Statistical Analysis - Final Assignment Part 2

Kshitij Mittal

2022-12-09

#In Continuation with the code for Part 1

Importing Relevant Packages

Creating a new data frame with fitted prices and residuals

```
data_mod4 = read.csv('/Users/kshitijmittal/Documents/UChicago Acad/01 Quarter 1/01 Stat Analysis/99 Fina
trans.lm12 = lm(I(sqrt(price))~new_neigh_level+bldclasscat+I(log(1+landsqft))+I(log(grosssqft))+localit

mod_preds = data.frame(data_mod4$date, data_mod4$quarter, data_mod4$price, (trans.lm12$fitted.values)^2
colnames(mod_preds) = c('date','quarter','price','fitted_price','residual')

mod_preds_f=mod_preds[(mod_preds$quarter=="2020_Q3"|mod_preds$quarter=="2020_Q4"),]
head(mod_preds_f)
```

```
##             date quarter   price fitted_price    residual
## 11858 2020-07-20 2020_Q3 1188000    1140067.2  -47932.783
## 11860 2020-10-15 2020_Q4  870000     867409.8   -2590.246
## 11861 2020-12-23 2020_Q4 1250000     690388.1 -559611.876
## 11862 2020-12-02 2020_Q4  805000     919264.6  114264.583
## 11866 2020-11-05 2020_Q4  740000     675450.3  -64549.654
## 11867 2020-10-07 2020_Q4  740000     852070.6  112070.646
```

Analyzing difference between Q3 2020 and Q4 2020 using our linear regression model

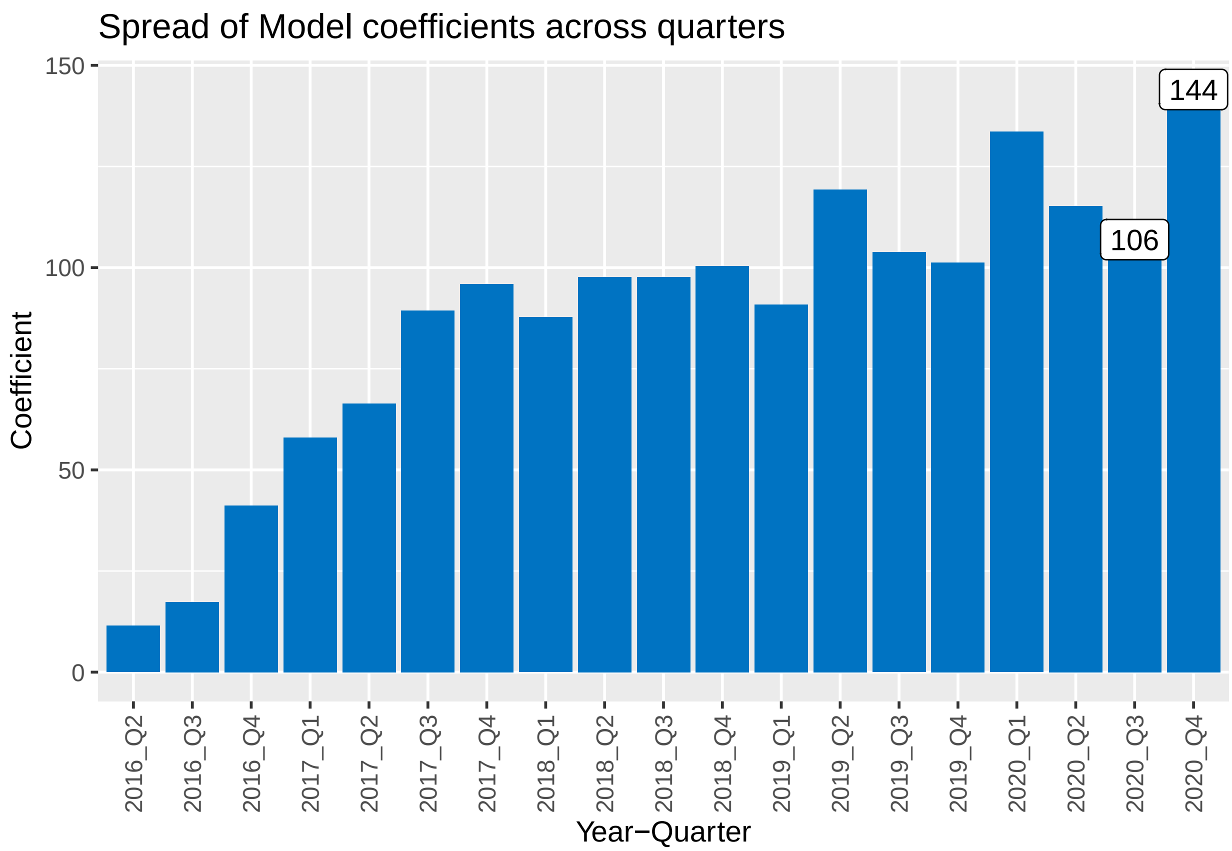
```
# Checking coefficients for our final model (trans.lm12)

q_coeffs=data.frame(trans.lm12$coefficients[23:41])
q_coeffs$quarter=as.factor(substr(rownames(q_coeffs),8,14))
colnames(q_coeffs)=c('coefficient','quarter')
q_coeffs
```

```
##                coefficient quarter
## quarter2016_Q2    11.46175 2016_Q2
## quarter2016_Q3    17.34296 2016_Q3
```

```
## quarter2016_Q4    41.18935 2016_Q4
## quarter2017_Q1    58.00057 2017_Q1
## quarter2017_Q2    66.43780 2017_Q2
## quarter2017_Q3    89.38824 2017_Q3
## quarter2017_Q4    95.93402 2017_Q4
## quarter2018_Q1    87.74432 2018_Q1
## quarter2018_Q2    97.69316 2018_Q2
## quarter2018_Q3    97.64674 2018_Q3
## quarter2018_Q4   100.40760 2018_Q4
## quarter2019_Q1    90.89884 2019_Q1
## quarter2019_Q2   119.31580 2019_Q2
## quarter2019_Q3   103.79556 2019_Q3
## quarter2019_Q4   101.24999 2019_Q4
## quarter2020_Q1   133.59325 2020_Q1
## quarter2020_Q2   115.24729 2020_Q2
## quarter2020_Q3   106.92373 2020_Q3
## quarter2020_Q4   144.03203 2020_Q4
```

```r
#plot(trans.lm12$coefficients[23:41])
ggplot(data = q_coeffs, aes(x=as.factor(quarter), y=coefficient)) + geom_bar(stat = "identity", fill="#0
  geom_label(data=q_coeffs %>% filter(quarter=="2020_Q3"|quarter=="2020_Q4"),aes(label=floor(coefficien
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

```
## Warning: Ignoring unknown parameters: y_nudge
```



Keeping Quarter 1 as the reference, we can see that coefficients for the model are subsequently

increasing as quarters pass. ## For the year 2020, the coefficient takes a sharp jump between Q3(106) to Q4(144)

This means that if all the other variables were controlled, a price for a house would jump between these two quarters

Simulating this test by identifying houses sold in Q3 2020, and predicting prices if only the quarter variable was changed

```r
houses_2020Q3=data_mod4[data_mod4$quarter=="2020_Q3",]
houses_2020Q3_dummy = houses_2020Q3
houses_2020Q3_dummy$quarter = "2020_Q4"

head(houses_2020Q3)
```

```
##            X neighborhood             bldclasscat taxclasscurr block lot
## 11858 141006   BATH BEACH 01 ONE FAMILY DWELLINGS            1  6371  60
## 11868  31106   BATH BEACH 01 ONE FAMILY DWELLINGS            1  6461 246
## 11869  33106   BATH BEACH 01 ONE FAMILY DWELLINGS            1  6464 113
## 11873  25916    BAY RIDGE 01 ONE FAMILY DWELLINGS            1  5839   3
## 11875  26216    BAY RIDGE 01 ONE FAMILY DWELLINGS            1  5851  25
## 11885  27516    BAY RIDGE 01 ONE FAMILY DWELLINGS            1  5865  47
##       bldclasscurr   zip landsqft grosssqft yrbuilt taxclasssale bldclasssale
## 11858           A9 11214     2417      2106    1930            1           A9
## 11868           A5 11214     1649       928    1945            1           A5
## 11869           A5 11214     1551      1320    1940            1           A5
## 11873           A9 11220     2574      1914    1925            1           A9
## 11875           A1 11220     2750      1650    1899            1           A1
## 11885           A5 11220     2080      1456    1930            1           A5
##         price       date     locality quarter quartf yearsl  new_neigh_level
## 11858 1188000 2020-07-20 Southwestern 2020_Q3     Q3   2020 new_neigh_level4
## 11868  750000 2020-08-26 Southwestern 2020_Q3     Q3   2020 new_neigh_level4
## 11869  800000 2020-08-07 Southwestern 2020_Q3     Q3   2020 new_neigh_level4
## 11873  350000 2020-09-23 Southwestern 2020_Q3     Q3   2020 new_neigh_level6
## 11875  700000 2020-09-15 Southwestern 2020_Q3     Q3   2020 new_neigh_level6
## 11885  975000 2020-07-07 Southwestern 2020_Q3     Q3   2020 new_neigh_level6
##        new_bld_sale
## 11858 bld_sale_Alow
## 11868 bld_sale_Alow
## 11869 bld_sale_Alow
## 11873 bld_sale_Alow
## 11875 bld_sale_Alow
## 11885 bld_sale_Alow
```

```r
head(houses_2020Q3_dummy)
```

```
##            X neighborhood             bldclasscat taxclasscurr block lot
## 11858 141006   BATH BEACH 01 ONE FAMILY DWELLINGS            1  6371  60
## 11868  31106   BATH BEACH 01 ONE FAMILY DWELLINGS            1  6461 246
## 11869  33106   BATH BEACH 01 ONE FAMILY DWELLINGS            1  6464 113
```

```
## 11873  25916    BAY RIDGE 01 ONE FAMILY DWELLINGS            1  5839   3
## 11875  26216    BAY RIDGE 01 ONE FAMILY DWELLINGS            1  5851  25
## 11885  27516    BAY RIDGE 01 ONE FAMILY DWELLINGS            1  5865  47
##       bldclasscurr   zip landsqft grosssqft yrbuilt taxclasssale bldclasssale
## 11858           A9 11214     2417      2106    1930            1           A9
## 11868           A5 11214     1649       928    1945            1           A5
## 11869           A5 11214     1551      1320    1940            1           A5
## 11873           A9 11220     2574      1914    1925            1           A9
## 11875           A1 11220     2750      1650    1899            1           A1
## 11885           A5 11220     2080      1456    1930            1           A5
##         price       date     locality quarter quartf yearsl  new_neigh_level
## 11858 1188000 2020-07-20 Southwestern 2020_Q4     Q3   2020 new_neigh_level4
## 11868  750000 2020-08-26 Southwestern 2020_Q4     Q3   2020 new_neigh_level4
## 11869  800000 2020-08-07 Southwestern 2020_Q4     Q3   2020 new_neigh_level4
## 11873  350000 2020-09-23 Southwestern 2020_Q4     Q3   2020 new_neigh_level6
## 11875  700000 2020-09-15 Southwestern 2020_Q4     Q3   2020 new_neigh_level6
## 11885  975000 2020-07-07 Southwestern 2020_Q4     Q3   2020 new_neigh_level6
##        new_bld_sale
## 11858 bld_sale_Alow
## 11868 bld_sale_Alow
## 11869 bld_sale_Alow
## 11873 bld_sale_Alow
## 11875 bld_sale_Alow
## 11885 bld_sale_Alow
```

Making the prediction on this new dummy data set

```
dummy_pred_2020Q3=predict(trans.lm12, newdata = houses_2020Q3_dummy)
dummy_pred_2020Q3
```

```
##     11858     11868     11869     11873     11875     11885     11897     11899
## 1104.8476  822.4574  942.3246 1160.5222 1110.3083 1065.6915 1197.7168 1119.9171
##     11907     11914     11915     11916     11932     11933     11934     11935
##  912.4514 1265.7584 1092.4588 1092.4588 1115.8943 1115.8943 1078.0482 1455.2265
##     11936     11956     11967     11970     11973     11979     11981     11984
## 1054.9254 1212.2425 1028.9192  965.9521 1131.4145  980.3969  982.8768 1243.0353
##     11993     11994     11998     11999     12013     12017     12019     12020
## 1151.3809 1105.7958 1226.3085 1106.4109  917.0761  943.1998 1180.1417 1025.0892
##     12022     12024     12025     12027     12030     12034     12041     12048
## 1149.4669 1059.3479 1086.3648  930.8096  861.6783  848.7440  951.2956 1071.1000
##     12059     12061     12073     12076     12077     12078     12083     12093
##  928.8081 1100.6118  915.0945 1113.7076  915.0945 1032.9986 1015.7477 1001.5610
##     12100     12116     12117     12126     12134     12135     12137     12144
## 1440.3950 1848.9191 1836.7113  806.1047  673.9687  673.9687  675.6505  673.9763
##     12145     12148     12153     12163     12171     12187     12188     12189
##  687.3675  683.6197 1014.8365 1077.5327  667.5483 1039.0842  683.9301  653.8807
##     12194     12200     12213     12219     12230     12238     12242     12247
##  881.8918  703.9875  564.8877  882.4126  707.1302  630.5810  763.5729 1882.5603
##     12250     12252     12259     12270     12271     12287     12290     12292
## 1491.0568 1657.2570 1333.5183  736.2189  795.3589 1050.3998 1258.7599 1365.1239
##     12295     12300     12301     12311     12316     12325     12326     12333
```

```
## 1042.1031  909.3594  752.0002  814.6409  664.1922  716.0391 1673.6095  923.5659
##     12335     12352     12358     12359     12361     12362     12367     12385
## 1037.9185 1124.3836 1160.1703 1160.1703 1025.3267 1043.4695  947.2045  679.4309
##     12387     12388     12391     12395     12396     12398     12399     12428
##  716.1013  679.4309  734.3695  678.8283  678.8283  678.8283  678.8283  849.6864
##     12431     12432     12433     12447     12460     12461     12464     12465
##  741.3067  726.0680  623.8116  673.2672 1090.6351 1106.5766 1066.8783 1267.0445
##     12466     12477     12482     12483     12487     12489     12495     12501
## 1226.8355 1342.0441 1287.9155 1210.2520 1219.6047 1220.1216 1119.5190 1337.1960
##     12503     12507     12508     12511     12513     12514     12522     12529
##  811.2164  716.1144  803.0315  815.0406  796.4281  810.3640  847.7947  754.8831
##     12531     12535     12539     12540     12541     12548     12566     12570
##  732.0466  758.8759  774.3959  788.8817  898.2409  759.3717  762.8225  969.8769
##     12577     12584     12592     12593     12594     12595     12598     12599
##  774.6382  871.1007  780.0143  855.3689 1027.0260  805.1991  861.4842  713.0630
##     12607     12609     12616     12628     12632     12633     12637     12638
##  805.1991  793.1737  921.2656  742.6851  778.4104  851.2295  804.7033  766.9663
##     12639     12641     12657     12658     12665     12679     12683     12689
##  866.6963  921.8679 1359.3071 1272.2189  834.7130  789.4156  677.2861  768.4974
##     12698     12706     12710     12726     12729     12740     12746     12748
##  798.6964  914.1349  773.3461  524.5989  899.6446  871.4152  509.6455  717.0318
##     12753     12754     12765     12774     12782     12783     12784     12785
##  911.6539  739.0645  888.0227  953.9735 1304.5824  891.6113 1303.5226  694.0056
##     12789     12791     12794     12799     12800     12802     12812     12813
##  811.5681  811.1041  849.7189  780.0980  903.3287  769.6894  970.1839 1073.5203
##     12826     12827     12828     12838     12839     12840     12842     12847
## 1024.5022  880.8557 1077.9267  969.6855 1051.8042  969.6855  964.6660  993.8700
##     12848     12849     12850     12851     12855     12859     12861     12862
## 1029.4498  942.8257  942.8257 1009.8476 1050.2432  930.6900  999.1309  932.8714
##     12869     12870     12872     12875     12877     12884     12886     12896
##  969.9256  935.9288  922.2608 1050.5788  951.7811 1033.6018 1349.2005 1169.6369
##     12899     12904     12922     12923     12932     12933     12934     12935
## 1198.7982 1235.8378  801.2354  932.3635  865.5353  895.4142  925.1872  904.8204
##     12940     12950     12960     12984     12986     12987     12994     12995
##  792.3679  894.4936  827.2574  789.2187  897.6464  967.6828  789.5281  749.3400
##     13012     13017     13018     13023     13024     13025     13026     13028
##  959.9904  800.9793  854.5920  819.1918  822.9328  911.6941  698.5182  784.8707
##     13035     13037     13042     13046     13067     13068     13074     13075
##  998.2363 1129.7964 1045.7252 1206.3116 1055.1046 1051.4708 1121.2405 1121.8420
##     13082     13085     13087     13089     13091     13094     13098     13099
## 1027.6331  974.5009  948.0254  980.4505 1237.2439 1253.6315 1189.2862 1122.7825
##     13101     13108     13119     13128     13133     13137     13142     13146
## 1222.5061 1128.1121 1347.1623 1423.8310 1230.9710 1286.1949 1165.3309 1133.3971
##     13156     13157     13159     13162     13168     13176     13182     13187
## 1047.1567 1136.3155 1449.1537 1192.9357 1092.2491 1310.2098 1134.7066 1146.1969
##     13196     13197     13198     13200     13201     13202     13209     13234
## 1496.7277 1437.6437 1361.2471 1487.5219 1790.7854 1175.4446  690.8967  727.3173
##     13238     13244     13248     13257     13258     13267     13271     13275
##  743.5113  790.0587  784.5899  783.9751  779.7343  770.2754  770.2754  829.8862
##     13276     13286     13290     13299     13302     13303     13306     13311
##  788.9883 1513.2829 1898.5624 1347.9057 1190.4510 1465.7288 1465.4373 1465.4373
##     13319     13320     13321     13323     13343     13344     13348     13353
##  914.8529 1047.4081 1017.0317  945.7348  703.1896 1020.2846  963.3697  969.4120
##     13356     13362     13363     13365     13384     13387     13390     13391
```

```
## 866.2019  938.4285 1100.7228  930.8326  807.4438  806.1047  806.1047  806.1047
##     13392     13393     13394     13403     13404     13410     13415     13416
## 806.1047  807.0637  807.0637 1016.0144  879.3186 1293.8466 1459.7808 1380.4341
##     13420     13423     13428     13429     13430     13432
## 1596.2629 1344.0374 1130.9004 1210.6798 1157.6725 1217.6711
```

Calculating the price deltas between 2020 Q3 and 2020 Q4

```
mod_preds_2020Q3=mod_preds_f[mod_preds_f$quarter=="2020_Q3",]
mod_preds_2020Q3$fitted_price_2020Q4 = (dummy_pred_2020Q3)^2
mod_preds_2020Q3$price_delta = mod_preds_2020Q3$fitted_price_2020Q4 - mod_preds_2020Q3$fitted_price
head(mod_preds_2020Q3)
```

```
##             date quarter   price fitted_price   residual fitted_price_2020Q4
## 11858 2020-07-20 2020_Q3 1188000    1140067.2  -47932.78           1220688.2
## 11868 2020-08-26 2020_Q3  750000     616773.2 -133226.75            676436.2
## 11869 2020-08-07 2020_Q3  800000     819416.6   19416.58            887975.7
## 11873 2020-09-23 2020_Q3  350000    1262058.9  912058.89           1346811.9
## 11875 2020-09-15 2020_Q3  700000    1151758.2  451758.20           1232784.5
## 11885 2020-07-07 2020_Q3  975000    1057983.4   82983.37           1135698.4
##       price_delta
## 11858    80621.02
## 11868    59662.98
## 11869    68559.11
## 11873    84753.00
## 11875    81026.29
## 11885    77714.98
```

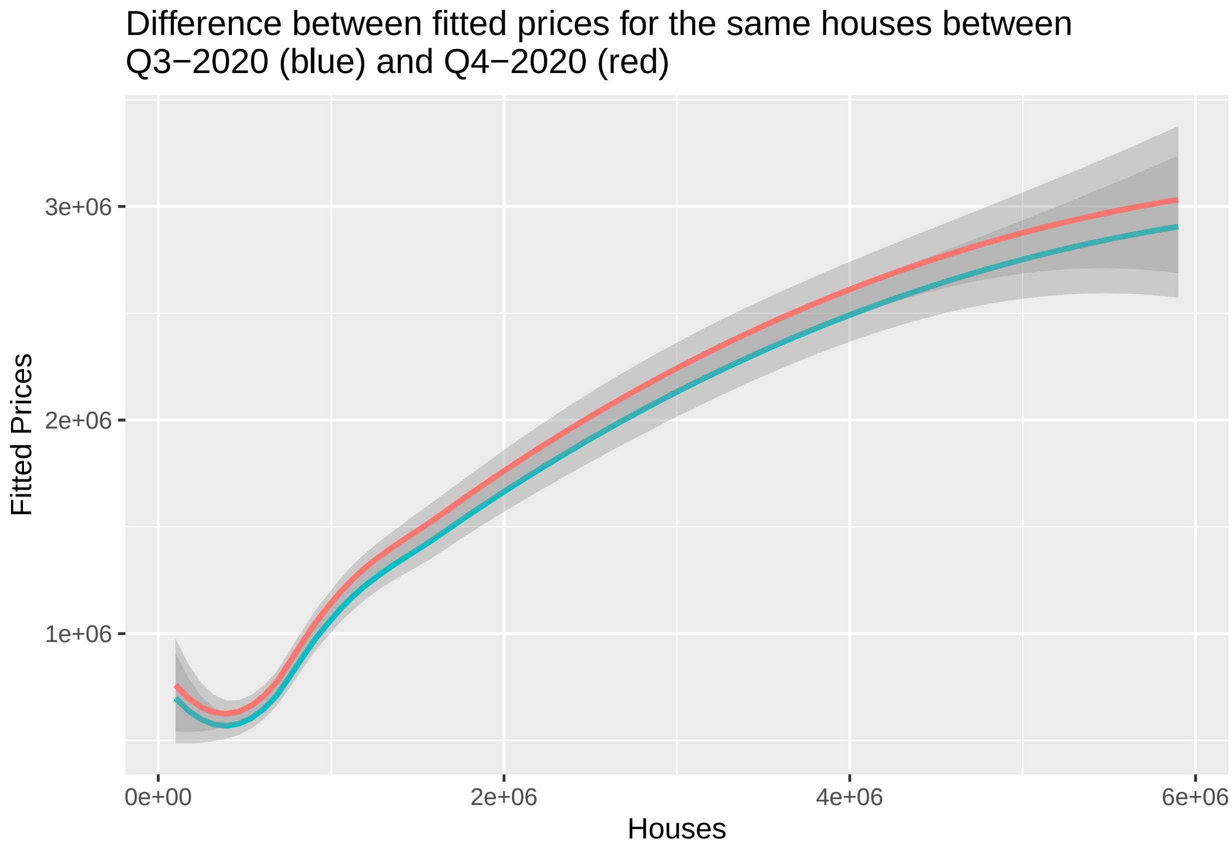
```
summary(mod_preds_f[mod_preds_f$quarter=="2020_Q4",]$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  135000  639500  843000 1113344 1232500 6500000
```

Plotting the increase in prices from Q3 2020 to Q4 2024

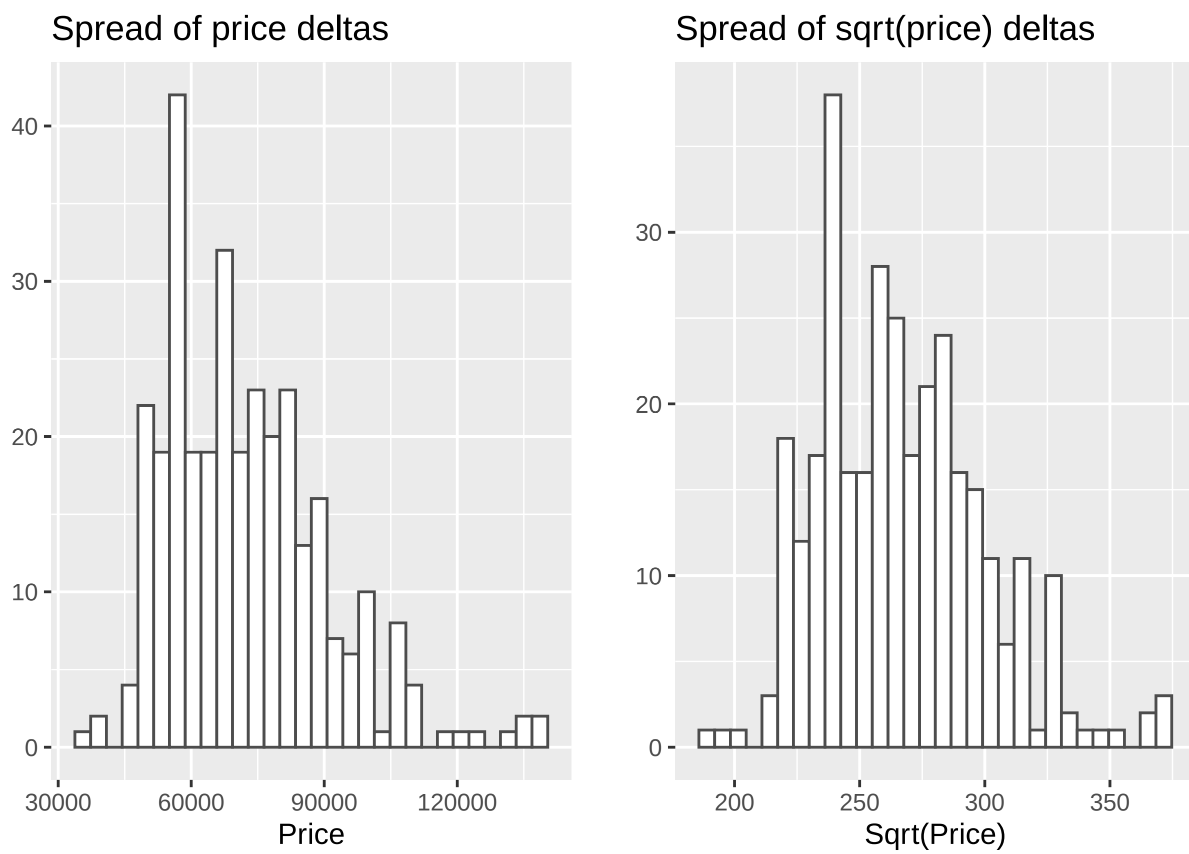
```
ggplot() +
  geom_line(data = mod_preds_2020Q3, mapping = aes(x=price, y=fitted_price, color="red")) +
  geom_line(data = mod_preds_2020Q3, mapping = aes(x=price, y=fitted_price_2020Q4, color="blue")) +
  theme(legend.position = "none") +
  ggtitle("Difference between fitted prices for the same houses between \nQ3-2020 (blue) and Q4-2020 (re
```


```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Analyzing the spread of price deltas

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



#Calculating 95% Confidence Intervals for Price Delta

```
## [1] 70358.22
```

```
## [1] 74365.55
```

Further Validation

First comparing the prices for Q3 2020 and Q4 2020 from original data

```
summary(data_mod4[data_mod4$quarter=="2020_Q3",]$price)
```

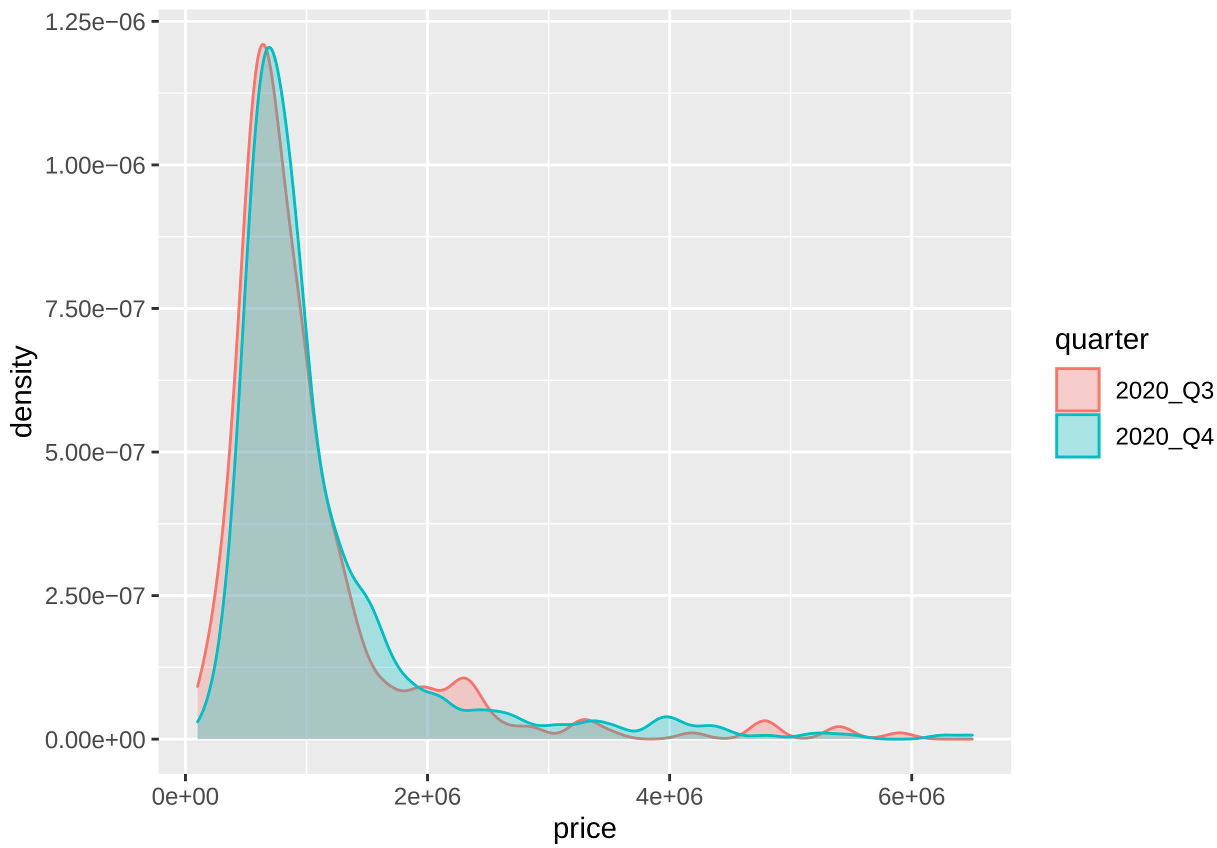
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  100000  576250  780000 1020542 1117500 5900000
```

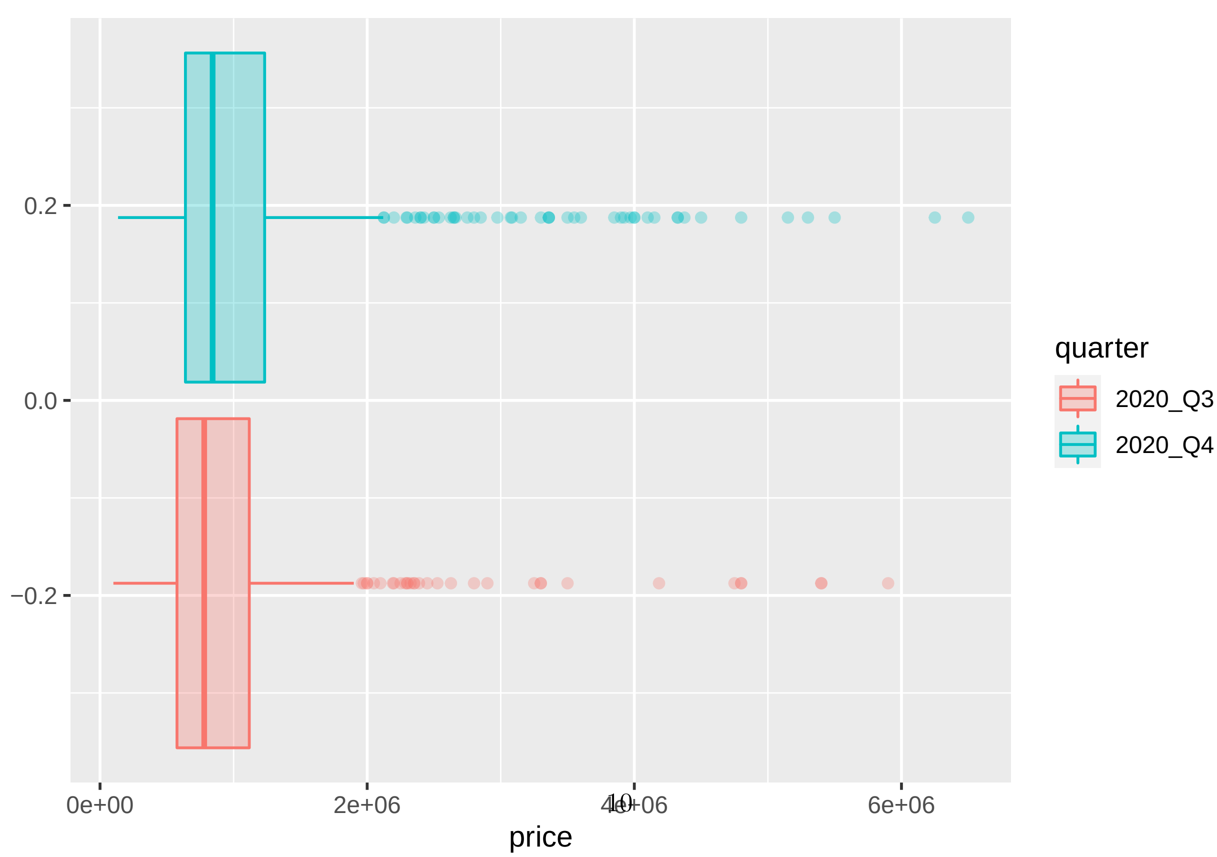
```
summary(data_mod4[data_mod4$quarter=="2020_Q4",]$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  135000  639500  843000 1113344 1232500 6500000
```

We can see that there is some movement in price between 2020_Q3 and 2020_Q4

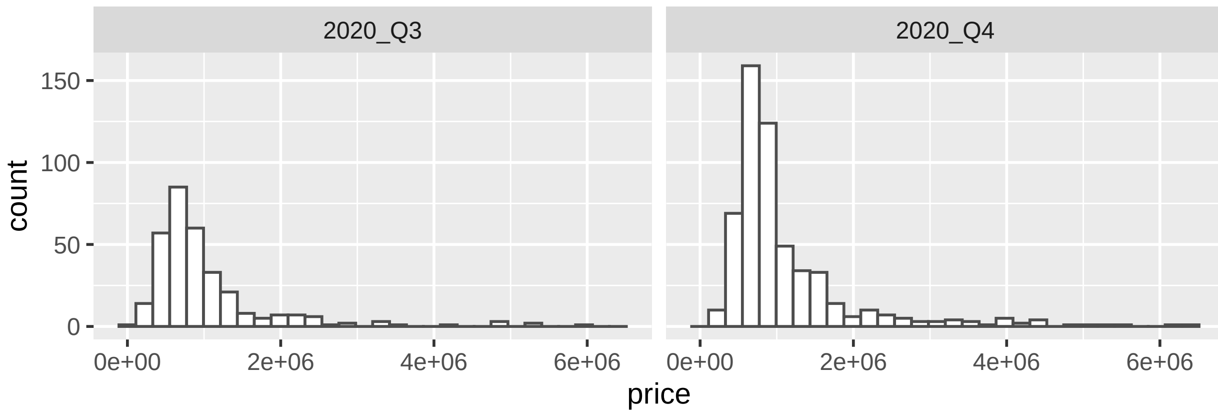
Plotting these prices

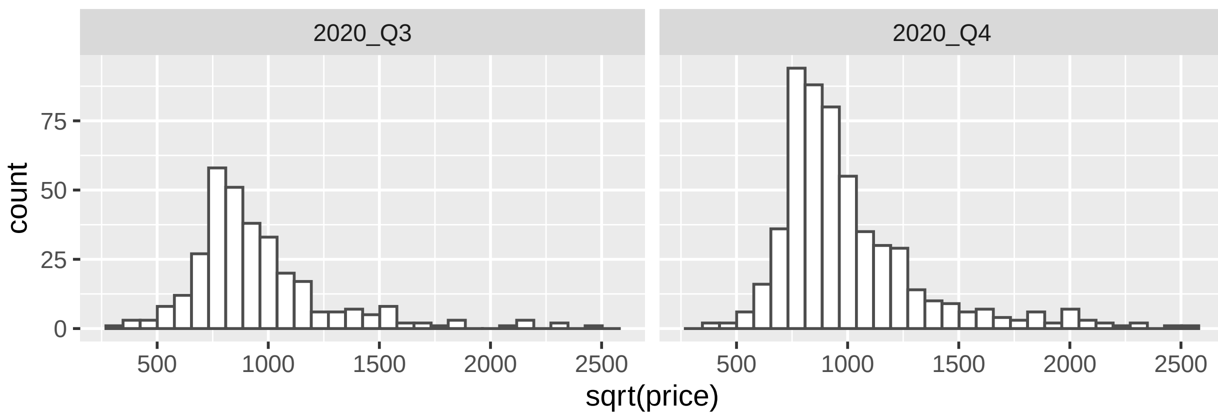




Plotting histograms for these prices (and their sqrt transformations)

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





Two-sample t-test

Comparing the difference between actual housing prices of Q3 2020 and Q4 2020 using Welch's T-Test

```r
t.test(price~quarter, data = mod_preds_f)
```

```
##
##  Welch Two Sample t-test
##
## data:  price by quarter
## t = -1.5565, df = 687.26, p-value = 0.12
## alternative hypothesis: true difference in means between group 2020_Q3 and group 2020_Q4 is not equal
## 95 percent confidence interval:
##  -209864.8   24260.4
## sample estimates:
## mean in group 2020_Q3 mean in group 2020_Q4
##               1020542               1113344
```

When comparing prices directly, we were getting a p-value>0.01. This made us unable to reject the null hypothesis that there is a statistically significant difference between prices along the two quarters.

Comparing the difference between sqrt(housing prices) of Q3 2020 and Q4 2020 using Welch's T-Test

```
t.test(sqrt(price)~quarter, data = mod_preds_f)
```

```
##
##  Welch Two Sample t-test
##
## data:  sqrt(price) by quarter
## t = -2.0166, df = 665.98, p-value = 0.04414
## alternative hypothesis: true difference in means between group 2020_Q3 and group 2020_Q4 is not equal
## 95 percent confidence interval:
##  -91.565089  -1.220377
## sample estimates:
## mean in group 2020_Q3 mean in group 2020_Q4
##              956.4552             1002.8479
```

However, when we compare the sqrt transformation of prices across two quarters, we do see a statistically significant difference (p-value = 0.044). This became an essential insight, as we are using our model to predict sqrt transformation of prices. In later steps we are squaring them to a price metric for gauging deltas.