Prediction of Singapore HDB Resale Flat Prices

HarvardX PH125.9x Data Science: Capstone - Choose Your Own Project

Loy Jong Sheng

01/05/2021

Contents

1 Introduction

Housing Development Board (HDB), Singapore's public housing authority and a statutory board under the Ministry of National Development, was set up in 1960 to look into the housing shortage and its related problems, such as overcrowding and squatter colonies, due to fast-growing population. In 1964, Singapore government introduced the Home Ownership for the People Scheme to give Singapore a tangible asset in the country and a stake in its nation-building. Over the years, this push for home ownership has improved Singapore's overall economic, social, and political stability. By 2018, about 81% of Singapore resident

population were staying in HDB flats. And there are more than 1 million flats built across 24 towns and 3 estates in Singapore (data.gov.sg, 2020). New HDB flats are sold directly by HDB at subsidized prices. A new HDB flat has a 99 years lease. Once the lease expired, the flat is returned to the government. Only Singaporeans who fulfilled HDB's eligibility criteria are allowed to buy new HDB flats and resale flats. For most Singaporeans, owning a HDB flat happens as soon as they get married.

In the recent years, the demand for HDB resale flat has been on the rising trend. These are mainly driven by several factors, namely: 1) Singapore Permanent Residents (PRs) can only buy resale flats. 2) Most new HDB flats are built in new towns, which will take many years to build up the amenities such as shopping malls, sport hall, MRT etc. 3) The Cash Over Valuation (COV) factor, which is the barrier to resale flat, was regulated by the government in 2014. 4) Newly married Singaporeans might not want to wait up to three years for their new HDB flats to be ready. 5) Older HDB resale flats in matured towns are generally bigger in size compared to the new HDB flats.

Being a Singaporean, whether to own a resale HDB flat is an important decision to make because it has a huge financial commitment. By understanding the factors driving the HDB resale flat prices and able to predict its prices can help to answer the following typical question: if you are going to buy your HDB resale flat, what it the right price for it? Of course being able to predict the prices also can help to answer what is the right price to sell if you going to sell your HDB flat.

2 Aim

This Capstone project aims to explore and determine a Machine Learning (ML) model (with least Residual Mean Squared Error (RMSE)) that is capable of predicting Singapore HDB resale flat prices for buyers who want to make decisions on buying HDB resale flats based on the past transaction trends.

3 Approach

This project shall perform the following key steps: 1) Gather Singapore HDB Resale Flat Prices information, 2) Prepare datasets for analysis, 2) Discover and visualize datasets to gain insights, 3) Explore and evaluate Regression and Decision Trees approaches 5) Select a Machine Learning (ML) model that achieve the lowest RMSE value, 6) Present the final solution.

4 Gathering of Singapore HDB Resale Flat Information

The Resale Flat Prices dataset is downloaded from the following link https://data.gov.sg/dataset/resale-flat-prices. Additional information such as whether the transacted resale flat is near popular school(s) or MRT station(s) are added. This data are derived based on a manual survey using the HDB Map Services (https://services2.hdb.gov.sg/web/fi10/emap.html#), and the geolocations of top popular primary schools and MRT stations.

5 Description of Singapore HDB Resale Flat Dataset

This project shall use the Resale Flat Prices dataset to train and evaluate the Machine Learning (ML) models. This dataset contained 117,527 HDB resales flat transaction records from January 2015 to September 2020. It has 13 variables namely: 1) month - the transaction month and year. 2) town - name of the town where the HDB flat located. 3) flat_type - Types of HDB flat. 4) block - Block number of the HDB flat. 5) street_name - street name of the HDB flat. 6) storey_range - level range of HDB flat unit. 7) floor_area_sqm - area of the HDB flat unit. 8) flat_model - category of the flat unit. 9) lease_commence_year - year which HDB

flat commenced. 10) remaining_lease_year - remaining HDB flat lease year. 11) resale_price - price of the HDB flat unit transaction. 12) pop_pri_school_nearby - Is popular school nearby 13) mrt_nearby - Is MRT Station(s) nearby

Additional factors whether a Mass Rapid Transit (MRT) station is nearby or a popular primary school is nearby are added to the dataset. These factors affect the resale prices of the HDB flats.

6 Data Preparation

The HDB Resale Flat Prices dataset is splitted into two datasets namely HDBResaletrain, which is used to train the ML algorithm, and HDBResalesvalidation, which is used to evaluate the algorithm.

```
#Download the data file from github
urlfile <-
  "https://raw.githubusercontent.com/jonaloy729/HDBdata/main
/resale-flat-prices%20from-jan-2015-onwards.csv"

HDBResales<-read_csv(url(urlfile))
```

6.1 Create 70% training dataset and 30% validation dataset from Singapore HDB Resale Flat Prices dataset.

According to Bradley, B & Brandon, G (2020), it is good to consider the following points when splitting the dataset for training and validation of the ML models: - Spending too much in training, for example >80%, won't allow us to get a good assessment of predictive performance. We may find a model that fits the training data very well, but is not generalizable (overfitting). - Sometimes too much spent in testing, for example >40%, won't allow us to get a good assessment of model parameters.

Since there are 117,527 records in the Resale Flat Prices dataset, a split of 70% of the datasets for training, which is 82,267 records, and 30% of the datasets for validation, which 35,260 records, is recommended in this project.

```
set.seed(123, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(123)`

#Splitting the dataset randomly for training and validating the models.
test_index <- createDataPartition(y = HDBResales$resale_price, times = 1, p = 0.3, list = FALSE)

#70% of the dataset use for training the models.
HDBResalestrain <- HDBResales[-test_index,]
#30% of the dataset use for validating the models.
HDBResalesvalidation <- HDBResales[test_index,]
```

7 Discover and visualize the HDBResalestrain datasets to gain insights

In this section, the goal is to analyze and visualize the training dataset to gain insights.

```
head(HDBResalestrain)
```

```
## # A tibble: 6 x 13
##   month town  flat_type block street_name storey_range floor_area_sqm flat_model
##   <chr> <chr> <chr>     <chr> <chr>       <chr>                 <dbl> <chr>
## 1 2015~ ANG ~ 3 ROOM    174   ANG MO KIO~ 07 TO 09                 60 Improved
## 2 2015~ ANG ~ 3 ROOM    163   ANG MO KIO~ 01 TO 03                 69 New Gener~
## 3 2015~ ANG ~ 3 ROOM    446   ANG MO KIO~ 01 TO 03                 68 New Gener~
## 4 2015~ ANG ~ 3 ROOM    557   ANG MO KIO~ 07 TO 09                 68 New Gener~
## 5 2015~ ANG ~ 3 ROOM    603   ANG MO KIO~ 07 TO 09                 67 New Gener~
## 6 2015~ ANG ~ 3 ROOM    504   ANG MO KIO~ 04 TO 06                 82 New Gener~
## # ... with 5 more variables: lease_commence_year <dbl>,
## #   remaining_lease_year <dbl>, resale_price <dbl>,
## #   pop_pri_school_nearby <dbl>, mrt_nearby <dbl>
```

```
HDBResalestrain %>% summarize(Num_Distinct_Town = n_distinct(town),
                              Num_Distinct_Flat_Type = n_distinct(flat_type),
                              Num_Distinct_Storey_Range = n_distinct(storey_range),
                              Num_Distinct_Flat_Model = n_distinct(flat_model),
                              Tot_size = nrow(HDBResalestrain))
```

```
## # A tibble: 1 x 5
##   Num_Distinct_Town Num_Distinct_Fla~ Num_Distinct_St~ Num_Distinct_Fl~ Tot_size
##               <int>             <int>            <int>            <int>    <int>
## 1                26                 7               17               20    82267
```

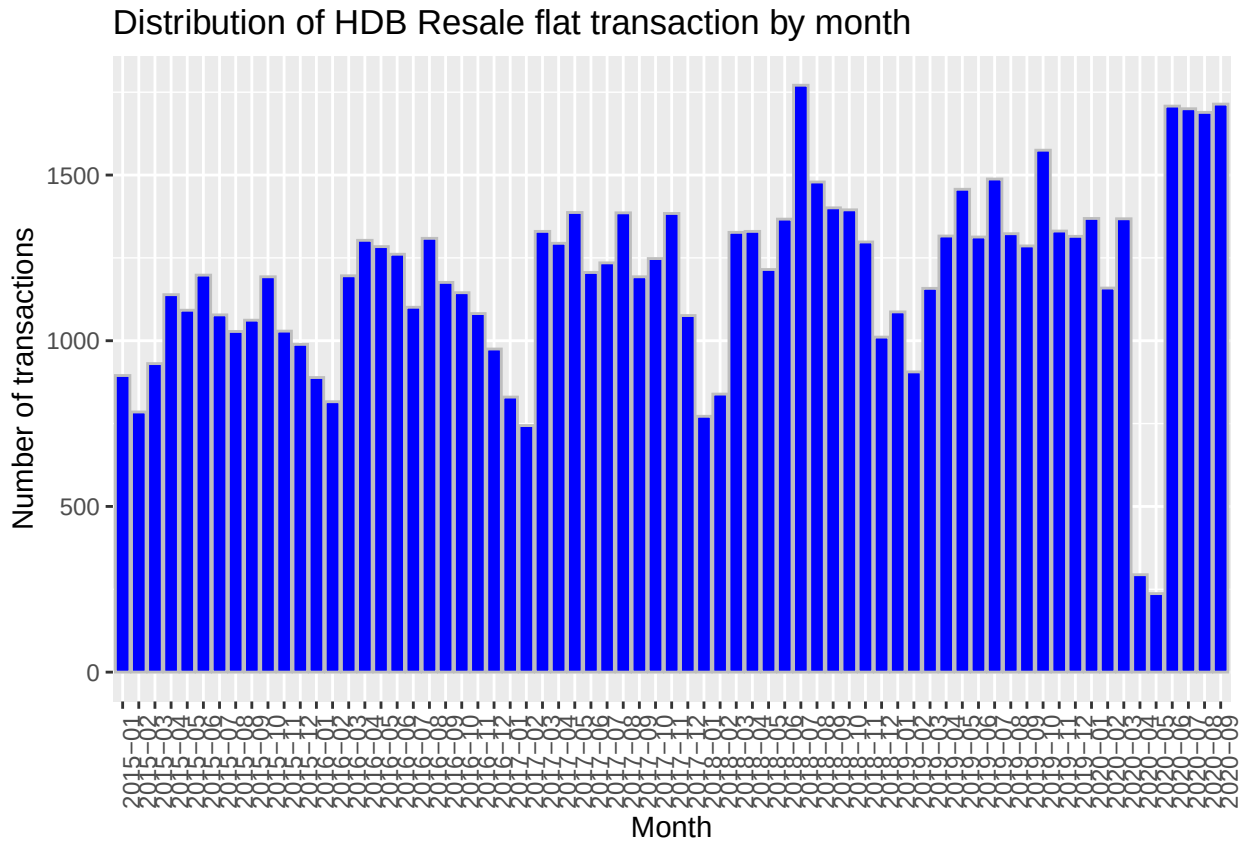
```
summary(HDBResalestrain)
```

```
##     month               town            flat_type            block
##  Length:82267       Length:82267       Length:82267       Length:82267
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  street_name        storey_range       floor_area_sqm    flat_model
##  Length:82267       Length:82267       Min.   : 31.00   Length:82267
##  Class :character   Class :character   1st Qu.: 77.00   Class :character
##  Mode  :character   Mode  :character   Median : 95.00   Mode  :character
##                                        Mean   : 97.45
##                                        3rd Qu.:112.00
##                                        Max.   :249.00
##  lease_commence_year remaining_lease_year  resale_price
##  Min.   :1966        Min.   :45.00        Min.   : 140000
##  1st Qu.:1984        1st Qu.:65.00        1st Qu.: 333000
##  Median :1992        Median :73.00        Median : 408888
##  Mean   :1993        Mean   :74.12        Mean   : 438736
##  3rd Qu.:2002        3rd Qu.:83.00        3rd Qu.: 508000
##  Max.   :2019        Max.   :97.00        Max.   :1258000
##  pop_pri_school_nearby   mrt_nearby
##  Min.   :0.0000        Min.   :0.000
##  1st Qu.:0.0000        1st Qu.:0.000
##  Median :0.0000        Median :0.000
##  Mean   :0.1076        Mean   :0.398
##  3rd Qu.:0.0000        3rd Qu.:1.000
##  Max.   :1.0000        Max.   :1.000
```

7.1 Exploring HDB Resale Flat Transactions

7.1.1 Exploring the HDB Resale Flat monthly transactions from Jan 2015 and Sept 2020

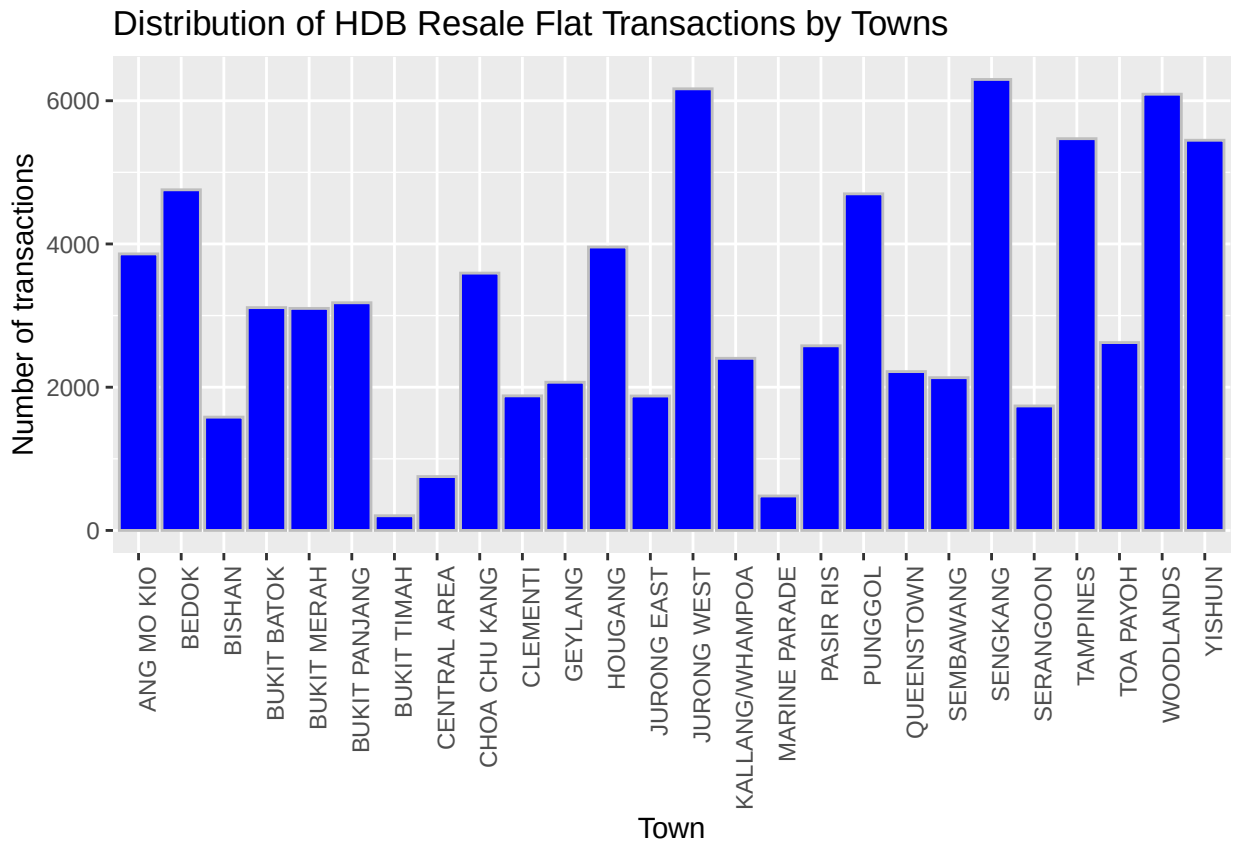
```
HDBresaledemandbymth <- HDBResalestrain %>% group_by(month) %>%
  summarize(Num_transactions = n())
#Plot Number of HDB Resale flat transaction by month
HDBresaledemandbymth %>% ggplot(mapping = aes(x = month, y = Num_transactions)) +
  geom_col(fill = "blue", color = "grey") +
  labs(y = "Number of transactions", x = "Month") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of HDB Resale flat transaction by month")
```



From the HDB Resale flat monthly transactions from January 2015 to September 2020, it is observed that there is a seasonal pattern. Around November of the year, the number of transaction started to decline and continue toward Lunar New Year. During this period, the festive and school holiday period begun. Both HDB flat sellers and buyers and even property agents started to enjoy their holidays. Once they are back from their holidays, they will start to prepare for school reopening and subsequently preparing for the Lunar New Year. Around February 2020, there is a global outbreak of COVID-19 pandemic which resulted in Singapore government declaring a circuit breaker (CB) period from April to May 2020. During this period, the number of transactions sharply declined because movement restriction and housing viewings have been barred. But it sharply recovered from June to September 2020 with the ending of circuit breaker in May 2020.

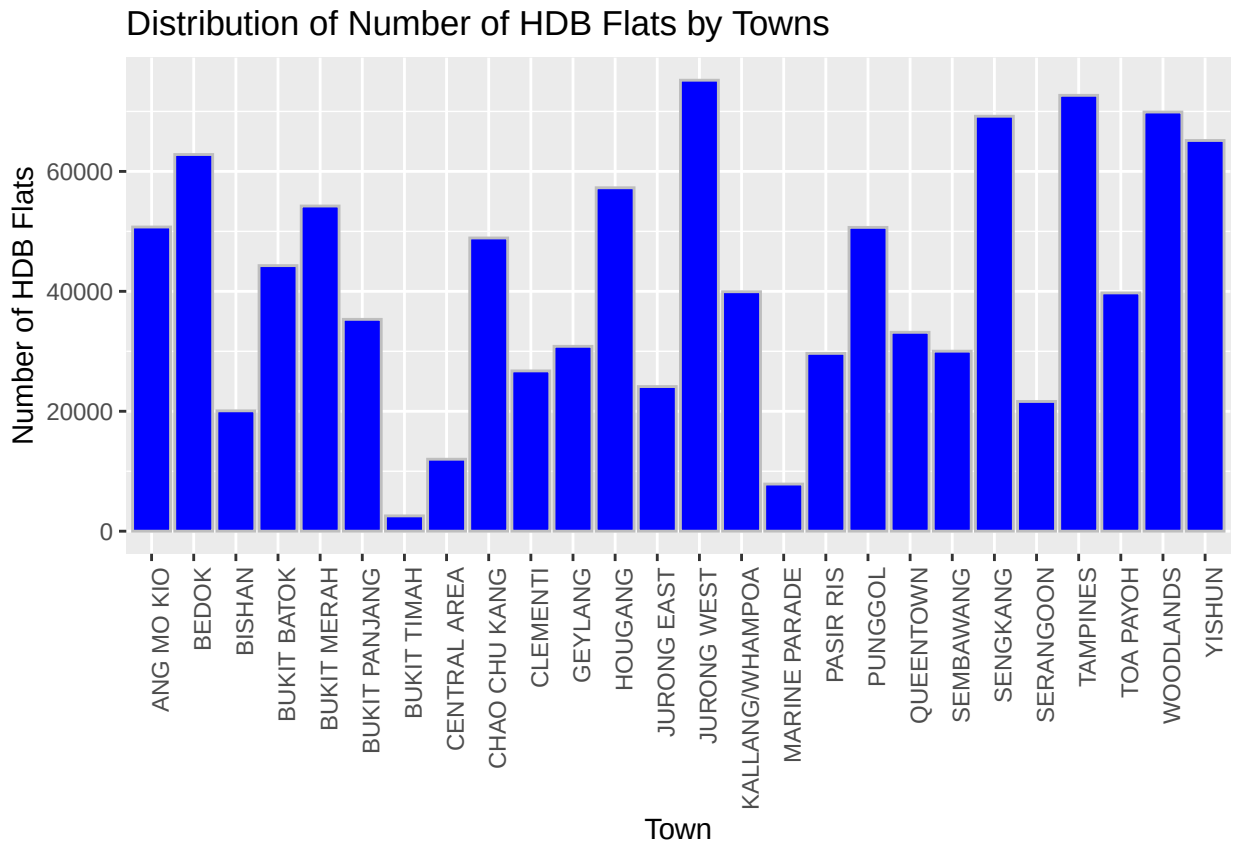
7.1.2 Exploring HDB Resale Flat transactions by Towns

```
NumResalesBytown <- HDBResalestrain %>% group_by(town) %>%
  summarize(Num_transaction = n())
#Plot Number of HDB Resale Flat Transactions by Towns
NumResalesBytown %>% ggplot(mapping = aes(x = town, y = Num_transaction)) +
  geom_col(fill = "blue", color = "grey") +
  labs(y = "Number of transactions", x = "Town") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of HDB Resale Flat Transactions by Towns")
```




```
HDBTowns <- c("ANG MO KIO", "BEDOK", "BISHAN",
              "BUKIT BATOK", "BUKIT MERAH", "BUKIT PANJANG",
              "BUKIT TIMAH", "CENTRAL AREA", "CHAO CHU KANG",
              "CLEMENTI", "GEYLANG", "HOUGANG",
              "JURONG EAST", "JURONG WEST", "KALLANG/WHAMPOA",
              "MARINE PARADE", "PASIR RIS", "PUNGGOL",
              "QUEENTOWN", "SEMBAWANG", "SENGKANG",
              "SERANGOON", "TAMPINES", "TOA PAYOH",
              "WOODLANDS", "YISHUN")
#HDB data as of 31 March 2020
NumHDBFlatsInTowns <- c(50726, 62816, 20072,
                        44285, 54227, 35325,
                        2554, 12003, 48900,
```

```
                        26730, 30829, 57272,
                        24122, 75208, 39931,
                        7860, 29654, 50663,
                        33164, 30020, 69196,
                        21632, 72683, 39737,
                        69900, 65158)
#Plot the Number of HDB Flats by Towns
ggplot(mapping = aes(x = HDBTowns, y = NumHDBFlatsInTowns)) +
  geom_col(fill = "blue", color = "grey") +
  labs(y = "Number of HDB Flats", x = "Town") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Number of HDB Flats by Towns")
```

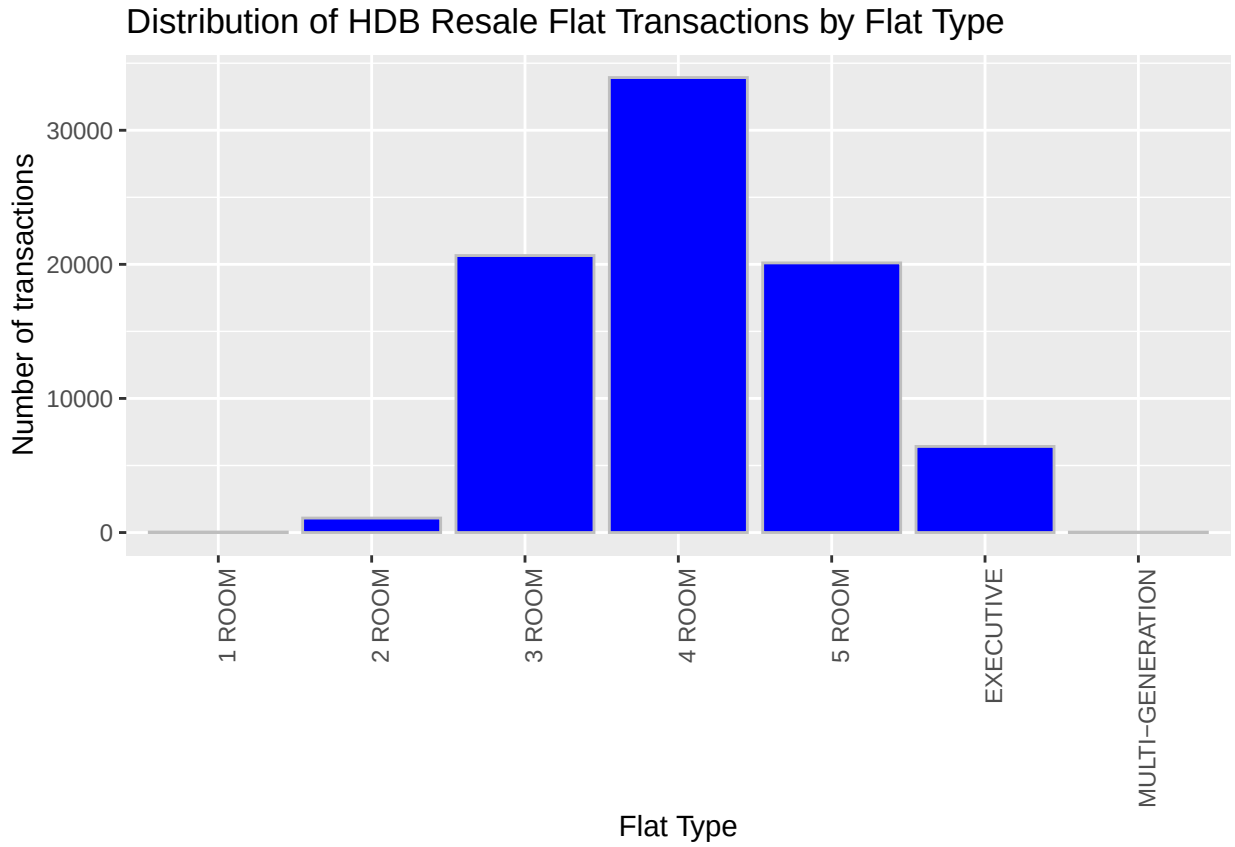


From the HDB resale flat transactions by towns, the top three towns that have the most HDB resale flats transactions are Sengkang, Jurong West and Woodlands. Sengkang is a relatively new modern town with lots of public amenities such as community hospital, sport complex. It has 69,196 flats. Jurong West is undergoing a complete makeover and is slated to be Singapore's next business hub. It has 75,208 flats. Woodlands also known for as a gateway to Johor Bahru. Being in north, its resale flat price remains affordable compared to those towns near the central Business District (CBD). It has 69,900 flats.

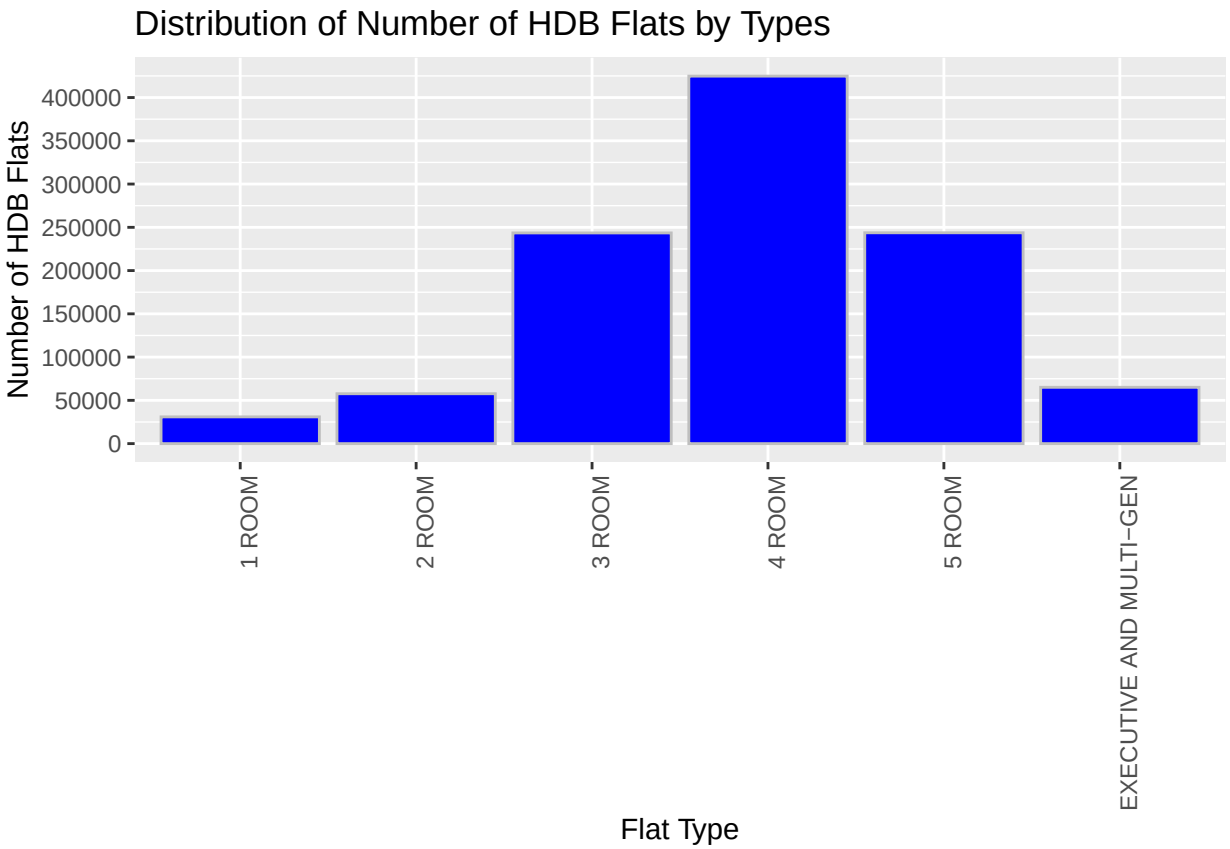
It is also observed that Bukit Timah, Central Area and Marine Parade are among the lowest. This is mainly due Bukit Timah, Central Area and Marine Parade are those towns which have the lowest number of HDB flats. They have 2,554 HDB flats, 12,003 HDB flats and 7,860 HDB flats respectively. Majority of their residents stayed private properties, i.e. landed, condominium, in these towns. The HDB resale flat price in these towns are also relatively more expensive.

7.1.3 Exploring HDB Resale Flat transactions by Flat Types

```
NumResalesbyflattype <- HDBResalestrain %>% group_by(flat_type) %>%
  summarize(Num_transaction = n())
#Plot Resale Flat transactions by Flat Types
NumResalesbyflattype %>%
  ggplot(mapping = aes(x = flat_type, y = Num_transaction)) +
  geom_col(fill = "blue", color = "grey") +
  labs(y = "Number of transactions", x = "Flat Type") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of HDB Resale Flat Transactions by Flat Type")
```



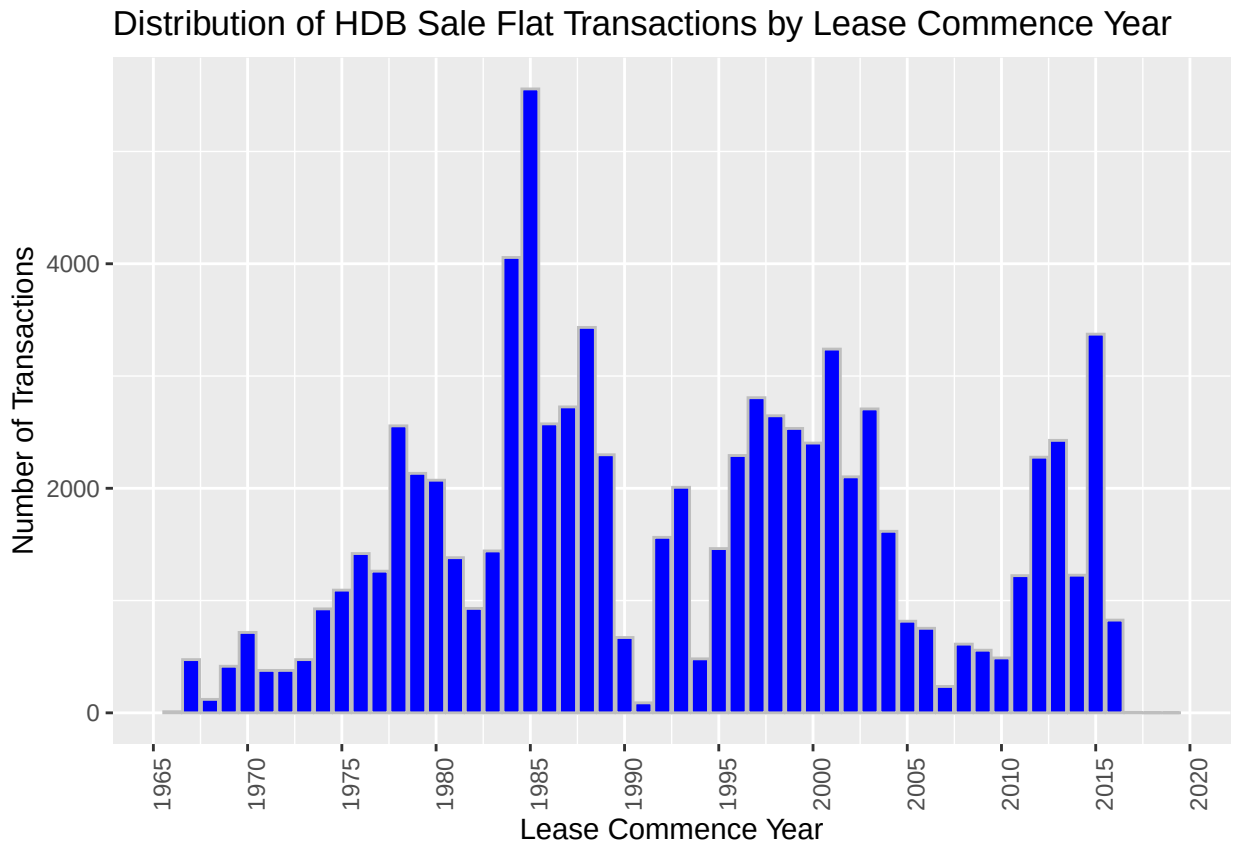

```
HDBFlatType <- c("1 ROOM", "2 ROOM", "3 ROOM", "4 ROOM", "5 ROOM", "EXECUTIVE AND MULTI-GEN")
#HDB data as of 31 March 2020
NumHDBFlatTypes <- c(30906, 57660, 243519, 424769, 243707, 65107)
#Plot Number of HDB Flats by Types
ggplot(mapping = aes(x = HDBFlatType, y = NumHDBFlatTypes)) +
  geom_col(fill = "blue", color = "grey") +
  scale_y_continuous(breaks= seq(0, 450000, by=50000)) +
  labs(y = "Number of HDB Flats", x = "Flat Type") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Number of HDB Flats by Types")
```



From the distribution above, it is observed that 4-room is the most transacted HDB flat type followed by 3-room and 5-room. According to HDB data (as of March 2020), the three most flat type built are 4-room, followed by 5-room and 3-room. There are 424,769 4-room flats, 243,707 5-room flats and 243,519 3-room flats being built.

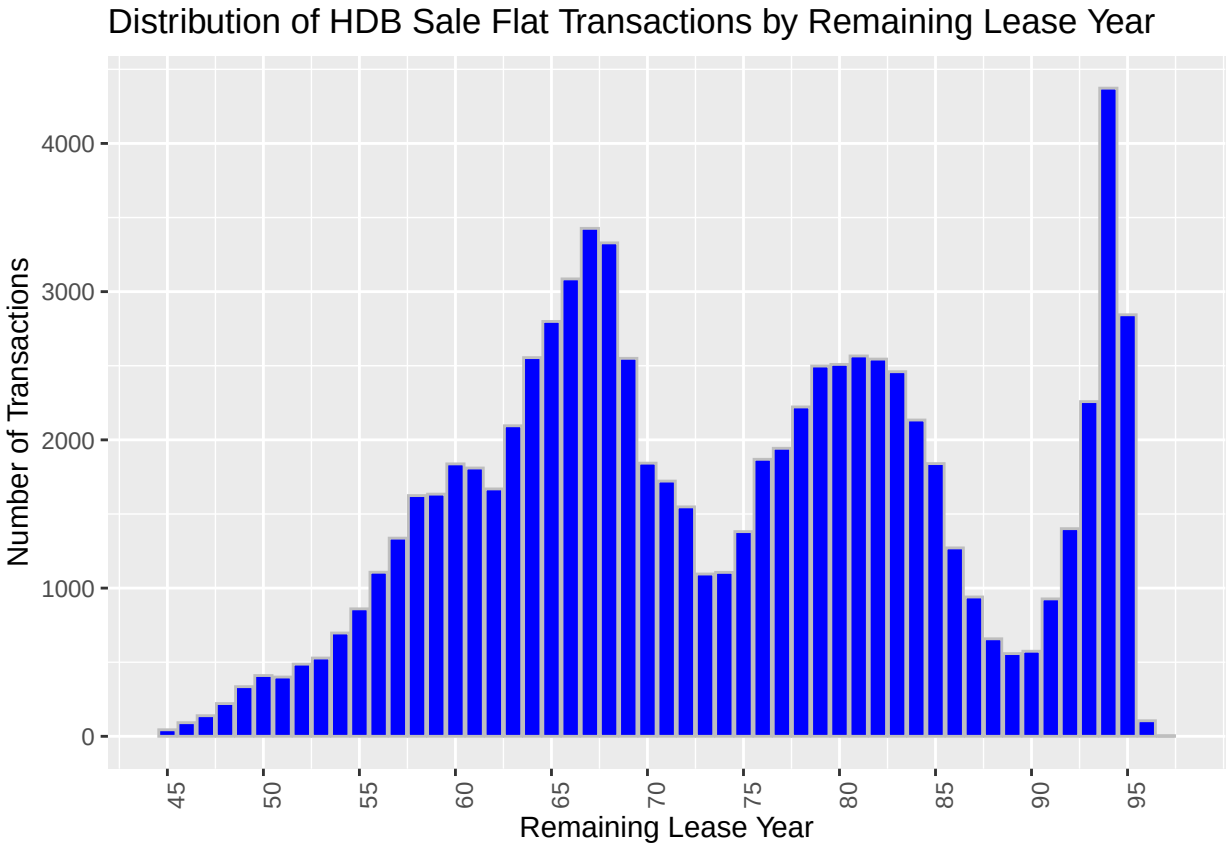
7.1.4 Exploring HDB Resale Flat transactions by Lease Commence Years

```
NumResalesbyleasecommence <- HDBResalestrain %>% group_by(lease_commence_year) %>%
  summarize(Num_transaction = n())
#Plot Number of HDB Sale Flat Transactions by Lease Commence Year
NumResalesbyleasecommence %>%
  ggplot(mapping = aes(x = lease_commence_year, y = Num_transaction)) +
  scale_x_continuous(breaks= seq(1960, 2020, by=5)) +
  geom_col(fill = "blue", color = "grey") +
  labs(y = "Number of Transactions", x = "Lease Commence Year") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of HDB Sale Flat Transactions by Lease Commence Year")
```



7.1.5 Exploring HDB Resale Flat transactions by Remaining Lease Years

```
NumResalesbyremainlease <- HDBResalestrain %>% group_by(remaining_lease_year) %>%
  summarize(Num_transaction = n())
#Plot Number of HDB Sale Flat Transactions by Remaining Lease Year
NumResalesbyremainlease %>%
  ggplot(mapping = aes(x = remaining_lease_year, y = Num_transaction)) +
  scale_x_continuous(breaks= seq(40, 99, by=5)) +
  geom_col(fill = "blue", color = "grey") +
  labs(y = "Number of Transactions", x = "Remaining Lease Year") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of HDB Sale Flat Transactions by Remaining Lease Year")
```



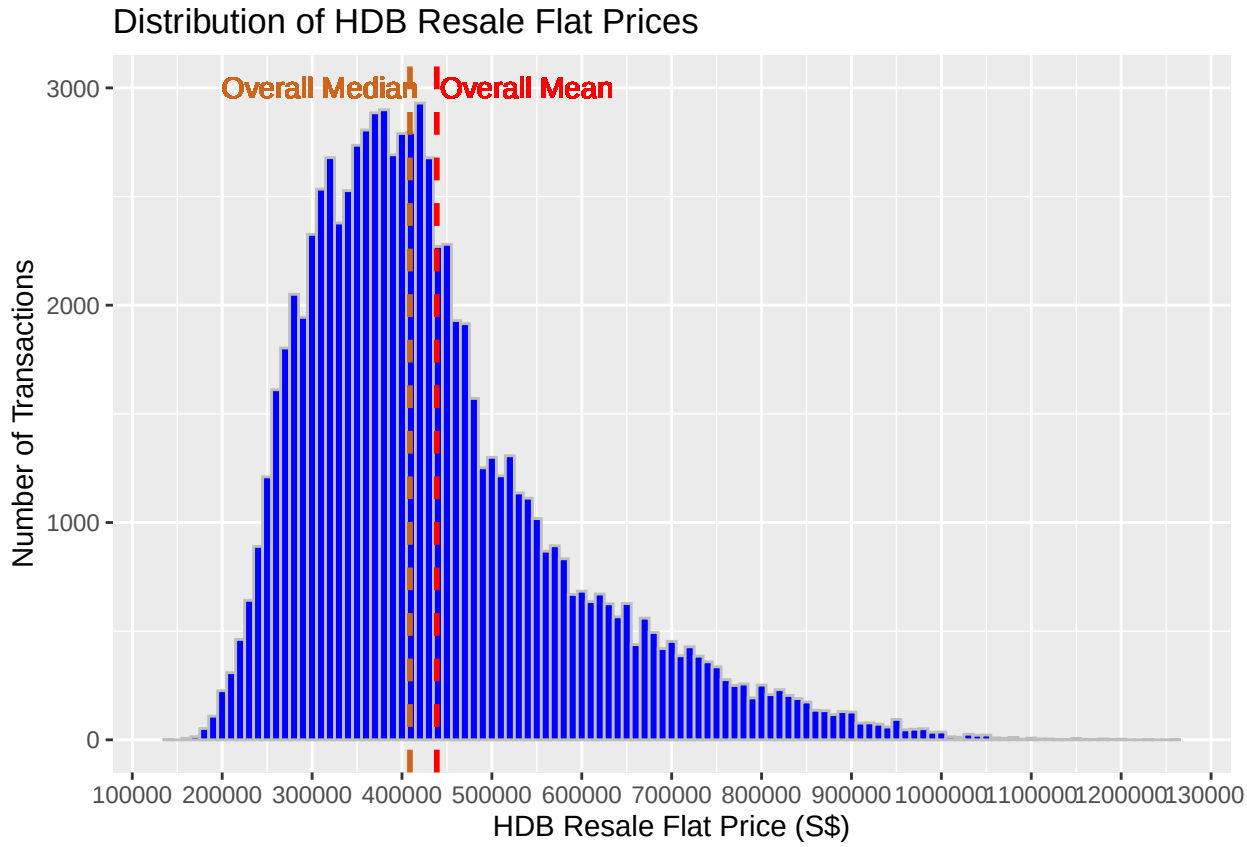
All HDB flats have a 99 year leases. For those who bought the new flats from HDB, they are not allow to sell their flats within the minimum occupancy period of 5 years stipulated by HDB. From the HDB Resale flat transactions from January 2015 to September 2020, it is observed that those newer flats that have just exceeded the minimum occupancy period have a high number transactions. Some owners of these newer HDB considered monetising their flats. Some considered upgrading to private property or bigger flats due to larger family. The newer flats, with abundance of greenery, also attracted buyers who do not to wait for a few years for new flats to be built.

There are also high transactions for older HDB flats. Older HDB flats enjoy the benefits of upgrading efforts such as the Singapore government's Home Improvement Programme that help reduce the aging effects more effectively. In August 2018, Singapore government introduced the Voluntary Early Redevelopment Scheme, which gives flat owners aged 70 years and older a chance to sell their homes to the government. This gave buyer confidence in older flats. Older flats are also attractive to larger family because those built in the 1980s and 1990s are bigger than the ones built in 2010s. For example, the size of a 1980s built 5-Room flat is 135 sqm compared to the size of a 2010s built 5-room flat which is 105 sqm. Furthermore older flats are located in matured towns, whereby amenities and transportation network are well established.The proximity housing grant of $20,000 also encouraged HDB buyer to buy resale flat within same town or within four kilometers from their parent's flat.

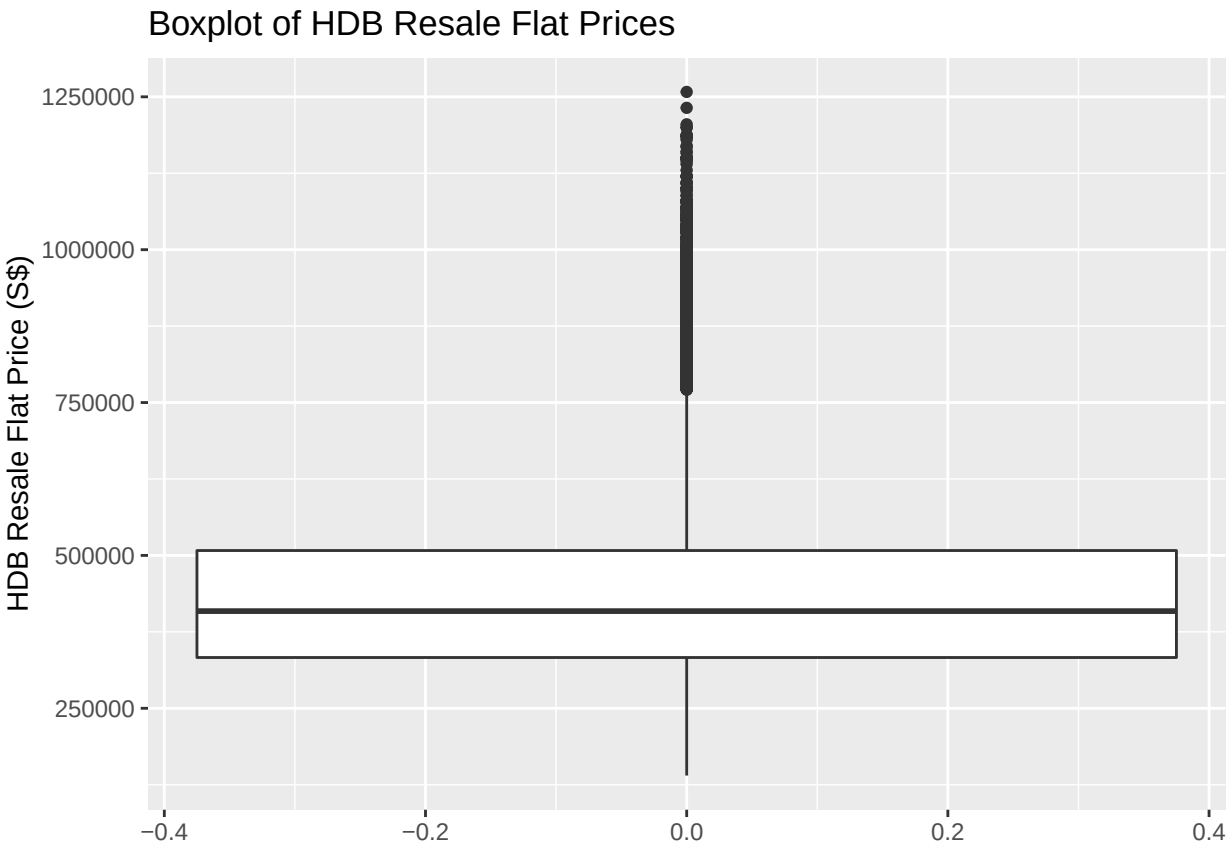
7.2 Exploring HDB Resale Flat Prices

```
#Plot HDB Resale Flat Prices transacted
ggplot(HDBResalestrain, aes(x=resale_price)) +
  geom_histogram(binwidth = 10000, fill = "blue", color = "grey") +
  scale_x_continuous(breaks= seq(0, 1300000, by=100000)) +
```

```
  geom_vline(aes(xintercept=mean(resale_price)),color="red",
             linetype="dashed", size=1, alpha = 1) +
  geom_text(aes(x = mean(resale_price)+100000, y = 3000,
                label = "Overall Mean"), color="red") +
  geom_vline(aes(xintercept=median(resale_price)),color="chocolate3",
             linetype="dashed", size=1, alpha = 1) +
  geom_text(aes(x = median(resale_price)-100000, y = 3000,
                label = "Overall Median"), color="chocolate3") +
  labs(y = "Number of Transactions", x = "HDB Resale Flat Price (S$)") +
  ggtitle("Distribution of HDB Resale Flat Prices")
```



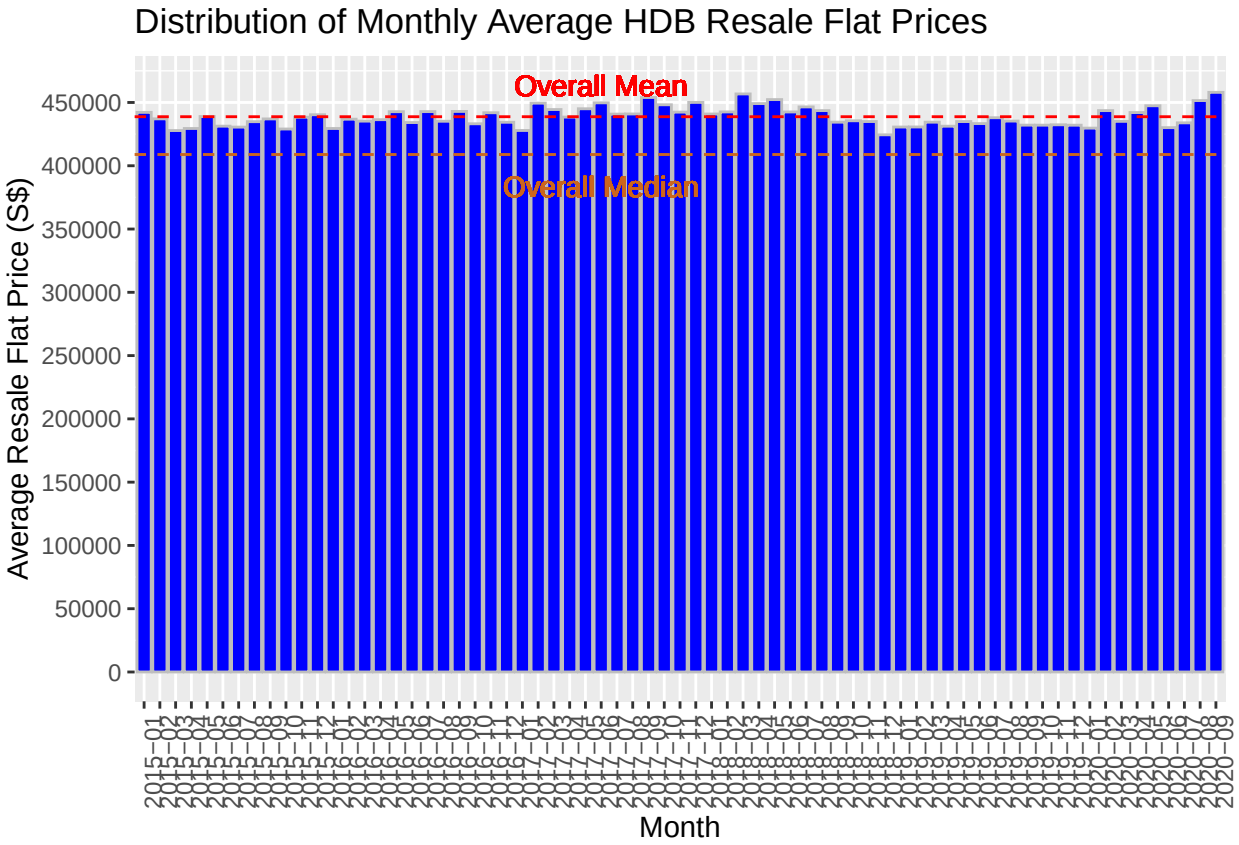

```
ggplot(HDBResalestrain, aes(y=resale_price)) +
  geom_boxplot() +
  labs(y = "HDB Resale Flat Price (S$)") +
  ggtitle("Boxplot of HDB Resale Flat Prices")
```



From the distribution of HDB Resale Flat Prices, it is observed that it is righted skewed with a mean price of $438,736 and median price of $408,888. The resale flat prices ranged from $140,000 to $1,258,000. It is expected because not many HDB buyers are willing to pay high prices for HDB flats. These outliers can be observed on the boxplot chart.

7.2.1 Exploring the Average HDB Resale Flat Prices from Jan 2015 to Sept 2020

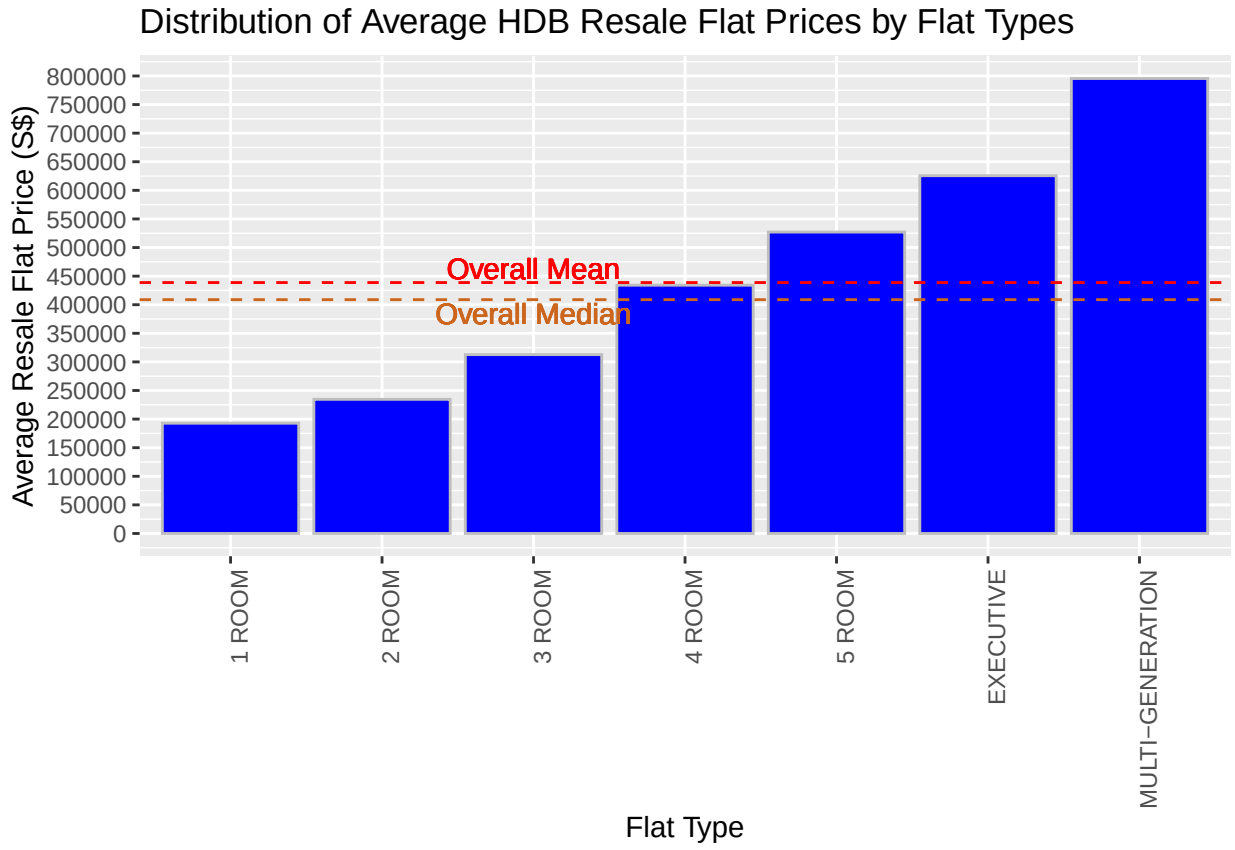
```
AvgHDBresalepricebymth <- HDBResalestrain %>% group_by(month) %>%
  summarize(Avg_Resales_Price = mean(resale_price))
#Plot Monthly Average HDB Resale Flat Prices
AvgHDBresalepricebymth %>%
  ggplot(mapping = aes(x = month, y = Avg_Resales_Price)) +
  geom_col(fill = "blue", color = "grey") +
  scale_y_continuous(breaks= seq(0, 500000, by=50000)) +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=30), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=30), colour="chocolate3", angle=0) +
  labs(y = "Average Resale Flat Price (S$)", x = "Month") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Monthly Average HDB Resale Flat Prices")
```



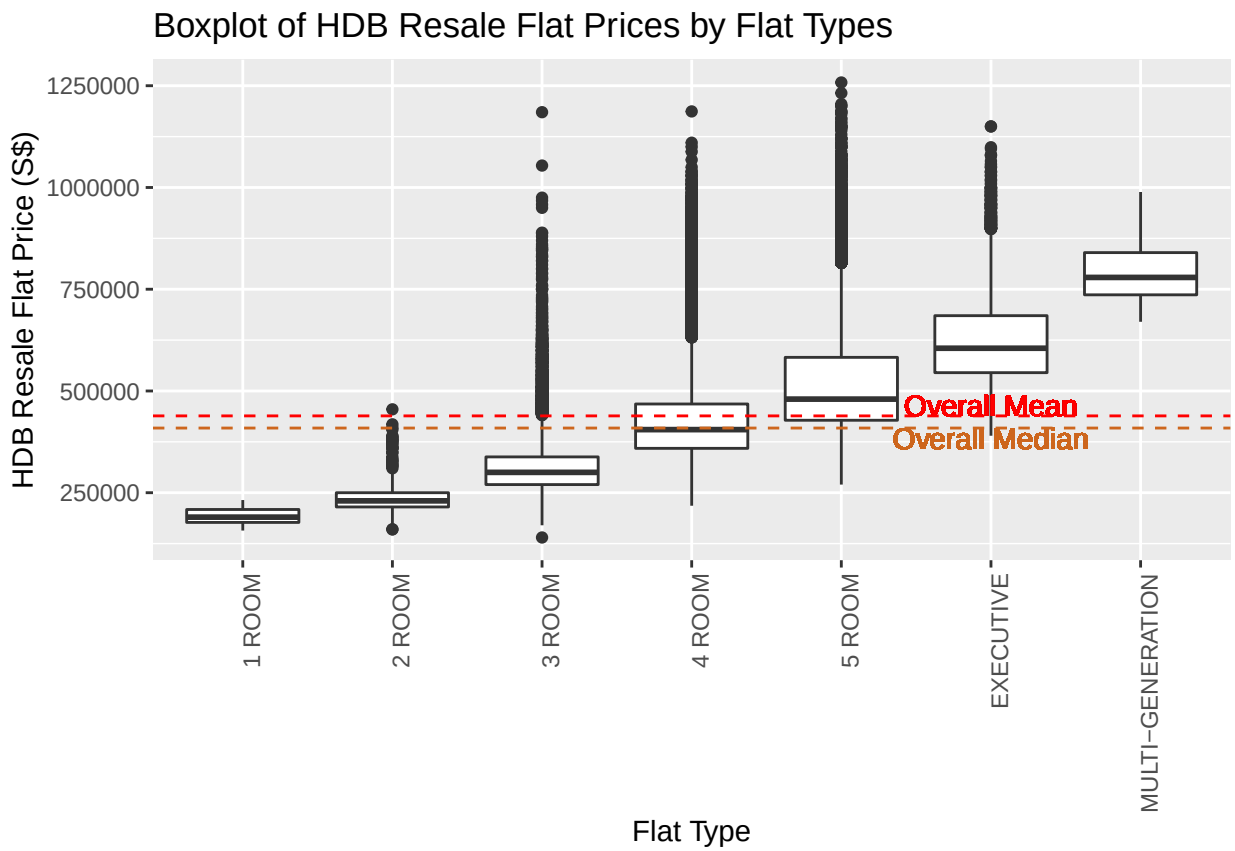
From the distribution above, it is observed that from January 2015 to September 2020, the monthly average resale flat prices did not fluctuated drastically.

7.2.2 Exploring the Average HDB Resale Flat Prices by Flat Types

```
AvgHDBresalepricebyflattype <- HDBResalestrain %>%
  group_by(flat_type) %>%
  summarize(Avg_Resales_Price = mean(resale_price))
#Plot Average HDB Resale Flat Prices by Flat Types
AvgHDBresalepricebyflattype %>%
  ggplot(mapping = aes(x = flat_type, y = Avg_Resales_Price)) +
  geom_col(fill = "blue", color = "grey") +
  scale_y_continuous(breaks= seq(0, 1000000, by=50000)) +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=3), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=3), colour="chocolate3", angle=0) +
  labs(y = "Average Resale Flat Price (S$)", x = "Flat Type") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Average HDB Resale Flat Prices by Flat Types")
```



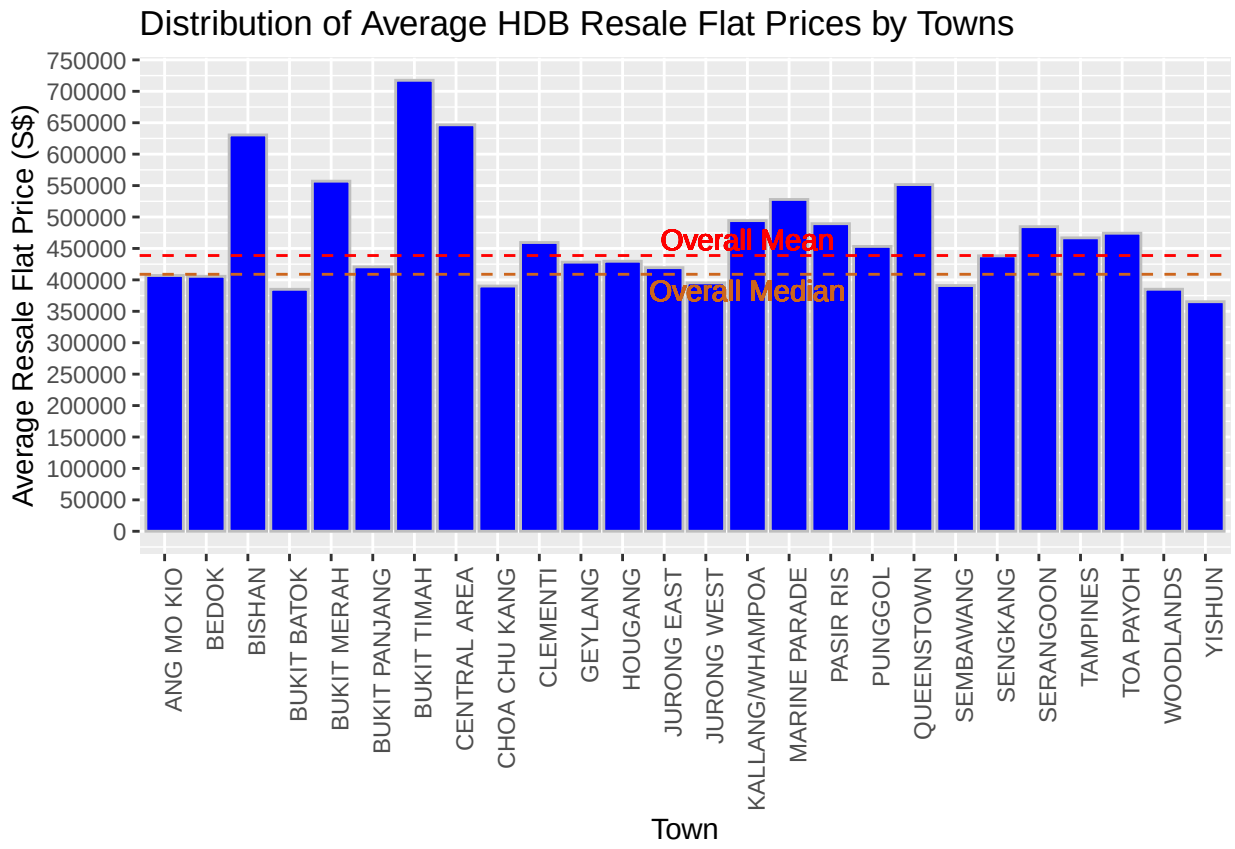

```
ggplot(HDBResalestrain, aes(x = flat_type, y=resale_price)) +
  geom_boxplot() + geom_hline(yintercept = mean(HDBResalestrain$resale_price),
                              linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=6), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=6), colour="chocolate3", angle=0) +
  labs(x = "Flat Type", y = "HDB Resale Flat Price (S$)") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Boxplot of HDB Resale Flat Prices by Flat Types")
```



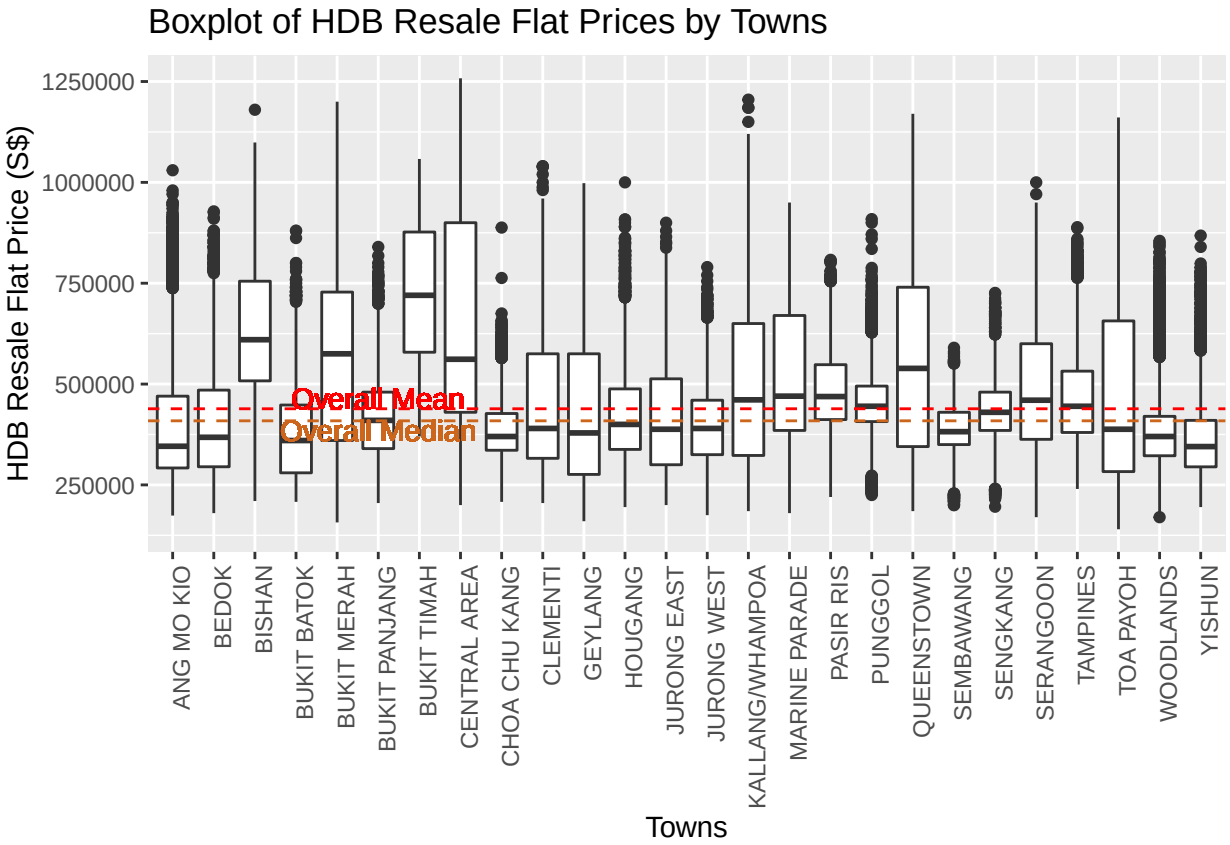
From the distribution above, the 5 room, executive and multi-generation flat types have a higher resale value. From the boxplot of HDB Resale Flat Prices by Flat Types, there are 2-room to executive flat types have outliers. These outliers highlighted buyers are willing to pay much more for HDB resale flats due to its good location, the convenience of being near to the city, the availability of amenities and good schools.

7.2.3 Exploring the Average HDB Resale Flat Prices by Towns

```
AvgHDBresalepricebytown <- HDBResalestrain %>% group_by(town) %>%
  summarize(Avg_Resales_Price = mean(resale_price))
#Plot Average HDB Resale Flat Prices by Towns
AvgHDBresalepricebytown %>%
  ggplot(mapping = aes(x = town, y = Avg_Resales_Price)) +
  geom_col(fill = "blue", color = "grey") +
  scale_y_continuous(breaks= seq(0, 1000000, by=50000)) +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=15), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=15), colour="chocolate3", angle=0) +
  labs(y = "Average Resale Flat Price (S$)", x = "Town") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Average HDB Resale Flat Prices by Towns")
```




```
ggplot(HDBResalestrain, aes(x = town, y=resale_price)) +
  geom_boxplot() +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=6), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=6), colour="chocolate3", angle=0) +
  labs(x = "Towns", y = "HDB Resale Flat Price (S$)") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Boxplot of HDB Resale Flat Prices by Towns")
```

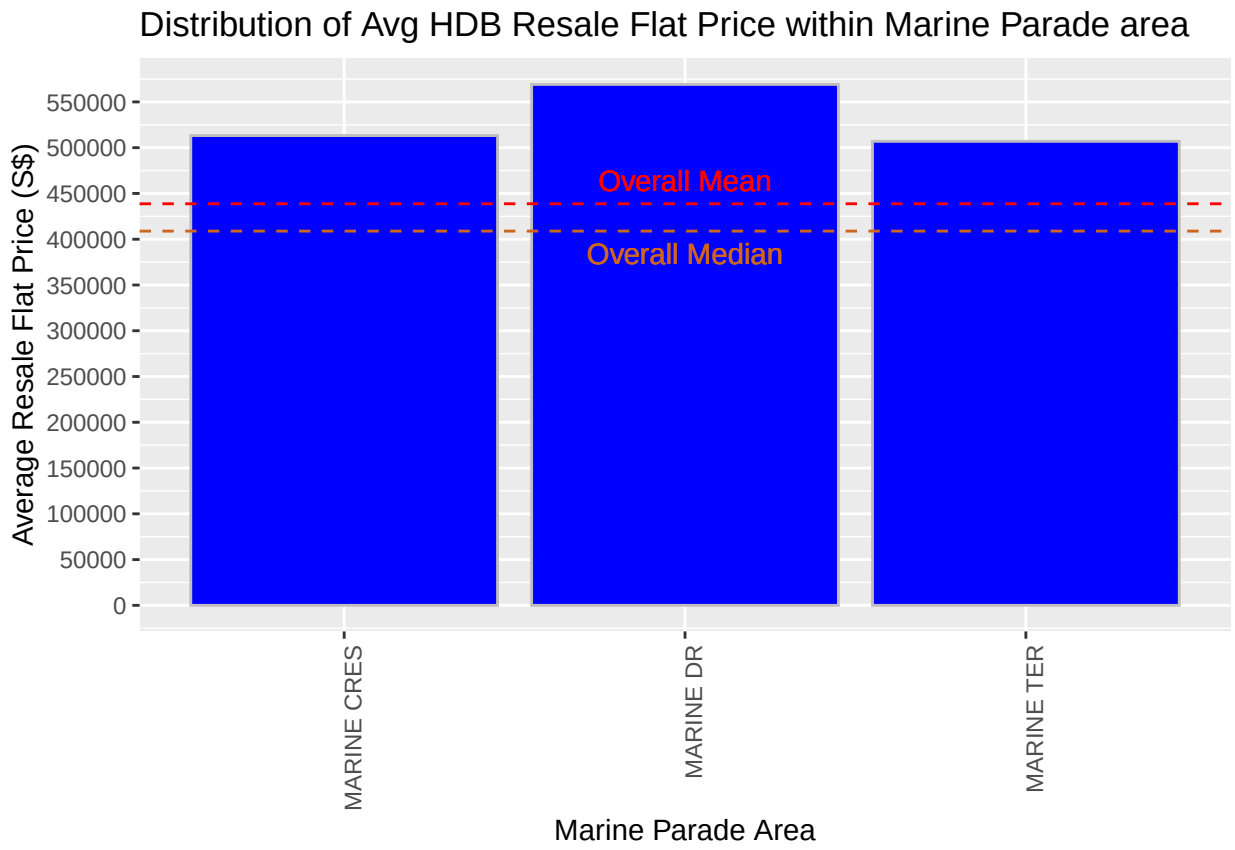


From the distribution of Average HDB Resale Flat Prices by Town, Bishan, Bukit Merah, Bukit Timah, Central Area, Marine Parade and Queenstown have a higher resale values. These are mature towns and the convenience of being near to the city. From the Boxplot of HDB Resale Flat Prices by Towns, it is observed Bukit Merah, Bukit Timah, Central Area, Geylang, Marine Parade, Queenstown, and Toa Payoh do not have outliers as compared to the rest of the town.

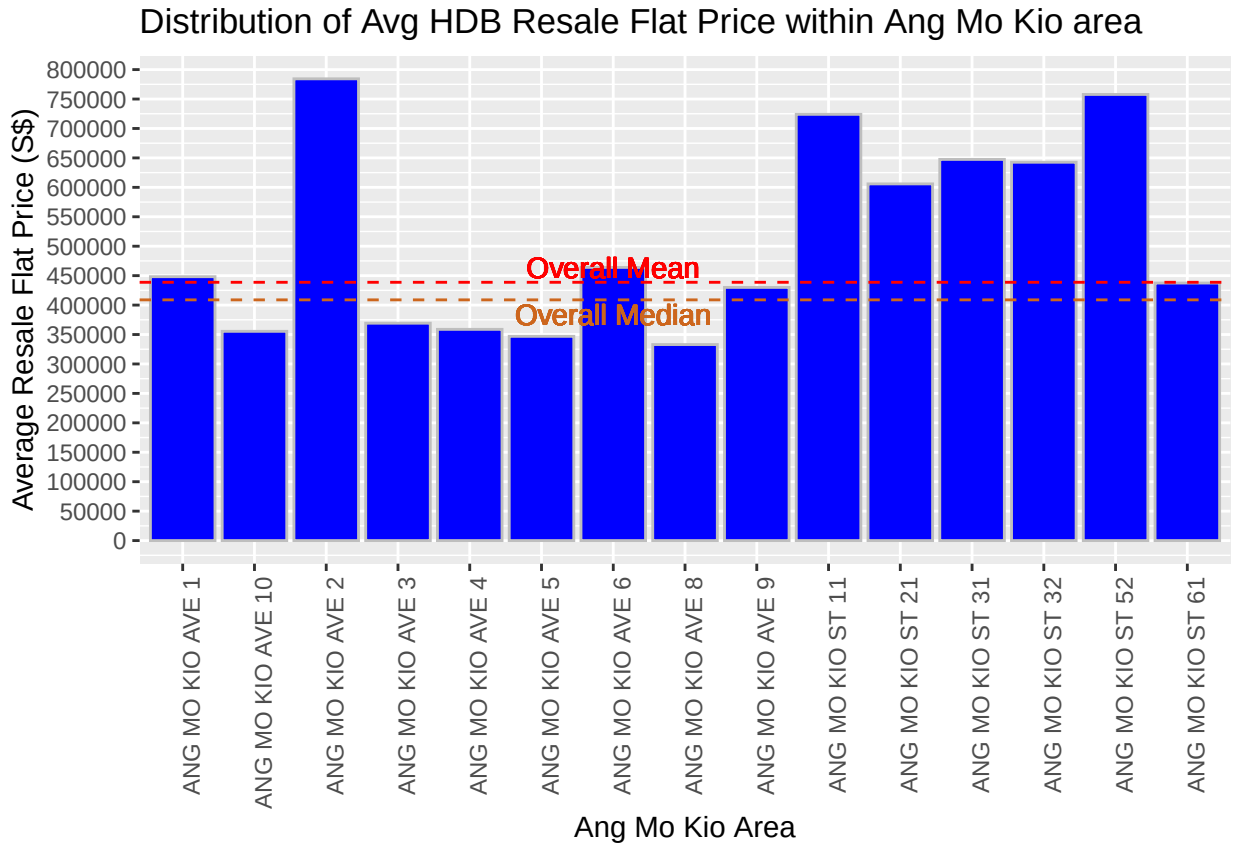
7.2.4 Exploring Avg HDB Resale Flat Prices in Towns

```
AvgHDBresalepricebyaddress <- HDBResalestrain %>% filter(town == "MARINE PARADE") %>%
  group_by(street_name) %>%
  summarize(Avg_Resales_Price = mean(resale_price))
AvgHDBresalepricebyaddress %>%
  ggplot(mapping = aes(x = street_name, y = Avg_Resales_Price)) +
  geom_col(fill = "blue", color = "grey") +
  scale_y_continuous(breaks= seq(0, 700000, by=50000)) +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=2), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=2), colour="chocolate3", angle=0) +
  labs(y = "Average Resale Flat Price (S$)", x = "Marine Parade Area") +
```

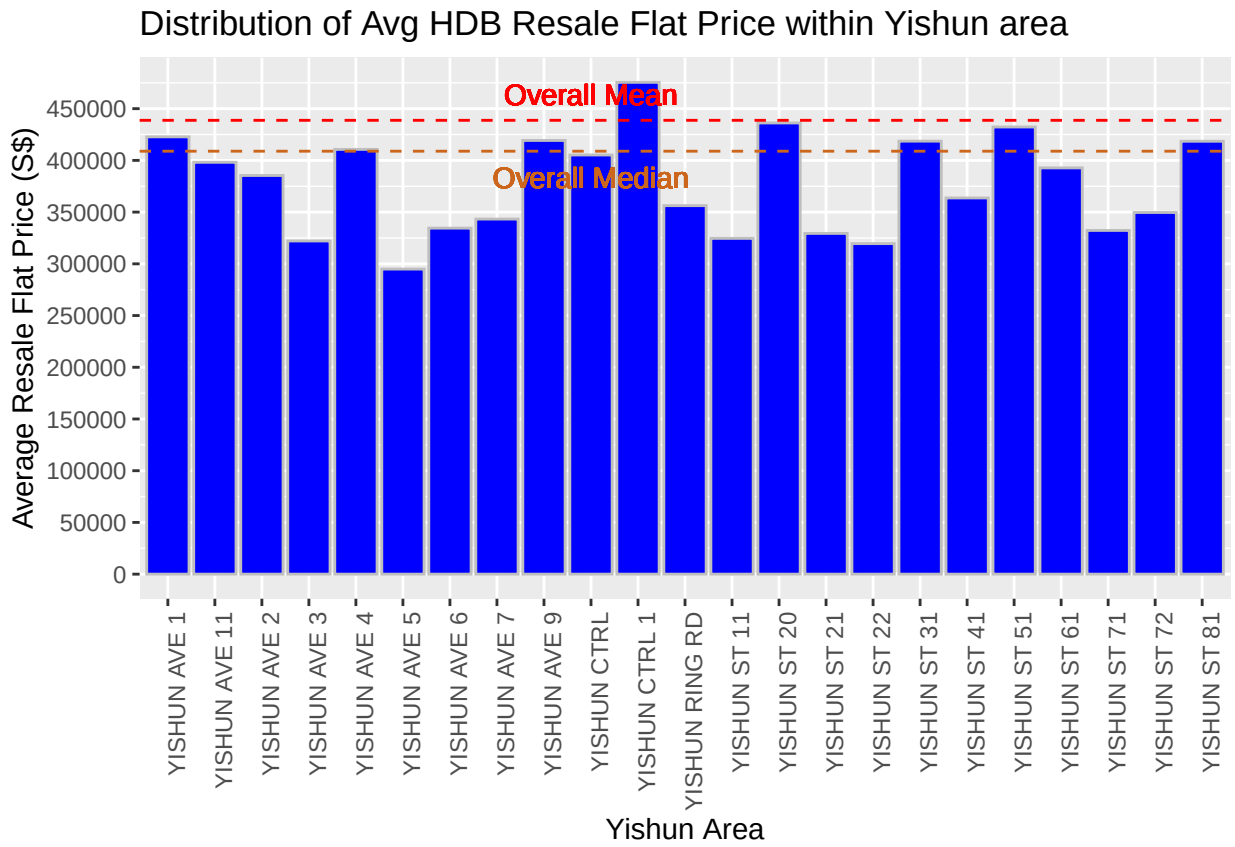
```
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Avg HDB Resale Flat Price within Marine Parade area")
```




```
AvgHDBresalepricebyaddress <- HDBResalestrain %>% filter(town == "ANG MO KIO") %>%
  group_by(street_name) %>% summarize(Avg_Resales_Price = mean(resale_price))

AvgHDBresalepricebyaddress %>%
  ggplot(mapping = aes(x = street_name, y = Avg_Resales_Price)) +
  geom_col(fill = "blue", color = "grey") +
  scale_y_continuous(breaks= seq(0, 900000, by=50000)) +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=7), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=7), colour="chocolate3", angle=0) +
  labs(y = "Average Resale Flat Price (S$)", x = "Ang Mo Kio Area") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Avg HDB Resale Flat Price within Ang Mo Kio area")
```



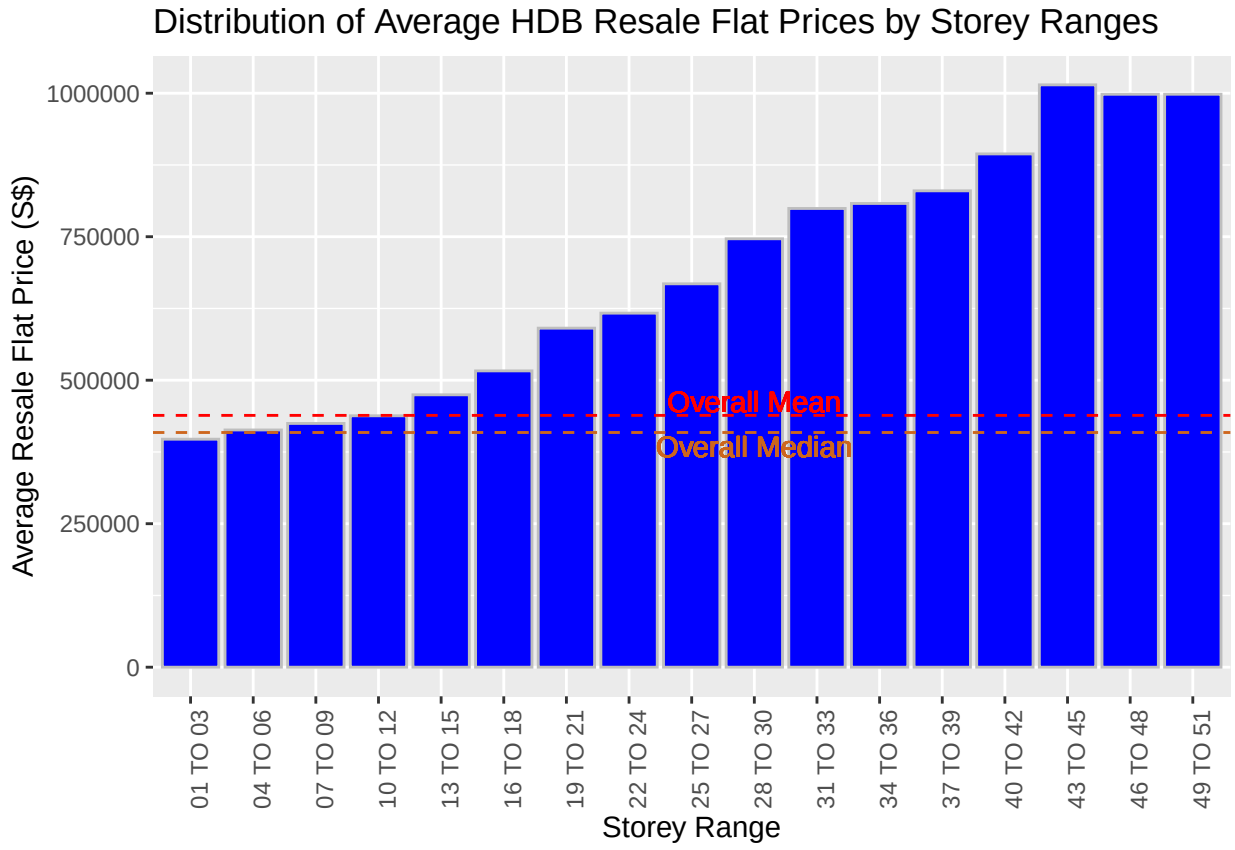

```
AvgHDBresalepricebyaddress <- HDBResalestrain %>%
  filter(town == "YISHUN") %>% group_by(street_name) %>%
  summarize(Avg_Resales_Price = mean(resale_price))
#Plot Average HDB Resale Flat Price within Yishun area
AvgHDBresalepricebyaddress %>%
  ggplot(mapping = aes(x = street_name, y = Avg_Resales_Price)) +
  geom_col(fill = "blue", color = "grey") +
  scale_y_continuous(breaks= seq(0, 600000, by=50000)) +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=10), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=10), colour="chocolate3", angle=0) +
  labs(y = "Average Resale Flat Price (S$)", x = "Yishun Area") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Avg HDB Resale Flat Price within Yishun area")
```



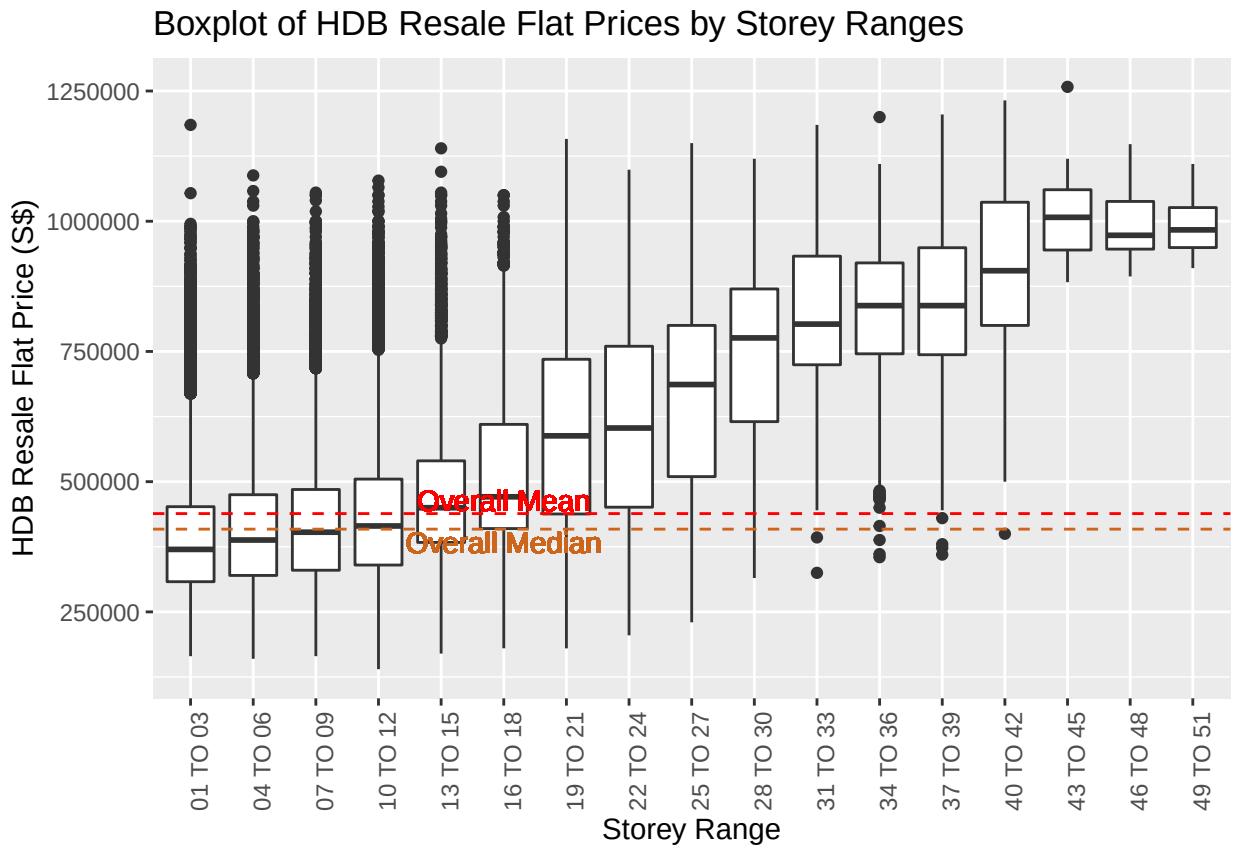
From the above distributions, it is observed that within a town, the resale flats proximity to amenities such as shopping malls, MRT stations, popular schools and public parks etc do influence its prices.

7.2.5 Exploring the Average HDB Resale Flat Prices by Storey Range

```
AvgHDBresalepricebystorey <- HDBResalestrain %>%
  group_by(storey_range) %>%
  summarize(Avg_Resales_Price = mean(resale_price))
#Plot Average HDB Resale Flat Prices by Storey Ranges
AvgHDBresalepricebystorey %>%
  ggplot(mapping = aes(x = storey_range, y = Avg_Resales_Price)) +
  geom_col(fill = "blue", color = "grey") +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=10), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=10), colour="chocolate3", angle=0) +
  labs(y = "Average Resale Flat Price (S$)", x = "Storey Range") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Average HDB Resale Flat Prices by Storey Ranges")
```




```
ggplot(HDBResalestrain, aes(x = storey_range, y=resale_price)) +
  geom_boxplot() +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=6), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=6), colour="chocolate3", angle=0) +
  labs(x = "Storey Range", y = "HDB Resale Flat Price (S$)") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Boxplot of HDB Resale Flat Prices by Storey Ranges")
```

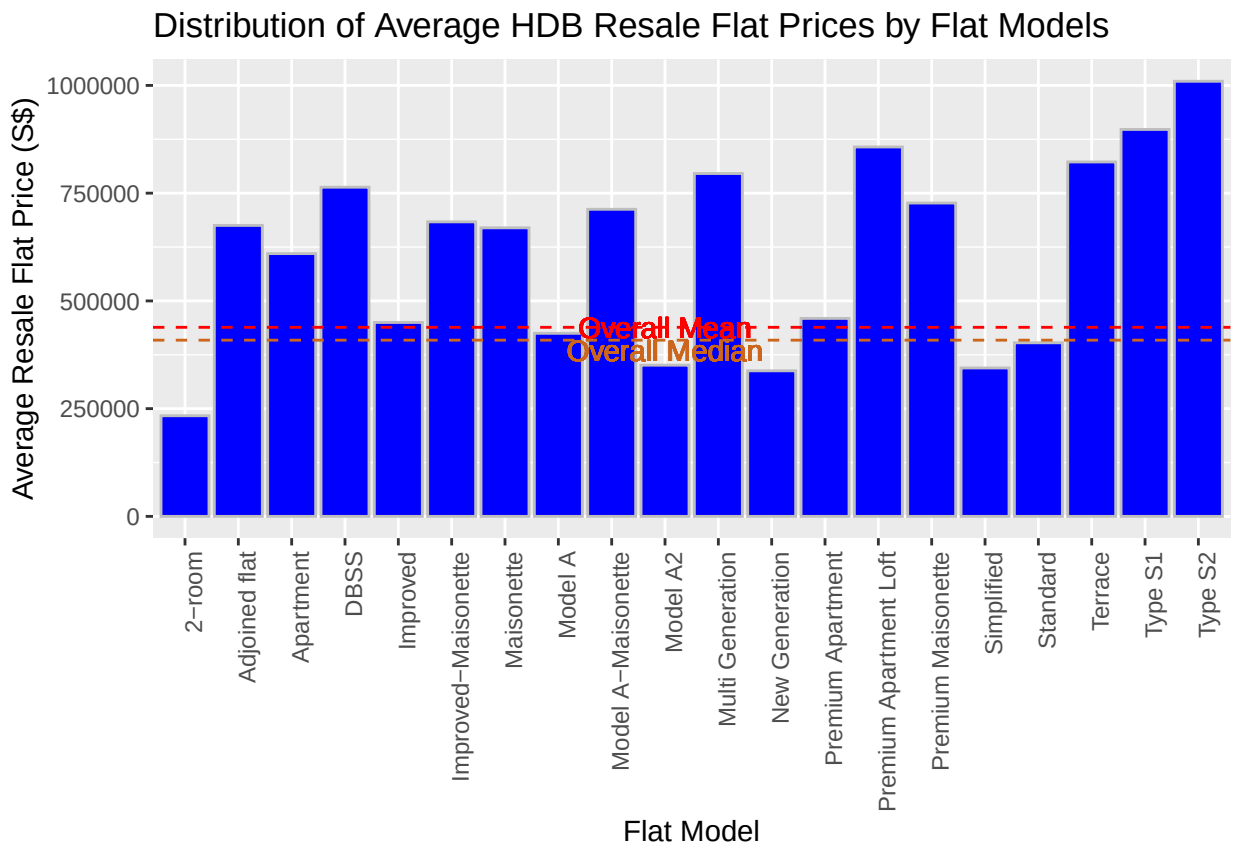


From the distribution Average HDB Resale Flat Prices by Storey Ranges, it is observed that the higher the storey, the higher resale price is the HDB flat. Majority of HDB flat buyers prefer higher storey due to lesser ambient noises, more privacy, unblocked views, and less dust from traffic and insects, etc. From the boxplot, it is also observed that there are significant of outliers in pricing for storey ranging from 1 to 18. This could be due to older HDB flats built in 1970s, 1980s and 1990s in matured towns.

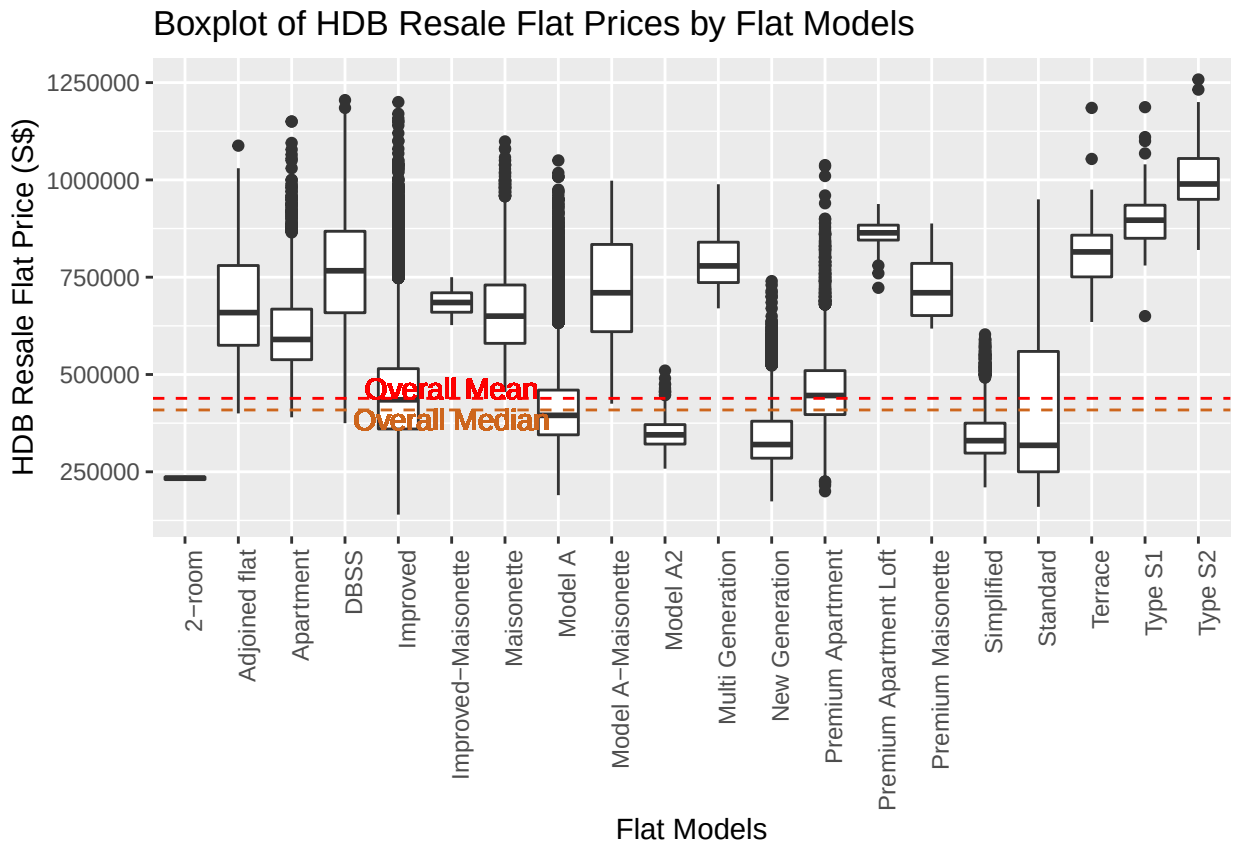
7.2.6 Exploring the Average HDB Resale Flat Prices by Flat Models

```
AvgHDBresalepricebyflatmodel <- HDBResalestrain %>%
  group_by(flat_model) %>%
  summarize(Avg_Resales_Price = mean(resale_price))
#Plot Average HDB Resale Flat Prices by Flat Models
AvgHDBresalepricebyflatmodel %>%
  ggplot(mapping = aes(x = flat_model, y = Avg_Resales_Price)) +
  geom_col(fill = "blue", color = "grey") +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+200,
                label="Overall Mean", x=10), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=10), colour="chocolate3", angle=0) +
  labs(y = "Average Resale Flat Price (S$)", x = "Flat Model") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
```

```
  ggtitle("Distribution of Average HDB Resale Flat Prices by Flat Models")
```




```
ggplot(HDBResalestrain, aes(x = flat_model, y=resale_price)) +
  geom_boxplot() +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=6), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=6), colour="chocolate3", angle=0) +
  labs(x = "Flat Models", y = "HDB Resale Flat Price (S$)") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Boxplot of HDB Resale Flat Prices by Flat Models")
```

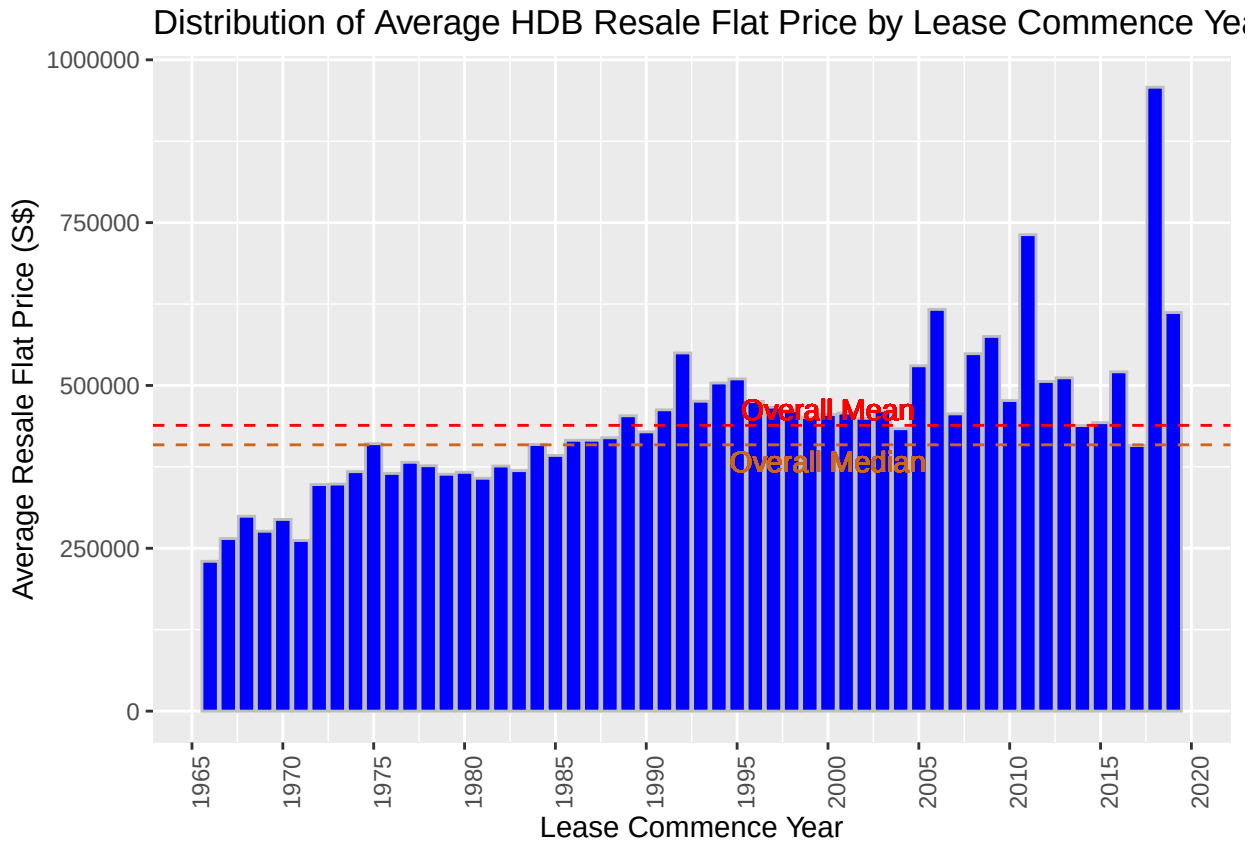


From the distribution of Average HDB Resale Flat Prices by Flat Models, it is observed that HDB has built 20 flat models to cater to different household sizes and needs. For example, Multi-Generation flats can only be bought by multi-generational families, comprising of a married or courting couple, and one set of parents. There are types of HDB models that are no longer built. They are Adjoined flat, Multi-Generation, Terrace, Premium Apartment with Loft, Maisonette, DBSS, and Apartment. Due to its limited supply, these models commanded a premium price in the HDB resale flat market. Type S1 and S2 models are built under a HDB unique project called PinnacleAtDuxton. It is the tallest public residential building located in Singapore's city center, next to CBD. These two special flat models also commanded a premium price in the market due to its uniqueness and the convenience of being in the city.

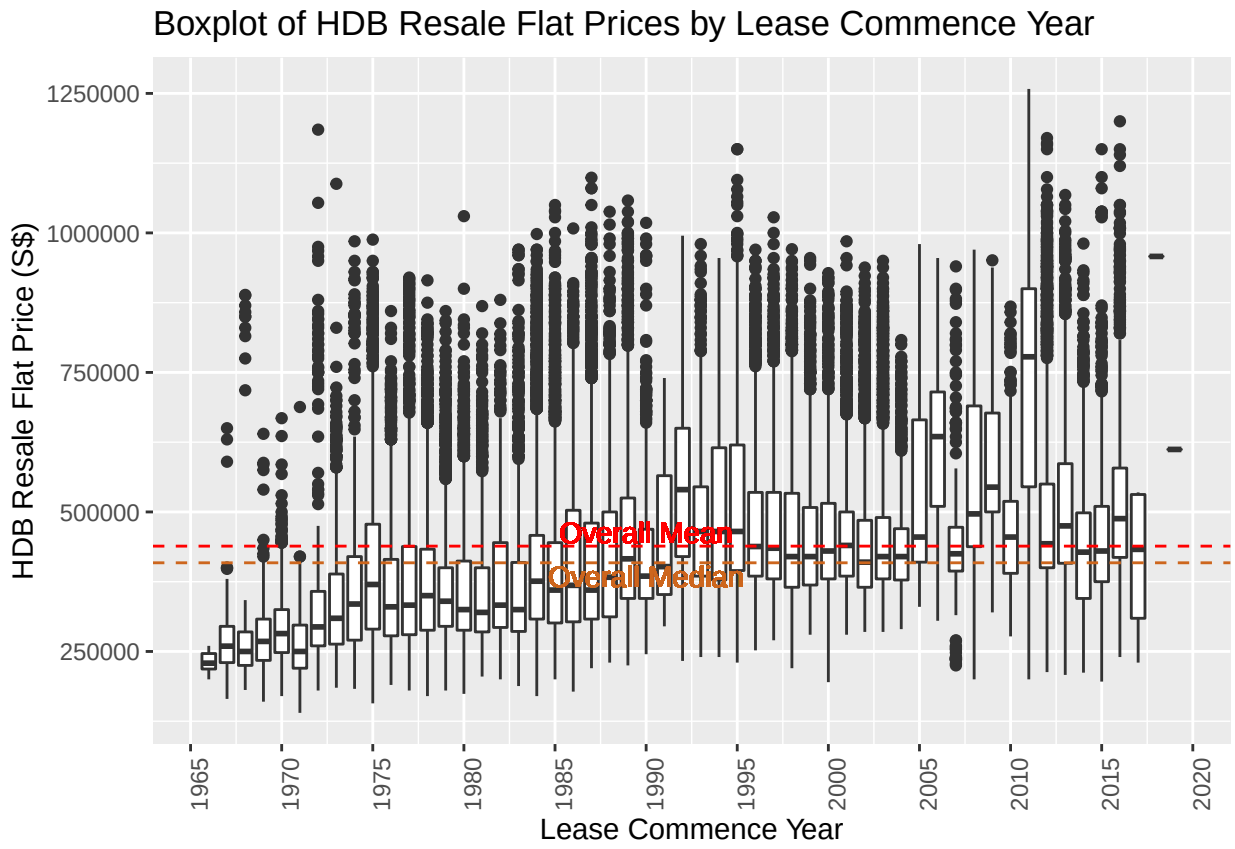
7.2.7 Exploring the Average HDB Resale Flat Prices by Lease Commence Year

```
AvgHDBresalepricebyleasecommence <- HDBResalestrain %>%
  group_by(lease_commence_year) %>%
  summarize(Avg_Resales_Price = mean(resale_price))
AvgHDBresalepricebyleasecommence %>%
  ggplot(mapping = aes(x = lease_commence_year, y = Avg_Resales_Price)) +
  geom_col(fill = "blue", color = "grey") +
  scale_x_continuous(breaks= seq(1965, 2020, by=5)) +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=2000), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
```

```
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=2000), colour="chocolate3", angle=0) +
  labs(y = "Average Resale Flat Price (S$)", x = "Lease Commence Year") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Average HDB Resale Flat Price by Lease Commence Year")
```




```
ggplot(HDBResalestrain, aes(x = lease_commence_year, y=resale_price, group=lease_commence_year)) +
  geom_boxplot() +
  scale_x_continuous(breaks= seq(1965, 2020, by=5)) +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=1990), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=1990), colour="chocolate3", angle=0) +
  labs(x = "Lease Commence Year", y = "HDB Resale Flat Price (S$)") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Boxplot of HDB Resale Flat Prices by Lease Commence Year")
```




```
TempRecord <- HDBResalestrain %>%
  filter(lease_commence_year == 2018 | lease_commence_year == 2019) %>%
  select(month, lease_commence_year, resale_price)
TempRecord
```

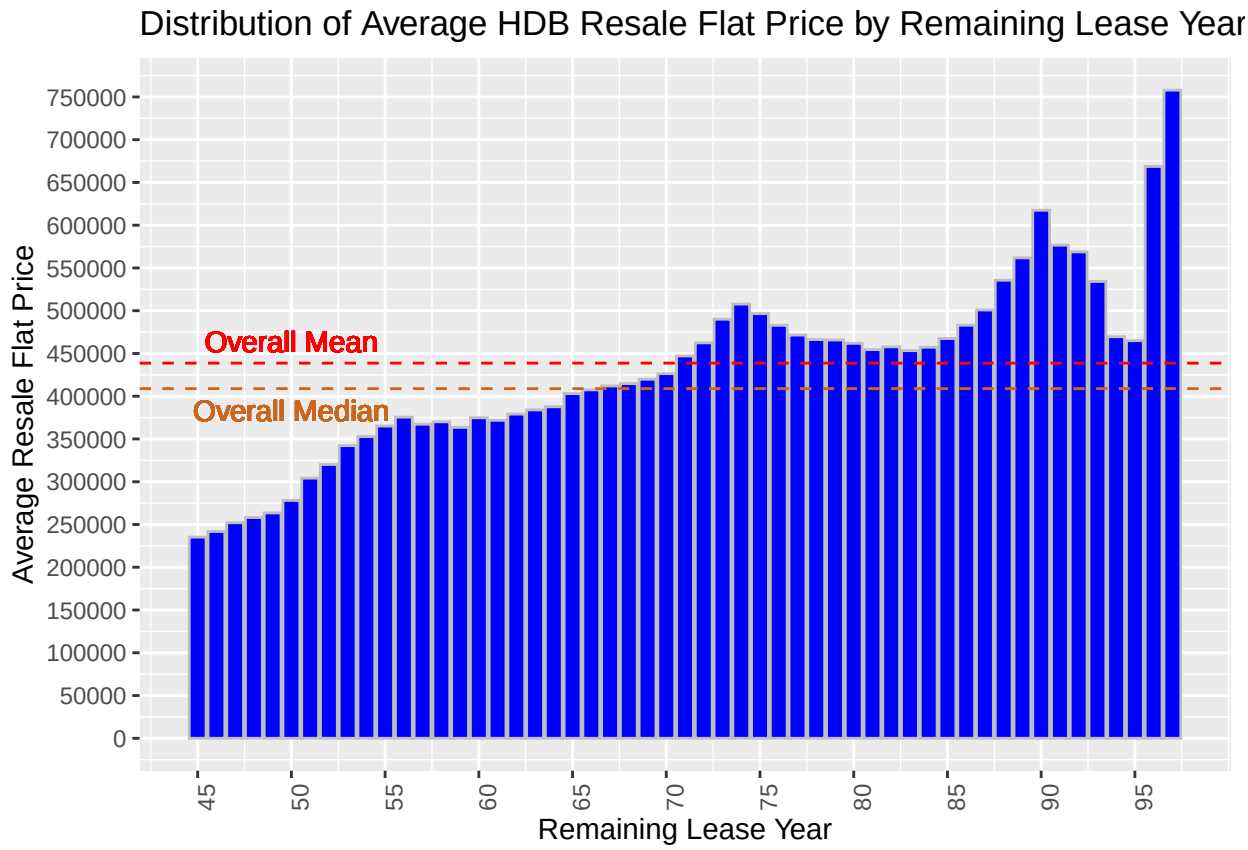
```
## # A tibble: 2 x 3
##   month   lease_commence_year resale_price
##   <chr>                 <dbl>        <dbl>
## 1 2020-07                2018       957888
## 2 2020-08                2019       612000
```

From the distribution of HDB Resale Flat Prices by Lease Commence Year, in the 2018 and 2019 lease commence year, there is only 1 and 1 transactions. It explained the spike in year 2018. It also explained the boxplot display for year 2018 and 2019. From the boxplot, it is observed there are significant number of outliers in prices for older flats.

7.2.8 Exploring the Average HDB Resale Flat Prices by Remaining Lease Year

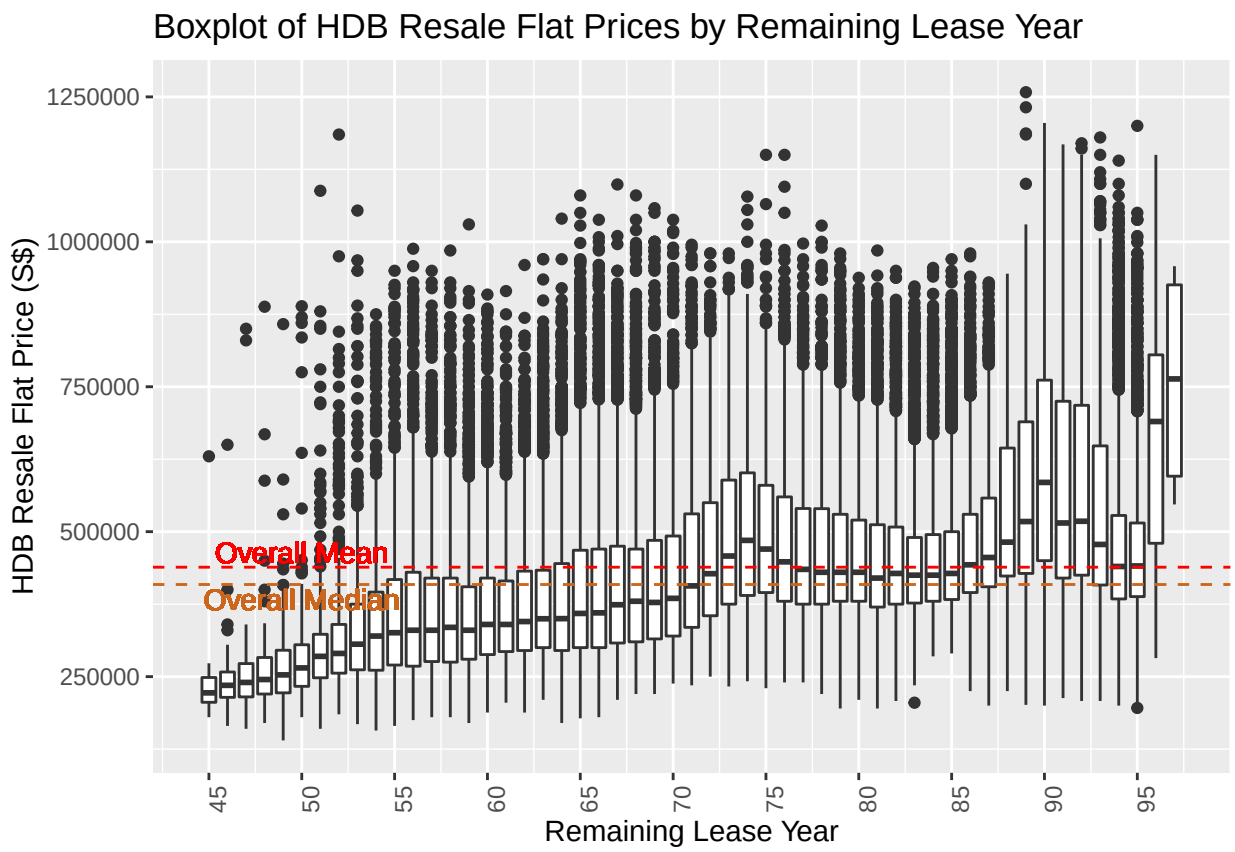
```
AvgHDBresalepricebyremainlease <- HDBResalestrain %>%
  group_by(remaining_lease_year) %>%
  summarize(Avg_Resales_Price = mean(resale_price))
#Plot Average HDB Resale Flat Price by Remaining Lease Year
AvgHDBresalepricebyremainlease %>%
```

```
  ggplot(mapping = aes(x = remaining_lease_year, y = Avg_Resales_Price)) +
  geom_col(fill = "blue", color = "grey") +
  scale_x_continuous(breaks= seq(40, 99, by=5)) +
  scale_y_continuous(breaks= seq(0, 800000, by=50000)) +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=50), colour="red", angle=0) +
  geom_hline(yintercept=median(HDBResalestrain$resale_price),
             linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=50), colour="chocolate3", angle=0) +
  labs(y = "Average Resale Flat Price", x = "Remaining Lease Year") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Distribution of Average HDB Resale Flat Price by Remaining Lease Year")
```




```
ggplot(HDBResalestrain, aes(x = remaining_lease_year, y=resale_price,
                            group=remaining_lease_year)) +
  geom_boxplot() +
  scale_x_continuous(breaks= seq(40, 99, by=5)) +
  geom_hline(yintercept= mean(HDBResalestrain$resale_price),
             linetype="dashed", color = "red") +
  geom_text(aes(y=mean(HDBResalestrain$resale_price)+25000,
                label="Overall Mean", x=50), colour="red", angle=0) +
  geom_hline(yintercept= median(HDBResalestrain$resale_price),
```

```
            linetype="dashed", color = "chocolate3") +
  geom_text(aes(y=median(HDBResalestrain$resale_price)-25000,
                label="Overall Median", x=50), colour="chocolate3", angle=0) +
  labs(x = "Remaining Lease Year", y = "HDB Resale Flat Price (S$)") +
  theme(axis.text.x= element_text(angle=90,hjust=1)) +
  ggtitle("Boxplot of HDB Resale Flat Prices by Remaining Lease Year")
```



From the above distribution, it is observed that higher prices were transacted for older flats that are 19 and 26 years old. From the above boxplots, it is also observed that there are significant number of outliers in prices for older flats.

As previously explained, older HDB flats enjoy the benefits of upgrading efforts such as the Home Improvement Programme and with the Voluntary Early Redevelopment Scheme introduced in August 2018, it further gave HDB buyers confidence in older flats. Furthermore older flats are bigger in size and the located in matured towns which amenities and transportation network are well established. Newly married couples are more inclined to buy resale flats without waiting for years for their new HDB flats to be built. The proximity housing grant also encourages buyers to buy resale flat near to their parent's flat.

8 Factorizing the categorical variables

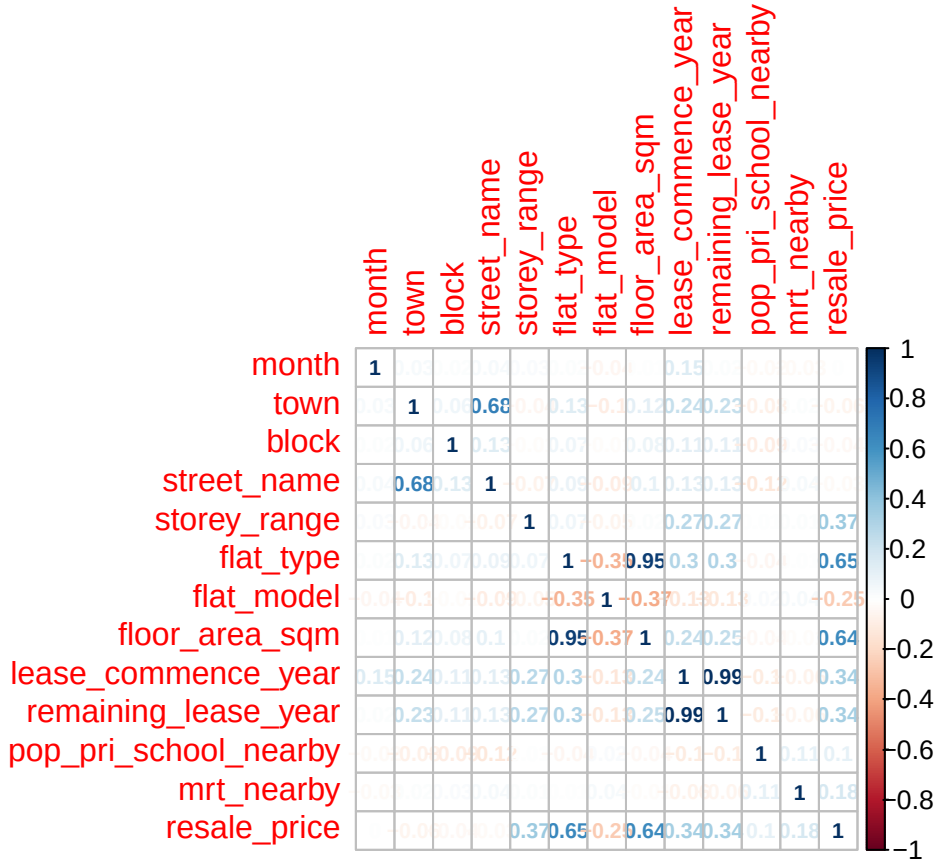
Before building any models, the categorical variables are factorized for better processing by ML algorithms.

```
tempHDBResalestrain <- HDBResalestrain %>%
  select(month, town, block, street_name, storey_range,
         flat_type, flat_model, floor_area_sqm,
```

```
         lease_commence_year, remaining_lease_year,
         pop_pri_school_nearby, mrt_nearby, resale_price)

tempHDBResalestrain$month <-
  as.numeric(factor(HDBResalestrain$month))
tempHDBResalestrain$town <-
  as.numeric(factor(HDBResalestrain$town))
tempHDBResalestrain$block <-
  as.numeric(factor(HDBResalestrain$block))
tempHDBResalestrain$street_name <-
  as.numeric(factor(HDBResalestrain$street_name))
tempHDBResalestrain$storey_range <-
  as.numeric(factor(HDBResalestrain$storey_range))
tempHDBResalestrain$flat_type <-
  as.numeric(factor(HDBResalestrain$flat_type))
tempHDBResalestrain$flat_model <-
  as.numeric(factor(HDBResalestrain$flat_model))
tempHDBResalestrain$lease_commence_year <-
  as.numeric(factor(HDBResalestrain$lease_commence_year))
tempHDBResalestrain$remaining_lease_year <-
  as.numeric(factor(HDBResalestrain$remaining_lease_year))

tempHDBResalesvalidation <- HDBResalesvalidation %>%
  select(month, town, block, street_name, storey_range,
         flat_type, flat_model, floor_area_sqm,
         lease_commence_year, remaining_lease_year,
         pop_pri_school_nearby, mrt_nearby, resale_price)

tempHDBResalesvalidation$month <-
  as.numeric(factor(HDBResalesvalidation$month))
tempHDBResalesvalidation$town <-
  as.numeric(factor(HDBResalesvalidation$town))
tempHDBResalesvalidation$block <-
  as.numeric(factor(HDBResalesvalidation$block))
tempHDBResalesvalidation$street_name <-
  as.numeric(factor(HDBResalesvalidation$street_name))
tempHDBResalesvalidation$storey_range <-
  as.numeric(factor(HDBResalesvalidation$storey_range))
tempHDBResalesvalidation$flat_type <-
  as.numeric(factor(HDBResalesvalidation$flat_type))
tempHDBResalesvalidation$flat_model <-
  as.numeric(factor(HDBResalesvalidation$flat_model))
tempHDBResalesvalidation$lease_commence_year <-
  as.numeric(factor(HDBResalesvalidation$lease_commence_year))
tempHDBResalesvalidation$remaining_lease_year <-
  as.numeric(factor(HDBResalesvalidation$remaining_lease_year))
```

9 Checking the Correlations among the Variables within the HDBResalestrain dataset

```
corresult<- cor(tempHDBResalestrain)
corrplot(corresult, method = "number", number.cex = .7)
```



It is observed that lease_commence_year and remaining_lease_year variables have the highest positive correlation of 0.99. It is followed by floor_area_sqm and flat_type variables with a strong positive correlation of 0.95.

10 Detecting Multicollinearity with Variance Inflation Factors (VIF)

Multicollinearity occurs when two or more independent variables are highly correlated with one another in a regression model. Variance Inflation Factors (VIF) is used to detect multicollinearity. VIF value exceeding 5 or 10 indicates high multicollinearity between this independent variable and the others.

```
#Train the model
modellm = lm(resale_price ~ .,
             data = tempHDBResalestrain)
#Detecting Multicollinearity with Variance Inflation Factors (VIF)
vif(modellm)
```

```
##                month                town               block
##            16.504294            1.995861            1.042112
##          street_name         storey_range           flat_type
##             1.963982            1.114678           10.956755
##           flat_model       floor_area_sqm lease_commence_year
##             1.172586           10.867786          887.239008
##  remaining_lease_year pop_pri_school_nearby           mrt_nearby
##           868.019899            1.041552            1.028994
```

It is observed that month, flat_type, floor_area_sqm, lease_commence_year and remaining_lease_year variables have a VIF value greater than 10. It is also observed in the above correlation plot that flat type and floor_area_sq are strongly correlated, and it is the same for lease_commence_year and remaining_lease_year. But it is unexpected to observe that the VIF for month variable has a value greater than 10.

To remove multicollinearity, the attribute that has the highest VIF value will be removed at a time until the model is free of multicollinearity. The lease_commence_year variable has the highest VIF value of 887.239 and to be removed first.

```
#Remove lease_commence_year variable
modellm <- update(modellm, .~. - lease_commence_year,
                  data = tempHDBResalestrain)
vif(modellm)
```

```
##                month                town               block
##             1.006035            1.994089            1.041967
##          street_name         storey_range           flat_type
##             1.961266            1.112749           10.953938
##           flat_model       floor_area_sqm  remaining_lease_year
##             1.171788           10.861396            1.292838
## pop_pri_school_nearby           mrt_nearby
##             1.041515            1.028178
```

It is observed that month, flat_type, floor_area_sqm, have a VIF value greater than 10. The flat_type variable has the highest VIF value of 10.953 and to be removed.

```
#Remove flat_type variable
modellm <- update(modellm, .~. - flat_type,
                  data = tempHDBResalestrain)
vif(modellm)
```

```
##                month                town               block
##             1.005021            1.982651            1.041257
##          street_name         storey_range          flat_model
##             1.943850            1.104179            1.170507
##       floor_area_sqm  remaining_lease_year pop_pri_school_nearby
##             1.224708            1.247616            1.041096
##           mrt_nearby
##             1.021764
```

It is observed that there is no VIF value greater than 5 or 10.

11 Derive the Multiple Regression Formula

```
#Print the model summary
summary(modellm)
```

```
##
## Call:
## lm(formula = resale_price ~ month + town + block + street_name +
##     storey_range + flat_model + floor_area_sqm + remaining_lease_year +
##     pop_pri_school_nearby + mrt_nearby, data = tempHDBResalestrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -385020  -59914  -13362   40760  594721
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -3.543e+04  2.190e+03 -16.175  < 2e-16 ***
## month                  2.288e+01  1.622e+01   1.411 0.158360
## town                  -3.270e+03  5.582e+01 -58.581  < 2e-16 ***
## block                 -1.914e+01  4.708e-01 -40.649  < 2e-16 ***
## street_name            3.993e+01  2.685e+00  14.871  < 2e-16 ***
## storey_range           2.382e+04  1.786e+02 133.400  < 2e-16 ***
## flat_model            -3.308e+02  9.313e+01  -3.552 0.000383 ***
## floor_area_sqm         3.775e+03  1.454e+01 259.565  < 2e-16 ***
## remaining_lease_year   2.093e+03  2.910e+01  71.925  < 2e-16 ***
## pop_pri_school_nearby  5.068e+04  1.050e+03  48.277  < 2e-16 ***
## mrt_nearby             5.594e+04  6.583e+02  84.976  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91440 on 82256 degrees of freedom
## Multiple R-squared:  0.6207, Adjusted R-squared:  0.6206
## F-statistic: 1.346e+04 on 10 and 82256 DF,  p-value: < 2.2e-16
```

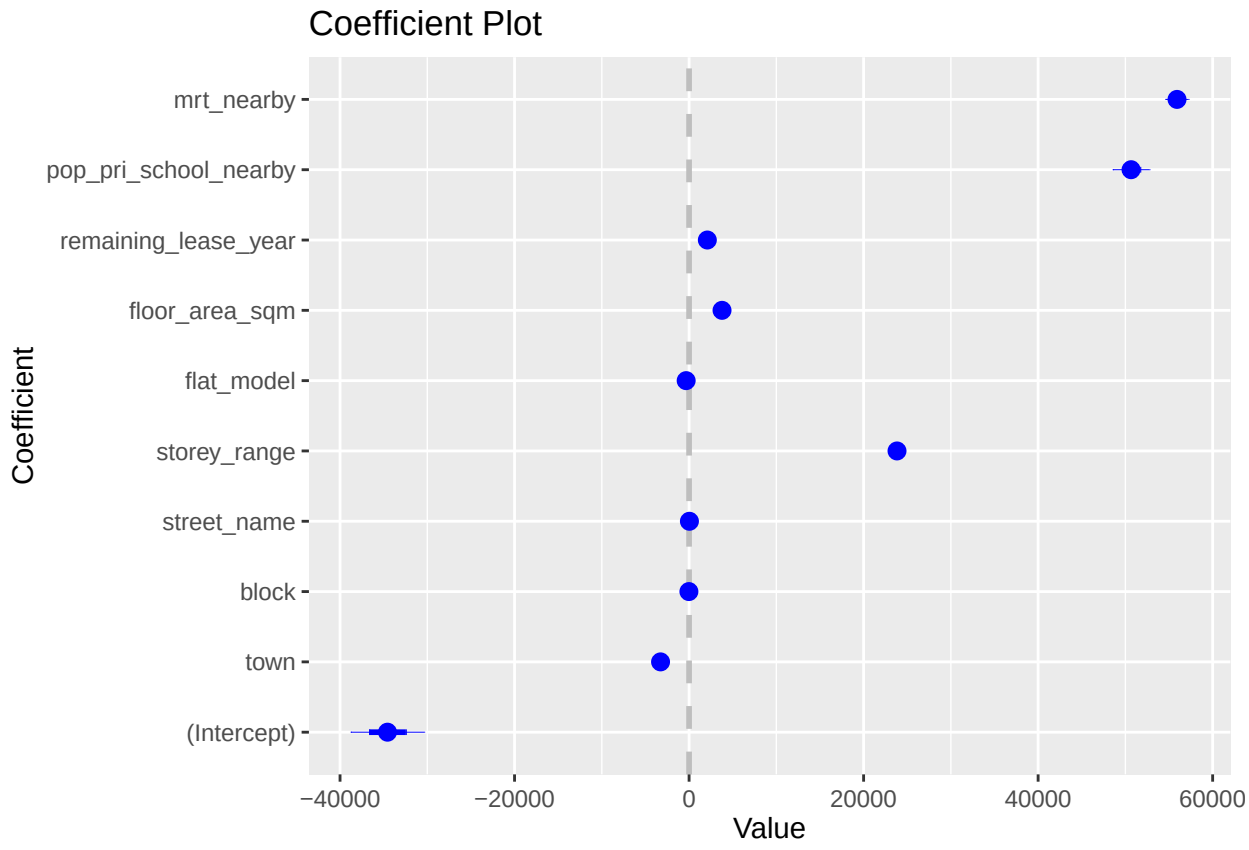
It is observed that "month" variable is not statistically significant because it has a P value of more than 0.05. Therefore, it is removed.

```
#Remove month variable because it is not significant.
modellm <- update(modellm, .~. - month, data = tempHDBResalestrain)
summary(modellm)
```

```
##
## Call:
## lm(formula = resale_price ~ town + block + street_name + storey_range +
##     flat_model + floor_area_sqm + remaining_lease_year + pop_pri_school_nearby +
##     mrt_nearby, data = tempHDBResalestrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -385409  -59905  -13393   40716  594550
```

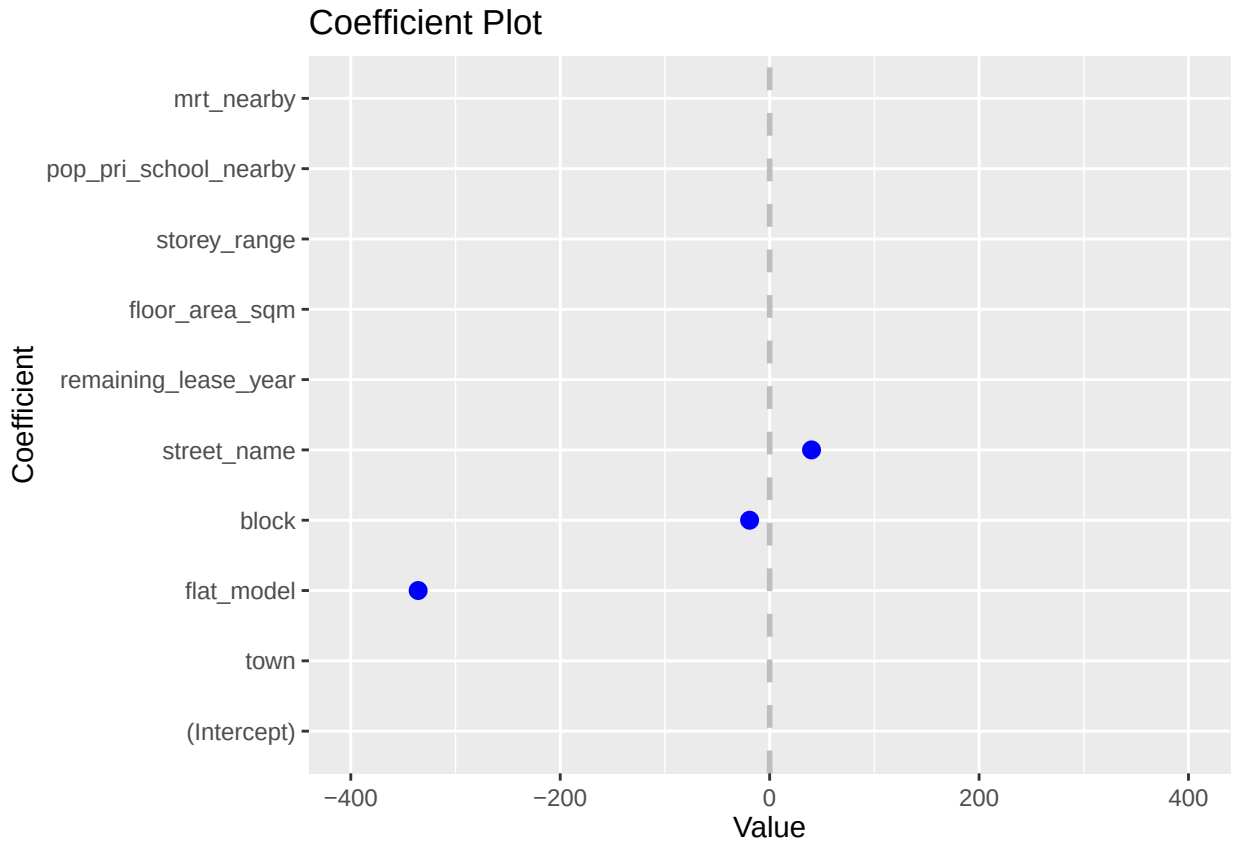
```
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -3.456e+04  2.103e+03 -16.437  < 2e-16 ***
## town                  -3.269e+03  5.581e+01 -58.570  < 2e-16 ***
## block                 -1.913e+01  4.707e-01 -40.633  < 2e-16 ***
## street_name            4.000e+01  2.685e+00  14.898  < 2e-16 ***
## storey_range           2.383e+04  1.785e+02 133.484  < 2e-16 ***
## flat_model            -3.357e+02  9.307e+01  -3.607  0.00031 ***
## floor_area_sqm         3.775e+03  1.454e+01 259.567  < 2e-16 ***
## remaining_lease_year   2.093e+03  2.910e+01  71.920  < 2e-16 ***
## pop_pri_school_nearby  5.066e+04  1.050e+03  48.261  < 2e-16 ***
## mrt_nearby             5.592e+04  6.581e+02  84.970  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91440 on 82257 degrees of freedom
## Multiple R-squared:  0.6207, Adjusted R-squared:  0.6206
## F-statistic: 1.495e+04 on 9 and 82257 DF,  p-value: < 2.2e-16
```

```
#Using coefficient plot to visualize the regression results
coefplot(modellm)
```



It is observed that all the variables are statistically significant because there are no variables' two standard error confidence interval contained 0. mrt_nearby variable has the largest effect on the HDB resale flat price.From the coefficient plot, flat_model, block and street_name variables seems to have little effect on the resale flat price.

```
#Enlarge to check those variables that seems to have little effect on resale flat prices.
coefplot(modellm, sort='mag') + scale_x_continuous(limits=c(-400, 400))
```



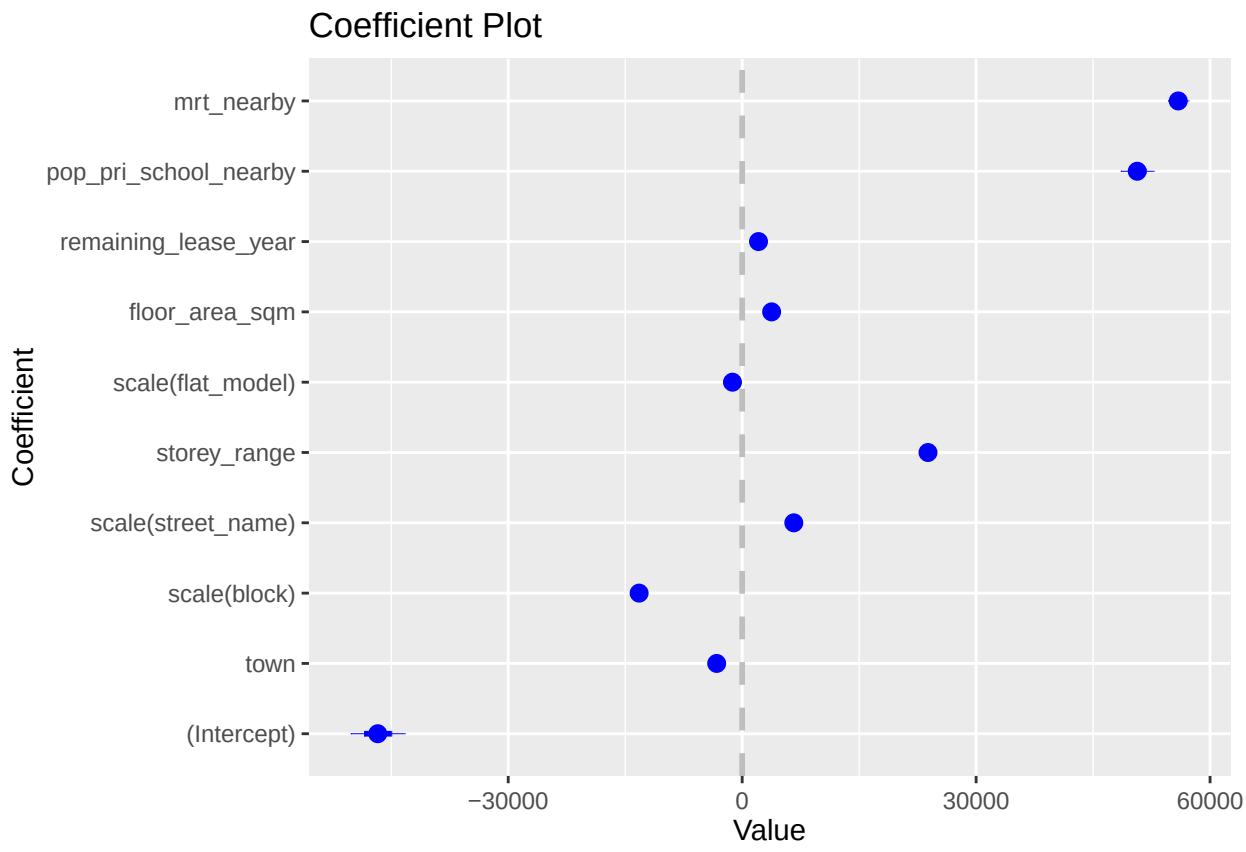
Zooming into the coefficient plot found that the coefficients for flat_model, block and street_name variables are non-zero. This could be a scaling issue. This can be resolved by scaling these three variables. It subtracts the mean and divides by the standard deviation.

```
#Train the model
modellm = lm(resale_price ~ town +
               scale(block) +
               scale(street_name) +
               storey_range +
               scale(flat_model) +
               floor_area_sqm +
               remaining_lease_year +
               pop_pri_school_nearby +
               mrt_nearby, data = tempHDBResalestrain)
summary(modellm)
```

```
##
## Call:
## lm(formula = resale_price ~ town + scale(block) + scale(street_name) +
##     storey_range + scale(flat_model) + floor_area_sqm + remaining_lease_year +
##     pop_pri_school_nearby + mrt_nearby, data = tempHDBResalestrain)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -385409  -59905  -13393   40716  594550
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -46731.09    1728.51 -27.036  < 2e-16 ***
## town                   -3268.98      55.81 -58.570  < 2e-16 ***
## scale(block)          -13216.99     325.28 -40.633  < 2e-16 ***
## scale(street_name)      6620.94     444.41  14.898  < 2e-16 ***
## storey_range           23827.78     178.51 133.484  < 2e-16 ***
## scale(flat_model)      -1243.19     344.67  -3.607  0.00031 ***
## floor_area_sqm          3774.70      14.54 259.567  < 2e-16 ***
## remaining_lease_year    2092.66      29.10  71.920  < 2e-16 ***
## pop_pri_school_nearby  50658.22    1049.67  48.261  < 2e-16 ***
## mrt_nearby             55917.12     658.08  84.970  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 91440 on 82257 degrees of freedom
## Multiple R-squared:  0.6207, Adjusted R-squared:  0.6206
## F-statistic: 1.495e+04 on 9 and 82257 DF,  p-value: < 2.2e-16
```

```
#Review Coefficient Plot after scaling
coefplot(modellm)
```



12 Finalizing the Multiple Regression Formula

```
#Derive the final formula
HDBResalePriceFormula <- resale_price ~ town +
  scale(block) +
  scale(street_name) +
  storey_range +
  scale(flat_model) +
  floor_area_sqm +
  remaining_lease_year +
  pop_pri_school_nearby +
  mrt_nearby
```

13 Evaluating the Models using Residual Mean Squared Error (RMSE)

A typical performance measure for regression model is the Root Mean Square Error (RMSE). It gives an idea of how much error the model typically makes in its predictions, with higher weight for large errors.

```
#Define the Loss function
RMSE <- function(actual, predicted){
    sqrt(mean((actual - predicted)^2))
}
#Create a RMSE results dataframe to store and compare the results.
RMSE_results <- tibble()
```

14 Define the Train Control for the Models

A 10-fold cross-validation will be used throughout this project.

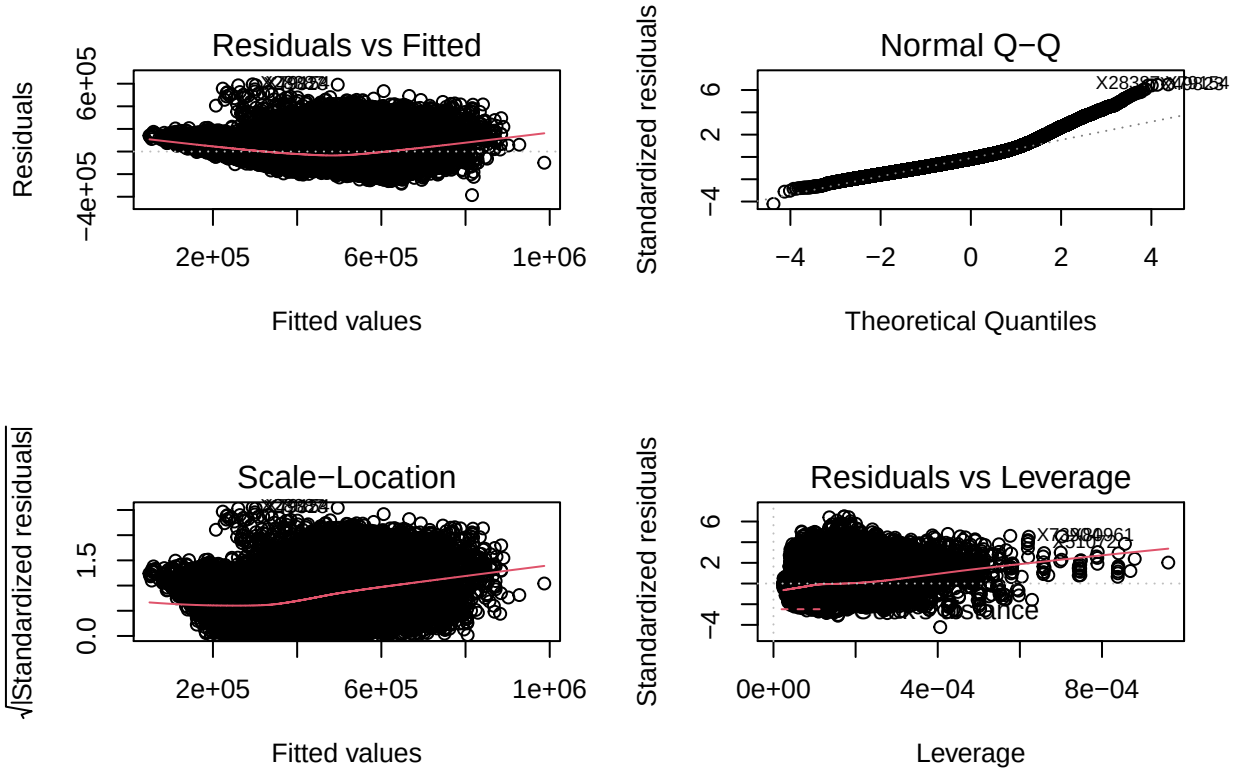
```
#10-fold cross-validation
TrainCtrl <- trainControl(method = "cv", number = 10, verboseIter = FALSE)
```

15 Building and Evaluating the Models

15.1 Exploring Multiple Regression Model

```
set.seed(123, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(123)`
#Train the model
HDBResalesPriceModellm <- train(HDBResalePriceFormula,
                                data=tempHDBResalestrain,
                                method = "lm",
                                trControl = TrainCtrl)
```

```r
par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(HDBResalesPriceModellm$finalModel)  # Plot the model information
```



From The Residuals vs. Fitted plot, it is observed that non-linearity is present. Furthermore, the variance appears to be increased with the fitted values. From the Normal QQ plot, the residuals appeared highly non normal. The upper tail is heavier than expected under normality. This meant the model is not a good fit. This may be due to the non-constant variance observed in the Residuals vs. Fitted plot. From Scale-location plot, it is observed that there is an increasing trend in residual variance. This is indicated by the upward slope of the red line. On Residuals vs Leverage plot, none of the points appeared to be outliers.

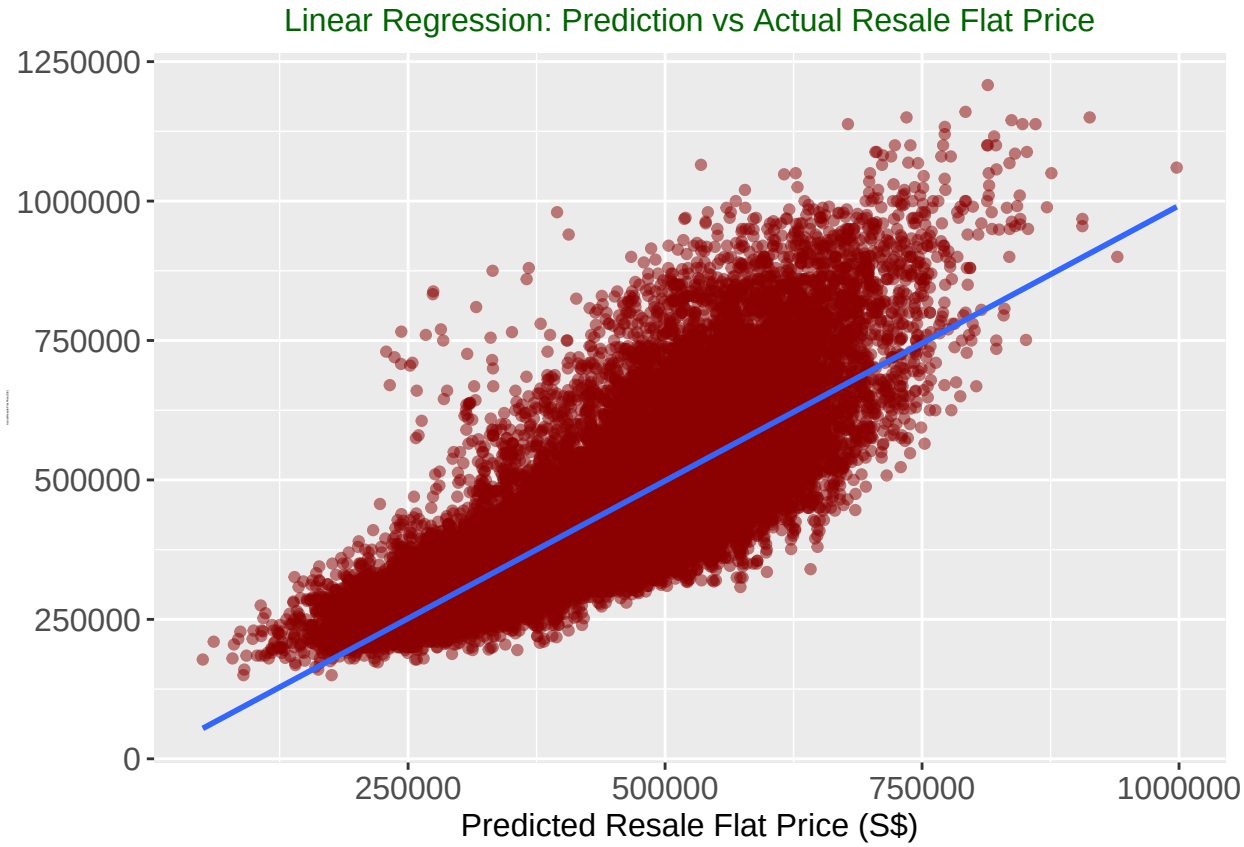
15.2 Using Linear Regression Model to predict HDB Resale Flat Prices

```r
#Make prediction with the validation dataset
HDBResalePricePredictlm <- predict(HDBResalesPriceModellm, newdata = tempHDBResalesvalidation)
#Calculate RMSE
rmse <- RMSE(tempHDBResalesvalidation$resale_price, HDBResalePricePredictlm)
RMSE_results <- tibble(Model = "lm", RMSE_value = rmse)
RMSE_results
```

```
## # A tibble: 1 x 2
##   Model RMSE_value
##   <chr>      <dbl>
## 1 lm        90626.92
```

15.3 Plot Linear Regression Model predictions vs Actual Resale Flat Prices

```
Plotdata = as.data.frame(cbind(predicted = HDBResalePricePredictlm,
                               actual = tempHDBResalesvalidation$resale_price))
ggplot(Plotdata,aes(predicted, actual)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method=lm) +
  ggtitle("Linear Regression: Prediction vs Actual Resale Flat Price") +
  xlab("Predicted Resale Flat Price (S$)") +
  ylab("Actual Resale Flat Price (S$)") +
  theme(plot.title = element_text(color="darkgreen",size=12,hjust = 0.5),
        axis.text.y = element_text(size=12),
        axis.text.x = element_text(size=12,hjust=.5),
        axis.title.x = element_text(size=12),
        axis.title.y = element_text(size=1))
```



15.4 Decision Tree Approaches

The above regression approach has achieved a RMSE value of 90,626.92. To further improve the RMSE result, Decision Tree approaches have been explored. Decision Trees are machine learning algorithms that can perform both regression and classification tasks. It is a technique for fitting nonlinear models. Individual decision trees generally do not often achieve state-of-the-art predictive accuracy (Bradley, B & Brandon, G, 2020). However, powerful ensemble algorithms, such as Bagged Decision Tree, Random Forests and Gradient

Boosting Machines, which are constructed by combining together many decision trees in a clever way, can achieve much better predictive accuracy.

15.4.1 Bootstrap aggregating (Bagging) Ensemble Algorithms

Bootstrap aggregating (bagging) is a general method for fitting multiple versions of a prediction model and then combining (or ensembling) them into an aggregated prediction (Breiman 1996a). It is designed to improve the stability and accuracy of regression and classification algorithms. By model averaging, bagging helps to reduce variance and minimize overfitting (Bradley, B & Brandon, G, 2020).

15.4.1.1 Exploring Bagged Decision Tree Model

Bagged Decision Tree Model is a ensemble algorithm based on bagging method.

```
set.seed(123, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(123)`
#Train the model
HDBResalePriceModelTreeBag <- train(HDBResalePriceFormula,
                                    data = tempHDBResalestrain,
                                    method = "treebag",
                                    trControl = TrainCtrl,
                                    nbagged = 200,
                                    control = rpart.control(minsplit = 2, cp = 0))
```

```
#Print the model results
HDBResalePriceModelTreeBag
```

```
## Bagged CART
##
## 82267 samples
##     9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 74040, 74040, 74040, 74040, 74040, 74041, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   27793.81  0.9649473  19239.62
```

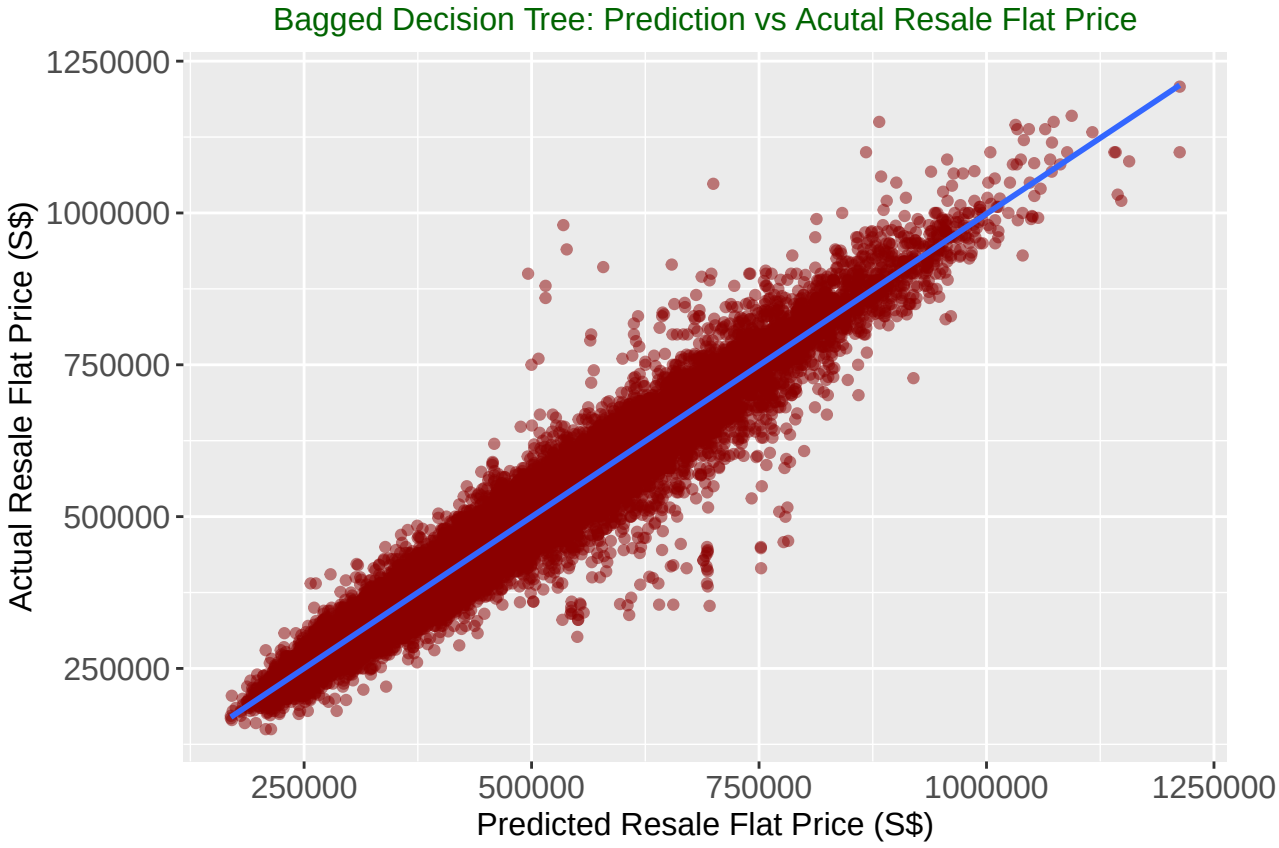
15.4.1.2 Using Bagged Decision Tree to predict HDB Resale Flat Prices

```
#Make prediction with the validation dataset
HDBResalePricePredictTreeBag <- predict(HDBResalePriceModelTreeBag,
                                        newdata = tempHDBResalesvalidation)
#Calculate RMSE
rmse <- RMSE(tempHDBResalesvalidation$resale_price,
             HDBResalePricePredictTreeBag)
RMSE_results <- RMSE_results %>% add_row(Model = "Bagged Decision Tree",
                                         RMSE_value = rmse)
RMSE_results
```

```
## # A tibble: 2 x 2
##   Model                RMSE_value
##   <chr>                     <dbl>
## 1 lm                       90626.92
## 2 Bagged Decision Tree     31480.44
```

15.4.1.3 Plot Bagged Decision Tree predictions vs Actual Resale Flat Prices

```
PlotdataBaggedTree = as.data.frame(cbind(predicted = HDBResalePricePredictTreeBag,
                                         actual = tempHDBResalesvalidation$resale_price))
ggplot(PlotdataBaggedTree,aes(predicted, actual)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method=lm) +
  ggtitle("Bagged Decision Tree: Prediction vs Acutal Resale Flat Price") +
  xlab("Predicted Resale Flat Price (S$)") +
  ylab("Actual Resale Flat Price (S$)") +
  theme(plot.title = element_text(color="darkgreen",size=12,hjust = 0.5),
        axis.text.y = element_text(size=12),
        axis.text.x = element_text(size=12,hjust=.5),
        axis.title.x = element_text(size=12),
        axis.title.y = element_text(size=12))
```



15.4.1.4 Exploring Random Forest Model

Bagged Decision Tree model has a characteristic known as tree correlation and prevents bagging from further reducing the variance of the base learner. Random forests are a modification of Bagged Decision Trees that build a large collection of de-correlated trees to further improve predictive performance.

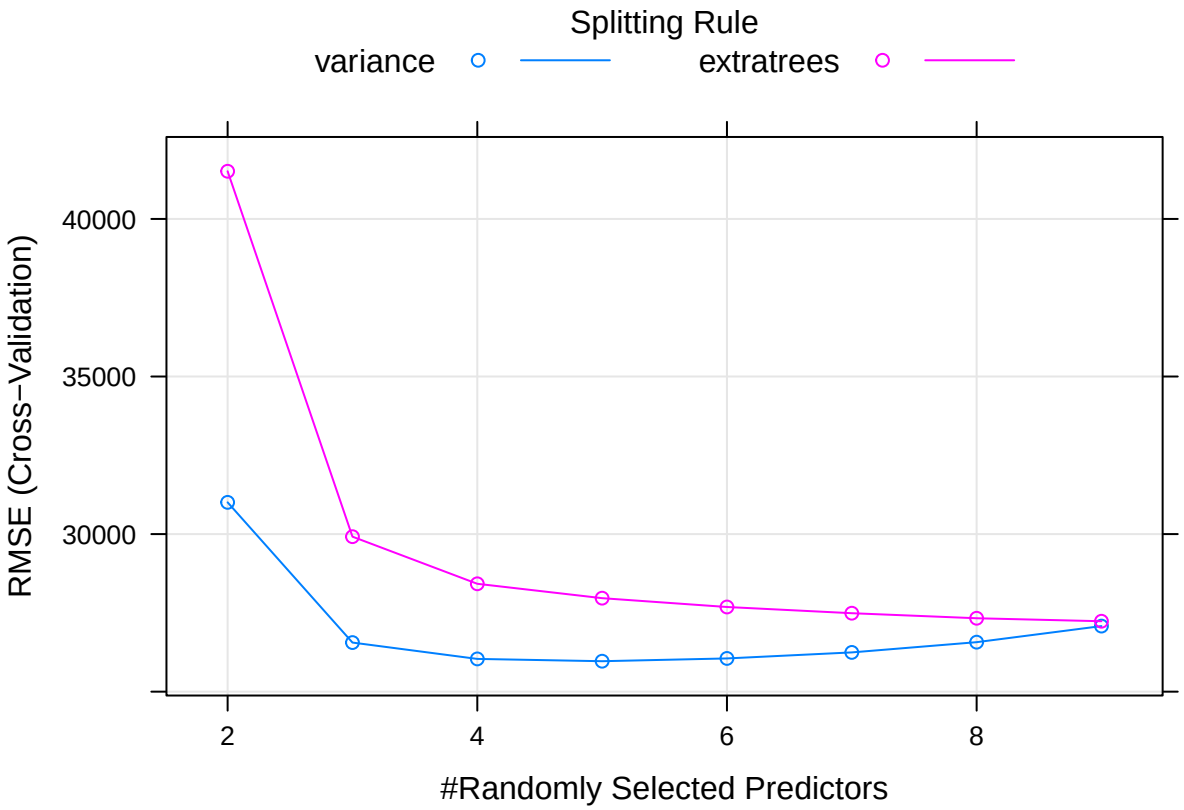
```r
set.seed(123, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(123)`
#Train the model
HDBResalePriceModelRF <- train(HDBResalePriceFormula,
                               data=tempHDBResalestrain,
                               method = "ranger",
                               importance = 'impurity',
                               tuneLength = 10,
                               trControl = TrainCtrl)
```

```
## note: only 8 unique complexity parameters in default grid. Truncating the grid to 8 .
```

```r
#Print model results
HDBResalePriceModelRF
```

```
## Random Forest
##
## 82267 samples
##     9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 74040, 74040, 74040, 74040, 74040, 74041, ...
## Resampling results across tuning parameters:
##
##   mtry  splitrule   RMSE      Rsquared   MAE
##   2     variance    31007.27  0.9596165  22063.90
##   2     extratrees  41513.27  0.9418232  30113.69
##   3     variance    26557.69  0.9683536  18720.53
##   3     extratrees  29918.13  0.9615464  21021.04
##   4     variance    26038.87  0.9693851  18335.89
##   4     extratrees  28422.46  0.9642457  19821.17
##   5     variance    25965.82  0.9694976  18279.80
##   5     extratrees  27966.57  0.9651175  19483.15
##   6     variance    26053.87  0.9692608  18329.07
##   6     extratrees  27686.44  0.9656803  19311.98
##   7     variance    26245.52  0.9687883  18413.74
##   7     extratrees  27489.43  0.9660869  19203.33
##   8     variance    26570.43  0.9679955  18547.93
##   8     extratrees  27329.72  0.9664278  19114.97
##   9     variance    27080.80  0.9667421  18753.25
##   9     extratrees  27234.88  0.9666236  19064.93
##
## Tuning parameter 'min.node.size' was held constant at a value of 5
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were mtry = 5, splitrule = variance
##  and min.node.size = 5.
```

```r
#Plot the RMSE
plot(HDBResalePriceModelRF)
```



15.4.1.5 Using Random Forest to predict HDB Resale Flat Prices

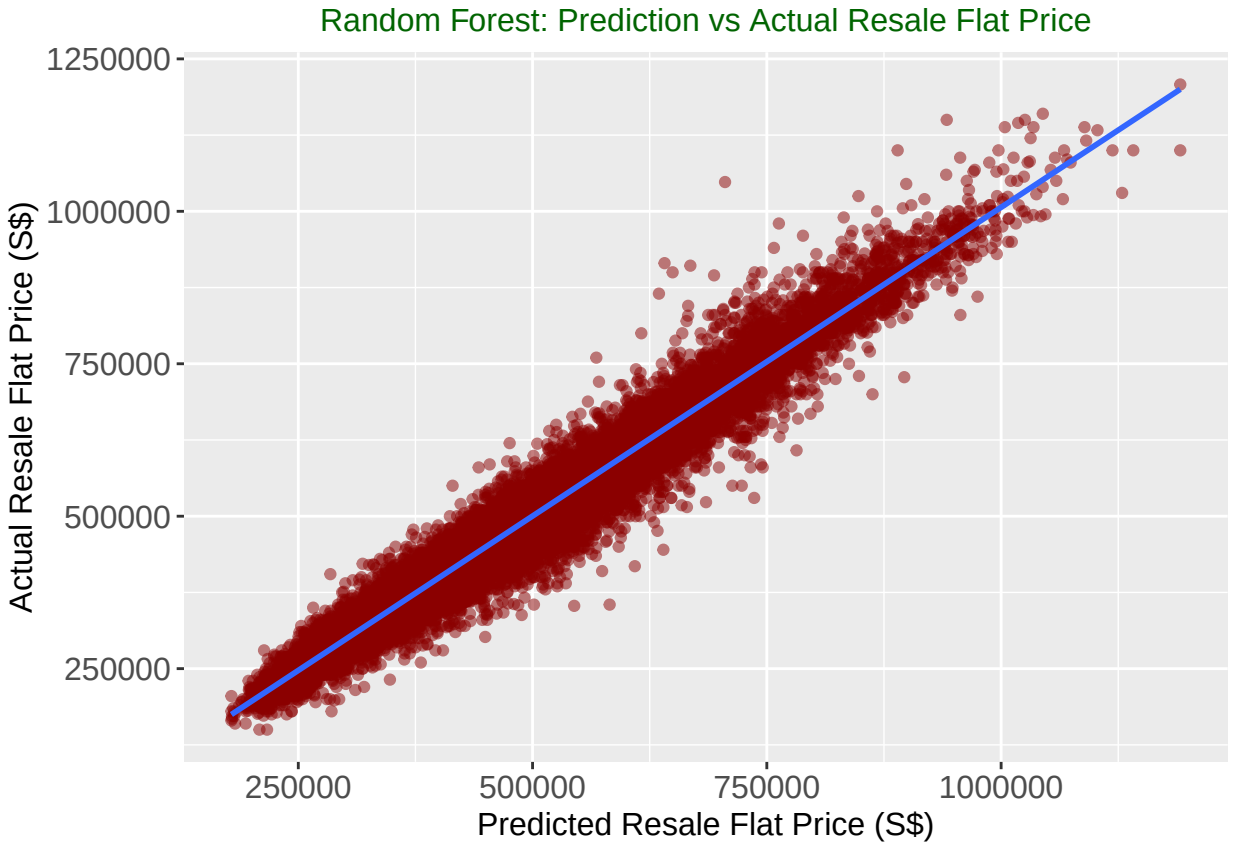
```
#Make prediction with the validation dataset
HDBResalePricePredictRF <- predict(HDBResalePriceModelRF,
                                   newdata = tempHDBResalesvalidation)
#Calculate RMSE
rmse <- RMSE(tempHDBResalesvalidation$resale_price,
             HDBResalePricePredictRF)
RMSE_results <- RMSE_results %>%
  add_row(Model = "Random Forest", RMSE_value = rmse)
RMSE_results
```

```
## # A tibble: 3 x 2
##   Model                RMSE_value
##   <chr>                     <dbl>
## 1 lm                       90626.92
## 2 Bagged Decision Tree     31480.44
## 3 Random Forest            27890.49
```

15.4.1.6 Plot Random Forest predictions vs Actual Resale Flat Prices

```
PlotdataRF = as.data.frame(cbind(predicted = HDBResalePricePredictRF,
                                 actual = tempHDBResalesvalidation$resale_price))
```

```
ggplot(PlotdataRF,aes(predicted, actual)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method=lm) +
  ggtitle("Random Forest: Prediction vs Actual Resale Flat Price") +
  xlab("Predicted Resale Flat Price (S$)") +
  ylab("Actual Resale Flat Price (S$)") +
  theme(plot.title = element_text(color="darkgreen",size=12,hjust = 0.5),
        axis.text.y = element_text(size=12),
        axis.text.x = element_text(size=12,hjust=.5),
        axis.title.x = element_text(size=12),
        axis.title.y = element_text(size=12))
```



15.4.2 Boosting Ensemble Algorithms

From above, Random Forest has significantly improved the RSME value to 27,891.70. Taking another ensemble approach, Boosting algorithms is explored to further improve the RMSE value.

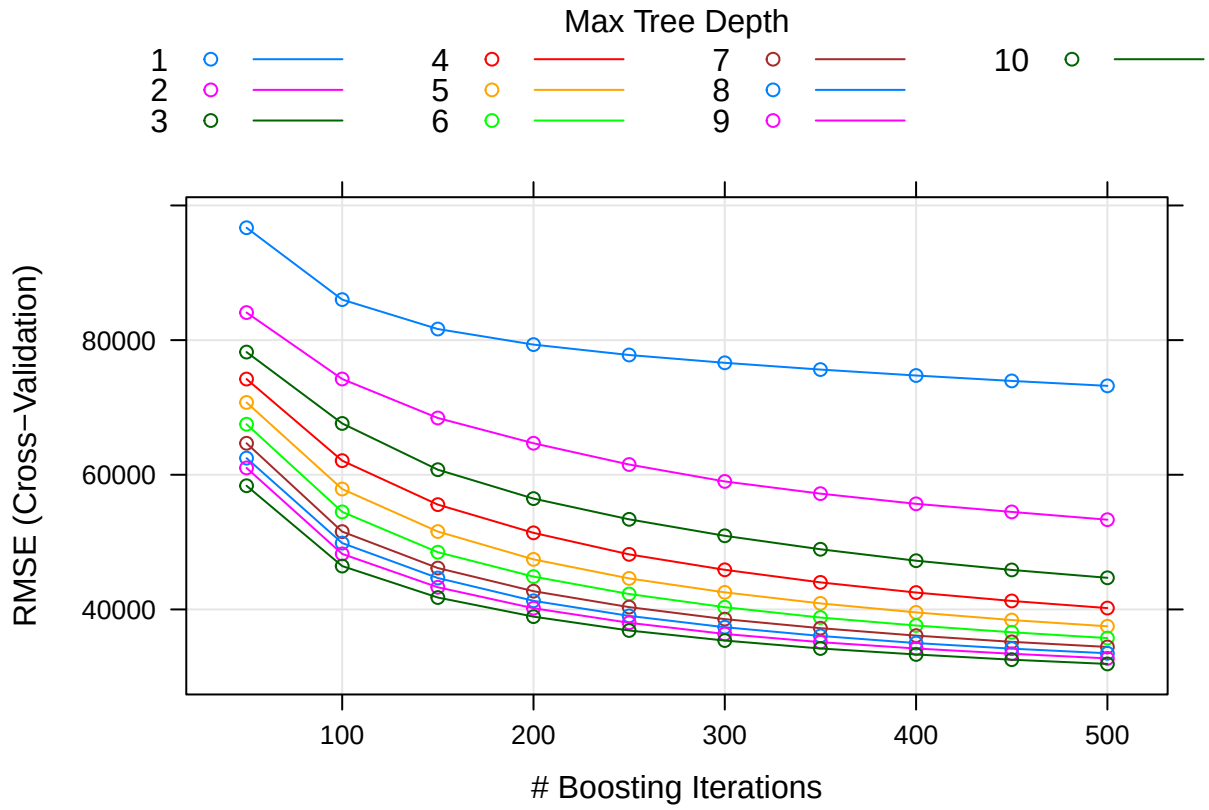
Boosting refers to any ensemble method that can combine several weak learners into a strong learner. Instead of training the predictors independently from each other in parallel like bagging, the general concept of boosting is to train the predictors sequentially, each correcting its predecessor. It is more effectively applied to models with high bias and low variance.

15.4.2.1 Exploring Stochastic Gradient Boosting Machine (GBM) Model

Gradient boosting machine (GBM) is an additive modeling algorithm that gradually builds a composite model by iteratively adding weak sub-models based on the performance of the prior iteration's composite (Bradley, B & Brandon, G, 2020).

```
set.seed(123, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(123)`
#Train the model
HDBResalePriceModelGBM <- train(HDBResalePriceFormula,
                                data = tempHDBResalestrain,
                                method = "gbm",
                                tuneLength = 10,
                                trControl = TrainCtrl,
                                verbose = FALSE)
```

```
#Plot the RMSE
plot(HDBResalePriceModelGBM)
```



15.4.2.2 Using Stochastic GBM to predict HDB Resale Flat Prices

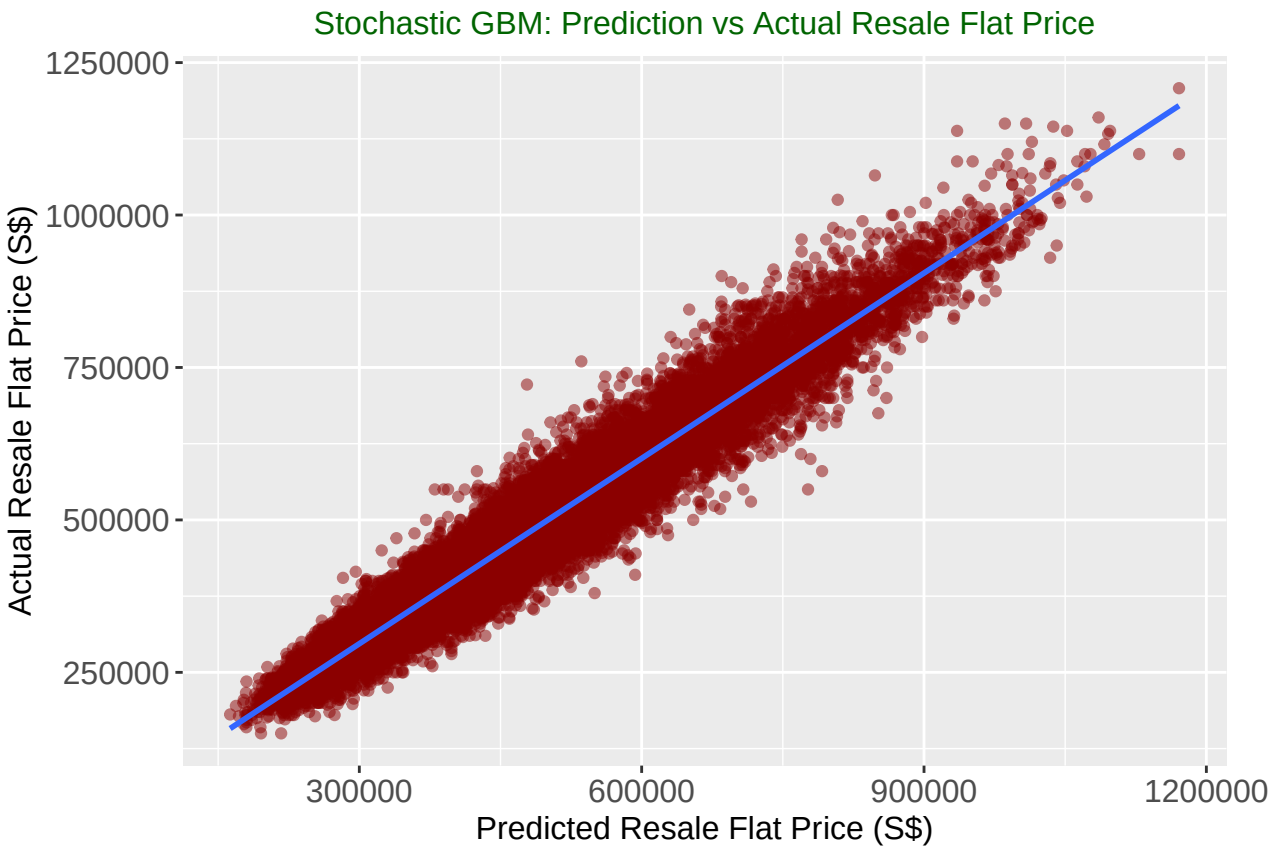
```
#Make prediction with the validation dataset
HDBResalePricePredictGBM <- predict(HDBResalePriceModelGBM,
                                    newdata = tempHDBResalesvalidation)
#Calculate RMSE
rmse <- RMSE(tempHDBResalesvalidation$resale_price,
             HDBResalePricePredictGBM)
RMSE_results <- RMSE_results %>%
```

```
  add_row(Model = "Stochastic Gradient Boosting", RMSE_value = rmse)
RMSE_results
```

```
## # A tibble: 4 x 2
##   Model                          RMSE_value
##   <chr>                               <dbl>
## 1 lm                                90626.92
## 2 Bagged Decision Tree              31480.44
## 3 Random Forest                     27890.49
## 4 Stochastic Gradient Boosting      33955.43
```

15.4.2.3 Plot Stochastic GBM predictions vs Actual Resale Flat Prices

```
PlotdataGBM = as.data.frame(cbind(predicted = HDBResalePricePredictGBM,
                                  actual = tempHDBResalesvalidation$resale_price))
ggplot(PlotdataGBM,aes(predicted, actual)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method=lm) +
  ggtitle("Stochastic GBM: Prediction vs Actual Resale Flat Price") +
  xlab("Predicted Resale Flat Price (S$)") +
  ylab("Actual Resale Flat Price (S$)") +
  theme(plot.title = element_text(color="darkgreen",size=12,hjust = 0.5),
        axis.text.y = element_text(size=12),
        axis.text.x = element_text(size=12,hjust=.5),
        axis.title.x = element_text(size=12),
        axis.title.y = element_text(size=12))
```



15.4.2.4 Exploring Extreme Gradient Boosting Model

Extreme Gradient Boosting is an extension to gradient boosted decision trees (GBM) and specially designed to improve speed and performance through parallel and distributed computing. It also avoid overfitting through regularisation.

```
set.seed(123, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(123)`
#Train the model
HDBResalePriceModelXGB <- train(HDBResalePriceFormula,
                                data = tempHDBResalestrain,
                                method = "xgbTree",
                                tuneLength = 10,
                                trControl = TrainCtrl,
                                objective ="reg:squarederror",
                                verbose = FALSE)
```

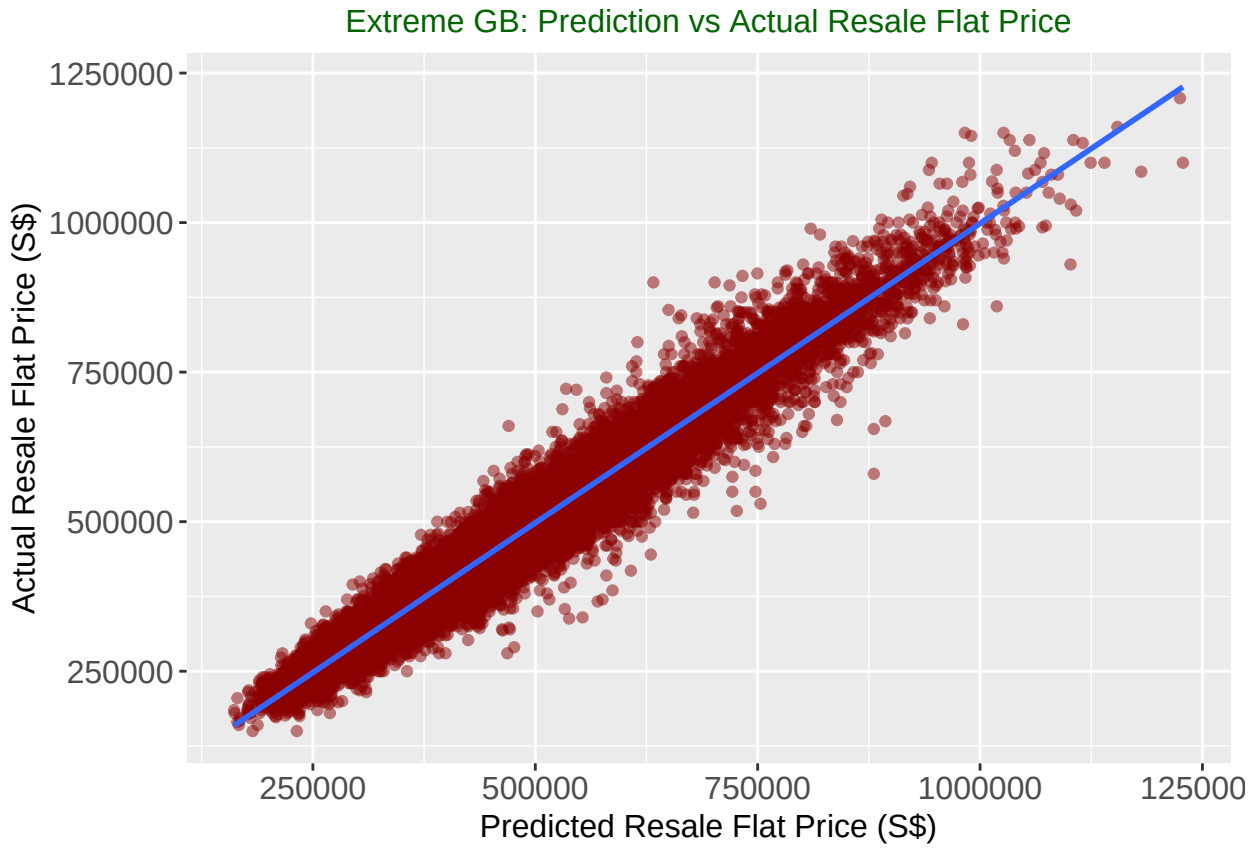
15.4.2.5 Using Extreme Gradient Boosting to predict HDB Resale Flat Prices

```
#Make prediction with the validation dataset
HDBResalePricePredictXGB <- predict(HDBResalePriceModelXGB,
                                    newdata = tempHDBResalesvalidation)
#Calculate RMSE
rmse <- RMSE(tempHDBResalesvalidation$resale_price, HDBResalePricePredictXGB)
RMSE_results <- RMSE_results %>%
  add_row(Model = "Extreme Gradient Boosting", RMSE_value = rmse)
RMSE_results
```

```
## # A tibble: 5 x 2
##   Model                          RMSE_value
##   <chr>                               <dbl>
## 1 lm                                90626.92
## 2 Bagged Decision Tree              31480.44
## 3 Random Forest                     27890.49
## 4 Stochastic Gradient Boosting      33955.43
## 5 Extreme Gradient Boosting         30764.41
```

15.4.2.6 Plot Extreme Gradient Boosting predictions vs Actual Resale Flat Prices

```
PlotdataXGB = as.data.frame(cbind(predicted = HDBResalePricePredictXGB,
                                  actual = tempHDBResalesvalidation$resale_price))
ggplot(PlotdataXGB,aes(predicted, actual)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method=lm) +
  ggtitle("Extreme GB: Prediction vs Actual Resale Flat Price") +
  xlab("Predicted Resale Flat Price (S$)") +
  ylab("Actual Resale Flat Price (S$)") +
  theme(plot.title = element_text(color="darkgreen",size=12,hjust = 0.5),
        axis.text.y = element_text(size=12),
        axis.text.x = element_text(size=12,hjust=.5),
        axis.title.x = element_text(size=12),
        axis.title.y = element_text(size=12))
```



16 Results Discussion

16.1 Summary of Resampling metrics for Regression Trees

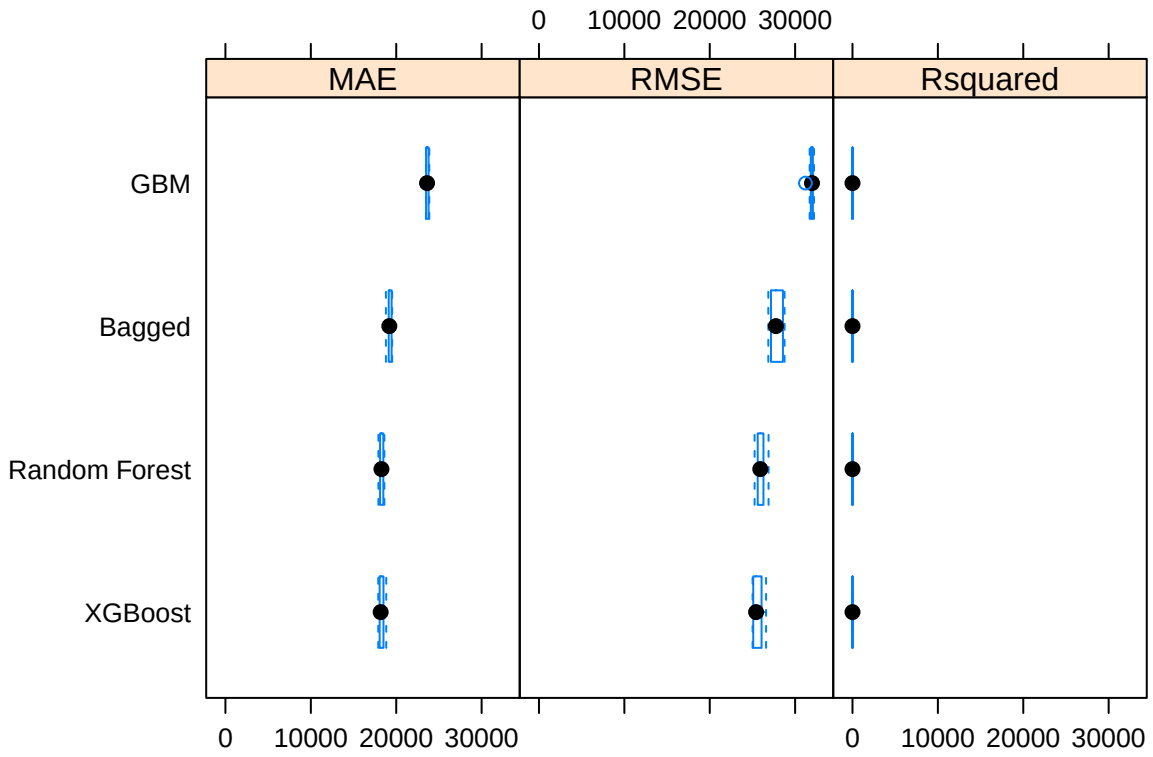
This section summarized the resampling performance on the final model produced by the train function.

```
Resamples <- resamples(list(
                            "Bagged" = HDBResalePriceModelTreeBag,
                            "Random Forest" = HDBResalePriceModelRF,
                            "GBM" = HDBResalePriceModelGBM,
                            "XGBoost" = HDBResalePriceModelXGB
                           ))
summary(Resamples)
```

```
##
## Call:
## summary.resamples(object = Resamples)
##
## Models: Bagged, Random Forest, GBM, XGBoost
## Number of resamples: 10
##
## MAE
```

```
##                   Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## Bagged        18792.73 19131.04 19192.99 19239.62 19423.64 19527.59    0
## Random Forest 17914.16 18144.83 18269.59 18279.80 18435.38 18612.12    0
## GBM           23482.52 23508.81 23611.04 23639.86 23764.40 23872.36    0
## XGBoost       17895.14 18081.01 18177.82 18261.30 18430.00 18833.50    0
##
## RMSE
##                   Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## Bagged        26858.38 27183.29 27749.43 27793.81 28457.35 28759.30    0
## Random Forest 25249.09 25631.00 25922.05 25965.82 26239.73 26885.69    0
## GBM           31207.52 31861.26 31978.45 31904.44 32063.90 32208.65    0
## XGBoost       25014.90 25155.27 25430.29 25592.03 25940.46 26598.06    0
##
## Rsquared
##                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## Bagged        0.9630836 0.9639607 0.9648608 0.9649473 0.9660788 0.9670476    0
## Random Forest 0.9676605 0.9687824 0.9696067 0.9694976 0.9701400 0.9712729    0
## GBM           0.9520926 0.9533053 0.9543369 0.9539745 0.9548257 0.9553133    0
## XGBoost       0.9683257 0.9695299 0.9705886 0.9702849 0.9710140 0.9714225    0
```

```
#Boxplot of Resampling metrics for Regression Trees
bwplot(Resamples)
```



From the resamples, based on the mean RMSE score, it is observed that XGBoost is the best performing model. It also had the lowest mean MAE score and highest mean R-squared.

16.2 Summary of RMSE results obtained from validation dataset

```
#Show the summary of RMSE value obtained from validation dataset
RMSE_results
```

```
## # A tibble: 5 x 2
##   Model                          RMSE_value
##   <chr>                               <dbl>
## 1 lm                                 90626.92
## 2 Bagged Decision Tree               31480.44
## 3 Random Forest                      27890.49
## 4 Stochastic Gradient Boosting       33955.43
## 5 Extreme Gradient Boosting          30764.41
```

16.3 Multiple Regression Model

From the summary of the RMSE table above, the multiple regression model achieved the worst RMSE value of 90,626.92.

16.4 Bagged Decision Tree Model

The RMSE value achieved by the Bagged Decision Tree algorithm is 31,480.44 This is significantly 65% improvement from the multiple regression algorithm. Comparing with the mean RMSE value obtained above from the training dataset, the difference is 3,686.63. This model is clearly overfitted.

16.5 Random Forest Model

Random Forest model achieved a RMSE value of 27,890.49. It is a slight 11% improvement from the Bagged Decision Tree model. Comparing with the mean RMSE value obtained from training dataset, the difference is 1,924.67. This model is also least overfitted among others.

16.6 Stochastic Gradient Boosting Machine Model

Using Boosting Ensemble method, the stochastic gradient boosting model achieved a RMSE value of 33,955.43 with the validation dataset. It performed worst than the Bagged Decision Tree, Random Forest and Extreme Gradient Boosting models. Comparing with the mean RMSE value obtained by the training dataset, the difference is 2,050.99. This model is overfitted too.

16.7 Extreme Gradient Boosting Model

Extreme Gradient Boosting model obtained a RMSE value of 30,726.87 with the validation dataset. It is an slight improvement of 9% compared to stochastic gradient boosting model. Comparing with the mean RMSE value obtained with the training dataset, the difference is 5,172.38. It is clearly the most overfitted model, which can be observed too from the above summary of resampling metrics it is the best performing model, i.e. it learned the training dataset to a high degree of granularity.

17 Conclusion

Being a Singaporean, owning a resale HDB flat is an important decision to make because of the huge financial commitment. What is the right price the buyer should pay? To answer that, there is a need to predict the HDB resale price for buyers with accordance to their budgets and needs based on past transaction.

From the RMSE results, multiple regression model performed the worst with a RMSE value of 90,626.92. Random Forest model has the best prediction with the lowest RMSE value of 27,890.49, It is a better model because the difference between the RMSE value of training and validation set is the narrowest among all the decision tree models. Therefore Random Forest Model is recommended as the predictive algorithm in this project for predicting the HDB Resale flat prices.

During the evaluation of the various Decision Tree algorithms, it is observed that they are prone to overfitting because these algorithms can learn the training dataset to a high degree of granularity.

To further improve the accuracy of predicting the HDB resale flat prices, additional data such as population demographic, geo-location of the flats, and its proximity to the surrounding amenities, such as MRT, bus interchange, hospital, schools, sport hubs, shopping malls etc should be gathered. For example, the enrollment in primary schools in Singapore is performed in phases based on the place of residence, with priority given to children living within 1km of the school, followed by those living between 1–2km, and thereafter all other children. There is much anecdotal evidence to suggest that parents exhibit a strong desire to enroll their children in top-performing primary schools, and the list of popular schools that tend to be oversubscribed. This desire from the parent can influence the HDB resale flat prices. In addition, the variations in demographic attributes, such as household income and racial composition, across geographical space can also influence HDB resale flat prices.

18 References

Breiman, Leo. 1996a. “Bagging Predictors.” Machine Learning 24 (2). Springer: 123–40.

Rafael A. Irizarry, 2020, Introduction to Data Science: Data Analysis and Prediction Algorithms with R, Chapman and Hall/CRC.

Bradley, Boehmke & Brandon, Greenwell, 2020, Hands-On Machine Learning with R, Chapman and Hall/CRC.

HDB Annual Report (2019/2020), https://www20.hdb.gov.sg/fi10/fi10221p.nsf/hdb/2020/index.html

HDB Resale Flat Prices, https://data.gov.sg/dataset/resale-flat-prices