R Practice 1 ALY6015

R practice week 1 - Module 1 Assignment — Regression Diagnostics with R

by Mohammad Hossein Movahedi

movahed.m@northeastern.edu

20 April 2022

Table Of content

- Table Of content
- Introduction
- Analysis
  - Step 1: importing the dataset
  - Step 2: Perform Exploratory Data Analysis
  - step 3: Preparing the dataset for modeling
  - Step 4 : using the "cor()" function
  - Step 5: Plot of the correlation matrix
  - Step 6: making a scatter plot
    - plot for most correlated
    - Plot for least correlated
    - Plot for variables with near 0.5 correlation
  - Step 7: fitting a regression model in R
  - step 8 : regression equation
  - Step 9: Plotting regression model
  - Step 10: checking model for multicollinearity
  - Step 11: Checking model for outliers
  - Step 12: correcting any issues in the model
  - Step 13: Using the all subsets regression method to identify the "best" model
  - Step 14: Comparing the preferred model from step 13 with my model.
- Conclusion
- Bibliography

Introduction

This R practice mainly focuses on performing exploratory data analysis, interpreting and evaluating a regression model, and fitting two regression models. For this R practice, I will use the AmesHousing dataset provided by the instructor. The referencing will follow APA style also R code used for this report will be included in an appendix. Also, fourteen steps are instructed to be followed for this assignment.

Analysis

for this part, I go through 14 steps that are instructed one by one. but first, I import the necessary libraries

Step 1: importing the dataset

first, I put the CSV file in the folder, then call it.

```
#Step 1 Importing the dataset
data <- read.csv("AmesHousing.csv")
```

Step 2: Perform Exploratory Data Analysis

many things can be done in this step. For this assignment, I will check the structure and summary of the dataset.

```
#step 2 Perform Exploratory Data Analysis
str(data)
summary(data)
```

The dataset is vast. It has 82 variables and 2930 rows; therefore, the exploratory analytics does not help that much, but it is nice.

step 3: Preparing the dataset for modeling

in this part, I will clean the data set through smaller steps. I will replace Na data with the average as instructed. I also used some of the methods mentioned in R-bloggers (R-bloggers, 2021)

First of all, I correct name formats to more readable names

```
#names cleaning
data<-clean_names(data)
colnames(clean)
```

then I remove fully empty columns and rows

```
#removing empty rows and cols
data<-data %>% remove_empty(whic=c("rows"))
data<-data %>% remove_empty(whic=c("cols"))
```

Based on the results, no empty columns or rows existed.

Then I remove duplicate records

```
#removing duplicate rows
data <- data %>% distinct()
```

No duplicate rows were found

At last, I replace the null data with the mean of columns

```
#replacing missing values with mean
library(imputeTS)
data <-na_mean(data)
```

Now the data frame is ready to proceed to the next step.

Step 4 : using the "cor()" function

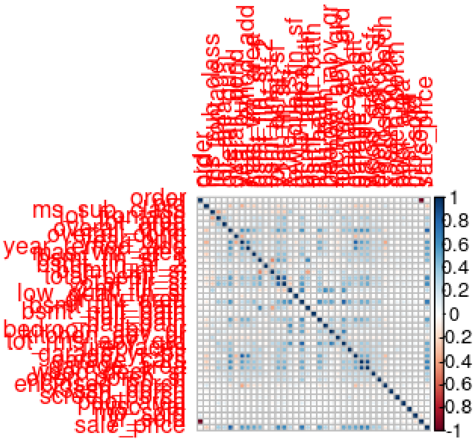
For this part first I create a subset of dataset only containing numeric data and then I use `cor()` function.

```
#step 4 the cor()
datan <- (data[, unlist(lapply(data, is.numeric))])
m = cor(datan)
```

Step 5: Plot of the correlation matrix

this part is very straightforward, but actually, the `corplot()` function has some limitations in that it loses its ability to depict the correlation when more than ten variables are compared at the same time; therefore, just using it on our row data set will give us the plot below.

```
#step 5 corplot
corrplot(m)
```



corrplot() on all variables at the same time

As it can be seen, the graph is useless, and the best approach here is to ask an expert in the field to look at the dataset to determine the group of variables that he thinks are related, and then we do a separate `corrplot` on them.

Step 6: making a scatter plot

For this part, first, we have to find the variables with the least and most correlation with `SalePrice`.

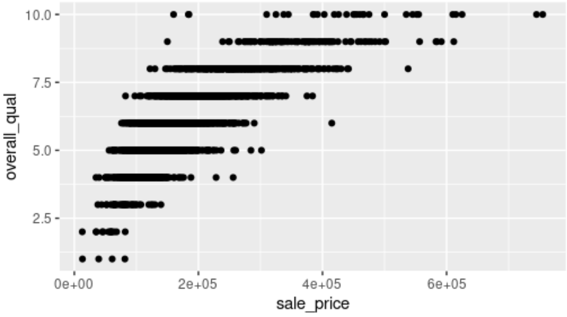
```
#finding variables correlation
m2 = cor(datan[-39], datan$sale_price)
```

The result shows that `overall_qual` is the most correlated variable to `SalePrice` while `misc_val` has the most little correlation with `SalePrice`. Also, the `mas_vnr_area` variable has an almost 0.5 correlation with `SalePrice`. Therefore the below three scatter plots are created of these variables.

plot for most correlated

```
#scatter plot for most correlated
ggplot(datan, aes(x=sale_price, y=overall_qual)) + geom_point()
```

As seen in the graph, although there is a correlation between data, the variance keeps increasing as variables increase, making this correlation unsuitable for regression modeling alone.

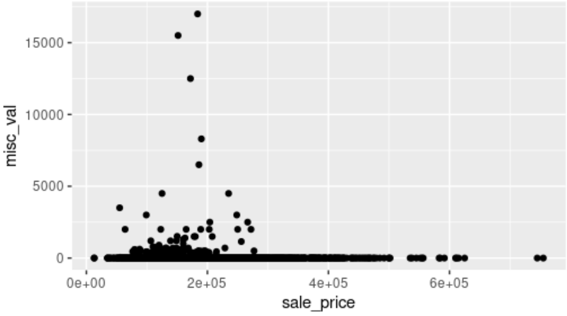


Scatter plot for most related variables

Plot for least correlated

```
#scatter plot for least correlated
ggplot(datan, aes(x=sale_price, y=misc_val)) + geom_point()
```

As it can be seen there is no visible connection between two variables

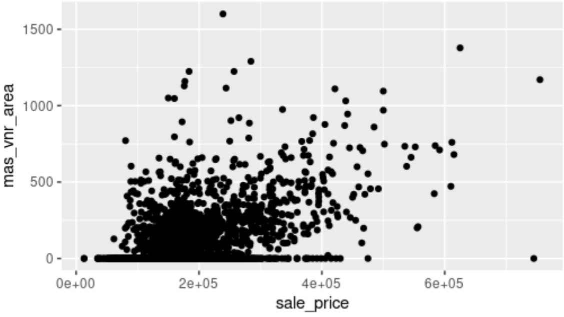


lest related variables

Plot for variables with near 0.5 correlation

```
#scatter plot for 0.5 correlation
ggplot(datan, aes(x=sale_price, y=mas_vnr_area)) + geom_point()
```

As it can be seen, there is a correlation between variables but not a strong one per se.



Plot for 0.5 correlation

Step 7: fitting a regression model in R

For this part first I find the most correlated variables based on `corr()` function and then I fit a regression model on one of them

```
#rearranging cor matrix and removing duplicates
msort <- m2 %>%
  as.data.frame() %>%
  mutate(var1 = rownames(.)) %>%
  gather(var2, value, -var1) %>%
  arrange(desc(value)) %>%
  group_by(value) %>%
  filter(row_number()==1)

msort$value <- abs(msort$value)
msort<-msort[!(msort$value==1),]
msort <-arrange(msort,desc(value))
head(msort,3)
```

the result is shown in the table below

Var2	Corr
overall_qual	0.799
gr_liv_area	0.707
garage_cars	0.648

then I create the regression Model

```
#creating regression model
reg <- lm(sale_price ~ overall_qual + gr_liv_area + garage_cars, data = datan)
summary(reg)
```

the result of summery is shown below

```
Call:
lm(formula = sale_price ~ overall_qual + gr_liv_area + garage_cars,
data = datan)
```

```
Residuals:
Min      1Q  Median      3Q     Max
-352157  -22484   -2008   19487  292498
```

```
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.043e+05 3.301e+03 -31.60 <2e-16 ***
overall_qual 2.819e+04 7.076e+02 39.84 <2e-16 ***
gr_liv_area 5.230e+01 1.812e+00 28.86 <2e-16 ***
garage_cars 1.970e+04 1.236e+03 15.94 <2e-16 ***
```

```
Signif. codes:  0 ‘’ 0.001 ‘’ 0.01 ‘’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 39690 on 2926 degrees of freedom
Multiple R-squared:  0.7535,  Adjusted R-squared:  0.7532
F-statistic:  2981 on 3 and 2926 DF,  p-value: < 2.2e-16
```

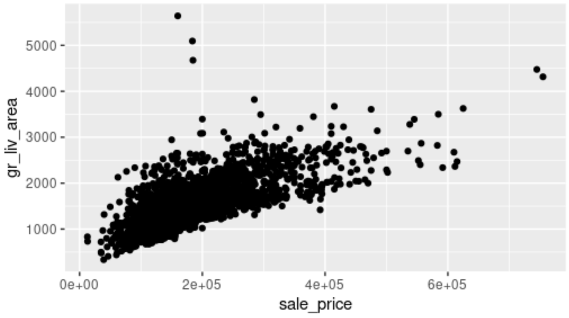
As it can be seen adjusted R-squared is very close to R-squared showing that the predicors didn't improve the model that much.

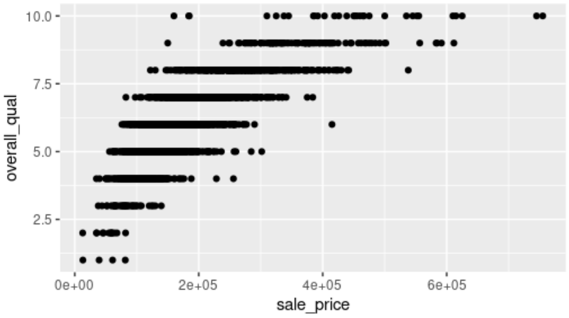
step 8 : regression equation

$$sale.price = -1.043e^5 + 2.819e^4(overall.qual) + 5.230e^1(gr.liv.area) + 1.970e^4(garage.cars)$$

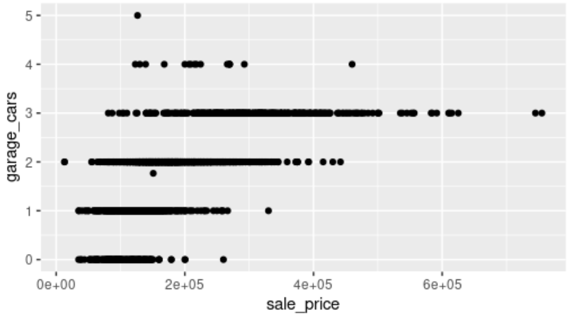
the equation shows that `overall qual` and `garage cars` have more influence on sale price than `gr liv area`

Step 9: Plotting regression model





the graphs show that for `gr_liv_area` there is a continuous relationship also there are some outlier in each variables that they need to be deleted.



Step 10: checking model for multicollinearity

for this part, first, I do the Farrar – Glauber Test

```
# step 10 checking multicollinearity
library(mctest)
omcdiag(reg)
```

the results are shown below

```
Call:
omcdiag(mod = reg)
```

```
Overall Multicollinearity Diagnostics

                       MC Results detection
Determinant |X'X|:         0.4105         0
Farrar Chi-Square:      2606.0085         1
Red Indicator:             0.5549         1
Sum of Lambda Inverse:     5.0572         0
Theil's Method:           -0.2959         0
Condition Number:         14.0380         0

1 --> COLLINEARITY is detected by the test
0 --> COLLINEARITY is not detected by the test
```

The value of the standardized determinant is 0.41, which is a good thing also. The calculated value of the Chi-square test statistic is 2606.0085, and it is highly significant, implying the presence of multicollinearity in the model specification.

therefore I do the F – test

```
> imcdiag(reg)

Call:
imcdiag(mod = reg)


All Individual Multicollinearity Diagnostics Result

                VIF    TOL        Wi       Fi Leamer    CVIF Klein  IND1   IND2
overall_qual 1.8539 0.5394 1249.6186 2500.091 0.7344 -0.8190     0 4e-04 1.1410
gr_liv_area  1.5604 0.6408  820.1964 1640.953 0.8005 -0.6894     0 4e-04 0.8897
garage_cars  1.6429 0.6087  940.8520 1882.347 0.7802 -0.7258     0 4e-04 0.9694

1 --> COLLINEARITY is detected by the test
0 --> COLLINEARITY is not detected by the test

* all coefficients have significant t-ratios

R-square of y on all x: 0.7535

* use method argument to check which regressors may be the reason of collinearity
===================================
```

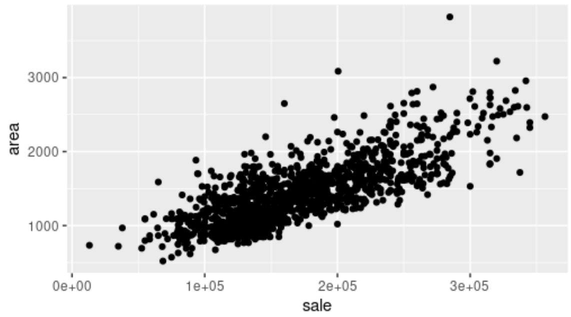
The F value for all variables is very High, indicating they all cause the multicollinearity in the model. as it turns out, the model that I used isn’t helpful due to multicollinearity. Therefore it can’t be used in the future.

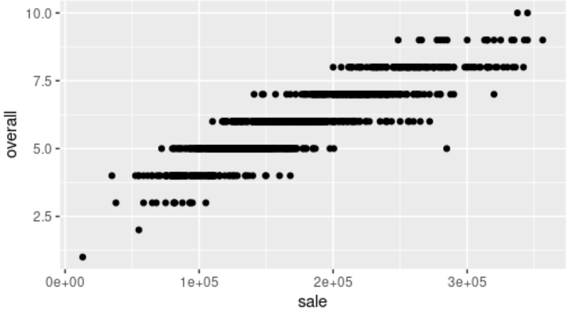
Step 11: Checking model for outliers

In this step, I have to choose the threshold to be able to determine outliers. I choose the threshold based on 1.5 IQRs to use. (user3816990, 2015)

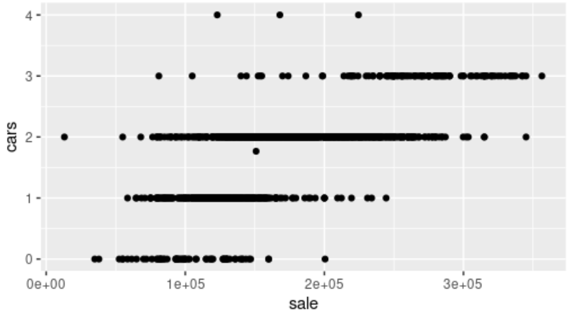
```
#step 11 dealing with outliesrs
# Create data frame for regression model
rdata <- data.frame(order = datan$order,sale = datan$sale_price,overall = datan$overall_qual,area = datan$gr_liv_area,
                    cars = datan$garage_cars)

# Calculate residuals
rdata$residuals <- as.numeric(residuals(reg))

# Choose a threshold
outlier_threshold <- 19487

# Print only names of outliers
outliers <- rdata[ abs(rdata$residuals) > outlier_threshold, ]
print(outliers$order)
rodata <-filter(rdata,!order %in% outliers$order,)
#ploting
ggplot(rodata, aes(x=sale, y=overall)) + geom_point()
ggplot(rodata, aes(x=sale, y=area)) + geom_point()
ggplot(rodata, aes(x=sale, y=cars)) + geom_point()
```

There were 1500 data outliers with this method, but after cleaning does data, the model plots start to look better.





As it can be seen, there is more visible relation between data now.



I believe the observation should only be deleted in extreme cases because there is no way to be sure the data is outliers or not.

Step 12: correcting any issues in the model

as It turns out, in step 10, the model that I choose works poorly, so I would opt not to use it to predict the price and instead look for other models to implement.

Step 13: Using the all subsets regression method to identify the "best" model

to work in this step I used the `library(leaps)` to find the best model.(Sthda.com, 2018)

```
#step 13 leap of faith into submodels
models <- regsubsets(sale_price~., data = datan, nvmax = 5)
summary(models)
```

the results of summery indicate that the best model for predicting `sale_price` using three variables can be created based on `overall_qual` , `bsmt_fin_sf_1`, and `gr_liv_area`

Therefore, although the model that I used wasn’t the best, it was close to it.

Step 14: Comparing the preferred model from step 13 with my model.

I used `anova()` function to compare the two models.(Phillips, 2018)

```
#step 14 the part it ends in comparing
s13model <-lm(sale_price ~ overall_qual + gr_liv_area + bsmt_fin_sf_1, data = data)
anova(reg,s13model)
```

the results are shown below

```
> anova(reg,s13model)
Analysis of Variance Table

Model 1: sale_price ~ overall_qual + gr_liv_area + garage_cars
Model 2: sale_price ~ overall_qual + gr_liv_area + bsmt_fin_sf_1
  Res.Df        RSS Df Sum of Sq F Pr(>F)
1   2926 4.6085e+12
2   2926 4.2865e+12  0  3.22e+11
```

As can be seen, the residuals of the second model are more insignificant than the initial model indicating that it is a better model as it has more accuracy in prediction.

Conclusion

In this assignment, I learned that although the correlation can be a good indicator for choosing variables to create a regression model, it is not enough. Other measures like subsetting models should also be taken.

Bibliography

“Best Subsets Regression Essentials in R - Articles - STHDA.” *Sthda.com*, 11 Mar. 2018, www.sthda.com/english/articles/37-model-selection-essentials-in-r/155-best-subsets-regression-essentials-in-r/. Accessed 23 Apr. 2022.

“How to Clean the Datasets in R? | R-Bloggers.” *R-Bloggers*, 4 Apr. 2021, www.r-bloggers.com/2021/04/how-to-clean-the-datasets-in-r/. Accessed 23 Apr. 2022.

user3816990. “Identify All Outliers from Regression Analysis.” *Stack Overflow*, 4 Feb. 2015, stackoverflow.com/questions/28329321/identify-all-outliers-from-regression-analysis. Accessed 23 Apr. 2022.

Appendix

```
print('Mohammad Hossein Movahedi')
print('R practice week 1')
#importing and instaling libraries
#importing and instaling libraries
install.packages('FSA')
install.packages('magrittr')
install.packages('dplyr')
install.packages('tidyr')
install.packages('plyr')
install.packages('tidyverse')
install.packages('outliers')
install.packages('ggplot2')
install.packages('lubridate')
install.packages("janitor")
install.packages('imputeTS')
install.packages('corrplot')
install.packages('mctest')

install.packages('leaps')

library(FSA)
library(magrittr)
library(dplyr)
library(tidyr)
library(plyr)
library(tidyverse)
library(scales)
library(lubridate)
library(ggplot2)
library(outliers)
library(janitor)
library(corrplot)

library(leaps)


#Step 1 Importing dataset
data <- read.csv("AmesHousing.csv")


#step 2 perform Exploratory Data Analysis
str(data)
summary(data)

#step 3 data cleaning
#names cleaning
data<-clean_names(data)
colnames(clean)
#removing empty rows and cols
data<-data %>% remove_empty(whic=c("rows"))
data<-data %>% remove_empty(whic=c("cols"))
#removing duplicate rows
data <- data %>% distinct()
#replacing missing values with mean
library(imputeTS)
data <-na_mean(data)

#step 4 the cor()
datan <- (data[, unlist(lapply(data, is.numeric))])
m = cor(datan)
#step 5 corplot
corrplot(m)
#step 6 scater plot
#finding variables correlation
m2 = cor(datan[-39], datan$sale_price)
#scatter plot for most correlated
ggplot(datan, aes(x=sale_price, y=overall_qual)) + geom_point()
#scatter plot for least correlated
ggplot(datan, aes(x=sale_price, y=misc_val)) + geom_point()
#scatter plot for 0.5 correlation
ggplot(datan, aes(x=sale_price, y=mas_vnr_area)) + geom_point()


#step 7 fitting regression model
#first rearrenging cor matrix and remove duplicates
msort <- m2 %>%
  as.data.frame() %>%
  mutate(var1 = rownames(.)) %>%
  gather(var2, value, -var1) %>%
  arrange(desc(value)) %>%
  group_by(value) %>%
  filter(row_number()==1)

msort$value <- abs(msort$value)
msort<-msort[!(msort$value==1),]
msort <-arrange(msort,desc(value))
head(msort,3)
#creating regression model
reg <- lm(sale_price ~ overall_qual + gr_liv_area + garage_cars, data = datan)
summary(reg)

#step 9 plotting
ggplot(datan, aes(x=sale_price, y=overall_qual)) + geom_point()
ggplot(datan, aes(x=sale_price, y=gr_liv_area)) + geom_point()
ggplot(datan, aes(x=sale_price, y=garage_cars)) + geom_point()

# step 10 checking multicollinearity
library(mctest)
omcdiag(reg)
imcdiag(reg)
#step 11 dealing with outliesrs
```

```
# Create data frame for regression model
rdata <- data.frame(order = datan$order,sale = datan$sale_price,overall = datan$overall_qual,area = datan$gr_liv_area,
                    cars = datan$garage_cars)

# Calculate residuals
rdata$residuals <- as.numeric(residuals(reg))

# Choose a threshhold
outlier_threshold <- 19487

# Print only names of outliers
outliers <- rdata[ abs(rdata$residuals) > outlier_threshold, ]
print(outliers$order)
rodata <-filter(rdata,!order %in% outliers$order,)
#ploting
ggplot(rodata, aes(x=sale, y=overall)) + geom_point()
ggplot(rodata, aes(x=sale, y=area)) + geom_point()
ggplot(rodata, aes(x=sale, y=cars)) + geom_point()

#step 13 leap of faith into submodels
models <- regsubsets(sale_price~., data = datan, nvmax = 5)
summary(models)
#step 14 the part it ends in compareing
s13model <-lm(sale_price ~ overall_qual + gr_liv_area + bsmt_fin_sf_1, data = datan)
anova(reg,s13model)
```