testtest

2025-05-23

```
library(ggplot2)
library(readr)

df_train <- read_csv("~/Downloads/house-prices-advanced-regression-techniques/train.c
sv")
```

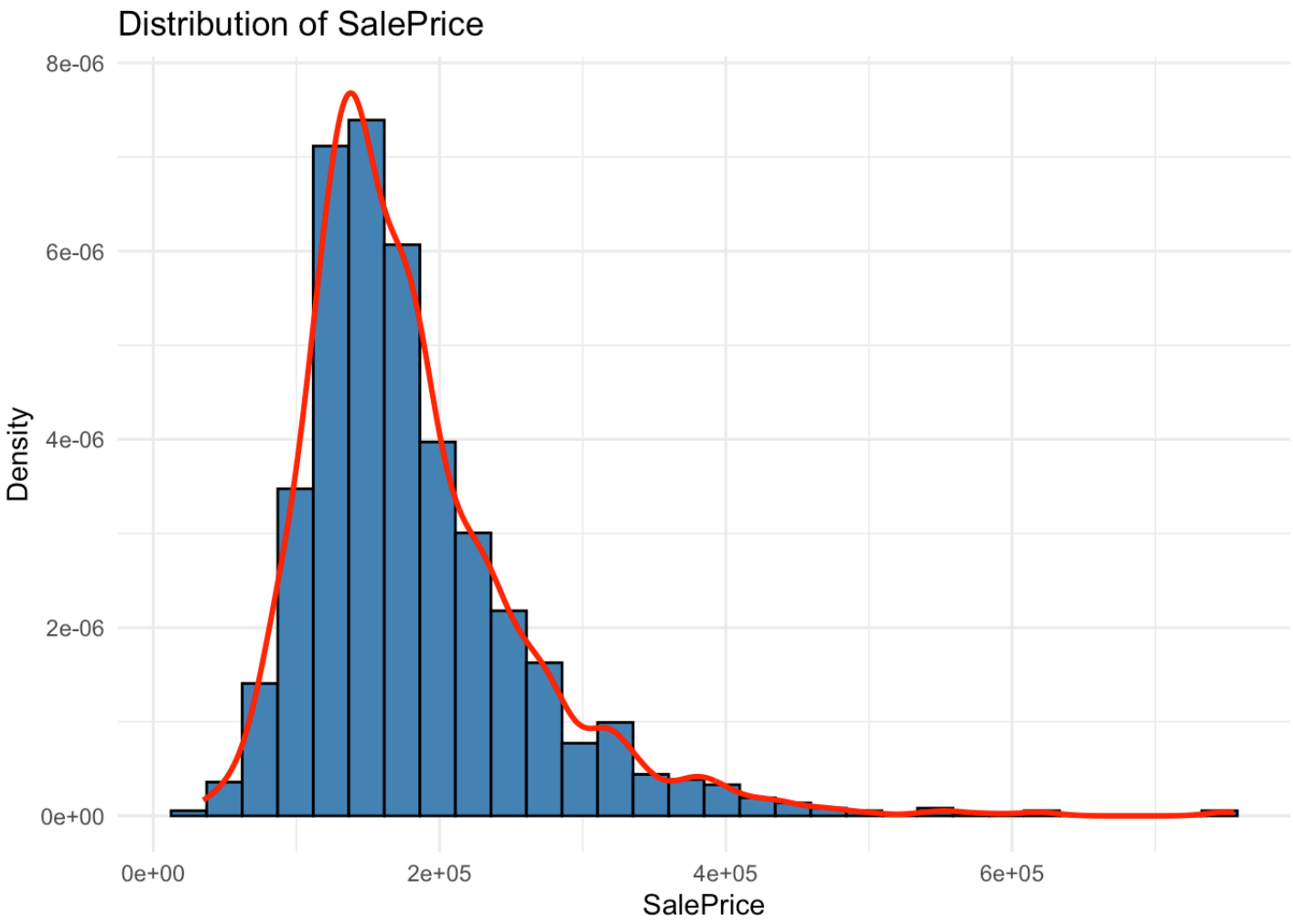
```
## Rows: 1460 Columns: 81
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (43): MSZoning, Street, Alley, LotShape, LandContour, Utilities, LotConf...
## dbl (38): Id, MSSubClass, LotFrontage, LotArea, OverallQual, OverallCond, Ye...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
print(names(df_train))
```

```
##  [1] "Id"            "MSSubClass"    "MSZoning"      "LotFrontage"
##  [5] "LotArea"       "Street"        "Alley"         "LotShape"
##  [9] "LandContour"   "Utilities"     "LotConfig"     "LandSlope"
## [13] "Neighborhood"  "Condition1"    "Condition2"    "BldgType"
## [17] "HouseStyle"    "OverallQual"   "OverallCond"   "YearBuilt"
## [21] "YearRemodAdd"  "RoofStyle"     "RoofMatl"      "Exterior1st"
## [25] "Exterior2nd"   "MasVnrType"    "MasVnrArea"    "ExterQual"
## [29] "ExterCond"     "Foundation"    "BsmtQual"      "BsmtCond"
## [33] "BsmtExposure"  "BsmtFinType1"  "BsmtFinSF1"    "BsmtFinType2"
## [37] "BsmtFinSF2"    "BsmtUnfSF"     "TotalBsmtSF"   "Heating"
## [41] "HeatingQC"     "CentralAir"    "Electrical"    "1stFlrSF"
## [45] "2ndFlrSF"      "LowQualFinSF"  "GrLivArea"     "BsmtFullBath"
## [49] "BsmtHalfBath"  "FullBath"      "HalfBath"      "BedroomAbvGr"
## [53] "KitchenAbvGr"  "KitchenQual"   "TotRmsAbvGrd"  "Functional"
## [57] "Fireplaces"    "FireplaceQu"   "GarageType"    "GarageYrBlt"
## [61] "GarageFinish"  "GarageCars"    "GarageArea"    "GarageQual"
## [65] "GarageCond"    "PavedDrive"    "WoodDeckSF"    "OpenPorchSF"
## [69] "EnclosedPorch" "3SsnPorch"     "ScreenPorch"   "PoolArea"
## [73] "PoolQC"        "Fence"         "MiscFeature"   "MiscVal"
## [77] "MoSold"        "YrSold"        "SaleType"      "SaleCondition"
## [81] "SalePrice"
```

```
ggplot(df_train, aes(x = SalePrice)) +
  geom_histogram(aes(y = after_stat(density)), fill = "steelblue", bins = 30, color =
"black") +
  geom_density(color = "red", size = 1) +
  labs(title = "Distribution of SalePrice", x = "SalePrice", y = "Density") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
library(readr)
library(ranger)
```

```
## Warning: package 'ranger' was built under R version 4.3.3
```

```
dataset_df <- read_csv("~/Downloads/house-prices-advanced-regression-techniques/trai
n.csv")
```

```
## Rows: 1460 Columns: 81
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (43): MSZoning, Street, Alley, LotShape, LandContour, Utilities, LotConf...
## dbl (38): Id, MSSubClass, LotFrontage, LotArea, OverallQual, OverallCond, Ye...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
colnames(dataset_df) <- make.names(colnames(dataset_df))

if ("Id" %in% names(dataset_df)) {
  dataset_df$Id <- NULL
}

for (col in names(dataset_df)) {
  if (is.character(dataset_df[[col]])) {
    dataset_df[[col]] <- as.factor(dataset_df[[col]])
  }
}

sale_price_col <- dataset_df$SalePrice
features_only <- dataset_df[, setdiff(names(dataset_df), "SalePrice")]

features_filtered <- features_only[, sapply(features_only, function(x) {
  if (is.factor(x)) {
    n <- nlevels(x)
    return(n > 1 && n <= 50)
  } else if (is.numeric(x)) {
    return(length(unique(x[!is.na(x)])) > 1)
  }
  return(FALSE)
})]

dataset_df <- cbind(features_filtered, SalePrice = sale_price_col)

for (col in names(dataset_df)) {
  if (is.numeric(dataset_df[[col]]) && any(is.na(dataset_df[[col]]))) {
    dataset_df[[col]][is.na(dataset_df[[col]])] <- median(dataset_df[[col]], na.rm =
TRUE)
  }
}
```

```
for (col in names(dataset_df)) {
  if (is.factor(dataset_df[[col]]) && any(is.na(dataset_df[[col]]))) {
    dataset_df[[col]] <- addNA(dataset_df[[col]])
    levels(dataset_df[[col]])[is.na(levels(dataset_df[[col]]))] <- "Missing"
  }
}

set.seed(123)
split_index <- runif(nrow(dataset_df)) >= 0.3
train_ds <- dataset_df[split_index, ]
valid_ds <- dataset_df[!split_index, ]

cat(nrow(train_ds), "training samples,", nrow(valid_ds), "validation samples\n")
```

```
## 1021 training samples, 439 validation samples
```

```
rf_model <- ranger(
  SalePrice ~ .,
  data = train_ds,
  num.trees = 300,
  mtry = floor(sqrt(ncol(train_ds) - 1)),
  importance = "impurity",
  seed = 123
)

rf_preds <- predict(rf_model, data = valid_ds)$predictions

mse <- mean((rf_preds - valid_ds$SalePrice)^2)
cat("Validation MSE:", round(mse, 2), "\n")
```

```
## Validation MSE: 1039602215
```

```
importance <- sort(rf_model$variable.importance, decreasing = TRUE)
cat("Top 10 important variables:\n")
```

```
## Top 10 important variables:
```

```
print(head(importance, 10))
```

```
##  OverallQual    GrLivArea   GarageArea   GarageCars  TotalBsmtSF    YearBuilt
## 564196005124 495603962167 383659651480 362228346411 279305181357 272251195577
##    ExterQual    X1stFlrSF     BsmtQual  KitchenQual
## 260549887231 257805096425 252811985865 229486679359
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.3
```

```
## randomForest 4.7-1.2
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ranger':
##
##     importance
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(randomForestExplainer)
```

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

```
library(dplyr)
```

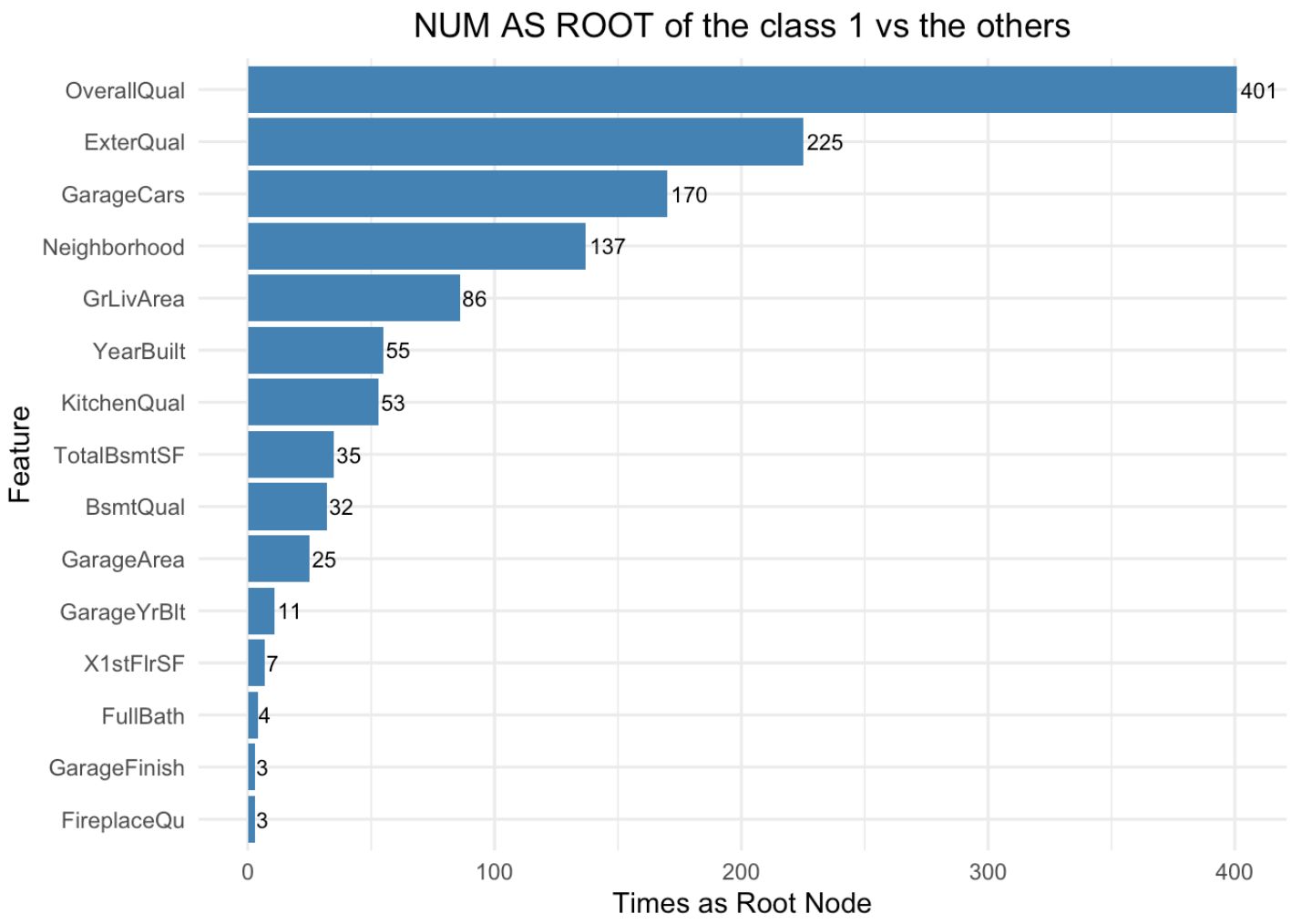
```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:randomForest':
##
##     combine
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)

df <- read.csv("~/Downloads/house-prices-advanced-regression-techniques/train.csv")
df$Id <- NULL

df <- df %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), median(., na.rm = TRUE), .))) %
>%
  mutate(across(where(is.character), ~ ifelse(is.na(.), "Missing", .))) %>%
  mutate(across(where(is.character), as.factor))

set.seed(123)
rf_model <- randomForest(SalePrice ~ ., data = df, ntree = 1250, importance = TRUE)

importance_df <- measure_importance(rf_model)

variable_importance_metric <- "times_a_root"
feature_names <- importance_df$variable
feature_importances <- importance_df[[variable_importance_metric]]

feature_ranks <- order(feature_importances, decreasing = TRUE)
ranked_df <- data.frame(
  Feature = feature_names[feature_ranks],
  Importance = feature_importances[feature_ranks]
)

ggplot(ranked_df[1:15, ], aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = round(Importance, 0)), hjust = -0.1, size = 3) +
  coord_flip() +
  labs(
    title = "NUM AS ROOT of the class 1 vs the others",
    x = "Feature",
    y = "Times as Root Node"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

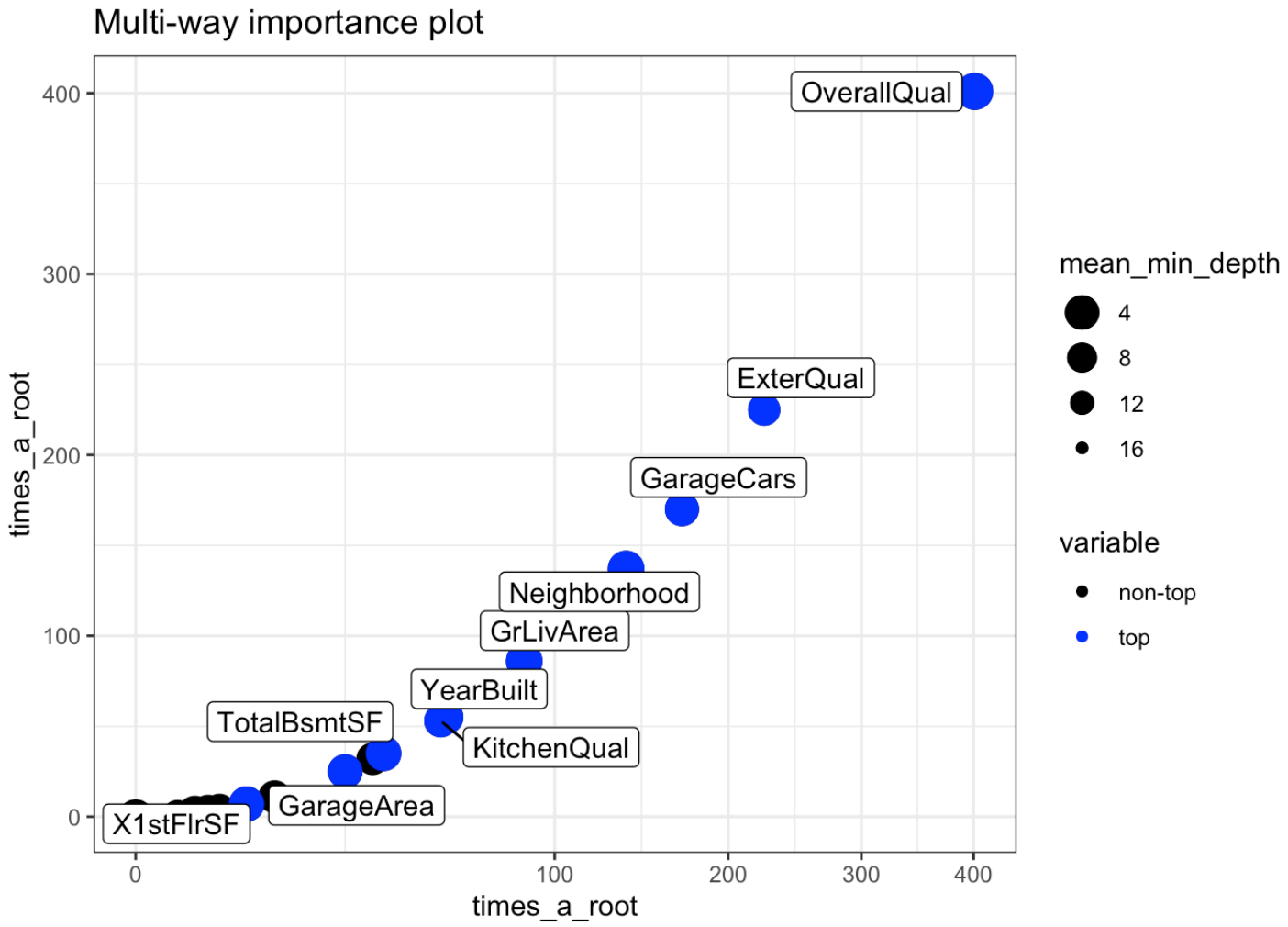


#This chart measures how much each feature reduces impurity (e.g., variance for regression) across all splits it contributes to in the forest. Since OverallQual reduces prediction error the most, it's the most powerful predictor in terms of explaining variance in SalePrice.

```
colnames(importance_df)
```

```
## [1] "variable"            "mean_min_depth"      "no_of_nodes"
## [4] "mse_increase"        "node_purity_increase" "no_of_trees"
## [7] "times_a_root"        "p_value"
```

```
library(randomForest)
library(randomForestExplainer)
library(dplyr)

df <- read.csv("~/Downloads/house-prices-advanced-regression-techniques/train.csv")
df$Id <- NULL

df <- df %>% mutate(across(where(is.numeric), ~ ifelse(is.na(.), median(., na.rm = TR
UE), .)))

df <- df %>% mutate(across(where(is.character), ~ ifelse(is.na(.), "Missing", .)))
df <- df %>% mutate(across(where(is.character), as.factor))

set.seed(123)
rf_model <- randomForest(SalePrice ~ ., data = df, ntree = 1250, importance = TRUE)

importance_df <- measure_importance(rf_model)

plot_multi_way_importance(
  importance_df,
  x_measure = "times_a_root",
  size_measure = "mean_min_depth"
)
```




```
# This chart shows how often each feature is selected as the root node of a tree.
# Since OverallQual was chosen as the root often, it's likely a key predictor.
```

```
importance_sorted <- importance_df[order(-importance_df$times_a_root), c("variable",
"times_a_root")]
head(importance_sorted, 10)
```

```
##        variable times_a_root
## 60  OverallQual          401
## 22    ExterQual          225
## 30   GarageCars          170
## 57 Neighborhood          137
## 36    GrLivArea           86
## 77    YearBuilt           55
## 42  KitchenQual           53
## 70  TotalBsmtSF           35
## 12     BsmtQual           32
## 29   GarageArea           25
```

 # Using a random forest model with 1250 trees, OverallQual was the strongest predicto r of house sale price, being chosen as the root node 337 times. Other key variables i nclude ExterQual, GarageCars, and Neighborhood, which were also frequently selected a s splits. This suggests that overall build quality, exterior condition, parking capac ity, and location are major contributors to home valuation.

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:randomForest':
##
##     combine
```

```
df <- read.csv("~/Downloads/house-prices-advanced-regression-techniques/train.csv")
df$Id <- NULL

predictors_num <- setdiff(names(df), "SalePrice")
predictors_num <- predictors_num[sapply(df[, predictors_num], is.numeric)]

predictors_cat <- setdiff(names(df), "SalePrice")
predictors_cat <- predictors_cat[sapply(df[, predictors_cat], function(x) is.factor(
x) || is.character(x))]

plots_num <- lapply(predictors_num, function(var) {
  ggplot(df, aes_string(x = var, y = "SalePrice")) +
    geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", color = "blue", se = FALSE) +
    ggtitle(paste("SalePrice vs", var)) +
    theme_minimal()
})
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
plots_cat <- lapply(predictors_cat, function(var) {
  ggplot(df, aes_string(x = var, y = "SalePrice")) +
    geom_boxplot() +
    ggtitle(paste("SalePrice vs", var)) +
    theme_minimal()
})

plots_all <- c(plots_num, plots_cat)

batch_size <- 4
num_pages <- ceiling(length(plots_all) / batch_size)

for (i in seq_len(num_pages)) {
  start <- (i - 1) * batch_size + 1
  end <- min(i * batch_size, length(plots_all))
  grid.arrange(grobs = plots_all[start:end], ncol = 2)
}
```

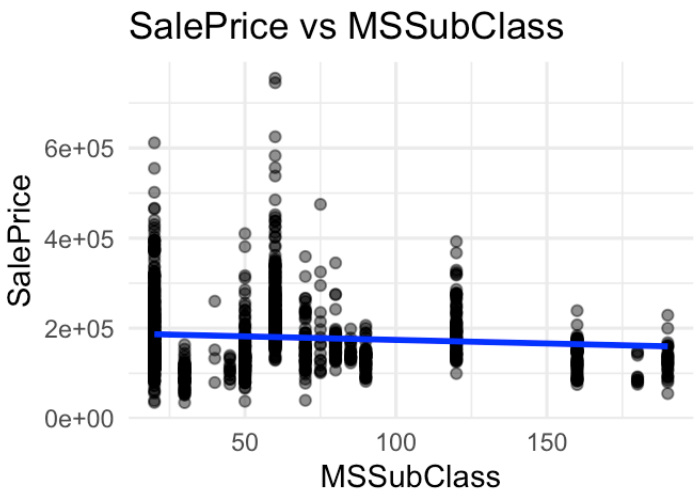
```
## `geom_smooth()` using formula = 'y ~ x'
```

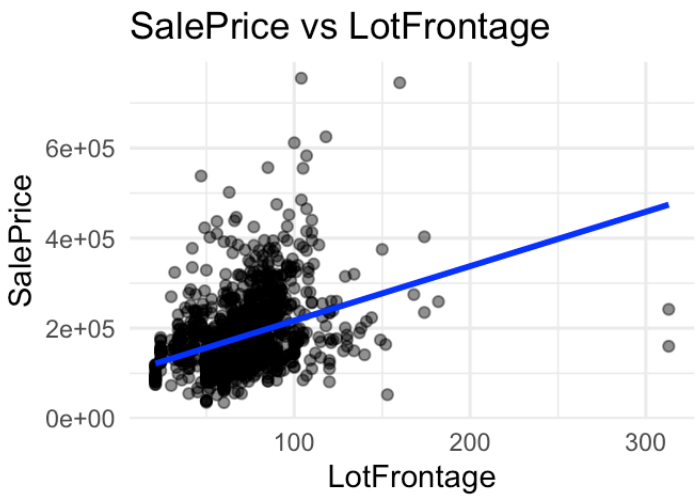
```
## `geom_smooth()` using formula = 'y ~ x'
```

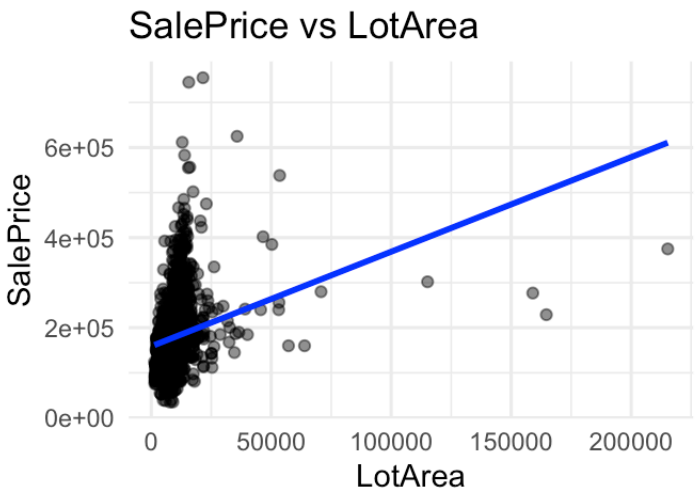
```
## Warning: Removed 259 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

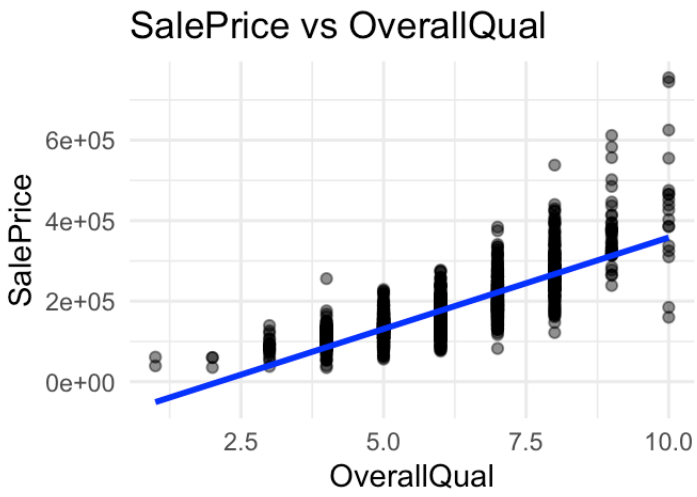
```
## Warning: Removed 259 rows containing missing values or values outside the scale ra
nge
## (`geom_point()`).
```

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





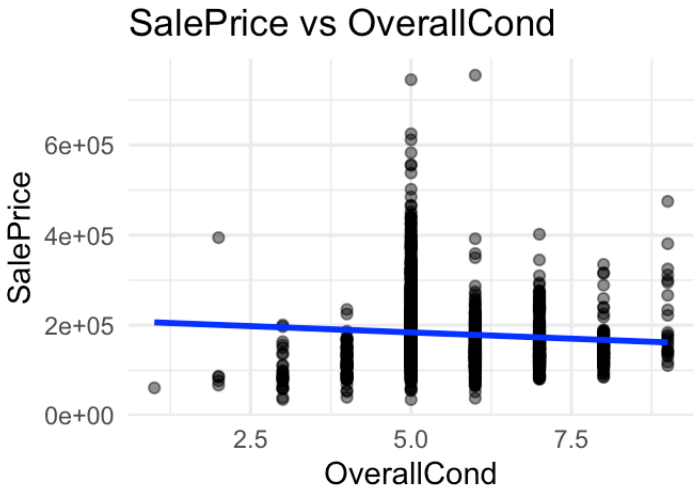


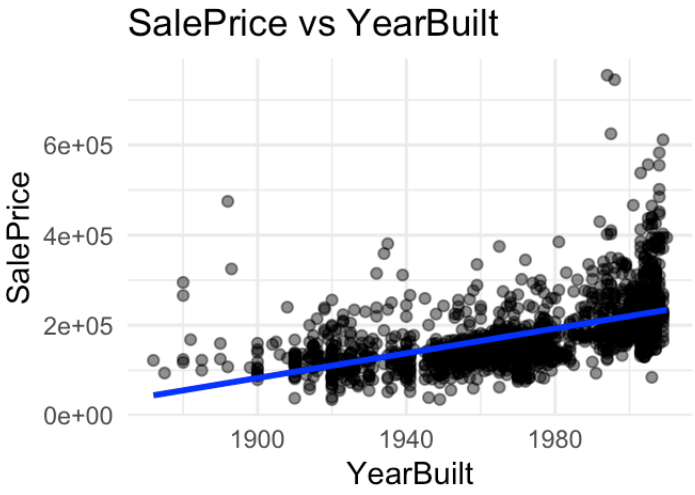


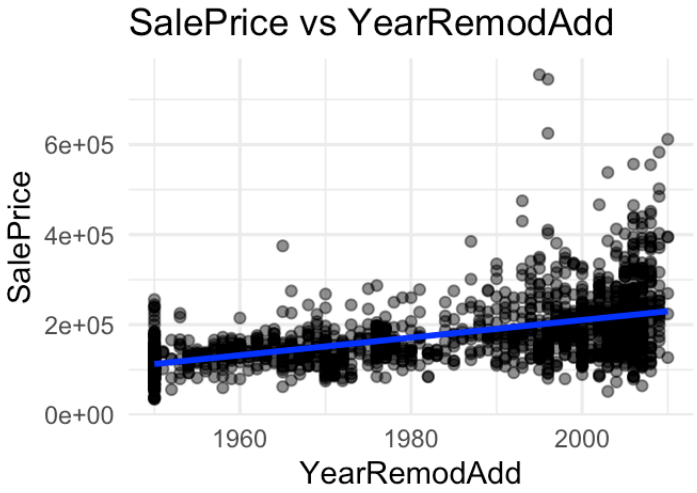

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

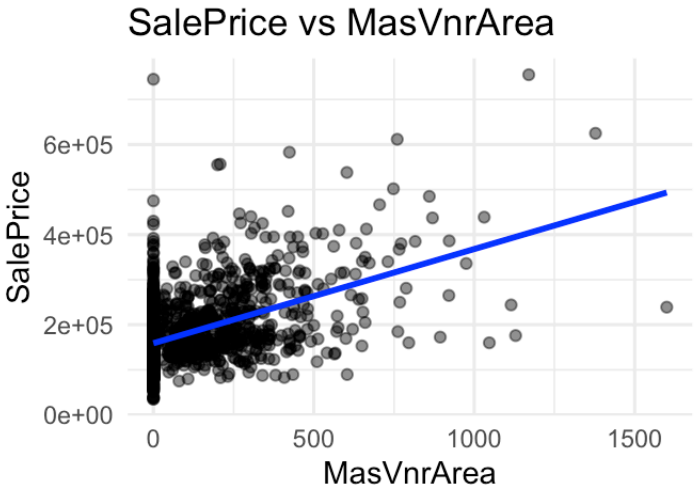
```
## Warning: Removed 8 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 8 rows containing missing values or values outside the scale rang
e
## (`geom_point()`).
```

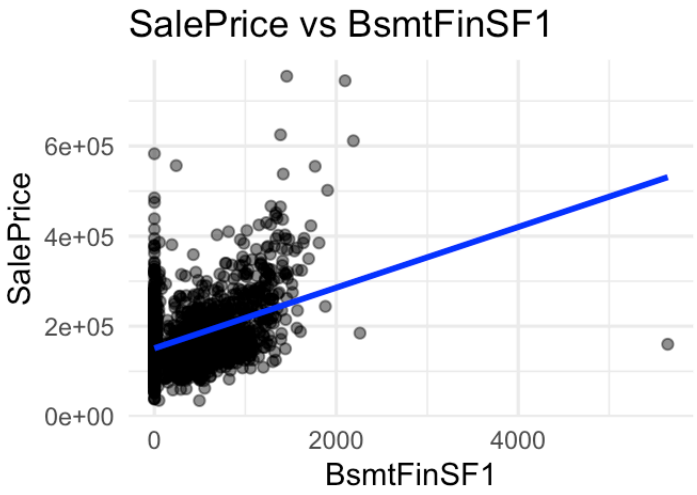


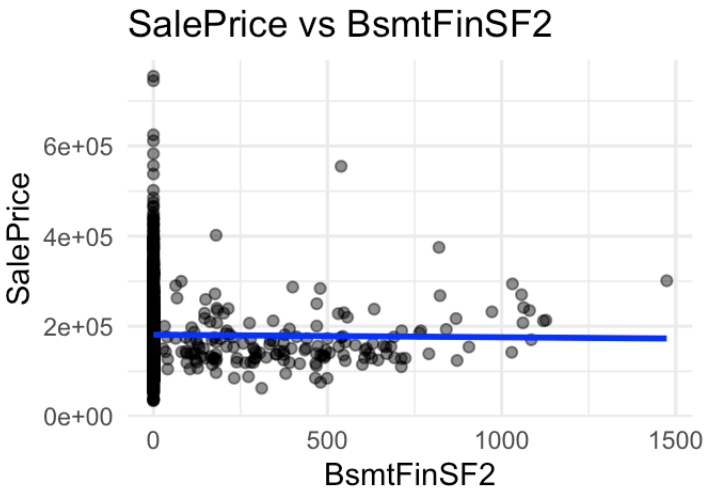


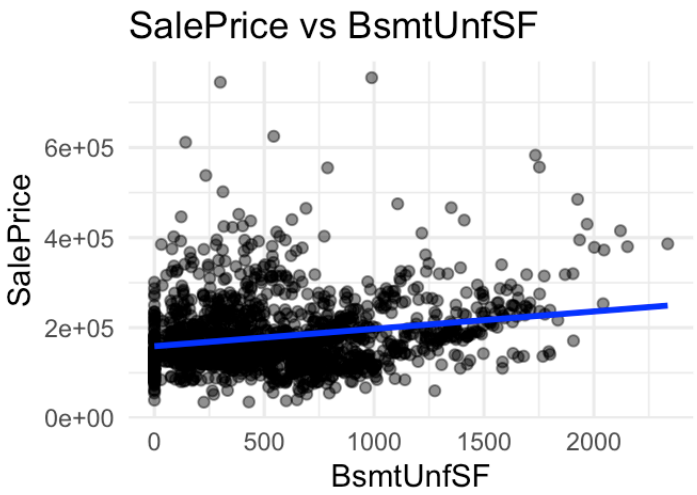


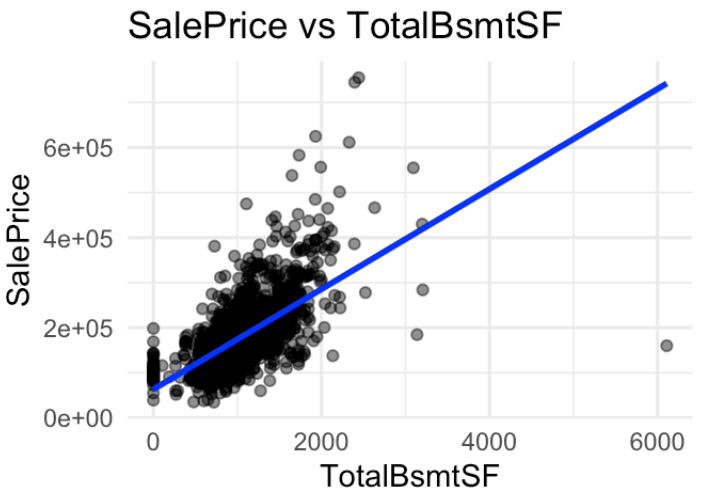



```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

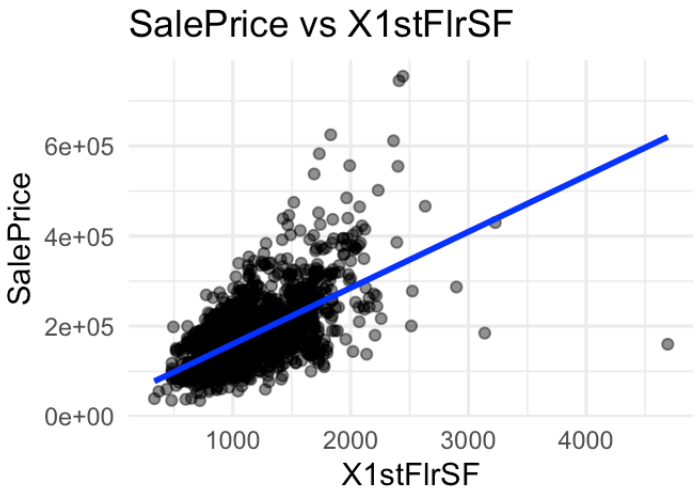


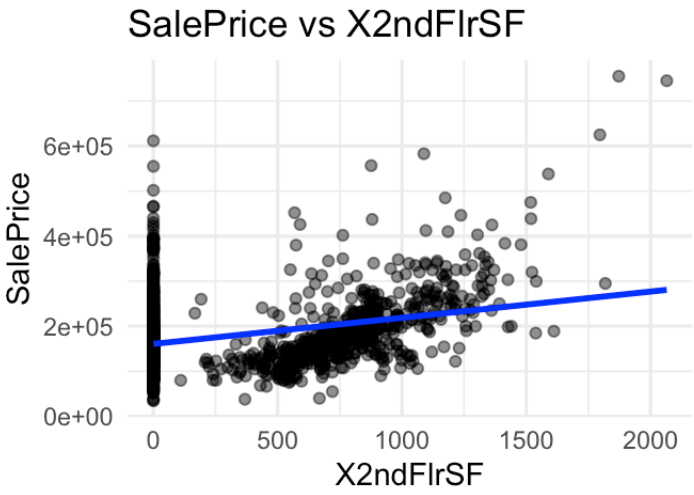


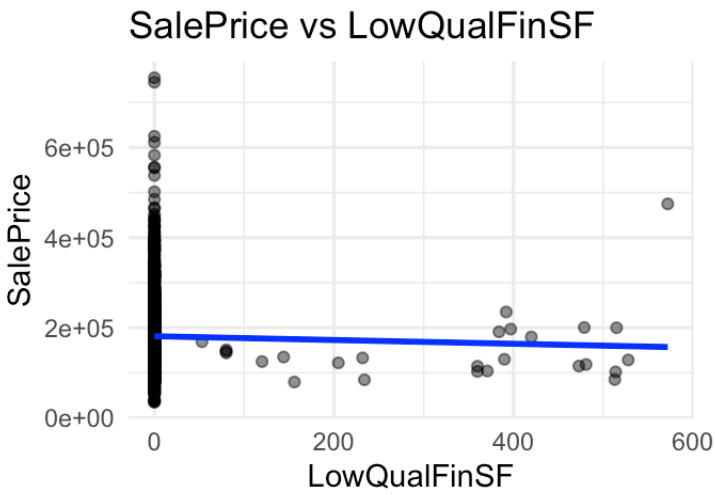


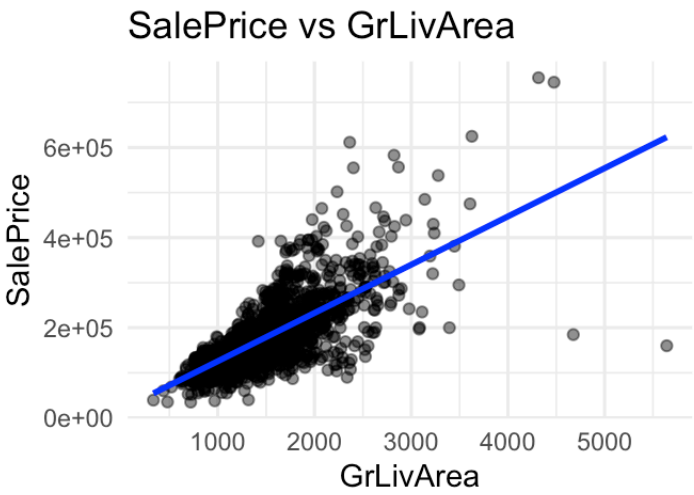



```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

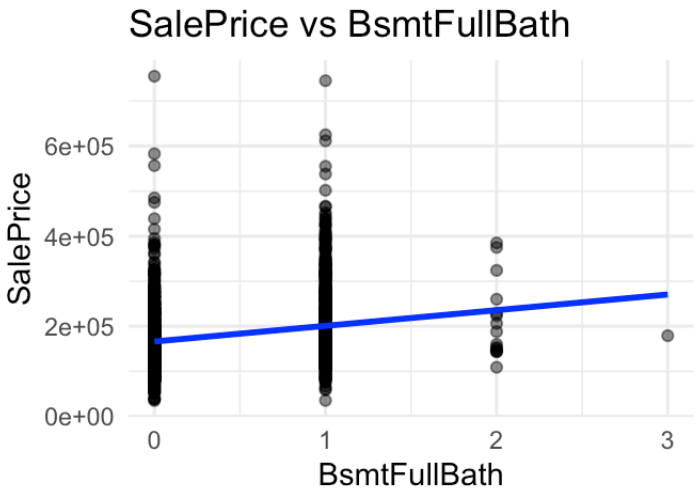


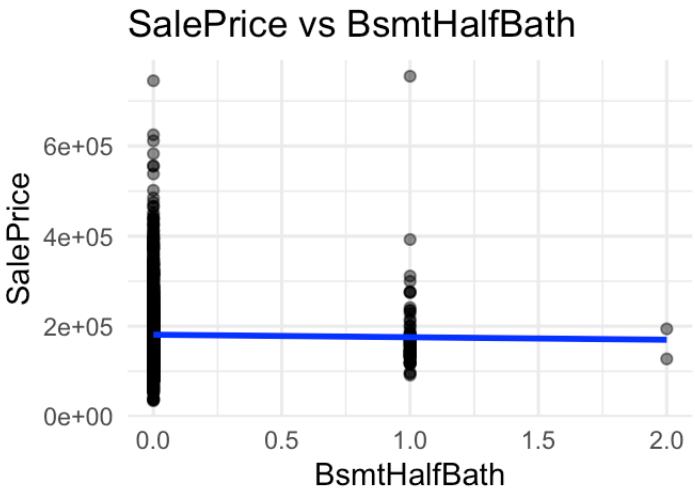


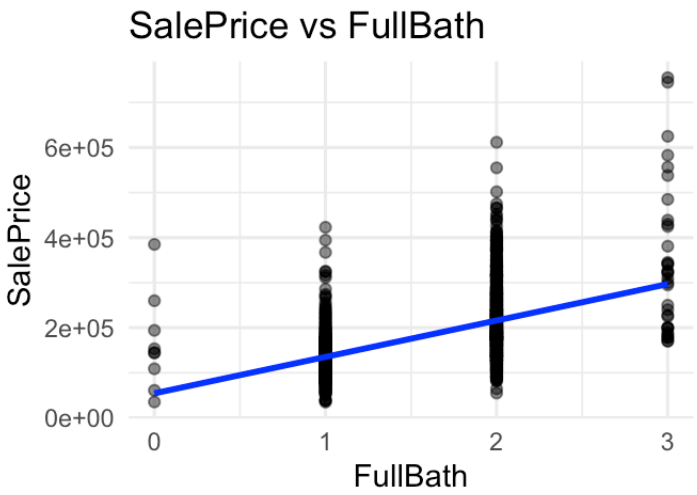


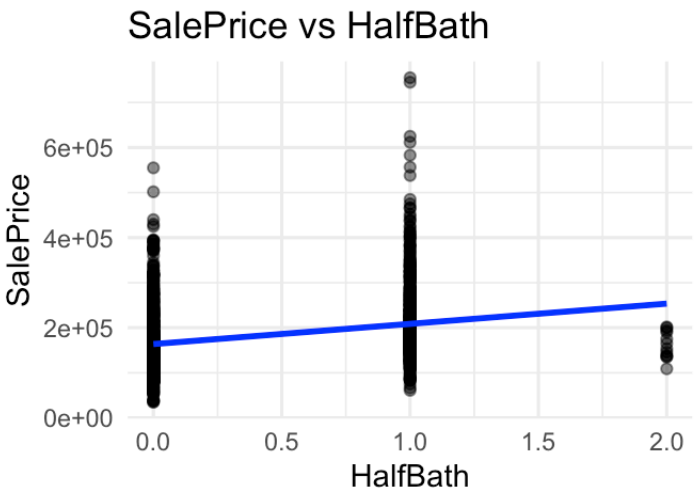



```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

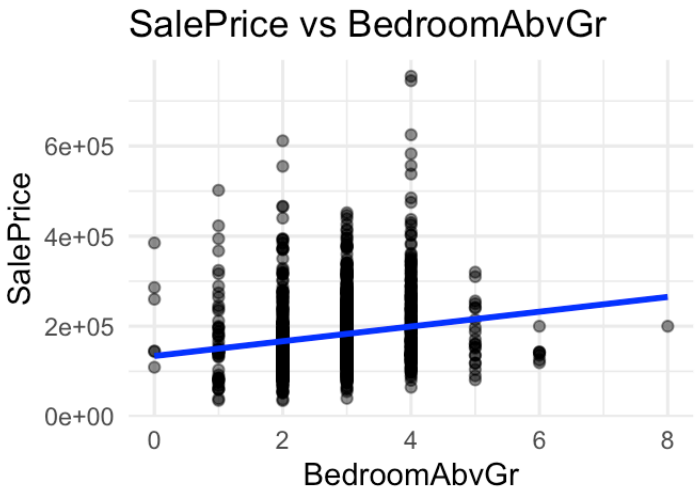


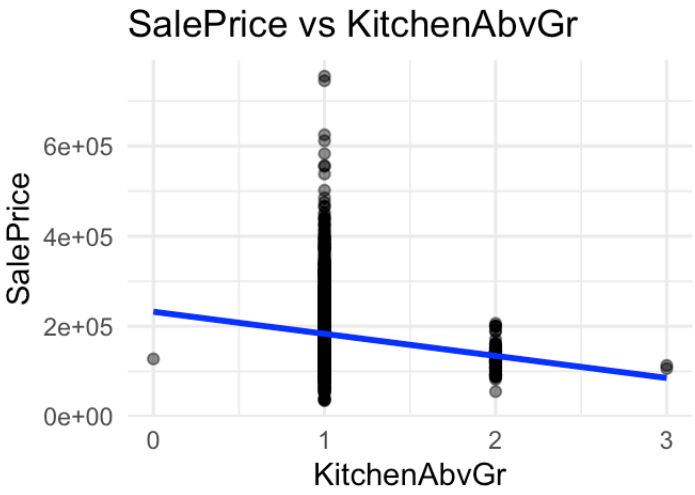


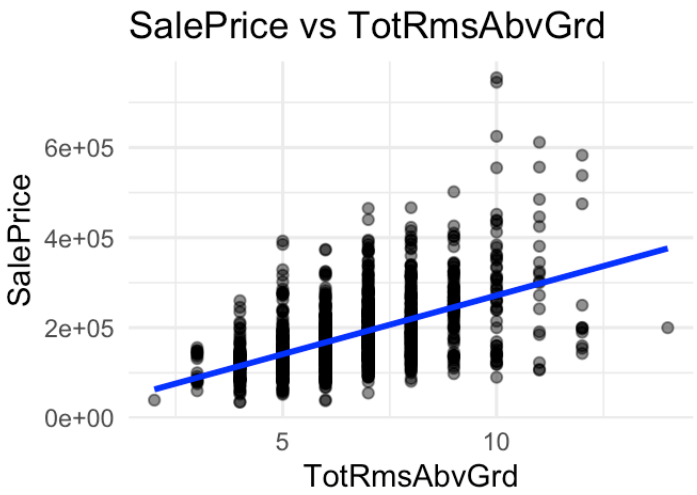


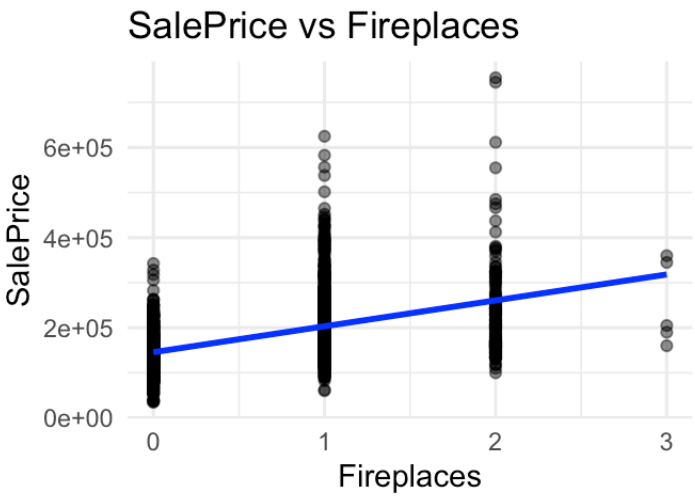



```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```







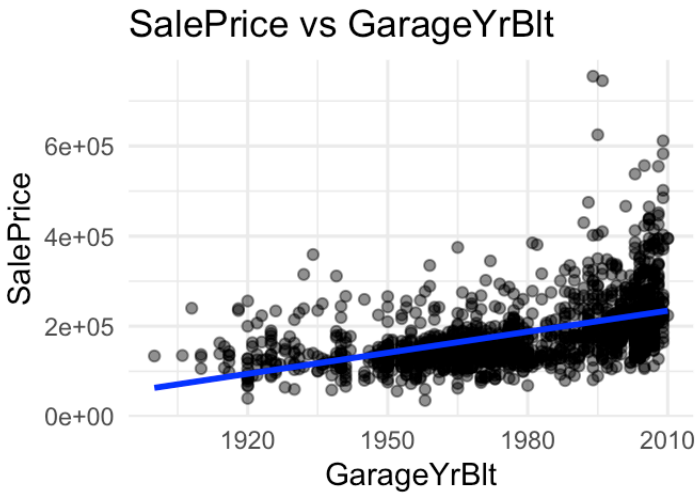


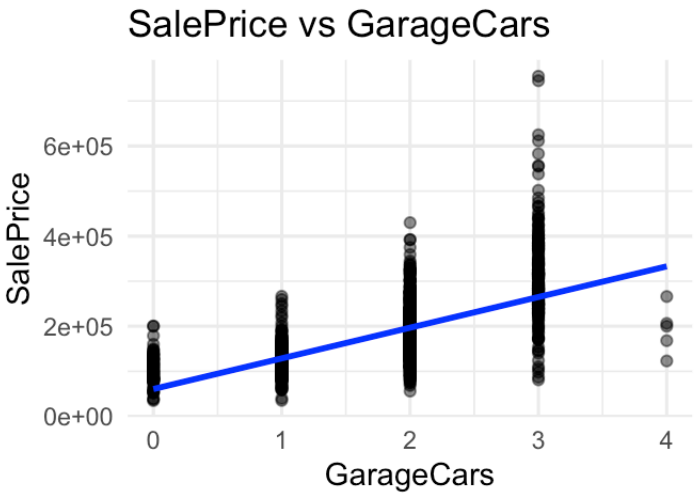

```
## `geom_smooth()` using formula = 'y ~ x'
```

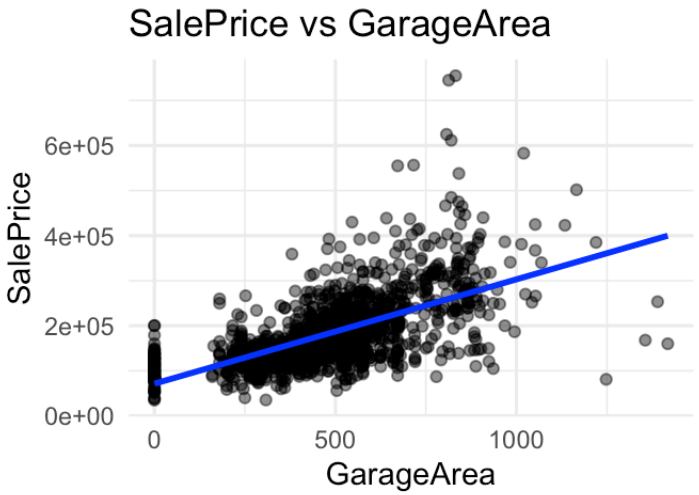
```
## Warning: Removed 81 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

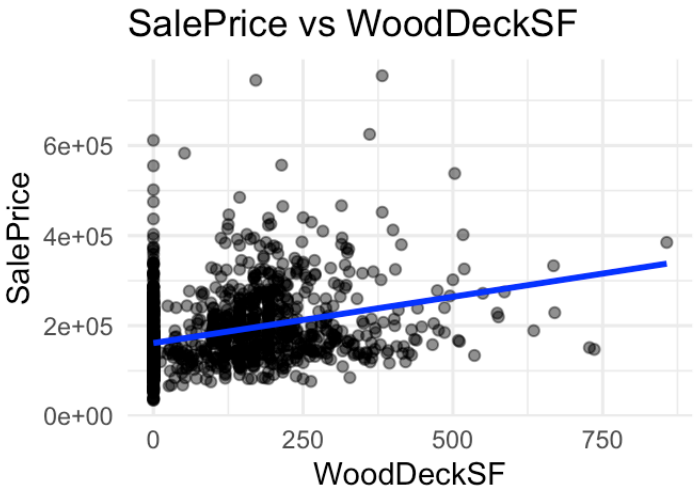
```
## Warning: Removed 81 rows containing missing values or values outside the scale ran
ge
## (`geom_point()`).
```

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

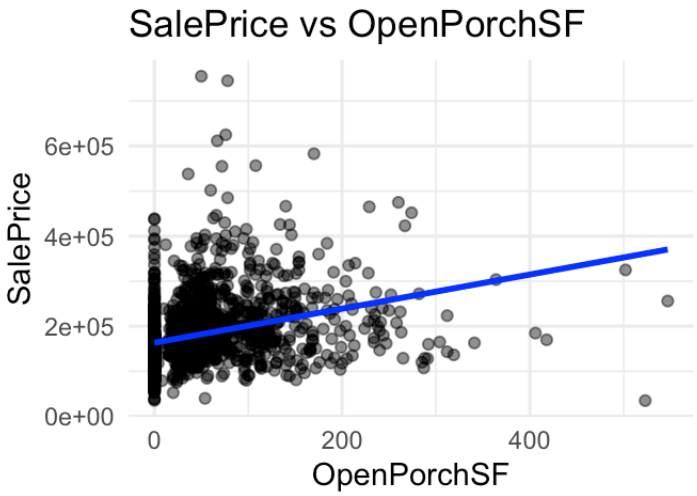


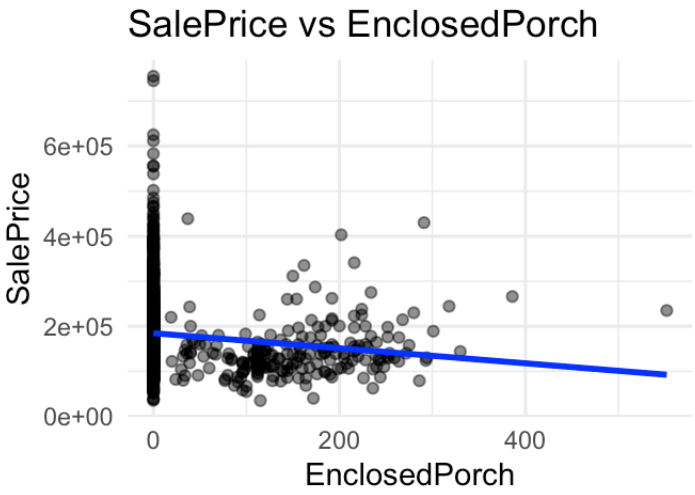


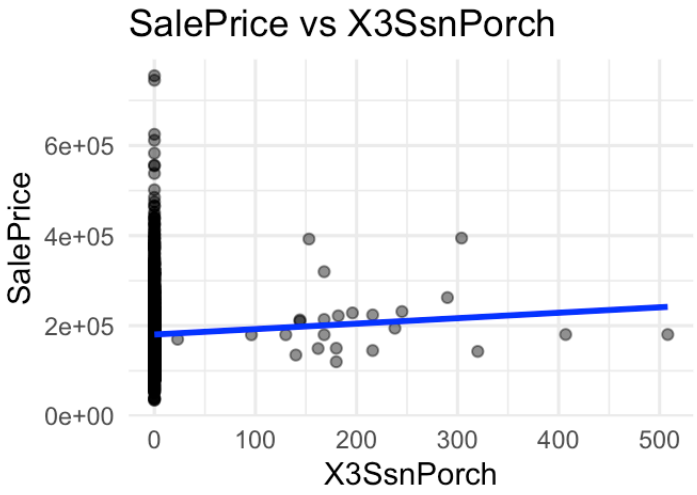


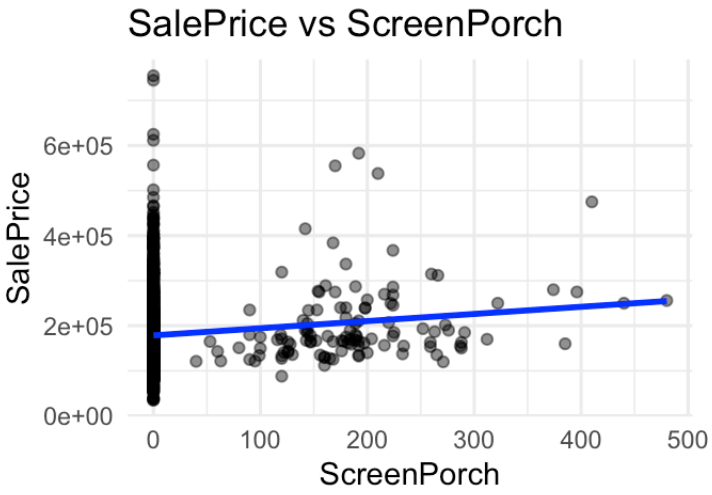



```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

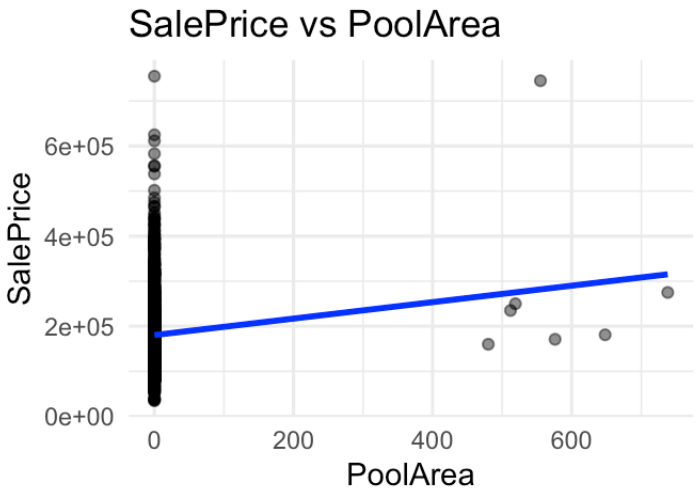


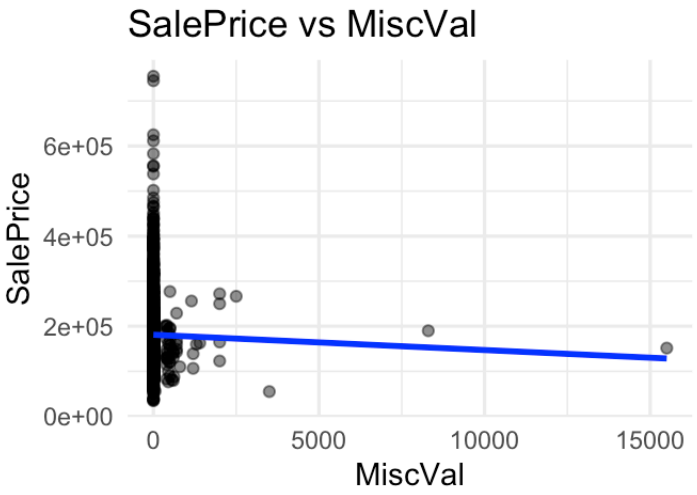


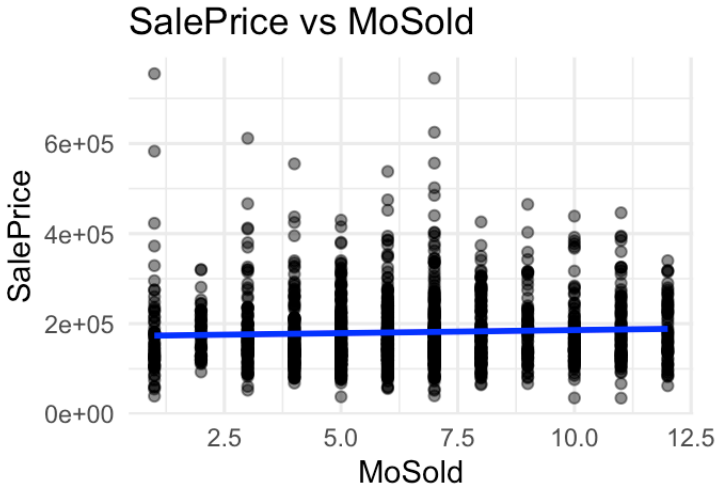


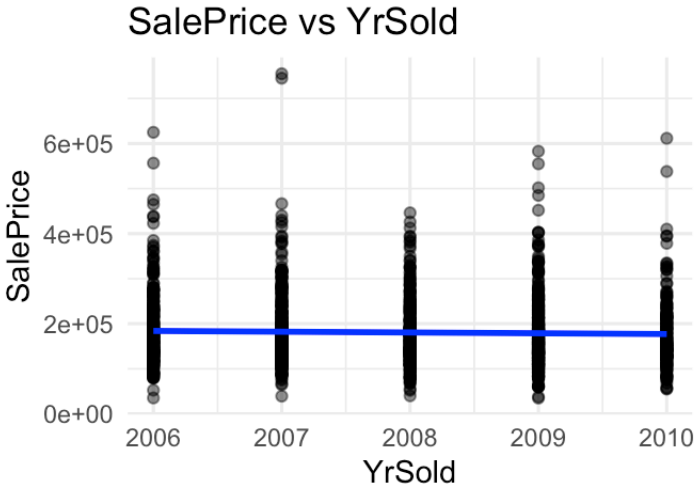


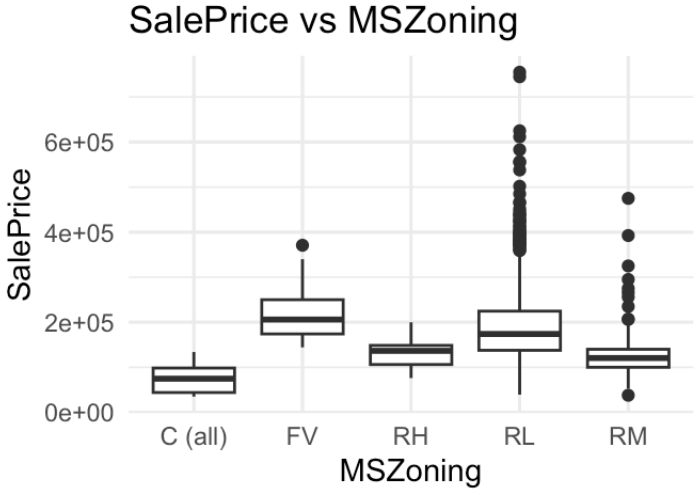

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```


SalePrice vs PoolArea
SalePrice
6e+05
4e+05
2e+05
0e+00
0
200
400
600
PoolArea


SalePrice vs MiscVal
SalePrice
6e+05
4e+05
2e+05
0e+00
0
5000
10000
15000
MiscVal

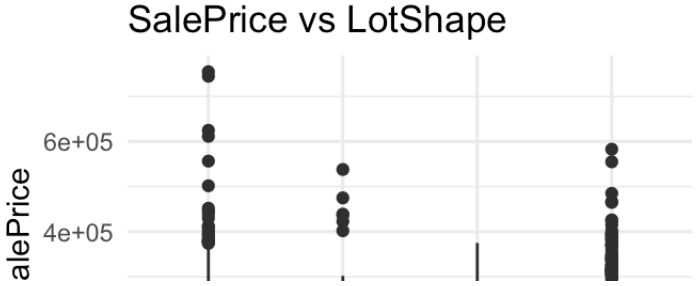

SalePrice vs MoSold
SalePrice
6e+05
4e+05
2e+05
0e+00
2.5
5.0
7.5
10.0
12.5
MoSold

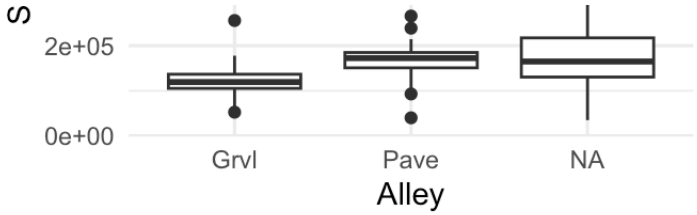

SalePrice vs YrSold
SalePrice
6e+05
4e+05
2e+05
0e+00
2006
2007
2008
2009
2010
YrSold

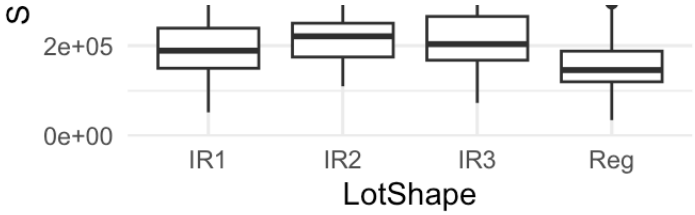

SalePrice vs MSZoning
SalePrice
6e+05
4e+05
2e+05
0e+00
C (all)
FV
RH
RL
RM
MSZoning

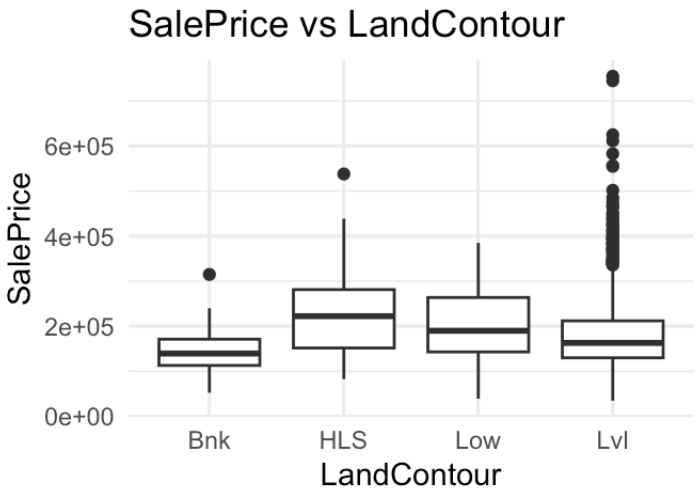

SalePrice vs Street
SalePrice
6e+05
4e+05
2e+05
0e+00
Grvl
Pave
Street

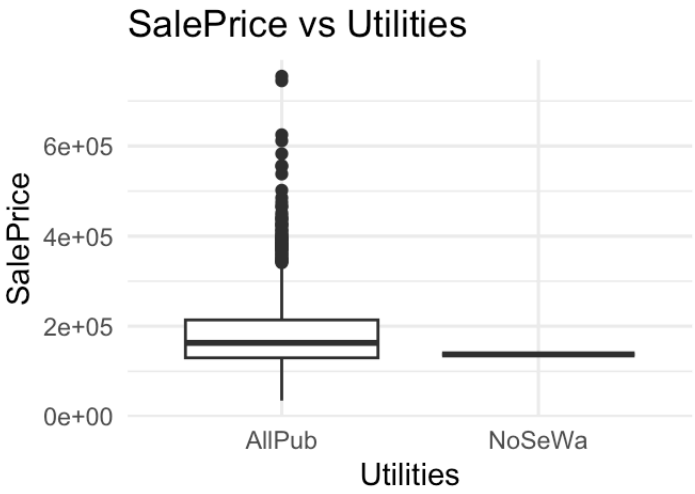

SalePrice vs Alley
alePrice
6e+05
4e+05

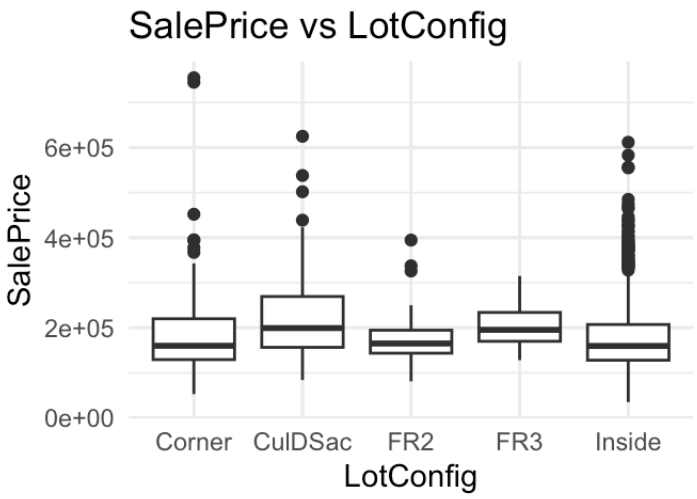

SalePrice vs LotShape
alePrice
6e+05
4e+05

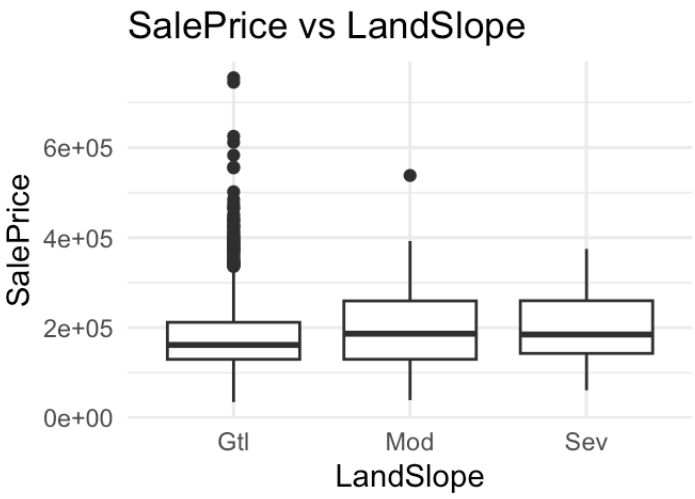

S
2e+05
0e+00
Grvl
Pave
NA
Alley

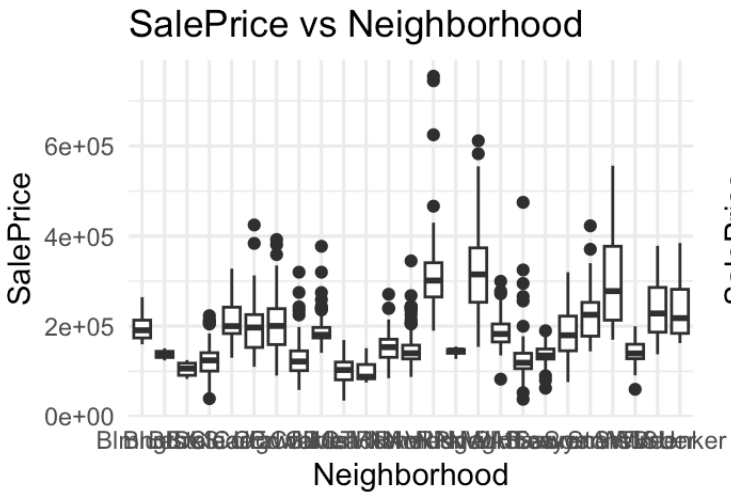

S
2e+05
0e+00
IR1
IR2
IR3
Reg
LotShape

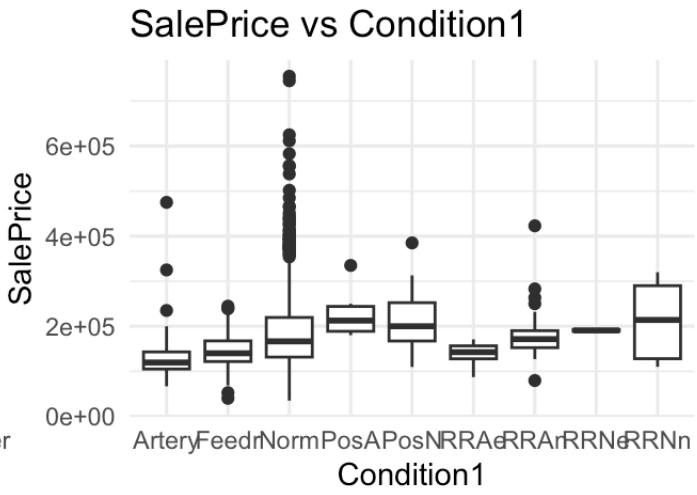

SalePrice vs LandContour
SalePrice
6e+05
4e+05
2e+05
0e+00
Bnk
HLS
Low
Lvl
LandContour

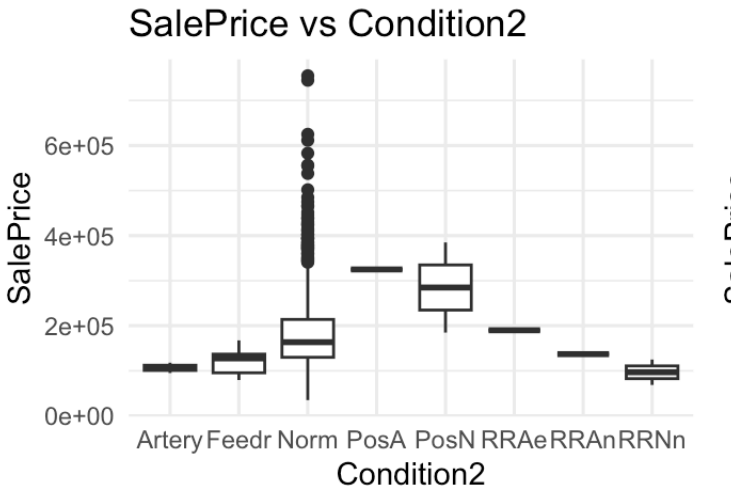

SalePrice vs Utilities
SalePrice
6e+05
4e+05
2e+05
0e+00
AllPub
NoSeWa
Utilities

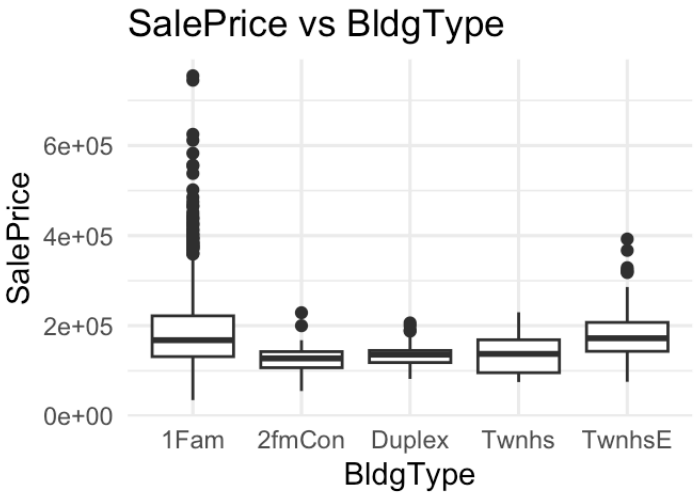

SalePrice vs LotConfig
SalePrice
6e+05
4e+05
2e+05
0e+00
Corner
CulDSac
FR2
FR3
Inside
LotConfig

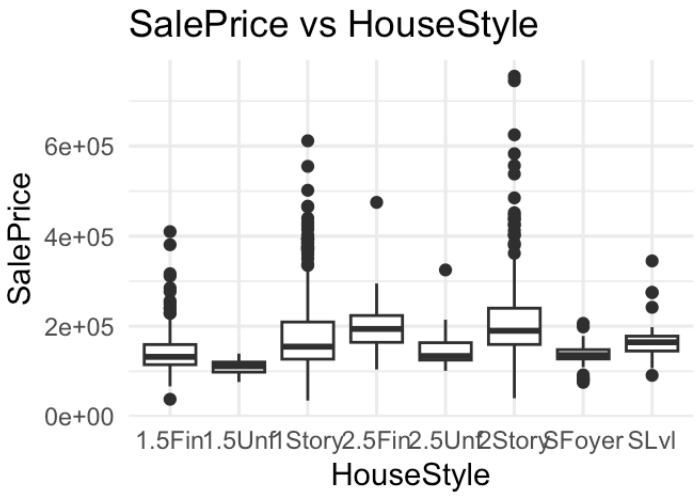

SalePrice vs LandSlope
SalePrice
6e+05
4e+05
2e+05
0e+00
Gtl
Mod
Sev
LandSlope

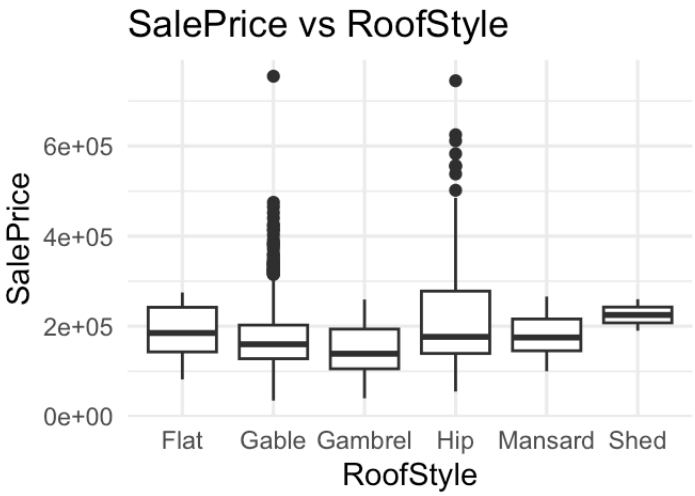

SalePrice vs Neighborhood
SalePrice
6e+05
4e+05
2e+05
0e+00
Neighborhood

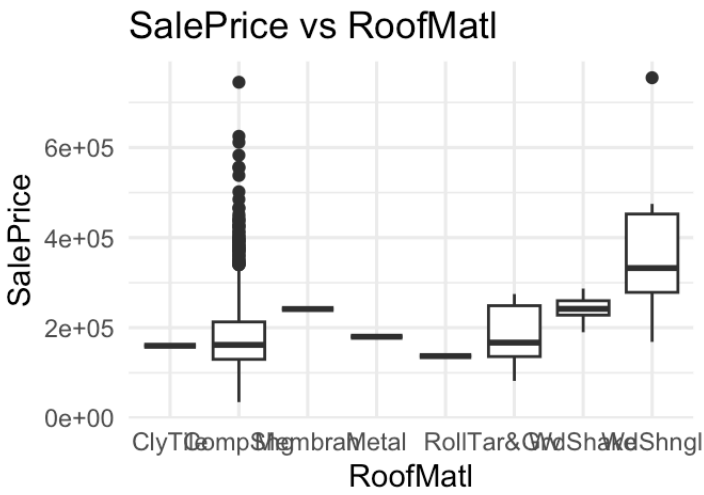

SalePrice vs Condition1
SalePrice
6e+05
4e+05
2e+05
0e+00
Artery Feedr Norm PosA PosN RRAe RRAn RRNn
Condition1

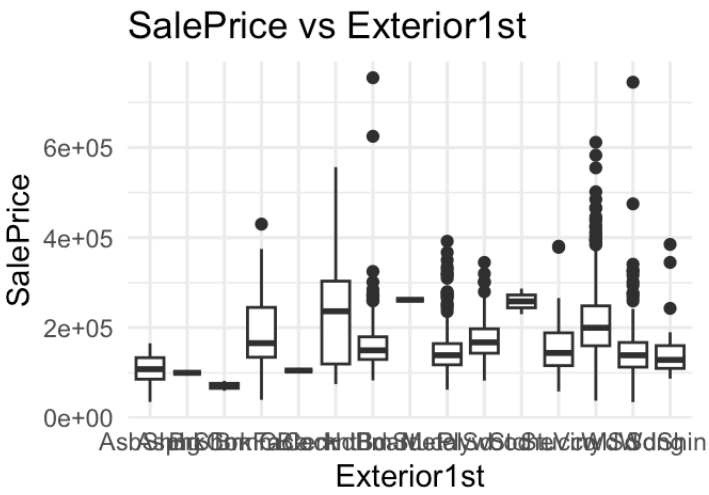

SalePrice vs Condition2
SalePrice
6e+05
4e+05
2e+05
0e+00
Artery Feedr Norm PosA PosN RRAe RRAn RRNn
Condition2

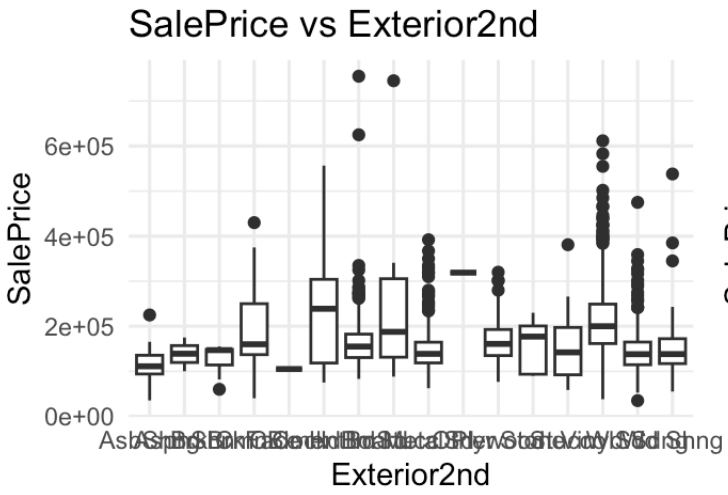

SalePrice vs BldgType
SalePrice
6e+05
4e+05
2e+05
0e+00
1Fam 2fmCon Duplex Twnhs TwnhsE
BldgType

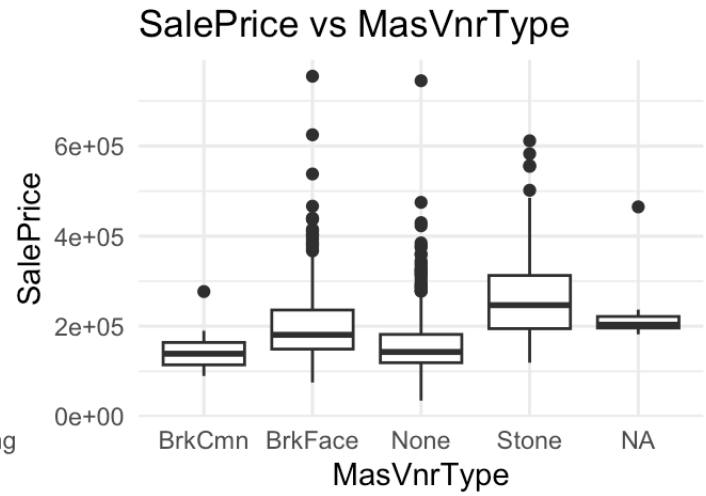

SalePrice vs HouseStyle
SalePrice
6e+05
4e+05
2e+05
0e+00
1.5Fin
1.5Unf
1Story
2.5Fin
2.5Unf
2Story
SFoyer
SLvl
HouseStyle

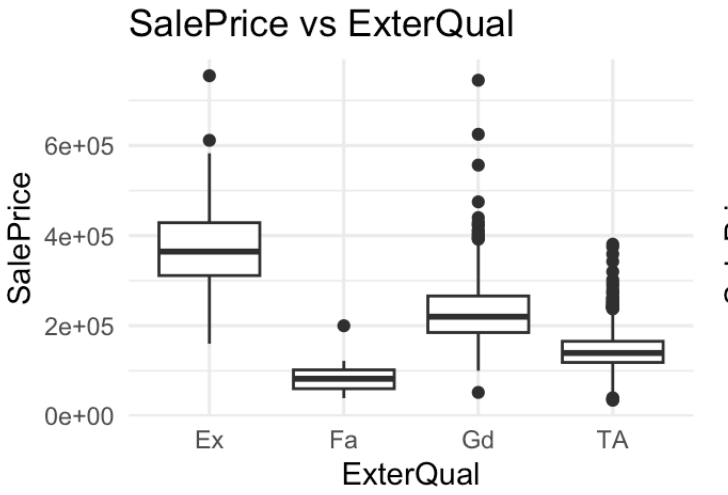

SalePrice vs RoofStyle
SalePrice
6e+05
4e+05
2e+05
0e+00
Flat
Gable
Gambrel
Hip
Mansard
Shed
RoofStyle

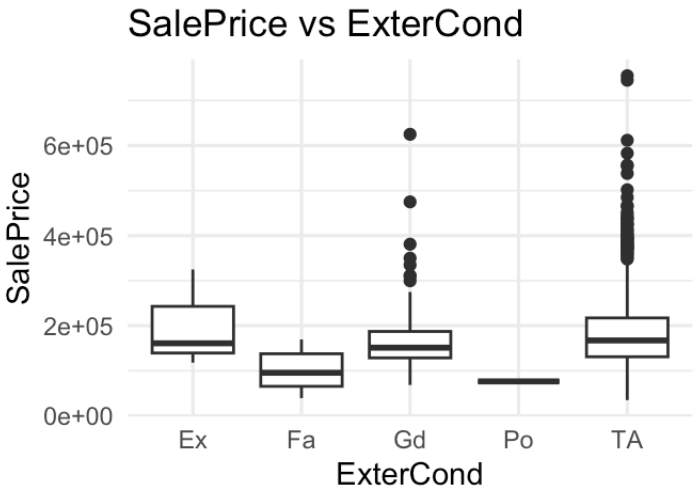

SalePrice vs RoofMatl
SalePrice
6e+05
4e+05
2e+05
0e+00
ClyTile
Membran
Metal
RollTar
Shake
Shngl
RoofMatl

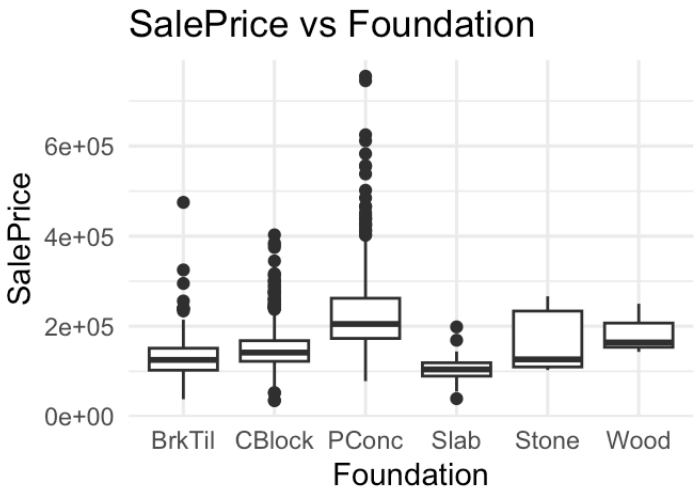

SalePrice vs Exterior1st
SalePrice
6e+05
4e+05
2e+05
0e+00
Exterior1st

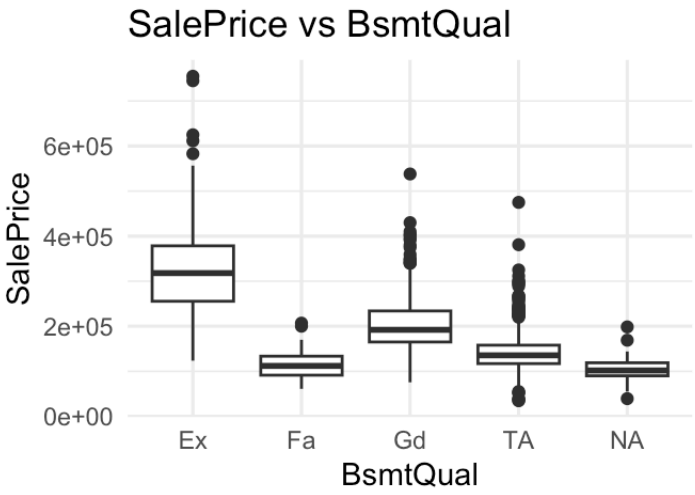

SalePrice vs Exterior2nd
SalePrice
6e+05
4e+05
2e+05
0e+00
Exterior2nd

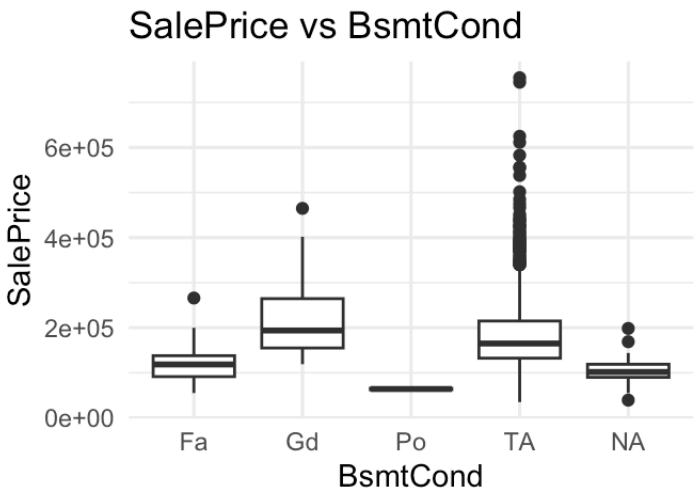

SalePrice vs MasVnrType
SalePrice
6e+05
4e+05
2e+05
0e+00
BrkCmn
BrkFace
None
Stone
NA
MasVnrType

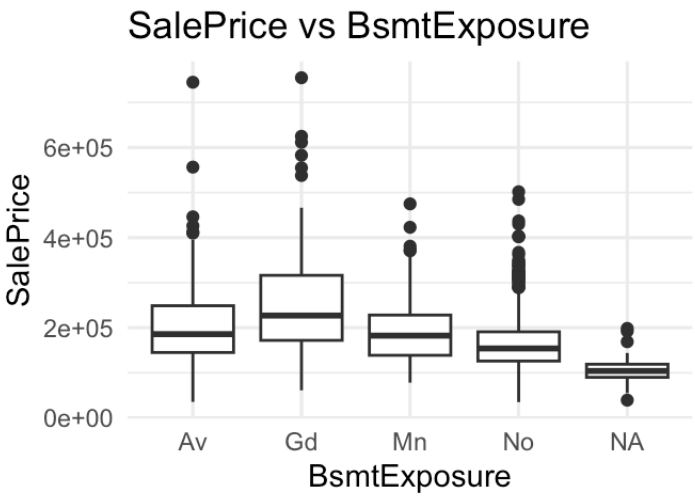

SalePrice vs ExterQual
SalePrice
6e+05
4e+05
2e+05
0e+00
Ex
Fa
Gd
TA
ExterQual

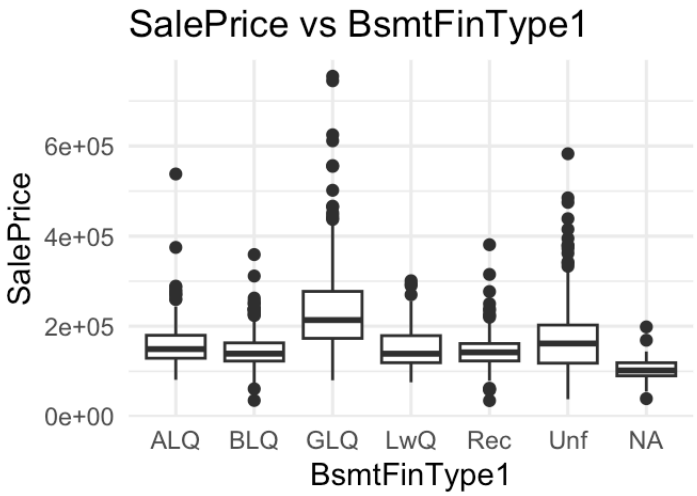

SalePrice vs ExterCond
SalePrice
6e+05
4e+05
2e+05
0e+00
Ex
Fa
Gd
Po
TA
ExterCond

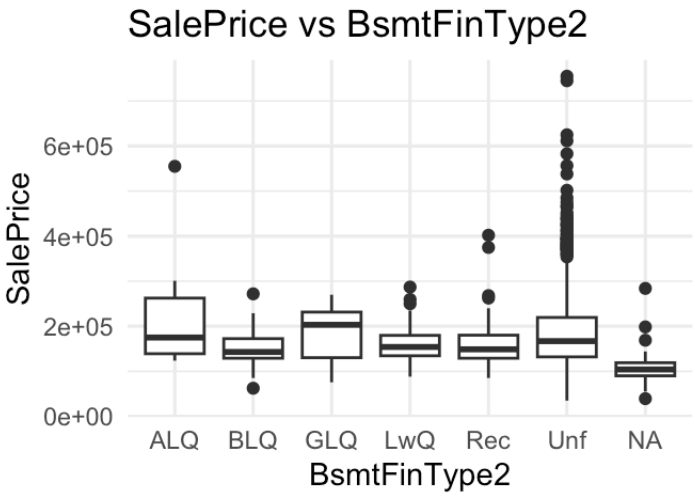

SalePrice vs Foundation
SalePrice
6e+05
4e+05
2e+05
0e+00
BrkTil
CBlock
PConc
Slab
Stone
Wood
Foundation

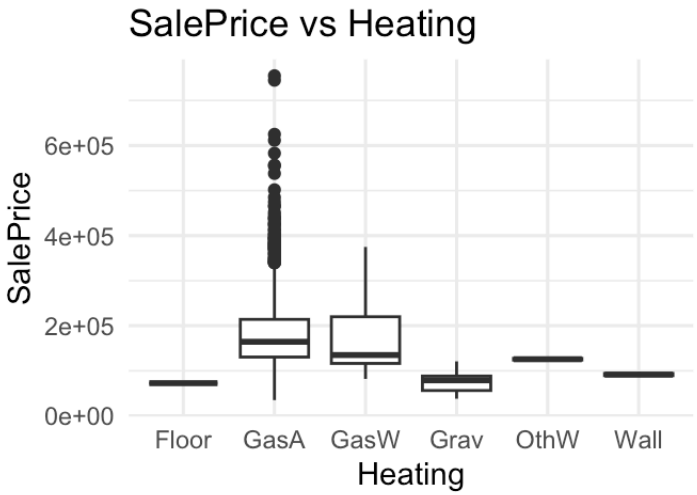

SalePrice vs BsmtQual
SalePrice
6e+05
4e+05
2e+05
0e+00
Ex
Fa
Gd
TA
NA
BsmtQual

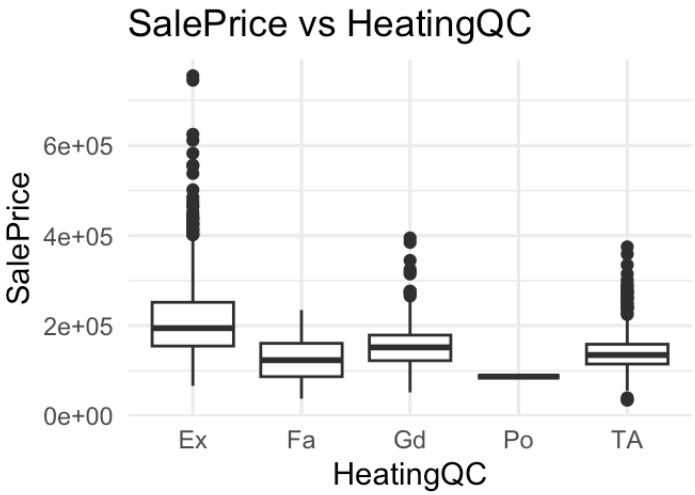

SalePrice vs BsmtCond
SalePrice
6e+05
4e+05
2e+05
0e+00
Fa
Gd
Po
TA
NA
BsmtCond

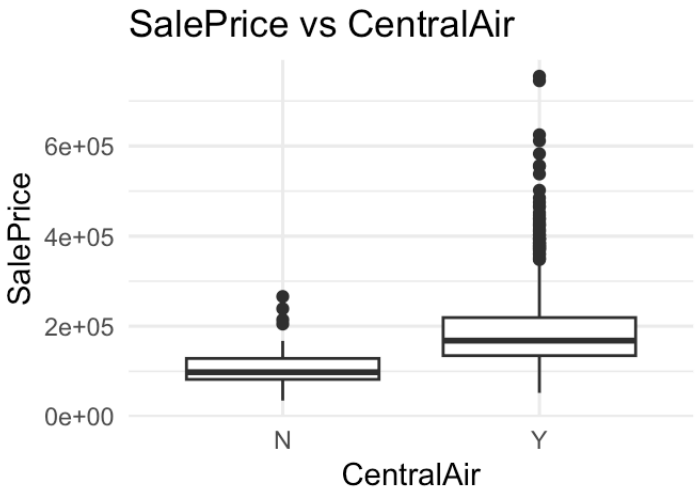

SalePrice vs BsmtExposure
SalePrice
6e+05
4e+05
2e+05
0e+00
Av
Gd
Mn
No
NA
BsmtExposure

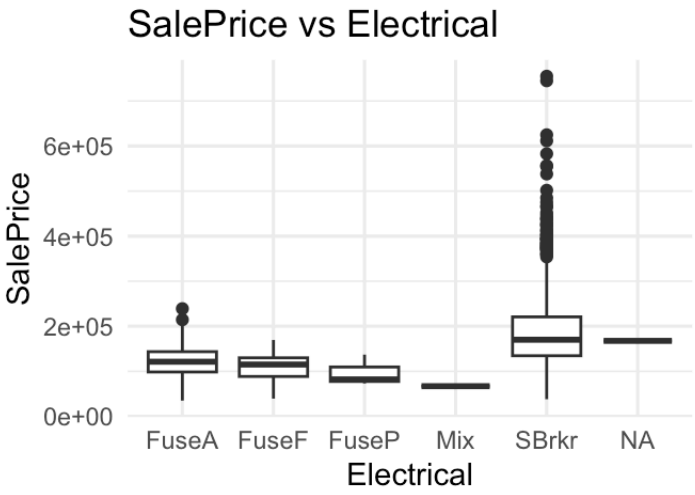

SalePrice vs BsmtFinType1
SalePrice
6e+05
4e+05
2e+05
0e+00
ALQ
BLQ
GLQ
LwQ
Rec
Unf
NA
BsmtFinType1

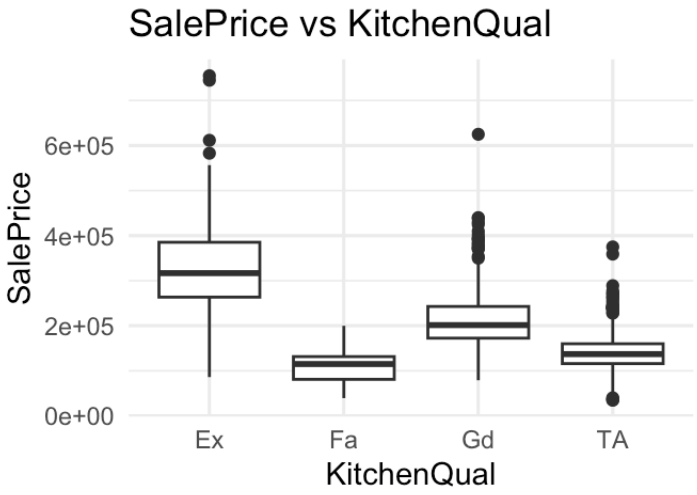

SalePrice vs BsmtFinType2
SalePrice
6e+05
4e+05
2e+05
0e+00
ALQ
BLQ
GLQ
LwQ
Rec
Unf
NA
BsmtFinType2

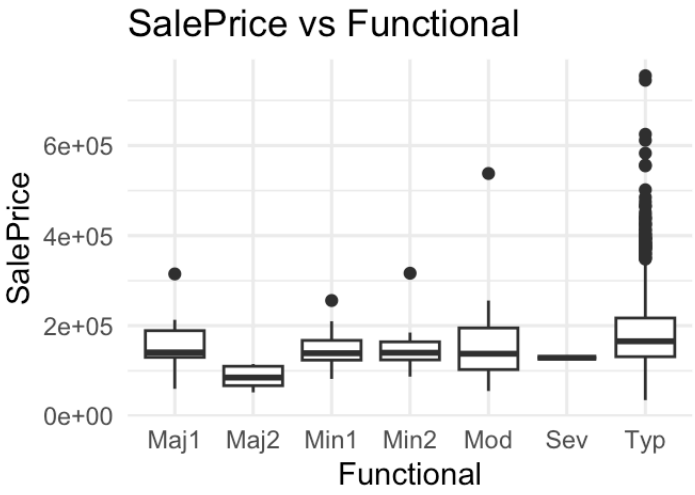

SalePrice vs Heating
SalePrice
6e+05
4e+05
2e+05
0e+00
Floor
GasA
GasW
Grav
OthW
Wall
Heating

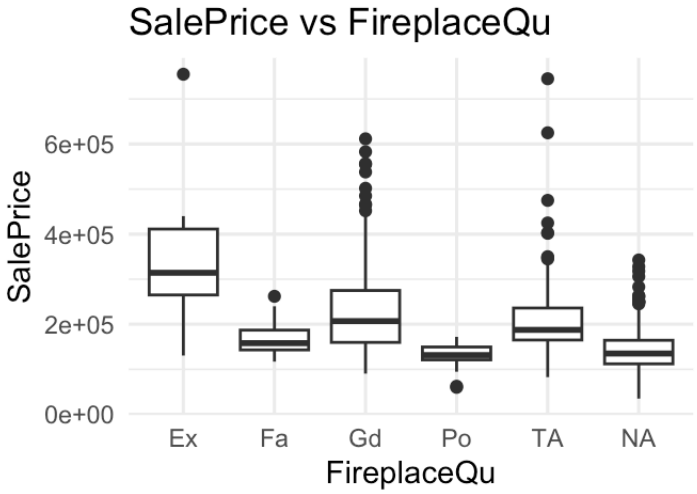

SalePrice vs HeatingQC
SalePrice
6e+05
4e+05
2e+05
0e+00
Ex
Fa
Gd
Po
TA
HeatingQC

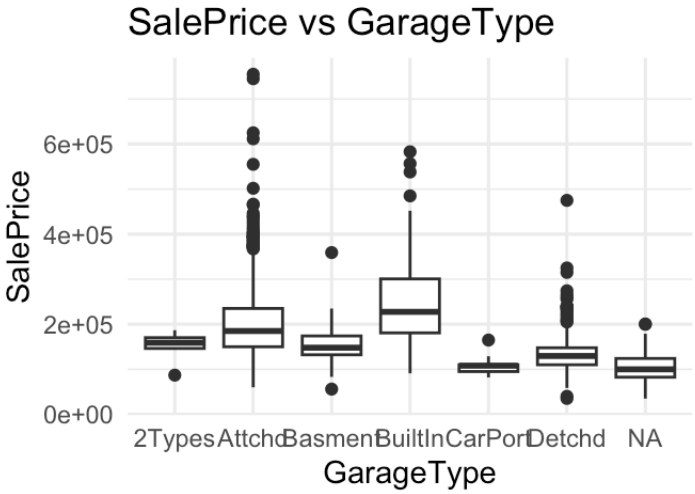

SalePrice vs CentralAir
SalePrice
6e+05
4e+05
2e+05
0e+00
N
Y
CentralAir

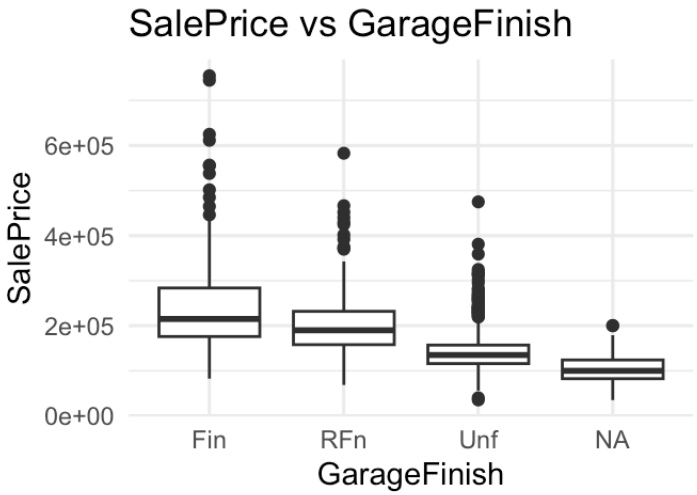

SalePrice vs Electrical
SalePrice
6e+05
4e+05
2e+05
0e+00
FuseA
FuseF
FuseP
Mix
SBrkr
NA
Electrical

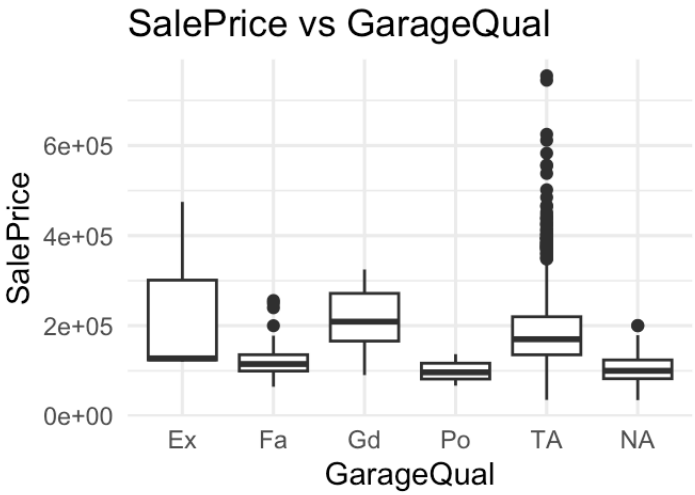

SalePrice vs KitchenQual
SalePrice
6e+05
4e+05
2e+05
0e+00
Ex
Fa
Gd
TA
KitchenQual

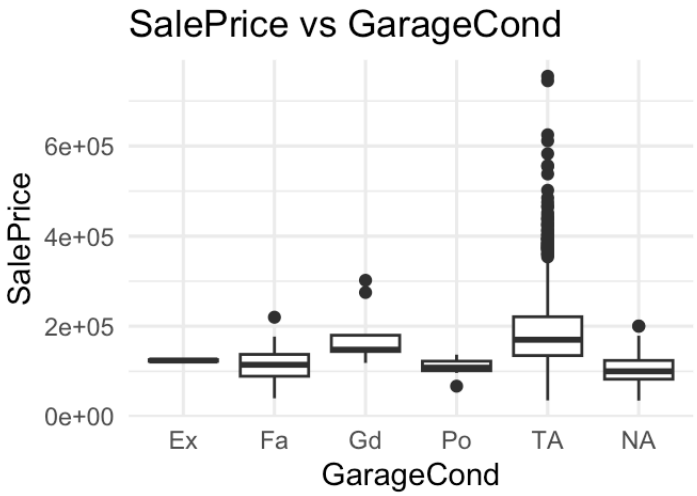

SalePrice vs Functional
SalePrice
6e+05
4e+05
2e+05
0e+00
Maj1
Maj2
Min1
Min2
Mod
Sev
Typ
Functional

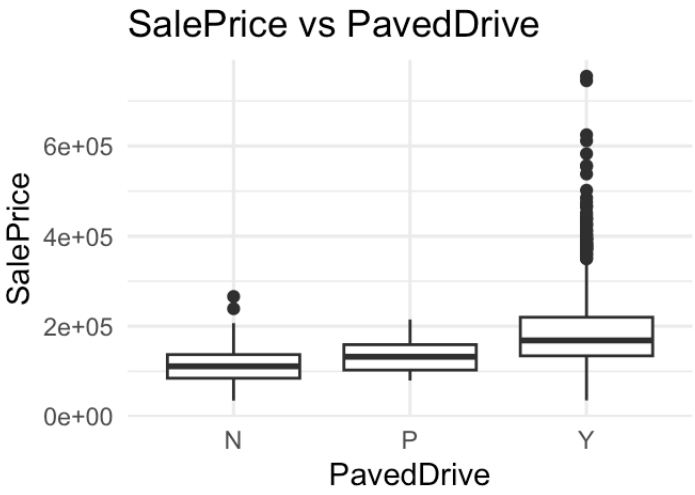

SalePrice vs FireplaceQu
SalePrice
6e+05
4e+05
2e+05
0e+00
Ex
Fa
Gd
Po
TA
NA
FireplaceQu

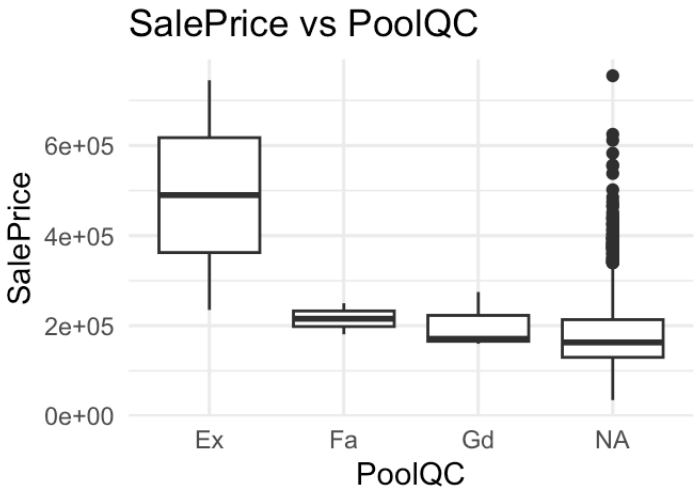

SalePrice vs GarageType
SalePrice
6e+05
4e+05
2e+05
0e+00
2Types
Attchd
Basment
BuiltIn
CarPort
Detchd
NA
GarageType

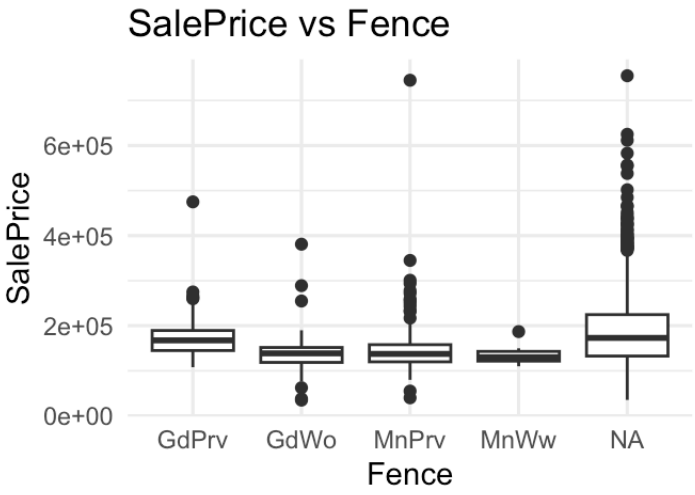

SalePrice vs GarageFinish
SalePrice
6e+05
4e+05
2e+05
0e+00
Fin
RFn
Unf
NA
GarageFinish

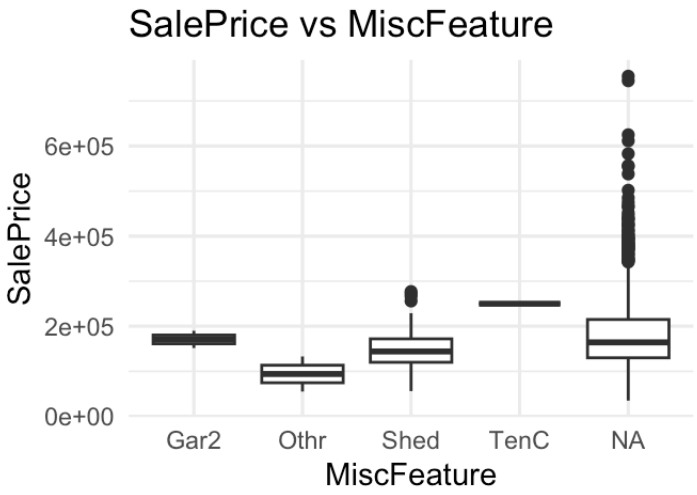

SalePrice vs GarageQual
SalePrice
6e+05
4e+05
2e+05
0e+00
Ex
Fa
Gd
Po
TA
NA
GarageQual

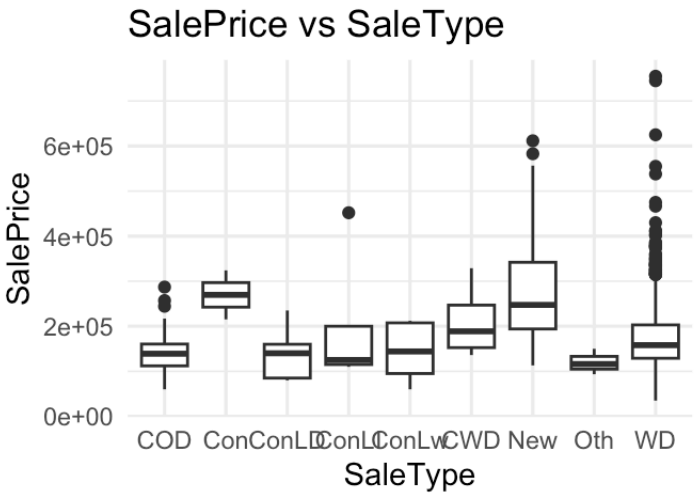

SalePrice vs GarageCond
SalePrice
6e+05
4e+05
2e+05
0e+00
Ex
Fa
Gd
Po
TA
NA
GarageCond

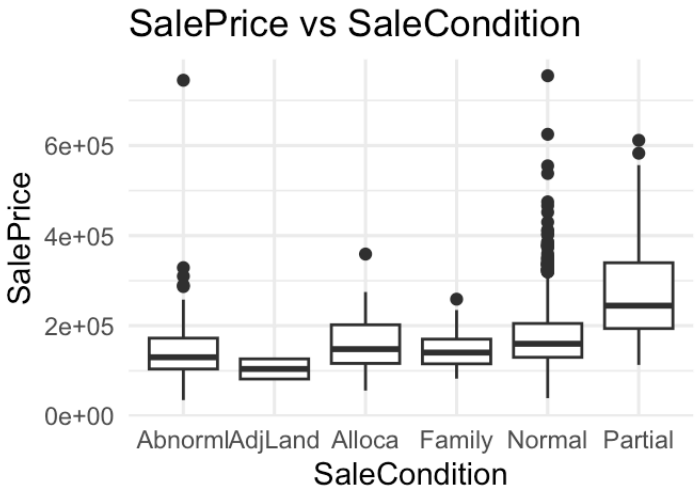

SalePrice vs PavedDrive
SalePrice
6e+05
4e+05
2e+05
0e+00
N
P
Y
PavedDrive


SalePrice vs PoolQC
SalePrice
6e+05
4e+05
2e+05
0e+00
Ex
Fa
Gd
NA
PoolQC


SalePrice vs Fence
SalePrice
6e+05
4e+05
2e+05
0e+00
GdPrv
GdWo
MnPrv
MnWw
NA
Fence


SalePrice vs MiscFeature
SalePrice
6e+05
4e+05
2e+05
0e+00
Gar2
Othr
Shed
TenC
NA
MiscFeature


SalePrice vs SaleType
SalePrice
6e+05
4e+05
2e+05
0e+00
COD
Con
ConLD
ConLI
ConLw
CWD
New
Oth
WD
SaleType


SalePrice vs SaleCondition
SalePrice
6e+05
4e+05
2e+05
0e+00
Abnorml
AdjLand
Alloca
Family
Normal
Partial
SaleCondition

```
df <- read.csv("~/Downloads/house-prices-advanced-regression-techniques/train.csv")
df$Id <- NULL  # Remove ID column

predictors <- setdiff(names(df), "SalePrice")
predictors <- predictors[sapply(df[, predictors], is.numeric)]

results <- lapply(predictors, function(var) {
  formula <- as.formula(paste("SalePrice ~", var))
  model <- lm(formula, data = df)
  summary_stats <- summary(model)

  list(
    variable = var,
    r_squared = summary_stats$r.squared,
    p_value = coef(summary_stats)[2, 4],
    estimate = coef(summary_stats)[2, 1]
  )
})

results_df <- do.call(rbind, lapply(results, as.data.frame))
results_df <- results_df[order(-results_df$r_squared), ]

print(head(results_df, 10))
```

```
##        variable r_squared       p_value   estimate
## 4   OverallQual 0.6256519 2.185675e-313 45435.8026
## 16    GrLivArea 0.5021487 4.518034e-223   107.1304
## 26   GarageCars 0.4101239 2.498644e-169 68077.9976
## 27   GarageArea 0.3886668 5.265038e-158   231.6456
## 12  TotalBsmtSF 0.3764811 9.484229e-152   111.1096
## 13    X1stFlrSF 0.3670569 5.394711e-147   124.5006
## 19     FullBath 0.3143439 1.236470e-121 80848.1668
## 23 TotRmsAbvGrd 0.2848604 2.772281e-108 26086.1808
## 6     YearBuilt 0.2734216 2.990229e-103  1375.3735
## 7  YearRemodAdd 0.2571514  3.164948e-96  1951.2994
```

#OverallQual and GarageCars consistently emerged as the strongest predictors and were statistically significant in both Random Forest and Logistic Regression.

```
library(randomForest)
library(randomForestExplainer)
library(dplyr)

df <- read.csv("~/Downloads/house-prices-advanced-regression-techniques/train.csv")
df$Id <- NULL

df <- df %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), median(., na.rm = TRUE), .))) %
>%
  mutate(across(where(is.character), ~ ifelse(is.na(.), "Missing", .))) %>%
  mutate(across(where(is.character), as.factor))

median_price <- median(df$SalePrice, na.rm = TRUE)
df$SalePriceBinary <- ifelse(df$SalePrice > median_price, 1, 0)

set.seed(123)
rf_model <- randomForest(SalePriceBinary ~ ., data = df %>% select(-SalePrice), ntree
= 1000, importance = TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values.  Are you sure you want to do regression?
```

```
importance_df <- measure_importance(rf_model)

top_features <- importance_df %>%
  arrange(desc(times_a_root)) %>%
  slice(1:10) %>%
  pull(variable)

formula <- as.formula(paste("SalePriceBinary ~", paste(top_features, collapse = " +
")))
log_model <- glm(formula, data = df, family = binomial())
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(log_model)
```

```
##
## Call:
## glm(formula = formula, family = binomial(), data = df)
##
## Coefficients:
```

```
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.485e+01  2.100e+01  -0.707 0.479419
## NeighborhoodBlueste -1.883e+01  4.417e+03  -0.004 0.996599
## NeighborhoodBrDale  -1.528e+01  1.600e+03  -0.010 0.992379
## NeighborhoodBrkSide  4.065e-01  1.175e+00   0.346 0.729305
## NeighborhoodClearCr  2.223e+00  1.159e+00   1.919 0.055031 .
## NeighborhoodCollgCr  3.360e-01  9.070e-01   0.370 0.711071
## NeighborhoodCrawfor  1.550e+00  1.136e+00   1.364 0.172488
## NeighborhoodEdwards -8.416e-01  1.017e+00  -0.827 0.408123
## NeighborhoodGilbert  1.671e+00  1.045e+00   1.599 0.109853
## NeighborhoodIDOTRR  -4.482e-01  1.516e+00  -0.296 0.767471
## NeighborhoodMeadowV -1.953e+01  1.163e+03  -0.017 0.986604
## NeighborhoodMitchel  1.323e+00  9.924e-01   1.333 0.182488
## NeighborhoodNAmes    5.541e-02  9.815e-01   0.056 0.954977
## NeighborhoodNoRidge  1.347e+01  8.484e+02   0.016 0.987330
## NeighborhoodNPkVill -1.770e+01  2.060e+03  -0.009 0.993144
## NeighborhoodNridgHt  1.037e+00  1.335e+00   0.777 0.437381
## NeighborhoodNWAmes   9.229e-01  9.809e-01   0.941 0.346748
## NeighborhoodOldTown -2.642e+00  1.251e+00  -2.112 0.034658 *
## NeighborhoodSawyer   2.566e-01  1.034e+00   0.248 0.803974
## NeighborhoodSawyerW  2.796e-01  1.001e+00   0.279 0.779945
## NeighborhoodSomerst  2.816e-01  9.245e-01   0.305 0.760657
## NeighborhoodStoneBr  1.519e+01  1.099e+03   0.014 0.988980
## NeighborhoodSWISU   -9.975e-01  1.265e+00  -0.788 0.430488
## NeighborhoodTimber   2.278e+00  1.237e+00   1.842 0.065460 .
## NeighborhoodVeenker  2.268e+00  1.453e+00   1.561 0.118482
## OverallQual          7.327e-01  1.677e-01   4.369 1.25e-05 ***
## YearBuilt            1.094e-02  1.044e-02   1.048 0.294566
## FullBath             4.578e-01  2.707e-01   1.691 0.090779 .
## GrLivArea            3.683e-03  4.202e-04   8.765  < 2e-16 ***
## ExterQualFa         -4.113e-01  2.601e+00  -0.158 0.874344
## ExterQualGd         -1.028e+00  1.443e+00  -0.712 0.476431
## ExterQualTA         -1.081e+00  1.423e+00  -0.760 0.447222
## BsmtQualFa           5.081e-01  1.186e+00   0.428 0.668461
## BsmtQualGd           8.585e-01  7.035e-01   1.220 0.222349
## BsmtQualMissing     -2.270e+00  1.174e+00  -1.934 0.053162 .
## BsmtQualTA           8.212e-02  7.601e-01   0.108 0.913970
## KitchenQualFa       -3.081e+00  1.329e+00  -2.319 0.020408 *
## KitchenQualGd       -9.538e-01  8.038e-01  -1.187 0.235349
## KitchenQualTA       -1.979e+00  7.969e-01  -2.483 0.013027 *
## GarageFinishMissing -2.874e+00  1.063e+00  -2.703 0.006872 **
## GarageFinishRFn     -1.481e-01  3.409e-01  -0.434 0.664035
## GarageFinishUnf     -1.244e+00  3.496e-01  -3.559 0.000372 ***
## GarageYrBlt         -7.468e-03  8.529e-03  -0.876 0.381271
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2024.0  on 1459  degrees of freedom
## Residual deviance:  590.2  on 1417  degrees of freedom
## AIC: 676.2
##
## Number of Fisher Scoring iterations: 17
```

```
#The logistic regression shows that OverallQual, GrLivArea, GarageFinishUnf, GarageFinishMissing, KitchenQualTA, and KitchenQualFa are significant predictors of high house prices (p < 0.05).
```

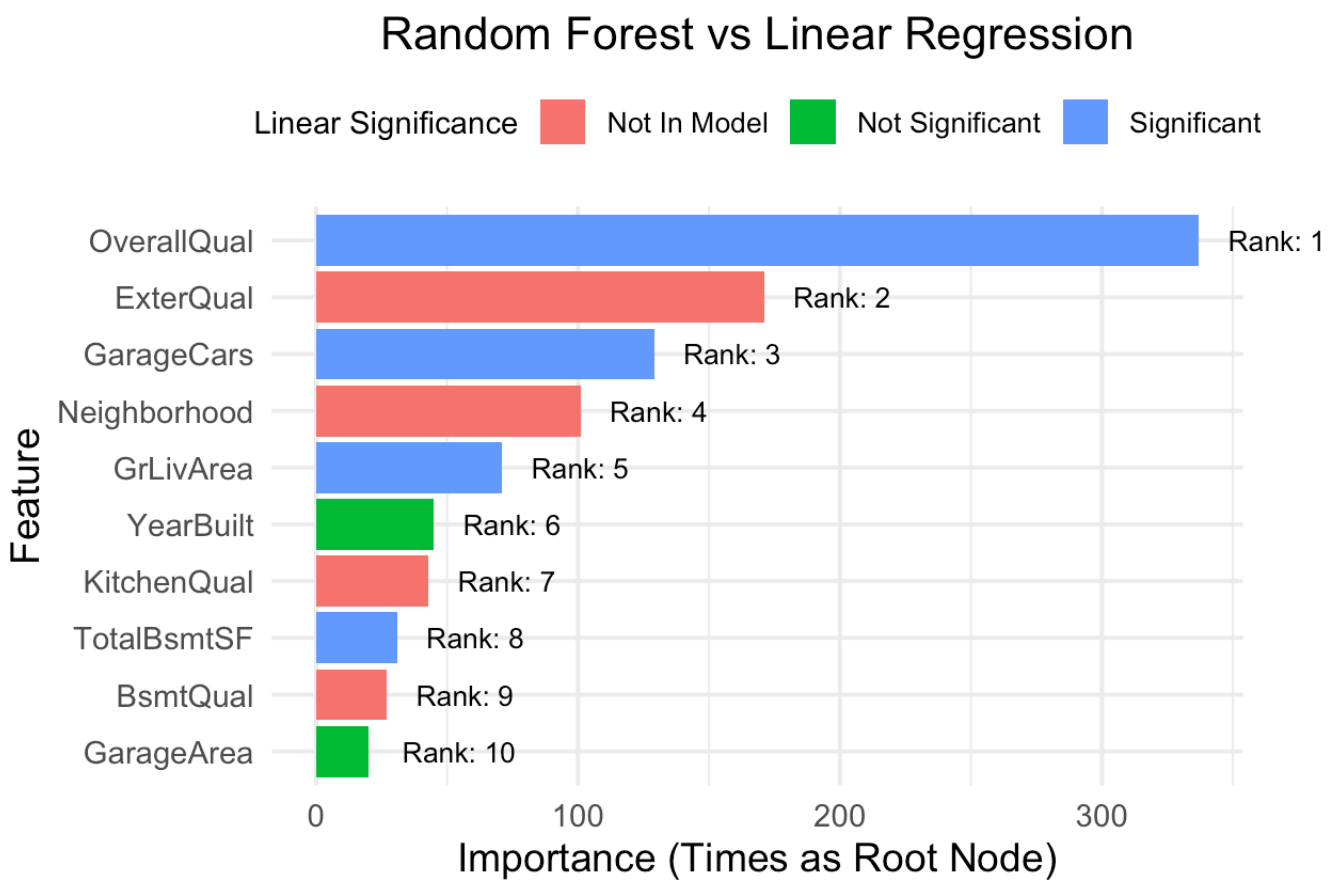
Both the random forest and logistic regression models identify OverallQual (overall quality) as the strongest predictor of house prices. The random forest further highlights ExterQual, GarageCars, and Neighborhood as key splitting variables, marking the importance of exterior condition, parking, and location.

The logistic regression confirms that OverallQual, GrLivArea (above-ground living area), and specific categories of KitchenQual and GarageFinish are statistically significant indicators of a house being in the higher price range.

Overall build quality is the most important factor, followed by interior living space, kitchen and garage quality, and location-related variables.

```
library(randomForest)
library(randomForestExplainer)
library(dplyr)
library(ggplot2)
library(broom)

df <- read.csv("~/Downloads/house-prices-advanced-regression-techniques/train.csv")
df$Id <- NULL

df <- df %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), median(., na.rm = TRUE), .))) %
>%
  mutate(across(where(is.character), ~ ifelse(is.na(.), "Missing", .))) %>%
  mutate(across(where(is.character), as.factor))

set.seed(123)
rf_model <- randomForest(SalePrice ~ ., data = df, ntree = 1000, importance = TRUE)
importance_df <- measure_importance(rf_model)

top_vars <- importance_df %>%
  arrange(desc(times_a_root)) %>%
  slice(1:10) %>%
  select(variable, times_a_root)
```

```
formula <- as.formula(paste("SalePrice ~", paste(top_vars$variable, collapse = " +
")))
lin_model <- lm(formula, data = df)

lin_summary <- tidy(lin_model) %>%
  filter(term != "(Intercept)") %>%
  mutate(Significance = ifelse(p.value < 0.05, "Significant", "Not Significant"))

comparison_df <- top_vars %>%
  rename(Variable = variable, Importance = times_a_root) %>%
  mutate(RandomForest_Rank = rank(-Importance)) %>%
  left_join(lin_summary, by = c("Variable" = "term")) %>%
  mutate(Significance = ifelse(is.na(Significance), "Not In Model", Significance))

comparison_df$Variable <- factor(comparison_df$Variable, levels = rev(comparison_df$V
ariable))

ggplot(comparison_df, aes(y = Variable, x = Importance)) +
  geom_bar(stat = "identity", aes(fill = Significance), show.legend = TRUE) +
  geom_text(aes(label = paste("Rank:", RandomForest_Rank)), hjust = -0.3, size = 3.5,
color = "black") +
  labs(
    title = "Random Forest vs Linear Regression",
    x = "Importance (Times as Root Node)",
    y = "Feature",
    fill = "Linear Significance"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.margin = unit(c(1, 1.5, 1, 1), "cm"),
    legend.position = "top",
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 11),
    axis.text = element_text(size = 11),
    plot.title = element_text(size = 16, hjust = 0.5)
  ) +
  coord_cartesian(clip = "off")
```




```
#Was among the top 10 most important in the Random Forest, but did not appear as a co
efficient in the final Linear Regression model
```