05222022_EDA

Load packages

```
library(tidyverse) # data manipulation
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)   # plotting
```

Load data

```
data <- read.csv("train.csv")
```

View data

```
data %>% glimpse
```

```
## Rows: 1,460
## Columns: 81
## $ Id            <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1~
## $ MSSubClass    <int> 60, 20, 60, 70, 60, 50, 20, 60, 50, 190, 20, 60, 20, 20,~
## $ MSZoning      <chr> "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RM", "R~
## $ LotFrontage   <int> 65, 80, 68, 60, 84, 85, 75, NA, 51, 50, 70, 85, NA, 91, ~
## $ LotArea       <int> 8450, 9600, 11250, 9550, 14260, 14115, 10084, 10382, 612~
## $ Street        <chr> "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", ~
## $ Alley         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ LotShape      <chr> "Reg", "Reg", "IR1", "IR1", "IR1", "IR1", "Reg", "IR1", ~
## $ LandContour   <chr> "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", ~
```

```
## $ Utilities     <chr> "AllPub", "AllPub", "AllPub", "AllPub", "AllPub", "AllPu~
## $ LotConfig     <chr> "Inside", "FR2", "Inside", "Corner", "FR2", "Inside", "I~
## $ LandSlope     <chr> "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", ~
## $ Neighborhood  <chr> "CollgCr", "Veenker", "CollgCr", "Crawfor", "NoRidge", "~
## $ Condition1    <chr> "Norm", "Feedr", "Norm", "Norm", "Norm", "Norm", "Norm",~
## $ Condition2    <chr> "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", ~
## $ BldgType      <chr> "1Fam", "1Fam", "1Fam", "1Fam", "1Fam", "1Fam", "1Fam", ~
## $ HouseStyle    <chr> "2Story", "1Story", "2Story", "2Story", "2Story", "1.5Fi~
## $ OverallQual   <int> 7, 6, 7, 7, 8, 5, 8, 7, 7, 5, 5, 9, 5, 7, 6, 7, 6, 4, 5,~
## $ OverallCond   <int> 5, 8, 5, 5, 5, 5, 5, 6, 5, 6, 5, 5, 6, 5, 5, 8, 7, 5, 5,~
## $ YearBuilt     <int> 2003, 1976, 2001, 1915, 2000, 1993, 2004, 1973, 1931, 19~
## $ YearRemodAdd  <int> 2003, 1976, 2002, 1970, 2000, 1995, 2005, 1973, 1950, 19~
## $ RoofStyle     <chr> "Gable", "Gable", "Gable", "Gable", "Gable", "Gable", "G~
## $ RoofMatl      <chr> "CompShg", "CompShg", "CompShg", "CompShg", "CompShg", "~
## $ Exterior1st   <chr> "VinylSd", "MetalSd", "VinylSd", "Wd Sdng", "VinylSd", "~
## $ Exterior2nd   <chr> "VinylSd", "MetalSd", "VinylSd", "Wd Shng", "VinylSd", "~
## $ MasVnrType    <chr> "BrkFace", "None", "BrkFace", "None", "BrkFace", "None",~
## $ MasVnrArea    <int> 196, 0, 162, 0, 350, 0, 186, 240, 0, 0, 0, 286, 0, 306, ~
## $ ExterQual     <chr> "Gd", "TA", "Gd", "TA", "Gd", "TA", "Gd", "TA", "TA", "T~
## $ ExterCond     <chr> "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "T~
## $ Foundation    <chr> "PConc", "CBlock", "PConc", "BrkTil", "PConc", "Wood", "~
## $ BsmtQual      <chr> "Gd", "Gd", "Gd", "TA", "Gd", "Gd", "Ex", "Gd", "TA", "T~
## $ BsmtCond      <chr> "TA", "TA", "TA", "Gd", "TA", "TA", "TA", "TA", "TA", "T~
## $ BsmtExposure  <chr> "No", "Gd", "Mn", "No", "Av", "No", "Av", "Mn", "No", "N~
## $ BsmtFinType1  <chr> "GLQ", "ALQ", "GLQ", "ALQ", "GLQ", "GLQ", "GLQ", "ALQ", ~
## $ BsmtFinSF1    <int> 706, 978, 486, 216, 655, 732, 1369, 859, 0, 851, 906, 99~
## $ BsmtFinType2  <chr> "Unf", "Unf", "Unf", "Unf", "Unf", "Unf", "Unf", "BLQ", ~
## $ BsmtFinSF2    <int> 0, 0, 0, 0, 0, 0, 0, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ BsmtUnfSF     <int> 150, 284, 434, 540, 490, 64, 317, 216, 952, 140, 134, 17~
## $ TotalBsmtSF   <int> 856, 1262, 920, 756, 1145, 796, 1686, 1107, 952, 991, 10~
## $ Heating       <chr> "GasA", "GasA", "GasA", "GasA", "GasA", "GasA", "GasA", ~
## $ HeatingQC     <chr> "Ex", "Ex", "Ex", "Gd", "Ex", "Ex", "Ex", "Ex", "Gd", "E~
## $ CentralAir    <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "~
## $ Electrical    <chr> "SBrkr", "SBrkr", "SBrkr", "SBrkr", "SBrkr", "SBrkr", "S~
## $ X1stFlrSF     <int> 856, 1262, 920, 961, 1145, 796, 1694, 1107, 1022, 1077, ~
## $ X2ndFlrSF     <int> 854, 0, 866, 756, 1053, 566, 0, 983, 752, 0, 0, 1142, 0,~
## $ LowQualFinSF  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ GrLivArea     <int> 1710, 1262, 1786, 1717, 2198, 1362, 1694, 2090, 1774, 10~
## $ BsmtFullBath  <int> 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1,~
## $ BsmtHalfBath  <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ FullBath      <int> 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 1, 3, 1, 2, 1, 1, 1, 2, 1,~
## $ HalfBath      <int> 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,~
## $ BedroomAbvGr  <int> 3, 3, 3, 3, 4, 1, 3, 3, 2, 2, 3, 4, 2, 3, 2, 2, 2, 2, 3,~
## $ KitchenAbvGr  <int> 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1,~
## $ KitchenQual   <chr> "Gd", "TA", "Gd", "Gd", "Gd", "TA", "Gd", "TA", "TA", "T~
## $ TotRmsAbvGrd  <int> 8, 6, 6, 7, 9, 5, 7, 7, 8, 5, 5, 11, 4, 7, 5, 5, 5, 6, 6~
## $ Functional    <chr> "Typ", "Typ", "Typ", "Typ", "Typ", "Typ", "Typ", "Typ", ~
## $ Fireplaces    <int> 0, 1, 1, 1, 1, 0, 1, 2, 2, 2, 0, 2, 0, 1, 1, 0, 1, 0, 0,~
## $ FireplaceQu   <chr> NA, "TA", "TA", "Gd", "TA", NA, "Gd", "TA", "TA", "TA", ~
## $ GarageType    <chr> "Attchd", "Attchd", "Attchd", "Detchd", "Attchd", "Attch~
## $ GarageYrBlt   <int> 2003, 1976, 2001, 1998, 2000, 1993, 2004, 1973, 1931, 19~
## $ GarageFinish  <chr> "RFn", "RFn", "RFn", "Unf", "RFn", "Unf", "RFn", "RFn", ~
## $ GarageCars    <int> 2, 2, 2, 3, 3, 2, 2, 2, 2, 1, 1, 3, 1, 3, 1, 2, 2, 2, 2,~
## $ GarageArea    <int> 548, 460, 608, 642, 836, 480, 636, 484, 468, 205, 384, 7~
```

```
## $ GarageQual    <chr> "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "Fa", "G~
## $ GarageCond    <chr> "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "T~
## $ PavedDrive    <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "~
## $ WoodDeckSF    <int> 0, 298, 0, 0, 192, 40, 255, 235, 90, 0, 0, 147, 140, 160~
## $ OpenPorchSF   <int> 61, 0, 42, 35, 84, 30, 57, 204, 0, 4, 0, 21, 0, 33, 213,~
## $ EnclosedPorch <int> 0, 0, 0, 272, 0, 0, 0, 228, 205, 0, 0, 0, 0, 0, 176, 0, ~
## $ X3SsnPorch    <int> 0, 0, 0, 0, 0, 320, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ ScreenPorch   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 176, 0, 0, 0, 0, 0, ~
## $ PoolArea      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ PoolQC        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ Fence         <chr> NA, NA, NA, NA, NA, "MnPrv", NA, NA, NA, NA, NA, NA, NA,~
## $ MiscFeature   <chr> NA, NA, NA, NA, NA, "Shed", NA, "Shed", NA, NA, NA, NA, ~
## $ MiscVal       <int> 0, 0, 0, 0, 0, 700, 0, 350, 0, 0, 0, 0, 0, 0, 0, 700,~
## $ MoSold        <int> 2, 5, 9, 2, 12, 10, 8, 11, 4, 1, 2, 7, 9, 8, 5, 7, 3, 10~
## $ YrSold        <int> 2008, 2007, 2008, 2006, 2008, 2009, 2007, 2009, 2008, 20~
## $ SaleType      <chr> "WD", "WD", "WD", "WD", "WD", "WD", "WD", "WD", "WD", "W~
## $ SaleCondition <chr> "Normal", "Normal", "Normal", "Abnorml", "Normal", "Norm~
## $ SalePrice     <int> 208500, 181500, 223500, 140000, 250000, 143000, 307000, ~
```

Check data

```
data %>% dim
```

```
## [1] 1460   81
```

Data cleaning

```
# Count missing data
data_missing <- data %>%
  select(everything()) %>%
  summarise_all(funs(sum(is.na(.))))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
data_missing
```

```
##   Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape LandContour
## 1  0          0        0         259       0      0  1369        0           0
##   Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgType
## 1         0         0         0            0          0          0        0
##   HouseStyle OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle RoofMatl
## 1          0           0           0         0            0         0        0
##   Exterior1st Exterior2nd MasVnrType MasVnrArea ExterQual ExterCond Foundation
## 1           0           0          8          8         0         0          0
##   BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2
## 1       37       37           38           37          0           38
##   BsmtFinSF2 BsmtUnfSF TotalBsmtSF Heating HeatingQC CentralAir Electrical
## 1          0         0           0       0         0          0          1
##   X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath
## 1         0         0            0         0            0            0        0
##   HalfBath BedroomAbvGr KitchenAbvGr KitchenQual TotRmsAbvGrd Functional
## 1        0            0           0           0            0          0
##   Fireplaces FireplaceQu GarageType GarageYrBlt GarageFinish GarageCars
## 1          0         690         81          81           81          0
##   GarageArea GarageQual GarageCond PavedDrive WoodDeckSF OpenPorchSF
## 1          0         81         81          0          0           0
##   EnclosedPorch X3SsnPorch ScreenPorch PoolArea PoolQC Fence MiscFeature
## 1             0          0           0        0   1453  1179        1406
##   MiscVal MoSold YrSold SaleType SaleCondition SalePrice
## 1       0      0      0        0             0         0
```

```
# colSums(is.na(data)) # base R
```

```
# # Remove Alley
# data_clean <- data %>% select(-c(Alley)) %>% head
# data_clean %>% dim # Odd
```

We can consider replace NAs with mean values.

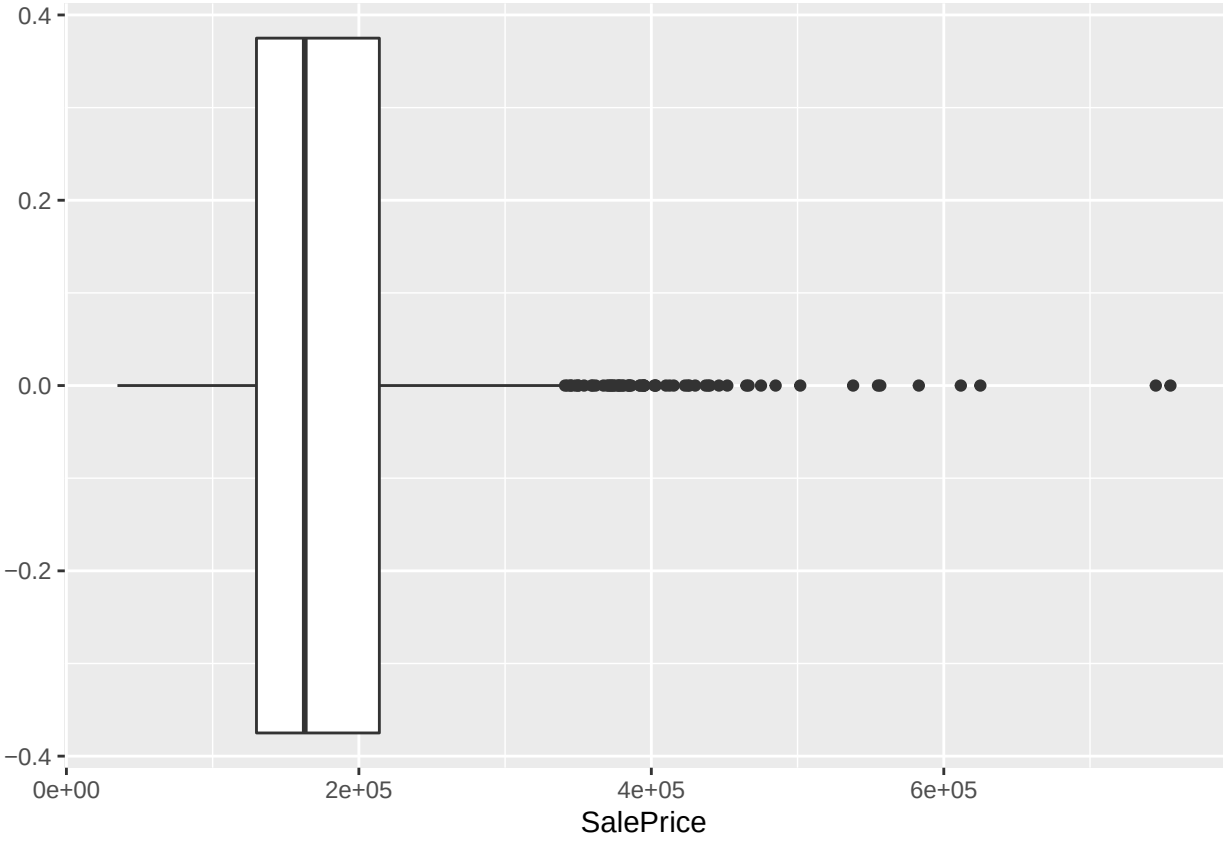
EDA

Check SalePrice (response variable)

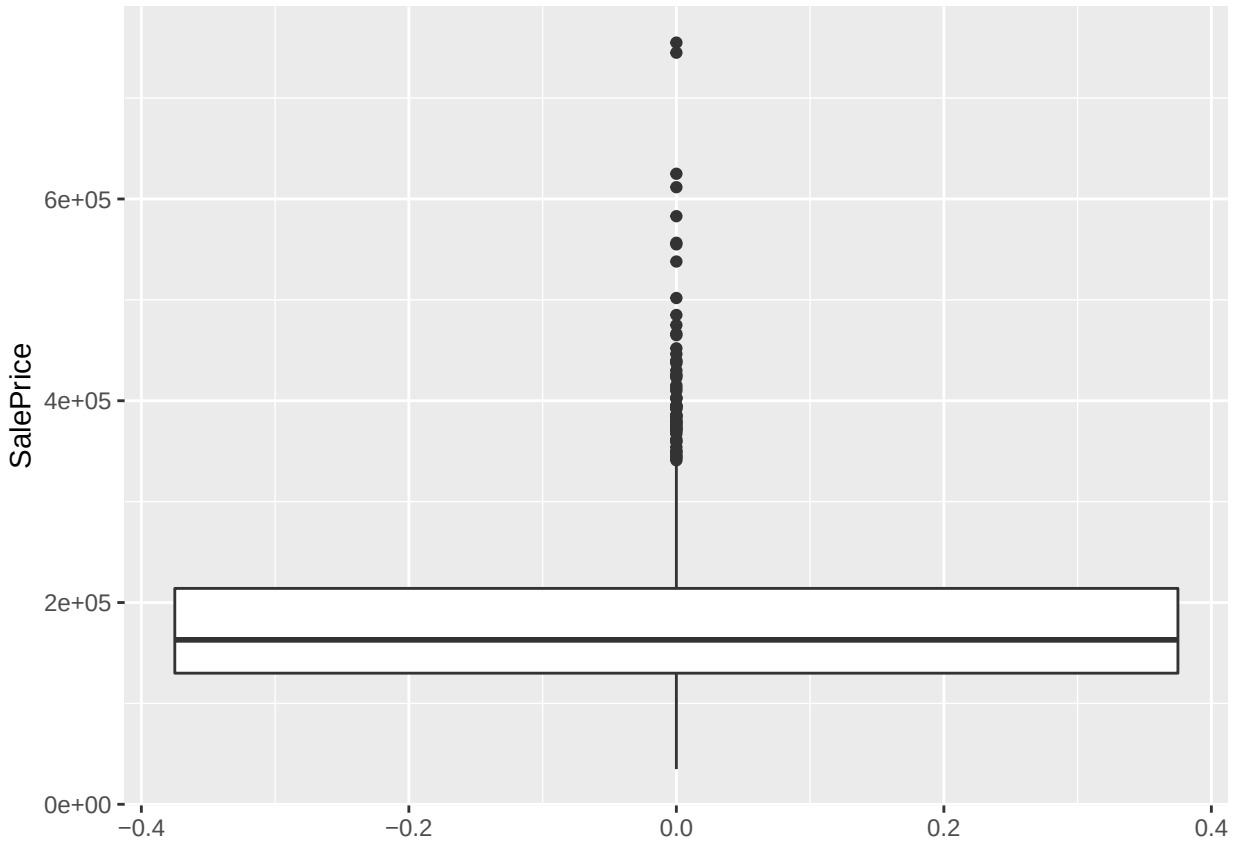
```
# Compute summary statistics
data %>% select(SalePrice) %>% summary()
```

```
##    SalePrice
##  Min.   : 34900
##  1st Qu.:129975
##  Median :163000
##  Mean   :180921
##  3rd Qu.:214000
##  Max.   :755000
```

```
# Check 25, 75 quantile, potential outliers and normality
data %>% ggplot(aes(x = SalePrice)) + geom_boxplot()
```




```
# Flip the plot
data %>% ggplot(aes(x = SalePrice)) + geom_boxplot() + coord_flip()
```



There may be outliers in the data set.

Remove the outliers

```
# Identify quantile values
quantile(data$SalePrice, prob = c(0.25, 0.75))
```

```
##    25%    75%
## 129975 214000
```

```
# Identify observations
row <- which(data$SalePrice < 129975 | data$SalePrice > 214000)
length(row)
```

```
## [1] 727
```

We may have remove too many observations but let's come back to fix it later.

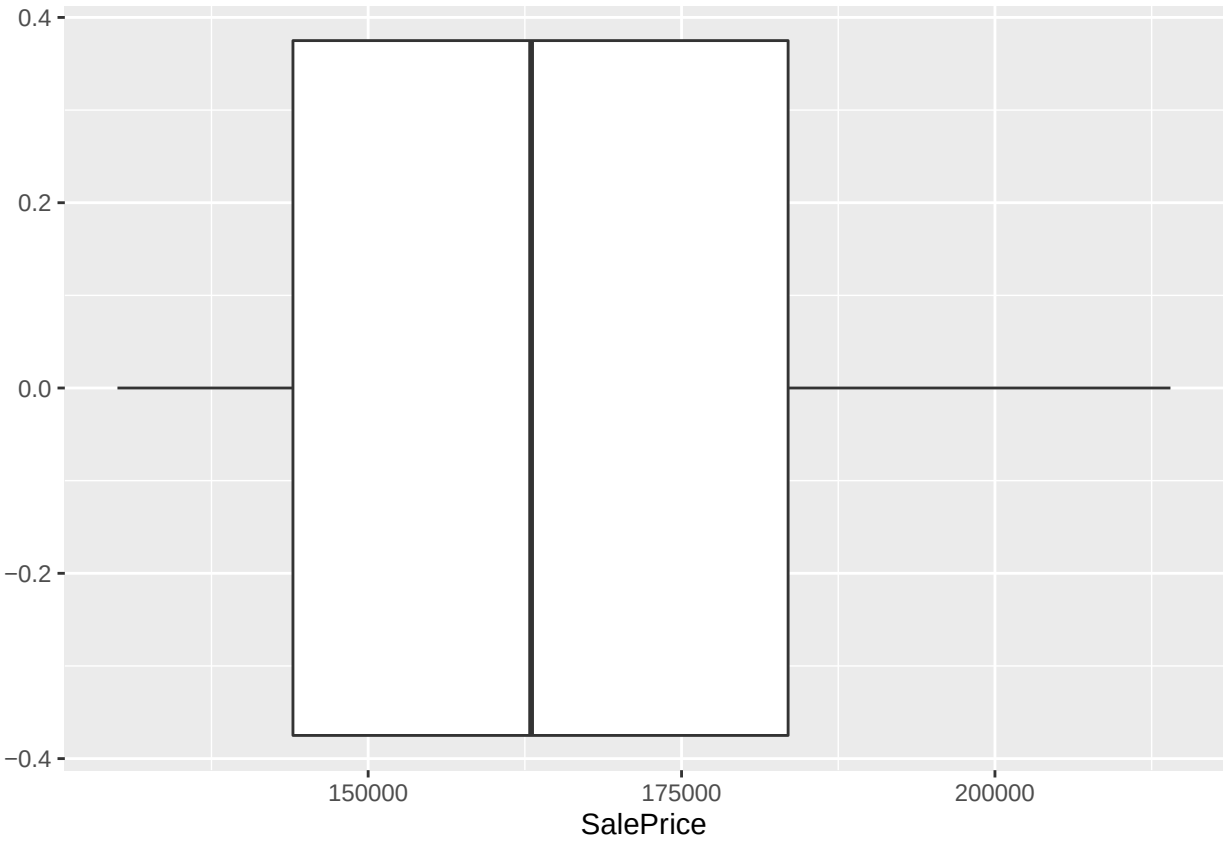
```
length(data$SalePrice)
```

```
## [1] 1460
```

```
# Subset data
data_sub <- data[-c(row), ]
data_sub %>% dim
```

```
## [1] 733  81
```

```
# Replot to see if it's better
data_sub %>% ggplot(aes(x = SalePrice)) + geom_boxplot()
```

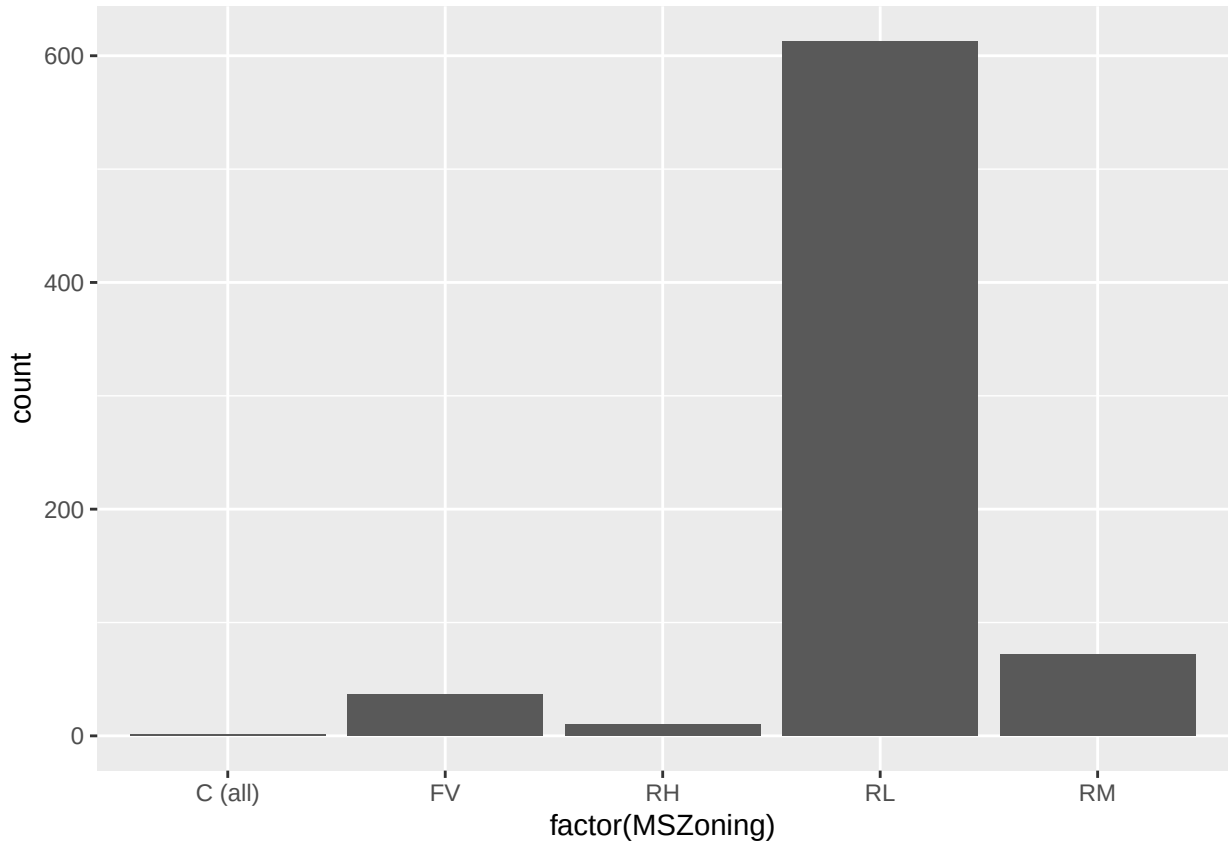


Look better but may not be correct.

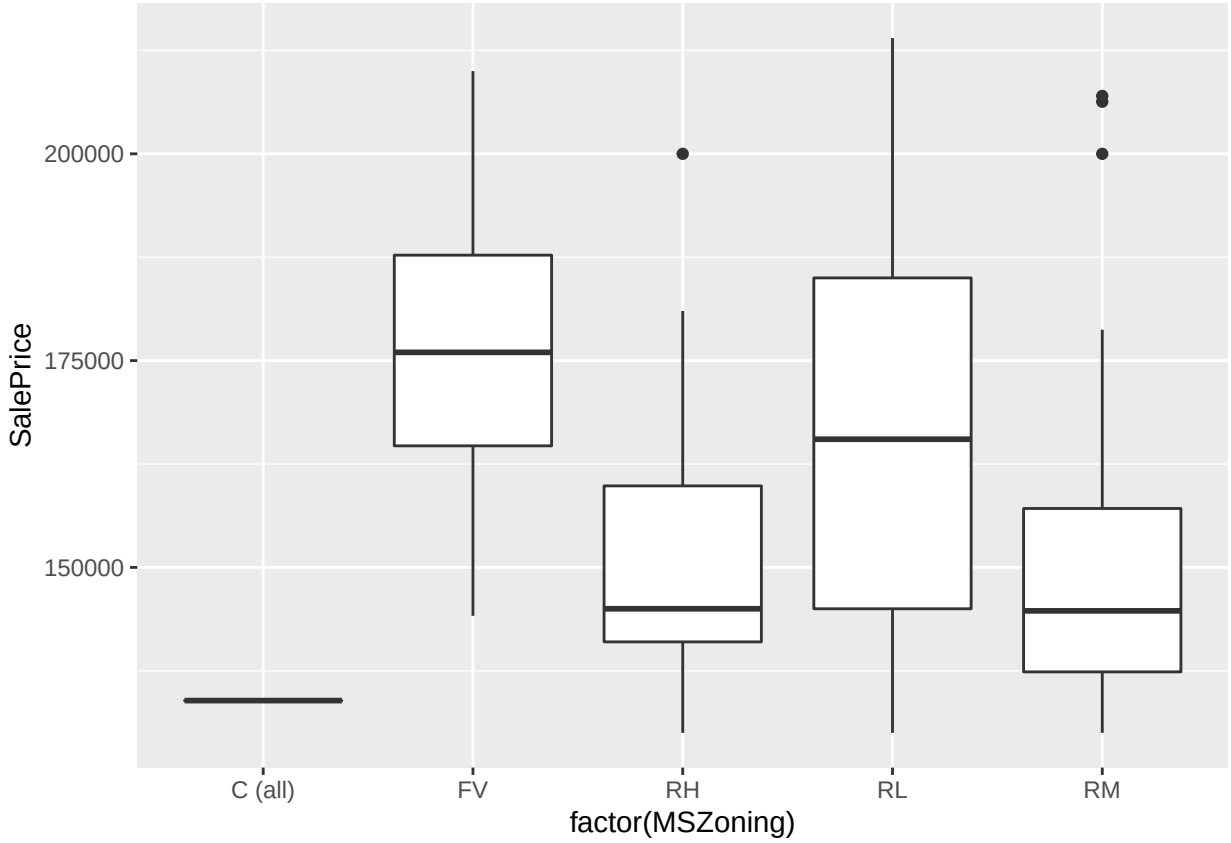
Check other predictor variables

Use bar plot for categorical variables.

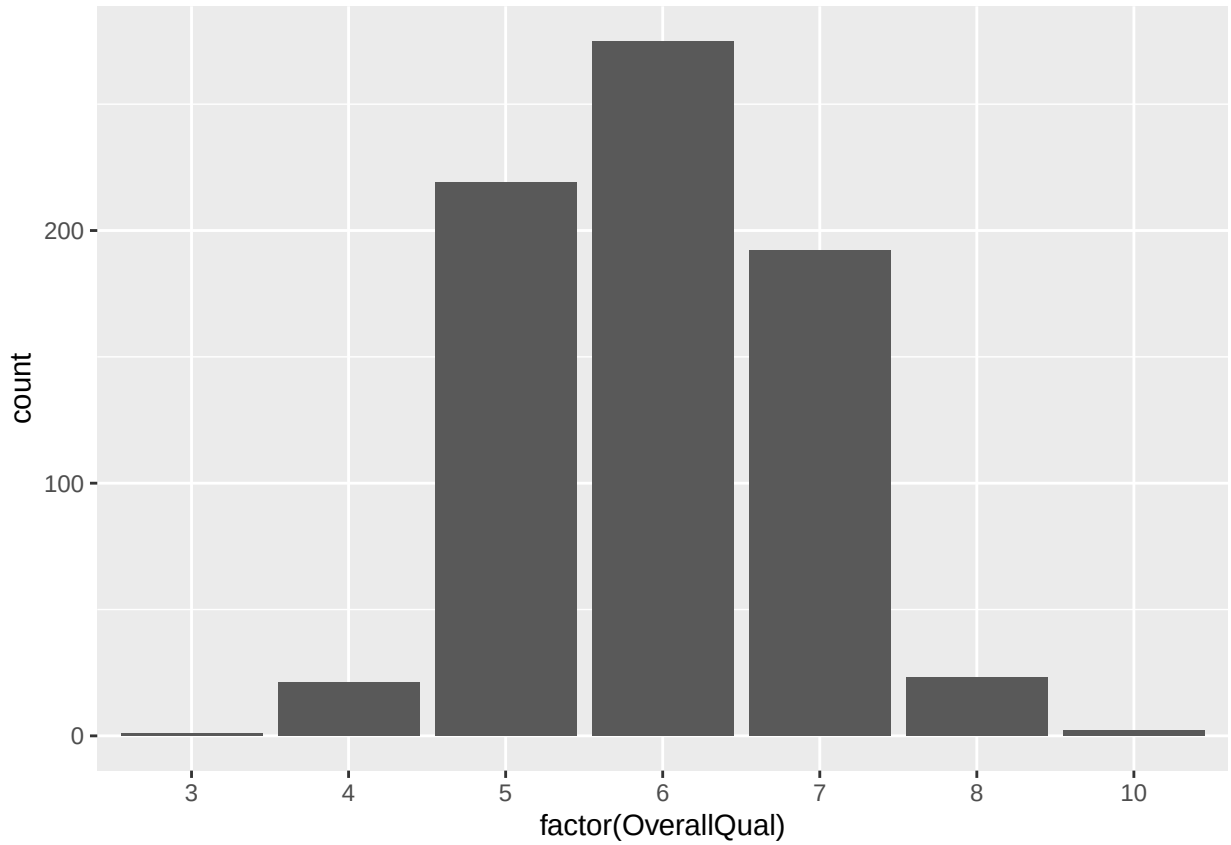
```
data_sub %>% ggplot(aes(x = factor(MSZoning))) + geom_bar()
```



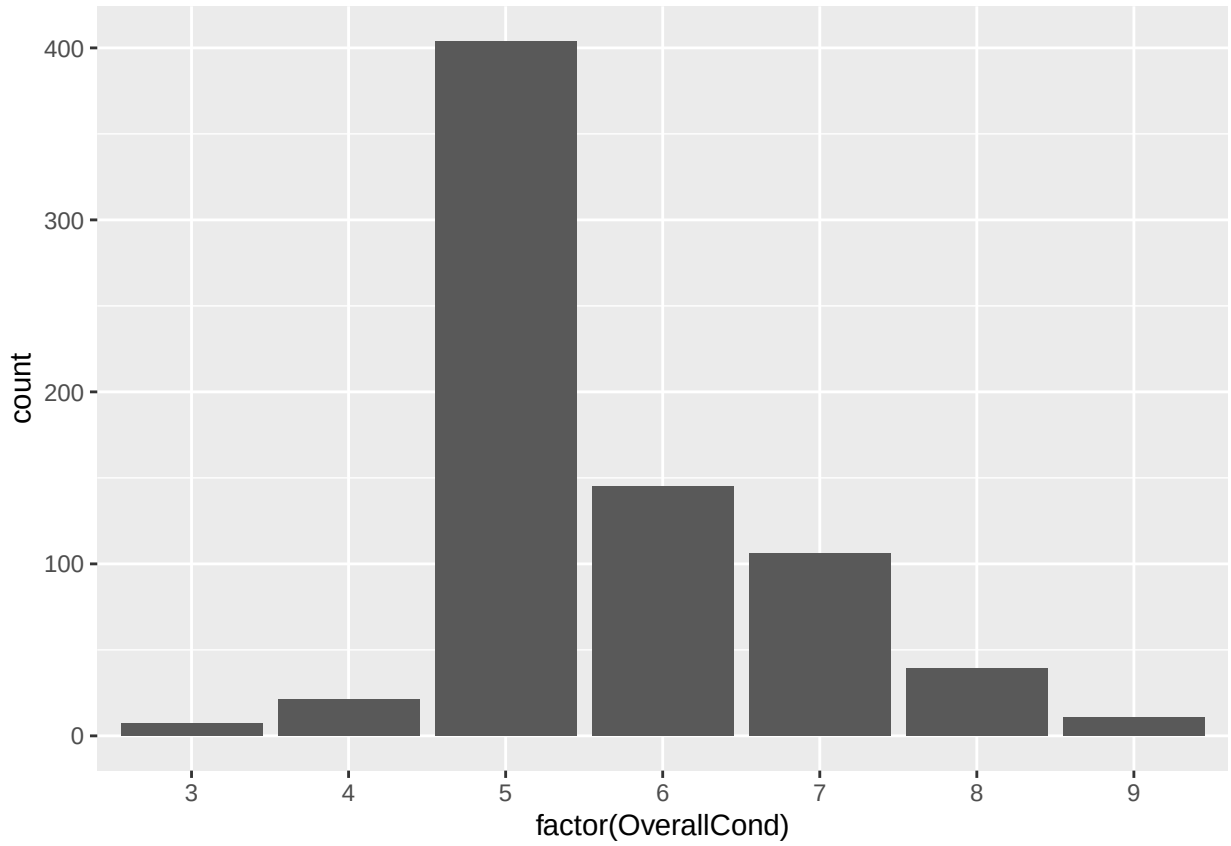

```
# Add SalePrice
data_sub %>% ggplot(aes(x = factor(MSZoning), y = SalePrice)) +
  geom_boxplot()
```




```
data_sub %>% ggplot(aes(x = factor(OverallQual))) + geom_bar()
```

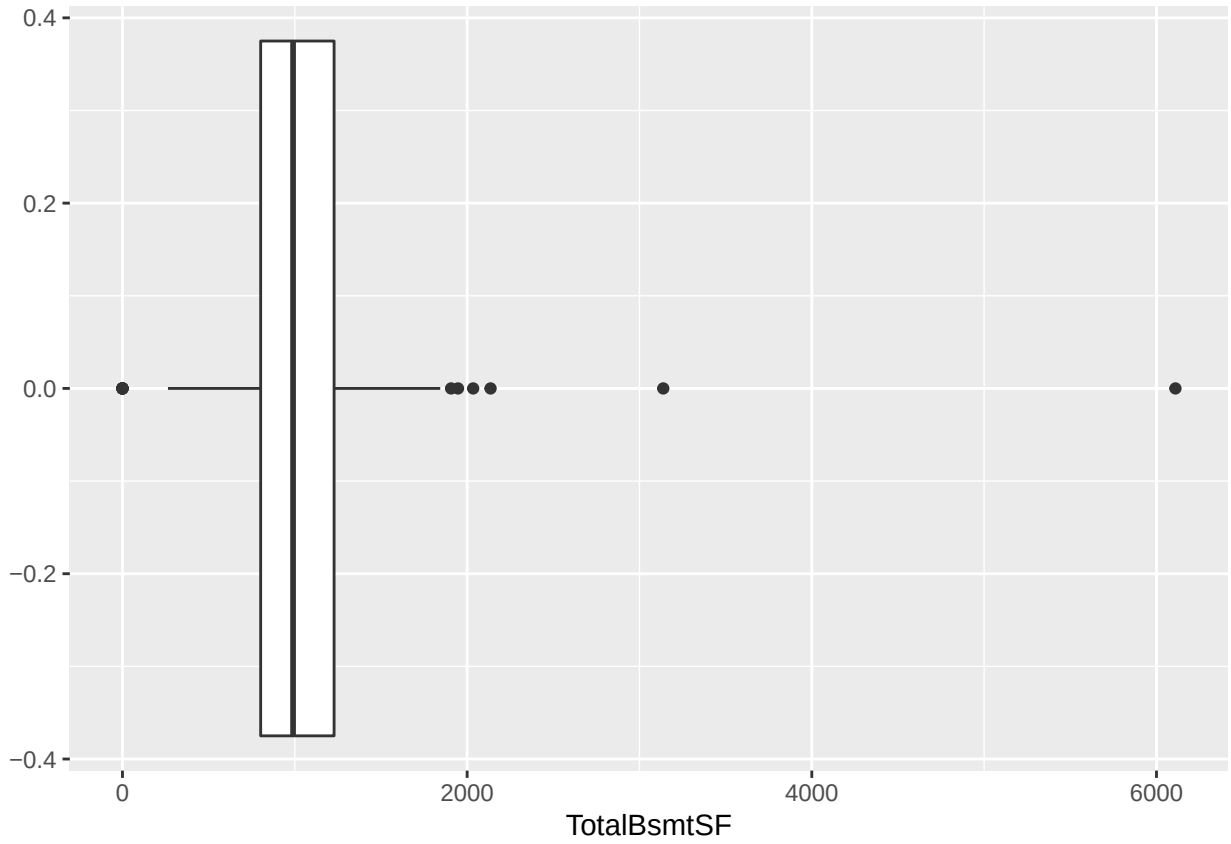



```
data_sub %>% ggplot(aes(x = factor(OverallCond))) + geom_bar()
```



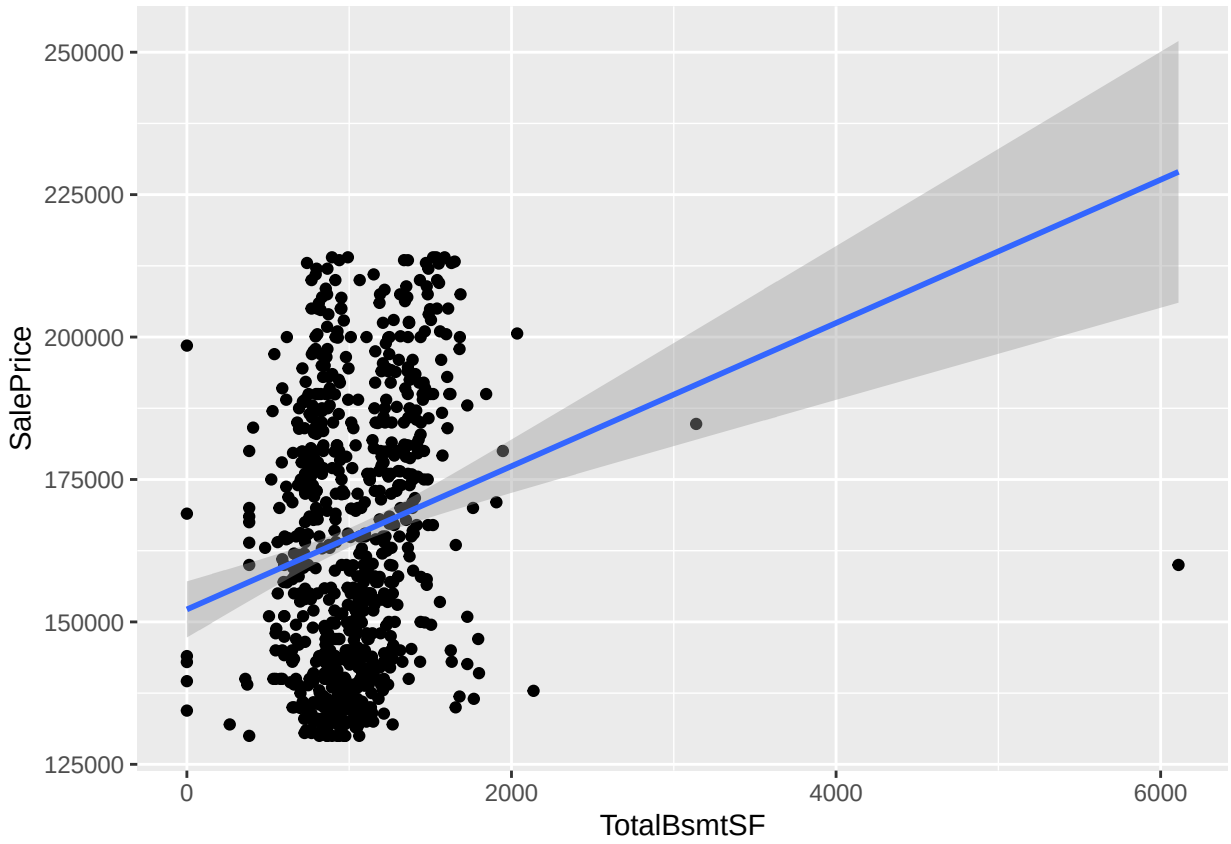
Probably better to write a function.

```
data_sub %>% ggplot(aes(x = TotalBsmtSF)) + geom_boxplot()
```




```
# Add SalePrice
data_sub %>% ggplot(aes(x = TotalBsmtSF, y = SalePrice)) + geom_point() +
  geom_smooth(method='lm') # Overlay a regression line
```

```
## `geom_smooth()` using formula 'y ~ x'
```



This implies that the model is a poor fit

$$\mu(SalePrice|TotalBsmtSF) = \beta_0 + \beta_1 TotalBsmtSF$$

Separate categorical variable from continuous variable

```
#rapply(data_sub, class = "numeric", f = levels, how = "list") # Not quite
```

```
#rapply(data_sub, class = "factor", f = levels, how = "list")
```

The list goes on...

```
data_sub %>% glimpse
```

```
## Rows: 733
## Columns: 81
## $ Id            <int> 1, 2, 4, 6, 8, 13, 15, 16, 17, 19, 20, 22, 25, 27, 29, 3~
## $ MSSubClass    <int> 60, 20, 70, 50, 60, 20, 20, 45, 20, 20, 20, 45, 20, 20, ~
## $ MSZoning      <chr> "RL", "RL", "RL", "RL", "RL", "RL", "RL", "RM", "RL", "R~
## $ LotFrontage   <int> 65, 80, 60, 85, NA, NA, NA, 51, NA, 66, 70, 57, NA, 60, ~
## $ LotArea       <int> 8450, 9600, 9550, 14115, 10382, 12968, 10920, 6120, 1124~
## $ Street        <chr> "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", ~
## $ Alley         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "Grvl", NA, ~
## $ LotShape      <chr> "Reg", "Reg", "IR1", "IR1", "IR1", "IR2", "IR1", "Reg", ~
```

```
## $ LandContour   <chr> "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", ~
## $ Utilities     <chr> "AllPub", "AllPub", "AllPub", "AllPub", "AllPub", "AllPu~
## $ LotConfig     <chr> "Inside", "FR2", "Corner", "Inside", "Corner", "Inside",~
## $ LandSlope     <chr> "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", ~
## $ Neighborhood  <chr> "CollgCr", "Veenker", "Crawfor", "Mitchel", "NWAmes", "S~
## $ Condition1    <chr> "Norm", "Feedr", "Norm", "Norm", "PosN", "Norm", "Norm",~
## $ Condition2    <chr> "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", ~
## $ BldgType      <chr> "1Fam", "1Fam", "1Fam", "1Fam", "1Fam", "1Fam", "1Fam", ~
## $ HouseStyle    <chr> "2Story", "1Story", "2Story", "1.5Fin", "2Story", "1Stor~
## $ OverallQual   <int> 7, 6, 7, 5, 7, 5, 6, 7, 6, 5, 5, 7, 5, 5, 5, 5, 8, 5, 5,~
## $ OverallCond   <int> 5, 8, 5, 5, 6, 6, 5, 8, 7, 5, 6, 7, 8, 7, 6, 6, 5, 5, 5,~
## $ YearBuilt     <int> 2003, 1976, 1915, 1993, 1973, 1962, 1960, 1929, 1970, 20~
## $ YearRemodAdd  <int> 2003, 1976, 1970, 1995, 1973, 1962, 1960, 2001, 1970, 20~
## $ RoofStyle     <chr> "Gable", "Gable", "Gable", "Gable", "Gable", "Hip", "Hip~
## $ RoofMatl      <chr> "CompShg", "CompShg", "CompShg", "CompShg", "CompShg", "~
## $ Exterior1st   <chr> "VinylSd", "MetalSd", "Wd Sdng", "VinylSd", "HdBoard", "~
## $ Exterior2nd   <chr> "VinylSd", "MetalSd", "Wd Shng", "VinylSd", "HdBoard", "~
## $ MasVnrType    <chr> "BrkFace", "None", "None", "None", "Stone", "None", "Brk~
## $ MasVnrArea    <int> 196, 0, 0, 0, 240, 0, 212, 0, 180, 0, 0, 0, 0, 0, 0, 0, ~
## $ ExterQual     <chr> "Gd", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "T~
## $ ExterCond     <chr> "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "T~
## $ Foundation    <chr> "PConc", "CBlock", "BrkTil", "Wood", "CBlock", "CBlock",~
## $ BsmtQual      <chr> "Gd", "Gd", "TA", "Gd", "Gd", "TA", "TA", "TA", "TA", "T~
## $ BsmtCond      <chr> "TA", "TA", "Gd", "TA", "TA", "TA", "TA", "TA", "TA", "T~
## $ BsmtExposure  <chr> "No", "Gd", "No", "No", "Mn", "No", "No", "No", "No", "N~
## $ BsmtFinType1  <chr> "GLQ", "ALQ", "ALQ", "GLQ", "ALQ", "ALQ", "BLQ", "Unf", ~
## $ BsmtFinSF1    <int> 706, 978, 216, 732, 859, 737, 733, 0, 578, 646, 504, 0, ~
## $ BsmtFinType2  <chr> "Unf", "Unf", "Unf", "Unf", "BLQ", "Unf", "Unf", "Unf", ~
## $ BsmtFinSF2    <int> 0, 0, 0, 0, 32, 0, 0, 0, 0, 0, 0, 0, 668, 486, 0, 0, 0, ~
## $ BsmtUnfSF     <int> 150, 284, 540, 64, 216, 175, 520, 832, 426, 468, 525, 63~
## $ TotalBsmtSF   <int> 856, 1262, 756, 796, 1107, 912, 1253, 832, 1004, 1114, 1~
## $ Heating       <chr> "GasA", "GasA", "GasA", "GasA", "GasA", "GasA", "GasA", ~
## $ HeatingQC     <chr> "Ex", "Ex", "Gd", "Ex", "Ex", "TA", "TA", "Ex", "Ex", "E~
## $ CentralAir    <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "~
## $ Electrical    <chr> "SBrkr", "SBrkr", "SBrkr", "SBrkr", "SBrkr", "SBrkr", "S~
## $ X1stFlrSF     <int> 856, 1262, 961, 796, 1107, 912, 1253, 854, 1004, 1114, 1~
## $ X2ndFlrSF     <int> 854, 0, 756, 566, 983, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ LowQualFinSF  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ GrLivArea     <int> 1710, 1262, 1717, 1362, 2090, 912, 1253, 854, 1004, 1114~
## $ BsmtFullBath  <int> 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,~
## $ BsmtHalfBath  <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,~
## $ FullBath      <int> 2, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1,~
## $ HalfBath      <int> 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1,~
## $ BedroomAbvGr  <int> 3, 3, 3, 1, 3, 2, 2, 2, 2, 3, 3, 3, 3, 3, 2, 3, 3, 4, 3,~
## $ KitchenAbvGr  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ KitchenQual   <chr> "Gd", "TA", "Gd", "TA", "TA", "TA", "TA", "TA", "TA", "G~
## $ TotRmsAbvGrd  <int> 8, 6, 7, 5, 7, 4, 5, 5, 5, 6, 6, 6, 6, 5, 6, 6, 7, 6, 6,~
## $ Functional    <chr> "Typ", "Typ", "Typ", "Typ", "Typ", "Typ", "Typ", "Typ", ~
## $ Fireplaces    <int> 0, 1, 1, 0, 2, 0, 1, 0, 1, 0, 0, 1, 1, 0, 2, 0, 0, 1, 0,~
## $ FireplaceQu   <chr> NA, "TA", "Gd", NA, "TA", NA, "Fa", NA, "TA", NA, NA, "G~
## $ GarageType    <chr> "Attchd", "Attchd", "Detchd", "Attchd", "Attchd", "Detch~
## $ GarageYrBlt   <int> 2003, 1976, 1998, 1993, 1973, 1962, 1960, 1991, 1970, 20~
## $ GarageFinish  <chr> "RFn", "RFn", "Unf", "Unf", "RFn", "Unf", "RFn", "Unf", ~
## $ GarageCars    <int> 2, 2, 3, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1, 2, 1, 1, 2, 2, 2,~
```

```
## $ GarageArea    <int> 548, 460, 642, 480, 484, 352, 352, 576, 480, 576, 294, 2~
## $ GarageQual    <chr> "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "T~
## $ GarageCond    <chr> "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "T~
## $ PavedDrive    <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "~
## $ WoodDeckSF    <int> 0, 298, 0, 40, 235, 140, 0, 48, 0, 0, 0, 0, 406, 222, 28~
## $ OpenPorchSF   <int> 61, 0, 35, 30, 204, 0, 213, 112, 0, 102, 0, 0, 90, 32, 2~
## $ EnclosedPorch <int> 0, 0, 272, 0, 228, 0, 176, 0, 0, 0, 0, 205, 0, 0, 0, 0, ~
## $ X3SsnPorch    <int> 0, 0, 0, 320, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ ScreenPorch   <int> 0, 0, 0, 0, 0, 176, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ PoolArea      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ PoolQC        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ Fence         <chr> NA, NA, NA, "MnPrv", NA, NA, "GdWo", "GdPrv", NA, NA, "M~
## $ MiscFeature   <chr> NA, NA, NA, "Shed", "Shed", NA, NA, NA, "Shed", NA, NA, ~
## $ MiscVal       <int> 0, 0, 0, 700, 350, 0, 0, 0, 700, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ MoSold        <int> 2, 5, 2, 10, 11, 9, 5, 7, 3, 6, 5, 6, 5, 5, 12, 6, 1, 4,~
## $ YrSold        <int> 2008, 2007, 2006, 2009, 2009, 2008, 2008, 2007, 2010, 20~
## $ SaleType      <chr> "WD", "WD", "WD", "WD", "WD", "WD", "WD", "WD", "WD", "W~
## $ SaleCondition <chr> "Normal", "Normal", "Abnorml", "Normal", "Normal", "Norm~
## $ SalePrice     <int> 208500, 181500, 140000, 143000, 200000, 144000, 157000, ~
```

Correlation

Models

Review SLR

SLR

```
fit1 <- lm(SalePrice ~ TotalBsmtSF, data = data_sub)
summary(fit1)
```

```
##
## Call:
## lm(formula = SalePrice ~ TotalBsmtSF, data = data_sub)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -68998 -20480  -1717  17460  51506
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.522e+05  2.509e+03  60.648  < 2e-16 ***
## TotalBsmtSF 1.257e+01  2.296e+00   5.474 6.05e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22900 on 731 degrees of freedom
## Multiple R-squared:  0.03938,    Adjusted R-squared:  0.03806
## F-statistic: 29.96 on 1 and 731 DF,  p-value: 6.051e-08
```

The model is given as

$$\mu(SalePrice|TotalBsmtSF) = \beta_0 + \beta_1 TotalBsmtSF$$

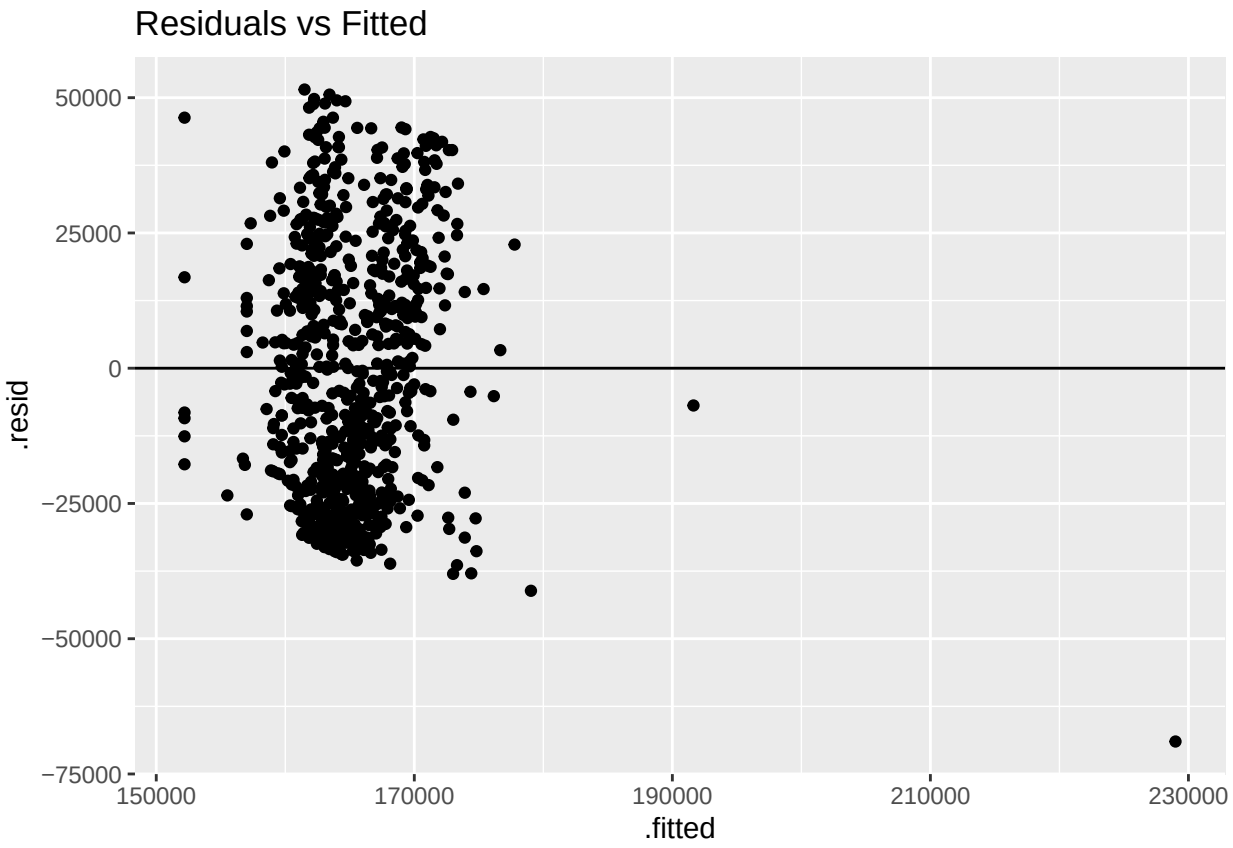
The fitted model is

$$\begin{aligned}\hat{\mu}(SalePrice|TotalBsmtSF) &= \hat{\beta}_0 + \hat{\beta}_1 TotalBsmtSF \\ &= 1.522e05 + 1.257e01 * TotalBsmtSF\end{aligned}$$

The model is a poor fit (Adjusted R-squared: 0.03806).

How about model assumption? One of the model assumptions is constant or equal variance, which is violated. Other assumptions include normality, independence, and linearlity.

```
library(broom)
fit1_df <- augment(fit1)
fit1_df %>% ggplot(aes(x = .fitted, y = .resid)) + geom_point() +
  geom_hline(yintercept = 0) +
  labs(title = "Residuals vs Fitted")
```



MLR

```
fit2 <- lm(SalePrice ~ TotalBsmtSF + LotArea, data = data_sub)
summary(fit2)
```

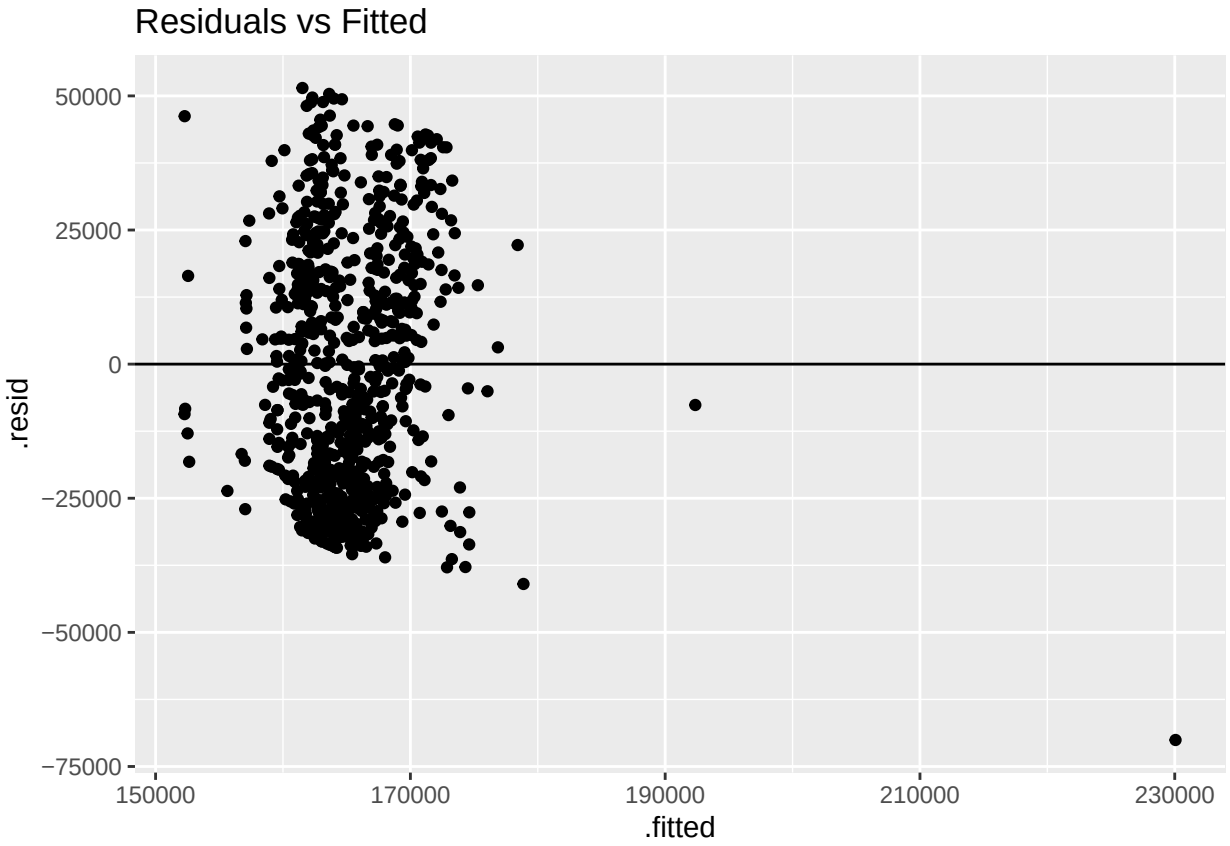
```
##
## Call:
## lm(formula = SalePrice ~ TotalBsmtSF + LotArea, data = data_sub)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -70063 -20444  -1867  17553  51473
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.520e+05  2.653e+03  57.301  < 2e-16 ***
## TotalBsmtSF 1.239e+01  2.455e+00   5.048 5.66e-07 ***
## LotArea     3.629e-02  1.779e-01   0.204    0.838
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22910 on 730 degrees of freedom
## Multiple R-squared:  0.03943,    Adjusted R-squared:  0.0368
## F-statistic: 14.98 on 2 and 730 DF,  p-value: 4.195e-07
```

The fitted model is

$$\hat{\mu}(SalePrice|TotalBsmtSF) = \hat{\beta}_0 + \hat{\beta}_1 TotalBsmtSF + \hat{\beta}_2 LotArea$$

The fit is till pretty poor.

```
fit2_df <- augment(fit2)
fit2_df %>% ggplot(aes(x = .fitted, y = .resid)) + geom_point() +
  geom_hline(yintercept = 0) +
  labs(title = "Residuals vs Fitted")
```



Fit 3 Add another continuous variable and see if R-squared or Adjusted R-squared changes.

Fit 4 Add another continuous variable.

Fit 5 Add another continuous variable.

Fit 6 Return to fit1 but add an additional categorical variable.

```
fit6 <- lm(SalePrice ~ TotalBsmtSF + MSZoning, data = data_sub)
summary(fit6)
```

```
##
## Call:
## lm(formula = SalePrice ~ TotalBsmtSF + MSZoning, data = data_sub)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -65048 -19369   -896  16681  52378
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.198e+05  2.238e+04   5.354 1.15e-07 ***
## TotalBsmtSF 1.159e+01  2.268e+00   5.108 4.15e-07 ***
## MSZoningFV  4.741e+04  2.252e+04   2.105   0.0356 *
## MSZoningRH  2.390e+04  2.331e+04   1.025   0.3055
## MSZoningRL  3.441e+04  2.223e+04   1.548   0.1221
## MSZoningRM  1.905e+04  2.238e+04   0.851   0.3948
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22210 on 727 degrees of freedom
## Multiple R-squared:  0.1011, Adjusted R-squared:  0.09494
## F-statistic: 16.36 on 5 and 727 DF,  p-value: 2.587e-15
```

First notice that MSZoning has 5 levels

```
#levels(data_sub$MSZoning) #NULL
# str(data_sub$MSZoning) #chr
# Convert to the right data type
data_sub$MSZoning <- factor(data_sub$MSZoning) # base R
levels(data_sub$MSZoning)
```

```
## [1] "C (all)" "FV"      "RH"      "RL"      "RM"
```

```
unique(data_sub$MSZoning)
```

```
## [1] RL      RM      FV      C (all) RH
## Levels: C (all) FV RH RL RM
```

Not quite sure why the default reference group is named C (all).

```
levels(data_sub$MSZoning)
```

```
## [1] "C (all)" "FV"      "RH"      "RL"      "RM"
```

The model is given as

$$\mu(SalePrice|TotalBsmtSF, MSZoning) = \beta_0 + \beta_1 TotalBsmtSF + \beta_2 MSZoningFV +$$

$$\beta_3 MSZoningRH + \beta_4 MSZoningRL + \beta_5 MSZoningRM,$$

where MSZoningFV, MSZoningFV, MSZoningRL, and MSZoningRM are all indivator variables, i.e., for MSZoningFV, then MSZoningFV = 1 if MSZoning = FV, MSZoningFV = 0 otherwise.

$$MSZoningFV = 1 \text{ if MSZoning} = \text{FV}$$

$$MSZoningFV = 0 \text{ otherwise.}$$

So then, for MSZoning = C (all) (the reference group), the regression line is given as

$$\mu(SalePrice|TotalBsmtSF, MSZoning = ref) = \beta_0 + \beta_1 TotalBsmtSF$$

For MSZoning = FV, the regression line is given as

$$\mu(SalePrice|TotalBsmtSF, MSZoning = FV) = \beta_0 + \beta_1 TotalBsmtSF + \beta_2 MSZoningFV$$

$$= (\beta_0 + \beta_2) + \beta_1 TotalBsmtSF$$

For MSZoning = RH, the regression line is given as

$$\mu(SalePrice|TotalBsmtSF, MSZoning = RH) = \beta_0 + \beta_1 TotalBsmtSF + \beta_3 MSZoningRH$$

$$= (\beta_0 + \beta_3) + \beta_1 TotalBsmtSF$$

For MSZoning = RL, the regression line is given as

$$\mu(SalePrice|TotalBsmtSF, MSZoning = RL) = \beta_0 + \beta_1 TotalBsmtSF + \beta_4 MSZoningRL$$

$$= (\beta_0 + \beta_4) + \beta_1 TotalBsmtSF,$$

etc.

If we were to plot these regression lines altogether, they'll be parallel to one another. They only differ by intercepts. Since the effects are additive, this model is called additive model.

Another thing to note is that

- β_2 is the estimated mean response switching from MSZoningFV to the reference group while keeping all else constant,
- β_3 is the estimated mean response switching from MSZoningFV to the reference group while keeping all else constant,
- β_4 is... MSZoningRH to the reference group while...,
- β_5 is... MSZoningRL to the reference group while...

Depends on our research of interest, sometimes we may want to use the relevel() function either to re-level/re-order the group or set a different reference group. When we do, we may end up with a different "re-parametrized" model, which is still a valid model so long as we keep the number of coefficients βs is right.

Again back to the model that fits the intercept, TotalBsmtSF and MSZoning, we know that for any model to be a valid model we need a total of 6 βs. This is because 1 (intercept) + 1 (TotalBsmtSF: continuous) + 4 (MSZoning has 5 levels but 1 of them is the reference group).

Is fit6 a valid model?

Fit 7 fit a re-parametrized model of fit6 while setting MSZoning = FV as the refernece group.

```
unique(data_sub$MSZoning)
```

```
## [1] RL      RM      FV      C (all) RH
## Levels: C (all) FV RH RL RM
```

```
data_sub$MSZoning <- relevel(data_sub$MSZoning, ref = "FV")
```

This didn't work because we didn't save it. Now it does. Now FV is the reference group.

```
levels(data_sub$MSZoning)
```

```
## [1] "FV"      "C (all)" "RH"      "RL"      "RM"
```

For fit 7, without fitting the model yet, can you write down the model?

For fit 7, without fitting the model yet, can you interpret what each β mean?

```
fit7 <- lm(SalePrice ~ TotalBsmtSF + MSZoning, data = data_sub)
summary(fit7)
```

```
##
## Call:
## lm(formula = SalePrice ~ TotalBsmtSF + MSZoning, data = data_sub)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -65048 -19369   -896  16681  52378
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     167242.338   4141.235  40.385  < 2e-16 ***
## TotalBsmtSF         11.588      2.268   5.108 4.15e-07 ***
## MSZoningC (all) -47410.069  22522.327  -2.105 0.035632 *
## MSZoningRH      -23509.739   7915.830  -2.970 0.003076 **
## MSZoningRL      -12996.874   3786.531  -3.432 0.000632 ***
## MSZoningRM      -28357.139   4492.716  -6.312 4.79e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22210 on 727 degrees of freedom
## Multiple R-squared:  0.1011, Adjusted R-squared:  0.09494
## F-statistic: 16.36 on 5 and 727 DF,  p-value: 2.587e-15
```

Is fit7 a valid model?

Referring back to the plot of SalePrice vs MSZoning, sometimes it is better to fit an ANOVA model instead especially if the trend is not clear. The advantage of ANOVA model is that it let us to test the overall effect of MSZoning on SalePrice.

```
fit7_2 <- lm(SalePrice ~ MSZoning, data = data_sub)
anova(fit7_2)
```

```
## Analysis of Variance Table
##
## Response: SalePrice
##            Df     Sum Sq    Mean Sq F value    Pr(>F)
## MSZoning    4 2.7471e+10 6867625293  13.458 1.375e-10 ***
## Residuals 728 3.7149e+11  510288418
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since p-val is low (Pr(>F) = 1.375e-10 ***), we conclude that there is a significant effect of MSZoning on SalePrice. However, without looking further, we wouldn't know which level of MSZoning has the effect since the null hypothesis Ho corresponding to the MSZoning row is to test $H_o : \alpha_i = 0$ for $i = 1, 2, 3, 4, 5$. When we reject the Ho, we can also say that at least one of the α_i differs from 0.

Fit 8 Return to fit2 and fit a model with interaction term. Such model is also called non-additive model

```r
fit8 <- lm(SalePrice ~ TotalBsmtSF + LotArea + TotalBsmtSF:LotArea, data = data_sub)
summary(fit8)
```

```
##
## Call:
## lm(formula = SalePrice ~ TotalBsmtSF + LotArea + TotalBsmtSF:LotArea,
##     data = data_sub)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -47084 -19896  -1420  17448  53229
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.401e+05  3.918e+03  35.758  < 2e-16 ***
## TotalBsmtSF          2.190e+01  3.357e+00   6.524 1.28e-10 ***
## LotArea              7.104e-01  2.408e-01   2.950  0.00328 **
## TotalBsmtSF:LotArea -4.151e-04  1.012e-04  -4.103 4.54e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22670 on 729 degrees of freedom
## Multiple R-squared:  0.06111,    Adjusted R-squared:  0.05725
## F-statistic: 15.82 on 3 and 729 DF,  p-value: 5.67e-10
```

The p-val associated with TotalBsmtSF:LotArea ($\Pr(>|t|) = 4.54e\text{-}05$) is significant, whcih means that there is a significant interactive effect between TotalBsmtSF and LotArea. That is, the effect of TotalBsmtSF on the response variable (SalePrice) depends on/interacts with LotArea, and vice versa.

None of the model we fit so far is a good fit. But it is important to start with simple cases.