Real Estate Transactions in Seoul

Group 7 (Choi Hae Min, Kim Ji Hyun, Son Sung Mo, Supatach Vanichayangkuranont)

Introdutction (Reasons for choosing the topic, explaination about data)

Real estate is very important as the necessities of life and an asset. However, there are not enough information about its present price, because it has only few transactions.Therefore, we wanted to give proper information about real estate price through this project.

Our dataset is about all real estate transactions in Seoul, from January 2018 to October 2022. It has 640,000 observations and 21 variables. We took it from Seoul open data and this is its URL.

https://data.seoul.go.kr/dataList/OA-21275/S/1/datasetView.do (https://data.seoul.go.kr/dataList/OA-21275/S/1/datasetView.do)

```
df <- read.csv("(translated) Real Estate Transactions in Seoul.csv", fileEncoding = "euc-kr")
head(df)
```

```
##   Year Gu.Code        Gu Dong.Code       Dong 지번구분 지번구분명 본번 부번
## 1 2022   11215  Gwangjin     10700     화양동        1       대지  113    1
## 2 2022   11500   Gangseo     10300     화곡동        1       대지  956    1
## 3 2022   11410 Seodaemun     11200     대현동        1       대지   90   58
## 4 2022   11410 Seodaemun     11700     연희동        1       대지  432    7
## 5 2022   11305   Gangbuk     10300     수유동        1       대지  516  127
## 6 2022   11290  Seongbuk     10800 동소문동5가        1       대지  120    0
##       Building.Name Contract.Date Price..1000won. Building.Area... Land.Area...
## 1        광진코지웰      20221027           13000            14.73         0.00
## 2          영주주택      20221027           14500            44.64        19.50
## 3           (90-58)      20221027           15000            18.98        23.03
## 4          우방빌라      20221027           12000            31.24        20.40
## 5 삼광빌라(516-127)      20221027           18000            54.27        69.07
## 6    돈암동일하이빌      20221027          100000            84.96         0.00
##   floor 권리구분 Cencellation.Date Construction.Year Building.Purpose
## 1     8                        NA              2014        apartment
## 2     4                        NA              2002        row house
## 3     4                        NA              2015 studio apartment
## 4    -1                        NA              1993        row house
## 5     2                        NA              1989        row house
## 6     9                        NA              2006        apartment
##   Transaction Region.of.Real.Estate.Agency
## 1   mediation               Seoul Gwangjin
## 2   mediation                Seoul Gangseo
## 3   mediation              Seoul Seodaemun
## 4   mediation              Seoul Seodaemun
## 5      direct
## 6   mediation               Seoul Seongbuk
```

Open basic Library

```
library(tidyverse)
library(lubridate)
library(ggcorrplot)
```

Preprocessing

Rename the column for intuitive understanding and delete unusable columns

```
names(df)[12] <- "Price.10000.won"
names(df)[13] <- "Building.Area"
names(df)[14] <- "Land.Area"
df <- df[-c(4:10)]
df <- df[-c(2,9,14)]
```

Check NA values of data

```
colSums(is.na(df))
```

```
##              Year                Gu     Contract.Date   Price.10000.won
##                 0                 0                 0                 0
##     Building.Area         Land.Area             floor Cencellation.Date
##                 0            158534             48821            623092
## Construction.Year  Building.Purpose       Transaction
##              2585                 0                 0
```

There are many NA values in land.area, floor, cancellation date, construction year

check number of unique value in each columns

```
apply(df,2,n_distinct)
```

```
##              Year                Gu     Contract.Date   Price.10000.won
##                 6                25              1741             14051
##     Building.Area         Land.Area             floor Cencellation.Date
##             28401             12556                76               886
## Construction.Year  Building.Purpose       Transaction
##               107                 4                 3
```

The number of unique values in the columns is as above.

Check unique value in some columns

```
unique(df$Gu) # (25 distinctions)
```

```
##  [1] "Gwangjin"     "Gangseo"      "Seodaemun"    "Gangbuk"      "Seongbuk"
##  [6] "Dongdaemun"   "Dobong"       "Seongdong"    "Gangdong"     "Nowon"
## [11] "Seocho"       "Mapo"         "Guro"         "Eunpyoung"    "Yangcheon"
## [16] "Gangnam"      "Geumcheon"    "Jongno"       "Songpa"       "Gwanak"
## [21] "Jungnang"     "Dongjak"      "Youngdeungpo" "Yongsan"      "Junggu"
```

```
n_distinct(df$Dong) # (420 dong)
```

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

```
sort(unique(df$floor)) # (floor exists from -3 to 73, and there are also NA values.)
```

```
##  [1] -3 -2 -1  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
## [26] 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
## [51] 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 71 73
```

```
sort(unique(df$Construction.Year)) #(Construction years exist from 1900 to 2022. 0 seems to be
 an outlier.)
```

```
##   [1]    0 1900 1901 1909 1912 1920 1921 1922 1923 1926 1927 1928 1929 1930 1931
##  [16] 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946
##  [31] 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961
##  [46] 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976
##  [61] 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991
##  [76] 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006
##  [91] 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021
## [106] 2022
```

```
unique(df$Building.Purpose) # apartment, row house, studio apartment, multi household
```

```
## [1] "apartment"             "row house"             "studio apartment"
## [4] "multi household house"
```

```
unique(df$Transaction) # mediation, direct, ""
```

```
## [1] "mediation" "direct"    ""
```

```
sum(is.na(df$Cencellation.Date) == FALSE)
```

```
## [1] 16908
```

In Gu and Dong column, there are 25 and 420 unique values. We assume that regional information also has a large impact on real estate prices. Therefore, we plan to create a derived variable Gwon that is easy to analyze using the Gu column.

Floor values exist from -3 to 73, and there are also NA values. A negative number means underground. Construction years values exist from 1900 to 2022. 0 seems to be a missing value.

There are apartment, row house, studio apartment, multi household in Building Purpose. “Multi house hold” seems to have some errors in the process of translation. It’s more appropriate to say “Single-family home”,so we will change it. As a categorical variable, this column will have a great impact on the analysis of real estate values.

There are mediation, direct, “” in Transaction column. “” seems to be a missing value. so we will change it to NA.

If the cancellation date column has a value, the transaction is a canceled transaction and should be excluded from the data. And There are 16908 cancellation dates in the column. so we will exclude that rows.

Process the data based on checking the data characterization

```
df$Building.Purpose[df$Building.Purpose == "multi household house"] <- "Single-family home"

df$Transaction[df$Transaction == ""] <- NA
df <- df[is.na(df$Cencellation.Date) == TRUE,] # Except when the transaction is cancelled. Update the data
```

Generate Base Rate column

To create the base rate column, we used data containing information about base rates. The base_rate data has the information of the base rate and the year and month to which the interest rate is applied. Here is URL of base rate dataset.

ecos.bok.or.kr/#/Short/89ebfb

To add the standard interest rate information to the real_estate data, the year variable of the existing data was used to create a ‘ym’ variable with only the year and month, and the two data were combined using ‘ym’ as a key.

Improvement: Base rate refers to a country’s representative interest rate determined by the central bank of each country. All buyers expend the interest costs as an opportunity cost. Therefore, when the base rate rises, interest costs increase, demand for real estate decreases and real estate prices fall. So we expected negative correlation between price and base rate.

```
base_rate <- read.csv("base rate (18-1-20_22-11-11) month.csv")
head(base_rate)
```

```
##    month base.rate
## 1 Jan-18       1.5
## 2 Feb-18       1.5
## 3 Mar-18       1.5
## 4 Apr-18       1.5
## 5 May-18       1.5
## 6 Jun-18       1.5
```

```
base_rate$ym<- my(base_rate$month)
str(base_rate)
```

```
## 'data.frame':    58 obs. of  3 variables:
##  $ month    : chr  "Jan-18" "Feb-18" "Mar-18" "Apr-18" ...
##  $ base.rate: num  1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 ...
##  $ ym       : Date, format: "2018-01-01" "2018-02-01" ...
```

```
df$ym <- ymd(df$Contract.Date)

base_rate$ym <- substr(base_rate$ym,1,7)
df$ym <- substr(df$ym,1,7)

df <- inner_join(df,base_rate,key = 'ym')
```

```
## Joining, by = "ym"
```

EDA Through Visualization

Now we will visualize the data to derive insights. Below are the results of visually showing the relationship between various variables using ggplot.

Mean price versus building purpose

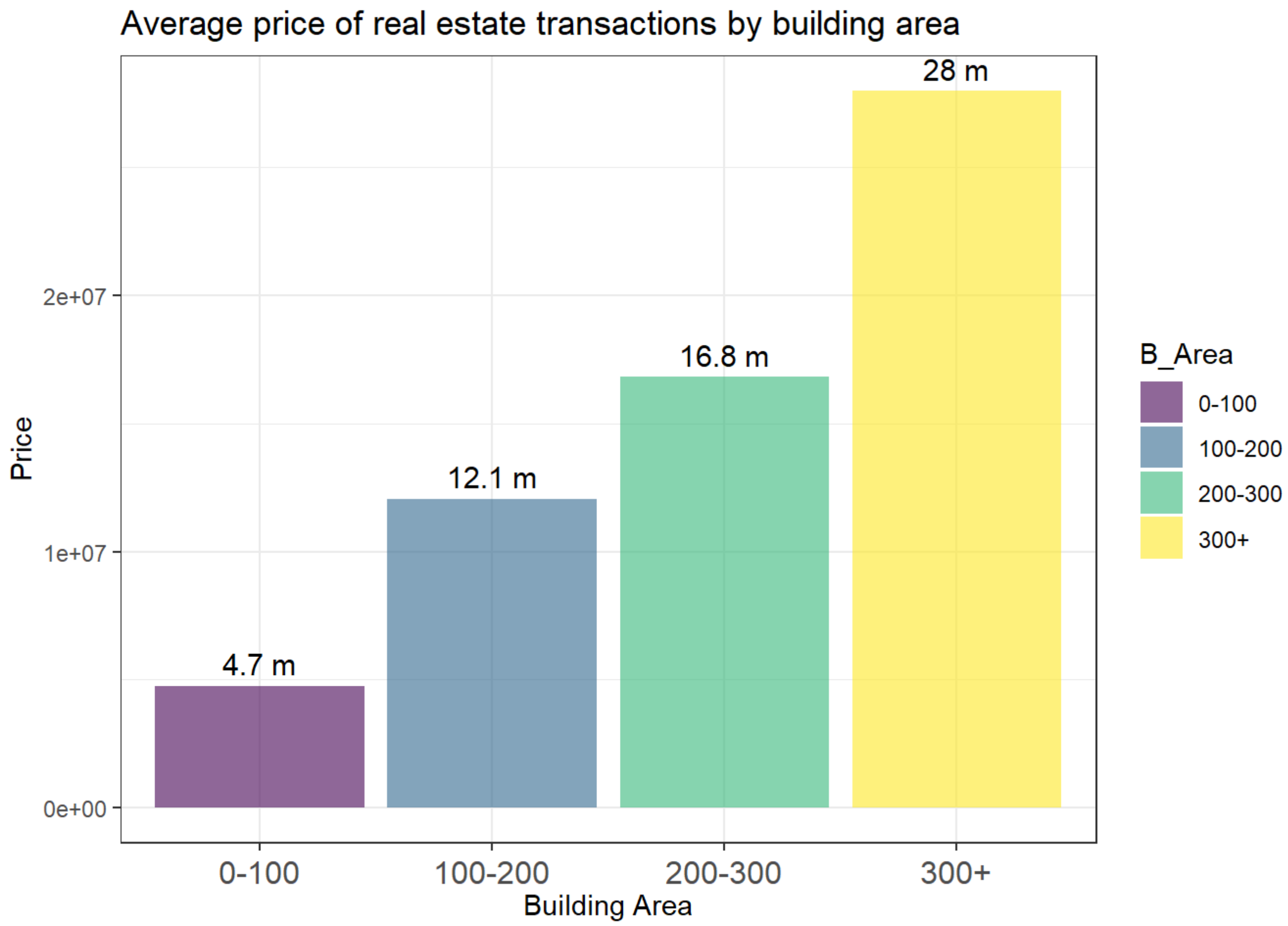
```
df %>%
  group_by(Building.Purpose) %>%
  summarise(mean_price = mean(Price.10000.won)) %>%
  ggplot(aes(x = Building.Purpose, y = mean_price, fill = Building.Purpose, alpha = 0.6))+
  geom_col()+
  theme_bw()+
  coord_flip()+
  theme(legend.position = 'none')+
  labs(x='Building purpose',y='Mean Price',title='Mean Price of real estate transactions by building purpose')
```

Looking at the above plot, it can be seen that Single-family homes are being traded at the highest price. Given that there is a difference in price depending on the purpose of the building, the building purpose variable can serve as an explanatory variable.

Average price of real estate transactions by building area

```
building_area_agg <- df %>%
  mutate(B_Area = case_when(
    Building.Area <= 100 ~ "0-100",
    Building.Area > 100 & Building.Area <= 200 ~ "100-200",
    Building.Area > 200 & Building.Area <= 300 ~ "200-300",
    Building.Area > 300 ~ "300+"
  ))

building_area_agg$B_Area <- factor(building_area_agg$B_Area, levels=c("0-100", "100-200", "200-300", "300+"), ordered = TRUE)
building_area_agg %>%
  group_by(B_Area) %>%
  summarise(Price = mean(Price.10000.won) * 100) %>%
  ggplot(aes(x = B_Area, y = Price, fill = B_Area))+
  geom_col(alpha = 0.6)+
  theme_bw()+
  labs(x='Building Area',y='Price',title='Average price of real estate transactions by building area') +
  geom_text(aes(x= B_Area, y= Price, label = paste(round(Price/1000000,1), 'm')), position = position_dodge(width = 1),vjust = -0.5, size = 4,check_overlap = T) +
  theme(axis.text.x = element_text(vjust = 0.5,size =12))
```



Clearly, the plot shows that there is a positive correlation between building area and price.

Integration Gu to Gwon

In the regression analysis, we tried to use Gu as a categorical variable. However, too many category labels can be a problem when modeling. Based on the fact that Seoul City integrates 25 Gu into 5 Gwon, we will create Gwon variables. Since there was no clean way, we handled it with 'ifelse' statement.

```
df$Gwon <- ifelse((df$Gu == "Jongno") |(df$Gu == "Junggu") | (df$Gu == "Yongsan"), "downtown area",
                  ifelse((df$Gu =="Gangdong") | (df$Gu == "Eunpyoung") |(df$Gu == "Seodaemun") | (df$Gu == "Mapo"),"northwest area",
                         ifelse((df$Gu == "Gangbuk") | (df$Gu == "Dobong") |(df$Gu == "Nowon") | (df$Gu == "Seongbuk") | (df$Gu == "Dongdaemun") |  (df$Gu == "Jungnang") | (df$Gu == "Seongdong") | (df$Gu == "Gwangjin"), "northeast area",
                                ifelse((df$Gu == "Gangseo") | (df$Gu == "Yangcheon") | (df$Gu == "Youngdeungpo") | (df$Gu == "Guro") | (df$Gu == "Geumcheon") | (df$Gu == "Dongjak") | (df$Gu == "Gwanak"), "southwest area",
                                       ifelse((df$GU == "Songpa") |(df$Gu == "Seocho") | (df$Gu == "Gangnam")| (df$Gu == "Seocho") | (df$GU == "Songpa") | (df$Gu == "Gangdong"), "southeast area", "others")))))

df$Gwon[df$Gu == "Seocho"] <- "southeast area"
df$Gwon[df$Gu == "Gangnam"] <- "southeast area"
df$Gwon[df$Gu == "Songpa"] <- "southeast area"
```

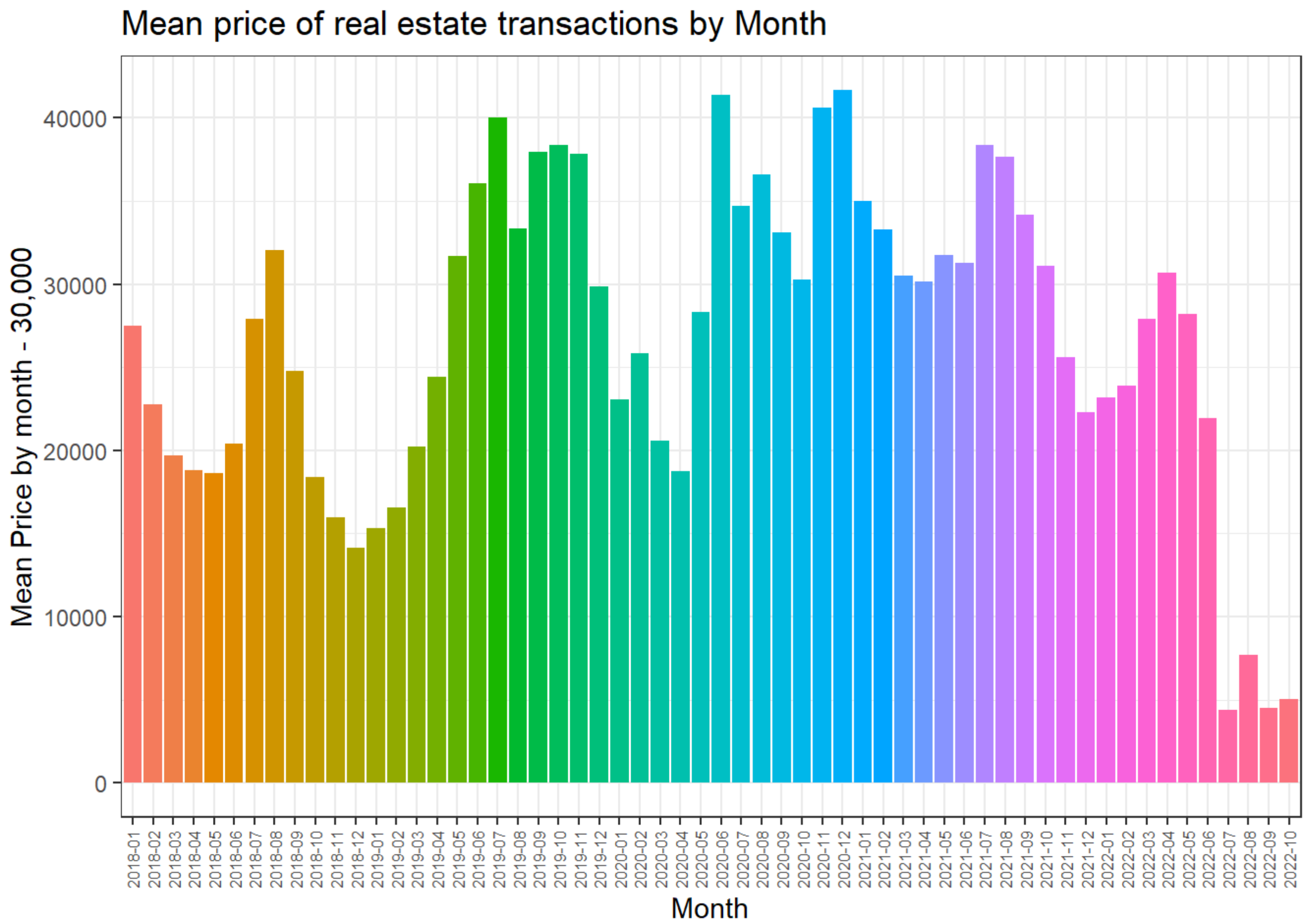
Average Real Estate Price by Gwon

```
df %>% select(Gwon, Price.10000.won) %>%
  group_by(Gwon) %>%
  summarise(mean_price = mean(Price.10000.won)) %>%
  arrange(desc(mean_price)) %>%
  ggplot(aes(x=Gwon, y=mean_price, fill=Gwon)) +
  geom_bar(stat="identity") +
  geom_hline(aes(yintercept = mean(df$Price.10000.won)), linetype= "dashed", size=1)+
  theme_bw() +
  theme(axis.text.x=element_text(angle=15, hjust=1)) +
  labs(title="Average Real Estate Price by Gwon", y="Average Price (10,000 won)")
```

Looking at the above plot, you can see that there is a difference in price for each Gwon.

Mean price of real estate transactions by Month

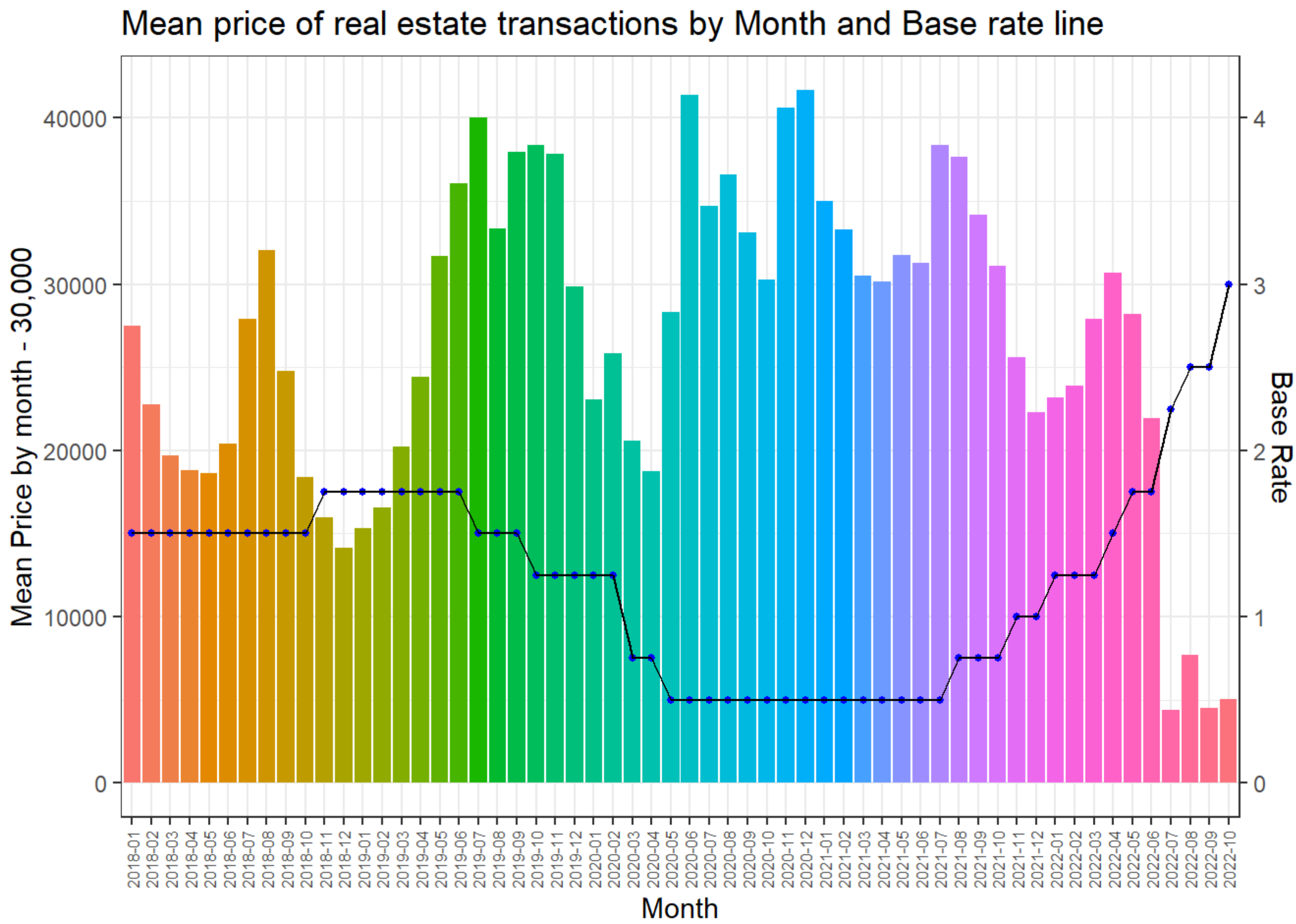
```
df %>%
  group_by(ym) %>%
  summarise(count = n(),
            base_rate = base.rate,
            mean_price = mean(Price.10000.won)) %>%
  unique() %>%
  ggplot(aes(x = ym, y = mean_price - 30000, fill = ym,group = 1))+
  geom_col()+
  theme_bw()+
  theme(legend.position = 'none')+
  labs(x='Month',y='Mean Price by month - 30,000',title='Mean price of real estate transactions
by Month')+
  theme(axis.text.x = element_text(angle= 90, vjust = 0.5,size =6))
```



The graph above shows the average price over time. No particular pattern can be found in this graph. But in our data, it is the base rate that changes over time. Let's look at the graph above by adding information on the base interest rate.

Mean price of real estate transactions by Month and Base rate line

```
df %>%
  group_by(ym) %>%
  summarise(count = n(),
            base_rate = base.rate,
            mean_price = mean(Price.10000.won)) %>%
  unique() %>%
  ggplot(aes(x = ym, y = mean_price - 30000, fill = ym,group = 1))+
  geom_col()+
  theme_bw()+
  theme(legend.position = 'none')+
  labs(x='Month',y='Mean Price by month - 30,000',title='Mean price of real estate transactions
by Month and Base rate line')+
  theme(axis.text.x = element_text(angle= 90, vjust = 0.5,size =6))+
  geom_point(aes(y = base_rate / 0.0001), shape = 21, size=1, fill='blue', colour='blue')+
  geom_line(mapping = aes(x= ym, y = base_rate / 0.0001)) +
  scale_y_continuous(sec.axis = sec_axis(~./10000, name = "Base Rate"))
```



In this case, the lower the base interest rate, the higher the price. Through this, it was judged that the base rate would be an important variable in predicting the transaction price.

Modeling

Based on the previous EDA process, we wanted to create a linear regression model which predict price through year, Gwon, building area, building purpose and base rate. Therefore, we separated the data into only the necessary variables.

```
df <- df %>% select(Year, Gwon, Price.10000.won, Building.Area, Building.Purpose, base.rate)

df$Price.10000.won <- df$Price.10000.won * 10000
names(df)[3] <- "Price"
```

Improvement:

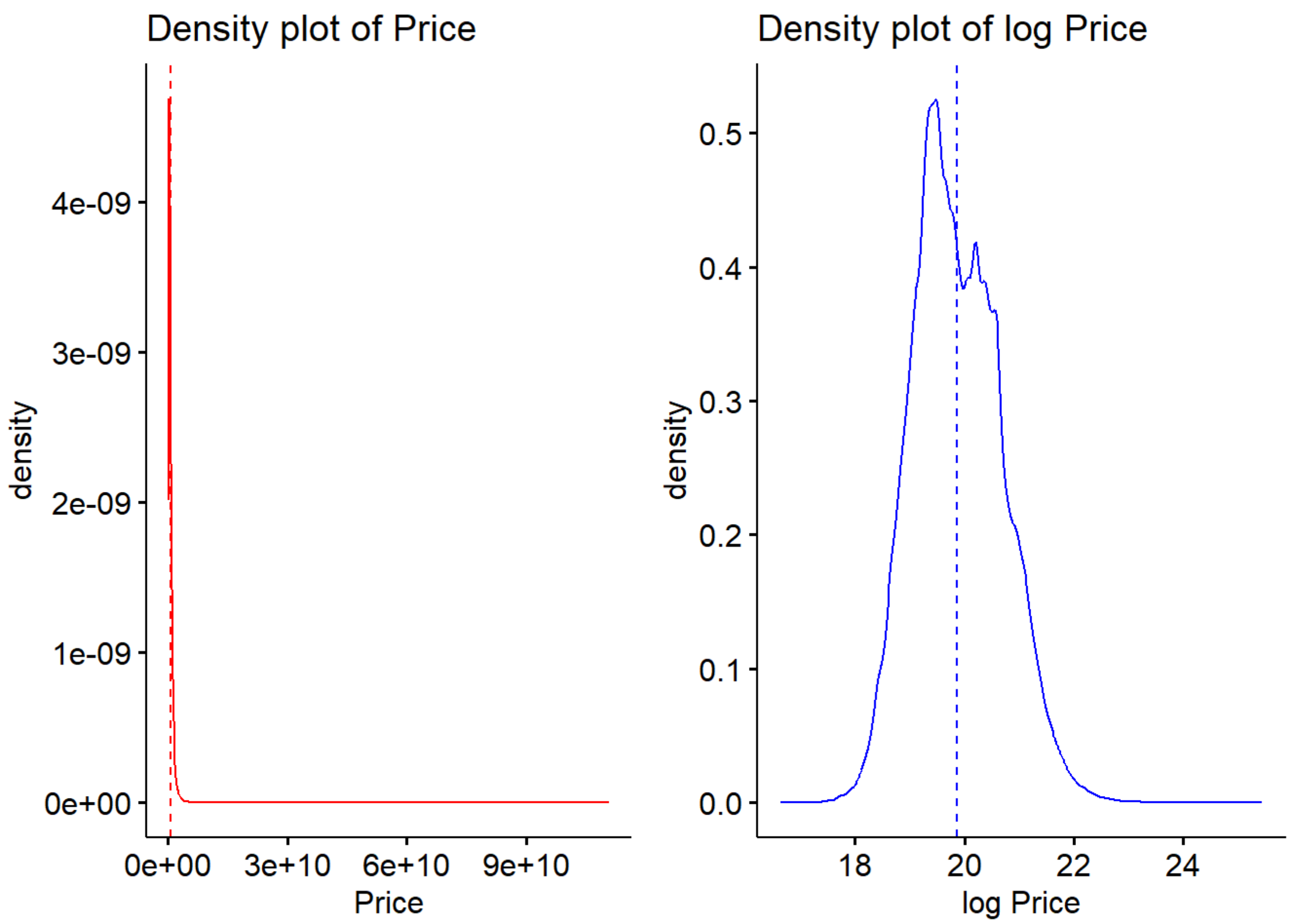
Linear regression assumes normality of the data. However, if you look at the graph below, the price distribution of the data does not satisfy normality at all. Log transformation is a way to solve this problem. Comparing the two plots below, the log transformation is not perfect, but it can be seen that the distribution is somewhat normal. Therefore, the analysis was carried out by log-transforming the price.

```
library(ggpubr)
par(mfrow =c(1,2))
p1 <- ggdensity(df$Price,
          add = "mean",
          color = 'red',
          title = 'Density plot of Price',
          xlab = 'Price')
p2 <- ggdensity(log(df$Price),
          add = "mean",
          color = 'blue',
          title = 'Density plot of log Price',
          xlab = 'log Price')
library(gridExtra)
```

```
##
## 다음의 패키지를 부착합니다: 'gridExtra'
```

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

```
grid.arrange(p1,p2,ncol=2)
```



Training & Test Set Split

We separated the data into a training set and a test set to check how well the model predicts with the test set.

```
set.seed(18)

n_row <- round(dim(df)[1]*0.7)

train_idx <- sample(1:dim(df)[1],n_row,replace = F)

df_train <- df[train_idx,]
df_test <- df[-train_idx,]
```

First Model : log(price) ~ all.

```
model1 <- lm(log(Price) ~ .,data = df_train)

summary(model1)
```

```
##
## Call:
## lm(formula = log(Price) ~ ., data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7369  -0.2799   0.0198   0.2992   2.7993
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         -1.137e+02  1.434e+00  -79.29   <2e-16 ***
## Year                                 6.637e-02  7.097e-04   93.52   <2e-16 ***
## Gwonnortheast area                  -4.608e-01  3.432e-03 -134.29   <2e-16 ***
## Gwonnorthwest area                  -2.918e-01  3.589e-03  -81.32   <2e-16 ***
## Gwonsoutheast area                   1.760e-01  3.727e-03   47.22   <2e-16 ***
## Gwonsouthwest area                  -3.943e-01  3.417e-03 -115.38   <2e-16 ***
## Building.Area                        4.795e-03  1.357e-05  353.24   <2e-16 ***
## Building.Purposerow house           -8.601e-01  1.692e-03 -508.46   <2e-16 ***
## Building.PurposeSingle-family home -2.920e-01  3.315e-03  -88.10   <2e-16 ***
## Building.Purposestudio apartment   -1.034e+00  2.617e-03 -395.13   <2e-16 ***
## base.rate                           -6.279e-02  1.748e-03  -35.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4795 on 436153 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6407
## F-statistic: 7.777e+04 on 10 and 436153 DF,  p-value: < 2.2e-16
```

When the price was used as the dependent variable and the rest of the variables were used as explanatory variables, the results confirmed that all variables were significant, and the Adjusted R-squared at this time was 0.6407. However, we suspected that the Year variable was categorical rather than continuous numerical data, and the results of the modeling were as follows.

Second Model: Change Neumeric Year to Categorical Year

```
model2 <- lm(log(Price) ~ as.factor(Year)+Gwon+Building.Area+Building.Purpose+base.rate, data =
df_train)
summary(model2)
```

```
##
## Call:
## lm(formula = log(Price) ~ as.factor(Year) + Gwon + Building.Area +
##     Building.Purpose + base.rate, data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7243  -0.2787   0.0205   0.2988   2.7640
##
## Coefficients:
##                                    Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                       2.034e+01  2.767e-01   73.525   <2e-16 ***
## as.factor(Year)2018              -7.512e-02  2.766e-01   -0.272    0.786
## as.factor(Year)2019               4.033e-02  2.766e-01    0.146    0.884
## as.factor(Year)2020               2.694e-02  2.766e-01    0.097    0.922
## as.factor(Year)2021               1.093e-01  2.766e-01    0.395    0.693
## as.factor(Year)2022               1.979e-01  2.766e-01    0.715    0.474
## Gwonnortheast area               -4.604e-01  3.429e-03 -134.269   <2e-16 ***
## Gwonnorthwest area               -2.915e-01  3.586e-03  -81.291   <2e-16 ***
## Gwonsoutheast area                1.744e-01  3.724e-03   46.837   <2e-16 ***
## Gwonsouthwest area               -3.937e-01  3.414e-03 -115.331   <2e-16 ***
## Building.Area                     4.795e-03  1.356e-05  353.601   <2e-16 ***
## Building.Purposerow house        -8.593e-01  1.694e-03 -507.172   <2e-16 ***
## Building.PurposeSingle-family home -2.907e-01  3.313e-03  -87.756   <2e-16 ***
## Building.Purposestudio apartment  -1.033e+00  2.621e-03 -393.995   <2e-16 ***
## base.rate                        -1.021e-01  3.095e-03  -33.006   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4791 on 436149 degrees of freedom
## Multiple R-squared:  0.6414, Adjusted R-squared:  0.6414
## F-statistic: 5.571e+04 on 14 and 436149 DF,  p-value: < 2.2e-16
```

In this case, the adjusted R squared value a little increased, but the result was that the Year variable as a factor was not significant. Therefore, we tried to proceed with the variable selection method, and this time, stepwise was applied.

```
step(lm(log(Price) ~ as.factor(Year)+Gwon+Building.Area+Building.Purpose+base.rate, data = df_t
rain),scope = list(lower = ~1, upper = ~.),direction = 'both')
```

```
## Start:  AIC=-641913.5
## log(Price) ~ as.factor(Year) + Gwon + Building.Area + Building.Purpose +
##     base.rate
##
##                     Df Sum of Sq    RSS     AIC
## <none>                           100105 -641914
## - base.rate          1       250 100356 -640827
## - as.factor(Year)    5      2206 102311 -632418
## - Gwon               4     19780 119885 -563277
## - Building.Area      1     28698 128803 -531975
## - Building.Purpose   3     71760 171866 -406178
```

```
##
## Call:
## lm(formula = log(Price) ~ as.factor(Year) + Gwon + Building.Area +
##     Building.Purpose + base.rate, data = df_train)
##
## Coefficients:
##                       (Intercept)                as.factor(Year)2018
##                         20.341694                          -0.075122
##                as.factor(Year)2019                as.factor(Year)2020
##                          0.040326                           0.026942
##                as.factor(Year)2021                as.factor(Year)2022
##                          0.109266                           0.197856
##                 Gwonnortheast area                 Gwonnorthwest area
##                         -0.460351                          -0.291471
##                 Gwonsoutheast area                 Gwonsouthwest area
##                          0.174407                          -0.393748
##                      Building.Area           Building.Purposerow house
##                          0.004795                          -0.859309
## Building.PurposeSingle-family home    Building.Purposestudio apartment
##                         -0.290748                          -1.032522
##                          base.rate
##                         -0.102139
```

As a result of variable selection, it can be confirmed that all variables are selected. We also tried to consider the interaction term case, thus we added Building.Area multiplied by base.rate variable as a new independent variable.

Third Model: Adding Interaction Term.

```
model3 <- lm(log(Price) ~ as.factor(Year)+Gwon+Building.Area+Building.Purpose+base.rate+Buildin
g.Area * base.rate , data = df_train)
summary(model3)
```

```
##
## Call:
## lm(formula = log(Price) ~ as.factor(Year) + Gwon + Building.Area +
##     Building.Purpose + base.rate + Building.Area * base.rate,
##     data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0560  -0.2788   0.0202   0.2988   2.7537
##
## Coefficients:
##                                  Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                     2.035e+01  2.766e-01   73.569   <2e-16 ***
## as.factor(Year)2018            -7.051e-02  2.766e-01   -0.255    0.799
## as.factor(Year)2019             4.517e-02  2.766e-01    0.163    0.870
## as.factor(Year)2020             3.288e-02  2.766e-01    0.119    0.905
## as.factor(Year)2021             1.149e-01  2.766e-01    0.415    0.678
## as.factor(Year)2022             2.047e-01  2.766e-01    0.740    0.459
## Gwonnortheast area             -4.605e-01  3.428e-03 -134.319   <2e-16 ***
## Gwonnorthwest area             -2.915e-01  3.585e-03  -81.321   <2e-16 ***
## Gwonsoutheast area              1.742e-01  3.723e-03   46.789   <2e-16 ***
## Gwonsouthwest area             -3.939e-01  3.414e-03 -115.389   <2e-16 ***
## Building.Area                   4.561e-03  2.735e-05  166.800   <2e-16 ***
## Building.Purposerow house      -8.594e-01  1.694e-03 -507.258   <2e-16 ***
## Building.PurposeSingle-family home -2.917e-01  3.314e-03  -88.005   <2e-16 ***
## Building.Purposestudio apartment   -1.032e+00  2.620e-03 -394.025   <2e-16 ***
## base.rate                      -1.165e-01  3.422e-03  -34.051   <2e-16 ***
## Building.Area:base.rate         2.184e-04  2.219e-05    9.841   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.479 on 436148 degrees of freedom
## Multiple R-squared:  0.6415, Adjusted R-squared:  0.6414
## F-statistic: 5.202e+04 on 15 and 436148 DF,  p-value: < 2.2e-16
```

In this case, the adjusted R squared value is same as the previous one. Again, we did variable selection.

```
step(lm(Price ~ as.factor(Year)+Gwon+Building.Area+Building.Purpose+base.rate+Building.Area * b
ase.rate , data = df_train),scope = list(lower = ~1, upper = ~.),direction = 'both')
```

```
## Start:  AIC=17368646
## Price ~ as.factor(Year) + Gwon + Building.Area + Building.Purpose +
##     base.rate + Building.Area * base.rate
##
##                           Df  Sum of Sq        RSS      AIC
## <none>                                  8.5865e+22 17368646
## - Building.Area:base.rate  1 2.4142e+20 8.6107e+22 17369868
## - as.factor(Year)          5 1.8016e+21 8.7667e+22 17377692
## - Building.Purpose         3 1.4213e+22 1.0008e+23 17435449
## - Gwon                     4 1.4832e+22 1.0070e+23 17438134
```

```
##
## Call:
## lm(formula = Price ~ as.factor(Year) + Gwon + Building.Area +
##     Building.Purpose + base.rate + Building.Area * base.rate,
##     data = df_train)
##
## Coefficients:
##                      (Intercept)                as.factor(Year)2018
##                        395236887                          157485529
##              as.factor(Year)2019                as.factor(Year)2020
##                        251615216                          232922629
##              as.factor(Year)2021                as.factor(Year)2022
##                        321594169                          399433546
##               Gwonnortheast area                 Gwonnorthwest area
##                       -359675324                         -253741555
##               Gwonsoutheast area                 Gwonsouthwest area
##                        193507325                         -310782497
##                    Building.Area           Building.Purposerow house
##                          6606338                         -375920228
## Building.PurposeSingle-family home   Building.Purposestudio apartment
##                       -294156195                         -423431520
##                        base.rate            Building.Area:base.rate
##                        -22599445                            -719717
```

All variables were selected in this case as well.

In this case, the problem of multicollinearity may arise due to the interaction term. After checking the multicollinearity problem, we selected the final model.

Multicollinearity Check

To check multicollinearity, correlation coefficients between explanatory variables should be checked. In our case, since both categorical and numeric variables exist, we will find Pearson's correlation coefficient between numeric variables, polyserial correlation between numerical and categorical variables, and Cramer's V between categorical and categorical variables. Based on the correlation coefficients obtained between the variables, we created a matrix and visualized it.

numeric vs numeric

```
num_df <- df %>% select(Building.Area, base.rate) %>%
  mutate(intersection = Building.Area * base.rate)

num_cor <- round(cor(num_df),2)
```

numeric vs categorical

```
library(polycor)
round(polyserial(df$Building.Area, df$Year, ML = FALSE, control = list(),
  std.err = FALSE, maxcor=.9999, bins=4, start, thresholds=FALSE),2)
```

```
## [1] -0.08
```

```
round(polyserial(df$Building.Area, df$Gwon, ML = FALSE, control = list(),
  std.err = FALSE, maxcor=.9999, bins=4, start, thresholds=FALSE),2)
```

```
## [1] -0.03
```

```
round(polyserial(df$Building.Area, df$Building.Purpose, ML = FALSE, control = list(),
  std.err = FALSE, maxcor=.9999, bins=4, start, thresholds=FALSE),2)
```

```
## [1] -0.01
```

```
round(polyserial(df$base.rate, df$Year, ML = FALSE, control = list(),
  std.err = FALSE, maxcor=.9999, bins=4, start, thresholds=FALSE),2)
```

```
## [1] -0.56
```

```
round(polyserial(df$base.rate, df$Gwon, ML = FALSE, control = list(),
  std.err = FALSE, maxcor=.9999, bins=4, start, thresholds=FALSE),2)
```

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

```
round(polyserial(df$base.rate, df$Building.Purpose, ML = FALSE, control = list(),
  std.err = FALSE, maxcor=.9999, bins=4, start, thresholds=FALSE),2)
```

```
## [1] -0.02
```

```
round(polyserial(df$Building.Area * df$base.rate, df$Building.Purpose, ML = FALSE, control = li
st(),
  std.err = FALSE, maxcor=.9999, bins=4, start, thresholds=FALSE),2)
```

```
## [1] -0.02
```

```
round(polyserial(df$Building.Area * df$base.rate, df$Gwon, ML = FALSE, control = list(),
  std.err = FALSE, maxcor=.9999, bins=4, start, thresholds=FALSE),2)
```

```
## [1] -0.03
```

```
round(polyserial(df$Building.Area * df$base.rate, df$Year, ML = FALSE, control = list(),
  std.err = FALSE, maxcor=.9999, bins=4, start, thresholds=FALSE),2)
```

```
## [1] -0.3
```

Categorical vs Categorical

```
library(DescTools)
round(CramerV(df$Gwon, df$Building.Purpose),2)
```

```
## [1] 0.12
```

```
round(CramerV(df$Year, df$Building.Purpose),2)
```

```
## [1] 0.13
```

```
round(CramerV(df$Year, df$Gwon),2)
```

```
## [1] 0.03
```

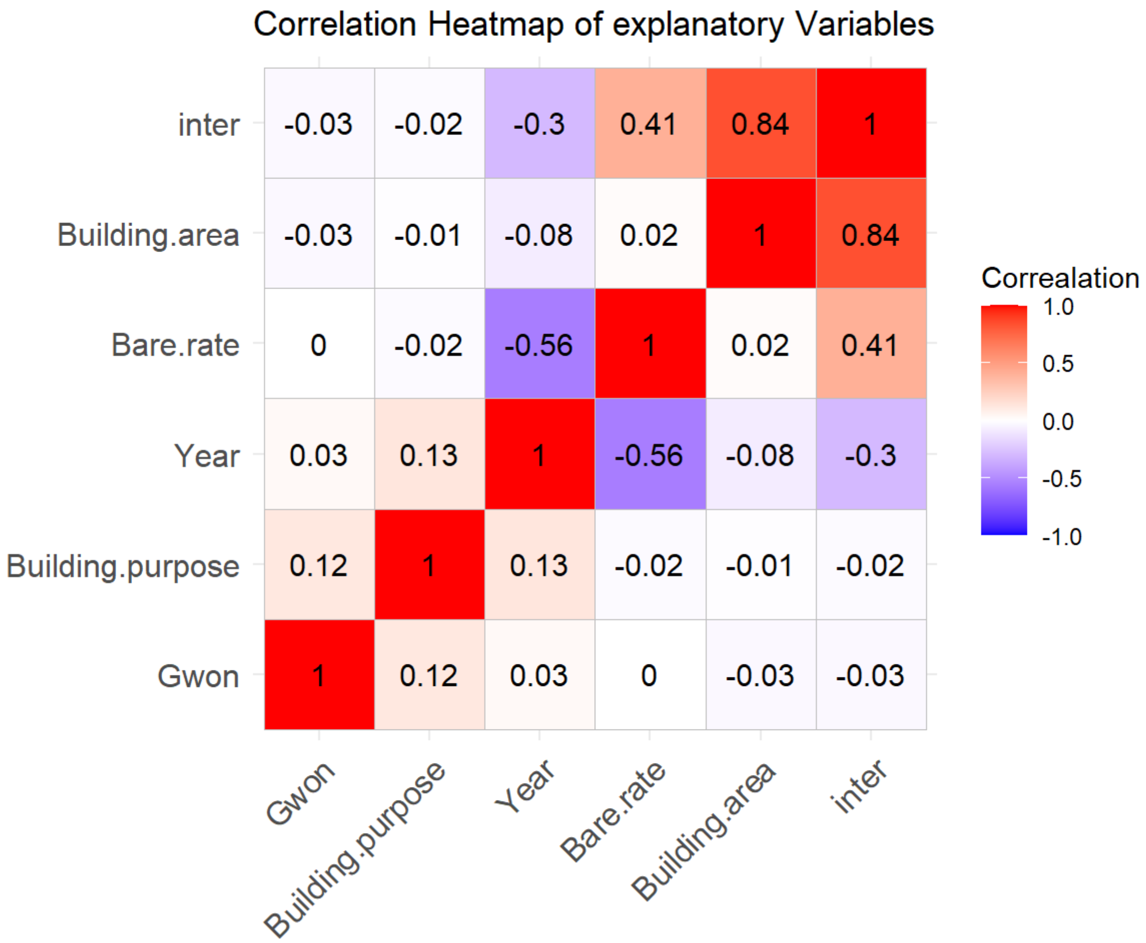
The matrix combining the correlation coefficient values is as follows.

```
Building.area <- c(1.00,0.02,0.84,-0.01,-0.03,-0.08)
Bare.rate <- c(0.02,1.00,0.41,-0.02,0,-0.56)
inter <- c(0.84,0.41,1.00,-0.02,-0.03,-0.3)
Building.purpose <- c(-0.01,-0.02,-0.02,1.00,0.12,0.13)
Gwon <- c(-0.03,0,-0.03,0.12,1,0.03)
Year <- c(-0.08,-0.56,-0.3,0.13,0.03,1.00)

cor <- as.matrix(cbind(Building.area,Bare.rate,inter,Building.purpose,Gwon,Year))
rownames(cor) <- c("Building.area","Bare.rate" ,"inter" ,"Building.purpose" ,"Gwon" ,"Year")
```

Heatmap of Explanatory Variables

```
library(ggcorrplot)
ggcorrplot(cor,hc.order = TRUE,
   lab = TRUE,title="Correlation Heatmap of explanatory Variables", legend.title = "Correalatio
n")
```



The correlation coefficient between the interaction term and the building area was shown to be high. Even if the interaction term originates from the building area, there may be a problem of multicollinearity because the value of the correlation coefficient is too large, and it is judged to be an unnecessary variable. Therefore, Model 2 was selected as the best model instead of Model 3.

Improvement: Confirmation of that the numerical meaning is lost when Year is factorized

After changing the years into character types, factoring them into the model results in the same results as model 2 above. Therefore, as a result of factorizing year variable, year is no longer a continuous nuemeric variable, but a categorical variable with 6 categories.

```
temp <- df_train
temp$Year <- ifelse(temp$Year == 2017, "2017",
                    ifelse(temp$Year == 2018, "2018",
                           ifelse(temp$Year == 2019, "2019",
                                  ifelse(temp$Year == 2020,"2020",
                                  ifelse(temp$Year == 2021,"2021","2022")))))

temp_model <- lm(log(Price) ~ Year+Gwon+Building.Area+Building.Purpose+base.rate, data = temp)
summary(temp_model)
```

```
##
## Call:
## lm(formula = log(Price) ~ Year + Gwon + Building.Area + Building.Purpose +
##     base.rate, data = temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.7243  -0.2787   0.0205   0.2988   2.7640
##
## Coefficients:
##                                   Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                      2.034e+01  2.767e-01   73.525   <2e-16 ***
## Year2018                        -7.512e-02  2.766e-01   -0.272    0.786
## Year2019                         4.033e-02  2.766e-01    0.146    0.884
## Year2020                         2.694e-02  2.766e-01    0.097    0.922
## Year2021                         1.093e-01  2.766e-01    0.395    0.693
## Year2022                         1.979e-01  2.766e-01    0.715    0.474
## Gwonnortheast area              -4.604e-01  3.429e-03 -134.269   <2e-16 ***
## Gwonnorthwest area              -2.915e-01  3.586e-03  -81.291   <2e-16 ***
## Gwonsoutheast area               1.744e-01  3.724e-03   46.837   <2e-16 ***
## Gwonsouthwest area              -3.937e-01  3.414e-03 -115.331   <2e-16 ***
## Building.Area                    4.795e-03  1.356e-05  353.601   <2e-16 ***
## Building.Purposerow house       -8.593e-01  1.694e-03 -507.172   <2e-16 ***
## Building.PurposeSingle-family home -2.907e-01  3.313e-03  -87.756   <2e-16 ***
## Building.Purposestudio apartment   -1.033e+00  2.621e-03 -393.995   <2e-16 ***
## base.rate                       -1.021e-01  3.095e-03  -33.006   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4791 on 436149 degrees of freedom
## Multiple R-squared:  0.6414, Adjusted R-squared:  0.6414
## F-statistic: 5.571e+04 on 14 and 436149 DF,  p-value: < 2.2e-16
```

Calculatation of Accuracy and Error Rates

Now, let’s check how well the model is trained by comparing the actual and predicted values of the test set.

```
# Training data
pred1 <- predict(model2, df_train)
actual_pred_tr <- data.frame(cbind(actual= log(df_train$Price), predicted = pred1))

train_correlation_accuracy <- cor(actual_pred_tr)
train_correlation_accuracy
```

```
##              actual predicted
## actual    1.0000000 0.8008564
## predicted 0.8008564 1.0000000
```

The predicted value of the model appears to have a correlation of about 0.80 with the actual value of the train data.

```
# Test data accuracy
pred2 <- predict(model2, df_test %>% select(-Price)) # test on test set

actual_pred_te <- data.frame(cbind(actual=log(df_test$Price), predicted = pred2))
test_corr_acc <- cor(actual_pred_te)
test_corr_acc
```

```
##              actual predicted
## actual    1.0000000 0.8023015
## predicted 0.8023015 1.0000000
```

The predicted value of the model appears to have a correlation of about 0.8 with the actual value of the test data.

```
# Approximate distribution of test data and predicted values
summary(exp(pred2));summary(df_test$Price)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 1.297e+08 2.380e+08 4.572e+08 5.482e+09 6.339e+08 4.146e+14
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 1.700e+07 2.380e+08 3.980e+08 5.998e+08 7.440e+08 1.109e+11
```

When returning the value predicted by log, you can see that it is similar to the price value of the actual test data.

Evaluation indicators of the model confirmed through the forecast library

```
library(forecast)
accuracy(model2)
```

```
##                         ME      RMSE      MAE         MPE     MAPE      MASE
## Training set -9.164632e-18 0.4790756 0.362307 -0.05856501 1.826932 0.5548401
```

```
#RMSE
sqrt(sum((model2$residuals)^2)/nrow(df_train))
```

```
## [1] 0.4790756
```

The RMSE value was shown to be about 0.48.

Conclusions:

Through analysis, it was figured out that the price of real estates in Seoul is influenced by various variables, especially base rate and building area. Furthermore, by considering interaction term and multicollinearity by correlation coefficients, it was possible to do in-depth evaluation to select the best one among three models. Ultimately, the best model could predict the price of real estates with high similarity to actual values. In regards of limitations, there was a problem in prediction in which the model returned the predicted price as negative quantity. There were about 100 negative quantities among 450,000 values, and it was because there were not enough exogenous variables such as LTV(Loan-to-Value) or DTI(Debt-to-Income) except for base rate. Secondly, when the predicted value in log form is returned to its original form, there are cases where the Max value becomes much larger than the actual value. This is a phenomenon caused by not handling outliers well. Last point in the limitations is that precise analysis would have been possible if we proceeded with more specified house location data such as “Dong” because the price of real estate tends to be strongly influenced by specifically where it is located. Above all these significances and limitations, it was a precious experience to try to predict the price of real estate by data analysis, trying to overcome the fluctuation of real estate market.