BITTIGER

课程负责人（小助手）

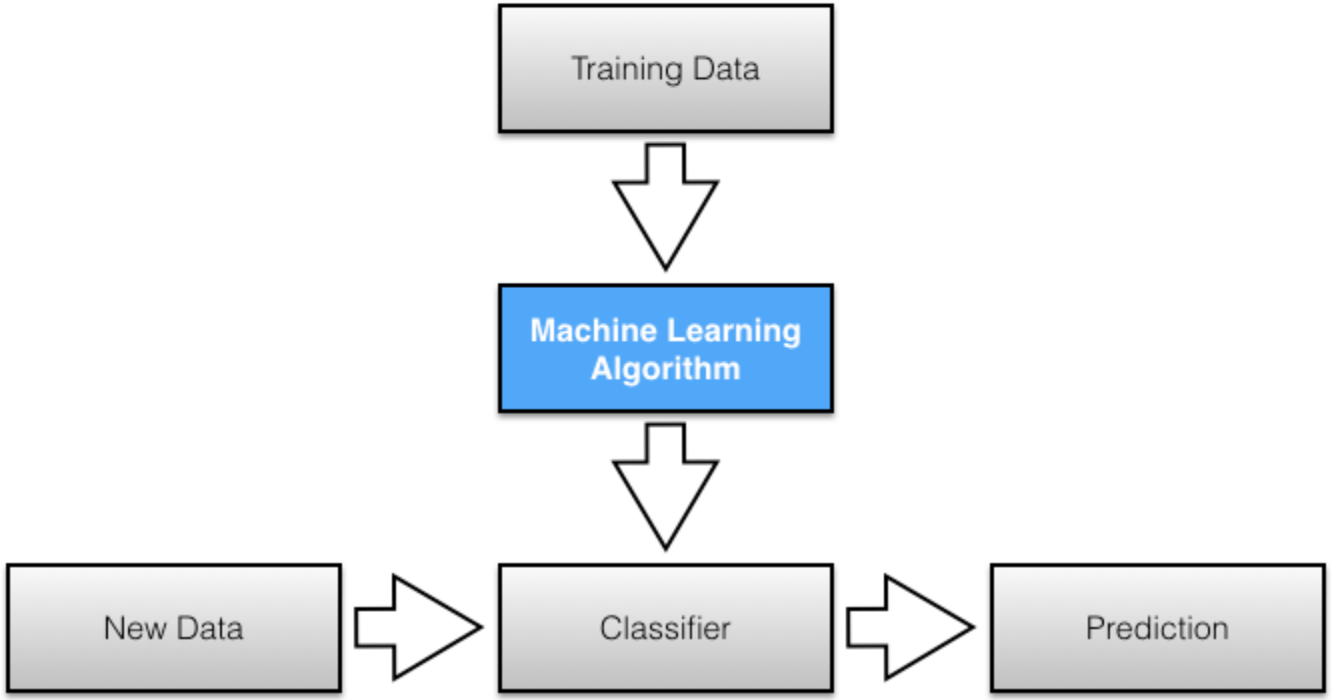
2/10 Predictive Modeling for House Price & Analytics in R

Joanne

Predictive Modeling

Supervised Learning:

build a model that makes predictions based on evidence in the presence of uncertainty.



Agenda

Regression Models & Prediction

EDA & Imputation

Linear Regression (线性回归模型)

- Multiple Linear Regression(多元线性回归)

- Model Diagnostics for Linear Regression(模型诊断)

- Interaction Terms (交互项)

- Non-linear Transformations(非线性转换)

Linear Model Selection and Regularization (变量选择和正则化)

-Best Subset/Stepwise

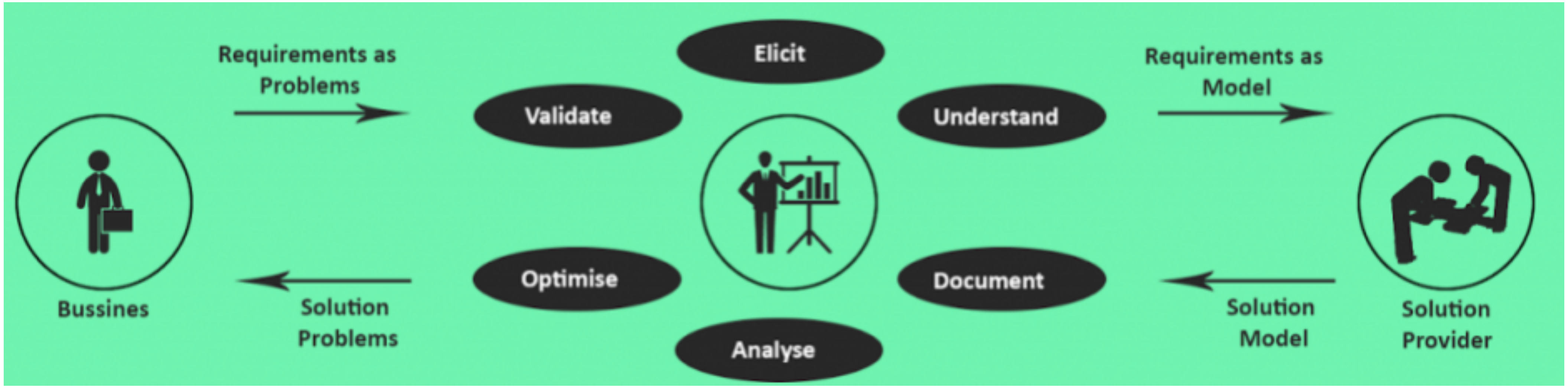
-LASSO

-RIDGE

***Regression Tree**

***Binary Logistic Regression** (逻辑回归)

Project-Process Flow



Glance through Data

"There are no routine statistical questions, only questionable statistical routines."

SalePrice Prediction-Ames,Iowa

Knowledge • 2,057 teams

House Prices: Advanced Regression Techniques

Tue 30 Aug 2016 — Wed 1 Mar 2017 (3 months to go)

Dashboard

- Home
  - Data
  - Make a submission
- Information

Competition Details » Get the Data » Make a submission

Sold! How do home features add up to its price tag?

Import Data

```
setwd("D:/.../model")
data=read.csv("train.csv",header=T,na.strings = "NA")
data2=read.csv("test.csv",header=T,na.strings = "NA")

# remove ID
data=data[,-c(1)]

summary(data)
str(data)
```

```
> summary(data)
   MSSubClass       MSZoning     LotFrontage        LotArea        Street      Alley     
 Min.   : 20.0   C (all):  10   Min.   : 21.00   Min.   :  1300   Grvl:   6   Grvl:  50  
 1st Qu.: 20.0   FV     :  65   1st Qu.: 59.00   1st Qu.:  7554   Pave:1454   Pave:  41  
 Median : 50.0   RH     :  16   Median : 69.00   Median :  9478               NA's:1369  
 Mean   : 56.9   RL     :1151   Mean   : 70.05   Mean   : 10517                          
 3rd Qu.: 70.0   RM     : 218   3rd Qu.: 80.00   3rd Qu.: 11602                          
 Max.   :190.0                  Max.   :313.00   Max.   :215245                          
                                NA's   :259                                              
```

Exploratory Data Analysis

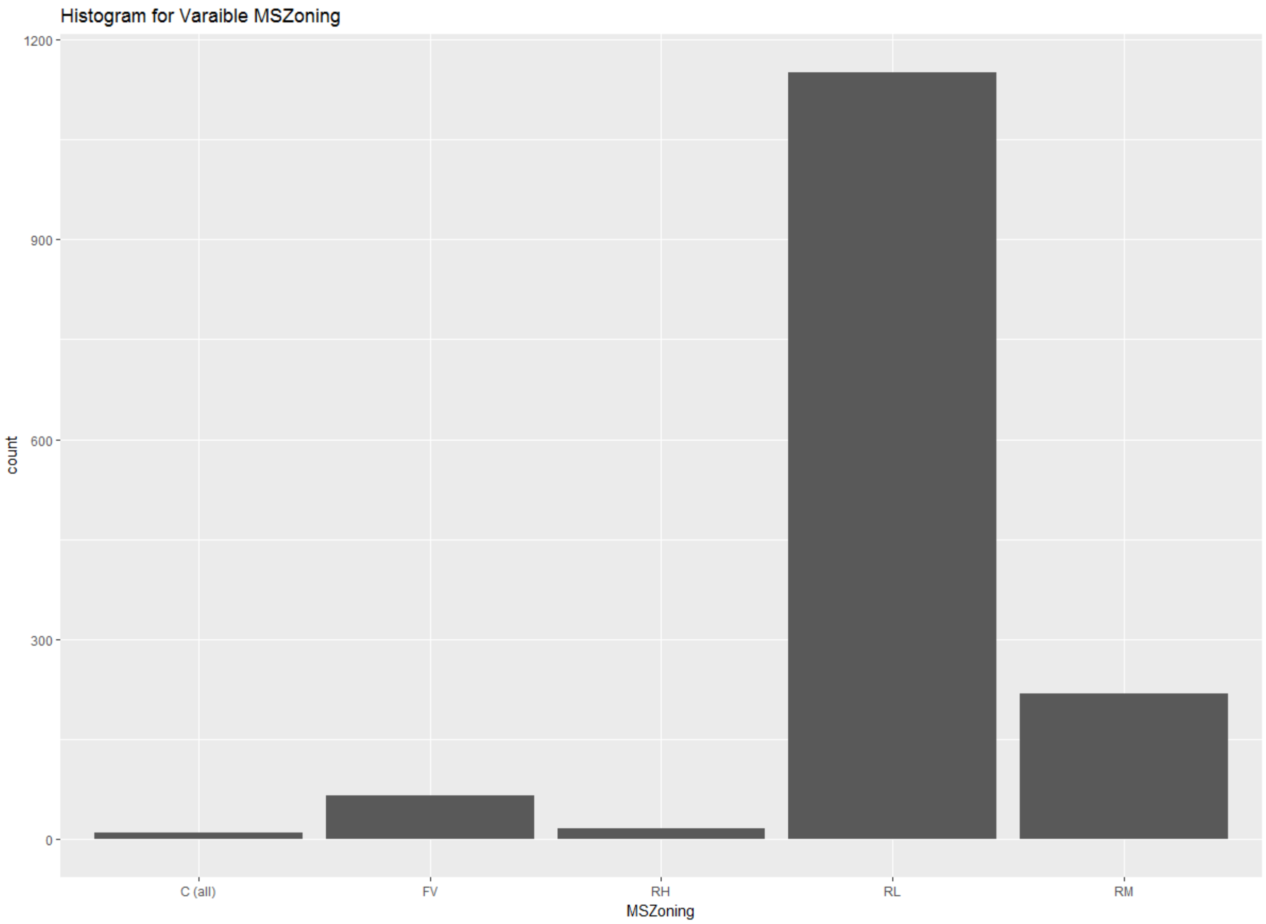
```
                                                                        (Other):    9
> str(data)
'data.frame':	1460 obs. of  80 variables:
 $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
 $ MSZoning     : Factor w/ 5 levels "C (all)","FV",..: 4 4 4 4 4 4 4 4 5 4 ...
 $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
 $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
 $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
 $ Alley        : Factor w/ 2 levels "Grvl","Pave": NA NA NA NA NA NA NA NA NA NA ...
 $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 4 4 1 1 1 1 4 1 4 4 ...
 $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ Utilities    : Factor w/ 2 levels "AllPub","NoSeWa": 1 1 1 1 1 1 1 1 1 1 ...
 $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...
 $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
 $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14 12 21 17 18 4 ...
 $ Condition1   : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 5 1 1 ...
 $ Condition2   : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 1 ...
 $ BldgType     : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 1 1 1 2 ...
```

optional: data$MSSubClass=as.factor(data$MSSubClass)

Exploratory Data Analysis

```
> ggplot(data = data) +
  geom_bar(mapping = aes(x = MSZoning )) # bar for categorical
+   ggtitle("Histogram for Varaible MSZoning")
```

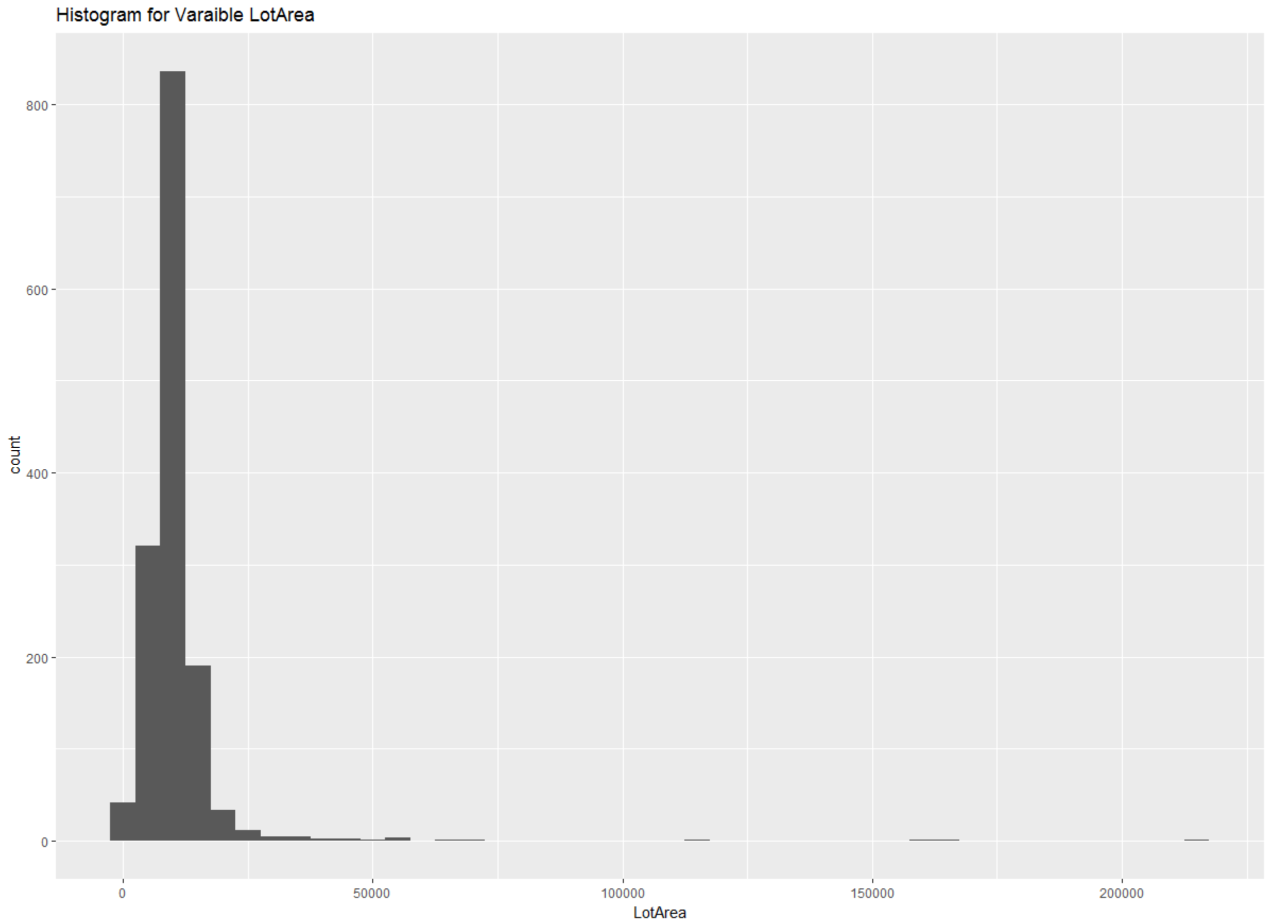
```
MSZoning: Identifies the general zoning classification of the sale.

       A        Agriculture
       C        Commercial
       FV       Floating Village Residential
       I        Industrial
       RH       Residential High Density
       RL       Residential Low Density
       RP       Residential Low Density Park
       RM       Residential Medium Density
```



Exploratory Data Analysis

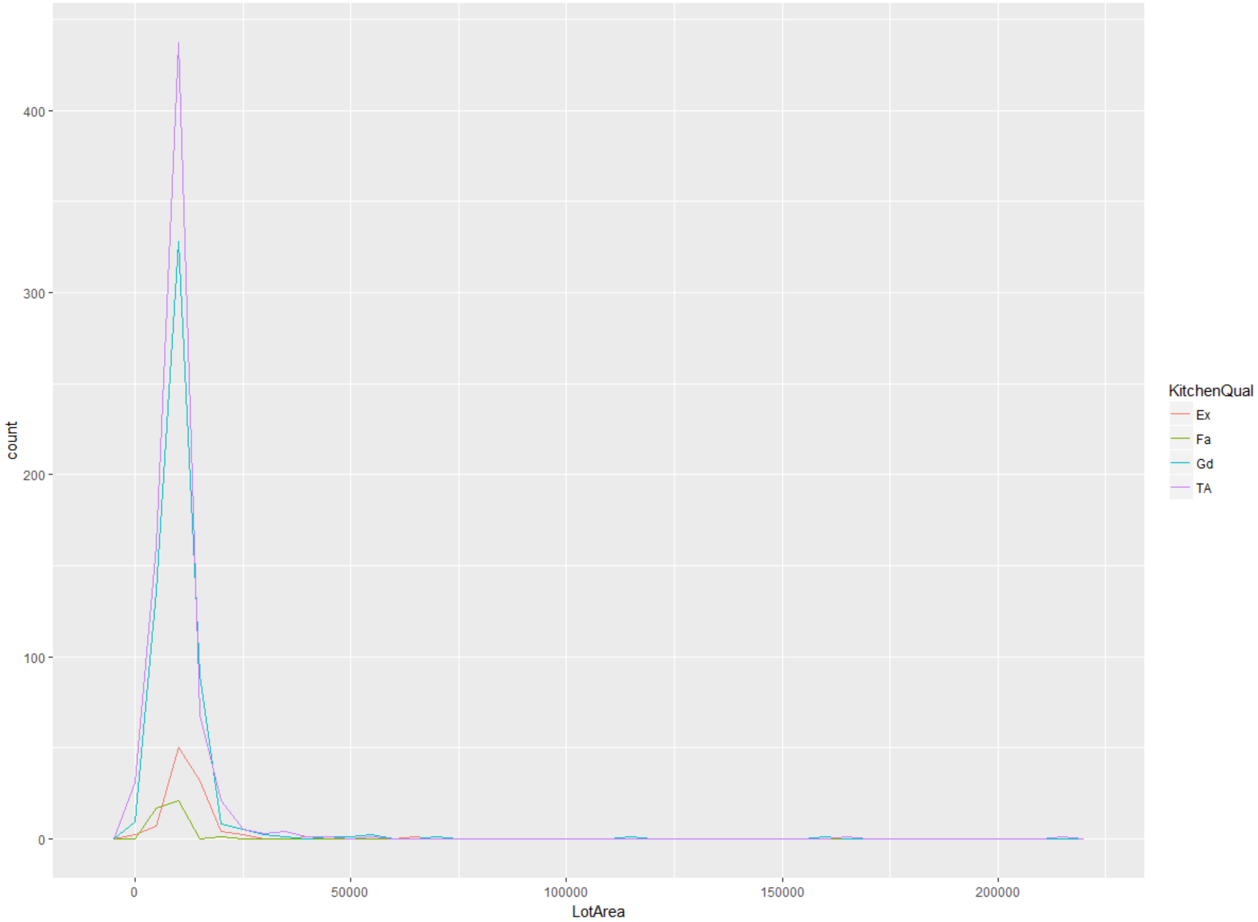
```
> summary(data$LotArea) # to determine binwidth
# LotArea: Lot size in square feet

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1300    7554    9478   10520   11600  215200
```

```
> ggplot(data = data) +
geom_histogram(mapping = aes(x = LotArea), binwidth =5000)
# histogram for continuous
+   ggtitle("Histogram for Varaible LotArea")
```



Exploratory Data Analysis

```
# visualize a categorical and a continuous variable
ggplot(data = data, mapping = aes(x = LotArea, colour =
KitchenQual)) +
  geom_freqpoly(binwidth = 5000)
```



Ex Excellent Gd Good TA Average/Typical Fa Fair

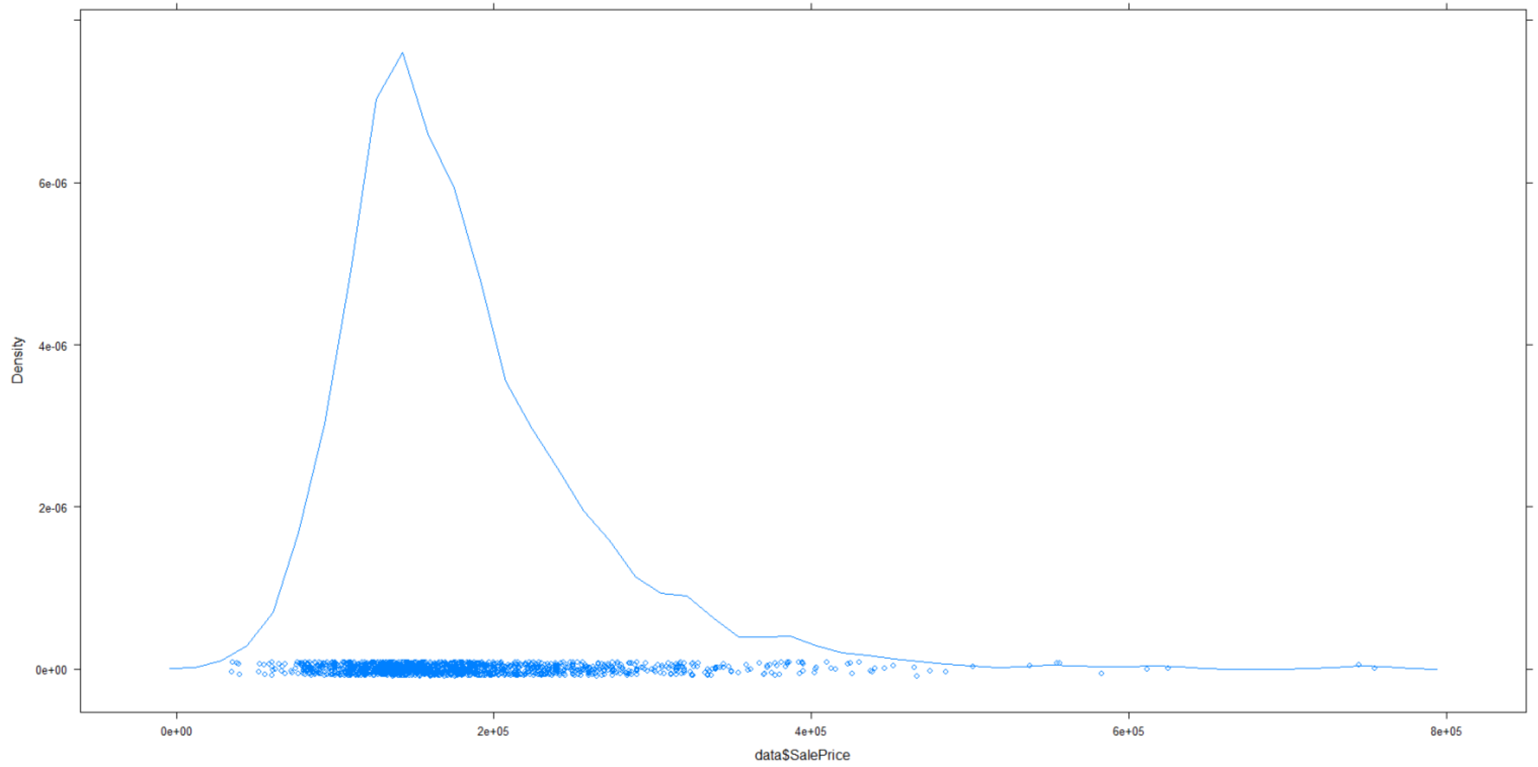
Exploratory Data Analysis- dplyr

```
> library(dplyr)
> data %>% count(MSZoning)
# A tibble: 5 × 2
  MSZoning     n
    <fctr> <int>
1  C (all)    10
2       FV    65
3       RH    16
4       RL  1151
5       RM   218
```

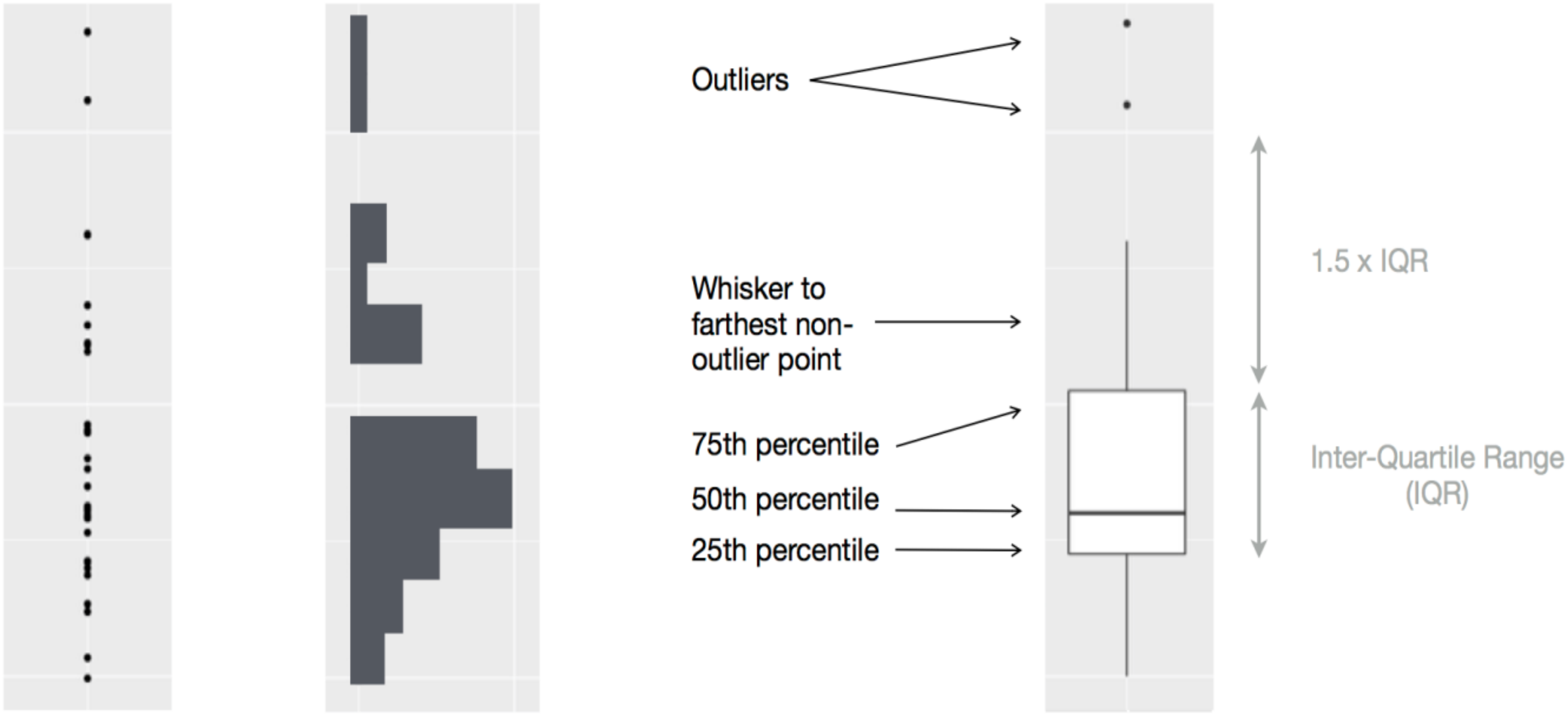
```
> data %>% count(cut_width(LotArea, 5000))
# A tibble: 18 × 2
   `cut_width(LotArea, 5000)`     n
                    <fctr> <int>
1         [-2.5e+03,2.5e+03]    42
2          (2.5e+03,7.5e+03]   321
3         (7.5e+03,1.25e+04]   836
4        (1.25e+04,1.75e+04]   190
5        (1.75e+04,2.25e+04]    34
6        (2.25e+04,2.75e+04]    12
7        (2.75e+04,3.25e+04]     5
8        (3.25e+04,3.75e+04]     5
9        (3.75e+04,4.25e+04]     2
10       (4.25e+04,4.75e+04]     2
11       (4.75e+04,5.25e+04]     1
12       (5.25e+04,5.75e+04]     4
13       (6.25e+04,6.75e+04]     1
14       (6.75e+04,7.25e+04]     1
15       (1.12e+05,1.18e+05]     1
16       (1.58e+05,1.62e+05]     1
17       (1.62e+05,1.68e+05]     1
18       (2.12e+05,2.18e+05]     1
```

Better Understand Y variable

```
library(lattice)
densityplot(data$SalePrice)
```



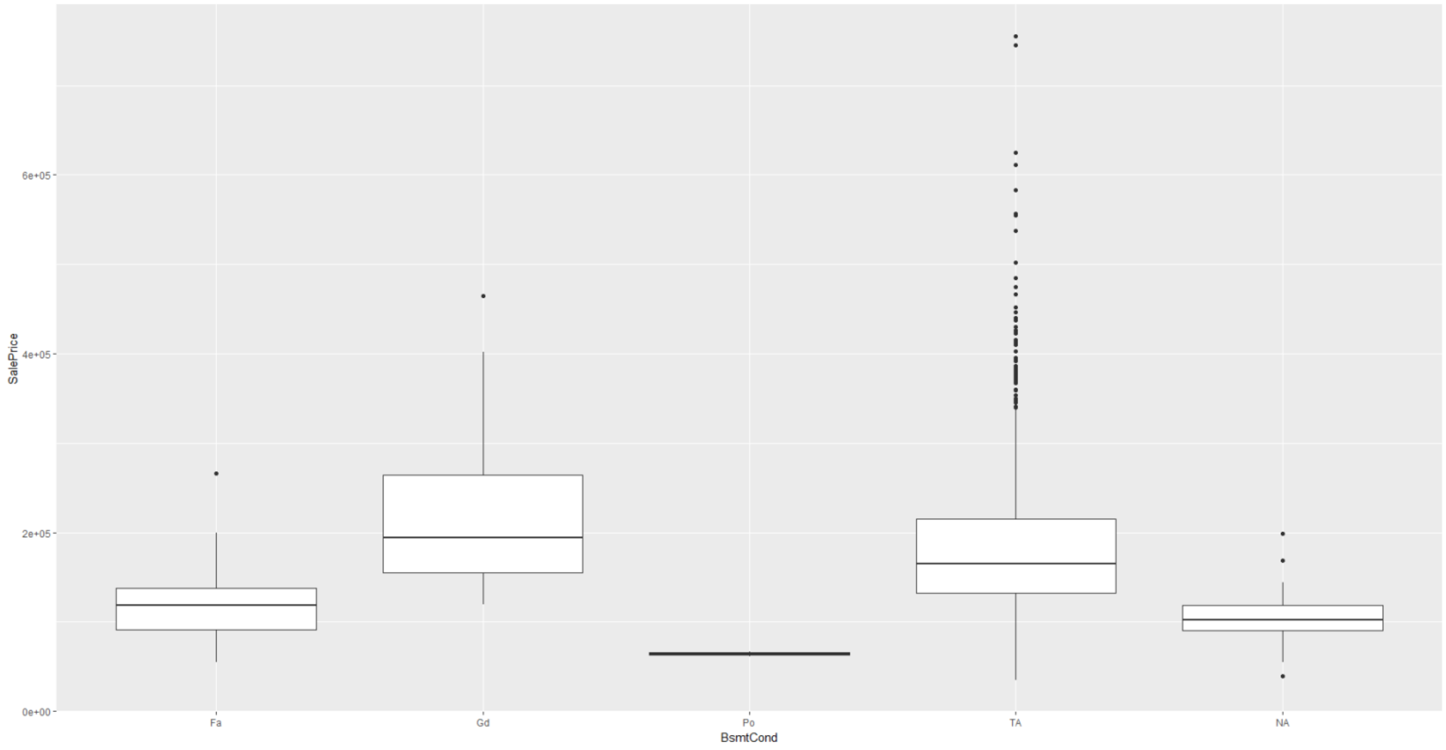
BoxPlot



Outliers
Whisker to farthest non-outlier point
75th percentile
50th percentile
25th percentile
1.5 x IQR
Inter-Quartile Range (IQR)

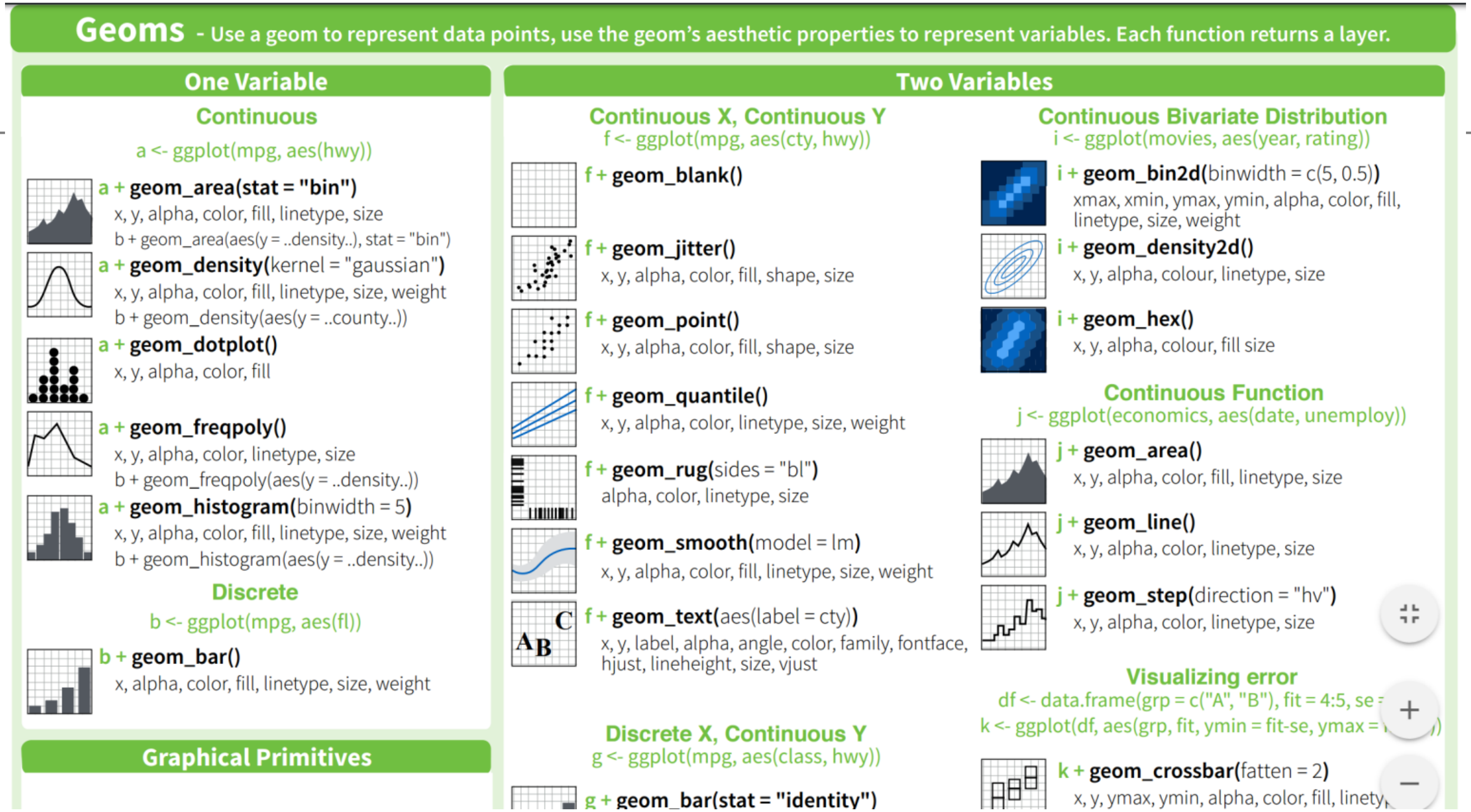
Exploratory Data Analysis

```
ggplot(data = data, mapping = aes(x =
BsmtCond , y = SalePrice)) +
  geom_boxplot()
```




```
BsmtCond: Evaluates the general condition of the basement

       Ex        Excellent
       Gd        Good
       TA        Typical - slight dampness allowed
       Fa        Fair - dampness or some cracking or settling
       Po        Poor - Severe cracking, settling, or wetness
       NA        No Basement
```

R for Data Visualization: ggplot2



Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

One Variable

Continuous

a <- ggplot(mpg, aes(hwy))

a + **geom_area(stat = "bin")**
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")

a + **geom_density(**kernel = "gaussian"**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))

a + **geom_dotplot()**
x, y, alpha, color, fill

a + **geom_freqpoly()**
x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))

a + **geom_histogram(**binwidth = 5**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))

Discrete

b <- ggplot(mpg, aes(fl))

b + **geom_bar()**
x, alpha, color, fill, linetype, size, weight

Graphical Primitives

Two Variables

Continuous X, Continuous Y

f <- ggplot(mpg, aes(cty, hwy))

f + **geom_blank()**

f + **geom_jitter()**
x, y, alpha, color, fill, shape, size

f + **geom_point()**
x, y, alpha, color, fill, shape, size

f + **geom_quantile()**
x, y, alpha, color, linetype, size, weight

f + **geom_rug(**sides = "bl"**)**
alpha, color, linetype, size

f + **geom_smooth(**model = lm**)**
x, y, alpha, color, fill, linetype, size, weight

f + **geom_text(**aes(label = cty)**)**
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Discrete X, Continuous Y

g <- ggplot(mpg, aes(class, hwy))

g + **geom_bar(stat = "identity")**

Continuous Bivariate Distribution

i <- ggplot(movies, aes(year, rating))

i + **geom_bin2d(**binwidth = c(5, 0.5)**)**
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight

i + **geom_density2d()**
x, y, alpha, colour, linetype, size

i + **geom_hex()**
x, y, alpha, colour, fill size

Continuous Function

j <- ggplot(economics, aes(date, unemploy))

j + **geom_area()**
x, y, alpha, color, fill, linetype, size

j + **geom_line()**
x, y, alpha, color, linetype, size

j + **geom_step(**direction = "hv"**)**
x, y, alpha, color, linetype, size

Visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se =

k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax =

k + **geom_crossbar(**fatten = 2**)**
x, y, ymax, ymin, alpha, color, fill, linety

Cheat Sheet:
https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

Clean Data: Imputation

Check % of Missing Data

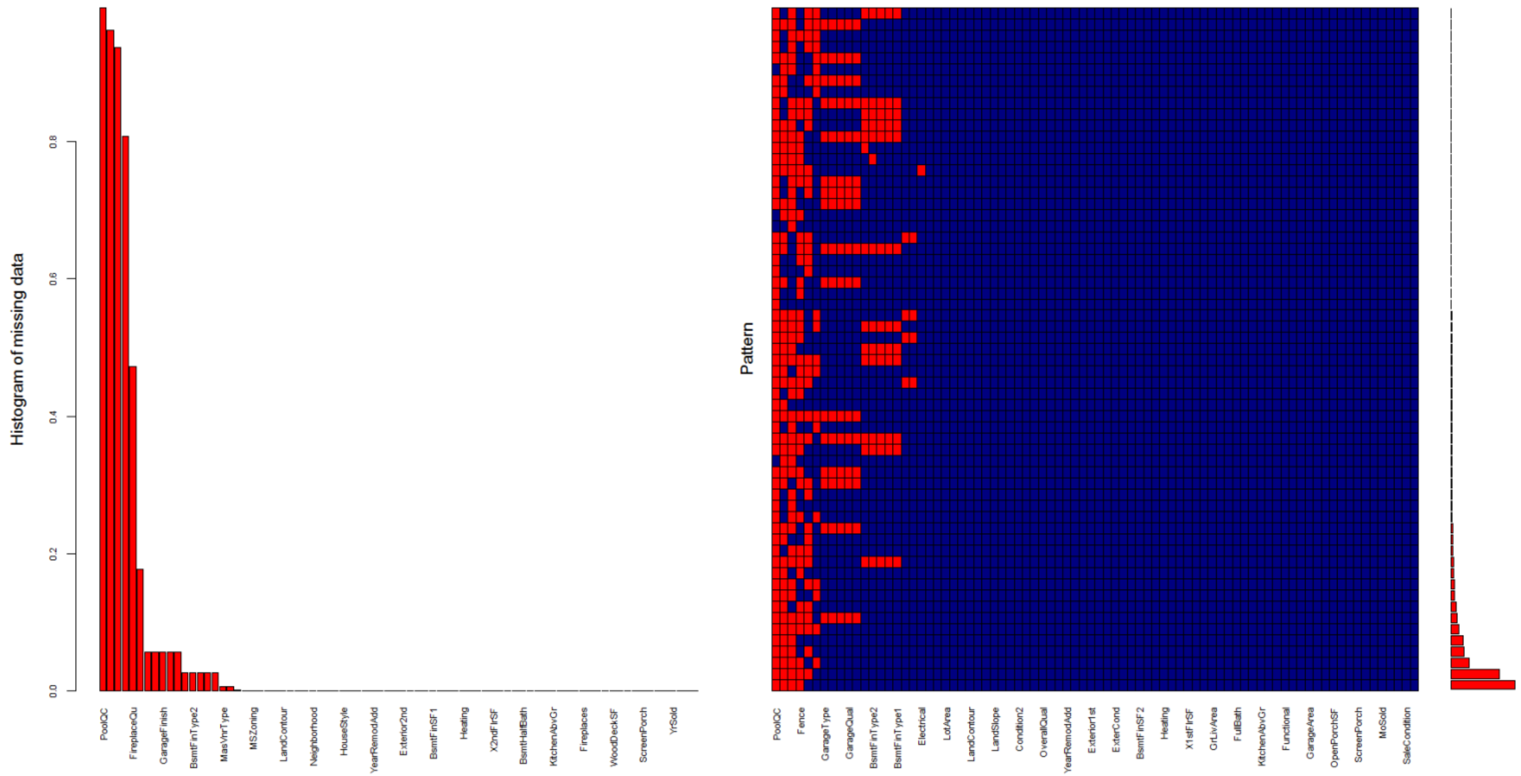
```
> MissingPercentage <- function(x){sum(is.na(x))/length(x)*100}
> sort(apply(data,2,MissingPercentage),decreasing=TRUE)
      PoolQC  MiscFeature        Alley        Fence  FireplaceQu  LotFrontage   GarageType  GarageYrBlt GarageFinish
 99.52054795  96.30136986  93.76712329  80.75342466  47.26027397  17.73972603   5.54794521   5.54794521   5.54794521
  GarageQual   GarageCond BsmtExposure BsmtFinType2     BsmtQual     BsmtCond BsmtFinType1   MasVnrType   MasVnrArea
  5.54794521   5.54794521   2.60273973   2.60273973   2.53424658   2.53424658   2.53424658   0.54794521   0.54794521
  Electrical   MSSubClass     MSZoning      LotArea       Street     LotShape  LandContour    Utilities    LotConfig
  0.06849315   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
   LandSlope Neighborhood   Condition1   Condition2     BldgType   HouseStyle  OverallQual  OverallCond    YearBuilt
  0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
YearRemodAdd    RoofStyle     RoofMatl  Exterior1st  Exterior2nd    ExterQual    ExterCond   Foundation   BsmtFinSF1
  0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
  BsmtFinSF2    BsmtUnfSF  TotalBsmtSF      Heating    HeatingQC   CentralAir    X1stFlrSF    X2ndFlrSF LowQualFinSF
  0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
   GrLivArea BsmtFullBath BsmtHalfBath     FullBath     HalfBath BedroomAbvGr KitchenAbvGr  KitchenQual TotRmsAbvGrd
  0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
  Functional   Fireplaces   GarageCars   GarageArea   PavedDrive   WoodDeckSF  OpenPorchSF EnclosedPorch  X3SsnPorch
  0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
```

Check # of Missing Data

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

Visualizing Missing Data and Delete

```
library(VIM)
aggr_plot <- aggr(data,
            col=c('navyblue','red'),
            numbers=TRUE,
            sortVars=TRUE,
            labels=names(data),
            cex.axis=.7,
            gap=3,
            ylab=c("Histogram of missing data","Pattern"))
```



Delete Columns with more than 5% Missig Data and Imputing the Rest

Assumption:

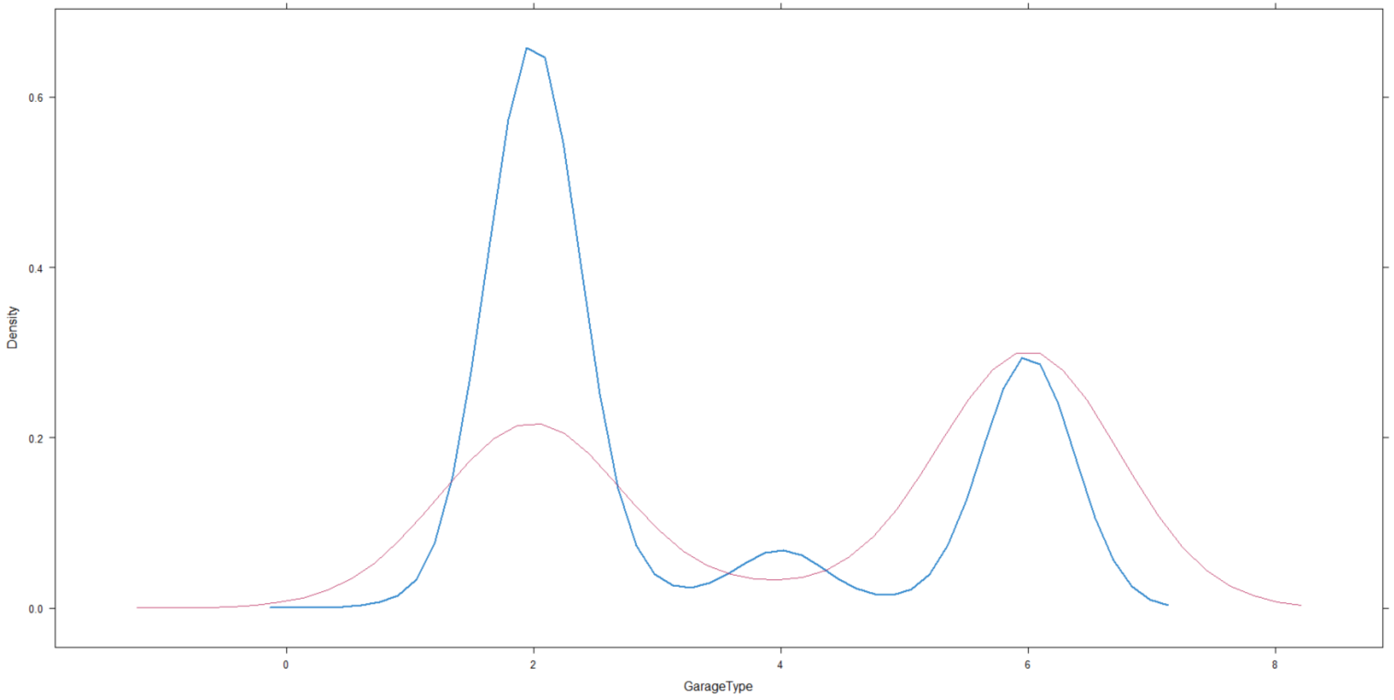
MCAR: missing completely at random.

```
# Delete columns with more than 5% missing data
library(dplyr)
data=select(data,-c(PoolQC,MiscFeature,Alley,Fence,FireplaceQu,LotFrontage))
```

```
# CART: classification and regression trees
library(mice)
imp_data<- mice(data, m=1, method='cart', printFlag=FALSE)
```

Test Result

```
> # Test Original and Imputed
> table(data$GarageType)

 2Types  Attchd Basment BuiltIn CarPort  Detchd
      6     870      19      88       9     387
> table(imp_data$imp$GarageType)

 2Types  Attchd BuiltIn  Detchd
      1      32       3      45
> # vasualize density blue:actual; red:imputed
> densityplot(imp_data, ~GarageType)
```



Imputing Done! Double Check!

```
# Merge to Original Data
data_complete <- complete(imp_data)

#Confirm no NAs
sum(sapply(data_complete, function(x) { sum(is.na(x)) }))

write.csv(data_complete, file = "data_complete.csv")
data_complete=read.csv("data_complete.csv",header=T)
```

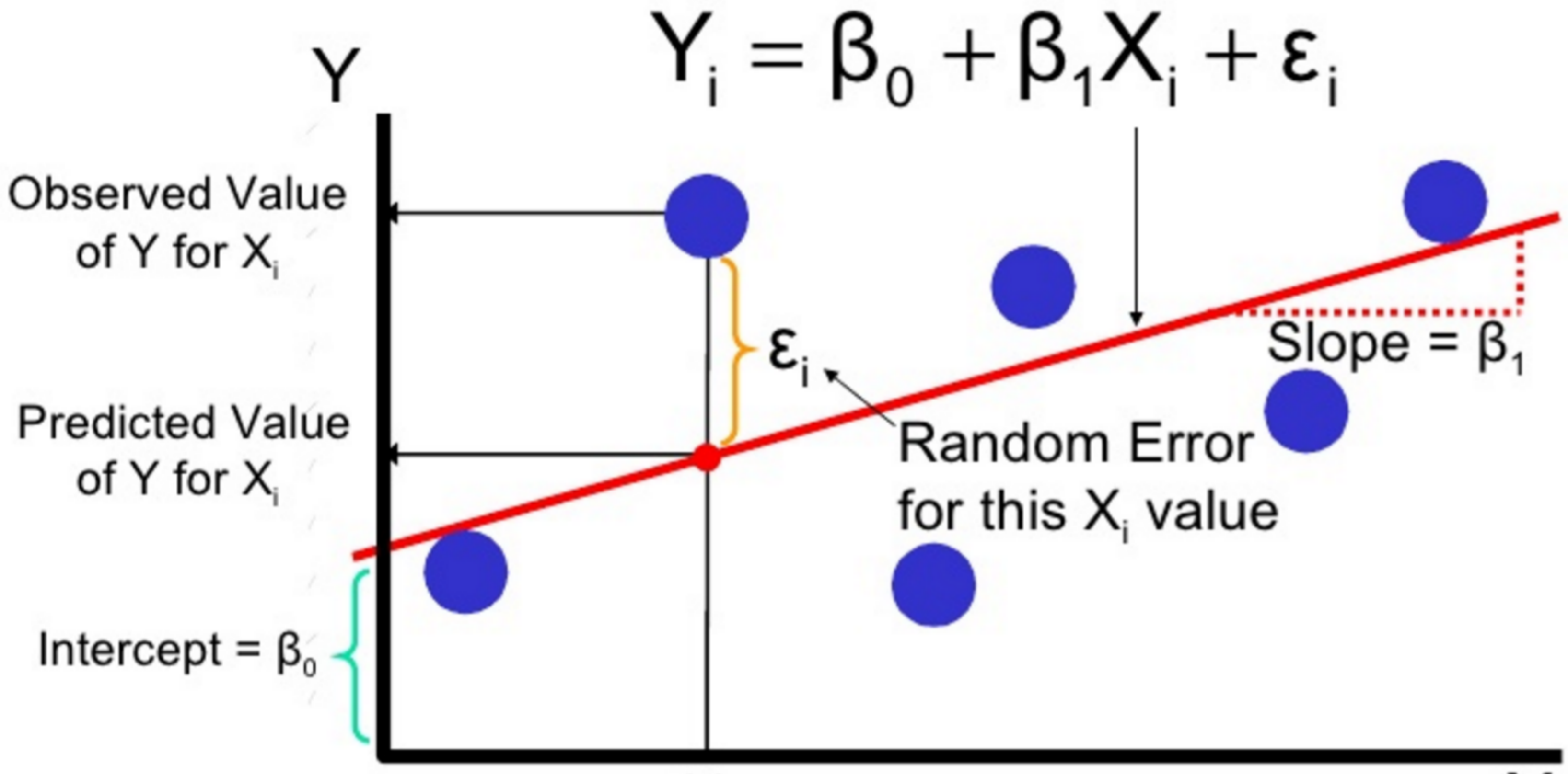
Multiple Linear Regression Model

Training and Testing Set

```
> set.seed(1)
> train = sample(1:nrow(data_complete),nrow(data_complete)/2)
> test = -train
> traindata = data_complete[train,]
> testdata = data_complete[test,]
```

testdata	730 obs. of 74 variables
traindata	730 obs. of 74 variables

Linear Regression



- *x*, is regarded as the **predictor**, **explanatory**, or **independent** variable.

- *y*, is regarded as the **response**, **outcome**, or **dependent** variable.

- Residual: The difference between an observed value of the dependent variable and the value of the dependent variable predicted from the regression line.

Fit A Model...Oops

```
> model=lm(SalePrice~.,data=traindata)
Warning messages:
1: contrasts dropped from factor Condition1 due to missing levels
2: contrasts dropped from factor Condition2 due to missing levels
3: contrasts dropped from factor RoofStyle due to missing levels
4: contrasts dropped from factor RoofMatl due to missing levels
5: contrasts dropped from factor Exterior1st due to missing levels
6: contrasts dropped from factor Exterior2nd due to missing levels
7: contrasts dropped from factor Heating due to missing levels
8: contrasts dropped from factor Functional due to missing levels
> distinct(data,RoofStyle)
  RoofStyle
1     Gable
2       Hip
3   Gambrel
4   Mansard
5      Flat
6      Shed
> distinct(traindata,RoofStyle)
  RoofStyle
1       Hip
2     Gable
3   Mansard
4      Flat
5   Gambrel
```

Understand the Model

```
> summary(model)

Call:
lm(formula = SalePrice ~ ., data = traindata)

Residuals:
    Min      1Q  Median      3Q     Max
-107928   -8614       0    9346  133886

Coefficients: (4 not defined because of singularities)
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        -1.515e+06  1.534e+06  -0.988 0.32380
MSSubClass         -1.612e+02  1.095e+02  -1.473 0.14144
MSZoning2           3.730e+04  1.775e+04   2.101 0.03615  *
MSZoning3           3.173e+04  1.883e+04   1.685 0.09260  .
MSZoning4           1.821e+04  1.539e+04   1.183 0.23743
MSZoning5           1.388e+04  1.465e+04   0.947 0.34392
LotArea             1.041e+00  1.612e-01   6.457 2.47e-1  ***
Street2             4.276e+04  1.991e+04   2.148 0.03218  *
LotShape2          -5.435e+03  6.616e+03  -0.821 0.41179
LotShape3          -1.367e+04  1.260e+04  -1.085 0.27863
LotShape4           3.276e+02  2.267e+03   0.145 0.88515
```

```
GarageCond5                NA         NA      NA       NA
PavedDrive2        -4.131e+03  9.145e+03  -0.452 0.651675
PavedDrive3        -3.067e+03  4.865e+03  -0.630 0.528707
WoodDeckSF          1.125e+01  8.841e+00   1.272 0.203790
OpenPorchSF        -1.118e+01  1.867e+01  -0.599 0.549485
EnclosedPorch      -1.328e+01  1.674e+01  -0.793 0.427997
X3SsnPorch          4.678e+01  3.813e+01   1.227 0.220410
ScreenPorch         1.709e+01  1.893e+01   0.903 0.367039
PoolArea            9.805e+01  2.284e+01   4.294 2.10e-05 ***
MiscVal             1.012e+00  1.452e+00   0.697 0.486105
MoSold              5.118e+02  3.653e+02   1.401 0.161853
YrSold              4.140e+02  7.557e+02   0.548 0.584041
SaleType2           3.966e+04  2.589e+04   1.532 0.126195
SaleType3           3.849e+04  2.053e+04   1.875 0.061373 .
SaleType4          -3.455e+03  1.616e+04  -0.214 0.830725
SaleType5          -8.056e+03  1.657e+04  -0.486 0.627019
SaleType6          -5.967e+04  3.107e+04  -1.920 0.055374 .
SaleType7          -2.875e+03  2.489e+04  -0.115 0.908098
SaleType8           1.414e+04  1.504e+04   0.940 0.347463
SaleType9          -2.739e+03  6.421e+03  -0.427 0.669828
SaleCondition2      3.815e+03  1.980e+04   0.193 0.847301
SaleCondition3      9.637e+03  1.307e+04   0.738 0.461083
SaleCondition4      2.132e+04  8.512e+03   2.504 0.012577 *
SaleCondition5      8.947e+03  4.326e+03   2.068 0.039127 *
SaleCondition6      2.114e+04  2.395e+04   0.883 0.377846
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21430 on 513 degrees of freedom
Multiple R-squared:  0.9518,    Adjusted R-squared:  0.9314
F-statistic: 46.86 on 216 and 513 DF,  p-value: < 2.2e-16
```

Calculate RMSE

```
> # Calculate Root Mean Squared Error
> RMSE <- sqrt(mean(model$residuals^2))
> RMSE
[1] 17965.74
```

Root Mean Squared Error (RMSE)

The square root of the mean/average of the square of all of the error.

The use of RMSE is very common and it makes an excellent general purpose error metric for numerical predictions.

Compared to the similar Mean Absolute Error, RMSE amplifies and severely punishes large errors.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Manually Select Variables (subset)

```
> #NA as a coefficient in a regression indicates that the variable in question is linearly related to the other variables.
> #a.k.a collinearity
> model=lm(SalePrice~
+            LotArea+OverallQual+OverallCond+YearBuilt+BsmtQual+BsmtFinSF1+
+            BsmtFinSF2+BsmtUnfSF+X1stFlrSF+X2ndFlrSF+BedroomAbvGr+
+            KitchenAbvGr+KitchenQual+GarageCars+PoolArea,
+          data=traindata)
> summary(model)

Call:
lm(formula = SalePrice ~ LotArea + OverallQual + OverallCond +
    YearBuilt + BsmtQual + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF +
    X1stFlrSF + X2ndFlrSF + BedroomAbvGr + KitchenAbvGr + KitchenQual +
    GarageCars + PoolArea, data = traindata)

Residuals:
    Min      1Q  Median      3Q     Max
-139861  -13659      97   13871  165297
```

RMSE INCREASED!

```
Call:
lm(formula = SalePrice ~ LotArea + OverallQual + OverallCond +
    YearBuilt + BsmtQual + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF +
    X1stFlrSF + X2ndFlrSF + BedroomAbvGr + KitchenAbvGr + KitchenQual +
    GarageCars + PoolArea, data = traindata)

Residuals:
    Min      1Q  Median      3Q     Max
-139861  -13659      97   13871  165297

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -8.501e+05  1.161e+05  -7.323 6.62e-13 ***
LotArea       6.085e-01  9.170e-02   6.636 6.42e-11 ***
OverallQual   1.070e+04  1.328e+03   8.054 3.38e-15 ***
OverallCond   5.608e+03  9.965e+02   5.628 2.62e-08 ***
YearBuilt     4.422e+02  5.781e+01   7.649 6.59e-14 ***
BsmtQual2    -2.068e+04  8.152e+03  -2.537 0.011396 *
BsmtQual3    -3.138e+04  4.374e+03  -7.175 1.83e-12 ***
BsmtQual4    -2.898e+04  5.574e+03  -5.199 2.62e-07 ***
BsmtFinSF1    5.046e+01  5.062e+00   9.969  < 2e-16 ***
BsmtFinSF2    3.020e+01  7.761e+00   3.891 0.000109 ***
BsmtUnfSF     2.568e+01  4.855e+00   5.289 1.64e-07 ***
X1stFlrSF     7.561e+01  5.492e+00  13.767  < 2e-16 ***
X2ndFlrSF     7.466e+01  3.611e+00  20.676  < 2e-16 ***
BedroomAbvGr -9.486e+03  1.619e+03  -5.859 7.14e-09 ***
KitchenAbvGr -1.764e+04  4.578e+03  -3.854 0.000127 ***
KitchenQual2 -2.775e+04  8.217e+03  -3.378 0.000771 ***
KitchenQual3 -3.469e+04  4.610e+03  -7.526 1.59e-13 ***
KitchenQual4 -3.580e+04  5.134e+03  -6.974 7.08e-12 ***
GarageCars    6.566e+03  1.814e+03   3.620 0.000316 ***
PoolArea      5.960e+01  2.368e+01   2.517 0.012045 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26570 on 710 degrees of freedom
Multiple R-squared:  0.8974,	Adjusted R-squared:  0.8946
F-statistic: 326.8 on 19 and 710 DF,  p-value: < 2.2e-16

> RMSE <- sqrt(mean(model$residuals^2))
> RMSE
[1] 26202.13
```

Make Prediction & RMSE for Test Set

```
# make prediction based on test set
predict_model= predict(model,testdata)
head(predict_model) #prediction results
head(testdata$SalePrice) # vs. actual Saleprice
```

```
> head(predict_model) #prediction results
       1        3        5        6        7        9 
212486.2 215267.0 265279.9 181560.2 293183.8 163363.9 
> head(testdata$SalePrice) # vs. actual Saleprice
[1] 208500 223500 250000 143000 307000 129900
```

```
# calculate the value of R-squared for the prediction model on the test data set as follows:
SSE <- sum((testdata$SalePrice - predict_model) ^ 2)
SST <- sum((testdata$SalePrice - mean(testdata$SalePrice)) ^ 2)
1 - SSE/SST

[1] 0.6781038
```

```
> # testset RMSE compare to traindata
> testRMSE <- sqrt(mean((predict_model - testdata$SalePrice)^2))
> testRMSE
[1] 43648.79
> trainRMSE <- sqrt(mean(model$residuals^2))
> trainRMSE
[1] 26230.96
```

Diagnostic Plots & Linear Regression Assumptions

```
par(mfrow=c(2,2))
plot(model)
```

Assumptions:

(i) linearity

(ii) Normality of the error distribution

(iii) statistical independence of the errors (No or little Autocorrelation)

(iv) homoscedasticity (constant variance) of the errors

+ No or litter **Multicollinearity**