Assignment 08 Part 2- Regressions

David Pahmer

2022-05-15

Using your skills in statistical correlation, multiple regression, and R programming, you are interested in the following variables: Sale Price and several other possible predictors.

```
library(readxl)
library(plyr)
library(dplyr)
```

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

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

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

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

```
setwd("C:/users/pahme/onedrive/documents/github/dsc520")
housingdf <- read_xlsx("data/week-7-housing.xlsx")
```

Explain any transformations or modifications you made to the dataset After examining the various fields, selecting those that I thought relevant, I modified them as follows:

- consolidate the bathrooms into one variable
- rename the sale date variable to sale_date (avoids problems)
- same for sale price
- make a latest build variable that takes the later of renovation or build year
- extract the sale year from sale date to aggregate the sales into bins of a year

```
housingdf$all_bathrooms <- housingdf$bath_full_count+housingdf$bath_3qtr_count+
  housingdf$bath_half_count/2

names(housingdf)[names(housingdf)=='Sale Date'] <- 'sale_date'
```

```
names(housingdf)[names(housingdf)=='Sale Price'] <- 'sale_price'

housingdf$last_build <- housingdf$year_built
housingdf$last_build[housingdf$year_renovated > 0] <-
  housingdf$year_renovated[housingdf$year_renovated > 0]

housingdf$sale_year <- format(as.Date(housingdf$sale_date,
                                      format="%d%m%Y"), "%Y")
housingdf$sale_year <- strtoi(housingdf$sale_year) # because we want it as a number not a string
```

Create two variables; one that will contain the variables Sale Price and Square Foot of Lot (same variables used from previous assignment on simple regression) and one that will contain Sale Price and several additional predictors of your choice. Explain the basis for your additional predictor selections.

It seems that in addition to the property size, the sale price ought to depend on the year sold, as housing prices change over time even without any actual change to the property. Since the dataset doesn't have a field with year sold, I made one from the sale date, and not as a factor variable but as a number. Likewise, number of bedrooms and bathrooms would affect price, although there is no field for bathrooms, so I needed to create one to capture the bathroom value in one field. Furthermore, the size of the building itself ought to be a factor. Finally, the year that the house was built or renovated should possibly be relevant. If I understood the building grade, sale reason, sale warning, property type, current use codes- those might be important, but I didn't.

```
simple_reg <- lm(sale_price ~ sq_ft_lot, data=housingdf)
multi_reg <- lm(sale_price ~ sq_ft_lot + square_feet_total_living + sale_year + bedrooms + all_bathrooms
```

```
summary(simple_reg)
```

Execute a summary() function on two variables defined in the previous step to compare the model results. What are the R2 and Adjusted R2 statistics? Explain what these results tell you about the overall model. Did the inclusion of the additional predictors help explain any large variations found in Sale Price?

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot, data = housingdf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2016064  -194842   -63293    91565  3735109
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.418e+05  3.800e+03  168.90   <2e-16 ***
## sq_ft_lot   8.510e-01  6.217e-02   13.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 401500 on 12863 degrees of freedom
## Multiple R-squared:  0.01435,    Adjusted R-squared:  0.01428
## F-statistic: 187.3 on 1 and 12863 DF,  p-value: < 2.2e-16
```

```
summary(multi_reg)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot + square_feet_total_living +
##     sale_year + bedrooms + all_bathrooms, data = housingdf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1962490  -115652   -38445    43236  3761586
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.029e+07  1.954e+06  -5.266 1.42e-07 ***
## sq_ft_lot                 7.295e-02  5.788e-02   1.260   0.2075
## square_feet_total_living  1.914e+02  5.143e+00  37.217  < 2e-16 ***
## sale_year                 5.233e+03  9.718e+02   5.385 7.39e-08 ***
## bedrooms                 -2.702e+04  4.539e+03  -5.953 2.71e-09 ***
## all_bathrooms             1.193e+04  6.393e+03   1.866   0.0621 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 359300 on 12859 degrees of freedom
## Multiple R-squared:  0.2107, Adjusted R-squared:  0.2104
## F-statistic: 686.5 on 5 and 12859 DF,  p-value: < 2.2e-16
```

The R-squared for the simple regression showed .014, implying a correlation of 0.12 which is pretty low, although the F value is much larger than 1 with a sufficiently low p-value to imply that the result is statistically significant.
However, the multiple regression, using the additional variables, gives an R-squared of .21, implying a correlation of 0.46, which is much better, and an F value over three times the first one, also with $p < .001$. This indicates that the additional variables included in our model improved the model, so we can better predict the sale price from these factors.

```
library(lm.beta)
lm.beta(multi_reg)
```

Considering the parameters of the multiple regression model you have created. What are the standardized betas for each parameter and what do the values indicate?

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot + square_feet_total_living +
##     sale_year + bedrooms + all_bathrooms, data = housingdf)
##
## Standardized Coefficients::
##              (Intercept)                sq_ft_lot square_feet_total_living
##                       NA               0.01027046               0.46850763
##                sale_year                 bedrooms            all_bathrooms
##               0.04224581              -0.05853959               0.02209376
```

The standardized betas provide each variable's coefficient adjusted to the same scale for purposes of comparing the predictive effects of each relative to the others. So, we can rank the variables in order of contribution toward the result:

1. Sq ft Total Living (.469)
2. bedrooms (-.059)
3. sale year (.042)
4. bathrooms (0.22)
5. sq ft lot (.01)

```
confint(multi_reg)
```

Calculate the confidence intervals for the parameters in your model and explain what the results indicate.

```
##                                 2.5 %        97.5 %
## (Intercept)              -1.411642e+07 -6.457556e+06
## sq_ft_lot                -4.049656e-02  1.863928e-01
## square_feet_total_living  1.813238e+02  2.014854e+02
## sale_year                 3.327763e+03  7.137423e+03
## bedrooms                 -3.591635e+04 -1.812216e+04
## all_bathrooms            -6.021146e+02  2.445968e+04
```

Looking at the confidence interval for each variable, we can see that for both sq ft lot and bathrooms the interval crosses 0, indicating that these are poor predictors of sale price (which was somewhat known from above), and that sale year seems to be uniformly positively correlated, but with a wide range so there is great variability. Number of bedrooms seems to be a better predictor with a smaller interval and all positive, while the strongest predictor is clearly total living sq ft, with the smallest interval. This was already indicated above.

```
anova(simple_reg, multi_reg)
```

Assess the improvement of the new model compared to your original model (simple regression model) by testing whether this change is significant by performing an analysis of variance.

```
## Analysis of Variance Table
##
## Model 1: sale_price ~ sq_ft_lot
## Model 2: sale_price ~ sq_ft_lot + square_feet_total_living + sale_year +
##     bedrooms + all_bathrooms
##   Res.Df        RSS Df Sum of Sq      F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12859 1.6604e+15  4  4.13e+14 799.62 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As suspected, the ANOVA shows a marked improvement in the multiple regression over the simple regression (especially since we see that the sq ft total variable is a relatively lousy predictor) with a very low p-value (<.001). Additionally, the large F-value ought to indicate something about the improvement in the model but I don't know what.

```
resid <- data.frame(resid(multi_reg))
```

Perform casewise diagnostics to identify outliers and/or influential cases, storing each function's output in a dataframe assigned to a unique variable name. There are several other casewise diagnostic tools, but we will get to those soon.

```
stand.resid <- data.frame(rstandard(multi_reg))
large.stresid <-  stand.resid > 2 | stand.resid < -2
```

Calculate the standardized residuals using the appropriate command, specifying those that are +-2, storing the results of large residuals in a variable you create. We now have standardized residuals, with large ones identified. For this purpose we are setting the cutoff for a large residual at ± 2

```
sum(large.stresid)
```

Use the appropriate function to show the sum of large residuals.

```
## [1] 321
```

Just to check, we expect the number of outliers to be about 5% of the observations, so let's check the percentage of outliers:

```
sum(large.stresid)*100/nrow(stand.resid)
```

```
## [1] 2.495142
```

which is actually half of that, so I wonder if I should be suspicious...

Which specific variables have large residuals (only cases that evaluate as TRUE)?
I would have to run the simple regression on each variable individually, right? Otherwise I don't know how.

```
lev <- hatvalues(multi_reg)
cook <- cooks.distance(multi_reg)
cov.ratio <- covratio(multi_reg)
```

Investigate further by calculating the leverage, cooks distance, and covariance ratios. Comment on all cases that are problematics.

For this we look at three statistics: Leverage (hat values), Cook's distance, and covariance ratios. For the hat values, we first need the average leverage which is k+1 /N , so in our case that is five predictor variables, and 12865 observations, giving an average of .00047, so for undue influence we look for hat values greater than three times that, or .0014.
Counting the number of cases whose hat values are greater than that we have 319 which is similar to the number of large residuals.
To use Cook's distance, we look for values greater than 1. sum(cook $>$ 1) which gives 0 which shows none to exert undue influence on the model.
For the covariance ratios, we need the lower and upper bounds: Since we have k=5, the lower bound is 1-18/12865, which is 0.9986009, and the upper bound is 1+18/12865, which is 1.0013991. So we check how many are outliers:

sum(cov.ratio < (1-18/12865) | cov.ratio > (1+18/12865)) , which gives us 684

```
library(car)
```

Perform the necessary calculations to assess the assumption of independence and state if the condition is met or not.

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

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

```
dwt(multi_reg)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.7307487      0.538501       0
##  Alternative hypothesis: rho != 0
```

Hmmm.... so it seems not. We were looking for values between 1 and 3, but we got .54
This suggests that we have not met the condition of independence, although we would have to dig in to locate the dependence...

```
vif(multi_reg)
```

Perform the necessary calculations to assess the assumption of no multicollinearity and state if the condition is met or not.

```
##                sq_ft_lot square_feet_total_living                sale_year
##                 1.081679                 2.581645                 1.002824
##                 bedrooms            all_bathrooms
##                 1.575536                 2.283971
```

```
1 / vif(multi_reg)
```

```
##                sq_ft_lot square_feet_total_living                sale_year
##                0.9244885                0.3873499                0.9971837
##                 bedrooms            all_bathrooms
##                0.6347045                0.4378340
```

```
mean(vif(multi_reg))
```

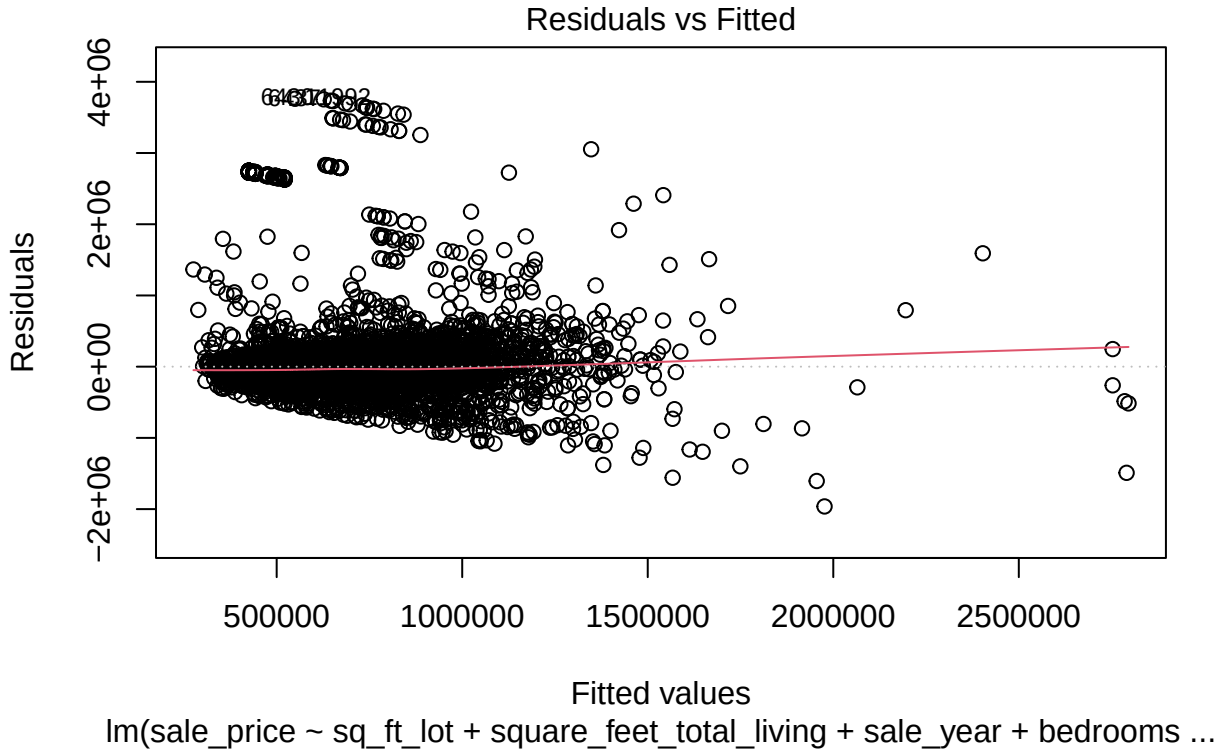
```
## [1] 1.705131
```

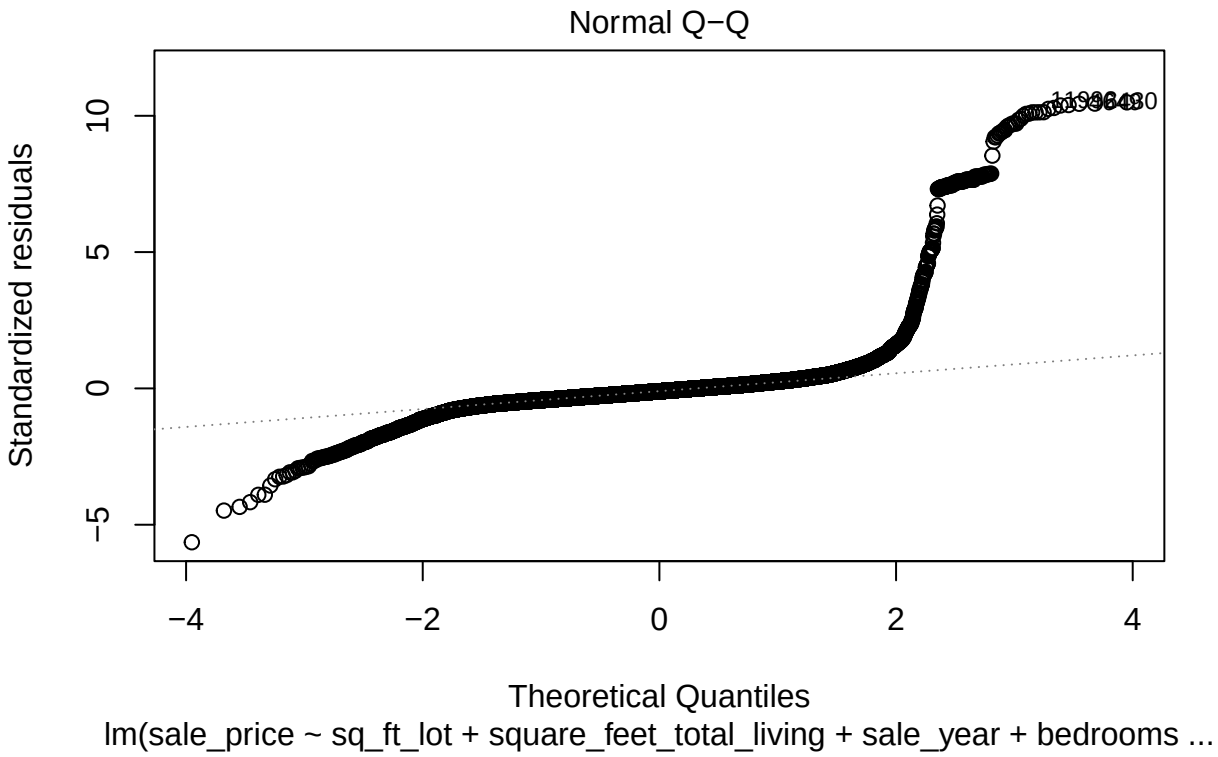
We are looking for several things here:
Highest VIF should be below 10. (It is.)
Tolerance (1/VIF) should be below 0.1. It is. (Actually, I don't understand how this is considered a separate consideration- it is identical to the first!)
Mean VIF should not be substantially greater than 1. (It's 1.7, so is that substantially greater than 1? Maybe.)

So it looks like we are avoiding multicolinearity.

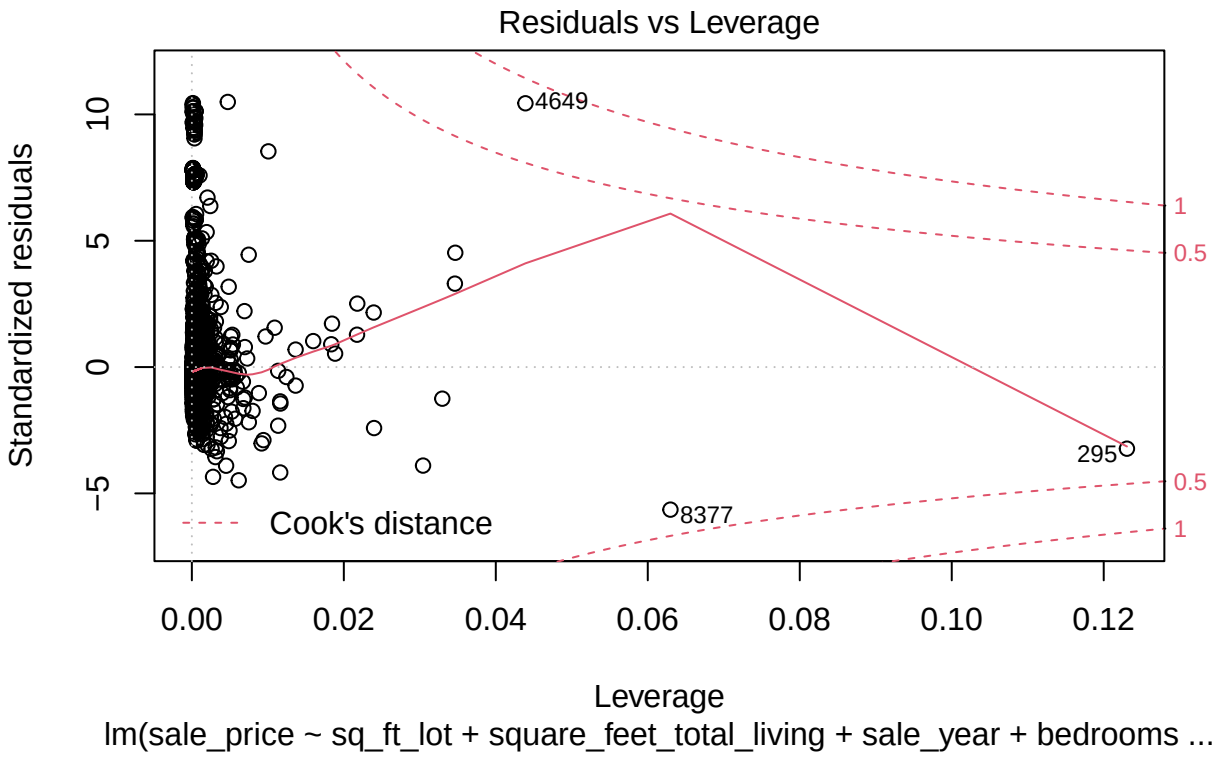
```
plot(multi_reg)
```

Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

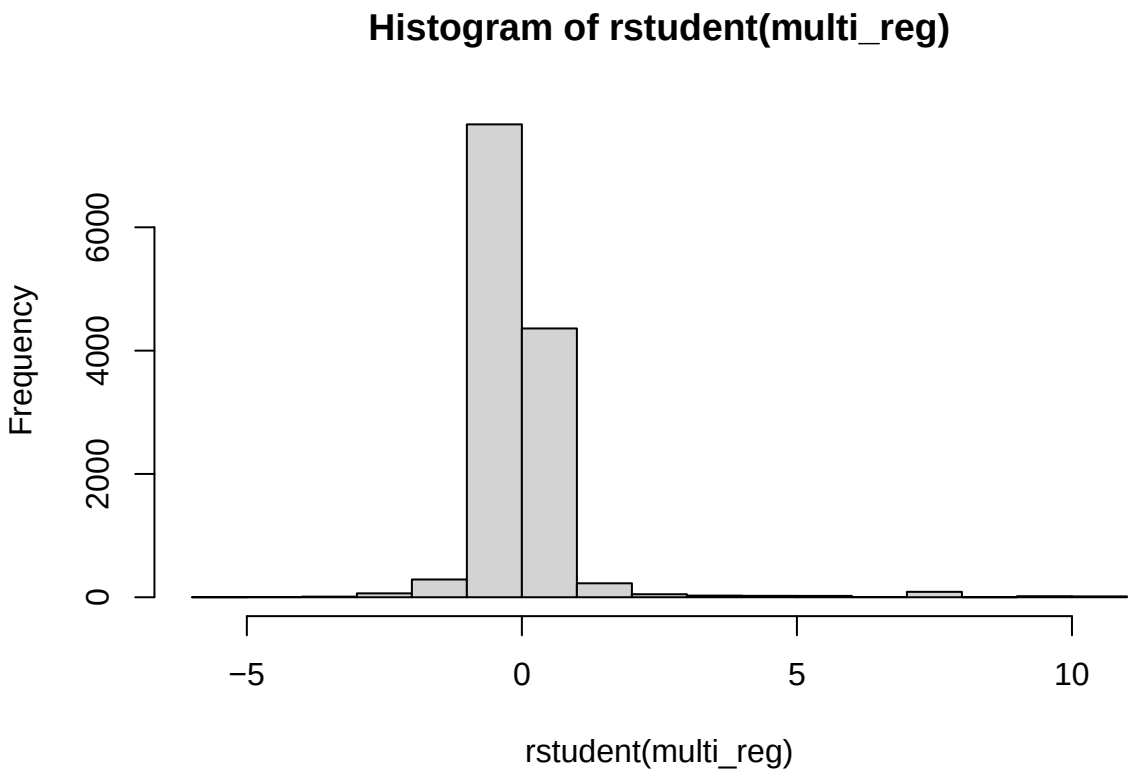



Normal Q−Q
Standardized residuals
10
5
0
−5
−4
−2
0
2
4
Theoretical Quantiles
lm(sale_price ~ sq_ft_lot + square_feet_total_living + sale_year + bedrooms ...


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
500000
1000000
1500000
2000000
2500000
Fitted values
lm(sale_price ~ sq_ft_lot + square_feet_total_living + sale_year + bedrooms ...




```
hist(rstudent(multi_reg))
```



For the Residuals scatterplot, it isn't quite like a shotgun, so it's suspicious, and possibly not homoscedastic. This does not necessarily invalidate the predictive value of the model though.
The Q-Q plot clearly shows very heavy tails, and it looks like a logistic curve. This might suggest that for the very low priced, or very high priced homes- the model doesn't predict well.
The spread - location plot shows a too-steep line, indicating a problem of heteroscedasticity, as we have already seen.
The residuals - leverage plot shows no observations exerting undue influence on the model, as we saw above.
The histogram of residuals, plotted using studentized residuals, shows either heavy-tailing, or right skew, or or maybe even a somewhat normal distribution- it's hard to say from that histogram.

Overall, is this regression model unbiased? If an unbiased regression model, what does this tell us about the sample vs. the entire population model?

Well, it might be unbiased, since our assumptions were mostly ok, but it clearly has a hard time with the extreme values. It might imply that we will not be able to reliably represent the general population, but we have a pretty large sample, so perhaps we could have confidence in the model. I suppose I would do this again after removing lot sq ft and bathrooms from the model and see what effect that would have.