STAT 230 - Analysis Component - Group 1B

Names: Charlotte Kellogg, Anna Zhou, Alison Ortiz Dimas

(Tentative) Project Title: Predicting Housing Prices in King County, Washington State

Our project is about predicting housing prices in King County, WA, using ten quantitative variables. We are researching all houses that were sold in King County between May 2014 and May 2015 using thousands of observations from 372 dates.

Read in the data

```
# Must start with a data-read in command from Prof. Wagaman
theData <- read_csv("https://awagaman.people.amherst.edu/stat495/kc_house_data.csv")
```

```
## Rows: 21597 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): date
## dbl (20): id, price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, wat...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
theData <- mutate(theData, renovated = ifelse(yr_renovated>0,"Yes","No"))
```

Summary command on data set

```
glimpse(theData)
```

```
## Rows: 21,597
## Columns: 22
## $ id            <dbl> 7129300520, 6414100192, 5631500400, 2487200875, 19544005~
## $ date          <chr> "10/13/2014", "12/9/2014", "2/25/2015", "12/9/2014", "2/~
## $ price         <dbl> 221900, 538000, 180000, 604000, 510000, 1230000, 257500,~
## $ bedrooms      <dbl> 3, 3, 2, 4, 3, 4, 3, 3, 3, 3, 3, 2, 3, 3, 5, 4, 3, 4, 2,~
## $ bathrooms     <dbl> 1.00, 2.25, 1.00, 3.00, 2.00, 4.50, 2.25, 1.50, 1.00, 2.~
## $ sqft_living   <dbl> 1180, 2570, 770, 1960, 1680, 5420, 1715, 1060, 1780, 189~
## $ sqft_lot      <dbl> 5650, 7242, 10000, 5000, 8080, 101930, 6819, 9711, 7470,~
## $ floors        <dbl> 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 2.0, 1.0, 1~
## $ waterfront    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ view          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0,~
## $ condition     <dbl> 3, 3, 3, 5, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 3, 3, 3, 4, 4,~
## $ grade         <dbl> 7, 7, 6, 7, 8, 11, 7, 7, 7, 7, 8, 7, 7, 7, 7, 9, 7, 7, 7~
## $ sqft_above    <dbl> 1180, 2170, 770, 1050, 1680, 3890, 1715, 1060, 1050, 189~
## $ sqft_basement <dbl> 0, 400, 0, 910, 0, 1530, 0, 0, 730, 0, 1700, 300, 0, 0, ~
## $ yr_built      <dbl> 1955, 1951, 1933, 1965, 1987, 2001, 1995, 1963, 1960, 20~
## $ yr_renovated  <dbl> 0, 1991, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ zipcode       <dbl> 98178, 98125, 98028, 98136, 98074, 98053, 98003, 98198, ~
## $ lat           <dbl> 47.5112, 47.7210, 47.7379, 47.5208, 47.6168, 47.6561, 47~
## $ long          <dbl> -122.257, -122.319, -122.233, -122.393, -122.045, -122.0~
## $ sqft_living15 <dbl> 1340, 1690, 2720, 1360, 1800, 4760, 2238, 1650, 1780, 23~
## $ sqft_lot15    <dbl> 5650, 7639, 8062, 5000, 7503, 101930, 6819, 9711, 8113, ~
## $ renovated     <chr> "No", "Yes", "No", "No", "No", "No", "No", "No", "No", "~
```

Data Codebook

List your variables and whether they are quantitative/qualitative (numeric vs categorical), along with other notes about the variables. Hint, to make a nice list, you need to put two spaces at the end of each line to force RMarkdown to start a new line. Check this out below:

Our variables are:
Variable 1 - date - categorical Variable 2 - bedrooms - numeric Variable 3 - bathrooms - numeric Variable 4 - sqft_living - numeric Variable 5 - sqft_lot - numeric Variable 6 - floors - numeric Variable 7 - condition - numeric Variable 8 - sqft_basement - numeric Variable 9 - yr_built - numeric Variable 10 - renovated - categorical

Analysis Plan

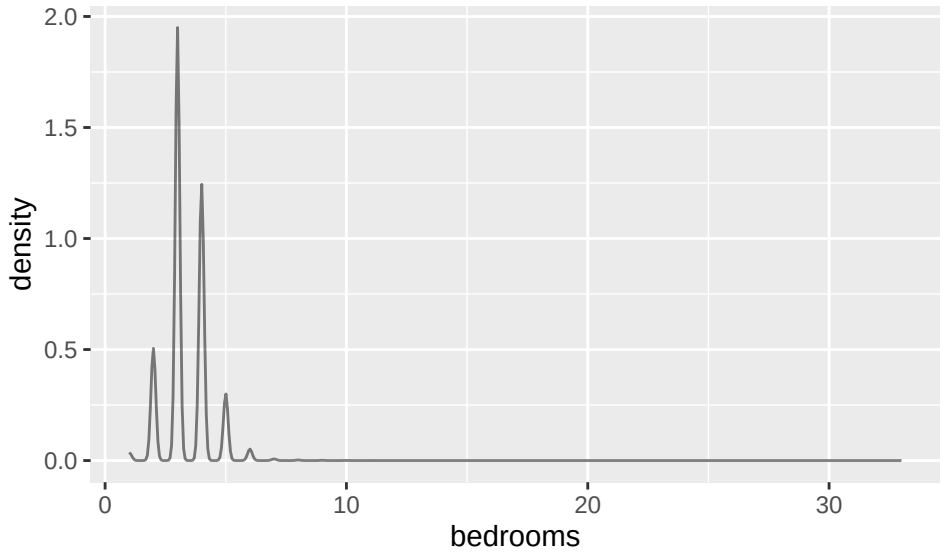
A rough outline for your proposed analysis, including univariate summaries, bivariate (or multivariate) relationships and plans for your model(s) and visualizations should follow via the sections below; as well as any additional thoughts about your randomization-based procedure.

This section doesn't need to have plots for EVERY variable in your data set if you have many, but it needs to demonstrate that you've started exploring the data, identifying issues that arise, and are looking into what appropriate models might be.

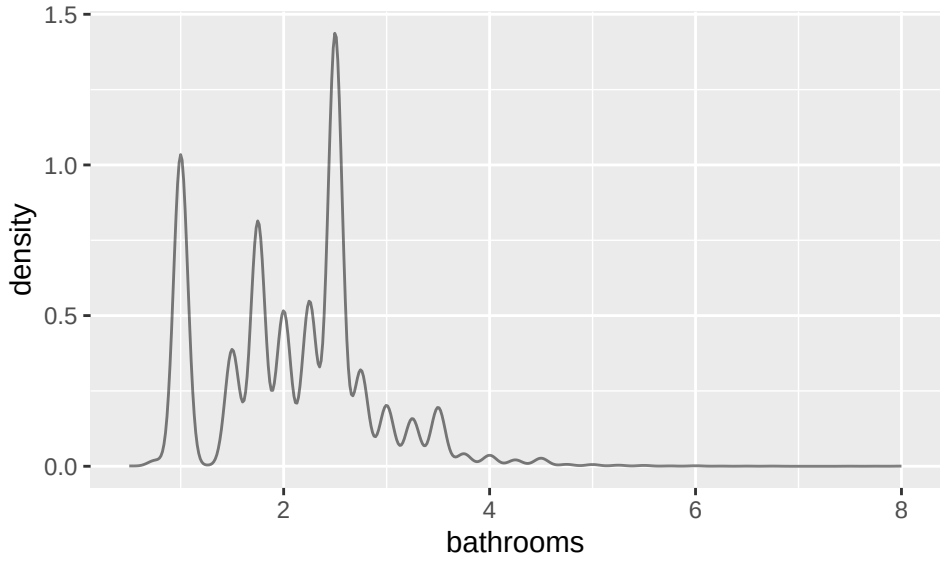
What you do here may be useful for your reports (i.e. may reuse it), so you may want to spend time making nice labels for plots, etc.

Prelim Univariate Analysis Obtain basic univariate descriptive statistics and graphs for variables relevant to your analysis.

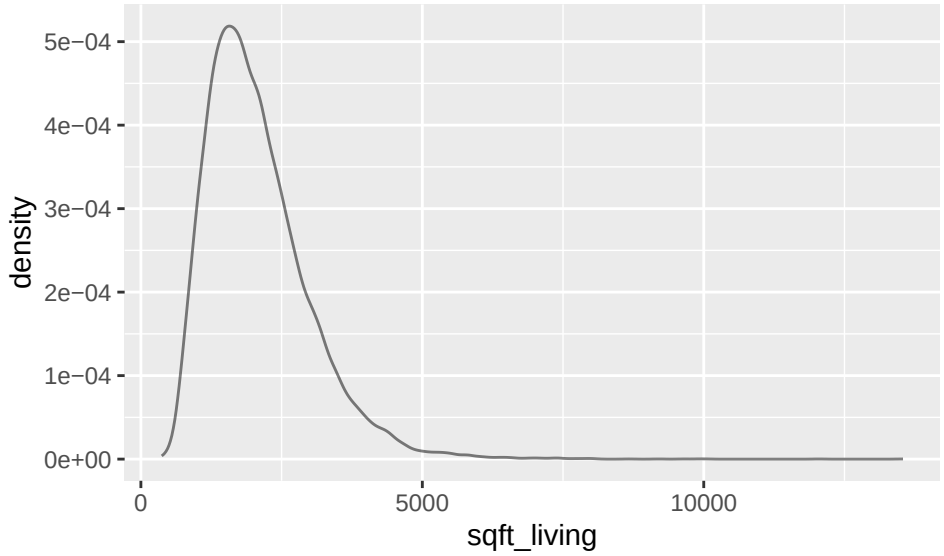
```
gf_dens(~bedrooms, data=theData)
```



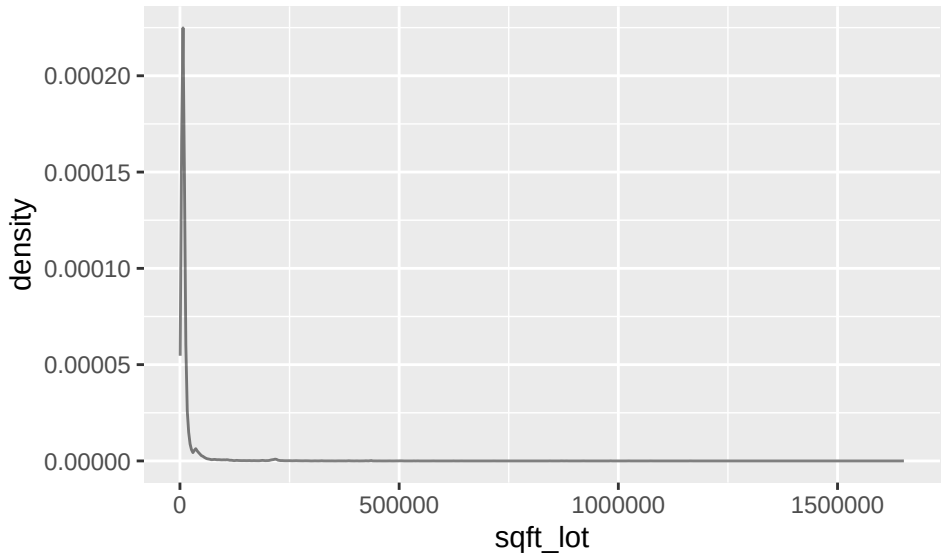

```
gf_dens(~bathrooms, data=theData)
```



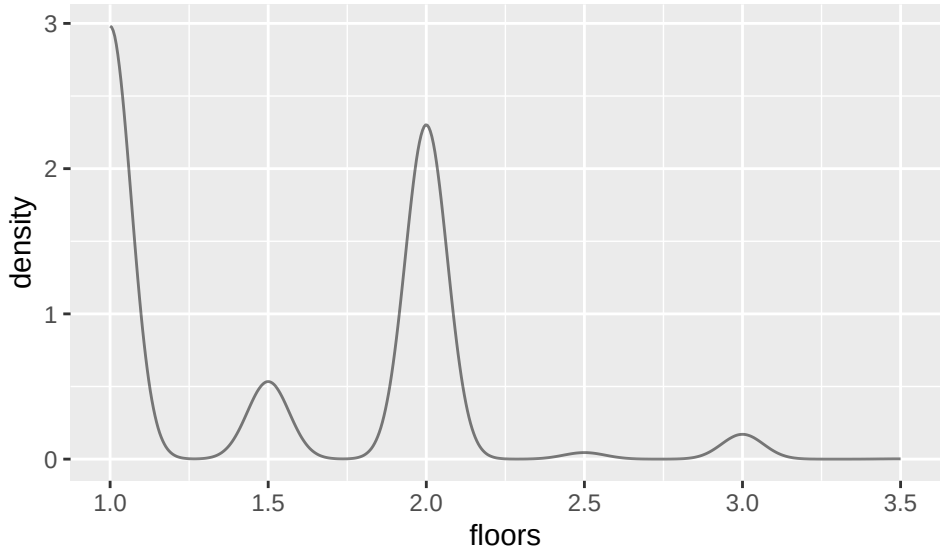

```
gf_dens(~sqft_living, data=theData)
```



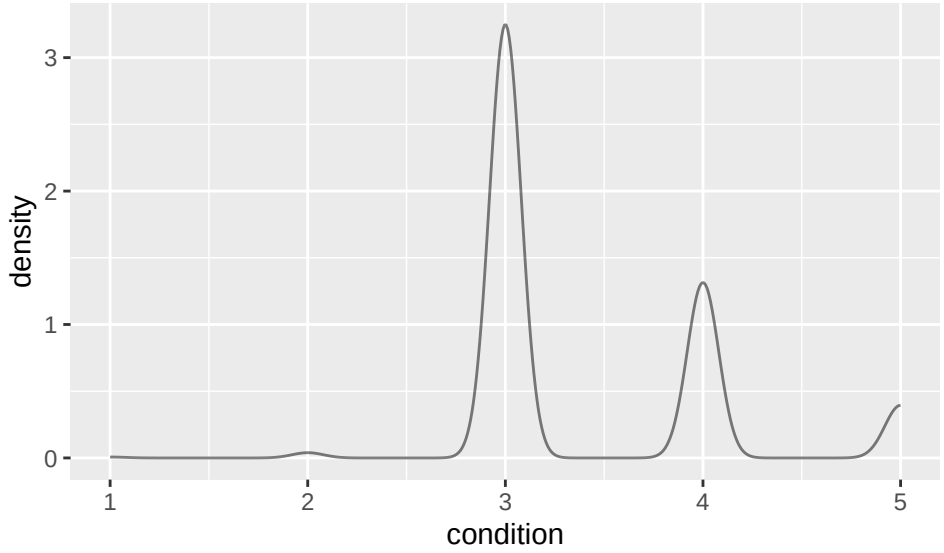

```
gf_dens(~sqft_lot, data=theData)
```



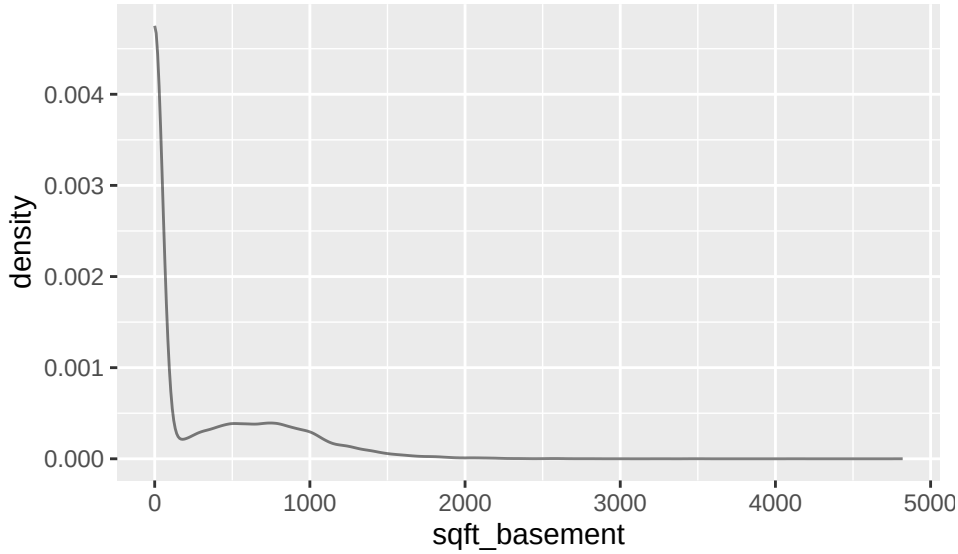

```
gf_dens(~floors, data=theData)
```



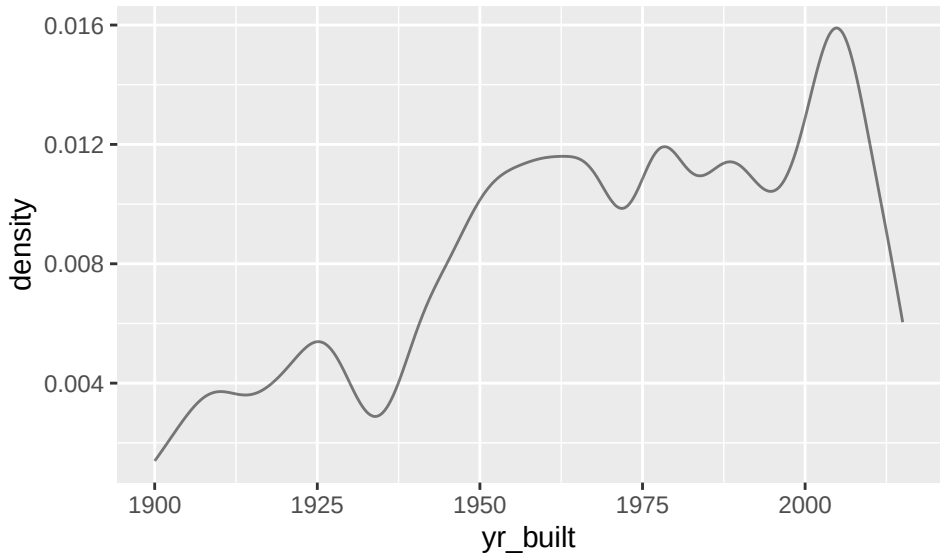

```
gf_dens(~condition, data=theData)
```




```
gf_dens(~sqft_basement, data=theData)
```



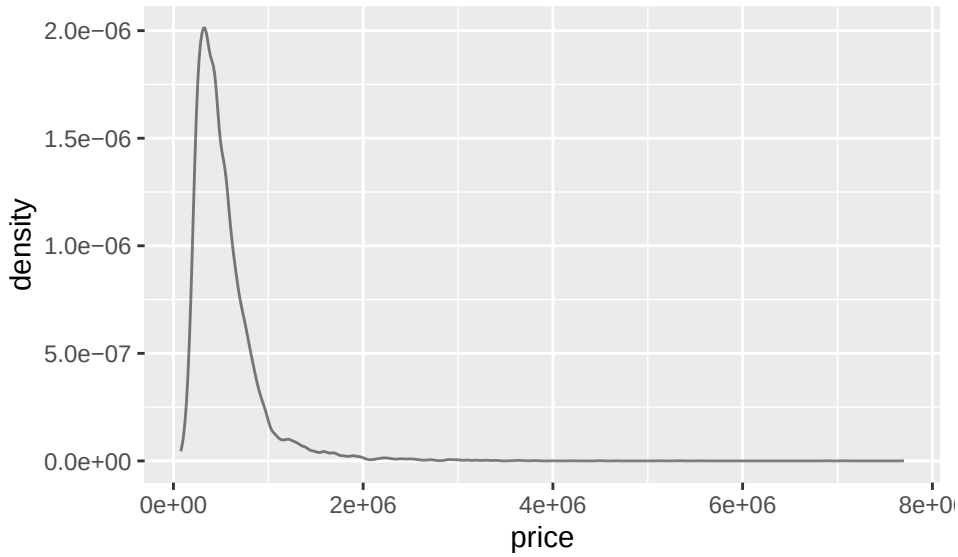

```
gf_dens(~yr_built, data=theData)
```




```
tally(~renovated, data=theData)
```

```
## renovated
##    No   Yes
## 20683   914
```

```
gf_dens(~price, data = theData)
```




```
msummary(theData$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   78000  322000  450000  540297  645000 7700000
```

```
msummary(theData$bedrooms)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   3.000   3.373   4.000  33.000
```

```
msummary(theData$bathrooms)
```

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

```
##   0.500   1.750   2.250   2.116   2.500   8.000
```

```
msummary(theData$sqft_living)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     370    1430    1910    2080    2550   13540
```

```
msummary(theData$sqft_lot)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     520    5040    7618   15099   10685 1651359
```

```
msummary(theData$floors)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.500   1.494   2.000   3.500
```

```
msummary(theData$condition)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    3.00    3.00    3.41    4.00    5.00
```

```
msummary(theData$sqft_basement)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0   291.7   560.0  4820.0
```

```
msummary(theData$yr_built)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1900    1951    1975    1971    1997    2015
```

```
msummary(theData$renovated)
```

```
##    Length     Class      Mode
##     21597 character character
```

COMMENT on what you see! Suggest doing this as you go. I.E. Do a summary for variable 1, then variable 2, etc.

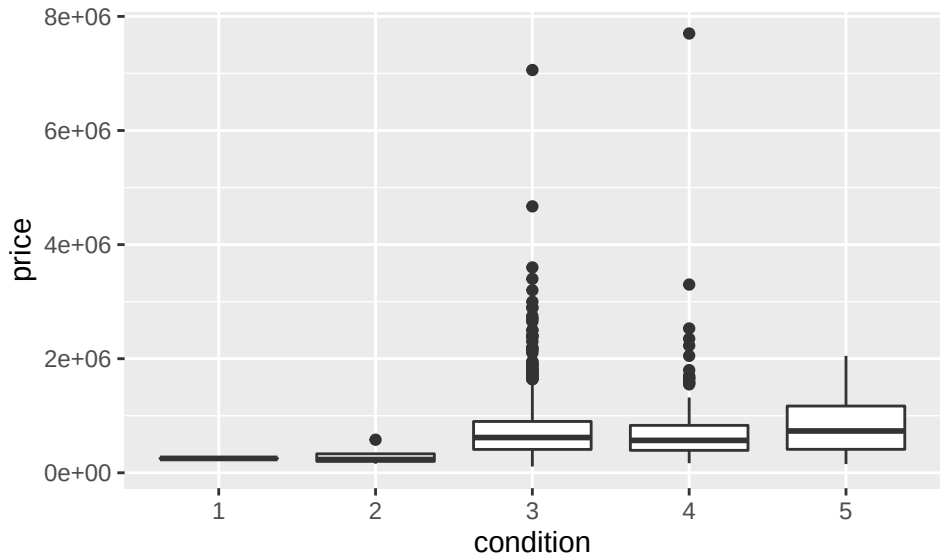
The density plots of bathrooms, bedrooms, sqft_living, sqft_lot, sqft_basement, and price are all highly skewed to the right, suggesting that the means are greater than the medians for those variables. There is a lot of variability within the yr_built variable. There are many more houses that were not renovated than there are that were.

It is very important to do this for your response variable(s), so be sure those are included here.

Prelim Multivariate Analysis Scatterplots and side-by-side boxplots to examine bivariate relationships that will be useful for building your models. I.E. relationships between your response(s) and predictors. You can investigate relationships between predictors of interest as desired.

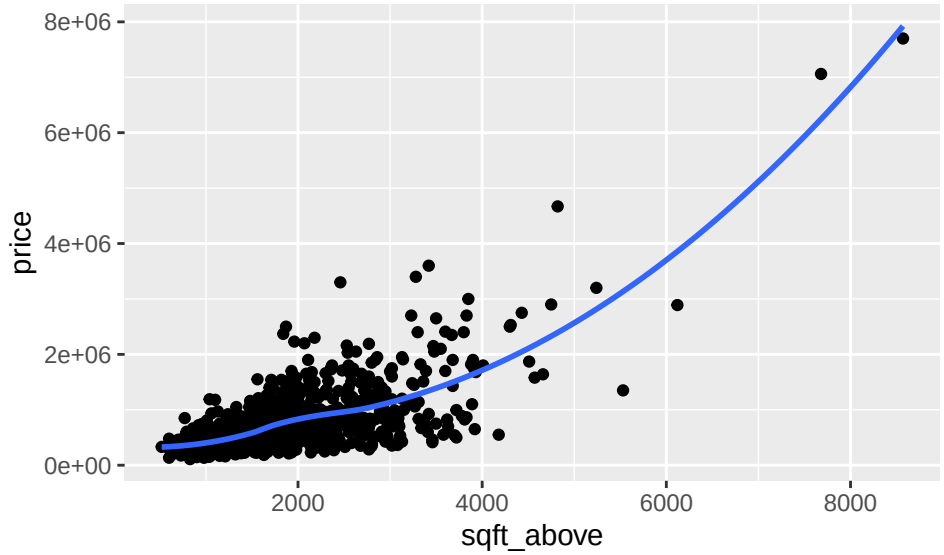
```
theData2 <- theData %>%  #selecting the variables that we decided to work with according to proposal
  select(c(price
           , sqft_above
           , sqft_basement
           , bedrooms
           , bathrooms
           , sqft_living
           , sqft_lot
           , condition
           , floors
```

```
          , yr_built
          , yr_renovated)) %>%
  filter(yr_renovated != 0) #removing values where there was no renovation

theData2$condition <- factor(theData2$condition) #converting condition into a categorical variable

gf_boxplot(price ~ condition, data = theData2)
```



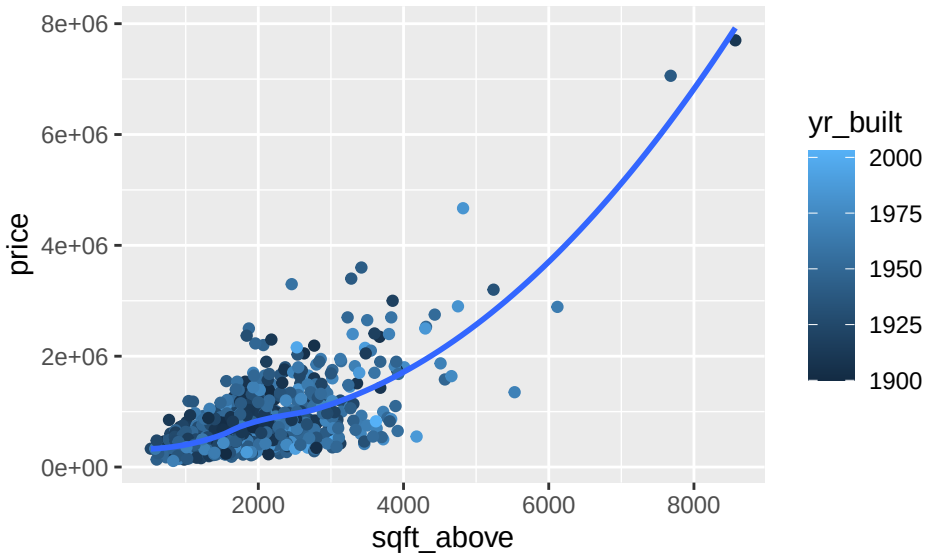

```
ggplot(theData2, aes(sqft_above, price)) + geom_point() + geom_smooth(se = F)
```

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




```
ggplot(theData2, aes(sqft_above, price, color=yr_built)) + geom_point() + geom_smooth(se = F)
```

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




```
ggplot(theData2, aes(sqft_above, price, color=condition)) + geom_point() + geom_smooth(se = F)
```

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

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

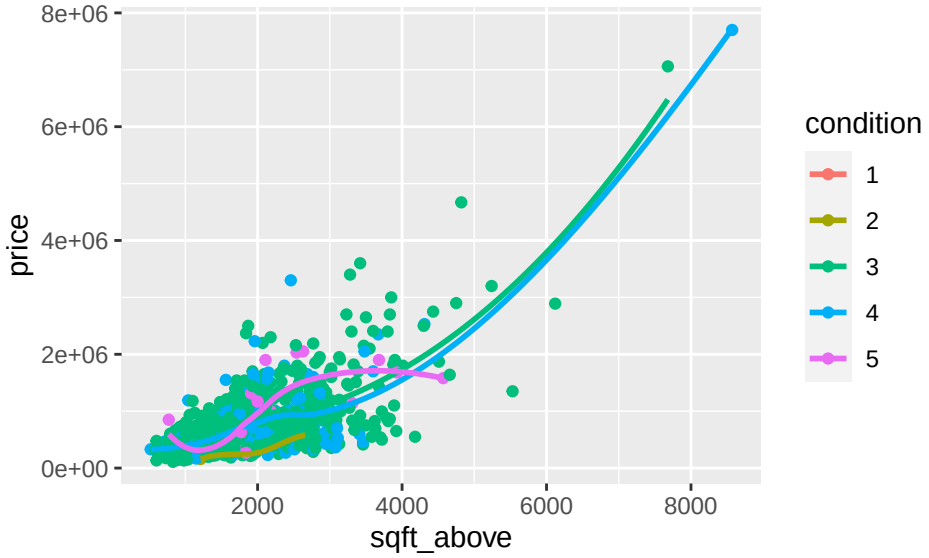
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1202.8
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 687.25
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

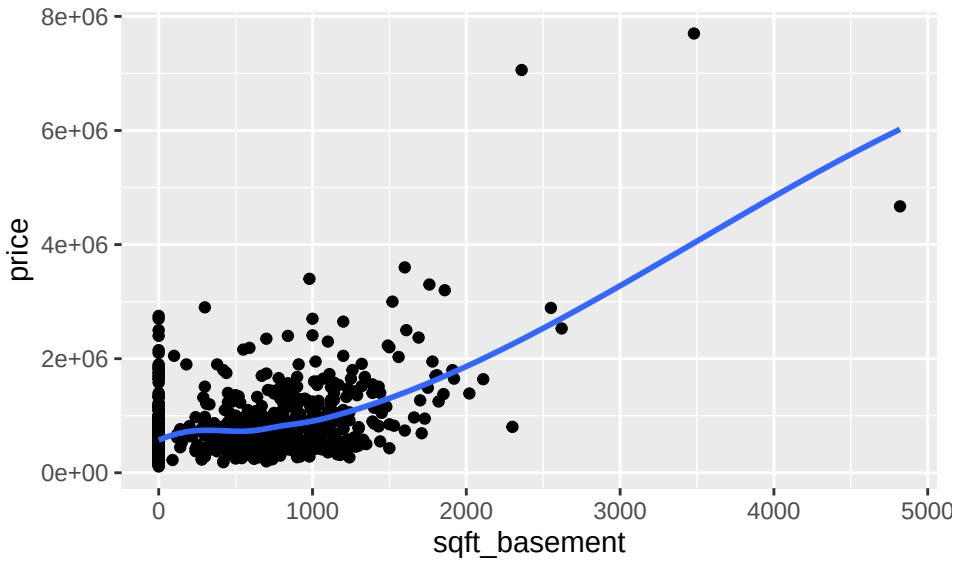
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.657e+06
```

```
## Warning in sqrt(sum.squares/one.delta): NaNs produced
```



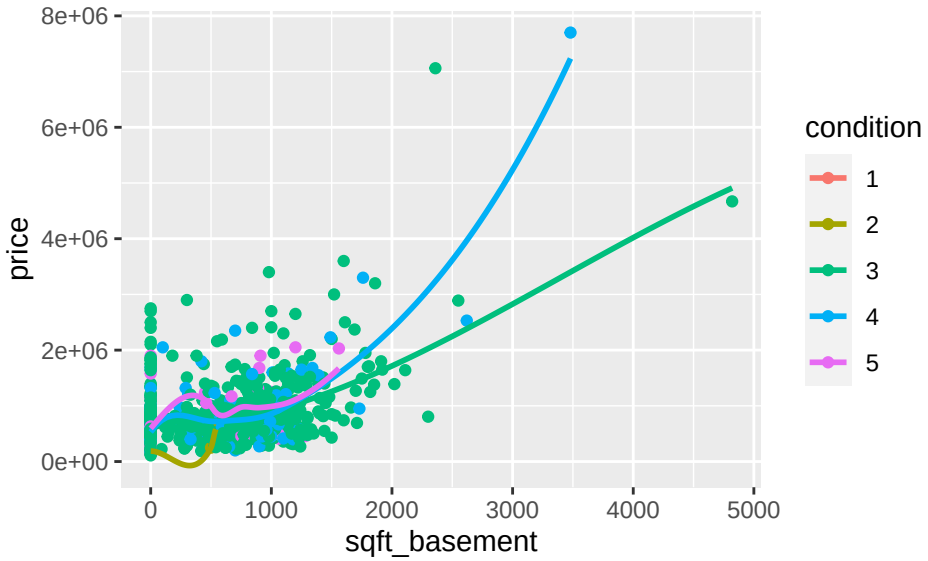
```
ggplot(theData2, aes(sqft_basement, price)) + geom_point() + geom_smooth(se = F)
```

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



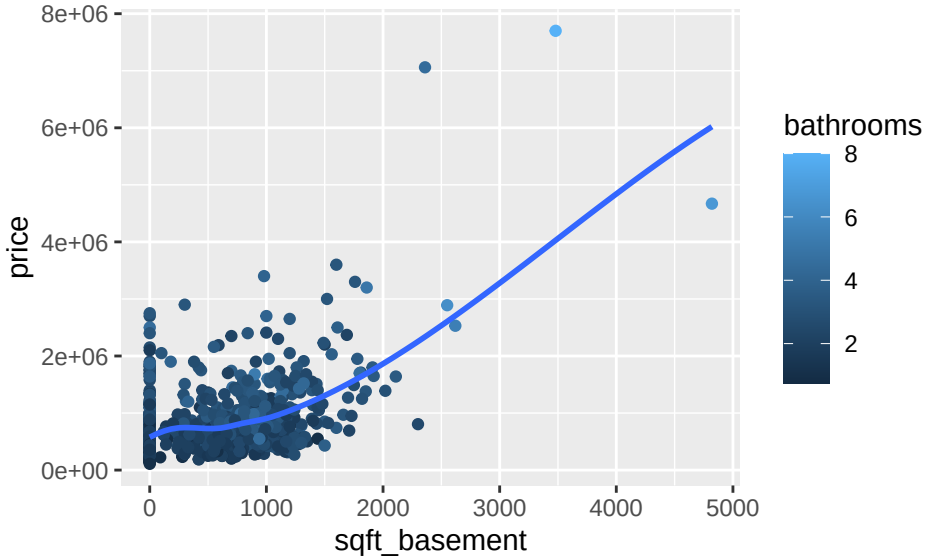

```
ggplot(theData2, aes(sqft_basement, price, color = condition )) + geom_point() + geom_smooth(se = F)
```

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

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -2.7

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 502.7

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 2.9452e+05
```



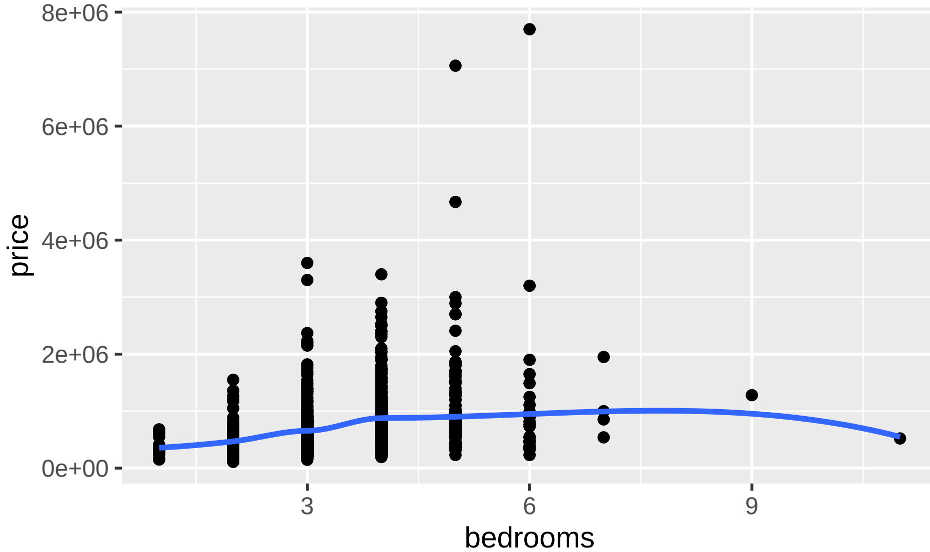

```
ggplot(theData2, aes(sqft_basement, price, color = bathrooms )) + geom_point() + geom_smooth(se = F)
```

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



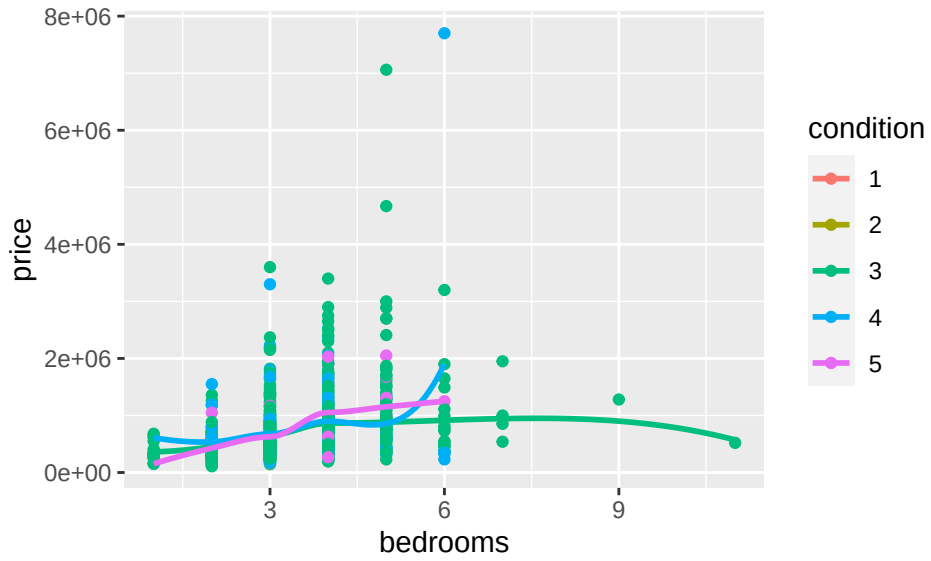

```
ggplot(theData2, aes(bedrooms, price)) + geom_point() + geom_smooth(se=F)
```

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

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 3

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1
```



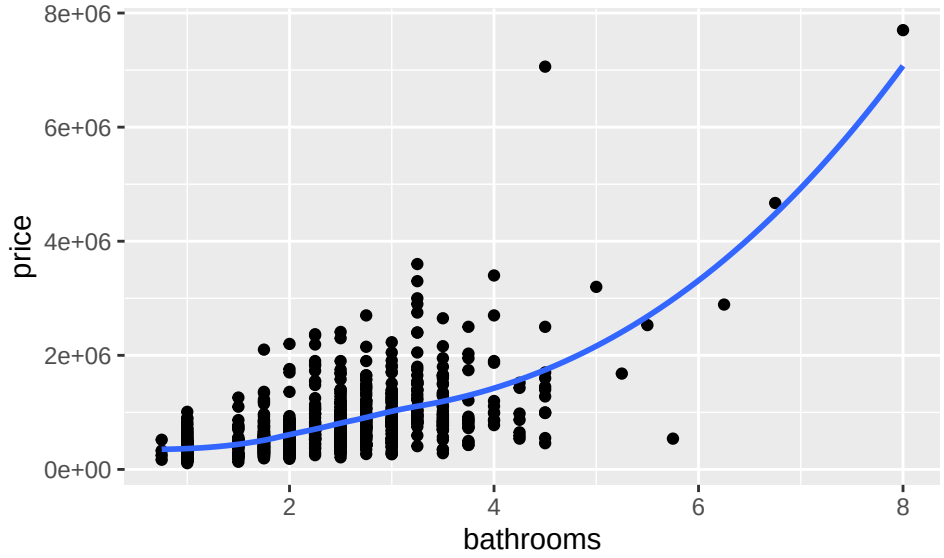

```
ggplot(theData2, aes(bedrooms, price, color = condition)) + geom_point() + geom_smooth(se=F)
```

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

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 3

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 3

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 3

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1
```




```
ggplot(theData2, aes(bathrooms, price)) + geom_point() + geom_smooth(se=F)
```

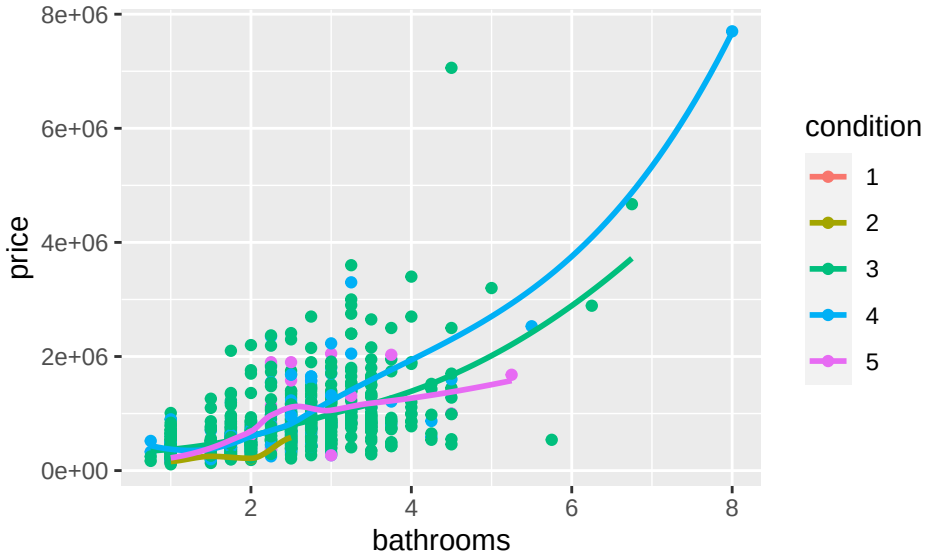
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(theData2, aes(bathrooms, price, color = condition)) + geom_point() + geom_smooth(se=F)
```

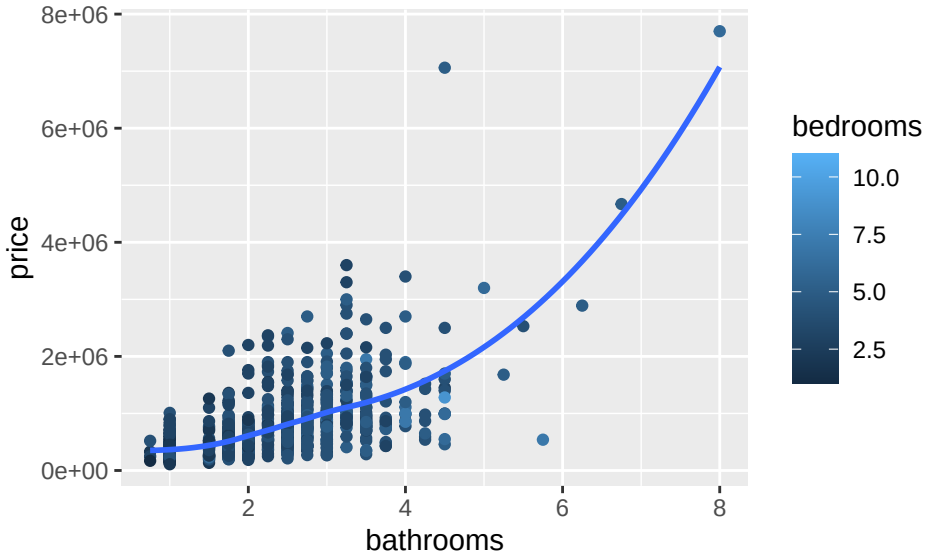
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.9925

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.0075

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
```

```
## parametric, : There are other near singularities as well. 1.0151
```



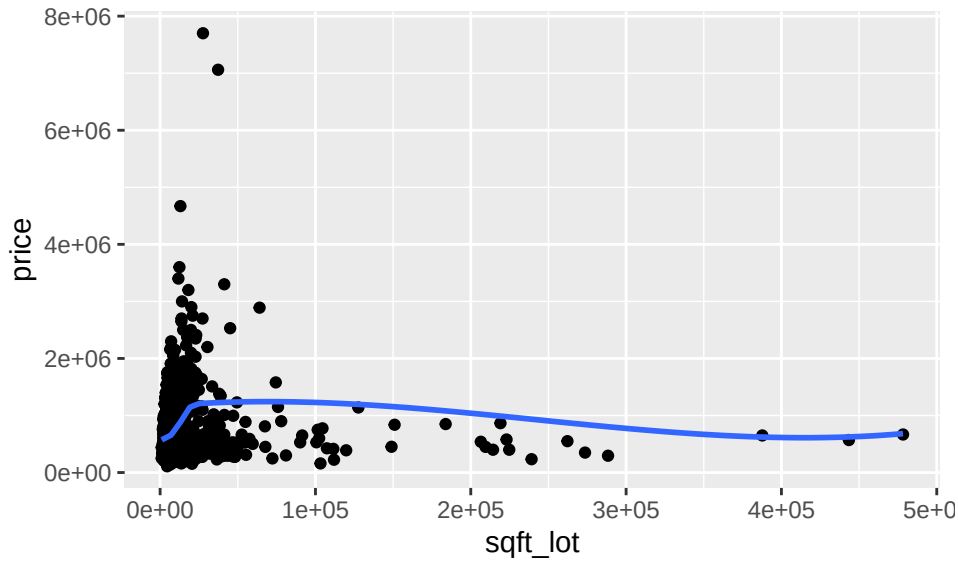

```
ggplot(theData2, aes(bathrooms, price, color = bedrooms)) + geom_point() + geom_smooth(se=F)
```

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



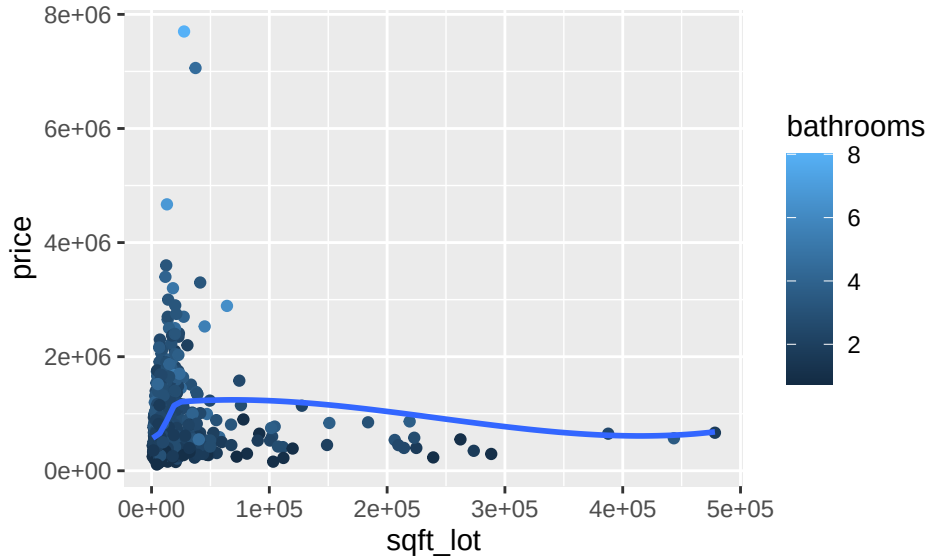

```
ggplot(theData2, aes(sqft_lot, price)) + geom_point() + geom_smooth(se = F)
```

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




```
ggplot(theData2, aes(sqft_lot, price, color = bathrooms)) + geom_point() + geom_smooth(se=F)
```

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




```
ggplot(theData2, aes(sqft_lot, price, color = condition)) + geom_point() + geom_smooth(se=F)
```

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

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 8528.8
```

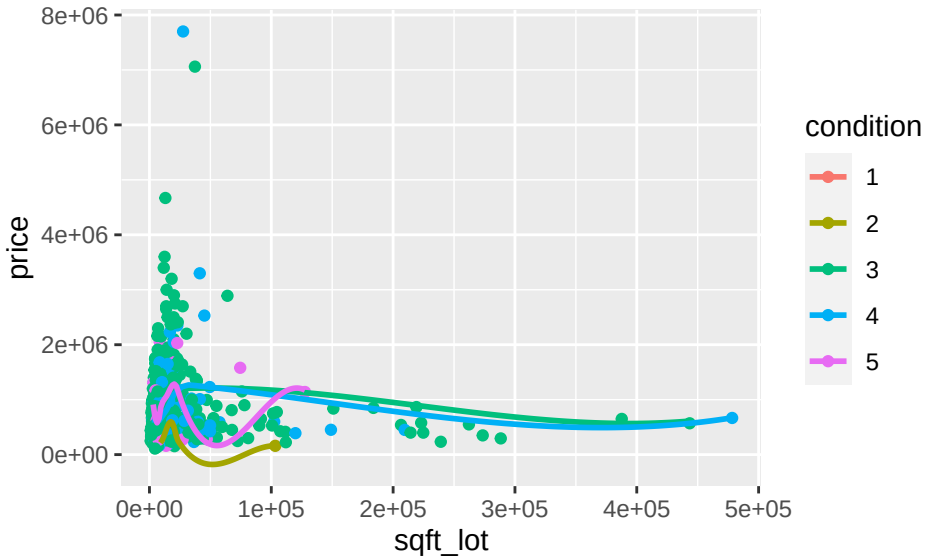
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 14993
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
```

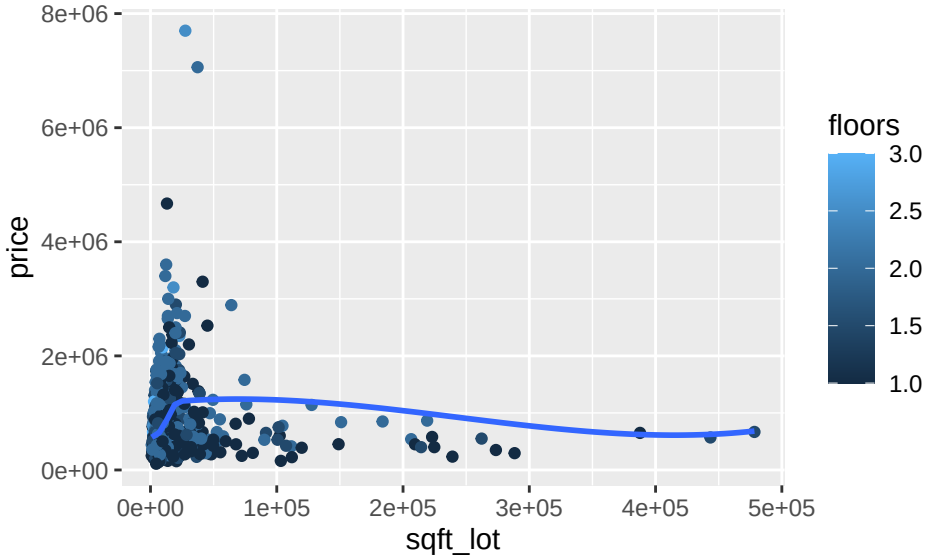
```
## parametric, : There are other near singularities as well. 7.2179e+09
```

```
## Warning in sqrt(sum.squares/one.delta): NaNs produced
```




```
ggplot(theDataZ, aes(sqft_lot, price, color = floors)) + geom_point() + geom_smooth(se=F)
```

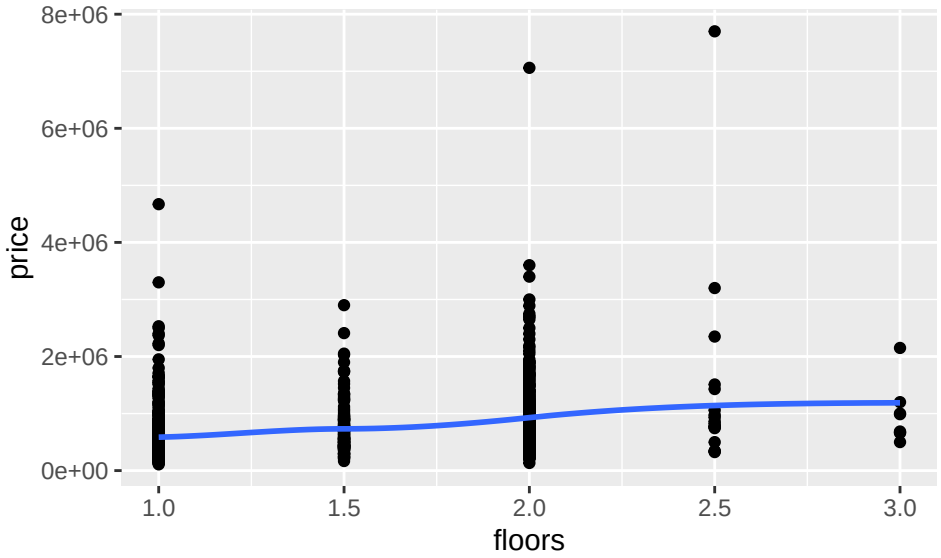
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(theDataZ, aes(floors, price)) + geom_point() + geom_smooth(se = F)
```

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

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.99

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.01

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 9.0365e-16

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
```

```
## parametric, : There are other near singularities as well. 1
```




```
ggplot(theData2, aes(floors, price, color = bedrooms)) + geom_point() + geom_smooth(se = F)
```

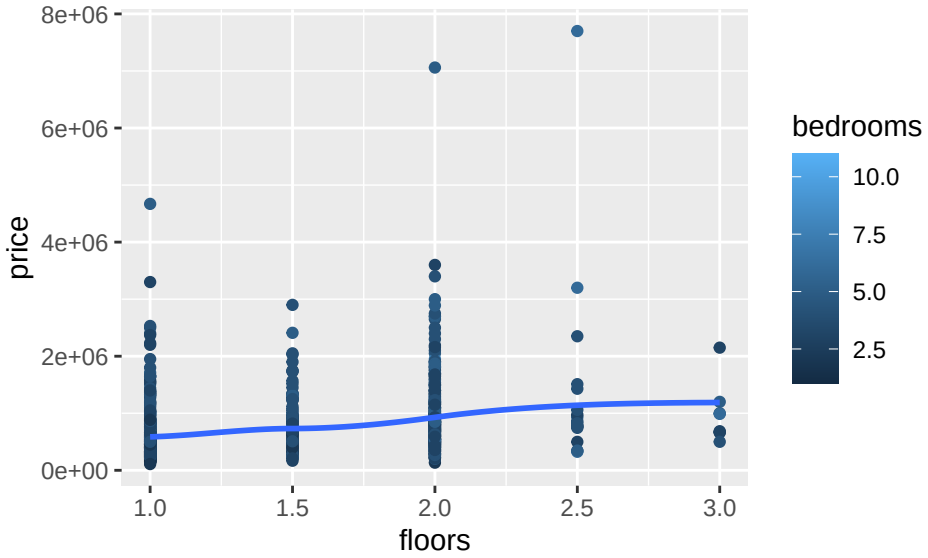
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.99
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.01
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 9.0365e-16
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1
```




```
ggplot(theData2, aes(floors, price, color = sqft_lot)) + geom_point() + geom_smooth(se = F)
```

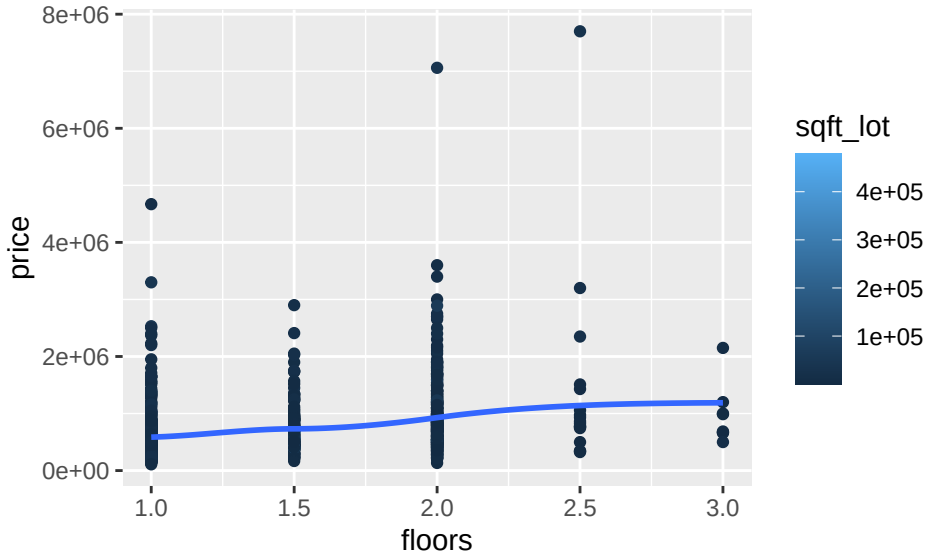
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.99
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.01
```

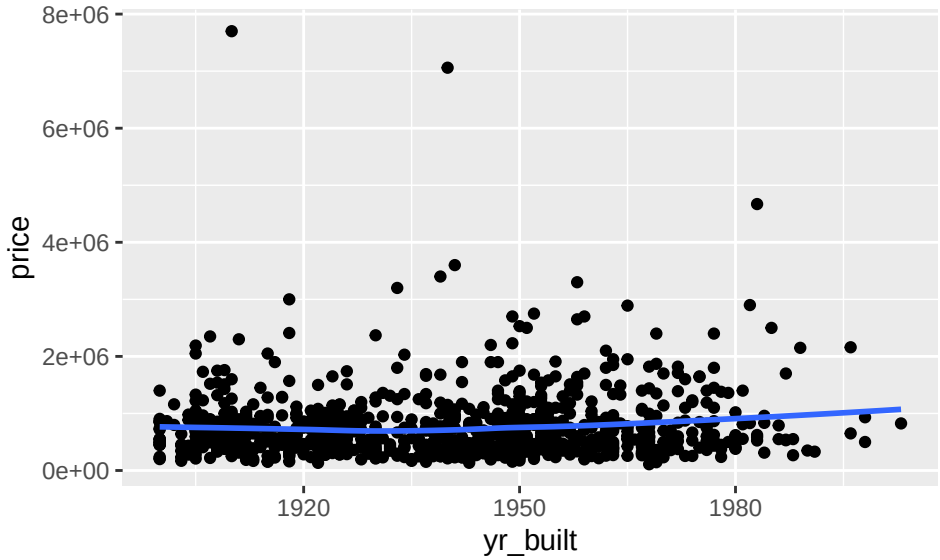
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 9.0365e-16
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1
```




```
ggplot(theData2, aes(yr_built, price)) + geom_point() +  geom_smooth(se = F)
```

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




```
ggplot(theData2, aes(yr_built, price, color = condition)) + geom_point() +  geom_smooth(se = F)
```

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

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
```

```
## parametric, : span too small. fewer data values than degrees of freedom.
```

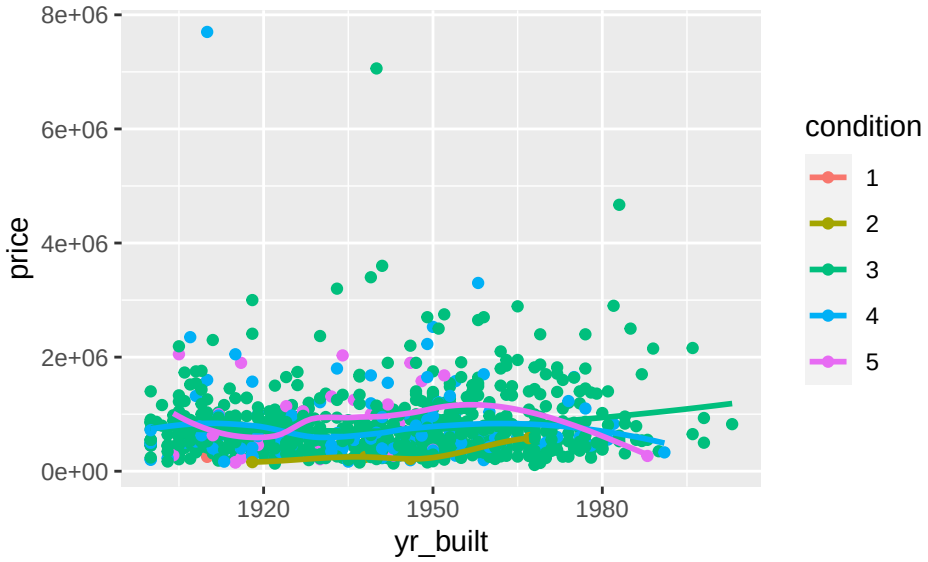
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1917.8
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 28.245
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

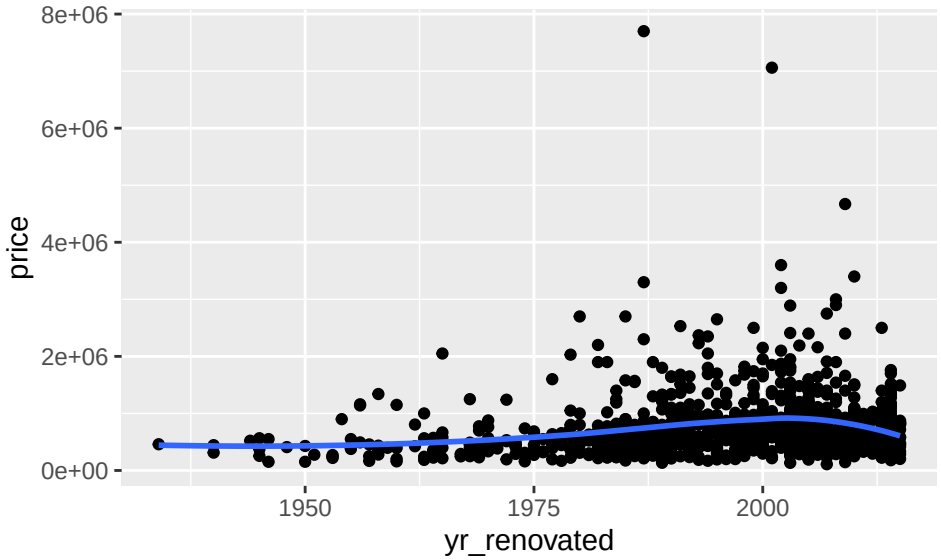
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 855.27
```

```
## Warning in sqrt(sum.squares/one.delta): NaNs produced
```




```
ggplot(theData2, aes(yr_renovated, price)) + geom_point() + geom_smooth(se=F)
```

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



```
ggplot(theData2, aes(yr_renovated, price, color=condition)) + geom_point() + geom_smooth(se=F)
```

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

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

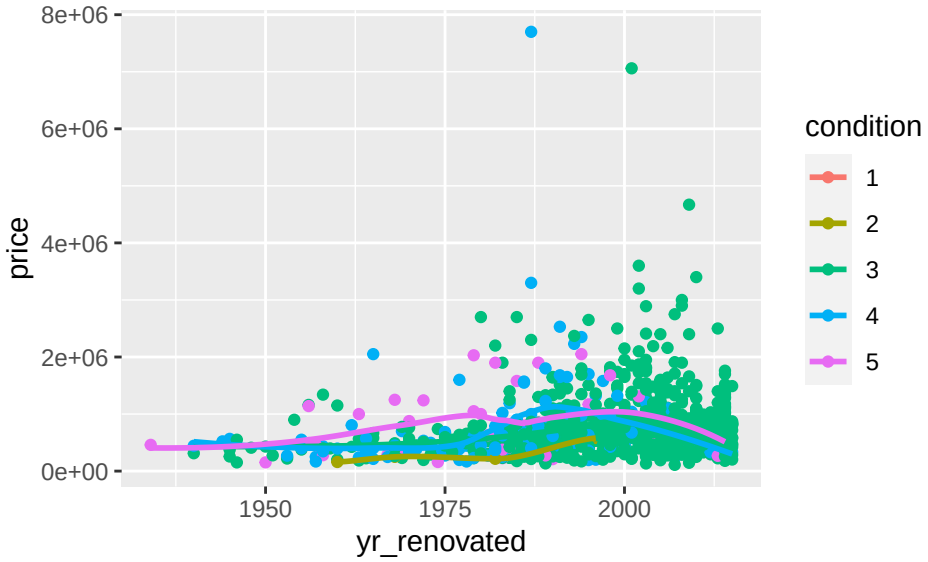
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1959.8
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 22.18
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 794.11
```

```
## Warning in sqrt(sum.squares/one.delta): NaNs produced
```




```
#gf_point or gf_boxplot #or if trying to do many at once
```

COMMENT on what you see!

You can do scatterplots by color, etc. to do more than bivariate relationships. Other options include faceting to include more variables.

We definitely expected there to be stronger relationships between price and condition, year built, numbers of bedrooms, and the total square feet of the lot. There are some clear outliers in the plots between price and sqft_above and in the plot between price and sqft_basement which might be making the relationship appear stronger than it actually is. We were particularly surprised to see that the relationship between sqft_lot and price wasn't as strong as we had thought it would have been. Again, we would have to examine the impact that condition had on the high selling price of certain homes despite a lower total lot area. We also need to examine the data and figure out whether all the data points were actually homes since there was one data point that had 33 bedrooms (which leads us to believe that it might be an apartment complex).

Randomization-Based Procedure Thoughts One potential randomization-based procedure we could try to use is a randomization F-test. We can try multiple different forms of randomization F-tests based on what our earlier analysis indicates and depending on whether we choose an additive or interaction model. We

can test for the main effects of each individual predictor by shuffling them, or also test for the overall model by shuffling the response. If we do end up choosing an interaction model, we can test for just the interaction effect while leaving the main effects intact by fitting the model without interaction and then permuting the residuals before adding them back to the fitted values. This allows us to see how the F-statistic of the original fit compares to the permuted stats.

Thoughts on what procedure you might like to try; aim to use, etc.

Questions for me

Any questions you have for me.