Multiple_Regression.R

2019-11-07

```r
##Author: Priyam Saxena
########  Multiple Regression Analysis ##########

library(data.table)
library(ggplot2)  # tidyverse data visualization package
library(stringr)
library(corrplot)

## corrplot 0.84 loaded

library(psych)

##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

#Importing csv file from my local computer
airbnbOriginalDF =read.csv("D:/Priyam/FirstSemester/MVA project/airbnb-host-
analysis-for-newyork/Airbnb Host Data For Newyork City.csv")

##Converting data frame to data table
setDT(airbnbOriginalDF)

#Removing values which are null and storing in new table.
airbnbNoNADT = airbnbOriginalDF[airbnbOriginalDF$reviews_per_month != 'NA']

#Converting datatype of last review date to DAte Format.
airbnbNoNADT[,last_review:=as.Date(last_review, '%m/%d/%Y')]

#As the neighbourhood_group column has 5 categorical values, we can factor
it, and convert our string data type.
airbnbNoNADT[,neighbourhood_group:= factor(neighbourhood_group)]

#For room type, we get 3 unique categorical values. we can factor it, and
convert our string datatype.
airbnbNoNADT[,room_type:= factor(room_type)]

#With earlier analysis/ summary and plot we found few ouliers, therefore that
data  we have dropped below, conforming it is not impact our main dataset.
airbnbCleaned = airbnbNoNADT[price<2500 & number_of_reviews<400 &
reviews_per_month<10]
```

```r
##Manhattan area dataset
airbnbManhattan = airbnbCleaned[neighbourhood_group=='Manhattan']
nrow(airbnbManhattan)

## [1] 16584

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(data.table)

##Taking the numeric columns that will contribute for variance in data
airbnbManhattanLM = data.frame(
  airbnbManhattan$id,
  airbnbManhattan$host_id,
  airbnbManhattan$room_type,
  airbnbManhattan$price,
  airbnbManhattan$minimum_nights,
  airbnbManhattan$number_of_reviews,
  airbnbManhattan$reviews_per_month,
  airbnbManhattan$availability_365)

setDT(airbnbManhattanLM)

##Setting column names for our new dataframe
names(airbnbManhattanLM) <- c(
  'id',
  'host_id',
  'room_type',
  'price',
  'minimum_nights',
  'number_of_reviews',
  'reviews_per_month',
  'availability_365')

head(airbnbManhattanLM, 5)
```

```
##      id host_id       room_type price minimum_nights number_of_reviews
## 1: 2595    2845 Entire home/apt   225              1                45
## 2: 5022    7192 Entire home/apt    80             10                 9
## 3: 5099    7322 Entire home/apt   200              3                74
## 4: 5203    7490    Private room    79              2               118
## 5: 5238    7549 Entire home/apt   150              1               160
##    reviews_per_month availability_365
## 1:              0.38              355
## 2:              0.10                0
## 3:              0.59              129
## 4:              0.99                0
## 5:              1.33              188
```

Performing Multiple Regression

```
# Performing multiple regression on Airbnb Manhattan dataset
fit_airbnb <-
lm(price~number_of_reviews+availability_365+minimum_nights+room_type,
data=airbnbManhattanLM)

#show the results
#Section1: How well does the model fit the data (before Coefficients).
#Section2: Is the hypothesis supported? (until sifnif codes).
#Section3: How well does data fit the model (again).

summary(fit_airbnb)
```

```
##
## Call:
## lm(formula = price ~ number_of_reviews + availability_365 + minimum_nights
+
##     room_type, data = airbnbManhattanLM)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -212.97  -63.72  -22.35   21.31 2109.47
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             2.067e+02  1.731e+00 119.398   <2e-16 ***
## number_of_reviews      -2.034e-01  2.397e-02  -8.484   <2e-16 ***
## availability_365        2.325e-01  8.499e-03  27.352   <2e-16 ***
## minimum_nights         -5.766e-01  5.184e-02 -11.124   <2e-16 ***
## room_typePrivate room -1.165e+02  2.213e+00 -52.656   <2e-16 ***
## room_typeShared room  -1.554e+02  7.370e+00 -21.090   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 136.2 on 16578 degrees of freedom
```

```
## Multiple R-squared:  0.189,  Adjusted R-squared:  0.1888 
## F-statistic: 772.7 on 5 and 16578 DF,  p-value: < 2.2e-16
```

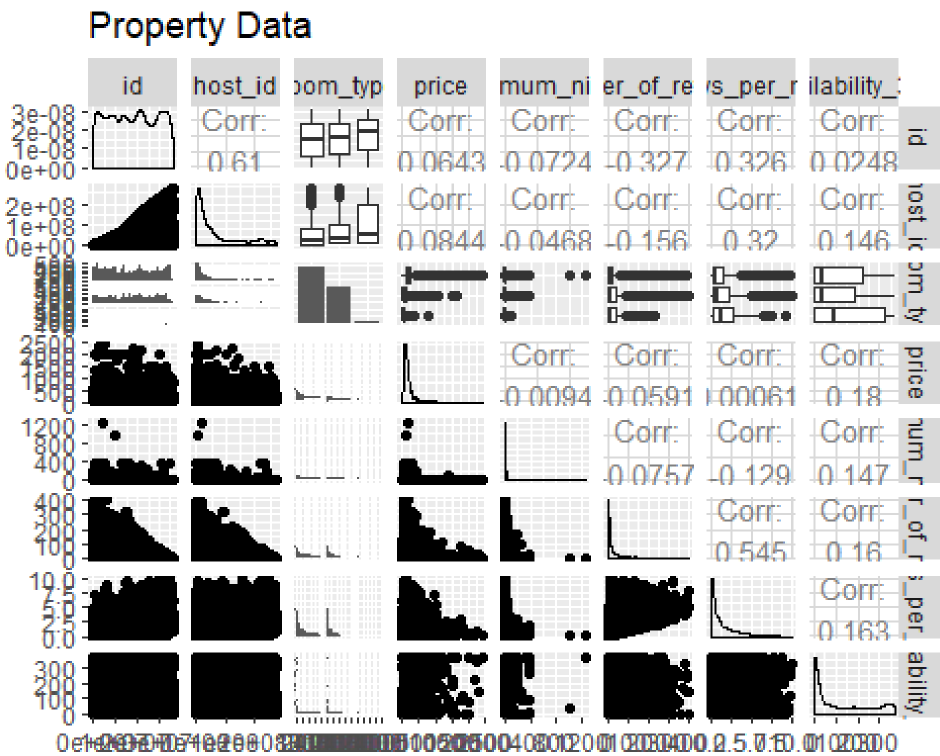
Analysis of model results:

```
#The coefficients describe the mathematical relationship between each
independent variable and the dependent variable.
#The p-values for the coefficients indicate whether these relationships are
statistically significant.

#From the summary no_of_reviews , availability_365, minimum_nights are
statistically significant because their p-values are very small.



#It is standard practice to use the coefficient p-values to decide whether to
include variables in the final model.
#For the results above, we would consider all variables. Keeping variables
that are not statistically significant can reduce the model's precision.

#The coefficient value signifies how much the mean of the dependent variable
changes given a one-unit shift in the independent variable
#while holding other variables in the model constant. This property of
holding the other variables constant is crucial because it allows
#you to assess the effect of each variable in isolation from the others.
```

Plotting a matrix to explore relationships:

```
#After fitting a regression model, check the residual plots first to be sure
that you have unbiased estimates.
#for combining plots into a matrix through the ggpairs function.
#matrix of plots can be useful for quickly exploring the relationships
between multiple columns of data in a data frame.
library(GGally)

## Registered S3 method overwritten by 'GGally':
##   method from   
##   +.gg   ggplot2

## 
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':
## 
##     nasa

ggpairs(data=airbnbManhattanLM,title="Property Data")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#To extract fitted values from objects returned by modeling functions
#fitted(fit_airbnb)
#To check residuals
#residuals(fit_airbnb)
```

Testing Outliers:

```
library(car)

## Loading required package: carData

##
## Attaching package: 'car'

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

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

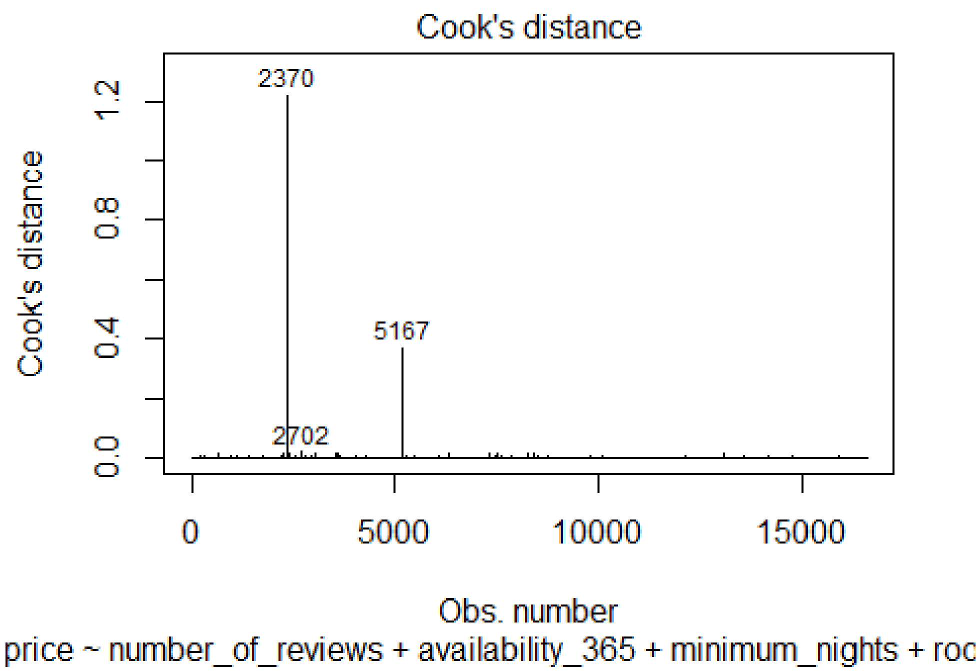
```r
outlierTest(fit_airbnb)
```

```
##       rstudent unadjusted p-value Bonferroni p
## 2702  15.60418         1.6649e-54   2.7611e-50
## 1734  15.30017         1.7415e-52   2.8881e-48
## 8403  14.55287         1.1022e-47   1.8279e-43
## 8258  14.55093         1.1335e-47   1.8798e-43
## 10097 14.37200         1.4771e-46   2.4497e-42
## 215   13.41219         8.3731e-41   1.3886e-36
## 1419  13.26418         5.9685e-40   9.8981e-36
## 4293  13.24590         7.5964e-40   1.2598e-35
## 13574 13.24288         7.9049e-40   1.3110e-35
## 5490  13.12700         3.6163e-39   5.9972e-35
```

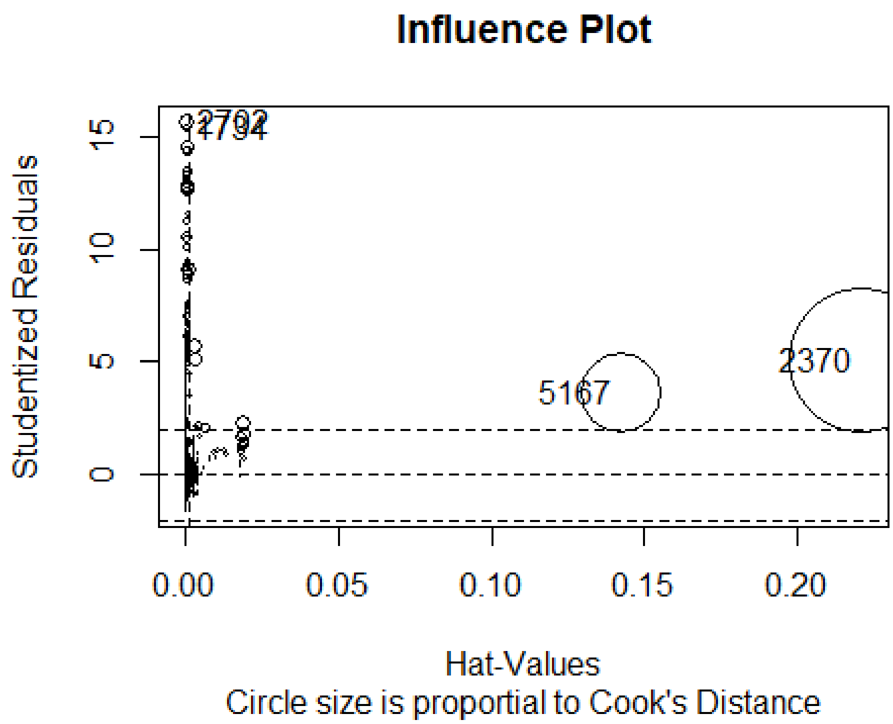
```r
#The result gives values at given row number are outliers.
```

Analysing Negative Effect of points on Model using Cook's Distance:

```r
# Cook's D plot
##it's a way to identify points that negatively affect your regression model.
#The measurement is a combination of each observation's leverage and residual
#values; the higher the leverage and residuals, the higher the Cook's
distance. Cook's distance
# identify D values > 4/(n-k-1)
cutoff <- 4/((nrow(airbnbManhattanLM)-length(fit_airbnb$coefficients)-2))
plot(fit_airbnb, which=4, cook.levels=cutoff)
```




```r
# Representation of above using Influence Plot
influencePlot(fit_airbnb, id.method="identify", main="Influence Plot",
sub="Circle size is proportial to Cook's Distance" )

## graphical parameter
```




```
##         StudRes          Hat       CookD
## 1734 15.300167 0.0001101862 0.004239866
## 2370  5.075466 0.2209089287 1.215562864
## 2702 15.604177 0.0004128472 0.016519338
## 5167  3.641068 0.1424798801 0.366855187
```

```
##THIS SHOWS THE RESULTING POINTS HAVE MUCH NEGATIVE EFFECT ON OUR MODEL.
```

Extracting Studentized Residuals:

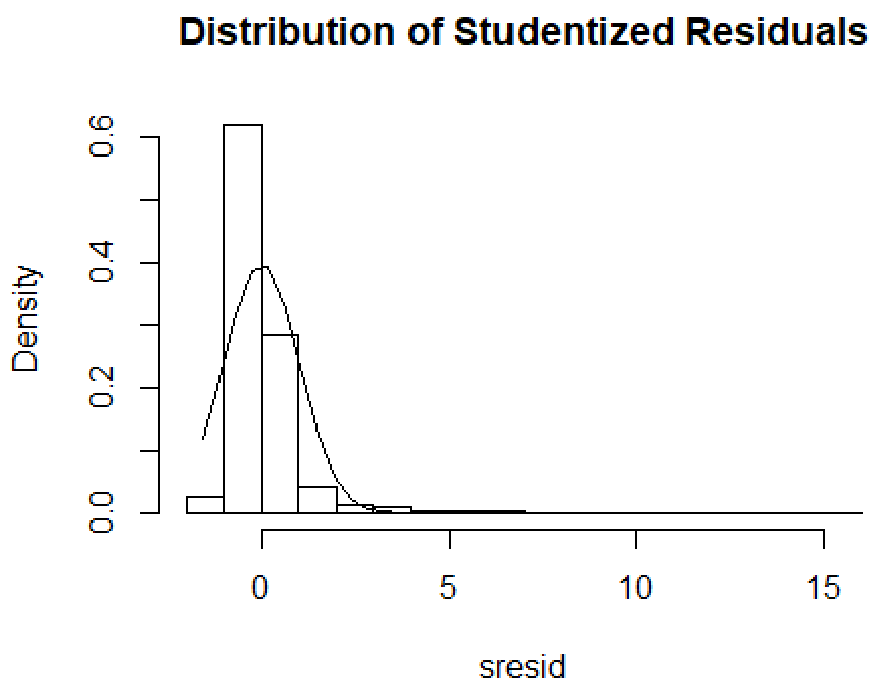
```
#Extract Studentized Residuals From A Linear Model
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

```
#A studentized residual is calculated by dividing the residual by an estimate of its
#standard deviation. The standard deviation for each residual is computed with the observation excluded.
#For this reason, studentized residuals are sometimes referred to as externally studentized residuals
```

```r
sresid <- studres(fit_airbnb)

##Lets view the distribution of theses studentized residuals.
hist(sresid, freq=FALSE,
     main="Distribution of Studentized Residuals")
xfit<-seq(min(sresid),max(sresid),length=40)
yfit<-dnorm(xfit)
lines(xfit, yfit)
```



Test for Auto correlated Errors- Durbin-Watson Test

```r
#Computes residual autocorrelations and generalized Durbin-Watson statistics
and their bootstrapped p-values
#Non-independence of Errors
durbinWatsonTest(fit_airbnb)

##  lag Autocorrelation D-W Statistic p-value
##    1      0.06305132      1.873888       0
##  Alternative hypothesis: rho != 0
```

Global Test of model assumptions

```r
library(gvlma)

## The gvlma( ) function in the gvlma package, performs a global validation
of
```

```r
#linear model assumptions as well separate evaluations of skewness, kurtosis,
#and heteroscedasticity
gvmodel <- gvlma(fit_airbnb)
summary(gvmodel)

##
## Call:
## lm(formula = price ~ number_of_reviews + availability_365 + minimum_nights
+
##     room_type, data = airbnbManhattanLM)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -212.97  -63.72  -22.35   21.31 2109.47
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.067e+02  1.731e+00 119.398   <2e-16 ***
## number_of_reviews     -2.034e-01  2.397e-02  -8.484   <2e-16 ***
## availability_365       2.325e-01  8.499e-03  27.352   <2e-16 ***
## minimum_nights        -5.766e-01  5.184e-02 -11.124   <2e-16 ***
## room_typePrivate room -1.165e+02  2.213e+00 -52.656   <2e-16 ***
## room_typeShared room  -1.554e+02  7.370e+00 -21.090   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 136.2 on 16578 degrees of freedom
## Multiple R-squared:  0.189,  Adjusted R-squared:  0.1888
## F-statistic: 772.7 on 5 and 16578 DF,  p-value: < 2.2e-16
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = fit_airbnb)
##
##                        Value  p-value                   Decision
## Global Stat        1.934e+06 0.000000 Assumptions NOT satisfied!
## Skewness           8.272e+04 0.000000 Assumptions NOT satisfied!
## Kurtosis           1.851e+06 0.000000 Assumptions NOT satisfied!
## Link Function      2.968e+02 0.000000 Assumptions NOT satisfied!
## Heteroscedasticity 1.045e+01 0.001223 Assumptions NOT satisfied!
```

The Akaike Information Criterion Estimator

```r
##The stepAIC() function performs backward model selection by starting from a
#"maximal" model, which is then trimmed down. The "maximal" model is a linear
#regression model which assumes independent model errors and includes only
main
#effects for the predictor variables

#The Akaike information criterion (AIC) is an estimator of the relative
quality of statistical models for a given set of data. Given a collection of
models
#for the data, AIC estimates the quality of each model, relative to each of
the other models.
#Thus, AIC provides a means for model selection.
library(MASS)
step <- stepAIC(fit_airbnb, direction="both")

## Start:  AIC=162997.6
## price ~ number_of_reviews + availability_365 + minimum_nights +
##     room_type
##
##                     Df Sum of Sq       RSS    AIC
## <none>                           307527573 162998
## - number_of_reviews  1   1335283 308862856 163067
## - minimum_nights     1   2295317 309822891 163119
## - availability_365   1  13878117 321405690 163728
## - room_type          2  55606592 363134165 165750

step$anova # display results

## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## price ~ number_of_reviews + availability_365 + minimum_nights +
##     room_type
##
## Final Model:
## price ~ number_of_reviews + availability_365 + minimum_nights +
##     room_type
##
##
##   Step Df Deviance Resid. Df Resid. Dev      AIC
## 1                      16578  307527573 162997.6

summary(step)$coeff

##                             Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)              206.6590617 1.73084326 119.397907  0.000000e+00
## number_of_reviews         -0.2033836 0.02397205  -8.484195  2.353230e-17
## availability_365           0.2324702 0.00849920  27.352014 3.784136e-161
```

```
## minimum_nights          -0.5766495 0.05184019 -11.123601  1.216277e-28
## room_typePrivate room -116.5375567 2.21319169 -52.655880  0.000000e+00
## room_typeShared room  -155.4420360 7.37033631 -21.090223  1.858219e-97
```

```r
summary(step)$r.squared
```

```
## [1] 0.1889967
```

```r
#The adjusted R^2 is 18.89% which means that the model explains 18% of the
variation in mpg
#indicating it is a robust and highly predictive model.
```

Stepwise Selection Model - anova

```r
#Stepwise selection
fit1 <- lm(price ~ number_of_reviews,data = airbnbManhattanLM)
fit2 <- lm(price ~ number_of_reviews+availability_365, data =
airbnbManhattanLM)
fit3 <- lm(price ~ number_of_reviews+availability_365+minimum_nights, data =
airbnbManhattanLM)
fit4 <- lm(price ~
number_of_reviews+availability_365+minimum_nights+room_type, data =
airbnbManhattanLM)

anova(fit1, fit2, fit3, fit4)
```

```
## Analysis of Variance Table
##
## Model 1: price ~ number_of_reviews
## Model 2: price ~ number_of_reviews + availability_365
## Model 3: price ~ number_of_reviews + availability_365 + minimum_nights
## Model 4: price ~ number_of_reviews + availability_365 + minimum_nights +
##     room_type
##   Res.Df       RSS Df Sum of Sq       F    Pr(>F)
## 1  16582 377867793
## 2  16581 363918095  1  13949698  751.99 < 2.2e-16 ***
## 3  16580 363134165  1    783930   42.26 8.221e-11 ***
## 4  16578 307527573  2  55606592 1498.80 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

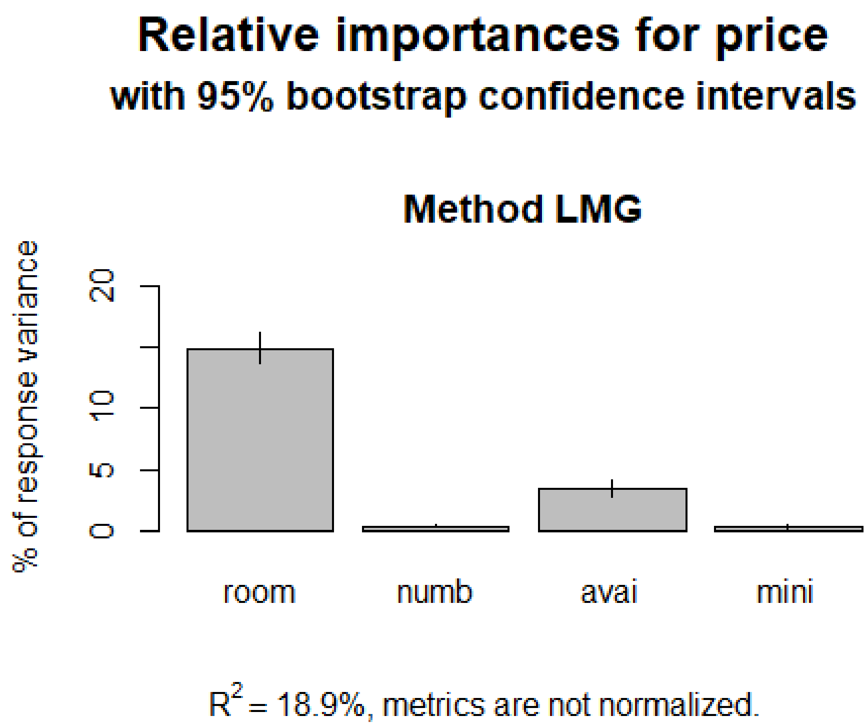
```r
#The above shows that result is consistent with stepwise selection model
```

Calculation of Relative Importance of each Predictor

```
# Calculate Relative Importance for Each Predictor
library(relaimpo)

## Loading required package: boot

##
## Attaching package: 'boot'

## The following object is masked from 'package:car':
##
##     logit

## The following object is masked from 'package:psych':
##
##     logit

## Loading required package: survey

## Loading required package: grid

## Loading required package: Matrix

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:boot':
##
##     aml

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart

## Loading required package: mitools

## This is the global version of package relaimpo.

## If you are a non-US user, a version with the interesting additional metric
pmvd is available

## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.

calc.relimp(fit_airbnb)

## Response variable: price
## Total response variance: 22866.43
```

```
## Analysis based on 16584 observations
##
## 5 Regressors:
## Some regressors combined in groups:
##         Group  room_type : room_typePrivate room room_typeShared room
##
##  Relative importance of 4 (groups of) regressors assessed:
##  room_type number_of_reviews availability_365 minimum_nights
##
## Proportion of variance explained by model: 18.9%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
##                          lmg
## room_type         0.148166616
## number_of_reviews 0.003771222
## availability_365  0.034419437
## minimum_nights    0.002639398
##
## Average coefficients for different model sizes:
##
##                              1group      2groups      3groups      4groups
## number_of_reviews       -0.19816704   -0.1938693   -0.1949841   -0.2033836
## availability_365         0.21250670    0.2185203    0.2252240    0.2324702
## minimum_nights          -0.06827944   -0.2346494   -0.4069745   -0.5766495
## room_typePrivate room -117.64089079 -117.3915297 -117.0340916 -116.5375567
## room_typeShared room  -144.64510472 -147.7185483 -151.2614771 -155.4420360
```

```
# Bootstrap Measures of Relative Importance (1000 samples)
bootresults<-boot.relimp(fit_airbnb, b=1000)
rel_imp <-booteval.relimp(bootresults) # print result
plot(rel_imp) # plot result
```



Predicting the model

```r
predict.lm(fit_airbnb, data.frame(number_of_reviews = 45, availability_365
=365,minimum_nights = 1, room_type = "Entire home/apt"))

##        1
## 281.7818

#Here we have thrown the values sample from our dataset on the model and got
almost same price, i.e. 281.7818
#Thus it shows our multinomial regression model is good to predict price.
```