## Multiple Linear Regression

In this example, we demonstrate multiple linear regression model-building techniques. The dataset is the Ames (Iowa) home sale price data. To make things more manageable, this large dataset is reduced to eleven variables (ten predictors and one response).

Begin by loading necessary packages. As usual, you will need to install any of these packages that have not been previously installed.

library(tidyverse) #tidyverse set of packages and functions

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.1.8  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 1.0.0 ──  
## ✔ broom 1.0.3 ✔ rsample 1.1.1  
## ✔ dials 1.1.0 ✔ tune 1.0.1  
## ✔ infer 1.0.4 ✔ workflows 1.1.2  
## ✔ modeldata 1.0.1 ✔ workflowsets 1.0.0  
## ✔ parsnip 1.0.3 ✔ yardstick 1.1.0  
## ✔ recipes 1.0.4   
## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ scales::discard() masks purrr::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ recipes::fixed() masks stringr::fixed()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ yardstick::spec() masks readr::spec()  
## ✖ recipes::step() masks stats::step()  
## • Learn how to get started at https://www.tidymodels.org/start/

library(glmnet) #for Lasso, ridge, and elastic net models

## Loading required package: Matrix  
##   
## Attaching package: 'Matrix'  
##   
## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack  
##   
## Loaded glmnet 4.1-6

library(GGally) #create ggcorr and ggpairs plots

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(ggcorrplot) #create an alternative to ggcorr plots  
library(MASS) #access to forward and backward selection algorithms

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select

library(leaps) #best subset selection  
library(lmtest) #for the dw test

## Loading required package: zoo  
##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(splines) #for nonlinear fitting  
library(car) #for calculating the variance inflation factor

## Loading required package: carData  
##   
## Attaching package: 'car'  
##   
## The following object is masked from 'package:dplyr':  
##   
## recode  
##   
## The following object is masked from 'package:purrr':  
##   
## some

Read in the data.

ames = read\_csv("AmesData.csv")

## Rows: 1460 Columns: 81  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (43): MSZoning, Street, Alley, LotShape, LandContour, Utilities, LotConf...  
## dbl (38): Id, MSSubClass, LotFrontage, LotArea, OverallQual, OverallCond, Ye...  
##   
## 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.

I’ve gone ahead and selected some important variables. Store these variables in a data frame named “ames2”. You may discard the “ames” data frame if you wish. However, since we are dealing with relatively small datasets, it will be OK to keep both datasets in our RStudio environment.

ames2 = ames %>% dplyr::select("SalePrice", "OverallQual", "GrLivArea", "GarageCars", "GarageArea", "TotalBsmtSF", "1stFlrSF", "FullBath", "YearBuilt", "YearRemodAdd", "TotRmsAbvGrd", "Neighborhood")

**IMPORTANT NOTE ABOUT ABOVE CODE**: If you carefully read the line of code above you would notice that I used “dplyr::select” rather than the usual “select”. I had to do this because the “MASS” and “dplyr” (part of the Tidyverse) packages both have a function called “select”. R does not know which function I am trying to use. I used “dplyr::select” to indicate that I wanted to use the “select” function from the “dplyr” package.

Summarize and examine the structure of the data. See the note about the use of “str” for data that is brought into R via the “read\_csv” function (part of the Tidyverse).

#str(ames) #the "read\_csv" function creates attributes that persist even after manipulation (use of select in this case), this makes "str" pretty messy  
summary(ames2) #statistical summary

## SalePrice OverallQual GrLivArea GarageCars   
## Min. : 34900 Min. : 1.000 Min. : 334 Min. :0.000   
## 1st Qu.:129975 1st Qu.: 5.000 1st Qu.:1130 1st Qu.:1.000   
## Median :163000 Median : 6.000 Median :1464 Median :2.000   
## Mean :180921 Mean : 6.099 Mean :1515 Mean :1.767   
## 3rd Qu.:214000 3rd Qu.: 7.000 3rd Qu.:1777 3rd Qu.:2.000   
## Max. :755000 Max. :10.000 Max. :5642 Max. :4.000   
## GarageArea TotalBsmtSF 1stFlrSF FullBath   
## Min. : 0.0 Min. : 0.0 Min. : 334 Min. :0.000   
## 1st Qu.: 334.5 1st Qu.: 795.8 1st Qu.: 882 1st Qu.:1.000   
## Median : 480.0 Median : 991.5 Median :1087 Median :2.000   
## Mean : 473.0 Mean :1057.4 Mean :1163 Mean :1.565   
## 3rd Qu.: 576.0 3rd Qu.:1298.2 3rd Qu.:1391 3rd Qu.:2.000   
## Max. :1418.0 Max. :6110.0 Max. :4692 Max. :3.000   
## YearBuilt YearRemodAdd TotRmsAbvGrd Neighborhood   
## Min. :1872 Min. :1950 Min. : 2.000 Length:1460   
## 1st Qu.:1954 1st Qu.:1967 1st Qu.: 5.000 Class :character   
## Median :1973 Median :1994 Median : 6.000 Mode :character   
## Mean :1971 Mean :1985 Mean : 6.518   
## 3rd Qu.:2000 3rd Qu.:2004 3rd Qu.: 7.000   
## Max. :2010 Max. :2010 Max. :14.000

glimpse(ames2) #use of glimpse to hide the read\_csv attributes (there are a bunch of them because of the 81 columns in original data)

## Rows: 1,460  
## Columns: 12  
## $ SalePrice <dbl> 208500, 181500, 223500, 140000, 250000, 143000, 307000, 2…  
## $ OverallQual <dbl> 7, 6, 7, 7, 8, 5, 8, 7, 7, 5, 5, 9, 5, 7, 6, 7, 6, 4, 5, …  
## $ GrLivArea <dbl> 1710, 1262, 1786, 1717, 2198, 1362, 1694, 2090, 1774, 107…  
## $ GarageCars <dbl> 2, 2, 2, 3, 3, 2, 2, 2, 2, 1, 1, 3, 1, 3, 1, 2, 2, 2, 2, …  
## $ GarageArea <dbl> 548, 460, 608, 642, 836, 480, 636, 484, 468, 205, 384, 73…  
## $ TotalBsmtSF <dbl> 856, 1262, 920, 756, 1145, 796, 1686, 1107, 952, 991, 104…  
## $ `1stFlrSF` <dbl> 856, 1262, 920, 961, 1145, 796, 1694, 1107, 1022, 1077, 1…  
## $ FullBath <dbl> 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 1, 3, 1, 2, 1, 1, 1, 2, 1, …  
## $ YearBuilt <dbl> 2003, 1976, 2001, 1915, 2000, 1993, 2004, 1973, 1931, 193…  
## $ YearRemodAdd <dbl> 2003, 1976, 2002, 1970, 2000, 1995, 2005, 1973, 1950, 195…  
## $ TotRmsAbvGrd <dbl> 8, 6, 6, 7, 9, 5, 7, 7, 8, 5, 5, 11, 4, 7, 5, 5, 5, 6, 6,…  
## $ Neighborhood <chr> "CollgCr", "Veenker", "CollgCr", "Crawfor", "NoRidge", "M…

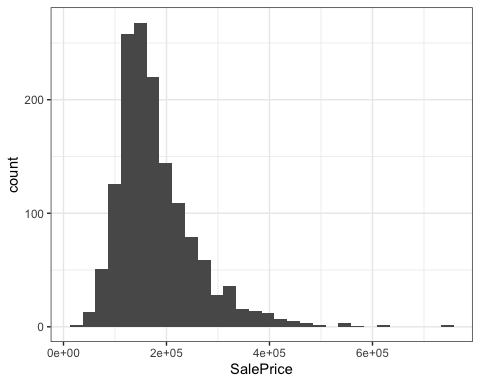
ames2 = ames2 %>% mutate(Neighborhood = as\_factor(Neighborhood))

### Data Exploration

Begin exploring the data by looking at a plot of our response variable only (choose a histogram for a single quantitative variable)

ggplot(ames2, aes(x=SalePrice)) + geom\_histogram() + theme\_bw()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

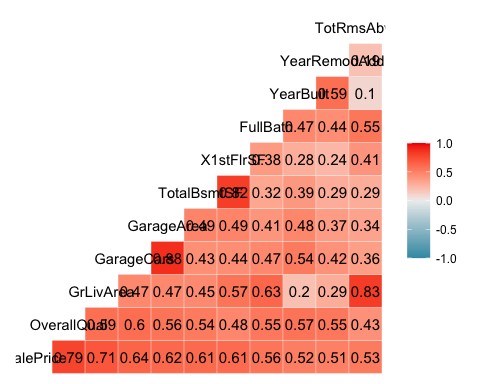


Notice that “SalePrice” is reported in scientific notation (e.g., 2e+05 = 200,000). The data is somewhat skewed with most homes having “SalePrice” values less than about $300,000. There are some (but not many) very expensive homes with high prices. This brings us to a point of having to make a decision: Do we care about trying to predict the prices of these outlier (high price) homes? If not, we could make a reasonable argument to remove them from our dataset. If we wish to keep these homes in our dataset, we should be prepared to deal with modeling results that might not be as strong as we would hope. One option would be to transform the SalesPrice variable. A logarithmic transformation would likely make the variable more Normal (nice, but not required) and make it very unlikely that we would have a negative SalesPrice prediction.

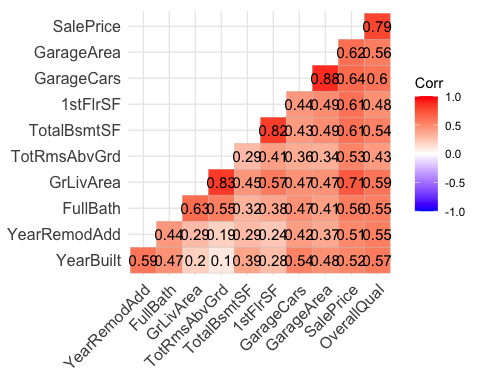
Next we look at correlation. This is a logical step since almost all of our variables are quantitative.

#use "ggcorr" to create a correlation matrix with labels and correlation reported to two decimals  
ggcorr(ames2, label = "TRUE", label\_round = 2)

## Warning in ggcorr(ames2, label = "TRUE", label\_round = 2): data in column(s)  
## 'Neighborhood' are not numeric and were ignored

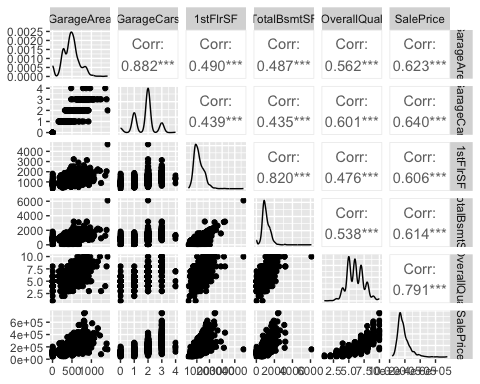


#Alternative using the "ggcorrplot" function  
corr = round(cor(ames2[,1:11]), 2) #Note the ,1:11 code to select the columns for inclusion  
ggcorrplot(corr, hc.order = TRUE, type = "lower",  
 lab = TRUE)

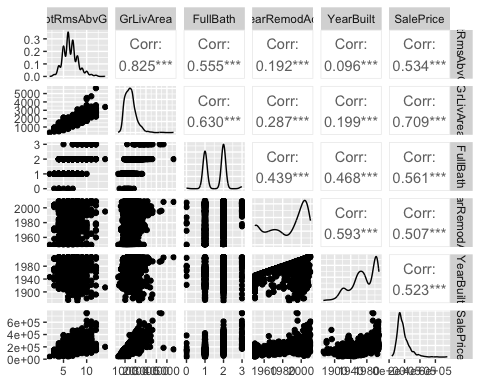
 All of the variables in the dataset are positively correlated with the response variable (SalePrice). They are also all positively correlated with each other.

Use the “ggpairs” function to plot all of the variables. There are too many variables to easily display in a single “ggpairs” plot. To see what I mean you can try to run ggpairs(ames2) and see what you get.

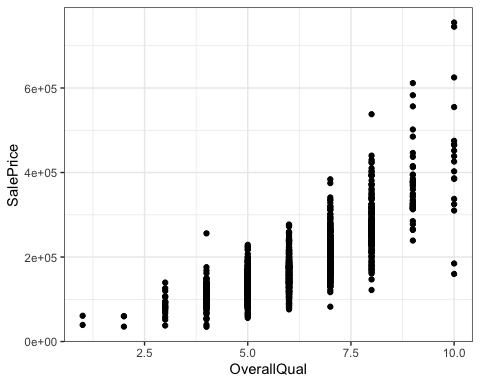
ggpairs(ames2, columns = c("GarageArea", "GarageCars","1stFlrSF", "TotalBsmtSF", "OverallQual","SalePrice"))



ggpairs(ames2, columns = c("TotRmsAbvGrd","GrLivArea","FullBath","YearRemodAdd","YearBuilt","SalePrice"))

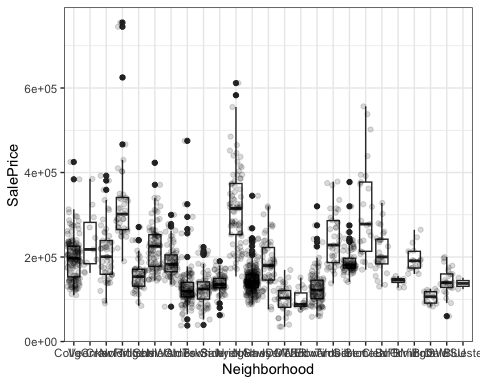
 We look primarily, at the bottom row in each of the “ggpairs” plots. Here we see that as each potential predictor increases, the “SalePrice” variable increases also. This is as we would expect from looking at the correlation matrix. Notice by carefully looking at the plots in the last rows, we can see that the this increase may not be linear. To see an example of this in better detail, examine the plot below for “SalePrice” and “OverallQual”.

ggplot(ames2, aes(x=OverallQual, y=SalePrice)) + geom\_point() + theme\_bw() #I like the clean look from theme\_bw()

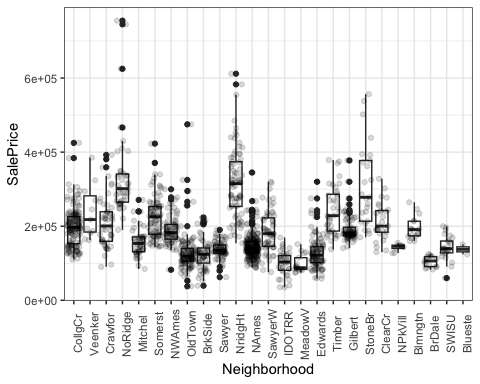
 In the plot above, there looks to be a bit of curvature to the data. We’ll see this in a more pronounced way when we build our first model.

Let’s finish up by looking at the Neighborhood variable versus SalePrice. I like boxplots for this situation.

ggplot(ames2,aes(x=Neighborhood,y=SalePrice)) + geom\_boxplot() + geom\_jitter(alpha = 0.15) + theme\_bw()

 Hard to see much here (Neighborhood clearly matters, but we can’t read the labels). Let’s rotate the labels.

ggplot(ames2,aes(x=Neighborhood,y=SalePrice)) + geom\_boxplot() + geom\_jitter(alpha = 0.15) + theme\_bw() +   
 theme(axis.text.x = element\_text(angle = 90))

 It’s clear that some neighborhoods are more expensive than others and that some neighborhoods have few homes. Let’s look at a table of counts of sales by neighborhood.

table(ames2$Neighborhood)

##   
## CollgCr Veenker Crawfor NoRidge Mitchel Somerst NWAmes OldTown BrkSide Sawyer   
## 150 11 51 41 49 86 73 113 58 74   
## NridgHt NAmes SawyerW IDOTRR MeadowV Edwards Timber Gilbert StoneBr ClearCr   
## 77 225 59 37 17 100 38 79 25 28   
## NPkVill Blmngtn BrDale SWISU Blueste   
## 9 17 16 25 2

We can sort (the Tidy way):

ames %>% group\_by(Neighborhood) %>% summarize(freq = n()) %>% arrange(desc(freq))

## # A tibble: 25 × 2  
## Neighborhood freq  
## <chr> <int>  
## 1 NAmes 225  
## 2 CollgCr 150  
## 3 OldTown 113  
## 4 Edwards 100  
## 5 Somerst 86  
## 6 Gilbert 79  
## 7 NridgHt 77  
## 8 Sawyer 74  
## 9 NWAmes 73  
## 10 SawyerW 59  
## # … with 15 more rows

It may be useful to collapse some of the infrequently occurring neighborhoods into a catch-all “Other” group. We’ll look at this in a minute.

### Models

The first model we’ll build uses the variable that is best correlated with “SalePrice”, “OverallQual”. This is a univariate (simple) linear regression model. We also plot this model.

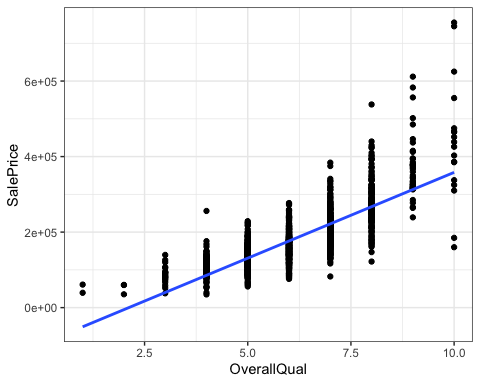
ames\_recipe = recipe(SalePrice ~ OverallQual, ames2)  
  
lm\_model = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use   
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(ames\_recipe)  
  
lm\_fit = fit(lm\_wflow, ames2)

summary(lm\_fit$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -198152 -29409 -1845 21463 396848   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -96206.1 5756.4 -16.71 <2e-16 \*\*\*  
## OverallQual 45435.8 920.4 49.36 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 48620 on 1458 degrees of freedom  
## Multiple R-squared: 0.6257, Adjusted R-squared: 0.6254   
## F-statistic: 2437 on 1 and 1458 DF, p-value: < 2.2e-16

ggplot(ames2, aes(x=OverallQual, y=SalePrice)) + geom\_point() + geom\_smooth(method = lm, se = FALSE) + theme\_bw()

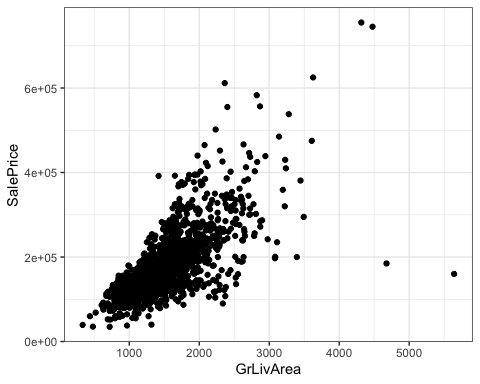
## `geom\_smooth()` using formula = 'y ~ x'

 Notice that this model tends to underpredict low and high quality homes. Homes with intermediate quality (between roughly 4 and 8) seem to predicted fairly well.

### Multivariate Regression

Now let’s move to models with more than one predictor (x) variable. We’ll add GrLivArea to the model by modifying the recipe. First let’s look at the relationship with the response.

ggplot(ames2, aes(x=GrLivArea, y=SalePrice)) + geom\_point() + theme\_bw()



ames\_recipe = recipe(SalePrice ~ OverallQual + GrLivArea, ames2)  
  
lm\_model = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use   
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(ames\_recipe)  
  
lm\_fit2 = fit(lm\_wflow, ames2)

summary(lm\_fit2$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -379572 -22266 -386 19895 289501   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -104092.67 5045.37 -20.63 <2e-16 \*\*\*  
## OverallQual 32849.05 999.20 32.88 <2e-16 \*\*\*  
## GrLivArea 55.86 2.63 21.24 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 42500 on 1457 degrees of freedom  
## Multiple R-squared: 0.7142, Adjusted R-squared: 0.7138   
## F-statistic: 1820 on 2 and 1457 DF, p-value: < 2.2e-16

Let’s do our diagnostics.

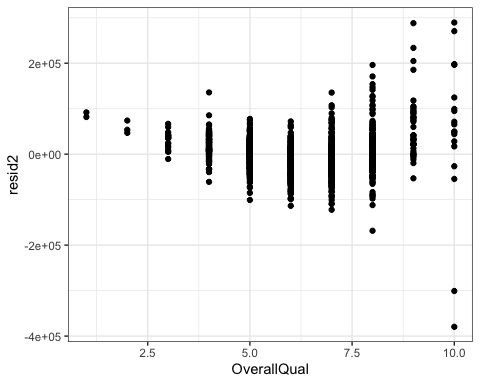
dwtest(lm\_fit2$fit$fit$fit)

##   
## Durbin-Watson test  
##   
## data: lm\_fit2$fit$fit$fit  
## DW = 1.9849, p-value = 0.3865  
## alternative hypothesis: true autocorrelation is greater than 0

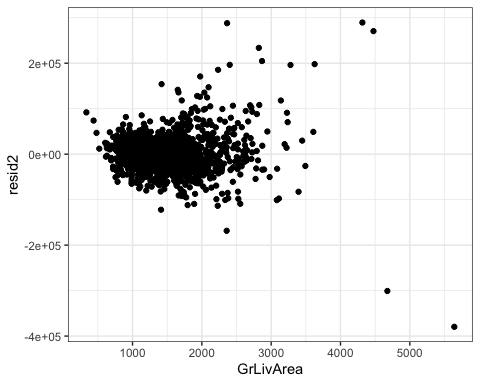
All looks good. We fail to reject the null.

Examine a plot of residuals. Notice we’re doing this all in one line. We are not permanently creating a residual variable.

ames2 %>% mutate(resid2 = lm\_fit2$fit$fit$fit$residuals) %>%   
ggplot(aes(x=OverallQual,y=resid2)) + geom\_point() + theme\_bw()

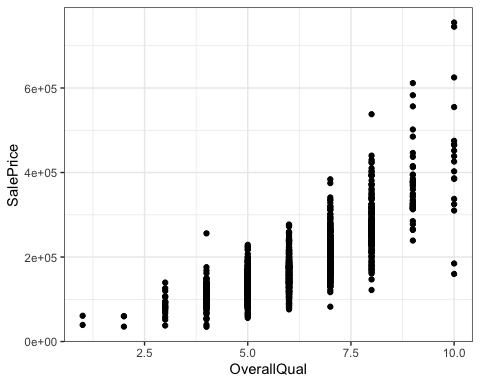


ames2 %>% mutate(resid2 = lm\_fit2$fit$fit$fit$residuals) %>%  
 ggplot(aes(x=GrLivArea,y=resid2)) + geom\_point() + theme\_bw()

 We might need to transform these variables in some way (to linearize them) before we proceed.

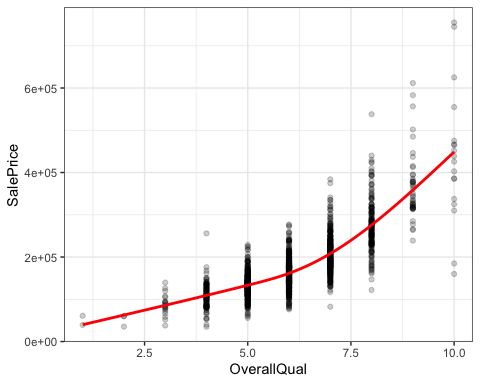
Let’s start with OverallQual.

ggplot(ames2, aes(x=OverallQual, y=SalePrice)) + geom\_point() + theme\_bw()



A reasonable solution might be a spline.

#Code borrowed from the TMWR text  
ggplot(ames2, aes(x = OverallQual, y = SalePrice)) +   
 geom\_point(alpha = .2) +   
 geom\_smooth(  
 method = lm,  
 formula = y ~ ns(x, df = 4),  
 col = "red",  
 se = FALSE) + theme\_bw()



Let’s see what the spline does for our model.

ames\_recipe = recipe(SalePrice ~ OverallQual + GrLivArea, ames2) %>%  
 step\_ns(OverallQual, deg\_free = 4) #add the spline transformation to the recipe  
  
lm\_model = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use   
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(ames\_recipe)  
  
lm\_fit3 = fit(lm\_wflow, ames2)

summary(lm\_fit3$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -439264 -19901 939 17600 250123   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3464.742 16819.632 -0.206 0.837   
## GrLivArea 51.021 2.455 20.782 < 2e-16 \*\*\*  
## OverallQual\_ns\_1 85726.768 16124.457 5.317 1.22e-07 \*\*\*  
## OverallQual\_ns\_2 125305.078 10752.678 11.653 < 2e-16 \*\*\*  
## OverallQual\_ns\_3 296883.956 35208.497 8.432 < 2e-16 \*\*\*  
## OverallQual\_ns\_4 284333.379 10726.563 26.507 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 39310 on 1454 degrees of freedom  
## Multiple R-squared: 0.756, Adjusted R-squared: 0.7551   
## F-statistic: 900.9 on 5 and 1454 DF, p-value: < 2.2e-16

We’ve improved our model, but at a potential loss of interpretability due to the spline coefficients. If this is not an issue (usually isn’t), then we can roll with it.

I ultimately decided not to transform GrLivArea.

Let’s turn our attention to neighborhood.

ames\_recipe2 = recipe(SalePrice ~ Neighborhood, ames2) %>%  
 step\_dummy(all\_nominal())  
  
lm\_model = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use   
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(ames\_recipe2)  
  
lm\_fit4 = fit(lm\_wflow, ames2)

summary(lm\_fit4$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -162271 -27552 -5324 19685 419705   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 197966 4409 44.900 < 2e-16 \*\*\*  
## Neighborhood\_Veenker 40807 16868 2.419 0.015677 \*   
## Neighborhood\_Crawfor 12659 8753 1.446 0.148324   
## Neighborhood\_NoRidge 137330 9516 14.431 < 2e-16 \*\*\*  
## Neighborhood\_Mitchel -41696 8885 -4.693 2.95e-06 \*\*\*  
## Neighborhood\_Somerst 27414 7304 3.753 0.000181 \*\*\*  
## Neighborhood\_NWAmes -8916 7706 -1.157 0.247474   
## Neighborhood\_OldTown -69740 6726 -10.368 < 2e-16 \*\*\*  
## Neighborhood\_BrkSide -73132 8349 -8.759 < 2e-16 \*\*\*  
## Neighborhood\_Sawyer -61173 7671 -7.975 3.10e-15 \*\*\*  
## Neighborhood\_NridgHt 118305 7570 15.628 < 2e-16 \*\*\*  
## Neighborhood\_NAmes -52119 5692 -9.156 < 2e-16 \*\*\*  
## Neighborhood\_SawyerW -11410 8298 -1.375 0.169351   
## Neighborhood\_IDOTRR -97842 9912 -9.871 < 2e-16 \*\*\*  
## Neighborhood\_MeadowV -99389 13819 -7.192 1.02e-12 \*\*\*  
## Neighborhood\_Edwards -69746 6971 -10.005 < 2e-16 \*\*\*  
## Neighborhood\_Timber 44282 9807 4.515 6.84e-06 \*\*\*  
## Neighborhood\_Gilbert -5111 7507 -0.681 0.496044   
## Neighborhood\_StoneBr 112533 11665 9.647 < 2e-16 \*\*\*  
## Neighborhood\_ClearCr 14600 11117 1.313 0.189284   
## Neighborhood\_NPkVill -55271 18532 -2.983 0.002907 \*\*   
## Neighborhood\_Blmngtn -3095 13819 -0.224 0.822820   
## Neighborhood\_BrDale -93472 14202 -6.582 6.50e-11 \*\*\*  
## Neighborhood\_SWISU -55374 11665 -4.747 2.27e-06 \*\*\*  
## Neighborhood\_Blueste -60466 38437 -1.573 0.115911   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 54000 on 1435 degrees of freedom  
## Multiple R-squared: 0.5456, Adjusted R-squared: 0.538   
## F-statistic: 71.78 on 24 and 1435 DF, p-value: < 2.2e-16

There’s a lot to digest here :) As is usual for categorical variables, we have one variable for each level in the Neighborhood variable (minus one). Some of the levels are significant. Some are not. We typically do NOT discard the Neighborhood variable (as a whole) if at least one of its levels is significant.

Let’s add Neighborhood to our prior model.

ames\_recipe = recipe(SalePrice ~ OverallQual + GrLivArea + Neighborhood, ames2) %>%  
 step\_ns(OverallQual, deg\_free = 4) %>% #add the spline transformation to the recipe  
 step\_dummy(all\_nominal())  
   
lm\_model = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use   
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(ames\_recipe)  
  
lm\_fit5 = fit(lm\_wflow, ames2)

summary(lm\_fit5$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -370817 -16366 10 14421 236805   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 53026.601 15602.693 3.399 0.000696 \*\*\*  
## GrLivArea 50.628 2.365 21.406 < 2e-16 \*\*\*  
## OverallQual\_ns\_1 50839.898 14630.624 3.475 0.000526 \*\*\*  
## OverallQual\_ns\_2 64828.907 10299.195 6.295 4.09e-10 \*\*\*  
## OverallQual\_ns\_3 193761.115 32098.706 6.036 2.00e-09 \*\*\*  
## OverallQual\_ns\_4 214889.067 10621.774 20.231 < 2e-16 \*\*\*  
## Neighborhood\_Veenker 29743.547 10831.773 2.746 0.006109 \*\*   
## Neighborhood\_Crawfor 4832.167 5722.704 0.844 0.398595   
## Neighborhood\_NoRidge 43354.904 6509.318 6.660 3.88e-11 \*\*\*  
## Neighborhood\_Mitchel -11715.444 5848.890 -2.003 0.045364 \*   
## Neighborhood\_Somerst 3509.854 4752.303 0.739 0.460296   
## Neighborhood\_NWAmes -11324.211 5069.129 -2.234 0.025640 \*   
## Neighborhood\_OldTown -49945.908 4575.229 -10.917 < 2e-16 \*\*\*  
## Neighborhood\_BrkSide -31790.609 5623.807 -5.653 1.90e-08 \*\*\*  
## Neighborhood\_Sawyer -19502.001 5282.561 -3.692 0.000231 \*\*\*  
## Neighborhood\_NridgHt 35575.791 5372.027 6.622 4.99e-11 \*\*\*  
## Neighborhood\_NAmes -19703.134 3969.289 -4.964 7.74e-07 \*\*\*  
## Neighborhood\_SawyerW -8676.014 5372.343 -1.615 0.106544   
## Neighborhood\_IDOTRR -50316.635 6653.510 -7.562 7.04e-14 \*\*\*  
## Neighborhood\_MeadowV -44257.011 9165.631 -4.829 1.52e-06 \*\*\*  
## Neighborhood\_Edwards -39083.204 4814.723 -8.117 1.02e-15 \*\*\*  
## Neighborhood\_Timber 12525.642 6334.080 1.977 0.048177 \*   
## Neighborhood\_Gilbert -8263.184 4867.493 -1.698 0.089796 .   
## Neighborhood\_StoneBr 40699.390 7745.163 5.255 1.71e-07 \*\*\*  
## Neighborhood\_ClearCr 15478.166 7275.304 2.127 0.033550 \*   
## Neighborhood\_NPkVill -25672.978 12027.168 -2.135 0.032965 \*   
## Neighborhood\_Blmngtn -8200.874 8911.472 -0.920 0.357592   
## Neighborhood\_BrDale -55027.401 9255.676 -5.945 3.46e-09 \*\*\*  
## Neighborhood\_SWISU -48730.886 7738.402 -6.297 4.02e-10 \*\*\*  
## Neighborhood\_Blueste -37930.070 24716.794 -1.535 0.125107   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 34630 on 1430 degrees of freedom  
## Multiple R-squared: 0.8138, Adjusted R-squared: 0.81   
## F-statistic: 215.5 on 29 and 1430 DF, p-value: < 2.2e-16

Also add to our recipe a condition that Neighborhoods that have very few sales in them are collapsed into an “Other” category.

ames\_recipe = recipe(SalePrice ~ OverallQual + GrLivArea + Neighborhood, ames2) %>%  
 step\_ns(OverallQual, deg\_free = 4) %>% #add the spline transformation to the recipe  
 step\_other(Neighborhood, threshold = 0.01) %>% #collapses small Neighborhoods into an "Other" group  
 step\_dummy(all\_nominal()) #makes Neighborhood categorical  
   
   
lm\_model = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use   
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(ames\_recipe)  
  
lm\_fit6 = fit(lm\_wflow, ames2)

lm\_fit6 %>%  
 pull\_workflow\_fit() %>%  
 tidy()

## Warning: `pull\_workflow\_fit()` was deprecated in workflows 0.2.3.  
## ℹ Please use `extract\_fit\_parsnip()` instead.

## # A tibble: 28 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 51173. 15667. 3.27 1.12e- 3  
## 2 GrLivArea 50.7 2.38 21.3 6.57e-88  
## 3 OverallQual\_ns\_1 51453. 14697. 3.50 4.78e- 4  
## 4 OverallQual\_ns\_2 66725. 10335. 6.46 1.47e-10  
## 5 OverallQual\_ns\_3 198792. 32221. 6.17 8.89e-10  
## 6 OverallQual\_ns\_4 216024. 10667. 20.3 2.15e-80  
## 7 Neighborhood\_Crawfor 5032. 5749. 0.875 3.82e- 1  
## 8 Neighborhood\_NoRidge 42755. 6537. 6.54 8.54e-11  
## 9 Neighborhood\_Mitchel -11350. 5875. -1.93 5.36e- 2  
## 10 Neighborhood\_Somerst 3264. 4774. 0.684 4.94e- 1  
## # … with 18 more rows

summary(lm\_fit6$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -372040 -16607 29 14450 236523   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 51173.148 15667.312 3.266 0.001116 \*\*   
## GrLivArea 50.683 2.376 21.333 < 2e-16 \*\*\*  
## OverallQual\_ns\_1 51452.783 14697.335 3.501 0.000478 \*\*\*  
## OverallQual\_ns\_2 66725.399 10335.204 6.456 1.47e-10 \*\*\*  
## OverallQual\_ns\_3 198792.098 32220.798 6.170 8.89e-10 \*\*\*  
## OverallQual\_ns\_4 216024.290 10666.806 20.252 < 2e-16 \*\*\*  
## Neighborhood\_Crawfor 5032.074 5748.896 0.875 0.381551   
## Neighborhood\_NoRidge 42754.719 6537.445 6.540 8.54e-11 \*\*\*  
## Neighborhood\_Mitchel -11349.683 5875.137 -1.932 0.053579 .   
## Neighborhood\_Somerst 3264.232 4773.826 0.684 0.494227   
## Neighborhood\_NWAmes -10995.667 5091.835 -2.159 0.030979 \*   
## Neighborhood\_OldTown -49722.122 4595.994 -10.819 < 2e-16 \*\*\*  
## Neighborhood\_BrkSide -31394.140 5648.845 -5.558 3.26e-08 \*\*\*  
## Neighborhood\_Sawyer -19183.659 5306.317 -3.615 0.000310 \*\*\*  
## Neighborhood\_NridgHt 34957.518 5394.462 6.480 1.26e-10 \*\*\*  
## Neighborhood\_NAmes -19323.786 3986.415 -4.847 1.39e-06 \*\*\*  
## Neighborhood\_SawyerW -8444.440 5396.821 -1.565 0.117873   
## Neighborhood\_IDOTRR -50021.975 6683.798 -7.484 1.25e-13 \*\*\*  
## Neighborhood\_MeadowV -44109.907 9207.875 -4.790 1.84e-06 \*\*\*  
## Neighborhood\_Edwards -38802.657 4836.416 -8.023 2.13e-15 \*\*\*  
## Neighborhood\_Timber 12272.423 6362.985 1.929 0.053963 .   
## Neighborhood\_Gilbert -8042.339 4889.639 -1.645 0.100236   
## Neighborhood\_StoneBr 40000.659 7778.825 5.142 3.09e-07 \*\*\*  
## Neighborhood\_ClearCr 15758.358 7308.540 2.156 0.031238 \*   
## Neighborhood\_Blmngtn -8399.850 8952.474 -0.938 0.348263   
## Neighborhood\_BrDale -54403.673 9297.004 -5.852 6.02e-09 \*\*\*  
## Neighborhood\_SWISU -48324.856 7773.431 -6.217 6.65e-10 \*\*\*  
## Neighborhood\_other 1324.643 8015.769 0.165 0.868767   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 34790 on 1432 degrees of freedom  
## Multiple R-squared: 0.8118, Adjusted R-squared: 0.8083   
## F-statistic: 228.8 on 27 and 1432 DF, p-value: < 2.2e-16

Let’s try a model with all predictors and see what happens.

ames\_recipe = recipe(SalePrice ~., ames2) %>%  
 step\_ns(OverallQual, deg\_free = 4) %>% #add the spline transformation to the recipe  
 step\_other(Neighborhood, threshold = 0.01) %>% #collapses small Neighborhoods into an "Other" group  
 step\_dummy(all\_nominal()) #makes Neighborhood categorical  
   
   
lm\_model = #give the model type a name   
 linear\_reg() %>% #specify that we are doing linear regression  
 set\_engine("lm") #specify the specify type of linear tool we want to use   
  
lm\_wflow =   
 workflow() %>%   
 add\_model(lm\_model) %>%   
 add\_recipe(ames\_recipe)  
  
lm\_fit7 = fit(lm\_wflow, ames2)

lm\_fit7 %>%  
 pull\_workflow\_fit() %>%  
 tidy()

## # A tibble: 36 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -1020974. 157230. -6.49 1.16e-10  
## 2 GrLivArea 42.7 3.97 10.8 5.00e-26  
## 3 GarageCars 10409. 2772. 3.76 1.80e- 4  
## 4 GarageArea 4.25 9.43 0.451 6.52e- 1  
## 5 TotalBsmtSF 14.8 3.78 3.90 1.02e- 4  
## 6 `1stFlrSF` 4.88 4.55 1.07 2.84e- 1  
## 7 FullBath -2109. 2438. -0.865 3.87e- 1  
## 8 YearBuilt 198. 68.3 2.90 3.78e- 3  
## 9 YearRemodAdd 335. 57.4 5.83 6.70e- 9  
## 10 TotRmsAbvGrd 688. 1008. 0.682 4.95e- 1  
## # … with 26 more rows

options(scipen = 999)  
summary(lm\_fit7$fit$fit$fit)

##   
## Call:  
## stats::lm(formula = ..y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -435348 -13633 257 12608 223324   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1020973.791 157229.899 -6.494 0.00000000011555 \*\*\*  
## GrLivArea 42.744 3.971 10.763 < 0.0000000000000002 \*\*\*  
## GarageCars 10409.160 2771.502 3.756 0.000180 \*\*\*  
## GarageArea 4.249 9.431 0.451 0.652416   
## TotalBsmtSF 14.752 3.785 3.898 0.000102 \*\*\*  
## `1stFlrSF` 4.881 4.553 1.072 0.283893   
## FullBath -2108.590 2437.517 -0.865 0.387153   
## YearBuilt 197.957 68.252 2.900 0.003784 \*\*   
## YearRemodAdd 334.863 57.401 5.834 0.00000000669982 \*\*\*  
## TotRmsAbvGrd 687.747 1007.834 0.682 0.495096   
## OverallQual\_ns\_1 29207.848 14093.651 2.072 0.038407 \*   
## OverallQual\_ns\_2 39262.605 10044.126 3.909 0.00009702979196 \*\*\*  
## OverallQual\_ns\_3 139265.305 31016.253 4.490 0.00000769427742 \*\*\*  
## OverallQual\_ns\_4 170301.043 10756.566 15.832 < 0.0000000000000002 \*\*\*  
## Neighborhood\_Crawfor 27869.795 6456.204 4.317 0.00001692871847 \*\*\*  
## Neighborhood\_NoRidge 46549.310 6294.470 7.395 0.00000000000024 \*\*\*  
## Neighborhood\_Mitchel -8095.646 5632.628 -1.437 0.150859   
## Neighborhood\_Somerst 2964.948 4561.506 0.650 0.515801   
## Neighborhood\_NWAmes -2178.792 5043.617 -0.432 0.665814   
## Neighborhood\_OldTown -21643.056 6165.262 -3.510 0.000461 \*\*\*  
## Neighborhood\_BrkSide -2785.775 6612.295 -0.421 0.673598   
## Neighborhood\_Sawyer -7676.309 5355.640 -1.433 0.151988   
## Neighborhood\_NridgHt 29586.779 5129.271 5.768 0.00000000981218 \*\*\*  
## Neighborhood\_NAmes -4489.865 4417.419 -1.016 0.309612   
## Neighborhood\_SawyerW -2256.685 5159.692 -0.437 0.661911   
## Neighborhood\_IDOTRR -18808.576 7502.485 -2.507 0.012287 \*   
## Neighborhood\_MeadowV -23519.497 8889.608 -2.646 0.008241 \*\*   
## Neighborhood\_Edwards -18466.165 5151.741 -3.584 0.000349 \*\*\*  
## Neighborhood\_Timber 12364.727 6080.309 2.034 0.042180 \*   
## Neighborhood\_Gilbert -2627.727 4790.584 -0.549 0.583422   
## Neighborhood\_StoneBr 42056.830 7394.638 5.687 0.00000001562294 \*\*\*  
## Neighborhood\_ClearCr 25733.752 7158.128 3.595 0.000335 \*\*\*  
## Neighborhood\_Blmngtn -16524.866 8612.242 -1.919 0.055214 .   
## Neighborhood\_BrDale -28504.567 9041.991 -3.152 0.001653 \*\*   
## Neighborhood\_SWISU -12509.250 8471.718 -1.477 0.140007   
## Neighborhood\_other 8535.295 7699.075 1.109 0.267784   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 32840 on 1424 degrees of freedom  
## Multiple R-squared: 0.8332, Adjusted R-squared: 0.8291   
## F-statistic: 203.2 on 35 and 1424 DF, p-value: < 0.00000000000000022

options(scipen = 0)

Checking autocorrelation of residuals

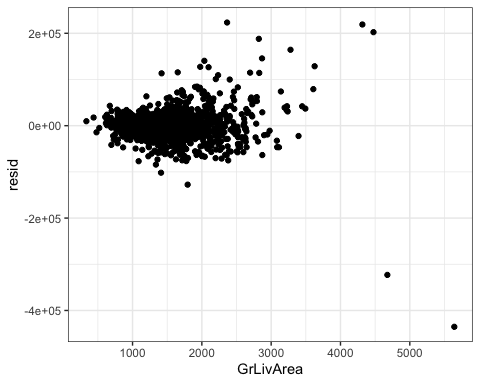
dwtest(lm\_fit7$fit$fit$fit)

##   
## Durbin-Watson test  
##   
## data: lm\_fit7$fit$fit$fit  
## DW = 1.9443, p-value = 0.1432  
## alternative hypothesis: true autocorrelation is greater than 0

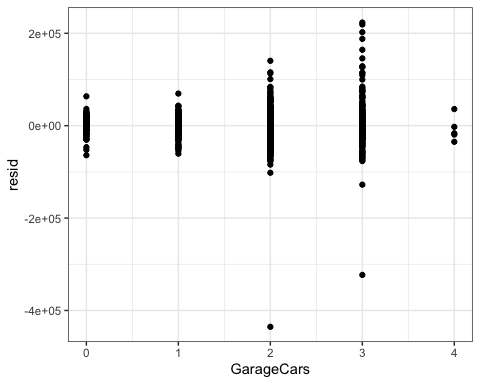
No evidence of autocorrelation.

Examine residuals for each numeric variable:

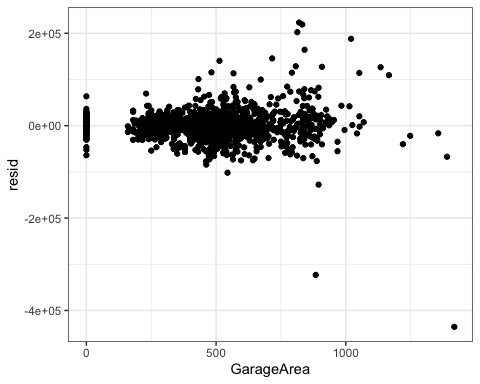
ames2 %>% mutate(resid = lm\_fit7$fit$fit$fit$residuals) %>%   
ggplot(aes(x=GrLivArea,y=resid)) + geom\_point() + theme\_bw()



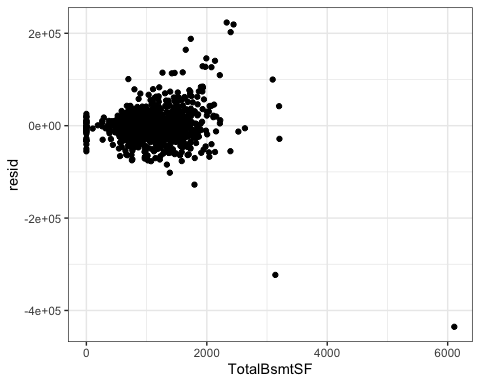
ames2 %>% mutate(resid = lm\_fit7$fit$fit$fit$residuals) %>%   
ggplot(aes(x=GarageCars,y=resid)) + geom\_point() + theme\_bw()



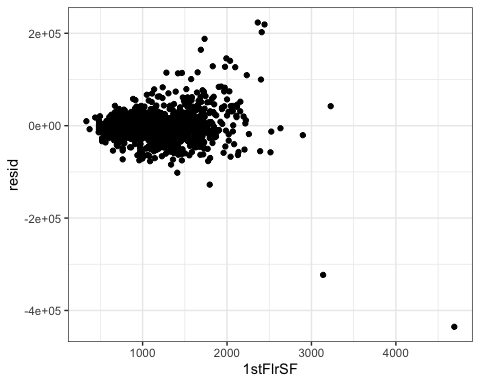
ames2 %>% mutate(resid = lm\_fit7$fit$fit$fit$residuals) %>%   
ggplot(aes(x=GarageArea,y=resid)) + geom\_point() + theme\_bw()



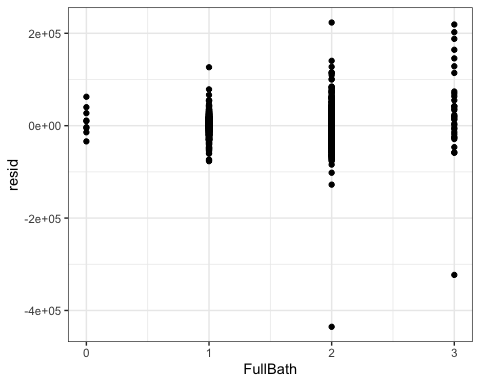
ames2 %>% mutate(resid = lm\_fit7$fit$fit$fit$residuals) %>%   
ggplot(aes(x=TotalBsmtSF,y=resid)) + geom\_point() + theme\_bw()



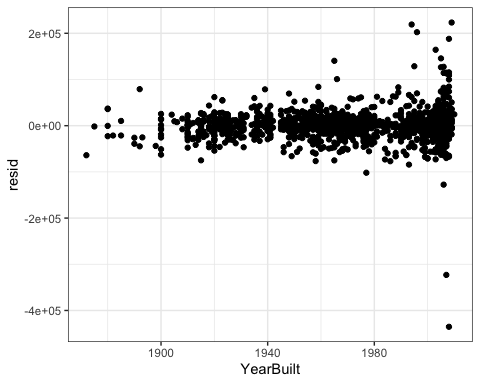
ames2 %>% mutate(resid = lm\_fit7$fit$fit$fit$residuals) %>%   
ggplot(aes(x=`1stFlrSF`,y=resid)) + geom\_point() + theme\_bw()



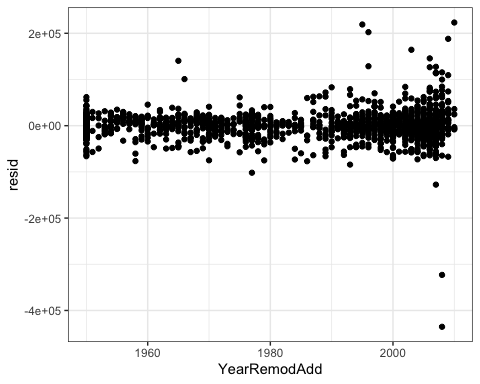
ames2 %>% mutate(resid = lm\_fit7$fit$fit$fit$residuals) %>%   
ggplot(aes(x=FullBath,y=resid)) + geom\_point() + theme\_bw()



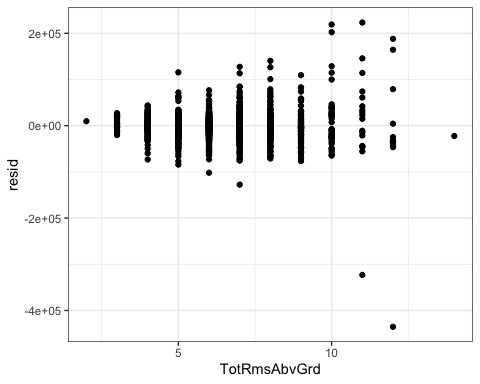
ames2 %>% mutate(resid = lm\_fit7$fit$fit$fit$residuals) %>%   
ggplot(aes(x=YearBuilt,y=resid)) + geom\_point() + theme\_bw()



ames2 %>% mutate(resid = lm\_fit7$fit$fit$fit$residuals) %>%   
ggplot(aes(x=YearRemodAdd,y=resid)) + geom\_point() + theme\_bw()



ames2 %>% mutate(resid = lm\_fit7$fit$fit$fit$residuals) %>%   
ggplot(aes(x=TotRmsAbvGrd,y=resid)) + geom\_point() + theme\_bw()



STOPPING HERE FOR PART 1