2023 ADS2 Week 7 Data Cleaning Problem Set Notes

ADS2

2023-11-06

We hope that you have run all the scripts in this week's Problem Set! Now let's see how you went on in cleaning the data.

Import data

There are serval function to import data (including: `read.delim, read.delim2, read.csv, read.csv2`).

Since our data is in the `xxx.csv` format, so we can use the `read.csv()` function

```
diamond_sample = read.csv("Rdata_diamonds_samples100_mdf.csv")
```

Data Cleaning

Now let's try to clean our data. Still remember the **screen-diagnosis-treat-document** rules?

Missing data

Screen-Diagnosis

We first want to screen the missing data. We want to know, how the missing data is coded? (`NA` or `NAN` or `ND` or `blank`)? which rows have a missing data? which column have a missing data?

```
anyNA(diamond_sample)
```

```
## [1] TRUE
```

```
head(diamond_sample)
```

```
##   X carat     cut color clarity depth table price    x    y    z
## 1 1  2.01    Fair     G     SI1  70.6  64.0 18574 7.43 6.64 4.69
## 2 2  2.08   Ideal     J    <NA>  61.0  55.0 17986 8.32 8.25 5.05
## 3 3  1.22   Ideal     G     VS2  61.4  56.0  8362 6.90 6.88 4.23
## 4 4  0.82 Premium  <NA>     SI1  62.4  56.0    NA 6.01 5.98 3.74
## 5 5  0.32   Ideal     D     SI1  62.7  54.0   589 4.35 4.39 2.74
## 6 6  0.34   Ideal     E     VS2  62.1  54.1   758 4.47 4.50 2.78
```

```r
head(is.na(diamond_sample))
```

```
##          X carat   cut color clarity depth table price     x     y     z
## [1,] FALSE FALSE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE    TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE FALSE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE FALSE FALSE  TRUE   FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## [5,] FALSE FALSE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [6,] FALSE FALSE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```r
tail(diamond_sample)
```

```
##       X carat       cut color clarity depth table price    x    y    z
## 97   97  1.02      Good     D     VS2    NA  58.0  7539 6.36 6.40 4.03
## 98   98  1.52      <NA>     J     SI1  60.4  59.0  5618 7.44 7.39 4.48
## 99   99  1.71 Very Good     D     SI2  63.5  59.0 11873   NA 7.59 4.81
## 100 100  0.44 Very Good     E     VS1  59.7  59.0  1090 4.94 5.01 2.97
## 101  14  0.58      Good     E     VS1  61.5  61.9  2041 5.30 5.34 3.27
## 102  99  1.71 Very Good     D     SI2  63.5  59.0 11873   NA 7.59 4.81
```

```r
tail(is.na(diamond_sample))
```

```
##            X carat   cut color clarity depth table price     x     y     z
##  [97,] FALSE FALSE FALSE FALSE   FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [98,] FALSE FALSE  TRUE FALSE   FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [99,] FALSE FALSE FALSE FALSE   FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## [100,] FALSE FALSE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [101,] FALSE FALSE FALSE FALSE   FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [102,] FALSE FALSE FALSE FALSE   FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
```

```r
apply(is.na(diamond_sample), 2, which) #this is to find the row,col of NA
```

```
## $X
## [1] 77 89
##
## $carat
## [1] 50 76
##
## $cut
## [1] 31 98
##
## $color
## [1]  4 63
##
## $clarity
## [1]  2 32 59 85
##
## $depth
## [1] 10 88 97
##
## $table
```

```
## integer(0)
##
## $price
## [1] 4
##
## $x
## [1]  86  99 102
##
## $y
## [1] 19 80
##
## $z
## [1] 72 74 83
```

Looks like our missing data is coded with `NA`. As we don't know how to handle those missing data (no data to fill it up), the treat we would like to take is to remove those entries. However, it is not the only possible option as you may wish to analyse all the available data even it is incomplete – it depends on your experiment, and you need to think about it ahead.

Treat

```
dim(diamond_sample)
```

```
## [1] 102  11
```

```
data.noNA = diamond_sample[complete.cases(diamond_sample), ]
dim(data.noNA)
```

```
## [1] 79 11
```

Document

We found several missing values in the data set. here are ID values of the respective diamonds: 2, 4, 10, 19, 31, 32, 50, 59, 63, 72, 74, 76, NA, 80, 83, 85, 86, 88, NA, 97, 98, 99, 99. As there are no clues about how to fill missing values, it is best to remove these cases. After removing incomplete cases, the data set reduced from 102 rows to 79 rows.

Duplicated data

Screen-Diagnosis

We then want to screen the duplicated data. We want to know whether there is **duplicated rows** in the dataset?

```
duplicated(data.noNA)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [73] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
```

```
frw.idx = which(duplicated(data.noNA)) #duplciated() will only give you the duplicated rows,
# but not the original rows, so we need the next line to get the originals

rvs.idx = which(duplicated(data.noNA, fromLast = TRUE))
data.noNA[c(frw.idx, rvs.idx), ]
```

```
##      X carat  cut color clarity depth table price   x    y    z
## 101 14  0.58 Good     E     VS1  61.5  61.9  2041 5.3 5.34 3.27
## 14  14  0.58 Good     E     VS1  61.5  61.9  2041 5.3 5.34 3.27
```

Treat

Since duplicated entries are obviously errors, we need to delete them from the table.

```
dim(data.noNA)
```

```
## [1] 79 11
```

```
data.noNA.noDup = data.noNA[!duplicated(data.noNA),]
dim(data.noNA.noDup)
```

```
## [1] 78 11
```

This time, no duplicates in the data.

Documet

We found several duplicated values in the data set: c(14, 14), c(0.58, 0.58), c("Good", "Good"), c("E", "E"), c("VS1", "VS1"), c(61.5, 61.5), c(61.9, 61.9), c(2041, 2041), c(5.3, 5.3), c(5.34, 5.34), c(3.27, 3.27)

We decided to delete these values. After cleaning, we have got 78 row (1 row removed).

Strange pattern

After removing the missing data and duplicated data, we now want to see whether there are any outliers or strange patterns. You can work on outliers in this problem set on your own. Let's see if there is any strange pattern.

Screen

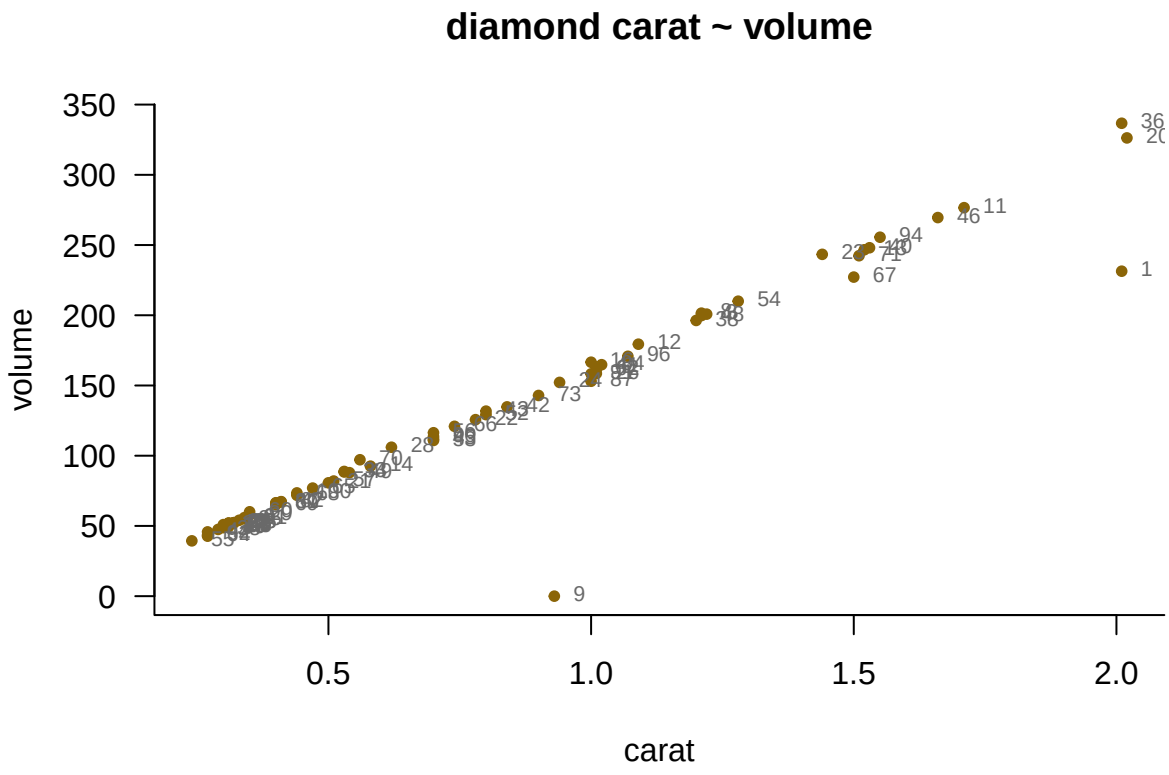
We know that diamond and it's volume have a linear relationship. Thus, we would like to investigate the relationship between carat vs. volume. In order to test this idea, we need to generate a new vector called `volume = x * y * z`.

```
data.noNA.noDup$volume = data.noNA.noDup$x * data.noNA.noDup$y * data.noNA.noDup$z %>%
  round(2)
head(data.noNA.noDup)
```

```
##   X carat       cut color clarity depth table price    x    y    z    volume
## 1 1  2.01      Fair     G     SI1  70.6  64.0 18574 7.43 6.64 4.69 231.38209
## 3 3  1.22     Ideal     G     VS2  61.4  56.0  8362 6.90 6.88 4.23 200.80656
## 5 5  0.32     Ideal     D     SI1  62.7  54.0   589 4.35 4.39 2.74  52.32441
## 6 6  0.34     Ideal     E     VS2  62.1  54.1   758 4.47 4.50 2.78  55.91970
## 7 7  0.31     Ideal     H     VS2  62.2  55.0   628 4.37 4.35 2.71  51.51574
## 8 8  1.21 Very Good     G    VVS2  60.4  56.0  9878 6.96 6.91 4.19 201.51218
```

we then plot scatterplot of carat vs volume to see whether they have a linear relationship.

```
plot(x = data.noNA.noDup$carat, y = data.noNA.noDup$volume,
     pch = 20, col = "darkgoldenrod4",
     las = 1, xlab = "carat", ylab = "volume",
     main = "diamond carat ~ volume", bty = "l")
text(data.noNA.noDup$carat, data.noNA.noDup$volume,
     labels = data.noNA.noDup$X, col = "dimgray",
     cex = 0.7, pos = 4)
```



Diagnosis

Here we found a few strange data points, IDs 1 and 9. We decide to take a look at these IDs.

```
data.noNA.noDup[which(data.noNA.noDup$carat > 1.9 &
                        data.noNA.noDup$volume < 250), ]
```

```
##   X carat  cut color clarity depth table price    x    y    z   volume
## 1 1  2.01 Fair     G     SI1  70.6    64 18574 7.43 6.64 4.69 231.3821
```

```
data.noNA.noDup[which(data.noNA.noDup$volume < 20), ]
```

```
##   X carat       cut color clarity depth table price    x       y       z volume
## 9 9  0.93 Very Good     G     SI1  61.7    56  4513 6.22 0.00626 0.00385      0
```

```
data.noNA.noDup[data.noNA.noDup$X == 9, c(9:11)]
```

```
##      x       y       z
## 9 6.22 0.00626 0.00385
```

From this results, we find ID1 as having nothing special, even though its weight is unusual. It may be just a peculiar diamond or a mistake.

But ID9 apparently does not look good: numeric(0), numeric(0), numeric(0). Y and Z dimensions have values below zero, which is practically barely possible. It might be a mistake, a typo, or a really unique diamond. If you cannot figure out the reason for this unusual entry, best would be to remove it from the following analysis.

Treat

From our analysis on the carat vs volume, we found the data for ID=9 is a bit strange. We could leave it for further analysis, but for the illustrational purposes we will delete this data point as well.

```
data.noNA.noDup.noStrg = data.noNA.noDup[-which(data.noNA.noDup$X == 9), ]
dim(data.noNA.noDup)
```

```
## [1] 78 12
```

```
dim(data.noNA.noDup.noStrg)
```

```
## [1] 77 12
```

Document

We found two entries that are very unusual: ID1 and ID9. Of these two, ID9 has values of several dimensions below zero, which looks like an error. Thus, we decided to remove it. ID1 is also different compared with other entries, but we do not see enough evidence to exclude it right away.

Thus, we removed one more row (ID9) and we have 77 entries now.

Correct typos in the dataset.

After running all the procedure above (clean missing, duplicated, strange data), first we want to see whether there is any typo. **Please check those character/factor vectors in the diamonds data, see whether you can find any typos and then correct those typo in R? Remember to document any edit you do properly. Use screen-diagnosis-treat-document strategy.**

Screen

```
table(data.noNA.noDup.noStrg$cut)
```

```
##
##      Fair      Good      Idea     Ideal   Premium Very Good
##         8         6         1        30        18        14
```

`Idea`? What an unusual label! Let's have a closer look at this value.

Diagnosis

```
data.noNA.noDup.noStrg %>%
  filter(cut == "Idea")
```

```
##    X carat  cut color clarity depth table price    x    y    z   volume
## 1 95  0.33 Idea     G      IF  61.7    58   968 4.42 4.46 2.74 54.01417
```

Looks like a clear typo. Need to correct it.

Treat

```
data.noNA.noDup.noStrg.noTypo <- data.noNA.noDup.noStrg
data.noNA.noDup.noStrg.noTypo[data.noNA.noDup.noStrg.noTypo$X == 95,
                              "cut"] <- "Ideal"
table(data.noNA.noDup.noStrg.noTypo$cut)
```

```
##
##      Fair      Good     Ideal   Premium Very Good
##         8         6        31        18        14
```

Document

ID95 had a typo in the `cut` column: it was labeled as `Idea`. We changed it to `Ideal`.

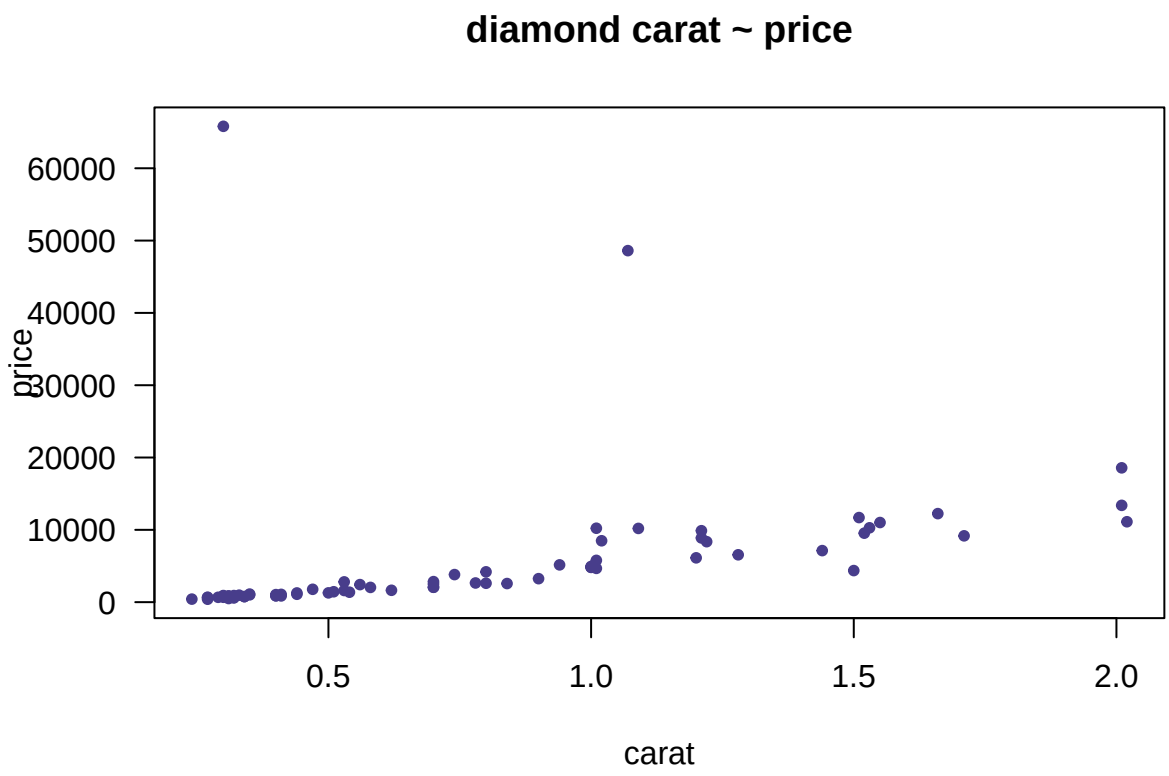
Find outliers in the dataset.

After removing the missing data, duplicated data, strange data and typos, we now want to see whether there is any outliers. For example, is there any outlier if we investigate the relationship between carat vs price. Since we know, the diamond price is positively correlate with its carat! (the bigger the diamond is the more expensive).

Hint: if the data looks suspicious, and you don't know whether you should remove it or not, you can generate a new indicator vector to the dataframe to indicate whether this observation is suspicious (but you don't have evidence to delete it).

Screen

```
plot(x = data.noNA.noDup.noStrg.noTypo$carat,
     y = data.noNA.noDup.noStrg.noTypo$price,
     pch = 20, col = "darkslateblue",
     las = 1, xlab = "carat", ylab = "price",
     main = "diamond carat ~ price")
text(data.noNA.noDup.noStrg.noTypo$carat,
     data.noNA.noDup.noStrg.noTypo$price,
     labels = data.noNA.noDup.noStrg.noTypo$ID,
     col = "dimgray",
     cex = 0.7, pos = 4)
```



Diagnose

From the scatterplot above, we realize that very likely there maybe two outliers, ID96 and ID27. So let's take a more detail view on this two ID.

```
data.noNA.noDup.noStrg.noTypo %>%
  filter(price > 40000)
```

```
##    X carat   cut color clarity depth table price    x    y    z    volume
## 1 27  0.30 Ideal     E     VS2  61.5    55 65800 4.32 4.40 2.68  50.94144
## 2 96  1.07  Good     G     SI1  63.1    59 48610 6.45 6.49 4.08 170.79084
```

Even those two outliers look very suspicious, but we didn't have evidence to show it's wrong.So we decide to let's view some IDs surrondings to those two strange IDs.

```
data.noNA.noDup.noStrg.noTypo[which(data.noNA.noDup.noStrg.noTypo$X ==
                                    27-2):which(
                                      data.noNA.noDup.noStrg.noTypo$X == 27+2),]
```

```
##     X carat     cut color clarity depth table price    x    y    z    volume
## 25 25  0.31   Ideal     G     VS2  60.8    56   500 4.40 4.42 2.68  52.12064
## 26 26  1.01 Premium     D    VVS2  62.4    60 10221 6.31 6.36 3.95 158.51982
## 27 27  0.30   Ideal     E     VS2  61.5    55 65800 4.32 4.40 2.68  50.94144
## 28 28  0.62    Fair     F     SI1  55.1    66  1641 5.85 5.70 3.18 106.03710
## 29 29  0.35 Premium     G      IF  59.1    59  1116 4.62 4.59 2.72  57.67978
```

```
tail(data.noNA.noDup.noStrg.noTypo)
```

```
##       X carat       cut color clarity depth table price    x    y    z
## 92   92  0.27     Ideal     E    VVS2  62.5    57   622 4.10 4.13 2.57
## 93   93  0.31     Ideal     F     VS2  61.3    57   802 4.34 4.30 2.65
## 94   94  1.55     Ideal     F     SI1  61.3    56 11011 7.45 7.49 4.58
## 95   95  0.33     Ideal     G      IF  61.7    58   968 4.42 4.46 2.74
## 96   96  1.07      Good     G     SI1  63.1    59 48610 6.45 6.49 4.08
## 100 100  0.44 Very Good     E     VS1  59.7    59  1090 4.94 5.01 2.97
##        volume
## 92   43.51781
## 93   49.45430
## 94  255.56629
## 95   54.01417
## 96  170.79084
## 100  73.50572
```

Treat

We have two outlier ID=96 and ID=27, but we don't have evidence to show they are wrong, so we can add a vector to indicate whether they are suspicious.

```
outlier.idx = rep(0, nrow(data.noNA.noDup.noStrg.noTypo))
outlier.idx[which(data.noNA.noDup.noStrg.noTypo$X == 27)] = 1
outlier.idx[which(data.noNA.noDup.noStrg.noTypo$X == 96)] = 1

data.noNA.noDup.noStrg.notypo.mkOtlr = data.frame(data.noNA.noDup.noStrg.noTypo,
                                                  otlr = outlier.idx)
head(data.noNA.noDup.noStrg.notypo.mkOtlr)
```

```
##   X carat       cut color clarity depth table price    x    y    z    volume
## 1 1  2.01      Fair     G     SI1  70.6  64.0 18574 7.43 6.64 4.69 231.38209
## 3 3  1.22     Ideal     G     VS2  61.4  56.0  8362 6.90 6.88 4.23 200.80656
## 5 5  0.32     Ideal     D     SI1  62.7  54.0   589 4.35 4.39 2.74  52.32441
## 6 6  0.34     Ideal     E     VS2  62.1  54.1   758 4.47 4.50 2.78  55.91970
## 7 7  0.31     Ideal     H     VS2  62.2  55.0   628 4.37 4.35 2.71  51.51574
## 8 8  1.21 Very Good     G    VVS2  60.4  56.0  9878 6.96 6.91 4.19 201.51218
```

```
##   otlr
## 1    0
## 3    0
## 5    0
## 6    0
## 7    0
## 8    0
```

```
tail(data.noNA.noDup.noStrg.notypo.mkOtlr)
```

```
##       X carat       cut color clarity depth table price    x    y    z
## 92   92  0.27     Ideal     E    VVS2  62.5    57   622 4.10 4.13 2.57
## 93   93  0.31     Ideal     F     VS2  61.3    57   802 4.34 4.30 2.65
## 94   94  1.55     Ideal     F     SI1  61.3    56 11011 7.45 7.49 4.58
## 95   95  0.33     Ideal     G      IF  61.7    58   968 4.42 4.46 2.74
## 96   96  1.07      Good     G     SI1  63.1    59 48610 6.45 6.49 4.08
## 100 100  0.44 Very Good     E     VS1  59.7    59  1090 4.94 5.01 2.97
##        volume otlr
## 92   43.51781    0
## 93   49.45430    0
## 94  255.56629    0
## 95   54.01417    0
## 96  170.79084    1
## 100  73.50572    0
```

Document

During investigation of the relationship between carat vs price, we found two obvious outlier with ID27 and ID96. After detailed diagnosis, she found no evidence showing that those two outliers are clearly errorous. So we added another vector called `otlr` into the dataframe to indicate whether it is an outlier or not.

Bonus Question (Optional)

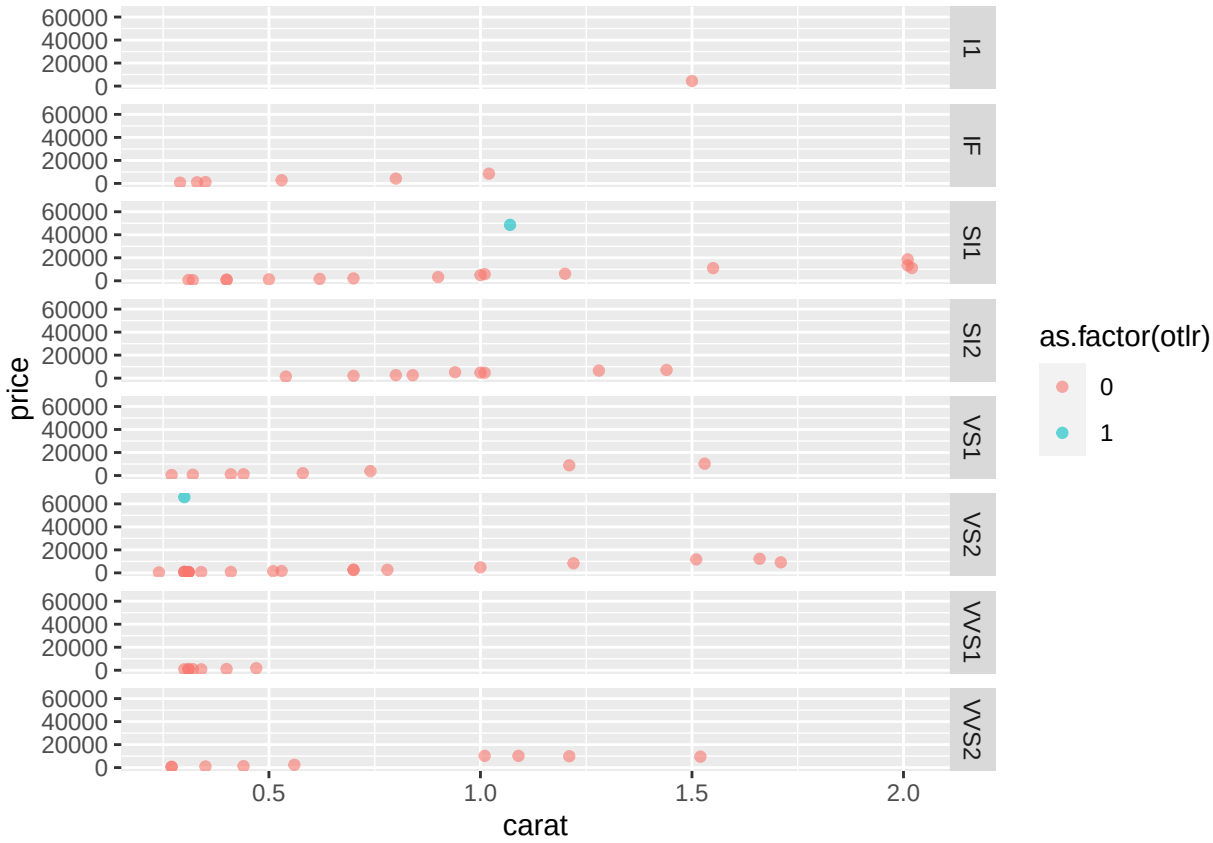
For the outliers you identified above, those have strange pattern of `carat ~ price`, try to make more plot to see whether this strange pattern is actually correlate with other features?

Task: try to plot the relationship between carat ~ price, but separate data points by their clarity. (hint: use `ggplot2, facet_grid()` function).

```
plot.df = data.noNA.noDup.noStrg.notypo.mkOtlr
#the name is too long, let's simplify it a little bit

library(ggplot2)

p = ggplot(plot.df, aes(x = carat,
                        y = price,
                        color = as.factor(otlr))) # in case some points overlap
p+geom_point(alpha=0.6) + facet_grid(clarity~.)
```



Very likely that they are not outliers. What do you think?

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Originally created by Wanlu Liu in 2019.

Last update by Dmytro Shytikov in 2023