Final_Project

Libaries

```
pacman::p_load(data.table,tidyverse,magrittr,YARF,skimr,plyr,tidyr,YARF,mltools,caret)
```

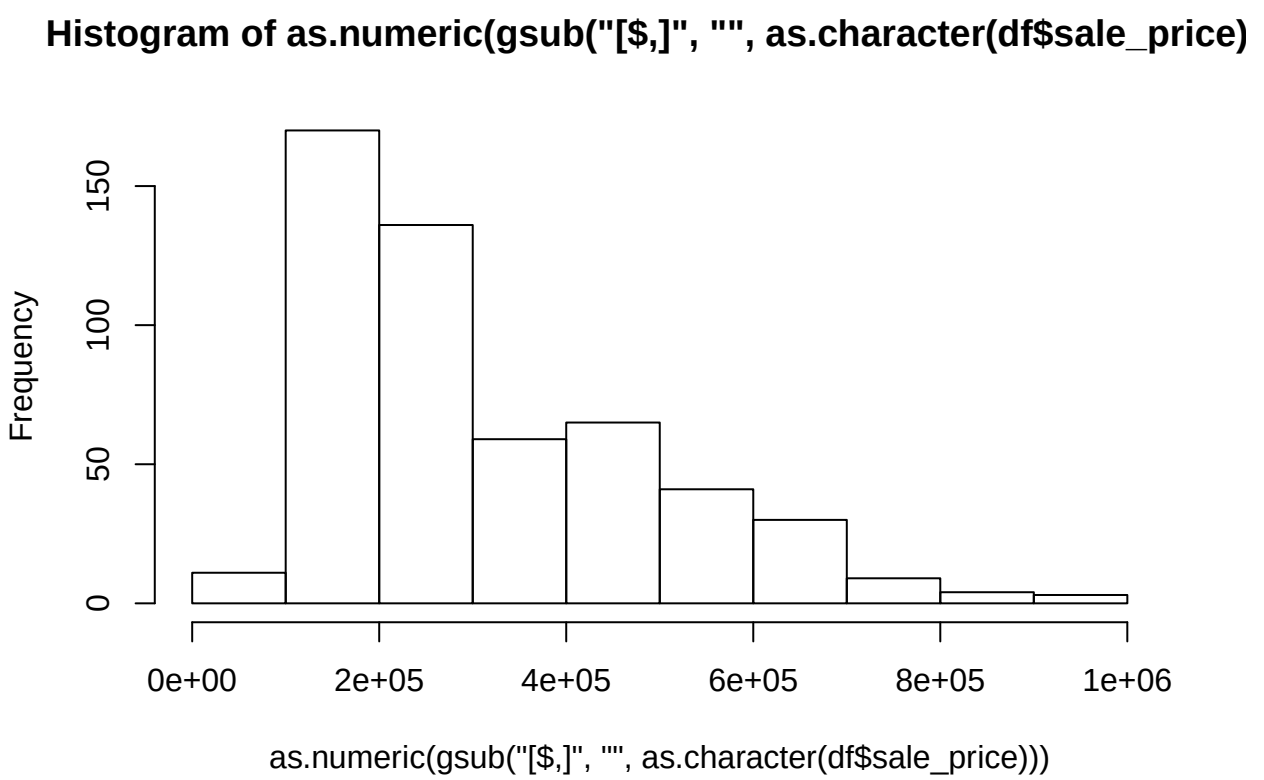
Loading Data

```
df= read.csv("housing_data_2016_2017.csv")
head(df,1)
```

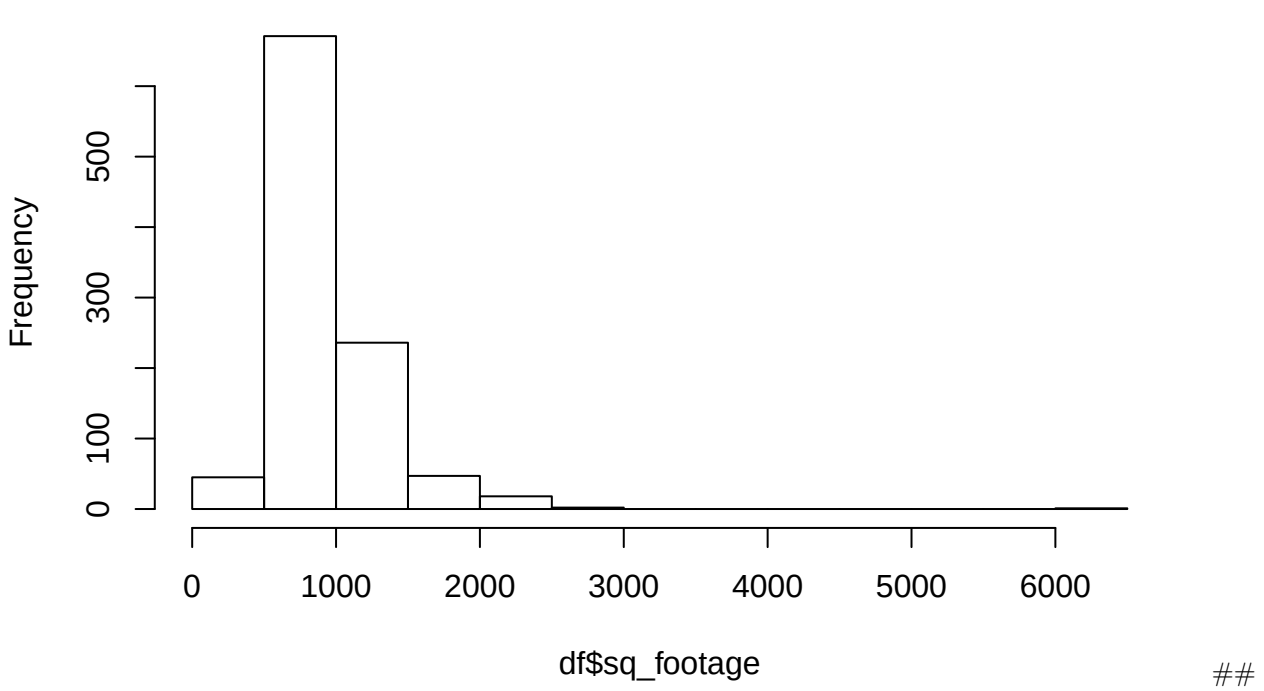
```
##                            HITId                       HITTypeId
## 1 3OID399FXG7F26JWONXF0Y86J90FD4 36BILMLQB75QQNBTYKGYCZWDN8TVAU
##                                                                Title
## 1 Find Information about Housing To Help a Student Project -- Very easy
##                                    Description Keywords Reward
## 1 Go to a link and copy information into the HIT       NA $0.05
##                   CreationTime MaxAssignments
## 1 Wed Feb 15 22:13:37 PST 2017              1
##                              RequesterAnnotation AssignmentDurationInSeconds
## 1 BatchId:2689947;OriginalHitTemplateId:920937336;                        900
##   AutoApprovalDelayInSeconds                   Expiration NumberOfSimilarHITs
## 1                         60 Wed Feb 22 22:13:37 PST 2017                  NA
##   LifetimeInSeconds                    AssignmentId       WorkerId
## 1                NA 32KTQ2V7RDFCSAWQOW1SXC5AZIC9MB A231MNJJDDF3LS
##   AssignmentStatus                    AcceptTime                    SubmitTime
## 1         Approved Thu Feb 16 05:32:36 PST 2017 Thu Feb 16 05:35:37 PST 2017
##               AutoApprovalTime            ApprovalTime RejectionTime
## 1 Thu Feb 16 05:36:37 PST 2017 2017-02-16 13:37:11 UTC            NA
##   RequesterFeedback WorkTimeInSeconds LifetimeApprovalRate
## 1                NA               181       100% (187/187)
##   Last30DaysApprovalRate Last7DaysApprovalRate
## 1         100% (187/187)        100% (187/187)
##                                                                                                                                  URL
## 1 http://www.mlsli.com/homes-for-sale/address-not-available-from-broker-Flushing-NY-11355-149238320
##   approx_year_built cats_allowed common_charges community_district_num
## 1              1955           no          $767                      25
##   coop_condo date_of_sale dining_room_type dogs_allowed fuel_type
## 1      co-op    2/16/2016            combo           no       gas
##   full_address_or_zip_code garage_exists kitchen_type maintenance_cost
## 1       Flushing NY, 11355          <NA>       eat in             <NA>
##          model_type num_bedrooms num_floors_in_building num_full_bathrooms
## 1 Mitchell Garden 3            2                      6                  1
##   num_half_bathrooms num_total_rooms parking_charges pct_tax_deductibl
## 1                 NA               5            <NA>                NA
##   sale_price sq_footage total_taxes walk_score listing_price_to_nearest_1000
## 1  $228,000          NA        <NA>         82                          <NA>
##    url
```

```
## 1 <NA>
```

```
hist(as.numeric(gsub('[$,]','',as.character(df$sale_price))))
```




```
hist(df$sq_footage)
```



##

Useful Summary of Data This gives a broad overview on the data,

```
skim(df)
```

Table 1: Data summary

Name	df
Number of rows	2230
Number of columns	55
Column type frequency:	
factor	36
logical	5
numeric	14
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
HITId	758	0.66	FALSE	1472	301: 1, 301: 1, 301: 1, 302: 1
HITTypeId	758	0.66	FALSE	2	310: 944, 36B: 528
Title	758	0.66	FALSE	1	Fin: 1472
Description	758	0.66	FALSE	2	Got: 944, Go : 528
Reward	758	0.66	FALSE	1	$0.: 1472
CreationTime	758	0.66	FALSE	62	Thu: 43, Thu: 40, Wed: 39, Thu: 37
RequesterAnnotation	758	0.66	FALSE	2	Bat: 944, Bat: 528
Expiration	758	0.66	FALSE	62	Thu: 43, Thu: 40, Wed: 39, Thu: 37
AssignmentId	758	0.66	FALSE	1472	301: 1, 301: 1, 304: 1, 304: 1
WorkerId	758	0.66	FALSE	73	A23: 187, A1S: 129, A3C: 124, AHX
AssignmentStatus	758	0.66	FALSE	1	App: 1472
AcceptTime	758	0.66	FALSE	1457	Thu: 2, Thu: 2, Thu: 2, Thu: 2
SubmitTime	758	0.66	FALSE	1460	Thu: 2, Thu: 2, Thu: 2, Thu: 2
AutoApprovalTime	758	0.66	FALSE	1460	Thu: 2, Thu: 2, Thu: 2, Thu: 2
ApprovalTime	758	0.66	FALSE	929	201: 6, 201: 6, 201: 5, 201: 5
LifetimeApprovalRate	758	0.66	FALSE	32	100: 187, 100: 126, 100: 124, 100: 10
Last30DaysApprovalRate	758	0.66	FALSE	32	100: 187, 100: 126, 100: 124, 100: 10
Last7DaysApprovalRate	758	0.66	FALSE	32	100: 187, 100: 126, 100: 124, 100: 10
URL	758	0.66	FALSE	1450	htt: 2, htt: 2, htt: 2, htt: 2
cats_allowed	0	1.00	FALSE	3	no: 1402, yes: 826, y: 2
common_charges	1684	0.24	FALSE	258	$25: 11, $17: 10, $27: 9, $29: 8
coop_condo	0	1.00	FALSE	2	co-: 1661, con: 569
date_of_sale	1702	0.24	FALSE	222	6/3: 7, 10/: 6, 12/: 6, 2/2: 6
dining_room_type	448	0.80	FALSE	5	com: 957, for: 620, oth: 201, din: 2
dogs_allowed	0	1.00	FALSE	3	no: 1684, yes: 544, yes: 2
fuel_type	112	0.95	FALSE	6	gas: 1348, oil: 664, ele: 62, oth: 40
full_address_or_zip_code	0	1.00	FALSE	1177	70-: 22, 269: 17, 270: 16, 73-: 14
garage_exists	1826	0.18	FALSE	6	yes: 361, Yes: 39, 1: 1, eys: 1
kitchen_type	16	0.99	FALSE	13	eat: 733, eff: 505, com: 349, eff: 338
maintenance_cost	623	0.72	FALSE	609	$54: 10, $67: 10, $68: 10, $70: 10
model_type	40	0.98	FALSE	875	1 B: 63, One: 59, 2 B: 50, Hi-: 41
parking_charges	1671	0.25	FALSE	89	$15: 42, $60: 41, $75: 27, $13: 23
sale_price	1702	0.24	FALSE	315	$15: 11, $17: 10, $13: 7, $22: 7

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
total_taxes	1646	0.26	FALSE	293	$13: 13, $25: 12, $4,: 11, $2,: 10
listing_price_to_nearest_1000	534	0.76	FALSE	292	$34: 28, $39: 26, $28: 25, $23: 23
url	758	0.66	FALSE	1450	htt: 2, htt: 2, htt: 2, htt: 2

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
Keywords	2230	0	NaN	:
NumberOfSimilarHITs	2230	0	NaN	:
LifetimeInSeconds	2230	0	NaN	:
RejectionTime	2230	0	NaN	:
RequesterFeedback	2230	0	NaN	:

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
MaxAssignments	758	0.66	1.00	0.00	1	1	1	1	1	
AssignmentDurationInSeconds	758	0.66	900.00	0.00	900	900	900	900	900	
AutoApprovalDelayInSeconds	758	0.66	60.00	0.00	60	60	60	60	60	
WorkTimeInSeconds	758	0.66	162.39	111.69	22	89	127	197	815	
approx_year_built	40	0.98	1962.71	21.08	1893	1950	1958	1970	2017	
community_district_num	19	0.99	26.33	2.95	3	25	26	28	32	
num_bedrooms	115	0.95	1.65	0.74	0	1	2	2	6	
num_floors_in_building	650	0.71	7.79	7.52	1	3	6	7	34	
num_full_bathrooms	0	1.00	1.23	0.44	1	1	1	1	3	
num_half_bathrooms	2058	0.08	0.95	0.30	0	1	1	1	2	
num_total_rooms	2	1.00	4.14	1.35	0	3	4	5	14	
pct_tax_deductibl	1754	0.21	45.40	6.95	20	40	50	50	75	
sq_footage	1210	0.46	955.36	380.86	100	743	881	1100	6215	
walk_score	0	1.00	83.92	14.75	7	77	89	95	99	

There is a lot of data that is completely missing and some that is heavily missing. I decided to remove them. Some examples below.

Keywords,NumberOfSimilarHITs, LigetimeInSeconds, RejectionTime,RequesterFeedback all completely missing. ommon_charges(missing 1684),garage_exists(missing 1826)

```
cat("Data has",nrow(df),"number of rows\n")
```

```
## Data has 2230 number of rows
```

```
cat("Data has",ncol(df), "number of columns")
```

```
## Data has 55 number of columns
```

```
sort(colMeans(is.na(df)), decreasing = TRUE)
```

```
##                       Keywords           NumberOfSimilarHITs
##                    1.000000000                   1.000000000
##             LifetimeInSeconds                 RejectionTime
```

```
##                  1.000000000                  1.000000000
##             RequesterFeedback            num_half_bathrooms
##                  1.000000000                  0.922869955
##                garage_exists             pct_tax_deductibl
##                  0.818834081                  0.786547085
##                date_of_sale                   sale_price
##                  0.763228700                  0.763228700
##               common_charges               parking_charges
##                  0.755156951                  0.749327354
##                total_taxes                          url
##                  0.738116592                  0.660089686
##                 sq_footage                        HITId
##                  0.542600897                  0.339910314
##                  HITTypeId                        Title
##                  0.339910314                  0.339910314
##                Description                       Reward
##                  0.339910314                  0.339910314
##               CreationTime                MaxAssignments
##                  0.339910314                  0.339910314
##          RequesterAnnotation   AssignmentDurationInSeconds
##                  0.339910314                  0.339910314
##    AutoApprovalDelayInSeconds                   Expiration
##                  0.339910314                  0.339910314
##                AssignmentId                     WorkerId
##                  0.339910314                  0.339910314
##              AssignmentStatus                   AcceptTime
##                  0.339910314                  0.339910314
##                  SubmitTime              AutoApprovalTime
##                  0.339910314                  0.339910314
##                ApprovalTime             WorkTimeInSeconds
##                  0.339910314                  0.339910314
##         LifetimeApprovalRate        Last30DaysApprovalRate
##                  0.339910314                  0.339910314
##        Last7DaysApprovalRate                          URL
##                  0.339910314                  0.339910314
##       num_floors_in_building              maintenance_cost
##                  0.291479821                  0.279372197
## listing_price_to_nearest_1000              dining_room_type
##                  0.239461883                  0.200896861
##                num_bedrooms                    fuel_type
##                  0.051569507                  0.050224215
##             approx_year_built                   model_type
##                  0.017937220                  0.017937220
##       community_district_num                 kitchen_type
##                  0.008520179                  0.007174888
##              num_total_rooms                cats_allowed
##                  0.000896861                  0.000000000
##                   coop_condo                dogs_allowed
##                  0.000000000                  0.000000000
##     full_address_or_zip_code            num_full_bathrooms
##                  0.000000000                  0.000000000
##                   walk_score
##                  0.000000000
```

Data Cleaning Remove all missing y

```
df_drops = df %>% drop_na(sale_price)
skim(df_drops) %>%
  summary()
```

Table 5: Data summary

Name	df_drops
Number of rows	528
Number of columns	55
Column type frequency:	
factor	36
logical	5
numeric	14
Group variables	None

Meaningful Features Data Cleaning

Finding meaningful features. These are features I believe are meaningful. df_mutated has all the features that I will be using. I am not looking at whats missing yet or how the data looks like, just looking for features that would be best to predict sales price.

```
colnames(df)
```

```
##  [1] "HITId"                         "HITTypeId"
##  [3] "Title"                         "Description"
##  [5] "Keywords"                      "Reward"
##  [7] "CreationTime"                  "MaxAssignments"
##  [9] "RequesterAnnotation"           "AssignmentDurationInSeconds"
## [11] "AutoApprovalDelayInSeconds"    "Expiration"
## [13] "NumberOfSimilarHITs"           "LifetimeInSeconds"
## [15] "AssignmentId"                  "WorkerId"
## [17] "AssignmentStatus"              "AcceptTime"
## [19] "SubmitTime"                    "AutoApprovalTime"
## [21] "ApprovalTime"                  "RejectionTime"
## [23] "RequesterFeedback"             "WorkTimeInSeconds"
## [25] "LifetimeApprovalRate"          "Last30DaysApprovalRate"
## [27] "Last7DaysApprovalRate"         "URL"
## [29] "approx_year_built"             "cats_allowed"
## [31] "common_charges"                "community_district_num"
## [33] "coop_condo"                    "date_of_sale"
## [35] "dining_room_type"              "dogs_allowed"
## [37] "fuel_type"                     "full_address_or_zip_code"
## [39] "garage_exists"                 "kitchen_type"
## [41] "maintenance_cost"              "model_type"
## [43] "num_bedrooms"                  "num_floors_in_building"
## [45] "num_full_bathrooms"            "num_half_bathrooms"
## [47] "num_total_rooms"               "parking_charges"
## [49] "pct_tax_deductibl"             "sale_price"
## [51] "sq_footage"                    "total_taxes"
```

```
## [53] "walk_score"                    "listing_price_to_nearest_1000"
## [55] "url"
```

```
df_mutated = copy(df_drops)
df_mutated %<>%
  select(cats_allowed,common_charges,coop_condo,dining_room_type,dogs_allowed,fuel_type,garage_exists,ma
sort(colMeans(is.na(df_mutated)), decreasing = TRUE)
```

```
##          garage_exists            total_taxes         common_charges
##            0.821969697            0.751893939            0.750000000
##             sq_footage       maintenance_cost       dining_room_type
##            0.596590909            0.268939394            0.227272727
## num_floors_in_building              fuel_type             model_type
##            0.204545455            0.045454545            0.028409091
##      approx_year_built community_district_num           cats_allowed
##            0.011363636            0.001893939            0.000000000
##             coop_condo           dogs_allowed           num_bedrooms
##            0.000000000            0.000000000            0.000000000
##     num_full_bathrooms        num_total_rooms             sale_price
##            0.000000000            0.000000000            0.000000000
##             walk_score
##            0.000000000
```

Feature Data Cleaning

I am now looking more closely to the data.Looking at this there are too many types of model_types 875 different times from original data with NA sale price this seems difficult to deal with so I will remove this. I discarded data with more than 50% of missinginess.

```
df_mutated_features = copy(df_mutated)
df_mutated_features %<>%
  select(-model_type,-total_taxes)#,-common_charges,-sq_footage)
skim(df_mutated_features) %>%
  summary()
```

Table 6: Data summary

Name	df_mutated_features
Number of rows	528
Number of columns	17
Column type frequency:	
factor	9
numeric	8
Group variables	None

```
sort(colMeans(is.na(df_mutated_features)), decreasing = TRUE)
```

```
##          garage_exists         common_charges              sq_footage
##            0.821969697            0.750000000            0.596590909
##       maintenance_cost       dining_room_type num_floors_in_building
##            0.268939394            0.227272727            0.204545455
##             fuel_type      approx_year_built community_district_num
```

```
##           0.045454545           0.011363636           0.001893939
##         cats_allowed             coop_condo          dogs_allowed
##           0.000000000           0.000000000           0.000000000
##         num_bedrooms     num_full_bathrooms       num_total_rooms
##           0.000000000           0.000000000           0.000000000
##           sale_price            walk_score
##           0.000000000           0.000000000
```

Oberservations Data Cleaning

I am okay with the number of features I have now. Now Ill be cleaning the observations.

```
df_clean = copy(df_mutated_features)


# Fixing y to be just yes and reducing factors to just yes and no.
df_clean %<>%
  mutate(cats_allowed = as.factor(ifelse(cats_allowed =='y' | cats_allowed =='yes','yes','no'))) %>%

#Fixing yes89 to just yes and reducing factors to just yes and no
  mutate(dogs_allowed = as.factor(ifelse(dogs_allowed =='yes89' | dogs_allowed =='yes','yes','no'))) %>%

  #mutate(sale_price = as.numeric(gsub('[$]','',as.character(df_clean$sale_price))))
  mutate(sale_price = as.numeric(gsub('[$,]','',as.character(df_clean$sale_price))) )%>%

  mutate(common_charges = as.numeric(gsub('[$,]','',as.character(df_clean$common_charges)))) %>%

  mutate(maintenance_cost = as.numeric(gsub('[$,]','',as.character(df_clean$maintenance_cost)))) %>%

  mutate(garage_exists = ifelse(is.na(garage_exists), 0, 1))

#Very annoying this best way I found to combine two factor lvels
library(forcats)
df_clean$fuel_type = fct_collapse(df_clean$fuel_type, other = c("other","Other"))



# df_clean_sub = copy(df_clean)
# df_clean_sub = df_clean_sub[df_clean_sub$sale_price < 700000,]
#
# df_clean = df_clean_sub
```

```
max(df_clean$sq_footage, na.rm = TRUE)
```

```
## [1] 6215
```

```
min(df_clean$sq_footage, na.rm = TRUE)
```

```
## [1] 375
```

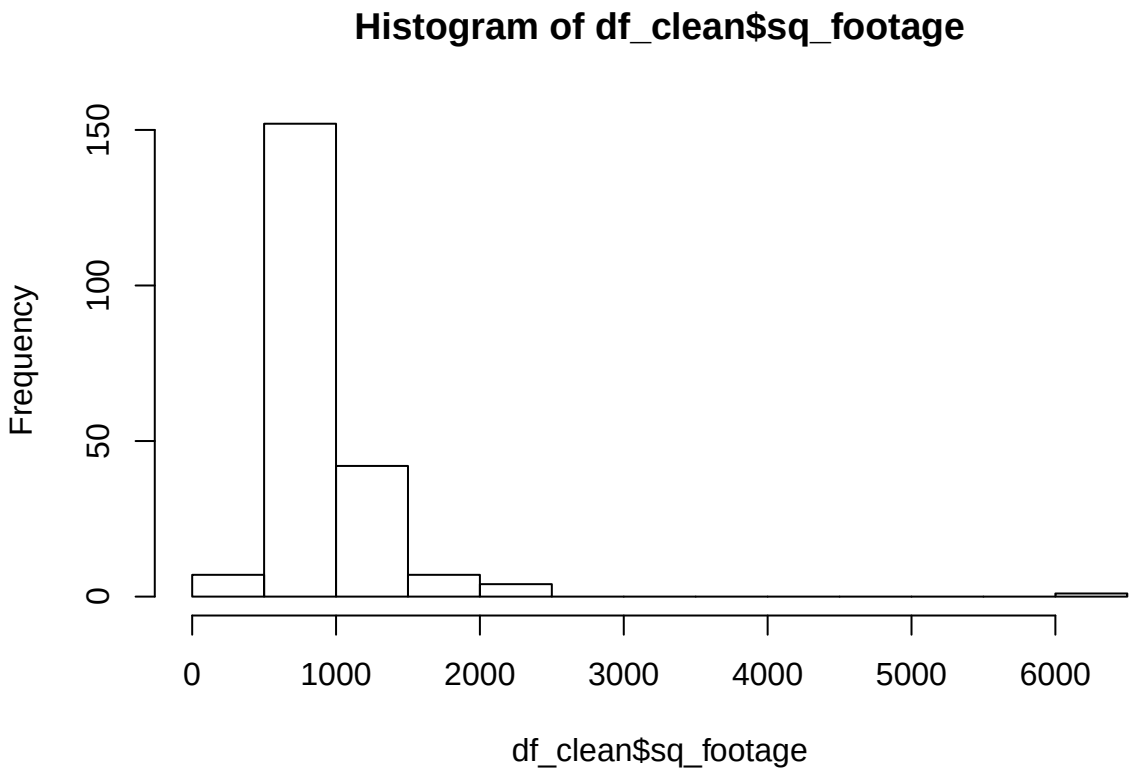
```
max(df_clean$sale_price, na.rm = TRUE)
```

```
## [1] 999999
```

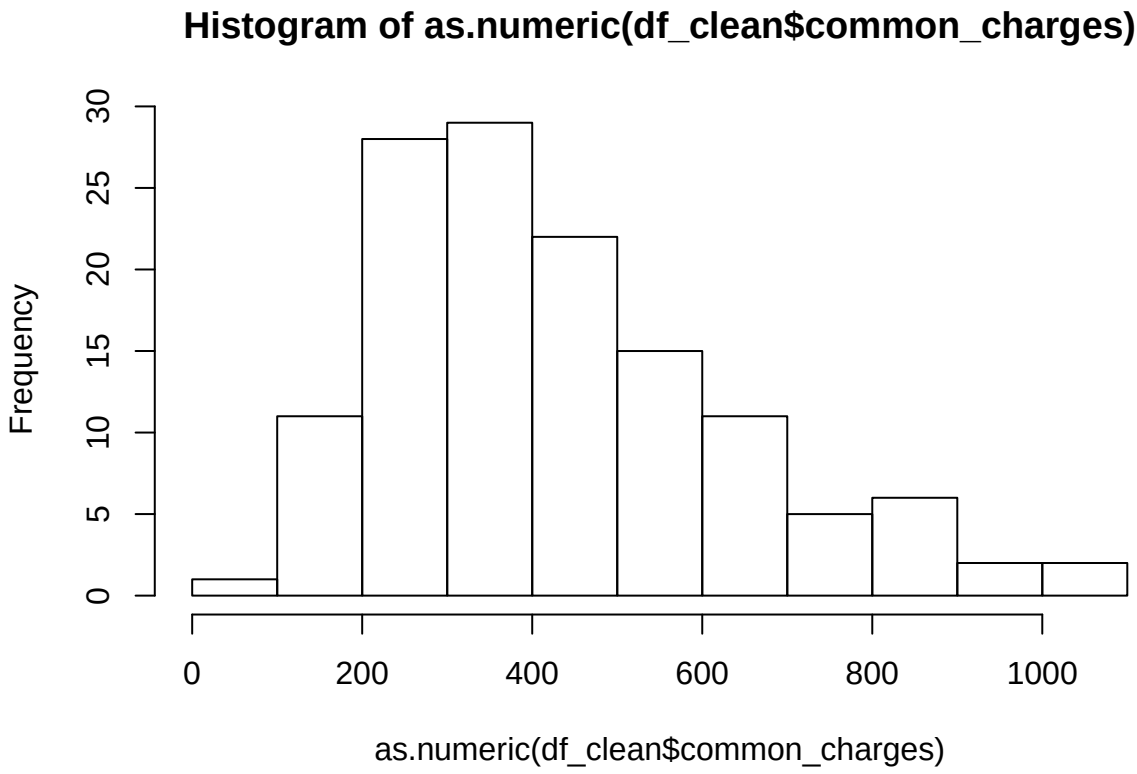
```
min(df_clean$sale_price, na.rm = TRUE)
```

```
## [1] 55000
```

```r
hist(df_clean$sq_footage)
```




```r
hist(as.numeric(df_clean$common_charges))
```



```
#df_clean %<>%
  #select(-sq_footage)
```

```
df_clean$fuel_type
```

```
##   [1] gas      oil      <NA>     gas      gas      oil      gas      gas
##   [9] oil      <NA>     gas      oil      gas      oil      gas      oil
##  [17] other    oil      gas      oil      oil      oil      gas      oil
##  [25] oil      gas      gas      gas      other    oil      gas      gas
##  [33] gas      <NA>     gas      gas      gas      gas      gas      oil
##  [41] gas      gas      oil      oil      gas      gas      oil      oil
##  [49] oil      gas      gas      gas      gas      gas      oil      oil
##  [57] oil      gas      gas      oil      gas      gas      gas      gas
##  [65] gas      oil      gas      gas      gas      gas      gas      oil
##  [73] gas      gas      gas      gas      oil      oil      oil      gas
##  [81] gas      gas      oil      gas      gas      gas      oil      gas
##  [89] oil      oil      gas      oil      oil      oil      oil      gas
##  [97] electric gas      <NA>     oil      oil      gas      oil      gas
## [105] oil      oil      gas      <NA>     oil      gas      gas      gas
## [113] oil      gas      gas      oil      oil      oil      gas      gas
## [121] gas      oil      gas      other    gas      gas      oil      oil
## [129] gas      gas      oil      gas      gas      oil      gas      gas
## [137] gas      oil      gas      gas      oil      oil      gas      oil
## [145] gas      gas      oil      oil      gas      gas      <NA>     gas
## [153] gas      oil      gas      gas      gas      oil      gas      oil
## [161] gas      gas      gas      oil      electric oil      oil      gas
## [169] <NA>     gas      gas      oil      oil      oil      gas      gas
## [177] gas      <NA>     oil      gas      gas      gas      gas      gas
## [185] oil      gas      oil      oil      gas      gas      none     gas
## [193] oil      gas      gas      oil      gas      oil      oil      gas
## [201] oil      gas      gas      oil      gas      gas      gas      gas
## [209] gas      gas      gas      gas      oil      gas      gas      oil
## [217] oil      oil      gas      gas      electric oil      <NA>     gas
## [225] oil      gas      gas      electric other    gas      gas      gas
## [233] gas      gas      gas      none     oil      gas      gas      oil
## [241] gas      oil      gas      gas      gas      gas      oil      gas
## [249] oil      oil      gas      gas      oil      oil      oil      gas
## [257] oil      <NA>     gas      electric <NA>     gas      oil      gas
## [265] oil      oil      gas      gas      gas      oil      oil      oil
## [273] gas      oil      oil      gas      oil      oil      gas      oil
## [281] oil      gas      oil      gas      gas      oil      oil      gas
## [289] gas      oil      gas      oil      gas      gas      gas      gas
## [297] oil      oil      <NA>     oil      gas      oil      gas      oil
## [305] gas      oil      oil      oil      gas      oil      gas      gas
## [313] oil      oil      gas      oil      gas      gas      gas      oil
## [321] gas      oil      oil      electric oil      oil      gas      oil
## [329] oil      <NA>     gas      gas      oil      oil      oil      gas
## [337] gas      gas      oil      gas      <NA>     gas      gas      electric
## [345] gas      gas      gas      gas      gas      oil      gas      oil
## [353] gas      gas      gas      gas      gas      gas      gas      oil
## [361] gas      gas      gas      gas      electric oil      oil      oil
## [369] gas      <NA>     oil      gas      gas      gas      <NA>     oil
## [377] gas      gas      gas      gas      gas      gas      oil      gas
## [385] gas      gas      gas      gas      oil      oil      oil      oil
```

```
## [393] gas      oil      gas      gas      <NA>     oil      gas      gas
## [401] oil      oil      gas      gas      oil      gas      gas      gas
## [409] gas      oil      gas      gas      gas      oil      gas      oil
## [417] gas      oil      oil      oil      oil      gas      other    gas
## [425] oil      oil      oil      oil      oil      gas      gas      gas
## [433] other    gas      other    gas      gas      gas      oil      gas
## [441] gas      oil      gas      gas      oil      gas      none     oil
## [449] gas      gas      oil      gas      gas      gas      oil      gas
## [457] gas      gas      oil      <NA>     <NA>     <NA>     gas      gas
## [465] gas      gas      gas      gas      gas      gas      gas      oil
## [473] gas      gas      gas      gas      <NA>     gas      oil      oil
## [481] <NA>     gas      gas      gas      gas      electric gas      gas
## [489] gas      gas      gas      gas      gas      gas      gas      gas
## [497] gas      electric gas      other    gas      gas      gas      gas
## [505] oil      gas      <NA>     gas      gas      oil      oil      gas
## [513] oil      gas      gas      gas      oil      oil      <NA>     oil
## [521] oil      gas      gas      gas      gas      gas      electric other
## Levels: electric gas none oil other
```

```
skim(df_clean)
```

Table 7: Data summary

Name	df_clean
Number of rows	528
Number of columns	17
Column type frequency:	
factor	5
numeric	12
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
cats_allowed	0	1.00	FALSE	2	no: 285, yes: 243
coop_condo	0	1.00	FALSE	2	co-: 399, con: 129
dining_room_type	120	0.77	FALSE	4	com: 241, for: 116, oth: 49, din: 2
dogs_allowed	0	1.00	FALSE	2	no: 381, yes: 147
fuel_type	24	0.95	FALSE	5	gas: 301, oil: 180, ele: 11, oth: 9

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p
common_charges	396	0.25	433.92	205.40	70	288.50	390.5	537.
garage_exists	0	1.00	0.18	0.38	0	0.00	0.0	0.
maintenance_cost	142	0.73	821.85	378.77	155	639.25	734.0	880.
approx_year_built	6	0.99	1962.38	20.56	1915	1950.00	1957.0	1968.
community_district_num	1	1.00	26.30	2.99	3	25.00	26.0	28.
num_bedrooms	0	1.00	1.54	0.75	0	1.00	1.0	2.
num_floors_in_building	108	0.80	7.08	6.83	1	2.00	6.0	7.

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p
num_full_bathrooms	0	1.00	1.20	0.42	1	1.00	1.0	1.
num_total_rooms	0	1.00	4.02	1.20	1	3.00	4.0	5.
sq_footage	315	0.40	965.28	490.42	375	750.00	874.0	1010.
sale_price	0	1.00	314956.56	179526.60	55000	171500.00	259500.0	428875.
walk_score	0	1.00	83.10	13.09	15	76.00	85.0	94.

```
M = tbl_df(apply(is.na(df_clean), 2, as.numeric))
colnames(M) = paste("is_missing_", colnames(df_clean), sep = "")
M %<>%
  select_if(function(x){sum(x) > 0})
head(M)
```

```
## # A tibble: 6 x 8
##   is_missing_comm~ is_missing_dini~ is_missing_fuel~ is_missing_main~
##              <dbl>            <dbl>            <dbl>            <dbl>
## 1                0                0                0                1
## 2                1                0                0                0
## 3                0                0                1                1
## 4                0                0                0                1
## 5                1                0                0                0
## 6                1                0                0                0
## # ... with 4 more variables: is_missing_approx_year_built <dbl>,
## #   is_missing_community_district_num <dbl>,
## #   is_missing_num_floors_in_building <dbl>, is_missing_sq_footage <dbl>
```

```
pacman::p_load(missForest)
dfimp = missForest(data.frame(df_clean))$ximp
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
##   missForest iteration 5 in progress...done!
##   missForest iteration 6 in progress...done!
```

```
df_final = cbind(dfimp, M)
```

```
skim(df_final)
```

Table 10: Data summary

Name	df_final
Number of rows	528
Number of columns	25
Column type frequency:	
factor	5
numeric	20
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
cats_allowed	0	1	FALSE	2	no: 285, yes: 243
coop_condo	0	1	FALSE	2	co-: 399, con: 129
dining_room_type	0	1	FALSE	4	com: 330, for: 140, oth: 55, din: 3
dogs_allowed	0	1	FALSE	2	no: 381, yes: 147
fuel_type	0	1	FALSE	5	gas: 312, oil: 191, ele: 11, oth: 11

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	
common_charges	0	1	512.70	139.80	70	441.41	499
garage_exists	0	1	0.18	0.38	0	0.00	0
maintenance_cost	0	1	810.08	359.95	155	602.05	720
approx_year_built	0	1	1962.27	20.47	1915	1950.00	1956
community_district_num	0	1	26.30	2.98	3	25.00	26
num_bedrooms	0	1	1.54	0.75	0	1.00	1
num_floors_in_building	0	1	7.12	6.33	1	3.00	6
num_full_bathrooms	0	1	1.20	0.42	1	1.00	1
num_total_rooms	0	1	4.02	1.20	1	3.00	4
sq_footage	0	1	904.70	367.52	375	729.51	829
sale_price	0	1	314956.56	179526.60	55000	171500.00	259500
walk_score	0	1	83.10	13.09	15	76.00	85
is_missing_common_charges	0	1	0.75	0.43	0	0.75	1
is_missing_dining_room_type	0	1	0.23	0.42	0	0.00	0
is_missing_fuel_type	0	1	0.05	0.21	0	0.00	0
is_missing_maintenance_cost	0	1	0.27	0.44	0	0.00	0
is_missing_approx_year_built	0	1	0.01	0.11	0	0.00	0
is_missing_community_district_num	0	1	0.00	0.04	0	0.00	0
is_missing_num_floors_in_building	0	1	0.20	0.40	0	0.00	0
is_missing_sq_footage	0	1	0.60	0.49	0	0.00	1

```
# Tried to one hot incode data. MAJOR FAIL. R takes care of this since data is already factors
df_dummy = copy(df_final)
df_dummy$cats_allowed = model.matrix(~df_dummy$cats_allowed + 0)
df_dummy$coop_condo = model.matrix(~df_dummy$coop_condo + 0)
df_dummy$dining_room_type = model.matrix(~df_dummy$dining_room_type + 0)
df_dummy$dogs_allowed = model.matrix(~df_dummy$dogs_allowed + 0)
df_dummy$fuel_type = model.matrix(~df_dummy$fuel_type + 0)
library(data.table,mltools)
something = copy(df_final)

something$fuel_type = cbind(model.matrix(~something$fuel_type))
```

```
skim(df_dummy)
```

Table 13: Data summary

Name	df_dummy
Number of rows	528
Number of columns	25

Table 13: Data summary

Column type frequency: numeric	25
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	
cats_allowed	0	1	0.54	0.50	0	0.00	0
common_charges	0	1	512.70	139.80	70	441.41	499
coop_condo	0	1	0.76	0.43	0	0.00	0
dining_room_type	0	1	0.62	0.48	0	0.00	0
dogs_allowed	0	1	0.72	0.45	0	0.00	0
fuel_type	0	1	0.02	0.14	0	0.00	0
garage_exists	0	1	0.18	0.38	0	0.00	0
maintenance_cost	0	1	810.08	359.95	155	602.05	720
approx_year_built	0	1	1962.27	20.47	1915	1950.00	1956
community_district_num	0	1	26.30	2.98	3	25.00	26
num_bedrooms	0	1	1.54	0.75	0	1.00	1
num_floors_in_building	0	1	7.12	6.33	1	3.00	6
num_full_bathrooms	0	1	1.20	0.42	1	1.00	1
num_total_rooms	0	1	4.02	1.20	1	3.00	4
sq_footage	0	1	904.70	367.52	375	729.51	829
sale_price	0	1	314956.56	179526.60	55000	171500.00	259500
walk_score	0	1	83.10	13.09	15	76.00	85
is_missing_common_charges	0	1	0.75	0.43	0	0.75	1
is_missing_dining_room_type	0	1	0.23	0.42	0	0.00	0
is_missing_fuel_type	0	1	0.05	0.21	0	0.00	0
is_missing_maintenance_cost	0	1	0.27	0.44	0	0.00	0
is_missing_approx_year_built	0	1	0.01	0.11	0	0.00	0
is_missing_community_district_num	0	1	0.00	0.04	0	0.00	0
is_missing_num_floors_in_building	0	1	0.20	0.40	0	0.00	0
is_missing_sq_footage	0	1	0.60	0.49	0	0.00	1

```
set.seed(28)
train.control <- trainControl(method = "repeatedcv",
                              number = 10, repeats = 3)
prop_test = 0.10
test_indices = sample(1 : nrow(df_final), round((prop_test) * nrow(df_final)))
df_test = df_final[test_indices, ]
y_test = df_test$sale_price
X_test = cbind(1, df_test)
#X_test$sale_price = NULL


train_indices = setdiff(1 : nrow(df_final), test_indices)
df_train = df_final[train_indices, ]
y_train = df_train$sale_price
X_train = cbind(1, df_train)
```

```r
#X_train$sale_price = NULL

n_train = nrow(X_train)
```

```r
mod = train(sale_price ~ ., df_final, trControl = train.control,method = "lm")
#mod = lm(sale_price ~ ., df_final)
summary(mod)$r.squared
```

```
## [1] 0.7769604
```

```r
summary(mod)$sigma
```

```
## [1] 87218.86
```

```r
y_hat = predict(mod,data.frame(X_test))

e = y_test - y_hat
Rsq = (var(y_test) - var(e)) / var(y_test)
Rsq
```

```
## [1] 0.8458464
```

```r
mod =lm(sale_price ~., data.frame(X_train),set.seed(28))
summary(mod)$r.squared
```

```
## [1] 0.76963
```

```r
summary(mod)$sigma
```

```
## [1] 88226.96
```

```r
y_hat = predict(mod,data.frame(X_test))
e = y_test - y_hat
Rsq_oos = (var(y_test) - var(e)) / var(y_test)

cat("My R Squared in sample is ",summary(mod)$r.squared, "My RSME is:", summary(mod)$sigma)
```

```
## My R Squared in sample is  0.76963 My RSME is: 88226.96
```

```r
cat("\nMy R Squared out of sample is ",Rsq_oos, "My RSME is:", sd(e))
```

```
##
## My R Squared out of sample is  0.8122566 My RSME is: 83883.78
```

REGRESSION TREEES.

Here the trees overfit in sample but they did pretty decent out of sample but not better than OLS

```r
options(java.parameters = "-Xmx4000m")

X_train_CART = X_train
X_train_CART$sale_price = NULL

X_test_CART = X_test
X_test_CART$sale_price = NULL
tree_model = YARFCART(X_train_CART, y_train, bootstrap_indices = 1 : n_train,calculate_oob_error = TRUE
```

```
## YARF initializing with a fixed 1 trees...
```

```
## YARF factors created...
## YARF after data preprocessed... 35 total features...
## Beginning YARF regression model construction...done.
## Calculating OOB error...done.
```

```
#illustrate_trees(tree_model, max_depth = 4, open_file = TRUE)
get_tree_num_nodes_leaves_max_depths(tree_model)
```

```
## $num_nodes
## [1] 377
##
## $num_leaves
## [1] 189
##
## $max_depths
## [1] 18
```

```
#In Sample Error
y_hat_train = predict(tree_model,X_train)
```

```
## Warning in predict.YARF(tree_model, X_train): Prediction set column names did not match training set
## Attempting to subset to training set columns.
```

```
e_train = y_train - y_hat_train
rsme_train = sd(e_train)
rsquared_train = (var(y_train) - var(e_train)) / var(y_train)

#Out of Sample Error
y_hat_test = predict(tree_model, X_test)
```

```
## Warning in predict.YARF(tree_model, X_test): Prediction set column names did not match training set
## Attempting to subset to training set columns.
```

```
e_test = y_test - y_hat_test
rsme_test = sd(e_test)
rsquared_test = (var(y_test) - var(e_test)) / var(y_test)

cat("My R Squared in sample is ",rsquared_train, "My RSME is:", rsme_train)
```

```
## My R Squared in sample is  0.990009 My RSME is: 17802.63
```

```
cat("\nMy R Squared out of sample is ",rsquared_test, "My RSME is:",rsme_test)
```

```
##
## My R Squared out of sample is  0.818939 My RSME is: 82377.4
```

#RANDOM FOREST

```
set.seed(28)
X_train_RF = X_train
X_train_RF$sale_price = NULL

X_test_RF = X_test
X_test_RF$sale_price = NULL
RF_model = YARF(X_train_RF, y_train, num_trees = 500, seed = 28 )
```

```
## YARF initializing with a fixed 500 trees...
## YARF factors created...
## YARF after data preprocessed... 35 total features...
```

```
## Beginning YARF regression model construction...done.
## Calculating OOB error...done.
```

```
#In Sample Error
y_hat_train = predict(RF_model,X_train)
```

```
## Warning in predict.YARF(RF_model, X_train): Prediction set column names did not match training set c
## Attempting to subset to training set columns.
```

```
e_train = y_train - y_hat_train
rsme_train = sd(e_train)
rsquared_train = (var(y_train) - var(e_train)) / var(y_train)

#Out of Sample Error
y_hat_test = predict(RF_model, X_test)
```

```
## Warning in predict.YARF(RF_model, X_test): Prediction set column names did not match training set co
## Attempting to subset to training set columns.
```

```
e_test = y_test - y_hat_test
rsme_test = sd(e_test)
rsquared_test = (var(y_test) - var(e_test)) / var(y_test)

cat("My R Squared in sample is ",rsquared_train, "My RSME is:", rsme_train)
```

```
## My R Squared in sample is  0.9684723 My RSME is: 31624.61
```

```
cat("\nMy R Squared out of sample is ",rsquared_test, "My RSME is:",rsme_test)
```

```
##
## My R Squared out of sample is  0.8472357 My RSME is: 75667.01
```

```
# library(rpart)
# library(rpart.plot)
# fit = rpart(sale_price ~., data.frame(X_train),method="anova")
# rpart.plot(fit)
# summary(fit)
# pred
#
# in_e = y_train - pred
# sd(in_e)
# (var(y_train) - var(e)) / var(y_train)
# e = y_test - pred
# sd(e)
#
# Rsq_oos = (var(y_test) - var(e)) / var(y_test)
# #sd(e)
# cat("My R Squared in sample is ",summary(mod)$r.squared, "My RSME is:", sd(in_e))
# cat("\nMy R Squared out of sample is ",Rsq_oos, "My RSME is:", sd(e))
# ```
# ```{r}
# library(randomForest)
# control <- trainControl(method="cv", number=10)
#
#
# RegressionTree1 = train(sale_price~., data=data.frame(X_train), method="rpart", trControl=control)
# y_hat = predict(object = RegressionTree1,newdata = data.frame(X_test))
```

```
# sqrt(mean((y_hat-y_test)^2))
#
# RegressionTree = train(sale_price~., data=df_final, method="rpart", trControl=control)
# print(RegressionTree)
#
#
# ##
#
# fit = rpart(sale_price ~., data.frame(X_train),method = 'anova')
# printcp(fit)
# rpart.plot(fit)
# summary(fit)
# y_hat = predict(object = fit,newdata = data.frame(X_test))
# sqrt(mean((y_hat-y_test)^2))
```

```
# RandomForest = train(sale_price~., data=df_final, method="rf", trControl=control)
# print(RandomForest)
#
#
```