Tidyverse & ggplot

Ed Gonzalez

I am using a dataset collected by a professor on my campus which recorded physical attributes about students

```
install.packages("readr", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/tz/sh20cj15711657_9_1d4v6m00000gn/T//RtmpbIOc0J/downloaded_packages
```

```
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/tz/sh20cj15711657_9_1d4v6m00000gn/T//RtmpbIOc0J/downloaded_packages
```

```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v dplyr   1.0.10
## v tibble  3.1.8      v stringr 1.5.0
## v tidyr   1.2.1      v forcats 0.5.2
## v purrr   0.3.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
KimData <- read_csv("Downloads/Clean-KimData.csv")
```

```
## Rows: 377 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): Gender, Birth Order, dog vs cat, Handed, On/Off Campus, Phone
## dbl (19): Semester, Siblings, Shoe Size, Height, Weight, Calories per day, S...
##
## 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.
```

Utilizing some dplyr functions here!

```r
# Demonstrating the | operator when filtering to display male or freshman students as well as the usage
FreshOrMales <-KimData %>%
  filter(Semester < 2 | Gender == "M")

# Another demonstration of piping to create a tibble with selected variables
KimDataPhysical <- KimData %>%
  select(Semester, Gender, `Shoe Size`, Height, Weight, Handed)
head(KimDataPhysical)
```

```
## # A tibble: 6 x 6
##   Semester Gender `Shoe Size` Height Weight Handed
##      <dbl> <chr>        <dbl>  <dbl>  <dbl> <chr>
## 1        6 F             11       71    195 Right
## 2        4 F             10       64    187 Right
## 3        6 F              9.5     69    150 Right
## 4        7 F              9.5     64    193 Right
## 5        6 M             13       73    181 Right
## 6        6 M             10       68    167 Right
```

```r
# Using mutate() to create new variables in the data set
KimDataPhysical <- KimDataPhysical %>%
  mutate(Year = round(Semester/2))
head(KimDataPhysical)
```

```
## # A tibble: 6 x 7
##   Semester Gender `Shoe Size` Height Weight Handed  Year
##      <dbl> <chr>        <dbl>  <dbl>  <dbl> <chr>  <dbl>
## 1        6 F             11       71    195 Right      3
## 2        4 F             10       64    187 Right      2
## 3        6 F              9.5     69    150 Right      3
## 4        7 F              9.5     64    193 Right      4
## 5        6 M             13       73    181 Right      3
## 6        6 M             10       68    167 Right      3
```

```r
KimDataPhysical <- KimDataPhysical %>%
  mutate(BMI = 703 * (Weight/Height^2))
head(KimDataPhysical)
```

```
## # A tibble: 6 x 8
##   Semester Gender `Shoe Size` Height Weight Handed  Year   BMI
##      <dbl> <chr>        <dbl>  <dbl>  <dbl> <chr>  <dbl> <dbl>
## 1        6 F             11       71    195 Right      3  27.2
## 2        4 F             10       64    187 Right      2  32.1
## 3        6 F              9.5     69    150 Right      3  22.1
## 4        7 F              9.5     64    193 Right      4  33.1
## 5        6 M             13       73    181 Right      3  23.9
## 6        6 M             10       68    167 Right      3  25.4
```

```r
KimDataPhysical <- KimDataPhysical %>%
  mutate(Obese = BMI >= 30)
head(KimDataPhysical)
```

```
## # A tibble: 6 x 9
##   Semester Gender `Shoe Size` Height Weight Handed  Year   BMI Obese
##      <dbl> <chr>        <dbl>  <dbl>  <dbl> <chr>  <dbl> <dbl> <lgl>
## 1        6 F             11       71    195 Right      3  27.2 FALSE
## 2        4 F             10       64    187 Right      2  32.1 TRUE
## 3        6 F              9.5     69    150 Right      3  22.1 FALSE
## 4        7 F              9.5     64    193 Right      4  33.1 TRUE
## 5        6 M             13       73    181 Right      3  23.9 FALSE
## 6        6 M             10       68    167 Right      3  25.4 FALSE
```

Using group_by() and summarize() functions

```r
KimDataPhysical %>% group_by(Year) %>%
  summarize( Year_BMI = mean(BMI, na.rm=TRUE))
```

```
## # A tibble: 6 x 2
##    Year Year_BMI
##   <dbl>    <dbl>
## 1     0     25.1
## 2     1     23.1
## 3     2     24.0
## 4     3     23.5
## 5     4     24.6
## 6     5     28.5
```

Finding the average shoe size, but then correcting for those (like me) that have a size 16 shoe

```r
shoe_size_Kim <- KimDataPhysical %>%
  summarize("Total number" = n(), "Average Shoe Size" = mean(`Shoe Size`, na.rm=TRUE))

shoe_size_Kim
```

```
## # A tibble: 1 x 2
##   `Total number` `Average Shoe Size`
##            <int>               <dbl>
## 1            377                9.50
```

```r
regular_shoe_sizes <- KimDataPhysical %>% filter(`Shoe Size` < 16) %>%
  summarize(count = n(), mean = mean(`Shoe Size`, na.rm = TRUE))

regular_shoe_sizes
```

```
## # A tibble: 1 x 2
##   count  mean
##   <int> <dbl>
## 1   374  9.20
```

More dplyr! This time I'm poking around everyone's favorite data set, the Ames housing data set.

```
install.packages("AmesHousing", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/tz/sh20cj15711657_9_1d4v6m00000gn/T//RtmpbIOc0J/downloaded_packages
```

```
install.packages("dplyr", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/tz/sh20cj15711657_9_1d4v6m00000gn/T//RtmpbIOc0J/downloaded_packages
```

```
library(AmesHousing)
library(dplyr)

ames<-make_ames()

Remodeled <- ames$Year_Built != ames$Year_Remod_Add

set.seed(248)
ames.500 <- sample_n(ames, 500)
ames.500
```

```
## # A tibble: 500 x 81
##    MS_Sub~1 MS_Zo~2 Lot_F~3 Lot_A~4 Street Alley Lot_S~5 Land_~6 Utili~7 Lot_C~8
##    <fct>    <fct>     <dbl>   <int> <fct>  <fct> <fct>   <fct>   <fct>   <fct>
##  1 One_Sto~ Reside~      50    6000 Pave   No_A~ Regular Lvl     AllPub  Inside
##  2 One_Sto~ Reside~      66    7742 Pave   No_A~ Regular Lvl     AllPub  Inside
##  3 Two_Sto~ Reside~      60    7200 Pave   No_A~ Regular Lvl     AllPub  Corner
##  4 One_and~ Reside~      60    7200 Pave   No_A~ Regular Lvl     AllPub  Inside
##  5 Two_Sto~ Reside~      75    9073 Pave   No_A~ Slight~ Lvl     AllPub  Inside
##  6 One_Sto~ Reside~      53    4045 Pave   No_A~ Regular Lvl     AllPub  Inside
##  7 One_Sto~ Reside~      65    7832 Pave   No_A~ Regular Lvl     AllPub  Inside
##  8 One_Sto~ Reside~      40   13673 Pave   No_A~ Slight~ Lvl     AllPub  CulDSac
##  9 One_Sto~ Reside~      80    9600 Pave   No_A~ Regular Lvl     AllPub  Corner
## 10 One_Sto~ Floati~      47    4230 Pave   Paved Regular Lvl     AllPub  Corner
## # ... with 490 more rows, 71 more variables: Land_Slope <fct>,
## #   Neighborhood <fct>, Condition_1 <fct>, Condition_2 <fct>, Bldg_Type <fct>,
## #   House_Style <fct>, Overall_Qual <fct>, Overall_Cond <fct>,
## #   Year_Built <int>, Year_Remod_Add <int>, Roof_Style <fct>, Roof_Matl <fct>,
## #   Exterior_1st <fct>, Exterior_2nd <fct>, Mas_Vnr_Type <fct>,
## #   Mas_Vnr_Area <dbl>, Exter_Qual <fct>, Exter_Cond <fct>, Foundation <fct>,
## #   Bsmt_Qual <fct>, Bsmt_Cond <fct>, Bsmt_Exposure <fct>, ...
```

```r
ames.500$Remodeled <- ames.500$Year_Built != ames.500$Year_Remod_Add
ames.500$Remodeled
```

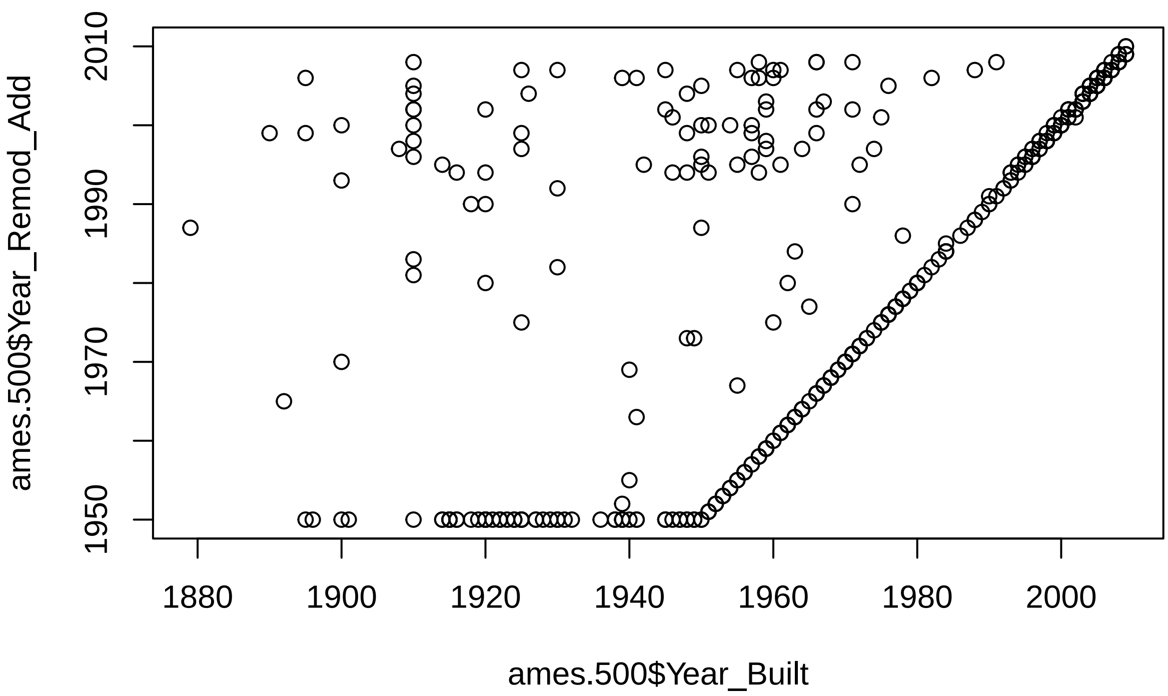
```
##   [1]  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [13]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##  [37] FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE
##  [61]  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
##  [73]  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE
##  [85] FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
##  [97] FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
## [109]  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
## [121]  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## [133]  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## [157] FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [169]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE
## [181]  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE
## [193] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
## [205]  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
## [217] FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE
## [229] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## [241]  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE
## [253] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [265] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## [277]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE
## [289] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
## [301]  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
## [313] FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE
## [325] FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [337]  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE
## [349]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE
## [361]  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
## [373]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
## [385]  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
## [397]  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE
## [409] FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [421] FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
## [433] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE
## [445] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
## [457]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
## [469] FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE
## [481] FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [493] FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE
```

```r
t.test(Sale_Price ~ Remodeled, data = ames.500)
```

```
##
##  Welch Two Sample t-test
##
## data:  Sale_Price by Remodeled
## t = 2.5661, df = 448.12, p-value = 0.01061
```

```
## alternative hypothesis: true difference in means between group FALSE and group TRUE is not equal to 0
## 95 percent confidence interval:
##   4324.58 32613.90
## sample estimates:
## mean in group FALSE  mean in group TRUE
##            191067.9            172598.7
```

```
plot(x = ames.500$Year_Built, y = ames.500$Year_Remod_Add)
```



Running simulated data sets

```
sample_1 <- rnorm(50,15,8)
sample_2 <- rnorm(50,17,8)

t.test(sample_1, sample_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  sample_1 and sample_2
## t = -2.3408, df = 83.592, p-value = 0.02163
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -6.4093123 -0.5211034
## sample estimates:
## mean of x mean of y
##  13.22125  16.68646
```

Increasing the sample size to see how it affects the p-value and our confidence in the results

```
sample_1 <- rnorm(100,15,8)
sample_2 <- rnorm(100,17,8)

t.test(sample_1, sample_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  sample_1 and sample_2
## t = -2.1183, df = 197.99, p-value = 0.0354
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.1606230 -0.1488166
## sample estimates:
## mean of x mean of y
##  14.52700  16.68172
```

```
sample_1 <- rnorm(200,15,8)
sample_2 <- rnorm(200,17,8)

t.test(sample_1, sample_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  sample_1 and sample_2
## t = 0.014557, df = 398, p-value = 0.9884
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.603460  1.627383
## sample estimates:
## mean of x mean of y
##  16.17849  16.16653
```

Finding the linear regression between the sale price and square footage and the year it was built.

```
fit.original <- lm(Sale_Price ~ Year_Built + First_Flr_SF, data=ames)

fit.original
```

```
##
## Call:
## lm(formula = Sale_Price ~ Year_Built + First_Flr_SF, data = ames)
##
## Coefficients:
##  (Intercept)    Year_Built  First_Flr_SF
##   -2042141.4        1068.1         101.1
```

```
summary(fit.original)$adj.r.squared
```

```
## [1] 0.5339375
```

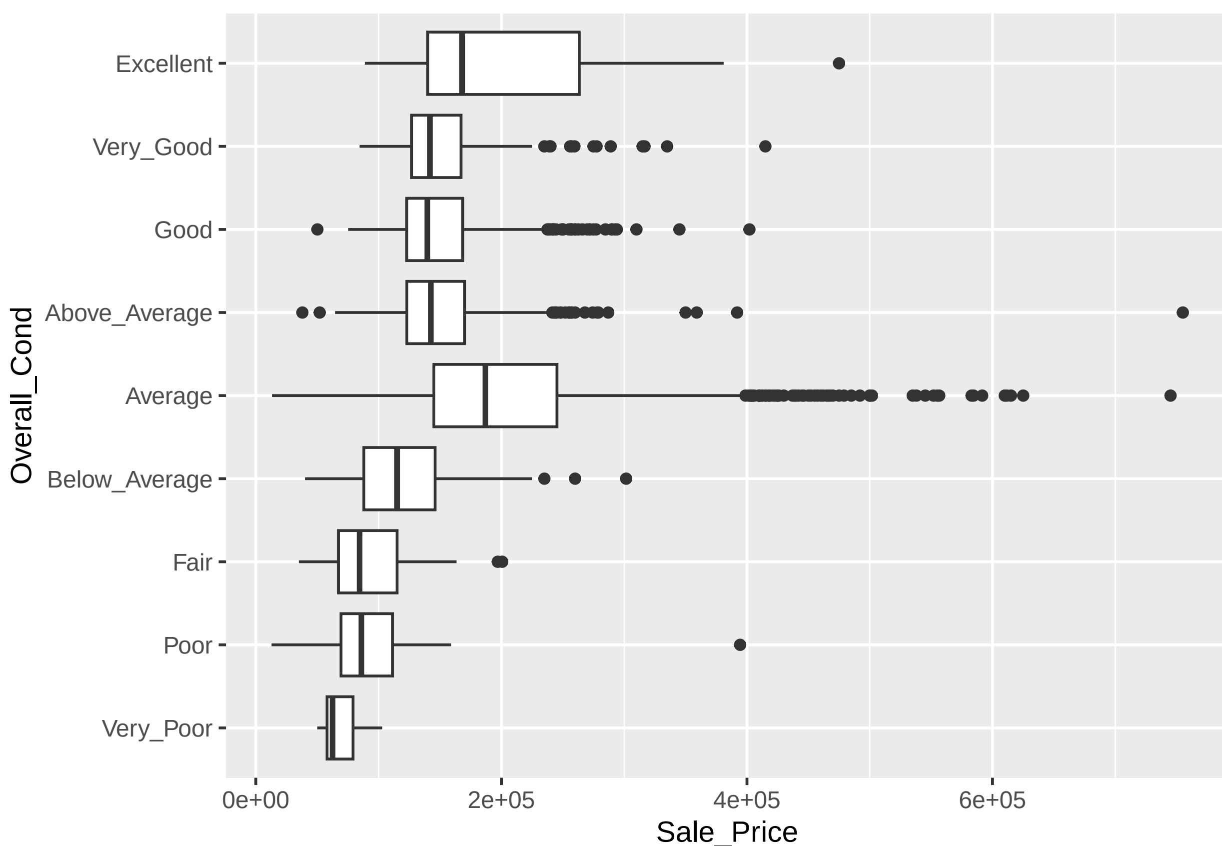
```
fit.updated <- lm(Sale_Price ~ Lot_Area + Overall_Qual, data=ames)

summary(fit.updated)$adj.r.squared
```

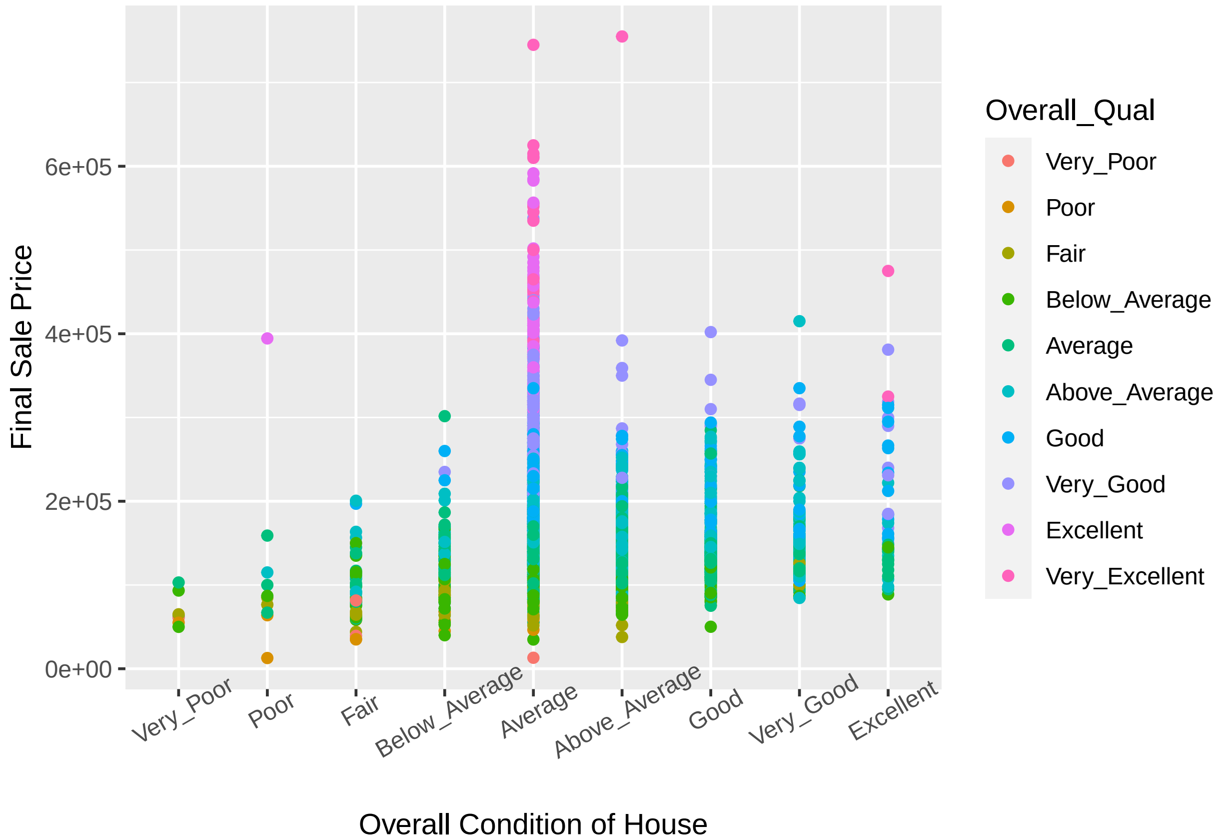
```
## [1] 0.7311099
```

How about some ggplot?

```
ggplot(ames, aes(x = Sale_Price, y = Overall_Cond)) + geom_boxplot()
```



```
ggplot(ames, aes(x = Overall_Cond, y = Sale_Price, color = Overall_Qual)) + geom_jitter( width = 0) + tl
```



By adding a legend and color to the data points, we can see how the data is distributed in relation to its quality and sale price. This is where I first started to really learn ggplot. Later on I start using the piping method to better organize my code and improve replicability.