To: The manager of the Ski Resort in Colorado

From: Chinmayi Suryakant Mahadik, Vedant Dashora, Yogesh Selvaraj Narayanan
Subject: Report on the factors affecting selling price of the property.
Date: 4/15/2020
A report to discuss the factors that are associated with the selling price of homes in a Colorado ski resort.

EXECUTIVE SUMMARY

Major Findings:

- The ski resort data tells that the selling price of the property is highly associated with a few factors like listing price of the property, miles from the property to the downtown and mountain area, size of the house (Sq_Ft) and size of the property(Lot Size).

- The listing price of the property is approximately the same as the selling price. Based on this we performed analysis with and as well as without listing price of the property.

- Based on regression analysis (multiple regression), when we consider listing price of the property, the selling price of the property can me most accurately predicted with the help of the proximity of the house to downtown in miles and listing price of the property

- Based on regression analysis when we do not consider listing price of the property, the selling price of the property can be predicted with help of the proximity of the house to the mountain area, the size of the house and property.

Recommendation for Action:

- The recommendation to predict the final selling price when listing price is considered is by using following equation:

  Selling price = 0.10949 + 0.98426(List price) - 1.67259(√Downtown)

  This states that if the list price of the property increased by $1000, the selling price of the property will increase by $984.26. If the square root distance from the downtown area increases by 1 mile, the selling price will decrease by $1672.59.

- To predict the final selling price when listing price is not considered by following equation:

  Selling price = 263.063 + 21.83 (Bedrooms) + 0.04(Sq_ft) -4.302(Mountain distance) + 4.08 (Lot size) +13.65 (Garage)
- This states that if number of bedroom is increased by1, the selling price increases by \$21.83k ; If size of house is increased to 1square feet, the selling price increases by \$40; If the mountain distance is increased by 1 mile, the selling price decreases by \$4302; If lot size is increased by 1 acre, the selling price increases by \$4.08k; if number of garages is increased to 1, the selling price increases by \$13650.

- The first model with listing price gives 98.14% variance in the selling price of the property whereas without listing price gives 85.57% variance in the selling price. We recommend using the model with listing price. But if the listing price is not available in the Business process then considering the Second model is the best suitable solution to predict Selling price value.

- The selling price of the property dependence on location and size of house and property.

<u>Analytical Overview:</u>

- Exploratory data analysis was first used on all the variables to determine their correlation with the selling price and to check normal distribution of all variables.
- To conduct a predictive model, we tried a stepwise regression approach to determine the selling price of the property while considering the list price of the property. To determine the selling price without considering the list price of the property we used the best -subset approach which estimates the best possible models using all possible combinations.
- All the assumptions were verified by plotting different graphs and summary results
- To validate the models, we tried k fold cross validation method to determine accuracy in prediction in both models.

APPENDIX

Process used in Data Analysis

- Data Checking
- Data Summarizing
- Handling Outliers
- Inferences from univariant charts
- Inferences from bi-variant charts
- Inferences from multivariate charts
- Model recognition from Stepwise Process
- Model recognition from Subset Process
    - A) Including List Price as Predictor Variable
    - B) Not Including List Price as Predictor Variable

Data Checking

In skiData there were no missing data or incorrect data. But while performing EDA, we realized there were few extreme outliers in some independent variables. The outliers are influential to the regression model hence it needs to be handled. More details of how we handled outliers are mentioned in "Handling outlier section".

Data Summarizing

With the help of the installed packages, we have used various functions in order to derive multiple conclusions that would help in further obtaining models to predict required variable.

Installed packages and the library functions used are shown below:
install.packages("tidyverse")
install.packages("funModeling")
install.packages("Hmisc")
install.packages ("corrplot")

library(tidyverse)
library(funModeling)
library(Hmisc)
library(corrplot)
library(readxl)

ski= read_excel("C:/Users/vedan/OneDrive/Desktop/Statistics/Homework/Assignment 5/ski.xlsx")
library(tidyverse)
library(funModeling)
library(Hmisc)

- **Glimpse** function is revealed the dimensions (Observations) and names of the variables in the dataset.

glimpse(ski)

```
Console C:/Users/vedan/AppData/Local/Temp/Temp1_Vedant Yogesh Chinmayi (1).zip/Assignment 2/Assignment Part 1/
> glimpse(ski)
Observations: 39
Variables: 12
$ `rty #`         <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 2...
$ Bedrooms        <dbl> 3, 3, 3, 3, 4, 3, 2, 3, 3, 2, 3, 3, 4, 3, 4, 2, 4, 3, 3, 3, 3, 4, 3, 4, 2, 3, 6, 3, 4, 2, 4, 5, 4, 4...
$ Bathrooms       <dbl> 2.00, 2.00, 2.50, 2.00, 2.50, 2.25, 1.00, 2.00, 2.00, 2.00, 2.75, 2.50, 2.50, 2.00, 2.00, 1.75, 1.00...
$ Sq_Ft           <dbl> 1771, 1213, 1884, 1922, 1858, 1948, 1200, 2002, 1160, 1300, 1816, 1774, 1700, 1400, 3250, 1040, 1348...
$ Downtwon        <dbl> 15, 5, 2, 1, 0, 1, 52, 2, 15, 3, 0, 7, 3, 2, 31, 5, 0, 20, 20, 0, 7, 7, 3, 10, 5, 0, 2, 20, 0, 5, 10...
$ Mountain        <dbl> 20, 1, 7, 6, 5, 6, 48, 7, 20, 2, 5, 2, 8, 7, 36, 10, 5, 15, 15, 5, 2, 2, 8, 15, 5, 5, 3, 15, 5, 10, ...
$ `Lot size`      <dbl> 0.230, 0.170, 0.180, 0.290, 0.520, 21.910, 40.000, 0.180, 2.520, 0.670, 0.100, 0.180, 0.250, 0.230, ...
$ Garage          <dbl> 2, 1, 2, 1, 2, 2, 2, 1, 0, 2, 2, 2, 2, 1, 2, 3, 0, 2, 0, 0, 2, 2, 1, 0, 0, 2, 0, 2, 2, 2, 1, 2, 2, 3...
$ Age             <dbl> 4, 5, 16, 80, 9, 20, 21, 18, 17, 67, 18, 3, 19, 14, 19, 26, 23, 15, 4, 20, 13, 4, 78, 19, 11, 20, 23...
$ `On market`     <dbl> 127, 98, 105, 103, 39, 403, 211, 126, 296, 158, 191, 150, 308, 228, 94, 126, 121, 173, 20, 54, 87, 1...
$ `Selling price` <dbl> 365.0, 400.0, 408.0, 410.0, 448.0, 545.0, 315.0, 424.0, 326.0, 386.3, 389.0, 415.0, 437.0, 339.0, 34...
$ `List price`    <dbl> 368.0, 406.0, 418.0, 417.0, 457.5, 550.0, 335.0, 429.0, 365.0, 399.5, 397.0, 421.0, 445.0, 339.5, 34...
> |
```

- **df_status** function identified the type of values in the attributes and the unique numbers present in it.

df_status(ski)

```
Console C:/Users/vedan/AppData/Local/Temp/Temp1_Vedant Yogesh Chinmayi (1).zip/Assignment 2/Assignment I
> df_status(ski)
        variable q_zeros p_zeros q_na p_na q_inf p_inf    type unique
1          rty #       0    0.00    0    0     0     0 numeric     39
2       Bedrooms       0    0.00    0    0     0     0 numeric      5
3      Bathrooms       0    0.00    0    0     0     0 numeric     10
4          Sq_Ft       0    0.00    0    0     0     0 numeric     38
5       Downtwon       7   17.95    0    0     0     0 numeric     13
6       Mountain       0    0.00    0    0     0     0 numeric     12
7       Lot size       0    0.00    0    0     0     0 numeric     29
8         Garage       8   20.51    0    0     0     0 numeric      5
9            Age       0    0.00    0    0     0     0 numeric     22
10     On market       0    0.00    0    0     0     0 numeric     37
11 Selling price       0    0.00    0    0     0     0 numeric     33
12    List price       0    0.00    0    0     0     0 numeric     35
> 
```

- **freq** function is used to give the number of times each value is repeated in the dataset. But unfortunately, it did not work as we had no categorical value recognized by software in the data set.

freq(ski)

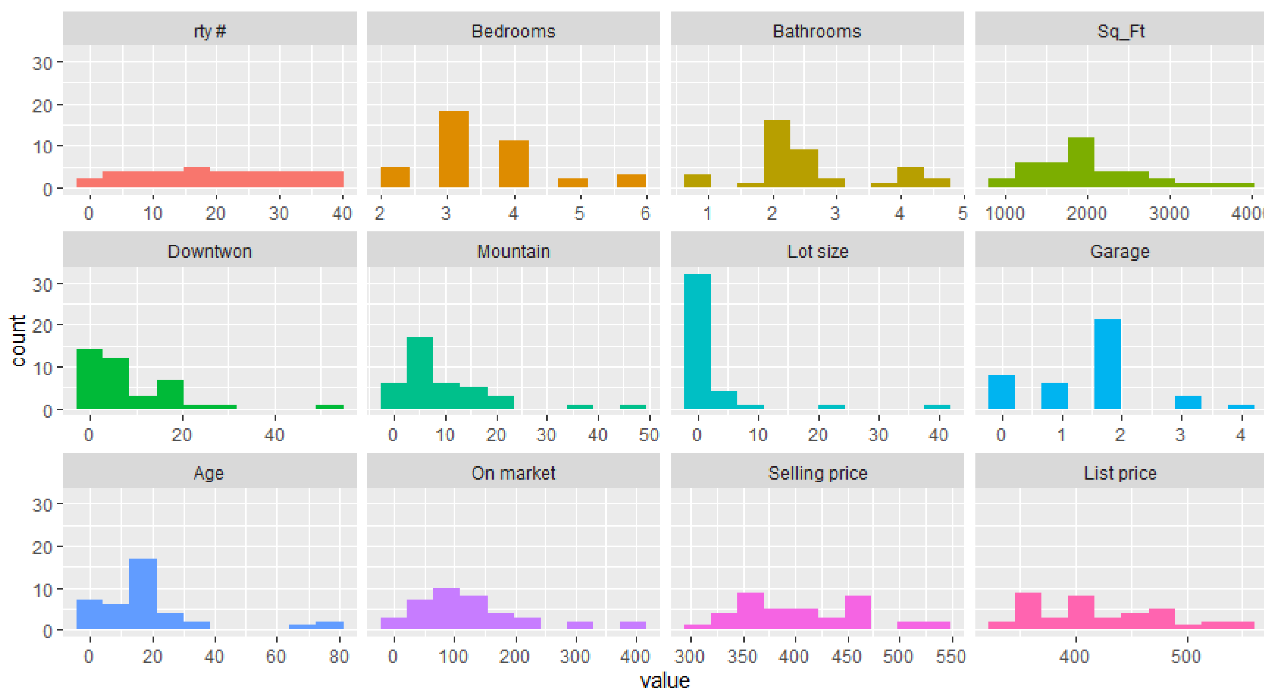
```
Console C:/Users/vedan/AppData/Local/Temp/Temp1_Vedant Yogesh Chinmayi (1).zip/Assignment 2/Assignment
> freq(ski)
NULL
Warning message:
In freq(ski) : None of the input variables are factor nor character
> 
```

- **profiling_num** function is used to get more detailed information on statistical summary of the dataset. Through this we obtained the skewness, kurtosis, and some characteristic values of each attribute, which helped in understanding behavior of data set.

```
Source
Console C:/Users/vedan/AppData/Local/Temp/Temp1_Vedant Yogesh Chinmayi (1).zip/Assignment 2/Assignment Part 1/
> profiling_num(ski)
        variable        mean     std_dev variation_coef     p_01      p_05    p_25    p_50     p_75     p_95      p_99
1          rty #   20.000000  11.4017543      0.5700877   1.3800    2.9000   10.50   20.00   29.500   37.100   38.6200
2       Bedrooms    3.487179   1.0481009      0.3005583   2.0000    2.0000    3.00    3.00    4.000    6.000    6.0000
3      Bathrooms    2.512821   0.9335640      0.3715204   1.0000    1.0000    2.00    2.25    2.625    4.050    4.6550
4          Sq_Ft 2003.794872 664.4719383      0.3316068 995.3600 1148.0000 1550.00 1922.00 2290.000 3282.000 3759.1000
5       Downtwon    8.692308  10.6774574      1.2283801   0.0000    0.0000    2.00    5.00   13.500   25.600   44.0200
6       Mountain    9.666667   9.3761549      0.9699471   1.3800    2.0000    5.00    7.00   12.500   21.600   43.4400
7       Lot size    2.596667   7.2502668      2.7921438   0.1038    0.1235    0.23    0.34    1.100   11.641   33.1258
8         Garage    1.564103   0.9945872      0.6358836   0.0000    0.0000    1.00    2.00    2.000    3.000    3.6200
9            Age   19.461538  18.0712714      0.9285634   3.0000    3.9000   10.00   16.00   20.500   68.100   79.2400
10     On market  131.000000  95.2067003      0.7267687  17.5200   20.9000   69.50  105.00  165.000  317.500  408.5800
11 Selling price  409.943590  58.7972242      0.1434276 319.1800  337.7000  362.75  400.00  458.250  521.000  539.3000
12    List price  420.443590  58.5369869      0.1392267 336.7100  348.8600  367.75  409.00  464.250  528.800  548.1000
     skewness  kurtosis     iqr              range_98                   range_80
1   0.0000000  1.798421  19.000         [1.38, 38.62]                [4.8, 35.2]
2   0.8678691  3.498607   1.000                [2, 6]                     [2, 5]
3   0.8040347  3.006134   0.625            [1, 4.655]                  [1.95, 4]
4   0.8633321  3.652949 740.000      [995.36, 3759.1]             [1210.4, 2794]
5   2.0895276  8.243992  11.500 [0, 44.0199999999999]                    [0, 20]
6   2.3954358  9.470721   7.500         [1.38, 43.44]                    [2, 20]
7   4.1781239 20.537496   0.870     [0.1038, 33.1258] [0.161, 3.46400000000001]
8  -0.1792244  2.682989   1.000             [0, 3.62]                   [0, 2.2]
9   2.2512647  7.825822  10.500            [3, 79.24]                    [4, 31]
10  1.3976895  4.771464  95.500       [17.52, 408.58]              [35.8, 241.6]
11  0.5247684  2.450196  95.500       [319.18, 539.3]                 [345, 477]
12  0.5827667  2.407303  96.500       [336.71, 548.1]             [359.58, 496.4]
>
```

- **plot_num** function is used to find frequency count of observations for a specific category/range of each variables through graphical representation of bar chart/histogram.

plot_num(ski)



- describe function is used to give tabular information in missing/distinct values in the dataset with its proportion percentage.

describe(ski)

#no missisng value found

Console C:/Users/vedan/AppData/Local/Temp/Temp1_Vedant Yogesh Chinmayi (1).zip/Assignment 2/Assignment Part 1/

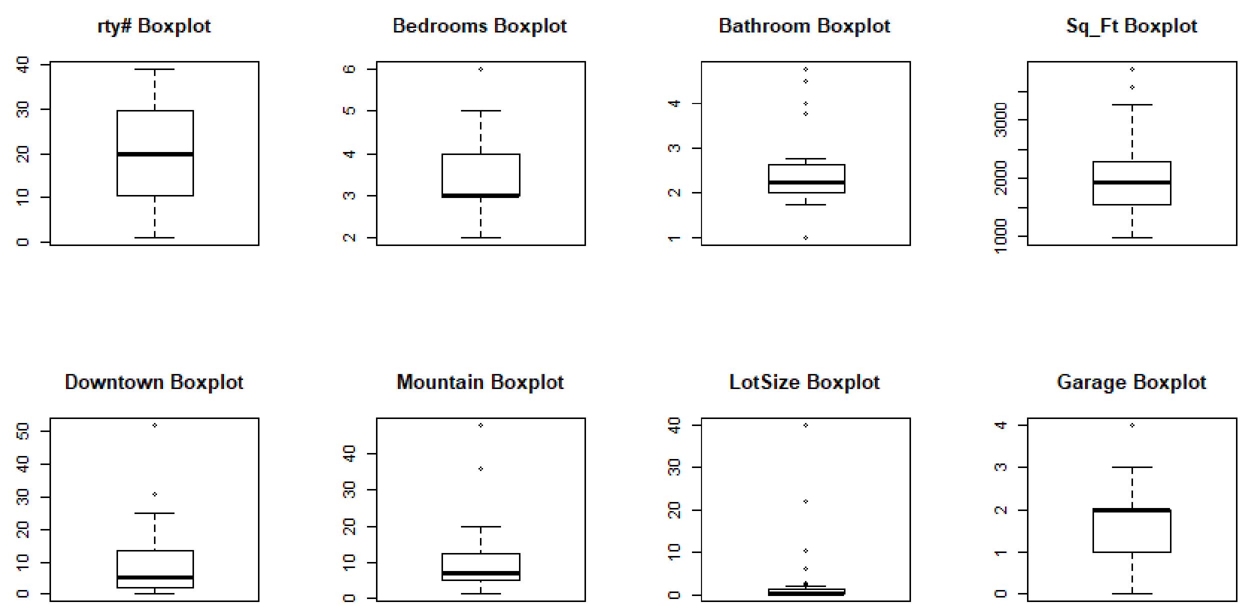
```
> describe(ski)
ski

 12  Variables      39  Observations
------------------------------------------------------------------------------------------------------------------------
rty #
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      39        0       39        1       20    13.33      2.9      4.8     10.5     20.0     29.5     35.2     37.1

lowest :  1  2  3  4  5, highest: 35 36 37 38 39
------------------------------------------------------------------------------------------------------------------------
Bedrooms
       n  missing distinct     Info     Mean      Gmd
      39        0        5    0.877    3.487    1.101

lowest : 2 3 4 5 6, highest: 2 3 4 5 6

Value          2     3     4     5     6
Frequency      5    18    11     2     3
Proportion 0.128 0.462 0.282 0.051 0.077
------------------------------------------------------------------------------------------------------------------------
Bathrooms
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      39        0       10    0.929    2.513   0.9838    1.000    1.950    2.000    2.250    2.625    4.000    4.050

lowest : 1.00 1.75 2.00 2.25 2.50, highest: 2.75 3.75 4.00 4.50 4.75

Value       1.00  1.75  2.00  2.25  2.50  2.75  3.75  4.00  4.50  4.75
Frequency      3     1    15     1     9     2     1     5     1     1
Proportion 0.077 0.026 0.385 0.026 0.231 0.051 0.026 0.128 0.026 0.026
------------------------------------------------------------------------------------------------------------------------
Sq_Ft
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      39        0       38        1     2004    733.9     1148     1210     1550     1922     2290     2794     3282

lowest :  968 1040 1160 1200 1213, highest: 2755 2950 3250 3570 3875
------------------------------------------------------------------------------------------------------------------------
Downtwon
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      39        0       13    0.987    8.692    10.45      0.0      0.0      2.0      5.0     13.5     20.0     25.6

lowest :  0  1  2  3  5, highest: 15 20 25 31 52

Value          0     1     2     3     5     7    10    12    15    20    25    31    52
Frequency      7     2     5     3     5     4     2     1     3     4     1     1     1
Proportion 0.179 0.051 0.128 0.077 0.128 0.103 0.051 0.026 0.077 0.103 0.026 0.026 0.026
------------------------------------------------------------------------------------------------------------------------
Mountain
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      39        0       12    0.981    9.667    8.764      2.0      2.0      5.0      7.0     12.5     20.0     21.6

lowest :  1  2  3  5  6, highest: 10 15 20 36 48

Value          1     2     3     5     6     7     8    10    15    20    36    48
Frequency      1     5     2     9     2     4     2     4     5     3     1     1
Proportion 0.026 0.128 0.051 0.231 0.051 0.103 0.051 0.103 0.128 0.077 0.026 0.026
------------------------------------------------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------------------------------------------------
Lot size
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      39        0       29    0.998    2.597    4.295   0.1235   0.1610   0.2300   0.3400   1.1000   3.4640  11.6410

lowest :  0.100  0.110  0.125  0.170  0.180, highest:  2.800  6.120 10.500 21.910 40.000

Value        0.1   0.2   0.3   0.4   0.5   0.7   0.8   1.0   1.2   1.3   1.9   2.5   2.8   6.1  10.5  21.9  40.0
Frequency      4    13     3     1     2     3     1     2     1     1     1     2     1     1     1     1     1
Proportion 0.103 0.333 0.077 0.026 0.051 0.077 0.026 0.051 0.026 0.026 0.026 0.051 0.026 0.026 0.026 0.026 0.026

For the frequency table, variable is rounded to the nearest 0.1
--------------------------------------------------------------------------------------------------------------------------
Garage
       n  missing distinct     Info     Mean      Gmd
      39        0        5    0.832    1.564    1.047

lowest : 0 1 2 3 4, highest: 0 1 2 3 4

Value          0     1     2     3     4
Frequency      8     6    21     3     1
Proportion 0.205 0.154 0.538 0.077 0.026
--------------------------------------------------------------------------------------------------------------------------
Age
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      39        0       22    0.995    19.46    16.61      3.9      4.0     10.0     16.0     20.5     31.0     68.1

lowest :  3  4  5  9 11, highest: 30 35 67 78 80
--------------------------------------------------------------------------------------------------------------------------
On market
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      39        0       37        1      131    100.2     20.9     35.8     69.5    105.0    165.0    241.6    317.5

lowest :  16  20  21  23  39, highest: 228 296 308 403 412
--------------------------------------------------------------------------------------------------------------------------
Selling price
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      39        0       33    0.999    409.9    67.17    337.7    345.0    362.8    400.0    458.2    477.0    521.0

lowest : 315 326 339 345 350, highest: 470 505 520 530 545
--------------------------------------------------------------------------------------------------------------------------
List price
       n  missing distinct     Info     Mean      Gmd      .05      .10      .25      .50      .75      .90      .95
      39        0       35        1    420.4    66.73    348.9    359.6    367.8    409.0    464.2    496.4    528.8

lowest : 335.0 339.5 349.9 357.9 360.0, highest: 490.0 522.0 527.0 545.0 550.0
--------------------------------------------------------------------------------------------------------------------------
```

Handling Outliers

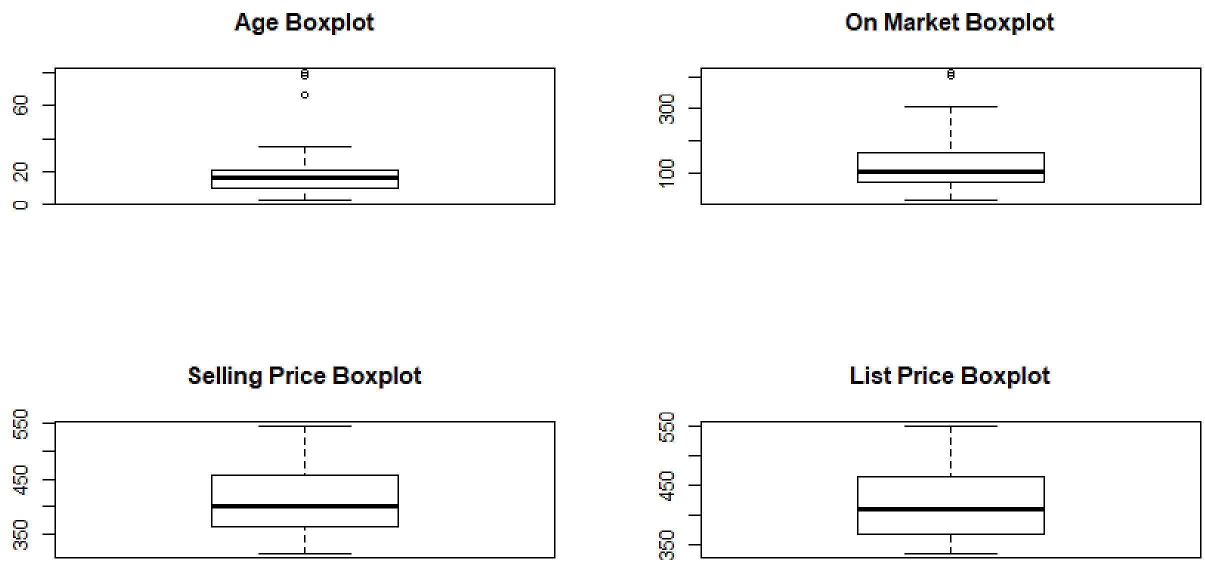
- After plotting box plot graphs for each variable, we found extreme outliers in Sq_ft , Bathroom and LotSize variables.
- Since the bathroom is discrete numeric data and has low correlations with selling price, we ignored those outliers.
- Whereas the effect of outliers in Sq_ft and LotSize were significant. We handled extreme outliers (ie. 3 * IQR) in excel before using them in R. Since, the data set is extremely small, eliminating rows containing outliers was not appreciated.
- For outliers in Sq_ft, we noticed they were found in rows that have a number of bedrooms as 6. The outliers were substituted by average value of Sq_ft with six bedrooms. For outliers in LotSize, mean value was substituted.

```
par(mfrow=c(2,4))
boxplot(ski$`rty #`, ma.in = "rty# Boxplot")
boxplot(ski$Bedrooms, main = "Bedrooms Boxplot")
boxplot(ski$Bathrooms, main = "Bathroom Boxplot")
boxplot(ski$Sq_Ft, main = "Sq_Ft Boxplot" )
boxplot(ski$Downtwon, main = "Downtown Boxplot")
boxplot(ski$Mountain, main = "Mountain Boxplot")
boxplot(ski$`Lot size`, main = "LotSize Boxplot" )
boxplot(ski$Garage, main = "Garage Boxplot")
```



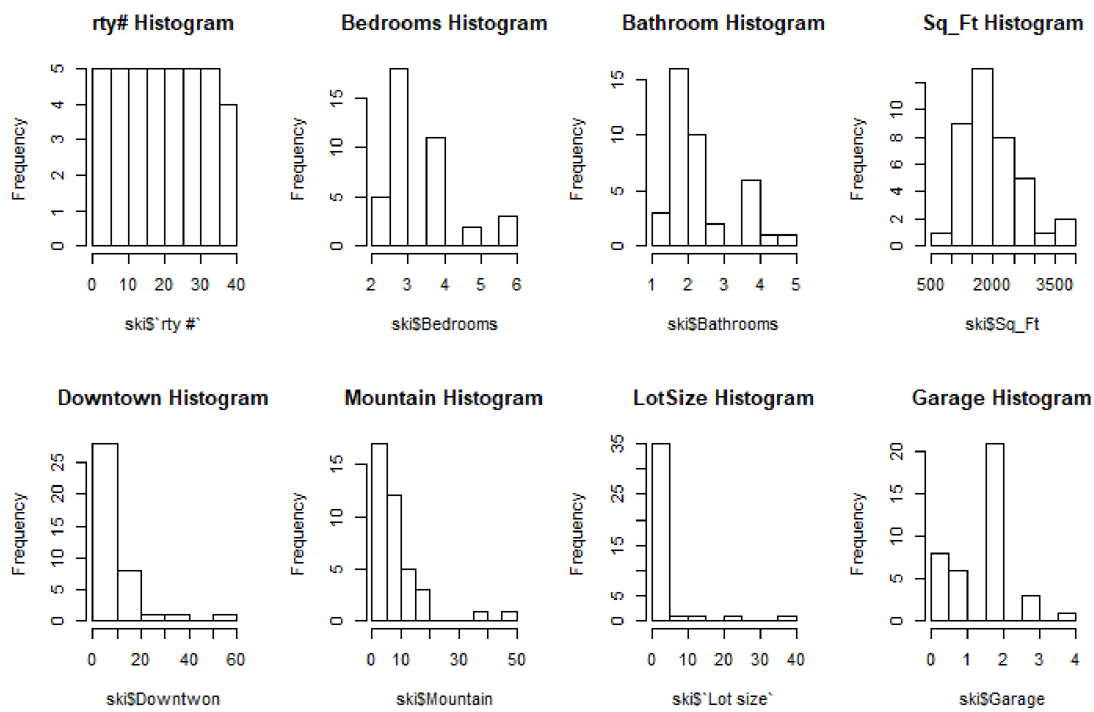

```
par(mfrow=c(2,2))
boxplot(ski$Age, main = "Age Boxplot")
boxplot(ski$`On market`, main = "On Market Boxplot" )
```

```
boxplot(ski$`Selling price`, main = "Selling Price Boxplot")
boxplot(ski$`List price`, main = "List Price Boxplot" )
```

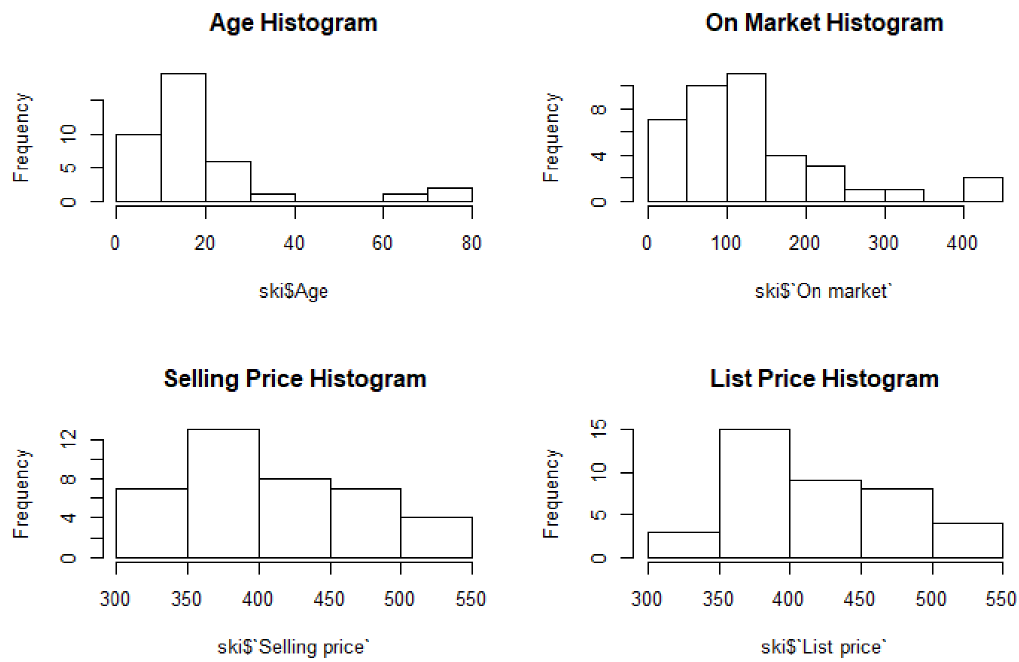


Univariant Analysis

```
par(mfrow=c(2,4))
hist(ski$`rty #`, main = "rty# Histogram")
hist(ski$Bedrooms, main = "Bedrooms Histogram")
hist(ski$Bathrooms, main = "Bathroom Histogram")
hist(ski$Sq_Ft, main = "Sq_Ft Histogram" )
hist(ski$Downtwon, main = "Downtown Histogram")
hist(ski$Mountain, main = "Mountain Histogram")
hist(ski$`Lot size`, main = "LotSize Histogram" )
hist(ski$Garage, main = "Garage Histogram")
```



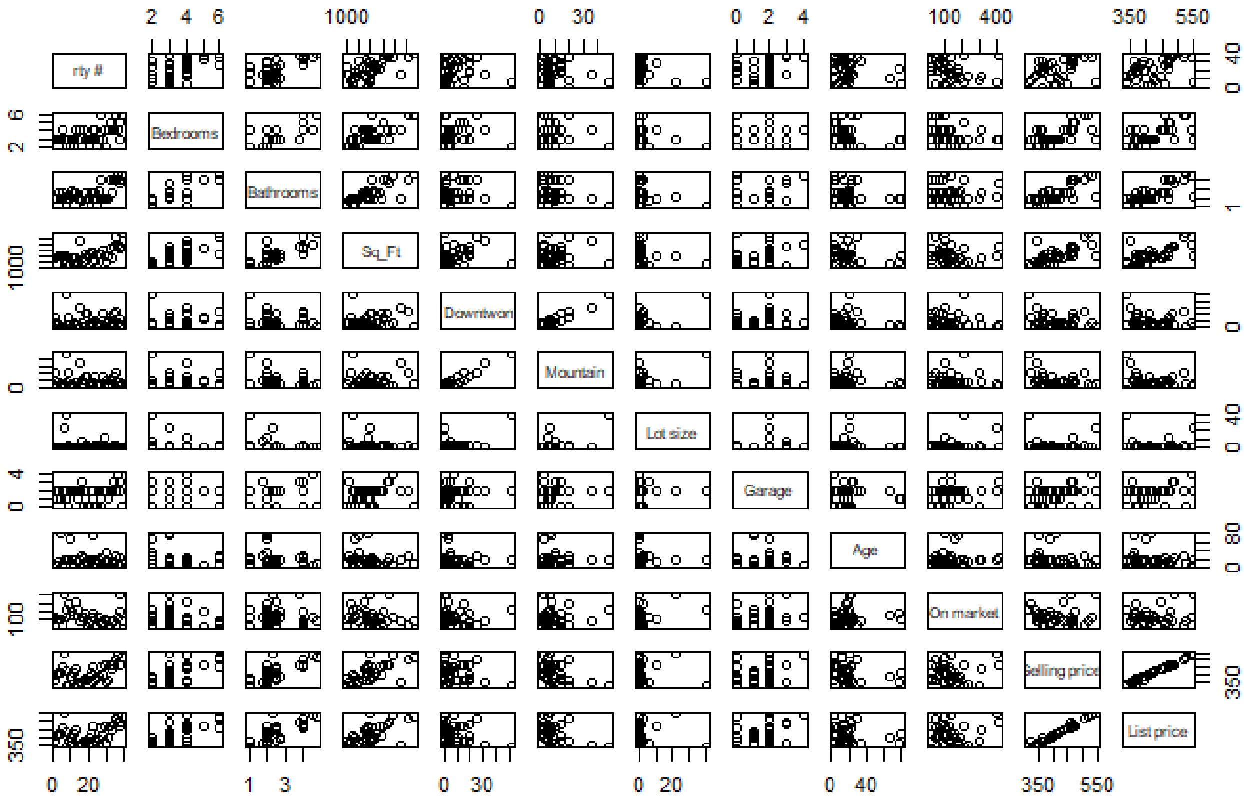

```
par(mfrow=c(2,2))

hist(ski$Age, main = "Age Histogram")
hist(ski$`On market`, main = "On Market Histogram" )
hist(ski$`Selling price`, main = "Selling Price Histogram")
hist(ski$`List price`, main = "List Price Histogram" )
```



Inferences from Univariate Graphs:

- From the Histogram representations, we can see that people prefer to buy houses with 3 bedrooms and 5 bedrooms had the least preference.
- For bathrooms, the most preferred number of bathrooms is 2 whereas the least preferred is 3.75.
- For Sq_ft, people buy houses with scales of between 1500-2000 sq.ft.
- The most preferred houses are near to the downtown vicinity between the radius of 5 miles.
- The most bought houses are 5 miles located from the mountain resorts.
- People buy houses with 0.2 - 0.5 acres of lot sizes.
- Customers buy houses with at least 2 garages and do not prefer with anything less than 2.
- Most of the houses that are bought are between 10- 20 years of Age and the least bought houses are of 40-60 years old.
- Most of the houses sold are on market tenure of between 100-150 days. After being on market for more than 250 days, it becomes difficult to sell those houses.
- Highest bought houses have the selling price between the range of $350k– $400k. Only a few sets of people afforded to buy houses priced more than $500k. Listing price is similar to selling price.

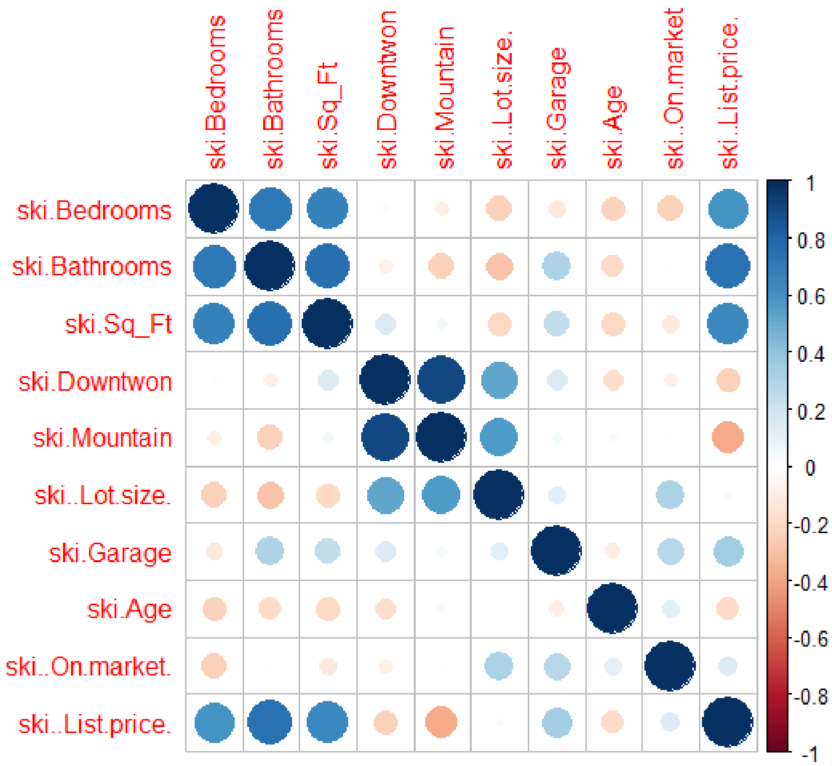
pairs(ski)



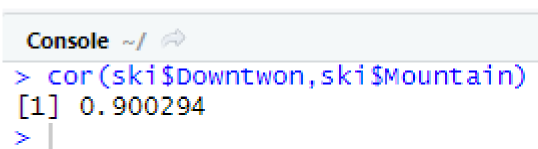
Correlation plot

library(corrplot)

```
#include all Predictors
SkiInfluential = data.frame(ski$Bedrooms,ski$Bathrooms, ski$Sq_Ft,
                    ski$Downtwon, ski$Mountain, ski$`Lot size`,
                    ski$Garage, ski$Age, ski$`On market`, ski$`List price`); SkiInfluential
SkiInfCor = cor(SkiInfluential)
corrplot(SkiInfCor)
```

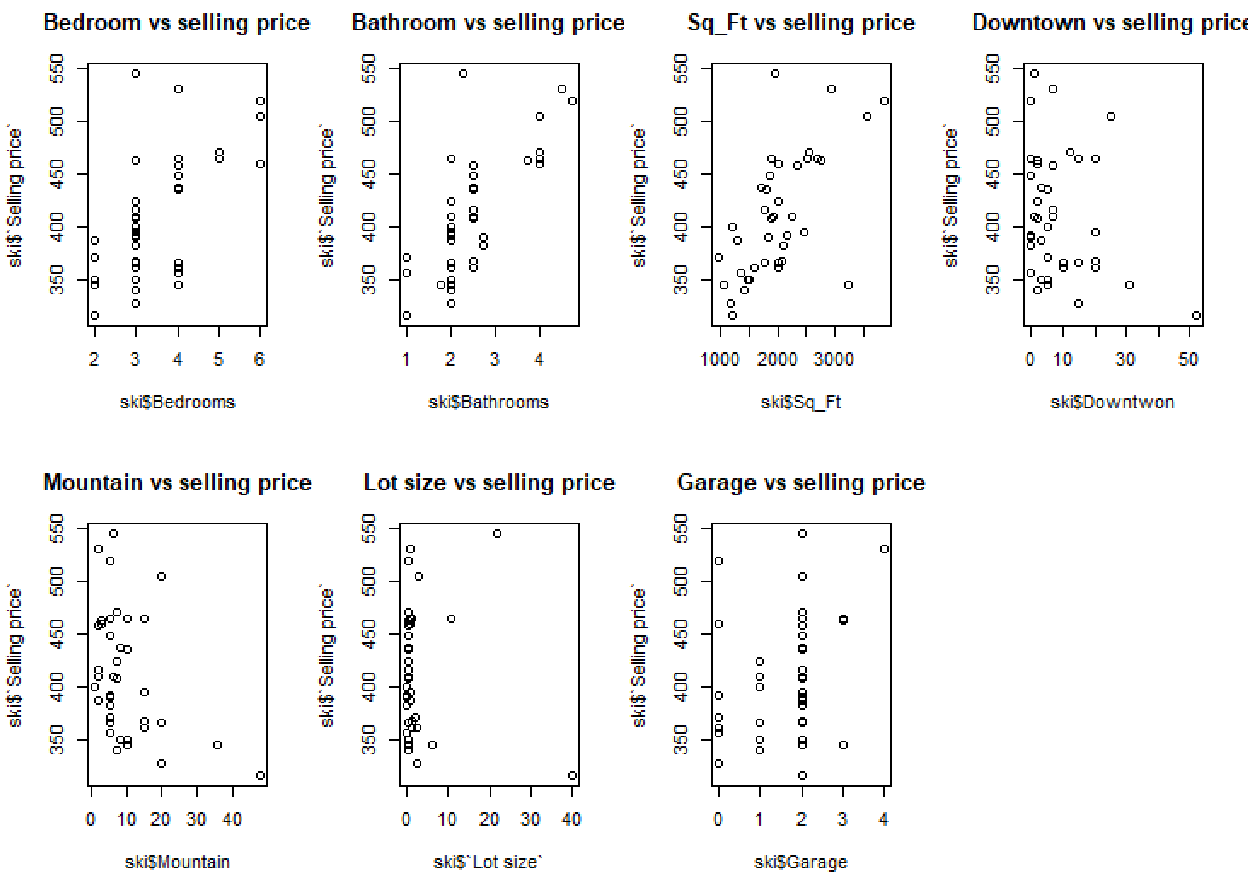


- From the model created from correlation value plot, we can see that mountain and downtown are the most correlated independent variables and thus, it has to be handled carefully while using them as an independent variable (Predictor variable) in obtaining a model.
- The correlation value is found to be 0.9 for mountain and downtown which is strong and can influence the model.



Bivariant Graphs

```
par(mfrow=c(2,4))
plot(ski$Bedrooms, ski$`Selling price`, main = “Bedroom vs selling price”)
plot(ski$Bathrooms, ski$`Selling price`, main = “Bathroom vs selling price”)
plot(ski$Sq_Ft, ski$`Selling price`, main = “Sq_Ft vs selling price”)
plot(ski$Downtwon, ski$`Selling price`, main = “Downtown vs selling price”)
plot(ski$Mountain, ski$`Selling price`, main = “Mountain vs selling price”)
plot(ski$`Lot size`, ski$`Selling price`, main = “Lot size vs selling price”)
plot(ski$Garage, ski$`Selling price`, main = “Garage vs selling price”)
```




```
par(mfrow=c(1,2))
plot(ski$Age, ski$`Selling price`, main = “Age vs selling price”)
plot(ski$`On market`, ski$`Selling price`, main = “On market vs selling price”)
```

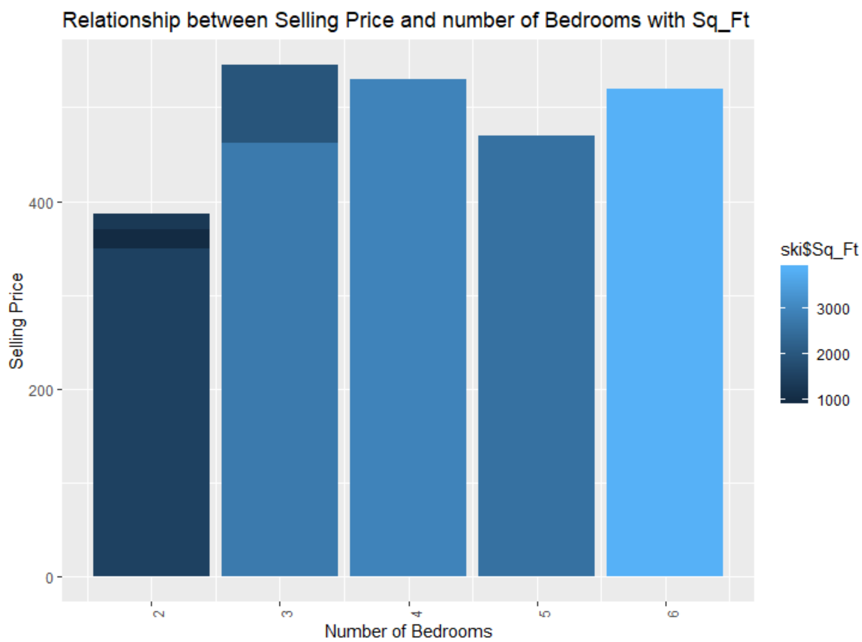
According to the plots from bivariate graphs, we can conclude the following:

- For all 3 bedroom and 2 bathrooms houses, the average selling price is set between $350k - $400k.
- With increase in square feet, the average of selling price also increases.
- The more distant a property from downtown, the selling price becomes low.
- If property is located far from the mountain, the selling price is lower.
- Lot sizes with 0.4-0.6 acres are the most commonly sold property costing around $400k apart from one outlier which costs $315k for 40 acres.
- Houses with two garages are sold most at different selling prices without a common range, thus it might be a bad predictor variable for model development.
- Age and on market do not have common range or pattern whereas houses aged between 0-40 years are sold at different selling prices and similarly houses which are on market from 0- 200 days range randomly.

<u>Multivariant Analysis</u>

Relationship between Selling Price and number of Bedrooms with Sq_Ft

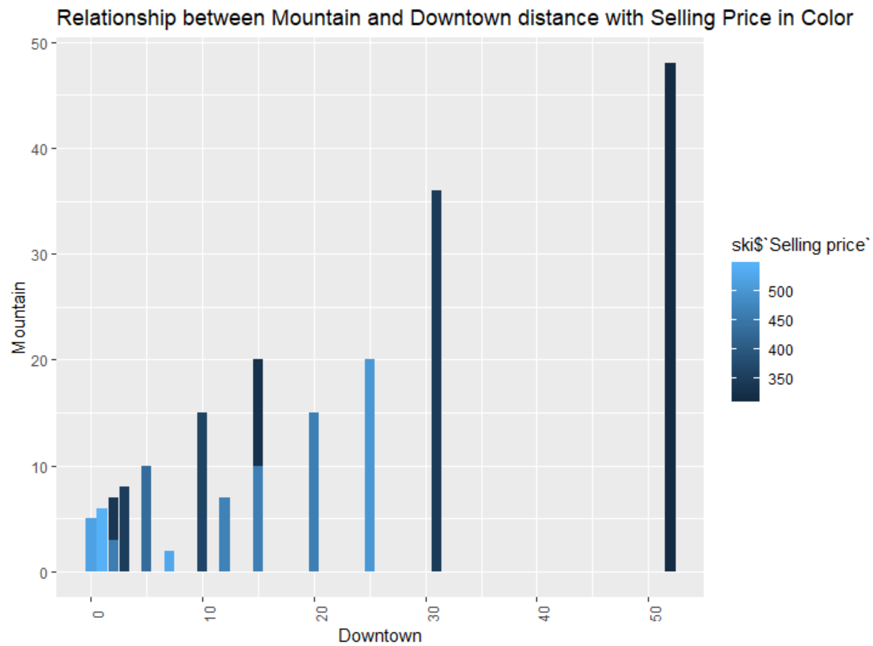
```
ggplot(ski, aes(x=ski$Bedrooms, y=ski$`Selling price`)) +
  geom_bar(aes(fill = ski$Sq_Ft), stat="identity",position=position_dodge()) +
  labs(title = "Relationship between Selling Price and number of Bedrooms with Sq_Ft",
       x= "Number of Bedrooms",
       y = "Selling Price",
       colour="Sq_Ft")+
  theme(axis.text.x = element_text(angle = 90))
```



- From the graph, we can see that lower the number of bedrooms and size of Square feet, lesser the selling price of the house. It has a linear relationship where the number of bedrooms and size of the house in Square feet is linear with the selling price.
- Higher the square feet size of the house, more the number of bedrooms in it.

Relationship between Mountain and Downtown distance with Selling Price in Color

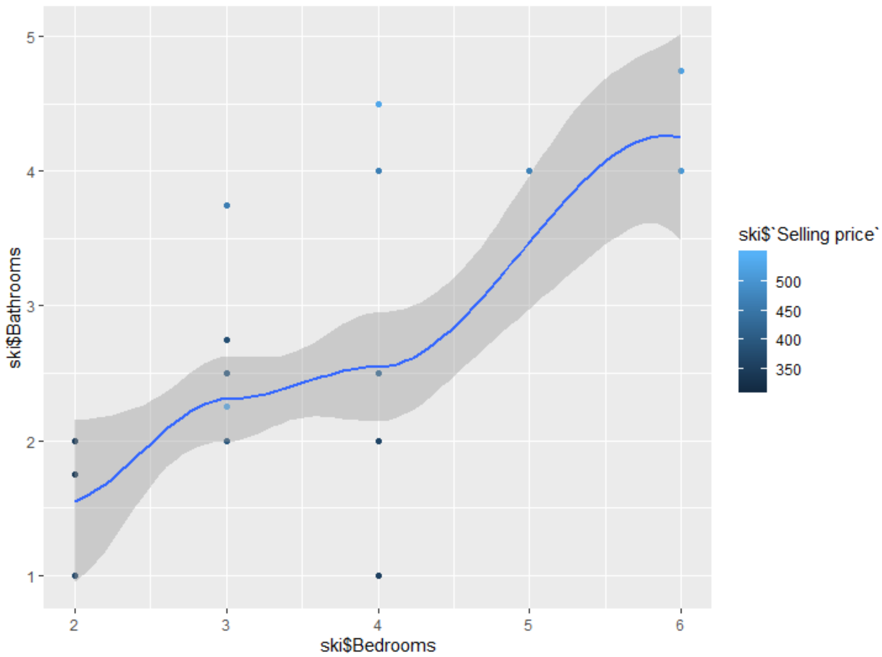
```
ggplot(ski, aes(x=ski$Downtwon, y=ski$Mountain)) +
  geom_bar(aes(fill = ski$`Selling price`), stat="identity",position=position_dodge()) +
  labs(title = "Relationship between Mountain and Downtown distance with Selling Price in
Color",
       x= "Downtown",
       y = "Mountain",
       colour="Selling price") +
  theme(axis.text.x = element_text(angle = 90))
```



- From the graph in which selling price is ranged in color, we can see that houses located far from downtown and mountain are priced low whereas properties which are located nearer to downtown and mountain are priced high.
- But as an exception, houses which are closest to the mountain and downtown (0-2 miles), are cheaper than houses which are located at a distance of 3-5 miles.
- We can see an outlier which is located 7 miles from downtown and 2 miles from mountain is priced extremely higher than other houses located at this range.

Relationship between bedroom and bathroom with selling price in Color

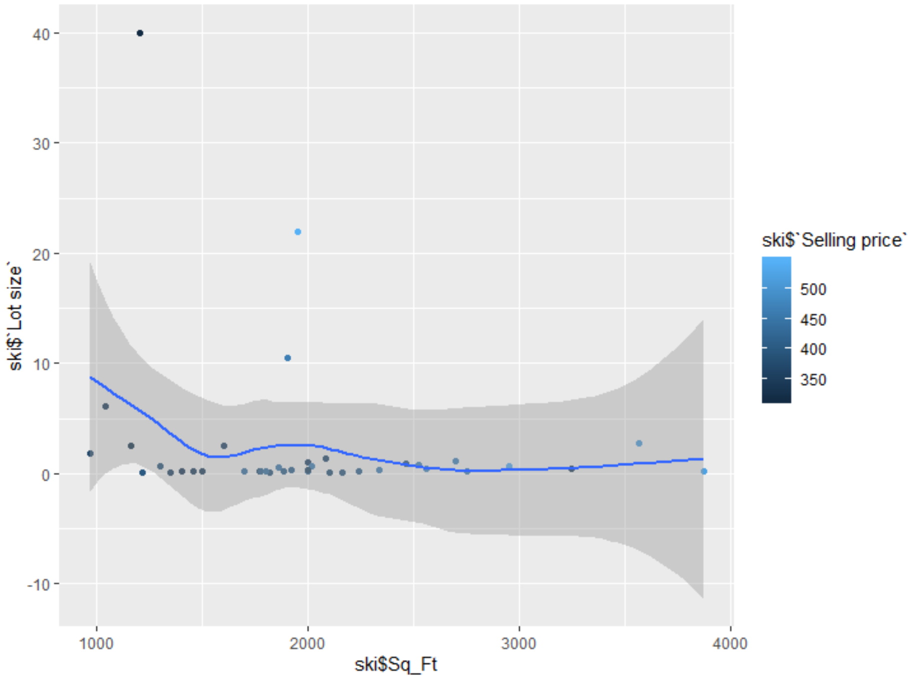
qplot(ski$Bedrooms,ski$Bathrooms, color = ski$`Selling price`,geom = c("point","smooth"))



- From the graph, we can see that as the number of bedrooms and bathrooms increases, the selling price also increases giving a linear relationship.
- There is an outlier where for the 3 bedrooms 2.5 bathrooms house, it is priced at more than $500k while the other houses in that range are priced around $300k.

Relationship between Lot size and Sq_ft with selling price in Color

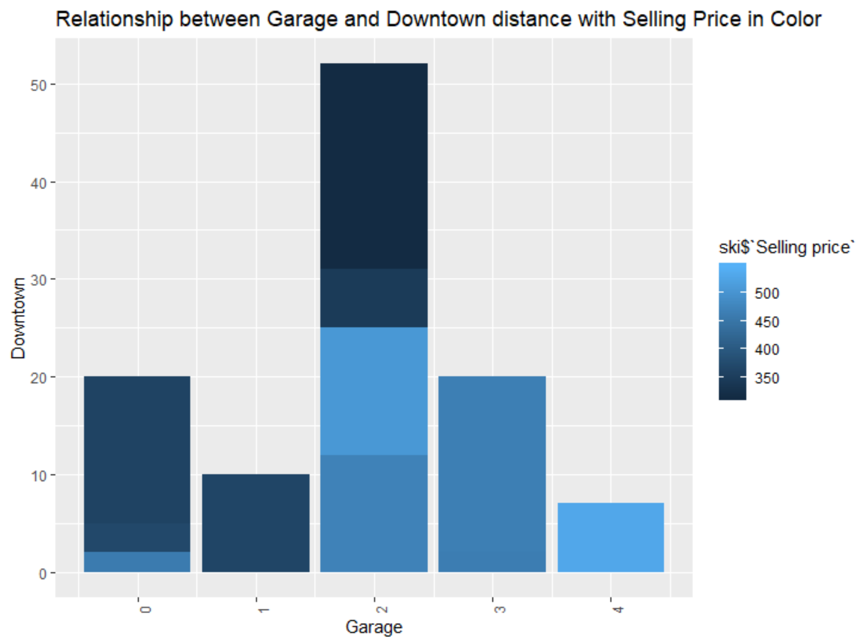
qplot(skiSq_Ft,ski`Lot size`, color = ski$`Selling price`,geom = c("point","smooth"))



- From the graph, higher the square feet the selling price keeps increasing for various prices of lot sizes in acres.

Relationship between Garage and Downtown distance with Selling Price in Color

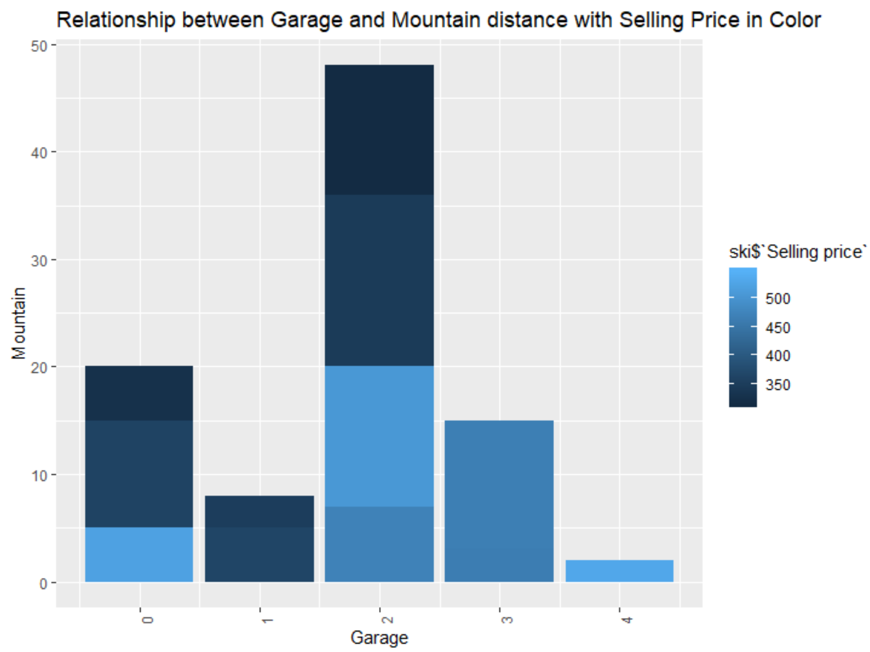
```
ggplot(ski, aes(x=ski$Garage, y=ski$Downtwon)) +
  geom_bar(aes(fill = ski$`Selling price`), stat="identity",position=position_dodge()) +
  labs(title = "Relationship between Garage and Downtown distance with Selling Price in Color",
       x= "Garage",
       y = "Downtown",
       colour="Selling price") +
  theme(axis.text.x = element_text(angle = 90))
```



- From the graph, we can observe that houses having two garage and far from downtown have the lowest selling prices and maximum frequency of this behavior is noticed.
- Also, the houses which are near to the downtown and have no garage space have very low value in selling price. The house with the highest selling price has 4 garages and nearest to downtown.
- The average $400k-$450k houses have 3 garages and from a distance of 0-20 miles.

Relationship between Garage and Mountain distance with Selling Price in Color

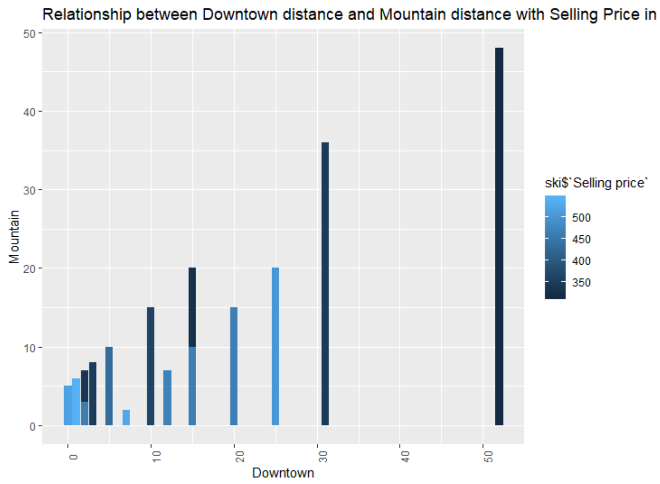
```
qplot(ski$Garage,ski$Mountain,color = ski$`Selling price`,geom = c("point","smooth"))

ggplot(ski, aes(x=ski$Garage, y=ski$Mountain)) +
  geom_bar(aes(fill = ski$`Selling price`), stat="identity",position=position_dodge()) +
  labs(title = "Relationship between Garage and Mountain distance with Selling Price in Color",
       x= "Garage",
       y = "Mountain",
       colour="Selling price") +
  theme(axis.text.x = element_text(angle = 90))
```



- From the graph, it is observed that distance from 20 -35 miles from mountain and having 2 garages are cheaper at $250k- $300k in selling prices.
- It is cheapest when it increased from 35 miles from the mountain.
- Having no garages but closer to the mountain have a selling price of more than $500k.

Relationship between Downtown distance and Mountain distance with Selling Price in Color

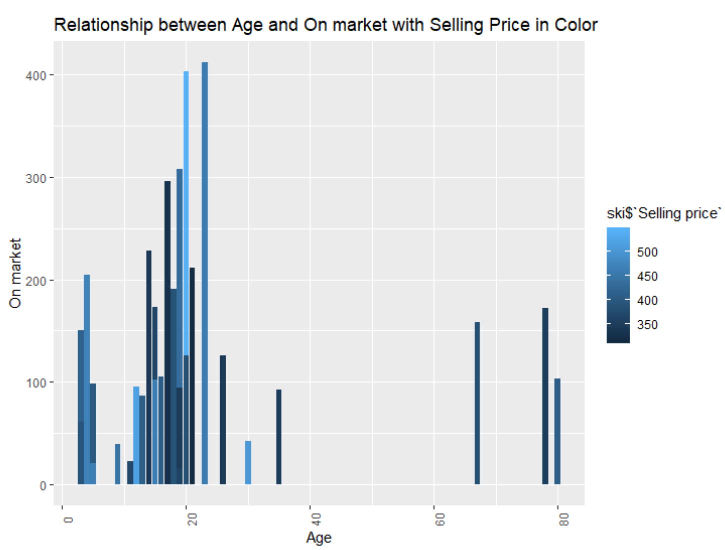
```
qplot(ski$Downtwon,ski$Mountain,color = ski$`Selling price`,geom = c(“point”,”smooth”))

ggplot(ski, aes(x=ski$Downtwon, y=ski$Mountain)) +
  geom_bar(aes(fill = ski$`Selling price`), stat=”identity”,position=position_dodge()) +
  labs(title = “Relationship between Downtown distance and Mountain distance with Selling
Price in Color”,
       x= “Downtown”,
       y = “Mountain”,
       colour=”Selling price”) +
  theme(axis.text.x = element_text(angle = 90))
```



- This is a combination of the above two graphs, where we can see that the house which is beyond 50 miles from downtown and mountain has the lowest selling price (below $350k).
- We have an outlier where a house located at 25 miles from Downtown and 20 miles from mountain but costs higher at $500k.
- Also, the houses, nearly 3 miles away from downtown and 6 miles away from the mountain are cheaper than houses at its range.

Relationship between Age and On market with Selling Price in Color

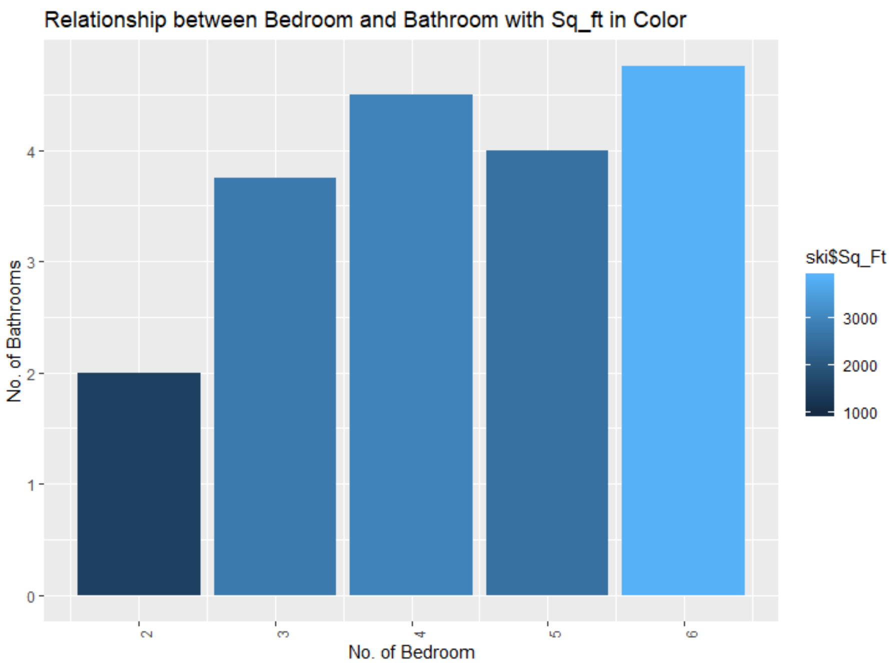
```
ggplot(ski, aes(x=ski$Age, y=ski$`On market`)) +
  geom_bar(aes(fill = ski$`Selling price`), stat=”identity”,position=position_dodge()) +
  labs(title = “Relationship between Age and On market with Selling Price in Color”,
       x= “Age”,
       y = “On market”,
       colour=”Selling price”) +
  theme(axis.text.x = element_text(angle = 90))
```



- We can see that there is no particular relationship between these variables and there are no patterns found between the age of the house and its time on market.

Relationship between Bedroom and Bathroom with Sq_ft in Color

```
ggplot(ski, aes(x=ski$Bedrooms, y=ski$Bathrooms)) +
  geom_bar(aes(fill = ski$Sq_Ft), stat="identity",position=position_dodge()) +
  labs(title = "Relationship between Bedroom and Bathroom with Sq_ft in Color",
       x= "No. of Bedroom",
       y = "No. of Bathrooms",
       colour="Sq_ft Range") +
  theme(axis.text.x = element_text(angle = 90))
```



- From the graph, the number of bedrooms and number of bathrooms are directly proportional to the size of square feet of the house. Greater the number of bedrooms and bathrooms, larger the square feet.

Obtaining Model from Subset Method for Predicting Selling Price

A) Considering List Price as one of the Independent Variables (Predictor) in the Model.

install.packages("caret")
install.packages("leaps")
library(tidyverse)
library(caret)
library(leaps)

#considering Selling price as dependent variable and all other as independent

modelSub = lm(ski$`Selling price` ~ ski$`List price` + ski$Bedrooms
    + ski$Bathrooms + ski$Sq_Ft + ski$Downtwon
    + ski$Mountain + ski$`Lot size`
    + ski$Garage + ski$Age + ski$`On market`, data = ski)

summary(modelSub)

```
Console ~/
> summary(modelSub)

Call:
lm(formula = ski$`Selling price` ~ ski$`List price` + ski$Bedrooms +
    ski$Bathrooms + ski$Sq_Ft + ski$Downtwon + ski$Mountain +
    ski$`Lot size` + ski$Garage + ski$Age + ski$`On market`,
    data = ski)

Residuals:
     Min       1Q   Median       3Q      Max
-22.2959  -3.0547  -0.7773   5.8168  14.2653

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      17.9287440 17.6660885   1.015   0.3189
ski$`List price`  0.9067751  0.0599607  15.123 5.32e-15 ***
ski$Bedrooms      2.2628061  2.6483800   0.854   0.4001
ski$Bathrooms     0.1529843  3.1595113   0.048   0.9617
ski$Sq_Ft         0.0042022  0.0041864   1.004   0.3241
ski$Downtwon     -0.6280715  0.3427051  -1.833   0.0775 .
ski$Mountain     -0.0004277  0.4496518  -0.001   0.9992
ski$`Lot size`    0.5598171  0.3490627   1.604   0.1200
ski$Garage        1.3949132  1.9058533   0.732   0.4703
ski$Age          -0.0551863  0.0811600  -0.680   0.5021
ski$`On market`  -0.0231021  0.0175786  -1.314   0.1994
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.347 on 28 degrees of freedom
Multiple R-squared:  0.9852,    Adjusted R-squared:  0.9798
F-statistic: 185.8 on 10 and 28 DF,  p-value: < 2.2e-16
```

#The independent variable found to be less significant
#analysis of variance table found less significant independent variables

```
> anova(modelsub)
Analysis of Variance Table

Response: ski$`Selling price`
                  Df Sum Sq Mean Sq   F value  Pr(>F)    
ski$`List price`   1 128737  128737 1847.8381 < 2e-16 ***
ski$Bedrooms       1     65      65    0.9296 0.34322    
ski$Bathrooms      1      9       9    0.1313 0.71986    
ski$Sq_Ft          1      4       4    0.0618 0.80550    
ski$Downtwon       1    303     303    4.3450 0.04636 *  
ski$Mountain       1      4       4    0.0613 0.80628    
ski$`Lot size`     1    111     111    1.5945 0.21709    
ski$Garage         1     31      31    0.4502 0.50775    
ski$Age            1     35      35    0.4990 0.48579    
ski$`On market`    1    120     120    1.7272 0.19944    
Residuals         28   1951      70                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

#Using regsubsets to find out the 10 best subset model to predict selling price

```
Submodels = regsubsets(ski$`Selling price` ~ ski$`List price`
                + ski$Bedrooms + ski$Bathrooms + ski$Sq_Ft
                + ski$Downtwon + ski$Mountain + ski$`Lot size`
                + ski$Garage + ski$Age + ski$`On market`, data = ski, nvmax = 10)
summary(Submodels)
```

```
Console ~/
> summary(Submodels)
Subset selection object
Call: regsubsets.formula(ski$`Selling price` ~ ski$`List price` + 
    ski$Bedrooms + ski$Bathrooms + ski$Sq_Ft + ski$Downtwon + 
    ski$Mountain + ski$`Lot size` + ski$Garage + ski$Age + 
    ski$`On market`, data = ski, nvmax = 10)
10 Variables  (and intercept)
                 Forced in Forced out
ski$`List price`     FALSE      FALSE
ski$Bedrooms         FALSE      FALSE
ski$Bathrooms        FALSE      FALSE
ski$Sq_Ft            FALSE      FALSE
ski$Downtwon         FALSE      FALSE
ski$Mountain         FALSE      FALSE
ski$`Lot size`       FALSE      FALSE
ski$Garage           FALSE      FALSE
ski$Age              FALSE      FALSE
ski$`On market`      FALSE      FALSE
1 subsets of each size up to 10
Selection Algorithm: exhaustive
          ski$`List price` ski$Bedrooms ski$Bathrooms ski$Sq_Ft ski$Downtwon ski$Mountain ski$`Lot size` ski$Garage ski$Age ski$`On market`
1  ( 1 )  "*"              " "          " "           " "       " "          " "          " "            " "        " "     " "            
2  ( 1 )  "*"              " "          " "           " "       "*"          " "          " "            " "        " "     " "            
3  ( 1 )  "*"              " "          " "           " "       "*"          " "          " "            " "        " "     "*"            
4  ( 1 )  "*"              " "          " "           " "       "*"          " "          "*"            " "        " "     "*"            
5  ( 1 )  "*"              " "          " "           "*"       "*"          " "          "*"            " "        " "     "*"            
6  ( 1 )  "*"              "*"          " "           "*"       "*"          " "          "*"            " "        " "     "*"            
7  ( 1 )  "*"              "*"          " "           "*"       "*"          " "          "*"            "*"        " "     "*"            
8  ( 1 )  "*"              "*"          " "           "*"       "*"          " "          "*"            "*"        "*"     "*"            
9  ( 1 )  "*"              "*"          "*"           "*"       "*"          " "          "*"            "*"        "*"     "*"            
10  ( 1 ) "*"              "*"          "*"           "*"       "*"          "*"          "*"            "*"        "*"     "*"            
> 
```

#considering parameters and finding best possible subset

```
Var_sum = summary(Submodels)
data.frame(
  Adj_R_Sq = which.max(Var_sum$adjr2),
  CP_Value = which.min(Var_sum$cp),
  BIC_Value = which.min(Var_sum$bic)
)
```

```
Console ~/
> #considering parameters and finding best possible subset
> Var_sum = summary(Submodels)
> data.frame(
+   Adj_R_Sq = which.max(Var_sum$adjr2),
+   CP_Value = which.min(Var_sum$cp),
+   BIC_Value = which.min(Var_sum$bic)
+ )
  Adj_R_Sq CP_Value BIC_Value
1        5        2         2
> 
```

#no substantial solution to the model found, each of these criteria will lead to slightly different models.

#adjusted R2 tells us that the best model has 5 predictor variables. But as per, BIC and Cp criteria, we should go for the model with 2 variables.

#Model with 5 variables

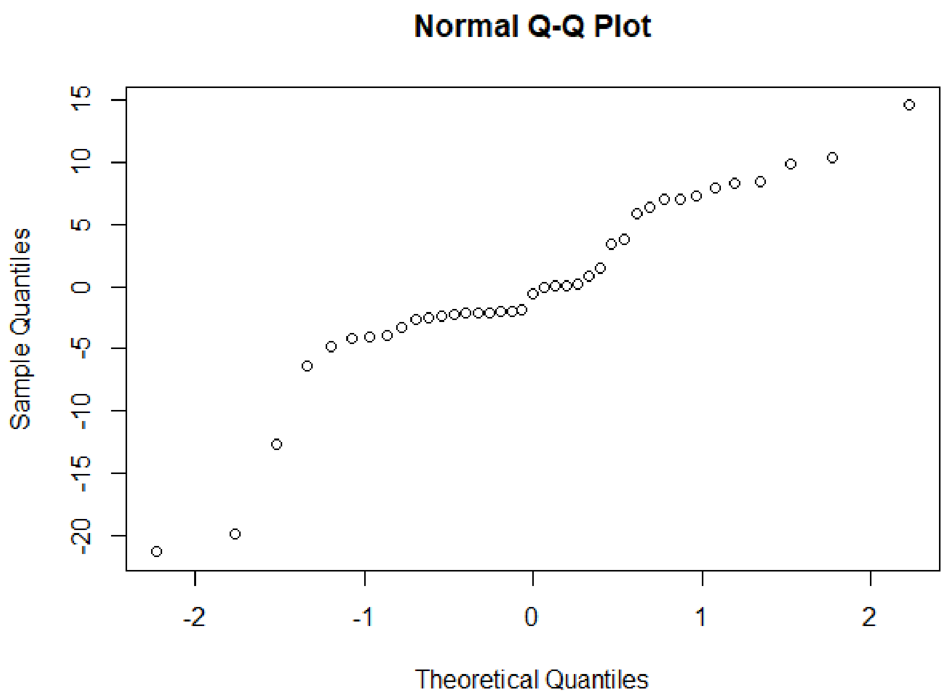
```
modelSub5Var = lm(ski$`Selling price` ~ ski$`List price` + ski$Sq_Ft
                  + ski$Downtwon + ski$`Lot size`
                  + ski$`On market`, data = ski)
summary(modelSub5Var)
anova(modelSub5Var)
library(car)
vif(modelSub5Var)
qqnorm(modelSub5Var$residuals)
AIC(modelSub5Var)
BIC(modelSub5Var)
```

```
Console ~/
> summary(modelSub5Var)

Call:
lm(formula = ski$`Selling price` ~ ski$`List price` + ski$Sq_Ft +
    ski$Downtwon + ski$`Lot size` + ski$`On market`, data = ski)

Residuals:
     Min       1Q   Median       3Q      Max
-21.3916  -2.5665  -0.5458   6.1540  14.6149

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       9.959120  11.868016   0.839   0.4074
ski$`List price`  0.943459   0.037781  24.972   <2e-16 ***
ski$Sq_Ft         0.004975   0.003460   1.438   0.1599
ski$Downtwon     -0.519804   0.197749  -2.629   0.0129 *
ski$`Lot size`    0.447494   0.277291   1.614   0.1161
ski$`On market`  -0.025186   0.014982  -1.681   0.1022
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.917 on 33 degrees of freedom
Multiple R-squared:  0.9843,    Adjusted R-squared:  0.9819
F-statistic: 412.6 on 5 and 33 DF,  p-value: < 2.2e-16
```

```
Console ~/
> anova(modelSub5Var)
Analysis of Variance Table

Response: ski$`Selling price`
                 Df Sum Sq Mean Sq   F value Pr(>F)
ski$`List price`  1 128737  128737 2054.0579 <2e-16 ***
ski$Sq_Ft         1      1       1    0.0197 0.8892
ski$Downtwon      1    306     306    4.8814 0.0342 *
ski$`Lot size`    1     81      81    1.2912 0.2640
ski$`On market`   1    177     177    2.8260 0.1022
Residuals        33   2068      63
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

```
Console ~/
> library(car)
> vif(modelSub5Var)
ski$`List price`        ski$Sq_Ft     ski$Downtwon   ski$`Lot size`  ski$`On market`
        2.965515         3.205170         2.703068         2.450604         1.233597
>
```

VIF value less than 5.



Console ~/

```
> AIC(modelSub5Var)
[1] 279.5423
> BIC(modelSub5Var)
[1] 291.1872
>
```

#Avoding largest r_sq value, and considering 2 variables

#But, get a less significant variable

#Also, according to the principle of Parsimony the least possible independent variable is best practice.

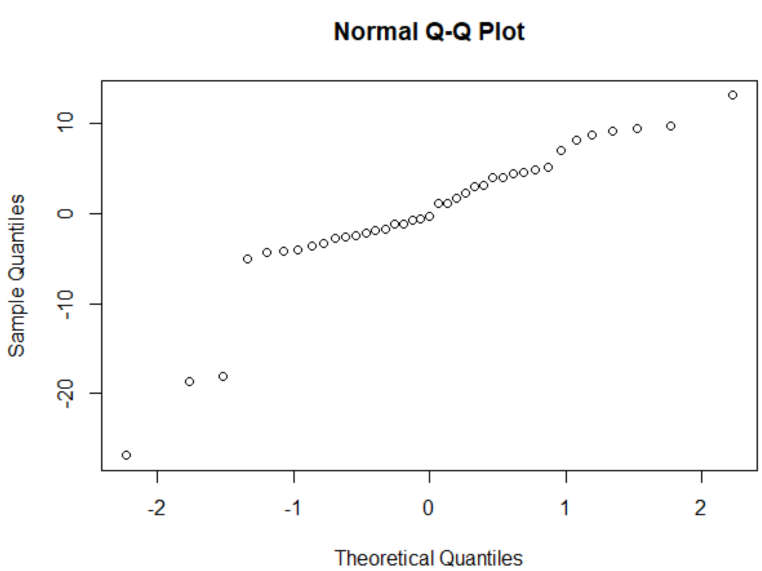
modelSub2Var = lm(ski$`Selling price` ~ ski$`List price` + ski$Downtwon, data = ski)
summary(modelSub2Var)

```
Console ~/
> modelSub2Var = lm(ski$`Selling price` ~ ski$`List price` + ski$Downtwon, data = ski)
> summary(modelSub2Var)

Call:
lm(formula = ski$`Selling price` ~ ski$`List price` + ski$Downtwon,
    data = ski)

Residuals:
     Min       1Q   Median       3Q      Max
-27.8521  -2.6534   0.0183   4.5037  12.4006

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      -1.70125   10.16087  -0.167   0.8680
ski$`List price`  0.98409    0.02322  42.380   <2e-16 ***
ski$Downtwon     -0.24258    0.12730  -1.906   0.0647 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.152 on 36 degrees of freedom
Multiple R-squared:  0.9818,	Adjusted R-squared:  0.9808
F-statistic: 970.5 on 2 and 36 DF,  p-value: < 2.2e-16
```

```
Console ~/
> anova(modelSub2Var)
Analysis of Variance Table

Response: ski$`Selling price`
                   Df Sum Sq Mean Sq   F value  Pr(>F)
ski$`List price`    1 128737  128737 1997.5467 < 2e-16 ***
sqrt(ski$Downtwon)  1    313     313    4.8623 0.03392 *
Residuals          36   2320      64
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

#Performing sqrt transformation to ski$Downtwon in order to increase significance

```
Console ~/
> modelSub2Varsqrt = lm(ski$`Selling price` ~ ski$`List price` + sqrt(ski$Downtwon), data = ski)
> summary(modelSub2Varsqrt)

Call:
lm(formula = ski$`Selling price` ~ ski$`List price` + sqrt(ski$Downtwon),
    data = ski)

Residuals:
     Min       1Q   Median       3Q      Max
-26.8876  -2.6539  -0.2958   4.4935  13.1539

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         0.10949   10.15172   0.011   0.9915
ski$`List price`    0.98426    0.02271  43.339   <2e-16 ***
sqrt(ski$Downtwon) -1.67259    0.75852  -2.205   0.0339 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.028 on 36 degrees of freedom
Multiple R-squared:  0.9823,    Adjusted R-squared:  0.9814
F-statistic:  1001 on 2 and 36 DF,  p-value: < 2.2e-16

> anova(modelSub2Varsqrt)
Analysis of Variance Table

Response: ski$`Selling price`
                   Df Sum Sq Mean Sq   F value  Pr(>F)
ski$`List price`    1 128737  128737 1997.5467 < 2e-16 ***
sqrt(ski$Downtwon)  1    313     313    4.8623 0.03392 *
Residuals          36   2320      64
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> vif(modelSub2Varsqrt)
  ski$`List price` sqrt(ski$Downtwon)
          1.042086           1.042086
> qqnorm(modelSub2Varsqrt$residuals)
> AIC(modelSub2Varsqrt)
[1] 278.0237
> BIC(modelSub2Varsqrt)
[1] 284.678
>
```



B) **Removing List Price as Independent Variable (Predictor) in the Model**

#The list price all the time found to be most significant as it has comparably good resemblance with selling price. Need to build a model depends on other factors.

#It will help determine how Selling prices rely on other factors than list price.

#Also, will help in predicting Selling Price if list price is not available

#Using regsubsets to find out the best subset model with each number of variables
#Not Considering List Price

```
Submodels2 = regsubsets(ski$`Selling price` ~ ski$Bedrooms
                + ski$Bathrooms + ski$Sq_Ft
                + ski$Downtwon + ski$Mountain + ski$`Lot size`
                + ski$Garage + ski$Age + ski$`On market`, data = ski, nvmax = 9)
summary(Submodels2)
```

```
Console ~/
> summary(Submodels2)
Subset selection object
Call: regsubsets.formula(ski$`Selling price` ~ ski$Bedrooms +
    ski$Bathrooms + ski$Sq_Ft + ski$Downtwon + ski$Mountain +
    ski$`Lot size` + ski$Garage + ski$Age + ski$`On market`,
    data = ski, nvmax = 10)
9 Variables  (and intercept)
                Forced in Forced out
ski$Bedrooms        FALSE      FALSE
ski$Bathrooms       FALSE      FALSE
ski$Sq_Ft           FALSE      FALSE
ski$Downtwon        FALSE      FALSE
ski$Mountain        FALSE      FALSE
ski$`Lot size`      FALSE      FALSE
ski$Garage          FALSE      FALSE
ski$Age             FALSE      FALSE
ski$`On market`     FALSE      FALSE
1 subsets of each size up to 9
Selection Algorithm: exhaustive
         ski$Bedrooms ski$Bathrooms ski$Sq_Ft ski$Downtwon ski$Mountain ski$`Lot size` ski$Garage ski$Age ski$`On market`
1  ( 1 ) " "          "*"           " "       " "          " "          " "            " "        " "     " "
2  ( 1 ) " "          "*"           " "       " "          "*"          " "            " "        " "     " "
3  ( 1 ) " "          " "           "*"       " "          "*"          "*"            " "        " "     " "
4  ( 1 ) " "          "*"           "*"       " "          "*"          "*"            " "        " "     " "
5  ( 1 ) "*"          " "           "*"       " "          "*"          "*"            "*"        " "     " "
6  ( 1 ) "*"          "*"           "*"       " "          "*"          "*"            "*"        " "     " "
7  ( 1 ) "*"          "*"           "*"       "*"          "*"          "*"            "*"        " "     " "
8  ( 1 ) "*"          "*"           "*"       "*"          "*"          "*"            "*"        "*"     " "
9  ( 1 ) "*"          "*"           "*"       "*"          "*"          "*"            "*"        "*"     "*"
> |
```

#considering parameters and finding best possible subset

```
Var_sum = summary(Submodels2)
data.frame(
  Adj_R_Sq = which.max(Var_sum$adjr2),
  CP_Value = which.min(Var_sum$cp),
  BIC_Value = which.min(Var_sum$bic)
)
```

```
> #considering parameters and finding best possible subset
> Var_sum = summary(Submodels2)
> data.frame(
+   Adj_R_Sq = which.max(Var_sum$adjr2),
+   CP_Value = which.min(Var_sum$cp),
+   BIC_Value = which.min(Var_sum$bic)
+
+ )
  Adj_R_Sq CP_Value BIC_Value
1        5        5         5
```

#interprets that the best r square, Cp and, BIC value found, is with 5 predictor variables, hence from summary(Submodels2)

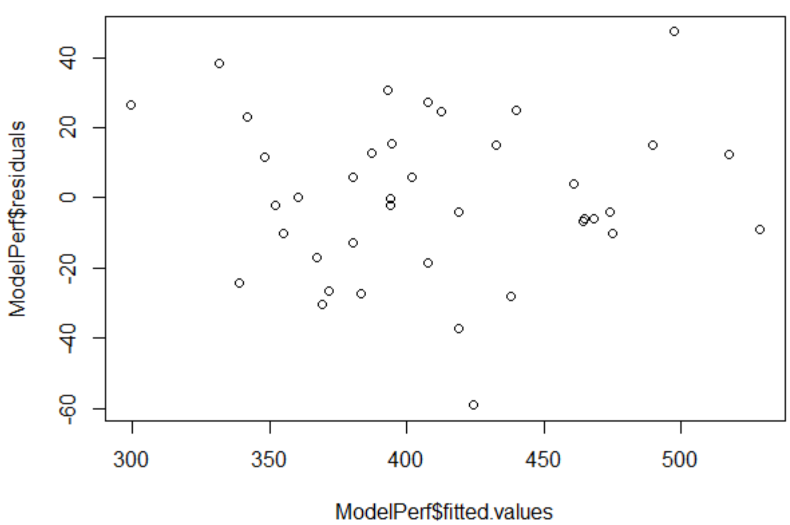
ModelPerf = lm(ski$`Selling price`~ ski$Bedrooms + ski$Sq_Ft + ski$Mountain + ski$`Lot size` + ski$Garage, data = ski)
summary(ModelPerf)
Anova(ModelPerf)
BIC(ModelPerf)
AIC(ModelPerf)
library(car)
vif(ModelPerf)
plot(ModelPerf$fitted.values,ModelPerf$residuals)
qqnorm(ModelPerf$residuals)

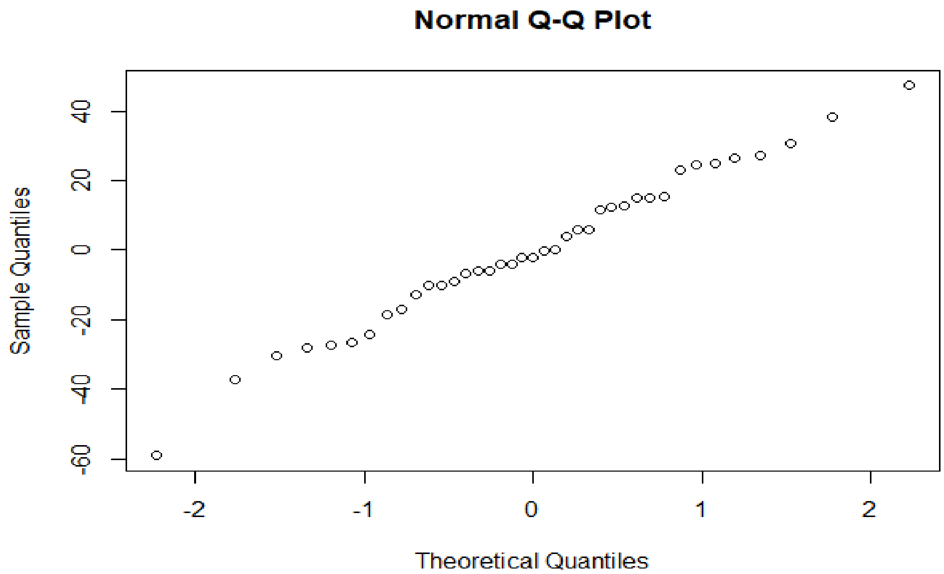
```
> summary(ModelPerf)

Call:
lm(formula = ski$`Selling price` ~ ski$Bedrooms + ski$Sq_Ft +
    ski$Mountain + ski$`Lot size` + ski$Garage, data = ski)

Residuals:
    Min      1Q  Median      3Q     Max
-59.104 -11.554  -2.188  15.104  47.318

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    263.063011  16.343425  16.096  < 2e-16 ***
ski$Bedrooms    21.836826   5.583632   3.911 0.000433 ***
ski$Sq_Ft        0.040099   0.009305   4.309 0.000139 ***
ski$Mountain    -4.302038   0.523829  -8.213 1.75e-09 ***
ski$`Lot size`   4.084563   0.695742   5.871 1.41e-06 ***
ski$Garage      13.657168   4.502471   3.033 0.004688 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23.97 on 33 degrees of freedom
Multiple R-squared:  0.8557,	Adjusted R-squared:  0.8338
F-statistic: 39.13 on 5 and 33 DF,  p-value: 6.087e-13
```

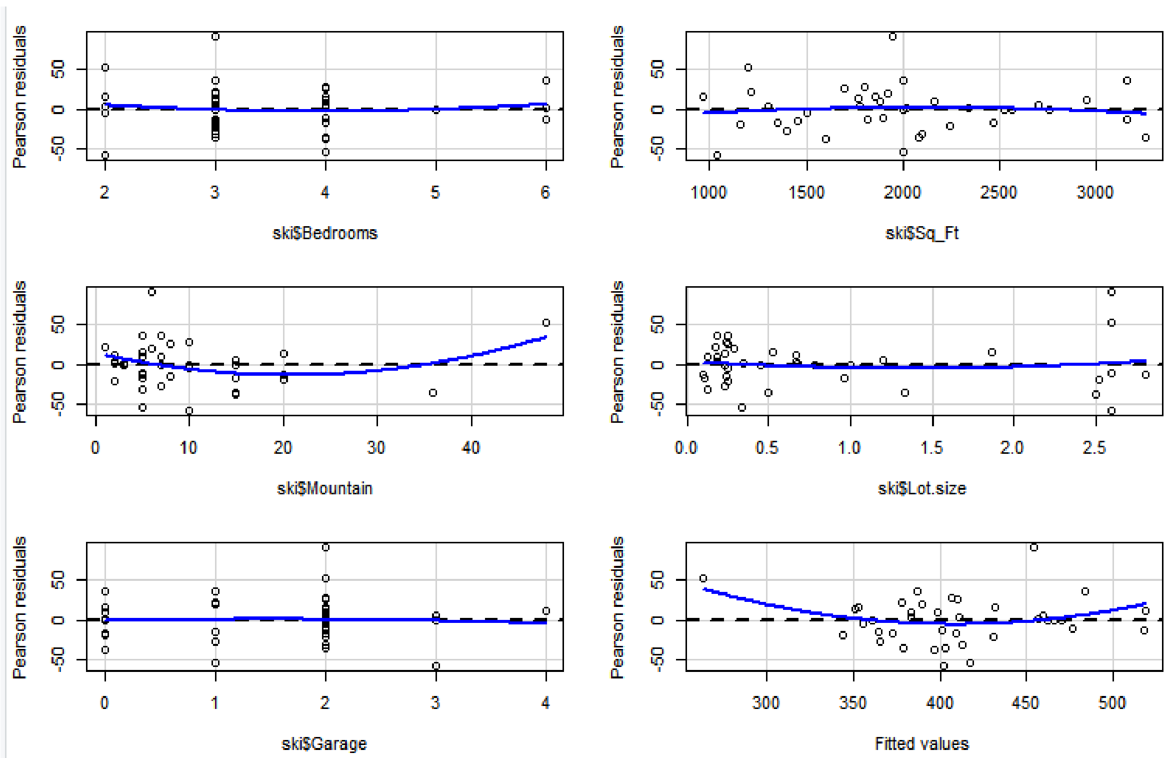
```
Console ~/
> Anova(ModelPerf)
Anova Table (Type II tests)

Response: ski$`Selling price`
                Sum Sq Df F value    Pr(>F)    
ski$Bedrooms      8788  1 15.2949 0.0004330 ***
ski$Sq_Ft        10671  1 18.5715 0.0001387 ***
ski$Mountain     38755  1 67.4482 1.752e-09 ***
ski$`Lot size`   19804  1 34.4663 1.411e-06 ***
ski$Garage        5287  1  9.2007 0.0046877 ** 
Residuals        18961 33                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Console ~/
> BIC(ModelPerf)
[1] 377.5994
> AIC(ModelPerf)
[1] 365.9545
>
```

```
Console ~/
> vif(ModelPerf)
  ski$Bedrooms      ski$Sq_Ft   ski$Mountain ski$`Lot size`     ski$Garage 
      2.264996       2.528167       1.595349       1.682800       1.326219 
>
```



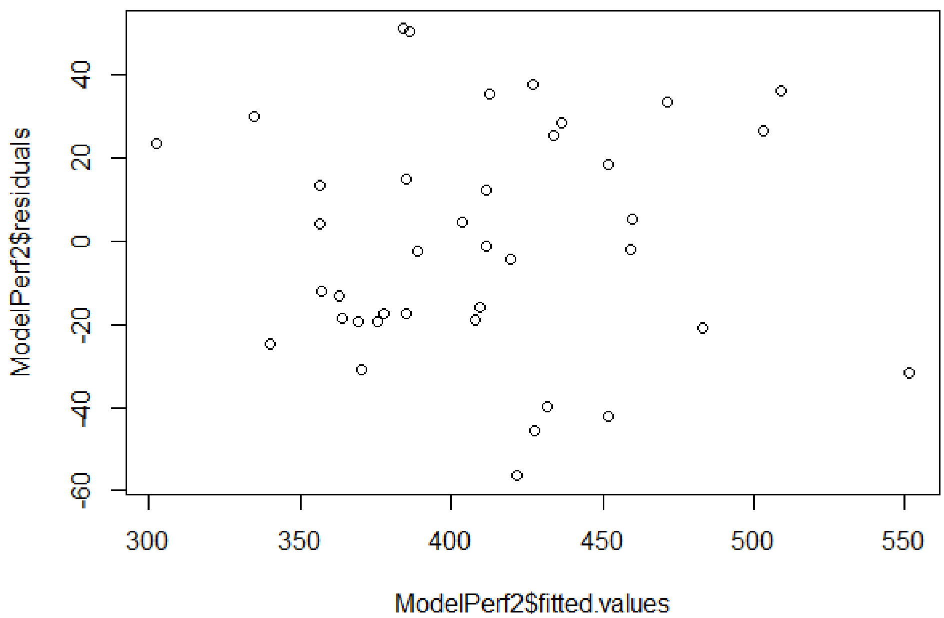

Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
-60
-40
-20
0
20
40
-2
-1
0
1
2

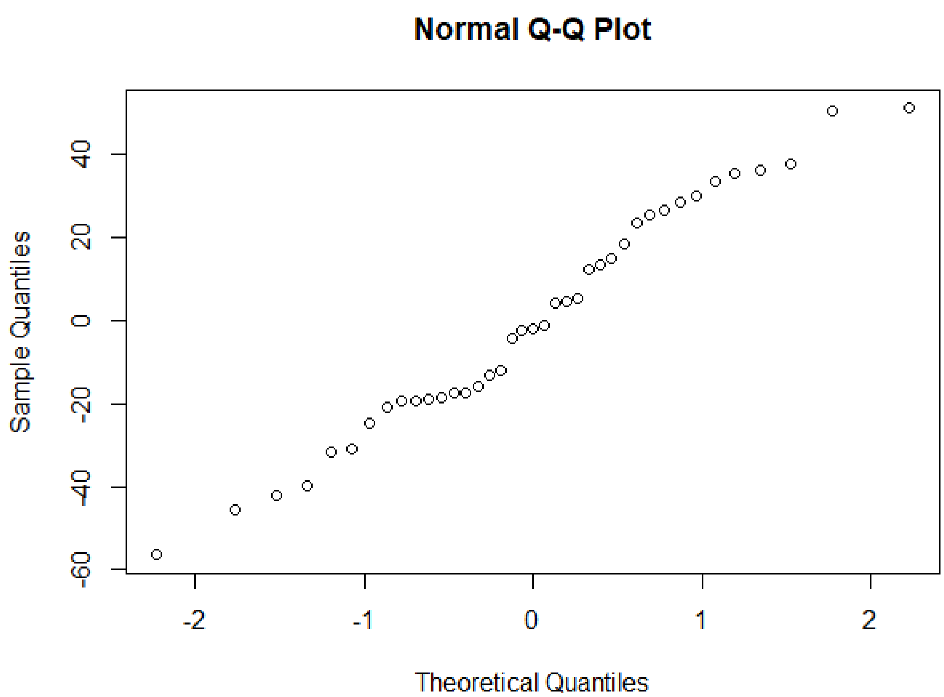

Pearson residuals
ski$Bedrooms
ski$Sq_Ft
ski$Mountain
ski$Lot.size
ski$Garage
Fitted values

Ski$garage has less significance hence, and bedroom has less significance after removing garage, so

ModelPerf2 = lm(ski$`Selling price` ~ ski$Sq_Ft + ski$Mountain + ski$`Lot size` , data = ski)
summary(ModelPerf2)
anova(ModelPerf2)
BIC(ModelPerf2)
AIC(ModelPerf2)
vif(ModelPerf2)

```
Console ~/
> ModelPerf2 = lm(ski$`Selling price` ~  ski$Sq_Ft + ski$Mountain + ski$`Lot size` , data = ski )
> summary(ModelPerf2)

Call:
lm(formula = ski$`Selling price` ~ ski$Sq_Ft + ski$Mountain + 
    ski$`Lot size`, data = ski)

Residuals:
    Min      1Q  Median      3Q     Max 
-56.552 -18.992  -1.824  24.491  51.234 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    304.523903  15.702795  19.393  < 2e-16 ***
ski$Sq_Ft        0.069524   0.007369   9.434 3.80e-11 ***
ski$Mountain    -4.696469   0.621013  -7.563 7.29e-09 ***
ski$`Lot size`   4.431383   0.820452   5.401 4.76e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 28.89 on 35 degrees of freedom
Multiple R-squared:  0.7776,	Adjusted R-squared:  0.7586 
F-statistic:  40.8 on 3 and 35 DF,  p-value: 1.605e-11

> anova(ModelPerf2)
Analysis of Variance Table

Response: ski$`Selling price`
               Df Sum Sq Mean Sq F value    Pr(>F)    
ski$Sq_Ft       1  53331   53331  63.898 2.094e-09 ***
ski$Mountain    1  24479   24479  29.329 4.553e-06 ***
ski$`Lot size`  1  24348   24348  29.172 4.758e-06 ***
Residuals      35  29212     835                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> BIC(ModelPerf2)
[1] 387.1273
> AIC(ModelPerf2)
[1] 378.8095
> vif(ModelPerf2)
     ski$Sq_Ft   ski$Mountain ski$`Lot size` 
      1.091659       1.543617       1.611028 
> 
```


ModelPerf2$residuals
ModelPerf2$fitted.values
-60
-40
-20
0
20
40
300
350
400
450
500
550


Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
-60
-40
-20
0
20
40
-2
-1
0
1
2

Cross Validation:

library(caret)

set.seed(123)

train.control <- trainControl(method = "cv", number = 5)

Train the model

modelcv <- train(Selling.price~ Bedrooms + Sq_Ft + Mountain + Lot.size + Garage, data = ski ,method = "lm",trControl = train.control)

Summarize the results

print(modelcv)

#Rsquare value of every fold

modelcv$resample

```
> modelcv$resample
      RMSE  Rsquared      MAE Resample
1 21.66389 0.9466982 17.02800    Fold1
2 48.15509 0.8214165 30.89720    Fold2
3 45.97276 0.3393708 35.46791    Fold3
4 49.25181 0.4831209 35.55161    Fold4
5 35.33540 0.6589489 29.42096    Fold5
```

set.seed(42)

partition <- createDataPartition(y = ski$Selling.price, p = 0.8, list = F)

trainingdata = ski[partition,]

test <- ski[-partition,]

pcv = predict(modelcv,test)

errorcv <- (pcv- test$Selling.price)

```
errorcv
        10          14          19          20          33          35          36
-2.8911462  26.5373019   0.9714156  -9.0882245 -27.9394185  12.8804506 -11.4493852
```

There is no sign of overfiting but RMSE value is big for the model.

Multiple Regression model – including listing price

For this we used stepwise regression approach

Model 1: Considering all independent variables.

<u>R code:</u>
model1 = lm(skiData$Selling.price~skiData$List.price + skiData$Bathrooms + skiData$Bedrooms + skiData$Sq_Ft + skiData$Downtwon + skiData$Mountain + skiData$Lot.size + skiData$Garage + skiData$Age + skiData$On.market)

summary(model1)

<u>Summary output:</u>

```
Residuals:
     Min       1Q   Median       3Q      Max 
-23.3606  -2.7887   0.8157   4.1485  15.1550 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -2.8588629 15.4506521  -0.185    0.855    
skiData$List.price  0.9834442  0.0541988  18.145   <2e-16 ***
skiData$Bathrooms -0.8453894  3.1604707  -0.267    0.791    
skiData$Bedrooms   1.4025077  2.7286082   0.514    0.611    
skiData$Sq_Ft      0.0004229  0.0046657   0.091    0.928    
skiData$Downtwon  -0.5374185  0.3533876  -1.521    0.140    
skiData$Mountain   0.3588297  0.4451512   0.806    0.427    
skiData$Lot.size  -0.9916845  2.1604711  -0.459    0.650    
skiData$Garage     0.7262106  1.9961089   0.364    0.719    
skiData$Age       -0.0466918  0.0843547  -0.554    0.584    
skiData$On.market -0.0188091  0.0180107  -1.044    0.305    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.682 on 28 degrees of freedom
Multiple R-squared:  0.9839,	Adjusted R-squared:  0.9782 
F-statistic: 171.5 on 10 and 28 DF,  p-value: < 2.2e-16
```

Except Listing price all other independent variable has p-value < 0.05, hence they are not significant.

Using stepAIC() function from MASS library, it helps to screen the best independent variables that can improve the model accuracy.

```
- skiData$On.market   1       152  2285 170.75
- skiData$Downtwon    1       205  2338 171.65
- skiData$List.price  1     91004 93136 315.35

Step:  AIC=168.58
skiData$Selling.price ~ skiData$List.price + skiData$Downtwon + 
    skiData$Mountain + skiData$Lot.size + skiData$On.market

                     Df Sum of Sq   RSS    AIC
- skiData$Lot.size    1        42  2204 167.34
- skiData$Mountain    1        75  2236 167.91
<none>                             2161 168.58
- skiData$On.market   1       166  2327 169.46
- skiData$Downtwon    1       180  2341 169.70
- skiData$List.price  1     94440 96601 314.78

Step:  AIC=167.34
skiData$Selling.price ~ skiData$List.price + skiData$Downtwon + 
    skiData$Mountain + skiData$On.market

                     Df Sum of Sq    RSS    AIC
- skiData$Mountain    1        54   2258 166.28
<none>                              2204 167.34
- skiData$On.market   1       171   2375 168.26
- skiData$Downtwon    1       179   2383 168.39
- skiData$List.price  1     98829 101033 314.53

Step:  AIC=166.28
skiData$Selling.price ~ skiData$List.price + skiData$Downtwon + 
    skiData$On.market

                     Df Sum of Sq    RSS    AIC
<none>                              2258 166.28
- skiData$On.market   1       135   2392 166.54
- skiData$Downtwon    1       259   2517 168.52
- skiData$List.price  1    118449 120707 319.46
> 
```

Using stepAIC( ) function we concluded that we can work on model with list price , downtown and on.market. This function determines best model with less AIC value.

Model 2:

<u>R code:</u>

model2 = lm(skiData$Selling.price ~ skiData$List.price + skiData$Downtwon +

skiData$On.market)

summary(model2)

```
Call:
lm(formula = skiData$Selling.price ~ skiData$List.price + skiData$Downtwon +
    skiData$On.market)

Residuals:
    Min      1Q  Median      3Q     Max
-24.262  -3.287   1.236   4.628  13.568

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.77112   10.03158  -0.077   0.9392
skiData$List.price  0.98830    0.02306  42.853   <2e-16 ***
skiData$Downtwon   -0.25181    0.12559  -2.005   0.0527 .
skiData$On.market  -0.01999    0.01384  -1.445   0.1575
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.031 on 35 degrees of freedom
Multiple R-squared:  0.9828,	Adjusted R-squared:  0.9813
F-statistic: 667.2 on 3 and 35 DF,  p-value: < 2.2e-16


> library(car)
> vif(model2)
skiData$List.price   skiData$Downtwon  skiData$On.market
          1.073707           1.059316           1.023043
>
```

Downtown and On.market are not correlated but p-value of Downtown and on.market > 0.05, hence they are not significant.

Model3: Let's transform Downtown and On.market to their square root value.

<u>R code:</u>

model3 = lm(skiData$Selling.price ~ skiData$List.price + sqrt(skiData$Downtwon) +

sqrt(skiData$On.market))

summary(model3)

```
Call:
lm(formula = skiData$Selling.price ~ skiData$List.price + sqrt(skiData$Downtwon) +
    sqrt(skiData$On.market))

Residuals:
    Min      1Q  Median      3Q     Max
-23.840  -3.342  -0.291   4.343  14.512

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              4.07689   10.49986   0.388   0.7002
skiData$List.price       0.98643    0.02255  43.744   <2e-16 ***
sqrt(skiData$Downtwon)  -1.77457    0.75516  -2.350   0.0245 *
sqrt(skiData$On.market) -0.43068    0.32909  -1.309   0.1992
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.95 on 35 degrees of freedom
Multiple R-squared:  0.9832,    Adjusted R-squared:  0.9817
F-statistic: 681.3 on 3 and 35 DF,  p-value: < 2.2e-16
```

```
vif(model3)
   skiData$List.price  sqrt(skiData$Downtwon) sqrt(skiData$On.market)
             1.047714                1.053303                1.020038
```

After transformation, p-value of downtown < 0.05 which means that variable is significant.

But p – value of On.market after transformation > 0.05 hence that variable is not significant.

We will eliminate On.market in next model

Model4 : Eliminating sqrt(on.market)

<u>R code:</u>

model4 = lm(skiData$Selling.price ~ skiData$List.price + sqrt(skiData$Downtwon))

summary(model4)

vif(model4)

```
Call:
lm(formula = skiData$Selling.price ~ skiData$List.price + sqrt(skiData$Downtwon))

Residuals:
     Min       1Q   Median       3Q      Max 
-26.8876  -2.6539  -0.2958   4.4935  13.1539 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)             0.10949   10.15172   0.011   0.9915    
skiData$List.price      0.98426    0.02271  43.339   <2e-16 ***
sqrt(skiData$Downtwon) -1.67259    0.75852  -2.205   0.0339 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.028 on 36 degrees of freedom
Multiple R-squared:  0.9823,	Adjusted R-squared:  0.9814 
F-statistic:  1001 on 2 and 36 DF,  p-value: < 2.2e-16

> vif(model4)
    skiData$List.price sqrt(skiData$Downtwon) 
              1.042086               1.042086 
> 
```

In model 4, both variables listing price and downtown have significant value.

According to T-test:

p-value of listing price < 0.001, hence variable has 99.9% of confidence whereas p-value of downtown variable < 0.05, hence variable has 95% of confidence.

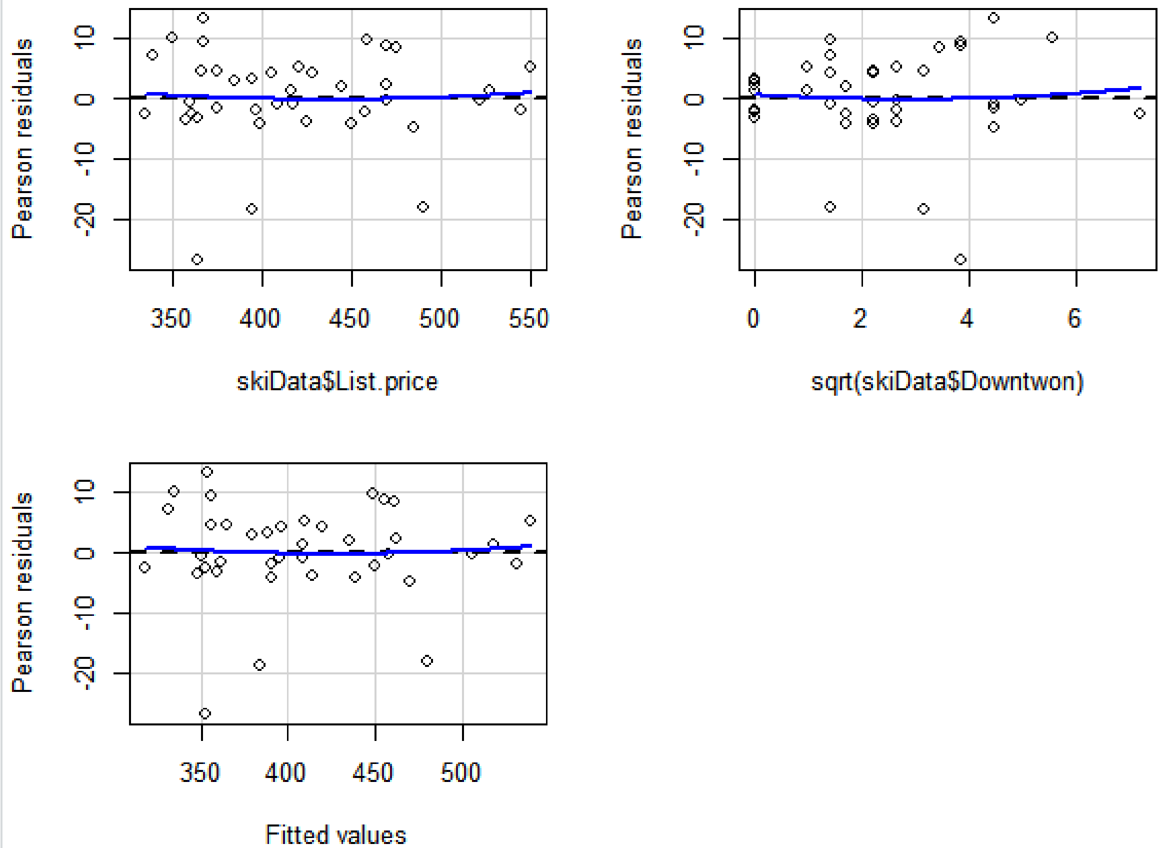
Adjusted R-square = 0.9814, which determines 98.14% variance in selling price of the property.

```
> durbinWatsonTest(model4)
 lag Autocorrelation D-W Statistic p-value
   1      -0.1452316      2.253698    0.47
 Alternative hypothesis: rho != 0
> 
```

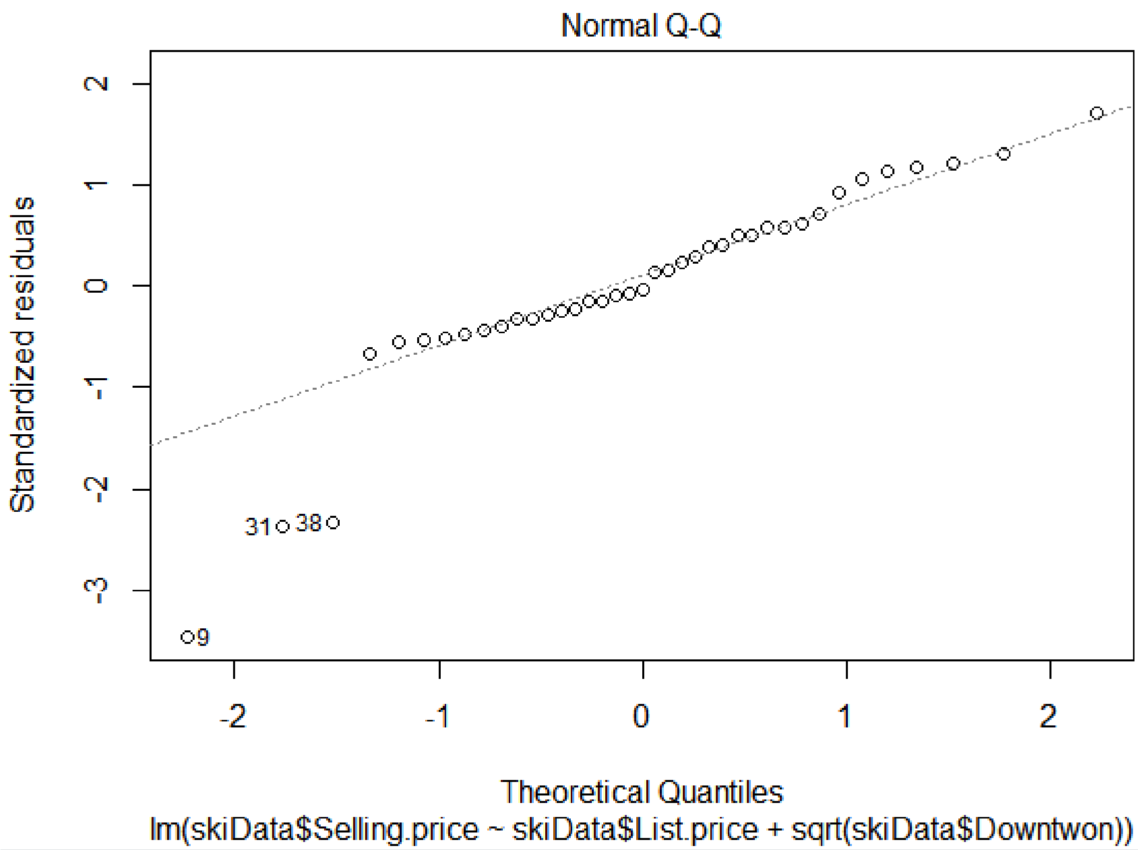
DW ~ 2 ; indicating that the residuals are not correlated, and no evidence of violation of the assumption of randomness and independence.

<u>Verifying assumptions of multiple linear regression</u>

- Using VIF function, it is proved that VIF < 5, hence independent variables are not correlated.
- None of the plots below have a pattern, so the assumption of linearity and independence meet. There is constant variance in residuals, hence assumption of homoscedasticity has been met.



- Residuals following normal pattern since Q-Q plot is almost linear.



- DW ~ 2 ; indicating that the residuals are not correlated, and no evidence of violation of the assumption of randomness and independence.

```
> durbinWatsonTest(model4)
 lag Autocorrelation D-W Statistic p-value
   1      -0.1452316      2.253698    0.47
 Alternative hypothesis: rho != 0
>
```

Hence Model4 satisfies all assumptions of linear regression, hence it is best model.

Model validation:

Selling price = 0.10949 + 0.98426(List price) - 1.67259(√Downtown)

Actual values of selling price	Predicted values of selling price	Error %
470	461.8406	1.736042553
462	480.0306	-3.902727273
520	518.81	0.228846154

CROSS VALIDATION:

Cross Validation is performed to check whether there is no overfiting in model.

```
library(caret)

set.seed(123)

train.control <- trainControl(method = "cv", number = 5)

# Train the model

model <- train(Selling.price~ List.price + sqrt(Downtwon), data = ski ,method = "lm",trControl =
train.control)

# Summarize the results

print(model)

#Rsquare value of every fold

model$resample
```

```
Linear Regression

39 samples
 2 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 31, 31, 32, 31, 31
Resampling results:

  RMSE      Rsquared   MAE
  7.808625  0.9863039  5.733

Tuning parameter 'intercept' was held constant at a value of TRUE
> #Rsquare value of every fold
> model$resample
       RMSE  Rsquared      MAE Resample
1 10.703795 0.9797948 6.487318    Fold1
2  5.430750 0.9944547 4.417934    Fold2
3  4.341128 0.9931585 4.218803    Fold3
4  8.240515 0.9856634 6.161757    Fold4
5 10.326934 0.9784479 7.379187    Fold5
>
```

The R-square value of each fold is significant high.

set.seed(42)

partition <- createDataPartition(y = ski$Selling.price, p = 0.8, list = F)

trainingdata = ski[partition,]

test <- ski[-partition,]

pcv = predict(model,test)

errorcv <- (pcv- test$Selling.price)

RMSE_NewDatacv <- sqrt(mean(errorcv^2))

```
>
> RMSE_NewDatacv
[1] 3.838869
>
```

RMSE value of test data is low (i.e 3.8388), hence the model is best fit and there is no overfiting.