DATA CLEANING AND MANIPULATION IN R STUDIO

To apply any statistical model or operation on data first we need to clean our data with different parameters so that we would be able to extract correct information from data. A very usual example of this cleaning is the detection and removal of outliers. Having outliers in data make our results biased, a very common example in this context is to find out the average income a person lives in Mumbai. If we try to find out the average income of people lives in Mumbai then it would contain the income of Ambani too and our average income would be too high than the actual average income of individual lives in Mumbai. To get a true(nearer) result we should remove Ambani income from the group. Not only outliers in data create problems with result interpretation but even some values prevent us from getting the result. Like, the missing values or string value in numerical columns. Sometimes it would be very difficult to get rid of from these problems and get the correct result.

This paper will guide you through the methods of dealing with all such problems that can occur to you at the time of data analysis, this entire paper is based on R Studio software and all the analysis is included in the R Script that can only be executed in R Studio(I have used R version 3.5.1). The data that I've used for this analysis is 'Survey' data that is available in 'MASS' package of R and this can be installed from the R repository.

- Install.package("MASS", dependencies = T, repos = 'https://cran.r-project.org/index.html')
- library('MASS')

Above command will install 'MASS' package in your system and enable it for your usage(until R Studio is running).

Now, let's call the dataset that we require

- data(survey)
- View(survey)

These commands will import the survey data into R Studio and present it in R Script section.

Here you can see that in this data set there are many values which are missing(named as 'na') and will create a problem if you will try to run a statistical or mathematical operation on it. As if you will try to compute the mean of a row that contains an 'NA' value it will return you the result as 'NA'(Since R won't be able to calculate with a string value).There are plenty of ways to deal with this problem in one case we can use 'na.omit' command that will remove the entire row from the data. But sometime we won't have enough data and we can't remove the entire row(one's responses) just because a single value in that row is missing.

Our first job here is to find whether our data or columns in the data have missing values, if yes then how many. So that we would be able to decide what treatment can we adapt to deal with missing values.

To check whether my data column contains missing values apply:

- Is.na(data$column name) # It will return you “T” at all places where it finds an NA value and “F” where it will find non “NA”.

But here comes a problem, this function only detects if the value is "NA" but what if we will have a blank value? For that, we first have to change all blank values to "NA"(the same operation will work with any other value).

Change blank values to “NA”

- Data[Column name==""] <- NA # It will change all blank values to “NA” and then we can easily remove “NAs”.

Now you can apply the previous formula to see how many NAs are there in your data. One question still remains, I have to find out how many missing values are actually there(Count of missing value).

To find count of missing values:

- sum(is.na(data$column name)) or
- table(is,na(data$column name)),

Best way to detect number of missing values in each row is :

- colsums(is.na(data)) # It will give you the count of all missing values in each columns.

Also, if we have enough data and fewer missing values we can remove those rows by:

- na.omit(data)) # or,
- data[complete.cases(data),] # it will keep only those rows that have no missing values.

So far we have seen the scenarios in which we can remove the rows containing missing values but sometimes removing rows is not an option and we have to generate those missing values. In this scenario of generating new values in the dataset we have to keep in mind a few things about data type, data distribution, and skewness.

The scenario in which my data column that contains missing value is normally distributed in that case best suitable value to fill for missing value is the mean value of rest of the values. But, how do we decide whether that data is normally distributed or not?

The best way to check whether the data is normally distributed or not is plot and the best plot for this is a histogram.

To plot histogram in R:

- hist(data$column_name)

This will show you the graph of your column and now you have to check whether the distribution of your data is normal or not, if you find higher bars on the center and adjacent bars symmetric then your data is normally distributed and we can fill up missing values with mean value. To insert mean value first

we have to compute it, and the best way to compute the summary of your dataset is through summary command.

- Summary(data) # it will give us all the details(statistical) about your dataset.

Now, if our data is normally distributed than fill-up missing values with the mean value:

- Data[column name == “”] -> mean

What if my data is not normally distributed and skewed towards a side, in that case, we should replace missing values with the median value,

- Data[column name == “”] -> median

So far we have discussed the continuous data but what if my data is categorical? In that case, best value to replace with missing value is the mode of that table.

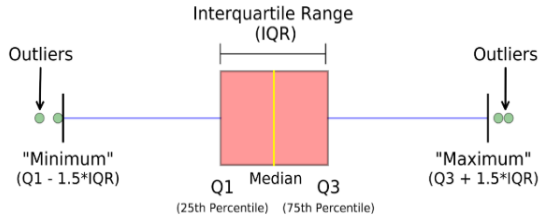
- Data[column name == “”] -> mode

Now we have dealt with missing values in our data, the next step is to detect outliers. Outliers are extreme values that change your results by a significant amount and lies in the extent of data.

The best(Conventional) method to detect outliers in the dataset is plotting boxplot(whisker plot) and you can see the dots outside from the plots which are to be considered as outliers. In the analysis of data or computation of any statistical result from data first, we have to remove these values.

To plot boxplot:

- Boxplot(data[column name])



In the boxplot, all the outliers would come out as dots from the plot(Kind of brick with two tails), As shown in this picture.

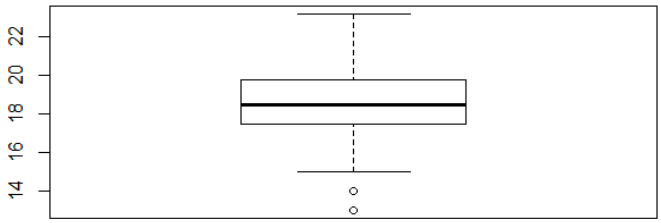
The brick represents entire data and both tails are the limits for which we are allowed to take the values from. These tails represent the range of both side as the maximum value and minimum value till then we can accept our data value.

Values	**Meaning**	**Relevance**
Minimum Value	The minimum range till we can expect a value to be included in the analysis.	Any value lesser than this should consider an outlier.
Q_1	First 25^{th} percentile of the data.	First quarter of the data.
Median	The mid value of the data	Half of the data.
Q_3	75^{th} percentile of the data	$3/4^{th}$ of the data.
Maximum Value	The maximum range till we can expect a value to be included in the analysis.	Any value greater than this should consider an outlier.

The minimum value that we can consider as to be the part of our data should be greater than $Q_1 * 1.5$ IQR and the maximum value $Q_3 * 1.5$IQR, here is Interquartile range and this is calculated by $Q_3 – Q_1$. The line in the mid is the median of the data.

As if we plot a boxplot for the column ‘Wr.Hnd' from survey data we see that there will be two dots below the lower tail(value).These two dots should be considered as outliers when we do analysis and keeping this column as the part of the analysis.

- Boxplot(survey$ Wr.Hnd)



Next step is to remove these outliers from data while conducting the analysis. Or exclude these values.

Now there are two simple ways through which you can remove outliers from data, first, a user-defined function that will assign ‘NA' to outliers and then you can replace or remove these ‘NA’ values as you want or, Second, a very useful package “Outliers” that not just remove the outliers from dataset but gives you the liberty to replace these values from any statistical parameter mean or median.

User defined function:

```
rm.outliers <- function(data, na.rm = TRUE, ...) {
  qnt <- quantile(data, probs=c(.25, .75), na.rm = na.rm, ...)
  value <- 1.5 * IQR(data, na.rm = na.rm)
  data2 <- data
  data2[data < (qnt[1] - value)] <- NA
  data2[data > (qnt[2] + value)] <- NA
  data2
}
```

Description:

A user-defined function named rm.outliers in which I passed the data as ‘data'(A default value that is used for data insertion),

‘qnt’ variable that contains two quartile values first and third.

‘value’ variable that is the value from which we can get upper and lower limit of our dataset(by subtracting and adding first and third quartile value).

Then value replacements after subtraction and addition for limits.

So, after all this, you can apply this function to your dataset and it will refine your dataset and will replace outliers with ‘NA' values. After which you can either remove or replace these ‘NA' values.

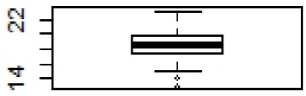
Another way to deal with outliers is the use of "Outliers" package. This package contains all necessary functions and features to deal with outliers in the dataset. You can not only remove outliers with the help of this package but you can also replace these values with mean and median.

- outlier(x, opposite = FALSE, logical = FALSE) # Function to show you outliers in your dataset. Outlier function actually finds out the largest difference value from the mean.
  Here,
  x; is the dataset in which you search for outliers.
  Opposite: if set to be true then it will find out the lower(Minimum) outlier value while if this is set to be true than it will give you the higher(maximum) value in the dataset.
  Logical: It gives logical values and wherever it finds(not) outlier it mark it as True(False) if this condition set to be true.

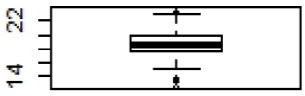
But there is a limitation of this package and I would recommend users to use the user-defined function since "Outliers" package is only able to detect single upper or lower value at a time while user-defined function is capable of converting all outer values in a single click. Using outlier function does not follow any statistical conditioning and gives you directly the distant value from mean at both ends. Even if there is no outlier in the dataset still it will give the value since it does not follow the rule of IQR.

Let see all outliers in dataset 'survey':

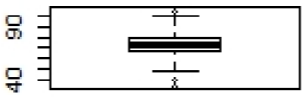
- x11(height = 40, width = 40)# will pop up a new window where graph will appear,
- par(mfrow = c(3,2))# Matrix to represent graph's order,
- boxplot(survey$Wr.Hnd)
- boxplot(survey$NW.Hnd)
- boxplot(survey$Pulse)
- boxplot(survey$Height)
- boxplot(survey$Age)# five boxplots, as shown in picture below



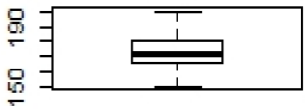
survey$Wr.Hnd



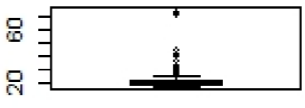
survey$NW.Hnd



survey$Pulse



survey$Height



survey$Age

If you would look closely to the previous diagrams then you will find out that there are no outliers in the 4th(survey$Height) diagram. But, still, if you would apply outlier function to it than it will give you a value which is highly distant from the mean of that column. This is the reason I would recommend user to use above user-defined function instead of "outlier".

Now, we have detected the columns that contain outliers and the methods that we can execute to remove these values.

Our last task is to either remove or replace these outliers.

Let's apply our above function to replace outliers,

- rm.outliers(survey$Wr.Hnd)
- rm.outliers (survey$NW.Hnd)
- rm.outliers (survey$Pulse)
- rm.outliers (survey$Height)
- rm.outliers (survey$Age)

These commands will replace all the values to 'NA' that should not be considered as the part of this data column(Outlier). If you try to execute the last command to replace outliers in the Age column than it will give you 21 'NA' values and then you can replace these values through any method discussed at the beginning of this paper. Similarly, if you apply same formula for height column it won't change anything because of there no outlier in this column. I tried to run the outlier command from Outliers package and that gave me value 200 which not an outlier but the farthest value from the mean of this column.

At the end of this paper I think the missing value treatment and data filtering is clear, it is good to create a function as per your need, R gives you that liberty to create any function that you require at a moment this not only helps you get the exact result you wanted but also helps you to focus your mind into different algorithms than memorizing multiple packages. I think this paper helps in data cleaning and filling by right values to the readers please notify me with your question and feedbacks.

References:

- van der Loo, M. and de Jonge, E., 2018. *Statistical data cleaning with applications in R*. John Wiley & Sons.
- De Jonge, E. and van der Loo, M., 2013. An introduction to data cleaning with R. *Heerlen: Statistics Netherlands*.
- Adler, J., 2010. *R in a nutshell: A desktop quick reference*. " O'Reilly Media, Inc.".