Part 4

Nick George

5/3/2018

Nick George and Spencer Louie

Introduction:

This study is based on a data set of home sales in King County, Washington from 2014 to 2015 (May) and comes from the Center for Spatial Data Science. The relevant variables are price, the sale price of the home; sqft_living, the square footage of living space; sqft_lot, the square footage of lot space; the number of floors; whether the property is watefront; the number of bedrooms/bathrooms, and the condition of the house (based on King County's grading system). From this data we are hoping to infer more about the general population of home sales in the U.S. As such specific variables like condition, will be extrapolated as roughly how much the structure of the house matters rather than the specific value because it is based on King County's system. The goal of this research is to figure out which factors are important in deciding the sale price of a home.

Ridge Regresion & Lasso Models

We run a ridge regression and lasso to compare coefficients with our MLR model. The two lists below show the RR and lasso coefficients, respectively.

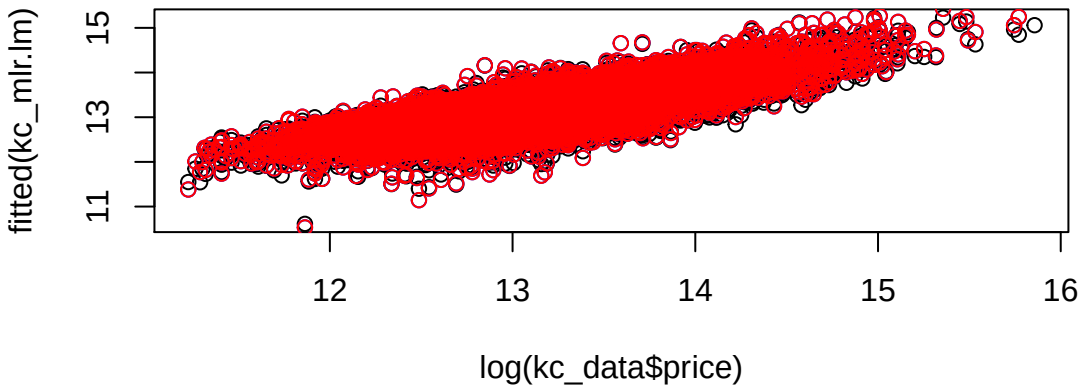
```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                           1
## (Intercept)                    8.557524e+00
## floors                         8.729300e-02
## waterfront                     3.782238e-01
## view                           5.667551e-02
## condition                      3.617602e-02
## grade                          2.397156e-01
## sqft_basement                  5.080626e-05
## bedbath                       -2.718327e-02
## bedbathi                       2.429013e-02
## lsqft_living <- log(sqft_living)  4.131819e-01
## lsqft_lot <- log(sqft_lot)       -4.034055e-02
## yr_built.adj <- yr_built - 1899  -5.157407e-03
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                                           1
## (Intercept)                    8.559741e+00
## floors                         8.724844e-02
## waterfront                     3.782168e-01
## view                           5.668150e-02
## condition                      3.615375e-02
## grade                          2.397558e-01
## sqft_basement                  5.082196e-05
## bedbath                       -2.619233e-02
## bedbathi                       2.340752e-02
## lsqft_living <- log(sqft_living)  4.127741e-01
## lsqft_lot <- log(sqft_lot)       -4.032229e-02
## yr_built.adj <- yr_built - 1899  -5.158198e-03
```

The ridge regression model suggests that we use all of the coefficients just as we would expect. The lasso model similarly suggests we use all of the variables which includes one more than we did in our own model. It suggests that we should include the amount of square footage in the basement, as well as the year the home was built. Note that the year built variable has been slightly adjusted so that 1900, the oldest home, is considered year 1 and then goes up normally from there. We then compare this with our full multiple linear regression model.

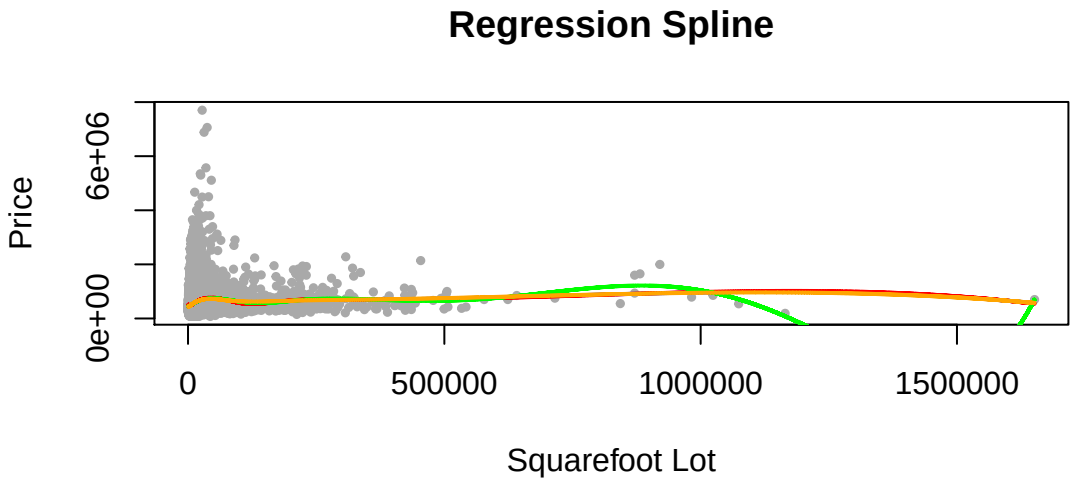
```
##      (Intercept)           floors       waterfront             view
##       8.13561484      -0.01401879        0.37942178       0.08740351
##        condition            grade          bedbath log(sqft_living)
##       0.09438878       0.20124267       -0.01973695       0.48274391
##    log(sqft_lot)
##      -0.05445465
```

All of the coefficients for predictors in all three models are the same across the model in terms of direction, though not necessarily in magnitude (but not great changes in magnitude either). Except for floors, which went from having a negative coefficient in our multiple linear regression model to a postive one in the lasso and ridge regression models. We originally did not include the variables added in these models for logistical reasons. First the amount of squarefootage in the basement was problematic because it did not include a way to qualify when the basement was finished or not. Furthermore for the year the house was built, some of the homes have had renovations, but we have no way of qualifying the quality or scale of renovations. And due to those problems we did not originally include those predictors.

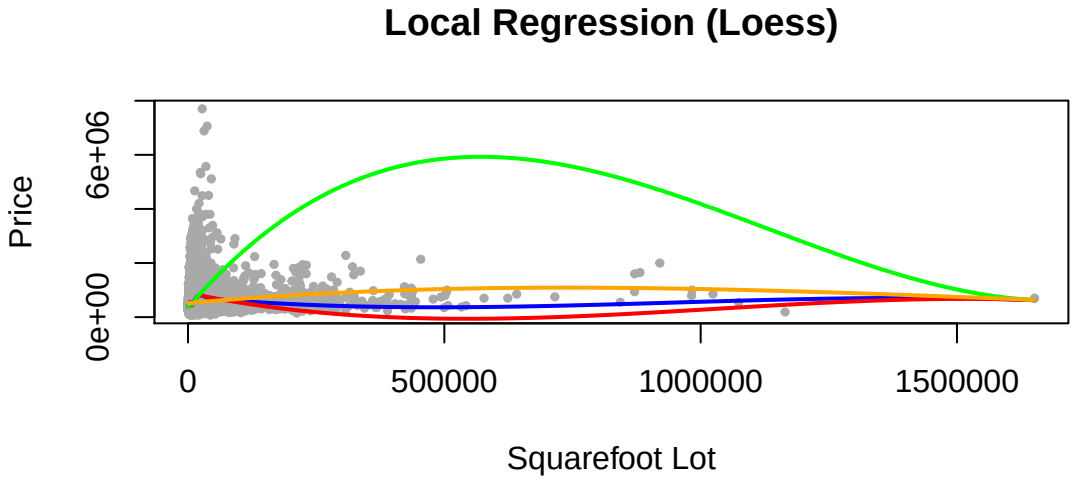


This graph shows the true values of the log of price on the x axis and the fitted values for each of the regression types on the y axis. Ideally we would a one-to-one relationship or a slope of 1 (although admittedly that could mean overfitting, which would be problematic). In black are multiple linear regression values, in blue are the ridge regression values and in red are the lasso values. The lasso and ridge regression models are very similar as you can see in the table with their coefficients, so in most places they are just overtop of each other. The black multiple linear regression points are also quite close to their blue/red counterparts implying that the models predict fairly similarly. Overall we have decided to stick with our originally multiple linear regression model because we are warry of the two logical reasons for excluding the two added variables in ridge regression and lasso and including them may cause some kind of unintended bias.

Spline & Kernel Smoother



The graph above illustrates four different possible regression splines for square footage of lot against price. All four models contain knots at 100,000, 200,000 and 300,000 in an effort to better explain the early variability. The blue line contains those three knots at degree 3. The red line contains those three knots as well as another early knot at 10,000, at degree 3 in an effort to sort out the variability we see extremely early on. The orange line adds a knot at the end of the blue model, at 750,000 to look at the differences at the higher levels. Lastly, the red line contains the blue model's knots at degree 5. We see that all four models seem to be fairly similar, but each has a slight difference based on the added portion to it. For example the orange is line is far more straight in the middle to end with it's knot at 750,000 and the green line has more curvature due to it's increased degree.



The graph above shows another way to smooth our curve in a local regression method called Loess. The four lines predict on the same variables for the same values, but differ in the span for each curver. The blue curve has a span of .2, the red curve has a span of .5, the green curve has a span of .7, and the orange curve has a span of 1. The span effects the curve in that it dictates how many of the nearby points are used to predict the line, amongst those points the ones closest to it are given higher weights, and that is again dependent on

the span. The smaller the span, the closer the points have to be to be considered in the local regression, and the lower the weight they are given. We see that he blue, orange, and red lines seem to fit the data fairly well, while the green line seems to be fairly off. This likely is because the points that were used for the local regression, particularly in the middle, caused a strange fit. The blue and red curve's use only the data right around each point to predict to it seems to fit the data well. The orange curve uses all of the data and runs the regression (though with weights) and also seems to fit alright.

Overall we would the blue Loess model to predict future values. We think it is important to have a low span so that you are only regressing using the points right around the point in question. This is especially true for this variable because the data does not have a clear functional form. That is also why we are not choosing to use regression splines because, again, there is not a clear functional form for the data, so doing a local regression seems more beneficial.

Conclusion

Ridge regression and lasso yielded the results we were expecting. With the number of observations far outweighing the number of predictors, it was expected that the RR and lasso to find all variables significant to the model. However, due to the way the variables are constructed it makes more sense to use the MLR that was determined from the last part of the project. Further, given the type and number of variables we have, RR or lasso are solving a problem that is not present in the data.

LOESS and splines also yielded expeced results. Choosing the appropriate number and location of knots was an important determination, but ultimately the spline curves looked quite similar. Much more important was choosing the span of the LOESS, which made the smoothed curve vary widely. Overall, smooth regressions are much more appropriate for the kind of data we have compared with sparse models. Choosing an appropriate smooth regression function yielded curves that fit the data well.

Lack of Fit

One factor that we are particularly cognisant of in our research is the fit of our model. Without an approriate fit the inference of our predictors is fairly meaningless and inaccurate. It is also important because the shape of our model is not necesarilly obvious. So we have decided to look at a few things to help us ensure an appropriate model. First, we have decided to look at the Lack of Fit test. This is an F-Test with the simple goal of evaluating fit. One of the convenient aspects of this test is that it does not require any additional assumptions then the one we have used to run our regressions as it is simply looking at two regressions. In fact, it does not even require linearity as that is what it is meant to test for. This test uses a nested F-test, so it is comprised of both a full model and a reduced model. The full model looks at a regression where instead of assuming a linear relationship it counts each variable as a factor variable, so it measure the mean Y at each level of a variable. In other words what is the average value of Y when X1=1. The reduced model is the regression you wish to test, normally one that assumes linearity and so we would have our normal βs telling us the slope of a predictor instead of giving as a value at each level of it. This model only works if the variables you are testing are categorical in this sense. There must be multiple observations at each level. Below we have run this test on our variables that fit these requirements. Note that sqft_living and sqft_lot are not run as factor variables because they are continuous and there are not multiple observations at each level.

```
## Analysis of Variance Table
##
## Model 1: log(price) ~ log(sqft_living) + log(sqft_lot) + factor(floors) +
##     factor(waterfront) + factor(view) + factor(condition) + factor(grade) +
##     factor(as.integer(bedbath))
## Model 2: log(price) ~ log(sqft_living) + log(sqft_lot) + floors + waterfront +
##     view + log(condition) + grade + bedbathi
##   Res.Df    RSS  Df Sum of Sq      F    Pr(>F)
```

```
## 1  21568 2291.8
## 2  21603 2448.4 -35   -156.61 42.111 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

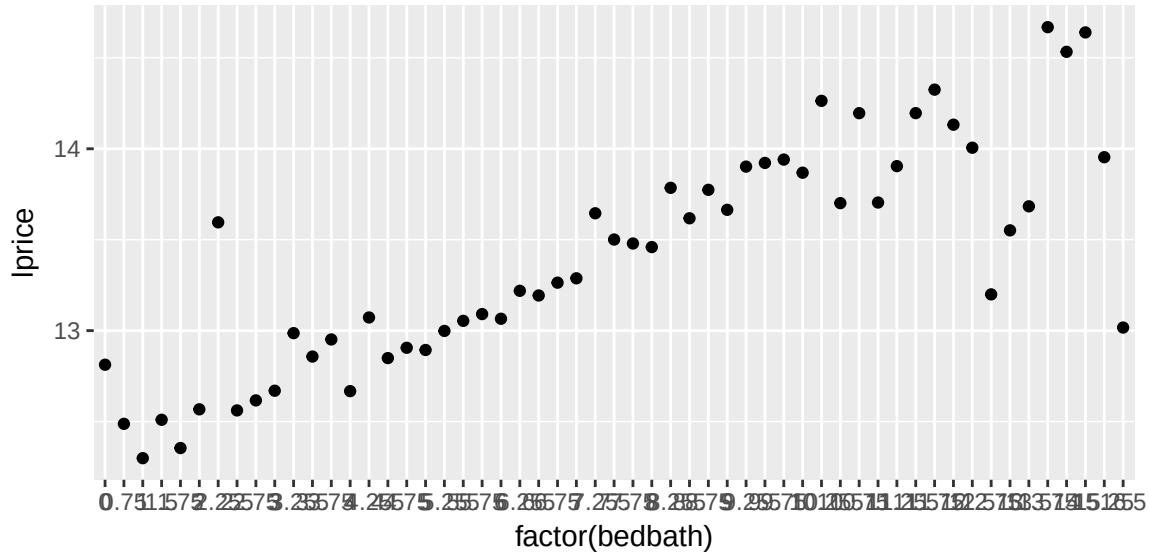
Here we find that we reject the null hypothesis, that the model fits, because our p-value is near 0, thus implying that the model does not fit. So instead of assuming a linear relationship with these stepped variables we will instead consider them as factor variables, finding the mean price at each given level, for example the mean price for a home with 5 beds and baths. The results for such a regression are shown below.

```
##
## Call:
## lm(formula = log(price) ~ log(sqft_living) + log(sqft_lot) +
##     factor(as.integer(floors)) + factor(waterfront) + factor(view) +
##     factor(condition) + factor(grade) + factor(bedbathi), data = kc_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24777 -0.22812  0.01224  0.21960  1.38226
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  9.487103   0.334842  28.333  < 2e-16 ***
## log(sqft_living)             0.531999   0.011361  46.825  < 2e-16 ***
## log(sqft_lot)               -0.064335   0.002954 -21.777  < 2e-16 ***
## factor(as.integer(floors))2 -0.070014   0.006132 -11.419  < 2e-16 ***
## factor(as.integer(floors))3  0.014625   0.015027   0.973 0.330453
## factor(waterfront)1          0.394884   0.032219  12.256  < 2e-16 ***
## factor(view)1                0.216060   0.018336  11.783  < 2e-16 ***
## factor(view)2                0.151602   0.011059  13.708  < 2e-16 ***
## factor(view)3                0.194048   0.015112  12.840  < 2e-16 ***
## factor(view)4                0.299117   0.023495  12.731  < 2e-16 ***
## factor(condition)2          -0.052915   0.066261  -0.799 0.424544
## factor(condition)3           0.028207   0.061648   0.458 0.647277
## factor(condition)4           0.103911   0.061693   1.684 0.092136 .
## factor(condition)5           0.227108   0.062059   3.660 0.000253 ***
## factor(grade)3              -0.950956   0.424632  -2.239 0.025135 *
## factor(grade)4              -0.646047   0.366251  -1.764 0.077755 .
## factor(grade)5              -0.644983   0.361445  -1.784 0.074364 .
## factor(grade)6              -0.497203   0.361268  -1.376 0.168752
## factor(grade)7              -0.309966   0.361323  -0.858 0.390976
## factor(grade)8              -0.111261   0.361458  -0.308 0.758229
## factor(grade)9               0.127765   0.361646   0.353 0.723875
## factor(grade)10              0.330526   0.361827   0.913 0.360995
## factor(grade)11              0.505423   0.362197   1.395 0.162899
## factor(grade)12              0.716389   0.363099   1.973 0.048510 *
## factor(grade)13              1.035229   0.374239   2.766 0.005676 **
## factor(bedbathi)1            0.869184   0.221506   3.924 8.74e-05 ***
## factor(bedbathi)2            0.449089   0.137091   3.276 0.001055 **
## factor(bedbathi)3            0.407335   0.134974   3.018 0.002548 **
## factor(bedbathi)4            0.250330   0.134684   1.859 0.063091 .
## factor(bedbathi)5            0.228077   0.134631   1.694 0.090263 .
## factor(bedbathi)6            0.183909   0.134579   1.367 0.171780
## factor(bedbathi)7            0.234391   0.134807   1.739 0.082099 .
## factor(bedbathi)8            0.232129   0.134811   1.722 0.085106 .
```

```
## factor(bedbathi)9            0.307271   0.136350   2.254 0.024235 *
## factor(bedbathi)10           0.298175   0.137220   2.173 0.029794 *
## factor(bedbathi)11           0.297289   0.145828   2.039 0.041499 *
## factor(bedbathi)12           0.407625   0.156558   2.604 0.009230 **
## factor(bedbathi)13           0.161015   0.212649   0.757 0.448944
## factor(bedbathi)14           0.555381   0.185347   2.996 0.002735 **
## factor(bedbathi)15           0.058779   0.269328   0.218 0.827240
## factor(bedbathi)16           0.027506   0.355514   0.077 0.938330
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3289 on 21571 degrees of freedom
## Multiple R-squared:  0.6107, Adjusted R-squared:   0.61
## F-statistic:   846 on 40 and 21571 DF,  p-value: < 2.2e-16
```

Now we have significantly more coefficients as we have one for each factor level, or really we have one less than that as the intercept term contains all of the base factor levels, such as view=0 (not having a view) and grade=2 (the lowest grade in the data set). Some interesting notes are that grade is only significant for it's bottom three and final two (3,4,5 and 12 and 13), which implies that grades 6 through 11 all have the same effect on home price. So only when a home's grade is really high will it be important benefit in the price. Similarly condition is only significant for its two highest levels 4 and 5, so again only if the condition of a home is extraordinary will it be associated with a benefit in the sale price. when we look a the bedbath coefficients we can see them somewhat more like cutoffs. 5, 5.25, 5.5 etc are not necessarily significant but 6 is, so within a certain value for bedbath there is not necesarilly an extra benefit, though having more is generally better as at certain points it becomes significant. View we find significant at every level and has positive coefficients so each factor higher in view is associated with an increase in price.

```
## Analysis of Variance Table
##
## Model 1: price ~ bedbath + bedbath2
## Model 2: price ~ as.factor(bedbath)
##   Res.Df        RSS Df Sum of Sq      F    Pr(>F)
## 1  21609 2.1954e+15
## 2  21557 2.1006e+15 52 9.483e+13 18.715 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



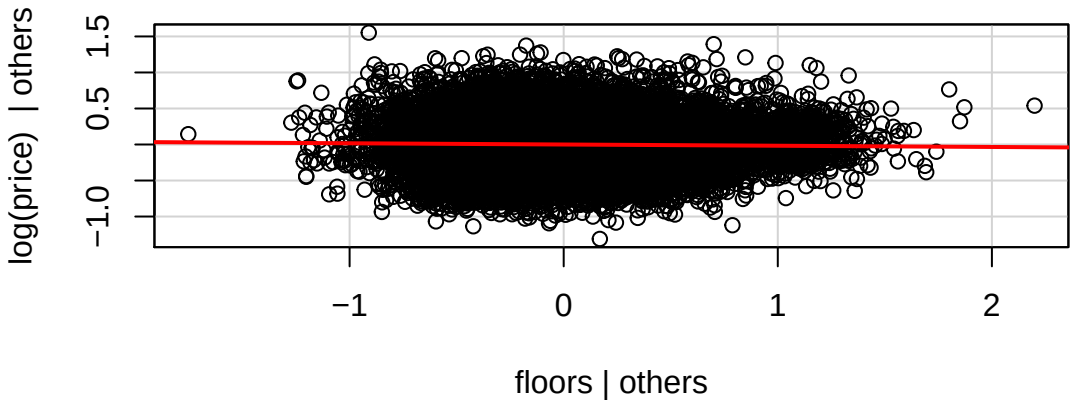
Another way to check for fit as well as significanace is through added variable plots. The premise behind these plots is to see how and if an additional predictor matters when you consider the other predictors already included in the model. We are more interested in the how part of the added variable plot than the if. The plot gives us the ability to again help evaluate fit. We can see if the fit is linear or something else. Fit is extremely important for our model and any model, because if the model does not accurately represent the data then the inferences from it will be innaccurate. Our model fit is particularly important because it is not obviously linear as is, and to address that we have already made several transformations to the data. The plots show the residuals of one variale given that the other variables are in teh model. Each point is $e_i(Y|X_2 + X_3 + X_4) = Y_i - \hat{Y}(X_2) - \hat{Y}(X_3) - \hat{Y}(X_4)$ on the Y axis. On the X axis it is, $e_i(X_1|X_2 + X_3 + X_4) = X_{i1} - \hat{X_{i1}}(X_2) - \hat{X_{i1}}(X_3) - \hat{X_{i1}}(X_4)$ In order to get these points we are doing a regression so we do require the normal assumptions or technical conditions for a regression, normality and constant variance for error terms, independence and linearity. With our dataset being so large we are able to satisfy the normality condition and we have checked for the previous assumption earlier.

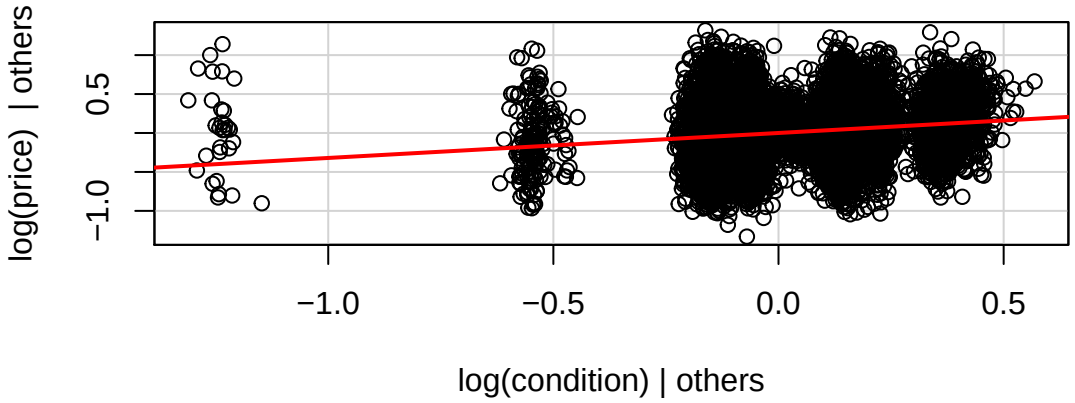
```
## Loading required package: car
```

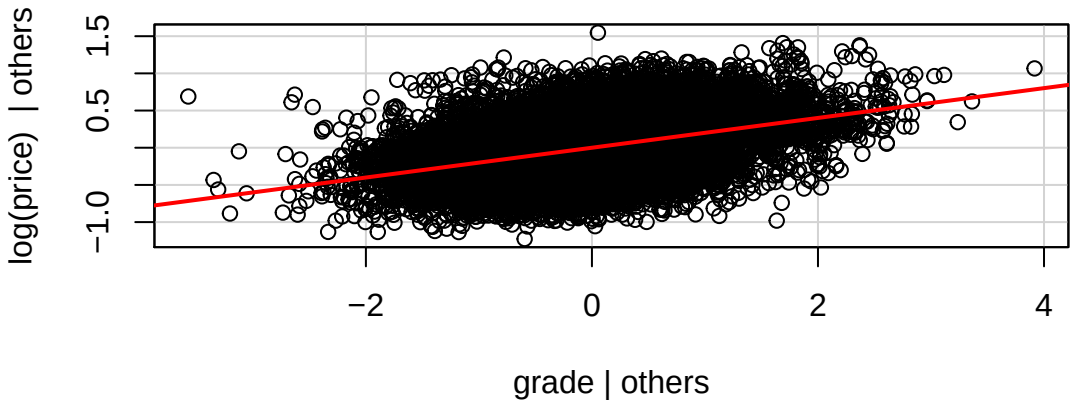
```
##
## Attaching package: 'car'
```

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

```
## The following object is masked from 'package:openintro':
##
##     densityPlot
```

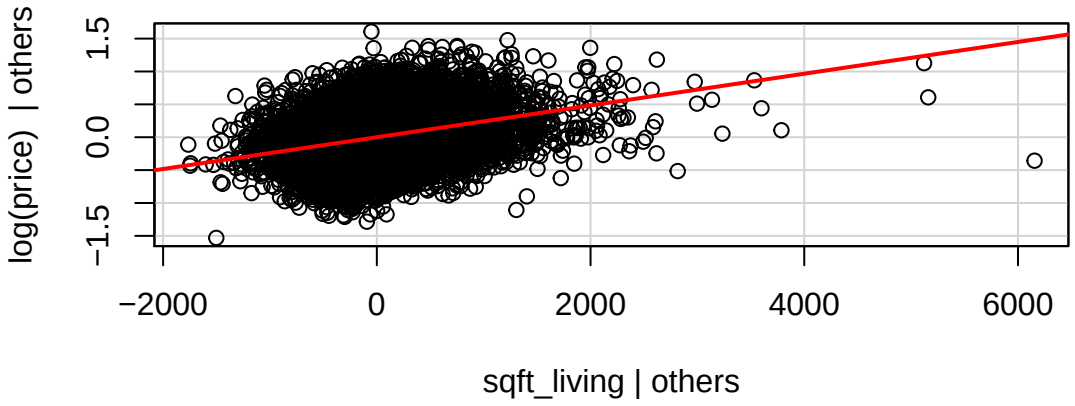

log(price) | others
1.5
0.5
−1.0
−1
0
1
2
floors | others


log(price) | others
0.5
−1.0
−1.0
−0.5
0.0
0.5
log(condition) | others



Here we have several added variable plots for a few variables of interest. Since none of them have straight horizontal line we do see a relationship for all of them and therefore each one adds something to the plot. Floors and bedbath have some what negative relationships, when the other variables are considered in the model. While condition and grade have fairly significant postive relationships, when considering the other variables in the model. We also see the the lines themselves as well as the points seem to imply linear relationships, the lines are straight and the points are not mostly above or below the line at certain parts, rather they are even distributed at ever segement of the line.

```
avPlots(lm(log(price) ~ sqft_living + log(sqft_lot) + floors + waterfront + view + log(condition) + gra
```



Here we have a graph that all parts of the model included except for sqft_living, which we are looking at without a transformation. We do see that the line is straight, but the points are not as evenly distributed as we have seen the in past. Earlier and later on it seems more points are below and in the middle more are above. The curvature implies we should use a transformation, which we have done.

Generalized Additive Model

Overall Summary