Real Estate

R Markdown

1. Boxplot to check the trend for house price across all transaction dates.

```
real = read.csv('Real estate.csv',sep = ',', header = TRUE)
head(real)
```

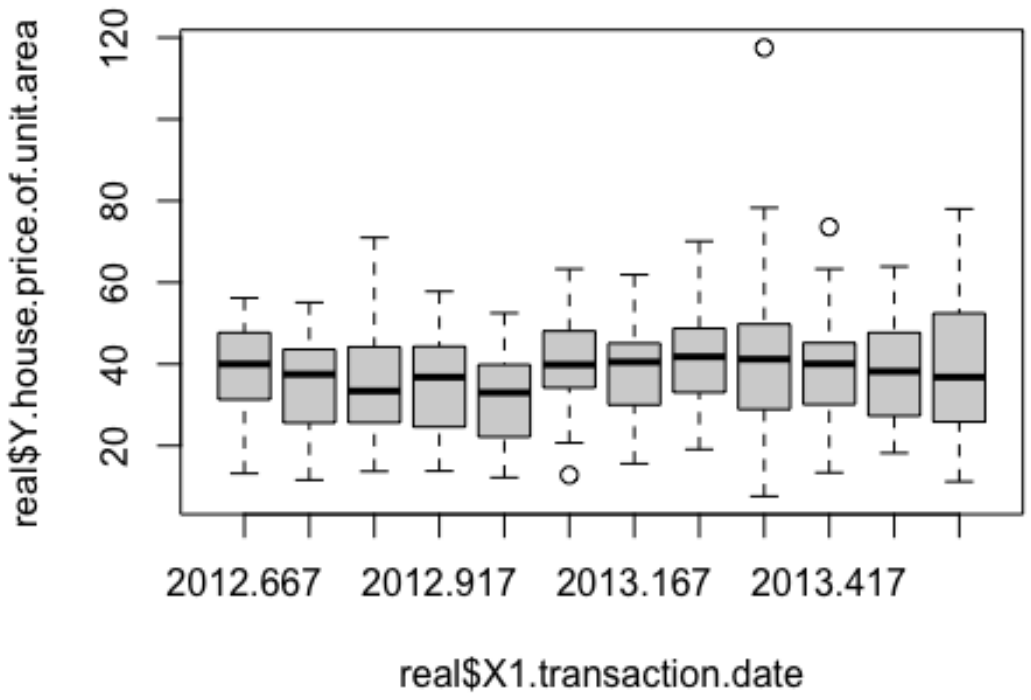
```
##   No X1.transaction.date X2.house.age
X3.distance.to.the.nearest.MRT.station
## 1  1            2012.917         32.0
84.87882
## 2  2            2012.917         19.5
306.59470
## 3  3            2013.583         13.3
561.98450
## 4  4            2013.500         13.3
561.98450
## 5  5            2012.833          5.0
390.56840
## 6  6            2012.667          7.1
2175.03000
##   X4.number.of.convenience.stores X5.latitude X6.longitude
## 1                              10    24.98298     121.5402
## 2                               9    24.98034     121.5395
## 3                               5    24.98746     121.5439
## 4                               5    24.98746     121.5439
## 5                               5    24.97937     121.5425
## 6                               3    24.96305     121.5125
##   Y.house.price.of.unit.area
## 1                       37.9
## 2                       42.2
## 3                       47.3
## 4                       54.8
## 5                       43.1
## 6                       32.1
```

```
summary(real)
```

```
##        No        X1.transaction.date  X2.house.age
##  Min.   :  1.0   Min.   :2013        Min.   : 0.000
##  1st Qu.:104.2   1st Qu.:2013        1st Qu.: 9.025
##  Median :207.5   Median :2013        Median :16.100
##  Mean   :207.5   Mean   :2013        Mean   :17.713
##  3rd Qu.:310.8   3rd Qu.:2013        3rd Qu.:28.150
##  Max.   :414.0   Max.   :2014        Max.   :43.800
##  X3.distance.to.the.nearest.MRT.station X4.number.of.convenience.stores
##  Min.   :  23.38                        Min.   : 0.000
```

```
##  1st Qu.: 289.32                          1st Qu.: 1.000
##  Median : 492.23                          Median : 4.000
##  Mean   :1083.89                          Mean   : 4.094
##  3rd Qu.:1454.28                          3rd Qu.: 6.000
##  Max.   :6488.02                          Max.   :10.000
##   X5.latitude     X6.longitude   Y.house.price.of.unit.area
##  Min.   :24.93   Min.   :121.5   Min.   :  7.60
##  1st Qu.:24.96   1st Qu.:121.5   1st Qu.: 27.70
##  Median :24.97   Median :121.5   Median : 38.45
##  Mean   :24.97   Mean   :121.5   Mean   : 37.98
##  3rd Qu.:24.98   3rd Qu.:121.5   3rd Qu.: 46.60
##  Max.   :25.01   Max.   :121.6   Max.   :117.50
```

```
boxplot(data = real,
real$Y.house.price.of.unit.area~real$X1.transaction.date)
```



2. Running initial LM model:

- Based on the p-value (alpha = 0.05), No and Longitude are not strong predictors.

```
r_model1 = lm(real$Y.house.price.of.unit.area~., data = real)
summary(r_model1)
```

```
##
## Call:
```

```
## lm(formula = real$Y.house.price.of.unit.area ~ ., data = real)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.003  -5.196  -0.990   4.181  75.384
##
## Coefficients:
##                                         Estimate Std. Error t value
Pr(>|t|)
## (Intercept)                            -1.404e+04  6.788e+03  -2.068
0.03927
## No                                     -3.593e-03  3.653e-03  -0.984
0.32590
## X1.transaction.date                     5.079e+00  1.559e+00   3.259
0.00121
## X2.house.age                           -2.708e-01  3.855e-02  -7.026
9.04e-12
## X3.distance.to.the.nearest.MRT.station -4.521e-03  7.189e-04  -6.289
8.28e-10
## X4.number.of.convenience.stores         1.129e+00  1.882e-01   6.000
4.37e-09
## X5.latitude                             2.247e+02  4.458e+01   5.040
7.02e-07
## X6.longitude                           -1.442e+01  4.863e+01  -0.297
0.76691
##
## (Intercept)                            *
## No
## X1.transaction.date                    **
## X2.house.age                           ***
## X3.distance.to.the.nearest.MRT.station ***
## X4.number.of.convenience.stores        ***
## X5.latitude                            ***
## X6.longitude
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.858 on 406 degrees of freedom
## Multiple R-squared:  0.5834, Adjusted R-squared:  0.5762
## F-statistic: 81.21 on 7 and 406 DF,  p-value: < 2.2e-16
```

3. Creating another LM model, with reduced predictors. – All the predictors have a p-value < 0.05.

```
r_model2 = lm(Y.house.price.of.unit.area~X1.transaction.date+
X2.house.age+X3.distance.to.the.nearest.MRT.station+
X4.number.of.convenience.stores +X5.latitude, data = real)

summary(r_model2)
```

```
##
## Call:
## lm(formula = Y.house.price.of.unit.area ~ X1.transaction.date +
##     X2.house.age + X3.distance.to.the.nearest.MRT.station +
X4.number.of.convenience.stores +
##     X5.latitude, data = real)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.623  -5.371  -1.020   4.244  75.346
##
## Coefficients:
##                                        Estimate Std. Error t value
Pr(>|t|)
## (Intercept)                          -1.596e+04  3.233e+03  -4.936
1.17e-06
## X1.transaction.date                   5.135e+00  1.555e+00   3.303
0.00104
## X2.house.age                         -2.694e-01  3.847e-02  -7.003
1.04e-11
## X3.distance.to.the.nearest.MRT.station -4.353e-03  4.899e-04  -8.887  <
2e-16
## X4.number.of.convenience.stores       1.136e+00  1.876e-01   6.056
3.17e-09
## X5.latitude                           2.269e+02  4.417e+01   5.136
4.36e-07
##
## (Intercept)                          ***
## X1.transaction.date                  **
## X2.house.age                         ***
## X3.distance.to.the.nearest.MRT.station ***
## X4.number.of.convenience.stores      ***
## X5.latitude                          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.848 on 408 degrees of freedom
## Multiple R-squared:  0.5823, Adjusted R-squared:  0.5772
## F-statistic: 113.8 on 5 and 408 DF,  p-value: < 2.2e-16
```

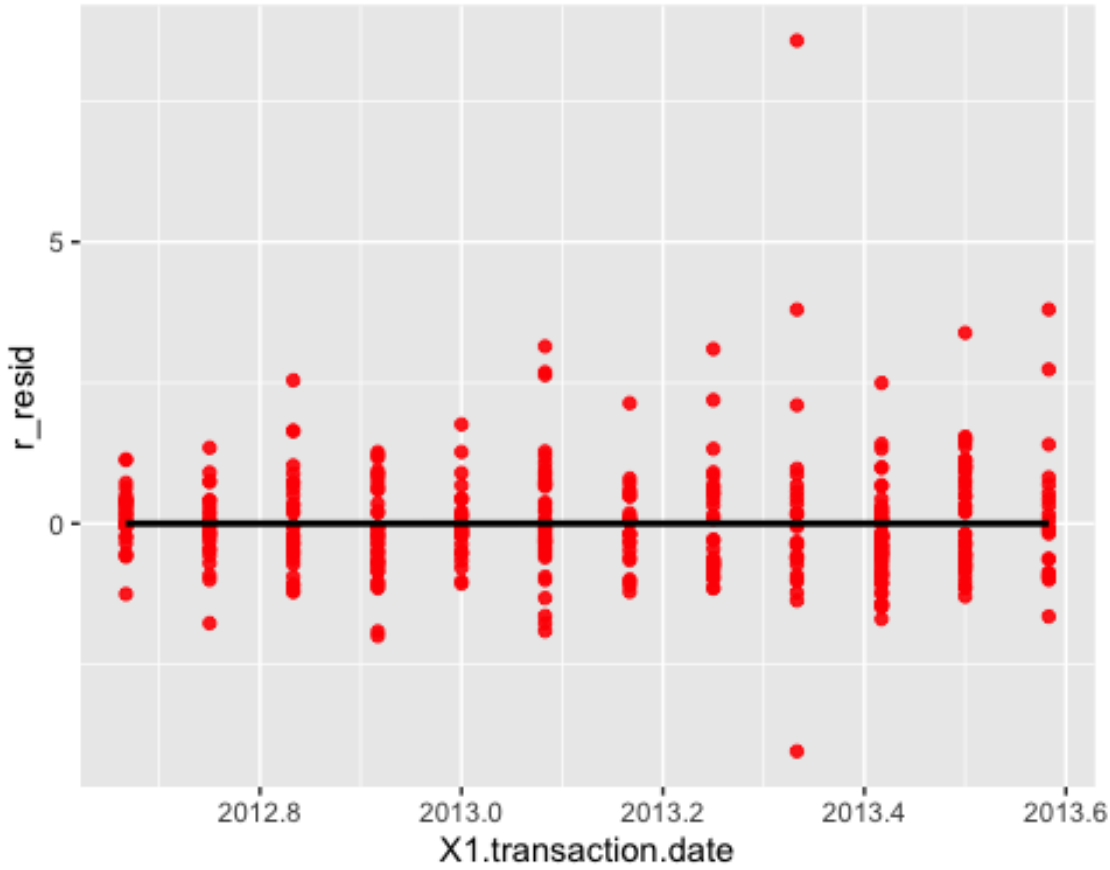
4. Residual analysis to check the goodness of fit for model2:

- Linearity Assumption - It shows departure from the assumption, as some patterns can be observed.
- Constant/Independent - It shows departure as the residuals variability increases with increase in fitted values.
- Normality - It shows departure both in qqplot and Histogram.

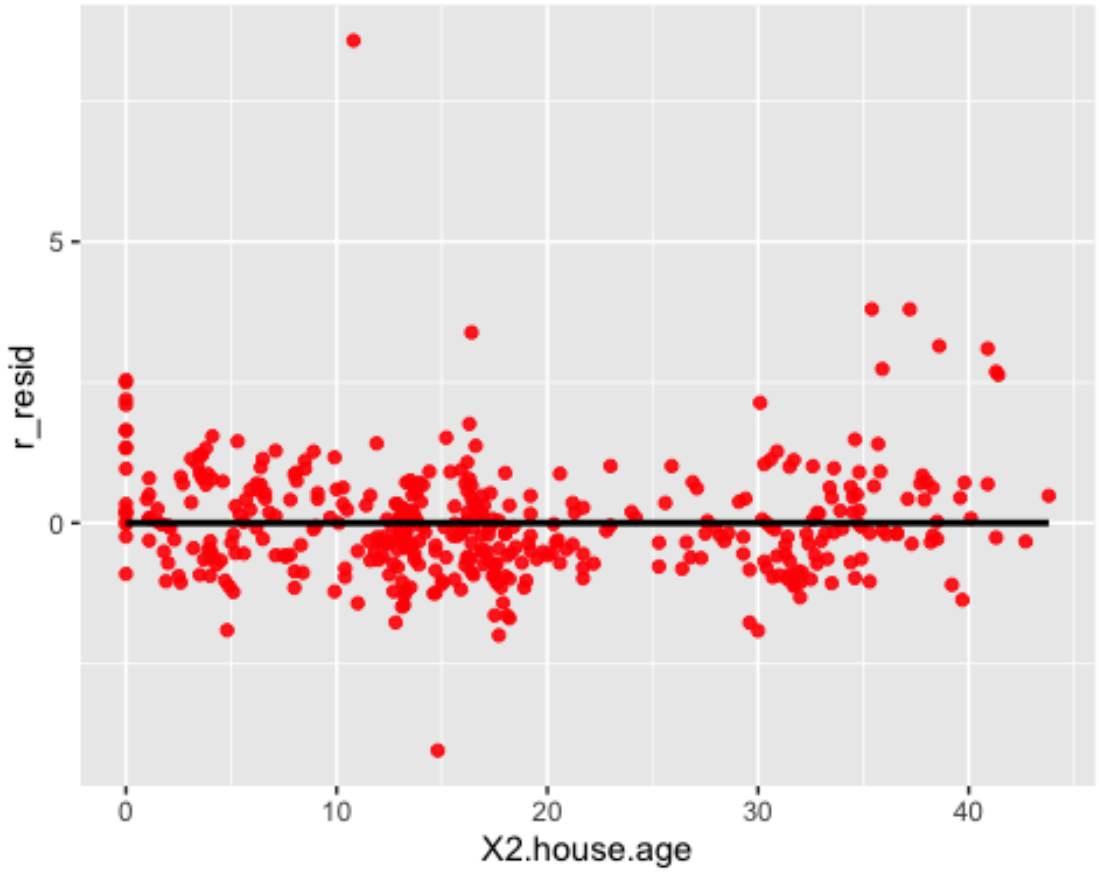
Based on the above analysis, I will perform boxcox transformation.

```
r_resid = rstandard(r_model2)

for (i in 2:6 ){

print(ggplot(data = real, aes(x = real[,i], y = r_resid )) +
 xlab(colnames(real)[i])+
  geom_point(color = 'red', alpha = I(0.9))+
    geom_smooth(method = 'lm',se = F, color = 'black'))

}
```

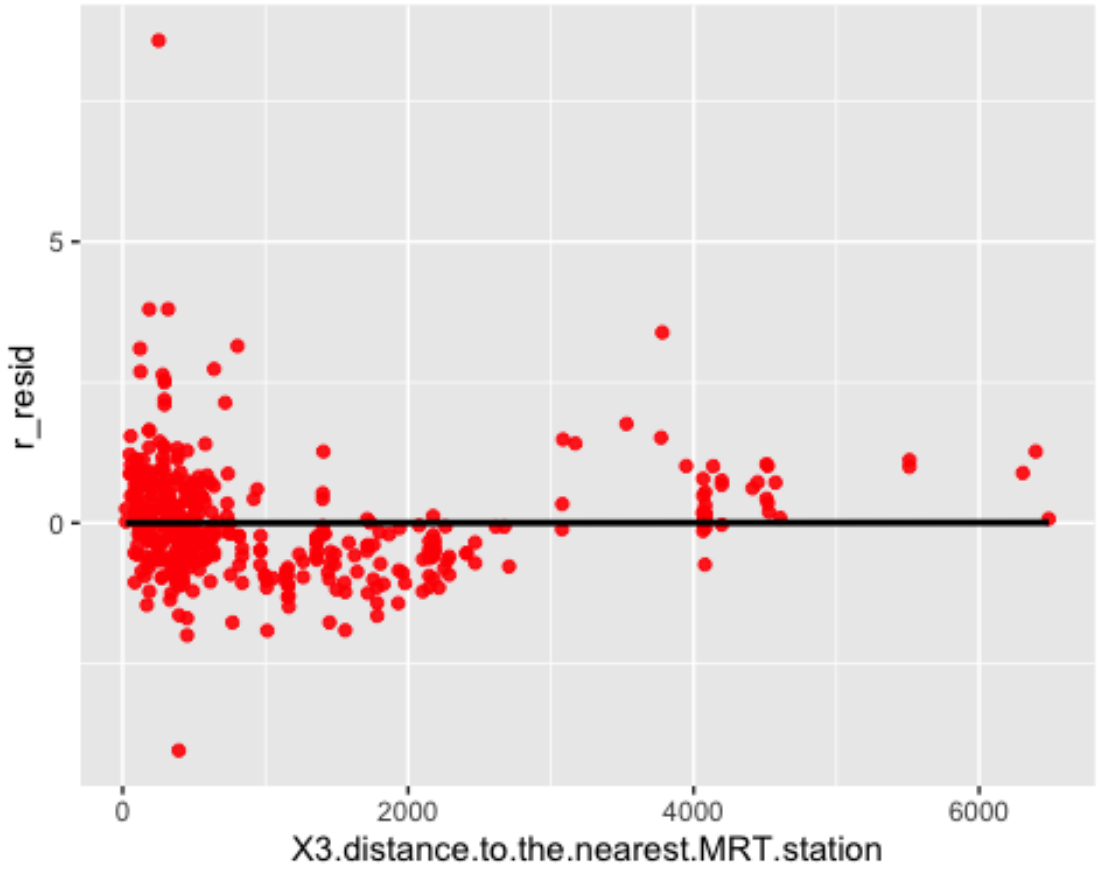
```
## `geom_smooth()` using formula 'y ~ x'
```



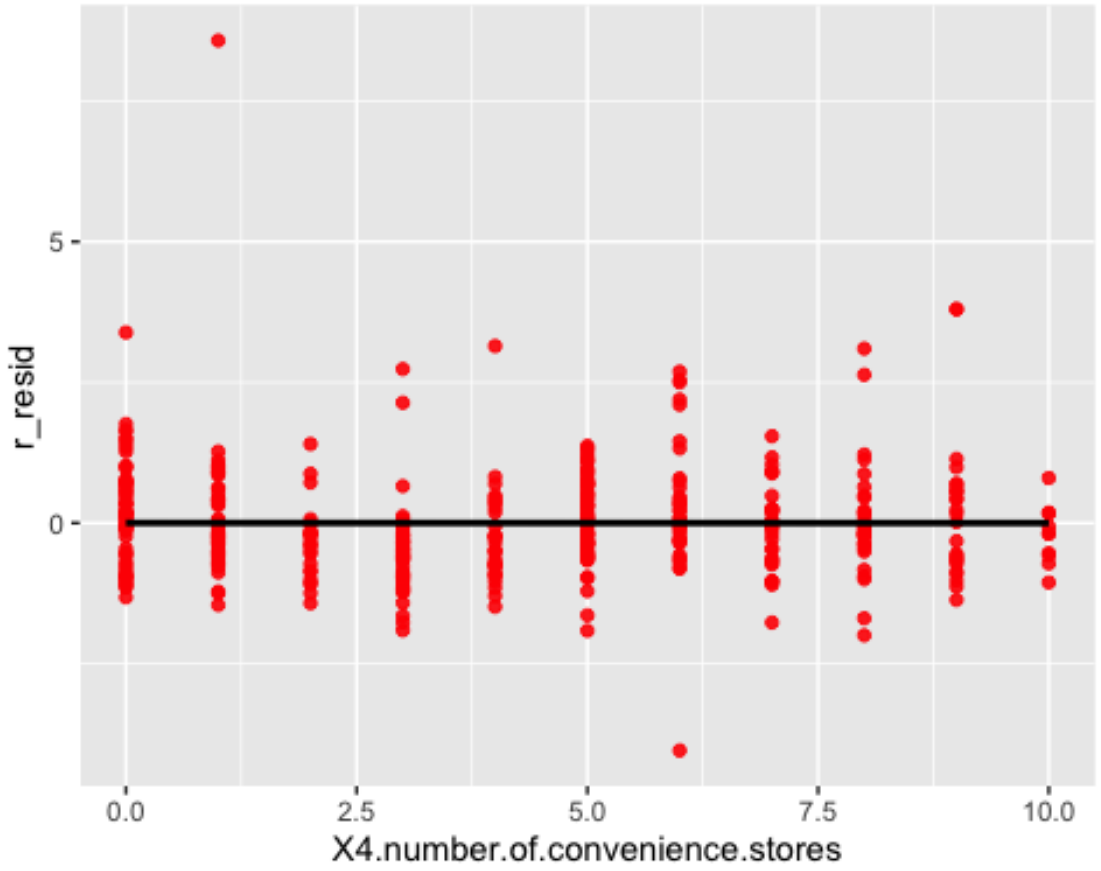

```
## `geom_smooth()` using formula 'y ~ x'
```



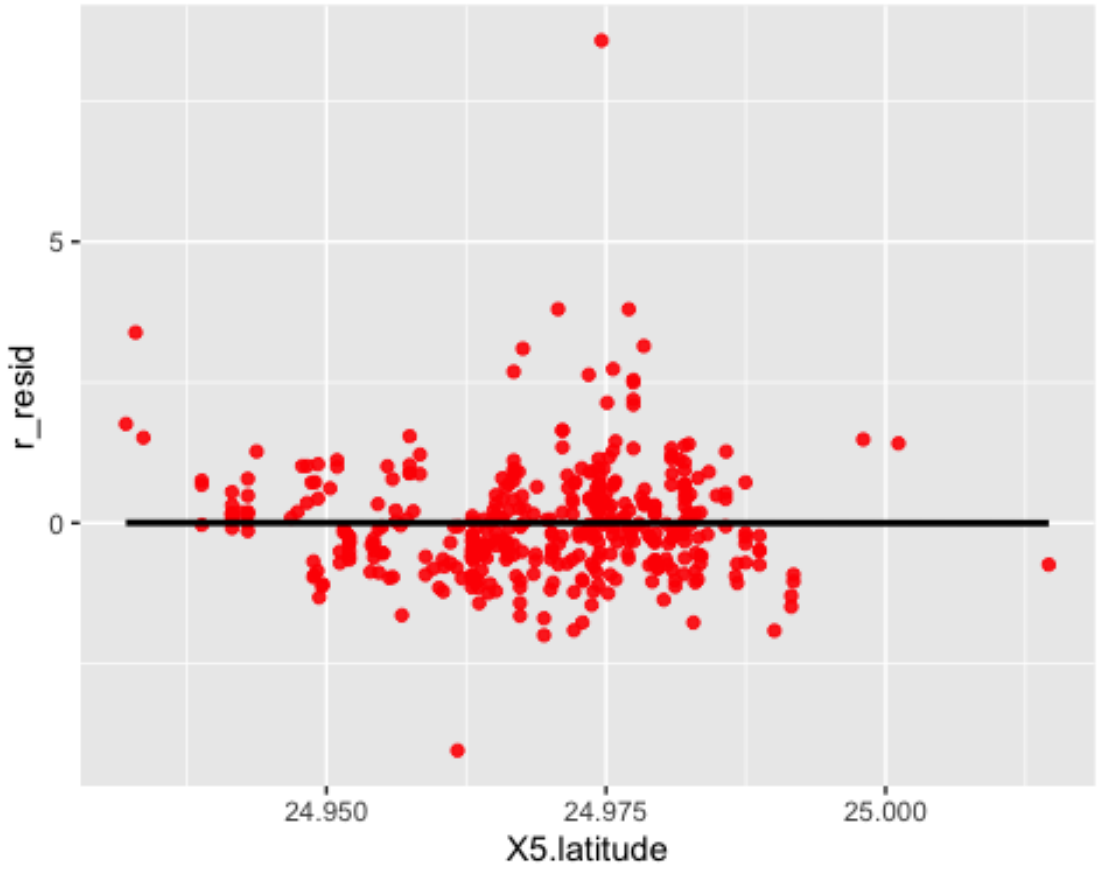

```
## `geom_smooth()` using formula 'y ~ x'
```



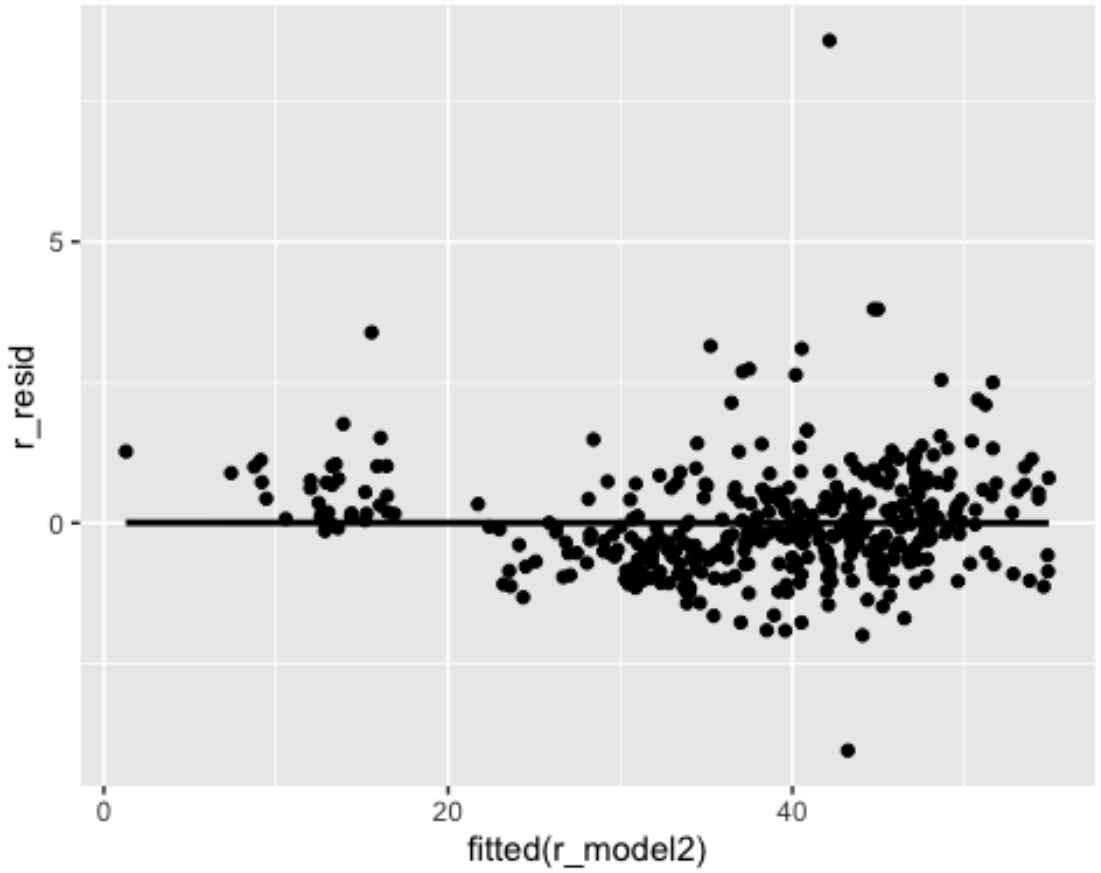

```
## `geom_smooth()` using formula 'y ~ x'
```



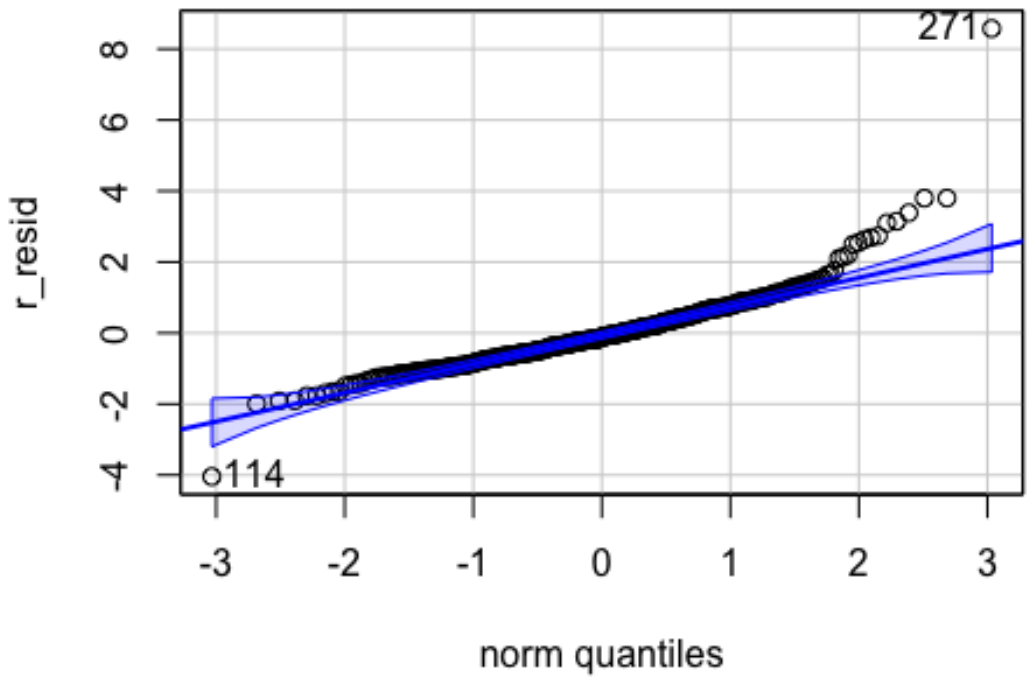

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggplot(data = real, aes(x = fitted(r_model2), y = r_resid))+
  geom_point()+
  geom_smooth(method = 'lm', se = F, color = 'black')

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

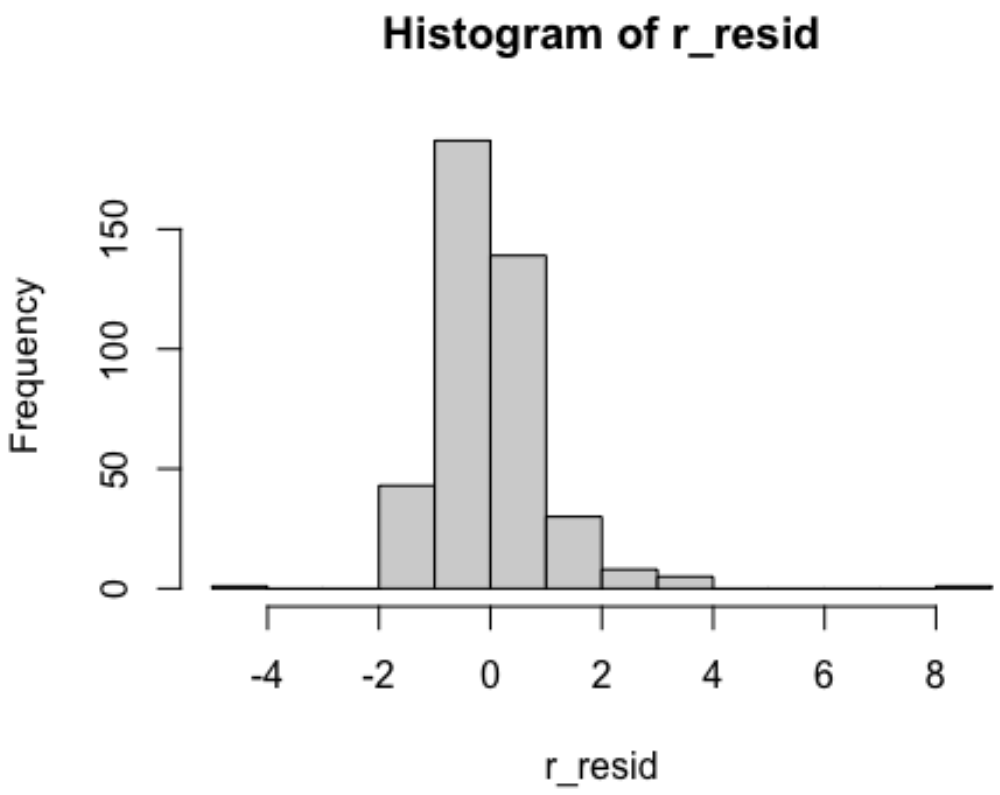



```
qqPlot(r_resid)
```



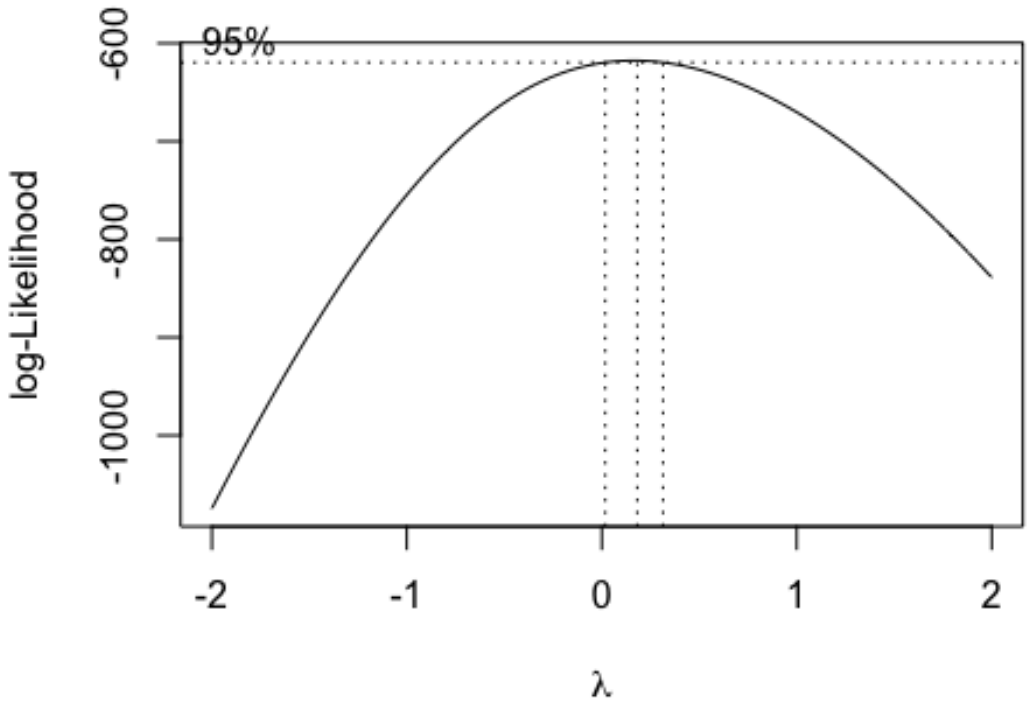

```
## [1] 271 114
```

```
hist(r_resid)
```



5. Since the optimum lambda value is 0, I will perform log transformation on the response variable and run a new lm model.

```
rb_cox = boxcox(r_model2)
```




```
lam = rb_cox$x[which.max(rb_cox$y)]
optimum_lam = round(lam/0.5)*0.5
optimum_lam
```

```
## [1] 0
```

6. Transformed Model:

```
r_model3 = lm(log(Y.house.price.of.unit.area)~X1.transaction.date+
X2.house.age+X3.distance.to.the.nearest.MRT.station+
X4.number.of.convenience.stores +X5.latitude, data = real)

summary(r_model3)
```

```
##
## Call:
## lm(formula = log(Y.house.price.of.unit.area) ~ X1.transaction.date +
##     X2.house.age + X3.distance.to.the.nearest.MRT.station +
X4.number.of.convenience.stores +
##     X5.latitude, data = real)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.68218 -0.11505  0.00055  0.11262  1.04395
```

```
##
## Coefficients:
##                                          Estimate Std. Error t value
Pr(>|t|)
## (Intercept)                             -4.665e+02  8.091e+01  -5.766
1.61e-08
## X1.transaction.date                      1.358e-01  3.890e-02   3.491
0.000533
## X2.house.age                            -6.977e-03  9.625e-04  -7.248
2.13e-12
## X3.distance.to.the.nearest.MRT.station -1.495e-04  1.226e-05 -12.194  <
2e-16
## X4.number.of.convenience.stores          2.766e-02  4.694e-03   5.892
7.97e-09
## X5.latitude                              7.883e+00  1.105e+00   7.132
4.54e-12
##
## (Intercept)                             ***
## X1.transaction.date                     ***
## X2.house.age                            ***
## X3.distance.to.the.nearest.MRT.station ***
## X4.number.of.convenience.stores         ***
## X5.latitude                             ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2214 on 408 degrees of freedom
## Multiple R-squared:  0.6857, Adjusted R-squared:  0.6818
## F-statistic:   178 on 5 and 408 DF,  p-value: < 2.2e-16
```

```
confint(r_model3)
```

```
##                                                 2.5 %        97.5 %
## (Intercept)                             -6.255404e+02 -3.074476e+02
## X1.transaction.date                      5.933901e-02  2.122782e-01
## X2.house.age                            -8.868633e-03 -5.084448e-03
## X3.distance.to.the.nearest.MRT.station -1.735674e-04 -1.253755e-04
## X4.number.of.convenience.stores          1.843368e-02  3.689038e-02
## X5.latitude                              5.710131e+00  1.005580e+01
```

```
cor(real)
```

```
##                                                  No X1.transaction.date
## No                                       1.00000000        -0.048657949
## X1.transaction.date                     -0.04865795         1.000000000
## X2.house.age                            -0.03280811         0.017548767
## X3.distance.to.the.nearest.MRT.station -0.01357349         0.060879953
## X4.number.of.convenience.stores         -0.01269895         0.009635445
## X5.latitude                             -0.01010966         0.035057756
## X6.longitude                            -0.01105928        -0.041081778
## Y.house.price.of.unit.area              -0.02858717         0.087490606
```

```
##                                         X2.house.age
## No                                       -0.03280811
## X1.transaction.date                       0.01754877
## X2.house.age                              1.00000000
## X3.distance.to.the.nearest.MRT.station    0.02562205
## X4.number.of.convenience.stores           0.04959251
## X5.latitude                               0.05441990
## X6.longitude                             -0.04852005
## Y.house.price.of.unit.area               -0.21056705
##
X3.distance.to.the.nearest.MRT.station
## No                                                                -
0.01357349
## X1.transaction.date
0.06087995
## X2.house.age
0.02562205
## X3.distance.to.the.nearest.MRT.station
1.00000000
## X4.number.of.convenience.stores                                   -
0.60251914
## X5.latitude                                                       -
0.59106657
## X6.longitude                                                      -
0.80631677
## Y.house.price.of.unit.area                                        -
0.67361286
##                                         X4.number.of.convenience.stores
## No                                                        -0.012698946
## X1.transaction.date                                        0.009635445
## X2.house.age                                               0.049592513
## X3.distance.to.the.nearest.MRT.station                    -0.602519145
## X4.number.of.convenience.stores                            1.000000000
## X5.latitude                                                0.444143306
## X6.longitude                                               0.449099007
## Y.house.price.of.unit.area                                 0.571004911
##                                         X5.latitude X6.longitude
## No                                       -0.01010966  -0.01105928
## X1.transaction.date                       0.03505776  -0.04108178
## X2.house.age                              0.05441990  -0.04852005
## X3.distance.to.the.nearest.MRT.station   -0.59106657  -0.80631677
## X4.number.of.convenience.stores           0.44414331   0.44909901
## X5.latitude                               1.00000000   0.41292394
## X6.longitude                              0.41292394   1.00000000
## Y.house.price.of.unit.area                0.54630665   0.52328651
##                                         Y.house.price.of.unit.area
## No                                                      -0.02858717
## X1.transaction.date                                      0.08749061
## X2.house.age                                            -0.21056705
## X3.distance.to.the.nearest.MRT.station                  -0.67361286
```

```
## X4.number.of.convenience.stores                         0.57100491
## X5.latitude                                             0.54630665
## X6.longitude                                            0.52328651
## Y.house.price.of.unit.area                              1.00000000
```

7. Using Model 3 to make predictions

```
r_new_data = data.frame(X1.transaction.date = c(2020.917,2021.111),
X2.house.age = c(1.0,5.0), X3.distance.to.the.nearest.MRT.station = c(10.0,
800),X4.number.of.convenience.stores = c(6, 20), X5.latitude = c(25, 29),
X6.longitude = c(120, 100))
r_new_data
```

```
##   X1.transaction.date X2.house.age X3.distance.to.the.nearest.MRT.station
## 1            2020.917            1                                      10
## 2            2021.111            5                                     800
##   X4.number.of.convenience.stores X5.latitude X6.longitude
## 1                               6          25          120
## 2                              20          29          100
```

```
prediction1 = predict(r_model3, r_new_data, interval = 'prediction', level =
0.95)
prediction1
```

```
##         fit       lwr      upr
## 1  5.195624  4.458539  5.93271
## 2 36.995119 28.272660 45.71758
```