
# List locations with more than one property
sfh[which(duplicated(sfh$Location)), ]$Location
```

```
## [1] "1 RESERVOIR ST"  "115 CRANSTON ST"
# Inspect corresponding properties, commented out for brevity
# sfh[which(sfh$Location %in% c("1 RESERVOIR ST", "115 CRANSTON ST")), ]
```

As a sanity check, we also plotted the properties superimposed on a map of New Haven leveraging their shapefiles. We then inspected any properties that were unusually large and checked Vision Appraisal, revealing several more areas of land with multiple buildings’ assessments represented by only one of the properties.

```
# Convert to latitude/longitude
sfh <- st_transform(sfh, crs = 4326)
```

```
# Map plot
leaflet() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  
  # Center on New Haven
  setView(lng = -72.93, lat = 41.30, zoom = 12) %>%
  
  # Add sfh homes
  addPolygons(data = sfh,
              color = "black",
              weight = 1,
              fillColor = "black",
              popup = ~paste("Location:", Location,
                            "<br>State Use: ", State_Use_Description),
              group = "Homes")
```



```
# Remove properties that aren't single-family and incorrect duplicate
to_exclude <- c("52070-392 1188 00101", "52070-094 0999 02600",
             "52070-033 0868 00900", "52070-102 1030 00100",
             "52070-280 0249 02701", "52070-142 1060 01400",
             "52070-125 1039 01341", "52070-098 1001 00200",
             "52070-024 0920 02800")
sfh_final <- sfh[which(!(sfh$link %in% to_exclude)), ]
```

Base Features

We consider the following variables:

1. **Effective Area** - By accounting for different parts of a property (e.g. a garage or porch) in addition to living area, effective area better captures the contributory value of these property features than living area alone. We would expect that properties with larger effective areas tend to have higher total assessed values.
2. **Number of Bedrooms** - The number of bedrooms is related to the amount of living space in a home, which in turn we expect to be associated with a home's assessed value.
3. **Total Number of Bathrooms** - Like the number of bedrooms, the number of bathrooms is also related to the amount of living space in a home. To get a single total, we define Total Bathrooms as `Number_of_Baths + 0.5 * Number_of_Half_Baths`.
4. **Condition** - Property condition categories (e.g., Excellent, Poor) are expected to show variation in housing prices, with “worse” conditions being associated with lower assessed values.
5. **Zone** - Different zoning regulations (e.g., residential, commercial) that affect property use, including permissions for commercial activities like restaurants, can influence assessed value.
6. **Effective Age** - Effective year built (EYB), unlike actual year built (AYB), accounts for home improvements. We can create a variable for Effective Age = (2024 - EYB) which is more intuitive, where we expect an “older” home to have a lower assessed value from depreciation due to lack of renovations/modernization.

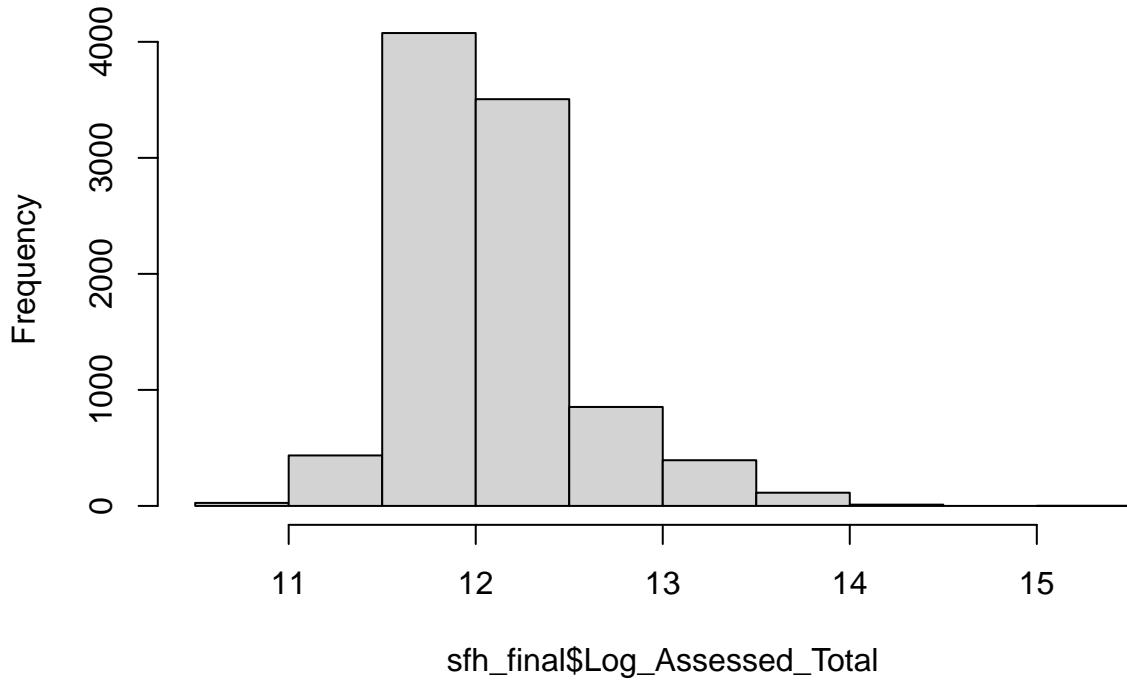
Since we're dealing with price data in our response, we will work with the logarithm of assessed total value.

```
# Check for missing values
sum(is.na(sfh_final$Assessed_Total))

## [1] 0

# Create Log_Assessed_Total variable and plot histogram as sanity check
# Histogram looks reasonable
sfh_final$Log_Assessed_Total <- log(sfh_final$Assessed_Total)
hist(sfh_final$Log_Assessed_Total)
```

Histogram of sfh_final\$Log_Assessed_Total

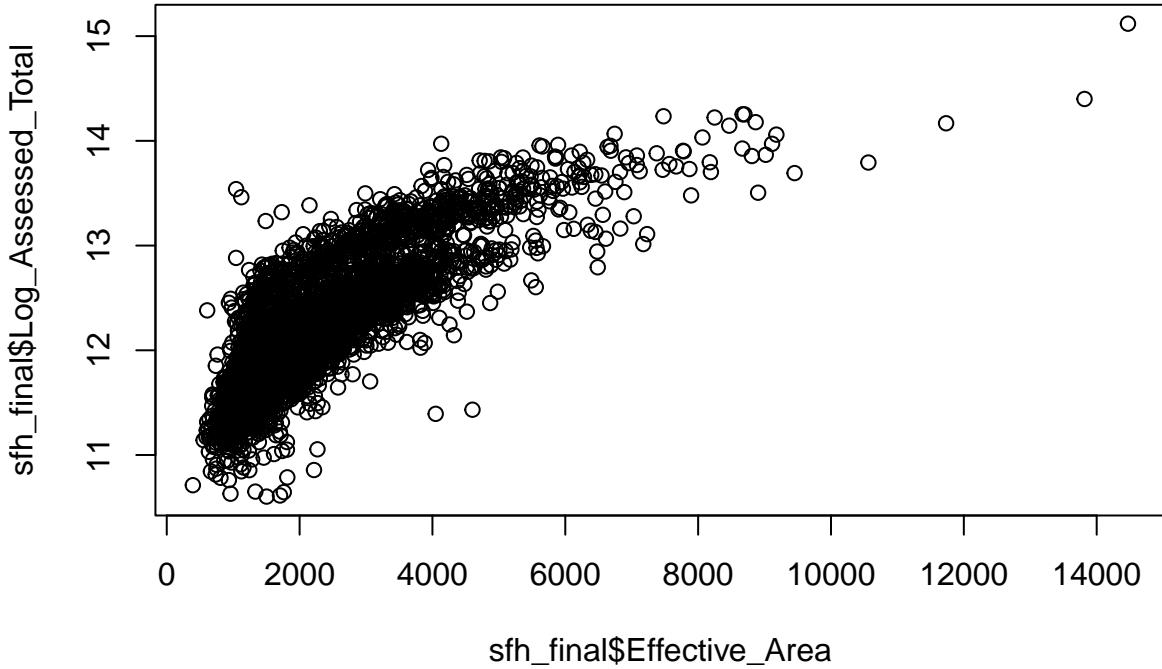


Next, we'll take a look at Effective Area. When plotting vs. log assessed total, we see that using a square root transformation results in a more linear relationship in the scatterplot while still having a natural interpretation.

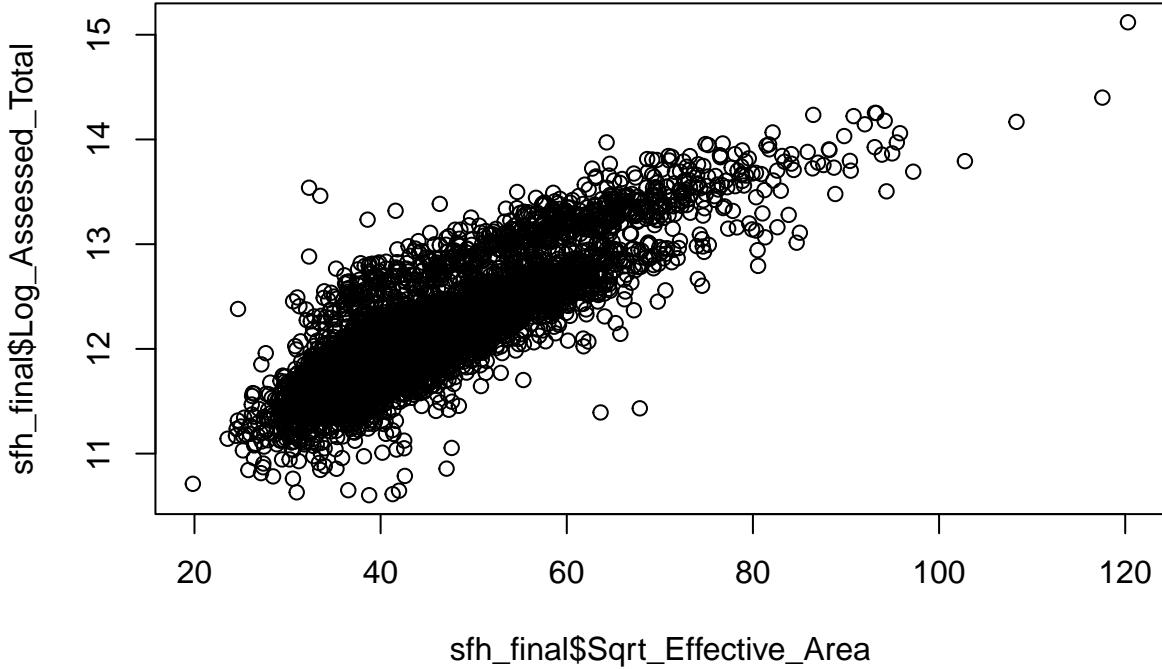
```
# Check for missing values
sum(is.na(sfh_final$Effective_Area))

## [1] 0

# Plot vs. Log Assessed Total, untransformed
plot(sfh_final$Effective_Area, sfh_final$Log_Assessed_Total)
```



```
# Has some curvature, try square root transformation
# More linear relationship
sfh_final$Sqrt_Effective_Area <- sqrt(sfh_final$Effective_Area)
plot(sfh_final$Sqrt_Effective_Area, sfh_final$Log_Assessed_Total)
```



We observe that there are a number of bedrooms that are equal to 0 and has NA values. We inspect further to find out what these properties are.

Since properties with 0 number of bedrooms are small, we manually check every property. After cross-checking with Vision Appraisal, it appears that some missing values have been converted to 0. We fix the values for these properties as we expect single-family homes to have at least 1 bedroom unless they are studios such as the properties located at “34 BURTON ST” and “132 COVE ST”, using sites like Zillow, Redfin, and Realtor

as references.

```
# Get table of bedroom counts
table(sfh_final$Number_of_Bedroom, useNA = "always")

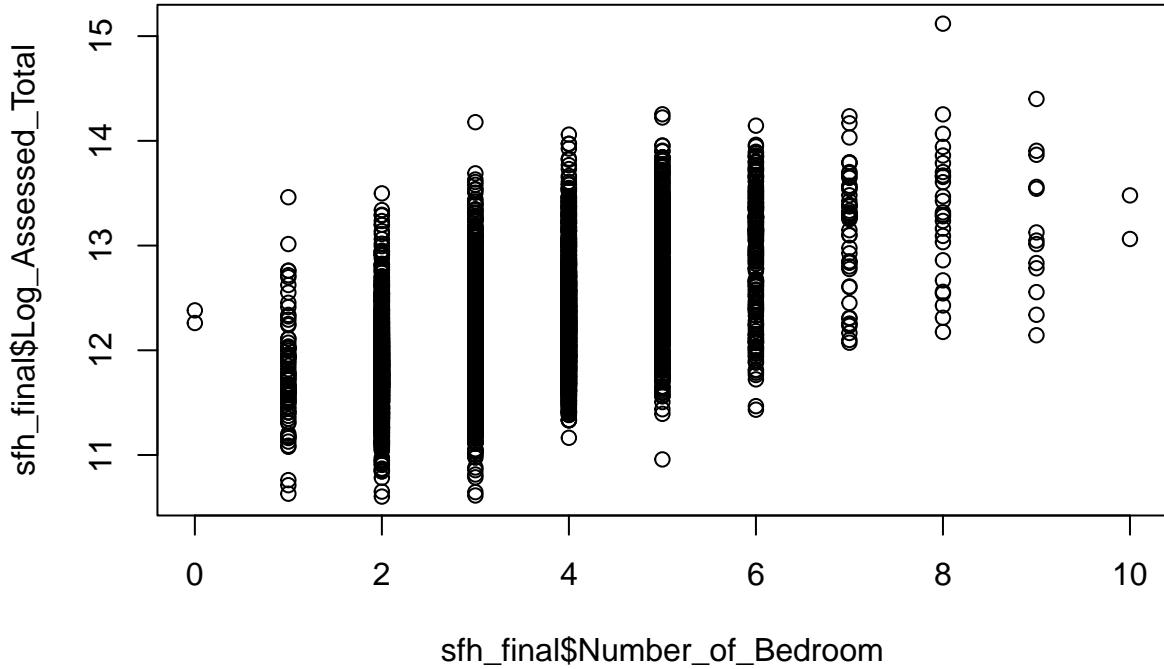
##          0      1      2      3      4      5      6      7      8      9      10 <NA>
##      5    89 1585 5224 1747   477   195    52    26    13     2     1

# Identify properties with 0 or missing number of bedrooms
st_drop_geometry(sfh_final[which(sfh_final$Number_of_Bedroom == 0 |
                                is.na(sfh_final$Number_of_Bedroom)),
                           c("Owner", "Location", "Number_of_Bedroom",
                             "Number_of_Baths", "Number_of_Half_Baths")])

##           Owner      Location Number_of_Bedroom
## 778843 STEVENS MARGARET J 34 BURTON ST        NA
## 780965 HELLYAR E MICHAEL 132 COVE ST         0
## 762212 DASCANIO JOSEPH 532 CHAPEL ST         0
## 771060 CONNECTION FUND INC 98 PARK ST         0
## 771919 GARLAND TRUST ASSOCIATION INC 348 ELM ST        0
## 773938 SUB-SCRIBE ASSOCIATES LLC 445 GEORGE ST        0
##       Number_of_Baths Number_of_Half_Baths
## 778843            1                  0
## 780965            0                  1
## 762212            2                  0
## 771060            0                  0
## 771919            3                  0
## 773938            0                  0

bedroom_rows <- row.names(sfh_final[which(sfh_final$Number_of_Bedroom == 0 |
                                           is.na(sfh_final$Number_of_Bedroom)), ])
corrected_bedrooms <- c(0, 0, 1, 3, 6, 4)
sfh_final[bedroom_rows, "Number_of_Bedroom"] <- corrected_bedrooms

# Plot vs. log assessed total
plot(sfh_final$Number_of_Bedroom, sfh_final$Log_Assessed_Total)
```



We repeat the same procedure for the number of bathrooms and half bathrooms. Since there are more than 4000 properties listed with 0 half bathrooms, we will focus on missing values since it would be impractical to check for true 0 half bathrooms and distinguish them from artifacts of missingness.

```
# Get table of bathroom counts
table(sfh_final$Number_of_Baths, useNA = "always")

##
##      0      1      2      3      4      5      6      7      8      9      10 <NA>
##    6 5556 2914 735 151 34 10 4 2 2 1 1

# Identify properties with 0 or missing number of bathrooms
st_drop_geometry(sfh_final[which(sfh_final$Number_of_Baths == 0 |
                                is.na(sfh_final$Number_of_Baths)),
                           c("Owner", "Location", "Number_of_Bedroom",
                             "Number_of_Baths", "Number_of_Half_Baths")])

##
##          Owner           Location Number_of_Bedroom
## 758255 SANMARTIN MEJIA NELLY 255 LEXINGTON AV        1
## 763284 QUINNIPAC SHELFISH FARM LLC 269 FRONT ST        3
## 777198 S&E APARTMENTS LLC 26 DOWNES ST        4
## 780965 HELLYAR E MICHAEL 132 COVE ST        0
## 771060 CONNECTION FUND INC 98 PARK ST        3
## 773938 SUB-SCRIBE ASSOCIATES LLC 445 GEORGE ST        4
## 774921 ALBANIA DENTAL LLC 1395 CHAPEL ST        2
##          Number_of_Baths Number_of_Half_Baths
## 758255 NA 1
## 763284 0 1
## 777198 0 2
## 780965 0 1
## 771060 0 0
## 773938 0 0
## 774921 0 0
```

```

bath_rows <- row.names(sfh_final[which(sfh_final$Number_of_Baths == 0 |
                                         is.na(sfh_final$Number_of_Baths)), ]]

# 1395 CHAPEL ST does not have any information about number of bathrooms
# Replace 0 with NA to reflect missingness
corrected_baths <- c(2, 0, 2, 1, 2, 2, NA)
sfh_final[bath_rows, "Number_of_Baths"] <- corrected_baths

# Get table of half bathroom counts
table(sfh_final$Number_of_Half_Baths, useNA = "always")

## 
##      0     1     2     3     6 <NA>
## 4855 4345 171    8    1   36

# Identify properties with missing half bathroom counts
# Show first few rows for brevity
head(st_drop_geometry(sfh_final[which(is.na(sfh_final$Number_of_Half_Baths)),
                                c("Owner", "Location", "Number_of_Bedroom",
                                  "Number_of_Baths", "Number_of_Half_Baths")]))
```

	Owner	Location	Number_of_Bedroom	Number_of_Baths
## 754940	UBIERA BRAYANN J	145 HALL ST	3	2
## 755109	WALL JACOB	99 MYRON ST	3	3
## 755307	MCNAMARA KELLY MARIE	37 SHOREHAM RD	4	2
## 755349	TARGOVE MARGARET W	20 ALDEN ST	3	2
## 755625	SCHEIBEL RONALD W	72 OCEAN VIEW ST	2	2
## 756335	ZUNIGA GALO	1015 TOWNSEND AV	3	2
	Number_of_Half_Baths			
## 754940		NA		
## 755109		NA		
## 755307		NA		
## 755349		NA		
## 755625		NA		
## 756335		NA		

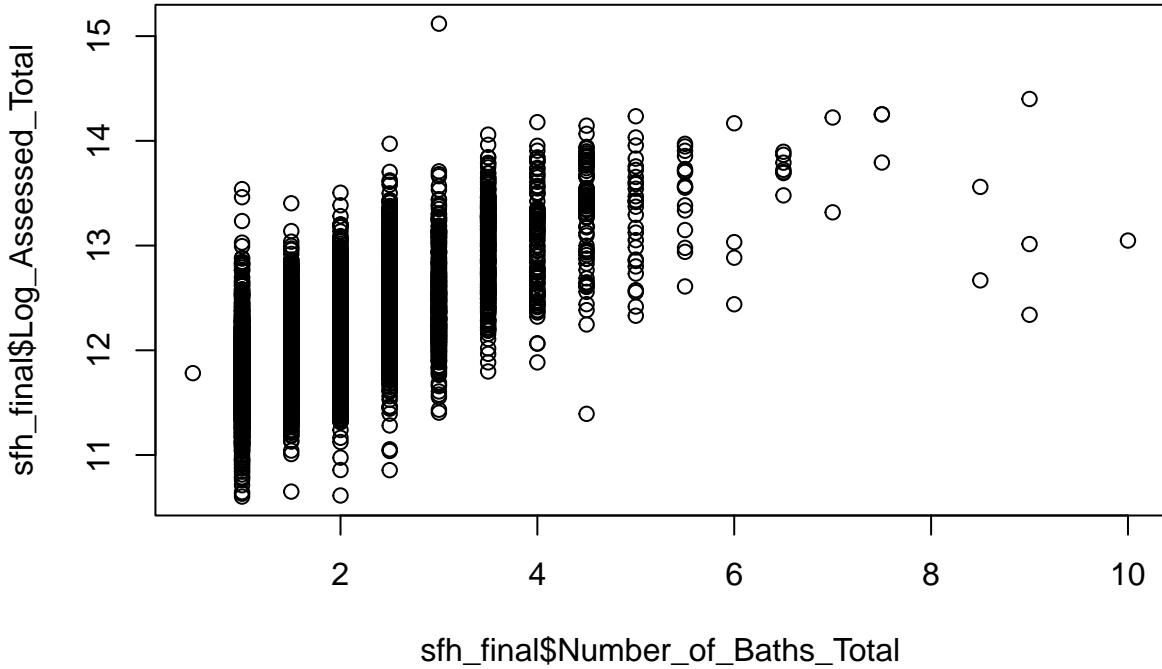
```

half_bath_rows <- row.names(sfh_final[which(is.na(
                                         sfh_final$Number_of_Half_Baths)), ])

corrected_half_baths <- c(1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1,
                           0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1,
                           1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0)
sfh_final[half_bath_rows, "Number_of_Half_Baths"] <- corrected_half_baths

# Create total bathrooms variable
sfh_final$Number_of_Baths_Total <- sfh_final$Number_of_Baths +
                                         0.5 * sfh_final$Number_of_Half_Baths

# Plot vs. log assessed total
plot(sfh_final$Number_of_Baths_Total, sfh_final$Log_Assessed_Total)
```



Next, we inspect property condition, observing there is a nice 1-to-1 correspondence between conditions and their descriptions.

```
# Table of condition and descriptions
table(sfh_final$Condition, sfh_final$Condition_Description, useNA = "always")
```

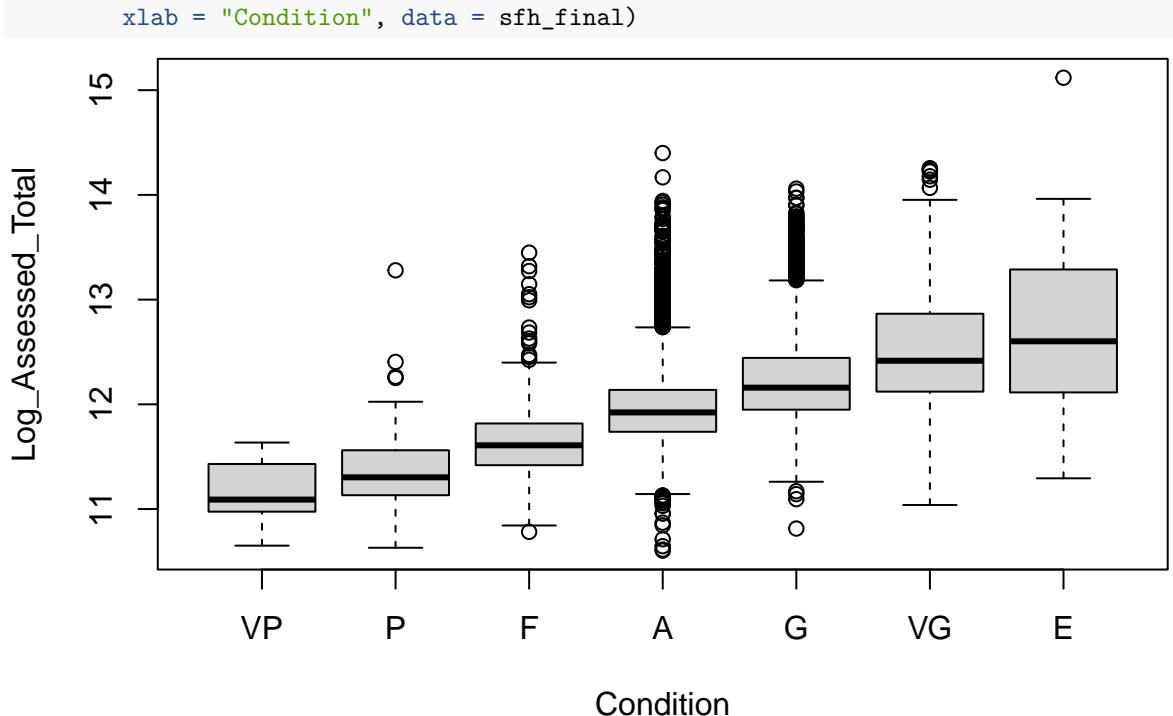
	Average	Excellent	F	Good	Poor	U	Very Good	Very Poor	<NA>
A	5216	0	0	0	0	0	0	0	0
E	0	77	0	0	0	0	0	0	0
F	0	0	326	0	0	0	0	0	0
G	0	0	0	3304	0	0	0	0	0
P	0	0	0	0	37	0	0	0	0
U	0	0	0	0	0	2	0	0	0
VG	0	0	0	0	0	0	441	0	0
VP	0	0	0	0	0	0	0	13	0
<NA>	0	0	0	0	0	0	0	0	0

There are two properties with condition listed as “U”. Inspecting these locations in Vision Appraisal, it’s likely that “U” represents Unknown or Unspecified as the condition data field is missing on Vision Appraisal. Using Redfin, we replace these unknown conditions.

```
# Substitute with Redfin's condition data
unk_cond_rows <- row.names(sfh_final[which(sfh_final$Condition == "U"), ])
sfh_final[unk_cond_rows, "Condition"] <- c("A", "A")
sfh_final[unk_cond_rows, "Condition_Description"] <- c("Average", "Average")
```

We also see there is a condition listed as “F” with description “F”. We initially thought this may correspond to “Fair”, and so to justify this we made boxplots grouped by condition. We should expect the mean log assessed total to increase with “better” condition levels.

```
# Boxplot of log assessed total by condition
boxplot(Log_Assessed_Total ~ factor(Condition,
                                     levels = c("VP", "P", "F", "A", "G",
                                               "VG", "E"))),
```



```

# Update description to fair
sfh_final[which(sfh_final$Condition == "F"), "Condition_Description"] <- "Fair"

# We'll use condition description so that our model summaries later are easier
# to read and interpret
# Convert condition description to a factor and set "Average" as the reference
sfh_final$Condition_Description <- as.factor(sfh_final$Condition_Description)
sfh_final$Condition_Description <- relevel(sfh_final$Condition_Description,
                                             ref = "Average")
```

Next, we take a closer look at zoning districts. Because there are so many different zones, with many only corresponding to a handful of homes, we decided to group them based on their descriptions into five broader categories that represent their overall use: Residential, Mixed-Use, Commercial, Industrial, and Planned Development. The result is a more interpretable categorical variable. To achieve this, we referred to Article II of New Haven's Zoning Ordinances available on the Municode Library.

```

# Table of zone counts
table(sfh_final$Zone)

##          BA    BA/RM1   BA/RM2   BA/RS2      BA1       BB       BC       BD      BD1      CEM
##        56        3        5        1        7        3       10        6        7        3
##        IH    IH/RM2      IL    PDD  119    PDD  26    PDD  39    PDD  49    PDD  52    PDU 102    PDU 106
##        24        1        7       12        2        4       13        1        4       30
##    PDU 108    PDU 16    PDU 72    PDU 75     RH1     RH2     RM1     RM2  RM2/RS2      RO
##        7        1        1        2       27       21    1892    1688        5       22
##      RS1  RS1/RS2      RS2
##      374        3      5174

# Define broader groupings based on zone descriptions
residential <- c("RS1", "RS2", "RS1/RS2", "RM1", "RM2", "RM2/RS2", "RH1", "RH2")
mixed_use <- c("RO", "BA/RM1", "BA/RM2", "BA/RS2", "CEM")
```

```

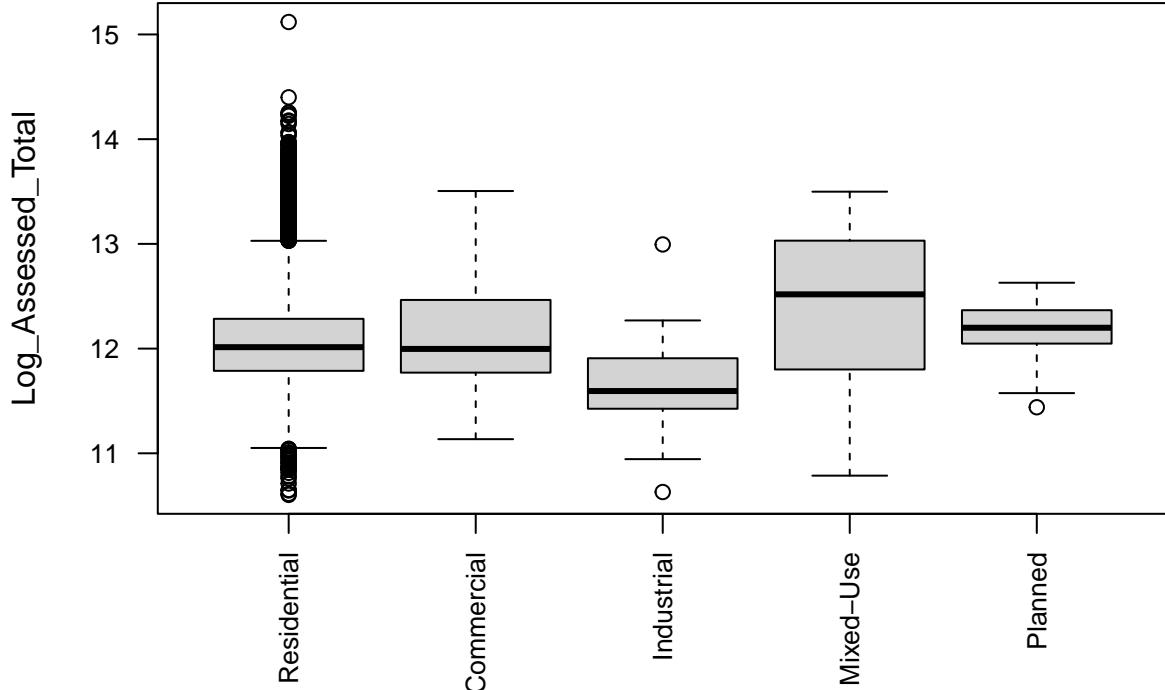
commercial <- c("BA", "BA1", "BB", "BC", "BD", "BD1")
industrial <- c("IL", "IH", "IH/RM2")
planned_development <- c("PDD 119", "PDD 26", "PDD 39", "PDD 49", "PDD 52",
                           "PDU 102", "PDU 106", "PDU 108", "PDU 16", "PDU 72",
                           "PDU 75", "PDU 95")

# Assign zone categories
sfh_final$Zone_Category <- ifelse(sfh_final$Zone %in% residential,
                                    "Residential", NA)
sfh_final$Zone_Category <- ifelse(sfh_final$Zone %in% mixed_use,
                                    "Mixed-Use", sfh_final$Zone_Category)
sfh_final$Zone_Category <- ifelse(sfh_final$Zone %in% commercial,
                                    "Commercial", sfh_final$Zone_Category)
sfh_final$Zone_Category <- ifelse(sfh_final$Zone %in% industrial,
                                    "Industrial", sfh_final$Zone_Category)
sfh_final$Zone_Category <- ifelse(sfh_final$Zone %in% planned_development,
                                    "Planned",
                                    sfh_final$Zone_Category)

# Convert to factor, set "Residential" as reference
sfh_final$Zone_Category <- as.factor(sfh_final$Zone_Category)
sfh_final$Zone_Category <- relevel(sfh_final$Zone_Category, ref = "Residential")

# Boxplot of log assessed total by zone category
boxplot(Log_Assessed_Total ~ Zone_Category, data = sfh_final, cex.axis = 0.8,
        las = 2, xlab = "")

```



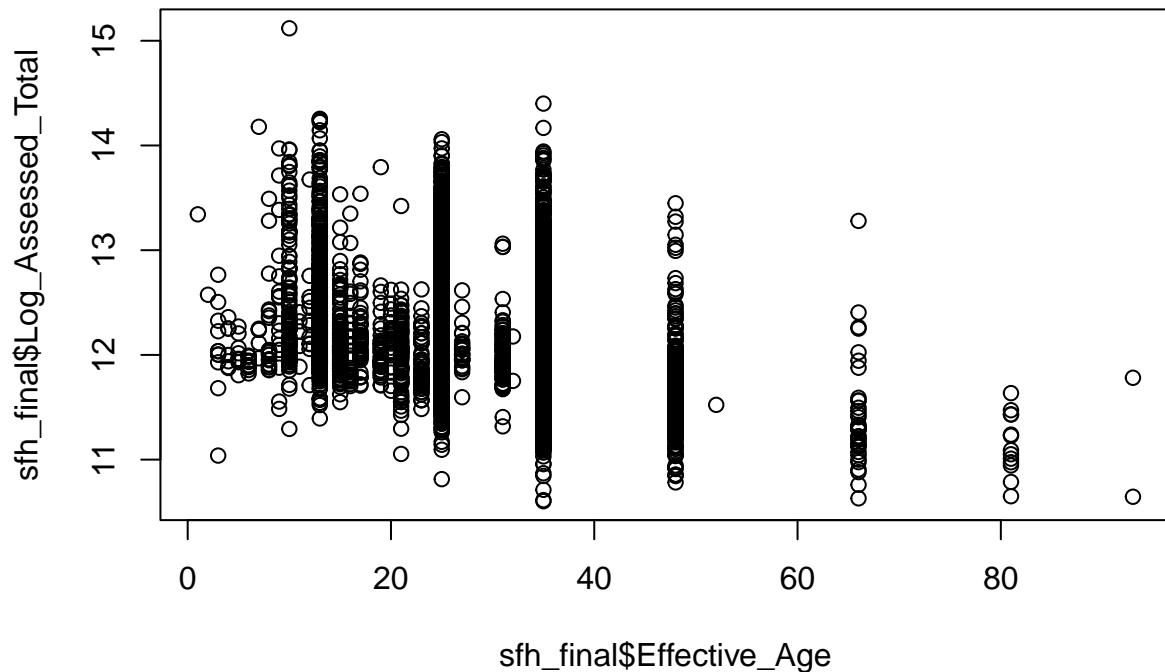
Lastly, we create the effective age variable.

```

# Check for missing effective year built
sum(is.na(sfh_final$EYB))

```

```
## [1] 0  
# Create effective age variable and plot against log assessed total  
sfh_final$Effective_Age <- 2024 - sfh_final$EYB  
plot(sfh_final$Effective_Age, sfh_final$Log_Assessed_Total)
```



Food Access Features

To create meaningful and interpretable features related to proximity to food, we took two approaches to the idea of proximity:

1. Distance to nearest food retailers
2. Number of food retailers within some radius

```
# Load SNAP data
s <- read.csv("data/CT_SNAP_Authorized_Retailers_20240920.csv")

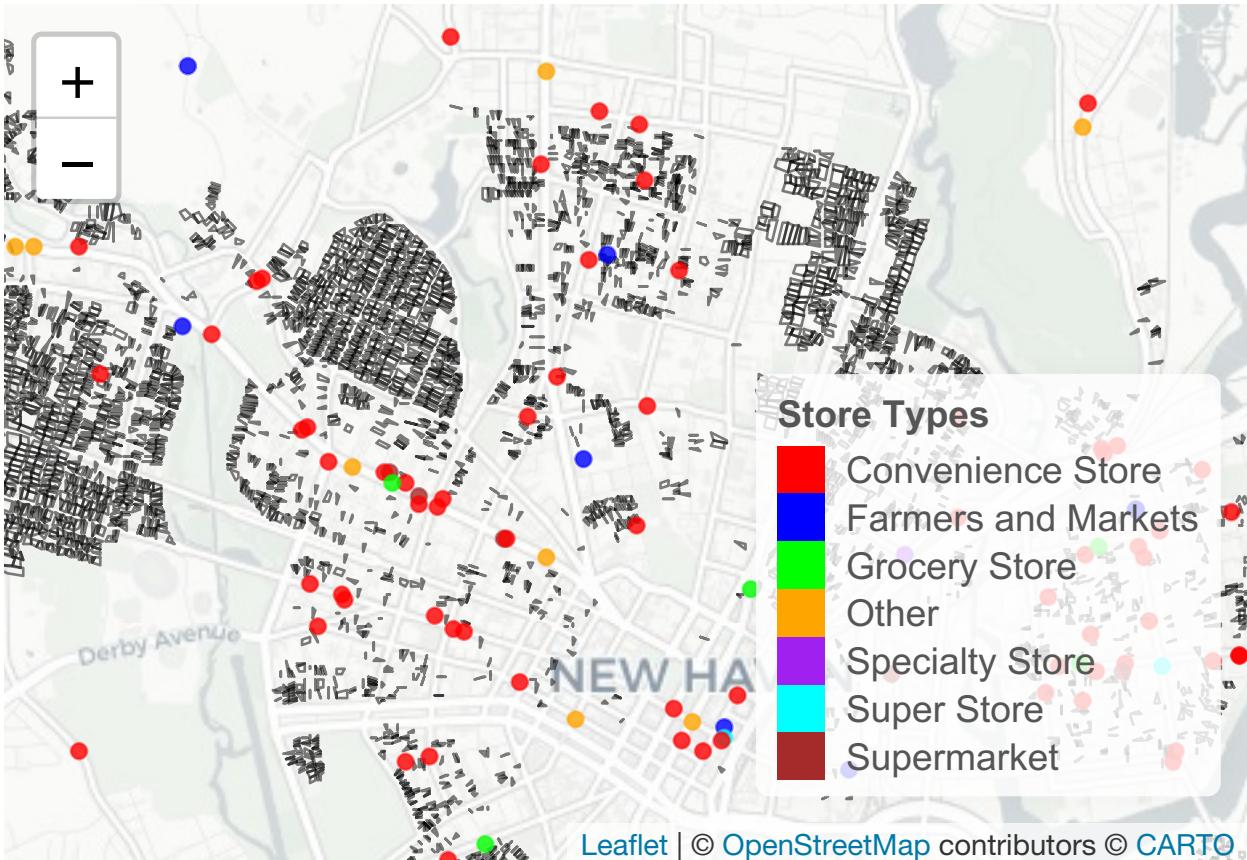
# Convert to spatial
sg <- st_as_sf(s, coords = c("Longitude", "Latitude"), crs = 4326, remove=FALSE)
sfh_final <- st_transform(sfh_final, crs = 4326)

# Visualize store locations, superimposed on New Haven map and geometries of
# single-family homes in our subset
pal <- colorFactor(
  palette = c("red", "blue", "green", "orange", "purple", "cyan", "brown"),
  domain = sg$Store.Type
)
leaflet() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%

  # Center on New Haven
  setView(lng = -72.93, lat = 41.32, zoom = 13) %>%

  # Add sfh homes
  addPolygons(data = sfh_final,
    color = "black",
    weight = 1,
    fillColor = "transparent",
    popup = ~paste("Location:", Location,
      "  
State Use: ", State_Use_Description),
    group = "Homes") %>%

  # Add stores, colored by type
  addCircleMarkers(data = sg,
    radius = 3,
    color = ~pal(Store.Type),
    stroke = TRUE,
    weight = 1,
    fillOpacity = 0.8,
    popup = ~paste0("Store: ", Store.Name,
      "  
Type: ", Store.Type),
    group = "Stores") %>%
  addLegend(
    position = "bottomright",
    pal = pal,
    values = sg$Store.Type,
    title = "Store Types",
    opacity = 1
  )
```



Certain store types like specialty stores are relatively infrequent. Moreover, we could technically consider super stores and supermarkets together, since we can view super stores as extensions of supermarkets that happen to include non-food items as well. As a result, we decided to group super stores and supermarkets together, leave convenience stores and grocery stores as is as separate groups, and bucket the rest into “Other.”

```
# View counts by store type





```

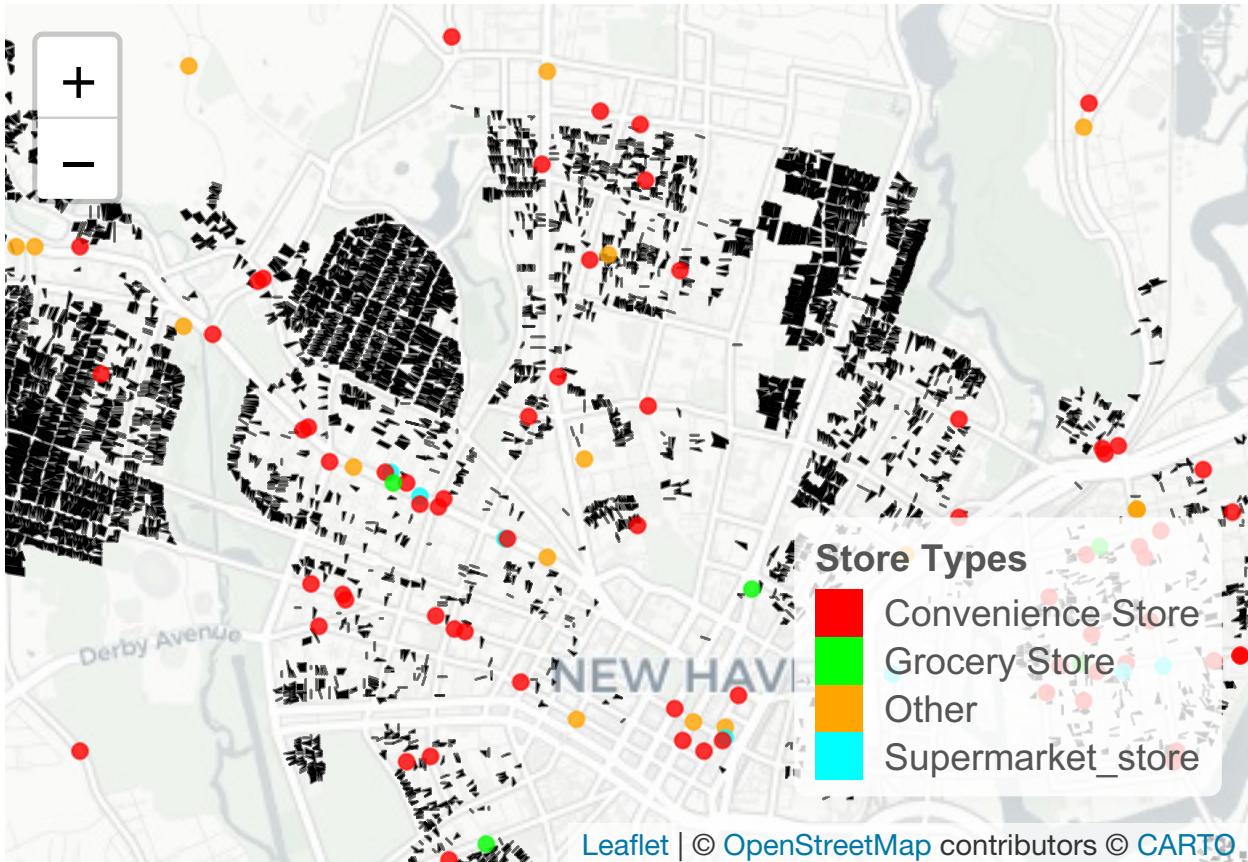
```

# Center on New Haven
setView(lng = -72.93, lat = 41.32, zoom = 13) %>%

# Add sfh homes
addPolygons(data = sfh_final,
            color = "black",
            weight = 1,
            fillColor = "black",
            popup = ~paste("Location:", Location,
                          "<br>State Use: ", State_Use_Description),
            group = "Homes") %>%

# Add stores, colored by type
addCircleMarkers(data = sg,
                  radius = 3,
                  color = ~palGrouped(Store.Type.Grouped),
                  stroke = TRUE,
                  weight = 1,
                  fillOpacity = 0.8,
                  popup = ~paste0("Store: ", Store.Name,
                                 "<br>Type: ", Store.Type),
                  group = "Stores") %>%
addLegend(
  position = "bottomright",
  pal = palGrouped,
  values = sg$Store.Type.Grouped,
  title = "Store Types",
  opacity = 1
)

```



For our first approach to “proximity”, we compute the distance in meters to the nearest store of each grouped store type we defined earlier.

```
# Transform back to WGS 84 Mercator so calculations are in meters
sfh_final <- st_transform(sfh_final, crs = 3857)
sg <- st_transform(sg, crs = 3857)

# Define the store types to analyze
target_store_types <- unique(sg$Store.Type.Grouped)

# Initialize distance and area columns for each target store type
for (type in target_store_types) {

  # Define column names for distance and area
  dist_col <- paste0("dist_", type)

  # Subset SNAP retailers of the current Store Type
  stores_type <- sg[sg$Store.Type.Grouped == type, ]

  # Find the nearest store of the current type for each home
  nearest_store_indices <- st_nearest_feature(sfh_final, stores_type)

  # Calculate distances to the nearest store
  distances <- st_distance(sfh_final,
                            stores_type[nearest_store_indices, ],
                            by_element = TRUE)}
```

```

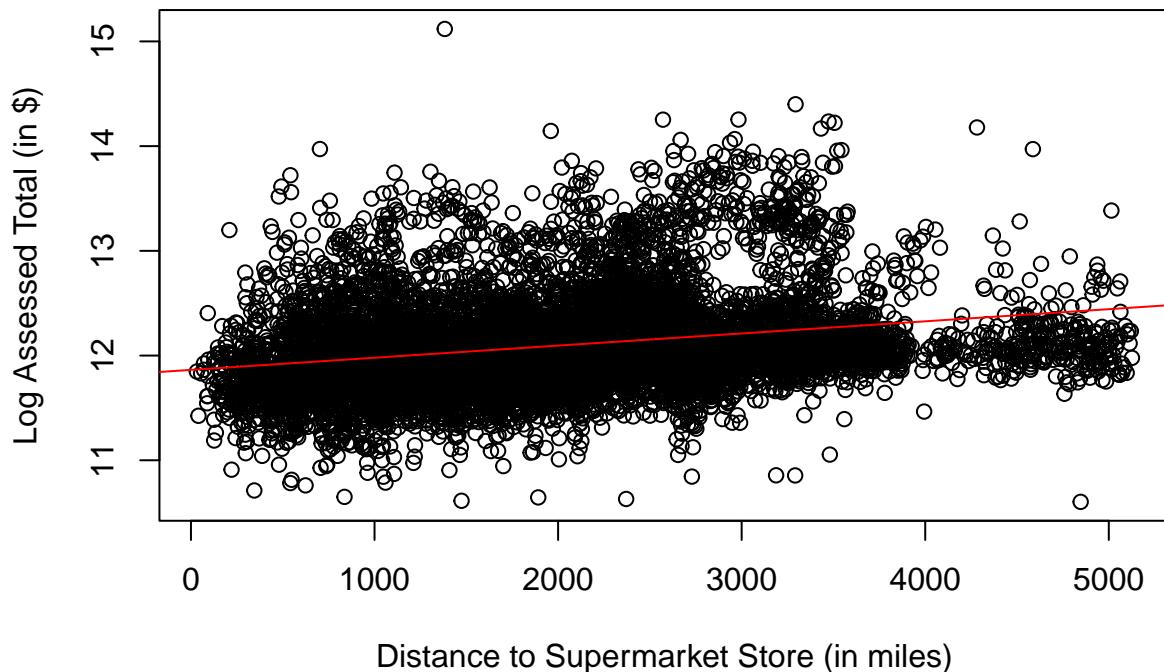
# Assign distances to the home data (convert to numeric, e.g., meters)
sfh_final[[dist_col]] <- as.numeric(distances)
}

# Plot new distance features vs. log assessed total

# Plot distance to Supermarket Store vs. Log Assessed Total
ssdist_ls <- lm(Log_Assessed_Total ~ dist_Supermarket_Store, data = sfh_final)
plot(sfh_final$dist_Supermarket_Store, sfh_final$Log_Assessed_Total,
      main = "Distance to Supermarket Store vs. Log Assessed Total",
      xlab = "Distance to Supermarket Store (in miles)",
      ylab = "Log Assessed Total (in $)")
abline(ssdist_ls, col = "red")

```

Distance to Supermarket Store vs. Log Assessed Total

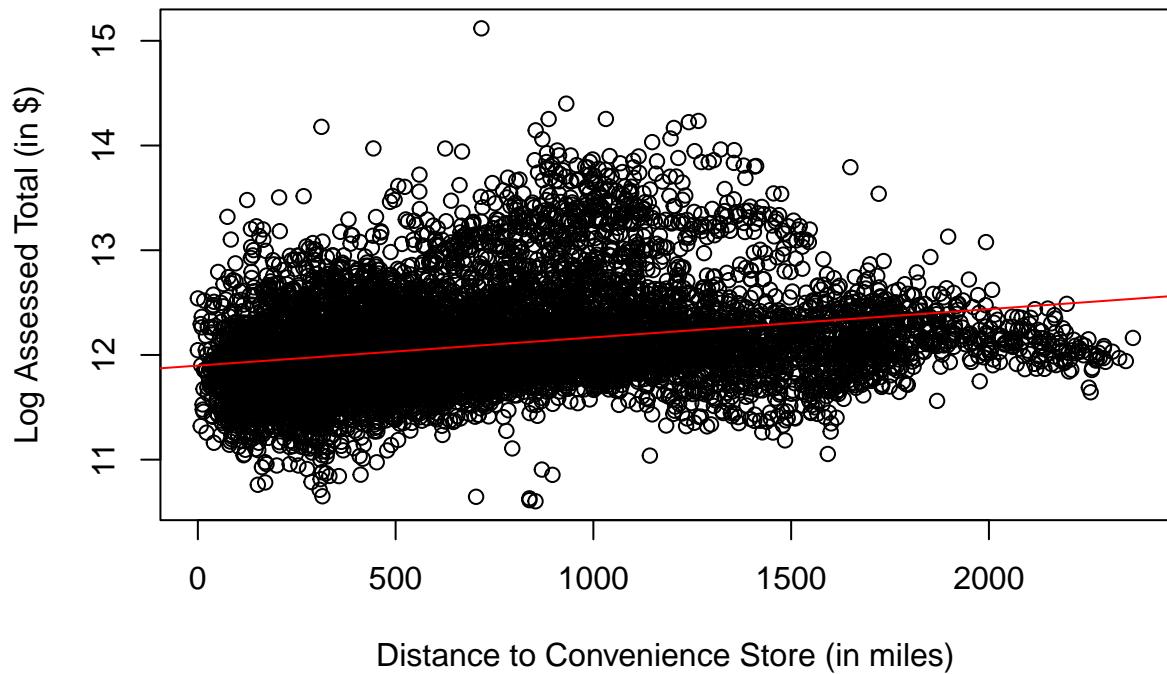


```

# Plot distance to Convenience Store vs. Log Assessed Total
csdist_ls <- lm(Log_Assessed_Total ~ `dist_Convenience_Store`, data = sfh_final)
plot(sfh_final$`dist_Convenience_Store`, sfh_final$Log_Assessed_Total,
      main = "Distance to Convenience Store vs. Log Assessed Total",
      xlab = "Distance to Convenience Store (in miles)",
      ylab = "Log Assessed Total (in $)")
abline(csdist_ls, col = "red")

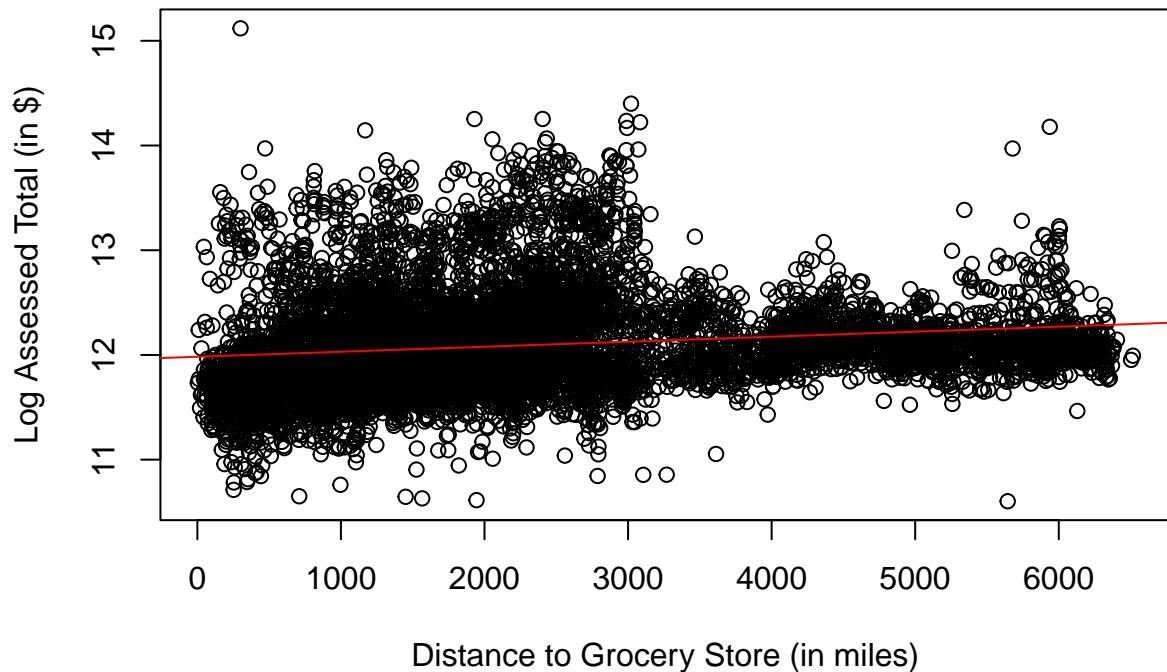
```

Distance to Convenience Store vs. Log Assessed Total



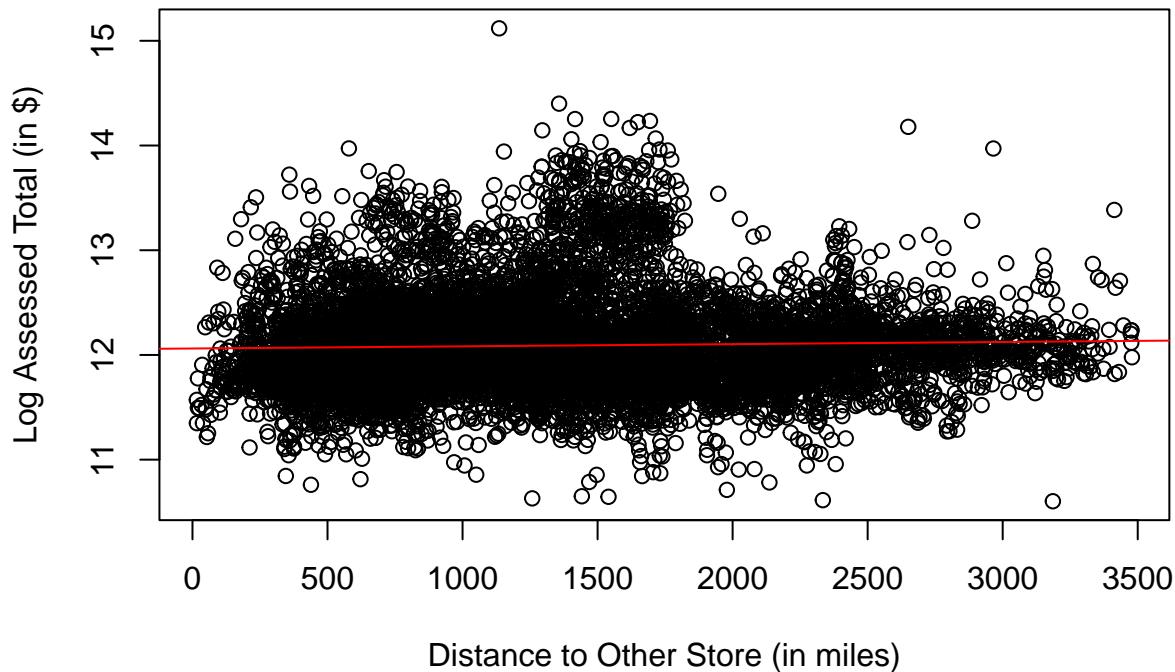
```
# Plot distance to Grocery Store vs. Log Assessed Total
gsdist_ls <- lm(Log_Assessed_Total ~ `dist_Grocery Store`, data = sfh_final)
plot(sfh_final$`dist_Grocery Store`, sfh_final$Log_Assessed_Total,
     main = "Distance to Grocery Store vs. Log Assessed Total",
     xlab = "Distance to Grocery Store (in miles)",
     ylab = "Log Assessed Total (in $)")
abline(gsdist_ls, col = "red")
```

Distance to Grocery Store vs. Log Assessed Total



```
# Plot distance to Other Store vs. Log Assessed Total
otdist_ls <- lm(Log_Assessed_Total ~ `dist_Other`, data = sfh_final)
plot(sfh_final$dist_Other, sfh_final$Log_Assessed_Total,
     main = "Distance to Other Store vs. Log Assessed Total",
     xlab = "Distance to Other Store (in miles)",
     ylab = "Log Assessed Total (in $)")
abline(otdist_ls, col = "red")
```

Distance to Other Store vs. Log Assessed Total



Peter Added (Comment what we observe)

Our analysis reveals a weak positive relationship between proximity to stores and the log of the total assessed value of single-family homes, with the strongest relationship observed for supermarkets. Interestingly, for stores classified as “other,” the relationship appears much weaker and lacks a clear linear trend. These findings are surprising, as we initially expected closer proximity to food retailers—especially supermarkets—to be associated with higher property values.

One possible explanation could be that while living close to a supermarket offers convenience, it may also bring downsides, such as increased traffic, noise, and higher crime rates. Homes with higher assessed values are often found in more secluded areas with fewer stores. In affluent neighborhoods, residents may value privacy and prefer upscale dining options rather than frequenting nearby grocery stores, which could reduce the impact of store proximity on property values.

Additionally, economic challenges may influence this dynamic. Grocery stores in expensive areas face high operating costs, which can make them unsustainable. Even upscale retailers like Foxtrot and Erewhon have faced closures in affluent areas due to these challenges.

Overall, the relationship between store proximity and property values seems more complex than we initially thought. Factors like neighborhood affluence, store types, and other socioeconomic conditions likely influence the effect of proximity, and further analysis is needed to fully understand these dynamics.

```
# Compute distance matrix
dist_matrix <- st_distance(sfh_final, sg, by_element = FALSE)

# ~1 mi = 1609 m, ~2 mi = 3218 m, and ~10 mi = 16090 m
radii_labels <- c("1_mile", "2_miles", "10_miles")
radii_meters <- c(1609, 3218, 16090)

for (i in 1:3) {
  # Count the number of stores within the distance for each home
  counts <- apply(dist_matrix, 1, function(x) sum(x <= radii_meters[i]))
```

```

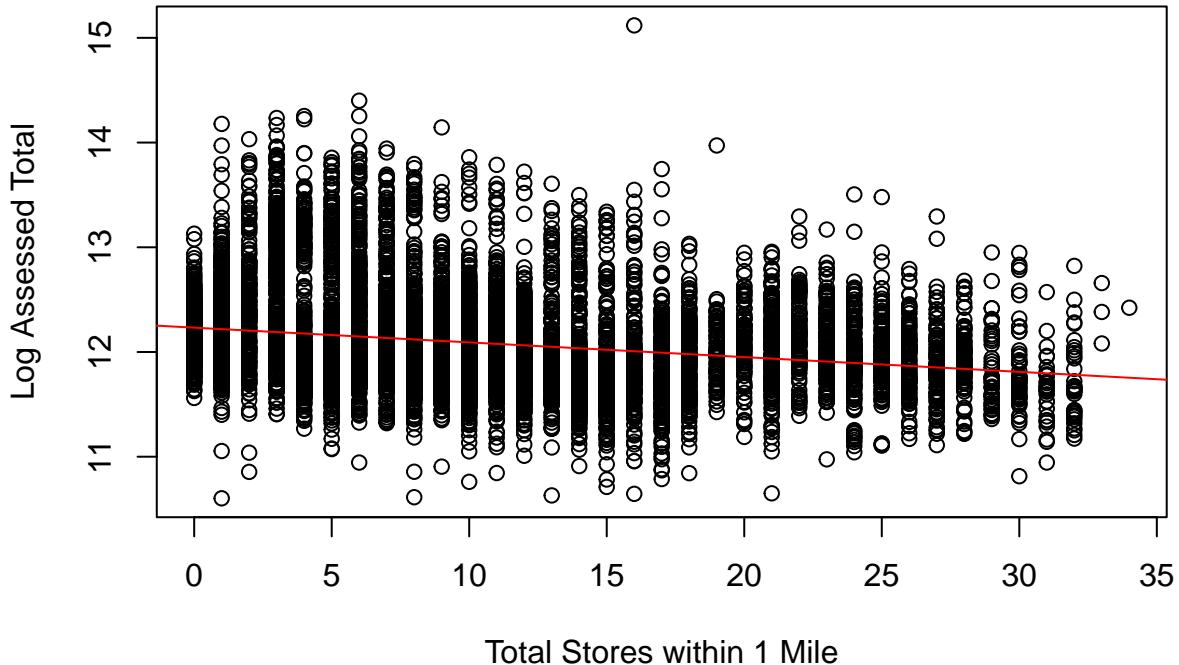
# Add the counts to the data frame with a descriptive column name
column_name <- paste0("total_", radii_labels[i])
sfh_final[[column_name]] <- counts
}

# Plot new count features vs. log assessed total

# Plot Total Stores within 1 Mile vs. Log Assessed Total
mile1_ls <- lm(Log_Assessed_Total ~ total_1_mile, data = sfh_final)
plot(sfh_final$total_1_mile, sfh_final$Log_Assessed_Total,
     main = "Total Stores within 1 Mile vs. Log Assessed Total",
     xlab = "Total Stores within 1 Mile", ylab = "Log Assessed Total")
abline(mile1_ls, col = "red")

```

Total Stores within 1 Mile vs. Log Assessed Total

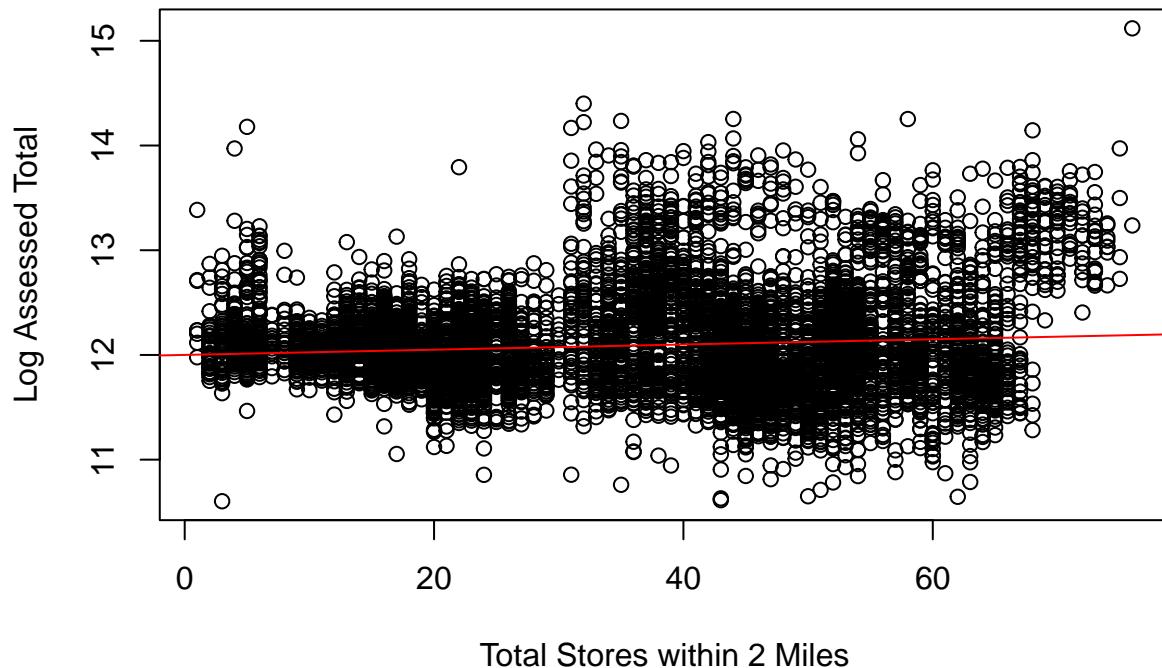


```

# Plot Total Stores within 2 Miles vs. Log Assessed Total
mile2_ls <- lm(Log_Assessed_Total ~ total_2_miles, data = sfh_final)
plot(sfh_final$total_2_mile, sfh_final$Log_Assessed_Total,
     main = "Total Stores within 2 Miles vs. Log Assessed Total",
     xlab = "Total Stores within 2 Miles", ylab = "Log Assessed Total")
abline(mile2_ls, col = "red")

```

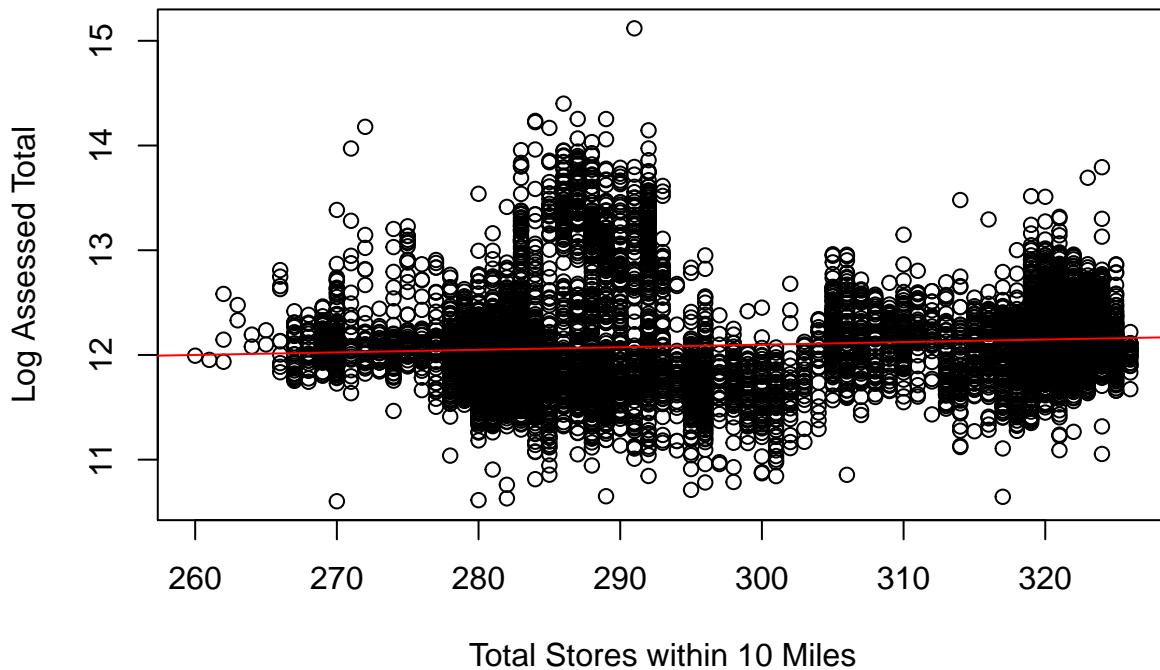
Total Stores within 2 Miles vs. Log Assessed Total



Total Stores within 2 Miles

```
# Plot Total Stores within 10 Miles vs. Log Assessed Total
mile10_ls <- lm(Log_Assessed_Total ~ total_10_miles, data = sfh_final)
plot(sfh_final$total_10_mile, sfh_final$Log_Assessed_Total,
     main = "Total Stores within 10 Miles vs. Log Assessed Total",
     xlab = "Total Stores within 10 Miles", ylab = "Log Assessed Total")
abline(mile10_ls, col = "red")
```

Total Stores within 10 Miles vs. Log Assessed Total



Peter Added: Comment what we observe

Our analysis of the number of stores within specified radii reveals an unexpected pattern: there is a zero to negative association between the total number of stores nearby and the log of the total assessed value of single-family homes. This pattern holds across the 1-mile, 2-mile, and 10-mile radii, contrary to our initial hypothesis that more nearby stores would be associated with higher property values. This trend is consistent with earlier exploratory plots, reinforcing the recurring pattern.

We had anticipated that a higher density of nearby stores, particularly grocery stores, would increase convenience and thus be linked to higher property values. However, the results suggest that areas with higher property values tend to have fewer stores nearby. This could be attributed to affluent neighborhoods valuing privacy and seclusion, making them less reliant on nearby retail outlets, while high operating costs prevent many businesses from thriving in such areas.

In wealthier areas, demand may shift toward upscale restaurants and specialty stores rather than supermarkets and convenience stores. This aligns with broader urban trends, where the retail landscape changes as neighborhood incomes rise. Affluent areas may attract boutique-style stores catering to higher-income residents, reducing the need for everyday stores.

Overall, our findings suggest that the relationship between store density and property values is more complex than initially expected. While we anticipated a positive correlation, the data reflects a deeper socioeconomic dynamic that requires further analysis to fully understand.

Modeling

While these marginal plots provide valuable intuition, they do not tell the full story, as they only show relationships in isolation. To capture the joint dynamics of multiple factors that determine housing values, we now turn to a more rigorous modeling approach.

Step 1: Baseline Model

Our baseline model includes essential predictors that we believed would sufficiently explain the total assessed values of properties. These predictors are:

- (a) Square Root of Effective Area,
- (b) Number of Bedrooms,
- (c) Total Number of Bathrooms,
- (d) Property Condition,
- (e) Zoning Categories, and
- (f) Effective Age of the Property.

These variables were previously found to be positively correlated with the log of total assessed values, providing a solid foundation for our model. However, it is important to remain open to the possibility that the addition of other factors could shift these results.

Step 2: Proposed Model

In the advanced model, we introduce two new sets of variables: (a) distance to stores and (b) the number of stores within specified radii. We initially expected these factors to have a positive association with housing values, given that proximity to retail stores offers convenience. However, based on the earlier findings, we are open to the idea that this relationship may be more complex and could vary by neighborhood characteristics.

By comparing the baseline and advanced models, we can assess the changes in coefficients, adjusted R-squared, residual standard error, and importantly, determine whether the added predictors provide statistically significant effects on property values. This will be formally tested using an ANOVA test, allowing us to evaluate the overall fit and complexity of the model.

Conclusion:

Through this two-step approach, we aim to uncover how both traditional property features and proximity to stores influence the assessed values of homes. While we initially expected proximity to stores to have a positive impact, especially for supermarkets, our findings so far suggest that this relationship might be more nuanced. This is consistent with earlier patterns we observed, where affluent areas had fewer stores nearby, which likely affects housing values differently than we anticipated.

```
# Fit baseline model
m1 <- lm(Log_Assessed_Total ~ Sqrt_Effective_Area + Number_of_Bedroom +
           Number_of_Baths_Total + Condition_Description + Zone_Category +
           Effective_Age, data = sfh_final)

# Check model summary
summary(m1)

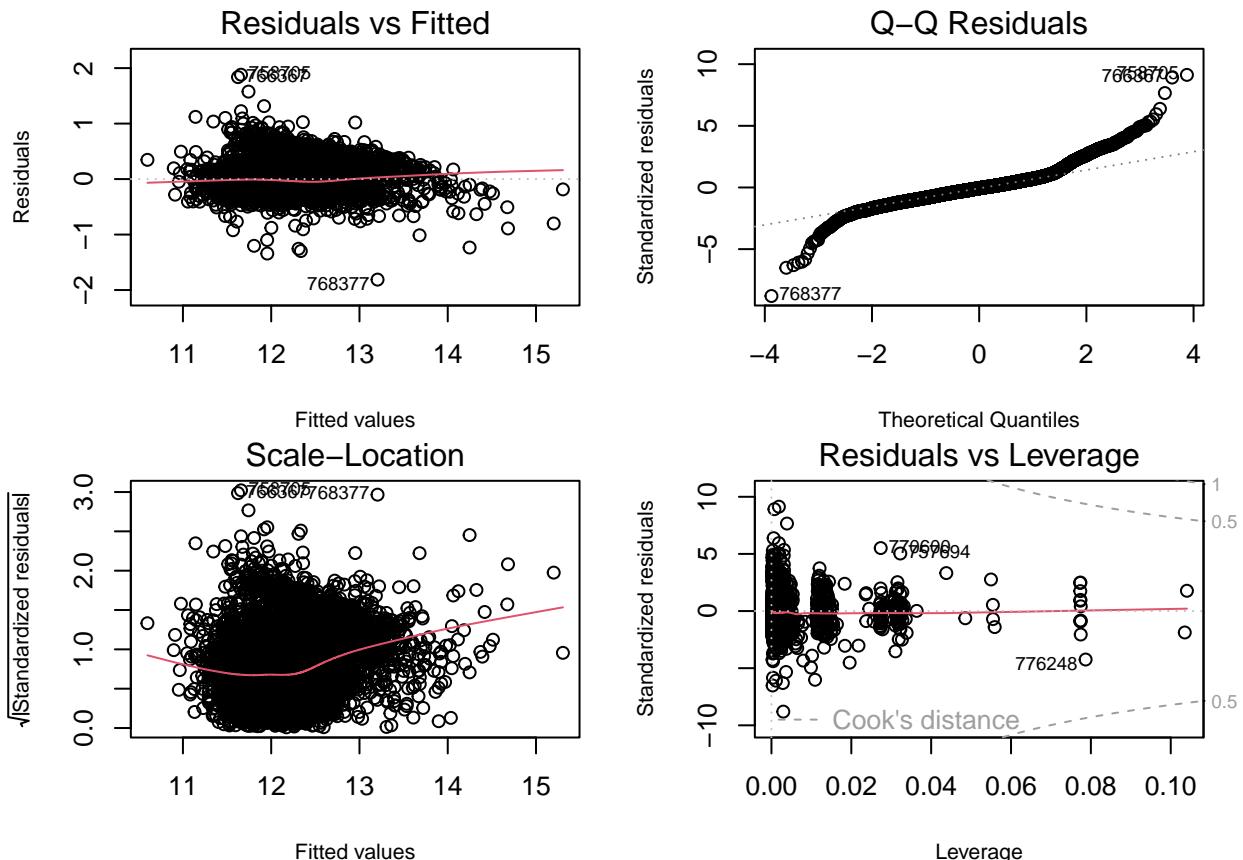
##
## Call:
## lm(formula = Log_Assessed_Total ~ Sqrt_Effective_Area + Number_of_Bedroom +
##     Number_of_Baths_Total + Condition_Description + Zone_Category +
##     Effective_Age, data = sfh_final)
##
## Residuals:
```

```

##      Min     1Q   Median     3Q    Max
## -1.81187 -0.11658 -0.01217  0.08936  1.88106
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)           10.5278120  0.0188636 558.101 < 2e-16 ***
## Sqrt_Effective_Area  0.0372393  0.0003809  97.776 < 2e-16 ***
## Number_of_Bedroom     -0.0141046  0.0029393 -4.799 1.62e-06 ***
## Number_of_Baths_Total  0.0699904  0.0039629  17.661 < 2e-16 ***
## Condition_DescriptionExcellent 0.2594459  0.0262929  9.868 < 2e-16 ***
## Condition_DescriptionFair    -0.1542599  0.0137657 -11.206 < 2e-16 ***
## Condition_DescriptionGood   0.0606080  0.0063445  9.553 < 2e-16 ***
## Condition_DescriptionPoor   -0.2457624  0.0375466 -6.546 6.24e-11 ***
## Condition_DescriptionVery Good 0.1399730  0.0143437  9.759 < 2e-16 ***
## Condition_DescriptionVery Poor -0.4274557  0.0619092 -6.905 5.36e-12 ***
## Zone_CategoryCommercial   -0.0806801  0.0220031 -3.667 0.000247 ***
## Zone_CategoryIndustrial   -0.1893669  0.0366378 -5.169 2.41e-07 ***
## Zone_CategoryMixed-Use     -0.0819313  0.0361773 -2.265 0.023553 *
## Zone_CategoryPlanned      0.0440296  0.0247806  1.777 0.075637 .
## Effective_Age            -0.0059266  0.0004862 -12.189 < 2e-16 ***
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2062 on 9400 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8007, Adjusted R-squared:  0.8005
## F-statistic: 2698 on 14 and 9400 DF, p-value: < 2.2e-16

# Diagnostic plots
par(mfrow = c(2, 2))
par(mar = c(4, 4, 1.5, 1.5))
plot(m1, cex.lab = 0.8)

```



```
# Fit proposed model with food access features
m2 <- lm(Log_Assessed_Total ~ Sqrt_Effective_Area + Number_of_Bedroom +
  Number_of_Baths_Total + Condition_Description + Zone_Category +
  Effective_Age + `dist_Convenience Store` + dist_Other +
  `dist_Grocery Store` + dist_Supermarket_store + total_1_mile +
  total_2_miles + total_10_miles, data = sfh_final)

# Check model summary
summary(m2)

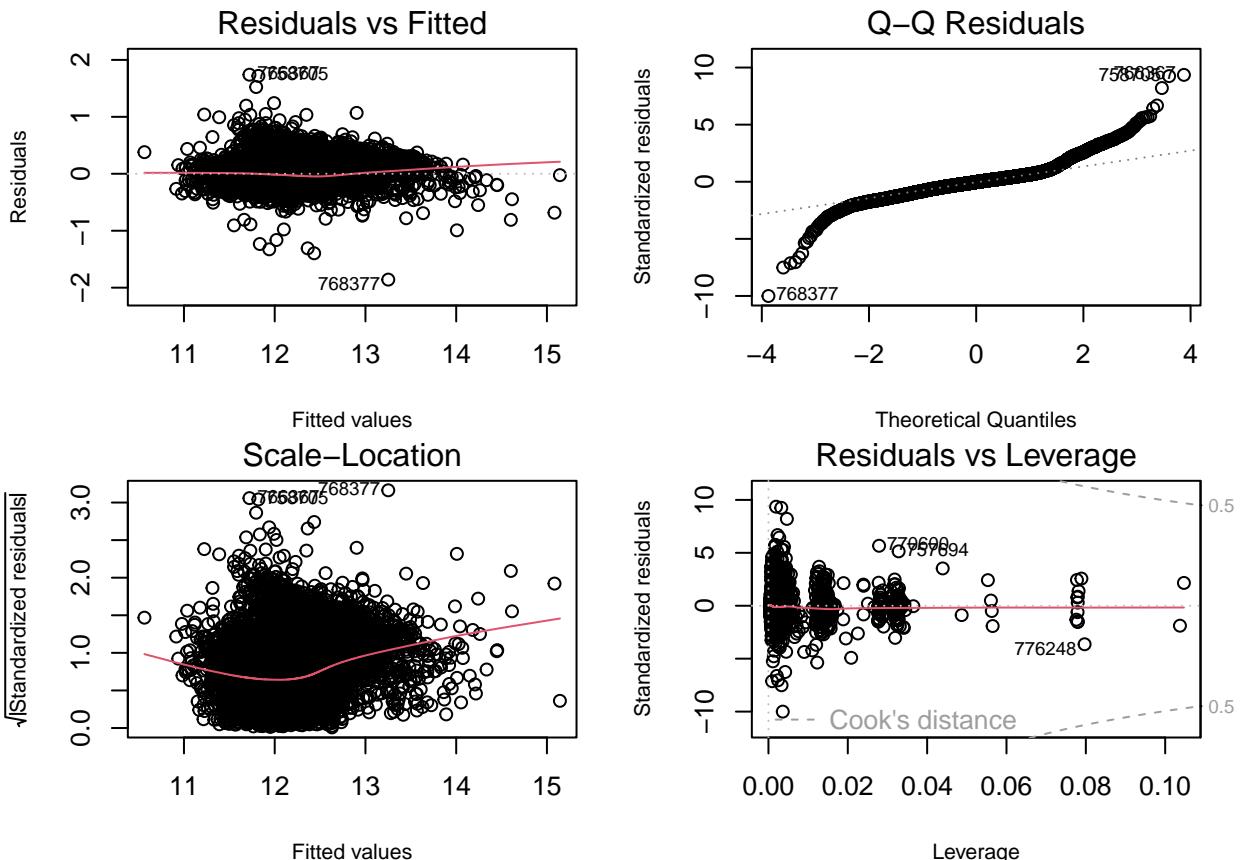
##
## Call:
## lm(formula = Log_Assessed_Total ~ Sqrt_Effective_Area + Number_of_Bedroom +
##     Number_of_Baths_Total + Condition_Description + Zone_Category +
##     Effective_Age + `dist_Convenience Store` + dist_Other + `dist_Grocery Store` +
##     dist_Supermarket_store + total_1_mile + total_2_miles + total_10_miles,
##     data = sfh_final)
##
## Residuals:
##      Min        1Q        Median       3Q        Max 
## -1.85860 -0.09659 -0.00503  0.07749  1.73991 
##
## Coefficients:
## (Intercept)           Estimate Std. Error t value Pr(>|t|)    
## 1.071e+01  4.789e-02 223.639 < 2e-16 ***  
## Sqrt_Effective_Area 3.430e-02 3.595e-04  95.416 < 2e-16 ***
```

```

## Number_of_Bedroom           -2.306e-03  2.693e-03 -0.856  0.391945
## Number_of_Baths_Total       6.144e-02  3.597e-03 17.083 < 2e-16 ***
## Condition_DescriptionExcellent 2.383e-01  2.376e-02 10.031 < 2e-16 ***
## Condition_DescriptionFair    -1.220e-01  1.251e-02 -9.750 < 2e-16 ***
## Condition_DescriptionGood    4.960e-02  5.806e-03  8.544 < 2e-16 ***
## Condition_DescriptionPoor   -1.944e-01  3.400e-02 -5.718 1.11e-08 ***
## Condition_DescriptionVery Good 1.204e-01  1.298e-02  9.276 < 2e-16 ***
## Condition_DescriptionVery Poor -4.209e-01  5.597e-02 -7.520 5.97e-14 ***
## Zone_CategoryCommercial      -3.952e-03  2.003e-02 -0.197  0.843571
## Zone_CategoryIndustrial     -1.923e-01  3.315e-02 -5.799 6.88e-09 ***
## Zone_CategoryMixed-Use       -3.747e-02  3.280e-02 -1.142  0.253309
## Zone_CategoryPlanned        3.302e-02  2.255e-02  1.464  0.143199
## Effective_Age               -6.133e-03  4.415e-04 -13.893 < 2e-16 ***
## `dist_Convenience_Store`    7.220e-05  5.536e-06 13.041 < 2e-16 ***
## dist_Other                  -6.933e-05  4.090e-06 -16.952 < 2e-16 ***
## `dist_Grocery_Store`        3.406e-05  3.741e-06  9.103 < 2e-16 ***
## dist_Supermarket_store      1.268e-05  5.225e-06  2.427  0.015263 *
## total_1_mile                -9.516e-03  4.767e-04 -19.963 < 2e-16 ***
## total_2_miles                3.518e-03  1.984e-04 17.729 < 2e-16 ***
## total_10_miles               -5.206e-04  1.402e-04 -3.713  0.000206 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1861 on 9393 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8378, Adjusted R-squared:  0.8374
## F-statistic: 2310 on 21 and 9393 DF,  p-value: < 2.2e-16

# Diagnostic plots
par(mfrow = c(2, 2))
par(mar = c(4, 4, 1.5, 1.5))
plot(m2, cex.lab = 0.8)

```



```
# Model comparison
anova(m1, m2)
```

```
## Analysis of Variance Table
##
## Model 1: Log_Assessed_Total ~ Squrt_Effective_Area + Number_of_Bedroom +
##           Number_of_Baths_Total + Condition_Description + Zone_Category +
##           Effective_Age
## Model 2: Log_Assessed_Total ~ Squrt_Effective_Area + Number_of_Bedroom +
##           Number_of_Baths_Total + Condition_Description + Zone_Category +
##           Effective_Age + `dist_Convenience Store` + dist_Other + `dist_Grocery Store` +
##           dist_Supermarket_store + total_1_mile + total_2_miles + total_10_miles
##   Res.Df   RSS Df Sum of Sq    F    Pr(>F)
## 1    9400 399.66
## 2    9393 325.44  7     74.219 306.03 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Results

The results of our baseline model show that traditional property features like square root of effective area, number of bathrooms, property condition, zoning categories, and effective age explain a significant portion of the variation in the log of total assessed value. The adjusted R-squared value of 0.8005 indicates that about 80% of the variability is explained by these features, while the residual standard error of 0.2062 suggests that some variation remains unexplained, leaving room for further improvement.

In the proposed model, the adjusted R-squared increases to 0.8374, reflecting an improvement of over 3%. While this change suggests that store-related variables add some explanatory power, the modest improvement indicates that proximity and the number of stores may not be as important in determining property values as initially anticipated. This calls for further analysis to explore more nuanced interactions between store location and housing values.

Proximity to convenience stores, grocery stores, and supermarkets shows significant associations with property values, with convenience stores having a positive effect—suggesting that properties farther from these stores are more valuable. Additionally, the number of stores within 1 mile has a negative association with property values, while those within 2 miles show a positive relationship. These results suggest that moderate distances from stores are more desirable for homeowners. However, store proximity may not be the most critical variable for assessing property values, and other factors likely have a larger influence.

The ANOVA comparison shows that the inclusion of store-related variables significantly improves the model ($F = 306.03$, $p < 2.2e-16$). However, the modest increase in explanatory power suggests that we may be missing key factors—such as neighborhood amenities, services, or broader socioeconomic conditions—that could better explain the variation in property values.

In conclusion, while store proximity and count are relevant, their impact on property values appears limited in the context of New Haven, and there may be more important variables that warrant further exploration.

Limitations of the Model and Analysis

One key limitation in our model diagnostics is the Q-Q plot, which shows that the residuals do not align well with the theoretical normal distribution. This deviation from normality may suggest that the model's assumptions are not fully met, potentially affecting the reliability of our inferences. Similarly, the Scale-Location plot indicates heteroscedasticity, meaning that the variance of residuals is not constant across fitted values, which can impact the accuracy of coefficient estimates and standard errors.

Multicollinearity is another concern in our model. Given the inclusion of multiple highly related predictors—such as distances to different types of stores and store counts within varying radii—there may be correlation among these variables. This can inflate the standard errors of the coefficients and make it difficult to identify the individual effect of each variable. We did not conduct a formal test for multicollinearity (such as VIF), but this should be considered in future work.

Another limitation stems from the observational nature of the dataset. Since we are working with existing property and store data, it is difficult to make causal statements about the relationship between proximity to stores and property values. Without experimental or quasi-experimental design, we cannot rule out unobserved confounders that could bias the results, such as local amenities, crime rates, or public services, which may also influence property values.

Lastly, our analysis focuses on a specific geographic area—New Haven, CT—so the generalizability of our findings may be limited. Property dynamics in other regions may differ, especially in areas with different urban layouts or economic conditions. Moreover, our store classification could be refined further, as some types of stores may impact property values in more nuanced ways than others. These limitations should be kept in mind when interpreting our results and considering future research.

Conclusion

In this study, we explored how proximity to food retailers, such as grocery stores and supermarkets, influences the total assessed value of single-family homes in New Haven, CT. By combining property parcel data with store location data from the Connecticut SNAP dataset, we aimed to understand how convenience affects property values. Our findings suggest that while proximity to stores is important, its relationship with property value is more nuanced than expected.

The results show that homes farther from convenience stores tend to have higher assessed values, which may reflect a preference for peaceful environments away from high-traffic retail areas. Having too many stores within close proximity, particularly within 1 mile, is associated with lower property values. This pattern aligns with earlier exploratory findings, suggesting that store proximity alone may not drive property values.

While store proximity and count improved the model's performance, their impact remains context-dependent. Proximity to stores may seem important at first glance, but further analysis suggests that other factors—such as crime rates, neighborhood services, or economic conditions—may play a more significant role in determining property values. Our model may also be missing key influences, which could be explored in future research.

Further investigation could help reveal a more comprehensive understanding of the factors that drive property values in urban areas beyond just retail proximity.

Sessioninfo

```
sessionInfo()

## R version 4.3.2 (2023-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.5
##
## Matrix products: default
## BLAS:    /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK:  /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics   grDevices utils      datasets   methods    base
##
## other attached packages:
## [1] webshot_0.5.5 leaflet_2.2.1 sf_1.0-15
##
## loaded via a namespace (and not attached):
## [1] jsonlite_1.8.8          highr_0.10           dplyr_1.1.4
## [4] compiler_4.3.2          tidyselect_1.2.0     Rcpp_1.0.12
## [7] callr_3.7.3            jquerylib_0.1.4     scales_1.3.0
## [10] yaml_2.3.8             fastmap_1.1.1       R6_2.5.1
## [13] generics_0.1.3         classInt_0.4-10     knitr_1.45
## [16] htmlwidgets_1.6.4       tibble_3.2.1        units_0.8-5
## [19] leaflet.providers_2.0.0 munsell_0.5.0      DBI_1.2.0
## [22] pillar_1.9.0           rlang_1.1.4         utf8_1.2.4
## [25] xfun_0.41              cli_3.6.2           magrittr_2.0.3
## [28] ps_1.7.6                class_7.3-22        crosstalk_1.2.1
## [31] processx_3.8.3         digest_0.6.34      grid_4.3.2
## [34] rstudioapi_0.15.0       lifecycle_1.0.4     vctrs_0.6.5
## [37] KernSmooth_2.23-22     proxy_0.4-27       evaluate_0.23
## [40] glue_1.7.0              farver_2.1.1       colorspace_2.1-0
## [43] fansi_1.0.6             e1071_1.7-14      rmarkdown_2.25
## [46] ellipsis_0.3.2          tools_4.3.2        pkgconfig_2.0.3
## [49] htmltools_0.5.7
```