HousePricePrediction

Yasko

2022-10-05

R Markdown

```
library(knitr)
library(ggplot2)
library(plyr)
library(dplyr)
```

```
##
##           : 'dplyr'
##                  'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##                  'package:stats':
##
##     filter, lag
##                  'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(caret)
```

```
##                : lattice
```

```
library(gridExtra)
```

```
##
##           : 'gridExtra'
##                'package:dplyr':
##
##     combine
```

```
library(scales)
library(Rmisc)
library(ggrepel)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
##           : 'randomForest'

##                 'package:gridExtra':
##
##     combine

##                 'package:dplyr':
##
##     combine

##                 'package:ggplot2':
##
##     margin
```

```
library(psych)
```

```
##
##           : 'psych'

##                 'package:randomForest':
##
##     outlier

##                  'package:scales':
##
##     alpha, rescale

##                  'package:ggplot2':
##
##     %+%, alpha
```

```
library(xgboost)
```

```
##
##           : 'xgboost'

##                 'package:dplyr':
##
##     slice
```

```
train <- read.csv("train.csv", stringsAsFactors = F)
test <- read.csv("test.csv", stringsAsFactors = F)
```

```
dim(train)
```

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

```
str(train[,c(1:10, 81)])
```

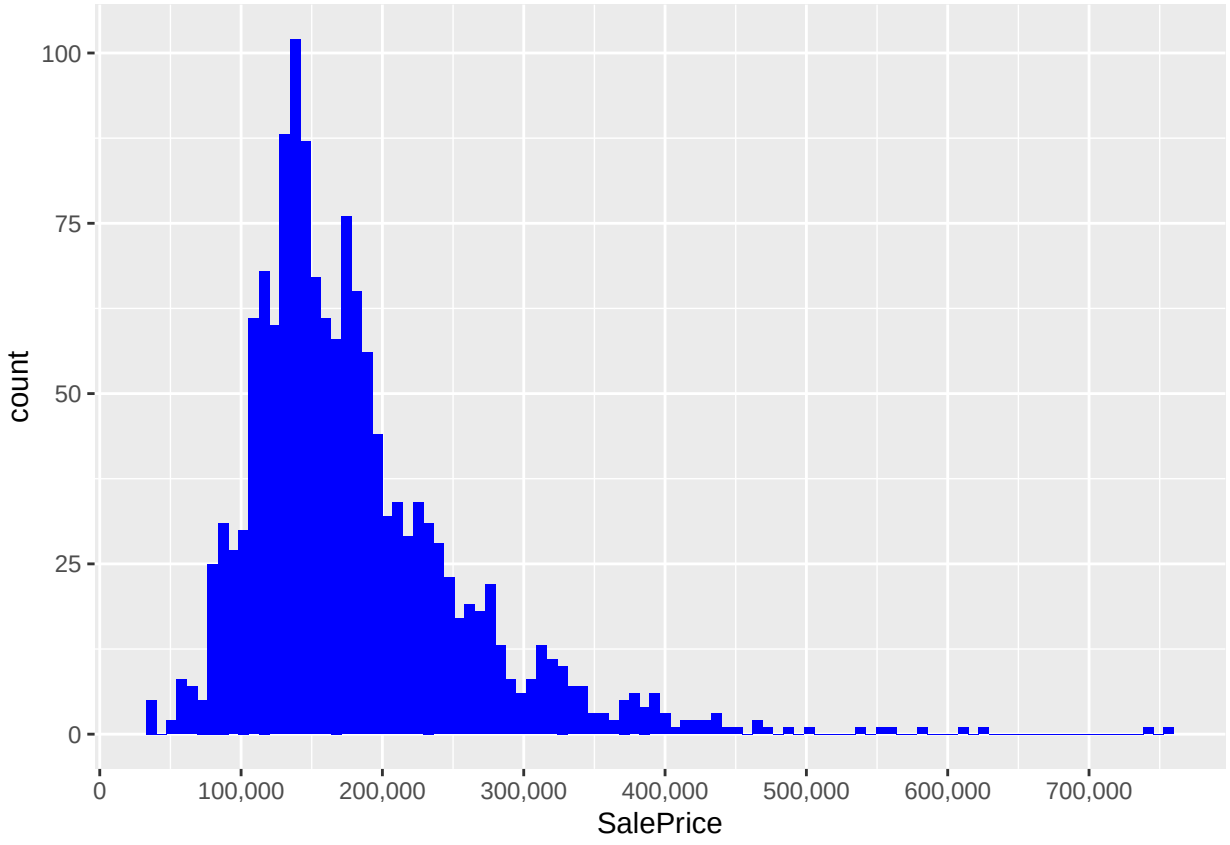
```
## 'data.frame':    1460 obs. of  11 variables:
##  $ Id         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning   : chr  "RL" "RL" "RL" "RL" ...
##  $ LotFrontage: int  65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea    : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ Street     : chr  "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley      : chr  NA NA NA NA ...
##  $ LotShape   : chr  "Reg" "Reg" "IR1" "IR1" ...
##  $ LandContour: chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
##  $ Utilities  : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
```

```
##  $ SalePrice  : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
```

```
test_labels <- test$Id
test$Id <- NULL
train$Id <- NULL
test$SalePrice <- NA
all <-rbind(train, test)
dim(all)
```

```
## [1] 2919   80
```

```
ggplot(data = all[!is.na(all$SalePrice),], aes(x = SalePrice)) +
  geom_histogram(fill = "blue", bins = 100) +
  scale_x_continuous(breaks = seq(0, 800000, by = 100000), labels = comma)
```



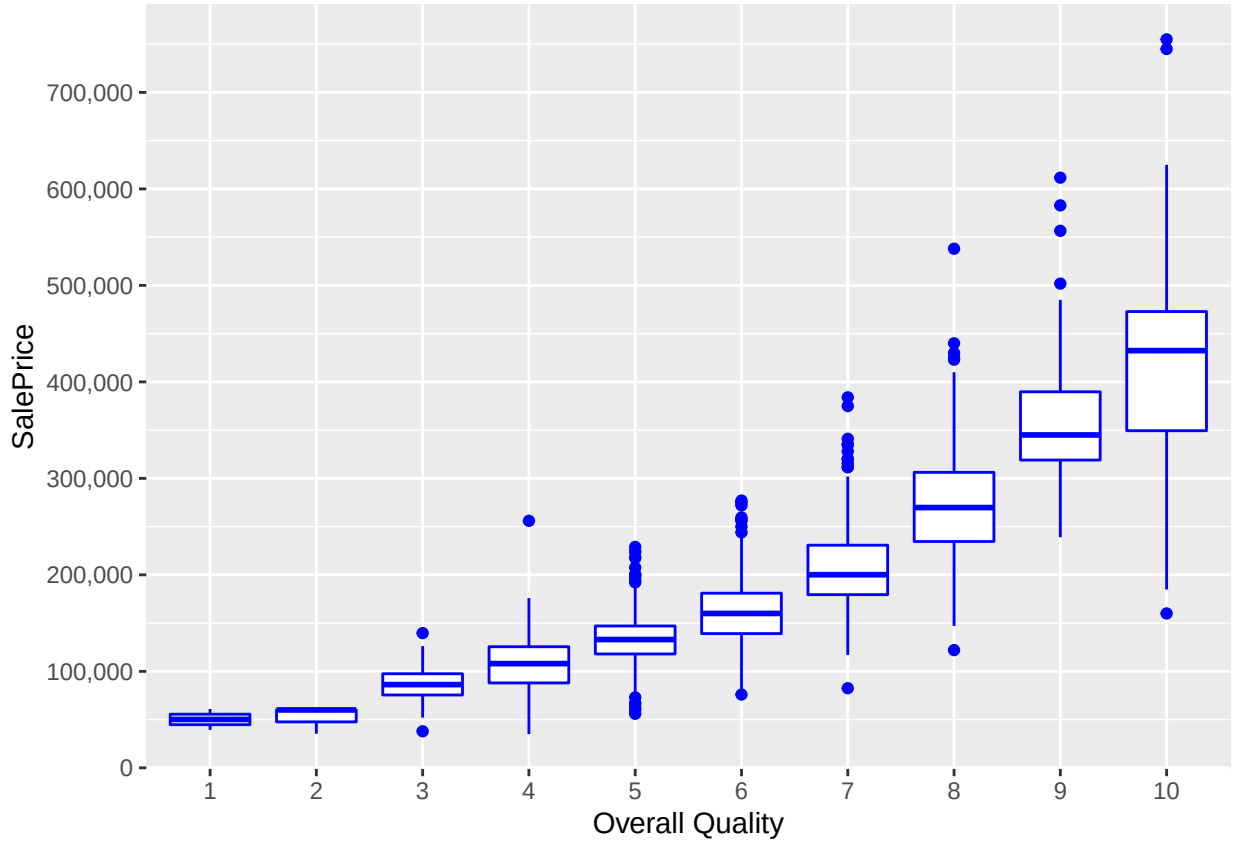

```
summary(all$SalePrice)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   34900  129975  163000  180921  214000  755000    1459
```

```
numericVars <- which(sapply(all, is.numeric))
numericVarNames <- names(numericVars)
cat('There are', length(numericVars), 'numeric variables')
```

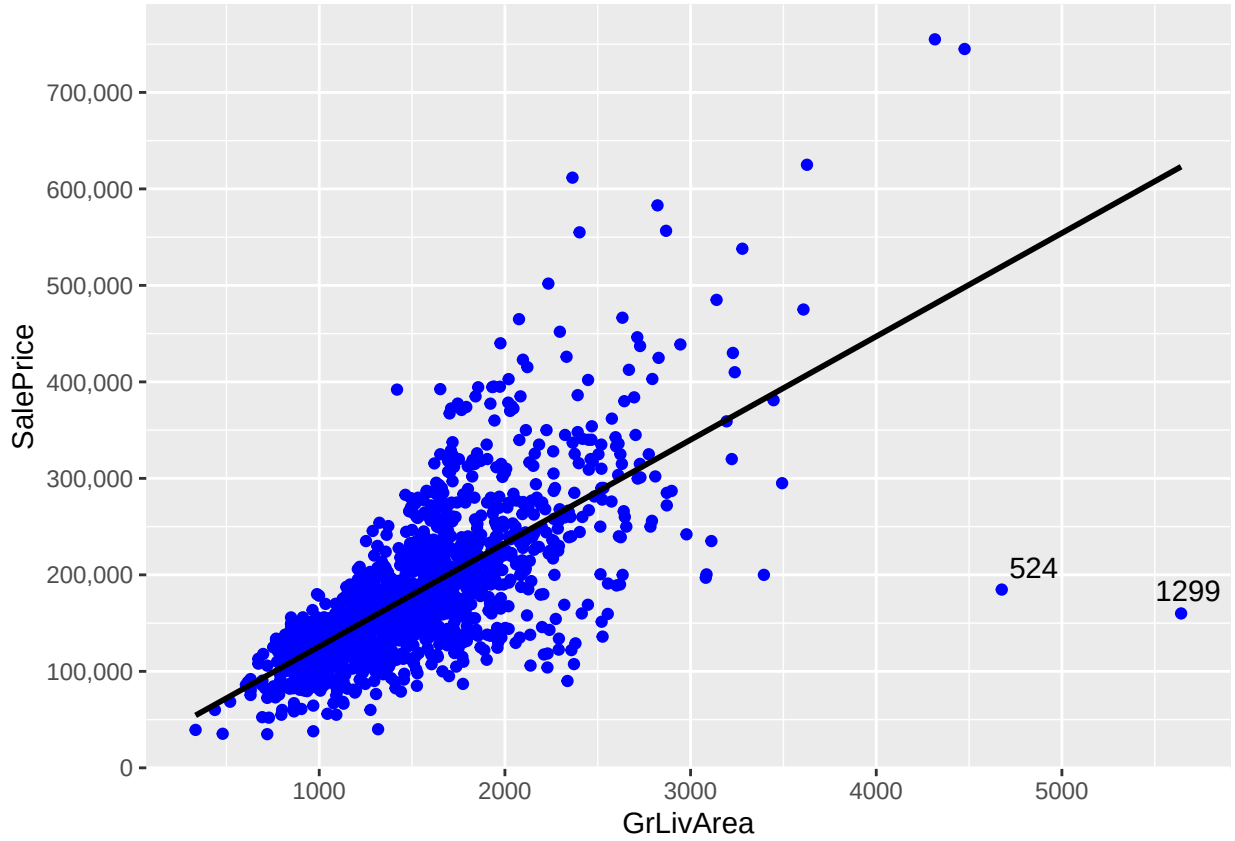
```
## There are 37 numeric variables
```

```
ggplot(data = all[!is.na(all$SalePrice), ], aes(x = factor(OverallQual), y = SalePrice)) +
  geom_boxplot(col = "blue") + labs(x = 'Overall Quality') +
  scale_y_continuous(breaks = seq(0, 800000, by = 100000), labels = comma)
```




```
ggplot(data = all[!is.na(all$SalePrice),], aes(x = GrLivArea, y = SalePrice)) +
  geom_point(col = 'blue') + geom_smooth(method = 'lm', se = FALSE, color = "black", aes(group = 1)) +
  scale_y_continuous(breaks = seq(0, 800000, by = 100000), labels = comma) +
  geom_text_repel(aes(label = ifelse(all$GrLivArea[!is.na(all$SalePrice)]>4500, rownames(all), '')))
```

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




```
all[c(524, 1299), c('SalePrice', "GrLivArea", "OverallQual")]
```

```
##      SalePrice GrLivArea OverallQual
## 524     184750      4676          10
## 1299    160000      5642          10
```

```
NAcol <- which(colSums(is.na(all)) > 0)
sort(colSums(sapply(all[NAcol], is.na)), decreasing = TRUE)
```

```
##       PoolQC  MiscFeature        Alley        Fence    SalePrice  FireplaceQu
##         2909         2814         2721         2348         1459         1420
##  LotFrontage  GarageYrBlt GarageFinish   GarageQual   GarageCond   GarageType
##          486          159          159          159          159          157
##     BsmtCond BsmtExposure     BsmtQual BsmtFinType2 BsmtFinType1   MasVnrType
##           82           82           81           80           79           24
##   MasVnrArea     MSZoning    Utilities BsmtFullBath BsmtHalfBath   Functional
##           23            4            2            2            2            2
##  Exterior1st  Exterior2nd   BsmtFinSF1   BsmtFinSF2    BsmtUnfSF  TotalBsmtSF
##            1            1            1            1            1            1
##   Electrical  KitchenQual   GarageCars   GarageArea     SaleType
##            1            1            1            1            1
```

```
all$PoolQC[is.na(all$PoolQC)] <- 'None'
Qualities <- c('None' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
all$PoolQC<-as.integer(revalue(all$PoolQC, Qualities))
```

```
## The following `from` values were not present in `x`: Po, TA
```

```
table(all$PoolQC)
```

```
##
##    0    2    4    5
## 2909    2    4    4
```

```
all[all$PoolArea>0 & all$PoolQC==0, c('PoolArea', 'PoolQC', 'OverallQual')]
```

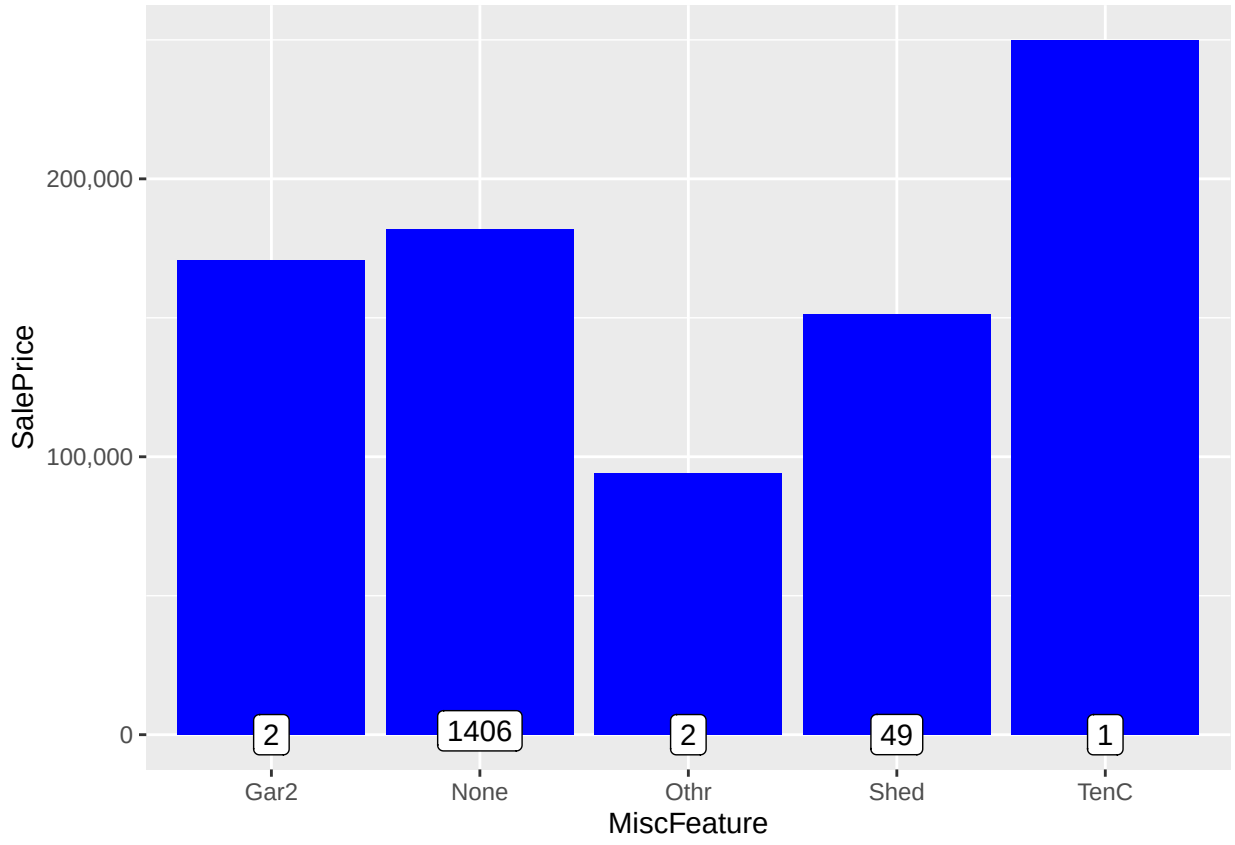
```
##      PoolArea PoolQC OverallQual
## 2421      368      0           4
## 2504      444      0           6
## 2600      561      0           3
```

```
all$PoolQC[2421] <- 2
all$PoolQC[2504] <- 3
all$PoolQC[2600] <- 2
```

```
all$MiscFeature[is.na(all$MiscFeature)] <- 'None'
all$MiscFeature <- as.factor(all$MiscFeature)

ggplot(all[!is.na(all$SalePrice),], aes(x=MiscFeature, y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue') +
        scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..))
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```



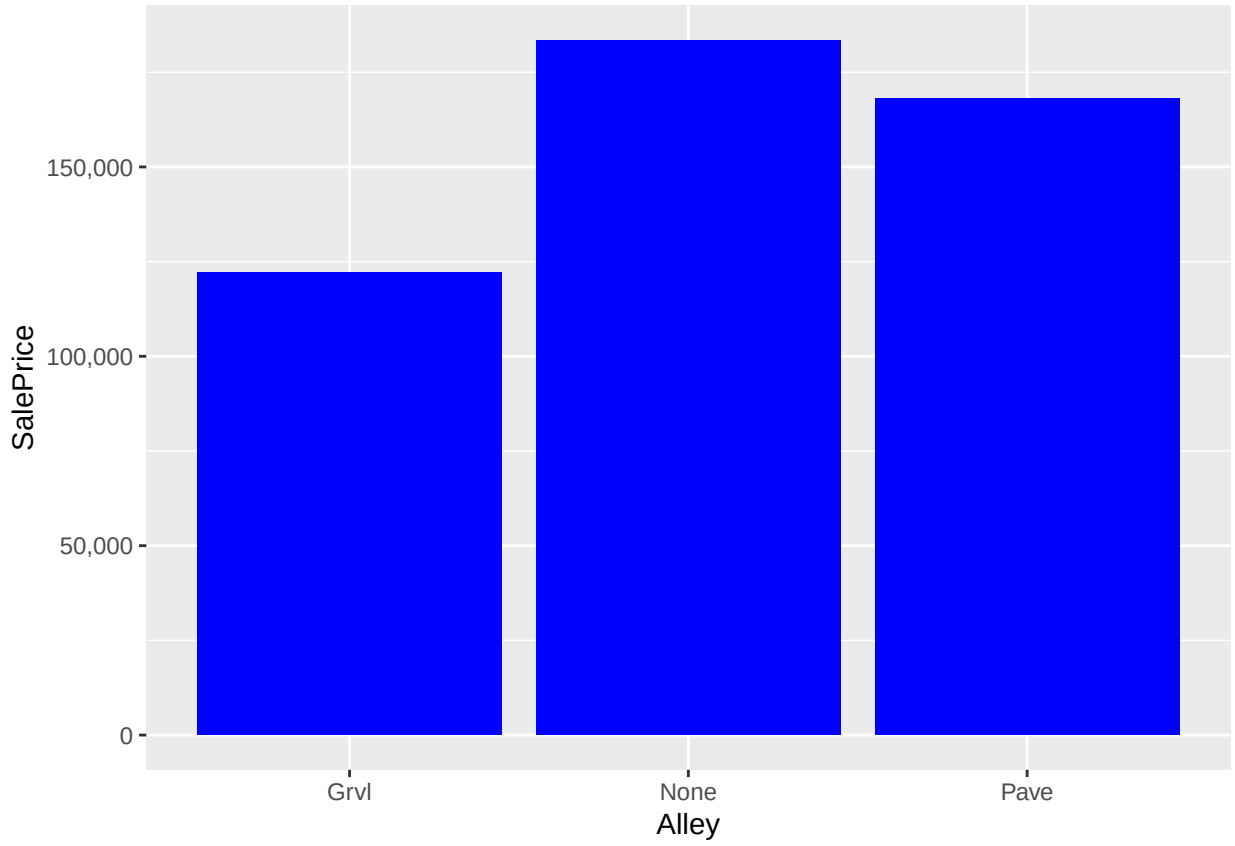
```
table(all$MiscFeature)
```

```
##
## Gar2 None Othr Shed TenC
##    5 2814    4   95    1
```

```
all$Alley[is.na(all$Alley)] <- 'None'
all$Alley <- as.factor(all$Alley)

ggplot(all[!is.na(all$SalePrice),], aes(x=Alley, y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue')+
        scale_y_continuous(breaks= seq(0, 200000, by=50000), labels = comma)
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
table(all$Alley)
```

```
##
## Grvl None Pave
##  120 2721   78
```

```
all$Fence[is.na(all$Fence)] <- 'None'
table(all$Fence)
```

```
##
## GdPrv  GdWo MnPrv  MnWw  None
##   118   112   329    12  2348
```

```
all[!is.na(all$SalePrice),] %>% group_by(Fence) %>% summarise(median = median(SalePrice), counts=n())
```

```
## # A tibble: 5 x 3
##   Fence median counts
##   <chr>  <dbl>  <int>
## 1 GdPrv 167500     59
## 2 GdWo  138750     54
## 3 MnPrv 137450    157
## 4 MnWw  130000     11
## 5 None  173000   1179
```

```
all$Fence <- as.factor(all$Fence)
```

```
all$FireplaceQu[is.na(all$FireplaceQu)] <- 'None'
all$FireplaceQu<-as.integer(revalue(all$FireplaceQu, Qualities))
table(all$FireplaceQu)
```

```
##
##    0    1    2    3    4    5
## 1420   46   74  592  744   43
```

```
table(all$Fireplaces)
```

```
##
##    0    1    2    3    4
## 1420 1268  219   11    1
```

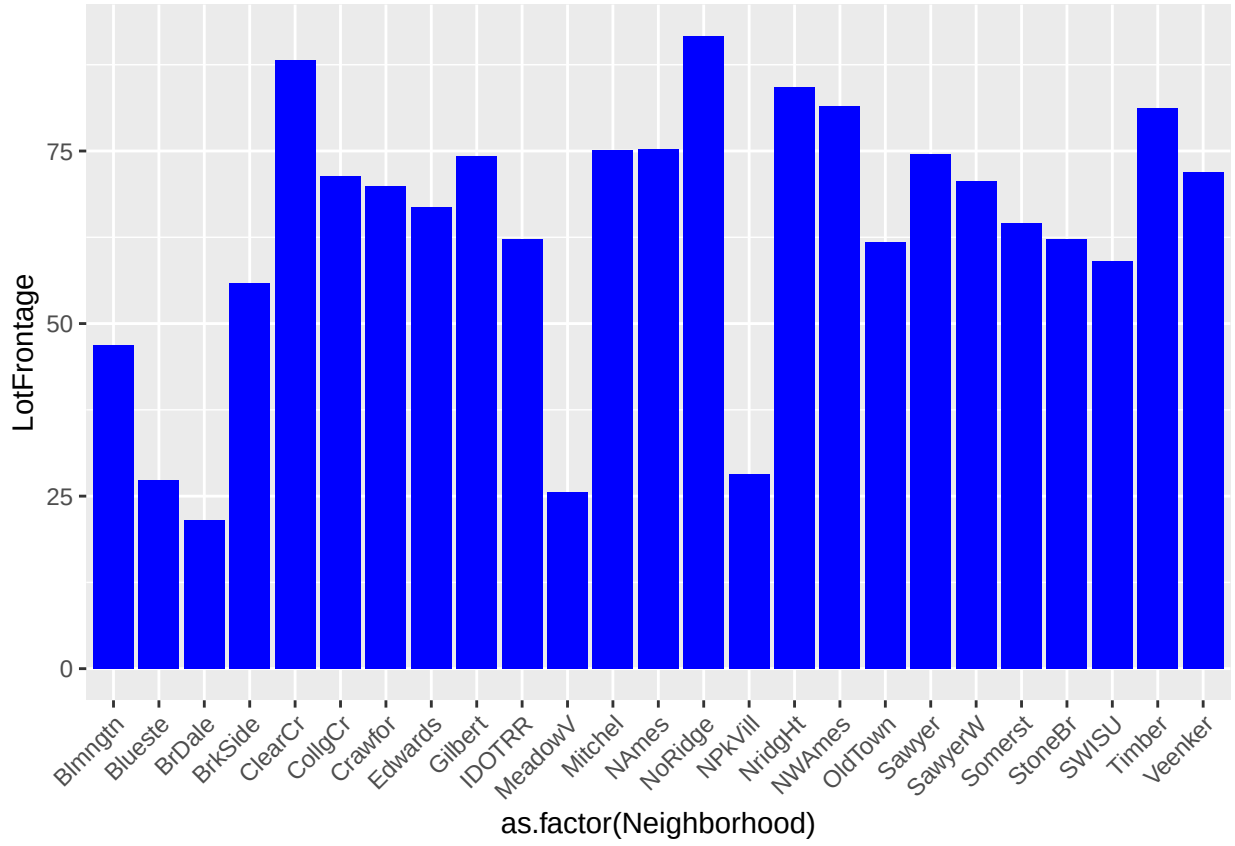
```
sum(table(all$Fireplaces))
```

```
## [1] 2919
```

```
ggplot(all[!is.na(all$LotFrontage),], aes(x=as.factor(Neighborhood), y=LotFrontage)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue') +
        theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
for (i in 1:nrow(all)){
        if(is.na(all$LotFrontage[i])){
               all$LotFrontage[i] <- as.integer(median(all$LotFrontage[all$Neighborhood==all$Neighborho
        }
}
all$LotShape<-as.integer(revalue(all$LotShape, c('IR3'=0, 'IR2'=1, 'IR1'=2, 'Reg'=3)))
table(all$LotShape)
```

```
##
##    0    1    2    3
##   16   76  968 1859
```

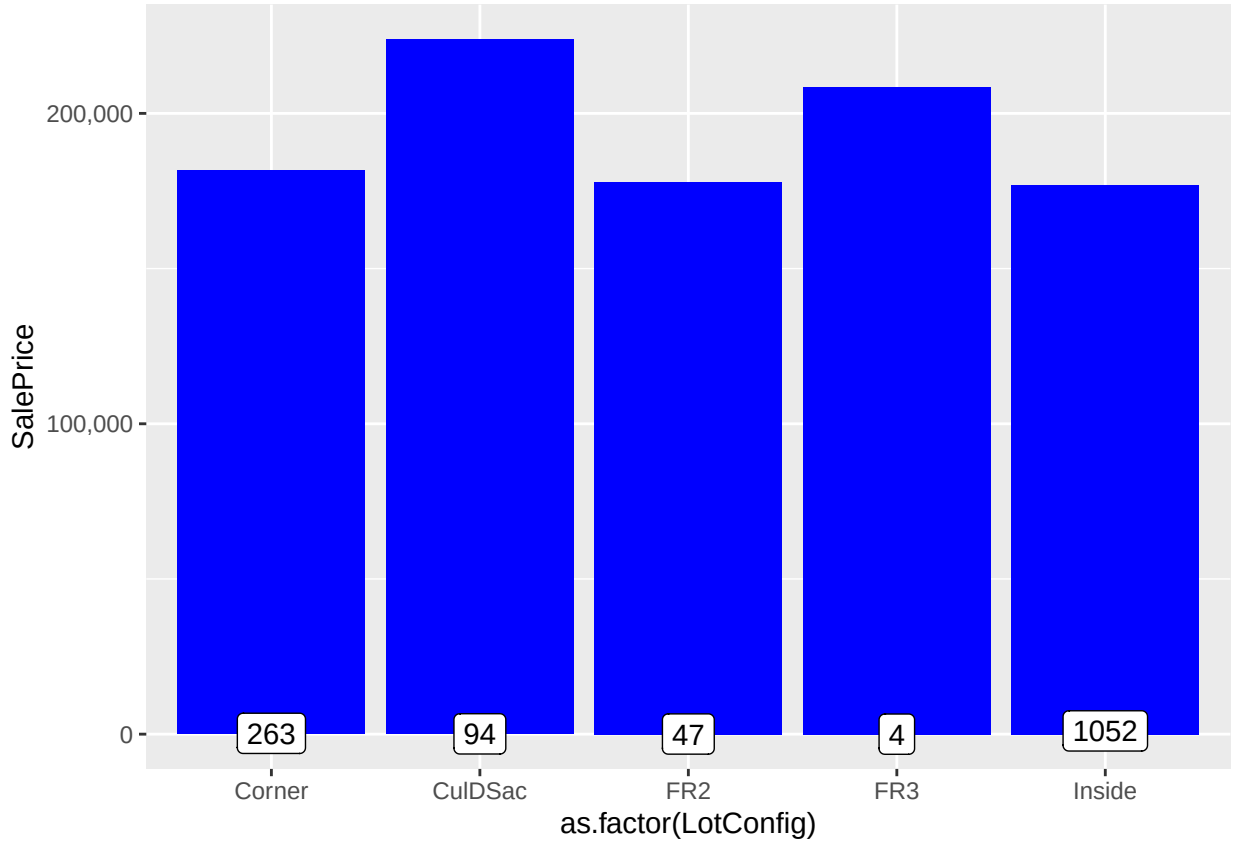
```
sum(table(all$LotShape))
```

```
## [1] 2919
```

```
ggplot(all[!is.na(all$SalePrice),], aes(x=as.factor(LotConfig), y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue')+
        scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..))
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
all$LotConfig <- as.factor(all$LotConfig)
table(all$LotConfig)
```

```
##
##  Corner CulDSac     FR2     FR3  Inside
##     511     176      85      14    2133
```

```
sum(table(all$LotConfig))
```

```
## [1] 2919
```

```
all$GarageYrBlt[is.na(all$GarageYrBlt)] <- all$YearBuilt[is.na(all$GarageYrBlt)]
length(which(is.na(all$GarageType) & is.na(all$GarageFinish) & is.na(all$GarageCond) & is.na(all$Garage
```

```
## [1] 157
```

```
kable(all[!is.na(all$GarageType) & is.na(all$GarageFinish), c('GarageCars', 'GarageArea', 'GarageType',
```

	GarageCars	GarageArea	GarageType	GarageCond	GarageQual	GarageFinish
2127	1	360	Detchd	NA	NA	NA
2577	NA	NA	Detchd	NA	NA	NA

```
all$GarageCond[2127] <- names(sort(-table(all$GarageCond)))[1]
all$GarageQual[2127] <- names(sort(-table(all$GarageQual)))[1]
all$GarageFinish[2127] <- names(sort(-table(all$GarageFinish)))[1]
```

```
#display "fixed" house
kable(all[2127, c('GarageYrBlt', 'GarageCars', 'GarageArea', 'GarageType', 'GarageCond', 'GarageQual',
```

	GarageYrBlt	GarageCars	GarageArea	GarageType	GarageCond	GarageQual	GarageFinish
2127	1910	1	360	Detchd	TA	TA	Unf

```
#fixing 3 values for house 2577
all$GarageCars[2577] <- 0
all$GarageArea[2577] <- 0
all$GarageType[2577] <- NA

#check if NAs of the character variables are now all 158
length(which(is.na(all$GarageType) & is.na(all$GarageFinish) & is.na(all$GarageCond) & is.na(all$Garage
```

```
## [1] 158
```

```
all$GarageType[is.na(all$GarageType)] <- 'No Garage'
all$GarageType <- as.factor(all$GarageType)
table(all$GarageType)
```

```
##
##    2Types    Attchd   Basment   BuiltIn   CarPort    Detchd No Garage
##        23      1723        36       186        15       778       158
```

```
all$GarageFinish[is.na(all$GarageFinish)] <- 'None'
Finish <- c('None'=0, 'Unf'=1, 'RFn'=2, 'Fin'=3)

all$GarageFinish<-as.integer(revalue(all$GarageFinish, Finish))
table(all$GarageFinish)
```

```
##
##    0    1    2    3
##  158 1231  811  719
```

```
all$GarageQual[is.na(all$GarageQual)] <- 'None'
all$GarageQual<-as.integer(revalue(all$GarageQual, Qualities))
table(all$GarageQual)
```

```
##
##    0    1    2    3    4    5
##  158    5  124 2605   24    3
```

```
all$GarageCond[is.na(all$GarageCond)] <- 'None'
all$GarageCond<-as.integer(revalue(all$GarageCond, Qualities))
table(all$GarageCond)
```

```
##
##    0    1    2    3    4    5
##  158   14   74 2655   15    3
```

```
length(which(is.na(all$BsmtQual) & is.na(all$BsmtCond) & is.na(all$BsmtExposure) & is.na(all$BsmtFinTyp
```

```
## [1] 79
```

```
all[!is.na(all$BsmtFinType1) & (is.na(all$BsmtCond)|is.na(all$BsmtQual)|is.na(all$BsmtExposure)|is.na(a
```

```
##      BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2
```

```
## 333        Gd       TA           No          GLQ         <NA>
## 949        Gd       TA         <NA>          Unf          Unf
## 1488       Gd       TA         <NA>          Unf          Unf
## 2041       Gd     <NA>           Mn          GLQ          Rec
## 2186       TA     <NA>           No          BLQ          Unf
## 2218     <NA>       Fa           No          Unf          Unf
## 2219     <NA>       TA           No          Unf          Unf
## 2349       Gd       TA         <NA>          Unf          Unf
## 2525       TA     <NA>           Av          ALQ          Unf
```

```
all$BsmtFinType2[333] <- names(sort(-table(all$BsmtFinType2)))[1]
all$BsmtExposure[c(949, 1488, 2349)] <- names(sort(-table(all$BsmtExposure)))[1]
all$BsmtCond[c(2041, 2186, 2525)] <- names(sort(-table(all$BsmtCond)))[1]
all$BsmtQual[c(2218, 2219)] <- names(sort(-table(all$BsmtQual)))[1]
all$BsmtQual[is.na(all$BsmtQual)] <- 'None'
all$BsmtQual<-as.integer(revalue(all$BsmtQual, Qualities))
```

```
## The following `from` values were not present in `x`: Po
```

```
table(all$BsmtQual)
```

```
##
##    0    2    3    4    5
##   79   88 1285 1209  258
```

```
all$BsmtCond[is.na(all$BsmtCond)] <- 'None'
all$BsmtCond<-as.integer(revalue(all$BsmtCond, Qualities))
```

```
## The following `from` values were not present in `x`: Ex
```

```
table(all$BsmtCond)
```

```
##
##    0    1    2    3    4
##   79    5  104 2609  122
```

```
all$BsmtExposure[is.na(all$BsmtExposure)] <- 'None'
Exposure <- c('None'=0, 'No'=1, 'Mn'=2, 'Av'=3, 'Gd'=4)

all$BsmtExposure<-as.integer(revalue(all$BsmtExposure, Exposure))
table(all$BsmtExposure)
```

```
##
##    0    1    2    3    4
##   79 1907  239  418  276
```

```
all$BsmtFinType1[is.na(all$BsmtFinType1)] <- 'None'
FinType <- c('None'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)

all$BsmtFinType1<-as.integer(revalue(all$BsmtFinType1, FinType))
table(all$BsmtFinType1)
```

```
##
##   0   1   2   3   4   5   6
##  79 851 154 288 269 429 849
```

```
all$BsmtFinType2[is.na(all$BsmtFinType2)] <- 'None'
FinType <- c('None'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)
```

```
all$BsmtFinType2<-as.integer(revalue(all$BsmtFinType2, FinType))
table(all$BsmtFinType2)
```

```
##
##    0    1    2    3    4    5    6
##   79 2494   87  105   68   52   34
```

```
all[(is.na(all$BsmtFullBath)|is.na(all$BsmtHalfBath)|is.na(all$BsmtFinSF1)|is.na(all$BsmtFinSF2)|is.na(
```

```
##      BsmtQual BsmtFullBath BsmtHalfBath BsmtFinSF1 BsmtFinSF2 BsmtUnfSF
## 2121        0           NA           NA         NA         NA        NA
## 2189        0           NA           NA          0          0         0
##      TotalBsmtSF
## 2121          NA
## 2189           0
```

```
all$BsmtFullBath[is.na(all$BsmtFullBath)] <-0
table(all$BsmtFullBath)
```

```
##
##    0    1    2    3
## 1707 1172   38    2
```

```
all$BsmtHalfBath[is.na(all$BsmtHalfBath)] <-0
table(all$BsmtHalfBath)
```

```
##
##    0    1    2
## 2744  171    4
```

```
all$BsmtFinSF1[is.na(all$BsmtFinSF1)] <-0
all$BsmtFinSF2[is.na(all$BsmtFinSF2)] <-0
all$BsmtUnfSF[is.na(all$BsmtUnfSF)] <-0
all$TotalBsmtSF[is.na(all$TotalBsmtSF)] <-0
```

```
length(which(is.na(all$MasVnrType) & is.na(all$MasVnrArea)))
```

```
## [1] 23
```

```
all[is.na(all$MasVnrType) & !is.na(all$MasVnrArea), c('MasVnrType', 'MasVnrArea')]
```

```
##      MasVnrType MasVnrArea
## 2611       <NA>        198
```

```
all$MasVnrType[2611] <- names(sort(-table(all$MasVnrType)))[2] #taking the 2nd value as the 1st is 'non
all[2611, c('MasVnrType', 'MasVnrArea')]
```

```
##      MasVnrType MasVnrArea
## 2611    BrkFace        198
```

```
all$MasVnrType[is.na(all$MasVnrType)] <- 'None'

all[!is.na(all$SalePrice),] %>% group_by(MasVnrType) %>% summarise(median = median(SalePrice), counts=n
```

```
## # A tibble: 4 x 3
##   MasVnrType median counts
##   <chr>       <dbl>  <int>
## 1 BrkCmn     139000     15
## 2 None       143125    872
```

```
## 3 BrkFace    181000    445
## 4 Stone      246839    128
```

```
Masonry <- c('None'=0, 'BrkCmn'=0, 'BrkFace'=1, 'Stone'=2)
all$MasVnrType<-as.integer(revalue(all$MasVnrType, Masonry))
table(all$MasVnrType)
```

```
##
##    0    1    2
## 1790  880  249
```

```
all$MasVnrArea[is.na(all$MasVnrArea)] <-0
```

```
all$MSZoning[is.na(all$MSZoning)] <- names(sort(-table(all$MSZoning)))[1]
all$MSZoning <- as.factor(all$MSZoning)
table(all$MSZoning)
```

```
##
## C (all)      FV      RH      RL      RM
##      25     139      26    2269     460
```

```
sum(table(all$MSZoning))
```

```
## [1] 2919
```

```
all$KitchenQual[is.na(all$KitchenQual)] <- 'TA' #replace with most common value
all$KitchenQual<-as.integer(revalue(all$KitchenQual, Qualities))
```

```
## The following `from` values were not present in `x`: None, Po
```

```
table(all$KitchenQual)
```

```
##
##    2    3    4    5
##   70 1493 1151  205
```

```
sum(table(all$KitchenQual))
```

```
## [1] 2919
```

```
table(all$KitchenAbvGr)
```

```
##
##    0    1    2    3
##    3 2785  129    2
```

```
sum(table(all$KitchenAbvGr))
```

```
## [1] 2919
```

```
table(all$Utilities)
```

```
##
## AllPub NoSeWa
##   2916      1
```

```
kable(all[is.na(all$Utilities) | all$Utilities=='NoSeWa', 1:9])
```

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities
945	20	RL	82	14375	Pave	None	2	Lvl	NoSeWa
1916	30	RL	109	21780	Grvl	None	3	Lvl	NA

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities
1946	20	RL	64	31220	Pave	None	2	Bnk	NA

```
all$Utilities <- NULL
all$Functional[is.na(all$Functional)] <- names(sort(-table(all$Functional)))[1]

all$Functional <- as.integer(revalue(all$Functional, c('Sal'=0, 'Sev'=1, 'Maj2'=2, 'Maj1'=3, 'Mod'=4, '
```

```
## The following `from` values were not present in `x`: Sal
```

```
table(all$Functional)
```

```
##
##    1    2    3    4    5    6    7
##    2    9   19   35   70   65 2719
```

```
sum(table(all$Functional))
```

```
## [1] 2919
```

```
all$Exterior1st[is.na(all$Exterior1st)] <- names(sort(-table(all$Exterior1st)))[1]

all$Exterior1st <- as.factor(all$Exterior1st)
table(all$Exterior1st)
```

```
##
## AsbShng AsphShn BrkComm BrkFace  CBlock CemntBd HdBoard ImStucc MetalSd Plywood
##      44       2       6      87       2     126     442       1     450     221
##   Stone  Stucco VinylSd Wd Sdng WdShing
##       2      43    1026     411      56
```

```
sum(table(all$Exterior1st))
```

```
## [1] 2919
```

```
all$Exterior2nd[is.na(all$Exterior2nd)] <- names(sort(-table(all$Exterior2nd)))[1]

all$Exterior2nd <- as.factor(all$Exterior2nd)
table(all$Exterior2nd)
```

```
##
## AsbShng AsphShn Brk Cmn BrkFace  CBlock CmentBd HdBoard ImStucc MetalSd   Other
##      38       4      22      47       3     126     406      15     447       1
## Plywood   Stone  Stucco VinylSd Wd Sdng Wd Shng
##     270       6      47    1015     391      81
```

```
sum(table(all$Exterior2nd))
```

```
## [1] 2919
```

```
all$ExterQual<-as.integer(revalue(all$ExterQual, Qualities))
```

```
## The following `from` values were not present in `x`: None, Po
```

```
table(all$ExterQual)
```

```
##
##    2    3    4    5
##   35 1798  979  107
```

```
sum(table(all$ExterQual))
```

```
## [1] 2919
```

```
all$ExterCond<-as.integer(revalue(all$ExterCond, Qualities))
```

```
## The following `from` values were not present in `x`: None
```

```
table(all$ExterCond)
```

```
##
##    1    2    3    4    5
##    3   67 2538  299   12
```

```
sum(table(all$ExterCond))
```

```
## [1] 2919
```

```
all$Electrical[is.na(all$Electrical)] <- names(sort(-table(all$Electrical)))[1]

all$Electrical <- as.factor(all$Electrical)
table(all$Electrical)
```

```
##
## FuseA FuseF FuseP   Mix SBrkr
##   188    50     8     1  2672
```

```
sum(table(all$Electrical))
```

```
## [1] 2919
```

```
all$SaleType[is.na(all$SaleType)] <- names(sort(-table(all$SaleType)))[1]

all$SaleType <- as.factor(all$SaleType)
table(all$SaleType)
```

```
##
##   COD   Con ConLD ConLI ConLw   CWD   New   Oth    WD
##    87     5    26     9     8    12   239     7  2526
```

```
sum(table(all$SaleType))
```

```
## [1] 2919
```

```
all$SaleCondition <- as.factor(all$SaleCondition)
table(all$SaleCondition)
```

```
##
## Abnorml AdjLand  Alloca  Family  Normal Partial
##     190      12      24      46    2402     245
```

```
sum(table(all$SaleCondition))
```

```
## [1] 2919
```

```
NAcol <- which(colSums(is.na(all)) > 0)
sort(colSums(sapply(all[NAcol], is.na)), decreasing = TRUE)
```

```
## SalePrice
##      1459
```

```
Charcol <- names(all[,sapply(all, is.character)])
Charcol
```

```
##  [1] "Street"       "LandContour"  "LandSlope"    "Neighborhood" "Condition1"
##  [6] "Condition2"   "BldgType"     "HouseStyle"   "RoofStyle"    "RoofMatl"
## [11] "Foundation"   "Heating"      "HeatingQC"    "CentralAir"   "PavedDrive"
```

```
cat('There are', length(Charcol), 'remaining columns with character values')
```

```
## There are 15 remaining columns with character values
```

```
all$Foundation <- as.factor(all$Foundation)
table(all$Foundation)
```

```
##
## BrkTil CBlock  PConc   Slab  Stone   Wood
##    311   1235   1308     49     11      5
```

```
sum(table(all$Foundation))
```

```
## [1] 2919
```

```
all$Heating <- as.factor(all$Heating)
table(all$Heating)
```

```
##
## Floor  GasA  GasW  Grav  OthW  Wall
##     1  2874    27     9     2     6
```

```
sum(table(all$Heating))
```

```
## [1] 2919
```

```
all$HeatingQC<-as.integer(revalue(all$HeatingQC, Qualities))
```

```
## The following `from` values were not present in `x`: None
```

```
table(all$HeatingQC)
```

```
##
##    1    2    3    4    5
##    3   92  857  474 1493
```

```
sum(table(all$HeatingQC))
```

```
## [1] 2919
```

```
all$CentralAir<-as.integer(revalue(all$CentralAir, c('N'=0, 'Y'=1)))
table(all$CentralAir)
```

```
##
##    0    1
##  196 2723
```

```
sum(table(all$CentralAir))
```

```
## [1] 2919
```

```
all$RoofStyle <- as.factor(all$RoofStyle)
table(all$RoofStyle)
```

```
##
```

```
##    Flat   Gable Gambrel     Hip Mansard    Shed
##      20    2310      22     551      11       5
```

```
sum(table(all$RoofStyle))
```

```
## [1] 2919
```

```
all$RoofMatl <- as.factor(all$RoofMatl)
table(all$RoofMatl)
```

```
##
## ClyTile CompShg Membran   Metal    Roll Tar&Grv WdShake WdShngl
##       1    2876       1       1       1      23       9       7
```

```
sum(table(all$RoofMatl))
```

```
## [1] 2919
```

```
all$LandContour <- as.factor(all$LandContour)
table(all$LandContour)
```

```
##
##  Bnk  HLS  Low  Lvl
##  117  120   60 2622
```

```
sum(table(all$LandContour))
```

```
## [1] 2919
```

```
all$LandSlope<-as.integer(revalue(all$LandSlope, c('Sev'=0, 'Mod'=1, 'Gtl'=2)))
table(all$LandSlope)
```

```
##
##    0    1    2
##   16  125 2778
```

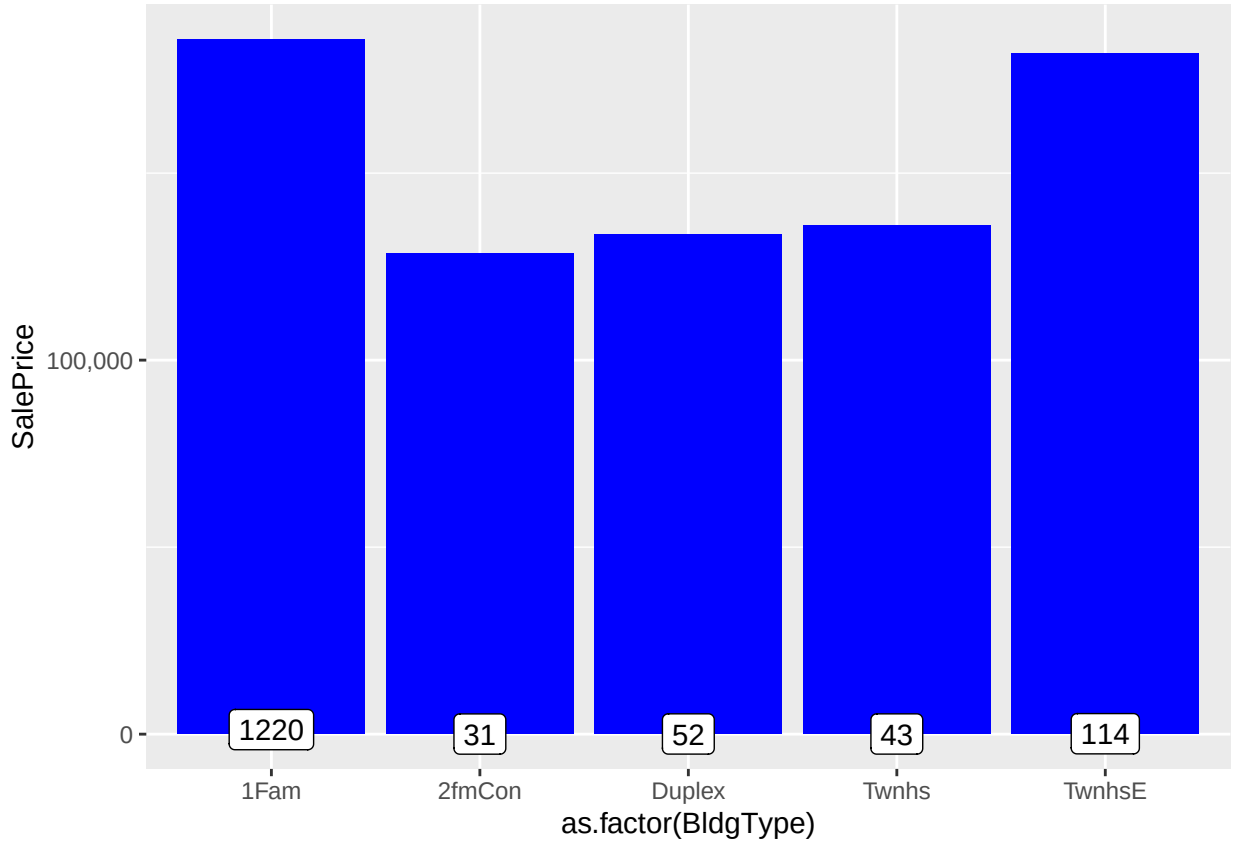
```
sum(table(all$LandSlope))
```

```
## [1] 2919
```

```
ggplot(all[!is.na(all$SalePrice),], aes(x=as.factor(BldgType), y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue')+
        scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..))
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
all$BldgType <- as.factor(all$BldgType)
table(all$BldgType)
```

```
##
##   1Fam 2fmCon Duplex  Twnhs TwnhsE
##   2425     62    109     96    227
```

```
sum(table(all$BldgType))
```

```
## [1] 2919
```

```
all$HouseStyle <- as.factor(all$HouseStyle)
table(all$HouseStyle)
```

```
##
## 1.5Fin 1.5Unf 1Story 2.5Fin 2.5Unf 2Story SFoyer   SLvl
##    314     19   1471      8     24    872     83    128
```

```
sum(table(all$HouseStyle))
```

```
## [1] 2919
```

```
all$Neighborhood <- as.factor(all$Neighborhood)
table(all$Neighborhood)
```

```
##
## Blmngtn Blueste  BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert  IDOTRR
##      28      10      30     108      44     267     103     194     165      93
## MeadowV Mitchel   NAmes NoRidge NPkVill NridgHt  NWAmes OldTown  Sawyer SawyerW
```

```
##      37     114     443      71      23     166     131     239     151     125
## Somerst StoneBr   SWISU  Timber Veenker
##     182      51      48      72      24
```

```
sum(table(all$Neighborhood))
```

```
## [1] 2919
```

```
all$Condition1 <- as.factor(all$Condition1)
table(all$Condition1)
```

```
##
## Artery  Feedr   Norm   PosA   PosN   RRAe   RRAn   RRNe   RRNn
##     92    164   2511     20     39     28     50      6      9
```

```
sum(table(all$Condition1))
```

```
## [1] 2919
```

```
all$Condition2 <- as.factor(all$Condition2)
table(all$Condition2)
```

```
##
## Artery  Feedr   Norm   PosA   PosN   RRAe   RRAn   RRNn
##      5     13   2889      4      4      1      1      2
```

```
sum(table(all$Condition2))
```

```
## [1] 2919
```

```
all$Street<-as.integer(revalue(all$Street, c('Grvl'=0, 'Pave'=1)))
table(all$Street)
```

```
##
##    0    1
##   12 2907
```

```
sum(table(all$Street))
```

```
## [1] 2919
```

```
all$PavedDrive<-as.integer(revalue(all$PavedDrive, c('N'=0, 'P'=1, 'Y'=2)))
table(all$PavedDrive)
```

```
##
##    0    1    2
##  216   62 2641
```

```
sum(table(all$PavedDrive))
```

```
## [1] 2919
```

```
str(all$YrSold)
```

```
##  int [1:2919] 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
```

```
str(all$MoSold)
```

```
##  int [1:2919] 2 5 9 2 12 10 8 11 4 1 ...
```

```
all$MoSold <- as.factor(all$MoSold)
ys <- ggplot(all[!is.na(all$SalePrice),], aes(x=as.factor(YrSold), y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue')+
        scale_y_continuous(breaks= seq(0, 800000, by=25000), labels = comma) +
```

```
        geom_label(stat = "count", aes(label = ..count.., y = ..count..)) +
        coord_cartesian(ylim = c(0, 200000)) +
        geom_hline(yintercept=163000, linetype="dashed", color = "red") #dashed line is median SalePric
```

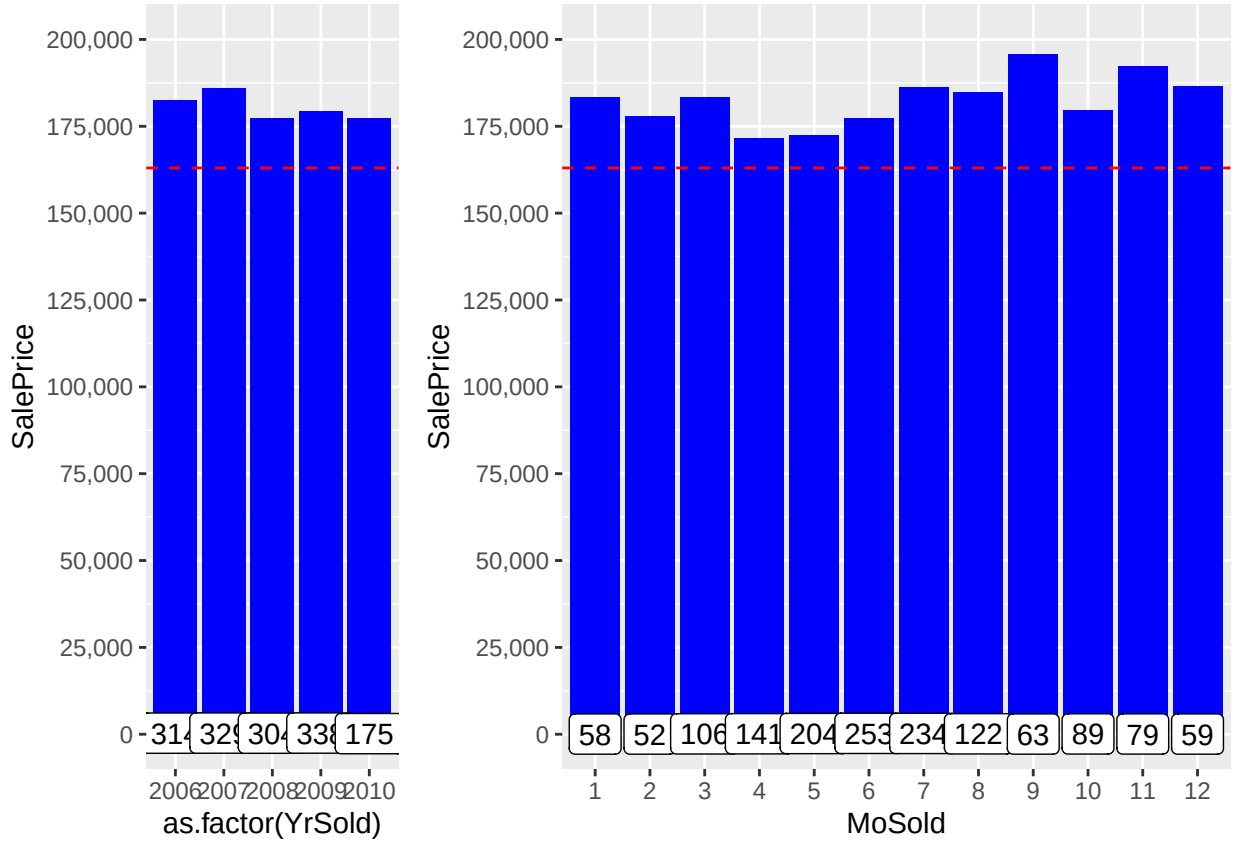
```
## Warning: Ignoring unknown parameters: fun.y
```

```
ms <- ggplot(all[!is.na(all$SalePrice),], aes(x=MoSold, y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue')+
        scale_y_continuous(breaks= seq(0, 800000, by=25000), labels = comma) +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..)) +
        coord_cartesian(ylim = c(0, 200000)) +
        geom_hline(yintercept=163000, linetype="dashed", color = "red") #dashed line is median SalePric
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
grid.arrange(ys, ms, widths=c(1,2))
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```




```
str(all$MSSubClass)
```

```
##  int [1:2919] 60 20 60 70 60 50 20 60 50 190 ...
```

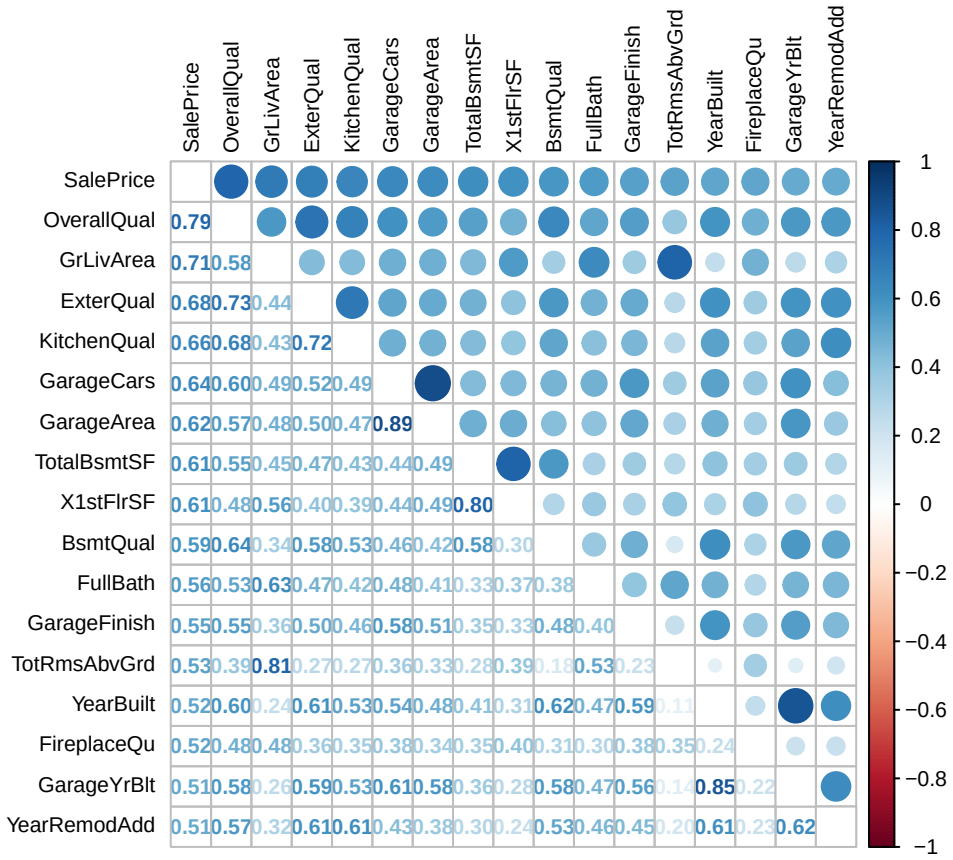
```
all$MSSubClass <- as.factor(all$MSSubClass)
all$MSSubClass<-revalue(all$MSSubClass, c('20'='1 story 1946+', '30'='1 story 1945-', '40'='1 story unf

str(all$MSSubClass)
```

```
##  Factor w/ 16 levels "1 story 1946+",..: 6 1 6 7 6 5 1 6 5 16 ...
```

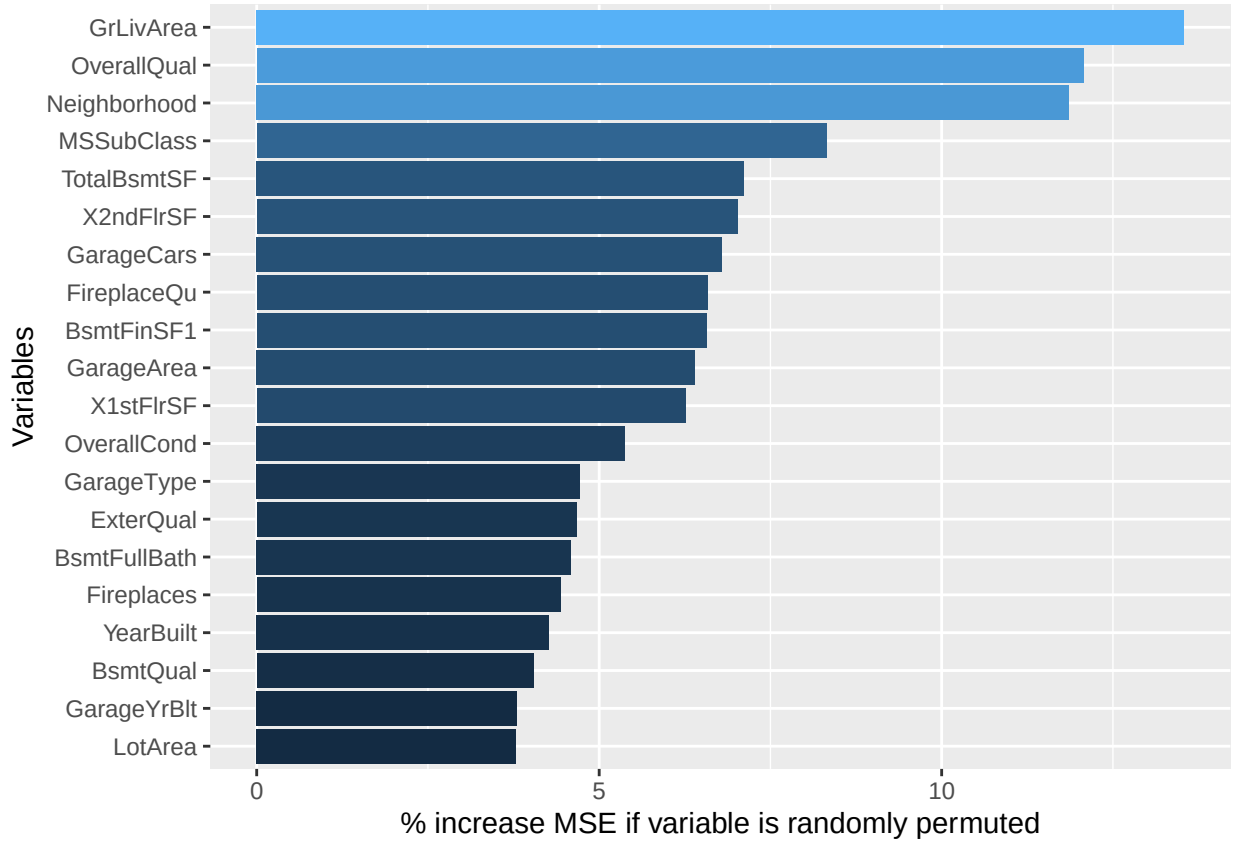
```
numericVars <- which(sapply(all, is.numeric)) #index vector numeric variables
factorVars <- which(sapply(all, is.factor)) #index vector factor variables
cat('There are', length(numericVars), 'numeric variables, and', length(factorVars), 'categoric variables
```

```
## There are 56 numeric variables, and 23 categoric variables
```

```
all_numVar <- all[, numericVars]
cor_numVar <- cor(all_numVar, use="pairwise.complete.obs") #correlations of all numeric variables

#sort on decreasing correlations with SalePrice
cor_sorted <- as.matrix(sort(cor_numVar[,'SalePrice'], decreasing = TRUE))
 #select only high corelations
CorHigh <- names(which(apply(cor_sorted, 1, function(x) abs(x)>0.5)))
cor_numVar <- cor_numVar[CorHigh, CorHigh]

corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt", tl.cex = 0.7,cl.cex = .7, number.cex=.7)
```




```
set.seed(2018)
quick_RF <- randomForest(x=all[1:1460,-79], y=all$SalePrice[1:1460], ntree=100,importance=TRUE)
imp_RF <- importance(quick_RF)
imp_DF <- data.frame(Variables = row.names(imp_RF), MSE = imp_RF[,1])
imp_DF <- imp_DF[order(imp_DF$MSE, decreasing = TRUE),]

ggplot(imp_DF[1:20,], aes(x=reorder(Variables, MSE), y=MSE, fill=MSE)) + geom_bar(stat = 'identity') + l
```

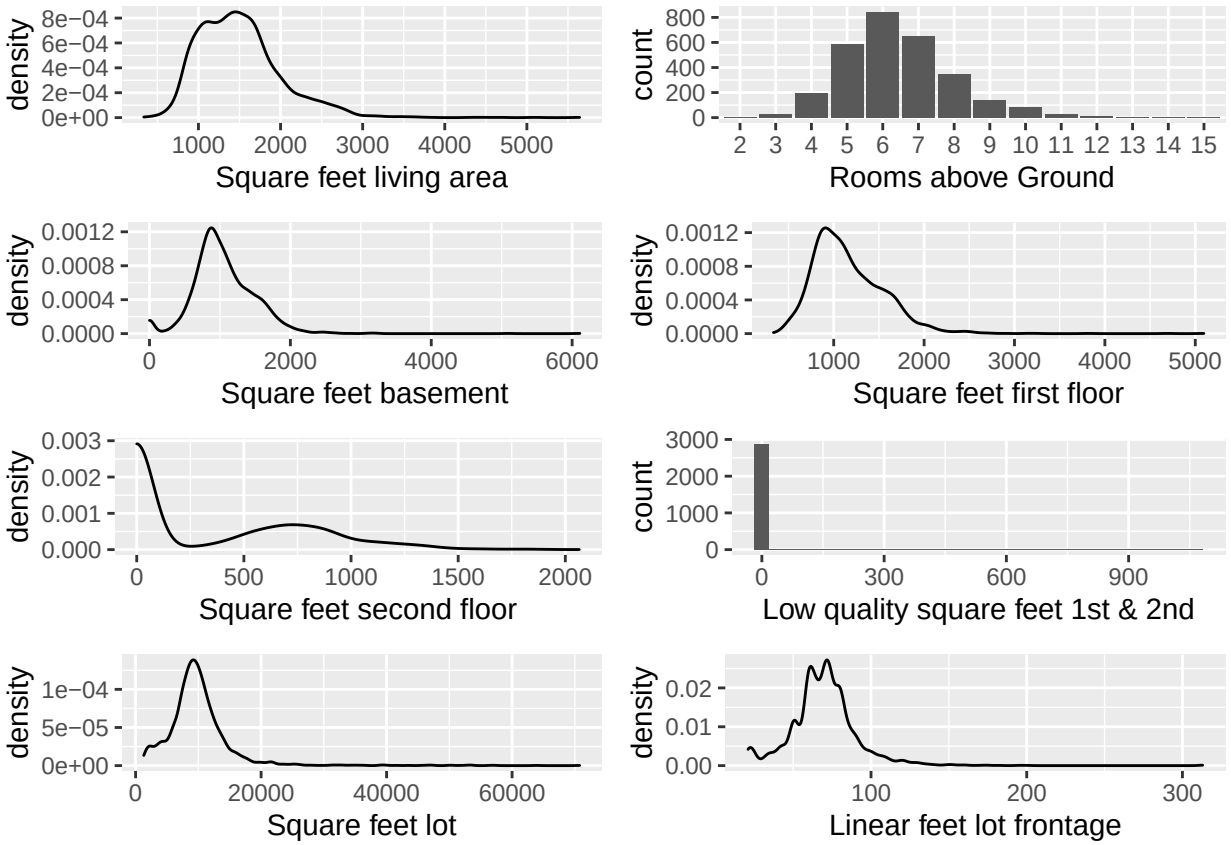



```
s1 <- ggplot(data= all, aes(x=GrLivArea)) +
        geom_density() + labs(x='Square feet living area')
s2 <- ggplot(data=all, aes(x=as.factor(TotRmsAbvGrd))) +
        geom_histogram(stat='count') + labs(x='Rooms above Ground')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
s3 <- ggplot(data= all, aes(x=X1stFlrSF)) +
        geom_density() + labs(x='Square feet first floor')
s4 <- ggplot(data= all, aes(x=X2ndFlrSF)) +
        geom_density() + labs(x='Square feet second floor')
s5 <- ggplot(data= all, aes(x=TotalBsmtSF)) +
        geom_density() + labs(x='Square feet basement')
s6 <- ggplot(data= all[all$LotArea<100000,], aes(x=LotArea)) +
        geom_density() + labs(x='Square feet lot')
s7 <- ggplot(data= all, aes(x=LotFrontage)) +
        geom_density() + labs(x='Linear feet lot frontage')
s8 <- ggplot(data= all, aes(x=LowQualFinSF)) +
        geom_histogram() + labs(x='Low quality square feet 1st & 2nd')

layout <- matrix(c(1,2,5,3,4,8,6,7),4,2,byrow=TRUE)
multiplot(s1, s2, s3, s4, s5, s6, s7, s8, layout=layout)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
cor(all$GrLivArea, (all$X1stFlrSF + all$X2ndFlrSF + all$LowQualFinSF))
```

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

```
head(all[all$LowQualFinSF>0, c('GrLivArea', 'X1stFlrSF', 'X2ndFlrSF', 'LowQualFinSF')])
```

```
##     GrLivArea X1stFlrSF X2ndFlrSF LowQualFinSF
## 52       1176       816         0          360
## 89       1526      1013         0          513
## 126       754       520         0          234
## 171      1382       854         0          528
## 186      3608      1518      1518          572
## 188      1656       808       704          144
```

```
n1 <- ggplot(all[!is.na(all$SalePrice),], aes(x=Neighborhood, y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue') +
        theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3) +
        geom_hline(yintercept=163000, linetype="dashed", color = "red") #dashed line is median SalePric
```

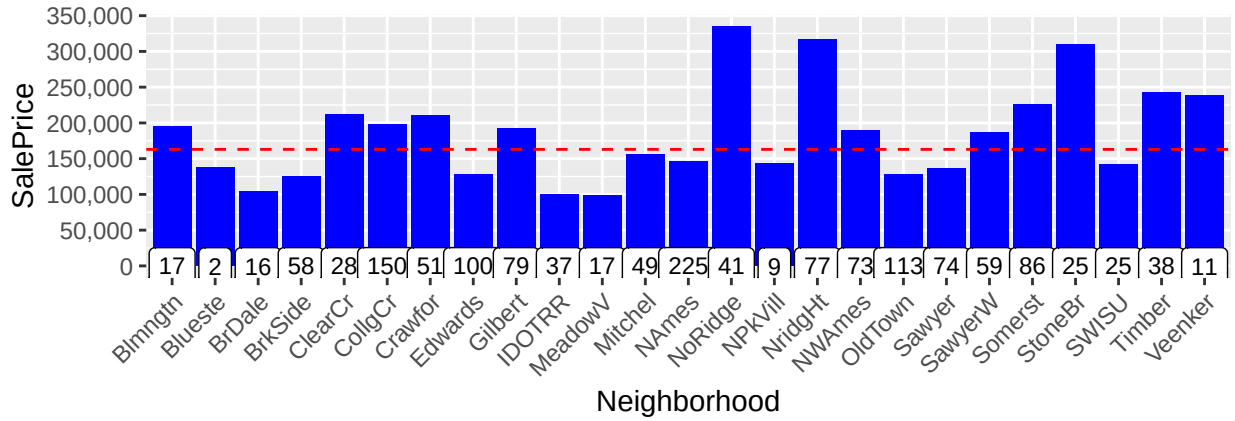
```
## Warning: Ignoring unknown parameters: fun.y
```

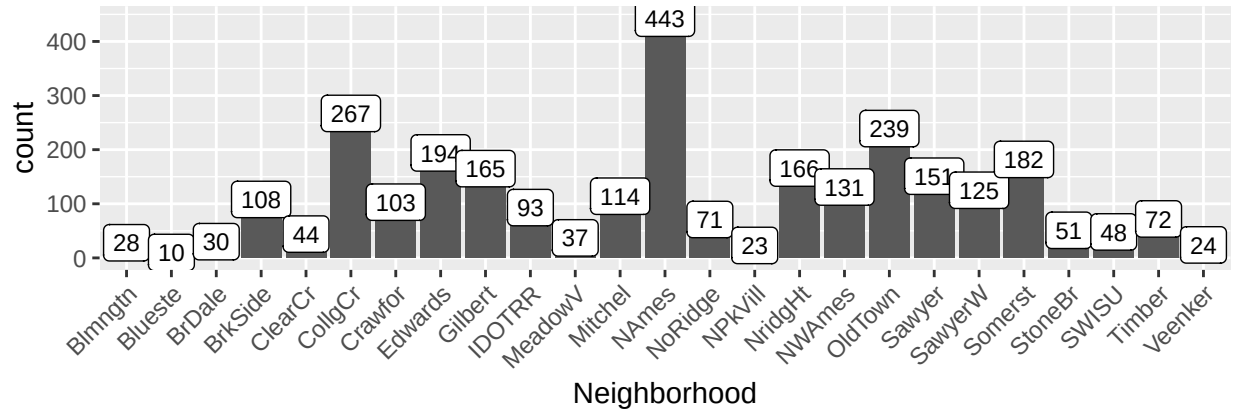
```
n2 <- ggplot(data=all, aes(x=Neighborhood)) +
        geom_histogram(stat='count')+
        geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3)+
        theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
grid.arrange(n1, n2)
```

```
## No summary function supplied, defaulting to `mean_se()`
```






```
q1 <- ggplot(data=all, aes(x=as.factor(OverallQual))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
q2 <- ggplot(data=all, aes(x=as.factor(ExterQual))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
q3 <- ggplot(data=all, aes(x=as.factor(BsmtQual))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
q4 <- ggplot(data=all, aes(x=as.factor(KitchenQual))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
q5 <- ggplot(data=all, aes(x=as.factor(GarageQual))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

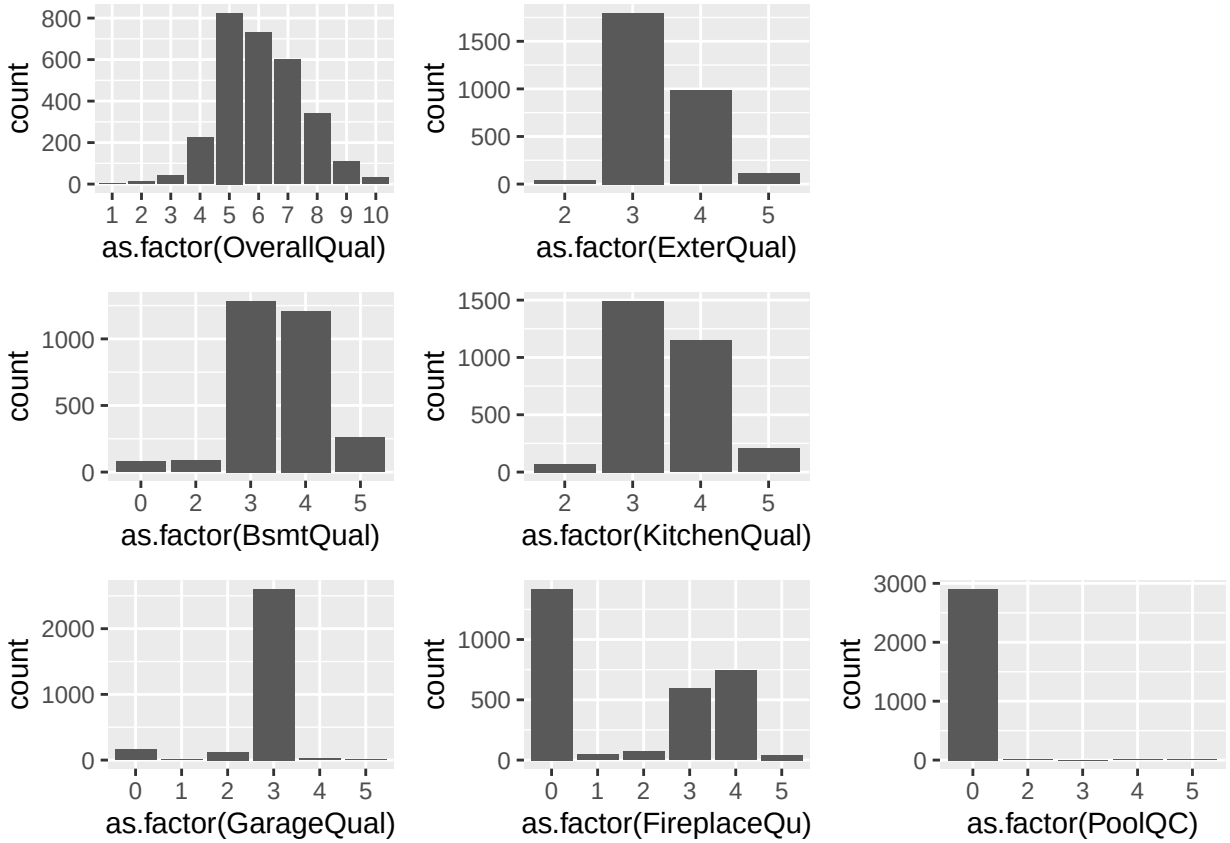
```
q6 <- ggplot(data=all, aes(x=as.factor(FireplaceQu))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
q7 <- ggplot(data=all, aes(x=as.factor(PoolQC))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
layout <- matrix(c(1,2,8,3,4,8,5,6,7),3,3,byrow=TRUE)
multiplot(q1, q2, q3, q4, q5, q6, q7, layout=layout)
```




```
ms1 <- ggplot(all[!is.na(all$SalePrice),], aes(x=MSSubClass, y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue') +
        theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3) +
        geom_hline(yintercept=163000, linetype="dashed", color = "red") #dashed line is median SalePric
```

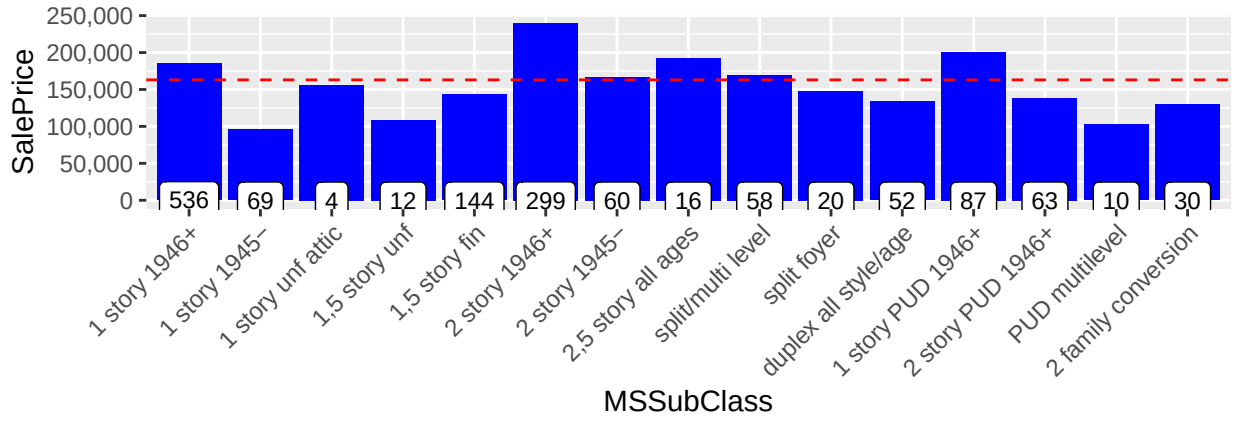
```
## Warning: Ignoring unknown parameters: fun.y
```

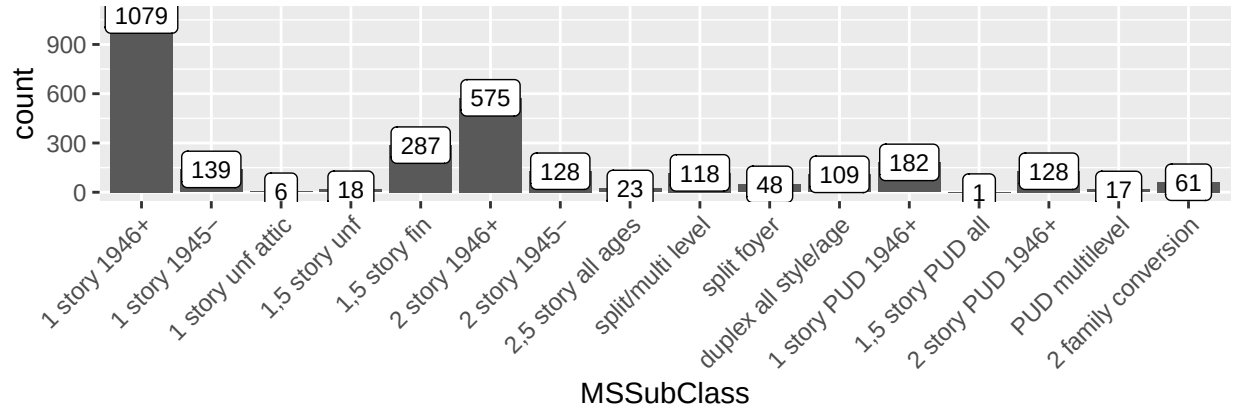
```
ms2 <- ggplot(data=all, aes(x=MSSubClass)) +
        geom_histogram(stat='count')+
        geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3) +
        theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
grid.arrange(ms1, ms2)
```

```
## No summary function supplied, defaulting to `mean_se()`
```






```
all$GarageYrBlt[2593] <- 2007
g1 <- ggplot(data=all[all$GarageCars !=0,], aes(x=GarageYrBlt)) +
        geom_histogram()
g2 <- ggplot(data=all, aes(x=as.factor(GarageCars))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
g3 <- ggplot(data= all, aes(x=GarageArea)) +
        geom_density()
g4 <- ggplot(data=all, aes(x=as.factor(GarageCond))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
g5 <- ggplot(data=all, aes(x=GarageType)) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
g6 <- ggplot(data=all, aes(x=as.factor(GarageQual))) +
        geom_histogram(stat='count')
```

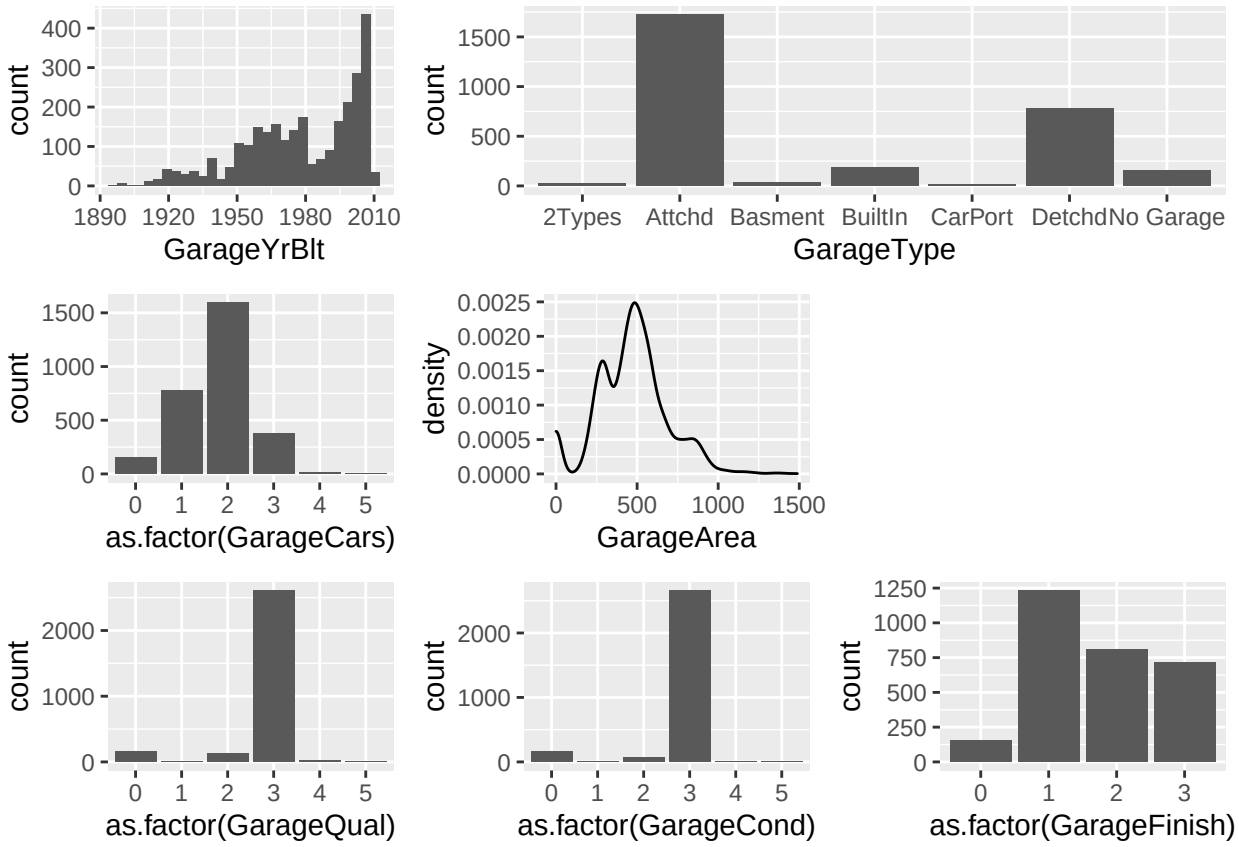
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
g7 <- ggplot(data=all, aes(x=as.factor(GarageFinish))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
layout <- matrix(c(1,5,5,2,3,8,6,4,7),3,3,byrow=TRUE)
multiplot(g1, g2, g3, g4, g5, g6, g7, layout=layout)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
b1 <- ggplot(data=all, aes(x=BsmtFinSF1)) +
        geom_histogram() + labs(x='Type 1 finished square feet')
b2 <- ggplot(data=all, aes(x=BsmtFinSF2)) +
        geom_histogram()+ labs(x='Type 2 finished square feet')
b3 <- ggplot(data=all, aes(x=BsmtUnfSF)) +
        geom_histogram()+ labs(x='Unfinished square feet')
b4 <- ggplot(data=all, aes(x=as.factor(BsmtFinType1))) +
        geom_histogram(stat='count')+ labs(x='Rating of Type 1 finished area')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
b5 <- ggplot(data=all, aes(x=as.factor(BsmtFinType2))) +
        geom_histogram(stat='count')+ labs(x='Rating of Type 2 finished area')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
b6 <- ggplot(data=all, aes(x=as.factor(BsmtQual))) +
        geom_histogram(stat='count')+ labs(x='Height of the basement')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
b7 <- ggplot(data=all, aes(x=as.factor(BsmtCond))) +
        geom_histogram(stat='count')+ labs(x='Rating of general condition')
```

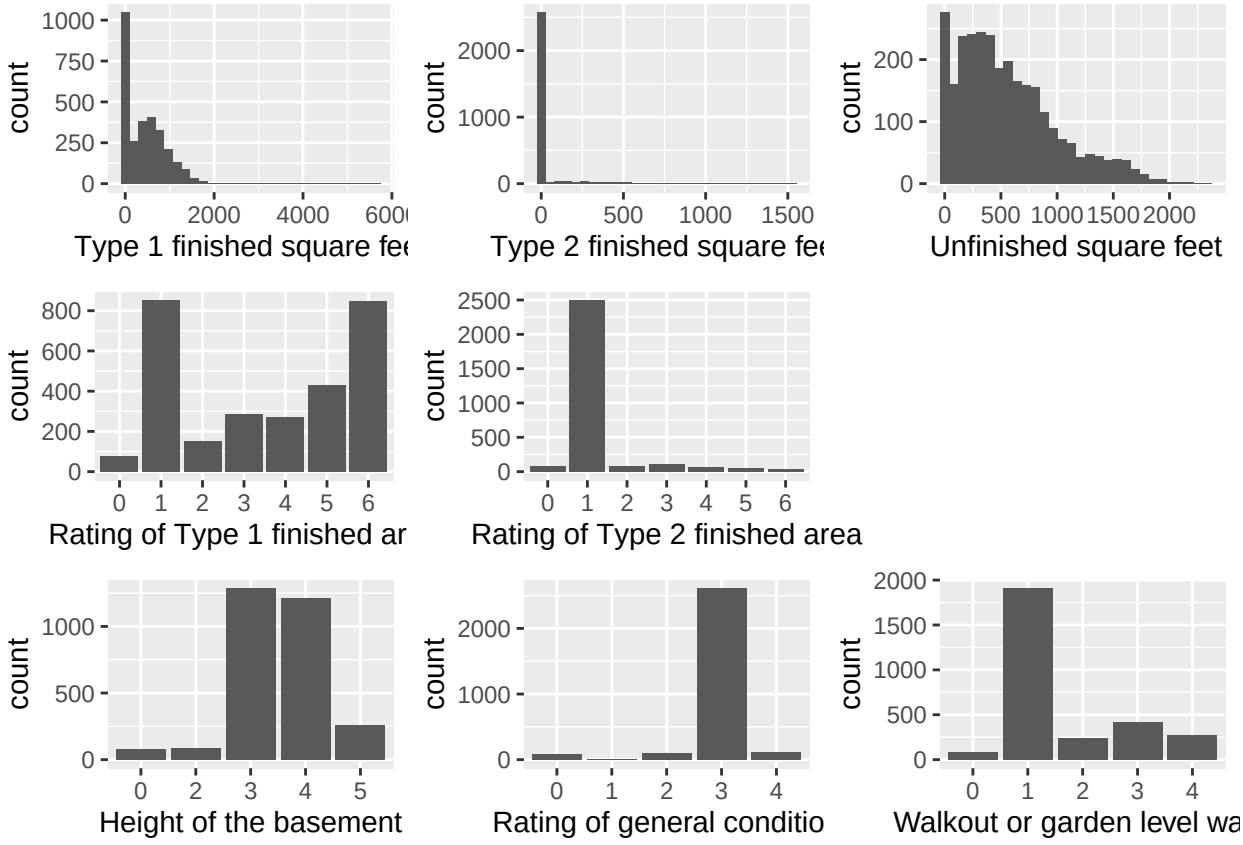
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
b8 <- ggplot(data=all, aes(x=as.factor(BsmtExposure))) +
        geom_histogram(stat='count')+ labs(x='Walkout or garden level walls')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
layout <- matrix(c(1,2,3,4,5,9,6,7,8),3,3,byrow=TRUE)
multiplot(b1, b2, b3, b4, b5, b6, b7, b8, layout=layout)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



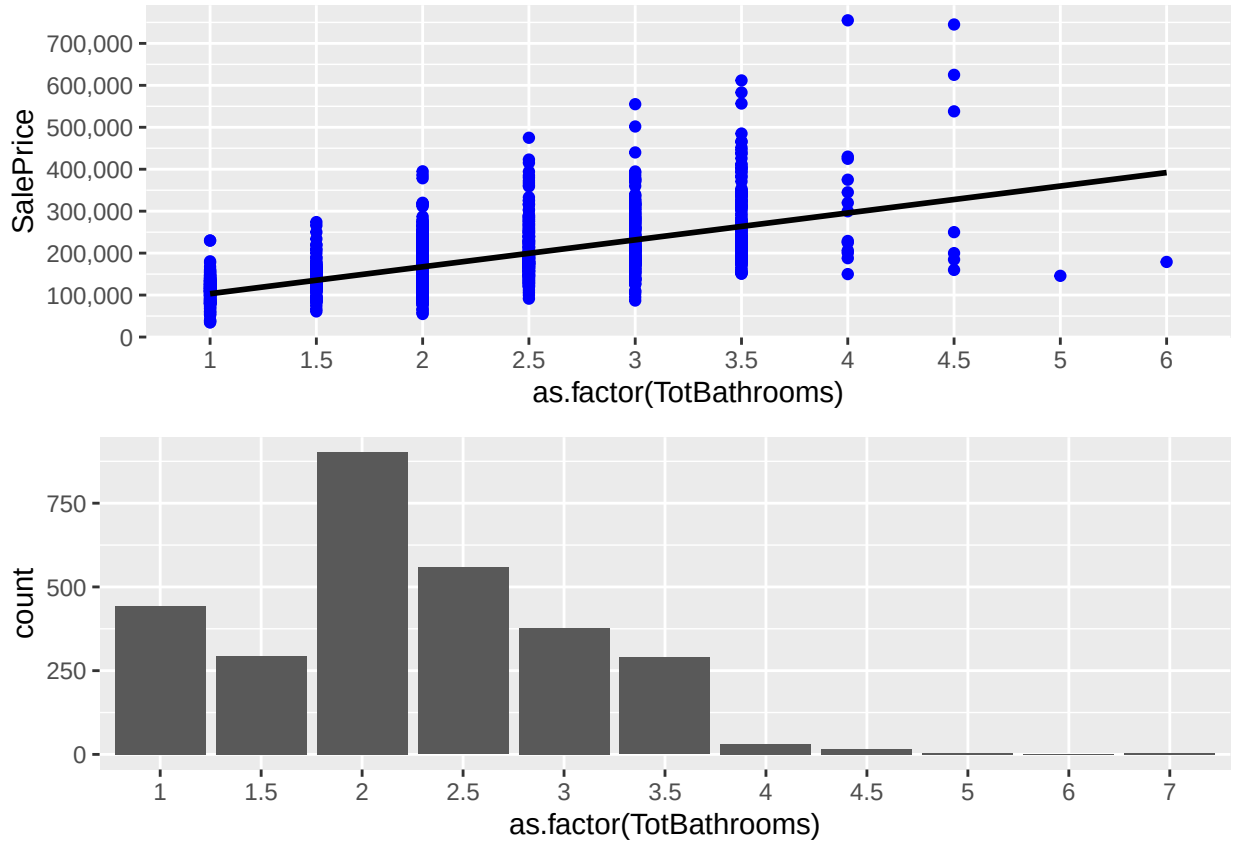

```
all$TotBathrooms <- all$FullBath + (all$HalfBath*0.5) + all$BsmtFullBath + (all$BsmtHalfBath*0.5)
```

```
tb1 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=as.factor(TotBathrooms), y=SalePrice))+
        geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
tb2 <- ggplot(data=all, aes(x=as.factor(TotBathrooms))) +
        geom_histogram(stat='count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

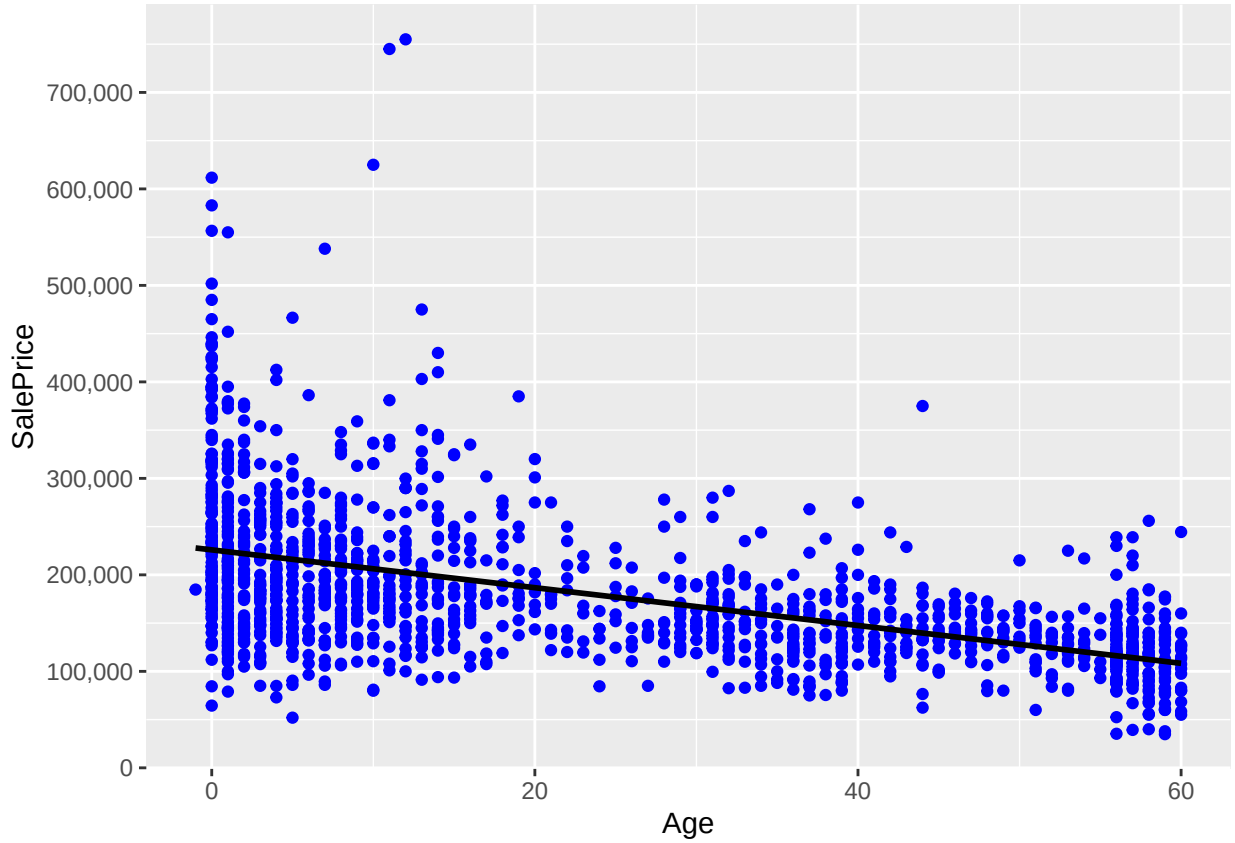
```
grid.arrange(tb1, tb2)
```

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




```
all$Remod <- ifelse(all$YearBuilt==all$YearRemodAdd, 0, 1) #0=No Remodeling, 1=Remodeling
all$Age <- as.numeric(all$YrSold)-all$YearRemodAdd
ggplot(data=all[!is.na(all$SalePrice),], aes(x=Age, y=SalePrice))+
        geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
```

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



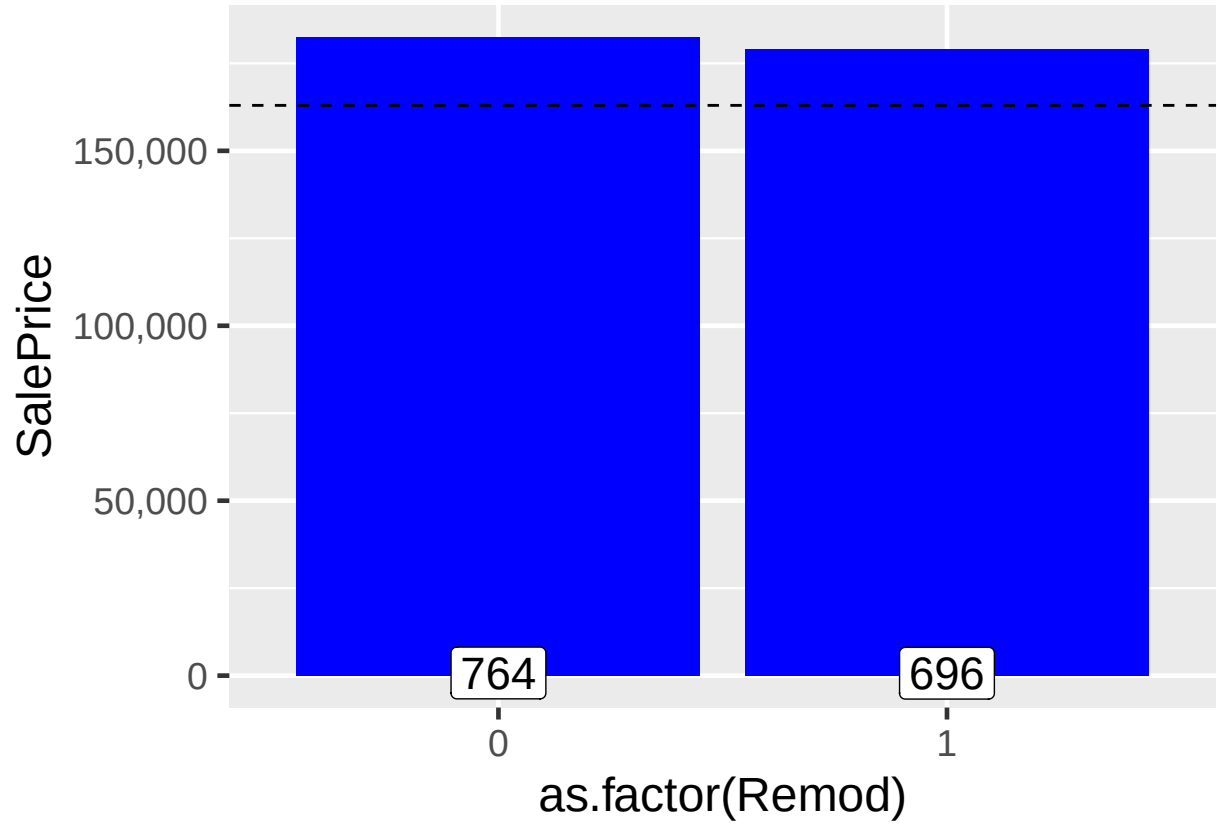

```
cor(all$SalePrice[!is.na(all$SalePrice)], all$Age[!is.na(all$SalePrice)])
```

```
## [1] -0.5090787
```

```
ggplot(all[!is.na(all$SalePrice),], aes(x=as.factor(Remod), y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue') +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=6) +
        scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
        theme_grey(base_size = 18) +
        geom_hline(yintercept=163000, linetype="dashed")
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```



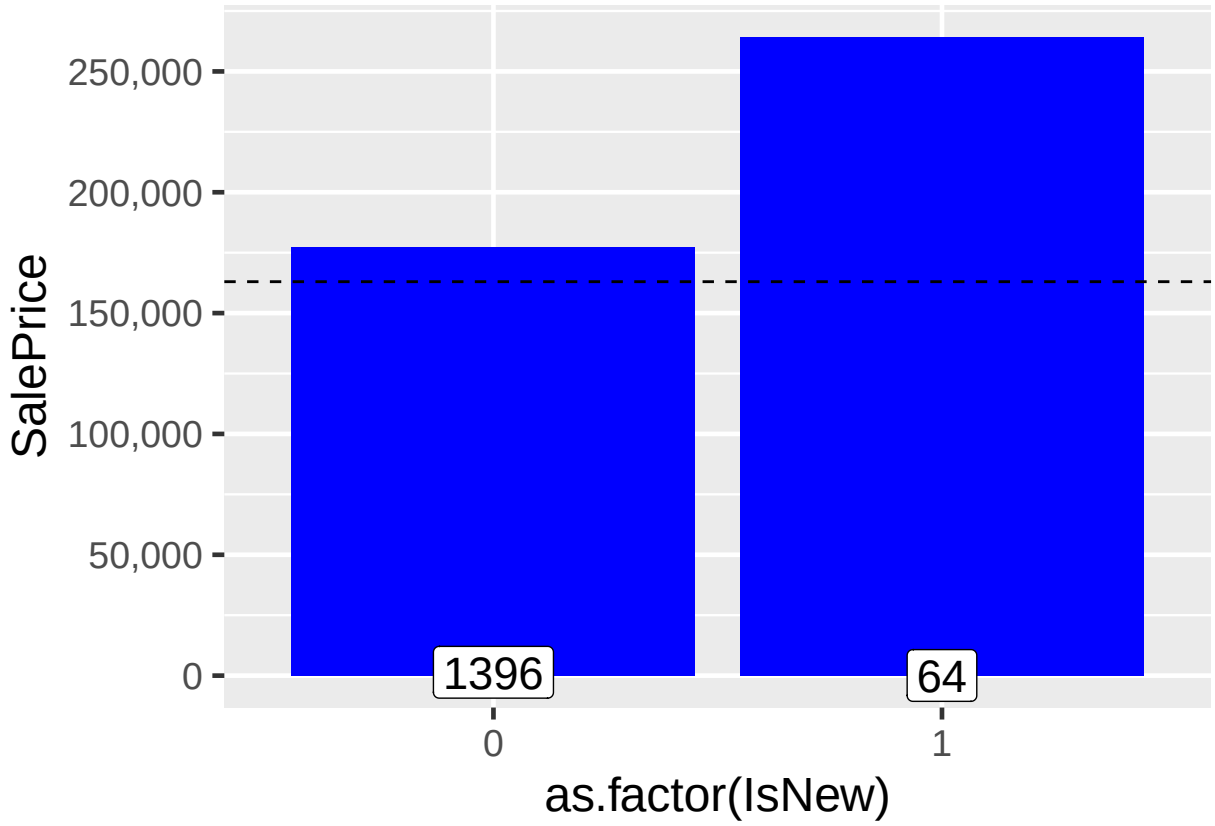

```
all$IsNew <- ifelse(all$YrSold==all$YearBuilt, 1, 0)
table(all$IsNew)
```

```
##
##    0    1
## 2803  116
```

```
ggplot(all[!is.na(all$SalePrice),], aes(x=as.factor(IsNew), y=SalePrice)) +
        geom_bar(stat='summary', fun.y = "median", fill='blue') +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=6) +
        scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
        theme_grey(base_size = 18) +
        geom_hline(yintercept=163000, linetype="dashed")
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
all$YrSold <- as.factor(all$YrSold) #the numeric version is now not needed anymore
```

```
nb1 <- ggplot(all[!is.na(all$SalePrice),], aes(x=reorder(Neighborhood, SalePrice, FUN=median), y=SalePri
        geom_bar(stat='summary', fun.y = "median", fill='blue') + labs(x='Neighborhood', y='Median Salel
        theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3) +
        geom_hline(yintercept=163000, linetype="dashed", color = "red") #dashed line is median SalePric
```

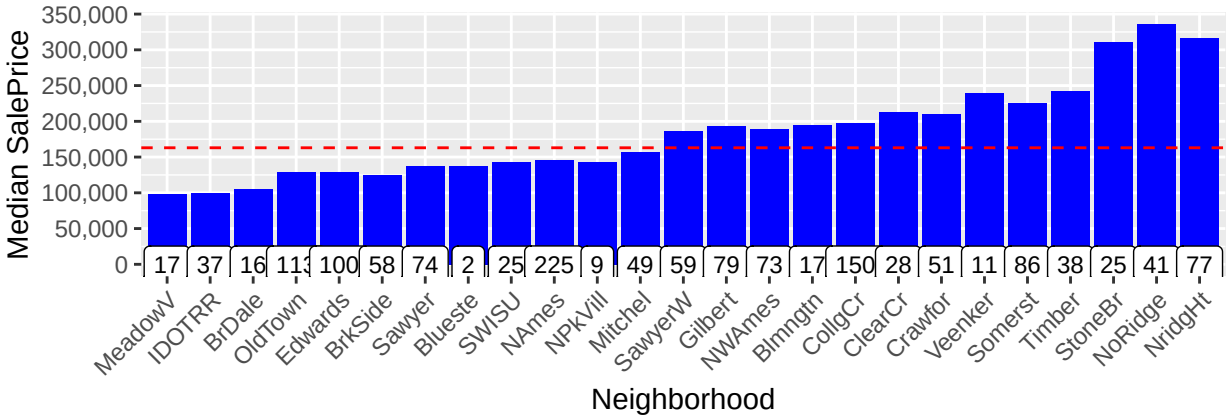
```
## Warning: Ignoring unknown parameters: fun.y
```

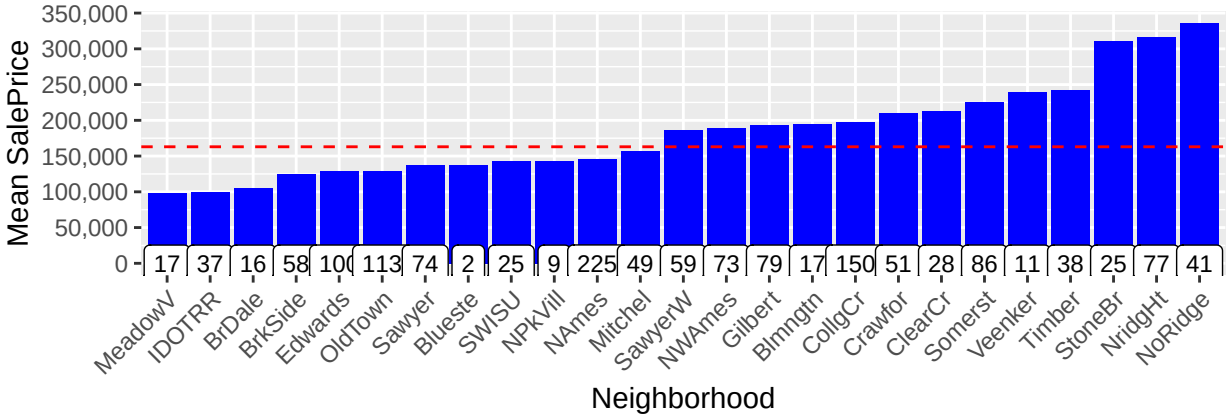
```
nb2 <- ggplot(all[!is.na(all$SalePrice),], aes(x=reorder(Neighborhood, SalePrice, FUN=mean), y=SalePric
        geom_bar(stat='summary', fun.y = "mean", fill='blue') + labs(x='Neighborhood', y="Mean SalePric
        theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
        geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3) +
        geom_hline(yintercept=163000, linetype="dashed", color = "red") #dashed line is median SalePric
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
grid.arrange(nb1, nb2)
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```



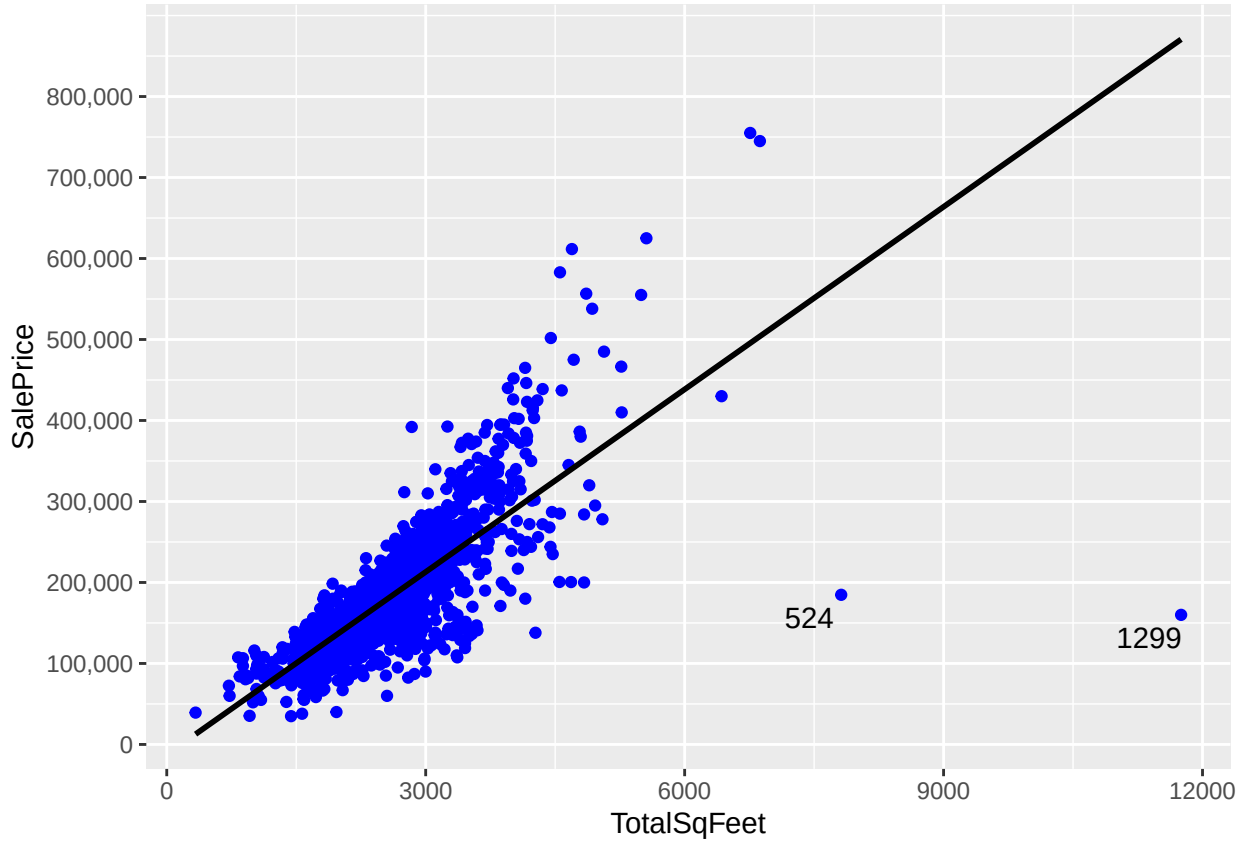



```
all$NeighRich[all$Neighborhood %in% c('StoneBr', 'NridgHt', 'NoRidge')] <- 2
all$NeighRich[!all$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale', 'StoneBr', 'NridgHt', 'NoRidge')]
all$NeighRich[all$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale')] <- 0
table(all$NeighRich)
```

```
##
##    0    1    2
##  160 2471  288
```

```
all$TotalSqFeet <- all$GrLivArea + all$TotalBsmtSF
ggplot(data=all[!is.na(all$SalePrice),], aes(x=TotalSqFeet, y=SalePrice))+
        geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
        geom_text_repel(aes(label = ifelse(all$GrLivArea[!is.na(all$SalePrice)]>4500, rownames(all), ''
```

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




```
cor(all$SalePrice, all$TotalSqFeet, use= "pairwise.complete.obs")
```

```
## [1] 0.7789588
```

```
cor(all$SalePrice[-c(524, 1299)], all$TotalSqFeet[-c(524, 1299)], use= "pairwise.complete.obs")
```

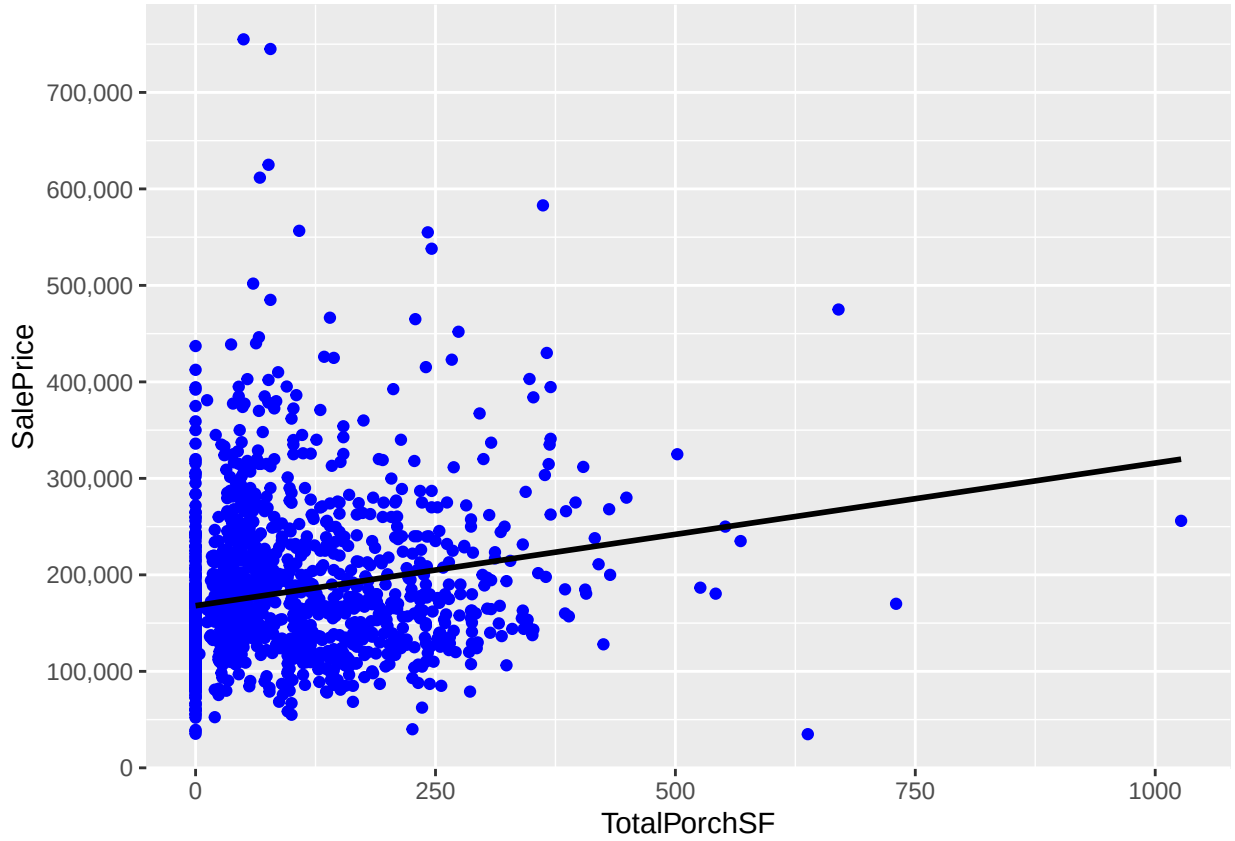
```
## [1] 0.829042
```

```
all$TotalPorchSF <- all$OpenPorchSF + all$EnclosedPorch + all$X3SsnPorch + all$ScreenPorch
cor(all$SalePrice, all$TotalPorchSF, use= "pairwise.complete.obs")
```

```
## [1] 0.1957389
```

```
ggplot(data=all[!is.na(all$SalePrice),], aes(x=TotalPorchSF, y=SalePrice))+
        geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
```

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




```
dropVars <- c('YearRemodAdd', 'GarageYrBlt', 'GarageArea', 'GarageCond', 'TotalBsmtSF', 'TotalRmsAbvGrd
all <- all[,!(names(all) %in% dropVars)]
all <- all[-c(524, 1299),]
numericVarNames <- numericVarNames[!(numericVarNames %in% c('MSSubClass', 'MoSold', 'YrSold', 'SalePric
numericVarNames <- append(numericVarNames, c('Age', 'TotalPorchSF', 'TotBathrooms', 'TotalSqFeet'))

DFnumeric <- all[, names(all) %in% numericVarNames]

DFfactors <- all[, !(names(all) %in% numericVarNames)]
DFfactors <- DFfactors[, names(DFfactors) != 'SalePrice']

cat('There are', length(DFnumeric), 'numeric variables, and', length(DFfactors), 'factor variables')
```

```
## There are 30 numeric variables, and 49 factor variables
```

```
for(i in 1:ncol(DFnumeric)){
        if (abs(skew(DFnumeric[,i]))>0.8){
                DFnumeric[,i] <- log(DFnumeric[,i] +1)
        }
}
PreNum <- preProcess(DFnumeric, method=c("center", "scale"))
print(PreNum)
```

```
## Created from 2917 samples and 30 variables
##
## Pre-processing:
##   - centered (30)
```

```
##   - ignored (0)
##   - scaled (30)
```

```
DFnorm <- predict(PreNum, DFnumeric)
dim(DFnorm)
```

```
## [1] 2917   30
```

```
DFdummies <- as.data.frame(model.matrix(~.-1, DFfactors))
dim(DFdummies)
```

```
## [1] 2917  201
```

```
ZerocolTest <- which(colSums(DFdummies[(nrow(all[!is.na(all$SalePrice),])+1):nrow(all),])==0)
colnames(DFdummies[ZerocolTest])
```

```
##  [1] "Condition2RRAe"     "Condition2RRAn"     "Condition2RRNn"
##  [4] "HouseStyle2.5Fin"   "RoofMatlMembran"    "RoofMatlMetal"
##  [7] "RoofMatlRoll"       "Exterior1stImStucc" "Exterior1stStone"
## [10] "Exterior2ndOther"   "HeatingOthW"        "ElectricalMix"
## [13] "MiscFeatureTenC"
```

```
DFdummies <- DFdummies[,-ZerocolTest]
ZerocolTrain <- which(colSums(DFdummies[1:nrow(all[!is.na(all$SalePrice),]),])==0)
colnames(DFdummies[ZerocolTrain])
```

```
## [1] "MSSubClass1,5 story PUD all"
```

```
DFdummies <- DFdummies[,-ZerocolTrain]
fewOnes <- which(colSums(DFdummies[1:nrow(all[!is.na(all$SalePrice),]),])<10)
colnames(DFdummies[fewOnes])
```

```
##  [1] "MSSubClass1 story unf attic" "LotConfigFR3"
##  [3] "NeighborhoodBlueste"         "NeighborhoodNPkVill"
##  [5] "Condition1PosA"              "Condition1RRNe"
##  [7] "Condition1RRNn"              "Condition2Feedr"
##  [9] "Condition2PosA"              "Condition2PosN"
## [11] "RoofStyleMansard"            "RoofStyleShed"
## [13] "RoofMatlWdShake"             "RoofMatlWdShngl"
## [15] "Exterior1stAsphShn"          "Exterior1stBrkComm"
## [17] "Exterior1stCBlock"           "Exterior2ndAsphShn"
## [19] "Exterior2ndBrk Cmn"          "Exterior2ndCBlock"
## [21] "Exterior2ndStone"            "FoundationStone"
## [23] "FoundationWood"              "HeatingGrav"
## [25] "HeatingWall"                 "ElectricalFuseP"
## [27] "GarageTypeCarPort"           "MiscFeatureOthr"
## [29] "SaleTypeCon"                 "SaleTypeConLD"
## [31] "SaleTypeConLI"               "SaleTypeConLw"
## [33] "SaleTypeCWD"                 "SaleTypeOth"
## [35] "SaleConditionAdjLand"
```

```
DFdummies <- DFdummies[,-fewOnes] #removing predictors
dim(DFdummies)
```

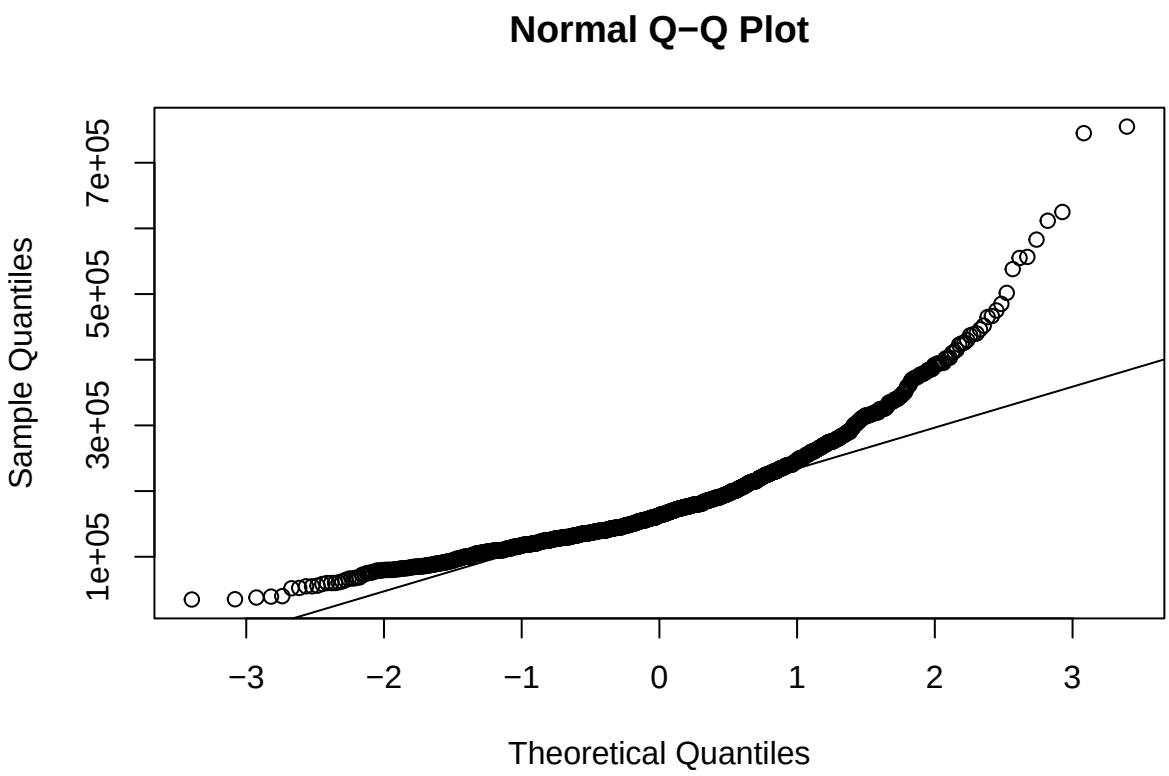
```
## [1] 2917  152
```

```
combined <- cbind(DFnorm, DFdummies) #combining all (now numeric) predictors into one dataframe
```

```
skew(all$SalePrice)
```

```
## [1] 1.877427
```

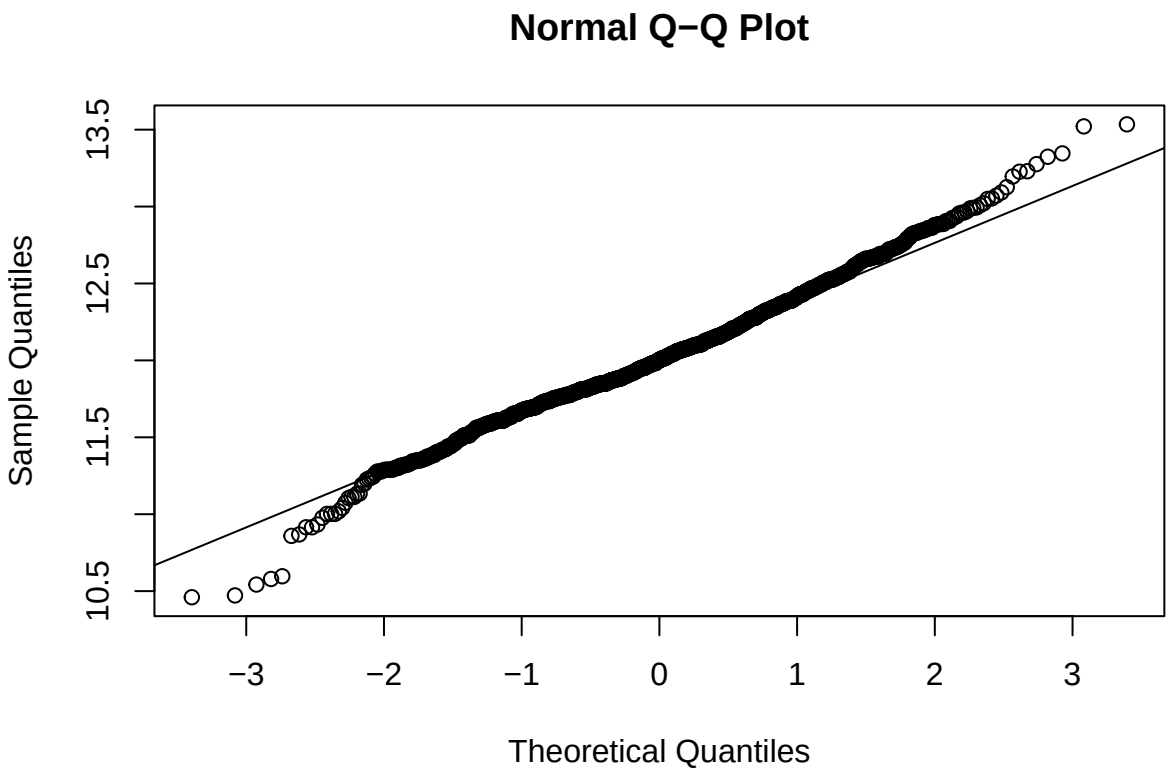
```
qqnorm(all$SalePrice)
qqline(all$SalePrice)
```




```
all$SalePrice <- log(all$SalePrice) #default is the natural logarithm, "+1" is not necessary as there a
skew(all$SalePrice)
```

```
## [1] 0.1213182
```

```
qqnorm(all$SalePrice)
qqline(all$SalePrice)
```




```
train1 <- combined[!is.na(all$SalePrice),]
test1 <- combined[is.na(all$SalePrice),]
```

```
set.seed(06102022)
my_control <-trainControl(method="cv", number=5)
lassoGrid <- expand.grid(alpha = 1, lambda = seq(0.001,0.1,by = 0.0005))

lasso_mod <- train(x=train1, y=all$SalePrice[!is.na(all$SalePrice)], method='glmnet', trControl= my_con
lasso_mod$bestTune
```

```
##   alpha lambda
## 6     1 0.0035
```

```
min(lasso_mod$results$RMSE)
```

```
## [1] 0.1146827
```

```
lassoVarImp <- varImp(lasso_mod,scale=F)
lassoImportance <- lassoVarImp$importance

varsSelected <- length(which(lassoImportance$Overall!=0))
varsNotSelected <- length(which(lassoImportance$Overall==0))

cat('Lasso uses', varsSelected, 'variables in its model, and did not select', varsNotSelected, 'variabl
```

```
## Lasso uses 77 variables in its model, and did not select 105 variables.
```

```
LassoPred <- predict(lasso_mod, test1)
predictions_lasso <- exp(LassoPred) #need to reverse the log to the real values
```

```
head(predictions_lasso)
```

```
##     1461     1462     1463     1464     1465     1466
## 113873.3 161028.9 178977.1 197311.5 205218.1 170421.9
```

```
xgb_grid = expand.grid(
nrounds = 1000,
eta = c(0.1, 0.05, 0.01),
max_depth = c(2, 3, 4, 5, 6),
gamma = 0,
colsample_bytree=1,
min_child_weight=c(1, 2, 3, 4 ,5),
subsample=1
)
label_train <- all$SalePrice[!is.na(all$SalePrice)]

# put our testing & training data into two seperates Dmatrixs objects
dtrain <- xgb.DMatrix(data = as.matrix(train1), label= label_train)
dtest <- xgb.DMatrix(data = as.matrix(test1))
```

```
default_param<-list(
        objective = "reg:linear",
        booster = "gbtree",
        eta=0.05, #default = 0.3
        gamma=0,
        max_depth=3, #default=6
        min_child_weight=4, #default=1
        subsample=1,
        colsample_bytree=1
)
xgbcv <- xgb.cv( params = default_param, data = dtrain, nrounds = 500, nfold = 5, showsd = T, stratifie
```

```
## [11:09:38] WARNING: amalgamation/../src/objective/regression_obj.cu:203: reg:linear is now deprecate
## [11:09:38] WARNING: amalgamation/../src/objective/regression_obj.cu:203: reg:linear is now deprecate
## [11:09:38] WARNING: amalgamation/../src/objective/regression_obj.cu:203: reg:linear is now deprecate
## [11:09:38] WARNING: amalgamation/../src/objective/regression_obj.cu:203: reg:linear is now deprecate
## [11:09:38] WARNING: amalgamation/../src/objective/regression_obj.cu:203: reg:linear is now deprecate
## [1]  train-rmse:10.955589+0.006285   test-rmse:10.955562+0.026742
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [41] train-rmse:1.428250+0.000862    test-rmse:1.428528+0.011939
## [81] train-rmse:0.219673+0.000573    test-rmse:0.231201+0.007559
## [121]    train-rmse:0.102169+0.000732    test-rmse:0.129382+0.009652
## [161]    train-rmse:0.090049+0.000644    test-rmse:0.122665+0.008848
## [201]    train-rmse:0.084101+0.000608    test-rmse:0.120433+0.008246
## [241]    train-rmse:0.079497+0.000599    test-rmse:0.119146+0.007892
## [281]    train-rmse:0.075714+0.000642    test-rmse:0.118201+0.007580
## [321]    train-rmse:0.072556+0.000727    test-rmse:0.117622+0.007366
## [361]    train-rmse:0.069737+0.000760    test-rmse:0.117263+0.007178
## Stopping. Best iteration:
## [364]    train-rmse:0.069536+0.000730    test-rmse:0.117207+0.007163
```

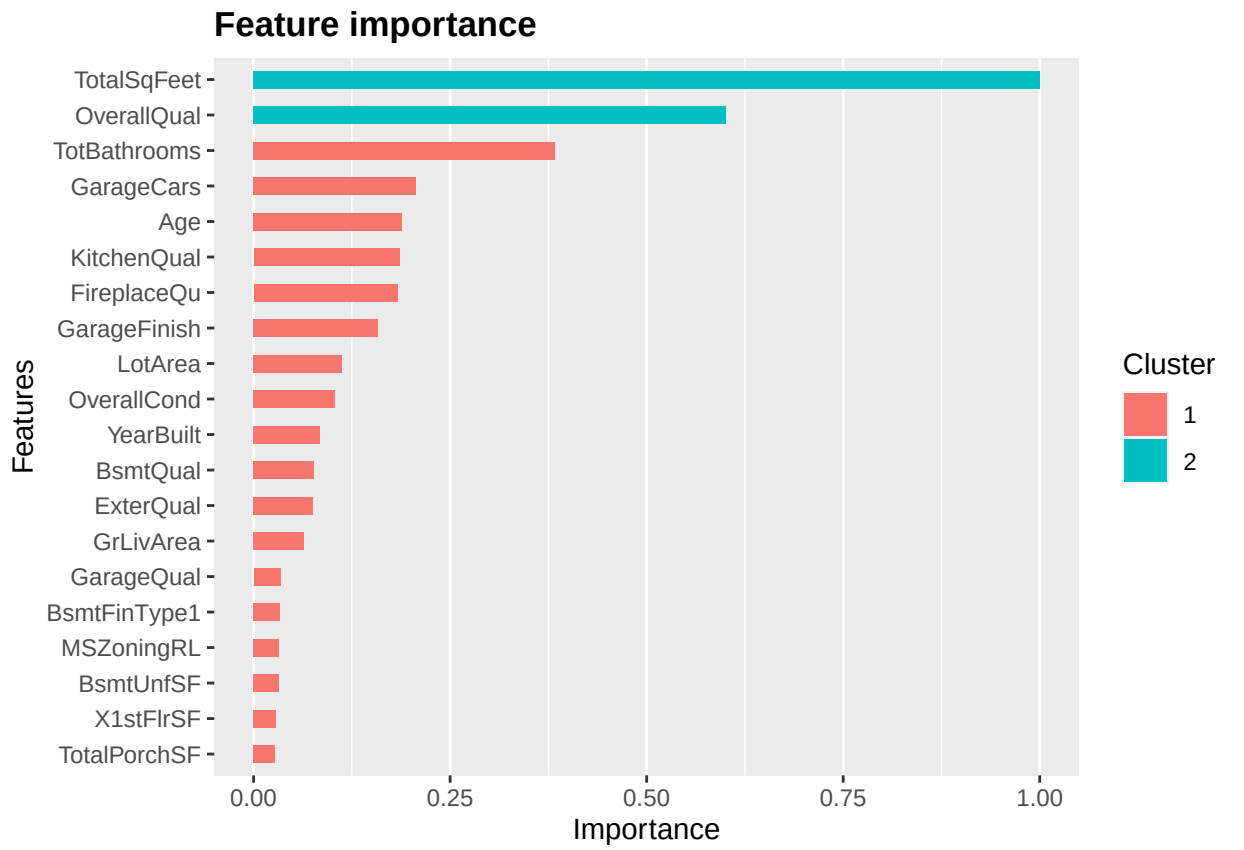
```
xgb_mod <- xgb.train(data = dtrain, params=default_param, nrounds = 454)
```

```
## [11:09:42] WARNING: amalgamation/../src/objective/regression_obj.cu:203: reg:linear is now deprecated
```

```
XGBpred <- predict(xgb_mod, dtest)
predictions_XGB <- exp(XGBpred) #need to reverse the log to the real values
head(predictions_XGB)
```

```
## [1] 116387.0 162307.1 186493.8 187440.4 187258.2 166241.1
```

```
library(Ckmeans.1d.dp) #required for ggplot clustering
mat <- xgb.importance (feature_names = colnames(train1),model = xgb_mod)
xgb.ggplot.importance(importance_matrix = mat[1:20], rel_to_first = TRUE)
```




```
sub_avg <- data.frame(Id = test_labels, SalePrice = (predictions_XGB+2*predictions_lasso)/3)
head(sub_avg)
```

```
##        Id SalePrice
## 1461 1461  114711.2
## 1462 1462  161455.0
## 1463 1463  181482.7
## 1464 1464  194021.1
## 1465 1465  199231.5
## 1466 1466  169028.3
```

```
write.csv(sub_avg, file = 'average.csv', row.names = F)
```