**Ames, Iowa Housing Data**

The data used for this project was the Ames, Iowa housing data downloaded from Canvas, (Sign In - Google Accounts, n.d.-ag). This data contains 1,460 entries broken into two separate tables, training and testing, that contain the same variables. The training data consists of 1000 entries and the testing is 460 entries. There are 25 quantitative variables in each consisting of things such as number of rooms, year built, overall quality, etc. The variable of interest is the “Sale Price”. For this project I tried to determine if the sale price could be predicted, based on the other 24 variables.

**Sale Price**

I combined the two tables together, so I could look at the combined dataset to give me an overall look at the prices of all homes sold in Ames. I looked at the summary statistics for the variable “SalePrice” for the combined data set.

my\_data<-combine(my\_train1, my\_test1)

> summary(my\_data$SalePrice)

Min. 1st Qu. Median Mean 3rd Qu. Max.

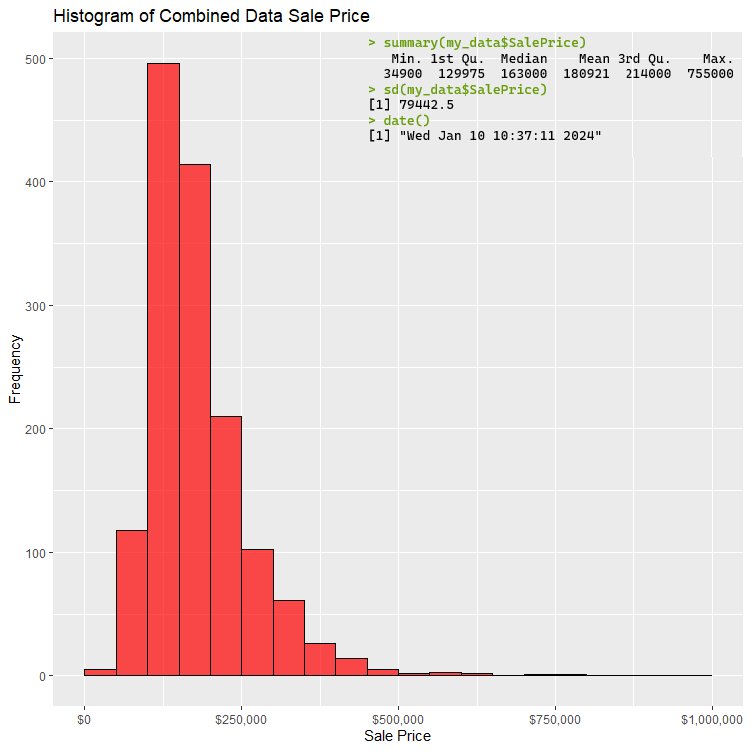
34900 129975 163000 180921 214000 755000

> sd(my\_data$SalePrice)

[1] 79442.5

From this we can see that the average price of a home was $180,921, the most expensive home sold for $755,000.00, the least expensive sold for just $34,900.00, and the standard deviation was $79,442.50 . The standard deviation was quite high suggests that the prices vary quite a bit from the average price. A higher standard deviation can also imply a higher risk or uncertainty when dealing with financial data, like the price of a home.

I also created a histogram of the sale price:



The histogram is clearly skewed to the right, meaning that the distribution of the sale price is not symmetric, the side with the higher values is longer than the lower values. We can also see this in the summary statistics, the mean, the average price, is higher than the median, the price that is the middle value, this is because the average is influenced by the long tail of the higher values.

This type of distribution is to expected when dealing with data like home prices. This is because there is a lower limit for the cost of a home, you can’t have a negative sale price, and there is no strict upper limit for the price of a home, the bigger and fancier it is the more it will cost.

I also did the same statistics and histogram for the testing data to ensure that it was representative of the whole.

summary(my\_test$SalePrice)

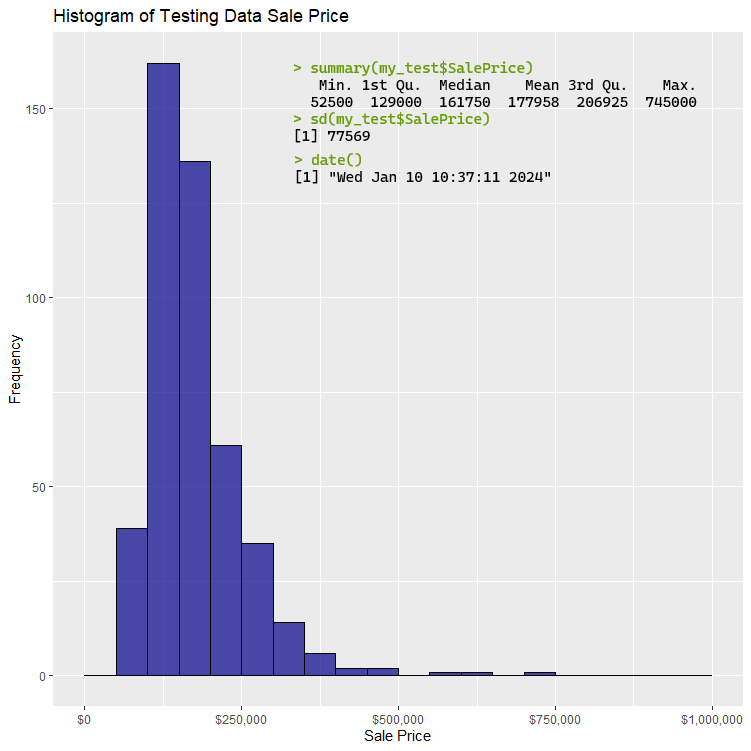
Min. 1st Qu. Median Mean 3rd Qu. Max.

52500 129000 161750 177958 206925 745000

> sd(my\_test$SalePrice)

[1] 77569

The testing data is very similar to the total dataset. The mean is only different by less than $3,000 and the standard deviation only varies by $1,873.50. Let’s look at the histogram to see if it has a similar distribution.



The distribution of the two datasets is very similar. All the things we said about the combined sale price apply to the testing dataset sale price.

**Making Predictions**

I first attempted a linear regression model using all 24 variables to attempt to make predictions. Here is the model, constructed in R.

model<-lm(my\_train1$SalePrice~.,data=my\_train1)

> summary(model)

Call:

lm(formula = my\_train1$SalePrice ~ ., data = my\_train1)

Residuals:

Min 1Q Median 3Q Max

-51940 -7739 0 7736 139326

Coefficients: (5 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2114420.8076 1624324.1131 1.302 0.193588

MSSubClass -35.6610 32.7107 -1.090 0.276134

LotFrontage101 -66291.6049 38258.8333 -1.733 0.083741 .

LotFrontage102 22184.5767 19538.8307 1.135 0.256728

LotFrontage103 -10026.1361 28068.2396 -0.357 0.721083

LotFrontage104 88233.5968 26839.9548 3.287 0.001080 \*\*

LotFrontage105 12613.4313 15628.7071 0.807 0.419997

LotFrontage106 -31863.6124 28759.9461 -1.108 0.268411

LotFrontage107 225225.0836 36576.7999 6.158 0.00000000148045832 \*\*\*

LotFrontage108 36967.3479 32374.8170 1.142 0.254041

LotFrontage109 2862.4342 34212.2116 0.084 0.933353

LotFrontage110 19725.6388 21062.2288 0.937 0.349430

LotFrontage111 32152.7863 25576.8546 1.257 0.209281

LotFrontage112 -26638.1164 27566.9386 -0.966 0.334341

LotFrontage114 28744.9302 25897.9205 1.110 0.267542

LotFrontage115 -8475.6439 26132.6643 -0.324 0.745818

LotFrontage116 -42943.2420 24611.2681 -1.745 0.081601 .

LotFrontage118 -47221.9991 26216.2626 -1.801 0.072244 .

LotFrontage120 -35418.8878 17266.5736 -2.051 0.040739 \*

LotFrontage121 -4863.5522 22680.9204 -0.214 0.830293

LotFrontage122 -33234.9931 18817.4090 -1.766 0.077953 .

LotFrontage128 -36278.8307 31894.6236 -1.137 0.255871

LotFrontage129 -20045.2417 25280.6228 -0.793 0.428193

LotFrontage130 -3699.4735 19496.8535 -0.190 0.849581

LotFrontage134 50390.2837 30898.9449 1.631 0.103537

LotFrontage137 -48260.1307 26715.5579 -1.806 0.071428 .

LotFrontage140 38714.5819 32230.9936 1.201 0.230237

LotFrontage141 14739.0715 36295.2958 0.406 0.684846

LotFrontage144 43429.6449 36302.3466 1.196 0.232113

LotFrontage149 -14162.0064 25191.3230 -0.562 0.574237

LotFrontage150 38044.5490 32973.6956 1.154 0.249119

LotFrontage174 -1484.7301 37515.5446 -0.040 0.968446

LotFrontage21 -13454.9851 14180.1018 -0.949 0.343132

LotFrontage24 -17787.9140 12310.4844 -1.445 0.149078

LotFrontage30 -388.2882 14366.9973 -0.027 0.978449

LotFrontage313 8785.2647 26892.9973 0.327 0.744046

LotFrontage32 22042.8949 18145.2718 1.215 0.224994

LotFrontage33 24943.9407 26712.6218 0.934 0.350847

LotFrontage34 -32304.4635 17908.2525 -1.804 0.071829 .

LotFrontage35 -13583.9477 15134.0648 -0.898 0.369829

LotFrontage36 -14190.9581 16380.0152 -0.866 0.386694

LotFrontage37 -2448.7891 27591.5650 -0.089 0.929314

LotFrontage38 35022.8822 26319.2233 1.331 0.183873

LotFrontage39 -11178.9143 27707.5903 -0.403 0.686776

LotFrontage40 -14379.8921 12808.8669 -1.123 0.262105

LotFrontage41 1742.4582 16886.2831 0.103 0.917854

LotFrontage42 24002.1779 25380.0087 0.946 0.344736

LotFrontage43 13185.1031 17482.4200 0.754 0.451076

LotFrontage44 -17984.3468 15727.8321 -1.143 0.253370

LotFrontage45 11680.8793 19131.1321 0.611 0.541752

LotFrontage47 -5480.6154 19335.1165 -0.283 0.776942

LotFrontage48 -18868.9460 16052.7307 -1.175 0.240360

LotFrontage49 25957.3640 16618.6722 1.562 0.118912

LotFrontage50 -1927.6986 9747.4469 -0.198 0.843307

LotFrontage51 -4514.7068 12430.9410 -0.363 0.716616

LotFrontage52 -9978.6973 12641.3737 -0.789 0.430257

LotFrontage53 -3365.6749 14185.0612 -0.237 0.812542

LotFrontage54 12524.5973 16201.8043 0.773 0.439852

LotFrontage55 2140.9492 11006.9007 0.195 0.845853

LotFrontage56 -14655.9789 25159.7200 -0.583 0.560471

LotFrontage57 7842.6128 12989.3787 0.604 0.546260

LotFrontage58 -940.4763 14606.4816 -0.064 0.948686

LotFrontage59 -12579.3831 12241.0348 -1.028 0.304598

LotFrontage60 -2757.0793 8996.2492 -0.306 0.759370

LotFrontage61 22614.8527 16960.9506 1.333 0.183002

LotFrontage62 2820.9182 14127.7479 0.200 0.841815

LotFrontage63 -11400.7157 11540.6343 -0.988 0.323674

LotFrontage64 -1750.7240 11997.8180 -0.146 0.884041

LotFrontage65 -11569.1748 9640.7078 -1.200 0.230674

LotFrontage66 -880.6405 11915.3509 -0.074 0.941112

LotFrontage67 -13506.6822 12555.0437 -1.076 0.282518

LotFrontage68 -10033.2615 11326.4536 -0.886 0.376122

LotFrontage69 10293.7112 12485.3699 0.824 0.410056

LotFrontage70 2799.1507 9218.3086 0.304 0.761515

LotFrontage71 1929.1858 13164.4481 0.147 0.883548

LotFrontage72 -9465.5268 12458.1844 -0.760 0.447729

LotFrontage73 -26246.9812 12005.3260 -2.186 0.029241 \*

LotFrontage74 4043.2188 11832.3511 0.342 0.732708

LotFrontage75 -790.0100 9630.9030 -0.082 0.934656

LotFrontage76 -2395.1800 15117.5569 -0.158 0.874174

LotFrontage77 664.2575 13539.7372 0.049 0.960890

LotFrontage78 -11085.9385 11336.3396 -0.978 0.328574

LotFrontage79 11592.4932 12072.8505 0.960 0.337396

LotFrontage80 -6177.7315 9788.0358 -0.631 0.528219

LotFrontage81 -3308.0108 15703.3945 -0.211 0.833239

LotFrontage82 -8879.5686 15127.2438 -0.587 0.557464

LotFrontage83 -14008.1641 20977.2864 -0.668 0.504572

LotFrontage84 341.7902 13785.0321 0.025 0.980229

LotFrontage85 -11436.8676 10850.2496 -1.054 0.292344

LotFrontage86 36230.6911 15385.4123 2.355 0.018900 \*

LotFrontage87 -27696.3444 18333.9676 -1.511 0.131484

LotFrontage88 5519.8562 16868.7671 0.327 0.743631

LotFrontage89 -12657.1352 25082.3058 -0.505 0.614037

LotFrontage90 10047.6888 11960.3579 0.840 0.401249

LotFrontage91 15198.2216 20483.8864 0.742 0.458447

LotFrontage92 4900.5990 13293.3898 0.369 0.712539

LotFrontage93 -150.7743 20282.9724 -0.007 0.994072

LotFrontage94 -21239.5030 16688.4568 -1.273 0.203692

LotFrontage95 -18259.4049 18631.2526 -0.980 0.327522

LotFrontage96 15326.2866 25907.6916 0.592 0.554394

LotFrontage97 -80978.5520 37995.6089 -2.131 0.033537 \*

LotFrontage98 -52812.9744 19387.8910 -2.724 0.006667 \*\*

LotFrontage99 -13456.7433 26761.5520 -0.503 0.615291

LotFrontageNA -3666.6446 8652.4565 -0.424 0.671910

LotArea 0.4967 0.1138 4.366 0.00001528425781881 \*\*\*

OverallQual 10324.7996 1399.4744 7.378 0.00000000000064305 \*\*\*

OverallCond 5455.4756 1139.6873 4.787 0.00000221291951365 \*\*\*

YearBuilt 415.8431 69.9905 5.941 0.00000000518412976 \*\*\*

YearRemodAdd 198.4933 73.1308 2.714 0.006864 \*\*

MasVnrArea1 11440.2510 24429.3475 0.468 0.639767

MasVnrArea100 35553.3223 17828.8616 1.994 0.046659 \*

MasVnrArea101 -3202.9154 18507.8379 -0.173 0.862674

MasVnrArea102 -5141.2122 25799.6406 -0.199 0.842126

MasVnrArea1031 83982.4243 26112.7025 3.216 0.001380 \*\*

MasVnrArea104 -199.5458 15342.7621 -0.013 0.989628

MasVnrArea1047 26811.0047 26311.6776 1.019 0.308688

MasVnrArea105 15703.7821 17893.7505 0.878 0.380560

MasVnrArea106 -5262.3157 15541.9165 -0.339 0.735057

MasVnrArea108 -4620.4426 10217.6919 -0.452 0.651314

MasVnrArea109 1366.6321 24135.0677 0.057 0.954866

MasVnrArea11 -39655.4918 28228.5692 -1.405 0.160679

MasVnrArea110 16937.1157 17732.4453 0.955 0.339947

MasVnrArea1115 -39332.7509 30321.1680 -1.297 0.195137

MasVnrArea112 -9200.0084 24278.3146 -0.379 0.704888

MasVnrArea1129 26521.5457 29093.9079 0.912 0.362411

MasVnrArea113 -46985.5597 17555.6630 -2.676 0.007678 \*\*

MasVnrArea115 -25181.4078 25071.9758 -1.004 0.315671

MasVnrArea116 4409.8979 15193.7491 0.290 0.771745

MasVnrArea117 -13449.8023 28557.3804 -0.471 0.637857

MasVnrArea1170 198447.8118 36087.3058 5.499 0.00000005998875378 \*\*\*

MasVnrArea120 -411.6323 12997.1526 -0.032 0.974747

MasVnrArea122 -33255.1261 24739.8140 -1.344 0.179473

MasVnrArea125 -17248.1387 19807.5524 -0.871 0.384274

MasVnrArea126 -7501.8348 24570.0853 -0.305 0.760242

MasVnrArea127 10706.9938 27629.8316 0.388 0.698533

MasVnrArea128 -4128.3221 16856.4857 -0.245 0.806623

MasVnrArea130 390.4297 25176.3342 0.016 0.987633

MasVnrArea132 -10434.0435 20243.7466 -0.515 0.606479

MasVnrArea135 13827.9940 24877.2568 0.556 0.578554

MasVnrArea136 2957.1404 14225.7373 0.208 0.835410

MasVnrArea137 8949.3356 24551.8866 0.365 0.715628

MasVnrArea138 -21039.8102 26912.4664 -0.782 0.434696

MasVnrArea140 -11035.5778 27083.9166 -0.407 0.683839

MasVnrArea142 13081.7569 28516.4456 0.459 0.646610

MasVnrArea144 -25075.8952 25211.5319 -0.995 0.320385

MasVnrArea145 53052.8805 25557.8902 2.076 0.038405 \*

MasVnrArea146 7492.5191 25358.1207 0.295 0.767754

MasVnrArea147 -9220.2998 24206.5194 -0.381 0.703432

MasVnrArea148 -7900.9878 18260.6829 -0.433 0.665429

MasVnrArea150 NA NA NA NA

MasVnrArea151 28505.5165 25425.8681 1.121 0.262754

MasVnrArea153 12846.0303 18762.2364 0.685 0.493855

MasVnrArea156 5514.5803 25557.0427 0.216 0.829248

MasVnrArea157 -26121.8844 18558.5230 -1.408 0.159866

MasVnrArea158 -7356.4356 26197.6110 -0.281 0.778971

MasVnrArea16 -42875.5422 16604.7911 -2.582 0.010092 \*

MasVnrArea160 21661.1086 26678.6594 0.812 0.417206

MasVnrArea1600 12873.0198 25671.0603 0.501 0.616260

MasVnrArea161 1868.8139 24251.8327 0.077 0.938607

MasVnrArea162 45759.7260 18533.0307 2.469 0.013867 \*

MasVnrArea164 699.5083 26042.6645 0.027 0.978582

MasVnrArea166 -19357.8310 29589.1637 -0.654 0.513260

MasVnrArea167 16369.2082 25235.4851 0.649 0.516846

MasVnrArea168 4187.3466 14808.1770 0.283 0.777464

MasVnrArea169 -3392.2266 17848.6983 -0.190 0.849341

MasVnrArea170 -7577.2763 18610.6704 -0.407 0.684068

MasVnrArea172 3936.9251 19120.2160 0.206 0.836947

MasVnrArea174 -5223.8631 18315.1870 -0.285 0.775589

MasVnrArea175 -19484.7659 25929.0020 -0.751 0.452713

MasVnrArea176 24659.9971 15528.9856 1.588 0.112896

MasVnrArea178 -7264.4692 15099.2763 -0.481 0.630638

MasVnrArea18 -10223.8231 17301.5373 -0.591 0.554831

MasVnrArea180 -14294.2002 11609.1842 -1.231 0.218774

MasVnrArea182 27513.5273 29134.5513 0.944 0.345425

MasVnrArea183 -2453.2293 13850.6620 -0.177 0.859483

MasVnrArea184 -25158.0606 25429.0240 -0.989 0.322956

MasVnrArea186 52400.3092 24424.5379 2.145 0.032385 \*

MasVnrArea188 22719.2243 24875.3347 0.913 0.361497

MasVnrArea189 -16355.1794 24681.0332 -0.663 0.507841

MasVnrArea192 -31800.2607 19233.2938 -1.653 0.098855 .

MasVnrArea196 12102.8676 17400.4878 0.696 0.487024

MasVnrArea200 41217.2068 13398.4362 3.076 0.002207 \*\*

MasVnrArea202 -11846.5338 24641.5441 -0.481 0.630894

MasVnrArea203 -36837.8237 18470.2108 -1.994 0.046626 \*

MasVnrArea204 11402.7668 24503.3772 0.465 0.641873

MasVnrArea205 13239.1695 17620.9065 0.751 0.452793

MasVnrArea207 -9717.7768 25302.4033 -0.384 0.701087

MasVnrArea208 39020.9215 22979.6244 1.698 0.090095 .

MasVnrArea209 -100741.7507 25371.7380 -3.971 0.00008179894762723 \*\*\*

MasVnrArea210 -14202.8005 14938.6805 -0.951 0.342179

MasVnrArea212 -8304.1638 25400.2298 -0.327 0.743851

MasVnrArea215 -853.9285 18080.0639 -0.047 0.962348

MasVnrArea216 18559.5921 15769.5856 1.177 0.239766

MasVnrArea219 -16155.2527 26358.1123 -0.613 0.540202

MasVnrArea22 -12749.5674 26148.8414 -0.488 0.626056

MasVnrArea220 38310.6298 15013.7312 2.552 0.011005 \*

MasVnrArea223 23073.2977 27443.7176 0.841 0.400875

MasVnrArea224 13466.4447 34352.7125 0.392 0.695215

MasVnrArea225 -12269.7115 24786.6644 -0.495 0.620801

MasVnrArea226 18040.2981 19334.1235 0.933 0.351212

[ reached getOption("max.print") -- omitted 286 rows ]

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22970 on 519 degrees of freedom

Multiple R-squared: 0.9575, Adjusted R-squared: 0.9182

F-statistic: 24.36 on 480 and 519 DF, p-value: < 0.00000000000000022

That is a lot of information, but I will explain the important parts. First the residuals:

Residuals:

Min 1Q Median 3Q Max

-51940 -7739 0 7736 139326

The residuals tell us the difference between observed and predicted values in the training data. Here the values range from underestimating the sales price by $51,940 to overestimating by $139,326.

The coefficients are the long string of data in the middle. Each of the coefficients tell the estimated change in the variable for a corresponding change in the sale price. For example:

LotFrontage104 88233.5968 26839.9548 3.287 0.001080 \*\*

“LotFrontage104” has a coefficient of 88233.5968. This means that if all other variables remained the same and there was a one-unit increase in “LotFrontage104” there would be an increase in sales price of $88,233.60. If the coefficient is negative, then an increase in that variable would have a negative effect on the price. The P-values associated with each coefficient suggests the significance of the estimated coefficient, the lower the more significant it is.

The following variables all showed a significant relationship with the sale price at the P < 0.05 level:

> summary\_model <- summary(price\_model)

> coefficients\_table <- summary\_model$coefficients

> significant\_coefficients <- coefficients\_table[coefficients\_table[, "Pr(>|t|)"] < 0.05, ]

> print(significant\_coefficients)

Estimate Std. Error t value Pr(>|t|)

LotFrontage104 8.822674e+04 2.684482e+04 3.286546 1.082721e-03

LotFrontage107 2.255048e+05 3.658253e+04 6.164276 1.421553e-09

LotFrontage120 -3.489290e+04 1.726296e+04 -2.021259 4.376483e-02

LotFrontage73 -2.570059e+04 1.199703e+04 -2.142245 3.263751e-02

LotFrontage86 3.721458e+04 1.536170e+04 2.422556 1.575242e-02

LotFrontage97 -7.983520e+04 3.798802e+04 -2.101589 3.606980e-02

LotFrontage98 -5.241627e+04 1.938799e+04 -2.703543 7.085024e-03

LotArea 4.904601e-01 1.136461e-01 4.315679 1.905232e-05

OverallQual 1.000423e+04 1.368482e+03 7.310458 1.010932e-12

OverallCond 5.482269e+03 1.139629e+03 4.810575 1.974398e-06

YearBuilt 4.207161e+02 6.986032e+01 6.022247 3.254811e-09

YearRemodAdd 2.017166e+02 7.308430e+01 2.760054 5.983124e-03

MasVnrArea1031 8.455778e+04 2.611210e+04 3.238260 1.279384e-03

MasVnrArea113 -4.635397e+04 1.754928e+04 -2.641360 8.505885e-03

MasVnrArea1170 1.994450e+05 3.608225e+04 5.527509 5.145898e-08

MasVnrArea145 5.164906e+04 2.553006e+04 2.023069 4.357720e-02

MasVnrArea16 -4.417369e+04 1.656504e+04 -2.666682 7.899026e-03

MasVnrArea162 4.622224e+04 1.853153e+04 2.494248 1.293182e-02

MasVnrArea186 5.393821e+04 2.438819e+04 2.211653 2.742471e-02

MasVnrArea200 4.109150e+04 1.340037e+04 3.066446 2.278889e-03

MasVnrArea203 -3.742704e+04 1.846565e+04 -2.026847 4.318784e-02

MasVnrArea209 -1.002325e+05 2.537203e+04 -3.950510 8.874738e-05

MasVnrArea220 3.798432e+04 1.501347e+04 2.530016 1.169983e-02

MasVnrArea240 -5.501533e+04 1.864910e+04 -2.950025 3.320643e-03

MasVnrArea256 -3.761765e+04 1.664234e+04 -2.260358 2.421176e-02

MasVnrArea268 7.194815e+04 2.745456e+04 2.620627 9.033568e-03

MasVnrArea287 -7.762586e+04 2.566984e+04 -3.024011 2.617679e-03

MasVnrArea304 1.125982e+05 3.073955e+04 3.662975 2.748436e-04

MasVnrArea305 7.579283e+04 2.809292e+04 2.697933 7.203825e-03

MasVnrArea312 7.462370e+04 2.570114e+04 2.903517 3.846745e-03

MasVnrArea315 -1.348394e+05 4.289662e+04 -3.143356 1.765551e-03

MasVnrArea333 -6.754134e+04 2.691504e+04 -2.509427 1.239559e-02

MasVnrArea375 7.239727e+04 3.079205e+04 2.351168 1.908693e-02

MasVnrArea378 7.675233e+04 3.018253e+04 2.542939 1.128111e-02

MasVnrArea380 1.010732e+05 4.440132e+04 2.276355 2.323039e-02

MasVnrArea388 7.465655e+04 3.521688e+04 2.119908 3.448682e-02

MasVnrArea420 1.485306e+05 4.770528e+04 3.113504 1.950612e-03

MasVnrArea435 -6.089423e+04 2.677060e+04 -2.274668 2.333226e-02

MasVnrArea436 -1.377239e+05 4.274830e+04 -3.221739 1.353937e-03

MasVnrArea466 6.986411e+04 2.731587e+04 2.557638 1.082106e-02

MasVnrArea50 -4.525957e+04 1.978194e+04 -2.287924 2.254244e-02

MasVnrArea506 8.827308e+04 4.468134e+04 1.975614 4.872679e-02

MasVnrArea528 8.918336e+04 2.731785e+04 3.264655 1.168124e-03

MasVnrArea530 5.148133e+04 2.530980e+04 2.034047 4.245386e-02

MasVnrArea562 8.538369e+04 2.550248e+04 3.348055 8.727694e-04

MasVnrArea594 6.088019e+04 2.969558e+04 2.050143 4.085134e-02

MasVnrArea60 4.452866e+04 1.806297e+04 2.465190 1.401604e-02

MasVnrArea603 1.596764e+05 3.039858e+04 5.252757 2.188949e-07

MasVnrArea673 5.768498e+04 2.798701e+04 2.061134 3.978687e-02

MasVnrArea72 -3.213147e+04 1.237452e+04 -2.596582 9.682079e-03

MasVnrArea748 1.263593e+05 2.758255e+04 4.581133 5.795366e-06

MasVnrArea760 2.268394e+05 2.741413e+04 8.274543 1.090585e-15

MasVnrArea762 -3.471152e+05 3.216247e+04 -10.792556 1.218491e-24

MasVnrArea870 1.097987e+05 3.404312e+04 3.225283 1.337611e-03

MasVnrArea975 1.120789e+05 2.999577e+04 3.736488 2.072702e-04

TotalBsmtSF 3.021695e+01 3.545192e+00 8.523359 1.688552e-16

GrLivArea 5.370902e+01 5.165608e+00 10.397426 3.927581e-23

BedroomAbvGr -1.127731e+04 1.913558e+03 -5.893375 6.807603e-09

KitchenAbvGr -2.023811e+04 5.326573e+03 -3.799462 1.621571e-04

TotRmsAbvGrd 3.718940e+03 1.444019e+03 2.575409 1.028728e-02

Fireplaces 6.169817e+03 1.938254e+03 3.183184 1.543817e-03

GarageYrBlt1916 6.503600e+04 3.003049e+04 2.165666 3.079062e-02

GarageYrBlt1921 8.126411e+04 2.910295e+04 2.792298 5.426200e-03

GarageYrBlt1922 6.944286e+04 2.776946e+04 2.500692 1.270172e-02

GarageYrBlt1925 5.196587e+04 2.590786e+04 2.005796 4.539553e-02

GarageYrBlt1926 5.862835e+04 2.679328e+04 2.188174 2.910118e-02

GarageYrBlt1930 5.284309e+04 2.614766e+04 2.020949 4.379695e-02

GarageYrBlt1934 9.641237e+04 2.926434e+04 3.294535 1.053028e-03

GarageYrBlt1939 5.999835e+04 2.580675e+04 2.324909 2.046087e-02

GarageYrBlt1940 5.401978e+04 2.493446e+04 2.166471 3.072876e-02

GarageYrBlt1945 6.420227e+04 2.894352e+04 2.218192 2.697301e-02

GarageYrBlt1949 5.558730e+04 2.535100e+04 2.192706 2.877084e-02

GarageYrBlt1950 6.012988e+04 2.460640e+04 2.443668 1.487015e-02

GarageYrBlt1954 4.986544e+04 2.473302e+04 2.016148 4.429820e-02

GarageYrBlt1956 5.599307e+04 2.596817e+04 2.156219 3.152442e-02

GarageYrBlt1959 5.905010e+04 2.585373e+04 2.284007 2.277338e-02

GarageYrBlt1960 5.960233e+04 2.587920e+04 2.303098 2.166712e-02

GarageYrBlt1961 5.708347e+04 2.517776e+04 2.267218 2.378660e-02

GarageYrBlt1962 5.344158e+04 2.506991e+04 2.131702 3.349952e-02

GarageYrBlt1964 5.195194e+04 2.548544e+04 2.038495 4.200580e-02

GarageYrBlt1966 5.409461e+04 2.588986e+04 2.089413 3.715601e-02

GarageYrBlt1972 7.117429e+04 2.727040e+04 2.609946 9.316668e-03

GarageYrBlt1973 6.086931e+04 2.584027e+04 2.355599 1.886325e-02

GarageYrBlt1978 5.540312e+04 2.555211e+04 2.168240 3.059318e-02

GarageYrBlt1981 5.936771e+04 2.667484e+04 2.225607 2.646851e-02

GarageYrBlt1988 5.152272e+04 2.595705e+04 1.984922 4.767818e-02

GarageYrBlt1992 5.216213e+04 2.583450e+04 2.019088 4.399065e-02

GarageYrBlt1995 5.595121e+04 2.680506e+04 2.087338 3.734388e-02

GarageYrBlt1999 5.122453e+04 2.517692e+04 2.034583 4.239963e-02

GarageYrBlt2000 5.344056e+04 2.524223e+04 2.117109 3.472478e-02

GarageYrBlt2002 5.098895e+04 2.550281e+04 1.999346 4.609076e-02

GarageYrBlt2007 5.500002e+04 2.510270e+04 2.191000 2.889481e-02

GarageYrBlt2008 9.148832e+04 2.619951e+04 3.491985 5.202565e-04

GarageYrBlt2009 7.089243e+04 2.659377e+04 2.665753 7.920572e-03

GarageYrBlt2010 1.091812e+05 3.462131e+04 3.153584 1.705969e-03

GarageYrBltNA 6.913172e+04 2.433726e+04 2.840571 4.679816e-03

GarageArea 3.333766e+01 1.183435e+01 2.817026 5.031359e-03

WoodDeckSF 2.320860e+01 9.079042e+00 2.556283 1.086275e-02

YrSold -1.678465e+03 8.079313e+02 -2.077485 3.824700e-02

> date()

[1] "Wed Jan 10 16:15:21 2024"

Here is the model overall significance, the F-Test.

> f\_test <- summary(price\_model)$fstatistic

> f\_test <- summary(price\_model)$fstatistic

> print(f\_test)

value numdf dendf

24.39635 479.00000 520.00000

> date()

[1] "Wed Jan 10 16:19:49 2024"

To perform a hypothesis test, we have all the information we need in the two tables above. Let’s use the variable ‘LotFrontage104’ again. The null hypothesis, H0, assumes that ‘LotFrontage104’ has no effect on the sale price meaning that the coefficient is equal to zero. We just need to look at the P-value, which in this case is 1.082721e-03. We can write R code to do this for us:

> t\_value <- summary(price\_model)$coefficients["LotFrontage104", "t value"]

> p\_value <- summary(price\_model)$coefficients["LotFrontage104", "Pr(>|t|)"]

> if (p\_value < 0.05) {

+ print("Reject the null hypothesis; there is a significant effect.")

+ } else {

+ print("Fail to reject the null hypothesis; there is no significant effect.")

+ }

[1] **"Reject the null hypothesis; there is a significant effect."**

> date()

[1] "Wed Jan 10 16:34:15 2024"

So, we can reject the null hypothesis for ‘LotFrontage’, it does have a significant effect on the variable ‘SalePrice’.

The R-square value measures the amount of variance in the dependent variable, ‘SalePrice’, that is predictable from the independent variables. Adjusted R-square adjusts for the number of predictor variables. In this case the values are:

Multiple R-squared: 0.9575, Adjusted R-squared: 0.9182

The R-squared and adjusted R-squared will have a value between 0 and 1, a higher value indicates a better fit, so this model with an adjusted R-square value of 0.9182 is considered a “good fit”.

The F-statistic tests the overall significance of the model.

F-statistic: 24.36 on 480 and 519 DF, p-value: < 0.00000000000000022

The extremely low P-value suggests that at least one of the variables is significantly related to the sale price.

The next step is to use the model on the testing data to see how well it performs. I ran it on only the first twenty entries of the testing data. During my first attempt I received this error:

> predicted\_prices <- predict(price\_model, newdata = test\_subset)

Error: variables ‘LotFrontage’, ‘MasVnrArea’, ‘GarageYrBlt’ were specified with different types from the fit

In addition: Warning messages:

1: In model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels) :

variable 'LotFrontage' is not a factor

2: In model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels) :

variable 'MasVnrArea' is not a factor

3: In model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels) :

variable 'GarageYrBlt' is not a factor

This was a result of three variables, ‘LotFrontage’, ‘MasVnrArea’, and ‘GarageYrBlt’ being character variables instead of numeric variables. So, I converted those three variables to numeric and tried again.

> numeric\_vars <- c("LotFrontage", "MasVnrArea", "GarageYrBlt")

> test\_subset[numeric\_vars] <- lapply(test\_subset[numeric\_vars], function(x) ifelse(is.na(x), mean(x, na.rm = TRUE), x))

> test\_subset[numeric\_vars] <- lapply(test\_subset[numeric\_vars], as.numeric)

predicted\_prices <- predict(price\_model, newdata = test\_subset)

> print(predicted\_prices)

1 2 3 4 5 6 7 8 9 10

35589.37 63222.57 266903.21 157856.39 240631.92 134710.73 187951.38 71714.16 244256.53 77179.47

11 12 13 14 15 16 17 18 19 20

109155.11 54357.02 152342.65 76979.63 128895.42 206917.67 214552.51 212286.52 180754.30 220027.07

I then compared the predicted values to the actual values:

> print(test\_subset$SalePrice)

[1] 82000 86000 232000 136905 181000 149900 163500 88000 240000 102000 135000 100000 165000 85000

[15] 119200 227000 203000 187500 160000 213490

> predictedvsactual<-(predicted\_prices-test\_subset$SalePrice)

> print(predictedvsactual)

1 2 3 4 5 6 7 8 9

-46410.634 -22777.426 34903.206 20951.385 59631.917 -15189.266 24451.383 -16285.840 4256.531

10 11 12 13 14 15 16 17 18

-24820.530 -25844.889 -45642.977 -12657.355 -8020.368 9695.423 -20082.331 11552.512 24786.517

19 20

20754.301 6537.072

> date()

[1] "Tue Jan 9 20:08:05 2024"

I then tested the Mean Absolute Percentage Error, MAPE, for the model.

> MAPE<-mean(abs(100\*predictedvsactual/test\_subset$SalePrice))

> MAPE

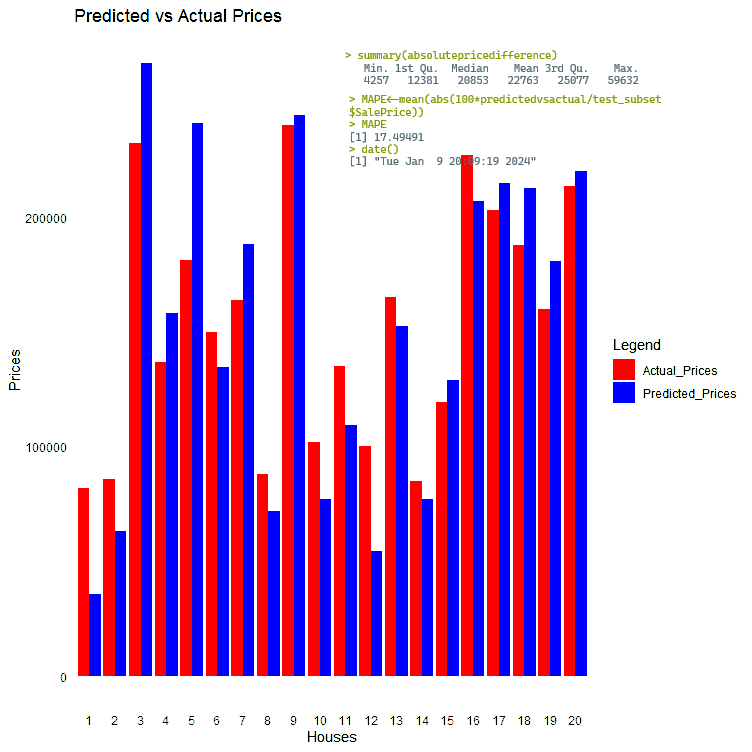
[1] 17.49491

> date()

[1] "Tue Jan 9 20:09:19 2024"

This means that the model’s predictions, on average, have an absolute error of 17.495%.

I also created a plot showing the predictions next to the actual values:



A model with an error rate of 17.5% did not seem like very useful model to me, so I tried to devise a better one. I ran a correlation test to see if I could find the variables that were the most strongly related to the sale price:

print(correlation\_results)

Variable Correlation PValue

cor YrSold -0.02892259 2.694132e-01

cor1 LotArea 0.26384335 1.123139e-24

cor2 OverallQual 0.79098160 2.185675e-313

cor3 OverallCond -0.07785589 2.912351e-03

cor4 YearBuilt 0.52289733 2.990229e-103

cor5 YearRemodAdd 0.50710097 3.164948e-96

cor6 TotalBsmtSF 0.61358055 9.484229e-152

cor7 GrLivArea 0.70862448 4.518034e-223

cor8 FullBath 0.56066376 1.236470e-121

cor9 HalfBath 0.28410768 1.650473e-28

cor10 BedroomAbvGr 0.16821315 9.927497e-11

cor11 KitchenAbvGr -0.13590737 1.860426e-07

cor12 TotRmsAbvGrd 0.53372316 2.772281e-108

cor13 GarageYrBlt 0.48636168 8.705128e-83

cor14 Fireplaces 0.46692884 6.141487e-80

cor15 GarageCars 0.64040920 2.498644e-169

cor16 GarageArea 0.62343144 5.265038e-158

cor17 WoodDeckSF 0.32441344 3.972217e-37

cor18 OpenPorchSF 0.31585623 3.493374e-35

cor19 MoSold 0.04643225 7.612758e-02

cor20 YrSold -0.02892259 2.694132e-01

‘OverallQuality’, ‘GarageArea’, ‘GarageCars’, ‘TotRmsAbvGrd’, ‘FullBath’, ‘GrLivArea’, ‘TotalBsmtSF’, ‘YearRemodAdd’, and ‘YearBuilt’, all have a correlation value above 0.50 and P-values below 0.05, indicating strong, and significant correlations. So, I created a linear regression model using only those variables:

> new\_price\_model<-lm(SalePrice~OverallQual+YearBuilt+YearRemodAdd+TotalBsmtSF+GrLivArea+FullBath+TotRmsAbvGrd+GarageCars+GarageArea, data = house\_train)

> summary(new\_price\_model)

Call:

lm(formula = SalePrice ~ OverallQual + YearBuilt + YearRemodAdd +

TotalBsmtSF + GrLivArea + FullBath + TotRmsAbvGrd + GarageCars +

GarageArea, data = house\_train)

Residuals:

Min 1Q Median 3Q Max

-359144 -18165 -1537 14467 271518

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -1.367e+06 1.481e+05 -9.227 < 2e-16 \*\*\*

OverallQual 1.744e+04 1.371e+03 12.725 < 2e-16 \*\*\*

YearBuilt 2.863e+02 5.621e+01 5.093 4.21e-07 \*\*\*

YearRemodAdd 3.692e+02 7.239e+01 5.099 4.08e-07 \*\*\*

TotalBsmtSF 3.934e+01 3.336e+00 11.791 < 2e-16 \*\*\*

GrLivArea 5.768e+01 4.763e+00 12.110 < 2e-16 \*\*\*

FullBath -1.028e+04 3.044e+03 -3.378 0.000759 \*\*\*

TotRmsAbvGrd 1.527e+03 1.269e+03 1.203 0.229121

GarageCars 7.665e+02 3.499e+03 0.219 0.826640

GarageArea 4.426e+01 1.186e+01 3.732 0.000201 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 35540 on 990 degrees of freedom

Multiple R-squared: 0.8059, Adjusted R-squared: 0.8041

F-statistic: 456.6 on 9 and 990 DF, p-value: < 2.2e-16

> date()

[1] "Wed Jan 10 13:26:42 2024"

The adjusted R-square value is only 0.8041 which is lower that the previous model, suggesting that is not going to be as good as the model that used all of the variables. I ran the model against the twenty testing values anyway:

|  |
| --- |
| newprediction<-predict(new\_price\_model, test\_subset)  > print(newprediction)  1 2 3 4 5 6 7 8 9 10 11 12 13  40066.40 63059.75 269077.08 195037.06 221336.02 115596.86 202838.08 90896.02 250995.90 109148.36 104216.93 95919.10 144602.86  14 15 16 17 18 19 20  115812.74 129037.50 225601.62 220684.65 221861.06 170038.65 219431.83  > date()  [1] "Wed Jan 10 13:29:58 2024" |
|  |
|  |

> newpricedifference<-test\_subset$SalePrice - newprediction

> MAPE2<-mean(abs(100\*newpricedifference/test\_subset$SalePrice))

> MAPE2

[1] 17.04127

> date()

[1] "Wed Jan 10 13:34:16 2024"

The MAPE dropped but only by 0.46% so this model isn’t really any better that using all the variables.

**SUMMARY**

I really enjoyed doing this assignment. I hoped to find a better way to predict sale price, but I could not. I only tried various linear regression models. Beyond the two listed in this paper, I also tried using only ‘OverallQuality’, the variable that had the strongest correlation, and various combinations of variables that were statistically significant, but none of the models I created performed noticeably better than using all 24 variables. I will continue to try, using various machine learning methods. I am sure that there must be a better way to predict the sale price of homes. (Edit, I did run a random forest algorithm and it was much better at predicting the sale price, I have included it after the references)

Overall, this course was great. It taught me a lot, and definitely showed me that I don’t know enough, which might be more important than the things it taught me. I will work hard to master the things that I need to learn.

Thank you for this class,

Michael Hiett

**References**

Fox, J., & Weisberg, S. (2019). *An R Companion to Applied Regression* (3rd ed.). Sage Publications.

Porras, E. M. (2022, December 5). *Linear regression in R tutorial*. <https://www.datacamp.com/tutorial/linear-regression-R>

*Research Guides: Linear regression in R: Linear Regression Hands on R tutorial*. (n.d.). <https://libguides.princeton.edu/R-linear_regression>

*Sign in - Google Accounts*. (n.d.-ag). <https://csuglobal.instructure.com/courses/83902/assignments/1689632?module_item_id=4370354>

**Random Forest Algorithm**

I wanted to see if I could find a better method of predicting the sale price. The linear regression models seemed to have pretty poor accuracy. I ran a random forest model and it was much better at predicting the accuracy. Here is the code:

> library(randomForest)

> house\_train<-MIS470HousingTraining\_1000x25\_

> house\_test<-MIS470housingtesting\_460x25\_

> set.seed(123)

> rf\_model<-randomForest(SalePrice~ ., data = house\_train)

> predictions <- predict(rf\_model, newdata = house\_test)

> predictions <- predict(rf\_model, newdata = test\_subset)

> print(predictions)

1 2 3 4 5 6 7 8 9 10 11 12 13

84682.68 89609.66 257107.26 155386.22 191949.01 128911.55 170612.10 88705.24 235741.54 125104.42 132913.05 125800.81 145388.00

14 15 16 17 18 19 20

109792.00 133462.89 220091.61 213677.86 207610.07 175790.49 192482.92

> absdifference<-abs(difference)

> summary(absdifference)

Min. 1st Qu. Median Mean 3rd Qu. Max.

705.2 6245.9 15026.7 13902.4 20993.1 25800.8

> MAPE2<-mean(abs(100\*difference/test\_subset$SalePrice))

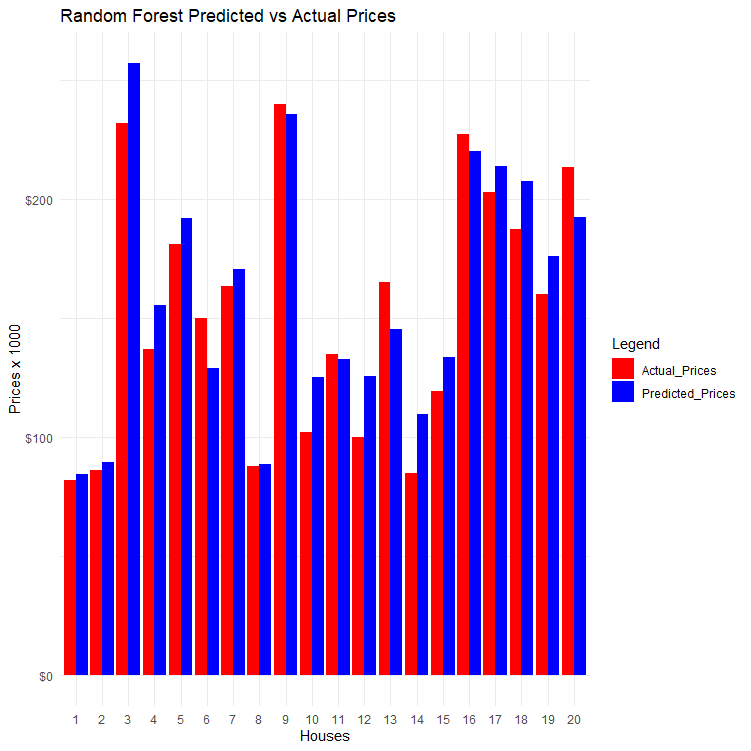
> MAPE2

[1] 10.02605

> date()

[1] "Sun Jan 14 17:04:27 2024"

**The Plot**

****

The Mean Absolute Percentage Error, or MAPE, dropped from 17.49491 to 10.02605. I calculated the R-Squared value as well and it also improved from 0.9182 to 0.9980478.

> rsquared <- 1 - (sum(house\_test$SalePrice - predictions)^2) / sum((house\_test$SalePrice - mean(house\_test$SalePrice))^2)

> print(rsquared)

[1] 0.9980478

date()

[1] "Sun Jan 14 17:40:15 2024"