Predicting a Sale Price of a House

Using Multiple Linear Regression and Neutral Network to Predict the sale price of a house in Ames city

Alexis Mekueko

City University of New York-School of Professional Studies
Machine Learning and Big Data(DATA 622)

24 May 2022

To view all the files in this project, please go to My Github site.

Summary

Let’s say someone is interested in buying a house and don’t know how to prepare for it. Imagine this person who is looking for a dream home and don’t even know where to start. Buying a house start with knowing the sale price. If this person who is wondering about buying house and has no clue, then there is a problem. Buying a house is a lifetime experience for many first-time home buyers. It is an exciting moment, but it can be a bad one if things turn out unexpectedly. Thus, the question is: how can a home buyer know about the sale price of a house? Meaning, how to predict a sale price of a house? Well! Machine learning has the answer to this question. Machine learning algorithm is a computer algorithm that teaches itself a pattern based on data. In other words, machine learning builds a pattern that helps identify the future event occurring.

When we said machine learning is a computer algorithm to build a pattern for a sale price of a house that is not fully correct. It is not correct because there are many algorithms, and the pattern relies heavily on the data. If there is no data, then there is no price prediction. Therefore, data plays an important role in predicting the price of a something with the machine learning algorithm. The dataset used to predict the sale price of a house is from the Ames housing division in Iowa. The dataset is available at Kaggle.com. The data has about 1460 observations and 81 features with sale price (the target variable) included. These features describe just about everything that is part of a house. We are talking about the lot size, the year the house was built, the garage condition, fireplace conditions, etc. Although there are enough variables in this dataset, we still need to clean the data before fitting it to the model for analysis. The cleaning involved removing all unnecessary variables, the variables that have equivalent meaning with other variables or variables that have way more missing values by visual inspection. Finally, we used correlation technique to find which variable has strong influence on the target variable.

The purpose of cleaning the data is to ensure that we have the right data for the method used for the analysis. Let’s recall that data analysis is the process of collecting, modeling, and analyzing data to extract meaningful information that ultimately leads to decision-making. In other words, the leadership want to make decisions based on solid facts. Building these solid facts is where the methodology comes to play. There are several methods and techniques use in analyzing the data. The choice of the method and technique will depend on the aspect of the data (quantitative or qualitative). This Ames Housing data has quantitative aspect because it has finite number (discrete data) of values resulted from measurements collected on each house individually. For example, the number of bedrooms in the house, the number of car garage attached to the house and the size of the lot. Although some values are qualitative such as the condition of the garage (excellent, good, fair and none), it still refers to quantitative data. These data are taken as quantitative because this is not a survey where the responses are not uniformed. It is measurable data with standard in place. This type of measurements requires expertise that would apply the standards and unformalized for all houses. Cleaning the data is a step that precedes the analysis. We used multiple linear regression and Neutral Network method to analyze the data. The technique used is basically fitting the data to the model. The model here refers to the computerized formulas for each machine learning algorithm.

The purpose of performing the analysis is to predict the sale price of a house. Initially, we had a question about how one can predict the sale price of a house. Well! This is where the magic happens. The accuracy of predicting this price depends on the formulas. The choice of multiple linear regression is because while preparing the data for the analysis, we plotted few predictors against the target variable and saw some linear relationship. The second choice of neutral network is because we wanted to see how an unsupervised machine learning will perform against a supervised one. At the end, the verdict was tight because each algorithm performed well. Since the neutral network seems to be popular in neuron studies (complicated topic), we thought it would have easily outperformed the multiple linear regression. The assumption was not valid because this regression analysis showed better results with a low mean squared error value compared with the one from neutral network. In addition, based on the output values such as p-value, the chosen explanatory variables explained well the sale price of a house. Furthermore, we like the regression performance better because we can apply it into a business. Knowing a set of features along with the machine learning algorithm that can directly influence the sale price of a house, we can build an application that can tell customers the sale price of a house. We think this can be a starting point for many home buyers. There are other factors such as speculation (demand and supply) and inflation that influence the sale price of a house, but we believe the home buyers want to know this price first.

Problem Statement

How can we predict the sale price of a house?

Data Collection

Data source

We found some interesting dataset from data source: https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data

This data was originally generated the Ames Housing, a division within the Ames city in Iowa state. Thanks to Kaggle for making the data available free of use. The data is about a sale record of houses with all the defined criteria. Below is the description of the variable present in the dataset.

For a quicker look, we made the data (no modification) available at My Github site

Data Dictionary

MSSubClass..Identifies.the.type.of.dwelling.involved.in.the.sale. <chr>
20
30
40
45
50
60
70
75
80
85

1-10 of 460 rows | 1-1 of 2 columns

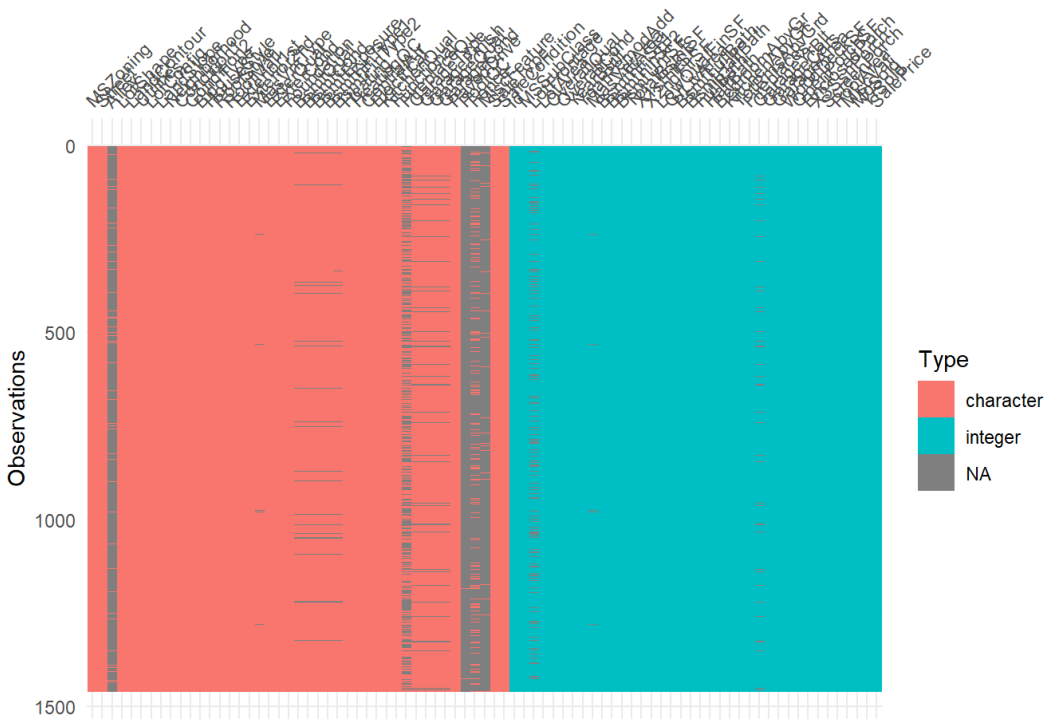
Previous 1 2 3 4 5 6 … 46 Next

Data Overview/Charateristic

	Id <int>	MSSubClass <int>	MSZoning <chr>	LotFrontage <int>	LotArea <int>	Street <chr>	Alley <chr>	LotShape <chr>	LandContour <chr>
1	1	60	RL	65	8450	Pave	*NA*	Reg	Lvl
2	2	20	RL	80	9600	Pave	*NA*	Reg	Lvl
3	3	60	RL	68	11250	Pave	*NA*	IR1	Lvl
4	4	70	RL	60	9550	Pave	*NA*	IR1	Lvl
5	5	60	RL	84	14260	Pave	*NA*	IR1	Lvl
6	6	50	RL	85	14115	Pave	*NA*	IR1	Lvl

6 rows | 1-10 of 82 columns

```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## Please use `gather()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```



This data is composed of 80 features for 1460 observations. The datatypes are mixed of integer and character. The target variable or the variable of interest is ‘SalePrice’. According to the dictionary, ‘SalePrice’ is the price at which a house was sold. Other variables are criteria used to derive the price of the house. We believe 81 variables for a price determination is excessive. Not all these variable are relevant/pertinent to the target variable. In addition, some variables have missing values. Therefore, we will need to reduce the number of feature by pertinence to SalePrice and cleaning the missing values.

Data Cleaning

Let’s see the missing values.

```
##
## Attaching package: 'skimr'
```

```
## The following object is masked from 'package:naniar':
##
##     n_complete
```

```
##
## Attaching package: 'lubridate'
```

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

```
## The following object is masked from 'package:tsibble':
##
##     interval
```

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

```
## Selecting by numeric.p100
```

n_missing <int>	complete_rate <dbl>	character.min <int>	character.max <int>	character.empty <int>	character.n_unique <int>
0	1.0000000	NA	NA	NA	NA
0	1.0000000	NA	NA	NA	NA
0	1.0000000	NA	NA	NA	NA
0	1.0000000	NA	NA	NA	NA
0	1.0000000	NA	NA	NA	NA
0	1.0000000	NA	NA	NA	NA
0	1.0000000	NA	NA	NA	NA
0	1.0000000	NA	NA	NA	NA
0	1.0000000	NA	NA	NA	NA
81	0.9445205	NA	NA	NA	NA

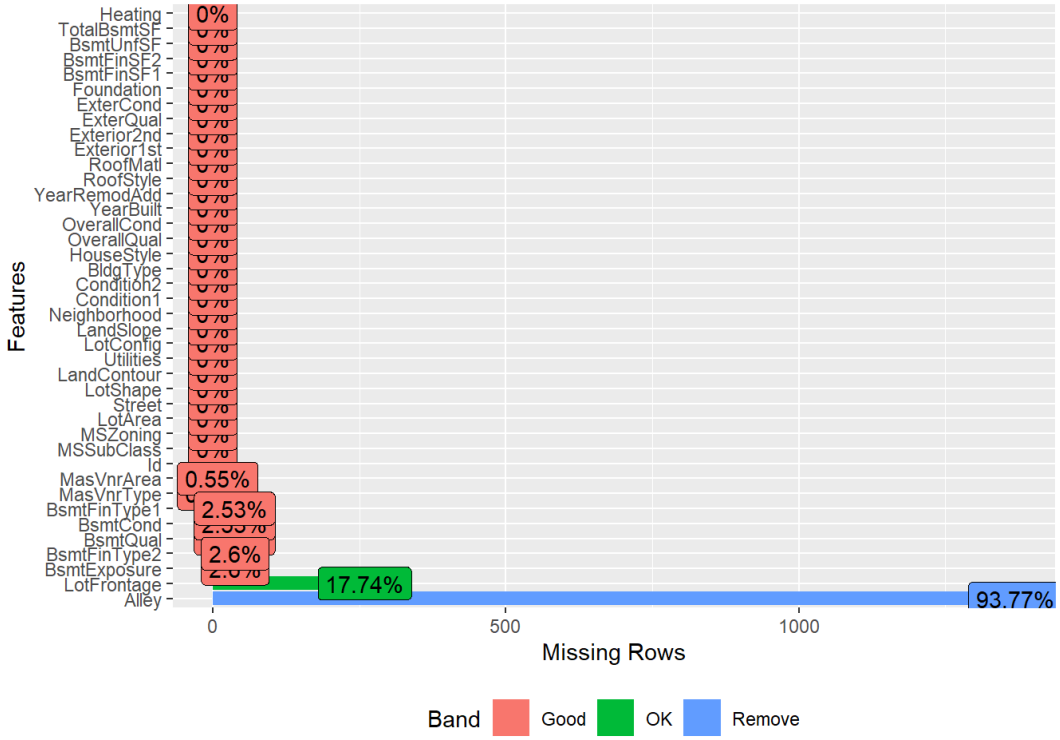
1-10 of 13 rows | 1-6 of 14 columns Previous 1 2 Next

There are 43 character(categorical) variables and 38 numerical variables. We know there are 1460 observations, so seeing the number (or percent) missing values per variable can give us an idea of the cleaning approach.

	x
Id	0
MSSubClass	0
MSZoning	0

	x
LotFrontage	259
LotArea	0
Street	0
Alley	1369
LotShape	0
LandContour	0
Utilities	0
LotConfig	0
LandSlope	0
Neighborhood	0
Condition1	0
Condition2	0
BldgType	0
HouseStyle	0
OverallQual	0
OverallCond	0
YearBuilt	0
YearRemodAdd	0
RoofStyle	0
RoofMatl	0
Exterior1st	0
Exterior2nd	0
MasVnrType	8
MasVnrArea	8
ExterQual	0
ExterCond	0

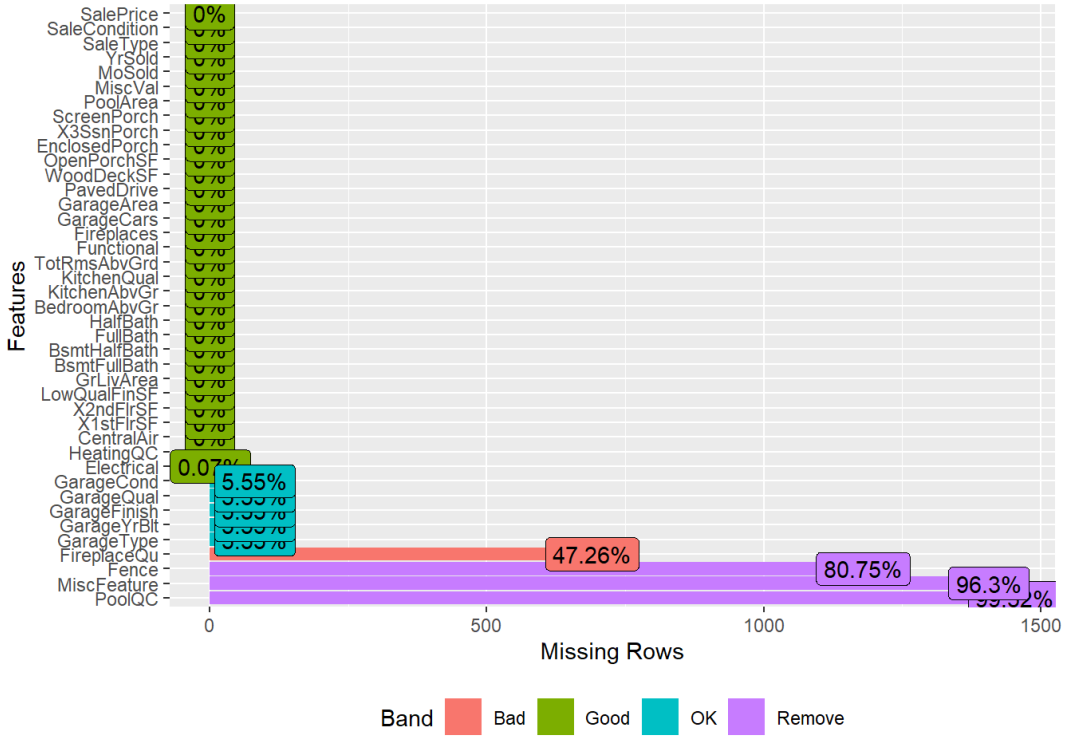
	x
Foundation	0
BsmtQual	37
BsmtCond	37
BsmtExposure	38
BsmtFinType1	37
BsmtFinSF1	0
BsmtFinType2	38
BsmtFinSF2	0
BsmtUnfSF	0
TotalBsmtSF	0
Heating	0



We selected half of the variables. Let’s see the other half.

	x
HeatingQC	0
CentralAir	0
Electrical	1
X1stFlrSF	0
X2ndFlrSF	0
LowQualFinSF	0
GrLivArea	0
BsmtFullBath	0
BsmtHalfBath	0
FullBath	0
HalfBath	0
BedroomAbvGr	0
KitchenAbvGr	0
KitchenQual	0
TotRmsAbvGrd	0
Functional	0
Fireplaces	0
FireplaceQu	690
GarageType	81
GarageYrBlt	81
GarageFinish	81
GarageCars	0
GarageArea	0
GarageQual	81
GarageCond	81

	x
PavedDrive	0
WoodDeckSF	0
OpenPorchSF	0
EnclosedPorch	0
X3SsnPorch	0
ScreenPorch	0
PoolArea	0
PoolQC	1453
Fence	1179
MiscFeature	1406
MiscVal	0
MoSold	0
YrSold	0
SaleType	0
SaleCondition	0
SalePrice	0



By visual inspection, we can remove these variables: PoolQC, Fence, Alley, MiscFeature, FireplaceQu, LotFrontage. because missing too many value…it probably means these variables are not important … houses that have these variables with values are probably extra..extra…or too special…

LotFrontage = LotFrontage: Linear feet of street connected to property. We want to remove this variable due to the 17.4% missing values and the fact that every house is built under the city regulation.

We will remove variable 'Id'. There is no need for it or because it is just an indication for record purpose.

	x
LotFrontage	259
Alley	1369
MasVnrType	8
MasVnrArea	8
BsmtQual	37
BsmtCond	37
BsmtExposure	38
BsmtFinType1	37

	x
BsmtFinType2	38
Electrical	1
FireplaceQu	690
GarageType	81
GarageYrBlt	81
GarageFinish	81
GarageQual	81
GarageCond	81
PoolQC	1453
Fence	1179
MiscFeature	1406

Above are variables with missing values greater than 0. I like this already!

This is where the benefit of having friends who are in housing for sale or construction count.

Next, we will evaluate (does it really matter? can it be done easily? does it cost a lot ?) all other variables with missing values

GarageFinish (81 missing values): Interior finish of the garage…average homebuyers don't care about it… the condition of the garage is sufficient … remove

GarageQual (81 missing values): Garage quality, this might mean the quality of the materials used to build the garage. Remove…

GarageCond (81 missing values): Garage condition, this might mean the condition(look) at the time the house is sold.

GarageType(81 missing values): garage location……garage condition is enough ….remove

GarageYrBlt(81 missing values): garage year built…same with yearbuilt…remove

MasVnrType(07 missing values): Masonry veneer type…the type of material used to construct the house(bricks, stone…)…we can keep it for now

MasVnrArea(07 missing values): Masonry veneer area in square feet…this is the kind of work homebuyers won't do. Because it involve doing the math to find the ratio of the veneer area covered Vs. not covered….remove

BsmtQual(037 missing values): Evaluates the height of the basement…will be removed because another variable described the same thing.

BsmtCond (37 missing values): Evaluates the general condition of the basement….we can keep…

BsmtExposure(38 missing values): Refers to walkout or garden level walls..average homebuyers don't care about it. remove

BsmtFinType1 (37 missing values): Rating of basement finished area…described already…remove

BsmtFinType2 (38 missing values): Type 1 finished square feet…no tricky math for buyer… remove

Electrical…we will only remove the 01 missing values…no hurts

```
##  [1] "MSSubClass"    "MSZoning"      "LotArea"       "Street"
##  [5] "LotShape"      "LandContour"   "Utilities"     "LotConfig"
##  [9] "LandSlope"     "Neighborhood"  "Condition1"    "Condition2"
## [13] "BldgType"      "HouseStyle"    "OverallQual"   "OverallCond"
## [17] "YearBuilt"     "YearRemodAdd"  "RoofStyle"     "RoofMatl"
## [21] "Exterior1st"   "Exterior2nd"   "MasVnrType"    "ExterQual"
## [25] "ExterCond"     "Foundation"    "BsmtCond"      "BsmtFinSF1"
## [29] "BsmtFinSF2"    "BsmtUnfSF"     "TotalBsmtSF"   "Heating"
## [33] "HeatingQC"     "CentralAir"    "Electrical"    "X1stFlrSF"
## [37] "X2ndFlrSF"     "LowQualFinSF"  "GrLivArea"     "BsmtFullBath"
## [41] "BsmtHalfBath"  "FullBath"      "HalfBath"      "BedroomAbvGr"
## [45] "KitchenAbvGr"  "KitchenQual"   "TotRmsAbvGrd"  "Functional"
## [49] "Fireplaces"    "GarageCars"    "GarageArea"    "GarageCond"
## [53] "PavedDrive"    "WoodDeckSF"    "OpenPorchSF"   "EnclosedPorch"
## [57] "X3SsnPorch"    "ScreenPorch"   "PoolArea"      "MiscVal"
## [61] "MoSold"        "YrSold"        "SaleType"      "SaleCondition"
## [65] "SalePrice"
```

Above list is the name of the remaining variables after we removed the first set as described early. We continue to remove by pertinence (if it seems like extra or one can live without it in the area…) to the sale price.

LandContour,LotShape, Street, LotConfig, LandSlope, Condition2, RoofStyle, Exterior2nd, OverallQual, Foundation, TotalBsmtSF,BsmtUnfSF, Heating, Electrical, X1stFlrSF, X2ndFlrSF, LowQualFinSF, BsmtFullBath, BsmtHalfBath, TotRmsAbvGrd, PavedDrive, EnclosedPorch, 3SsnPorch, ScreenPorch, MoSold,MSSubClass,Condition1

```
##  [1] "MSZoning"      "LotArea"       "Utilities"     "Neighborhood"
##  [5] "BldgType"      "HouseStyle"    "OverallCond"   "YearBuilt"
##  [9] "YearRemodAdd"  "RoofMatl"      "Exterior1st"   "MasVnrType"
## [13] "ExterQual"     "ExterCond"     "BsmtCond"      "BsmtFinSF1"
## [17] "BsmtFinSF2"    "HeatingQC"     "CentralAir"    "X1stFlrSF"
## [21] "GrLivArea"     "FullBath"      "HalfBath"      "BedroomAbvGr"
## [25] "KitchenAbvGr"  "KitchenQual"   "Functional"    "Fireplaces"
## [29] "GarageCars"    "GarageArea"    "GarageCond"    "WoodDeckSF"
## [33] "OpenPorchSF"   "PoolArea"      "MiscVal"       "YrSold"
## [37] "SaleType"      "SaleCondition" "SalePrice"
```

```
## MasVnrType   BsmtCond GarageCond
##          8         37         81
```

We significantly reduced the number of variable by nearly half. No need to do imputation by mean or other numerical imputations. We just need to replace the value according to the definition of the variable and remove the row where ‘NA’ has no meaning and it is very low missing values (ex: 10 missing values will no infere on 1460 observations)

```
##  [1] "MSZoning"      "LotArea"       "Utilities"     "Neighborhood"
##  [5] "BldgType"      "HouseStyle"    "OverallCond"   "YearBuilt"
##  [9] "YearRemodAdd"  "RoofMatl"      "Exterior1st"   "MasVnrType"
## [13] "ExterQual"     "ExterCond"     "BsmtCond"      "BsmtFinSF1"
## [17] "BsmtFinSF2"    "HeatingQC"     "CentralAir"    "X1stFlrSF"
## [21] "GrLivArea"     "FullBath"      "HalfBath"      "BedroomAbvGr"
## [25] "KitchenAbvGr"  "KitchenQual"   "Functional"    "Fireplaces"
## [29] "GarageCars"    "GarageArea"    "GarageCond"    "WoodDeckSF"
## [33] "OpenPorchSF"   "PoolArea"      "MiscVal"       "YrSold"
## [37] "SaleType"      "SaleCondition" "SalePrice"
```

Remained 36 variables after cleaning… let's convert character variables to categorical ones.

Exploratory Data Analysis (EDA)

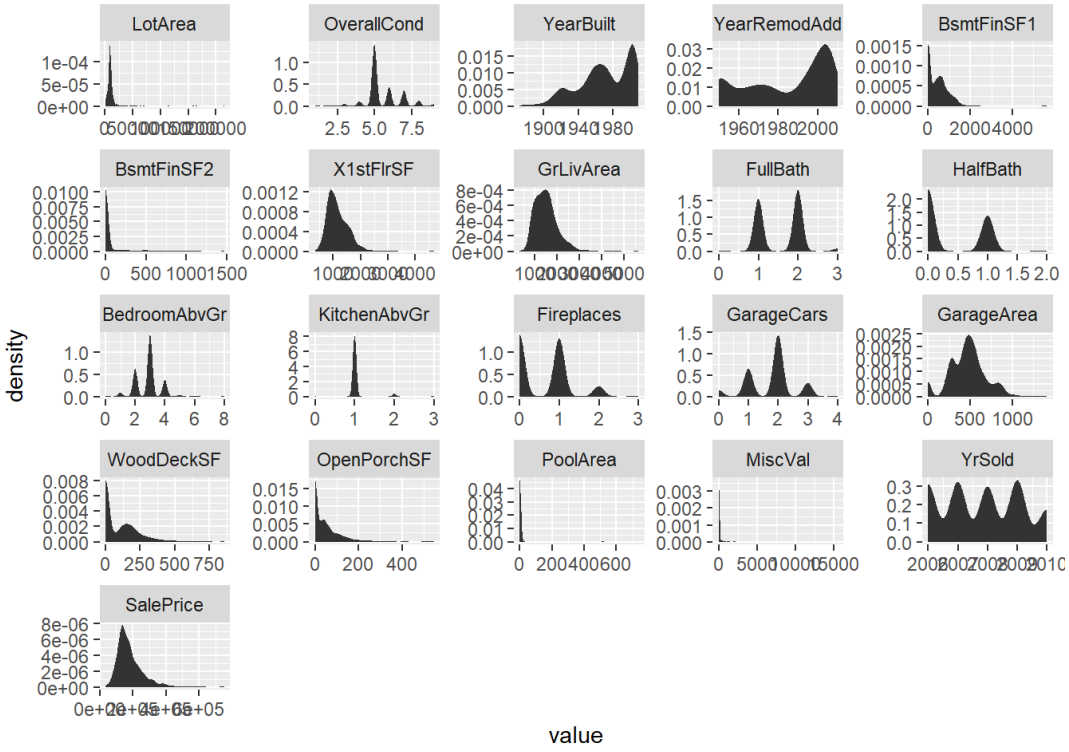
```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
MSZoning*	1	1452	4.033058e+00	6.269907e-01	4.0	4.062823e+00	0.0000	1	5	4	-1.7341136	6.4223786	0.0164542
LotArea	2	1452	1.050728e+04	9.989564e+03	9478.5	9.560115e+03	2962.2348	1300	215245	213945	12.2147550	202.7382100	262.1580559
Utilities*	3	1452	1.000689e+00	2.624320e-02	1.0	1.000000e+00	0.0000	1	2	1	38.0264243	1445.0041308	0.0006887
Neighborhood*	4	1452	1.313912e+01	5.889512e+00	13.0	1.309380e+01	7.4130	1	25	24	0.0198860	-1.0506232	0.1545596
BldgType*	5	1452	1.493113e+00	1.197554e+00	1.0	1.144578e+00	0.0000	1	5	4	2.2406039	3.4060800	0.0314276
HouseStyle*	6	1452	4.037879e+00	1.913519e+00	3.0	4.028399e+00	1.4826	1	8	7	0.3058160	-0.9646592	0.0502169
OverallCond	7	1452	5.579201e+00	1.113136e+00	5.0	5.481067e+00	0.0000	1	9	8	0.6934936	1.0807585	0.0292122
YearBuilt	8	1452	1.971116e+03	3.019376e+01	1972.0	1.973943e+03	37.0650	1872	2010	138	-0.6076580	-0.4482779	0.7923807
YearRemodAdd	9	1452	1.984775e+03	2.065247e+01	1993.0	1.986256e+03	19.2738	1950	2010	60	-0.4962541	-1.2807746	0.5419867
RoofMatl*	10	1452	2.075758e+00	6.007500e-01	2.0	2.000000e+00	0.0000	1	8	7	8.0694933	65.8874243	0.0157656
Exterior1st*	11	1452	1.062052e+01	3.197594e+00	13.0	1.092427e+01	1.4826	1	15	14	-0.7230384	-0.3648052	0.0839151
MasVnrType*	12	1452	2.761019e+00	6.157107e-01	3.0	2.728916e+00	0.0000	1	4	3	-0.0673056	-0.1344926	0.0161582
ExterQual*	13	1452	3.544077e+00	6.916750e-01	4.0	3.654905e+00	0.0000	1	4	3	-1.8375828	3.9126787	0.0181518

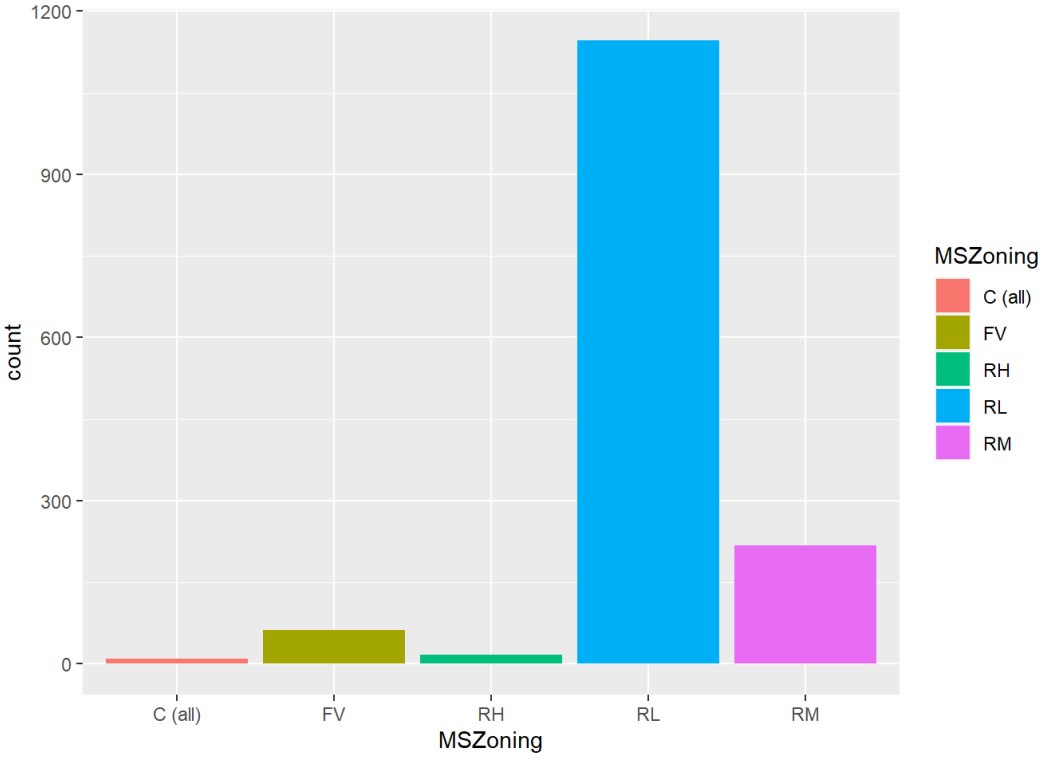
	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
ExterCond*	14	1452	4.732094e+00	7.335536e-01	5.0	4.944062e+00	0.0000	1	5	4	-2.5500640	5.2369125	0.0192508
BsmtCond*	15	1452	4.691460e+00	9.493638e-01	5.0	4.996558e+00	0.0000	1	5	4	-2.9401349	7.1989685	0.0249143
BsmtFinSF1	16	1452	4.419704e+02	4.553603e+02	381.0	3.842900e+02	564.8706	0	5644	5644	1.6993684	11.2275285	11.9501082
BsmtFinSF2	17	1452	4.680579e+01	1.617262e+02	0.0	1.492255e+00	0.0000	0	1474	1474	4.2331421	19.8767340	4.2442131
HeatingQC*	18	1452	2.544077e+00	1.739828e+00	1.0	2.430293e+00	0.0000	1	5	4	0.4761525	-1.5173517	0.0456586
CentralAir*	19	1452	1.934573e+00	2.473630e-01	2.0	2.000000e+00	0.0000	1	2	1	-3.5112263	10.3358300	0.0064916
X1stFlrSF	20	1452	1.161271e+03	3.850184e+02	1086.0	1.129282e+03	347.6697	334	4692	4358	1.3705592	5.7935362	10.1041132
GrLivArea	21	1452	1.514092e+03	5.256278e+02	1461.5	1.466049e+03	483.3276	334	5642	5308	1.3715369	4.9011921	13.7941514
FullBath	22	1452	1.562672e+00	5.502313e-01	2.0	1.558520e+00	0.0000	0	3	3	0.0363711	-0.8723593	0.0144398
HalfBath	23	1452	3.815427e-01	5.026637e-01	0.0	3.416523e-01	0.0000	0	2	2	0.6816203	-1.0673211	0.0131915
BedroomAbvGr	24	1452	2.867080e+00	8.148122e-01	3.0	2.852840e+00	0.0000	0	8	8	0.2171317	2.2390257	0.0213833
KitchenAbvGr	25	1452	1.046143e+00	2.194982e-01	1.0	1.000000e+00	0.0000	0	3	3	4.5052679	21.7240175	0.0057603
KitchenQual*	26	1452	3.342287e+00	8.297051e-01	4.0	3.506885e+00	0.0000	1	4	3	-1.4239365	1.7256217	0.0217741
Functional*	27	1452	6.752066e+00	9.705320e-01	7.0	7.000000e+00	0.0000	1	7	6	-4.0811771	16.4229147	0.0254699
Fireplaces	28	1452	6.122590e-01	6.434218e-01	1.0	5.344234e-01	1.4826	0	3	3	0.6473945	-0.2166199	0.0168854
GarageCars	29	1452	1.765151e+00	7.484957e-01	2.0	1.771945e+00	0.0000	0	4	4	-0.3374301	0.2003861	0.0196429
GarageArea	30	1452	4.724752e+02	2.141064e+02	478.0	4.692849e+02	174.9468	0	1418	1418	0.1829216	0.8988780	5.6188357
GarageCond*	31	1452	5.761708e+00	8.046860e-01	6.0	6.000000e+00	0.0000	1	6	5	-3.5643891	12.3529271	0.0211175
WoodDeckSF	32	1452	9.441667e+01	1.253937e+02	0.0	7.194234e+01	0.0000	0	857	857	1.5391213	2.9784831	3.2907314
OpenPorchSF	33	1452	4.639050e+01	6.604862e+01	24.0	3.301291e+01	35.5824	0	547	547	2.3807985	8.6198899	1.7333267
PoolArea	34	1452	2.774105e+00	4.028739e+01	0.0	0.000000e+00	0.0000	0	738	738	14.7566829	220.9454623	1.0572698
MiscVal	35	1452	4.372865e+01	4.974783e+02	0.0	0.000000e+00	0.0000	0	15500	15500	24.3594783	693.8077959	13.0554190
YrSold	36	1452	2.007815e+03	1.328927e+00	2008.0	2.007769e+03	1.4826	2006	2010	4	0.0956796	-1.1956650	0.0348753
SaleType*	37	1452	8.511019e+00	1.563458e+00	9.0	8.925129e+00	0.0000	1	9	8	-3.8350611	14.5699843	0.0410301
SaleCondition*	38	1452	4.768595e+00	1.101421e+00	5.0	5.000000e+00	0.0000	1	6	5	-2.7410913	6.8301526	0.0289048
SalePrice	39	1452	1.806151e+05	7.928554e+04	162700.0	1.705286e+05	55894.0200	34900	755000	720100	1.8801536	6.5401878	2080.7058509

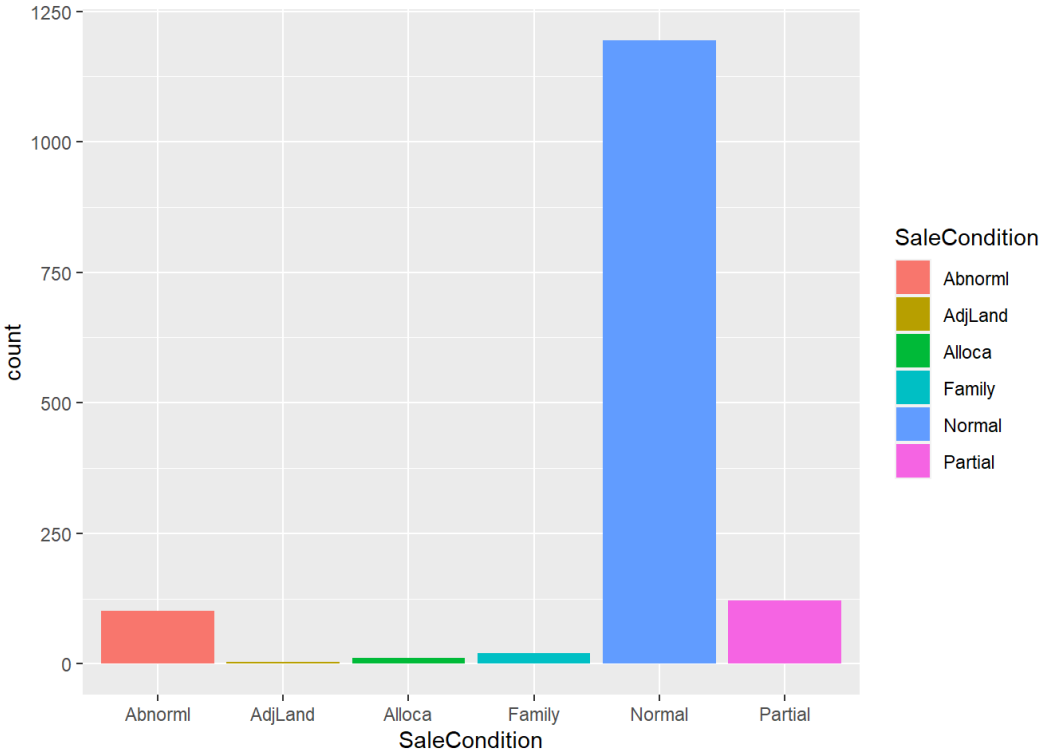
```
## No id variables; using all as measure variables
```

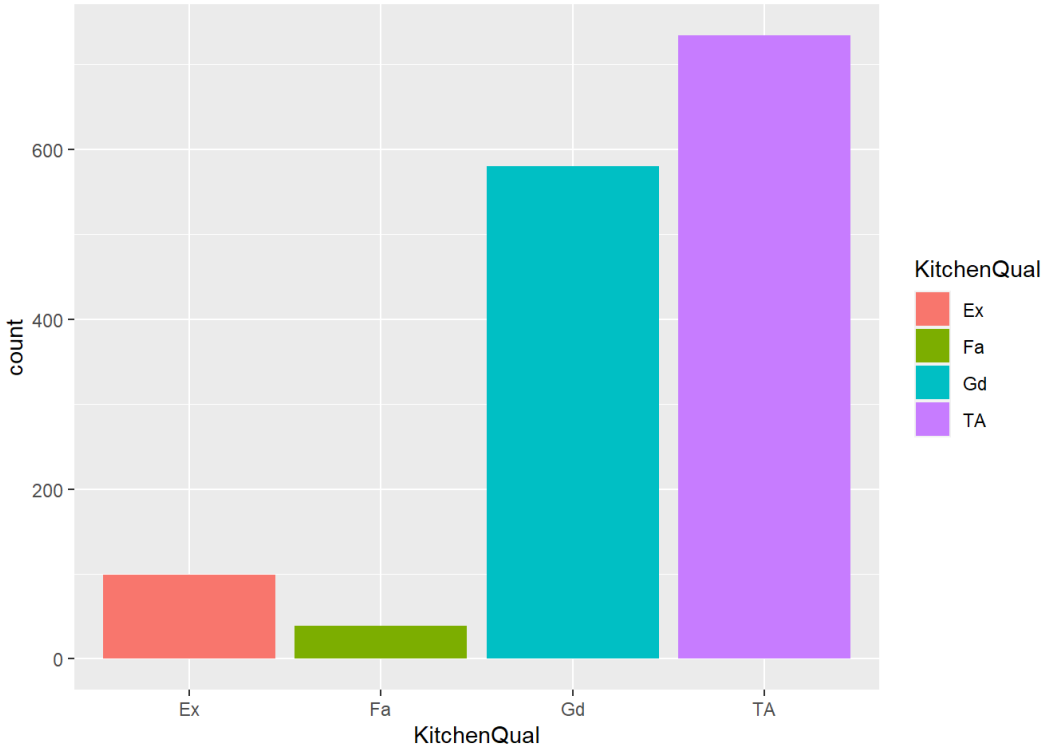


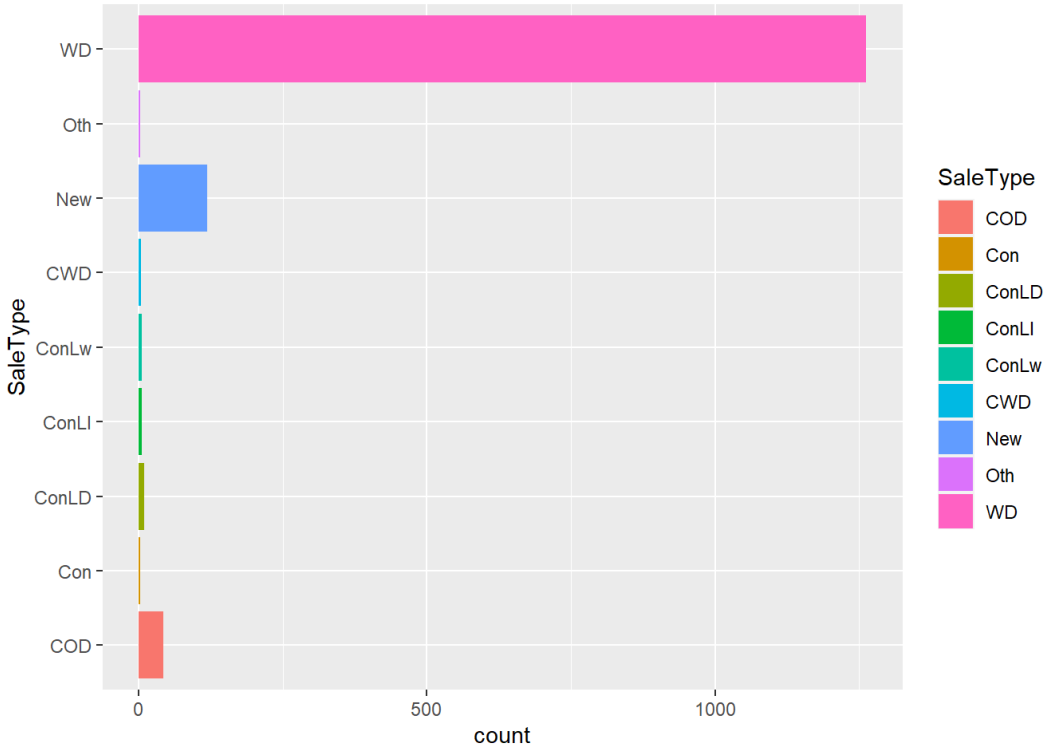
Above is the distribution of numerical variables…yearsold looks like many years counted. in fact, those variables with multiple mode, mean the variable have a set of of values that repeat. Hard to see off values (outliers).

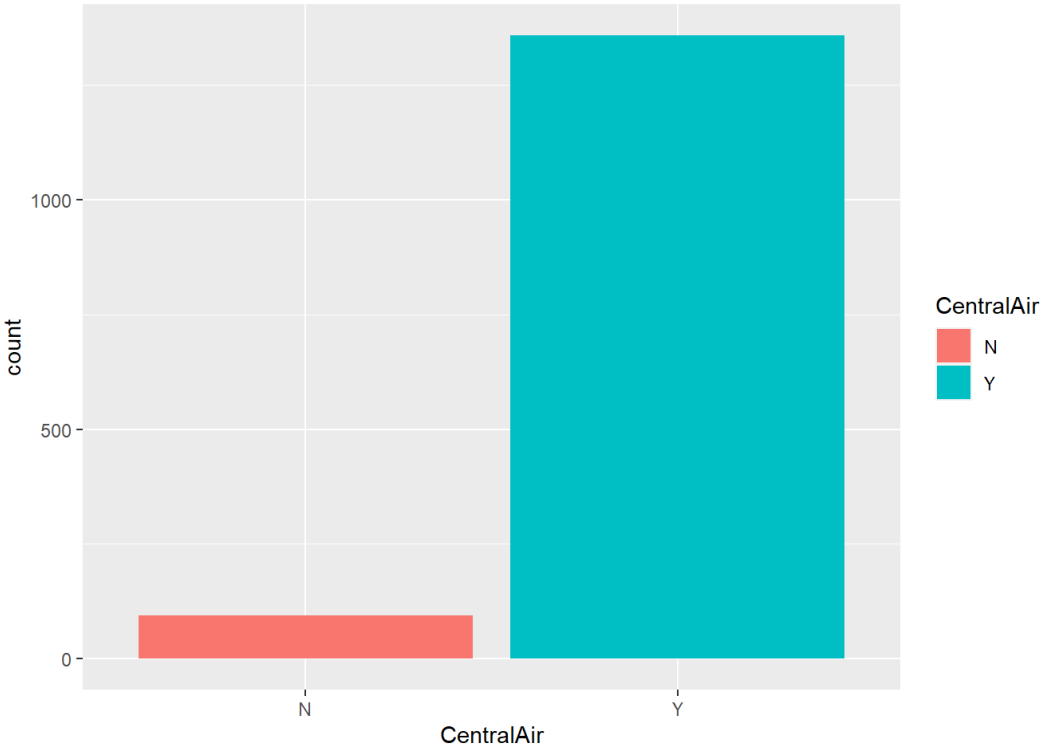
below is the distribution of categorical variables, not all them will be plot because some of them have too many levels to be plotted all at once…maybe there is another technique

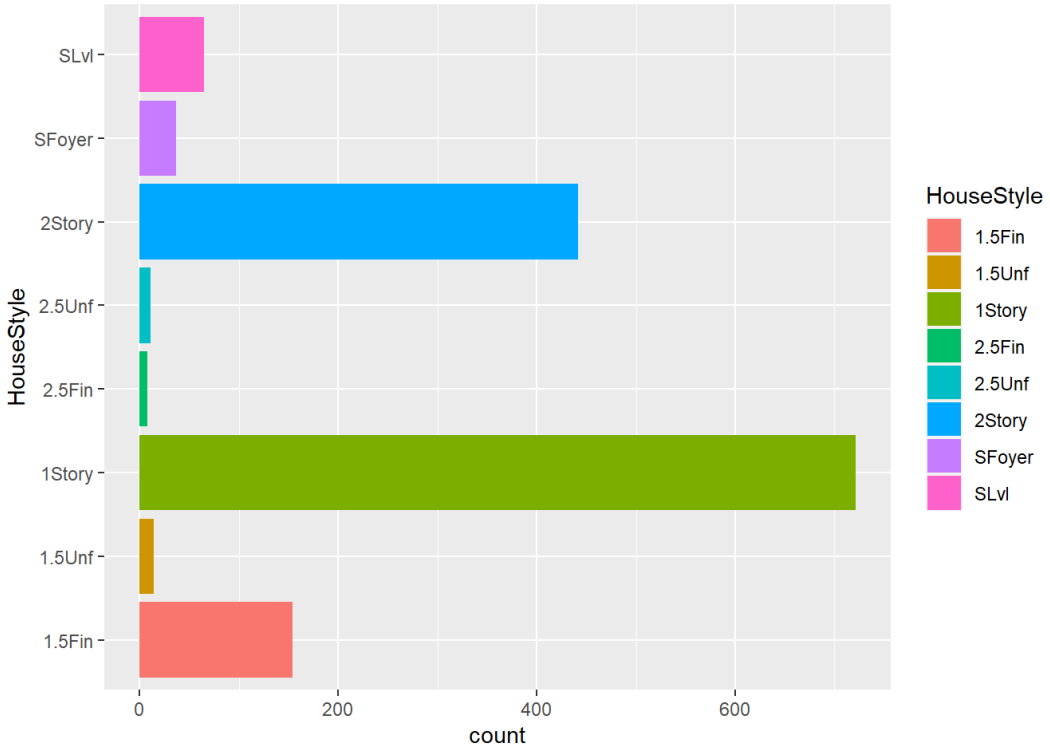

1200
900
600
300
0
count
C (all)
FV
RH
RL
RM
MSZoning
MSZoning
C (all)
FV
RH
RL
RM

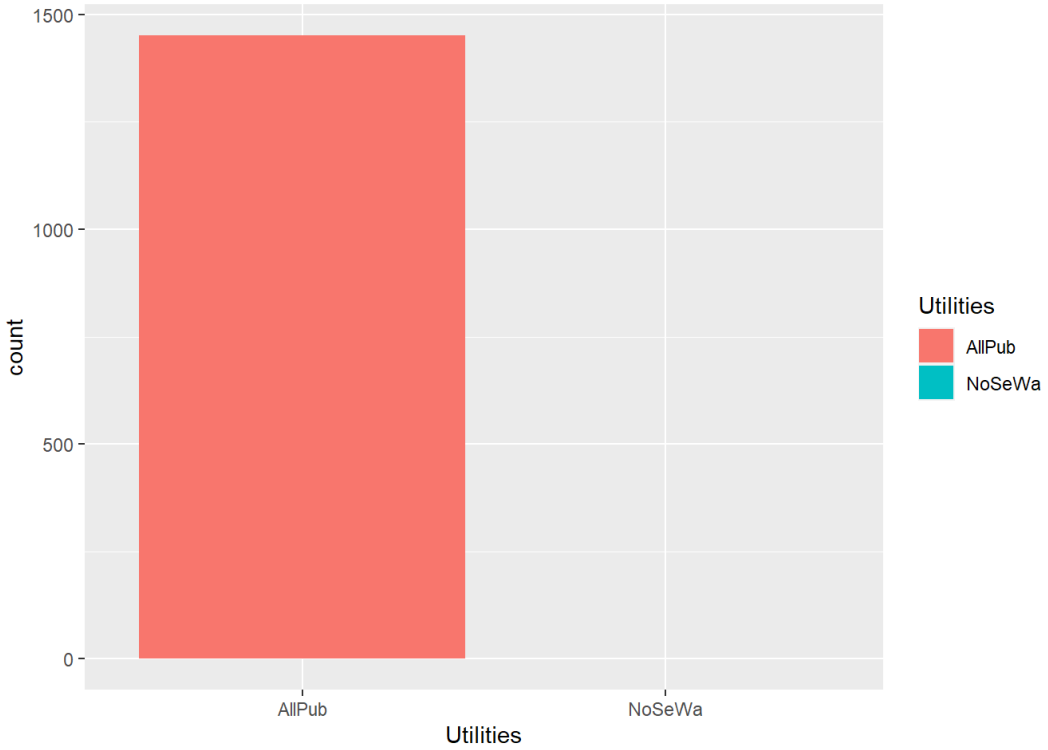

1250
1000
750
500
250
0
count
Abnorml
AdjLand
Alloca
Family
Normal
Partial
SaleCondition
SaleCondition
Abnorml
AdjLand
Alloca
Family
Normal
Partial

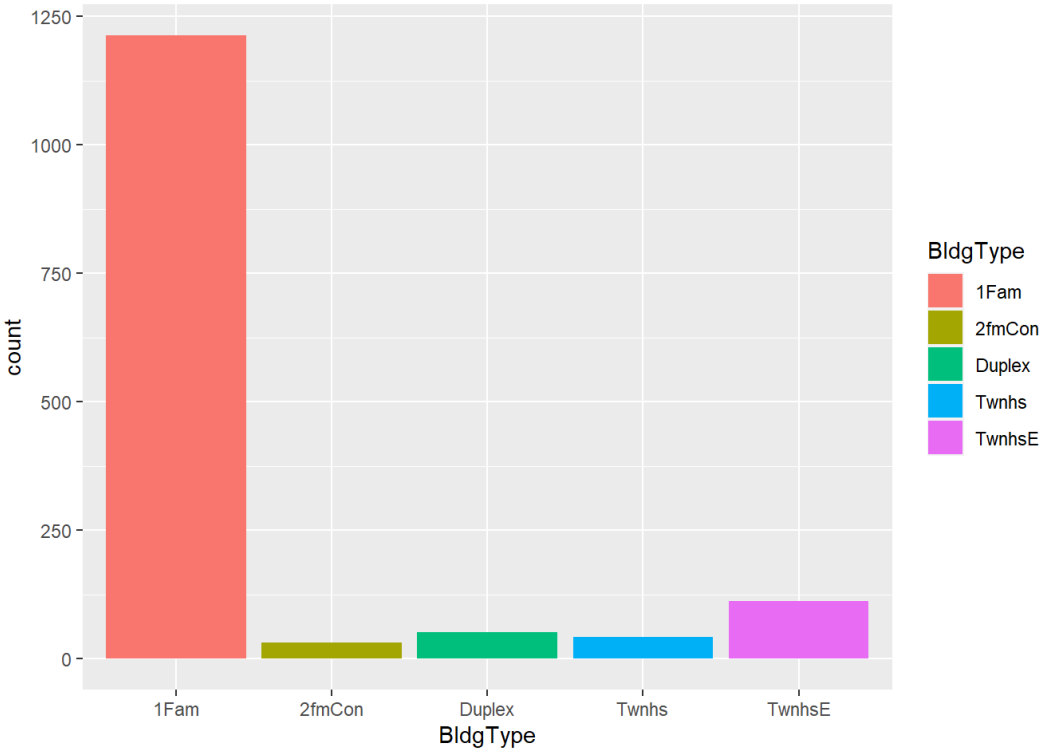

600
400
200
0
count
Ex
Fa
Gd
TA
KitchenQual
KitchenQual
Ex
Fa
Gd
TA


WD
Oth
New
CWD
ConLw
ConLI
ConLD
Con
COD
SaleType
0
500
1000
count
SaleType
COD
Con
ConLD
ConLI
ConLw
CWD
New
Oth
WD


1000
500
0
count
N
Y
CentralAir
CentralAir
N
Y


SLvl
SFoyer
2Story
2.5Unf
2.5Fin
1Story
1.5Unf
1.5Fin
HouseStyle
0
200
400
600
count
HouseStyle
1.5Fin
1.5Unf
1Story
2.5Fin
2.5Unf
2Story
SFoyer
SLvl


1500
1000
500
0
count
AllPub
NoSeWa
Utilities
Utilities
AllPub
NoSeWa




```
##  [1] "MSZoning"      "LotArea"       "Utilities"     "Neighborhood"
##  [5] "BldgType"      "HouseStyle"    "OverallCond"   "YearBuilt"
##  [9] "YearRemodAdd"  "RoofMatl"      "Exterior1st"   "MasVnrType"
## [13] "ExterQual"     "ExterCond"     "BsmtCond"      "BsmtFinSF1"
## [17] "BsmtFinSF2"    "HeatingQC"     "CentralAir"    "X1stFlrSF"
## [21] "GrLivArea"     "FullBath"      "HalfBath"      "BedroomAbvGr"
## [25] "KitchenAbvGr"  "KitchenQual"   "Functional"    "Fireplaces"
## [29] "GarageCars"    "GarageArea"    "GarageCond"    "WoodDeckSF"
## [33] "OpenPorchSF"   "PoolArea"      "MiscVal"       "YrSold"
## [37] "SaleType"      "SaleCondition" "SalePrice"
```

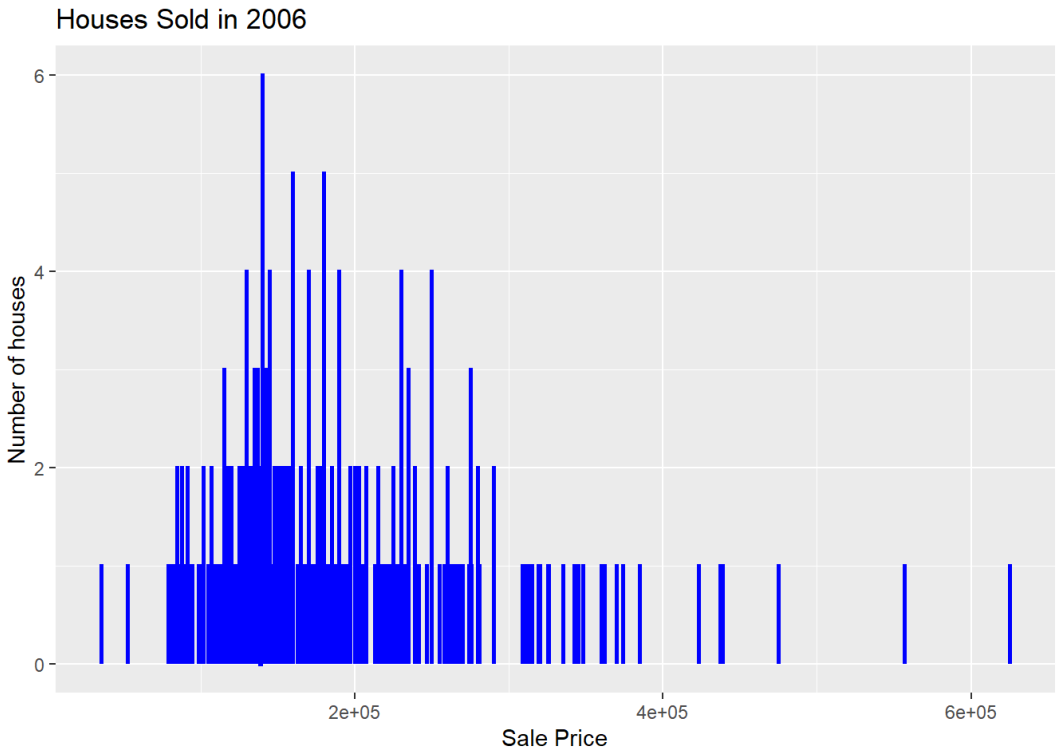
The way we visualize these categorical variables is that some of them can be redefined numerical since the categorical level in some of them are null or close to null. Potentially use some rating… 0 means nothing 1 means poor and so on with …2, 3, 4, 5..

Of course this is not the only way to transform them…

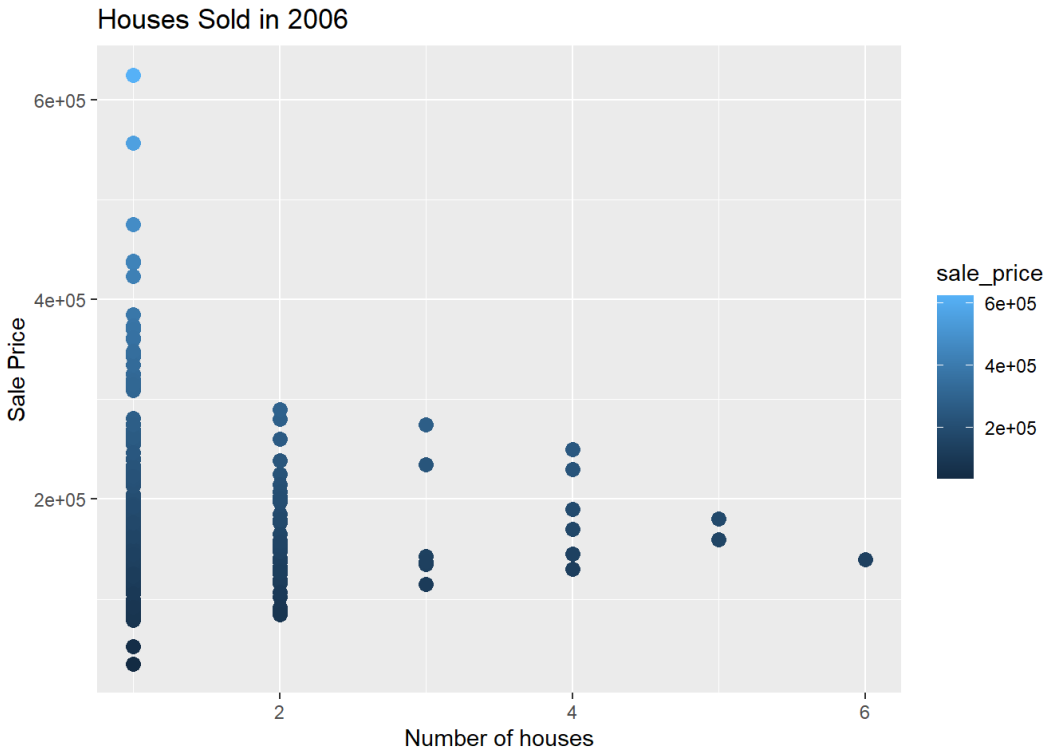
Feature Engineering

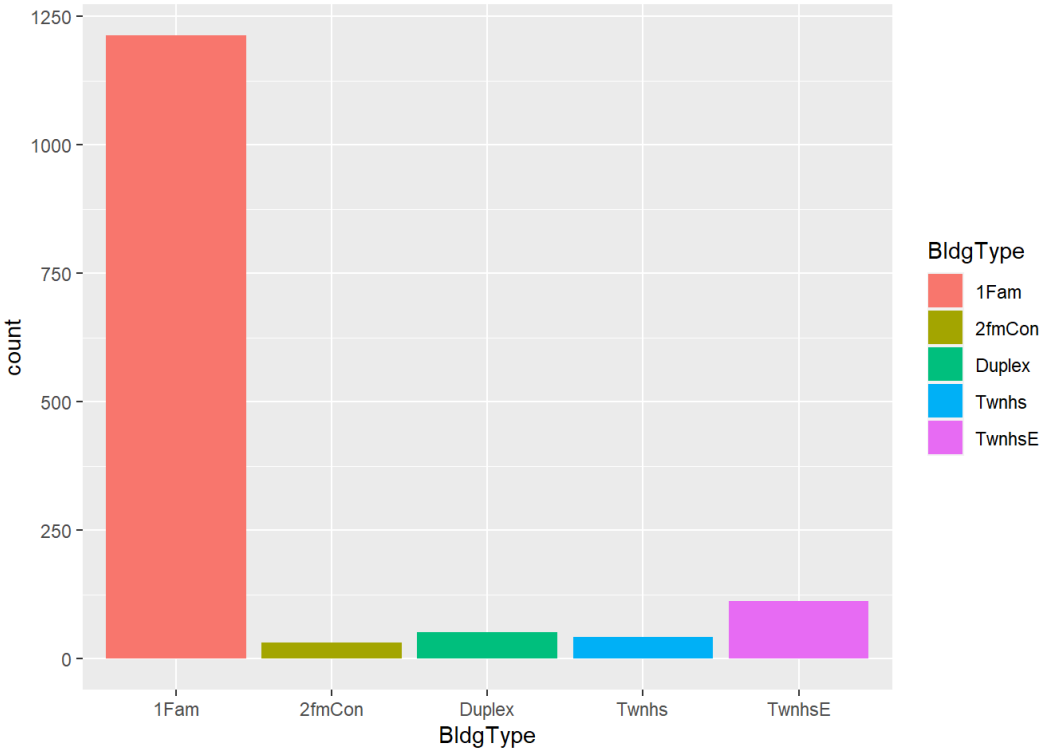
Let's see the range in the variables years…

```
## The earliest house was sold in  2006
```



Another way of seeing the houses sold in 2006…


Houses Sold in 2006
Sale Price
6e+05
4e+05
2e+05
2
4
6
Number of houses
sale_price
6e+05
4e+05
2e+05



let's see some values

```
##
##
## The latest house was sold in  2010
```

```
##
##
## The earliest house was built in  1872
```

```
##
##
## The latest house was built in  2010
```

```
##
##
## The house that was the first to be rebuilt was in  1950
```

```
##
##
## The house that was latest to be rebuilt was in  2010
```

Let's see the distribution at which house were sold.

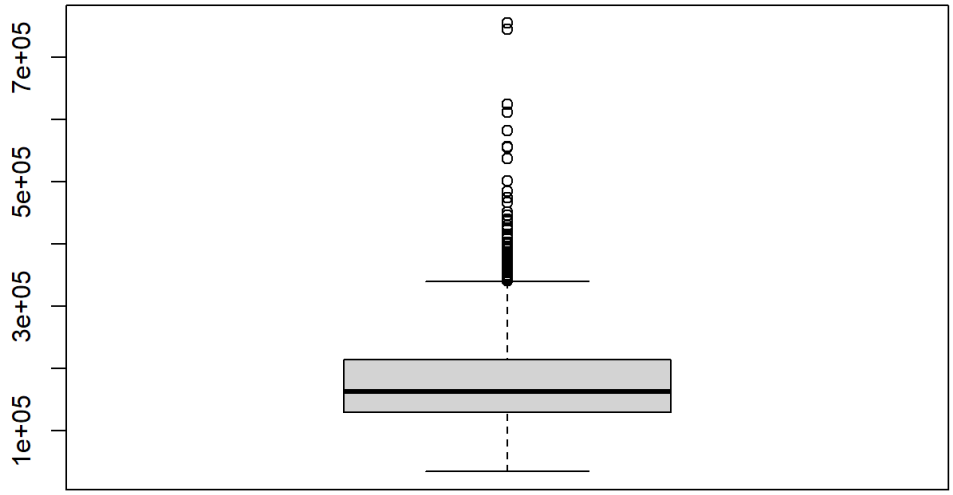
Var1	Freq
2006	313
2007	327
2008	301
2009	337
2010	174

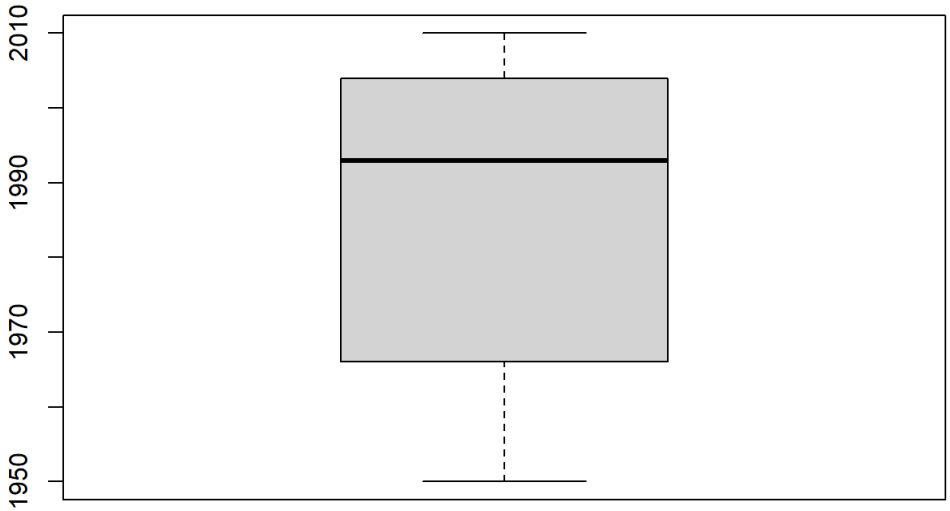
Not sure what to think of the variables with years…does it affect the target variable? Let's transform categorical to factor.

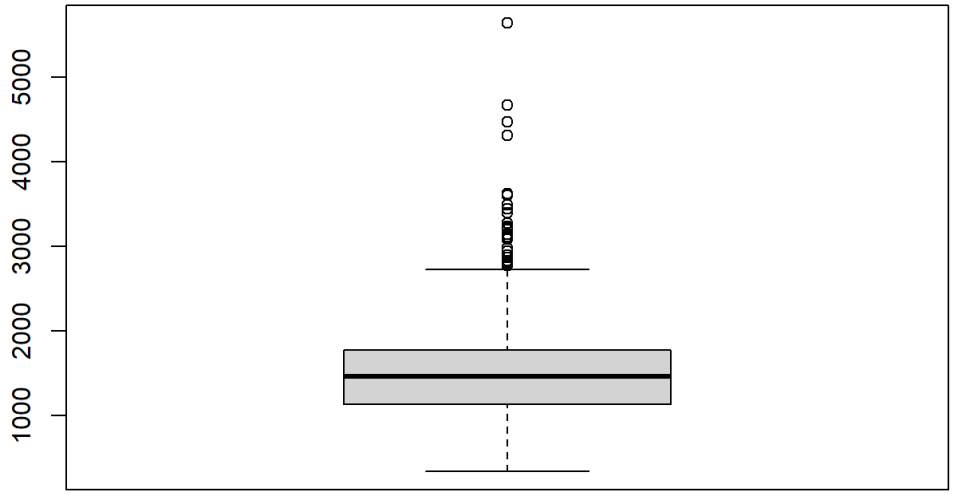
	MSZon… <fct>	LotArea <int>	Utilities <fct>	Neighborhood <fct>	BldgTy… <fct>	HouseStyle <fct>	OverallCond <int>	YearBuilt <int>	YearRemod… <int>
1	RL	8450	AllPub	CollgCr	1Fam	2Story	5	2003	2003
2	RL	9600	AllPub	Veenker	1Fam	1Story	8	1976	1976
3	RL	11250	AllPub	CollgCr	1Fam	2Story	5	2001	2002
4	RL	9550	AllPub	Crawfor	1Fam	2Story	5	1915	1970
5	RL	14260	AllPub	NoRidge	1Fam	2Story	5	2000	2000
6	RL	14115	AllPub	Mitchel	1Fam	1.5Fin	5	1993	1995

6 rows | 1-10 of 40 columns

Let's see some boxplot

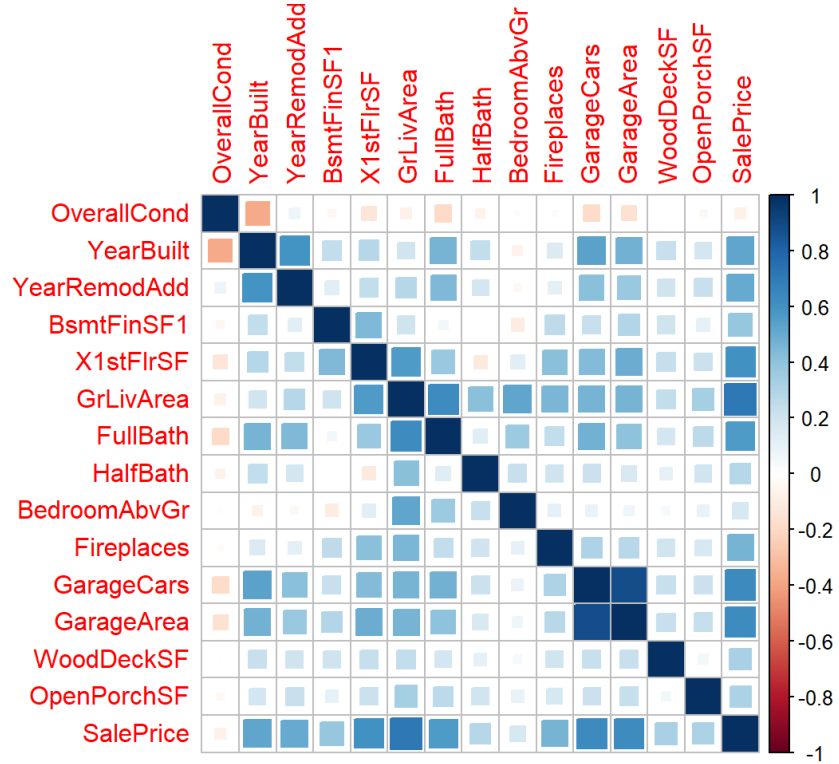

1e+05
3e+05
5e+05
7e+05


2010
1990
1970
1950



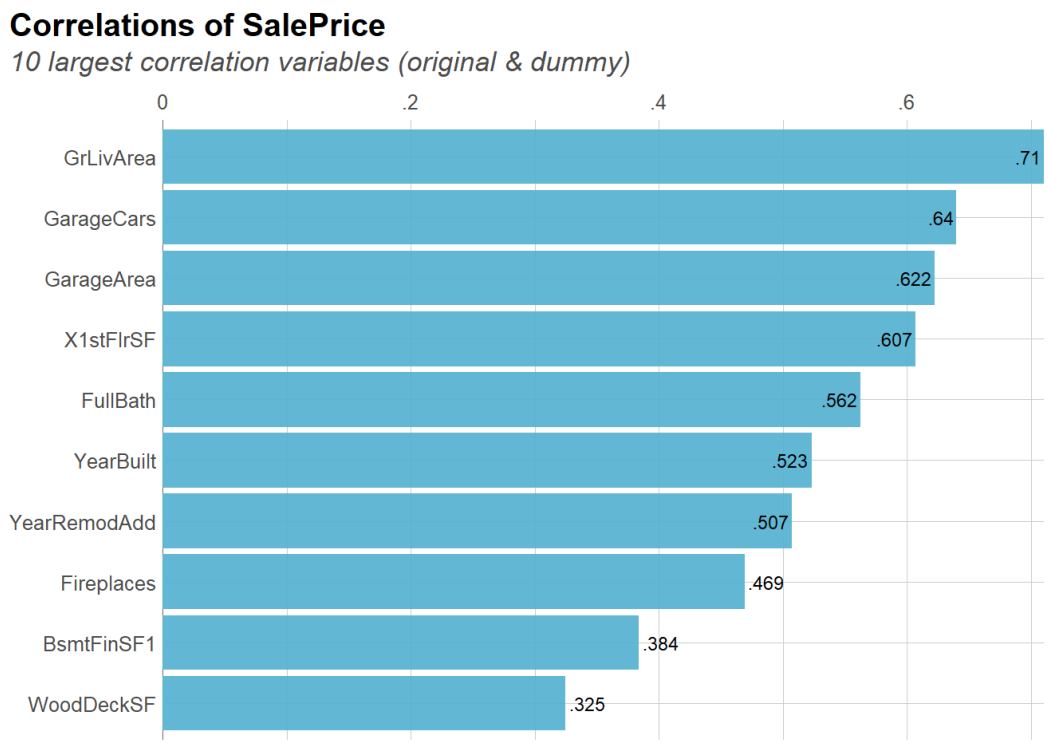
As we can see, SalePrice, GrLivArea have some outliers… For the sake that we want to run regression analysis, we limit the variables to numerical moving forward.

Let’s see some correlations



Based on the correlation plot, if we want to select those with high correlation with the target variable. Let's run another correlation function to help us select these variables rather than relying on visual (nothing wrong with visual)…

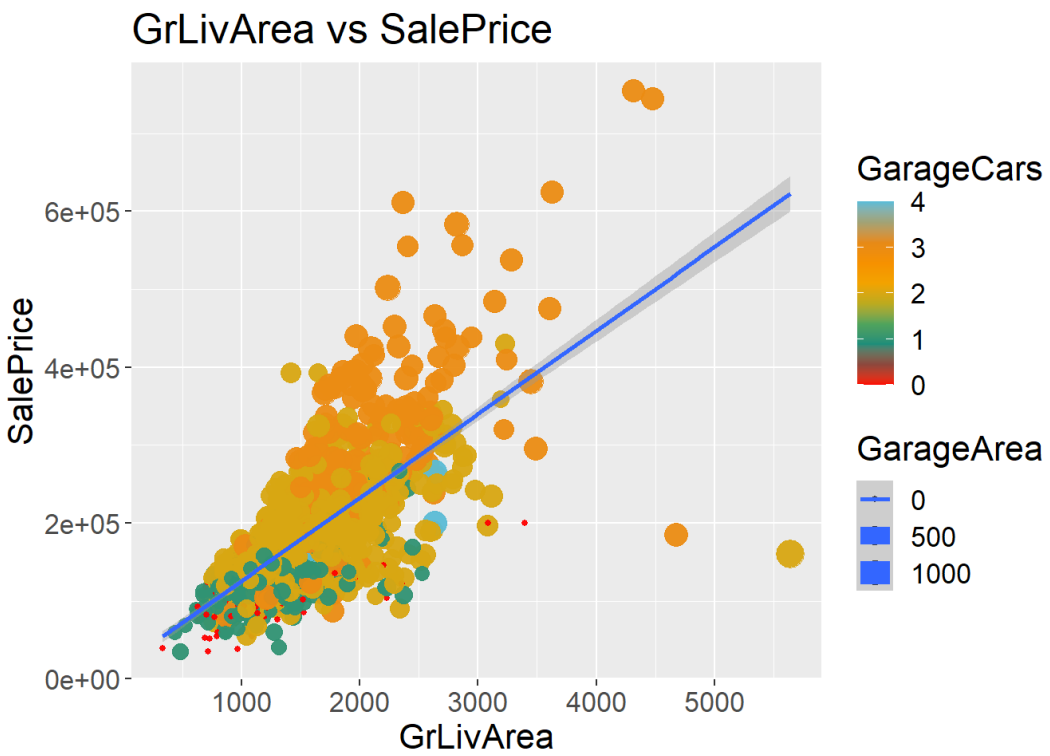
```
## Warning in .font_global(font, quiet = FALSE): Font 'Arial Narrow' is not
## installed, has other name, or can't be found
```



We will keep the variables with 0.5 above… there are really 08 variables ( )

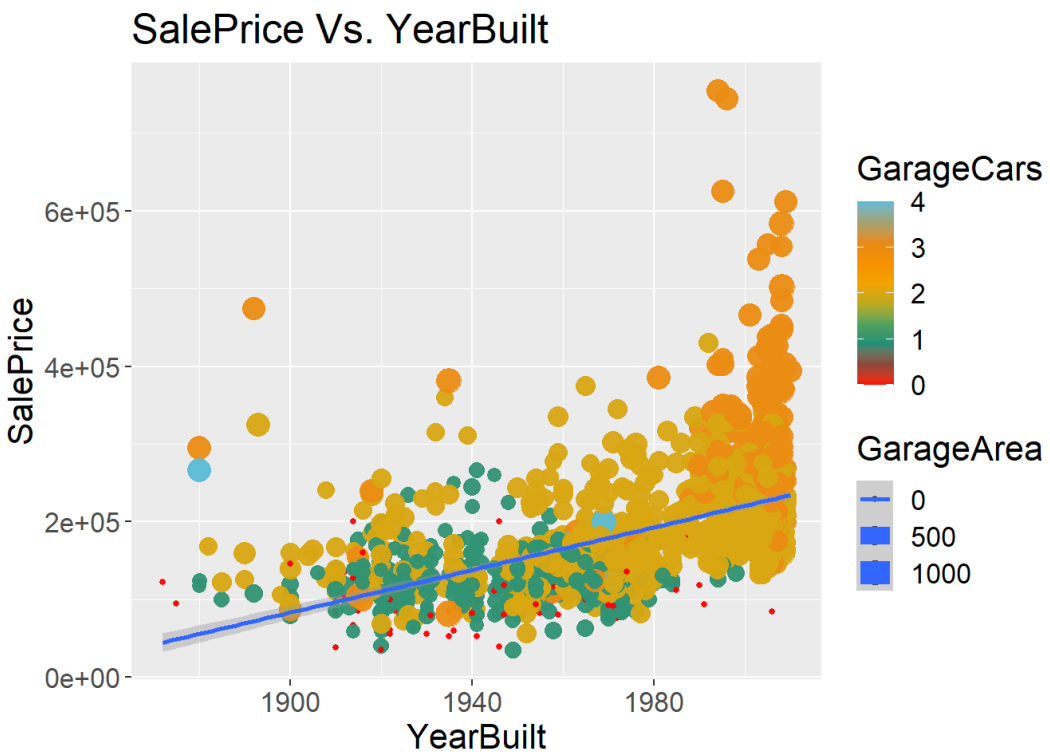
Let's see SalesPrice Vs. GrLivArea

```
## `geom_smooth()` using formula 'y ~ x'
```



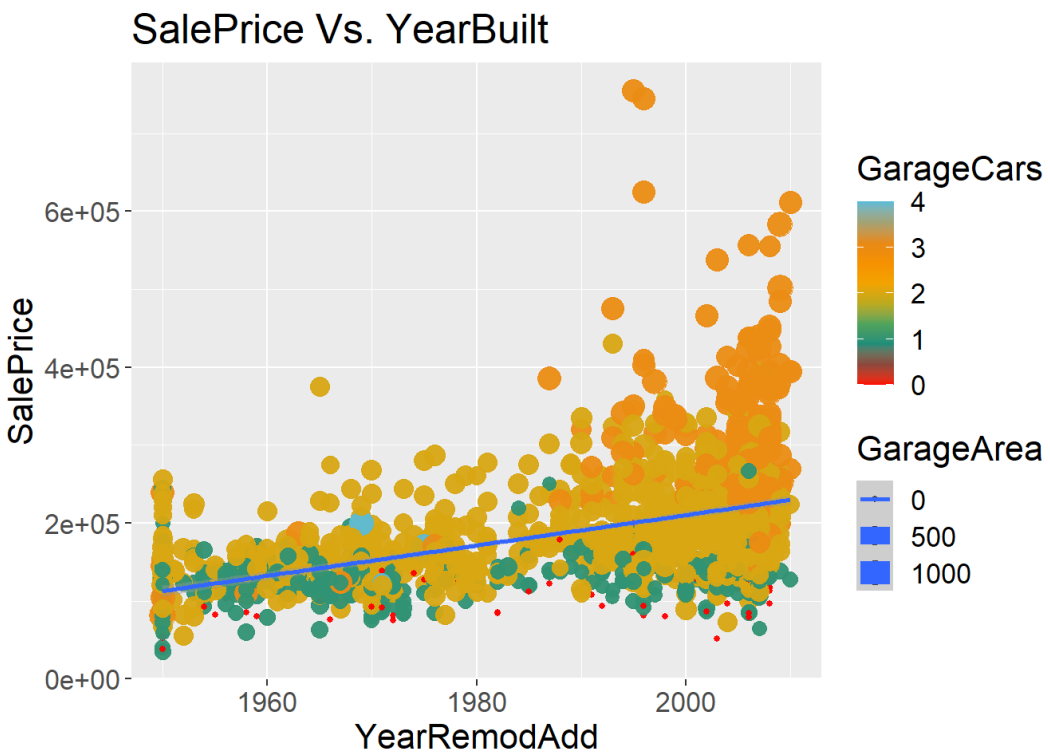
Let's see SalesPrice Vs. YearBuilt

```
## `geom_smooth()` using formula 'y ~ x'
```



Let's see SalesPrice Vs. YearRemodAdd

```
## `geom_smooth()` using formula 'y ~ x'
```



Definitely high correlation and linear relationship….

Modelling

We will run multiple linear regression and Neutral Network. There are definitely regression variables in this dataset. the neutral network is to see how other algorithm will perform, specially those on the unsupervised category. Neutral network seem to have perfomance in neuron analysis (according to neuroscientists who performed them…)

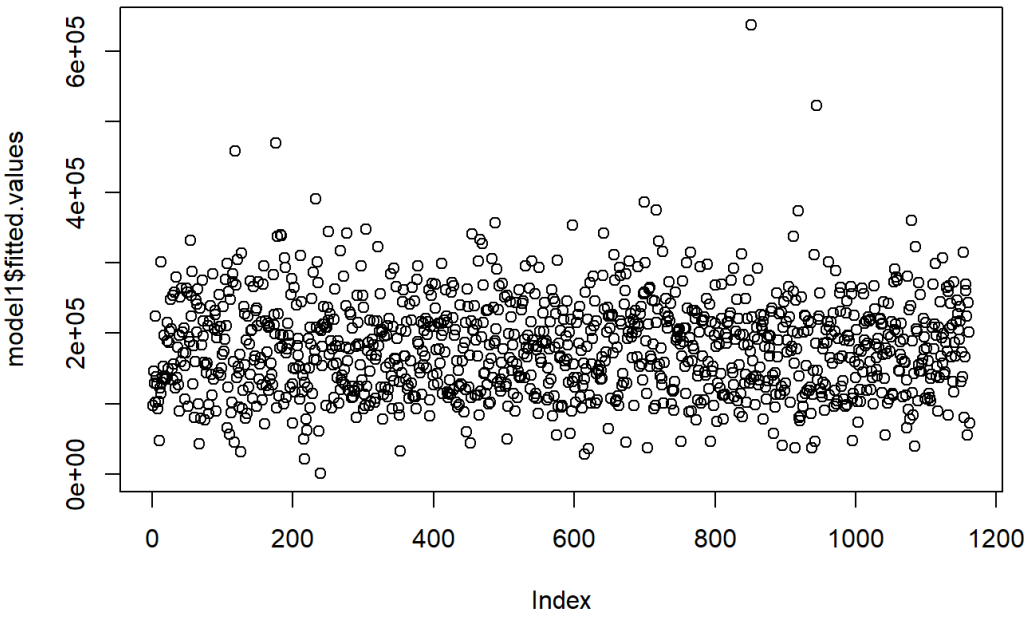
Splitting the data

Split will be 0.8 train and 0.2 test

Multiple Linear Regression (model1)

We want to use glm() function due to more than one independent/predictors.

```
##
## Call:
## glm(formula = SalePrice ~ ., data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -476681   -20824    -3747    17360   296387
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.114e+06  1.469e+05 -14.391  < 2e-16 ***
## YearBuilt     4.827e+02  6.049e+01   7.980 3.52e-15 ***
## YearRemodAdd  5.875e+02  7.809e+01   7.524 1.07e-13 ***
## X1stFlrSF     3.499e+01  4.143e+00   8.445  < 2e-16 ***
## GrLivArea     6.928e+01  3.665e+00  18.902  < 2e-16 ***
## FullBath     -4.560e+03  3.309e+03  -1.378    0.169
## GarageCars    1.886e+04  3.732e+03   5.053 5.04e-07 ***
## GarageArea    1.271e+01  1.294e+01   0.982    0.326
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1844051805)
##
##     Null deviance: 7.4845e+12  on 1161  degrees of freedom
## Residual deviance: 2.1280e+12  on 1154  degrees of freedom
## AIC: 28099
##
## Number of Fisher Scoring iterations: 2
```

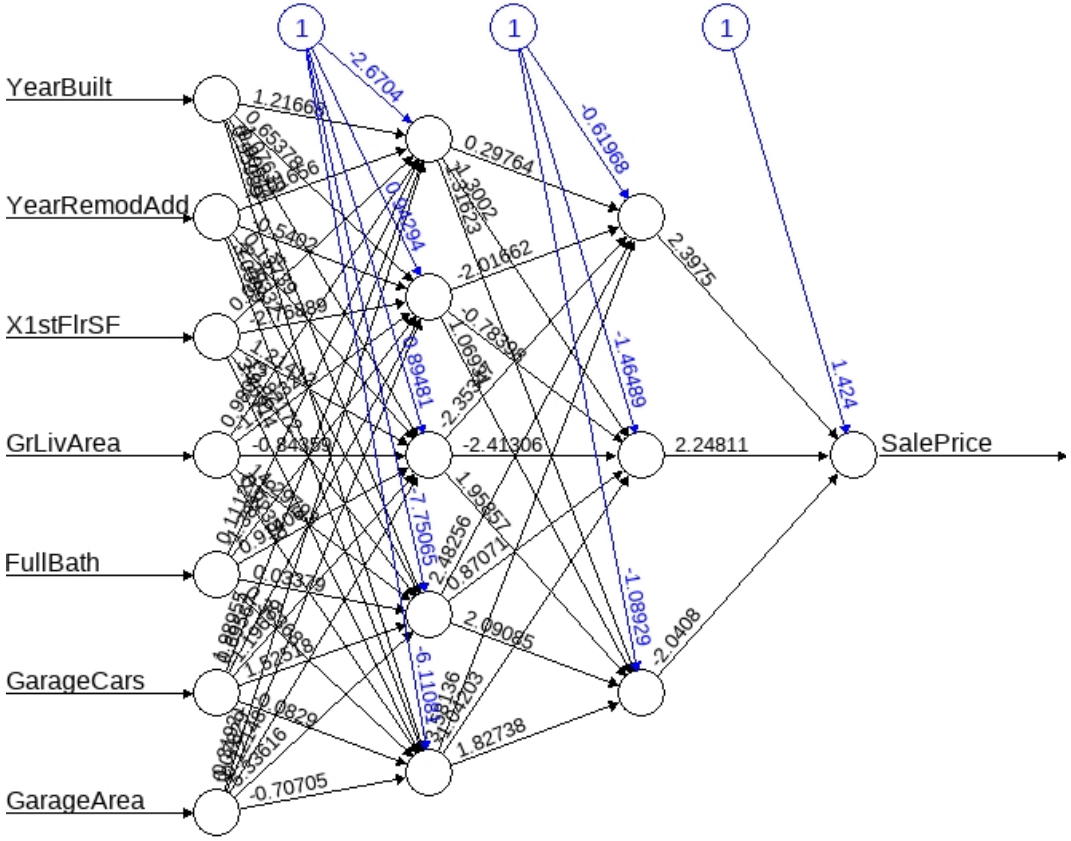


Wow! All the variables show good significance to the SalePrice.

Neutral Network (model2)

```
##
## Attaching package: 'neuralnet'
```

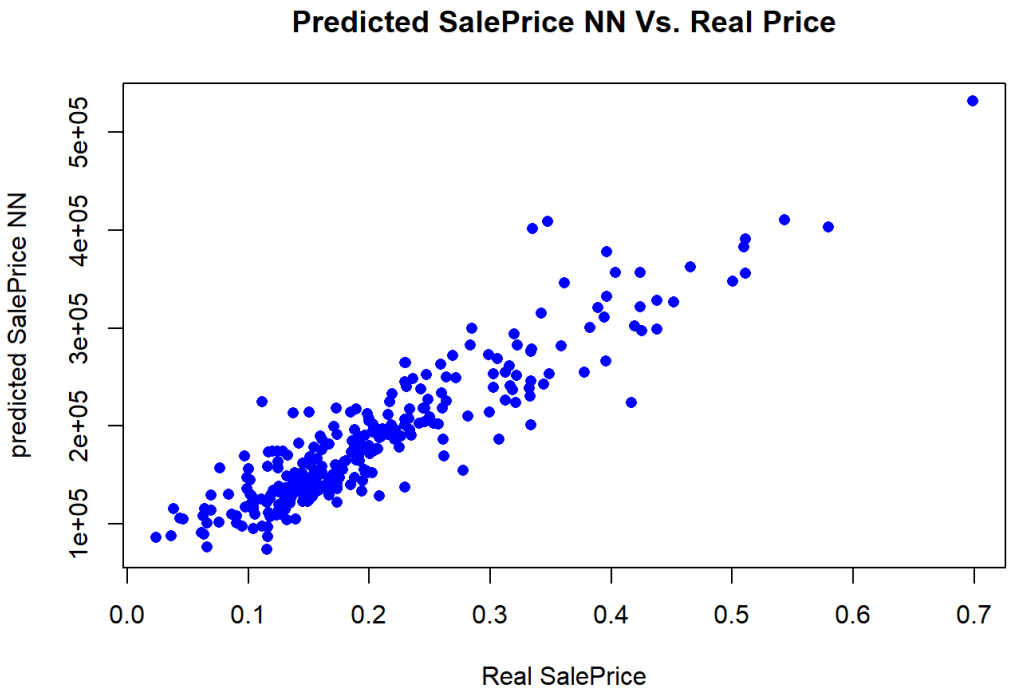
```
## The following object is masked from 'package:dplyr':
##
##     compute
```



Neutral network plot

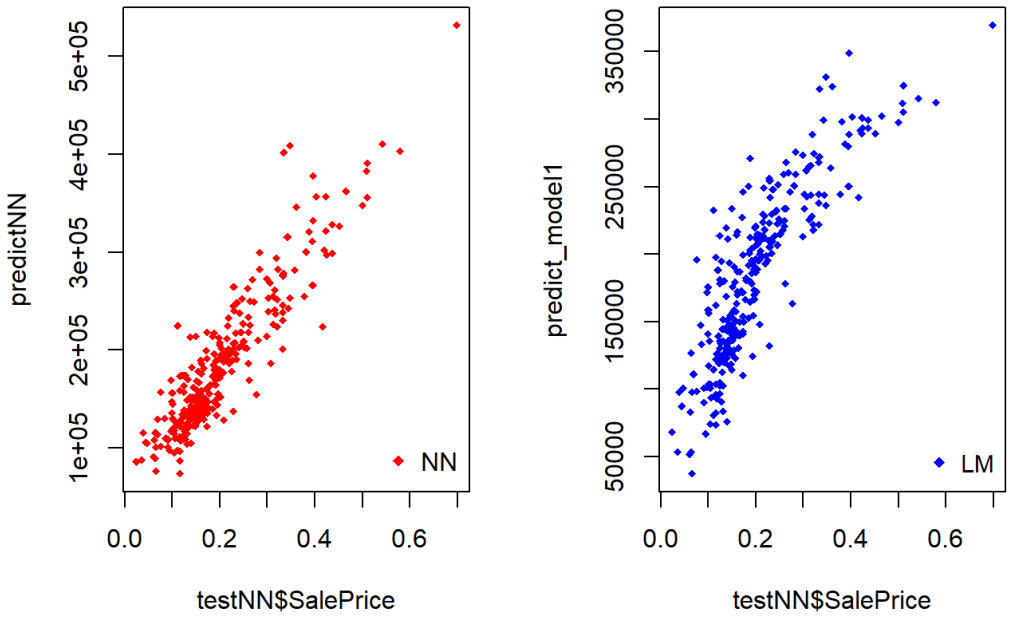
Well done!

predicting NN



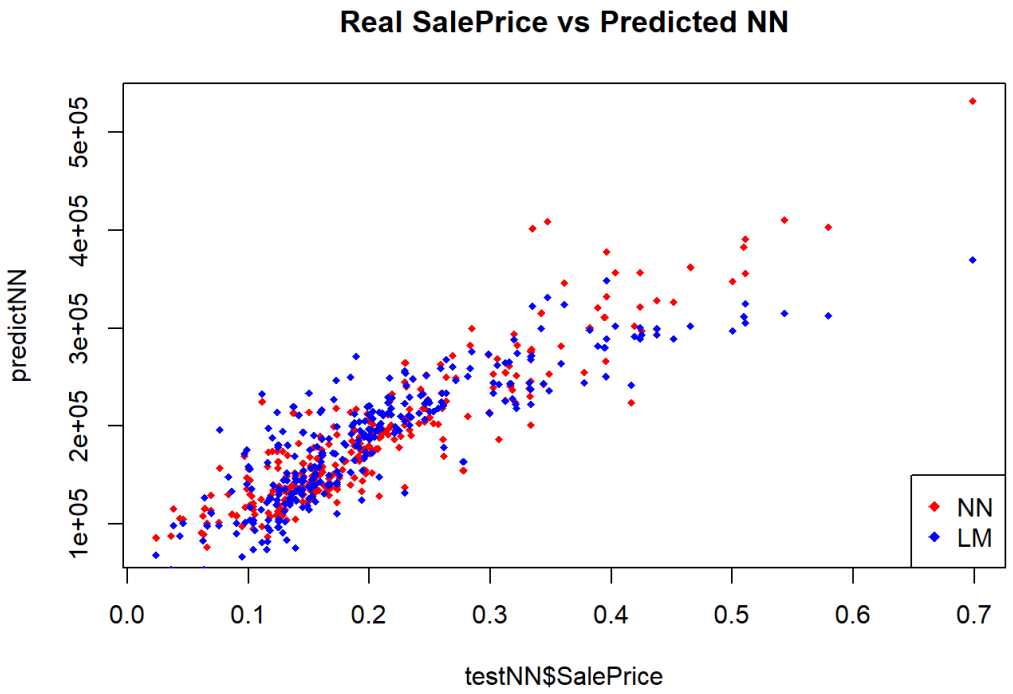

```
##                  MSE_model2
## MSE_model1         922874054.912377
##   1477004184.67592               1
```

Real SalesPrice vs Predicted SalePricl SalePrice vs Predicted SalePrice reg



Neutral Network seems about equal performance with multiple linear regression.

A closer look



It is a hard call here…but we want to go with Multiple linear regression since it has a better MSE and p-value.

References

1 - file:///C:/Users/owner/Downloads/622-Article%20Text-961-2-10-20220308.pdf

2 - http://neuralnetworksanddeeplearning.com/chap1.html

3- https://cran.r-project.org/web/packages/dlookr/vignettes/EDA.html

4- https://stats.oarc.ucla.edu/stat/data/rmarkdown/rmarkdown_seminar_flat.html#links-internal-and-external

5- https://plotly.com/r/line-charts/

6- https://www.r-bloggers.com/2015/09/fitting-a-neural-network-in-r-neuralnet-package/