Housing Project

9/22/21

Team Members: Tanvi Yende, Linda Mao, Dov Greenwood

Reading in Cleaned Data & Completing Formatting as Necessary

```
#reading in required libraries
library(ggplot2)
```

```
## Registered S3 methods overwritten by 'tibble':
##   method     from  
##   format.tbl pillar
##   print.tbl  pillar
```

```
library(scales)
library(gridExtra)
library(dplyr)
```

```
## 
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:gridExtra':
## 
##     combine
```

```
## The following objects are masked from 'package:stats':
## 
##     filter, lag
```

```
## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------- tidyve
rse 1.3.0 --
```

```
## v tibble  3.0.3     v purrr   0.3.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------ tidyverse_co
nflicts() --
## x readr::col_factor() masks scales::col_factor()
## x dplyr::combine()    masks gridExtra::combine()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
```

```
options(warn=-1)
```

```
#reading in data
dat <- read.csv("data/finalData.csv")
head(dat)
```

	pid	address	value	buildings	year	land	living	go…	style	▸
	<int>	<fctr>	<int>	<int>	<int>	<dbl>	<int>	<int>	<fctr>	
1	3	199 SOUTH END RD	145670	1	1950	0.25	1475	73	Cape Cod	
2	4	11 URIAH ST	150710	1	1989	0.18	1792	82	Colonial	
3	5	181 SOUTH END RD	115010	1	1945	0.36	864	73	Ranch	
4	6	169 SOUTH END RD	105280	1	1940	0.32	1040	63	Bungalow	
5	7	173 SOUTH END RD	129290	1	1986	0.35	1512	77	Cape Cod	
6	8	165 SOUTH END RD	95060	1	1940	0.17	1080	63	Cape Cod	

6 rows | 1-10 of 18 columns

```
#basic data cleaning
dat <- dat[dat$beds != "0",]
dat <- unique(dat)

#coercing ac variable to be a binary instead of yes/no format
dat$ac <- ifelse(dat$ac == 'yes', TRUE, FALSE)
```

Explanation of Data Reconciliation

Data Reconciliation: There were multiple differences in our data resulting from our different approaches to wrangling the raw HTML data, which required reconciliation. Primarily, the differences resulted from different grep searches; these included comprehensiveness (e.g. whether we searched for the word “Ttl” as well as “Total”) and use of different categories for the same category (namely, “land model” versus “building model”). A more significant difference came in whether or not to include half bathrooms in the bathroom count or not, especially since two half bathrooms would some to one full bathroom in the raw sum, even though in reality two half bathrooms do not equate to one whole bathroom. We ultimately decided to include half bathrooms in the sum as we would lose significant data from excluding them.

Data Cleaning: After the data was reconciled, some cleaning was necessary. Some of the entries had extra whitespace before and after the actual data, which we removed using the trimws() function. In the specific case of the heating fuel column, identical options were sometimes labeled differently—namely, “Gas/Oil” and “Oil/Gas”—which we standardized by assigning both such values to “Gas/Oil.” Then, we examined each variable and if there were any glaring outliers. In the case of the “Percent Good” variable, only two properties exceeded 100% Good. However, upon examining the properties, we found that they were unusually nice (high value, many bedrooms, etc.), which warranted the very high value.

For style, we removed “Apt House,” which appeared to have been erroneously labeled with the model “Single Family.” In this project, we are only looking at true single family homes, and this apartment house would have skewed bedroom/bathroom numbers–thus, we deleted “Apt House” styles from the dataset. Other potentially disruptive styles, like “Inn,” had normal bedroom/bathroom values, so we decided to keep them.

Finally, we went through the categorical variables and then factorized them to be able to be used in lm(). Through this, we factorized style (“Colonial” as the reference level), beds, and neighborhood (“0101” as reference level because most homes were in 0101). For AC, we realized that it would be more informative to draw the line between having AC or not having AC. Thus, we changed the AC variable accordingly and factorized with “no” as reference level.

Exploratory Analysis

Descriptions

```
#creating decade function for the sake of exploratory analysis
floor_decade    = function(value){ return(value - value %% 10) }
```

```
#Visualizing the type and size of data
str(dat)
```

```
## 'data.frame':    9134 obs. of  17 variables:
##  $ pid      : int  3 4 5 6 7 8 9 10 23 24 ...
##  $ address  : Factor w/ 9135 levels "1 BEACON AV",..: 2919 429 2573 2215 2356 2133 2040 1889 
6511 395 ...
##  $ value    : int  145670 150710 115010 105280 129290 95060 95970 93520 123340 182280 ...
##  $ buildings: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ year     : int  1950 1989 1945 1940 1986 1940 1930 1900 1940 1940 ...
##  $ land     : num  0.25 0.18 0.36 0.32 0.35 0.17 0.14 0.2 0.16 0.13 ...
##  $ living   : int  1475 1792 864 1040 1512 1080 1040 985 1484 2124 ...
##  $ good     : int  73 82 73 63 77 63 63 63 68 78 ...
##  $ style    : Factor w/ 13 levels "2 Family","Bungalow",..: 3 4 11 2 3 3 3 11 3 4 ...
##  $ model    : Factor w/ 1 level "Single Family": 1 1 1 1 1 1 1 1 1 1 ...
##  $ grade    : Factor w/ 16 levels "Above Average",..: 1 4 4 4 4 4 1 4 4 3 ...
##  $ beds     : int  3 3 3 3 5 3 2 1 2 3 ...
##  $ baths    : num  2.5 2.5 1 1 2 1.5 1 1 1.5 2 ...
##  $ heatfuel : Factor w/ 11 levels "Electr Basebrd",..: 7 7 2 11 2 2 2 2 5 3 ...
##  $ heattype : Factor w/ 11 levels "Electr Basebrd",..: 7 7 2 11 2 2 2 2 5 3 ...
##  $ ac       : logi  FALSE TRUE FALSE FALSE TRUE FALSE ...
##  $ nghd     : Factor w/ 34 levels "1000","101","1100",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
#Finding descriptive information on value, our variable of interest
summary(dat$value)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   27300   92890  119000  149882  160930 3037930       1 
```

Explanation of Data Exploration & Analysis

Data Transformation: For land (in acres), there was an outlier at 16.91, when the median was 0.17. However, we did not remove it, as we performed a log transformation on this variable for our model, which brought the value significantly closer to the rest of the data. We also put log transformations on the variables living, land, and value because they were skewed.

Next, we eliminated variables to use in the model. We removed the “buildings” and “model” variables, as all of the values were 1 (because these were the only ones selected for building the model); and the “heatfuel” variable, as 9140 out of the 9200 data point were “Gas/Oil” (so the data seemed relatively insubstantial).

We also checked for collinearity with good vs. grade, beds vs. baths, beds vs. living space, and baths vs. living space. For all of these comparisons except good vs. grade, the adjusted R-squareds were relatively low < 0.5, so we opted to keep both. For good vs. grade, the adjusted R-squared was high and significantly correlated, so we determined that we should try to keep good over grade because it was numeric.

How Strongly/Weakly Associated Are Certain Variables with Assessed Value:

Living, beds, baths, percent good, grade, land are all positively associated and appear to be strongly associated with assessed value. Year built appears to be weakly associated with assessed value, and it is unclear if it is positively or negatively associated.

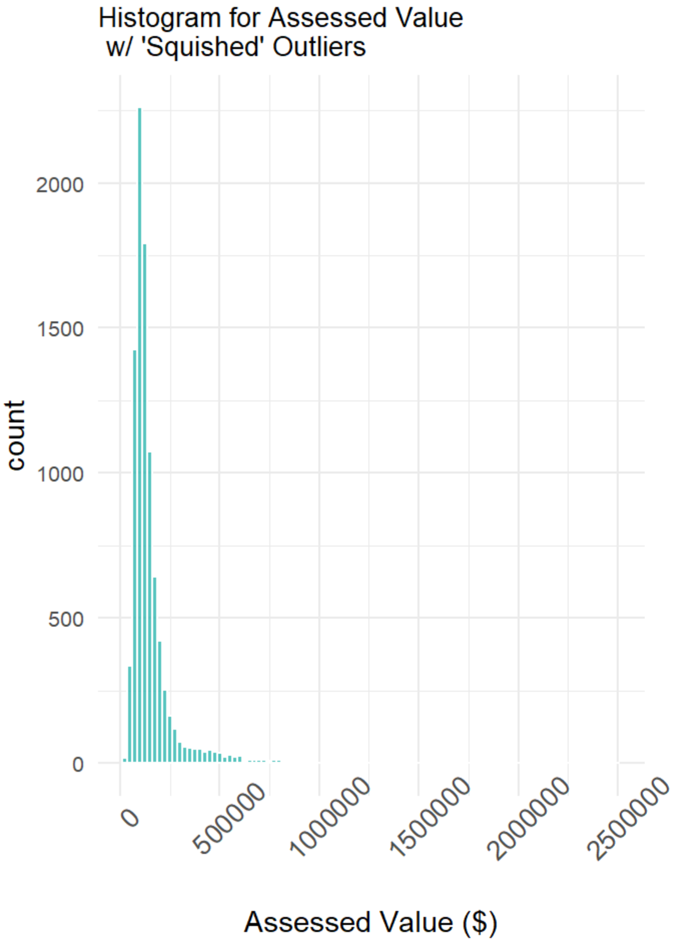
Possible misleading results (because we are considering only one variable at a time): It was difficult to interpret the relationship between style and log value, and this can be misleading (in our final model, style was an important independent variable). For heat fuel, homes overwhelmingly fell into the Gas/Oil type, so if we were to look at this variable just by itself, we would miss a lot of how this variable actually affects assessed value.

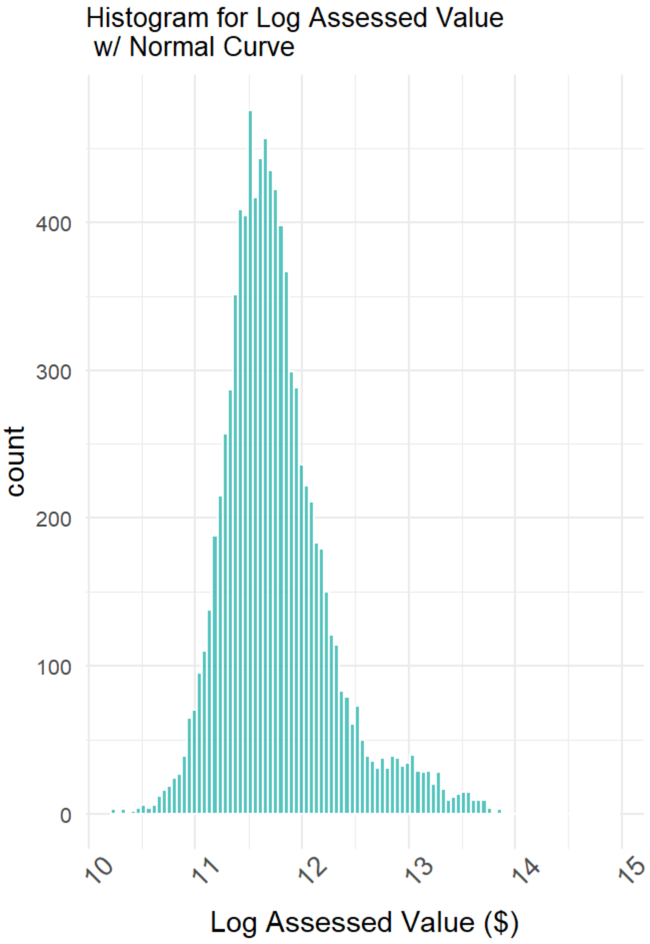
Benefits & Drawbacks of Defining Beds and Baths as Numeric Data:

The benefit of defining beds and baths as numeric data is interpretability–people understand the number of beds and baths intuitively on a number scale. The drawbacks of defining them as numeric data is that the data doesn't vary linearly with the dependent variable, log value, and it was very discrete data (in our model, we actually chose to define bed as a categorical variable).

Explanatory Plots

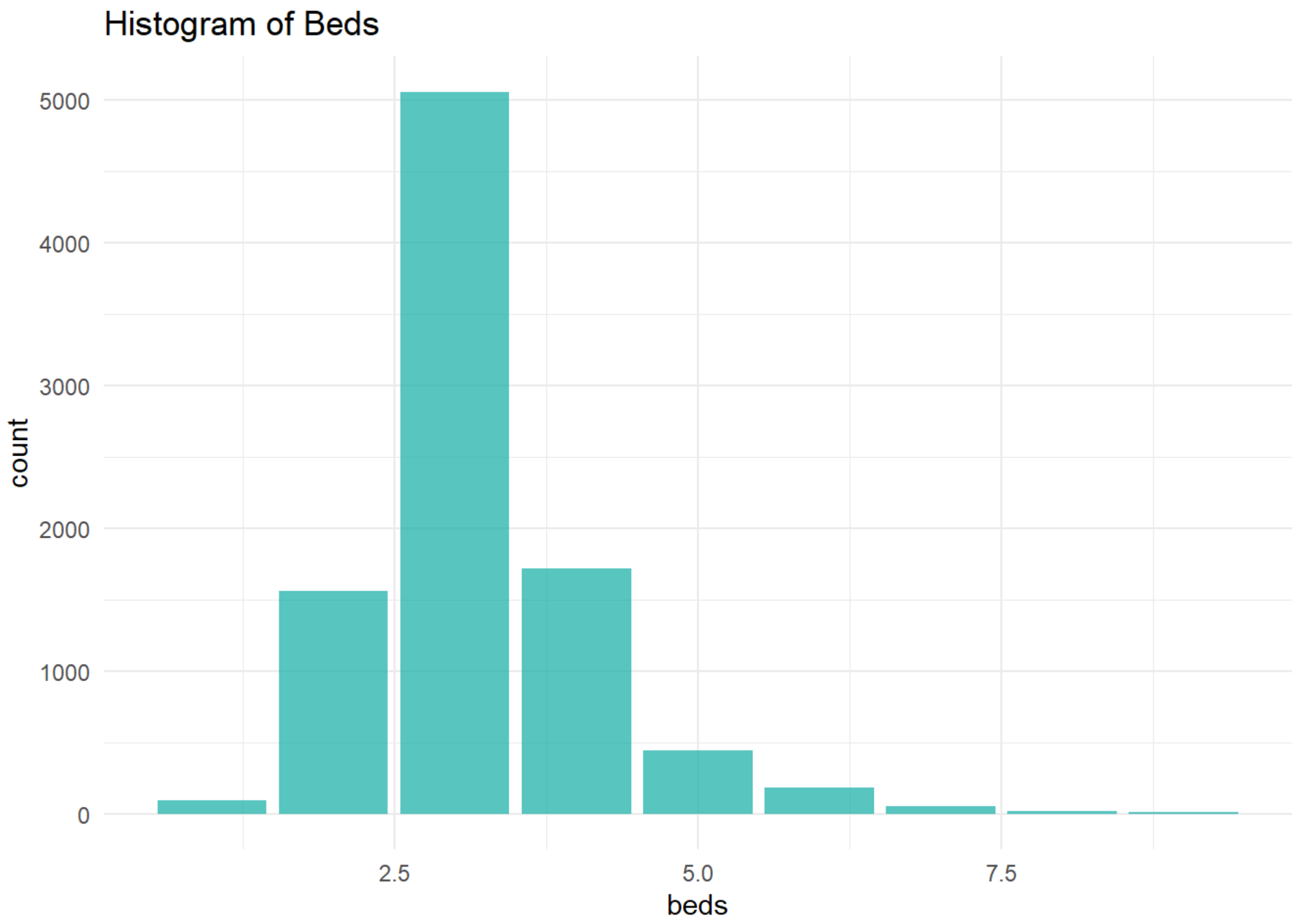
Visualizing Assessed Value





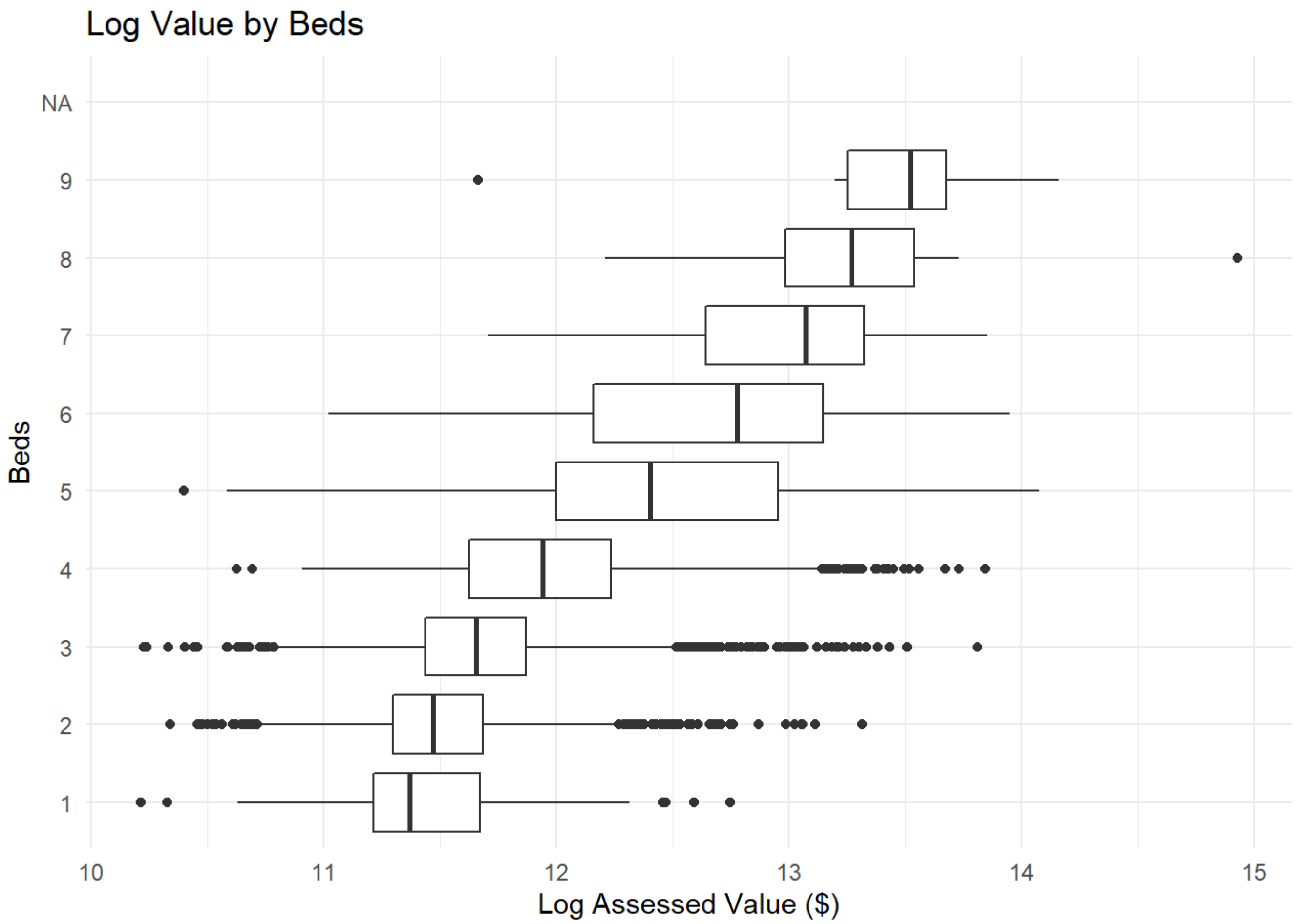
Assessed value is right-skewed and exponential, which will be an issue for a linear model. The log transformation shows that this transformation appropriately creates a normal distribution.

Visualizing Number of Beds



The distribution of beds appears to be right-skewed.

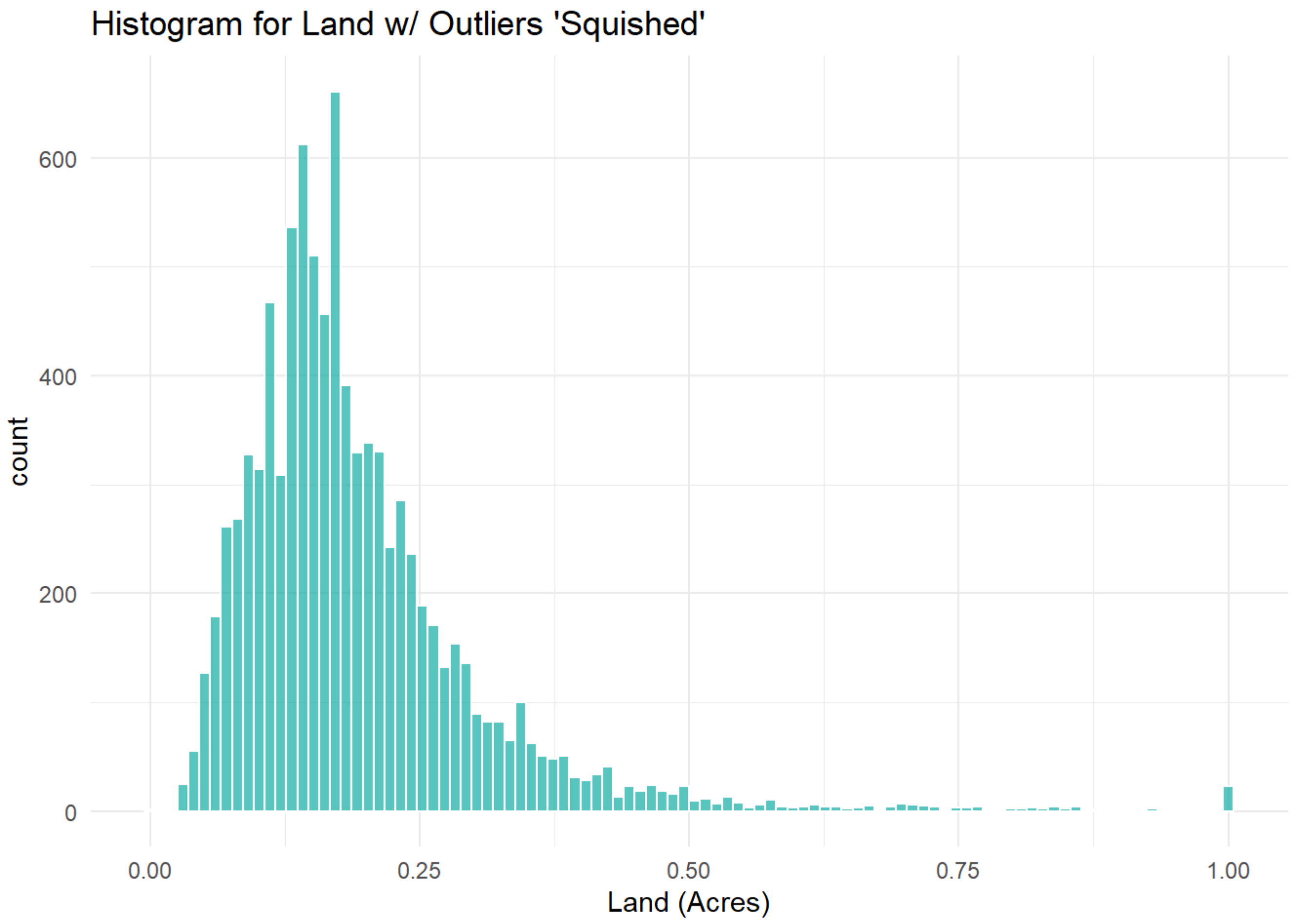
```
p16
```



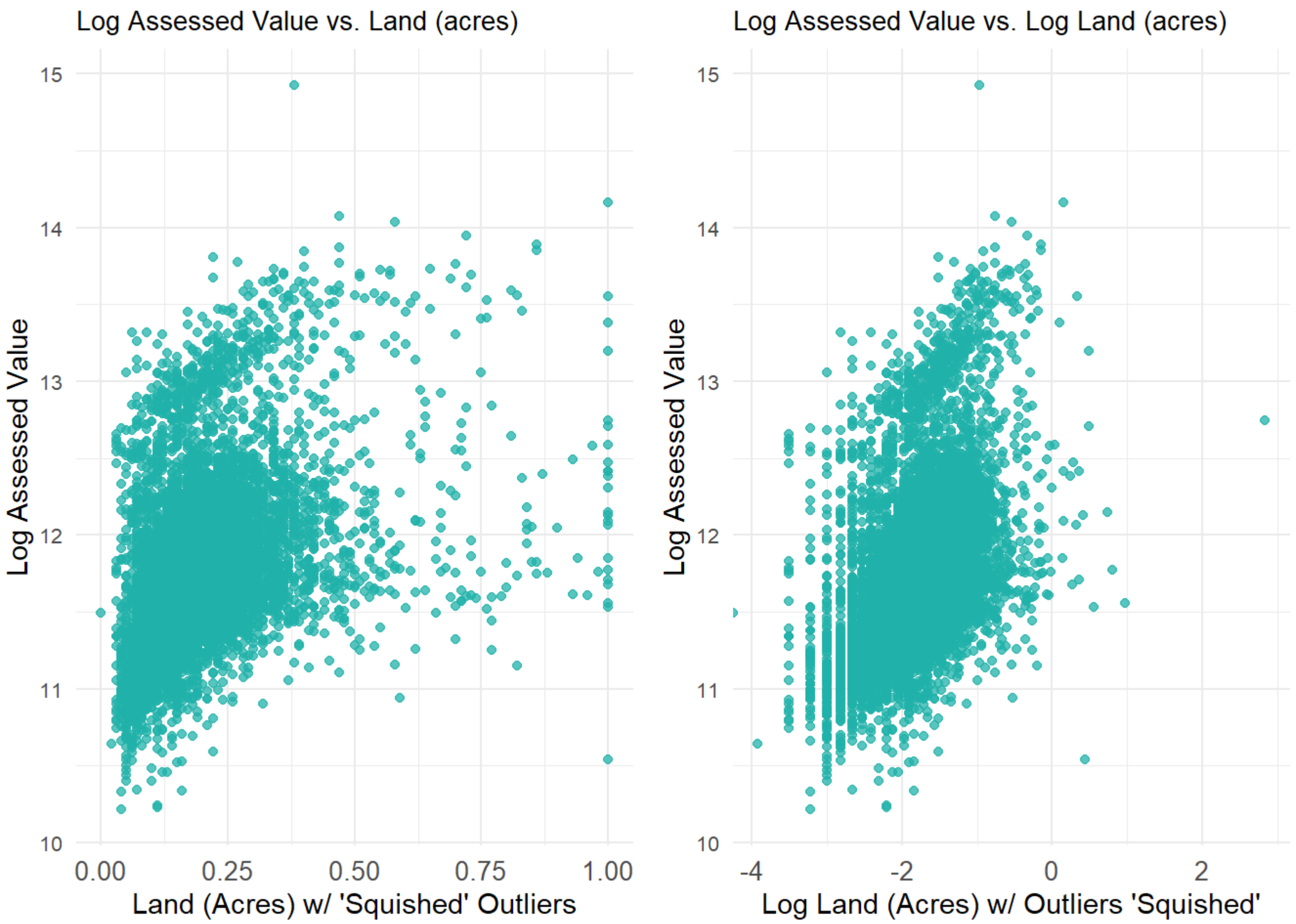
Because the relationship between log value and log beds still does not vary linearly, we will treat beds as a categorical variable.

Visualizing Land Variable

```
p5
```

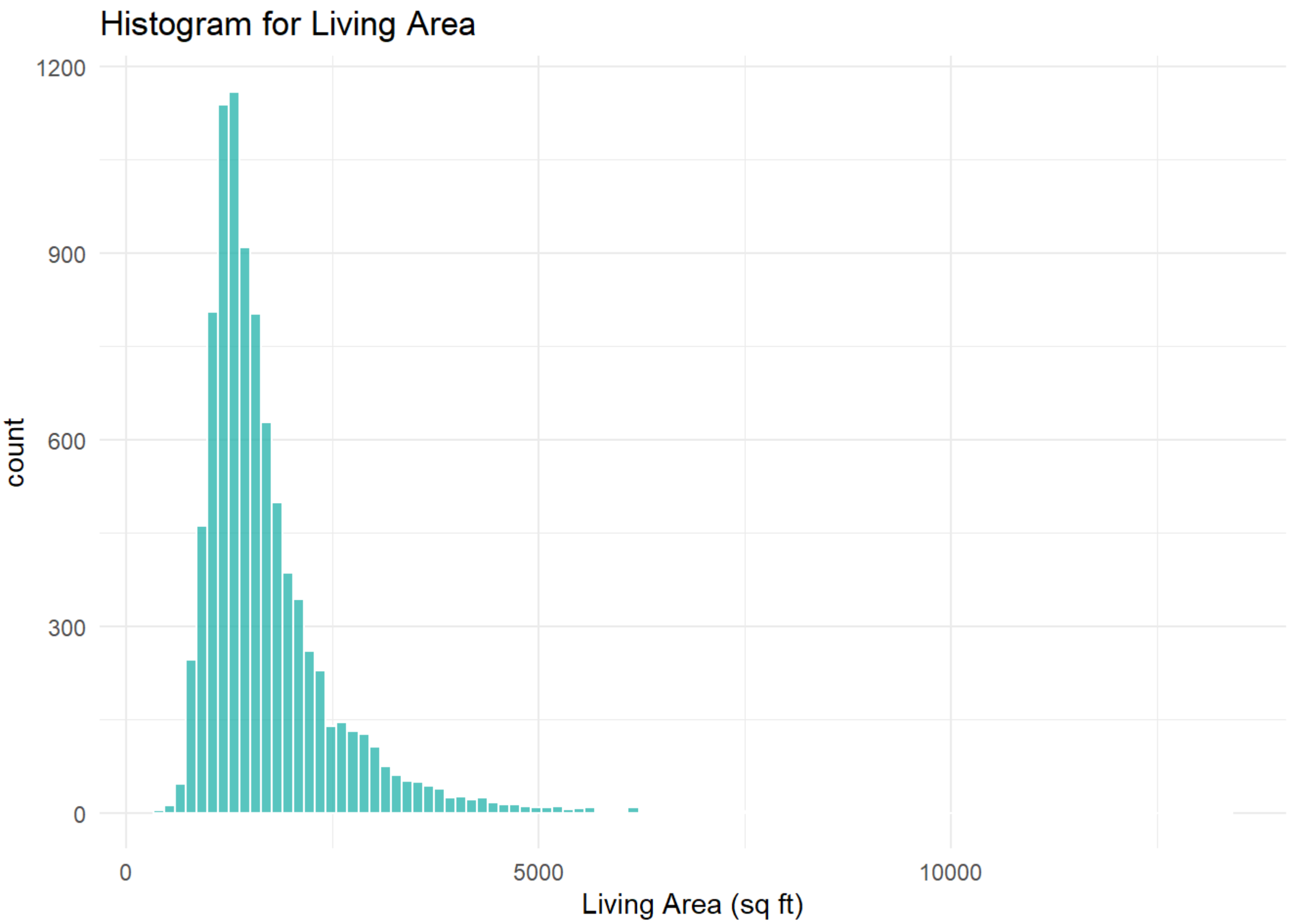


The histogram shows that the land variable appears to be right-skewed.

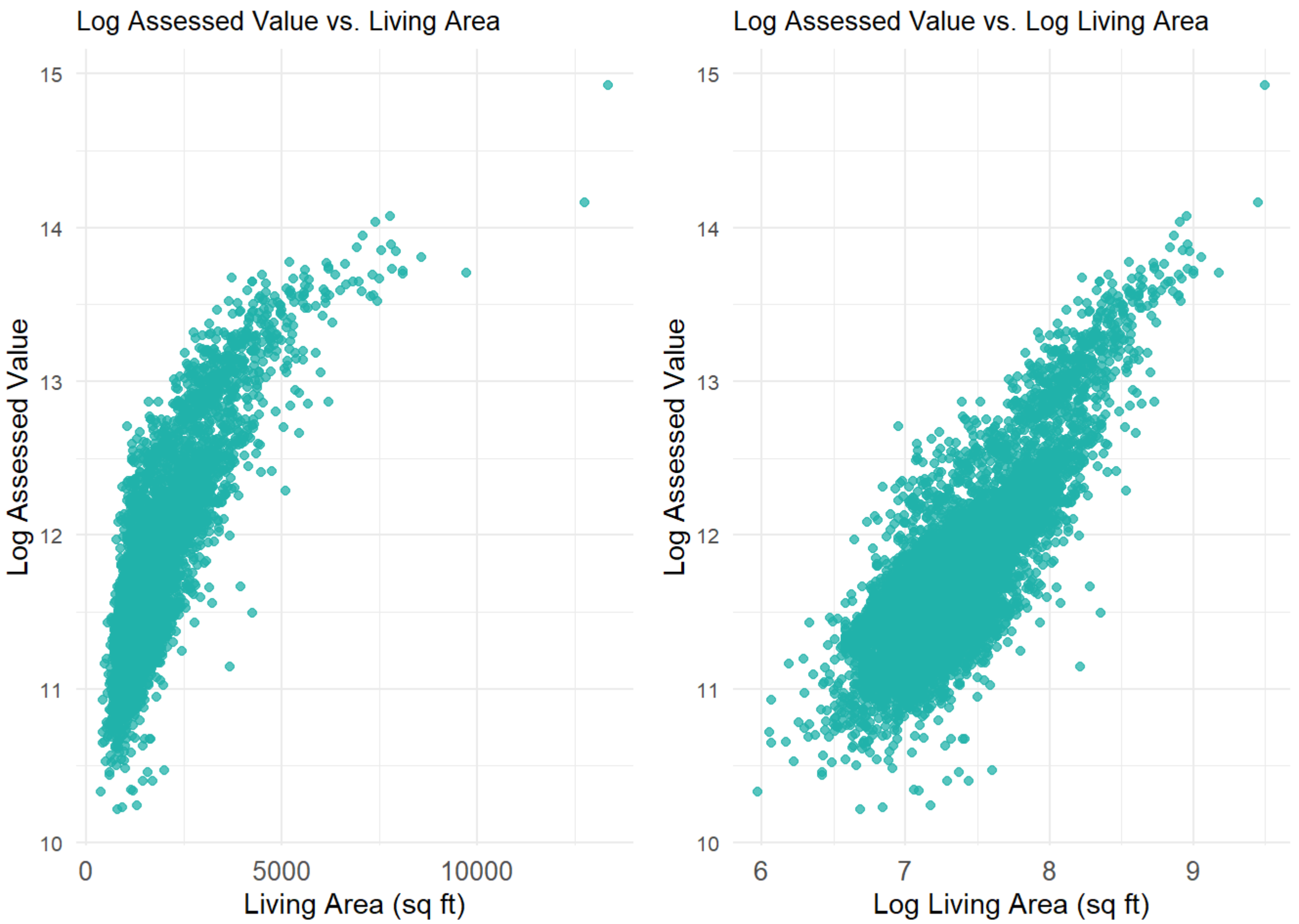


The relationship between log value and land appears exponential, but log value and log land vary more linearly. In our model, we will use log land for this reason.

Visualizing Living Area



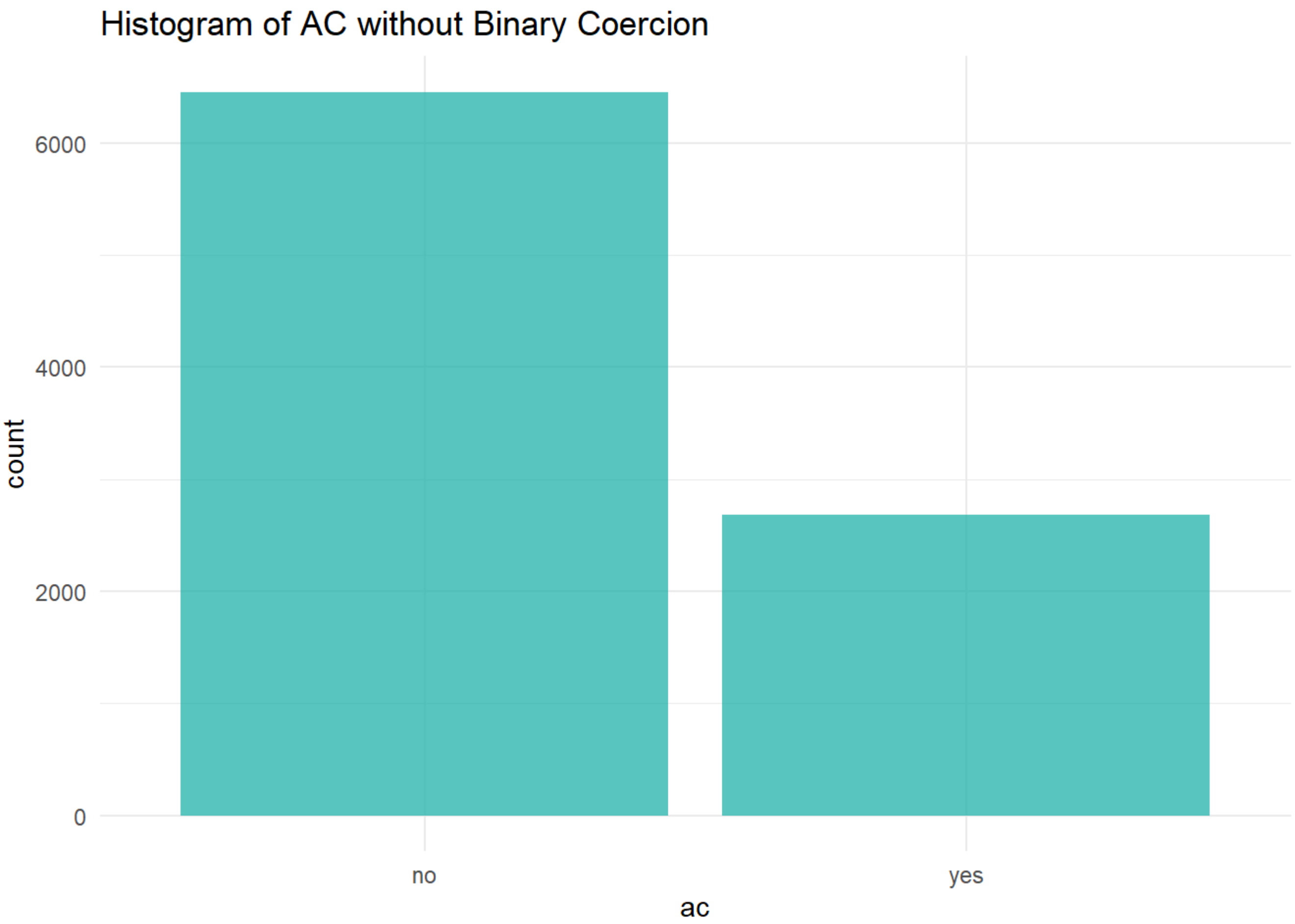
The histogram for living area shows that the variable is right-skewed.



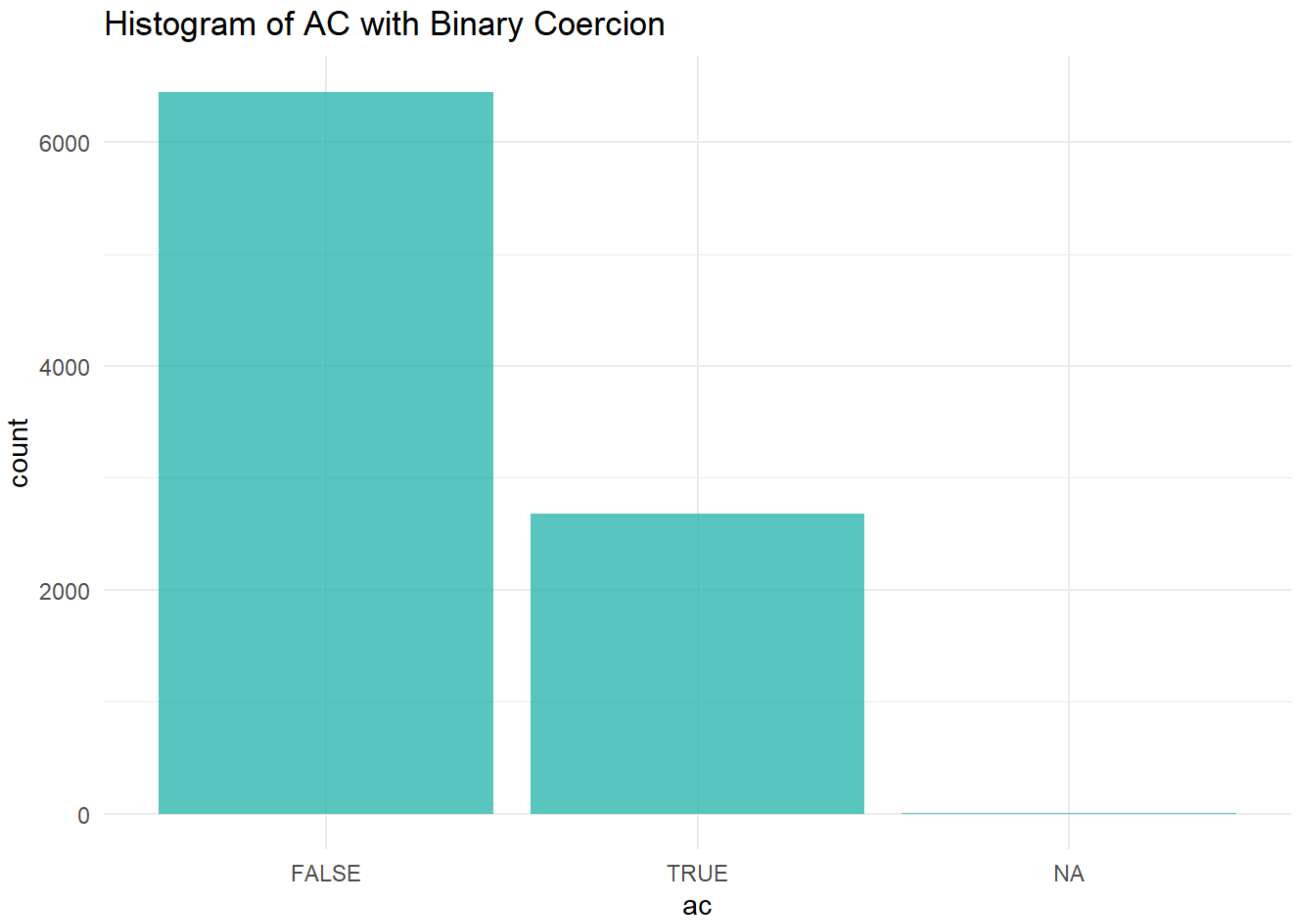
These plots show that while living area varies exponentially with log value, while log value and log land vary linearly.

Visualizing AC Type

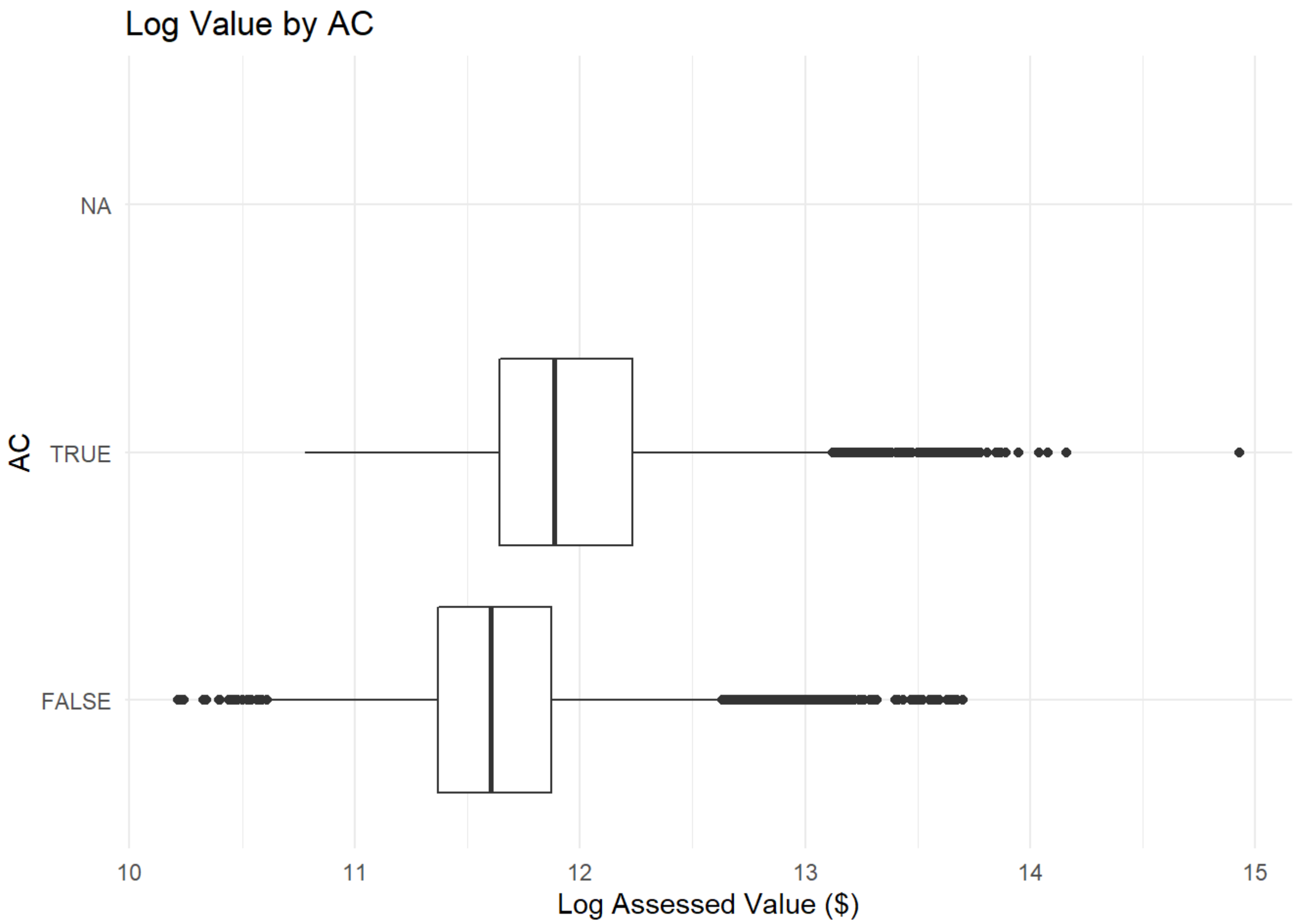
Note that we coerced ac to be a binary of either with or without ac



This graph shows the categories before we created a binary variable. &nbsp & None were considered to be indicators of no AC, while other categories were considered to be AC.

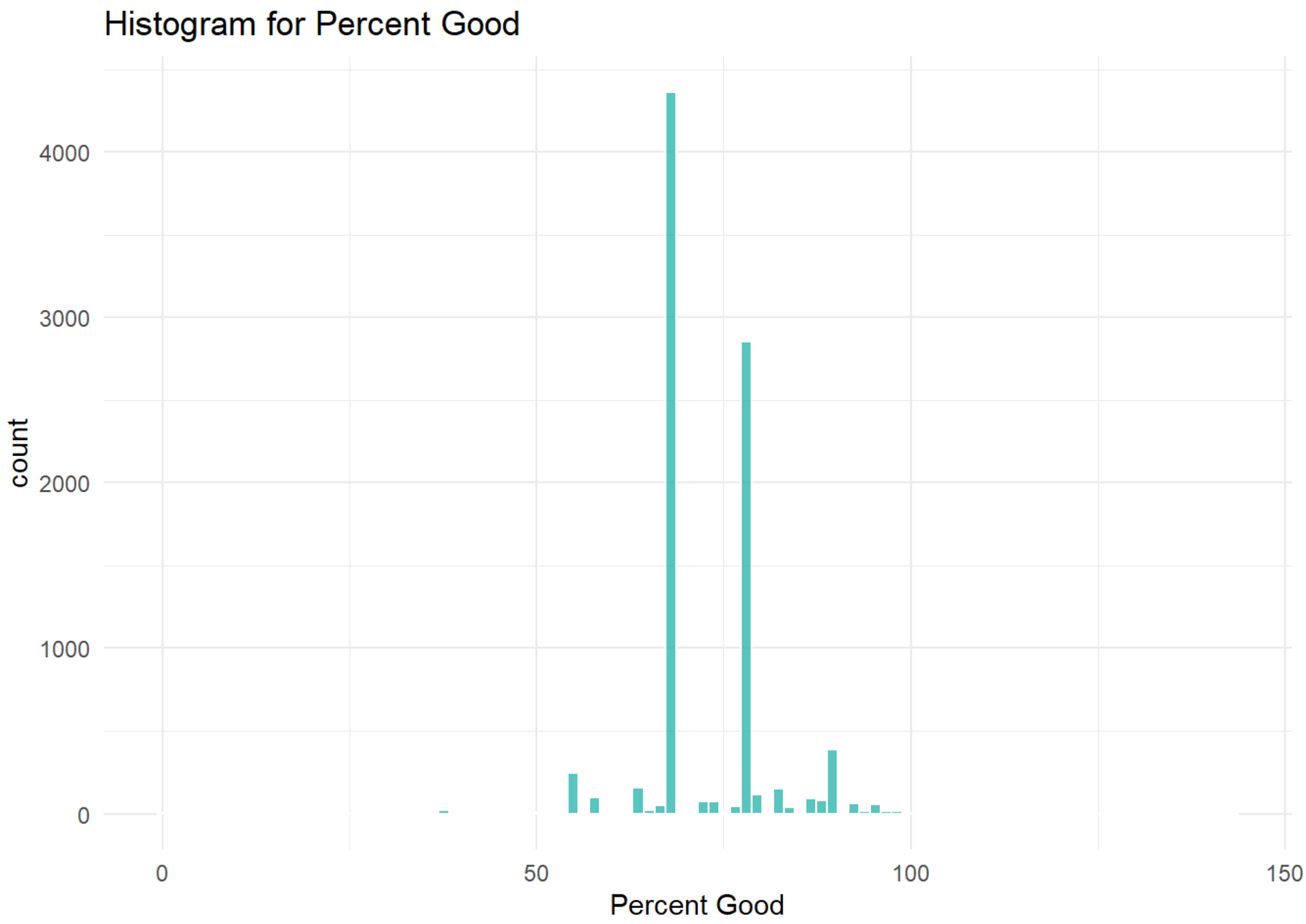


The distribution shows more houses don’t have ac than do.

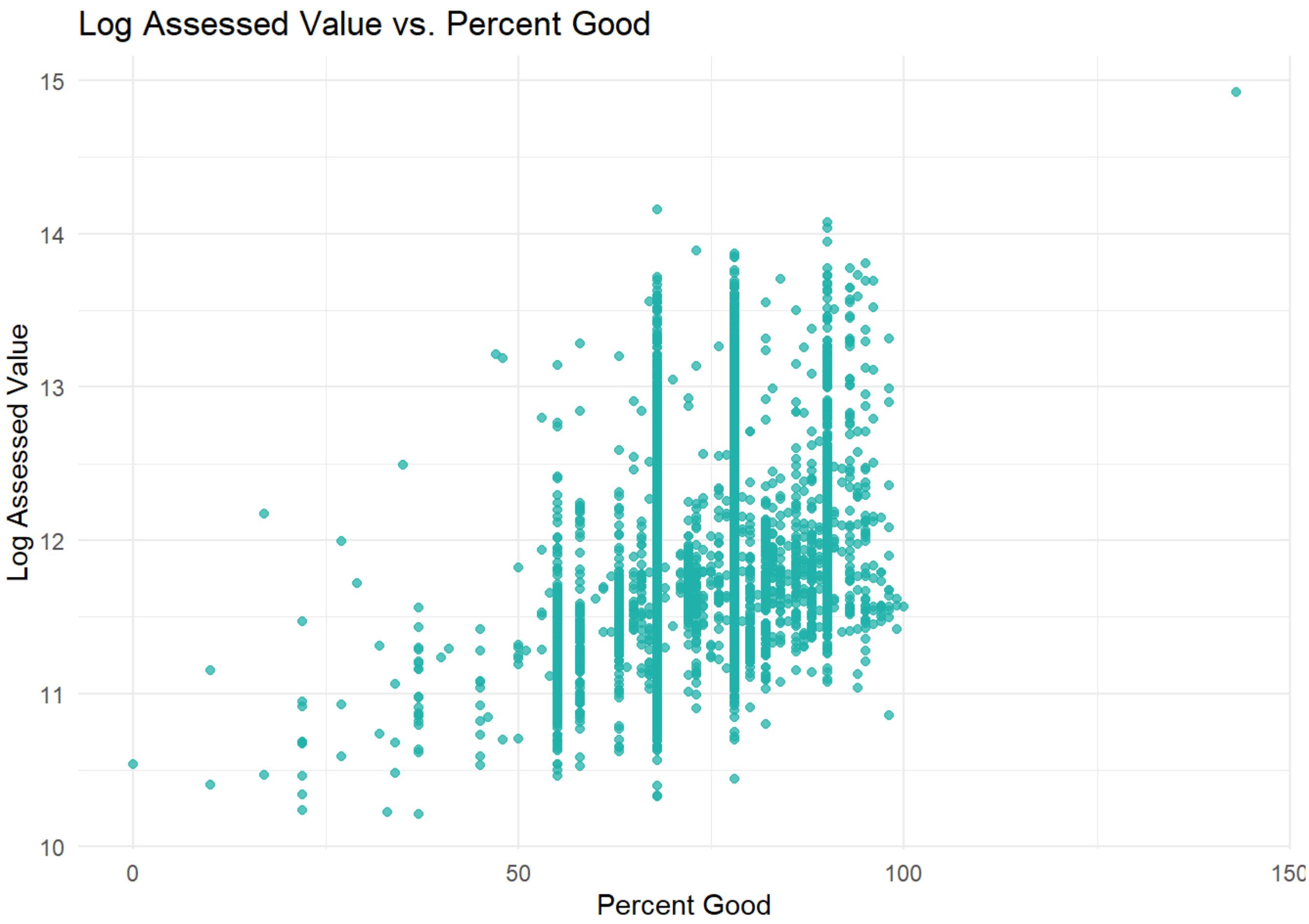


Homes with AC may have a higher value than those without, but this relationship is weak from this visualization.

Visualizing Percent Good

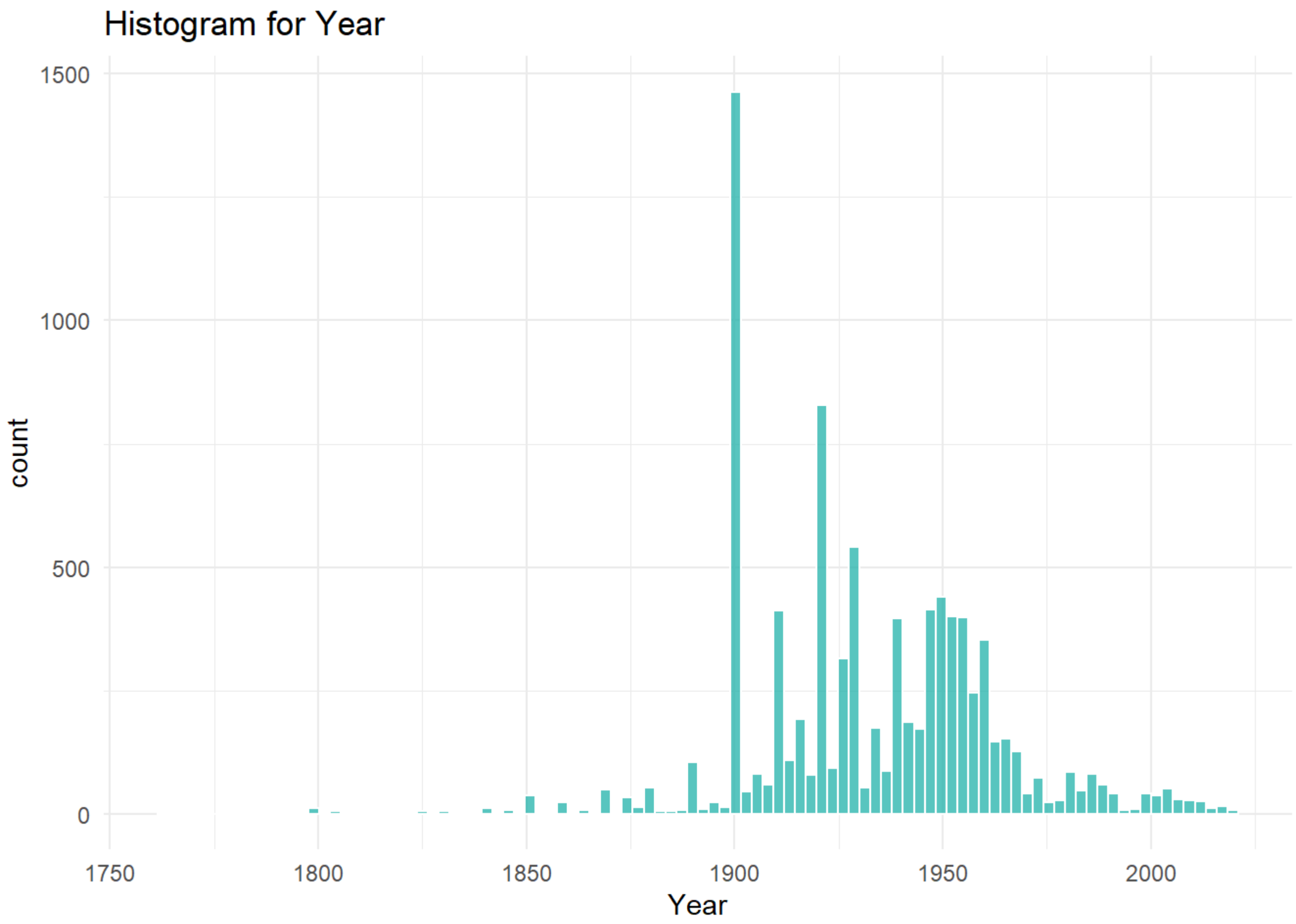


The distribution is not clear from the histogram.

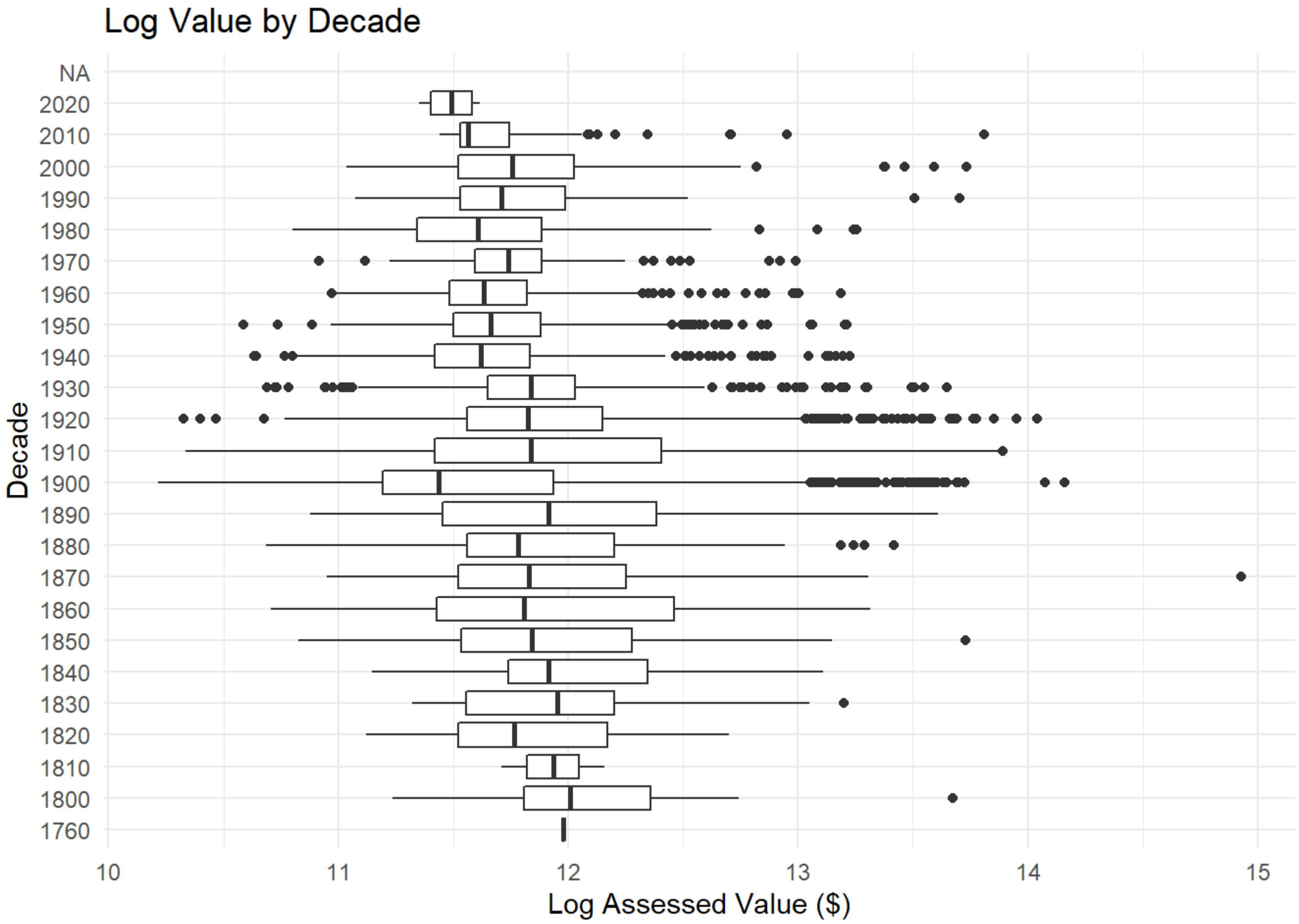


Percent good seems to vary linearly with log value.

Visualizing Year Variable

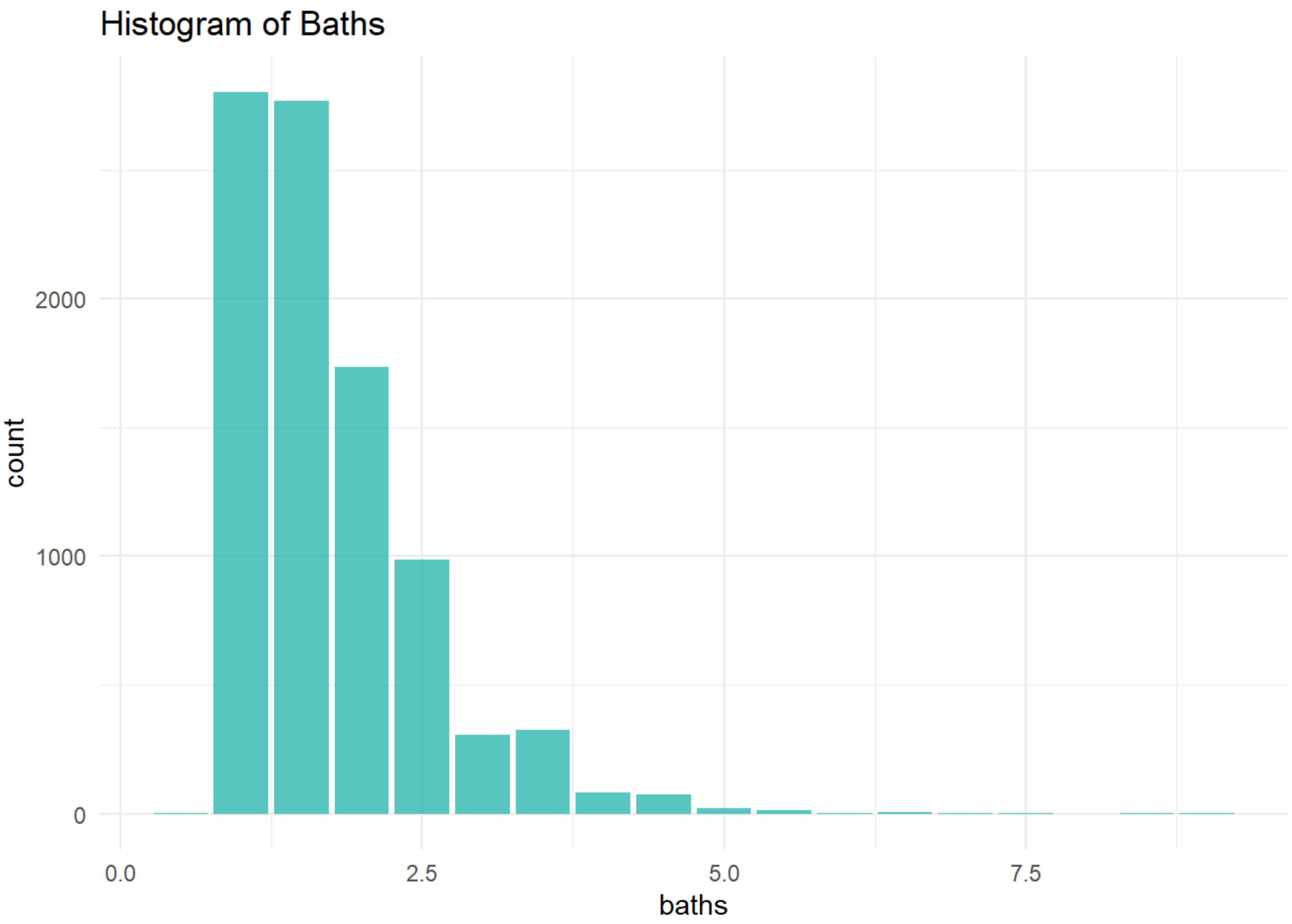


There seems to be many houses built in 1900, possibly indicating a data error or a large volume of houses built in 1900. We couldn’t resolve this question via historical research. There also seems to be a cluster of houses built around 1950.



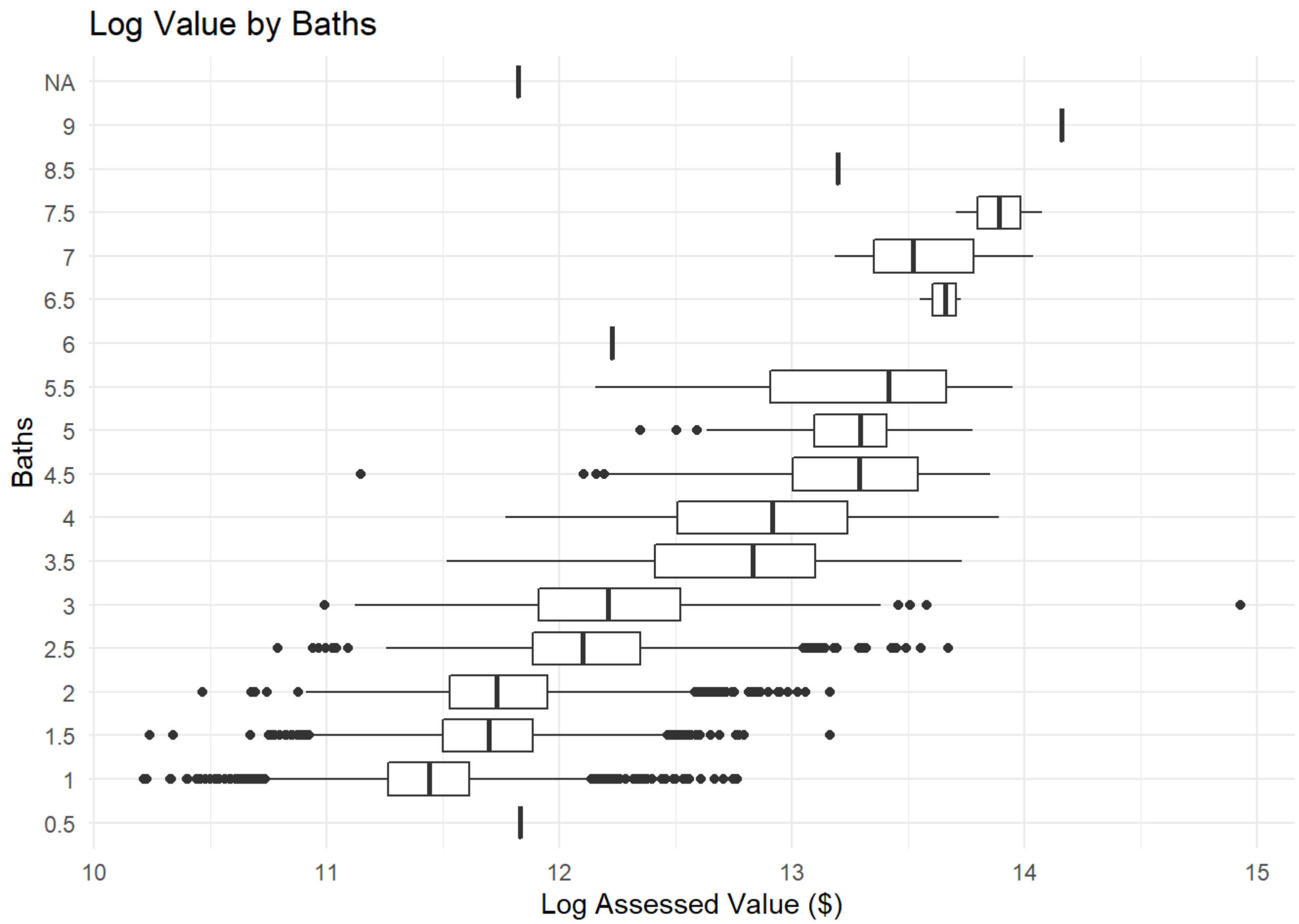
There doesn’t seem to be a clear relationship between year the house was built in and its assessed value.

Visualizing Baths



The distribution of baths appears to be right-skewed.

```
p18
```

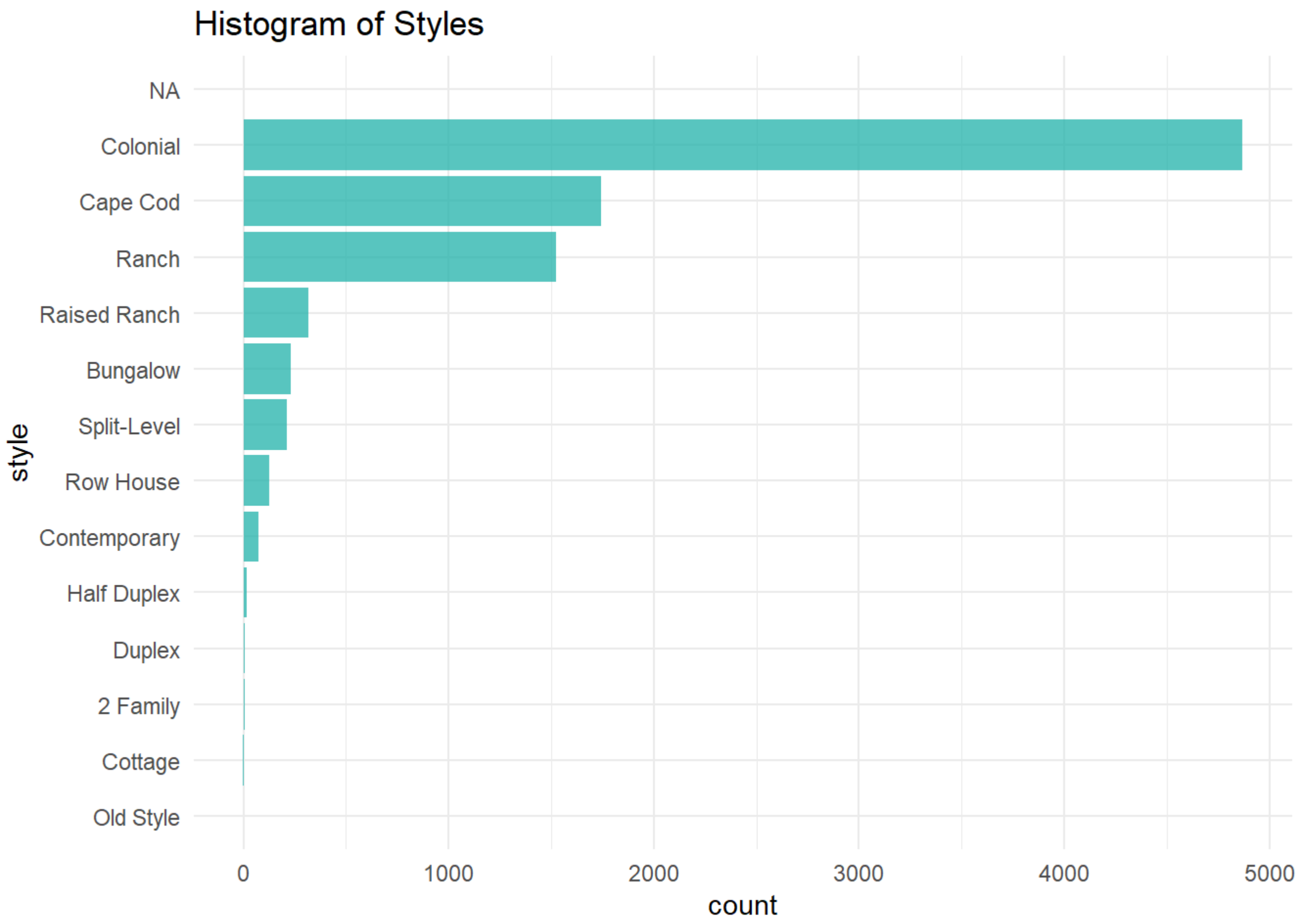


There appears to be a somewhat linear relationship between baths and log value. Upon testing in the model there was not a significant difference between treating baths numerically or categorically, so we kept it numeric to allow for half baths.

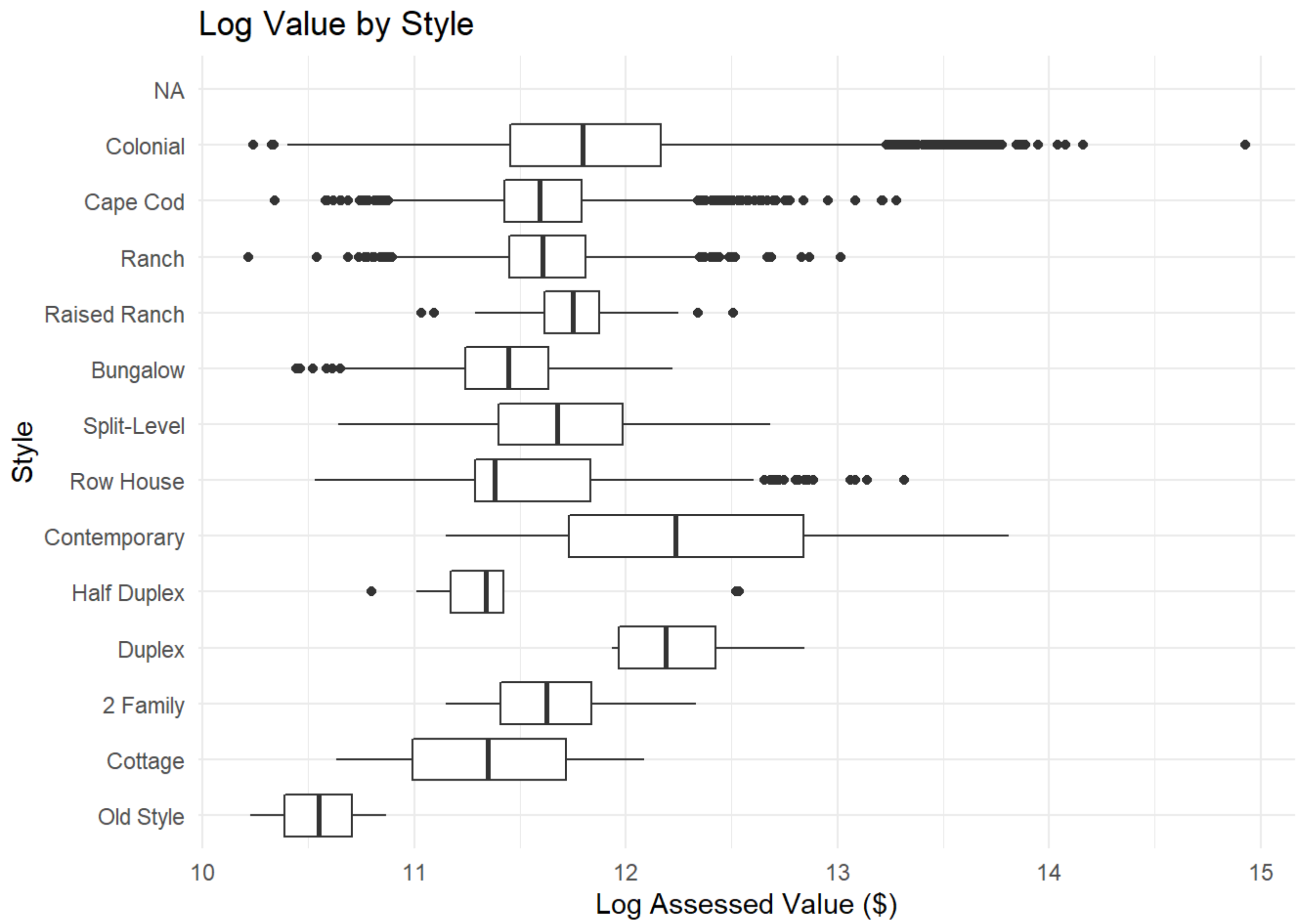
Visualizing Style

```
dat$style <- reorder(as.factor(dat$style),as.factor(dat$style), FUN = length) #reordering levels
so that they appear in order of frequency
p11 <- ggplot(data = dat, aes(style)) +
  geom_bar(fill="#20B2AA", alpha=0.75) +
  theme_minimal() +
  coord_flip() +
  labs(title = "Histogram of Styles")


p12 <- ggplot(data = dat, aes(x=style,y=log(value))) +
  geom_boxplot() +
  coord_flip() +
  labs(x="Style", title = "Log Value by Style", y="Log Assessed Value ($)")+
  theme_minimal()
```



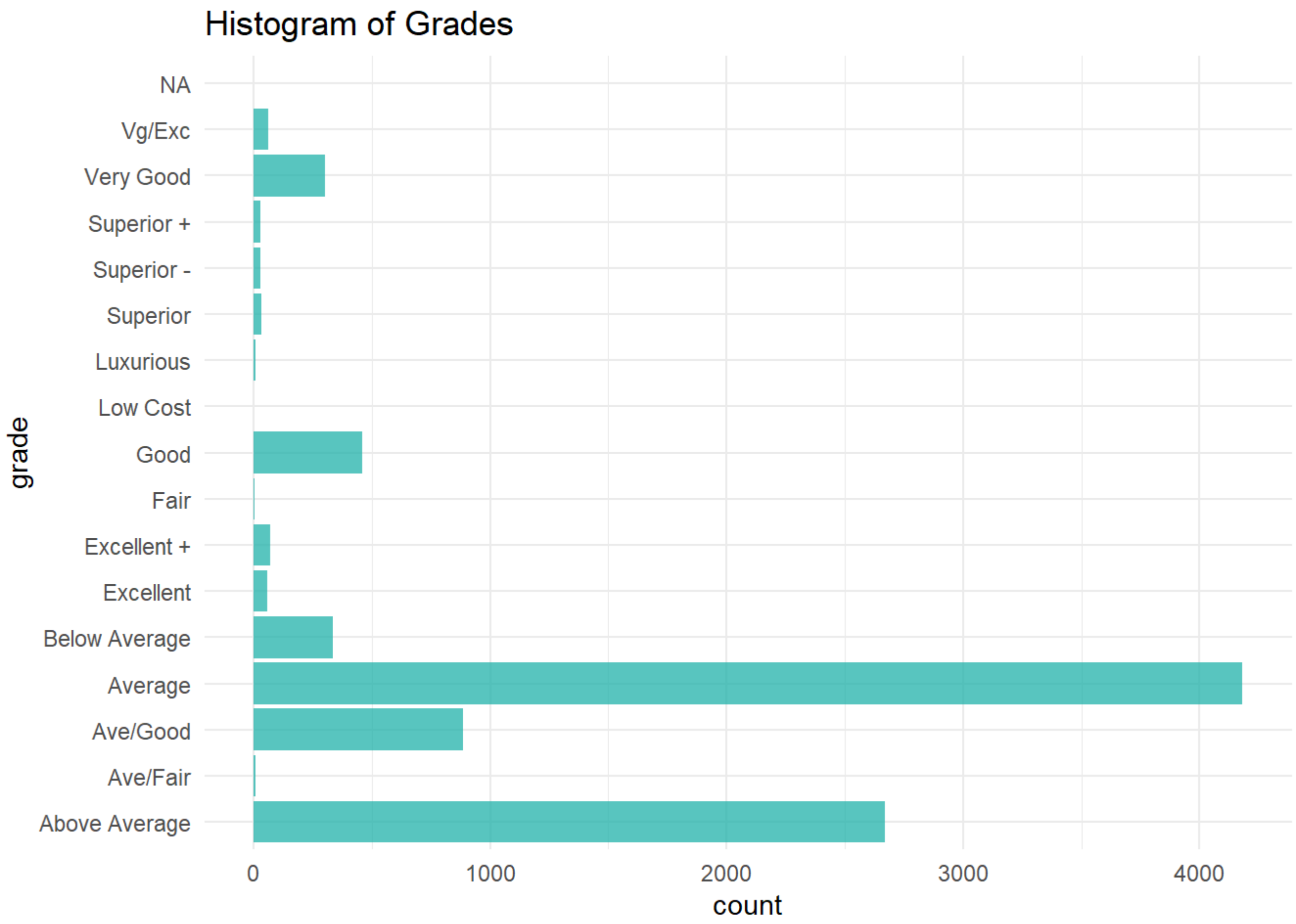
Colonial is by far the most frequent style in the data set.



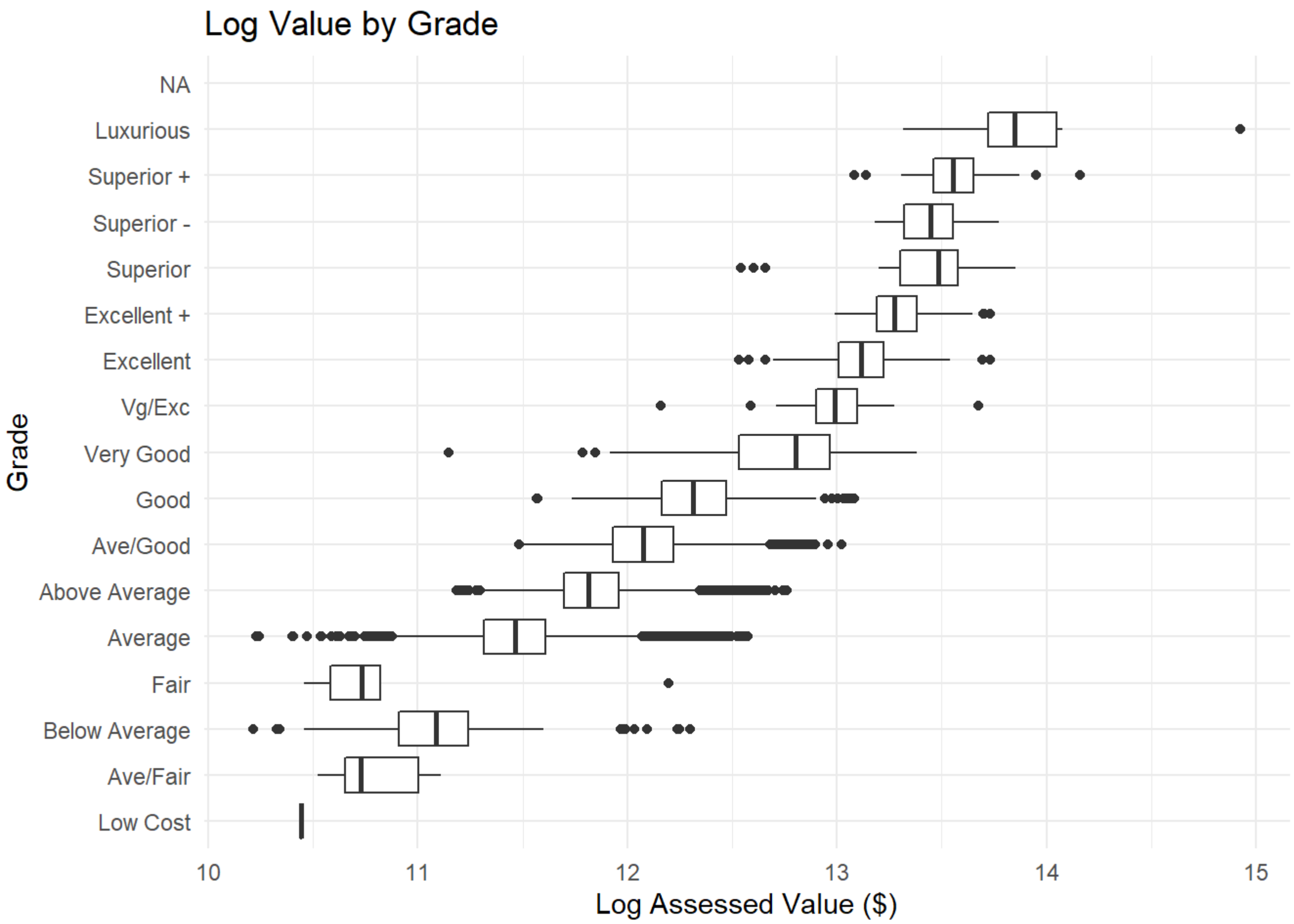
There doesn’t appear to be an easily interpretable relationship between style and log value from this plot.

Visualizing Grade

```
p13
```

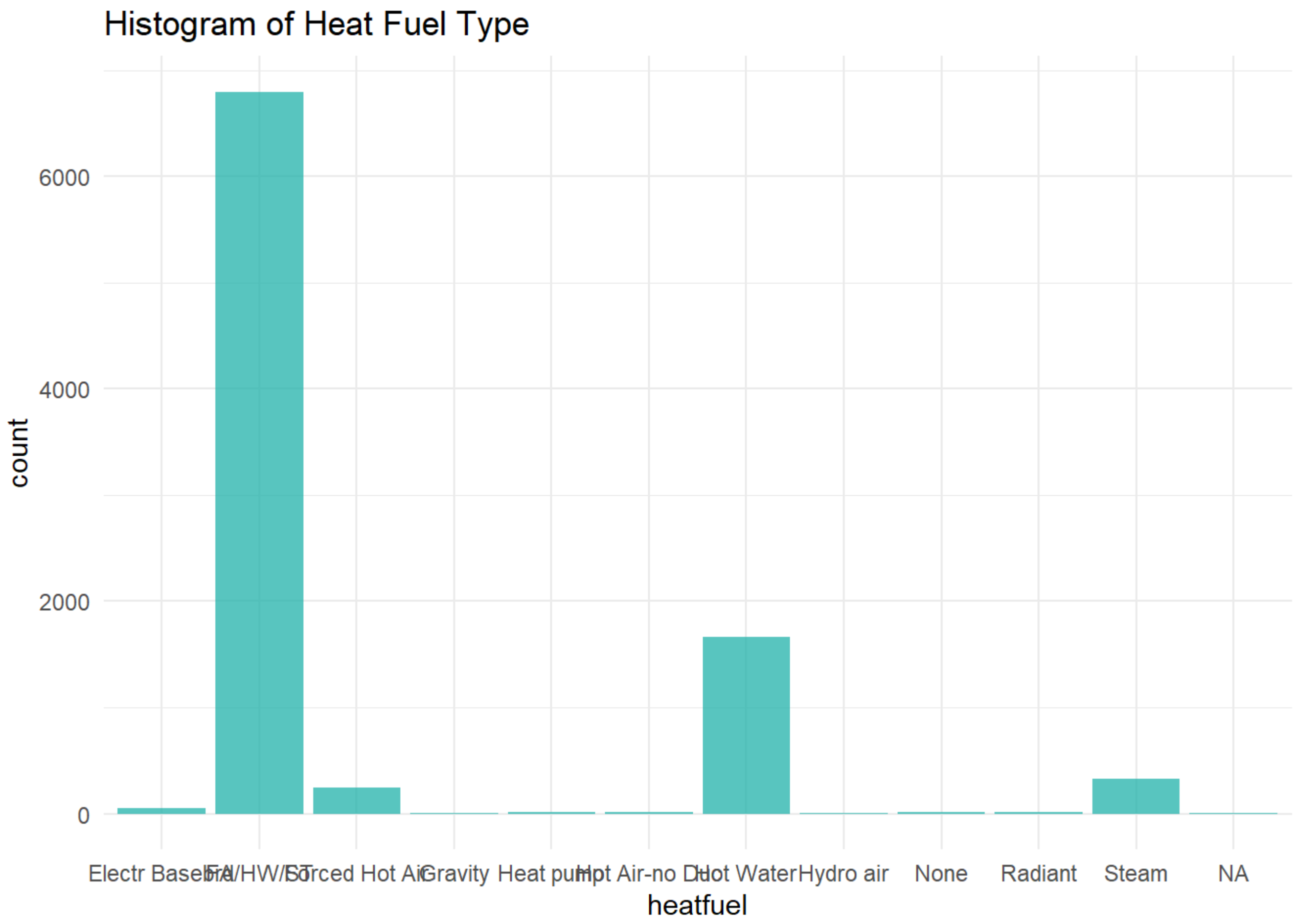


Most houses are given an average grade



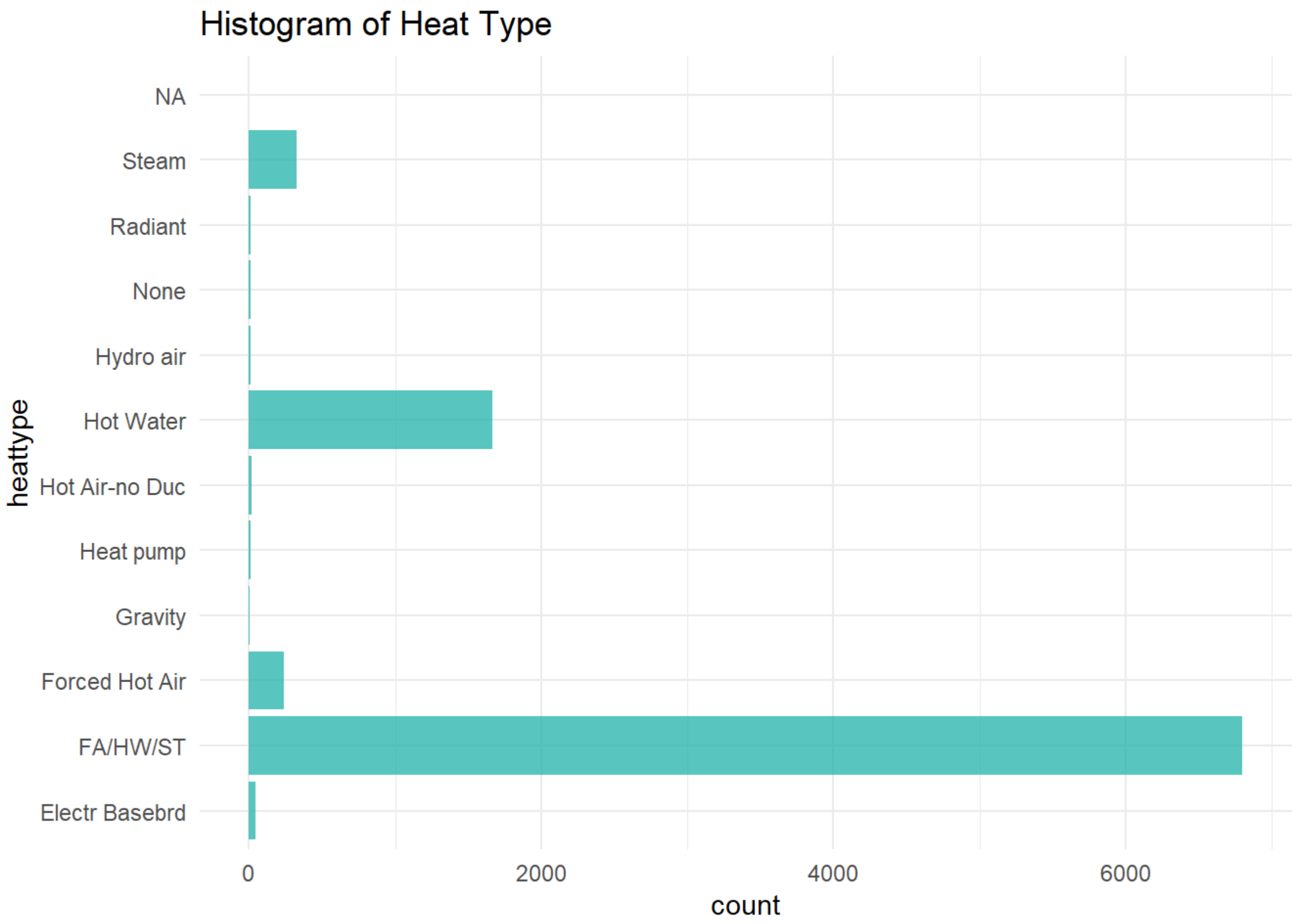
There is a predictable linear relationship between the grade and value of the houses in our data set.

Visualizing Heat Fuel

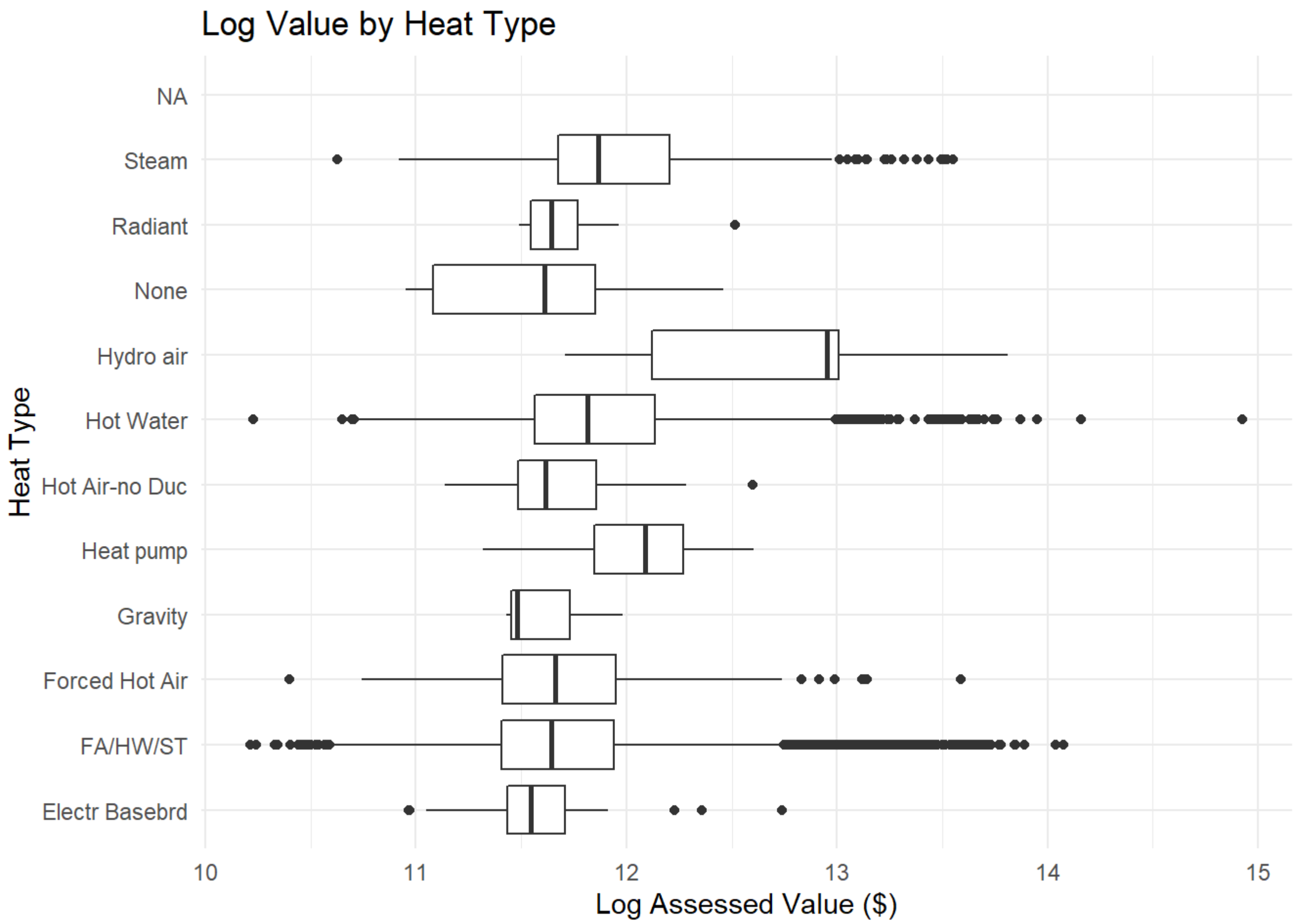


Given that Gas/Oil makes up the vast majority of data points, this may not be a variable of useful intepretability.

Visualizing Heat Type

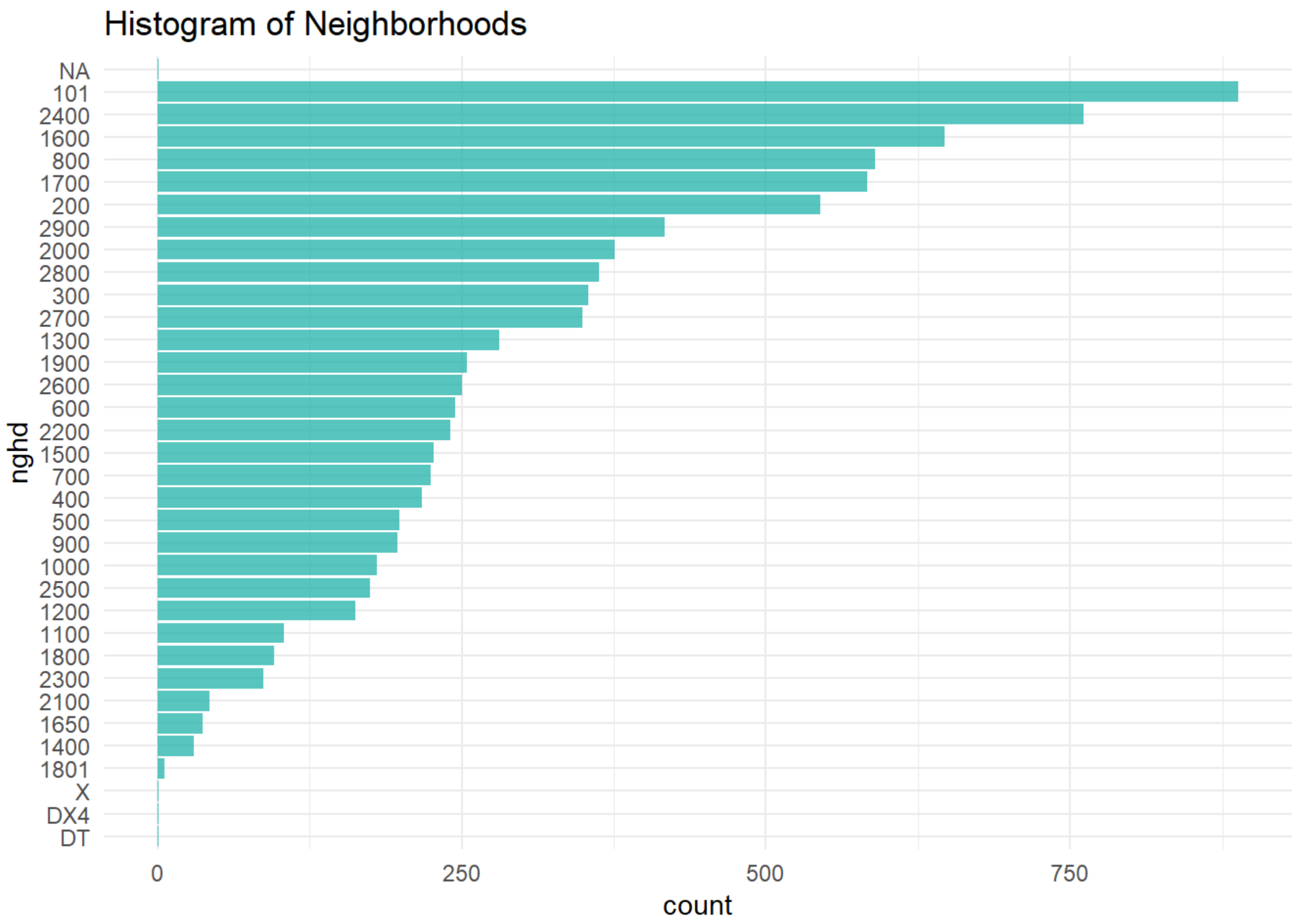


The histogram shows that most houses have FA/HW/ST heating. Research indicates that this probably stands for Forced Air/Hot Water/Steam.



There doesn’t appear to be an obvious relationship between heat type and log value.

Visualizing Neighborhoods



The most common neighborhood the houses are found in is labeled by code “0101”.

Building the Statistical Model

We tested both forwards and backwards stepwise regression and chose to use backwards regression, as it resulted in better and more interpretable models. However, the two didn’t produce vastly different results.

Building the Linear Model

```
####-----------SETUP-----------####
set.seed(31415)
folds <- 10

# Remove extraneous data, set rownames to PID
data <- dat[, !(names(dat) %in% c('pid', 'buildings', 'model', 'address', 'grade', 'heatfuel'))]

# Remove outliers
data <- data[data$beds > 0,]
data <- data[data$baths > 0,]

# Formatting the data
data$style <- as.factor(trimws(data$style))
data$heattype <- as.factor(trimws(data$heattype))
data$nghd <- as.factor(trimws(data$nghd))
data$beds <- as.factor(data$beds)
#data$ac <- as.factor(data$ac)

# Remove rows that are now identical
data <- unique(data)

# From previous analysis, we find that the log of certain values gives greater linear correlatio
n
# than the values themselves
data$value <- log(data$value)
data$land <- log(data$land)
data$living <- log(data$living)

# After performing the log, we need to remove 'Inf's that may have appeared
data <- filter(data, is.finite(data$land))


####-----------MAIN-----------####
# Randomize the order of checking the columns. Also ignore the value column, as it is the respon
se
# variable, and add '' to ensure that the first loop will use all columns.
selected <- names(data)[sample(length(data))]
selected <- c('', selected[selected != 'value'])

prev.RMSE <- Inf

for(col in selected) {
  avg.RMSE <- 0

  fields <- selected[selected != col]
  sub.data <- data[c('value', fields)]

  print(fields)

  for(i in 1:folds) {
    row.order <- sample(nrow(data))
    train.rows <- sub.data[head(row.order, nrow(data)*0.8),]
    test.rows <- sub.data[tail(row.order, -nrow(data)*0.8),]
    # The test.rows DF sometimes gets an NA row for no apparent reason...
    test.rows <- filter(test.rows, !is.na(test.rows$value))

    curr.model <- suppressWarnings(lm(value ~ ., train.rows))

    # Fill out the factor levels that were not used during training
    curr.model$xlevels$heattype <- levels(data$heattype)
    curr.model$xlevels$style <- levels(data$style)
    curr.model$xlevels$nghd <- levels(data$nghd)
    curr.model$xlevels$beds <- levels(data$beds)
    #curr.model$xlevels$ac <- levels(data$ac)

    predictions <- predict.lm(curr.model, test.rows)
    avg.RMSE = avg.RMSE + sqrt(mean((test.rows$value - predictions)^2))
  }

  avg.RMSE <- avg.RMSE / folds

  print(paste('Average:', avg.RMSE))
  if(avg.RMSE < prev.RMSE) {
    selected <- fields
    prev.RMSE <- avg.RMSE
    print('Updated!')
  }
}
```

```
##  [1] "year"     "land"     "baths"    "style"    "ac"       "good"    
##  [7] "living"   "nghd"     "heattype" "beds"    
## [1] "Average: 0.108163363624453"
## [1] "Updated!"
## [1] "land"     "baths"    "style"    "ac"       "good"     "living"   "nghd"    
## [8] "heattype" "beds"    
## [1] "Average: 0.106663423646665"
## [1] "Updated!"
## [1] "baths"    "style"    "ac"       "good"     "living"   "nghd"     "heattype"
## [8] "beds"    
## [1] "Average: 4.21678001351338"
## [1] "land"     "style"    "ac"       "good"     "living"   "nghd"     "heattype"
## [8] "beds"    
## [1] "Average: 4.08485096354615"
## [1] "land"     "baths"    "ac"       "good"     "living"   "nghd"     "heattype"
## [8] "beds"    
## [1] "Average: 0.104090558126585"
## [1] "Updated!"
## [1] "land"     "baths"    "good"     "living"   "nghd"     "heattype" "beds"    
## [1] "Average: 0.105018026768596"
## [1] "land"     "baths"    "ac"       "living"   "nghd"     "heattype" "beds"    
## [1] "Average: 0.159500597173696"
## [1] "land"     "baths"    "ac"       "good"     "nghd"     "heattype" "beds"    
## [1] "Average: 0.171328405031148"
## [1] "land"     "baths"    "ac"       "good"     "living"   "heattype" "beds"    
## [1] "Average: 0.236697641855314"
## [1] "land"   "baths"  "ac"     "good"   "living" "nghd"   "beds"  
## [1] "Average: 0.0983962067494233"
## [1] "Updated!"
## [1] "land"   "baths"  "ac"     "good"   "living" "nghd"  
## [1] "Average: 0.0985291174281575"
```

```
print(paste('Final RMSE:', prev.RMSE))
```

```
## [1] "Final RMSE: 0.0983962067494233"
```

```
print(selected)
```

```
## [1] "land"   "baths"  "ac"     "good"   "living" "nghd"   "beds"  
```

```
final.model <- lm(value ~ ., data[c('value', selected)])

summary(final.model)
```

```
##
## Call:
## lm(formula = value ~ ., data = data[c("value", selected)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26508 -0.05570 -0.00171  0.05138  0.67777
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.0552523  0.0372172 189.570  < 2e-16 ***
## land         0.0948661  0.0026075  36.383  < 2e-16 ***
## baths        0.0431866  0.0020323  21.250  < 2e-16 ***
## acTRUE       0.0322486  0.0024853  12.976  < 2e-16 ***
## good         0.0109429  0.0001273  85.974  < 2e-16 ***
## living       0.5358127  0.0047898 111.866  < 2e-16 ***
## nghd101      0.1404368  0.0080065  17.540  < 2e-16 ***
## nghd1100    -0.2535069  0.0120205 -21.090  < 2e-16 ***
## nghd1200     0.6354113  0.0105431  60.268  < 2e-16 ***
## nghd1300     0.7444863  0.0099210  75.041  < 2e-16 ***
## nghd1400     0.5038948  0.0193827  25.997  < 2e-16 ***
## nghd1500     0.8184892  0.0106075  77.161  < 2e-16 ***
## nghd1600    -0.3270813  0.0082152 -39.814  < 2e-16 ***
## nghd1650     0.1593599  0.0177706   8.968  < 2e-16 ***
## nghd1700    -0.0126871  0.0084237  -1.506  0.13207
## nghd1800    -0.1920428  0.0124751 -15.394  < 2e-16 ***
## nghd1801     0.8220537  0.0404107  20.342  < 2e-16 ***
## nghd1900    -0.2219309  0.0095471 -23.246  < 2e-16 ***
## nghd200      0.0818217  0.0085402   9.581  < 2e-16 ***
## nghd2000    -0.3754592  0.0088775 -42.293  < 2e-16 ***
## nghd2100     0.1059406  0.0165027   6.420 1.44e-10 ***
## nghd2200     0.0794300  0.0096562   8.226  < 2e-16 ***
## nghd2300     0.0283142  0.0127590   2.219  0.02650 *
## nghd2400     0.2396596  0.0082885  28.915  < 2e-16 ***
## nghd2500     0.0307950  0.0103506   2.975  0.00294 **
## nghd2600    -0.0595004  0.0096375  -6.174 6.95e-10 ***
## nghd2700    -0.0453697  0.0089851  -5.049 4.52e-07 ***
## nghd2800     0.0136613  0.0090111   1.516  0.12954
## nghd2900     0.0886729  0.0089915   9.862  < 2e-16 ***
## nghd300      0.0014686  0.0089698   0.164  0.86995
## nghd400     -0.0647794  0.0098454  -6.580 4.98e-11 ***
## nghd500     -0.1099634  0.0100751 -10.914  < 2e-16 ***
## nghd600     -0.0727095  0.0097991  -7.420 1.28e-13 ***
## nghd700     -0.1799674  0.0098558 -18.260  < 2e-16 ***
## nghd800     -0.0641142  0.0084947  -7.548 4.86e-14 ***
## nghd900     -0.4117049  0.0100542 -40.949  < 2e-16 ***
## nghdDT       1.3632460  0.0974998  13.982  < 2e-16 ***
## nghdDX4      0.0027585  0.0974582   0.028  0.97742
## nghdX       -0.7210976  0.0975515  -7.392 1.58e-13 ***
## beds2        0.0236142  0.0104993   2.249  0.02453 *
## beds3        0.0324954  0.0103663   3.135  0.00173 **
## beds4        0.0291275  0.0107269   2.715  0.00663 **
## beds5        0.0309984  0.0117878   2.630  0.00856 **
## beds6        0.0387007  0.0132142   2.929  0.00341 **
## beds7        0.0097379  0.0177804   0.548  0.58393
## beds8        0.0839936  0.0258326   3.251  0.00115 **
## beds9        0.0698782  0.0368742   1.895  0.05812 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09715 on 9062 degrees of freedom
## Multiple R-squared:  0.963,  Adjusted R-squared:  0.9628
## F-statistic:  5132 on 46 and 9062 DF,  p-value: < 2.2e-16
```

Observations about the Model

Our final linear model indicates highly significant relationships between assessed value and land, baths, ac, good, living, and neighborhood (most p-values < 2e-16, a couple p-value < 0.05 in neighborhood). Year was initially included and then removed due to a non-significant contribution. Because we had factor variables, the intercept describes a single family home with all the baseline reference levels (“no” AC and “0101” neighborhood). The fact that the chosen independent variables for our model had significant influences over our dependent variable are not surprising, as we went through a rigorous selection process to choose these variables to ensure that they benefited the model.

Land has a positive relationship with the assessed value of a single family home. For every 1% increase in land (in acres), assessed value increases by 0.095% (both variables are log-transformed, so the coefficient represents the percent change in the dependent variable for every 1% increase in the independent variable). This makes sense intuitively—land is desirable, and the larger the plot of land the home is on, the more likely it would be worth more.

Baths also has a positive relationship with the assessed value of a single family home. As baths increases by one (either through 1 full bath or 2 half baths), assessed value increases by about 4.44% (coefficient is 0.0435, so (exp(0.1231)-1)*100 = 4.44%). 4.44%% is relatively large, and again, this makes sense on an intuitive level. The more bathrooms, the more expensive the home would be.

Having AC has a positive impact with the assessed value of a single family home. Going from having no AC to having AC would increase the home value by about 5% (coefficient = 0.032, so (exp(0.032)-1)*100 = 3.25%). AC is a highly desirable amenity for a home, so accordingly it would also boost property value.

Percent good has a positive relationship with the assessed value of a single family home. As percent good increases by 1%, assessed value increases by 1.15% (coefficient = 0.011, so (exp(0.011)-1)*100 = 1.15%). Percent good is already an estimate of value by professionals, so again this makes sense that there would be a boost in assessed value if percent good increases.

Neighborhood also matters. The rationale for why neighborhoods “X,” “2300,” “1800,” “1100,” “2500,” “1000,” “0900,” “0500,” “0400,” “0700,” “0600,” “2600,” “1900,” “2700,” “0300,” “2800,” “2000,” “0200,” “1700,” “0800,” and “1600” had lesser assessed value, but why neighborhoods “DT,” “1801,” “1400,” “1200,” “1500,” “2200,” “1300,” “2900,” and “2400” had higher assessed value when compared to the reference level of “0100” is not easily understood without knowing more about these individual neighborhoods. A vast majority of these neighborhoods have highly significant coefficients (p-value <2e-16), so they are clearly distinguished from each other.

Predictions

```
pids <- c('1000', '12824', '12773', '19128')

dat$value <- log(dat$value)
dat$land <- log(dat$land)
dat$living <- log(dat$living)
dat$beds <- as.factor(dat$beds)

pred <- data.frame(1:4)
pred$pid <- pids
pred$clevergroupname <- NA
pred$yhat <- NA
pred$lower <- NA
pred$upper <- NA

for(i in 1:length(pids)) {
  p = pids[i]
  interval <- predict.lm(final.model, dat[dat$pid == p,][1,], interval = 'predict')
  print(paste('True value:', exp(dat[dat$pid == p,][1,]$value)))
  print('Prediction interval:', )
  print(exp(interval))

  pred[i,"yhat"] <- exp(interval[1])
  pred[i, "lower"] <- exp(interval[2])
  pred[i, "upper"] <- exp(interval[3])
}
```

```
## [1] "True value: 140980"
## [1] "Prediction interval:"
##          fit      lwr    upr
## 707 136820.1 113078.1 165547
## [1] "True value: 343350"
## [1] "Prediction interval:"
##           fit      lwr      upr
## 4337 294119.8 242762.7 356341.6
## [1] "True value: 547330"
## [1] "Prediction interval:"
##           fit      lwr    upr
## 4322 514882.4 425370.9 623230
## [1] "True value: 89040"
## [1] "Prediction interval:"
##           fit      lwr      upr
## 5392 96686.54 79886.19 117020.1
```

```
pred$clevergroupname <- "DTL"
```

```
pred
```

X1.4	pid	clevergroupname	yhat	lower	upper
<int>	<chr>	<chr>	<dbl>	<dbl>	<dbl>

X1.4	pid	clevergroupname	yhat	lower	upper
<int>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
1	1000	DTL	136820.09	113078.07	165547.0
2	12824	DTL	294119.78	242762.72	356341.6
3	12773	DTL	514882.42	425370.88	623230.0
4	19128	DTL	96686.54	79886.19	117020.1

4 rows

```
write.csv(pred,"DTL.csv")
```