Apartment price investigation

Radhika zawar S3734939

23 April 2019

Content

Chapter 1

Introduction

1.1 Objective

This document presents an anlysis of data oncerning apartments and their prices. The analysis is based on 5891 observations of apartment data along with their 30 descriptive features. All the observations are of Daebong strict, Daegu city, South Korea. The data sets were sourced from the kaggle data Repository at https://www.kaggle.com/gunhee/koreahousedata (https://www.kaggle.com/gunhee/koreahousedata) [1]

This project has two phases. Phase I focuses on data preprocessing and exploration, as covered in this report. I shall present model building in Phase II. The rest of this report is organised as follow. Section 2 describes the data sets and their attributes. Section 3 covers data pre-processing. In Section 4, we explore each attribute and their inter-relationships. The last section presents a brief summary. Compiled from R-markdown, this report contains both narratives and the R-codes used for data pre-processing and exploration.

1.2 Data-Sets

The kaggle dataset provides one .csv file of Daegu_Real_Estate_data.csv (here refered as apartment) with 5891 variables and 30 features. This dataset is about Apartment transaction data which are generated from Aug/2007 to Aug/2017 in Daebong strict, Daegu city, South Korea. In tgis analysis for this dataset the SalePrice of apartment is taken as target features and each instant has other 29 descriptive features. In Phase II, we would build the classifiers from the combined the data set and evaluate their performance using cross-validation.

1.2.1 Target feature

The response feature is SalePrice which is given in numerical format and the range varies from USD 32743 to USD 585840. However, for this analysis the target feature converted to categorical variables in data preparation phase. The target feature then has twelve classes and hence classification problem. To reiterate, The goal is to predict the salePrice category for the instance with given descriptive features.

1.2.2 Descriptive Features

The variable description is produced here from Daegu_Real_Estate_data.csv (here refered as apartment) :

- YearBuilt (Numerical) : range 1978 to 2015
- YrSold (Numerical) : range 2007 to 2017
- MonthSold (Numerical) : range 1 to 12
- Size.sqf.(Numerical) - size of apartment in square feet : range 135 to 2337
- Floor (Numerical) - what floor is property located : range 1 to 42
- HallwayType (Categorical) : terraced, corridor, mixed
- HeatingType (Categorical) : individual_heating, central_heating
- AptManageType (Categorical) - type of how apartment was managed by: management_in_trust, self_management
- N_Parkinglot.Ground. (Numerical) - count number of parking spaces on the ground : range 0 to 713
- N_Parkinglot.Basement. (Numerical) - count number of parking spaces on basement : range 0 to 1321
- TimeToBusStop (Categorical) - measure time takes from apartment to bus stop : 5min~10min, 0~5min, 10min~15min
- TimeToSubway (Categorical) - measure time takes from apartment to subway station : 10min~15min, 5min~10min, 0-5min, 15min~20min, no_bus_stop_nearby
- N_APT (Numerical) - number of apartment building in a apartment complex : range 1 to 13
- N_manager (Numerical) - number of people manage apartment facilities (eg. security, cleaner etc) : range 1 to 14
- N_elevators (Numerical) - total number of elevators in an apartment complex : range 0 to 27
- SubwayStation (Categorical) - name of subway station nearby apartment : Kyungbuk_uni_hospital, Daegu, Sin-nam, Myung-duk, Chil-sung-market, Bangoge, Banwoldang, no_subway_nearby
- N_FacilitiesNearBy.PublicOffice. (Numerical) - number of public offices nearby apartment : range 0 to 7
- N_FacilitiesNearBy.Hospital. (Numerical) - number of hospitals nearby apartment : range 0 to 2
- N_FacilitiesNearBy.Dpartmentstore. (Numerical) - number of department stores nearby apartment : range 0 to 2
- N_FacilitiesNearBy.Mall. (Numerical) - number of malls nearby apartment : range 0 to 2
- N_FacilitiesNearBy.ETC. (Numerical) - like hotels, special school : range 0 to 5
- N_FacilitiesNearBy.Park. (Numerical) - number of parks nearby apartment : range 0 to 2
- N_SchoolNearBy.Elementary. (Numerical) - number of elementary schools nearby apartment : range 0 to 6
- N_SchoolNearBy.Middle. (Numerical) - number of middle schools nearby apartment : range 0 to 6
- N_SchoolNearBy.High. (Numerical) - number of high schools nearby apartment : range 0 to 5
- N_SchoolNearBy.University. (Numerical) - number of universities nearby apartment : range 0 to 5
- N_FacilitiesInApt (Numerical) - number of facilities for residents like swimming pool, gym, play ground : range 1 to 10
- N_FacilitiesNearBy.Total. (Numerical) - total number of facilities nearby apartment : range 0 to 16
- N_SchoolNearBy.Total. (Numerical) - total number of schools nearby apartment : range 0 to 17

Most of the descriptive features are self-explanatory, but those which are not, for them explanation has been added. [1]

Chapter 2

Data Pro-processsing

2.1 Preliminaries

The dataset Daegu_Real_Estate_data.csv[1] is loaded in R memory by the name apartment from the local system. perfomred basic investigation of data.

2.2 Data Cleaning and Transformation

Firstly, we confirmed that the feature types matched the description as outlined in the documentation, presented in section 1.2.2

```
dim(apartment) #diamensions
```

```
## [1] 5891   30
```

```
str(apartment) #format of variables
```

```
## 'data.frame':    5891 obs. of  30 variables:
##  $ SalePrice                          : int  141592 51327 48672 380530 221238 35840 783
18 61946 84070 83185 ...
##  $ YearBuilt                          : int  2006 1985 1985 2006 1993 1992 1992 1993 19
93 1992 ...
##  $ YrSold                             : int  2007 2007 2007 2007 2007 2007 2007 2007 20
07 2007 ...
##  $ MonthSold                          : int  8 8 8 8 8 8 8 8 8 ...
##  $ Size.sqf.                          : int  814 587 587 2056 1761 355 644 644 644 64
4 ...
##  $ Floor                              : int  3 8 6 8 3 5 2 10 3 13 ...
##  $ HallwayType                        : Factor w/ 3 levels "corridor","mixed",..: 3 1
1 3 2 1 2 2 2 2 ...
##  $ HeatingType                        : Factor w/ 2 levels "central_heating",..: 2 2 2
2 2 2 2 2 2 2 ...
##  $ AptManageType                      : Factor w/ 2 levels "management_in_trust",..: 1
2 2 1 1 1 2 1 1 2 ...
##  $ N_Parkinglot.Ground.               : num  111 80 80 249 523 200 142 523 523 142 ...
##  $ N_Parkinglot.Basement.             : num  184 76 76 536 536 0 79 536 536 79 ...
##  $ TimeToBusStop                      : Factor w/ 3 levels "0~5min","10min~15min",..:
3 1 1 1 1 3 3 1 1 3 ...
##  $ TimeToSubway                       : Factor w/ 5 levels "0-5min","10min~15min",..:
2 4 4 1 3 2 3 3 3 3 ...
##  $ N_APT                              : num  3 1 1 6 8 3 3 8 8 3 ...
##  $ N_manager                          : num  3 2 2 5 8 5 4 8 8 4 ...
##  $ N_elevators                        : num  0 2 2 11 20 10 8 20 20 8 ...
##  $ SubwayStation                      : Factor w/ 8 levels "Bangoge","Banwoldang",..:
5 4 4 8 6 6 6 6 6 6 ...
##  $ N_FacilitiesNearBy.PublicOffice.   : num  2 5 5 1 6 7 5 6 6 5 ...
##  $ N_FacilitiesNearBy.Hospital.       : int  1 1 1 1 2 1 1 2 2 1 ...
##  $ N_FacilitiesNearBy.Dpartmentstore.: num  1 2 2 0 0 1 1 0 0 1 ...
##  $ N_FacilitiesNearBy.Mall.           : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ N_FacilitiesNearBy.ETC.            : num  1 2 2 0 5 5 1 5 5 1 ...
##  $ N_FacilitiesNearBy.Park.           : num  0 1 1 0 0 1 0 0 0 0 ...
##  $ N_SchoolNearBy.Elementary.         : num  3 2 2 2 4 4 3 4 4 3 ...
##  $ N_SchoolNearBy.Middle.             : num  2 1 1 2 3 3 3 3 3 3 ...
##  $ N_SchoolNearBy.High.               : num  2 1 1 1 5 5 4 5 5 4 ...
##  $ N_SchoolNearBy.University.         : num  2 0 0 2 5 5 4 5 5 4 ...
##  $ N_FacilitiesInApt                  : int  5 3 3 5 4 3 3 4 4 3 ...
##  $ N_FacilitiesNearBy.Total.          : num  6 12 12 3 14 16 9 14 14 9 ...
##  $ N_SchoolNearBy.Total.              : num  9 4 4 7 17 17 14 17 17 14 ...
```

```
colnames(apartment) #column names
```

```
##  [1] "SalePrice"
##  [2] "YearBuilt"
##  [3] "YrSold"
##  [4] "MonthSold"
##  [5] "Size.sqf."
##  [6] "Floor"
##  [7] "HallwayType"
##  [8] "HeatingType"
##  [9] "AptManageType"
## [10] "N_Parkinglot.Ground."
## [11] "N_Parkinglot.Basement."
## [12] "TimeToBusStop"
## [13] "TimeToSubway"
## [14] "N_APT"
## [15] "N_manager"
## [16] "N_elevators"
## [17] "SubwayStation"
## [18] "N_FacilitiesNearBy.PublicOffice."
## [19] "N_FacilitiesNearBy.Hospital."
## [20] "N_FacilitiesNearBy.Dpartmentstore."
## [21] "N_FacilitiesNearBy.Mall."
## [22] "N_FacilitiesNearBy.ETC."
## [23] "N_FacilitiesNearBy.Park."
## [24] "N_SchoolNearBy.Elementary."
## [25] "N_SchoolNearBy.Middle."
## [26] "N_SchoolNearBy.High."
## [27] "N_SchoolNearBy.University."
## [28] "N_FacilitiesInApt"
## [29] "N_FacilitiesNearBy.Total."
## [30] "N_SchoolNearBy.Total."
```

```
apartment %>% head(5) #first 5 observation
```

	SalePrice <int>	YearBuilt <int>	YrS… <int>	MonthS… <int>	Size.sqf. <int>	Floor <int>	HallwayType <fctr>	HeatingType <fctr>
1	141592	2006	2007	8	814	3	terraced	individual_heating
2	51327	1985	2007	8	587	8	corridor	individual_heating
3	48672	1985	2007	8	587	6	corridor	individual_heating
4	380530	2006	2007	8	2056	8	terraced	individual_heating
5	221238	1993	2007	8	1761	3	mixed	individual_heating

5 rows | 1-9 of 31 columns

Dimension of the data set is (5891, 30) and all the column names and datatypes matches to documented

datatypes.

Exploring Summary and descriptive statistics:

```
table(apartment[,c(7,8,9,12,13,17)] %>% summary()) #summary statistics of categorical dat
a
```

```
##
## (Other)              : 200      0-5min            :2759
##                           1                             1
##           0~5min     :4509      10min~15min       : 806
##                           1                             1
##           10min~15min:  55      15min~20min       : 953
##                           1                             1
##    5min~10min        :1135             5min~10min :1327
##                           1                             1
## Bangoge              : 737   Banwoldang           : 748
##                           1                             1
##    central_heating   : 300               corridor: 637
##                           1                             1
##    individual_heating:5591   Kyungbuk_uni_hospital:1644
##                           1                             1
##   management_in_trust:5542               mixed   :1690
##                           1                             1
## Myung-duk            :1507      no_bus_stop_nearby: 238
##                           1                             1
## no_subway_nearby     : 404     self_management    : 349
##                           1                             1
## Sin-nam              : 651               terraced:3564
##                           1                             1
```

```
describe(apartment[,-c(7,8,9,12,13,17)]) #Descriptive statistics for numerical data
```

	v...	n	mean	sd	med..
	<int>	<dbl>	<dbl>	<dbl>	<dbl:
SalePrice	1	5891	2.212181e+05	1.063842e+05	20796
YearBuilt	2	5891	2.002967e+03	8.811782e+00	200(
YrSold	3	5891	2.012692e+03	2.905488e+00	201:
MonthSold	4	5891	6.160244e+00	3.387752e+00	(
Size.sqf.	5	5891	9.555692e+02	3.824641e+02	91(
Floor	6	5891	1.202631e+01	7.548743e+00	1
N_Parkinglot.Ground.	7	5891	1.958836e+02	2.185972e+02	10(
N_Parkinglot.Basement	8	5891	5.707617e+02	4.086211e+02	53(

	v…	n	mean	sd	med…
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
N_APT	9	5891	5.613648e+00	2.811831e+00	7
N_manager	10	5891	6.310304e+00	3.174088e+00	6

1-10 of 24 rows | 1-7 of 14 columns Previous **1** 2 3 Next

Varifying specification of each feature :

```
range(apartment$SalePrice)
```

```
## [1]  32743 585840
```

```
unique(apartment$YearBuilt)
```

```
##  [1] 2006 1985 1993 1992 1986 2007 1997 2005 2003 1978 2009 2008 1980 2013
## [15] 2014 2015
```

```
unique(apartment$YrSold)
```

```
##  [1] 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017
```

```
unique(apartment$MonthSold)
```

```
##  [1]  8  9 10 11 12  1  2  3  4  5  6  7
```

```
range(apartment$Size.sqf.)
```

```
## [1]  135 2337
```

```
unique(apartment$Floor)
```

```
##  [1]  3  8  6  5  2 10 13  4 11 18  7 24  1 20 12 39 38 15  9 17 16 14 42
## [24] 28 22 25 19 21 30 26 41 27 29 31 23 43 35 33 34 37 40 36 32
```

```
unique(apartment$HallwayType)
```

```
## [1] terraced corridor mixed
## Levels: corridor mixed terraced
```

```
unique(apartment$HeatingType)
```

```
## [1] individual_heating central_heating
## Levels: central_heating individual_heating
```

```
unique(apartment$AptManageType)
```

```
## [1] management_in_trust self_management
## Levels: management_in_trust self_management
```

```
range(apartment$N_Parkinglot.Ground.)
```

```
## [1]   0 713
```

```
range(apartment$N_Parkinglot.Basement.)
```

```
## [1]    0 1321
```

```
unique(apartment$TimeToBusStop)
```

```
## [1] 5min~10min  0~5min      10min~15min
## Levels: 0~5min 10min~15min 5min~10min
```

```
unique(apartment$TimeToSubway)
```

```
## [1] 10min~15min        5min~10min         0-5min
## [4] 15min~20min        no_bus_stop_nearby
## 5 Levels: 0-5min 10min~15min 15min~20min ... no_bus_stop_nearby
```

```
range(apartment$N_APT)
```

```
## [1]  1 13
```

```
range(apartment$N_manager)
```

```
## [1]  1 14
```

```
range(apartment$N_elevators)
```

```
## [1]  0 27
```

```
unique(apartment$SubwayStation)
```

```
## [1] Kyungbuk_uni_hospital Daegu                 Sin-nam
## [4] Myung-duk             Chil-sung-market      Bangoge
## [7] Banwoldang            no_subway_nearby
## 8 Levels: Bangoge Banwoldang Chil-sung-market ... Sin-nam
```

```
range(apartment$N_FacilitiesNearBy.PublicOffice.)
```

```
## [1] 0 7
```

```
range(apartment$N_FacilitiesNearBy.Hospital.)
```

```
## [1] 0 2
```

```
range(apartment$N_FacilitiesNearBy.Dpartmentstore.)
```

```
## [1] 0 2
```

```
range(apartment$N_FacilitiesNearBy.Mall.)
```

```
## [1] 0 2
```

```
range(apartment$N_FacilitiesNearBy.ETC.)
```

```
## [1] 0 5
```

```
range(apartment$N_FacilitiesNearBy.Park.)
```

```
## [1] 0 2
```

```
range(apartment$N_SchoolNearBy.Elementary.)
```

```
## [1] 0 6
```

```
range(apartment$N_SchoolNearBy.Middle.)
```

```
## [1] 0 4
```

```
range(apartment$N_SchoolNearBy.High.)
```

```
## [1] 0 5
```

```
range(apartment$N_SchoolNearBy.University.)
```

```
## [1] 0 5
```

```
range(apartment$N_FacilitiesInApt)
```

```
## [1]  1 10
```

```
range(apartment$N_FacilitiesNearBy.Total.)
```

```
## [1]  0 16
```

```
range(apartment$N_SchoolNearBy.Total.)
```

```
## [1]  0 17
```

On surface, no attributes contain NaN values (though the missing values might be coded with different labels) as shown in the code chunk.

Investing for missing values in following code chunk:

```
colSums(is.na(apartment))
```

```
##                          SalePrice                          YearBuilt
##                                  0                                  0
##                             YrSold                          MonthSold
##                                  0                                  0
##                          Size.sqf.                              Floor
##                                  0                                  0
##                        HallwayType                        HeatingType
##                                  0                                  0
##                      AptManageType               N_Parkinglot.Ground.
##                                  0                                  0
##             N_Parkinglot.Basement.                      TimeToBusStop
##                                  0                                  0
##                       TimeToSubway                              N_APT
##                                  0                                  0
##                          N_manager                         N_elevators
##                                  0                                  0
##                      SubwayStation   N_FacilitiesNearBy.PublicOffice.
##                                  0                                  0
##       N_FacilitiesNearBy.Hospital. N_FacilitiesNearBy.Dpartmentstore.
##                                  0                                  0
##           N_FacilitiesNearBy.Mall.            N_FacilitiesNearBy.ETC.
##                                  0                                  0
##           N_FacilitiesNearBy.Park.         N_SchoolNearBy.Elementary.
##                                  0                                  0
##             N_SchoolNearBy.Middle.               N_SchoolNearBy.High.
##                                  0                                  0
##         N_SchoolNearBy.University.                  N_FacilitiesInApt
##                                  0                                  0
##          N_FacilitiesNearBy.Total.              N_SchoolNearBy.Total.
##                                  0                                  0
```

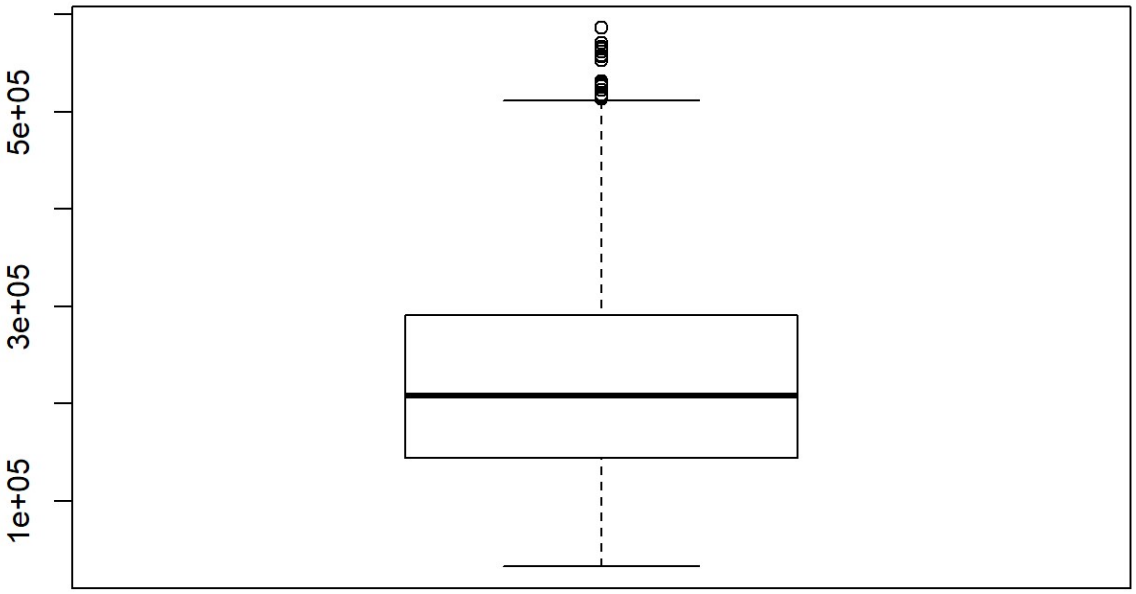
No missing value in any variable.

Data Transformation and outlier detection:

```
#Target Variable : SalePrice
summary(apartment$SalePrice)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   32743  144247  207964  221218  291150  585840
```

```
summary(boxplot(apartment$SalePrice))
```




```
##       Length Class   Mode
## stats  5     integer numeric
## n      1     -none-  numeric
## conf   2     -none-  numeric
## out   34     -none-  numeric
## group 34     -none-  numeric
## names  1     -none-  character
```

```
library(outliers)
```

```
##
## Attaching package: 'outliers'
```

```
## The following object is masked from 'package:psych':
##
##     outlier
```

```
#Investigating outliers
z.scores <- apartment$SalePrice %>%  scores(type = "z")
z.scores %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.7716 -0.7235 -0.1246  0.0000  0.6574  3.4274
```

```
length(which( abs(z.scores) >3 ))
```

```
## [1] 10
```

```
print(which( abs(z.scores) >3 ))
```

```
##  [1]  470 4328 4390 4423 4495 5314 5559 5562 5882 5883
```

```
apartment[c( 470, 4328, 4390, 4423, 4495, 5314, 5559, 5562, 5882, 5883),]
```

	SalePrice	YearBuilt	YrS…	MonthS…	Size.sqf.	Floor	HallwayType	HeatingType
	<int>	<int>	<int>	<int>	<int>	<int>	<fctr>	<fctr>
470	556637	2007	2008	10	1928	31	terraced	individual_hea
4328	570796	2007	2015	6	1928	31	terraced	individual_hea
4390	566371	2007	2015	7	1928	16	terraced	individual_hea
4423	553097	2007	2015	8	1928	21	terraced	individual_hea
4495	561946	2007	2015	10	1928	26	terraced	individual_hea
5314	564601	2007	2017	1	1928	33	terraced	individual_hea
5559	566371	2007	2017	5	1643	37	terraced	individual_hea
5562	585840	2007	2017	5	1928	31	terraced	individual_hea
5882	557522	2007	2017	8	1928	20	terraced	individual_hea
5883	570796	2007	2017	8	1928	24	terraced	individual_hea

1-10 of 10 rows | 1-9 of 31 columns

```
#By analysing the possible outlier instances, it seems that this are not real outliers an
d the data is not random and has strong pattern

apartment$SalePrice <- cut(apartment$SalePrice, c(0,50000,100000,150000,200000,250000,300
000,350000,400000,450000,500000,550000,600000),
      labels = c(">50k","50k-100k","100k-150k","150k-200k","200k-250k","250k-300k","300k-
350k","350k-400k","400k-450k","450k-500k","500k-550k","550k-600k"))

apartment$SalePrice <- as.ordered(apartment$SalePrice)
```

Target variable SalePrice is converted to categorical from numerical to run classification algorithm in phase II of the project. The categorical variable Saleprice has 12 levels as : >50k, 50k-100k ,100k-150k, 150k-200k, 200k-250k, 250k-300k, 300k-350k, 350k-400k, 400k-450k, 450k-500k, 500k-550k, 550k-600k.

In the following piece of code a new variable age of the property has been created using existing variables YrSold and YearBuild, so that we can use AgeWhenSold as new descriptive feature for apartment as, in general, newer apartments are costlier than older ones.

```
unique(apartment$YearBuilt)
```

```
##  [1] 2006 1985 1993 1992 1986 2007 1997 2005 2003 1978 2009 2008 1980 2013
## [15] 2014 2015
```

```
unique(apartment$YrSold)
```

```
##  [1] 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017
```

```
#creating new variable AgeWhenSold which is nothing but age of the property on selling
apartment <- apartment %>% mutate(AgeWhenSold = YrSold - YearBuilt)
colnames(apartment)
```

```
##  [1] "SalePrice"
##  [2] "YearBuilt"
##  [3] "YrSold"
##  [4] "MonthSold"
##  [5] "Size.sqf."
##  [6] "Floor"
##  [7] "HallwayType"
##  [8] "HeatingType"
##  [9] "AptManageType"
## [10] "N_Parkinglot.Ground."
## [11] "N_Parkinglot.Basement."
## [12] "TimeToBusStop"
## [13] "TimeToSubway"
## [14] "N_APT"
## [15] "N_manager"
## [16] "N_elevators"
## [17] "SubwayStation"
## [18] "N_FacilitiesNearBy.PublicOffice."
## [19] "N_FacilitiesNearBy.Hospital."
## [20] "N_FacilitiesNearBy.Dpartmentstore."
## [21] "N_FacilitiesNearBy.Mall."
## [22] "N_FacilitiesNearBy.ETC."
## [23] "N_FacilitiesNearBy.Park."
## [24] "N_SchoolNearBy.Elementary."
## [25] "N_SchoolNearBy.Middle."
## [26] "N_SchoolNearBy.High."
## [27] "N_SchoolNearBy.University."
## [28] "N_FacilitiesInApt"
## [29] "N_FacilitiesNearBy.Total."
## [30] "N_SchoolNearBy.Total."
## [31] "AgeWhenSold"
```

```
str(apartment$AgeWhenSold)
```

```
##  int [1:5891] 1 22 22 1 14 15 15 14 14 15 ...
```

```
unique(apartment$AgeWhenSold)
```

```
##  [1]  1 22 14 15 21  0 10  2 11 23 16  3  5 30  4 17 31 12 24  6 29 25 18
## [24] 13 32 19 26  8 33 20  7 34 27  9 28 35 36 37 38 39
```

In the following piece of code various numerical variables are converted into categical variables as they will be treated as categories and if needed ordinal for the relevance of this project:

```
##-- YearBuilt --## column 2

apartment$YearBuilt <- cut(apartment$YearBuilt, c(1977,1985,1990,1995,2000,2005,2010,201
5))
str(apartment$YearBuilt)
```

```
##  Factor w/ 7 levels "(1977,1985]",..: 6 1 1 6 3 3 3 3 3 3 ...
```

```
##-- YrSold --## column 3

apartment$YrSold <- as.factor(apartment$YrSold)
str(apartment$YrSold)
```

```
##  Factor w/ 11 levels "2007","2008",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
##-- MonthSold --## column 4

apartment$MonthSold <- apartment$MonthSold %>% factor(c(1,2,3,4,5,6,7,8,9,10,11,12), leve
ls = c(1,2,3,4,5,6,7,8,9,10,11,12), ordered = TRUE)
str(apartment$MonthSold)
```

```
##  Ord.factor w/ 12 levels "1"<"2"<"3"<"4"<..: 8 8 8 8 8 8 8 8 8 8 ...
```

```
##-- Floor --## column 6

apartment$Floor <- as.factor(apartment$Floor)
str(apartment$Floor)
```

```
##  Factor w/ 43 levels "1","2","3","4",..: 3 8 6 8 3 5 2 10 3 13 ...
```

```
##-- AptManageType --## column 9

apartment$AptManageType <- apartment$AptManageType %>%  factor(c('self_management','manag
ement_in_trust'), levels = c('self_management','management_in_trust'), ordered = FALSE)
str(apartment$AptManageType)
```

```
##  Factor w/ 2 levels "self_management",..: 2 1 1 2 2 2 1 2 2 1 ...
```

```
##-- TimeToBusStop --## column 12

apartment$TimeToBusStop <- apartment$TimeToBusStop %>% factor(c("10min~15min","5min~10mi
n","0~5min"), levels = c("10min~15min","5min~10min","0~5min"),ordered = TRUE)
str(apartment$TimeToBusStop)
```

```
##  Ord.factor w/ 3 levels "10min~15min"<..: 2 3 3 3 3 2 2 3 3 2 ...
```

```
##-- TimeToSubway --## column 13

apartment$TimeToSubway <- apartment$TimeToSubway %>% factor(c("no_bus_stop_nearby","15min
~20min","10min~15min","5min~10min","0-5min"), levels = c("no_bus_stop_nearby","15min~20mi
n","10min~15min","5min~10min","0-5min"),ordered = TRUE)
str(apartment$TimeToSubway)
```

```
##  Ord.factor w/ 5 levels "no_bus_stop_nearby"<..: 3 4 4 5 2 3 2 2 2 2 ...
```

Chapter 3

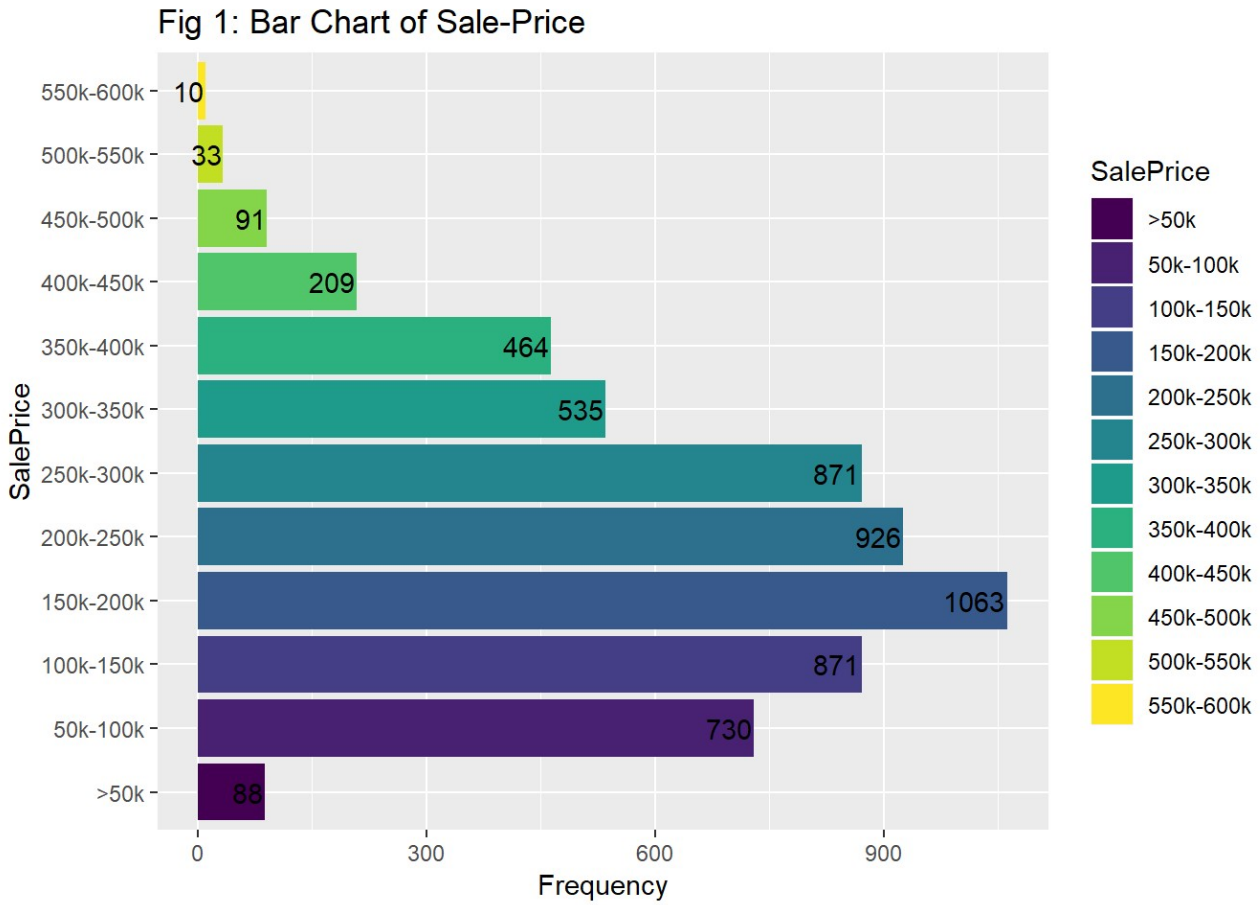
Data Exploration

3.1 Univariate Visualisation

In following code chunks all the 31 varables have been plotted into graph and closely observed for their distribution and value counts.

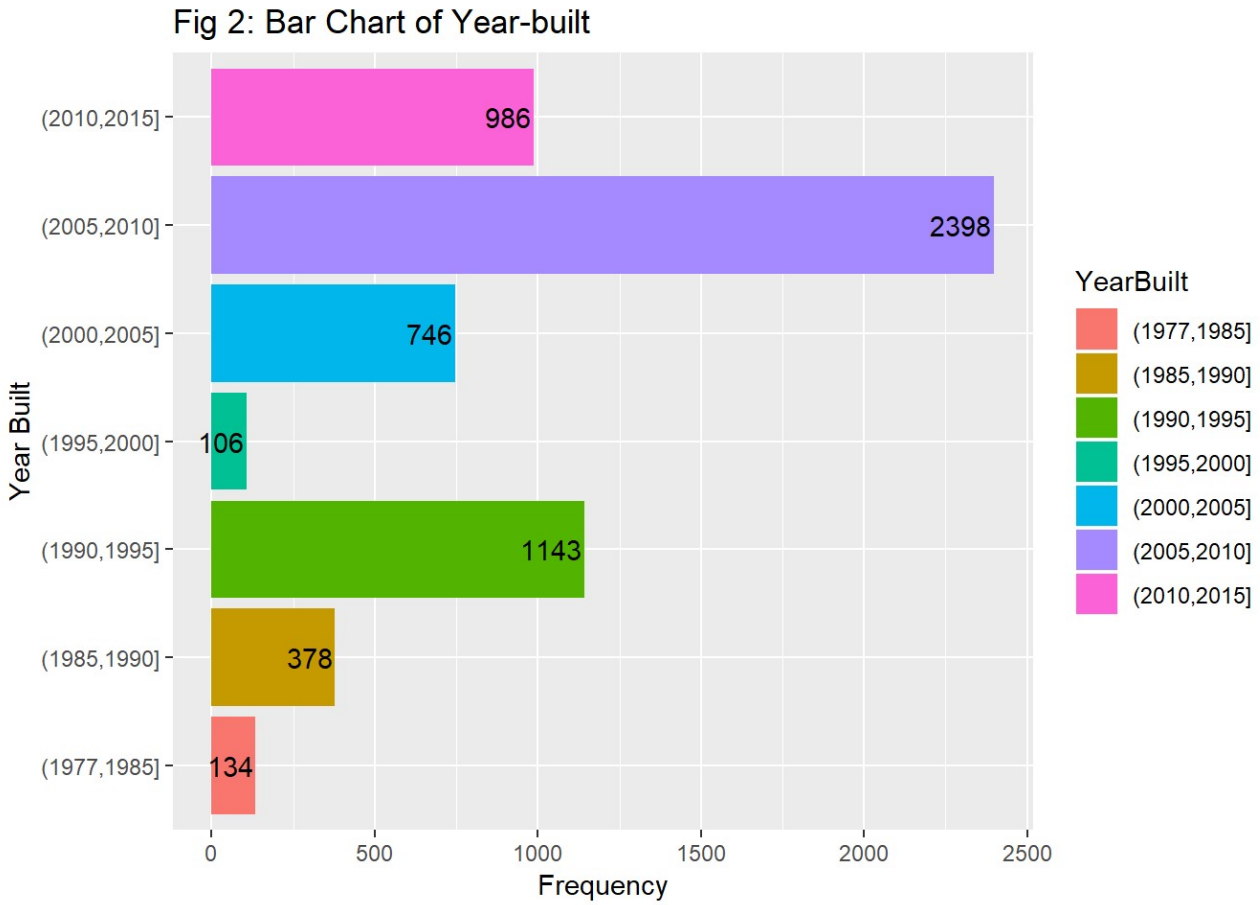
For following analysis ‘baser’, ‘ggplot2’, ‘psych’ and ‘plotly’ are mejorly used. Specific functions are not designed for the plots as we intent to customise plots as per our need of visualition and presentation.

```
##-- salePrice --## column 1  Target Variable

#https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf [2]
ggplot(apartment, aes(SalePrice)) + geom_bar(aes(fill = SalePrice)) +
  coord_flip()+
  geom_text(stat='count', aes(label=..count..), hjust=1.05) +
  scale_y_continuous(name="Frequency") +
  scale_x_discrete(name="SalePrice")+theme(legend.position = "right") +
  ggtitle("Fig 1: Bar Chart of Sale-Price")
```



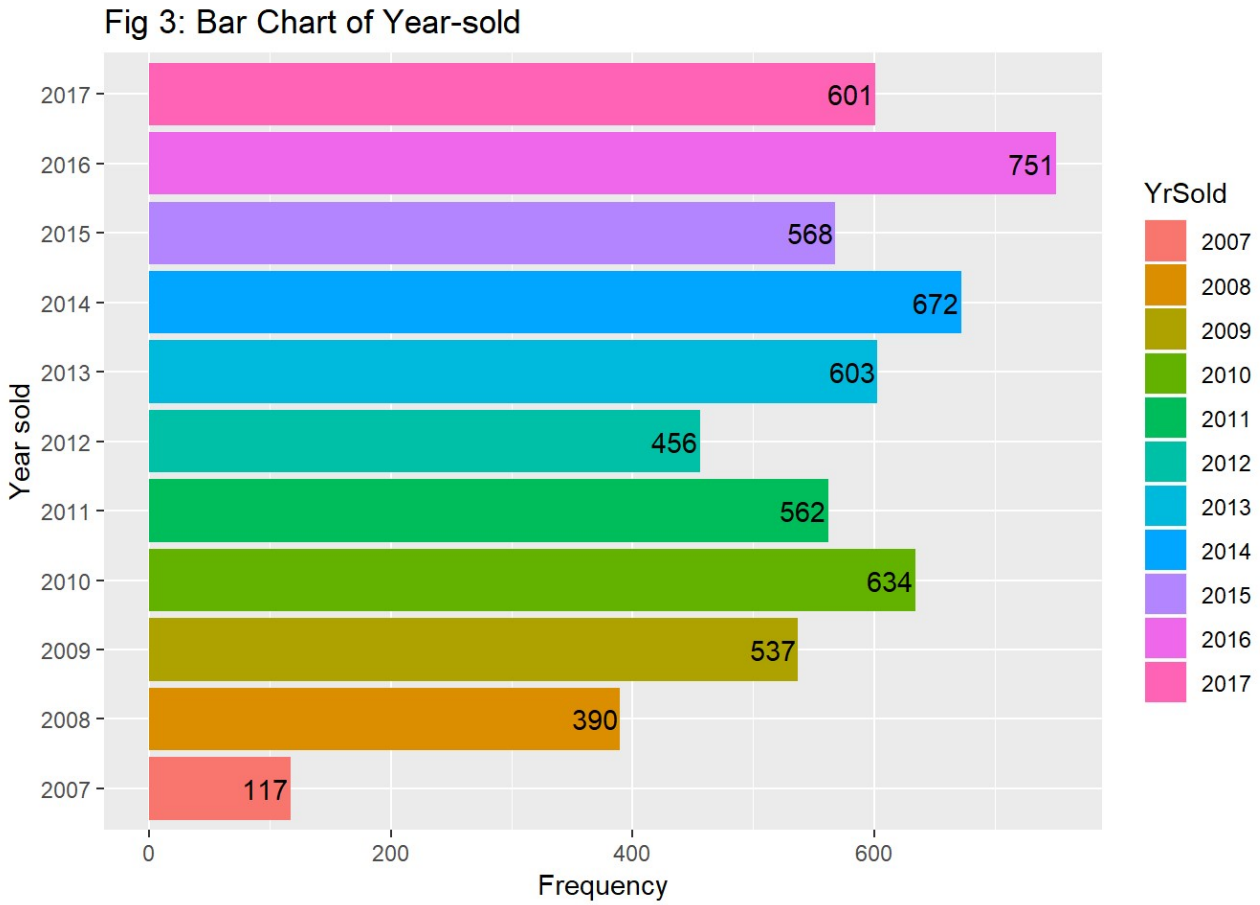
From fig. 1 it clear that most of the apartments were sold in less than USD 300K and relatively very few were apartments were sold for more USD 550k. Therefore it is safe to say that the distribution on saleprice is right skewed.

```
##-- YearBuilt --## column 2

ggplot(apartment, aes(YearBuilt)) + geom_bar(aes(fill = YearBuilt)) + coord_flip()+
  geom_text(stat='count', aes(label=..count..), hjust=1.05) +
  scale_y_continuous(name="Frequency") +
  scale_x_discrete(name="Year Built")+theme(legend.position = "right") +
  ggtitle("Fig 2: Bar Chart of Year-built")
```



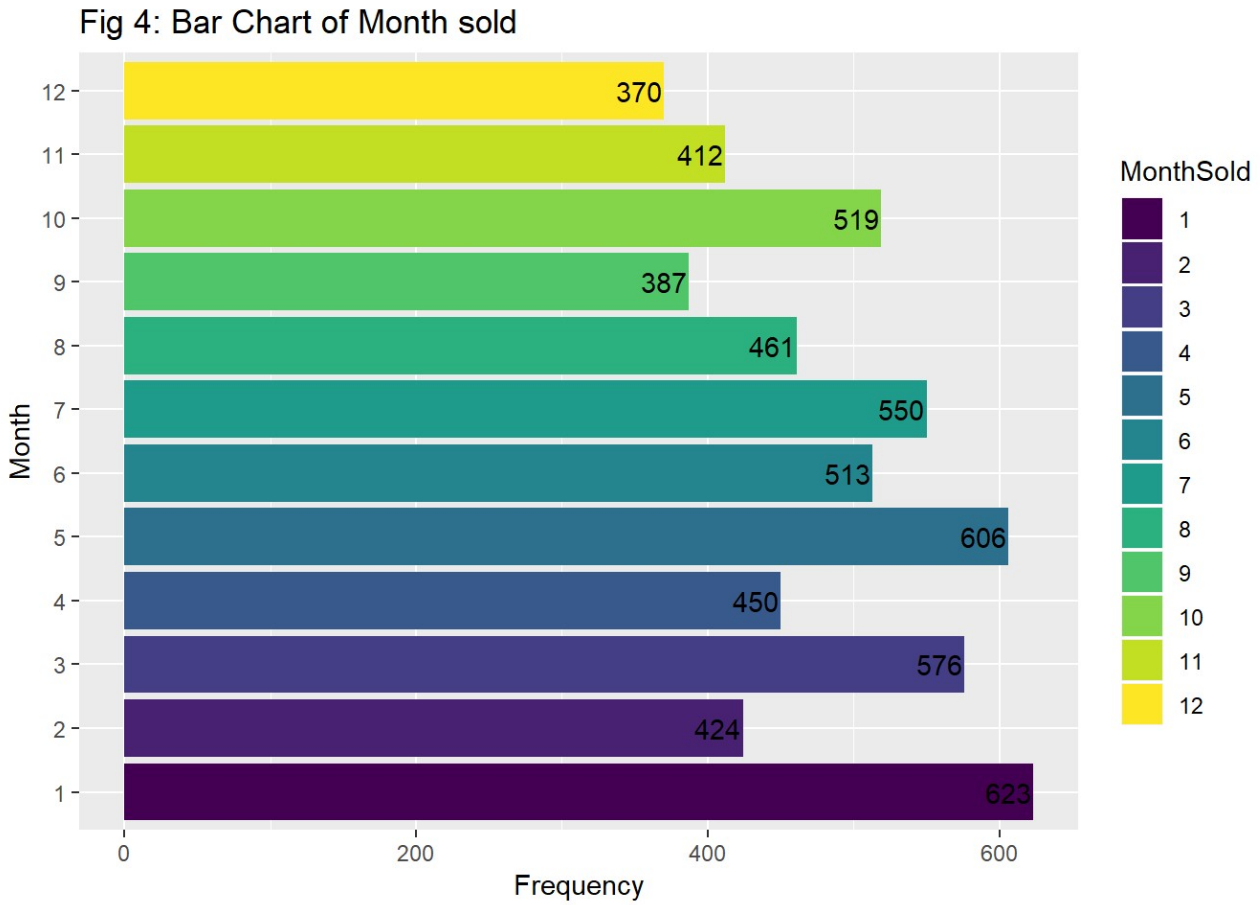
From fig.2 it clear that most of the apartments are built during 1990 to 1995 and 2000 and after. Overall, still there is no strong pattern.

```
##-- YrSold --## column 3

#https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf
ggplot(apartment, aes(YrSold)) + geom_bar(aes(fill = YrSold)) + coord_flip()+
  geom_text(stat='count', aes(label=..count..), hjust=1.05) +
  scale_y_continuous(name="Frequency") +
  scale_x_discrete(name="Year sold")+theme(legend.position = "right") +
  ggtitle("Fig 3: Bar Chart of Year-sold")
```



From fig 3 it comes to notice that there has been fluctuations in number of apartment property sold each year.

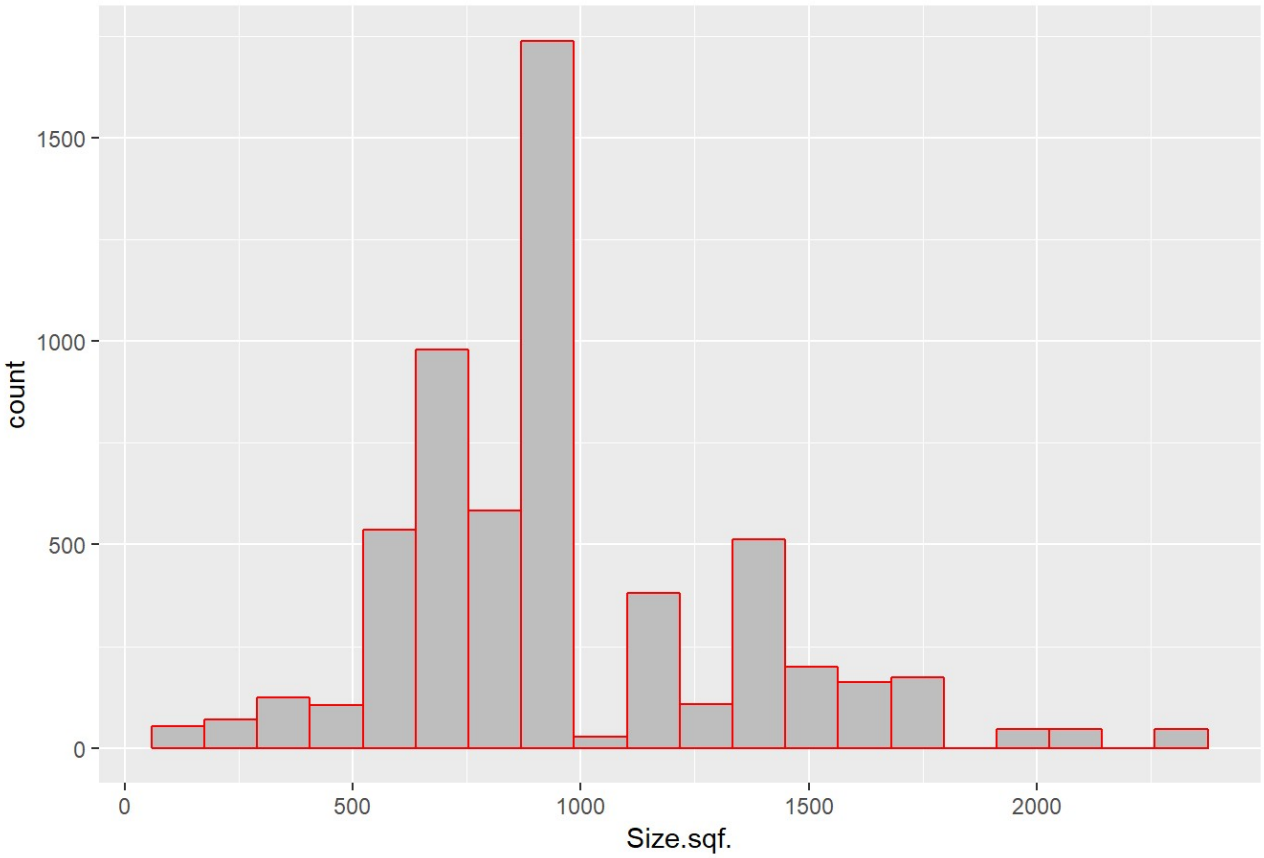
```
##-- MonthSold --## column 4

ggplot(apartment, aes(MonthSold)) + geom_bar(aes(fill = MonthSold)) + coord_flip()+
  geom_text(stat='count', aes(label=..count..), hjust=1.05) +
  scale_y_continuous(name="Frequency") +
  scale_x_discrete(name="Month")+theme(legend.position = "right") +
  ggtitle("Fig 4: Bar Chart of Month sold")
```



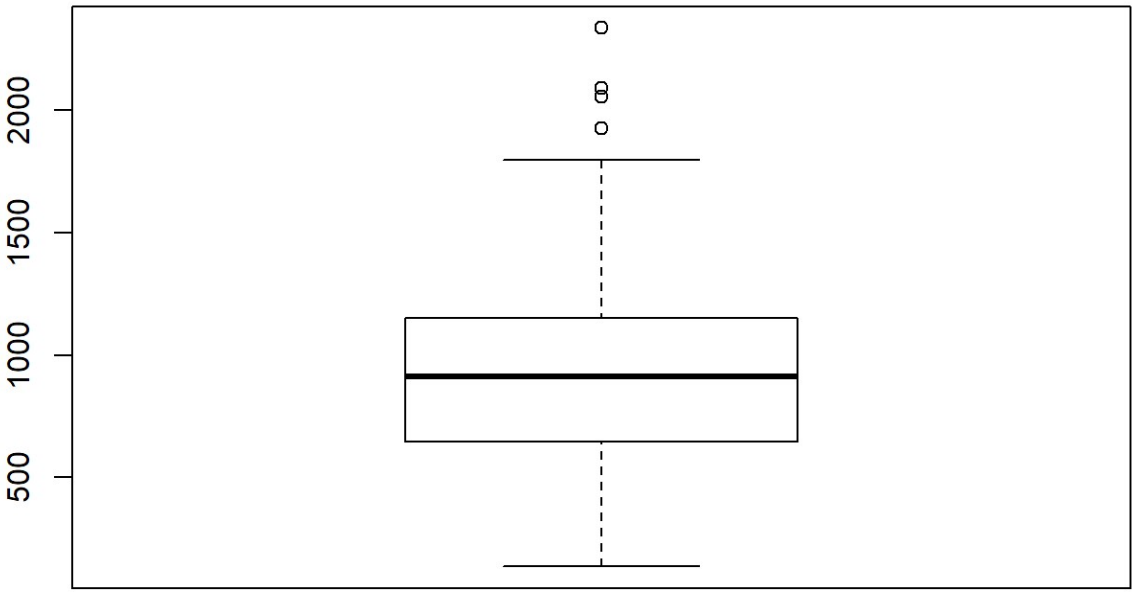
From fig 4 it comes to highlight that the month January has hightest sales of apartments and least sales are noticable in December.

```
##-- Size.sqf. --## column 5

ggplot(apartment,aes(Size.sqf.)) +
  geom_histogram(bins = 20, color = "red",fill="grey") +
  ggtitle("Fig 5: Histogram of Size of apartment in squere feet")
```

Fig 5: Histogram of Size of apartment in squere feet




```
OutVals <- boxplot(apartment$Size.sqf.)$out
```



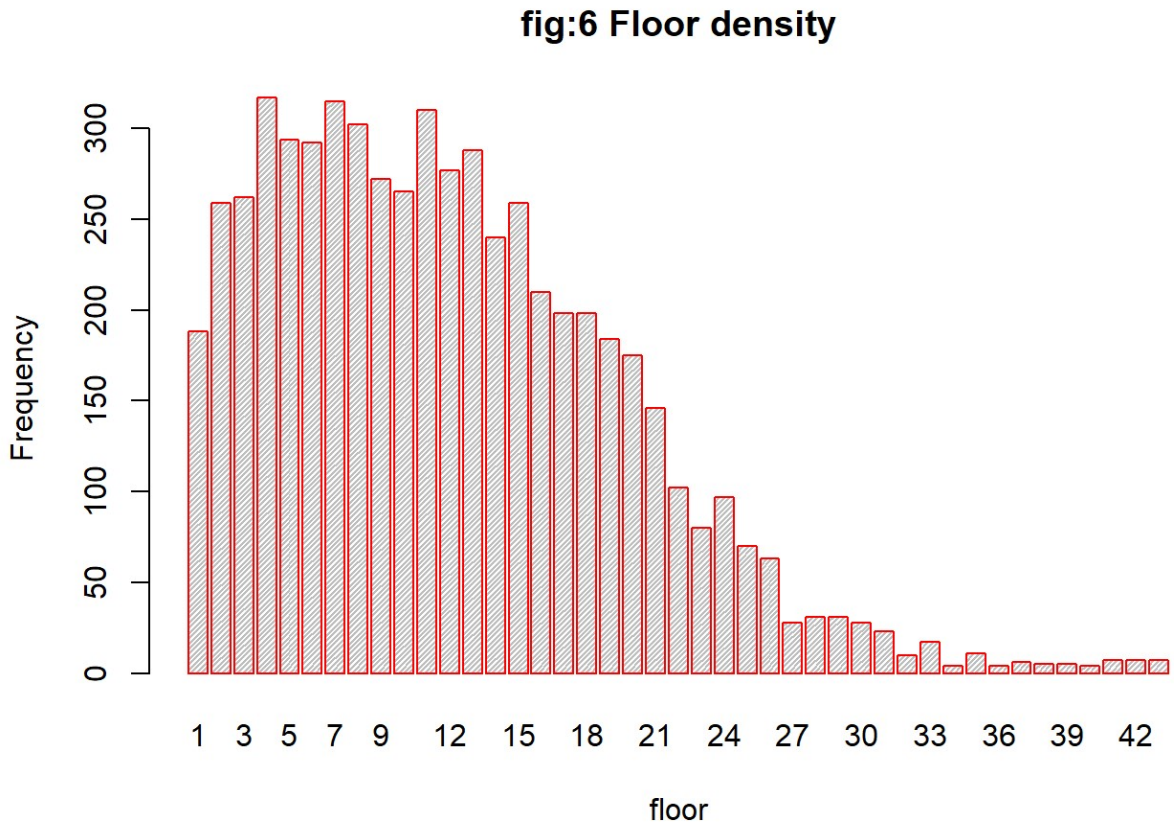

```
apartment[which(apartment$Size.sqf. %in% OutVals),] %>% head(10)
```

	SalePrice <ord>	YearBuilt <fctr>	YrS… <fctr>	MonthS… <ord>	Size.sqf. <int>	Floor <fctr>	HallwayType <fctr>	HeatingTy <fctr>
4	350k-400k	(2005,2010]	2007	8	2056	8	terraced	individual_
34	250k-300k	(2005,2010]	2007	10	2056	2	terraced	individual_
85	150k-200k	(1990,1995]	2007	11	2337	18	mixed	individual_
162	200k-250k	(1990,1995]	2008	2	2337	14	mixed	individual_
254	200k-250k	(1990,1995]	2008	4	2337	9	mixed	individual_
279	150k-200k	(1990,1995]	2008	5	2337	24	mixed	individual_
298	300k-350k	(2005,2010]	2008	6	2056	15	terraced	individual_
334	350k-400k	(2005,2010]	2008	7	2056	17	terraced	individual_
335	350k-400k	(2005,2010]	2008	7	2056	26	terraced	individual_
362	350k-400k	(2005,2010]	2008	8	2056	11	terraced	individual_

1-10 of 10 rows | 1-9 of 32 columns

From fig 5 it is quite evident that most properties has lower squere area and very few have more that 1800 sqft area.

```
##-- Floor --## column 6

barplot(table(apartment$Floor),
main="fig:6 Floor density",
xlab="floor",
ylab="Frequency",
border="red",
col="grey",
density=50)
```



From fig 6 it is clear that not many apartment sold were on the highter floors and this distribution is right skewed.

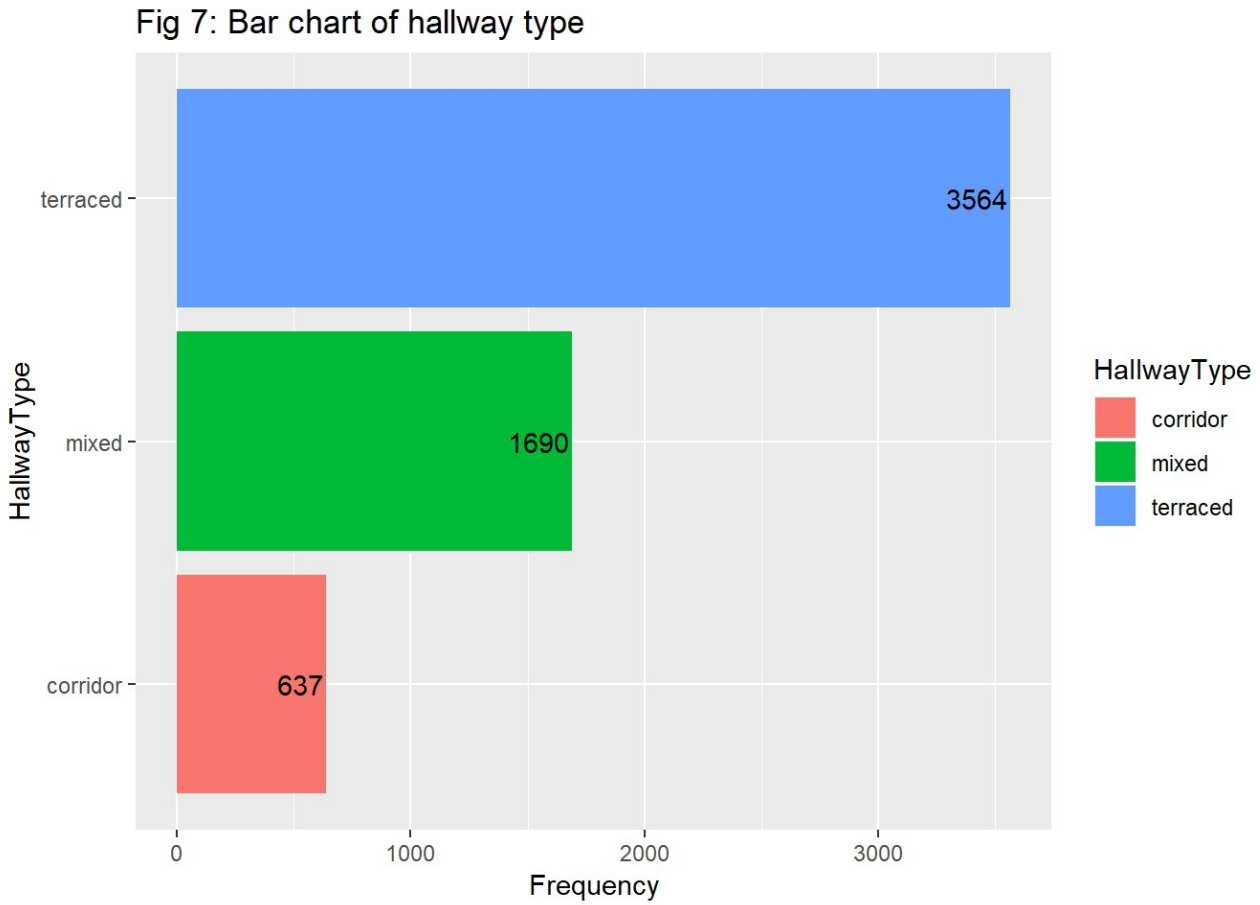
```
##-- HallwayType --## column 7

str(apartment$HallwayType)
```

```
##  Factor w/ 3 levels "corridor","mixed",..: 3 1 1 3 2 1 2 2 2 2 ...
```

```
unique(apartment$HallwayType)
```

```
## [1] terraced corridor mixed
## Levels: corridor mixed terraced
```

```
ggplot(apartment, aes(HallwayType)) +
  geom_bar(aes(fill = HallwayType)) +
  coord_flip()+
  geom_text(stat='count', aes(label=..count..), hjust=1.05) +
  scale_y_continuous(name="Frequency") +
  scale_x_discrete(name="HallwayType")+
  theme(legend.position = "right")+
  ggtitle("Fig 7: Bar chart of hallway type")
```



From fig 7, it is very clear that teracced apartments are very popular followed by mixed and corridor respectively.

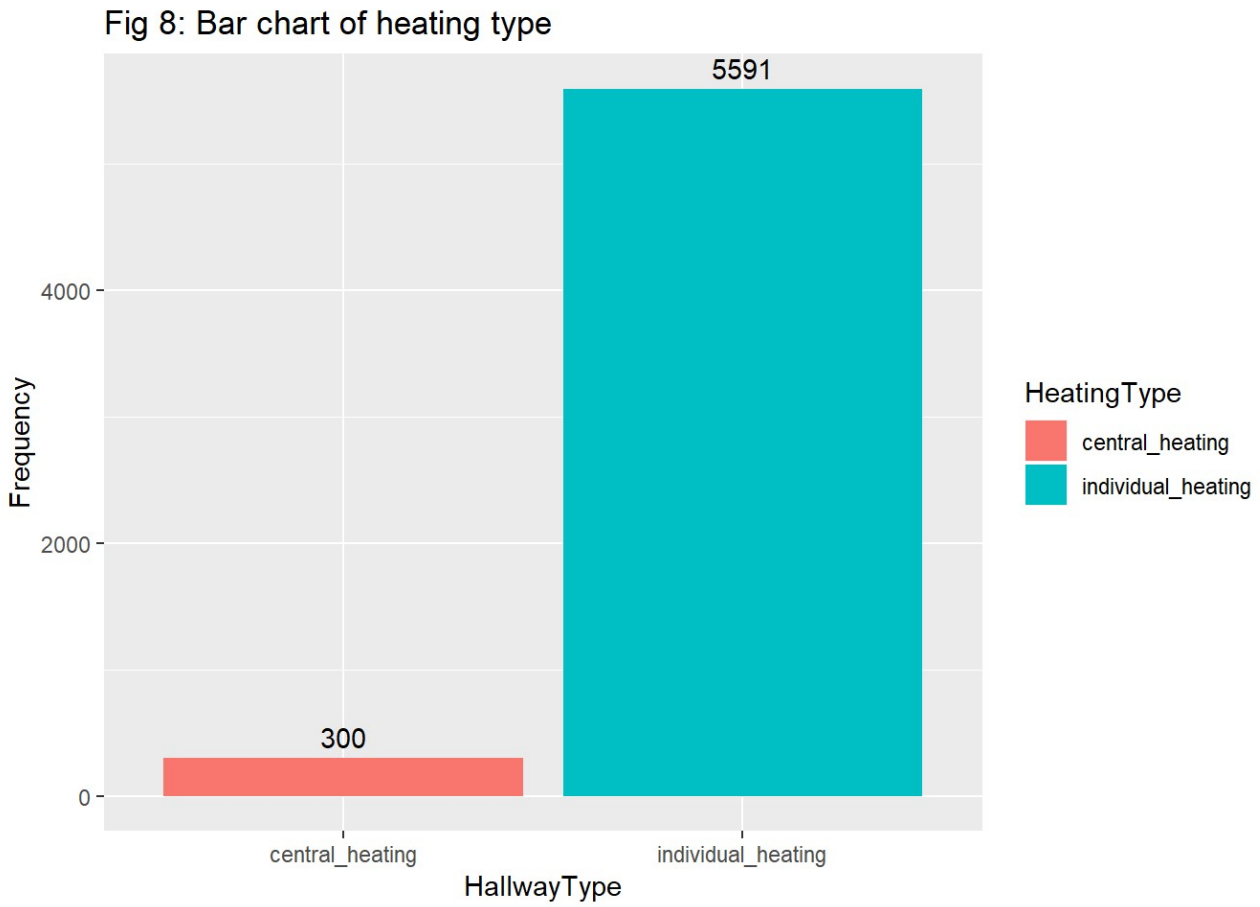
```
##-- HeatingType --## column 8

str(apartment$HeatingType)
```

```
##  Factor w/ 2 levels "central_heating",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
unique(apartment$HeatingType)
```

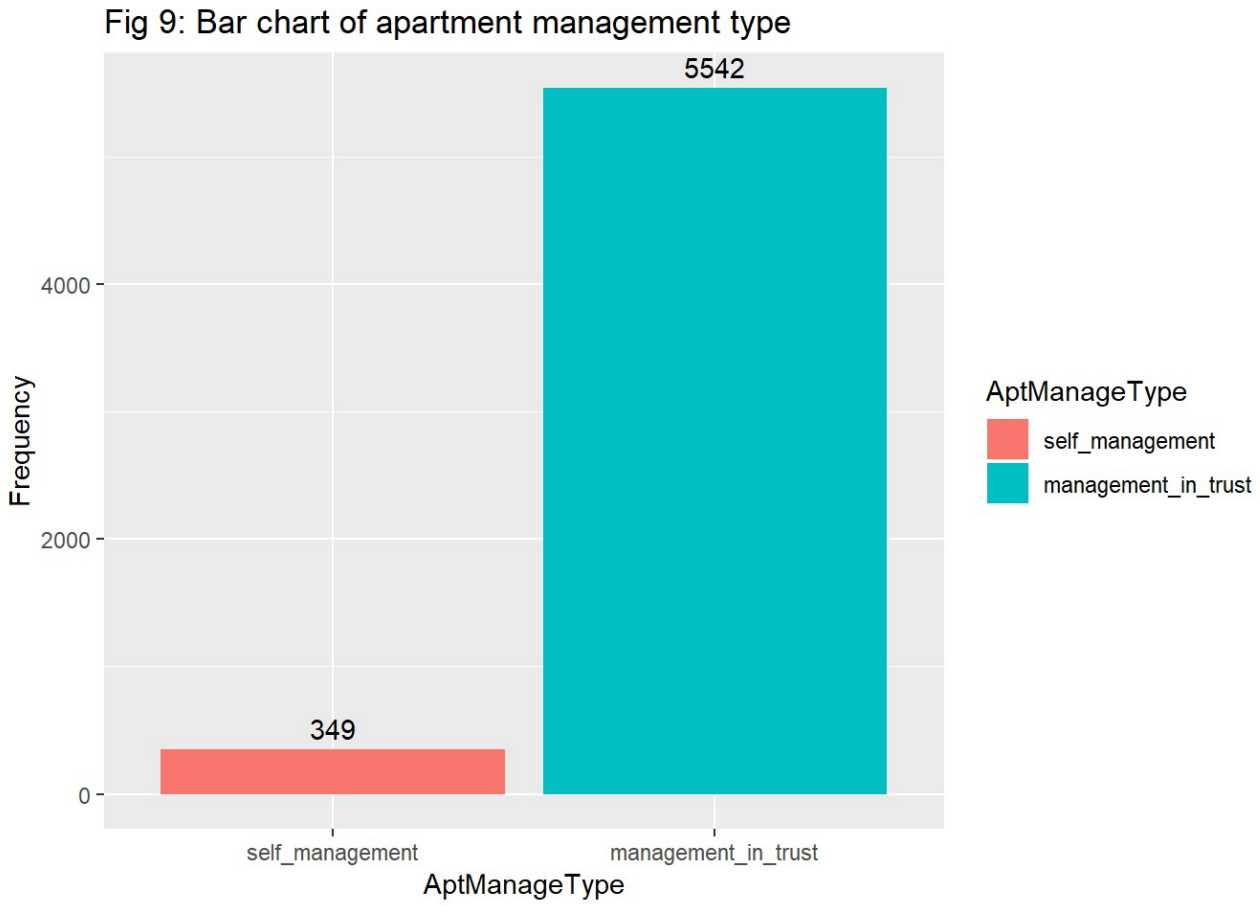
```
## [1] individual_heating central_heating
## Levels: central_heating individual_heating
```

```
ggplot(apartment, aes(HeatingType)) +
  geom_bar(aes(fill = HeatingType)) +
  geom_text(stat='count', aes(label=..count..), vjust = -0.5) +
  scale_y_continuous(name="Frequency") +
  scale_x_discrete(name="HallwayType")+
  theme(legend.position = "right")  +
  ggtitle("Fig 8: Bar chart of heating type")
```



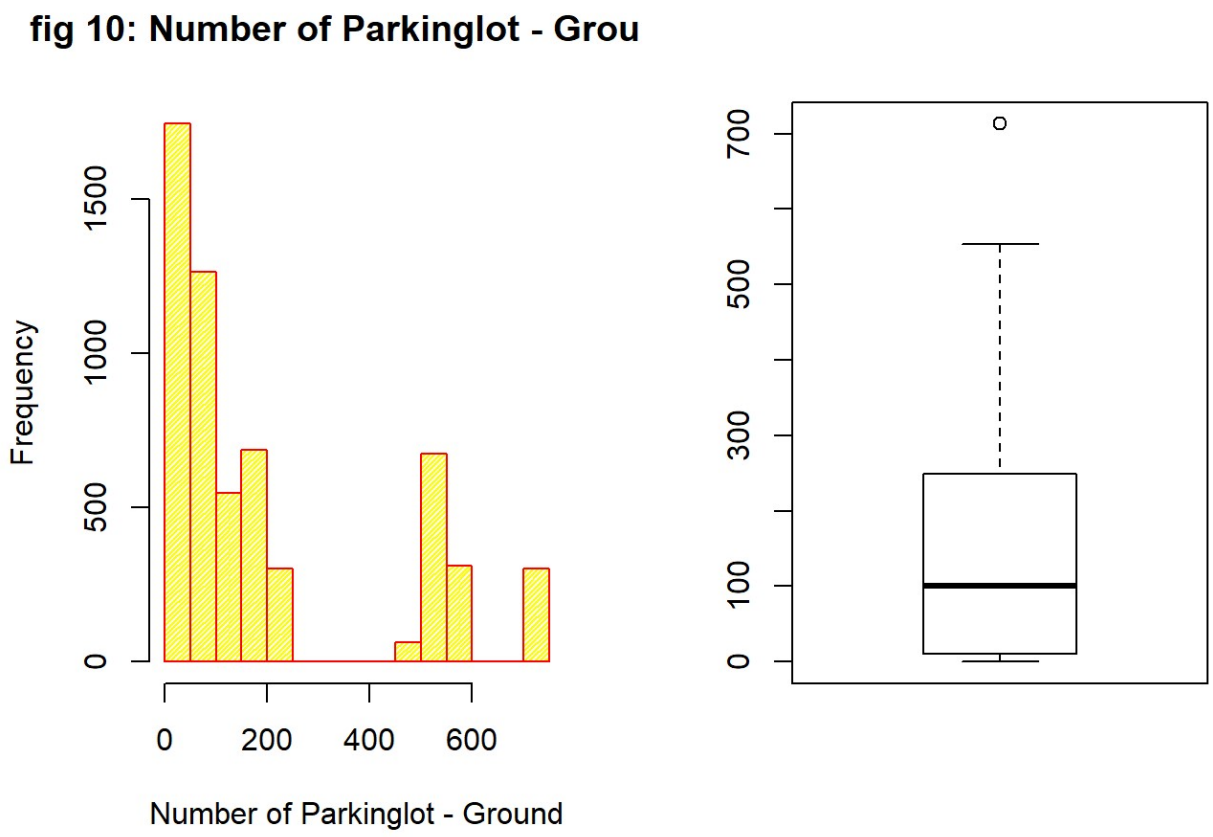
From fig 8, it is clear that central heating is very popular choice that individual heating in apartments.

```
##-- AptManageType --## column 9

ggplot(apartment, aes(AptManageType)) +
  geom_bar(aes(fill = AptManageType)) +
  geom_text(stat='count', aes(label=..count..), vjust = -0.5) +
  scale_y_continuous(name="Frequency") +
  scale_x_discrete(name="AptManageType")+
  theme(legend.position = "right")  +
  ggtitle("Fig 9: Bar chart of apartment management type")
```



From fig 9, it is noticiable that trust management is popular choice than self management for apartments.

```
##-- N_Parkinglot.Ground. --## column 10

par(mfrow=c(1,2))

hist(apartment$N_Parkinglot.Ground.,
     main="fig 10: Number of Parkinglot - Ground",
     xlab="Number of Parkinglot - Ground",
     ylab="Frequency",
     border="red",
     col="yellow",
     density=50)

boxplot(apartment$N_Parkinglot.Ground.)
```

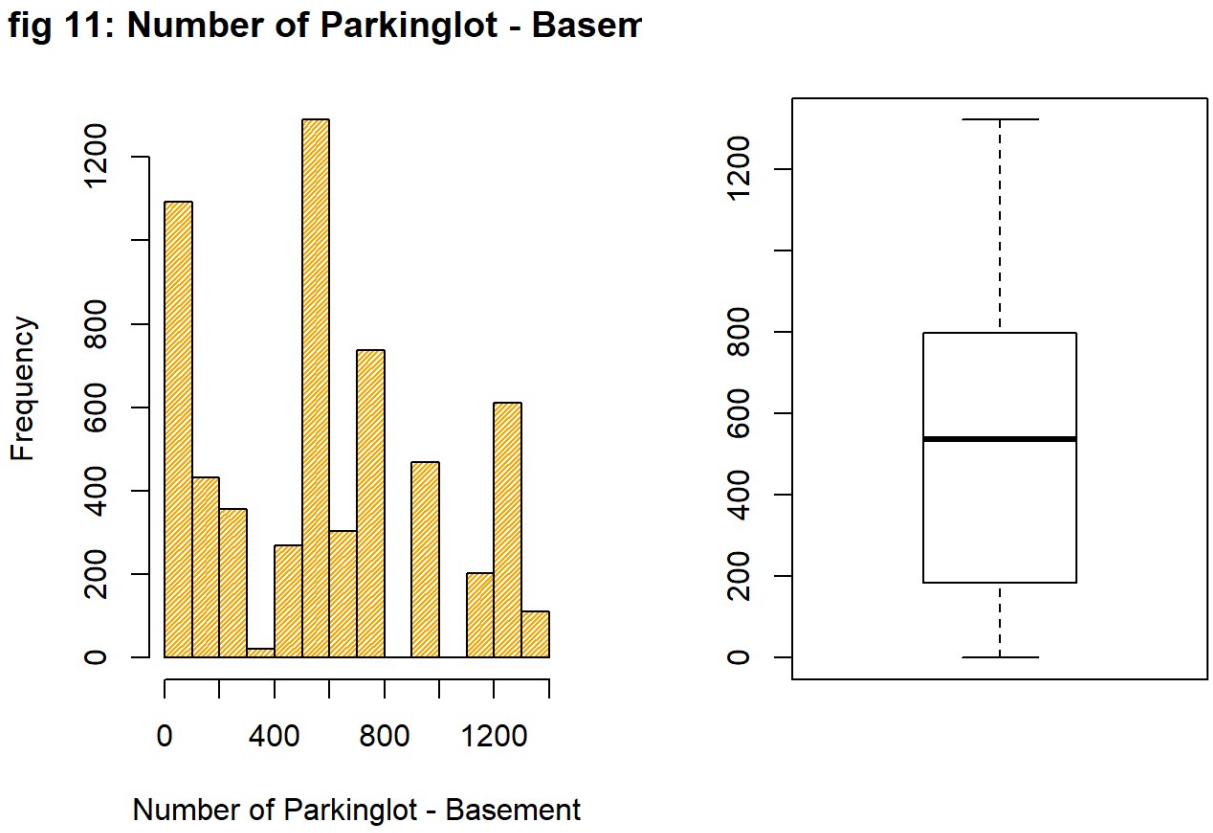



```
#--- dealing with outlier for ground parking
length(which(apartment$N_Parkinglot.Ground. == 713))
```

```
## [1] 300
```

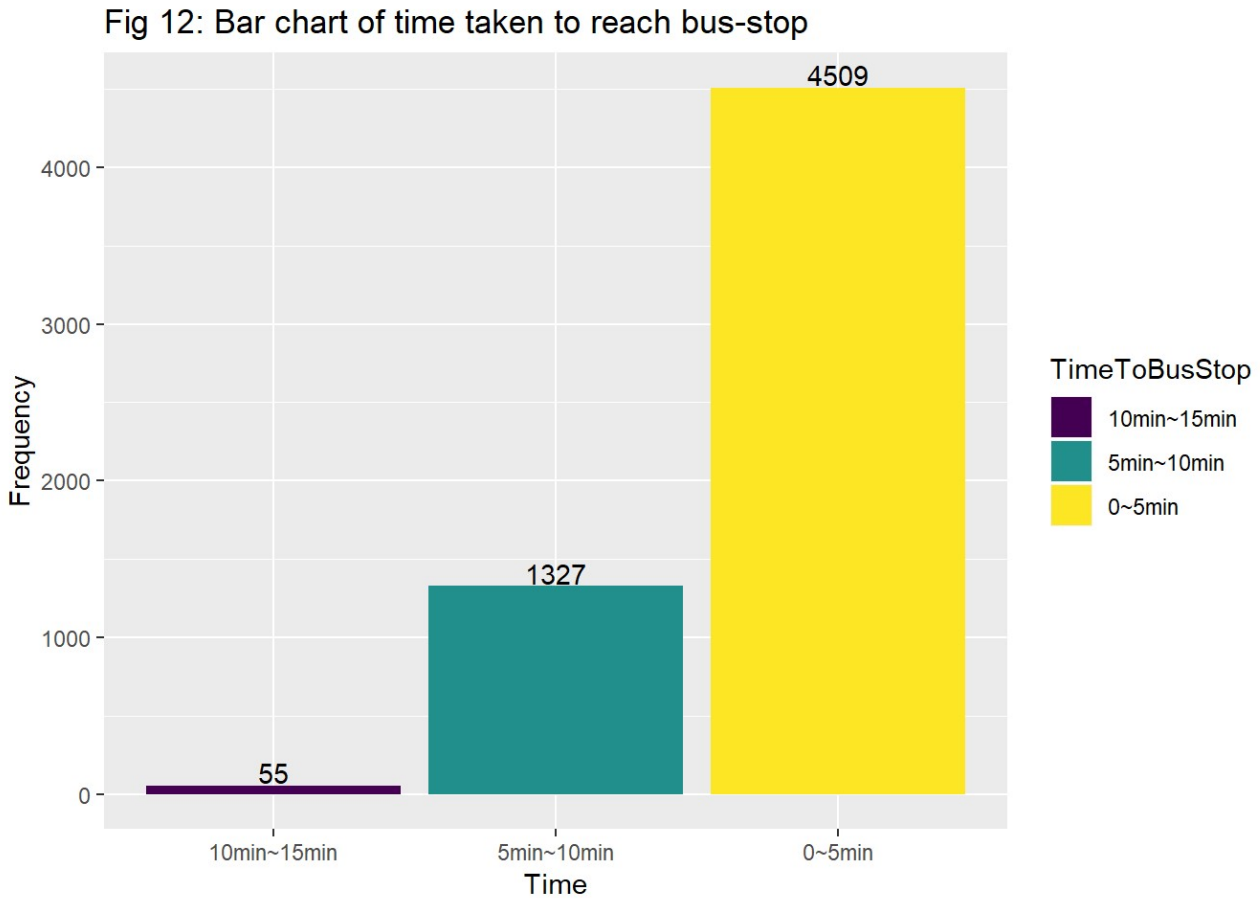
In fig 10, bar plot it is quite clear that most of the apartment has less that 250 ground parkings. From box plot there is noticiable outlier at around 700, however after further investigation, there are 300 apartments where ground parking has capacity of 713. Therefore, it is definitely not outlier value.

```
##-- N_Parkinglot.Basement. --## column 11

par(mfrow=c(1,2))

hist(apartment$N_Parkinglot.Basement.,
     main="fig 11: Number of Parkinglot - Basement",
     xlab="Number of Parkinglot - Basement",
     ylab="Frequency",
     border="black",
     col="orange",
     density=50)

boxplot(apartment$N_Parkinglot.Basement.)
```



From fig 11, it is clear that basement parking count is relatively destributed.

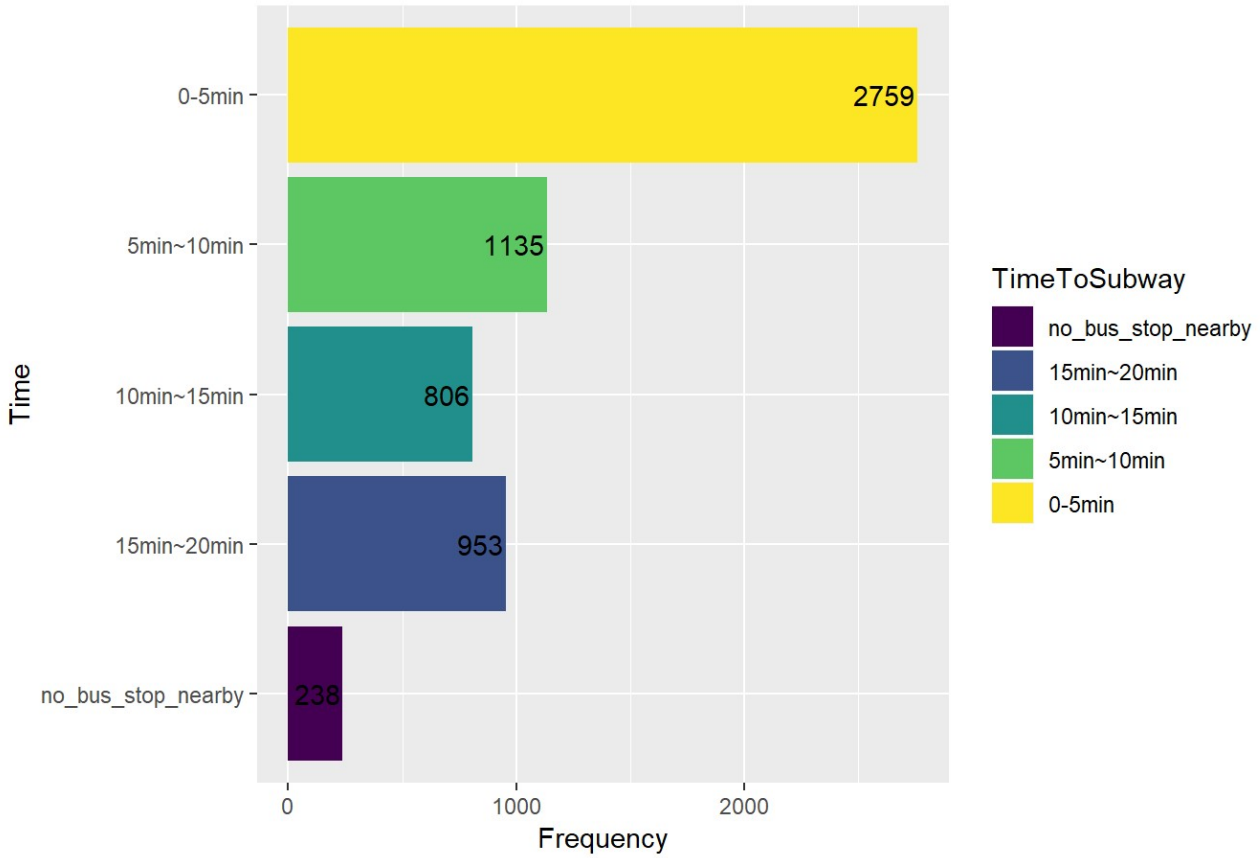
```
##-- TimeToBusStop --## column 12

ggplot(apartment, aes(TimeToBusStop)) +
  geom_bar(aes(fill = TimeToBusStop)) +
  geom_text(stat='count', aes(label=..count..), vjust=-0.1) +
  scale_y_continuous(name="Frequency") +
  scale_x_discrete(name="Time")+
  theme(legend.position = "right") +
  ggtitle("Fig 12: Bar chart of time taken to reach bus-stop")
```



From fig 12. we learn that most of the apartment has 0 to 5 min proximity to bus stops and very few are on the distance more than 10 min.

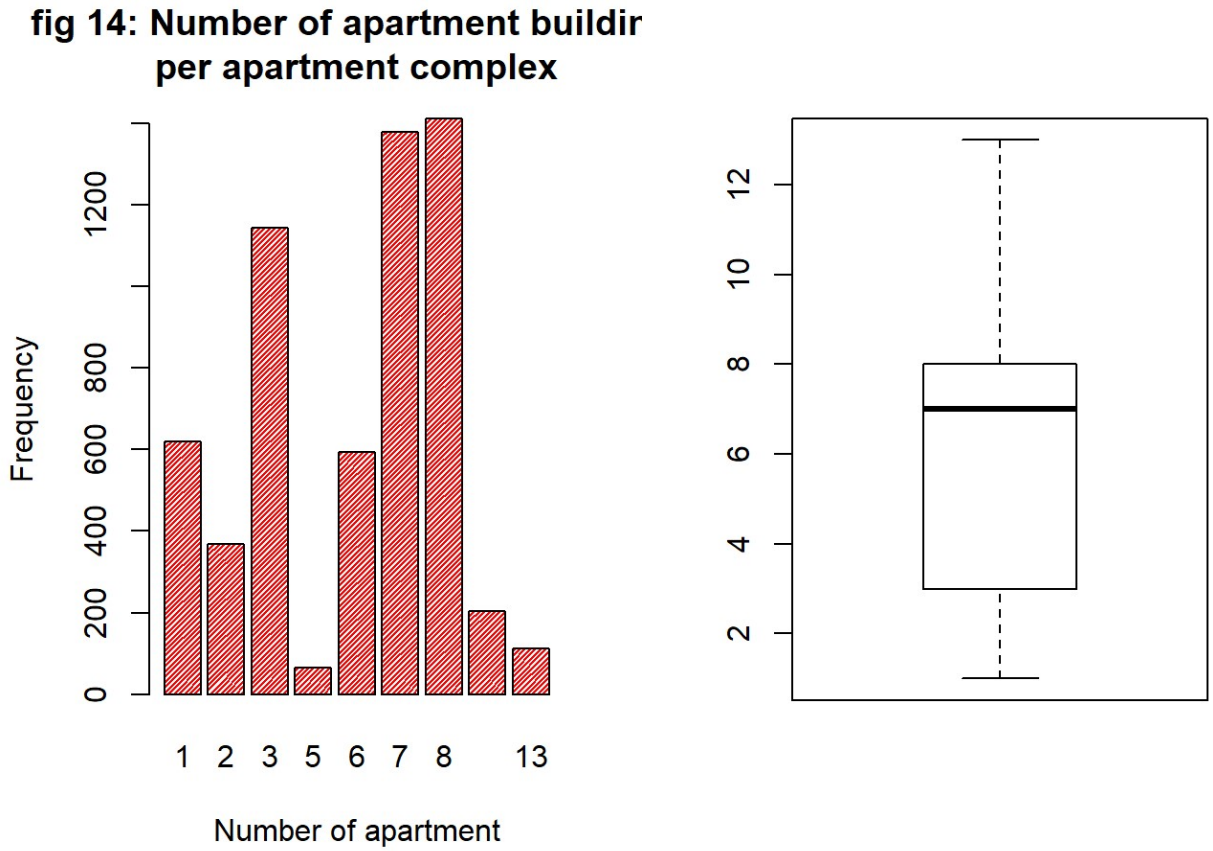
```
##-- TimeToSubway --## column 13

ggplot(apartment, aes(TimeToSubway)) +
  geom_bar(aes(fill = TimeToSubway)) +
  coord_flip()+
  geom_text(stat='count', aes(label=..count..), hjust=1.05) +
  scale_y_continuous(name="Frequency") +
  scale_x_discrete(name="Time")+theme(legend.position = "right") +
  ggtitle("Fig 13: Bar chart of time taken to reach subway")
```

Fig 13: Bar chart of time taken to reach subway



From fig 13. we learn that most of the apartment has 0 to 5 min proximity to subway and very few are on the distance more than 20 min.

```
##-- N_APT --## column 14

par(mfrow=c(1,2))

barplot(table(apartment$N_APT),
        main="fig 14: Number of apartment building \nper apartment complex",
        xlab="Number of apartment",
        ylab="Frequency",
        border="black",
        col="red",
        density=50)

boxplot(apartment$N_APT)
```



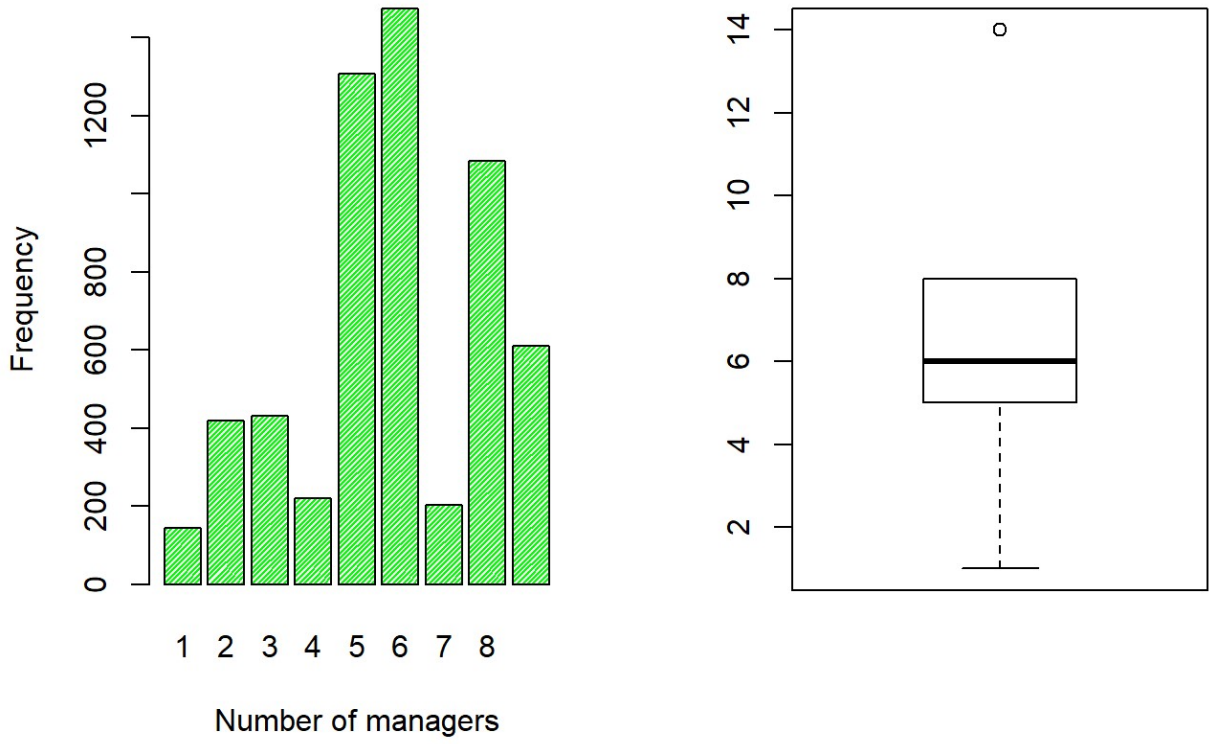
From fig 14, it is seen that most apartment has 3, 7 or 8 apartments in the complex and very few has 5 or 13.

```
##-- N_manager --## column 15

summary(apartment$N_manager)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    5.00    6.00    6.31    8.00   14.00
```

```
par(mfrow=c(1,2))

barplot(table(apartment$N_manager),
        main="fig 15 : Number of managers",
        xlab="Number of managers",
        ylab="Frequency",
        border="black",
        col="green",
        density=50)

boxplot(apartment$N_manager)
```

fig 15 : Number of managers




```
length(which(apartment$N_manager == 14))
```

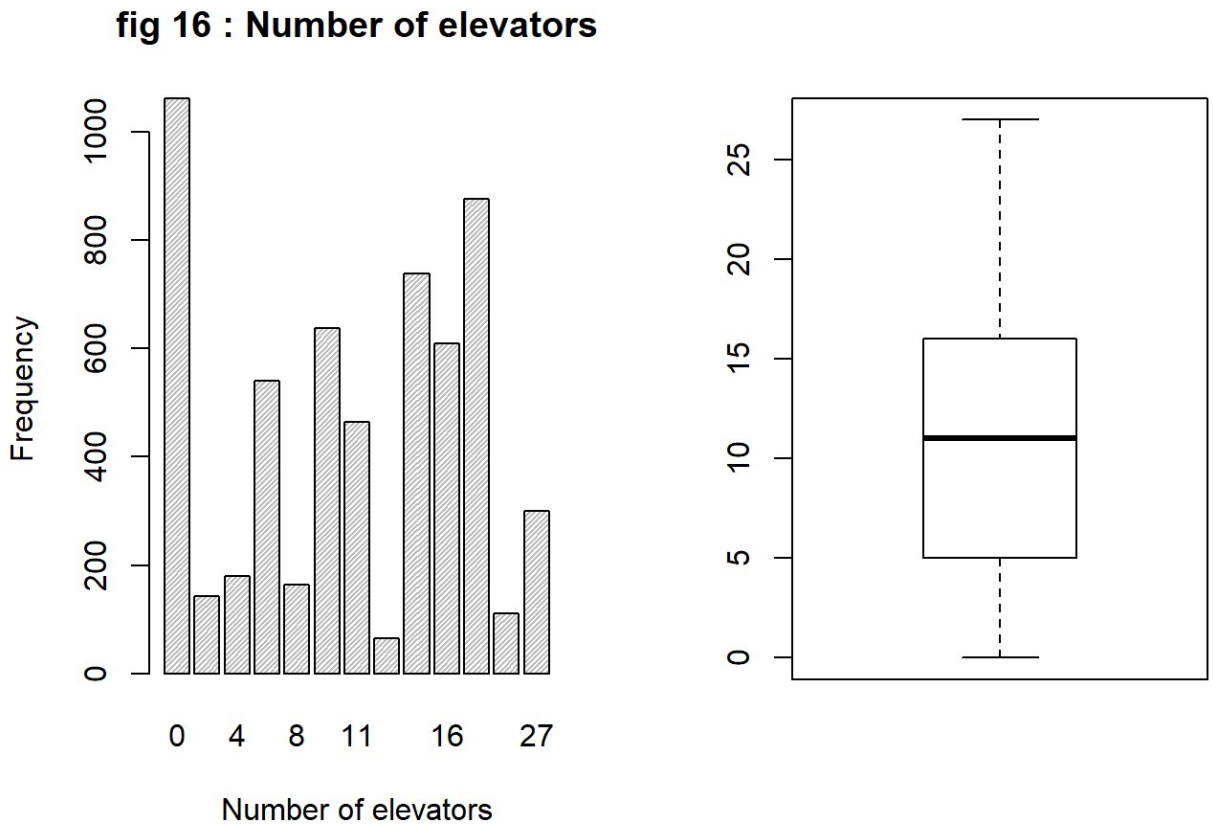
```
## [1] 610
```

From fig 15, it is clear that most apartments has 5,6 or 8 managers to manage works like cleaning etc.

```
##-- N_elevators --## column 16

summary(apartment$N_elevators)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    5.00   11.00   11.15   16.00   27.00
```

```
par(mfrow=c(1,2))

barplot(table(apartment$N_elevators),
        main="fig 16 : Number of elevators",
        xlab="Number of elevators",
        ylab="Frequency",
        border="black",
        col="grey",
        density=50)

boxplot(apartment$N_elevators)
```



From fig 16, it is clear that most apartment has no elevators while still others has generally more than 6 to 27 elevators.

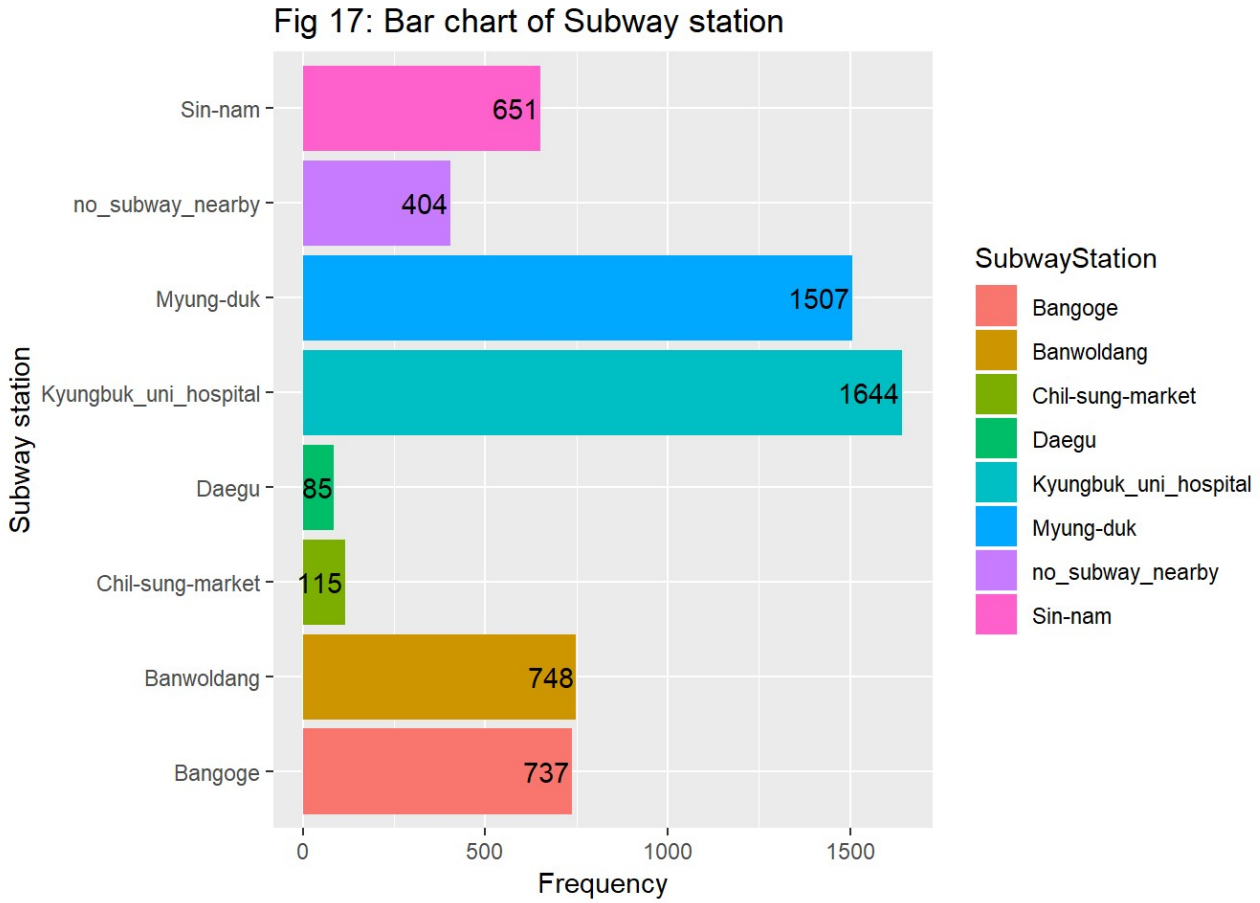
```
##-- SubwayStation --## column 17

str(apartment$SubwayStation)
```

```
##  Factor w/ 8 levels "Bangoge","Banwoldang",..: 5 4 4 8 6 6 6 6 6 6 ...
```

```
unique(apartment$SubwayStation)
```

```
## [1] Kyungbuk_uni_hospital Daegu                Sin-nam
## [4] Myung-duk             Chil-sung-market     Bangoge
## [7] Banwoldang            no_subway_nearby
## 8 Levels: Bangoge Banwoldang Chil-sung-market ... Sin-nam
```

```
#https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf
ggplot(apartment, aes(SubwayStation)) +
  geom_bar(aes(fill = SubwayStation)) +
  coord_flip()+
  geom_text(stat='count',aes(label=..count..), hjust=1.05) +
  scale_y_continuous(name="Frequency") +
  scale_x_discrete(name="Subway station")+theme(legend.position = "right") +
  ggtitle("Fig 17: Bar chart of Subway station")
```



From fig 17, we get interesting insight that most properties are near Kyungbuk university hopsital and myung-duk while very few are around Daegu and chil-sun market.

```
##-- N_FacilitiesNearBy.PublicOffice. --## column 18
##-- N_FacilitiesNearBy.Hospital. --## column 19
##-- N_FacilitiesNearBy.Dpartmentstore. --## column 20
##-- N_FacilitiesNearBy.Mall. --## column 21
##-- N_FacilitiesNearBy.ETC. --## column 22
##-- N_FacilitiesNearBy.Park. --## column 23


summary(apartment$N_FacilitiesNearBy.PublicOffice.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   5.000   4.142   5.000   7.000
```

```
summary(apartment$N_FacilitiesNearBy.Hospital.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   1.000   1.296   2.000   2.000
```

```
summary(apartment$N_FacilitiesNearBy.Dpartmentstore.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  1.0000  0.8963  2.0000  2.0000
```

```
summary(apartment$N_FacilitiesNearBy.Mall.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  1.0000  1.0000  0.9414  1.0000  2.0000
```

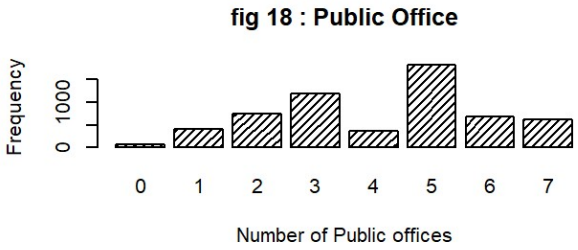
```
summary(apartment$N_FacilitiesNearBy.ETC.)
```

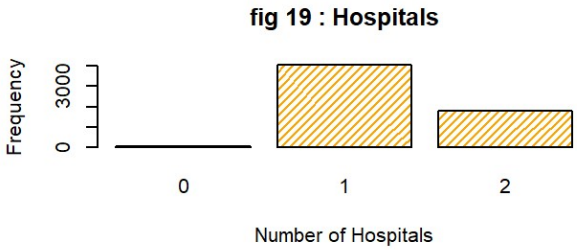
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   1.941   5.000   5.000
```

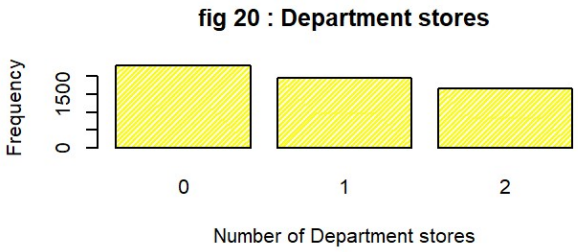
```
summary(apartment$N_FacilitiesNearBy.Park.)
```

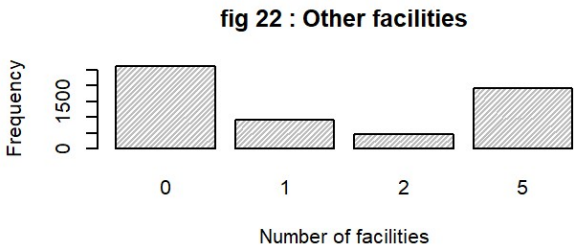
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  1.0000  0.6542  1.0000  2.0000
```

```
par(mfrow=c(3,2))

barplot(table(apartment$N_FacilitiesNearBy.PublicOffice.),
        main="fig 18 : Public Office",
        xlab="Number of Public offices",
        ylab="Frequency",
        border="black",
        col="black",
        density=30)

barplot(table(apartment$N_FacilitiesNearBy.Hospital.),
        main="fig 19 : Hospitals",
        xlab="Number of Hospitals",
        ylab="Frequency",
        border="black",
        col="orange",
        density=30)

barplot(table(apartment$N_FacilitiesNearBy.Dpartmentstore.),
        main="fig 20 : Department stores",
        xlab="Number of Department stores",
        ylab="Frequency",
        border="black",
        col="yellow",
        density=50)

barplot(table(apartment$N_FacilitiesNearBy.Mall.),
        main="fig 21 : Malls",
        xlab="Number of malls",
        ylab="Frequency",
        border="black",
        col="pink",
        density=50)

barplot(table(apartment$N_FacilitiesNearBy.ETC.),
        main="fig 22 : Other facilities",
        xlab="Number of facilities",
        ylab="Frequency",
        border="black",
        col="grey",
        density=50)

barplot(table(apartment$N_FacilitiesNearBy.Park.),
        main="fig 23 : Parks",
        xlab="Number of parks",
        ylab="Frequency",
        border="black",
        col="green",
        density=80)
```









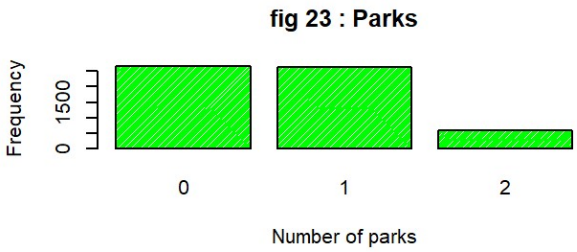


Fig 18 to fig 23 provide visual details about number of different facilities around the apartments.

```
##-- N_SchoolNearBy.Elementary. --## column 24
##-- N_SchoolNearBy.Middle. --## column 25
##-- N_SchoolNearBy.High. --## column 26
##-- N_SchoolNearBy.University. --## column 27


summary(apartment$N_SchoolNearBy.Elementary.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   3.022   4.000   6.000
```

```
summary(apartment$N_SchoolNearBy.Middle.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   2.418   3.000   4.000
```

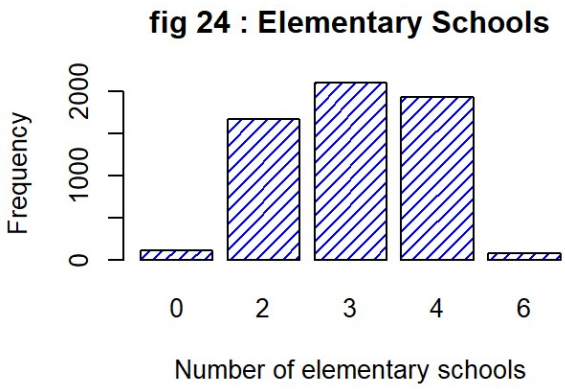
```
summary(apartment$N_SchoolNearBy.High.)
```

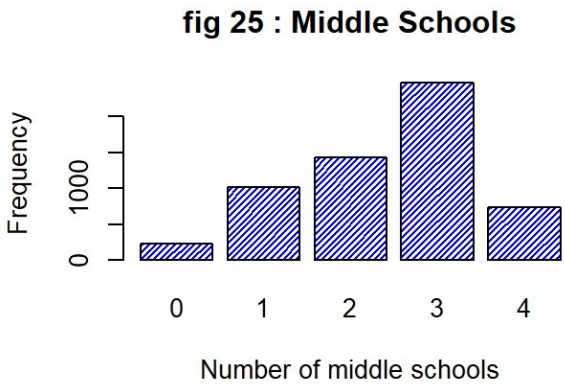
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   2.000   2.659   4.000   5.000
```

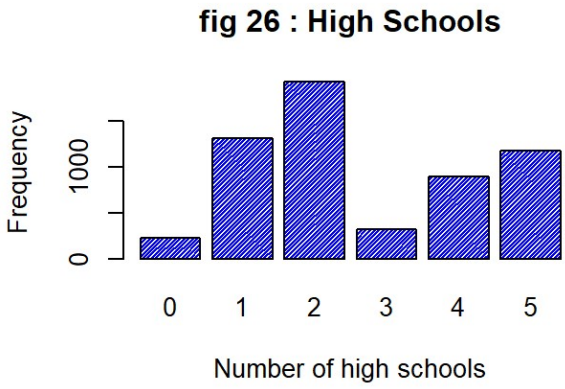
```
summary(apartment$N_SchoolNearBy.University.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   2.000   2.765   4.000   5.000
```

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

barplot(table(apartment$N_SchoolNearBy.Elementary.),
        main="fig 24 : Elementary Schools",
        xlab="Number of elementary schools",
        ylab="Frequency",
        border="black",
        col="blue",
        density=20)

barplot(table(apartment$N_SchoolNearBy.Middle.),
        main="fig 25 : Middle Schools",
        xlab="Number of middle schools",
        ylab="Frequency",
        border="black",
        col="blue",
        density=40)

barplot(table(apartment$N_SchoolNearBy.High.),
        main="fig 26 : High Schools",
        xlab="Number of high schools",
        ylab="Frequency",
        border="black",
        col="blue",
        density=60)

barplot(table(apartment$N_SchoolNearBy.University.),
        main="fig 27 : University",
        xlab="Number of university",
        ylab="Frequency",
        border="black",
        col="blue",
        density=80)
```







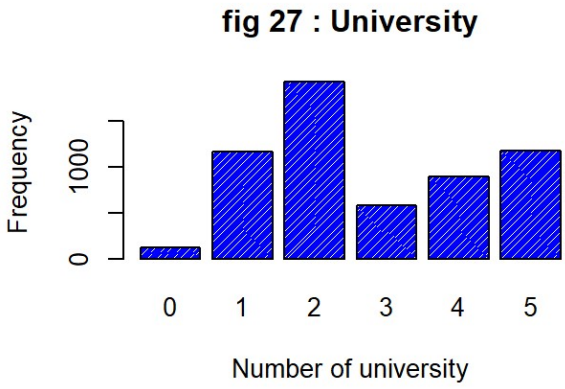


Fig 24 to fig 27 provides good presentation of number of educational center near to the apartments.

```
##-- N_FacilitiesInApt --## column 28
##-- N_FacilitiesNearBy.Total. --## column 29
##-- N_SchoolNearBy.High. --## column 30

summary(apartment$N_FacilitiesInApt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    4.00    5.00    5.81    7.00   10.00
```

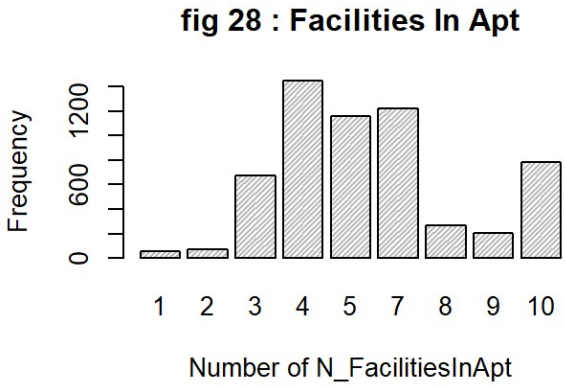
```
summary(apartment$N_FacilitiesNearBy.Total.)
```

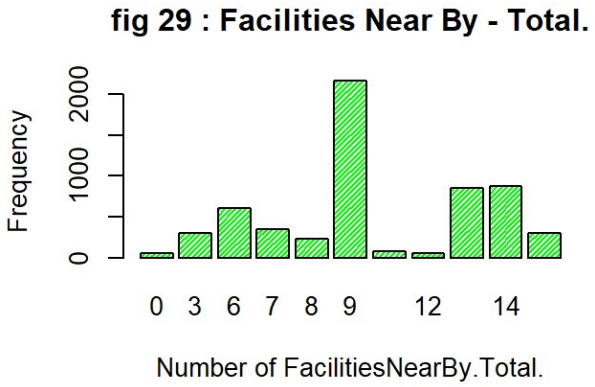
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   8.000   9.000   9.871  13.000  16.000
```

```
summary(apartment$N_SchoolNearBy.Total.)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    7.00   10.00   10.86   15.00   17.00
```

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

barplot(table(apartment$N_FacilitiesInApt),
        main="fig 28 : Facilities In Apt",
        xlab="Number of N_FacilitiesInApt",
        ylab="Frequency",
        border="black",
        col="grey",
        density=50)

barplot(table(apartment$N_FacilitiesNearBy.Total.),
        main="fig 29 : Facilities Near By - Total.",
        xlab="Number of FacilitiesNearBy.Total.",
        ylab="Frequency",
        border="black",
        col="green",
        density=50)

barplot(table(apartment$N_SchoolNearBy.Total.),
        main="fig 30 : School Near By - Total.",
        xlab="Number of SchoolNearBy.Total.",
        ylab="Frequency",
        border="black",
        col="blue",
        density=50)
```





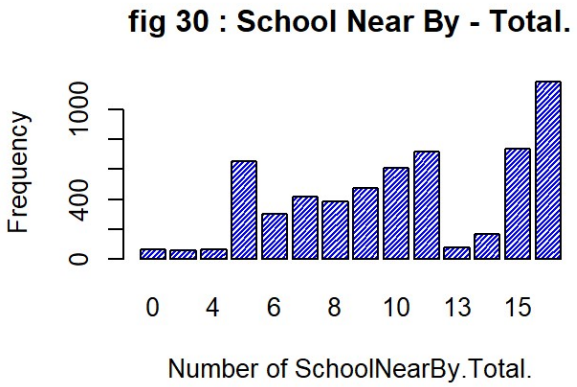
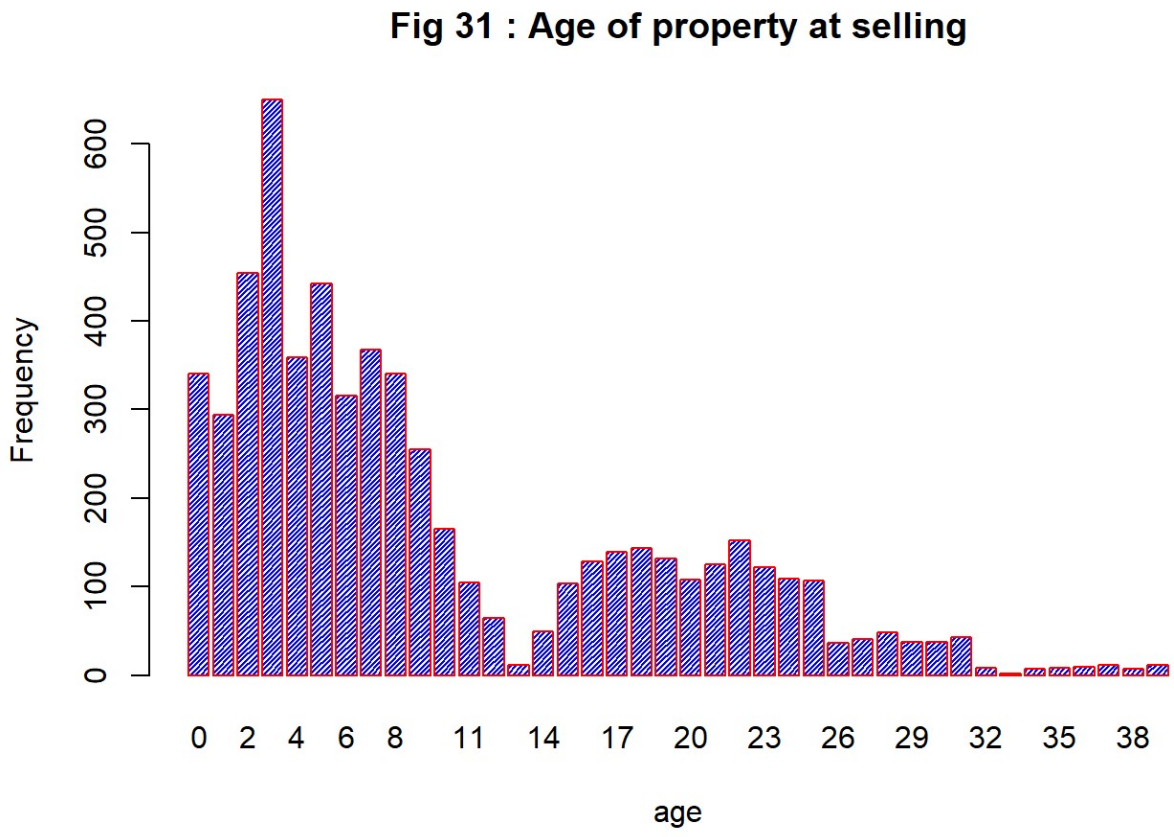


Fig 28 provides information about number of facilities in the apartment and it is quite clear thatmost apartments have more than 3 facilities and has at max 10 facilities.

fig 29 provides insight of total number of facilities (depicted in fig 18 to 23)

Fig 30 provides insight of total number of educational opportunities (depicted in fig 24 to fig 27)

```
##-- AgeWhenSold --## column 31 NEWLY CREATED

barplot(table(apartment$AgeWhenSold),
main="Fig 31 : Age of property at selling",
xlab="age",
ylab="Frequency",
border="red",
col="blue",
density=50)
```

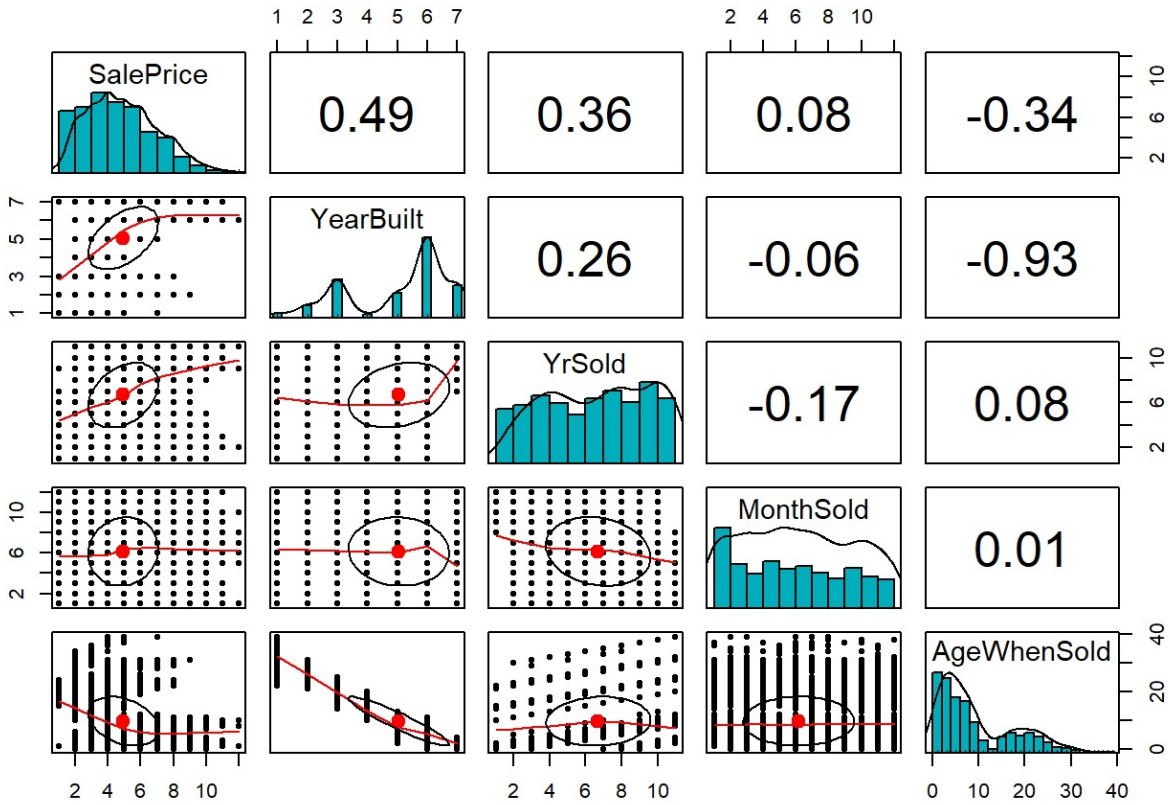


From fig 31, it is quite clear that most of the poperties sold are newer and age less that 12 years. Surprisingly properties aged around 13 seems to be sold less for their age.

3.2 Bi-variate and data revision

In the following plot many desciptive features are calculated and observed against each other and target feature SalePrice

```
##-- SalePrice, YearBuilt, YrSold, MonthSold--##

pairs.panels(apartment[,c(1,2,3,4,31)],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```

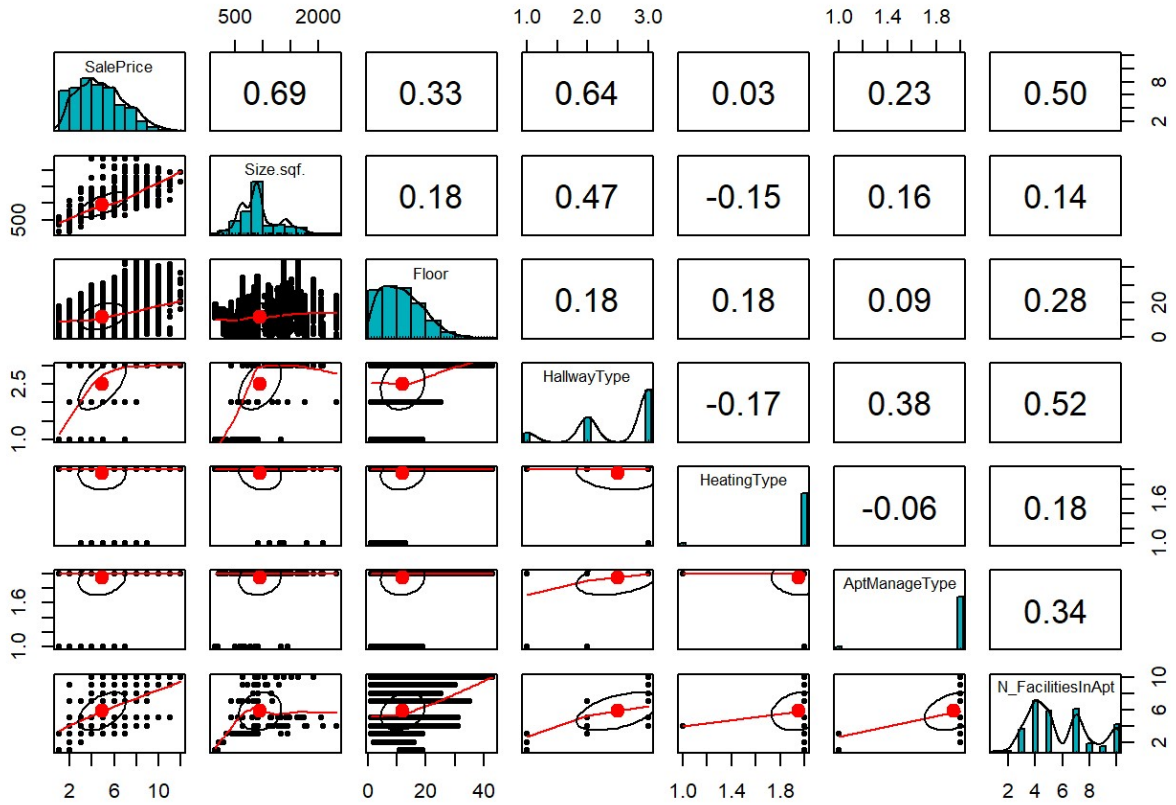



```
#scatter matrix
#http://www.sthda.com/english/wiki/scatter-plot-matrices-r-base-graphs [3]
```

From above scatter matrix it becomes quite clear that there is no significant impact (cor 0.08) of monthSold on target variable soldPrice thus it can be eliminated for funther analysis in phase II. Further, newly created variable AgeWhenSold has very strong negative correlation (-0.93) with YearBuilt; thus variable YearBulid can also be eliminated to avoid redundancy.

Outcome 1 : eliminating variable monthSold on the ground of irrelevancy Outcome 2 : eliminating variable YearBuilt on the ground of redundancy

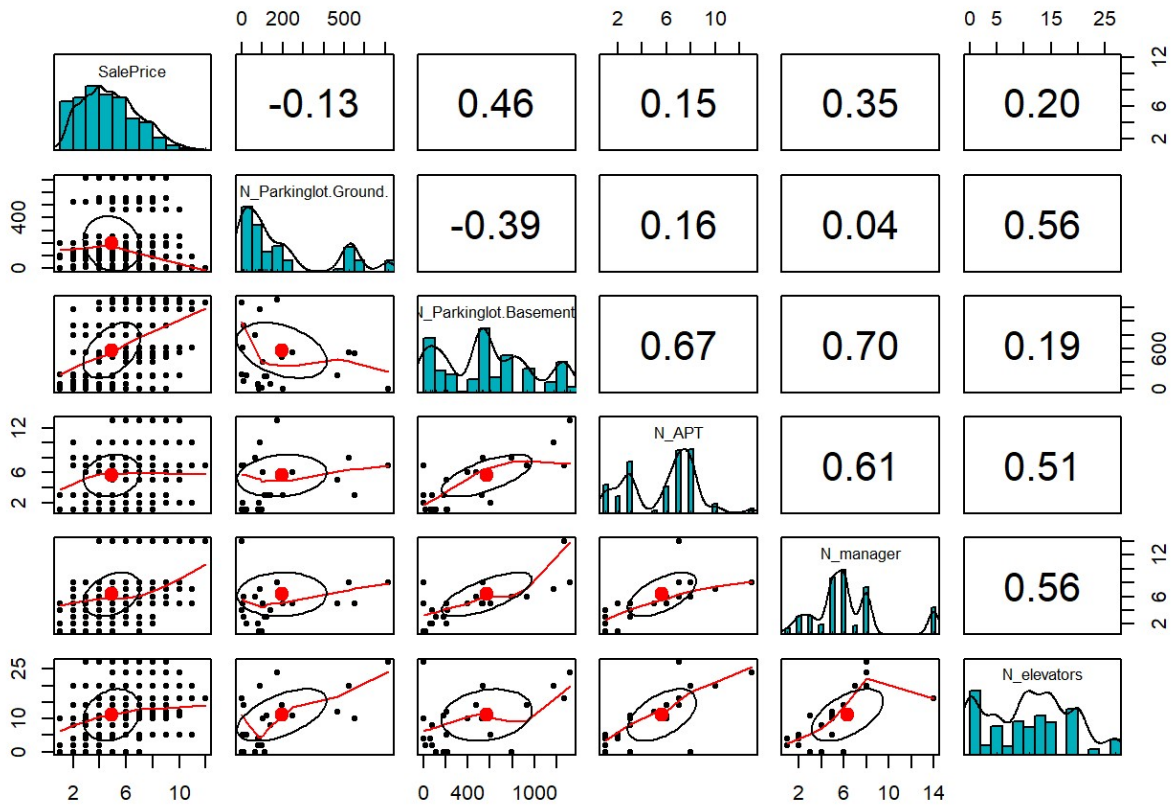
```
##-- SalePrice, Size.sqf., Floor, HallwayType, HeatingType, AptManageType --##

pairs.panels(apartment[,c(1,5,6,7,8,9,28)],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



FRom above scatter matrix, it is clear that heating type has no much relevance (0.03) with target feature saleprice, thus it can be eliminated. Also the variable AptManageType be investigated to seek evedence for its relevance as it seems to be thinly related to salePrice(target feature).

Outcome 3 : eliminating variable HeatingType on the ground of irrelevancy

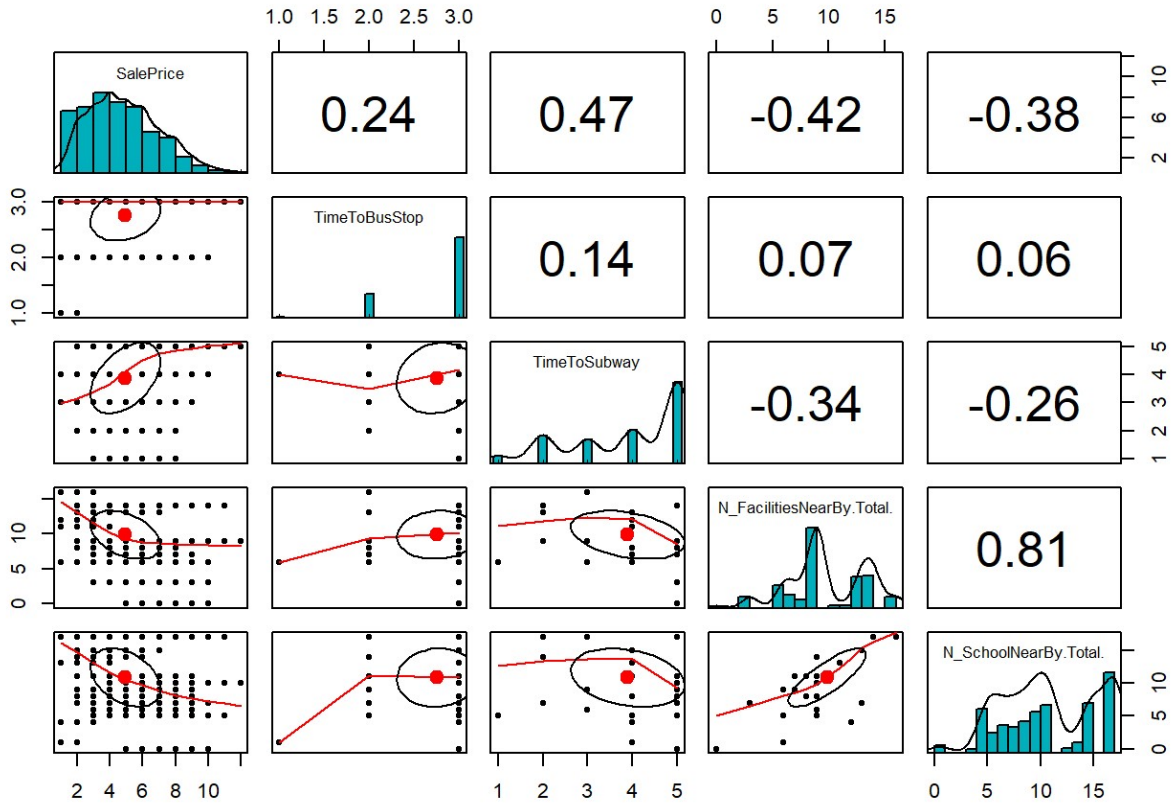
```
##-- SalePrice, N_Parkinglot.Ground, N_Parkinglot.Basement., N_APT, N_manager, N_elevator
s--##

pairs.panels(apartment[,c(1,10,11,14,15,16)],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



From above scatter matrix, it is observed that N_parkinglot.Ground is not much related (-0.13) to target feature saleprice thus it can be eliminated. Further, N_Apt is not much related (0.15) to target feature but has high correlation with N_Parkinglot.Basemnet (0.67) so feature N_apt can be eliminated. Although, desciptive feature N_elevators is not very strongly related (0.20) to salePrice, it has high relevance to eliminated feature (0.56 and 0.51 respectively) so it should be kept for further analysis in phase II.

Outcome 4 : eliminating variable N_parkinglot.Ground on the ground of irrelevancy Outcome 5 : eliminating variable N_Apt on the ground of irrelevancy and redundancy

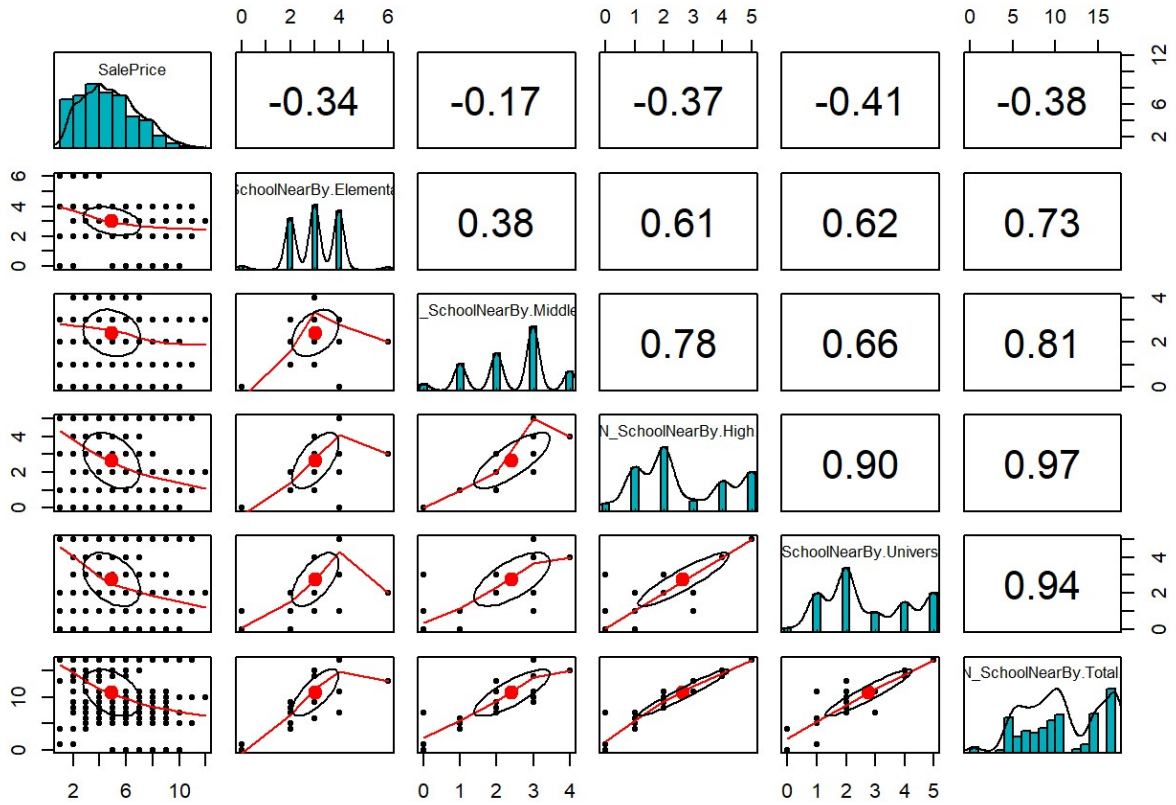
```
##-- SalePrice, TimeToBusStop, TimeToSubway, facilitiesNearBy--##

pairs.panels(apartment[,c(1,12,13,29,30)],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



From above. scatter matrix, all the proximity related variables are gathered and the descrive feature TimeToBusStop seems extremly odd based as it has no much significance to any other variable or to the target feature.

Outcome 6 : eliminating variable TimeToBusStop on the ground of irrelevancy

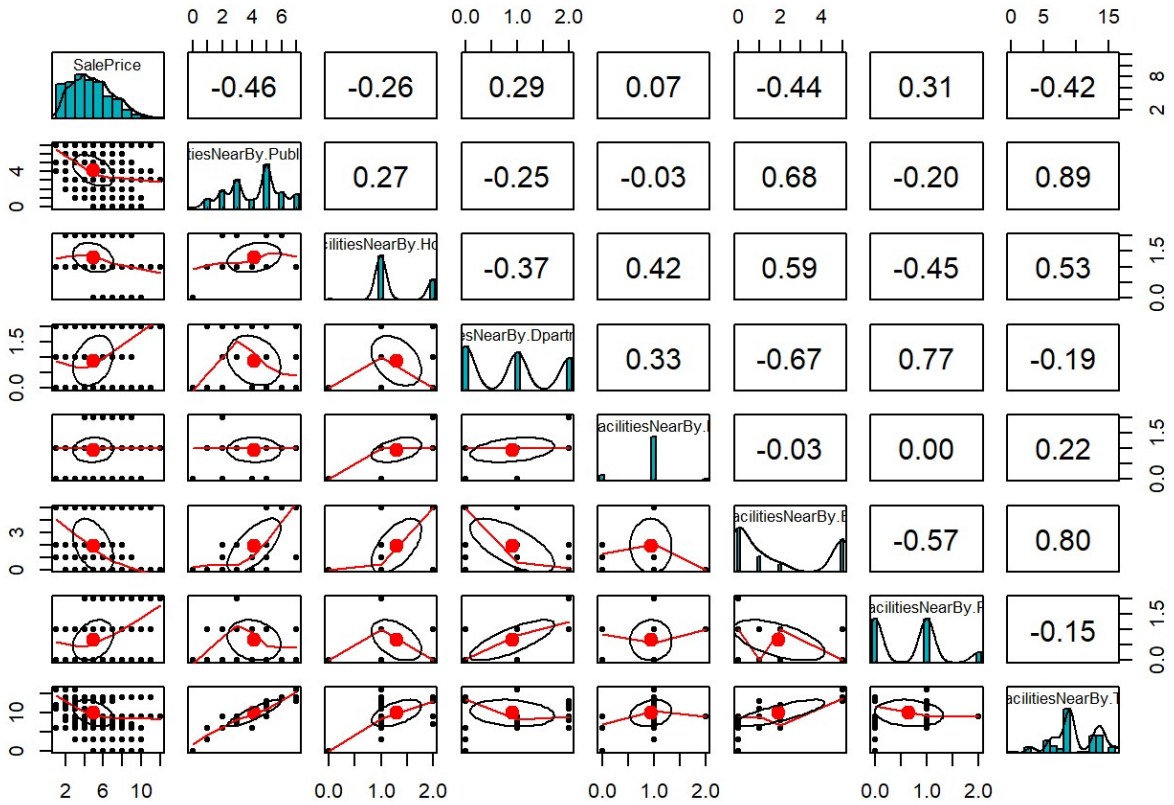
```
##-- educationalFacilities--##

pairs.panels(apartment[,c(1,24:27,30)],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



From above scatter matrix, it is clear that all educational facility variables are fairly relevant to SalePrice; HOwever the aggregate value of all the education related facilities is provided in feature N_SchoolNearBy.Total and this feature is negatively correlated (-0.38) to saleprice. So only N_SchoolNearBy.Total should be kept and others should be eliminated on the basis of strong redundancy.

Outcome 7 : eliminating variable N_SchoolNearBy.Elementary. on the ground of redundancy Outcome 8 : eliminating variable N_SchoolNearBy.Middle. on the ground of redundancy Outcome 9 : eliminating variable N_SchoolNearBy.High. on the ground of redundancy Outcome 10 : eliminating variable N_SchoolNearBy.University. on the ground of redundancy

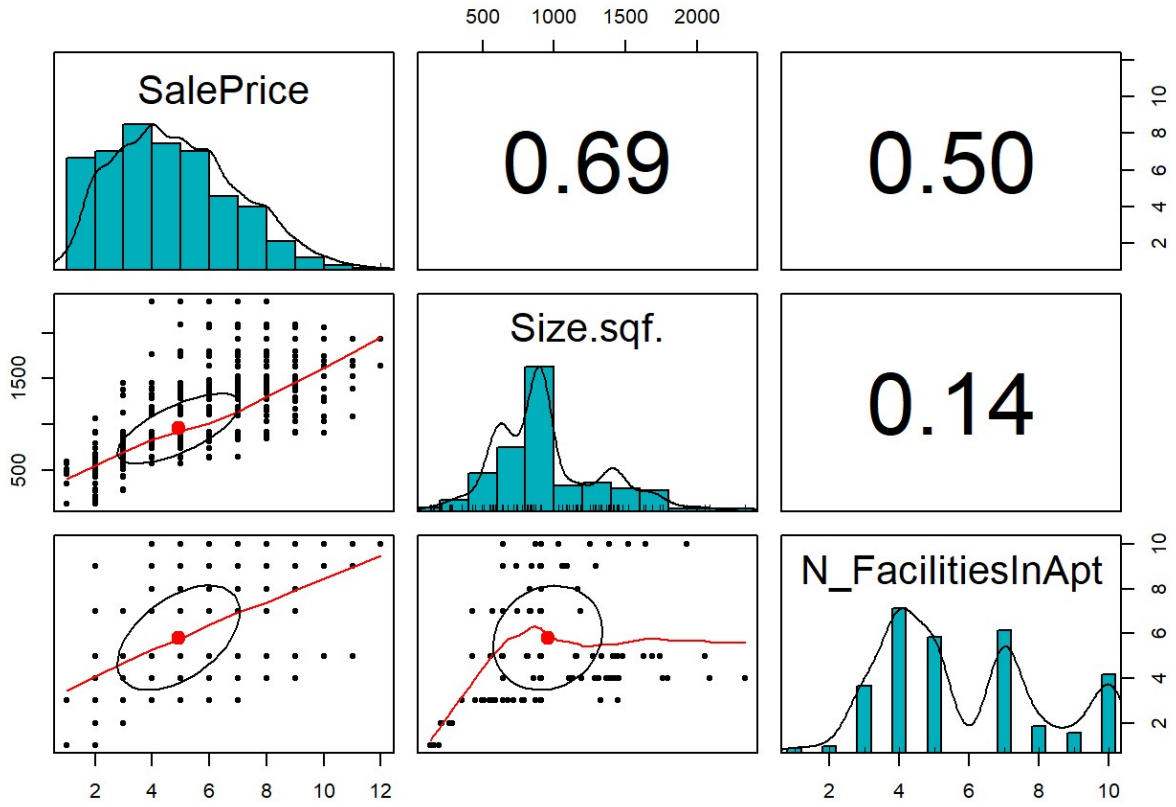
```
##-- Facilities nearBy --##

pairs.panels(apartment[,c(1,18:23,29)],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



From above Scatter matrix, it is clear that N_FacilitiesNearBy.Total is highlighly correlated to all the relevant variables for target feature other than N_FacilitiesNearBy.Dpartmnet (-0.19) and this variable is fairly (0.29) related to SalePrice. It seems clear that park facility is also relevant to sale price (0.31) but not related to total facilities(-0.15); therefore, to ensure the trend depicted by N_FacilitiesNearBy.Dpartmnet and N_FacilitiesNearBy.Park should be kept.

Outcome 11 : eliminating variable N_FacilitiesNearBy.PublicOffice. on the ground of redundancy Outcome 12 : eliminating variable N_FacilitiesNearBy.Hospital. on the ground of redundancy Outcome 13 : eliminating variable N_FacilitiesNearBy.Mall. on the ground of redundancy Outcome 14 : eliminating variable N_FacilitiesNearBy.ETC. on the ground of redundancy

```
##-- Apartment Characteristic --##

pairs.panels(apartment[,c(1,5,28)],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```



From above scatter matrix it is clear that both Sixe.sqf and N_facilitiesInApt are important as they bring some unique trends and not covered in each other as their relative correlation is 0.14 only. So both should be kept. To determine further evidence for indentifying significant of remaining descriptive features, we will follow on with mutlivariate analysis.

3.3 Multivariate Visualisation and interaction between numerical and categorical variables

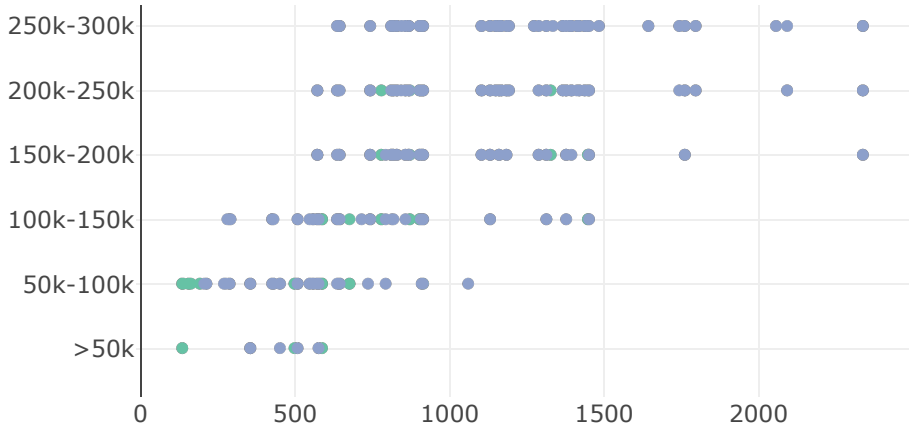
In the following code chuck we are trying to varify and observe relationship amongst various descriptive features and target feature.

```
## --------- Subway station, size, sale price --------- ##
#https://plot.ly/r/3d-scatter-plots/ [4]
plot_ly(apartment, x = apartment$Size.sqf. , y = apartment$SalePrice, color = apartment$S
ubwayStation) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'Size.sqf.'),
                      yaxis = list(title = 'SalePrice')))
```

From above scatter plot it becomes clear that location has significant impact on on the soldPrice of the apartment given that the apartment has similar squere feet area. For example, apartment near Myung-duk has bigger sqr feet area but lesser sold price wherein apartment near kyungbuk_uni_hospital has very high prices on even small apartments.

```
plot_ly(apartment, x = apartment$Size.sqf. , y = apartment$SalePrice, color = apartment$A
ptManageType) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'Size.sqf.'),
                      yaxis = list(title = 'SalePrice')))
```

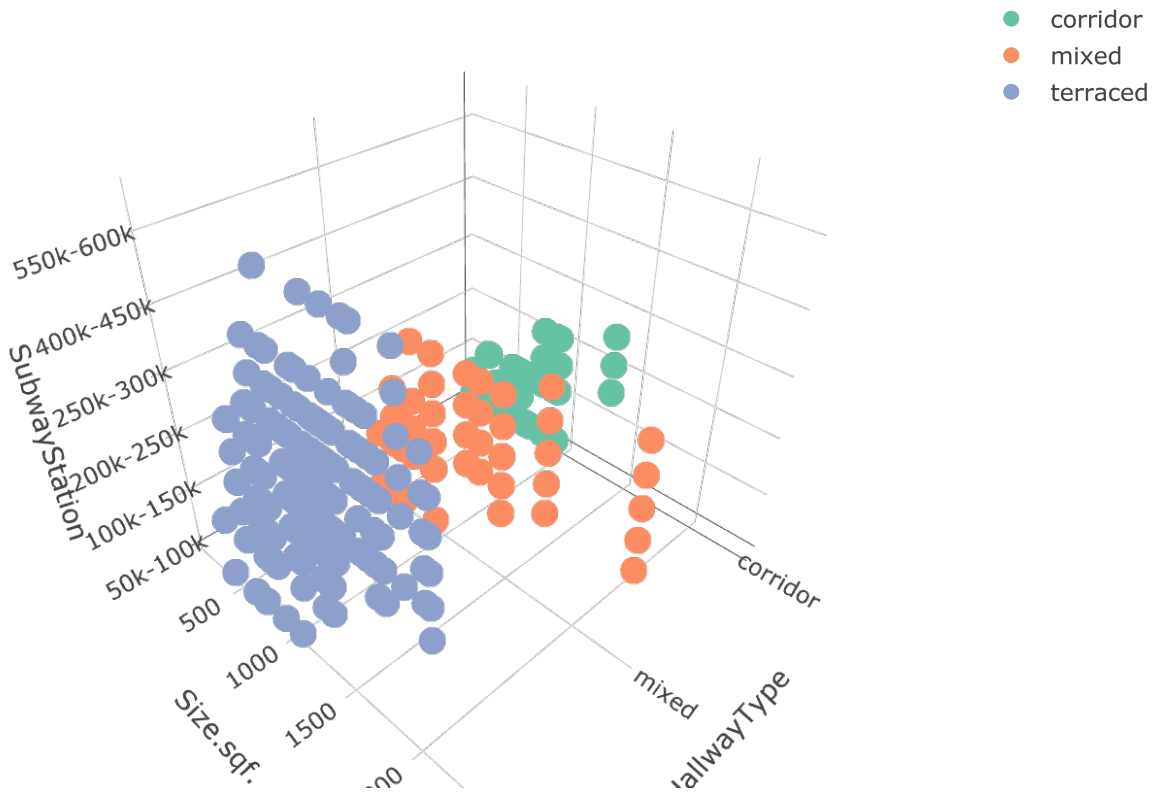
```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning re
quested palette with 3 different levels

## Warning in RColorBrewer::brewer.pal(N, "Set2"): minimal value for n is 3, returning re
quested palette with 3 different levels
```



From the above scatter plot it became evident that aptManageType is not very significant to the saleprice of apartment as it seems to be highly controlled by squere feet area thus AptManageType can be eleiminated.

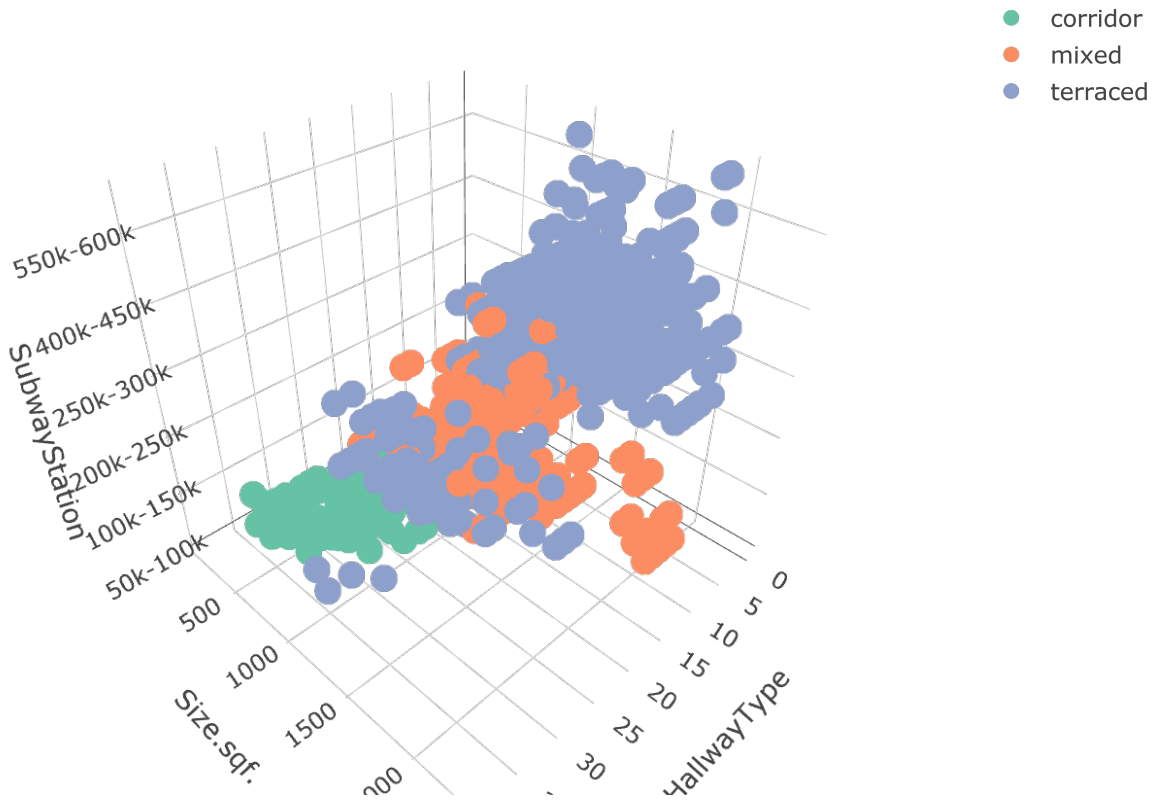
Outcome 15 : eliminating variable AptManageType on the ground of irrelevancy

```
plot_ly(apartment, x = apartment$HallwayType , y = apartment$Size.sqf., z = apartment$Sal
ePrice, color = apartment$HallwayType) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'HallwayType'),
                      yaxis = list(title = 'Size.sqf.'),
                      zaxis = list(title = 'SubwayStation')))
```



In the above scatter plot, a signifact relationship is identified and it seems clear that apartment with corridor are cheaper and smaller where apartment with mixed type has very random impact. Apartment with terrece are costlier in comparison.

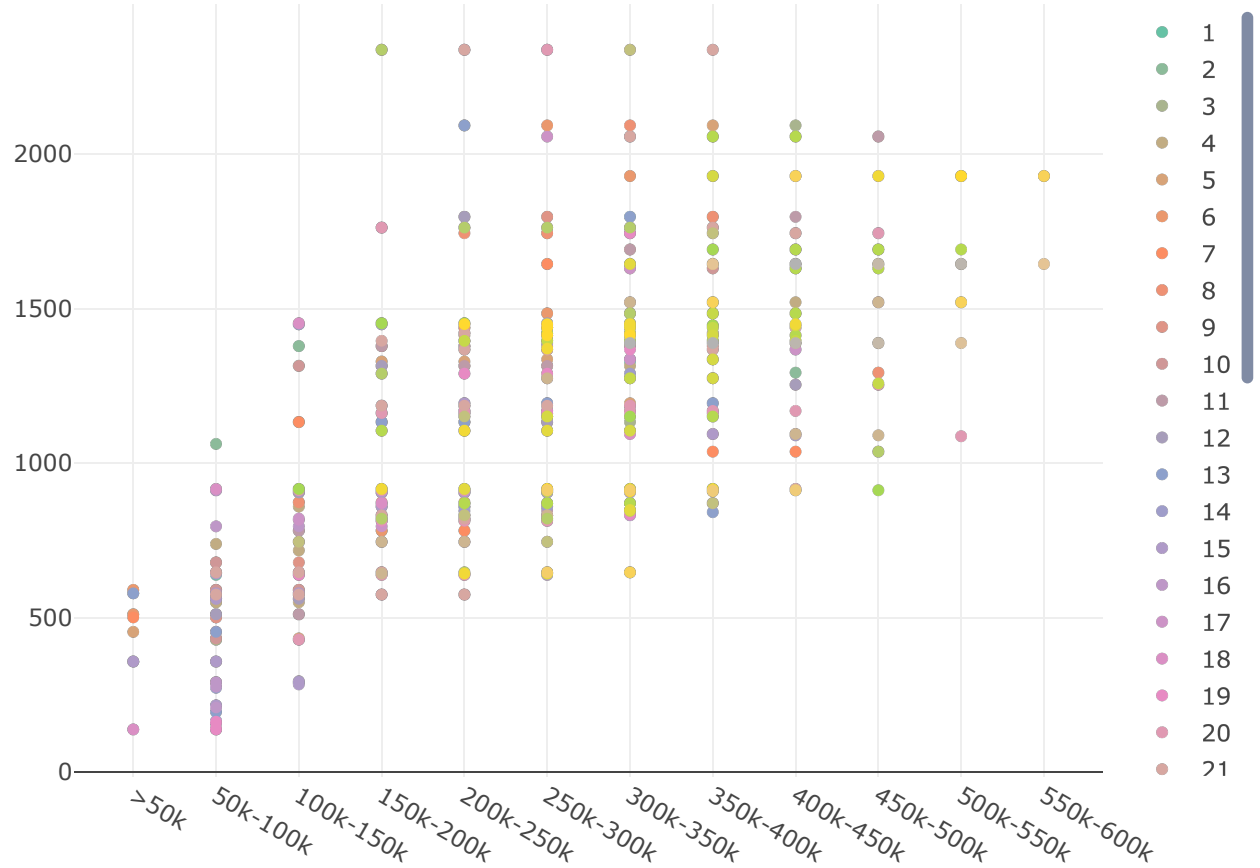
```
plot_ly(apartment, x = apartment$AgeWhenSold , y = apartment$Size.sqf., z = apartment$Sal
ePrice, color = apartment$HallwayType) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'HallwayType'),
                      yaxis = list(title = 'Size.sqf.'),
                      zaxis = list(title = 'SubwayStation')))
```



From above plot it became clear that all the apartments which are old has corridor and they are comparitively smaller and cheaper. on the other hand, all new apartments has terraced and are mostly costlier.

```
plot_ly(apartment, x = apartment$SalePrice , y = apartment$Size.sqf., color = apartment$F
loor) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'SalePrice'),
                      yaxis = list(title = 'Size.sqf.')))
```

```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for palet
te Set2 is 8
## Returning the palette you asked for with that many colors

## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for palet
te Set2 is 8
## Returning the palette you asked for with that many colors
```



From the above scatter plot it became evident that Floor is not very significant to the saleprice of apartment as it seems to be highly controlled by squere feet area and other features thus Floor can be eleiminated.

Outcome 16 : eliminating variable Floor on the ground of irrelevancy

Chapter 4

Final outcome of Data pre-processing

In the following piece of code the final dataframe is created which would be used for further analysis in phase II of this project.

Here, all the redudance and irrelevent features are dropped which are thinly or not related to the target feature.

```
apartment_final <- apartment[,-c(2,4,6,8,9,10,12,14,18,19,21,22,24,25,26,27)]
head(apartment_final)
```

	SalePrice <ord>	YrS… <fctr>	Size.sqf. <int>	HallwayType <fctr>	N_Parkinglot.Basement. <dbl>	TimeToSub… <ord>	N_
1	100k-150k	2007	814	terraced	184	10min~15min	
2	50k-100k	2007	587	corridor	76	5min~10min	
3	>50k	2007	587	corridor	76	5min~10min	
4	350k-400k	2007	2056	terraced	536	0-5min	
5	200k-250k	2007	1761	mixed	536	15min~20min	
6	>50k	2007	355	corridor	0	10min~15min	

6 rows | 1-8 of 16 columns

```
str(apartment_final)
```

```
## 'data.frame':    5891 obs. of  15 variables:
##  $ SalePrice                          : Ord.factor w/ 12 levels ">50k"<"50k-100k"<..:
3 2 1 8 5 1 2 2 2 2 ...
##  $ YrSold                             : Factor w/ 11 levels "2007","2008",..: 1 1 1 1
1 1 1 1 1 1 ...
##  $ Size.sqf.                          : int  814 587 587 2056 1761 355 644 644 644 64
4 ...
##  $ HallwayType                        : Factor w/ 3 levels "corridor","mixed",..: 3 1
1 3 2 1 2 2 2 2 ...
##  $ N_Parkinglot.Basement.             : num  184 76 76 536 536 0 79 536 536 79 ...
##  $ TimeToSubway                       : Ord.factor w/ 5 levels "no_bus_stop_nearb
y"<..: 3 4 4 5 2 3 2 2 2 2 ...
##  $ N_manager                          : num  3 2 2 5 8 5 4 8 8 4 ...
##  $ N_elevators                        : num  0 2 2 11 20 10 8 20 20 8 ...
##  $ SubwayStation                      : Factor w/ 8 levels "Bangoge","Banwoldang",..:
5 4 4 8 6 6 6 6 6 6 ...
##  $ N_FacilitiesNearBy.Dpartmentstore.: num  1 2 2 0 0 1 1 0 0 1 ...
##  $ N_FacilitiesNearBy.Park.           : num  0 1 1 0 0 1 0 0 0 0 ...
##  $ N_FacilitiesInApt                  : int  5 3 3 5 4 3 3 4 4 3 ...
##  $ N_FacilitiesNearBy.Total.          : num  6 12 12 3 14 16 9 14 14 9 ...
##  $ N_SchoolNearBy.Total.              : num  9 4 4 7 17 17 14 17 17 14 ...
##  $ AgeWhenSold                        : int  1 22 22 1 14 15 15 14 14 15 ...
```

```
colnames(apartment_final)
```

```
##  [1] "SalePrice"
##  [2] "YrSold"
##  [3] "Size.sqf."
##  [4] "HallwayType"
##  [5] "N_Parkinglot.Basement."
##  [6] "TimeToSubway"
##  [7] "N_manager"
##  [8] "N_elevators"
##  [9] "SubwayStation"
## [10] "N_FacilitiesNearBy.Dpartmentstore."
## [11] "N_FacilitiesNearBy.Park."
## [12] "N_FacilitiesInApt"
## [13] "N_FacilitiesNearBy.Total."
## [14] "N_SchoolNearBy.Total."
## [15] "AgeWhenSold"
```

This Data is in correct format which is desired for further analysis.

Chapter 5

Summary

In Phase I, we corrected the data format wherever required. We removed descriptive feature such as YearBuilt, N_Apt, N_FacilitiesNearBy.PublicOffice., N_FacilitiesNearBy.Hospital., N_FacilitiesNearBy.Mall., N_FacilitiesNearBy.ETC., N_SchoolNearBy.Elementary., N_SchoolNearBy.Middle., N_SchoolNearBy.High. and N_SchoolNearBy.University. on the basis of redundancy and we also dropped other descriptive features such as HeatingType, N_Parkinglot.Ground., TimeToBusStop, Floor, MonthSold and AptManageType on teh basis of irrelevancy. From the data exploration, we found that Size.sqf., TimeToSubway, SubwayStation, N_FacilitiesInApt and AgeWhenSold were potentially useful features in predicting the SalePrice category.

Chapter 6

Reference

[1] https://www.kaggle.com/gunhee/koreahousedata (https://www.kaggle.com/gunhee/koreahousedata) [2] https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf (https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf) [3] http://www.sthda.com/english/wiki/scatter-plot-matrices-r-base-graphs (http://www.sthda.com/english/wiki/scatter-plot-matrices-r-base-graphs) [4] https://plot.ly/r/3d-scatter-plots/ (https://plot.ly/r/3d-scatter-plots/) [5] https://stackoverflow.com (https://stackoverflow.com) [6] https://www.century21global.com/for-sale-residential/South-Korea (https://www.century21global.com/for-sale-residential/South-Korea)