

# S&DS 625: Housing Project

Matt Shu, Peter Yun, Eugene Han

2024-10-15

## Contents

<b>Introduction</b>	<b>1</b>
<b>Data Preprocessing and Exploration</b>	<b>1</b>
Subset Construction . . . . .	1
Base Features . . . . .	9
Food Access Features . . . . .	19
<b>Modeling</b>	<b>36</b>
Baseline Model . . . . .	36
Model Results . . . . .	41
Model Comparison and Discussion . . . . .	41
Limitations of the Model and Analysis . . . . .	42
<b>Conclusion</b>	<b>43</b>

## 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(grep("FAMILY", dnh$State_Use_Description) &
                     !grep("[2-9]", dnh$State_Use_Description) &
                     !grep("TWO", dnh$State_Use_Description) &
                     !grep("THREE", dnh$State_Use_Description) &
                     !grep("FOUR", dnh$State_Use_Description) &
                     !grep("FIVE", dnh$State_Use_Description) &
                     !grep("SIX", dnh$State_Use_Description) &
                     !grep("MULTI", dnh$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 (abbreviated as SFR), 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(grepl("MDL-01", dnh$State_Use_Description) &
                     !grepl("APT", dnh$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.

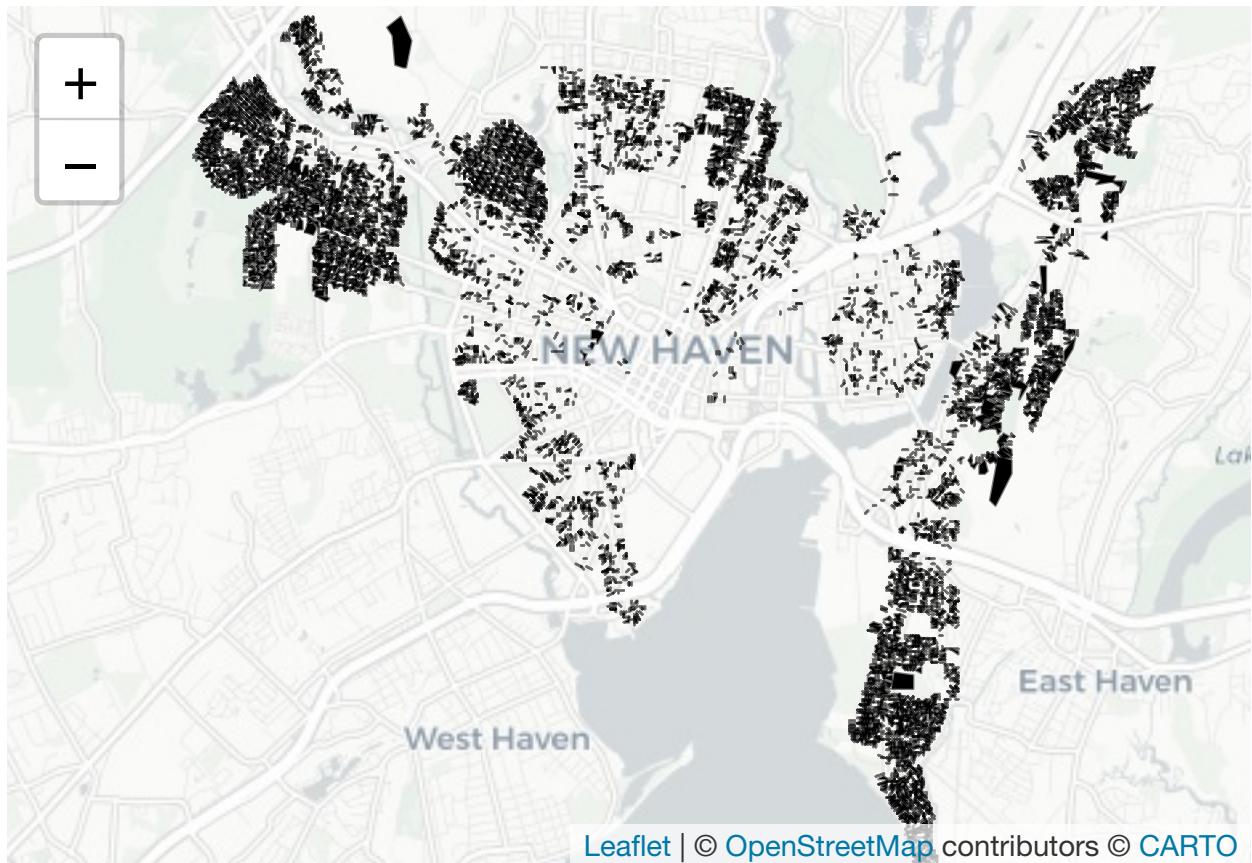
```
# List locations with more than one property
sfh[which(duplicated(sfh$Location)), ]$Location
```

```
## [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.

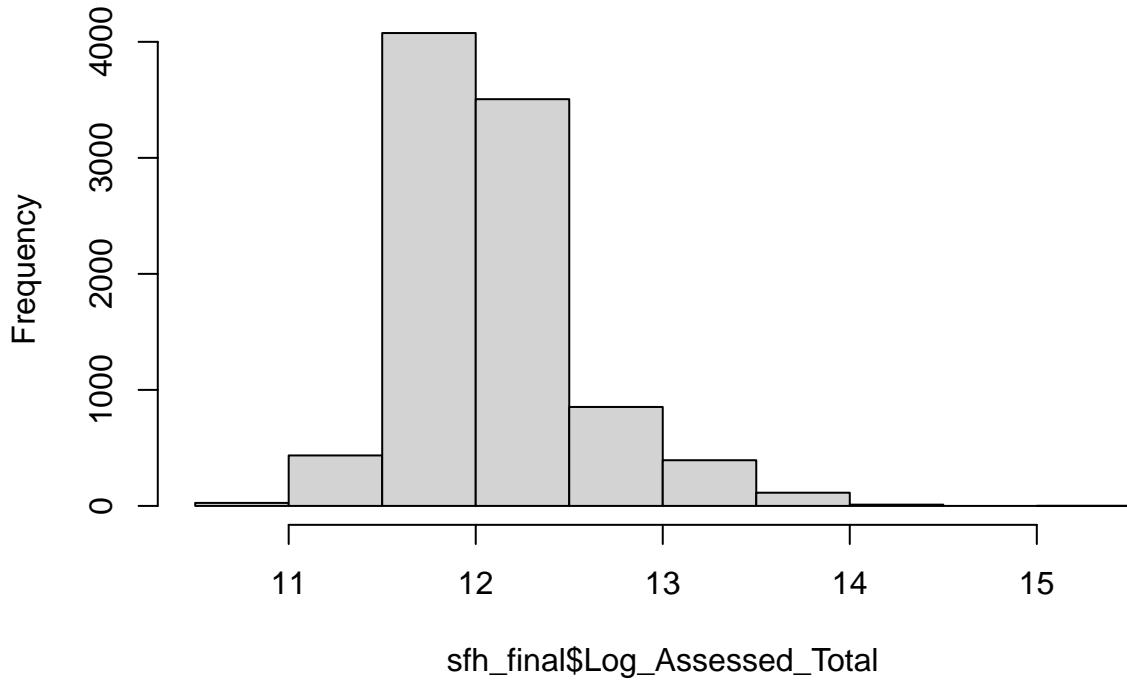
Since we're dealing with price data in our response, we will work with the logarithm of assessed total value.

```
# 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

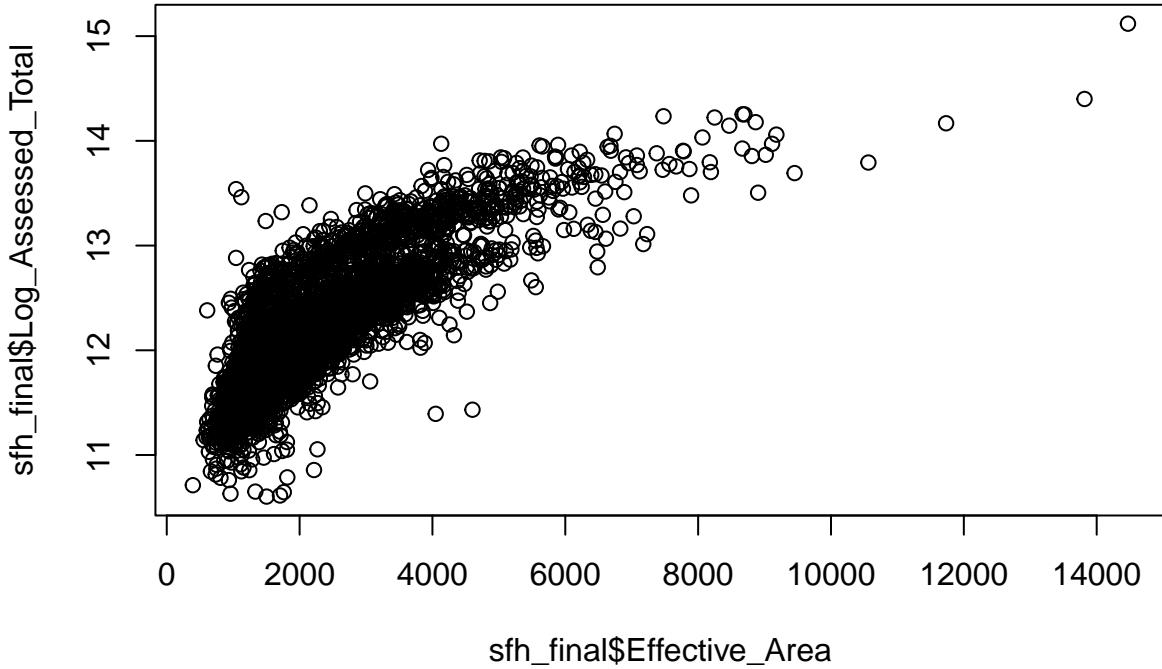


Next, we'll take a look at Effective Area. When plotting vs. log assessed total, we see that using a square root transformation results in a more linear relationship in the scatterplot while still having a natural interpretation.

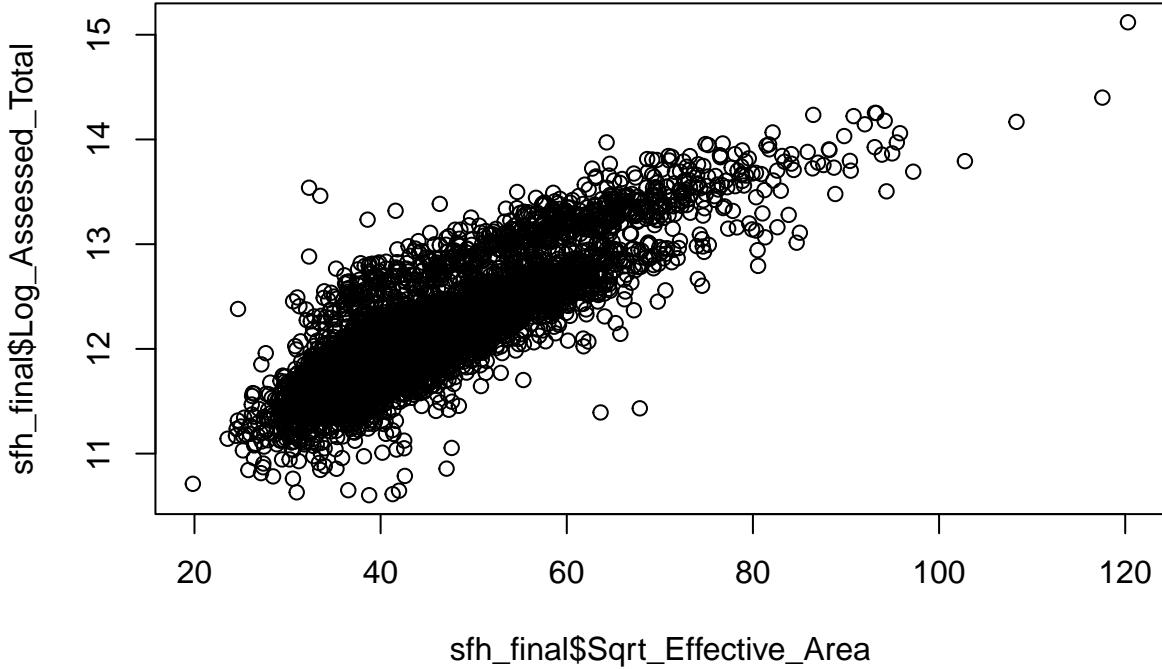
```
# Check for missing values
sum(is.na(sfh_final$Effective_Area))

## [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")

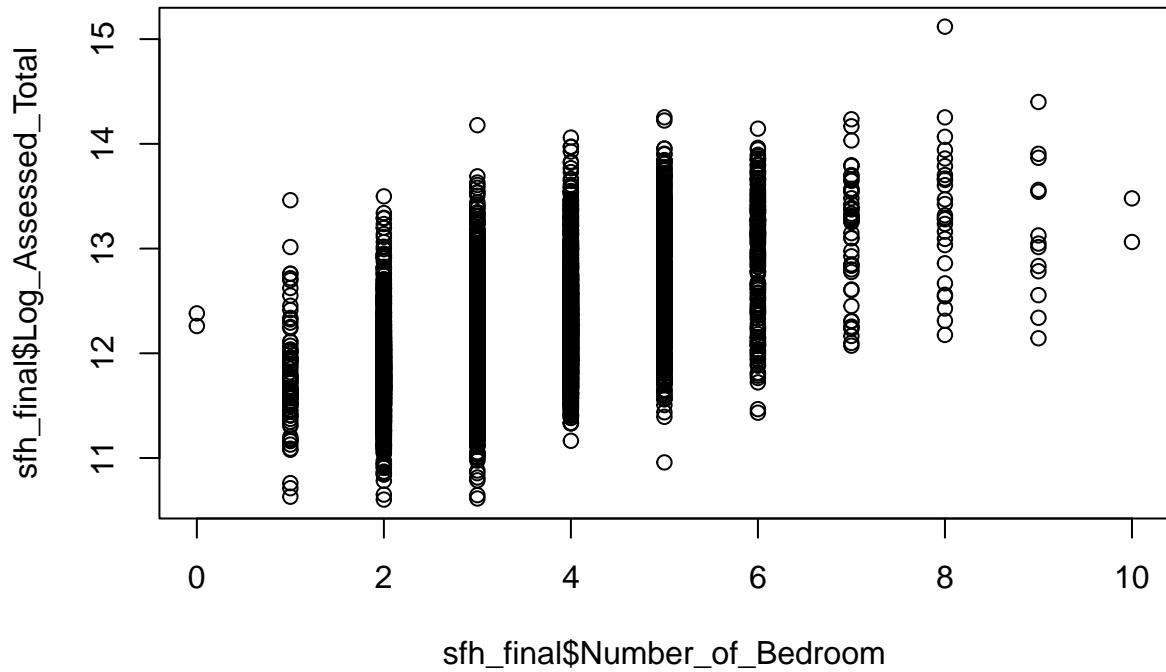
##          0      1      2      3      4      5      6      7      8      9      10 <NA>
##      5    89 1585 5224 1747   477   195    52    26    13     2     1

# 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		

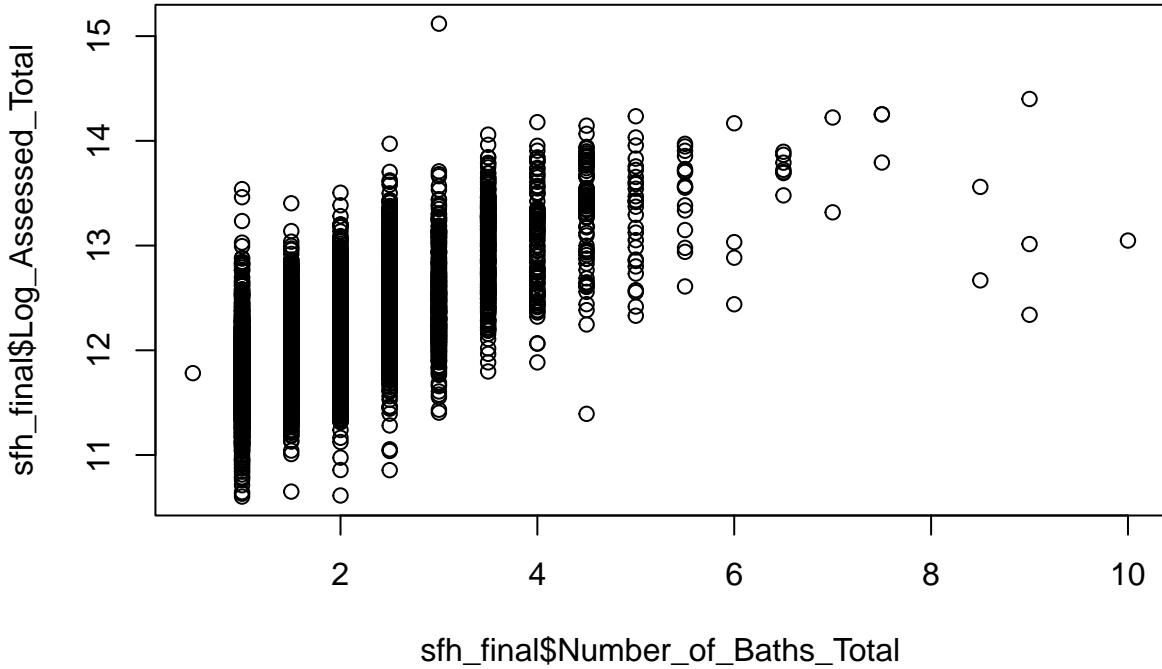
```

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")
```

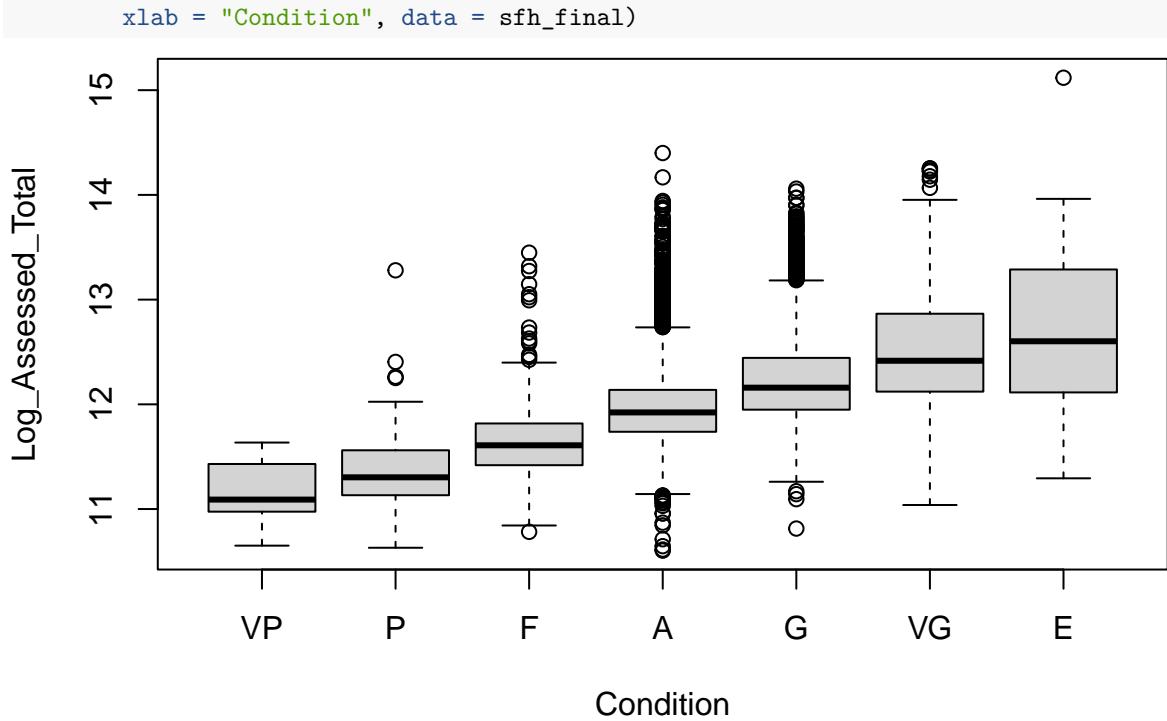
	Average	Excellent	F	Good	Poor	U	Very Good	Very Poor	<NA>
A	5216	0	0	0	0	0	0	0	0
E	0	77	0	0	0	0	0	0	0
F	0	0	326	0	0	0	0	0	0
G	0	0	0	3304	0	0	0	0	0
P	0	0	0	0	37	0	0	0	0
U	0	0	0	0	0	2	0	0	0
VG	0	0	0	0	0	0	441	0	0
VP	0	0	0	0	0	0	0	13	0
<NA>	0	0	0	0	0	0	0	0	0

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")
```

```

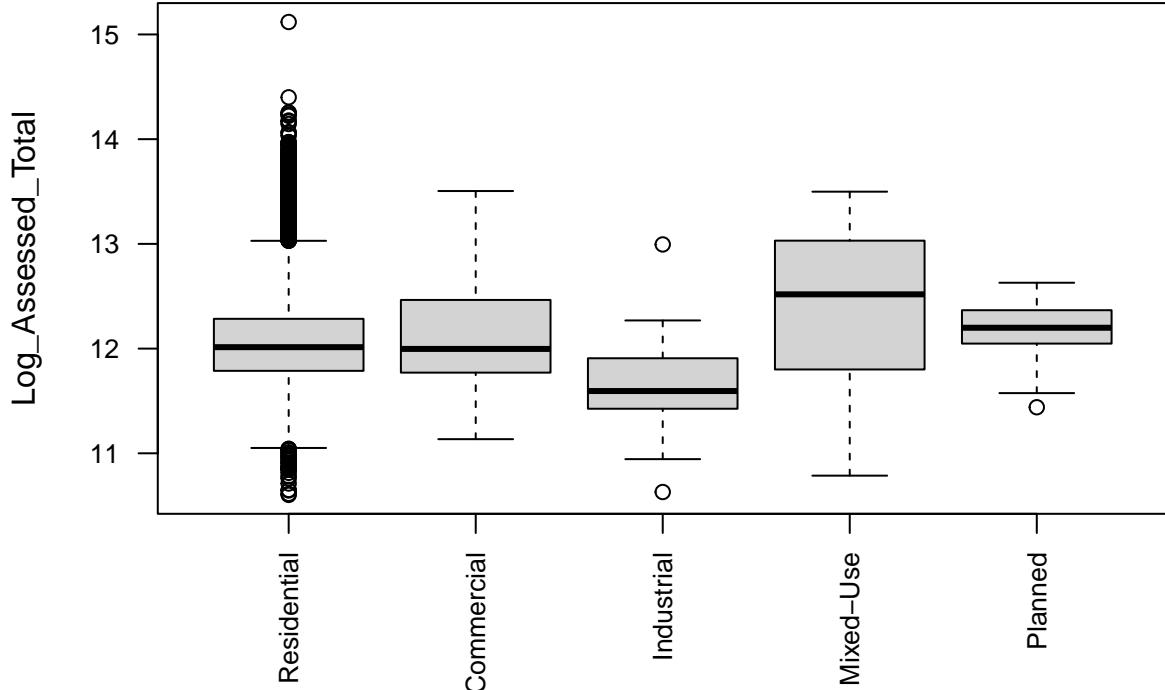
commercial <- c("BA", "BA1", "BB", "BC", "BD", "BD1")
industrial <- c("IL", "IH", "IH/RM2")
planned_development <- c("PDD 119", "PDD 26", "PDD 39", "PDD 49", "PDD 52",
                         "PDU 102", "PDU 106", "PDU 108", "PDU 16", "PDU 72",
                         "PDU 75", "PDU 95")

# 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 = "")

```



Now, we check the table of zone categories

```
table(sfh_final$Zone_Category)
```

```
##
```

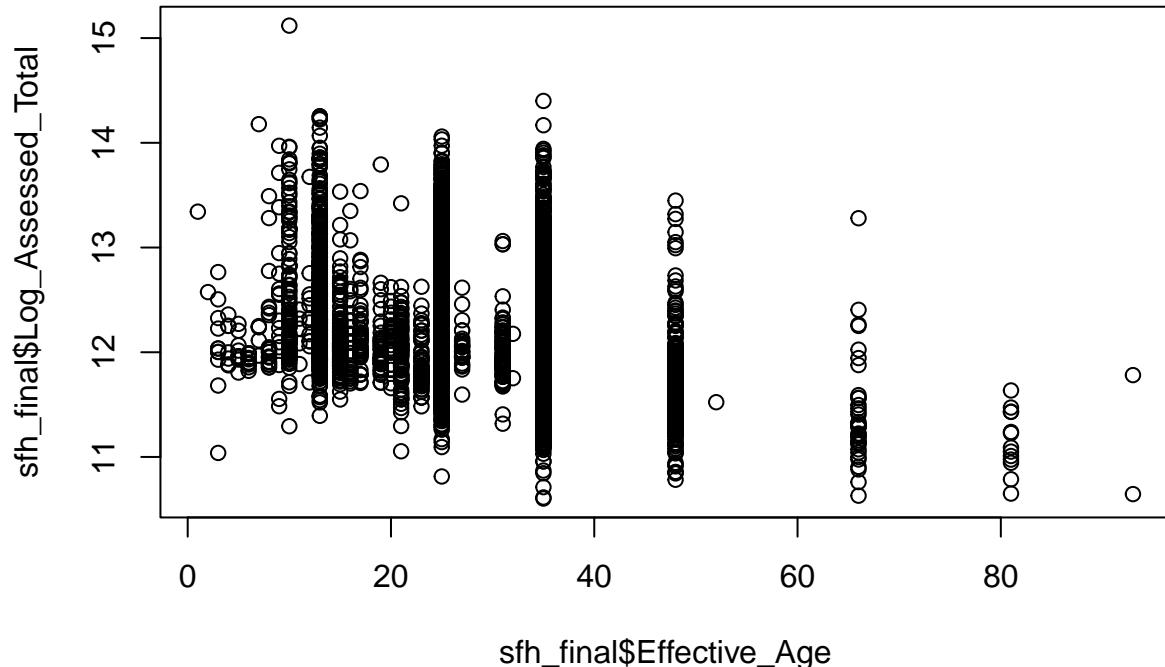
```
## Residential Commercial Industrial Mixed-Use Planned
##         9184        89        32        34        77
```

Note that most properties fall under the Residential category, which is expected for single-family homes. However, there are a decent number of houses that lie in the other categories.

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")

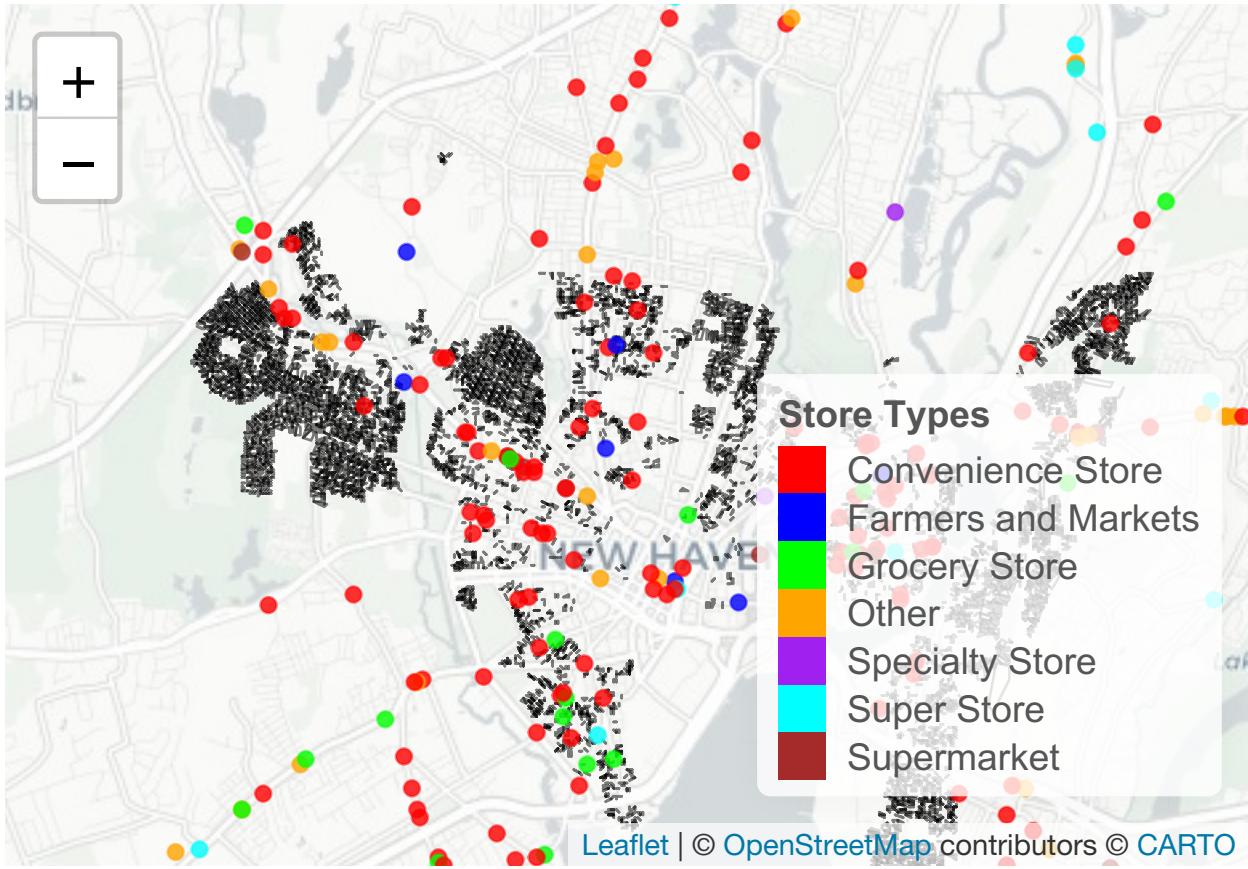
# Convert to spatial
sg <- st_as_sf(s, coords = c("Longitude", "Latitude"), crs = 4326, remove=FALSE)
sfh_final <- st_transform(sfh_final, crs = 4326)

# 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 = 12) %>%

  # 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
  )
```



We can contextualize this by looking at the number of stores for each type.

```
# View counts by store type
table(sg$Store.Type)

##          Convenience Store Farmers and Markets      Grocery Store           Other
##                1108                      94                  255                 570
##          Specialty Store        Super Store      Supermarket
##                   25                      224                  194
```

Let's now zoom into the stores that are relatively nearby New Haven

```
# Found via Google Maps, cross referencing with Leaflet plot
min_lon <- -73.06
max_lon <- -72.78
min_lat <- 41.12
max_lat <- 41.39

names(sg)

## [1] "Record.ID"              "Store.Name"            "Store.Street.Address"
## [4] "Additional.Address"     "City"                  "State"
## [7] "Zip.Code"               "Zip4"                 "County"
## [10] "Store.Type"             "Latitude"              "Longitude"
## [13] "Incentive.Program"      "Grantee.Name"          "geometry"

sg_nh <- sg[((sg$Longitude > min_lon) & (sg$Longitude < max_lon) &
              (sg$Latitude > min_lat) & (sg$Latitude < max_lat)), ]
```

```



```

Some store types like specialty stores are relatively infrequent and the “Other” category remains quite common for stores around New Haven. So, we explore these stores further to determine the categories we use in our model.

```

# We start creating the new column.
sg$Store.Type.Grouped <- "Other"

```

There aren't that many specialty stores in and around New Haven. Let's look at them.

```

st_drop_geometry(sg_nh[sg_nh$Store.Type == "Specialty Store",
                     c("Store.Name", "Store.Street.Address", "City")])

```

```

##          Store.Name Store.Street.Address           City
## 717          #1 FISH MARKET       2239 State St    Hamden
## 1379 STATE FISH FARM MARKET     892 State St New Haven
## 1626   Bimbo Bakeries 6007     284 Bull Hill Ln Orange

```

It looks like these are stores that specialize in fish or bread. Let's now look at farmers and markets. We use head going forward for brevity.

```

head(st_drop_geometry(sg_nh[sg_nh$Store.Type == "Farmers and Markets",
                           c("Store.Name", "Store.Street.Address", "City")]))

```

```

##          Store.Name Store.Street.Address           City
## 541          East Haven Farmers Market      100 River St East Haven
## 735          Hindinger Farm Llc       810 Dunbar Hill Rd Hamden
## 1299         Cityseed Downtown Farmers Market 145 Orange St New Haven
## 1300         Cityseed Edgewood Park Farmers Market 381 W Rock Ave New Haven
## 1301          Cityseed Q House Farmers Market 197 Dixwell Ave New Haven
## 1302  CitySeed Wooster Square Farmers Market 511 Chapel St New Haven

```

These are farmers markets and farm stands. We choose to combine these two categories as they both offer a certain type of ‘specialized’ goods.

```

sg$Store.Type.Grouped[sg$Store.Type %in%
                     c("Specialty Store",
                       "Farmers and Markets")] <- "Specialty/Farmers"

```

Let's now look at convenience stores.

```

head(st_drop_geometry(sg_nh[sg_nh$Store.Type == "Convenience Store",
                           c("Store.Name", "Store.Street.Address", "City")]))

```

```

##          Store.Name Store.Street.Address           City
## 88  350 Foxon Food Mart LLC 01      202 E Main St Branford
## 92          Caron's Corner      147 Montowese St Branford
## 93  Cumberland Farms Inc 4751      227 East Main St Branford
## 97          Jhan Food Mart Llc N/A     380 E Main St Branford
## 98          Krauszers Food Store      170 Cedar St Branford
## 533          Alex Grocery          418 Main St East Haven

```

These generally look coherent as a category and there aren't similar groups that we believed should be combined with them.

```
sg$Store.Type.Grouped[sg$Store.Type == "Convenience Store"] <- "Convenience Store"
```

Let's now look at grocery, supermarket, and superstores.

```
head(st_drop_geometry(sg_nh[sg_nh$Store.Type == "Grocery Store",
                           c("Store.Name", "Store.Street.Address", "City")]))
```

	Store.Name	Store.Street.Address	City
## 96	Four Seasons Market	875 W Main St	Branford
## 739	Lina's Place	119 Sanford St	Hamden
## 1291	ANNIES DELI GROCERY	188 Rosette St	New Haven
## 1298	CAYEY MARKET	155 Kimberly Ave	New Haven
## 1305	Congress Market	620 Congress Ave	New Haven
## 1317	Eddys Food Center	276 Howard Ave	New Haven

Grocery stores look like relatively small stores that sell a variety of goods, including fresh produce.

```
head(st_drop_geometry(sg_nh[sg_nh$Store.Type == "Super Store",
                           c("Store.Name", "Store.Street.Address", "City")]))
```

	Store.Name	Store.Street.Address
## 90	Big Y Supermarkets	12 1060 Main St
## 103	Walmart	2232 120 Commercial Pkwy
## 454	Big Y World Class Market	117 656 New Haven Ave
## 542	East Haven Markets LLC/Shoprite Of East Haven	745 Foxon Rd
## 554	T&J Supermarket Inc	875 N High St
## 556	Walmart SUPERCENTER	5439 315 Foxon Rd
##	City	
## 90	Branford	
## 103	Branford	
## 454	Derby	
## 542	East Haven	
## 554	East Haven	
## 556	East Haven	

```
head(st_drop_geometry(sg_nh[sg_nh$Store.Type == "Supermarket",
                           c("Store.Name", "Store.Street.Address", "City")]))
```

	Store.Name	Store.Street.Address	City
## 89	Aldi	73116 1151 W Main St	Branford
## 551	Stop & Shop	692 370 Hemingway Ave	East Haven
## 719	Aldi	73074 2300 Dixwell Ave	Hamden
## 745	Stop & Shop	600 2335 Dixwell Ave	Hamden
## 1108	Adam's IGA Supermarket	425 1391 New Haven Ave	Milford
## 1137	Stop & Shop	663 1360 E Town Rd	Milford

Meanwhile supermarkets and superstores are much larger stores based on our knowledge of the stores and online searches. We choose to leave grocery x leave grocery stores unchanged and combine these two categories as a super store is much like a supermarket that happens to also have some non-food items.

```
sg$Store.Type.Grouped[sg$Store.Type == "Grocery Store"] <- "Grocery Store"
sg$Store.Type.Grouped[sg$Store.Type %in%
                      c("Supermarket", "Super Store")] <- "Supermarket/Store"
```

Let's look at stores categorized as 'Other' to assign these stores into groups that make more sense.

```
# Look at what's currently in "Other". We show head here for brevity but looked
# at all stores in and around New Haven to create our groupings.
head(st_drop_geometry(sg_nh[sg_nh$Store.Type == "Other",
                           c("Store.Name", "Store.Street.Address", "City")]))
```

	Store.Name	Store.Street.Address	City
## 94	CVS PHARMACY 172	2-6 Short Beach Rd	Branford
## 95	DOLLAR TREE 3550	869 W Main St	Branford
## 101	Walgreens 03944	329 E Main St	Branford
## 102	Walgreens 10309	1036 W Main St	Branford
## 453	Big Lots 1783	600 New Haven Ave	Derby
## 457	CVS PHARMACY 1044	656 New Haven Ave	Derby

Many are drug stores, we include a group for that.

```
drugs_pttn <- "CVS|RITE AID|WALGREENS"
head(st_drop_geometry(sg_nh[sg_nh$Store.Type == "Other" &
                           grep1(drugs_pttn, sg_nh$Store.Name,
                                 ignore.case = TRUE),
                           c("Store.Name", "Store.Street.Address", "City")]))
```

	Store.Name	Store.Street.Address	City
## 94	CVS PHARMACY 172	2-6 Short Beach Rd	Branford
## 101	Walgreens 03944	329 E Main St	Branford
## 102	Walgreens 10309	1036 W Main St	Branford
## 457	CVS PHARMACY 1044	656 New Haven Ave	Derby
## 536	CVS PHARMACY 1968	369 Main St	East Haven
## 537	CVS Pharmacy 857	660 Foxon Rd	East Haven

```
sg$Store.Type.Grouped[sg$Store.Type == "Other" &
                           grep1(drugs_pttn,
                                 sg$Store.Name,
                                 ignore.case = TRUE)] <- "Drug Store"
```

Now let's group discount stores, which include Ocean State Job Lot, Dollar Tree, Family Dollar, and Dollar General.

```
dscnt_pttn <- "Ocean State Job Lot|Dollar|Big Lots"
head(st_drop_geometry(sg_nh[sg_nh$Store.Type == "Other" &
                           grep1(dscnt_pttn, sg_nh$Store.Name,
                                 ignore.case = TRUE),
                           c("Store.Name", "Store.Street.Address", "City")]))
```

	Store.Name	Store.Street.Address	City
## 95	DOLLAR TREE 3550	869 W Main St	Branford
## 453	Big Lots 1783	600 New Haven Ave	Derby
## 459	Dollar Tree 3850	656 New Haven Ave	Derby
## 534	Big Lots 4671	78 Frontage Rd	East Haven
## 539	Dollar General 16088	650 Coe Ave	East Haven
## 540	Dollar Tree 5246	32 Frontage Rd	East Haven

```
sg$Store.Type.Grouped[sg$Store.Type == "Other" &
                           grep1(dscnt_pttn,
                                 sg$Store.Name,
                                 ignore.case = TRUE)] <- "Discount Store"
```

What remains, for stores nearby New Haven, are only a few stores: GMart, Motherland Market, African

Queens Market, and T-asian Market

```
# Look at all stores that are still Other in Store.Type.Grouped and near
# New Haven. We re-subset sg to surface the nearby stores.
sg_nh <- sg[((sg$Longitude > min_lon) & (sg$Longitude < max_lon) &
              (sg$Latitude > min_lat) & (sg$Latitude < max_lat)), ]
st_drop_geometry(sg_nh[sg_nh$Store.Type.Grouped == "Other",
                      c("Store.Name",
                        "Store.Street.Address",
                        "City")])
```

```
##                               Store.Name Store.Street.Address      City
## 553          T-asian Market 0           692 Foxon Rd East Haven
## 741        Motherland Market       897 Dixwell Ave Hamden
## 1124            Gmart Llc       143 Cherry St Milford
## 2294 African Queens Market     489 Campbell Ave West Haven
```

Looking at CT in general, we see that this pattern of specialty/ethnic food stores actually fits all remaining stores. So, we group them as part of Specialty/Farmers.

```
head(st_drop_geometry(sg[sg$Store.Type.Grouped == "Other",
                      c("Store.Name",
                        "Store.Street.Address",
                        "City")]))
```

```
##                               Store.Name Store.Street.Address      City
## 48            Shayona Grocery Llc    279 Berlin Tpke Berlin
## 106          Africana Market Llc   2375 Main St Bridgeport
## 143        E & A Meat And Seafood Llc 933 E Main St Bridgeport
## 212 Rosadel's African Market. Llc 1761 Main St Bridgeport
## 319          Valley Indian Market   50 Albany Tpke Canton
## 382        Bombay Sweets & Spices    5 Padanaram Rd Danbury
sg$Store.Type.Grouped[sg$Store.Type.Grouped == "Other"] <- "Specialty/Farmers"
```

Now we have 6 store types and we can visualize them on the map.

```
table(sg$Store.Type.Grouped)
```

```
##
```

	Convenience Store	Discount Store	Drug Store	Grocery Store
##	1108	269	264	255
##	156	418		

```
sg_nh <- sg[((sg$Longitude > min_lon) & (sg$Longitude < max_lon) &
              (sg$Latitude > min_lat) & (sg$Latitude < max_lat)), ]
table(sg_nh$Store.Type.Grouped)
```

```
##
```

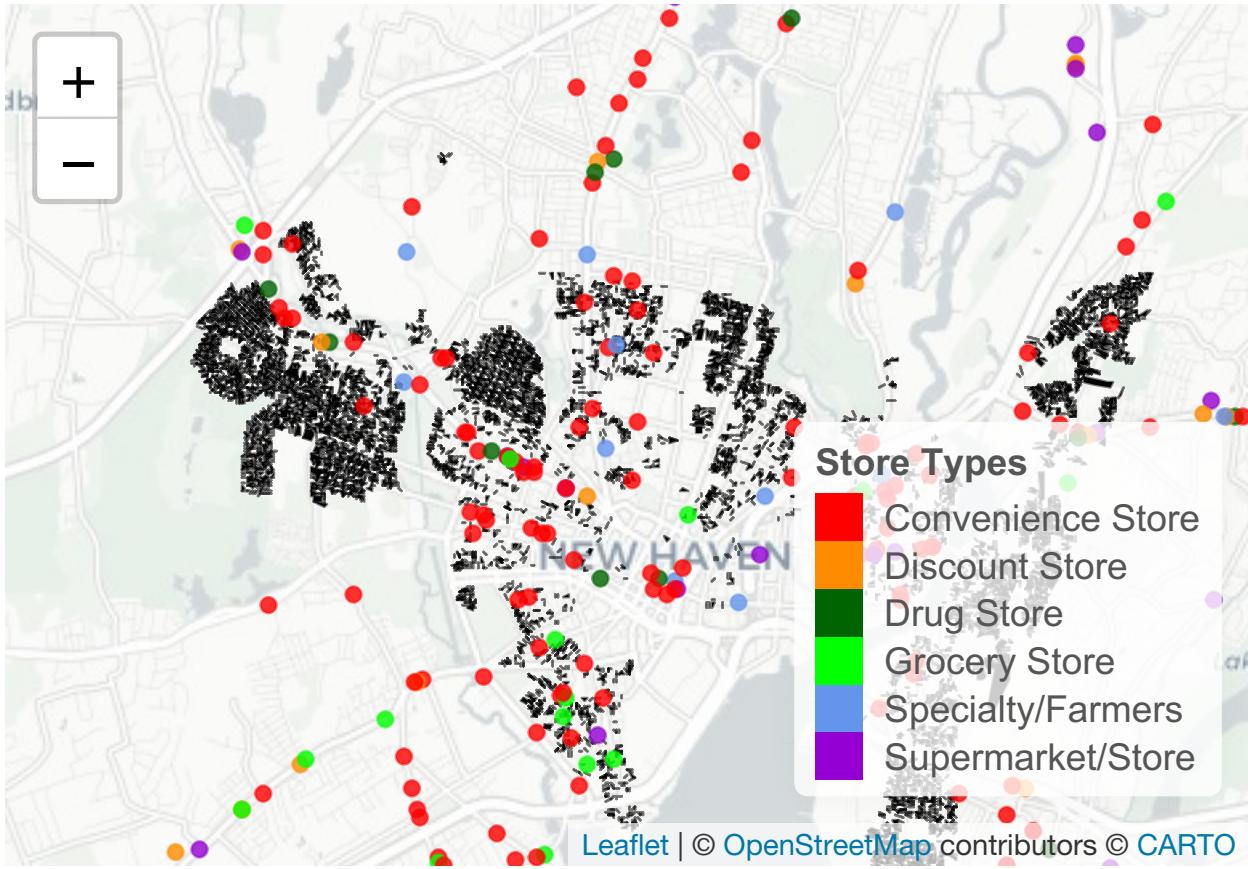
	Convenience Store	Discount Store	Drug Store	Grocery Store
##	154	29	26	30
##	18	41		

```
# Re-visualize with new groups
palGrouped <- colorFactor(
  palette = c("red", "darkorange", "darkgreen", "green",
             "cornflowerblue", "darkviolet"))
```

```

domain = sg$Store.Type.Grouped
)
leaflet() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  
  # Center on New Haven
  setView(lng = -72.93, lat = 41.32, zoom = 12) %>%
  
  # 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)
}

```

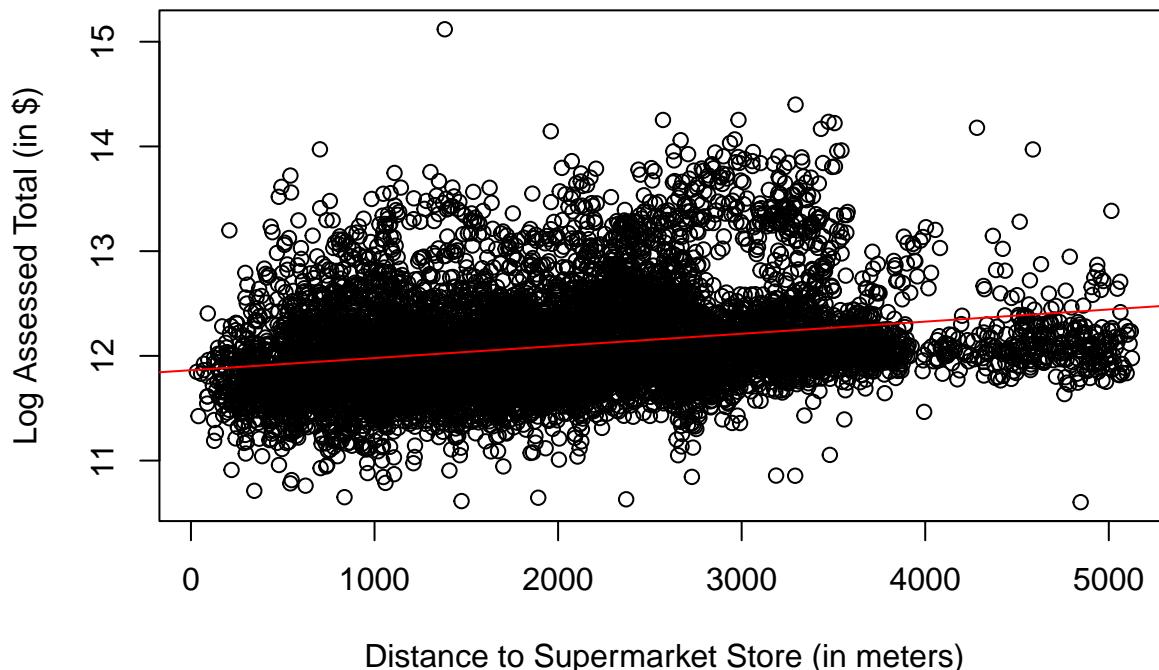
We now plot log assessed total vs. these new distance features

```

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

```

## Log Assessed Total vs. Distance to Supermarket Store

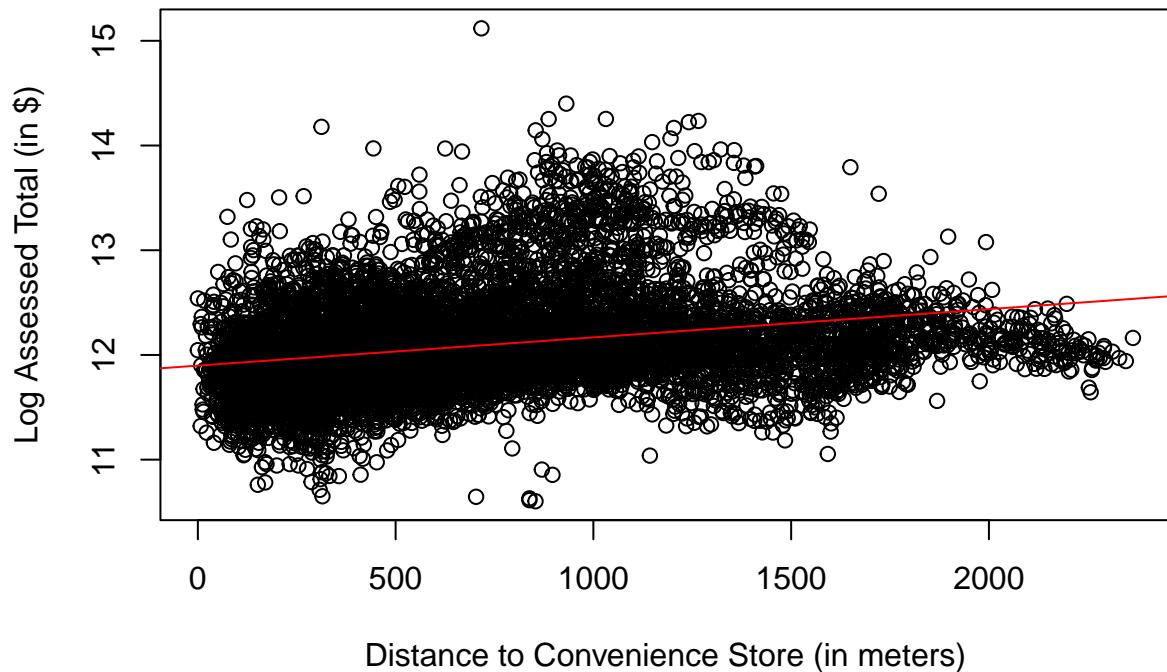


```

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

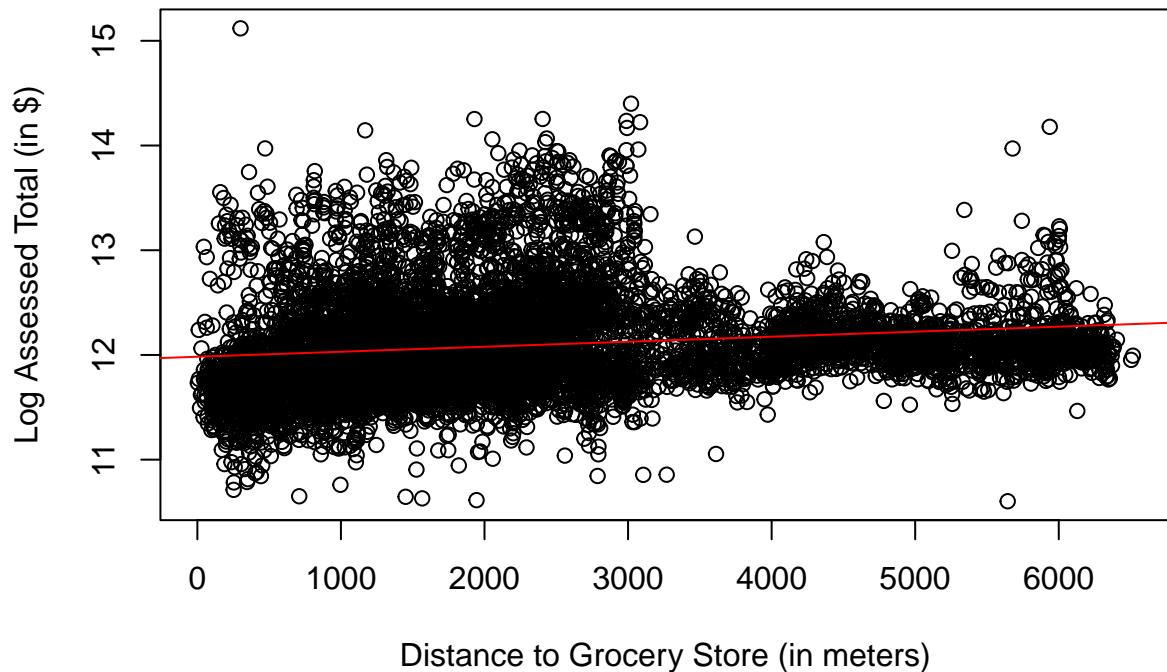
```

## Log Assessed Total vs. Distance to Convenience Store



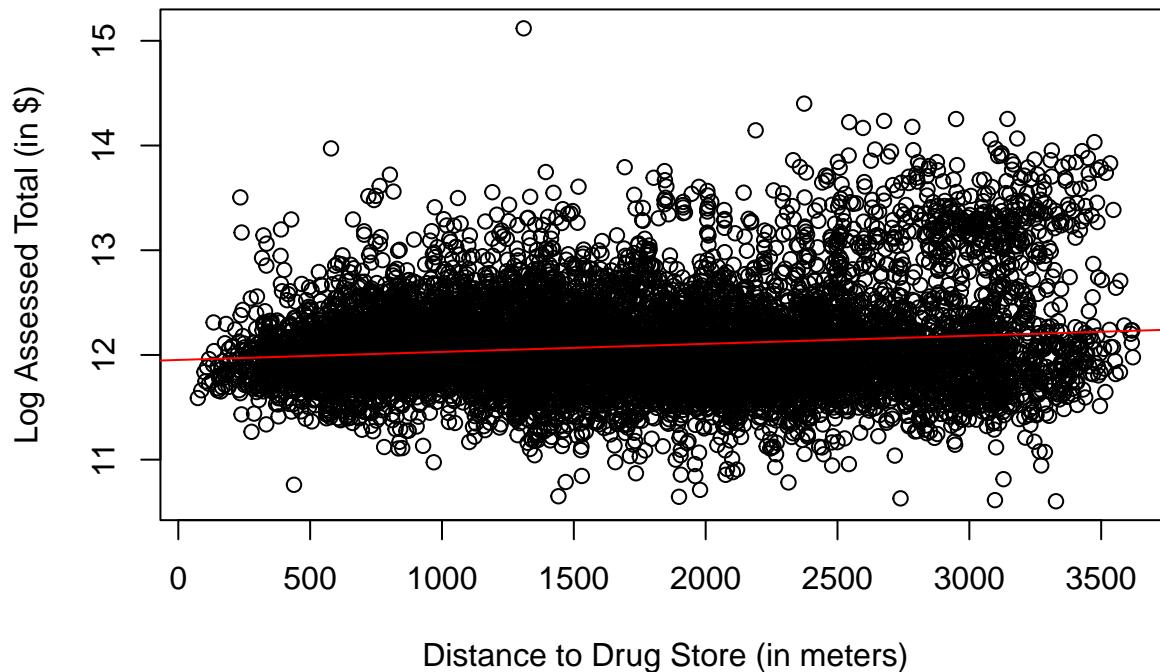
```
# Plot distance to Log Assessed Total vs. Distance to Grocery Store
gsdist_ls <- lm(Log_Assessed_Total ~ `dist_Grocery Store`, data = sfh_final)
plot(sfh_final$`dist_Grocery Store`, sfh_final$Log_Assessed_Total,
     main = "Log Assessed Total vs. Distance to Grocery Store",
     xlab = "Distance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

## Log Assessed Total vs. Distance to Grocery Store



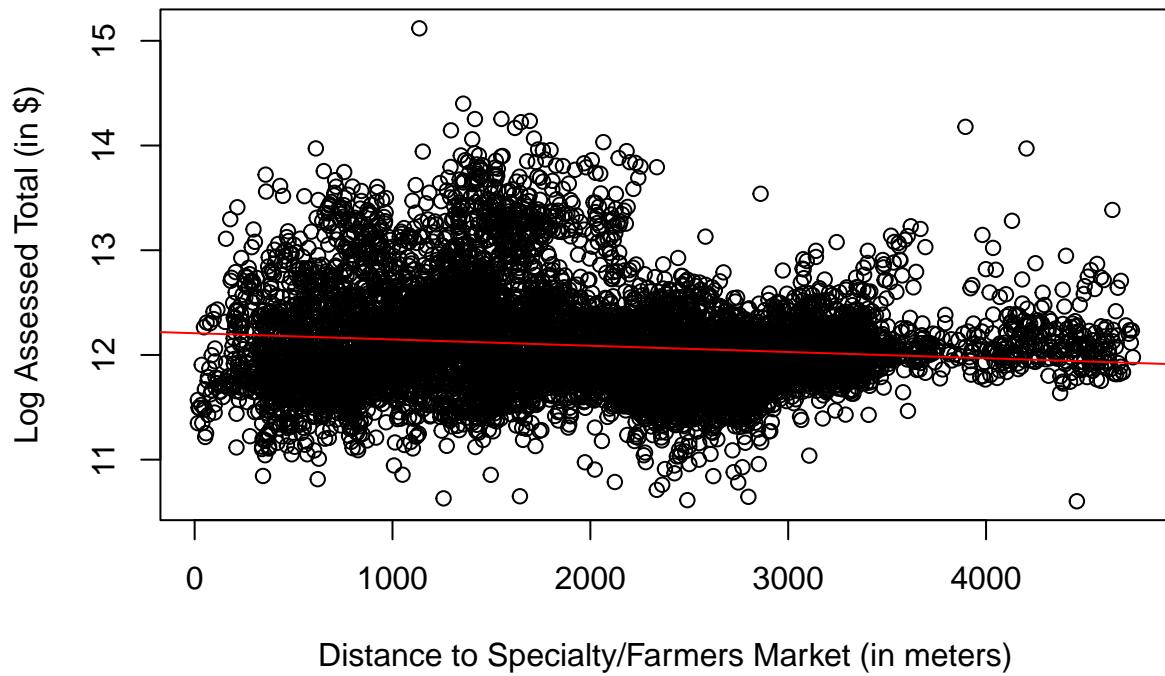
```
# Plot distance to Log Assessed Total vs. Distance to Drug Store
otdist_ls <- lm(Log_Assessed_Total ~ `dist_Drug Store`, data = sfh_final)
plot(sfh_final$`dist_Drug Store`, sfh_final$Log_Assessed_Total,
     main = "Log Assessed Total vs Distance to Drug Store",
     xlab = "Distance to Drug Store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

## Log Assessed Total vs Distance to Drug Store



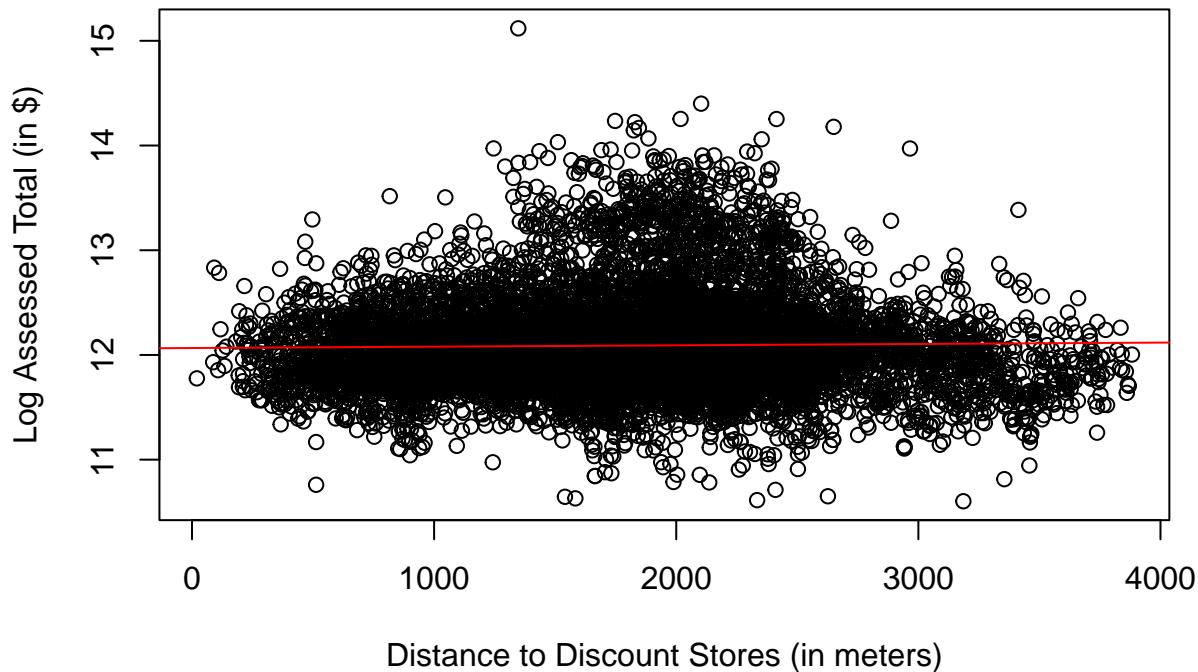
```
# Plot distance to Log Assessed Total vs. Distance to Specialty/Farmers Market
otdist_ls <- lm(Log_Assessed_Total ~ `dist_Specialty/Farmers`, data = sfh_final)
plot(sfh_final$`dist_Specialty/Farmers`, sfh_final$Log_Assessed_Total,
     main = "Log Assessed Total vs. Distance to Specialty/Farmers Market",
     xlab = "Distance to Specialty/Farmers Market (in meters)",
     ylab = "Log Assessed Total (in $)")
abline(otdist_ls, col = "red")
```

## Log Assessed Total vs. Distance to Specialty/Farmers Market



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

## Log Assessed Total vs. Distance to Discount Stores



Across all these plots, we observe a weak positive relationship between proximity to most store types and the log of the total assessed value of single-family homes. This effect seems strongest for supermarket and convenience stores, which was counterintuitive to our initial expectations that closer proximity to food retailers would be associated with higher values. Interestingly, the only store type with a weak negative trend is distance to specialty stores and farmers markets.

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 may be more likely found in more secluded areas with fewer stores. However, the negative trend for specialty stores and farmers markets may also provide a clue—perhaps the wealthy are more likely to prefer such specialized and organic produce.

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. Generic grocery stores in expensive areas might also 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. Locally, Elm City Market just closed and are planning to relocate because of such upkeep costs.

Overall, these marginal plots suggest that the relationship between store proximity and property values may be more complex than our initial hypothesis. ### Number of food retailers within some radius The other measure we use, instead of just closest distance to particular store types is capturing how many stores are around a given home. In this approach, we calculate the total number of stores present within a 1 mile, 2 mile, and 10 mile radius for each property. We chose these radii because they are used by the USDA to define food deserts.

```
# Compute distance matrix between each single family home and each snap store
dist_matrix <- st_distance(sfh_final, sg)

# ~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]))

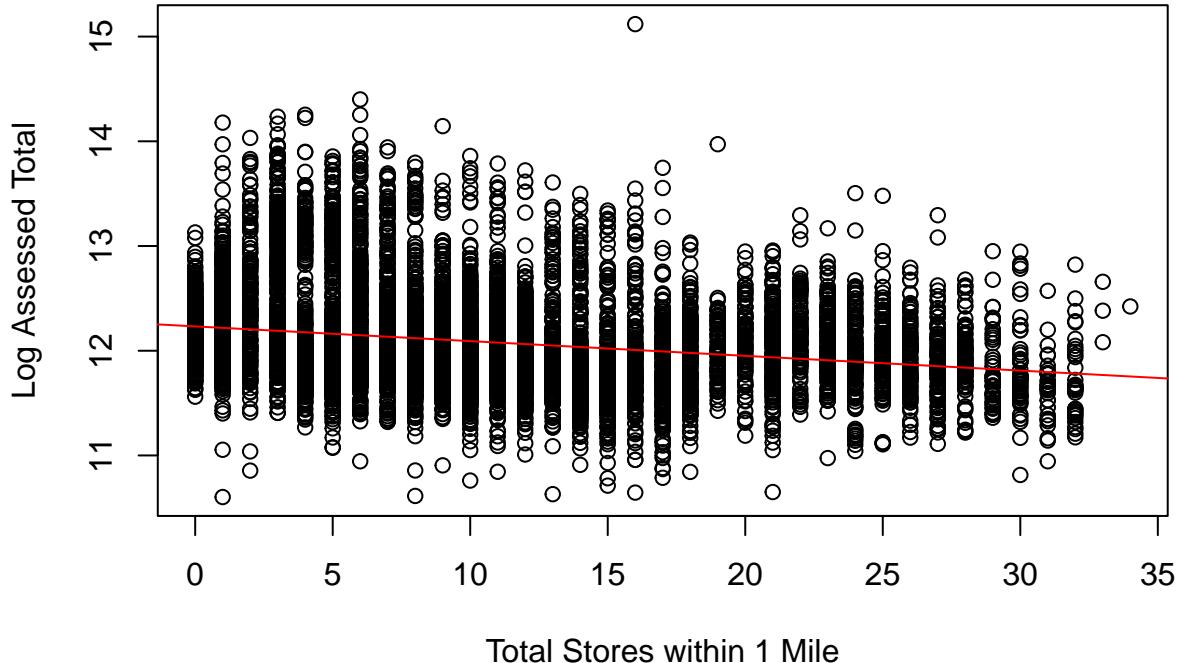
  # Add the counts to the data frame with a descriptive column name
  column_name <- paste0("total_", radii_labels[i])
  sfh_final[[column_name]] <- counts
}

# 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 = "Log Assessed Total vs. Total Stores within 1 Mile",
     xlab = "Total Stores within 1 Mile", ylab = "Log Assessed Total")
abline(mile1_ls, col = "red")

```

## Log Assessed Total vs. Total Stores within 1 Mile

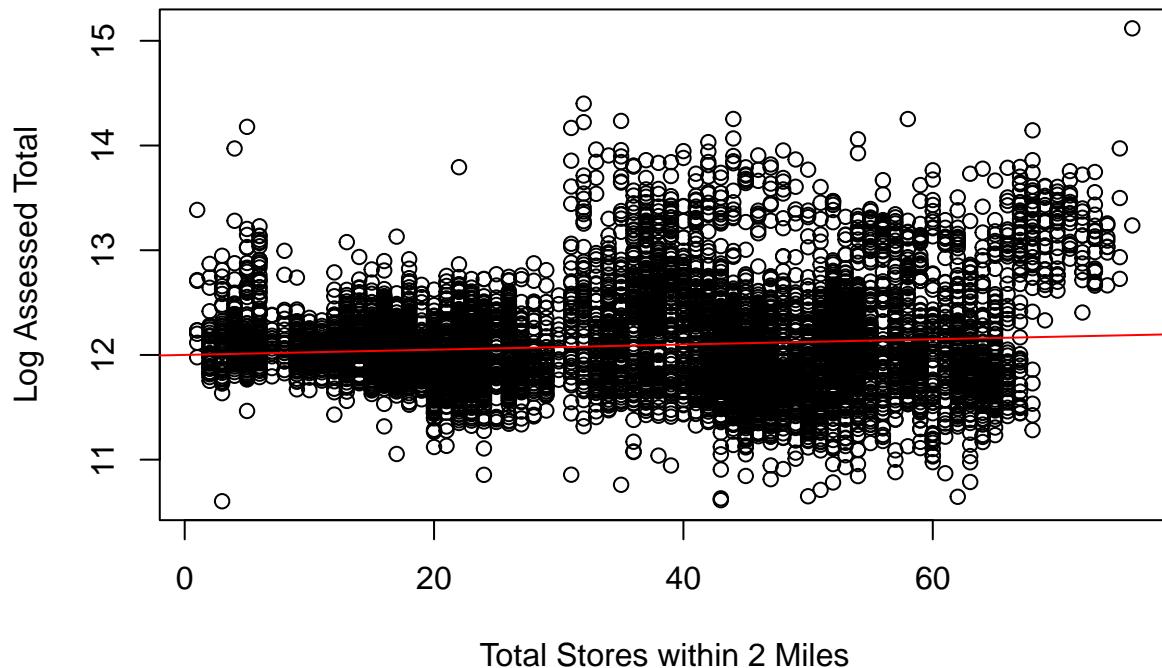


```

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

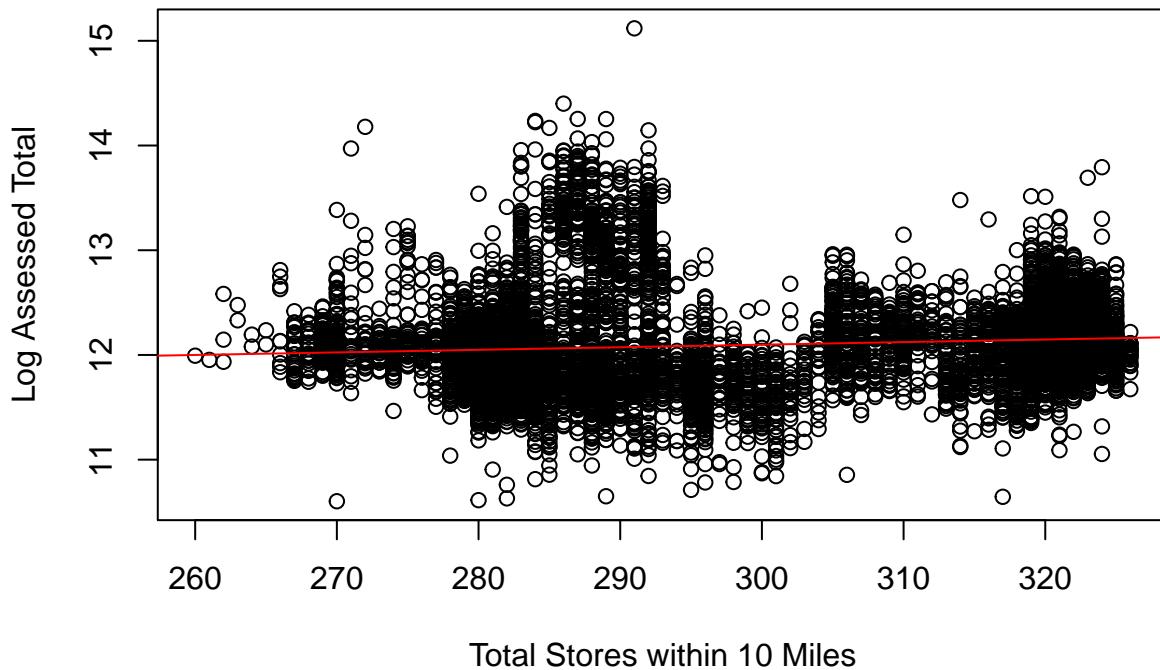
```

## Log Assessed Total vs. 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 = "Log Assessed Total vs. Total Stores within 10 Miles",
     xlab = "Total Stores within 10 Miles", ylab = "Log Assessed Total")
abline(mile10_ls, col = "red")
```

## Log Assessed Total vs. Total Stores within 10 Miles



Our analysis of the number of stores within specified radii reveals an unexpected pattern: while there is a weak but clear negative association between the total stores within 1 mile vs. log assessed total, there appears to be a slightly positive association between the total stores within 2 miles and log asssssed total that is repeated for stores within 10 miles. Note, however, that the equal variance assumption does not seem to hold for 2 miles and especially not for 10 miles.

The 1-mile radius graph is contrary to our initial hypothesis that more nearby stores would be associated with higher property values. However, it reinforces the earlier exploratory distance plots where nearer distances to most store types was weakly associated with lower property values.

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 that are very close. However, the 2-mile and 10-mile radius graphs may suggest that this store density penalty only applies for shorter radii and it is perhaps desirable to be moderately near a greater number of stores. Note though that these slight positive associations are not very clear in the data and could instead be negligible. Overall, these marginal plots suggest other factors are likely also at play.

# 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.

## Baseline Model

```
# 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
# Extract and back-transform RSE given response is log-transformed
exp(summary(m1)$sigma)

## [1] 1.228993
levels(sfh_final$Zone_Category)[1]

## [1] "Residential"
levels(sfh_final$Condition_Description)[1]

## [1] "Average"

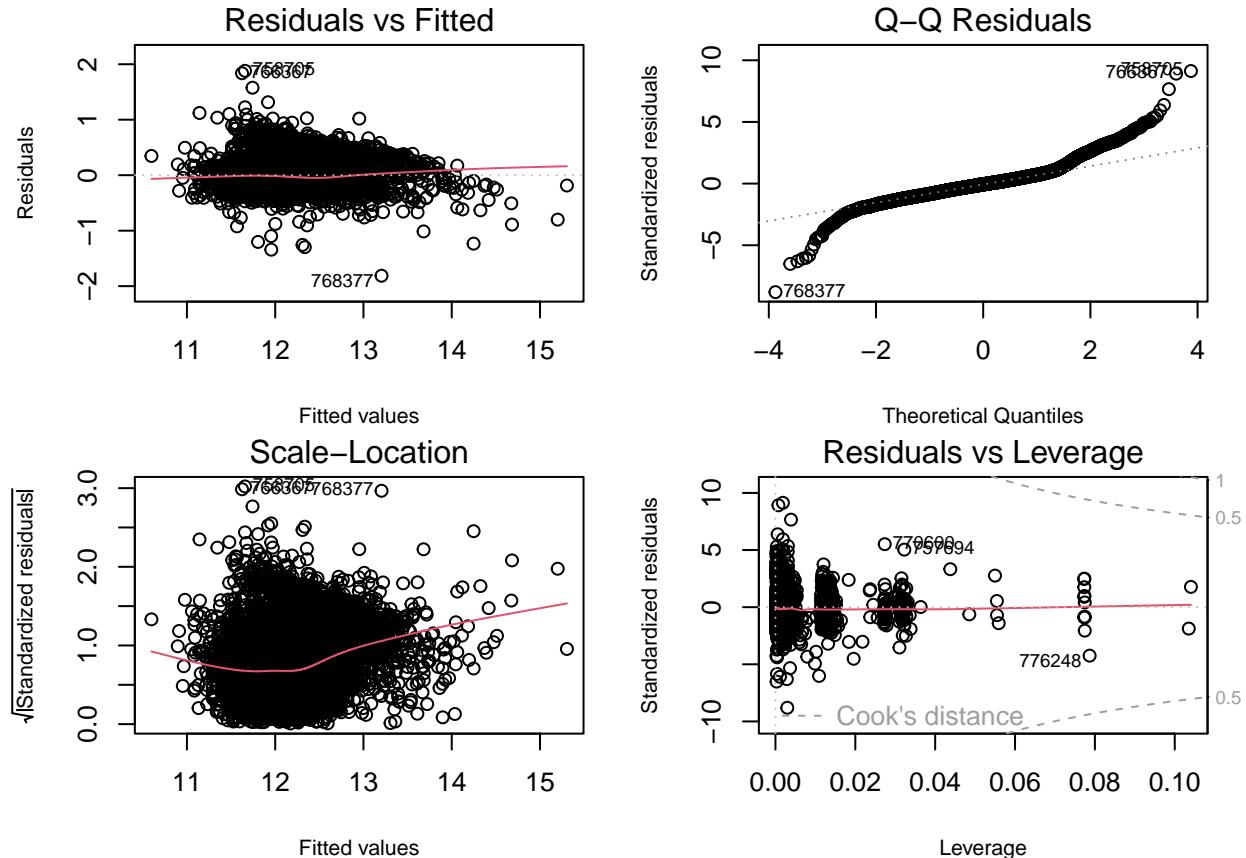
```

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 large 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. However, there is still room for improvement. The RSE, back transformed from the log transformed response variable, is 0.2062, which suggests that model's predictions on average deviate by about 22.9% in either direction.

Looking at the direction of the coefficients, we generally see trends that make sense, as more effective area increases values as does number of baths. However, the negative coefficient for the number of bedrooms is surprising. This might be due to collinearity with effective area or baths. Among the Condition Description categories, the relative order of the coefficient values are in line with their relationship to each other and of the reference level of Average. As we might expect, the houses in commercial and industrial zones are not as valuable as those in residential zones, likely due to their unusual placement which might lead to more

noise and pollution. Interestingly, mixed-use also has a negative coefficient although this is only significant at alpha=0.05. This may fall in-line with the notion that houses that are near non-residential developments could be actually worth less, an observation we made after looking at the plots for houses near stores.

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



Looking at these diagnostic plots, we see some concerns related to some linear regression assumptions. The residuals vs fitted plot shows a slightly non-linear pattern at the higher values. This suggests there may be a slightly non-linear relationship not captured by the model. We also see perhaps less variance as values get higher, indicating also some level of heteroscedasticity that is reflected in the Scale-Location plot. Meanwhile, the Q-Q plot shows that residuals may not be normally distributed at the tails. Note that no points appear to pass Cook's distance lines. Overall, while these plots indicate some concerns over non-normality and non-linearity, the violations to linear regression assumptions seem fairly minor given this real world application.

```
## Model with Food-Related Features
```

```
# 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_Specialty/Farmers` +
  `dist_Discount Store` + `dist_Drug Store` + `dist_Grocery Store` +
  `dist_Supermarket/Store` + total_1_mile + total_2_miles +
  total_10_miles, data = sfh_final)
```

```
# Check model summary
```

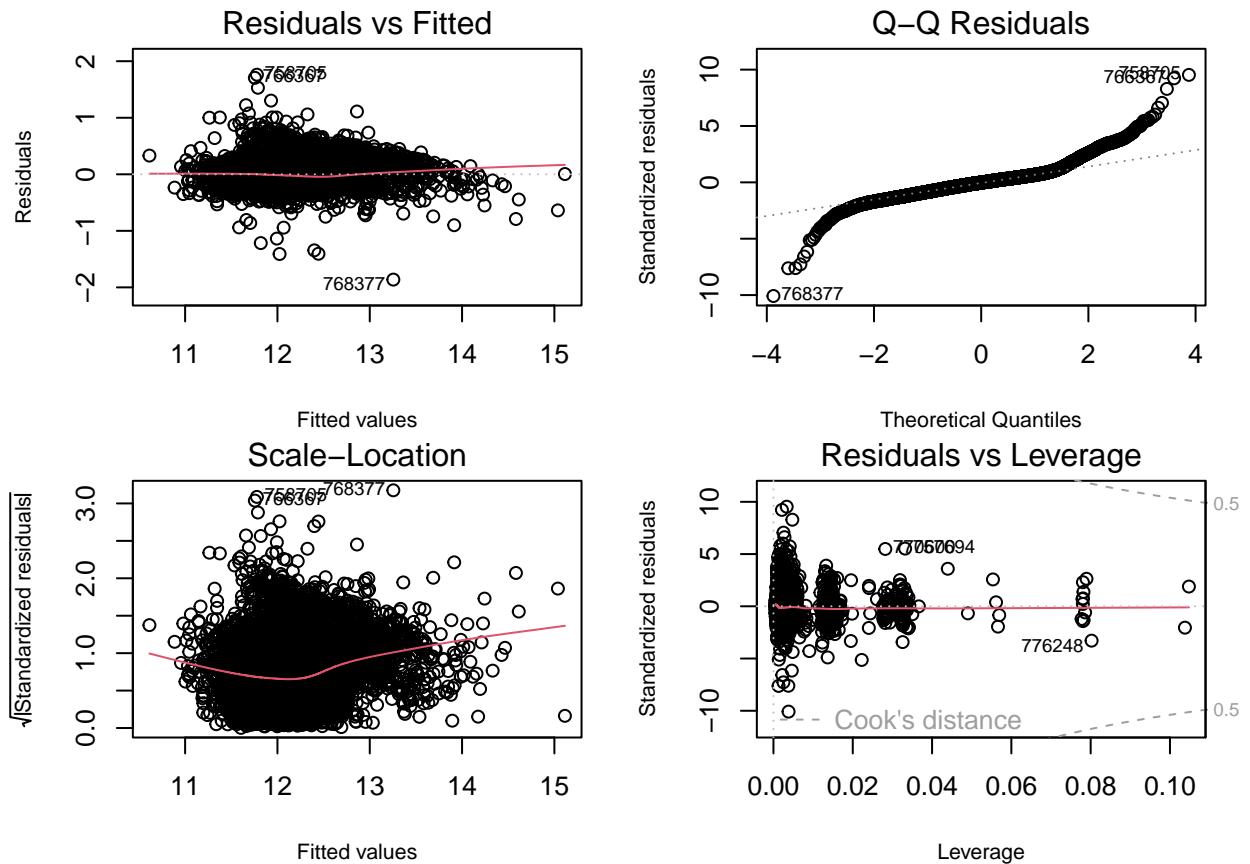
```

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_Specialty/Farmers` +
##     `dist_Discount Store` + `dist_Drug Store` + `dist_Grocery Store` +
##     `dist_Supermarket/Store` + total_1_mile + total_2_miles +
##     total_10_miles, data = sfh_final)
##
## Residuals:
##      Min        1Q     Median        3Q       Max
## -1.86269 -0.10204 -0.00357  0.08017  1.76154
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.046e+01  5.745e-02 182.034 < 2e-16 ***
## Sqrt_Effective_Area       3.403e-02  3.591e-04  94.774 < 2e-16 ***
## Number_of_Bedroom          -2.430e-03 2.679e-03 -0.907  0.3644
## Number_of_Baths_Total      5.772e-02  3.589e-03 16.082 < 2e-16 ***
## Condition_DescriptionExcellent 2.366e-01  2.364e-02 10.012 < 2e-16 ***
## Condition_DescriptionFair   -1.259e-01 1.244e-02 -10.118 < 2e-16 ***
## Condition_DescriptionGood   5.014e-02  5.783e-03  8.671 < 2e-16 ***
## Condition_DescriptionPoor   -2.150e-01 3.380e-02 -6.360 2.12e-10 ***
## Condition_DescriptionVery Good 1.196e-01  1.293e-02  9.252 < 2e-16 ***
## Condition_DescriptionVery Poor -4.187e-01 5.566e-02 -7.523 5.87e-14 ***
## Zone_CategoryCommercial    -1.052e-02 1.993e-02 -0.528  0.5975
## Zone_CategoryIndustrial    -2.509e-01 3.313e-02 -7.572 4.04e-14 ***
## Zone_CategoryMixed-Use      2.209e-03  3.278e-02  0.067  0.9463
## Zone_CategoryPlanned       -7.533e-03 2.262e-02 -0.333  0.7391
## Effective_Age              -6.019e-03 4.392e-04 -13.704 < 2e-16 ***
## `dist_Convenience Store`    7.164e-05 5.459e-06 13.124 < 2e-16 ***
## `dist_Specialty/Farmers`   -3.682e-05 3.202e-06 -11.499 < 2e-16 ***
## `dist_Discount Store`      -7.110e-05 4.628e-06 -15.361 < 2e-16 ***
## `dist_Drug Store`          6.484e-05 5.151e-06 12.588 < 2e-16 ***
## `dist_Grocery Store`       4.433e-05 3.890e-06 11.394 < 2e-16 ***
## `dist_Supermarket/Store`   -1.053e-05 5.615e-06 -1.876  0.0607 .
## total_1_mile               -7.329e-03 4.882e-04 -15.013 < 2e-16 ***
## total_2_miles              3.029e-03 2.245e-04 13.493 < 2e-16 ***
## total_10_miles             4.174e-04 1.654e-04  2.524  0.0116 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1851 on 9391 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8396, Adjusted R-squared:  0.8392
## F-statistic: 2137 on 23 and 9391 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 Results

With the exception of distance to supermarket/superstores, all food related predictors are statistically significant at alpha= 0.05. The p-values of all these other predictors are all less than 2e-16 except for the total number of stores within 10 miles, which was .0104. This suggests that the values are significant, and compared to the base model, the adjusted R-squared increases from .8005 to 0.8396.

This provides an indication that store-related features add a modest level of explanatory power, and the direction of these coefficients generally align with the somewhat surprising observations we found in the marginal plots. Farther closest distances to convenience, drug, and grocery stores lead to increased predicted value while closer closest distance to specialty/farmers markets increased predicted value. Interestingly, however, a closer closest distance to a discount store improves predicted value which both differs from the marginal plot and from our initial intuition about where discount stores are placed. Literature suggests that discount stores are generally placed in depressed neighborhoods, where they may be one of few food options. However, perhaps it is the case that within those neighborhoods (after accounting for other measures of neighborhood affluence like area, baths, and condition), the houses nearby the one food option are a little more variable.

After adding these variables, we note that some of the zone categoricals are no longer statistically significant predictors. This further suggests that the store-related variables are capturing some of the variance related to the general attractiveness of a region/neighborhood, particularly as it relates to whether non-residential properties are nearby.

Notably, the coefficient directions for the 1 and 2 mile total stores point in opposite directions. However, this aligns with what we saw in the marginal plots and aligns with an understanding that folks want stores that are close but perhaps not *too* close. stores within a moderate distance

Overall, these details suggest that store-related variables do have some explanatory power, but the modest improvement and our revised explanations suggest that these features may not be as important as initially anticipated. The closest distance features and their direction signs point to how different types of foods have a different effect on property values, likely due to different behavioral patterns among the affluent.

## Model Comparison and Discussion

To confirm this analysis, however, we should conduct further tests to see if a model with these store-related variables is actually significantly better than the baseline model. To do this, we use an F-test.

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

## Analysis of Variance Table
##
## Model 1: Log_Assessed_Total ~ Sqrt_Effective_Area + Number_of_Bedroom +
##           Number_of_Baths_Total + Condition_Description + Zone_Category +
##           Effective_Age
## Model 2: Log_Assessed_Total ~ Sqrt_Effective_Area + Number_of_Bedroom +
##           Number_of_Baths_Total + Condition_Description + Zone_Category +
##           Effective_Age + `dist_Convenience Store` + `dist_Specialty/Farmers` +
##           `dist_Discount Store` + `dist_Drug Store` + `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    9391 321.77  9    77.888 252.58 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-test shows that the inclusion of store-related variables significantly improves the model ( $F = 306.03$ ,  $p < 2.2e-16$ ).

This lends further indication that our more advanced model does improve predictions and these features of store proximity and count are relevant, even though ‘better’ access to these stores does not necessarily lead to more valuable property like we expected. Our findings suggest that there is a ‘happy middle’ in terms of distance to food and higher property values, and that different types of stores have different effects on these property values.

## Limitations of the Model and Analysis

As mentioned earlier, the QQ-plot and other diagnostic plot suggest that some linear regression assumptions are not fully met, which affects the reliability of our influences. As the Scale-Location plot indicates heteroscedasticity, 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.

Additionally, we did not focus on investigating interaction effects between different predictors. While we believe these features may interact with notions such as neighborhood affluence, we did not see particular interaction effects that made sense to include given our focus on identifying if food-related features in general are important.

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.

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. 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. Yet, having more stores within a 2 mile proximity increased property values, suggesting that people may prefer stores that are close but “not too close.”

While store proximity and count improved the model’s performance, their impact remains context-dependent. Additionally, it would be worthwhile to expand this analysis to other regions to better understand how place affects this relationship between food access and property values.