Regression

2022-10-20

Regrssion on Diamonds Dataset

For my dataset I used "diamonds" from the tidyverse library. It has over 50000 observations and the data is useful for data analysis.

Reading in the Data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
df <-(diamonds)
head(df)
```

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
colnames(df)
```

```
##  [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"   "price"
##  [8] "x"       "y"       "z"
```

#Split data

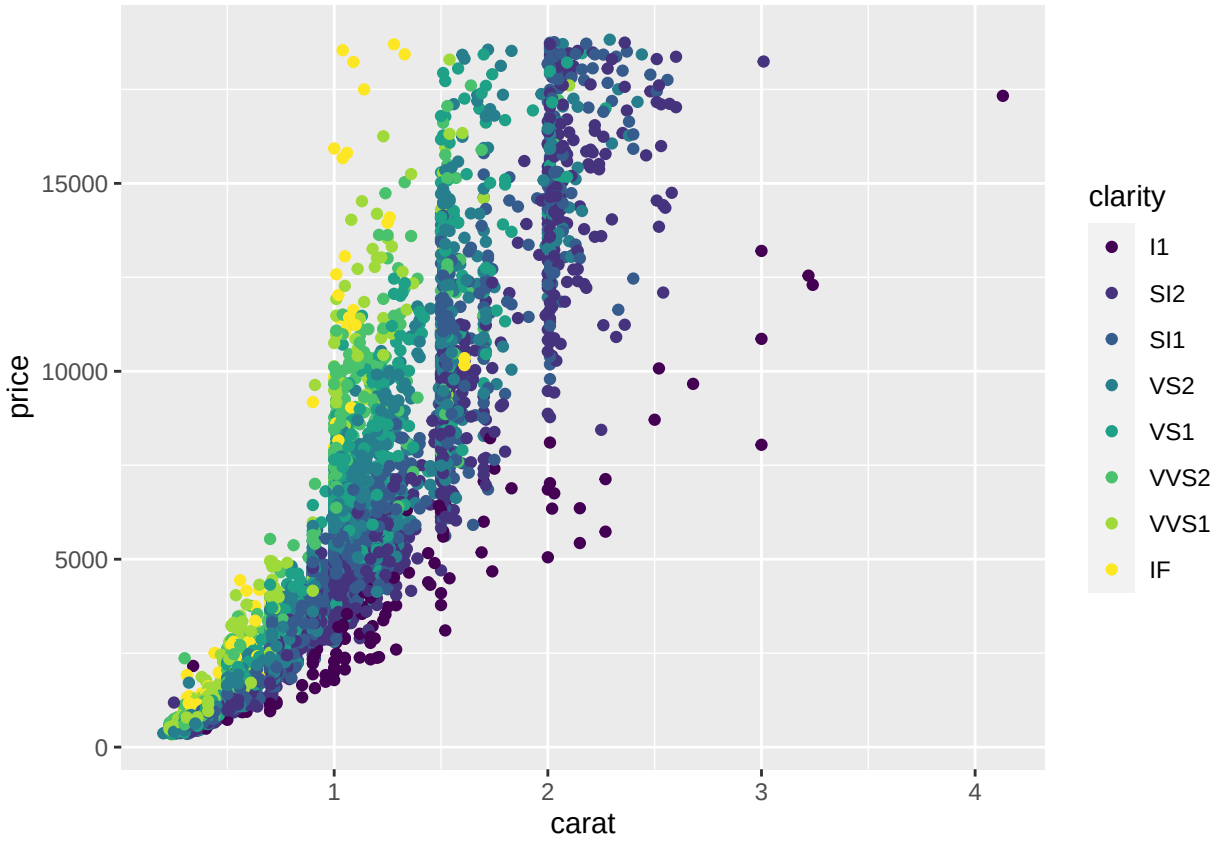
Now we will split the data 80/20. Into train and test data.

```
set.seed(1234)
cut<- sample(1:nrow(df), nrow(df)*0.2, replace= FALSE)
smaller <- df[cut,]
i <- sample(1:nrow(smaller), nrow(smaller)*0.8, replace=FALSE)
train <- smaller[i,]
test <- smaller[-i,]
```

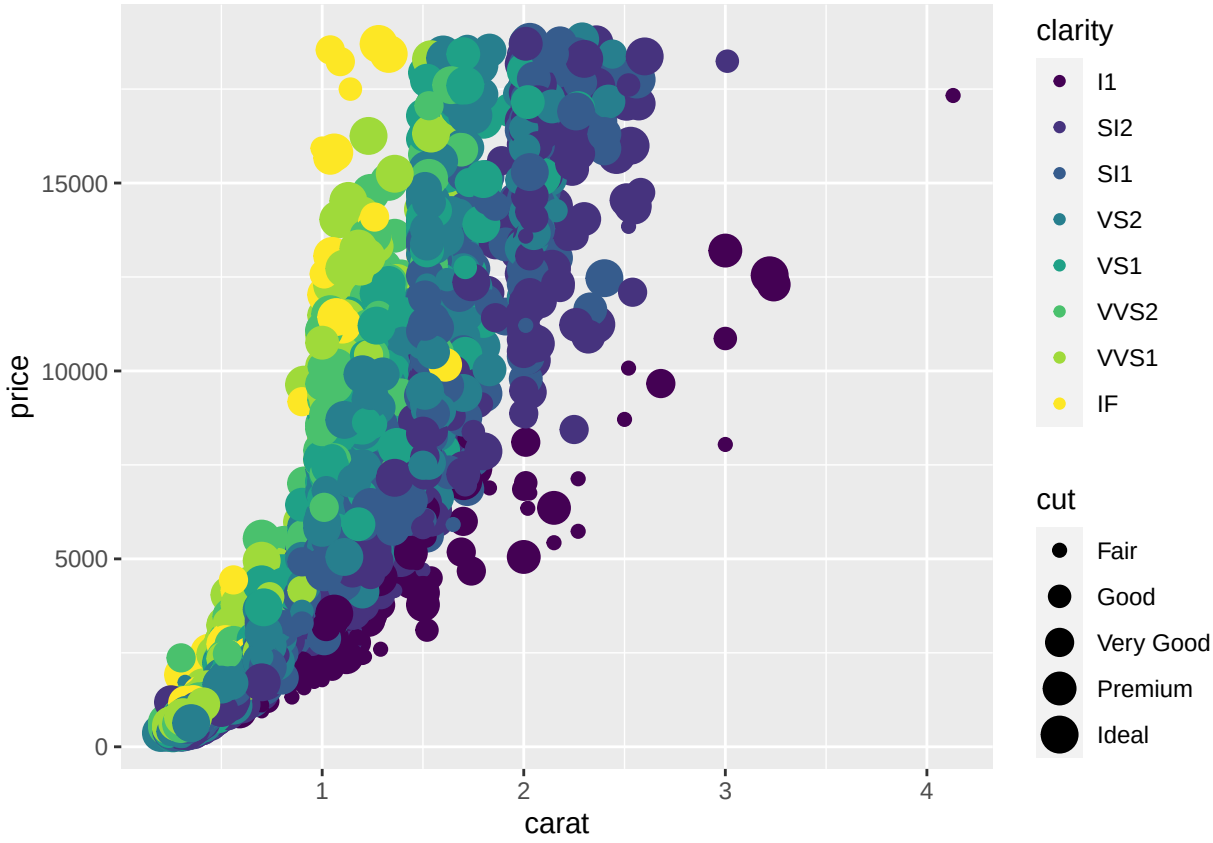
#Plotting the training data

Next, we will plot the data:

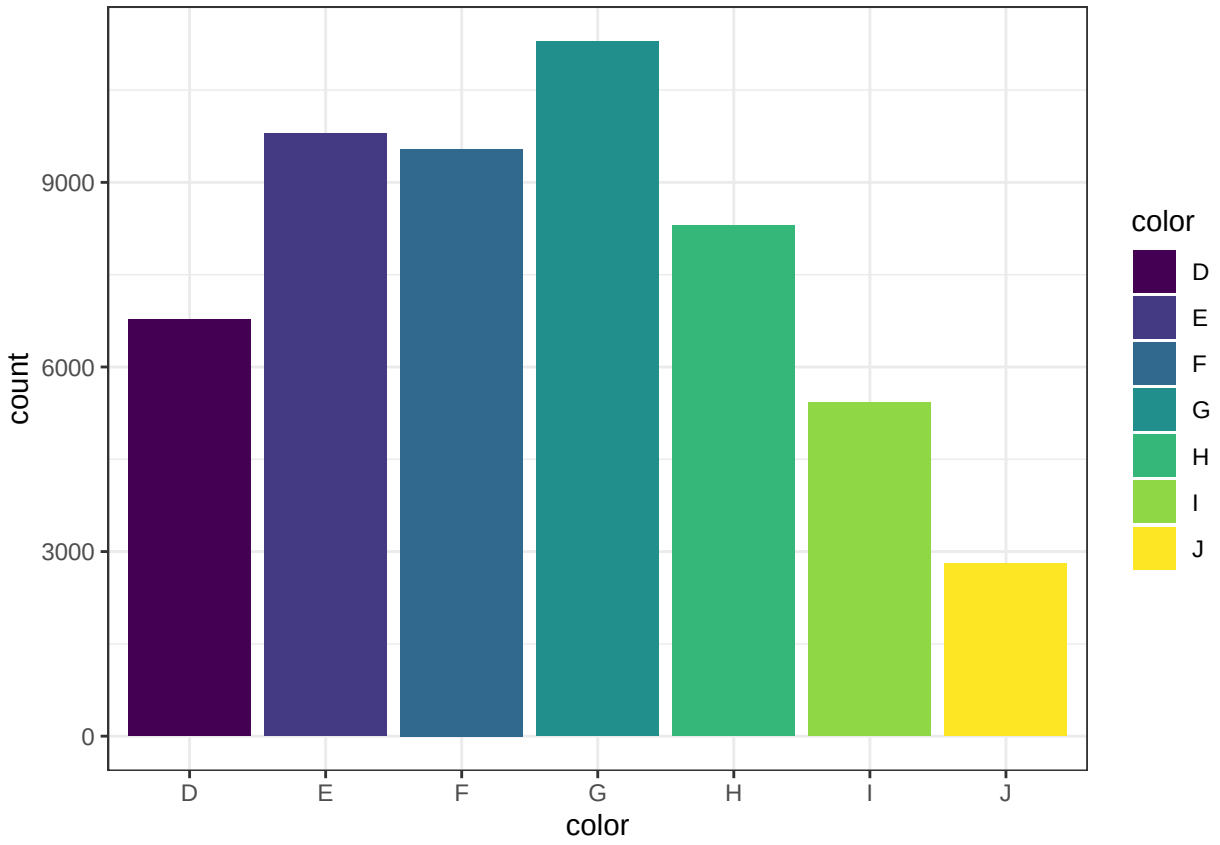
```
ggplot(train, aes(x=carat, y=price, color=clarity))+geom_point()
```




```
ggplot(train, aes(x=carat, y=price, color=clarity, size=cut))+geom_point()
```




```
ggplot(diamonds, aes(x=color, fill=color))+ theme_bw()+geom_bar()
```



#Regression

Performing SVM Regression:

```
library(e1071)
library(MASS)
```

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

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

```
svmRegression <- svm(price~carat+depth+table+x+y+z, data=train)
svmRegression
```

```
##
## Call:
## svm(formula = price ~ carat + depth + table + x + y + z, data = train)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
```

```
##        cost:  1
##       gamma:  0.1666667
##     epsilon:  0.1
##
##
## Number of Support Vectors:  3684
```

```
svmLinear <- tune(e1071::svm, price~carat+depth+table+x+y+z, data=train, kernel="linear", ranges= list(
summary(svmLinear)
```

```
##
## Parameter tuning of 'e1071::svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   0.1
##
## - best performance: 2339433
##
## - Detailed performance results:
##    cost   error dispersion
## 1   0.1 2339433   418173.9
## 2   1.0 2361318   442388.5
## 3  10.0 2363615   445786.2
## 4 100.0 2367579   452219.2
```

```
svmPolynomial <- tune(e1071::svm,price~carat+depth+table+x+y+z, data=train, kernel="polynomial", ranges=
summary(svmPolynomial)
```

```
##
## Parameter tuning of 'e1071::svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   0.1
##
## - best performance: 4801464
##
## - Detailed performance results:
##    cost      error dispersion
## 1   0.1    4801464    1161742
## 2   1.0   29124422   70638166
## 3  10.0  326561672  991867789
## 4 100.0 2283788916 6875122722
```