Predicting the price of apartments in Tehran using a Linear Regression Model

Author:

Mahdi Keshavarz

June 2022

Summary

In this project, we have used a linear regression model to predict house prices in Tehran which is the capital and the most important city in Iran economically. First, different charts and methods have been used to make an overview. To do that, different statistical subjects like mean, variance, etc. have been exerted to clearly understand the structure of the dataset.

In the second phase of this project, pre-processing has been done which is a crucial task in every data-driven project. These important processes make the data ready to create the model based on them.

After that, the model is created with acceptable parameters and it can be used to identify the important factors and make a prediction. The results have been analyzed in the end.

Table of Contents

Introduction

This project aims to use a linear regression model to identify different factors that affect house prices in Tehran. There are many variables like area, city zone, facilities, and so on but the most important elements have been considered in our dataset. We can use the model to identify how much the price would change if we change a factor. So, linear regression is the best Machine Learning model that can be used.

The model aims to provide two prominent insights:

- What are the most important variables?
- How much every variable affects the price?

We use R programming language throughout this project many statistical packages have been specially designed for data science. In the first, we visualize the data to gain a better understanding of the model. After that, the linear regression model will be implemented step by step.

Overall view of the data

At first, it is really important to have an overall understanding of the data. We can achieve this through descriptive analysis, different charts, and so on. So, we should examine the data construction.

There are 8 variables (columns) in the dataset:

- Are: this shows the area of the home on a scale of the squared meter
- Room: this variable shows the number of rooms in every house
- Parking: determines whether the house possesses devoted parking or not
- Warehouse: it specifies whether the house has a store or not.
- Elevator: if the house has an elevator
- Address: it is a categorical variable that shows in which district of Tehran city the house is located.
- Price: the value of the house in IRR unit
- Price(USD): the value of the house in USD unit

So we have 8 variables in total and we want to make a model to specify important factors contributing to the house price. So, “Price” or “Price(USD)” will be the dependent variable and the others are the independent ones.

There are 3479 records (rows) in the dataset, so we have enough amount of data to build the model properly. Although more records make the model better, still, there are enough records. We have to carefully use the data to divide them into training and test sets because although the number of records is enough they are limited.

We will use R studio to make the programming easier. First, we have to enter the dataset into the R studio. The dataset is in CSV format which is suitable for R studio. All the headings are in the proper format so we don’t have to be worried about misspelled and inaccurate words. To do that, we use the “summary” function that provides useful information about all the variables in the dataset.

The area is a numeric variable that has been categorized as a string, so we have to convert it to numeric. To do that, we first create a factor and then convert this factor to numeric and put it into the “Area” factor. The code is as follows:

```
HouseAreaFactor=factor(house$Area)
house$Area=as.numeric(HouseAreaFactor)
```

we also have to convert “Parking”, “warehouse”, “Elevator”, and “Address” to factor because they have been identified as “logical” by default. The code is written below:

```
house$Parking=factor(house$Parking)
house$Warehouse=factor(house$Warehouse)
house$Elevator=factor(house$Elevator)
```

Again, we use the “summary” function to examine every variable:

Area Min. : 1.00 1st Qu.: 37.00 Median : 58.00 Mean : 69.66 3rd Qu.: 88.00 Max. :237.00 NA's :6	Room Min. :0.00 1st Qu.:2.00 Median :2.00 Mean :2.08 3rd Qu.:2.00 Max. :5.00
Parking FALSE:529 TRUE :2950	Warehouse FALSE:297 TRUE :3182
Elevator FALSE:740 TRUE :2739	Address Punak: 161 Pardis: 146 West Ferdows Boulevard: 145 Gheitarieh: 141 Shahran: 130 Saadat Abad: 129 (Other): 2627
Price Min. :3.600e+06 1st Qu.:1.420e+09 Median :2.900e+09 Mean :5.368e+09 3rd Qu.:6.000e+09 Max. :9.240e+10 NA's :15	Price.USD. Min. : 120 1st Qu.: 47333 Median : 96667 Mean : 178930 3rd Qu.: 200000 Max. :3080000 NA's :15

Now, we have very good insight into the data. The Median area for houses is 58 which makes sense because Tehran is a large city with many small apartments. The median number of rooms is 2 and this is also reasonable.

The area has been plotted against the price in the scatterplot below. The scatterplot almost represents an exponential distribution. We can see more areas induce more prices exponentially.

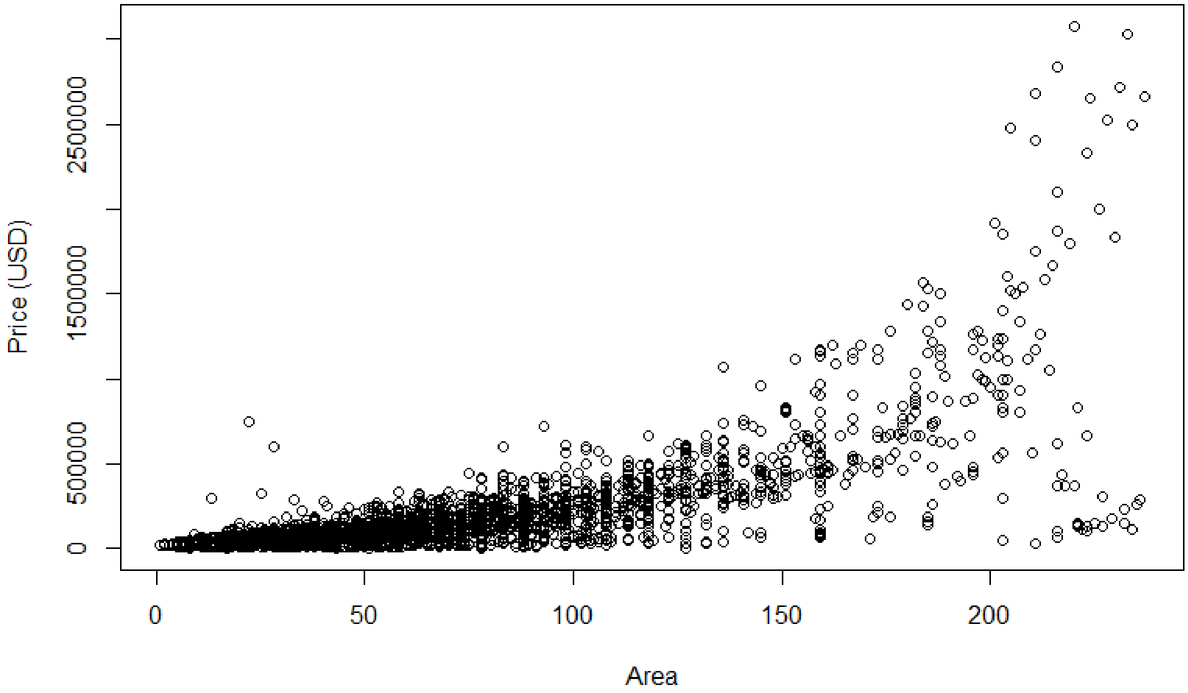


Figure 1- Area vs. price (USD)

The histogram below shows the distribution of the “Room” variable. So, we can see that most of the houses have 2 rooms.

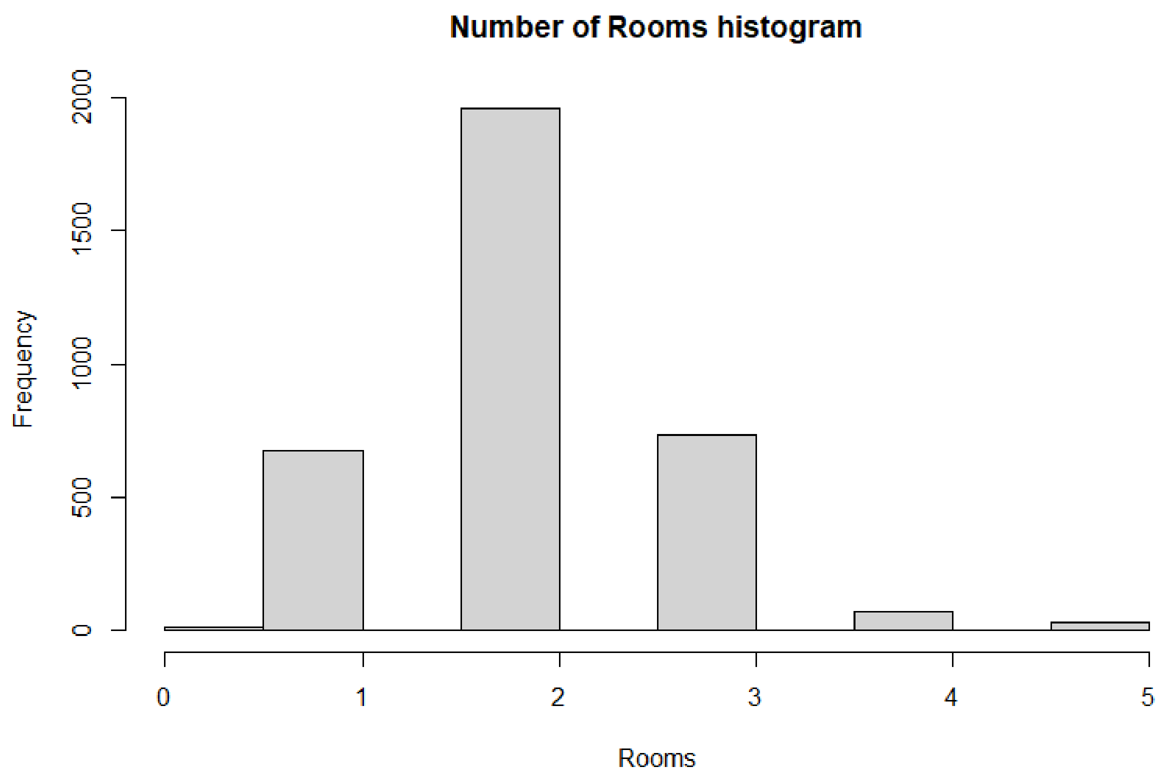


Figure 2- distribution of Room

We can also draw suitable charts for Parking, Warehouse, and Elevator using the "plot" function. As we can see, most of the houses possess Parking but we still have to build the model and then decide whether it is a significant variable or not.

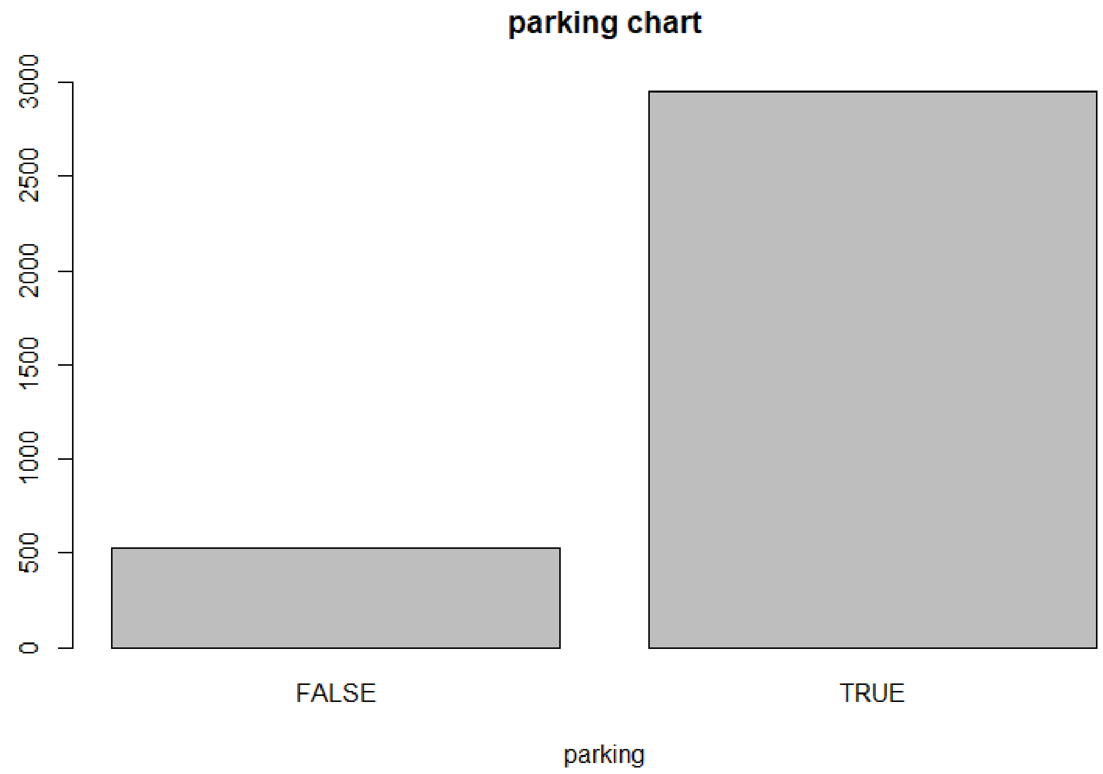


Figure 3- parking status in the database

The chart below indicates how many houses have warehouses. As we can see, there are only a few houses that lack warehouses.

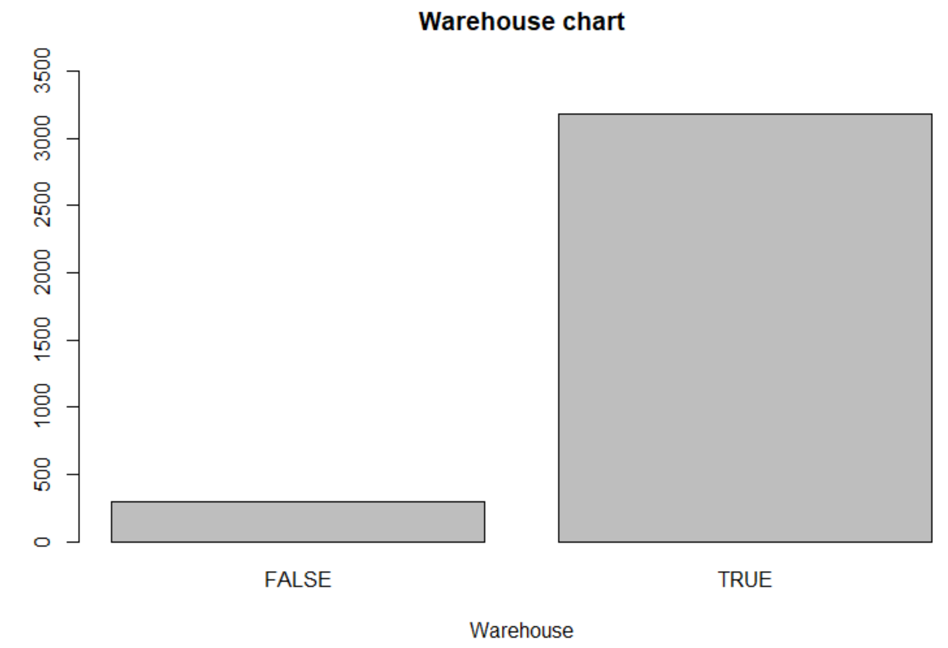


Figure 4- warehouse status in the dataset

The next figure represents the elevator which is another variable in the dataset. Again, most of the houses have access to Elevator.

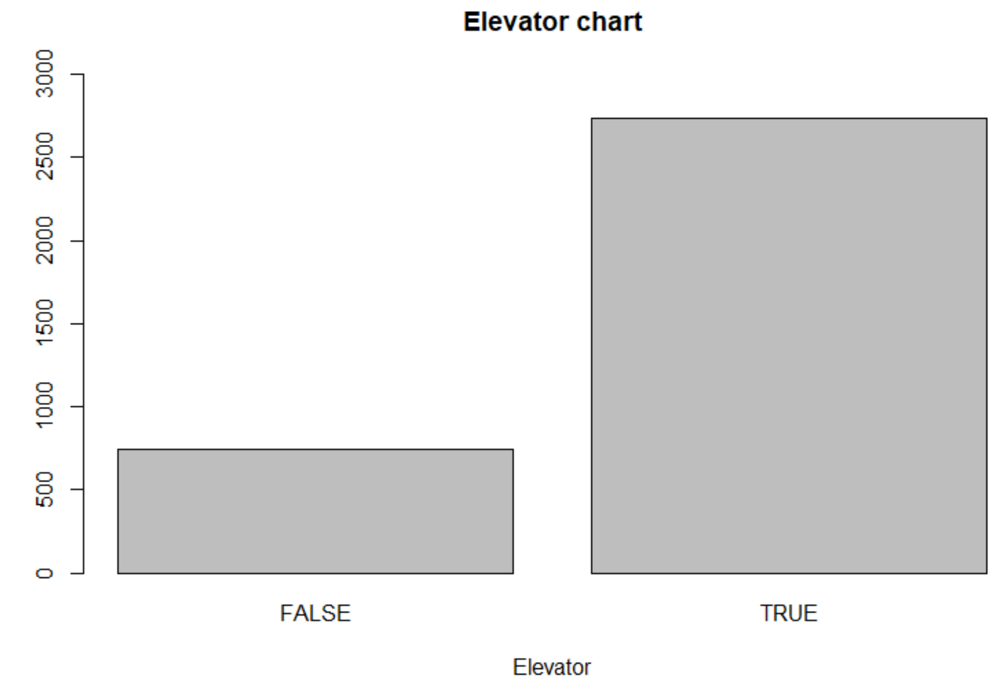


Figure 5- Elevator status in the dataset

Pre-Processing

Now we have good insight into the dataset itself. We know 6 variables can be used to generate a linear regression model. Two other variables both show price and we can use either of them as the response variable. All variables seem to be good for use, but we also have to prepare them.

First of all, we should notice that only numeric variables are suitable for linear models. So, we have to decide whether we use the variable "address" or not. Second, we have to identify outliers and missing values. We will also consider data transformation which are two other ways to prepare data.

The first variable is "area". The Median and mean are close but they are close. The median (185) is larger than the mean (140), so there is some skewness and it may be because of outliers. The minimum area is one and seems unreasonable, so this also confirms there are outliers.

To solve this, we use the Six-sigma approach which is a popular way to treat outliers. Every record which is smaller than 0.3*p1[1] will be replaced with 0.3*p1. For the records with larger values than 3*p99[2], we will replace them with 3*p99. For "Area", there are only small outliers, so we use the above formula using the codes below:

*Area_LL1=0.3*quantile(house$Area,0.01)*

house$Area[house$Area<Area_LL1]=Area_LL1

The problem is that the lower bound is 6 squared meters which also seems unreasonable. But, we can ignore it because there are very small apartments in some city zones.

As it was mentioned earlier, there are two variables indicating price, so we have to delete one of them. Variable "Price.USD." seems to be better because of two reasons:

1. It is on a scale of USD and it is more understandable that IRR
2. The values are smaller so there will be less time consumption in calculations when either creating or running the model.

[1] First quartile

[2] 99th quartile

The code to do that is written below:

```
house=house[,-7]
```

There are 15 missing values in “price.USD.” and this is not acceptable for a linear regression model. One of the ways to solve this is to replace these values with the average price of the address. This approach makes sense because every address which shows a distinct zone of the city, has a specific range, and an average for prices. But, there’s a problem with this approach. Price is the response variable and it is better to omit these records with missing values. Only 15 records will be deleted, so it will not significantly harm the dataset. So, we will delete these records and then, keep on building the model.

```
empty=which(is.na(house$Price.USD.))
house=house[-empty,]
```

There are also some outlier variables in price. The minimum value is 120 dollars whereas the first quantile is 47333 dollars. This surge in price is a considerable phenomenon and it proves there are surely outliers. There’s also a large difference between the median (96667) and the mean (178930). The outliers are in the response variable, so we will make our model in two ways. First, with these outliers, and then, with outlier treatment. After that, we will compare the model in two situations.

Parking, Warehouse, and Elevetaor are strings and we have to convert them to numeric variables. To do that, we could use dummy variables. This approach expands every field (variable) to the number of unique values in that specific field. For example, parking has values of “TRUE” and “FALSE”. So, two new variables will emerge. One of them is “parking true”, and the other one is “parking false”.

But, we only need one of them, so we will one, later. This rule applies to all the other string variables. We use the “dummies” package, and then, the code below to create dummy variables. Before that, there is something else that we should notice. There are many unique values in “Address”. So, there will be a lot of dummy variables. To prevent that, we could delete it, but it is an important variable. To deal with it, we use a reasonable approach. Every Address is located in a specific city district. So, we will use variable transformation and put every address in a new variable. For example, Darabad and Elahieh are both in District 1 of Tehran. So, we can replace these values with a new value called “Dist1”.

First, we have created a CSV file containing unique values in the Address column (192 values) named “Address.csv”. Then, we entered every matching district in another column one by one. After that, we created another CSV file from the data frame in R studio named “New_House.csv”. This file is the dataset and then, we used the values in “Address.csv”, and “vlookup” function to create a new column in “New_House.csv”. in the end, we enter “New_House.csv” and remove unnecessary fields.

```
unique_addresses=unique(house$Address)
write.csv(unique_addresses,"Adress.csv")
write.csv(house,"New_House.csv")
house=read.csv("New_House.csv")
house=house[,-6]
```

Now, it is time to make dummy variables because there are fewer unique values in the address.

```
house$Parking=factor(house$Parking)
house$Warehouse=factor(house$Warehouse)
house$Elevator=factor(house$Elevator)
house$New_Address=factor(house$New_Address)


house=dummy.data.frame(house)  #dummy variables
house=house[,c(-3,-5,-7)]  #dummy variables
house=house[,-7]
```

Now, all the variables are in the correct numeric format.

in the end, we have to check whether there is a significant correlation between variables or not. If so, we have to delete one of them, because there is no need to have two strongly correlated variables.

```
cor(house)
```

There is no significant correlation between any pair of variables, so no column needs to be deleted.

Train-Test Split

Now we have the dataset ready to use, but there is one important step before creating the linear regression model. We need to divide data into two sets:

- Train set: it is used to train the model. This set is the source that makes Machine learn
- Test set: this set is used to test the accuracy of the model after it is trained by the training set.

There are some approaches for train-test split, but we have chosen the 80-20 way. In this method, 80 percent of the records will be the training set, and the remaining 20 percent are the test set. This is a simple but efficient way that is being used in many data-driven projects.

```
set.seed(0)

x=sample.split(house, SplitRatio=0.8)

training_set= subset(house, x==TRUE)

test_set= subset(house, x==FALSE)
```

Linear Regression Model

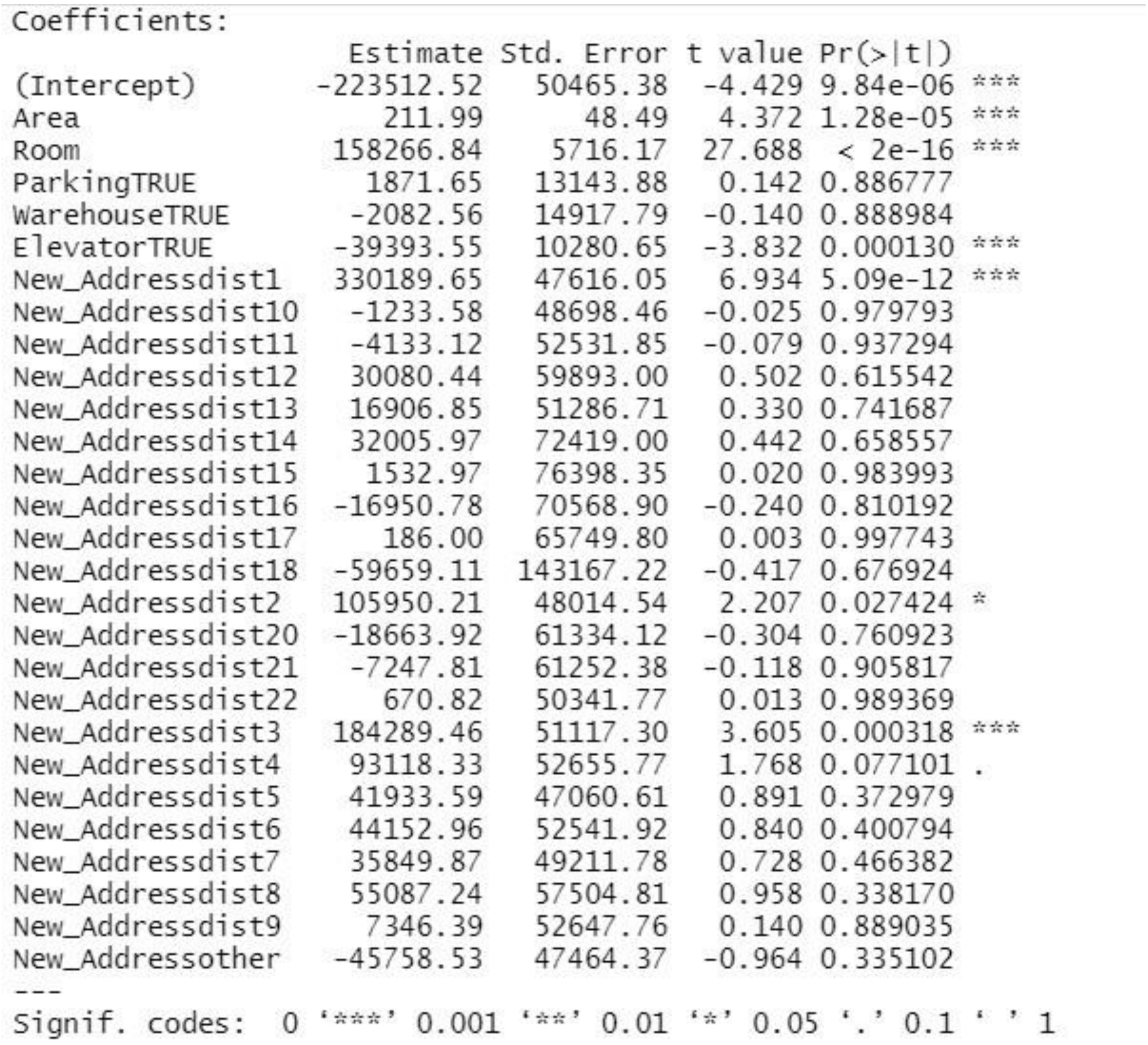
Now, it is time to create our model. We use the “lm” function which is used to create linear models:

linear_model=lm(Price.USD.~ . ,data=training_set)

summary(linear_model)

The result is as follows:

Residuals:				
Min	1Q	Median	3Q	Max
-580367	-80461	-26599	59432	2379484



```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)        -223512.52   50465.38  -4.429 9.84e-06 ***
Area                   211.99      48.49   4.372 1.28e-05 ***
Room                158266.84    5716.17  27.688  < 2e-16 ***
ParkingTRUE           1871.65   13143.88   0.142 0.886777    
WarehouseTRUE        -2082.56   14917.79  -0.140 0.888984    
ElevatorTRUE        -39393.55   10280.65  -3.832 0.000130 ***
New_Addressdist1    330189.65   47616.05   6.934 5.09e-12 ***
New_Addressdist10    -1233.58   48698.46  -0.025 0.979793    
New_Addressdist11    -4133.12   52531.85  -0.079 0.937294    
New_Addressdist12    30080.44   59893.00   0.502 0.615542    
New_Addressdist13    16906.85   51286.71   0.330 0.741687    
New_Addressdist14    32005.97   72419.00   0.442 0.658557    
New_Addressdist15     1532.97   76398.35   0.020 0.983993    
New_Addressdist16   -16950.78   70568.90  -0.240 0.810192    
New_Addressdist17      186.00   65749.80   0.003 0.997743    
New_Addressdist18   -59659.11  143167.22  -0.417 0.676924    
New_Addressdist2    105950.21   48014.54   2.207 0.027424 *  
New_Addressdist20   -18663.92   61334.12  -0.304 0.760923    
New_Addressdist21    -7247.81   61252.38  -0.118 0.905817    
New_Addressdist22      670.82   50341.77   0.013 0.989369    
New_Addressdist3    184289.46   51117.30   3.605 0.000318 ***
New_Addressdist4     93118.33   52655.77   1.768 0.077101 .  
New_Addressdist5     41933.59   47060.61   0.891 0.372979    
New_Addressdist6     44152.96   52541.92   0.840 0.400794    
New_Addressdist7     35849.87   49211.78   0.728 0.466382    
New_Addressdist8     55087.24   57504.81   0.958 0.338170    
New_Addressdist9      7346.39   52647.76   0.140 0.889035    
New_Addressother    -45758.53   47464.37  -0.964 0.335102    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Residual standard error:	191400 on 2693 degrees of freedom
Multiple R-squared:	0.4849
Adjusted R-squared:	0.4797
F-statistic:	93.88 on 27 and 2693 DF
p-value:	< 2.2e-16

The model indicates some significant variables affect the house price in Tehran:

- Are with 99.9 percent confidence
- Having a Parking with 99.9 percent confidence
- Having an elevator with 99.9 percent confidence
- Being located in District 1 with 99.9 percent confident
- Being located in District 2 with 95 percent confident
- Being located in District 3 with 99.9 percent confident
- Being located in District 4 with 90 percent confident

There is also the Intercept with 99.9 percent confidence and this means there are some variables that we have overlooked. R-squared is 48 percent, meaning only 48 percent of the variance in price has been defined by the current variable. R-squared also is 47 percent meaning the model is not good enough to predict or interpret the variance in the prices of houses in Tehran.

We should also consider the F test and the P-value. The P-value is less than 2.2e-16 meaning there is no significant evidence to decline the hypothesis test. So, the results are considered to be approvable.

Alternative approaches

We have to create a more precise model. Some approaches may result in better results.

1- First, we will not replace neighborhoods (values in the Address). This might cause us to have many columns after creating dummy variables, but we hope to reach more precise results.

The results are as follows:

Residuals:				
Min	1Q	Median	3Q	Max
-892630	-57492	-11108	49563	2108514

Significant variables are written below:

- Area: 99.9 percent
- Room: 99.9 percent
- Having an elevator: 90 percent
- Some neighborhoods are also significant

Again, Intercept is also significant with 99.9 percent confident

Residual standard error:	172900 on 2576 degrees of freedom
Multiple R-squared:	0.6168
Adjusted R-squared:	0.5891
F-statistic:	22.29 on 186 and 2576 DF
p-value:	< 2.2e-16

As can be seen, R-squared rose to 61 percent. So, 61 percent of the variation is now under the current variables' control. R-squared is 58 percent which is better than before, but the model is not good enough yet. F-test and P-values indicate the results can be trusted and there's no reason to decline them.

So far, we used OLS or the ordinary least square method to minimize residuals and create a linear regression model. Now, we have to use other methods, because the model is not precise enough.

Other models

First, we want to use the best subset selection method. This approach aims to reduce variables in order to get the best subset with significant variables. The number of variables is now large, and this method could be very useful. The code to that is as follows:

```
x1=regsubset(dependent_variable~ . ,data=dataframe_name, nvmax=6)

summary(x1)

summary(x1)$adjr2

y= which.max(summary(x1)$adjr2)

coef(x1)

coef(x1, y1)
```

the result shows:

```
0.3218180 0.4427317 0.4572662 0.4679709 0.4736044 0.4784010
```

There are no adjusted R-Squarred with significant values. So, this approach seems to not be useful.

Conclusion

Now we are at the end and we should determine what the outcome is. The OLS method is the approach for the simple linear regression model and we could create a model with R-Squarred of 60% which is a very good number.

There is one variable that the dataset lacks which is age. This indicates how many years the apartment has aged and it is a very important factor to determine the price. But, we still have reasonable results and should suffice.

The significant variables are:

- Area
- Room
- Elevator
- Some specific districts

All these variables seem to be reasonable because obviously, they are important for pricing apartments in Tehran.