امیر سالاری تلمادره - ۶۱۰۳۹۷۱۱۹

پروژه نهایی روش های آماری

تفسیر ستون ها

MSSubClass: کلاس ساختمان MSZoning: طبقه بندی عمومی منطقه بندی LotArea: اندازه لات در فوت مربع
SaleCondition: شرایط فروش SalePrice: متغیر وابسته

In [1]:
```
data = read.csv("train.csv")
head(data)
```

A data.frame: 6 × 6

	Id	MSSubClass	MSZoning	LotArea	SaleCondition	SalePrice
	<int>	<int>	<chr>	<int>	<chr>	<int>
1	1	60	RL	8450	Normal	208500
2	2	20	RL	9600	Normal	181500
3	3	60	RL	11250	Normal	223500
4	4	70	RL	9550	Abnorml	140000
5	5	60	RL	14260	Normal	250000
6	6	50	RL	14115	Normal	143000

تفسیر نوع داده ستون ها

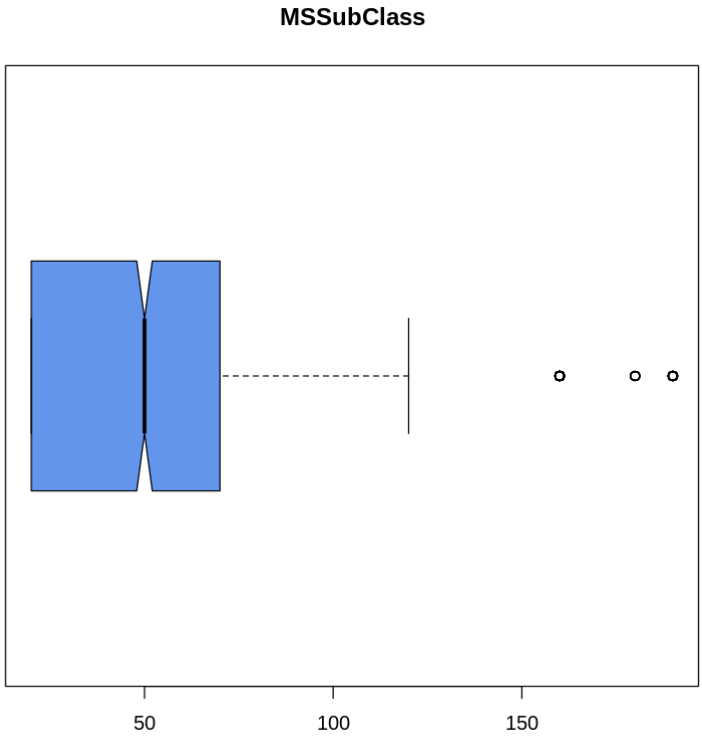
همانطور که در جدول بالا که فقط ۱۰ سطر اول داده ها در آن نشان داده شدهاست مشخص است ستون ها به دو دسته عددی و کاراکتری تقسیم شده اند: ستون های عددی: Line_No Id MSSubClass LotArea SalePrice ستون های کاراکتری:
MSZoning SaleCondition

تفسیر کلی ستون های عددی:

در زیر تفسیری کلی از داده ها قرار داده شده است که عبارت است از: مینیمم, چارک اول, میانه, میانگین, چارک سوم, ماکسیمم

In [2]:
```
print("MSSubClass")
summary(data$MSSubClass)
boxplot(data$MSSubClass,
    main = "MSSubClass",
    col = "cornflowerblue",
    border = "black",
    horizontal = TRUE,
    notch = TRUE
)
```

```
[1] "MSSubClass"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   20.0    20.0    50.0    56.9    70.0   190.0 
```

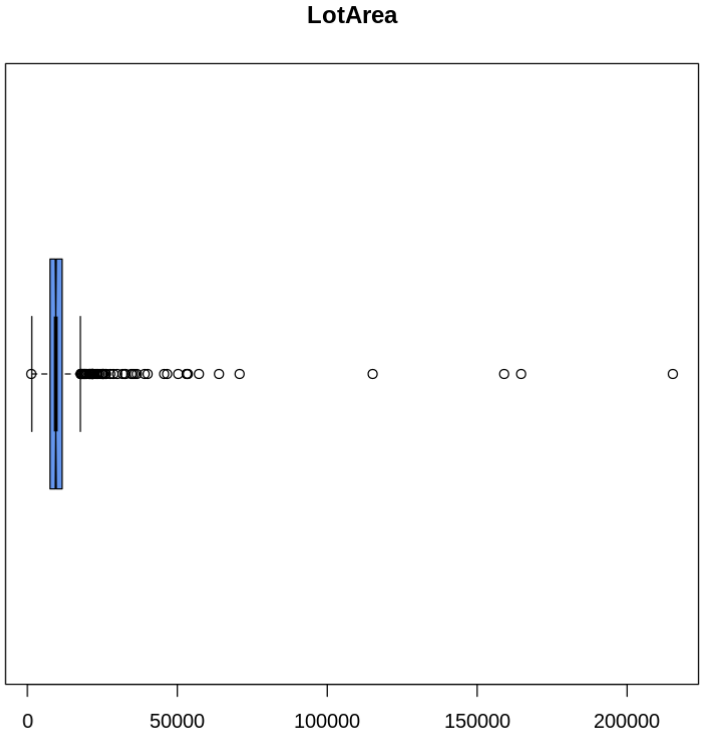


از جدول بالا مشخص میشود که ستون MSSubClass داده های پرت بسیار کمی دارد وپراکندگی تقریبا بالایی دارد. اکثر داده ها نیز قبل از میانه قرار دارند.

In [3]:

```
print("LotArea")
summary(data$LotArea)
boxplot(data$LotArea,
    main = "LotArea",
    col = "cornflowerblue",
    border = "black",
    horizontal = TRUE,
    notch = TRUE
)
```

```
[1] "LotArea"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1300    7554    9478   10517   11602  215245 
```

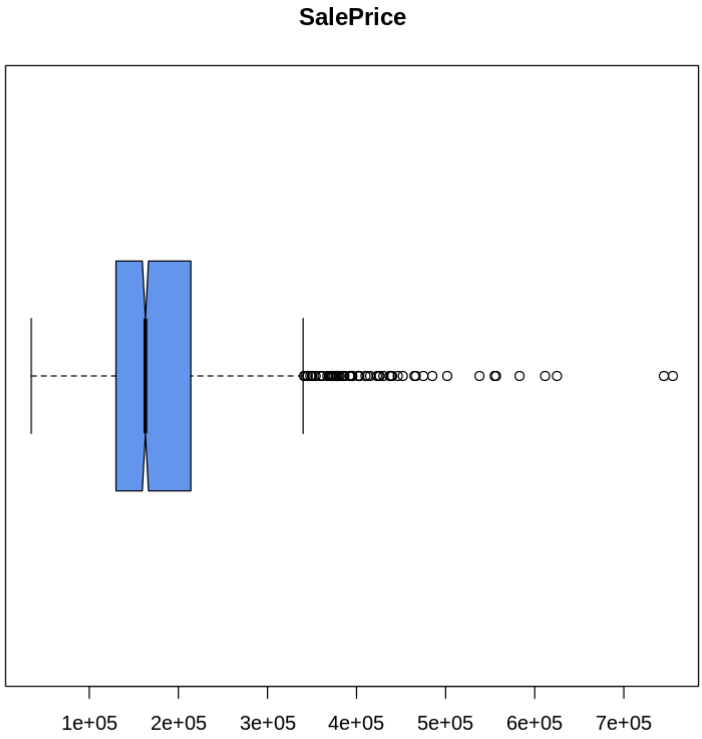


از جدول بالا مشخص می شود ستون LotArea پراکندگی بسیار زیادی دارد.(وجود زیاد داده های پرت)

In [4]:

```
print("SalePrice")
summary(data$SalePrice)
boxplot(data$SalePrice,
    main = "SalePrice",
    col = "cornflowerblue",
    border = "black",
    horizontal = TRUE,
    notch = TRUE
)
```

```
[1] "SalePrice"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  34900  129975  163000  180921  214000  755000 
```



از جدول بالا مشخص می شود ستون SalePrice نیز پراکندگی زیادی دارد.(وجود زیاد داده های پرت)

تست نرمال بودن ستون های عددی:

با استفاده از آزمون shapiro به بررسی نرمال بودن ستون های عددی میپردازیم.

In [5]:
```
shapiro.test(data$MSSubClass)
```

```
        Shapiro-Wilk normality test

data:  data$MSSubClass
W = 0.80457, p-value < 2.2e-16
```

چون در آزمون بالا p-value کمتر از ۰.۰۵ است در نتیجه فرض نرمال بودن ستون MSSubClass رد میشود.

In [6]:
```
shapiro.test(data$LotArea)
```

```
        Shapiro-Wilk normality test

data:  data$LotArea
W = 0.35106, p-value < 2.2e-16
```

چون در آزمون بالا p-value کمتر از ۰.۰۵ است در نتیجه فرض نرمال بودن ستون LotArea رد میشود.

In [7]:
```
shapiro.test(data$SalePrice)
```

```
        Shapiro-Wilk normality test

data:  data$SalePrice
W = 0.86967, p-value < 2.2e-16
```

چون در آزمون بالا p-value کمتر از ۰.۰۵ است در نتیجه فرض نرمال بودن ستون SalePrice رد میشود.

Preprocessing

In [8]:
```
sum(is.na(data))
```

0

طبق دستور بالا هیچ داده not availabe وجود ندارد پس نیازی به تغییر داده های NA نداریم.

حال به سراغ داده های categorical می رویم. میتوان این داده ها را به روش One-Hot encoding برای هر نوع داده در ستون کاراکتری به یک ستون جدید تبدیل کرد.

In [9]:
```
unique(data$MSZoning)
unique(data$SaleCondition)
```

'RL' · 'RM' · 'C (all)' · 'FV' · 'RH'

'Normal' · 'Abnorml' · 'Partial' · 'AdjLand' · 'Alloca' · 'Family'

In [10]:
```
require(caret)
dmy <- dummyVars(" ~ .", data = data)
new_data <- data.frame(predict(dmy, newdata = data))
head(new_data)
```

```
Loading required package: caret

Loading required package: lattice

Loading required package: ggplot2
```

	Id	MSSubClass	MSZoningC..all.	MSZoningFV	MSZoningRH	MSZoningRL	MSZoningRM	L
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	1	60	0	0	0	1	0	
2	2	20	0	0	0	1	0	
3	3	60	0	0	0	1	0	
4	4	70	0	0	0	1	0	
5	5	60	0	0	0	1	0	
6	6	50	0	0	0	1	0	

مشاهده میشود که برای هر یک از ستون های MSZoning و SaleCondition چند ستون اضافه شده است که معادل تعداد داده های منحصر به فرد این ستون هاست. و به ازای هر داده ای که وجود داشت عدد ۱ در سطر معادل آن قرار میگیرد و در بقیه ستون ها عدد ۰ قرار میگیرد.

Preprocessing of test data

In [11]:
```
test = read.csv("test1.csv")
dmy <- dummyVars(" ~ .", data = test)
new_test <- data.frame(predict(dmy, newdata = test))
head(new_test)
```

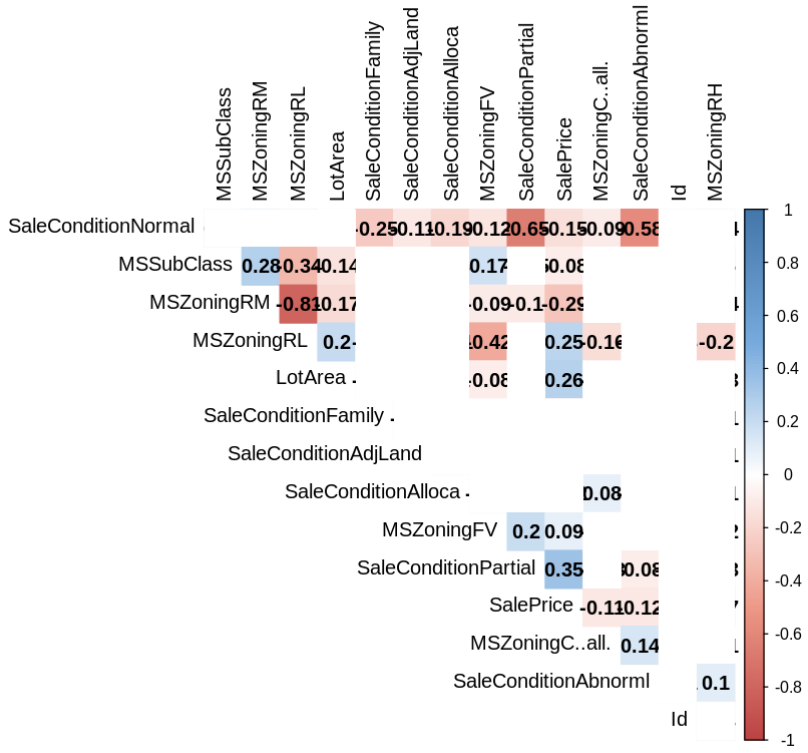
X	Id	MSSubClass	MSZoningC..all.	MSZoningFV	MSZoningRH	MSZoningRL	MSZonin

	X	Id	MSSubClass	MSZoningC (all)	MSZoningFV	MSZoningRH	MSZoningRL	MSZoning
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0	16	45	0	0	0	0	
2	1	23	20	0	0	0	1	
3	2	25	20	0	0	0	1	
4	3	30	30	0	0	0	0	
5	4	35	120	0	0	0	1	
6	5	36	60	0	0	0	1	

روابط بین ستون ها

In [49]:

```
# mat : is a matrix of data
# ... : further arguments to pass to the native R cor.test function
cor.mtest <- function(mat, ...) {
    mat <- as.matrix(mat)
    n <- ncol(mat)
    p.mat<- matrix(NA, n, n)
    diag(p.mat) <- 0
    for (i in 1:(n - 1)) {
        for (j in (i + 1):n) {
            tmp <- cor.test(mat[, i], mat[, j], ...)
            p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
        }
    }
  colnames(p.mat) <- rownames(p.mat) <- colnames(mat)
  p.mat
}
require("corrplot")
m = cor(new_data)
p.mat = cor.mtest(new_data)
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477A
corrplot(m, method="color", col=col(200),
         type="upper", order="hclust",
         addCoef.col = "black", # Add coefficient of correlation
         tl.col="black", #Text label color and rotation
         # Combine with significance
         p.mat = p.mat, sig.level = 0.01, insig = "blank",
         # hide correlation coefficient on the principal diagonal
         diag=FALSE
         )
```



در نمودار بالا هرجایی که P-value برای ضریب همبستگی از ۰.۰۱ کمتر بود ضریب همبستگی در نمودار نشان داده نشده است چون بی ارزش است. نمودار بالا ضریب همبستگی دو به دوی تمام ستون ها را مشخص کرده است. همانطور که مشاهده میکنید ستون ها باهم رابطه خاصی ندارند زیرا ضریب همبستگی بین ستون ها اکثرا نزدیک به صفر است. در جاهایی که صفر نیست هم بین تو ستون است که قبلا یکی بودند و بعد از preprocessing اینگونه شدند. ما به دنبال یافتن بیشترین ضریب همبستگی با ستون SalePrice هستیم که LotArea بیشترین ضریب را دارد. از بین بقیه ستون ها, ستون های زیر بیشترین ضریب همبستگی را دارند:

- MSZoningRL - MSZoningRM: cor = 0.81
- SaleConditionabnormal - SaleConditionNormal: cor = -0.58
- SaleConditionPartial - SaleConditionNormal: cor = 0.65
- SalePrice - LotArea: cor = 0.26

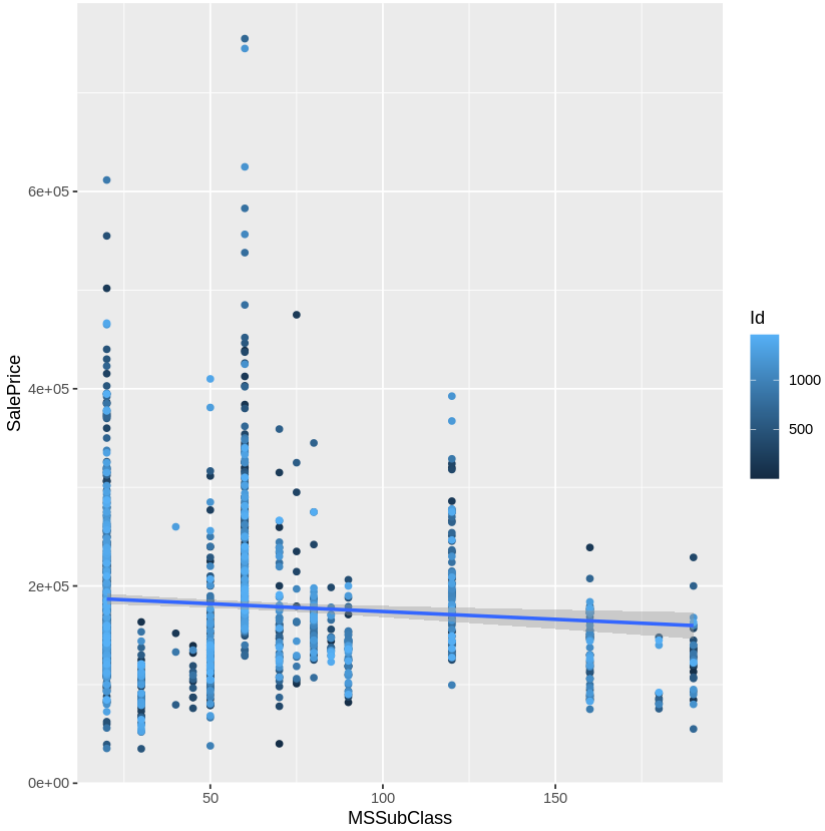
MSSubClass - SalePrice

In [24]:
```
cor.test(data$MSSubClass, data$SalePrice)
```

```
	Pearson's product-moment correlation

data:  data$MSSubClass and data$SalePrice
t = -3.2298, df = 1458, p-value = 0.001266
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.13500269 -0.03312506
sample estimates:
       cor
-0.08428414
```

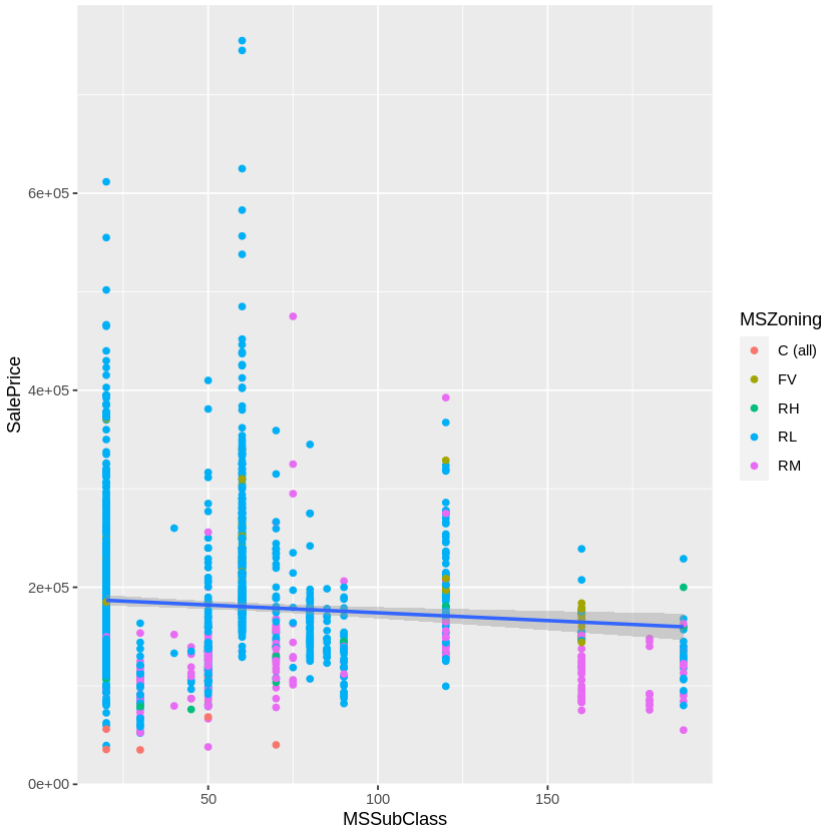
In [25]:
```
require(ggplot2)
ggplot(data = data, mapping = aes (x = MSSubClass, y = SalePrice )) + geom_pc
```

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



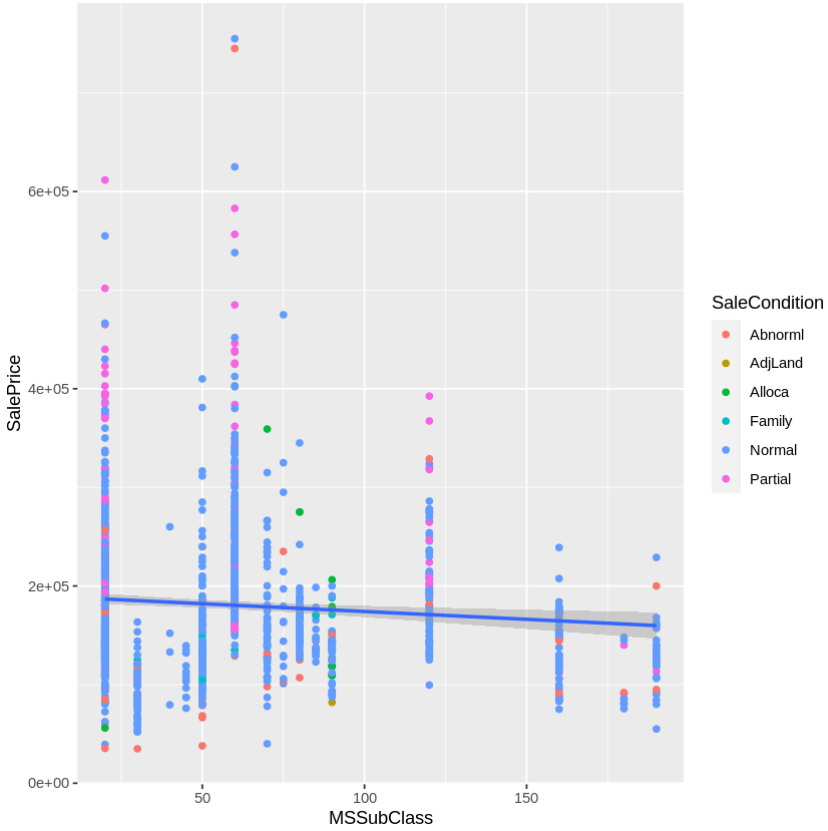
In [26]:
```
ggplot(data = data, mapping = aes (x = MSSubClass, y = SalePrice )) + geom_pc
```

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



In [27]:
```
ggplot(data = data, mapping = aes (x = MSSubClass, y = SalePrice )) + geom_pc
```

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



همانطور که مشاهده میکنید ضریب همبستگی خیلی کم-۰.۰۸ بین این دو ستون باعث شده است بسیار رابطه ضعیفی داشته باشند.
یکی از دلایل آن نیز وجود داده پرت است. طبق نمودار و خط بدست آمده از رگرسیون خطی که در نمودار مشاهده میکنید اکثر داده ها با فاصله ای بسیار زیاد از خط قرار دارند و این نشان میدهد این دو ستون رابطه بسیار بسیار کمی دارند. - در نتیجه رابطه خطی بین این دو ستون وجود ندارد.

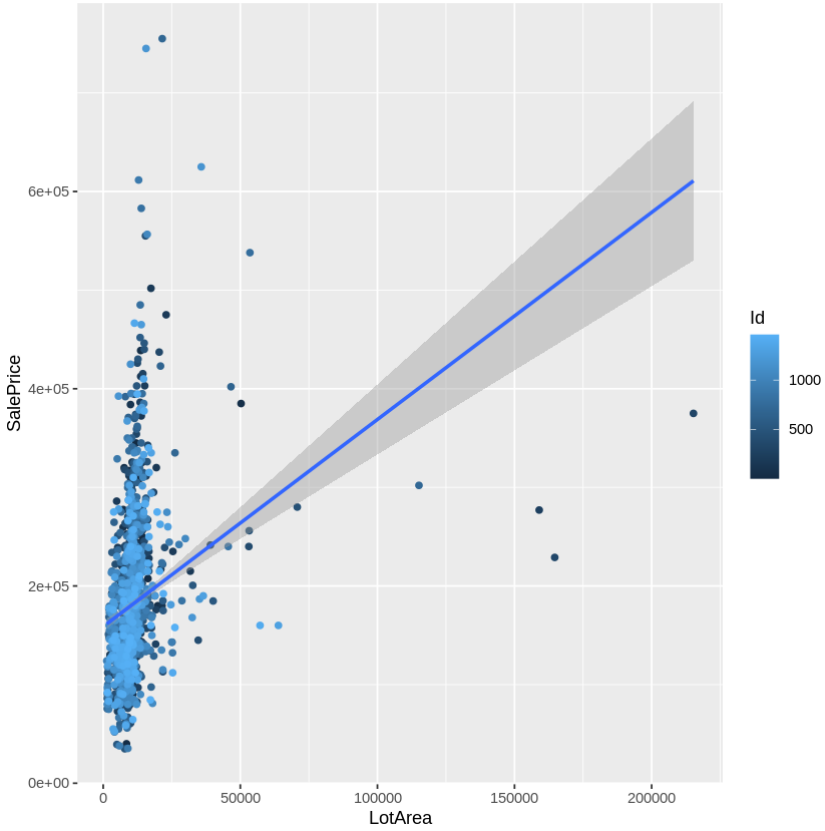
LotArea - SalePrice

In [16]:
```
cor.test(data$LotArea, data$SalePrice)
```

```
        Pearson's product-moment correlation

data:  data$LotArea and data$SalePrice
t = 10.445, df = 1458, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2154574 0.3109369
sample estimates:
      cor
0.2638434
```

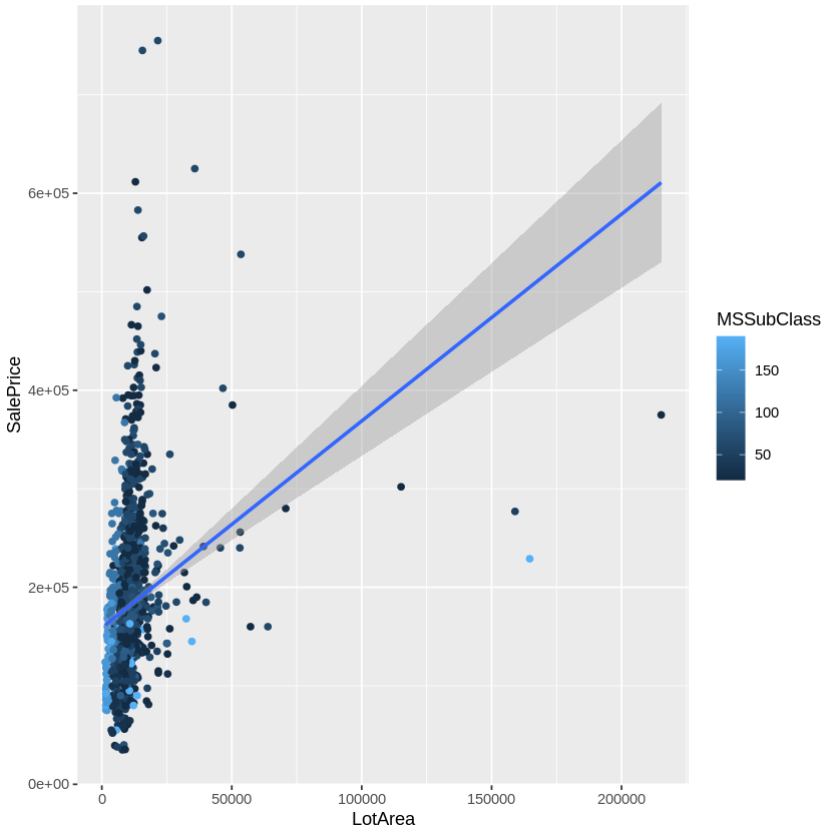
In [21]:
```
ggplot(data = data, mapping = aes (x = LotArea, y = SalePrice )) + geom_point
```

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



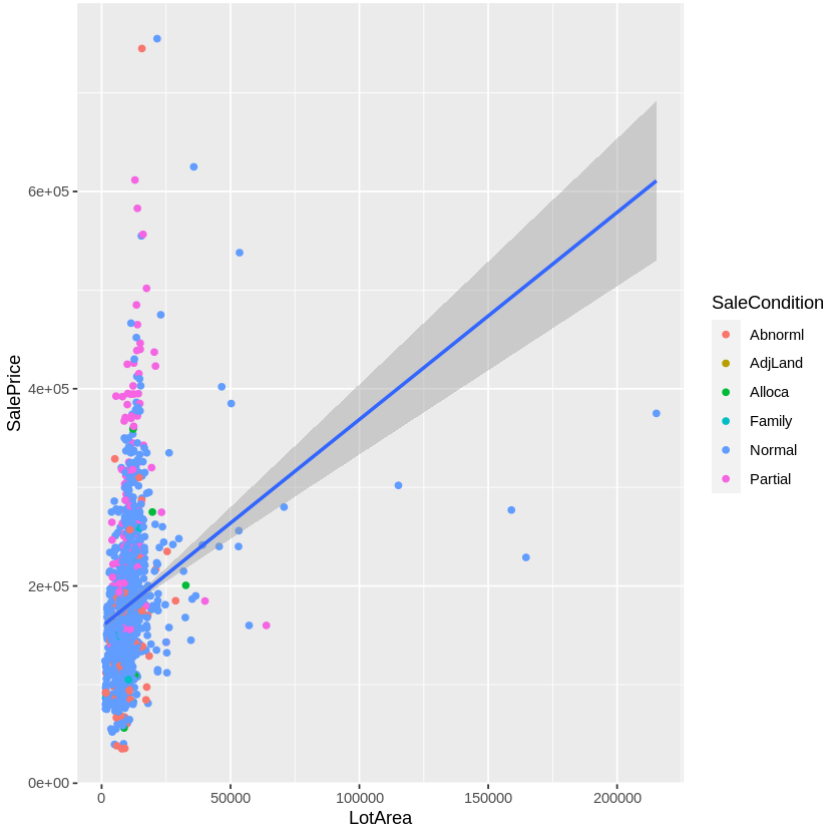
In [22]:
```
ggplot(data = data, mapping = aes (x = LotArea, y = SalePrice )) + geom_point
```

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



In [23]:
```
ggplot(data = data, mapping = aes (x = LotArea, y = SalePrice )) + geom_point
```

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

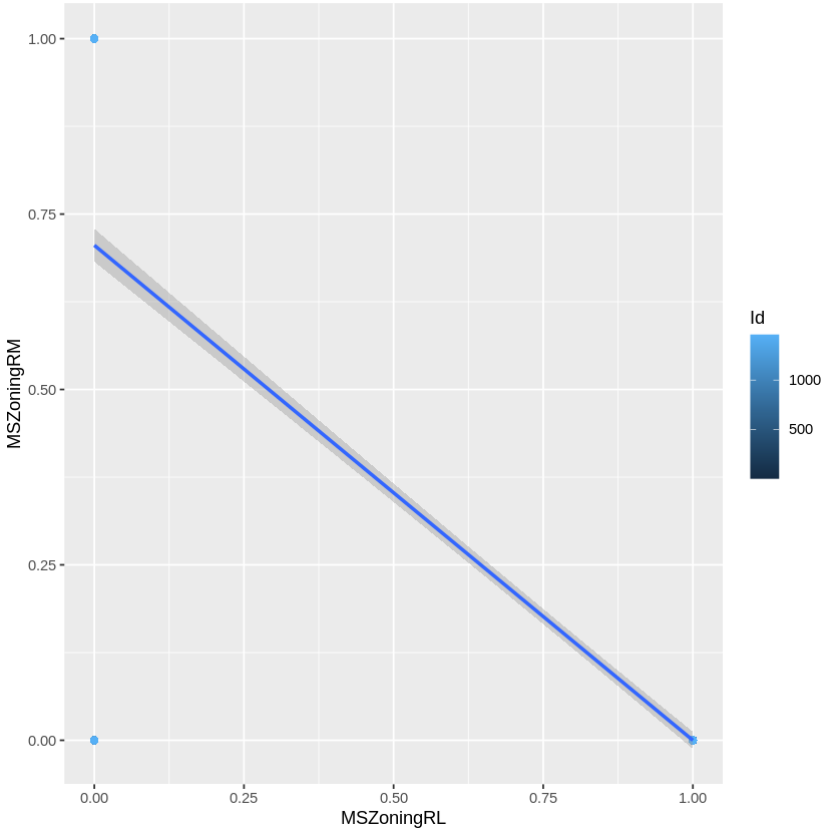


همانطور که مشاهده میکنید ضریب همبستگی خیلی کم 0.26 بین این دو ستون ضریب همبستگی بدی نیست و میتواند گزینه خوبی برای تشکیل رگرسیون خطی با ستون SalePrice باشید. طبق نمودار و خط بدست آمده از رگرسیون خطی که در نمودار مشاهده میکنید که خط در ابتدا بد عمل نمیکند و میانه است ولی از جایی به بعد بسیار دور میشود. دلیل این نیز وجود داده های بسیار پرت است. - در نتیجه رابطه خطی بین این دو ستون وجود ندارد.

MSZoningRM - MSZoningRL

In [28]:
```
ggplot(data = new_data, mapping = aes (x = MSZoningRL, y = MSZoningRM )) + ge
```

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

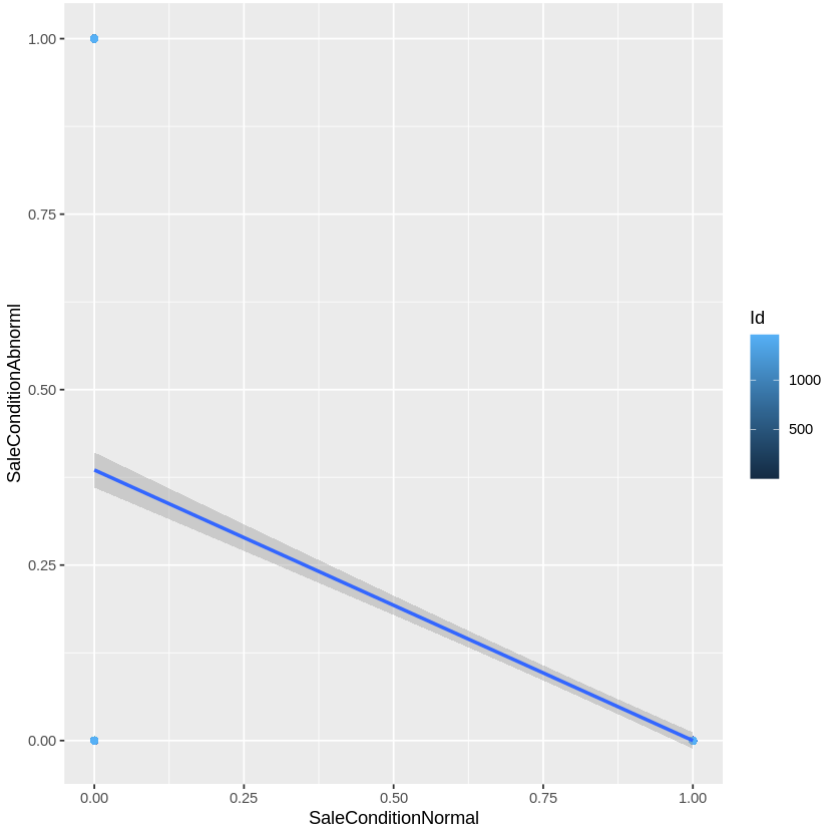


جون داده های بسیاری کمی وجود دارد باعث شده است دو ستون ضریب همبستگی بالایی داشته باشند. طبق خطی که از رگرسیون خطی ساده بدست آمده است از دو نقطه کیی را پوشش میدهد که بنظر خیلی خوب میرسد. - در نتیجه رابطه خطی بین این دو ستون وجود دارد.

SaleConditionNormal - SaleConditionAbnormal

In [30]:
```
ggplot(data = new_data, mapping = aes (x = SaleConditionNormal, y = SaleCondi
```

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



جون داده های بسیاری کمی وجود دارد باعث شده است دو ستون ضریب همبستگی تقریبا بالایی داشته باشند. طبق خطی که از رگرسیون خطی ساده بدست آمده است از دو نقطه کیی را پوشش میدهد که بنظر خیلی خوب میرسد. - در نتیجه رابطه خطی بین این دو ستون وجود دارد.

نتیجه گیری بین بقیه ستون ها:

از بین بقیه ستون ها هیچکدام ضریب همبستگی قابل قبولی نداشتند برای همین مشخص است که بین بقیه ستون ها رابطه ای وجود ندارد و اکثرا به دلیل پراکندگی بسیار بالا این اتفاق برایشان افتاده است.(نمودار کامل ضریب همبستگی ها در جدول بالاتر قرار دارد.)

مدل رگرسیون خطی ساده

از آنجایی که ستون LotArea ضریب همبستگی بیشتری داشت ضرایب رگرسیون خطی را با این ستون بررسی میکنیم.

In [31]:
```
model1 = lm(SalePrice ~ LotArea, data = new_data)
summary(model1)
```

```
Call:
lm(formula = SalePrice ~ LotArea, data = new_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-275668  -48169  -17725   31248  553356 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.588e+05  2.915e+03   54.49   <2e-16 ***
LotArea     2.100e+00  2.011e-01   10.45   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 76650 on 1458 degrees of freedom
Multiple R-squared:  0.06961,   Adjusted R-squared:  0.06898 
F-statistic: 109.1 on 1 and 1458 DF,  p-value: < 2.2e-16
```

مدل پیش بینی شده : SalePrice = 1.58*10^5 + 2.1 * LotArea

In [32]:
```
pred1 = predict(model1, newdata = new_test)
require(Metrics)
print("Sqrt of MSE:")
sqrt(mse(pred1, test$SalePrice)) #sqrt of MSE
```

```
Loading required package: Metrics


Attaching package: ‘Metrics’


The following objects are masked from ‘package:caret’:

    precision, recall


[1] "Sqrt of MSE:"
75364.1113894224
```

مشاهده میشود که mse برابر با ۷۵ هزار شده است و از ۷۰ هزار بیشتر است. در نتیجه رگرسیون ساده با ستون LotArea فاییده ایندارد. از طرفی میدانیم ستون LotArea بیشترین ضریب همبستگی را با ستون Sale Price داشت پس هر رگرسیون ساده دیگری نیز از ۷۵ هزار کمتر نخواهد شد. در نتیجه به رگرسیون چندگانه خطی میپردازیم.

مدل رگرسیون خطی چندگانه

میدانیم ضریب همبستگی ها چقد بد هستند. در نتیجه از اول بجای اینکه با دو یا سه تا ستون بخواهیم رگرسیون چندگانه خطی برای ستون Sale Price بسازیم سعی میکنیم با همه ستون های این مدل را بسازیم.

In [33]:
```
model2 = lm(SalePrice ~ .,
            data = new_data)
summary(model2)
```

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

Residuals:
    Min      1Q  Median      3Q     Max
-227905  -42624  -10723   26104  573962

Coefficients: (2 not defined because of singularities)
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          2.018e+05  9.271e+03  21.768  < 2e-16 ***
Id                  -3.362e+00  4.280e+00  -0.785    0.432
MSSubClass           5.133e+01  4.564e+01   1.125    0.261
MSZoningC..all.     -4.614e+04  2.256e+04  -2.046    0.041 *
MSZoningFV           5.800e+04  9.949e+03   5.829 6.86e-09 ***
MSZoningRH           1.015e+04  1.789e+04   0.567    0.571
MSZoningRL           5.142e+04  5.415e+03   9.496  < 2e-16 ***
MSZoningRM                  NA         NA      NA       NA
LotArea              1.719e+00  1.849e-01   9.297  < 2e-16 ***
SaleConditionAbnorml -1.082e+05  9.409e+03 -11.495  < 2e-16 ***
SaleConditionAdjLand -1.649e+05  3.498e+04  -4.716 2.64e-06 ***
SaleConditionAlloca  -9.024e+04  2.094e+04  -4.310 1.74e-05 ***
SaleConditionFamily  -1.051e+05  1.667e+04  -6.308 3.76e-10 ***
SaleConditionNormal  -8.803e+04  6.622e+03 -13.294  < 2e-16 ***
SaleConditionPartial         NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 68690 on 1447 degrees of freedom
```

```
Multiple R-squared:  0.2584,    Adjusted R-squared:  0.2523
F-statistic: 42.03 on 12 and 1447 DF,  p-value: < 2.2e-16
```

مدل پیش بینی شده :

SalePrice = 2.018*10^5* + *-3.362* Id + 51.3 *MSSubClass* + *-4.614*10^4 *MSZoningC..all.* + *5.8*10^4 *MSZoningFV* + *1.015*10^4 *MSZoningRH* + *5.142*10^4 *MSZoningRL* + *1.719* LotArea - 1.082*10^5* SaleConditionAbnorml - 1.649*10^5* SaleConditionAdjLand - 9.024*10^4* SaleConditionAlloca - 1.051*10^5* SaleConditionFamily - 8.803*10^4* SaleConditionPartial

In [34]:

```
pred2 = predict(model2, newdata = new_test)
print("Sqrt of MSE:")
sqrt(mse(pred2, test$SalePrice)) #sqrt of mse
```

```
Warning message in predict.lm(model2, newdata = new_test):
“prediction from a rank-deficient fit may be misleading”
[1] "Sqrt of MSE:"
67105.7998810891
```

حال در بالا جذر MSE برابر ۶۷ هزار شده است که از ۷۰ هزار کمتر است در نتیجه مدلی درست میتواند باشد.

Random Forest

اکنون برای بهتر کردن مدل از random forrest استفاده میکنیم.

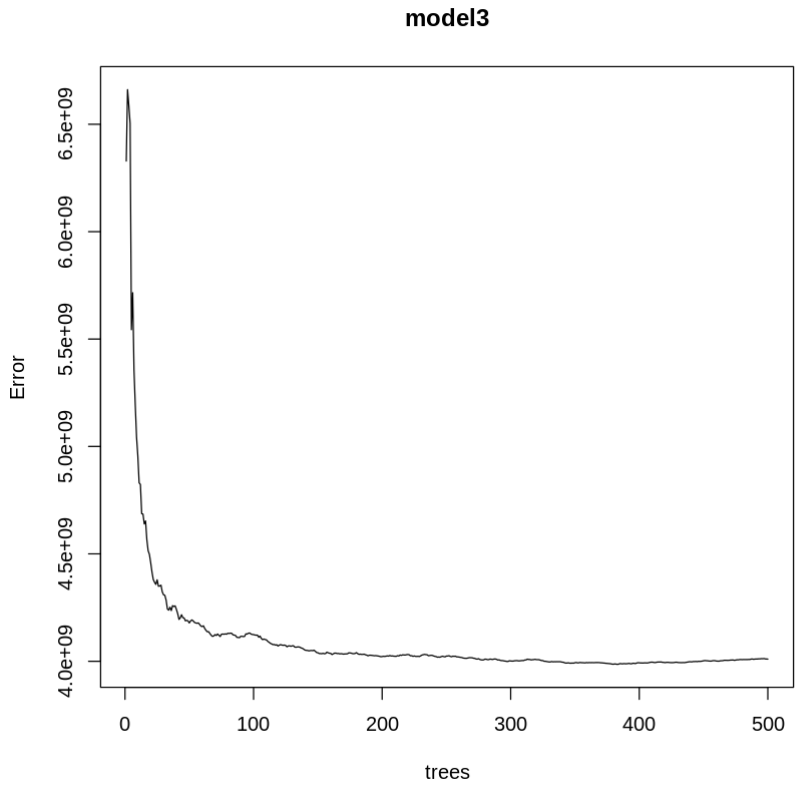
In [35]:

```
require("randomForest")
model3 = randomForest(SalePrice ~ .,
                      data = new_data,
                      mtry = 13,
                      importance = TRUE,
                      method = "anova"
                     )
print(model3)
```

```
Loading required package: randomForest

randomForest 4.6-14

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


Attaching package: ‘randomForest’


The following object is masked from ‘package:ggplot2’:

    margin


Call:
 randomForest(formula = SalePrice ~ ., data = new_data, mtry = 13,      impor
tance = TRUE, method = "anova")
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 13

          Mean of squared residuals: 4010039855
                    % Var explained: 36.42
```

در آزمون بالا با mtry برابر ۱۳ بهترین جواب را خواهیم داشت که از آن تا ۵۰۰ درخت را ساخته است که در نمودار زیر نشان داده شده است.

In [36]:
```
plot(model3)
```



In [37]:
```
pred3 = predict(model3, newdata = new_test)
print("Sqrt of MSE:")
sqrt(mse(pred3, test$SalePrice)) #sqrt of mse
```

```
[1] "Sqrt of MSE:"
28174.0606490255
```

بعد از پیش بینی جواب های tes1.csv و بدست آوردن mse مشخص میشود که mse آن برابر با ۲۸۰۰۰ است و خیلی بهتر از مدل رگرسیون خطی چندگانه عمل کرده است.

خروجی برنامه:

In [38]:
```
output = data.frame("Id" = new_test$Id)
output$SalePrice = pred3
output
```

A data.frame: 191 × 2

Id <dbl>	SalePrice <dbl>
16	124758.44
23	207031.03
25	144376.78
30	86695.61
35	255037.20
36	271799.52

Id	SalePrice
<dbl>	<dbl>
42	184593.02
45	137297.09
66	260129.13
69	84381.00
75	109748.42
76	89769.15
113	340923.51
118	167827.81
120	176393.52
127	179990.29
146	123437.59
149	138641.95
155	132033.77
156	111121.09
161	175180.13
166	135690.84
168	301856.01
175	200492.13
192	183169.81
199	104076.85
201	132800.02
206	196020.63
210	139875.88
230	195001.93
⋮	⋮
1245	207434.75
1248	175229.87
1255	162219.48
1257	298921.70
1262	151024.43
1298	198893.72
1301	222245.24
1307	214141.97
1308	149977.98
1313	268054.95
1318	213339.17
1320	149865.25
1327	81930.41

Id	SalePrice
<dbl>	<dbl>
1344	154424.37
1351	169668.22
1353	130845.02
1358	188238.66
1362	265760.04
1395	232156.19
1396	244318.45
1402	186017.43
1405	128451.51
1424	251459.83
1431	245914.57
1433	99343.45
1438	368340.48
1443	282157.55
1444	110782.08
1450	95911.39
1453	157715.85

In [31]:
```
write.csv(output, "./projectanswer.csv")
```